April 2006

Multimodal MP3 Jukebox

Alexander Lee White
Worcester Polytechnic Institute

Brett William Dickson
Worcester Polytechnic Institute

MULTIMODAL MP3 JUKEBOX

A Major Qualifying Project Report Submitted To:
Professor Edward A. Clancy
Professor Donald R. Brown
WORCESTER POLYTECHNIC INSTITUTE

By:

Brett W. Dickson

Alexander L. White

In Cooperation With:
Bruce Zhang
Bose Corporation, Framingham, MA

April 26, 2006

Approved by:

Professor Edward A. Clancy

Professor D. Richard Brown, III

Abstract

This report, prepared for the Worcester Polytechnic Institute, describes the design, testing, and analysis of a voice- and button-controlled MP3 Jukebox for automotive applications. This project, sponsored by the Bose Corporation, incorporated modern software components and complete interaction features. An inexpensive, high-function driving simulation system was developed and used in conjunction with the Peripheral Detection Task (PDT) to measure driver distraction. As tested with four subjects, Jukebox interaction increased the overall median reaction time by 133 milliseconds.

Authorship Page

Brett W. Dickson	Designed test runs Conducted testing on four subjects Scored recorded test runs for driving violations Contributed to development of grammar file for speech recognition engine Major contributions to: Abstract, Executive Summary, Introduction, Background, Focus Group, Testing Methodology, Results and Analysis (SUI Accuracy Analysis), Conclusions, Recommendations, Appendices
Alexander L. White	Conducted Focus Group Implemented the MP3 Jukebox Interfaced Java with VoCon 3200 speech recognition engine Interfaced Java with RealSpeak Solo text-to-speech engine Implemented testing suite Flagged testing data by event conditions Contributed to development of grammar file for speech recognition engine Performed SAS instances on Peripheral Detection Task data Major contributions to: Abstract, Results and Analysis (Driver Distraction Analysis), Conclusions, Recommendations, Appendices Minor contributions to: Background
Jonathan Leslie	Contributions to Background (Previous Project Summary, Portable Music)

Acknowledgments

Prof. Edward A. Clancy

Prof. D. Richard Brown, III

Bruce Zhang

Cathy Emmerton

Jonathan Leslie

Paul Liberman

Michael Sikorski

Stefan Slonevskiy

WPI's Electrical and Computer Engineering Department

WPI's Academic Technology Center

And all the other ECE and Bose staff members who helped us along the way

Executive Summary

Recent advances in the field of electronics have altered many aspects of everyday life. Electrical devices are providing today's world with new levels of entertainment, convenience and safety; much of which seemed unfeasible ten to fifteen years ago. This trend has also held true in the automotive industry, and is evident by the plethora of features available in today's automobiles. Many of today's vehicles are equipped with entertainment features such as rear seat DVD players and multiple disc CD changers. Satellite radio is becoming increasingly popular, and interactive navigation systems have taken precedence over maps and road atlases. Recently, automobiles have began to make use of another entertainment technology; MP3 compression. This technology enables efficient storage of large volumes of music with a minimal loss of sound quality. Newer audio systems can read both traditional music CDs and MP3 CDs allowing users to carry hundreds of songs on a single disc. One of the inherent problems associated with carrying large music databases is that the user must navigate the entire collection in order to choose a specific song or artist. This task can become very challenging in an automotive environment and may cause excessive distraction to the driver of the automobile. This project explores the possibility of using speech recognition technology as a solution to this problem. Speech recognition allows users to maintain visual contact with the road as well as physical contact with the steering wheel and other automotive controls.

This project is the second phase in the exploration of a speech-controlled MP3 player in an automotive environment. The first phase was completed during the 2004-2005 academic year by a group of three WPI students. At the completion of their project, they had successfully developed an MP3 Jukebox that allowed users to verbally select songs from large music collections as well as perform basic control functions such as pause, resume, next song, and previous song. The main goal for our project was to expand upon the previous design by adding new and innovative interaction features that would allow users to easily navigate their music collections. Additionally, we created an inexpensive testing environment that enabled us to analyze the amount of distraction associated with the new song selection methods. Driver distraction analysis would allow us to implement the design with the most user friendly interface while minimizing distraction.

The first step in implementing new song selection techniques into the Jukebox was to determine which features would be both practical and feasible in an automotive setting. Background research was conducted to examine existing song selection methods in PC-based MP3 players. Also, a focus group was held to brainstorm any possible song selection features using speech recognition. At the completion of this process, all methods were reviewed and the final song selection techniques were chosen based on feasibility, time to implement, and practicality in an automotive environment. Some of the features implemented into the design include: user ratings, playcounts, shuffle modes, and Jukebox feedback regarding song and playlist information. Additionally, two separate user interfaces were implemented. The first was a strictly speech-oriented interface where all commands were issued verbally by the driver, and the second was a multimodal interface where buttons located on the steering wheel could be used to execute some commands. To determine which interface would be best suited in a driving environment, driver distraction testing was completed using human subjects. The main goal of the testing phase was to gain insight regarding any trends associated with different interfaces or command types that may cause excessive distraction to a driver.

To gather driver distraction information, a testing suite had to be developed that would allow for easy analysis of distraction levels. Additional background research was completed to investigate common methods used to measure driver distraction. It was decided to use a combination of techniques reviewed during research. We employed a driving simulator in conjunction with the Peripheral Detection Task (PDT) as the focal point of the testing suite which is further detailed below. These methods provided us with the pertinent information needed to determine driver distraction levels at a price within the scope of this project.

The PC racing game *Need for Speed: Underground 2* by Electronic Arts was used as the driving simulator. The game display was projected on a large white wall with a display size of approximately eight feet by six feet. The test subject sat in front of a table about ten feet from the wall; a microphone, a set of speakers, and a PC racing wheel made by Logitech were placed on the table. At the subject's feet were mounted foot pedals that were part of the Logitech racing wheel. Test participants were asked to navigate a specific course on the video game while interacting with the Jukebox. As the subject was driving, they were asked to issue specific commands while maintaining driving performance and responding to PDT targets. Each command that they were asked to issue was part of a test run that had been developed prior to

testing. Each test run consisted of different types of commands (questions, navigation, song selection etc.) as well as a specific interface they were required to use (button vs. non-buttons). Driving performance metrics such as lane violations and speed maintenance were recorded from each test run. The participants were also asked to complete the Peripheral Detection Task while navigating the course and interacting with the Jukebox. The PDT required subjects to respond to visual targets (large red dots) presented in their peripheral field of vision by pressing one of the buttons on the steering wheel. The amount of time necessary for a participant to respond to a target would give an indication of the amount of distraction they were currently experiencing. The PDT target reaction times in conjunction with driving performance metrics allowed for analysis of driver distraction associated with various commands and interfaces. The accuracy of the design was also examined by recording the number of SUI errors made during each test run.

Upon completion of driver distraction testing, driving performance and PDT results as well as SUI accuracy data were analyzed and interpreted. Overall, the Jukebox exhibited an accuracy rate of 93.21%. This figure is down from 98% which was displayed in last year's design, however the number of acceptable commands has increased from about fifteen to approximately 1,350. The number of user errors were also recorded corresponding to errors made by test participants such as issuing an inaccurate command or fumbling the wording of a command. The number of user errors during test runs using the multimodal button interface was half of that when using strictly verbal commands. This trend illustrates that users may have been more comfortable using the button interface than they were when using all verbal commands.

The test results exhibited some trends regarding the overall distraction associated with the Jukebox; however no clear differences existed between interface and command types. While there were no significant changes in distraction between button- and voice-controlled interfaces, qualitative data indicated that the test subjects preferred button control when applicable. The overall median distraction time during test runs using the Jukebox was approximately 640 ms while the distraction time during control runs (runs without Jukebox interaction) was 507 ms. These results show that the Jukebox produced an increase in distraction time of approximately 133 ms on runs that required Jukebox interaction. The increase in distraction time rose to approximately 300 ms when a PDT target was issued at the same instance a command was being issued. This distraction time appears similar to distractions caused by other common sources used in an automotive setting such as cell phones. Cell phones have been shown to cause

between 400 ms and 500 ms of distraction beyond normal human reaction time (Green, 2000). No explicit trends were exhibited between driving errors during the separate test runs.

Upon the completion of the project, the design process that we adopted along with the driver distraction testing has enabled us to make recommendations for future work on this project. One of the major recommendations is to continue researching and implementing new song selection and interaction features. We had hoped to implement some of the following features, but due to time restrictions, we were not able to include them in our final design:

- Selection by a keyword that would play all songs containing that keyword in their title
- Classify and choose songs based on the signal's power distribution
- Filter or modify playlists on the fly by removing or adding specific songs
- Implement news and email retrieval from the Jukebox through the use of a wireless internet connection
- Implement additional automobile controls such as temperature, windows and locks

Additional recommendations include determining optimal speech recognition parameters (settings used by the speech recognizer to distinguish the incoming signal) to increase recognition accuracy, as well as using a driver education simulator instead of a video game during testing.

Table of Contents

Table of Figures

Table of Tables

1 Introduction

In recent years, giant strides have been made in the field of consumer electronics. These advances have given rise to new levels of convenience and entertainment for consumers. Much of this progress is evident in the automotive industry. Today's vehicles allow people to control an increasing number of functions while driving. Simple four-speed manual transmissions have given rise to clutch-less, automatically shifted six or seven speed automobiles with a variety of modes such as Winter or Sport. Most new vehicles today are equipped with power locks, power steering, and power windows, all of which were a luxury eight to ten years ago. Navigation systems have taken precedence over maps and road atlases. Recently, cars have developed into more than just a method of transportation. Entertainment modules are becoming more common in today's automobiles. Many new vehicles are equipped with rear seat DVD players for the passengers as well as multiple-disc CD changers. The radio is also evolving. Satellite radio is gaining popularity compared to standard AM and FM tuning. These features are just a few examples of some of the newer technologies available to today's drivers, giving them more control over the vehicle as well as various entertainment options.

Another entertainment technology becoming increasingly popular is digital music libraries. MP3 compression allows for efficient storage of music with a virtually unnoticeable loss of sound quality. People are able to carry thousands of songs on devices no larger than a single cassette tape. MP3 technology is proving to be well suited for automotive use. Stereo systems capable of reading both traditional music and MP3 CDs can be purchased enabling users to carry hundreds of songs on a single disc. One of the problems associated with carrying large music libraries in an automotive environment is that the hands-on interface in most automobiles makes it difficult to navigate music collections without inhibiting driving performance. While these interfaces may be acceptable for the radio or a single CD, they become a problem with large quantities of music MP3s. Speech recognition technology could be considered as a possible solution to this problem. Speech recognition would allow users to search through their music libraries without constant hands-on interaction. This technology has already been adopted in certain automotive navigation systems, and could greatly enhance modern car stereos if implemented properly.

During the 2004-2005 academic year, a group of WPI students implemented a voice-controlled MP3 Jukebox for an automotive environment. At the completion of their project, they successfully developed an MP3 player that allowed users to verbally select songs from large music libraries as well as perform basic control functions such as pause, resume, next song, and previous song. Additionally, they provided the user with audio feedback from the system through the use of a text-to-speech generator. One of the limitations of their design was that users must be aware of all songs in their music library and know each song by name in order to select it. This task becomes difficult once music databases grow larger than two or three hundred songs. The project sponsor, Bose Corporation, requested that the development of the MP3 Jukebox be designed in such a way that they could easily embed the product into their existing technologies. To realize this flexibility, Bose asked the programming be written in a platform independent language such as C or Java which would enable the use of the program with different operating systems such as Windows, Unix, or Linux. Last year's group complied with these requests and used Scansoft's (now Nuance) VoCon speech recognition engine along with a Java compatible MP3 decoder to play the music files.

The main goal for our project was to implement new and innovative song selection methods that would allow users to easily navigate their music libraries. We also hoped to implement these new song selection methods to allow for the most user friendly interface without causing unnecessary driver distraction. Our goal was to modify the previous design to allow for more complete and innovative song selection techniques as well as to analyze the amount of distraction various Jukebox features may impose on a driver.

Background research was performed to examine various methods for song selection. A focus group was also conducted to brainstorm song selection techniques that may be advantageous in an automotive environment. After we had implemented new song selection techniques, our focus shifted to measuring the amount of distraction associated with the new features. Driver distraction analysis would help determine which interface is most appropriate in an automotive setting. A testing suite was developed to simulate an automotive environment, and human subjects were recruited to test our design. The results from these tests allowed for analysis of both driver distraction as well as the accuracy of the design. Additionally, different multimodal interfaces were tested to determine which interface was the most efficient and easiest to use.

This document details the methodology of the development of our MP3 Jukebox design. The process begins with the background research which discusses some of the technical aspects of the project such as speech recognition and MP3 compression. It also examines various song selection methods in common, computer-based MP3 players. Next, the document discusses the song selection methods we decided to implement along with the functionality of the design. It then details the testing methodology followed by the test results and analysis of driver distraction and SUI accuracy. The document ends with a conclusion regarding MP3 Jukebox design process along with recommendations for future work on the project.

2 Background

This section describes information that is pertinent to the understanding and scope of our multimodal MP3 Jukebox project. The work of the previous project team is summarized from the design to the testing of a voice-controlled MP3 Jukebox. Additionally, this section provides an overview of driver distraction and commonly accepted methods for its measurement. A background of portable music is presented with a discussion of MP3 players including those that incorporate voice control or are tailored for automotive uses. Methods for selecting songs from a large music library on existing MP3 players are provided. Also, current technologies are summarized for digital media management. Currently, there is a market for an in-car MP3 Jukebox which can be enhanced though both voice and physical control.

2.1 Previous Project Summary

The project completed last year was the initial foray into the development of a voice-controlled MP3 Jukebox by a group of three WPI students. The project's stated goals were threefold. The first was to interface a speech recognition engine with an MP3 player with both a Speech User Interface (SUI) and a Graphical User Interface (GUI). The second was to complete usability testing in order to refine the design of their SUI to make it user friendly. The third was to measure the usability of their SUI design against that of the GUI through both qualitative and quantitative data measurements.

The first step was to divide the project into several steps in order to allow work on several stages at once. The group decided to go with a layered approach because that approach is the most straightforward to work with and also allowed for the concurrent stage work to be performed easiest. In a layered approach, the layers can only interact with the layers that are touching each other.

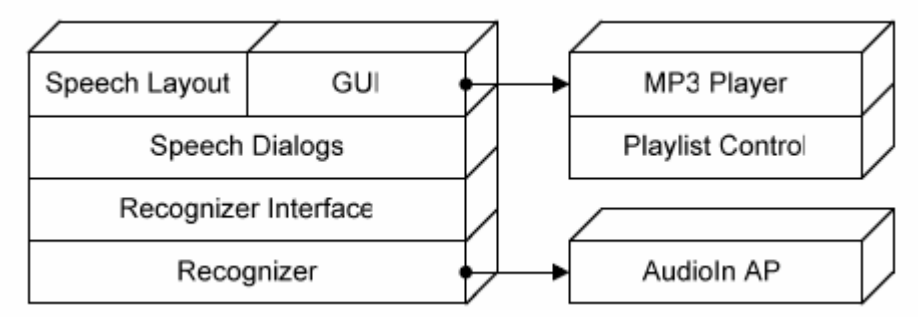


Figure 1: Previous project's software design (Source: Liberman et al.)

As shown in the above Figure 1, the recognizer is the lowest level layer. The recognizer only interacts with the recognizer interface. The recognizer interface interacts with the speech dialogs which contain the various grammar interactions between the user and the computer. Next, the speech layout layer brings all the separate dialogs together to create a single dialog. The speech layout layer interacts with the GUI and the MP3 player.

The group next developed a pair of Speech User Interfaces. Both allowed the user to interact with the MP3 player and select desired songs from the database. Figure 2 and Figure 3 show the song selection methodology for each SUI design.

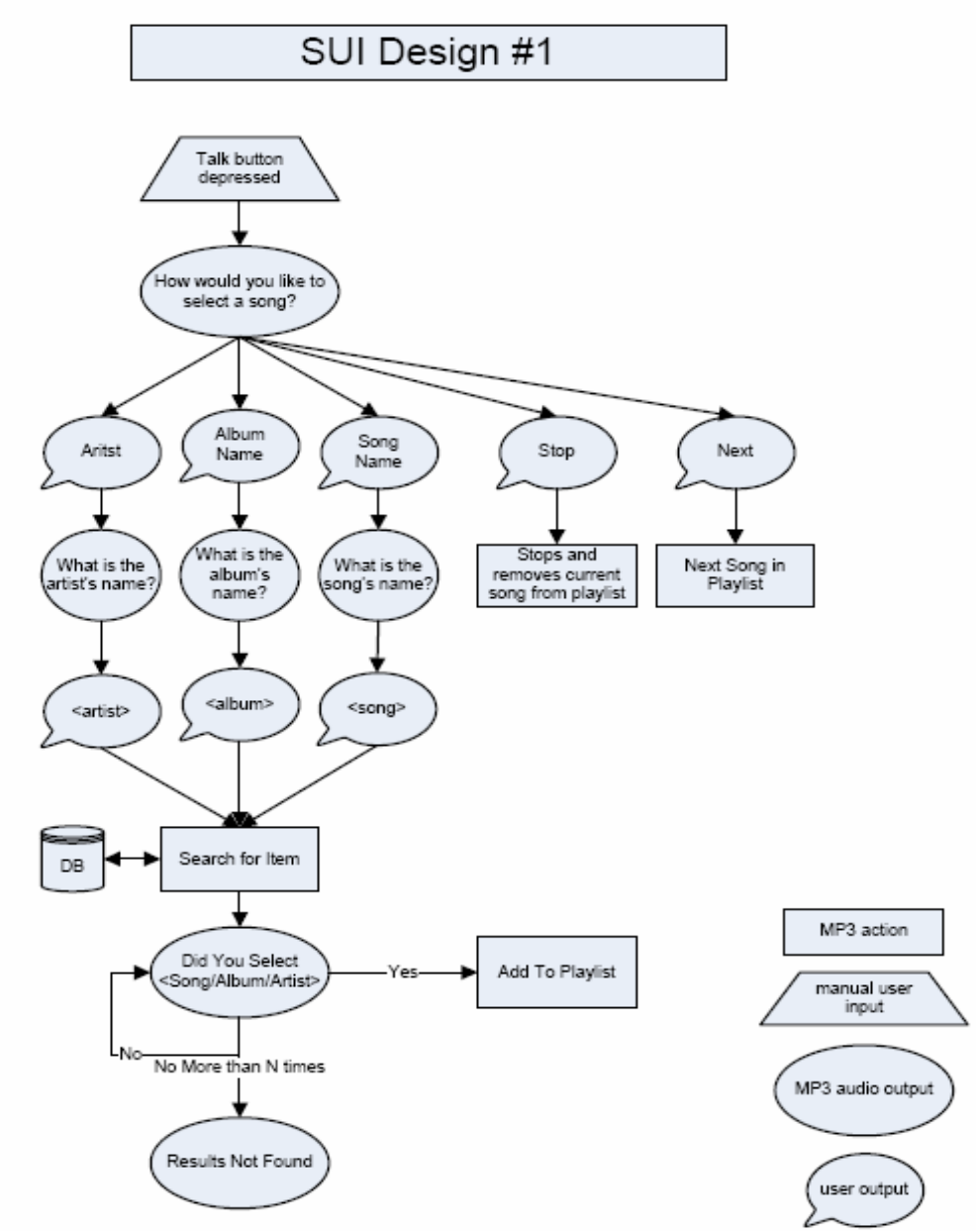


Figure 2: SUI Design 1 (Source: Liberman et al.)

As seen in Figure 2, the first step in using this interface is to depress the Talk button. Next the user can speak a command out of five potential options: artist, album name, song name, stop, or next. If artist, album name, or song name is stated, the computer asks for the artist, album name, or song name respectively. Then the user states the answer to the question and the system searches for the item.

SUI Design 1 and SUI Design 2, shown below in Figure 3, are very similar. The main difference is that there are several more options for playback including previous and pause. The

second design does not have the additional prompt from the text-to–speech component asking, "What is the artist/album/song to be selected?"

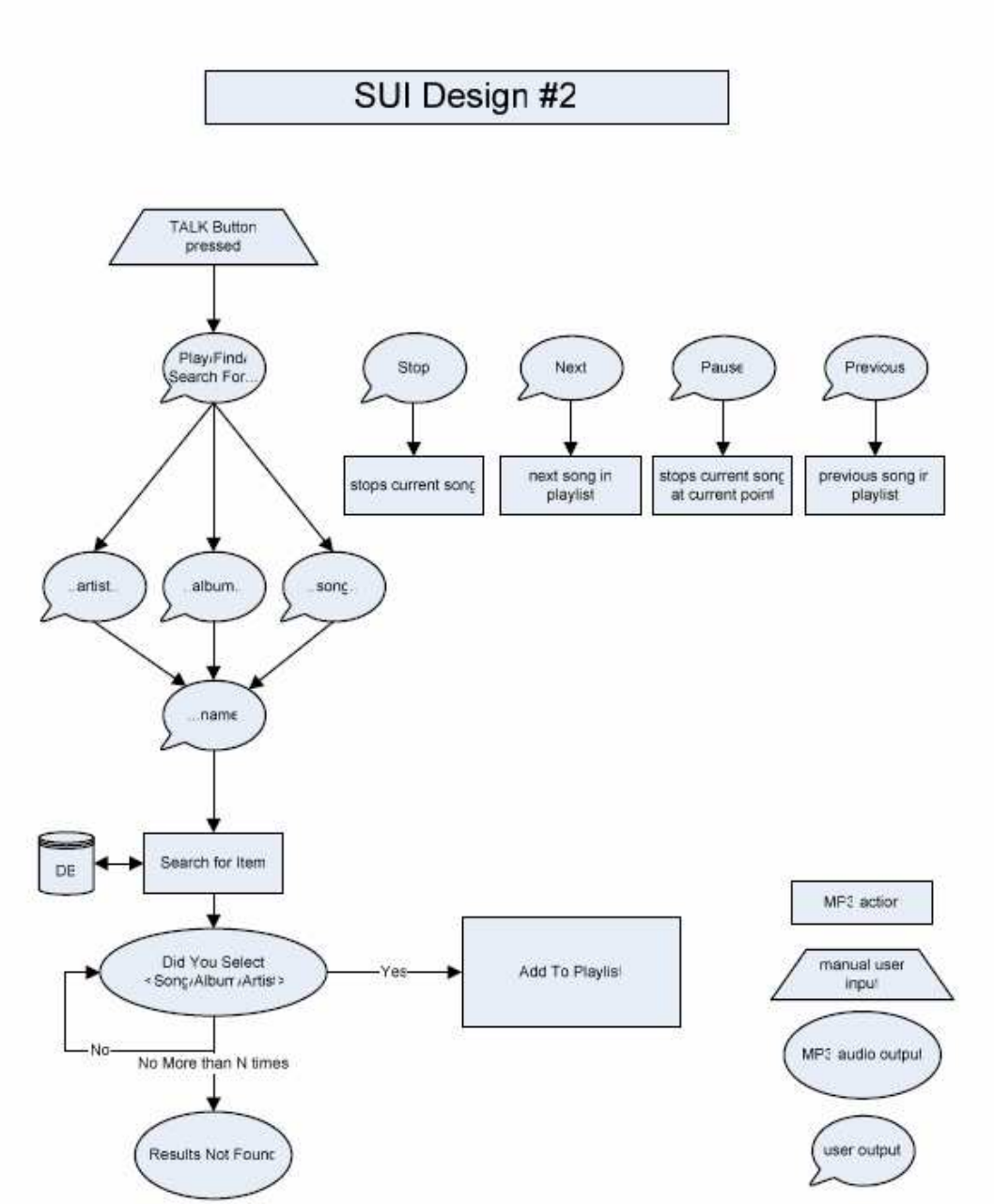


Figure 3: SUI Design 2 (Source: Liberman et al.)

The group had two specific experimental design phases. The first phase was meant to develop a SUI that was most usable. In order to complete the two design phases, they initially developed two different SUIs. After development, they performed user testing to create a third improved design using the best of both original designs. At the conclusion of the design phase,

an improved design had been developed. The purpose of testing was to compare the usability of the final SUI to that of the GUI. The testing used the speed of song selection, error rates, speed of recovery from errors and the distraction level to judge the usability of the SUI versus the GUI. The testing phase used 10 subjects with the following characteristics:

- Native English speaking
- Equal amount of male and female
- Various technical backgrounds
- Age range from 18-30+
- No visual impairments or mouse control difficulties

The subject sat at a table with a display screen in front of them. This display screen would show a simulated driving environment which was created by mounting a digital video camera to the dashboard of a moving automobile and then filming the travel. Experimental design phases were executed using this setup. During the experimentation, the subjects would be asked to perform several tasks which would test the ability to select songs, skip songs, and stop playback. After completion of the experiment each subject was asked to fill out a survey covering the problems and strengths of both designs and any changes that the user would recommend. Experimental Design Phase 1 had three rounds of testing before the SUI met the standards desired by the development team. Experimental Design Phase 2 had fours main parts. During part one, the subjects were asked to select songs using the SUI developed in Design Phase 1. Second, the subjects were asked to select songs using a GUI similar to those currently on the market. The third part measured the distraction level of the SUI using the Peripheral Detection Task, where the subjects are asked to respond to objects appearing on the screen while also using the SUI at the same time. Part four had the same approach as part three with the exception of using the GUI as opposed to the SUI. The results of testing are shown in Figure 4 below.

	Accuracy		Time to perform task (s)		Time to react to distraction (ms)	
Selection	SUI	GUI	SUI	GUI	SUI	GUI
Song	98%	99%	296	264	1020	888
Artist	98%	100%	373	161	955	910
Album	98%	100%	327	130	985	933

Figure 4: Results from the previous project (Source: Liberman et al.)

Accuracy was rated as the number of times the correct song, artist or album was selected. The time to perform the task was measured, in seconds, as the amount of time necessary to change to a specific song, artist or, album. The time to react to distraction was measured in milliseconds and was used to determine the amount of time needed to respond to a distraction during a task.

The group originally expected the SUI to be more accurate, quicker and less distracting then a GUI. As you can see, the results are the opposite of what was expected. The SUI proved to be less accurate, took longer to react, and the tasks took longer to perform. These results may be attributed to the fact that most of the subjects were accustomed to using a GUI as opposed to a SUI. Additionally, the experimental environment did not truly simulate a real driving experience. Because the subjects were not actually driving, they could focus on the PDT targets and use a GUI much better than if the subjects had actually been driving.

After the subjects were surveyed, it was discovered that part of the problem with the SUI was due to the text-to-speech engine used in the project. The text-to-speech engine was very mechanical and difficult to understand which may have lead to confusion for the driver. Also, the subjects thought that some of the speech commands could be executed using a multimodal interface such as buttons on the steering wheel.

After Experimental Design Phase 2, the project group developed the following recommendations for future work:

- Implement a more understandable Text-To-Speech engine
- Utilize multimodal design for better usability
- Develop new methods to classify and select songs
- Implement the embedded version of the speech recognition engine

2.2 Driver Distraction

The primary reason for making a car MP3 Jukebox voice-controlled is to decrease any distraction associated with the operation of the device. For this reason, we had to test the device on human subjects to determine how much distraction is caused by a particular interface. This section discusses driver distraction in detail including common causes of distraction, driving performance metrics, and common driver distraction measurement techniques.

2.2.1 What is Driver Distraction?

Before we can begin to test for driver distraction, we must first define it. The American Automobile Association Foundation for Traffic Safety defines driver distraction as occurring "when a driver is delayed in the recognition of information needed to safely accomplish the driving task because some event, activity, or object within or outside of the vehicle compelled or tended to induce the driver's shifting attention away from the driving task (Hammer, 2003)." As one would assume, the potential for a car accident is much more likely if a driver is distracted by a secondary task. The National Highway Traffic Safety Commission (NHTSC) estimates that driver inattention contributes to approximately 25 percent of police reported crashes (Stutts, 2001).

The NHTSC states that there are four distinct types of driver distraction which are visual, auditory, biomechanical (physical), and cognitive. Visual distraction occurs when a driver neglects to look at the road because they are concentrating on another visual target such as the radio or a billboard. Auditory distraction is when a driver focuses their attention on a sound or auditory signal rather than the road. Some common causes of auditory distraction include listening to the radio, holding a conversation with a passenger, or talking on a mobile phone. Biomechanical or physical distraction occurs when a driver removes one or more hands from the wheel to control a secondary object. For example, changing a CD or scanning the radio are

common forms of physical distraction. Cognitive driver distraction happens when a driver's thoughts absorb their attention to the point that their driving reaction time is inhibited. Conversation with passengers and interaction with in-car navigation system are forms of cognitive distraction. (Hammer, 2003)

It is important to understand that these four forms of driver distraction are not mutually exclusive, meaning more than one form of distraction can impair driving performance at a given point in time. For example, a voice controlled navigation system may cause auditory distraction though its audio prompts, along with cognitive distraction if the driver focuses on instructions from the navigation system more than the task of driving.

2.2.2 Common Causes of Distraction

There are many objects and events that can distract a driver while driving. Most causes for distraction fall in one of two categories; technology based or non-technology based. Technology based distractions include mobile phones, in-car navigation systems, and entertainment systems such as CD and DVD players. The type of distraction each activity causes can vary. For example, a cell phone may cause auditory and cognitive distraction while controlling a CD player can cause visual and physical distraction. Non-technology based distractions include activities such as eating, drinking, smoking and talking to a passenger. Once again, the type of distraction can vary between activities (Hammer, 2003).

It is important to examine some common causes of distraction to determine some inherent flaws in particular in-car devices. We will then be able to avoid reproducing these flaws in our design. The American Automobile Association's (AAA) Foundation for Traffic Safety conducted a study in which they examined detailed crash records from a large database called the Crashworthiness Data System. They looked at records from crashes that occurred between 1995 and 1999. They found that driver distraction caused 8.3% of the crashes examined (Stutts, 2001). The following table identifies the most common sources of driver distraction reported by drivers involved in a crash.

Table 1: Percentage of Drivers Citing Each Distraction Source as Contributing to Crashes (Source: Stutts, 2001)

Distraction Source	% of Drivers
Outside person, object or event	29.4
Adjusting radio, cassette, CD	11.4
Other occupant in vehicle	10.9
Moving object ahead	4.3
Other device brought in vehicle	2.9
Adjusting climate controls	2.8
Eating/Drinking	1.7
Using a mobile phone	1.5
Smoking related	0.9
Other distraction	25.6
Unknown distraction	8.6

It is important to note that this study does not take into account the frequency of each activity, meaning that an activity that has a fairly low level of driver distraction may have accounted for more crashes because it occurred more frequently than an activity with an inherently higher level of driver distraction. For example, conversing with a passenger may be less distracting than interacting with a mobile phone, yet the frequency at which a driver talks to a passenger may occur far greater than the frequency at which a driver uses a mobile phone. This means that the frequency at which an activity is carried out should be considered when measuring driver distraction. Also, because of recent advances in technology, it can be assumed that the frequency of interaction with technology based devices has increased since this study, and will continue to increase as technology continues to advance. Since the completion of this study, additional research has been done regarding the frequency of some common distracting activities, as well as distraction levels associated with some new in-car technologies such as email/Internet and navigation systems. With this information, Hammer et al. (2003) were able to produce a rough ranking of distraction sources based on the number of crashes caused by each source as well as the frequency at which each source was carried out.

The following list shows some of these sources ranked from the most distracting to the least distracting (Hammer, 2003).

- Email/Internet
- Navigation Systems
- Using a mobile phone
- Adjusting radio, cassette, CD
- Adjusting climate controls
- Eating or drinking
- Smoking
- Talking to a passenger

From this list, it appears as if technology based distractions tend to be more distracting to drivers than non-technology based distractions. Because our MP3 Jukebox is a technology based design, we must avoid some of the inherent causes of distraction associated with technological devices.

2.2.3 Driving Performance Variables

When researching the level of driver distraction associated with a particular device, it is necessary to explore what effects the device has on different driving performance measures. For example, it may be important to find out if a secondary task that causes visual distraction inhibits a driver from maintaining a constant speed or staying in the middle of the lane or both. This section will discuss the effects that various distraction types and distraction sources have on different driving performance metrics. The metrics that will be examined include lateral car position, car speed maintenance, driver reaction times, and gap acceptance.

Lateral position refers to the position of the vehicle on the road in relation to the center of the lane where it should be driven. Studies have found that visually and physically distracting tasks severely affect a driver's ability to maintain their lateral position on the road. Some tasks that affect lateral position the most include dialing a mobile phone, manually entering information into a navigation system and changing a CD (Hammer, 2003). Studies have also shown that the use of voice inputs rather than manual inputs improve a driver's ability to maintain lateral position. For example, this performance measure has been shown to improve if voice-controlled CD players or navigation systems are used over manually controlled systems (Gartner, 2002).

As opposed to lateral position, speed maintenance is affected mainly by cognitive and auditory distractions. Drivers showed large variations in driving speeds when tested using both a

hands-free and a hand-held mobile phone. In particular, drivers tended to drive slower when using a mobile phone, than with no distractions (Burns, 2002). Researchers produced similar results when testing drivers with both manual and voice controlled navigation systems (Gartner, 2002). The reasons for people driving at lower speeds are still not clear. Some believe drivers paid less attention to maintaining their speed while other felt that they engaged in compensatory behavior while distracted in order to reduce their crash risk (Hammer, 2003).

Drivers' reaction times to external events or objects are adversely affected by all kinds of driver distractions. Studies have shown that many common sources of driver distraction increase reaction time in some way. Researchers have found that using either a hands-free or hand-held mobile phone can increase reactions to common road events such as changing traffic lights or a breaking lead vehicle by up to 30 percent (Burns, 2002). If a driver is not concentrating exclusively on the driving task, their reaction times can be expected to increase.

The last driver performance measure to be examined is gap acceptance. Gap acceptance is defined as the normal distance a driver is willing to travel behind another vehicle. This performance metric has been found to be affected by both cognitive and visual distractions. If visually or cognitively distracted, drivers tend to accept shorter gaps in traffic than if not distracted. Additionally, distraction also causes drivers to disregard weather conditions and the surface of the road while determining acceptable traveling gaps (Hammer, 2003).

2.2.4 Measuring Driver Distraction

To determine the level of distraction associated with our design, an accurate method of measuring driver distraction must be employed. This section will examine some established methods used to determine how much driver distraction is caused by a secondary task.

Driving Simulators

Driving simulators are a common tool used in measuring driver distraction. The main advantage of using a driving simulator is that a number of driving performance metrics can be obtained in a relatively safe environment. In most studies that measure driver distraction as a result of a secondary task, it is assumed that attention devoted to the secondary task may be detrimental to the participant's driving performance. If a driving simulator is used, there is little threat to both the driver and the test administrator if an accident were to occur. Driving

simulators can also create scenarios involving multiple vehicles that could be dangerous to recreate in real life (Green, 1999).

Along with increased safety, greater experimental control is available if using a simulator. During tests, it is possible to select the type and difficulty of the driving tasks to be performed; however this is not always the case during on-road tests due to variables such as weather and other drivers. It is also easier to gather driving performance metrics than in on-road tests. Statistics such as speed control and lateral position are easily attainable in a driving simulator. Lastly, a number of different test conditions can be applied with relative ease. Conditions such as time of day, weather, road environments and terrain can be controlled using a simulator, while these conditions would be difficult and dangerous to recreate in real life (Green, 1999).

The main disadvantage to using a driving simulator is that the characteristics of simulators can vary significantly which can affect the realism of the test and the results obtained. Simulators that offer a realistic driving environment are said to be high-fidelity. These types of simulators are normally equipped with a realistic visual scene, complete with roadside objects such as signposts and trees. On the other hand, low-fidelity simulators contain less realistic driving environments with no more than major markings such as lane lines. The cost of simulators can vary depending on the realism that they provide. High-fidelity simulators are generally expensive, and difficult to find on the commercial market. Low-fidelity simulators are more affordable, but perhaps at the cost of realistic test results. One last disadvantage of driving simulators is the driver's perception of the simulator. Test drivers tend to act differently in a driving simulator than they would in a real car, because there are no serious consequences that result from driving errors in a simulator (Green, 1999).

On-Road and Test Track Studies

On-road and test track studies are the most realistic method of measuring driver distraction from in-vehicle technologies. In on-road studies, drivers are asked to drive an actual vehicle while interacting with in-vehicle devices. The drivers will drive for a specified period of time while data loggers record driving performance metrics. These metrics are then compared to a baseline test (no interaction with devices) for a determination of the distraction caused by the

in-vehicle device. Data such as speed control and lane excursions tend to be the most telling measurements in determining driver distraction.

Despite the realism of these procedures, it is not always possible to conduct on-road driving tests. One inherent problem in these types of tests is the danger posed to both the driver and the test administrators. If a driver becomes too distracted by the secondary task, there is a legitimate possibility that an accident may occur which could result in serious injury. For these reasons, drivers may be reluctant to participate in these types of tests and test administrators may have difficulty getting these tests approved (Green, 1999). Also, these tests can be very expensive to conduct. Obviously, a vehicle and a driver are necessary in these studies, as well as a closed test track in some cases. For these reasons, on-road studies are rarely used in determining driver distraction (Hammer, 2003).

Peripheral Detection Task (PDT)

The Peripheral Detection Task (PDT) is a recently developed method for measuring driving distraction. During the PDT, test participants are asked to perform a series of tasks while responding to visual targets presented in their periphery. Targets may include lights or colored dots in their field of vision, and they must respond to these targets when they appear by pressing a button. The basis for the PDT is that as test participants devote more of their attention to the primary task, they will respond slower and fail to detect more PDT targets. Therefore, a measure of the level of distraction caused by the primary task can be determined by performance during the PDT (Hammer, 2003).

The validity of the Peripheral Detection Task has been examined on a number of occasions. In 1999, a study was conducted using the PDT in conjunction with a driving simulator (Martens, 1999). Test participants were asked to navigate a virtual car through simulated roads while performing various tasks with an in-car driver support system. They were also asked to respond to a red square presented on the simulator screen at random intervals. Three different scenarios were carried out including the driver receiving tactile warnings from the support system, the driver receiving auditory warnings from the system, and no interaction with the support system. Reaction times to and detection rates of the red square were recorded throughout the test. Long reaction times and failure to detect the red light were interpreted as high distraction to the driver due to an increased workload from interaction with the support system.

The results showed that as the complexity of the driving task increased, such as a breaking lead vehicle, reaction times and failures to detect the signal increased. These measures also increased when the driver received auditory prompts, yet did not increase when receiving tactile warnings. From these results, the test administrators determined that the PDT is a valid and sensitive test for measuring driver distraction and driver workload during interaction with a driver support system (Hammer, 2003).

A year later, another study of the validity of the PDT was conducted, this time in a real traffic environment. Thirteen participants were asked to carry out tasks including changing a CD, tuning the radio, and counting backwards while driving along motorways and country roads. While performing these tasks, they had to respond to a small red light randomly presented in their peripheral vision on the windshield by pressing a button. Reaction times and failure rates were recorded and compared to a baseline performance. The PDT results revealed significant differences between the three tasks. Mean reaction time was slowest while counting backwards, yet failure rates were highest while changing a CD (Burns, 2000). From the results, the authors concluded that the PDT is a sensitive measure of visual and cognitive distraction caused by in-vehicle tasks.

From these tests on the validity of the PDT, it is evident that this method fairly accurately measures both visual and cognitive distraction from in-vehicle technologies. Also, implementation of the Peripheral Detection Task is cheap and fairly simple, and it works regardless of whether a driving simulator or a real car is used during the testing.

Eye Glance Studies

One of the first methods developed for measuring driver distraction is the eye glance technique. The eye glance technique is intended to be used in conjunction with a driving simulator or in an on-road study. This method measures the visual behavior of the driver by recording the frequency and duration of eye glances at particular objects in their field of vision. While drivers perform secondary tasks in a car, they normally complete the task through a series of brief glances, normally one to two seconds in duration. By measuring the frequency and duration of these glances, a total “eyes off road time” can be calculated, giving a quantitative measure of how much visual distraction the was caused by the secondary task. The total “eyes

off road time" has been found to be closely correlated to the number of lane excursions committed during secondary task performance (Hammer, 2003).

To obtain the glance frequency and duration metrics, researchers have traditionally video recorded the driving test, and reviewed the tape frame-by-frame for the results. In recent years, sophisticated head and eye tracking devices have been developed to automatically produce the metrics. Additional information such as eye-closures and over-the-shoulder head turns can be determined by these devices. Although eye tracking equipment can be very helpful in gathering eye glance statistics, it is fairly expensive and difficult to find on the commercial market (Hammer, 2003).

Visual Occlusion Technique

The Visual Occlusion Technique is an inexpensive alternative for measuring driver distraction. Unlike the other two methods described in this section, visual occlusion tests are completed independent of a driving simulator or real car. This method is based on the assumption that drivers need to observe the road only part of the time, and the rest of the time is available for other purposes, such as interacting with in-car devices (Hammer, 2003). During tests involving the Visual Occlusion Technique, the driver's vision is partially or completely impaired at various time intervals. The time that the participant's vision is blocked simulates the time they are looking at the road, while the open phase simulates the time they are able to look at the secondary task. Using this method, one can determine if an in-vehicle task can be performed through a series of short glances, and if it can be easily resumed after interruption. If a task can be executed using only short, periodic glances, it is said to be "chunkable." The assumption is that a task that can be chunked is acceptable to be carried out while driving, while tasks that cannot be chunked are deemed unacceptable (Fagerstrom, 2001).

In 2003, a study was conducted to examine the validity of the Visual Occlusion Technique (Baums, 2003). The study was a part of the ADAM (Advanced Driver Attention Metrics) project, which is a joint effort by Daimler Chysler and BMW to gain a better understanding of attention demands on drivers while using in-vehicle technologies. In this study, the researchers had twenty-four participants complete various in-car tasks under both a visual occlusion condition, and under a PDT condition. Some of the test participants were asked to complete common driving tasks including: talking on the phone, unwrapping a candy, changing

a cassette, and adjusting the radio volume. The tests produced significant correlations under both the visual occlusion condition and under the PDT condition for which tasks were the most visually demanding. The test administrators concluded that the Visual Occlusion Technique is a valid method for measuring the visual demands of an in-vehicle task (Baumann, 2003). However, they made no determination on the method's ability to measure driver distraction as a result of a task's cognitive demands.

2.3 Portable Music

Portable music has played a pivotal role in the development of musical technology over the past thirty years. The current explosion of portable music is due to the emergence of digital music. Within this section we review a history of portable music from the first "Boom-box" to the current players used today, highlighting the rate at which changes occur. We also discuss digital music formats, current products, and provide a market forecast.

2.3.1 Portable Music History

Portable music was first developed in 1967 by a pair of priests and was called the Illiatoni, which was a combination of the priests' names. The Illiatoni was unable to sell well and the idea was shelved until 1976 when the Rodells, a married couple from Kansas, purchased the rights to the Illiatoni. The Rodell's improved the original Illiatoni and added the capability to play 8 tracks and later cassettes. As the sale of the Illiatoni took off, RCA bought the rights and increased production even more. By 1979, Sony had taken the idea of the "Boom-box," as the Rodelliatoni was now called, and began to reduce the size to a small hand-held player. Within two months of release the early estimates of 5,000 units sold per month were shattered just in Japan.

In 1984, the first portable CD player was released in Japan by Sony and was soon brought to the United States. The original sale price of the D-50, which later became known as the Discman, was approximately half of the production price. Within a year and a half the technology had become developed enough for the D-50 to become profitable (Sony).

The first portable digital audio player was released in the American market in 1998. The Eiger Labs MPFMan F10 was a very basic player with 32MB of memory. The MPFMan F10 could play either WAV or MP3 files. The second player released was the Rio PMP300 from

Diamond Multimedia. This player truly redefined the genre; its sales took off during the Christmas season of 1998 leading to a lawsuit by the Recording Industry Association of America which claimed that the digital devices helped encourage the use of illegal music. The organization supposed that if people could play MP3s downloaded from the Internet, then CD sales would drop. In Sony Corp. v. Universal City Studios, digital audio players were ruled to be legal devices (Wikipedia, Discman).

At the end of 1999, a company called Remote Solutions developed the first player with an actual hard drive as opposed to Flash memory. Players with a hard drive provide more space per dollar, giving rise to the term "MP3 Jukebox." The PJB-100 utilized a laptop hard drive with a storage capacity of 4.8 gigabytes, which was enough space to store approximately 1200 songs in the MP3 format. This is significantly larger than the Flash devices that could only hold around 20 songs. In 2001 with the introduction of the Apple iPod, the real portable music boom began. In 2004 the iPod made up approximately 90% of all hard-drive based MP3 player sales and 70% of all digital audio players sold (Wikipedia, iPod, 2005).

2.3.2 Digital Music Formats

An audio file stored on a computer is made up of three parts: the wrapper, header, and audio data.

Figure 5: Representation of audio file (Source: Liberman et al.)

The header is used to save information about the data, namely the sampling rate, resolution, and type of compression. The audio data is a stored variation of the actual music of the song. The wrapper is sometimes added to include information such as licensing or streaming capability. It can also include the ID3 tag which contains information such as the artist and song title. (Fries, 2005)

The format of a digital audio file refers to how the information is structured within the file. The same format can be used by different song files without a problem. There are five common formats; PCM, DPCM, ADPCM, μ-law compression, and MPEG Audio.

PCM, or Pulse Code Modulation, is a generic form of storing and can be read by most audio applications. PCM is represented by binary digits, 1 is the presence of a voltage pulse, and 0 is the absence of a pulse. DPCM, or Differential Pulse Code Modulation, stores the differential between samples. DPCM uses 4 bits to store the difference no matter the resolution of the original file. ADPCM, or Adaptive Differential Pulse Code Modulation, is similar to DPCM except the number of bits used to store the difference is varied depending on the complexity of the system. ADPCM works by predicting the value of the next sample and then storing two numbers, the actual value and the predicted number. The μ -Law type Compression is similar to ADPCM although it is less accurate in its predictions. MPEG Audio is a family of open standards used for compressed audio including MP2, MP3, and AAC (Fries, 2005).

2.3.3 Current Products

As of September 2005, there were currently no products developed that were designed to play and store MP3s within a car environment. There are three portable MP3 players with voice recognition technology built-in. Two are produced by Olympus, the DM 10 and DM 20; the other is the "FID iROCK!" These three products have basic voice commands which include play, fast forward, rewind, pause and stop functions (CNet, 2005).

Prior to 2004, there was one hard drive/flash memory based car MP3 player on the market which was developed by the British company Empeg. The Empeg Car was a Linux based unit that transferred MP3 tracks from the user's computer using Ethernet, USB or a serial port connection. Prices ran from $1100 for a 4 GB to $2400 for a 28 GB player. SONICblue, the company that had earlier bought the Rio brand purchased Empeg in November of 2000 and the unit was renamed the Rio Car. The line was discontinued on September 24, 2001 after only selling 6000 units, mostly in Europe (Wikipedia, Empeg Car, 2005). The main problems according to consumers were fourfold. First, the user interface was awkward to use during driving. Secondly, the cost was prohibitive for the size, especially for the smallest player. Third, the Rio Car was released before the prime of the MP3 had come about. Finally, advertising for such a breakthrough idea was extremely limited and almost completely nonexistent in the United

States. While the idea was promising, these factors led to the small sales of the Rio Car (Wikipedia, Empeg Car, 2005).

Within the car market today there are several new hard drive based MP3 Jukeboxes, however none of these Jukeboxes are voice-controlled. As the market continues to evolve and grow there will eventually be a player with voice control. Until this Jukebox is realized there will be a large void in the market.

The first, the NeoCar, features a car bay and a remote control that are installed into the automobile. Then you take the "caddy," which is in essence a hard drive that can be plugged into the car bay, and load it with all of your songs via a cable that connects the "caddy" to a USB port on a computer. The caddy is then placed into the car bay allowing MP3s to be played on the road. Song navigation is done through the remote. This remote allows you to shuffle through the various folders containing stored music. Whole folders can also be played instead of picking an individual song. The NeoCar also has a random play method. A new feature of the NeoCar is a Text-To-Speech engine that can be installed and can read the contents of a folder to the user. However, there is no option to include a speech recognizer with the NeoCar. The NeoCar is current priced at approximately $500 for an 80 GB system.

The second car MP3 Jukebox released is called the PhatBox by a relatively new company called PhatNoise. The PhatBox's use is somewhat limited because it can only be installed in the following automobile models: Acura, BMW, Ferrari, Ford, Honda, Lincoln, Mercury, Mini, Porsche, Range Rover, Scion, and Toyota. Even within those car manufacturers, the PhatBox is only designed for certain models.

The PhatBox operates in almost the same way as the NeoCar where there is an installed docking port which contains the software to use the MP3s loaded onto a 40GB portable hard drive. The PhatBox automatically organizes the MP3s using their data tags into categories by artist, album, playlist and genre. The PhatBox then uses a pre-installed multiple CD changer faceplate as the basis for song selection. Each of the six CD selection buttons is changed into a method to select any of the categories. The selection methods available are track playlist, album, artist, genre, and options. Once a category is selected, a voice prompt reads through all of the possible selections. It is also possible to change within each category. For example, if the user is listening to the Beatles and wants more rock, he or she would press the genre button. If the user

wanted to just listen to that specific album, he or she would press the album button. With a 40 GB hard drive, the PhatBox costs $800.

As these two products exemplify, there is a growing variety of car-based MP3 Jukeboxes coming onto the market. There are several more expected to be released within the next year or two. The success of these products could determine the future of MP3 Jukeboxes in cars.

2.3.4 Market Forecast

As we work on the development of an MP3 player with voice control, it is extremely important to understand the market into which the product will be released. In 2004 and the first quarter of 2005, digital music sales accounted for approximately 2% of the music industry's sales world wide. However projections show that the percentage should grow to 10% by 2008. The percentage is even larger in the United States with projections showing that the online sales of music will account for 16% of American sales by 2008 (OECD). The portable MP3 player has had even a larger growth, with sales more then doubling over the course of 2004 to approximately 6.9 million units. The growth was most explosive in the fourth quarter of 2004, with a 500% sales increase in iPods from the fourth quarter of 2003 (iPod, 2005). The trend continued into the first quarter of 2005 with sales increasing 18% from the fourth quarter in 2004 to the first quarter in 2005 with sales continuing to increase through the third quarter of 2005. In Europe from 2002 to 2003, the MP3 player went from 3.5% of portable audio player sales to 13.7% which is a growth of almost 400% in one year (OECD). The boom is expected to continue through 2009 with sales of approximately 104 million units (In-Stat).

Voice recognition technology in automotive applications is rising quickly as well. In 2004, about 6 million items with voice recognition were sold. Sales are expected to nearly double every year until 2009. Telematics should be in approximately 43% of all cars by 2009 (ScanSoft, 2005). This projected increase in both voice recognition technology sales along with the current explosion in portable MP3 player sales shows that the voice-activated MP3 player has a huge potential market within the United States and around the world as well.

2.4 Digital Music Media Technologies by Gracenote

As MP3s and other forms of compressed music have evolved into a permanent fixture of the music industry, many companies have invested efforts in releasing products into this wide market. The personal storage of songs has become ubiquitous with music itself; consumers can easily amass a library of thousands of songs, all connected on a single storage system. By having song data instantly accessible on a device, software can be written to better manage music libraries and give consumers control unthinkable in the CD or physical medium era.

One company that seeks to improve user-music interaction is Gracenote. "Gracenote provides its award-winning media management technology and global media database of digital entertainment information to the mobile, auto, portable, home and PC markets" (Gracenote, 2005). Gracenote's strategy is to provide other developers with innovative music technologies. The developers then integrate these features into their products and provide them to the end-user. For this reason the company is not widely known to the consumer; however their products are already incorporated into virtually every piece of music software. Their ultimate market strategy appears to be the seamless integration of digital entertainment with everyday life.

MusicID™ and Mobile MusicID™

Gracenote's most common product is MusicID™ which has become the industry standard for music identification. At the backbone lies a database of over 48 million songs as of 2006. MusicID™ stems from Gracenote's proven CD recognition technology known as CDDB. CDDB analyzes the table of contents of a music CD and matches its unique "fingerprint" through a patented process to an online database. Because the music CD format cannot provide metadata such as artist and song title information, a database lookup is necessary. From these beginnings, their media recognition technology has evolved to recognize digital music files through a variety of methods to provide superior accuracy. In addition to the traditional metadata included in a digital media file's tags, MusicID™ uses external information to identify songs, such as the file name, the directory path, and related songs in the music collection. More innovatively, by analyzing a song's audio waveform fingerprint, MusicID™ can match the pattern to a database of waveforms online. As one would expect, MusicID™ also allows users to retag the embedded information with the complete and accurate results obtained through

recognition (Gracenote, 2005). An associated implementation of MusicID™ known as Mobile MusicID™ provides waveform song recognition over a phone device, allowing users to identify a song anywhere by hearing it. Gracenote describes how the process, which can take as little as ten seconds, works:

1. When someone hears a song he wants to identify, he taps a command on the phone keypad to start the audio recognition process, and then holds the phone up to the music source.
2. The phone captures a few seconds of the audio and extracts a waveform fingerprint of the snippet. The snippet can be from any section of the song, even the last few seconds.
3. The fingerprint is sent to the Mobile MusicID™ recognition service from the service provider that may be located anywhere in the world.
4. The Mobile MusicID™ recognition server compares the fingerprint to its database of reference fingerprints and responds with the exact match.
5. The artist, song title and related information, as well as content like album art and download links are relayed to the user (Gracenote, 2005).

Playlist Plus™

Gracenote's Playlist Plus™ allows intelligent playlist creation once songs have been identified. "Playlist Plus™ analyzes text data available in file tags and filenames to link the music to a sophisticated internal database of relational music information, enabling advanced playlisting" (Gracenote, 2005). One the song-matching is done, the actual playlist creation is done primarily by using a genre system that classifies songs according to 1,600 unique, yet related, genre categories. Playlist Plus™ also provides a feature known as "More Like This™" which instantly generates a playlist of related songs from the user's music library. This feature is especially useful on devices with small screens and limited controls such as car audio players, MP3 players, or cell phones where manual playlist creation would be difficult or unsafe. The most important and innovative feature of Playlist Plus™ is its ability to adapt to an individual user. As new songs are added to a library, the playlists can be dynamically updated. Playlists are intelligently created by taking a user's rating of a song, album, or artist as well as listening

activity into account. To complete this product, users can manually classify songs and save generated playlists.

MediaVOCS™

One of Gracenote's most recent products is known as MediaVOCS™, allowing hands-free interaction with a music collection through speech. The product's official press release occurred on January 5th, 2006. When released by the industry standard in media management, it clearly shows the need for such a product and its importance in the marketplace. Gracenote produced this software to provide a simple way for users to listen to their music. This is especially useful as music libraries become so large that navigation through traditional graphical approaches becomes tedious and difficult or where graphical navigation would be impossible, such as while driving or exercising. It seeks to provide a speech interface capable of recognizing various user commands and speech feedback to the user.

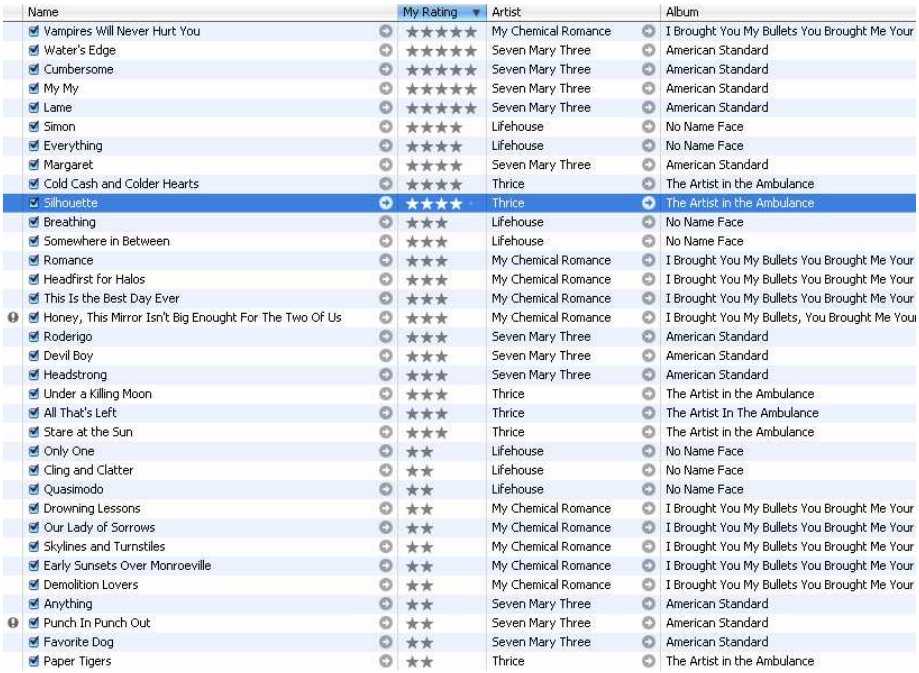
2.5 Common Song Selection Methods

A major component of this project involves implementing new and creative ways to select songs from large music databases. We reviewed some current MP3 players on the market and examined the different ways they enable users to select songs. We looked at three different categories of MP3 players including PC MP3 players, home entertainment MP3 players, and current voice controlled MP3 players.

2.5.1 PC MP3 Players

We found that MP3 players for computers offered the most versatile methods for categorizing and selecting songs. The three MP3 players that we analyzed were Windows Media Player 10, iTunes, and Winamp. All three media players had similar features when it came to storing and playing songs. In all three media players, the user can choose which columns to show in the main music library. The user can then select a specific column to sort the music according to that column, making it easy to navigate through the database and select a song. For example, if he or she wanted to see all the songs in the library released in 1997, the "Date Released" column could be clicked to view the matching songs. A playcount is automatically kept to track the number of times each song has been played. Users can sort songs according to their playcount to

view which songs they play the most. The MP3 players also allowed users to rate songs, normally on a one to five scale, and sort songs that way as well. Figure 7 is a screen shot taken from iTunes which shows an example of how a user can select a song based on the rating given.



Name	My Rating	Artist	Album
Vampires Will Never Hurt You	★★★★★	My Chemical Romance	I Brought You My Bullets You Brought Me Your
Water's Edge	★★★★★	Seven Mary Three	American Standard
Cumbersome	★★★★★	Seven Mary Three	American Standard
My My	★★★★★	Seven Mary Three	American Standard
Lame	★★★★★	Seven Mary Three	American Standard
Simon	★★★★	Lifehouse	No Name Face
Everything	★★★★	Lifehouse	No Name Face
Margaret	★★★★	Seven Mary Three	American Standard
Cold Cash and Colder Hearts	★★★★	Thrice	The Artist in the Ambulance
Silhouette	★★★★	Thrice	The Artist in the Ambulance
Breathing	★★★	Lifehouse	No Name Face
Somewhere in Between	★★★	Lifehouse	No Name Face
Romance	★★★	My Chemical Romance	I Brought You My Bullets You Brought Me Your
Headfirst for Halos	★★★	My Chemical Romance	I Brought You My Bullets You Brought Me Your
This Is the Best Day Ever	★★★	My Chemical Romance	I Brought You My Bullets You Brought Me Your
Honey, This Mirror Isn't Big Enought For The Two Of Us	★★★	My Chemical Romance	I Brought You My Bullets, You Brought Me You
Roderigo	★★★	Seven Mary Three	American Standard
Devil Boy	★★★	Seven Mary Three	American Standard
Headstrong	★★★	Seven Mary Three	American Standard
Under a Killing Moon	★★★	Thrice	The Artist in the Ambulance
All That's Left	★★★	Thrice	The Artist In The Ambulance
Stare at the Sun	★★★	Thrice	The Artist in the Ambulance
Only One	★★	Lifehouse	No Name Face
Cling and Clatter	★★	Lifehouse	No Name Face
Quasimodo	★★	Lifehouse	No Name Face
Drowning Lessons	★★	My Chemical Romance	I Brought You My Bullets You Brought Me Your
Our Lady of Sorrows	★★	My Chemical Romance	I Brought You My Bullets You Brought Me Your
Skylines and Turnstiles	★★	My Chemical Romance	I Brought You My Bullets You Brought Me Your
Early Sunsets Over Monroeville	★★	My Chemical Romance	I Brought You My Bullets You Brought Me Your
Demolition Lovers	★★	My Chemical Romance	I Brought You My Bullets You Brought Me Your
Anything	★★	Seven Mary Three	American Standard
Punch In Punch Out	★★	Seven Mary Three	American Standard
Favorite Dog	★★	Seven Mary Three	American Standard
Paper Tigers	★★	Thrice	The Artist in the Ambulance

Figure 6: iTunes music library sorted by user rating (Source: Apple iTunes)

The following list shows all of the options available in the three MP3 players for which music databases can be sorted. We will consider these valid song selection methods.

- Song Title
- Artist
- Album
- Date Added
- Length
- User Rating
- Track Number on CD
- File Type
- Bit Rate
- Size
- Number of Times Played
- Conductor
- Composer
- Year Released
- Language
- Beats Per Minute
- Sample Rate

Along with these various methods of sorting and selecting songs, the MP3 players offer a feature called auto-playlists. This feature automatically produces a playlist with a pre-determined set of criteria. For example, Windows Media Player 10 can produce a playlist of songs a user normally listens to at night or that was rated with 4 or 5 stars. The user can specify how many songs to play along with other criteria to define the auto-playlist. Figure 8 is a screen shot of Windows Media Player 10 showing its list of auto-playlists.

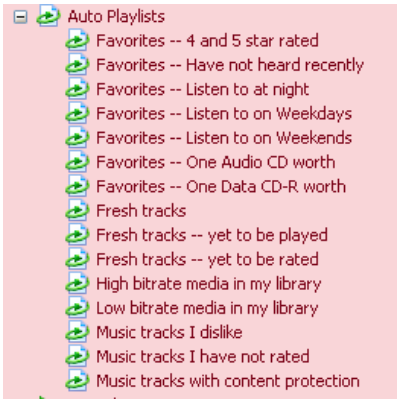


Figure 7: Windows Media Player 10 auto-playlists (Source: Microsoft Windows Media Player 10)

The last notable feature of these PC MP3 Players is their search feature. Using this function, a user can type in a specific keyword to search through the database. The MP3 Player will then return all files that contain the keyword in its song name, artist name, or album title. The following figure shows a screen shot of Windows Media Player 10 where a user searched for the word “beautiful.”

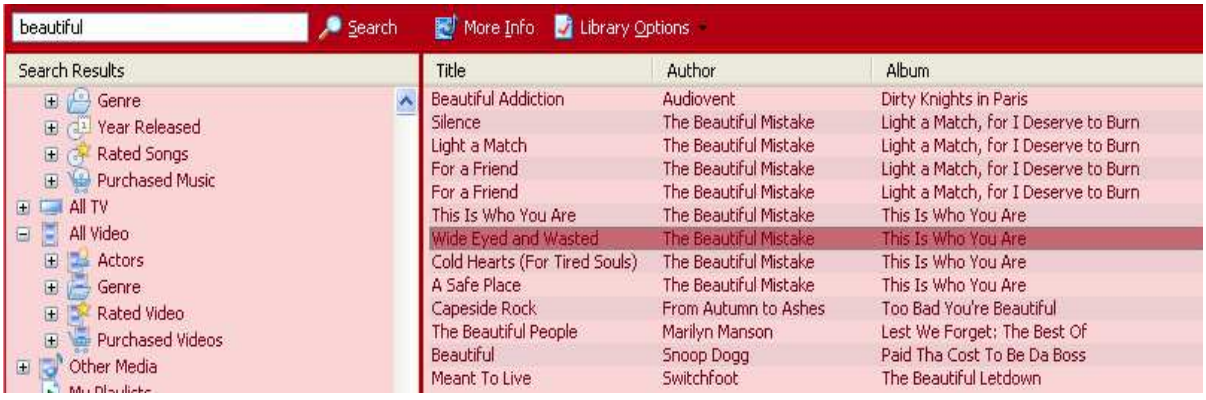


Figure 8: Windows Media Player 10 search for keyword "beautiful" (Source: Microsoft Windows Media Player 10)

While these PC supported MP3 players offer a number of methods for sorting and selecting songs, it should be noted that the concern of user distraction was not considered while implementing these features.

2.5.2 Home Entertainment MP3 Players: Bose uMusic

The Bose Corporation recently released a product called uMusic which is an MP3 player intended for home entertainment. The uMusic system has a hard drive that can store up to 350 CDs for playback. The main feature of the uMusic system is that it learns the user's listening habits as the user cycles through their music collection. It keeps track of which songs are commonly played and which songs are commonly skipped over. Users can also assign songs either a plus or minus rating using a remote control. The uMusic system then uses this information to create automated playlists suited to the user's listening preferences. The uMusic system can be used to simulate the diversity of a radio station, playing only the songs the user wishes to hear. Similar functions can be carried out on PC supported MP3 players as previously mentioned, but the uMusic system is intended for people who do not want to connect their computer to their stereo.

2.5.3 Voice Controlled MP3 Players

Current MP3 players with voice control offer much less diversity regarding song selection methods than other MP3 players. All of the voice-controlled players we examined had fairly simple methods for selecting songs. For the most part, users could only select songs by song name, artist, or album title. Most of the MP3 players also implemented shuffle and repeat commands once a playlist had been selected. None of the voice-controlled MP3 players we looked at contained any song selection features that were not included in the PC or home entertainment players.

One voice controlled MP3 player for the PC had an interesting browsing function that could make selecting songs much easier. The VoiceMP3 by BCL Technologies implemented a command called "browse," where the player will read all of the songs in the current playlist out loud. The user can then play a particular song once it is read to them. This feature could be important in an automotive environment because users may not know exactly what songs are on

an album or in a playlist. Instead of scanning through song by song, they could quickly find out what is in the playlist using a command similar to this browse function.

2.6 Multimodal Interfaces

Currently, the most prominent methods of peripheral control in automobiles are knobs and buttons. However, only recently have automotive manufacturers begun to capitalize on the space available toward the center of the steering wheel. Providing the functions of traditional dashboard controls on the steering wheel allows motorists to keep their hands on the wheel. However there is a limit on the number of buttons that can be put on a wheel, due to concerns with space and confusing the driver.

Given the rise in computing technology over the past decade, it is possible to integrate speech recognition technology into the automobiles themselves. Automotive manufacturers are eager to adopt this technology, with some already using it for climate control and cellular phones. Using both manual input and voice commands provides the user with far greater functionality while minimizing driver distraction.

2.6.1 Manual Input

Car makers have always been obsessed with adding new features that allow greater control over the environment. In most cases more features means more controls, turning the automotive cabin into a warehouse of buttons and knobs. In an effort to provide motorists with more convenience and safety, manufacturers have placed the most commonly used features on the steering wheel itself.

While older cars may have only a few buttons on the steering wheel, current models can offer ten or more, controlling car features from window defrosting to navigation systems. Toyota's current Prius hybrid sedan offers 15 different buttons, some of which can perform multiple features.

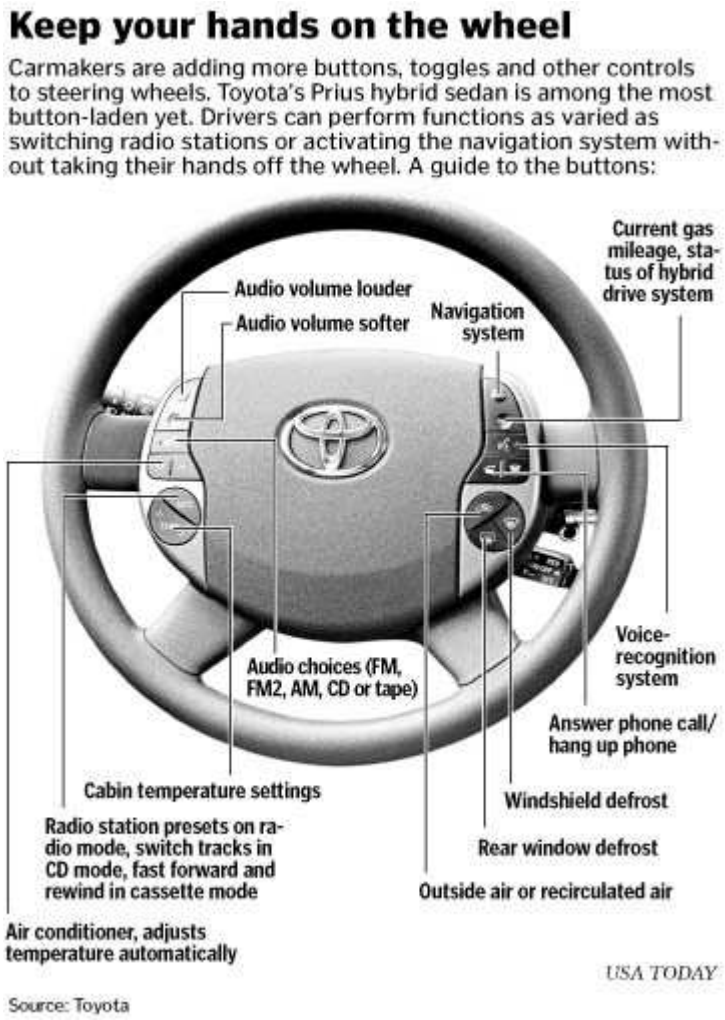


Figure 9: Controls on Toyota Prius steering wheel (Source: USA Today 5/4/2005)

By taking advantage of the extra controls available in modern automobiles, it is possible to improve usability and driver safety. Although adding manual input gives more freedom and convenience to the user, vocal commands via speech recognition provides more flexibility and can incorporate a larger number of functions.

2.6.2 Speech Recognition

Speech recognition technology has existed for decades; however like many other technologies, improvements in design and computing power have resulted in products ready for real-world use. Early speech recognition technologies relied on discrete recognition where users were required to dictate each word individually, separated by unnatural pauses. This design allowed more accurate speech recognition at the sacrifice of usability and speed. The opposite

and more preferred method is known as continuous recognition which allows users to speak naturally to the system. Although this is inherently a less accurate method, recent recognition technologies have reduced the error rate to a point where it is preferred for most human-computer interactions. Some recognition technologies additionally rely on speaker-dependent implementations. This implementation requires the users to read predefined passages to adapt to an individual's voice and manner of speaking. Similar to discrete recognition, this method produces more accurate recognition at the cost of flexibility and user setup. The opposite method is speaker-independent, allowing any user to address the system. While this method is less accurate than a trained system, for less needy applications, such as issuing commands, the accuracy level remains high.

Certain speech recognition software, such as IBM's ViaVoice or Dragon NaturallySpeaking, is designed to transcribe user speech into text. These speaker-dependent programs give the user almost unlimited control in what can be said. While this is very useful for dictation, it provides features unnecessary for a command driven system. Additionally, these multi-purpose software programs require considerable processing power and memory considerations, restricting their use in embedded environments such as an automobile. For this reason, companies have released speech recognition technologies designed specifically for a more command driven interaction. One such program is released by Nuance, a leader in speech recognition that holds the world's largest collection of speech and imaging software. Their program, released as VoCon 3200, is a member of a larger family of speech recognition engines designed specifically for embedded applications. VoCon 3200 is specifically designed for automotive environments in applications such as navigation systems, MP3 player control, and voice dialing. VoCon 3200 provides continuous recognition while being speaker-independent.

Any speech recognition system requires some base vocabulary of words or phrases to recognize. Nuance's VoCon 3200 can produce these recognizable statements through a text file known as a "grammar." This grammar is compiled at run-time and provides the recognition engine with a set of acceptable statements. When speech is fed to the system, the engine attempts to match what was said to these acceptable commands. The end result is the word or phrase that was said with an accompanying confidence level, indicating how sure the engine is that the command was acceptably recognized.

The following text is an example of an acceptable grammar file that recognizes the phrases “Hello” and “Hello there.” The pipes character ‘|’ is equivalent to an “or” statement. The first three lines are part of the header. The “!start” command indicates where recognizable phrases begin.

```
#BNF+EM V1.0;
!grammar "HelloDialog";
!language "American English";

!start <hello>;
<hello>: hello | hello there;
```

Under default conditions, the return value is the recognized phrase. However the grammar can be modified so that something entirely different is returned. This returned result is especially useful when multiple phrases are designed to issue the same function. The VoCon software provides this feature through the use of “!action” and “!ignore” flags. The “!action” flag returns the associated value when the corresponding phrase is recognized. The “!ignore” flag simply tells the engine not to return any of the associated words. In the following snippet, the value “next” is returned when “play the next song,” “play the next one,” or “go to the next song” is said. Parentheses can be used to subsection words or phrases.

```
!start <next_track>;
<next_track>: !action(“next”) !ignore( play the next (song
     | one) | go to the next song );
```

Additionally, grammar creation is simplified through the use of the “!optional” flag. In the following example, “what is this?” and “what is this song?” are recognized as the same action.

```
!start <what_song>;
<what_song>: !action(“song”) !ignore( what is this
     !optional(song) );
```

VoCon provides a simple way to create machine-recognizable commands from multiple user statements. Because VoCon 3200 is designed for automotive applications, it also offers adaptation to noise characteristics, advanced voice-activity detection, and a built-in noise cancellation algorithm (ScanSoft, 2005). This robustness in grammar and features makes it an excellent choice for command recognition.

3 Focus Group

On October, 28 2005, a half-hour focus group was conducted to brainstorm innovative song selection methods. A recruitment email had been sent out to WPI undergraduates and graduates seeking interested participants. Upon completion of recruitment, we conducted the focus group with nine individuals. The main goal of the focus group was to come up with a list of any possible song selection techniques for a voice-controlled automotive environment no matter how difficult they might be to implement. Before the focus group was conducted, we had compiled a list of existing song selection methods through background research. By conducting a focus group, we were able to expand our list with the help of other WPI students.

At the commencement of the focus group, we provided the participants with background information regarding the project and the main goals of the session. We then had one of our team members "lead" the discussion by introducing a few of our existing ideas, and asking questions about them. The list of questions provided in Appendix A: Focus Group Guide served as the overall guide of the session. Although we attempted to follow this guide, it was encouraged to present relevant ideas at any time even if they were slightly off-topic from the guide. It was important that the group feed off each other's ideas and we did not wish to constrict them with specific questions. If the group went off on a tangent for too long, the leader made sure the discussion was refocused to song selection features. Another member of our team wrote down all the ideas on a white board as they were presented. By writing down each suggestion, we hoped to spark new ideas from other members of the group.

Upon completion of the focus group, we had many new song selection features worth considering for the project. We also had additional interesting ideas that may be beyond the scope of this project, but could perhaps be examined in the future. We broke down the ideas into two categories: "mild" and "wild." The mild ideas were features that seemed plausible to be included in our design, while the wild ideas would require additional research and more time to implement. The results of the focus group can be seen in Appendix B: Focus Group Results. The actual song selection features that ended up in the design are explained in detail in the following section (see Jukebox Functionality).

4 Jukebox Functionality

Upon completion of song selection investigation, we were able to implement 44 new functions in the MP3 Jukebox. The functions include song selection methods and playlist navigation features as well as Jukebox feedback regarding song and playlist navigation. The grammar developed for the speech recognizer is able to recognize approximately 1350 distinct commands that perform various functions. Most features have fifteen to twenty acceptable word or phrase variants that will execute the desired action. We tested the accuracy of each available command by reading each command a single time and marking an error any time the recognizer failed to identify the appropriate action. The Jukebox exhibited a success rate of 95.53%, corresponding to 60 errors out of 1343 commands. This test was conducted with a music database consisting of 1708 songs, 149 artists, and 138 albums. A variety of genres were available including rap, rock, classical, jazz, and techno. This section will discuss in detail each of the features implemented in the MP3 Jukebox design and how to perform each function. Each available function and its associated commands can be seen in Appendix E: List of Acceptable Commands.

4.1 Software Functionality

The software begins by scanning though a designated directory that holds the MP3 files. When an MP3 file is recognized, the metadata stored in the MP3s tags are examined. The metadata includes the song title, the album name, the artist name, the release year, the track number, and the genre. This information is joined with the file path, a rating, and a playcount where the latter two are initialized to zero. The resulting song data is then uniquely tagged and added to a Microsoft Access database. In the database, tables are created for songs, albums, and artists. Having multiple tables for these values simplifies selection during speech recognition and song selection.

Once scanning of the folder is complete, the Jukebox software then goes through the created tables and creates a grammar file that becomes the format for speech recognition. Each song, album, and artist name is included as a recognizable word or set of words. The grammar file consists of all the accepted ways to say a command. The grammar file is created so that

every variation returns the same action. Once the grammar file has been created, it is sent to the speech recognition engine to be compiled. At this point, all the necessary components are initialized and the Jukebox begins the cycle of waiting until the user initiates speech recognition.

Each time the user wishes to issue a verbal command, they must first press the "talk" button to prompt the speech recognizer that a command is going to be issued. On the Logitech racing wheel used in driver distraction testing (see Testing Methodology), the "talk" button is located in the top right corner of the center of the steering wheel (see Figure 10). Once the button has been pressed, the volume of the Jukebox (if a song is playing) is reduced and the speech recognizer is told to wait for a command. The volume reduction serves two purposes – telling the user it is waiting for input and ensuring that the song playing is not confused for speech. The recognition engine waits until audio input is registered and then listens until it detects trailing silence (see Background: Speech Recognition). After the user has pressed the talk button, they may then issue any of the commands detailed in this section.

When a user has finished saying a command, the recognition engine makes the best match to an accepted phrase in the grammar file. The complete grammar file structure is provided in Appendix F: Jukebox Grammar File. The song, artist, and albums names have been removed to conserve space. Upon recognition, it returns the best matches to the Jukebox and the corresponding confidence level for each match. The confidence level is a general indicator for how well it was able to match the command. Our initial testing has shown that for our testing environment, confidence levels lower than 4000 are strong indicators of a miss-recognized command, and these actions are rejected. Factors that could cause a miss-recognized command include background noise or an unaccepted phrase. If an acceptable confidence level is returned, the Jukebox takes the best match (often multiple matches are returned) and interprets the action.

4.2 Standard Song Selection

Standard song selection commands enable users to select songs by song name, artist name or album title. When selecting songs in this fashion, the Jukebox clears any songs currently in the playlist and replaces them with the appropriate song, artist, or album. When an artist is selected, every song by that particular artist is queued up in the playlist. The order that the songs are entered into the playlist depends on the order that the songs were added to the music database. When a user selects an album, all songs from that album are queued up in the same

order as the album track list. Selecting a song by its name clears the current playlist and queues up the requested song. When using standard song selection methods, the Jukebox will confirm each request. For example, if the user asks to select the artist Michael Jackson, the Jukebox will reply, "Michael Jackson is playing." The following statements are some examples of standard song selection commands:

"Select Stairway to Heaven" (select by song name)
"Select artist Coldplay" (select by artist)
"Select album Morning View" (select by album)
"Select song Freebird" (select by song name)

4.3 Adding Songs to a Playlist

The "add" command appends a song, artist, or album to the end of an existing playlist. This command differs from the "select" command because it does not clear the current playlist; rather it adds the selection to the end. When adding music, the Jukebox uses the same method for ordering the songs as it does when using the "select" command. When songs are added to a playlist the Jukebox will reply with the confirmation, "<song> by <artist> has been added to the playlist." The following statements are some examples of adding songs to a playlist:

"Add song Under the Bridge" (adds a song)
"Add artist The Beatles" (adds an artist)
"Add album Thriller" (adds an album)

4.4 Shuffle

The "shuffle" command allows users to randomize the order of songs in the playlist. There are two different methods for shuffling songs using the Jukebox. The first method shuffles the remaining songs in the playlist, after a playlist is already queued up. If the "shuffle" command is issued while a playlist is running, the current song will continue playing, however all of the remaining songs in the playlist will be randomly shuffled. Below are a few examples of how to shuffle the remaining songs in a playlist:

"Shuffle" (shuffles the remainder of the playlist)
"Shuffle these songs" (shuffles the remainder of the playlist)

The second shuffle method randomizes the songs before they are entered into the playlist. Using this method, users can change the standard order that albums and artist songs are normally

queued up in the playlist. This method functions similarly to the "select" command because it clears the current playlist, and replaces it with the appropriate request. The following commands are examples of shuffling songs before they are entered into a playlist:

"Shuffle artist Michael Jackson" (shuffles an artist)
"Shuffle album Abbey Road" (shuffles and album)

Both shuffle techniques will prompt a confirmation from the Jukebox. If a running playlist is shuffled, the Jukebox will reply, "The remaining songs have been shuffled," while if songs are shuffled before they are queued up, it will respond, "<artist/album> has been shuffled."

4.5 Genre Selection

The MP3 Jukebox allows users to select songs based on their genre classification. Users must insert the appropriate genres into the music database before querying songs by genre. When a user selects a specific genre, all songs with that genre will be randomly queued up in the playlist. The Jukebox will give a confirmation if the command was processed correctly. The following statements are acceptable commands for selecting a music genre:

"Play genre rock"
"Play all songs from the genre rap"

All of the supported genres can be seen in Table 2 on the following page. The genres with a word in parenthesis refer to genres in which that specific word is optional. For example, "alternative" and "alternative rock" give the same genre, as do "metal" and "heavy metal."

Table 2: Available Genres

• acoustic • alternative (rock) • bluegrass • blues • classic rock • classical • club • comedy • country • dance • disco • duet • easy listening • electronic • emo • folk (rock) • freestyle • funk • gangsta (rap)	• gospel • gothic (rock) • grunge • hard rock • hardcore • hip hop • holiday • instrumental • jazz • latin • (heavy) metal • new age • oldies • pop • power ballad • progressive (rock) • psychedelic (rock) • punk (rock)	• R & B • rap • reggae • religious • rock • rock and roll • screamo • ska • slow jam • slow rock • soft rock • soul • soundtrack • southern rock • swing • symphony • techno • top 40

4.6 Beats Per Minute (BPM)

The Jukebox design allows users to select songs based on their tempo or beats per minute. In order for this command to be executed properly, the beats per minute of each song must be inserted into the music database. Currently, this task must be done by the user; however it is foreseeable that this could be an automated feature incorporated into the Jukebox in future designs. A program such as the MixMeister BPM Analyzer can be used to determine the beats per minute of each MP3 file. Once the music database is updated, users can request to listen to songs above or below a certain BPM value. When songs are selected according to their beats per minute, they are randomly inserted into a playlist, and clear whatever was previously in the playlist. The following commands are examples of acceptable BPM requests:

"Play all songs with beats per minute above 75"
"Play songs with beats per minute below 100"

4.7 Mood Selection

Users can assign a mood to specific songs, and later query the music database to play all songs with a particular mood. The current implementation requires moods to be manually assigned by the user in the music database. When songs are selected according to their mood,

they are randomly inserted into a playlist, and clear the previous contents. The following moods are supported by the MP3 Jukebox:

- angry
- dance
- groovy
- happy
- party
- rock
- sad/depressing
- soothing/relaxing

Some valid commands for selecting songs based on their mood include:

"Play me something groovy"
"Play mood happy"
"Play something I can party to"

4.8 Playlist Navigation

The MP3 Jukebox contains a number of commands for simple playlist navigation. These commands allow users to cycle through the playlist as well as pause and resume songs. Included in the standard playlist navigation commands are "next song," "previous song," "stop," "pause," "resume," and "command quit." The following statements are some examples of navigational commands:

"Next song"
"Play the previous song"
"Command quit"
"Pause this"
"Resume"

4.9 User Ratings

The MP3 Jukebox allows users to rate the songs in their music database. Songs can be manually rated within the music database, or they can be verbally rated while a song is playing. The ratings are based on a scale of one to five, with a rating of five corresponding to the user's most favorite songs. Users can then play all songs with a specific rating, or choose to play all songs rated above or below a certain number. When songs are queued up in the playlist according to their rating, they are randomly ordered. After a song has been rated, the rating can

be cleared, increased or decreased. By default, all songs are unrated (rated zero) until the user explicitly rates the song. Confirmations are given by the Jukebox upon all successful commands associated with user ratings. For example, if the user rates a song a three, the Jukebox will reply, "This song has been rated a three." Below are some examples of acceptable commands associated with user ratings:

"Rate this song a 4"
"Play all songs rated 3"
"Play song rated greater than 2"
"Increase the rating of this song" (increases rating by 1)
"I don't like this song" (decreases rating by 1)
"Play my favorite songs" (queues up all songs rated 4 or 5)

4.10 Playcounts

The MP3 player automatically keeps track of how many times each song has been played. Every time a song is started, its playcount is increased by one. Users can then request to listen to songs that have been played above or below a certain number. For example, the user can ask to listen to all songs that have been played under five times. This command will prompt a confirmation from the Jukebox saying, "Songs played under five times are now playing." Similar to user ratings, songs are randomly ordered in the playlist when they are queued up according to their playcount. Playing songs by playcount will replace the current playlist with all songs that match the request. Users can also clear the playcount at any time prompting the confirmation, "the playcount has been reset." The following statements are examples of acceptable playcount commands:

"Play all songs that have been played more than ten times"
"Play songs with a playcount of less than four"
"Reset the playcount of this song"

4.11 Song Information

New features regarding song and playlist information have been added to the MP3 Jukebox design. These features allow users to retrieve information about a playing song or playlist such as what year a song released, or how many songs are on the current playlist. Questions to the Jukebox regarding a specific song must be asked while the song is playing. The

following questions are some examples that will prompt the Jukebox to return specific information regarding the playing song:

"What song is this?" (returns song title)
"Who is this by?" (returns artist name)
"What is this album?"(returns album title)
"What year was this released?" (returns year of release)
"What is this song rated?" (returns assigned user rating)
"What is the playcount of this song?" (returns playcount)

Information regarding a playlist can be queried during any song in the playlist. These commands allow the users to easily navigate a playlist by returning information about the next and previous songs as well as information about the total number of songs in the playlist. The following examples will cause the Jukebox to return the appropriate playlist information field:

"What song is next?" (returns next song name)
"What was the last song?" (returns previous song name)
"How many total songs are there?" (returns total number of songs in the playlist)
"How many songs are left?" (returns the number of songs remaining in the playlist)
"How many have already been played?" (returns the number of songs that have already been played in the current playlist)

4.12 Multimodal Interface: Steering Wheel Buttons

In conjunction with the aforementioned verbal commands, a multimodal interface incorporating buttons on the steering wheel has been implemented in the Jukebox design. The buttons on the steering wheel allow the user to execute some of the more common commands without the need to consistently verbalize the command. Because we used the Logitech racing wheel during our driver distraction testing (see Testing Methodology), six buttons were available for use. One button was used as the "talk" button to prompt the speech recognizer that the user was ready to issue a command, leaving five buttons for a multimodal interface. The commands that are available through the use of steering wheel buttons include "pause/resume," "next track," "previous track," "song/artist name," and "shuffle." Figure 10, on the following page, shows how we implemented the buttons on the steering wheel.

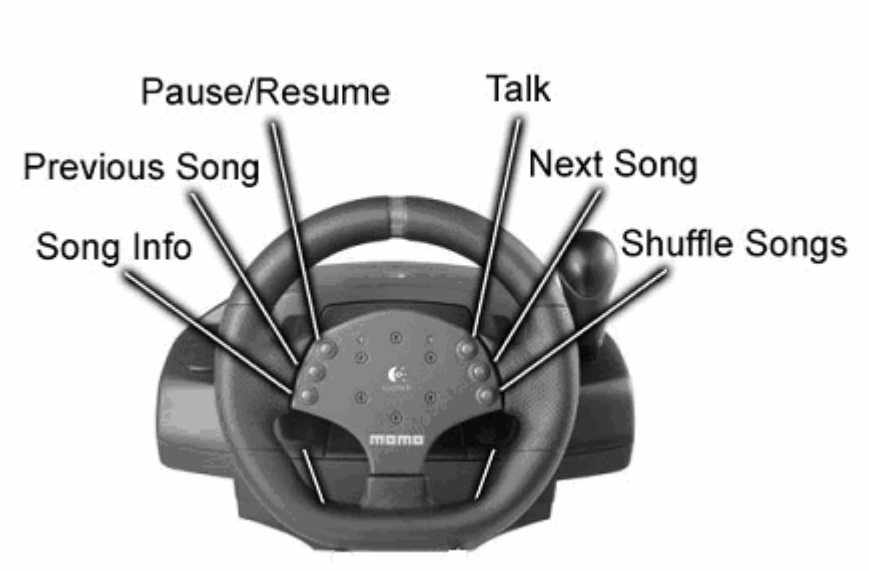


Figure 10: Steering wheel button interface

5 Testing Methodology

One of the most crucial aspects of this project was to implement song selection methods that would cause the least amount of distraction to the user. During our background research, we examined various methods of measuring driver distraction including driving simulators, on-road and test track studies, eye glance studies, and the Peripheral Detection Task (PDT). To test driver distraction, we decided to combine two of the aforementioned techniques. We implemented a driving simulator which incorporated the PDT to determine the distraction caused by different features of the MP3 Jukebox. The monetary cost of conducting test track studies or purchasing equipment for eye glance studies would be too expensive for this project. The driving simulator, in conjunction with the PDT, enabled us to obtain important driving performance metrics at a relatively low cost. This section will discuss in detail how we obtained data to measure how much distraction was caused by the MP3 Jukebox.

5.1 Testing Environment

The first step in obtaining driving performance data was to create a controlled environment where the actual testing was to occur. For a location, we used room 113B in the Atwater Kent building on the WPI campus that could be devoted strictly to testing. Each participant sat at a table facing the front wall of the room. On the table was a set of speakers, a microphone, and a PC steering wheel made by Logitech. At the participant's feet were mounted foot pedals; one for gas and one for brake which were part of the Logitech racing wheel. During testing, the lights in the room were turned off and a projector was used to cast the display of the driving on the white wall in front of them. The size of the display was approximately eight feet by six feet, and the subject sat ten feet from the projection wall. Another projector was used to cast PDT targets on the visual display in front of them. The computer game recording program *FRAPS* was used to record each test run so driving performance metrics could be later reviewed and scored. A schematic of an overhead view can be seen in Figure 11.

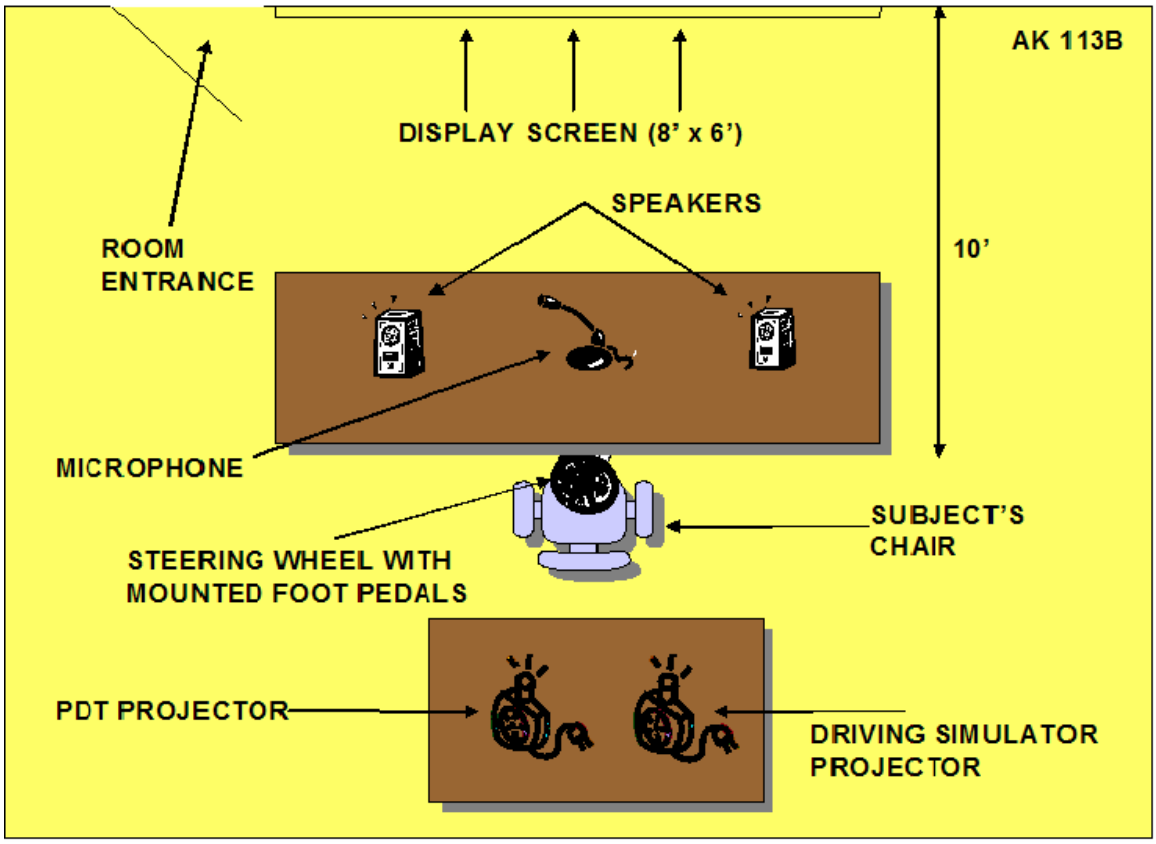


Figure 11: Testing environment

For a driving simulator, we used the PC video game *Need For Speed: Underground 2* by Electronic Arts. This game allowed for a variety of different testing scenarios and contained essential game settings for a controlled test environment. Also, it has short driving courses called "sprints" which do not allow the driver to deviate from the course. These sprints ensured that each test run was the same length and that no time was needed for the test participant to memorize a particular navigation route. Lastly, the game uses real-life driving environments that can be modified such as the time of day, weather, and location of driving (city versus freeway etc.).

The specific course we used during testing was called "Wall Center." This course is 2.42 miles long and has a combination of city and freeway driving. On average, the course took four to five minutes of driving to complete, and as mentioned before, the game did not allow the driver to stray from the course. The vehicle used by all test participants was the Honda Civic with automatic transmission. The traffic level was set to "off" to ensure no random obstacles

occurred and each test run was the same. Lastly, the "competition mode" was turned off so that no reckless racers caused distractions on the course.

The final component of our testing environment was the Peripheral Detection Task. To implement the PDT, we wrote a separate computer program that was run during the driving task. The program displayed PDT targets at random intervals during the test. A target was displayed once every four seconds plus or minus a random time interval between zero and two seconds. The PDT targets were bright red circles that appeared approximately six inches in diameter. The circles were projected on top of the driving display. Subjects had up to four seconds to respond to the target by hitting one of the paddles located just behind the steering wheel. If the user did not click the paddle before four seconds, it was logged as a missed target, and the target was removed. The targets were flashed at a rate such that approximately 60 targets were displayed during each test run. Subjects were asked to respond to the targets as quickly and safely as possible during the driving task.

5.2 Testing Protocol

The first step in completing human subject testing was to recruit participants to take part in the study. A recruitment email, which can be seen in Appendix C: Recruitment Email, was sent to the WPI faculty, staff, and students asking interested persons to reply. The only restrictions we placed on interested participants were that they must be fluent in English with no strong accent, and that they have no visual impairment that could affect the ability to use a driving simulator. We tested four participants, all of whom were males. After the subjects were chosen for testing, we scheduled two, one-hour testing sessions with each person. All subjects gave written informed consent, and the testing procedure was approved by the WPI Institutional Review Board (IRB).

Before actual testing began, we designed three different test runs that would force the participant to use many of the different Jukebox features. These test runs can be seen in Appendix D: Test Runs. We designed the test runs so that each run contained similar types of commands to interact with the MP3 Jukebox. Comparisons could then be made regarding the amount distraction associated with a specific type of command such as a question versus a song selection command. The first run consisted of mainly questions regarding song information. The

second test run was filled mostly with different song selection techniques, and the last one contained a mix of all the various commands.

Each subject completed each test run twice. The first time, they were instructed to use only verbal commands, while the second time they were instructed to use the buttons on the steering wheel for the associated commands. By performing each run twice, we were able to measure possible differences in driver distraction and Jukebox accuracy between the two different interfaces (all verbal versus buttons). They were also asked to complete one test run while issuing no commands which was used as the control scenario to compare all other test runs. Lastly, the order that the participants completed each test run was chosen at random to attempt to eliminate any learning curve as testing proceeded.

The first testing session was used to allow the subject to become familiar with both the MP3 Jukebox as well as the driving simulator. No data were collected during the first session. When the subject first entered the room, they were provided with background information about the project as well as the overall goals of the testing. We then gave a brief demonstration of the Jukebox to show them some of the available features in the design. Next, we gave them a list of each of the available functions as well as acceptable commands that the speech recognizer should be able to recognize. This list can be seen in Appendix E: List of Acceptable Commands. We also showed them the functionality of the six buttons on the steering wheel. The subject was then allowed to freely interact with the system for twenty to thirty minutes. During this time, we helped them if they were having trouble getting the Jukebox to recognize commands. For example, if they were talking too loud or not clear enough, we would alert them. We also provided them with a list of all available songs and artists in the database. If they were not familiar with any of the songs, we asked for some artists they were familiar with so that the database could be updated before the second testing session.

After they seemed comfortable interacting with the Jukebox, we gave them time to use the driving simulator. During this time, we explained to them in detail what driving performance metrics they would be scored on as well as how to respond to PDT targets. Each subject was allowed to navigate the course four to five times to become comfortable with the controls of the video game. The last part of the first testing session was to explain to them exactly what would occur during their next testing session so there would be no surprises. Subjects were informed that we would be asking them to execute specific functions while driving, and they must issue

the appropriate command to the Jukebox. We gave each participant their own list of acceptable commands so that they could memorize the commands they were most likely to use.

The second session was where the actual driving distraction data were collected. Each subject was given the first ten to fifteen minutes to navigate the course a few more times before actual testing began. We also reinforced the specific driving performance metrics that they would be evaluated by. When the actual testing started, the test runs were selected at random, and the subject was asked to perform specific Jukebox functions. The functions they were asked to execute were read verbally to the participant. They then had to remember one of the acceptable commands for that function and issue it to the Jukebox. For example, the test administrator may have said, "Rate this song whatever you want between one and five," and the subject would then have to issue an appropriate command such as, "Rate this song a three." If the subject forgot a command or the Jukebox performed an incorrect action, the test administrator would help them get back on track so that the test run could be completed. All such instances were marked as Jukebox errors, and would later be used to determine the accuracy of the system.

At the conclusion of the test, the participants were asked to fill out a questionnaire regarding their testing experience. The questionnaire allowed users to provide subjective comments concerning the testing methodology as well as the MP3 Jukebox design. We used the data from the questionnaire to gain an understanding of the usability of the design, as well as opinions regarding distraction caused by Jukebox interaction. The questionnaire can be seen in Appendix G: Testing Questionnaire.

5.3 Performance Metrics

After the second testing session was completed, we recorded both driving performance metrics and Jukebox performance metrics by reviewing the screen recordings of each individual test run. The driving performance metrics that we recorded were lateral position of the car, speed maintenance, and PDT reaction times. The Jukebox performance metrics recorded included the number of Jukebox errors when issued a proper command, the number of errors when issued an improper command, and errors when using the buttons on the steering wheel. The screen recordings of the test runs were randomly chosen for scoring so as to "blind" the scorer to which test participant and which test run he was scoring, making it impossible to bias the results.

5.3.1 Driving Performance Metrics

To measure the lateral position of the car, we reviewed the screen recordings of each individual test run. The test participant was asked to stay in a single lane while navigating the course. A lane violation was recorded when a wheel of the vehicle touched or crossed either the right or left lane line. We did not score a subsequent lane violation until the vehicle returned to a traveling lane for a minimum of two seconds. This procedure ensured drivers were not penalized for multiple lane violations when hovering near the side of the lane. On turns with no obvious lane lines, lane violations were not scored. Four such turns existed on the course.

The next driving performance metric was speed maintenance. Once again, this metric was scored subsequent to testing during a review of the screen recordings. Each driver was asked to travel between the speeds of 25 and 45 mph during the test. A speed violation was recorded each time the driver deviated from that range for greater than two seconds. The driver was allowed to travel below 25 mph during three specific sharp turns. The driver was never allowed to exceed 45 mph.

PDT reaction times were measured during the actual test by the PDT program. The program automatically recorded how long it took to react to each target, the number of targets missed, and a time stamp of when each target was displayed. The driver was asked to respond to PDT targets a quickly as possible to ensure that longer PDT reaction times corresponded to increased distraction rather than a participant taking their time reacting to the targets.

5.3.2 Jukebox Performance Metrics

During testing, we recorded each Jukebox error as it occurred. Each time the MP3 Jukebox performed an action inconsistent with the command issued, it was marked as an SUI error. We broke down the Jukebox errors into two categories. The first type of Jukebox error was when the test participant correctly issued a command, and the Jukebox executed an inappropriate action due to an error in speech recognition. These types of errors counted against the accuracy of the Jukebox.

The second type of SUI error was when the user issued a command that was not available in our design and the Jukebox performed an inaccurate function. Examples of these errors are when the participants said a command that was not included in the grammar, or when they said an acceptable command but hesitated or mixed up their wording. These types of errors were not

counted against the accuracy of our design because the speech recognizer could not be expected to accurately execute these commands. A few instances occurred where the user said a command that should not have been recognized due to its absence from the grammar; however the Jukebox still performed the appropriate function. These instances were counted as accurately executed commands.

.

6 Results and Analysis

This section presents and discusses the results obtained by performing the procedures described in the preceding chapter. Four WPI students were selected as test subjects and completed both testing sessions. These subjects will be referred to as test subjects A - D.

6.1 Driver Distraction Analysis

To simplify discussion of the results, the term "speech" will refer to the user speaking to the Jukebox to issue commands via the "Talk" button. Similarly, the term "buttons" will refer to the user executing actions though the other five buttons located on the steering wheel. Initiating the "Talk" sequence constitutes a button press; however, we are considering this specific button press to be part of the Speech User Interface. In our analysis, additional cognitive load will be attributed to: listening to the tester issuing instructions, interpreting the instructions into Jukebox-recognizable commands, listening to Jukebox prompts, or changes in music. "Cognitive load" excludes the actual speech recognition cycle, from the "Talk" button press to completion, as well as the load cause by driving. The cognitive load experienced while driving is considered as the baseline because it is common across all test runs, including the control. The term "error" in this section will refer to lane violations, speed violations, missed PDT targets, and false PDT triggers.

The following table shows the number of commands, broken down by type, issued for each test run. The "Type of Command" column indicates the type of command executed by the user to the Jukebox. "Selects" are any commands that play songs or add them to a playlist. "Questions" are commands that ask the Jukebox for song information. "Commands" instruct the Jukebox to modify something such as the song rating or resetting the playcount of a song. "Navigation/Control" commands traverse the playlist or pause the Jukebox. The three columns under "Test Run Interaction Type" are the three different test runs designed to focus on the different types of commands described above. As described in the Testing Methodology, each of the three test runs was completed in speech-only and button-when-possible variations. The number of commands that can also be issued to the Jukebox through a button press is indicated

for each test run as "Button-Able." The actual commands issued in the test runs can be found in Appendix D: Test Runs. The "Overall" column combines the types of commands for the three test runs, providing a reasonable breakdown of commands that would be issued in real world use. Ultimately our results provide comparisons between different types of speech interaction as well as between buttoned and non-buttoned use.

Table 3: Number of Commands Broken-down by Type for the Test Runs

	Test Run Interaction Type			
Type of Command	**Selection**	**Question**	**Mixed**	**Overall**
Selects	5	1	3	9
Questions	0	7	5	12
Commands	0	3	1	4
Navigation/Control	12	3	6	21
Total	17	14	15	46
Button-Able	12	4	7	23
% Button-Able	70.6%	28.6%	46.7%	50%

6.1.1 Rules for Analysis

Our testing simulator was designed to provide us with comprehensive and accurate data. These data include timestamps of when a button was pressed, when each PDT (Peripheral Detection Task) appeared and was triggered, the start and end of each speech recognition cycle, and the start and end of each driving violation. In addition to looking at statistics measured over the entire test run, we flagged specific events that occurred either during or in close proximity to other events. The following criteria were used to evaluate the conditions for possible influence.

An event was considered to be influenced by speech if:

- For PDT targets
    - Appear before the "Talk" button has been pressed, but are triggered or missed after it has been pressed
    - Appear after the "Talk" button has been pressed, but before the speech recognition has completed
    - Appear after the "Talk" button has been pressed and are triggered or missed after recognition has been complete

- False PDT triggers
    - After the “Talk” button has been pressed, but before the speech recognition has completed
- For lane and speed violations
    - Occurs or begins after the “Talk” button has been pressed, but before the speech recognition has been completed

An event is considered to be influenced by the buttons if:

- For PDT targets
    - Appear before a button has been pressed and are triggered or missed afterwards
- For lane and speed violations
    - Occurs or begins within five seconds after a button has been pressed

An event was considered to be influenced by cognitive load if:

- For PDT targets
    - Appear and are triggered or missed within five seconds before any button press, “Talk” or otherwise
    - Appear and are triggered or missed within five seconds after any button press, “Talk” or otherwise
- False PDT triggers
    - Within five seconds before the “Talk” button has been pressed
    - Within five seconds after the “Talk” button has been pressed
- For lane and speed violations
    - Occurs or begins within five seconds before a button, “Talk” or otherwise, has been pressed
    - Occurs or begins within five seconds after a “Talk” sequence has completed

An event is considered to be influenced by the tester issuing commands if:

- For PDT targets
    - Appear and are triggered or missed within five seconds before any button press, “Talk” or otherwise
- False PDT triggers
    - Within five seconds before the “Talk” button has been pressed
- For lane and speed violations
    - Occurs or begins within five seconds before a button, “Talk” or otherwise, has been pressed

Events that did not meet any of the above criteria were considered to occur without active interaction with the Jukebox, where the test subject was simply listening to music. Some events overlap in possible influences. For example, a PDT target that appears during a “Talk” sequence may not be triggered until after recognition has completed, meanwhile a new instruction has

been issued by the tester. Events such as these receive multiple tags where such an event would be considered to be affected by both "Speech" and "Cognitive Load." For PDT reaction times, the values were counted for both influences. For errors, these events were scored as half an error for each influence.

6.1.2 Skewness Analysis

In the analysis, specific attention is paid to the median of each distribution, as well as its difference from the control median. The median values are more important because the distributions have shown that each is skewed to a significant degree. This skewness is true of all distributions except the "Manual" data where skewness cannot be commented on due to a small number of observations (less than 10 per test subject). The following distribution is an example of the PDT reaction times, in milliseconds, recorded by Subject C during Jukebox interaction.

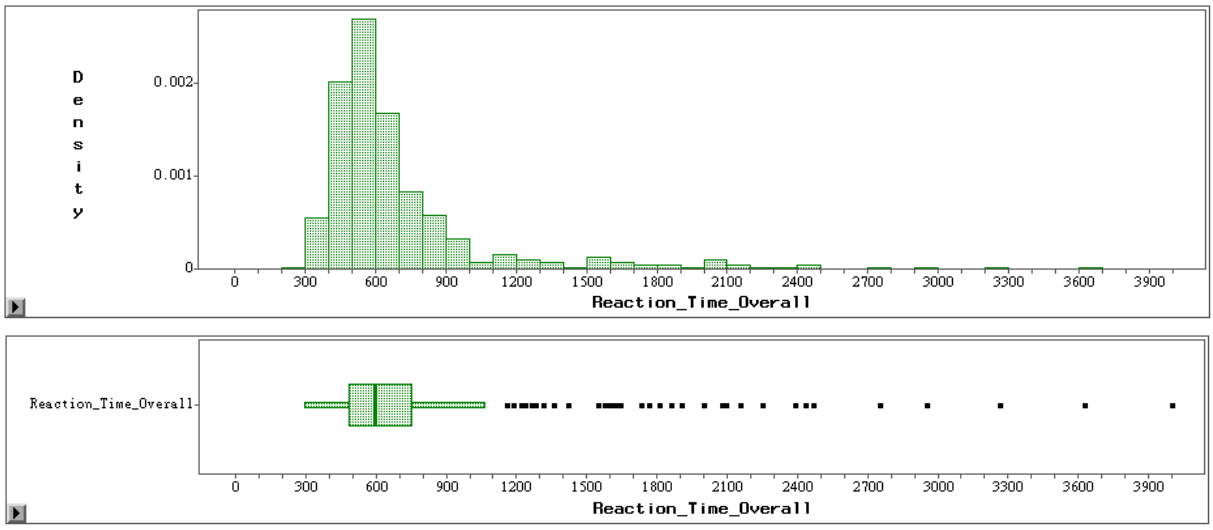


Figure 12: Subject C PDT reaction times (ms) for all tests with Jukebox interaction

Skewness values of two standard errors of skewness or more are probably skewed to a significant degree (Brown, 1997). The standard error of skewness can be estimated with the formula $\sqrt{\frac{6}{N}}$ where N is the number of observations in the sample set (Tabachnick and Fidell, 1996). For the number of observations in our samples, the standard error of skewness generally lies between 0.17 for the "Overall" data set to 0.30 for individual test runs. All data distributions,

"Manual" excepted, have shown skewness values ranging between 1.0 and 7.0. The positively or right skewed distributions for all data sets can be seen in Appendices H – K.

6.1.3 Summary of Test Run Data by Type

Data analysis was performed with the computer program SAS whereby distributions were charted for every aspect of interest. The aspects of interest by run type were:

- the control test run (Control)
- the six test runs with Jukebox interaction (Overall)
- the three test runs with speech only interaction (Speech)
- the three test runs where buttons were used when possible (Button)
- the two test runs where Jukebox interaction was primarily by song selection (Selection)
- the two test runs where Jukebox interaction was primarily with questions and user assignments (Question)
- the two test runs where Jukebox interaction was a mix of command types (Mixed)

The following tables show the summary data for each test subject for the above aspects. For each type of test run, the following data are reported: the number of driving errors (lane and speed violations), the number of PDT targets displayed, the number of PDT targets missed, the number of false PDT triggers, the average PDT reaction time in milliseconds, the standard deviation of PDT reaction times in milliseconds, the median PDT reaction times in milliseconds, and the difference of the median PDT reaction time from the control median. Additionally, each of the six individual test run summary data are provided for each test subject in Appendix L: Summary Data for Individual Test Runs.

Table 4: Subject A - Summary Data for Test Runs by Type

	Test Run Type						
			Interface Type		Command Type		
	Control	Overall	Speech	Button	Selection	Question	Mixed
Lane Violations	0	14	6	8	4	5	5
Speed Violations	0	20	8	12	10	4	6
PDT Observations	66	373	182	191	128	119	126
Missed PDT Targets	0	15	10	5	7	1	7
False PDT Trigger	0	9	4	5	3	4	2
Mean PDT Reaction Time (ms)	632	1145	1231	1064	1204	1055	1171
Standard Deviation of Reaction Times (ms)	254	843	930	744	931	657	903
Median PDT Reaction Time (ms)	**593**	**813**	**836.5**	**812**	**851.5**	**797**	**820.5**
Difference from Control Median (ms)		**+220.0**	**+243.5**	**+219.0**	**+258.5**	**+204.0**	**+227.5**

Table 5: Subject B - Summary Data for Test Runs by Type

	Test Run Type						
			Interface Type		Command Type		
	Control	Overall	Speech	Button	Selection	Question	Mixed
Lane Violations	1	7	4	3	2	3	2
Speed Violations	0	5	2	3	1	2	2
PDT Observations	62	348	176	172	116	118	114
Missed PDT Targets	0	0	0	0	0	0	0
False PDT Trigger	0	2	1	1	2	0	0
Mean PDT Reaction Time (ms)	539	785	747	823	801	811	742
Standard Deviation of Reaction Times (ms)	119	464	332	567	544	425	413
Median PDT Reaction Time (ms)	**500**	**656**	**641**	**657**	**602**	**687.5**	**625**
Difference from Control Median (ms)		**+156.0**	**+141.0**	**+157.0**	**+102.0**	**+187.5**	**+125.0**

Table 6: Subject C - Summary Data for Test Runs by Type

	Test Run Type						
			Interface Type		Command Type		
	Control	Overall	Speech	Button	Selection	Question	Mixed
Lane Violations	3	17	6	11	7	4	6
Speed Violations	0	11	7	4	4	4	3
PDT Observations	57	358	187	171	123	117	118
Missed PDT Targets	0	2	1	1	0	1	1
False PDT Trigger	0	5	3	2	1	3	1
Mean PDT Reaction Time (ms)	611	747	721	774	748	719	773
Standard Deviation of Reaction Times (ms)	512	521	4.78	564	517	462	580
Median PDT Reaction Time (ms)	**485**	**593.5**	**594**	**593**	**610**	**579**	**563**
Difference from Control Median (ms)		**+108.5**	**+109.0**	**+108.0**	**+125.0**	**+94.0**	**+78.0**

Table 7: Subject D - Summary Data for Test Runs by Type

	Test Run Type						
			Interface Type		Command Type		
	Control	Overall	Speech	Button	Selection	Question	Mixed
Lane Violations	2	20	10	10	7	8	5
Speed Violations	1	5	3	2	2	1	2
PDT Observations	65	357	179	178	115	125	117
Missed PDT Targets	0	0	0	0	0	0	0
False PDT Trigger	0	3	3	0	3	0	0
Mean PDT Reaction Time (ms)	481	609	614	604	635	576	618
Standard Deviation of Reaction Times (ms)	95	376	372	381	409	224	463
Median PDT Reaction Time (ms)	**453**	**500**	**515**	**485**	**500**	**500**	**500**
Difference from Control Median (ms)		**+47.0**	**+62.0**	**+32.0**	**+47.0**	**+47.0**	**+47.0**

Control and Overall Data

The following figures are sample distributions of the PDT reaction times, in milliseconds, for the Control and Overall test runs. These distributions are of data recorded by subject B.

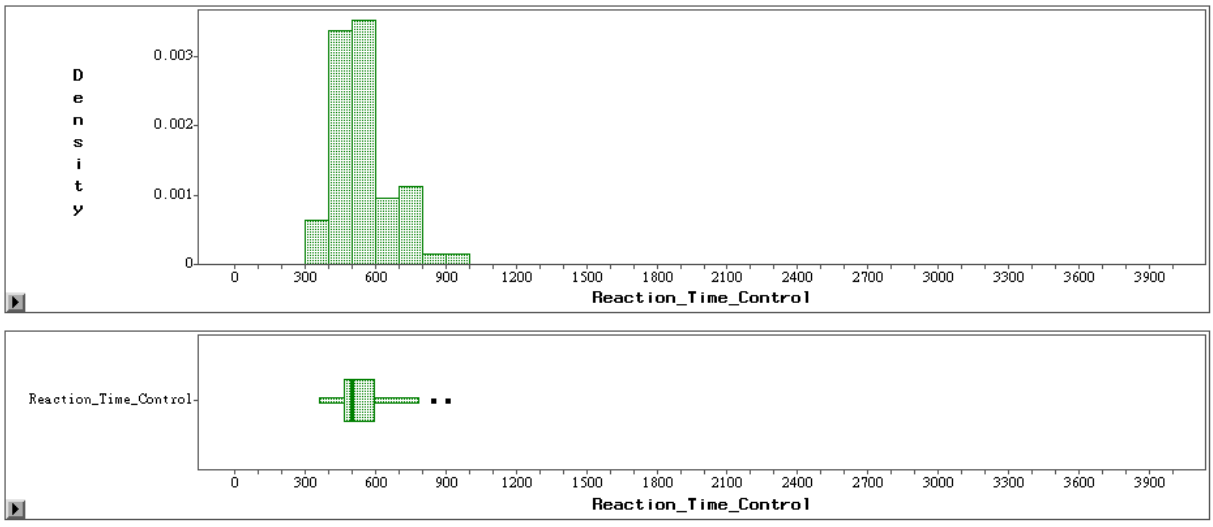


Figure 13: Subject B PDT reaction times (ms) for the control test – no Jukebox interaction

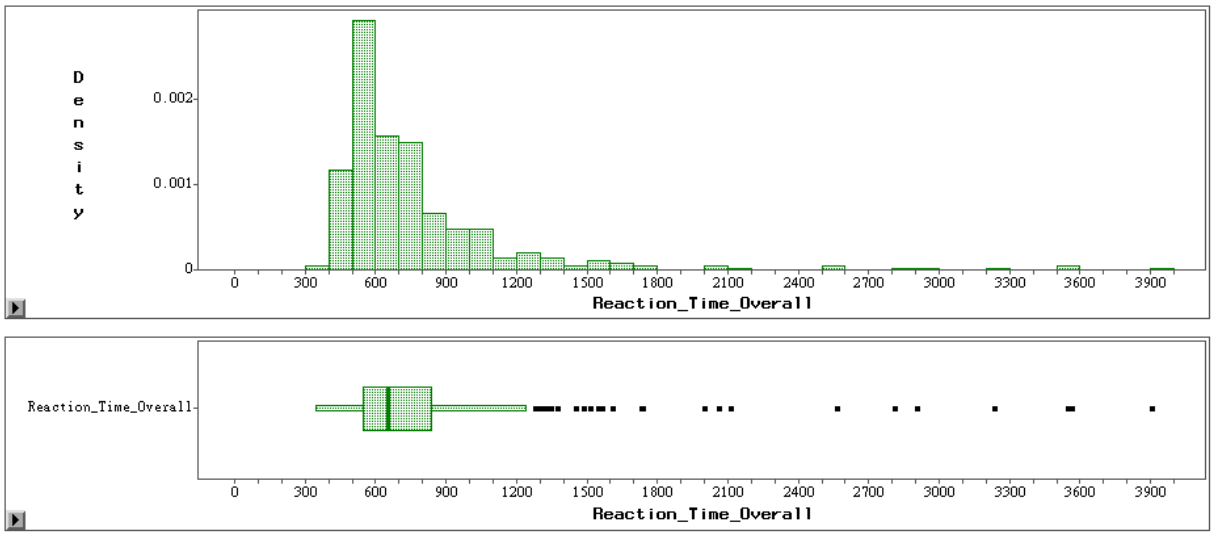


Figure 14: Subject B PDT reaction times (ms) for all tests with Jukebox interaction

Subject A

The control test for subject A shows a slight positive skew with a median value of 593 ms. Although a few outliers exist, with one response time reaching 2078 ms, the distribution is

reasonably dense with a standard deviation of 254 ms. The upper quartile contains values 656 ms to 2078 ms. This test subject recorded a perfect control run with no driving or PDT errors. The data from all test runs with Jukebox interaction exhibit a very large positive skew, with a standard deviation of 843 ms. The median value is 813 ms, 220 ms above the control median. The upper quartile contains values 1266 ms to 4000 ms. Overall, test subject A recorded 14 lane violations, 20 speed violations, 15 missed PDT targets, and 9 false PDT triggers. Because this subject's test run was perfect, we cannot comment on the effect the Jukebox had on driving performance without more data.

Subject B

The control test for subject B shows a very slight positive skew with a median value of 500 ms. The distribution is dense with a standard deviation of 119 ms. The upper quartile contains values 594 ms to 906 ms. This test subject recorded 1 lane violation and no PDT errors. The Overall data exhibit a large positive skew, with a standard deviation of 464 ms. The median value is 656 ms, 156 ms above the control median. The upper quartile contains values 835.5 ms to 3906 ms. Over the course of six Jukebox interaction runs, this test subject recorded 7 lane violations, 5 speed violations, and 2 false PDT triggers. Though the per-run lane violations match well with the control run at approximately one each, more control data would be needed to comment on the differences in driving performance.

Subject C

The control test for subject C shows a positive skew with a median value of 485 ms. The distribution is less dense that other subjects' control data. Additionally, the control data includes two reaction times greater than 3000 ms. The upper quartile contains values 610 ms to 3250 ms. This test subject recorded 3 lane violations and no PDT errors. The data from all test runs with Jukebox interaction exhibit a positive skew with a very large tail. The standard deviation is 521 ms. The median value is 593.5 ms, 108.5 ms above the control median. The upper quartile contains values 750 ms to 4000 ms. Overall, this test subject recorded 17 lane violations, 11 speed violations, 2 missed PDT targets, and 2 false PDT triggers. More control data would be needed to comment on the differences in driving performance. Similar to test subject B, the per-run lane violations match well with the control run at approximately 3 each.

Subject D

The control test for subject D shows a slightly positive skew with a median value of 453 ms. The distribution is the most dense of the four subjects with a standard deviation of 95 ms. Additionally, the control data includes two reaction times greater than 3000 ms. The upper quartile contains values 516 ms to 859 ms. This test subject recorded 2 lane violations, 1 speed violation, and no PDT errors. The Overall data exhibit a positive skew with a tail smaller than the other three subjects' data. The standard deviation is 376 ms. The median value is 500 ms, a small 47 ms above the control median. The upper quartile contains values 625 ms to 3922 ms. Overall, this test subject recorded 20 lane violations, 5 speed violations, no missed PDT targets, and 3 false PDT triggers. Though the amount of data is insufficient, the per-run speed and lane violations match somewhat well with the control run.

Overall Analysis

Driver reaction time studies have shown that when a driver is alert and aware of the possibility for an event, the best estimate of reaction time is 0.7 seconds. Of this, 0.5 seconds is perception and 0.2 seconds is attributed to the time required to release the accelerator and to depress the brake pedal (Green, 2000). In our testing environment, the driver's fingers are very near to, or may already be on, the trigger, reducing the movement time an estimated 50 to 100 milliseconds. The median reaction time of the test subjects' control runs match these values well, supporting the validity of our measurements of reaction time.

The four test subjects exhibited an increase in PDT reaction time on average of 133 ms when doing test runs with Jukebox interaction compared to their control runs. Though there is not enough driving data to make definitive claims, more test runs may show a trend that Jukebox interaction does not affect lane and speed violations. Subject A's reaction times appear to be considerably influenced by Jukebox interaction whereas subject D's reaction times are affected significantly less. Subject B did very will with the driving errors, producing less than half the lane violations of the other subjects. Since each of these drivers show different diving abilities, the results are more likely to represent a larger population. Comparisons of the Overall and Control show two trends:

- Any Jukebox interaction noticeably increases PDT reaction times,
- Jukebox interaction does not seem to influence the number of driving errors.

This implies that PDT is a more sensitive test to driver distraction than driving violations.

Speech versus Button Interfaces

One of the primary goals of this MQP was to design a Jukebox interface that incorporates manual or buttoned input. To see if this distinction made an impact on the distraction caused to the driver, we completed test runs with identical commands, once using speech commands only and once using buttons whenever possible. The following figures are sample distributions of the PDT reaction times for the Speech and Button test runs for subject D.

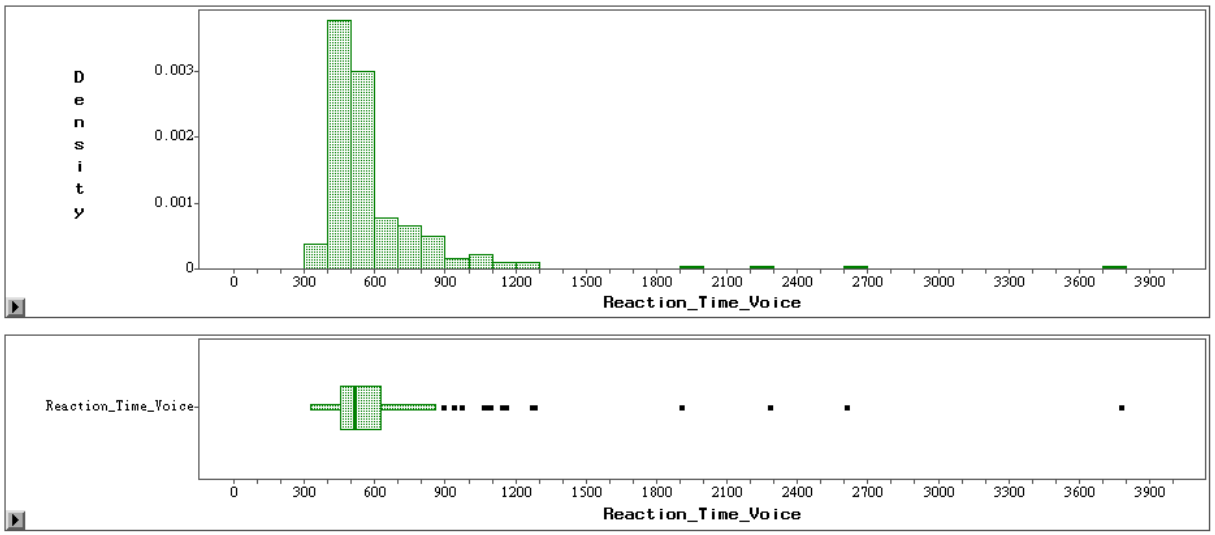


Figure 15: Subject D PDT reaction times (ms) for tests using speech input only

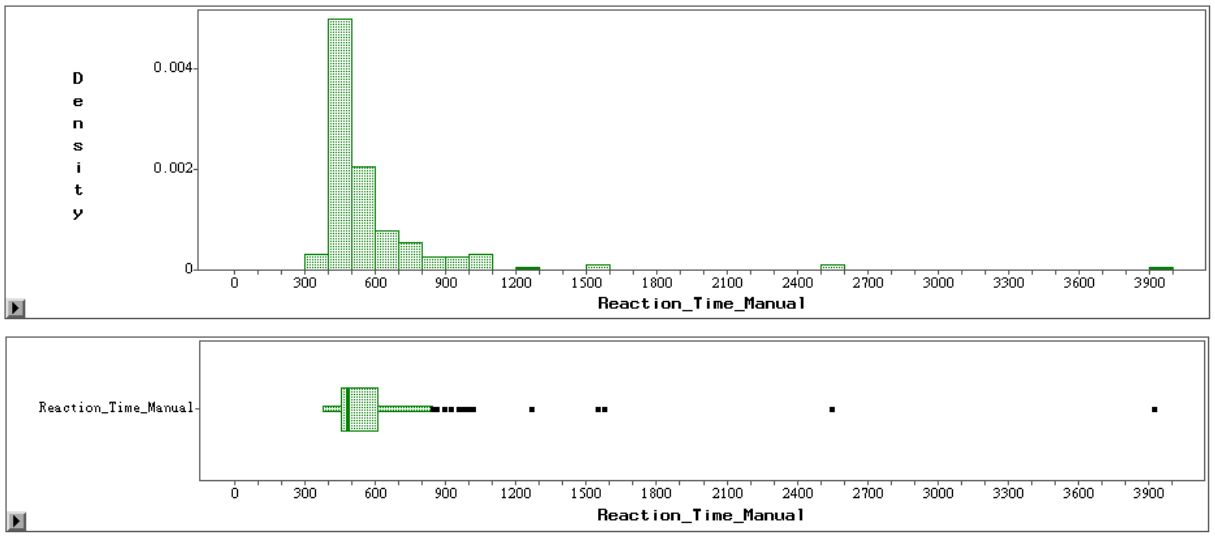


Figure 16: Subject D PDT reaction times (ms) for tests using buttoned input when possible

Subject A

Comparing the Speech test runs to the Button test runs for subject A, the distributions are very similar with both exhibiting large positive skews. The median values are 836.5 ms and 812 ms respectively. Although the Button data show a slightly lower mean reaction time and smaller standard deviation, these differences are marginal and cannot alone indicate any trend. The driving errors do not alone indicate a trend between interfaces, although it may be of note that half as many PDT targets were missed (5 compared to 10) when using the buttoned interface.

Subject B

The Speech test runs and the Button test runs for subject B have very similar distributions with median values of 641 and 657 respectively. The Button data exhibit a larger tail, causing a standard deviation 235 ms greater than the Speech data which does not support trend seen in subject A's data. The driving errors for this subject do not indicate a trend between interfaces.

Subject C

The Speech test runs and the Button test runs for subject C have nearly identical distributions with median values of 594 and 593 respectively. The mean values and standard deviations are also very similar. The driving errors for this subject do not indicate a trend between interfaces.

Subject D

The data for subject D's Speech and Button test runs have very similar distributions with the exception of a less pronounced tail in the Button distribution. The median values are 515 for the Speech data and 485 for the Button data. The mean values and standard deviations are very similar. The driving errors for this subject do not indicate a trend between interfaces.

Overall Analysis

The data suggest that the buttoned interface does not increase or decrease driver distraction through our metrics of driving performance and reaction time. While two subjects support a possible trend that a buttoned interface decreases distraction, one opposes such a trend; more data are needed to effectively comment on such trends.

Different Command Types

Breaking test runs into different types of commands allowed us to see the influence that the command types may have on distraction. The three different test runs we examined are Jukebox interaction primarily by song selection (Selection), Jukebox interaction primarily with questions and user assignments (Question), and Jukebox interaction with a mix of command types (Mixed). Subject A's following three distributions of PDT reaction times are examples of the different test runs by command type.

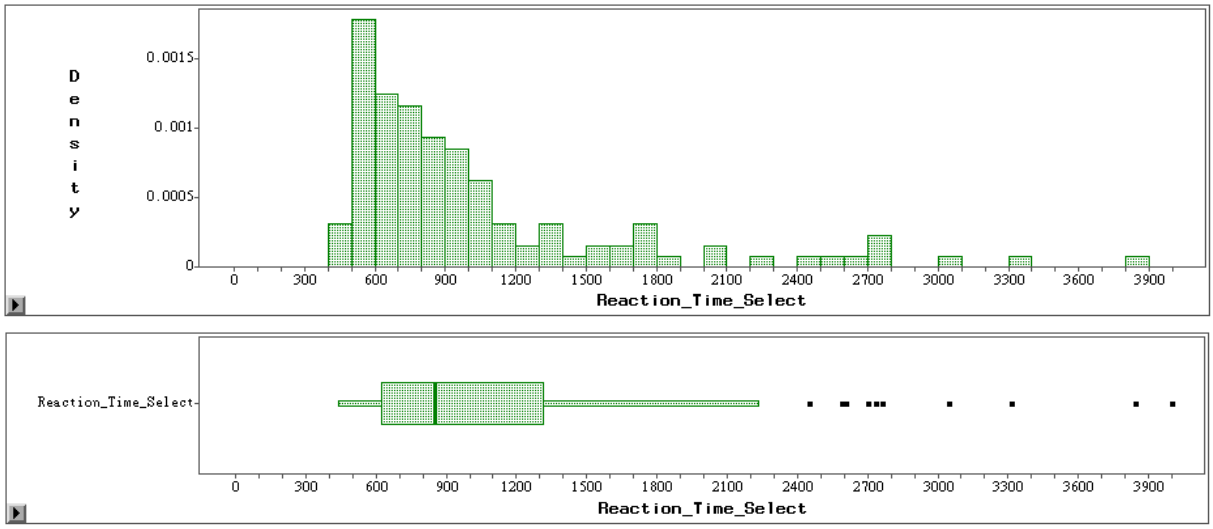


Figure 17: Subject A PDT reaction times (ms) for tests with Jukebox interaction using song selection

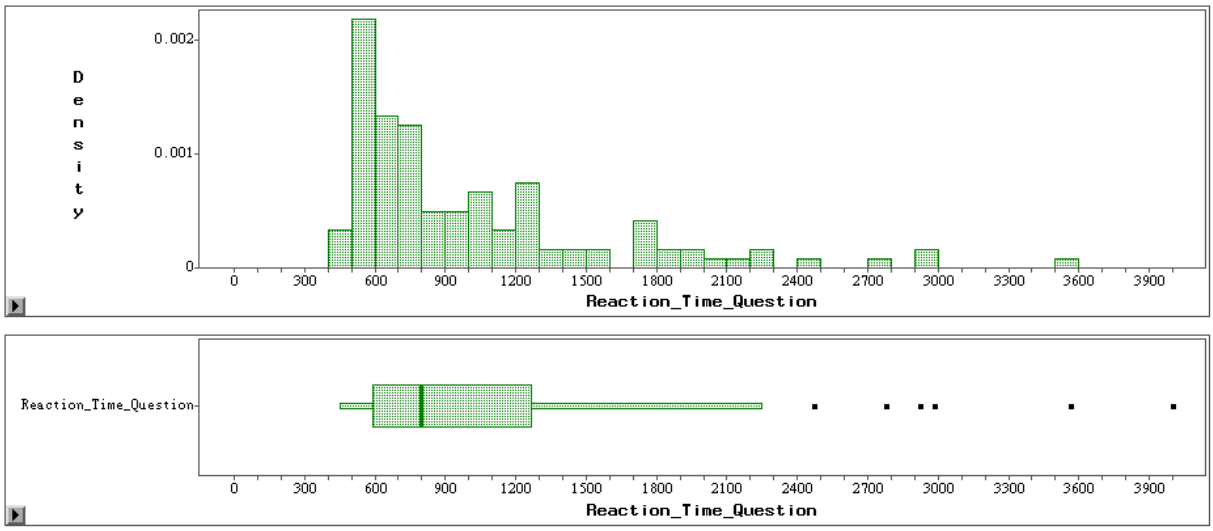


Figure 18: Subject A PDT reaction times (ms) for tests with Jukebox interaction using questions and user assignments

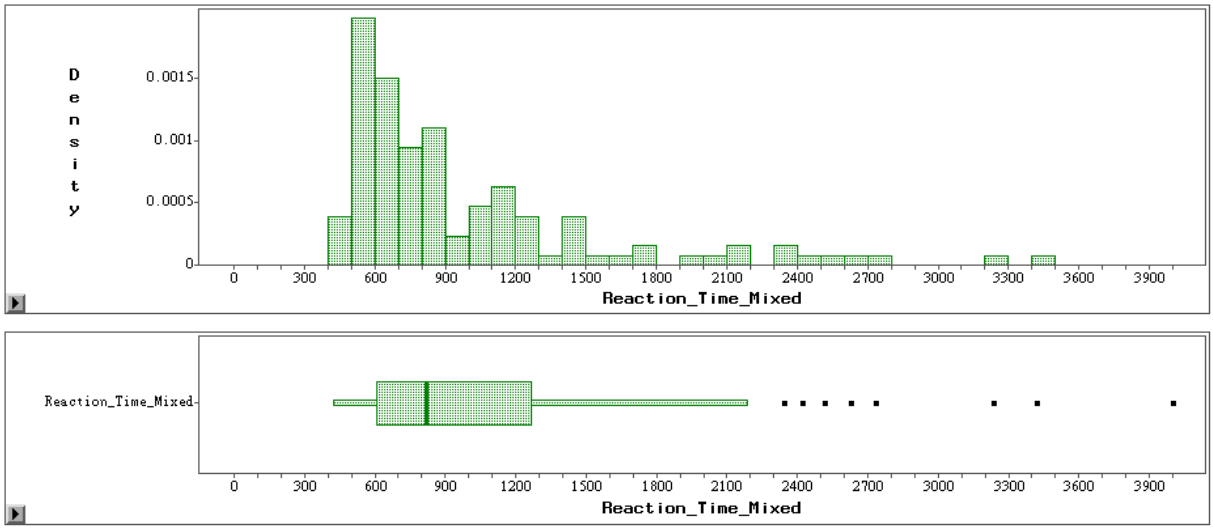


Figure 19: Subject A PDT reaction times for tests with Jukebox interaction using a mix of commands

Subject A

Subject A's test runs for the three different types of commands show similar distributions with median values of 851.5, 797, and 820.5 ms for Selection, Question, and Mixed. The Question data may indicate a possible trend of less distraction compared to Selection or Mixed interaction, showing a 20-50 ms lower median reaction time, a 150-200 ms lower average, as well as a 250 ms lower standard deviation. Supporting this possible trend are the fewer speed violations, missed PDT targets, and false PDT triggers.

Subject B

For Subject B, the three types show similar distributions with median values of 602, 687.5, and 625 ms for Selection, Question, and Mixed. The Question data appear to be centered higher than the other two types, opposite the trend seen in subject A. The driving errors for this subject do not indicate a trend between command types.

Subject C

The three runs also show similar distributions for subject C with median values of 619, 579, and 563 ms for Selection, Question, and Mixed. The Question data has a smaller tail than the other two command types, leading to a lower mean and standard deviation. This trend of

lower distraction is shared by subject A. The driving errors for this subject do not indicate a trend between command types.

Subject D

The data for subject D yield the exact same median reaction time for all three types at 500 ms. However, the tails for the Question and Mixed types are less pronounced with Question's mean and standard deviations are 57 ms and 185 ms lower than Selection's. The Mixed data has the smallest tail though it is offset by two reaction times above 3700 ms. The driving errors for this subject do not indicate a trend between command types.

Overall Analysis

These distributions suggest a slight possibility for a trend that Question type commands cause less distraction, however the small differences and opposition from one subject make it seem less likely; much more testing will be needed to validate such claims.

6.1.4 Summary of Data by Specific Event Condition

The data for specific event conditions exemplify the differences, if any, that Jukebox interaction creates. By looking at when each event occurred, we can gain a better understanding of the probable causes of distraction. Data analysis was performed for the aspects of interest according to specific event conditions:

- events during test runs that occurred while simply listening to music (Listening)
- events during test runs that occurred during any sort of interaction conditions (Interaction)
- events during test runs that occurred during cognitive load conditions (Cognitive Load)
- events during test runs that were possibly influenced by the tester issuing commands (Command Issue)
- events during test runs that were possibly influenced by speech (Dialog)
- events during test runs that were possibly influenced by button use (Manual)

As with the previous tables of summary data, the same driving violations and PDT information are presented.

Table 8: Subject A - Summary Data for Specific Event Conditions

	Event Conditions						
	Control	Listening	Interaction	Cognitive Load	Command Issue	Dialog	Manual
Lane Violations	0	5	9	8	2	1	0
Speed Violations	0	6	14	5.5	2.5	8.5	0
PDT Observations	66	152	221	130	68	82	11
Missed PDT Targets	0	1	14	2.5	2.5	10.5	1
False PDT Trigger	0	5	4	2	1	2	0
Mean PDT Reaction Time (ms)	632	832	1361	1086	1190	1704	2221
Standard Deviation of Reaction Times (ms)	254	454	972	726	819	1150	1127
Median PDT Reaction Time (ms)	**593**	**672**	**953**	**804.5**	**844**	**1281**	**2188**
Difference from Control Median (ms)		**+79.0**	**+360.0**	**+211.5**	**+251.0**	**+688.0**	**+1595.0**

Table 9: Subject B - Summary Data for Specific Event Conditions

	Event Conditions						
	Control	Listening	Interaction	Cognitive Load	Command Issue	Dialog	Manual
Lane Violations	1	2	5	4	1.5	1	0
Speed Violations	0	1	4	2	0	2	1
PDT Observations	62	116	232	144	81	82	7
Missed PDT Targets	0	0	0	0	0	0	0
False PDT Trigger	0	1	1	0	0	1	0
Mean PDT Reaction Time (ms)	539	644	855	763	792	947	1701
Standard Deviation of Reaction Times (ms)	119	178	541	383	447	628	1150
Median PDT Reaction Time (ms)	**500**	**594**	**688**	**679.5**	**687**	**719**	**1313**
Difference from Control Median (ms)		**+94.0**	**+188.0**	**+179.5**	**+187.0**	**+219.0**	**+813.0**

Table 10: Subject C - Summary Data for Specific Event Conditions

	Event Conditions						
	Control	Listening	Interaction	Cognitive Load	Command Issue	Dialog	Manual
Lane Violations	3	7	10	5	2	5	0
Speed Violations	0	2	9	8	1.5	1	0
PDT Observations	57	109	249	146	86	92	16
Missed PDT Targets	0	0	2	0	0	1	1
False PDT Trigger	0	1	4	1	1	3	0
Mean PDT Reaction Time (ms)	611	636	795	695	723	817	1624
Standard Deviation of Reaction Times (ms)	512	321	581	406	393	622	939
Median PDT Reaction Time (ms)	**485**	**562**	**609**	**586**	**601.5**	**609.5**	**1570**
Difference from Control Median (ms)		**+77.0**	**+124.0**	**+101.0**	**+116.5**	**+124.5**	**+1085.0**

Table 11: Subject D - Summary Data for Specific Event Conditions

	Event Conditions						
	Control	Listening	Interaction	Cognitive Load	Command Issue	Dialog	Manual
Lane Violations	2	6	14	13	7	1	0
Speed Violations	1	0	5	3	0	2	0
PDT Observations	65	133	224	143	83	72	9
Missed PDT Targets	0	0	0	0	0	0	0
False PDT Trigger	0	0	3	2	2	1	0
Mean PDT Reaction Time (ms)	481	511	667	572	582	854	687
Standard Deviation of Reaction Times (ms)	95	171	446	233	184	678	203
Median PDT Reaction Time (ms)	**453**	**469**	**531**	**500**	**547**	**656.5**	**609**
Difference from Control Median (ms)		**+16.0**	**+78.0**	**+47.0**	**+94.0**	**+203.5**	**+156.0**

Listening, Interaction, Cognitive Load, and Command Issue Influence

Subject A

Subject A's Interaction with the Jukebox shows a median reaction time of 953 ms, 360 ms greater than the control median. Its distribution is heavily tailed, with the upper quartile beginning at 1719 ms. The data taken when the subject was just listening to music is significantly less skewed with a median 79 ms above the control median. The Cognitive Load PDT and error data indicate the possibility that direct Jukebox interaction is more distracting than listening to the tester, listening to Jukebox prompts, or command results such as songs changes.

Subject B

Subject B's Interaction with the Jukebox shows a median reaction time of 688 ms, 188 ms greater than the control median. Its distribution has a large tail, with the upper quartile beginning at 906 ms. The data taken when the subject was just listening to music is significantly less skewed with a median 94 ms above the control median. The Cognitive Load data is less tailed than the interaction data but shows similar values for other measures and errors.

Subject C

Subject C's Interaction yields a median reaction time of 609 ms, 124 ms greater than the control median. Its distribution has a smaller tail, with the upper quartile beginning at 812 ms. The Listening data is significantly less skewed with a median 77 ms above the control median. The Cognitive Load data is slightly less tailed than the interaction data with a mean 100 ms and standard deviation 175 ms lower, indicating the possibility that direct Jukebox interaction is more distracting than indirect interaction.

Subject D

Subject D's Interaction produced a median reaction time of 531 ms, 78 ms greater than the control median. Its distribution is slightly tailed, with the upper quartile beginning at 695.5 ms. The Listening data is significantly less skewed with a median only 16 ms above the control median. The Cognitive Load data is less tailed than the interaction data with a mean 95 ms and standard deviation 213 ms lower, further supporting the possibility that direct Jukebox interaction is more distracting than indirect interaction.

Overall Analysis

Driving performance cannot be commented on as easily because driving errors are greatly affected by the amount of time driving and these statistics are extracted from the test runs. However we can get a rough estimate by using the number of PDT observations as a basis for the amount of time allocated to each event condition. Doing such an analysis shows that error rates are similar across all four test subjects.

Three of the four test subjects support the trend that direct Jukebox interaction, such as engaging in commands, produces more driver distraction than just listening to music. It should be noted that because these data are extracted, other factors such as stress during the test run may affect the data.

Dialog versus Manual

While we were hoping to receive the best data from these event conditions, they occurred infrequently to show noteworthy trends. The Manual event condition averaged just 10 observations per test subject, though Dialog events were more common, averaging 82 events per subject. The Dialog distributions from all four test subjects show tails more heavily pronounced than any of the other distributions by that test subject. For subjects A – D, respectively, the median reaction times during speech recognition are 688, 219, 124.5, and 203.5 ms greater than that of the control data. These data provide the strongest evidence that Jukebox interaction caused more driver distraction than simply driving.

While the number of Dialog observations is sufficient to comment on, the small number of Manual observations prevents us from comparing the data sets in any meaningful way. The Manual data does show higher median and mean values, but this increase is likely attributed to: the small number of observations, an issue with the buttons whereby a button press is not registered if another button is held down, and the test subjects' difficulty in pressing multiple buttons in a short span. The latter two reasons may indicate that this method of measuring distraction while using other buttons is not representative of real conditions. We would like to have seen at least 30 such Manual observations per test subject to effectively comment on trends. For this number to be reached, subjects would need to complete approximately three times as many (nine) Button test runs.

6.1.5 Jukebox Distraction Summary

While looking at the median reaction times provides a better measure for skewed distributions, the tail cannot be ignored. In the instance of driver safety, prolonged distraction is especially important. The longer a driver is in lapse, the higher the likelihood that an accident can occur. A reaction time greater than 2 seconds, for example, could pose more of a hazard while driving than multiple 0.7 second reaction times. By doing additional testing, a trend in the size of the tails may appear.

Our results do not show a trend between buttoned and speech-only use of the Jukebox. More test data would be needed to comment on the existence of a trend. The results show a possible trend that Question commands produce less distraction than other commands. This trend was supported by two subjects and marginally supported by one subject, but opposed by one

subject. This reduction in distraction may be due to the natural language implementation of the questions that requires no translation into a Jukebox-recognizable command. For example, to find out who a song is by, “Who is this by?” or any other common variation is accepted. But because these commands are just features of a complete Jukebox interface, the comparison between different types of commands is of less relevance. The most important piece of information to gather is the possibility that natural language dialog produces less distraction than direct, unnatural commands. Specific dialog testing may support this trend.

We found noticeable added distraction when using our multimodal Jukebox. But this added distraction is expected; this issue at hand is the acceptability of the added distraction. The “worst case” Dialog data of our subjects showed median values 688, 219, 124.5, and 203.5 milliseconds greater than their control reaction times, averaging to 309 ms greater. However these increases in reaction time are seen at the precise moments when the Jukebox is being interacted with. During each test run, the subject was issued about 16 commands. With each test run lasting approximately 4 minutes and 30 seconds, it equates to approximately one Jukebox interaction every 17 seconds. This rapid issuing of commands was done to pronounce affects more in our data but does not accurately represent real world use. As such, average or median distraction times while using the Jukebox would be significantly lower than the worst case values discussed here.

“When other driving or nondriving matters consume the driver's attention, then brake time becomes longer. For example, on a winding road, the driver must attend more to steering the car through the turns. Another major load on attention is the use of in-car displays and cell phones. There is no doubt that both cause delays in reaction times, with estimates ranging from 0.3 to as high a 1 second or more, depending on the circumstances” (Green, 2000). It has been shown that the use of a cell phone when driving increases the reaction time by about 0.4 to 0.5 seconds (Green, 2000). We have found our Jukebox to produce distraction at levels slightly below cell phone usage and well within common sources of distraction according to the “worst case” average.

6.2 SUI Accuracy Analysis

During human subject testing, Jukebox errors were recorded to determine the accuracy of the SUI design. The errors were then examined to determine if trends existed during specific test runs. We investigated the overall errors as well as errors in runs that used button inputs, runs that used only verbal commands, and runs that had primarily one type of Jukebox interaction (playlist navigation vs. questions). We were able determine the overall accuracy of the MP3 Jukebox as well as determine if specific types of interfaces or questions have higher error rates. The song database used during testing included 1707 songs, 148 artists and 137 albums, with a variety of music genres.

Each time the Jukebox did not accurately perform the user request, a Jukebox error was recorded. Jukebox errors were broken down into two categories: SUI errors and user errors. The number of unrecognizable commands was also scored; however they were not considered errors because the Jukebox did not perform inappropriate actions in these cases. SUI errors corresponded to errors that occurred when the user made a clear and acceptable request which was included in the grammar file, but were not correctly recognized. SUI errors counted against the accuracy of the design, because the speech recognizer was capable of identifying the appropriate command, yet did not perform the appropriate action.

User errors are defined as Jukebox errors that occurred when the user gave an unacceptable command not included in the grammar, or when the command was muttered or unclear. Errors of this nature were not counted against the accuracy of the design because the speech recognizer could not be expected to correctly identify these commands.

The last metric was unrecognizable commands, which correspond to clear and acceptable commands which prompt the Jukebox to return, “That command could not be recognized.” The Jukebox returns the aforementioned statement when the speech recognizer does not have a confidence rating over a set threshold. For our design, the speech recognizer must have a confidence rating over 4000 to perform a function; otherwise it will declare that the command could not be recognized (see Background: Speech Recognition). Unrecognizable commands differ from SUI errors because SUI errors return an inappropriate function, while unrecognizable commands do not perform any action. Unrecognizable commands did not count against the accuracy of the design.

6.2.1 Cumulative Error Analysis

At the completion of human subject testing, the four test participants issued a total of 368 commands to the MP3 Jukebox (92 commands per test session). Table 12 summarizes the total number of each error type along with the percentage of each error relative to the total number of commands.

Table 12: Overall SUI Results by Subject

	SUI Errors	User Errors	Unrecognizable Commands
Subject 1	8	4	2
Subject 2	7	1	0
Subject 3	6	6	2
Subject 4	4	7	0
Total	**25**	**18**	**4**
Percentage of Total Commands (368)	6.79%	4.89%	1.09%

These results showed that the Jukebox performed inaccurate functions when given acceptable commands 6.79% of the time. This error rate corresponds to an overall Jukebox accuracy of 93.21%. Additionally, the user error rate was 4.89% after only one day of experience with the Jukebox. The low user error rate demonstrates a relatively short learning period needed to become functional with the design.

6.2.2 Button Interface vs. Non-Button Interface Analysis

Error rates were compared between test runs that required the subject to make use of the steering wheel buttons and runs where only verbal commands were permitted. We hoped to gain insight on the most accurate and user friendly interface by comparing these results. Participants issued 184 total commands using each interface type. Table 13 and Table 14 show the error rates while participants used the button and verbal interfaces respectively.

Table 13: Results on Test Runs Using the Button Interface

	SUI Errors	User Errors	Unrecognizable Commands
Subject 1	3	2	0
Subject 2	4	1	0
Subject 3	2	1	1
Subject 4	2	2	0
Total	**11**	**6**	**1**
Percentage of Total Commands (184)	5.98%	3.26%	0.54%

Table 14: Results on Test Runs Using Only Verbal Commands (No Buttons)

	SUI Errors	User Errors	Unrecognizable Commands
Subject 1	5	2	2
Subject 2	3	0	0
Subject 3	4	5	1
Subject 4	2	5	0
Total	**14**	**12**	**3**
Percentage of Total Commands (184)	7.61%	6.52%	1.63%

As shown in the table above, no large differences can be seen when comparing the SUI errors between each interface. There is a trend in the user error rate however. Participants committed twice as many user errors when using the verbal commands than they did when using the button interface. These results illustrate that users may be more comfortable using the button interface than they were when using all verbal commands.

6.2.3 Interaction Type Analysis

The test runs were designed to contain specific types of interaction with the Jukebox so that problems with specific types of commands could be identified. Test run 1 contained mainly song selection commands as well as playlist navigation (next, pause, previous etc.). Test run 2 included additional song selection commands as well as questions to the Jukebox to prompt feedback regarding a song (name of song, current rating etc.). The last test run consisted of a mix of commands from the other two test runs. Table 15 on the following page shows the total number of errors that occurred during each test run. After examining the results, no obvious trend between SUI errors and the specific type of Jukebox interaction could be identified.

Table 15: Results by Test Run

	SUI Errors	User Errors	Unrecognizable Commands
Test Run 1 (Navigation)	9	7	2
Test Run 2 (Questions)	8	4	2
Test Run 3 (Mix)	8	7	0

6.3 Subjective Feedback

After each subject completed the second testing session, they were asked to fill out a questionnaire to give subjective feedback regarding the functionality of our design as well as the testing methods. The questionnaire they were asked to complete can be seen in Appendix G: Testing Questionnaire. This subsection will highlight the common opinions expressed by the test participants. All four test participants agreed that learning how to use the Jukebox was easy, and did not take an unnecessary amount of training. All subjects were also in agreement that there were sufficient methods for selecting songs from a music database, and that the button interface on the steering wheel was more useful than using the associated verbal commands. Lastly, all four subjects stated that they would use our design in their vehicle; however three out of the four subjects believed that the Jukebox needs additional refinement before they would actually use it.

Three out of the four participants thought that there should be more navigational commands for traversing a playlist. One subject said that they thought it would be beneficial to be able to skip more than one song at a time, while another suggested that the Jukebox have the ability to list more than just the next song in the playlist. Two separate participants thought that the ability to remove songs from a running playlist would enhance the design. Another suggestion regarding Jukebox design was to implement a feature that would read off all of the available albums from the music database if given the artist name.

Lastly, the participants provided feedback relating to the testing environment. Only two out of the four subjects believed that the driving simulator provided a fair representation of their driving capabilities. These two participants thought that it was too difficult to control the speed of the vehicle during testing. A suggestion to fix this problem was to examine the possibility of using a driver's education simulator rather than a video racing game. The subjective feedback provided by the test participants enabled us to formulate some recommendations for future work on this project (see Recommendations).

7 Conclusions

Background research, the design of our Jukebox and the analysis of our test results have led us to four major conclusions. The first major conclusion is that there is a demand for a multimodal music Jukebox inside the automobile. In the past, consumers have been limited to the music played on radio stations within range or cassettes or CDs in their car. All of these methods for listening to music have been limiting. Since the invention of portable music spearheaded by the MP3 format, music has expanded to every environment. Inside the automobile is not an exception, especially as the digital generation becomes older and capable of driving. Car audio decks have appeared that support MP3 CDs, include a USB port for storage devices with MP3s, and have internal hard drives for MP3s. Because consumers have larger libraries to select from, a simple navigation interface consisting of buttons is no longer sufficient. Additionally, attention while driving is imperative and the use of graphical interfaces, such as those found on PCs or portable music players, to select songs would severely impair the ability to drive safely. Gracenote, a leader in media management, has recently released a developer's kit that is designed to accomplish precisely what we have designed. Their release certainly shows the desire for such an interface for music management. The difference between our project and their product is that ours is a complete MP3 player tailored for an automotive application, from speech input to audio output with buttoned or physical control, whereas Gracenote's product provides the speech interface to such a player.

The second major conclusion is that a multimodal interface is best suited for this application. We have designed our speech user interface (SUI) in such a manner that it incorporates natural language as well as direct dialogs. Users have the option of issuing direct commands such as "Next" or complete instructions such as "Play the next song." As the number of features increases, it becomes more difficult to memorize direct commands whereas with natural language dialog users can say what comes to mind naturally with little to no prior knowledge. However, certain keywords were required to produce useable accuracy. For example, to select an artists a user must say "Select artist …" because allowing users to say "Play artist …" often resulted in incorrect recognitions. While over 1,400 variations of

commands are accepted, an accuracy rate over 92% was experienced. Overall, all four test subjects said they would use the Jukebox in its current state though three of the four indicated that more refinement is still needed. Through our analysis of acceptable commands, we have found that there are recognition problems with specific phrases which can be addressed in additional refinement. The data from our distraction testing suggest that the buttoned interface does not increase or decrease driver distraction. While the distraction data does not indicate a trend, the accuracy data does show that using the buttons results in higher accuracy over a speech-only interface. Additionally, the test subjects have indicated a preference to using the buttons over the parallel speech commands.

The third major conclusion is that our Jukebox provides enough song selection methods but should provide additional navigation and information features. All of the test subjects felt that our Jukebox provided enough song selection methods. Additionally, our subjective feedback indicated that the information available about the current song was helpful. The test subjects did indicate that they would like to navigate by more than a single song, indicating that being able to skip albums, artists, and a specific number of songs would be useful. There was also a consensus that more information about the music library would be useful. In our design, the user could ask what the previous, current, and next song is. Ideas such as a listing of upcoming songs in the playlist or browsing the albums or songs available by an artist were proposed.

The fourth major conclusion is that the added distraction our multimodal Jukebox creates is comparable to that of other in-car devices such as cell phones. Any object or event that grasps a driver's attention, by definition, causes distraction ultimately leading to slower reaction times and accidents. In our testing, we developed realistic scenarios of Jukebox commands that were issued at a rigorous pace. Test subjects interpreted the commands issued and performed the appropriate action either via speech or buttoned input to the Jukebox. Subjects were placed at a driving simulator with a steering wheel and pedals and required to drive though a city. While driving and interacting with the Jukebox, subjects responded to randomly appearing PDT targets. This testing environment created situations more taxing than real world conditions, exemplifying the distraction our Jukebox may cause. Additionally, our detailed logging method allowed us to look at when specific events occurred in relation to others. This time stamping allowed us to extract the specific PDT triggers and driving errors that occurred during interaction with the Jukebox.

It has been shown that common driver distractions, such as cell phones and in-car displays, increase the reaction time from 300 milliseconds to 1 second or more. The use of our Jukebox at a pace of one command every 17 seconds exhibits an average increase of 133 milliseconds in reaction time. The "worst case" Dialog data of our subjects showed median values 688, 219, 124.5, and 203.5 milliseconds greater than their control reaction times. It is impossible to create a system designed for a driving environment that will not increase distraction. It is, however, possible to minimize that distraction. By providing drivers with both manual and speech input as well as a natural language interface, we have shown that a full-feature MP3 Jukebox can be created for an automotive environment that exhibits distraction comparable to other in-car devices. The data does not show a correlation between Jukebox interaction and speed or lane violations while driving.

Ultimately, driver safety is the number one concern. With distractions readily abundant on today's road and in today's cars, consumers do not need another one. However, we have designed a Jukebox that produces similar levels of distraction compared to other in-car devices. If a driver selects songs and creates a playlist before leaving the driveway or at stop lights when interaction can be done safely, there is little need for further interaction with the Jukebox. Also, the most common features, such as navigation and song information, are available on the steering wheel. We have found that using the buttons results in higher accuracy which in turn minimizes interaction time, increasing driver safety. Additionally, by investigating and implementing additional song selection methods, such as the intelligent playlist creation provided by Gracenote, users would sacrifice little control. Our suggestions for further design and testing are described in the following Recommendations section.

8 Recommendations

The prevalence of MP3s and other portable music cannot be overlooked. With their incorporation into computers, personal players, and cell phones this entertainment is becoming integrated into everyday life. The automobile is no exception; with expanding music collections and current speech recognition technology providing accurate voice control, there is a market and demand for such a Jukebox – especially one that limits distraction on the driver. We make a few recommendations to anyone who builds on our Jukebox design to provide more and better functionality as well as improved testing. Additionally, the voice control and text-to-speech components of our project are not limited to selecting music. These features could be modified to control other automotive features and can also be taken outside the car cabin and into a person's home or pocket device.

8.1 Additional Jukebox Features

Recommendation 1: That additional novel song selection methods be implemented and tested.

Although we were able to employ a number of new song selection methods, we feel that the design could be greatly enhanced through the implementation of additional Jukebox features. This subsection will discuss some of the features we had originally hoped to implement, yet were unable to due to time restrictions. The first feature we had hoped to integrate into the design was timestamps. The timestamp feature would automatically record when each song had been played. The user could then select music based on when they had played the song. For example, they could ask to hear the most played songs from last month, or to listen to the songs they had played last night after eight o'clock. They could also make ask for songs that are normally played on the way to work, or songs generally listened to on weekends. The integration of a timestamp feature would allow for numerous innovative song selection techniques.

Other interesting song selection features that we considered implementing included keyword searching, lyric matching, similar song/artist/album lookup, and selection based on signal characteristics. Using keyword searching, the user could issue a keyword such as "acoustic" or "heart," and every song with that keyword in its title would be played. Many

random playlists could be created based on the selected keyword. Lyric matching refers to selecting a specific song by singing or saying partial lyrics from the song to the Jukebox. For example, the user could say, "as I wind on down the road" and the Jukebox would queue up "Stairway to Heaven" by "Led Zeppelin." This feature appeared very practical for a voice-controlled Jukebox; however the complexity and time for implementation made it unfeasible to integrate into our design. Similar song/artist/album lookups would enable users to select music similar to what they were currently listening to. For example, they may be listening to a hard rock band and ask to play songs by similar artists. The Jukebox would then create a playlist with songs from artists similar to the last song. Online music classification databases, such as the "Music Genome Project" (Pandora, 2006), could be used to link related music. The last song selection featured we contemplated implementing was selection based on the signal content of the MP3. Classifications based on power distributions may be able to more accurately categorize songs into appropriate genres; however not enough time was available to examine this feature in great depth.

Additional playlist features were also considered for implementation into the Jukebox. One such feature was on-the-fly playlist filtering. This feature would enable users to remove songs from a playlist according to specific criteria. For example, a user may be listening to all songs with a playcount over ten, but they wish to remove all songs from the "rap" genre. A command such as "filter genre rap" could be issued, and the Jukebox would remove all of the rap songs from the playlist. Another interesting playlist feature we examined was a browse feature. This feature would cause the Jukebox to read all of the upcoming songs so the user could become more aware of the contents of the playlist.

Recommendation 2: That the MP3 Jukebox be expanded to read news feeds and emails to the user.

Two features not involving music were also considered for integration into the design. These features were email and news retrieval from the Jukebox. If implemented in an actual automobile, theses features would require a wireless internet connection to download online news feeds and personal emails to the system. Users could then ask to have the Jukebox read them the news or emails. We feel that these features could have greatly enhanced our design;

however we did not examine them in great detail because we felt that they were beyond the scope of this project.

8.2 Recommendations for Testing

Recommendation 3: That an actual driving simulator be used to increase the validity of the results.

Through driver distraction testing, we were successfully able to measure the amount of distraction caused by our design; however in future endeavors, we feel that improvements to the testing suite can be made to increase the validity of the results. The first improvement that can be made is to upgrade the driving simulator. Although an actual driving simulator would have been too expensive for this project, the possibility of using a driver education game rather than a racing video game should be examined. A driver education program would have more realistic driving conditions than a game made for racing.

Recommendation 4: That more participants conduct longer test runs of approximately 15 minutes to provide additional data.

Also, more participants could be tested using longer test runs to increase the amount of data collected. By testing only four subjects, we did not have enough data for convincing statistical analysis. We were only able to point out trends from the data that were collected during testing. We recommend that approximately ten subjects be tested that represent an appropriate sample of the targeted population. The target audience for such a Jukebox is likely individuals between the ages of 16 and 60. Our testing data indicates that three times the current number of test runs would produce enough events to comment on distraction and accuracy effectively. These two testing improvements may be able to provide stronger analysis regarding the amount of driver distraction caused by the Jukebox.

8.3 Recommendations for Software Design

The following recommendations are for the software design and implementation of the MP3 Jukebox.

Recommendation 5: That once Jukebox prototyping and design is complete, the Jukebox be ported into the C/C++ programming language.

Java was an excellent programming language choice because it allows designs to be prototyped very quickly, speeding development. We were able to implement a workable multimodal Jukebox with speech recognition in a few short months. Our testing system was able to handle the Jukebox program and our driving same simultaneously without causing noticeable slowdowns. However our test system hardly resembles that which would be used in an automotive environment. To minimize power and cost, an embedded system would be used that would be much less forgiving with a Java program. For real world automotive use, a C/C++ port would make more efficient use of processing time and require a smaller memory footprint.

Recommendation 6: That additional portable music formats are accepted.

The MP3 format is becoming dated and lacks the digital rights management from in newer formats. We recommend that common formats such as OGG, WMA, and AAC also be accepted by the Jukebox. This update can be done without modifying the underlying speech recognition and interface components. Only the player and file-reader components would need to be updated.

8.3.1 Recommendations for the Interface

Recommendation 7: That additional methods of initiating speech recognition be implemented and tested.

Our speech user interface (SUI) incorporated one method for initiating the recognition cycle via a button press. It is possible to start the process with a specific keyword or phrase, eliminating the need for a button press. By having the Jukebox and recognition engine always be processing input, the engine could recognize something like “Hey Jukebox…” and begin the

actual recognition process only after receiving such a phrase. This feature would be interesting as it would be possible to perform every feature of the Jukebox without a single button press. Another method for speech recognition that we recommend testing is the ability to hold down a "Talk" button for the duration of the speech that the user wants to recognize. This method is especially useful in noisy environments, such as an automobile cabin, where it is difficult for the recognition engine to detect when the user has stopped speaking, delaying the recognition and introducing errors. In real world usage, this interface method may have a significant impact on recognition accuracy.

Recommendation 8: That a study be done to minimize the distraction caused by the placement of the buttons on the steering wheel and maximize the effectiveness of their corresponding functions.

Our testing environment consisted of the Logitech MOMO steering wheel which has six buttons available in the middle of the steering wheel. These buttons allowed us to test the effects of a multimodal design but they did not allow us to test the placement of the buttons as would be found on an actual automotive steering wheel. Additionally, we were limited to six buttons where many new automobiles provide more to a driver's disposal with the addition of switches and knobs. Usability studies should be done to find the optimal placement of the manual inputs and the functions assigned to them. We recommend that such usability studies examine the possibility of assigning multiple commands to the same button. For example, double tapping the "next" button might skip five songs or pressing the "talk" button during recognition could cancel the process.

Recommendation 9: That future Jukebox designs be aware of the text-to-speech (TTS) output of the program.

In our design, a phrase is passed to the TTS function, which returns instantly. This prevents the program from determining characteristics such as when the speech output has completed. It is possible to write the Jukebox program and corresponding TTS code in a way such that the program is aware of what is being said to the user. This feature could also be implemented by stepping through lists as individual calls to the TTS function. This TTS

awareness could be very useful in certain features such as browsing. For example, if a user wanted to browse though all artists that began with the letter 'A' the system could read back those artists to the user. When he hears an artist he would like to play, he could press a "select" button on the steering wheel or say "Play that one."

8.3.2 Recommendations for the Recognizer

Recommendation 10: That optimal recognition settings are found or dynamically assigned to increase accuracy for an automotive environment.

The speech recognition engine has a number of internal settings that affect different characteristics of recognition. When the Jukebox design is complete, it would be beneficial to find the values that maximize recognition speed and accuracy in an automotive environment. Additionally, these settings could be dynamically set according to environmental or user variables.

Recommendation 11: That user learning be implemented in the recognition engine.

The recognition engine has the capability to store and utilize user data, allowing the engine to learn as recognition occurs. This feature, if implemented, could increase recognition accuracy. It may also be possible to restrict Jukebox control to specific individuals. For example, if a mother is driving with kids in the vehicle, the kids would not be able to change songs when a speech recognition cycle has begun. This feature could also be used as a security provision for other automotive controls such as starting the vehicle.

Recommendation 12: That a music specific dictionary is used to supplement the pronunciation data used by the recognition and text-to-speech engines.

The recognition engine is capable of accepting additional dictionaries to correct for pronunciations that are not generated correctly. For example, many music artists have unique pronunciations such as "Frou Frou." By default, it is interpreted as "frow frow" when it is correctly pronounced "froo froo." Additional pronunciations could be provided that accept nicknames and acronyms as well. For example, one could say "Select artist 'Weird Al'" in

addition to “Select artist ‘Weird Al’ Yankovic.” A benefit of the designed Jukebox is that the maker of the recognition engine, Nuance, is also the maker of the text-to-speech engine. The same pronunciation dictionary can be used by the TTS engine so that proper pronunciations are said to the user.

References

Baumann, M., Rösler, D., Jahn, G., & Krems, J.F. (2003). *Assessing driver distraction using Occlusion Method and Peripheral Detection Task*. Quality of Work and Products in Enterprises of the Future (pp. 53-56). Chemnitz, Germany.

Brown, James Dean. *JALT Testing & Evaluation SIG* Newsletter. Vol. 1 No. 1 Apr. 1997. (16 – 18)

Burns, P.C., Olsson, S. (2000). *Measuring Distraction with the Peripheral Detection Task.* Accessed 9/27/2005. Available at www.nrd.nhtsa.dot.gov/departments/nrd-13/driver-distraction/welcome/.htm.

Burns, P.C., Parkes, A., Burton, S., Smithe, R.K. (2002). *How Dangerous is Driving With a Mobile Phone? Benchmarking the Impairment to Alcohol.* Berkshire, United Kingdom.

CNet. *Voice Activated MP3 Players.* Accessed 10/05/05. Available at http://reviews.cnet.com/4566-6490_7-0.html?filter=500891_5414570_500909_5444470_

Gartner, U., Konig, W., Wittig, T. (2002). *Evaluation of Manual vs. Speech Input When Using A Driver Information System in Real Traffic.* Accessed 10/14/2005. Available at http://ppc.uiowa.edu/drivingasse...nt%20Papers/02_Gartner_Wittig.htm.

Gracenote. *Gracenote Products*. Accessed 4/16/06. Available at http://www.gracenote.com/gn_products/

Green, Marc. *Driver Reaction Time.* Accessed 4/2/2006. Available at http://www.visualexpert.com/Resources/reactiontime.html

Green, P.A., Reed, M.P. (1999). *Comparison of Driving Performance On-Road and In Low-Cost Simulator Using a Concurrent Telephone Dialing Task.* Ergonomics, 42, 1015-1037.

Fagerstrom, M., Fichtenberg, N., Karlsson, R. (2001). *How Different Occlusion Intervals Affect Total Shutter Open Time.* Volvo Car Company, Sweden.

Fries, B., Fries, M.(2005). Digital Audio Essentials: A Comprehensive Guide to Creating, Recording, Editing and Sharing Music and Other Audio. Accessed 10/10/2005. Available at http://www.teamcombooks.com/mp3handbook/12.htm.

Hammer, M., Regan, M., Young, K. (2003). *Driver Distraction: A Review of the Literature.* [Electronic Version] Available at http://www.monash.edu.au/muarc/reports/muarc206.html.

Kwon, S. (1999). *Sony Walkman Turns 20 Years old.* Press Release for the 20th Anniversary of the Sony Walkman. Accessed 10/4/2005. Available at http://history.acusd.edu/gen/recording/walkman2.html.

Liberman, P., Sikorski, M., Slonevskiy, S. *MP3 Jukebox With Voice Recognition.* 3/4/2005. WPI. Worcester, MA.

Martens, M.H., Van Winsum, W. (1999). *Measuring Distraction*: *The Peripheral Distraction Task*. Accessed 9/28/05. Available at www.nrd.nhtsa.dot.gov/departments/nrd-13/driver-distraction/welcome/.htm.

Pandora. Accessed 4/11/2006. Available at http://www.pandora.com/mgp.shtml

ScanSoft Incorporated. *VoCon 3200 Embedded Development System*, 03/05. Accessed 4/24/05. Available at ftp://ftp.scansoft.com/pub/embedded/ds_VoCon3200_Embedded.pdf

Sony Corporation (2005). *From a Small Piece of Wood.* Accessed 10/07/2005. Available at http://www.sony.net/Fun/SH/1-21/h1.html

Stutts, J. C., Rodgman, E., Reinfurt, D. (2001). *The Role of Driver Distraction is Traffic Crashes.* Report prepared for AAA Foundation for Traffic Safety, Washington D.C.

Tabachnick, B. G., & Fidell, L. S. (1996). *Using multivariate statistics (3rd ed.)*. New York: Harper Collins.

Vickery, G., Wunch-Vincent, S. (2005). *Working Party on the Information Economy: Digital Broadband Content: Music.* Written for the Organization of Economic Co-Operation and Development. [Electronic Version] Available at http://www.oecd.org/dataoecd/13/2/34995041.pdf

Wikipedia (2005). *Digital Audio Player.* Accessed 10/07/05. Available at http://en.wikipedia.org/wiki/Digital_audio_player.

Wikipedia (2005). *Empeg Car.* Accessed 10/07/05. Available at http://en.wikipedia.org/wiki/Empeg_Car.

Wikipedia (2005). *Flash Memory.* Accessed 10/05/05. Available at http://en.wikipedia.org/wiki/Flash_Memory.

Wikipedia (2005). *iPod.* Accessed 10/05/05. Available at http://en.wikipedia.org/wiki/iPod.

Wikipedia (2005). *MP3.* Accessed 10/05/05. Available at http://en.wikipedia.org/wiki/MP3.

Appendix A: Focus Group Guide

This appendix shows the guide that was followed to conduct the focus group.

Hello, we are a group of ECE students currently working on an MQP to develop an MP3 Jukebox or player for an automotive application, meaning most interaction will be done through speech or audio and buttons on the steering wheel. We are actually expanding upon work done by a project team last year. Last year's project constructed a speech-controlled MP3 player with a very limited speech interface. The focus for our project is to implement innovative methods for song selection.

1. When using PC supported MP3 players such as Winamp or iTunes, what are your favorite methods for selecting songs from your music database?
    a. Keyword
    b. Year
    c. Rating (auto)
    d. Play count
    e. Time of day played
    f. Song title/album/artist/genre
2. Can you think of any selection methods that these programs don't have that you would use if they were available?
    a. Similar artists/songs
    b. Popularity (chart ratings)
    c. Live/studio
3. Some programs can create auto-playlists where playlists are generated based on ratings, play counts, genres, and time of day normally played. Can you think of any auto-playlists that would be beneficial to have in an automotive environment?
    a. To/from work/weekend
    b. Favorites
    c. Highway/rural/city (GPS)
    d. Weather
4. Can you think of any out-of-the-box ideas to verbally select songs from a music database?
    a. Lyrics
    b. Humming
    c. BPM
    d. Mood
5. A major problem associated with a speech controlled MP3 player is that the user is unable to see all of the songs/artists/albums available to them. How do you recommend informing the user what is in their database/playlist?
    a. Next track
    b. Browse feature w/ filter
    c. Display with some songs (in playlist or returned from a search) on dashboard

Appendix B: Focus Group Results

This appendix shows the results from the focus group. All ideas in bold were ultimately implemented into the final design.

Innovative Song Selection Ideas:

Mild:

- Real-time keyword searching
  - Closest match
- **Rating System (auto and user)**
- **Playcount (takes how long you listen to a song into consideration)**
- **Favorites (User playlists)**
- **Year of release**
- Time of day a song is usually played
- Browse by letter
- Intelligent shuffle mode
  - Display shows next songs in queue
  - iTunes party mix
  - Popular mode (playcount, ratings, charts)
  - Take into account what artists have already been played (no consecutive songs, unless requested/preference)
  - "Give me more <artist>" (adjusts what songs will be played on the fly, user preference with some "shuffle", different levels of shuffling)

Wild:

- "Old school" selections (based on when it was last played or decades when it was produced, allow user to specify time frames)
- Play artists that have a recent album released or upcoming album/concert
- Instruments played by artist
- Download radio station (DJ selected) playlist – able to listen to own high quality songs
- Save songs that have been played on the radio
- Humming a song/rhythm
- Search by lyric or chorus
- Tap Fingers and find beat matching
- "Moodlogic"
  - Detect BPM, genre, other criteria (energy?)
  - Determine mood of songs an cater to user
- "Songs I sing to" category selection
- Record what parts of the song the user actually sings along to (and can skip to that part)
- Environmental Factors

- Based on speed on vehicle – “city music” vs. “country driving”
- Notification of safety features – specific songs or audio cues (unbuckled seatbelts, object behind car, etc.)
- Number of people in the car (more popular songs if more people, “no embarrassing songs” mode)
- Kids in the car (kids songs, swear-free mode)

Button Uses:

- Ability to change song ratings on the fly
- Button change to reflect mode (enables different functions for when browsing or playing songs)
  - Next button/fast forward
- LCD touch screen changes with respect to Jukebox mode
- Skip to chorus

General Comments:

- Natural language speech interface (less command driven)
- Limiting control other users can have in car (kids or friends can’t mess with radio, secure user voice mode)

Appendix C: Recruitment Email

This appendix shows the email that was sent out to recruit test participants for driver distraction testing.

Hello,

We are a team of ECE students currently working on an MQP to develop an MP3 player with voice recognition control. We are recruiting subjects to test our design while using a driving simulator to mimic an automotive environment.

Subjects will be asked to complete two separate one hour testing sessions spaced one to two days apart. During the first session, the subject will become familiar with both the voice controlled MP3 Jukebox as well as the driving simulator which consists of a PC racing game, a steering wheel, and pedals. In the second session, the subject will complete 5-10 test runs on the driving simulator while interacting with the Jukebox. Driving performance metrics will be observed during each test run to determine how much driver distraction is caused by the MP3 Jukebox.

If you are selected to participate in the experiment, you will receive $10 per testing session, and no more than $20 total. Each testing session should last approximately one hour. We are looking for two subjects to complete testing next week starting Monday (2/20), and 4 to 5 more subjects to complete testing during the first week of D-term. We will do our best to work around you schedule.

The following criteria are required of all test subjects:

- Fluent in English (no strong accent)
- No physical or visual impairments that could affect the ability to interact with the system

If you are interested in participating in this study, please email me back with the following information:

- Gender
- Age
- Affiliation of WPI (student, faculty, staff)

Your information will be kept confidential and destroyed at the completion of testing if selected, otherwise it will be destroyed immediately. Thank you.

MP3 Jukebox Team
Brett Dickson
Alex White

Appendix D: Test Runs

This appendix shows the individual test runs that were used during driver distraction testing. Each of the three test runs was completed twice; once using the button interface, and once using all verbal commands.

- 7 test runs total
- 1 just driving and PDT (Control)
- 3 x2 (with and without buttons)
    - 1 plain selection (Selection)
    - 1 lots of questions and user assignments (Question)
    - 1 mix of both (Mixed)

Selection:

1. Select songs by an artist
2. Next track
3. Next track
4. Previous track
5. Shuffle remainder of playlist
6. Next track
7. Select a specific album
8. Pause
9. Resume
10. Next track
11. Select a specific song
12. Add another song
13. Next track
14. Play all songs with a rating greater than 3
15. Shuffle the remainder
16. Next track
17. Pause

Command Type Breakdown

5 Selects
0 Questions
0 Commands
12 Navigation/Control
12 Button-Able Commands

Questions:

1. Shuffle an album
2. Find out how many total songs are in the playlist
3. Find the rating of this song
4. Assign this song a rating
5. Next track
6. Find the name of this song
7. Find out what year this song was released
8. Find out what the next song is
9. Next track
10. Find the playcount
11. Reset the playcount
12. Find out how many songs have been played
13. Increase the rating of this song
14. Pause

Command Type Breakdown

1 Selects
7 Questions
3 Commands
3 Navigation/Control
4 Button-Able Commands

Mix:

1. Play my favorite songs
2. Find out how many songs remain in the playlist
3. Next track
4. Find out what the last song was
5. Select all songs by an artist
6. Shuffle remainder of playlist
7. Next track
8. Find the rating of this song
9. Find out what album this is on
10. Assign this song a rating
11. Add an album to this playlist
12. Shuffle the playlist again
13. Next track
14. Find out what this song is
15. Pause

Command Type Breakdown

3 Selects
5 Questions
1 Commands
6 Navigation/Control
7 Button-Able commands

Appendix E: List of Acceptable Commands

This appendix shows the list of all acceptable commands available from the Jukebox. This list was given to each test participant during the first testing session to teach them how to use the Jukebox.

Standard Song Selection

Action	Acceptable Commands
Play by song name	"Select song <song>" "Select track <song>" "Select <song>"
Play by artist	"Select artist <artist>" "Select band <artist>"
Play by album	"Select album <album>"

Adding Songs to End of Playlist

Action	Acceptable Commands
Add song/album/artist to an existing playlist	"Add <song>" "Add artist <artist>" "Add band <artist>" "Add album <album>" "Add CD <album>"

Shuffle

Action	Acceptable Commands
Shuffles the remaining songs in the playlist (Playlist must already be queued up)	"Shuffle" "Shuffle this" "Shuffle songs" "Shuffle the songs" "Shuffle these" "Shuffle these songs" "Shuffle all the songs" "Shuffle all these" "Shuffle all these songs"
Shuffle all the songs by an artist or on an album	"Shuffle artist <artist>" "Shuffle band <artist>" "Shuffle album <album>" "Shuffle CD <album>"

Genre Selection

Action	Acceptable Commands
Select songs in a specific genre	“Play genre <genre>” “Play songs from genre <genre>” “Play all songs from genre <genre>” “Play songs from the genre <genre>” “Play all songs from the genre <genre>”

Standard Commands/Navigation

Action	Acceptable Commands
Pause track	“Stop” “Stop this” “Pause” “Pause this”
Resume track	“Resume” “Resume this” “Start” “Start this” “Play”
Skip to next track	“Next” “Play the next” “Play the next song” “Play the next track” “Go to the next” “Go to the next song” “Go to the next track” “Skip” “Skip this” “Skip this song” “Skip this track” “Skip song” “Skip track”
Skip to previous track	“Last” “Previous” “Play the last” “Play the last song” “Play the last track” “Play the previous” “Play the previous song” “Play the previous track” “Go to the last” “Go to the last song” “Go to the last track” “Go to the previous” “Go to the previous song”

Skip to previous track (cont'd)	"Go to the previous track"
Quit Jukebox program	"quit" "exit" "goodbye"

Ratings

Action	Acceptable Commands
Rate a song (1-5)	"Rate this song (1-5)" "Rate this song a (1-5)" "Give this song a (1-5)" "Give this song a rating of (1-5)" "Assign this song a (1-5)" "Assign this song a rating of (1-5)"
Increase the rating by 1	"Increase rating" "Increase the rating" "Increase rating of this" "Increase rating of this song" "Increase the rating of this" "Increase the rating of this song" "I like this" "I like this song" "I really like this" "I really like this song" "I enjoy this" "I enjoy this song" "I really enjoy this" "I really enjoy this song" "I love this" "I love this song" "I really love this" "I really love this song"
Decrease the rating by 1	"Decrease rating" "Decrease the rating" "Decrease rating of this" "Decrease rating of this song" "Decrease the rating of this" "Decrease the rating of this song" "I don't like this" "I don't like this song" "I really don't like this" "I really don't like this song" "I hate this" "I hate this song" "I really hate this" "I really hate this song" "I dislike this"

Decrease the rating by 1 (cont'd)	"I dislike this song" "I really dislike this" "I really dislike this song" "I can't stand this" "I can't stand this song" "I really can't stand this" "I really can't stand this song"
Play songs rated (1-5)	"Play songs rated (1-5)" "Play all songs rated (1-5)" "Play songs I have rated (1-5)" "Play all songs I have rated (1-5)" "Play songs rated a (1-5)" "Play all songs rated a (1-5)" "Play songs I have rated a (1-5)" "Play all songs I have rated a (1-5)" "Play songs with a (1-5)" "Play all songs with a (1-5)"
Play all unrated songs (0)	"Play songs unrated" "Play all songs unrated" "Play songs that are unrated" "Play all songs that are unrated" "Play songs with a zero" "Play songs with a rating of zero" "Play songs rated zero" "Play all songs with a zero" "Play all songs with a rating of zero" "Play all songs rated zero" "Play unrated" "Play all unrated" "Play unrated song" "Play all unrated song"
Play all songs rated 4 or 5	"Play favorite songs" "Play my favorite songs" "Play favorites" "Play my favorites"
Play all songs rated 5	"Play most favorite songs" "Play best songs" "Play my most favorite songs" "Play my best songs"
Play all songs rated 1 or 2	"Play least favorite songs" "Play worst songs" "Play my least favorite songs" "Play my worst songs"
Play songs rated greater than or equal to #	"Play songs rated # or above" "Play songs rated # or greater" "Play songs rated # or higher"

Play songs rated greater than or equal to # (cont'd)	"Play songs rated # or more" "Play all songs rated # or above" "Play all songs rated # or greater" "Play all songs rated # or higher" "Play all songs rated # or more" "Play songs with a rating of # or above" "Play songs with a rating of # or greater" "Play songs with a rating of # or higher" "Play songs with a rating of # or more" "Play all songs with a rating of # or above" "Play all songs with a rating of # or greater" "Play all songs with a rating of # or higher" "Play all songs with a rating of # or more"
Play songs rated less than or equal to #	"Play songs rated # or less" "Play songs rated # or smaller" "Play songs rated # or lower" "Play songs rated # or below" "Play all songs rated # or less" "Play all songs rated # or smaller" "Play all songs rated # or lower" "Play all songs rated # or below" "Play songs with a rating of # or less" "Play songs with a rating of # or smaller" "Play songs with a rating of # or lower" "Play songs with a rating of # or below" "Play all songs with a rating of # or less" "Play all songs with a rating of # or smaller" "Play all songs with a rating of # or lower" "Play all songs with a rating of # or below"
Play songs rated greater than #	"Play songs rated greater than #" "Play songs rated higher than #" "Play songs rated more than #" "Play songs rated above #" "Play songs rated greater than a #" "Play songs rated higher than a #" "Play songs rated more than a #" "Play songs rated above a #" "Play all songs rated greater than #" "Play all songs rated higher than #" "Play all songs rated more than #" "Play all songs rated above #" "Play all songs rated greater than a #" "Play all songs rated higher than a #" "Play all songs rated more than a #" "Play all songs rated above a #" "Play songs with a rating of greater than #"

Play songs rated greater than # (cont'd)	"Play songs with a rating of higher than #" "Play songs with a rating of more than #" "Play songs with a rating of above #" "Play songs with a rating of greater than a #" "Play songs with a rating of higher than a #" "Play songs with a rating of more than a #" "Play songs with a rating of above a #" "Play all songs with a rating of greater than #" "Play all songs with a rating of higher than #" "Play all songs with a rating of more than #" "Play all songs with a rating of above #" "Play all songs with a rating of greater than a #" "Play all songs with a rating of higher than a #" "Play all songs with a rating of more than a #" "Play all songs with a rating of above a #"
Play songs rated less than #	"Play songs rated less than #" "Play songs rated smaller than #" "Play songs rated lower than #" "Play songs rated less than a #" "Play songs rated smaller than a #" "Play songs rated lower than a #" "Play all songs rated less than #" "Play all songs rated smaller than #" "Play all songs rated lower than #" "Play all songs rated less than a #" "Play all songs rated smaller than a #" "Play all songs rated lower than a #" "Play songs with a rating of less than #" "Play songs with a rating of smaller than #" "Play songs with a rating of lower than #" "Play songs with a rating of less than a #" "Play songs with a rating of smaller than a #" "Play songs with a rating of lower than a #" "Play all songs with a rating of less than #" "Play all songs with a rating of smaller than #" "Play all songs with a rating of lower than #" "Play all songs with a rating of less than a #" "Play all songs with a rating of smaller than a #" "Play all songs with a rating of lower than a #"

Playcount

Action	Acceptable Commands
Play songs with a playcount more than #	"Play songs I have listened to more than # times" "Play songs I have listened to greater than # times" "Play songs I have listened to over # times" "Play songs I have played more than # times"

Play songs with a playcount more than # (cont'd)	"Play songs I have played to greater than # times" "Play songs I have played to over # times" "Play all songs I have listened to more than # times" "Play all songs I have listened to greater than # times" "Play all songs I have listened to over # times" "Play all songs I have played more than # times" "Play all songs I have played to greater than # times" "Play all songs I have played to over # times"
Play songs with a playcount less than #	"Play songs I have listened to less than # times" "Play songs I have played less than # times" "Play all songs I have listened to less than # times" "Play all songs I have played less than # times"
Reset the playcount of a song to zero	"Reset the playcount" "Reset the playcount of this" "Reset the playcount of this song"

Mood/Type

Action	Acceptable Commands
Play songs assigned a(n) Angry Groovy Happy Sad/Depressing Relaxing/Soothing mood	"Play something <mood>" "Play me something <mood>" "I'm in the mood for something <mood>" "Play mood <mood>" "Play songs with a(n) <mood> mood" "Play all songs with a(n) <mood> mood" "Play <mood> music" "Play <mood> songs" "Play some <mood> music" "Play some <mood> songs"
Play songs assigned a Party Rock (out) Dance type	play something I can <type> to play me something I can <type> to play <type> songs play <type> music play some <type> songs play some <type> music

Beats Per Minute (BPM)

Action	Acceptable Commands
Play songs faster than the current one	"Play something faster" "Play something more upbeat" "Play me something faster" "Play me something more upbeat"
Play songs slower than the current one	"Play something slower" "Play me something slower"
Play songs faster than a certain BPM	"Play something faster than # beats per minute" "Play something faster than # BPM"

Play songs faster than a certain BPM (cont'd)	"Play me something faster than # beats per minute" "Play me something faster than # BPM"
Play songs slower than a certain BPM	"Play something slower than # beats per minute" "Play something slower than # BPM" "Play me something slower than # beats per minute" "Play me something slower than # BPM"

Timestamp

Action	Acceptable Commands
Play the # most recent songs played	"Play the last <number> songs played" "Play the last <number> songs I listened to" "Play the last <number> songs I played" "Play the last <number> songs I have listened to" "Play the last <number> songs I have played"
Play the # most recent songs played by a specific artist	"Play the last # songs played by <artist>" "Play the last # songs I listened to by <artist>" "Play the last # songs I played by <artist>" "Play the last # songs I have listened to by <artist>" "Play the last # songs I have played by <artist>"
Play the # most recent songs played on a specific album	"Play the last # songs played off of <album>" "Play the last # songs I listened to off of <album>" "Play the last # songs I played off of <album>" "Play the last # songs I have listened to off of <album>" "Play the last # songs I have played off of <album>" "Play the last # songs played from <album>" "Play the last # songs I listened to from <album>" "Play the last # songs I played from <album>" "Play the last # songs I have listened to from <album>" "Play the last # songs I have played from <album>" "Play the last # songs played on <album>" "Play the last # songs I listened to on <album>" "Play the last # songs I played on <album>" "Play the last # songs I have listened to on <album>" "Play the last # songs I have played on <album>" "Play the last # songs played off of the album <album>" "Play the last # songs I listened to off of the album <album>" "Play the last # songs I played off of the album <album>" "Play the last # songs I have listened to off of the album <album>" "Play the last # songs I have played off of the album <album>" "Play the last # songs played from the album <album>" "Play the last # songs I listened to from the album <album>" "Play the last # songs I played from the album <album>" "Play the last # songs I have listened to from the album <album>" "Play the last # songs I have played from the album <album>"

Play the # most recent songs played on a specific album (cont'd)	"Play the last # songs played on the album <album>" "Play the last # songs I listened to on the album <album>" "Play the last # songs I played on the album <album>" "Play the last # songs I have listened to on the album <album>" "Play the last # songs I have played on the album <album>"

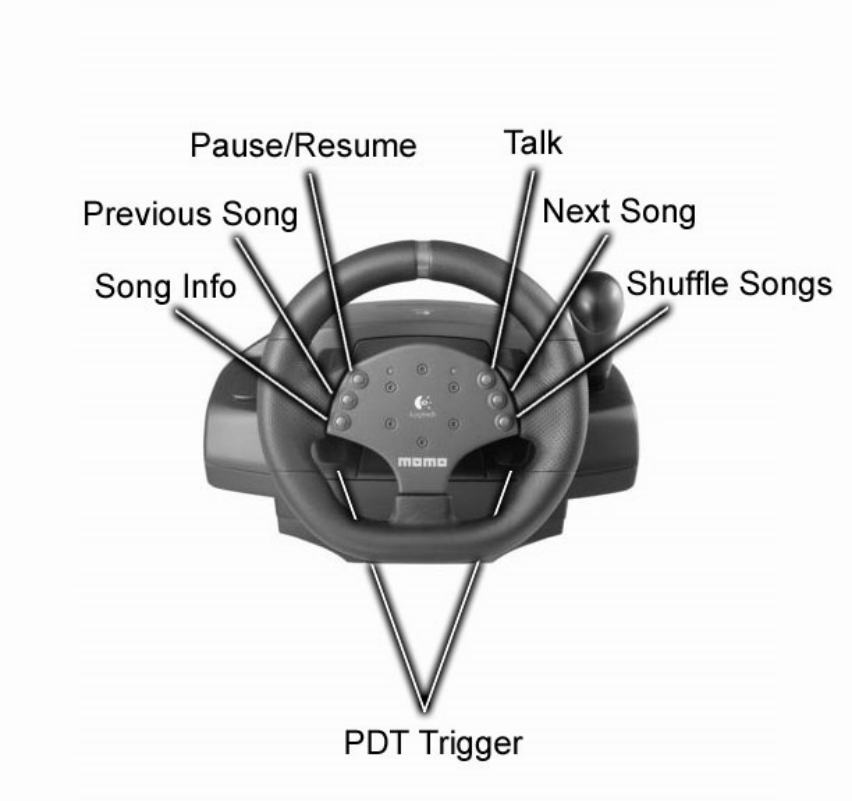
Questions/Information

Action	Acceptable Commands
Returns current song name	"What is this" "What song is this" "What is this song" "What is the name" "What is the name of this" "What is the name of this song"
Returns current song artist	"Who sings this song" "Who wrote this song" "What artist sings this song" "What artist wrote this song" "Which artist sings this song" "Which artist wrote this song" "What band sings this song" "What band wrote this song" "Which band sings this song" "Which band wrote this song" "Who is this by" "Who is this song by" "What artist is this" "What band is this"
Returns current song album	"What album is this off" "What album is this from" "What album is this on" "Which album is this off" "Which album is this from" "Which album is this on" "What album is this song off" "What album is this song from" "What album is this song on" "Which album is this song off" "Which album is this song from" "Which album is this song on"
Returns current song year released	"What year was this released" "What year was this song released" "What year was this album released" "When was this released" "When was this song released" "When was this album released"

Returns current song year released (cont'd)	"When is this from" "When is this song from"
Returns current song rating	"What is the rating of this" "What have I rated this" "What is the rating of this song" "What have I rated this song" "What is this rated" "What is this song rated"
Returns current song playcount	"How many times have I played this" "How many times have I listened to this" "How many times have I played this song" "How many times have I listened to this song" "What is the playcount" "What is the playcount of this" "What is the playcount of this song"
Returns next song name/artist	"What is next" "What is next song" "What is next track" "What is the next" "What is the next song" "What is the next track"
Returns previous song name/artist	"What is last" "What is last song" "What is last track" "What is the last" "What is the last song" "What is the last track" "What is previous" "What is previous song" "What is previous track" "What is the previous" "What is the previous song" "What is the previous track" "What was last" "What was last song" "What was last track" "What was the last" "What was the last song" "What was the last track" "What was previous" "What was previous song" "What was previous track" "What was the previous" "What was the previous song" "What was the previous track"

Returns total number of songs in playlist	“How many are there” “How many are there total” “How many are there in the playlist” “How many are in the playlist” “How many songs are there” “How many songs are there total” “How many songs are there in the playlist” “How many songs are in the playlist”
Returns number of songs remaining in playlist	“How many are queued” “How many are queued up” “How many are there remaining” “How many are there left” “How many songs are queued” “How many songs are queued up” “How many songs are there remaining” “How many songs are there left”
Returns number of song already played in playlist	“How many were played” “How many have I played” “How many have been played” “How many songs were played” “How many songs have I played” “How many songs have been played”

Steering Wheel Buttons



Appendix F: Jukebox Grammar File

```
#BNF+EM V1.0;
!grammar "JukeboxDialog";
!language "American English";

<shuffle>: !action("shuffle") shuffle;
<add>: !action("add") add;
<select>: !action("select") select;
<select2>: !action("select") select | change;
<play>: play;


!start <ignore>;
<ignore>: <...> nevermind;

!start <control>;
<control>: !ignore ( !action("pause") <pause> | !action("resume") <resume> | !action("next")
           <next> | !action("previous") <previous> | !action("quit") <quit> );

<pause>:    (stop | pause) !optional(this);
<resume>:   (resume | start) !optional(this) | play;
<next>:     (!optional(play the | go to the) next | skip !optional(this)) !optional( one |
           song | track);
<previous>: !optional(play the | go to the | skip to the) (last | previous) !optional( song |
           track);
<quit>:     (command quit | command exit | command goodbye);


!start <genre>;
<genre>: !ignore(<play> !optional(!optional(all) !optional(the) songs from !optional(the))
           genre) <genres>;

<genres>: acoustic | alternative !optional (rock) | bluegrass | blues | classic rock |
           classical | club | comedy | country | dance | disco | duet | easy listening |
           electronic | emo | folk !optional (rock) | freestyle | funk | gangsta !optional
           (rap) | gospel | gothic !optional (rock) | grunge | hard rock | hard core | hip
           hop | holiday | instrumental | jazz | latin | !optional (heavy) metal | new age |
           oldies | pop | power ballad | progressive !optional (rock) | psychedelic !optional
           (rock) | punk !optional (rock) | R and B | rap | reggae | religious | Rock | rock
           and roll | screamo | ska | slow jam | slow rock | soft rock | soul | soundtrack |
           southern rock | swing | symphony | techno | top 40;


!start <mood>;
<mood>: !ignore( !action("angry") <angry> | !action("dance") <dance> | !action("groovy")
           <groovy> | !action("happy") <happy> | !action("party") <party> | !action("rock")
           <rock> | !action("sad") <sad> | !action("soothing") <soothing> );

<angry>: play !optional (me) something angry | I'm in the mood for something angry | play
           mood angry | play !optional (all) songs with an angry mood | play !optional (some)
           angry (music | songs);
<groovy>: play !optional (me) something groovy | I'm in the mood for something groovy | play
           mood groovy | play !optional (all) songs with a groovy mood | play !optional
           (some) groovy (music | songs);
<happy>: play !optional (me) something happy | I'm in the mood for something happy | play
           mood happy | play !optional (all) songs with a happy mood | play !optional (some)
           happy (music | songs);
<sad>: play !optional (me) something (sad | depressing) | I'm in the mood for something (sad
           | depressing) | play mood (sad | depressing) | play !optional (all) songs with a
           (sad | depressing) mood | play !optional (some) (sad | depressing) (music |
           songs);
<soothing>: play !optional (me) something (soothing | relaxing) | I'm in the mood for
           something (soothing | relaxing) | play mood (soothing | relaxing) | play !optional
           (all) songs with a (soothing | relaxing) mood | play !optional (some) (soothing |
           relaxing) (music | songs);
```

```
<party>: play !optional (me) something I can party to | play !optional (some) party (songs |
          music);
<rock>: play !optional (me) something I can rock !optional (out) to | play !optional (some)
          rock (songs | music);
<dance>: play !optional (me) something I can dance to | play !optional (some) dance (songs |
          music);


!start <bpm>;
<bpm>: !action("faster") !ignore(<faster>) | !action("slower") !ignore(<slower>) |
          !action("greater") <above> | !action("less") <below>;

<faster>: play !optional (me) something (faster | more upbeat);
<slower>: play !optional (me) something (slower);
<above>: !ignore( play !optional (me) something faster than ) <bpm_numbers> !ignore( beats
          per minute | BPM);
<below>: !ignore( play !optional (me) something slower than ) <bpm_numbers> !ignore( beats
          per minute | BPM);

<bpm_numbers>: !action("25") 25 | !action("50") 50 | !action("75") 75 | !action("100") 100 |
          !action("125") 125 | !action("150") 150 | !action("175") 175 | !action("200") 200;


!start <change_rating>;
<change_rating>: !ignore ( !action("hard") <hard_rate> | !action("cng") <modify_rating>);

<hard_rate>: (rate | give | assign) this !optional(song) !optional(a | a rating of)
          <rating_num>;
<modify_rating>: !action("1") <increase_rating1> | !action("-1") <decrease_rating1> |
          !action("2") <increase_rating2> | !action("-2") <decrease_rating2>;
<increase_rating1>: increase !optional(the) rating !optional(of this !optional(song)) | I
          (like | enjoy) this !optional(song);
<decrease_rating1>: decrease !optional(the) rating !optional(of this !optional(song)) | I
          (don't like | dislike) this !optional(song);
<increase_rating2>: increase !optional(the) rating !optional(of this !optional(song)) by 2 |
          I (really (like | enjoy) | love) this !optional (song);
<decrease_rating2>: decrease !optional(the) rating !optional(of this !optional(song)) by 2 |
          I (really (don't like | dislike) | hate | (can't | cannot) stand) this !optional
          (song);

!start <rating>;
<rating>: !ignore( !action("play") <play_rating> | !action("play_fav") <play_fav> |
          !action("play_most_fav") <play_most_fav> | !action("play_least_fav")
          <play_least_fav> | !action("play_greater_e") <play_greater_e> |
          !action("play_less_e") <play_less_e> | !action("play_greater") <play_greater> |
          !action("play_less") <play_less> );

<play_rating>: play !optional(all) !optional(the) songs ( !optional (I have) rated !optional
          (a) | with a !optional(rating of) ) <rating_num> | play !optional (all) songs with
          a <rating_num> | !action("0") ( play !optional (all) songs that are unrated | play
          !optional(all) unrated !optional(songs)) ;
<play_fav>: play !optional(my) (favorite songs | favorites);
<play_most_fav>: play !optional(my) (most favorite | best) songs;
<play_least_fav>: play !optional(my) (least favorite | worst) songs;
<play_greater_e>: play !optional(all) !optional(the) songs ( !optional (I have) rated
          !optional (a) | with a rating of ) <rating_num> or (above | greater | higher |
          more);
<play_less_e>: play !optional(all) !optional(the) songs ( !optional (I have) rated !optional
          (a) | with a rating of ) <rating_num> or (less | smaller | lower | below);
<play_greater>: play !optional(all) !optional(the) songs ( !optional (I have) rated | with a
          rating ) (greater than| higher than| more than| above) !optional (a) <rating_num>;
<play_less>: play !optional(all) !optional(the) songs ( !optional (I have) rated | with a
          rating ) (less than | smaller than | lower than | below) !optional (a)
          <rating_num>;

<rating_num>: !action("1") 1 | !action("2") 2 | !action("3") 3 | !action("4") 4 |
          !action("5") 5;


!start <playcount>;
```

```
<playcount>: !ignore (!action("most") <most_played> | !action("more") <play_more_than> |
           !action("less") <play_less_than> | !action("reset") <reset_playcount>);

<most_played>: play (my | the) most played !optional (songs);
<play_more_than>: play !optional (all | the | all the | all of the ) songs I have (listened
           to | played) (more than | greater than | over) <number> times;
<play_less_than>: play !optional (all | the | all the | all of the ) songs I have (listened
           to | played) (less | fewer) than <number> times;
<reset_playcount>: (reset | clear) !optional(the) playcount !optional(of this !optional
           (song)) | set ( !optional(this song's) playcount | !optional(the) playcount
           !optional(of this !optional(song)) ) to 0;


!start <timestamp>;
<timestamp>: <last_num> | <last_art> | <last_alb>;

<last_num>: !action("num") !ignore( play the last <number> songs (played | I !optional (have)
           (listened to | played)) );
<last_art>: !action("") !ignore( play the last <number> !action(":") songs (played | I
           !optional (have) (listened to | played)) by <artist> );
<last_alb>: !action("") !ignore( play the last <number> !action(":") songs (played | I
           !optional (have) (listened to | played)) (off of | from | on) !optional (the
           album) <album> );

<number>: !action("1") 1 | !action("2") 2 | !action("3") 3 | !action("4") 4 | !action("5") 5
           | !action("6") 6 | !action("7") 7 | !action("8") 8 | !action("9") 9 |
           !action("10") 10 | !action("11") 11 | !action("12") 12 | !action("13") 13 |
           !action("14") 14 | !action("15") 15 | !action("16") 16 | !action("17") 17 |
           !action("18") 18 | !action("19") 19 | !action("20") 20 | !action("30") 30 |
           !action("40") 40 | !action("50") 50 | !action("60") 60 | !action("70") 70 |
           !action("80") 80 | !action("90") 90 | !action("100") 100;


!start <do_shuffle>;
<do_shuffle>: !ignore (!action("shuffle") shuffle !optional( this | these | !optional(
           !optional(all !optional(of)) the | !optional(all !optional(of)) these ) songs  |
           !optional(the | this) playlist ) );


!start <question>;
<question>: !ignore ( !action("song_name") <song_name> | !action("artist_name") <artist_name>
           | !action("album_name") <album_name> | !action("year_released") <year_released> |
           !action("song_rating") <que_rating> | !action("song_playcount") <que_playcount> |
           !action("next_track") <next_track> | !action("last_track") <last_track> |
           !action("num_played") <num_played> | !action("num_remain") <num_remain> |
           !action("num_total") <num_total>);

<song_name>: what ( !optional(song) is this | is this song | is the (name | title)
           !optional(of this) !optional(song) ) ;
<artist_name>: who (sings | wrote) this !optional(song) | (what | which) (artist | band)
           (sings | wrote) this !optional(song) | who is this !optional(song) by | what
           (artist | band) is this | (what | who) is the artist !optional(of this
           !optional(song));
<album_name>: (what | which) (album | CD) is this !optional (song (off of | from | on));
<year_released>: (when | what year) was this !optional(song | album) (released | recorded) |
           (when | what year) (was | is) this !optional (song) from | (when | what year) did
           this !optional(song) come out;
<que_rating>: (what is the rating of | what have I rated) this !optional (song) | what is
           this !optional(song) rated;
<que_playcount>: how many times has this !optional(song) been played | how many times have I
           (played | listened to) this !optional(song) | (what is | what's) the playcount
           !optional (of this !optional (song));
<next_track>: what is !optional(!optional(the name of) the) next !optional(song | track) |
           what (song | track) is next;
<last_track>: what (is | was) !optional(!optional(the name of) the) (last | previous)
           !optional(song | track) | what (song | track) was (last | previous);
<num_played>: how many !optional(songs) ( were | have (I | been) ) !optional(already) (played
           | listened to) !optional((so | thus) far);
<num_remain>: how many !optional(songs) (!optional(are !optional(there)) (remaining | left) |
           remain ) !optional((in | on) (the | this) playlist);
```

```
<num_total>: how many !optional(songs) !optional(are !optional(there)) !optional(total)
           !optional((in | on) (the | this) playlist);


!start <song>;
<song>: !ignore( (<select> | <add>) !optional(song | track) ( <songs> | <art_song> |
           <song_art> ) ));

<songs>: <songs0> | <songs1> | ... ;

<art_song>: Weird Al Yankovic !optional(by !optional(artist | band | group)) <songs0> | ... ;

<song_art>: <songs0> !optional(by !optional(artist | band | group)) Weird Al Yankovic | ... ;

<songs0>: Amish Paradise !action("0") | ... ;
<songs1>: ... ;
...

!start <artist>;
<artist>: !ignore( (<select> | <shuffle> | <add>) !optional(!optional(all) songs by
           !optional(the) ) (artist | band | group) ( Weird Al Yankovic !action("0") | ...
           ));

!start <album>;
<album>: !ignore( (<select> | <shuffle> | <add>) !optional(!optional(all) songs (on | from |
           off !optional(of)) !optional(the) (album | disc)) (14 59 !action("102") | ... ));
```

Appendix G: Testing Questionnaire

This appendix shows the questionnaire that was given to each test participant after the second testing session. This questionnaire was used to gather subjective feedback regarding the Jukebox design as well as our testing methods.

Jukebox

On a scale from 1 – 5, please circle a rating for each of the following questions

The MP3 Jukebox overall was easy to use:

(Disagree) 1 2 3 4 5 (Agree)

I found the commands to be natural and easy to learn:

(Disagree) 1 2 3 4 5 (Agree)

I felt the Jukebox was accurate enough to use in an automobile:

(Disagree) 1 2 3 4 5 (Agree)

I felt the Jukebox provided enough methods of song selection:

(Disagree) 1 2 3 4 5 (Agree)

I felt the Jukebox provided enough ways to navigate/control song playback:

(Disagree) 1 2 3 4 5 (Agree)

I felt the Jukebox provided enough ways to find out song/music library information:

(Disagree) 1 2 3 4 5 (Agree)

I felt the buttons on the steering wheel were useful:

(Disagree) 1 2 3 4 5 (Agree)

I felt the buttons on the steering wheel were more useful than the associated verbal commands:

(Disagree) 1 2 3 4 5 (Agree)

I would use the Jukebox in my automobile:

(Disagree) 1 2 3 4 5 (Agree)

I feel the Jukebox needs additional refinement before I would actually use it:

(Disagree) 1 2 3 4 5 (Agree)

Additional Questions:

Looking back on all of the available Jukebox actions, are there any verbal commands associated with specific actions that you feel we left out?

Can you think of any additional features or song selections methods that you would use in the MP3 Jukebox?

Are there any other button configuration or uses that you can think of that would make it easier to interact with the Jukebox?

Where do you feel future effort should be put regarding verbal commands (examples)?

Song selection (ratings/genre etc):

Navigation (next/pause etc):

Questions (album info/next song etc):

Driving Simulator/Environment

I found the driving simulator to be a fair representation of my driving capability:

(Disagree) 1 2 3 4 5 (Agree)

I found driving while responding to the PDT targets taxing:

(Disagree) 1 2 3 4 5 (Agree)

While driving and responding to PDT targets, I found additional Jukebox interaction more distracting:

(Disagree) 1 2 3 4 5 (Agree)

Do you feel the driving restrictions (speed between 25 and 45 mph; stay in single lane) are too demanding? Or do you feel they are fairly simple to follow if not distracted by additional tasks?

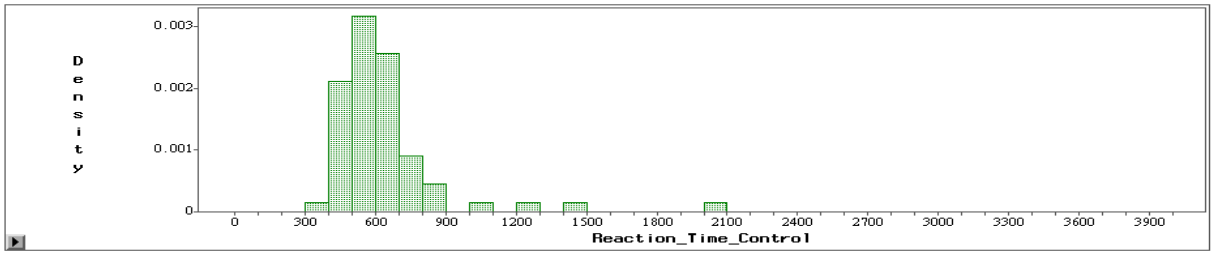
In what ways could the testing environment or driving simulator be improved to more accurate simulate real world conditions?

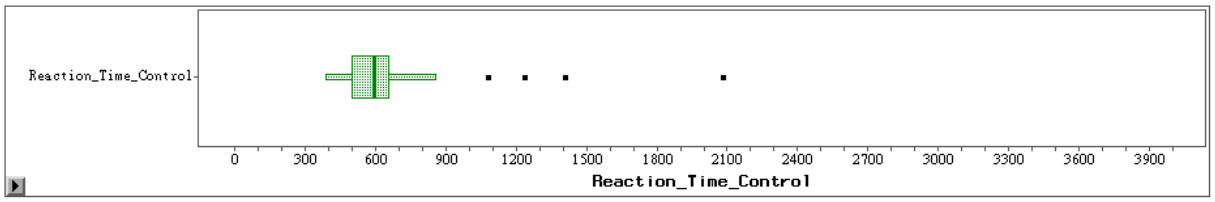
Are there any additional comments you have for us regarding our MP3 Jukebox or testing environment designs?

Appendix H: Subject A Test Results

This appendix shows the PDT reaction time test results for Subject A.

Control vs. Overall

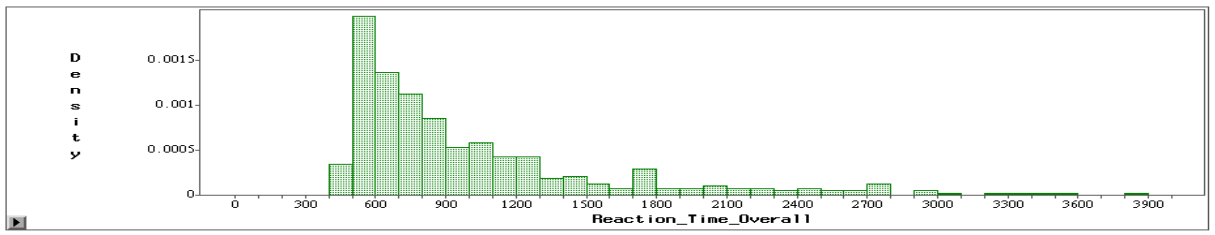


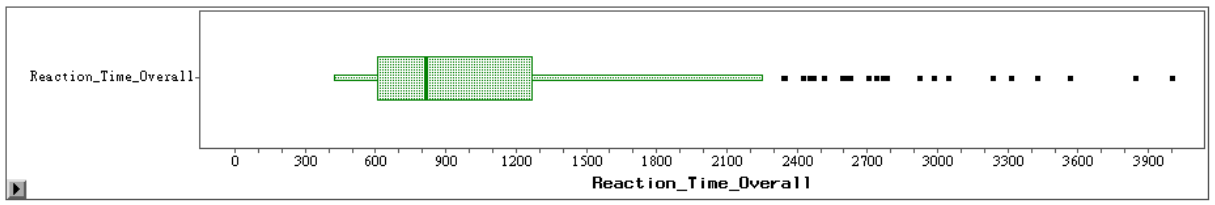


Moments			
N	66.0000	Sum Wgts	66.0000
Mean	632.0909	Sum	41718.0000
Std Dev	253.9220	Variance	64476.3608
Skewness	3.5991	Kurtosis	16.8591
USS	30560532.0	CSS	4190963.45
CV	40.1717	Std Mean	31.2556

Quantiles			
100% Max	2078.0000	99.0%	2078.0000
75% Q3	656.0000	97.5%	1406.0000
50% Med	593.0000	95.0%	1078.0000
25% Q1	500.0000	90.0%	813.0000
0% Min	391.0000	10.0%	453.0000
Range	1687.0000	5.0%	437.0000
Q3-Q1	156.0000	2.5%	406.0000
Mode	469.0000	1.0%	391.0000

Subject A PDT reaction times for the control test – no Jukebox interaction



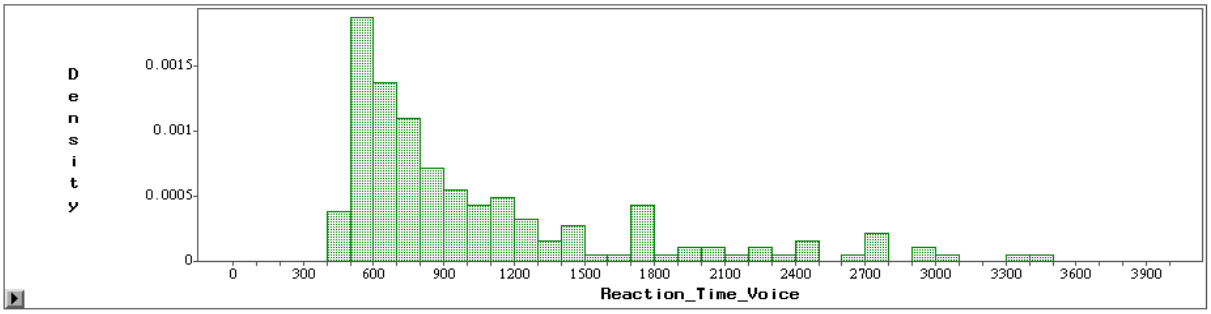


Moments			
N	373.0000	Sum Wgts	373.0000
Mean	1145.3217	Sum	427205.000
Std Dev	842.9345	Variance	710538.504
Skewness	2.0848	Kurtosis	3.8985
USS	753607487	CSS	264320323
CV	73.5981	Std Mean	43.6455

Quantiles			
100% Max	4000.0000	99.0%	4000.0000
75% Q3	1266.0000	97.5%	4000.0000
50% Med	813.0000	95.0%	3313.0000
25% Q1	610.0000	90.0%	2250.0000
0% Min	422.0000	10.0%	547.0000
Range	3578.0000	5.0%	515.0000
Q3-Q1	656.0000	2.5%	484.0000
Mode	4000.0000	1.0%	453.0000

Subject A PDT reaction times for all tests with Jukebox interaction

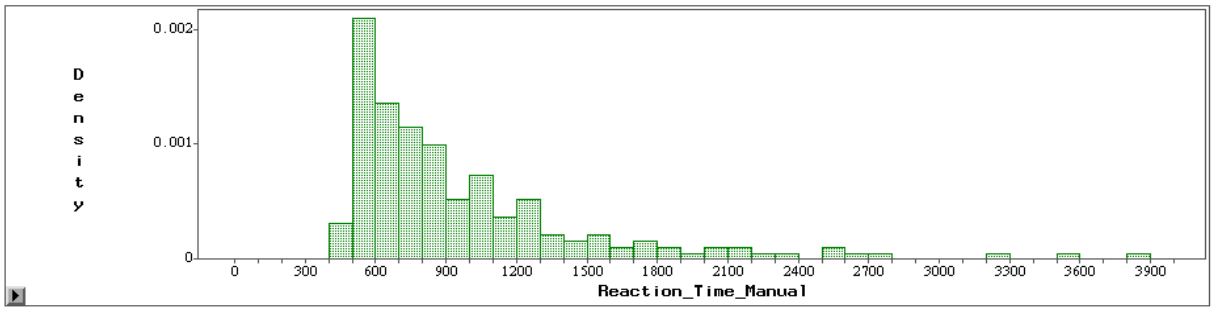
Speech vs. Button



Moments			
N	182.0000	Sum Wgts	182.0000
Mean	1230.7802	Sum	224002.000
Std Dev	929.7777	Variance	864486.570
Skewness	1.7798	Kurtosis	2.3852
USS	432169300	CSS	156472069
CV	75.5438	Std Mean	68.9197

Quantiles			
100%	Max	4000.0000	99.0% 4000.0000
75%	Q3	1453.0000	97.5% 4000.0000
50%	Med	836.5000	95.0% 4000.0000
25%	Q1	625.0000	90.0% 2735.0000
0%	Min	422.0000	10.0% 531.0000
	Range	3578.0000	5.0% 500.0000
	Q3-Q1	828.0000	2.5% 484.0000
	Mode	4000.0000	1.0% 453.0000

Subject A PDT reaction times for tests using speech input only

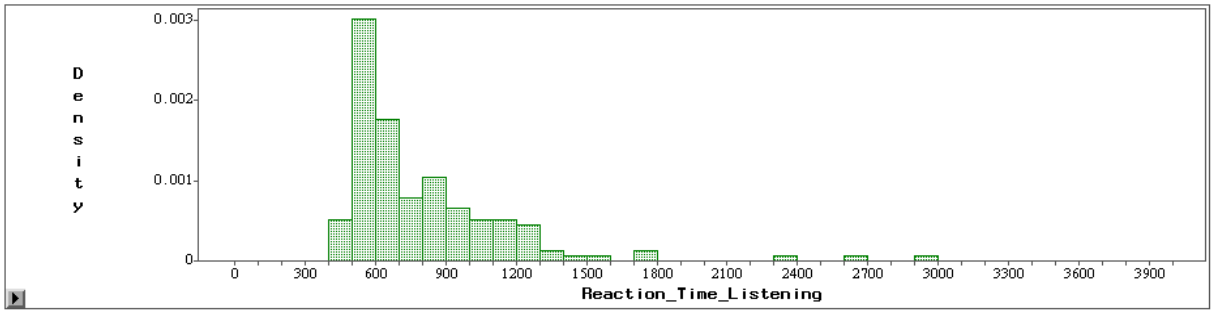


Moments			
N	191.0000	Sum Wgts	191.0000
Mean	1063.8901	Sum	203203.000
Std Dev	744.2854	Variance	553960.719
Skewness	2.4706	Kurtosis	6.3668
USS	321438187	CSS	105252537
CV	69.9589	Std Mean	53.8546

Quantiles			
100%	Max	4000.0000	99.0% 4000.0000
75%	Q3	1187.0000	97.5% 4000.0000
50%	Med	812.0000	95.0% 2610.0000
25%	Q1	609.0000	90.0% 1891.0000
0%	Min	438.0000	10.0% 547.0000
	Range	3562.0000	5.0% 516.0000
	Q3-Q1	578.0000	2.5% 469.0000
	Mode	625.0000	1.0% 453.0000

Subject A PDT reaction times for tests using buttoned input when possible

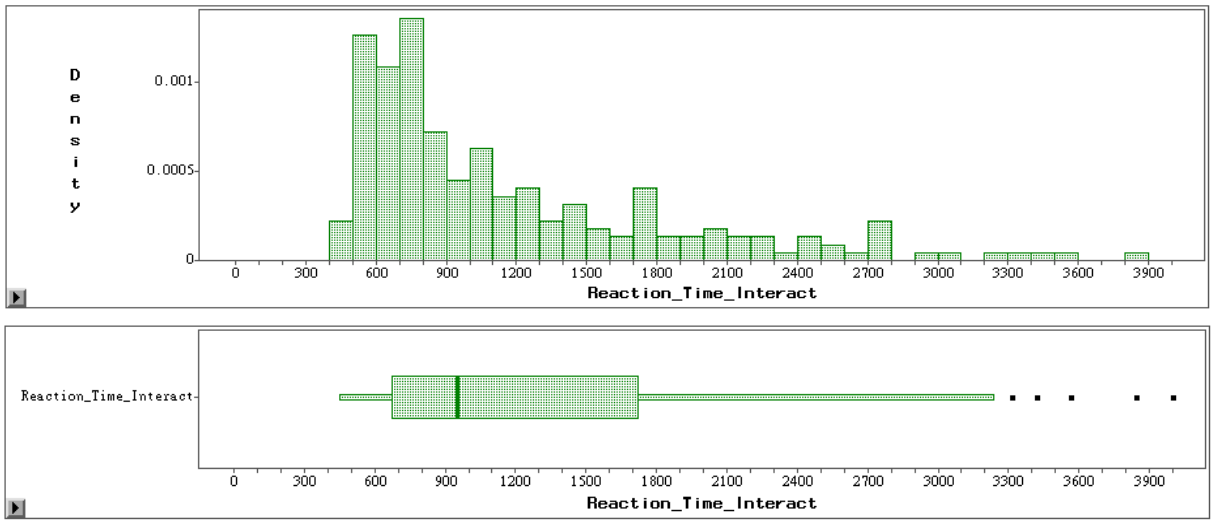
Listening vs. Interaction



Moments			
N	152.0000	Sum Wgts	152.0000
Mean	831.8816	Sum	126446.000
Std Dev	454.3838	Variance	206464.622
Skewness	3.7640	Kurtosis	19.5985
USS	136364256	CSS	31176157.9
CV	54.6212	Std Mean	36.8554

Quantiles			
100% Max	4000.0000	99.0%	2985.0000
75% Q3	937.5000	97.5%	2343.0000
50% Med	672.0000	95.0%	1485.0000
25% Q1	578.0000	90.0%	1203.0000
0% Min	422.0000	10.0%	532.0000
Range	3578.0000	5.0%	485.0000
Q3-Q1	359.5000	2.5%	453.0000
Mode	578.0000	1.0%	438.0000

Subject A PDT reaction times while only music is playing – no active interaction

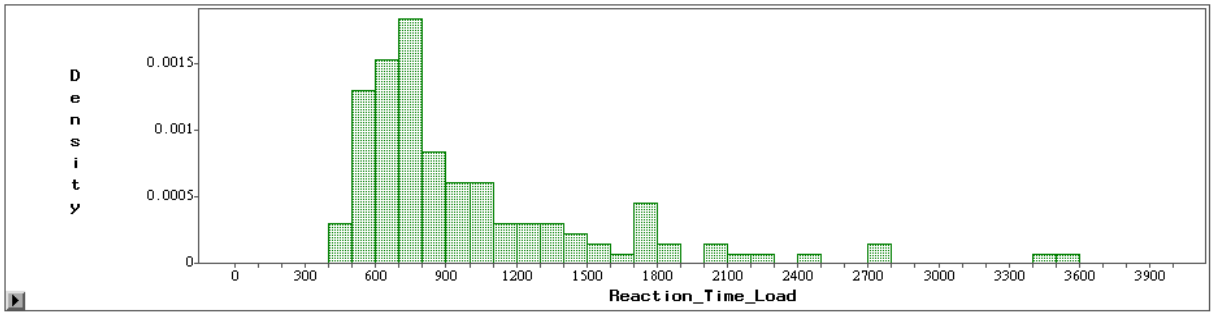


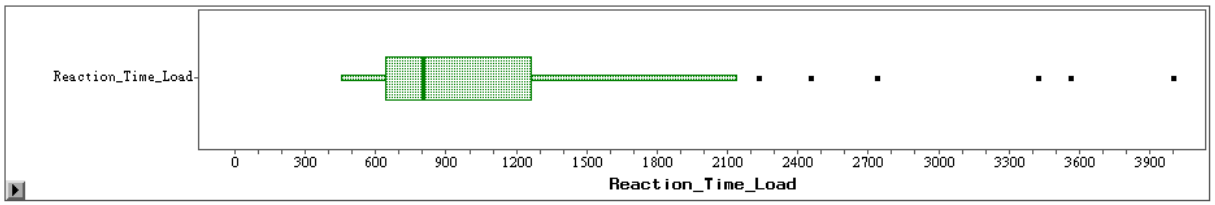
Moments			
N	221.0000	Sum Wgts	221.0000
Mean	1360.9005	Sum	300759.000
Std Dev	972.2050	Variance	945182.599
Skewness	1.5310	Kurtosis	1.4679
USS	617243231	CSS	207940172
CV	71.4384	Std Mean	65.3976

Quantiles			
100% Max	4000.0000	99.0%	4000.0000
75% Q3	1719.0000	97.5%	4000.0000
50% Med	953.0000	95.0%	4000.0000
25% Q1	672.0000	90.0%	2766.0000
0% Min	453.0000	10.0%	562.0000
Range	3547.0000	5.0%	516.0000
Q3-Q1	1047.0000	2.5%	500.0000
Mode	4000.0000	1.0%	469.0000

Subject A PDT reaction times during any type of Jukebox interaction – cognitive load, speech, and button

Cognitive Load vs. Command Issue

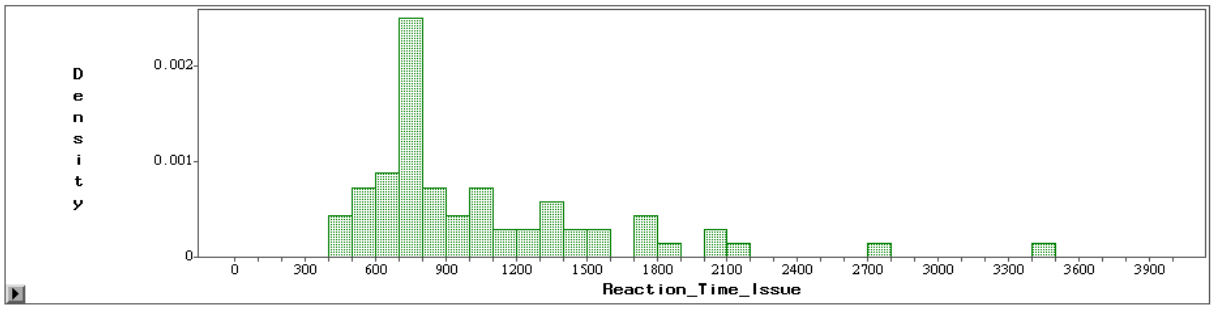


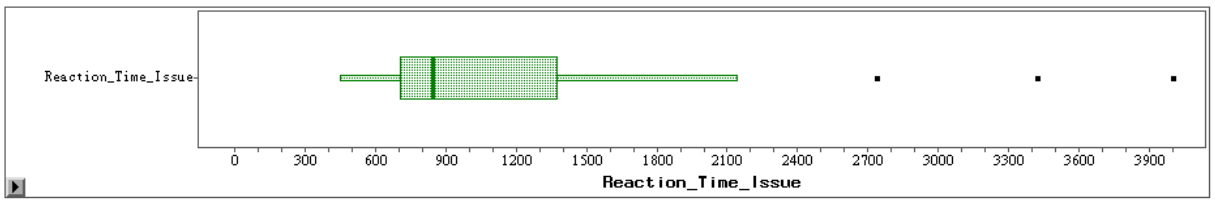


Moments			
N	130.0000	Sum Wgts	130.0000
Mean	1085.8769	Sum	141164.000
Std Dev	726.2288	Variance	527408.279
Skewness	2.3669	Kurtosis	6.0413
USS	221322398	CSS	68035668.0
CV	66.8795	Std Mean	63.6945

Quantiles			
100%	Max	4000.0000	99.0% 4000.0000
75%	Q3	1265.0000	97.5% 3563.0000
50%	Med	804.5000	95.0% 2735.0000
25%	Q1	641.0000	90.0% 1844.0000
0%	Min	453.0000	10.0% 562.0000
	Range	3547.0000	5.0% 515.0000
	Q3-Q1	624.0000	2.5% 484.0000
	Mode	625.0000	1.0% 468.0000

Subject A PDT reaction times during cognitive load conditions



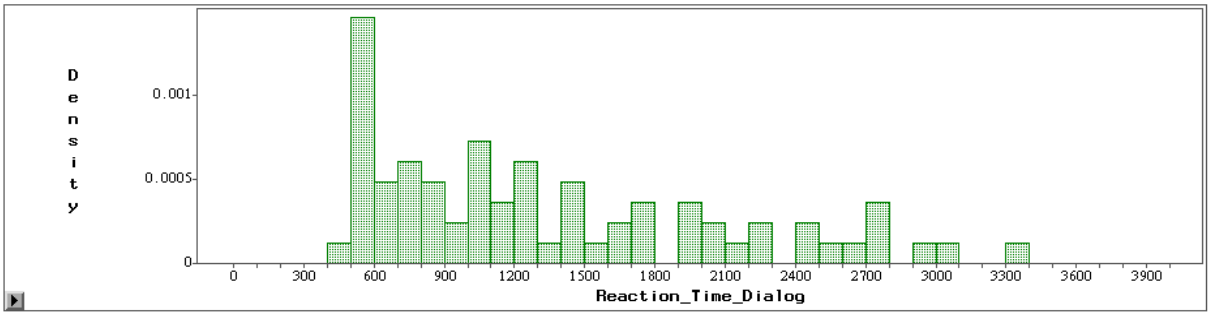


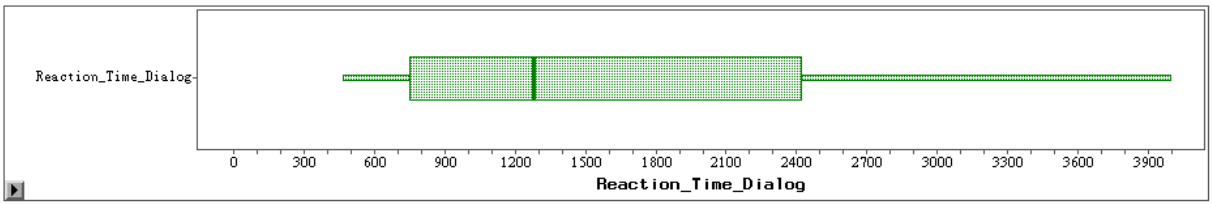
Moments			
N	68.0000	Sum Wgts	68.0000
Mean	1189.7500	Sum	80903.0000
Std Dev	818.7344	Variance	670325.951
Skewness	2.2278	Kurtosis	4.9600
USS	141166183	CSS	44911838.7
CV	68.8157	Std Mean	99.2861

Quantiles			
100%	Max	4000.0000	99.0% 4000.0000
75%	Q3	1375.0000	97.5% 4000.0000
50%	Med	844.0000	95.0% 3422.0000
25%	Q1	704.0000	90.0% 2078.0000
0%	Min	453.0000	10.0% 593.0000
	Range	3547.0000	5.0% 562.0000
	Q3-Q1	671.0000	2.5% 468.0000
	Mode	704.0000	1.0% 453.0000

Subject A PDT reaction times while the tester issues commands

Dialog vs. Manual

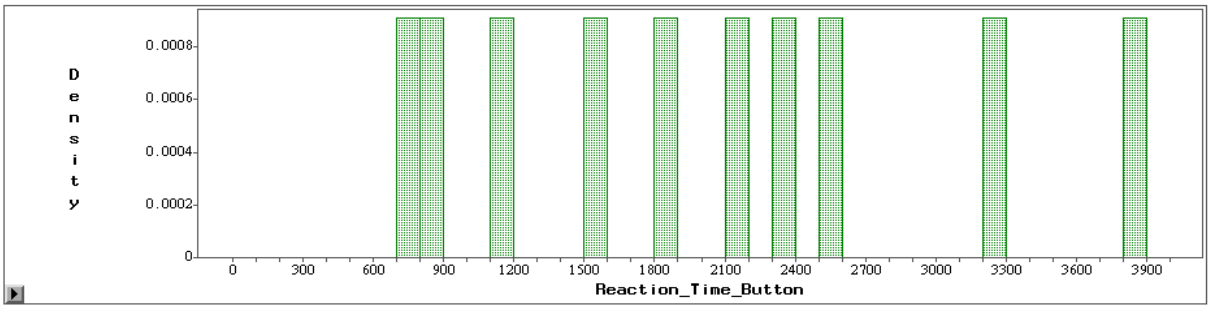


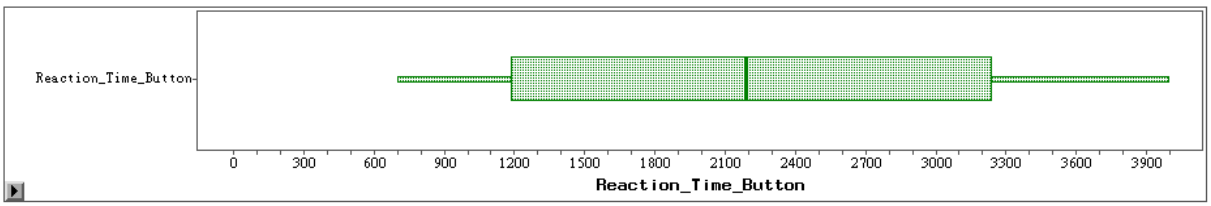


Moments			
N	82.0000	Sum Wgts	82.0000
Mean	1703.9390	Sum	139723.000
Std Dev	1150.2603	Variance	1323098.85
Skewness	0.9253	Kurtosis	-0.3656
USS	345250479	CSS	107171007
CV	67.5060	Std Mean	127.0250

Quantiles			
100%	Max	4000.0000	99.0% 4000.0000
75%	Q3	2422.0000	97.5% 4000.0000
50%	Med	1281.0000	95.0% 4000.0000
25%	Q1	750.0000	90.0% 4000.0000
0%	Min	469.0000	10.0% 547.0000
	Range	3531.0000	5.0% 531.0000
	Q3-Q1	1672.0000	2.5% 516.0000
	Mode	4000.0000	1.0% 469.0000

Subject A PDT reaction times during any Jukebox interaction with speech



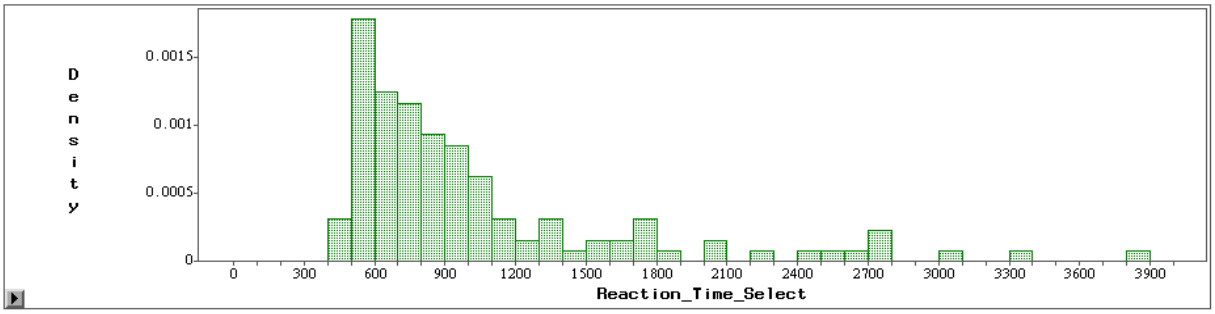


Moments			
N	11.0000	Sum Wgts	11.0000
Mean	2221.3636	Sum	24435.0000
Std Dev	1127.3111	Variance	1270830.25
Skewness	0.3040	Kurtosis	-0.9806
USS	66987323.0	CSS	12708302.5
CV	50.7486	Std Mean	339.8971

Quantiles			
100%	Max	4000.0000	99.0% 4000.0000
75%	Q3	3234.0000	97.5% 4000.0000
50%	Med	2188.0000	95.0% 4000.0000
25%	Q1	1187.0000	90.0% 3843.0000
0%	Min	703.0000	10.0% 859.0000
	Range	3297.0000	5.0% 703.0000
	Q3-Q1	2047.0000	2.5% 703.0000
	Mode	.	1.0% 703.0000

Subject A PDT reaction times during any Jukebox interaction with steering wheel button use

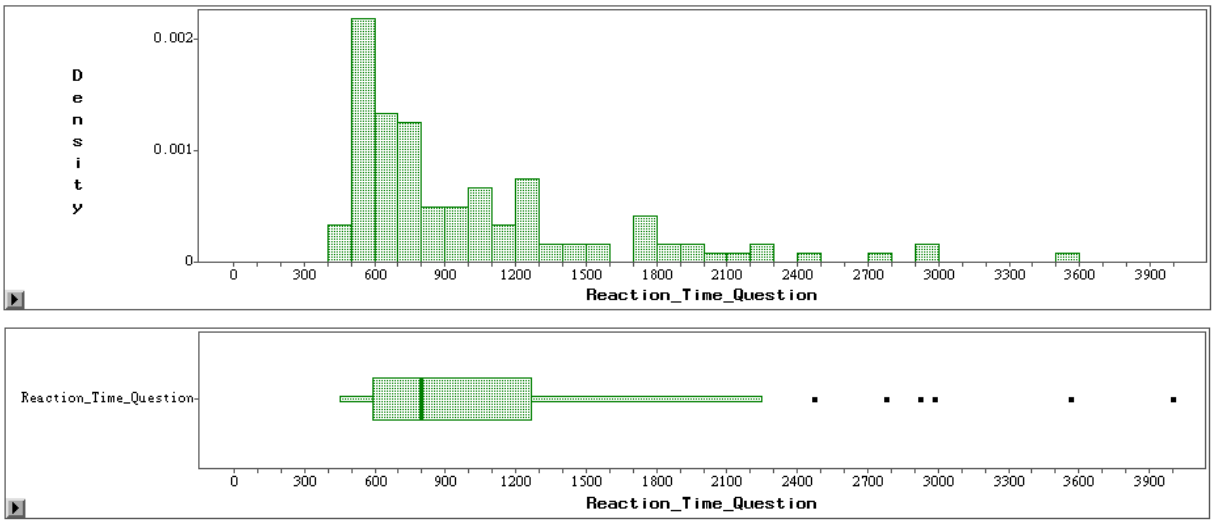
Selection vs. Question vs. Mixed



Moments			
N	128.0000	Sum Wgts	128.0000
Mean	1204.4609	Sum	154171.000
Std Dev	930.7768	Variance	866345.432
Skewness	1.9390	Kurtosis	2.9202
USS	295718817	CSS	110025870
CV	77.2775	Std Mean	82.2698

Quantiles			
100%	Max	4000.0000	99.0% 4000.0000
75%	Q3	1312.0000	97.5% 4000.0000
50%	Med	851.5000	95.0% 4000.0000
25%	Q1	625.0000	90.0% 2703.0000
0%	Min	438.0000	10.0% 532.0000
	Range	3562.0000	5.0% 500.0000
	Q3-Q1	687.0000	2.5% 469.0000
	Mode	4000.0000	1.0% 453.0000

Subject A PDT reaction times for tests with Jukebox interaction using plain selection

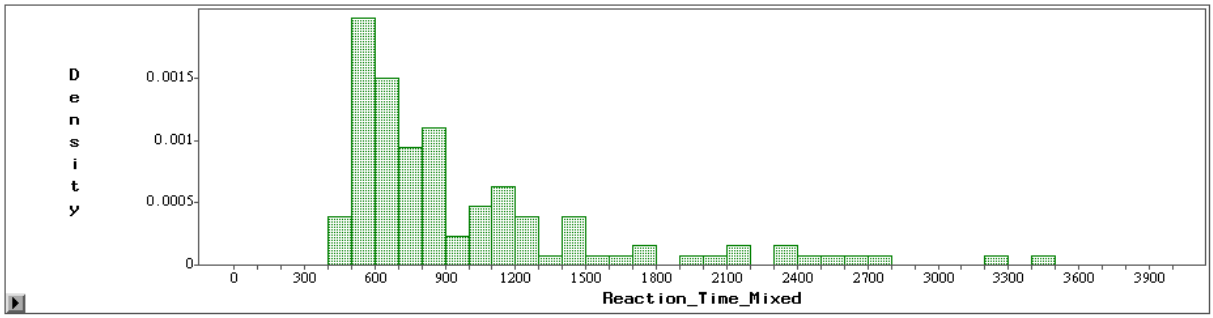


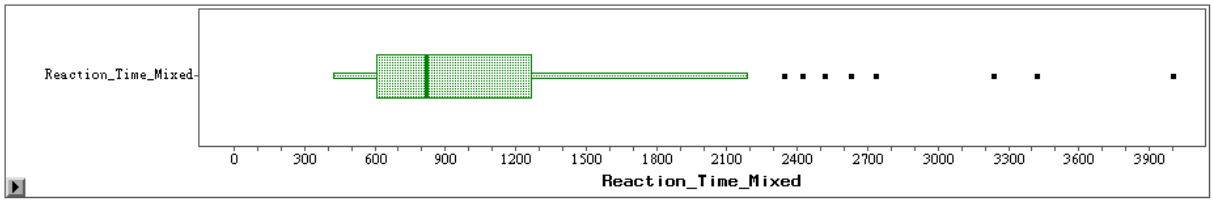
Moments			
N	119.0000	Sum Wgts	119.0000
Mean	1054.5042	Sum	125486.000
Std Dev	656.8222	Variance	431415.439
Skewness	2.0622	Kurtosis	4.8927
USS	183232536	CSS	50907021.7
CV	62.2873	Std Mean	60.2108

Quantiles			
100%	Max	4000.0000	99.0% 3563.0000
75%	Q3	1266.0000	97.5% 2985.0000
50%	Med	797.0000	95.0% 2469.0000
25%	Q1	594.0000	90.0% 1969.0000
0%	Min	453.0000	10.0% 547.0000
	Range	3547.0000	5.0% 516.0000
	Q3-Q1	672.0000	2.5% 484.0000
	Mode	594.0000	1.0% 469.0000

Subject A PDT reaction times for tests with Jukebox interaction using questions and user assignments

Selection vs. Question vs. Mixed (cont'd)





Moments

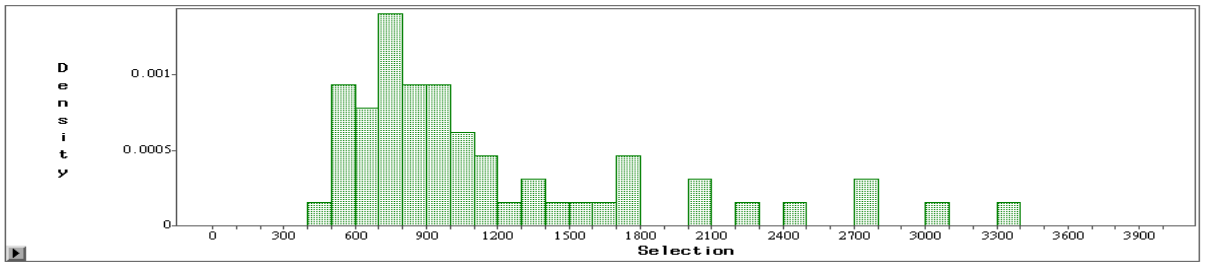
N	126.0000	Sum Wgts	126.0000
Mean	1171.0159	Sum	147548.000
Std Dev	902.7739	Variance	815000.672
Skewness	2.0474	Kurtosis	3.5262
USS	274656134	CSS	101875084
CV	77.0932	Std Mean	80.4255

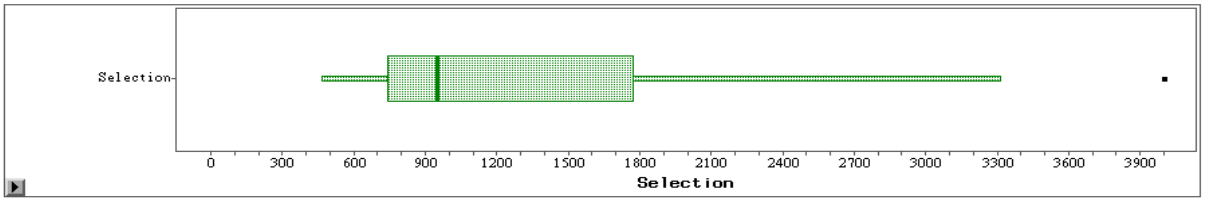
Quantiles

100% Max	4000.0000	99.0%	4000.0000
75% Q3	1265.0000	97.5%	4000.0000
50% Med	820.5000	95.0%	4000.0000
25% Q1	609.0000	90.0%	2422.0000
0% Min	422.0000	10.0%	531.0000
Range	3578.0000	5.0%	516.0000
Q3-Q1	656.0000	2.5%	484.0000
Mode	4000.0000	1.0%	453.0000

Subject A PDT reaction times for tests with Jukebox interaction using a mix of questions and selection methods

Selection vs. Selection w/ Buttons

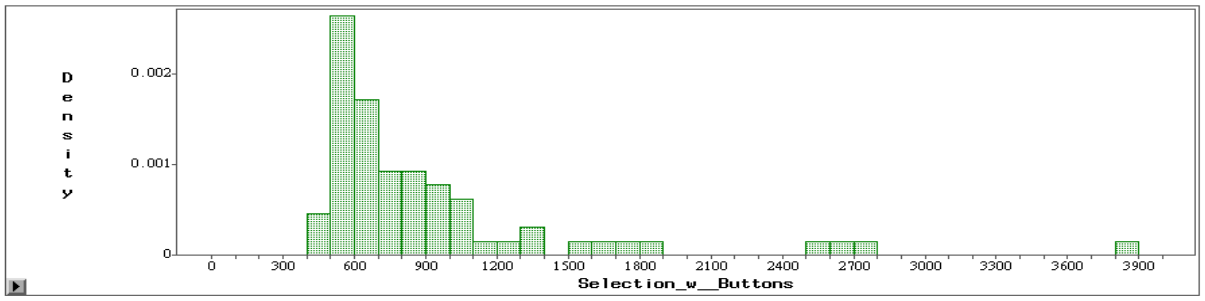


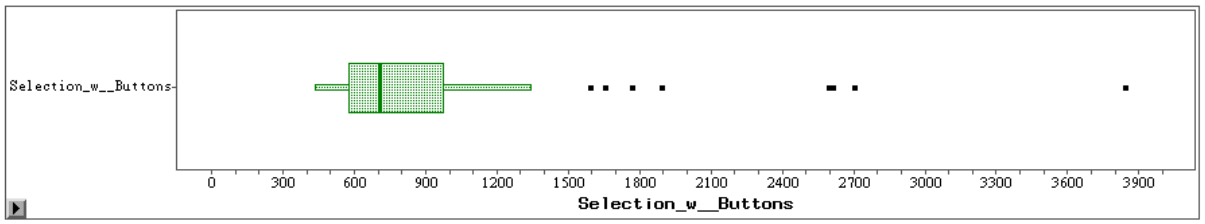


Moments			
N	64.0000	Sum Wgts	64.0000
Mean	1477.7813	Sum	94578.0000
Std Dev	1097.1418	Variance	1203720.21
Skewness	1.4046	Kurtosis	0.7022
USS	215599968	CSS	75834372.9
CV	74.2425	Std Mean	137.1427

Quantiles			
100%	Max	4000.0000	99.0% 4000.0000
75%	Q3	1773.5000	97.5% 4000.0000
50%	Med	953.0000	95.0% 4000.0000
25%	Q1	742.5000	90.0% 4000.0000
0%	Min	469.0000	10.0% 593.0000
	Range	3531.0000	5.0% 531.0000
	Q3-Q1	1031.0000	2.5% 500.0000
	Mode	4000.0000	1.0% 469.0000

Subject A PDT reaction times for the "selection" test – speech only



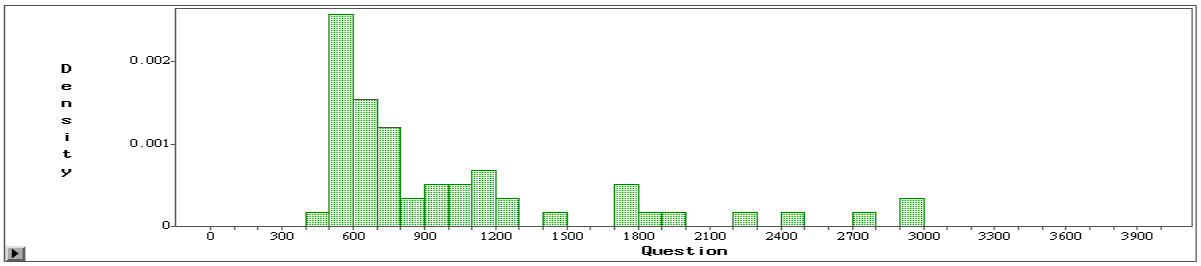


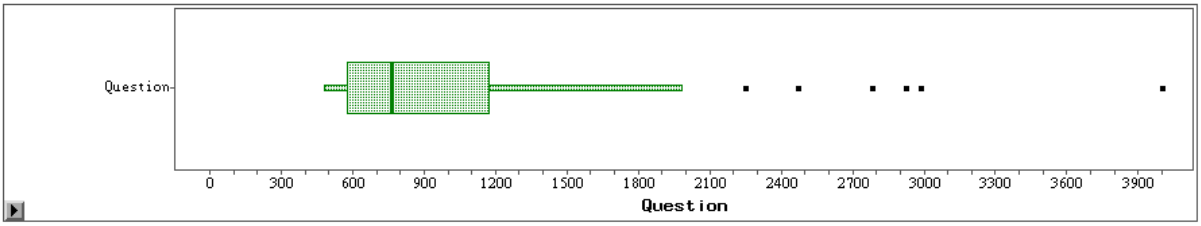
Moments			
N	64.0000	Sum Wgts	64.0000
Mean	931.1406	Sum	59593.0000
Std Dev	625.2541	Variance	390942.631
Skewness	2.6993	Kurtosis	8.2515
USS	80118849.0	CSS	24629385.7
CV	67.1493	Std Mean	78.1568

Quantiles			
100%	Max	3843.0000	99.0% 3843.0000
75%	Q3	976.5000	97.5% 2703.0000
50%	Med	711.0000	95.0% 2593.0000
25%	Q1	578.0000	90.0% 1656.0000
0%	Min	438.0000	10.0% 531.0000
	Range	3405.0000	5.0% 500.0000
	Q3-Q1	398.5000	2.5% 453.0000
	Mode	625.0000	1.0% 438.0000

Subject A PDT reaction times for the "selection" test – buttons when possible

Question vs. Question w/ Buttons

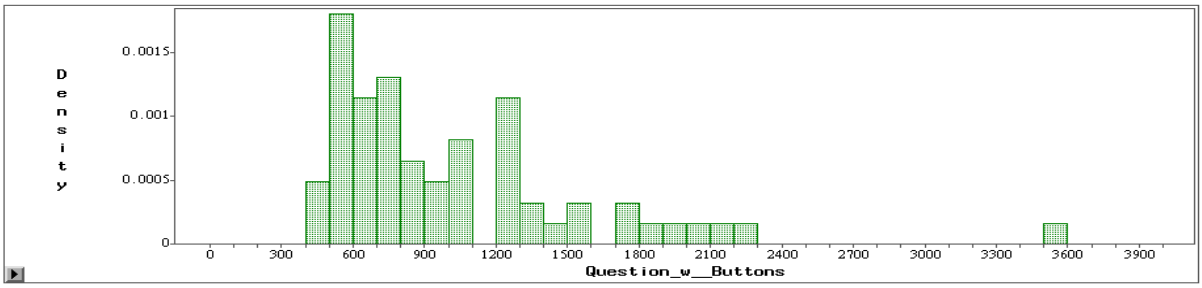


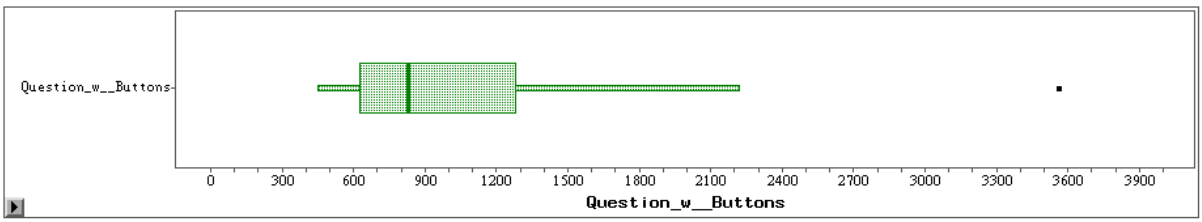


Moments			
N	58.0000	Sum Wgts	58.0000
Mean	1076.2241	Sum	62421.0000
Std Dev	746.7517	Variance	557638.107
Skewness	2.0164	Kurtosis	4.0665
USS	98964359.0	CSS	31785372.1
CV	69.3863	Std Mean	98.0533

Quantiles			
100%	Max	4000.0000	99.0% 4000.0000
75%	Q3	1172.0000	97.5% 2985.0000
50%	Med	765.0000	95.0% 2922.0000
25%	Q1	578.0000	90.0% 2250.0000
0%	Min	484.0000	10.0% 547.0000
	Range	3516.0000	5.0% 516.0000
	Q3-Q1	594.0000	2.5% 500.0000
	Mode	547.0000	1.0% 484.0000

Subject A PDT reaction times for the "question" test – speech only



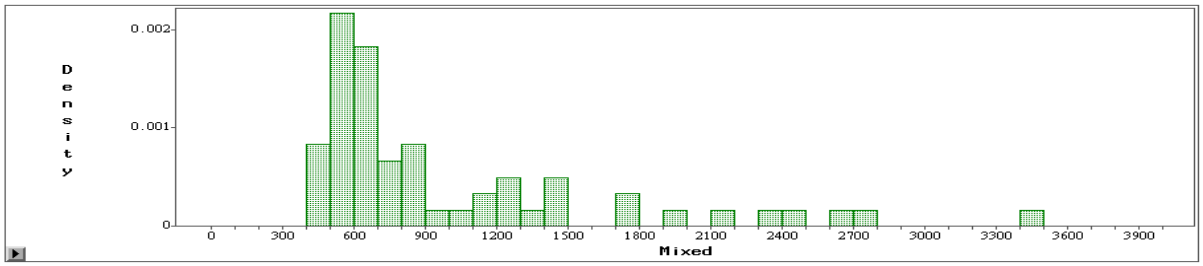


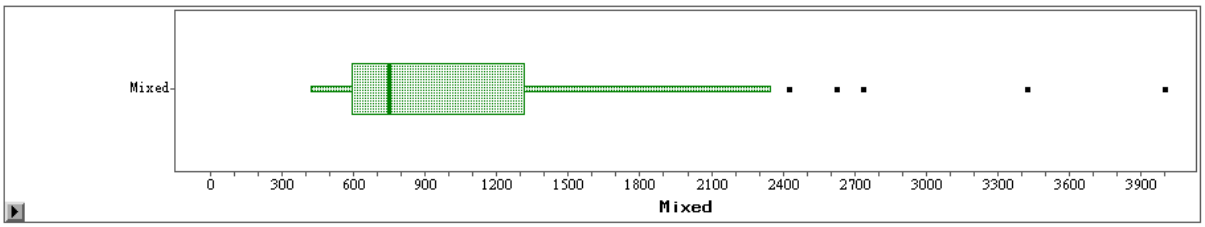
Moments			
N	61.0000	Sum Wgts	61.0000
Mean	1033.8525	Sum	63065.0000
Std Dev	563.7415	Variance	317804.528
Skewness	1.9903	Kurtosis	5.7013
USS	84268177.0	CSS	19068271.7
CV	54.5282	Std Mean	72.1797

Quantiles			
100%	Max	3563.0000	99.0% 3563.0000
75%	Q3	1281.0000	97.5% 2219.0000
50%	Med	828.0000	95.0% 2093.0000
25%	Q1	625.0000	90.0% 1766.0000
0%	Min	453.0000	10.0% 562.0000
	Range	3110.0000	5.0% 531.0000
	Q3-Q1	656.0000	2.5% 469.0000
	Mode	594.0000	1.0% 453.0000

Subject A PDT reaction times for the "question" test – buttons when possible

Mixed vs. Mixed w/ Buttons

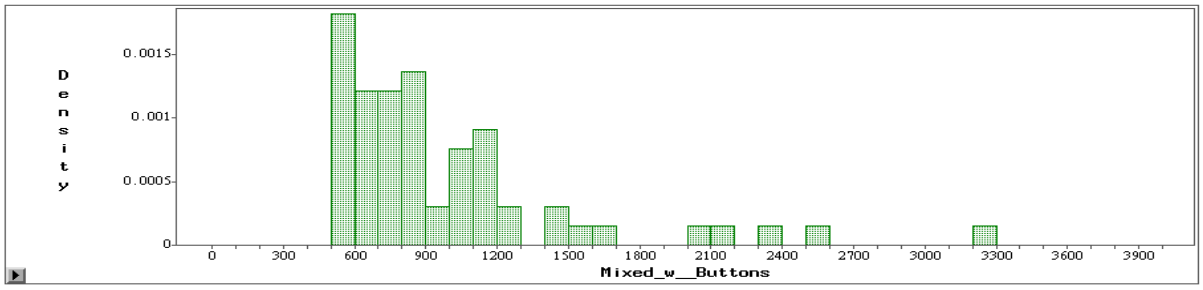


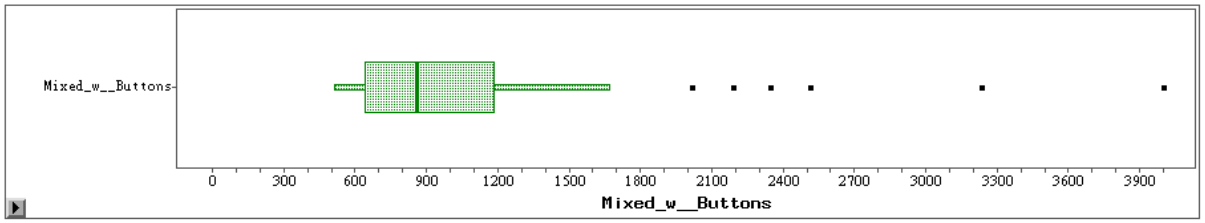


Moments			
N	60.0000	Sum Wgts	60.0000
Mean	1116.7167	Sum	67003.0000
Std Dev	851.5351	Variance	725111.969
Skewness	1.9646	Kurtosis	3.5760
USS	117604973	CSS	42781606.2
CV	76.2535	Std Mean	109.9327

Quantiles			
100% Max	4000.0000	99.0%	4000.0000
75% Q3	1312.5000	97.5%	4000.0000
50% Med	750.0000	95.0%	3078.5000
25% Q1	593.5000	90.0%	2382.5000
0% Min	422.0000	10.0%	508.0000
Range	3578.0000	5.0%	484.0000
Q3-Q1	719.0000	2.5%	453.0000
Mode	516.0000	1.0%	422.0000

Subject A PDT reaction times for the "mixed" test – speech only





Moments			
N	66.0000	Sum Wgts	66.0000
Mean	1220.3788	Sum	80545.0000
Std Dev	950.7548	Variance	903934.639
Skewness	2.1118	Kurtosis	3.5546
USS	157051161	CSS	58755751.5
CV	77.9065	Std Mean	117.0298

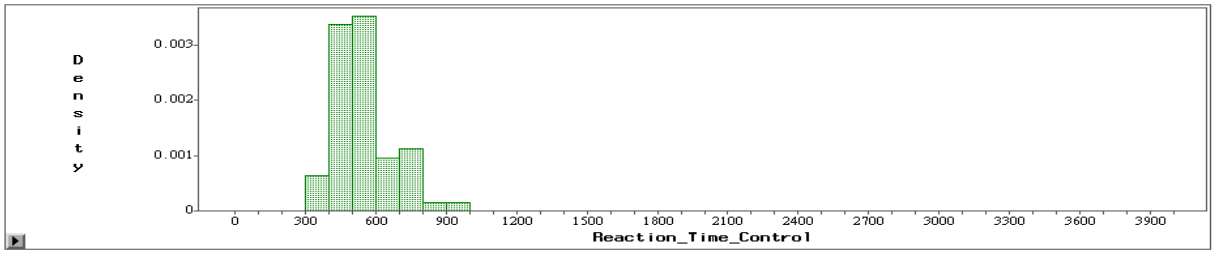
Quantiles			
100% Max	4000.0000	99.0%	4000.0000
75% Q3	1187.0000	97.5%	4000.0000
50% Med	859.0000	95.0%	4000.0000
25% Q1	641.0000	90.0%	2515.0000
0% Min	516.0000	10.0%	578.0000
Range	3484.0000	5.0%	547.0000
Q3-Q1	546.0000	2.5%	531.0000
Mode	4000.0000	1.0%	516.0000

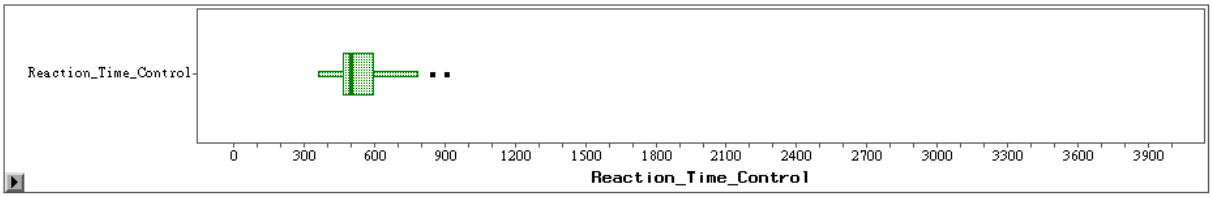
Subject A PDT reaction times for the "mixed" test – buttons when possible

Appendix I: Subject B Test Results

This appendix shows the PDT reaction time test results for Subject B.

Control vs. Overall

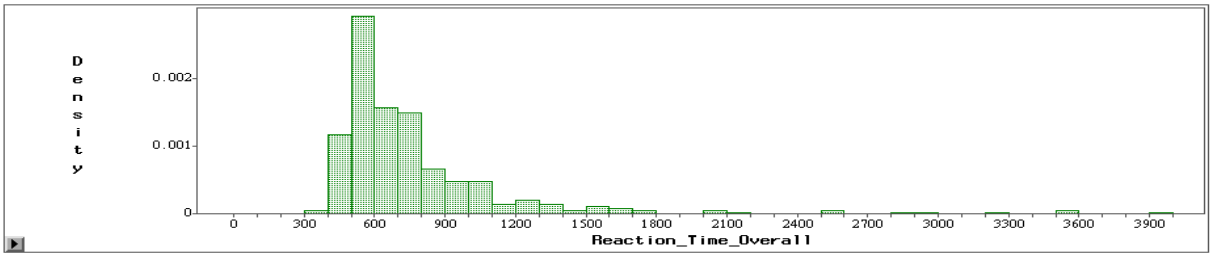


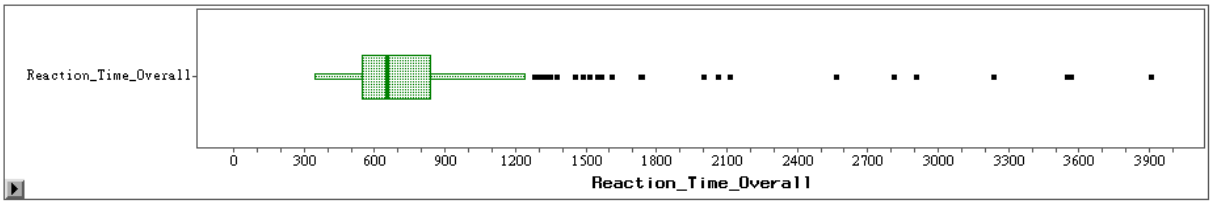


Moments			
N	62.0000	Sum Wgts	62.0000
Mean	539.3387	Sum	33439.0000
Std Dev	118.8352	Variance	14121.8014
Skewness	1.0667	Kurtosis	0.7904
USS	18896377.0	CSS	861429.887
CV	22.0335	Std Mean	15.0921

Quantiles			
100%	Max	906.0000	99.0% 906.0000
75%	Q3	594.0000	97.5% 844.0000
50%	Med	500.0000	95.0% 750.0000
25%	Q1	468.0000	90.0% 719.0000
0%	Min	360.0000	10.0% 406.0000
	Range	546.0000	5.0% 391.0000
	Q3-Q1	126.0000	2.5% 375.0000
	Mode	500.0000	1.0% 360.0000

Subject B PDT reaction times for the control test – no Jukebox interaction



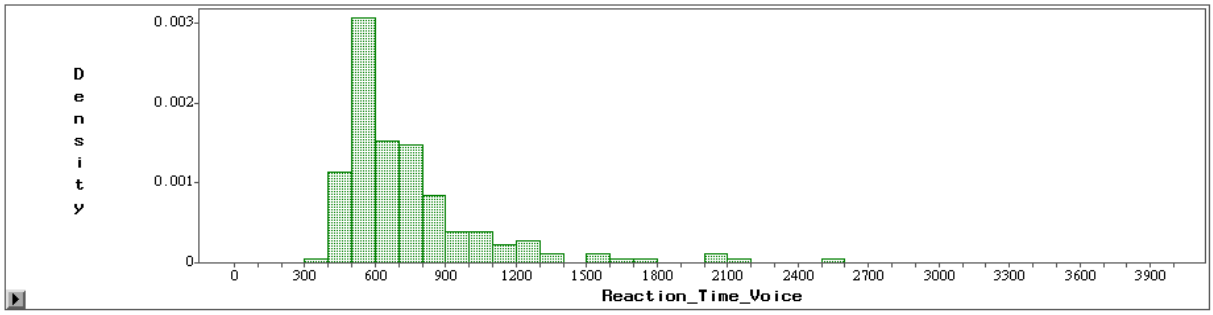


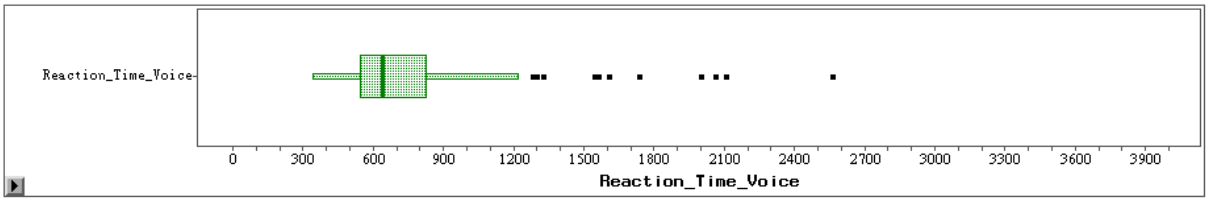
Moments			
N	348.0000	Sum Wgts	348.0000
Mean	784.7069	Sum	273078.000
Std Dev	464.2183	Variance	215498.617
Skewness	3.7020	Kurtosis	17.1988
USS	289064210	CSS	74778020.1
CV	59.1582	Std Mean	24.8847

Quantiles			
100%	Max	3906.0000	99.0% 3234.0000
75%	Q3	835.5000	97.5% 2109.0000
50%	Med	656.0000	95.0% 1547.0000
25%	Q1	547.0000	90.0% 1187.0000
0%	Min	344.0000	10.0% 484.0000
	Range	3562.0000	5.0% 468.0000
	Q3-Q1	288.5000	2.5% 438.0000
	Mode	547.0000	1.0% 406.0000

Subject B PDT reaction times for all tests with Jukebox interaction

Speech vs. Button

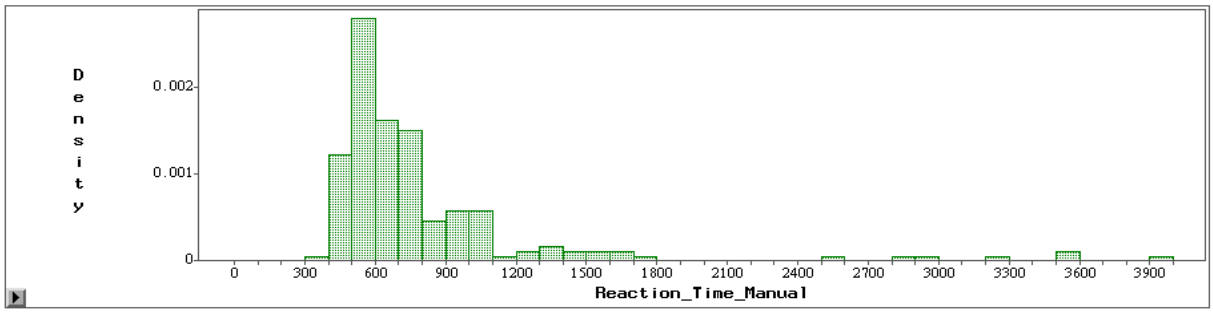


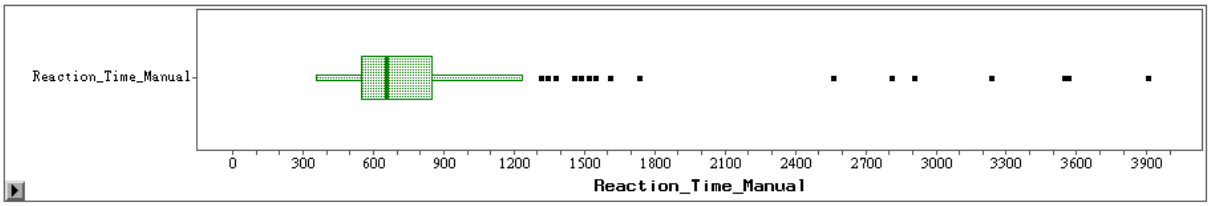


Moments			
N	176.0000	Sum Wgts	176.0000
Mean	746.8068	Sum	131438.000
Std Dev	331.9844	Variance	110213.620
Skewness	2.5198	Kurtosis	8.2045
USS	117446178	CSS	19287383.4
CV	44.4538	Std Mean	25.0243

Quantiles			
100% Max	2563.0000	99.0%	2109.0000
75% Q3	828.0000	97.5%	1735.0000
50% Med	641.0000	95.0%	1328.0000
25% Q1	547.0000	90.0%	1110.0000
0% Min	344.0000	10.0%	484.0000
Range	2219.0000	5.0%	454.0000
Q3-Q1	281.0000	2.5%	438.0000
Mode	531.0000	1.0%	406.0000

Subject B PDT reaction times for tests using speech input only



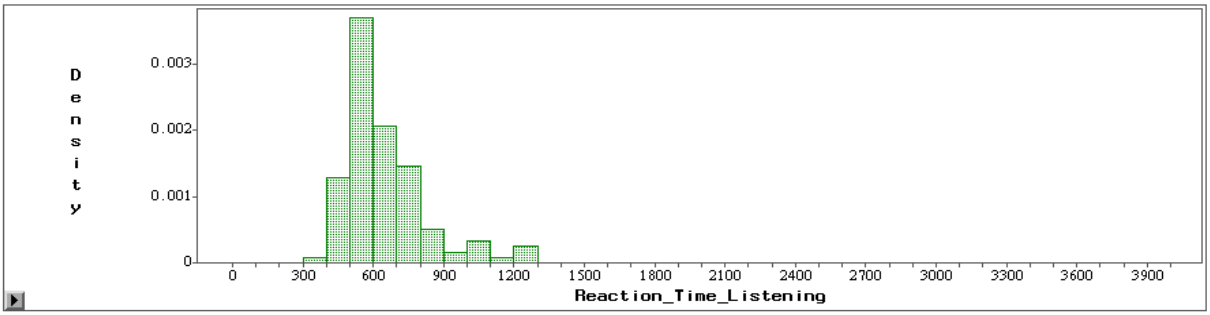


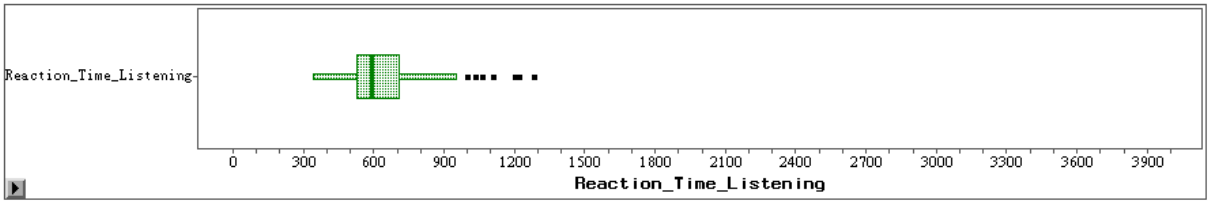
Moments			
N	172.0000	Sum Wgts	172.0000
Mean	823.4884	Sum	141640.000
Std Dev	567.0233	Variance	321515.433
Skewness	3.4926	Kurtosis	13.4853
USS	171618032	CSS	54979139.0
CV	68.8563	Std Mean	43.2351

Quantiles			
100% Max	3906.0000	99.0%	3562.0000
75% Q3	851.0000	97.5%	2906.0000
50% Med	657.0000	95.0%	1610.0000
25% Q1	547.0000	90.0%	1235.0000
0% Min	359.0000	10.0%	484.0000
Range	3547.0000	5.0%	468.0000
Q3-Q1	304.0000	2.5%	422.0000
Mode	547.0000	1.0%	406.0000

Subject B PDT reaction times for tests using buttoned input when possible

Listening vs. Interaction

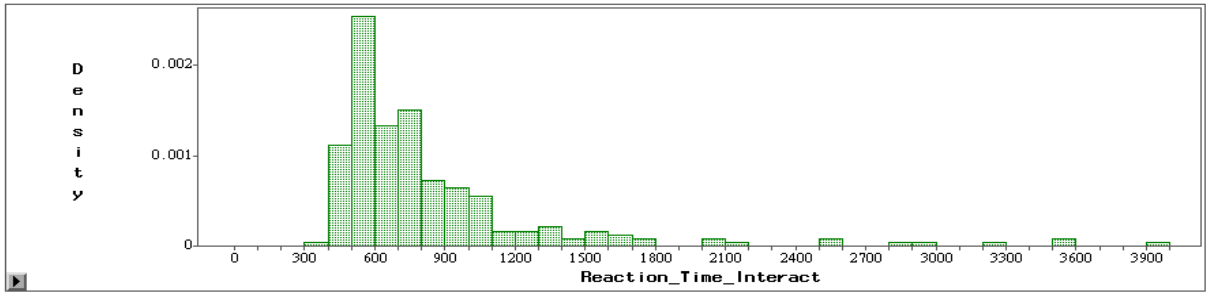


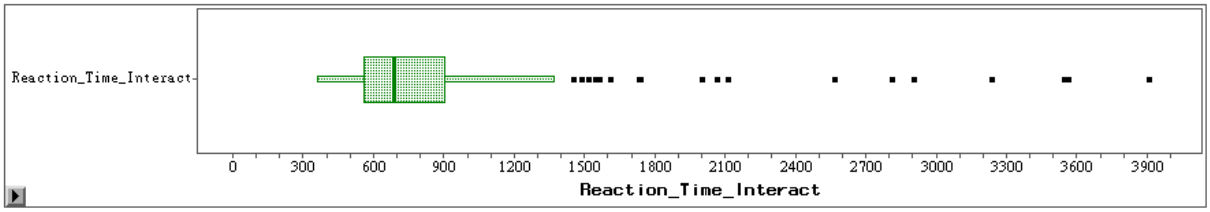


Moments			
N	116.0000	Sum Wgts	116.0000
Mean	643.8276	Sum	74684.0000
Std Dev	177.6953	Variance	31575.6048
Skewness	1.4246	Kurtosis	2.2948
USS	51714814.0	CSS	3631194.55
CV	27.5998	Std Mean	16.4986

Quantiles			
100%	Max	1281.0000	99.0% 1219.0000
75%	Q3	710.5000	97.5% 1203.0000
50%	Med	594.0000	95.0% 1062.0000
25%	Q1	531.0000	90.0% 860.0000
0%	Min	344.0000	10.0% 469.0000
	Range	937.0000	5.0% 422.0000
	Q3-Q1	179.5000	2.5% 406.0000
	Mode	547.0000	1.0% 406.0000

Subject B PDT reaction times while only music is playing – no active interaction



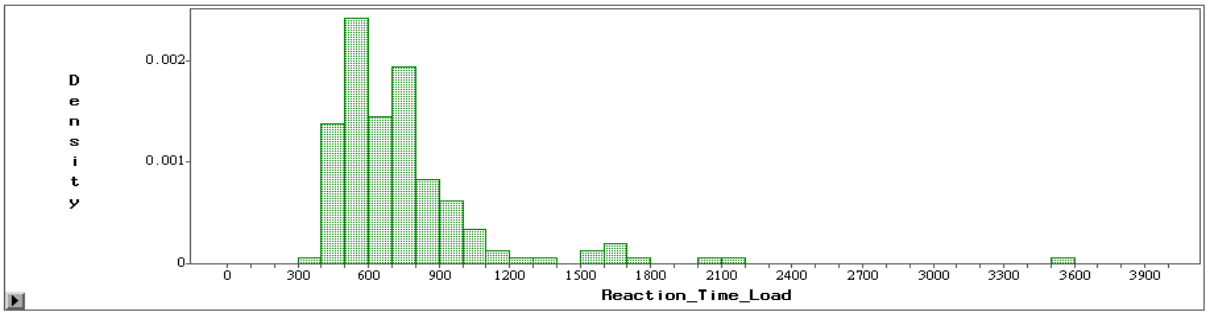


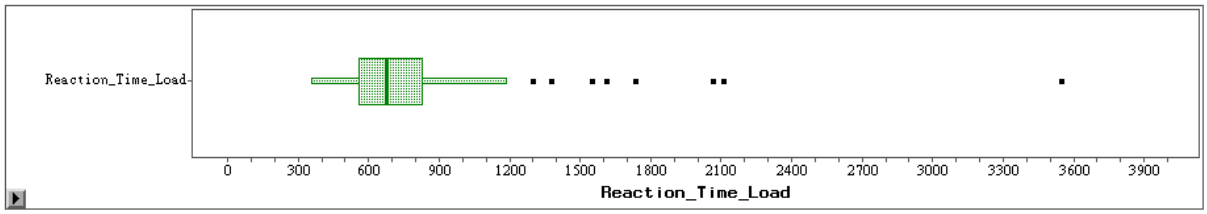
Moments			
N	232.0000	Sum Wgts	232.0000
Mean	855.1466	Sum	198394.000
Std Dev	541.3365	Variance	293045.242
Skewness	3.1473	Kurtosis	11.7334
USS	237349396	CSS	67693451.0
CV	63.3034	Std Mean	35.5405

Quantiles			
100%	Max	3906.0000	99.0% 3547.0000
75%	Q3	906.0000	97.5% 2812.0000
50%	Med	688.0000	95.0% 1735.0000
25%	Q1	562.0000	90.0% 1344.0000
0%	Min	359.0000	10.0% 485.0000
	Range	3547.0000	5.0% 469.0000
	Q3-Q1	344.0000	2.5% 453.0000
	Mode	750.0000	1.0% 438.0000

Subject B PDT reaction times during any type of Jukebox interaction – cognitive load, speech, and button

Cognitive Load vs. Command Issue

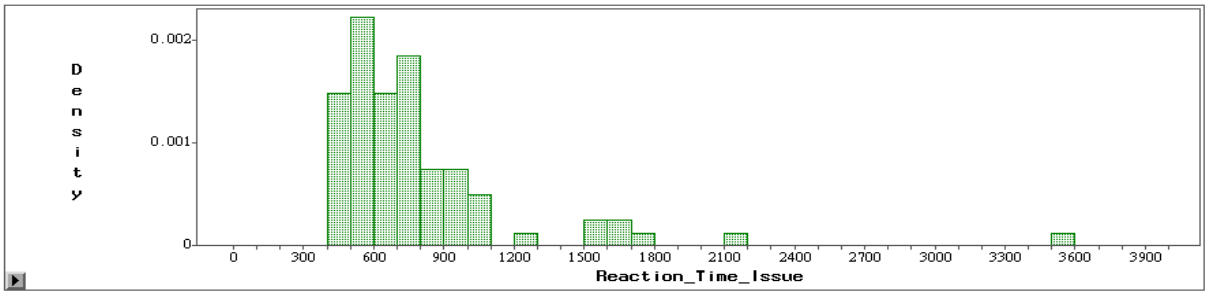


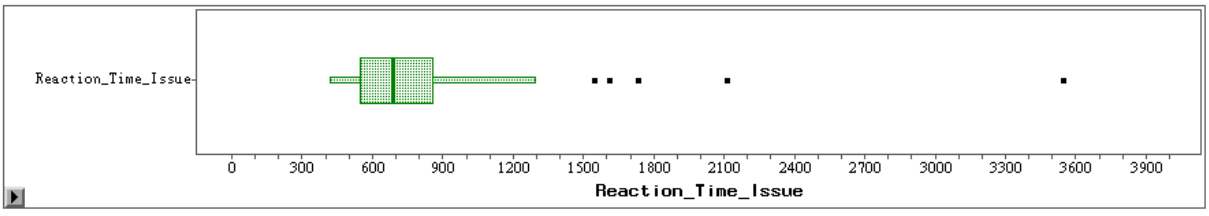


Moments			
N	144.0000	Sum Wgts	144.0000
Mean	763.0972	Sum	109886.000
Std Dev	382.9542	Variance	146653.907
Skewness	3.7408	Kurtosis	20.6508
USS	104825210	CSS	20971508.6
CV	50.1842	Std Mean	31.9128

Quantiles			
100% Max	3547.0000	99.0%	2109.0000
75% Q3	828.0000	97.5%	1734.0000
50% Med	679.5000	95.0%	1547.0000
25% Q1	562.0000	90.0%	1078.0000
0% Min	359.0000	10.0%	484.0000
Range	3188.0000	5.0%	454.0000
Q3-Q1	266.0000	2.5%	438.0000
Mode	750.0000	1.0%	422.0000

Subject B PDT reaction times during cognitive load conditions



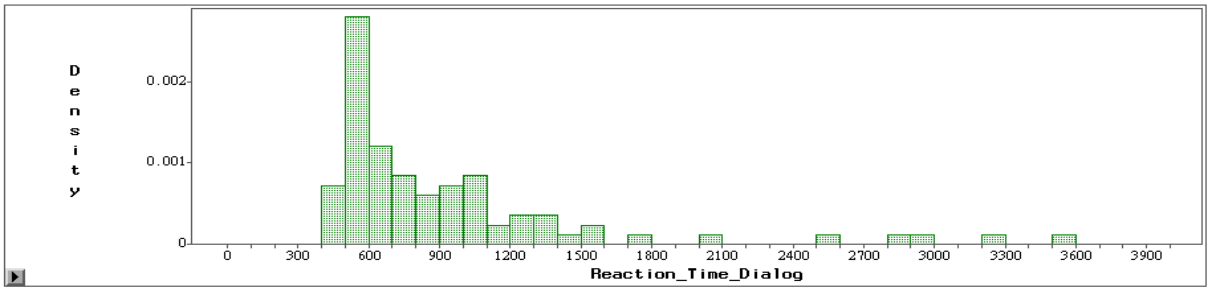


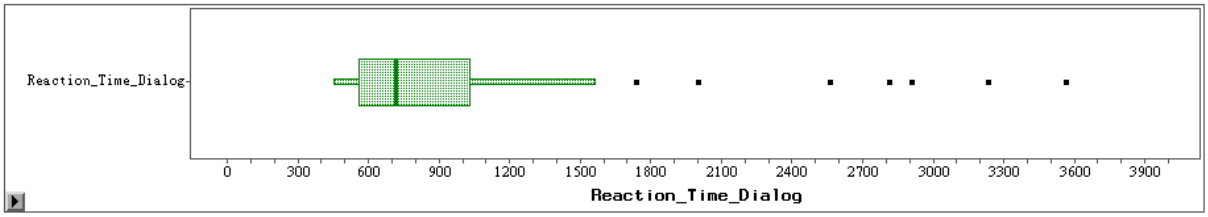
Moments			
N	81.0000	Sum Wgts	81.0000
Mean	792.0370	Sum	64155.0000
Std Dev	446.5082	Variance	199369.536
Skewness	3.6673	Kurtosis	18.3187
USS	66762699.0	CSS	15949562.9
CV	56.3747	Std Mean	49.6120

Quantiles			
100% Max	3547.0000	99.0%	3547.0000
75% Q3	859.0000	97.5%	1734.0000
50% Med	687.0000	95.0%	1609.0000
25% Q1	547.0000	90.0%	1094.0000
0% Min	422.0000	10.0%	484.0000
Range	3125.0000	5.0%	469.0000
Q3-Q1	312.0000	2.5%	453.0000
Mode	750.0000	1.0%	422.0000

Subject B PDT reaction times while the tester issues commands

Dialog vs. Manual

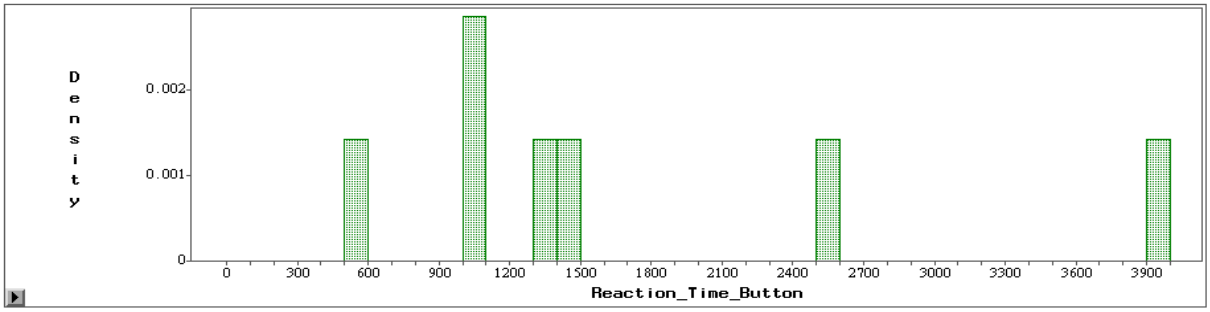


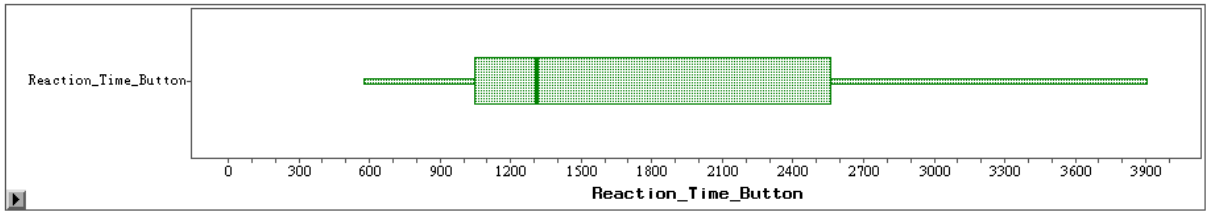


Moments			
N	82.0000	Sum Wgts	82.0000
Mean	946.5366	Sum	77616.0000
Std Dev	627.5532	Variance	393823.042
Skewness	2.4644	Kurtosis	6.3929
USS	105366050	CSS	31899666.4
CV	66.2999	Std Mean	69.3017

Quantiles			
100%	Max	3562.0000	99.0% 3562.0000
75%	Q3	1031.0000	97.5% 2906.0000
50%	Med	719.0000	95.0% 2563.0000
25%	Q1	562.0000	90.0% 1516.0000
0%	Min	453.0000	10.0% 516.0000
	Range	3109.0000	5.0% 484.0000
	Q3-Q1	469.0000	2.5% 469.0000
	Mode	532.0000	1.0% 453.0000

Subject B PDT reaction times during any Jukebox interaction with speech



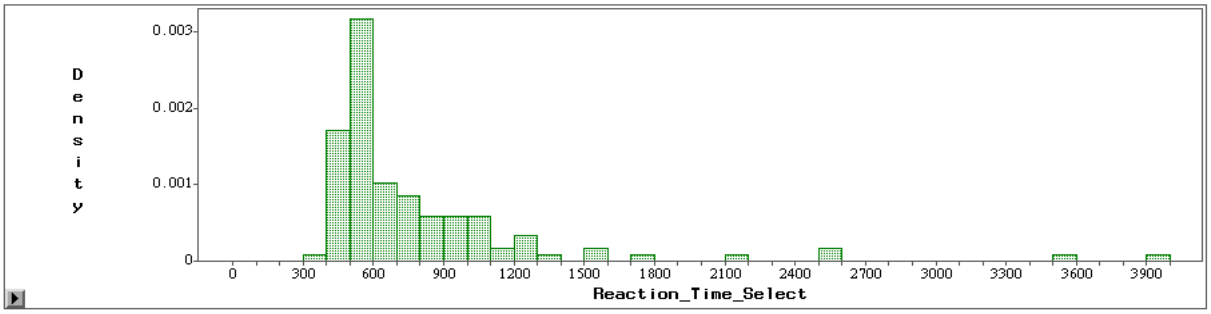


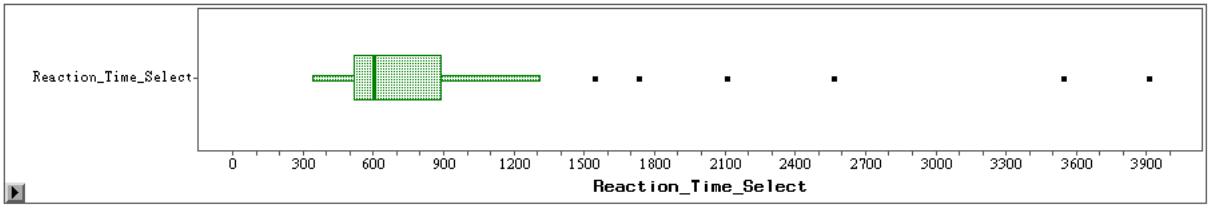
Moments			
N	7.0000	Sum Wgts	7.0000
Mean	1701.1429	Sum	11908.0000
Std Dev	1149.8683	Variance	1322197.14
Skewness	1.4286	Kurtosis	1.5855
USS	28190392.0	CSS	7933182.86
CV	67.5939	Std Mean	434.6094

Quantiles			
100%	Max	3906.0000	99.0% 3906.0000
75%	Q3	2563.0000	97.5% 3906.0000
50%	Med	1313.0000	95.0% 3906.0000
25%	Q1	1047.0000	90.0% 3906.0000
0%	Min	578.0000	10.0% 578.0000
	Range	3328.0000	5.0% 578.0000
	Q3-Q1	1516.0000	2.5% 578.0000
	Mode	1047.0000	1.0% 578.0000

Subject B PDT reaction times during any Jukebox interaction with steering wheel button use

Selection vs. Question vs. Mixed

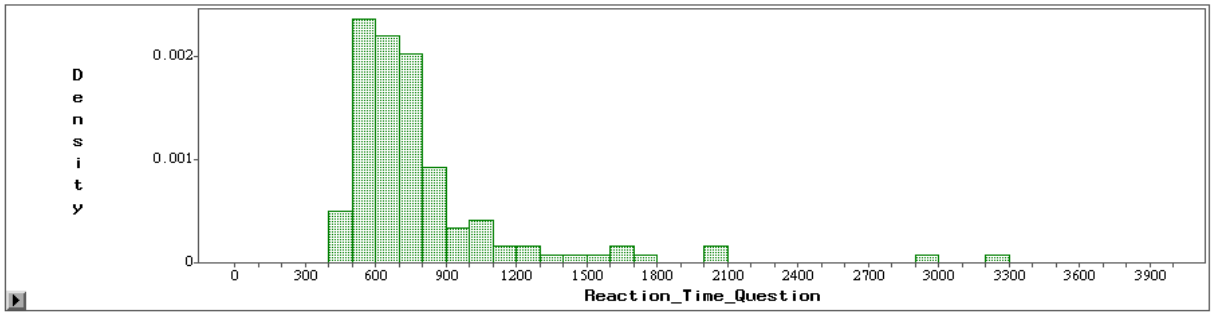


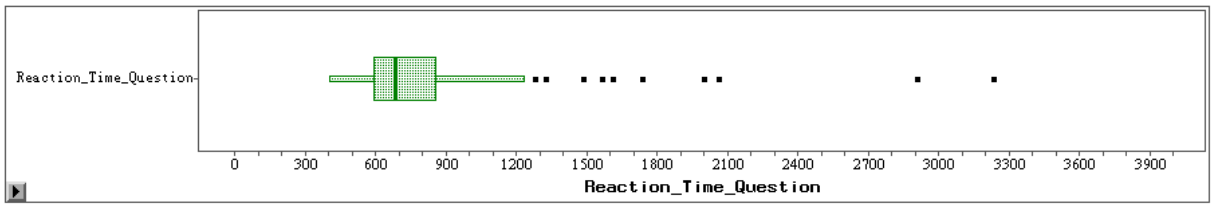


Moments			
N	116.0000	Sum Wgts	116.0000
Mean	800.8103	Sum	92894.0000
Std Dev	544.0978	Variance	296042.433
Skewness	3.5474	Kurtosis	15.1277
USS	108435356	CSS	34044879.8
CV	67.9434	Std Mean	50.5182

Quantiles			
100%	Max	3906.0000	99.0% 3547.0000
75%	Q3	890.5000	97.5% 2563.0000
50%	Med	602.0000	95.0% 1734.0000
25%	Q1	516.0000	90.0% 1219.0000
0%	Min	344.0000	10.0% 469.0000
	Range	3562.0000	5.0% 454.0000
	Q3-Q1	374.5000	2.5% 438.0000
	Mode	531.0000	1.0% 422.0000

Subject B PDT reaction times for tests with Jukebox interaction using plain selection



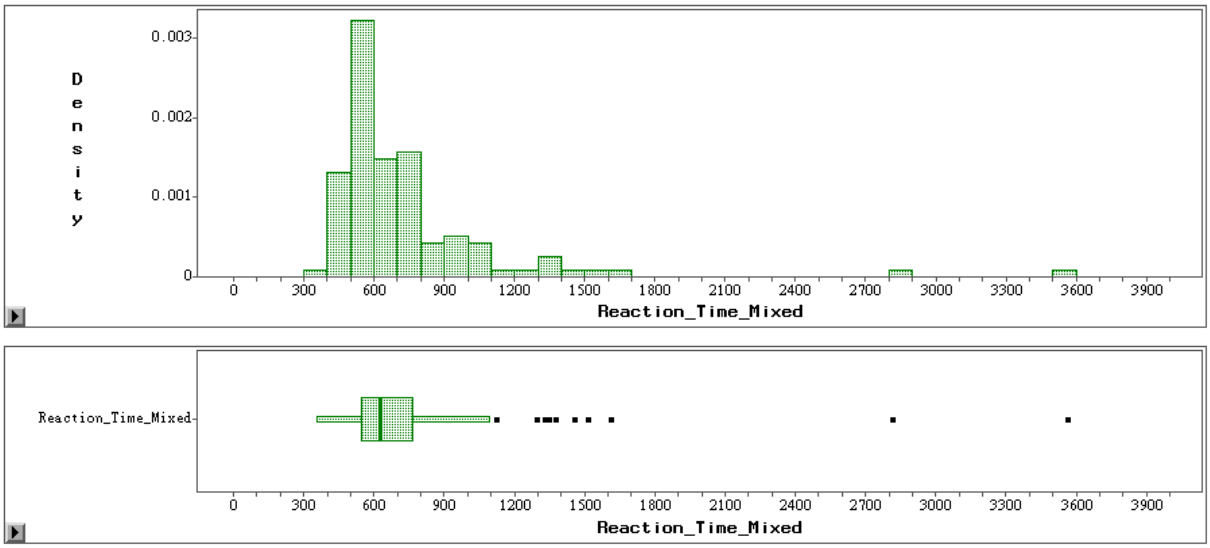


Moments			
N	118.0000	Sum Wgts	118.0000
Mean	810.6017	Sum	95651.0000
Std Dev	424.8775	Variance	180520.908
Skewness	3.3100	Kurtosis	13.6900
USS	98655809.0	CSS	21120946.3
CV	52.4151	Std Mean	39.1131

Quantiles			
100%	Max	3234.0000	99.0% 2906.0000
75%	Q3	859.0000	97.5% 2062.0000
50%	Med	687.5000	95.0% 1609.0000
25%	Q1	593.0000	90.0% 1235.0000
0%	Min	406.0000	10.0% 516.0000
	Range	2828.0000	5.0% 485.0000
	Q3-Q1	266.0000	2.5% 469.0000
	Mode	656.0000	1.0% 453.0000

Subject B PDT reaction times for tests with Jukebox interaction using questions and user assignments

Selection vs. Question vs. Mixed (cont'd)



Moments

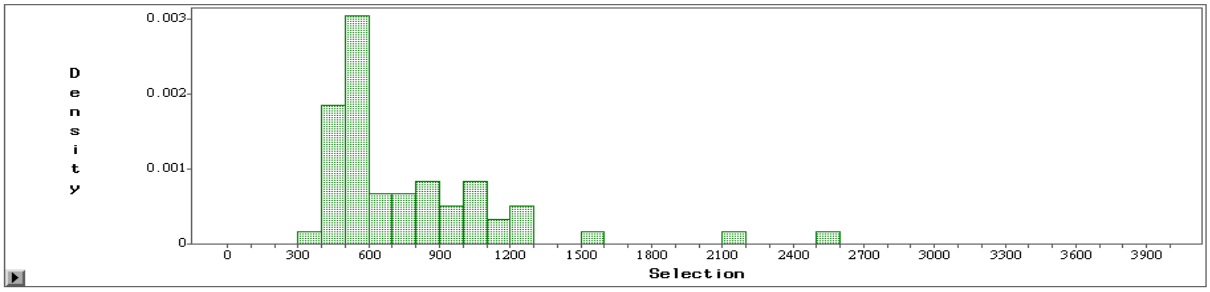
N	114.0000	Sum Wgts	114.0000
Mean	741.5175	Sum	84533.0000
Std Dev	413.1719	Variance	170710.995
Skewness	4.2413	Kurtosis	23.6929
USS	81973045.0	CSS	19290342.5
CV	55.7198	Std Mean	38.6971

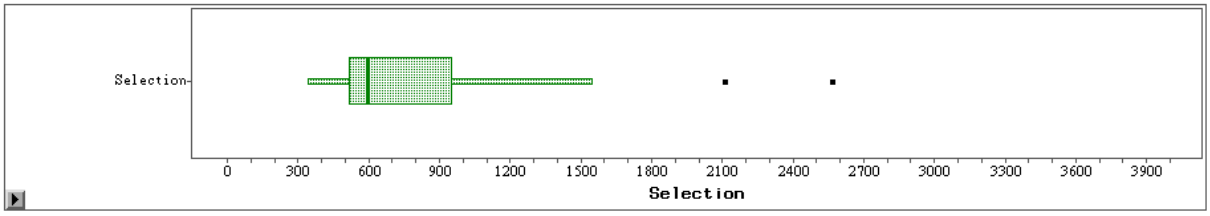
Quantiles

100% Max	3562.0000	99.0%	2812.0000
75% Q3	766.0000	97.5%	1610.0000
50% Med	625.0000	95.0%	1375.0000
25% Q1	547.0000	90.0%	1062.0000
0% Min	359.0000	10.0%	484.0000
Range	3203.0000	5.0%	453.0000
Q3-Q1	219.0000	2.5%	422.0000
Mode	547.0000	1.0%	406.0000

Subject B PDT reaction times for tests with Jukebox interaction using a mix of questions and selection methods

Selection vs. Selection w/ Buttons

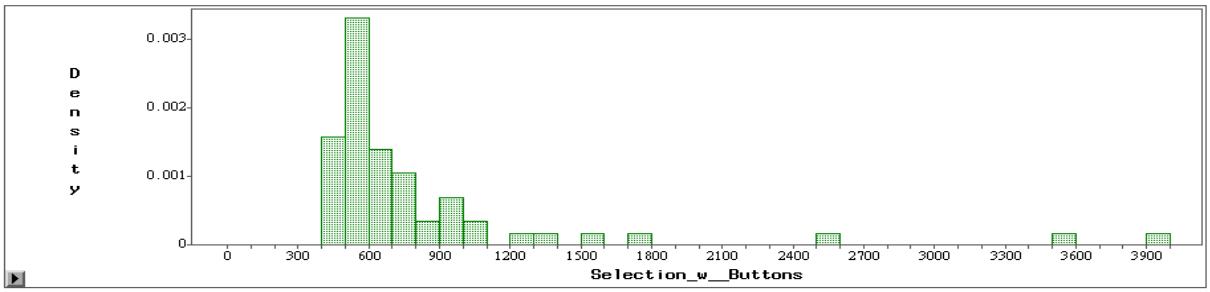


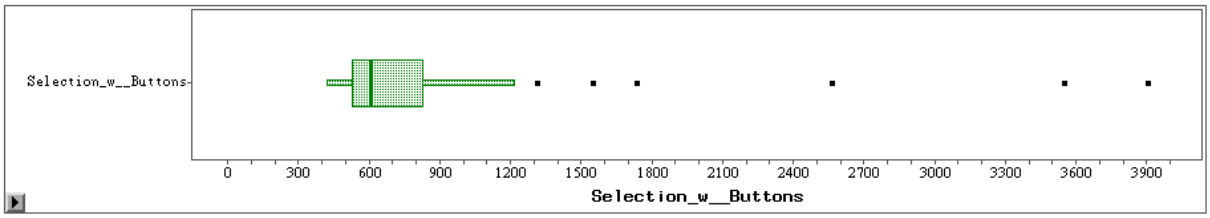


Moments			
N	59.0000	Sum Wgts	59.0000
Mean	768.3220	Sum	45331.0000
Std Dev	398.2015	Variance	158564.463
Skewness	2.4242	Kurtosis	7.7631
USS	44025545.0	CSS	9196738.88
CV	51.8274	Std Mean	51.8414

Quantiles			
100%	Max	2563.0000	99.0% 2563.0000
75%	Q3	953.0000	97.5% 2109.0000
50%	Med	594.0000	95.0% 1547.0000
25%	Q1	516.0000	90.0% 1203.0000
0%	Min	344.0000	10.0% 454.0000
	Range	2219.0000	5.0% 438.0000
	Q3-Q1	437.0000	2.5% 438.0000
	Mode	531.0000	1.0% 344.0000

Subject B PDT reaction times for the "selection" test – speech only



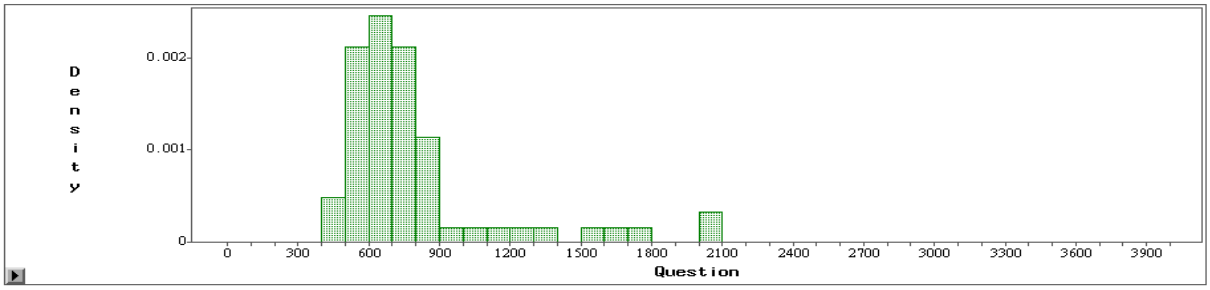


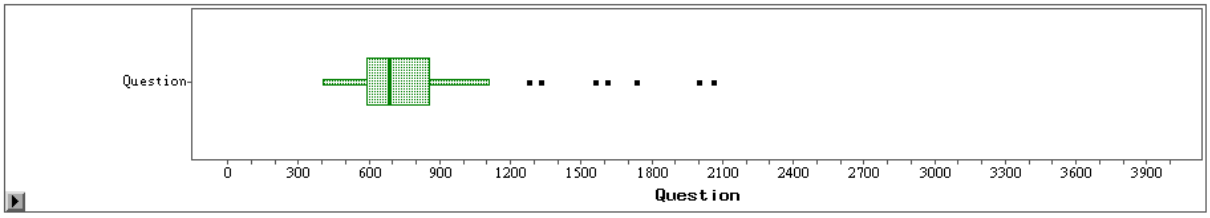
Moments			
N	57.0000	Sum Wgts	57.0000
Mean	834.4386	Sum	47563.0000
Std Dev	664.4198	Variance	441453.715
Skewness	3.4326	Kurtosis	12.4771
USS	64409811.0	CSS	24721408.0
CV	79.6248	Std Mean	88.0046

Quantiles			
100%	Max	3906.0000	99.0% 3906.0000
75%	Q3	828.0000	97.5% 3547.0000
50%	Med	610.0000	95.0% 2563.0000
25%	Q1	531.0000	90.0% 1313.0000
0%	Min	422.0000	10.0% 469.0000
	Range	3484.0000	5.0% 468.0000
	Q3-Q1	297.0000	2.5% 454.0000
	Mode	547.0000	1.0% 422.0000

Subject B PDT reaction times for the "selection" test – buttons when possible

Question vs. Question w/ Buttons

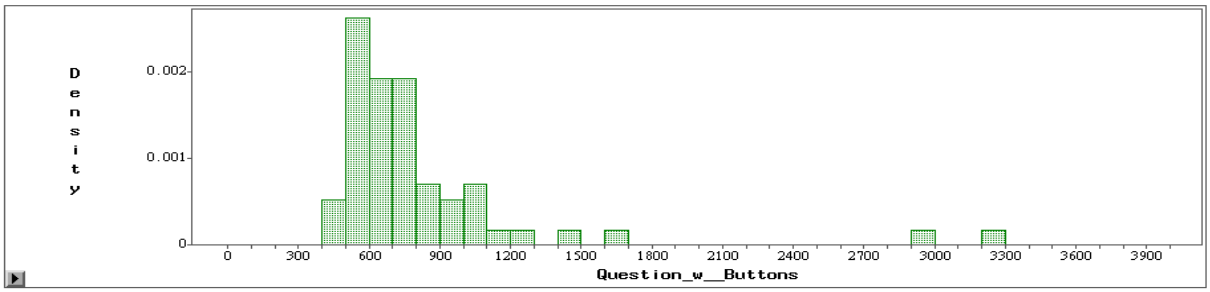


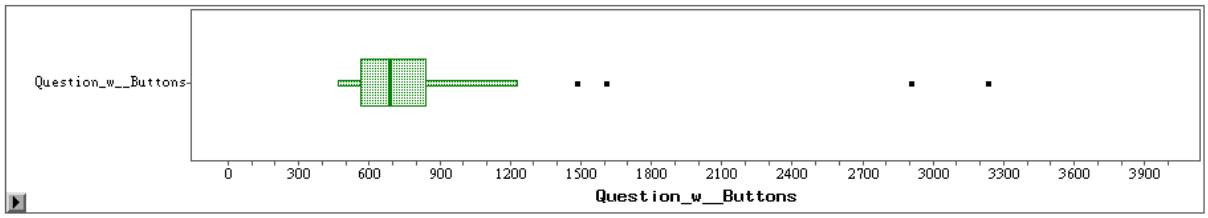


Moments			
N	61.0000	Sum Wgts	61.0000
Mean	798.3443	Sum	48699.0000
Std Dev	351.9512	Variance	123869.630
Skewness	2.1676	Kurtosis	4.5620
USS	46310745.0	CSS	7432177.77
CV	44.0851	Std Mean	45.0627

Quantiles			
100%	Max	2062.0000	99.0% 2062.0000
75%	Q3	859.0000	97.5% 2000.0000
50%	Med	688.0000	95.0% 1609.0000
25%	Q1	594.0000	90.0% 1281.0000
0%	Min	406.0000	10.0% 532.0000
	Range	1656.0000	5.0% 516.0000
	Q3-Q1	265.0000	2.5% 453.0000
	Mode	625.0000	1.0% 406.0000

Subject B PDT reaction times for the "question" test – speech only



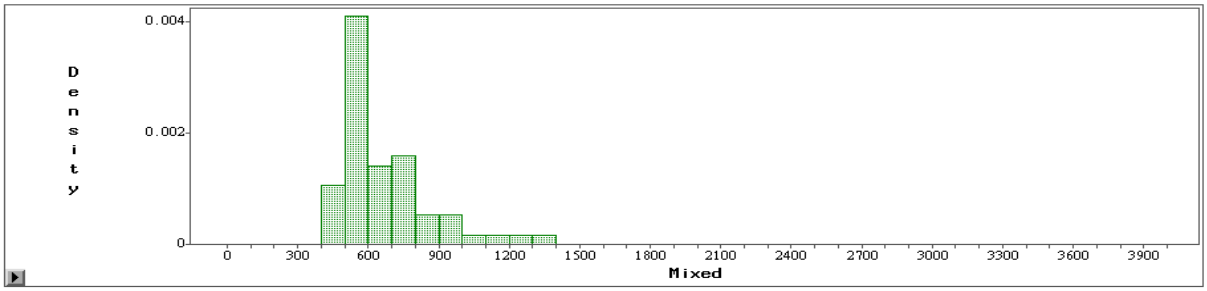


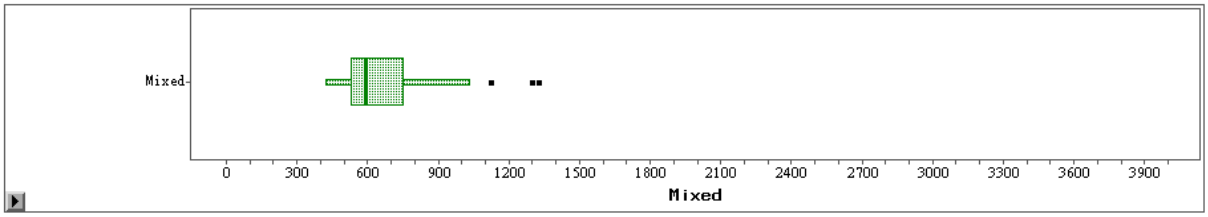
Moments			
N	57.0000	Sum Wgts	57.0000
Mean	823.7193	Sum	46952.0000
Std Dev	494.0683	Variance	244103.491
Skewness	3.5985	Kurtosis	14.6215
USS	52345064.0	CSS	13669795.5
CV	59.9802	Std Mean	65.4409

Quantiles			
100%	Max	3234.0000	99.0% 3234.0000
75%	Q3	843.0000	97.5% 2906.0000
50%	Med	687.0000	95.0% 1609.0000
25%	Q1	563.0000	90.0% 1187.0000
0%	Min	469.0000	10.0% 516.0000
	Range	2765.0000	5.0% 485.0000
	Q3-Q1	280.0000	2.5% 469.0000
	Mode	750.0000	1.0% 469.0000

Subject B PDT reaction times for the "question" test – buttons when possible

Mixed vs. Mixed w/ Buttons

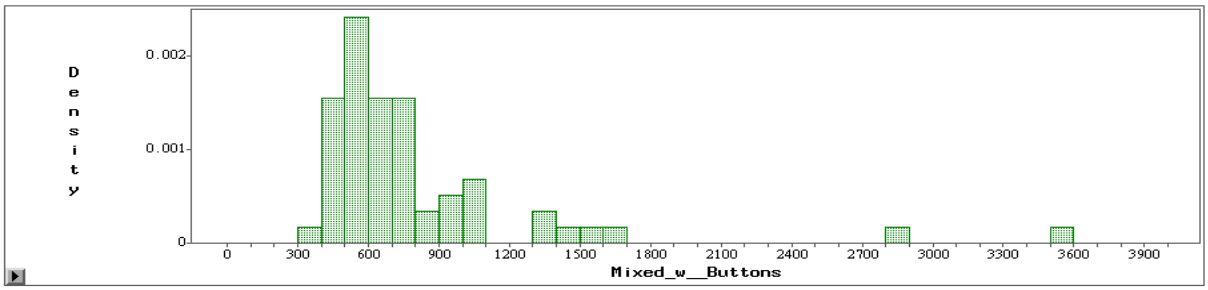


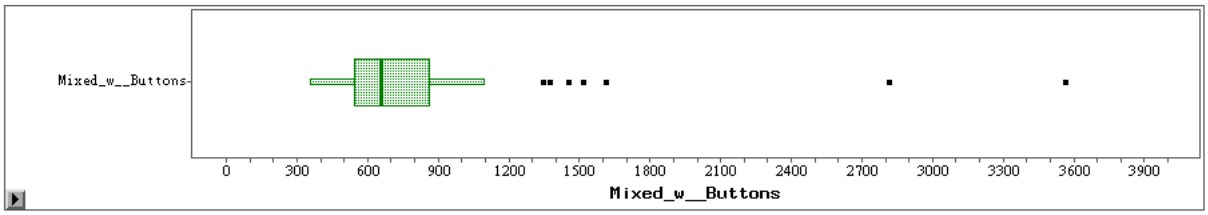


Moments			
N	56.0000	Sum Wgts	56.0000
Mean	668.0000	Sum	37408.0000
Std Dev	196.3922	Variance	38569.8909
Skewness	1.6620	Kurtosis	2.8805
USS	27109888.0	CSS	2121344.00
CV	29.4000	Std Mean	26.2440

Quantiles			
100%	Max	1328.0000	99.0% 1328.0000
75%	Q3	750.0000	97.5% 1297.0000
50%	Med	594.0000	95.0% 1125.0000
25%	Q1	531.5000	90.0% 906.0000
0%	Min	422.0000	10.0% 484.0000
	Range	906.0000	5.0% 484.0000
	Q3-Q1	218.5000	2.5% 453.0000
	Mode	484.0000	1.0% 422.0000

Subject B PDT reaction times for the "mixed" test – speech only





Moments			
N	58.0000	Sum Wgts	58.0000
Mean	812.5000	Sum	47125.0000
Std Dev	539.2343	Variance	290773.588
Skewness	3.4167	Kurtosis	13.9916
USS	54863157.0	CSS	16574094.5
CV	66.3673	Std Mean	70.8049

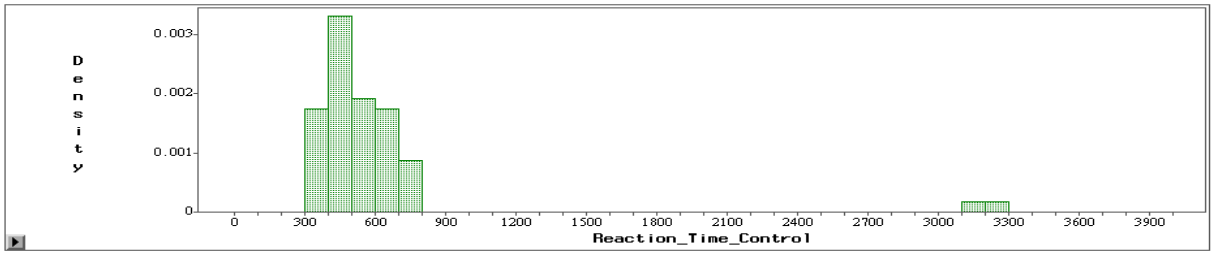
Quantiles			
100%	Max	3562.0000	99.0% 3562.0000
75%	Q3	860.0000	97.5% 2812.0000
50%	Med	657.0000	95.0% 1610.0000
25%	Q1	547.0000	90.0% 1375.0000
0%	Min	359.0000	10.0% 468.0000
	Range	3203.0000	5.0% 422.0000
	Q3-Q1	313.0000	2.5% 406.0000
	Mode	547.0000	1.0% 359.0000

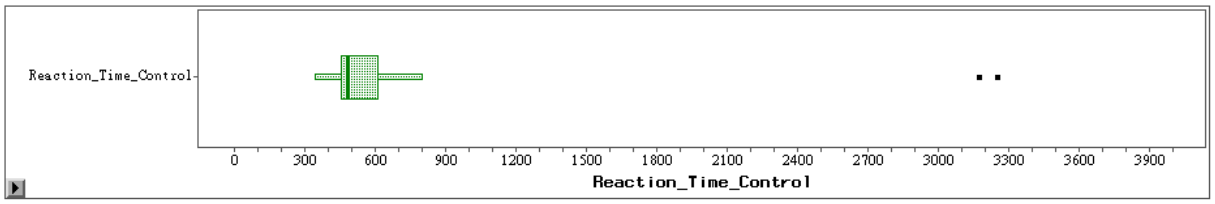
Subject B PDT reaction times for the "mixed" test – buttons when possible

Appendix J: Subject C Test Results

This appendix shows the PDT reaction time test results for Subject C.

Control vs. Overall

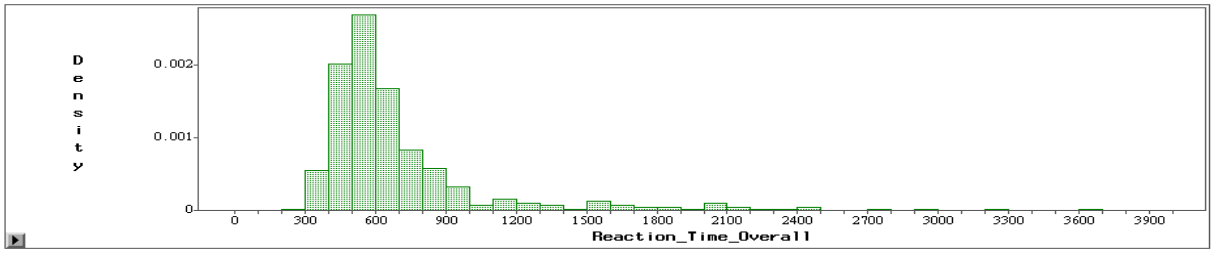


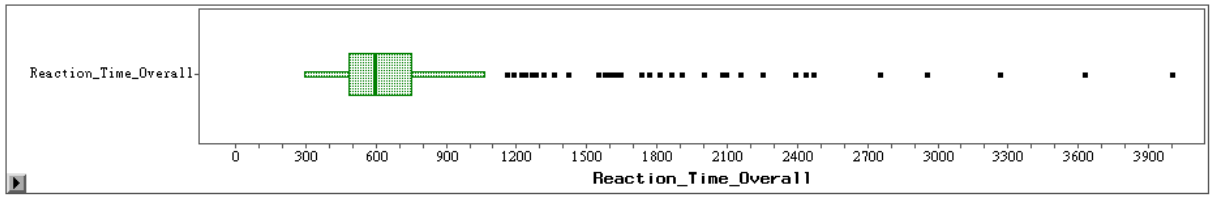


Moments			
N	57.0000	Sum Wgts	57.0000
Mean	610.7368	Sum	34812.0000
Std Dev	511.5211	Variance	261653.876
Skewness	4.8406	Kurtosis	23.4262
USS	35913588.0	CSS	14652617.1
CV	83.7548	Std Mean	67.7526

Quantiles			
100%	Max	3250.0000	99.0% 3250.0000
75%	Q3	610.0000	97.5% 3172.0000
50%	Med	485.0000	95.0% 797.0000
25%	Q1	453.0000	90.0% 703.0000
0%	Min	344.0000	10.0% 391.0000
	Range	2906.0000	5.0% 375.0000
	Q3-Q1	157.0000	2.5% 375.0000
	Mode	391.0000	1.0% 344.0000

Subject C PDT reaction times for the control test – no Jukebox interaction



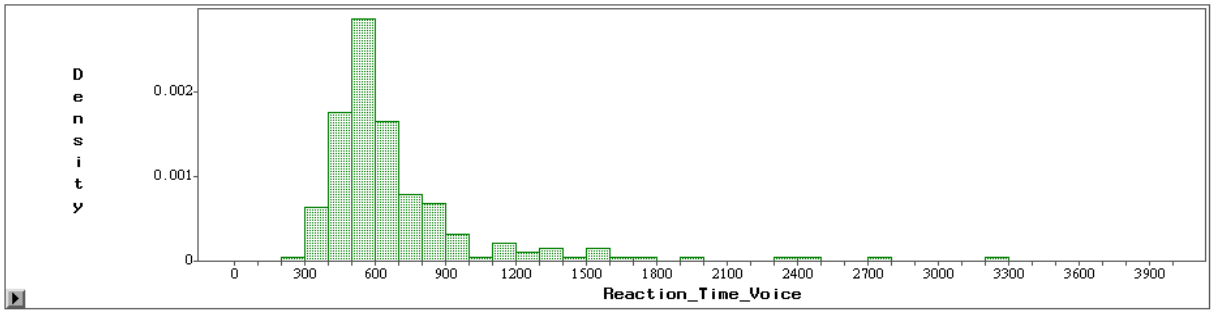


Moments			
N	358.0000	Sum Wgts	358.0000
Mean	746.5978	Sum	267282.000
Std Dev	520.9133	Variance	271350.661
Skewness	3.3696	Kurtosis	13.8187
USS	296424330	CSS	96872186.1
CV	69.7716	Std Mean	27.5311

Quantiles			
100%	Max	4000.0000	99.0% 3266.0000
75%	Q3	750.0000	97.5% 2390.0000
50%	Med	593.5000	95.0% 1859.0000
25%	Q1	484.0000	90.0% 1235.0000
0%	Min	297.0000	10.0% 422.0000
	Range	3703.0000	5.0% 391.0000
	Q3-Q1	266.0000	2.5% 344.0000
	Mode	500.0000	1.0% 328.0000

Subject C PDT reaction times for all tests with Jukebox interaction

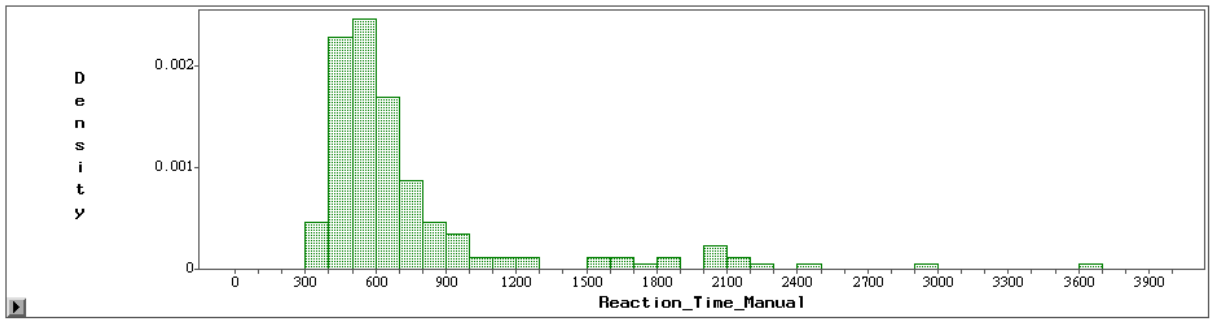
Speech vs. Button



Moments			
N	187.0000	Sum Wgts	187.0000
Mean	721.3690	Sum	134896.000
Std Dev	478.3985	Variance	228865.094
Skewness	3.7472	Kurtosis	17.9848
USS	139878698	CSS	42568907.5
CV	66.3181	Std Mean	34.9840

Quantiles			
100%	Max	4000.0000	99.0% 3266.0000
75%	Q3	765.0000	97.5% 2390.0000
50%	Med	594.0000	95.0% 1578.0000
25%	Q1	500.0000	90.0% 1188.0000
0%	Min	297.0000	10.0% 422.0000
	Range	3703.0000	5.0% 375.0000
	Q3-Q1	265.0000	2.5% 344.0000
	Mode	500.0000	1.0% 313.0000

Subject C PDT reaction times for tests using speech input only

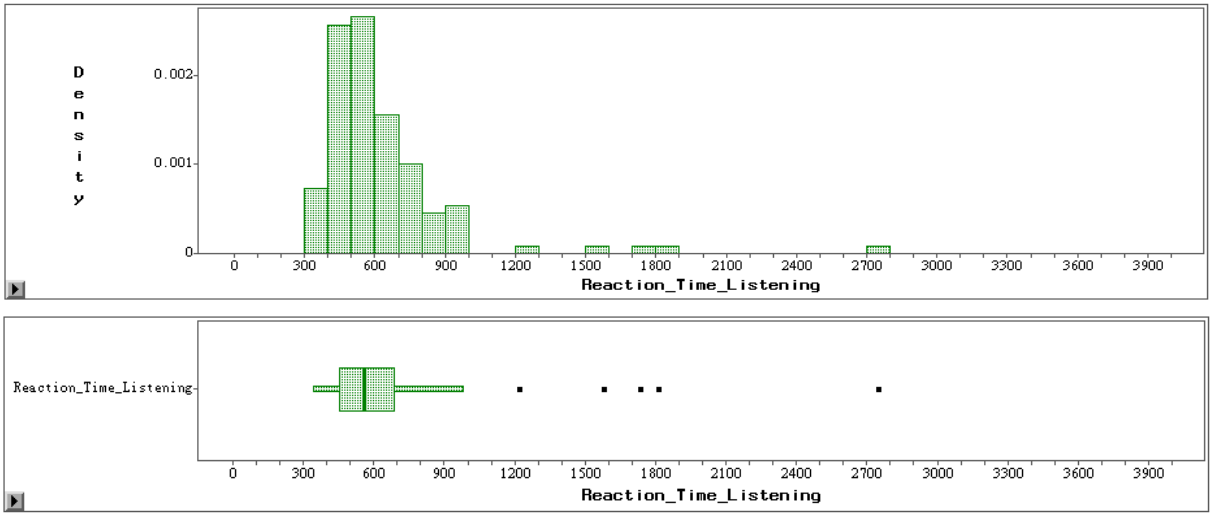


Moments			
N	171.0000	Sum Wgts	171.0000
Mean	774.1871	Sum	132386.000
Std Dev	563.8841	Variance	317965.259
Skewness	3.0654	Kurtosis	11.0495
USS	156545632	CSS	54054094.0
CV	72.8356	Std Mean	43.1213

Quantiles			
100%	Max	4000.0000	99.0% 3625.0000
75%	Q3	750.0000	97.5% 2250.0000
50%	Med	593.0000	95.0% 2078.0000
25%	Q1	484.0000	90.0% 1594.0000
0%	Min	343.0000	10.0% 422.0000
	Range	3657.0000	5.0% 406.0000
	Q3-Q1	266.0000	2.5% 391.0000
	Mode	656.0000	1.0% 344.0000

Subject C PDT reaction times for tests using buttoned input when possible

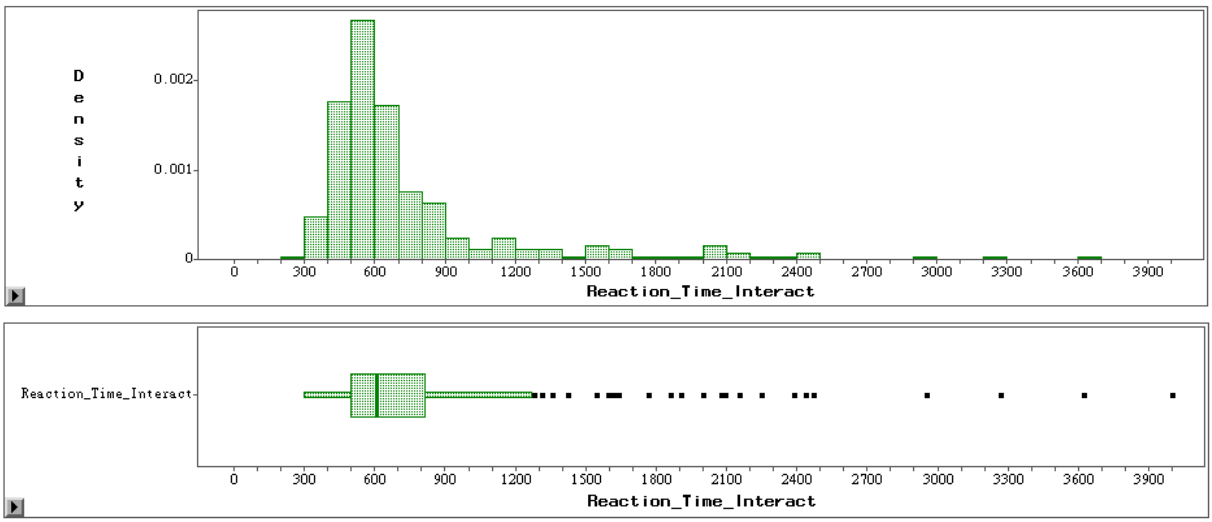
Listening vs. Interaction



Moments			
N	109.0000	Sum Wgts	109.0000
Mean	635.7156	Sum	69293.0000
Std Dev	321.3186	Variance	103245.613
Skewness	3.7784	Kurtosis	19.3382
USS	55201167.0	CSS	11150526.2
CV	50.5444	Std Mean	30.7767

Quantiles			
100% Max	2750.0000	99.0%	1813.0000
75% Q3	688.0000	97.5%	1734.0000
50% Med	562.0000	95.0%	984.0000
25% Q1	454.0000	90.0%	906.0000
0% Min	343.0000	10.0%	406.0000
Range	2407.0000	5.0%	375.0000
Q3-Q1	234.0000	2.5%	344.0000
Mode	500.0000	1.0%	344.0000

Subject C PDT reaction times while only music is playing – no active interaction

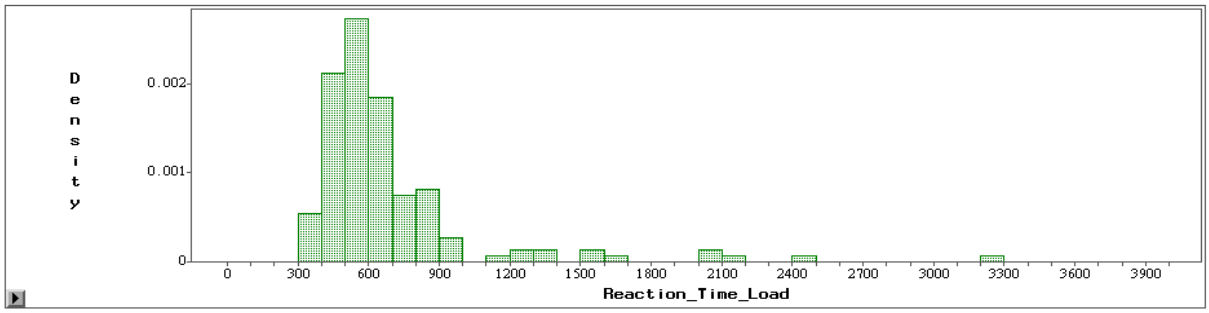


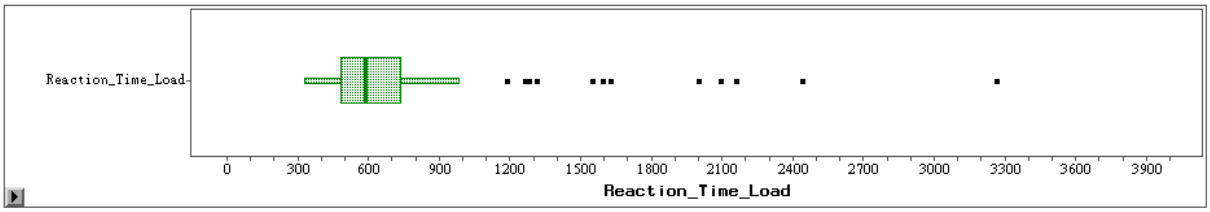
Moments			
N	249.0000	Sum Wgts	249.0000
Mean	795.1365	Sum	197989.000
Std Dev	581.2767	Variance	337882.554
Skewness	3.0512	Kurtosis	10.9384
USS	241223163	CSS	83794873.4
CV	73.1040	Std Mean	36.8369

Quantiles			
100% Max	4000.0000	99.0%	3625.0000
75% Q3	812.0000	97.5%	2437.0000
50% Med	609.0000	95.0%	2078.0000
25% Q1	500.0000	90.0%	1547.0000
0% Min	297.0000	10.0%	438.0000
Range	3703.0000	5.0%	391.0000
Q3-Q1	312.0000	2.5%	344.0000
Mode	500.0000	1.0%	328.0000

Subject C PDT reaction times during any type of Jukebox interaction – cognitive load, speech, and button

Cognitive Load vs. Command Issue

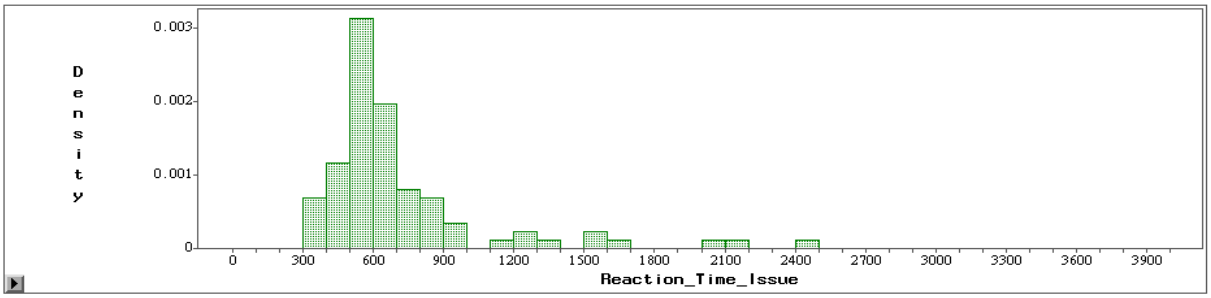


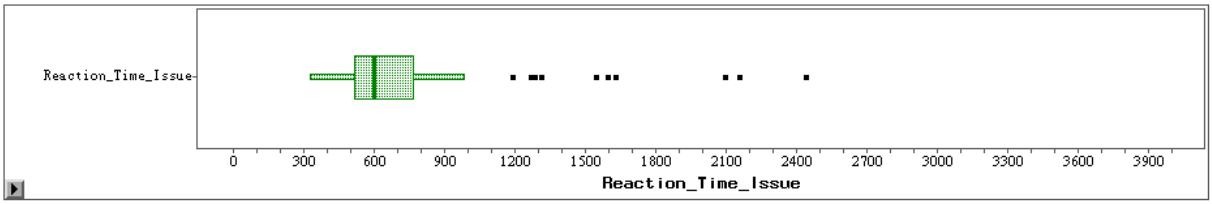


Moments			
N	146.0000	Sum Wgts	146.0000
Mean	695.2123	Sum	101501.000
Std Dev	406.1602	Variance	164966.072
Skewness	3.4085	Kurtosis	14.6190
USS	94484827.0	CSS	23920080.4
CV	58.4225	Std Mean	33.6141

Quantiles			
100% Max	3266.0000	99.0%	2437.0000
75% Q3	735.0000	97.5%	2094.0000
50% Med	586.0000	95.0%	1547.0000
25% Q1	484.0000	90.0%	953.0000
0% Min	328.0000	10.0%	422.0000
Range	2938.0000	5.0%	391.0000
Q3-Q1	251.0000	2.5%	344.0000
Mode	469.0000	1.0%	344.0000

Subject C PDT reaction times during cognitive load conditions



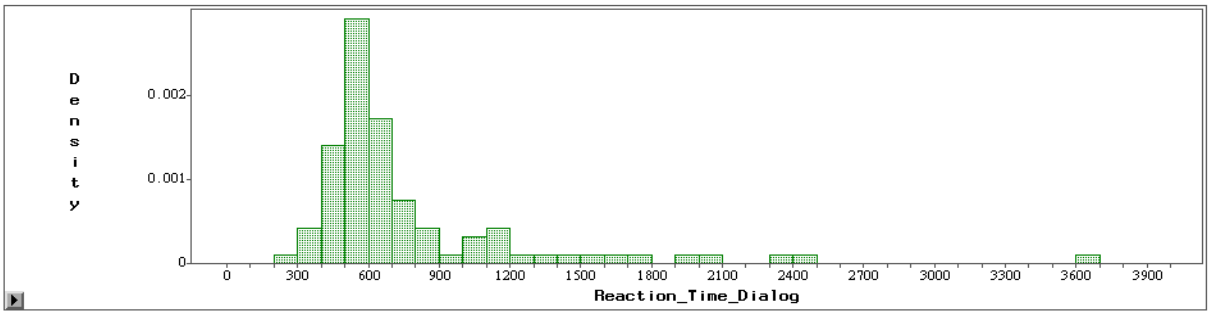


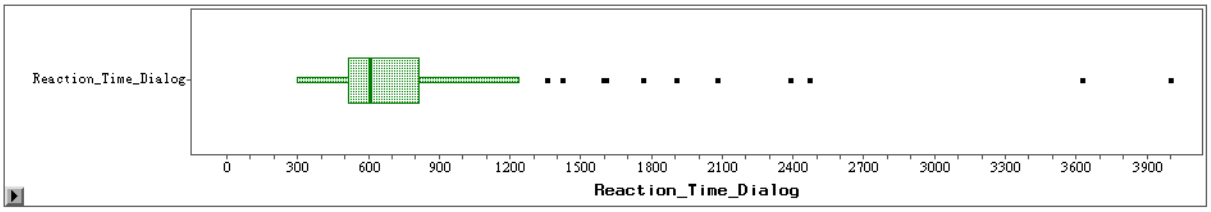
Moments			
N	86.0000	Sum Wgts	86.0000
Mean	723.0814	Sum	62185.0000
Std Dev	392.9301	Variance	154394.052
Skewness	2.4860	Kurtosis	6.7671
USS	58088311.0	CSS	13123494.4
CV	54.3411	Std Mean	42.3707

Quantiles			
100% Max	2437.0000	99.0%	2437.0000
75% Q3	766.0000	97.5%	2094.0000
50% Med	601.5000	95.0%	1593.0000
25% Q1	515.0000	90.0%	1266.0000
0% Min	328.0000	10.0%	422.0000
Range	2109.0000	5.0%	391.0000
Q3-Q1	251.0000	2.5%	344.0000
Mode	500.0000	1.0%	328.0000

Subject C PDT reaction times while the tester issues commands

Dialog vs. Manual

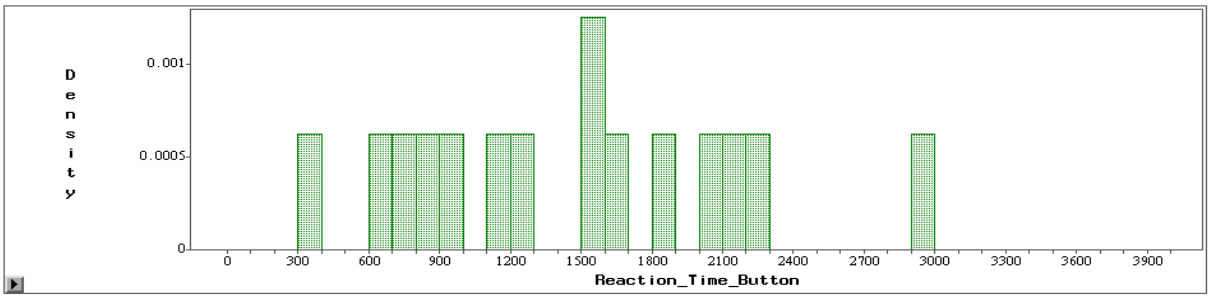


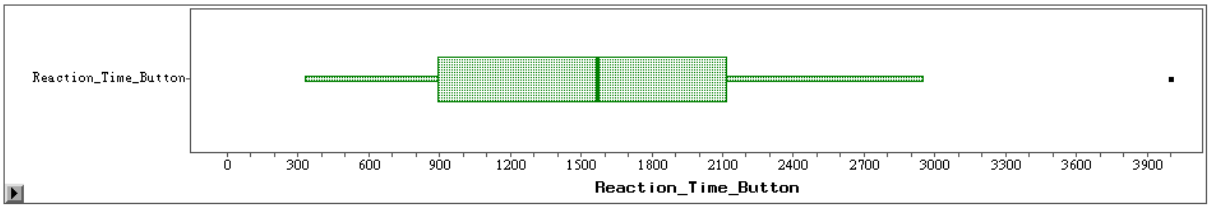


Moments			
N	92.0000	Sum Wgts	92.0000
Mean	817.1304	Sum	75176.0000
Std Dev	621.8319	Variance	386674.862
Skewness	3.1165	Kurtosis	11.4129
USS	96616010.0	CSS	35187412.4
CV	76.0995	Std Mean	64.8305

Quantiles			
100% Max	4000.0000	99.0%	4000.0000
75% Q3	812.0000	97.5%	2469.0000
50% Med	609.5000	95.0%	2078.0000
25% Q1	515.0000	90.0%	1422.0000
0% Min	297.0000	10.0%	438.0000
Range	3703.0000	5.0%	391.0000
Q3-Q1	297.0000	2.5%	328.0000
Mode	500.0000	1.0%	297.0000

Subject C PDT reaction times during any Jukebox interaction with speech



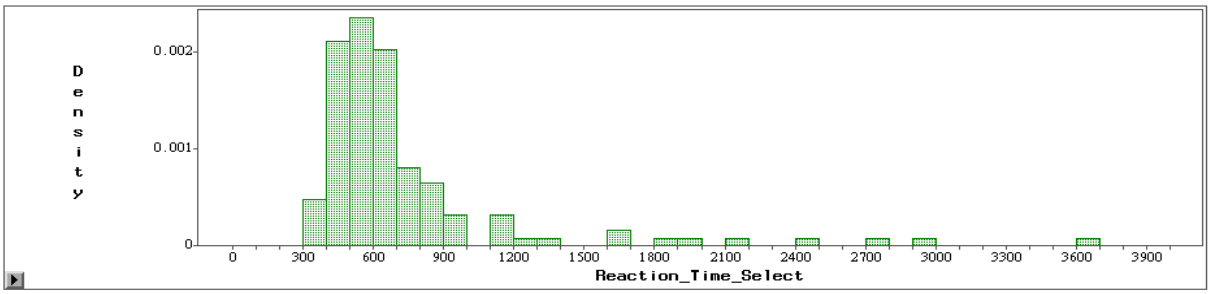


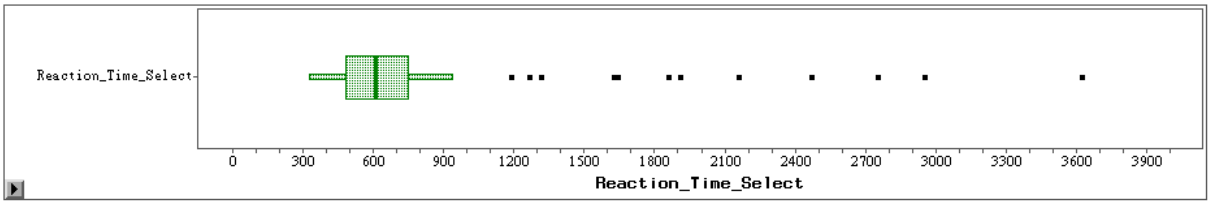
Moments			
N	16.0000	Sum Wgts	16.0000
Mean	1624.0000	Sum	25984.0000
Std Dev	938.6985	Variance	881154.933
Skewness	1.0591	Kurtosis	1.4607
USS	55415340.0	CSS	13217324.0
CV	57.8016	Std Mean	234.6746

Quantiles			
100% Max	4000.0000	99.0%	4000.0000
75% Q3	2117.5000	97.5%	4000.0000
50% Med	1570.5000	95.0%	4000.0000
25% Q1	890.5000	90.0%	2953.0000
0% Min	328.0000	10.0%	671.0000
Range	3672.0000	5.0%	328.0000
Q3-Q1	1227.0000	2.5%	328.0000
Mode	.	1.0%	328.0000

Subject C PDT reaction times during any Jukebox interaction with steering wheel button use

Selection vs. Question vs. Mixed

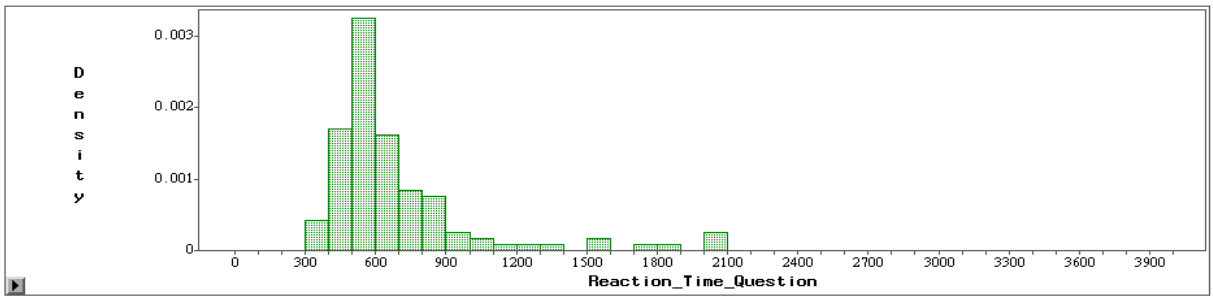


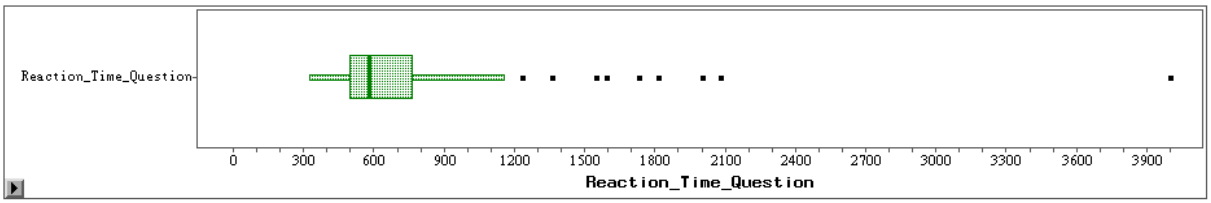


Moments			
N	123.0000	Sum Wgts	123.0000
Mean	747.8374	Sum	91984.0000
Std Dev	516.9183	Variance	267204.482
Skewness	3.2628	Kurtosis	12.1513
USS	101388022	CSS	32598946.7
CV	69.1217	Std Mean	46.6089

Quantiles			
100%	Max	3625.0000	99.0% 2953.0000
75%	Q3	750.0000	97.5% 2469.0000
50%	Med	610.0000	95.0% 1859.0000
25%	Q1	484.0000	90.0% 1188.0000
0%	Min	328.0000	10.0% 422.0000
	Range	3297.0000	5.0% 406.0000
	Q3-Q1	266.0000	2.5% 344.0000
	Mode	656.0000	1.0% 343.0000

Subject C PDT reaction times for tests with Jukebox interaction using plain selection



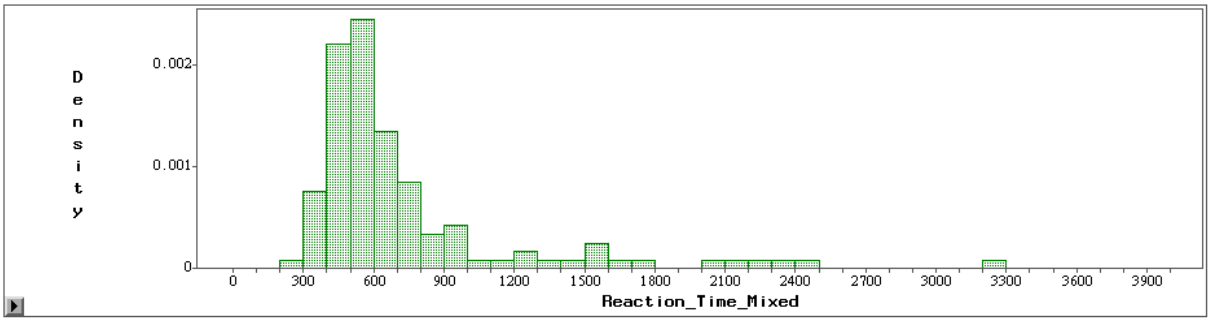


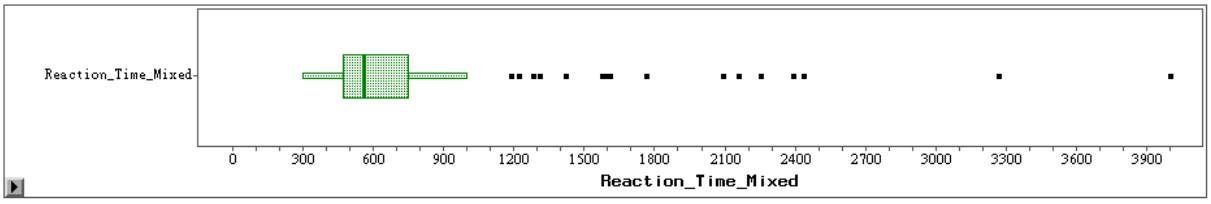
Moments			
N	117.0000	Sum Wgts	117.0000
Mean	719.0256	Sum	84126.0000
Std Dev	461.6597	Variance	213129.680
Skewness	4.0786	Kurtosis	22.8026
USS	85211794.0	CSS	24723042.9
CV	64.2063	Std Mean	42.6805

Quantiles			
100%	Max	4000.0000	99.0% 2078.0000
75%	Q3	765.0000	97.5% 2078.0000
50%	Med	579.0000	95.0% 1734.0000
25%	Q1	500.0000	90.0% 1063.0000
0%	Min	328.0000	10.0% 422.0000
	Range	3672.0000	5.0% 406.0000
	Q3-Q1	265.0000	2.5% 391.0000
	Mode	500.0000	1.0% 344.0000

Subject C PDT reaction times for tests with Jukebox interaction using questions and user assignments

Selection vs. Question vs. Mixed (cont'd)





Moments

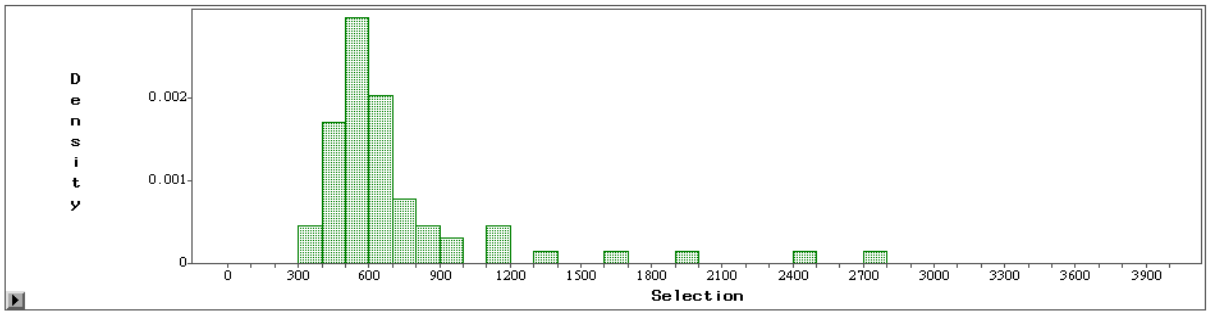
N	118.0000	Sum Wgts	118.0000
Mean	772.6441	Sum	91172.0000
Std Dev	580.1636	Variance	336589.821
Skewness	3.0316	Kurtosis	10.9799
USS	109824514	CSS	39381009.1
CV	75.0881	Std Mean	53.4084

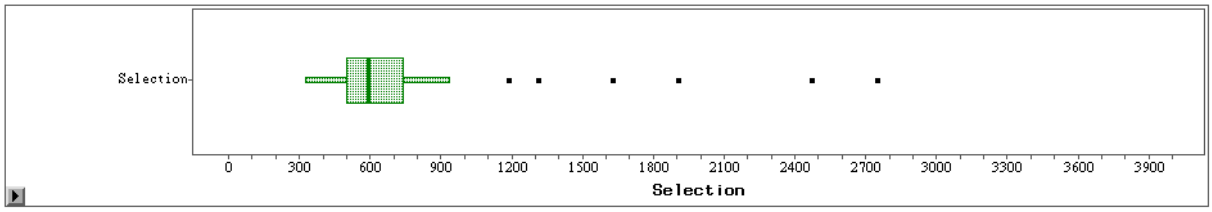
Quantiles

100% Max	4000.0000	99.0%	3266.0000
75% Q3	750.0000	97.5%	2437.0000
50% Med	563.0000	95.0%	2157.0000
25% Q1	469.0000	90.0%	1578.0000
0% Min	297.0000	10.0%	406.0000
Range	3703.0000	5.0%	375.0000
Q3-Q1	281.0000	2.5%	344.0000
Mode	500.0000	1.0%	313.0000

Subject C PDT reaction times for tests with Jukebox interaction using a mix of questions and selection methods

Selection vs. Selection w/ Buttons

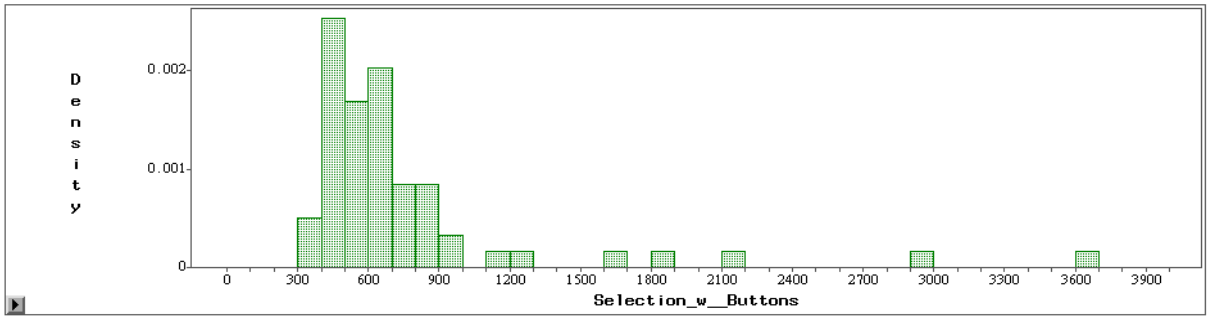


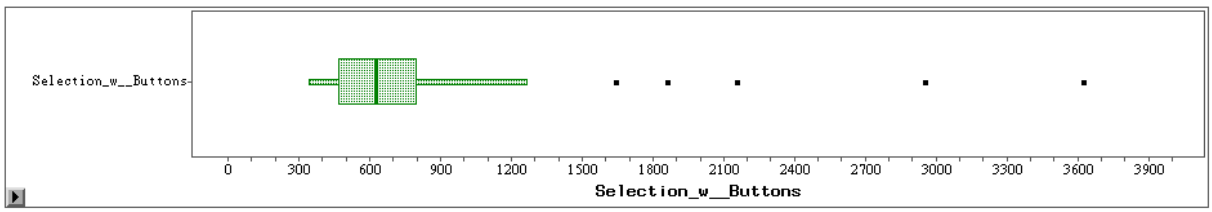


Moments			
N	64.0000	Sum Wgts	64.0000
Mean	725.6406	Sum	46441.0000
Std Dev	445.1258	Variance	198136.996
Skewness	2.9641	Kurtosis	9.7657
USS	46182107.0	CSS	12482630.7
CV	61.3425	Std Mean	55.6407

Quantiles			
100% Max	2750.0000	99.0%	2750.0000
75% Q3	742.5000	97.5%	2469.0000
50% Med	594.0000	95.0%	1625.0000
25% Q1	500.0000	90.0%	1187.0000
0% Min	328.0000	10.0%	438.0000
Range	2422.0000	5.0%	421.0000
Q3-Q1	242.5000	2.5%	344.0000
Mode	500.0000	1.0%	328.0000

Subject C PDT reaction times for the "selection" test – speech only



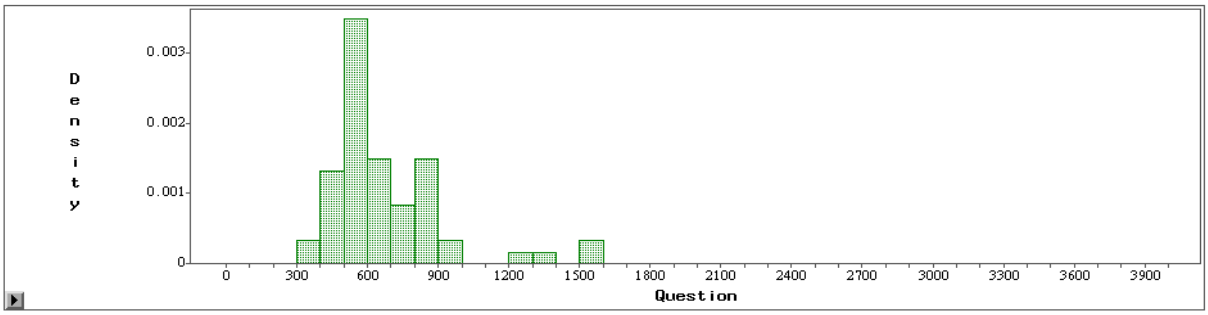


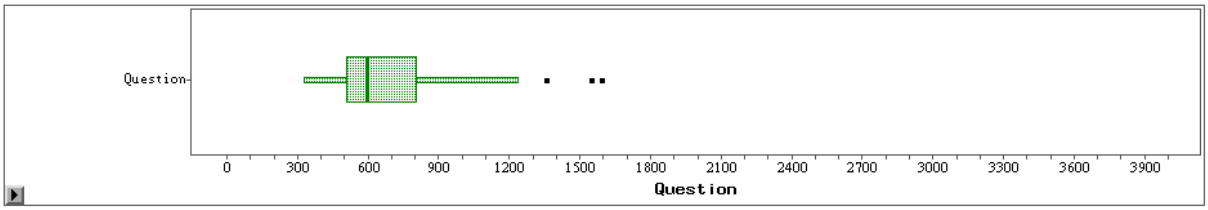
Moments			
N	59.0000	Sum Wgts	59.0000
Mean	771.9153	Sum	45543.0000
Std Dev	587.9623	Variance	345699.631
Skewness	3.2940	Kurtosis	12.0263
USS	55205915.0	CSS	20050578.6
CV	76.1693	Std Mean	76.5462

Quantiles			
100% Max	3625.0000	99.0%	3625.0000
75% Q3	797.0000	97.5%	2953.0000
50% Med	625.0000	95.0%	2157.0000
25% Q1	469.0000	90.0%	1266.0000
0% Min	343.0000	10.0%	421.0000
Range	3282.0000	5.0%	375.0000
Q3-Q1	328.0000	2.5%	344.0000
Mode	656.0000	1.0%	343.0000

Subject C PDT reaction times for the "selection" test – buttons when possible

Question vs. question w/ Buttons

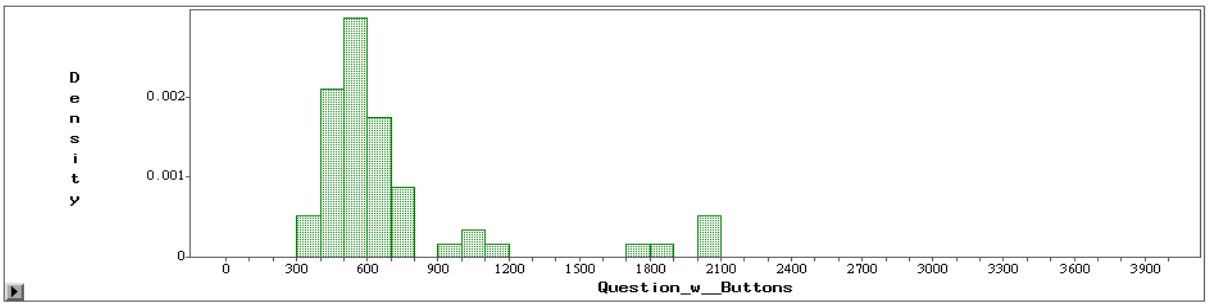


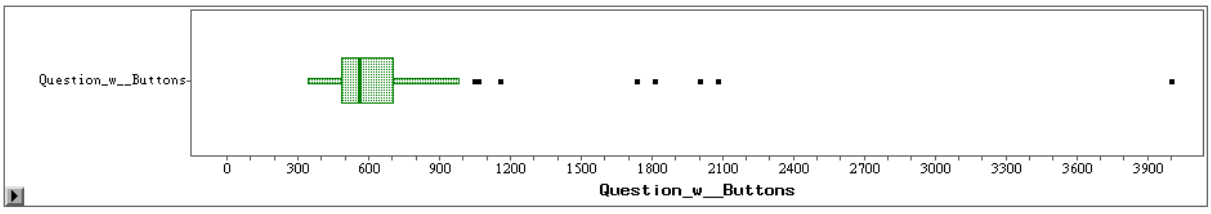


Moments			
N	60.0000	Sum Wgts	60.0000
Mean	672.6500	Sum	40359.0000
Std Dev	256.4102	Variance	65746.1975
Skewness	1.8956	Kurtosis	4.2037
USS	31026507.0	CSS	3879025.65
CV	38.1194	Std Mean	33.1024

Quantiles			
100% Max	1593.0000	99.0%	1593.0000
75% Q3	804.5000	97.5%	1547.0000
50% Med	594.0000	95.0%	1297.0000
25% Q1	507.5000	90.0%	906.5000
0% Min	328.0000	10.0%	438.0000
Range	1265.0000	5.0%	414.0000
Q3-Q1	297.0000	2.5%	391.0000
Mode	500.0000	1.0%	328.0000

Subject C PDT reaction times for the "question" test – speech only



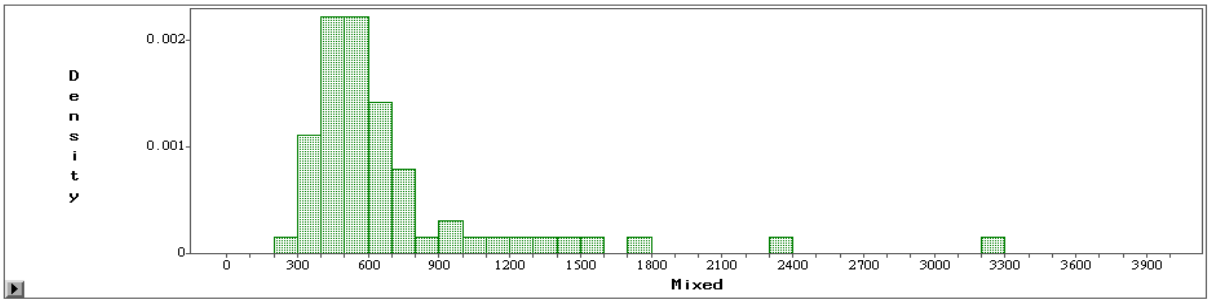


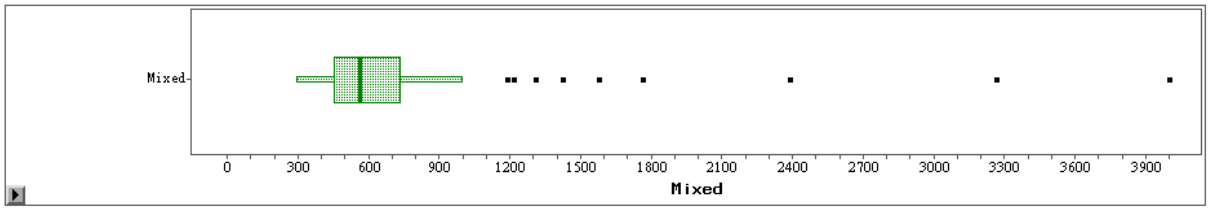
Moments			
N	57.0000	Sum Wgts	57.0000
Mean	767.8421	Sum	43767.0000
Std Dev	606.2051	Variance	367484.671
Skewness	3.4442	Kurtosis	14.4515
USS	54185287.0	CSS	20579141.6
CV	78.9492	Std Mean	80.2938

Quantiles			
100% Max	4000.0000	99.0%	4000.0000
75% Q3	703.0000	97.5%	2078.0000
50% Med	563.0000	95.0%	2078.0000
25% Q1	484.0000	90.0%	1734.0000
0% Min	344.0000	10.0%	406.0000
Range	3656.0000	5.0%	391.0000
Q3-Q1	219.0000	2.5%	391.0000
Mode	406.0000	1.0%	344.0000

Subject C PDT reaction times for the "question" test – buttons when possible

Mixed vs. Mixed w/ Buttons

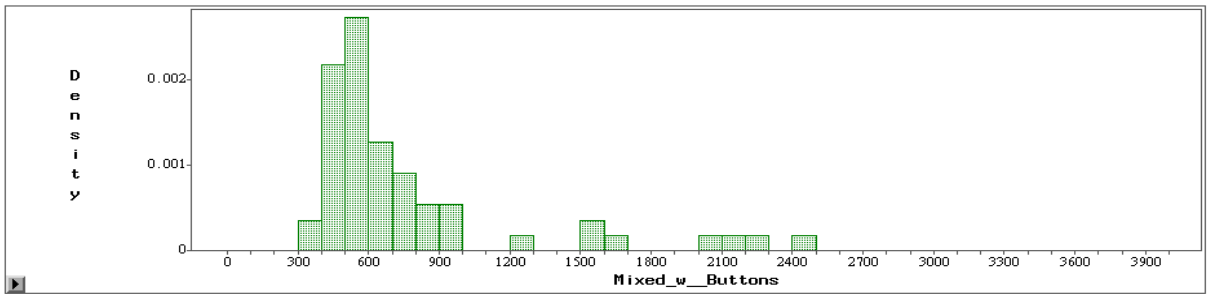


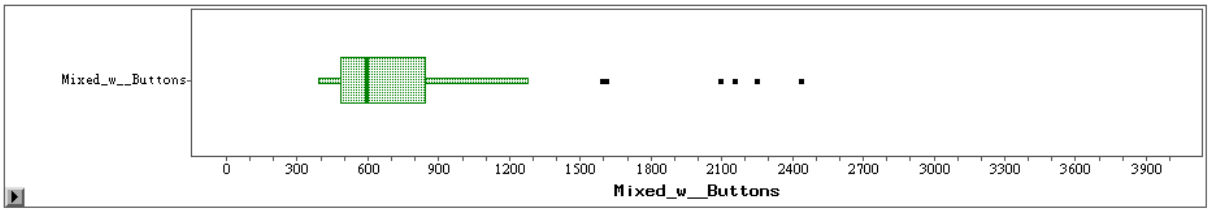


Moments			
N	63.0000	Sum Wgts	63.0000
Mean	763.4286	Sum	48096.0000
Std Dev	646.9809	Variance	418584.249
Skewness	3.3739	Kurtosis	12.8929
USS	62670084.0	CSS	25952223.4
CV	84.7467	Std Mean	81.5119

Quantiles			
100%	Max	4000.0000	99.0% 4000.0000
75%	Q3	735.0000	97.5% 3266.0000
50%	Med	563.0000	95.0% 1766.0000
25%	Q1	453.0000	90.0% 1312.0000
0%	Min	297.0000	10.0% 375.0000
	Range	3703.0000	5.0% 344.0000
	Q3-Q1	282.0000	2.5% 313.0000
	Mode	500.0000	1.0% 297.0000

Subject C PDT reaction times for the "mixed" test – speech only





Moments			
N	55.0000	Sum Wgts	55.0000
Mean	783.2000	Sum	43076.0000
Std Dev	498.4663	Variance	248468.644
Skewness	2.0995	Kurtosis	3.6153
USS	47154430.0	CSS	13417306.8
CV	63.6448	Std Mean	67.2132

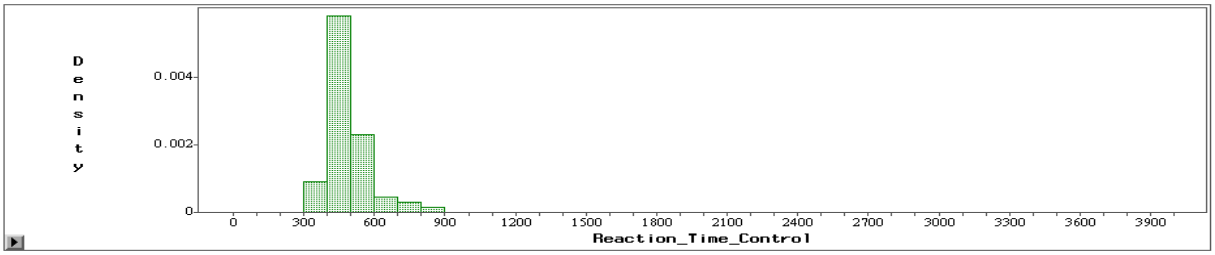
Quantiles			
100%	Max	2437.0000	99.0% 2437.0000
75%	Q3	844.0000	97.5% 2250.0000
50%	Med	593.0000	95.0% 2157.0000
25%	Q1	485.0000	90.0% 1594.0000
0%	Min	391.0000	10.0% 453.0000
	Range	2046.0000	5.0% 437.0000
	Q3-Q1	359.0000	2.5% 391.0000
	Mode	500.0000	1.0% 391.0000

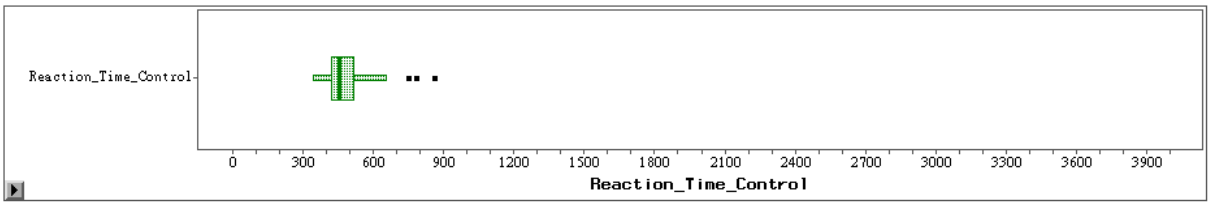
Subject C PDT reaction times for the "mixed" test – buttons when possible

Appendix K: Subject D Test Results

This appendix shows the PDT reaction time test results for Subject D.

Control vs. Overall

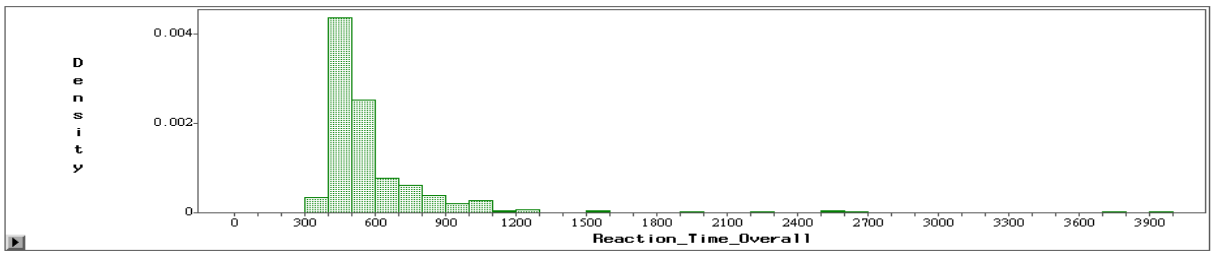


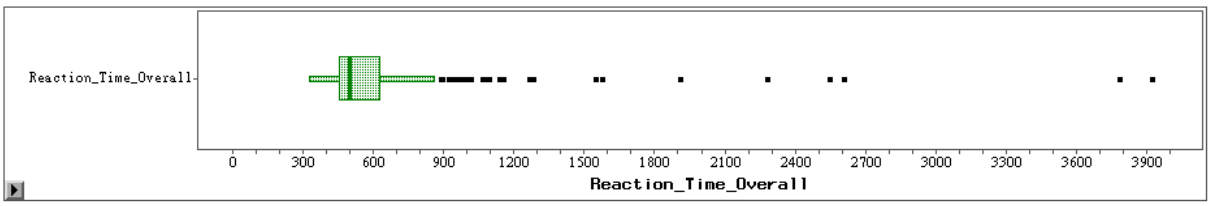


Moments			
N	65.0000	Sum Wgts	65.0000
Mean	480.5231	Sum	31234.0000
Std Dev	95.0248	Variance	9029.7221
Skewness	1.9630	Kurtosis	4.7601
USS	15586560.0	CSS	577902.215
CV	19.7753	Std Mean	11.7864

Quantiles			
100%	Max	859.0000	99.0% 859.0000
75%	Q3	516.0000	97.5% 781.0000
50%	Med	453.0000	95.0% 657.0000
25%	Q1	422.0000	90.0% 579.0000
0%	Min	344.0000	10.0% 406.0000
	Range	515.0000	5.0% 375.0000
	Q3-Q1	94.0000	2.5% 359.0000
	Mode	453.0000	1.0% 344.0000

Subject D PDT reaction times for the control test – no Jukebox interaction



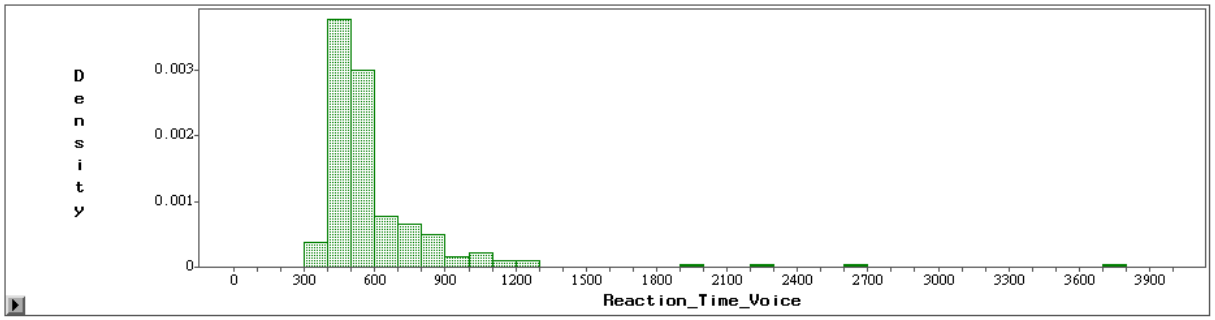


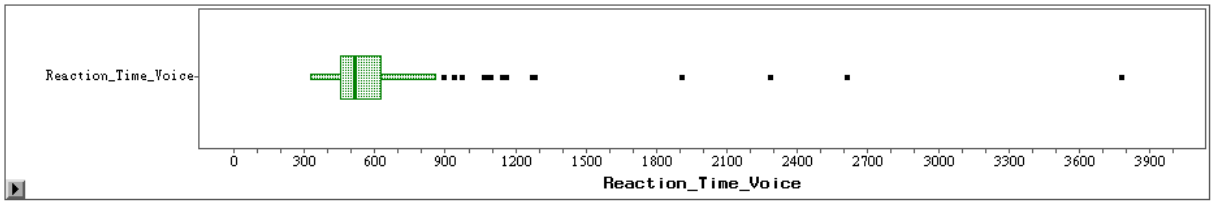
Moments			
N	357.0000	Sum Wgts	357.0000
Mean	608.9524	Sum	217396.000
Std Dev	376.0423	Variance	141407.815
Skewness	5.3737	Kurtosis	36.9305
USS	182724994	CSS	50341182.2
CV	61.7523	Std Mean	19.9023

Quantiles			
100%	Max	3922.0000	99.0% 2547.0000
75%	Q3	625.0000	97.5% 1547.0000
50%	Med	500.0000	95.0% 1063.0000
25%	Q1	453.0000	90.0% 860.0000
0%	Min	329.0000	10.0% 422.0000
	Range	3593.0000	5.0% 406.0000
	Q3-Q1	172.0000	2.5% 391.0000
	Mode	453.0000	1.0% 375.0000

Subject D PDT reaction times for all tests with Jukebox interaction

Speech vs. Button





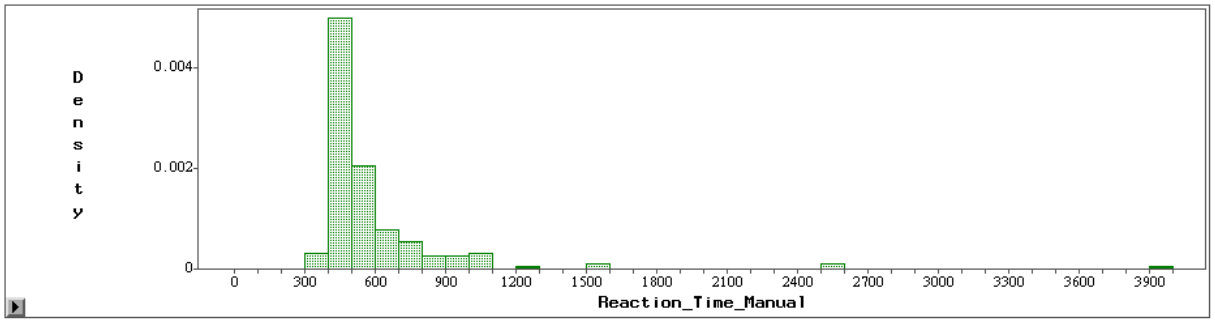
Moments

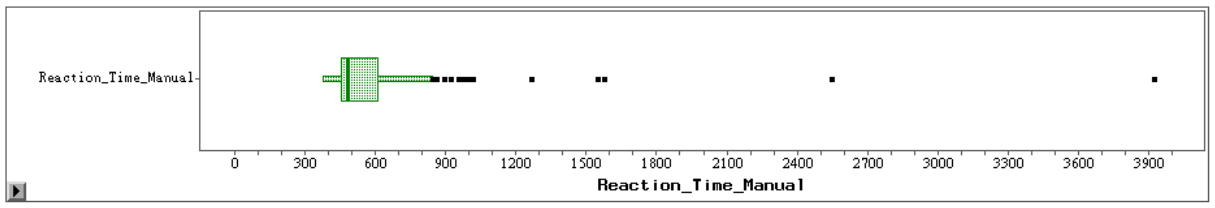
N	179.0000	Sum Wgts	179.0000
Mean	614.3408	Sum	109967.000
Std Dev	372.1041	Variance	138461.462
Skewness	5.2460	Kurtosis	35.6839
USS	92203353.0	CSS	24646140.2
CV	60.5697	Std Mean	27.8124

Quantiles

100% Max	3781.0000	99.0%	2609.0000
75% Q3	625.0000	97.5%	1282.0000
50% Med	515.0000	95.0%	1094.0000
25% Q1	453.0000	90.0%	860.0000
0% Min	329.0000	10.0%	422.0000
Range	3452.0000	5.0%	406.0000
Q3-Q1	172.0000	2.5%	390.0000
Mode	453.0000	1.0%	359.0000

Subject D PDT reaction times for tests using speech input only





Moments

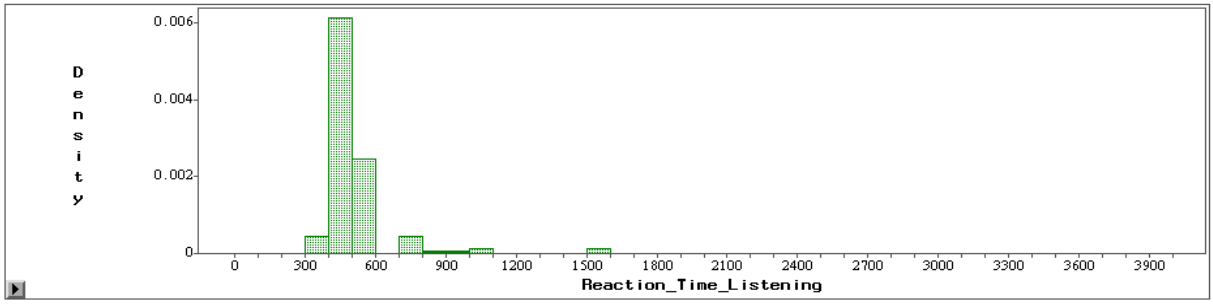
N	178.0000	Sum Wgts	178.0000
Mean	603.5337	Sum	107429.000
Std Dev	380.9342	Variance	145110.838
Skewness	5.5407	Kurtosis	39.1376
USS	90521641.0	CSS	25684618.3
CV	63.1173	Std Mean	28.5522

Quantiles

100% Max	3922.0000	99.0%	2547.0000
75% Q3	609.0000	97.5%	1547.0000
50% Med	485.0000	95.0%	1015.0000
25% Q1	453.0000	90.0%	891.0000
0% Min	375.0000	10.0%	422.0000
Range	3547.0000	5.0%	406.0000
Q3-Q1	156.0000	2.5%	391.0000
Mode	453.0000	1.0%	375.0000

Subject D PDT reaction times for tests using buttoned input when possible

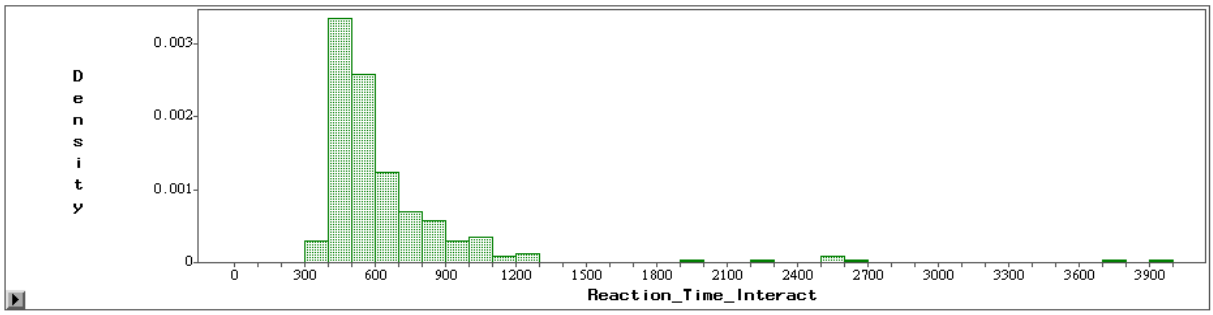
Listening vs. Interaction



Moments			
N	133.0000	Sum Wgts	133.0000
Mean	510.8722	Sum	67946.0000
Std Dev	171.2238	Variance	29317.5820
Skewness	4.2273	Kurtosis	21.4724
USS	38581642.0	CSS	3869920.83
CV	33.5160	Std Mean	14.8470

Quantiles			
100% Max	1578.0000	99.0%	1547.0000
75% Q3	515.0000	97.5%	1016.0000
50% Med	469.0000	95.0%	765.0000
25% Q1	438.0000	90.0%	594.0000
0% Min	329.0000	10.0%	422.0000
Range	1249.0000	5.0%	406.0000
Q3-Q1	77.0000	2.5%	391.0000
Mode	453.0000	1.0%	375.0000

Subject D PDT reaction times while only music is playing – no active interaction

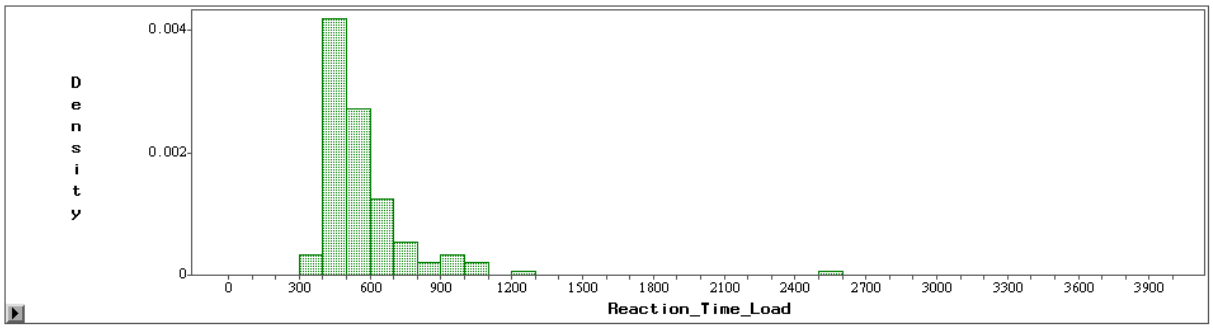


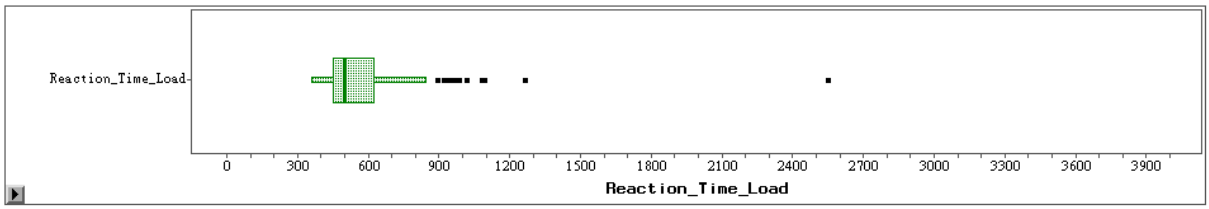
Moments			
N	224.0000	Sum Wgts	224.0000
Mean	667.1875	Sum	149450.000
Std Dev	446.3714	Variance	199247.445
Skewness	4.6731	Kurtosis	26.6137
USS	144143352	CSS	44432180.1
CV	66.9034	Std Mean	29.8244

Quantiles			
100% Max	3922.0000	99.0%	2609.0000
75% Q3	695.5000	97.5%	2282.0000
50% Med	531.0000	95.0%	1140.0000
25% Q1	453.5000	90.0%	969.0000
0% Min	359.0000	10.0%	422.0000
Range	3563.0000	5.0%	406.0000
Q3-Q1	242.0000	2.5%	391.0000
Mode	453.0000	1.0%	375.0000

Subject D PDT reaction times during any type of Jukebox interaction – cognitive load, speech, and button

Cognitive Load vs. Command Issue

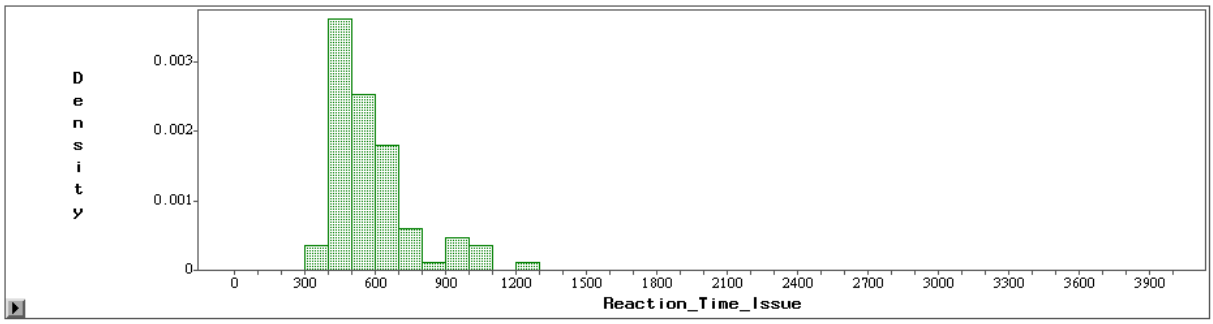


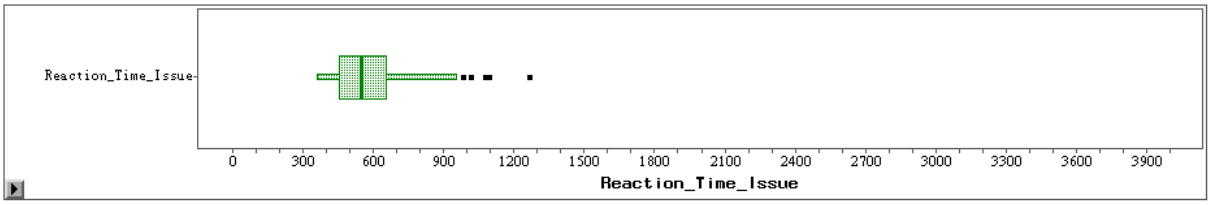


Moments			
N	143.0000	Sum Wgts	143.0000
Mean	571.9720	Sum	81792.0000
Std Dev	232.8346	Variance	54211.9710
Skewness	4.8865	Kurtosis	36.3310
USS	54480836.0	CSS	7698099.89
CV	40.7073	Std Mean	19.4706

Quantiles			
100% Max	2546.0000	99.0%	1266.0000
75% Q3	625.0000	97.5%	1078.0000
50% Med	500.0000	95.0%	954.0000
25% Q1	453.0000	90.0%	797.0000
0% Min	359.0000	10.0%	422.0000
Range	2187.0000	5.0%	406.0000
Q3-Q1	172.0000	2.5%	391.0000
Mode	453.0000	1.0%	360.0000

Subject D PDT reaction times during cognitive load conditions



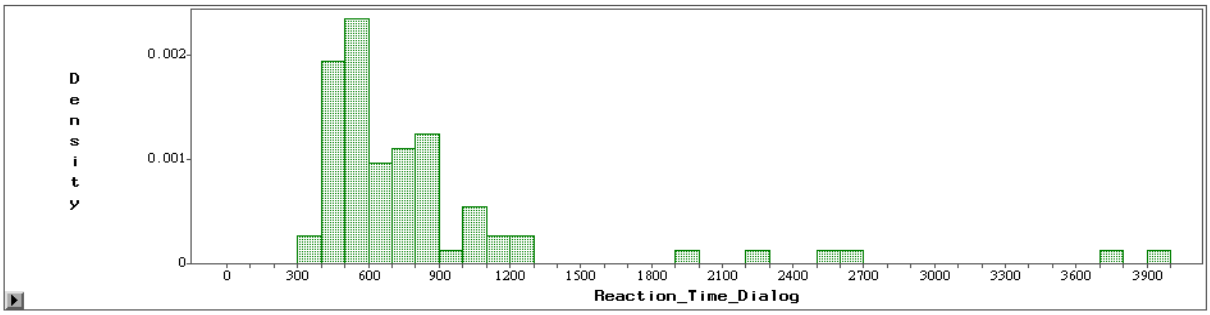


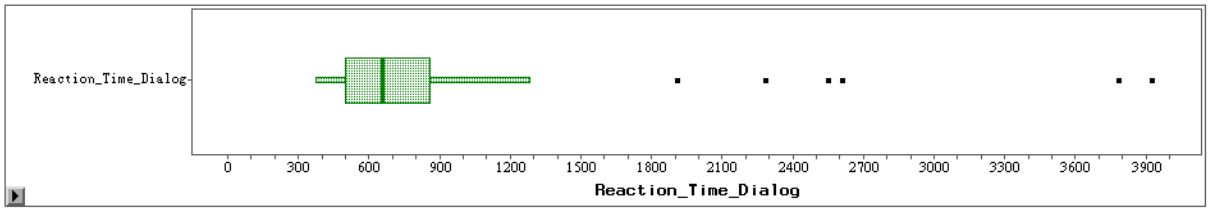
Moments			
N	83.0000	Sum Wgts	83.0000
Mean	582.4578	Sum	48344.0000
Std Dev	184.3510	Variance	33985.3000
Skewness	1.5759	Kurtosis	2.4085
USS	30945136.0	CSS	2786794.60
CV	31.6505	Std Mean	20.2352

Quantiles			
100% Max	1266.0000	99.0%	1266.0000
75% Q3	656.0000	97.5%	1078.0000
50% Med	547.0000	95.0%	984.0000
25% Q1	453.0000	90.0%	828.0000
0% Min	360.0000	10.0%	421.0000
Range	906.0000	5.0%	406.0000
Q3-Q1	203.0000	2.5%	391.0000
Mode	453.0000	1.0%	360.0000

Subject D PDT reaction times while the tester issues commands

Dialog vs. Manual

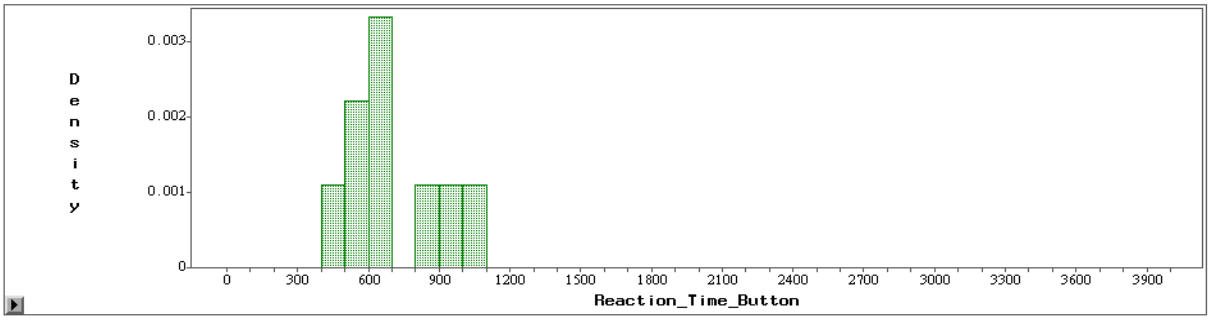


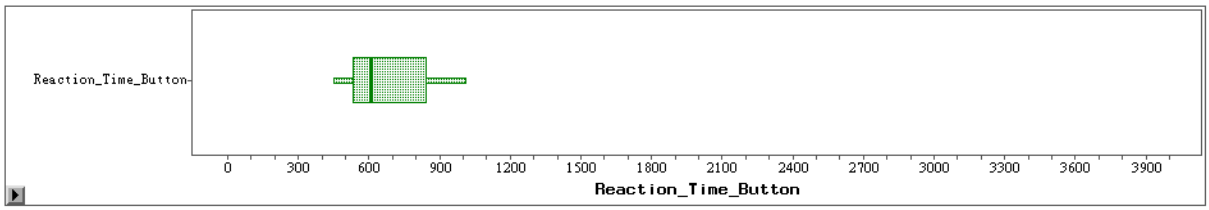


Moments			
N	72.0000	Sum Wgts	72.0000
Mean	853.7778	Sum	61472.0000
Std Dev	677.5686	Variance	459099.274
Skewness	3.1094	Kurtosis	10.3873
USS	85079476.0	CSS	32596048.4
CV	79.3612	Std Mean	79.8522

Quantiles			
100% Max	3922.0000	99.0%	3922.0000
75% Q3	859.0000	97.5%	3781.0000
50% Med	656.5000	95.0%	2547.0000
25% Q1	500.0000	90.0%	1266.0000
0% Min	375.0000	10.0%	437.0000
Range	3547.0000	5.0%	407.0000
Q3-Q1	359.0000	2.5%	390.0000
Mode	500.0000	1.0%	375.0000

Subject D PDT reaction times during any Jukebox interaction with speech



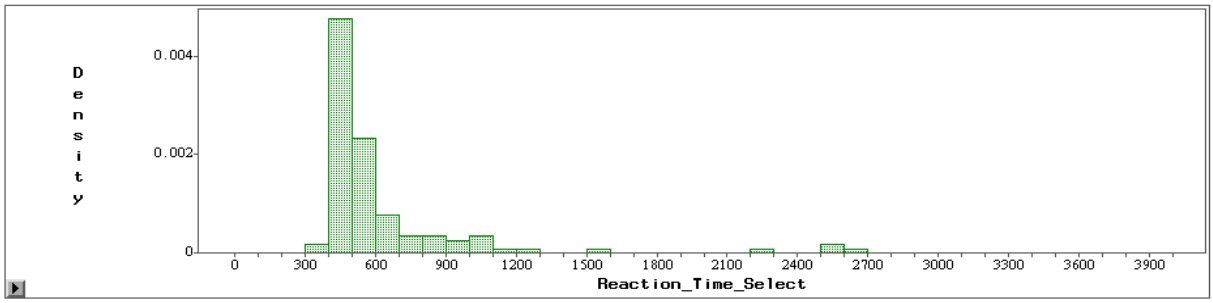


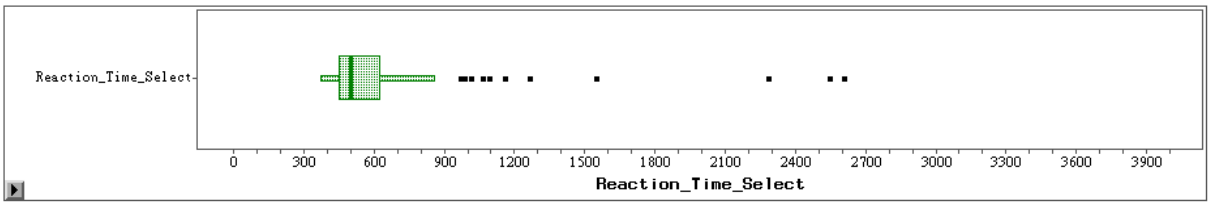
Moments			
N	9.0000	Sum Wgts	9.0000
Mean	687.3333	Sum	6186.0000
Std Dev	203.4687	Variance	41399.5000
Skewness	0.7567	Kurtosis	-1.0200
USS	4583040.00	CSS	331196.000
CV	29.6026	Std Mean	67.8229

Quantiles			
100% Max	1015.0000	99.0%	1015.0000
75% Q3	844.0000	97.5%	1015.0000
50% Med	609.0000	95.0%	1015.0000
25% Q1	531.0000	90.0%	1015.0000
0% Min	453.0000	10.0%	453.0000
Range	562.0000	5.0%	453.0000
Q3-Q1	313.0000	2.5%	453.0000
Mode	531.0000	1.0%	453.0000

Subject D PDT reaction times during any Jukebox interaction with steering wheel button use

Selection vs. Question vs. Mixed

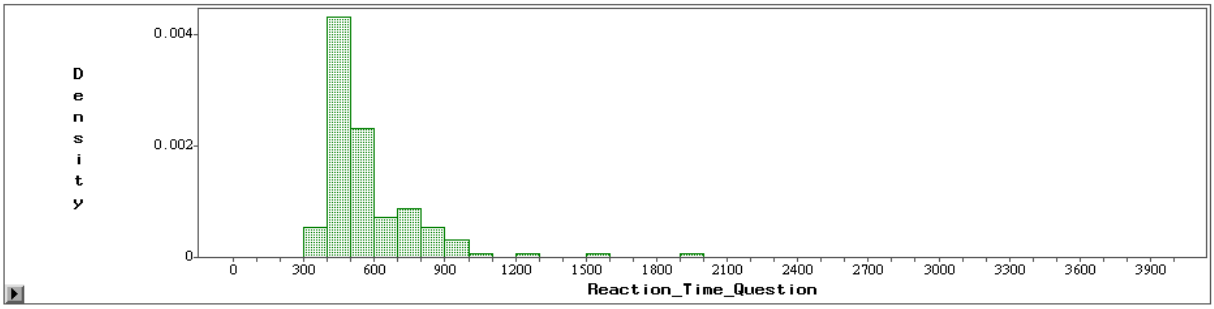


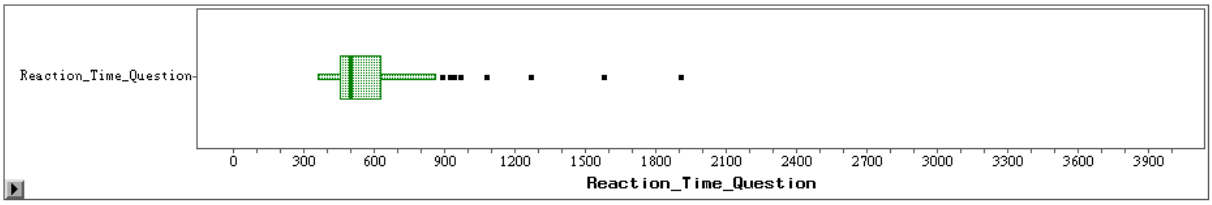


Moments			
N	115.0000	Sum Wgts	115.0000
Mean	635.0870	Sum	73035.0000
Std Dev	408.9092	Variance	167206.747
Skewness	3.5683	Kurtosis	13.5967
USS	65445145.0	CSS	19061569.1
CV	64.3863	Std Mean	38.1310

Quantiles			
100% Max	2609.0000	99.0%	2547.0000
75% Q3	625.0000	97.5%	2546.0000
50% Med	500.0000	95.0%	1266.0000
25% Q1	453.0000	90.0%	984.0000
0% Min	375.0000	10.0%	437.0000
Range	2234.0000	5.0%	422.0000
Q3-Q1	172.0000	2.5%	406.0000
Mode	453.0000	1.0%	391.0000

Subject D PDT reaction times for tests with Jukebox interaction using plain selection



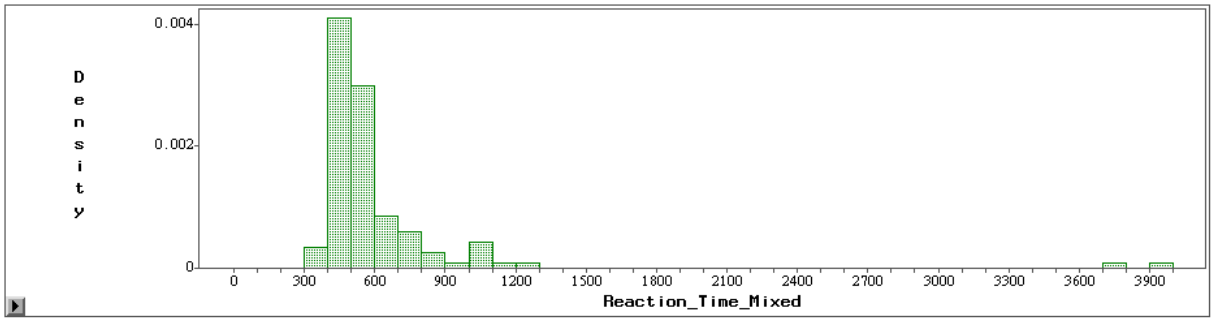


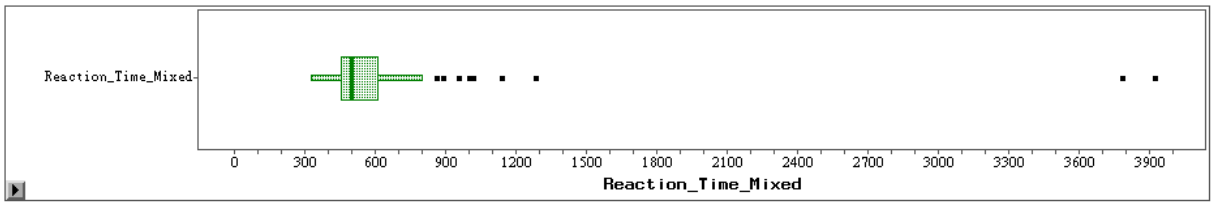
Moments			
N	125.0000	Sum Wgts	125.0000
Mean	576.4880	Sum	72061.0000
Std Dev	223.8382	Variance	50103.5422
Skewness	2.9585	Kurtosis	12.3597
USS	47755141.0	CSS	6212839.23
CV	38.8279	Std Mean	20.0207

Quantiles			
100% Max	1906.0000	99.0%	1578.0000
75% Q3	625.0000	97.5%	1078.0000
50% Med	500.0000	95.0%	937.0000
25% Q1	453.0000	90.0%	829.0000
0% Min	360.0000	10.0%	422.0000
Range	1546.0000	5.0%	391.0000
Q3-Q1	172.0000	2.5%	375.0000
Mode	453.0000	1.0%	375.0000

Subject D PDT reaction times for tests with Jukebox interaction using questions and user assignments

Selection vs. Question vs. Mixed (cont'd)





Moments

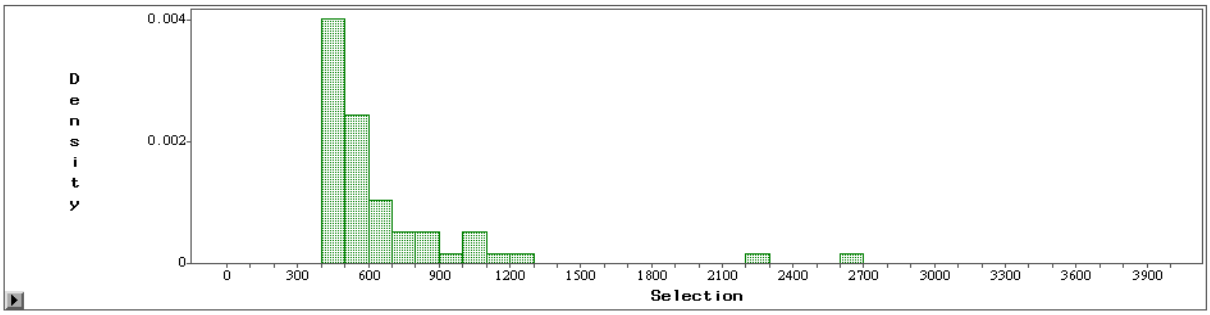
N	117.0000	Sum Wgts	117.0000
Mean	617.9487	Sum	72300.0000
Std Dev	462.8157	Variance	214198.411
Skewness	6.0357	Kurtosis	40.4035
USS	69524708.0	CSS	24847015.7
CV	74.8955	Std Mean	42.7873

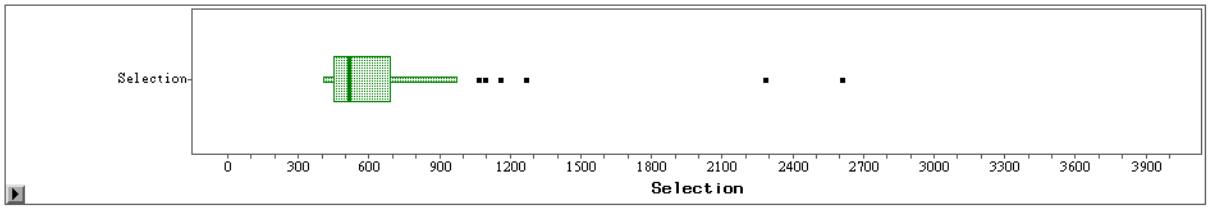
Quantiles

100%	Max	3922.0000	99.0% 3781.0000
75%	Q3	609.0000	97.5% 1282.0000
50%	Med	500.0000	95.0% 1015.0000
25%	Q1	453.0000	90.0% 890.0000
0%	Min	329.0000	10.0% 422.0000
	Range	3593.0000	5.0% 406.0000
	Q3-Q1	156.0000	2.5% 390.0000
	Mode	453.0000	1.0% 359.0000

Subject D PDT reaction times for tests with Jukebox interaction using a mix of questions and selection methods

Selection vs. Selection w/ Buttons

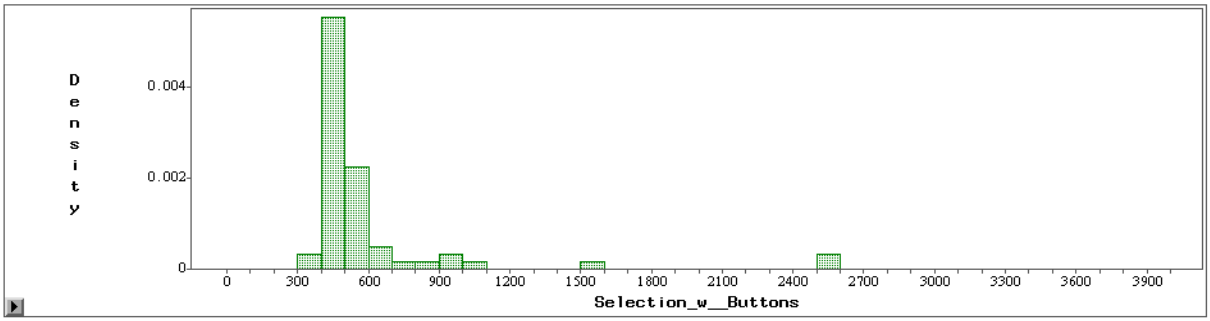


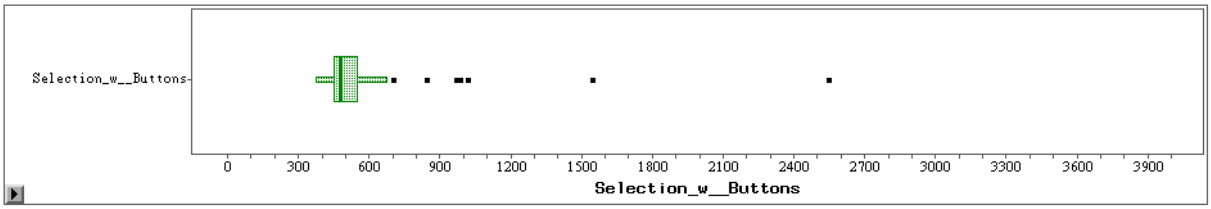


Moments			
N	57.0000	Sum Wgts	57.0000
Mean	663.4211	Sum	37815.0000
Std Dev	402.6215	Variance	162104.034
Skewness	3.3858	Kurtosis	13.1296
USS	34165093.0	CSS	9077825.89
CV	60.6887	Std Mean	53.3285

Quantiles			
100%	Max	2609.0000	99.0% 2609.0000
75%	Q3	688.0000	97.5% 2282.0000
50%	Med	516.0000	95.0% 1266.0000
25%	Q1	453.0000	90.0% 1093.0000
0%	Min	406.0000	10.0% 437.0000
	Range	2203.0000	5.0% 422.0000
	Q3-Q1	235.0000	2.5% 422.0000
	Mode	453.0000	1.0% 406.0000

Subject D PDT reaction times for the "selection" test – speech only



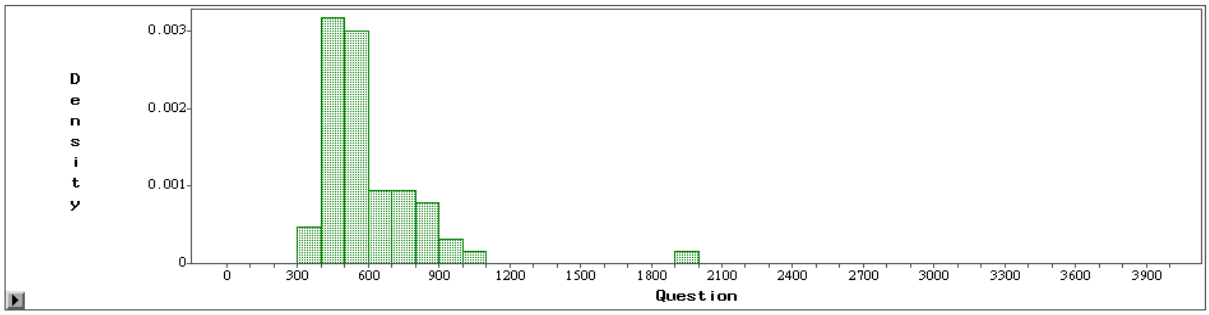


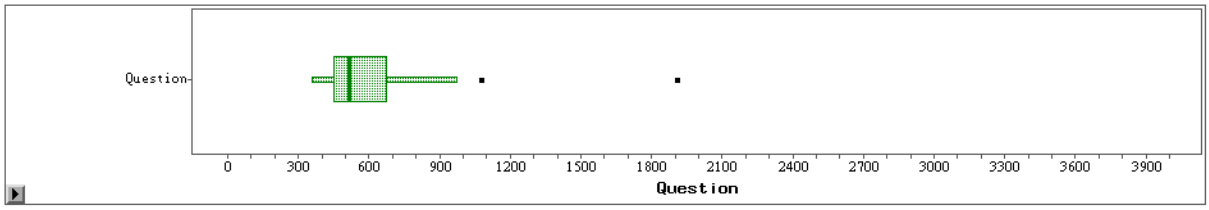
Moments			
N	58.0000	Sum Wgts	58.0000
Mean	607.2414	Sum	35220.0000
Std Dev	416.6072	Variance	173561.590
Skewness	3.8807	Kurtosis	15.7044
USS	31280052.0	CSS	9893010.62
CV	68.6065	Std Mean	54.7032

Quantiles			
100%	Max	2547.0000	99.0% 2547.0000
75%	Q3	547.0000	97.5% 2546.0000
50%	Med	476.5000	95.0% 1547.0000
25%	Q1	453.0000	90.0% 969.0000
0%	Min	375.0000	10.0% 422.0000
	Range	2172.0000	5.0% 406.0000
	Q3-Q1	94.0000	2.5% 391.0000
	Mode	453.0000	1.0% 375.0000

Subject D PDT reaction times for the "selection" test – buttons when possible

Question vs. Question w/ Buttons

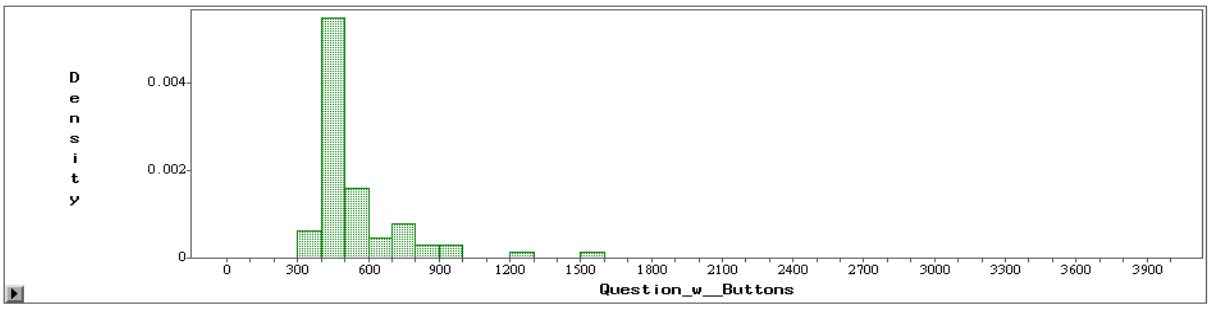


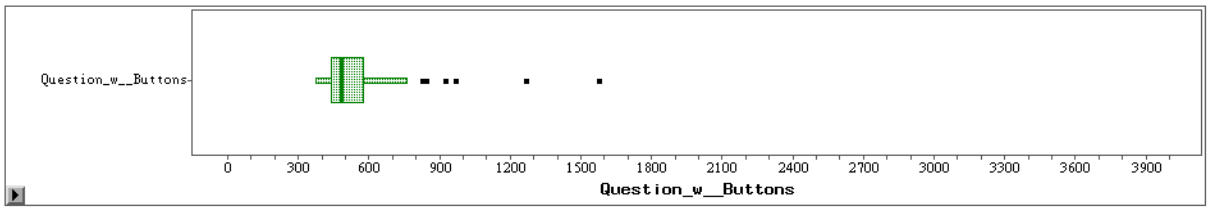


Moments			
N	63.0000	Sum Wgts	63.0000
Mean	597.9048	Sum	37668.0000
Std Dev	234.4157	Variance	54950.7327
Skewness	3.1490	Kurtosis	14.8517
USS	25928822.0	CSS	3406945.43
CV	39.2062	Std Mean	29.5336

Quantiles			
100% Max	1906.0000	99.0%	1906.0000
75% Q3	672.0000	97.5%	1078.0000
50% Med	516.0000	95.0%	937.0000
25% Q1	453.0000	90.0%	859.0000
0% Min	360.0000	10.0%	422.0000
Range	1546.0000	5.0%	406.0000
Q3-Q1	219.0000	2.5%	375.0000
Mode	437.0000	1.0%	360.0000

Subject D PDT reaction times for the "question" test – speech only



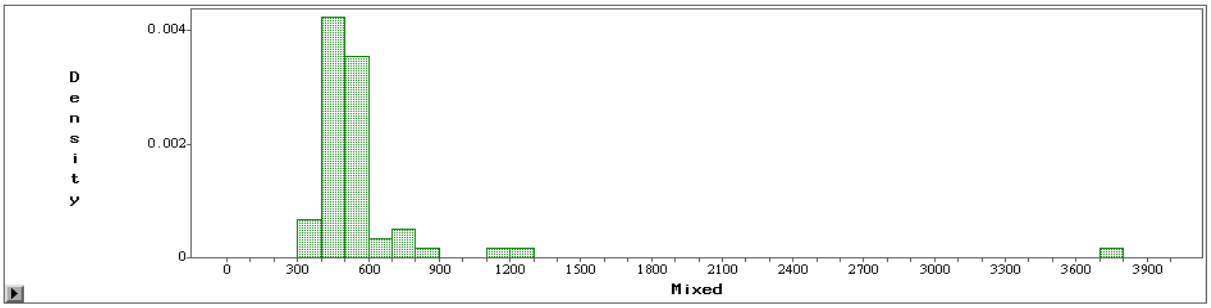


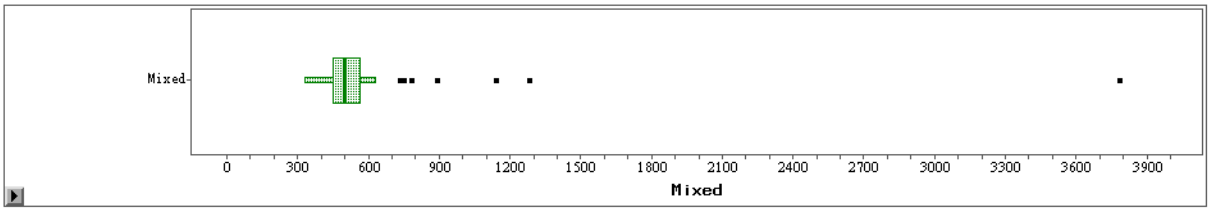
Moments			
N	62.0000	Sum Wgts	62.0000
Mean	554.7258	Sum	34393.0000
Std Dev	212.2338	Variance	45043.1859
Skewness	2.7890	Kurtosis	9.5880
USS	21826319.0	CSS	2747634.34
CV	38.2592	Std Mean	26.9537

Quantiles			
100% Max	1578.0000	99.0%	1578.0000
75% Q3	578.0000	97.5%	1266.0000
50% Med	484.0000	95.0%	922.0000
25% Q1	438.0000	90.0%	765.0000
0% Min	375.0000	10.0%	407.0000
Range	1203.0000	5.0%	391.0000
Q3-Q1	140.0000	2.5%	375.0000
Mode	469.0000	1.0%	375.0000

Subject D PDT reaction times for the "question" test – buttons when possible

Mixed vs. Mixed w/ Buttons

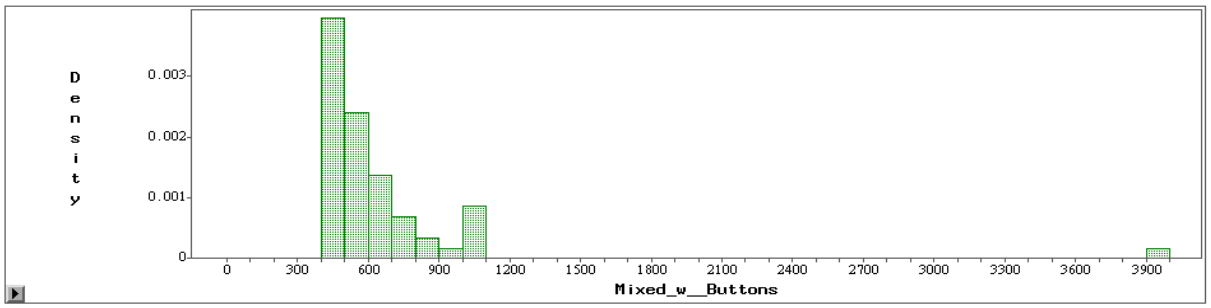


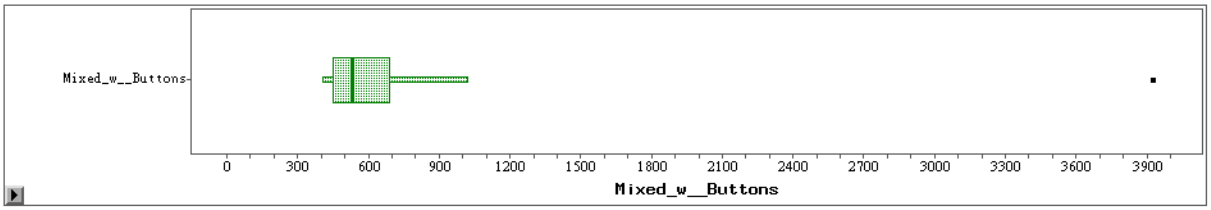


Moments			
N	59.0000	Sum Wgts	59.0000
Mean	584.4746	Sum	34484.0000
Std Dev	453.9941	Variance	206110.633
Skewness	6.3119	Kurtosis	43.9946
USS	32109438.0	CSS	11954416.7
CV	77.6756	Std Mean	59.1050

Quantiles			
100%	Max	3781.0000	99.0% 3781.0000
75%	Q3	562.0000	97.5% 1282.0000
50%	Med	500.0000	95.0% 1140.0000
25%	Q1	453.0000	90.0% 750.0000
0%	Min	329.0000	10.0% 406.0000
	Range	3452.0000	5.0% 390.0000
	Q3-Q1	109.0000	2.5% 359.0000
	Mode	453.0000	1.0% 329.0000

Subject D PDT reaction times for the "mixed" test – speech only





Moments			
N	58.0000	Sum Wgts	58.0000
Mean	652.0000	Sum	37816.0000
Std Dev	473.1240	Variance	223846.281
Skewness	6.0089	Kurtosis	41.3688
USS	37415270.0	CSS	12759238.0
CV	72.5650	Std Mean	62.1242

Quantiles			
100%	Max	3922.0000	99.0% 3922.0000
75%	Q3	687.0000	97.5% 1016.0000
50%	Med	531.0000	95.0% 1015.0000
25%	Q1	453.0000	90.0% 1000.0000
0%	Min	406.0000	10.0% 422.0000
	Range	3516.0000	5.0% 422.0000
	Q3-Q1	234.0000	2.5% 407.0000
	Mode	453.0000	1.0% 406.0000

Subject D PDT reaction times for the "mixed" test – buttons when possible

Appendix L: Summary Data for Individual Test Runs

Table 16: Subject A - Summary Data for Individual Test Runs

	Test Run						
	Control	Selection	Selection w/ Buttons	Question	Question w/ Buttons	Mixed	Mixed w/ Buttons
Lane Violations	0	3	1	0	5	3	2
Speed Violations	0	6	4	2	2	0	6
PDT Observations	66	64	64	58	61	60	66
Missed PDT Targets	0	7	0	1	0	2	5
False PDT Trigger	0	3	0	1	3	0	2
Mean PDT Reaction Time (ms)	632	1478	931	1076	1034	1117	1220
Standard Deviation of Reaction Times (ms)	254	1097	625	747	564	852	951
Median PDT Reaction Time (ms)	**593**	**953**	**711**	**765**	**828**	**750**	**859**
Difference from Control Median (ms)		**+360.0**	**+118.0**	**+172.0**	**+235.0**	**+157.0**	**+266.0**

Table 17: Subject B - Summary Data for Individual Test Runs

	Test Run						
	Control	Selection	Selection w/ Buttons	Question	Question w/ Buttons	Mixed	Mixed w/ Buttons
Lane Violations	1	2	0	0	3	2	0
Speed Violations	0	0	1	1	1	1	1
PDT Observations	62	59	57	61	57	56	58
Missed PDT Targets	0	0	0	0	0	0	0
False PDT Trigger	0	1	1	0	0	0	0
Mean PDT Reaction Time (ms)	539	768	834	798	824	668	813
Standard Deviation of Reaction Times (ms)	119	398	664	352	494	196	539
Median PDT Reaction Time (ms)	**500**	**594**	**610**	**688**	**687**	**594**	**657**
Difference from Control Median (ms)		**+94.0**	**+110.0**	**+188.0**	**+187.0**	**+94.0**	**+157.0**

Table 18: Subject C - Summary Data for Individual Test Runs

	Test Run						
	Control	Selection	Selection w/ Buttons	Question	Question w/ Buttons	Mixed	Mixed w/ Buttons
Lane Violations	3	2	5	2	2	2	4
Speed Violations	0	3	1	2	2	2	1
PDT Observations	57	64	59	60	57	63	55
Missed PDT Targets	0	0	0	0	1	1	0
False PDT Trigger	0	1	0	1	2	1	0
Mean PDT Reaction Time (ms)	611	726	772	673	768	763	783
Standard Deviation of Reaction Times (ms)	512	445	588	256	606	647	498
Median PDT Reaction Time (ms)	**485**	**594**	**625**	**594**	**563**	**563**	**593**
Difference from Control Median (ms)		**+109.0**	**+140.0**	**+109.0**	**+78.0**	**+78.0**	**+108.0**

Table 19: Subject D - Summary Data for Individual Test Runs

	Test Run						
	Control	Selection	Selection w/ Buttons	Question	Question w/ Buttons	Mixed	Mixed w/ Buttons
Lane Violations	2	3	4	4	4	3	2
Speed Violations	1	1	1	1	0	1	1
PDT Observations	65	57	58	63	62	59	58
Missed PDT Targets	0	0	0	0	0	0	0
False PDT Trigger	0	3	0	0	0	0	0
Mean PDT Reaction Time (ms)	481	663	607	598	555	584	652
Standard Deviation of Reaction Times (ms)	95	403	417	234	212	454	473
Median PDT Reaction Time (ms)	**453**	**516**	**476.5**	**516**	**484**	**500**	**531**
Difference from Control Median (ms)		**+308.0**	**+322.0**	**+139.0**	**+117.0**	**+359.0**	**+378.0**