

ABSTRACT

Title of dissertation: AN INITIAL EVALUATION OF IBI VIZEDIT: AN RSHINY APPLICATION FOR OBTAINING ACCURATE ESTIMATES OF AUTONOMIC REGULATION OF CARDIAC ACTIVITY

Matthew George Barstead, Doctor of Philosophy, 2018

Dissertation directed by: Professor Kenneth H. Rubin
Department of Human Development & Quantitative
Methodology

Photoplethysmogram (PPG) sensors are increasingly used to collect individual heart rate data during laboratory assessments and psychological experiments. PPG sensors are relatively cheap, easy to use, and non-invasive alternatives to the more common electrodes used to produce electrocardiogram recordings. The downside is that these sensors are more susceptible to signal distortion. Often, the most relevant measures for understanding psychological processes that underlie emotions and behaviors are measures of heart rate variability. As with all measures of variability, outliers (i.e., signal artifacts) can have outsized effects on the final estimates; and, given that these scores represent a primary variable of interest in many research contexts, the successful elimination of artefactual points is critical to the ability to make valid inferences with the data. Prior to the development of IBI VizEdit, there was no single, integrated processing and editing pipeline for PPG data. The present pair of studies offers an initial evaluation

of the program's performance. Study 1 is focused on the efficacy of a novel approach to imputing sections of particularly corrupted PPG signal. Study 2 tests the ability of trained editors to reliably use IBI VizEdit as well as the validity of estimates of cardiac activity during a prescribed set of laboratory tasks. Study 1 suggests that the novel imputation approach, under certain conditions and using certain parameterizations may hold promise as a means of accurately imputing missing sections of data. However, Study 1 also clearly demonstrates the need for further refinement and the consideration of alternative implementations. The results from Study 2 indicate that IBI VizEdit can be reliably used by trained editors and that estimates of cardiac activity derived from its output are likely valid.

AN INITIAL EVALUATION OF IBI VIZEDIT: AN RSHINY APPLICATION FOR
OBTAINING ACCURATE ESTIMATES OF AUTONOMIC REGULATION OF
CARDIAC ACTIVITY

by

Matthew G. Barstead

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2018

Committee Members:

Professor Kenneth H. Rubin, Ph.D. (Primary Advisor)

Professor Andrea Chronis-Tuscano, Ph.D. (Dean's Representative)

Professor Jeffrey R. Harring, Ph.D.

Assistant Professor Alexander J. Shackman, Ph.D.

Assistant Professor Nicholas J. Wagner, Ph.D.

©Copyright by

Matthew G. Barstead

2018

ACKNOWLEDGEMENTS

There are so many people to thank in terms of getting me here to this final step in my Ph.D. program. I will start with my committee members, who during my dissertation defense stepped in and saved me from myself. I had initially proposed three studies and they wisely suggested that I might have been over-reaching given my proposed timeline for completion. With how long it took to put this document together as it was, I will be forever glad I agreed with their assessment and took their advice.

I would also like to single out my advisor Ken Rubin. Not only did he admit me into the program, seeing a future successful graduate student in my application materials when no one else did, he provided countless opportunities to grow as a researcher. More than anything though, I am glad he offered me the space I needed to find a scientific problem I wanted to solve. He knew when to interject himself and provide more direct guidance and when to hang back and let me go through the process of learning and growing on my own. Ken, thank you always for being so well-balanced as a mentor and for giving me the tools and opportunities to find my own path.

Then of course there are my parents, Tara and Greg. My mother is a consummate teacher whose patience and ability to incite and foster curiosity have had outsized effects on the man and researcher I am today. My father is one of the most intelligent people I know. He is circumspect in his thinking, able to identify broad patterns in complex data, and capable of making his point with unrivaled wit and clarity. They have stood as incredible examples for me throughout my life and have provided me with every form of support imaginable during my winding professional and academic career. It is hard to know a counterfactual, but I am fairly certain I would not be in this position had they

made different choices or elected to sacrifice less in their efforts to pick me up when I stumbled and to clear life's path of what debris they could. I love you both, and thank you for everything you have done and continue to do for me. There is no way this former college dropout earns a Ph.D. without the support you have both offered in the intervening years.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	ii
TABLE OF CONTENTS	iv
LIST OF FIGURES	vi
LIST OF TABLES	viii
CHAPTER 1	1
CHAPTER 2	7
Linking the Mind and the Body: The Role of the Autonomic Nervous System in Emotion and Emotion Regulation	7
Polyvagal Theory	10
Neurovisceral Integration Model	13
Integrating Polyvagal Theory and the Neurovisceral Integration Model	14
Measuring Heart Rate Variability	15
Electrocardiogram	15
Photoplethysmography	18
Understanding Common Measures of Cardiac Activity	21
Heart Rate/Heart Period	21
Standard Deviation of the Normal-Normal Interval	21
Root Mean Square of Successive Differences	22
Respiratory Sinus Arrhythmia & High-Frequency Heart Rate Variability	23
Tonic vs. Dynamic Measures of Variability	25
Current Editing and Analytic Strategies	27
Identifying and Managing Low Quality Data	29
Introducing <i>IBI VizEdit</i>	30
Summary and Current Research	31
CHAPTER 3	34
An Overview of <i>IBI VizEdit</i>	34
Peak Detection Algorithm	34
Manual Editing	38
Gaussian Process Models	41
Implementing Gaussian Process Models in <i>IBI VizEdit</i>	45
Research Overview	47
Study 1: Evaluating the Novel Imputation Tool in <i>IBI VizEdit</i>	48
Overview	48

Participants.....	49
Procedures.....	50
Data Re-sampling Design: Testing the Three Gaussian Process Models	52
Measures	55
Hypotheses and Analyses.....	57
Study 2: Reliability and Validity of IBI VizEdit Outputs	58
Study 2 Overview	58
Participants.....	59
Procedures.....	60
Measures	65
Hypotheses and Proposed Analyses.....	73
CHAPTER 4	75
Study 1 Results	75
Study 2 Results	91
CHAPTER 5	107
Study 1: Is there Evidence to Support the Use of Gaussian Process Models?	107
Study 2: Can Trained Editors reliably use <i>IBI VizEdit</i> ?	117
Study 2: Can <i>IBI VizEdit</i> be used to produce valid estimates of heart rate and autonomic regulation of cardiac activity in a laboratory setting?	120
Study 2: Can estimates of parasympathetic regulation of cardiac activity derived from files edited in IBI VizEdit be used to predict children’s social behavior in a real-world context?	123
Conclusions.....	128
APPENDICES	129
Appendix A: Programming Components of IBI VizEdit.....	129
Appendix B: Model and Data Analysis/Summary Code	138
Appendix C: Code Used to Generate Figures.....	145
Appendix D: Code for Re-Sampling Program.....	172
REFERENCES	208

LIST OF FIGURES

Figure 2.1 Typical Heartbeat as detected by electrocardiogram recordings	17
Figure 2.2 Screenshot of photoplethysmogram signal with artifacts in IBI VizEdit	19
Figure 2.3 AcqKnowledge screenshot of simultaneously collected electrocardiogram (middle) and photoplethysmogram (top) data.....	20
Figure 2.4 Spectral analysis of photoplethysmogram-derived interbeat intervals.....	25
Figure 3.1 Effects of pre-processing photoplethysmogram data with random noise.....	35
Figure 3.2 Identifying peaks and troughs in periodic data using IBI VizEdit's peak detection algorithm	36
Figure 3.3 Results of iterative peak detection algorithm applied to simulated data	37
Figure 3.4 Screenshot depicting peak detection algorithm performance with data including artifacts.....	38
Figure 3.5 Screenshot depicting peak detection algorithm performance with artifact-free data.....	38
Figure 3.6 Screenshot depicting misidentification of an early heartbeat	40
Figure 3.7 Screenshot depicting IBI signal deflection that requires averaging of two points.....	41
Figure 3.8 Screenshot depicting failure to identify a heartbeat	41
Figure 3.9 Intervention study timeline	48
Figure 3.10 Graphic depiction of varying conditions during physiological assessments	51
Figure 3.11 Depicting of Gaussian process imputation model testing.....	54
Figure 3.12 Respiratory sinus arrhythmia distributions by task	66
Figure 3.13 Heart period distribution by task	67
Figure 3.14 RMSSD distributions by task	69
Figure 3.15 Distributions of proportion of 10-second epochs children engaged in each coded play behavior	72
Figure 4.1 Convergence of Gaussian process models under varying conditions as measured by scale reduction factor of the model-estimated heart rate parameter	76
Figure 4.2 Model run time for each Gaussian process model under varying conditions.	84
Figure 4.3 Model run time as a function of total time to be imputed	86
Figure 4.4 Distributions of imputation-derived RMSSD deviations from original values	87
Figure 4.5 Distributions of imputation-derived RMSSD deviations from original values from models with adequate convergence.....	90
Figure 4.6 Distributions of the proportion of edits required by task.....	92
Figure 4.7 RSA difference scores for primary editors: Comparing editors' RSA values from IBI VizEdit and Cardio Edit.....	93
Figure 4.8 RSA difference scores for primary editors: Comparing RSA values from a third editor using IBI VizEdit	95
Figure 4.9 Vizualization of missing data for RSA values	96
Figure 4.10 Vizualization of missing data for IBI VizEdit-Generated values	97
Figure 5.1 Relatively successful imputation (in red) compared with original values (in black).....	108
Figure 5.2 Example of imputed values (in red) that are offset from original values (in black).....	109

Figure 5.3 Example of a failed imputation run (in red) relative to the original signal (in black).....	110
Figure 5.4 Divergence in hyperparameter estimates for Gaussian process model 2 based on absolute differences in estimates from two sets of random starting values	112
Figure 5.5 Divergence in hyperparameter estimates for Gaussian process model 3 based on absolute differences in estimates from two sets of random starting values	113
Figure 5.6 Example of Gaussian process model with incorrectly specified length parameters that are too short	114
Figure 5.7 Example of a Gaussian process model with incorrectly specified length parameters that are too long.....	115
Figure 5.8 Comparing RSA scores from files edited in Cardio Edit vs. IBI VizEdit.....	119
Figure 5.9 Within-task correlations of physiological variables derived from edited interbeat interval files	122

LIST OF TABLES

Table 3.1	68
Table 4.1	77
Table 4.2	78
Table 4.3	79
Table 4.4	80
Table 4.5	81
Table 4.6	82
Table 4.7	98
Table 4.8	99
Table 4.9	99
Table 4.10	100
Table 4.11	101
Table 4.12	102
Table 4.13	102
Table 4.14	103
Table 4.15	103
Table 4.16	104
Table 4.17	105
Table 4.18	105
Table 4.19	105
Table 4.20	106
Table 4.21	106

CHAPTER 1

The two studies described herein tested a recently developed program designed to process, and allow for the manual editing of, heart rate data. The overarching aim of these studies is best characterized as a preliminary methodological investigation of this new program's capacity to return reliable and valid estimates of heart rate variability. Thus, the two studies do not represent an exhaustive evaluation of this new program. Instead, the investigations balanced the assessment of reliability and validity of the program's outputs with the testing of issues related to the program's applied use with data collected as part of an ongoing intervention study for extremely shy and inhibited preschoolers.

The newly created, open-sourced program, *IBI VizEdit* (Barstead, 2018) was designed specifically for the processing and manual editing of photoplethysmographic heart rate data (in contrast to heart rate data obtained using the more common electrocardiogram). Photoplethysmogram sensors are typically attached to an individual at peripheral sites of the body (such as a fingertip or ear lobe) and transduce changes in light absorption to estimate blood volume flowing through local capillary beds (Allen, 2007). These non-invasive sensors have several practical advantages over sensors that may offer a more precise measure of cardiac activity (e.g., electrocardiogram). For instance, they are easy to use and can be correctly attached to participants with minimal training. However, there are some downsides. In contrast to electrocardiogram sensors that measure the heart's electrical activity, photoplethysmogram sensors are more susceptible to a variety of artifacts (Couceiro, Carvalho, Paiva, Henriques, & Muehlsteff, 2012; Sukor, Redmond, & Lovell, 2011). For sensors attached to a participant's fingertip (as was the case in the present data sets), movement artifacts, in particular, can present

non-trivial editing challenges as changes in blood volume become a combination of the effects of (a) true cardiac activity, (b) volumetric changes due to the position of the hand in relation to the heart, and (c) changes in the pressure of the sensor on the local capillary bed (i.e., if the participant depresses or alters the relative location of the sensor on his or her fingertip). The waveform of the photoplethysmogram signal further complicates matters as it lacks the characteristic and easily identifiable spiking used to mark heart beats in electrocardiogram recordings (Sukor et al., 2011).

Problems arising from artifacts are neither new to the collection of physiological data, in general, nor are they new to heart rate data in particular (Berntson et al., 1997; Berntson, Quigley, Jang, & Boysen, 1990; Berntson & Stowell, 1998; Kaufmann, Sutterlin, Schulz, & Vogele, 2011; Kim & Yoo, 2006; Lee, Lee, Jung, & Lee, 2007). However, most editing approaches designed to correct for these artifacts were designed based on signals obtained from electrocardiogram sensors (Berntson et al., 1997; Kaufmann et al., 2011; Porges & Byrne, 1992). Recently, automated machine learning approaches have been applied to electrocardiogram data and have shown promise as a potential replacement of the current editing gold standard, which is the more arduous manual identification and removal of artifacts (Hegarty-Craver et al., 2017). However, it is not clear that these approaches are appropriate for use with photoplethysmogram data, which produce a slower oscillating waveform than the more pronounced rapid onset signal of the heart's electrical activity. Moreover, the detection of heartbeats, and the subsequent calculation of time between successive beats (i.e., interbeat intervals) that are the focus of numerous analyses and calculations, requires a modified approach when working with photoplethysmogram data, again raising questions about the wholesale

adoption of editing techniques that have been developed for use with electrocardiogram data. Given the proliferation of the use of photoplethysmogram sensors due, in part, to their practical advantages over electrocardiogram electrodes, developing appropriate processing and editing techniques for these data represents an important and timely contribution to the field of psychophysiological research.

One modest goal of the program is to eliminate the need for relying on multiple custom-built programs to separately complete each of these steps in the editing process. To that end, *IBI VizEdit* offers a comprehensive editing interface that allows users to seamlessly process raw data using an adaptive peak detection algorithm, employ a graphic interface for the identification and removal of artifacts, and save edited files and editing summaries for subsequent analyses. In addition to these functional improvements, a key novel feature of *IBI VizEdit* is its inclusion of an advanced imputation tool built using Gaussian process models within a Bayesian estimation framework. In theory, this imputation tool has the potential to provide users with a series of suggested edits that better preserve the naturally-occurring variation in the timing of heart beats than would otherwise be possible with standard editing approaches.

There are multiple aspects regarding the use of *IBI VizEdit* that require examination, not all of which can be evaluated in a single pair of studies. As a result, particular components were targeted for this initial set of evaluations. The first of the two studies focused primarily on the effectiveness of the aforementioned imputation model. Three different parameterizations of the imputation model were tested under varying modeling and data collection conditions (e.g., amount of time to be imputed). Effectiveness of the imputation models was evaluated via the repeated sampling and

imputation of photoplethysmogram data that were visually confirmed to be artifact free. In this first study, measures of heart rate variability from the original (i.e., “true”) interbeat interval values were compared with measures of heart rate variability derived from imputed data. Imputed values were also compared with two relatively naïve imputation solutions – a mean replacement and a hotdeck random replacement strategy, both of which involve well-known statistical downsides but are easy to implement from a statistical programming standpoint. If the imputation models were not capable of producing less biased and more precise estimates of cardiac activity than these simpler, imperfect replacement strategies, there may be little practical benefit to their use.

The imputation tool included in *IBI VizEdit* is exactly that, a tool, and it represents a small, though potentially important component of the program. In practice, a human editor is still required to evaluate and accept or reject the model-implied values. The Gaussian process models tested in the present report were never intended to replace manual editing; they were intended to enhance the manual editing strategies available by adding a model-based approach. Indeed, the bulk of editing expected to occur while using *IBI VizEdit* involves editing techniques long incorporated in established editing software. These main editing functions rely on subjective appraisals made by trained editors. With only visual cues and general guidelines to rely upon, editors must make decisions about adjusting or not adjusting the relative location of interbeat intervals. As with any human activity in research, this process introduces a potential source of error to final scores as a function of variation in editor-to-editor appraisals of each file.

Thus, in addition to evaluating the novel imputation tool included in *IBI VizEdit*, the present report also included an examination of the primary editing interface when

used to edit data collected in an ongoing intervention study. As part of this ongoing research, heart rate data were collected during three different laboratory assessments across the intervention period. During the assessment, children (and their parents) were exposed to a series of conditions expected to cause changes in their emotional and cognitive states – which would ultimately be manifested in changes in cardiac functioning. These data presented an opportunity to evaluate the ability of trained editors to use the program reliably. Reliability was assessed in two primary ways. The first was the degree to which different editors, using *IBI VizEdit*, arrived at similar cardiac activity scores. The second was the degree to which the same editors arrived at equivalent scores using *IBI VizEdit* and a similar, more established, program *Cardio Edit* (Brain-Body Center, 2007).

Of course, *reliability* (i.e., the degree to which an instrument returns a measure free from error) is an important property to demonstrate in a new tool. However, the ability to generate valid inferences from the program's outputs is equally vital to its use in research. To that end, *validity* (i.e., the degree to which an instrument measures the phenomenon of interest) was assessed in two ways. The first evaluation involved testing whether a series of prescribed laboratory tasks produced expected changes in heart rate and heart rate variability in a selected sample of young children using *IBI VizEdit*. The second effort to establish validity focused on the ability of heart rate and heart rate variability scores from the aforementioned assessment conditions to predict theoretically related social behaviors among those same children in a classroom setting.

In sum, the two studies sought to provide a foundation from which to continue to build and refine *IBI VizEdit*. The first study evaluated a novel implementation of

Gaussian process models to assist editors in making statistically informed edits to sections of particularly problematic data. The second study evaluated *IBI VizEdit*'s ability to return reliable and valid estimates of cardiac activity during a series of prescribed laboratory tasks. The goal of this, still nascent, program of research was to design a completely open-sourced program for editing heart rate data that can be easily, reliably, and validly used by trained editors. These two studies represents a critical first step in meeting this broader objective.

CHAPTER 2

What kind of an emotion of fear would be left if the feeling neither of quickened heart-beats nor of shallow breathing, neither of trembling lips nor of weakened limbs, neither of goose-flesh nor of visceral stirrings, were present, it is quite impossible for me to think ... I say that for us emotion dissociated from all bodily feeling is inconceivable.

- Williams James

Linking the Mind and the Body: The Role of the Autonomic Nervous System in Emotion and Emotion Regulation

Walking home late at night, you sense a vague shape in your periphery. Suddenly a loud noises screeches through the blackness, emanating from the same direction. As information passes along sensory fibers, neural circuits work to make sense of the input. Messages carried along cranial nerves cause muscles in the neck to relax and contract, pivoting the body's two primary sensory systems (vision and hearing) toward the amorphous noise-generating shape. At effectively the same moment as your eyes and ears are positioned to acquire additional information about the threat, your heart rate begins to increase, as does your pace of breathing. Alterations to the dilation of blood vessels subsequently divert the resulting influx of oxygenated blood away from non-essential systems as the body becomes primed to fight or flee.

On its own, the brain cannot enact a response to a lurking shape in the darkness. It cannot initiate a sexual encounter or acquire its own energy. Brains can, however, coordinate a diverse array of biological systems in the service of a unified goal. Neural regulation of internal biological systems takes many forms and relies on several different

levers, two of the most important of which are the *parasympathetic* and *sympathetic* branches of the *autonomic nervous system*. When the neural machinery involved in attention and emotion regulation pulls on these levers, it leaves traces; traces that await a good detective with the right set of tools.

Over the years, these tools have grown to include measures of salivary cortisol (Hellhammer, Wüst, & Kudielka, 2009), galvanic skin response (Vetrugno, Liguori, Cortelli, & Montagna, 2003), blood pressure (Chida & Hamer, 2008), and heart rate variability (Beauchaine & Thayer, 2015), among others. To varying degrees and in different ways, each of these measures taps the functioning of more centrally located neural circuits responsible for emotional and behavioral regulation. Not surprisingly, cardiac activity figures prominently on this list of meaningful peripheral measures. Once thought to be the locus of emotion, the heart may not be involved in the active processing of emotional sensations, but variation in its activity is certainly related to emotional states. Indeed, it is hard to think of a system more central to the acquisition and distribution of energy stores than the cardiovascular system. As a result, changes in cardiac output underlie a number of reactions, including emotional responses.

Reciprocal connections, various self-limiting feedback loops, and distinct signaling pathways create a network of neural-autonomic-cardiac connections that provide moment-to-moment regulation of cardiac output. At rest and in the absence of threat, most of that regulation takes the form of tonically active inhibitory signaling along the myelinated *vagus nerve*, part of the parasympathetic nervous system. The vagus nerve is the 10th cranial nerve and is largely responsible for managing communication between the body's viscera (i.e., organs) and the brain (Haines & Mihailoff, 2013). The cell bodies

that make up the vagus nerve originate in the brain stem and synapse onto the heart at the sinoatrial node, which is sometimes described as the heart's internal pacemaker. Without tonic inhibitory signaling along the myelinated vagus, sinoatrial pacing would result in higher baseline heart rate (Porges, 2007). With upstream connections to limbic and cortical structures (in addition to reciprocal connections with other brainstem nuclei) and downstream connections with the heart and other viscera, the vagus nerve can be considered a neural-visceral bridge of sorts. Importantly, monitoring activity along this anatomical connector can provide insight into the overall functioning of the entire internal regulatory system.

To better understand the myelinated vagus' role in regulating cardiac output, Porges has often used the metaphor of an automobile's acceleration and braking systems (Doussard-Roosevelt, Montgomery, & Porges, 2003; Porges, 1995, 2007; Porges, Doussard-Roosevelt, Portales, & Greenspan, 1996). Imagine a car with a jammed accelerator pedal (the sinoatrial node) incapable of completely returning to an idling position. Under this system of control, maneuvering along narrow, local streets requires constant application of the brake (vagus) to maintain a safe speed. Circumstances may arise that require careful course adjustments, and further application of the brake (vagus) can be used to improve control under these conditions. Occasionally expediency, or the presence of danger, requires a rapid response. In these instances, the brake (vagus) can be released allowing the car to increase speed quickly up to the point of the stuck accelerator pedal (sinoatrial node). In moments that are especially intense, the vehicle's speed can be increased even further by depressing the accelerator (via sympathetic inputs).

Two theoretical frameworks, one rooted in evolutionary psychology (Polyvagal Theory; Porges, 1995, 2007), the other in dynamic systems theory (Neurovisceral Integration Model; Smith, Thayer, Khalsa, & Lane, 2017; Thayer, Hansen, Saus-Rose, & Johnsen, 2009; Thayer & Lane, 2000) emphasize slightly different aspects of vagal control of heart rate. Despite their differences, much of the evidence that supports one framework is entirely consistent with the other and both make predictions supported by data. Most importantly, both theories argue that it should be possible to observe effects of more centrally located neural control circuits in peripheral measures of cardiac activity.

Polyvagal Theory

Porges' (1995, 2007) Polyvagal Theory rests on a mix of phylogenetic, anatomical, and functional evidence showing that vagal tone is the output of a complex neural circuit involved in regulating the autonomic nervous system. As a starting point, Polyvagal Theory characterizes emotional responses as experiences entirely dependent on reciprocal connections between the brain, the autonomic nervous system, and interoceptive sensory signals (i.e., sensory information about internal states). Moreover, the theory suggests that evolutionary processes have resulted in changes to the connectivity and functioning of this neural-autonomic-visceral network.

Early on in the evolution of vertebrate species, the vagus nerve primarily supported the behavioral response strategy of freezing in the presence of perceived threats (Porges, 2001, 2003). Freezing is a relatively common response strategy across species that involves a drastic reduction in metabolic output and activity, including a slower heart rate. Freezing can be a risky gambit, though, as the response strategy does little to distance the organism from the source of the potential threat. Clearly a wider

range of behavioral responses would be evolutionarily beneficial; however, such responses require control systems and signaling pathways that can be utilized to simultaneously effect change on multiple systems.

With the phylogenetic development of the sympathetic nervous system, evolutionary pressures resulted in a new (generally excitatory) communication pathway linking the brain and the body. Through signaling along the unmyelinated vagus nerve, organisms could rapidly slow down the body's metabolic activity. The advent of the sympathetic nervous system meant that organisms now had an option to significantly increase their metabolic output for a brief period. Thinking through what the two systems meant for behavioral response options, organisms at this evolutionary stage could now respond to a threat by freezing (unmyelinated vagus), fleeing (sympathetic inputs), or fighting (sympathetic inputs). One downside of the functioning of this newer system is that it can be energetically costly to activate the sympathetic response, and its activation reduces activity in important ongoing background processes involved in immune response, digestion, and waste removal (Haines, 2013).

It is at this evolutionary crossroads that mammals diverge from other vertebrates, with the development of the myelinated vagus nerve (Taylor et al., 2014). Myelin is a lipoprotein sheathing that provides insulation to neurons and improves the speed at which electrical signals propagate along a neuron's axon (Naftel, Ard, & Fratkin, 2013). The presence of myelin is generally thought to indicate that there is a functional need for rapid signaling along the pathway. Because activation of the sympathetic nervous system often involves dispersal of chemical signaling via the bloodstream, it can also take time for its effects to dissipate, limiting regulatory options. For instance, after a fear-inducing

experience, it can take time to return to a homeostatic baseline. At an evolutionary level, the emergence of the myelinated vagus nerve offered a means of moderately adjusting metabolic output without the costs and limited flexibility afforded by signaling along sympatho-excitatory circuits.

Porges (1995) viewed the development of the myelinated vagus nerve in mammals as driven primarily by the need to increase the number of adaptive response options during social interactions. As the size and complexity of a species' social group increases, it is increasingly challenging to automate social responses. Humans, in particular, have extended their behavioral repertoire well beyond the simpler options of freezing, fighting, and fleeing. According to Porges, evolutionary pressures related to successful engagement in complex social milieux are a chief reason why the myelinated vagus nerve is found only in mammals, which tend to be more social species with longer periods of offspring dependency than most other vertebrates.

Porges relied heavily on anatomical evidence in making his case that the myelinated vagus nerve is, first and foremost, designed to support effective social engagement with conspecifics. A particularly convincing line of evidence is the evolutionary co-location of cell bodies implicated in vagal control over heart rate with the cell bodies of other cranial nerves that innervate striated muscle in the face (facial expression), muscles in the neck (orienting to sounds and sights), muscles in the middle ear (modulating auditory perception) and muscles surrounding the larynx and pharynx (speech production and prosody) (Porges, 1995). The myelinated vagus nerve's signaling speed and its integration with other neural structures involved in emotional and social expressions results in a system that can simultaneously tune internal states and outward

expressions to match environmental demands. Unsurprisingly, ineffective signaling along this pathway is associated with a number of emotional and social maladies such as depression (Gentzler, Santucci, Kovacs, & Fox, 2009; Rottenberg, Wilhelm, Gross, & Gotlib, 2002), anxiety (Thayer, Friedman, & Borkovec, 1996), and loneliness (Grippo, Lamb, Carter, & Porges, 2007).

Neurovisceral Integration Model

In contrast to focusing on the evolutionary origins of the myelinated vagus nerve, the Neurovisceral Integration Model (Thayer & Lane, 2000) focuses on the dynamic systems involved in the neural processing and regulation of emotions. The model posits that measures such as heart rate variability, cortisol, and blood pressure represent the outputs of a neural network involved in cognitive and affective regulation (Park & Thayer, 2014; Smith et al., 2017; Thayer et al., 2009). Referred to as the central autonomic network, this interconnected set of neural structures is charged with regulating the activity of the heart and other viscera via parasympathetic (generally inhibitory) and sympathetic (generally excitatory) signaling pathways. According to this view, heart rate variability represents a distal measure of the overall functioning of the central autonomic network.

The Neurovisceral Integration Model places particular importance on the interconnected neural and autonomic structures that underlie emotion reactivity and regulation (Smith et al., 2017; Thayer et al., 2009). This emphasis is the result of investigations that have shown that autonomic activity is modulated by a series of direct and indirect paths involving prefrontal, cortical, and subcortical structures (Sakaki et al., 2016; Thayer, Åhs, Fredrikson, Sollers, & Wager, 2012). This network of neural regions

is complex and involves multiple reciprocal connections among its centrally located neural structures as well as connections with sensory neurons relaying information back to the network from the periphery. The complexity of this dynamic system of interlocking circuits and feedback loops should not be surprising given its purpose of coordinating organism-wide states, altering the functioning of nearly every internal system. In fact, the complexity of neural regulation of the autonomic nervous system is one of the reasons Thayer and colleagues relied on dynamic systems theories when developing the Neurovisceral Integration Model.

The implication of the Neurovisceral Integration Model is subtle but powerful: the functional connectivity of a network of neural regions involved in responding to environmental cues can be detected in a signal derived from a relatively distal physiological source: cardiac activity. Within this framework, it is possible then to conceptualize vagal tone as a transdiagnostic biomarker of psychopathology (T. P. Beauchaine, 2015; T. P. Beauchaine & Thayer, 2015). Such a conceptualization has the added benefit of positioning heart rate variability squarely within the boundaries of the National Institute of Mental Health's *Research Domain Criteria (RDoC)* for the diagnosis of mental health disorders (Insel et al., 2010). As a meaningful biomarker within an evolving diagnostic framework then, measures of heart rate variability have the potential to be clinically useful in addition to their value as measures used by basic and applied researchers.

Integrating Polyvagal Theory and the Neurovisceral Integration Model

It is important to stress that the Polyvagal Theory and Neurovisceral Integration Model are largely in agreement with one another. Both theories place importance on the

autonomic nervous system as a critical signaling pathway that links neural and visceral internal regulatory systems. Where Porges (1995; 2007) detailed the different evolutionary pressures that may have influenced changes in the autonomic nervous system (namely the appearance of the myelinated vagus in mammals and the need for increased complexity in social interactions), Thayer and colleagues (Jennings, Allen, Gianaros, Thayer, & Manuck, 2015; Smith et al., 2017; Thayer et al., 2009; Thayer & Lane, 2000) focused primarily on how the central autonomic network makes use of the myelinated vagus nerve when regulating internal states.

These two theories provide an overall framework for understanding *how and why the appropriate and effective processing and editing of heart rate data is important.*

Whether conceptualized as an output of a social engagement system or the central autonomic network, measures of heart rate variability can provide access to relatively cheap, diagnostically relevant, empirically supported, and theoretically grounded information about individuals' general ability to regulate their own internal states, emotions included. In short, these two theories suggest that there are neural fingerprints that can be found in the heart rate signal, provided we know how and where to look. Best practices and guidelines for detecting these fingerprints have been described (Denver, Reed, & Porges, 2007; Huikuri et al., 1999; Jennings et al., 1981), and added to (Laborde, Mosley, & Thayer, 2017; Quintana, Alvares, & Heathers, 2016), but there remains wide variability in methodologies across disciplines interested in quantifying autonomic influence on heart rate.

Measuring Heart Rate Variability

Electrocardiogram

The most recognizable technique used to collect heart rate is, arguably, *electrocardiography* (ECG or EKG), which measures the electrical activity of the heart. The signal returned by ECG sensors includes a series of features that can be used to derive multiple indicators of cardiac activity on a very precise timescale (Berntson & Stowell, 1998; McKinley et al., 2003; Newlin & Levenson, 1979; Sherwood et al., 1990). Of the different ECG signal properties, the one most directly relevant to assessing heart rate variability is the RR interval, which researchers sometimes refer to as the NN interval as a means of highlighting that the heartbeats referenced are in fact normal beats (i.e., not the result of abnormal rhythms). The letter “R” is used to identify the peak electrical activity detected within QRS complex – a stereotyped deflection present in the ECG signal (see **Figure 2.1**). Most heart rate and heart rate variability analyses using ECG data utilize the RR interval as a measure of the time lapse between successive heart beats, also known as the interbeat interval. As a result, pioneering empirical and theoretical work based on measures of heart rate variability has largely relied on ECG data. Naturally then, the vast majority of tools originally developed for researchers in this field were created using the ECG waveform (Berntson et al., 1990; Brain-Body Center, 2007; Kaufmann et al., 2011).

One downside of the use of ECG for recording heart rate data is that it can be somewhat more invasive than alternatives. It is not that the electrodes used for these recordings pose a risk or are likely to harm participants in any way; it is simply that the application of adhesive electrodes and isotropic gels can be off-putting for participants, particularly when their use often requires sensors to be placed on the chest and abdomen (e.g., Biopac Systems, 2017). To normalize and ease the experience of collecting these

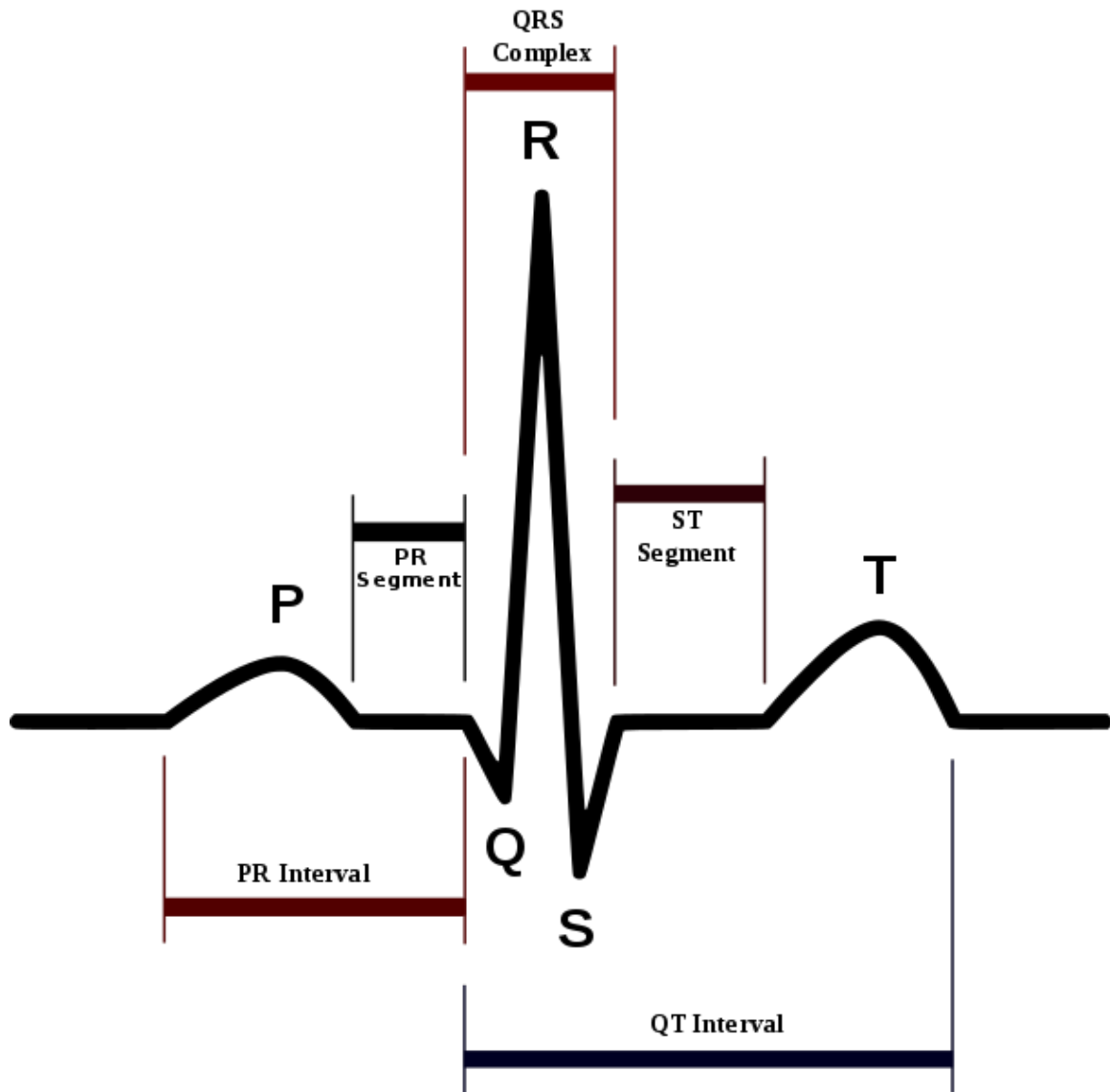


Figure 2.1 Typical Heartbeat as detected by electrocardiogram recordings

data, some device makers have created vests or chest bands that are designed to be easier to apply and feel more natural to wear. These sorts of recording devices combined with wireless technology allow investigators to collect heart rate data from participants in ambulatory states. Though beneficial, heart rate chest bands and vests still require the removal of certain articles of clothing and/or the placement of sensors in sensitive areas. When the goal of research falls in the domain of exercise physiology, the ultimate

psychological discomfort associated with using particular measurement tools may be of limited concern. For researchers interested in assessing the emotional and psychological correlates of cardiac activity, utilizing sensors that minimize residual sources of discomfort is more central to safeguarding the internal validity of their work. Some populations may even find the use of electrodes so off-putting they refuse to participate¹.

Photoplethysmography

Ease of use, reduced invasiveness, and relatively low cost are among the reasons for the increased use of *photoplethysmography* (PPG) in recent years. The ubiquity of PPG sensors has even grown to the point that they can be found in many smart watches and other portable consumer technology. As compared with the ECG waveform, PPG sensors produce a longer, slower-moving waveform that measures changes in blood flow (see **Figure 2.2**). Whereas ECG recordings are based on the electrical activity that causes the heart's atria and ventricles to contract, PPG sensors record the changes in blood flow associated with the pulsatile wave that is the direct result of said contractions. Pairing a light emitting diode with a photoreceptor, PPG sensors monitor changes in light absorption that accompany cyclical increases and decreases in the amount of red blood cells moving in rhythm with the heart's contractions. Common locations for these sensors include the fingertip and ear lobe, both of which are sites with well-distributed capillary beds. Additionally, fingertips and earlobes are generally exposed and easy to access, reducing the potential discomfort of having to access regions typically covered by clothing, as can be the case when recording ECG data.

¹ The data used in the present two studies were drawn from a sample of extremely inhibited and anxious preschoolers. Anecdotally, we had several children who found the application of a Velcro strap around their finger unbearable and refused to wear the device.

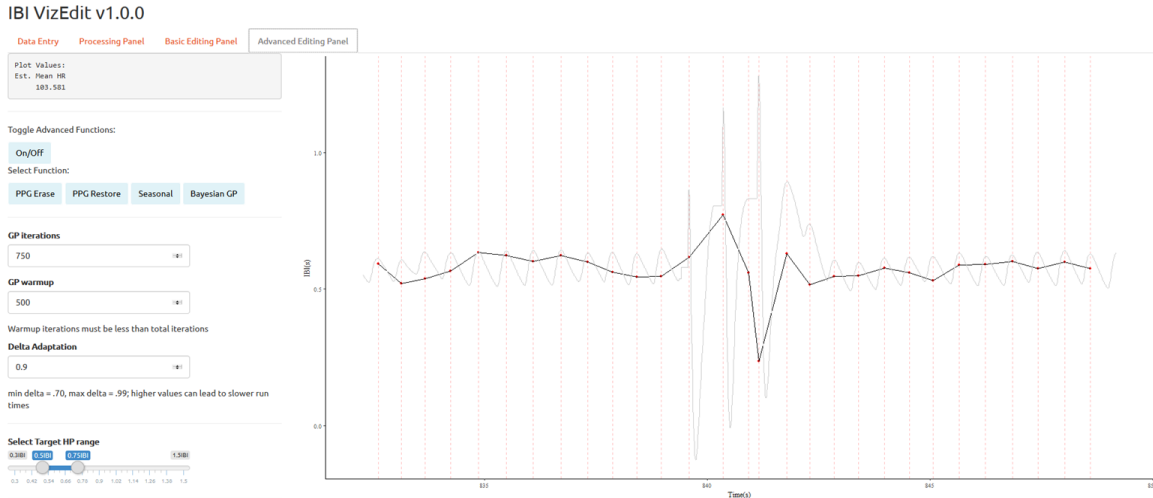


Figure 2.2 Screenshot of photoplethysmogram signal with artifacts in IBI VizEdit

Simultaneously captured PPG and ECG signals reveal how the different physiological and physical sources generating each signal manifest (see **Figure 2.3**). Both signals contain obvious peaks that can be used as regular markers of heart beats. PPG signals, however, lack the sharp signal deflections that characterize ECG recordings. As a result, the amount of time between local maxima and local minima associated with each peak in the PPG signal is much longer than the corresponding time associated with local minima and maxima at the site of peak electrical activity in the ECG signal. The very practical problem this presents is that the amount of time within which occurrence of a beat could be identified is much larger for PPG than ECG data. This property means that any algorithm designed to detect heart beats using a PPG signal has a wider target to hit when marking the timing of each beat. The wider interval also means that identification of heart beat timing is more easily influenced by natural and artificial sources of interference (i.e., arrhythmias or motion artifacts). In spite of these potential downsides, the peak-to-peak (PP) intervals that can be detected using PPG sensors show good

correspondence with the RR intervals used to identify peaks in the ECG waveform (Bolanos, Nazeran, & Haltiwanger, 2006; Giardino, Lehrer, & Edelberg, 2002; Selvaraj, Jaryal, Santhosh, Deepak, & Anand, 2008).

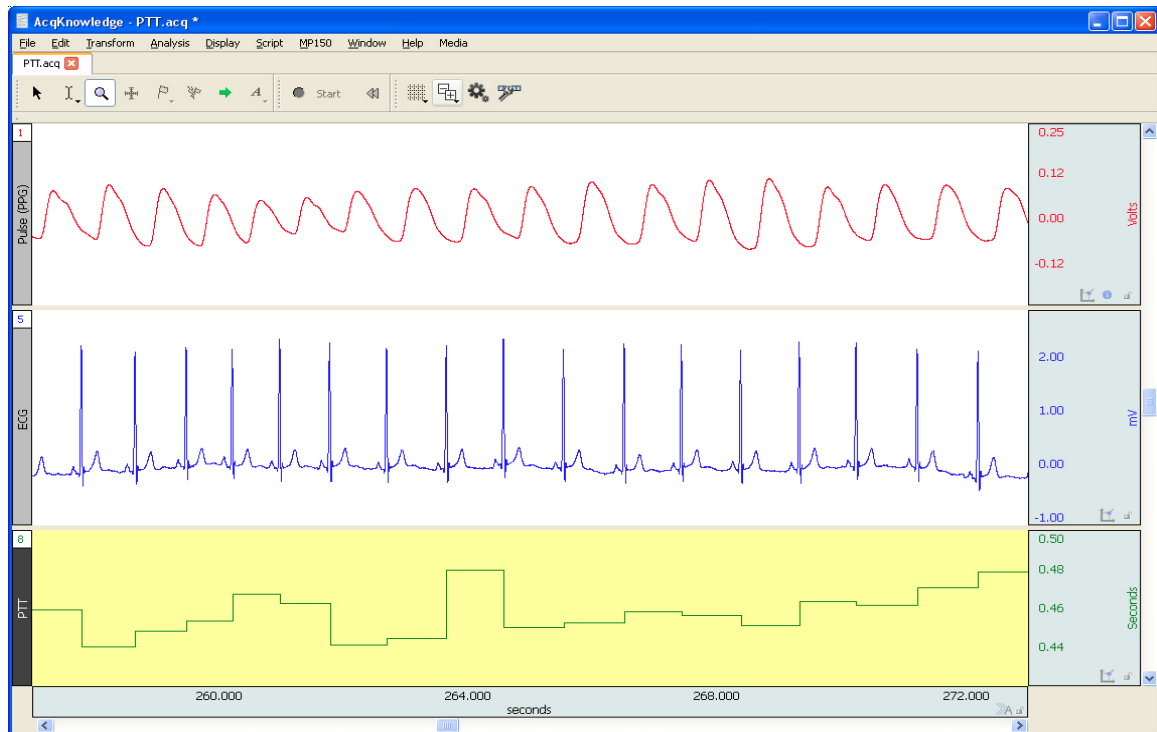


Figure 2.3 AcqKnowledge screenshot of simultaneously collected electrocardiogram (middle) and photoplethysmogram (top) data

The ability to integrate findings collected using both PPG and ECG technologies is certainly a benefit to the field. Yet, some challenges still remain as researchers utilize a wide variety of hardware and techniques to extract measures of heart rate and heart rate variability when using heart rate signals (Quintana et al., 2016; Schäfer & Vagedes, 2013). This heterogeneity in approaches is also reflected in the plethora of existing techniques and tools and may be one of several reasons that, until recently, the development of clear reporting standards in this field have lagged (Laborde et al., 2017; Quintana et al., 2016).

Understanding Common Measures of Cardiac Activity

Heart Rate/Heart Period

The simplest measure of cardiac activity is the rate at which a heart beats on average. Depending on the particular field of study, researchers may prefer to calculate either heart rate (typically beats per minute) or heart period (time between successive heart beats). Note that one measure is the inverse of the other. There are multiple physiological factors that influence heart rate (Saul, 1990). As a result, while heart rate and heart period can be used to measure reactivity to a stimulus (Hamilton & Alloy, 2016) or overall health (Acharya, Joseph, Kannathal, Lim, & Suri, 2006), changes in heart rate are difficult to tie back to a specific physiological process when no other information about the individual is available.

Standard Deviation of the Normal-Normal Interval

The standard deviation of normal-to-normal intervals (SDNN) is a measure of total heart rate variability following the identification and removal of ectopic beats (i.e., beats that deviate from the normal heart rhythm) and/or other artifacts. As a measure of total beat-to-beat variability in heart period, SDNN has several physiological sources. Parasympathetic inputs affect variability within the respiration cycle (see below), but interbeat intervals are also modified over longer periods by other physiological processes (Saul, 1990). When based on heart rate recordings that last at least as long as 24 hours, SDNN is considered to be the ‘gold standard’ used for stratifying cardiac risk, with values below 50 ms indicating elevated risk (Huikuri & Stein, 2013; Kleiger, Miller, Bigger, & Moss, 1987; Malik et al., 1996). Calculating SDNN is straightforward once a vector of interbeat intervals has been identified:

$$SDNN = \sqrt{\frac{\sum_{i=1}^N (x_i - \bar{x})^2}{N-1}} \quad (\text{Equation 1})$$

Where x is a vector of normal interbeat intervals of length N . SDNN over long observation windows is strongly correlated with ultra-low frequency heart rate variability (0 Hz to 0.03Hz), low frequency heart rate variability (0.03 Hz to 0.15 Hz), and high frequency heart rate variability (0.15 Hz to 0.40 Hz). Over brief data collection intervals, variation in heart period as a function of respiration (i.e., parasympathetically mediated) tends to be the predominant source of SDNN. Contributions to normal heart rate variability from variation in lower frequency bands tends to increase with the window of observation (Shaffer & Ginsberg, 2017).

Root Mean Square of Successive Differences

The root mean square of successive differences (RMSDD) is a measure of variability in sequential beat-to-beat differences in heart period. In contrast to SDNN and other related measures, which measure variability from a grand mean, RMSSD measures variability in successive peak-to-peak differences in heart period. As such, RMSSD provides a much more focused measure of the variability in short-term heart rate changes. Given the brief period over which change occurs between beats, RMSSD is thought to largely tap vagally-mediated changes in heart period (Shaffer & Ginsberg, 2017). Despite the strong correlation between RMSSD and “purer” measures of vagally-mediated heart rate variability (Kleiger, Stein, & Bigger, 2005), the specific properties of vagal influence on RMSSD values remain poorly understood (Penttila et al., 2001; Schipke, Pelzer, & Arnold, 1999). The formula for calculating RMSSD represents a straightforward time-series measure of signal variability:

$$RMSSD = \sqrt{\frac{\sum_{t=1}^{N-1} (x_{t+1} - x_t)^2}{N-1}} \quad (\text{Equation 2})$$

Where N is the number of total normal-to-normal successive interbeat intervals, x . The nature of this equation highlights an additional difference between SDNN and RMSSD, as the latter explicitly leverages the time-series nature of the data and the former largely ignores this property.

Respiratory Sinus Arrhythmia & High-Frequency Heart Rate Variability

Respiratory sinus arrhythmia (RSA) and high-frequency heart rate variability (HF-HRV) are each calculated in slightly different manners, despite the fact that both measures are closely tied to vagally-mediated heart rate variability (Saul, 1990; Shaffer & Ginsberg, 2017). RSA is variation in the timing between successive heart beats as a function of the respiration cycle. During inhalation, heart rate increases with the influx of fresh oxygen from the lungs. Exhalation is associated with a drop in heart rate by comparison. The specific quantification of RSA is generally achieved using Porges' polynomial moving window algorithm (Porges, 1985). Importantly, Porges' moving polynomial method requires a file of edited interbeat intervals as its only input. Other methods for obtaining measures of RSA exist as well. For instance, certain recording configurations allow for the simultaneous collection of heart and respiration rate via different sensors. Proprietary software is then capable of taking these two different signals and using them to calculate RSA values (e.g., AcqKnowledge; Biopac Systems, 2016). Both approaches have been shown to be highly correlated in the past (Denver et al., 2007). Unfortunately, both approaches are also "black boxes" in the sense that the specific formulae and programming rules that underlie the Porges method of calculating

RSA and the values generated by proprietary software are not always easily accessed by researchers (i.e., these resources and tools are often not open-source).

As an alternative to Porges' adaptive moving polynomial algorithm or the simultaneous collection of both respiration and heart rate, researchers can perform spectral analyses of the data. Spectral analyses are used across a number of fields, typically ones in which the periodic properties of a signal are of interest. **Figure 2.4** depicts the spectral density of an interbeat interval file derived from five minutes of continuously collected PPG data. By focusing on either the relative spectral power or spectral power density within a specific frequency band (e.g., 0.15-0.40 Hz in adults, 0.24-1.04 Hz in children), it is possible to obtain an accurate measure of variability in the beat-to-beat intervals within the respiration cycle. High-frequency heart rate variability calculated in this manner has been shown to reliably correlate with other estimates of parasympathetically-mediated heart rate variability (Shaffer & Ginsberg, 2017). Due to utilization across a wide range of scientific fields, numerous open-source and proprietary programs have been developed to aid in power spectral analysis (Barbour & Parker, 2014; Seilmayer, 2016; Stoffer, 2017). Each of these programs requires slightly different input, produces slightly different output, and relies on slightly different machinery, differences that can make it difficult to compare and integrate results across studies.

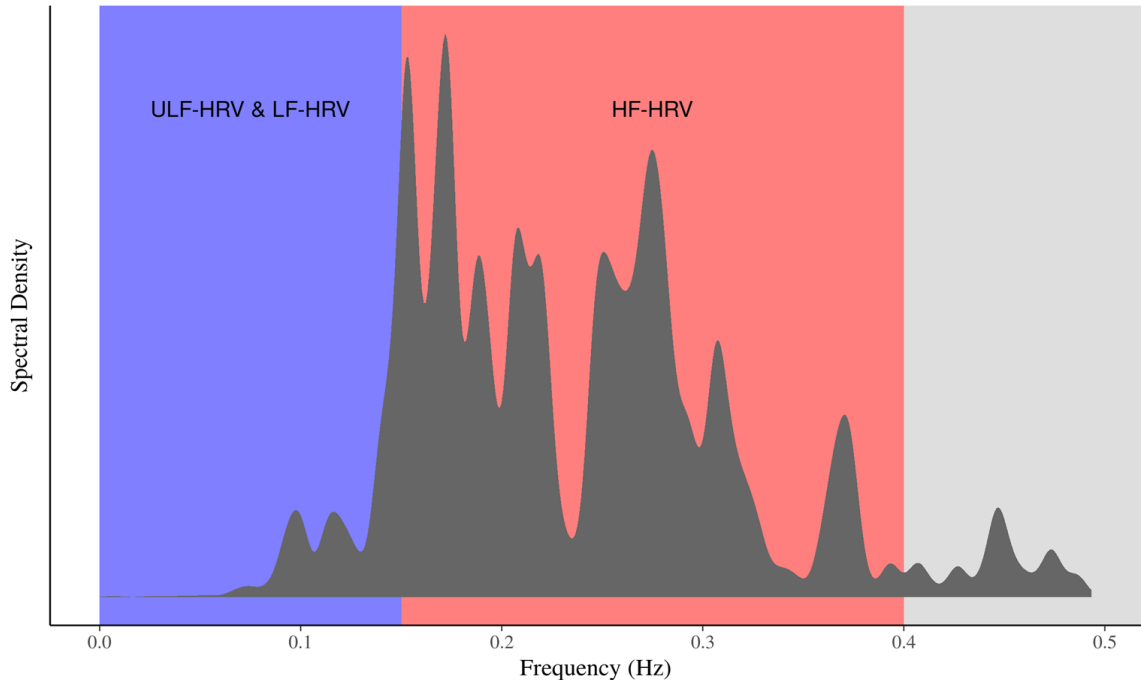


Figure 2.4 Spectral analysis of photoplethysmogram-derived interbeat intervals

Note. Data were collected at from a single seated adult over a 5-minute interval while participating in a series of simple motor activities. ULF = Ultra-low frequency; LF = Low frequency; HF = High frequency; HRV = Heart rate variability.

Tonic vs. Dynamic Measures of Variability

Any one of the heart rate variability measures described above can be used to varying degrees as a static indicator of autonomic regulation of heart rate, or they can be calculated over time to allow for dynamic modeling of autonomic regulation in response to varying conditions. Tonic heart-rate variability, especially vagally-controlled heart rate variability (e.g., RMSSD, RSA, and high frequency heart rate variability), is often used as a biomarker of emotion regulation capacity (Balzarotti, Biassoni, Colombo, & Ciceri, 2017; Geisler, Vennewald, Kubiak, & Weber, 2010; Thayer & Lane, 2000). More specifically, higher levels of parasympathetically-mediated heart rate variability at rest are thought to enable greater flexibility in responding to challenges (Porges, 2007; Thayer & Lane, 2000). Recent work, however, has begun to suggest that there may be a

curvilinear relation between tonic measures of heart rate variability and an individual's regulatory capacity (Kogan, Gruber, Shallcross, Ford, & Mauss, 2013). These findings suggest that there may be something of a Goldilocks zone for vagally-mediated heart rate variability at rest, which is to say that especially high or especially low levels of tonic heart rate variability may each prevent the initiation of socially and emotionally adaptive responses, albeit in different ways (Kogan et al., 2014, 2013; Miller, Kahle, & Hastings, 2017).

For many research questions, tonic levels of heart rate variability provide little insight. Instead, how vagally-mediated heart rate variability *changes in response* to certain tasks may be more theoretically relevant. There are two general approaches to modeling dynamic changes in heart rate variability. The first involves calculating a summary score for an entire task and then determining the difference between that average task score and some baseline or reference score, typically derived from another task or condition. Depending on the specific measure used, recommendations for task duration vary, with measures of RSA and high frequency heart rate variability typically requiring a longer window to obtain stable estimates than RMSSD (Shaffer & Ginsberg, 2017). The second common approach is to break the interbeat interval file into smaller, equally sized epochs, calculate a measure of heart rate variability for each epoch and then analyze the data using non-linear growth models (Miller et al., 2013). As with task duration, the epoch length would generally need to be longer for measures of RSA and HF-HRV than for RMSSD.

Broken into epochs or tasks, measures of heart rate variability can either remain the same, increase, or decrease. When measures of vagally-mediated heart rate variability

increase over time it is referred to as *augmentation*. *Vagal suppression*, sometimes referred to as *vagal withdrawal*, occurs when measures of vagally-mediated heart rate variability decrease. Augmentation is thought to underlie attention and complex cognitive activity. By comparison, vagal withdrawal in response to challenging and/or stressful tasks is thought to underlie an individual's ability to effectively recruit metabolic resources and respond adaptively to challenging environmental demands (Calkins, 1997; El-Sheikh & Buckhalt, 2005; Miller et al., 2013). Highlighting the importance of good design, on their own, augmentation and withdrawal are neither adaptive nor maladaptive. Dynamic changes in heart rate variability are only interpretable when specific tasks/conditions, the study population, and the measure of heart rate variability are taken into account (Laborde et al., 2017). It is, therefore, important that any researcher interested in heart rate variability as a study variable incorporates changing conditions that vary in theoretically meaningful ways.

Current Editing and Analytic Strategies

The effective processing and editing of heart rate data remains an active area of research (Couceiro et al., 2012; Elgendi, 2012; Pflugradt, Geissdoerfer, Goernig, & Orglmeister, 2017; Schäfer & Vagedes, 2013; Shaffer & Ginsberg, 2017). What is clear is that motion artifacts and naturally occurring trends in the heart rate signal must be edited to appropriately measure heart rate variability, particularly when variation within specific frequency bands (e.g., RSA, HF-HRV) is the target measure (Akar, Kara, Latifoğlu, & Bilgiç, 2013). For methodologies that rely on the analysis of interbeat intervals, the common starting point is turning the raw heart rate waveform into a separate file of interbeat intervals. Most programs achieve this goal algorithmically with a set of

mathematical rules used to identify local maxima (peaks) and/or minima (troughs) in the waveform (Akar et al., 2013; Lu et al., 2008; Sukor et al., 2011). The distance in time between the peaks can then be used to calculate interbeat intervals. As measures of successive changes in heart period, the file of interbeat intervals then becomes the main focus of editing and analysis.

After creating a vector of interbeat intervals, editing approaches diverge. One of the earliest editing techniques involved the manual identification of aberrant interbeat interval values (MxEdit: Porges, 1988). *MxEdit* is a MSDOS-based program that was later updated and released for Windows with a graphic user interface known as *Cardio Edit* (Brain-Body Center, 2007). *Cardio Edit* and its predecessor *MxEdit* allow researchers to alter the interbeat interval file using one of three general functions. Editors can divide a point into two (or more) separate points, a useful editing strategy if a peak detection algorithm skips over a peak incorrectly. They can combine points, often necessary when an early beat is incorrectly identified in the original file. Finally, these programs also allow the averaging of two or more successive points. While using these functions, editors must always be wary of over-editing or under-editing as each will bias measures of heart rate variability downward or upward, respectively.

Despite being the gold standard for processing and analyzing heart rate data, manual editing can be prohibitive in terms of the need for personnel. Algorithmic and model-based automated editing tools offer an attractive alternative for research teams working with limited staff. Recently, machine learning techniques have been added to the editing toolbox, and comparisons with manually edited versions of the same files have yielded promising results (Hegarty-Craver et al., 2017). Building on this work, it is

possible to envision a potential marriage of methods in which trained editors employ a machine learning algorithm for the initial editing pass and then choose to accept, reject, or modify the algorithm's proposed edits. Such an editing process would leverage the speed and efficiency promised by model-based and machine learning editing techniques while retaining the accuracy of manual editing. The proposed set of studies offers a modest step in this general direction by combining advanced modeling with human oversight of the final edited file.

Identifying and Managing Low Quality Data

To date, very little in the way of standards regarding the editing of heart rate data exist. As pointed out in a pair of recent reviews (Laborde et al., 2017; Quintana et al., 2016), researchers vary widely in their transparent reporting of heart rate editing decisions, and there exist no empirically-based standards for distinguishing usable data from data that have been too corrupted by artifacts to warrant inclusion in analyses. Strategies for dealing with heart rate data that include a large number of artifacts are similarly inconsistent from study to study. A common approach is to exclude certain segments of data from final analyses (Quintana et al., 2016), a questionable decision from a statistical standpoint that likely results in non-random missingness, particularly if participants are exposed to different conditions during data collection. Alternatively, it is possible to manually edit these sections to bring the resulting interbeat interval values within ostensibly reasonable values. However, in some cases the underlying signal provides little information about an individual's heart rate, making editing choices somewhat arbitrary. This is particularly true when artifacts affect the signal over several seconds, disrupting the detection of multiple sequential beats. The precise conditions

under which manually editing these sections results in biased estimates, particularly for PPG-derived data, is unclear and should be of concern to psychophysicists working with these data.

Introducing *IBI VizEdit*

IBI VizEdit (Barstead, 2018) is a recently developed RShiny application designed to exploit *R*'s extensive statistics and graphics libraries in order to improve editing options for researchers working with PPG data. *IBI VizEdit* is intended to add to existing editing tools in several meaningful ways. To start, the program is completely open-sourced relying on *R* (<https://cran.r-project.org/>) and *Stan* (<http://mc-stan.org/>). The beta version of the program can be found on GitHub (github.com/matgbar/IBI_VizEdit), with plans for the development of a complete *R* package that is centered around the program's editing interface. *IBI VizEdit* promises no black boxes, and its core code is available to any interested researcher who wants to understand the specific choices being made at each stage of data processing. In short, the development of *IBI VizEdit* is adhering as closely as possible to principles of the Open Science Framework (<http://openscienceframework.org>).

The program includes practical benefits in addition to its attempts at scientific integrity. For one, it was created, first and foremost, for editing PPG signals, from start to finish. From raw file to cleaned, processed, and summarized data, *IBI VizEdit* prevents researchers from having to move between multiple specialized programs each capable of performing only one step of the data preparation process. In an important boon to improving editing practice transparency, the program also provides a detailed summary of all the edited points retained in the final file. As a result, the number of edits, their type,

and their relative location in the interbeat interval series are all output in a simple summary file once editing is completed. The reporting of these simple but comprehensive editing statistics will hopefully go a long way toward motivating increased transparency in summarizing the amount and type of editing required.

Finally, *IBI VizEdit* includes a novel imputation technique, designed to be an alternative to manual editing that often leads to artificial reductions in variability. The driving engine of this imputation feature is a Gaussian Process model, several parameters of which need to be set by the user. The incorporation of a Gaussian Process imputation model has the potential to retain naturally occurring heart rate variability in sections of raw data where the true heart rate signal is completely unidentifiable, provided sufficiently ‘good’ data exist on either side of the section targeted for imputation. Investigating the specific parameterization of the imputation model as well as its accuracy represent major components of the present report.

Summary and Current Research

Heart rate variability is an evolutionarily-rooted biomarker that taps the functional connectivity of the central autonomic network, a collection of neural circuits, physiological feedback loops, and chemical signaling pathways responsible for altering internal states to match external environmental demands (Thayer & Lane, 2000). The degree to which this network is functionally connected generally corresponds to the capacity for generating adaptive social and emotional responses. Variation in functional connectivity of the central autonomic network, as measured by vagally-mediated heart rate variability, may therefore carry meaningful diagnostic information (Beauchaine & Thayer, 2015). Researchers or clinicians interested in measuring heart rate variability can

do so using ECG, a more established, more precise, if slightly more invasive technique compared with PPG, which is gaining in popularity in certain research contexts². In addition to recording options, researchers have their choice of heart rate variability measures, some of which correspond to different underlying physiological processes (Saul, 1990; Shaffer & Ginsberg, 2017). To achieve accurate summary values for these different variability metrics, raw signals need to be processed and edited. The techniques for processing and editing heart rate files have largely been developed for use with ECG signals. While there are a number of approaches that have been used to process PPG data, no discipline-wide standards have taken hold, to the field's detriment.

IBI VizEdit was created to streamline and standardize PPG data processing, editing, and analysis. In addition to basic manual editing strategies included in other programs (e.g., *Cardio Edit*; Brain-Body Center, 2007), *IBI VizEdit* pushes the editing strategy envelope somewhat. Specifically, the development of a model to guide imputation is a key novel addition to *IBI VizEdit* and could represent a natural evolution in the mixing of automated editing with manual review. Finally, to support standardization of reporting in this area of research in general, accessible summarized information is automatically generated based on edited data points contained in the final file. Following an editing session, the program outputs all edited and raw files, separated by task if timing information is available for different conditions. This feature ensures that a complete and comprehensive record of editing decisions is saved each time the program is used.

² Other collection methods exist, but PPG and ECG recordings make up the vast majority of published heart rate data (Quintana, Alvares, & Heathers, 2016).

The present set of studies represents an initial evaluation of the program. Study 1 examined the degree to which the different parameterizations of several proposed imputations models effectively recovered true heart rate variability values. Study 2 provided a more direct assessment of the full deployment of *IBI VizEdit*, and in so doing offered a preliminary probe into the ability of editors to reliably use the program as well as the validity of the values generated from files edited within this newly created software.

CHAPTER 3

The only relevant test of the validity of a hypothesis is comparison of prediction with experience.

- Milton Friedman

An Overview of *IBI VizEdit*

Peak Detection Algorithm

Before any edits can be made, the first step in processing raw heart rate data is to identify the relative location of heart beats in time. To accomplish this task, the peak detection algorithm incorporated in *IBI VizEdit* involves several programmatic steps. First, the raw file is smoothed using a low-pass median filter to reduce signal spiking (Tukey, 1977), followed by a cubic smoothing spline (Shumway & Stoffer, 2017) (see **Figure 3.1**)³. Then, the smoothed file is de-trended (Akar et al., 2013). Following these preliminary processing steps, a differenced time series is derived by subtracting successive data points from one another:

$$x_D = x_{t+1} - x_t \quad (\text{Equation 3})$$

The resulting vector of differenced values, x_D , has several useful properties. The first is that, in a well-defined periodic or quasi-periodic signal, difference values will switch from positive to negative at each peak (i.e., where a first-order derivative of the signal would be equal to 0). As the wave approaches a peak, each successive value is higher than the previous, leading to positive difference scores. Following the peak, the difference scores turn negative as each successive value is lower than the previous value in the original heart rate signal (see **Figure 3.2**). Using this property, the point at which

³ Code for all figures created for the dissertation can be found in **Appendix C**

positive values transition to negative values marks a local peak. The code in **Appendix A.1** leverages these features to identify maximum values within a user-defined bandwidth. The code in **Appendix A.2** converts the output of the peak detection algorithm into a vector of inter-peak intervals.

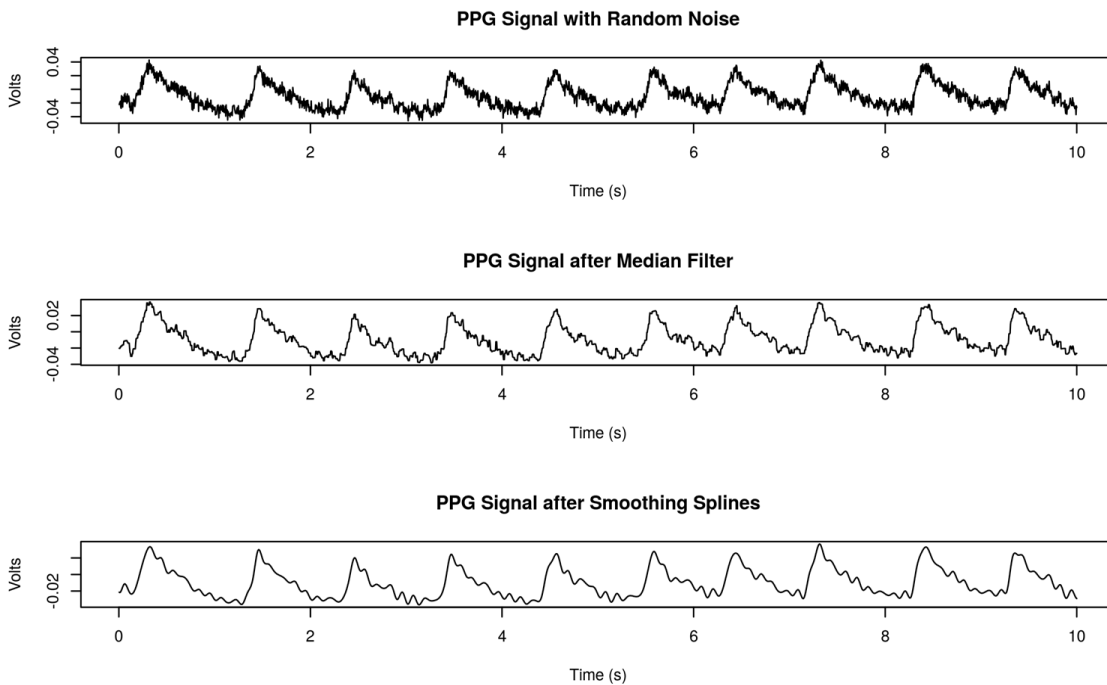


Figure 3.1 Effects of pre-processing photoplethysmogram data with random noise

Note. These data were not taken from any of the cases used in the present data set. Instead, they are an example of data collected in an fMRI recording session, which is one of the factors causing the additional noise that was recorded in the signal.

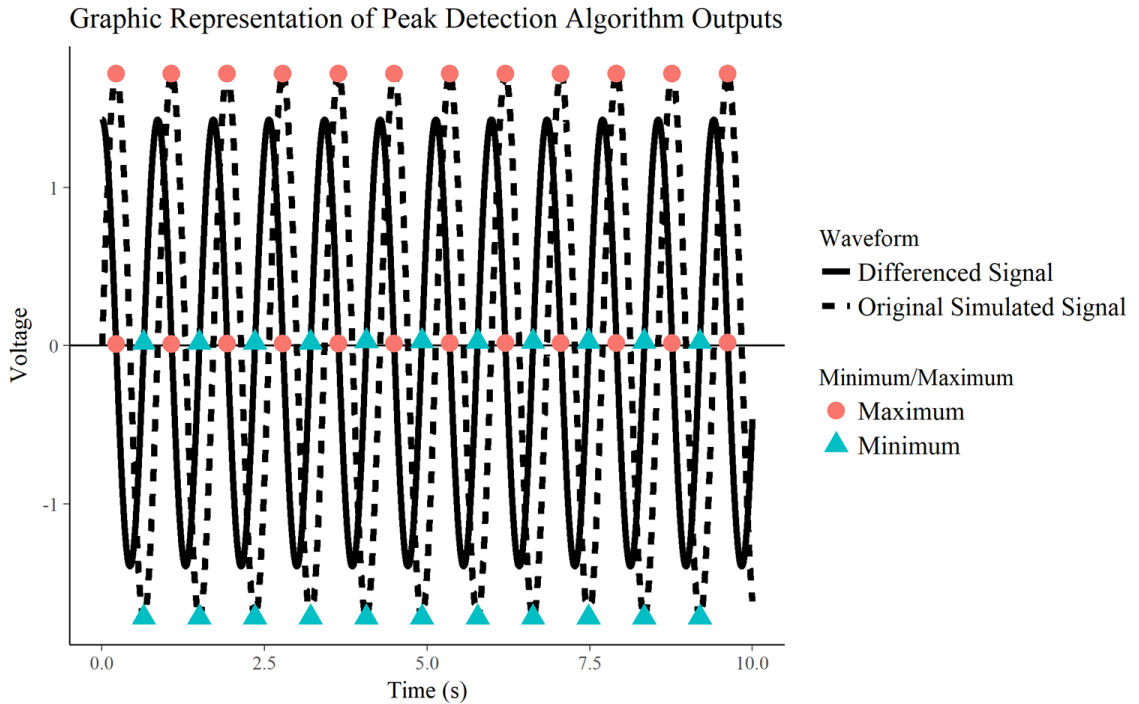


Figure 3.2 Identifying peaks and troughs in periodic data using IBI VizEdit's peak detection algorithm

The next step of program is to use this basic peak detection and inter-peak interval machinery to iterate over several different bandwidths (see **Appendix A.3**). The bandwidth in this set of functions defines how wide an interval to use when identifying peaks. With a slowly oscillating and occasionally disrupted underlying signal, finding the specific location of a given peak can be difficult without defining a specific bandwidth within which to look. While iterating over different bandwidths, the program optimizes the final vector of inter-peak intervals by minimizing first the RMSSD and then the range of the interbeat intervals created, using each bandwidth attempted, thereby ensuring that extreme values and very large deviations in the timing of successive heartbeats are reduced in the resulting vector of interbeat intervals. The bandwidths over which the program iterates are restricted to values between .15s and 0.75s. These two values correspond to half of the mean interval for a heart beating 200 times per minute and half

the mean interval of a heart beating 40 times per minute, respectively. Both values represent extremes in a typical psychophysiological experimental context. **Figure 3.3** displays the results of this iterative peak detection algorithm applied to a 10-second segment of simulated data.

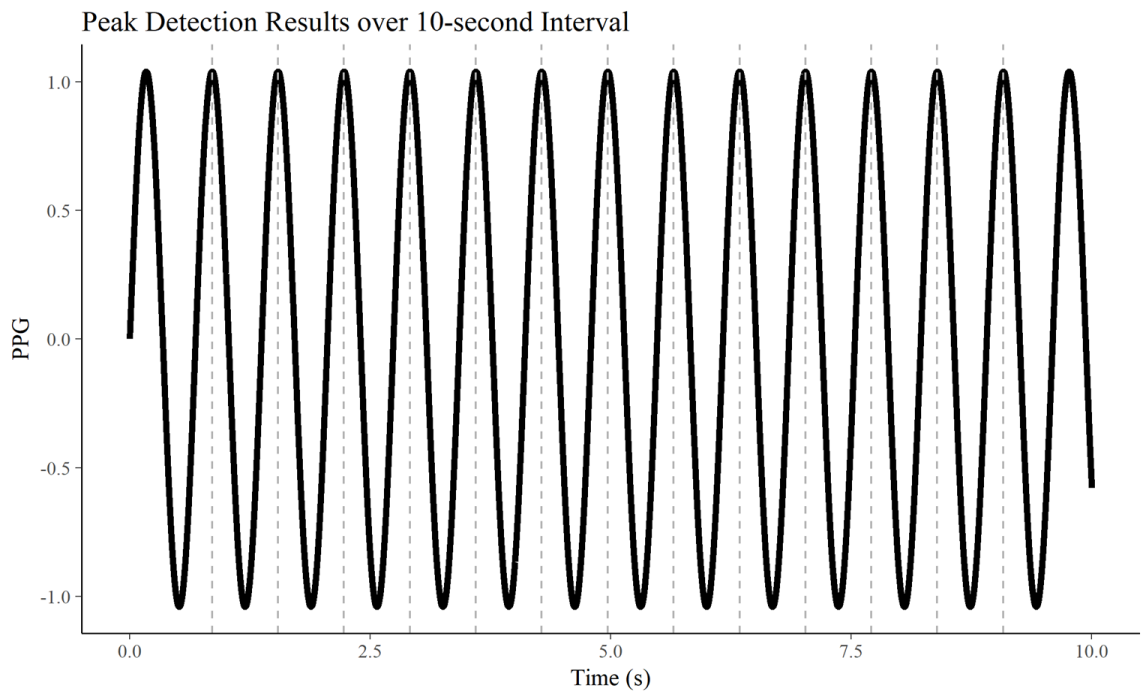


Figure 3.3 Results of iterative peak detection algorithm applied to simulated data

Note. The mean period for this simulated data was .5904, and the mean estimated period for the 10-second segment was .5903, demonstrating an excellent degree of correspondence between obtained and expected estimates.

When tasked with identifying peaks in a highly controlled, periodic signal, it is not surprising that the algorithm correctly identifies the relative location of peaks. A more meaningful test is the algorithm's efficacy when challenged to find interbeat intervals from PPG data. As seen in **Figure 3.4**, when the PPG signal is relatively well-defined and regular, the peak detection algorithm is largely successful at identifying peaks and converting the PPG data into interbeat intervals (the black line with red points). However, the quality of the algorithm's output is largely governed by the quality of the original

PPG data (see **Figure 3.5**), a fact that serves to highlight the importance of manual editing after implementing the peak detection algorithm.

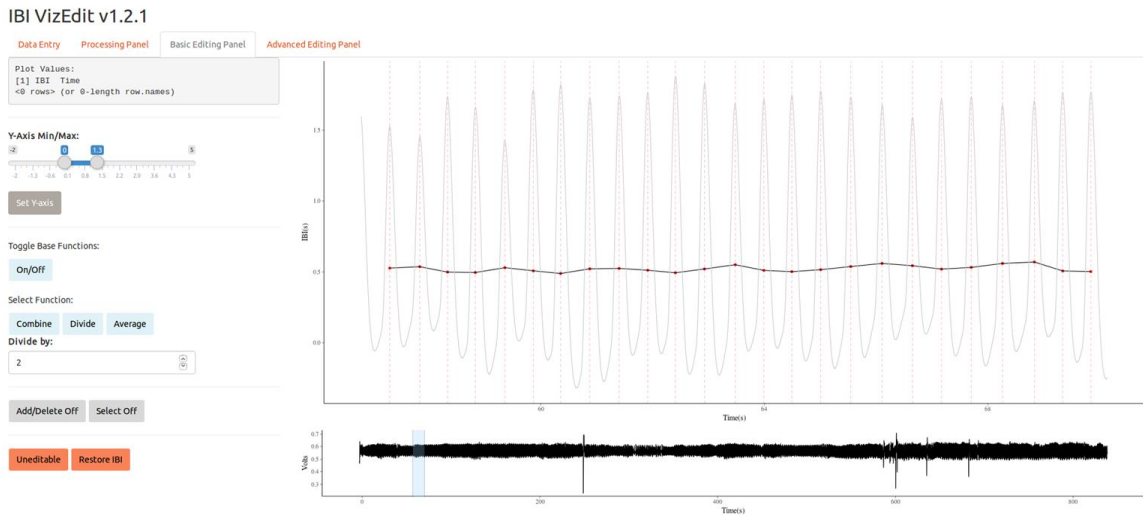


Figure 3.5 Screenshot depicting peak detection algorithm performance with artifact-free data

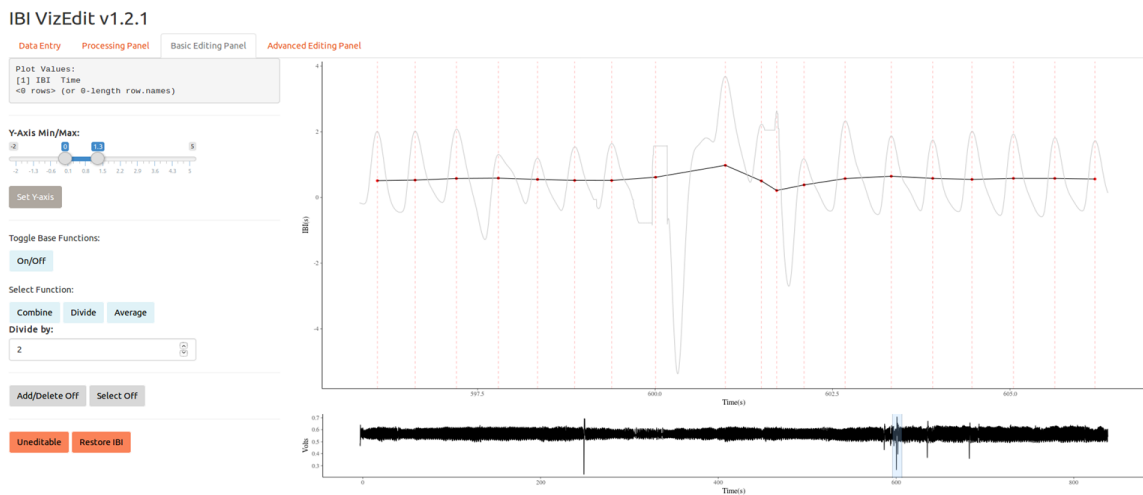


Figure 3.4 Screenshot depicting peak detection algorithm performance with data including artifacts

Manual Editing

After finding the interbeat intervals, *IBI VizEdit* allows users to seamlessly overlay the de-spiked, smoothed, and de-trended PPG signal with the resulting timing of

interbeat intervals identified (see **Figure 3.4**). The editor can then simultaneously inspect interbeat intervals and the underlying signal that led to their detection to better inform editing decisions. The basic editing strategies incorporated in the program are modeled after editing choices available in the interbeat interval editing program *Cardio Edit* (Brain-Body Center, 2007). Editors can choose to combine two or more sequential interbeat intervals when signal noise results in the erroneous identification of an early beat (see **Figure 3.6**). In other instances, a heartbeat may be skipped if the signal is particularly distorted (see **Figure 3.7**). When the peak detection algorithm fails to identify a heartbeat, the editor can divide the resulting inflated interbeat interval value into two or more points, corresponding to the approximate number of beats that should have been identified in the preceding segment. Finally, when subtle signal distortions lead to minor beat-to-beat deviations from each other that are not consistent with the overall pattern (see **Figure 3.8**), the editor can choose to average these beat-to-beat differences. In addition to the ability to combine, divide, and average points, editors can manually delete erroneous points as well as add missed points directly to the file itself, using the processed PPG signal as a guide.

These five operations – *combining, dividing, averaging, deleting* and *adding* points – represent the backbone of manual heart rate editing and have been included in, what is likely the most widely used manual heart rate editing program to date, *Cardio Edit* (Glenn et al., 2018; Guy, Souders, Bradstreet, Delussey, & Herringto, 2014; Kenkel, Suboc, & Carter, 2014; Lamm, Porges, Cacioppo, & Decety, 2008; Lewis, Furman, McCool, & Porges, 2012). However, the integration of these operations in *Cardio Edit* is not always straightforward as editors cannot use the first three functions when using the

latter two and vice versa. Ideally, editors should not have to load and re-load the same file to make use of different sets of manual editing functions. In addition to being a self-contained processing and editing program, the simultaneous incorporation of all five functions at once in *IBI VizEdit* represents a modest practical improvement in this regard. The more substantive extension of existing manual editing techniques is through the incorporation of advanced time series modeling that has the potential to guide researchers' decisions when working on sections of data that are particularly difficult to edit. Combining, dividing, and averaging points are ultimately crude solutions to dealing with problematic sections of heart rate data. A model-based approach has the capacity to improve on these basic tools by suggesting a set of reasonable locations for the timing of heart beats, based on the individual's own cardiac activity.

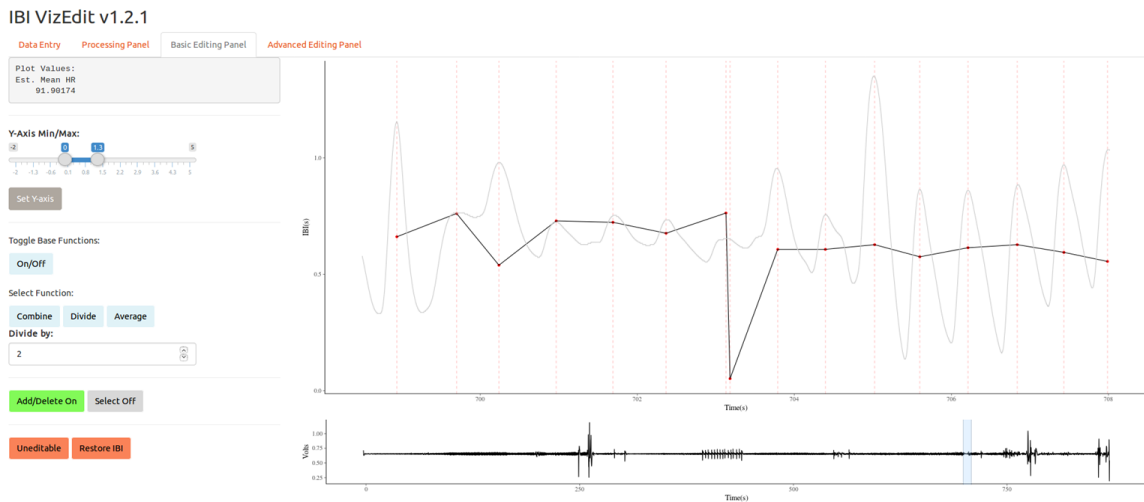


Figure 3.6 Screenshot depicting misidentification of an early heartbeat

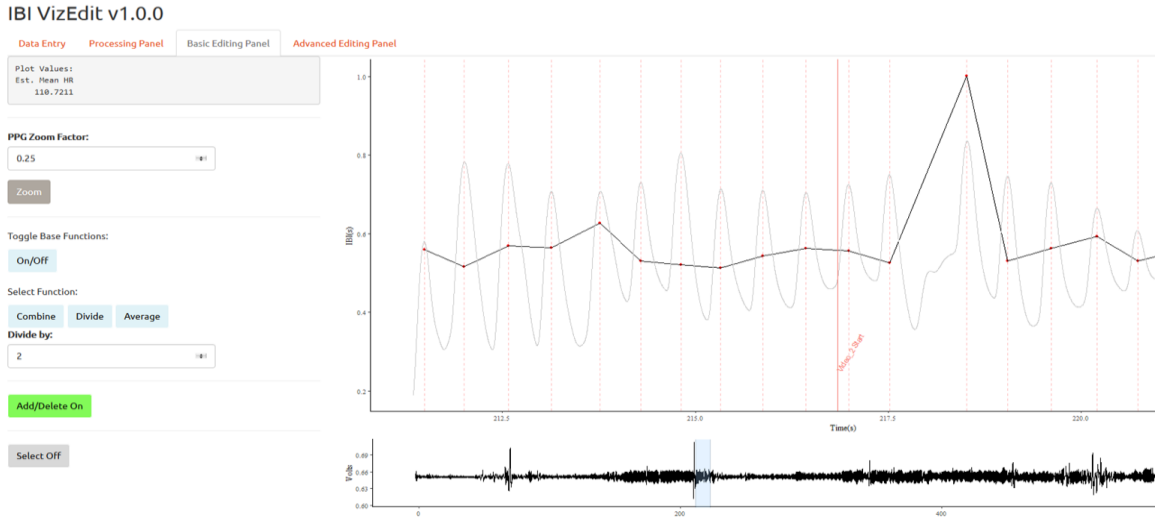


Figure 3.8 Screenshot depicting failure to identify a heartbeat

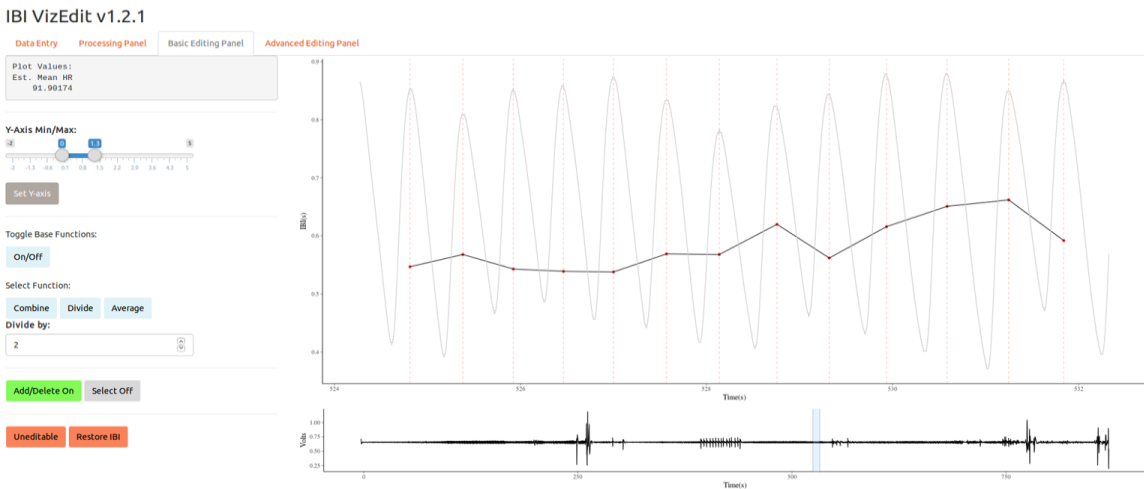


Figure 3.7 Screenshot depicting IBI signal deflection that requires averaging of two points

Gaussian Process Models

Due to the ever-increasing power of standard desktop computers, analytic and modeling options once deemed too computationally demanding to be practically useful can now be implemented in open-source programming languages (e.g., *R*, *Python*). These improvements make it possible for *IBI VizEdit* to incorporate an advanced imputation feature that relies on Gaussian process modeling using *Stan* and *rstan* (<http://mc->

stan.org/). Gaussian process models can be especially useful for modeling time-series data, particularly when there are known sources of signal variation (Gelman et al., 2013; Rasmussen & Williams, 2006).

The application of these models to heart rate data relies on the fact that univariate time series data can be decomposed as the sum of an infinite number of Gaussian processes (Gelman et al., 2013). The imputation models created for the proposed set of studies included up to four different processes thought to explain covariation in PPG signal values across time (see **Appendix A.4-A.6**). This new imputation component of *IBI VizEdit* is the program's least well-tested editing feature, and understanding the potential costs and benefits of utilizing Gaussian process models for data imputation represents a key component of the current report.

Testing of different imputation models unfolded in order of model complexity, beginning with a two-process model that incorporated a single decay function and a single function designed to model the contribution of the individual's average heart rate to the covariance between values in the PPG signal. Building on this relatively simple two-process model, three- and four- process models were also tested in an attempt to account for sources of heart rate variability within different frequency bandwidths (see Equation 4).

$$f(t) = g_1(t) + g_2(t) + g_3(t) + g_4(t) + \epsilon_t \quad (\text{Equation 4})$$

The function $g_1(t)$ (Equation 5) was included in every version of the imputation function. The function is defined by the squared exponential covariance equation k_1 , which models a general decline in the covariance between two points as a function of the amount of time separating the points in the PPG waveform. Another way of thinking

about this equation is that, as the time between two points increases, the expected covariance approaches 0. This approach is similar to autoregressive specifications of covariance matrices without the need to specify, a priori, the order of the autoregressive model. Instead, the specific amount of time affecting the decay rate is modeled as the parameter l_1 , and it is a value that can be estimated on a case-by-case basis. The amount of overall variability in the original series accounted for by this first process was represented by σ_1^2 , another value that varies from participant to participant.

$$g_1(t) \sim N(0, k_1) \quad (\text{Equation 5})$$

$$k_1 = \sigma_1^2 \exp\left(-\frac{(t - t')^2}{2l_1^2}\right)$$

The second Gaussian process (see Equation 6) was specified as a means of addressing the quasi-periodic nature of heart rate data (see Gelman et al., 2013 for a related example, p. 505-506, Equations 3 and 4). This second function allowed the model to incorporate the changing periodic nature of an individual's heart rate, f_{HR} , when predicting missing values, a problem often termed non-stationarity. Non-stationary time-series such as heart rate data are particularly difficult to model mathematically, and typical approaches often involve imposing or assuming some form of stationarity prior to conducting further analyses (Shumway & Stoffer, 2017). The covariance function k_2 offered a means of addressing this well-established statistical problem by adding a squared exponential covariance function to the time-frequency covariance function that constitutes the first half of k_2 . The upshot of this parameterization is that, as the distance in time between two points increases, the model-predicted covariance between the points, which, in this this equation, is driven mainly by the model-estimated heart rate parameter, decreases.

$$g_2(t) \sim N(0, k_2) \quad (\text{Equation 6})$$

$$k_2 = \sigma_2^2 \exp\left(-\frac{2\sin^2(\pi(t-t')f_{HR})}{l_2^2}\right) \exp\left(-\frac{(t-t')^2}{2l_3^2}\right)$$

In addition to these two, relatively simple covariance functions, two more complex processes were also tested to determine if their inclusion yielded any improvement in imputation performance. The two functions, g_3 and g_4 were intended as a means of incorporating sources of high frequency (i.e., respiration) and low frequency (i.e., background metabolic processes) heart rate variability into the model. The hope was that the inclusion of these known sources of heart rate variation in the imputation model would improve the prediction of missing values. Function g_3 was based on a covariance function that incorporated variation in heart rate as a function of respiration, f_R , and took the following mathematical form:

$$g_3(t) \sim N(0, k_3) \quad (\text{Equation 7})$$

$$k_3 = \sigma_3^2 \exp\left(-\frac{2\sin^2(\pi(t-t')f_{HR})}{l_4^2}\right) \exp\left(-\frac{2\sin^2(\pi(t-t')f_R)}{l_5^2}\right)$$

Function g_4 was designed to incorporate sources of low-frequency heart rate variability. To reduce the burden of estimating another parameter, the covariance function fixed the low-frequency value at .1 (the equivalent of an event occurring once every 10 seconds), a common threshold used to separate low and high-frequency sources of heart rate variability (Shaffer & Ginsberg, 2017). Thus, the Gaussian process g_4 performed much the same function as g_3 , only for slower occurring changes to heart rate patterns.

$$g_4(t) \sim N(0, k_4) \quad (\text{Equation 8})$$

$$k_4 = \sigma_4^2 \exp\left(-\frac{2\sin^2(\pi(t-t')f_{HR})}{l_6^2}\right) \exp\left(-\frac{2\sin^2(\pi(t-t').1)}{l_7^2}\right)$$

The code for the three Gaussian process imputation models is available in **Appendices A.4-A.6**.

Implementing Gaussian Process Models in *IBI VizEdit*

One way of thinking about the different equations outlined in the preceding section is that they represent a series of processes thought to account for the covariance between any two values in the original heart rate signal, as a function of time. Once appropriate parameter estimates for each function are found, the model can be used to generate data for any segment in the time series. The main software used for the Bayesian estimation of these models was *Stan*, which has some speed advantages over other Bayesian analysis software (e.g., JAGS/BUGS) as it is written in compiled C++. Additionally, the No-U-Turns-Sampler (NUTS), a modification of the Hamiltonian Monte Carlo sampling algorithm is computationally efficient and does not require the distributional conjugacy required of other Bayesian samplers (e.g., Gibbs) to generate posterior distributions for model parameters (Stan Development Team, 2017).

These computational advantages notwithstanding, there were still practical advantages to consider when implementing these models. Most importantly, establishing defensible approaches to shrinking the parameter space each model had to explore is an important means of reducing computational burden and improving model convergence. Mechanistically within the estimation of the parameters this occurs because a smaller parameter space improves the likelihood that the sampler, on two or more independent chains, converges on similar posterior distributions for each parameter being estimated, and that convergence occurs with fewer warm-up iterations (Gelman et al., 2013). Setting appropriate, strict priors for the length (l) and variance (σ^2) terms that titrate the

contributions of each process to the combined model could certainly restrict the range of values for the sampler to iterate over. However, these values will necessarily vary from case to case and even within different segments of the same file, making a single set of strong priors difficult to specify. An alternative to setting strict priors would be to set relatively uniformed priors, forcing the model to infer the posterior distributions from the data. Although, choosing to set diffuse priors for these hyperparameter distributions to address the problem has notable practical downsides, as doing so leads to increased model run time while the sampler explores the much larger parameter space.

As a middle-ground solution, the specific implementation of Gaussian process models in *IBI VizEdit* relies on a two-step process. First, estimates are determined for the length, variance, and respiration (when included) parameters incorporated in each function via maximum likelihood estimation. Specifically, this initial step is executed using a quasi-Newton optimization algorithm (Stan Development Team, 2017) and reduces the parameter space by eliminating the need to specify priors for each hyperparameter. In the second step of this process, which relies on Bayesian estimation, hyperparameters for each process are then fixed to their estimates derived in Step 1, and a posterior distribution is estimated for the imputed values as well as the heart rate frequency parameter. In practice, when using *IBI VizEdit*, users can provide prior information about heart rate using the interbeat intervals surrounding the section of data targeted for imputation. In the current report, the prior distribution for heart rate frequency was informed by the mean and standard deviation of the interbeat intervals contained in segments of artifact-free data, values easily determined from each case's file.

Research Overview

The data used in the current research were obtained from an ongoing NIMH-funded intervention study (MH103253-04; PIs A. Chronis-Tuscano & K. H. Rubin). The primary aim of this intervention study was to compare and contrast two programs for extremely anxious and inhibited preschoolers. A total of 151 children (aged 3 years, 9 months to 5 years, 4 months), and their parents, were randomly assigned to one of two treatment groups. The first treatment condition was *Cool Little Kids*, a well-established psychoeducation program for parents rooted in cognitive-behavioral therapeutic practices that has been shown to reduce anxiety among extremely inhibited young children (Bayer et al., 2011; Mifsud & Rapee, 2005; Rapee & Jacobs, 2002; Rapee, Kennedy, Ingram, Edwards, & Sweeney, 2005). *Cool Little Kids* requires approximately 12 hours when delivered in small groups and is typically broken into manageable chunks of time (e.g., six weekly two-hour sessions). The second treatment condition was the *Turtle Program*, a multicomponent intervention for extremely shy and inhibited preschoolers that incorporates parent psychoeducation, parent-child interaction therapy, and social skills training (Barstead et al., 2018; Chronis-Tuscano et al., 2015; Chronis-Tuscano, Danko, Rubin, Coplan, & Novick, 2018). The *Turtle Program* was also designed to take approximately 12 hours to deliver, and the modules are segmented into eight 90-minute sessions.

As part of this intervention study, heart rate data were obtained at three different time points for both parents and children: prior to the start of the intervention; halfway through the intervention; and following the intervention (see **Figure 3.9**). In collecting these data at multiple time points, the principal investigators hoped to explore

physiological moderators and mechanisms of treatment response. The complete dataset, therefore, contains heart rate data collected from both adults and children at three different time points as they participated in one of two different interventions. To minimize overlap between the samples used in Study 1 and Study 2, data collected from parents and children assigned to *Cool Little Kids* were used in Study 1. Study 2 relied solely on data from children assigned to the *Turtle Program*, specifically their physiological data collected prior to the start of treatment. Study 2 also made use of play

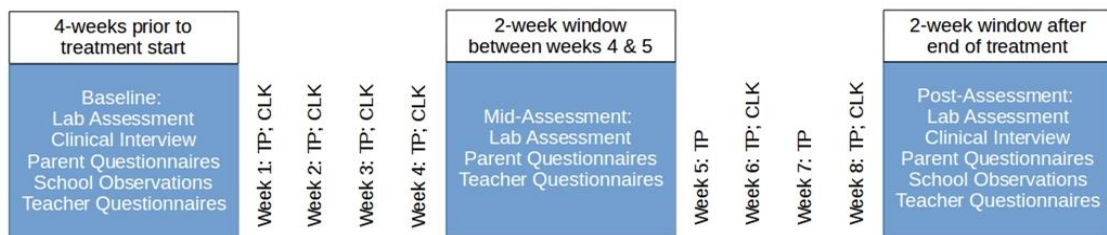


Figure 3.9 Intervention study timeline

observation data collected in children’s preschool classrooms.

Note. School observations were collected 2 weeks prior to the start of the intervention, while other pre-assessment measures could be collected up to 4 weeks before the first treatment session.

Study 1: Evaluating the Novel Imputation Tool in *IBI VizEdit*

Overview

The least tested component of *IBI VizEdit* is its incorporation of an imputation model based on a composition of Gaussian processes. This novel imputation feature therefore represents a natural starting point for the methodological investigation of *IBI VizEdit*. Study 1, then sought to address the following two broadly construed questions: (1) Which of the three imputation models does the best job of recovering true (i.e., original) heart rate variability values? And: (2) Under what conditions do each of the

three programs perform the most efficiently (i.e., return the smallest variance in estimates)? Most accurately (i.e., are unbiased in their estimates)?

Data from parents and children enrolled in *Cool Little Kids* were used to evaluate the different candidate models. Testing this component of the program with existing data required identification of artifact-free data, a task completed by a pair of trained editors who reviewed raw PPG files from parents and children looking for sections that contained at least 3 minutes of continuously collected data that was artifact-free. A total of 153 different heart rate files were prepared in time for inclusion in the present study, each of which underwent visual review and sorting.

The code used to create models and otherwise summarize the data is available in **Appendix B**. Briefly, the three different models were used to recover randomly targeted sections of artifact-free PPG data under varying conditions that included altering the size of the window to be imputed, the sampling rate of the data used for imputation, and the window length of the surrounding data used for imputation. The efficacy of each Gaussian process imputation model was evaluated based on the degree to which a measure of heart rate variability (RMSSD from the imputed section) matched original values. Additionally, model run time was calculated as a means of understanding the practical costs of using each model. Finally, several diagnostic values and plots from each imputation run were stored to evaluate potential factors that govern model convergence and data-model fit.

Participants

Data in Study 1 were taken from children and parents who had been assigned to receive the *Cool Little Kids* intervention. All laboratory assessments (up to three per

participating parent-child dyad, $N = 76$) for which physiological data were successfully recorded and prepared in time for analysis ($N_{assessments} = 153$) were included in the initial sample frame. Overall, children enrolled in the *Cool Little Kids* program were somewhat racially diverse (44.59% White, 28.38% Mixed/Other Races, 17.57% Black/African-American, and 9.56% Asian/Asian-American) and predominantly male (54.05% boys vs. 45.95% girls) with an average age of 53.22 months ($SD = 5.52$). The majority of participating parents were the child's mother (68.92%) with an average age of 39.49 years ($SD = 5.74$).

Procedures

Heart Rate Data Collection: Hardware and Software. All data used in Study 1 were collected using the Biopac MP150 data acquisition system at a sampling rate of 2000Hz. During the assessment, target children and their parents wore separate wireless Bio-Nomadix PPG-ED transmitters (Biopac Systems, 2018). The receivers for both the child and parent devices were attached to the MP150 acquisition device in a control room approximately 9 feet from the parent and child during the assessment. Concurrent with the collection of heart rate, two separate video streams were recorded in the room. The first captured video of the child from a wall-mounted camera feed, and the second captured the presentation screen where videos and other visual cues were presented during the assessment. These two video streams were used by research assistants to identify the time boundaries for the four different conditions that occurred during each

assessment: Video (repeated three times), Clown, Kids, and Introduction (see **Figure 3.10**).

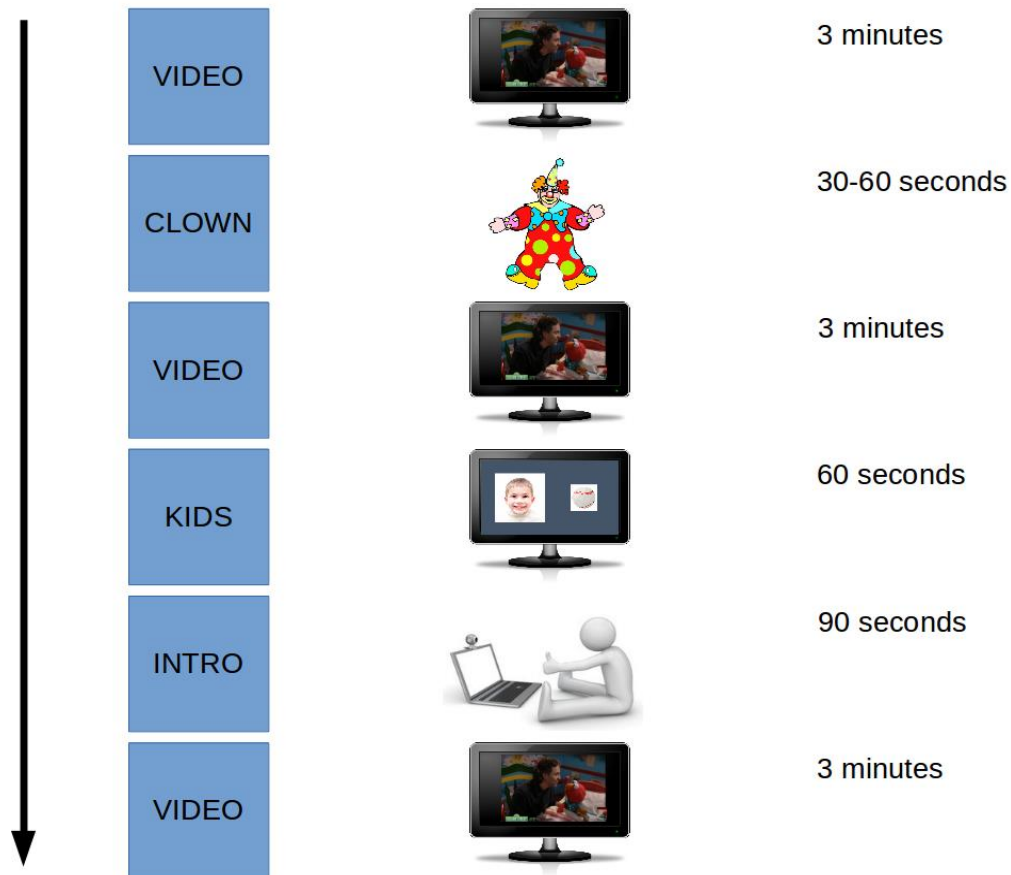


Figure 3.10 Graphic depiction of varying conditions during physiological assessments

Once in the assessment room, the experimenter attached the Bio-Nomadix PPG-ED wireless transmitter to the parent first and then the child. The transmitter was always attached to the parent's (or child's) left arm. Experimenters then used a Velcro strip to attach the light-emitting diode and photoreceptor combination that make up the standard Biopac PPG sensor. The sensor was positioned on the left index finger, with the photosensor and light-emitting diode oriented toward the palmar surface of the digit. The PPG sensor was then fastened in place using Velcro with care taken to not secure the sensor too tightly so as to reduce or limit blood flow or too loosely allowing the sensor to

easily change positions on the participant's finger. Once both the parent and the child sensors were attached, the experimenter introduced "Freddy the duck," a rubber duck fixed to the child's desk. The child was asked to help keep Freddy warm by keeping his or her hand resting on his back. If the child began to move his or her hand or fingers during the observation, the experimenter would gently remind the child to remain still and to help keep Freddy warm.

While this physical setup took place, a trained technician in the control room monitored the parent and child PPG waveform to ensure the sensors were correctly recording data before the start of the first condition (Video 1). The technician was also responsible for initially triggering the recording in *Observer 12.0*, a program designed to support the collection and coding of observational data (Noldus, 2015). The use of *Observer* in the present study allowed for the synchronization of heart rate data and the two video feeds recorded during each assessment. Biopac's proprietary software *AcqKnowledge 4.4* managed collection of the raw PPG signal (Biopac Systems, 2016). Upon completion of the physiological recording session, the raw data file was converted into a tab-delimited text file to allow for easier importing of the file into *Observer* as well as *IBI VizEdit*.

Data Re-sampling Design: Testing the Three Gaussian Process Models

Data Selection. Two trained editors independently reviewed 153 raw PPG files to identify sections of data that contained 3 minutes of continuously collected, artifact-free signal. After compiling all files that contained at least one 3-minute segment of artifact-free data (39 segments of child files and 40 segments of parent files), a re-sampling and

modeling script iterated over several different conditions, selecting random sections of data from the targeted files for imputation.

Imputation Window Length. To examine the effect of the total time to be imputed on model performance, the length of the targeted imputation windows was randomly determined for each run, drawn from a uniform distribution with a minimum value of 2 seconds and a maximum value of 8 seconds. In terms of the scripts that governed the simulation program, the selection of the portion of a given file to impute occurred first, followed by adjusting the sampling rate of the signal, then the model sampling rate, and finally the amount of data on either side of the imputation window used to generate predicted values. Thus, these aspects of the imputation model were varied within each segment targeted for imputation. Importantly, the three Gaussian process models executed their re-sampling scripts independently. Thus, while they were each based on the same basic re-sampling procedures, the three models did not sequentially attempt to impute the exact same sections of data.

Signal Sampling Rate. The number of data points per unit of time may affect imputation in one of two ways. The first is that higher sampling rates result in more data per unit of time and therefore more computationally burdensome imputation runs, leading to longer model run times. The second is that lower sampling rates reduce the amount of information the model has access to when tuning parameters and making predictions, and it is plausible that the imputed values may be less accurate as a result. As part of the imputation runs, the data re-sampling program ran the same set of imputations for each segment of data targeted for imputation at two different, down-sampled rates (125 Hz and 250 Hz). Note, the original signal was sampled at 2000 Hz.

Model Data Window. When imputing data in a univariate time series, the only available information that can be used to create predicted values are the data surrounding the targeted segment. In the present study, the size of the window used to tune the parameters of the imputation model varied proportionally with the size of the segment targeted for imputation by a factor of 1x, 2x, or 3x. What this meant is that when the imputation window was 2 seconds long, data used to impute values were taken from the 2, 4, or 6 seconds immediately preceding and immediately following the imputation window. The sampling rate of the Gaussian process model data was either 4 Hz, 8 Hz, or 12 Hz. **Figure 3.11** offers a visual depiction of how the re-sampling program worked and the resulting 18 conditions created by the different levels of the independent variables in this re-sampling study.

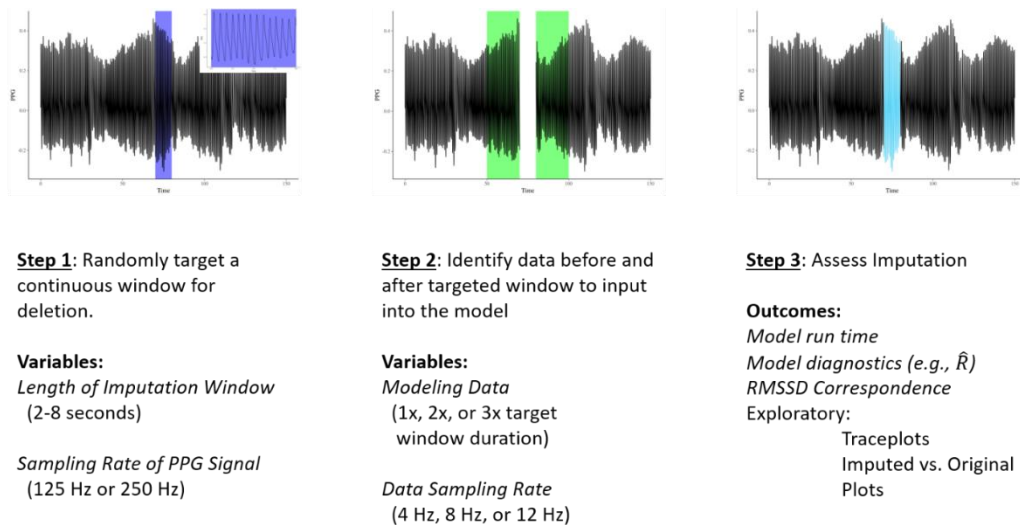


Figure 3.11 Depicting of Gaussian process imputation model testing

Note. RMSSD = Root mean square of successive differences

Bayesian MCMC Sampling Parameters. In the present study, Gaussian process models were estimated (mostly) within a Bayesian framework. In these models, the total number of warm-up iterations was fixed at 2000 per Monte Carlo Markov chain followed

by 500 sampling iterations, upon which imputed values were based. The total number of chains was fixed at 2. These values, particularly the warm-up and sampling iterations, are notably lower than recommendations often made for other Bayesian programs (Gelman et al., 2013; Gelman & Hill, 2007), and this is due, in part, to an efficient sampling algorithm employed in Stan (i.e., NUTS). These aspects of the imputation models may, however, warrant further investigation in future research. One component of the present study, assessing model convergence, may provide a useful foundation for future efforts in that vein.

Computing Environment. Model Imputation was conducted using a series of R scripts running on one of three computers. One computer was equipped with an Intel i7-8700K processor (3.7 GHz) that has 6 cores (12 logical processors with hyper-threading). The other two computers were each equipped with Intel Xeon E5-1620 processors (3.7 GHz) that each have 4 cores (8 logical processors with hyperthreading, per computer). The former hardware configuration is unlikely to represent a typical setup in a psychology laboratory. However, the latter two computers do represent mid-to-high end hardware that is more likely to be available in a typical laboratory. To reduce the total amount of time needed to complete the simulation runs, 14 models were run simultaneously across these three platforms, with 2 logical processors per model occupied by the MCMC chains in each model.

Measures

Model Fit and Diagnostics. For each simulation run, the model estimated posterior distributions of heart rate as well as basic convergence criteria for those distributions. As a means of assessing convergence, the current study relied primarily on

\hat{R} values, a measure of the degree to which two or more independent Monte Carlo Markov chains converge on similar posterior distributions. Values under 1.1 are typically thought to be indicative of acceptable convergence (Gelman et al., 2013) and regularly exceeding a value of 1.1 can be used to distinguish between good and poor model convergence on a particular parameter's posterior distribution. Additionally, traceplots of the heart rate parameter from every model were saved as part of the re-sampling and modeling script.

Model Run Time. Model run time was determined by subtracting the system date and time at the start of each model run from the system date and time recorded after the model's execution. Because Gaussian process models have a well-defined optimization problem (the larger the input matrix, the exponentially longer the model run time), it was important to understand how variation in the imputation models and the data properties influenced the time cost associated with their use.

Imputation Performance. The accuracy of each imputation run was judged based on the degree to which each model run recovered original RMSSD scores. That is, RMSSD values from the original section of data were subtracted from RMSSD values based on the imputed file. These difference scores provided a means of assessing whether a given model returned unbiased (i.e., accurate on average) and precise (i.e., less variable) estimates. Specifically, accurate and precise models would be expected to return distributions of RMSSD difference scores centered at zero with a relatively narrow distribution. To evaluate the relative fit of the Gaussian process imputation models, two additional sets of RMSSD difference scores were created using naïve approaches for dealing with missing data. The first involved employing a mean replacement strategy for

the interbeat intervals in the targeted range. The second relied on a hotdeck replacement in which interbeat interval values were randomly drawn from the remainder of the file.

Hypotheses and Analyses

Hypothesis 1. Longer windows targeted for imputation were expected to result in exponentially longer run times. This hypothesis was tested graphically, plotting the total imputation window on the x -axis and the total model run time on the y -axis.

Hypothesis 2. Increased model complexity was expected to lead to longer model run times. This hypothesis was tested by comparing average model run times across the three different Gaussian process models.

Hypothesis 3. Gaussian process models that incorporated more data (i.e., higher model sampling rate, higher signal sampling rate, and data inputs covering a longer time range relative to the section targeted for imputation) were expected to be less biased in terms of recovering heart rate variability scores. This hypothesis is multifaceted in that multiple aspects of the imputation model may differentially influence the degree to which the models returned biased estimates on average. Therefore, addressing this component of the present study entailed the creation of several graphs, with different aspects of each imputation run plotted on the x -axis and RMSSD difference scores (relative to the original scores) plotted on the y -axis.

Hypothesis 4. Gaussian process imputation models were expected to be less biased than a mean replacement strategy or a hotdeck replacement strategy. As with previous hypotheses, this aspect of the present research was evaluated using a mix of graphical and analytical approaches, comparing RMSSD difference scores and their

variability across the three Gaussian process models and the two naïve missing data strategies.

Study 2: Reliability and Validity of IBI VizEdit Outputs

Study 2 Overview

The purpose of Study 2 was to examine a real-world deployment of *IBI VizEdit*. The data from Study 2 were drawn from the same ongoing intervention study described in Study 1 of the present report. While Study 1 utilized physiological data obtained from children and parents who were assigned to receive the *Cool Little Kids* intervention and from all available time points, data for Study 2 were taken from participants who had been randomly assigned to participate in a multicomponent intervention known as the *Turtle Program* (Chronis-Tuscano et al., 2015) and only from the first of the three laboratory assessments. Additionally, Study 2 incorporated measures of children’s play behaviors, as observed in their preschool classrooms.

Given the nature of the tasks during the laboratory assessment and the inclusion of play behaviors in Study 2, it is important to stress that children in the present data set were selectively recruited based on high levels of parent-reported BI. Recruitment efforts targeted parents of “shy” children, and to be eligible parent-reported BI Questionnaire (BIQ; Bishop, Spence, & McDonald, 2003) scores needed to exceed the 85th percentile (i.e., ≥ 132).

Being selected on BIQ scores is relevant to the current study because, BI (BI) is a temperamental trait characterized by the expression of fear and wariness in novel settings and with unfamiliar individuals (Fox, Henderson, Marshall, Nichols, & Ghera, 2005). Variation in measures of BI has been consistently shown to be associated with variation

in underlying physiology, with autonomic dysregulation being a frequent biomarker of elevated BI scores (Fox, Nichols, et al., 2005; Kagan, Reznick, & Snidman, 1987, 1988). In line with the temperamental nature of BI, high levels of BI in early childhood are predictive of future anxiety disorders, social anxiety disorder in particular (Chronis-Tuscano et al., 2009; Clauss & Blackford, 2012; Hirshfeld-Becker et al., 2007; Muris, van Brakel, Arntz, & Schouten, 2011), a motivating factor underlying the development of the *Turtle Program*. Perhaps as a means of coping with their dysregulated emotional responses to novelty and uncertainty, behaviorally inhibited children are more likely to withdraw from peers in social contexts, beginning at relatively young ages (Rubin, Coplan, & Bowker, 2009). Thus, the targeted sample in Study 2 were more likely than typical children to exhibit evidence of autonomic dysregulation and engage in solitary and withdrawn behaviors around peers.

As briefly described in the Study 1 overview, one aspect of participating in the intervention study was the completion of a series of laboratory assessments. During these assessments child and parent physiology was recorded while engaging in a prescribed set of tasks, some of which were intended to be particularly distressing to a population of children elevated in BI. Leveraging this aspect of the study design to evaluate *IBI VizEdit*, Study 2 sought to address whether separate, trained editors could produce similar RSA values during each task, whether RSA would vary as expected with the different tasks during the laboratory assessment, and finally whether variation in physiology during the lab assessment was related to the enactment of socially appropriate behaviors in a real-world peer context.

Participants

Pre-intervention physiological data were successfully collected from a total of 66 out of a potential 75 child participants ($M_{\text{age}} = 52.65$ months, $SD = 6.17$). Two of the 9 missing cases were unavailable due to technician/experimenter error in the execution of the study protocols. One child refused to wear the device and continually removed it throughout the assessment. For the remaining 6 cases, their absence in the final data set was due to particularly poor signal quality. Editors simply did not have enough good heart rate signal to work from. In addition to these cases for which there were no available physiological data at the pre-intervention assessment, there was one instance in which only partial data were available due to hardware failure (the first video, the clown, and the second video could be identified in the time series, nothing else).

Children in the obtained sample were somewhat racially diverse (59.46% White, 17.57% Asian/Asian-America, 6.76% Black/African-American, and 16.22% Other or mixed-race), and majority female (56.76% girls vs. 43.24% boys). To enroll in the study, children were required to be elevated in parent-reported BI, and not surprisingly BIQ scores, as reported by parents at baseline, were high ($M = 154.50$, with possible scores ranging from 30 to 210).

Procedures

All instrumentation and physiological data collection protocols were the same for Study 2 as described for Study 1. However, the heart rate data used for Study 2 were collected at the initial laboratory assessment (i.e., prior to the start of the intervention). To the degree possible, all segments (Video, Clown, Kids, and Introduction) were edited for every *Turtle* pre-assessment child heart rate file. In Study 1, condition or task being performed during the assessment was irrelevant as the focus was primarily on testing the

functioning of the imputation program. For Study 2, the child's physiological response to each condition was a primary variable of interest (see **Figure 3.10**).

Measuring Physiological Response to Different Social Experiences. Once the sensors were tested and attached to the participant, the session started with an age-appropriate video, referred to as the *Video* condition (see **Figure 3.10**). During the assessment, children were asked to watch the same video three times, before and after certain socially challenging conditions. The Video condition was primarily intended to help establish a within-session measure of tonic, baseline physiological activity.

The *Clown* condition began as soon as the first Video condition finished. The Clown condition was designed to provide an in vivo social challenge for the child, specifically the social challenge of being asked to speak to an unfamiliar individual. A research assistant dressed as a clown knocked on the outer door of the laboratory to initiate the condition. The experimenter asked the visitor to come in, at which point the research assistant entered and waited 10 seconds while staring at the child with neutral affect. After the 10 seconds lapsed, she introduced herself as "Binky, the clown" (Binky was always performed by a female research assistant). She then asked the child, "What's your name?" If the child responded, the research assistant thanked the child and turned to the parent to inquire about his or her name. If the child did not respond within 5 seconds, the research assistant repeated, "My name is Binky. What is your name?" If the child still did not respond, the research assistant addressed the parent and said, "You must be his [or her] parent. What is your name?" After the parent responded, the research assistant asked, "Can you ask your daughter [or son] if she is willing to tell me her name?" The research assistant patiently waited as the parent prompted the child to introduce him- or herself to

Binky. After a maximum of 60 seconds had passed, Binky left the room, and the child's attention was returned to the computer screen where the Video condition repeated a second time.

Once the second run of the video finished, the experimenter explained to the child that she wanted to introduce several children who were not in the lab that day, but who the child might meet when he or she came back during future visits to the lab. The purpose of the *Kids* condition was to determine whether variation in physiology while learning about unfamiliar peers is related to treatment response. During the Kids condition, images of smiling children's faces and a simple clip art object were displayed at 10-second intervals. Smiling images for the fictitious children were taken from the *Child Affective Facial Expression* set (LoBue & Thrasher, 2014). For the first 5 seconds, only the image of a fictitious child appeared on the screen. For the next 5 seconds, the image of the child was accompanied by clip art indicating an object, activity, or place the fictitious child enjoys. When presenting each slide, the experimenter said the fictitious child's name (e.g., "This is Billy."). Then, after the first 5 seconds and in sync with the appearance of the clip art image, the experimenter followed by describing what the child liked (e.g., "Billy likes baseball."). After six such presentations of novel children the Kids condition ended.

To start the next condition, a self-introduction task, the experimenter began by directing the child's attention to a camera mounted on a nearby wall. The experimenter asked the child to record a, "brief introduction so the kids you just learned about can learn something about you." These instructions were designed to mimic aspects of the Trier Social Stress Test, which involves public speaking in front of an audience

(Kirschbaum, Pirke, & Hellhammer, 1993). The Trier Social Stress Test, is a well-established task known to elicit changes in autonomic signaling, particularly along the hypothalamic-pituitary-adrenal axis, which is a sympathetic signaling circuit (Kudielka, Hellhammer, & Kirschbaum, 2010). Following the experimenter's instructions, children were given 90 seconds to record a greeting. Notably, the camera they were asked to speak into was mounted next to a mirror to reinforce self-conscious emotional states. If the child remained silent (which was common), the experimenter offered different prompts such as, "You can tell them how old you are" or "You can describe something you like to do." Once the 90 seconds were complete, the baseline Video played for the third and final time.

Heart Rate Editing Training. A total of three different editors participated in varying degrees in the present set of studies. To certify their ability to effectively and accurately edit heart rate data, all editors were trained according to protocols and procedures outlined by the Brain-Body Center at the University of Illinois – Chicago, publishers and maintainers of *Cardio Edit* and *Cardio Batch* software. The former is a graphic user interface for editing interbeat interval files; the latter employs Porges' method for calculating RSA from interbeat interval files. To become certified as reliable editors using these programs, trainees completed an initial hour-long workshop in which they learned about the underlying neurophysiology that influences heart rate as well as the mechanics of using the aforementioned programs to obtain accurate measures of cardiac activity. After this initial workshop, the trainees worked independently on a set of 20 files provided by the Brain-Body Center for training purposes. While working on this initial set, trainees could seek advice and ask questions about files that they found

challenging from supervisors and peers. The accompanying training manual also contained notes from a master editor about strategies for the more difficult sections of these files. To demonstrate mastery and move on to the reliability files, trainees were required obtain RSA values within $\pm .05$ of the values published in the software manual for each training file.

Once a trainee successfully completed all 20 training files, they moved on to a set of 20 reliability files. The scores for these reliability files are kept by the Brain-Body Center, and trainees were required to work independently on these files. Seeking advice from supervisors and peers in the laboratory regarding these files was prohibited. Scores were emailed to the Brain-Body Center where staff evaluated whether the trainee was within the $\pm .05$ threshold for their obtained RSA values on the reliability set. If scores fell outside the target window, trainees were advised to re-edit those cases. One of the three editors needed to re-complete one of the files to establish reliability, one editor needed to re-complete two of the reliability cases, and one editor needed to re-complete four of the reliability cases. All achieved reliability upon the second submission of their scores.

After editors completed their *Cardio Edit* training, they moved on to establish their reliability with *IBI VizEdit*. Employing a similar standard of accuracy for RSA values (i.e., $\pm .05$), editors needed to complete 4 adult cases and 4 child cases drawn from the present study. These cases were initially edited by a master editor (MGB) and his scores were used as references for the purposes of establishing reliability. Importantly, the 8 cases used to certify editors to use *IBI VizEdit* were not part of the data included in Study 2. This training approach is comparable to the standards used to become a certified

editor using *Cardio Edit*, requiring the editing of 24 different segments of heart rate data for adult and child populations separately (there are a total of 6 tasks per file and editors had to complete 4 files per population). Editors trained until they obtained RSA values within threshold for each section of the file. Only after completing this step were they able to begin work editing files for the present study.

Measures

Respiratory Sinus Arrhythmia. *Cardio Batch Plus*, a custom program developed by the Brain-Body Center (2007) was used to calculate RSA values. The program relies on a series of procedures developed by Porges (1985) to extract measures of variability within the typical respiration rate of children (.24 Hz to 1.04 Hz). The approach performs well when signal to noise ratio is low, a reason to ensure accurate editing of interbeat interval files. To calculate RSA scores, the Porges procedure uses a moving polynomial filter to remove sources of variation in heart rate that result from processes slower than typical breathing rates (Porges, 1985). Next, *Cardio Batch* calculates a series of RSA estimates sequentially within user-defined epochs to reduce the effects of aberrant changes in RSA and to minimize violations of stationarity that can cause problems for frequency-based calculations of time-series data (Shumway & Stoffer, 2017). Epoch lengths were set to 15 seconds for the creation of the present set of RSA scores. The default setting for the program is 30 seconds, which, in the case of the current data, would not allow for the calculation of RSA values during the Clown condition for many participants, as the task could take as little as 30 seconds of time depending on the child's responses. *Cardio Batch* relies on averaging these user-defined epochs to create a final score, and therefore requires a minimum of two such epochs within a given file to

return a value. Once these processing steps and calculations have completed, the final phase of the procedure is to perform a logarithmic transformation of the obtained values to coerce final scores into a normal distribution.

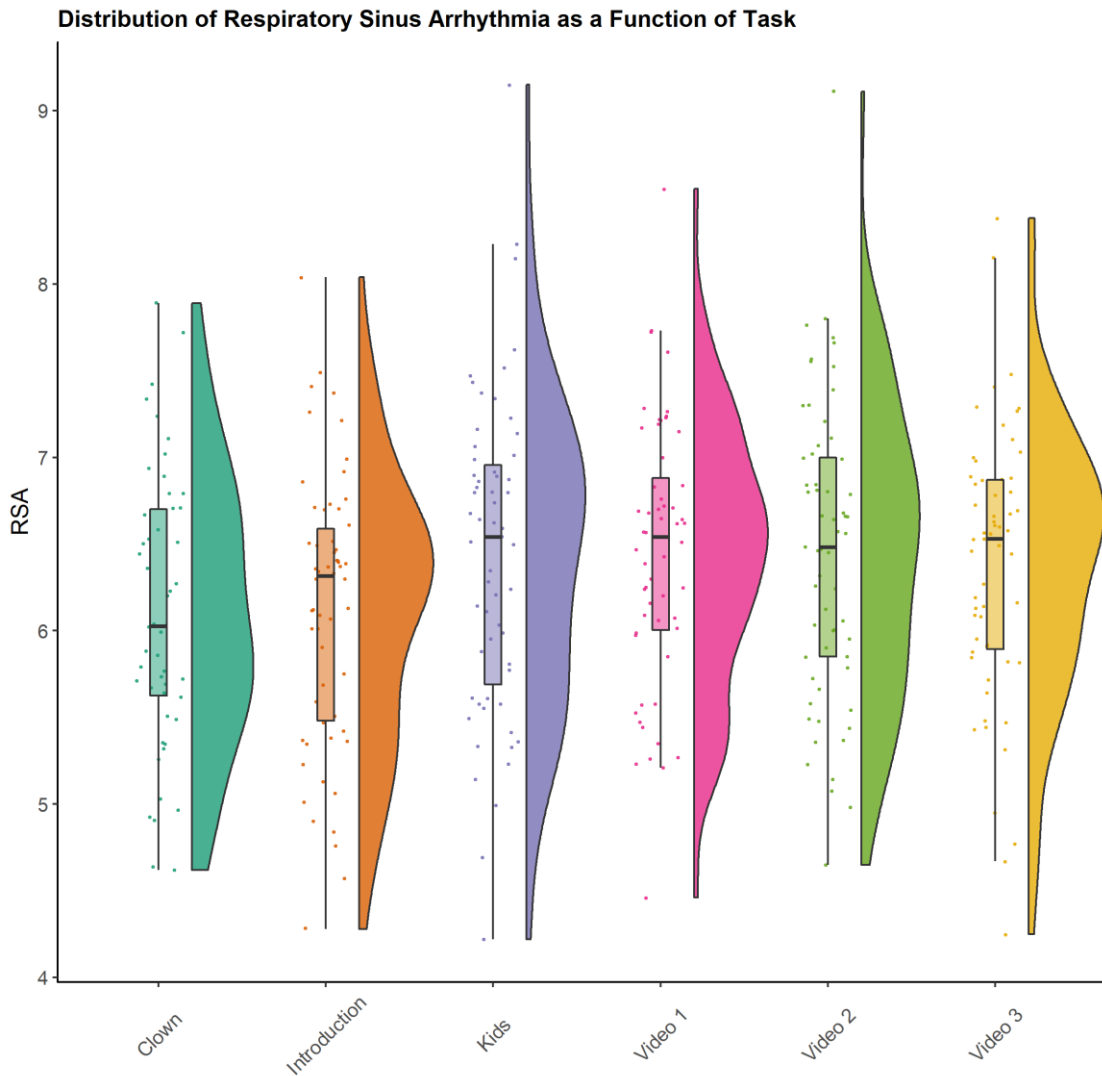


Figure 3.12 Respiratory sinus arrhythmia distributions by task

Note. RSA = respiratory sinus arrhythmia.

Calculating RSA scores using the Porges method requires continuous interbeat interval data, and cannot, therefore be performed when uneditable sections of data exists within a task. Thus, when cases included uneditable data within a task (i.e., the signal was

too weak or too contaminated with artifacts to effectively edit) RSA values were not calculated for that specific task. Study 2 descriptive statistics for RSA scores derived from *IBI VizEdit* are displayed in **Table 3.1** and raw distributions of scores, by task presented in **Figure 3.12**.

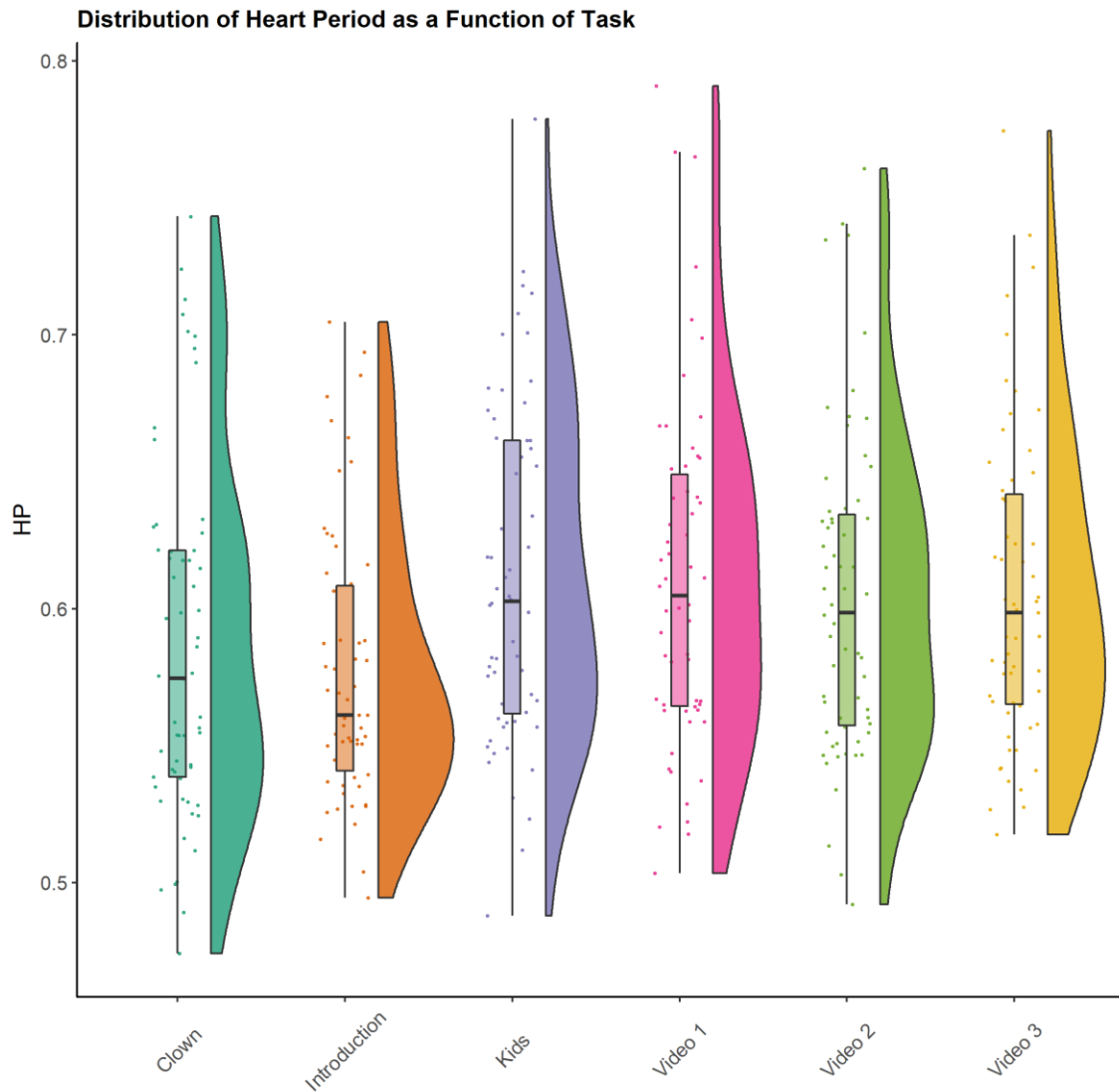


Figure 3.13 Heart period distribution by task

Note. HP = heart period.

Heart Period. The calculation of heart period is a relatively straightforward process that involves averaging the interbeat intervals in a segment of the heart rate file.

Higher values indicate a slower overall heart rate. When a portion of a task was identified as uneditable, that segment of the task was removed from the calculation and an average score was calculated from the remaining data, provided that it represented at least 60% of the task. This is an admittedly arbitrary value for a threshold, and it relies on the logic that more than half of the data from a given task was deemed usable. Descriptive statistics for heart period by task are displayed in **Table 3.1**, with graphic depictions of the resulting distributions presented in **Figure 3.13**.

Table 3.1

Table 3.1 Heart Rate Variability and Heart Period Descriptive Statistics by Task

Task	RSA		RMSSD		HP	
	<i>M</i>	<i>SD</i>	<i>M</i>	<i>SD</i>	<i>M</i>	<i>SD</i>
Video 1	6.44 (<i>n</i> = 56)	0.75	0.05 (<i>n</i> = 59)	0.02	0.61 (<i>n</i> = 58)	.06
Clown	6.15 (<i>n</i> = 51)	0.83	0.04 (<i>n</i> = 62)	0.02	0.59 (<i>n</i> = 62)	.06
Video 2	6.44 (<i>n</i> = 61)	0.83	0.05 (<i>n</i> = 62)	0.02	0.60 (<i>n</i> = 63)	.06
Kids	6.42 (<i>n</i> = 59)	0.89	0.05 (<i>n</i> = 60)	0.02	0.61 (<i>n</i> = 60)	.06
Introduction	6.11 (<i>n</i> = 58)	0.77	0.04 (<i>n</i> = 58)	0.02	0.58 (<i>n</i> = 58)	.05
Video 3	6.38 (<i>n</i> = 59)	0.79	0.05 (<i>n</i> = 59)	0.02	0.61 (<i>n</i> = 59)	.06

Root Mean Square of Successive Differences. As with heart period, the root mean square of successive differences (RMSSD) was calculated for each portion of the laboratory assessment in Study 2, provided that at least 60% of the data from a task was deemed to be usable. RMSSD is a measure of variability that assesses the amount of variation in a set of values that is due to the size of successive differences between points. Larger beat-to-beat jumps in heart rate will drive up RMSSD values, and RMSSD scores tend to correlate positively with RSA scores, though the former is a less “pure” measure of parasympathetic nervous system influence on cardiac activity than the latter (Lewis et al., 2012; Shaffer & Ginsberg, 2017). Descriptive statistics for RMSSD scores separated

by task for Study 2 are displayed in **Table 3.1**, with their distributions depicted graphically in **Figure 3.14**.

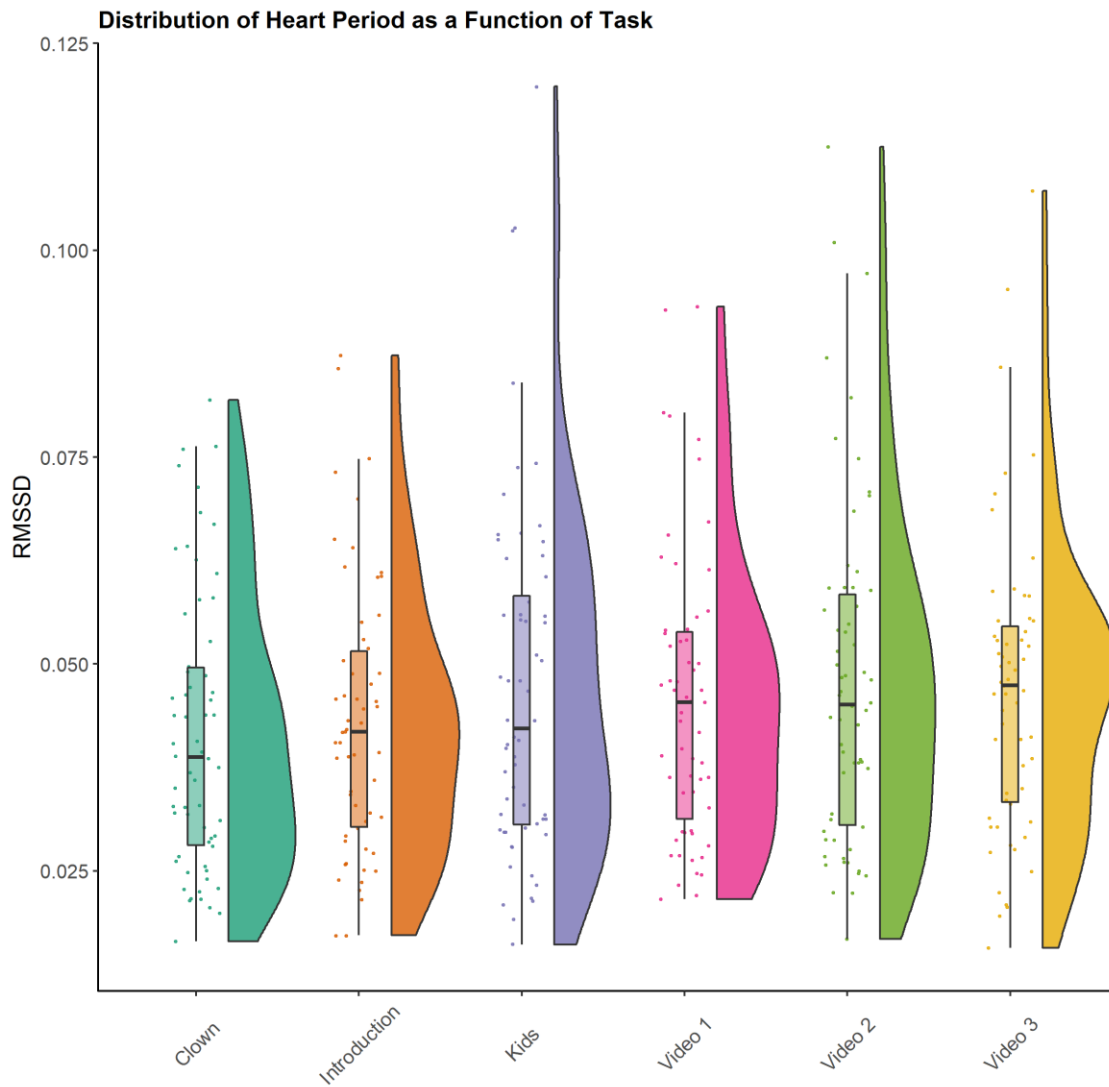


Figure 3.14 RMSSD distributions by task

Note. RMSSD = root mean square of successive differences.

Behavioral Inhibition. The intervention from which these data were drawn was focused on reducing social and emotional risk associated with elevated levels of BI and anxiety in young children. Parents were asked to report on their children’s levels of BI, a temperamental trait characterized by fear and wariness in the presence of novel objects,

situations, and individuals. BI scores were measured using the BI Questionnaire (BIQ; Bishop et al., 2003), which assessed parents' perceptions of their child's inhibition in social contexts and when facing novel situations. Parents rated how often a series of 30 statements apply to their child (e.g., *Approaches new situations or activities very hesitantly; Is reluctant to approach a group of unfamiliar children to ask to join in*) using a 7-point scale anchored by *Hardly Ever* and *Almost Always*. Average BIQ scores were used as an auxiliary variable when imputing missing values for Study 2 as variation in BI has long been tied to differences in underlying autonomic functioning (Calkins, Fox, & Marshall, 1996; Hastings, Sullivan, et al., 2008; Kagan et al., 1987; Schmidt, Fox, Schulkin, & Gold, 1999). Internal consistency scores for the parent-report version of the BIQ have ranged from adequate to good when used with preschool-aged children in the past (i.e., α 's = .72 to .97; Bishop et al., 2003). Cronbach's alpha for the total scale score in the present study was .87.

Play Observation Scale. The Play Observation Scale (POS; Rubin, 2008) was created to assess young children's play behaviors during periods of free play in the presence of similarly aged peers. POS coders were trained to distinguish between onlooking/reticent behavior, solitary functional-dramatic play, solitary constructive-exploratory play, parallel play, group/social play, and teacher/adult interactions. These play categories are mutually exclusive within the coding scheme, which breaks the observation window into 10-second epochs. For each 10-second period, a trained coder identifies the dominant play behavior for that segment of time. Onlooking/reticent behavior occurs when children remain unoccupied, often locate themselves at the periphery of the social group and display a tendency to watch others play rather than join

in. Solitary functional-dramatic play involves pretense, but absent a social partner who shares in the invented reality. Playing with figures and dolls alone, away from other children is one example of such play. This category also includes repetitive, non-exploratory behaviors that appear to have little purpose beyond engaging in the behavior itself. For example, a child who repeatedly strikes one object with another without a clear objective beyond causing the two items to strike one another is coded as functional-dramatic (emphasis on the functional-sensorimotor component in this case). Solitary constructive behavior encompasses exploratory behaviors that appear to be aimed at understanding how to use an object as well as the creation of more complex items from less complex ones (e.g., building with blocks). Parallel play occurs when two children are engaged in similar play activities near one another and those play behaviors are not interdependent in any way. Often the children are aware of each other, and may even adopt similar play activities over time, but there is no contingency to their actions. Group play, in contrast, does involve interaction and requires that two or more children are engaged in play activities that involve shared expectations/goals. Interactive games, shared pretend play, and conversations all fall into this category. The teacher/adult category is a straightforward code, and it occurs any time the child is interacting with one of the adults in the room (as opposed to peers) for the plurality of time in a ten-second epoch. Focusing solely on peer-related behaviors, this latter category was not utilized in the present study.

Training for the POS involved several stages. First, trainees attended an hour-long training. After the initial training, trainees had to achieve a kappa of .80 on a total of 8 of pre-recorded videos with a set of master codes. The videos were 10-minute periods of

free play involving between 4 and 7 children at a time, one of which was the target child. Once they had established reliability using pre-recorded periods of free play, trainees then completed a live observation, in which they were required to obtain a kappa of .80 with another coder. Having completed all steps in this training protocol coders were then allowed to complete school observations for the present study.

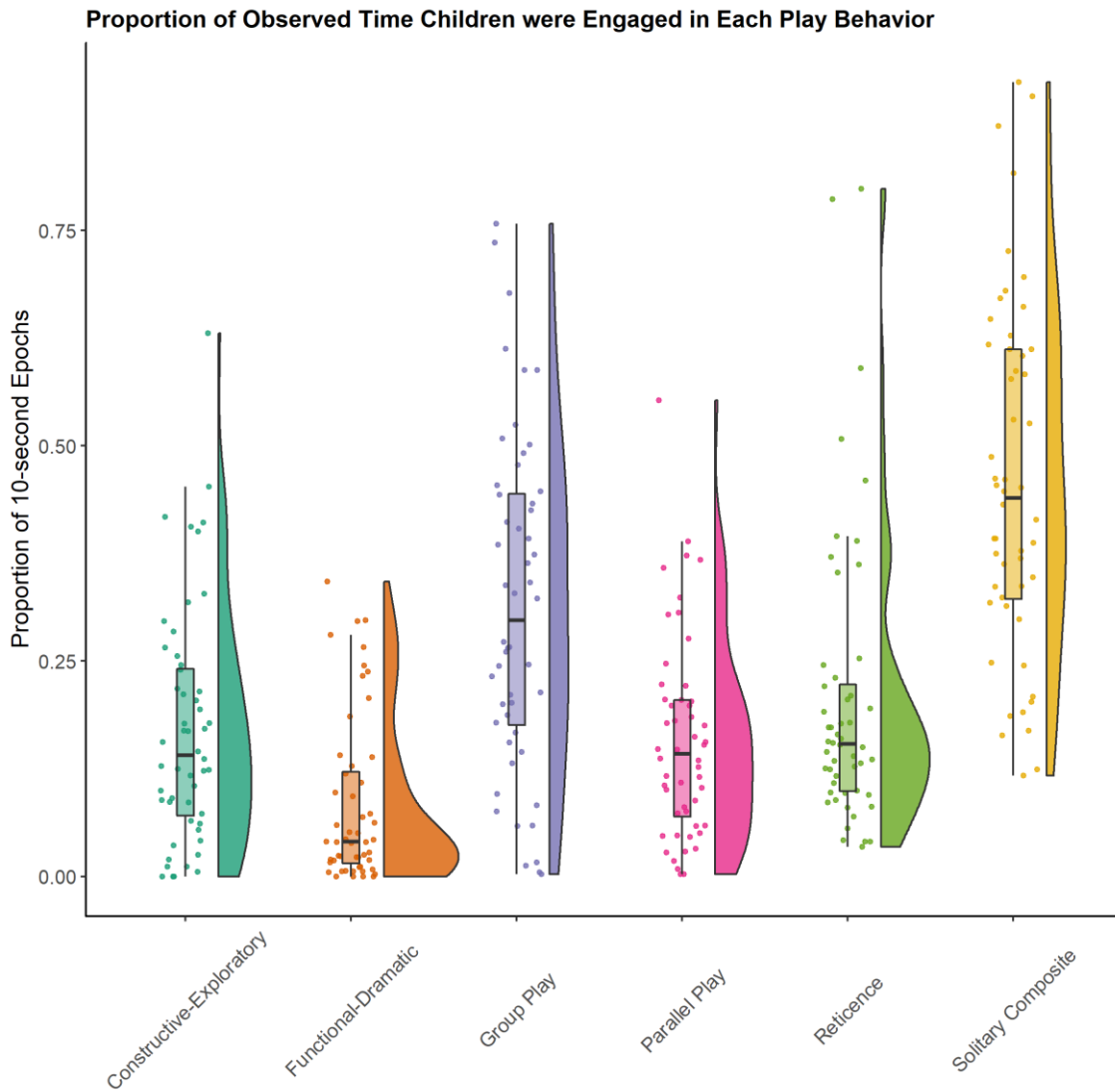


Figure 3.15 Distributions of proportion of 10-second epochs children engaged in each coded play behavior

In terms of data collection, the goal was to observe children for 60 minutes prior to the start of the treatment period. Typically, coders would obtain 30 minutes of observational data, per visit, during periods of free play in children's preschools, necessitating at least 2 observations during this baseline period. Due to classroom schedules, and variations in the amount of free play that took place on any one day, observations may have occurred over three or more days. These practical limitations also occasionally resulted in the inability to obtain a full 60 minutes of data for a given child, in which case the goal was always to obtain as much observational data as possible. Within the subsample of children who were assigned to the *Turtle Program*, who had usable physiological data, and whose schools allowed study personnel onsite for observations ($n = 52$ out of the 75 children randomly assigned to the treatment, see **Figure 3.15**), total classroom observation time lasted an average of 55.65 minutes ($SD = 10.36$, minimum = 17.33).

Hypotheses and Proposed Analyses

Hypothesis 1. Editors were expected to be able to reliably use *IBI VizEdit* to edit individual heart rate data as evidenced by *intra*-editor comparisons between RSA values derived from *IBI VizEdit* and a similar, established editing program. Correspondence was evaluated using the Brain-Body Center's training standard of obtaining RSA values within +/- .05 of each other. Additionally, simple correlations were used to test relative correspondence between scores for each editor.

Hypothesis 2. Editors were expected to be able to reliably use *IBI VizEdit* to edit individual heart rate data as evidenced by *inter*-editor reliability. Independent editors separately edited the same files using *IBI VizEdit* and correspondence between RSA

scores for each co-edited task was assessed using the Brain-Body Center's standard of +/- .05 in addition to bivariate correlations.

Hypothesis 3. Editors using *IBI VizEdit* were expected to arrive at valid estimates of heart rate and heart rate variability as evidenced by changes in relevant scores (i.e., RSA, RMSSD, and heart period) in response to socially stressful tasks experienced during laboratory assessment.

Hypothesis 4. Editors using *IBI VizEdit* were expected to arrive at valid estimates of heart rate and heart rate variability as evidenced by the capability of RSA and heart period scores derived from the laboratory tasks to predict children's social behaviors during periods of free play.

CHAPTER 4

Study 1 Results

A total of 40 segments of data were randomly targeted for deletion and imputation for each Gaussian process model. Thus, a combined 40 randomly targeted segments x 18 conditions x 3 different imputation models resulted in a total of 2,160 separate model runs. Summary statistics for each of the three Gaussian process imputation models are presented in **Tables 4.1** through **4.6**⁴. Each table includes summary information regarding model convergence, model run time, model-imputed RMSSD deviation from original RMSSD values and more, separately by model (i.e., imputation model 1, model 2, or model 3) and the down-sampling rate of the original signal (i.e., either 125 Hz or 250 Hz). Below, several key elements of these summaries are briefly described and in several instances accompanying graphics have been created to highlight certain aspects of each model's performance in recovering original RMSSD values.

Model Convergence. In the present modeling framework, there was one posterior distribution that was estimated for all three imputation models – heart rate frequency. Using the potential scale reduction factor score (i.e., \hat{R}) as a means of assessing convergence of the two MCMC chains on the same posterior distribution in each model run, the results (depicted in **Figure 4.1**) clearly indicate that the two more complex imputation models regularly failed to achieve \hat{R} values at or below the commonly referenced standard of 1.1, and instead frequently generated MCMC chains that exhibited relatively large divergence in their final posterior distributions. Relatedly, all three models, at least on certain runs, converged on mean heart rate values that were

⁴ Code governing model aggregation is available in **Appendix D**

implausibly high, with some maximum mean estimates exceeding 150 BPM, a value typically associated with aerobic activity (see **Tables 4.1-4.6**).

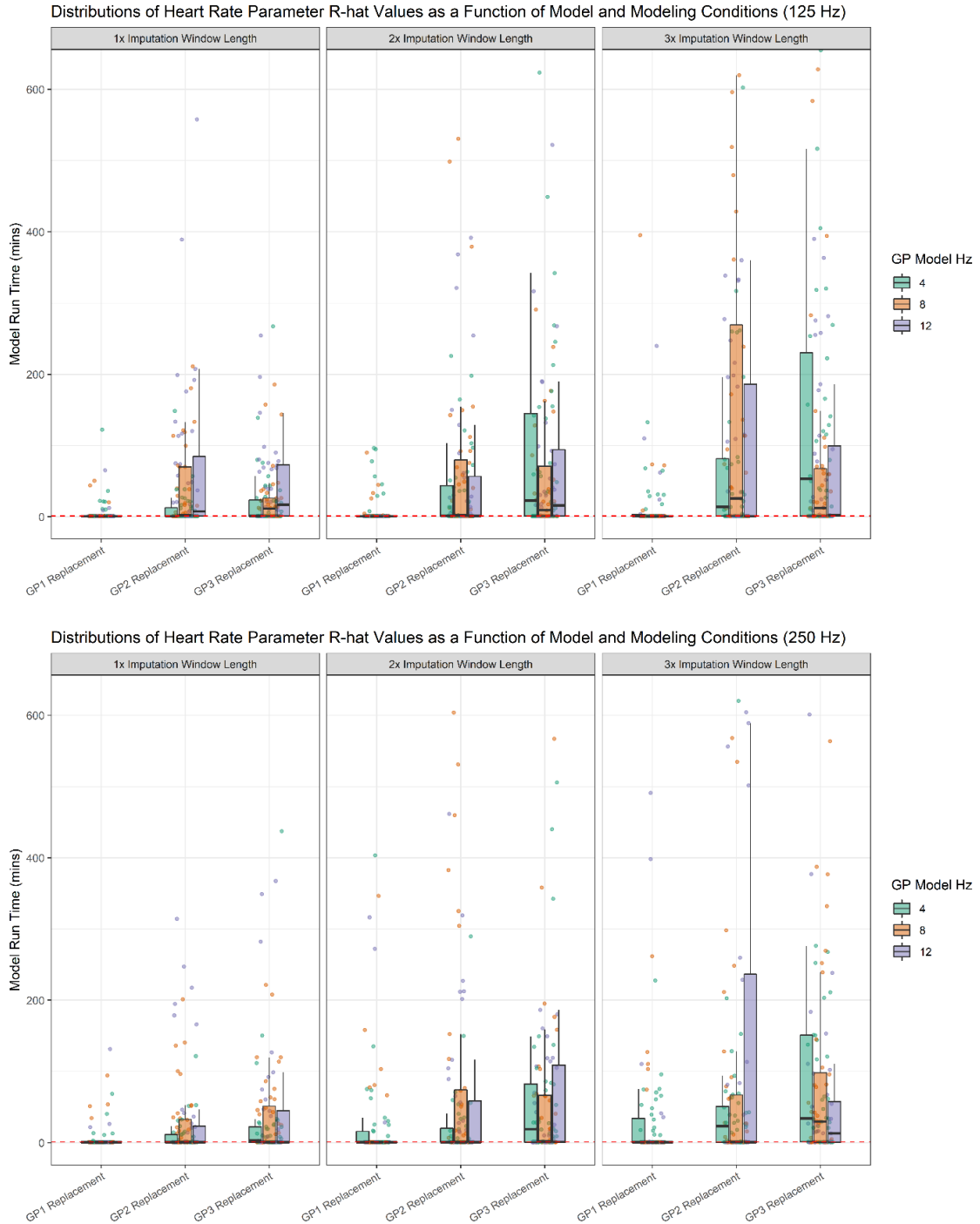


Figure 4.1 Convergence of Gaussian process models under varying conditions as measured by scale reduction factor of the model-estimated heart rate parameter

Table 4.1

Table 4.1 Gaussian Process Model 1 Summary with Data Down-Sampled to 125 Hz

	GP Model = 4 Hz			GP Model = 8 Hz			GP Model = 12 Hz		
	1x Window	2x Window	3x Window	1x Window	2x Window	3x Window	1x Window	2x Window	3x Window
Median Bias	0.013	0.024	0.023	0.0002	0.002	0.003	0.002	0.002	0.002
Bias Q75	0.052	0.050	0.043	0.022	0.034	0.021	0.029	0.017	0.011
Bias Q25	0.002	0.006	0.005	-0.002	-0.002	-0.002	-0.001	-0.001	-0.002
Mean Bias	0.029	0.033	0.044	0.037	0.026	0.011	0.065	0.014	0.029
SD Bias	0.050	0.033	0.116	0.153	0.065	0.019	0.238	0.026	0.095
Max Deviation	0.291	0.119	0.741	0.964	0.365	0.087	1.622	0.095	0.579
Min Deviation	-0.007	-0.004	-0.001	-0.015	-0.022	-0.007	-0.020	-0.019	-0.009
Mean HR (BPM)	88.607	89.234	87.086	86.247	84.993	86.600	83.991	84.616	84.327
SD HR (BPM)	21.075	23.960	19.193	18.007	15.983	18.755	16.969	17.219	17.608
Max HR (BPM)	146.071	166.807	146.580	135.550	112.784	139.842	116.849	118.197	138.937
Min HR (BPM)	54.532	53.663	55.074	53.481	53.730	54.590	53.347	52.358	53.216
Median HR \hat{R}	1.004	1.003	1.005	1.002	1.001	1.003	1.001	1.002	1.001
Mean HR \hat{R}	6.962	11.844	35.930	22.275	5.867	14.652	3.100	20.158	11.866
% HR $\hat{R}'s < 1.1$	82.50	80	75	90	85	90	92.500	97.500	87.500
Impute Total (secs)	4.606	4.606	4.606	4.606	4.606	4.606	4.606	4.606	4.606
Median Runtime (mins)	3.220	4.330	5.859	4.351	8.109	14.294	5.948	13.594	30.271
Mean Runtime (mins)	4.707	6.259	9.198	6.146	11.131	25.861	8.386	20.865	45.272
Max Runtime (mins)	17.192	24.495	35.039	21.307	36.729	160.860	27.129	72.240	225.060
Min Runtime (mins)	0.745	1.008	1.222	0.966	1.654	2.671	1.198	2.838	5.647

Note. Bias is measured as the deviation of model imputed RMSSD values from original values in a segment of data that combines the region of the file targeted for imputation with 15 seconds of data, evenly distributed on either side of the segment targeted for imputation. Bias Q75 = the RMSSD deviation value at the 75th percentile. Bias Q25 = the RMSSD deviation value at the 25th percentile. HR = heart rate. BPM = beats per minute. \hat{R} = potential scale reduction factor. Results are based on 40 separate, randomly sampled sections of data.

Table 4.2 Gaussian Process Model 1 Summary with Data Down-Sampled to 250 Hz

	GP Model = 4 Hz			GP Model = 8 Hz			GP Model = 12 Hz		
	1x Window	2x Window	3x Window	1x Window	2x Window	3x Window	1x Window	2x Window	3x Window
Median Bias	0.018	0.019	0.024	0.0001	0.005	0.0002	0.002	0.001	0.002
Bias Q75	0.049	0.044	0.048	0.017	0.020	0.023	0.029	0.018	0.033
Bias Q25	0.001	0.004	0.012	-0.004	-0.002	-0.003	-0.001	-0.002	-0.001
Mean Bias	0.074	0.057	0.063	0.009	0.069	0.049	0.041	0.013	0.042
SD Bias	0.211	0.179	0.209	0.021	0.275	0.157	0.153	0.025	0.156
Max Deviation	1.109	1.136	1.341	0.081	1.534	0.746	0.965	0.094	0.990
Min Deviation	-0.012	-0.004	-0.011	-0.020	-0.020	-0.006	-0.019	-0.018	-0.007
Mean HR (BPM)	86.852	88.829	89.370	86.637	86.676	84.621	83.842	84.969	84.562
SD HR (BPM)	18.274	22.341	23.591	18.337	19.858	17.609	17.040	18.385	18.172
Max HR (BPM)	123.324	146.917	141.129	144.475	143.608	145.062	121.430	137.729	147.436
Min HR (BPM)	53.868	53.613	55.071	53.297	53.707	54.549	53.327	52.377	53.197
Median HR \hat{R}	1.001	1.004	1.004	1.001	1.001	1.001	1.001	1.002	1.000
Mean HR \hat{R}	4.597	23.407	23.679	6.753	21.681	17.820	5.418	16.357	27.817
% HR $\hat{R}'_s < 1.1$	85	70	60	90	85	85	92.500	92.500	82.500
Impute Total (secs)	4.610	4.610	4.610	4.610	4.610	4.610	4.610	4.610	4.610
Median Runtime (mins)	13.911	14.460	18.556	16.142	20.840	28.425	18.526	29.301	42.649
Mean Runtime (mins)	18.473	20.887	24.233	20.704	27.381	37.549	23.778	38.363	64.101
Max Runtime (mins)	57.328	61.800	66.420	64.440	76.800	111	76.560	145.620	184.560
Min Runtime (mins)	2.977	3.246	3.616	3.241	3.968	5.143	3.574	5.520	7.793

Note. Bias is measured as the deviation of model imputed RMSSD values from original values in a segment of data that combines the region of the file targeted for imputation with 15 seconds of data, evenly distributed on either side of the segment targeted for imputation. Bias Q75 = the RMSSD deviation value at the 75th percentile. Bias Q25 = the RMSSD deviation value at the 25th percentile. HR = heart rate. BPM = beats per minute. \hat{R} = potential scale reduction factor. Results are based on 40 separate, randomly sampled sections of data.

Table 4.2

Table 4.3

Table 4.3 Gaussian Process Model 2 Summary with Data Down-Sampled to 125 Hz

	GP Model = 4 Hz			GP Model = 8 Hz			GP Model = 12 Hz		
	1x Window	2x Window	3x Window	1x Window	2x Window	3x Window	1x Window	2x Window	3x Window
Median Bias	0.016	0.040	0.044	0.013	0.026	0.012	0.010	0.030	0.018
Bias Q75	0.050	0.060	0.067	0.043	0.059	0.042	0.043	0.051	0.038
Bias Q25	0.002	0.007	0.013	0.002	0.002	0.00003	0.004	0.008	0.001
Mean Bias	0.049	0.040	0.041	0.024	0.032	0.030	0.030	0.041	0.027
SD Bias	0.151	0.033	0.033	0.030	0.033	0.049	0.038	0.045	0.033
Max Deviation	0.963	0.114	0.115	0.115	0.103	0.254	0.156	0.155	0.123
Min Deviation	-0.005	-0.007	-0.009	-0.004	-0.010	-0.026	-0.007	-0.011	-0.003
Mean HR (BPM)	88.319	91.327	87.652	89.862	87.340	94.086	85.294	89.185	88.379
SD HR (BPM)	21.694	26.388	23.487	25.614	21.996	28.247	26.077	22.600	25.236
Max HR (BPM)	127.625	146.151	135.643	146.653	135.955	162.550	180.401	128.577	141.926
Min HR (BPM)	54.241	53.962	51.939	51.375	52.998	52.078	50.005	50.105	50.280
Median HR \hat{R}	1.007	1.531	14.056	2.701	2.793	25.782	7.677	1.034	1.007
Mean HR \hat{R}	11.748	34.606	137.840	57.640	96.947	418.962	65.946	150.873	97.976
% HR $\hat{R}'s < 1.1$	67.500	47.500	37.500	50	50	45	45	52.500	60
Impute Total (secs)	4.522	4.522	4.522	4.522	4.522	4.522	4.522	4.522	4.522
Median Runtime (mins)	6.407	9.355	13.870	8.257	17.389	34.971	14.107	34.767	68.970
Mean Runtime (mins)	7.881	12.397	18.644	11.317	24.214	63.456	16.761	55.308	99.515
Max Runtime (mins)	22.558	45.340	66.600	30.281	90.780	691.380	61.260	503.400	772.800
Min Runtime (mins)	1.957	2.367	3.561	2.254	4.045	5.538	2.805	8.575	9.518

Note. Bias is measured as the deviation of model imputed RMSSD values from original values in a segment of data that combines the region of the file targeted for imputation with 15 seconds of data, evenly distributed on either side of the segment targeted for imputation. Bias Q75 = the RMSSD deviation value at the 75th percentile. Bias Q25 = the RMSSD deviation value at the 25th percentile. HR = heart rate. BPM = beats per minute. \hat{R} = potential scale reduction factor. Results are based on 40 separate, randomly sampled sections of data.

Table 4.4 Gaussian Process Model 2 Summary with Data Down-Sampled to 250 Hz

	GP Model = 4 Hz			GP Model = 8 Hz			GP Model = 12 Hz		
	1x Window	2x Window	3x Window	1x Window	2x Window	3x Window	1x Window	2x Window	3x Window
Median Bias	0.011	0.027	0.031	0.008	0.006	0.012	0.011	0.024	0.005
Bias Q75	0.050	0.047	0.051	0.027	0.024	0.027	0.029	0.044	0.025
Bias Q25	0.0003	0.009	0.009	-0.0005	-0.0002	0.0003	0.002	0.004	-0.001
Mean Bias	0.093	0.032	0.034	0.017	0.019	0.024	0.026	0.029	0.042
SD Bias	0.319	0.033	0.028	0.023	0.030	0.035	0.040	0.032	0.160
Max Deviation	1.821	0.135	0.116	0.092	0.133	0.123	0.152	0.137	1.010
Min Deviation	-0.013	-0.005	-0.003	-0.006	-0.007	-0.006	-0.014	-0.004	-0.037
Mean HR (BPM)	87.915	88.006	88.854	85.946	88.436	86.920	86.736	91.344	92.343
SD HR (BPM)	23.132	22.314	25.527	23.620	26.100	22.059	22.211	29.199	27.175
Max HR (BPM)	133.172	132.193	143.701	143.643	144.511	135.465	136.045	172.873	152.295
Min HR (BPM)	54.246	54.052	54.260	51.301	50.563	53.733	53.877	53.965	53.518
Median HR \hat{R}	1.006	1.021	23.458	1.008	1.007	1.052	1.005	1.003	1.003
Mean HR \hat{R}	10.532	22.551	48.734	25.592	102.897	122.893	38.954	54.956	214.582
% HR \hat{R} 's < 1.1	65	52.500	37.500	60	57.500	52.500	62.500	57.500	60
Impute Total (secs)	4.526	4.526	4.526	4.526	4.526	4.526	4.526	4.526	4.526
Median Runtime (mins)	23.396	27.727	31.315	26.465	39.356	50.383	30.579	48.615	86.190
Mean Runtime (mins)	29.327	33.497	42.958	33.937	50.306	66.672	51.744	68.733	138.831
Max Runtime (mins)	74.460	88.620	214.860	90.780	129.720	174	578.820	200.040	714.180
Min Runtime (mins)	6.961	7.604	8.271	7.492	9.184	11.032	8.107	10.990	15.002

Note. Bias is measured as the deviation of model imputed RMSSD values from original values in a segment of data that combines the region of the file targeted for imputation with 15 seconds of data, evenly distributed on either side of the segment targeted for imputation. Bias Q75 = the RMSSD deviation value at the 75th percentile. Bias Q25 = the RMSSD deviation value at the 25th percentile. HR = heart rate. BPM = beats per minute. \hat{R} = potential scale reduction factor. Results are based on 40 separate, randomly sampled sections of data.

Table 4.4

Table 4.5 Gaussian Process Model 3 Summary with Data Down-Sampled to 125 Hz

	GP Model = 4 Hz			GP Model = 8 Hz			GP Model = 12 Hz		
	1x Window	2x Window	3x Window	1x Window	2x Window	3x Window	1x Window	2x Window	3x Window
Median Bias	0.014	0.025	0.029	0.015	0.011	0.022	0.024	0.009	0.018
Bias Q75	0.040	0.067	0.049	0.042	0.047	0.045	0.044	0.031	0.048
Bias Q25	0.004	0.007	0.011	0.001	0.001	0.002	0.004	-0.0001	0.002
Mean Bias	0.025	0.040	0.034	0.023	0.025	0.026	0.030	0.020	0.029
SD Bias	0.026	0.039	0.030	0.029	0.032	0.029	0.033	0.029	0.036
Max Deviation	0.110	0.128	0.115	0.109	0.133	0.107	0.151	0.120	0.123
Min Deviation	-0.004	-0.015	-0.011	-0.007	-0.013	-0.018	-0.003	-0.012	-0.018
Mean HR (BPM)	89.765	90.493	92.673	89.621	93.281	89.012	86.784	89.155	92.538
SD HR (BPM)	23.048	25.277	27.198	23.485	27.499	26.224	21.046	25.787	31.846
Max HR (BPM)	146.485	138.487	152.727	153.001	160.818	175.407	141.030	156.625	182.429
Min HR (BPM)	51.916	58.731	58.839	56.174	56.811	50.121	57.898	50.008	50.041
Median HR \hat{R}	1.136	22.884	53.588	11.581	9.588	12.437	17.229	16.232	2.595
Mean HR \hat{R}	23.898	119.558	189.061	24.969	133.558	94.382	93.785	123.053	86.429
% HR $\hat{R}'_e < 1.1$	47.500	40	30	47.500	45	45	42.500	42.500	47.500
Impute Total (secs)	4.382	4.382	4.382	4.382	4.382	4.382	4.382	4.382	4.382
Median Runtime (mins)	8.159	12.393	20.894	12.236	26.603	51.485	21.409	44.805	114.420
Mean Runtime (mins)	10.424	16.104	25.044	14.630	36.711	66.965	22.134	69.039	205.574
Max Runtime (mins)	28.383	57.819	69.660	38.249	224.700	343.440	58.741	285.180	2,512.800
Min Runtime (mins)	1.261	1.719	2.721	1.556	2.750	5.777	2.735	6.718	11.539

Note. Bias is measured as the deviation of model imputed RMSSD values from original values in a segment of data that combines the region of the file targeted for imputation with 15 seconds of data, evenly distributed on either side of the segment targeted for imputation. Bias Q75 = the RMSSD deviation value at the 75th percentile. Bias Q25 = the RMSSD deviation value at the 25th percentile. HR = heart rate. BPM = beats per minute. \hat{R} = potential scale reduction factor. Results are based on 40 separate, randomly sampled sections of data.

Table 4.5

Table 4.6 Gaussian Process Model 3 Summary with Data Down-Sampled to 250 Hz

	GP Model = 4 Hz			GP Model = 8 Hz			GP Model = 12 Hz		
	1x Window	2x Window	3x Window	1x Window	2x Window	3x Window	1x Window	2x Window	3x Window
Median Bias	0.019	0.031	0.045	0.020	0.010	0.016	0.013	0.014	0.025
Bias Q75	0.045	0.061	0.079	0.028	0.040	0.069	0.040	0.032	0.050
Bias Q25	0.003	0.008	0.019	0.001	0.002	0.006	0.003	-0.0004	0.001
Mean Bias	0.025	0.035	0.048	0.022	0.025	0.040	0.025	0.027	0.030
SD Bias	0.023	0.031	0.034	0.035	0.036	0.043	0.031	0.039	0.033
Max Deviation	0.069	0.107	0.117	0.189	0.159	0.144	0.111	0.136	0.116
Min Deviation	-0.009	-0.007	-0.001	-0.009	-0.007	-0.020	-0.006	-0.014	-0.025
Mean HR (BPM)	91.131	93.068	92.569	85.118	90.848	92.925	88.628	88.482	88.743
SD HR (BPM)	23.969	26.377	26.030	18.787	26.784	27.394	22.646	29.099	27.741
Max HR (BPM)	143.535	150.536	147.507	119.789	151.976	160.532	144.372	162.937	157.777
Min HR (BPM)	58.112	58.236	56.116	58.082	58.231	50.151	54.661	50.406	50.808
Median HR, \hat{R}	3.423	19.033	34.252	1.019	2.462	29.823	1.006	1.423	13.223
Mean HR, \hat{R}	25.940	84.932	166.348	35.164	98.408	108.308	62.344	107.726	74.759
% HR, $\hat{R}' \leq 1.1$	50	30	22.500	52.500	50	32.500	62.500	47.500	40
Impute Total (secs)	4.386	4.386	4.386	4.386	4.386	4.386	4.386	4.386	4.386
Median Runtime (mins)	31.189	36.593	47.026	36.070	51.312	79.170	41.526	71.400	113.250
Mean Runtime (mins)	38.205	43.995	54.518	43.192	73.025	97.639	51.762	92.079	184.237
Max Runtime (mins)	109.740	128.100	151.080	120.420	566.280	265.740	151.860	408.240	1,396.200
Min Runtime (mins)	4.662	5.152	5.961	5.138	7.078	11.350	6.557	9.868	15.327

Note: Bias is measured as the deviation of model imputed RMSSD values from original values in a segment of data that combines the region of the file targeted for imputation with 15 seconds of data, evenly distributed on either side of the segment targeted for imputation. Bias Q75 = the RMSSD deviation value at the 75th percentile. Bias Q25 = the RMSSD deviation value at the 25th percentile. HR = heart rate. BPM = beats per minute. \hat{R} = potential scale reduction factor. Results are based on 40 separate, randomly sampled sections of data.

Table 4.6

Across all conditions, the second Gaussian process imputation model, which included a function to model heart rate as well as heart rate variability as a function of respiration, resulted in \hat{R} values of 1.1 or greater in 46.67% of the runs. The third Gaussian process model, which added a function to model heart rate variability related to background metabolic processes, performed slightly worse in terms of convergence, producing \hat{R} values ≥ 1.1 in 56.94% of the model runs. By comparison, only 15.42% of the first Gaussian process model runs resulted in heart rate parameter \hat{R} values that exceeded 1.1. This first model was the simplest of the three and included a singular heart rate parameter that decayed in its effect over time. Additional detail regarding convergence of models on a posterior distribution for heart rate is displayed in **Tables 4.1 – 4.6**.

Model Run Time. In line with expectations, as model complexity increased, and as the total amount of data increased, corresponding and exponential increases in model run time were observed. As displayed in **Figure 4.2**, there were some instances in which model run time (in minutes) exceeded what any researcher may deem practically reasonable, particularly for the two more complex Gaussian process models. The longest model run took 2512.80 minutes, or 41.88 hours (not depicted in **Figure 4.2**), an exceedingly long time to impute a section of data representing only 6.07 seconds of total assessment time.

The exponential increase in model run time as a function of increased data incorporated in the imputation model can be seen in **Figure 4.3**. As the sampling rate for the imputation model (either 4 Hz, 8 Hz, or 12 Hz) increases, a non-linear increase in model

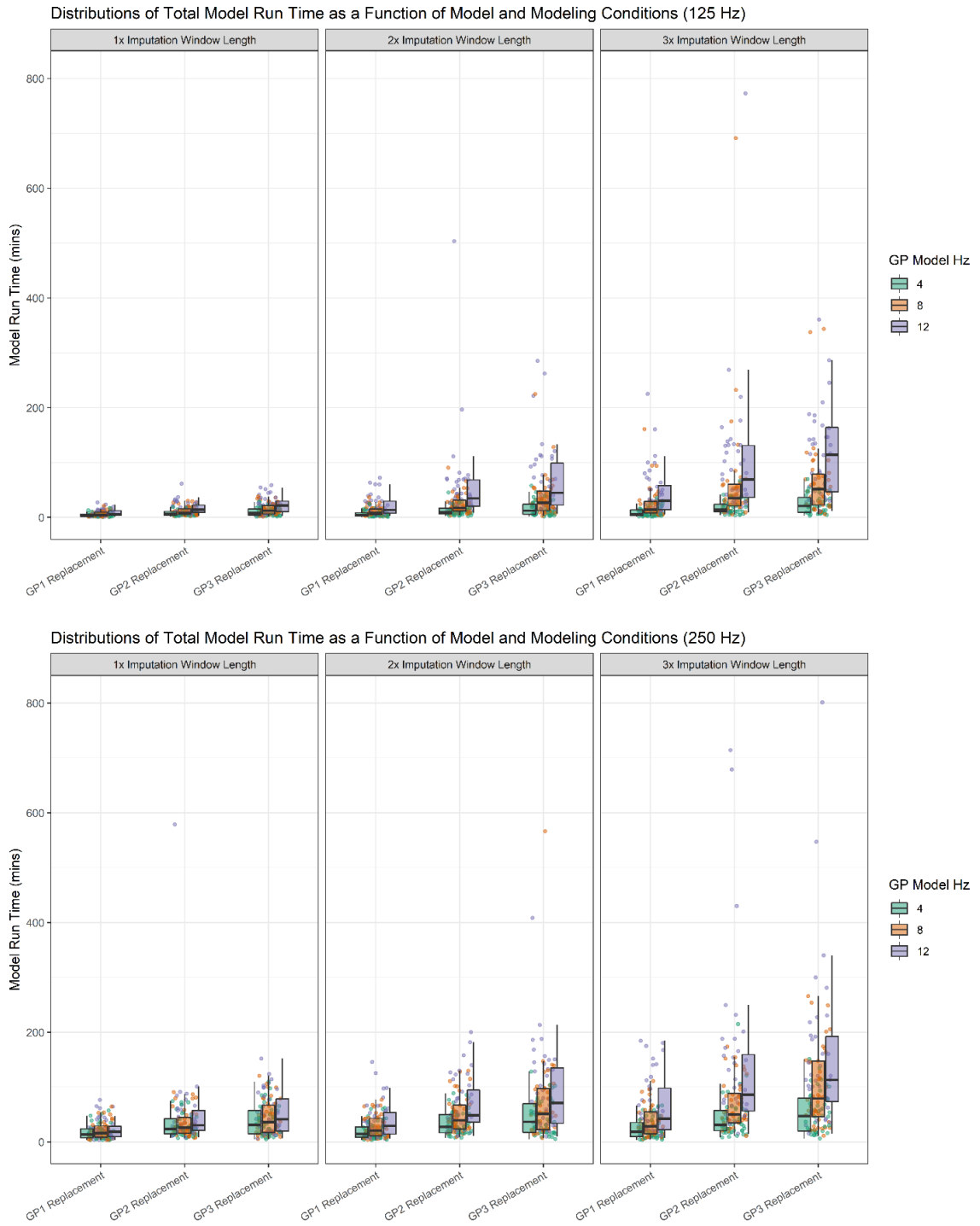


Figure 4.2 Model run time for each Gaussian process model under varying conditions

run time can be detected. A more obvious display of this phenomenon can be observed by plotting model run time as a function of the total amount of time being imputed. Across all models, the amount of data entering the model is driven by a combination of the total amount of time targeted for imputation, the data sampling rate (either 125 Hz or 250 Hz), the length of the window used for imputation, and the model sampling rate. As expected factors driving up the amount of data used to tune the model parameters and the amount of data output in the form of predicted values was related to exponentially increasing model run times. This pattern is most obvious in the two more complex imputation models (see **Figure 4.3**).

Visualizing Bias and Precision. From a visual standpoint, the bias and precision of estimated RMSSD values can be evaluated by the relative location of the deviation score boxplots for each model tested under varying conditions including the size of the data window used for imputation (1x, 2x, or 3x the size of the target window), the sampling rate (4 Hz, 8 Hz, or 12 Hz), and the data sampling rate (125 Hz or 250 Hz) (see **Figure 4.4**). Results of the two naïve imputation runs are presented in each window of **Figure 4.4** for comparison purposes. The values depicted in the plots are based on RMSSD values calculated from a section of data that included the portion targeted for imputation as well as 15 seconds of the surrounding data (split evenly before and after the imputation window). To calculate these scores using the imputed data, the peak detection algorithm used to extract IBIs was applied to a file in which imputed values replaced the original values. Mean and hotdeck replacement scores were calculated in a similar fashion, replacing the original interbeat intervals in the targeted section with the mean interbeat interval or a random value selected, with replacement, from the vector of interbeat

intervals. RMSSD values were then calculated from these new, updated vectors of interbeat interval values.

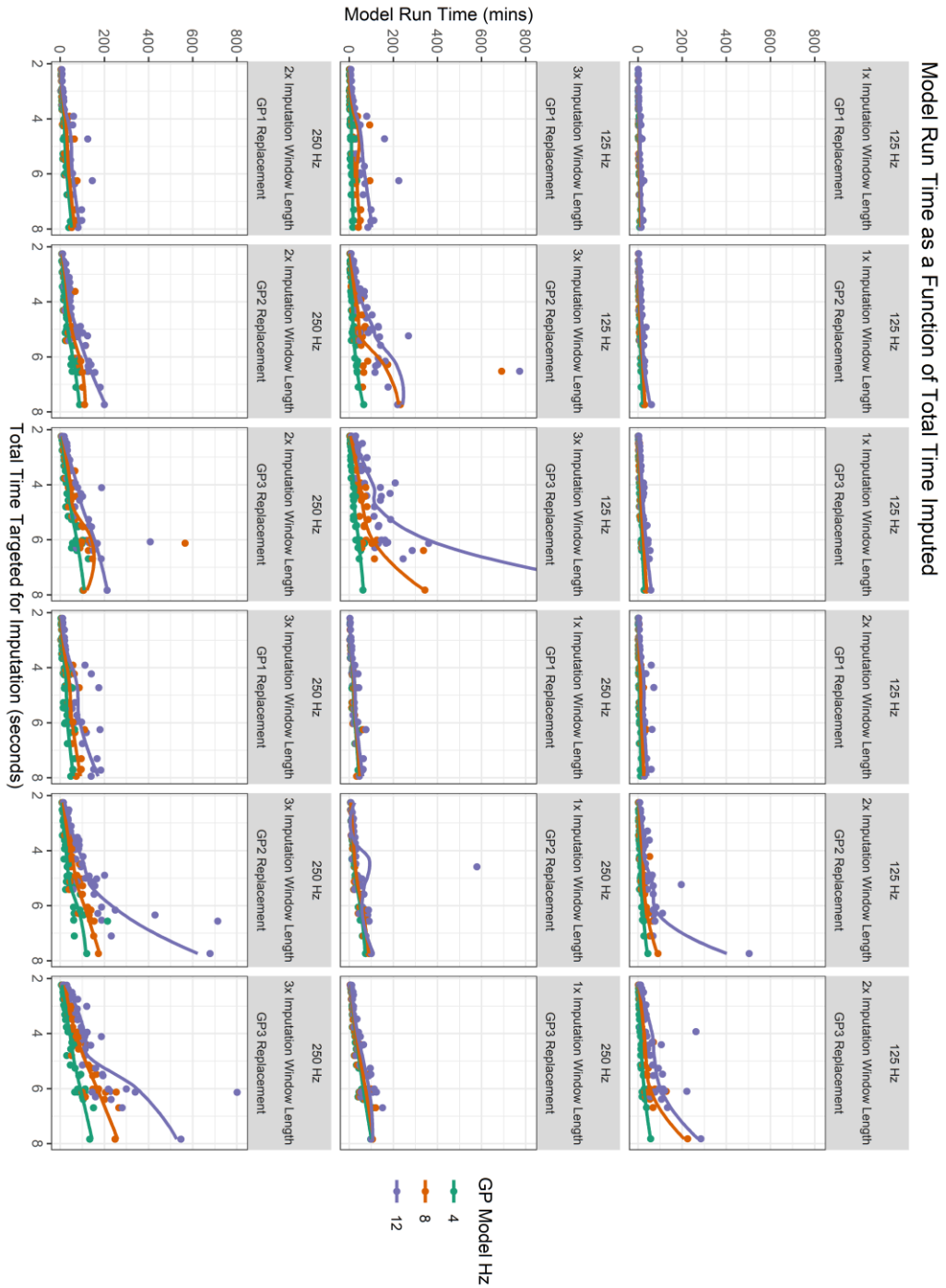


Figure 4.3 Model run time as a function of total time to be imputed

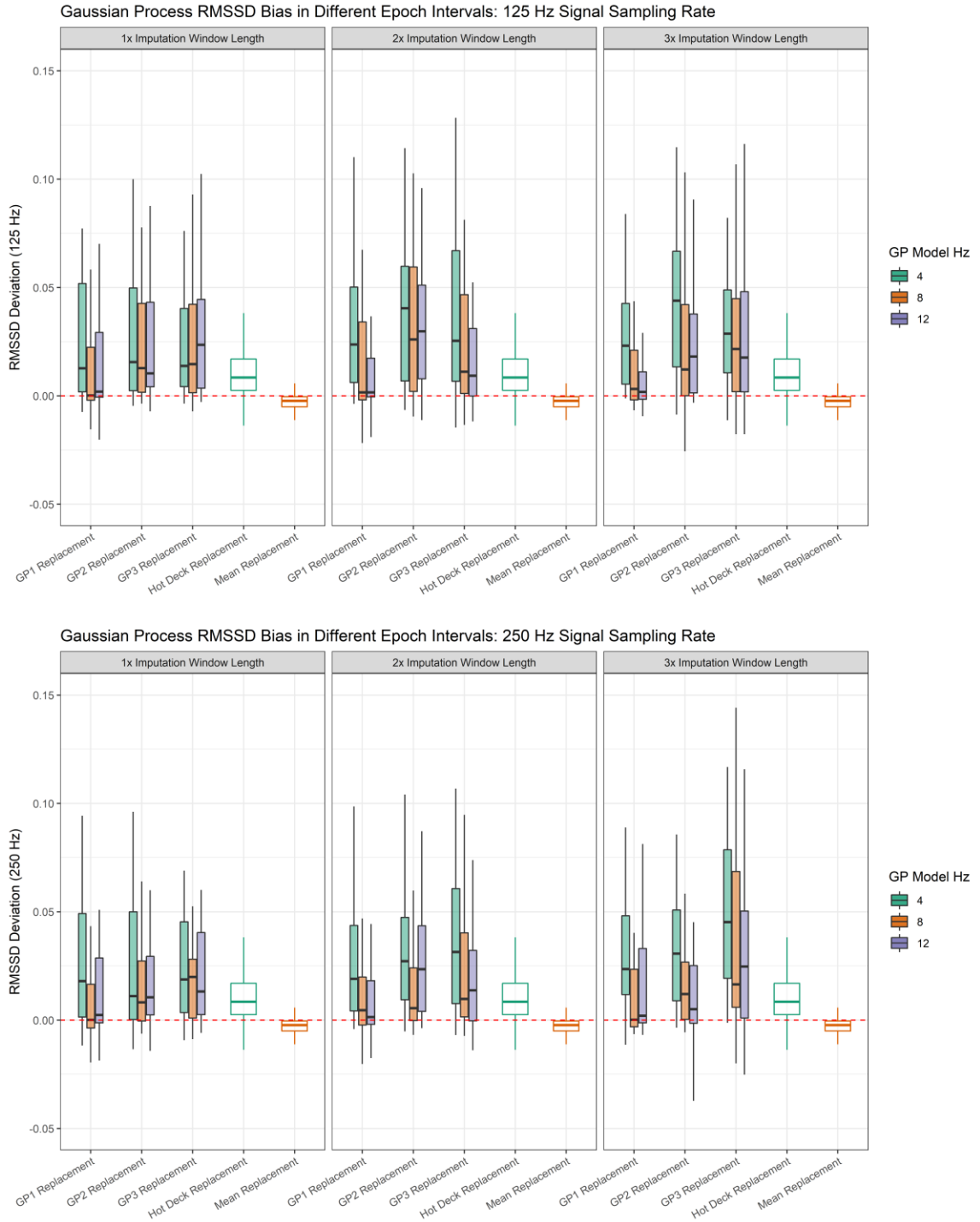


Figure 4.4 Distributions of imputation-derived RMSSD deviations from original values

As expected, the hotdeck replacement approach yielded estimates of RMSSD that were biased upwards (median bias = 0.0085) whereas a mean replacement strategy

resulted in estimates of RMSSD that were downwardly biased (median bias = -0.0023). Gaussian process models tended to exhibit positive bias and were relatively more spread out (i.e., less precise) than the two naïve strategies. There were some notable exceptions to these trends. For instance, when using the first Gaussian process model, which included a term for heart rate but no attempts to model variability in heart rate as a function of other physiological processes (e.g., breathing), imputing values based on model data sampled at 8 Hz from a window of time equal to the imputed section (i.e., 1x) yielded relatively accurate series of interbeat intervals overall with a median bias of 0.0002 and 0.00012 when the data were down-sampled to 125 Hz and 250 Hz respectively. It was also the case that under these conditions the precision of the imputed values increased slightly when the down-sampling rate went from 125 Hz ($\tau_{\text{IQR}} = 38.37$) to 250 Hz ($\tau_{\text{IQR}} = 45.50$)⁵. For comparison purposes the precision of the hotdeck replacement strategy was $\tau_{\text{IQR}} = 69.00$ and the precision of the mean replacement strategy was $\tau_{\text{IQR}} = 217.46$. Combined, these results suggest that at a down-sampled rate of 125 Hz and 250 Hz, the first Gaussian process model provided relatively less biased results than the two naïve imputation strategies, but the imputed values and resulting estimates of RMSSD derived from the imputed and surrounding data were less precise (i.e., more spread out).

⁵ Note that the interquartile range (IQR) is being used as the basis of precision in this case. Precision is often defined as the inverse of variance. In this case there are two reasons for using the IQR, or more accurately its inverse, as a precision metric. The first is that RMSSD is a variance statistic and is therefore lower-bounded by 0. Beyond a certain point, it is impossible for the model to result in a more negative deviation from the expected value. Conversely, the positively biased values have no upper limit. Secondly, the degree of mis-estimation of RMSSD in a positive direction, past an as yet undetermined value becomes qualitatively irrelevant. An imputed RMSSD value that misses the target mark by .15 is likely as useless from a practical standpoint as one that misses the mark by .25 or .75. Thus, a precision measure that relies on variance would provide a potentially misleading view of the degree of a given model's practical precision, even if it were providing a more accurate picture of its technical precision.

The first Gaussian process model also appeared to perform reasonably well when relying on data that were twice as long as the original section targeted for imputation, were sampled at 12 Hz for inclusion in the model, and when the overall signal was down-sampled to 250 Hz. The median bias under these conditions was positive (0.00209), but lower than the bias evident when using a hot deck replacement strategy and (slightly) smaller in absolute terms than the negative bias exhibited by a mean replacement strategy.

Based on the visual depiction of model results, the remaining conditions and models left much to be desired in terms of general model accuracy and precision. Opposite expectations, many of the conditions thought to possibly improve imputation performance did not result in appreciably better results than the statistically flawed, but easily implemented hotdeck and mean replacement strategies. Estimates of RMSSD derived from the predictions of the second and third Gaussian process models tended to consistently exhibit similar or greater levels of median bias and less precision as evidenced by the boxplots in **Figure 4.4** and the data summarized **Tables 4.1-4.6**.

Selecting only those models for which there was evidence of convergence on the heart rate posterior (i.e., $\hat{R} < 1.1$) and the first Gaussian process model still tended to provide the best prediction of the original RMSSD values (see **Figure 4.5**). Median bias did improve for the two more complex models when selecting models based on convergence, as did the precision of many of the RMSSD estimates. However, the smallest distribution

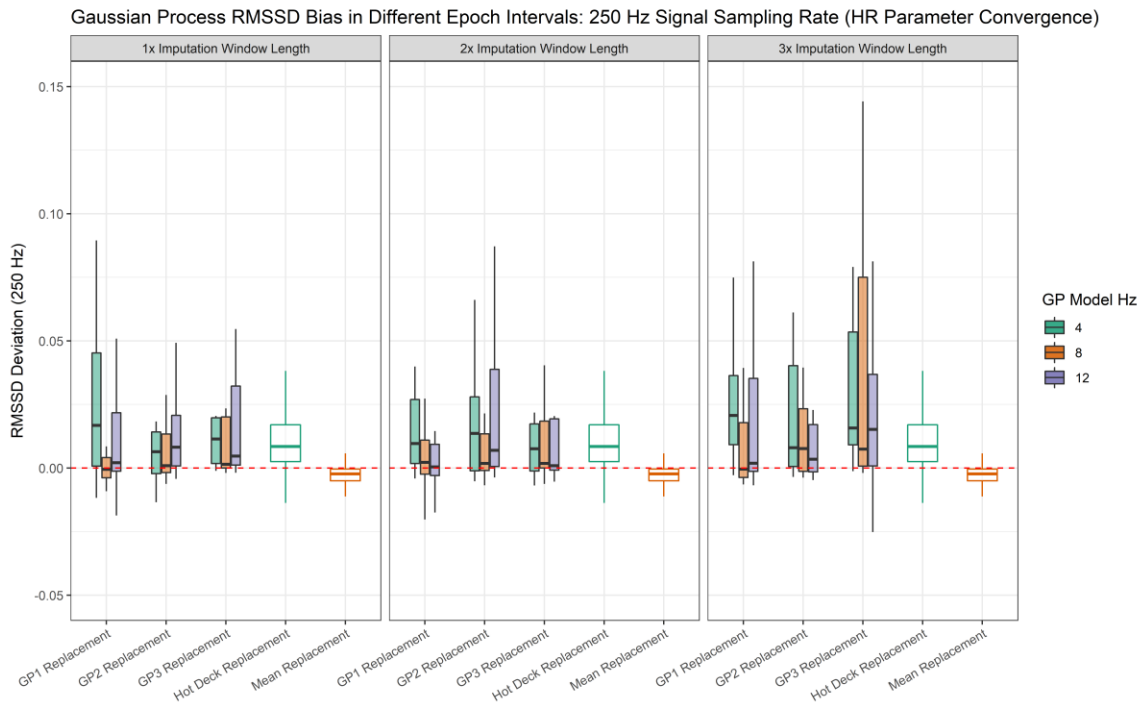
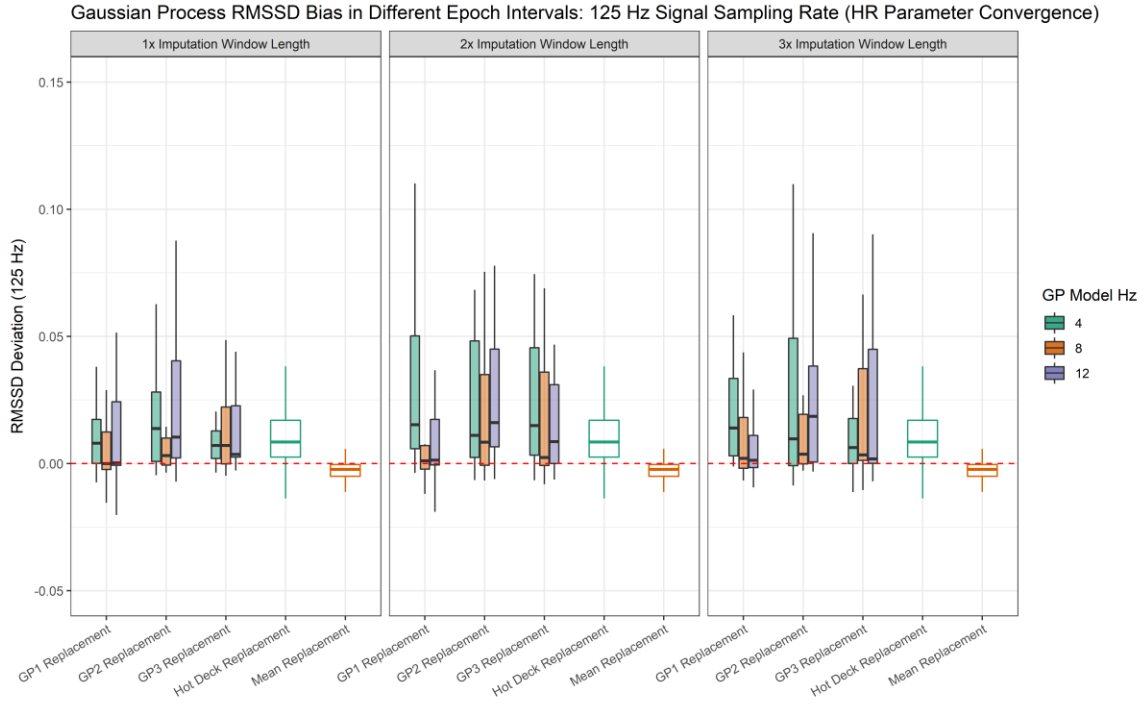


Figure 4.5 Distributions of imputation-derived RMSSD deviations from original values from models with adequate convergence

of RMSSD deviations centered at or about 0 were still those based on imputations from the first, simplest Gaussian process model. These results suggest that taking model convergence on the heart rate parameter into account is important. Moreover the findings indicate that model simplicity may be more predictive of successful imputation than the amount of data or the inclusion of more complex terms to explain different sources of heart rate variability.

Study 2 Results

Editing Summary. One useful feature of IBI VizEdit is that the program outputs simple summaries of the number and proportion of data points in the final files that represent manually adjusted and edited values. These outputs can identify sections of data that typically require editing in a study, allowing for transparent reporting. As displayed in **Figure 4.6**, the Clown and Introduction tasks were the most heavily edited as measured by the proportion of final data points that represented edited interbeat interval values (i.e., not in the original vector returned by the peak detection algorithm). This result is not surprising as these tasks were the most distressing for the children to participate in and tended to generate the most movement, even while seated in their chairs.

Intra-Editor Reliability. As a first test of the ability of *IBI VizEdit* to yield reliable results, the two primary editors who worked on Study 2 re-edited a subset of tasks from the files they were assigned. These tasks were randomly selected from all the cases edited by each of the two editors, provided that no part of the task was deemed uneditable. Editor 1 re-edited a total of 37 tasks from their assigned files and Editor 2 re-edited a total of 38 tasks from their files. To assess performance across programs,

differences scores were created by subtracting the *Cardio Edit* RSA values from the *IBI VizEdit* RSA values. On average, RSA score deviations between the two programs were 0.03 ($SD =$

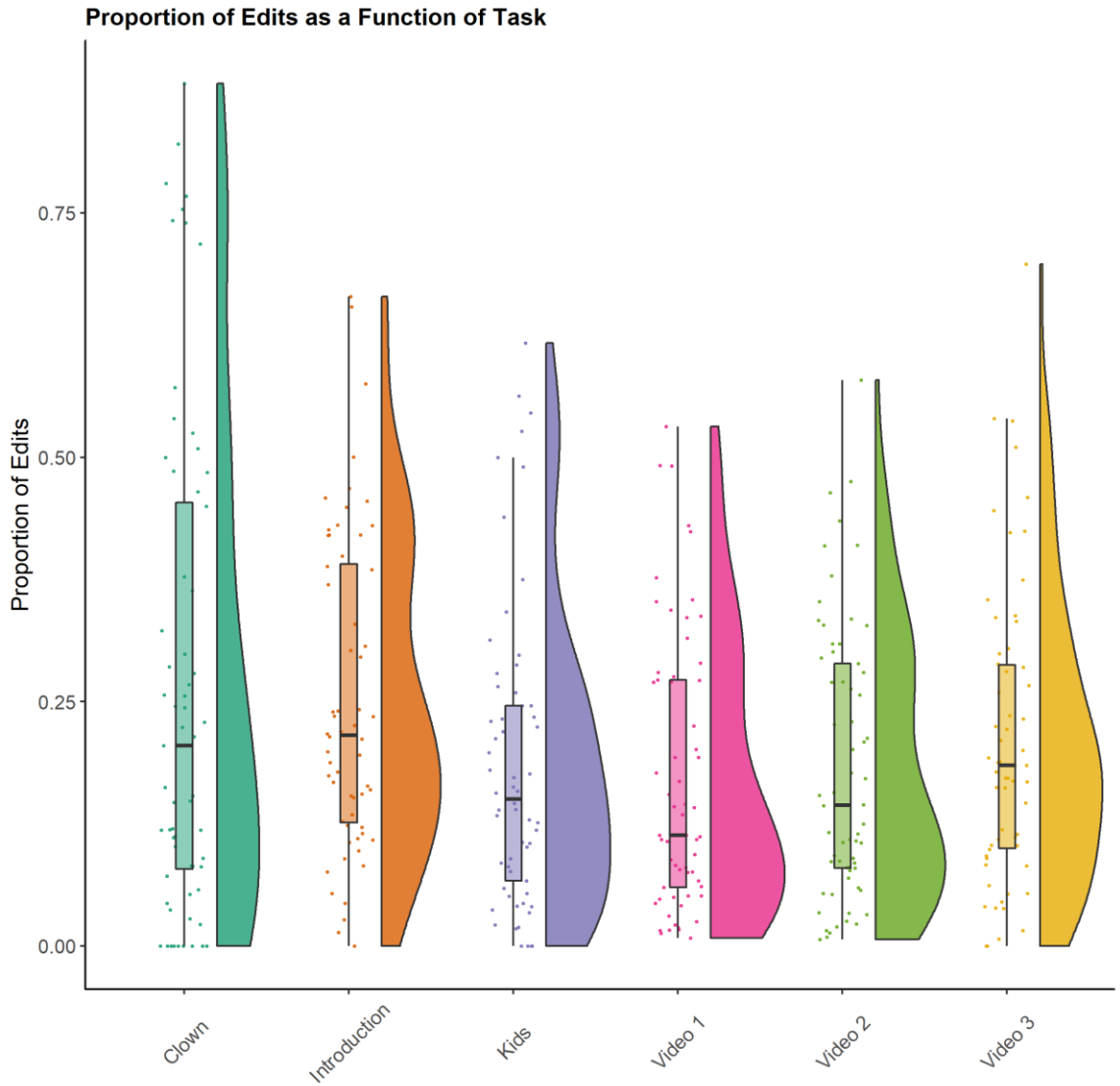


Figure 4.6 Distributions of the proportion of edits required by task

0.11) for Editor 1 and 0.01 ($SD = 0.08$) for Editor 2. As seen in **Figure 4.7**, when Editor 1 used *IBI VizEdit* to process heart rate files, their obtained RSA values tended to be slightly higher than when using *Cardio Edit*, though the average difference score was not

significantly greater than 0, $t(36) = 1.39, p = .174$. An alternative to examining average differences scores across programs is to determine the number of times the difference in RSA scores fell outside a $\pm .05$ range, the threshold used by the Brain-Body Center for establishing reliability using *Cardio Edit*. Applying this threshold 18 tasks (48.65%) fell outside the target window for Editor 1 and 11 tasks (28.95%) fell outside the window for Editor 2. A slightly more relaxed threshold of $\pm .10$ revealed a somewhat more encouraging pattern from a cross-program consistency standpoint. A total of 11 tasks (29.73%) for Editor 1 and 5 (13.16%) for Editor 2 fell outside this slightly expanded range (see **Figure 4.7**). Finally, scores derived from both programs, despite the slight difference in absolute values, were highly correlated, $r(75) = .99, p < .001$. Combined, these results suggest that trained editors can use *IBI VizEdit* to obtain scores for RSA that are highly similar in a relative, not absolute, sense to those obtained when using *Cardio Edit*.

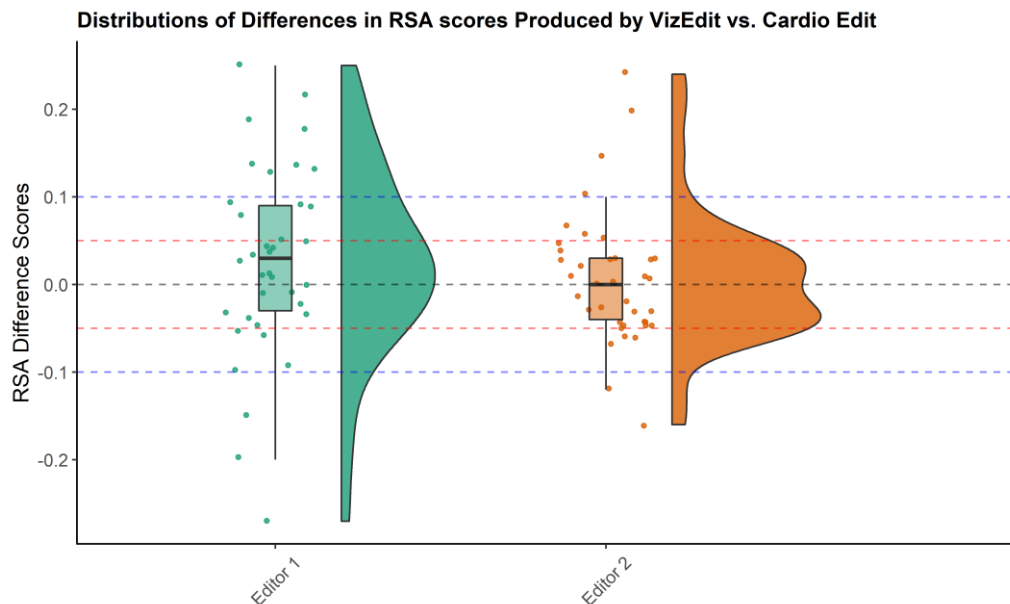


Figure 4.7 RSA difference scores for primary editors: Comparing editors' RSA values from IBI VizEdit and Cardio Edit

Inter-Editor Reliability. As a second check on the program outputs, a third editor re-edited a total of 17 complete heart rate files randomly taken from Study 2 participants using *IBI VizEdit*. These cases were drawn from files edited initially by both primary editors, if there were no uneditable segments identified within a given case. The third editor edited every task within a given file and calculated RSA scores using *Cardio Batch*. Examining difference scores revealed that, on average, the third editor was within +/- .05 of the RSA values derived from Editor 1 100% of the time (60 out of the 60 co-edited tasks). Except for one large deviation (omitted from **Figure 4.8**) in which the RSA values for Editor 3 and Editor 2 (the other primary editor) were separated by .49, Editor 3 and Editor 2 produced RSA scores within the +/- .05 threshold most of the time (41 out of the 42 tasks that were co-edited). Not surprisingly, given the degree to which inter-editor scores were largely in agreement on an absolute basis, Editor 1's and Editor 3's scores for the same tasks were correlated at $r(58) = .9997, p < .001$, and Editor 2's and Editor 3's scores were correlated at $r(40) = .9965, p < .001$. Overall, these data suggest that different editors who completed the same *IBI VizEdit* training protocol were able to use the program to generate highly similar results, not only in a relative sense, as evidenced by correlations approaching 1, but also in an absolute sense with nearly all differences in RSA scores between editors falling between +/- .05.

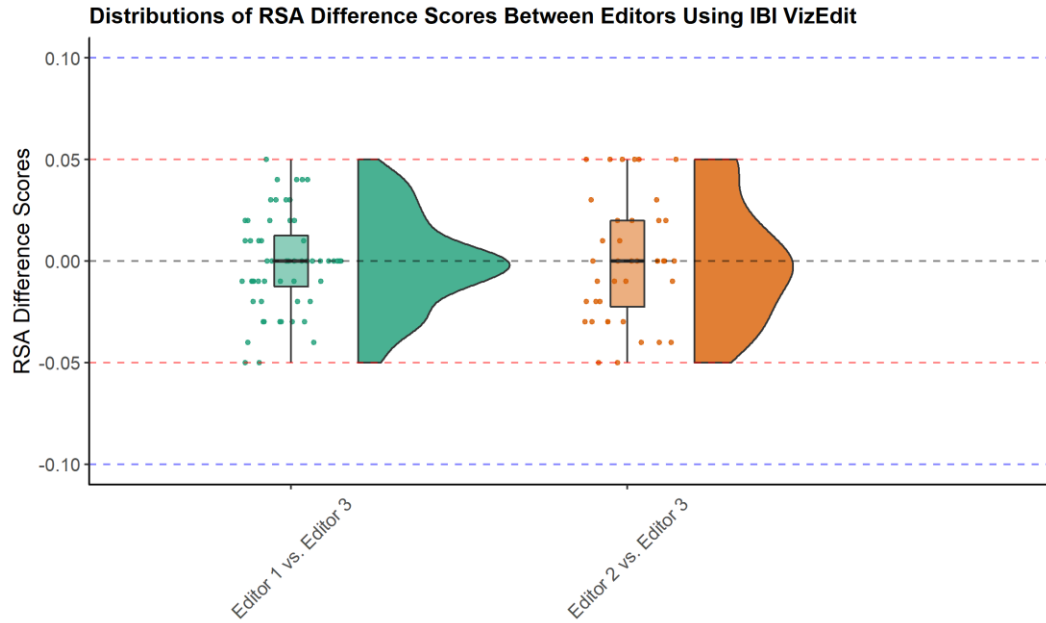


Figure 4.8 RSA difference scores for primary editors: Comparing RSA values from a third editor using IBI VizEdit

Task Analyses. A series of Bayesian multilevel models were created to determine whether estimates of heart rate variability (RSA and RMSSD) and heart period during a baseline laboratory assessment varied as expected. As a brief reminder, during periods of engaged attention and relaxation heart rate variability and heart period are expected to be higher relative to moments characterized by threat or distress. In the laboratory assessment, the Video and Kids tasks were expected to lead to engaged and attentive states. Interacting with the research assistant dressed as a clown and being asked to record a video introduction were both expected to be distressing, leading to a reduction in heart rate variability and heart period as parasympathetic influence was withdrawn from affecting cardiac pacing. Data were missing in the present set of physiological variables when the signal was of such low quality that at least 40% of a task or more was deemed uneditable. Additionally, in the case of RSA, if the task was too brief to create at least two 15-second epochs to average together or if any portion of the task was uneditable,

RSA values were not calculated. As summarized in **Figure 4.9**, these constraints resulted in 49.3% of the cases analyzed having completely usable data through all six tasks for the purposes of calculating RSA, with 13.3% containing insufficient data for all tasks. Several other missing data patterns emerged and are summarized in **Figure 4.9**. As might be expected, the short duration of the Clown task resulted in the largest proportion of cases with insufficient data to estimate RSA values, even when epoch lengths were set to 15 seconds.

Because *IBI VizEdit* is able to ignore uneditable segments of data when creating summary scores for RMSSD and heart period, these variables were more complete in the present data set. So long as at least 60% of a task was deemed editable *IBI VizEdit* returned estimated values for these two measures of cardiac activity. The missingness patterns for these variables are displayed in **Figure 4.10**. Overall, a full two-thirds (66.7%) of participants included in Study 2 had sufficient data to obtain RMSSD and heart period scores across all six laboratory tasks, with 12.0% of the sample having insufficient editable data for all tasks. Again, several other missing data patterns emerged and are presented in **Figure 4.10**.

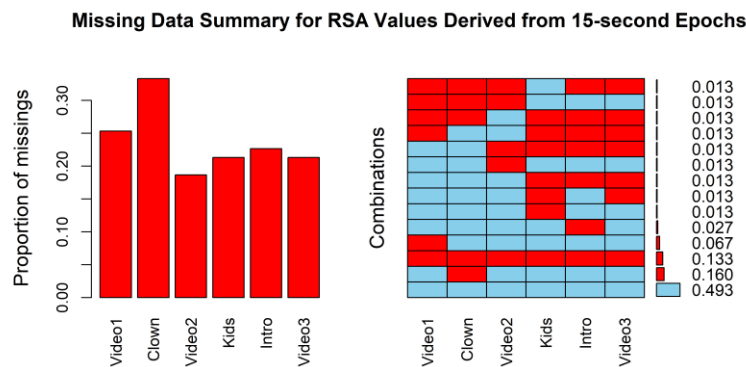


Figure 4.9 Visualization of missing data for RSA values

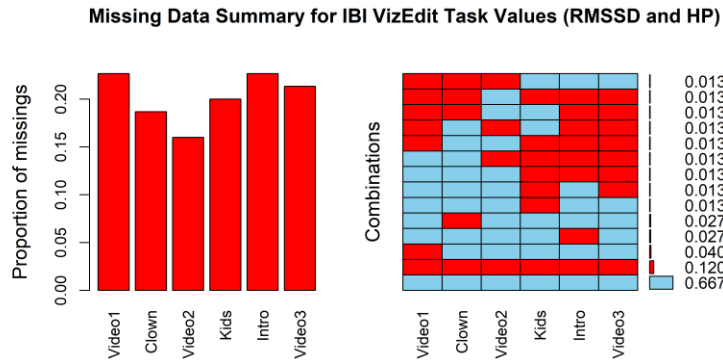


Figure 4.10 Visualization of missing data for IBI VizEdit-Generated values

Multiple imputation was used to address statistical issues arising from missing data in the present set of variables. Gender, age (in months), task, and baseline BI scores were included in the imputation model as auxiliary, completely observed variables. Use of the `mitml` package provided a straightforward means of incorporating the hierarchical structure of the data in the imputation model, while allowing completely observed auxiliary variables to serve as predictors of the missing values (Grund, Robitzsch, & Luedtke, 2018). Based on recommendations made by (Graham, Olchowski, & Gilreath, 2007), a total of 20 data sets with imputed values were created and included in the multilevel analyses. Following suggestions for the setting of priors in Stan (Stan Development Team, 2017), model parameters were provided weakly informative priors to ensure posterior estimates remained within the range of plausible values (as opposed to setting flat or uninformative priors as has been common practice among some Bayesian analysts in the past). Posterior parameter distributions derived from each of the twenty imputed data sets were integrated using the `brm_multiple()` function available in the `brms` package (Bürkner, 2017).

Respiratory Sinus Arrhythmia. Using 95% credibility intervals to test the hypothesis that differences in RSA between tasks was unlikely to be zero, the model revealed that children’s RSA was lower during the Clown and Introduction tasks compared with all other tasks. These differences remained after controlling for child gender and age as well as the proportion of edits required in each task (see **Table 4.7**). There were no differences in RSA scores across the other four tasks (i.e., the repeated viewing of the Video or the Kids social learning task), suggesting that children’s RSA scores were similar across these four conditions.

Table 4.7

Table 4.7 Respiratory Sinus Arrhythmia Laboratory Task Model Results

	Estimate	SE	Q2.5	Q97.5
Intercept	6.401	0.154	6.098	6.704
Clown vs. Video 1	-0.280	0.081	-0.439	-0.122
Video 2 vs. Video 1	0.028	0.081	-0.131	0.186
Kids vs. Video 1	0.030	0.080	-0.126	0.186
Introduction vs. Video 1	-0.300	0.081	-0.462	-0.143
Video 3 vs. Video 1	-0.026	0.082	-0.186	0.135

Note. The first video segment (Video 1) served as the reference condition for the model. Covariates (child age, gender, and proportion of task edits) not displayed. *SE* = standard error of the posterior distribution; Q2.5 = lower boundary of a 95% credibility interval (i.e., 2.5%); Q97.5 = upper boundary of 95% credibility interval (i.e., 97.5%). Bolded rows represent estimates for which 95% credibility intervals do not contain 0 (excluding the intercept).

RMSSD. As was the case with RSA, 95% credibility intervals were used to evaluate the hypothesis that RMSSD values differed as a function of each task during the assessment. Opposite expectations and counter to the results examining RSA variation, there were no differences in heartbeat RMSSD scores across the various conditions during the laboratory assessment (see **Table 4.8**). These null results held when controlling for child gender, age, and proportion of edits.

Table 4.8

Table 4.8 Root Mean Successive Difference Laboratory Task Model

	Estimate	SE	Q2.5	Q97.5
Intercept	0.044	0.006	0.030	0.055
Clown vs. Video 1	-0.003	0.005	-0.012	0.009
Video 2 vs. Video 1	0.004	0.005	-0.006	0.014
Kids vs. Video 1	0.004	0.005	-0.005	0.017
Introduction vs. Video 1	-0.001	0.006	-0.011	0.014
Video 3 vs. Video 1	0.002	0.005	-0.007	0.014

Note. The first video segment (Video 1) served as the reference condition for the model. Covariates (child age, gender, and proportion of task edits) not displayed. *SE* = standard error of the posterior distribution; Q2.5 = lower boundary of a 95% credibility interval (i.e., 2.5%); Q97.5 = upper boundary of 95% credibility interval (i.e., 97.5%). Bolded rows represent estimates for which 95% credibility intervals do not contain 0 (excluding the intercept).

Heart Period. Finally, using the same hypothesis testing procedures as described above, the model revealed that heart period was lower during the Clown and the Introduction task (i.e., the child’s heart was beating faster) compared with all other conditions during the assessment. These differences held after controlling for child gender, age, and the proportion of edits required in each task (see **Table 4.9**). As with the RSA models, there were no differences in heart period across the other tasks.

Table 4.9

Table 4.9 Heart Period Laboratory Task Model Results

	Estimate	SE	Q2.5	Q97.5
Intercept	0.617	0.013	0.592	0.641
Clown vs. Video 1	-0.024	0.007	-0.038	-0.012
Video 2 vs. Video 1	-0.004	0.007	-0.017	0.009
Kids vs. Video 1	0.002	0.007	-0.013	0.016
Introduction vs. Video 1	-0.035	0.007	-0.048	-0.022
Video 3 vs. Video 1	-0.005	0.007	-0.019	0.009

Note. The first video segment (Video 1) served as the reference condition for the model. Covariates (child age, gender, and proportion of task edits) not displayed. *SE* = standard error of the posterior distribution; Q2.5 = lower boundary of a 95% credibility interval (i.e., 2.5%); Q97.5 = upper boundary of 95% credibility interval (i.e., 97.5%).

Tonic RSA and Play Behaviors

RSA scores averaged across the three video conditions were used as a measure of children’s tonic RSA, which is thought to be related to individuals’ self-regulatory capacity (Beauchaine, 2015). A Bayesian binomial regression provided the modeling framework for examining the association between tonic RSA scores and the probability of engaging in various play behaviors. This modeling approach provided a natural means of incorporating variation in the total amount of time each child was observed, as cases with more total observations contributed more weight to the final fixed effects using this approach. In all models, gender and child age were included as covariates. Missing data were addressed using a similar imputation strategy as described for the RSA, RMSSD, and heart period analyses. The main difference was that, for these models, the proportions of play behaviors were included as an additional set of variables in the imputation model.

As can be seen in **Tables 4.10-4.15**, tonic RSA was unrelated to the likelihood children engaged in any of the coded behaviors. All parameter estimates representing the association between average RSA during the videos and each play behavior resulted in 95% credibility intervals that contained 0.

Table 4.10

Table 4.10 Predicting Relative Frequency of Reticent Behavior during Free Play Using RSA

	Estimate	SE	Q2.5	Q97.5
Tonic RSA	-0.13	0.18	-0.48	0.23
Video 1 RSA	-0.14	0.17	-0.48	0.20
Δ RSA (Clown)	0.25	0.22	-0.17	0.69
Δ RSA (Kids)	0.08	0.40	-0.70	0.88
Δ RSA (Introduction)	0.21	0.28	-0.35	0.75

Note. Models controlled for child age and gender (not displayed). Only RSA coefficients displayed above. *SE* = standard error of the posterior distribution; Q2.5 = lower boundary of a 95% credibility interval (i.e., 2.5%); Q97.5 = upper boundary of 95% credibility interval (i.e., 97.5%).

Table 4.11

Table 4.11 Predicting Relative Frequency of Functional-Dramatic Behavior during Free Play Using RSA

	Estimate	SE	Q2.5	Q97.5
Tonic RSA	0.05	0.30	-0.53	0.64
Video 1 RSA	0.21	0.29	-0.35	0.78
Δ RSA (Clown)	-0.56	0.36	-1.28	0.15
Δ RSA (Kids)	0.12	0.58	-1.03	1.24
Δ RSA (Introduction)	-0.11	0.42	-0.94	0.72

Note. Models controlled for child age and gender (not displayed). Only RSA coefficients displayed above. *SE* = standard error of the posterior distribution; Q2.5 = lower boundary of a 95% credibility interval (i.e., 2.5%); Q97.5 = upper boundary of 95% credibility interval (i.e., 97.5%).

It is possible that averaging RSA value across all three video segments of the laboratory assessment artificially drove down estimates of tonic RSA to some degree. The second and third displays of the video followed putatively distressing tasks (i.e., social interactions with an RA dressed as a clown and recording a video introduction) which, as is described above, led to lower RSA values. As the parasympathetic inputs to the heart came back online (i.e., vagal withdrawal was reversed), there would naturally be a period in which lower RSA values lingered into the video conditions. To rule out the latency to return to baseline as having an undue effect on estimates of tonic RSA, the same binomial regression models were run a second time, with average RSA during the first video as the measure of tonic RSA. As can be seen in **Tables 4.10-4.15**, the results remained the same and RSA was not related to the likelihood of engaging in any of the coded play behaviors.

Table 4.12

Table 4.12 Predicting Relative Frequency of Solitary-Constructive Behavior during Free Play Using RSA

	Estimate	SE	Q2.5	Q97.5
Tonic RSA	-0.23	0.26	-0.75	0.28
Video 1 RSA	-0.22	0.25	-0.71	0.27
Δ RSA (Clown)	0.06	0.33	-0.60	0.70
Δ RSA (Kids)	-0.02	0.51	-1.02	0.99
Δ RSA (Introduction)	0.36	0.36	-0.35	1.08

Note. Models controlled for child age and gender (not displayed). Only RSA coefficients displayed above. *SE* = standard error of the posterior distribution; Q2.5 = lower boundary of a 95% credibility interval (i.e., 2.5%); Q97.5 = upper boundary of 95% credibility interval (i.e., 97.5%).

Table 4.13

Table 4.13 Predicting Relative Frequency of Solitary Behavior Composite during Free Play Using RSA

	Estimate	SE	Q2.5	Q97.5
Tonic RSA	-0.09	0.18	-0.44	0.26
Video 1 RSA	-0.07	0.17	-0.41	0.28
Δ RSA (Clown)	0.02	0.22	-0.42	0.45
Δ RSA (Kids)	0.04	0.37	-0.67	0.77
Δ RSA (Introduction)	0.14	0.27	-0.38	0.67

Note. Models controlled for child age and gender (not displayed). The solitary behavior composite represented a combination of all three forms of solitary behavior: reticent, functional-dramatic, and constructive play. Only RSA coefficients displayed above. *SE* = standard error of the posterior distribution; Q2.5 = lower boundary of a 95% credibility interval (i.e., 2.5%); Q97.5 = upper boundary of 95% credibility interval (i.e., 97.5%).

Table 4.14

Table 4.14 Predicting Relative Frequency of Parallel Play during Free Play Using RSA

	Estimate	SE	Q2.5	Q97.5
Tonic RSA	-0.21	0.20	-0.59	0.18
Video 1 RSA	-0.15	0.19	-0.52	0.23
Δ RSA (Clown)	-0.38	0.25	-0.85	0.12
Δ RSA (Kids)	-0.21	0.43	-1.05	0.61
Δ RSA (Introduction)	-0.20	0.31	-0.79	0.40

Note. Models controlled for child age and gender (not displayed). Only RSA coefficients displayed above. *SE* = standard error of the posterior distribution; Q2.5 = lower boundary of a 95% credibility interval (i.e., 2.5%); Q97.5 = upper boundary of 95% credibility interval (i.e., 97.5%).

Table 4.15

Table 4.15 Predicting Relative Frequency of Group Play during Free Play Using RSA

	Estimate	SE	Q2.5	Q97.5
Tonic RSA	0.27	0.25	-0.21	0.76
Video 1 RSA	0.25	0.23	-0.21	0.72
Δ RSA (Clown)	0.33	0.29	-0.25	0.89
Δ RSA (Kids)	0.05	0.47	-0.87	0.99
Δ RSA (Introduction)	-0.38	0.34	-1.06	0.28

Note. Models controlled for child age and gender (not displayed). Only RSA coefficients displayed above. *SE* = standard error of the posterior distribution; Q2.5 = lower boundary of a 95% credibility interval (i.e., 2.5%); Q97.5 = upper boundary of 95% credibility interval (i.e., 97.5%).

Dynamic RSA and Play Behaviors

Next, a series of difference scores were created to represent dynamic RSA in response to the two socially stressful tasks (Clown and Introduction) and the social learning task (Kids). To create these scores, RSA during each of the three conditions was subtracted from average RSA across all three videos. Bayesian binomial regressions again provided the modeling framework to address the hypothesis that dynamic RSA in response to each condition would be related to variation in play behaviors in vivo.

The results for dynamic measures of RSA were similar to those using Tonic measures of RSA to predict play behaviors in the classroom context. RSA reactivity to

the laboratory tasks failed to predict children’s play in a real-world context (see **Tables 4.10-4.15**).

Tonic and Dynamic Heart Period and Play Behaviors

Mirroring the analyses involving RSA, tonic heart period across all three videos and changes in heart period relative to average heart period during the videos were used to predict the proportion of time children spent engaged in various behaviors during classroom free play. Also mirroring the RSA models, tonic heart period as well as changes in heart period failed to predict the likelihood of engaging in any one of the play behaviors recorded by trained coders, see **Tables 4.16-4.21**.

Table 4.16

Table 4.16 Predicting Relative Frequency of Reticence during Free Play Using Heart Period

	Estimate	SE	Q2.5	Q97.5
Tonic HP	0.84	1.56	-2.20	3.88
Video 1 HP	0.57	1.27	-1.94	3.10
Δ HP (Clown)	-0.61	1.58	-3.76	2.49
Δ HP (Kids)	-0.38	1.63	-3.47	2.77
Δ HP (Introduction)	-0.25	1.65	-3.42	3.10

Note. Models controlled for child age and gender (not displayed). Only RSA coefficients displayed above. *SE* = standard error of the posterior distribution; Q2.5 = lower boundary of a 95% credibility interval (i.e., 2.5%); Q97.5 = upper boundary of 95% credibility interval (i.e., 97.5%).

Table 4.17

Table 4.17 Predicting Relative Frequency of Functional-Dramatic Behavior during Free Play Using Heart Period

	Estimate	SE	Q2.5	Q97.5
Tonic HP	0.01	1.79	-3.48	3.51
Video 1 HP	0.31	1.62	-2.94	3.40
Δ HP (Clown)	0.48	1.75	-2.94	3.91
Δ HP (Kids)	0.24	1.95	-3.51	4.02
Δ HP (Introduction)	0.26	1.66	-2.96	3.57

Note. Models controlled for child age and gender (not displayed). Only RSA coefficients displayed above. *SE* = standard error of the posterior distribution; Q2.5 = lower boundary of a 95% credibility interval (i.e., 2.5%); Q97.5 = upper boundary of 95% credibility interval (i.e., 97.5%).

Table 4.18

Table 4.18 Predicting Relative Frequency of Constructive-Exploratory Behavior during Free Play Using Heart Period

	Estimate	SE	Q2.5	Q97.5
Tonic HP	-1.43	1.76	-4.87	2.04
Video 1 HP	-0.46	1.69	-3.75	2.82
Δ HP (Clown)	0.48	1.75	-2.94	3.91
Δ HP (Kids)	0.25	1.55	-2.83	3.29
Δ HP (Introduction)	-0.12	1.57	-3.20	2.95

Note. Models controlled for child age and gender (not displayed). Only RSA coefficients displayed above. *SE* = standard error of the posterior distribution; Q2.5 = lower boundary of a 95% credibility interval (i.e., 2.5%); Q97.5 = upper boundary of 95% credibility interval (i.e., 97.5%).

Table 4.19

Table 4.19 Predicting Relative Frequency of Solitary Behavior during Free Play Using Heart Period

	Estimate	SE	Q2.5	Q97.5
Tonic HP	0.14	1.55	-2.87	3.20
Video 1 HP	0.31	1.30	-2.36	2.70
Δ HP (Clown)	-0.51	1.42	-3.36	2.25
Δ HP (Kids)	-0.33	1.32	-2.86	2.32
Δ HP (Introduction)	-0.36	1.55	-3.35	2.77

Note. Models controlled for child age and gender (not displayed). Only RSA coefficients displayed above. The solitary behavior composite represented a combination of all three forms of solitary behavior: reticent, functional-dramatic, and constructive play. *SE* = standard error of the posterior distribution; Q2.5 = lower boundary of a 95% credibility interval (i.e., 2.5%); Q97.5 = upper boundary of 95% credibility interval (i.e., 97.5%).

Table 4.20

Table 4.20 Predicting Relative Frequency of Parallel Play during Free Play Using Heart Period

	Estimate	SE	Q2.5	Q97.5
Tonic HP	-0.89	1.60	-3.99	2.28
Video 1 HP	-0.40	1.25	-2.86	2.10
Δ HP (Clown)	0.25	1.40	-2.52	3.02
Δ HP (Kids)	0.24	1.66	-3.07	3.40
Δ HP (Introduction)	0.27	1.75	-3.49	3.41

Note. Models controlled for child age and gender (not displayed). Only RSA coefficients displayed above. *SE* = standard error of the posterior distribution; Q2.5 = lower boundary of a 95% credibility interval (i.e., 2.5%); Q97.5 = upper boundary of 95% credibility interval (i.e., 97.5%).

Table 4.21

Table 4.21 Predicting Relative Frequency of Group Play during Free Play Using Heart Period

	Estimate	SE	Q2.5	Q97.5
Tonic HP	0.38	1.71	-2.98	3.75
Video 1 HP	0.04	1.38	-2.57	2.84
Δ HP (Clown)	0.48	1.70	-2.81	3.79
Δ HP (Kids)	0.13	1.44	-2.75	2.98
Δ HP (Introduction)	0.18	1.46	-2.70	3.05

Note. Models controlled for child age and gender (not displayed). Only RSA coefficients displayed above. *SE* = standard error of the posterior distribution; Q2.5 = lower boundary of a 95% credibility interval (i.e., 2.5%); Q97.5 = upper boundary of 95% credibility interval (i.e., 97.5%).

CHAPTER 5

Study 1: Is there Evidence to Support the Use of Gaussian Process Models?

The results from the Study 1 indicate that there may be value in applying Gaussian process models to the task of imputing PPG data when the signal is lost due to artifacts or other technical issues (e.g., power disruption, hardware malfunction, etc.). The findings from Study 1 also clearly indicate that whatever *potential* utility Gaussian process models may hold in addressing missing values in the case of PPG data that utility has yet to be fully realized. **Figures 5.1-5.3**, demonstrate in a rather straightforward manner, the good, the bad, and the ugly results of imputed values achieved via Gaussian process models in the present research. While not perfectly aligned with the original values, **Figure 5.1** is an example of a relatively successful imputation run, with the peak locations of the imputed PPG signal roughly approximating the peak locations of the original signal. By comparison, **Figure 5.2** represents an instance in which the imputation model was able to produce a series of imputed values with an appropriate quasi-periodic waveform but obviously out of sync in its timing with the original values. More troubling though are imputation runs that fail to impose anything close to the structure of the individual's PPG waveform on the targeted section of data (see **Figure 5.3**).

Taken together, these three exemplars of different Gaussian Process model imputation runs simultaneously breathe life into the idea that this approach can be successfully employed and demonstrate that the optimal implementation of the imputation framework has yet to be achieved. Certainly, a degree of error is going to be unavoidable in any attempt to impute missing or corrupted data, and results such as those

in **Figure 5.2** might be acceptable if the majority of times the imputation model yielded something closer to the predicted values on display in **Figure 5.1**.

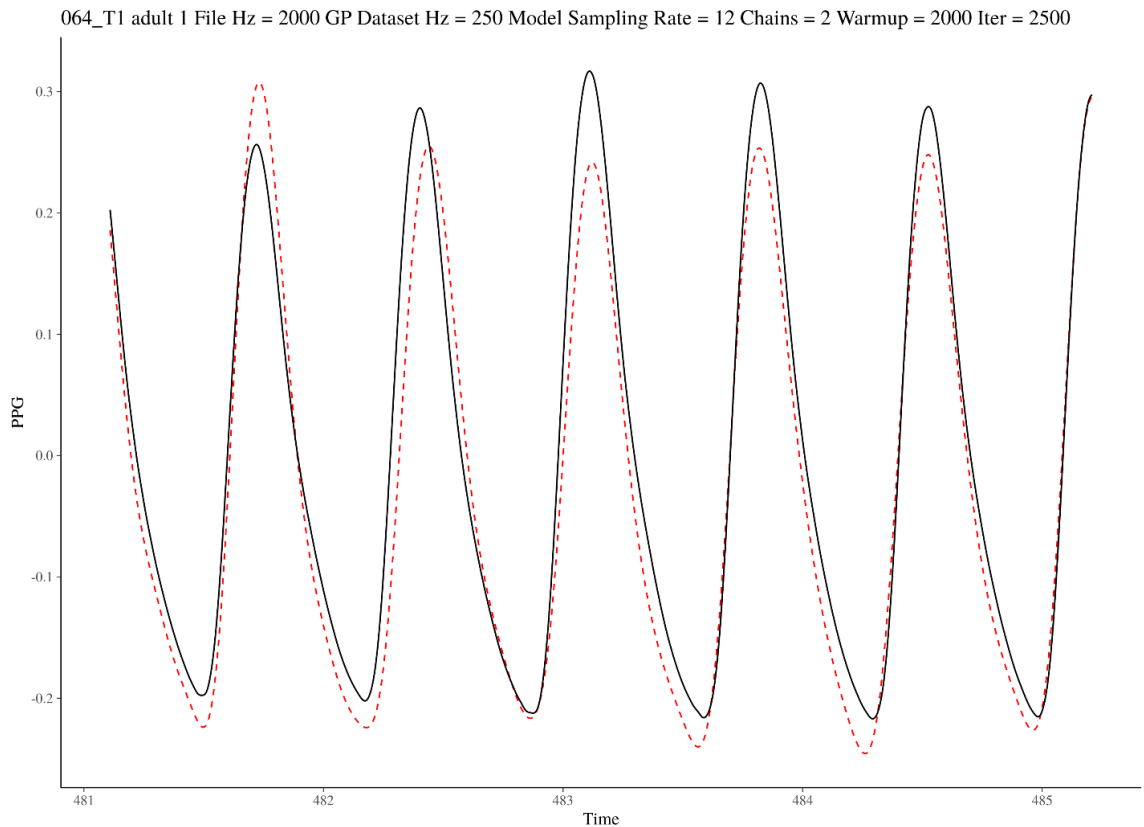


Figure 5.1 Relatively successful imputation (in red) compared with original values (in black)

Another, potentially related, issue that emerged in sifting through these results is that some of the models took such a long time to run (> 6 hours) that it is hard to imagine that Gaussian process imputation models, in their current form, represent a practical solution to the problems presented by missing or disrupted PPG data. Waiting 6 hours to, *maybe*, recover 8 seconds of data is an unacceptable tradeoff from a practical standpoint.

This may seem a relatively dour appraisal of the results, but it is a realistic one. Still, these findings offer something of a cornerstone around which to build and refine the current imputation approach. Clearly, the models work well under certain circumstances,

often only taking 5 to 10 minutes to complete a relatively accurate estimation of missing values. There is even evidence that the simpler parameterization of the Gaussian process imputation model that does not attempt to incorporate additional sources of heart rate variability into the model may in fact provide the best path forward.

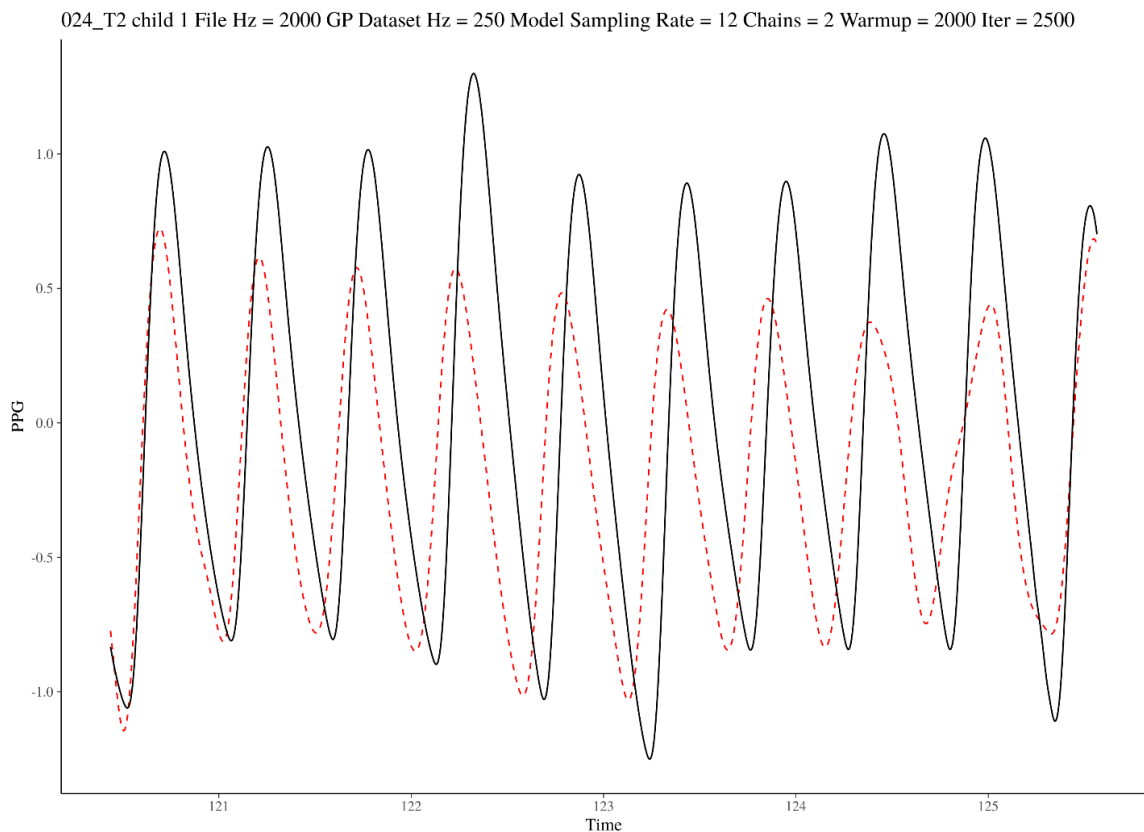


Figure 5.2 Example of imputed values (in red) that are offset from original values (in black)

As this line of research progresses, the focus should be placed on better understanding the properties of the data and the model that maximize the performance of the combined Gaussian processes in arriving at reasonably accurate estimates of missing values. Based on the current study the following aspects of the implementation approach are among the first to warrant scrutiny.

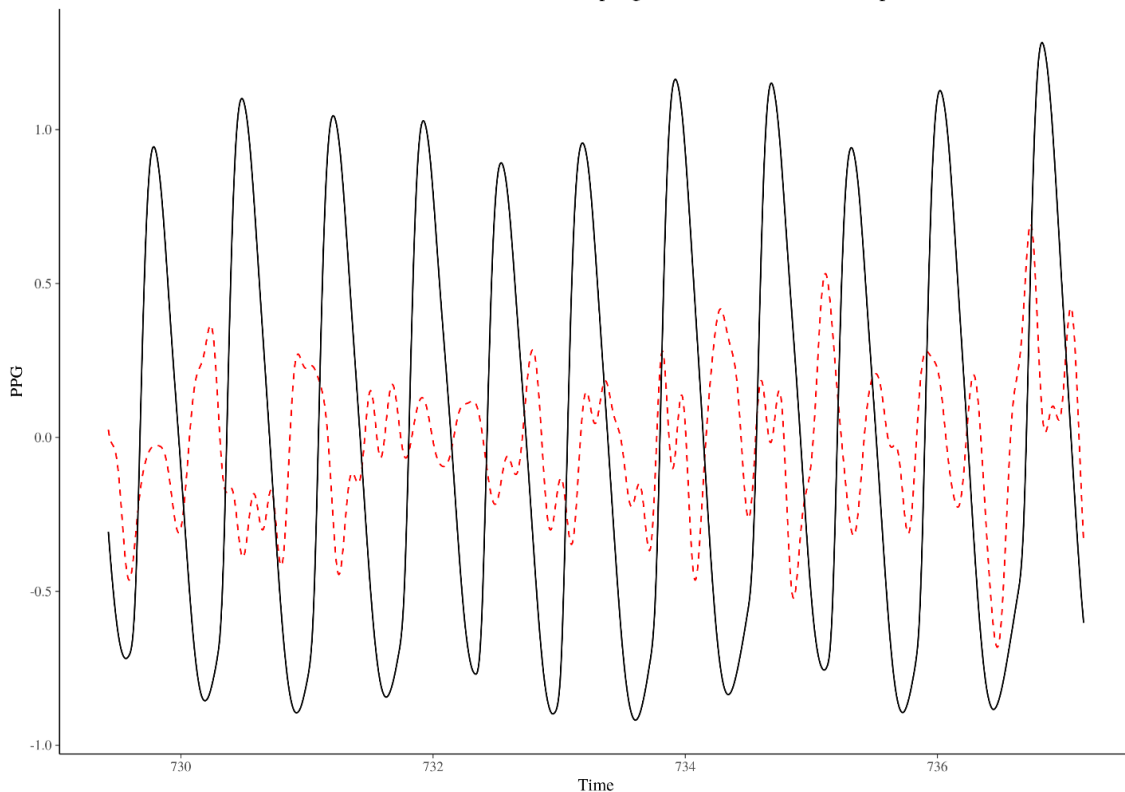


Figure 5.3 Example of a failed imputation run (in red) relative to the original signal (in black)

Keep It Simple. One theme that emerged from the present set of findings is that when a Gaussian process imputation model outperformed other, naïve, imputation strategies it was usually the result of the simplest model. As is often the case in science, greater complexity does not always mean greater accuracy or precision in the long run. Instead, it is often as likely that increased complexity can muddy the waters, allowing multiple factors unassociated with the true value of interest to cause undue influence on resulting predictions. This scientific and folk wisdom axiom, when applied in to the present research suggests that the first Gaussian process may be the best place to start when refining the implementation of imputation models in *IBI VizEdit*. Not only did it perform well, particularly when the model sampled the data at 8 Hz, it took much less

time to run, and converged more frequently on a heart rate frequency posterior see **Tables 4.1** and **4.2**.

Choosing the Right Estimation Framework. Narrowing the focus of future research in this vein to the first Gaussian process model is certainly defensible based on the results, but the findings in the present study are not enough to consider abandoning the pursuit of the two more complex imputation models entirely. Admittedly, relative to the first Gaussian process imputation model and even to more naïve imputation strategies they did not perform well as currently implemented. That is not to say that the present implementation is the optimal implementation of these models though, as it relies on a two-step process that first produces penalized maximum likelihood estimates of model hyperparameters and then, in a Bayesian framework, creates a posterior distribution of predicted values and a heart rate frequency parameter.

This two-step approach was adopted for computational convenience; however, in fixing parameters in the model to their maximum likelihood estimates, there is always the possibility that the values being fixed within the model represented a solution from a local maximum on the likelihood surface. As an exploratory step in the present study, maximum likelihood estimates were derived twice, using two different sets of random starting values. Comparing these values revealed that oftentimes the maximum likelihood estimates were not replicated (an indication that at least one of the points did not represent a global maximum), and in some cases the deviations in values could be quite large relative to the parameter being estimated (See **Figures 5.4** and **5.5**). Focusing on the second Gaussian process model (see **Figure 5.4**), the length hyperparameters, l_3 and l_5 , frequently exhibited wide divergence in the maximum likelihood point estimates returned

from two different random starts. In fact, several absolute difference scores exceeded 20 (not depicted in **Figure 5.4**).

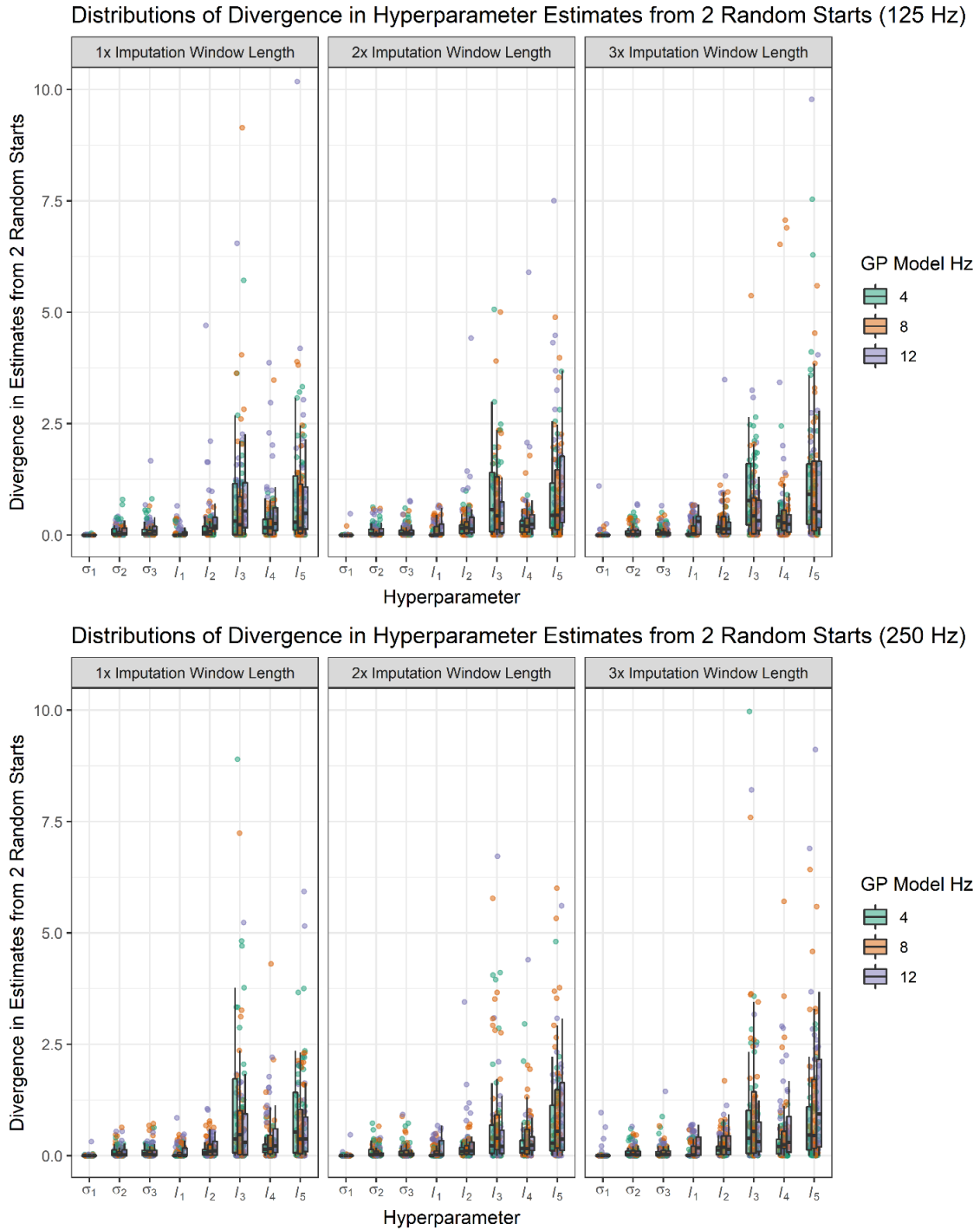


Figure 5.4 Divergence in hyperparameter estimates for Gaussian process model 2 based on absolute differences in estimates from two sets of random starting values

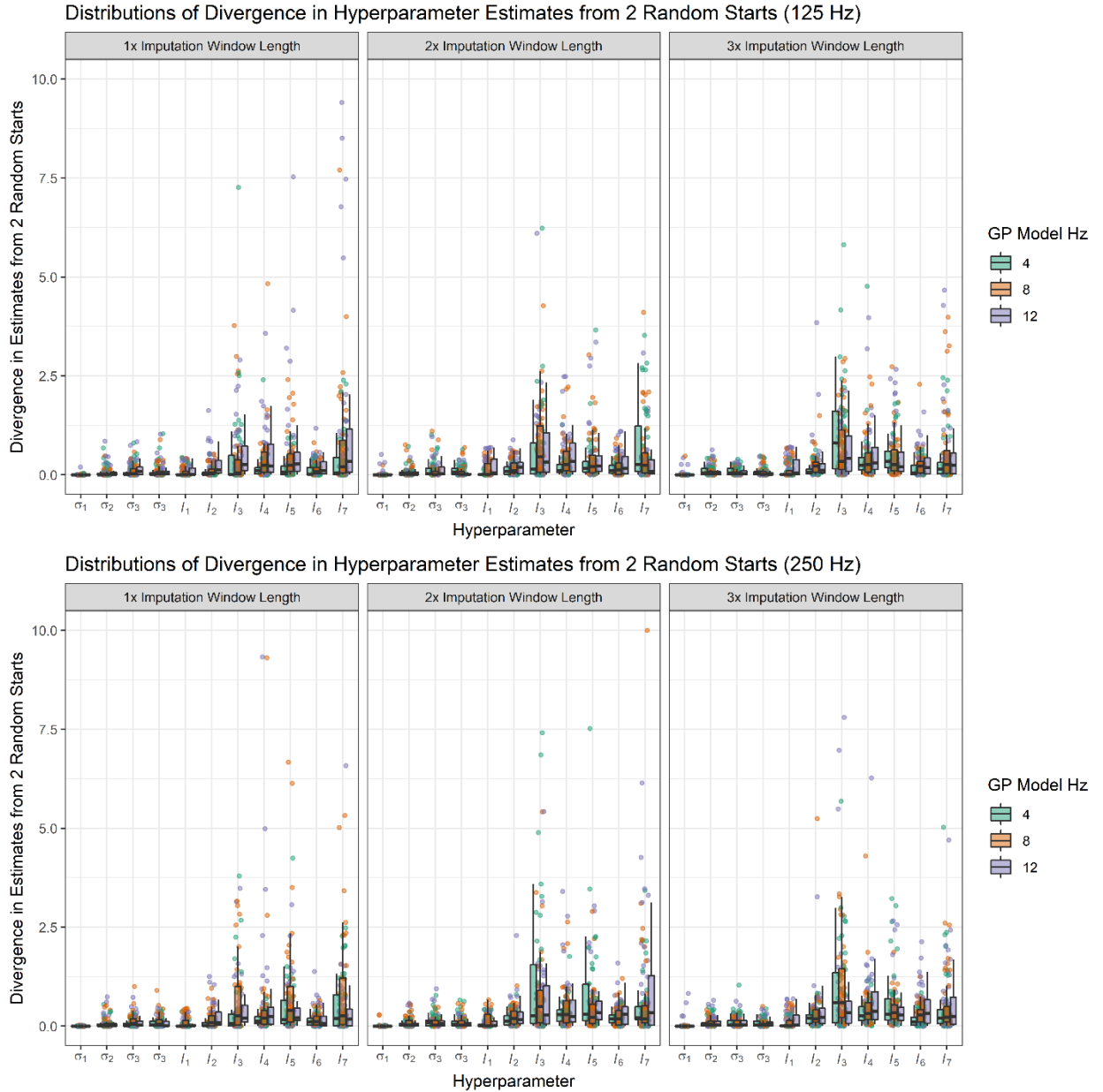


Figure 5.5 Divergence in hyperparameter estimates for Gaussian process model 3 based on absolute differences in estimates from two sets of random starting values

Mathematically, the length estimates impact the smoothness of the estimated mean function, which is used to derive the imputed values. Shorter estimates of the length hyperparameter will result in more rapid changes in the waveform (Gelman et al., 2013). In the context of modifying the imputation model, a length parameter for l_3 or l_5 that is

too small would result in a mean function that includes too many rapid changes in amplitude (see **Figure 5.6**). The opposite problem occurs if length parameters are too long as the wave form begins to “flatten” out (see **Figure 5.7**). Both would result in imputed values that do not align with the surrounding signal.

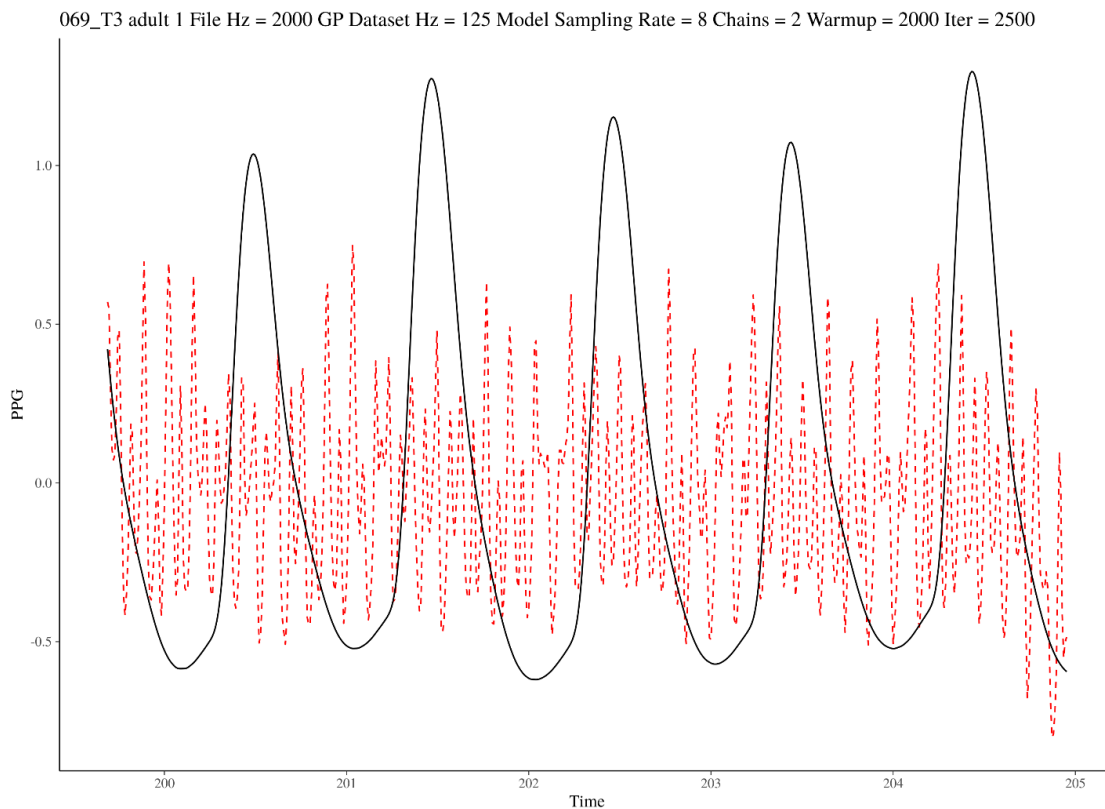


Figure 5.6 Example of Gaussian process model with incorrectly specified length parameters that are too short

One way to sidestep potential problems that arise from maximum likelihood estimates that represent a convergence on a local maximum is to iterate multiple times with random starting values and only take those parameter estimates that have been replicated. This approach is similar to that used by numerous implementations of maximum likelihood models widely available in a variety of software packages.

In thinking through the optimal implementation of these models as a means of imputing missing PPG data, it is worth considering whether an entirely frequentist approach may be more practically palatable than an entirely Bayesian or a mixed frequentist-Bayesian approach. Bayesian models can take a considerably long time to run, due to the time required for the independent chains to converge prior to drawing values for parameters from their posterior distributions. It is worth testing whether, using an approach based entirely on maximum likelihood estimates is worth considering, especially given the reduced computational burden such an approach might provide. At the very least, the present study indicates that concerns related to the convergence of model hyperparameters, the length parameters in particular, need to be addressed no matter the future implementation plans, be they entirely Bayesian, entirely frequentist, or a mixture of the two.

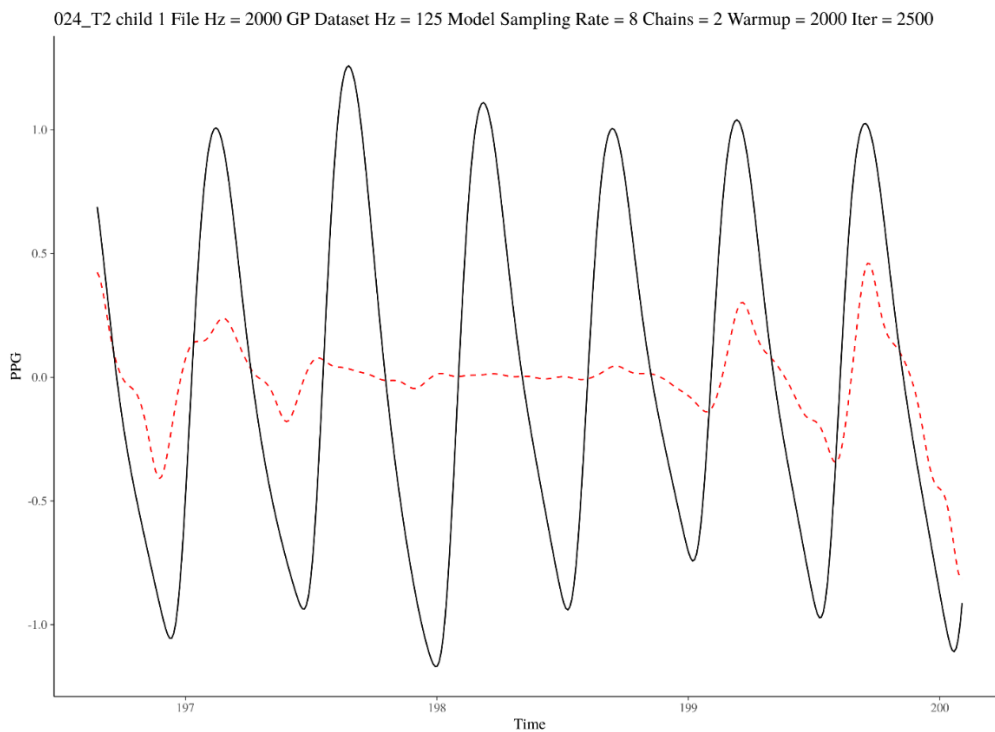


Figure 5.7 Example of a Gaussian process model with incorrectly specified length parameters that are too long

Summary. The results of the three imputation models do not provide convincing support for the use of any one model at this stage to impute missing PPG data. There are some indications, however, that the simplest of the three imputation models could be a useful tool to help guide editors in making reasonable choices when confronted with data that is too corrupted to effectively edit using existing strategies. The results from the present study suggest that there may be little benefit to increasing model complexity, and if anything, under the current implementation of the models, doing so tends to bias estimates of heart rate variability in the imputed section upward. The findings apply only to the current implementation of the imputation models and adjustments to the imputation process may yield an entirely different set of conclusions.

It is important to bear in mind that these models were never intended to replace human decision-making when editing heart rate data (at least not yet). Instead, they were meant to offer a reasonable guide, a trail of points, that an editor can choose to follow or not. As such, there is little “risk” to including a modified implementation of the first Gaussian process model in current versions of *IBI VizEdit*, provided that the editors understand that it may return untrustworthy values in some cases. As can be seen in **Figures 5.1, 5.2, 5.3, 5.6 and 5.7**, it is visually obvious when the results of an imputation cannot and should not be trusted to represent the missing data. Clearly, future work remains in further refining this imputation approach. At this early stage, there is enough evidence in the present report to at least consider using the first Gaussian process model on a provisional basis, with user oversight in the form of a trained editor choosing to accept or reject the proposed timing and location of heart beats. Long-term, there may be a way for these sorts of models to automatically process and edit sections of data,

reducing burden on study personnel. This method is not there yet, but there are enough successful imputation runs in the present data set to suggest that it some day could be.

Study 2: Can Trained Editors reliably use *IBI VizEdit*?

Gaussian process imputation models may represent the future of *IBI VizEdit*. Its present is still as a base graphical interface for processing, editing, and outputting usable heart rate data. For most applications in psychological and behavioral sciences we are interested in instruments, measures, and tools that provide both reliable (i.e., relatively free from error) and valid (i.e., measures the construct/phenomenon of interest) scores (but see Moss, 1994). Study 2 addressed the question of reliability of scores output by *IBI VizEdit* in two ways. The first was by evaluating intra-editor consistency in RSA scores across two different heart rate editing programs that rely on similar editing strategies: *IBI VizEdit* and *Cardio Edit*. The second involved the use of a third editor whose RSA scores were compared against the two primary editors who worked on the sample's heart rate files. RSA scores were the focus of efforts to evaluate reliable use of the program as they are more susceptible to slight changes in the final interbeat interval time series than other measures such as heart period.

Beginning with the intra-editor analyses, depending on the standards applied there was mixed evidence of *IBI VizEdit*'s reliable performance vis-à-vis the RSA scores produced by each editor on a randomly selected set of files. Using the Brain-Body Center's training standard of RSA values within +/- .05, difference in RSA scores between the two programs fell too often outside desired range. For Editor 1 nearly one in every two tasks that were edited in both programs resulted in RSA values that differed by more than .05. For Editor 2 it was approximately three out of every ten files.

It is worth asking though if this is the right standard to apply in this case. Scores generated by both programs were still highly correlated, and the distribution of scores was highly similar in range and measures of centrality (see **Figure 5.8**). If not always perfectly aligned, at least in a distributional sense, scores edited in each program tended to be located in very similar positions relative to other scores in the same set of cases. This latter feature may be more relevant for the purposes of making valid inferences than having these two programs more perfectly align in the absolute values of scores produced using each platform. It is after all unrealistic to expect a research team to edit a portion of its cases using one program, a portion using another program, and then combine the resulting values in a single set of scores to be analyzed. To apply the same strict threshold of +/- .05 to a cross-program comparison of values, with each program having slightly different sources of error, is likely too strict a standard. Comparing distributions of scores and the relative position of individual scores within those distributions (i.e., correlation) are likely better indicators of the degree to which *IBI VizEdit* can be used to reliably measure individuals' cardiac activity, at least when using a different program as the basis for the comparison.

On its face, a more encouraging set of results emerged when inter-editor reliability was evaluated using *IBI VizEdit*. For this analysis, a third editor was randomly assigned a subset of cases that had been previously edited by the two primary editors who worked on Study 2. Evaluating the editors on their consistency using the Brain-Body Center's threshold of .05 makes more sense in this case as all editors were working from the same platform. In all but one very extreme case, Editor 3 was in complete agreement with Editors 1 and 2 based on the .05 threshold. Not surprisingly, given the high rate of

correspondence in absolute values, correlations between the scores generated by Editor 3 and the two primary editors each exceeded .99. Taken together, these results suggest that editors who complete the training protocols associated with becoming certified to use IBI VizEdit are likely to generate highly similar scores working independently.

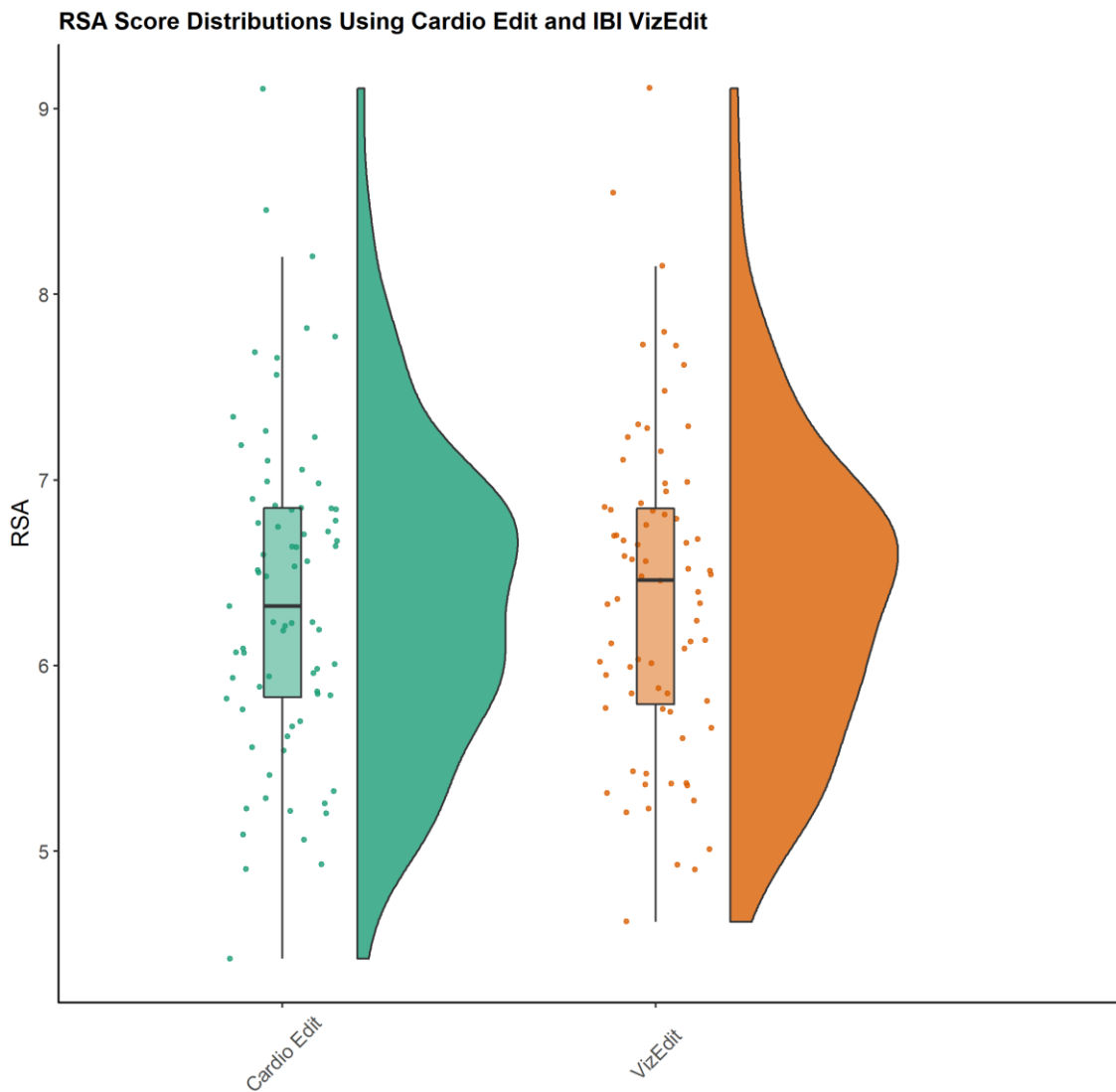


Figure 5.8 Comparing RSA scores from files edited in Cardio Edit vs. IBI VizEdit

Overall, the weight of evidence presented in the current report suggests that *IBI VizEdit* can be used to reliably edit photoplethysmographic heart rate data. Given that the main editing strategies in the program were reflective of those available in *Cardio Edit*, a

program that has been successfully used for years to edit heart rate data obtained via ECG sensors (Glenn et al., 2018; Guy et al., 2014; Kenkel et al., 2014; Lamm et al., 2008; Lewis et al., 2012), it is not that surprising that editors could be trained to a similar degree of reliability using *IBI VizEdit*. Additional questions remain and could be worth exploring as to whether one of these two programs is more accurate in its final estimates given the slight divergence in absolute values, but the present design was not positioned to test such a research question.

Study 2: Can *IBI VizEdit* be used to produce valid estimates of heart rate and autonomic regulation of cardiac activity in a laboratory setting?

The main evidence in the present set of studies that points to the ability to draw valid inferences when using *IBI VizEdit* to edit interbeat interval files is that RSA and heart period from files edited in *IBI VizEdit* varied as expected with a set of prescribed laboratory tasks. Three of the tasks involved watching the same age-appropriate Sesame Street video intended to engage children's attention while simultaneously engendering a relaxed state. Two of the tasks were designed to be social stressors, one an unexpected social interaction with an adult RA dressed as a clown, the other a request to record a video introduction so peers could watch later. The fourth task, learning about similar-aged peers, was meant to tap cognitive and emotional states while engaged in basic social information processing.

As expected RSA scores and heart period were both lower during the social stressor tasks and were higher during the videos and during the social learning task. These results reflect a series of neurophysiological responses in which vagal influence on cardiac activity is withdrawn during periods of distress (i.e., the Clown and Introduction

tasks) and is stabilized or even potentially increased during periods of engaged attention (Brooker & Buss, 2010; Hastings, Nuselovici, et al., 2008). Being the main source of short-term variability in heart rate, as signaling along the vagus nerve increases or decreases so too does heart-rate variability, especially within the respiration cycle as alterations to cardiac activity as a function of respiration are entirely dependent on the myelinated vagus nerve.

A natural consequence of the withdrawal of vagal inputs to the heart is a commensurate decrease in heart period (i.e., increase in heart rate), as the vagus nerve provides tonic inhibitory input into the heart's internal pacemaker, the sinoatrial node. Heart period will also decrease as a function of sympatho-excitatory inputs, though these were not measured directly in the present study. Absent other direct measures of autonomic activity, heart period has come to increasingly be used and described as an indicator of general arousal (Adenauer, Catani, Keil, Aichinger, & Neuner, 2010; Brosschot & Thayer, 2003; Sloan, 2004). The logic for this more general understanding of heart rate reactivity is that, with multiple autonomic sources, the decrease in heart period may reflect the withdrawal of vagal influence, or it could be a combination of vagal withdrawal and sympathetic excitation. Unable to disentangle the neurophysiological sourcing of the decrease in heart period, it is at least in line with expectations that children's overall levels of arousal increased when presented with the two different social challenges they experienced in the lab. Whether or not the decrease in heart period was due to elevated activity along sympathetic signaling pathways, such as the hypothalamic-pituitary-adrenal axis associated with the body's stress response,

remains a question for future studies that incorporate direct measures of sympathetic nervous system activity (e.g., galvanic skin response).

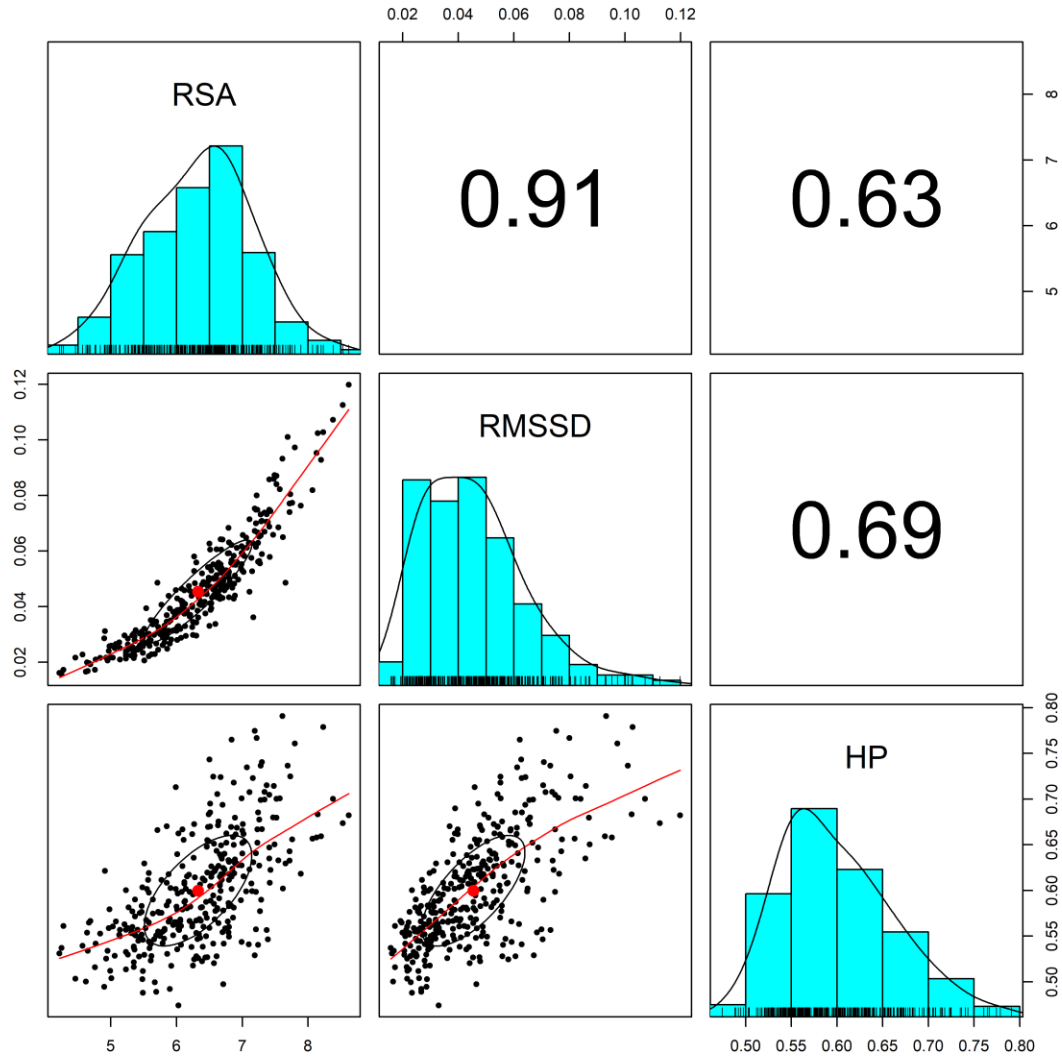


Figure 5.9 Within-task correlations of physiological variables derived from edited interbeat interval files

Note. RSA = respiratory sinus arrhythmia; RMSSD = root mean square of successive differences; and HP = heart period.

RMSSD values derived from interbeat intervals measured during each task did not respond to the different conditions as expected. Though highly correlated in the present sample with RSA (see **Figure 5.9**), changes in individuals' RMSSD values from task to task did not emerge in the present set of analyses. Despite estimates being in

expected directions, changes in RMSSD during the Clown and Introduction tasks (relative to the other tasks) did not reliably differ from 0, according to 95% credibility intervals. These null effects may be driven by two factors in the present data, both of which likely interacted to some degree. First and foremost is the fact that the models relied on data from a relatively small sample. Second, and more theoretically meaningful, is the fact that RMSSD is not considered to be as “pure” a measure of parasympathetic influence on the heart as Porges’ RSA, the latter of which has been shown to be especially responsive to chemical blockades of vagal signaling (Lewis et al., 2012). Thus, a noisier measure, in a relatively small sample is less likely to produce posterior distributions of parameter estimates in line with *a priori* expectations of task differences. As more data is included from the complete data frame (Study 2 only incorporated data from one of the two study conditions), it will of interest to see, as the number of usable cases approaches 120 subjects, if the null findings related to RMSSD hold.

Study 2: Can estimates of parasympathetic regulation of cardiac activity derived from files edited in IBI VizEdit be used to predict children’s social behavior in a real-world context?

According to the results from the present set of models, measures of parasympathetic regulation of heart rate in a prescribed series of laboratory tasks were unrelated to social behaviors in a real-world context. The rationale for including this set of analyses initially was as a means of testing a form of criterion validity. Previous work has shown that children who are anxious and withdrawn and tend to struggle socially tend to also have lower RSA, and depending on the task, condition, and/or context, exhibit potentially maladaptive regulatory patterns as measured by dynamic changes in RSA

(Beauchaine, 2001; Beauchaine, 2015; Fox, Henderson, et al., 2005). Thus, to the degree that the above statements can be accepted to be true, RSA values, both tonic and dynamic, were thought, *a priori*, to likely be related to children's patterns of social play. Though the primary play behaviors of interest were children's tendencies to engage in reticent/withdrawn behavior and group play behavior during periods of classroom free play, additional solitary behaviors were examined as well.

Despite these theoretically based expectations, none of the models revealed an association between RSA, tonic or dynamic, and any of the play behaviors coded as part of the study. The same was true of heart period and changes in heart period. Of course, failing to reject a null hypothesis is not evidence that the null hypothesis is true, simply that the evidence available in the data was not sufficiently strong to indicate otherwise. In the context of the present study, the failure to find an association between physiology as measured in a laboratory setting and play behaviors as observed in a real-world context may have been partly driven by several different factors.

Signal and Noise. Perhaps the most obvious issue with the present set of analyses is relative lack of statistical power. Due to issues collecting physiological recordings from some children and the fact that some schools refused to allow study personnel on site, only 52 participants out of a possible 75 had sufficient data to be included in the models. Additionally, RSA and heart period were not available for all tasks for all participants as there were sections of data that were deemed unusable, adding another layer of missingness and reducing the total information available for models to draw from. If these were the only factors at play, they would be enough to make it difficult to reject the null hypothesis under the conditions that a true small or moderate association between

laboratory-assessed physiology and play behaviors did exist in the population (i.e., highly inhibited children).

Adding to issues suppressing signal strength, various sources of noise existed including variation in the classroom climates in which each child was observed, the structure of free play during which children were observed, whether the play occurred inside or outside, and what sorts of play equipment and activities were available in the classroom. Sadly, there are no direct records of these sorts of fluctuations in classroom contexts, but they undoubtedly existed.

As hinted at above, if signal-to-noise ratio is a factor in obscuring a true connection between neurophysiological variables assessed in a lab and play behaviors observed in the real world, doubling the sample (i.e., adding data from the second group randomized the *Cool Little Kids*) in the analyses may lead to alternative conclusions. There is no guarantee that will take place of course, and there are other considerations, beyond statistical power that cannot be addressed by simply adding more data to the models.

The Problem of Restricted Range. The data in the present pair of studies were drawn from a selected sample, one that was elevated on a measure of early risk for the onset of anxiety disorders. Previous work has largely established that anxiety and BI are tied to dysregulated autonomic functioning, with some moderation effects observed that are related to parenting and parent-child relationships (Fox, Henderson, et al., 2005; Henderson, Marshall, Fox, & Rubin, 2004; Marshall & Stevenson-Hinde, 1998). The children involved in the Study 2, therefore likely represented a restricted range of tonic and dynamic RSA values relative to the population of preschool children at large. The

same is true of their tendency to engage in certain play behaviors. In both cases, the population sampled in the present study likely does not represent the full range of variability in terms of physiological responses to challenge or play among familiar peers. Many of the previous studies linking physiology to social behaviors involved populations and samples with wider ranges of scores along both dimensions (Boyce et al., 2001; Miller et al., 2017; Utendale et al., 2014). Thus, one inescapable and largely non-rectifiable component of the attempts to link laboratory-based physiology with classroom behaviors in Study 2 is the selected nature of the sample, which was critical for the intervention aspect of the overarching research. However, the fact that all children in the current study were at or above the 85th percentile on a measure of BI makes certain basic research questions, such as the linking of lab measures of physiology to play behaviors, more challenging from a statistical standpoint.

Alternative Features to Extract. In Study 2, tonic RSA and heart period, as measured during the video portions of the laboratory assessment, and dynamic RSA and heart period in response to the non-video components of the session represented the main features extracted from the cardiac data. These are not the only potentially relevant aspects of cardiac activity, however. For instance, Hastings and colleagues along with others have found the use of quadratic and other non-linear modeling approaches to represent maladaptation at the extremes of physiological functioning particularly useful when predicting indicators of social and emotional adaptation and maladaptation (Kleiger et al., 1987; Kogan et al., 2014, 2013; Miller et al., 2017). Additionally, more nuanced time-series analyses of the data would offer the opportunity to examine change, for example, at the moment the socially stressful stimulus is presented to the child (i.e., as

soon as the Clown opens the door). Along these lines, the latency to return to a baseline or latency to reach a minimum or maximum value all may represent more nuanced measures of autonomic regulation of heart rate in response to stressors and versions of each of these more precise measures of change (as opposed to a difference in averages) have been utilized by researchers in the past.

Misalignment of Tasks with Enacting Play Behaviors. Lurking beneath the discussion of various statistical and measurement issues that may underlie the null result, it is entirely possible that the task itself is not particularly well-designed as a means of capturing tonic physiology or changes in physiology that are relevant in a free play context. Taking the two social stressors as examples, interacting with an adult dressed as a clown is several social orders of magnitude away from assembling small structures of cardboard blocks in a recurring setting with a potential friend or, at the very least, an acquainted classmate. Speaking into a camera to record an introduction, though likely raising self-conscious emotional and cognitive states, is not the same as asking a familiar peer to play. Children who showed one physiological reaction while put in the highly artificial situation of interacting with a clown or recording an introduction in an unfamiliar context, may have a completely different physiological, emotional, and cognitive profile when interacting in a familiar context with a familiar agemate. In fact, it may even be likely that children who respond in an anxious or inhibited fashion in one context are not especially likely to do so in another (e.g., Asendorpf, 1990; Rubin, Hastings, Stewart, Henderson, & Chen, 1997). BI, as it was originally conceptualized, was defined as fear and wariness in the face of the *unfamiliar* (Kagan et al., 1987), which may or may not be particularly predictive of how individual children fare in their

everyday surroundings, particularly in the safe and likely welcoming environment that is their preschool classroom.

Conclusions

Across these two studies, the reality cast in the sharpest relief is that *IBI VizEdit* is a sapling of a program, containing considerable promise. As it stands, it can perform the basic tasks of editing heart rate data obtained from PPG sensors, conveniently taking raw data and outputting usable values that can be incorporated in models of the analyst's choosing. Editors can be trained to reliability, and, supporting the idea that the values derived from its output are valid, resulting scores vary as expected with a series of prescribed laboratory tasks. Branching out beyond existing editing software, the program provides users with an easily digested output report, replete with information about edited values, total number of edits, proportion of tasks deemed uneditable, and average heart rate and heart rate variability scores by task. It was designed to be a one-stop shop of sorts for editing PPG data, and so far, much of that pledge has been realized while preparing the data for analyses in the present pair of studies. That is not to say that there is not room for refinement and growth, both in terms of increased practical functionality and in continuing to push the envelope related to model-based approaches to dealing with problematic PPG data. Adjustments to the Gaussian process imputation models and a re-examination of imputation performance rank high in priority among the natural offshoots stemming from this dissertation. The development of an *R* package, modification of the program to allow a variety of file formats, inclusion of processing steps that aid in cleaning data obtained during fMRI sessions, each represents a new avenue of growth. What matters most is that there is now a base from which such growth can be supported.

APPENDICES

Appendix A: Programming Components of IBI VizEdit

Appendix A.1 Peak Detection Algorithm in R:

```
findpeaks <- function (x, m = 3){
  shape <- diff(sign(diff(x, na.pad = T)))
  pks <- sapply(which(shape < 0), FUN = function(i){
    z <- i - m + 1
    z <- ifelse(z > 0, z, 1)
    w <- i + m + 1
    w <- ifelse(w < length(x), w, length(x))
    if(all(x[c(z : i, (i + 2) : w)] <= x[i + 1])) return(i + 1) else return(numeric(0))
  })
  pks <- unlist(pks)
  pks
}
```

Appendix A.2 Extracting Interbeat Intervals

```
time.sum<-function(x){
  Z<-rep(NA, length(x))
  for(i in 1:length(x)){
    Z[i]<-ifelse(i==1, x[i], x[i]-x[i-1])
  }
  return(Z)
}

IBI.sum<-function(x){
  Z<-rep(NA, length(x))
  for(i in 1:length(x)){
    Z[i]<-sum(x[1:i])
  }
  return(Z)
}
```

Appendix A.3 Iterative Peak Detection Algorithm

```
iter.IBI<-function(x, ds=500, iter.length){
  require(psych)
  #browser()
  x.smooth<-as.numeric(smooth(x))
  x.smooth<-na.omit(x.smooth)
  TIME<-0:(length(x.smooth)-1)
  x.smooth<-x.smooth-predict(lm(x.smooth~TIME))
  s<-round(seq(round(ds*.15), round(ds*3/4), length.out = iter.length))
  s<-unique(s)
  Z<-data.frame(rep(NA, length(s)),
               rep(NA, length(s)),
               rep(NA, length(s)),
               rep(NA, length(s)),
               rep(NA, length(s)),
               rep(NA, length(s)))
  for(i in 1:length(s)){
    IBI<-findpeaks(x.smooth, s[i])
    time<-time.sum(IBEI)/ds
    Z[i,1]<-s[i]
    Z[i,2]<-sd(time)
    Z[i,3]<-max(time)-min(time)
  }
}
```



```

Z[i,4]<-rmssd(time)
Z[i,5]<-mean(acf(time, lag.max = length(time)/20, plot = F)$acf)
Z[i,6]<-s[i]/ds
}
colnames(Z)<-c('BW', 'SD', 'Range', 'RMSSD', 'AC', 'BW(s)')
Z<-Z[order(c(Z$RMSSD, Z$Range, Z$BW), decreasing = F),]
print(head(na.omit(Z)))
IBI.fin<-findpeaks(x.smooth, m=na.omit(Z)[1,1])-1
IBI.time<-IBI.fin/ds
IBI.done<-time.sum(IBI.time)
IBI<-data.frame(IBI=IBI.done, Time=IBI.time)
return(IBI[c(-1,-length(IBI$IBI)),])
}

```

Appendix A.4: Stan Code for 2-Process Gaussian Process Imputation Model

```

functions{
  //covariance function for main portion of the model
  matrix main_GP(
    int Nx,
    vector x,
    int Ny,
    vector y,
    real alpha1,
    real alpha2,
    real rho1,
    real rho2,
    real rho3,
    real HR_f){
    matrix[Nx, Ny] K1;
    matrix[Nx, Ny] K2;
    matrix[Nx, Ny] Sigma;

    //periodic covariance that does not decay
    for(i in 1:Nx){
      for(j in 1:Ny){
        K1[i, j] = alpha1*exp(-square(x[i]-y[j])/2/square(rho1));
      }
    }

    //specifying first quasi-periodic process that incorporates heart rate
    for(i in 1:Nx){
      for(j in 1:Ny){
        K2[i, j] = alpha2*exp(-2*square(sin(pi()*fabs(x[i]-
y[j]))*HR_f))/square(rho2))*
        exp(-square(x[i]-y[j])/2/square(rho3));
      }
    }
    Sigma = K1+K2;
    return Sigma;
  }
  //function for posterior calculations
  vector post_pred_rng(
    real a1,
    real a2,
    real r1,
    real r2,
    real r3,
    real HR,
    real sn,
    int No,
    vector xo,
    int Np,
    vector xp,
    vector yobs){
    matrix[No,No] Ko;

```

```

        matrix[Np,Np] Kp;
        matrix[No,Np] Kop;
        matrix[Np,No] Ko_inv_t;
        vector[Np] mu_p;
        matrix[Np,Np] Tau;
        matrix[Np,Np] L2;
        vector[Np] yp;

//-----
//Kernel Multiple GPs for observed data
Ko = main_GP(No, xo, No, xo, a1, a2, r1, r2, r3, HR);
for(n in 1:No) Ko[n,n] += sn;

//-----
//kernel for predicted data
Kp = main_GP(Np, xp, Np, xp, a1, a2, r1, r2, r3, HR);
for(n in 1:Np) Kp[n,n] += sn;

//-----
//kernel for observed and predicted cross
Kop = main_GP(No, xo, Np, xp, a1, a2, r1, r2, r3, HR);

//-----
//Algorithm 2.1 of Rassmussen and Williams...
Ko_inv_t = Kop'/Ko;
mu_p = Ko_inv_t*yobs;
Tau=Kp-Ko_inv_t*Kop;
L2 = cholesky_decompose(Tau);
yp = mu_p + L2*rep_vector(normal_rng(0,1), Np);
return yp;
}
}

data {
  int<lower=1> N1;
  int<lower=1> N2;
  vector[N1] X;
  vector[N1] Y;
  vector[N2] Xp;
  real<lower=0> mu_HR;
  real<lower=0> sigma_HR;
}

transformed data {
  vector[N1] mu;
  for(n in 1:N1) mu[n] = 0;
}

parameters {
  real<lower=0> a1;
  real<lower=0> a2;
  real<lower=0> r1;
  real<lower=0> r2;
  real<lower=0> r3;
  real<lower = 0.8333, upper = 3.3333> HR;
  real<lower=0> sigma_sq;
}

model{
  matrix[N1,N1] Sigma;
  matrix[N1,N1] L_S;

  //using GP function from above
  Sigma = main_GP(N1, X, N1, X, a1, a2, r1, r2, r3, HR);
  for(n in 1:N1) Sigma[n,n] += sigma_sq;

  L_S = cholesky_decompose(Sigma);
  Y ~ multi_normal_cholesky(mu, L_S);
}

```

```

//priors for parameters
a1 ~ normal(0,2);
a2 ~ normal(0,2);
//incorporate minimum and maximum distances - use invgamma
r1 ~ inv_gamma(4,4);
r2 ~ inv_gamma(4,4);
r3 ~ inv_gamma(4,4);
sigma_sq ~ normal(0,2);
HR ~ normal(mu_HR,sigma_HR);
}

generated quantities {
  vector[N2] Ypred = post_pred_rng(a1, a2, r1, r2, r3, HR, sigma_sq, N1, X, N2, Xp, Y);
}

```

Appendix A.5: Stan Code for 3-Process Gaussian Process Imputation Model

```

functions{
  //covariance function for main portion of the model
  matrix main_GP(
    int Nx,
    vector x,
    int Ny,
    vector y,
    real alpha1,
    real alpha2,
    real alpha3,
    real rho1,
    real rho2,
    real rho3,
    real rho4,
    real rho5,
    real HR_f,
    real R_f){
    matrix[Nx, Ny] K1;
    matrix[Nx, Ny] K2;
    matrix[Nx, Ny] K3;
    matrix[Nx, Ny] Sigma;

    //periodic covariance that does not decay
    for(i in 1:Nx){
      for(j in 1:Ny){
        K1[i, j] = alpha1*exp(-square(x[i]-y[j])/2/square(rho1));
      }
    }

    //specifying first quasi-periodic process that incorporates heart rate
    for(i in 1:Nx){
      for(j in 1:Ny){
        K2[i, j] = alpha2*exp(-2*square(sin(pi()*fabs(x[i]-
y[j]))*HR_f))/square(rho2))*
        exp(-square(x[i]-y[j])/2/square(rho3));
      }
    }

    //specifying second quasi-periodic process that incorporates heart rate
    for(i in 1:Nx){
      for(j in 1:Ny){
        K3[i, j] = alpha3*exp(-2*square(sin(pi()*fabs(x[i]-
y[j]))*HR_f))/square(rho4))*
        exp(-2*square(sin(pi()*fabs(x[i]-y[j]])*R_f))/square(rho5));
      }
    }
    Sigma = K1+K2+K3;
    return Sigma;
  }
}

```

```

//function for posterior calculations
vector post_pred_rng(
  real a1,
  real a2,
  real a3,
  real r1,
  real r2,
  real r3,
  real r4,
  real r5,
  real HR,
  real R,
  real sn,
  int No,
  vector xo,
  int Np,
  vector xp,
  vector yobs){
  matrix[No,No] Ko;
  matrix[Np,Np] Kp;
  matrix[No,Np] Kop;
  matrix[Np,No] Ko_inv_t;
  vector[Np] mu_p;
  matrix[Np,Np] Tau;
  matrix[Np,Np] L2;
  vector[Np] yp;

  //-----
  //Kernel Multiple GPs for observed data
  Ko = main_GP(No, xo, No, xo, a1, a2, a3, r1, r2, r3, r4, r5, HR, R);
  for(n in 1:No) Ko[n,n] += sn;

  //-----
  //kernel for predicted data
  Kp = main_GP(Np, xp, Np, xp, a1, a2, a3, r1, r2, r3, r4, r5, HR, R);
  for(n in 1:Np) Kp[n,n] += sn;

  //-----
  //kernel for observed and predicted cross
  Kop = main_GP(No, xo, Np, xp, a1, a2, a3, r1, r2, r3, r4, r5, HR, R);

  //-----
  //Algorithm 2.1 of Rasmussen and Williams...
  Ko_inv_t = Kop'/Ko;
  mu_p = Ko_inv_t*yobs;
  Tau=Kp-Ko_inv_t*Kop;
  L2 = cholesky_decompose(Tau);
  yp = mu_p + L2*rep_vector(normal_rng(0,1), Np);
  return yp;
}

}

data {
  int<lower=1> N1;
  int<lower=1> N2;
  vector[N1] X;
  vector[N1] Y;
  vector[N2] Xp;
  real<lower=0> mu_HR;
  real<lower=0> mu_R;
  real<lower=0> sigma_HR;
  real<lower=0> sigma_R;
}

transformed data {
  vector[N1] mu;
  for(n in 1:N1) mu[n] = 0;
}

```

```

parameters {
  real<lower=0> a1;
  real<lower=0> a2;
  real<lower=0> a3;
  real<lower=0> r1;
  real<lower=0> r2;
  real<lower=0> r3;
  real<lower=0> r4;
  real<lower=0> r5;
  real<lower=0> r6;
  real<lower = 0.8333, upper = 3.3333> HR;
  real<lower = 0.1667, upper = 0.5000> R;
  real<lower=0> sigma_sq;
}

model{
  matrix[N1,N1] Sigma;
  matrix[N1,N1] L_S;

  //using GP function from above
  Sigma = main_GP(N1, X, N1, X, a1, a2, a3, r1, r2, r3, r4, r5, HR, R);
  for(n in 1:N1) Sigma[n,n] += sigma_sq;

  L_S = cholesky_decompose(Sigma);
  Y ~ multi_normal_cholesky(mu, L_S);

  //priors for parameters
  a1 ~ normal(0,2);
  a2 ~ normal(0,2);
  a3 ~ normal(0,2);
  //incorporate minimum and maximum distances - use invgamma
  r1 ~ inv_gamma(4,4);
  r2 ~ inv_gamma(4,4);
  r3 ~ inv_gamma(4,4);
  r4 ~ inv_gamma(4,4);
  r5 ~ inv_gamma(4,4);
  sigma_sq ~ normal(0,2);
  HR ~ normal(mu_HR, sigma_HR);
  R ~ normal(mu_R, sigma_R);
}

generated quantities {
  vector[N2] Ypred = post_pred_rng(a1, a2, a3, r1, r2, r3, r4, r5, HR, R, sigma_sq, N1, X, N2,
  Xp, Y);
}

```

Appendix A.6: Stan Code for 4-Process Gaussian Process Imputation Model

```

functions{
  //covariance function for main portion of the model
  matrix main_GP(
    int Nx,
    vector x,
    int Ny,
    vector y,
    real alpha1,
    real alpha2,
    real alpha3,
    real alpha4,
    real rho1,
    real rho2,
    real rho3,
    real rho4,
    real rho5,
    real rho6,
    real rho7,

```

```

real HR_f,
real R_f){
    matrix[Nx, Ny] K1;
    matrix[Nx, Ny] K2;
    matrix[Nx, Ny] K3;
    matrix[Nx, Ny] K4;
    matrix[Nx, Ny] Sigma;

    //periodic covariance that does not decay
    for(i in 1:Nx){
        for(j in 1:Ny){
            K1[i, j] = alpha1*exp(-square(x[i]-y[j])/2/square(rho1));
        }
    }

    //specifying first quasi-periodic process that incorporates heart rate
    for(i in 1:Nx){
        for(j in 1:Ny){
            K2[i, j] = alpha2*exp(-2*square(sin(pi()*fabs(x[i]-
y[j]))*HR_f))/square(rho2))*
            exp(-square(x[i]-y[j])/2/square(rho3));
        }
    }

    //specifying second quasi-periodic process that incorporates heart rate
    for(i in 1:Nx){
        for(j in 1:Ny){
            K3[i, j] = alpha3*exp(-2*square(sin(pi()*fabs(x[i]-
y[j]))*HR_f))/square(rho4))*
            exp(-2*square(sin(pi()*fabs(x[i]-y[j])*R_f))/square(rho5));
        }
    }
    for(i in 1:Nx){
        for(j in 1:Ny){
            K4[i, j] = alpha4*exp(-2*square(sin(pi()*fabs(x[i]-
y[j]))*HR_f))/square(rho6))*
            exp(-2*square(sin(pi()*fabs(x[i]-y[j])*1))/square(rho7));
        }
    }
    Sigma = K1+K2+K3+K4;
    return Sigma;
}

//function for posterior calculations
vector post_pred_rng(
    real a1,
    real a2,
    real a3,
    real a4,
    real r1,
    real r2,
    real r3,
    real r4,
    real r5,
    real r6,
    real r7,
    real HR,
    real R,
    real sn,
    int No,
    vector xo,
    int Np,
    vector xp,
    vector yobs){
    matrix[No,No] Ko;
    matrix[Np,Np] Kp;
    matrix[No,Np] Kop;
    matrix[Np,No] Ko_inv_t;
    vector[Np] mu_p;
    matrix[Np,Np] Tau;
}

```

```

        matrix[Np,Np] L2;
        vector[Np] yp;

//-----
//Kernel Multiple GPs for observed data
Ko = main_GP(No, xo, No, xo, a1, a2, a3, a4, r1, r2, r3, r4, r5, r6, r7, HR, R);
for(n in 1:No) Ko[n,n] += sn;

//-----
//kernel for predicted data
Kp = main_GP(Np, xp, Np, xp, a1, a2, a3, a4, r1, r2, r3, r4, r5, r6, r7, HR, R);
for(n in 1:Np) Kp[n,n] += sn;

//-----
//kernel for observed and predicted cross
Kop = main_GP(No, xo, Np, xp, a1, a2, a3, a4, r1, r2, r3, r4, r5, r6, r7, HR, R);

//-----
//Algorithm 2.1 of Rasmussen and Williams...
Ko_inv_t = Kop'/Ko;
mu_p = Ko_inv_t*yobs;
Tau=Kp-Ko_inv_t*Kop;
L2 = cholesky_decompose(Tau);
yp = mu_p + L2*rep_vector(normal_rng(0,1), Np);
return yp;
}
}

data {
  int<lower=1> N1;
  int<lower=1> N2;
  vector[N1] X;
  vector[N1] Y;
  vector[N2] Xp;
  real<lower=0> mu_HR;
  real<lower=0> mu_R;
  real<lower=0> sigma_HR;
  real<lower=0> sigma_R;
}

transformed data {
  vector[N1] mu;
  for(n in 1:N1) mu[n] = 0;
}

parameters {
  real<lower=0> a1;
  real<lower=0> a2;
  real<lower=0> a3;
  real<lower=0> a4;
  real<lower=0> r1;
  real<lower=0> r2;
  real<lower=0> r3;
  real<lower=0> r4;
  real<lower=0> r5;
  real<lower=0> r6;
  real<lower=0> r7;
  real<lower = 0.8333, upper = 3.3333> HR;
  real<lower = 0.1667, upper = 0.5000> R;
  real<lower=0> sigma_sq;
}

model{
  matrix[N1,N1] Sigma;
  matrix[N1,N1] L_S;

  //using GP function from above
  Sigma = main_GP(N1, X, N1, X, a1, a2, a3, a4, r1, r2, r3, r4, r5, r6, r7, HR, R);
  for(n in 1:N1) Sigma[n,n] += sigma_sq;
}

```

```

L_S = cholesky_decompose(Sigma);
Y ~ multi_normal_cholesky(mu, L_S);

//priors for parameters
a1 ~ normal(0,2);
a2 ~ normal(0,2);
a3 ~ normal(0,2);
a4 ~ normal(0,2);
//incorporate minimum and maximum distances - use invgamma
r1 ~ inv_gamma(4,4);
r2 ~ inv_gamma(4,4);
r3 ~ inv_gamma(4,4);
r4 ~ inv_gamma(4,4);
r5 ~ inv_gamma(4,4);
r6 ~ inv_gamma(4,4);
r7 ~ inv_gamma(4,4);
sigma_sq ~ normal(0,2);
HR ~ normal(mu_HR, sigma_HR);
R ~ normal(mu_R, sigma_R);
}

generated quantities {
  vector[N2] Ypred = post_pred_rng(a1, a2, a3, a4, r1, r2, r3, r4, r5, r6, r7, HR, R, sigma_sq,
  N1, X, N2, Xp, Y);
}

```


Appendix B: Model and Data Analysis/Summary Code

Appendix B.1: Summarizing Gaussian Process Model Performance Under Varying Conditions - getting basic summary tables together.

```
#Extracting summary data from GP data sets
GP1_summary <- dat.GP1 %>%
  group_by(impute.Hz.vec, impute.fac.vec, GP.Hz.vec)%>%
  summarize(Med_bias = median(bias_rmssd15),
            Q75_bias = quantile(bias_rmssd15, .75),
            Q25_bias = quantile(bias_rmssd15, .25),
            Mean_bias = mean(bias_rmssd15),
            SD_bias = sd(bias_rmssd15),
            Max_bias = max(bias_rmssd15),
            Min_bias = min(bias_rmssd15),
            Mean_HR = mean(HR.mean.vec)*60,
            SD_HR = sd(HR.mean.vec)*60,
            Max_HR = max(HR.mean.vec)*60,
            Min_HR = min(HR.mean.vec)*60,
            Med_Rhat = median(HR.Rhat.vec),
            Mean_Rhat = mean(HR.Rhat.vec),
            Prop_Rhat_1.1 = length(HR.Rhat.vec[HR.Rhat.vec<1.1])/40*100,
            Mean_tot.time = mean(impute.tot),
            Med_runtime = median(runtime),
            Mean_runtime = mean(runtime),
            Max_runtime = max(runtime),
            Min_runtime = min(runtime))

GP1_125_summary_4<-GP1_summary[GP1_summary$impute.Hz.vec==125 & GP1_summary$GP.Hz.vec==4,]
t_GP1_125_summary_4<-t(GP1_125_summary_4)
colnames(t_GP1_125_summary_4)<-c('1x', '2x', '3x')

GP1_125_summary_8<-GP1_summary[GP1_summary$impute.Hz.vec==125 & GP1_summary$GP.Hz.vec==8,]
t_GP1_125_summary_8<-t(GP1_125_summary_8)
colnames(t_GP1_125_summary_8)<-c('1x', '2x', '3x')

GP1_125_summary_12<-GP1_summary[GP1_summary$impute.Hz.vec==125 & GP1_summary$GP.Hz.vec==12,]
t_GP1_125_summary_12<-t(GP1_125_summary_12)
colnames(t_GP1_125_summary_12)<-c('1x', '2x', '3x')

GP1_125_comp<-cbind(t_GP1_125_summary_4, t_GP1_125_summary_8, t_GP1_125_summary_12)
GP1_125_comp<-GP1_125_comp[-1:-3,]
rownames(GP1_125_comp)<-c('Median Bias',
                        'Bias Q75',
                        'Bias Q25',
                        'Mean Bias',
                        'SD Bias',
                        'Maximum Deviation',
                        'Minimum Deviation',
                        'Mean HR (BPM)',
                        'SD HR (BPM)',
                        'Maximum HR (BPM)',
                        'Minimum HR (BPM)',
                        'Median R-hat',
                        'Mean R-hat',
                        'R-hats < 1.1',
                        'Avg Impute Total (secs)',
                        'Median runtime (mins)',
                        'Mean runtime (mins)',
                        'Maximum runtime (mins)',
                        'Minimum runtime (mins)')

colnames(GP1_125_comp)<- paste(rep(c('1x', '2x', '3x'), 3), 'Window')
stargazer::stargazer(GP1_125_comp,
                     summary = FALSE,
                     rownames = TRUE,
                     out = paste0(final.graphics, '/test.html'))
```

```

GP1_250_summary_4<-GP1_summary[GP1_summary$impute.Hz.vec==250 & GP1_summary$GP.Hz.vec==4,]
t_GP1_250_summary_4<-t(GP1_250_summary_4)
colnames(t_GP1_250_summary_4)<-c('1x', '2x', '3x')

GP1_250_summary_8<-GP1_summary[GP1_summary$impute.Hz.vec==250 & GP1_summary$GP.Hz.vec==8,]
t_GP1_250_summary_8<-t(GP1_250_summary_8)
colnames(t_GP1_250_summary_8)<-c('1x', '2x', '3x')

GP1_250_summary_12<-GP1_summary[GP1_summary$impute.Hz.vec==250 & GP1_summary$GP.Hz.vec==12,]
t_GP1_250_summary_12<-t(GP1_250_summary_12)
colnames(t_GP1_250_summary_12)<-c('1x', '2x', '3x')

GP1_250_comp<-cbind(t_GP1_250_summary_4, t_GP1_250_summary_8, t_GP1_250_summary_12)
GP1_250_comp<-GP1_250_comp[-1:-3,]
rownames(GP1_250_comp)<-c('Median Bias',
                          'Bias Q75',
                          'Bias Q25',
                          'Mean Bias',
                          'SD Bias',
                          'Maximum Deviation',
                          'Minimum Deviation',
                          'Mean HR (BPM)',
                          'SD HR (BPM)',
                          'Maximum HR (BPM)',
                          'Minimum HR (BPM)',
                          'Median R-hat',
                          'Mean R-hat',
                          'R-hats < 1.1',
                          'Avg Impute Total (secs)',
                          'Median runtime (mins)',
                          'Mean runtime (mins)',
                          'Maximum runtime (mins)',
                          'Minimum runtime (mins)')

colnames(GP1_250_comp)<- paste(rep(c('1x', '2x', '3x'), 3), 'Window')
stargazer::stargazer(GP1_250_comp,
                     summary = FALSE,
                     rownames = TRUE,
                     out = paste0(final.graphics, '/test.html'))

#####

GP2_summary <- dat.GP2 %>%
  group_by(impute.Hz.vec, impute.fac.vec, GP.Hz.vec)%>%
  summarize(Med_bias = median(bias_rmssd15),
            Q75_bias = quantile(bias_rmssd15, .75),
            Q25_bias = quantile(bias_rmssd15, .25),
            Mean_bias = mean(bias_rmssd15),
            SD_bias = sd(bias_rmssd15),
            Max_bias = max(bias_rmssd15),
            Min_bias = min(bias_rmssd15),
            Mean_HR = mean(HR.mean.vec)*60,
            SD_HR = sd(HR.mean.vec)*60,
            Max_HR = max(HR.mean.vec)*60,
            Min_HR = min(HR.mean.vec)*60,
            Med_Rhat = median(HR.Rhat.vec),
            Mean_Rhat = mean(HR.Rhat.vec),
            Prop_Rhat_1.1 = length(HR.Rhat.vec[HR.Rhat.vec<1.1])/40*100,
            Mean_tot.time = mean(impute.tot),
            Med_runtime = median(runtime),
            Mean_runtime = mean(runtime),
            Max_runtime = max(runtime),
            Min_runtime = min(runtime))

GP2_125_summary_4<-GP2_summary[GP2_summary$impute.Hz.vec==125 & GP2_summary$GP.Hz.vec==4,]
t_GP2_125_summary_4<-t(GP2_125_summary_4)

```

```

colnames(t_GP2_125_summary_4)<-c('1x', '2x', '3x')

GP2_125_summary_8<-GP2_summary[GP2_summary$impute.Hz.vec==125 & GP2_summary$GP.Hz.vec==8,]
t_GP2_125_summary_8<-t(GP2_125_summary_8)
colnames(t_GP2_125_summary_8)<-c('1x', '2x', '3x')

GP2_125_summary_12<-GP2_summary[GP2_summary$impute.Hz.vec==125 & GP2_summary$GP.Hz.vec==12,]
t_GP2_125_summary_12<-t(GP2_125_summary_12)
colnames(t_GP2_125_summary_12)<-c('1x', '2x', '3x')

GP2_125_comp<-cbind(t_GP2_125_summary_4, t_GP2_125_summary_8, t_GP2_125_summary_12)
GP2_125_comp<-GP2_125_comp[-1:-3,]
rownames(GP2_125_comp)<-c('Median Bias',
                          'Bias Q75',
                          'Bias Q25',
                          'Mean Bias',
                          'SD Bias',
                          'Maximum Deviation',
                          'Minimum Deviation',
                          'Mean HR (BPM)',
                          'SD HR (BPM)',
                          'Maximum HR (BPM)',
                          'Minimum HR (BPM)',
                          'Median R-hat',
                          'Mean R-hat',
                          'R-hats < 1.1',
                          'Avg Impute Total (secs)',
                          'Median runtime (mins)',
                          'Mean runtime (mins)',
                          'Maximum runtime (mins)',
                          'Minimum runtime (mins)')

colnames(GP2_125_comp)<- paste(rep(c('1x', '2x', '3x'), 3), 'Window')
stargazer::stargazer(GP2_125_comp,
                     summary = FALSE,
                     rownames = TRUE,
                     out = paste0('final.graphics', '/test.html'))

GP2_250_summary_4<-GP2_summary[GP2_summary$impute.Hz.vec==250 & GP2_summary$GP.Hz.vec==4,]
t_GP2_250_summary_4<-t(GP2_250_summary_4)
colnames(t_GP2_250_summary_4)<-c('1x', '2x', '3x')

GP2_250_summary_8<-GP2_summary[GP2_summary$impute.Hz.vec==250 & GP2_summary$GP.Hz.vec==8,]
t_GP2_250_summary_8<-t(GP2_250_summary_8)
colnames(t_GP2_250_summary_8)<-c('1x', '2x', '3x')

GP2_250_summary_12<-GP2_summary[GP2_summary$impute.Hz.vec==250 & GP2_summary$GP.Hz.vec==12,]
t_GP2_250_summary_12<-t(GP2_250_summary_12)
colnames(t_GP2_250_summary_12)<-c('1x', '2x', '3x')

GP2_250_comp<-cbind(t_GP2_250_summary_4, t_GP2_250_summary_8, t_GP2_250_summary_12)
GP2_250_comp<-GP2_250_comp[-1:-3,]
rownames(GP2_250_comp)<-c('Median Bias',
                          'Bias Q75',
                          'Bias Q25',
                          'Mean Bias',
                          'SD Bias',
                          'Maximum Deviation',
                          'Minimum Deviation',
                          'Mean HR (BPM)',
                          'SD HR (BPM)',
                          'Maximum HR (BPM)',
                          'Minimum HR (BPM)',
                          'Median R-hat',
                          'Mean R-hat',
                          'R-hats < 1.1',
                          'Avg Impute Total (secs)',
                          'Median runtime (mins)',

```

```

        'Mean runtime (mins)',
        'Maximum runtime (mins)',
        'Minimum runtime (mins)')

colnames(GP2_250_comp)<- paste(rep(c('1x', '2x', '3x'), 3), 'Window')
stargazer::stargazer(GP2_250_comp,
                     summary = FALSE,
                     rownames = TRUE,
                     out = paste0(final.graphics, '/test.html'))

#####

GP3_summary <- dat.GP3 %>%
  group_by(impute.Hz.vec, impute.fac.vec, GP.Hz.vec)%>%
  summarize(Med_bias = median(bias_rmssd15),
            Q75_bias = quantile(bias_rmssd15, .75),
            Q25_bias = quantile(bias_rmssd15, .25),
            Mean_bias = mean(bias_rmssd15),
            SD_bias = sd(bias_rmssd15),
            Max_bias = max(bias_rmssd15),
            Min_bias = min(bias_rmssd15),
            Mean_HR = mean(HR.mean.vec)*60,
            SD_HR = sd(HR.mean.vec)*60,
            Max_HR = max(HR.mean.vec)*60,
            Min_HR = min(HR.mean.vec)*60,
            Med_Rhat = median(HR.Rhat.vec),
            Mean_Rhat = mean(HR.Rhat.vec),
            Prop_Rhat_1.1 = length(HR.Rhat.vec[HR.Rhat.vec<1.1])/40*100,
            Mean_tot.time = mean(impute.tot),
            Med_runtime = median(runtime),
            Mean_runtime = mean(runtime),
            Max_runtime = max(runtime),
            Min_runtime = min(runtime))

GP3_125_summary_4<-GP3_summary[GP3_summary$impute.Hz.vec==125 & GP3_summary$GP.Hz.vec==4,]
t_GP3_125_summary_4<-t(GP3_125_summary_4)
colnames(t_GP3_125_summary_4)<-c('1x', '2x', '3x')

GP3_125_summary_8<-GP3_summary[GP3_summary$impute.Hz.vec==125 & GP3_summary$GP.Hz.vec==8,]
t_GP3_125_summary_8<-t(GP3_125_summary_8)
colnames(t_GP3_125_summary_8)<-c('1x', '2x', '3x')

GP3_125_summary_12<-GP3_summary[GP3_summary$impute.Hz.vec==125 & GP3_summary$GP.Hz.vec==12,]
t_GP3_125_summary_12<-t(GP3_125_summary_12)
colnames(t_GP3_125_summary_12)<-c('1x', '2x', '3x')

GP3_125_comp<-cbind(t_GP3_125_summary_4, t_GP3_125_summary_8, t_GP3_125_summary_12)
GP3_125_comp<-GP3_125_comp[-1:-3,]
rownames(GP3_125_comp)<-c('Median Bias',
                        'Bias Q75',
                        'Bias Q25',
                        'Mean Bias',
                        'SD Bias',
                        'Maximum Deviation',
                        'Minimum Deviation',
                        'Mean HR (BPM)',
                        'SD HR (BPM)',
                        'Maximum HR (BPM)',
                        'Minimum HR (BPM)',
                        'Median R-hat',
                        'Mean R-hat',
                        'R-hats < 1.1',
                        'Avg Impute Total (secs)',
                        'Median runtime (mins)',
                        'Mean runtime (mins)',
                        'Maximum runtime (mins)',
                        'Minimum runtime (mins)')

colnames(GP3_125_comp)<- paste(rep(c('1x', '2x', '3x'), 3), 'Window')

```

```

stargazer::stargazer(GP3_125_comp,
  summary = FALSE,
  rownames = TRUE,
  out = paste0(final.graphics, '/test.html'))

GP3_250_summary_4<-GP3_summary[GP3_summary$impute.Hz.vec==250 & GP3_summary$GP.Hz.vec==4,]
t_GP3_250_summary_4<-t(GP3_250_summary_4)
colnames(t_GP3_250_summary_4)<-c('1x', '2x', '3x')

GP3_250_summary_8<-GP3_summary[GP3_summary$impute.Hz.vec==250 & GP3_summary$GP.Hz.vec==8,]
t_GP3_250_summary_8<-t(GP3_250_summary_8)
colnames(t_GP3_250_summary_8)<-c('1x', '2x', '3x')

GP3_250_summary_12<-GP3_summary[GP3_summary$impute.Hz.vec==250 & GP3_summary$GP.Hz.vec==12,]
t_GP3_250_summary_12<-t(GP3_250_summary_12)
colnames(t_GP3_250_summary_12)<-c('1x', '2x', '3x')

GP3_250_comp<-cbind(t_GP3_250_summary_4, t_GP3_250_summary_8, t_GP3_250_summary_12)
GP3_250_comp<-GP3_250_comp[-1:-3,]
rownames(GP3_250_comp)<-c('Median Bias',
  'Bias Q75',
  'Bias Q25',
  'Mean Bias',
  'SD Bias',
  'Maximum Deviation',
  'Minimum Deviation',
  'Mean HR (BPM)',
  'SD HR (BPM)',
  'Maximum HR (BPM)',
  'Minimum HR (BPM)',
  'Median R-hat',
  'Mean R-hat',
  'R-hats < 1.1',
  'Avg Impute Total (secs)',
  'Median runtime (mins)',
  'Mean runtime (mins)',
  'Maximum runtime (mins)',
  'Minimum runtime (mins)')

colnames(GP3_250_comp)<- paste(rep(c('1x', '2x', '3x'), 3), 'Window')
stargazer::stargazer(GP3_250_comp,
  summary = FALSE,
  rownames = TRUE,
  out = paste0(final.graphics, '/test.html'))

```

Appendix B.2: The following code is an example of the code used create a series of Bayesian multilevel models in which laboratory task predicted child physiology. Three versions of this model were analyzed, one for RSA, one for RMSSD, and one for HP. Imputation code appears in the top portion of the code block. The `bayes.to.txt()` function is a custom function the code for which can be found in

```

fml<-formula(RSA +pEdits ~ 1 + Task + CHILD.sex.F + AGE01 + BIQ_avg_MR85 + (1|ID))

imp<-panImpute(data = RSA.long.comp,
  formula = fml,
  n.burn = 50000,
  n.iter = 5000,
  m = 20)
plot(imp) #imputation largely successful - upping the burn-in phase

#Extracting Imputed Datasets
RSA.mitml<-mitmlComplete(imp)

```

```

RSA.list<-list()
for(i in 1:20){
  RSA.list[[i]]<-RSA.mitml[[i]]
  RSA.list[[i]]$V1<-ifelse(RSA.list[[i]]$Task=='Video1', 1, 0)
  RSA.list[[i]]$Clwn<-ifelse(RSA.list[[i]]$Task=='Clown', 1, 0)
  RSA.list[[i]]$V2<-ifelse(RSA.list[[i]]$Task=='Video2', 1, 0)
  RSA.list[[i]]$Kids<-ifelse(RSA.list[[i]]$Task=='Kids', 1, 0)
  RSA.list[[i]]$Intro<-ifelse(RSA.list[[i]]$Task=='Intro', 1, 0)
  RSA.list[[i]]$V3<-ifelse(RSA.list[[i]]$Task=='Video3', 1, 0)
}

RSA.MLM.fit.V1.crt1<-brm_multiple(formula = RSA ~ 1 + Clwn + V2 + Kids + Intro + V3 +
  c.AGE01 + CHILD.sex.F + pEdits +
  (1|ID),
  data = RSA.list,
  prior = c(set_prior('normal(0,2)', class='sd'),
            set_prior('normal(6.33, 1)', class='Intercept'),
            set_prior('normal(0,2)', class='b')),
  warmup = 4000,
  iter = 8000,
  thin = 2,
  chains = 4,
  control = list(adapt_delta=.80,
                 max_treedepth=10))

bayes.to.txt(model = RSA.MLM.fit.V1.crt1,
  out.folder = out.folder,
  file = 'RSA_Video1_intercept_ctrl_pEdits',
  DF = RSA.list[[sample(1:length(RSA.list), size = 1)]],
  tot.pars = 10,
  impute = T)

```

Appendix B.3: Code for the custom `bayes.to.txt()` function used to simplify

```

#Customized Summary function
bayes.to.txt<-function(model=NULL,
  out.folder=NULL,
  file=NULL,
  DF=NULL,
  tot.pars=3,
  impute=TRUE){
  require(parallel)
  require(benchmarkme)
  sink(paste0(out.folder, '/', file, '.txt'))
  cat('System Information:')
  cat('\n=====')
  cat(paste0('\nProcessor:', '\t\t\t', benchmarkme::get_cpu()$model_name))
  cat(paste0('\nNumber of Threads:', '\t\t\t', detectCores(logical=T)))
  cat(paste0('\nRAM:', '\t\t\t\t\t', paste(round(benchmarkme::get_ram()/1073741824), 'GB')))
  cat('\n=====')

  cat('\n\nModel Information:')
  cat('\n=====')
  cat('\n\nFormula (lme4 syntax):')
  cat('\n-----\n')
  print(model$formula)
  cat('\n\nPriors:')
  cat('\n-----\n')
  print(prior_summary(model))
  cat('\n\nStan Code:')
  cat('\n-----\n')
  print(make_stancode(model$formula, data=DF, family=model$family$family))
  cat('\n=====')

  cat('\n\nStan Arguments:')

```

```

cat('\n=====')
cat(paste0('\nAdapt Delta:', '\t\t\t', model$fit@stan_args[[1]]$control$adapt_delta))
cat(paste0('\nMaximum Tree Depth:', '\t\t\t', model$fit@stan_args[[1]]$control$max_treedepth))
cat('\n=====')

cat('\n\nVariance Explained:')
cat('\n=====')
cat('\n\nResidual Variance:')
cat('\n-----\n')
print(sjstats::icc(model, posterior = T), prob = .95, digits=3)
cat('\n\nBayesian R-squared (overall variance explained):')
cat('\n-----\n')
print(bayes_R2(model))
cat('\n=====')

cat('\n\nModel Summary:')
cat('\n=====')
print(summary(model))

if(impute==0){
  cat('WARNING! If using multiply imputed datasets ignore R=hat')
  cat('\nUse the code below to obtain interpretable R=hat values for each data set')
  cat('\nround(modelname$rhats[,1:tot.pars], 3)')
  cat('\nwhere "tot.pars" is the total number of model parameters to return (usually fixed
effects)')
}
else{
  cat('\n\nPotential Scale Reduction Factor for Each Imputed Dataset:')

cat('\n=====')
  print(round(model$rhats[,1:tot.pars], 3))

cat('\n=====')
}
sink()
}

```

Appendix B.4: Example model code for a laboratory-based physiological measure predicting child social behaviors during periods of free play in their preschool classrooms. This basic model was repeated multiple times - for tonic measures and dynamic measures of RSA.

```

#Tonic RSA predicting Reticence
Tonic_RSA.Ret<-brm_multiple(Ret.tot | trials(Obs.tot) ~ c.Age + CHILD.sex.F +
  Tonic_RSA+(1|ID),
  prior = c(set_prior('normal(0,2)', class='sd'),
    set_prior('normal(0,2)', class='b')),
  data = POS.RSA.list,
  family = binomial())
summary(Tonic_RSA.Ret)

```

Appendix C: Code Used to Generate Figures

Appendix C.1: Code used to generate figure 2.4

```
setwd('/home/mbarsted/Dropbox/Dissertation/IBI_pilot')
dat<-read.table('tap_lap2.txt', header=F, sep='\t', skip=11)
dat<-dat[,-4]
dat<-dat[1:300000, 1]

#Generating IBI file from raw heart rate data:
spec.dat<-pspectrum(DF.temp$IBI, x.frqsamp = 4)

DF.temp2<-data.frame(Frequency=spec.dat$freq[spec.dat$freq<.5],
                    Power=spec.dat$spec[spec.dat$freq<.5])

g1<-ggplot()+
  geom_rect(data=data.frame(xmin=0,
                           xmax=.15,
                           ymin=-Inf,
                           ymax=Inf),
           aes(xmin=xmin, xmax=xmax, ymin=ymin, ymax=ymax),
           fill='blue',
           alpha=.5)+
  geom_rect(data=data.frame(xmin=.15,
                           xmax=.40,
                           ymin=-Inf,
                           ymax=Inf),
           aes(xmin=xmin, xmax=xmax, ymin=ymin, ymax=ymax),
           fill='red',
           alpha=.5)+
  geom_rect(data=data.frame(xmin=.40,
                           xmax=Inf,
                           ymin=-Inf,
                           ymax=Inf),
           aes(xmin=xmin, xmax=xmax, ymin=ymin, ymax=ymax),
           fill='gray',
           alpha=.5)+
  geom_line(data=DF.temp2, aes(x=Frequency, y=Power))+
  ylab('Power Spectral Density (PSD)')+xlab('Frequency (Hz)')+
  annotate(geom='text', x=.075, y = .0035, label='ULF-HRV & LF-HRV')+
  annotate(geom='text', x=.275, y = .03, label='HF-HRV')

png(paste0(paste0(wd, '/Figure_2_X_Spectral Analysis.png')),
    res=300, units = 'in', width=8, height = 5)
g1
dev.off()
```

Appendix C.2: Code used to generate Figure 3.1

```
png(paste0('/home/mbarsted/Dropbox/Dissertation/CH1-3_Graphics',
          '/Filtering_MS.png'), res=300, units='in', height = 5, width=8)
par(mfrow=c(3,1))
plot(y=dat.noise, x=1:2500/250, type='l',
     main='PPG Signal with Random Noise',
     ylab='Volts',
     xlab='Time (s)')

plot(y=dat.med, x=1:2500/250, type='l',
     main='PPG Signal after Median Filter',
     ylab='Volts',
     xlab='Time (s)')

plot(y=dat.smooth$y, x=1:2500/250, type='l',
     main='PPG Signal after Smoothing Splines',
     ylab='Volts',
```



```
xlab='Time (s)')
dev.off()
```

Appendix C.3: Code used to generate figure 3.2

```
set.seed(430093)
Sample_rate<-1000
time<-seq(0, 10, by=1/Sample_rate) #putting time into seconds
pi<-3.14159265359
n.sims<-1000

#Generating Numerous Samples
x<-matrix(0, length(time), n.sims)
amps<-runif(n.sims, min = .5, max = 3)

Hfreq<-vector()
for(j in 1:n.sims){
  Hfreq.temp<-rnorm(1, 80/60, sd=.25)
  x[,j]<-amps[j]*sin(2*pi*time*Hfreq.temp*j)
  Hfreq<-c(Hfreq, Hfreq.temp)
}

#Using differenced values to identify peaks and troughs
shape <- c(NA, NA,diff(sign(diff(x[,1], na.pad = T))))

#Extracting data for plotting purposes
Sim_wav.diff<-diff(x[,1], na.pad=T)
DF<-data.frame(time=time,
               Sim_wv=x[,1],
               Sim_wav.diff=c(NA, as.vector(as.numeric(scale(Sim_wav.diff)))))
DF$Min_Max<-rep(NA, length(DF[,1]))
DF$Min_Max<-ifelse(shape<0, 'Maximum', DF$Min_Max)
DF$Min_Max<-ifelse(shape>0, 'Minimum', DF$Min_Max)

DF2<-DF[!is.na(DF$Min_Max),]

#Plotting simulated and differenced time series overlays
g1<-ggplot()+
  geom_line(data=DF, aes(y=Sim_wv, x=time,
                       lty=c('Original Simulated Signal')), lwd=1.5)+
  geom_point(data=DF2, aes(x=time, y=Sim_wv, color=Min_Max, group=Min_Max,
                          shape=Min_Max), size=4)+
  geom_line(data=DF, aes(y=Sim_wav.diff, x=time, lty='Differenced Signal'),
           lwd=1.5)+
  geom_hline(yintercept = 0)+
  geom_point(data=DF2, aes(x=time, y=Sim_wav.diff, color=Min_Max,
                          group=Min_Max, shape=Min_Max), size=4)+
  guides(color=guide_legend(title="Minimum/Maximum"),
         shape=guide_legend(title="Minimum/Maximum"),
         linetype=guide_legend(title="Waveform"))+
  ggtitle('Graphic Representation of Peak Detection Algorithm Outputs')+
  ylab('Voltage')+xlab('Time (s)')

png(paste0(paste0(wd, '/Figure_3_1_Peak Detection.png')), res=300,
    units = 'in', width=8, height = 5)
g1
dev.off()
```

Appendix C.4: Code used to generate Figure 3.3

```
set.seed(122411)
Sample_rate<-1000
```

```

time<-seq(0, 10, by=1/Sample_rate) #putting time into seconds
pi<-3.14159265359
n.sims<-1000

#simulating data
x<-matrix(0, length(time), n.sims)
amps<-runif(n.sims, min = .5, max = 3)

Hfreq<-vector()
for(j in 1:n.sims){
  Hfreq.temp<-rnorm(1, 80/60, sd=.25)
  x[,j]<-amps[j]*sin(2*pi*time*Hfreq.temp*j)
  Hfreq<-c(Hfreq, Hfreq.temp)
}

PPG.temp<-data.frame(PPG=x[,1], Time=time)

IBI<-iter.IBI(x=x[,1], ds=Sample_rate, iter.length = 50)
g1<-ggplot()+
  geom_line(data=PPG.temp, aes(y=PPG, x=Time))+
  geom_vline(data=IBI, aes(xintercept=Time), lty='dashed', color='gray70')+
  xlab('Time (s)')

png(paste0(paste0(wd, '/Figure_3_2_Peak_Detection.png')), res=300,
  units = 'in', width=8, height = 5)
g1
dev.off()

```

Appendix C.5: Code used to generate figure 3.11

```

setwd('/home/mbarsted/Dropbox/Dissertation/IBI_pilot')
dat<-read.table('tap_lap2.txt', header=F, sep='\t', skip=11)
dat<-dat[,-4]
dat<-dat[1:300000, 1]

#Plotting for figures and presentation:
DF.plot<-data.frame(Time = (1:300000)/2000,
  PPG = dat)
DF.plot2$PPG[140000:160000]<-NA

#Targeting the data
g1<-ggplot(data=DF.plot, aes(x=Time, y=PPG))+
  geom_rect(data=data.frame(xmin=140000/2000,
    xmax=160000/2000,
    ymin=-Inf,
    ymax=Inf),
    aes(xmin=xmin, xmax=xmax, ymin=ymin, ymax=ymax),
    fill='blue', alpha=.5, inherit.aes = F)+
  geom_line(data=DF.plot, aes(x=Time, y=PPG), inherit.aes = F)

png(paste0('/home/mbarsted/Dropbox/Dissertation/CH1-3_Graphics',
  '/Base_graphic.png'), res=300, units='in', height = 5, width=8)
g1
dev.off()

#Removing the data
g1<-ggplot(data=DF.plot, aes(x=Time, y=PPG))+
  geom_rect(data=data.frame(xmin=140000/2000,
    xmax=160000/2000,
    ymin=-Inf,
    ymax=Inf),
    aes(xmin=xmin, xmax=xmax, ymin=ymin, ymax=ymax),
    fill='blue', alpha=.5, inherit.aes = F)+
  geom_line(data=DF.plot2, aes(x=Time, y=PPG), inherit.aes = F)

```

```

png(paste0('/home/mbarsted/Dropbox/Dissertation/CH1-3_Graphics',
           '/Base_graphic_dataRemoved.png'), res=300, units='in', height = 5, width=8)
g1
dev.off()

#Targeting Good data
g1<-ggplot()+
  geom_rect(data=data.frame(xmin=100000/2000,
                           xmax=139999/2000,
                           ymin=-Inf, #Depiction of Imputation Run results
                           ymax=Inf),

g1<-ggplot()+
  geom_rect(data=data.frame(xmin=100000/2000,
                           xmax=139999/2000,
                           ymin=-Inf,
                           ymax=Inf),
            aes(xmin=xmin, xmax=xmax, ymin=ymin, ymax=ymax),
            fill='green',
            alpha=.5)+
  geom_rect(data=data.frame(xmin=160001/2000,
                           xmax=200000/2000,
                           ymin=-Inf,
                           ymax=Inf),
            aes(xmin=xmin, xmax=xmax, ymin=ymin, ymax=ymax),
            fill='green',
            alpha=.5)+
  geom_rect(data=data.frame(xmin=140000/2000,
                           xmax=160000/2000,
                           ymin=-Inf,
                           ymax=Inf),
            aes(xmin=xmin, xmax=xmax, ymin=ymin, ymax=ymax),
            fill='blue', alpha=.5, inherit.aes = F)+
  geom_line(data=DF.plot, aes(x=Time, y=PPG))
g1
  ymax=Inf),
  aes(xmin=xmin, xmax=xmax, ymin=ymin, ymax=ymax),
  fill='green',
  alpha=.5)+
  geom_rect(data=data.frame(xmin=160001/2000,
                           xmax=200000/2000,
                           ymin=-Inf,
                           ymax=Inf),
            aes(xmin=xmin, xmax=xmax, ymin=ymin, ymax=ymax),
            fill='green',
            alpha=.5)+
  geom_line(data=DF.plot2, aes(x=Time, y=PPG))
g1

png(paste0('/home/mbarsted/Dropbox/Dissertation/CH1-3_Graphics',
           '/Data_for_model.png'), res=300, units='in', height = 5, width=8)
g1
dev.off()

#Depiction of Imputation Run results
g1<-ggplot()+
  geom_rect(data=data.frame(xmin=100000/2000,
                           xmax=139999/2000,
                           ymin=-Inf,
                           ymax=Inf),
            aes(xmin=xmin, xmax=xmax, ymin=ymin, ymax=ymax),
            fill='green',
            alpha=.5)+
  geom_rect(data=data.frame(xmin=160001/2000,
                           xmax=200000/2000,
                           ymin=-Inf,
                           ymax=Inf),
            aes(xmin=xmin, xmax=xmax, ymin=ymin, ymax=ymax),
            fill='green',
            alpha=.5)+
  geom_rect(data=data.frame(xmin=140000/2000,

```

```

                                xmax=160000/2000,
                                ymin=-Inf,
                                ymax=Inf),
                                aes(xmin=xmin, xmax=xmax, ymin=ymin, ymax=ymax),
                                fill='blue', alpha=.5, inherit.aes = F)+
geom_line(data=DF.plot, aes(x=Time, y=PPG))
g1

png(paste0('/home/mbarsted/Dropbox/Dissertation/CH1-3_Graphics',
           '/Graphic_MS.png'), res=300, units='in', height = 5, width=8)
g1
dev.off()

#Just the imputed section:
g1<-ggplot()+
  geom_rect(data=data.frame(xmin=140000/2000,
                            xmax=160000/2000,
                            ymin=-Inf,
                            ymax=Inf),
            aes(xmin=xmin, xmax=xmax, ymin=ymin, ymax=ymax),
            fill='blue', alpha=.5, inherit.aes = F)+
  geom_line(data=DF.plot[140000:160000,], aes(x=Time, y=PPG))
g1

png(paste0('/home/mbarsted/Dropbox/Dissertation/CH1-3_Graphics',
           '/Cut_out.png'), res=300, units='in', height = 5, width=8)
g1
dev.off()

IBI<-iter.IBI(dat, ds=2000, iter.length = 50)
max(IBI$Time)-min(IBI$Time)

time.pred<-seq(0, 297.75, length.out=298*4)

dat<-list(N1=length(IBI$IBI),
          N2=length(time.pred),
          Xp=time.pred,
          X=IBI$Time,
          Y=IBI$IBI
          )

pars.to.monitor<-c('ULF', 'LF', 'HF', 'Ypred', paste0('a',1:4), paste0('r',1:7))

fit.stan<-stan(
  file='/home/mbarsted/Dropbox/Dissertation/R_Ch1-3/Stan_code/IBI_Data_Interpolation.stan',
  data = dat,
  warmup = 1000,
  iter = 1500,
  refresh=5,
  chains = 6,
  pars = pars.to.monitor,
  control = list(adapt_delta = .8,
                 max_treedepth = 10)
)

posterior<-extract(fit.stan, pars=pars.to.monitor[-4])
posterior<-as.matrix(fit.stan)

mcmc_areas(posterior, pars = pars.to.monitor[1:3])
summary(fit.stan, pars=pars.to.monitor[-4])

y_pred<-extract(fit.stan, 'Ypred')
time.new<-(1:1192)/4

plot(time.new, colMeans(y_pred$Ypred))
lines(IBI2$Time, IBI2$IBI, col='red')

DF.temp<-data.frame(Time=time.new,
                    IBI=colMeans(y_pred$Ypred))

```

```

#Depiction of Imputation Run results
setwd('/home/mbarsted/Dropbox/Dissertation/IBI_pilot')
dat<-read.table('tap_lap2.txt', header=F, sep='\t', skip=11)
dat<-dat[,-4]
dat<-dat[1:300000, 1]

DF.plot<-data.frame(Time = (1:300000)/2000,
                    PPG = dat)

g1<-ggplot()+
  geom_rect(data=data.frame(xmin=100000/2000,
                           xmax=139999/2000,
                           ymin=-Inf,
                           ymax=Inf),
            aes(xmin=xmin, xmax=xmax, ymin=ymin, ymax=ymax),
            fill='green',
            alpha=.5)+
  geom_rect(data=data.frame(xmin=160001/2000,
                           xmax=200000/2000,
                           ymin=-Inf,
                           ymax=Inf),
            aes(xmin=xmin, xmax=xmax, ymin=ymin, ymax=ymax),
            fill='green',
            alpha=.5)+
  geom_rect(data=data.frame(xmin=140000/2000,
                           xmax=160000/2000,
                           ymin=-Inf,
                           ymax=Inf),
            aes(xmin=xmin, xmax=xmax, ymin=ymin, ymax=ymax),
            fill='blue', alpha=.5, inherit.aes = F)+
  geom_line(data=DF.plot, aes(x=Time, y=PPG))

png(paste0('/home/mbarsted/Dropbox/Dissertation/CH1-3_Graphics',
          '/Graphic_MS.png'), res=300, units='in', height = 5, width=8)

g1
dev.off()

```

Appendix C.6: Code used to generate figure 3.12

```

#Initial Exploratory Analyses for RSA Data:
RSA.wide<-dat.main[c('File',
                   colnames(dat.main)[grep('RSA15', colnames(dat.main))])]

png(paste0(EDA.folder, '/RSA_pairs_panels.png'), units = 'in',
    height = 8, width = 8, res = 300)
psych::pairs.panels(RSA.wide[colnames(RSA.wide)!='File'])
dev.off()

#RSA by task
df.m<-reshape2::melt(RSA.wide[,2:7])
df.m$variable<-as.character(df.m$variable)

#Recoding variable names to align with the task names
df.m$variable[df.m$variable=='RSA15_Video1']<-'Video 1'
df.m$variable[df.m$variable=='RSA15_Video2']<-'Video 2'
df.m$variable[df.m$variable=='RSA15_Video3']<-'Video 3'
df.m$variable[df.m$variable=='RSA15_Clown']<-'Clown'
df.m$variable[df.m$variable=='RSA15_Kids']<-'Kids'
df.m$variable[df.m$variable=='RSA15_Intro']<-'Introduction'

g1<-ggplot(data = df.m, aes(y = value, x = variable, fill=variable)) +
  geom_flat_violin(position = position_nudge(x = .2, y = 0), alpha = .8) +
  geom_point(aes(y = value, color = variable),
             position = position_jitter(width = .15),
             size = .5, alpha = 0.8) +

```

```

geom_boxplot(width = .1, guides = FALSE, outlier.shape = NA, alpha = 0.5) +
expand_limits(x = 5.25) +
scale_color_brewer(palette = "Dark2") +
scale_fill_brewer(palette = "Dark2") +
theme_bw() +
raincloud_theme+
xlab('')+ylab('RSA')+
guides(fill=FALSE, color=FALSE)+
ggtitle('Distribution of Respiratory Sinus Arrhythmia as a Function of Task')

png(paste0(EDA.folder, '/RSA_raincloud.png'), units = 'in',
    height = 8, width = 8, res = 300)
g1
dev.off()

```

Appendix C.7: Code used to generate figure 3.13

```

#Initial Exploratory Analyses for HP Data:
HP.wide<-dat.main[c('File',
                   colnames(dat.main)[grep('HP', colnames(dat.main))])]

png(paste0(EDA.folder, '/HP_pairs_panels.png'), units = 'in',
    height = 8, width = 8, res = 300)
psych::pairs.panels(HP.wide[colnames(HP.wide)!='File'])
dev.off()

#HP by task
df.m<-reshape2::melt(HP.wide[,2:7])
df.m$variable<-as.character(df.m$variable)

#Recoding variable names to align with the task names
df.m$variable[df.m$variable=='HP_Video1']<-'Video 1'
df.m$variable[df.m$variable=='HP_Video2']<-'Video 2'
df.m$variable[df.m$variable=='HP_Video3']<-'Video 3'
df.m$variable[df.m$variable=='HP_Clown']<-'Clown'
df.m$variable[df.m$variable=='HP_Kids']<-'Kids'
df.m$variable[df.m$variable=='HP_Intro']<-'Introduction'

g1<-ggplot(data = df.m, aes(y = value, x = variable, fill=variable)) +
  geom_flat_violin(position = position_nudge(x = .2, y = 0), alpha = .8) +
  geom_point(aes(y = value, color = variable),
             position = position_jitter(width = .15),
             size = .5, alpha = 0.8) +
  geom_boxplot(width = .1, guides = FALSE, outlier.shape = NA, alpha = 0.5) +
  expand_limits(x = 5.25) +
  scale_color_brewer(palette = "Dark2") +
  scale_fill_brewer(palette = "Dark2") +
  theme_bw() +
  raincloud_theme+
  xlab('')+ylab('HP')+
  guides(fill=FALSE, color=FALSE)+
  ggtitle('Distribution of Heart Period as a Function of Task')

png(paste0(EDA.folder, '/HP_raincloud.png'), units = 'in',
    height = 8, width = 8, res = 300)
g1
dev.off()

```

Appendix C.8: Code used to generate figure 3.14

```

#Initial Exploratory Analyses for RMSSD Data:
RMSSD.wide<-dat.main[c('File',

```

```

colnames(dat.main)[grep('RMSSD', colnames(dat.main))]]

png(paste0(EDA.folder, '/RMSSD_pairs_panels.png'), units = 'in',
    height = 8, width = 8, res = 300)
psych::pairs.panels(RMSSD.wide[colnames(RMSSD.wide)!='File'])
dev.off()

#RMSSD by task
df.m<-reshape2::melt(RMSSD.wide[,2:7])
df.m$variable<-as.character(df.m$variable)

#Recoding variable names to align with the task names
df.m$variable[df.m$variable=='RMSSD_Video1']<-'Video 1'
df.m$variable[df.m$variable=='RMSSD_Video2']<-'Video 2'
df.m$variable[df.m$variable=='RMSSD_Video3']<-'Video 3'
df.m$variable[df.m$variable=='RMSSD_Clown']<-'Clown'
df.m$variable[df.m$variable=='RMSSD_Kids']<-'Kids'
df.m$variable[df.m$variable=='RMSSD_Intro']<-'Introduction'

g1<-ggplot(data = df.m, aes(y = value, x = variable, fill=variable)) +
  geom_flat_violin(position = position_nudge(x = .2, y = 0), alpha = .8) +
  geom_point(aes(y = value, color = variable),
             position = position_jitter(width = .15),
             size = .5, alpha = 0.8) +
  geom_boxplot(width = .1, guides = FALSE, outlier.shape = NA, alpha = 0.5) +
  expand_limits(x = 5.25) +
  scale_color_brewer(palette = "Dark2") +
  scale_fill_brewer(palette = "Dark2") +
  theme_bw() +
  raincloud_theme+
  xlab('')+ylab('RMSSD')+
  guides(fill=FALSE, color=FALSE)+
  ggtitle('Distribution of Heart Period as a Function of Task')

png(paste0(EDA.folder, '/RMSSD_raincloud.png'), units = 'in',
    height = 8, width = 8, res = 300)
g1
dev.off()

```

Appendix C.9: Code used to generate figure 3.15

```

#Getting a descriptive plot for POS variables
df.m<-reshape2::melt(POS.HP[,21:26])
df.m$variable<-as.character(df.m$variable)
df.m$variable[df.m$variable=='Const.prop']<-'Constructive-Exploratory'
df.m$variable[df.m$variable=='Func.prop']<-'Functional-Dramatic'
df.m$variable[df.m$variable=='Grp.prop']<-'Group Play'
df.m$variable[df.m$variable=='Par.prop']<-'Parallel Play'
df.m$variable[df.m$variable=='Ret.prop']<-'Reticence'
df.m$variable[df.m$variable=='Sol.prop']<-'Solitary Composite'

g1<-ggplot(data = df.m, aes(y = value, x = variable, fill=variable)) +
  geom_flat_violin(position = position_nudge(x = .2, y = 0), alpha = .8) +
  geom_point(aes(y = value, color = variable),
             position = position_jitter(width = .15),
             size = 1, alpha = 0.8) +
  geom_boxplot(width = .1, guides = FALSE, outlier.shape = NA, alpha = 0.5) +
  expand_limits(x = 3.25) +
  scale_color_brewer(palette = "Dark2") +
  scale_fill_brewer(palette = "Dark2") +
  theme_bw() +
  raincloud_theme+
  xlab('')+ylab('Proportion of 10-second Epochs')+
  guides(fill=FALSE, color=FALSE)+
  ggtitle('Proportion of Observed Time Children were Engaged in Each Play Behavior')

```

```
png(paste0(EDA.folder, '/POS_raincloud.png'), units = 'in',
    height = 8, width = 8, res = 300)
g1
dev.off()
```

Appendix C.10: Code used to generate figure 4.1

```
dat.HR.Rhat_1<-dat.GP1[c('impute.fac.vec',
                      'impute.Hz.vec',
                      'GP.Hz.vec',
                      'HR.Rhat.vec'
)]

#Getting GP1 prepped...
dat.HR.Rhat_1$GP_run<-rep('GP1 Replacement')
dat.HR.Rhat_1$impute.fac.vec<-as.character(dat.HR.Rhat_1$impute.fac.vec)
dat.HR.Rhat_1$impute.fac.vec[dat.HR.Rhat_1$impute.fac.vec==1]<- '1x Imputation Window Length'
dat.HR.Rhat_1$impute.fac.vec[dat.HR.Rhat_1$impute.fac.vec==2]<- '2x Imputation Window Length'
dat.HR.Rhat_1$impute.fac.vec[dat.HR.Rhat_1$impute.fac.vec==3]<- '3x Imputation Window Length'

#Cleaning up values
dat.HR.Rhat_1$impute.fac.vec<-as.character(dat.HR.Rhat_1$impute.fac.vec)
dat.HR.Rhat_1$impute.Hz.vec<-as.character(dat.HR.Rhat_1$impute.Hz.vec)
dat.HR.Rhat_1$GP.Hz.vec<-factor(dat.HR.Rhat_1$GP.Hz.vec)

=====
#Bringing in GP2 Data...
dat.HR.Rhat_2<-dat.GP2[c('impute.fac.vec',
                      'impute.Hz.vec',
                      'GP.Hz.vec',
                      'HR.Rhat.vec'
)]

#Getting GP2 prepped...
dat.HR.Rhat_2$GP_run<-rep('GP2 Replacement')
dat.HR.Rhat_2$impute.fac.vec<-as.character(dat.HR.Rhat_2$impute.fac.vec)
dat.HR.Rhat_2$impute.fac.vec[dat.HR.Rhat_2$impute.fac.vec==1]<- '1x Imputation Window Length'
dat.HR.Rhat_2$impute.fac.vec[dat.HR.Rhat_2$impute.fac.vec==2]<- '2x Imputation Window Length'
dat.HR.Rhat_2$impute.fac.vec[dat.HR.Rhat_2$impute.fac.vec==3]<- '3x Imputation Window Length'

#Cleaning up values
dat.HR.Rhat_2$impute.fac.vec<-as.character(dat.HR.Rhat_2$impute.fac.vec)
dat.HR.Rhat_2$impute.Hz.vec<-as.character(dat.HR.Rhat_2$impute.Hz.vec)
dat.HR.Rhat_2$GP.Hz.vec<-factor(dat.HR.Rhat_2$GP.Hz.vec)

=====
dat.HR.Rhat_3<-dat.GP3[c('impute.fac.vec',
                      'impute.Hz.vec',
                      'GP.Hz.vec',
                      'HR.Rhat.vec'
)]

#Getting GP3 prepped...
dat.HR.Rhat_3$GP_run<-rep('GP3 Replacement')
dat.HR.Rhat_3$impute.fac.vec<-as.character(dat.HR.Rhat_3$impute.fac.vec)
dat.HR.Rhat_3$impute.fac.vec[dat.HR.Rhat_3$impute.fac.vec==1]<- '1x Imputation Window Length'
dat.HR.Rhat_3$impute.fac.vec[dat.HR.Rhat_3$impute.fac.vec==2]<- '2x Imputation Window Length'
dat.HR.Rhat_3$impute.fac.vec[dat.HR.Rhat_3$impute.fac.vec==3]<- '3x Imputation Window Length'

#Cleaning up values
dat.HR.Rhat_3$impute.fac.vec<-as.character(dat.HR.Rhat_3$impute.fac.vec)
dat.HR.Rhat_3$impute.Hz.vec<-as.character(dat.HR.Rhat_3$impute.Hz.vec)
dat.HR.Rhat_3$GP.Hz.vec<-factor(dat.HR.Rhat_3$GP.Hz.vec)

df.m<-rbind(dat.HR.Rhat_1,
```



```

dat.HR.Rhat_2,
dat.HR.Rhat_3)

g125<-ggplot(data = df.m[df.m$impute.Hz.vec=='125',], aes(y = HR.Rhat.vec, x = GP_run)) +
  geom_point(data = df.m[df.m$impute.Hz.vec=='125',],
    aes(x = GP_run, color = GP.Hz.vec),
    position = position_jitter(width = .15),
    size = 1, alpha = 0.5) +
  geom_boxplot(data = df.m[df.m$impute.Hz.vec=='125',],
    aes(x = GP_run, y = HR.Rhat.vec, fill = GP.Hz.vec),
    width = .5, guides = FALSE, outlier.shape = NA, alpha = 0.5, inherit.aes = F) +
  #geom_point(data = df.m[df.m$variable!='GP1 Replacement' & df.m$impute.Hz.vec=='125',],
  # aes(x = variable, y = value, color = variable),
  # position = position_jitter(width = .15),
  # size = 1,
  # alpha = 0.5,
  # inherit.aes = F,
  # guides = F) +
  #expand_limits(x = 4.25) +
  coord_cartesian(ylim = c(0,625))+
  geom_hline(yintercept = 1.1, color='red', lty='dashed')+
  scale_color_brewer(palette = "Dark2") +
  scale_fill_brewer(palette = "Dark2", name = 'GP Model Hz') +
  theme_bw() +
  xlab('')+ylab('Model Run Time (mins)')+
  guides(color=FALSE)+
  ggtitle('Distributions of Heart Rate Parameter R-hat Values as a
    Function of Model and Modeling Conditions (125 Hz)')+
  theme(axis.text.x = element_text(angle = 30, hjust = 1))+
  facet_wrap(~impute.fac.vec)

#Looks Like a GP sample rate of 12 Hz with a factor of 2 window on either side is best...
#This may be unique the the GP 1 model specification...
#Will need to investigate more
g250<-ggplot(data = df.m[df.m$impute.Hz.vec=='250',], aes(y = HR.Rhat.vec, x = GP_run)) +
  geom_point(data = df.m[df.m$impute.Hz.vec=='250',],
    aes(x = GP_run, color = GP.Hz.vec),
    position = position_jitter(width = .15),
    size = 1, alpha = 0.5) +
  geom_boxplot(data = df.m[df.m$impute.Hz.vec=='250',],
    aes(x = GP_run, y = HR.Rhat.vec, fill = GP.Hz.vec),
    width = .5, guides = FALSE, outlier.shape = NA, alpha = 0.5, inherit.aes = F) +
  #geom_point(data = df.m[df.m$variable!='GP1 Replacement' & df.m$impute.Hz.vec=='250',],
  # aes(x = variable, y = value, color = variable),
  # position = position_jitter(width = .15),
  # size = 1,
  # alpha = 0.5,
  # inherit.aes = F,
  # guides = F) +
  #expand_limits(x = 4.25) +
  coord_cartesian(ylim = c(0,625))+
  geom_hline(yintercept = 1.1, color='red', lty='dashed')+
  scale_color_brewer(palette = "Dark2") +
  scale_fill_brewer(palette = "Dark2", name = 'GP Model Hz') +
  theme_bw() +
  xlab('')+ylab('Model Run Time (mins)')+
  guides(color=FALSE)+
  ggtitle('Distributions of Heart Rate Parameter R-hat Values as a
    Function of Model and Modeling Conditions (250 Hz)')+
  theme(axis.text.x = element_text(angle = 30, hjust = 1))+
  facet_wrap(~impute.fac.vec)

png(paste0(final.graphics, '/Model_HR_Rhat.png'),
  res=600,
  units = 'in',
  width = 11,
  height = 14)

```

```
cowplot::plot_grid(g125, g250, nrow = 2)
dev.off()
```

Appendix C.11: Code used to generate figure 4.2

```
dat.runtime_1<-dat.GP1[c('impute.fac.vec',
                        'impute.Hz.vec',
                        'impute.tot',
                        'GP.Hz.vec',
                        'runtime'
                        )]

#Getting GP1 prepped...
dat.runtime_1$GP_run<-rep('GP1 Replacement')
dat.runtime_1$impute.fac.vec<-as.character(dat.runtime_1$impute.fac.vec)
dat.runtime_1$impute.fac.vec[dat.runtime_1$impute.fac.vec==1]<- '1x Imputation Window Length'
dat.runtime_1$impute.fac.vec[dat.runtime_1$impute.fac.vec==2]<- '2x Imputation Window Length'
dat.runtime_1$impute.fac.vec[dat.runtime_1$impute.fac.vec==3]<- '3x Imputation Window Length'

#Cleaning up values
dat.runtime_1$impute.fac.vec<-as.character(dat.runtime_1$impute.fac.vec)
dat.runtime_1$impute.Hz.vec<-as.character(dat.runtime_1$impute.Hz.vec)
dat.runtime_1$GP.Hz.vec<-factor(dat.runtime_1$GP.Hz.vec)

#=====
=====
#Bringing in GP2 Data...
dat.runtime_2<-dat.GP2[c('impute.fac.vec',
                        'impute.Hz.vec',
                        'impute.tot',
                        'GP.Hz.vec',
                        'runtime'
                        )]

#Getting GP2 prepped...
dat.runtime_2$GP_run<-rep('GP2 Replacement')
dat.runtime_2$impute.fac.vec<-as.character(dat.runtime_2$impute.fac.vec)
dat.runtime_2$impute.fac.vec[dat.runtime_2$impute.fac.vec==1]<- '1x Imputation Window Length'
dat.runtime_2$impute.fac.vec[dat.runtime_2$impute.fac.vec==2]<- '2x Imputation Window Length'
dat.runtime_2$impute.fac.vec[dat.runtime_2$impute.fac.vec==3]<- '3x Imputation Window Length'

#Cleaning up values
dat.runtime_2$impute.fac.vec<-as.character(dat.runtime_2$impute.fac.vec)
dat.runtime_2$impute.Hz.vec<-as.character(dat.runtime_2$impute.Hz.vec)
dat.runtime_2$GP.Hz.vec<-factor(dat.runtime_2$GP.Hz.vec)

#=====
=====
dat.runtime_3<-dat.GP3[c('impute.fac.vec',
                        'impute.Hz.vec',
                        'GP.Hz.vec',
                        'impute.tot',
                        'runtime'
                        )]

#Getting GP3 prepped...
dat.runtime_3$GP_run<-rep('GP3 Replacement')
dat.runtime_3$impute.fac.vec<-as.character(dat.runtime_3$impute.fac.vec)
dat.runtime_3$impute.fac.vec[dat.runtime_3$impute.fac.vec==1]<- '1x Imputation Window Length'
dat.runtime_3$impute.fac.vec[dat.runtime_3$impute.fac.vec==2]<- '2x Imputation Window Length'
dat.runtime_3$impute.fac.vec[dat.runtime_3$impute.fac.vec==3]<- '3x Imputation Window Length'

#Cleaning up values
dat.runtime_3$impute.fac.vec<-as.character(dat.runtime_3$impute.fac.vec)
dat.runtime_3$impute.Hz.vec<-as.character(dat.runtime_3$impute.Hz.vec)
dat.runtime_3$GP.Hz.vec<-factor(dat.runtime_3$GP.Hz.vec)

df.m<-rbind(dat.runtime_1,
            dat.runtime_2,
```

```

dat.runtime_3)

#-----
#Making standard boxplot for run time (similar to others in M5)
g125<-ggplot(data = df.m[df.m$impute.Hz.vec=='125',], aes(y = runtime, x = GP_run)) +
  geom_point(data = df.m[df.m$impute.Hz.vec=='125',],
    aes(x = GP_run, color = GP.Hz.vec),
    position = position_jitter(width = .15),
    size = 1, alpha = 0.5) +
  geom_boxplot(data = df.m[df.m$impute.Hz.vec=='125',],
    aes(x = GP_run, y = runtime, fill = GP.Hz.vec),
    width = .5, guides = FALSE, outlier.shape = NA, alpha = 0.5, inherit.aes = F) +
  #geom_point(data = df.m[df.m$variable!='GP1 Replacement' & df.m$impute.Hz.vec=='125',],
  # aes(x = variable, y = value, color = variable),
  # position = position_jitter(width = .15),
  # size = 1,
  # alpha = 0.5,
  # inherit.aes = F,
  # guides = F) +
  #expand_limits(x = 4.25) +
  scale_color_brewer(palette = "Dark2", name = 'GP Model Hz') +
  scale_fill_brewer(palette = "Dark2", name='GP Model Hz') +
  theme_bw() +
  xlab('')+ylab('Model Run Time (mins)')+
  coord_cartesian(ylim = c(0, 810))+
  guides(color=FALSE)+
  ggtitle('Distributions of Total Model Run Time as a Function of Model and Modeling Conditions
(125 Hz)')+
  theme(axis.text.x = element_text(angle = 30, hjust = 1))+
  facet_wrap(~impute.fac.vec)

#Looks Like a GP sample rate of 12 Hz with a factor of 2 window on either side is best...
#This may be unique the the GP 1 model specification...
#Will need to investigate more
g250<-ggplot(data = df.m[df.m$impute.Hz.vec=='250',], aes(y = runtime, x = GP_run)) +
  geom_point(data = df.m[df.m$impute.Hz.vec=='250',],
    aes(x = GP_run, color = GP.Hz.vec),
    position = position_jitter(width = .15),
    size = 1, alpha = 0.5) +
  geom_boxplot(data = df.m[df.m$impute.Hz.vec=='250',],
    aes(x = GP_run, y = runtime, fill = GP.Hz.vec),
    width = .5, guides = FALSE, outlier.shape = NA, alpha = 0.5, inherit.aes = F) +
  #geom_point(data = df.m[df.m$variable!='GP1 Replacement' & df.m$impute.Hz.vec=='250',],
  # aes(x = variable, y = value, color = variable),
  # position = position_jitter(width = .15),
  # size = 1,
  # alpha = 0.5,
  # inherit.aes = F,
  # guides = F) +
  #expand_limits(x = 4.25) +
  scale_color_brewer(palette = "Dark2", name = 'GP Model Hz') +
  scale_fill_brewer(palette = "Dark2", name='GP Model Hz') +
  theme_bw() +
  xlab('')+ylab('Model Run Time (mins)')+
  coord_cartesian(ylim = c(0, 810))+
  guides(color=FALSE)+
  ggtitle('Distributions of Total Model Run Time as a Function of Model and Modeling Conditions
(250 Hz)')+
  theme(axis.text.x = element_text(angle = 30, hjust = 1))+
  facet_wrap(~impute.fac.vec)

png(paste0(final.graphics, '/Model_runtime.png'),
  res=600,
  units = 'in',
  width = 11,
  height = 14)
cowplot::plot_grid(g125, g250, nrow = 2)
dev.off()

```

Appendix C.12: Code used to generate figure 4.3

```
df.m$impute.Hz.vec[df.m$impute.Hz.vec=='125']<- '125 Hz'
df.m$impute.Hz.vec[df.m$impute.Hz.vec=='250']<- '250 Hz'

png(paste0(final.graphics, '/Run_time_facets.png'),
    res=600,
    units = 'in',
    width = 11,
    height = 8)
ggplot()+
  geom_point(data=df.m, aes(x=impute.tot, y=runtime, color = GP.Hz.vec))+
  stat_smooth(data = df.m, aes(x=impute.tot, y=runtime, color = GP.Hz.vec), method = 'loess',
se=F)+
  coord_cartesian(ylim=c(0, 810))+
  facet_wrap(~impute.Hz.vec*impute.fac.vec*GP_run, nrow=3, ncol=6)+
  ggtitle('Model Run Time as a Function of Total Time Imputed')+
  scale_color_brewer(palette = "Dark2", name = 'GP Model Hz') +
  theme_bw()+
  xlab('Total Time Targeted for Imputation (seconds)')+
  ylab('Model Run Time (mins)')+
  theme(strip.text.x = element_text(size=7.5))+
  theme(strip.background.x = element_rect(linetype = 'blank'))
dev.off()
```

Appendix C.13: Code used to generate figure 4.4

```
GP1.data.folder<- 'C:/Users/Mbars/Dropbox/Dissertation/Chapter 4/Sim_data/Output/Final_Data'

dat.GP1<-read.csv(paste0(GP1.data.folder, '/Final_GP1_40sims.csv'), stringsAsFactors = F)
dat.GP1$bias_rmssd15<-dat.GP1$IBI.GP.rmssd.15-dat.GP1$IBI.rmssd.15
dat.rmssd15_1<-dat.GP1[c('impute.fac.vec',
                        'impute.Hz.vec',
                        'GP.Hz.vec',
                        'IBI.rmssd.15',
                        'IBI.HD.rmssd.15',
                        'IBI.MR.rmssd.15',
                        'IBI.GP.rmssd.15'
                        )]

dat.rmssd15_1$HD.bias<-dat.rmssd15_1$IBI.HD.rmssd.15-dat.rmssd15_1$IBI.rmssd.15
dat.rmssd15_1$MR.bias<-dat.rmssd15_1$IBI.MR.rmssd.15-dat.rmssd15_1$IBI.rmssd.15
dat.rmssd15_1$GP.bias<-dat.rmssd15_1$IBI.GP.rmssd.15-dat.rmssd15_1$IBI.rmssd.15

#Getting GP1 prepped...
df.m1<-reshape2::melt(dat.rmssd15_1[,c(1:3, 8:10)], id.var=c('impute.fac.vec', 'GP.Hz.vec',
' impute.Hz.vec'))
df.m1$variable<-as.character(df.m1$variable)
df.m1$variable[df.m1$variable=='HD.bias']<- 'Hot Deck Replacement'
df.m1$variable[df.m1$variable=='MR.bias']<- 'Mean Replacement'
df.m1$variable[df.m1$variable=='GP.bias']<- 'GP1 Replacement'

df.m1$impute.fac.vec<-as.character(df.m1$impute.fac.vec)
df.m1$impute.fac.vec[df.m1$impute.fac.vec==1]<- '1x Imputation Window Length'
df.m1$impute.fac.vec[df.m1$impute.fac.vec==2]<- '2x Imputation Window Length'
df.m1$impute.fac.vec[df.m1$impute.fac.vec==3]<- '3x Imputation Window Length'

#Cleaning up values
df.m1$impute.fac.vec<-as.character(df.m1$impute.fac.vec)
df.m1$impute.Hz.vec<-as.character(df.m1$impute.Hz.vec)
```

```

df.m1$GP.Hz.vec<-factor(df.m1$GP.Hz.vec)

#####
#####
#Bringing in GP2 Data...
GP2.data.folder<-'C:/Users/Mbars/Dropbox/Dissertation/Chapter 4/Sim_data_2_b/Output/Final_Data'

dat.GP2<-read.csv(paste0(GP2.data.folder, '/Final_GP2_40sims.csv'), stringsAsFactors = F)
dat.GP2$bias_rmssd15<-dat.GP2$IBI.GP.rmssd.15-dat.GP2$IBI.rmssd.15
dat.rmssd15_2<-dat.GP2[c('impute.fac.vec',
                        'impute.Hz.vec',
                        'GP.Hz.vec',
                        'IBI.rmssd.15',
                        'IBI.HD.rmssd.15',
                        'IBI.MR.rmssd.15',
                        'IBI.GP.rmssd.15'
)]

dat.rmssd15_2$HD.bias<-dat.rmssd15_2$IBI.HD.rmssd.15-dat.rmssd15_2$IBI.rmssd.15
dat.rmssd15_2$MR.bias<-dat.rmssd15_2$IBI.MR.rmssd.15-dat.rmssd15_2$IBI.rmssd.15
dat.rmssd15_2$GP.bias<-dat.rmssd15_2$IBI.GP.rmssd.15-dat.rmssd15_2$IBI.rmssd.15

#Getting GP1 prepped...
df.m2<-reshape2::melt(dat.rmssd15_2[,c(1:3, 8:10)], id.var=c('impute.fac.vec', 'GP.Hz.vec',
' impute.Hz.vec'))
df.m2$variable<-as.character(df.m2$variable)
df.m2$variable[df.m2$variable=='HD.bias']<-'Hot Deck Replacement'
df.m2$variable[df.m2$variable=='MR.bias']<-'Mean Replacement'
df.m2$variable[df.m2$variable=='GP.bias']<-'GP2 Replacement'

df.m2$impute.fac.vec<-as.character(df.m2$impute.fac.vec)
df.m2$impute.fac.vec[df.m2$impute.fac.vec==1]<-'1x Imputation Window Length'
df.m2$impute.fac.vec[df.m2$impute.fac.vec==2]<-'2x Imputation Window Length'
df.m2$impute.fac.vec[df.m2$impute.fac.vec==3]<-'3x Imputation Window Length'

#####
#####
GP3.data.folder<-'C:/Users/Mbars/Dropbox/Dissertation/Chapter 4/Sim_data_3_b/Output/Final_Data'

dat.GP3<-read.csv(paste0(GP3.data.folder, '/Final_GP3_40sims.csv'), stringsAsFactors = F)
dat.GP3$bias_rmssd15<-dat.GP3$IBI.GP.rmssd.15-dat.GP3$IBI.rmssd.15
dat.rmssd15_3<-dat.GP3[c('impute.fac.vec',
                        'impute.Hz.vec',
                        'GP.Hz.vec',
                        'IBI.rmssd.15',
                        'IBI.HD.rmssd.15',
                        'IBI.MR.rmssd.15',
                        'IBI.GP.rmssd.15'
)]

dat.rmssd15_3$HD.bias<-dat.rmssd15_3$IBI.HD.rmssd.15-dat.rmssd15_3$IBI.rmssd.15
dat.rmssd15_3$MR.bias<-dat.rmssd15_3$IBI.MR.rmssd.15-dat.rmssd15_3$IBI.rmssd.15
dat.rmssd15_3$GP.bias<-dat.rmssd15_3$IBI.GP.rmssd.15-dat.rmssd15_3$IBI.rmssd.15

#Getting GP1 prepped...
df.m3<-reshape2::melt(dat.rmssd15_3[,c(1:3, 8:10)], id.var=c('impute.fac.vec', 'GP.Hz.vec',
' impute.Hz.vec'))
df.m3$variable<-as.character(df.m3$variable)
df.m3$variable[df.m3$variable=='HD.bias']<-'Hot Deck Replacement'
df.m3$variable[df.m3$variable=='MR.bias']<-'Mean Replacement'
df.m3$variable[df.m3$variable=='GP.bias']<-'GP3 Replacement'

df.m3$impute.fac.vec<-as.character(df.m3$impute.fac.vec)
df.m3$impute.fac.vec[df.m3$impute.fac.vec==1]<-'1x Imputation Window Length'
df.m3$impute.fac.vec[df.m3$impute.fac.vec==2]<-'2x Imputation Window Length'
df.m3$impute.fac.vec[df.m3$impute.fac.vec==3]<-'3x Imputation Window Length'

df.m<-rbind(df.m1, df.m2, df.m3)

```

```

g125<-ggplot(data = df.m[df.m$impute.Hz.vec=='125',], aes(y = value, x = variable)) +
#geom_point(data = df.m[df.m$variable == 'GP1 Replacement' & df.m$impute.Hz.vec=='125',],
# aes(x = variable, color = GP.Hz.vec),
# position = position_jitter(width = .15),
# size = 1, alpha = 0.5) +
geom_boxplot(data = df.m[df.m$variable == 'GP1 Replacement' & df.m$impute.Hz.vec=='125',],
aes(x = variable, y = value, fill = GP.Hz.vec),
width = .5, guides = FALSE, outlier.shape = NA, alpha = 0.5, inherit.aes = F) +
geom_boxplot(data = df.m[df.m$variable == 'GP2 Replacement' & df.m$impute.Hz.vec=='125',],
aes(x = variable, y = value, fill = GP.Hz.vec),
width = .5, guides = FALSE, outlier.shape = NA, alpha = 0.5, inherit.aes = F) +
geom_boxplot(data = df.m[df.m$variable == 'GP3 Replacement' & df.m$impute.Hz.vec=='125',],
aes(x = variable, y = value, fill = GP.Hz.vec),
width = .5, guides = FALSE, outlier.shape = NA, alpha = 0.5, inherit.aes = F) +
#geom_point(data = df.m[df.m$variable!='GP1 Replacement' & df.m$impute.Hz.vec=='125',],
# aes(x = variable, y = value, color = variable),
# position = position_jitter(width = .15),
# size = 1,
# alpha = 0.5,
# inherit.aes = F,
# guides = F) +
geom_boxplot(data = df.m[df.m$variable != 'GP1 Replacement' &
df.m$variable != 'GP2 Replacement' &
df.m$variable != 'GP3 Replacement' &
df.m$impute.Hz.vec=='125',],
aes(x = variable, y = value, color = variable),
width = .5, guides = FALSE, outlier.shape = NA, alpha = 0.5, inherit.aes = F) +
#expand_limits(x = 4.25) +
scale_color_brewer(palette = "Dark2") +
scale_fill_brewer(palette = "Dark2", name='GP Model Hz') +
theme_bw() +
xlab('')+ylab('RMSSD Deviation (125 Hz)')+
guides(color=FALSE)+
ggtitle('Gaussian Process RMSSD Bias in Different Epoch Intervals: 125 Hz Signal Sampling
Rate')+
coord_cartesian(ylim =c(-.05, .15))+
geom_hline(yintercept = 0, color='red', lty='dashed')+
theme(axis.text.x = element_text(angle = 30, hjust = 1))+
facet_wrap(~impute.fac.vec)

#Looks Like a GP sample rate of 12 Hz with a factor of 2 window on either side is best...
#This may be unique the the GP 1 model specification..
#Will need to investigate more
g250<-ggplot(data = df.m[df.m$impute.Hz.vec=='250',], aes(y = value, x = variable)) +
#geom_point(data = df.m[df.m$variable == 'GP1 Replacement' & df.m$impute.Hz.vec=='250',],
# aes(x = variable, color = GP.Hz.vec),
# position = position_jitter(width = .15),
# size = 1, alpha = 0.5) +
geom_boxplot(data = df.m[df.m$variable == 'GP1 Replacement' & df.m$impute.Hz.vec=='250',],
aes(x = variable, y = value, fill = GP.Hz.vec),
width = .5, guides = FALSE, outlier.shape = NA, alpha = 0.5, inherit.aes = F) +
geom_boxplot(data = df.m[df.m$variable == 'GP2 Replacement' & df.m$impute.Hz.vec=='250',],
aes(x = variable, y = value, fill = GP.Hz.vec),
width = .5, guides = FALSE, outlier.shape = NA, alpha = 0.5, inherit.aes = F) +
geom_boxplot(data = df.m[df.m$variable == 'GP3 Replacement' & df.m$impute.Hz.vec=='250',],
aes(x = variable, y = value, fill = GP.Hz.vec),
width = .5, guides = FALSE, outlier.shape = NA, alpha = 0.5, inherit.aes = F) +
#geom_point(data = df.m[df.m$variable!='GP1 Replacement' & df.m$impute.Hz.vec=='125',],
# aes(x = variable, y = value, color = variable),
# position = position_jitter(width = .15),
# size = 1,
# alpha = 0.5,
# inherit.aes = F,
# guides = F) +
geom_boxplot(data = df.m[df.m$variable != 'GP1 Replacement' &
df.m$variable != 'GP2 Replacement' &
df.m$variable != 'GP3 Replacement' &
df.m$impute.Hz.vec=='250',],

```

```

aes(x = variable, y = value, color = variable),
width = .5, guides = FALSE, outlier.shape = NA, alpha = 0.5, inherit.aes = F) +
#expand_limits(x = 4.25) +
scale_color_brewer(palette = "Dark2") +
scale_fill_brewer(palette = "Dark2", name='GP Model Hz') +
theme_bw() +
xlab('')+ylab('RMSSD Deviation (250 Hz)')+
guides(color=FALSE)+
ggtitle('Gaussian Process RMSSD Bias in Different Epoch Intervals: 250 Hz Signal Sampling
Rate')+
coord_cartesian(ylim =c(-.05, .15))+
geom_hline(yintercept = 0, color='red', lty='dashed')+
theme(axis.text.x = element_text(angle = 30, hjust = 1))+
facet_wrap(~impute.fac.vec)

png(paste0(final.graphics, '/GP_Bias_overall_graphic.png'),
res=600,
units = 'in',
width = 11,
height = 14)
cowplot::plot_grid(g125, g250, nrow = 2)
dev.off()

```

Appendix C.14: Code used to generate figure 4.5

```

dat.rmssd15_1b<-dat.GP1[c('impute.fac.vec',
                        'impute.Hz.vec',
                        'HR.Rhat.vec',
                        'GP.Hz.vec',
                        'IBI.rmssd.15',
                        'IBI.HD.rmssd.15',
                        'IBI.MR.rmssd.15',
                        'IBI.GP.rmssd.15'
                        )]

dat.rmssd15_1b$HD.bias<-dat.rmssd15_1b$IBI.HD.rmssd.15-dat.rmssd15_1b$IBI.rmssd.15
dat.rmssd15_1b$MR.bias<-dat.rmssd15_1b$IBI.MR.rmssd.15-dat.rmssd15_1b$IBI.rmssd.15
dat.rmssd15_1b$GP.bias<-dat.rmssd15_1b$IBI.GP.rmssd.15-dat.rmssd15_1b$IBI.rmssd.15

df.m1<-reshape2::melt(dat.rmssd15_1b[,c(1:4, 9:11)],
                      id.var=c('impute.fac.vec', 'GP.Hz.vec', 'impute.Hz.vec', 'HR.Rhat.vec'))
df.m1$variable<-as.character(df.m1$variable)
df.m1$variable[df.m1$variable=='HD.bias']<- 'Hot Deck Replacement'
df.m1$variable[df.m1$variable=='MR.bias']<- 'Mean Replacement'
df.m1$variable[df.m1$variable=='GP.bias']<- 'GP1 Replacement'

df.m1$impute.fac.vec<-as.character(df.m1$impute.fac.vec)
df.m1$impute.fac.vec[df.m1$impute.fac.vec==1]<- '1x Imputation Window Length'
df.m1$impute.fac.vec[df.m1$impute.fac.vec==2]<- '2x Imputation Window Length'
df.m1$impute.fac.vec[df.m1$impute.fac.vec==3]<- '3x Imputation Window Length'

dat.rmssd15_2b<-dat.GP2[c('impute.fac.vec',
                        'impute.Hz.vec',
                        'HR.Rhat.vec',
                        'GP.Hz.vec',
                        'IBI.rmssd.15',
                        'IBI.HD.rmssd.15',
                        'IBI.MR.rmssd.15',
                        'IBI.GP.rmssd.15'
                        )]

dat.rmssd15_2b$HD.bias<-dat.rmssd15_2b$IBI.HD.rmssd.15-dat.rmssd15_2b$IBI.rmssd.15
dat.rmssd15_2b$MR.bias<-dat.rmssd15_2b$IBI.MR.rmssd.15-dat.rmssd15_2b$IBI.rmssd.15
dat.rmssd15_2b$GP.bias<-dat.rmssd15_2b$IBI.GP.rmssd.15-dat.rmssd15_2b$IBI.rmssd.15

```



```

df.m2<-reshape2::melt(dat.rmssd15_2b[,c(1:4, 9:11)],
                      id.var=c('impute.fac.vec', 'GP.Hz.vec', 'impute.Hz.vec', 'HR.Rhat.vec'))
df.m2$variable<-as.character(df.m2$variable)
df.m2$variable[df.m2$variable=='HD.bias']<-'Hot Deck Replacement'
df.m2$variable[df.m2$variable=='MR.bias']<-'Mean Replacement'
df.m2$variable[df.m2$variable=='GP.bias']<-'GP2 Replacement'

df.m2$impute.fac.vec<-as.character(df.m2$impute.fac.vec)
df.m2$impute.fac.vec[df.m2$impute.fac.vec==1]<-'1x Imputation Window Length'
df.m2$impute.fac.vec[df.m2$impute.fac.vec==2]<-'2x Imputation Window Length'
df.m2$impute.fac.vec[df.m2$impute.fac.vec==3]<-'3x Imputation Window Length'

dat.rmssd15_3b<-dat.GP3[c('impute.fac.vec',
                        'impute.Hz.vec',
                        'HR.Rhat.vec',
                        'GP.Hz.vec',
                        'IBI.rmssd.15',
                        'IBI.HD.rmssd.15',
                        'IBI.MR.rmssd.15',
                        'IBI.GP.rmssd.15'
                        )]

dat.rmssd15_3b$HD.bias<-dat.rmssd15_3b$IBI.HD.rmssd.15-dat.rmssd15_3b$IBI.rmssd.15
dat.rmssd15_3b$MR.bias<-dat.rmssd15_3b$IBI.MR.rmssd.15-dat.rmssd15_3b$IBI.rmssd.15
dat.rmssd15_3b$GP.bias<-dat.rmssd15_3b$IBI.GP.rmssd.15-dat.rmssd15_3b$IBI.rmssd.15

df.m3<-reshape2::melt(dat.rmssd15_3b[,c(1:4, 9:11)],
                      id.var=c('impute.fac.vec', 'GP.Hz.vec', 'impute.Hz.vec', 'HR.Rhat.vec'))
df.m3$variable<-as.character(df.m3$variable)
df.m3$variable[df.m3$variable=='HD.bias']<-'Hot Deck Replacement'
df.m3$variable[df.m3$variable=='MR.bias']<-'Mean Replacement'
df.m3$variable[df.m3$variable=='GP.bias']<-'GP3 Replacement'

df.m3$impute.fac.vec<-as.character(df.m3$impute.fac.vec)
df.m3$impute.fac.vec[df.m3$impute.fac.vec==1]<-'1x Imputation Window Length'
df.m3$impute.fac.vec[df.m3$impute.fac.vec==2]<-'2x Imputation Window Length'
df.m3$impute.fac.vec[df.m3$impute.fac.vec==3]<-'3x Imputation Window Length'

df.m<-rbind(df.m1, df.m2, df.m3)

df.m$GP.Hz.vec<-as.factor(df.m$GP.Hz.vec)
#Getting Bias only among models that have converged
g125<-ggplot(data = df.m[df.m$impute.Hz.vec=='125' & df.m$HR.Rhat.vec<=1.1,], aes(y = value, x =
variable)) +
  #geom_point(data = df.m[df.m$variable == 'GP1 Replacement' & df.m$impute.Hz.vec=='125'],,
  # aes(x = variable, color = GP.Hz.vec),
  # position = position_jitter(width = .15),
  # size = 1, alpha = 0.5) +
  geom_boxplot(data = df.m[df.m$variable == 'GP1 Replacement' &
df.m$impute.Hz.vec=='125' & df.m$HR.Rhat.vec<=1.1,],
aes(x = variable, y = value, fill = GP.Hz.vec),
width = .5, guides = FALSE, outlier.shape = NA, alpha = 0.5, inherit.aes = F) +
  geom_boxplot(data = df.m[df.m$variable == 'GP2 Replacement' &
df.m$impute.Hz.vec=='125' & df.m$HR.Rhat.vec<=1.1,],
aes(x = variable, y = value, fill = GP.Hz.vec),
width = .5, guides = FALSE, outlier.shape = NA, alpha = 0.5, inherit.aes = F) +
  geom_boxplot(data = df.m[df.m$variable == 'GP3 Replacement' &
df.m$impute.Hz.vec=='125' & df.m$HR.Rhat.vec<=1.1,],
aes(x = variable, y = value, fill = GP.Hz.vec),
width = .5, guides = FALSE, outlier.shape = NA, alpha = 0.5, inherit.aes = F) +
  #geom_point(data = df.m[df.m$variable!='GP1 Replacement' & df.m$impute.Hz.vec=='125'],,
  # aes(x = variable, y = value, color = variable),
  # position = position_jitter(width = .15),
  # size = 1,
  # alpha = 0.5,
  # inherit.aes = F,
  # guides = F) +
  geom_boxplot(data = df.m[df.m$variable != 'GP1 Replacement' &
df.m$variable != 'GP2 Replacement' &

```



```

        df.m$variable != 'GP3 Replacement' &
        df.m$impute.Hz.vec=='125',],
        aes(x = variable, y = value, color = variable),
        width = .5, guides = FALSE, outlier.shape = NA, alpha = 0.5, inherit.aes = F) +
#expand_limits(x = 4.25) +
scale_color_brewer(palette = "Dark2") +
scale_fill_brewer(palette = "Dark2", name='GP Model Hz') +
theme_bw() +
xlab('')+ylab('RMSSD Deviation (125 Hz)')+
guides(color=FALSE)+
ggtitle('Gaussian Process RMSSD Bias in Different Epoch Intervals:
125 Hz Signal Sampling Rate (HR Parameter Convergence)')+
coord_cartesian(ylim =c(-.05, .15))+
geom_hline(yintercept = 0, color='red', lty='dashed')+
theme(axis.text.x = element_text(angle = 30, hjust = 1))+
facet_wrap(~impute.fac.vec)

#Looks Like a GP sample rate of 12 Hz with a factor of 2 window on either side is best...
#This may be unique the the GP 1 model specification...
#Will need to investigate more
g250<-ggplot(data = df.m[df.m$impute.Hz.vec=='250' &
df.m$HR.Rhat.vec<=1.1,], aes(y = value, x = variable)) +
#geom_point(data = df.m[df.m$variable == 'GP1 Replacement' & df.m$impute.Hz.vec=='250',],
# aes(x = variable, color = GP.Hz.vec),
# position = position_jitter(width = .15),
# size = 1, alpha = 0.5) +
geom_boxplot(data = df.m[df.m$variable == 'GP1 Replacement' &
df.m$impute.Hz.vec=='250' & df.m$HR.Rhat.vec<=1.1,],
aes(x = variable, y = value, fill = GP.Hz.vec),
width = .5, guides = FALSE, outlier.shape = NA, alpha = 0.5, inherit.aes = F) +
geom_boxplot(data = df.m[df.m$variable == 'GP2 Replacement' &
df.m$impute.Hz.vec=='250' & df.m$HR.Rhat.vec<=1.1,],
aes(x = variable, y = value, fill = GP.Hz.vec),
width = .5, guides = FALSE, outlier.shape = NA, alpha = 0.5, inherit.aes = F) +
geom_boxplot(data = df.m[df.m$variable == 'GP3 Replacement' &
df.m$impute.Hz.vec=='250' & df.m$HR.Rhat.vec<=1.1,],
aes(x = variable, y = value, fill = GP.Hz.vec),
width = .5, guides = FALSE, outlier.shape = NA, alpha = 0.5, inherit.aes = F) +
#geom_point(data = df.m[df.m$variable!='GP1 Replacement' & df.m$impute.Hz.vec=='250',],
# aes(x = variable, y = value, color = variable),
# position = position_jitter(width = .15),
# size = 1,
# alpha = 0.5,
# inherit.aes = F,
# guides = F) +
geom_boxplot(data = df.m[df.m$variable != 'GP1 Replacement' &
df.m$variable != 'GP2 Replacement' &
df.m$variable != 'GP3 Replacement' &
df.m$impute.Hz.vec=='250',],
aes(x = variable, y = value, color = variable),
width = .5, guides = FALSE, outlier.shape = NA, alpha = 0.5, inherit.aes = F) +
#expand_limits(x = 4.25) +
scale_color_brewer(palette = "Dark2") +
scale_fill_brewer(palette = "Dark2", name='GP Model Hz') +
theme_bw() +
xlab('')+ylab('RMSSD Deviation (250 Hz)')+
guides(color=FALSE)+
ggtitle('Gaussian Process RMSSD Bias in Different Epoch Intervals:
250 Hz Signal Sampling Rate (HR Parameter Convergence)')+
coord_cartesian(ylim =c(-.05, .15))+
geom_hline(yintercept = 0, color='red', lty='dashed')+
theme(axis.text.x = element_text(angle = 30, hjust = 1))+
facet_wrap(~impute.fac.vec)

png(paste0(final.graphics, '/GP_Bias_overall_graphic_Rhat_conv.png'),
res=600,
units = 'in',
width = 11,

```

```

height = 14)
cowplot::plot_grid(g125, g250, nrow = 2)
dev.off()

```

Appendix C.15: Code used to generate figure 4.6

```

dat.p_edits<-dat.main[,31:36]
df.m<-reshape2::melt(dat.p_edits)
df.m$variable<-as.character(df.m$variable)

#Recoding variable names to align with the task names
df.m$variable[df.m$variable=='pEdits_Video1']<-'Video 1'
df.m$variable[df.m$variable=='pEdits_Video2']<-'Video 2'
df.m$variable[df.m$variable=='pEdits_Video3']<-'Video 3'
df.m$variable[df.m$variable=='pEdits_Clown']<-'Clown'
df.m$variable[df.m$variable=='pEdits_Kids']<-'Kids'
df.m$variable[df.m$variable=='pEdits_Intro']<-'Introduction'

g1<-ggplot(data = df.m, aes(y = value, x = variable, fill=variable)) +
  geom_flat_violin(position = position_nudge(x = .2, y = 0), alpha = .8) +
  geom_point(aes(y = value, color = variable),
             position = position_jitter(width = .15),
             size = .5, alpha = 0.8) +
  geom_boxplot(width = .1, guides = FALSE, outlier.shape = NA, alpha = 0.5) +
  expand_limits(x = 5.25) +
  scale_color_brewer(palette = "Dark2") +
  scale_fill_brewer(palette = "Dark2") +
  theme_bw() +
  raincloud_theme+
  xlab('')+ylab('Proportion of Edits')+
  guides(fill=FALSE, color=FALSE)+
  ggtitle('Proportion of Edits as a Function of Task')

png(paste0(EDA.folder, '/Proportion of Edits by Task.png'), units = 'in',
    height = 8, width = 8, res = 300)
g1
dev.off()

```

Appendix C.16: Code used to generate figure 4.7

```

dat.intra<-read.csv(paste0(data.folder, '/Editing Assignments - CE_VizEdit_rel.csv'))

#I went back and forth as to whether I would out myself as Editor 1 in the end.
#I figured transparency is the way to go. So here it is in the code.
#I was Editor 1 - and I did not do that well going between programs :)
#This is what the kids call an Easter Egg I believe...

dat.intra$Editor_anon[dat.intra$Editor=='Matthew Barstead']<-'Editor 1'
dat.intra$Editor_anon[dat.intra$Editor=='Sophia Lamp']<-'Editor 2'

psych::describeBy(dat.intra, group='Editor')

df.m<-reshape2::melt(dat.intra[,6:7])
df.m$variable<-as.character(df.m$variable)

g1<-ggplot(data = dat.intra, aes(y = Diff, x = Editor_anon, fill=Editor_anon)) +
  geom_flat_violin(position = position_nudge(x = .2, y = 0), alpha = .8) +
  geom_point(aes(y = Diff, color = Editor_anon),
             position = position_jitter(width = .15, seed = 123),
             size = 1, alpha = 0.8) +

```

```

geom_boxplot(width = .1, guides = FALSE, outlier.shape = NA, alpha = 0.5) +
expand_limits(x = 3.25) +
scale_color_brewer(palette = "Dark2") +
scale_fill_brewer(palette = "Dark2") +
geom_hline(yintercept = .05, color = 'red', lty='dashed', alpha = .5)+
geom_hline(yintercept = -.05, color = 'red', lty='dashed', alpha = .5)+
geom_hline(yintercept = .10, color = 'blue', lty='dashed', alpha = .5)+
geom_hline(yintercept = -.10, color = 'blue', lty='dashed', alpha = .5)+
geom_hline(yintercept = 0, color = 'black', lty='dashed', alpha = .5)+
theme_bw() +
raincloud_theme+
xlab('')+ylab('RSA Difference Scores')+
guides(fill=FALSE, color=FALSE)+
ggtitle('Distributions of Differences in RSA scores Produced by VizEdit vs. Cardio Edit')

png(paste0(out.folder, '/Intra-Rater_CE_vs_VizEdit.png'),
    units = 'in',
    width = 8,
    height = 5,
    res=600)

g1
dev.off()

```

Appendix C.17: Code used to generate figure 4.8

```

dat.inter<-read.csv(paste0(data.folder, '/Editing Assignments - MB_Diss_rel.csv'),
stringsAsFactors = F)
RSA.wide2<-dat.inter[c('File',
                      colnames(dat.inter)[grep('RSA15', colnames(dat.inter))])]
RSA.wide2$Editor<-rep('Editor 3')
RSA.wide$Editor[dat.main$Editor=='Matthew Barstead']<-'Editor 1'
RSA.wide$Editor[dat.main$Editor=='Sophia Lamp']<-'Editor 2'

RSA.wide.E1<-RSA.wide[RSA.wide$Editor=='Editor 1',]
E3.Files<-RSA.wide2$File
RSA.E1_E3<-data.frame(stringsAsFactors = F)
for(i in 1:length(E3.Files)){
  if(sum(RSA.wide.E1$File==E3.Files[i])==1){
    RSA.E1_E3<-rbind(RSA.E1_E3, RSA.wide[RSA.wide$File==E3.Files[i],])
    RSA.E1_E3<-rbind(RSA.E1_E3, RSA.wide2[RSA.wide2$File==E3.Files[i],])
  }
}

RSA.long.E1_E3<-reshape(RSA.E1_E3,
                        idvar = c('File', 'Editor'),
                        varying = colnames(RSA.E1_E3)[2:7],
                        timevar = 'Task',
                        times = c('Video1',
                                  'Clown',
                                  'Video2',
                                  'Kids',
                                  'Intro',
                                  'Video3'
                                ),
                        v.names = 'RSA',
                        direction = 'long')

E1vE3.DF<-cbind(RSA.long.E1_E3$RSA[RSA.long.E1_E3$Editor=='Editor 1'],
               RSA.long.E1_E3$RSA[RSA.long.E1_E3$Editor=='Editor 3'])

#Difference Scores - Editor 1 vs. Editor 3
E1vE3_Diff<-E1vE3.DF[,1]-E1vE3.DF[,2]

#Getting the same difference Scores for Editor 2 vs. Editor 3
RSA.wide.E2<-RSA.wide[RSA.wide$Editor=='Editor 2',]
E3.Files<-RSA.wide2$File

```

```

RSA.E2_E3<-data.frame(stringsAsFactors = F)
for(i in 1:length(E3.Files)){
  if(sum(RSA.wide.E2$File==E3.Files[i])==1){
    RSA.E2_E3<-rbind(RSA.E2_E3, RSA.wide[RSA.wide$File==E3.Files[i],])
    RSA.E2_E3<-rbind(RSA.E2_E3, RSA.wide2[RSA.wide2$File==E3.Files[i],])
  }
}

RSA.long.E2_E3<-reshape(RSA.E2_E3,
                        idvar = c('File', 'Editor'),
                        varying = colnames(RSA.E2_E3)[2:7],
                        timevar = 'Task',
                        times = c('Video1',
                                  'Clown',
                                  'Video2',
                                  'Kids',
                                  'Intro',
                                  'Video3'
                                ),
                        v.names = 'RSA',
                        direction = 'long')

E2vE3.DF<-cbind(RSA.long.E2_E3$RSA[RSA.long.E2_E3$Editor=='Editor 2'],
               RSA.long.E2_E3$RSA[RSA.long.E2_E3$Editor=='Editor 3'])

#Difference Scores - Editor 1 vs. Editor 3
E2vE3_Diff<-E2vE3.DF[,1]-E2vE3.DF[,2]

#Plotting Inter-Editor Comparisons
E1vE3.DF<-as.data.frame(E1vE3.DF)
colnames(E1vE3.DF)<-c('Primary_RSA', 'Secondary_RSA')
E1vE3.DF$Diff<-E1vE3_Diff
E1vE3.DF$Comparison<-rep('Editor 1 vs. Editor 3')

E2vE3.DF<-as.data.frame(E2vE3.DF)
colnames(E2vE3.DF)<-c('Primary_RSA', 'Secondary_RSA')
E2vE3.DF$Diff<-E2vE3_Diff
E2vE3.DF$Comparison<-rep('Editor 2 vs. Editor 3')

Editor.comp<-rbind(E1vE3.DF, E2vE3.DF)
Edit.comp.graph<-Editor.comp[Editor.comp$Diff>-.35,]
Edit.comp.graph<-Edit.comp.graph[!is.na(Edit.comp.graph$Comparison),]

g1<-ggplot(data = Edit.comp.graph, aes(y = Diff, x = Comparison, fill=Comparison)) +
  geom_flat_violin(position = position_nudge(x = .2, y = 0), alpha = .8) +
  geom_point(aes(y = Diff, color = Comparison),
             position = position_jitter(width = .15),
             size = 1, alpha = 0.8) +
  geom_boxplot(width = .1, guides = FALSE, outlier.shape = NA, alpha = 0.5) +
  expand_limits(x = 3.25) +
  scale_color_brewer(palette = "Dark2") +
  scale_fill_brewer(palette = "Dark2") +
  geom_hline(yintercept = .05, color = 'red', lty='dashed', alpha = .5)+
  geom_hline(yintercept = -.05, color = 'red', lty='dashed', alpha = .5)+
  geom_hline(yintercept = .10, color = 'blue', lty='dashed', alpha = .5)+
  geom_hline(yintercept = -.10, color = 'blue', lty='dashed', alpha = .5)+
  geom_hline(yintercept = 0, color = 'black', lty='dashed', alpha = .5)+
  theme_bw() +
  raincloud_theme+
  xlab('')+ylab('RSA Difference Scores')+
  guides(fill=FALSE, color=FALSE)+
  ggtitle('Distributions of RSA Difference Scores Between Editors Using IBI VizEdit')

png(paste0(out.folder, '/RSA_Inter_Editor_Comparison.png'), units = 'in',
    height = 5, width = 8, res = 300)

g1
dev.off()

```

Appendix C.18: Code used to generate figure 4.9 and 4.10 (Missing Data)

```
#RSA Missingness
png(paste0(out.folder, '/RSA_missing_summary.png'),
    units = 'in',
    width = 8,
    height = 5,
    res = 600)
VIM::aggr(RSA.wide[,2:7],
          labels = unlist(strsplit(names(RSA.wide[,2:7]), split = '_'))[c(2,4,6,8,10,12)],
          numbers=T,
          cex.axis =.95, oma = c(9,5,5,3))
title('Missing Data Summary for RSA Values Derived from 15-second Epochs')
dev.off()

#IBI VizEdit measures missingness.
png(paste0(out.folder, '/VizEdit_missing_summary.png'),
    units = 'in',
    width = 8,
    height = 5,
    res = 600)
VIM::aggr(HP.wide[,2:7],
          labels = unlist(strsplit(names(HP.wide[,2:7]), split = '_'))[c(2,4,6,8,10,12)],
          numbers=T,
          cex.axis =.95, oma = c(9,5,5,3))
title('Missing Data Summary for IBI VizEdit Task Values (RMSSD and HP)')
dev.off()
```

Appendix C.19: Code used to generate figure 5.1 - 5.3, 5.6, & 5.7 (these figures were output as part of the re-sampling script - see Appendix D)

```
g1<-ggplot()+
  geom_line(data=PPG.DF.new, aes(x=Time, y = PPG), color = 'red', lty='dashed')+
  geom_line(data=PPG.impute.pred, aes(x=Time, y=PPG), color = 'black')+
  ggtitle(paste(dat.temp$ID, dat.temp$Population, dat.temp$Segment,
               'File Hz =', Hz.temp, 'GP Dataset Hz =', Hz.impute.temp,
               'Model Sampling Rate =', Hz.GP.temp, 'Chains =', chain.temp,
               'Warmup =', warmup.temp, 'Iter =', n.iter.temp))

ggsave(filename = paste0(graphics.folder, paste('/', dat.temp$ID,
                                                dat.temp$Population,
                                                dat.temp$Segment,
                                                Hz.temp,
                                                Hz.impute.temp,
                                                Hz.GP.temp,
                                                impute.fac.temp,
                                                chain.temp,
                                                warmup.temp,
                                                n.iter.temp,
                                                round(impute.LB.temp, digits = 2),
                                                sep = '_'), '.png'),

        plot = g1,
        width = 11,
        height = 8,
        units = 'in',
        dpi = 300,
        device = 'png')
```

Appendix C.20: Code used to generate figure 5.4

```
dat.HR.conv2<-dat.GP2[c('impute.fac.vec',
                       'impute.Hz.vec',
                       'GP.Hz.vec',
                       'HR.Rhat.vec',
                       'a1_conv',
                       'a2_conv',
                       'a3_conv',
                       'r1_conv',
                       'r2_conv',
                       'r3_conv',
                       'r4_conv',
                       'r5_conv'
                      )]

df.m2.conv<-reshape2::melt(dat.HR.conv2[c(1:3, 5:12)],
                           id.var=c('impute.fac.vec',
                                     'impute.Hz.vec',
                                     'GP.Hz.vec'))

df.m2.conv$variable<-as.character(df.m2.conv$variable)
df.m2.conv$GP.Hz.vec<-as.factor(df.m2.conv$GP.Hz.vec)
df.m2.conv$impute.fac.vec<-as.character(df.m2.conv$impute.fac.vec)
df.m2.conv$impute.fac.vec[df.m2.conv$impute.fac.vec=="1"]<-'1x Imputation Window Length'
df.m2.conv$impute.fac.vec[df.m2.conv$impute.fac.vec=="2"]<-'2x Imputation Window Length'
df.m2.conv$impute.fac.vec[df.m2.conv$impute.fac.vec=="3"]<-'3x Imputation Window Length'

g125<-ggplot(data = df.m2.conv[df.m2.conv$impute.Hz.vec=='125'], aes(y = value, x = variable)) +
  geom_point(data = df.m2.conv[df.m2.conv$impute.Hz.vec=='125'],
            aes(x = variable, color = GP.Hz.vec),
            position = position_jitter(width = .15),
            size = 1, alpha = 0.5) +
  geom_boxplot(data = df.m2.conv[df.m2.conv$impute.Hz.vec=='125'],
              aes(x = variable, y = value, fill = GP.Hz.vec),
              width = .5, outlier.shape = NA, alpha = 0.5, inherit.aes = F) +
  #geom_point(data = df.m2.conv[df.m2.conv$variable!='GP1 Replacement' &
df.m2.conv$impute.Hz.vec=='125'],
            # aes(x = variable, y = value, color = variable),
            # position = position_jitter(width = .15),
            # size = 1,
            # alpha = 0.5,
            # inherit.aes = F,
            # guides = F) +
  #expand_limits(x = 4.25) +
  coord_cartesian(ylim = c(0,10))+
  #geom_hline(yintercept = 1.1, color='red', lty='dashed')+
  scale_color_brewer(palette = "Dark2") +
  scale_fill_brewer(palette = "Dark2", name = 'GP Model Hz') +
  theme_bw() +
  xlab('Hyperparameter')+ylab('Divergence in Estimates from 2 Random Starts')+
  guides(color=FALSE)+
  ggtitle('Distributions of Divergence in Hyperparameter Estimates from 2 Random Starts (125
Hz)')+
  #theme(axis.text.x = element_text(angle = 30, hjust = 1))+
  scale_x_discrete(labels = c('a1_conv' = expression(sigma[1]),
                              'a2_conv' = expression(sigma[2]),
                              'a3_conv' = expression(sigma[3]),
                              'r1_conv' = expression(italic('1')[1]),
                              'r2_conv' = expression(italic('1')[2]),
                              'r3_conv' = expression(italic('1')[3]),
                              'r4_conv' = expression(italic('1')[4]),
                              'r5_conv' = expression(italic('1')[5])))+
  facet_wrap(~impute.fac.vec)

#Looks Like a GP sample rate of 12 Hz with a factor of 2 window on either side is best...
#This may be unique the the GP 1 model specification...
#Will need to investigate more
```

```

g250<-ggplot(data = df.m2.conv[df.m2.conv$impute.Hz.vec=='250'], aes(y = value, x = variable)) +
  geom_point(data = df.m2.conv[df.m2.conv$impute.Hz.vec=='250'],
    aes(x = variable, color = GP.Hz.vec),
    position = position_jitter(width = .15),
    size = 1, alpha = 0.5) +
  geom_boxplot(data = df.m2.conv[df.m2.conv$impute.Hz.vec=='250'],
    aes(x = variable, y = value, fill = GP.Hz.vec),
    width = .5, outlier.shape = NA, alpha = 0.5, inherit.aes = F) +
  #geom_point(data = df.m2.conv[df.m2.conv$variable!='GP1 Replacement' &
df.m2.conv$impute.Hz.vec=='250'],,
  # aes(x = variable, y = value, color = variable),
  # position = position_jitter(width = .15),
  # size = 1,
  # alpha = 0.5,
  # inherit.aes = F,
  # guides = F) +
  #expand_limits(x = 4.25) +
  coord_cartesian(ylim = c(0,10))+
  #geom_hline(yintercept = 1.1, color='red', lty='dashed')+
  scale_color_brewer(palette = "Dark2") +
  scale_fill_brewer(palette = "Dark2", name = 'GP Model Hz') +
  theme_bw() +
  xlab('Hyperparameter')+ylab('Divergence in Estimates from 2 Random Starts')+
  guides(color=FALSE)+
  ggtitle('Distributions of Divergence in Hyperparameter Estimates from 2 Random Starts (250
Hz)')+
  #theme(axis.text.x = element_text(angle = 30, hjust = 1))+
  scale_x_discrete(labels = c('a1_conv' = expression(sigma[1]),
    'a2_conv' = expression(sigma[2]),
    'a3_conv' = expression(sigma[3]),
    'r1_conv' = expression(italic('1')[1]),
    'r2_conv' = expression(italic('1')[2]),
    'r3_conv' = expression(italic('1')[3]),
    'r4_conv' = expression(italic('1')[4]),
    'r5_conv' = expression(italic('1')[5])))+
  facet_wrap(~impute.fac.vec)

png(paste0(final.graphics, '/GP2_Hyperparameters.png'),
  res = 600,
  units = 'in',
  height = 10,
  width = 8)
cowplot::plot_grid(g125, g250, nrow = 2)
dev.off()

```

Appendix C.21: Code used to generate figure 5.5

```

dat.HR.conv3<-dat.GP3[c('impute.fac.vec',
  'impute.Hz.vec',
  'GP.Hz.vec',
  'HR.Rhat.vec',
  'a1_conv',
  'a2_conv',
  'a3_conv',
  'a4_conv',
  'r1_conv',
  'r2_conv',
  'r3_conv',
  'r4_conv',
  'r5_conv',
  'r6_conv',
  'r7_conv'
)]

df.m3.conv<-reshape2::melt(dat.HR.conv3[c(1:3, 5:15)],
  id.var=c('impute.fac.vec',

```

```

                                'impute.Hz.vec',
                                'GP.Hz.vec'))
df.m3.conv$variable<-as.character(df.m3.conv$variable)
df.m3.conv$GP.Hz.vec<-as.factor(df.m3.conv$GP.Hz.vec)
df.m3.conv$impute.fac.vec<-as.character(df.m3.conv$impute.fac.vec)
df.m3.conv$impute.fac.vec[df.m3.conv$impute.fac.vec=="1"]<-'1x Imputation Window Length'
df.m3.conv$impute.fac.vec[df.m3.conv$impute.fac.vec=="2"]<-'2x Imputation Window Length'
df.m3.conv$impute.fac.vec[df.m3.conv$impute.fac.vec=="3"]<-'3x Imputation Window Length'

g125<-ggplot(data = df.m3.conv[df.m3.conv$impute.Hz.vec=='125'], aes(y = value, x = variable)) +
  geom_point(data = df.m3.conv[df.m3.conv$impute.Hz.vec=='125'],
             aes(x = variable, color = GP.Hz.vec),
             position = position_jitter(width = .15),
             size = 1, alpha = 0.5) +
  geom_boxplot(data = df.m3.conv[df.m3.conv$impute.Hz.vec=='125'],
              aes(x = variable, y = value, fill = GP.Hz.vec),
              width = .75, outlier.shape = NA, alpha = 0.5, inherit.aes = F) +
  #geom_point(data = df.m3.conv[df.m3.conv$variable!='GP1 Replacement' &
df.m3.conv$impute.Hz.vec=='125'],
             # aes(x = variable, y = value, color = variable),
             # position = position_jitter(width = .15),
             # size = 1,
             # alpha = 0.5,
             # inherit.aes = F,
             # guides = F) +
  #expand_limits(x = 4.25) +
  coord_cartesian(ylim = c(0,10))+
  #geom_hline(yintercept = 1.1, color='red', lty='dashed')+
  scale_color_brewer(palette = "Dark2") +
  scale_fill_brewer(palette = "Dark2", name = 'GP Model Hz') +
  theme_bw() +
  xlab('Hyperparameter')+ylab('Divergence in Estimates from 2 Random Starts')+
  guides(color=FALSE)+
  ggtitle('Distributions of Divergence in Hyperparameter Estimates from 2 Random Starts (125
Hz)')+
  #theme(axis.text.x = element_text(angle = 30, hjust = 1))+
  scale_x_discrete(labels = c('a1_conv' = expression(sigma[1]),
                              'a2_conv' = expression(sigma[2]),
                              'a3_conv' = expression(sigma[3]),
                              'a4_conv' = expression(sigma[3]),
                              'r1_conv' = expression(italic('1')[1]),
                              'r2_conv' = expression(italic('1')[2]),
                              'r3_conv' = expression(italic('1')[3]),
                              'r4_conv' = expression(italic('1')[4]),
                              'r5_conv' = expression(italic('1')[5]),
                              'r6_conv' = expression(italic('1')[6]),
                              'r7_conv' = expression(italic('1')[7])))+
  facet_wrap(~impute.fac.vec)

#Looks Like a GP sample rate of 12 Hz with a factor of 2 window on either side is best...
#This may be unique the the GP 1 model specification...
#Will need to investigate more
g250<-ggplot(data = df.m3.conv[df.m3.conv$impute.Hz.vec=='250'], aes(y = value, x = variable)) +
  geom_point(data = df.m3.conv[df.m3.conv$impute.Hz.vec=='250'],
             aes(x = variable, color = GP.Hz.vec),
             position = position_jitter(width = .15),
             size = 1, alpha = 0.5) +
  geom_boxplot(data = df.m3.conv[df.m3.conv$impute.Hz.vec=='250'],
              aes(x = variable, y = value, fill = GP.Hz.vec),
              width = .75, outlier.shape = NA, alpha = 0.5, inherit.aes = F) +
  #geom_point(data = df.m3.conv[df.m3.conv$variable!='GP1 Replacement' &
df.m3.conv$impute.Hz.vec=='250'],
             # aes(x = variable, y = value, color = variable),
             # position = position_jitter(width = .15),
             # size = 1,
             # alpha = 0.5,
             # inherit.aes = F,
             # guides = F) +

```



```

#expand_limits(x = 4.25) +
coord_cartesian(ylim = c(0,10))+
#geom_hline(yintercept = 1.1, color='red', lty='dashed')+
scale_color_brewer(palette = "Dark2") +
scale_fill_brewer(palette = "Dark2", name = 'GP Model Hz') +
theme_bw() +
xlab('Hyperparameter')+ylab('Divergence in Estimates from 2 Random Starts')+
guides(color=FALSE)+
ggtitle('Distributions of Divergence in Hyperparameter Estimates from 2 Random Starts (250
Hz)')+
#theme(axis.text.x = element_text(angle = 30, hjust = 1))+
scale_x_discrete(labels = c('a1_conv' = expression(sigma[1]),
'a2_conv' = expression(sigma[2]),
'a3_conv' = expression(sigma[3]),
'a4_conv' = expression(sigma[3]),
'r1_conv' = expression(italic('1')[1]),
'r2_conv' = expression(italic('1')[2]),
'r3_conv' = expression(italic('1')[3]),
'r4_conv' = expression(italic('1')[4]),
'r5_conv' = expression(italic('1')[5]),
'r6_conv' = expression(italic('1')[6]),
'r7_conv' = expression(italic('1')[7])))+
facet_wrap(~impute.fac.vec)

png(paste0(final.graphics, '/GP3_Hyperparameters.png'),
res = 600,
units = 'in',
height = 10,
width = 10)
cowplot::plot_grid(g125, g250, nrow = 2)
dev.off()

```

Appendix C.22: Code used to generate figure 5.8

```

#Getting a distribution plot for the Cardio Edit scores & IBI VizEdit Scores
df.m<-reshape2::melt(dat.intra)
df.m$variable<-as.character(df.m$variable)
df.m$variable[df.m$variable=='VizEdit_RSA15']<- 'VizEdit'
df.m$variable[df.m$variable=='CardioEdit_RSA15']<- 'Cardio Edit'

g1<-ggplot(data = df.m[df.m$variable!='Diff',], aes(y = value, x = variable, fill=variable)) +
geom_flat_violin(position = position_nudge(x = .2, y = 0), alpha = .8) +
geom_point(aes(y = value, color = variable),
position = position_jitter(width = .15),
size = 1, alpha = 0.8) +
geom_boxplot(width = .1, guides = FALSE, outlier.shape = NA, alpha = 0.5) +
expand_limits(x = 3.25) +
scale_color_brewer(palette = "Dark2") +
scale_fill_brewer(palette = "Dark2") +
theme_bw() +
raincloud_theme+
xlab('')+ylab('RSA')+
guides(fill=FALSE, color=FALSE)+
ggtitle('RSA Score Distributions Using Cardio Edit and IBI VizEdit')

png(paste0(EDA.folder, '/Cross-Program_RSA.png'), units = 'in',
height = 8, width = 8, res = 300)
g1
dev.off()

```

Appendix C.23: Code used to generate figure 5.9

```
png(paste0(EDA.folder, '/Phyio Variables.png'),  
    units = 'in',  
    height = 8,  
    width = 8,  
    res = 600)  
psych::pairs.panels(Long.dat)  
dev.off()
```

Appendix D: Code for Re-Sampling Program

Appendix D.1 Re-sampling Code for Gaussian Process Model Imputation Model 1

```
#Ensuring random starts for two different threads on same computer...
set.seed(as.numeric(Sys.time()))

#Original Processing:

dat.time.orig<-read.csv('~/.Dropbox/Dissertation/Chapter 4/Sim_data/Raw_coded.csv',
                        stringsAsFactors = F)

#cleaning up extra rows
dat.time.orig<-dat.time.orig[-91:-nrow(dat.time.orig),]

#Taking only cases with at least 180 continuous seconds worth of data:
dat.time.orig<-dat.time.orig[dat.time.orig$Window_Length>=180,]

table(dat.time.orig$Population) #not bad - about 50/50
                                #40 sections of adult data
                                #39 sections of child data

#Converting "Adult" to "adult" and "Child" to "child"
dat.time.orig$Population<-ifelse(dat.time.orig$Population=="Adult", 'adult', 'child')

wd<-'~/Dropbox/Dissertation/Chapter 4/Sim_data'

#Bringing in file information worksheet
file_info<-read.csv(paste0(wd, '/Physio_Tracking - Sheet1.csv'), stringsAsFactors = F)

#Converting ID to common structure:
file_info$ID<-substr(file_info$Video_1, start=1, stop=6)

#Task timing file
task.time<-read.table(paste0(wd, '/Timing_File_forVizEdit.txt'),
                      header = T, sep = '\t')
colnames(task.time)[1]<-'ID'

#Creating a test case folder - will run five imputations under each condition on 5 different
files:
#ALL test files will be with adult files.
data.folder<-paste0(wd, '/Good_3min')

file.names<-list.files(data.folder)

#Getting appropriate timing values for sections of "good" data
DF.segment<-merge(dat.time.orig, task.time[,1:2], by='ID')

#Will not be able to use case 051_T2 as there appears to be a task/timing issue there (Loses 1
child case)
DF.segment<-DF.segment[DF.segment$ID!="051_T2",]
DF.segment$Start.adj<-DF.segment$Start+DF.segment$Video
DF.segment$Stop.adj<-DF.segment$Stop+DF.segment$Video
DF.segment$time.min<-DF.segment$Start.adj+31
DF.segment$time.max<-DF.segment$Stop.adj-31

DF.segment<-merge(DF.segment, file_info[,c(1,14:17)], by='ID')

#####
#####
#Functions for identifying heart beats
#=====
#Function 1 - Finding Peakings Using Specified bandwidth:
findpeaks <- function (x, m = 3){
  shape <- diff(sign(diff(x, na.pad = FALSE)))
  pks <- sapply(which(shape < 0), FUN = function(i){
    z <- i - m + 1
    z <- ifelse(z > 0, z, 1)
  })
}
```

```

w <- i + m + 1
w <- ifelse(w < length(x), w, length(x))
if(all(x[c(z : i, (i + 2) : w)] <= x[i + 1])) return(i + 1) else return(numeric(0))
})
pks <- unlist(pks)
pks
}
#####
#Function 2 - Summing IBIs from Raw PPG file:
time.sum<-function(x){
Z<-rep(NA, length(x))
for(i in 1:length(x)){
Z[i]<-ifelse(i==1, x[i], x[i]-x[i-1])
}
return(Z)
}
#####
#Function 2b - Summing Time from IBIs
IBI.sum<-function(x){
Z<-rep(NA, length(x))
for(i in 1:length(x)){
Z[i]<-sum(x[1:i])
}
return(Z)
}
#####
#Function 3 - Iterative function for getting IBIs
iter.IBI<-function(x, ds=2000){
#browser()
require(psych)
s<-round(seq(round(ds/50), round(ds/2), length.out = 200))
Z<-data.frame(rep(NA, length(s)),
rep(NA, length(s)),
rep(NA, length(s)),
rep(NA, length(s)),
rep(NA, length(s)),
rep(NA, length(s)))
for(i in 1:length(s)){
IBI<-findpeaks(x, s[i])
time<-time.sum(IBI)/ds
Z[i,1]<-s[i]
Z[i,2]<-sd(time)
Z[i,3]<-max(time)-min(time)
Z[i,4]<-rmssd(time)
Z[i,5]<-mean(acf(time, lag.max = length(time)/20, plot = F)$acf)
Z[i,6]<-s[i]/ds
}
colnames(Z)<-c('BW', 'SD', 'Range', 'RMSSD', 'AC', 'BW(s)')
Z<-Z[order(Z$RMSSD, decreasing = F),]
IBI.fin<-findpeaks(x, m=Z[1,1])-1
IBI.fin<-IBI.fin/ds
IBI.done<-time.sum(IBI.fin)
IBI.comp<-list(IBI.done, Z)
names(IBI.comp)<-c('IBI.done', 'Z')
return(IBI.comp)
}
#####
#Function 4 - Obtaining Time Values for IBI
sum.rev<-function(x){
Z<-rep(NA, length(x))
for(i in 1:length(x)){
Z[i]<-ifelse(i==1, x[i], sum(x[1:(i-1)])+x[i])
}
return(Z)
}
#####
#setting up models:
chains<-2

```

```

impute.min<-2
impute.max<-8
warmup<-c(2000)
n.iter<-c(2500)
Data.sample.Hz<-c(2000)
impute.sample.Hz<-c(125, 250)
GP.sample.Hz<-c(4, 8, 12)
impute.fac<-c(1,2,3)

library(ggplot2)
library(signal)
library(psych)
library(bayesplot)
library(MCMCvis)
library(rstan)
library(rstanarm)
library(astsa)
options(mc.cores=parallel::detectCores())
rstan_options(auto_write = TRUE)
#Run the complete simulation from here
#Basing this on 5 simulations each to start:
n.sims<-20

#Basic simulation vectors
ID.vec<-vector()
impute.ID<-vector()
pop.vec<-vector()
start.vec<-vector()
stop.vec<-vector()
CPU.vec<-vector()
Cores.vec<-vector()
RAM.vec<-vector()
IBI.rmssd.15<-vector()
IBI.rmssd.30<-vector()
IBI.rmssd.45<-vector()
IBI.rmssd.60<-vector()
IBI.MR.rmssd.15<-vector()
IBI.MR.rmssd.30<-vector()
IBI.MR.rmssd.45<-vector()
IBI.MR.rmssd.60<-vector()
IBI.HD.rmssd.15<-vector()
IBI.HD.rmssd.30<-vector()
IBI.HD.rmssd.45<-vector()
IBI.HD.rmssd.60<-vector()
IBI.GP.rmssd.15<-vector()
IBI.GP.rmssd.30<-vector()
IBI.GP.rmssd.45<-vector()
IBI.GP.rmssd.60<-vector()
N.IBIs.vec<-vector()
runtime<-vector()
warmup.vec<-vector()
iter.vec<-vector()
Hz.vec<-vector()
impute.Hz.vec<-vector()
GP.Hz.vec<-vector()
impute.fac.vec<-vector()
impute.start<-vector()
impute.end<-vector()
impute.tot<-vector()

#Parameter Means
HR.mean.vec<-vector()

#Parameter sd's
HR.SD.vec<-vector()

#Parameter R-hats
HR.Rhat.vec<-vector()

```

```

#Parameter N_eff
#Parameter Means
HR.N_eff.vec<-vector()

#Simple alpha convergence check
a1.ML.conv.vec<-vector()
a2.ML.conv.vec<-vector()

#Simple rho convergence check
r1.ML.conv.vec<-vector()
r2.ML.conv.vec<-vector()
r3.ML.conv.vec<-vector()

#Creating folder structure to allow for quick saving of Files
out.folder<-paste0(wd, '/Output')
IBI.orig.folder<-paste0(out.folder, '/IBI_Files/Original')
IBI.MR.folder<-paste0(out.folder, '/IBI_Files/Mean_Replacement')
IBI.HD.folder<-paste0(out.folder, '/IBI_Files/Hot_Deck')
IBI.impute.folder<-paste0(out.folder, '/IBI_Files/Imputed')
Peak.summary.folder<-paste0(out.folder, '/Peak_Detection_Summaries')
graphics.folder<-paste0(out.folder, '/Graphics')
Bayes.graphics<-paste0(graphics.folder, '/Bayes_Graphics')
model.folder<-paste0(out.folder, '/Model_Summaries')
diagnostics.folder<-paste0(out.folder, '/Diagnostics')

#Main simulation program:
for(i in 1:n.sims){
  #browser()
  #Getting the specifics for each run:
  impute.window.temp<-runif(1, impute.min, impute.max)
  row.select.temp<-sample(x=1:nrow(DF.segment), size = 1)
  dat.temp<-DF.segment[row.select.temp,]
  header.temp<-ifelse(dat.temp$Population=='child', dat.temp$Child_Header, dat.temp$Parent_Header)
  col.select.temp<-ifelse(dat.temp$Population=='child', dat.temp$Child_Column,
dat.temp$Parent_Column)

  #Selecting imputation boundaries
  impute.LB.temp<-runif(1, dat.temp$time.min, dat.temp$time.max)
  impute.UB.temp<-impute.LB.temp+impute.window.temp

  #getting PPG data
  PPG<-read.table(paste0(data.folder, '/', dat.temp$ID, '.txt'),
                 skip = header.temp,
                 header = F,
                 sep = '\t'
  )

  PPG<-PPG[,col.select.temp]

  PPG<-data.frame(PPG,
                 Time = seq(from = 0, by = .0005, length.out = length(PPG))
  )

  #Selecting target window for the case
  PPG.temp<-PPG[PPG$Time>=dat.temp$Start.adj & PPG$Time<=dat.temp$Stop.adj,]

  #Cleaning signal - de-spiking and smoothing heart rate
  PPG.temp$PPG<-as.numeric(smooth(PPG.temp$PPG))
  PPG.temp$PPG<-smooth.spline(PPG.temp$PPG, nknots = 10000)$y
  PPG.temp$PPG<-PPG.temp$PPG-predict(lm(PPG~Time, data = PPG.temp))

  #Obtaining original IBI values for section:
  IBI.orig.temp<-iter.IBI(PPG.temp$PPG, ds=2000)

  write.table(round(IBI.orig.temp$IBI.done, digits = 4),
             paste0(
               out.folder,
               '/',
               paste(dat.temp$ID,
                     dat.temp$Population,

```

```

        dat.temp$Start,
        dat.temp$Stop,
        round(impute.LB.temp, digits = 2),
        sep = '_'),
        '.txt'),
        row.names = F
    )

write.table(round(head(IBE.orig.temp$Z, n=20), digits = 3),
            paste0(Peak.summary.folder,
                  '/',
                  paste(dat.temp$ID,
                        dat.temp$Population,
                        'original',
                        dat.temp$Start,
                        dat.temp$Stop,
                        round(impute.LB.temp, digits = 2),
                        sep = '_'),
                  '.txt'
            ),
            sep = '\t',
            row.names = F
        )

#Mean replacement strategy - Will replace the number of IBIs in the affected range with mean values
#Step 1 - Restoring accurate timing file

IBE.orig.time<-sum.rev(IBE.orig.temp$IBE.done)+dat.temp$Start.adj

#Running very simple mean imputation (not accounting for time-series nature of the data)
IBE.mean.replace<-IBE.orig.temp$IBE.done
IBE.mean.replace[IBE.orig.time>impute.LB.temp &
                 IBE.orig.time<impute.UB.temp]<-rep(mean(IBE.orig.temp$IBE.done))

#removing first and last IBI value (to make more equivalent comparison with imputed data set)
write.table(round(IBE.mean.replace[c(-1, -length(IBE.mean.replace))], digits = 4),
            paste0(IBE.MR.folder,
                  '/',
                  paste(dat.temp$ID,
                        dat.temp$Population,
                        'Mean_replace',
                        dat.temp$Start,
                        dat.temp$Stop,
                        round(impute.LB.temp, digits = 2),
                        sep = '_'),
                  '.txt'),
            row.names = F
        )

#Running hot deck imputation (does not account for time-series nature of the data)
IBE.sample.vals<-IBE.orig.temp$IBE.done[IBE.orig.time<impute.LB.temp |
IBE.orig.time>impute.UB.temp]
IBE.hotdeck.replace<-IBE.orig.temp$IBE.done
N.IBIs<-length(IBE.hotdeck.replace[IBE.orig.time>impute.LB.temp & IBE.orig.time<impute.UB.temp])
IBE.hotdeck.replace[IBE.orig.time>impute.LB.temp & IBE.orig.time<impute.UB.temp]<-
sample(x=IBE.sample.vals,
                                             size =
N.IBIs,
                                             replace
= T)
write.table(round(IBE.hotdeck.replace[c(-1, -length(IBE.hotdeck.replace))], digits = 4),
            paste0(IBE.HD.folder,
                  '/',
                  paste(dat.temp$ID,
                        dat.temp$Population,
                        'HotDeck_replace',
                        dat.temp$Start,
                        dat.temp$Stop,

```

```

        round(impute.LB.temp, digits = 2),
        sep = '_'),
        '.txt'),
        row.names = F
    )

for(j in 1:length(chains)){
  chain.temp<-chains[j]
  for(k in 1:length(warmup)){
    warmup.temp<-warmup[k]
    n.iter.temp<-n.iter[k]
    for(l in 1:length(Data.sample.Hz)){
      Hz.temp<-Data.sample.Hz[l]
      for(m in 1:length(impute.sample.Hz)){
        Hz.impute.temp<-impute.sample.Hz[m]
        for(n in 1:length(GP.sample.Hz)){
          Hz.GP.temp<-GP.sample.Hz[n]
          for(o in 1:length(impute.fac)){
            impute.fac.temp<-impute.fac[o]

            #Downsampling (if appropriate)
            select.int<-2000/Hz.temp
            PPG.impute<-PPG.temp[seq(1, nrow(PPG.temp), select.int),]
            IBI.impute<-iter.IBI(PPG.impute$PPG, ds=Hz.temp)

            #Getting some priors for the model
            mu_HP<-mean(1/IBI.impute$IBI.done[c(-1,-length(IBI.impute$IBI.done))], na.rm = T)
            sigma_HP<-sd(1/IBI.impute$IBI.done[c(-1,-length(IBI.impute$IBI.done))], na.rm = T)

            chain.temp<-2

            init.list<-list()
            for(x in 1:chain.temp){
              init.list[[x]]<-list(mu_HR = mu_HP)
            }

            if(Hz.temp>=Hz.impute.temp){
              #First selecting a series of time values to impute back in at the appropriate
              sampling rate
              select.int2<-Hz.temp/Hz.impute.temp
              PPG.impute2<-PPG.impute[seq(1, nrow(PPG.impute), select.int2),]
              PPG.impute.pred<-PPG.impute2[PPG.impute2$Time>impute.LB.temp &
                PPG.impute2$Time<impute.UB.temp,]

              #Getting what will be Prediction values:
              Xp<-PPG.impute.pred$Time
              N2<-length(PPG.impute.pred$Time)

              #Acquiring "good data"
              min.TIME2<-min(Xp)
              max.TIME2<-max(Xp)
              time.span<-max.TIME2-min.TIME2
              Y.vals<-rbind(PPG.impute2[PPG.impute2$Time>min.TIME2-impute.fac.temp*time.span &
                PPG.impute2$Time<min.TIME2,],
                PPG.impute2[PPG.impute2$Time>max.TIME2 &
                PPG.impute2$Time<max.TIME2+impute.fac.temp*time.span,])
              Y.vals<-na.omit(Y.vals)
              tot.Y.vals<-length(Y.vals[,1])
              sel.Y.vals<-round(seq(1, tot.Y.vals, length.out =
              round(tot.Y.vals/Hz.impute.temp*Hz.GP.temp)))
              sel.Y.vals<-unique(sel.Y.vals)
              Y<-Y.vals$PPG[sel.Y.vals]
              X<-Y.vals$Time[sel.Y.vals]
              N1<-length(X)

              #Running Model - there is no need for respiration priors
              dat<-list(N1=N1,
                N2=N2,

```



```

        Xp=Xp,
        X=X,
        Y=Y,
        mu_HR=mu_HP,
        sigma_HR=sigma_HP
    )

    #Obtaining estimates for hyper-parameters using ML approach
    opt_model<-stan_model(file=paste0(wd, '/Stan_code/GP_1_opt.stan'))

    #need to find some way to assess convergence...
    #Identifying values for hyperparameters
    print('Obtaining first ML estimate for hyperparameters')
    opt_fit1<-NULL
    attempt<-1
    while(is.null(opt_fit1) & attempt <=15){
        print(paste('Convergence attempt', attempt, 'out of 15'))
        attempt<- attempt + 1
        try(
            opt_fit1<-optimizing(opt_model,
                                data=dat,
                                init=list(mu_HR = mu_HP),
                                seed=sample(1:5000, size = 1), iter = 10000)
        )
    }

    print('Obtaining second ML estimate for hyperparameters')
    opt_fit2<-NULL
    attempt<-1
    while(is.null(opt_fit2) & attempt <=15){
        print(paste('Convergence attempt', attempt, 'out of 15'))
        attempt<- attempt + 1
        try(
            opt_fit2<-optimizing(opt_model,
                                data=dat,
                                init=list(mu_HR = mu_HP),
                                seed=sample(1:5000, size = 1), iter = 10000)
        )
    }

    if(!is.null(opt_fit1)){
        alpha1.1 <- opt_fit1$par['a1']
        alpha2.1 <- opt_fit1$par['a2']
        rho1.1 <- opt_fit1$par['r1']
        rho2.1 <- opt_fit1$par['r2']
        rho3.1 <- opt_fit1$par['r3']
    }

    if(!is.null(opt_fit2)){
        alpha1.2 <- opt_fit2$par['a1']
        alpha2.2 <- opt_fit2$par['a2']
        rho1.2 <- opt_fit2$par['r1']
        rho2.2 <- opt_fit2$par['r2']
        rho3.2 <- opt_fit2$par['r3']
    }

    if(!is.null(opt_fit1) & !is.null(opt_fit2)){
        #Simple Alpha convergence check
        a1.ML.conv.vec<-c(a1.ML.conv.vec,(abs(alpha1.1-alpha1.2)<.0001))
        a2.ML.conv.vec<-c(a2.ML.conv.vec,(abs(alpha2.1-alpha2.2)<.0001))

        #Simple rho convergence check
        r1.ML.conv.vec<-c(r1.ML.conv.vec,(abs(rho1.1-rho1.2)<.0001))
        r2.ML.conv.vec<-c(r2.ML.conv.vec,(abs(rho2.1-rho2.2)<.0001))
        r3.ML.conv.vec<-c(r3.ML.conv.vec,(abs(rho3.1-rho3.2)<.0001))
    }

    else{
        a1.ML.conv.vec<-c(a1.ML.conv.vec, NA)
    }

```

```

a2.ML.conv.vec<-c(a2.ML.conv.vec, NA)

#Simple rho convergence check
r1.ML.conv.vec<-c(r1.ML.conv.vec, NA)
r2.ML.conv.vec<-c(r2.ML.conv.vec, NA)
r3.ML.conv.vec<-c(r3.ML.conv.vec, NA)
}

if(is.null(opt_fit1) & !is.null(opt_fit2)){
  alpha1.1<-alpha1.2
  alpha2.1<-alpha2.2
  rho1.1<-rho1.2
  rho2.1<-rho2.2
  rho3.1<-rho3.2
}

pars.to.monitor<-c('HR', 'Ypred')
start.time.temp<-Sys.time()

dat.opt<-list(N1=N1,
             N2=N2,
             Xp=Xp,
             X=X,
             Y=Y,
             mu_HR=mu_HP,
             sigma_HR=sigma_HP,
             a1 = alpha1.1,
             a2 = alpha2.1,
             r1 = rho1.1,
             r2 = rho2.1,
             r3 = rho3.1
            )

fit.stan<-stan(file=paste0(wd, '/Stan_code/GP_1.stan'),
              data = dat.opt,
              init = init.list,
              warmup = warmup.temp,
              iter = n.iter.temp,
              refresh = 100,
              chains = 2,
              pars = pars.to.monitor,
              control = list(adapt_delta = .95,
                             max_treedepth = 15)
             )

run_time<-round(Sys.time()-start.time.temp, digits = 3)
units(run_time)<-'mins'
run_time<-as.numeric(run_time)

y_pred<-rstan::extract(fit.stan, 'Ypred')
PPG.new<-colMeans(y_pred$Ypred)
PPG.DF.new<-data.frame(PPG = PPG.new,
                      Time = Xp)

GP.summary<-as.data.frame(summary(fit.stan, pars = pars.to.monitor[-
length(pars.to.monitor)]),
                           probs = c(.1, .9))$summary)

write.table(round(GP.summary, digits = 3),
            paste0(model.folder,
                  '/',
                  paste(dat.temp$ID,
                        dat.temp$Population,
                        dat.temp$Segment,
                        'GP_impute',
                        round(impute.LB.temp, digits = 2),
                        round(impute.UB.temp, digits = 2),
                        Hz.temp,
                        Hz.impute.temp,

```

```

        Hz.GP.temp,
        impute.fac.temp,
        chain.temp,
        warmup.temp,
        n.iter.temp,
        sep = '_'),
    '.txt'),
  row.names = T,
  sep='\t'
)

g1<-ggplot()+
  geom_line(data=PPG.DF.new, aes(x=Time, y = PPG), color = 'red', lty='dashed')+
  geom_line(data=PPG.impute.pred, aes(x=Time, y=PPG), color = 'black')+
  ggtitle(paste(dat.temp$ID, dat.temp$Population, dat.temp$Segment,
    'File Hz =', Hz.temp, 'GP Dataset Hz =', Hz.impute.temp,
    'Model Sampling Rate =', Hz.GP.temp, 'Chains =', chain.temp,
    'Warmup =', warmup.temp, 'Iter =', n.iter.temp))

ggsave(filename = paste0(graphics.folder, paste('/',
  dat.temp$ID,
  dat.temp$Population,
  dat.temp$Segment,
  Hz.temp,
  Hz.impute.temp,
  Hz.GP.temp,
  impute.fac.temp,
  chain.temp,
  warmup.temp,
  n.iter.temp,
  round(impute.LB.temp, digits = 2),
  sep = '_'), '.png'),

  plot = g1,
  width = 11,
  height = 8,
  units = 'in',
  dpi = 300,
  device = 'png')

g2<-traceplot(fit.stan, pars='HR', inc_warmup=T)+
  ggtitle(paste('HR param', dat.temp$ID, dat.temp$Population, dat.temp$Segment,
    'File Hz =', Hz.temp, 'GP Dataset Hz =', Hz.impute.temp,
    'Model Sampling Rate =', Hz.GP.temp, '\n', 'Chains =', chain.temp,
    'Warmup =', warmup.temp, 'Iter =', n.iter.temp))

ggsave(filename = paste0(Bayes.graphics, paste('/',
  dat.temp$ID,
  dat.temp$Population,
  dat.temp$Segment,
  Hz.temp,
  Hz.impute.temp,
  Hz.GP.temp,
  impute.fac.temp,
  chain.temp,
  warmup.temp,
  n.iter.temp,
  round(impute.LB.temp, digits = 2),
  sep = '_'), '.png'),

  plot = g2,
  width = 11,
  height = 8,
  units = 'in',
  dpi = 300,
  device = 'png')

PPG.impute.fin<-PPG.impute2
PPG.impute.fin$PPG[PPG.impute.fin$Time>impute.LB.temp &
  PPG.impute.fin$Time<impute.UB.temp]<-PPG.new

```

```

IBI.GP.impute.temp<-iter.IBI(PPG.impute.fin$PPG, ds=Hz.impute.temp)$IBI.done
IBI.GP.impute.time<-sum.rev(IBI.GP.impute.temp)+dat.temp$Start.adj
write.table(round(IBI.GP.impute.temp[c(-1, -length(IBI.GP.impute.temp))], digits =
4),
            paste0(IBI.impute.folder,
                    '/',
                    paste(dat.temp$ID,
                          dat.temp$Population,
                          dat.temp$Segment,
                          'GP_impute',
                          round(impute.LB.temp, digits = 2),
                          round(impute.UB.temp, digits = 2),
                          Hz.temp,
                          Hz.impute.temp,
                          Hz.GP.temp,
                          impute.fac.temp,
                          chain.temp,
                          warmup.temp,
                          n.iter.temp,
                          sep = '_'),
                    '.txt'),
            row.names = F
)

HR.est<-rstan::extract(fit.stan, 'HR')

#Basic simulation vectors
ID.vec<-c(ID.vec, dat.temp$ID)
impute.ID<-c(impute.ID, paste('Segment', i, sep = '_'))
pop.vec<-c(pop.vec, dat.temp$Population)
CPU.vec<-c(CPU.vec, benchmarkme::get_cpu()$model_name)
Cores.vec<-c(Cores.vec, parallel::detectCores())
RAM.vec<-c(RAM.vec, paste(round(benchmarkme::get_ram()/1073741824), 'GB'))
start.vec<-c(start.vec, dat.temp$Start)
stop.vec<-c(stop.vec, dat.temp$Stop)
N.IBIs.vec<-c(N.IBIs.vec, N.IBIs)
IBI.GP.rmssd.15<-c(IBI.GP.rmssd.15,
                  rmssd(IBI.GP.impute.temp[IBI.GP.impute.time>impute.LB.temp-7.5
&
IBI.GP.impute.time<impute.UB.temp+7.5]))
IBI.GP.rmssd.30<-c(IBI.GP.rmssd.30,
                  rmssd(IBI.GP.impute.temp[IBI.GP.impute.time>impute.LB.temp-15 &
IBI.GP.impute.time<impute.UB.temp+15]))
IBI.GP.rmssd.45<-c(IBI.GP.rmssd.45,
                  rmssd(IBI.GP.impute.temp[IBI.GP.impute.time>impute.LB.temp-22.5
&
IBI.GP.impute.time<impute.UB.temp+22.5]))
IBI.GP.rmssd.60<-c(IBI.GP.rmssd.60,
                  rmssd(IBI.GP.impute.temp[IBI.GP.impute.time>impute.LB.temp-30 &
IBI.GP.impute.time<impute.UB.temp+30]))
runtime<-c(runtime, run_time)
warmup.vec<-c(warmup.vec, warmup.temp)
iter.vec<-c(iter.vec, n.iter.temp)
Hz.vec<-c(Hz.vec, Hz.temp)
impute.Hz.vec<-c(impute.Hz.vec, Hz.impute.temp)
GP.Hz.vec<-c(GP.Hz.vec, Hz.GP.temp)
impute.fac.vec<-c(impute.fac.vec, impute.fac.temp)
impute.start<-c(impute.start, min.TIME2)
impute.end<-c(impute.end, max.TIME2)
impute.tot<-c(impute.tot, time.span)

#Parameter Means
HR.mean.vec<-c(HR.mean.vec, mean(HR.est$HR))

```

```

#Parameter sd's
HR.SD.vec<-c(HR.SD.vec, sd(HR.est$HR))

#Parameter R-hats
HR.Rhat.vec<-c(HR.Rhat.vec, GP.summary$Rhat[rownames(GP.summary)=='HR'])

#Parameter N_eff
HR.N_eff.vec<-c(HR.N_eff.vec, GP.summary$n_eff[rownames(GP.summary)=='HR'])

#Getting a vector of original value rmssd's
IBI.rmssd.15<-c(IBI.rmssd.15,
               rmssd(IBI.orig.temp$IBI.done[IBI.orig.time>impute.LB.temp-7.5 &
               IBI.orig.time<impute.UB.temp+7.5]))
IBI.rmssd.30<-c(IBI.rmssd.30,
               rmssd(IBI.orig.temp$IBI.done[IBI.orig.time>impute.LB.temp-15 &
               IBI.orig.time<impute.UB.temp+15]))
IBI.rmssd.45<-c(IBI.rmssd.45,
               rmssd(IBI.orig.temp$IBI.done[IBI.orig.time>impute.LB.temp-22.5 &
               IBI.orig.time<impute.UB.temp+22.5]))
IBI.rmssd.60<-c(IBI.rmssd.60,
               rmssd(IBI.orig.temp$IBI.done[IBI.orig.time>impute.LB.temp-30 &
               IBI.orig.time<impute.UB.temp+30]))

#Getting a vector of mean replacement rmssd's
IBI.MR.rmssd.15<-c(IBI.MR.rmssd.15,
                  rmssd(IBI.mean.replace[IBI.orig.time>impute.LB.temp-7.5 &
                  IBI.orig.time<impute.UB.temp+7.5]))
IBI.MR.rmssd.30<-c(IBI.MR.rmssd.30,
                  rmssd(IBI.mean.replace[IBI.orig.time>impute.LB.temp-15 &
                  IBI.orig.time<impute.UB.temp+15]))
IBI.MR.rmssd.45<-c(IBI.MR.rmssd.45,
                  rmssd(IBI.mean.replace[IBI.orig.time>impute.LB.temp-22.5 &
                  IBI.orig.time<impute.UB.temp+22.5]))
IBI.MR.rmssd.60<-c(IBI.MR.rmssd.60,
                  rmssd(IBI.mean.replace[IBI.orig.time>impute.LB.temp-30 &
                  IBI.orig.time<impute.UB.temp+30]))

#Getting a vector of hotdeck replacement rmssd's
IBI.HD.rmssd.15<-c(IBI.HD.rmssd.15,
                  rmssd(IBI.hotdeck.replace[IBI.orig.time>impute.LB.temp-7.5 &
                  IBI.orig.time<impute.UB.temp+7.5]))
IBI.HD.rmssd.30<-c(IBI.HD.rmssd.30,
                  rmssd(IBI.hotdeck.replace[IBI.orig.time>impute.LB.temp-15 &
                  IBI.orig.time<impute.UB.temp+15]))
IBI.HD.rmssd.45<-c(IBI.HD.rmssd.45,
                  rmssd(IBI.hotdeck.replace[IBI.orig.time>impute.LB.temp-22.5 &
                  IBI.orig.time<impute.UB.temp+22.5]))
IBI.HD.rmssd.60<-c(IBI.HD.rmssd.60,
                  rmssd(IBI.hotdeck.replace[IBI.orig.time>impute.LB.temp-30 &
                  IBI.orig.time<impute.UB.temp+30]))

}
}
}
}
}
}
}
}
}
}

save(list = ls(),
      file = paste0(out.folder, '/', paste('GP_1', Sys.time(), n.sims, sep = '_'), '.RData'))

```

Appendix D.2 Re-sampling Code for Gaussian Process Model Imputation Model 2

```
#Ensuring random starts for two different threads on same computer...
set.seed(as.numeric(Sys.time()))

#Original Processing:

dat.time.orig<-read.csv('/home/mbarsted/Dropbox/Dissertation/Chapter
4/Sim_data_2_b/Raw_coded.csv',
                      stringsAsFactors = F)

#cleaning up extra rows
dat.time.orig<-dat.time.orig[-91:-nrow(dat.time.orig),]

#Taking only cases with at Least 180 continuous seconds worth of data:
dat.time.orig<-dat.time.orig[dat.time.orig$Window_Length>=180,]

table(dat.time.orig$Population) #not bad - about 50/50
#40 sections of adult data
#39 sections of child data

#Converting "Adult" to "adult" and "Child" to "Child"
dat.time.orig$Population<-ifelse(dat.time.orig$Population=='Adult', 'adult', 'child')

wd<-'/home/mbarsted/Dropbox/Dissertation/Chapter 4/Sim_data_2_b'

#Bringing in file information worksheet
file_info<-read.csv(paste0(wd, '/Physio_Tracking - Sheet1.csv'), stringsAsFactors = F)

#Converting ID to common structure:
file_info$ID<-substr(file_info$Video_1, start=1, stop=6)

#Task timing file
task.time<-read.table(paste0(wd, '/Timing_File_forVizEdit.txt'),
                    header = T, sep = '\t')
colnames(task.time)[1]<-'ID'

#Creating a test case folder - will run five imputations under each condition on 5 different
files:
#ALL test files will be with adult files.
data.folder<-paste0(wd, '/Good_3min')

file.names<-list.files(data.folder)

#Getting appropriate timing values for sections of "good" data
DF.segment<-merge(dat.time.orig, task.time[,1:2], by='ID')

#Will not be able to use case 051_T2 as there appears to be a task/timing issue there (Loses 1
child case)
DF.segment<-DF.segment[DF.segment$ID!="051_T2",]
DF.segment$Start.adj<-DF.segment$Start+DF.segment$Video
DF.segment$Stop.adj<-DF.segment$Stop+DF.segment$Video
DF.segment$time.min<-DF.segment$Start.adj+31
DF.segment$time.max<-DF.segment$Stop.adj-31

DF.segment<-merge(DF.segment, file_info[,c(1,14:17)], by='ID')

#####
#####
#Functions for identifying heart beats
#=====
#Function 1 - Finding Peakings Using Specified bandwidth:
findpeaks <- function (x, m = 3){
  shape <- diff(sign(diff(x, na.pad = FALSE)))
  pks <- sapply(which(shape < 0), FUN = function(i){
    z <- i - m + 1
    z <- ifelse(z > 0, z, 1)
    w <- i + m + 1
```

```

w <- ifelse(w < length(x), w, length(x))
if(all(x[c(z : i, (i + 2) : w)] <= x[i + 1])) return(i + 1) else return(numeric(0))
})
pks <- unlist(pks)
pks
}
#####
#Function 2 - Summing IBIs from Raw PPG file:
time.sum<-function(x){
  Z<-rep(NA, length(x))
  for(i in 1:length(x)){
    Z[i]<-ifelse(i==1, x[i], x[i]-x[i-1])
  }
  return(Z)
}
#####
#Function 2b - Summing Time from IBIs
IBI.sum<-function(x){
  Z<-rep(NA, length(x))
  for(i in 1:length(x)){
    Z[i]<-sum(x[1:i])
  }
  return(Z)
}
#####
#Function 3 - Iterative function for getting IBIs
iter.IBI<-function(x, ds=2000){
  #browser()
  require(psych)
  s<-round(seq(round(ds/50), round(ds/2), length.out = 200))
  Z<-data.frame(rep(NA, length(s)),
               rep(NA, length(s)),
               rep(NA, length(s)),
               rep(NA, length(s)),
               rep(NA, length(s)),
               rep(NA, length(s)))
  for(i in 1:length(s)){
    IBI<-findpeaks(x, s[i])
    time<-time.sum(IBI)/ds
    Z[i,1]<-s[i]
    Z[i,2]<-sd(time)
    Z[i,3]<-max(time)-min(time)
    Z[i,4]<-rmssd(time)
    Z[i,5]<-mean(acf(time, lag.max = length(time)/20, plot = F)$acf)
    Z[i,6]<-s[i]/ds
  }
  colnames(Z)<-c('BW', 'SD', 'Range', 'RMSSD', 'AC', 'BW(s)')
  Z<-Z[order(Z$RMSSD, decreasing = F),]
  IBI.fin<-findpeaks(x, m=Z[1,1])-1
  IBI.fin<-IBI.fin/ds
  IBI.done<-time.sum(IBI.fin)
  IBI.comp<-list(IBI.done, Z)
  names(IBI.comp)<-c('IBI.done', 'Z')
  return(IBI.comp)
}
#####
#Function 4 - Obtaining Time Values for IBI
sum.rev<-function(x){
  Z<-rep(NA, length(x))
  for(i in 1:length(x)){
    Z[i]<-ifelse(i==1, x[i], sum(x[1:(i-1)])+x[i])
  }
  return(Z)
}
}
#####
###+++++
#setting up models:
chains<-2
impute.min<-2

```

```

impute.max<-8
warmup<-c(2000)
n.iter<-c(2500)
Data.sample.Hz<-c(2000)
impute.sample.Hz<-c(125, 250)
GP.sample.Hz<-c(4, 8, 12)
impute.fac<-c(1,2,3)

#Going to need to bring in header information
#For now all test cases have 15 rows of header information:

#####
library(ggplot2)
library(signal)
library(psych)
library(bayesplot)
library(MCMCvis)
library(rstan)
library(rstanarm)
library(astsa)
library(benchmarkme)
options(mc.cores=parallel::detectCores())
rstan_options(auto_write = TRUE)
#Run the complete simulation from here
#Basing this on 5 simulations each to start:
n.sims<-1

#Basic simulation vectors
ID.vec<-vector()
impute.ID<-vector()
CPU.vec<-vector()
Cores.vec<-vector()
RAM.vec<-vector()
pop.vec<-vector()
start.vec<-vector()
stop.vec<-vector()
IBI.rmssd.15<-vector()
IBI.rmssd.30<-vector()
IBI.rmssd.45<-vector()
IBI.rmssd.60<-vector()
IBI.MR.rmssd.15<-vector()
IBI.MR.rmssd.30<-vector()
IBI.MR.rmssd.45<-vector()
IBI.MR.rmssd.60<-vector()
IBI.HD.rmssd.15<-vector()
IBI.HD.rmssd.30<-vector()
IBI.HD.rmssd.45<-vector()
IBI.HD.rmssd.60<-vector()
IBI.GP.rmssd.15<-vector()
IBI.GP.rmssd.30<-vector()
IBI.GP.rmssd.45<-vector()
IBI.GP.rmssd.60<-vector()
N.IBIs.vec<-vector()
runtime<-vector()
warmup.vec<-vector()
iter.vec<-vector()
Hz.vec<-vector()
impute.Hz.vec<-vector()
GP.Hz.vec<-vector()
impute.fac.vec<-vector()
impute.start<-vector()
impute.end<-vector()
impute.tot<-vector()

#Parameter Means
HR.mean.vec<-vector()

#Parameter sd's
HR.SD.vec<-vector()

```



```

#Parameter R-hats
HR.Rhat.vec<-vector()

#Parameter N_eff
HR.N_eff.vec<-vector()

#Getting ML Convergence test
a1.ML.conv.vec<-vector()
a2.ML.conv.vec<-vector()
a3.ML.conv.vec<-vector()
r1.ML.conv.vec<-vector()
r2.ML.conv.vec<-vector()
r3.ML.conv.vec<-vector()
r4.ML.conv.vec<-vector()
r5.ML.conv.vec<-vector()
Resp.ML.conv.vec<-vector()

#Creating folder structure to allow for quick saving of Files
out.folder<-paste0(wd, '/Output')
IBI.orig.folder<-paste0(out.folder, '/IBI_Files/Original')
IBI.MR.folder<-paste0(out.folder, '/IBI_Files/Mean_Replacement')
IBI.HD.folder<-paste0(out.folder, '/IBI_Files/Hot_Deck')
IBI.impute.folder<-paste0(out.folder, '/IBI_Files/Imputed')
Peak.summary.folder<-paste0(out.folder, '/Peak_Detection_Summaries')
graphics.folder<-paste0(out.folder, '/Graphics')
Bayes.graphics<-paste0(graphics.folder, '/Bayes_Graphics')
model.folder<-paste0(out.folder, '/Model_Summaries')
diagnostics.folder<-paste0(out.folder, '/Diagnostics')

#Main simulation program:
for(i in 1:n.sims){
  #browser()
  #Getting the specifics for each run:
  impute.window.temp<-runif(1, impute.min, impute.max)
  row.select.temp<-sample(x=1:nrow(DF.segment), size = 1)

  #Obtaining temporary dataset
  dat.temp<-DF.segment[row.select.temp,]
  header.temp<-ifelse(dat.temp$Population=='child', dat.temp$Child_Header, dat.temp$Parent_Header)
  col.select.temp<-ifelse(dat.temp$Population=='child', dat.temp$Child_Column,
dat.temp$Parent_Column)

  #Selecting imputation boundaries
  impute.LB.temp<-runif(1, dat.temp$time.min, dat.temp$time.max)
  impute.UB.temp<-impute.LB.temp+impute.window.temp

  #getting PPG data
  PPG<-read.table(paste0(data.folder, '/', dat.temp$ID, '.txt'),
                 skip = header.temp,
                 header = F,
                 sep = '\t'
                )

  PPG<-PPG[,col.select.temp]

  PPG<-data.frame(PPG,
                 Time = seq(from = 0, by = .0005, length.out = length(PPG))
                )

  #Selecting target window for the case
  PPG.temp<-PPG[PPG$Time>=dat.temp$Start.adj & PPG$Time<=dat.temp$Stop.adj,]

  #Cleaning signal - de-spiking and smoothing heart rate
  PPG.temp$PPG<-as.numeric(smooth(PPG.temp$PPG))
  PPG.temp$PPG<-smooth.spline(PPG.temp$PPG, nknots = 10000)$y
  PPG.temp$PPG<-PPG.temp$PPG-predict(lm(PPG~Time, data = PPG.temp))

  #Obtaining original IBI values for section:

```

```

IBI.orig.temp<-iter.IBI(PPG.temp$PPG, ds=2000)

write.table(round(IBI.orig.temp$IBI.done, digits = 4),
            paste0(IBC.orig.folder,
                  '/',
                  paste(dat.temp$ID,
                        dat.temp$Population,
                        dat.temp$Start,
                        dat.temp$Stop,
                        round(impute.LB.temp, digits = 2),
                        sep = '_'),
                  '.txt'),
            row.names = F
)

write.table(round(head(IBC.orig.temp$Z, n=20), digits = 3),
            paste0(IBC.summary.folder,
                  '/',
                  paste(dat.temp$ID,
                        dat.temp$Population,
                        'original',
                        dat.temp$Start,
                        dat.temp$Stop,
                        round(impute.LB.temp, digits = 2),
                        sep = '_'),
                  '.txt'
            ),
            sep = '\t',
            row.names = F
)

#Mean replacement strategy - Will replace the number of IBIs in the affected range with mean values
#Step 1 - Restoring accurate timing file

IBC.orig.time<-sum.rev(IBC.orig.temp$IBI.done)+dat.temp$Start.adj

#Running very simple mean imputation (not accounting for time-series nature of the data)
IBC.mean.replace<-IBC.orig.temp$IBI.done
IBC.mean.replace[IBC.orig.time>impute.LB.temp &
                 IBC.orig.time<impute.UB.temp]<-rep(mean(IBC.orig.temp$IBI.done))

#removing first and last IBI value (to make more equivalent comparison with imputed data set)
write.table(round(IBC.mean.replace[c(-1, -length(IBC.mean.replace))], digits = 4),
            paste0(IBC.MR.folder,
                  '/',
                  paste(dat.temp$ID,
                        dat.temp$Population,
                        'Mean_replace',
                        dat.temp$Start,
                        dat.temp$Stop,
                        round(impute.LB.temp, digits = 2),
                        sep = '_'),
                  '.txt'),
            row.names = F
)

#Running hot deck imputation (does not account for time-series nature of the data)
IBC.sample.vals<-IBC.orig.temp$IBI.done[IBC.orig.time<impute.LB.temp |
IBC.orig.time>impute.UB.temp]
IBC.hotdeck.replace<-IBC.orig.temp$IBI.done
N.IBIs<-length(IBC.hotdeck.replace[IBC.orig.time>impute.LB.temp & IBC.orig.time<impute.UB.temp])
IBC.hotdeck.replace[IBC.orig.time>impute.LB.temp & IBC.orig.time<impute.UB.temp]<-
sample(x=IBC.sample.vals,
                                             size =
N.IBIs,
                                             replace
= T)
write.table(round(IBC.hotdeck.replace[c(-1, -length(IBC.hotdeck.replace))], digits = 4),

```

```

paste0(IBM.HD.folder,
      '/',
      paste(dat.temp$ID,
            dat.temp$Population,
            'HotDeck_replace',
            dat.temp$Start,
            dat.temp$Stop,
            round(impute.LB.temp, digits = 2),
            sep = '_'),
      '.txt'),
      row.names = F
)

for(j in 1:length(chains)){
  chain.temp<-chains[j]
  for(k in 1:length(warmup)){
    warmup.temp<-warmup[k]
    n.iter.temp<-n.iter[k]
    for(l in 1:length(Data.sample.Hz)){
      Hz.temp<-Data.sample.Hz[l]
      for(m in 1:length(impute.sample.Hz)){
        Hz.impute.temp<-impute.sample.Hz[m]
        for(n in 1:length(GP.sample.Hz)){
          Hz.GP.temp<-GP.sample.Hz[n]
          for(o in 1:length(impute.fac)){
            impute.fac.temp<-impute.fac[o]

            #Downsampling (if appropriate)
            select.int<-2000/Hz.temp
            PPG.impute<-PPG.temp[seq(1, nrow(PPG.temp), select.int),]
            IBI.impute<-iter.IBI(PPG.impute$PPG, ds=Hz.temp)

            #Getting some priors for the model
            mu_HP<-mean(1/IBI.impute$IBI.done[c(-1,-length(IBM.impute$IBI.done))], na.rm = T)
            sigma_HP<-sd(1/IBI.impute$IBI.done[c(-1,-length(IBM.impute$IBI.done))], na.rm = T)

            #Estimating Respiration parameter & respiration SD if population age is child
            if(dat.temp$Population == 'child'){
              spec<-mvspec(PPG.temp,
                spans = c(7,7),
                taper=.1,
                demean = T,
                log='no',
                plot = F)
              min.R<-20/60/Hz.temp
              max.R<-30/60/Hz.temp

              spec.trunc<-data.frame(freq=spec$freq[spec$freq>=min.R&spec$freq<=max.R],
                spec=spec$spec[spec$freq>=min.R&spec$freq<=max.R])
              spec.trunc$prob<-spec.trunc$spec/sum(spec.trunc$spec)
              tmp.dist<-sample(spec.trunc$freq, size = 10000, replace = T, prob =
spec.trunc$prob)*Hz.temp
              mu_R<-mean(tmp.dist)
              sigma_R<-sd(tmp.dist)
            }

            #Estimating Respiration parameter & respiration SD if population age is adult
            if(dat.temp$Population == 'adult'){
              spec<-mvspec(PPG.temp,
                spans = c(7,7),
                taper=.1,
                demean = T,
                log='no',
                plot = F)
              min.R<-12/60/Hz.temp
              max.R<-20/60/Hz.temp

              spec.trunc<-data.frame(freq=spec$freq[spec$freq>=min.R&spec$freq<=max.R],
                spec=spec$spec[spec$freq>=min.R&spec$freq<=max.R])

```

```

spec.trunc$prob<-spec.trunc$spec/sum(spec.trunc$spec)
tmp.dist<-sample(spec.trunc$freq, size = 10000, replace = T, prob =
spec.trunc$prob)*Hz.temp
mu_R<-mean(tmp.dist)
sigma_R<-sd(tmp.dist)
}

init.list<-list()
for(x in 1:chain.temp){
  init.list[[x]]<-list(mu_HP = mu_HP,
                      sigma_HP = sigma_HP,
                      mu_R = mu_R,
                      sigma_R = sigma_R)
}

if(Hz.temp>=Hz.impute.temp){
sampling rate
  #First selecting a series of time values to impute back in at the appropriate
  select.int2<-Hz.temp/Hz.impute.temp
  PPG.impute2<-PPG.impute[seq(1, nrow(PPG.impute), select.int2),]
  PPG.impute.pred<-PPG.impute2[PPG.impute2$Time>impute.LB.temp &
                               PPG.impute2$Time<impute.UB.temp,]

  #Getting what will be Prediction values:
  Xp<-PPG.impute.pred$Time
  N2<-length(PPG.impute.pred$Time)

  #Acquiring "good data"
  min.TIME2<-min(Xp)
  max.TIME2<-max(Xp)
  time.span<-max.TIME2-min.TIME2
  Y.vals<-rbind(PPG.impute2[PPG.impute2$Time>min.TIME2-impute.fac.temp*time.span &
                          PPG.impute2$Time<min.TIME2,],
                PPG.impute2[PPG.impute2$Time>max.TIME2 &
                          PPG.impute2$Time<max.TIME2+impute.fac.temp*time.span,])
  Y.vals<-na.omit(Y.vals)
  tot.Y.vals<-length(Y.vals[,1])
  sel.Y.vals<-round(seq(1, tot.Y.vals, length.out =
round(tot.Y.vals/Hz.impute.temp*Hz.GP.temp)))
  sel.Y.vals<-unique(sel.Y.vals)
  Y<-Y.vals$PPG[sel.Y.vals]
  X<-Y.vals$Time[sel.Y.vals]
  N1<-length(X)

  #Running Model - there is no need for respiration priors
  dat<-list(N1=N1,
            N2=N2,
            Xp=Xp,
            X=X,
            Y=Y,
            mu_HP=mu_HP,
            sigma_HP=sigma_HP,
            mu_R = mu_R,
            sigma_R = sigma_R
            )

  #Obtaining estimates for hyper-parameters using ML approach
  opt_model<-stan_model(file=paste0(wd, '/Stan_code/GP_2_opt.stan'))

  #need to find some way to assess convergence...
  print('Obtaining first ML estimate for hyperparameters')
  opt_fit1<-NULL
  attempt<-1
  while(is.null(opt_fit1) & attempt <=15){
    print(paste('Convergence attempt', attempt, 'out of 15'))
    attempt<- attempt + 1
    try(
      opt_fit1<-optimizing(opt_model,

```

```

                                data=dat,
                                init=list(mu_HP = mu_HP, mu_R = mu_R),
                                seed=sample(1:5000, size = 1), iter = 10000)
    )
}

print('Obtaining second ML estimate for hyperparameters')
opt_fit2<-NULL
attempt<-1
while(is.null(opt_fit2) & attempt <=15){
  print(paste('Convergence attempt', attempt, 'out of 15'))
  attempt<- attempt + 1
  try(
    opt_fit2<-optimizing(opt_model,
                        data=dat,
                        init=list(mu_HP = mu_HP, mu_R = mu_R),
                        seed=sample(1:5000, size = 1), iter = 10000)
  )
}

if(!is.null(opt_fit1)){
  alpha1.1 <- opt_fit1$par['a1']
  alpha2.1 <- opt_fit1$par['a2']
  alpha3.1 <- opt_fit1$par['a3']
  rho1.1 <- opt_fit1$par['r1']
  rho2.1 <- opt_fit1$par['r2']
  rho3.1 <- opt_fit1$par['r3']
  rho4.1 <- opt_fit1$par['r4']
  rho5.1 <- opt_fit1$par['r5']
  Resp.1 <- opt_fit1$par['R']
}

if(!is.null(opt_fit2)){
  alpha1.2 <- opt_fit2$par['a1']
  alpha2.2 <- opt_fit2$par['a2']
  alpha3.2 <- opt_fit2$par['a3']
  rho1.2 <- opt_fit2$par['r1']
  rho2.2 <- opt_fit2$par['r2']
  rho3.2 <- opt_fit2$par['r3']
  rho4.2 <- opt_fit2$par['r4']
  rho5.2 <- opt_fit2$par['r5']
  Resp.2 <- opt_fit2$par['R']
}

if(!is.null(opt_fit1) & !is.null(opt_fit2)){
  #Simple Alpha convergence check
  a1.ML.conv.vec<-c(a1.ML.conv.vec, abs(alpha1.1-alpha1.2))
  a2.ML.conv.vec<-c(a2.ML.conv.vec, abs(alpha2.1-alpha2.2))
  a3.ML.conv.vec<-c(a3.ML.conv.vec, abs(alpha3.1-alpha3.2))

  #Simple rho convergence check
  r1.ML.conv.vec<-c(r1.ML.conv.vec, abs(rho1.1-rho1.2))
  r2.ML.conv.vec<-c(r2.ML.conv.vec, abs(rho2.1-rho2.2))
  r3.ML.conv.vec<-c(r3.ML.conv.vec, abs(rho3.1-rho3.2))
  r4.ML.conv.vec<-c(r4.ML.conv.vec, abs(rho4.1-rho4.2))
  r5.ML.conv.vec<-c(r5.ML.conv.vec, abs(rho5.1-rho5.2))

  #Respiration Convergence
  Resp.ML.conv.vec<-c(Resp.ML.conv.vec, abs(Resp.1-Resp.2))
}

else{
  #Simple Alpha convergence check
  a1.ML.conv.vec<-c(a1.ML.conv.vec, NA)
  a2.ML.conv.vec<-c(a2.ML.conv.vec, NA)
  a3.ML.conv.vec<-c(a3.ML.conv.vec, NA)

  #Simple rho convergence check
  r1.ML.conv.vec<-c(r1.ML.conv.vec, NA)

```

```

r2.ML.conv.vec<-c(r2.ML.conv.vec, NA)
r3.ML.conv.vec<-c(r3.ML.conv.vec, NA)
r4.ML.conv.vec<-c(r4.ML.conv.vec, NA)
r5.ML.conv.vec<-c(r5.ML.conv.vec, NA)

#Respiration convergence
Resp.ML.conv.vec<-c(Resp.ML.conv.vec, NA)
}

if(is.null(opt_fit1) & !is.null(opt_fit2)){
  alpha1.1<-alpha1.2
  alpha2.1<-alpha2.2
  alpha3.1<-alpha3.2
  rho1.1<-rho1.2
  rho2.1<-rho2.2
  rho3.1<-rho3.2
  rho4.1<-rho4.2
  rho5.1<-rho5.2
  Resp.1<-Resp.2
}

pars.to.monitor<-c('HR', 'Ypred')
start.time.temp<-Sys.time()

start.time.temp<-Sys.time()

dat.opt<-list(N1=N1,
             N2=N2,
             Xp=Xp,
             X=X,
             Y=Y,
             mu_HR=mu_HP,
             sigma_HR=sigma_HP,
             a1 = alpha1.1,
             a2 = alpha2.1,
             a3 = alpha3.1,
             r1 = rho1.1,
             r2 = rho2.1,
             r3 = rho3.1,
             r4 = rho4.1,
             r5 = rho5.1,
             R = Resp.1
)

fit.stan<-stan(file=paste0(wd, '/Stan_code/GP_2_b.stan'),
              data = dat.opt,
              init = init.list,
              warmup = warmup.temp,
              iter = n.iter.temp,
              refresh=100,
              chains = chain.temp,
              pars = pars.to.monitor,
              control = list(adapt_delta = .95,
                             max_treedepth = 15)
)

run_time<-round(Sys.time()-start.time.temp, digits = 3)
units(run_time)<-'mins'
run_time<-as.numeric(run_time)

y_pred<-rstan::extract(fit.stan, 'Ypred')
PPG.new<-colMeans(y_pred$Ypred)
PPG.DF.new<-data.frame(PPG = PPG.new,
                      Time = Xp)

GP.summary<-as.data.frame(summary(fit.stan, pars = pars.to.monitor[-
length(pars.to.monitor)],
                                probs = c(.1, .9))$summary)

```

```

write.table(round(GP.summary, digits = 3),
            paste0(model.folder,
                  '/',
                  paste(dat.temp$ID,
                        dat.temp$Population,
                        dat.temp$Segment,
                        'GP_impute',
                        round(impute.LB.temp, digits = 2),
                        round(impute.UB.temp, digits = 2),
                        Hz.temp,
                        Hz.impute.temp,
                        Hz.GP.temp,
                        impute.fac.temp,
                        chain.temp,
                        warmup.temp,
                        n.iter.temp,
                        round(impute.LB.temp, digits = 2),
                        sep = '_'),
                  '.txt'),
            row.names = T,
            sep = '\t'
)

g1<-ggplot()+
  geom_line(data=PPG.DF.new, aes(x=Time, y = PPG), color = 'red', lty='dashed')+
  geom_line(data=PPG.impute.pred, aes(x=Time, y=PPG), color = 'black')+
  ggtitle(paste(dat.temp$ID, dat.temp$Population, dat.temp$Segment,
                'File Hz =', Hz.temp, 'GP Dataset Hz =', Hz.impute.temp,
                'Model Sampling Rate =', Hz.GP.temp, 'Chains =', chain.temp,
                'Warmup =', warmup.temp, 'Iter =', n.iter.temp))

ggsave(filename = paste0(graphics.folder, paste('/',
                                                dat.temp$ID,
                                                dat.temp$Population,
                                                dat.temp$Segment,
                                                Hz.temp,
                                                Hz.impute.temp,
                                                Hz.GP.temp,
                                                impute.fac.temp,
                                                chain.temp,
                                                warmup.temp,
                                                n.iter.temp,
                                                round(impute.LB.temp, digits = 2),
                                                sep = '_'), '.png'),

        plot = g1,
        width = 11,
        height = 8,
        units = 'in',
        dpi = 300,
        device = 'png')

g2<-traceplot(fit.stan, pars='HR', inc_warmup=T)+
  ggtitle(paste('HR param', dat.temp$ID, dat.temp$Population, dat.temp$Segment,
                'File Hz =', Hz.temp, 'GP Dataset Hz =', Hz.impute.temp,
                'Model Sampling Rate =', Hz.GP.temp, '\n', 'Chains =', chain.temp,
                'Warmup =', warmup.temp, 'Iter =', n.iter.temp))

ggsave(filename = paste0(Bayes.graphics, paste('/',
                                                dat.temp$ID,
                                                dat.temp$Population,
                                                dat.temp$Segment,
                                                Hz.temp,
                                                Hz.impute.temp,
                                                Hz.GP.temp,
                                                impute.fac.temp,
                                                chain.temp,
                                                warmup.temp,
                                                n.iter.temp,

```

```

round(impute.LB.temp, digits = 2),
sep = '_'), '.png'),

plot = g2,
width = 11,
height = 8,
units = 'in',
dpi = 300,
device = 'png')

PPG.impute.fin<-PPG.impute2
PPG.impute.fin$PPG[PPG.impute.fin$Time>impute.LB.temp &
PPG.impute.fin$Time<impute.UB.temp]<-PPG.new

IBI.GP.impute.temp<-iter.IBI(PPG.impute.fin$PPG, ds=Hz.impute.temp)$IBI.done
IBI.GP.impute.time<-sum.rev(IBI.GP.impute.temp)+dat.temp$Start.adj
write.table(round(IBI.GP.impute.temp[c(-1, -length(IBI.GP.impute.temp))], digits =
4),

paste0(IBI.impute.folder,
'/',
paste(dat.temp$ID,
dat.temp$Population,
dat.temp$Segment,
'GP_impute',
round(impute.LB.temp, digits = 2),
round(impute.UB.temp, digits = 2),
Hz.temp,
Hz.impute.temp,
Hz.GP.temp,
impute.fac.temp,
chain.temp,
warmup.temp,
n.iter.temp,
round(impute.LB.temp, digits = 2),
sep = '_'),
'.txt'),
row.names = F
)

HR.est<-rstan::extract(fit.stan, 'HR')

#Basic simulation vectors
ID.vec<-c(ID.vec, dat.temp$ID)
impute.ID<-c(impute.ID, paste('Segment', i, sep = '_'))
CPU.vec<-c(CPU.vec, benchmarkme::get_cpu()$model_name)
Cores.vec<-c(Cores.vec, parallel::detectCores())
RAM.vec<-c(RAM.vec, paste(round(benchmarkme::get_ram()/1073741824), 'GB'))
pop.vec<-c(pop.vec, dat.temp$Population)
N.IBIs.vec<-c(N.IBIs.vec, N.IBIs)
start.vec<-c(start.vec, dat.temp$Start)
stop.vec<-c(stop.vec, dat.temp$Stop)
IBI.GP.rmssd.15<-c(IBI.GP.rmssd.15,
rmssd(IBI.GP.impute.temp[IBI.GP.impute.time>impute.LB.temp-7.5
&
IBI.GP.impute.time<impute.UB.temp+7.5]))
IBI.GP.rmssd.30<-c(IBI.GP.rmssd.30,
rmssd(IBI.GP.impute.temp[IBI.GP.impute.time>impute.LB.temp-15 &
IBI.GP.impute.time<impute.UB.temp+15]))
IBI.GP.rmssd.45<-c(IBI.GP.rmssd.45,
rmssd(IBI.GP.impute.temp[IBI.GP.impute.time>impute.LB.temp-22.5
&
IBI.GP.impute.time<impute.UB.temp+22.5]))
IBI.GP.rmssd.60<-c(IBI.GP.rmssd.60,
rmssd(IBI.GP.impute.temp[IBI.GP.impute.time>impute.LB.temp-30 &
IBI.GP.impute.time<impute.UB.temp+30]))
runtime<-c(runtime, run_time)

```



```

warmup.vec<-c(warmup.vec, warmup.temp)
iter.vec<-c(iter.vec, n.iter.temp)
Hz.vec<-c(Hz.vec, Hz.temp)
impute.Hz.vec<-c(impute.Hz.vec, Hz.impute.temp)
GP.Hz.vec<-c(GP.Hz.vec, Hz.GP.temp)
impute.fac.vec<-c(impute.fac.vec, impute.fac.temp)
impute.start<-c(impute.start, min.TIME2)
impute.end<-c(impute.end, max.TIME2)
impute.tot<-c(impute.tot, time.span)

#Parameter Means
HR.mean.vec<-c(HR.mean.vec, mean(HR.est$HR))

#Parameter sd's
HR.SD.vec<-c(HR.SD.vec, sd(HR.est$HR))

#Parameter R-hats
HR.Rhat.vec<-c(HR.Rhat.vec, GP.summary$Rhat[rownames(GP.summary)=='HR'])

#Parameter N_eff
HR.N_eff.vec<-c(HR.N_eff.vec, GP.summary$n_eff[rownames(GP.summary)=='HR'])

#Getting a vector of original value rmssd's
IBI.rmssd.15<-c(IBI.rmssd.15,
                rmssd(IBI.orig.temp$IBI.done[IBI.orig.time>impute.LB.temp-7.5 &
                    IBI.orig.time<impute.UB.temp+7.5]))
IBI.rmssd.30<-c(IBI.rmssd.30,
                rmssd(IBI.orig.temp$IBI.done[IBI.orig.time>impute.LB.temp-15 &
                    IBI.orig.time<impute.UB.temp+15]))
IBI.rmssd.45<-c(IBI.rmssd.45,
                rmssd(IBI.orig.temp$IBI.done[IBI.orig.time>impute.LB.temp-22.5 &
                    IBI.orig.time<impute.UB.temp+22.5]))
IBI.rmssd.60<-c(IBI.rmssd.60,
                rmssd(IBI.orig.temp$IBI.done[IBI.orig.time>impute.LB.temp-30 &
                    IBI.orig.time<impute.UB.temp+30]))

#Getting a vector of mean replacement rmssd's
IBI.MR.rmssd.15<-c(IBI.MR.rmssd.15,
                  rmssd(IBI.mean.replace[IBI.orig.time>impute.LB.temp-7.5 &
                      IBI.orig.time<impute.UB.temp+7.5]))
IBI.MR.rmssd.30<-c(IBI.MR.rmssd.30,
                  rmssd(IBI.mean.replace[IBI.orig.time>impute.LB.temp-15 &
                      IBI.orig.time<impute.UB.temp+15]))
IBI.MR.rmssd.45<-c(IBI.MR.rmssd.45,
                  rmssd(IBI.mean.replace[IBI.orig.time>impute.LB.temp-22.5 &
                      IBI.orig.time<impute.UB.temp+22.5]))
IBI.MR.rmssd.60<-c(IBI.MR.rmssd.60,
                  rmssd(IBI.mean.replace[IBI.orig.time>impute.LB.temp-30 &
                      IBI.orig.time<impute.UB.temp+30]))

#Getting a vector of hotdeck replacement rmssd's
IBI.HD.rmssd.15<-c(IBI.HD.rmssd.15,
                  rmssd(IBI.hotdeck.replace[IBI.orig.time>impute.LB.temp-7.5 &
                      IBI.orig.time<impute.UB.temp+7.5]))
IBI.HD.rmssd.30<-c(IBI.HD.rmssd.30,
                  rmssd(IBI.hotdeck.replace[IBI.orig.time>impute.LB.temp-15 &
                      IBI.orig.time<impute.UB.temp+15]))
IBI.HD.rmssd.45<-c(IBI.HD.rmssd.45,
                  rmssd(IBI.hotdeck.replace[IBI.orig.time>impute.LB.temp-22.5 &
                      IBI.orig.time<impute.UB.temp+22.5]))
IBI.HD.rmssd.60<-c(IBI.HD.rmssd.60,
                  rmssd(IBI.hotdeck.replace[IBI.orig.time>impute.LB.temp-30 &
                      IBI.orig.time<impute.UB.temp+30]))

}
}
}
}

```

```

    }
  }
}

save(list = ls(),
      file = paste0(out.folder, '/', paste('GP_2', Sys.time(), n.sims, sep = '_'), '.RData'))

```

Appendix D.3 Re-sampling Code for Gaussian Process Model Imputation Model 3

```

#Ensuring random starts for two different threads on same computer...
set.seed(as.numeric(Sys.time()))

#Original Processing:

dat.time.orig<-read.csv('/home/mbarsted/Dropbox/Dissertation/Chapter
4/Sim_data_3_b/Raw_coded.csv',
                       stringsAsFactors = F)

#Cleaning up extra rows
dat.time.orig<-dat.time.orig[-91:-nrow(dat.time.orig),]

#Taking only cases with at least 180 continuous seconds worth of data:
dat.time.orig<-dat.time.orig[dat.time.orig$Window_Length>=180,]

table(dat.time.orig$Population) #not bad - about 50/50
#40 sections of adult data
#39 sections of child data

#Converting "Adult" to "adult" and "Child" to "Child"
dat.time.orig$Population<-ifelse(dat.time.orig$Population=='Adult', 'adult', 'child')

wd<-'/home/mbarsted/Dropbox/Dissertation/Chapter 4/Sim_data_3_b'

#Bringing in file information worksheet
file_info<-read.csv(paste0(wd, '/Physio_Tracking - Sheet1.csv'), stringsAsFactors = F)

#Converting ID to common structure:
file_info$ID<-substr(file_info$Video_1, start=1, stop=6)

#Task timing file
task.time<-read.table(paste0(wd, '/Timing_File_forVizEdit.txt'),
                     header = T, sep = '\t')
colnames(task.time)[1]<-'ID'

#Creating a test case folder - will run five imputations under each condition on 5 different
files:
#All test files will be with adult files.
data.folder<-paste0(wd, '/Good_3min')

file.names<-list.files(data.folder)

#Getting appropriate timing values for sections of "good" data
DF.segment<-merge(dat.time.orig, task.time[,1:2], by='ID')

#Will not be able to use case 051_T2 as there appears to be a task/timing issue there (Loses 1
child case)
DF.segment<-DF.segment[DF.segment$ID!="051_T2",]
DF.segment$Start.adj<-DF.segment$Start+DF.segment$Video
DF.segment$Stop.adj<-DF.segment$Stop+DF.segment$Video
DF.segment$time.min<-DF.segment$Start.adj+31
DF.segment$time.max<-DF.segment$Stop.adj-31

DF.segment<-merge(DF.segment, file_info[,c(1,14:17)], by='ID')

#####

```

```

#####
#Functions for identifying heart beats
#=====
#Function 1 - Finding Peakings Using Specified bandwidth:
findpeaks <- function (x, m = 3){
  shape <- diff(sign(diff(x, na.pad = FALSE)))
  pks <- sapply(which(shape < 0), FUN = function(i){
    z <- i - m + 1
    z <- ifelse(z > 0, z, 1)
    w <- i + m + 1
    w <- ifelse(w < length(x), w, length(x))
    if(all(x[c(z : i, (i + 2) : w)] <= x[i + 1])) return(i + 1) else return(numeric(0))
  })
  pks <- unlist(pks)
  pks
}
#=====
#Function 2 - Summing IBIs from Raw PPG file:
time.sum<-function(x){
  Z<-rep(NA, length(x))
  for(i in 1:length(x)){
    Z[i]<-ifelse(i==1, x[i], x[i]-x[i-1])
  }
  return(Z)
}
#=====
#Function 2b - Summing Time from IBIs
IBI.sum<-function(x){
  Z<-rep(NA, length(x))
  for(i in 1:length(x)){
    Z[i]<-sum(x[1:i])
  }
  return(Z)
}
#=====
#Function 3 - Iterative function for getting IBIs
iter.IBI<-function(x, ds=2000){
  #browser()
  require(psych)
  s<-round(seq(round(ds/50), round(ds/2), length.out = 200))
  Z<-data.frame(rep(NA, length(s)),
                rep(NA, length(s)),
                rep(NA, length(s)),
                rep(NA, length(s)),
                rep(NA, length(s)),
                rep(NA, length(s)))
  for(i in 1:length(s)){
    IBI<-findpeaks(x, s[i])
    time<-time.sum(IBI)/ds
    Z[i,1]<-s[i]
    Z[i,2]<-sd(time)
    Z[i,3]<-max(time)-min(time)
    Z[i,4]<-rmssd(time)
    Z[i,5]<-mean(acf(time, lag.max = length(time)/20, plot = F)$acf)
    Z[i,6]<-s[i]/ds
  }
  colnames(Z)<-c('BW', 'SD', 'Range', 'RMSSD', 'AC', 'BW(s)')
  Z<-Z[order(Z$RMSSD, decreasing = F),]
  IBI.fin<-findpeaks(x, m=Z[1,1])-1
  IBI.fin<-IBI.fin/ds
  IBI.done<-time.sum(IBI.fin)
  IBI.comp<-list(IBI.done, Z)
  names(IBI.comp)<-c('IBI.done', 'Z')
  return(IBI.comp)
}
#=====
#Function 4 - Obtaining Time Values for IBI
sum.rev<-function(x){
  Z<-rep(NA, length(x))

```

```

for(i in 1:length(x)){
  Z[i]<-ifelse(i==1, x[i], sum(x[1:(i-1)])+x[i])
}
return(Z)
}

###+++++
#setting up models:
chains<-2
impute.min<-2
impute.max<-8
warmup<-c(2000)
n.iter<-c(2500)
Data.sample.Hz<-c(2000)
impute.sample.Hz<-c(125, 250)
GP.sample.Hz<-c(4, 8, 12)
impute.fac<-c(1,2,3)

#Going to need to bring in header information
#For now all test cases have 15 rows of header information:

#####
library(ggplot2)
library(signal)
library(psych)
library(bayesplot)
library(MCMCvis)
library(rstan)
library(rstanarm)
library(astsa)
library(benchmarkme)
options(mc.cores=parallel::detectCores())
rstan_options(auto_write = TRUE)
#Run the complete simulation from here
#Basing this on 5 simulations each to start:
n.sims<-5

#Basic simulation vectors
ID.vec<-vector()
impute.ID<-vector()
CPU.vec<-vector()
Cores.vec<-vector()
RAM.vec<-vector()
pop.vec<-vector()
start.vec<-vector()
stop.vec<-vector()
IBI.rmssd.15<-vector()
IBI.rmssd.30<-vector()
IBI.rmssd.45<-vector()
IBI.rmssd.60<-vector()
IBI.MR.rmssd.15<-vector()
IBI.MR.rmssd.30<-vector()
IBI.MR.rmssd.45<-vector()
IBI.MR.rmssd.60<-vector()
IBI.HD.rmssd.15<-vector()
IBI.HD.rmssd.30<-vector()
IBI.HD.rmssd.45<-vector()
IBI.HD.rmssd.60<-vector()
IBI.GP.rmssd.15<-vector()
IBI.GP.rmssd.30<-vector()
IBI.GP.rmssd.45<-vector()
IBI.GP.rmssd.60<-vector()
N.IBIs.vec<-vector()
runtime<-vector()
warmup.vec<-vector()
iter.vec<-vector()
Hz.vec<-vector()
impute.Hz.vec<-vector()
GP.Hz.vec<-vector()

```

```

impute.fac.vec<-vector()
impute.start<-vector()
impute.end<-vector()
impute.tot<-vector()

#Parameter Means
HR.mean.vec<-vector()

#Parameter sd's
HR.SD.vec<-vector()

#Parameter R-hats
HR.Rhat.vec<-vector()

#Parameter N_eff
HR.N_eff.vec<-vector()

#Getting ML Convergence test
a1.ML.conv.vec<-vector()
a2.ML.conv.vec<-vector()
a3.ML.conv.vec<-vector()
a4.ML.conv.vec<-vector()

r1.ML.conv.vec<-vector()
r2.ML.conv.vec<-vector()
r3.ML.conv.vec<-vector()
r4.ML.conv.vec<-vector()
r5.ML.conv.vec<-vector()
r6.ML.conv.vec<-vector()
r7.ML.conv.vec<-vector()

Resp.ML.conv.vec<-vector()

#Creating folder structure to allow for quick saving of Files
out.folder<-paste0(wd, '/Output')
IBI.orig.folder<-paste0(out.folder, '/IBI_Files/Original')
IBI.MR.folder<-paste0(out.folder, '/IBI_Files/Mean_Replacement')
IBI.HD.folder<-paste0(out.folder, '/IBI_Files/Hot_Deck')
IBI.impute.folder<-paste0(out.folder, '/IBI_Files/Imputed')
Peak.summary.folder<-paste0(out.folder, '/Peak_Detection_Summaries')
graphics.folder<-paste0(out.folder, '/Graphics')
Bayes.graphics<-paste0(graphics.folder, '/Bayes_Graphics')
model.folder<-paste0(out.folder, '/Model_Summaries')
diagnostics.folder<-paste0(out.folder, '/Diagnostics')

#Main simulation program:
for(i in 1:n.sims){
  #browser()
  #Getting the specifics for each run:
  impute.window.temp<-runif(1, impute.min, impute.max)
  row.select.temp<-sample(x=1:nrow(DF.segment), size = 1)

  #Obtaining temporary dataset
  dat.temp<-DF.segment[row.select.temp,]
  header.temp<-ifelse(dat.temp$Population=='child', dat.temp$Child_Header, dat.temp$Parent_Header)
  col.select.temp<-ifelse(dat.temp$Population=='child', dat.temp$Child_Column,
  dat.temp$Parent_Column)

  #Selecting imputation boundaries
  impute.LB.temp<-runif(1, dat.temp$time.min, dat.temp$time.max)
  impute.UB.temp<-impute.LB.temp+impute.window.temp

  #getting PPG data
  PPG<-read.table(paste0(data.folder, '/', dat.temp$ID, '.txt'),
                 skip = header.temp,
                 header = F,
                 sep = '\t'
  )
}

```

```

PPG<-PPG[,col.select.temp]

PPG<-data.frame(PPG,
                Time = seq(from = 0, by = .0005, length.out = length(PPG))
                )

#Selecting target window for the case
PPG.temp<-PPG[PPG$Time>=dat.temp$Start.adj & PPG$Time<=dat.temp$Stop.adj,]

#Cleaning signal - de-spiking and smoothing heart rate
PPG.temp$PPG<-as.numeric(smooth(PPG.temp$PPG))
PPG.temp$PPG<-smooth.spline(PPG.temp$PPG, nknots = 10000)$y
PPG.temp$PPG<-PPG.temp$PPG-predict(lm(PPG~Time, data = PPG.temp))

#Obtaining original IBI values for section:
IBI.orig.temp<-iter.IBI(PPG.temp$PPG, ds=2000)

write.table(round(IBI.orig.temp$IBI.done, digits = 4),
            paste0(IBI.orig.folder,
                  '/',
                  paste(dat.temp$ID,
                        dat.temp$Population,
                        dat.temp$Start,
                        dat.temp$Stop,
                        round(impute.LB.temp, digits = 2),
                        sep = '_'),
                  '.txt'),
            row.names = F
            )

write.table(round(head(IBI.orig.temp$Z, n=20), digits = 3),
            paste0(Peak.summary.folder,
                  '/',
                  paste(dat.temp$ID,
                        dat.temp$Population,
                        'original',
                        dat.temp$Start,
                        dat.temp$Stop,
                        round(impute.LB.temp, digits = 2),
                        sep = '_'),
                  '.txt'
            ),
            sep = '\t',
            row.names = F
            )

#Mean replacement strategy - Will replace the number of IBIs in the affected range with mean values
#Step 1 - Restoring accurate timing file

IBI.orig.time<-sum.rev(IBI.orig.temp$IBI.done)+dat.temp$Start.adj

#Running very simple mean imputation (not accounting for time-series nature of the data)
IBI.mean.replace<-IBI.orig.temp$IBI.done
IBI.mean.replace[IBI.orig.time>impute.LB.temp &
                 IBI.orig.time<impute.UB.temp]<-rep(mean(IBI.orig.temp$IBI.done))

#removing first and last IBI value (to make more equivalent comparison with imputed data set)
write.table(round(IBI.mean.replace[c(-1, -length(IBI.mean.replace))], digits = 4),
            paste0(IBI.MR.folder,
                  '/',
                  paste(dat.temp$ID,
                        dat.temp$Population,
                        'Mean_replace',
                        dat.temp$Start,
                        dat.temp$Stop,
                        round(impute.LB.temp, digits = 2),
                        sep = '_'),
                  '.txt'),
            )

```

```

        row.names = F
    )

    #Running hot deck imputation (does not account for time-series nature of the data)
    IBI.sample.vals<-IBI.orig.temp$IBI.done[IBI.orig.time<impute.LB.temp |
IBI.orig.time>impute.UB.temp]
    IBI.hotdeck.replace<-IBI.orig.temp$IBI.done
    N.IBIs<-length(IBI.hotdeck.replace[IBI.orig.time>impute.LB.temp & IBI.orig.time<impute.UB.temp])
    IBI.hotdeck.replace[IBI.orig.time>impute.LB.temp & IBI.orig.time<impute.UB.temp]<-
sample(x=IBI.sample.vals,                                     size =
N.IBIs,                                                    replace
= T)
    write.table(round(IBI.hotdeck.replace[c(-1, -length(IBI.hotdeck.replace))], digits = 4),
paste0(IBI.HD.folder,
        '/',
        paste(dat.temp$ID,
              dat.temp$Population,
              'HotDeck_replace',
              dat.temp$Start,
              dat.temp$Stop,
              round(impute.LB.temp, digits = 2),
              sep = '_'),
        '.txt'),
        row.names = F
    )

    for(j in 1:length(chains)){
        chain.temp<-chains[j]
        for(k in 1:length(warmup)){
            warmup.temp<-warmup[k]
            n.iter.temp<-n.iter[k]
            for(l in 1:length(Data.sample.Hz)){
                Hz.temp<-Data.sample.Hz[l]
                for(m in 1:length(impute.sample.Hz)){
                    Hz.impute.temp<-impute.sample.Hz[m]
                    for(n in 1:length(GP.sample.Hz)){
                        Hz.GP.temp<-GP.sample.Hz[n]
                        for(o in 1:length(impute.fac)){
                            impute.fac.temp<-impute.fac[o]

                            #Downsampling (if appropriate)
                            select.int<-2000/Hz.temp
                            PPG.impute<-PPG.temp[seq(1, nrow(PPG.temp), select.int),]
                            IBI.impute<-iter.IBI(PPG.impute$PPG, ds=Hz.temp)

                            #Getting some priors for the model
                            mu_HP<-mean(1/IBI.impute$IBI.done[c(-1, -length(IBI.impute$IBI.done))], na.rm = T)
                            sigma_HP<-sd(1/IBI.impute$IBI.done[c(-1, -length(IBI.impute$IBI.done))], na.rm = T)

                            #Estimating Respiration parameter child
                            if(dat.temp$Population == 'child'){
                                spec<-mvspec(PPG.temp,
                                                spans = c(7,7),
                                                taper=.1,
                                                demean = T,
                                                log='no',
                                                plot = F)

                                min.R<-20/60/Hz.temp
                                max.R<-30/60/Hz.temp

                                spec.trunc<-data.frame(freq=spec$freq[spec$freq>=min.R&spec$freq<=max.R],
                                                         spec=spec$spec[spec$freq>=min.R&spec$freq<=max.R])
                                spec.trunc$prob<-spec.trunc$spec/sum(spec.trunc$spec)
                                tmp.dist<-sample(spec.trunc$freq, size = 10000, replace = T, prob =
spec.trunc$prob)*Hz.temp
                                mu_R<-mean(tmp.dist)
                                sigma_R<-sd(tmp.dist)

```

```

}

#Estimating Respiration parameter adult
if(dat.temp$Population == 'adult'){
  spec<-mvspec(PPG.temp,
              spans = c(7,7),
              taper=.1,
              demean = T,
              log='no',
              plot = F)
  min.R<-12/60/Hz.temp
  max.R<-20/60/Hz.temp

  spec.trunc<-data.frame(freq=spec$freq[spec$freq>=min.R&spec$freq<=max.R],
                        spec=spec$spec[spec$freq>=min.R&spec$freq<=max.R])
  spec.trunc$prob<-spec.trunc$spec/sum(spec.trunc$spec)
  tmp.dist<-sample(spec.trunc$freq, size = 10000, replace = T, prob =
spec.trunc$prob)*Hz.temp
  mu_R<-mean(tmp.dist)
  sigma_R<-sd(tmp.dist)
}

init.list<-list()
for(x in 1:chain.temp){
  init.list[[x]]<-list(mu_HP = mu_HP,
                      sigma_HP = sigma_HP,
                      mu_R = mu_R,
                      sigma_R = sigma_R)
}

if(Hz.temp>=Hz.impute.temp){
  #First selecting a series of time values to impute back in at the appropriate
sampling rate
  select.int2<-Hz.temp/Hz.impute.temp
  PPG.impute2<-PPG.impute[seq(1, nrow(PPG.impute), select.int2),]
  PPG.impute.pred<-PPG.impute2[PPG.impute2$Time>impute.LB.temp &
PPG.impute2$Time<impute.UB.temp,]

  #Getting what will be Prediction values:
  Xp<-PPG.impute.pred$Time
  N2<-length(PPG.impute.pred$Time)

  #Acquiring "good data"
  min.TIME2<-min(Xp)
  max.TIME2<-max(Xp)
  time.span<-max.TIME2-min.TIME2
  Y.vals<-rbind(PPG.impute2[PPG.impute2$Time>min.TIME2-impute.fac.temp*time.span &
PPG.impute2$Time<min.TIME2,],
               PPG.impute2[PPG.impute2$Time>max.TIME2 &
PPG.impute2$Time<max.TIME2+impute.fac.temp*time.span,])
  Y.vals<-na.omit(Y.vals)
  tot.Y.vals<-length(Y.vals[,1])
  sel.Y.vals<-round(seq(1, tot.Y.vals, length.out =
round(tot.Y.vals/Hz.impute.temp*Hz.GP.temp)))
  sel.Y.vals<-unique(sel.Y.vals)
  Y<-Y.vals$PPG[sel.Y.vals]
  X<-Y.vals$Time[sel.Y.vals]
  N1<-length(X)

  #Running Model - there is no need for respiration priors
  dat<-list(N1=N1,
           N2=N2,
           Xp=Xp,
           X=X,
           Y=Y,
           mu_HP=mu_HP,
           sigma_HP=sigma_HP,
           mu_R = mu_R,

```



```

        sigma_R = sigma_R
    )

#Obtaining estimates for hyper-parameters using ML approach
opt_model<-stan_model(file=paste0(wd, '/Stan_code/GP_3_opt.stan'))

#need to find some way to assess convergence...
print('Obtaining first ML estimate for hyperparameters')
opt_fit1<-NULL
attempt<-1
while(is.null(opt_fit1) & attempt <=15){
  print(paste('Convergence attempt', attempt, 'out of 15'))
  attempt<- attempt + 1
  try(
    opt_fit1<-optimizing(opt_model,
                        data=dat,
                        init=list(mu_HP = mu_HP, mu_R = mu_R),
                        seed=sample(1:5000, size = 1), iter = 10000)
  )
}

print('Obtaining second ML estimate for hyperparameters')
opt_fit2<-NULL
attempt<-1
while(is.null(opt_fit2) & attempt <=15){
  print(paste('Convergence attempt', attempt, 'out of 15'))
  attempt<- attempt + 1
  try(
    opt_fit2<-optimizing(opt_model,
                        data=dat,
                        init=list(mu_HP = mu_HP, mu_R = mu_R),
                        seed=sample(1:5000, size = 1), iter = 10000)
  )
}

if(!is.null(opt_fit1)){
  alpha1.1 <- opt_fit1$par['a1']
  alpha2.1 <- opt_fit1$par['a2']
  alpha3.1 <- opt_fit1$par['a3']
  alpha4.1 <- opt_fit1$par['a4']
  rho1.1 <- opt_fit1$par['r1']
  rho2.1 <- opt_fit1$par['r2']
  rho3.1 <- opt_fit1$par['r3']
  rho4.1 <- opt_fit1$par['r4']
  rho5.1 <- opt_fit1$par['r5']
  rho6.1 <- opt_fit1$par['r6']
  rho7.1 <- opt_fit1$par['r7']
  Resp.1 <- opt_fit1$par['R']
}

if(!is.null(opt_fit2)){
  alpha1.2 <- opt_fit2$par['a1']
  alpha2.2 <- opt_fit2$par['a2']
  alpha3.2 <- opt_fit2$par['a3']
  alpha4.2 <- opt_fit2$par['a4']
  rho1.2 <- opt_fit2$par['r1']
  rho2.2 <- opt_fit2$par['r2']
  rho3.2 <- opt_fit2$par['r3']
  rho4.2 <- opt_fit2$par['r4']
  rho5.2 <- opt_fit2$par['r5']
  rho6.2 <- opt_fit2$par['r6']
  rho7.2 <- opt_fit2$par['r7']
  Resp.2 <- opt_fit2$par['R']
}

if(!is.null(opt_fit1) & !is.null(opt_fit2)){
  #Simple Alpha convergence check
  a1.ML.conv.vec<-c(a1.ML.conv.vec, abs(alpha1.1-alpha1.2))
  a2.ML.conv.vec<-c(a2.ML.conv.vec, abs(alpha2.1-alpha2.2))
}

```

```

a3.ML.conv.vec<-c(a3.ML.conv.vec, abs(alpha3.1-alpha3.2))
a4.ML.conv.vec<-c(a4.ML.conv.vec, abs(alpha4.1-alpha4.2))

#Simple rho convergence check
r1.ML.conv.vec<-c(r1.ML.conv.vec, abs(rho1.1-rho1.2))
r2.ML.conv.vec<-c(r2.ML.conv.vec, abs(rho2.1-rho2.2))
r3.ML.conv.vec<-c(r3.ML.conv.vec, abs(rho3.1-rho3.2))
r4.ML.conv.vec<-c(r4.ML.conv.vec, abs(rho4.1-rho4.2))
r5.ML.conv.vec<-c(r5.ML.conv.vec, abs(rho5.1-rho5.2))
r6.ML.conv.vec<-c(r6.ML.conv.vec, abs(rho6.1-rho6.2))
r7.ML.conv.vec<-c(r7.ML.conv.vec, abs(rho7.1-rho7.2))

#Respiration Convergence check
Resp.ML.conv.vec<-c(Resp.ML.conv.vec, abs(Resp.1-Resp.2))
}

else{
#Simple Alpha convergence check
a1.ML.conv.vec<-c(a1.ML.conv.vec, NA)
a2.ML.conv.vec<-c(a2.ML.conv.vec, NA)
a3.ML.conv.vec<-c(a3.ML.conv.vec, NA)
a4.ML.conv.vec<-c(a4.ML.conv.vec, NA)

#Simple rho convergence check
r1.ML.conv.vec<-c(r1.ML.conv.vec, NA)
r2.ML.conv.vec<-c(r2.ML.conv.vec, NA)
r3.ML.conv.vec<-c(r3.ML.conv.vec, NA)
r4.ML.conv.vec<-c(r4.ML.conv.vec, NA)
r5.ML.conv.vec<-c(r5.ML.conv.vec, NA)
r6.ML.conv.vec<-c(r6.ML.conv.vec, NA)
r7.ML.conv.vec<-c(r7.ML.conv.vec, NA)

#Respiration convergence
Resp.ML.conv.vec<-c(Resp.ML.conv.vec, NA)
}

if(is.null(opt_fit1) & !is.null(opt_fit2)){
alpha1.1<-alpha1.2
alpha2.1<-alpha2.2
alpha3.1<-alpha3.2
alpha4.1<-alpha4.2

rho1.1<-rho1.2
rho2.1<-rho2.2
rho3.1<-rho3.2
rho4.1<-rho4.2
rho5.1<-rho5.2
rho6.1<-rho6.2
rho7.1<-rho7.2

Resp.1<-Resp.2
}

pars.to.monitor<-c('HR', 'Ypred')
start.time.temp<-Sys.time()

start.time.temp<-Sys.time()

dat.opt<-list(N1=N1,
             N2=N2,
             Xp=Xp,
             X=X,
             Y=Y,
             mu_HR=mu_HP,
             sigma_HR=sigma_HP,
             mu_R = mu_R,
             sigma_R = sigma_R,
             a1 = alpha1.1,
             a2 = alpha2.1,

```

```

        a3 = alpha3.1,
        a4 = alpha4.1,
        r1 = rho1.1,
        r2 = rho2.1,
        r3 = rho3.1,
        r4 = rho4.1,
        r5 = rho5.1,
        r6 = rho6.1,
        r7 = rho7.1,
        R = Resp.1
    )

fit.stan<-stan(file=paste0(wd, '/Stan_code/GP_3_b.stan'),
              data = dat.opt,
              init = init.list,
              warmup = warmup.temp,
              iter = n.iter.temp,
              refresh=100,
              chains = chain.temp,
              pars = pars.to.monitor,
              control = list(adapt_delta = .95,
                             max_treedepth = 15)
    )

run_time<-round(Sys.time()-start.time.temp, digits = 3)
units(run_time)<-'mins'
run_time<-as.numeric(run_time)

y_pred<-rstan::extract(fit.stan, 'Ypred')
PPG.new<-colMeans(y_pred$Ypred)
PPG.DF.new<-data.frame(PPG = PPG.new,
                      Time = Xp)

GP.summary<-as.data.frame(summary(fit.stan, pars = pars.to.monitor[-
length(pars.to.monitor)],
                                probs = c(.1, .9))$summary)

write.table(round(GP.summary, digits = 3),
            paste0(model.folder,
                  '/',
                  paste(dat.temp$ID,
                        dat.temp$Population,
                        dat.temp$Segment,
                        'GP_impute',
                        round(impute.LB.temp, digits = 2),
                        round(impute.UB.temp, digits = 2),
                        Hz.temp,
                        Hz.impute.temp,
                        Hz.GP.temp,
                        impute.fac.temp,
                        chain.temp,
                        warmup.temp,
                        n.iter.temp,
                        round(impute.LB.temp, digits = 2),
                        sep = '_'),
                  '.txt'),
            row.names = T,
            sep='\t'
    )

g1<-ggplot()+
  geom_line(data=PPG.DF.new, aes(x=Time, y = PPG), color = 'red', lty='dashed')+
  geom_line(data=PPG.impute.pred, aes(x=Time, y=PPG), color = 'black')+
  ggtitle(paste(dat.temp$ID, dat.temp$Population, dat.temp$Segment,
                'File Hz =', Hz.temp, 'GP Dataset Hz =', Hz.impute.temp,
                'Model Sampling Rate =', Hz.GP.temp, 'Chains =', chain.temp,
                'Warmup =', warmup.temp, 'Iter =', n.iter.temp))

ggsave(filename = paste0(graphics.folder, paste('/',

```

```

dat.temp$ID,
dat.temp$Population,
dat.temp$Segment,
Hz.temp,
Hz.impute.temp,
Hz.GP.temp,
impute.fac.temp,
chain.temp,
warmup.temp,
n.iter.temp,
round(impute.LB.temp, digits = 2),
sep = '_'), '.png'),

plot = g1,
width = 11,
height = 8,
units = 'in',
dpi = 300,
device = 'png')

g2<-traceplot(fit.stan, pars='HR', inc_warmup=T)+
ggtitle(paste('HR param', dat.temp$ID, dat.temp$Population, dat.temp$Segment,
'File Hz =', Hz.temp, 'GP Dataset Hz =', Hz.impute.temp,
'Model Sampling Rate =', Hz.GP.temp, '\n', 'Chains =', chain.temp,
'Warmup =', warmup.temp, 'Iter =', n.iter.temp))

ggsave(filename = paste0(Bayes.graphics, paste('/',
dat.temp$ID,
dat.temp$Population,
dat.temp$Segment,
Hz.temp,
Hz.impute.temp,
Hz.GP.temp,
impute.fac.temp,
chain.temp,
warmup.temp,
n.iter.temp,
round(impute.LB.temp, digits = 2),
sep = '_'), '.png'),

plot = g2,
width = 11,
height = 8,
units = 'in',
dpi = 300,
device = 'png')

PPG.impute.fin<-PPG.impute2
PPG.impute.fin$PPG[PPG.impute.fin$Time>impute.LB.temp &
PPG.impute.fin$Time<impute.UB.temp]<-PPG.new

IBI.GP.impute.temp<-iter.IBI(PPG.impute.fin$PPG, ds=Hz.impute.temp)$IBI.done
IBI.GP.impute.time<-sum.rev(IBI.GP.impute.temp)+dat.temp$Start.adj
write.table(round(IBI.GP.impute.temp[c(-1, -length(IBI.GP.impute.temp))], digits =
4),
paste0(IBI.impute.folder,
'/',
paste(dat.temp$ID,
dat.temp$Population,
dat.temp$Segment,
'GP_impute',
round(impute.LB.temp, digits = 2),
round(impute.UB.temp, digits = 2),
Hz.temp,
Hz.impute.temp,
Hz.GP.temp,
impute.fac.temp,
chain.temp,
warmup.temp,
n.iter.temp,

```

```

        round(impute.LB.temp, digits = 2),
        sep = '_'),
        '.txt'),
        row.names = F
    )

HR.est<-rstan::extract(fit.stan, 'HR')

#Basic simulation vectors
ID.vec<-c(ID.vec, dat.temp$ID)
impute.ID<-c(impute.ID, paste('Segment', i, sep = '_'))
CPU.vec<-c(CPU.vec, benchmarkme::get_cpu()$model_name)
Cores.vec<-c(Cores.vec, parallel::detectCores())
RAM.vec<-c(RAM.vec, paste(round(benchmarkme::get_ram()/1073741824), 'GB'))
pop.vec<-c(pop.vec, dat.temp$Population)
N.IBIs.vec<-c(N.IBIs.vec, N.IBIs)
start.vec<-c(start.vec, dat.temp$Start)
stop.vec<-c(stop.vec, dat.temp$Stop)
IBI.GP.rmssd.15<-c(IBI.GP.rmssd.15,
                  rmssd(IBI.GP.impute.temp[IBI.GP.impute.time>impute.LB.temp-7.5
&
IBI.GP.impute.time<impute.UB.temp+7.5]))
                  IBI.GP.rmssd.30<-c(IBI.GP.rmssd.30,
                  rmssd(IBI.GP.impute.temp[IBI.GP.impute.time>impute.LB.temp-15 &
IBI.GP.impute.time<impute.UB.temp+15]))
                  IBI.GP.rmssd.45<-c(IBI.GP.rmssd.45,
                  rmssd(IBI.GP.impute.temp[IBI.GP.impute.time>impute.LB.temp-22.5
&
IBI.GP.impute.time<impute.UB.temp+22.5]))
                  IBI.GP.rmssd.60<-c(IBI.GP.rmssd.60,
                  rmssd(IBI.GP.impute.temp[IBI.GP.impute.time>impute.LB.temp-30 &
IBI.GP.impute.time<impute.UB.temp+30]))
runtime<-c(runtime, run_time)
warmup.vec<-c(warmup.vec, warmup.temp)
iter.vec<-c(iter.vec, n.iter.temp)
Hz.vec<-c(Hz.vec, Hz.temp)
impute.Hz.vec<-c(impute.Hz.vec, Hz.impute.temp)
GP.Hz.vec<-c(GP.Hz.vec, Hz.GP.temp)
impute.fac.vec<-c(impute.fac.vec, impute.fac.temp)
impute.start<-c(impute.start, min.TIME2)
impute.end<-c(impute.end, max.TIME2)
impute.tot<-c(impute.tot, time.span)

#Parameter Means
HR.mean.vec<-c(HR.mean.vec, mean(HR.est$HR))

#Parameter sd's
HR.SD.vec<-c(HR.SD.vec, sd(HR.est$HR))

#Parameter R-hats
HR.Rhat.vec<-c(HR.Rhat.vec, GP.summary$Rhat[rownames(GP.summary)=='HR'])

#Parameter N_eff
HR.N_eff.vec<-c(HR.N_eff.vec, GP.summary$n_eff[rownames(GP.summary)=='HR'])

#Getting a vector of original value rmssd's
IBI.rmssd.15<-c(IBI.rmssd.15,
               rmssd(IBI.orig.temp$IBI.done[IBI.orig.time>impute.LB.temp-7.5 &
               IBI.orig.time<impute.UB.temp+7.5]))
IBI.rmssd.30<-c(IBI.rmssd.30,
               rmssd(IBI.orig.temp$IBI.done[IBI.orig.time>impute.LB.temp-15 &
               IBI.orig.time<impute.UB.temp+15]))
IBI.rmssd.45<-c(IBI.rmssd.45,
               rmssd(IBI.orig.temp$IBI.done[IBI.orig.time>impute.LB.temp-22.5 &

```

```

IBI.orig.time<impute.UB.temp+22.5]))
  IBI.rmssd.60<-c(IBE.rmssd.60,
    rmssd(IBE.orig.temp$IBE.done[IBE.orig.time>impute.LB.temp-30 &
      IBE.orig.time<impute.UB.temp+30]))

  #Getting a vector of mean replacement rmssd's
  IBI.MR.rmssd.15<-c(IBE.MR.rmssd.15,
    rmssd(IBE.mean.replace[IBE.orig.time>impute.LB.temp-7.5 &
      IBE.orig.time<impute.UB.temp+7.5]))
  IBI.MR.rmssd.30<-c(IBE.MR.rmssd.30,
    rmssd(IBE.mean.replace[IBE.orig.time>impute.LB.temp-15 &
      IBE.orig.time<impute.UB.temp+15]))
  IBI.MR.rmssd.45<-c(IBE.MR.rmssd.45,
    rmssd(IBE.mean.replace[IBE.orig.time>impute.LB.temp-22.5 &
      IBE.orig.time<impute.UB.temp+22.5]))
  IBI.MR.rmssd.60<-c(IBE.MR.rmssd.60,
    rmssd(IBE.mean.replace[IBE.orig.time>impute.LB.temp-30 &
      IBE.orig.time<impute.UB.temp+30]))

  #Getting a vector of hotdeck replacement rmssd's
  IBI.HD.rmssd.15<-c(IBE.HD.rmssd.15,
    rmssd(IBE.hotdeck.replace[IBE.orig.time>impute.LB.temp-7.5 &
      IBE.orig.time<impute.UB.temp+7.5]))
  IBI.HD.rmssd.30<-c(IBE.HD.rmssd.30,
    rmssd(IBE.hotdeck.replace[IBE.orig.time>impute.LB.temp-15 &
      IBE.orig.time<impute.UB.temp+15]))
  IBI.HD.rmssd.45<-c(IBE.HD.rmssd.45,
    rmssd(IBE.hotdeck.replace[IBE.orig.time>impute.LB.temp-22.5 &
      IBE.orig.time<impute.UB.temp+22.5]))
  IBI.HD.rmssd.60<-c(IBE.HD.rmssd.60,
    rmssd(IBE.hotdeck.replace[IBE.orig.time>impute.LB.temp-30 &
      IBE.orig.time<impute.UB.temp+30]))

  }
}
}
}
}
}
}
}

save(list = ls(),
  file = paste0(out.folder, '/', paste('GP_3', Sys.time(), n.sims, sep = '_'), '.RData'))

```

References

- Acharya, U. R., Joseph, K. P., Kannathal, N., Lim, C. M., & Suri, J. S. (2006). Heart rate variability: A review. *Medical and Biological Engineering and Computing*, *44*(12), 1031–1051. <https://doi.org/10.1007/s11517-006-0119-0>
- Adenauer, H., Catani, C., Keil, J., Aichinger, H., & Neuner, F. (2010). Is freezing an adaptive reaction to threat? Evidence from heart rate reactivity to emotional pictures in victims of war and torture. *Psychophysiology*, *47*, 315–322. <https://doi.org/10.1111/j.1469-8986.2009.00940.x>
- Akar, S. A., Kara, S., Latifoğlu, F., & Bilgiç, V. (2013). Spectral analysis of photoplethysmographic signals: The importance of preprocessing. *Biomedical Signal Processing and Control*, *8*(1), 16–22. <https://doi.org/10.1016/j.bspc.2012.04.002>
- Allen, J. (2007). Photoplethysmography and its application in clinical physiological measurement. *Physiological Measurement*, *28*(3), R1–R39. <https://doi.org/10.1088/0967-3334/28/3/R01>
- Asendorpf, J. B. (1990). Development of inhibition during childhood: Evidence for situational specificity and a two-factor model. *Developmental Psychology*, *26*, 721–730. <https://doi.org/10.1037//0012-1649.26.5.721>
- Balzarotti, S., Biassoni, F., Colombo, B., & Ciceri, M. R. (2017). Cardiac vagal control as a marker of emotion regulation in healthy adults: A review. *Biological Psychology*. <https://doi.org/10.1016/j.biopsycho.2017.10.008>
- Barbour, A. J., & Parker, R. L. (2014). psd: Adaptive, sine multitaper power spectral density estimation for R. *Computers & Geosciences*, *63*, 1–8.

<https://doi.org/10.1016/j.cageo.2013.09.015>

Barstead, M. G. (2018). IBI VizEdit v.1.2-beta: An RShiny application.

<https://doi.org/10.5281/zenodo.1209474>

Barstead, M. G., Danko, C. M., Chronis-Tuscano, A., O'Brien, K. A., Coplan, R. J., &

Rubin, K. H. (2018). Generalization of an early intervention for inhibited preschoolers to the classroom setting. *Journal of Child and Family Studies*, 27, 2943–2953. <https://doi.org/10.1007/s10826-018-1142-0>

Bayer, J. K., Rapee, R. M., Hiscock, H., Ukoumunne, O. C., Mihalopoulos, C., Clifford, S., & Wake, M. (2011). The Cool Little Kids randomised controlled trial:

Population-level early prevention for anxiety disorders. *BMC Public Health*, 11, 1–10. <https://doi.org/10.1186/1471-2458-11-11>

Beauchaine, T. (2001). Vagal tone, development, and Gray's motivational theory: toward an integrated model of autonomic nervous system functioning in psychopathology.

Development and Psychopathology, 13(2), 183–214.

<https://doi.org/10.1017/S0954579401002012>

Beauchaine, T. P. (2015). Respiratory sinus arrhythmia: A transdiagnostic biomarker of emotion dysregulation and psychopathology. *Current Opinion in Psychology*, 3, 43–

47. <https://doi.org/10.1016/j.copsyc.2015.01.017>

Beauchaine, T. P., & Thayer, J. F. (2015). Heart rate variability as a transdiagnostic

biomarker of psychopathology. *International Journal of Psychophysiology*, 98(2), 338–350. <https://doi.org/10.1016/j.ijpsycho.2015.08.004>

Berntson, G. G., Bigger, J. T., Eckberg, D. L., Grossman, P., Kaufmann, P. G., Marek, M., ... van der Molen, M. W. (1997). Heart rate variability: Origins, methods, and

- interpretive caveats. *Psychophysiology*, 34(6), 623–648.
<https://doi.org/10.1111/j.1469-8986.1997.tb02140.x>
- Berntson, G. G., Quigley, K. S., Jang, J. F., & Boysen, S. T. (1990). An approach to artifact identification: Application to heart period data. *Psychophysiology*, 27(5), 586–598. <https://doi.org/10.1111/j.1469-8986.1990.tb01982.x>
- Berntson, G. G., & Stowell, J. R. (1998). ECG artifacts and heart period variability: Don't miss a beat! *Psychophysiology*, 35(1), 127–132.
<https://doi.org/10.1017/S0048577298001541>
- Biopac Systems. (2016). AcqKnowledge. Goleta, CA: Biopac Systems, Inc.
- Biopac Systems. (2017). Application Note 109: 1-, 3-, 6-, and 12-Lead ECG. Goleta, CA: Biopac Systems, Inc.
- Biopac Systems. (2018). Bionomadix Wireless PPG and EDA Amplifier. Retrieved from <https://www.biopac.com/product/bionomadix-ppg-and-eda-amplifier/#product-tabs>
- Bishop, G., Spence, S. H., & McDonald, C. (2003). Can parents and teachers provide a reliable and valid report of behavioral inhibition? *Child Development*, 74, 1899–1917. <https://doi.org/10.1046/j.1467-8624.2003.00645.x>
- Bolanos, M., Nazeran, H., & Haltiwanger, E. (2006). Comparison of heart rate variability signal features derived from electrocardiography and photoplethysmography in healthy individuals. In *2006 International Conference of the IEEE Engineering in Medicine and Biology Society* (pp. 4289–4294). IEEE.
<https://doi.org/10.1109/IEMBS.2006.260607>
- Boyce, W. T., Quas, J., Alkon, A., Smider, N. A., Essex, M. J., & Kupfer, D. J. (2001). Autonomic reactivity and psychopathology in middle childhood. *British Journal of*

- Psychiatry*, 179, 144–150. <https://doi.org/10.1192/bjp.179.2.144>
- Brain-Body Center. (2007). CardioEdit software. University of Chicago.
- Brooker, R. J., & Buss, K. A. (2010). Dynamic measures of RSA predict distress and regulation in toddlers. *Developmental Psychobiology*, 52, 372–382. <https://doi.org/10.1002/dev.20432>
- Brosschot, J. F., & Thayer, J. F. (2003). Heart rate response is longer after negative emotions than after positive emotions. *International Journal of Psychophysiology*, 50, 181–187. [https://doi.org/10.1016/S0167-8760\(03\)00146-6](https://doi.org/10.1016/S0167-8760(03)00146-6)
- Bürkner, P.-C. (2017). **brms** : An R Package for Bayesian Multilevel Models Using Stan. *Journal of Statistical Software*, 80(1). <https://doi.org/10.18637/jss.v080.i01>
- Calkins, S. D. (1997). Cardiac vagal tone indices of temperamental reactivity and behavioral regulation in young children. *Developmental Psychobiology*, 31(1990), 125–135.
- Calkins, S. D., Fox, N. A., & Marshall, T. R. (1996). Behavioral and physiological antecedents of inhibited and uninhibited behavior. *Child Development*, 67(2), 523–540. <https://doi.org/10.1111/j.1467-8624.1996.tb01749.x>
- Chida, Y., & Hamer, M. (2008). Chronic psychosocial factors and acute physiological responses to laboratory-induced stress in healthy populations: A quantitative review of 30 years of investigations. *Psychological Bulletin*, 134(6), 829–885. <https://doi.org/10.1037/a0013342>
- Chronis-Tuscano, A., Danko, C. M., Rubin, K. H., Coplan, R. J., & Novick, D. R. (2018). Future Directions for Research on Early Intervention for Young Children at Risk for Social Anxiety. *Journal of Clinical Child and Adolescent Psychology*.

<https://doi.org/10.1080/15374416.2018.1426006>

Chronis-Tuscano, A., Degnan, K. A., Pine, D. S., Perez-Edgar, K., Henderson, H. A., Diaz, Y., ... Fox, N. A. (2009). Stable early maternal report of behavioral inhibition predicts lifetime social anxiety disorder in adolescence. *Journal of the American Academy of Child and Adolescent Psychiatry*, *48*(9), 928–935.

<https://doi.org/10.1097/CHI.0b013e3181ae09df>

Chronis-Tuscano, A., Rubin, K. H., O'Brien, K. A., Coplan, R. J., Thomas, S. R., Dougherty, L. R., ... Wimsatt, M. (2015). Preliminary evaluation of a multimodal early intervention program for behaviorally inhibited preschoolers. *Journal of Consulting and Clinical Psychology*, *83*, 534–540. <https://doi.org/10.1037/a0039043>

Clauss, J. A., & Blackford, J. U. (2012). Behavioral inhibition and risk for developing social anxiety disorder: A meta-analytic study. *Journal of the American Academy of Child and Adolescent Psychiatry*, *51*(10), 1066–1075.

<https://doi.org/10.1016/j.jaac.2012.08.002>

Couceiro, R., Carvalho, P., Paiva, R. P., Henriques, J., & Muehlsteff, J. (2012). Detection of motion artifacts in photoplethysmographic signals based on time and period domain analysis. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (pp. 2603–2606).

<https://doi.org/10.1109/EMBC.2012.6346497>

Denver, J. W., Reed, S. F., & Porges, S. W. (2007). Methodological issues in the quantification of respiratory sinus arrhythmia. *Biological Psychology*, *74*(2), 286–294. <https://doi.org/10.1016/j.biopsycho.2005.09.005>

Doussard-Roosevelt, J. a, Montgomery, L. A., & Porges, S. W. (2003). Short-term

- stability of physiological measures in kindergarten children: respiratory sinus arrhythmia, heart period, and cortisol. *Developmental Psychobiology*, 43(3), 230–242. <https://doi.org/10.1002/dev.10136>
- El-Sheikh, M., & Buckhalt, J. A. (2005). Vagal regulation and emotional intensity predict children's sleep problems. *Developmental Psychobiology*, 46(4), 307–317. <https://doi.org/10.1002/dev.20066>
- Elgendi, M. (2012). On the analysis of fingertip photoplethysmogram signals. *Current Cardiology Reviews*, 8(1), 14–25. <https://doi.org/10.2174/157340312801215782>
- Fox, N. A., Henderson, H. A., Marshall, P. J., Nichols, K. E., & Ghera, M. M. (2005). Behavioral inhibition: linking biology and behavior within a developmental framework. *Annual Review of Psychology*, 56, 235–262. <https://doi.org/10.1146/annurev.psych.55.090902.141532>
- Fox, N. A., Nichols, K. E., Henderson, H. A., Rubin, K. H., Schmidt, L. A., Hamer, D., ... Pine, D. S. (2005). Evidence for a Gene- Environment Interaction in Predicting Behavioral Inhibition in Middle Childhood. *Society*, 16(12), 921–926.
- Geisler, F. C. M., Vennwald, N., Kubiak, T., & Weber, H. (2010). The impact of heart rate variability on subjective well-being is mediated by emotion regulation. *Personality and Individual Differences*, 49(7), 723–728. <https://doi.org/10.1016/j.paid.2010.06.015>
- Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2013). *Bayesian Data Analysis* (3rd ed.). New York: CRC Press.
- Gelman, A., & Hill, J. (2007). *Data analysis using regression and multilevel/hierarchical models*. New York: Cambridge University Press.

- Gentzler, A. L., Santucci, A. K., Kovacs, M., & Fox, N. A. (2009). Respiratory sinus arrhythmia reactivity predicts emotion regulation and depressive symptoms in at-risk and control children. *Biological Psychology*, *82*(2), 156–163.
<https://doi.org/10.1016/j.biopsycho.2009.07.002>
- Giardino, N. D., Lehrer, P. M., & Edelberg, R. (2002). Comparison of finger plethysmograph to ECG in the measurement of heart rate variability. *Psychophysiology*, *39*(2), 246–253. <https://doi.org/10.1017/S0048577202990049>
- Glenn, A. L., Lochman, J. E., Dishion, T., Powell, N. P., Boxmeyer, C., Kassing, F., ... Romero, D. (2018). Toward tailored interventions: Sympathetic and parasympathetic functioning predicts responses to an intervention for conduct problems delivered in two formats. *Prevention Science*. <https://doi.org/10.1007/s11121-017-0859-0>
- Graham, J. W., Olchowski, A. E., & Gilreath, T. D. (2007). How many imputations are really needed? Some practical clarifications of multiple imputation theory. *Prevention Science*, *8*(3), 206–213. <https://doi.org/10.1007/s11121-007-0070-9>
- Grippo, A. J., Lamb, D. G., Carter, C. S., & Porges, S. W. (2007). Social isolation disrupts autonomic regulation of the heart and influences negative affective behaviors. *Biological Psychiatry*, *62*(10), 1162–1170.
<https://doi.org/10.1016/j.biopsych.2007.04.011>
- Grund, S., Robitzsch, A., & Luedtke, O. (2018). mitml: Tools for multiple imputation in multilevel modeling. Retrieved from <https://cran.r-project.org/package=mitml>
- Guy, L., Souders, M., Bradstreet, L., Delussey, C., & Herringto, J. D. (2014). Brief report: Emotion regulation and respiratory sinus arrhythmia in autism spectrum disorder. *Journal of Autism and Developmental Disorders*, *44*(10), 2614–2620.

<https://doi.org/10.1007/s10803-014-2124-8>

Haines, D. E. (2013). *Fundamental neuroscience for basic and clinical applications* (4th ed.). Philadelphia, PA: Elsevier.

Haines, D. E., & Mihailoff, G. A. (2013). A synopsis of cranial nerves of the brainstem. In D. E. Haines (Ed.), *Fundamental Neuroscience for Basic and Clinical Applications* (4th ed., pp. 181–197). Philadelphia, PA: Elsevier.

Hamilton, J. L., & Alloy, L. B. (2016). Atypical reactivity of heart rate variability to stress and depression across development: Systematic review of the literature and directions for future research. *Clinical Psychology Review, 50*, 67–79.

<https://doi.org/10.1016/j.cpr.2016.09.003>

Hastings, P. D., Nuselovici, J. N., Utendale, W. T., Coutya, J., McShane, K. E., & Sullivan, C. (2008). Applying the polyvagal theory to children's emotion regulation: Social context, socialization, and adjustment. *Biological Psychology, 79*, 299–306.

<https://doi.org/10.1016/j.biopsycho.2008.07.005>

Hastings, P. D., Sullivan, C., McShane, K. E., Coplan, R. J., Utendale, W. T., & Vyncke, J. D. (2008). Parental socialization, vagal regulation, and preschoolers' anxious difficulties: Direct mothers and moderated fathers. *Child Development, 79*(1), 45–

64. <https://doi.org/10.1111/j.1467-8624.2007.01110.x>

Hegarty-Craver, M., Gilchrist, K. H., Propper, C. B., Lewis, G. F., DeFilipp, S. J., Coffman, J. L., & Willoughby, M. T. (2017). Automated respiratory sinus arrhythmia measurement: Demonstration using executive function assessment. *Behavior Research Methods, 1–8*. <https://doi.org/10.3758/s13428-017-0950-2>

Hellhammer, D. H., Wüst, S., & Kudielka, B. M. (2009). Salivary cortisol as a biomarker

- in stress research. *Psychoneuroendocrinology*, 34(2), 163–171.
<https://doi.org/10.1016/j.psyneuen.2008.10.026>
- Henderson, H. a., Marshall, P. J., Fox, N. a., & Rubin, K. H. (2004). Psychophysiological and Behavioral Evidence for Varying Forms and Functions of Nonsocial Behavior in Preschoolers. *Child Development*, 75(1), 251–263. <https://doi.org/10.1111/j.1467-8624.2004.00667.x>
- Hirshfeld-Becker, D. R., Biederman, J., Henin, A., Faraone, S. V., Davis, S., Harrington, K., & Rosenbaum, J. F. (2007). Behavioral inhibition in preschool children at risk is a specific predictor of middle childhood social anxiety: A five-year follow-up. *Journal of Developmental & Behavioral Pediatrics*, 28(3), 225–233.
<https://doi.org/10.1097/01.DBP.0000268559.34463.d0>
- Huikuri, H. V., Mäkikallio, T., Airaksinen, K. E. J., Mitrani, R., Castellanos, A., & Myerburg, R. J. (1999). Measurement of heart rate variability: A clinical tool or a research toy? *Journal of the American College of Cardiology*, 34(7), 1878–1883.
[https://doi.org/10.1016/S0735-1097\(99\)00468-4](https://doi.org/10.1016/S0735-1097(99)00468-4)
- Huikuri, H. V., & Stein, P. K. (2013). Heart rate variability in risk stratification of cardiac patients. *Progress in Cardiovascular Diseases*, 56, 153–159.
<https://doi.org/10.1016/j.pcad.2013.07.003>
- Insel, T., Cuthbert, B., Garvey, M., Heinssen, R., Pine, D. S., Quinn, K., ... Wang, P. (2010). Research Domain Criteria (RDoC): Toward a new classification framework for research on mental disorders. *American Journal of Psychiatry*, 167(7), 748–751.
<https://doi.org/10.1176/appi.ajp.2010.09091379>
- Jennings, J. R., Berg, W. K., Hutcheson, J. S., Obrist, P., Porges, S., & Turpin, G. (1981).

- Publication guidelines for heart rate studies in men. *Psychophysiology*, 18(3), 226–231. <https://doi.org/10.1016/j.jacc.2007.01.076>. White
- Kagan, J., Reznick, J. S., & Snidman, N. (1987). The physiology and psychology of behavioral inhibition in children. *Child Development*, 58(6), 1459–1473. <https://doi.org/10.2307/1130685>
- Kagan, J., Reznick, J. S., & Snidman, N. (1988). Biological bases of childhood shyness. *Science (New York, N.Y.)*, 240(4849), 167–171. <https://doi.org/10.1126/science.3353713>
- Kaufmann, T., Sutterlin, S., Schulz, S. M., & Vogeles, C. (2011). ARTiiFACT: A tool for heart rate artifact processing and heart rate variability analysis. *Behavior Research Methods*, 43(4), 1161–1170. <https://doi.org/10.3758/s13428-011-0107-7>
- Kenkel, W. M., Suboc, G., & Carter, S. C. (2014). Autonomic, behavioral and neuroendocrine correlates of paternal behavior in male prairie voles. *Physiology and Behavior*, 128, 252–259. <https://doi.org/10.1016/j.physbeh.2014.02.006>
- Kim, B. S., & Yoo, S. K. (2006). Motion artifact reduction in photoplethysmography using independent component analysis. *IEEE Transactions on Biomedical Engineering*, 53(3), 566–568. <https://doi.org/10.1109/TBME.2005.869784>
- Kirschbaum, C., Pirke, K.-M., & Hellhammer, D. H. (1993). The “Trier Social Stress Test”: A tool for investigating psychobiological stress responses in a laboratory setting. *Neuropsychobiology*, 28(1–2), 76–81. <https://doi.org/10.1159/000119004>
- Kleiger, R. E., Miller, J. P., Bigger, J. T., & Moss, A. J. (1987). Decreased heart rate variability and its association with increased mortality after acute myocardial infarction. *The American Journal of Cardiology*, 59, 256–262.

[https://doi.org/10.1016/0002-9149\(87\)90795-8](https://doi.org/10.1016/0002-9149(87)90795-8)

- Kleiger, R. E., Stein, P. K., & Bigger, J. T. (2005). Heart rate variability: Measurement and clinical utility. *Annals of Noninvasive Electrocardiography*, *10*, 88–101.
- Kogan, A., Gruber, J., Shallcross, A. J., Ford, B. Q., & Mauss, I. B. (2013). Too much of a good thing? Cardiac vagal tone's nonlinear relationship with well-being. *Emotion*, *13*, 599–604. <https://doi.org/10.1037/a0032725>
- Kogan, A., Oveis, C., Carr, E. W., Gruber, J., Mauss, I. B., Shallcross, A., ... Keltner, D. (2014). Vagal activity is quadratically related to prosocial traits, prosocial emotions, and observer perceptions of prosociality. *Journal of Personality and Social Psychology*, *107*, 1051–1063. <https://doi.org/10.1037/a0037509>
- Kudielka, B. M., Hellhammer, D. H., & Kirschbaum, C. (2010). Ten years of research with the Trier Social Stress Test - revisited. In E. Harmon-Jones & P. Winkielman (Eds.), *Social neuroscience: Integrating biological and psychological explanations for social behavior* (pp. 56–83). New York: Guilford.
- Laborde, S., Mosley, E., & Thayer, J. F. (2017). Heart rate variability and cardiac vagal tone in psychophysiological research - Recommendations for experiment planning, data analysis, and data reporting. *Frontiers in Psychology*, *8*(FEB). <https://doi.org/10.3389/fpsyg.2017.00213>
- Lamm, C., Porges, E. C., Cacioppo, J. T., & Decety, J. (2008). Perspective taking is associated with specific facial responses during empathy for pain. *Brain Research*, *1227*, 153–161. <https://doi.org/10.1016/j.brainres.2008.06.066>
- Lee, H., Lee, J., Jung, W., & Lee, G. (2007). The Periodic Moving Average Filter for Removing Motion Artifacts from PPG Signals. *International Journal Of Control*

Automation And Systems, 5(6), 701–706.

Lewis, G. F., Furman, S. A., McCool, M. F., & Porges, S. W. (2012). Statistical strategies to quantify respiratory sinus arrhythmia: Are commonly used metrics equivalent?

Biological Psychology, 89(2), 349–364.

<https://doi.org/10.1016/j.biopsycho.2011.11.009>

LoBue, V., & Thrasher, C. (2014). The Child Affective Facial Expression (CAFE) set:

Validity and reliability from untrained adults. *Frontiers in Psychology*, 5(OCT), 1–8.

<https://doi.org/10.3389/fpsyg.2014.01532>

Lu, S., Zhao, H., Ju, K., Shin, K., Lee, M., Shelley, K., & Chon, K. H. (2008). Can photoplethysmography variability serve as an alternative approach to obtain heart rate variability information? *Journal of Clinical Monitoring and Computing*, 22(1),

23–29. <https://doi.org/10.1007/s10877-007-9103-y>

Malik, M., Bigger, J. T., Camm, A. J., Kleiger, R. E., Malliani, A., Moss, A. J., &

Schwartz, P. J. (1996). Heart rate variability: Standards of measurement,

physiological interpretation, and clinical use. *European Heart Journal*, 17, 354–381.

<https://doi.org/10.1093/oxfordjournals.eurheartj.a014868>

Marshall, P. J., & Stevenson-Hinde, J. (1998). Behavioral inhibition, heart period, and

respiratory sinus arrhythmia in young children. *Developmental Psychobiology*,

33(3), 283–292. [https://doi.org/10.1002/\(SICI\)1098-2302\(199811\)33:3<283::AID-](https://doi.org/10.1002/(SICI)1098-2302(199811)33:3<283::AID-DEV8>3.0.CO;2-N)

[DEV8>3.0.CO;2-N](https://doi.org/10.1002/(SICI)1098-2302(199811)33:3<283::AID-DEV8>3.0.CO;2-N)

McKinley, P. S., Shapiro, P. A., Bagiella, E., Myers, M. M., De Meersman, R. E., Grant,

I., & Sloan, R. P. (2003). Deriving heart period variability from blood pressure

waveforms. *Journal of Applied Physiology*, 95(4), 1431–1438.

<https://doi.org/10.1152/japplphysiol.01110.2002>

Mifsud, C., & Rapee, R. M. (2005). Early intervention for childhood anxiety in a school setting: Outcomes for an economically disadvantaged population. *Journal of the American Academy of Child and Adolescent Psychiatry, 44*(10), 996–1004.

<https://doi.org/10.1097/01.chi.0000173294.13441.87>

Miller, J. G., Choccol, C., Nuselovici, J. N., Utendale, W. T., Simard, M., & Hastings, P. D. (2013). Children's dynamic RSA change during anger and its relations with parenting, temperament, and control of aggression. *Biological Psychology, 92*(2), 417–425. <https://doi.org/10.1016/j.biopsycho.2012.12.005>

Miller, J. G., Kahle, S., & Hastings, P. D. (2017). Moderate baseline vagal tone predicts greater prosociality in children. *Developmental Psychology, 53*, 274–289.

<https://doi.org/10.1037/dev0000238>

Moss, P. A. (1994). Can there Be validity without reliability? *Educational Researcher, 23*, 5–12. <https://doi.org/10.3102/0013189X023002005>

Muris, P., van Brakel, A. M. L., Arntz, A., & Schouten, E. (2011). Behavioral inhibition as a risk factor for the development of childhood anxiety disorders: A longitudinal study. *Journal of Child and Family Studies, 20*(2), 157–170.

<https://doi.org/10.1007/s10826-010-9365-8>

Naftel, J. P., Ard, M. D., & Fratkin, J. D. (2013). The cell biology of neurons and glia. In *Fundamental Neuroscience for Basic and Clinical Applications2* (4th ed., pp. 14–31). Philadelphia, PA: Elsevier.

Newlin, D. B., & Levenson, R. W. (1979). Pre-ejection period: Measuring beta-adrenergic influences upon the heart. *Psychophysiology, 16*(6), 546–552.

<https://doi.org/10.1111/j.1469-8986.1979.tb01519.x>

Noldus. (2015). Observer XT. Retrieved from <https://www.noldus.com/human-behavior-research/products/the-observer-xt>

Park, G., & Thayer, J. F. (2014). From the heart to the mind: Cardiac vagal tone modulates top-down and bottom-up visual perception and attention to emotional stimuli. *Frontiers in Psychology*. <https://doi.org/10.3389/fpsyg.2014.00278>

Penttila, J., Helminen, A., Jartti, T., Kuusela, T., Huikuri, H. V., Tulppo, M. P., ... Scheinin, H. (2001). Time domain, geometrical and frequency domain analysis of cardiac vagal outflow: effects of various respiratory patterns. *Clinical Physiology*, *21*(3), 365–376. <https://doi.org/10.1046/j.1365-2281.2001.00337.x>

Pflugradt, M., Geissdoerfer, K., Goernig, M., & Orglmeister, R. (2017). A Fast Multimodal Ectopic Beat Detection Method Applied for Blood Pressure Estimation Based on Pulse Wave Velocity Measurements in Wearable Sensors. *Sensors*, *17*(1), 158. <https://doi.org/10.3390/s17010158>

Porges, S. W. (1985). *4,510,944*. U.S.: U.S. Patent Office.

Porges, S. W. (1988). MxEdit v2.01. Bethesda, MD: Delta Biometrics.

Porges, S. W. (1995). Orienting in a defensive world: Mammalian modifications of our evolutionary heritage. A polyvagal theory. *Psychophysiology*, *32*, 301–318. <https://doi.org/10.1111/j.1469-8986.1995.tb01213.x>

Porges, S. W. (2001). The polyvagal theory: Phylogenetic substrates of a social nervous system. *International Journal of Psychophysiology*, *42*(2), 123–146. [https://doi.org/10.1016/S0167-8760\(01\)00162-3](https://doi.org/10.1016/S0167-8760(01)00162-3)

Porges, S. W. (2003). Social engagement and attachment: A phylogenetic perspective.

- Annals of the New York Academy of Sciences*, 1008, 31–47.
<https://doi.org/10.1196/annals.1301.004>
- Porges, S. W. (2007). The polyvagal perspective. *Biological Psychology*, 74(2), 116–143.
<https://doi.org/10.1016/j.biopsycho.2006.06.009>
- Porges, S. W., & Byrne, E. A. (1992). Research methods for measurement of heart rate and respiration. *Biological Psychology*, 34(2–3), 93–130.
[https://doi.org/10.1016/0301-0511\(92\)90012-J](https://doi.org/10.1016/0301-0511(92)90012-J)
- Porges, S. W., Doussard-Roosevelt, J. A., Portales, A. L., & Greenspan, S. I. (1996). Infant regulation of the vagai “brake” predicts child behavior problems: A psychobiological model of social behavior. *Developmental Psychobiology*, 29(August), 697–712. Retrieved from
<http://ezproxy.lib.utexas.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=psyhref&AN=DP.BI.FIG.PORGES.IRVPCB>
- Quintana, D. S., Alvares, G. A., & Heathers, J. A. J. (2016). Guidelines for Reporting Articles on Psychiatry and Heart rate variability (GRAPH): Recommendations to advance research communication. *Translational Psychiatry*, 6(5), e803.
<https://doi.org/10.1038/tp.2016.73>
- Rapee, R. M., & Jacobs, D. (2002). The reduction of temperamental risk for anxiety in withdrawn preschoolers: A pilot study. *Behavioural and Cognitive Psychotherapy*, 30(2), 211–215. <https://doi.org/10.1017/S1352465802002084>
- Rapee, R. M., Kennedy, S., Ingram, M., Edwards, S., & Sweeney, L. (2005). Prevention and early intervention of anxiety disorders in inhibited preschool children. *Journal of Consulting and Clinical Psychology*, 73(3), 488–497.

<https://doi.org/10.1037/0022-006X.73.3.488>

Rasmussen, C. E., & Williams, C. K. I. (2006). *Gaussian processes for machine learning*. Cambridge, MA: MIT Press.

Richard Jennings, J., Allen, B., Gianaros, P. J., Thayer, J. F., & Manuck, S. B. (2015). Focusing neurovisceral integration: Cognition, heart rate variability, and cerebral blood flow. *Psychophysiology*. <https://doi.org/10.1111/psyp.12319>

Rottenberg, J., Wilhelm, F. H., Gross, J. J., & Gotlib, I. H. (2002). Respiratory sinus arrhythmia as a predictor of outcome in major depressive disorder. *Journal of Affective Disorders*, *71*(1–3), 265–272. [https://doi.org/10.1016/S0165-0327\(01\)00406-2](https://doi.org/10.1016/S0165-0327(01)00406-2)

Rubin, K. H. (2001). *The Play Observation Scale*. College Park, MD: University of Maryland.

Rubin, K. H., Coplan, R. J., & Bowker, J. C. (2009). Social Withdrawal in Childhood. *Annual Review of Psychology*. <https://doi.org/10.1146/annurev.psych.60.110707.163642>

Rubin, K. H., Hastings, P. D., Stewart, S. L., Henderson, H. a, & Chen, X. (1997). The consistency and concomitants of inhibition: some of the children, all of the time. *Child Development*, *68*(3), 467–483. <https://doi.org/10.2307/1131672>

Sakaki, M., Yoo, H. J., Nga, L., Lee, T. H., Thayer, J. F., & Mather, M. (2016). Heart rate variability is associated with amygdala functional connectivity with MPFC across younger and older adults. *NeuroImage*, *139*, 44–52. <https://doi.org/10.1016/j.neuroimage.2016.05.076>

Saul, J. (1990). Beat-to-beat variations of heart rate reflect modulation of cardiac

autonomic outflow. *Physiology*, 5(1), 32–37.

<https://doi.org/10.1152/physiologyonline.1990.5.1.32>

Schäfer, A., & Vagedes, J. (2013). How accurate is pulse rate variability as an estimate of heart rate variability?: A review on studies comparing photoplethysmographic technology with an electrocardiogram. *International Journal of Cardiology*, 166(1), 15–29. <https://doi.org/10.1016/j.ijcard.2012.03.119>

Schipke, J. D., Pelzer, M., & Arnold, G. (1999). Effect of respiration on short-term heart rate variability. *Journal of Clinical and Basic Cardiology*, 2, 92–95. Retrieved from <http://www.kup.at/journals/summary/39.html>

Schmidt, L. a., Fox, N. a., Schulkin, J., & Gold, P. W. (1999). Behavioral and psychophysiological correlates of self-presentation in temperamentally shy children. *Developmental Psychobiology*, 35(2), 119–135. [https://doi.org/10.1002/\(SICI\)1098-2302\(199909\)35:2<119::AID-DEV5>3.0.CO;2-G](https://doi.org/10.1002/(SICI)1098-2302(199909)35:2<119::AID-DEV5>3.0.CO;2-G)

Seilmayer, M. (2016). spectral: Common methods of spectral data analysis. R package version 1.0.1. Retrieved from <https://cran.r-project.org/package=spectral%7D>

Selvaraj, N., Jaryal, A., Santhosh, J., Deepak, K. K., & Anand, S. (2008). Assessment of heart rate variability derived from finger-tip photoplethysmography as compared to electrocardiography. *Journal of Medical Engineering and Technology*, 32(6), 479–484. <https://doi.org/10.1080/03091900701781317>

Shaffer, F., & Ginsberg, J. P. (2017). An overview of heart rate variability metrics and norms. *Frontiers in Public Health*, 5, 1–17.

<https://doi.org/10.3389/fpubh.2017.00258>

Sherwood, A., Allen, M. T., Fahrenberg, J., Kelsey, R. M., Lovallo, W. R., & Doornen, L.

- J. P. (1990). Methodological Guidelines for Impedance Cardiography. *Psychophysiology*, 27(1), 1–23. <https://doi.org/10.1111/j.1469-8986.1990.tb02171.x>
- Shumway, R. H., & Stoffer, D. S. (2017). *Time series analysis and its applications* (4th ed.). New York: Springer Nature.
- Sloan, D. M. (2004). Emotion regulation in action: emotional reactivity in experiential avoidance. *Behaviour Research and Therapy*, 42, 1257–1270. <https://doi.org/10.1016/j.brat.2003.08.006>
- Smith, R., Thayer, J. F., Khalsa, S. S., & Lane, R. D. (2017). The hierarchical basis of neurovisceral integration. *Neuroscience and Biobehavioral Reviews*, 75, 274–296. <https://doi.org/10.1016/j.neubiorev.2017.02.003>
- Stan Development Team. (2017). Stan Modeling Language: A User's Guide and Reference Manual.
- Stoffer, D. (2017). astsa: Applied statistical time series analysis. R package version 1.8. Retrieved from <https://cran.r-project.org/package=astsa>
- Sukor, J. A., Redmond, S. J., & Lovell, N. H. (2011). Signal quality measures for pulse oximetry through waveform morphology analysis. *Physiological Measurement*, 32(3), 369–384. <https://doi.org/10.1088/0967-3334/32/3/008>
- Taylor, E. W., Leite, C. A. C., Sartori, M. R., Wang, T., Abe, A. S., & Crossley, D. A. (2014). The phylogeny and ontogeny of autonomic control of the heart and cardiorespiratory interactions in vertebrates. *Journal of Experimental Biology*, 217(5), 690–703. <https://doi.org/10.1242/jeb.086199>
- Thayer, J. F., Åhs, F., Fredrikson, M., Sollers, J. J., & Wager, T. D. (2012). A meta-analysis of heart rate variability and neuroimaging studies: Implications for heart

- rate variability as a marker of stress and health. *Neuroscience and Biobehavioral Reviews*, 36(2), 747–756. <https://doi.org/10.1016/j.neubiorev.2011.11.009>
- Thayer, J. F., Friedman, B. H., & Borkovec, T. D. (1996). Automatic characteristics of generalized anxiety disorder and worry. *Biological Psychiatry*, 39(95), 255–266. [https://doi.org/10.1016/0006-3223\(95\)00136-0](https://doi.org/10.1016/0006-3223(95)00136-0)
- Thayer, J. F., Hansen, A. L., Saus-Rose, E., & Johnsen, B. H. (2009). Heart rate variability, prefrontal neural function, and cognitive performance: The neurovisceral integration perspective on self-regulation, adaptation, and health. *Annals of Behavioral Medicine*, 37(2), 141–153. <https://doi.org/10.1007/s12160-009-9101-z>
- Thayer, J. F., & Lane, R. D. (2000). A model of neurovisceral integration in emotion regulation and dysregulation. *Journal of Affective Disorders*, 61(3), 201–216. [https://doi.org/10.1016/S0165-0327\(00\)00338-4](https://doi.org/10.1016/S0165-0327(00)00338-4)
- Tukey, J. W. (1977). *Exploratory data analysis*. Reading, MA: Addison-Wesley.
- Utendale, W. T., Nuselovici, J., Saint-Pierre, A. B., Hubert, M., Chochol, C., & Hastings, P. D. (2014). Associations between inhibitory control, respiratory sinus arrhythmia, and externalizing problems in early childhood. *Developmental Psychobiology*, 56, 686–699. <https://doi.org/10.1002/dev.21136>
- Vetrugno, R., Liguori, R., Cortelli, P., & Montagna, P. (2003). Sympathetic skin response. *Clinical Autonomic Research*, 13(4), 256–270. <https://doi.org/10.1007/s10286-003-0107-5>