5-2018

Verification and Validation of Forces from Hippotherapy Rein Simulator

Sonia Sanchez

Verification and Validation of Forces from Hippotherapy Rein Simulator

A Thesis

Submitted to the Faculty

Of

Rose-Hulman Institute of Technology

By

Sonia Sanchez

In Partial Fulfillment of the Requirements for the Degree

of

Master of Science in Biomedical Engineering

May 2018

ROSE-HULMAN INSTITUTE OF TECHNOLOGY

Final Examination Report

Sonia Sanchez	Biomedical Engineering
Name	Graduate Major

Thesis Title Verification and Validation of Forces from Hippotherapy Rein Simulator

DATE OF EXAM: April 20, 2018

EXAMINATION COMMITTEE:

	Thesis Advisory Committee	Department
Thesis Advisor:	Renee Rogge	BBE
	Alan Chiu	BBE
	Eric Reyes	MA

PASSED X **FAILED** ______

ABSTRACT

Sonia Sanchez

M.S.B.E.

Rose-Hulman Institute of Technology

May 2018

Verification and validation of forces from hippotherapy rein simulator

Thesis Advisor: Dr. Renee Rogge

The thesis describes the redevelopment, testing, and validation of a rein training device intended to help riders learn to guide a horse during hippotherapy sessions. The thesis aims to validate and verify that the forces read by the redeveloped rein simulator were operant within ± 150 grams of the actual weight applied, as well as to investigate the effect of temperature on the system. The simulator focuses on the use of the English style of riding that uses direct force from the reins to guide a horse to turn or stop.

Data was collected by hanging weights from the load cells in the direction a rider would pull on the reins. The error of the system was found to range between ±30 grams for forces below 750 grams, ±50 grams for forces between 750 and 2500 grams, and as high as ±130 grams for forces greater than 2500 grams; therefore, the system operated within the specified ranges. Room temperature conditions were found to cause greater variability in the error of the system as opposed to higher or lower temperature conditions, but the errors were contained within tolerance thus verifying the device was effective for its intended therapeutic application.

DEDICATION

Para mis padres, quienes me han ensenado como trabajar duro y quienes han sacrificado todo para asegurase que alcance todos mis sueños

For my parents, who have taught me the value of hard work and have sacrificed so much to ensure that I achieve my dreams.

ACKNOWLEDGEMENTS

I extend sincere gratitude to the following parties:

Dr. Renee Rogge, for her patience and understanding as well as her expertise and push for this thesis and its previous designs

Dr. Alan Chiu, for bringing his expertise to my thesis committee

Dr. Eric Reyes, for his constant advice and knowledge

Christy Menke and the rest of the Lakeland Center staff, for presenting the original project in 2015 and for their excitement and appreciation. It has been an honor to work with your therapeutic center throughout these years

Ray Bland and Tom Rogge, for putting up with me and for providing endless advice and support

Lisa Knott, Glen Livesay, William Weiner, and the rest of the Biomedical Engineering department, for all of the laughs and words of encouragement

To my friends, especially Christine Rollins and Angelica Rodriguez, for always being a source of light in the darkest of days and for always encouraging me to push forward

Thank you.

Table of Contents

List of Figures

List of Tables

List of Equations

List of Abbreviations

BT: Bluetooth

Glossary

Hippotherapy: Use of horses for therapeutic rehabilitation

User: Person in charge hippotherapy session

Rider: Person receiving therapeutic services for which data is being collected

Load: Force rider uses to pull on the simulators horse reins

RStudio: Open source software capable of big data distributions, statistical analysis, data visualization, and processing

1. INTRODUCTION

The purpose of this thesis was to validate and verify that the forces read by a developed horse rein simulator were accurate within a certain tolerance range and that the system was reliable.

The document details the construction and development of the hippotherapy simulation device. This device prepares horse riders in equine therapy to instruct a horse to stop and/or turn in an English style of riding, where the forces exerted to the horse's mouth using horse reins would guide the horse to turn left, turn right, or stop. The device utilizes a mobile application and online database to store recorded sensor data. A feedback mechanism was included in the device design to notify the rider when he or she was exerting enough force onto the horse reins to direct a command. The data obtained can then be used for further analysis of its rider's progress.

2. BACKGROUND

2.1 Hippotherapy:

2.1.1History of Hippotherapy

Hippotherapy is defined as the "the use of horseback riding as a therapeutic or rehabilitative treatment, especially as a means of improving coordination, balance, and strength." It originates from the Greek word, "hippos" meaning horse and its literal definition is "treatment with the help of the horse." [1]

Since the time of the ancient Greeks, horses have been used as a form of therapeutic aid for patients with incurable illnesses. The earliest recorded use of horses for therapeutic purposes can be dated back to Hippocrates and in 17th century literature where it was often prescribed as a way to benefit patients with gout, neurological disorders, and low morale. [2]

Hippotherapy methods have been improved throughout the past 30 years and have been used by a wide variety of therapists and pathologists to increase flexibility, balance, posture, mobility, and muscular strength in a wide variety of patients with disabilities [3], [4]. The improvement of patient conditions is thought to be linked to the warmth and shape of the horse; as well as the horse's three-dimensional movement, which forces riders to change their posture to maintain their balance on the horse [3]. Modern forms of hippotherapy documentation began in the 1960's when researchers began to document the effectiveness of hippotherapy as a rehabilitative therapy option for patients with disabilities and physical health problems but due to the lack of common terminology and standardized testing, most of the reviewed studies concerning hippotherapy date up to 2007 [2]. The lack of high-quality and consistent studies has also raised questions about the true effectiveness of the therapy and whether these therapies should replace other therapy options with more concrete evidence-based reviews [2].

2.1.2 Horses That Qualify for Use in Hippotherapy

Horses used for hippotherapy practices are typically trained for therapeutic use before being placed in a rehabilitative program with patients [3]. Trainers select horses with calm and gentle temperaments without much concern for the breed, age, or size of the horse. The main requirement for a horse used in hippotherapy sessions is a healthy gait and the ability to move freely [4]. Due to the low requirements for hippotherapy selection, horses chosen for the therapeutic practice tend to be older in age. [3]

2.2 The Facility:

This thesis focuses on meeting the needs of the Lakeland Center for Therapeutic Riding Research in Coatesville, IN. As mentioned before, the use of hippotherapy for patients with disabilities lacks substantial evidence due to the lack of method standardization and terminology. The Lakeland Center strives to produce substantial evidence of the benefits of the therapeutic practice through the establishment of a standard for testing patient progress throughout their time with the facility [5].

Lakeland Center for Therapeutic Riding Research was established in 2000 as the Hope Haven Horse Farm, Inc in Coatesville, IN [5]. The nonprofit facility focuses on the identification of "innovative and unique approaches to the field of equine assisted therapies."

In 2012, Hope Haven began to collaborate with Rose-Hulman Institute of Technology and their Biomedical Engineering senior design capstone students to design and implement advanced technology that would help provide hard data proving the benefits of hippotherapy. This collaboration and implementation of new technology sparked a change in the facilities mission and brand. On October 28, 2017, Hope Haven Horse Farm was officially rebranded as the Lakeland Center for therapeutic riding research [5].

Although the center has made large strides in the field of equine therapy, one of the most challenging problems the facility encounters is providing its clients with a safe and effective way of learning proper rein technique. Proper rein technique is important because it helps prevent injuries to the horse and rider by decreasing sudden turns, jerks, and stops. This challenge is even more prevalent in beginner riders and riders with disabilities; therefore, the center looked to the Rose-Hulman Institute of Technology team for a viable solution to their problem.

2.3 Horse Riding Styles:

The Lakeland Center for therapeutic riding research uses two riding styles with two separate rein techniques throughout their therapeutic programs: English style and “Two-Handed” Western style. The chosen rein technique for a particular client relies heavily on the client’s needs and capabilities, but the Western style is more commonly used with beginner riders and riders with disabilities.

2.3.1 English Style Riding

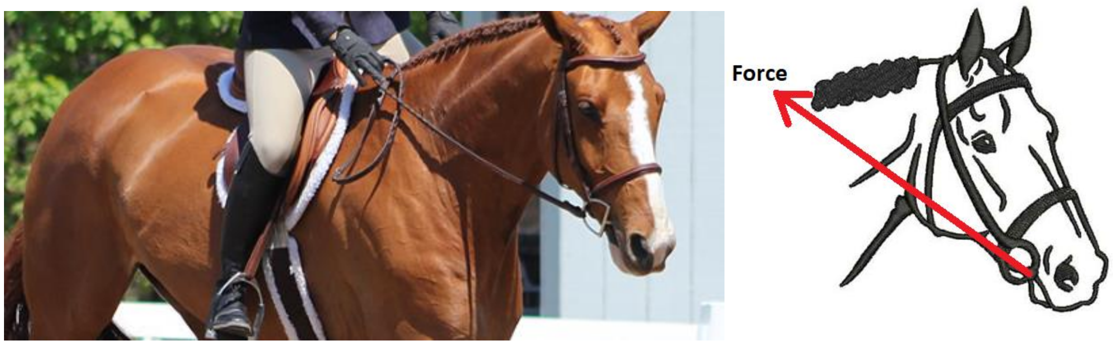


Figure 1: *English style riding and rein force direction [6]*

The English style of riding relies on plough or direct reining. This style depends on the rider’s consistent contact with the horse's mouth using the reins. In this style, a rein is held in each hand and is a held loosely by the sides of the horse, shown in **Figure 1**. In order to stop, the rider must gradually pull back both reins at about the same force and rate. When turning right or left, the rider should pull on the appropriate horse rein in the direction of travel while exerting little or no force onto the other.

2.3.2 "Two-Handed" Western Style Riding

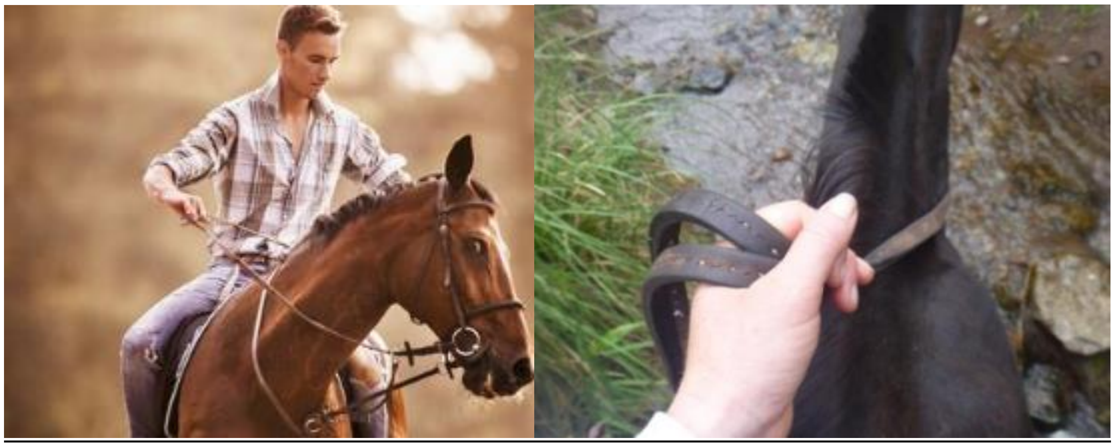


Figure 2: *Western style riding and rein positioning [4], [6]*

The Western style of riding relies on neck reining, meaning that the reins should be in contact with the horse's neck to turn left or right [6]. This method requires the user to clasp both horse reins in one hand and to lay the reins across the neck of the horse on the side that the rider would like to turn, as seen in **Figure 2** [6]; therefore, if the rider wished to turn right, they would lay the reins along the right side of the horses neck. In order to stop the horse, the rider pulls back on the reins.

In the "two-handed" Western style, the reins are held in the same manner, but the rider's free hand is wrapped on top of the other. This method is typically used to deter riders from grasping onto the horn of the horse saddle, therefore ensuring that the rider maintains their posture when riding [5].

3. LITERATURE REVIEWS AND PREVIOUS WORKS

3.1 Previous Works:

While a number of simulators are available for mechanically mimicking aspects of horse riding in equine therapy, such as a horse's gait, simulators focusing on obtaining the forces placed on a horses reins are nonexistent. In response to this opportunity, the Rose-Hulman Institute of Technology team has developed a series of horse rein simulators in an attempt to accommodate the needs of the Lakeland Center.

3.1.1 Prototype 1: "Mr. Giraffe"

The first simulator prototype was delivered for use to the Lakeland Center, then known as the Hope Haven Horse Farm, in May 2015 and was dubbed the name "Mr. Giraffe" by the designers. The device focused on the use of a "two-handed" Western style of riding [7]. The device alerted the rider when a command was successful with the use of lights and audio.

A schematic of the "Mr. Giraffe" device is shown in **Figure 3**. The device utilized flexible force sensors to gauge the amount of force exerted on the horse reins and photocells to identify if the reins were laid across the neck of the horse for simulated directional changes. The flexible sensors relied on a compressive force between a stationary plate and a plate attached to the reins to record the force exerted by the user [7]. The device successfully collected data and stored the information on an SD card that could be extracted from the Arduino used and uploaded to a computer for later analysis.

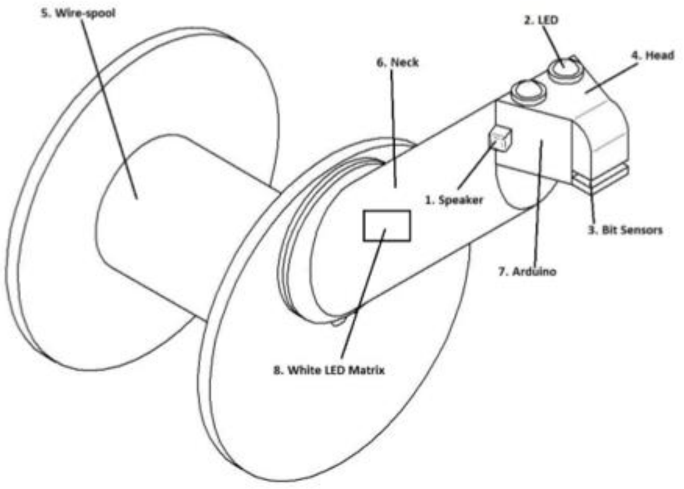


Figure 3:*Schematic of the Mr. Giraffe design. The structural subsystem consisted of a head (4), wire-spool (5), and neck (6). The neck was bolted to the body, as shown above, and acted as a support for the head. The wire-spool was where the user sits while using the device and also balanced the head and neck so that the device would not tip over. The head housed the electronic components of the other subsystems, including the Arduino (7), the LEDs (2), and the speakers (1). The sensing components were located at the mouth (3) and on the neck (8). The photocells used for neck reining were attached to the reins, which were not shown [7]*

The simulator was successful in simulating rein techniques but, after about 4 months of use, major problems with the device's structure were apparent. Due to constant use, the head and integrity of the neck began to diminish. The head of the device housed the electrical components of the feedback mechanism along with force sensors, so its integrity was vital to the survival of the device. **Figure 4** shows the design of the head and final construction with laser cut acrylic. The material used began to crack and break with use, and in turn, the circuitry components were also compromised resulting in the destruction of the system.

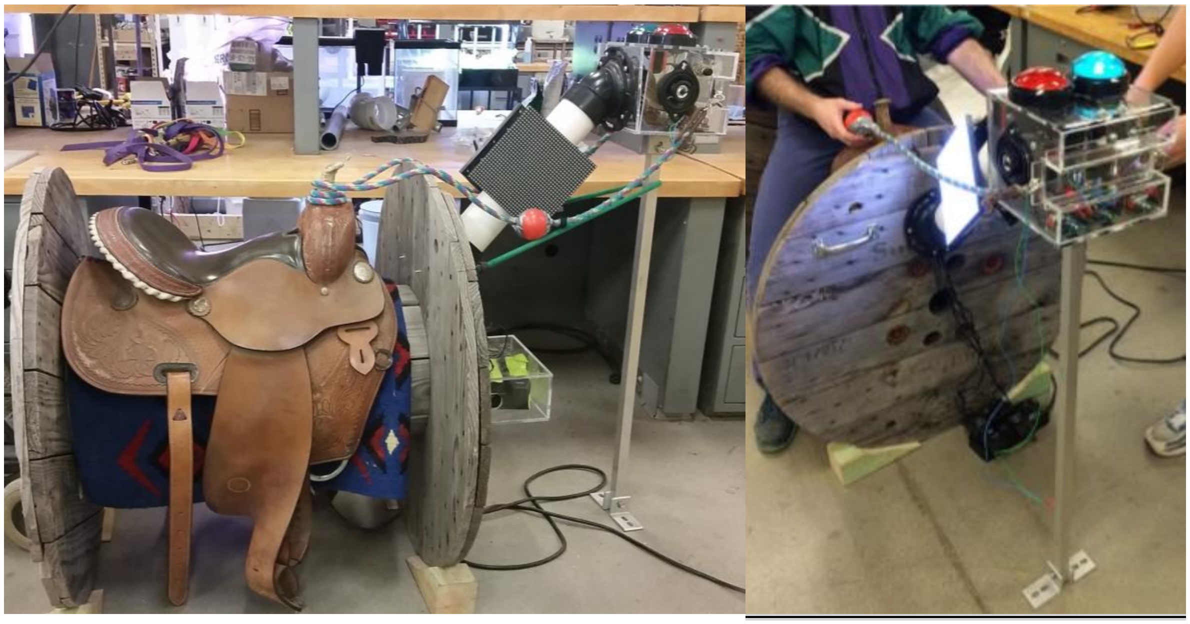


Figure 4:*Mr. Giraffe final presentation [7]*

3.1.2 Prototype 2: "Johnny 5"

The next iteration of the "Mr.Giraffe" simulator was the "Johnny 5" simulator prototype. This device was created in response to the malfunctions and limitations of the "Mr. Giraffe." It accommodated for both English and Western styles of riding and used round pressure sensors to detect and measure the amount of force exerted on the horse reins [8]. The force sensors were mounted between two acrylic plates on the mouth of the horse head and relied on the compressive force between the plates to record force values. Capacitance plates attached to the side of the horse head and conductive cloth located on the reins were used to detect if the reins were in contact with the horse's neck when the device was set to Western style.

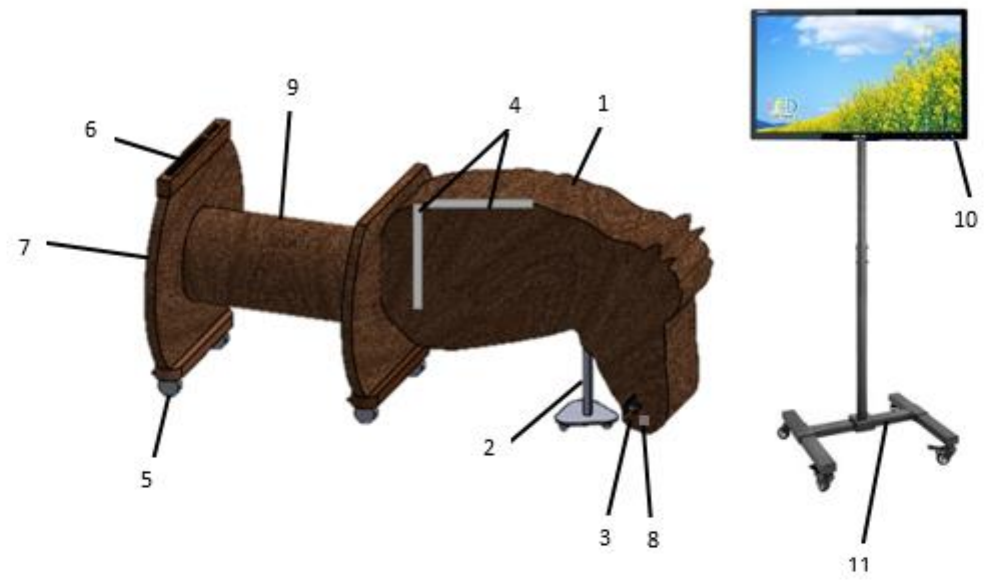


Figure 5:*Schematic of the "Johnny 5" simulator. The structural subsystems consisted of (1) a wooden head with inner layers of green foam support and an aluminum "L" support frame, (2) a tri-wheel stand to support the head for both English and Western positions, (3) bit assemblies with force sensors, (4) capacitance plates for evaluating how well rein movements match commands, (5) four locking, swivel wheels for easy maneuver, (6) a control panel with 10 buttons for commands and three potentiometers for setting difficulty level, force level, and time setting, (7) an internal sliding board which mounts the electronics, (8) a sensor at the bit for detecting when the rider's hands are being held too high, (9) the main spool support, and lastly (10) the tv monitor display and (11) rolling stand. [8]*

The device retained the wooden spool body previously used in the "Mr. Giraffe" design, but the head was completely redesigned. The PVC and acrylic box were replaced with a wooden and foam head, shown in **Figure 6**.

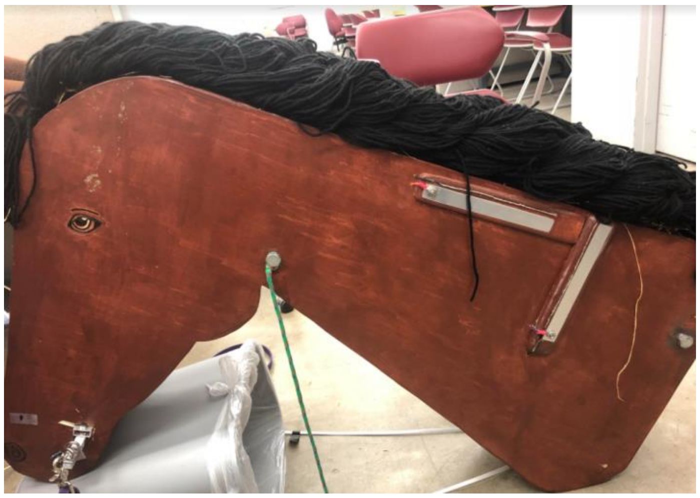


Figure 6: *Horse head created for the Johnny 5 design*

An “L” shaped beam was constructed and implemented into the head of the device to increase the structural stability of the head whereas the electronics were relocated to the middle of the wooden spool body. A switch controlled the testing style for the user, as seen in **Figure 7**, while a series of switches and knobs were used to record completed and failed technique trials as well as to increase or decrease the difficulty of the simulation, as seen in **Figure 8**. A monitor would provide the rider with images if a command was successfully done.

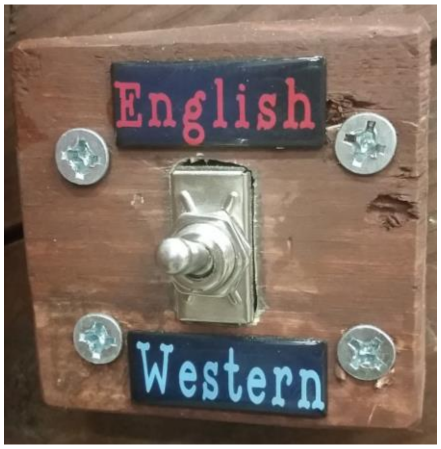


Figure 7: *English/Western Switch which altered the software settings from English to Western mode by altering how the "hands too high" setting is reached and how the turning is accurately detected. [8]*

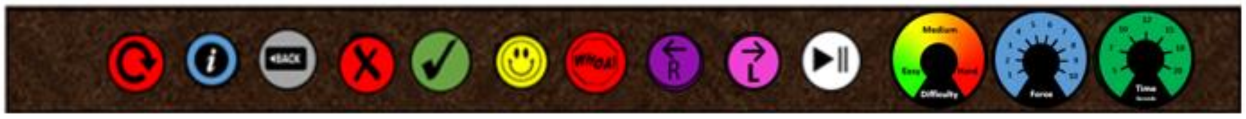


Figure 8: *The control panel with 10 buttons and 3 potentiometer knobs. [8]*

"Johnny 5" was delivered to Lakeland Center in June 2016 but soon began to malfunction. The controls did not power the device, and the monitor did not display images of correct or incorrect commands. The acquisition of data was also very unreliable since it required the trainers to press buttons when they believed a command was done correctly by the rider and had to be logged by hand. Due to these malfunctions, a new iteration of the device was necessary.

4. OVERVIEW OF DESIGN FUNCTION

The following section provides an overview of the current device design, followed by detailed descriptions of the device's primary components. These sections provide information on the improvements made on the previous device prototype called "Johnny 5."

The redesign goals centered around the incorporation of new sensors with verified forces, the development of a user-friendly control mechanism, real-time data acquisition and storage, a feedback mechanism, and the ability to obtain the stored data for later analysis.

The simulation device focuses on recording and processing the forces a rider would apply on horse reins when riding in the English style. The simulator relies on the use of an Android application and force sensors mounted at the mouth of the horse head to successfully collect data and provide the user with feedback regarding the rider's ability to properly instruct a change of direction or stop command during a hippotherapy session.

5. DESIGN

5.1 Overview of Current Device Design:

The following sections describe the design changes made to the previously existing "Johnny 5" rein simulator. These changes were made to create a more effective and user-friendly rein simulation device focused on the English style of riding for a hippotherapy research facility.

5.1.1 Electrical Components

As stated earlier in the document, the previous "Johnny 5" prototype used round pressure sensors, pressed between two acrylic plates, to read the forces the rider applied to the horse reins. The sensors relied on a compressive force, and the structure ultimately failed due to the cracking of the acrylic plates because of constant use and handling when transporting from location to location by the facility. After attempting to recreate the system used with the "Johnny 5" prototype and finding that the system did not produce constant and reliable forces, it was apparent that a new set sensors and a new sensor mounting system needed to be designed so that device could withstand the forces exerted on the reins during future sessions.

The solution to the problems mentioned above required the use of Sparkfun's 10k straight bar load cell, shown in **Figure 9**. The straight bar load cell replaces the round pressure sensors and acrylic plate setup. The load cell is made of an aluminum alloy and takes a measurement of the electrical resistance of the bar in proportion to the strain applied to it [9]. The output force of the load cell produces either positive or negative force, depending on the direction in which the force is applied. Therefore, if the direction of the force is applied in the direction shown by the arrow in **Figure 9**, the force would be read as a positive force. If the direction of force was opposite of that shown in the figure, the force would then be read as negative. The cell is also able to measure up to

10kg (~22lbs) of force [9], which is well within the limits of a typical force needed to lead a trained horse during therapeutic sessions.

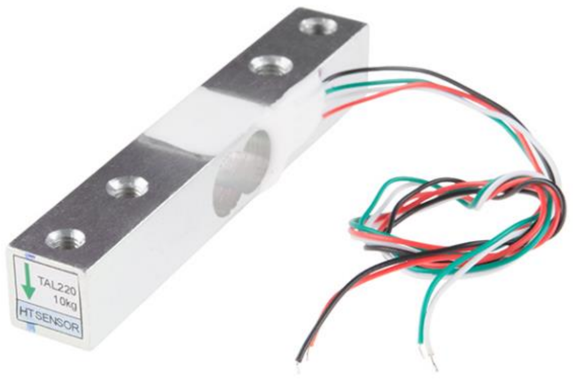


Figure 9: *Straight bar load cell* [9] *When a force was exerted in the direction of the arrow, the force was read as positive whereas if a force was exerted in the opposite direction, the force was read as a negative force.*

Two load cells were mounted on the device, one on each side of the horse head. Each load cell contained 4 strain gages, hook up in a Wheatstone bridge formation, as seen in **Appendix A**.

Sparkfun's HX711 load cell amplifier was added to the design to amplify the electrical signal from the load cells. The amplification of the signal allowed for the signal to more easily detected and processed through the Arduino UNO microcontroller, as shown in **Figure 10**. An HC-06 Bluetooth module was implemented to allow communication between the Arduino UNO and an Android application that was created to be coupled with the device.

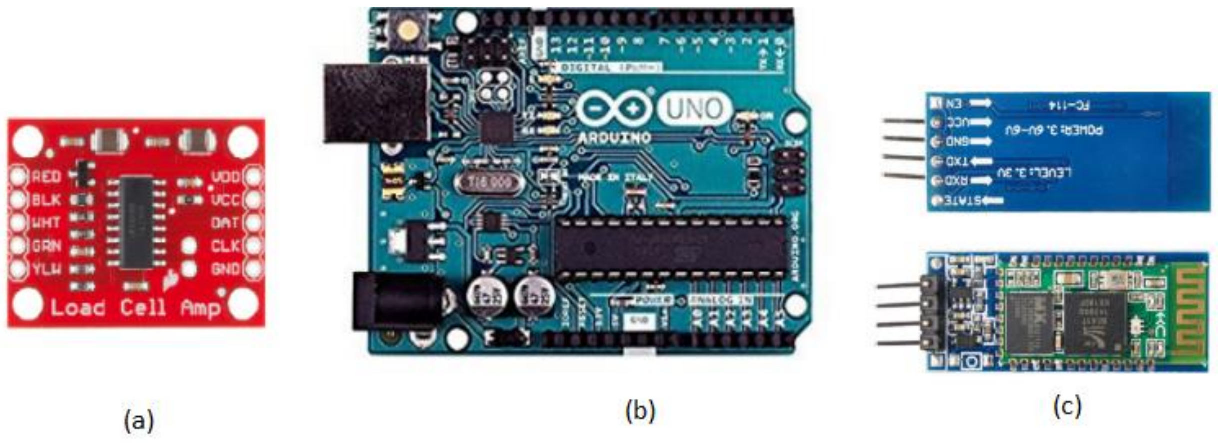


Figure 10: *(a) HX711 load cell amplifier, (b) Arduino UNO, (c) HC-06 Bluetooth Module [9]*

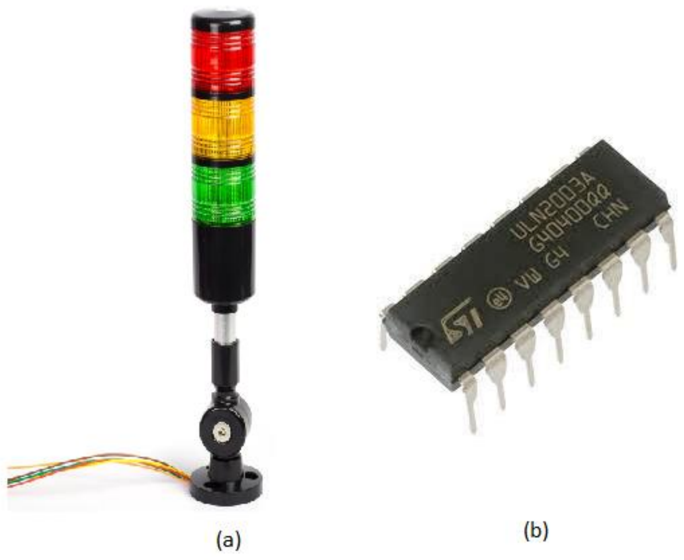


Figure 11: *(a) Adafruit Tower Light, (b) ULN2003 Darlington transistor [10]*

The feedback mechanism was created through the use two Adafruit tower lights, seen in **Figure 11 (a)**. The Adafruit tower light required a +12V power source and was equipped with three different lights: red, yellow, and green. It also had an embedded buzzer. The tower light had 5 different colored wires. The brown wire was connected to common +12V power line whereas the rest of the wires were grounded to activate or light their designated LED lights: the red wire for the red light, yellow wire for the yellow light, green wire for the green light, and the final orange wire for the buzzer [10]. The tower was coupled with a Darlington transistor in order to regulate the conditions and order by which the LED lights were lit. (**Appendix B**)

5.1.2 Mounting the Straight Bar Load Cell:

The mounting of the load cells required a modification to the previous structural support of the "Johnny 5" prototype. The previous model used a simple "L" shaped aluminum bar, much like the image shown in **Figure 12 (a)**, to support the device's structure. The bar was embedded into the layers of foam which formed the device's head and was positioned so that the device was entirely supported through the middle, shown in **Figure 12 (b)**. A hollow 3''X 1'' aluminum square bar was welded to each side of the "L" shaped square bar so that the load cells could be mounted in the manner suggested by the product page shown in **Figure 13**. The product page suggested the cells be mounted between two plates in a "Z" shape, shown in **Figure 13** so that the strain could be measured correctly.

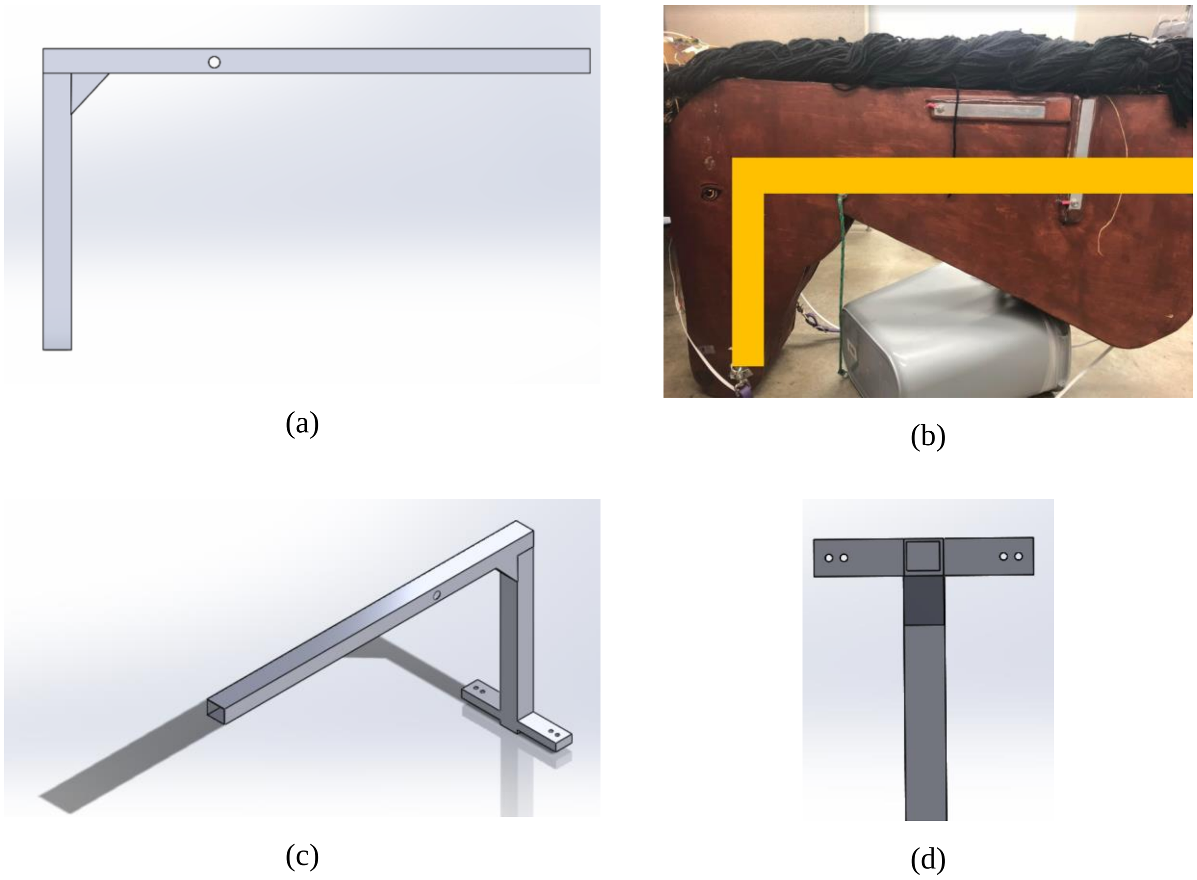


Figure 12: *Structural support figures: (a) original structural "L" shaped square bar from the "Johnny 5" design, (b) location of the structural support within the horse head, (c) drawing of the modified structural support, (d) bottom view of the modified "L" shape*

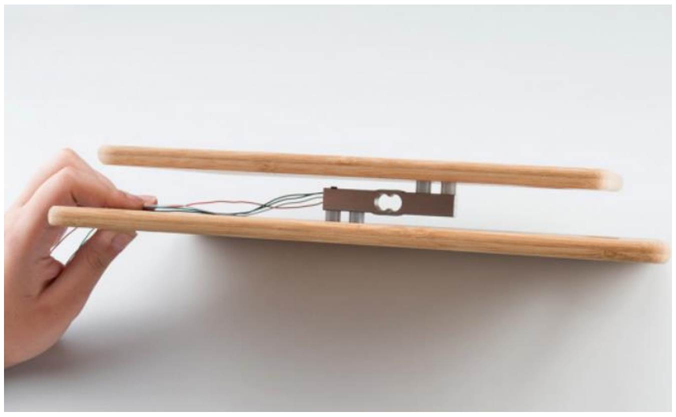
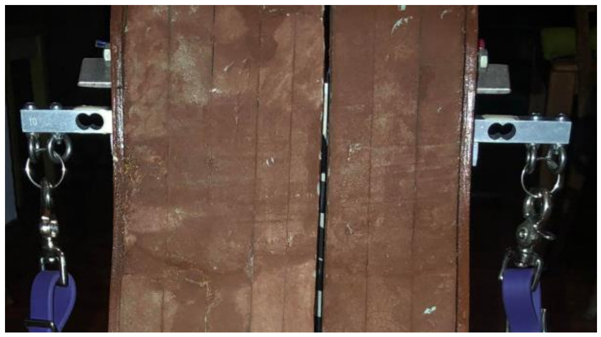


Figure 13: *Suggested setup for the load cell [9]*

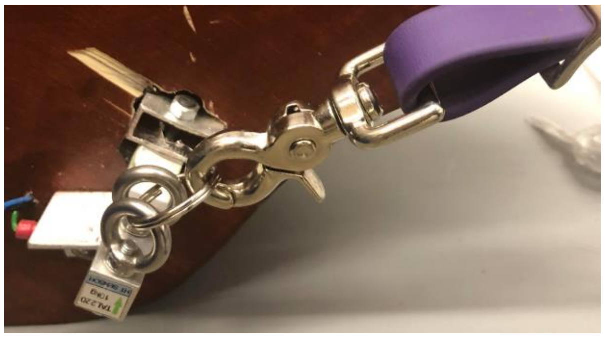
Two M5 sized holes were then drilled through the added bars in order to accommodate for the screws that would hold the load cells onto the structure as seen in **Figure 14 (c) and (d)**. The load cells were then screwed onto the structural bar and placed so that the direction of positive force would be the direction the user would pull the horse reins, as shown in **Figure 14 (c)**. Two M4 ring-shaped lifting eye nuts were then placed at the end of two M4 bolts which were bolted through the free end of the load cell. A key ring was then threaded through the two eye nuts in order to provide an attachment site for the reins, as seen in **Figure 14**. After the two load cells were mounted, the next task was to calibrate the two sensors.

HX711 amplifier.

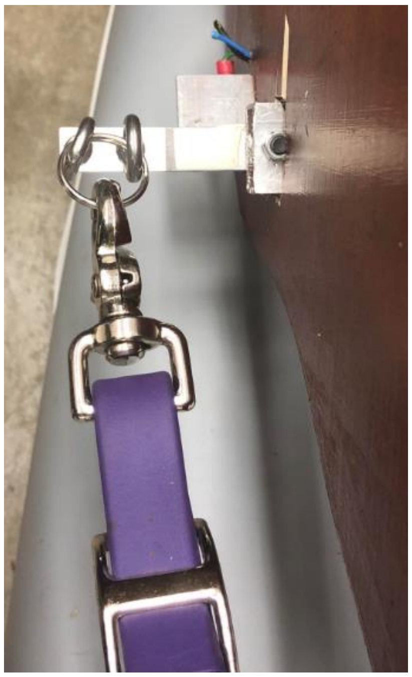
(a)



(b)



(c)



(d)

Figure 14: *(a) M4 ring-shaped lifting eye nuts, (b) bottom view of load cell placement, (c) Side view of the mounted load cell with attached reins, (d) Top view of the mounted load cell with attached reins*

6. METHODS USED TO MEET DESIGN GOALS

The following sections focus on the load cell calibrations and data acquisition, processing, and storage. Throughout the remainder of the document, the use of the phrase “load cell forces” or “load” will refer to the forces read from the straight bar load cell after the signal was amplified by the HX711 amplifier.

6.1 Sensor Calibration:

6.1.2 Calibrations without Reins:

The initial force measurements and load cell calibrations were taken without attaching the reins to the load cells. All data obtained for the force verification analysis was done without the incorporation of the horse reins to the system. The device was flipped upside down, much like seen in **Figure 14 (b).** This allowed known weights to be hung from the load cells in the direction of the positive force.

To calibrate the forces read from both force sensors. An Arduino program was written to read the initial forces being supplied by both sensors. Before the program could be written, the *Q2HX711.h* library had to be added to the Arduino software library. This was accomplished by going to *Sketch* > *Include Libraries* > *Manage Libraries* and searching for the *Q2HX711* library. Once the library was added the library had to be included in code by typing:

$$\#include < Q2HX711.h >$$

The cell was then defined bythe program using the following line of code:

$$Q2HX711\ cell\ (3,2);$$

The values of 3 and 2 were used to represent the pins on the Arduino UNO where the HX711 amplifier *DAT* and *CLK* outputs were plugged into. Once the initial values were successfully read, they were entered into **Equation 1** to obtain the most recent average.

Equation 1: $val = 0.5 * val + 0.5 * cell.read(\,);$

After obtaining the most recent average, the values were printed to the Arduino Serial Monitor using the *Serial.print* command. The program executed for a minute and then stopped to obtain a string of values generated using **Equation 1**. The string was then copied and pasted into an excel file, where an average of the values was obtained and named *averagedVal*.

Equation 2: $load = val - averagedVal$

Using **Equation 2** to read the incoming values and after placing a 100 g weight onto the load cells; the program was then run for a minute. After the minute was up, the array of values were once again transferred to excel and averaged. The averaged value was then called *n* and added to the **Equation 2** to form the final calibration equation shown in **Equation 3**.

Equation 3: $load = \frac{val - averagedVal}{(float)\, n} * 100$

The calibration equation was determined to be adequate for sensor testing after two trials of the force verification testing were completed and the observed forces closely match the weight applied to the system.

6.1.2 Calibrations with the Horse Reins

After the data was collected to verify that the forces of produced by the sensors were accurate and reproducible, the horse reins were added to the system in order to calibrate the sensors taking into account the weight of the reins. The device was turned upright, and forces were recorded based on the Lakeland Center's staff holding the reins in the starting position for English riding. The same steps used in the initial calibrations (6.1.1) were used to calibrate the system with the reins.

6.2 Device Data Acquisition:

The devices data acquisition system is made up of two main parts: the Arduino UNO and the Android mobile application.

6.2.1 Reading and Sending Forces: Arduino UNO

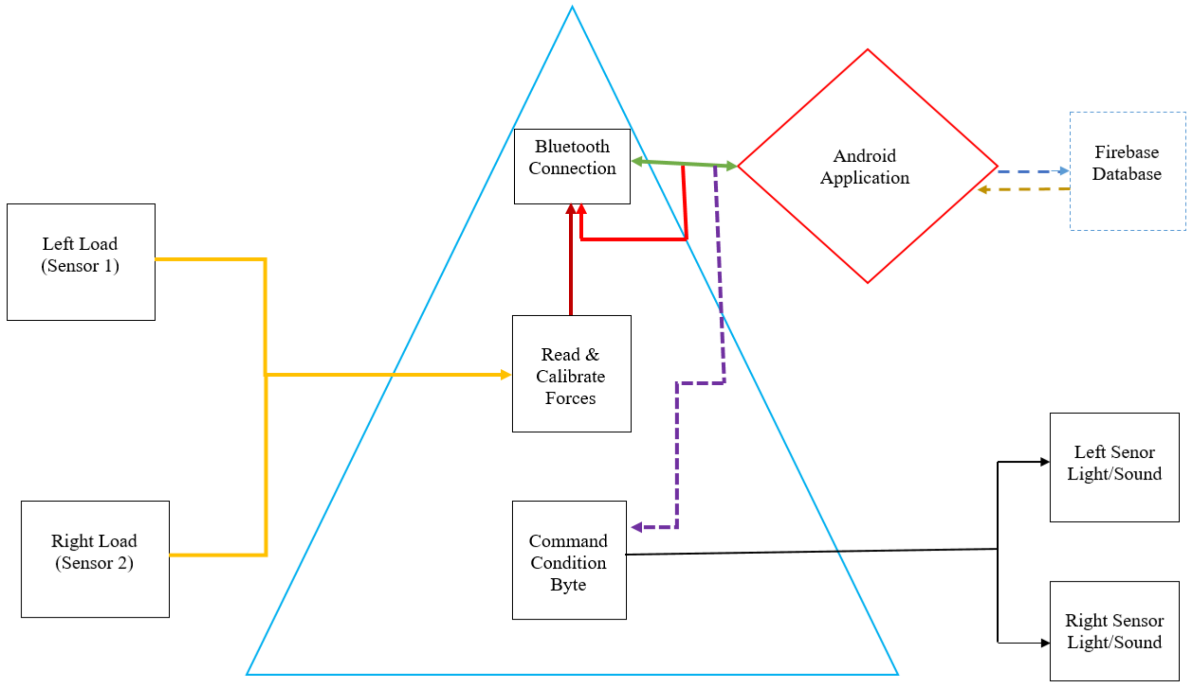


Figure 15: *Functions of the Arduino UNO, represented by the Δ, and its interactions with other external components of the devices data acquisition system, such as the load cells, feedback mechanism, and Android application.*

The Arduino UNO was used to calibrate the forces from the load cell, allow for Bluetooth communication, and to produce the output for the feedback mechanism, as shown in **Figure 15** where the triangle represents the Arduino UNO. The HC-06 Bluetooth module ran in a loop, constantly looking to establish Bluetooth communication with a device. If a device was not found, the module would once again begin its search until a connection was established. Once communication was established with the Android application, the Arduino UNO would then send the calibrated forces to the mobile device. The Android application would receive the forces and would send a byte to the Arduino UNO based on the forces received. The byte sent would then establish the state of the feedback mechanism, turning on particular lights when the forces read fell within a specific range. **Table 1** summarizes the byte received by the Arduino UNO and its designated output or feedback mechanism state.

Table 1: *Feedback mechanism output based on received byte to Arduino UNO from the mobile application. The forces from both Sensor 1 and Sensor 2 determine the byte that is sent from the mobile application to the Arduino UNO. If the force read falls within a certain range, then a particular light will turn on. A green light indicates that the rider is applying the correct amount of force to the reins whereas red indicates that too little or too much force is being applied. The yellow light indicates if the rider is about to approach the green or red light threshold. A trial was successful only when "A" is sent to the Arduino UN0, otherwise the trial was considered a failed run.*

Received Byte	**Sensor 1 (Left) Light Tower**	**Sensor 2 (Right) Light Tower**
"A"	Green	Green
"B"	Green	Yellow
"C"	Green	Red
"D"	Yellow	Green
"E"	Yellow	Yellow
"F"	Yellow	Red
"G"	Red	Green
"H"	Red	Yellow
"I"	Red	Red
"S"	-	-

6.2.2 Android Application: MIT App Inventor

An Android application was created using the MIT App Inventor 2 online program that utilized block programming to create mobile applications. The purpose of the mobile application was to provide an interface by which the Lakeland Center could input client information and run tests or sessions for a particular command: left, right, or whoa/stop. It was also used to collect and store force data obtained from the load cells. These tasks were accomplished by creating three main screens that would control the functions of the application. The flowchart shown in **Figure 16** provides an overview of the application functions and the flow of information between the mobile application, Arduino, and online database - Firebase.

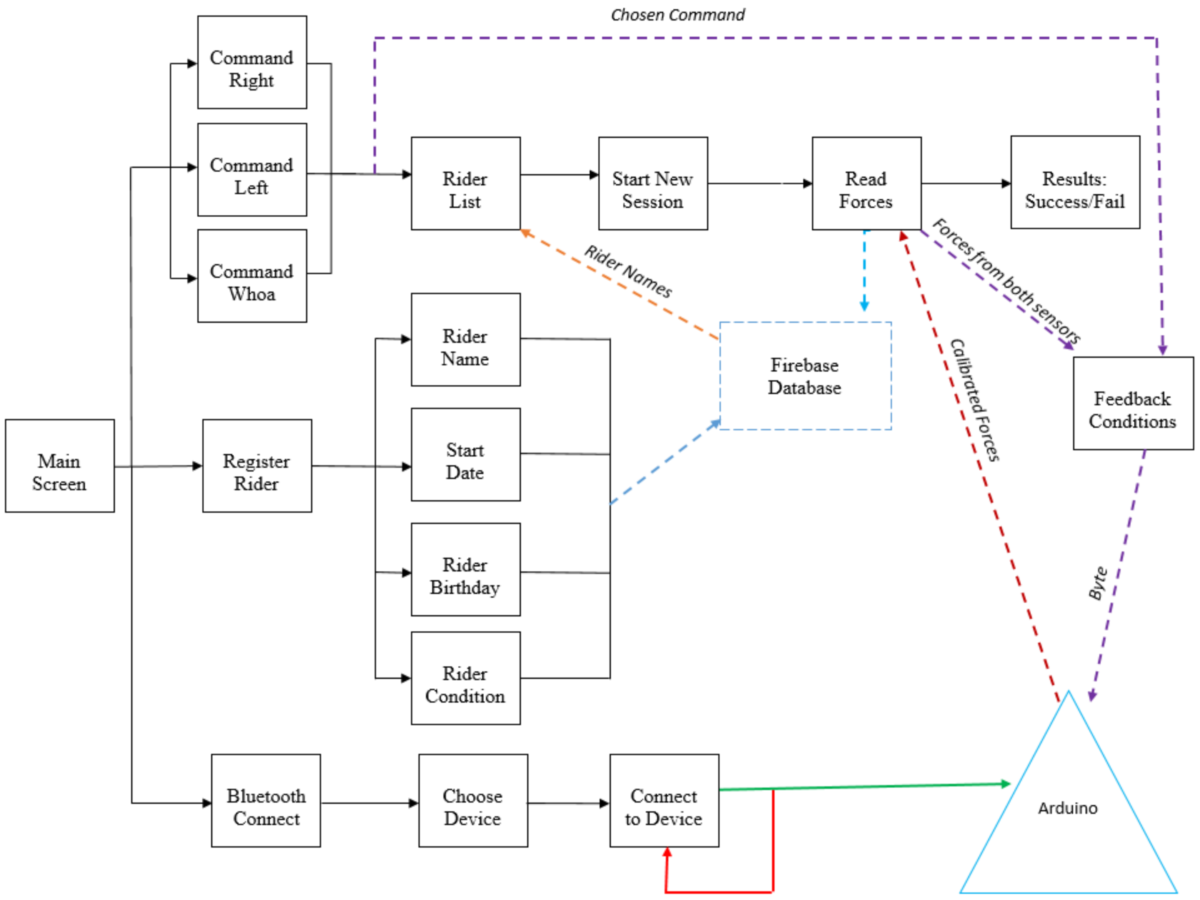


Figure 16: *Functions of the Android application and its interactions with the Google Firebase database and Arduino UNO.*

The application had three screens: the main screen, user registration screen, and test screen. The main screen, as seen in **Figure 17**, provided basic access to the rest of the application

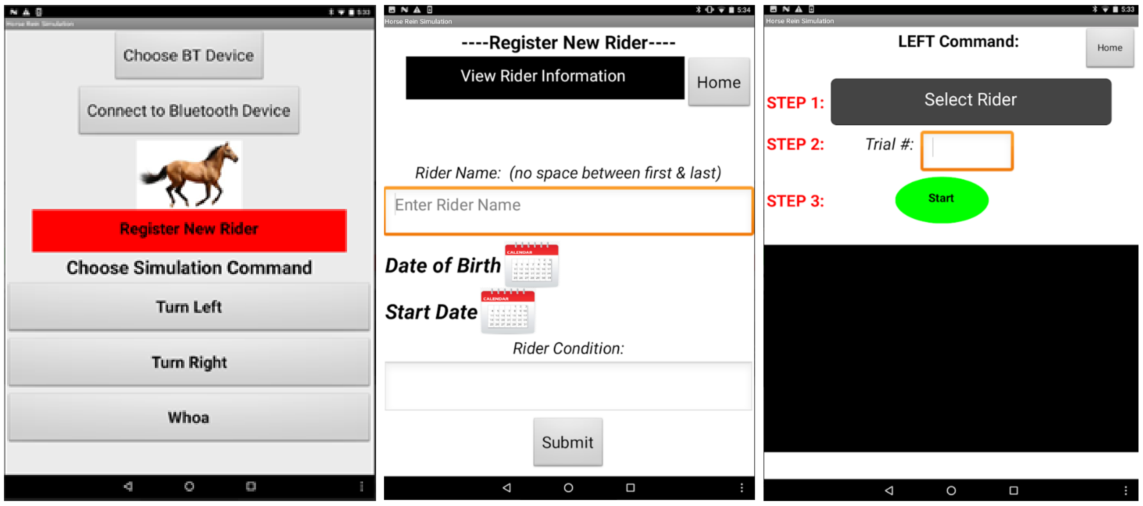
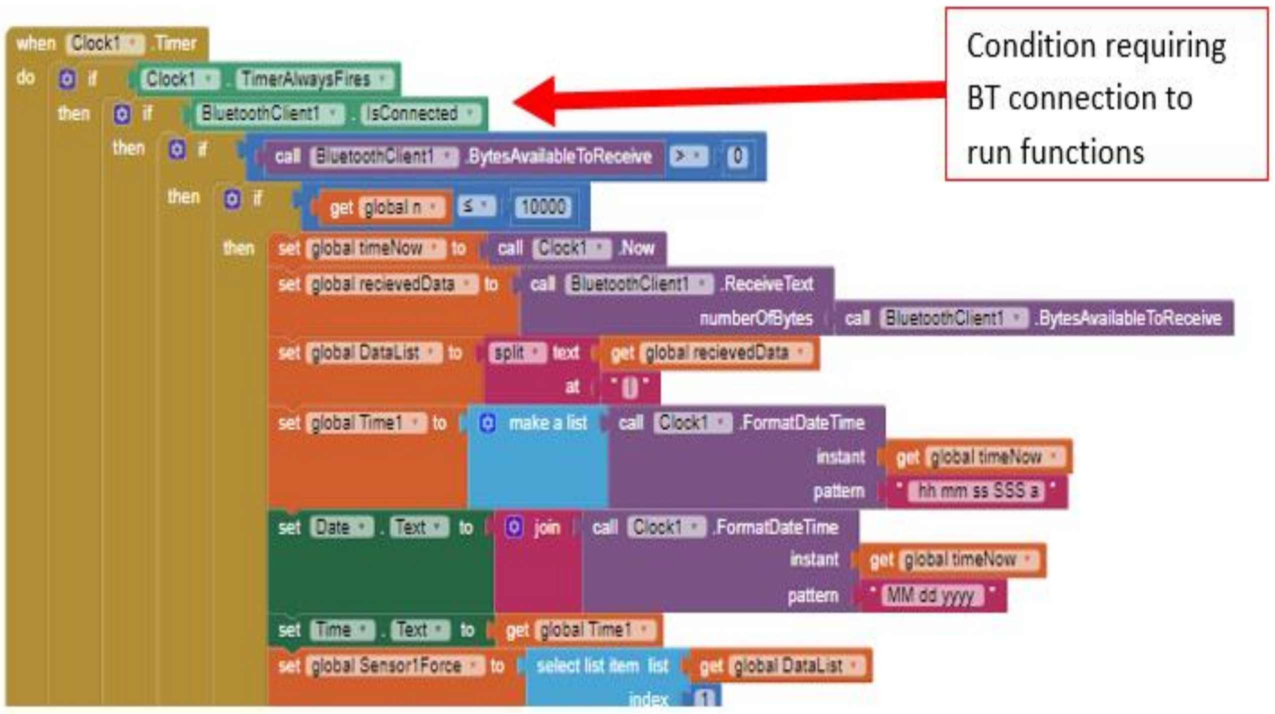


Figure 17: *Mobile application main screen, rider registration screen, and test screen*

The mobile application relied heavily on the establishment of Bluetooth communication. Without the Bluetooth communication, the application was rendered useless since most functions were programmed under the condition of its establishment, as seen in **Figure 18**. “Register New Rider” is the only function that was capable without a Bluetooth connection.

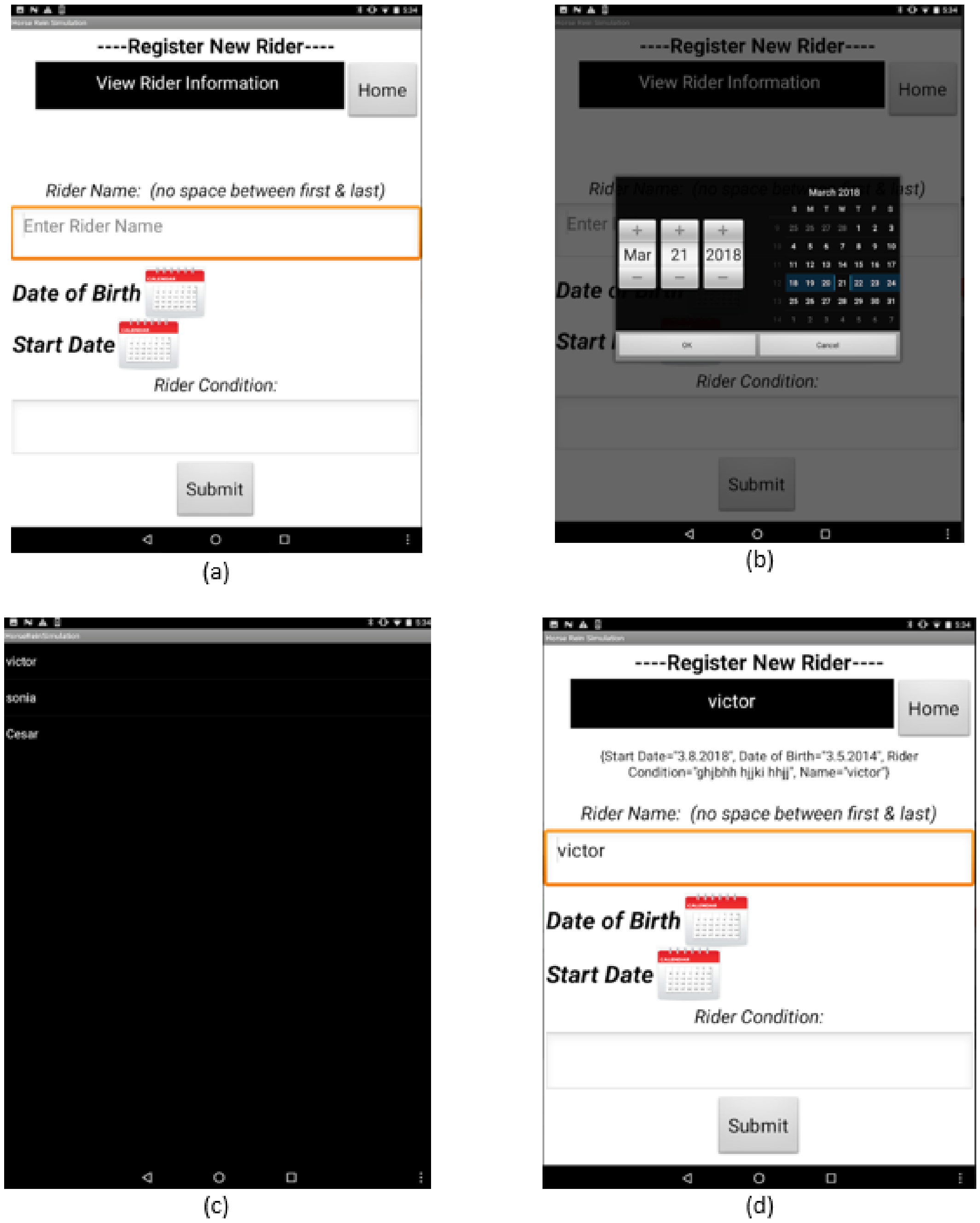


```
when Clock1 .Timer
do  if  Clock1 . TimerAlwaysFires
    then  if  BluetoothClient1 . IsConnected
          then  if  call BluetoothClient1 .BytesAvailableToReceive > 0
                then  if  get global n ≤ 10000
                      then  set global timeNow to  call Clock1 .Now
                            set global recievedData to  call BluetoothClient1 .ReceiveText
                                                                  numberOfBytes  call BluetoothClient1 .BytesAvailableToReceive
                            set global DataList to  split text  get global recievedData
                                                          at  " | "
                            set global Time1 to  make a list  call Clock1 .FormatDateTime
                                                                      instant  get global timeNow
                                                                      pattern  " hh mm ss SSS a "
                            set Date . Text to  join  call Clock1 .FormatDateTime
                                                              instant  get global timeNow
                                                              pattern  " MM dd yyyy "
                            set Time . Text to  get global Time1
                            set global Sensor1Force to  select list item  list  get global DataList
                                                                          index  1
```

Figure 18: *Partial image of the program block that required the use of the mobile devices internal clock (Appendix C).The clock was programmed to function as the main powerhouse for controlling data storage and feedback conditions. This partial image emphasizes the importance of the establishment of Bluetooth (BT) communication.*

6.2.2.1 Rider Registration

Before the simulator could be used to execute command trials, the rider had to be registered using the Android application. As shown in **Figure 19**, the rider registration form allowed for the input of the rider's name, birthday, condition for which they were receiving hippotherapy, as well as the first day they began to use the device. It was important to note that the rider's name could not have a space between the first and last name. After the submit button was pressed, the information was then pushed to the Google Firebase database and stored in a nested JSON structure, as seen in **Figure 20**.

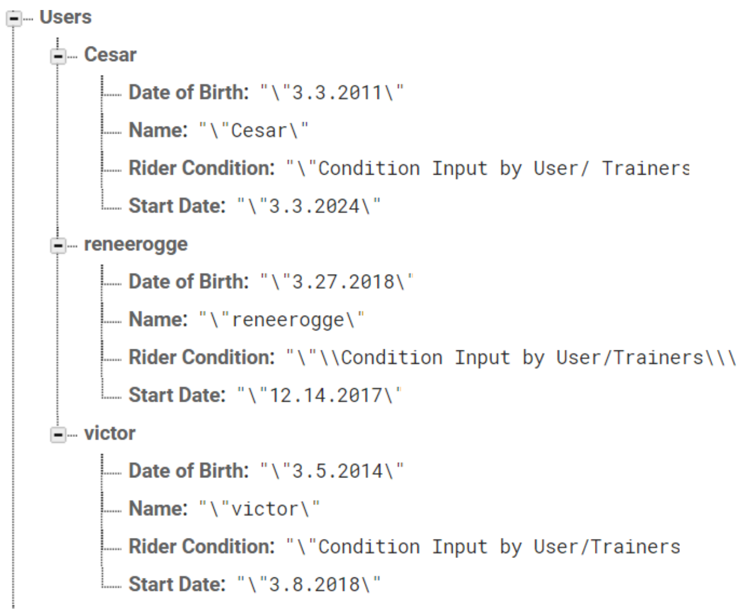


{Start Date="3.8.2018", Date of Birth="3.5.2014", Rider Condition="ghjbhh hjjki hhjj", Name="victor"}

Figure 19:*(a) Rider registration screen , (b) Date selector for birthday or start date input, (c) List of previously registered riders seen when View Rider Information was selected, (d) Screen after a rider was chosen from the list*

Rider information was edited either by changing the information directly in the database or by selecting the riders' name in the *View Rider Information* list. When a name was selected, the information for that particular rider was displayed, and their name was automatically placed in the name text box, as seen in **Figure 19 (d)**. The information that needed to be edited was placed in the appropriate location and once submitted it was updated in the Firebase database.

If the rider's name needed to be changed, it could not be changed using the *View Rider Information* because the application would recognize it as the registration of a new rider and therefore the data stored would no longer be added to the correct rider's profile. Therefore, name changes had to be done directly in the database and could not have a space between their first and last name as dictated by the application. Within the database, name changes had to be done in the first level under the title "Users" and also in the second level next to "Name:" for the designated rider.

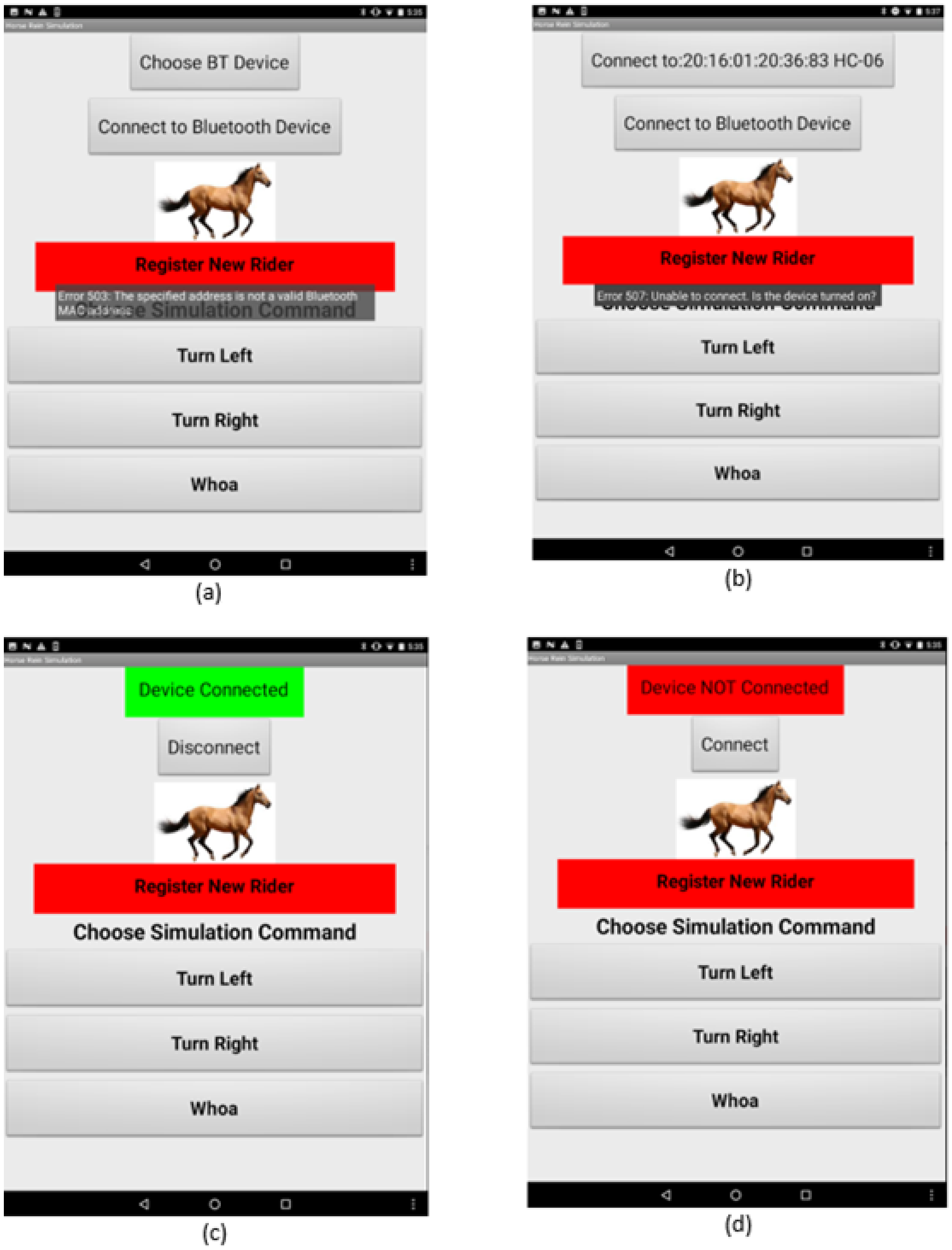


```
Users
  Cesar
    Date of Birth: "\"3.3.2011\"
    Name: "\"Cesar\"
    Rider Condition: "\"Condition Input by User/ Trainers
    Start Date: "\"3.3.2024\"
  reneerogge
    Date of Birth: "\"3.27.2018\'
    Name: "\"reneerogge\"
    Rider Condition: "\"\\Condition Input by User/Trainers\\\
    Start Date: "\"12.14.2017\'
  victor
    Date of Birth: "\"3.5.2014\"
    Name: "\"victor\"
    Rider Condition: "\"Condition Input by User/Trainers
    Start Date: "\"3.8.2018\"
```

Figure 20: *Rider registration information storage structure in Google Firebase*

6.2.2.2 Bluetooth Connection

Bluetooth communication was easily established through the main screen. The user had to find the device they wished to connect to, which was the HC-06 Bluetooth module previously mentioned, and then press the *Connect to Bluetooth Device* button. If a device was not selected or if the selected device was not available, then an error message would be displayed. These messages can be seen in **Figures 21 (a) and (b).**



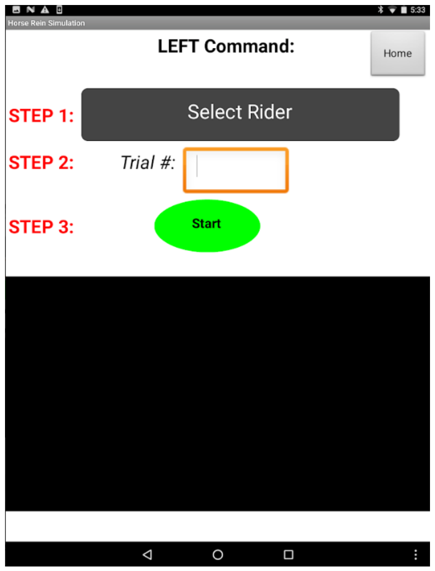
Error 503: The specified address is not a valid Bluetooth MAC address

Error 507: Unable to connect. Is the device turned on?

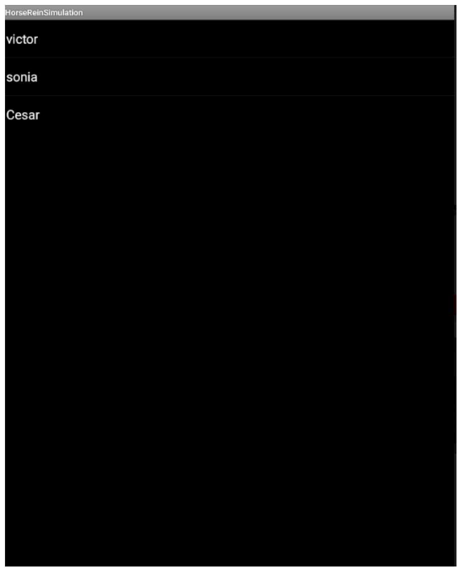
Figure 21*: (a) Error displayed when the device was not selected, (b) Error displayed when the device could not be found or not turned on, (c) Device connected, (d) Device disconnected*

Once the device had successfully connected to the HC-06 BT module, the screen would display "Device Connected" in a green box. If the connection was lost, then the box would turn red and display “Device Disconnected,” as seen in **Figure 21 (c) and (d)**. After Bluetooth communication was established, the user could begin to run their tests for each command: left, right, and whoa.

Testing/ Trial Runs



(a)

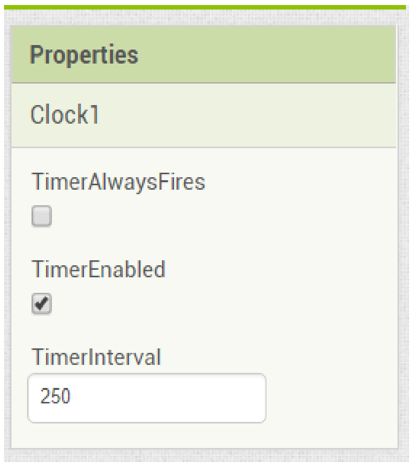


(b)

***Figure 22**: Test Screen. (a) Initial Screen, (b) Select Rider list screen*

The testing screen could be reached through the use of the simulation command buttons on the main screen. When a command button was clicked, the test screen would appear, and the chosen command would be displayed at the top of the screen beside the home button, as seen in **Figure 22 (a).** In order to properly document the forces that were read during the test, the user had to follow the steps displayed on the screen. They first had to select the rider name, then input the trial or run number, and once they thought the rider was ready, the user would press the “Start” button. When the user did not choose a rider name, the data was stored in an arbitrary location under the name of the command instead of being properly documented under the rider name.

Once the “Start” button was pressed, the application would begin to receive the forces sent from the Arduino through the Bluetooth connection previously established. Forces were read by the application at a frequency of 250 milliseconds per second (4 forces/ second), and each trial ran for 10 seconds. The rate at which the forces were read was established in the applications clock settings, as seen in **Figure 23**. As the forces were read, they were simultaneously being printed on the mobile screen, pushed to the online database, and being assessed by the feedback byte condition functions (**Appendix C & D**).


Properties

Clock1

TimerAlwaysFires ☐

TimerEnabled ✓

TimerInterval 250

Figure 23*:Setup for the clock properties in MIT App Inventor where "TimerInterval" represents the rate at which forces or information may be received or sent by the mobile application*

The data was stored and organized in the online database as JSON nested data, shown in **Figure 24**. The JSON organization of the data was done using the two blocks of code seen in **Figure 25**.

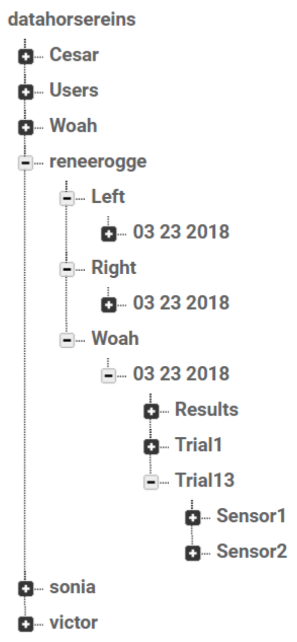

```
datahorsereins
├─ Cesar
├─ Users
├─ Woah
├─ reneerogge
│  ├─ Left
│  │  └─ 03 23 2018
│  ├─ Right
│  │  └─ 03 23 2018
│  └─ Woah
│     └─ 03 23 2018
│        ├─ Results
│        ├─ Trial1
│        └─ Trial13
│           ├─ Sensor1
│           └─ Sensor2
├─ sonia
└─ victor
```

Figure 24: *Nested data saved in Google Firebase*

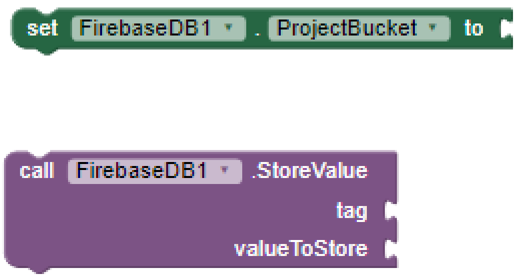
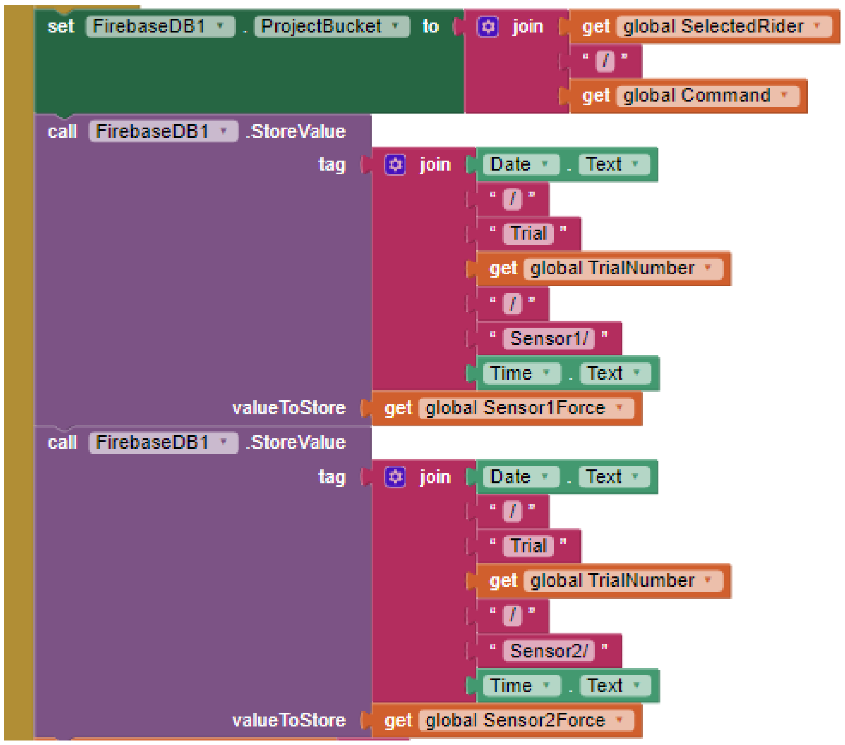

set FirebaseDB1 . ProjectBucket to
call FirebaseDB1 .StoreValue
tag
valueToStore

Figure 25: *Code blocks that controlled where and how the data would be stored in the online database*

To ensure that a rider's force data was stored correctly, the Firebase project bucket was set to look for the nested data under the selected riders name and the specific command. When found, the data would be further nested under the day of the trial, trial number, and whether the force is being received from Sensor 1 (left load cell) or Sensor 2 (right load cell). The force value would then be given a time stamp and stored under its sensor number, as seen in **Figure 24.**



```
set FirebaseDB1 . ProjectBucket to  join  get global SelectedRider
                                          " / "
                                          get global Command
call FirebaseDB1 .StoreValue
                        tag  join  Date . Text
                                   " / "
                                   " Trial "
                                   get global TrialNumber
                                   " / "
                                   " Sensor1/ "
                                   Time . Text
               valueToStore  get global Sensor1Force
call FirebaseDB1 .StoreValue
                        tag  join  Date . Text
                                   " / "
                                   " Trial "
                                   get global TrialNumber
                                   " / "
                                   " Sensor2/ "
                                   Time . Text
               valueToStore  get global Sensor2Force
```

Figure 26: *Code blocks that store sensor data to online database*

When the user was ready to conduct another run for the same command, they had to change the number of the trial they were conducting. If the trial number was not changed, then the new forces would be added to the values already previously supplied by the application during that particular trial number. Therefore, the number of forces would be greater than the number of values found in other trials that were set up correctly.

Apart from sending a byte character to the Arduino UNO for the feedback mechanism, the feedback byte conditions also determined when a trial was considered a success or a failure. A successful run was defined when the feedback condition would send the Arduino the byte character “A,” meaning that both feedback light towers would have the green LED light turned on. If the conditions for “A” were met, then the color of the force numbers on the application would also turn green. A “Success!” message would be displayed at the bottom of the screen, as seen in **Figure 27(a),** and a number “1” would be stored in the “Results” section in the Firebase database as seen in **Figure 27 (c)**.

Any other byte and light configuration indicated a failed run. A failed test would not change the color of the forces displayed in the application, would send a “0” to the database, and would display a “Needs More Practice!” message, as seen in **Figure 27(b)**. The user was finally able to switch the command type by navigating back to the main screen using the home button found at the top of the screen.

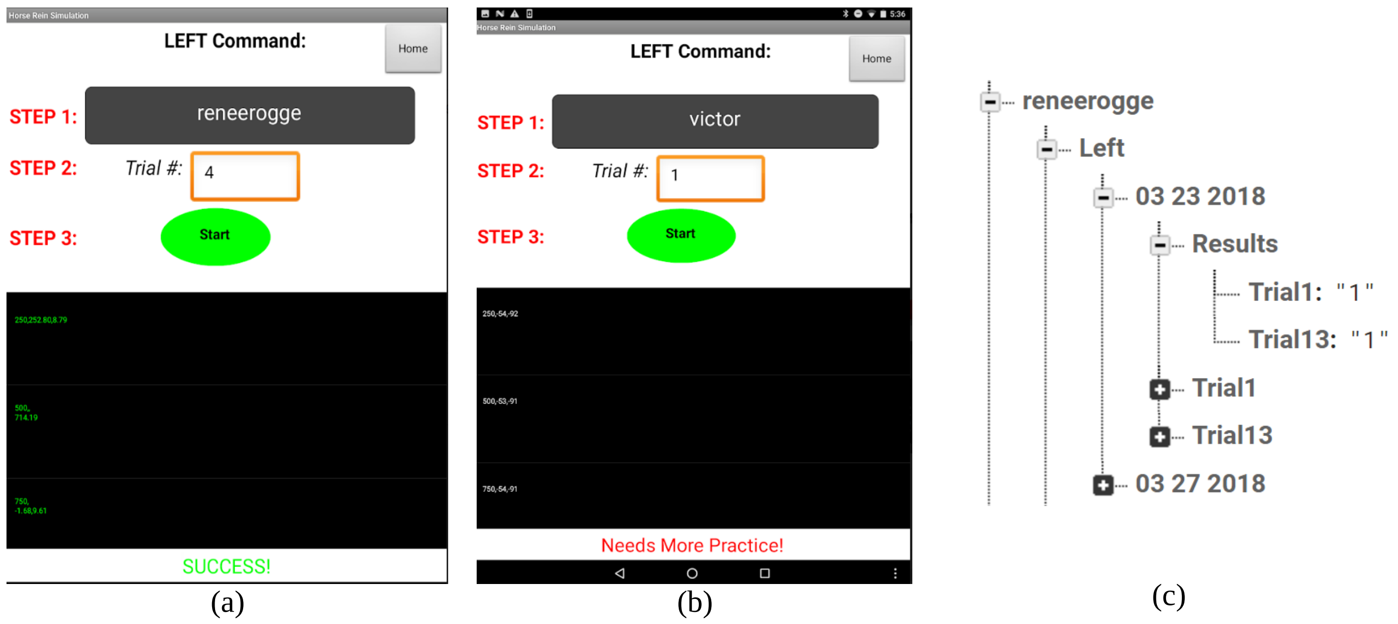
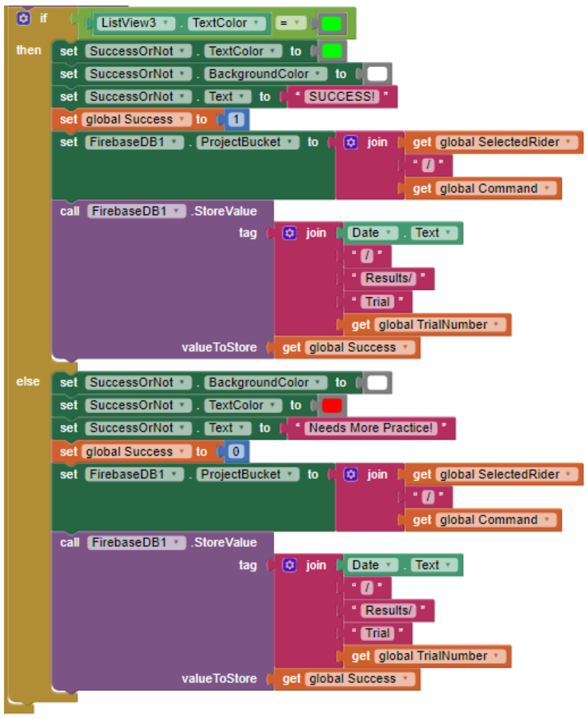


Figure 27: *Test Screen (a) Successful trial indicator (b) Failed trial indicator; (c) Storage in “Results” tab within Google Firebase*



```
if   ListView3 . TextColor = 🟩
then set SuccessOrNot . TextColor to 🟩
     set SuccessOrNot . BackgroundColor to ⬜
     set SuccessOrNot . Text to " SUCCESS! "
     set global Success to 1
     set FirebaseDB1 . ProjectBucket to join get global SelectedRider
                                             " / "
                                             get global Command
     call FirebaseDB1 .StoreValue
                           tag join Date . Text
                                    " / "
                                    " Results/ "
                                    " Trial "
                                    get global TrialNumber
                  valueToStore get global Success
else set SuccessOrNot . BackgroundColor to ⬜
     set SuccessOrNot . TextColor to 🟥
     set SuccessOrNot . Text to " Needs More Practice! "
     set global Success to 0
     set FirebaseDB1 . ProjectBucket to join get global SelectedRider
                                             " / "
                                             get global Command
     call FirebaseDB1 .StoreValue
                           tag join Date . Text
                                    " / "
                                    " Results/ "
                                    " Trial "
                                    get global TrialNumber
                  valueToStore get global Success
```

***Figure 28:** Code blocks for success and failure results*

6.2.2.3 Feedback Byte Conditions Functions:

The feedback mechanism attached to the Arduino UNO relied on the application to send a specific byte character to produce an output. The conditions by which a specific byte was sent depended on the command the rider was attempting to make and the forces being read or exerted on the device's reins. Each command was identified by a number and when the application recognized the command value, it would call the feedback condition function associated with the command, as seen in **Figure 27**. The functions would then determine the byte character to send based on the read forces. These conditions and coded functions could be found in **Tables 2, 3, and 4**. The ranges set for each command were based on the recommendation of the Lakeland Center's staff past experience handling therapeutic horses.

Table 2: *Force ranges used to determine the feedback mechanisms output when executing trials for the "Left" command*

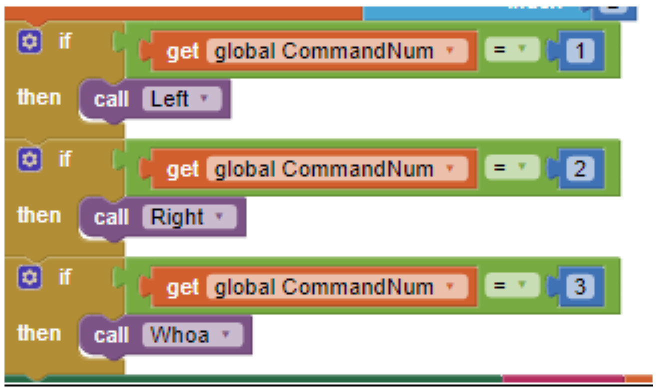
LEFT COMMAND				
		Left Rein Light Tower		
Right Rein Light Tower		Green	Yellow	Red
	Green	Sensor1: $f \geq 3500$: Sensor2: $f \leq 500$	Sensor1: $3500 > f \geq 1500$ Sensor2: $f \leq 500$	Sensor1: $f < 1500$ Sensor2: $f \leq 500$
	Yellow	Sensor1: $f \geq 3500$ Sensor2: $700 \geq f \geq 500$	Sensor1: $3500 > f \geq 1500$ Sensor2: $700 \geq f \geq 500$	Sensor1: $f < 1500$ Sensor2: $700 \geq f \geq 500$
	Red	Sensor1: $f \geq 3500$ Sensor2: $f > 700$	Sensor1: $3500 > f \geq 1500$ Sensor2: $f > 700$	Sensor1: $f < 1500$ Sensor2: $f > 700$

Table 3: Force ranges used to determine the feedback mechanisms output when executing trials for the "Right" command

RIGHT COMMAND				
	Left Rein Light Tower			
Right Rein Light Tower		Green	Yellow	Red
	Green	Sensor1: $f \leq 500$ Sensor2: $f \geq 3500$:	Sensor1: $700 \geq f > 500$ Sensor2: $f \geq 3500$:	Sensor1: $f > 700$ Sensor2: $f \geq 3500$:
	Yellow	Sensor1: $f \leq 500$ Sensor2: $3500 > f \geq 1500$	Sensor1: $700 \geq f \geq 500$ Sensor2: $3500 > f \geq 1500$	Sensor1: $f > 700$ Sensor2: $3500 > f \geq 1500$
	Red	Sensor1: $f \leq 500$ Sensor2: $f < 1500$	Sensor1: $700 \geq f \geq 500$ Sensor2: $f < 1500$	Sensor1: $f > 700$ Sensor2: $f < 1500$

Table 4: *Force ranges used to determine the feedback mechanisms output when executing trials for the "Whoa" command*

WHOA COMMAND		
All Green	**All Yellow**	**All Red**
$\boldsymbol{f > 1000}$ $\boldsymbol{diff \leq 500}$	$f > 1000$ $800 \geq diff > 500$	$f > 1000$ $diff > 800$



```
if   get global CommandNum = 1
then call Left
if   get global CommandNum = 2
then call Right
if   get global CommandNum = 3
then call Whoa
```

Figure 29: *Code blocks which recognized the command number and called the feedback condition functions*

6.2.2.4 Google Firebase/ Data Storage:

The Google Firebase database extension was added to the application through the use of MIT App Inventors experimental component for Google Firebase. In order to connect the database to the application, a new database had to be created in Firebase. This was done by visiting the Google Firebase website and establishing a new database profile.

After the database was created, the URL found in the “DATA” tab of the newly made database was copied and pasted in the properties of the Firebase experimental component in MIT App Inventor 2, as shown in **Figure 29**.

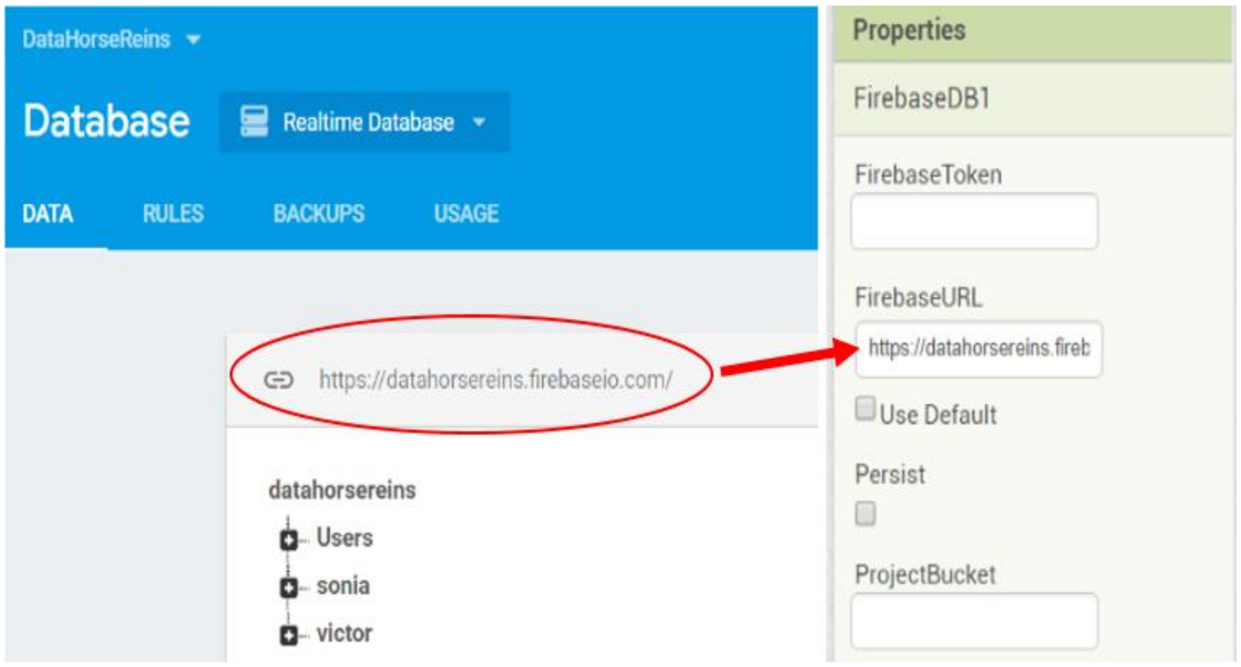
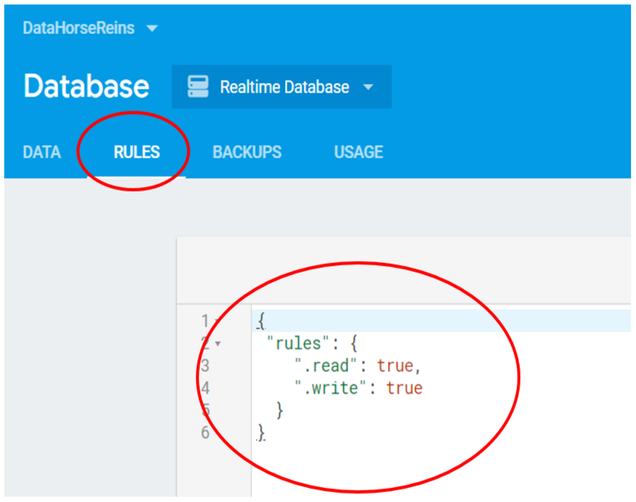


Figure 30: *Copied URL from Firebase website to the properties of the Firebase experimental component in MIT App Inventor 2*

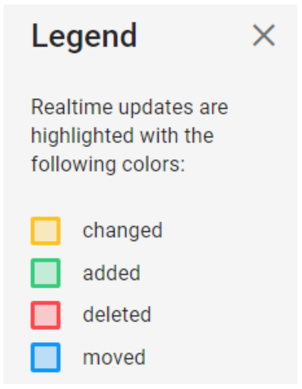
Permissions needed to be given to allow for changes within the database through the use the mobile application; therefore, the rules for the database had to be changed within the Firebase website. This was accomplished by clicking on the “Rules” tab and modifying the code to the code seen in **Figure 30**.



```
{
  "rules": {
    ".read": true,
    ".write": true
  }
}
```

Figure 31: *Modified rules for the database, which would allow the database to be edited by an outside source such as the mobile application.*

The manner in which the data was saved was established by the mobile application. As the database changed, Firebase would provide a real-time update of the data by highlighting objects in specific colors. The figure below provides an explanation of the highlighted objects color. The entirety of the data could then be exported and saved as a .JSON file for later analysis.


Legend

Realtime updates are highlighted with the following colors:

🟨 changed
🟩 added
🟥 deleted
🟦 moved

Figure 32: *Firebase object highlighting colors and meanings*

6.2.3 Description of the Data

The data collected by the device included readings from both load cell sensors and their associated timestamps determining the time of day the force was captured. During each run, the simulator collected a force every 250 milliseconds for 10 seconds. These forces were then split into two categories under the names “Sensor1” and “Sensor2” representing the respective left and right load cell mounted on the horse head. This data was then exported from the online database and loaded into RStudio script to produce an automated progress report for the rider.

6.2.4 Data Processing

The RStudio script processed the exported data and generated a progress report that supplied the simulator users with the overall progress of the individual rider they wished to review. The progress report included the daily success rate for each command: left, right, and whoa/stop. It also provided the user with the change in success rate comparing the last simulated horse ride to the rider's first simulated ride. The equations used to produce both the daily success rate and change in success are shown in **Equations (4) and (5).**

Equation 4: $$Success\ Rate = \frac{Number\ of\ Successful\ Trials}{Total\ Number\ of\ Trials} * 100$$

Equation 5: $$Change\ in\ Success\ Rate = \frac{(Rate_n - Rate_1)}{(Rate_1)}$$

7. DATA USED TO VERIFY FORCES READ

The next sections will discuss the methods by which forces were recorded and documented for statistical analysis. The basis of this thesis was to verify that the forces read and documented by the device were accurate within a certain range and that the system was reliable. The tolerance was set to ±150 g (.33 lb) based on the recommendation of the Lakeland Center.

7.1 Values Received by Sensors

The data was collected without adding the horse reins to the system and the forces were obtained by hanging weights directly off the load cells. These forces were recorded under three different temperature conditions: room temperatures, high temperatures, and low temperatures. The temperature of the room was regulated by an AC unit which maintained the insulated room at a set temperature. These temperature conditions were chosen in order to investigate whether temperature had a significant effect on the system.

Weights were added to the system in increasing order, seen in **Table 2**. The collected data was then split into forces read by each individual sensor under three separate temperature conditions; therefore, there were a total number of 6 data sets collected: 3 for Sensor 1 (Left Sensor) and 3 for Sensor 2 (Right Sensor). The data was collected in the order of high temperature, low temperature, and room temperature, where the sensors were exposed to the three temperature conditions in a controlled environment.

Table 5: *Weights applied to the system*

Applied Weights (g)	
50	750
100	800
150	850
200	900
250	950
300	1000
350	1500
400	2000
450	2500
500	3000
550	3500
600	4000
650	4500
700	5000

7.1.1 Room Temperature (65-70ºF)

Data collected at room temperatures were collected within 5 consecutive days at a rate of 10 trials per day.

7.1.2 High Temperature (85-95ºF)

Data collected at high temperatures were collected within 50 consecutive days at the rate of one trial per day.

7.1.3 Low Temperature (60-55ºF)

Data collected at low temperatures were collected within 50 consecutive days at a rate of one trial per day.

8. RESULTS

The basis of this thesis was to verify that the forces read by the device were accurate within a certain tolerance range and to verify the reliability of the system. The tolerance range was determined by the Lakeland Center and found to acceptable within ± 150 grams of the actual weight applied. Another point of interest was to investigate whether or not temperature had a significant effect on the systems error considering the device would be stored in a barn at the Lakeland Center's facility.

8.1 Statistical Analysis

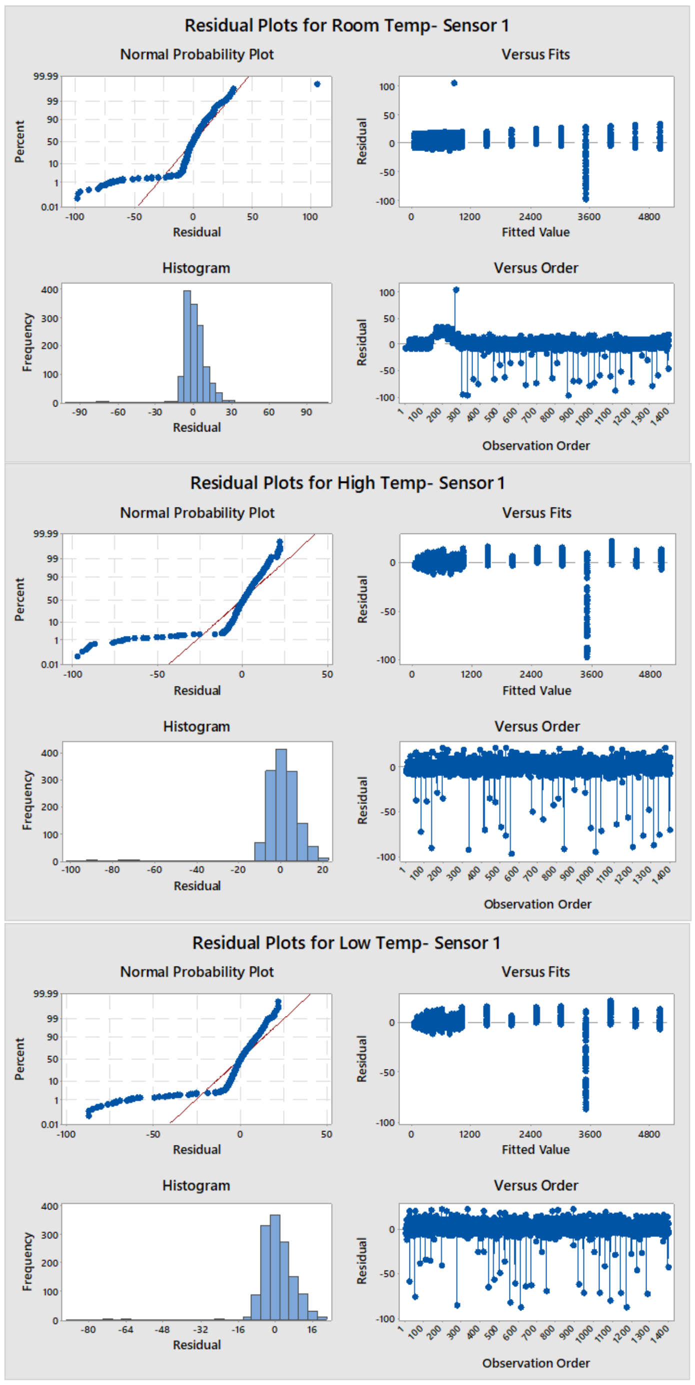


Figure 33: *Residual plots for Sensor 1 data sets; room temperature, high temperature, and low temperature. Residual plots display the data having a pattern, which indicates the data has no constant variance; therefore, a linear model would not accurately fit the data.*

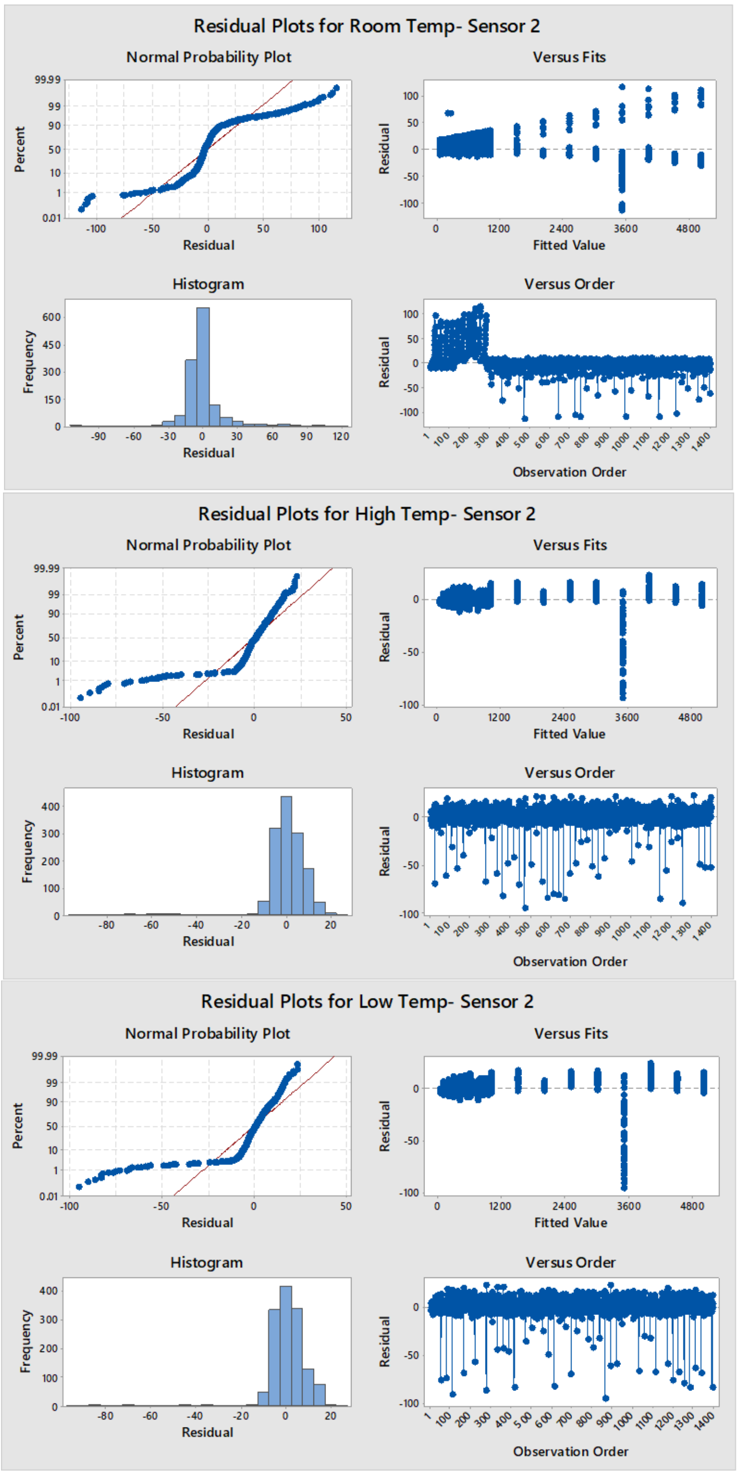


Figure 34: *Residual plots for Sensor 2 sets; room temperature, high temperature, and low temperature. Residual plots display the data having a pattern, which indicates the data has no constant variance; therefore, a linear model would not accurately fit the data.*

Linear regression models and residuals plots of the data were generated using Minitab 2017 in order to assess whether the errors within the data were normally distributed and to determine if there was constant variance. **Figures 33 and 34** show that the residuals were not normally distributed and that the variance of the data would increase as more weight was applied to the system. In order to more accurately model the data, it needed to be resampled; so a nonparametric pairwise bootstrap procedure was used.

RStudio was used to generate 5,000 resamples of the data. Confidence intervals for both slope and intercept were then generated for each dataset to test for the linearity of the data, as seen in **Table 3**.

$$\textbf{\textit{Equation 6}}: Force\ Read = B_o + B_1(Force\ Applied)$$

Table 6: *Confidence intervals for resampled data. An inference on the linearity of the system can be based on the B_1 and B_0 values. The confidence intervals assert with 95% confidence that the B_1 and B_0 values lie within the ranges shown. These confidence intervals also indicate the slope of the linear models vary little from the ideal slope of 1. The confidence intervals for the intercepts indicated that the values obtained from the sensors were often biased by at least 3 grams on average*

	Temperature	**95% CI** $\boldsymbol{B_1}$	**95% CI** $\boldsymbol{B_0}$
Left Sensor	Room	(0.9988 , 1.002)	(2.782 , 3.935)
	High	(0.9982 , 0.9995)	(2.304 , 3.220)
	Low	(0.9981 , 0.9995)	(2.168 , 3.047)
Right Sensor	Room	(1.002 , 1.005)	(0.921 , 3.085)
	High	(0.9981 , 0.9993)	(2.019 , 2.924)
	Low	(0.9978 , 0.992)	(2.261 , 3.192)

The confidence intervals shown in **Table 6** assert with 95% confidence that the B_1 and B_0 values lie within the ranges shown. These confidence intervals also indicate the slope of the linear models vary slighty from the ideal slope of 1. The datasets with slopes less than 1 would therefore read forces lower than the one applied. The confidence intervals with slopes greater than 1 would read forces greater than the weight applied when the intercept is not taken into consideration. The confidence intervals for the intercepts indicated that the values obtained from the sensors were often biased by at least 3 grams on average which can be negated considering that 3 grams is equivalent to less than a tenth of a pound of force.

The residual plots also indicated that as the applied weight was increased so did the variance of the read data. In order to further investigate this change in variance and to examine the precision of the system, the 97.5th and 2.5th quantiles were estimated using quantile regression. The parameter estimates shown in Table 4 were used to estimate the high and low quantile limits for each dataset.

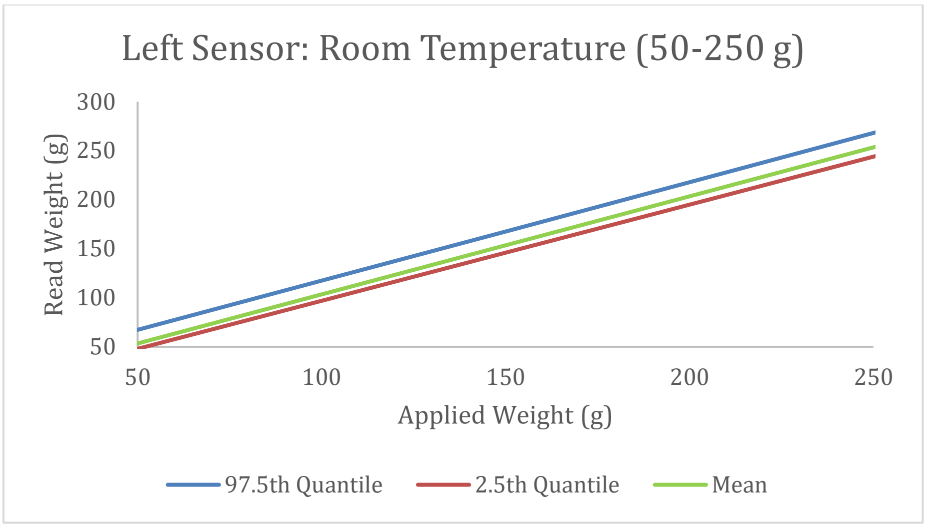
Equation 7: $97.5th\ quantile\ force = \gamma_o + \gamma_1(Applied)$

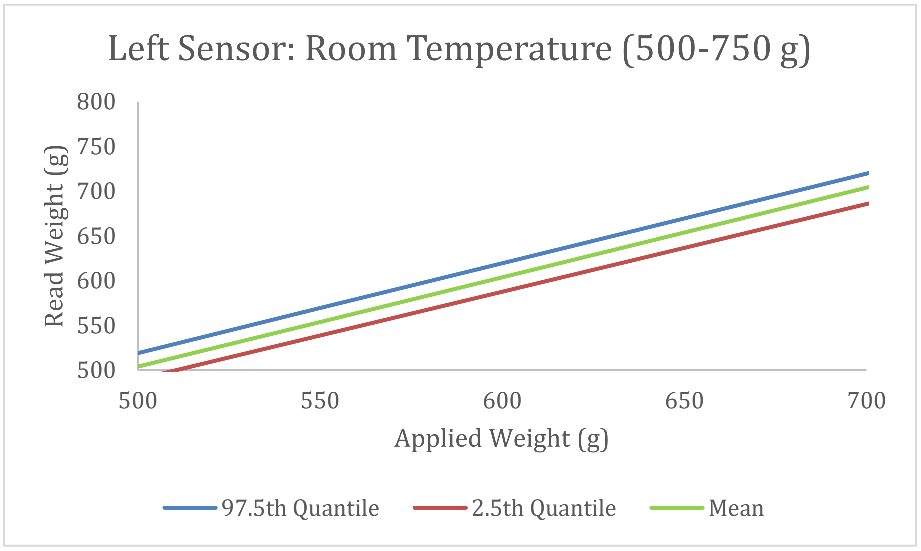
Equation 8: $2.5th\ quantile\ force = \alpha_o + \alpha_1(Applied)$

Table 7: *Constants for the 97.5th and 2.5th quantile linear model equations based on sensor number and temperature*

		97.5th Quantile		**2.5th Quantile**	
	Temperature	γ_1	γ_0	α_1	α_0
Left Sensor	Room	1.0034	17.132	0.98061	-1.1212
	High	1.0020	12.000	0.97969	-0.92308
	Low	1.0018	11.9412	0.98029	-1.0145
Right Sensor	Room	1.0232	13.513	0.98769	-2.9231
	High	1.0017	11.500	0.98029	-1.0145
	Low	1.0014	12.444	0.98	-1

Upon observation of the generated plots **(Figures 35-40)**, it was apparent that as weight was applied to the load cells, the average read values began to shift away from the 2.5th quantile and closer to the 97.5th quantile in all sets of data.





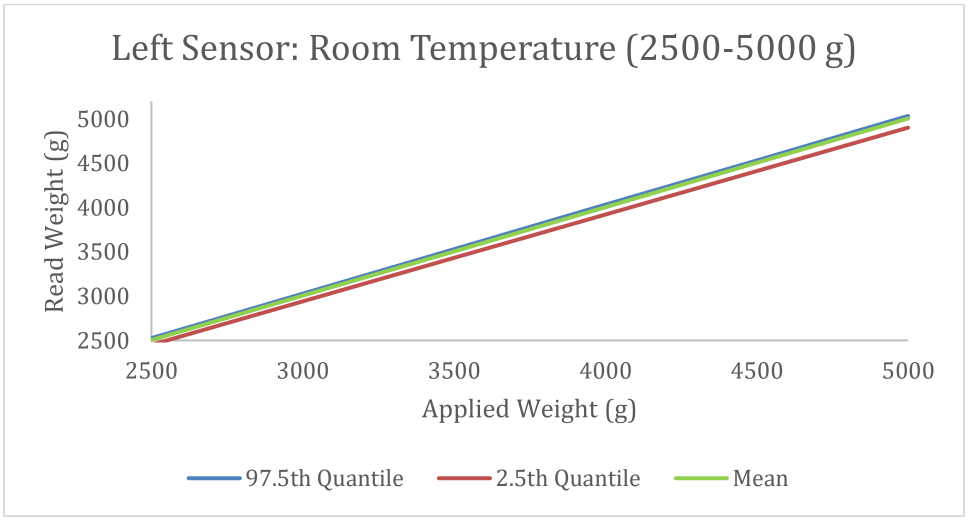
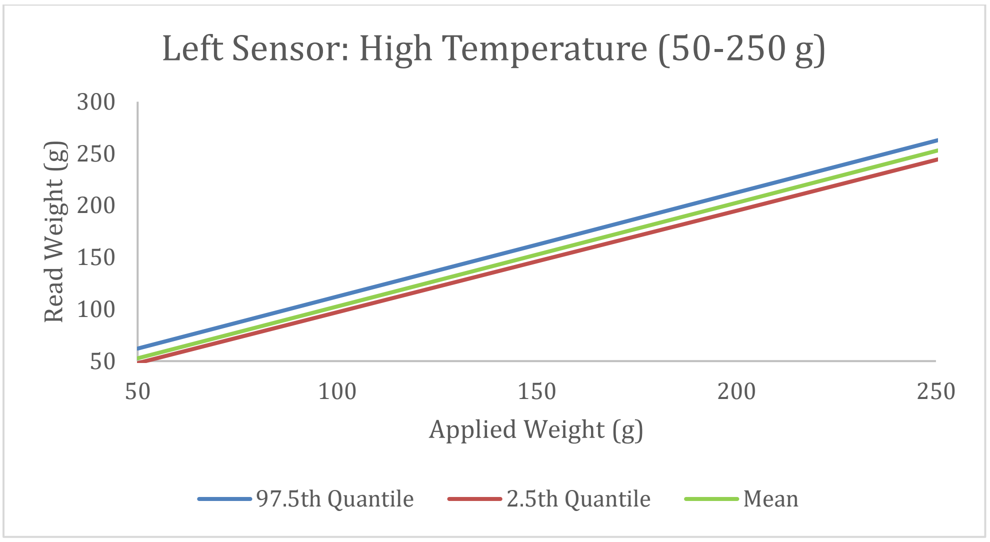
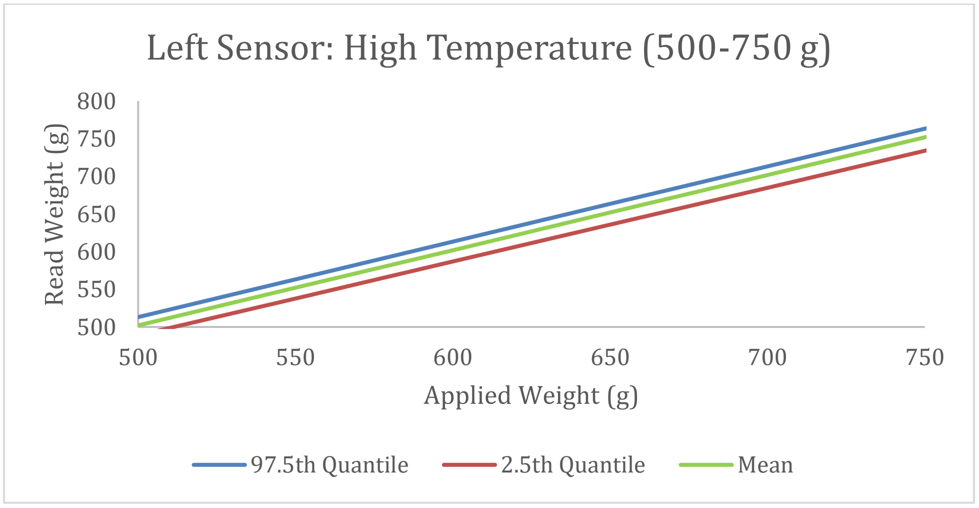


Figure 35*: Regression plots for Left Sensor at Room Temperature. As more weight is applied the mean of the data shifts from the lower quantile to the higher quantile*





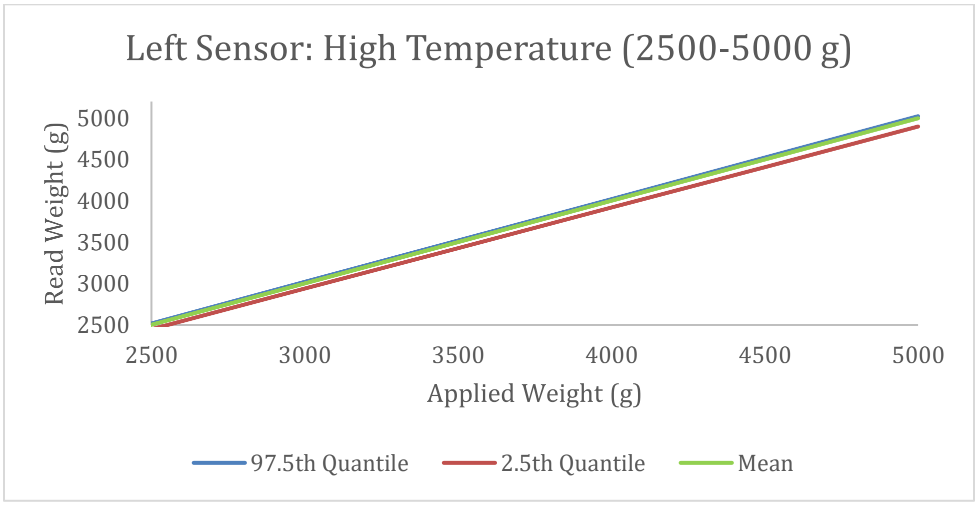
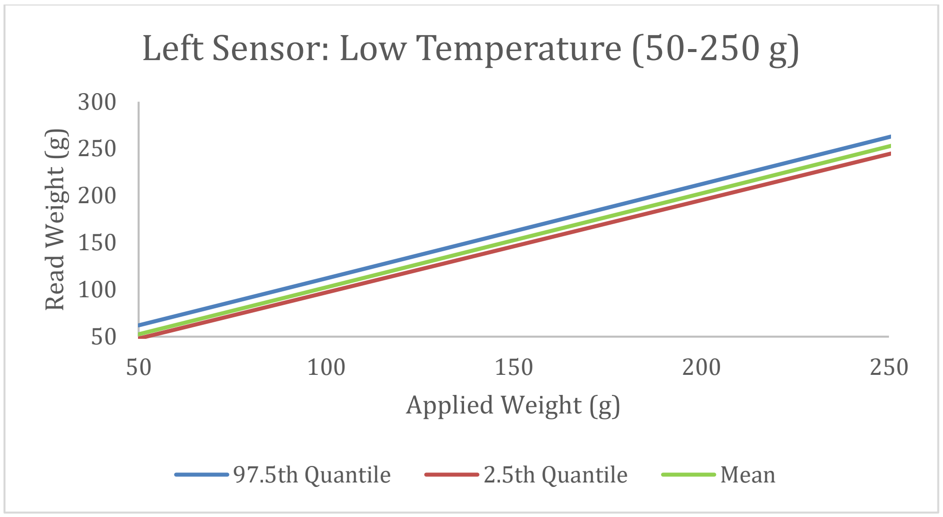
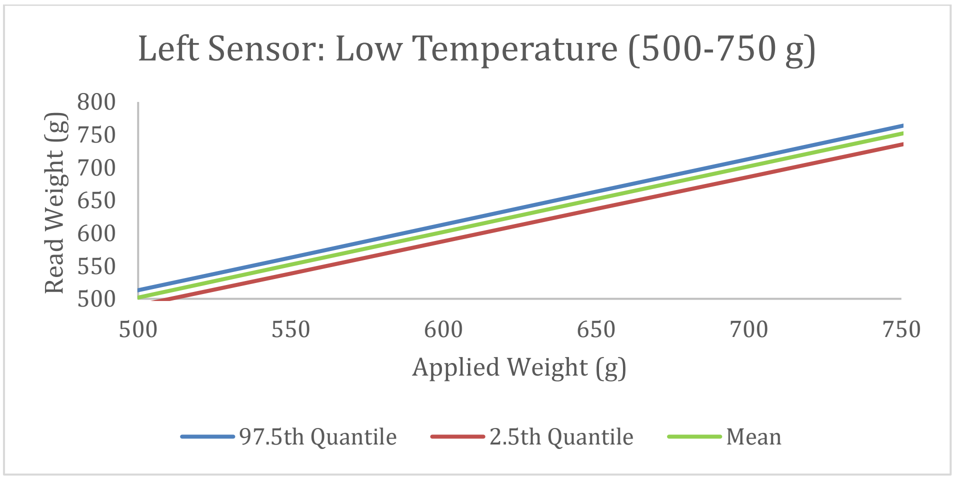


Figure 36: *Figure 35: Regression plots for Left Sensor at High Temperatures. As more weight is applied the mean of the data shifts from the lower quantile to the higher quantile*





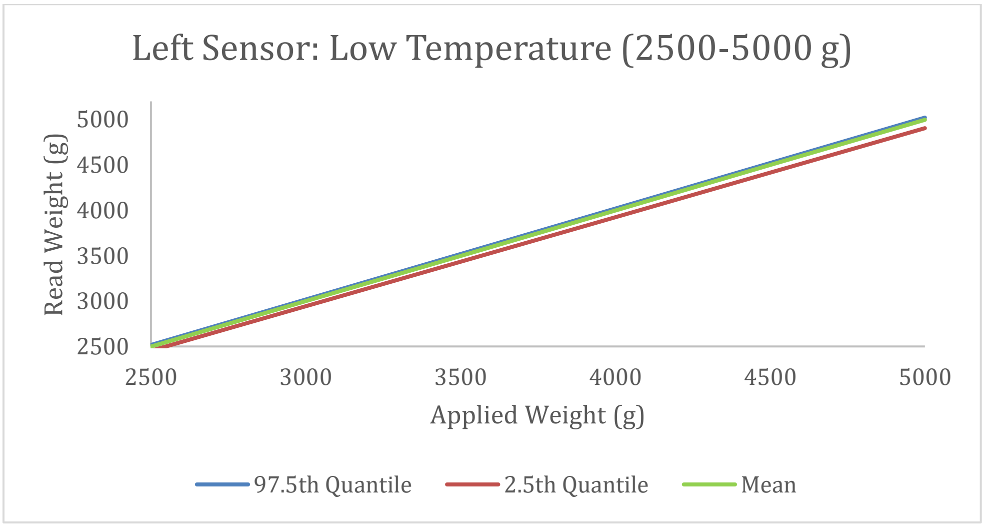
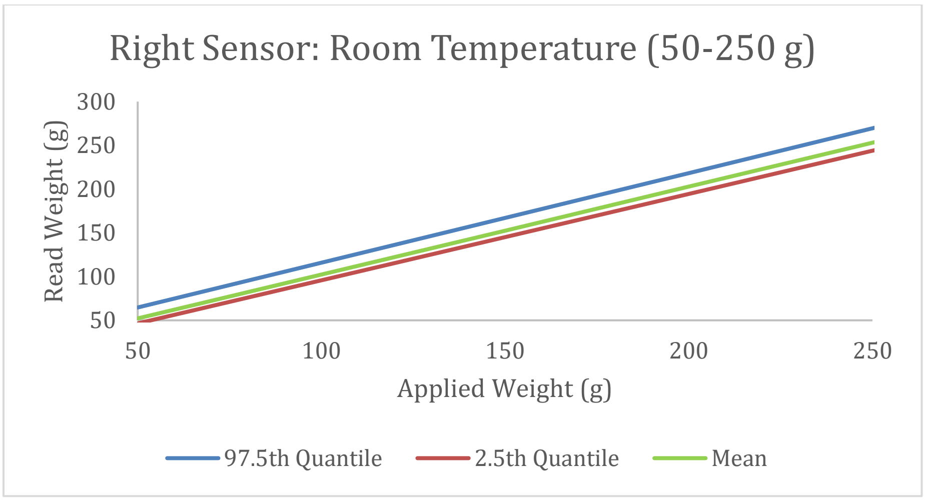
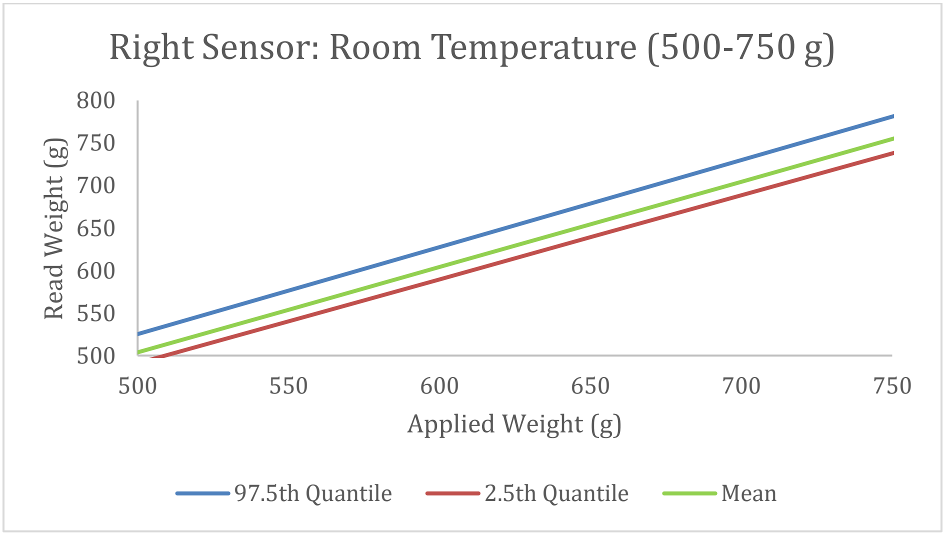


Figure 37: *Figure 35: Regression plots for Left Sensor at Low Temperatures. As more weight is applied the mean of the data shifts from the lower quantile to the higher quantile*





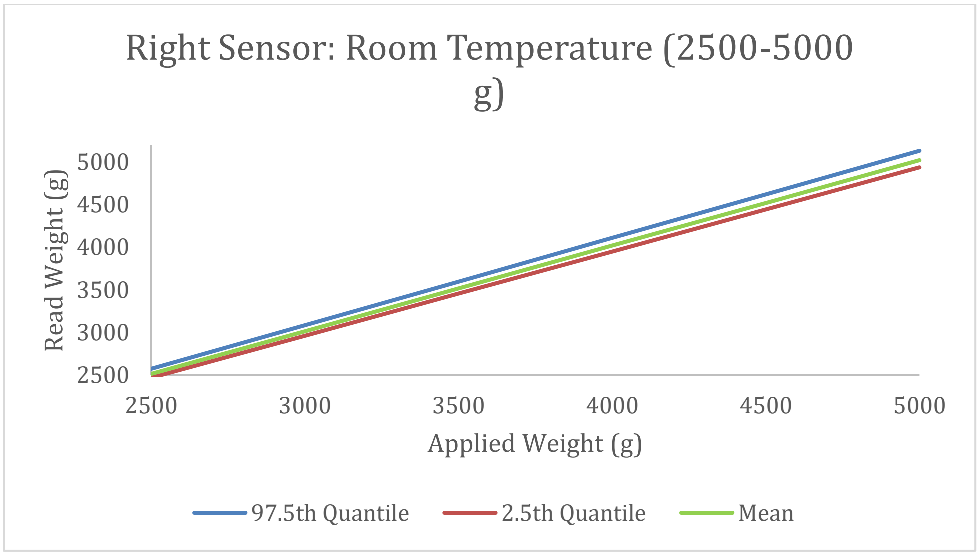
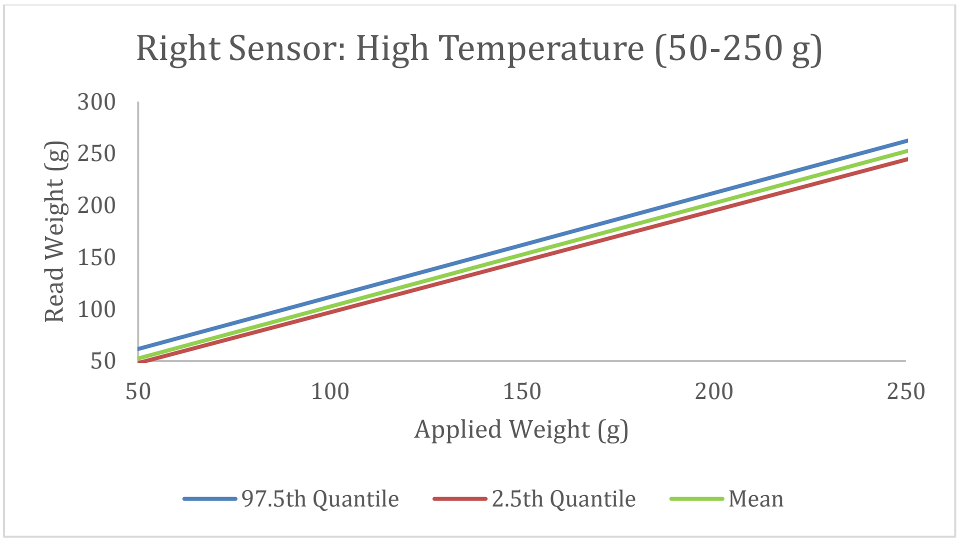
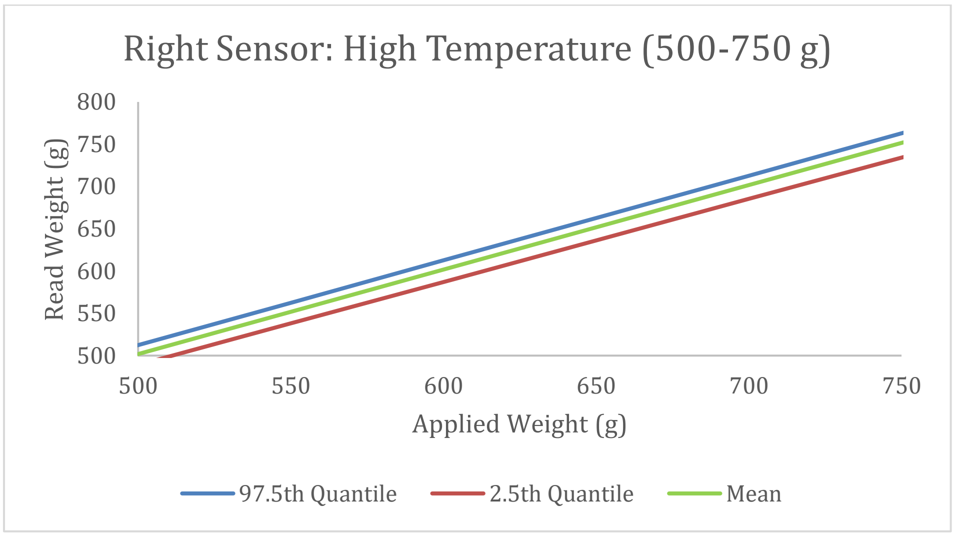


Figure 38: *Regression plots for Right Sensor at Room Temperature. As more weight is applied the mean of the data shifts from the lower quantile to the higher quantile*





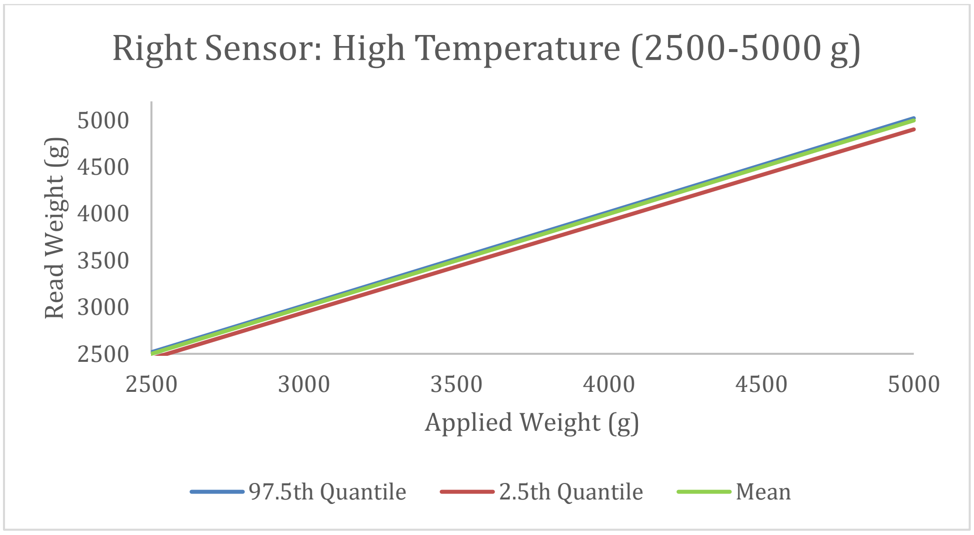


Figure 39: *Regression plots for Right Sensor at High Temperatures. As more weight is applied the mean of the data shifts from the lower quantile to the higher quantile*

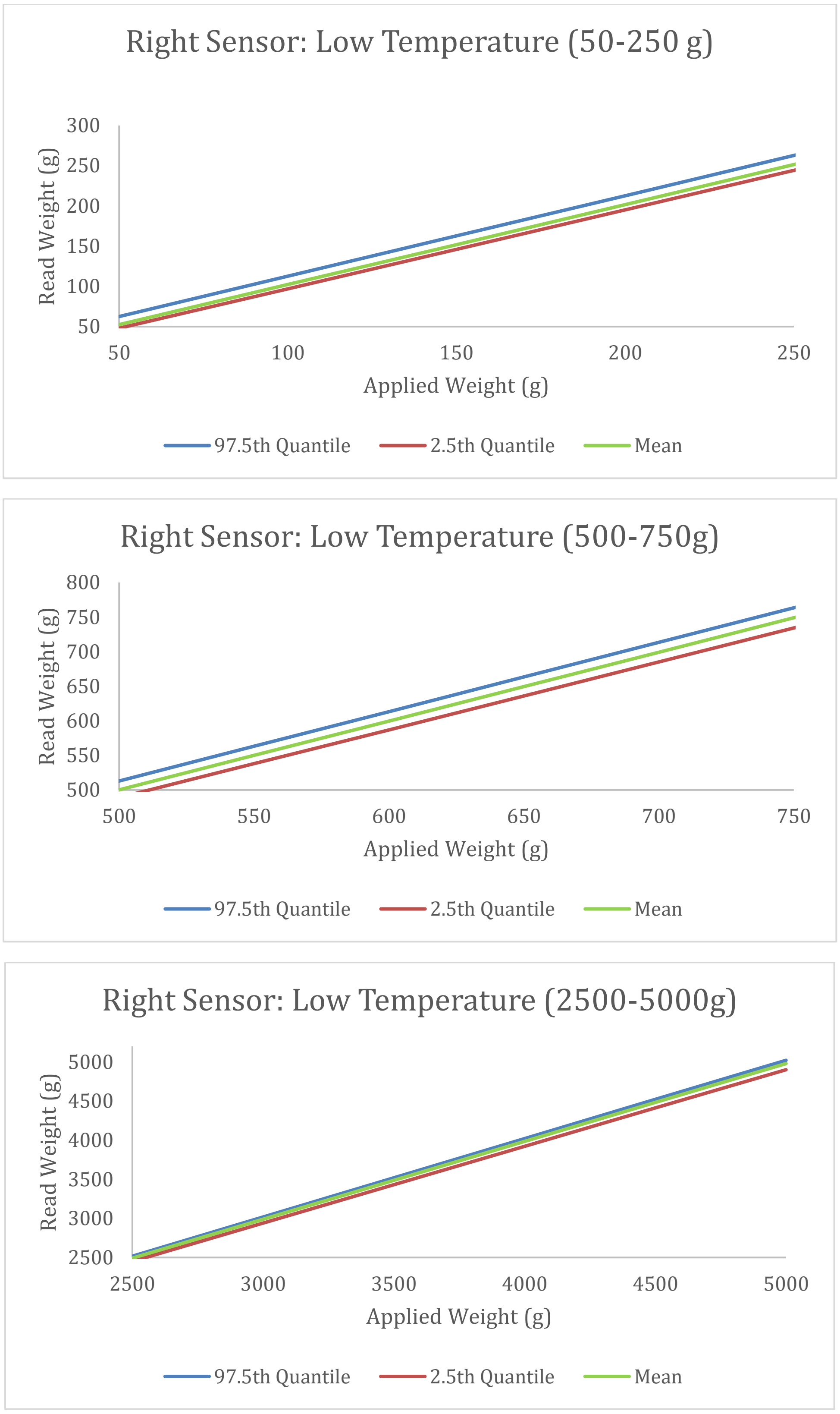


Figure 40*: Regression plots for Right Sensor at Low Temperatures. As more weight is applied the mean of the data shifts from the lower quantile to the higher quantile*

The effect of temperature on the overall system was modeled using **Equation 9**, for which 95% confidence intervals for each B_1 value were computed via bootstrapping and are shown in **Table 8**.

Equation 9: $Force\ Read = B_0 + B_1(Applied\ Force) + B_2(High\ Temp) + B_3(Low\ Temp)$

Table 8: *Linear model constants for temperature effect on data*

	Left Sensor	**Right Sensor**
$\boldsymbol{B_0}$	(3.305 , 4.559)	(5.763 , 7.333)
$\boldsymbol{B_1}$	(0.9987 , 0.9995)	(1.00, 1.001)
$\boldsymbol{B_2}$	(-2.414, -0.637)	(-7.496, -4.940)
$\boldsymbol{B_3}$	(-2.468, -0.695)	(-7.503 , -4.927)

These values suggested that there is an effect on the system based on the temperature. The negative values for B_2 and B_3 indicate that the forces read by the system at high and low temperatures were lower than the forces read at room temperature.

8.2 Quality Analysis

A capability analysis of the system under each temperature conditions was performed in order to investigate further its ability to meet the hippotherapy center's specifications. The capability study produced a number of capability ratios that estimated the variability in the standard deviation of the process, but before any conclusions were made, a number of conditions had to be met. The data had to meet the assumption of normality and independence.

Since the data obtained was discrete, a Ryan-Joiner probability plot was used to test for its normality, autocorrelation plots tested for independence. I-MR plots were used also to check if the system was in control (i.e, forces were within 3 standard deviations from the process mean). The Individuals (I) chart plotted the values for each individual observation around the mean value ($\bar{X}$). The upper and lower control limits (UCL, LCL) established the bounds for data 3 standard deviations from the mean of the data. The Moving Range (MR) provided information on the variation of the data and made trends apparent.

Based on the normality tests, the data was found to be non-normal. In order to meet the criteria of normality, the data were transformed using a Johnson transformation. The autocorrelation plots revealed that the data was independent whereas the I-MR charts displayed the data to be stable since the range of the data stayed within 3 standard deviations from its mean range value. There were some points beyond the upper control limit (UCL) and lower control limit (LCL) within the MR chart. The MR chart displayed the variation within the data points, and those beyond the bounds could be attributed to special cause variation where either the device was moved, or the weight applied was not allowed to settle before the reading was recorded. An example of the I-MR chart is shown in **Figure 4.**

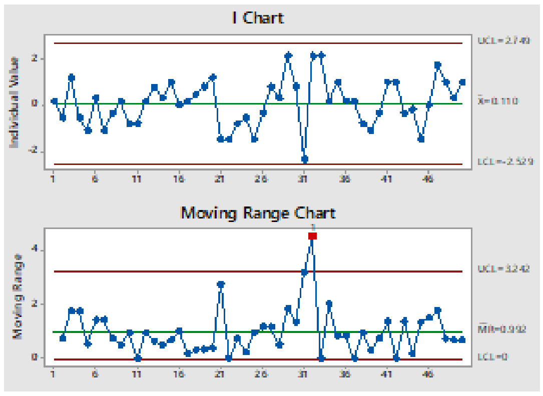


Figure 41: *I-MR chart of Sensor 1 at low temperatures when a 500g weight was added to the system. The red dot displays a Type 1 error, meaning that it fell beyond 3 std. dev from the mean range. This may have happened because the system was moved on that particular day.*

Once the assumptions were met, the data was split into three force ranges, and errors for each range were predicted by observing the highest error found within a specific force range, as shown in **Table 9**.

Table 9: *Predicted error for the forces read by the system based on highest error found within each force range*

Force (g)	Error (g)
$F < 750$	± 30
$750 < F < 2500$	± 50
$F \geq 2500$	± 150

Capability plots were then made providing upper and lower specification limits (USL, LSL) based on predicted error margins, shown in **Table 10.**

Table 10: *USL and LSL for capability plots*

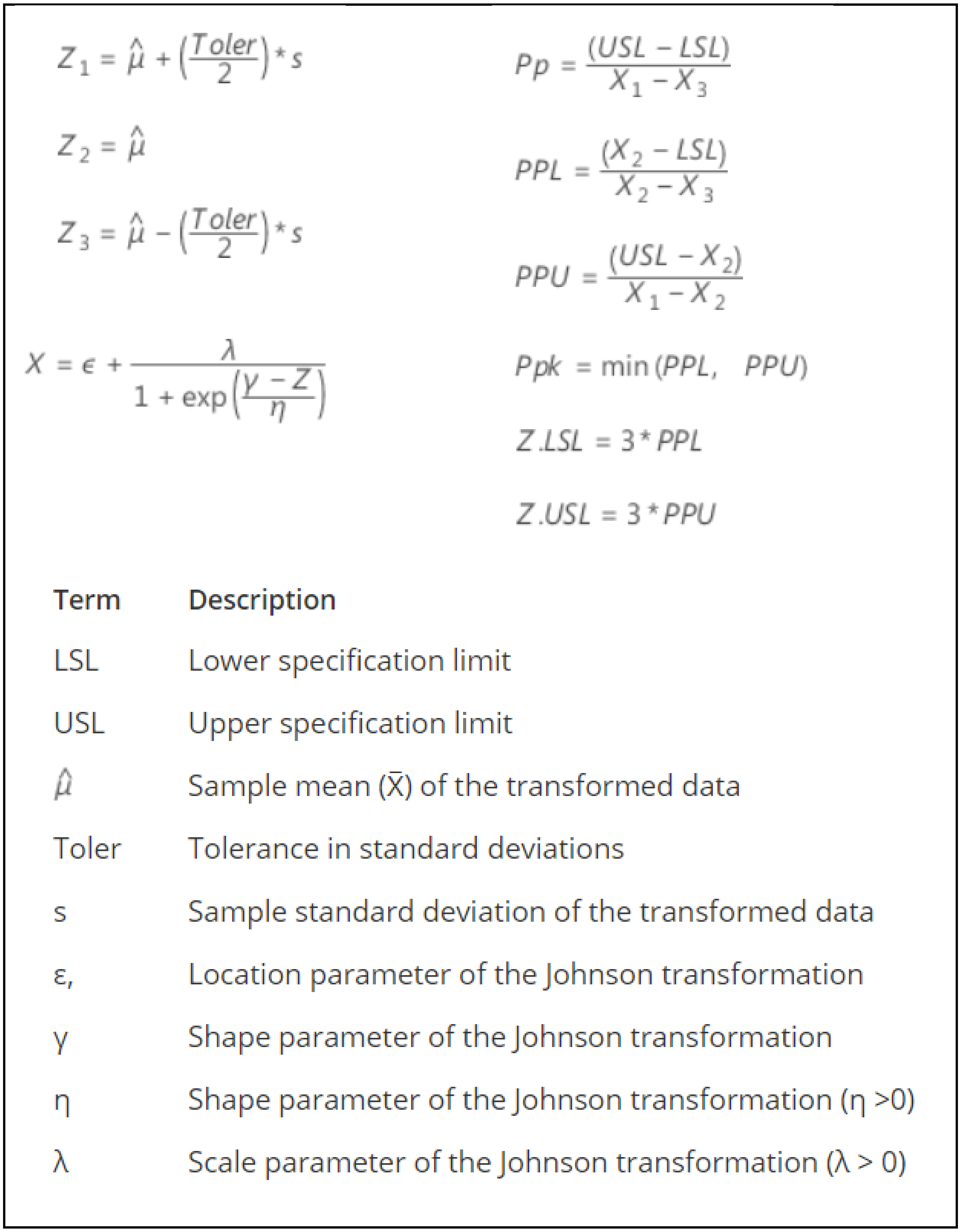
Force (g)	Error (g)	USL (g)	LSL (g)
500	± 30	470	530
1000	± 50	950	1050
5000	± 150	4850	5150

The capability ratios of the system were used to determine whether the variability of the system existed within the specified USL and LSL and how well the spread of the data fit within the specified limits, as seen in **Table 7**. Minitab automatically generated values for P_p and P_{pk} as opposed to C_p and C_{pk} values because the Johnson transformation produced values that changed the systems standard deviation used to calculate the C_p and C_{pk} and therefore the Minitab software defaulted and used the overall standard deviation to calculate the P_p and P_{pk} values. P_p measures the variability in the system and helps determine the capability of the system to operate within the set limits. P_{pk} measures how centered the process is. If the process is not centered, it is deemed not acceptable since the shift can cause the point to appear beyond the specified limits. Values greater than or equal to one for P_p and P_{pk} values indicate that the system operates within specifications whereas if P_p and P_{pk} are equal then the process is centered.

The ratios for the room temperature data were not obtained due to the small number of subgroups within the datasets (n<10). The subgroup size for the high and low-temperature data was 1, and therefore there were 50 subgroups for both datasets, and the I-MR chart was the appropriate model to test whether or not the data was in control.

Whereas the subgroup size of the room temperature data was 10, so there were only 5 total subgroups. Because the subgroup size was greater than 1, the appropriate chart to test for control would be an $\overline{X}$-R chart, which requires the subgroup number to be greater than 10 in order to identify a controlled system appropriately. Since control could not be determined for the system, capability values could not be obtained for the room temperature dataset.

The capability ratios P_p and P_{pk} were calculated using Minitab's alternative method [11] since the natural log in the Johnson transformation posed a problem for transforming the specified limits (USL and LSL) into LSL* and USL*. The alternative method was developed by Minitab in order to calculate indices when the limits were outside the range of the Johnson transformation function. This method is shown in **Figure 42**.



$$Z_1 = \hat{\mu} + \left(\frac{Toler}{2}\right) * s$$

$$Z_2 = \hat{\mu}$$

$$Z_3 = \hat{\mu} - \left(\frac{Toler}{2}\right) * s$$

$$X = \epsilon + \frac{\lambda}{1 + \exp\left(\frac{\gamma - Z}{\eta}\right)}$$

$$Pp = \frac{(USL - LSL)}{X_1 - X_3}$$

$$PPL = \frac{(X_2 - LSL)}{X_2 - X_3}$$

$$PPU = \frac{(USL - X_2)}{X_1 - X_2}$$

$$Ppk = \min(PPL, \quad PPU)$$

$$Z.LSL = 3 * PPL$$

$$Z.USL = 3 * PPU$$

Term	Description
LSL	Lower specification limit
USL	Upper specification limit
$\hat{\mu}$	Sample mean ($\bar{X}$) of the transformed data
Toler	Tolerance in standard deviations
s	Sample standard deviation of the transformed data
ε,	Location parameter of the Johnson transformation
γ	Shape parameter of the Johnson transformation
η	Shape parameter of the Johnson transformation (η >0)
λ	Scale parameter of the Johnson transformation (λ > 0)

Figure 42: *Equations from Minitab 18 Support page used to find the Pp and Ppk values of the system [11]*

Table 11: *Pp and Ppk values for Sensor 1(Left) and Sensor 2 (Right). The Pp and Ppk values indicated that the overall system variation were within the USL and LSL, therefore the system operated within specifications. Pp and Ppk greater than 1 indicate that the system operate within the specified limits. If the Pp and Ppk are equal, the spread of the system was centered. Values could not be determined for room temperature data since not enough data was collected to increase the number of subgroups to fit a $\bar{X}$-R. An individuals chart could not be used to model the room temperature data because the subgroup size was greater than one and therefore the capability analysis would rely on the variance within the subgroups to produce capability ratios as opposed to the use of the overall variance of the system.*

	Temperature	**Weight (g)**	P_p	P_{pk}
Left Sensor	Room	500	-	-
		1000	-	-
		5000	-	-
	High	500	3.44	3.19
		1000	5.20	4.38
		5000	14.23	13.60
	Low	500	3.26	2.41
		1000	4.52	3.26
		5000	12.92	12.18
Right Sensor	Room	500	-	-
		1000	-	-
		5000	-	-
	High	500	3.39	2.62
		1000	4.60	3.39
		5000	13.90	11.53
	Low	500	3.26	2.41
		1000	4.81	3.87
		5000	14.74	12.61

9. DISCUSSION

The statistical models validated that the forces received from the load cells were precise within the set tolerance of ±150 grams. This was verified by the range of the confidence intervals for the B_1 parameter for each dataset. These ranges indicated that the produced values were similar to the applied weight. The B_0 confidence intervals indicated that the forces read differed from the applied weight value on average by at least 3 grams. This discrepancy in the read values could be highly attributed to human error, such as a slight error in the calibration equation or over tightening of the bolts holding the sensors to the support bar. The quantile regression plots further verified that 95% of the read forces tend to lie within the bounds of the 97.5th and 2.5th quantiles. The plots show the error could be as high as ±30 grams for forces below 750 grams, ±50 grams for forces between 750 and 2500 grams, and as high as ±150 grams for forces greater than 2500 grams. The capability indices (P_p and P_{pk}) were greater than one for high and low temperature datasets, indicating the process variation was less than the specified limits and because the P_p and P_{pk} values were similar, the spread of the data was close to center. Based on the capability ratios, the system operated well within the specified limits and therefore the error ranges could be incorporated into the system's feedback mechanism to account for the error margins.

Room temperature conditions caused greater variability in the error of the system as opposed to higher or lower temperature conditions. This may have been a direct result of the system needing recalibrating since the room temperature data was the last to be collected. Although capability tests could not be generated for the room temperature data, the quantile plots suggest that the read forces error would lie within the ranges previously mentioned.

A number of factors may have affected the result of the read force values. If the sensors were screwed into the support bar too tightly, the natural movement of the load cell might have been restricted when the forces were applied. Therefore the strain gages were not able to properly read the correct applied weight. When obtaining the data, there were also times that force values were collected when the applied weights had not entirely settled on the load cells (i.e, weights swinging). The sensors were also calibrated once throughout the time of the data collection; therefore, the errors found in the data may be attributed to the fact that the sensors may have needed to be recalibrated. The calibration equations used for the sensors may have also introduced a margin of error to the read forces.

10. LIMITATIONS

Although the device and the accompanying research reached its goals, there were a number of limitations to the project. If time permitted, it would have been beneficial for research to be conducted on conditions other than the variability of temperature. Tensile and fatigue testing of the load cells would have provided greater understanding to the limitations of the load cells. Conducting an investigation on different locations for sensor placement (i.e, how far into the bar the sensors should be screwed, what side of the bar should hold the sensor) may have resulted in better placement of the load cells and may have improved the error range for the read force values. There was also a need for investigating the number of trials or runs that could be conducted before the system needed to be recalibrated. This may be accomplished by obtaining readings on a weekly or monthly bases after the device has been in constant use, in order to observe if the error of the system was beyond ±150 g.

11. CONCLUSIONS

Modifications to the “Johnny 5” prototype proved to be beneficial to the overall design of the device. The improvements made provided a simulator that produced precise forces, within the accepted tolerance range, that could be used to help prepare horse riders to guide a horse while riding in the English style successfully. The developed mobile application allowed the user to record and store forces from the device to an online database for later use.

The data collected for validation and verification purposes were obtained by mimicking the direction and magnitude of the force that would be applied by a rider to the reins. The forces read from the device were proven accurate within the range of tolerance provided by the hippotherapy facility. The error of the system could range from ±30 grams for forces below 750 grams, ±50 grams for forces between 750 and 2500 grams, and as high as ±150 grams for forces greater than 2500 grams. Room temperature conditions caused greater variability in the error of the system as opposed to higher or lower temperature conditions. This may have been a direct result of the system needing recalibrating, but the quantile plots suggest that the read forces would lie within the ranges previously mentioned. Overall, the forces of the system were verified, and the results supported that the read forces were within the specified error ranges.

12. FUTURE WORK

The device would benefit from the integration of simulations using the Western riding style and the ability for users to choose rein command difficulty. Integrating the Western riding style simulations would allow for the facility to begin to test clients that have been predetermined to use the style during their therapy sessions and to collect data to assess their progress from session to session. Whereas allowing the user to choose the difficulty of each rider’s simulation would prepare riders to guide horses with different bite sensitivities successfully.

The sensors could be further tested to investigate the effect of the sensor location to the read forces as well as to determine the number of trials the device may undergo before the sensors would need to be recalibrated.

REFERENCES

[1] A. H. INC., "Introduction to Hippotherapy," AHA, Inc, 2016. [Online]. Available: http://www.americanhippotherapyassociation.org/hippotherapy/introduction-to-hippotherapy/. [Accessed march 2018].

[2] Wikipedia, "Equine-assisted therapy," March 2017. [Online]. Available: https://en.wikipedia.org/wiki/Equine-assisted_therapy. [Accessed March 2018].

[3] T. T. K. a. H. Ataseven, "What is hippotherapy? The indications and effectiveness of hippotherapy," NCBI, Jan 2015. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5175116/. [Accessed March 2018].

[4] "Hippotherapy," CerbalPalsy, 2018. [Online]. Available: http://www.cerebralpalsy.org/about-cerebral-palsy/treatment/therapy/hippotherapy. [Accessed March 2018].

[5] C. Menke, " About the Center," Lakeland Center for Theraputic Research, Nov. 2017. [Online]. Available: https://www.lakelandctr.org/. [Accessed March 2018].

[6] K. Blocksdorf, "How to Hold the Reins When Riding," The Spruce, Feb 2018. [Online]. Available: https://www.thespruce.com/how-to-hold-the-reins-1886041. [Accessed March 2018].

[7] Rose-HulmanBESenior- Dark Roast Coffee, "Mr. Giraffe," Rose-Hulman Institute of Technology BBE Dept., Terre Haute, IN, 2015.

[8] Make It Happen, "Johnny 5," Rose-Hulman Institute of Technology BBE Dept, Terre Haute, IN, 2016.

[9] Sparkfun, "Load Cell - 10kg, Straight Bar (TAL220)," Sparkfun, Feb 2016. [Online]. Available: https://www.sparkfun.com/products/13329. [Accessed March 2018].

[10] AdaFruit, "Tower Light - Red Yellow Green Alert Light with Buzzer - 12VDC," Adafruit, [Online]. Available: https://www.adafruit.com/product/2993?gclid=CjwKCAjw4sLVBRAlEiwASblR-

4rEngUDA2y1hbD46dmZSlsomzswqplyDIZTMnm5yScWrv9zrMpQrxoCZqMQAvD_BwE. [Accessed March 2018].

[11] M. 1. Support, "Methods and formulas for Johnson transformed data in Normal Capability Analysis," 2017. [Online]. Available: http://support.minitab.com/en-us/minitab/18/help-and-how-to/quality-and-process-improvement/capability-analysis/how-to/capability-analysis/normal-capability-analysis/methods-and-formulas/johnson-transformed-data/. [Accessed April 2018].

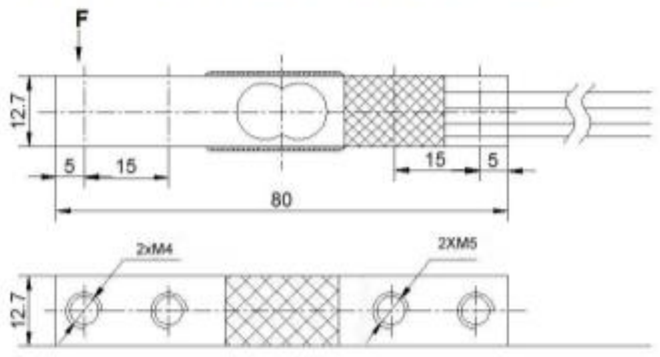
APPENDICES

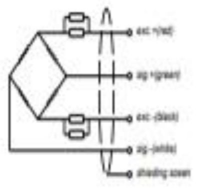
Appendix A: Straight Bar Load Cell Data Sheet

TAL220 **PARALLEL BEAM LOAD CELL**

Features:
- ◆Capacity：3-200kg
- ◆Material: aluminum-alloy or alloy steel
- ◆Type: Parallel beam type
- ◆Defend grade: IP65
- ◆Application : Palm scale, kitchen scale, electronic balance, fishing scale, electronic platform scale and other electronic weighing devices.

Electrical connection and Dimensions:(dimension unit: mm)



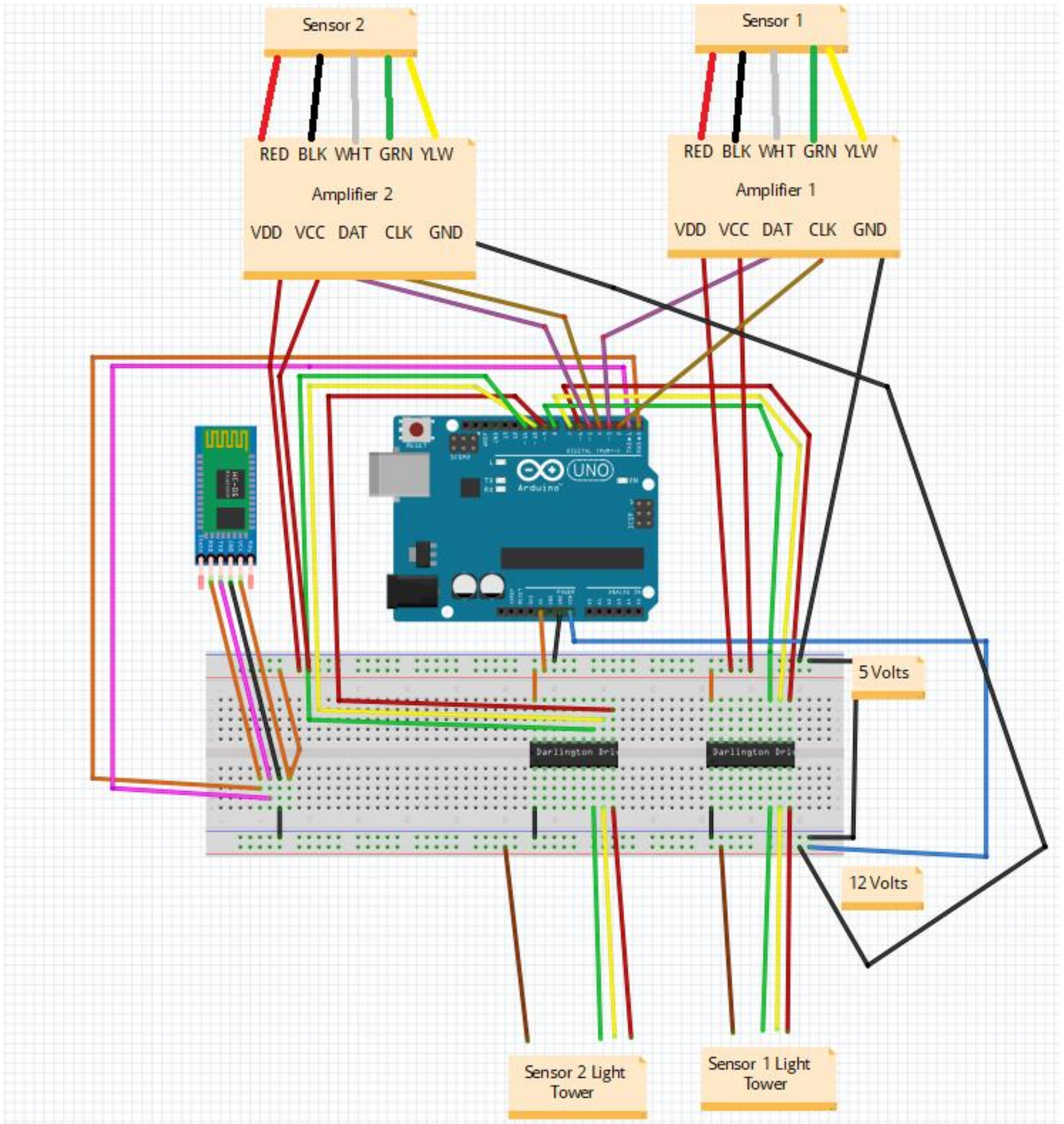


Specifications:		
capacity	kg	3,5,10,20,25,30,50(aluminum); 80,100,120,200(alloy steel)
safe overload	%FS	120
ultimate overload	%FS	150
rated output	mV/V	1.0 ± 0.15
excitation voltage	Vdc	5 ~ 10
combined error	%FS	± 0.05
zero unbalance	%FS	± 0.1
non-linearity	%FS	± 0.05
hysteresis	%FS	± 0.05
repeatability	%FS	± 0.03
creep	%FS/3min	± 0.05
input resistance	Ω	1000 ± 15
output resistance	Ω	1000 ± 10
insulation resistance	MΩ	≥ 2000
operating temperature range	℃	-10 ~ +55
compensated temperature range	℃	-10 ~ +40
temperature coefficient of SPAN	%FS/10℃	± 0.05
temperature coefficient of ZERO	%FS/10℃	± 0.05
Electrical connection	cable	4 color wire (standard)or 4 shielded PVC cable, Ø0.8 × 220 mm

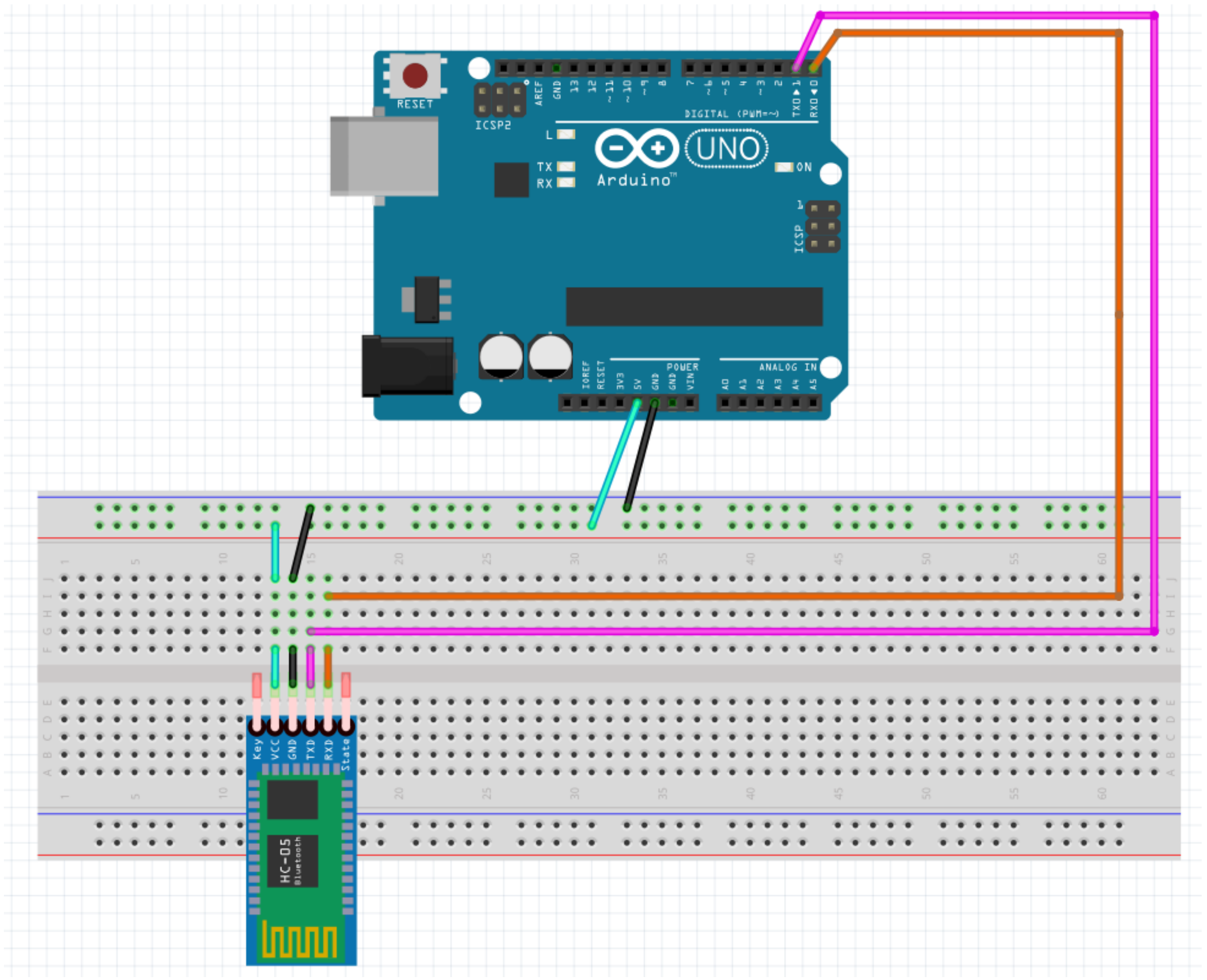
※**Ordering code:** model-capacity- rated output-accuracy-defend grade- the length of cable

Appendix B: Electrical Circuits

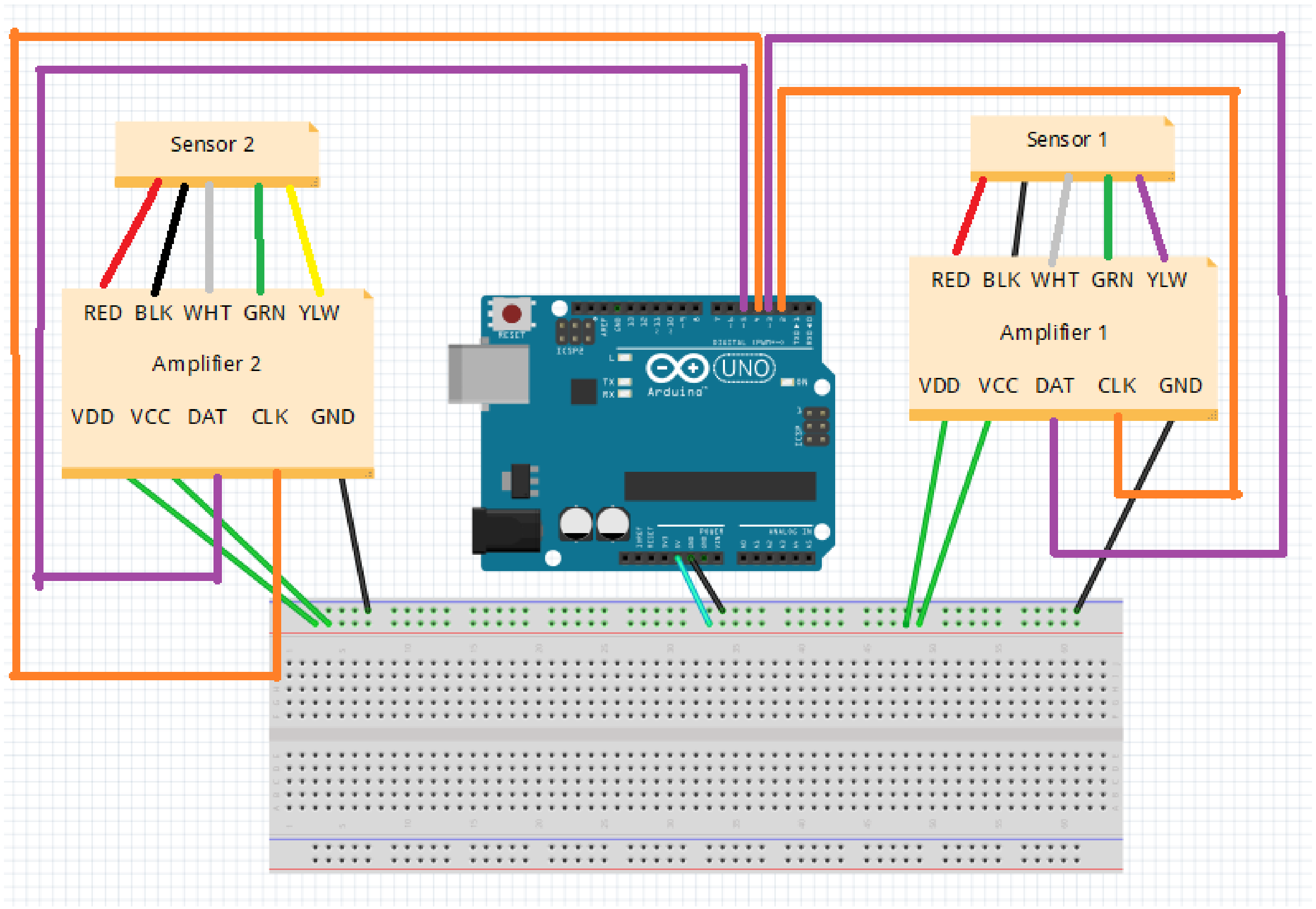
Complete Circuit Diagram



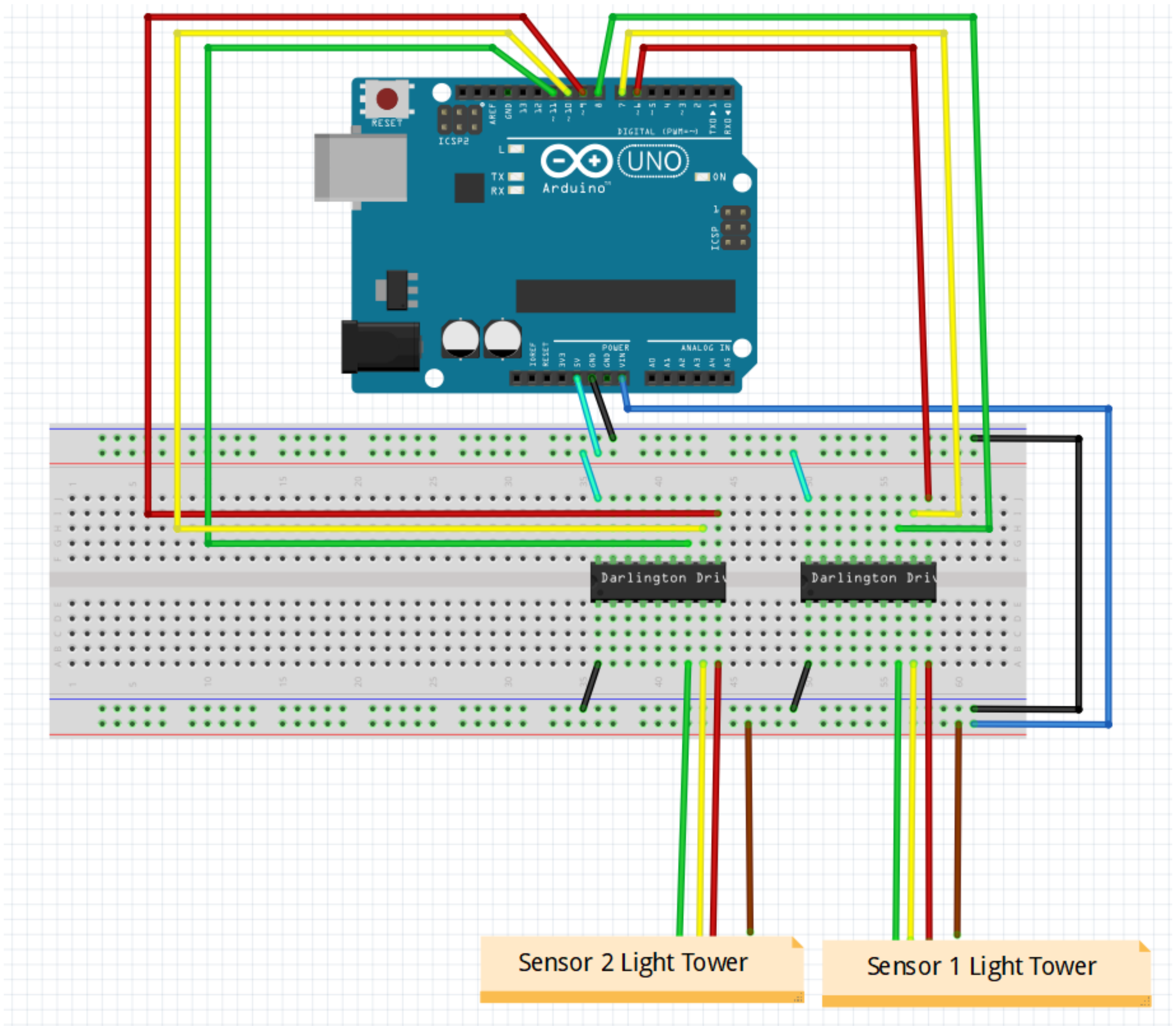
HC-06 Bluetooth module



Straight bar load cells and HX711 load cell amplifiers

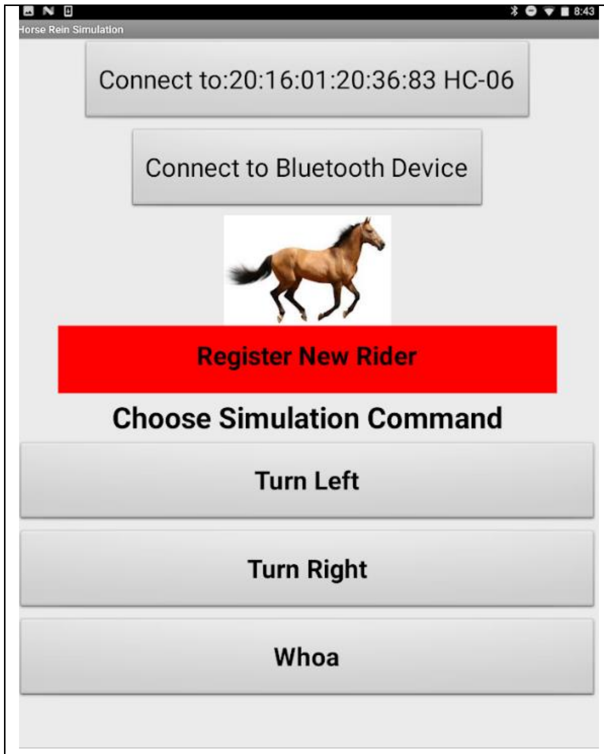


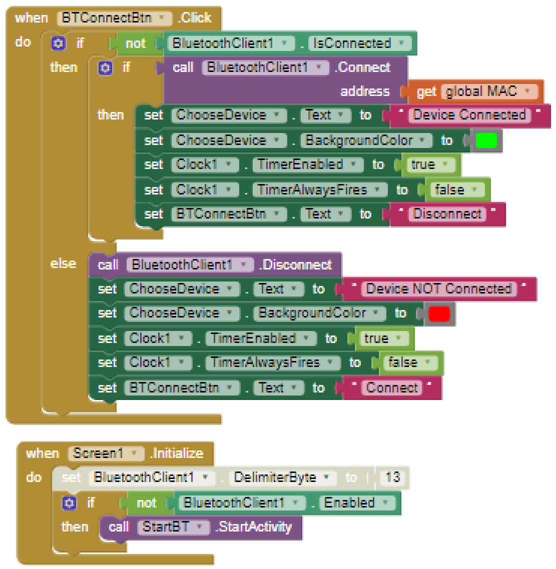
<u>Feedback Tower Lights</u>



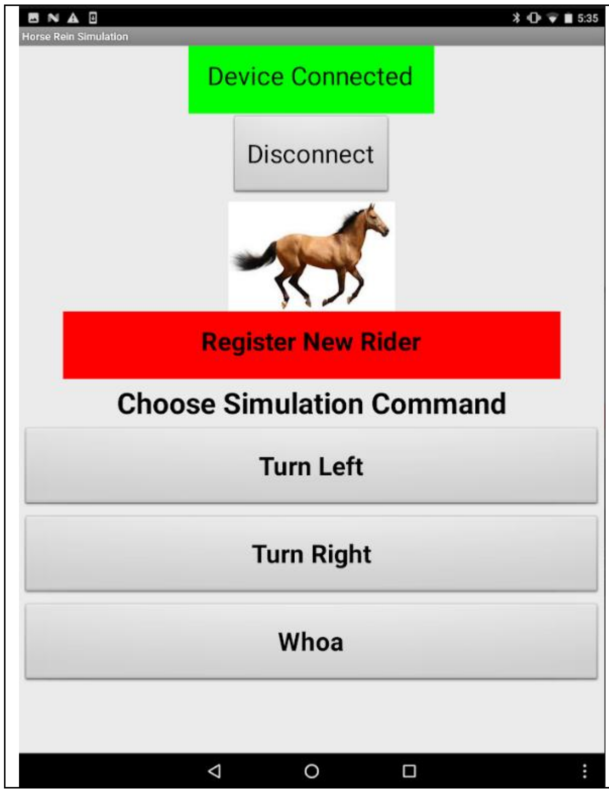
Appendix C: Code Blocks

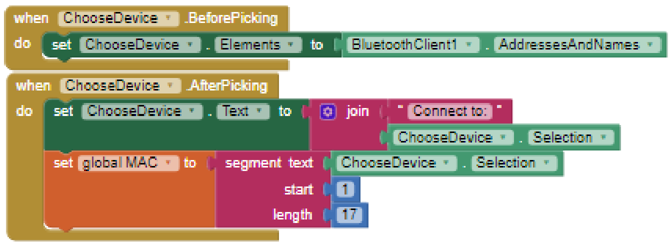
Bluetooth Code Blocks:


8:43
Horse Rein Simulation
Connect to:20:16:01:20:36:83 HC-06
Connect to Bluetooth Device
Register New Rider
Choose Simulation Command
Turn Left
Turn Right
Whoa



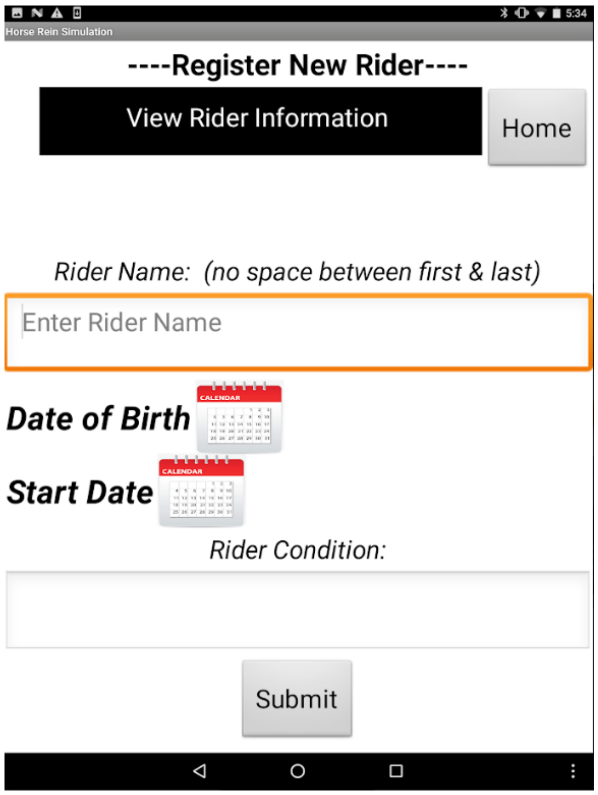
```
when BTConnectBtn .Click
do  if not BluetoothClient1 . IsConnected
    then  if call BluetoothClient1 .Connect
                 address get global MAC
          then  set ChooseDevice . Text to " Device Connected "
                set ChooseDevice . BackgroundColor to 🟩
                set Clock1 . TimerEnabled to true
                set Clock1 . TimerAlwaysFires to false
                set BTConnectBtn . Text to " Disconnect "
    else  call BluetoothClient1 .Disconnect
          set ChooseDevice . Text to " Device NOT Connected "
          set ChooseDevice . BackgroundColor to 🟥
          set Clock1 . TimerEnabled to true
          set Clock1 . TimerAlwaysFires to false
          set BTConnectBtn . Text to " Connect "

when Screen1 .Initialize
do  set BluetoothClient1 . DelimiterByte to 13
    if not BluetoothClient1 . Enabled
    then  call StartBT .StartActivity
```





```
when ChooseDevice .BeforePicking
do  set ChooseDevice . Elements to  BluetoothClient1 . AddressesAndNames

when ChooseDevice .AfterPicking
do  set ChooseDevice . Text to  join  " Connect to: "
                                      ChooseDevice . Selection
    set global MAC to  segment text  ChooseDevice . Selection
                               start  1
                              length  17
```

Rider Registration Code Blocks:



Horse Rein Simulation

----Register New Rider----

View Rider Information

Home

Rider Name: (no space between first & last)

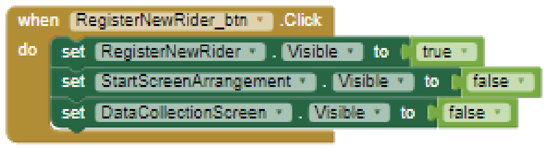
Enter Rider Name

Date of Birth

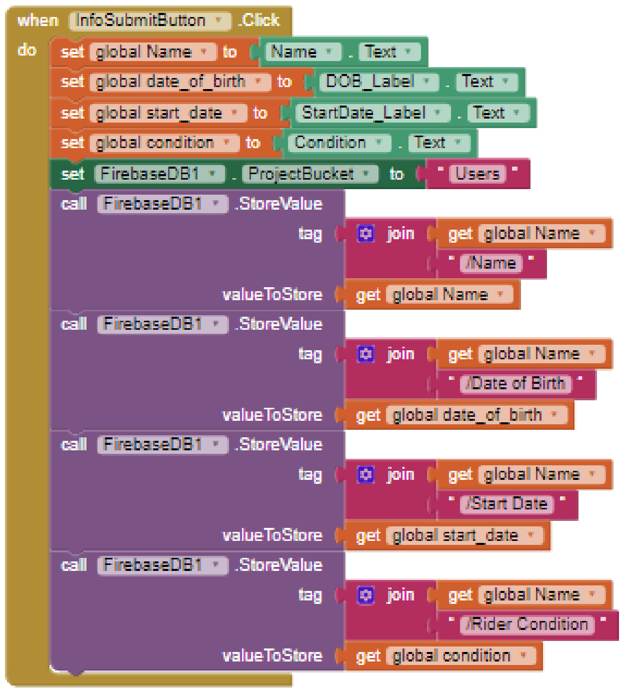
Start Date

Rider Condition:

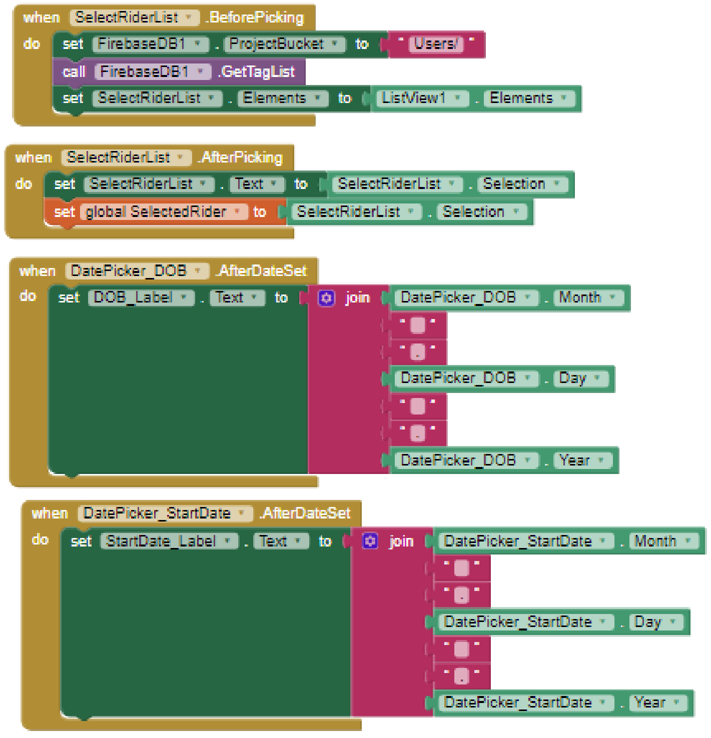
Submit



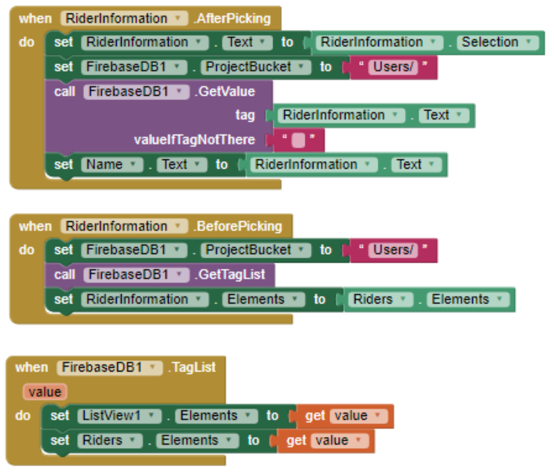
```
when RegisterNewRider_btn .Click
do  set RegisterNewRider . Visible to true
    set StartScreenArrangement . Visible to false
    set DataCollectionScreen . Visible to false
```



```
when InfoSubmitButton .Click
do  set global Name to Name . Text
    set global date_of_birth to DOB_Label . Text
    set global start_date to StartDate_Label . Text
    set global condition to Condition . Text
    set FirebaseDB1 . ProjectBucket to " Users "
    call FirebaseDB1 .StoreValue
                          tag  join  get global Name
                                     " /Name "
                 valueToStore  get global Name
    call FirebaseDB1 .StoreValue
                          tag  join  get global Name
                                     " /Date of Birth "
                 valueToStore  get global date_of_birth
    call FirebaseDB1 .StoreValue
                          tag  join  get global Name
                                     " /Start Date "
                 valueToStore  get global start_date
    call FirebaseDB1 .StoreValue
                          tag  join  get global Name
                                     " /Rider Condition "
                 valueToStore  get global condition
```

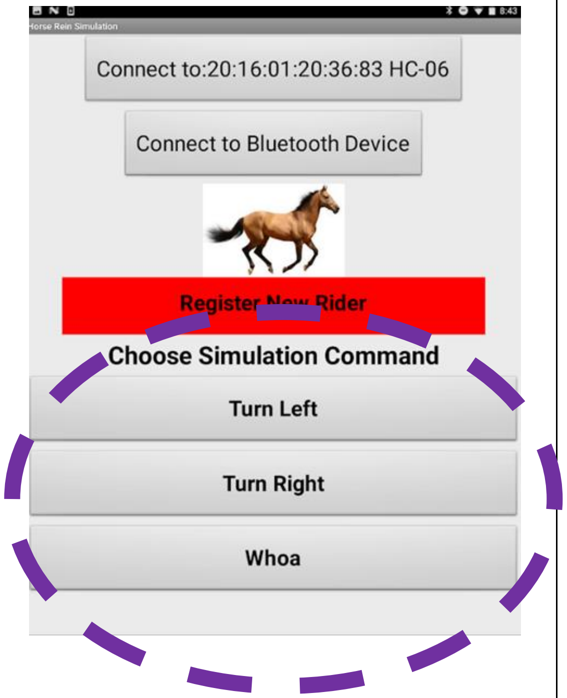


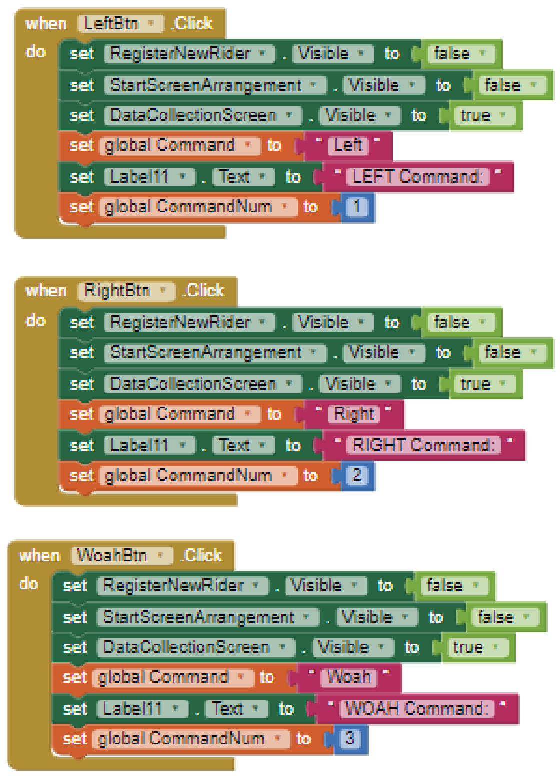
```
when SelectRiderList .BeforePicking
do  set FirebaseDB1 . ProjectBucket to " Users/ "
    call FirebaseDB1 .GetTagList
    set SelectRiderList . Elements to ListView1 . Elements

when SelectRiderList .AfterPicking
do  set SelectRiderList . Text to SelectRiderList . Selection
    set global SelectedRider to SelectRiderList . Selection

when DatePicker_DOB .AfterDateSet
do  set DOB_Label . Text to  join  DatePicker_DOB . Month
                                   " "
                                   " . "
                                   DatePicker_DOB . Day
                                   " "
                                   " . "
                                   DatePicker_DOB . Year

when DatePicker_StartDate .AfterDateSet
do  set StartDate_Label . Text to  join  DatePicker_StartDate . Month
                                         " "
                                         " . "
                                         DatePicker_StartDate . Day
                                         " "
                                         " . "
                                         DatePicker_StartDate . Year
```



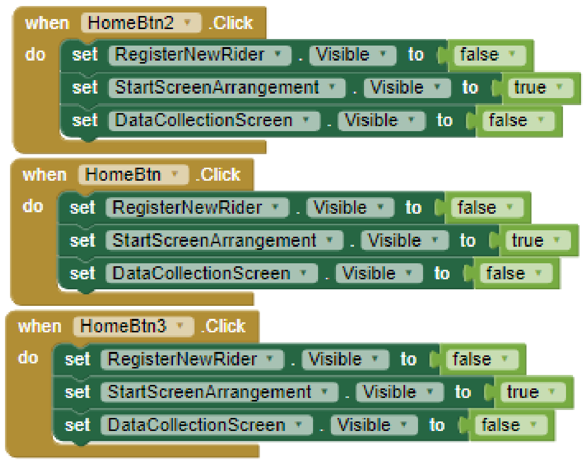
```
when RiderInformation .AfterPicking
do  set RiderInformation . Text to RiderInformation . Selection
    set FirebaseDB1 . ProjectBucket to " Users/ "
    call FirebaseDB1 .GetValue
                      tag  RiderInformation . Text
        valueIfTagNotThere  " "
    set Name . Text to RiderInformation . Text

when RiderInformation .BeforePicking
do  set FirebaseDB1 . ProjectBucket to " Users/ "
    call FirebaseDB1 .GetTagList
    set RiderInformation . Elements to Riders . Elements

when FirebaseDB1 .TagList
 value
do  set ListView1 . Elements to get value
    set Riders . Elements to get value
```

Command Buttons Code Blocks:


Horse Rein Simulation
Connect to:20:16:01:20:36:83 HC-06
Connect to Bluetooth Device
Register New Rider
Choose Simulation Command
Turn Left
Turn Right
Whoa

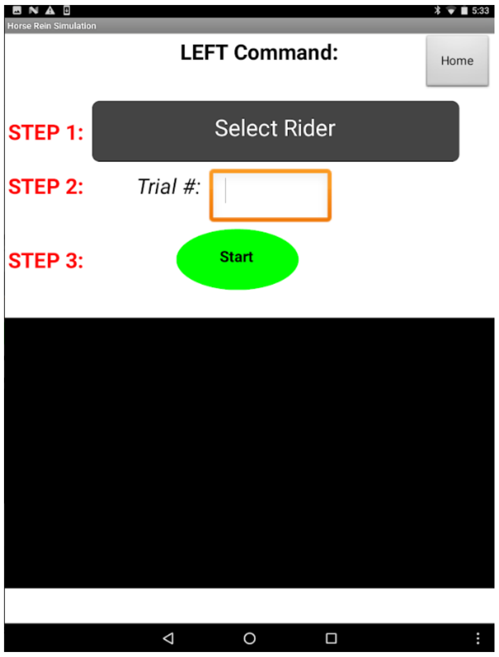
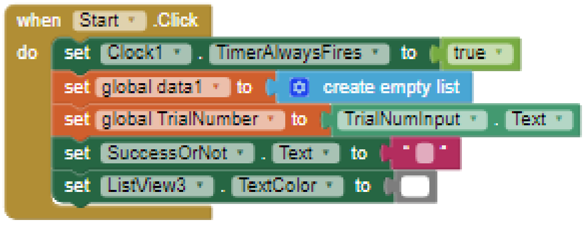


```
when LeftBtn .Click
do  set RegisterNewRider . Visible to false
    set StartScreenArrangement . Visible to false
    set DataCollectionScreen . Visible to true
    set global Command to " Left "
    set Label11 . Text to " LEFT Command: "
    set global CommandNum to 1

when RightBtn .Click
do  set RegisterNewRider . Visible to false
    set StartScreenArrangement . Visible to false
    set DataCollectionScreen . Visible to true
    set global Command to " Right "
    set Label11 . Text to " RIGHT Command: "
    set global CommandNum to 2

when WoahBtn .Click
do  set RegisterNewRider . Visible to false
    set StartScreenArrangement . Visible to false
    set DataCollectionScreen . Visible to true
    set global Command to " Woah "
    set Label11 . Text to " WOAH Command: "
    set global CommandNum to 3
```

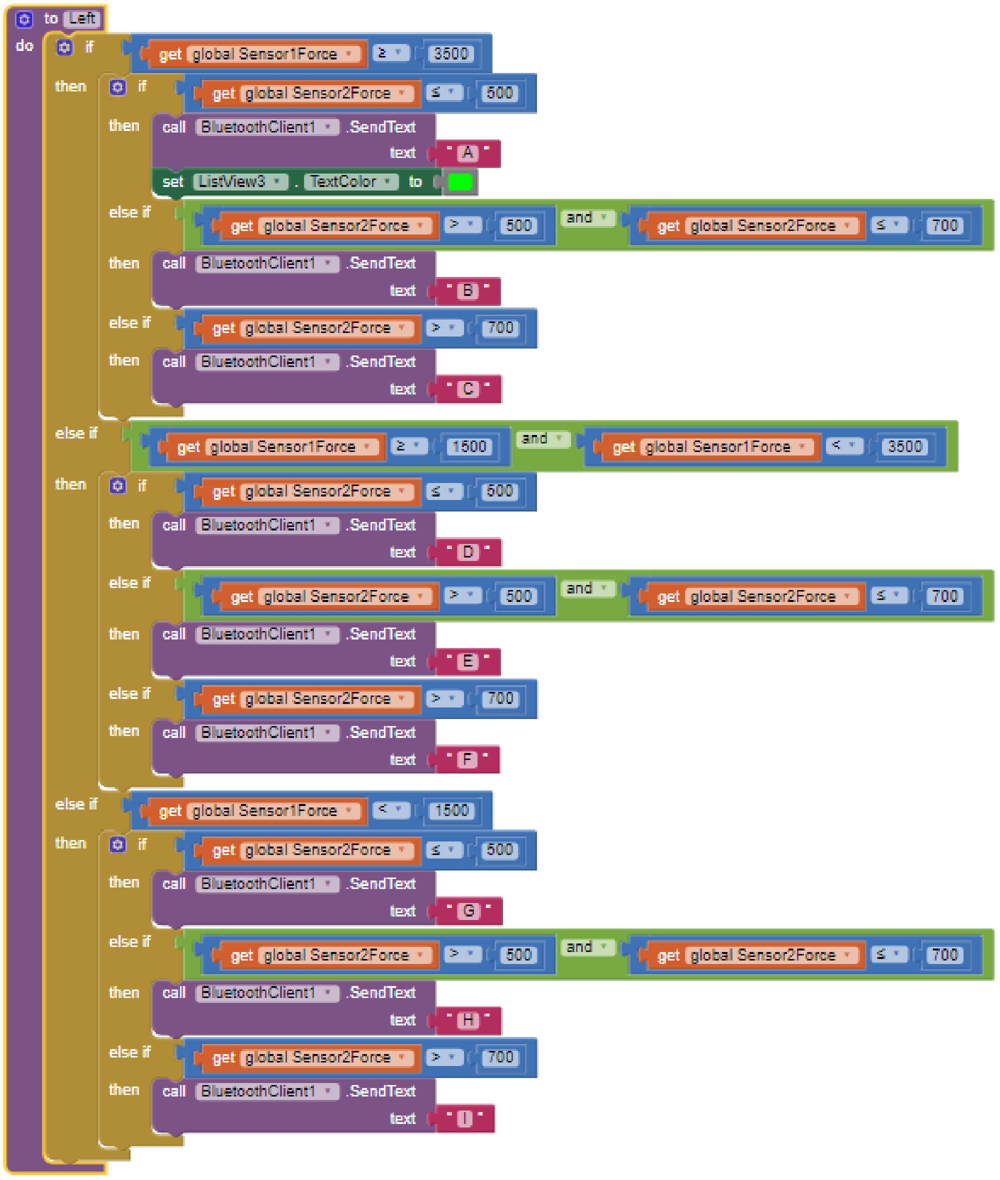


```
when HomeBtn2 .Click
do  set RegisterNewRider . Visible to false
    set StartScreenArrangement . Visible to true
    set DataCollectionScreen . Visible to false

when HomeBtn .Click
do  set RegisterNewRider . Visible to false
    set StartScreenArrangement . Visible to true
    set DataCollectionScreen . Visible to false

when HomeBtn3 .Click
do  set RegisterNewRider . Visible to false
    set StartScreenArrangement . Visible to true
    set DataCollectionScreen . Visible to false
```

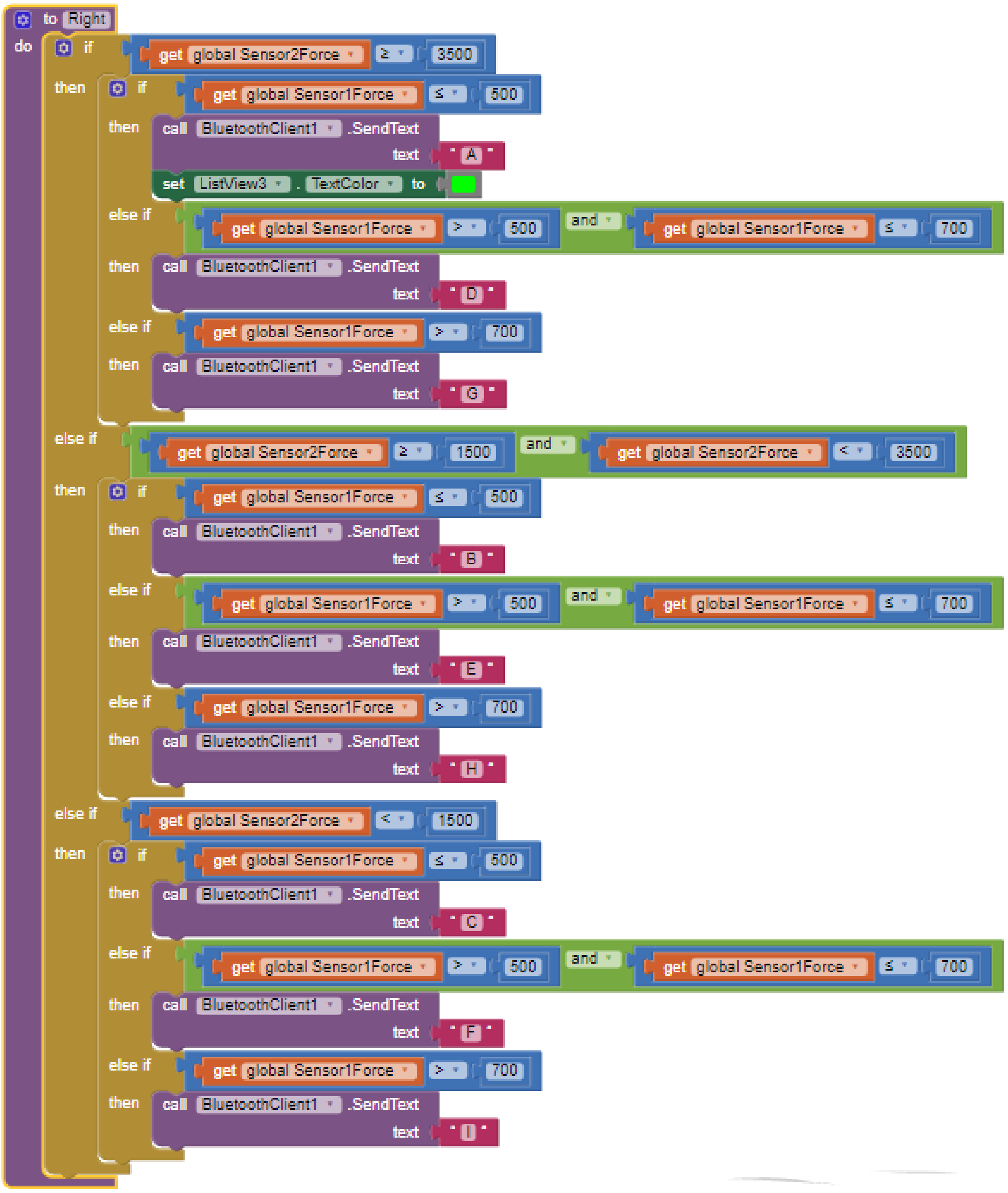
Testing/ Trial Screen Code Blocks:

 Horse Rein Simulation LEFT Command: Home STEP 1: Select Rider STEP 2: Trial #: STEP 3: Start	 when Start .Click do set Clock1 . TimerAlwaysFires to true set global data1 to create empty list set global TrialNumber to TrialNumInput . Text set SuccessOrNot . Text to " " set ListView3 . TextColor to **refer to the block of code utilizing the devices internal clock (*when Clock1.Timer)* for the rest of the code

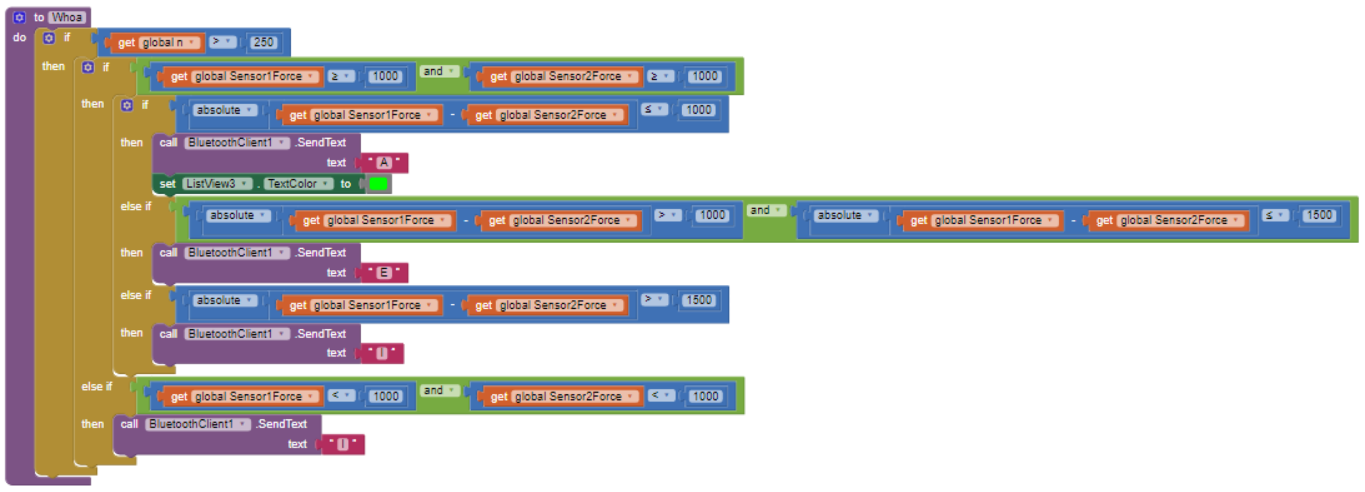
Feedback Byte Conditions Code Blocks



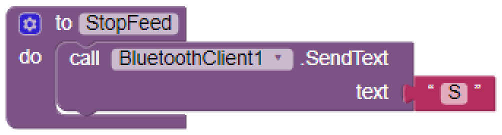
```
to Left
do  if  get global Sensor1Force ≥ 3500
    then  if  get global Sensor2Force ≤ 500
          then  call BluetoothClient1 .SendText
                                          text " A "
                set ListView3 . TextColor to 🟩
          else if  get global Sensor2Force > 500 and get global Sensor2Force ≤ 700
          then  call BluetoothClient1 .SendText
                                          text " B "
          else if  get global Sensor2Force > 700
          then  call BluetoothClient1 .SendText
                                          text " C "
    else if  get global Sensor1Force ≥ 1500 and get global Sensor1Force < 3500
    then  if  get global Sensor2Force ≤ 500
          then  call BluetoothClient1 .SendText
                                          text " D "
          else if  get global Sensor2Force > 500 and get global Sensor2Force ≤ 700
          then  call BluetoothClient1 .SendText
                                          text " E "
          else if  get global Sensor2Force > 700
          then  call BluetoothClient1 .SendText
                                          text " F "
    else if  get global Sensor1Force < 1500
    then  if  get global Sensor2Force ≤ 500
          then  call BluetoothClient1 .SendText
                                          text " G "
          else if  get global Sensor2Force > 500 and get global Sensor2Force ≤ 700
          then  call BluetoothClient1 .SendText
                                          text " H "
          else if  get global Sensor2Force > 700
          then  call BluetoothClient1 .SendText
                                          text " I "
```



```
to Right
do  if      get global Sensor2Force ≥ 3500
    then    if      get global Sensor1Force ≤ 500
            then    call BluetoothClient1 .SendText
                                          text " A "
                    set ListView3 . TextColor to 🟩
            else if get global Sensor1Force > 500 and get global Sensor1Force ≤ 700
            then    call BluetoothClient1 .SendText
                                          text " D "
            else if get global Sensor1Force > 700
            then    call BluetoothClient1 .SendText
                                          text " G "
    else if get global Sensor2Force ≥ 1500 and get global Sensor2Force < 3500
    then    if      get global Sensor1Force ≤ 500
            then    call BluetoothClient1 .SendText
                                          text " B "
            else if get global Sensor1Force > 500 and get global Sensor1Force ≤ 700
            then    call BluetoothClient1 .SendText
                                          text " E "
            else if get global Sensor1Force > 700
            then    call BluetoothClient1 .SendText
                                          text " H "
    else if get global Sensor2Force < 1500
    then    if      get global Sensor1Force ≤ 500
            then    call BluetoothClient1 .SendText
                                          text " C "
            else if get global Sensor1Force > 500 and get global Sensor1Force ≤ 700
            then    call BluetoothClient1 .SendText
                                          text " F "
            else if get global Sensor1Force > 700
            then    call BluetoothClient1 .SendText
                                          text " I "
```

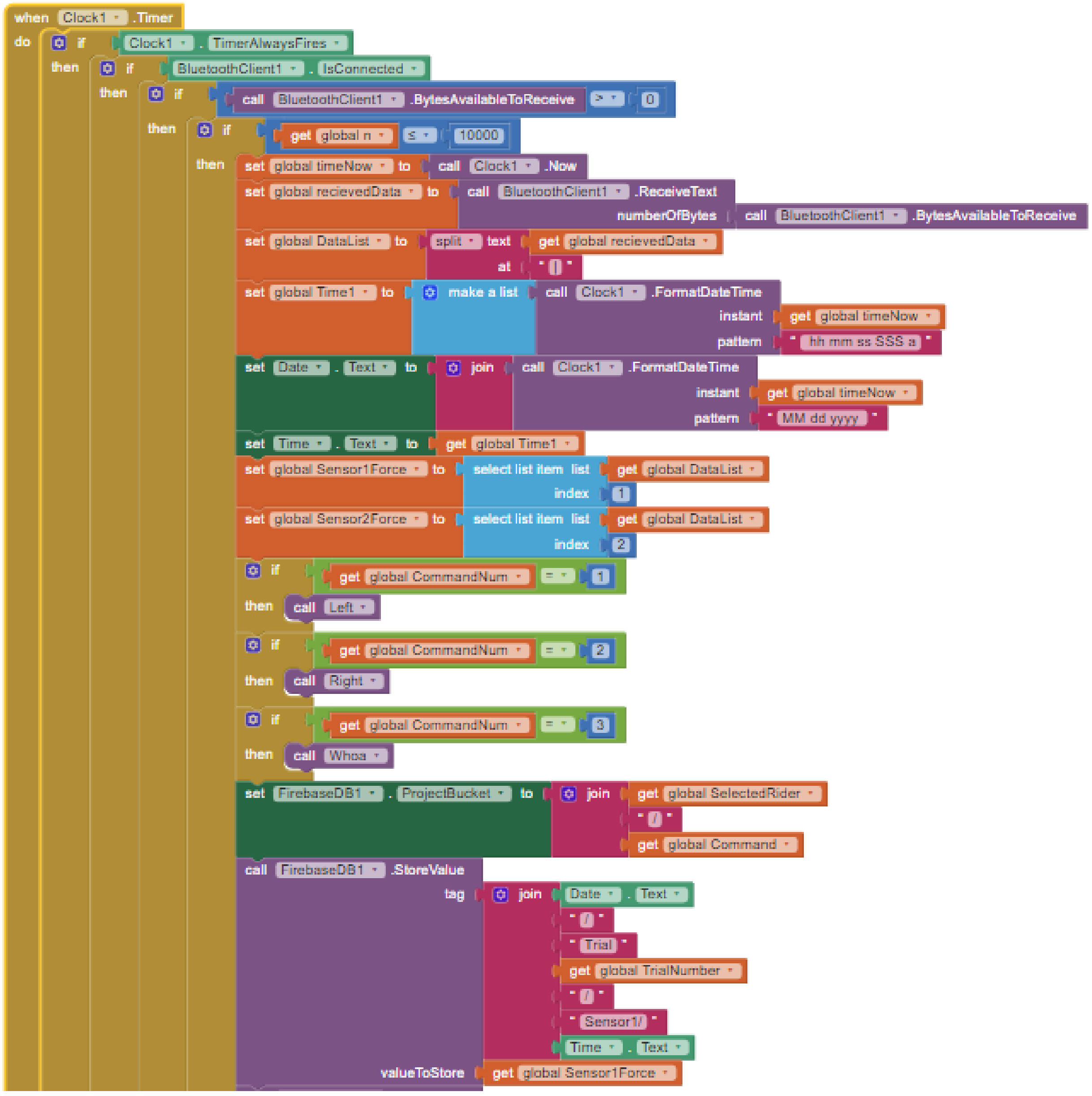


```
to Whoa
do  if    get global n  >  250
    then  if    get global Sensor1Force  ≥  1000  and  get global Sensor2Force  ≥  1000
          then  if    absolute  get global Sensor1Force  -  get global Sensor2Force  ≤  1000
                then  call BluetoothClient1 .SendText
                                              text  " A "
                      set ListView3 . TextColor to  🟩
                else if  absolute  get global Sensor1Force  -  get global Sensor2Force  >  1000  and  absolute  get global Sensor1Force  -  get global Sensor2Force  ≤  1500
                then  call BluetoothClient1 .SendText
                                              text  " E "
                else if  absolute  get global Sensor1Force  -  get global Sensor2Force  >  1500
                then  call BluetoothClient1 .SendText
                                              text  " I "
          else if  get global Sensor1Force  <  1000  and  get global Sensor2Force  <  1000
          then  call BluetoothClient1 .SendText
                                        text  " I "
```



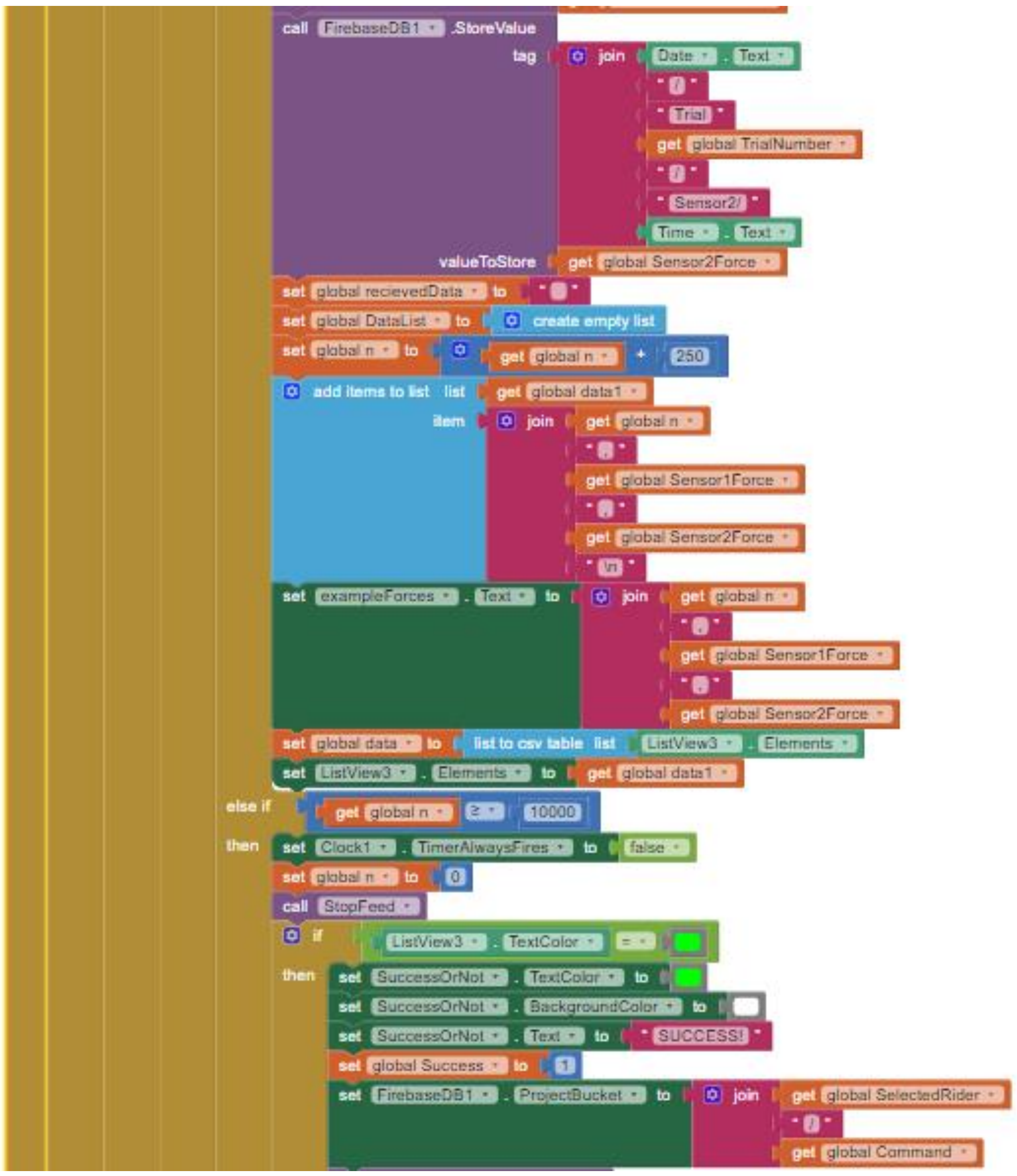
```
to StopFeed
do  call BluetoothClient1 .SendText
                            text  " S "
```

Internal Clock Code Blocks (1/3)



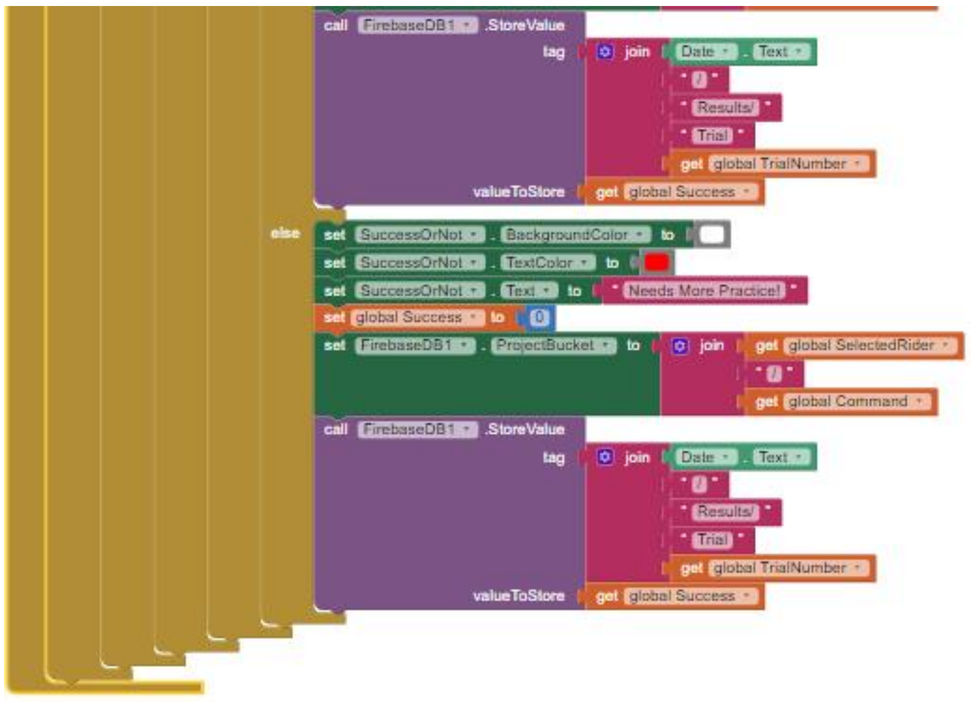
```
when Clock1 .Timer
do  if  Clock1 . TimerAlwaysFires
    then  if  BluetoothClient1 . IsConnected
          then  if  call BluetoothClient1 .BytesAvailableToReceive  >  0
                then  if  get global n  ≤  10000
                      then  set global timeNow to  call Clock1 .Now
                            set global recievedData to  call BluetoothClient1 .ReceiveText
                                                             numberOfBytes  call BluetoothClient1 .BytesAvailableToReceive
                            set global DataList to  split  text  get global recievedData
                                                           at    " | "
                            set global Time1 to  make a list  call Clock1 .FormatDateTime
                                                                   instant  get global timeNow
                                                                   pattern  " hh mm ss SSS a "
                            set Date . Text to  join  call Clock1 .FormatDateTime
                                                           instant  get global timeNow
                                                           pattern  " MM dd yyyy "
                            set Time . Text to  get global Time1
                            set global Sensor1Force to  select list item  list  get global DataList
                                                                          index  1
                            set global Sensor2Force to  select list item  list  get global DataList
                                                                          index  2
                            if    get global CommandNum  =  1
                            then  call Left
                            if    get global CommandNum  =  2
                            then  call Right
                            if    get global CommandNum  =  3
                            then  call Whoa
                            set FirebaseDB1 . ProjectBucket to  join  get global SelectedRider
                                                                      " / "
                                                                      get global Command
                            call FirebaseDB1 .StoreValue
                                         tag  join  Date . Text
                                                    " / "
                                                    " Trial "
                                                    get global TrialNumber
                                                    " / "
                                                    " Sensor1/ "
                                                    Time . Text
                                valueToStore  get global Sensor1Force
```

Internal Clock Code Blocks (2/3)



```
call FirebaseDB1 .StoreValue
                    tag   join  Date . Text
                                " / "
                                " Trial "
                                get global TrialNumber
                                " / "
                                " Sensor2/ "
                                Time . Text
           valueToStore   get global Sensor2Force
set global recievedData to " "
set global DataList to  create empty list
set global n to  get global n + 250
add items to list  list  get global data1
                   item  join  get global n
                               " , "
                               get global Sensor1Force
                               " , "
                               get global Sensor2Force
                               " \n "
set exampleForces . Text to  join  get global n
                                   " , "
                                   get global Sensor1Force
                                   " , "
                                   get global Sensor2Force
set global data to  list to csv table  list  ListView3 . Elements
set ListView3 . Elements to  get global data1
else if   get global n ≥ 10000
then  set Clock1 . TimerAlwaysFires to false
      set global n to 0
      call StopFeed
      if   ListView3 . TextColor = 🟩
      then  set SuccessOrNot . TextColor to 🟩
            set SuccessOrNot . BackgroundColor to ⬜
            set SuccessOrNot . Text to " SUCCESS! "
            set global Success to 1
            set FirebaseDB1 . ProjectBucket to  join  get global SelectedRider
                                                      " / "
                                                      get global Command
```

Internal Clock Code Blocks (3/3)



```
call FirebaseDB1 .StoreValue
    tag  join  Date . Text
               " / "
               " Results/ "
               " Trial "
               get global TrialNumber
    valueToStore  get global Success
else
    set SuccessOrNot . BackgroundColor to
    set SuccessOrNot . TextColor to
    set SuccessOrNot . Text to " Needs More Practice! "
    set global Success to 0
    set FirebaseDB1 . ProjectBucket to  join  get global SelectedRider
                                              " / "
                                              get global Command
    call FirebaseDB1 .StoreValue
        tag  join  Date . Text
                   " / "
                   " Results/ "
                   " Trial "
                   get global TrialNumber
        valueToStore  get global Success
```

Appendix D: Feedback Mechanism Lighting Conditions

LEFT COMMAND				
		Left Rein Light Tower		
Right Rein Light Tower		Green	Yellow	Red
	Green	Sensor1: $f \geq 3500$: Sensor2: $f \leq 500$	Sensor1: $3500 > f \geq 1500$ Sensor2: $f \leq 500$	Sensor1: $f < 1500$ Sensor2: $f \leq 500$
	Yellow	Sensor1: $f \geq 3500$ Sensor2: $700 \geq f \geq 500$	Sensor1: $3500 > f \geq 1500$ Sensor2: $700 \geq f \geq 500$	Sensor1: $f < 1500$ Sensor2: $700 \geq f \geq 500$
	Red	Sensor1: $f \geq 3500$ Sensor2: $f > 700$	Sensor1: $3500 > f \geq 1500$ Sensor2: $f > 700$	Sensor1: $f < 1500$ Sensor2: $f > 700$

RIGHT COMMAND				
		Left Rein Light Tower		
Right Rein Light Tower		Green	Yellow	Red
	Green	Sensor1: $f \leq 500$ Sensor2: $f \geq 3500$:	Sensor1: $700 \geq f > 500$ Sensor2: $f \geq 3500$:	Sensor1: $f > 700$ Sensor2: $f \geq 3500$:
	Yellow	Sensor1: $f \leq 500$ Sensor2: $3500 > f \geq 1500$	Sensor1: $700 \geq f \geq 500$ Sensor2: $3500 > f \geq 1500$	Sensor1: $f > 700$ Sensor2: $3500 > f \geq 1500$
	Red	Sensor1: $f \leq 500$ Sensor2: $f < 1500$	Sensor1: $700 \geq f \geq 500$ Sensor2: $f < 1500$	Sensor1: $f > 700$ Sensor2: $f < 1500$

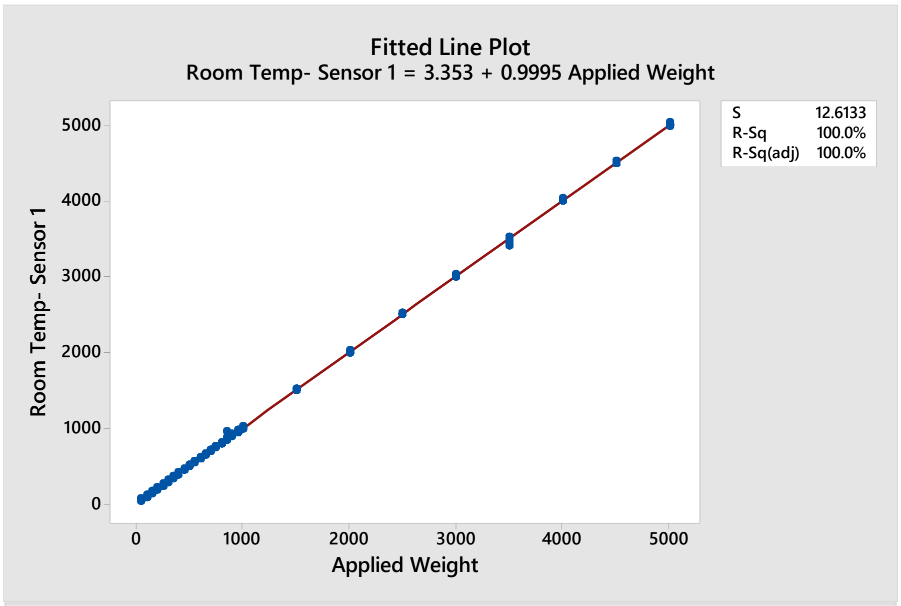
WHOA COMMAND		
All Green	**All Yellow**	**All Red**
$\boldsymbol{f > 1000}$ $\boldsymbol{diff \leq 500}$	$f > 1000$ $800 \geq diff > 500$	$f > 1000$ $diff > 800$

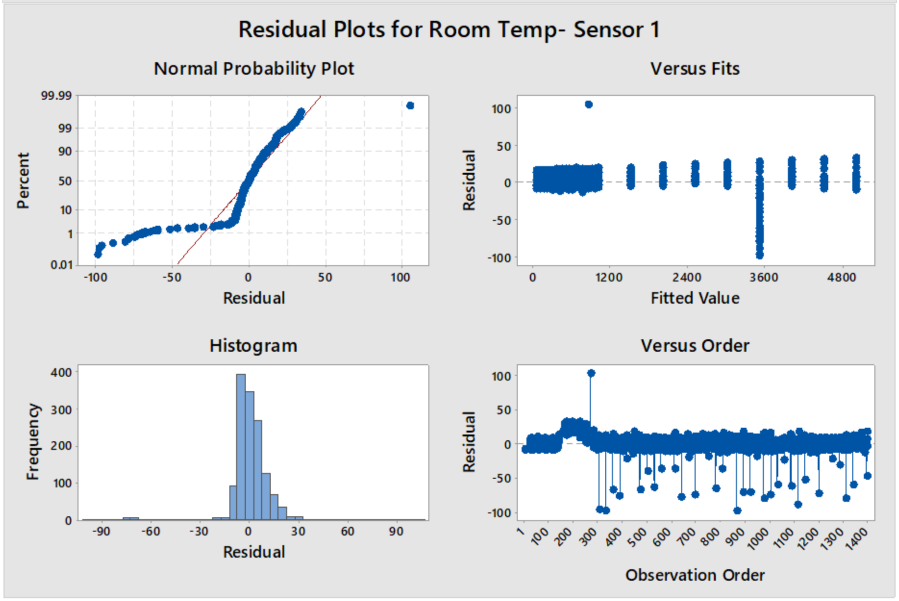
Appendix E: Statistical Figures

Minitab results

Sensor 1:

Room Temperature

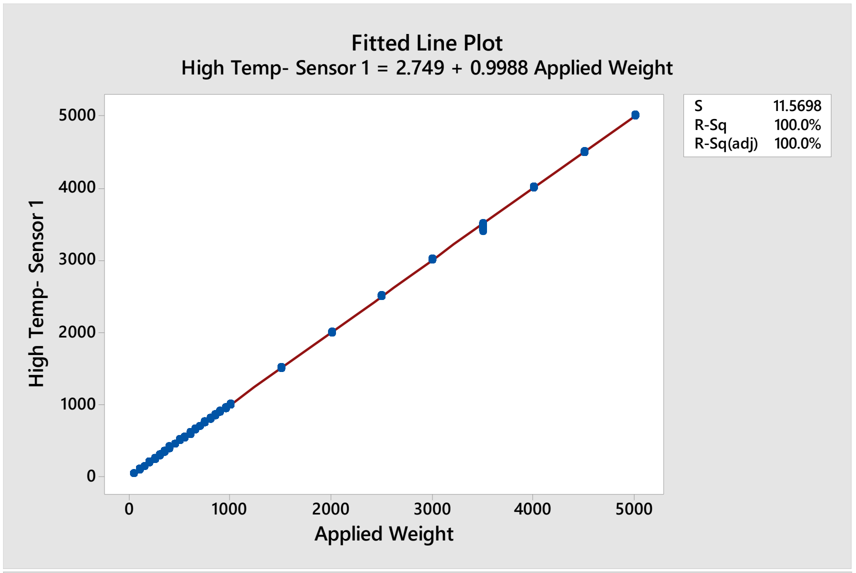


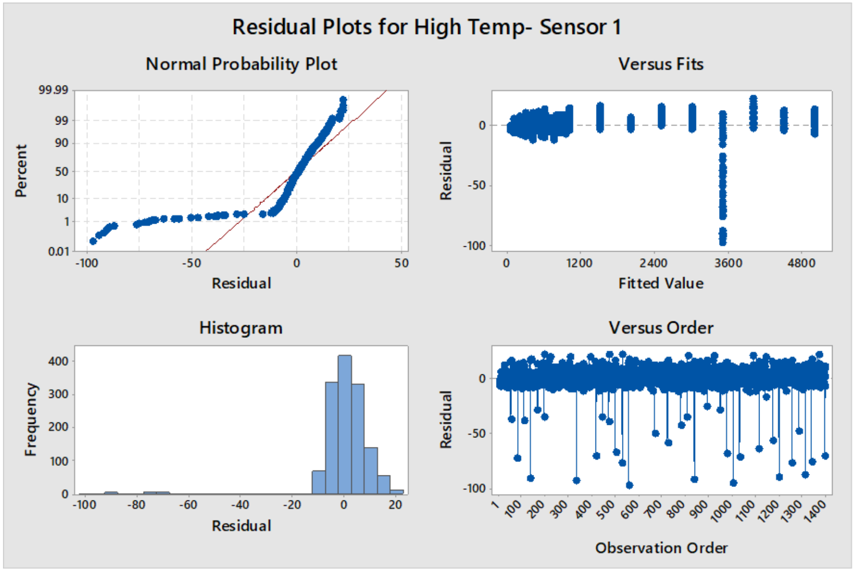


Regression Analysis: Room Temp- Sensor 1 versus Applied Weight

```
Analysis of Variance

Source              DF      Seq SS  Contribution      Adj SS      Adj MS     F-Value
Regression           1  2727190578        99.99%  2727190578  2727190578  17141992.74
  Applied Weight     1  2727190578        99.99%  2727190578  2727190578  17141992.74
Error             1398      222414         0.01%      222414         159
  Lack-of-Fit       26       78975         0.00%       78975        3037        29.05
  Pure Error      1372      143439         0.01%      143439         105
Total             1399  2727412991       100.00%

Source            P-Value
Regression          0.000
  Applied Weight    0.000
Error
  Lack-of-Fit       0.000
  Pure Error
Total


Model Summary

      S    R-sq  R-sq(adj)   PRESS  R-sq(pred)
12.6133  99.99%     99.99%  223390      99.99%


Coefficients

Term                Coef   SE Coef          95% CI          T-Value  P-Value   VIF
Constant           3.353     0.461  (   2.448,     4.257)      7.27    0.000
Applied Weight  0.999534  0.000241  (0.999061, 1.000008)   4140.29    0.000  1.00


Regression Equation

Room Temp- Sensor 1 = 3.353 + 0.999534 Applied Weight
```

High Temperature

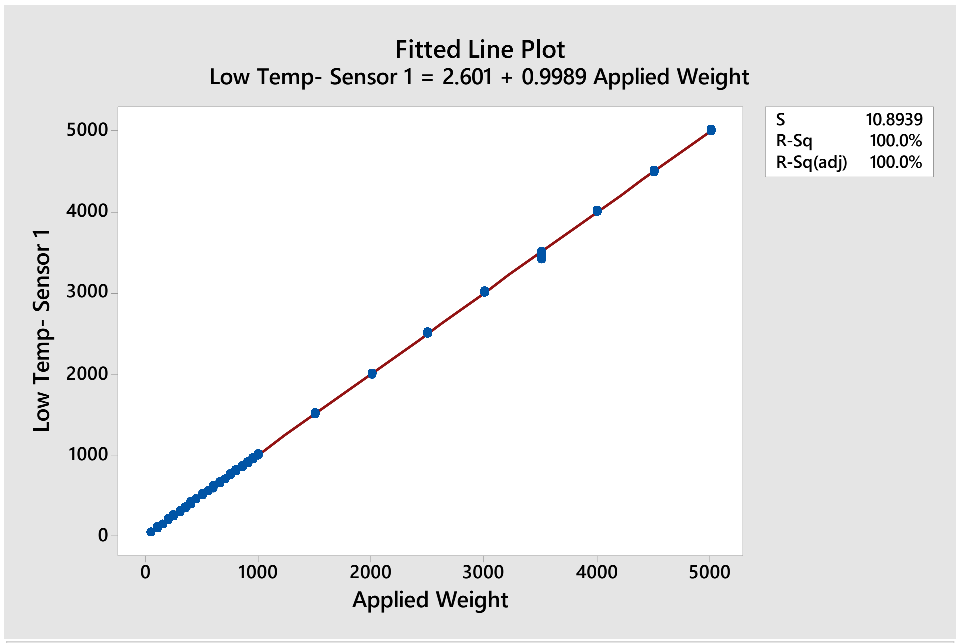


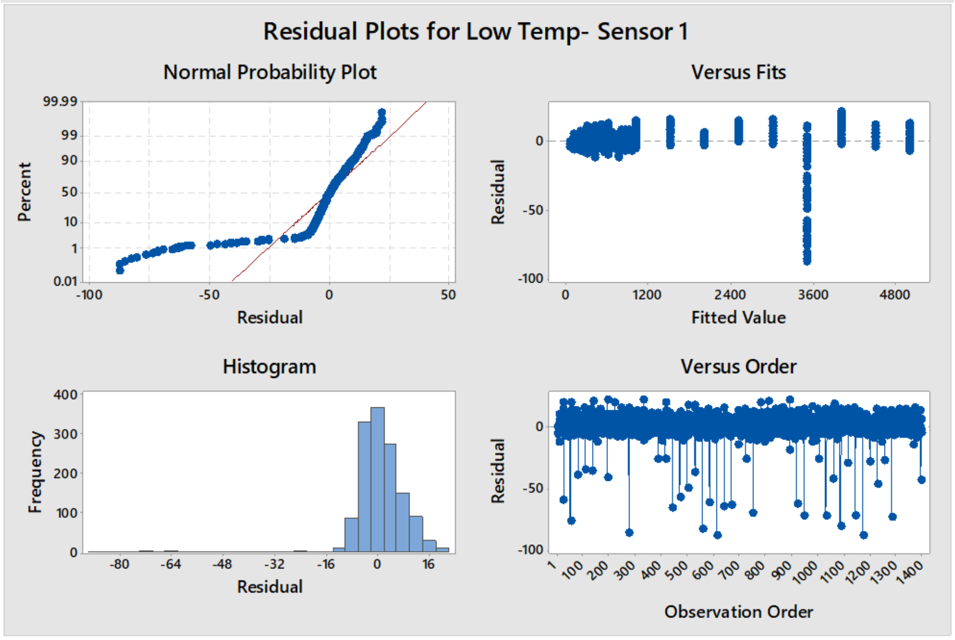


Regression Analysis: High Temp- Sensor 1 versus Applied Weight

```
Analysis of Variance

Source                DF      Seq SS  Contribution      Adj SS      Adj MS      F-Value
Regression             1  2723356613        99.99%  2723356613  2723356613  20344661.75
  Applied Weight       1  2723356613        99.99%  2723356613  2723356613  20344661.75
Error               1398      187138         0.01%      187138         134
  Lack-of-Fit         26       97903         0.00%       97903        3765        57.90
  Pure Error        1372       89235         0.00%       89235          65
Total               1399  2723543750       100.00%

Source              P-Value
Regression            0.000
  Applied Weight      0.000
Error
  Lack-of-Fit         0.000
  Pure Error
Total


Model Summary

      S    R-sq  R-sq(adj)   PRESS  R-sq(pred)
11.5698  99.99%     99.99%  187998      99.99%


Coefficients

Term                Coef   SE Coef          95% CI          T-Value  P-Value   VIF
Constant           2.749     0.423  (   1.919,     3.579)      6.50    0.000
Applied Weight  0.998832  0.000221  (0.998397, 0.999266)   4510.51    0.000  1.00


Regression Equation

High Temp- Sensor 1 = 2.749 + 0.998832 Applied Weight
```

Low Temperature



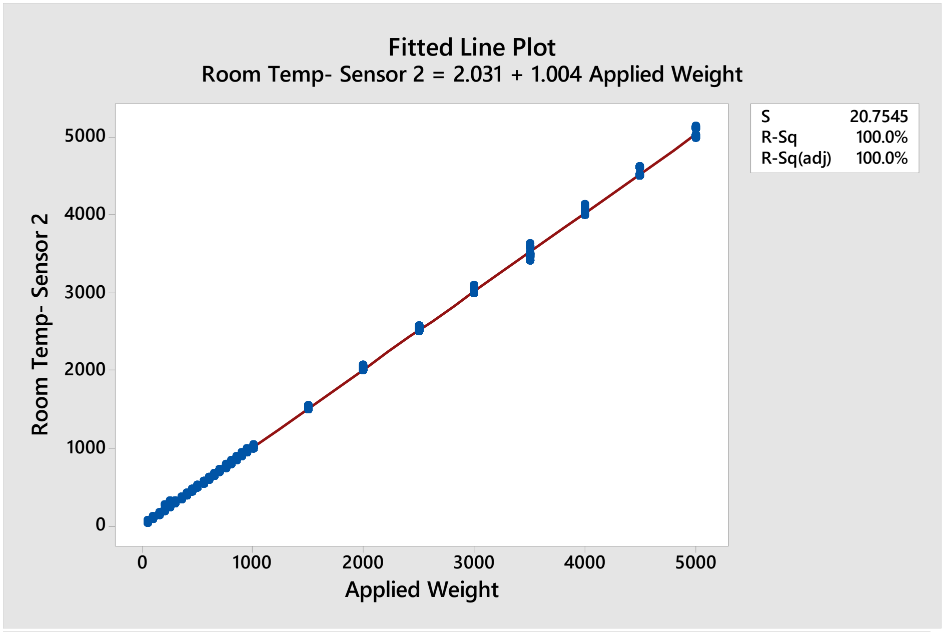


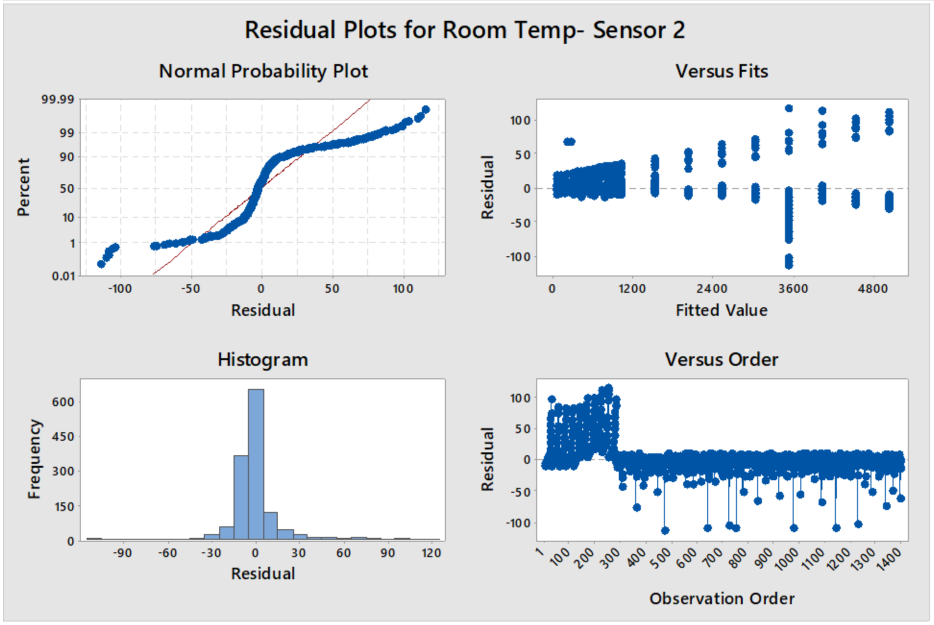
Regression Analysis: Low Temp- Sensor 1 versus Applied Weight

```
Analysis of Variance

Source               DF      Seq SS  Contribution      Adj SS      Adj MS      F-Value
Regression            1  2723723303        99.99%  2723723303  2723723303  22950719.62
  Applied Weight      1  2723723303        99.99%  2723723303  2723723303  22950719.62
Error              1398      165910         0.01%      165910         119
  Lack-of-Fit        26       93120         0.00%       93120        3582        67.51
  Pure Error       1372       72790         0.00%       72790          53
Total              1399  2723889214       100.00%

Source             P-Value
Regression           0.000
  Applied Weight     0.000
Error
  Lack-of-Fit        0.000
  Pure Error
Total


Model Summary

      S    R-sq  R-sq(adj)   PRESS  R-sq(pred)
10.8939  99.99%     99.99%  166676      99.99%


Coefficients

Term                 Coef   SE Coef          95% CI          T-Value  P-Value   VIF
Constant            2.601     0.398  (   1.819,     3.382)      6.53    0.000
Applied Weight   0.998899  0.000209  (0.998490, 0.999308)    4790.69    0.000  1.00


Regression Equation

Low Temp- Sensor 1 = 2.601 + 0.998899 Applied Weight
```

Sensor 2:

Room Temperature

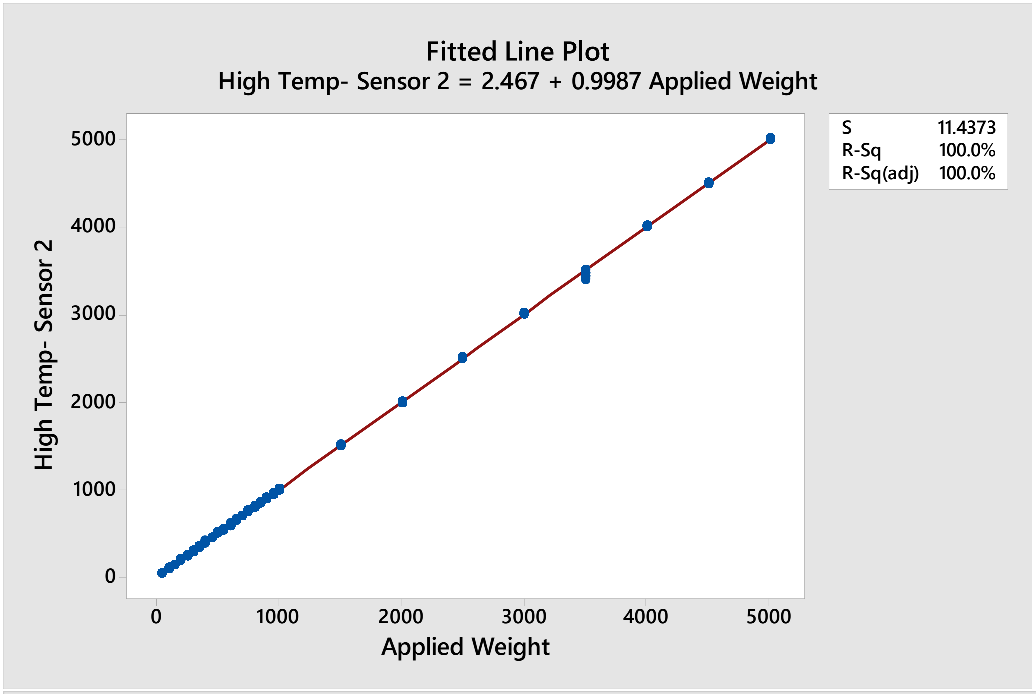


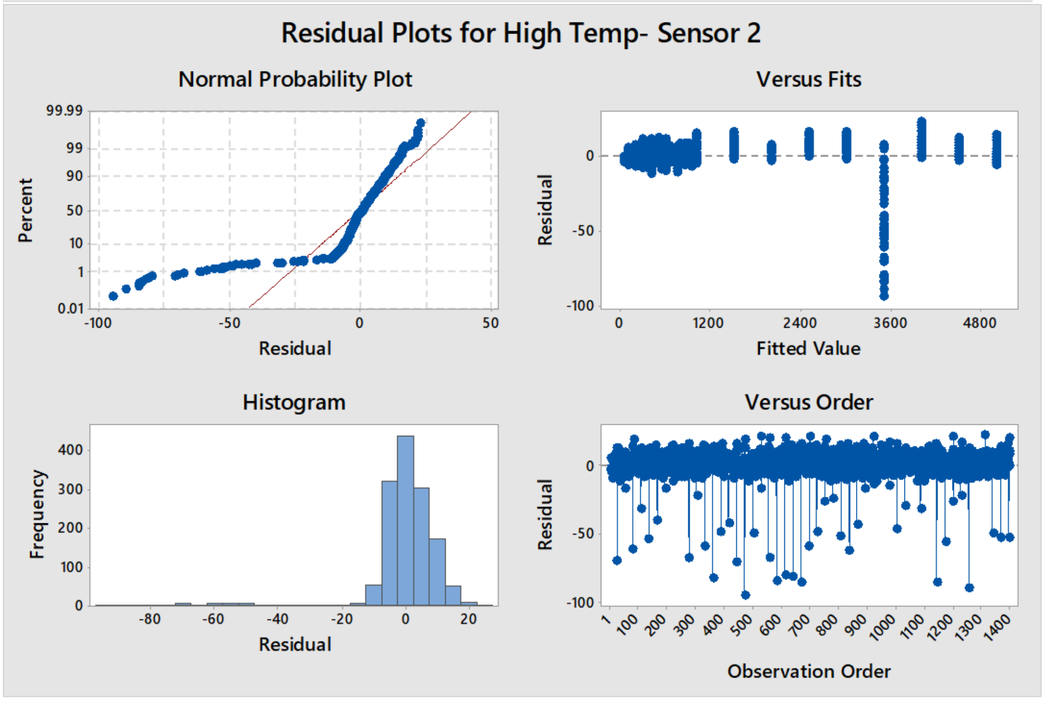


Regression Analysis: Room Temp- Sensor 2 versus Applied Weight

```
Analysis of Variance

Source              DF      Seq SS  Contribution      Adj SS      Adj MS     F-Value  P-Value
Regression           1  2750530237        99.98%  2750530237  2750530237  6385446.69    0.000
  Applied Weight     1  2750530237        99.98%  2750530237  2750530237  6385446.69    0.000
Error             1398      602188         0.02%      602188         431
  Lack-of-Fit       26       47779         0.00%       47779        1838        4.55    0.000
  Pure Error      1372      554409         0.02%      554409         404
Total             1399  2751132425       100.00%


Model Summary

      S    R-sq  R-sq(adj)   PRESS  R-sq(pred)
20.7545  99.98%     99.98%  605945      99.98%


Coefficients

Term               Coef  SE Coef        95% CI         T-Value  P-Value   VIF
Constant          2.031    0.759  (  0.542,   3.519)      2.68    0.008
Applied Weight  1.00380  0.00040  (1.00302, 1.00458)   2526.94    0.000  1.00


Regression Equation

Room Temp- Sensor 2 = 2.031 + 1.00380 Applied Weight
```

High Temperature

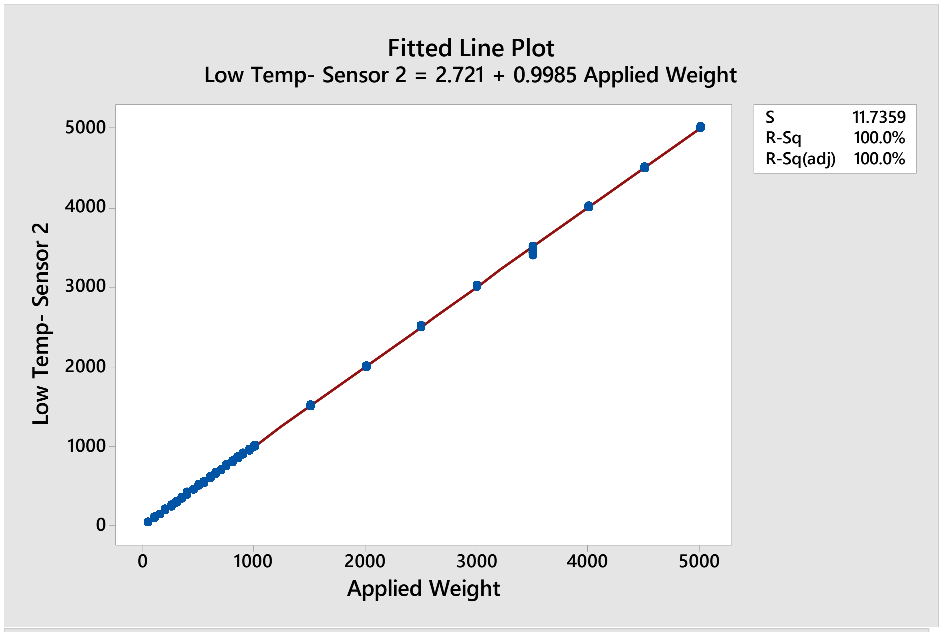


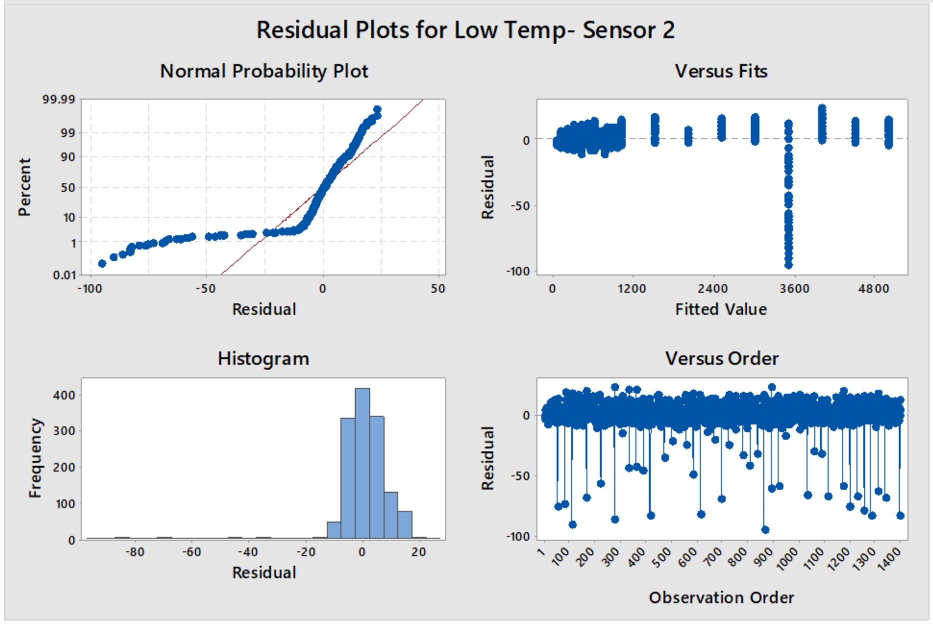


Regression Analysis: High Temp- Sensor 2 versus Applied Weight

```
Analysis of Variance

Source              DF      Seq SS  Contribution      Adj SS      Adj MS      F-Value
Regression           1  2722696817        99.99%  2722696817  2722696817  20813691.58
  Applied Weight     1  2722696817        99.99%  2722696817  2722696817  20813691.58
Error             1398      182876         0.01%      182876         131
  Lack-of-Fit       26      113329         0.00%      113329        4359        85.99
  Pure Error      1372       69548         0.00%       69548          51
Total             1399  2722879694       100.00%

Source            P-Value
Regression          0.000
  Applied Weight    0.000
Error
  Lack-of-Fit       0.000
  Pure Error
Total


Model Summary

      S    R-sq  R-sq(adj)   PRESS  R-sq(pred)
11.4373  99.99%     99.99%  183721      99.99%


Coefficients

Term                Coef   SE Coef          95% CI          T-Value  P-Value   VIF
Constant           2.467     0.418  (   1.647,     3.288)      5.90    0.000
Applied Weight  0.998711  0.000219  (0.998281, 0.999140)   4562.20    0.000  1.00


Regression Equation

High Temp- Sensor 2 = 2.467 + 0.998711 Applied Weight
```

Low Temperature



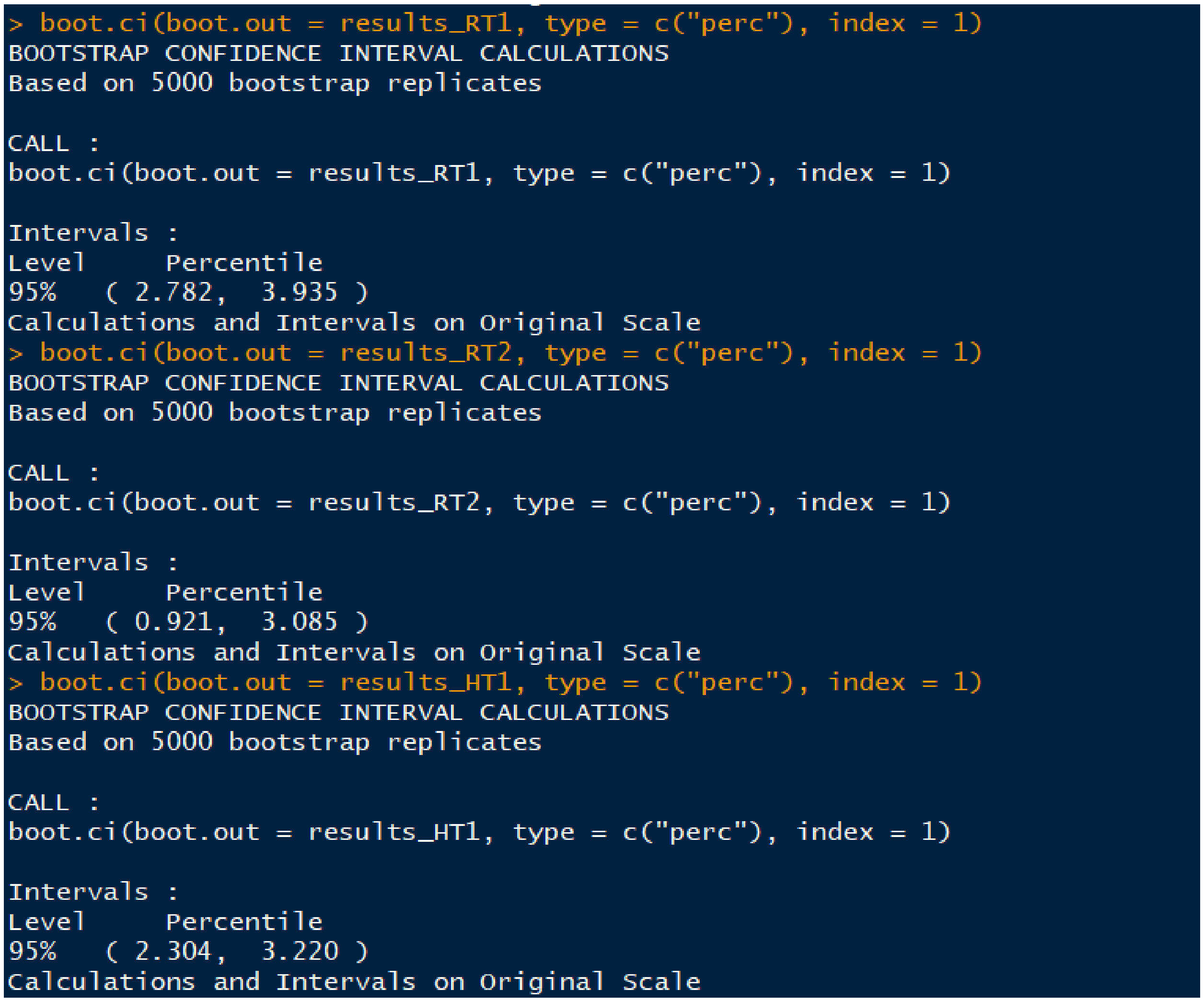


Regression Analysis: Room Temp- Sensor 2 versus Applied Weight

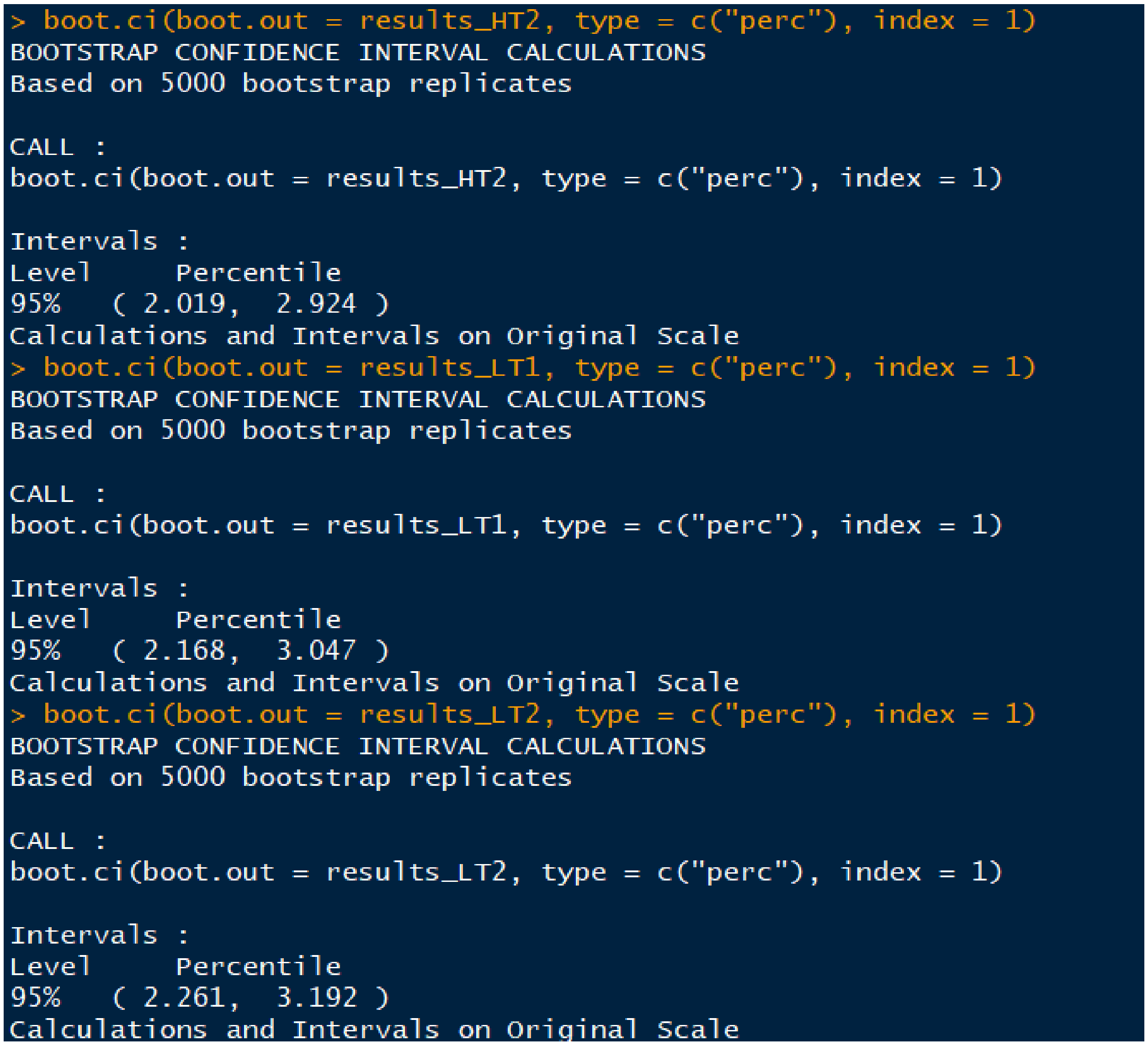
```
Analysis of Variance

Source             DF      Seq SS  Contribution      Adj SS      Adj MS     F-Value  P-Value
Regression          1  2750530237        99.98%  2750530237  2750530237  6385446.69    0.000
  Applied Weight    1  2750530237        99.98%  2750530237  2750530237  6385446.69    0.000
Error            1398      602188         0.02%      602188         431
  Lack-of-Fit      26       47779         0.00%       47779        1838        4.55    0.000
  Pure Error     1372      554409         0.02%      554409         404
Total            1399  2751132425       100.00%


Model Summary

      S    R-sq  R-sq(adj)   PRESS  R-sq(pred)
20.7545  99.98%     99.98%  605945      99.98%


Coefficients

Term               Coef  SE Coef        95% CI          T-Value  P-Value   VIF
Constant          2.031    0.759  (  0.542,    3.519)     2.68    0.008
Applied Weight  1.00380  0.00040  (1.00302, 1.00458)  2526.94    0.000  1.00


Regression Equation

Room Temp- Sensor 2 = 2.031 + 1.00380 Applied Weight
```

RStudio Output

Intercept Confidence Intervals After Resampling

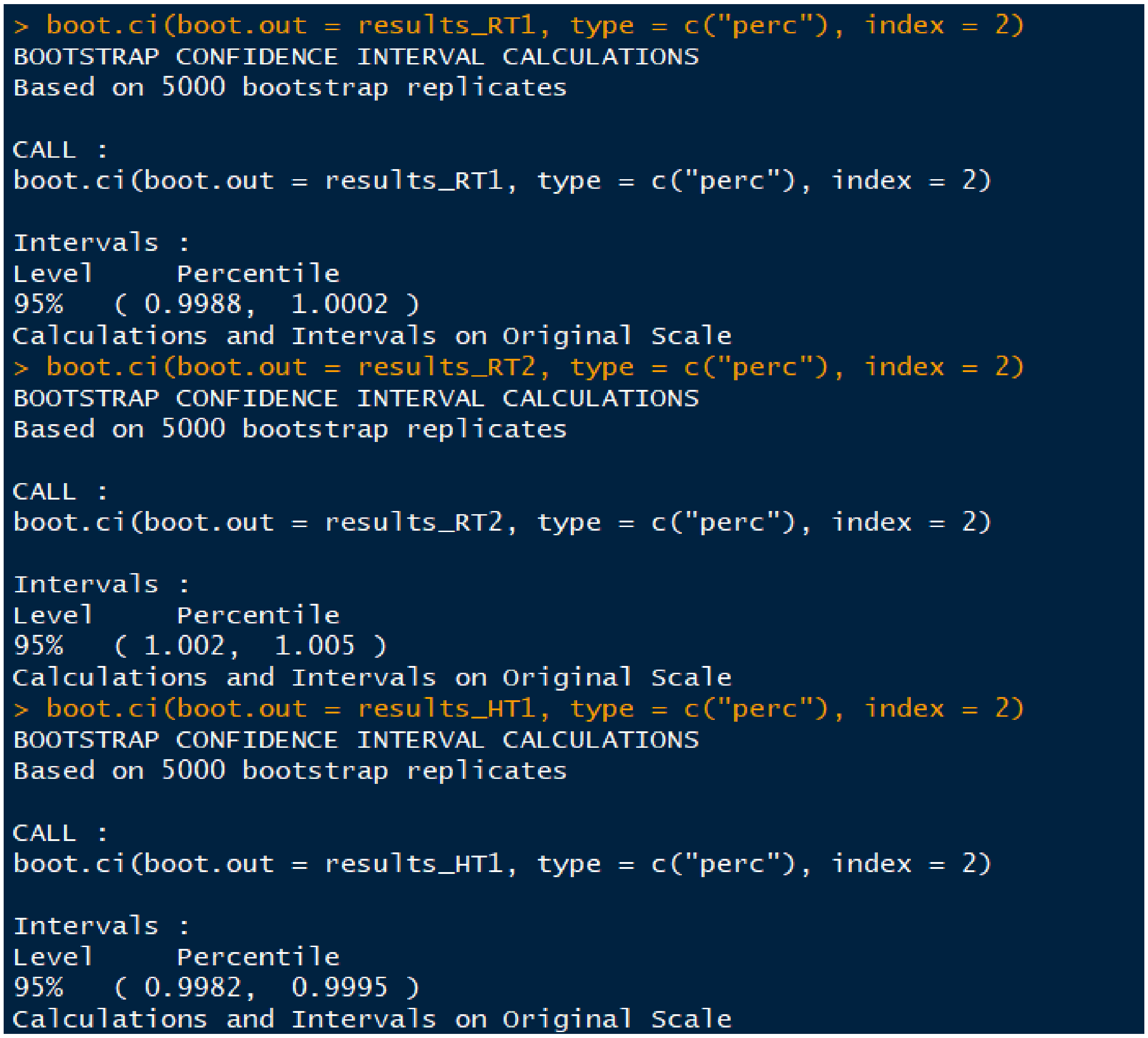


```
> boot.ci(boot.out = results_RT1, type = c("perc"), index = 1)
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 5000 bootstrap replicates

CALL :
boot.ci(boot.out = results_RT1, type = c("perc"), index = 1)

Intervals :
Level     Percentile
95%   ( 2.782,  3.935 )
Calculations and Intervals on Original Scale
> boot.ci(boot.out = results_RT2, type = c("perc"), index = 1)
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 5000 bootstrap replicates

CALL :
boot.ci(boot.out = results_RT2, type = c("perc"), index = 1)

Intervals :
Level     Percentile
95%   ( 0.921,  3.085 )
Calculations and Intervals on Original Scale
> boot.ci(boot.out = results_HT1, type = c("perc"), index = 1)
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 5000 bootstrap replicates

CALL :
boot.ci(boot.out = results_HT1, type = c("perc"), index = 1)

Intervals :
Level     Percentile
95%   ( 2.304,  3.220 )
Calculations and Intervals on Original Scale
```

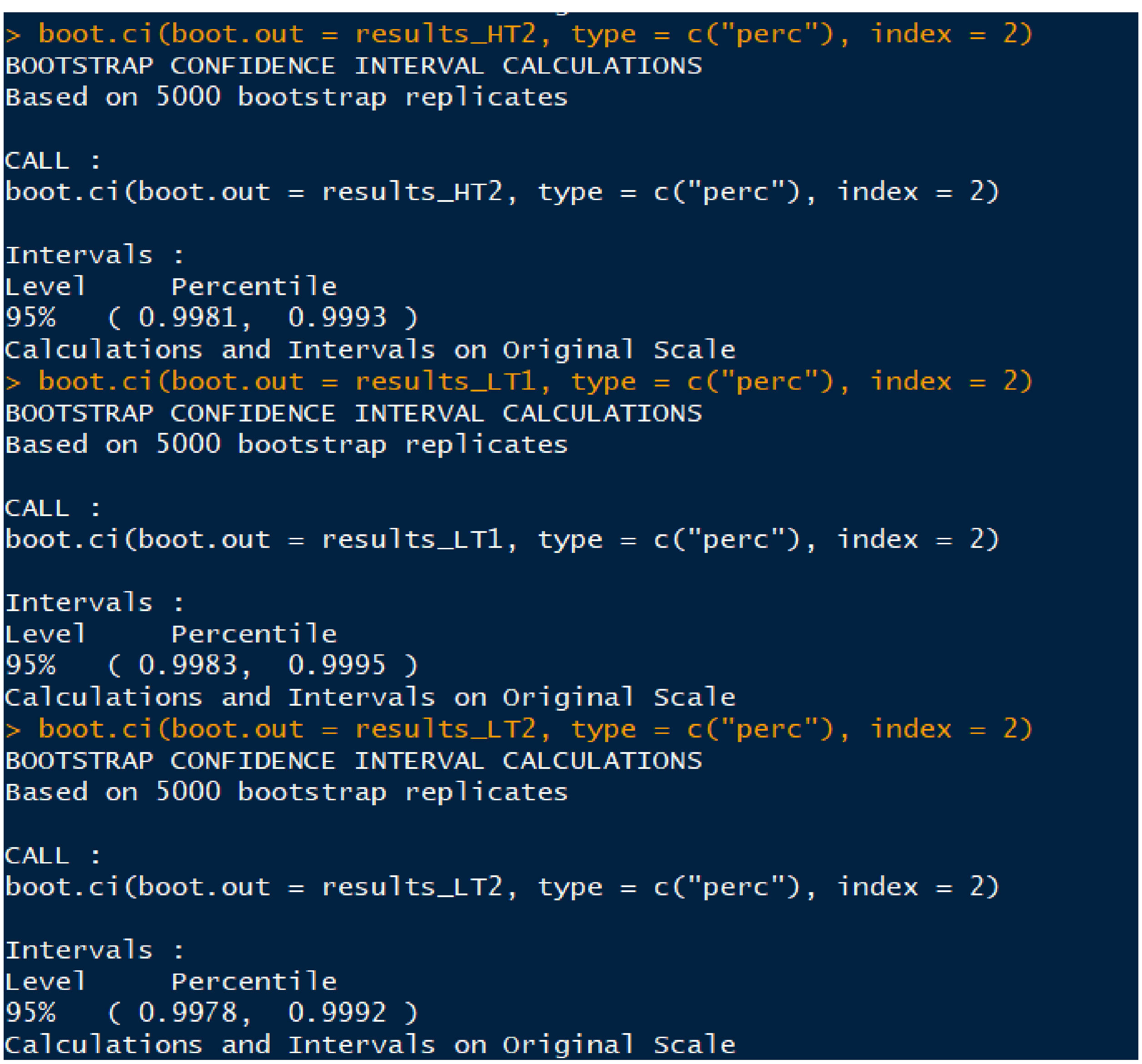


```
> boot.ci(boot.out = results_HT2, type = c("perc"), index = 1)
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 5000 bootstrap replicates

CALL :
boot.ci(boot.out = results_HT2, type = c("perc"), index = 1)

Intervals :
Level     Percentile
95%   ( 2.019,  2.924 )
Calculations and Intervals on Original Scale
> boot.ci(boot.out = results_LT1, type = c("perc"), index = 1)
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 5000 bootstrap replicates

CALL :
boot.ci(boot.out = results_LT1, type = c("perc"), index = 1)

Intervals :
Level     Percentile
95%   ( 2.168,  3.047 )
Calculations and Intervals on Original Scale
> boot.ci(boot.out = results_LT2, type = c("perc"), index = 1)
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 5000 bootstrap replicates

CALL :
boot.ci(boot.out = results_LT2, type = c("perc"), index = 1)

Intervals :
Level     Percentile
95%   ( 2.261,  3.192 )
Calculations and Intervals on Original Scale
```

Slope Confidence Intervals After Resampling



```
> boot.ci(boot.out = results_RT1, type = c("perc"), index = 2)
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 5000 bootstrap replicates

CALL :
boot.ci(boot.out = results_RT1, type = c("perc"), index = 2)

Intervals :
Level     Percentile
95%   ( 0.9988,  1.0002 )
Calculations and Intervals on Original Scale
> boot.ci(boot.out = results_RT2, type = c("perc"), index = 2)
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 5000 bootstrap replicates

CALL :
boot.ci(boot.out = results_RT2, type = c("perc"), index = 2)

Intervals :
Level     Percentile
95%   ( 1.002,  1.005 )
Calculations and Intervals on Original Scale
> boot.ci(boot.out = results_HT1, type = c("perc"), index = 2)
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 5000 bootstrap replicates

CALL :
boot.ci(boot.out = results_HT1, type = c("perc"), index = 2)

Intervals :
Level     Percentile
95%   ( 0.9982,  0.9995 )
Calculations and Intervals on Original Scale
```

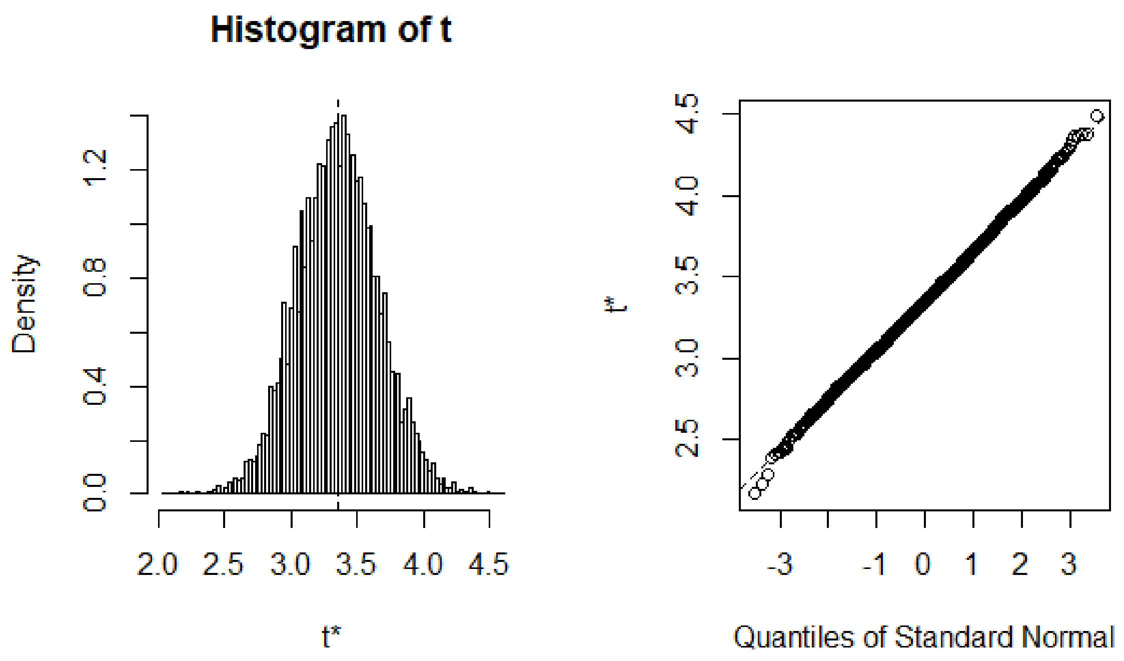


```
> boot.ci(boot.out = results_HT2, type = c("perc"), index = 2)
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 5000 bootstrap replicates

CALL :
boot.ci(boot.out = results_HT2, type = c("perc"), index = 2)

Intervals :
Level     Percentile
95%   ( 0.9981,  0.9993 )
Calculations and Intervals on Original Scale
> boot.ci(boot.out = results_LT1, type = c("perc"), index = 2)
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 5000 bootstrap replicates

CALL :
boot.ci(boot.out = results_LT1, type = c("perc"), index = 2)

Intervals :
Level     Percentile
95%   ( 0.9983,  0.9995 )
Calculations and Intervals on Original Scale
> boot.ci(boot.out = results_LT2, type = c("perc"), index = 2)
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 5000 bootstrap replicates

CALL :
boot.ci(boot.out = results_LT2, type = c("perc"), index = 2)

Intervals :
Level     Percentile
95%   ( 0.9978,  0.9992 )
Calculations and Intervals on Original Scale
```

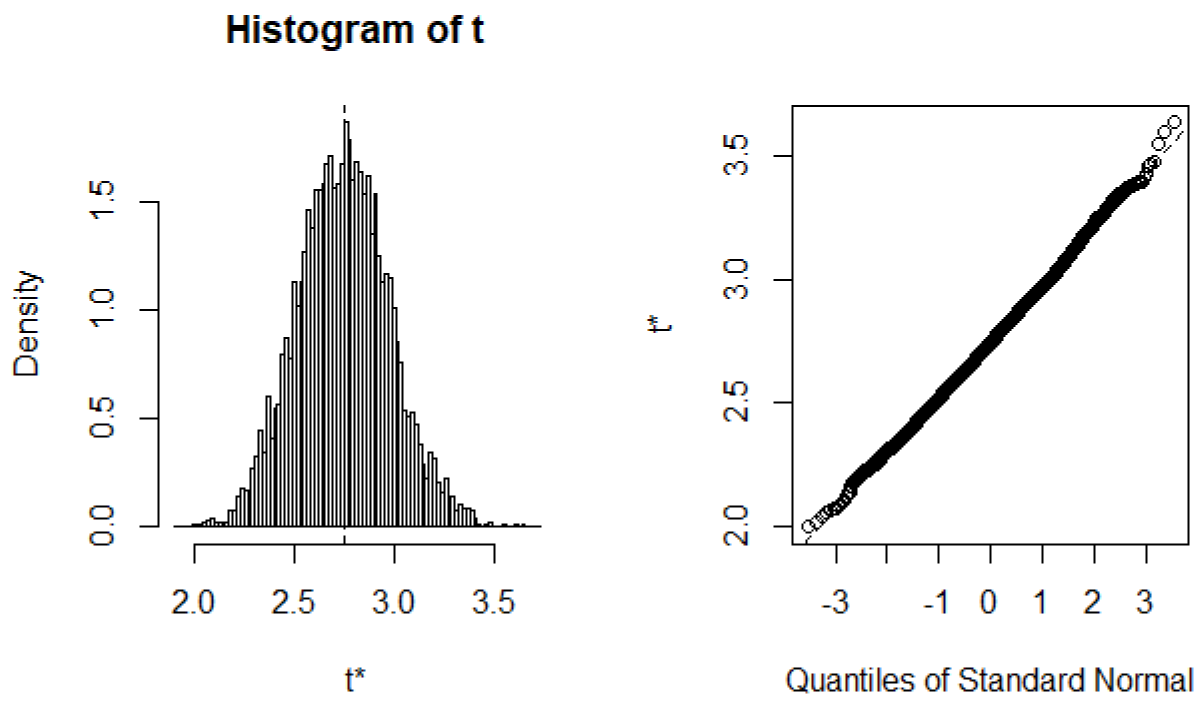
Linear Regression Plot after Resampling

Sensor 1:

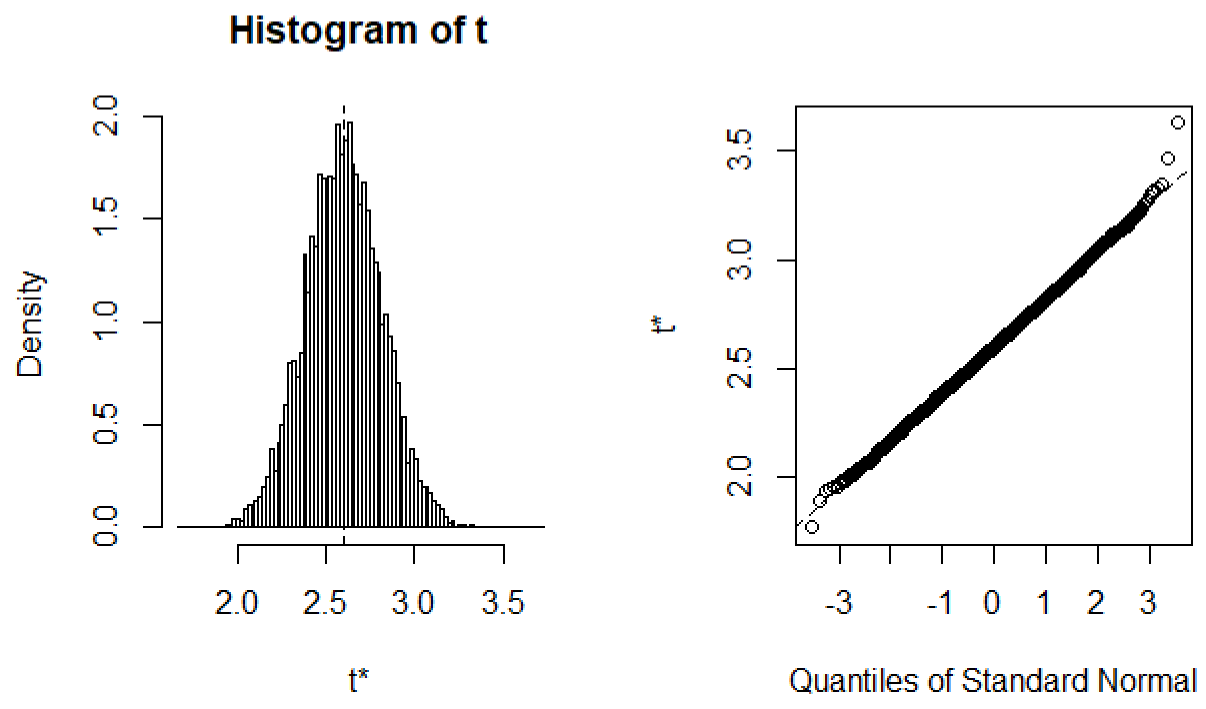
Room Temperature



High Temperature

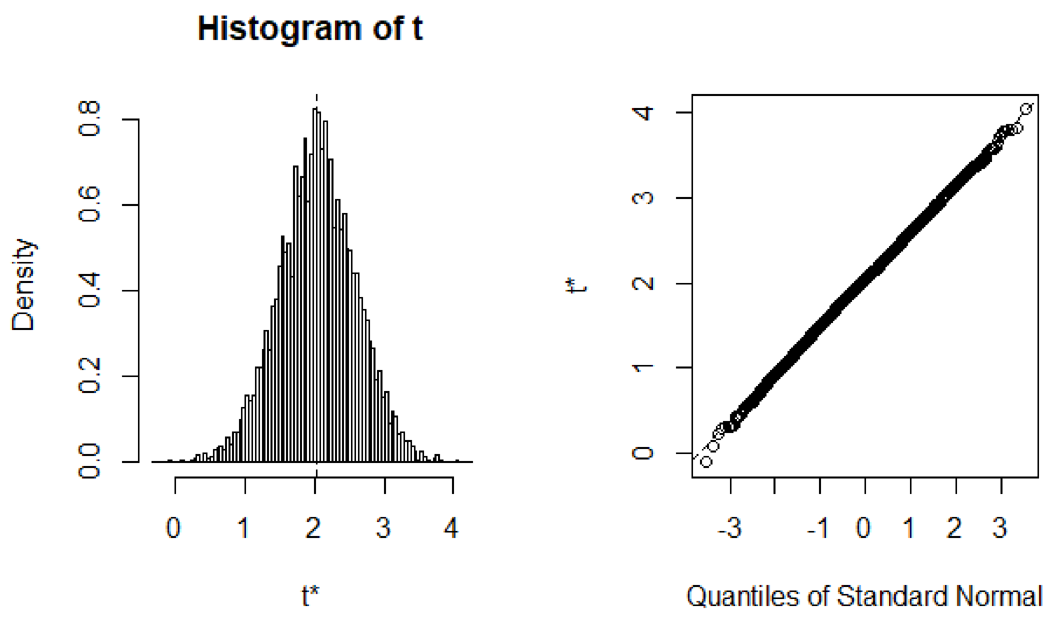


<u>Low Temperature</u>

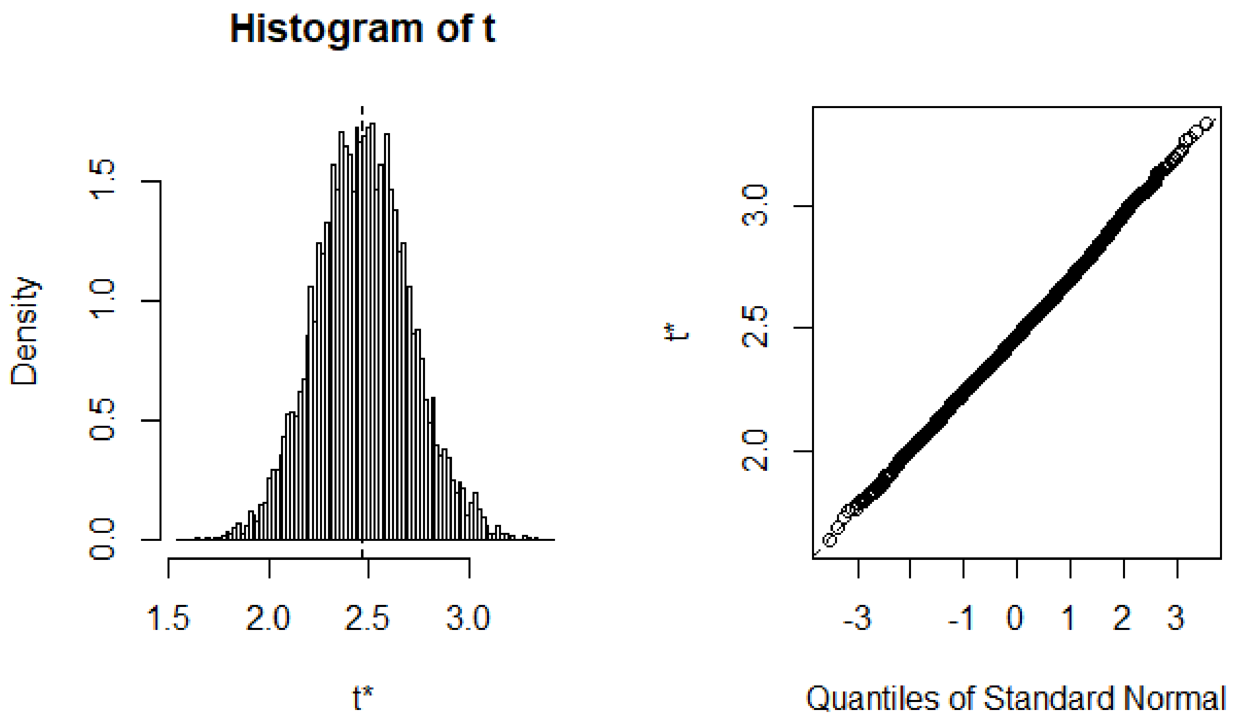


Sensor 2:

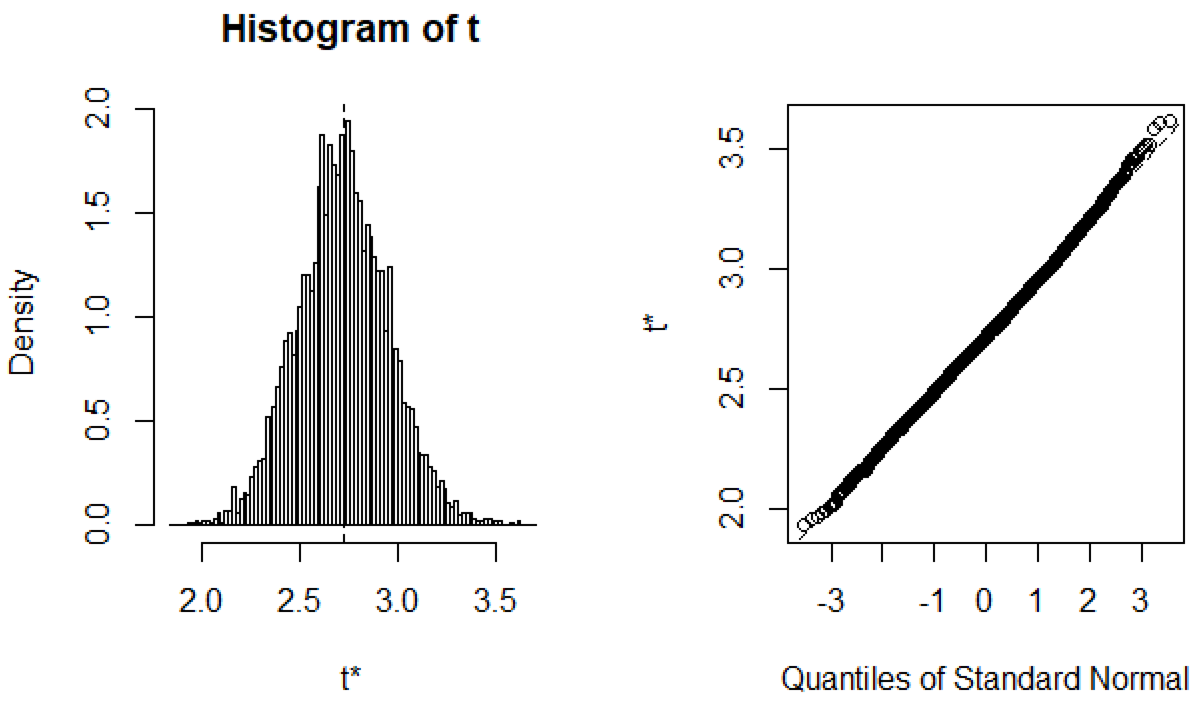
<u>Room Temperature</u>



High Temperature



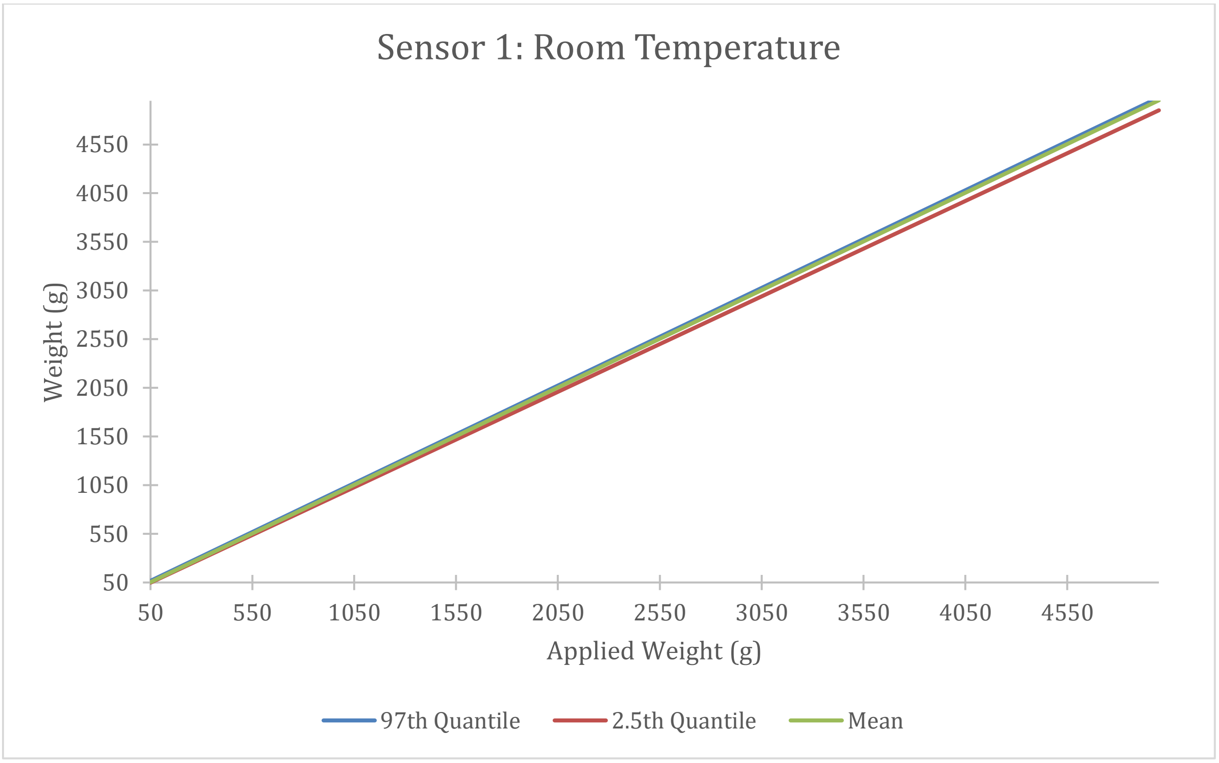
Low Temperature

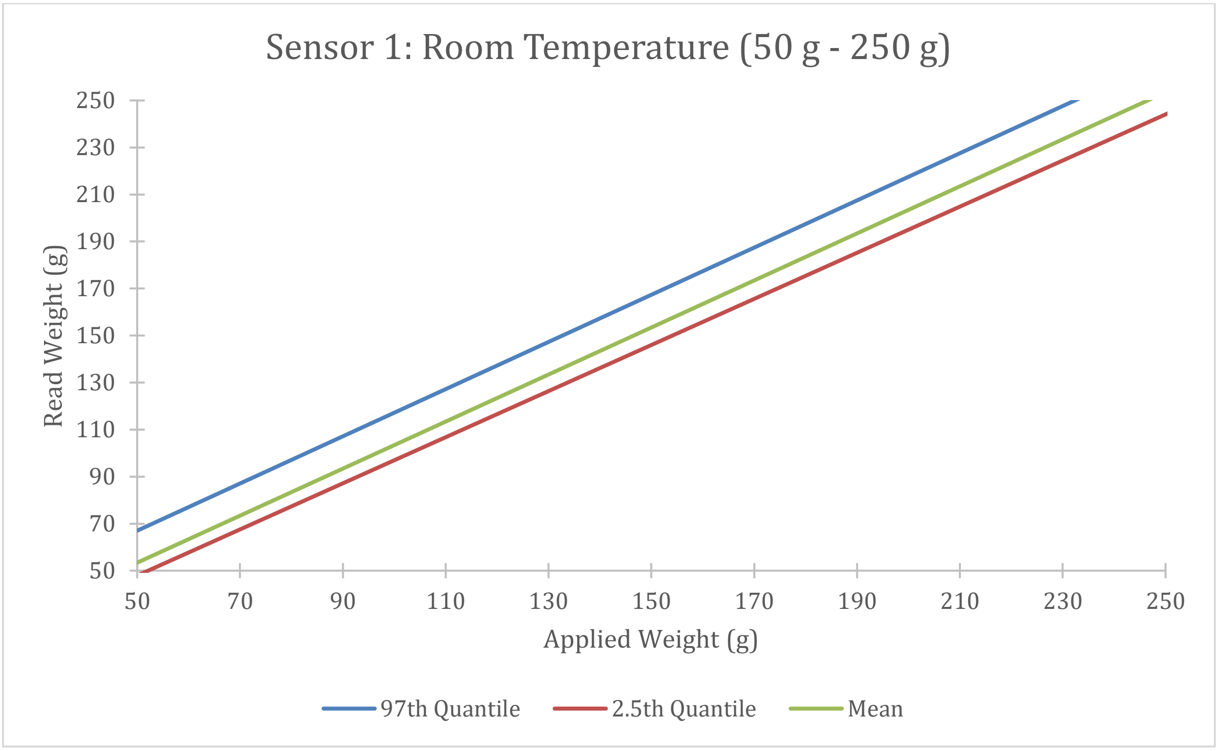

Histogram of t
Density
0.0
0.5
1.0
1.5
2.0
2.0
2.5
3.0
3.5
t*
t*
2.0
2.5
3.0
3.5
-3
-1
0
1
2
3
Quantiles of Standard Normal

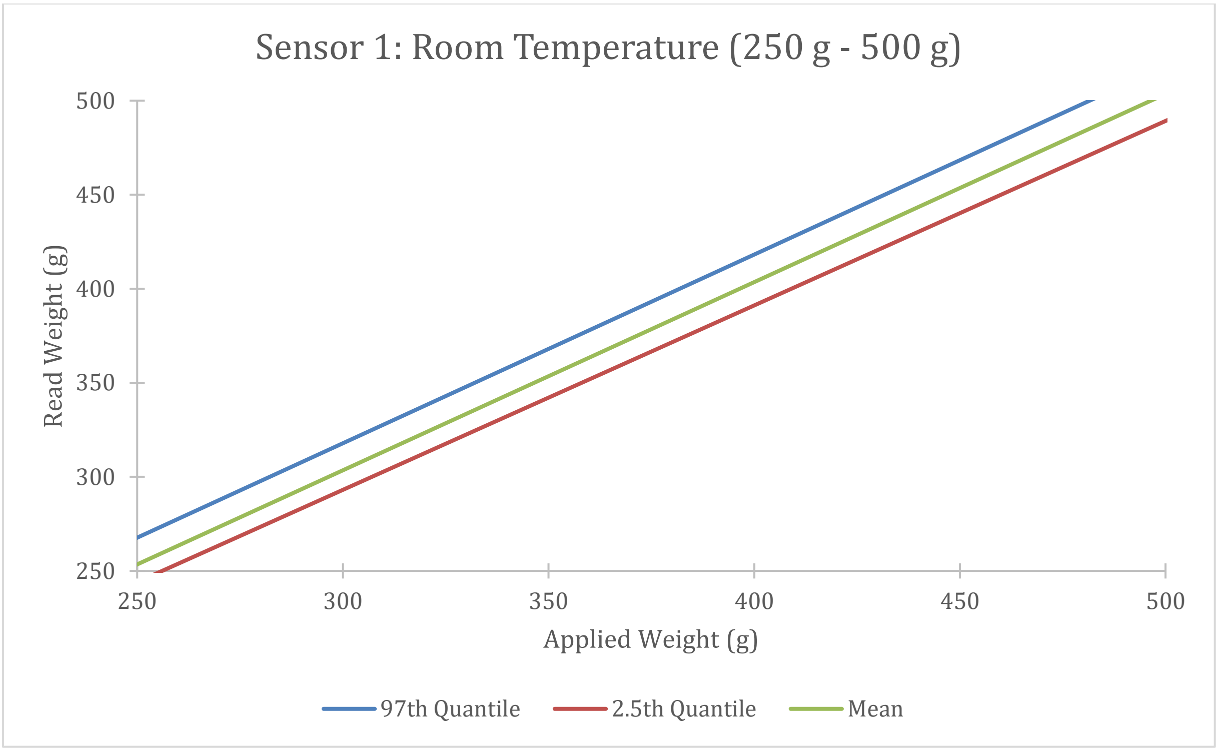
Quantile Regression Plots

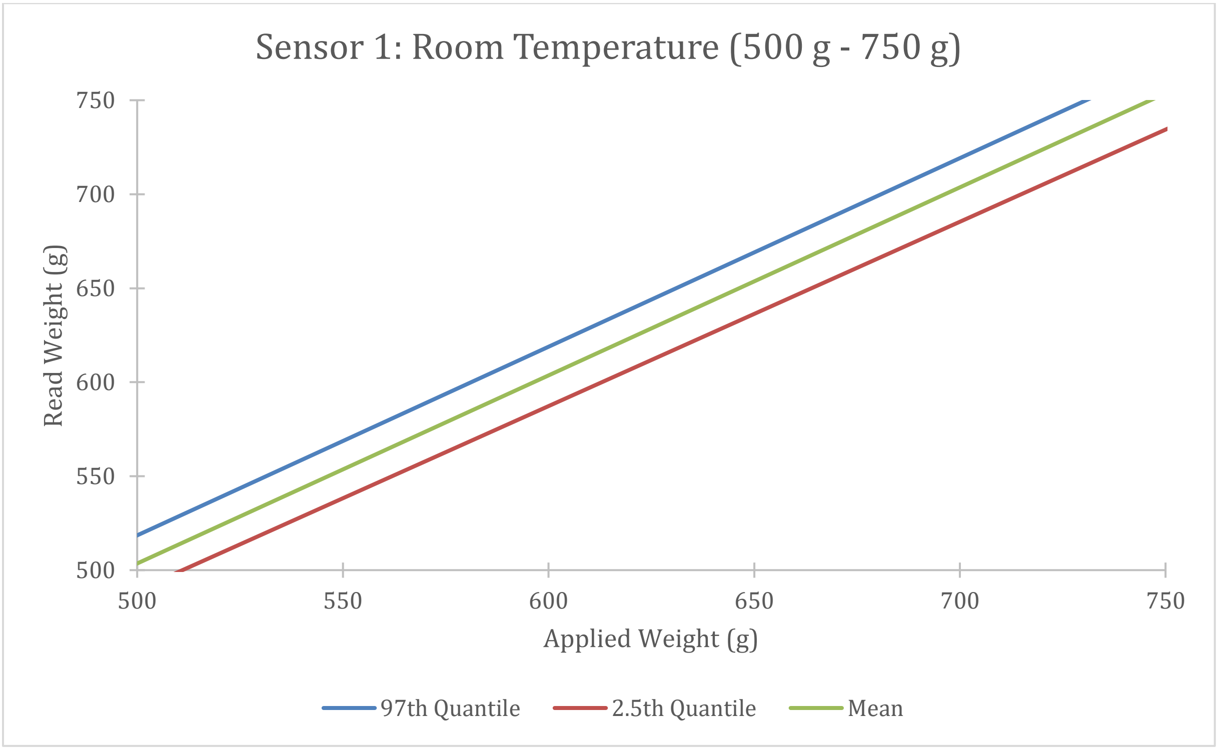
Sensor 1:

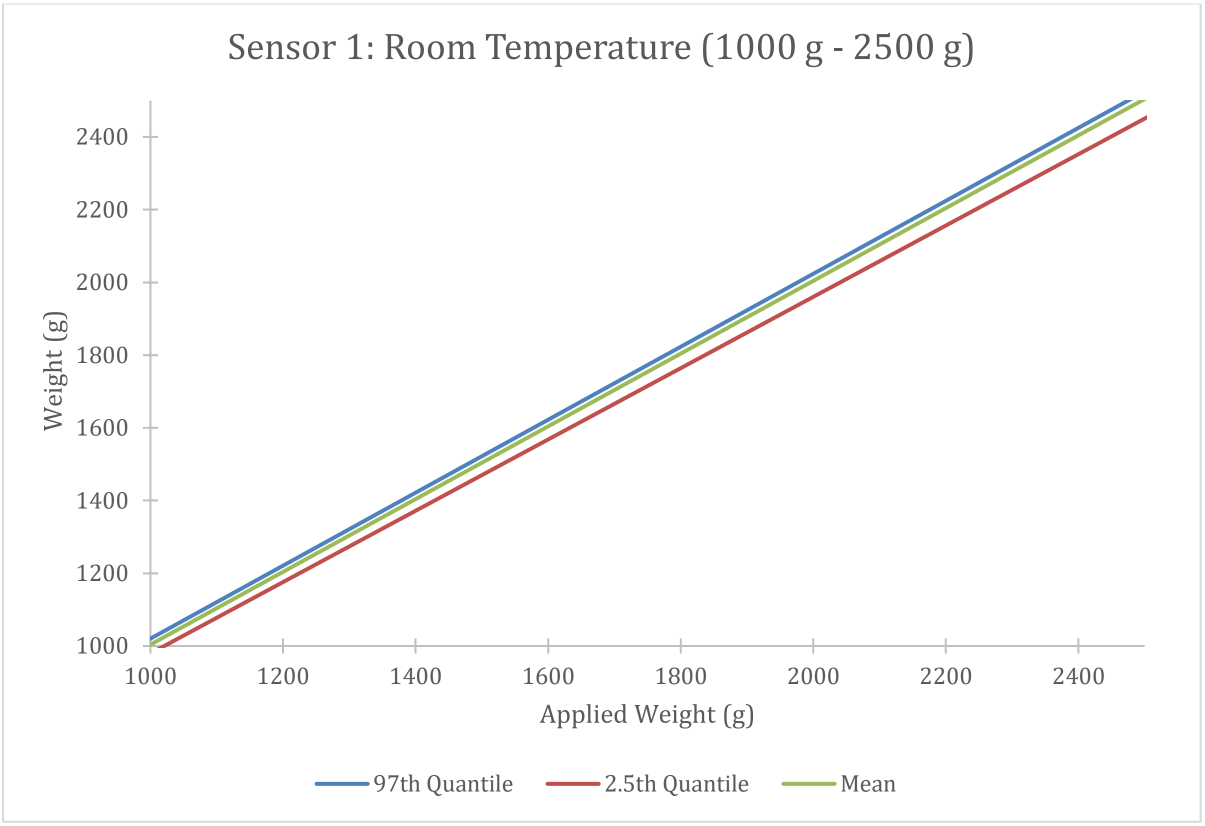
Room Temperature

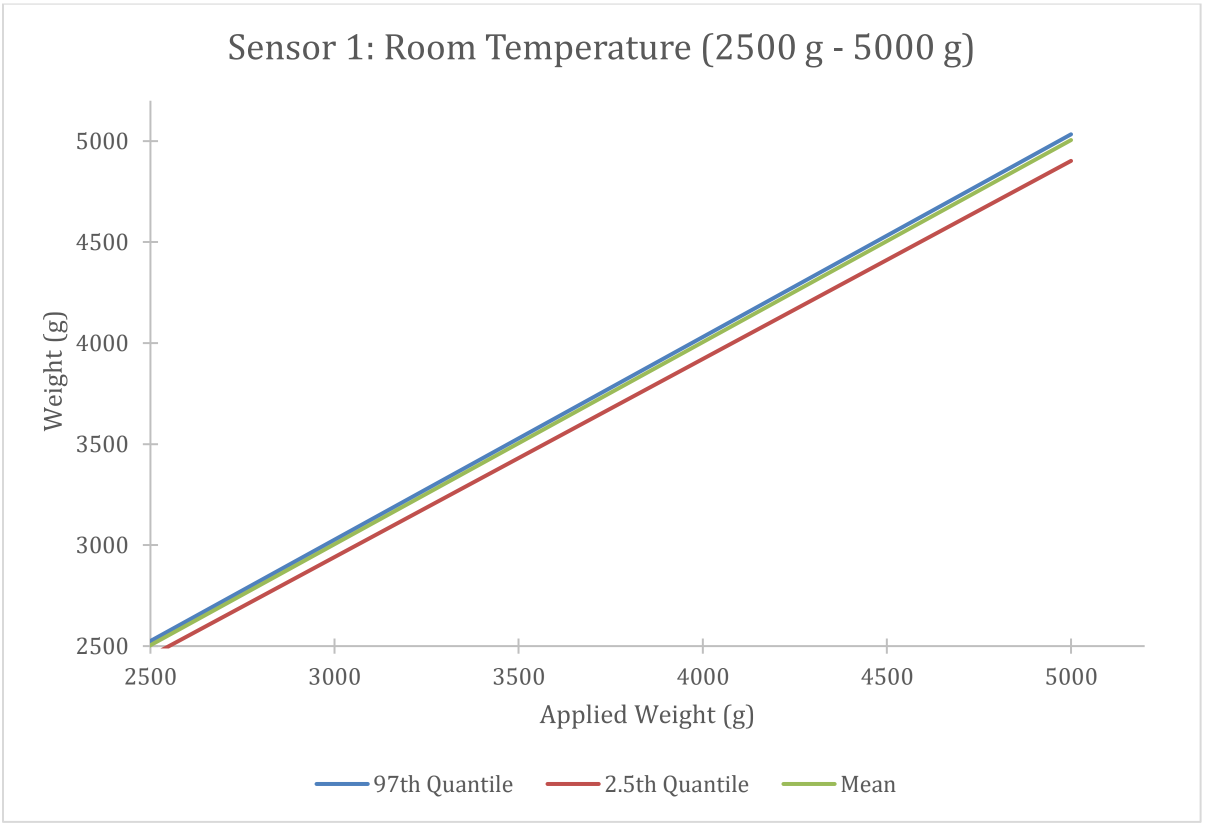




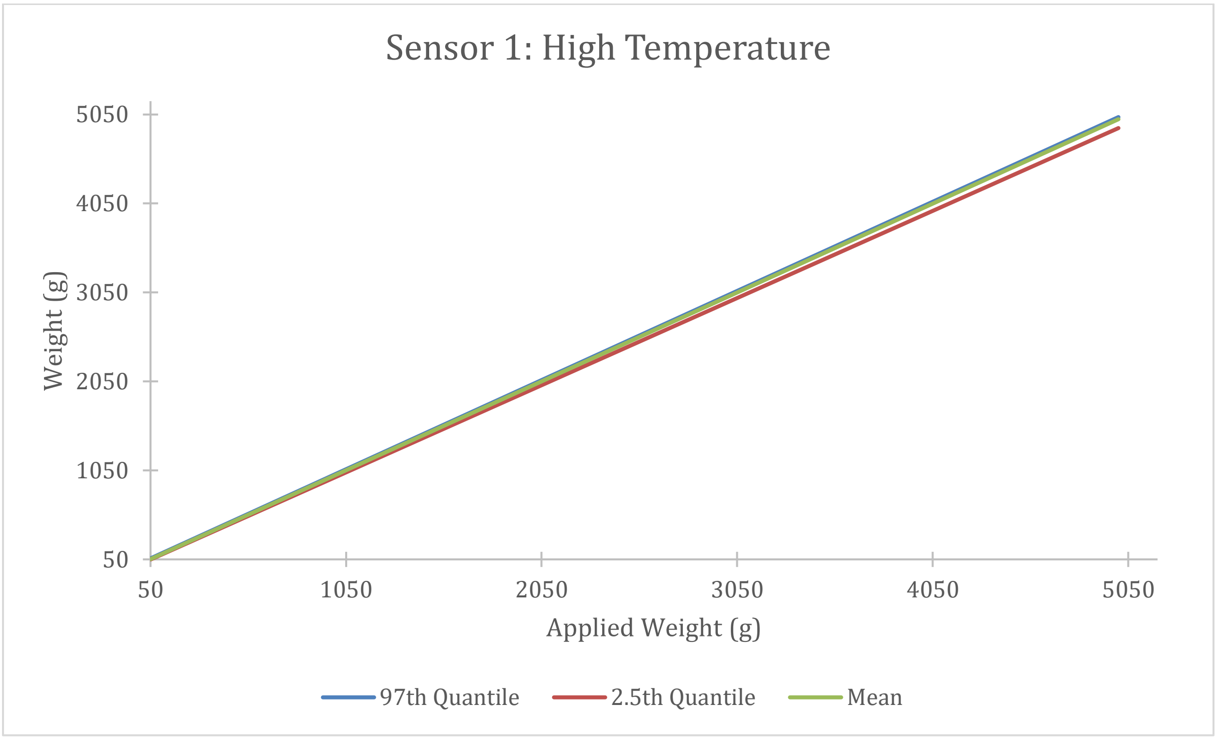

Sensor 1: Room Temperature (250 g - 500 g)
Read Weight (g)
500
450
400
350
300
250
250
300
350
400
450
500
Applied Weight (g)
97th Quantile
2.5th Quantile
Mean

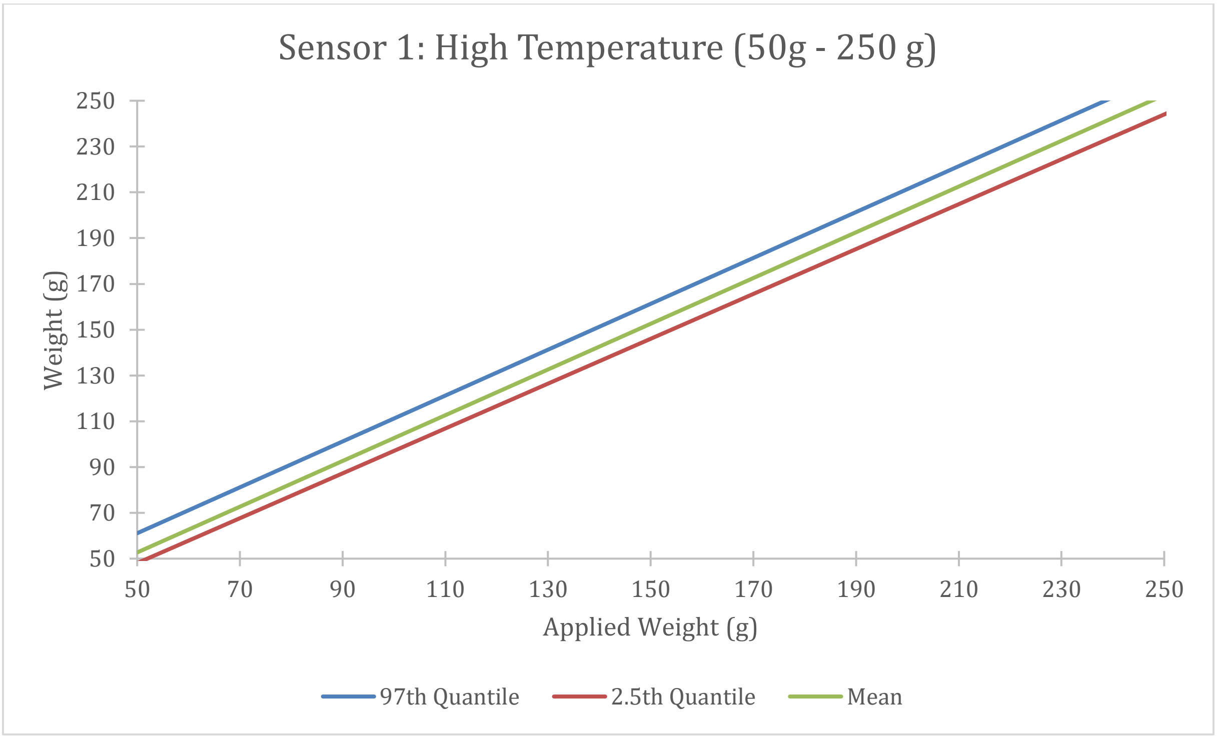

Sensor 1: Room Temperature (500 g - 750 g)
Read Weight (g)
750
700
650
600
550
500
500
550
600
650
700
750
Applied Weight (g)
97th Quantile
2.5th Quantile
Mean

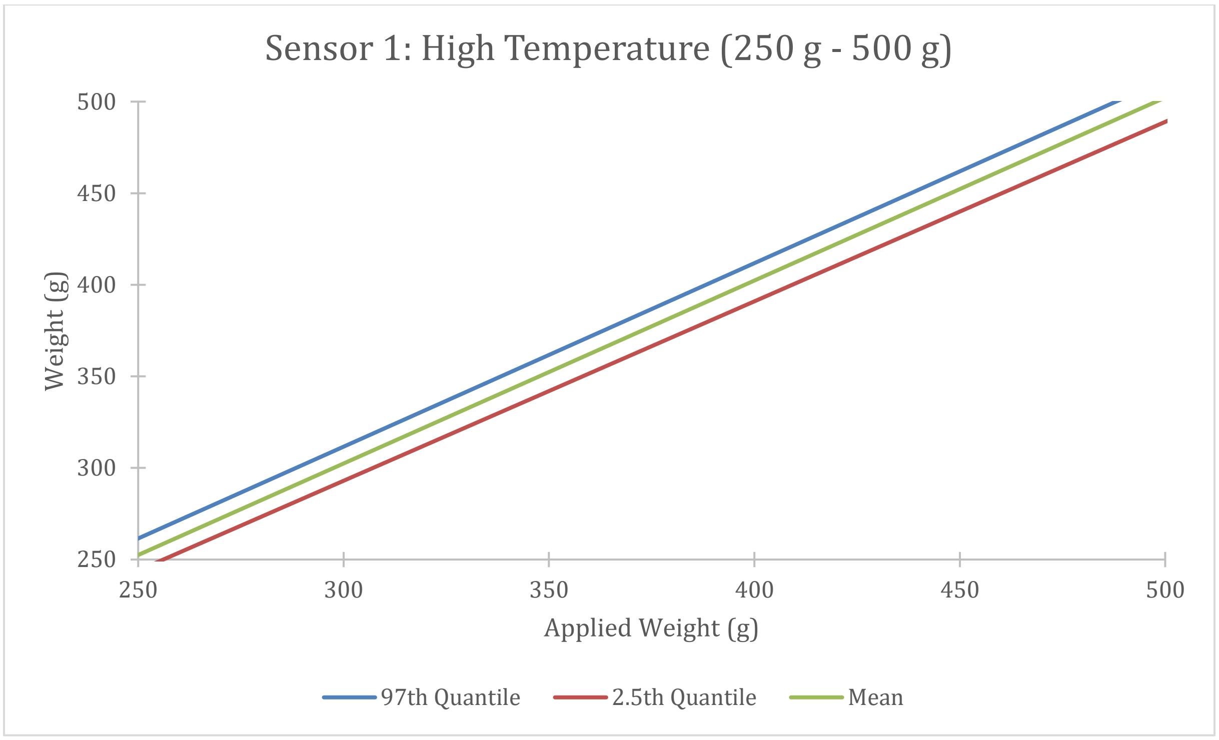

Sensor 1: Room Temperature (1000 g - 2500 g)
Weight (g)
2400
2200
2000
1800
1600
1400
1200
1000
1000
1200
1400
1600
1800
2000
2200
2400
Applied Weight (g)
97th Quantile
2.5th Quantile
Mean

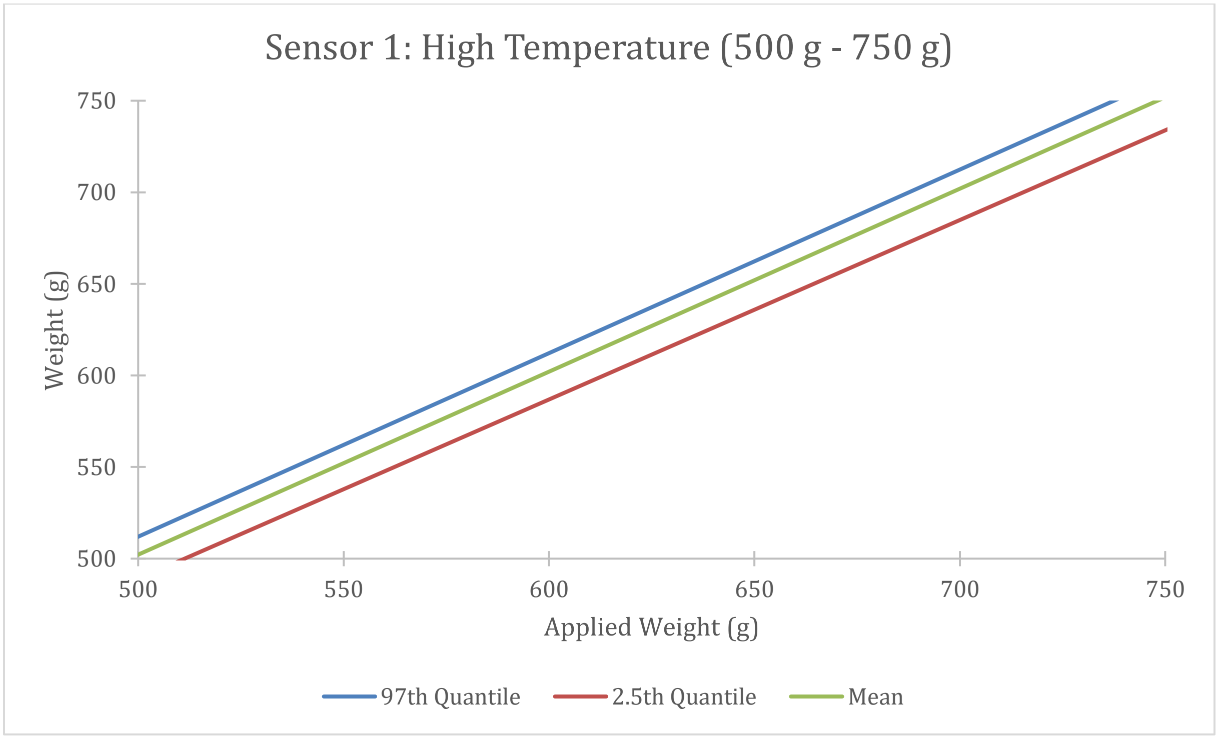

Sensor 1: Room Temperature (2500 g - 5000 g)
Weight (g)
5000
4500
4000
3500
3000
2500
2500
3000
3500
4000
4500
5000
Applied Weight (g)
97th Quantile
2.5th Quantile
Mean

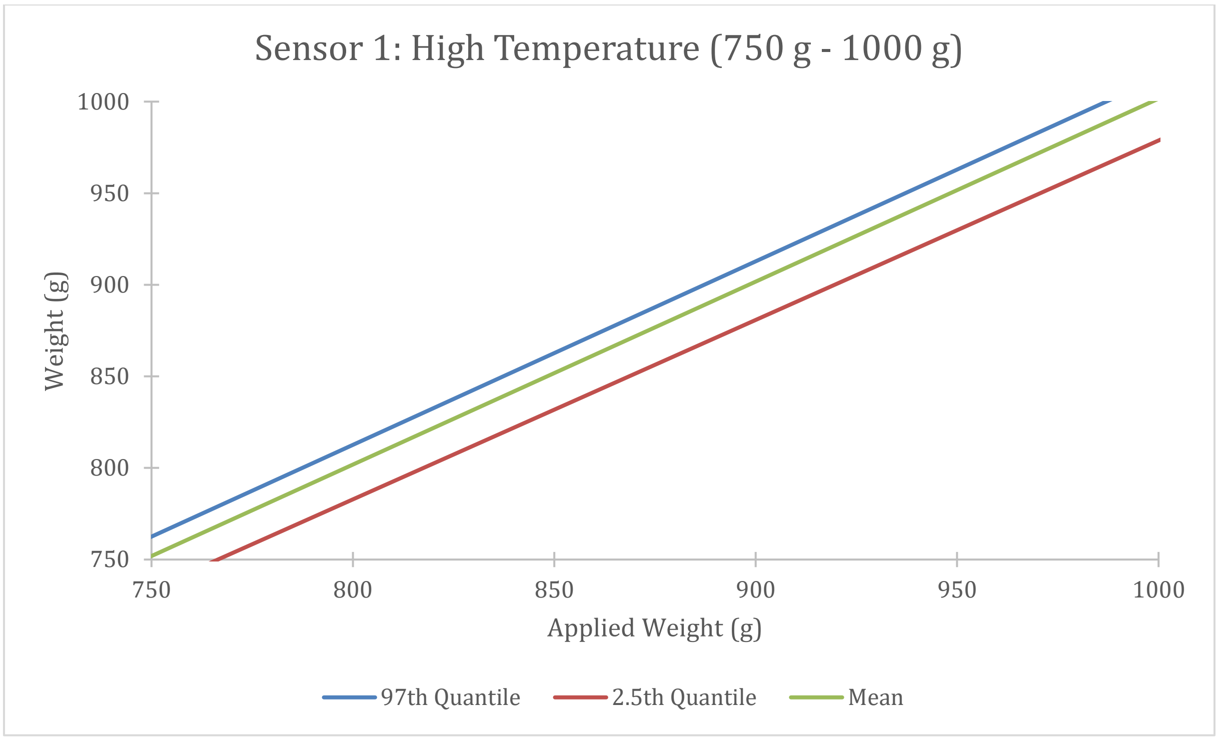
High Temperature

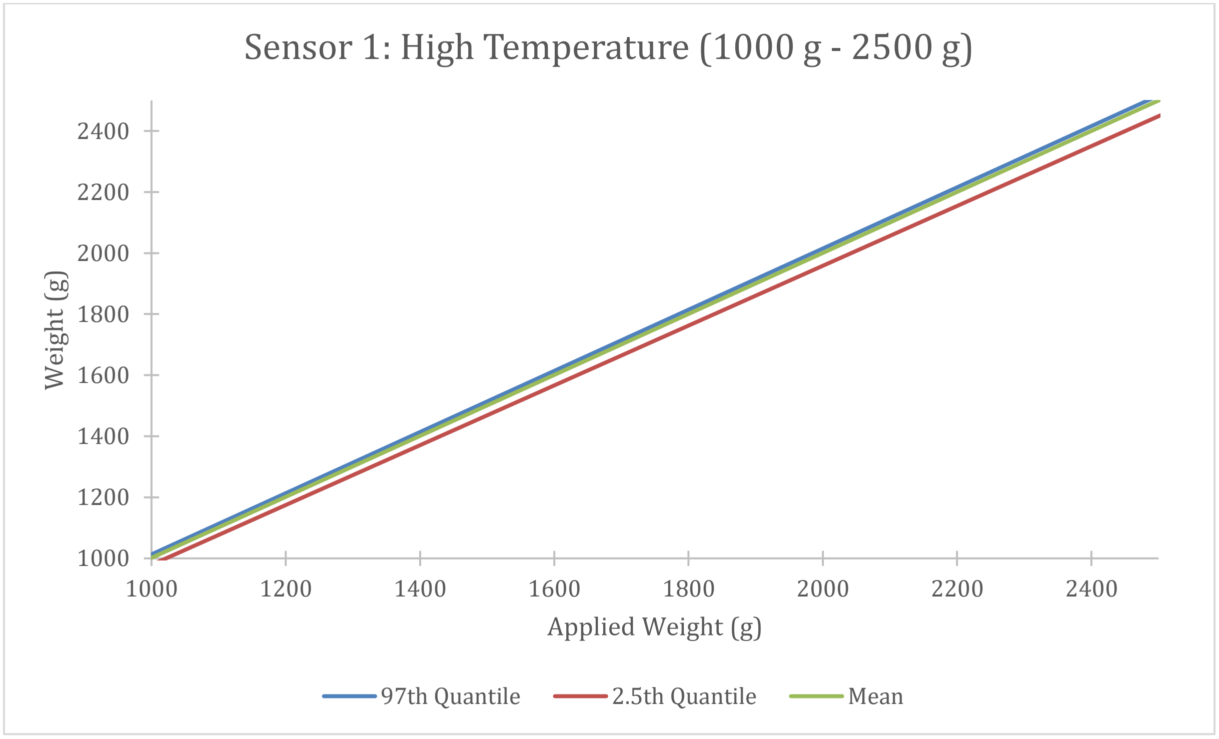


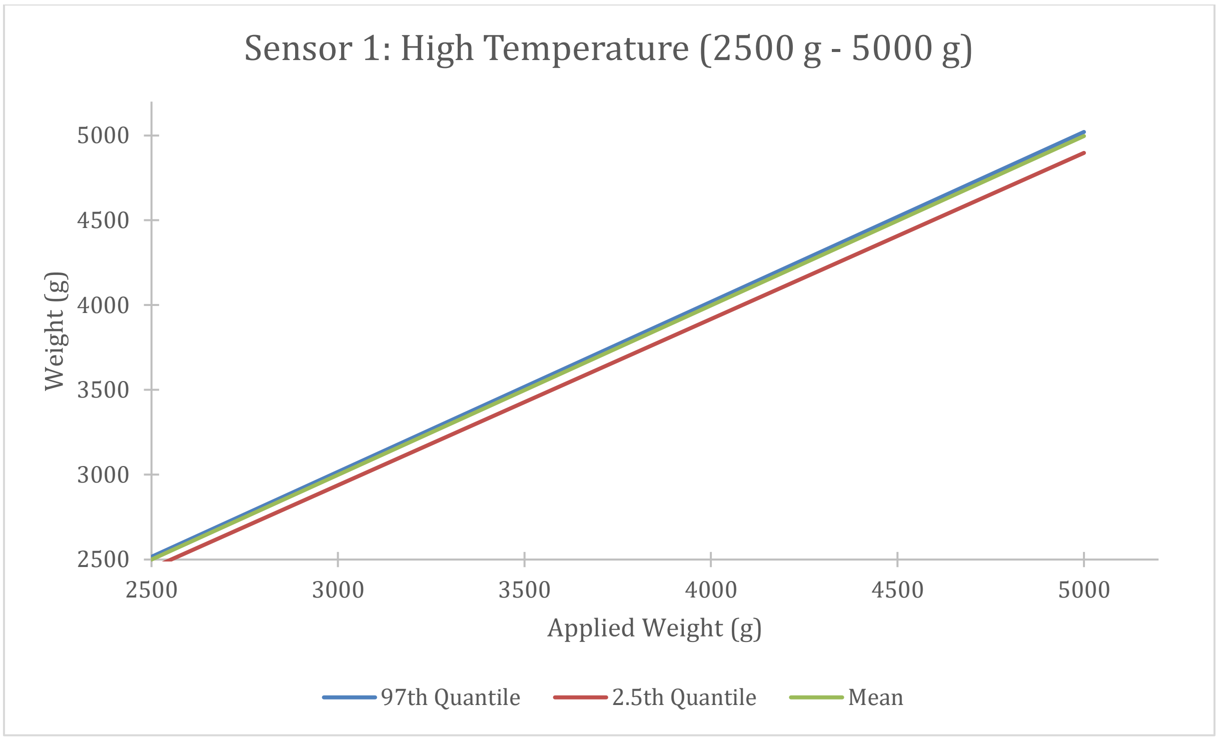



Sensor 1: High Temperature (250 g - 500 g)
Weight (g)
500
450
400
350
300
250
250
300
350
400
450
500
Applied Weight (g)
97th Quantile
2.5th Quantile
Mean

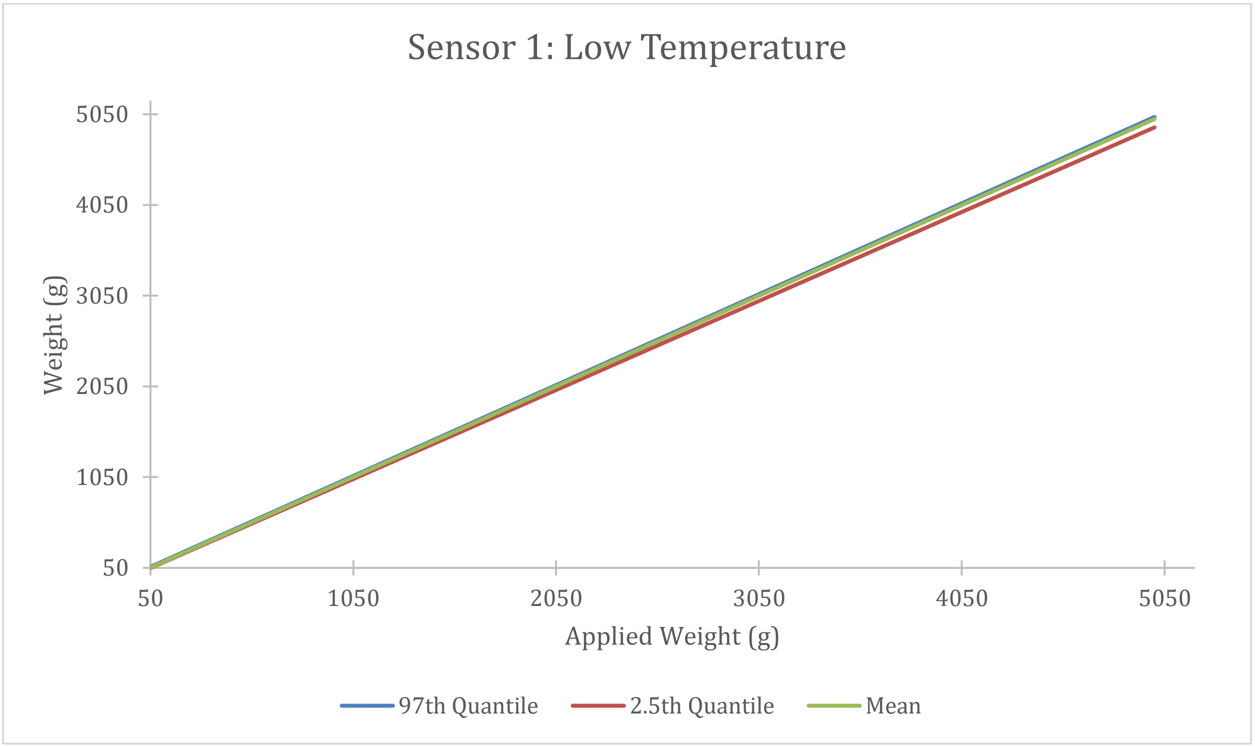

Sensor 1: High Temperature (500 g - 750 g)
Weight (g)
750
700
650
600
550
500
500
550
600
650
700
750
Applied Weight (g)
97th Quantile
2.5th Quantile
Mean

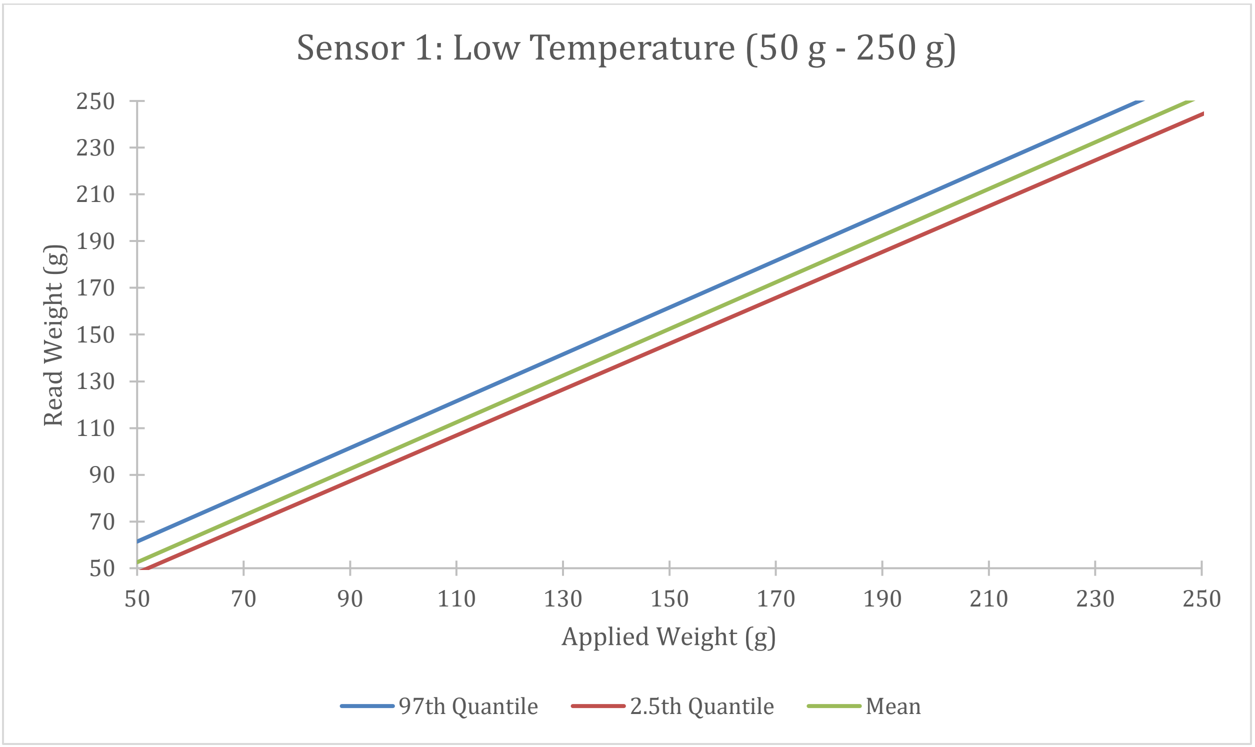

Sensor 1: High Temperature (750 g - 1000 g)
Weight (g)
Applied Weight (g)
750
800
850
900
950
1000
97th Quantile
2.5th Quantile
Mean

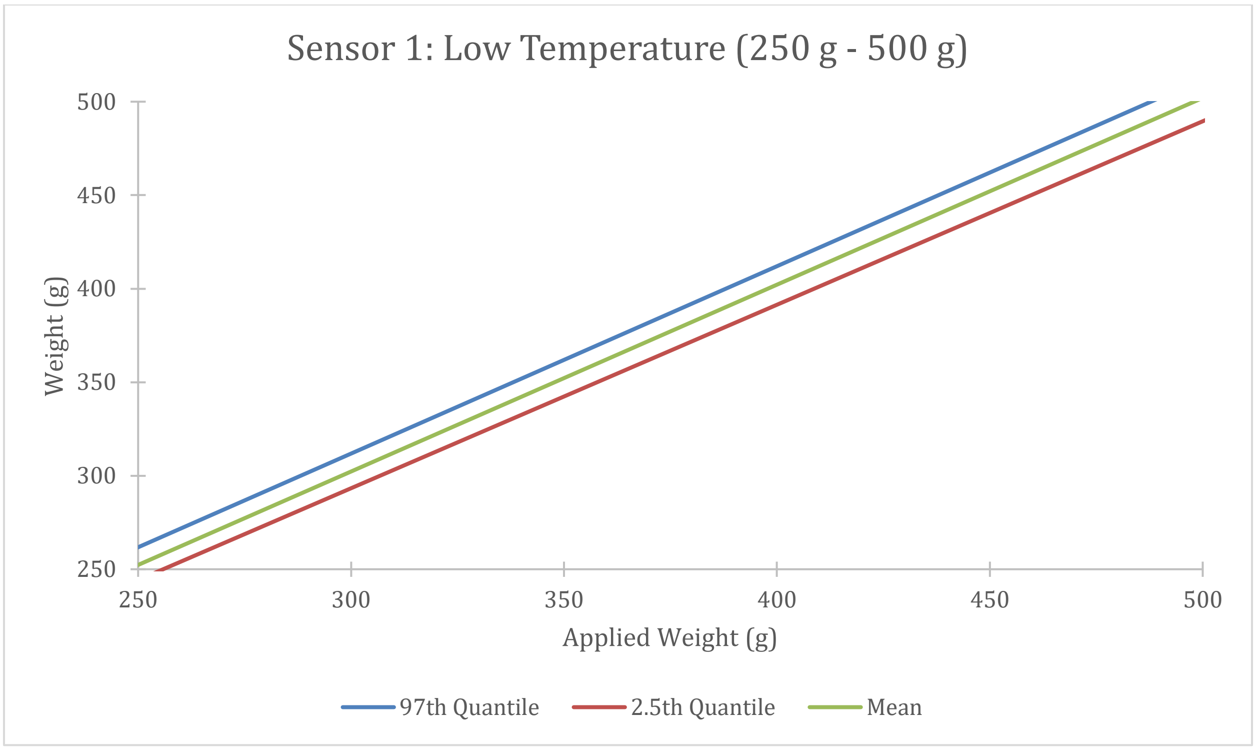

Sensor 1: High Temperature (1000 g - 2500 g)
Weight (g)
Applied Weight (g)
1000
1200
1400
1600
1800
2000
2200
2400
97th Quantile
2.5th Quantile
Mean

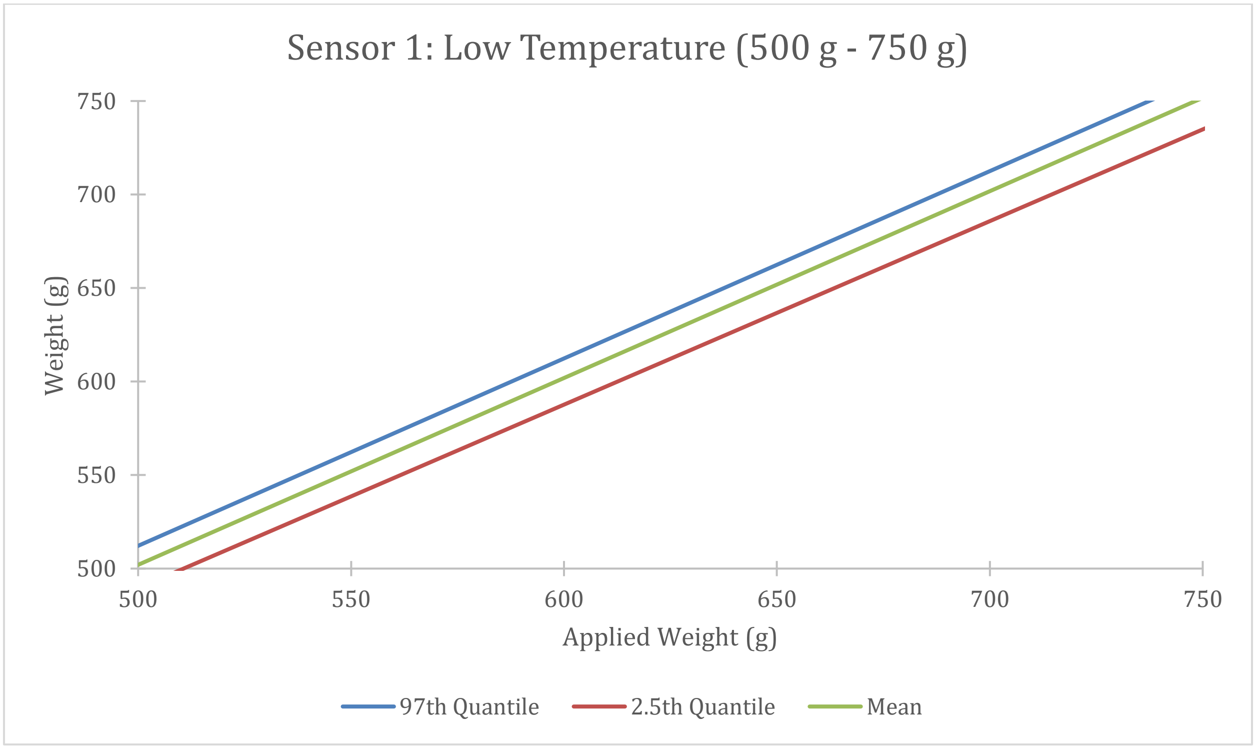

Sensor 1: High Temperature (2500 g - 5000 g)
Weight (g)
5000
4500
4000
3500
3000
2500
2500
3000
3500
4000
4500
5000
Applied Weight (g)
97th Quantile
2.5th Quantile
Mean

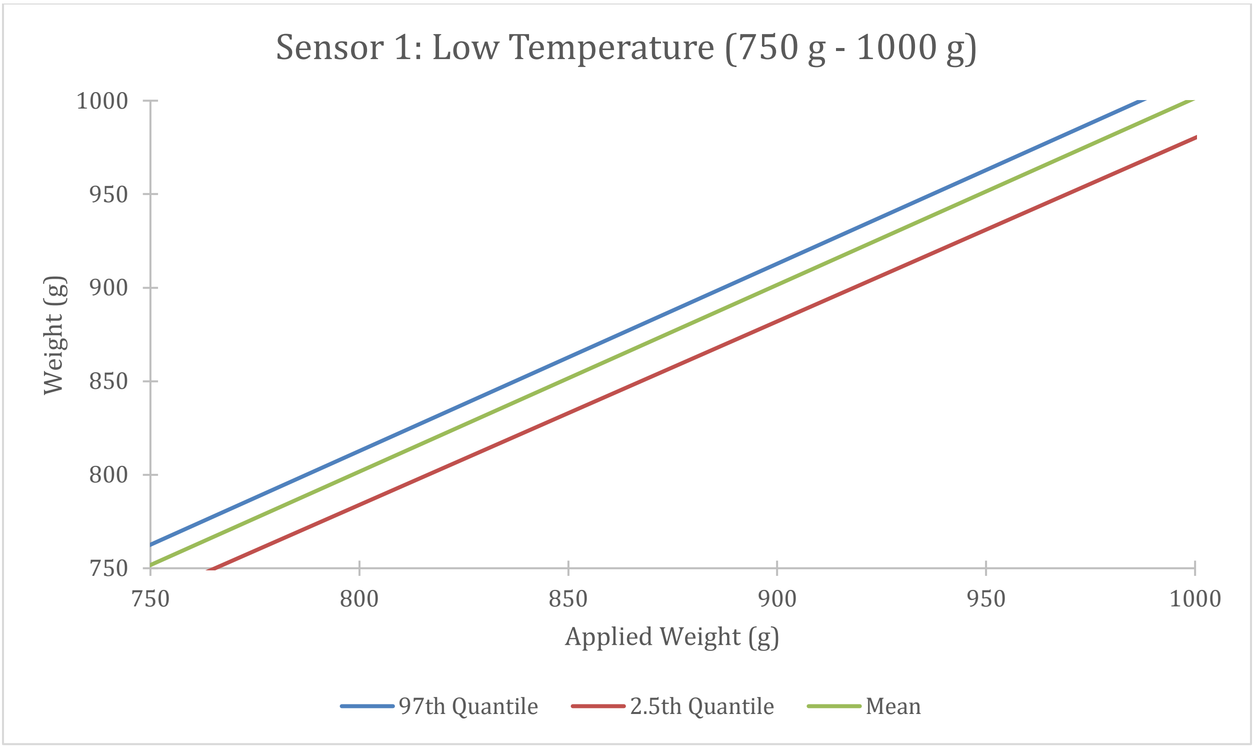
Low Temperature

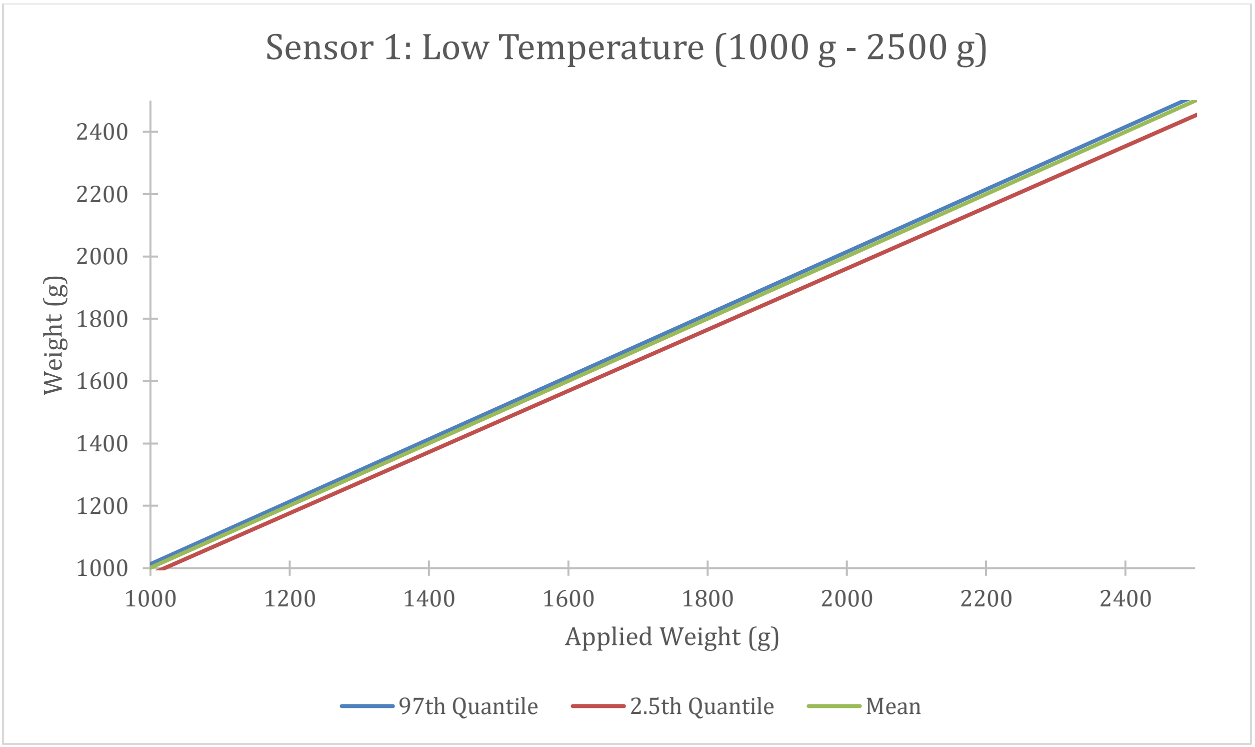


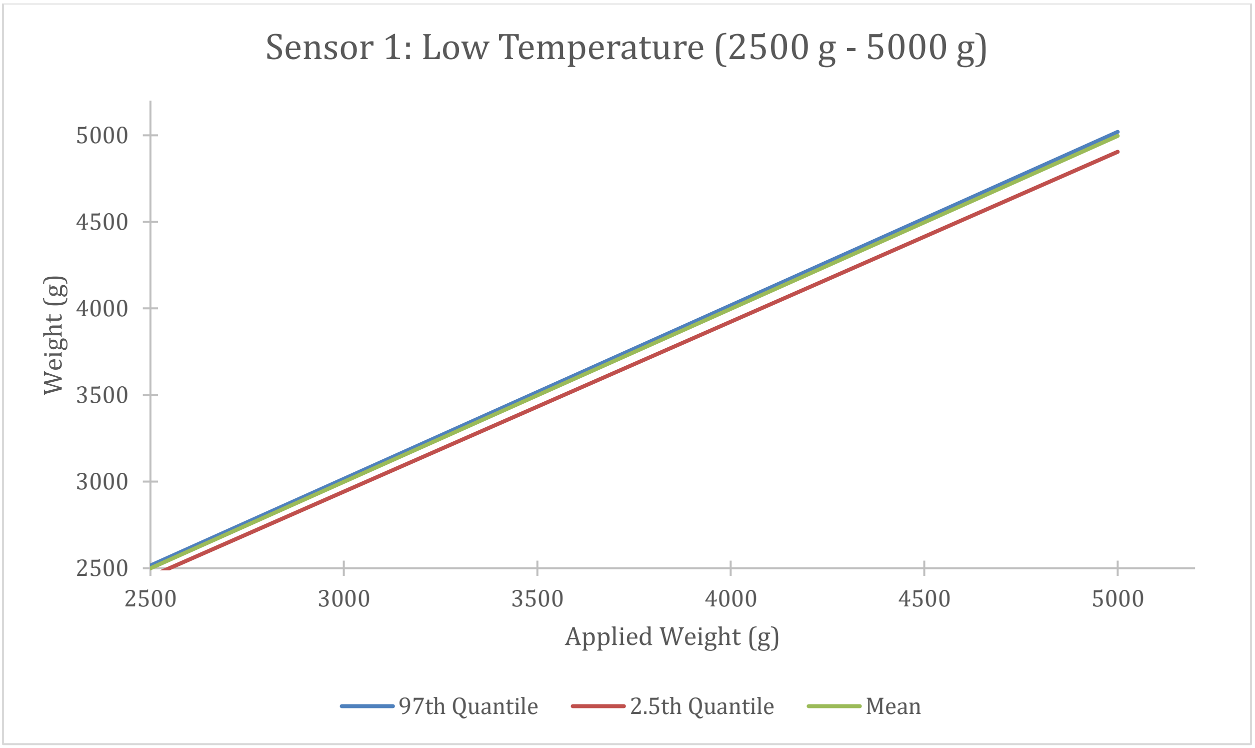



Sensor 1: Low Temperature (250 g - 500 g)
Weight (g)
Applied Weight (g)
250
300
350
400
450
500
97th Quantile
2.5th Quantile
Mean


Sensor 1: Low Temperature (500 g - 750 g)
Weight (g)
Applied Weight (g)
500
550
600
650
700
750
97th Quantile
2.5th Quantile
Mean

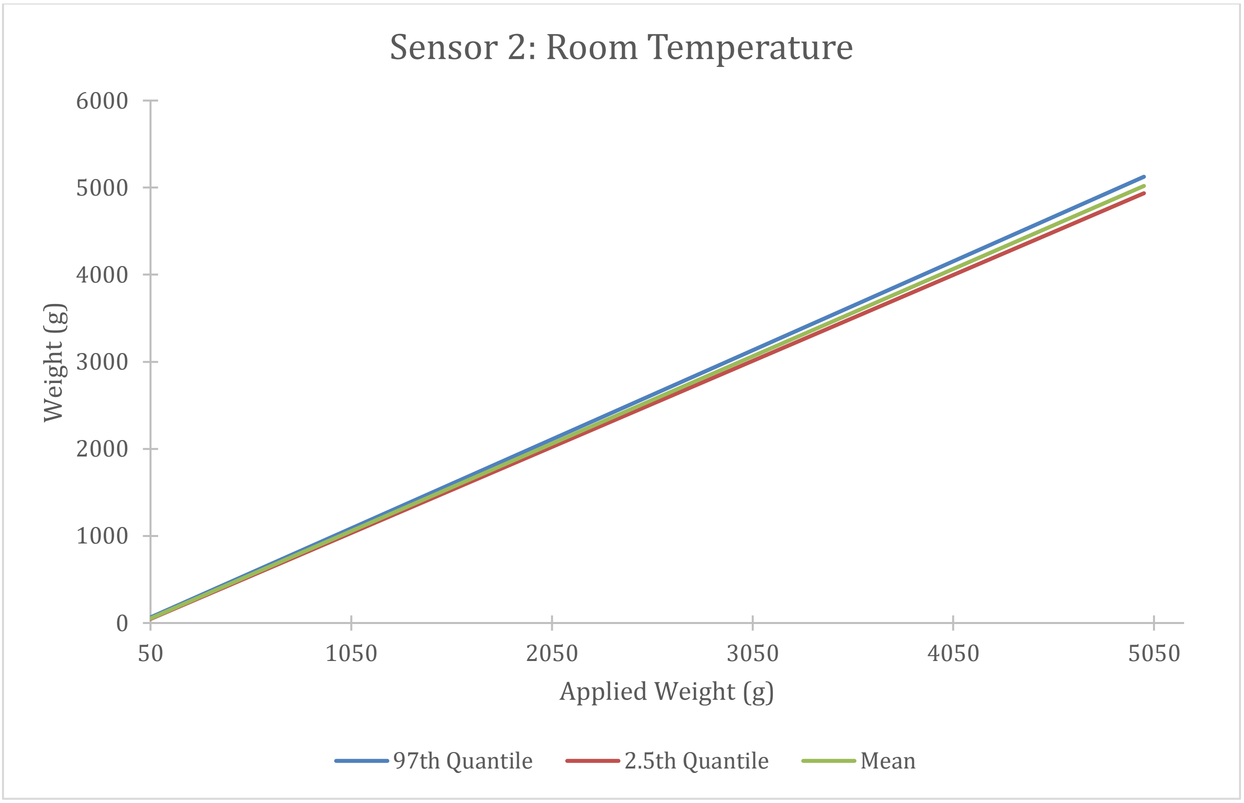

Sensor 1: Low Temperature (750 g - 1000 g)
Weight (g)
1000
950
900
850
800
750
750
800
850
900
950
1000
Applied Weight (g)
97th Quantile
2.5th Quantile
Mean

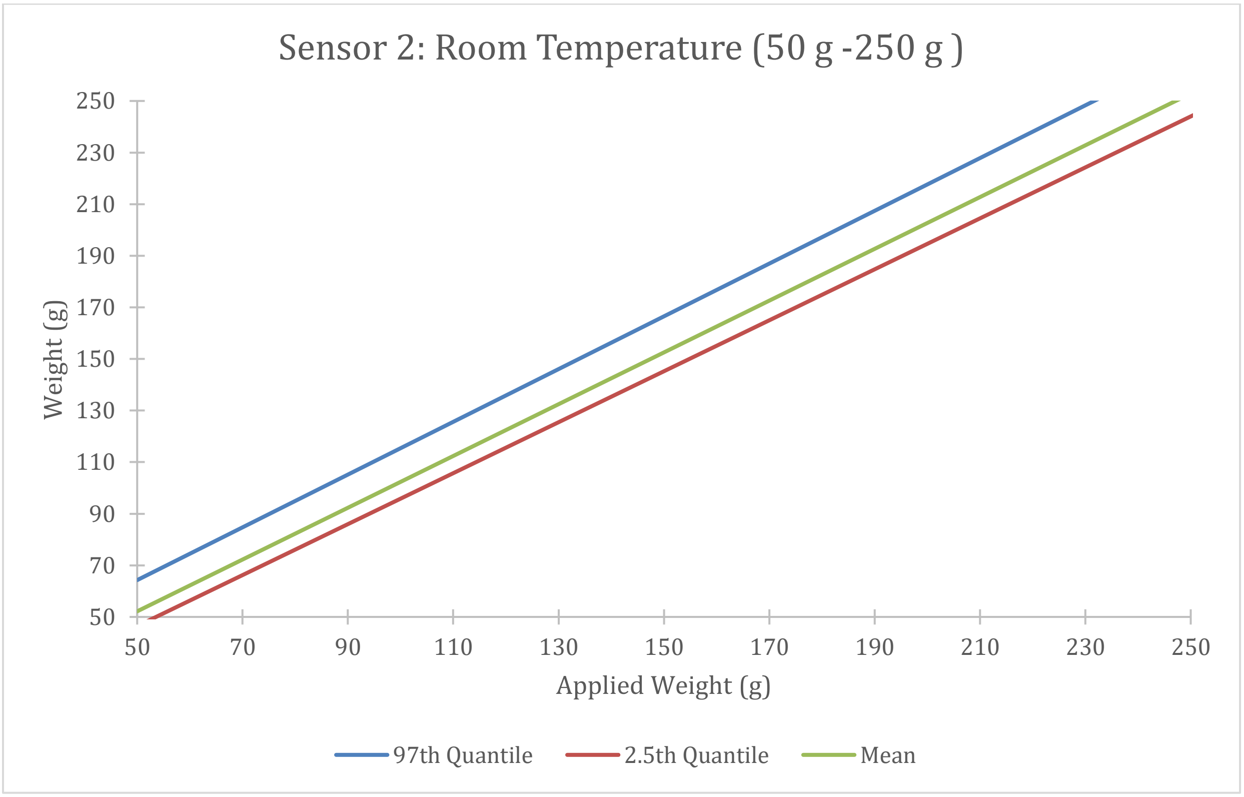

Sensor 1: Low Temperature (1000 g - 2500 g)
Weight (g)
2400
2200
2000
1800
1600
1400
1200
1000
1000
1200
1400
1600
1800
2000
2200
2400
Applied Weight (g)
97th Quantile
2.5th Quantile
Mean

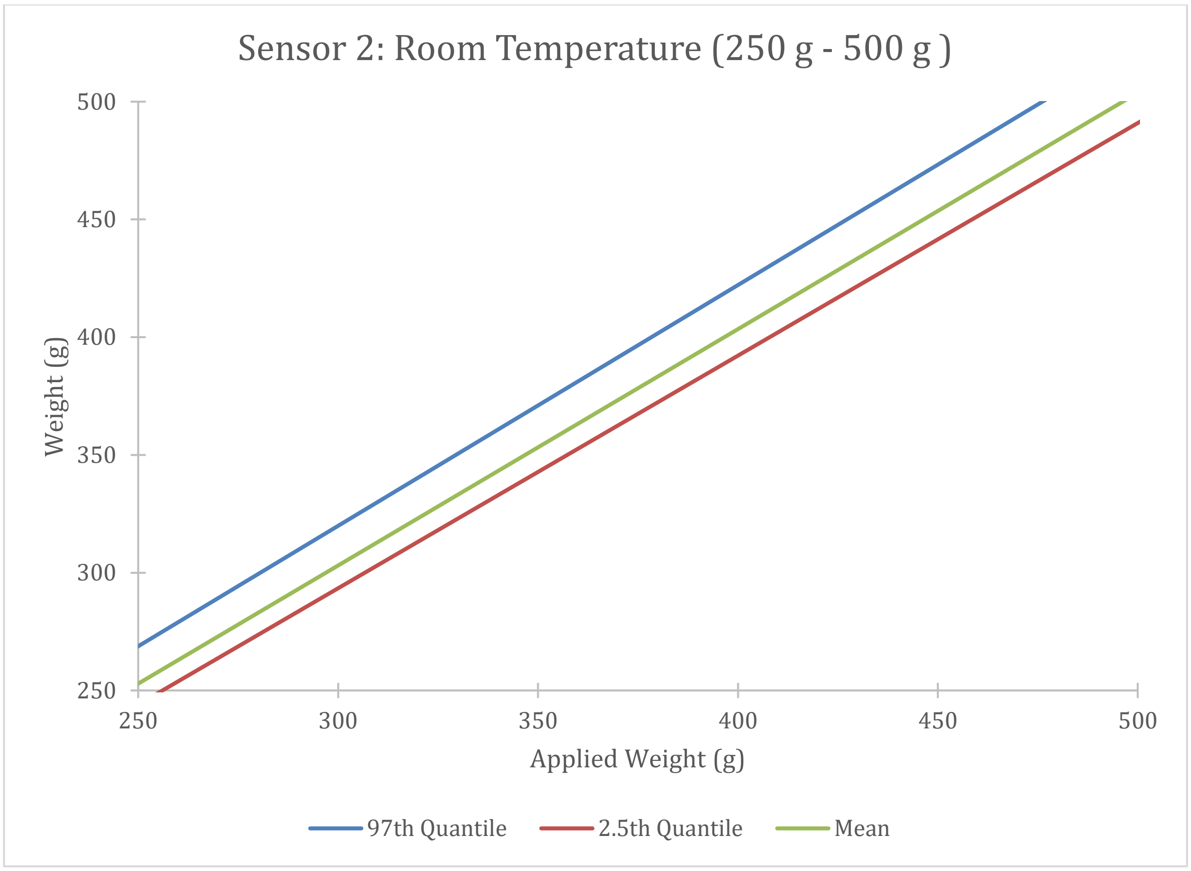

Sensor 1: Low Temperature (2500 g - 5000 g)
Weight (g)
5000
4500
4000
3500
3000
2500
2500
3000
3500
4000
4500
5000
Applied Weight (g)
97th Quantile
2.5th Quantile
Mean

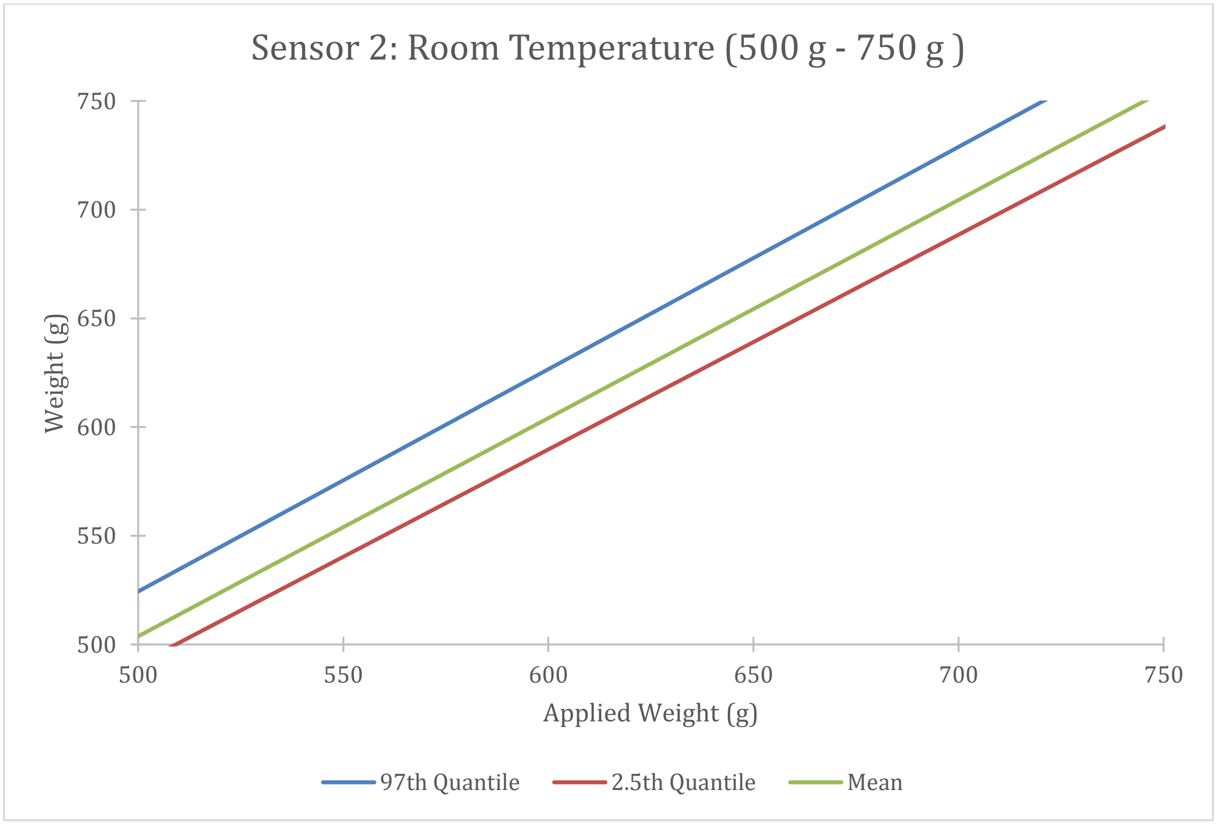
Sensor 2:

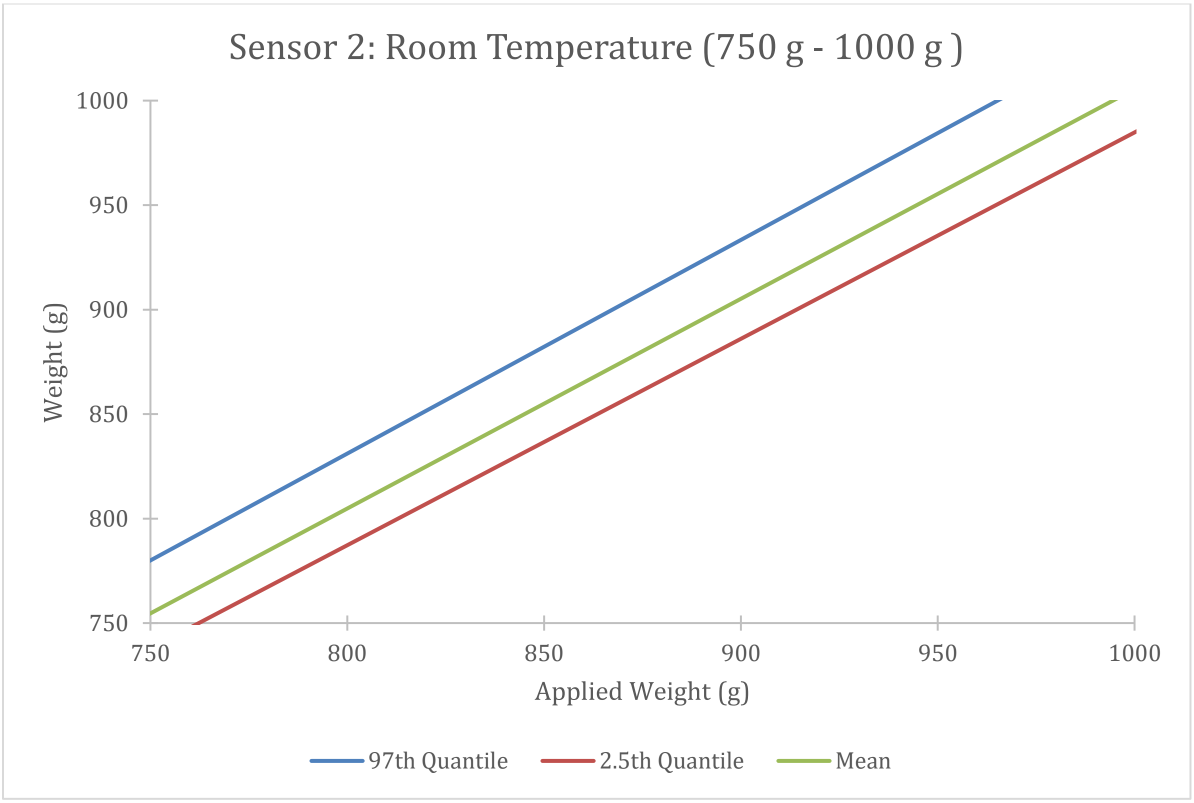
Room Temperature:

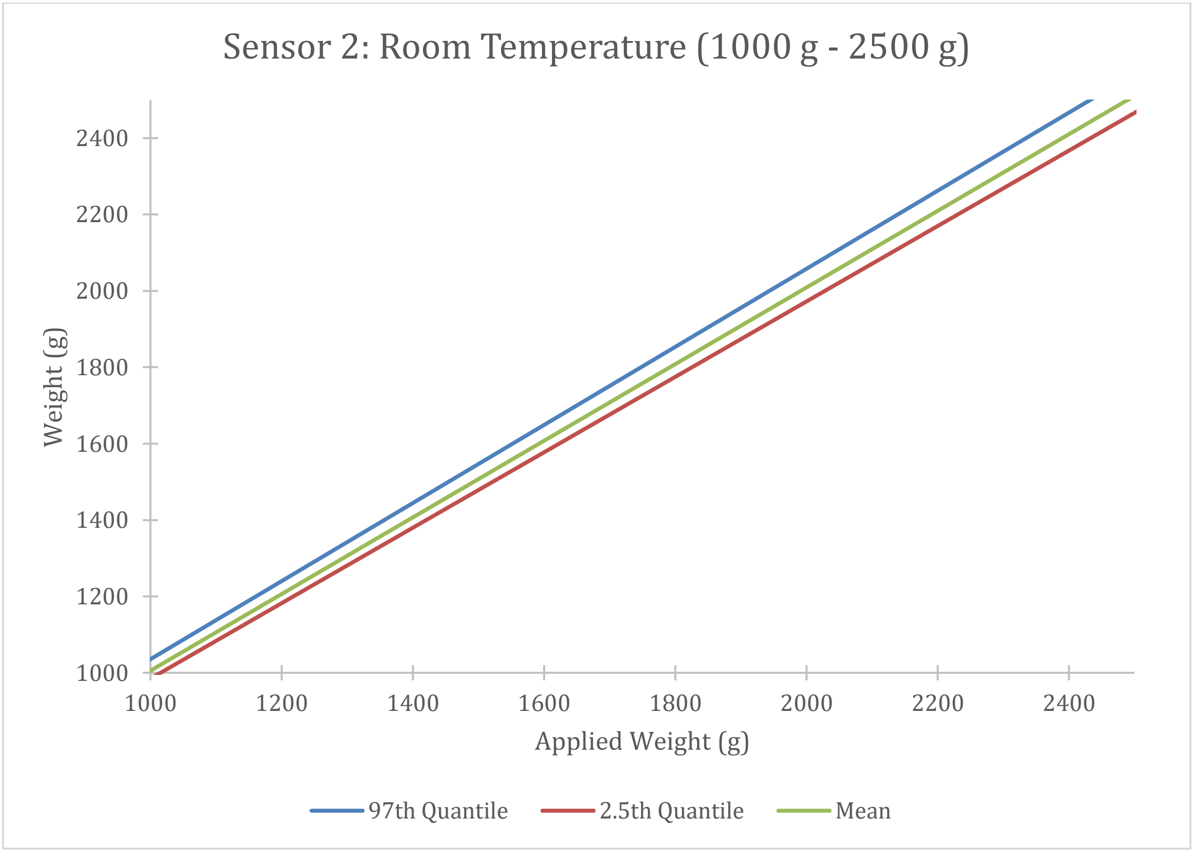


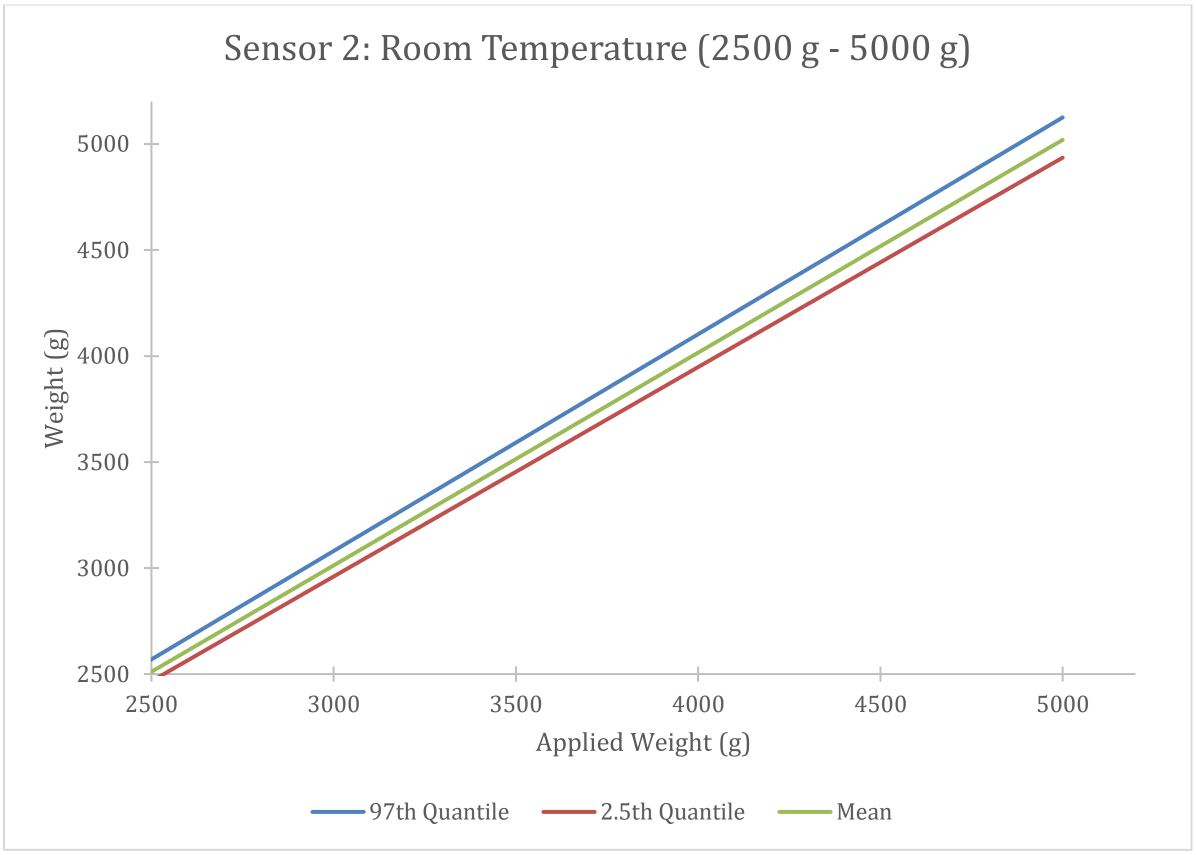



Sensor 2: Room Temperature (250 g - 500 g)
Weight (g)
Applied Weight (g)
250
300
350
400
450
500
97th Quantile
2.5th Quantile
Mean

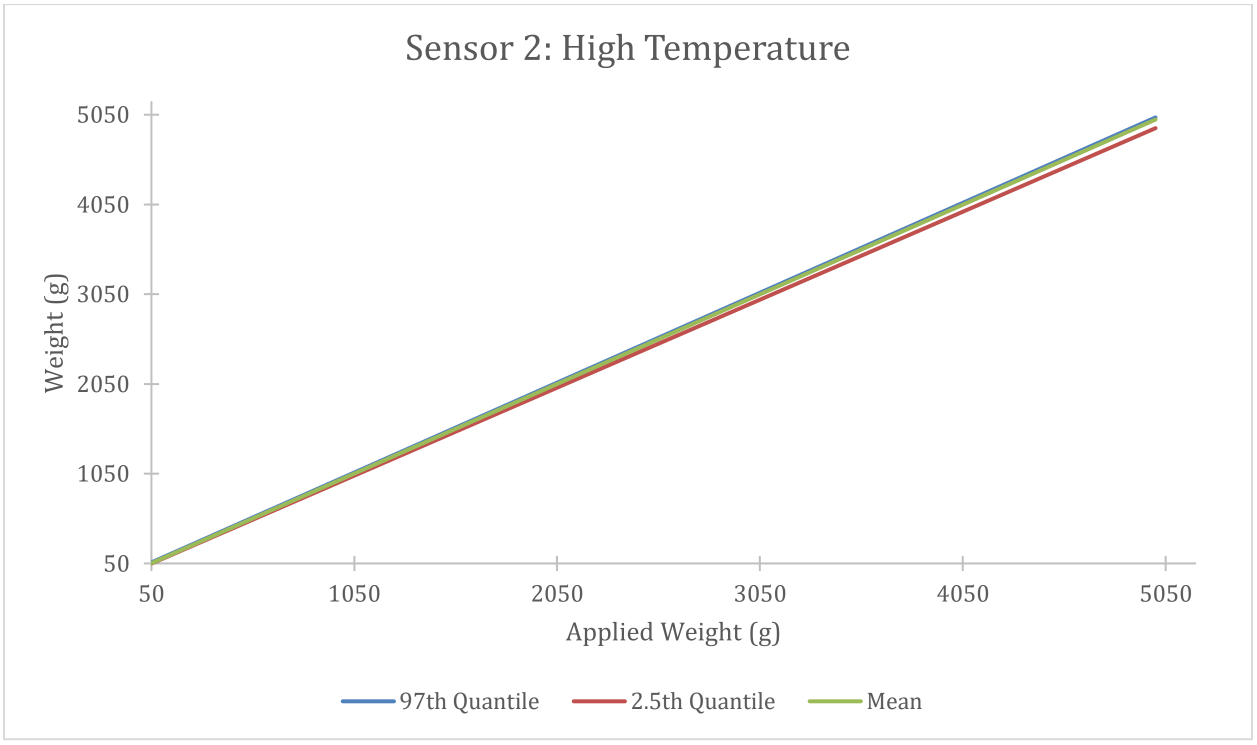

Sensor 2: Room Temperature (500 g - 750 g)
Weight (g)
Applied Weight (g)
500
550
600
650
700
750
97th Quantile
2.5th Quantile
Mean

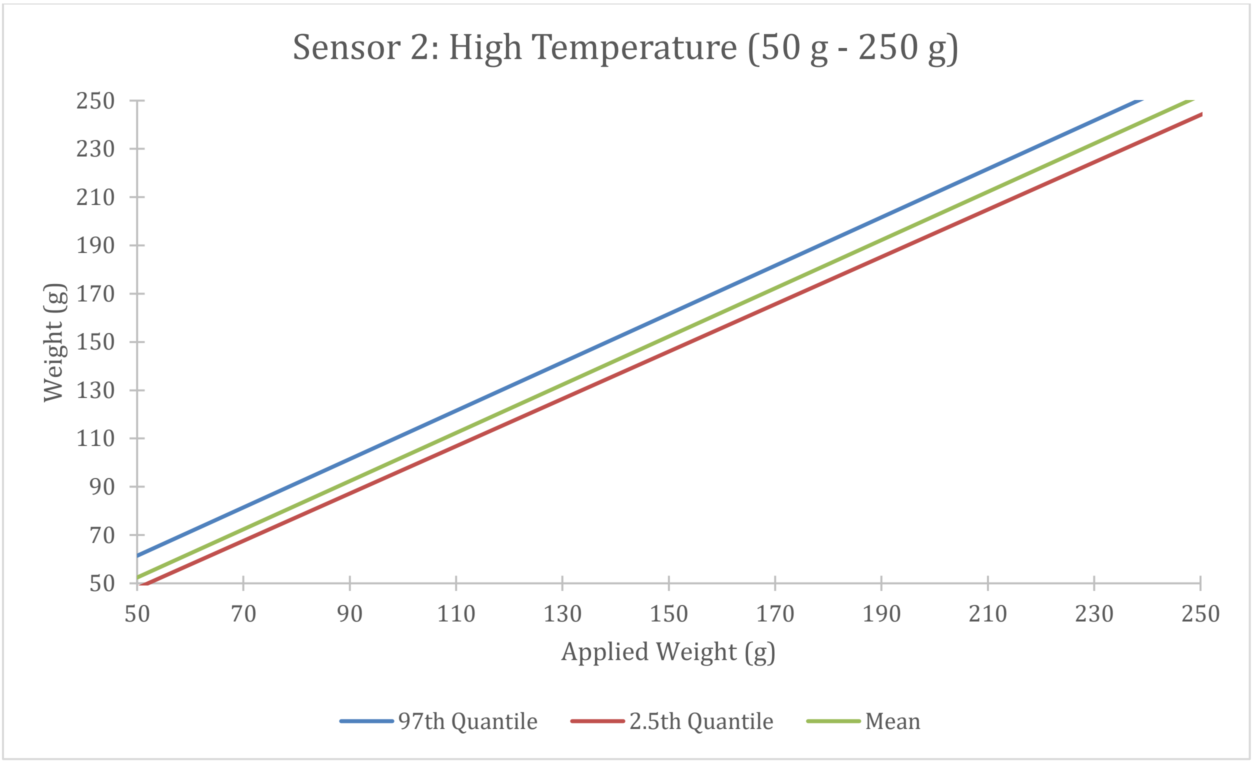

Sensor 2: Room Temperature (750 g - 1000 g)
Weight (g)
1000
950
900
850
800
750
750
800
850
900
950
1000
Applied Weight (g)
97th Quantile
2.5th Quantile
Mean

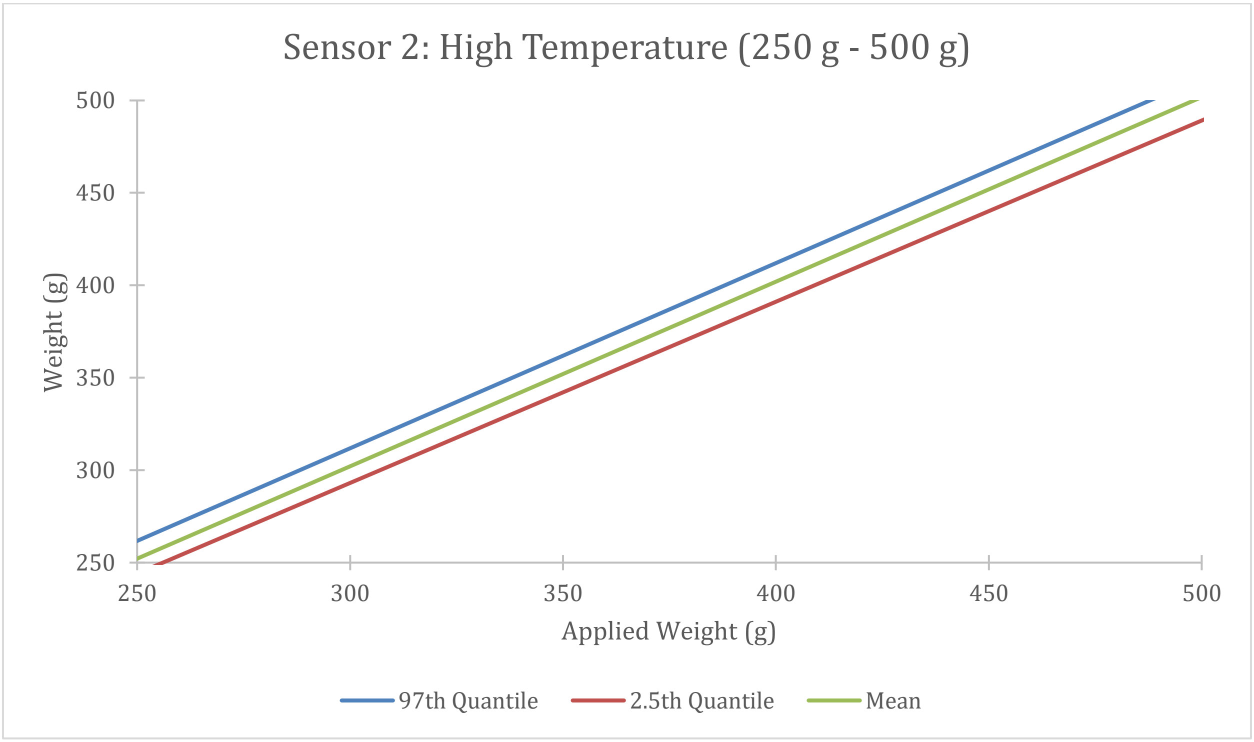

Sensor 2: Room Temperature (1000 g - 2500 g)
Weight (g)
2400
2200
2000
1800
1600
1400
1200
1000
1000
1200
1400
1600
1800
2000
2200
2400
Applied Weight (g)
97th Quantile
2.5th Quantile
Mean

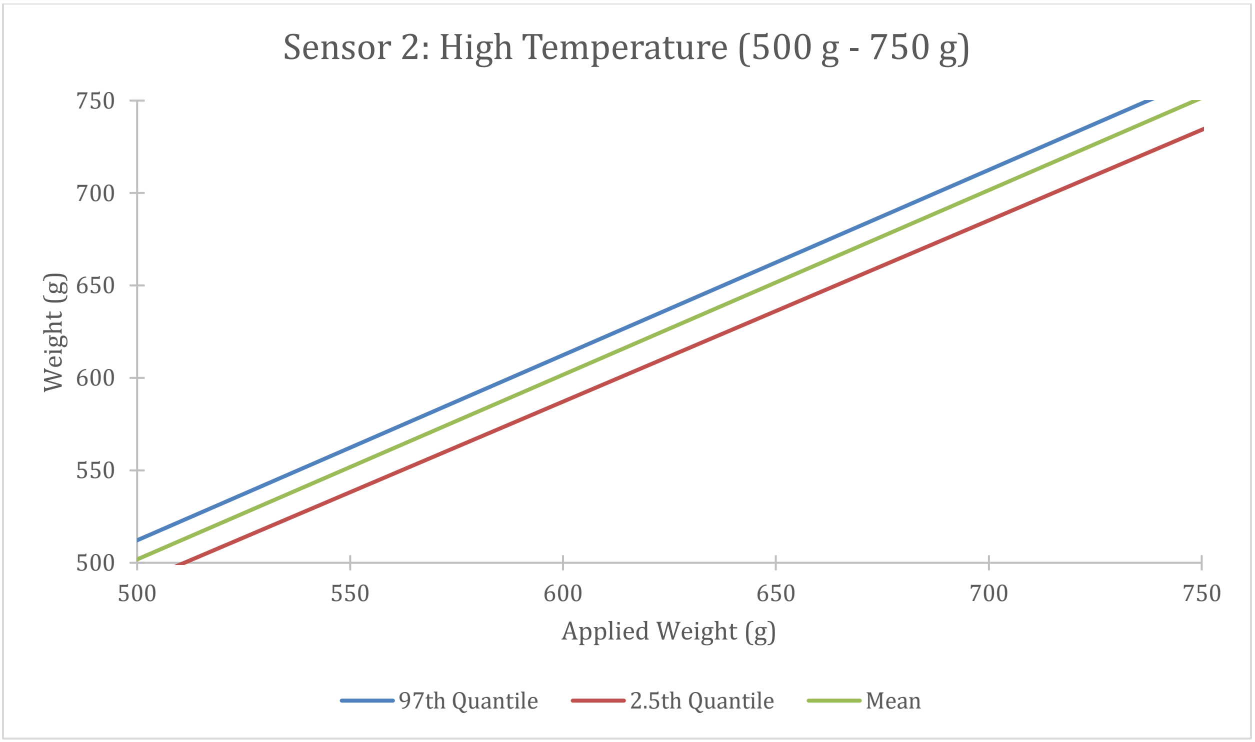


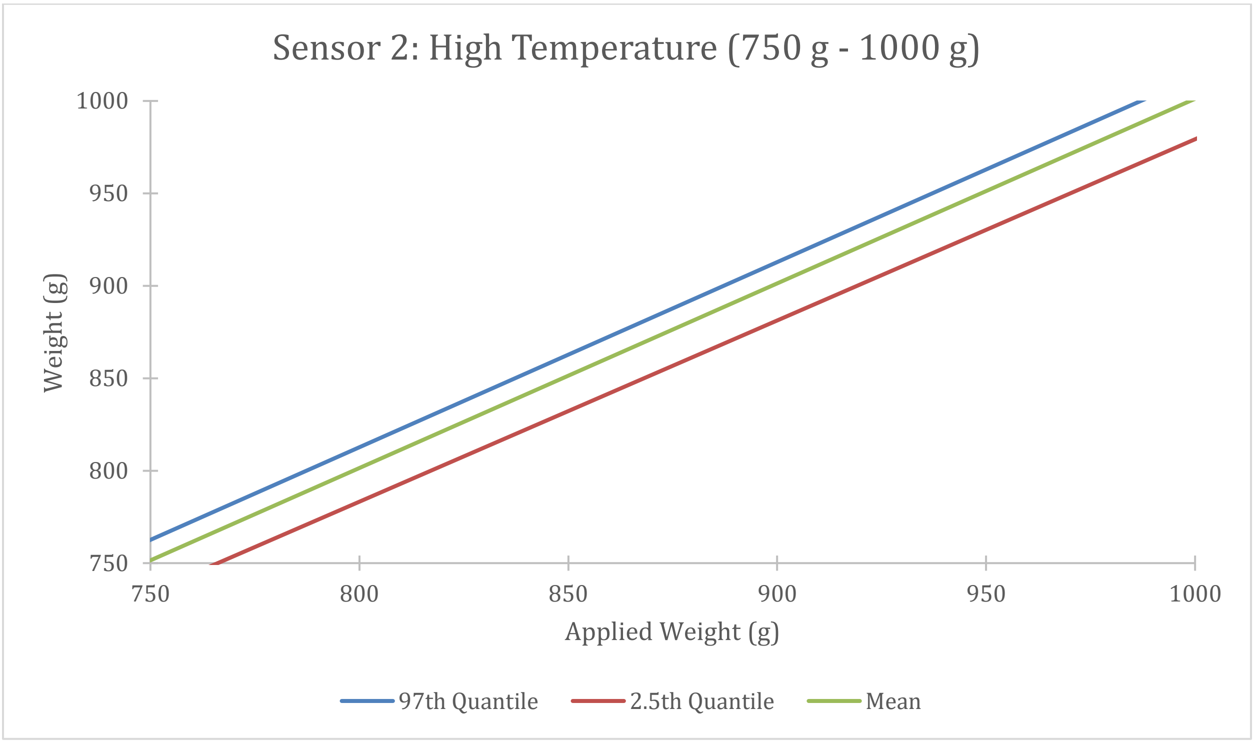
High Temperature:

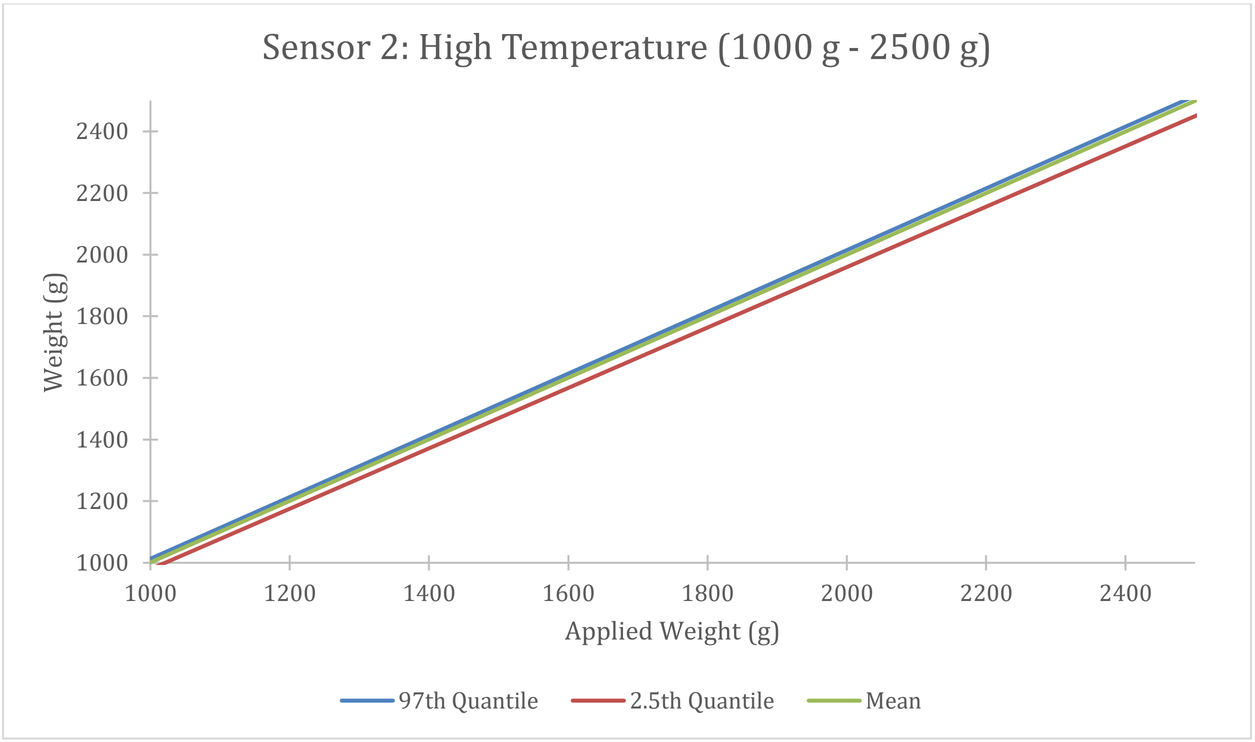


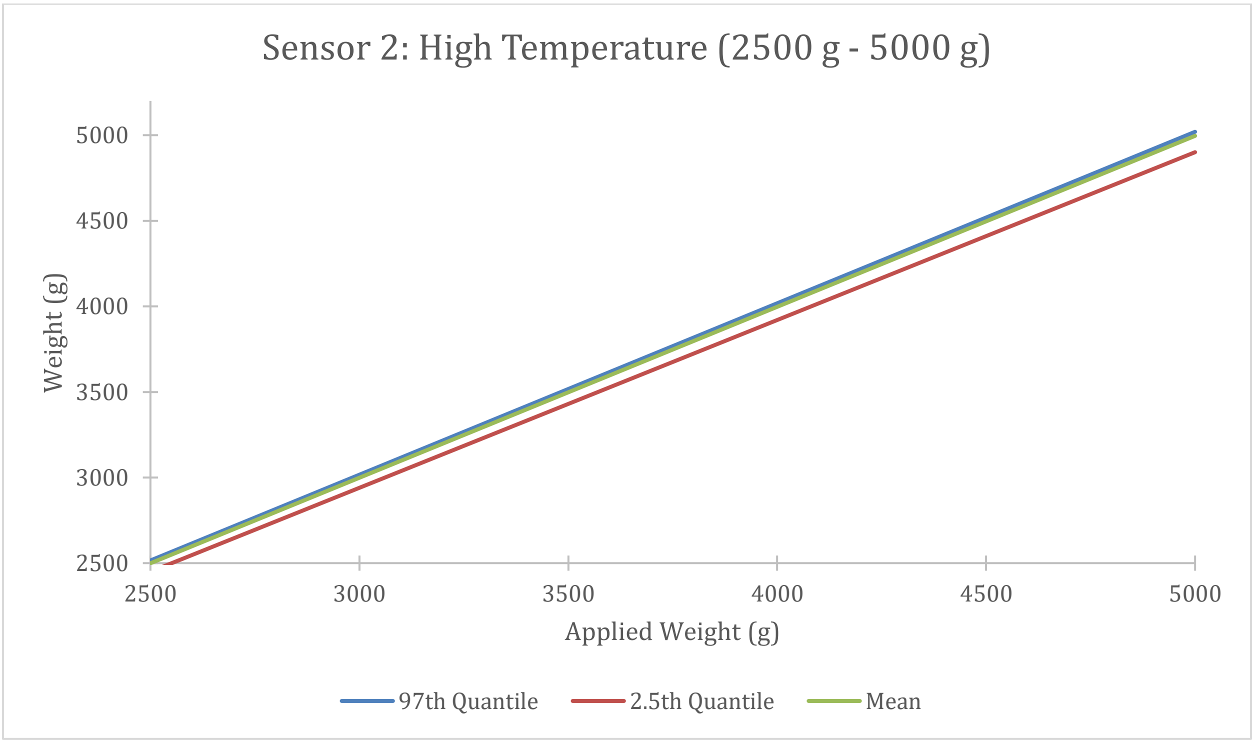

Sensor 2: High Temperature (50 g - 250 g)
Weight (g)
250
230
210
190
170
150
130
110
90
70
50
50
70
90
110
130
150
170
190
210
230
250
Applied Weight (g)
97th Quantile
2.5th Quantile
Mean


Sensor 2: High Temperature (250 g - 500 g)
Weight (g)
500
450
400
350
300
250
250
300
350
400
450
500
Applied Weight (g)
97th Quantile
2.5th Quantile
Mean

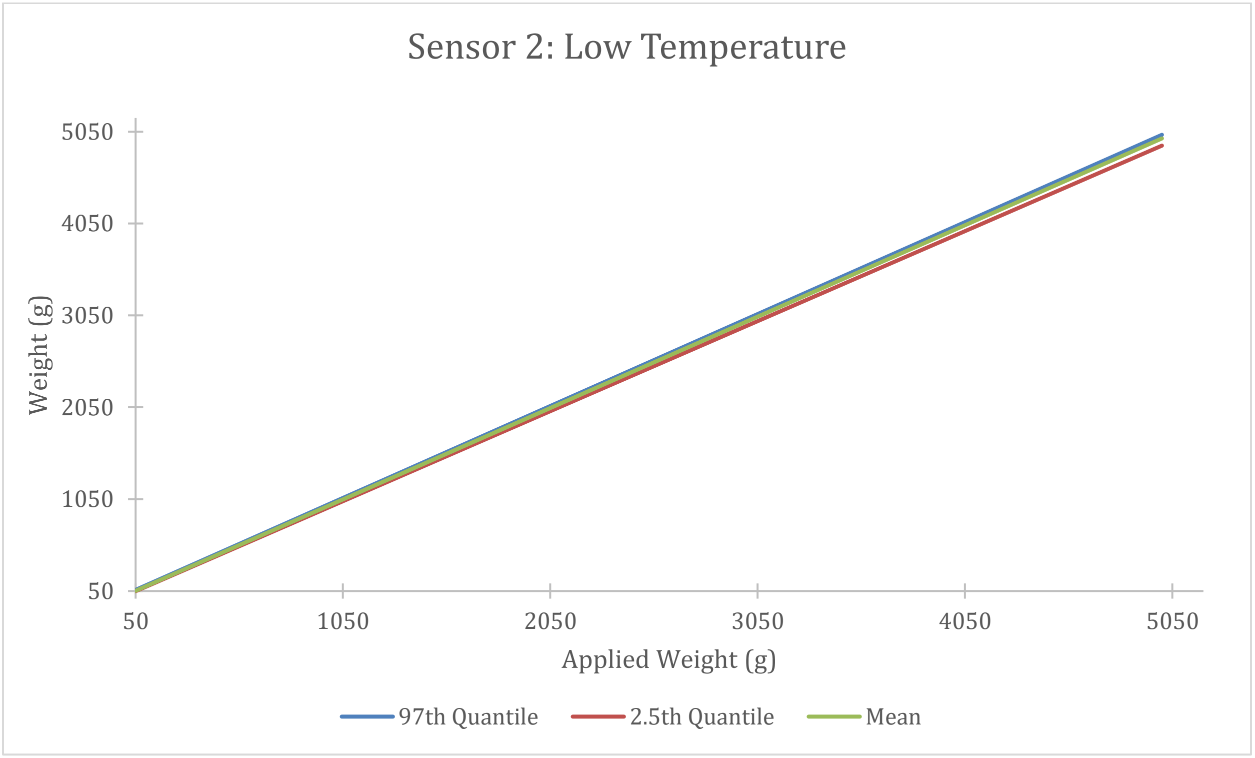

Sensor 2: High Temperature (500 g - 750 g)
Weight (g)
750
700
650
600
550
500
500
550
600
650
700
750
Applied Weight (g)
97th Quantile
2.5th Quantile
Mean

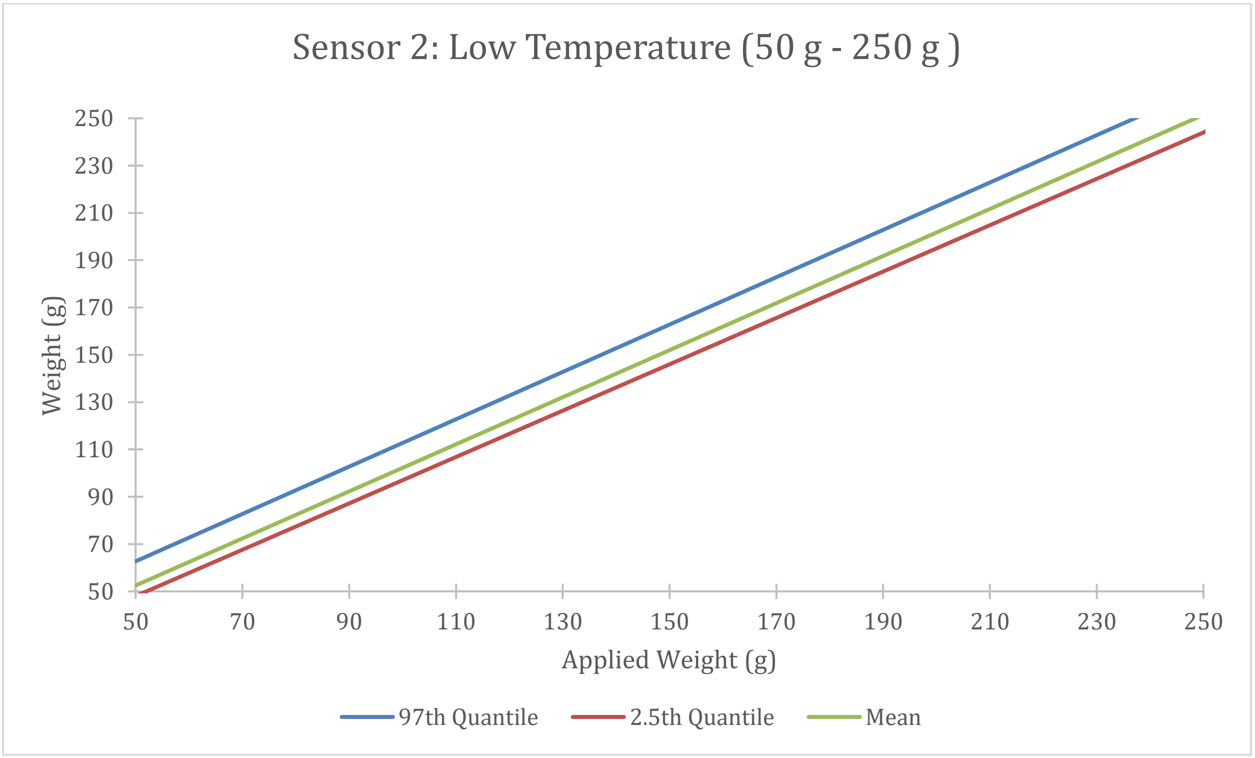

Sensor 2: High Temperature (750 g - 1000 g)
Weight (g)
1000
950
900
850
800
750
750
800
850
900
950
1000
Applied Weight (g)
97th Quantile
2.5th Quantile
Mean

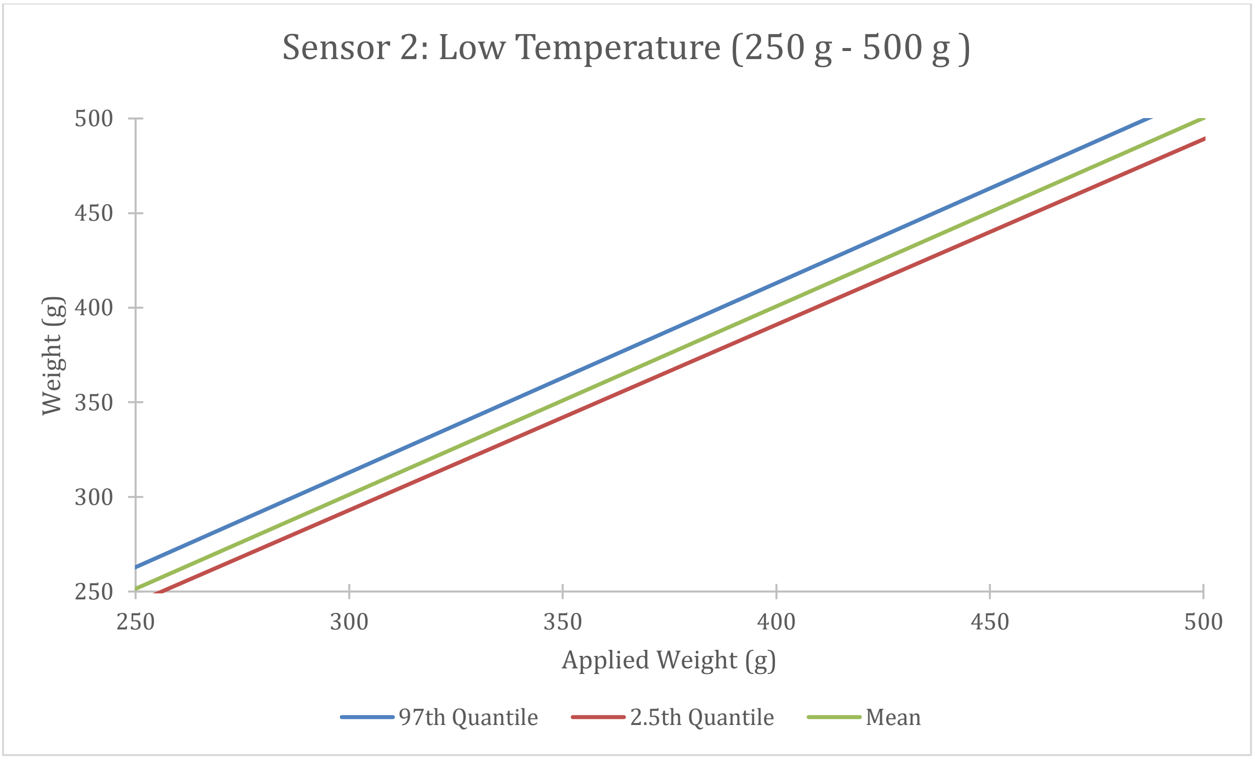

Sensor 2: High Temperature (1000 g - 2500 g)
Weight (g)
2400
2200
2000
1800
1600
1400
1200
1000
1000
1200
1400
1600
1800
2000
2200
2400
Applied Weight (g)
97th Quantile
2.5th Quantile
Mean

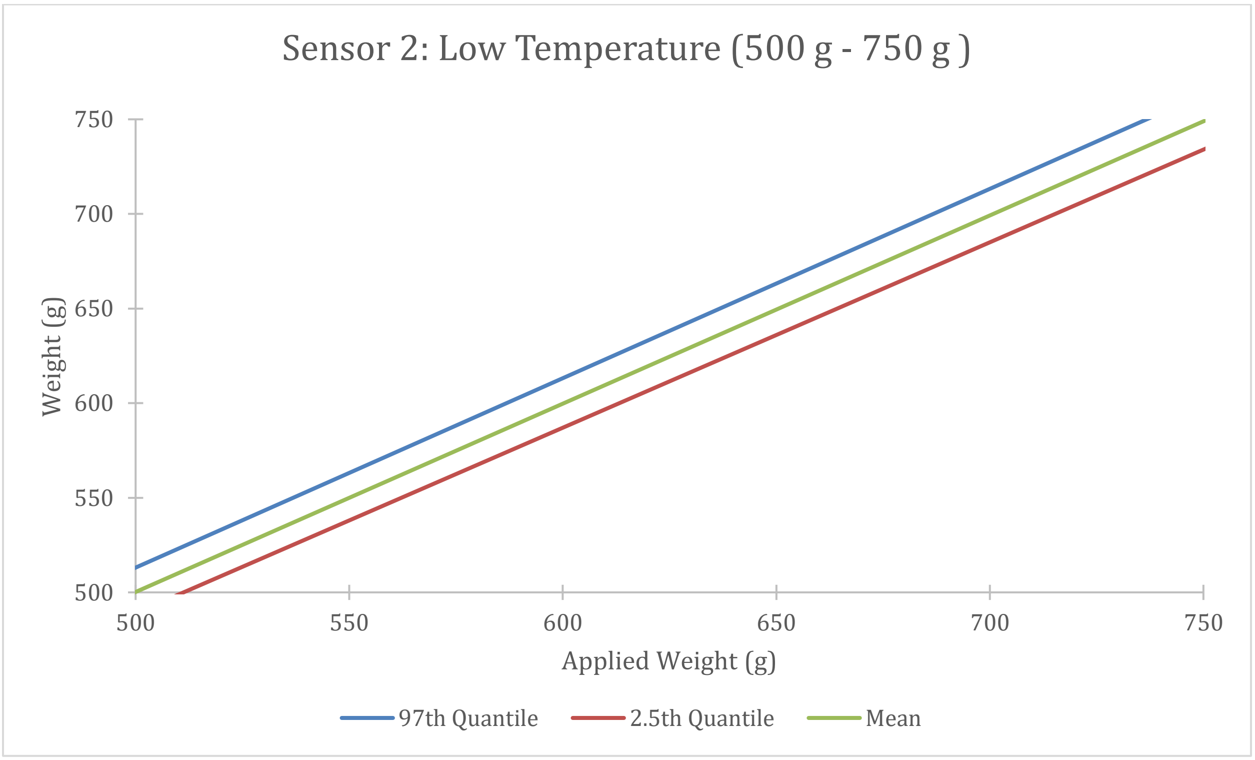

Sensor 2: High Temperature (2500 g - 5000 g)
Weight (g)
5000
4500
4000
3500
3000
2500
2500
3000
3500
4000
4500
5000
Applied Weight (g)
97th Quantile
2.5th Quantile
Mean

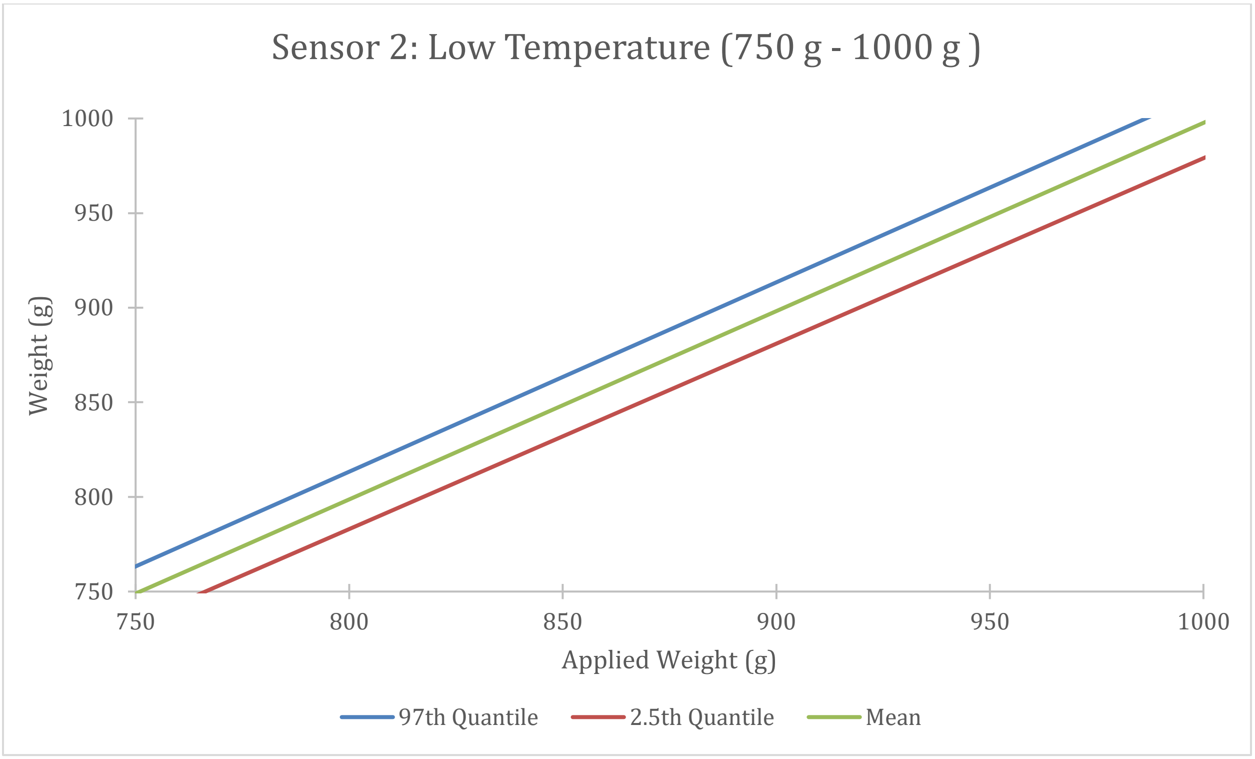
Low Temperature:

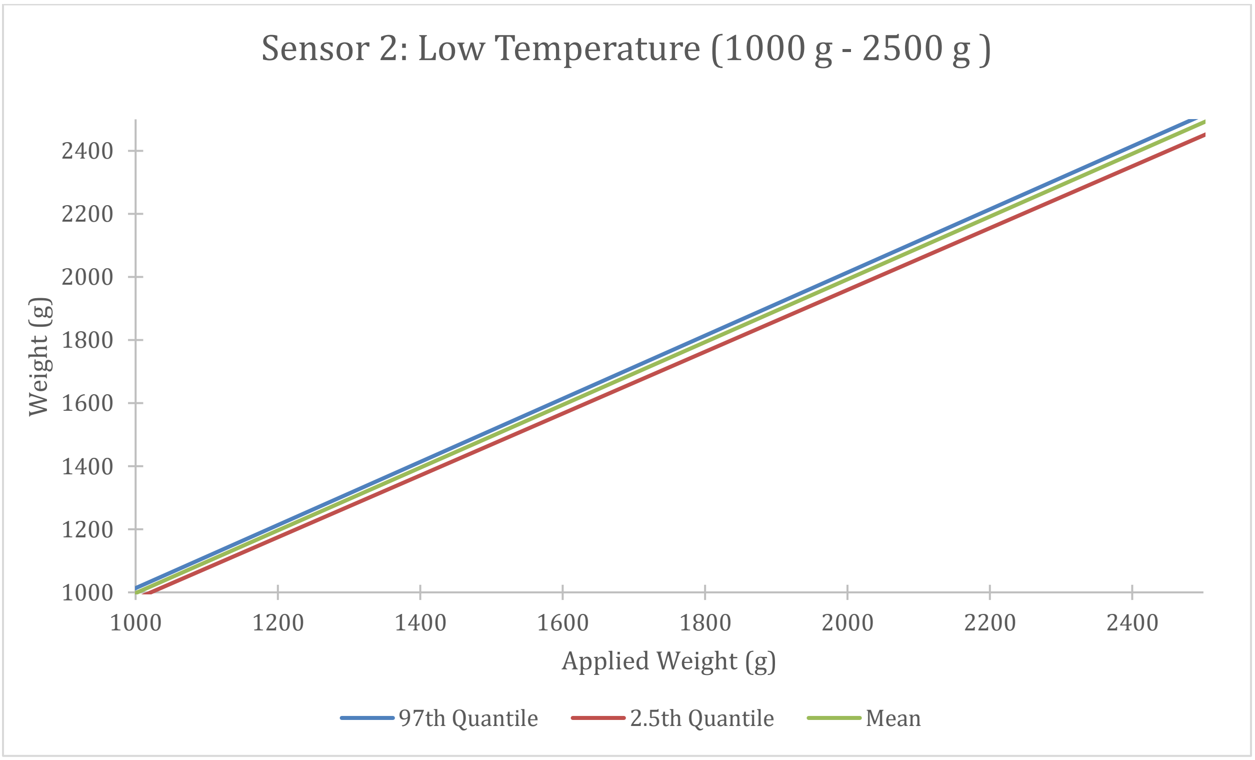


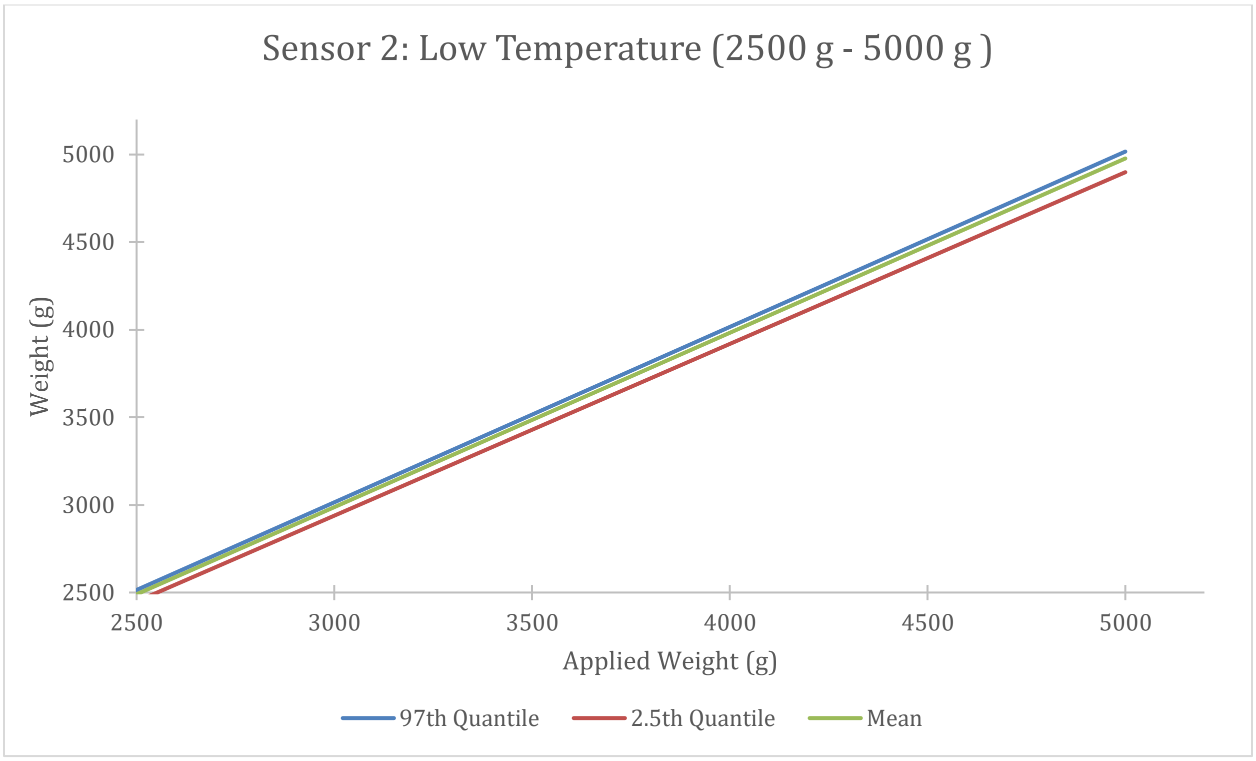



Sensor 2: Low Temperature (250 g - 500 g)
500
450
400
350
300
250
Weight (g)
250
300
350
400
450
500
Applied Weight (g)
97th Quantile
2.5th Quantile
Mean


Sensor 2: Low Temperature (500 g - 750 g)
750
700
650
600
550
500
Weight (g)
500
550
600
650
700
750
Applied Weight (g)
97th Quantile
2.5th Quantile
Mean


Sensor 2: Low Temperature (750 g - 1000 g)
1000
950
900
850
800
750
Weight (g)
750
800
850
900
950
1000
Applied Weight (g)
97th Quantile
2.5th Quantile
Mean


Sensor 2: Low Temperature (1000 g - 2500 g)
2400
2200
2000
1800
1600
1400
1200
1000
Weight (g)
1000
1200
1400
1600
1800
2000
2200
2400
Applied Weight (g)
97th Quantile
2.5th Quantile
Mean

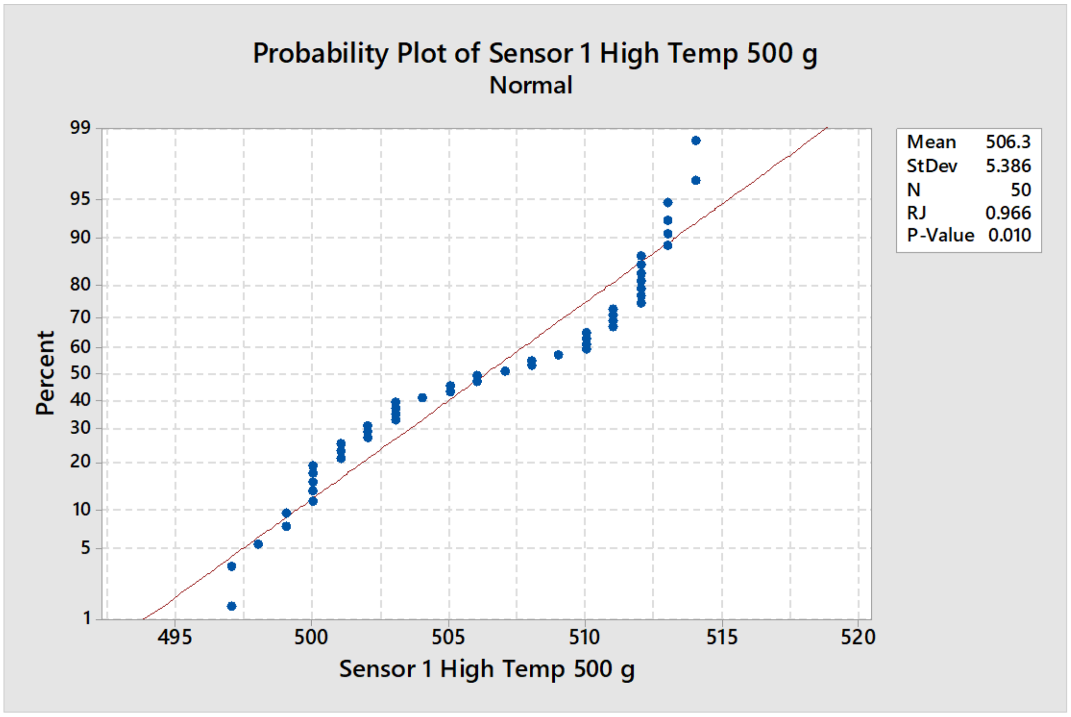


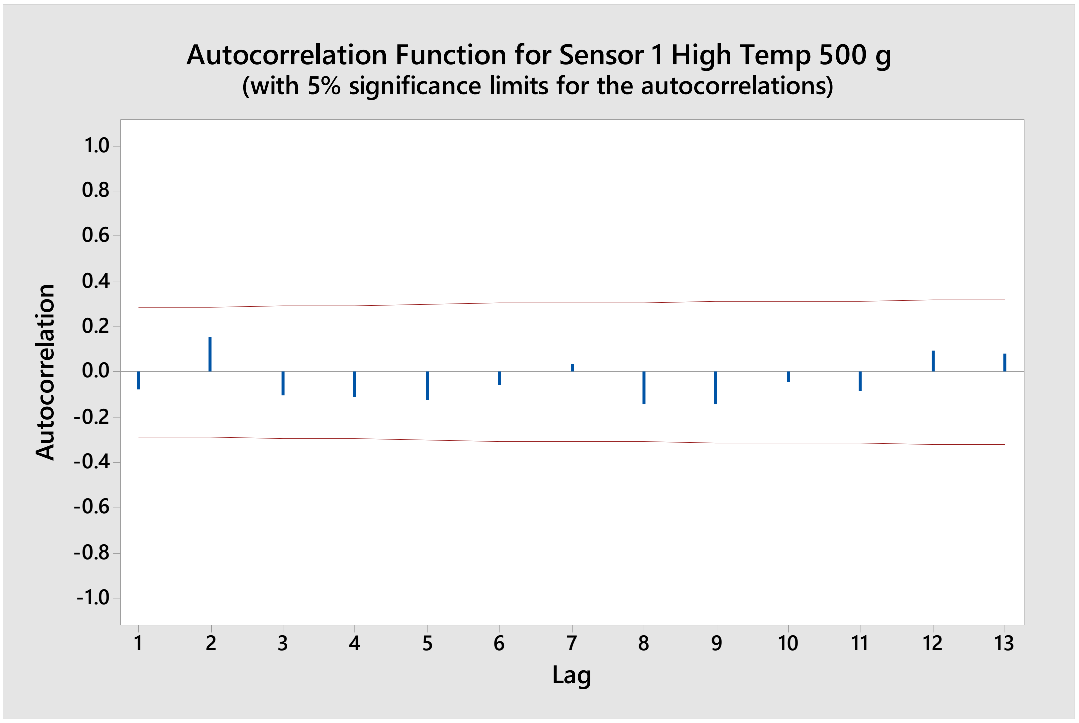
Sensor 2: Low Temperature (2500 g - 5000 g)
Weight (g)
5000
4500
4000
3500
3000
2500
2500
3000
3500
4000
4500
5000
Applied Weight (g)
97th Quantile
2.5th Quantile
Mean

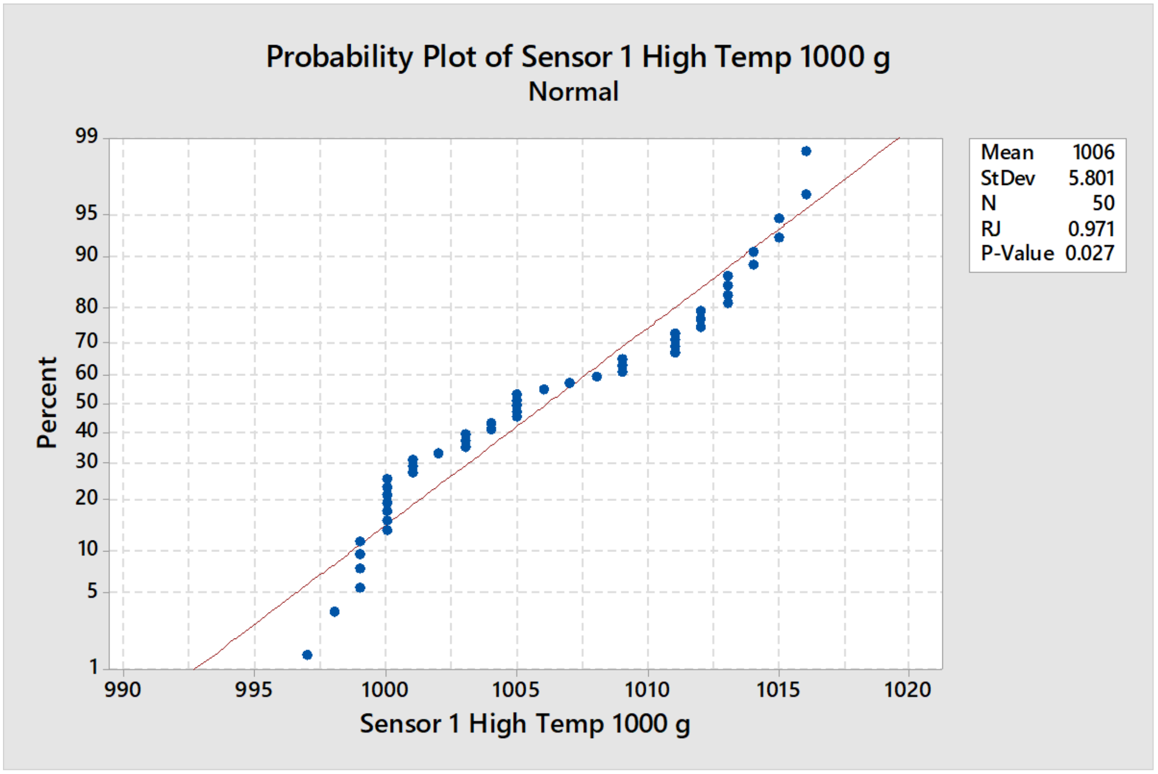
Normality Proof

Sensor 1:

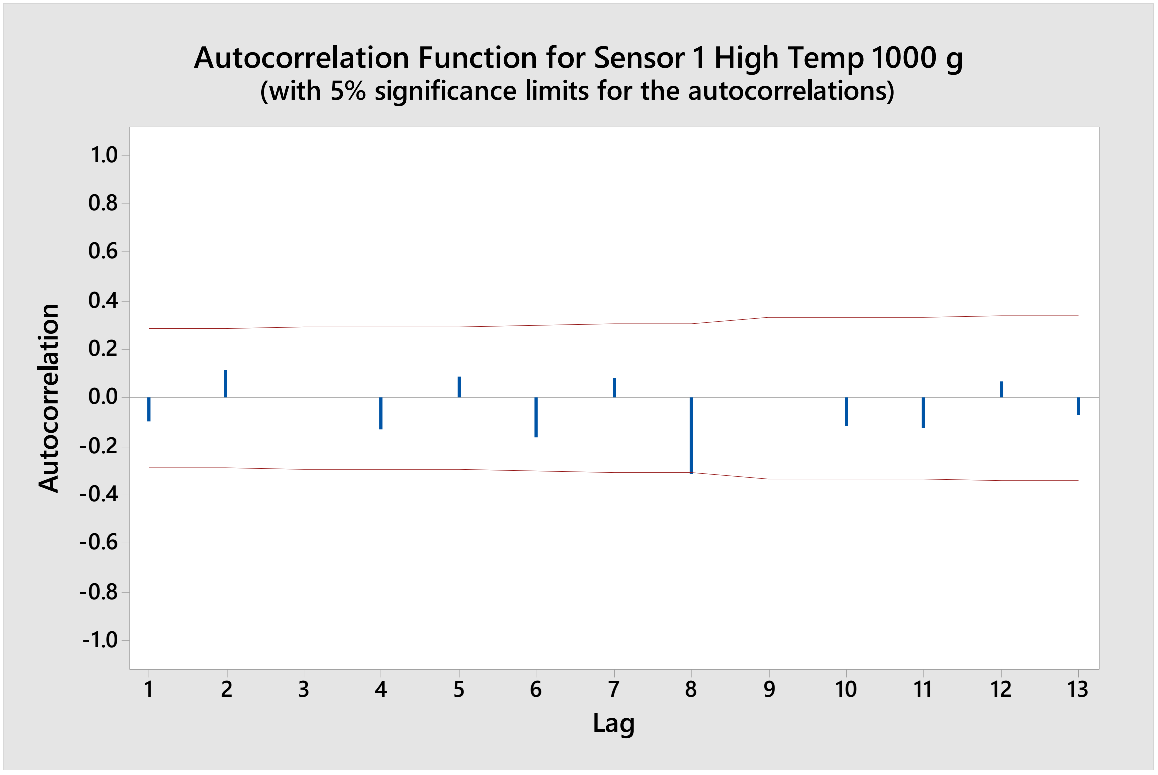
High Temperature



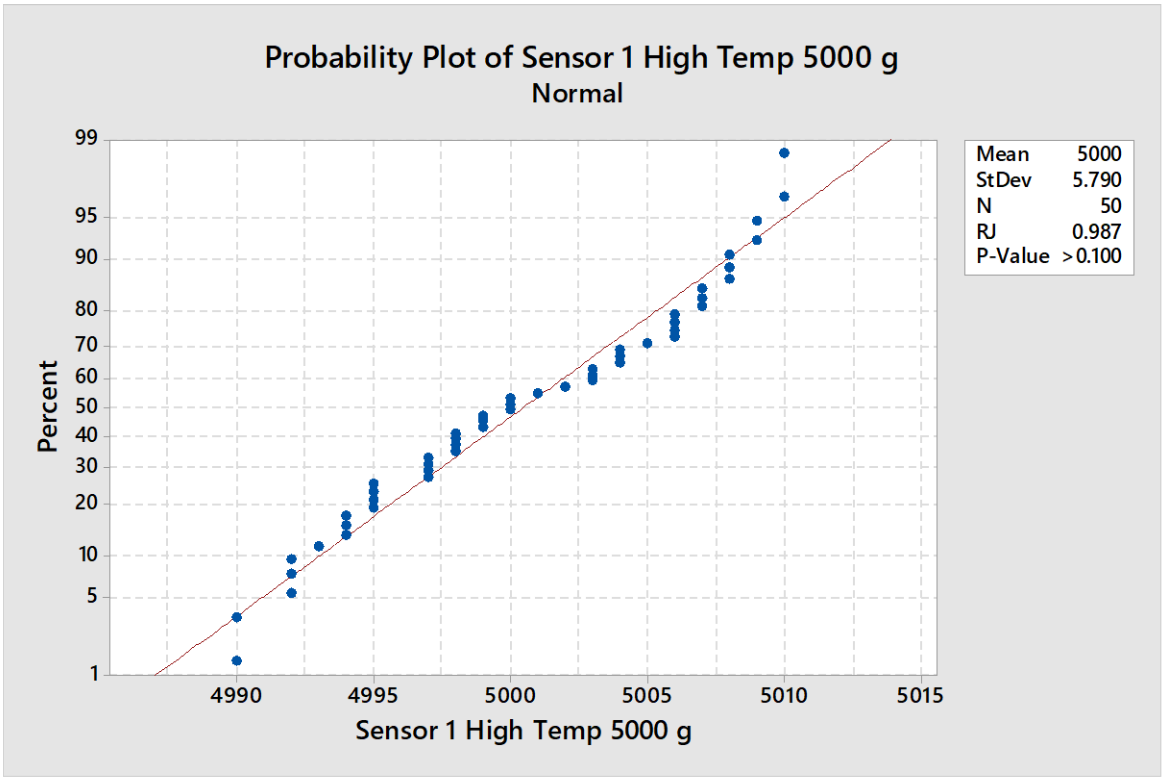


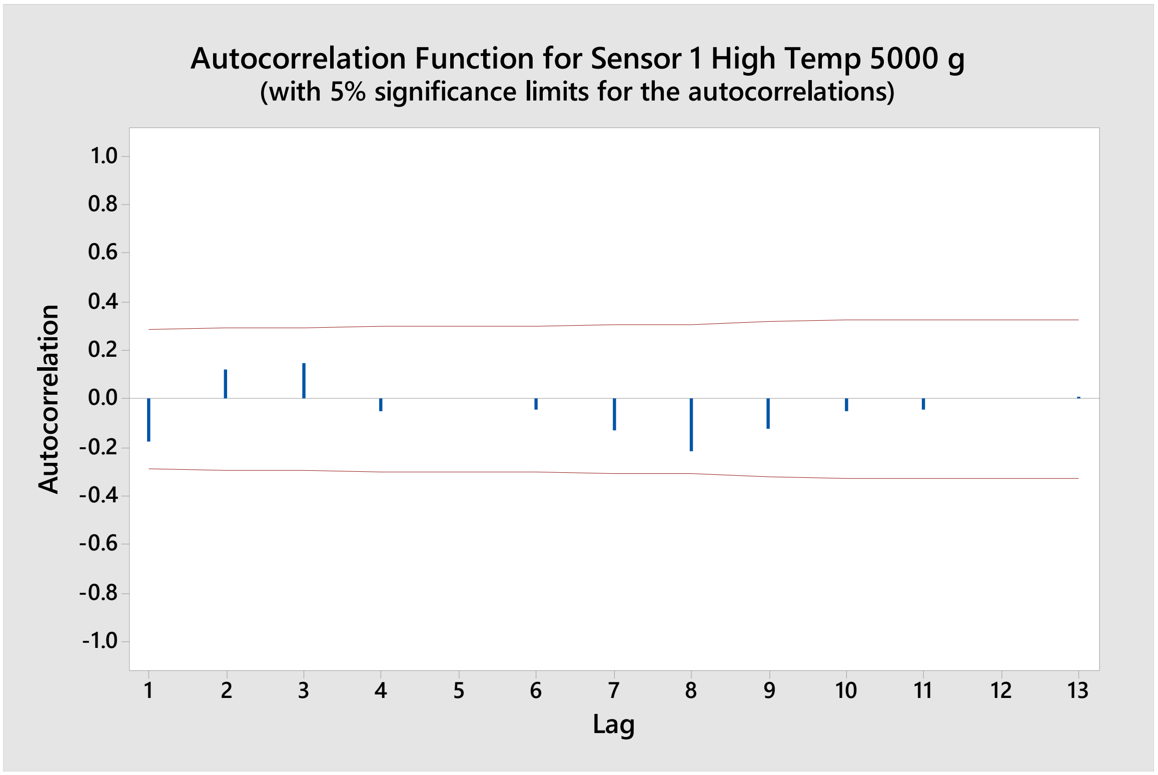


Probability Plot of Sensor 1 High Temp 1000 g
Normal
Percent
99
95
90
80
70
60
50
40
30
20
10
5
1
990
995
1000
1005
1010
1015
1020
Sensor 1 High Temp 1000 g
Mean 1006
StDev 5.801
N 50
RJ 0.971
P-Value 0.027

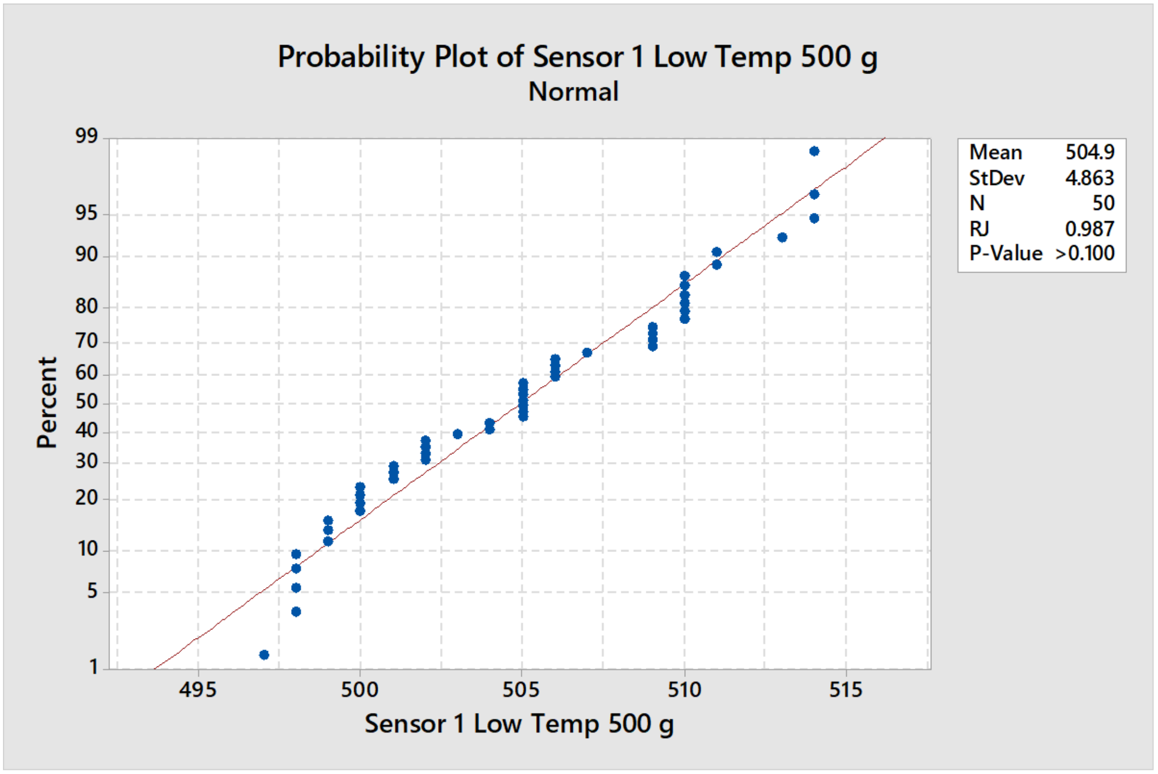


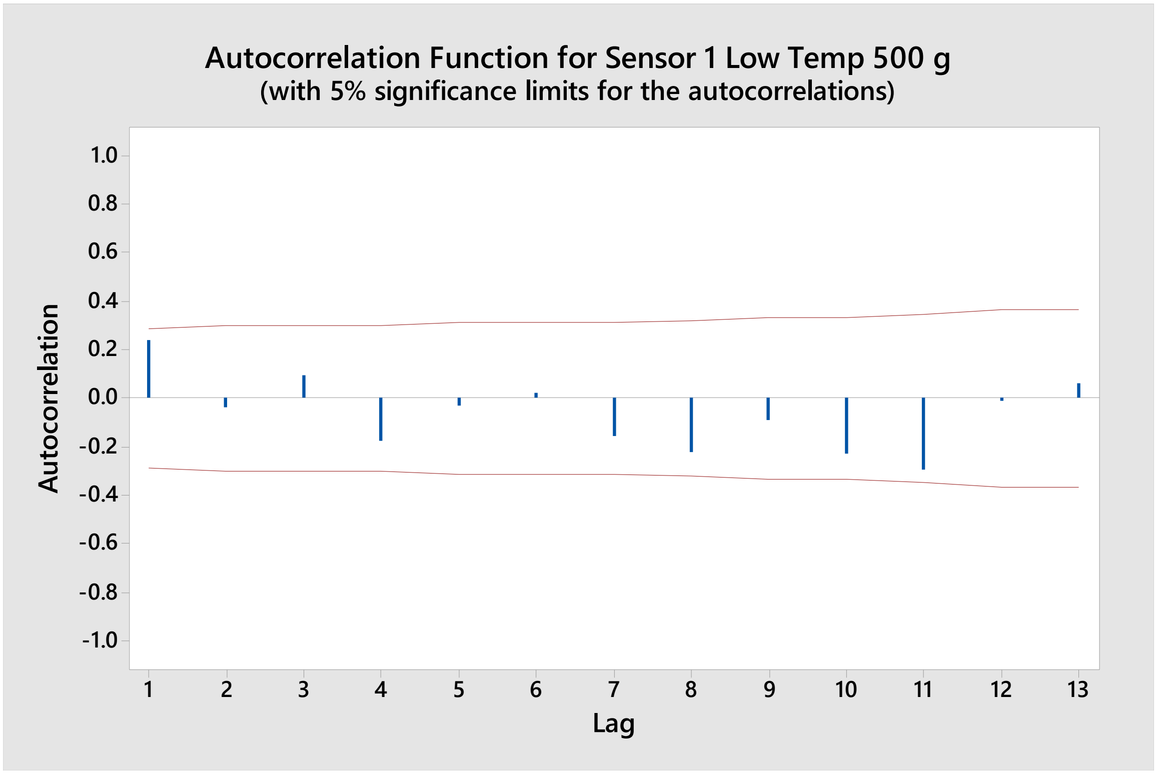
Autocorrelation Function for Sensor 1 High Temp 1000 g
(with 5% significance limits for the autocorrelations)
Autocorrelation
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-0.8
-1.0
1
2
3
4
5
6
7
8
9
10
11
12
13
Lag

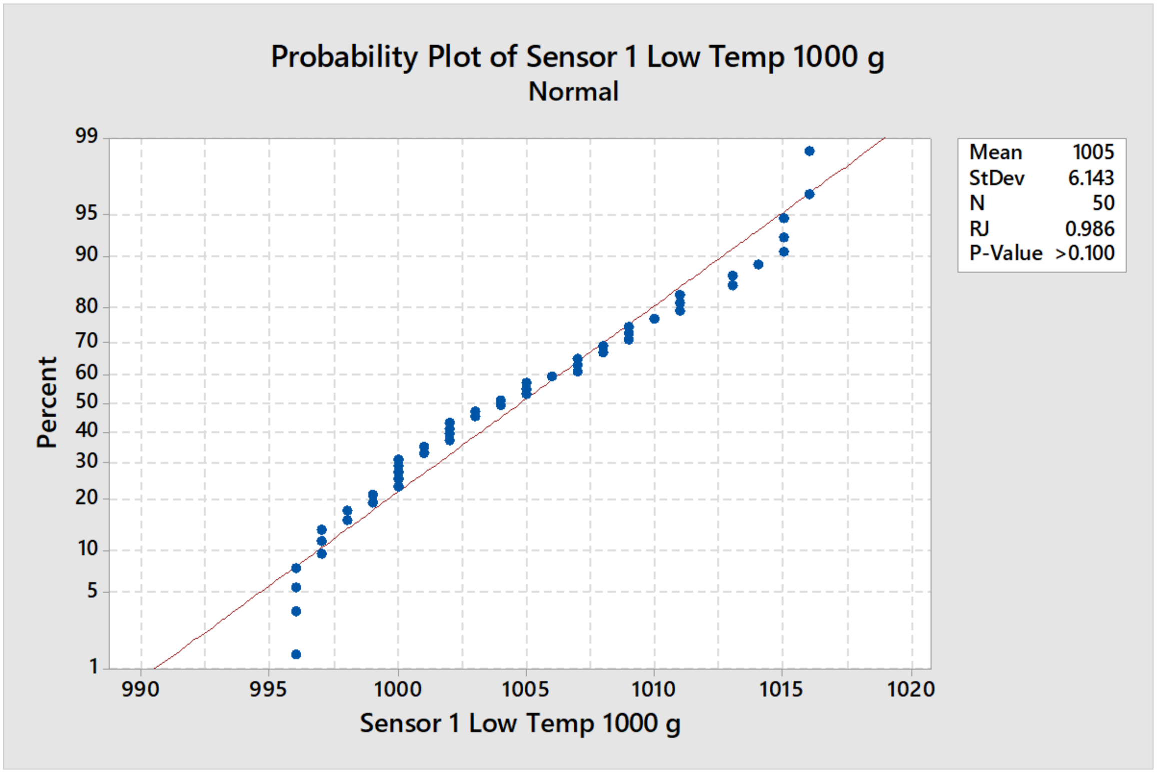


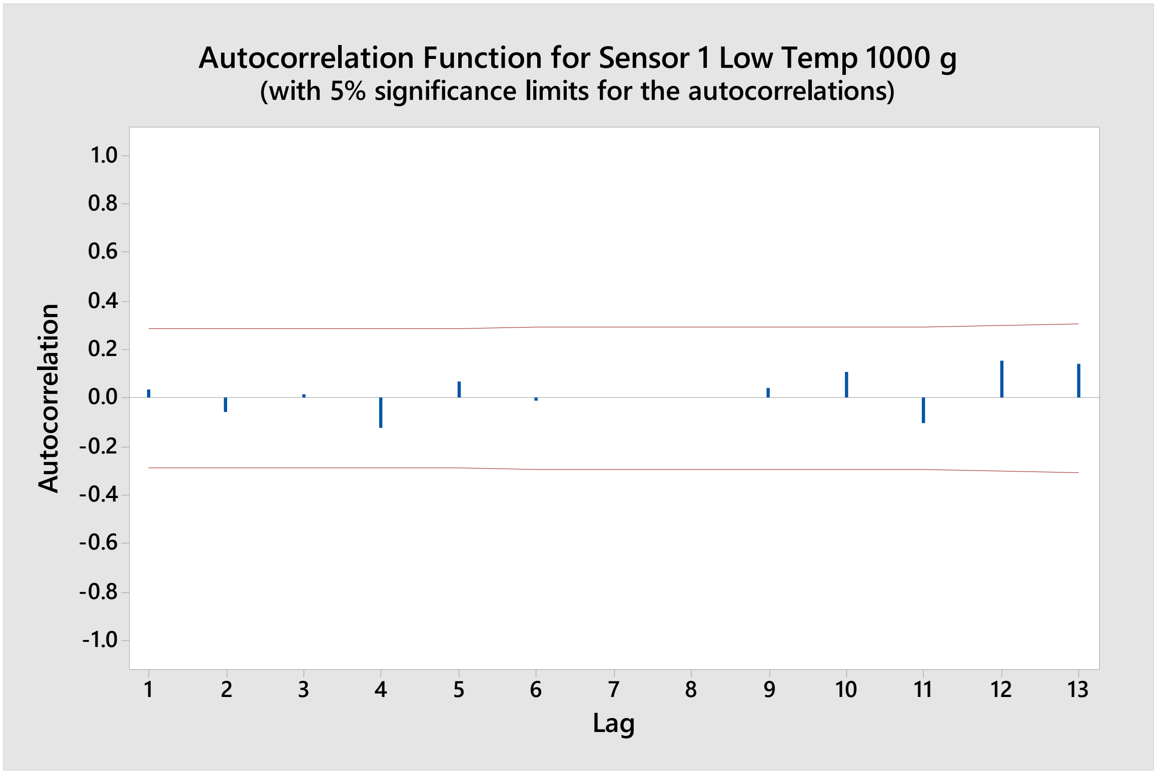


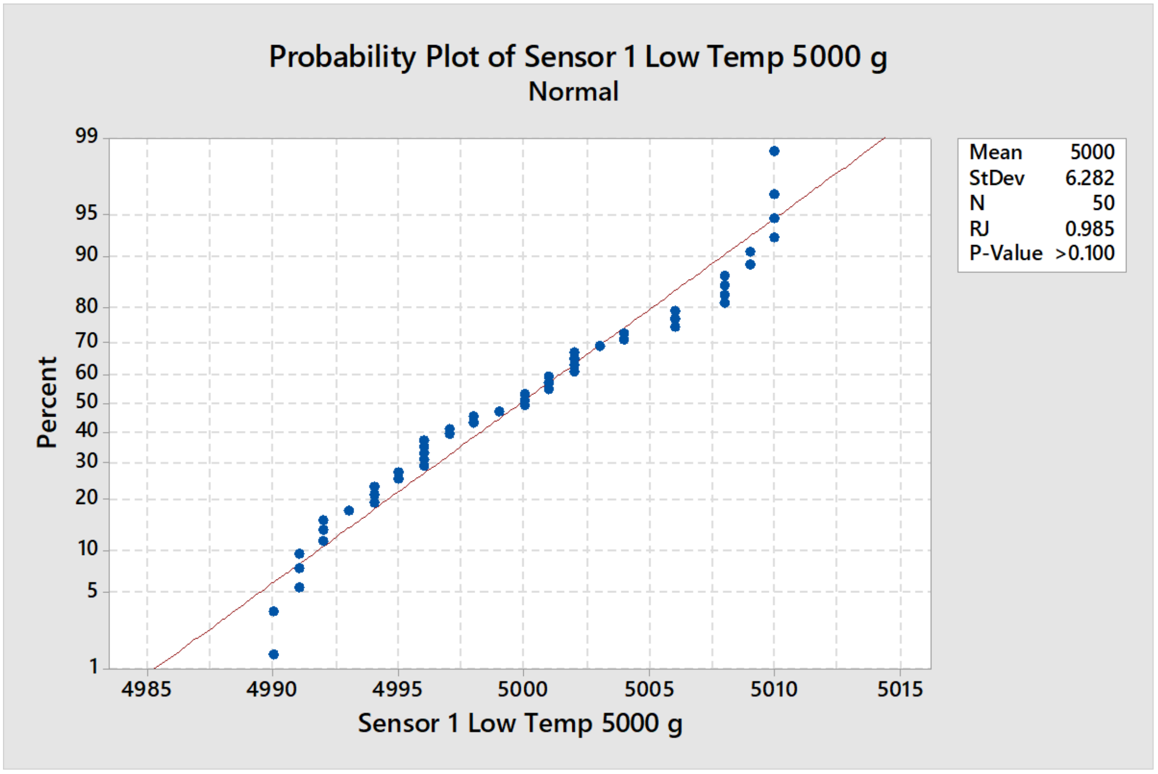
Low Temperature

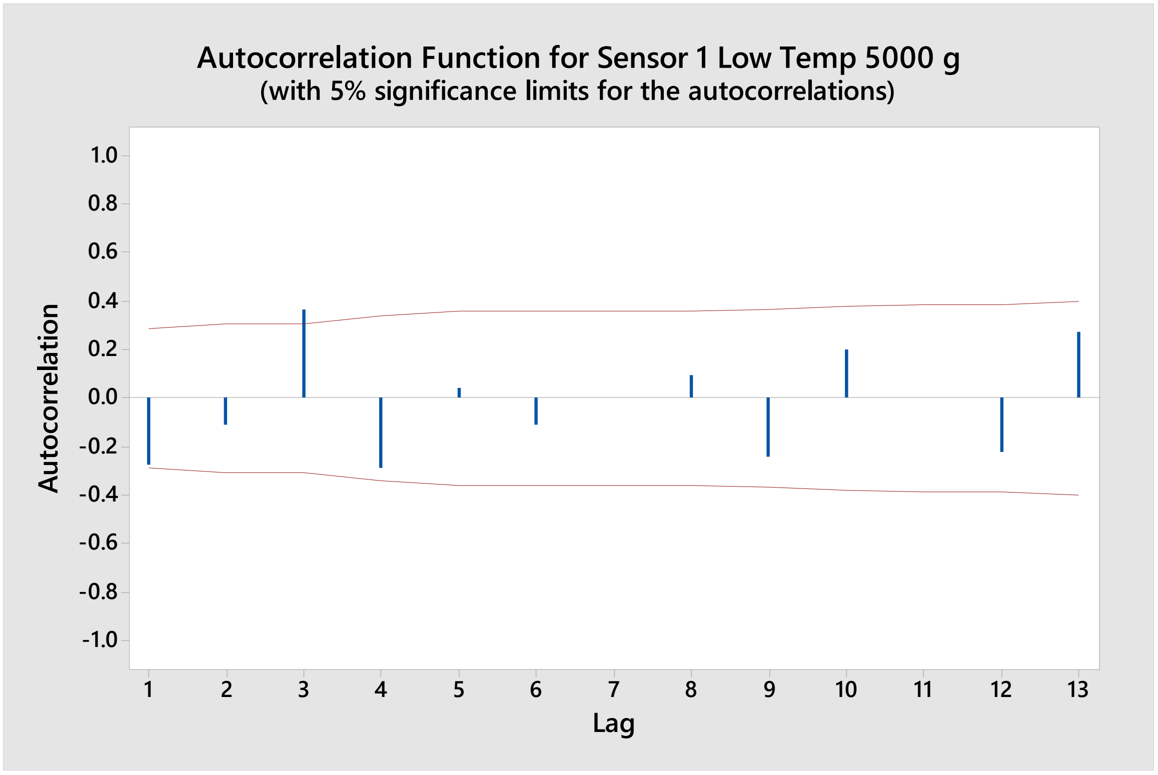

Probability Plot of Sensor 1 Low Temp 500 g
Normal
Percent
99
95
90
80
70
60
50
40
30
20
10
5
1
495
500
505
510
515
Sensor 1 Low Temp 500 g
Mean 504.9
StDev 4.863
N 50
RJ 0.987
P-Value >0.100


Autocorrelation Function for Sensor 1 Low Temp 500 g
(with 5% significance limits for the autocorrelations)
Autocorrelation
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-0.8
-1.0
1
2
3
4
5
6
7
8
9
10
11
12
13
Lag


Probability Plot of Sensor 1 Low Temp 1000 g
Normal
Percent
99
95
90
80
70
60
50
40
30
20
10
5
1
990
995
1000
1005
1010
1015
1020
Sensor 1 Low Temp 1000 g
Mean 1005
StDev 6.143
N 50
RJ 0.986
P-Value >0.100

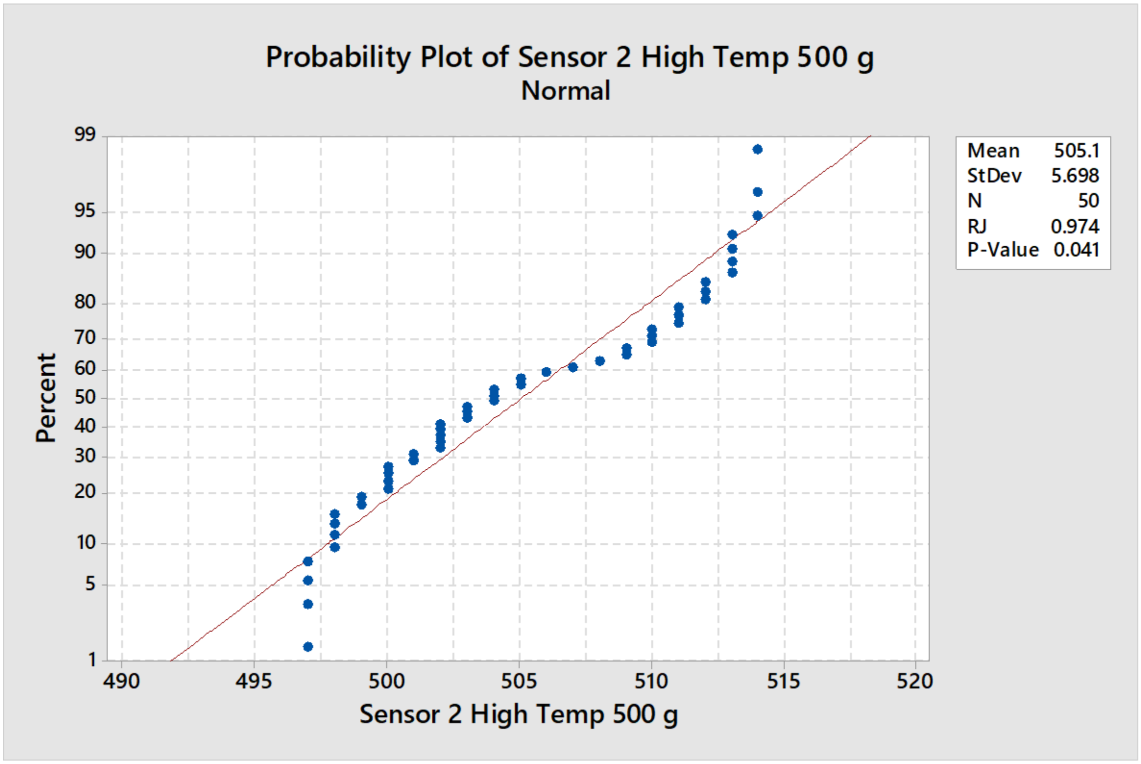

Autocorrelation Function for Sensor 1 Low Temp 1000 g
(with 5% significance limits for the autocorrelations)
Autocorrelation
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-0.8
-1.0
1
2
3
4
5
6
7
8
9
10
11
12
13
Lag

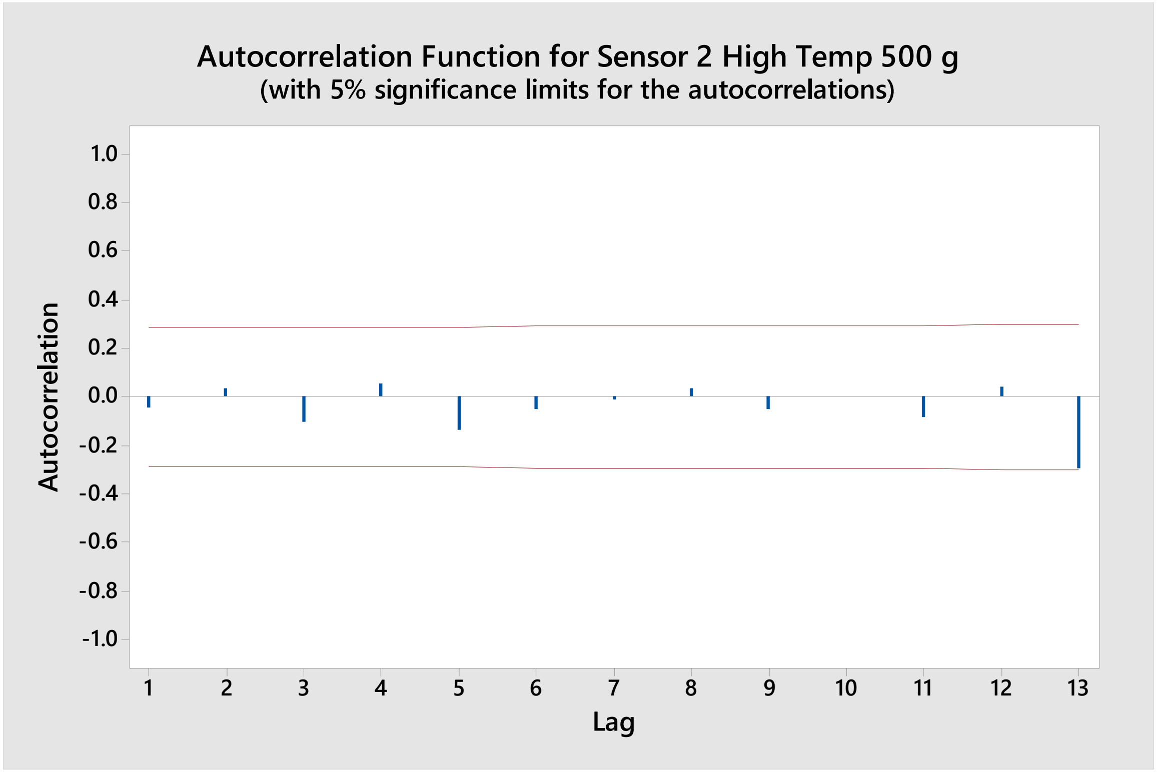

Probability Plot of Sensor 1 Low Temp 5000 g
Normal
Percent
99
95
90
80
70
60
50
40
30
20
10
5
1
4985
4990
4995
5000
5005
5010
5015
Sensor 1 Low Temp 5000 g
Mean 5000
StDev 6.282
N 50
RJ 0.985
P-Value >0.100

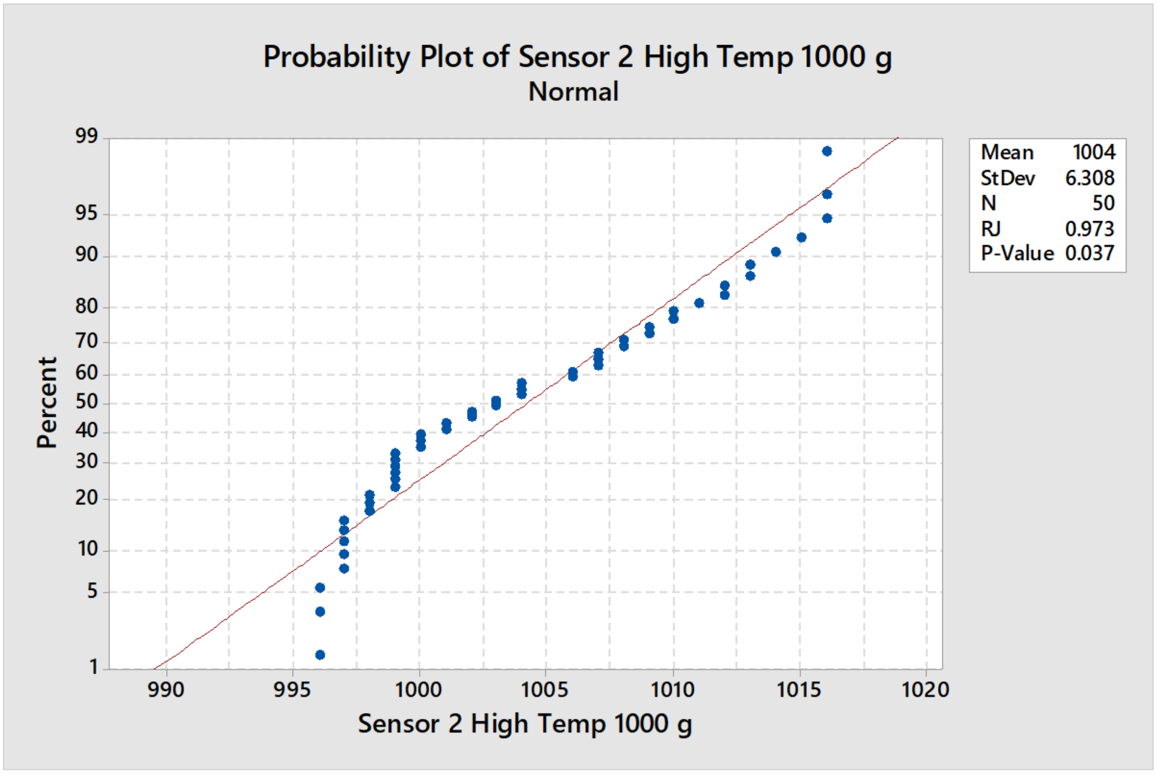

Autocorrelation Function for Sensor 1 Low Temp 5000 g
(with 5% significance limits for the autocorrelations)
Autocorrelation
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-0.8
-1.0
1
2
3
4
5
6
7
8
9
10
11
12
13
Lag

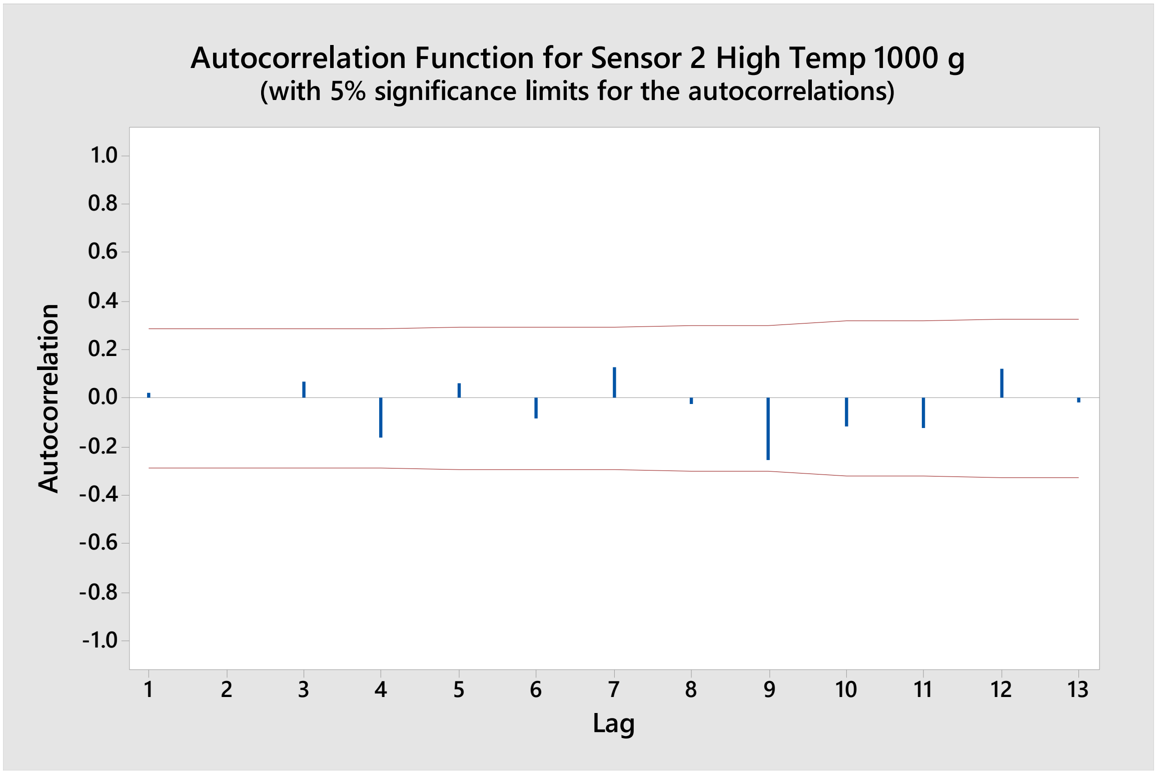
Sensor 2

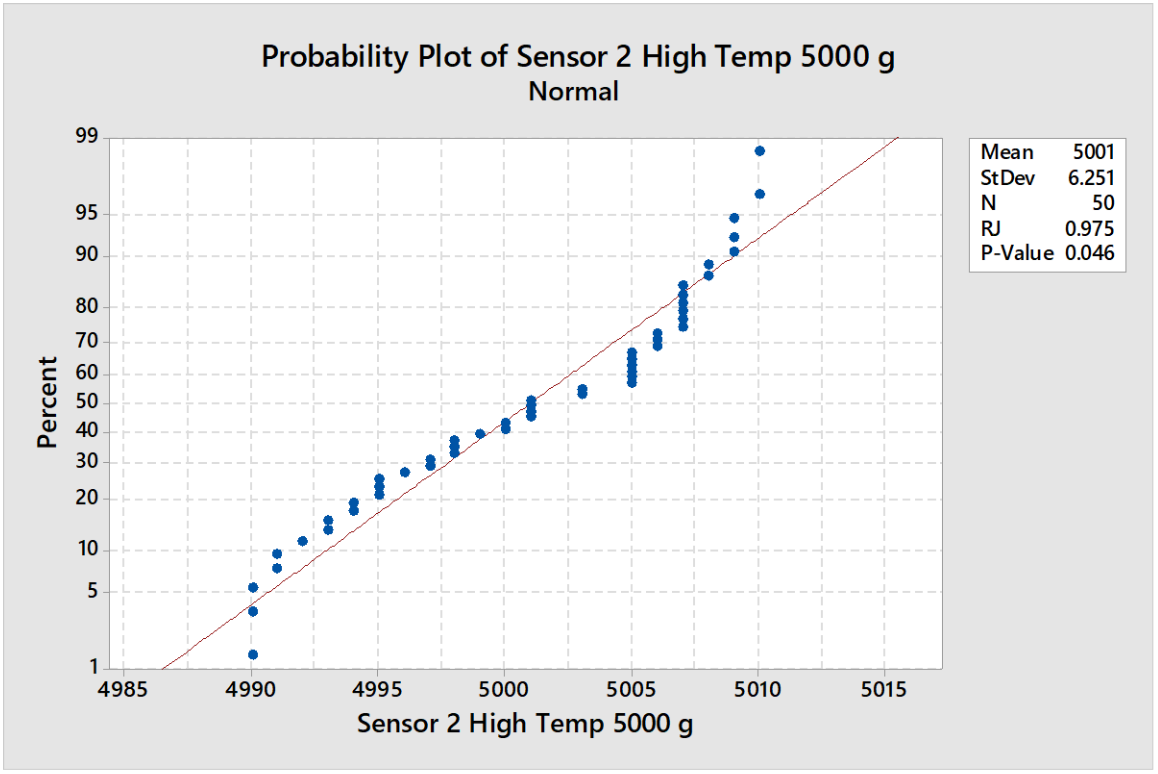
High Temperature

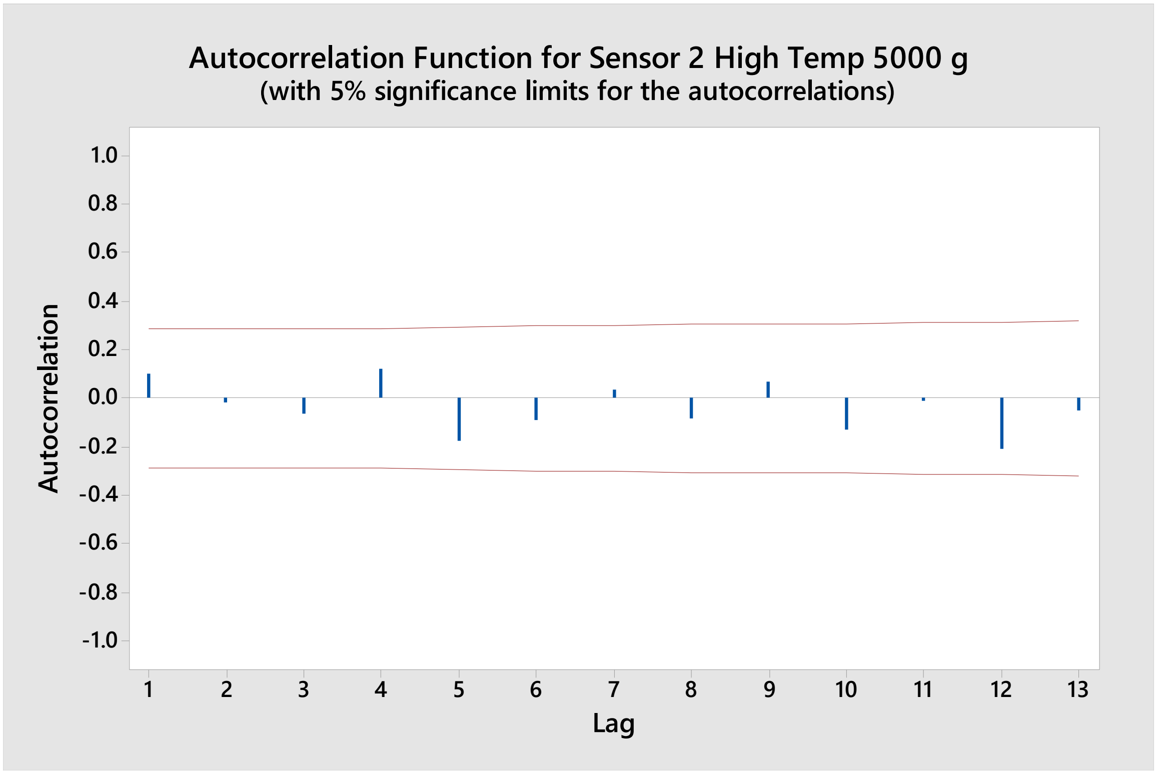




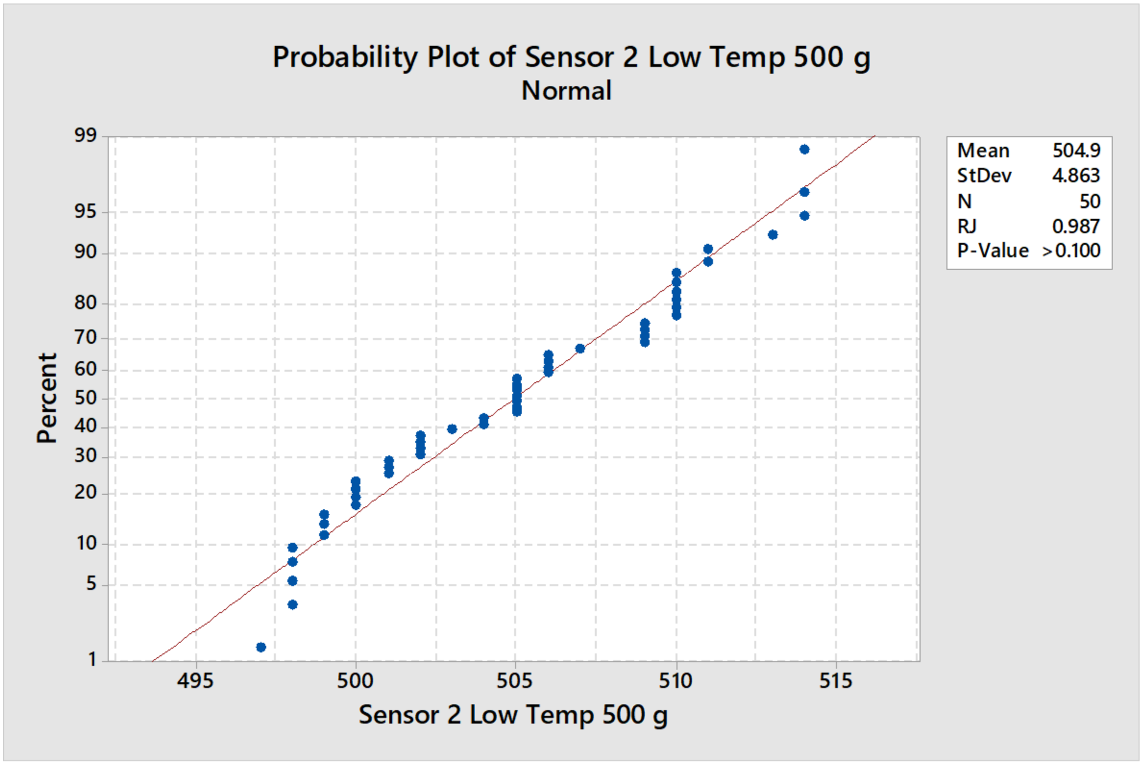

Probability Plot of Sensor 2 High Temp 1000 g
Normal
Percent
99
95
90
80
70
60
50
40
30
20
10
5
1
990
995
1000
1005
1010
1015
1020
Sensor 2 High Temp 1000 g
Mean 1004
StDev 6.308
N 50
RJ 0.973
P-Value 0.037

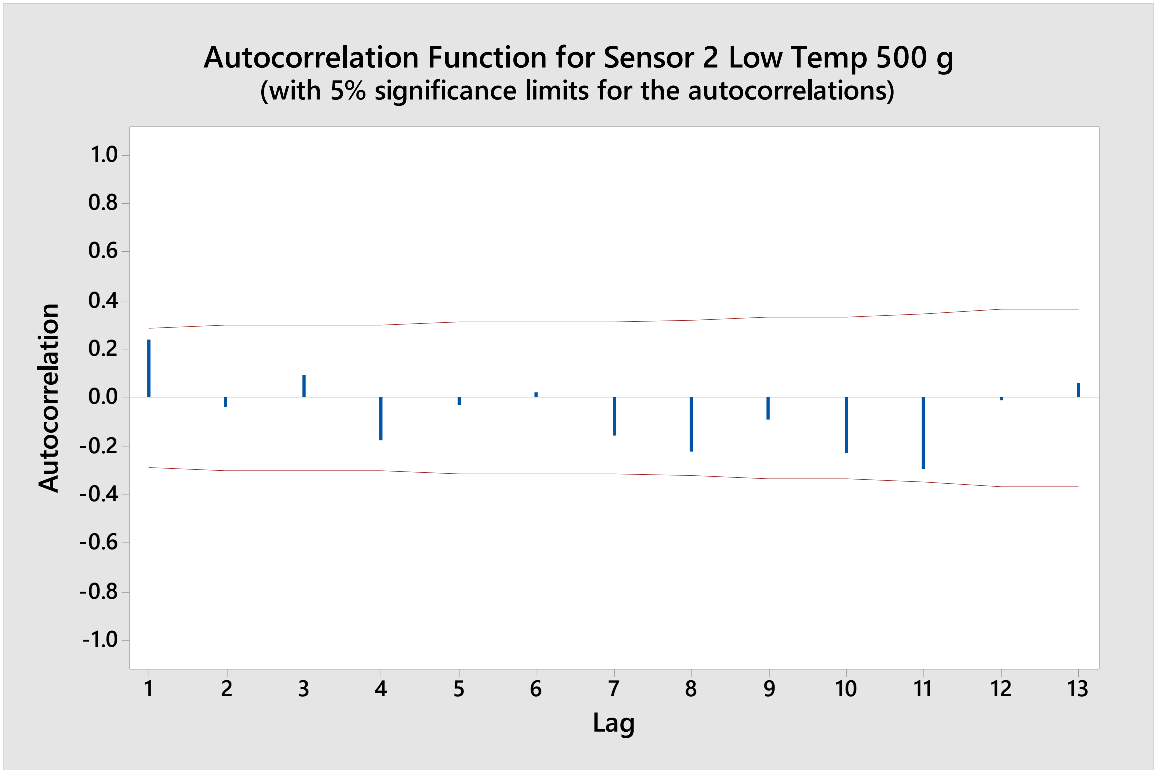

Autocorrelation Function for Sensor 2 High Temp 1000 g
(with 5% significance limits for the autocorrelations)
Autocorrelation
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-0.8
-1.0
1
2
3
4
5
6
7
8
9
10
11
12
13
Lag

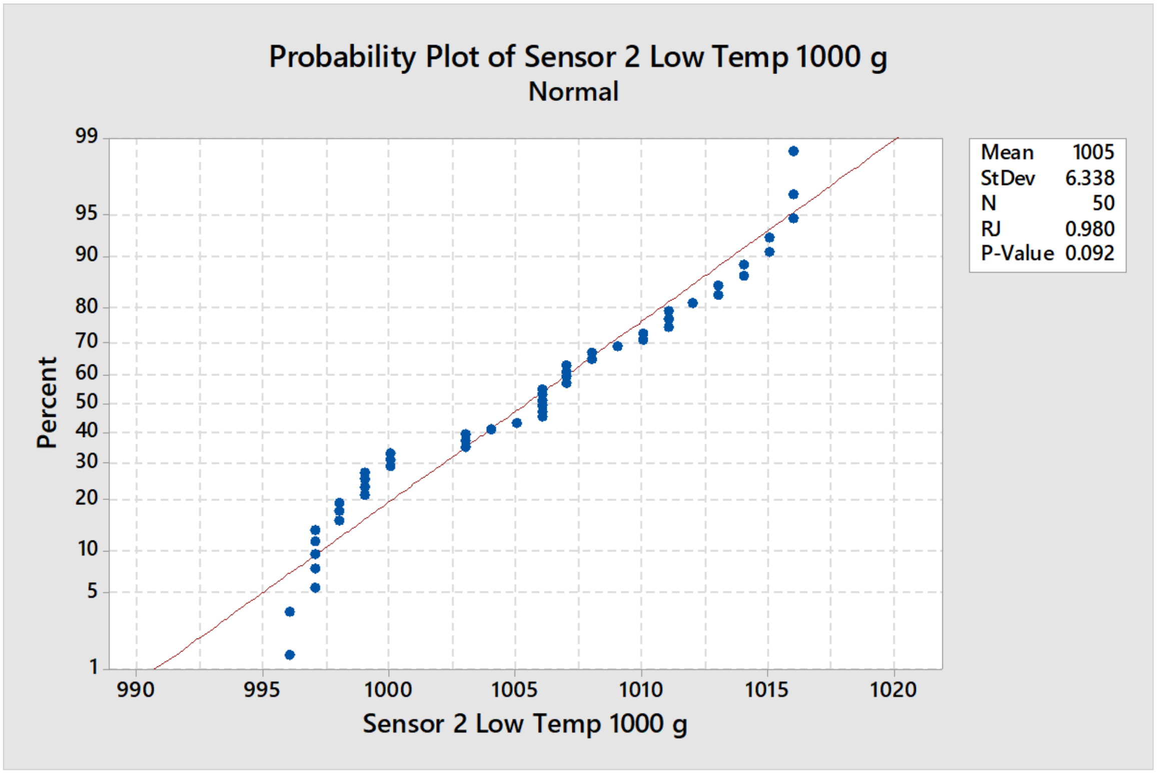

Probability Plot of Sensor 2 High Temp 5000 g
Normal
Percent
99
95
90
80
70
60
50
40
30
20
10
5
1
4985
4990
4995
5000
5005
5010
5015
Sensor 2 High Temp 5000 g
Mean 5001
StDev 6.251
N 50
RJ 0.975
P-Value 0.046


Autocorrelation Function for Sensor 2 High Temp 5000 g
(with 5% significance limits for the autocorrelations)
Autocorrelation
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-0.8
-1.0
1
2
3
4
5
6
7
8
9
10
11
12
13
Lag

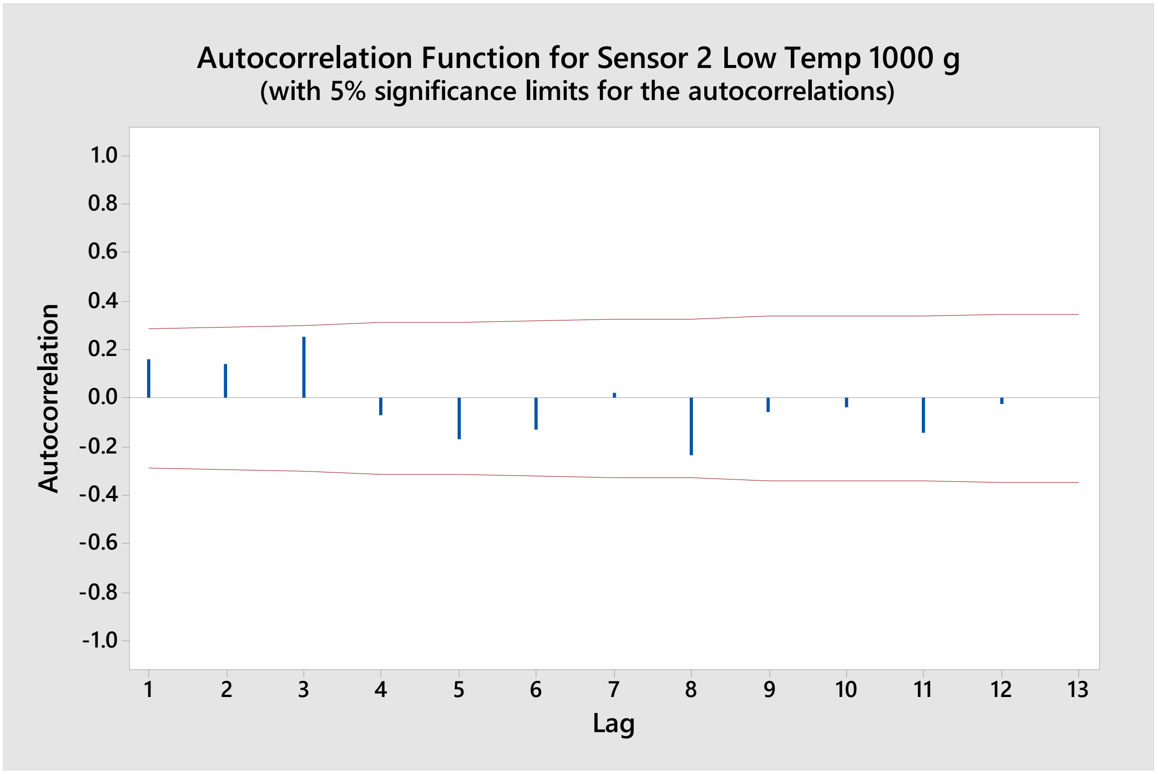
Low Temperature



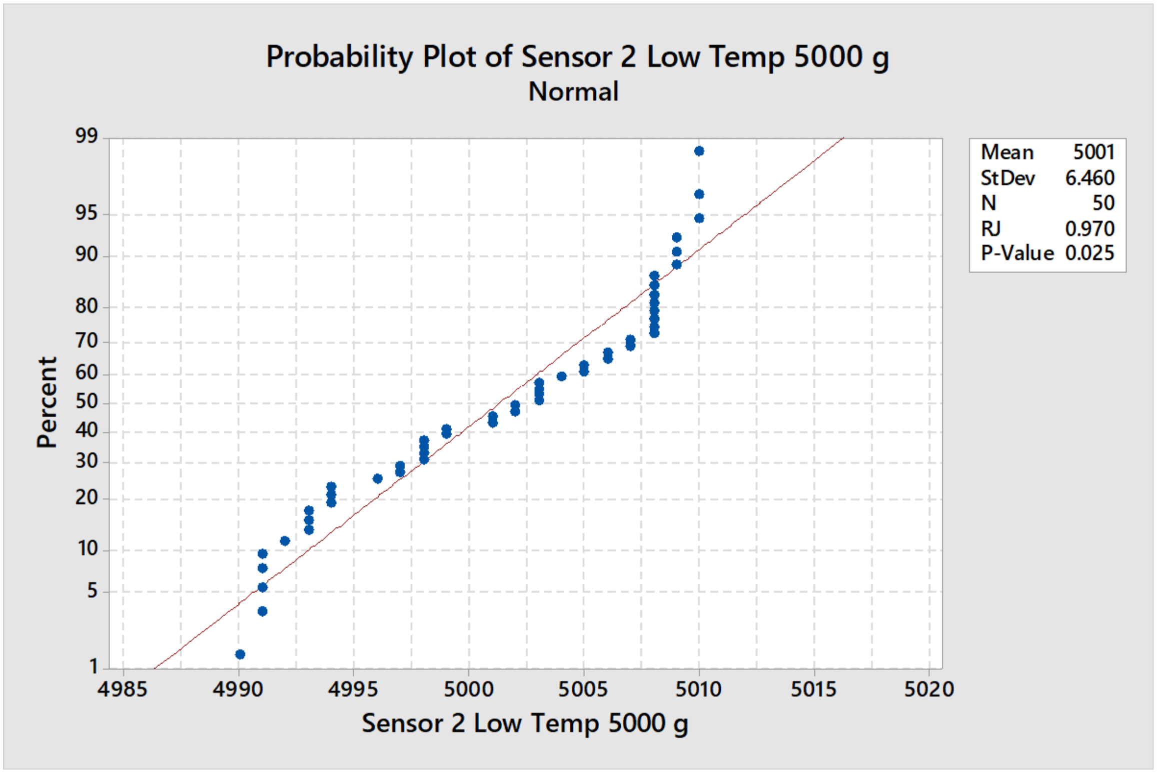


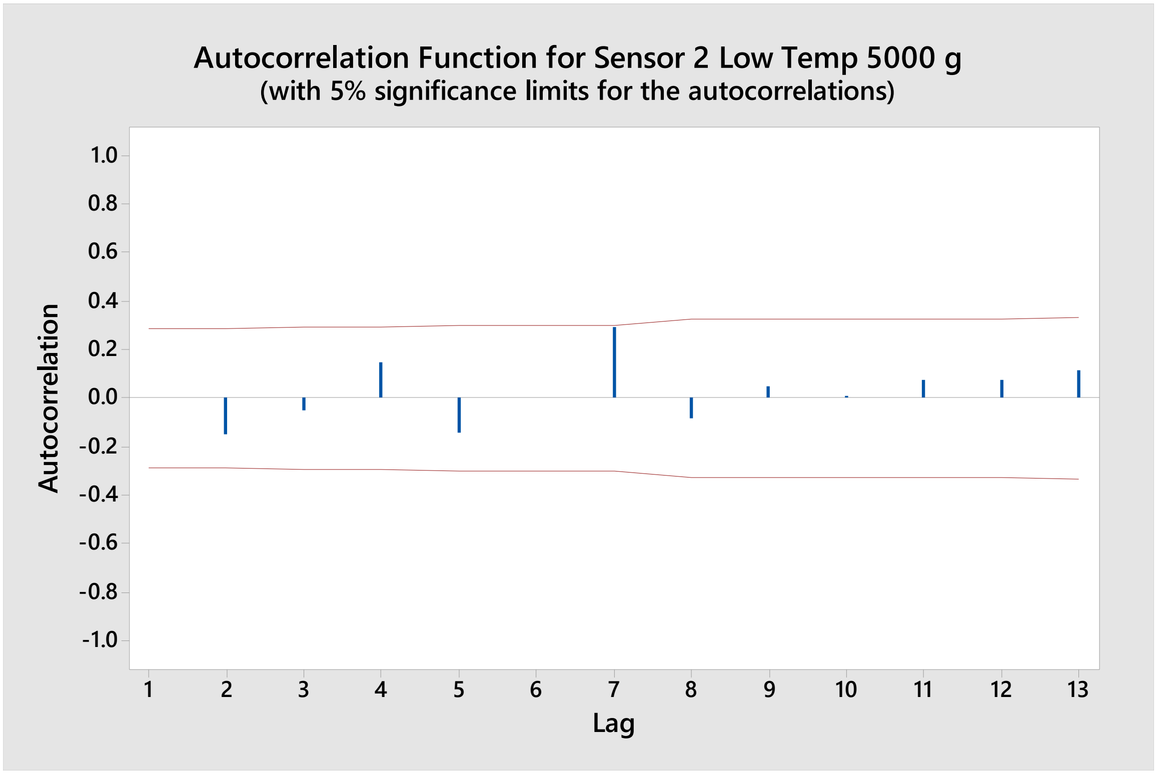


Probability Plot of Sensor 2 Low Temp 1000 g
Normal
Percent
99
95
90
80
70
60
50
40
30
20
10
5
1
990
995
1000
1005
1010
1015
1020
Sensor 2 Low Temp 1000 g
Mean 1005
StDev 6.338
N 50
RJ 0.980
P-Value 0.092



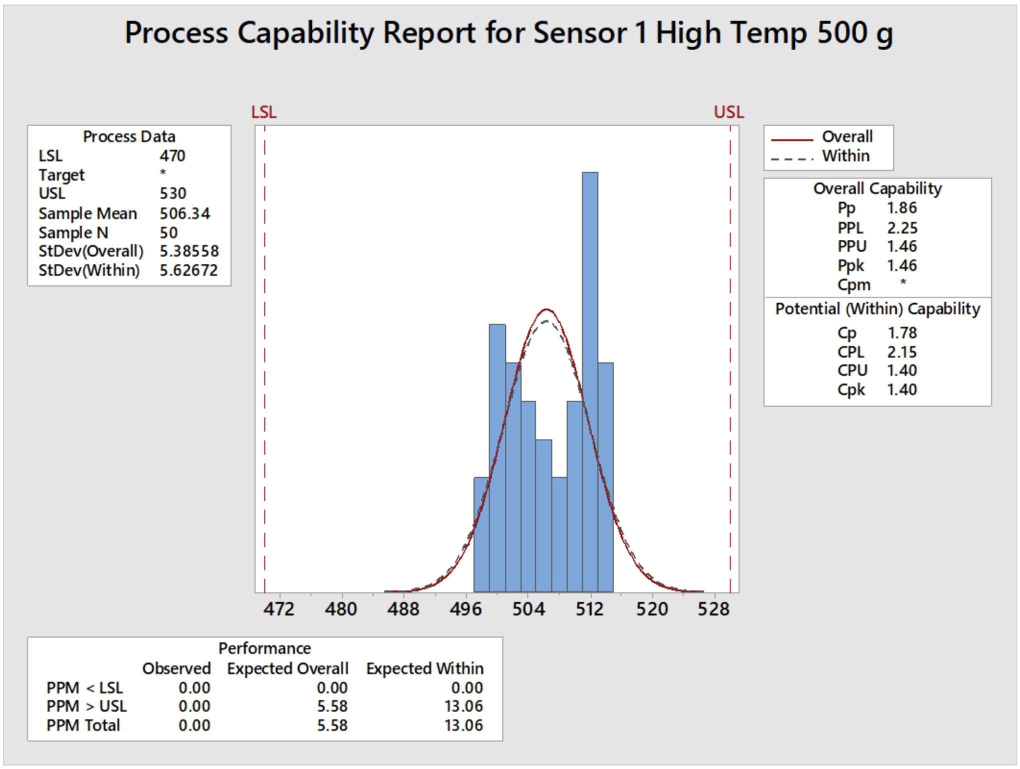
Autocorrelation Function for Sensor 2 Low Temp 1000 g
(with 5% significance limits for the autocorrelations)
Autocorrelation
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-0.8
-1.0
1
2
3
4
5
6
7
8
9
10
11
12
13
Lag


Probability Plot of Sensor 2 Low Temp 5000 g
Normal
Mean 5001
StDev 6.460
N 50
RJ 0.970
P-Value 0.025
Percent
99
95
90
80
70
60
50
40
30
20
10
5
1
4985
4990
4995
5000
5005
5010
5015
5020
Sensor 2 Low Temp 5000 g


Autocorrelation Function for Sensor 2 Low Temp 5000 g
(with 5% significance limits for the autocorrelations)
Autocorrelation
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-0.8
-1.0
1
2
3
4
5
6
7
8
9
10
11
12
13
Lag

Capability Plots

500 grams

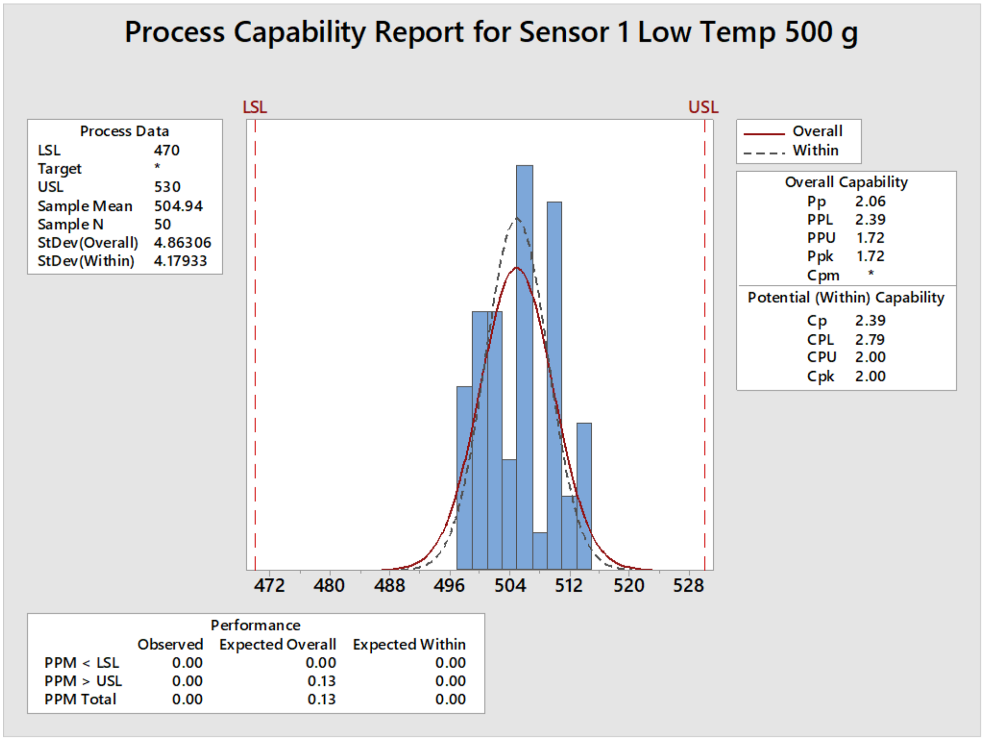


Process Data	
LSL	470
Target	*
USL	530
Sample Mean	506.34
Sample N	50
StDev(Overall)	5.38558
StDev(Within)	5.62672

Overall Capability	
Pp	1.86
PPL	2.25
PPU	1.46
Ppk	1.46
Cpm	*

Potential (Within) Capability	
Cp	1.78
CPL	2.15
CPU	1.40
Cpk	1.40

Performance	Observed	Expected Overall	Expected Within
PPM < LSL	0.00	0.00	0.00
PPM > USL	0.00	5.58	13.06
PPM Total	0.00	5.58	13.06

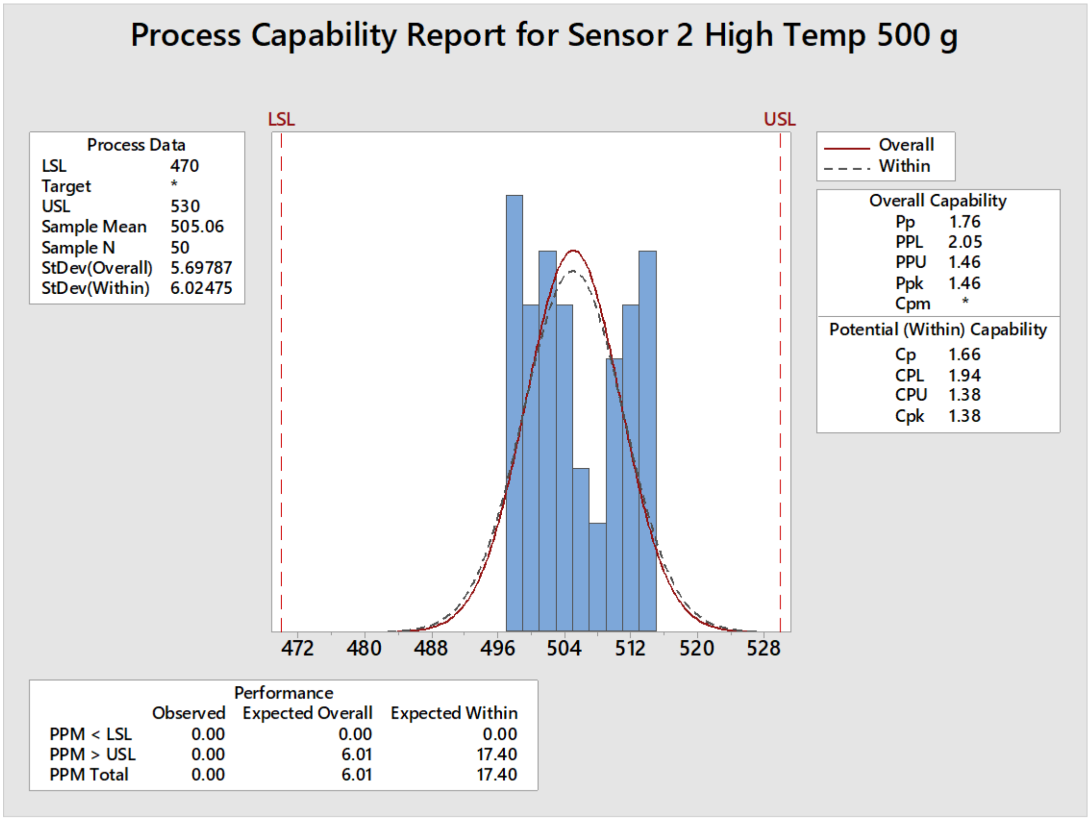


Process Data	
LSL	470
Target	*
USL	530
Sample Mean	504.94
Sample N	50
StDev(Overall)	4.86306
StDev(Within)	4.17933

Overall Capability	
Pp	2.06
PPL	2.39
PPU	1.72
Ppk	1.72
Cpm	*

Potential (Within) Capability	
Cp	2.39
CPL	2.79
CPU	2.00
Cpk	2.00

Performance	Observed	Expected Overall	Expected Within
PPM < LSL	0.00	0.00	0.00
PPM > USL	0.00	0.13	0.00
PPM Total	0.00	0.13	0.00

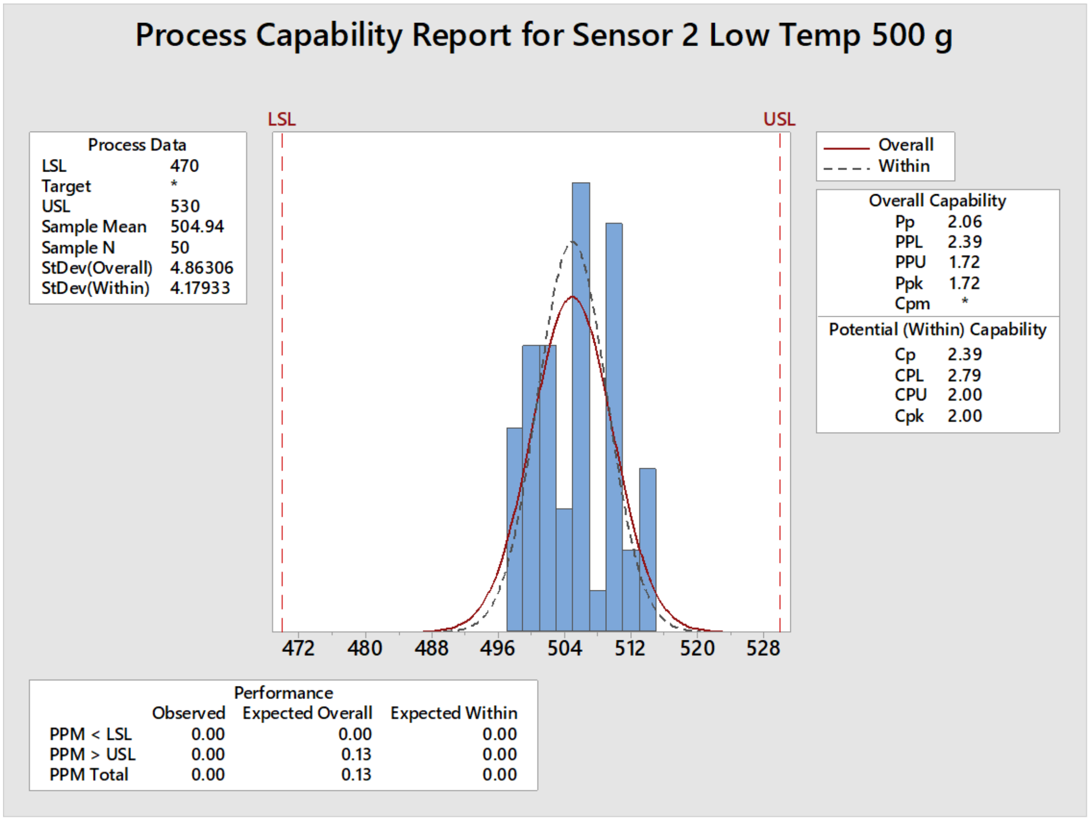


Process Capability Report for Sensor 2 High Temp 500 g

Process Data	
LSL	470
Target	*
USL	530
Sample Mean	505.06
Sample N	50
StDev(Overall)	5.69787
StDev(Within)	6.02475

Overall Capability	
Pp	1.76
PPL	2.05
PPU	1.46
Ppk	1.46
Cpm	*
Potential (Within) Capability	
Cp	1.66
CPL	1.94
CPU	1.38
Cpk	1.38

Performance	Observed	Expected Overall	Expected Within
PPM < LSL	0.00	0.00	0.00
PPM > USL	0.00	6.01	17.40
PPM Total	0.00	6.01	17.40



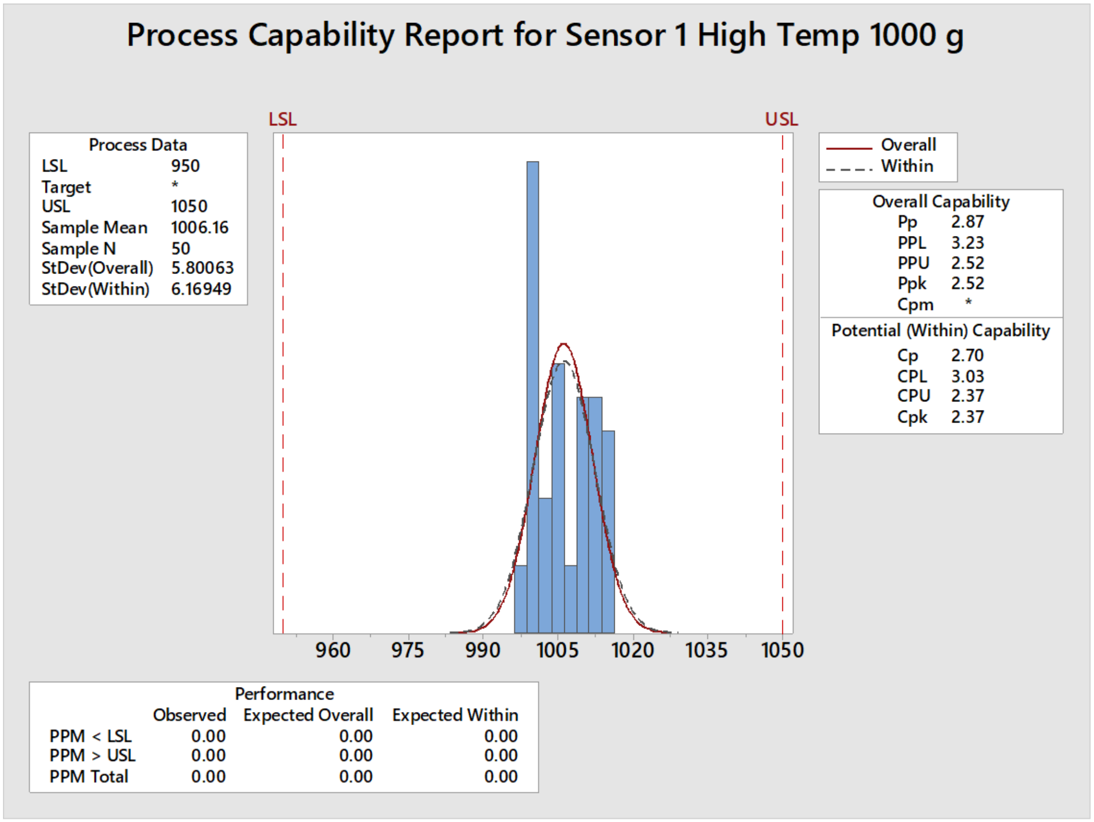
Process Capability Report for Sensor 2 Low Temp 500 g

Process Data	
LSL	470
Target	*
USL	530
Sample Mean	504.94
Sample N	50
StDev(Overall)	4.86306
StDev(Within)	4.17933

Overall Capability	
Pp	2.06
PPL	2.39
PPU	1.72
Ppk	1.72
Cpm	*
Potential (Within) Capability	
Cp	2.39
CPL	2.79
CPU	2.00
Cpk	2.00

Performance	Observed	Expected Overall	Expected Within
PPM < LSL	0.00	0.00	0.00
PPM > USL	0.00	0.13	0.00
PPM Total	0.00	0.13	0.00

1000 grams:

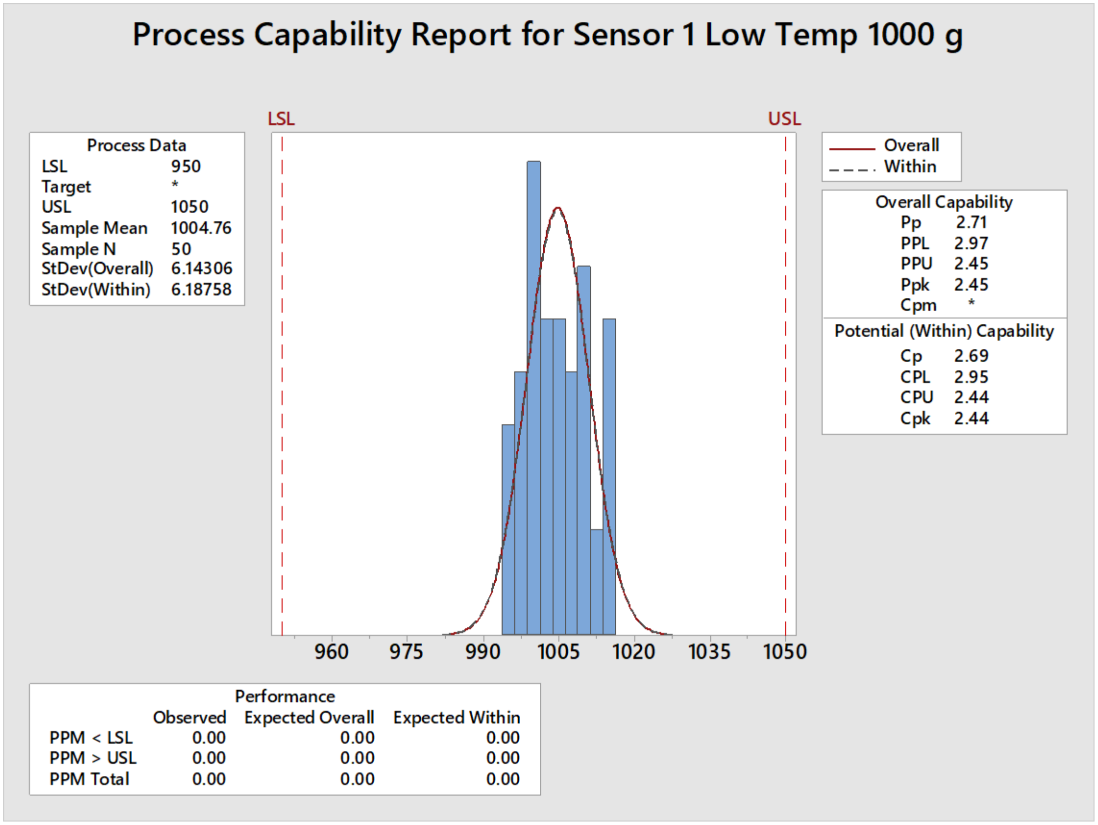


Process Capability Report for Sensor 1 High Temp 1000 g

Process Data	
LSL	950
Target	*
USL	1050
Sample Mean	1006.16
Sample N	50
StDev(Overall)	5.80063
StDev(Within)	6.16949

Overall Capability	
Pp	2.87
PPL	3.23
PPU	2.52
Ppk	2.52
Cpm	*
Potential (Within) Capability	
Cp	2.70
CPL	3.03
CPU	2.37
Cpk	2.37

Performance	Observed	Expected Overall	Expected Within
PPM < LSL	0.00	0.00	0.00
PPM > USL	0.00	0.00	0.00
PPM Total	0.00	0.00	0.00



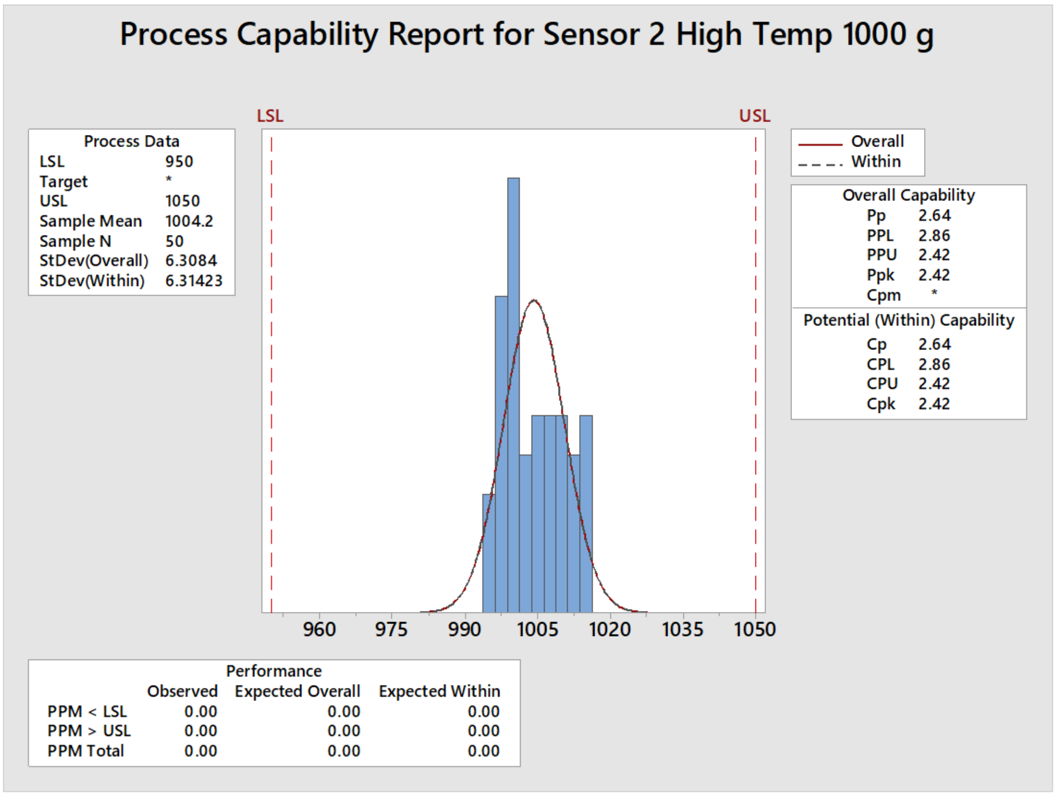
Process Capability Report for Sensor 1 Low Temp 1000 g

Process Data	
LSL	950
Target	*
USL	1050
Sample Mean	1004.76
Sample N	50
StDev(Overall)	6.14306
StDev(Within)	6.18758

Overall Capability	
Pp	2.71
PPL	2.97
PPU	2.45
Ppk	2.45
Cpm	*
Potential (Within) Capability	
Cp	2.69
CPL	2.95
CPU	2.44
Cpk	2.44

Performance	Observed	Expected Overall	Expected Within
PPM < LSL	0.00	0.00	0.00
PPM > USL	0.00	0.00	0.00
PPM Total	0.00	0.00	0.00

Process Capability Report for Sensor 2 High Temp 1000 g



Process Data

LSL	950
Target	*
USL	1050
Sample Mean	1004.2
Sample N	50
StDev(Overall)	6.3084
StDev(Within)	6.31423

—	Overall
- - -	Within

Overall Capability

Pp	2.64
PPL	2.86
PPU	2.42
Ppk	2.42
Cpm	*

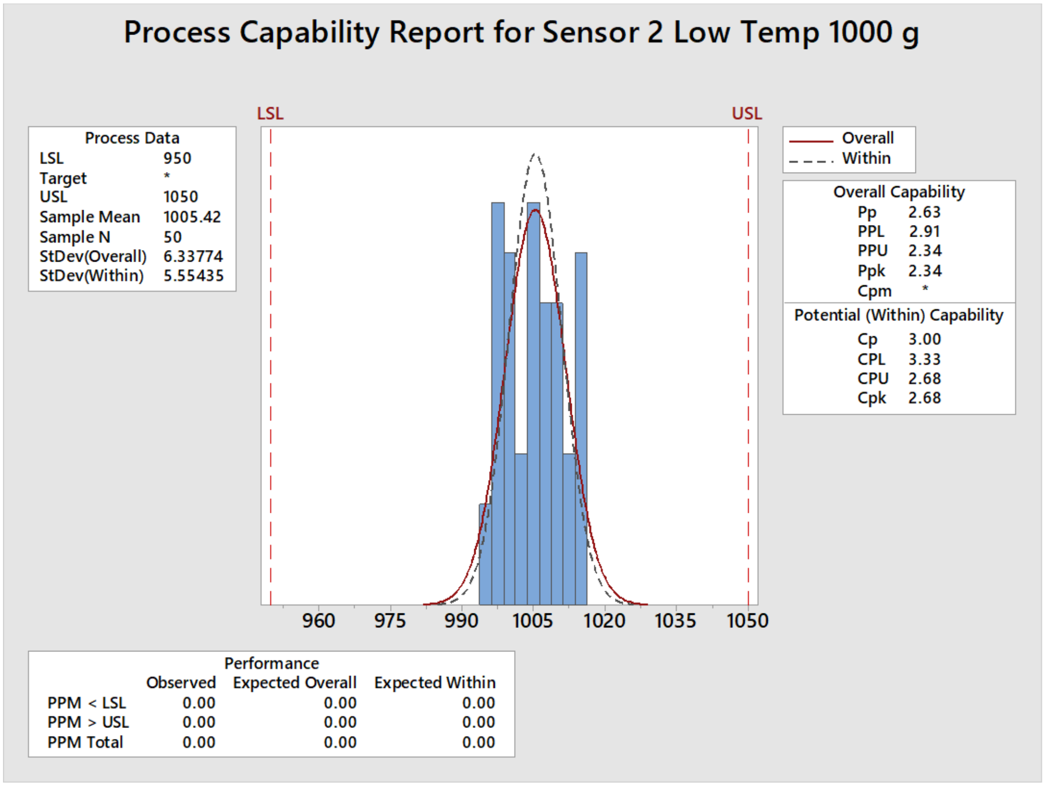
Potential (Within) Capability

Cp	2.64
CPL	2.86
CPU	2.42
Cpk	2.42

Performance

	Observed	Expected Overall	Expected Within
PPM < LSL	0.00	0.00	0.00
PPM > USL	0.00	0.00	0.00
PPM Total	0.00	0.00	0.00

Process Capability Report for Sensor 2 Low Temp 1000 g



Process Data

LSL	950
Target	*
USL	1050
Sample Mean	1005.42
Sample N	50
StDev(Overall)	6.33774
StDev(Within)	5.55435

—	Overall
- - -	Within

Overall Capability

Pp	2.63
PPL	2.91
PPU	2.34
Ppk	2.34
Cpm	*

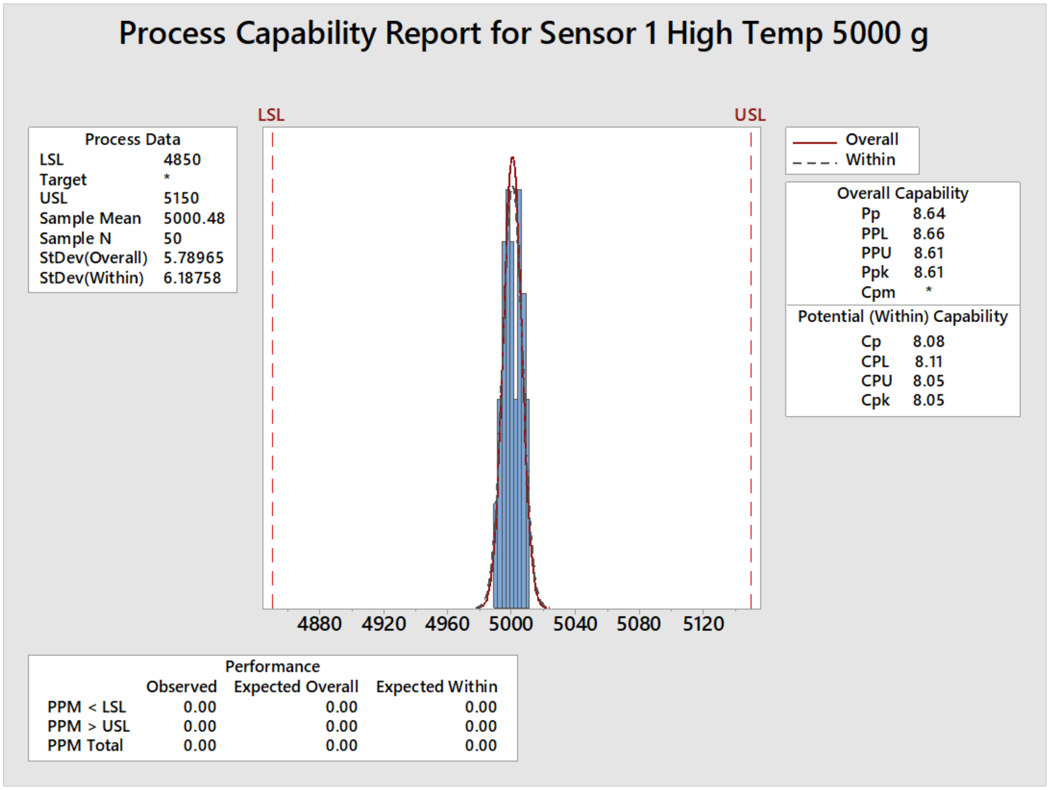
Potential (Within) Capability

Cp	3.00
CPL	3.33
CPU	2.68
Cpk	2.68

Performance

	Observed	Expected Overall	Expected Within
PPM < LSL	0.00	0.00	0.00
PPM > USL	0.00	0.00	0.00
PPM Total	0.00	0.00	0.00

5000 grams



Process Capability Report for Sensor 1 High Temp 5000 g

Process Data	
LSL	4850
Target	*
USL	5150
Sample Mean	5000.48
Sample N	50
StDev(Overall)	5.78965
StDev(Within)	6.18758

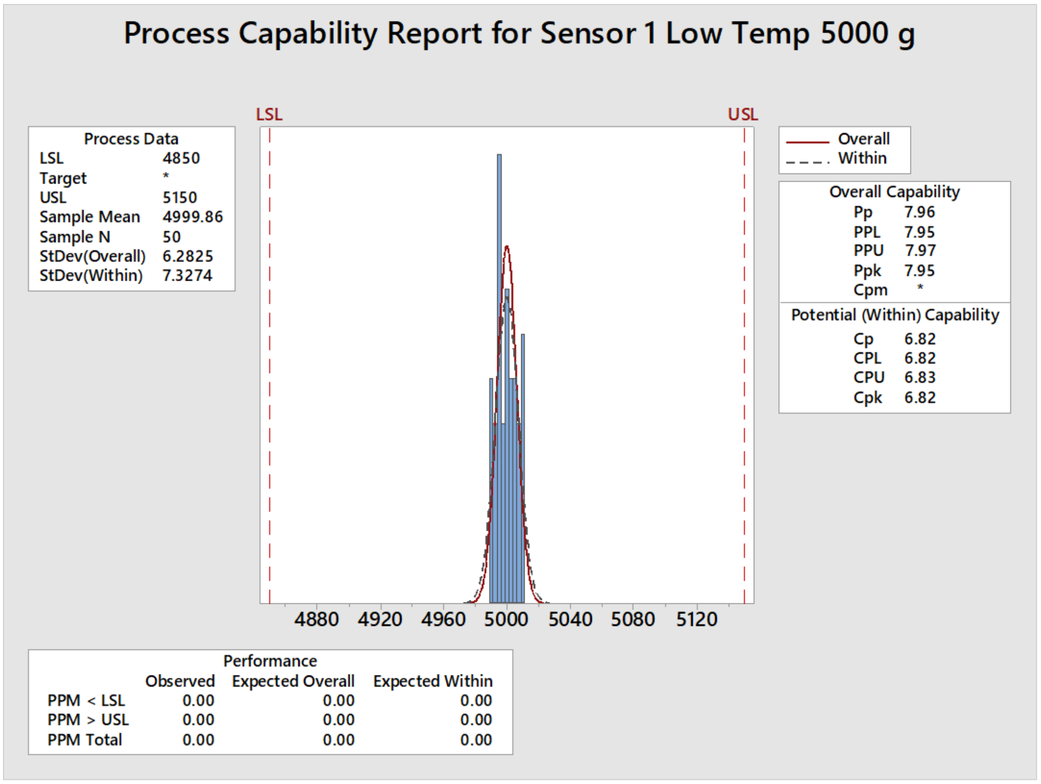
—	Overall
- - -	Within

Overall Capability	
Pp	8.64
PPL	8.66
PPU	8.61
Ppk	8.61
Cpm	*

Potential (Within) Capability	
Cp	8.08
CPL	8.11
CPU	8.05
Cpk	8.05

Performance

	Observed	Expected Overall	Expected Within
PPM < LSL	0.00	0.00	0.00
PPM > USL	0.00	0.00	0.00
PPM Total	0.00	0.00	0.00



Process Capability Report for Sensor 1 Low Temp 5000 g

Process Data	
LSL	4850
Target	*
USL	5150
Sample Mean	4999.86
Sample N	50
StDev(Overall)	6.2825
StDev(Within)	7.3274

—	Overall
- - -	Within

Overall Capability	
Pp	7.96
PPL	7.95
PPU	7.97
Ppk	7.95
Cpm	*

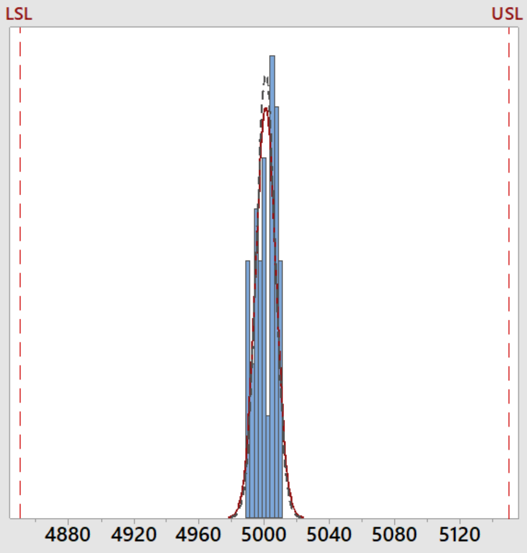
Potential (Within) Capability	
Cp	6.82
CPL	6.82
CPU	6.83
Cpk	6.82

Performance

	Observed	Expected Overall	Expected Within
PPM < LSL	0.00	0.00	0.00
PPM > USL	0.00	0.00	0.00
PPM Total	0.00	0.00	0.00

Process Capability Report for Sensor 2 High Temp 5000 g

Process Data	
LSL	4850
Target	*
USL	5150
Sample Mean	5000.98
Sample N	50
StDev(Overall)	6.2515
StDev(Within)	5.78955



───	Overall
- - - -	Within

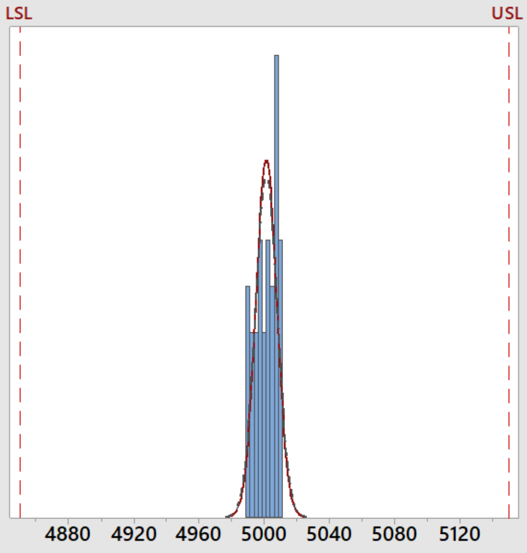
Overall Capability	
Pp	8.00
PPL	8.05
PPU	7.95
Ppk	7.95
Cpm	*
Potential (Within) Capability	
Cp	8.64
CPL	8.69
CPU	8.58
Cpk	8.58

Performance

	Observed	Expected Overall	Expected Within
PPM < LSL	0.00	0.00	0.00
PPM > USL	0.00	0.00	0.00
PPM Total	0.00	0.00	0.00

Process Capability Report for Sensor 2 Low Temp 5000 g

Process Data	
LSL	4850
Target	*
USL	5150
Sample Mean	5001.32
Sample N	50
StDev(Overall)	6.46005
StDev(Within)	6.78463



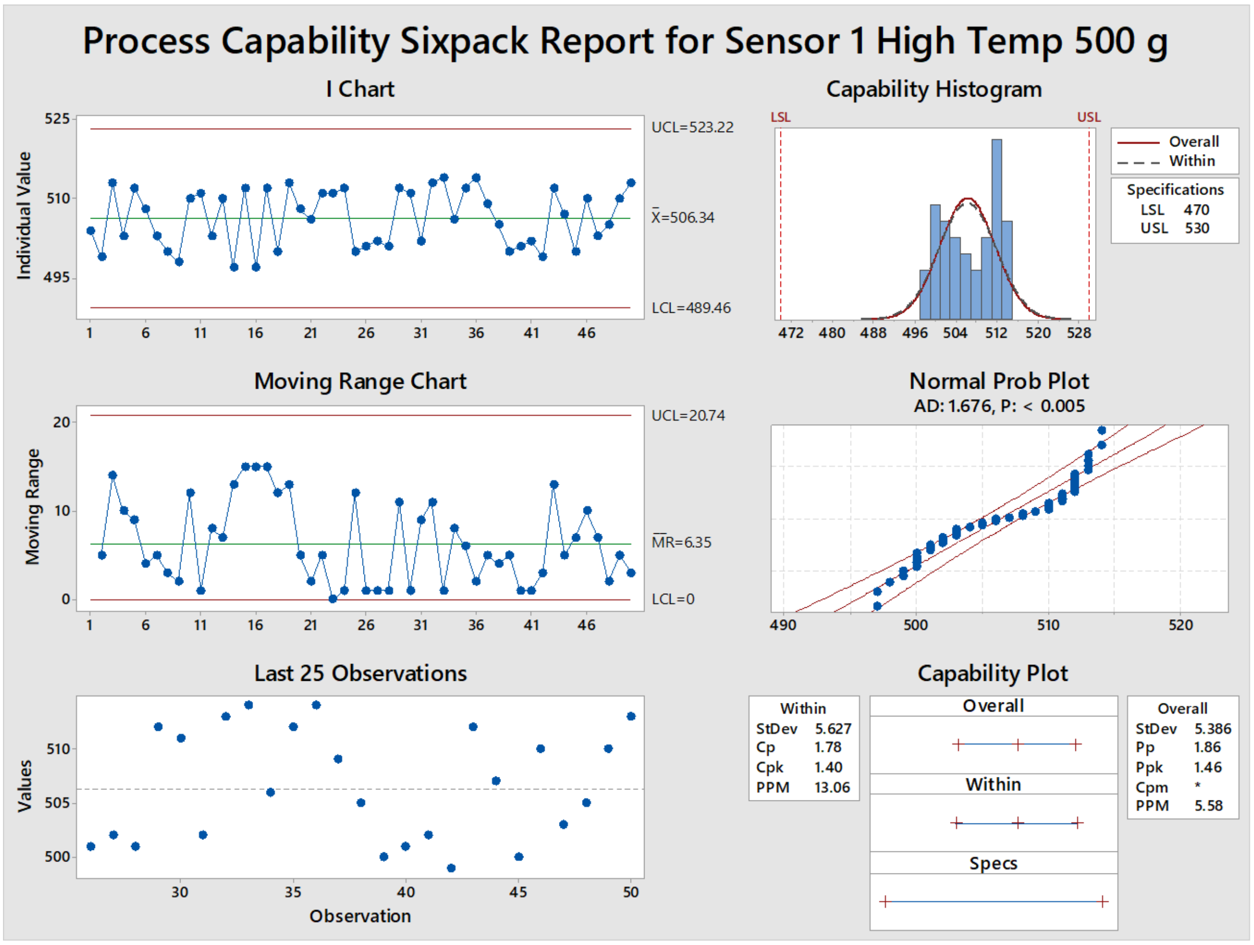
───	Overall
- - - -	Within

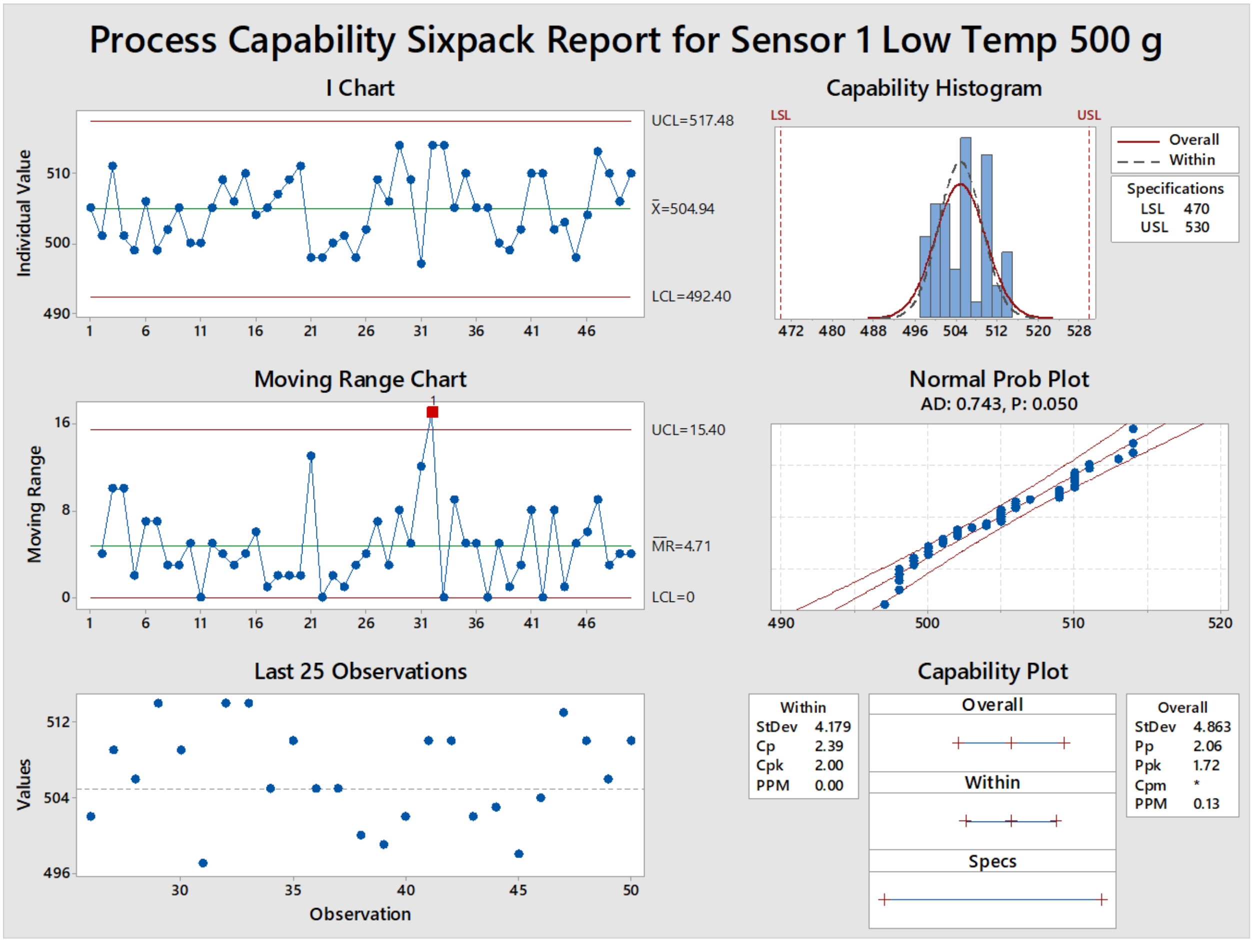
Overall Capability	
Pp	7.74
PPL	7.81
PPU	7.67
Ppk	7.67
Cpm	*
Potential (Within) Capability	
Cp	7.37
CPL	7.43
CPU	7.30
Cpk	7.30

Performance

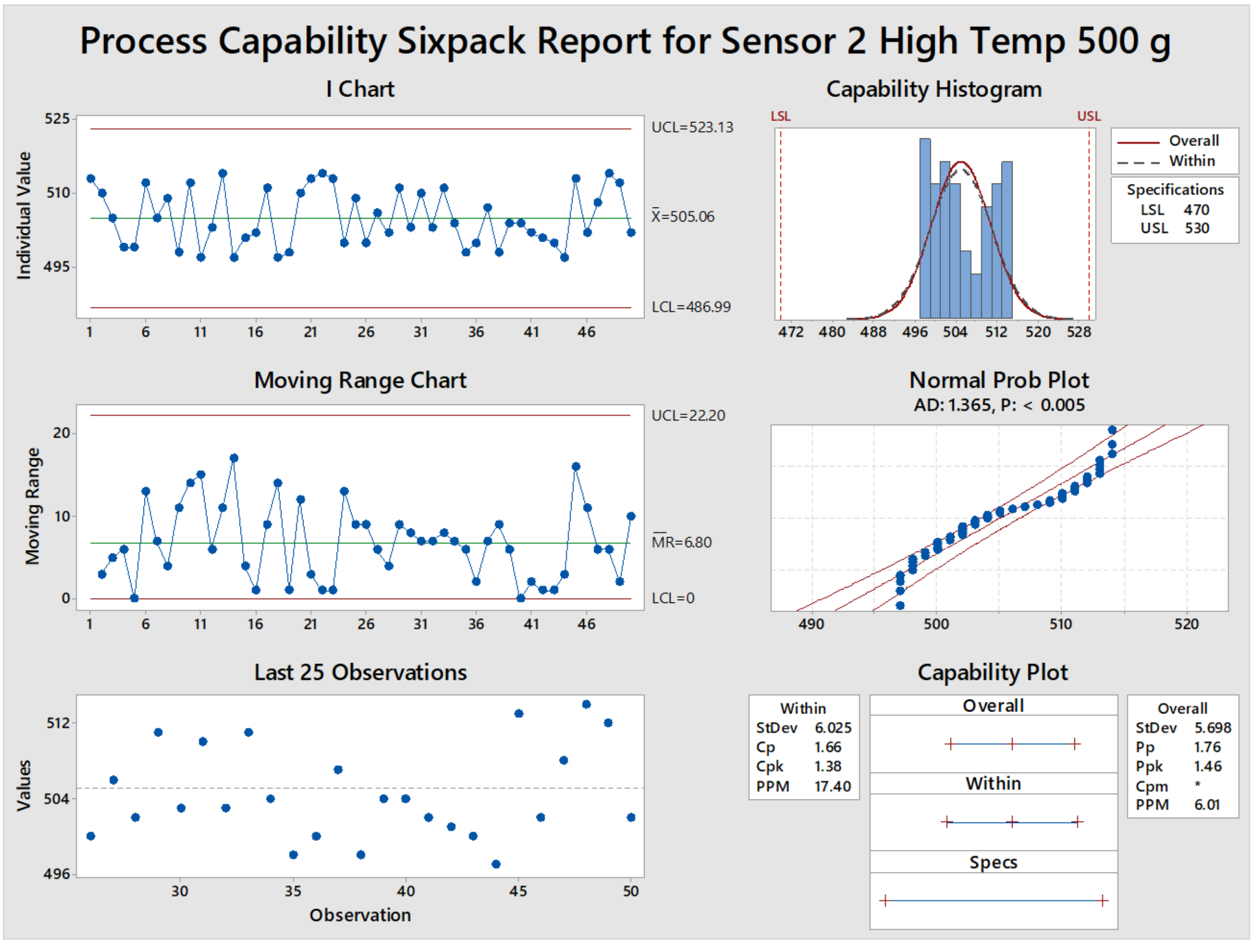
	Observed	Expected Overall	Expected Within
PPM < LSL	0.00	0.00	0.00
PPM > USL	0.00	0.00	0.00
PPM Total	0.00	0.00	0.00

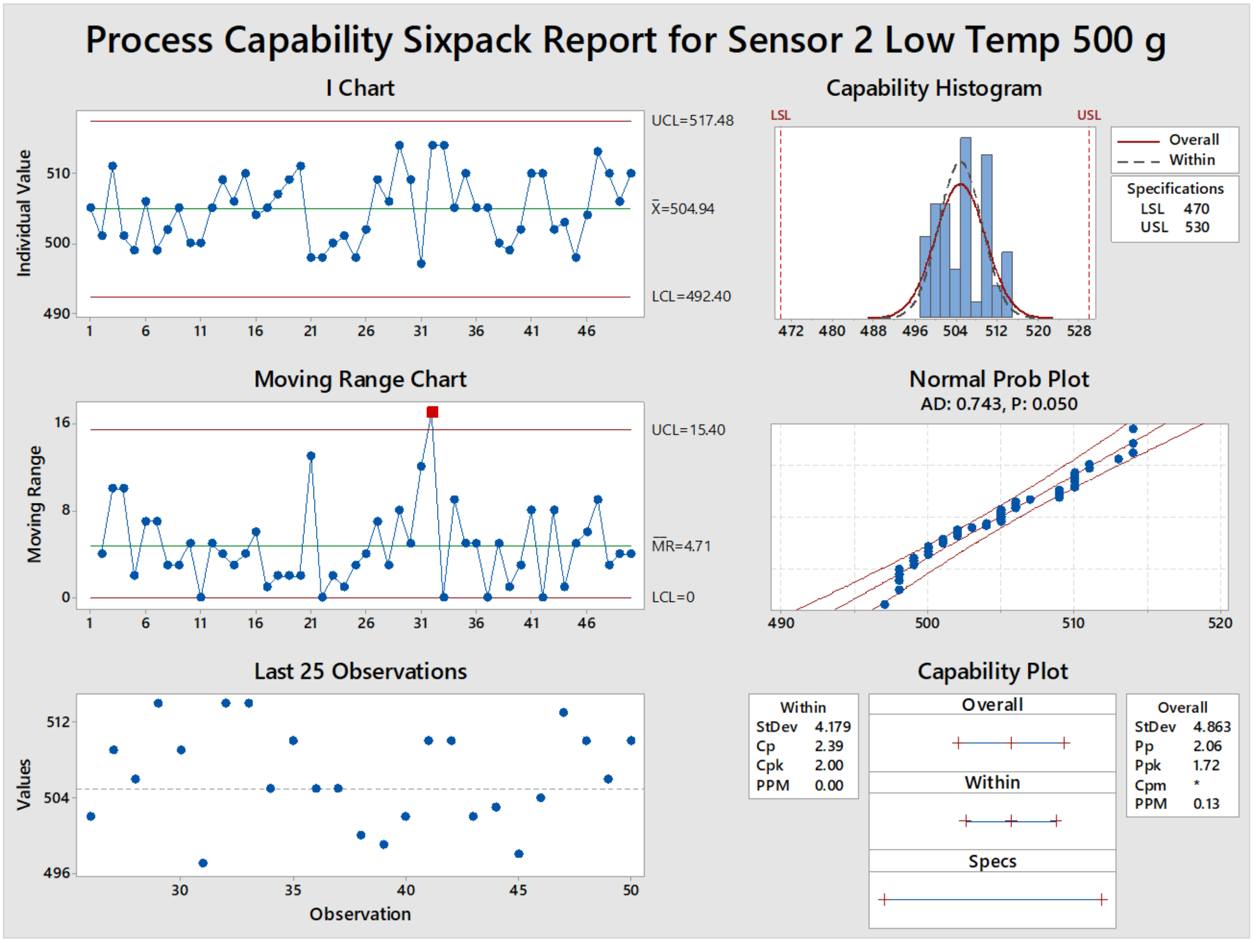
Six Pack Result

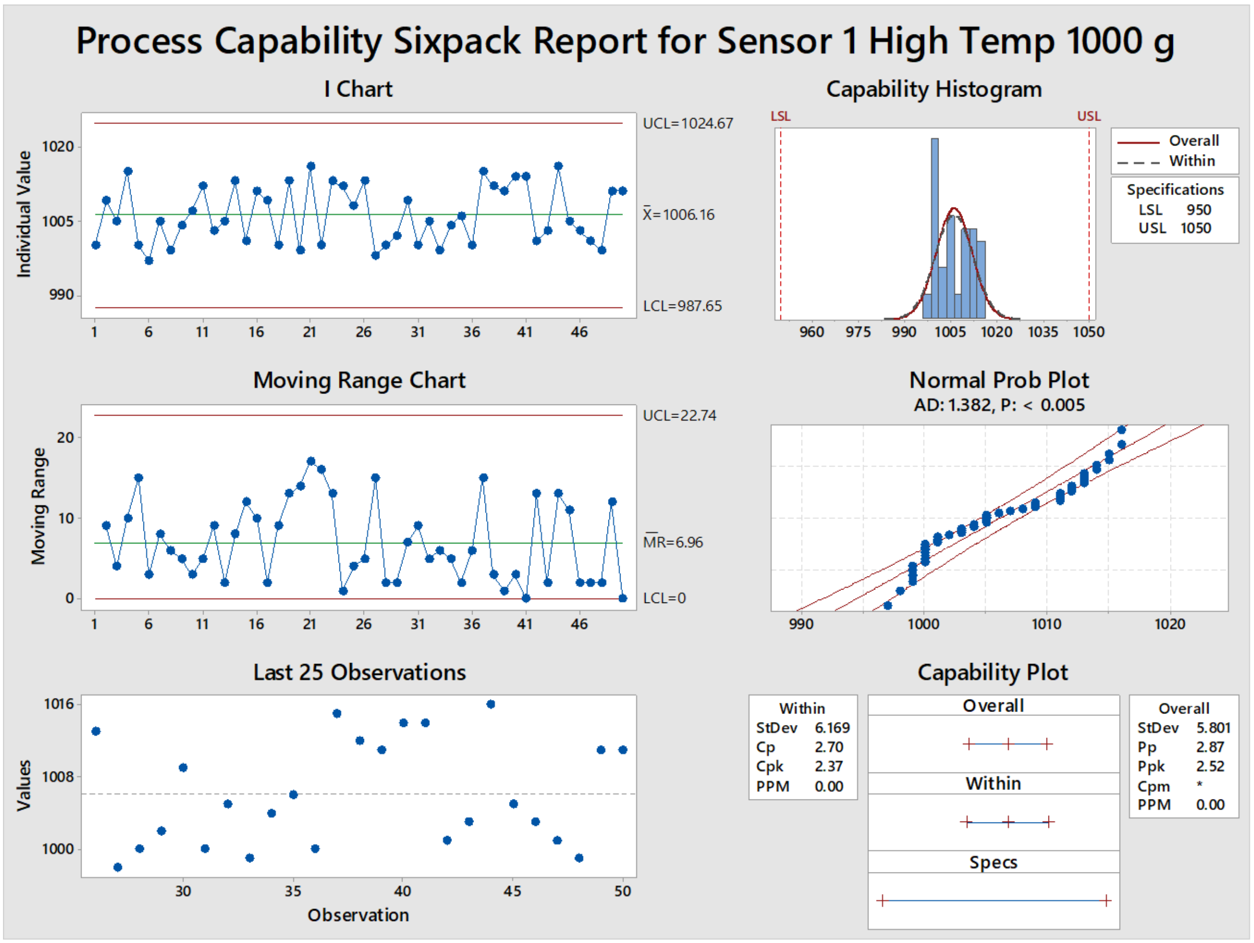


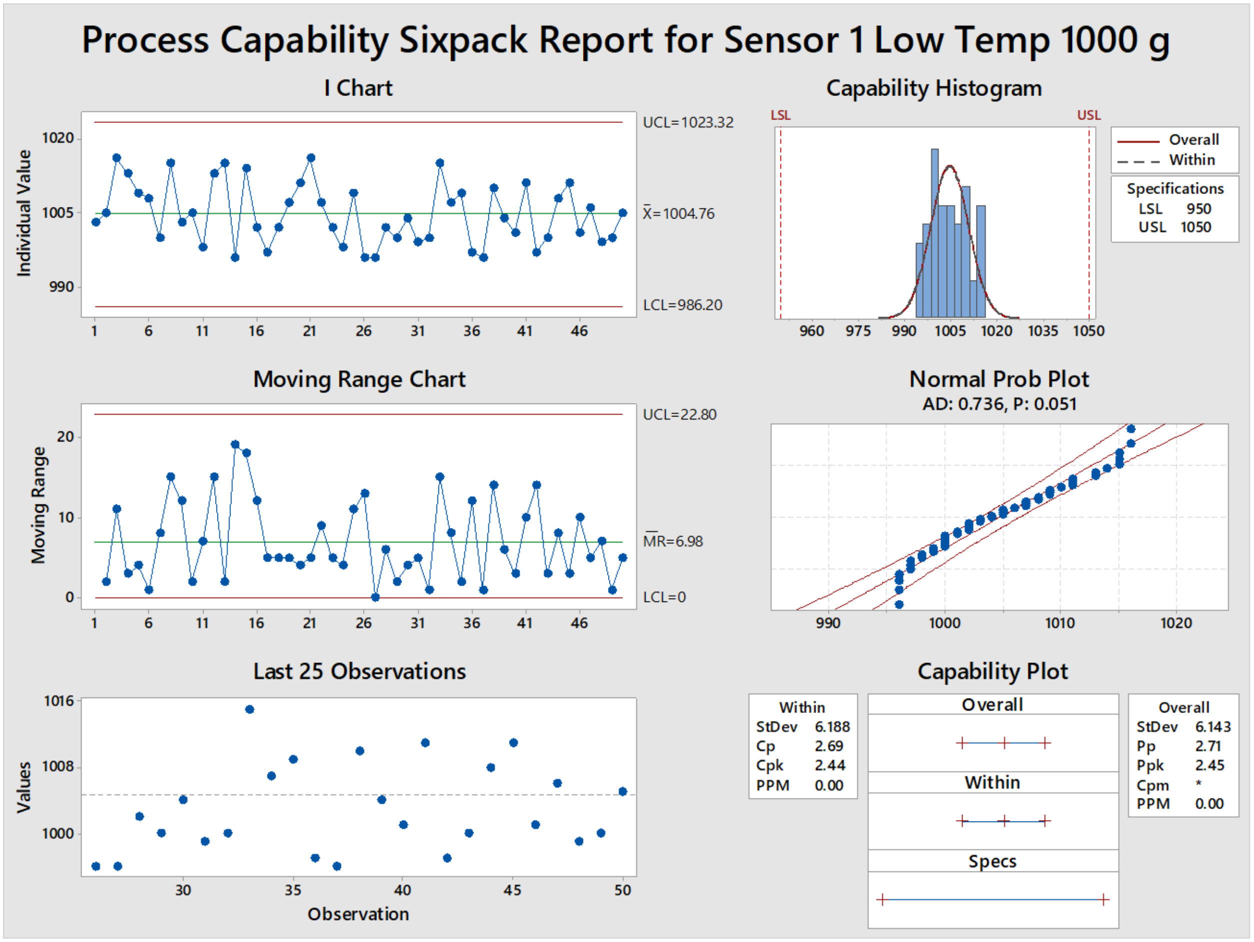


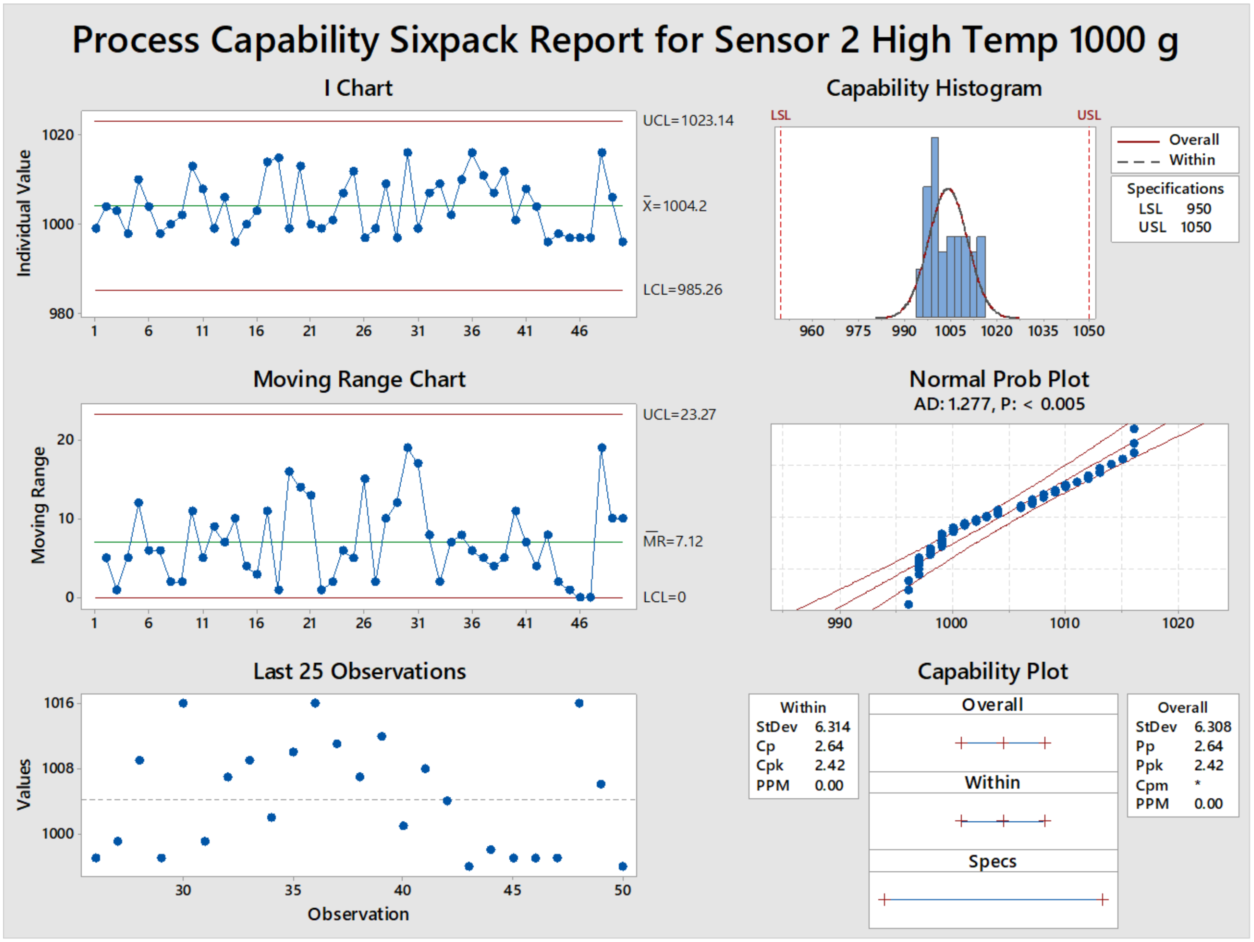
Process Capability Sixpack Report for Sensor 1 Low Temp 500 g
I Chart
Individual Value
UCL=517.48
X̄=504.94
LCL=492.40
Capability Histogram
LSL
USL
Overall
Within
Specifications
LSL 470
USL 530
Moving Range Chart
Moving Range
UCL=15.40
M̄R=4.71
LCL=0
Normal Prob Plot
AD: 0.743, P: 0.050
Last 25 Observations
Values
Observation
Capability Plot
Within
StDev 4.179
Cp 2.39
Cpk 2.00
PPM 0.00
Overall
Within
Specs
Overall
StDev 4.863
Pp 2.06
Ppk 1.72
Cpm *
PPM 0.13

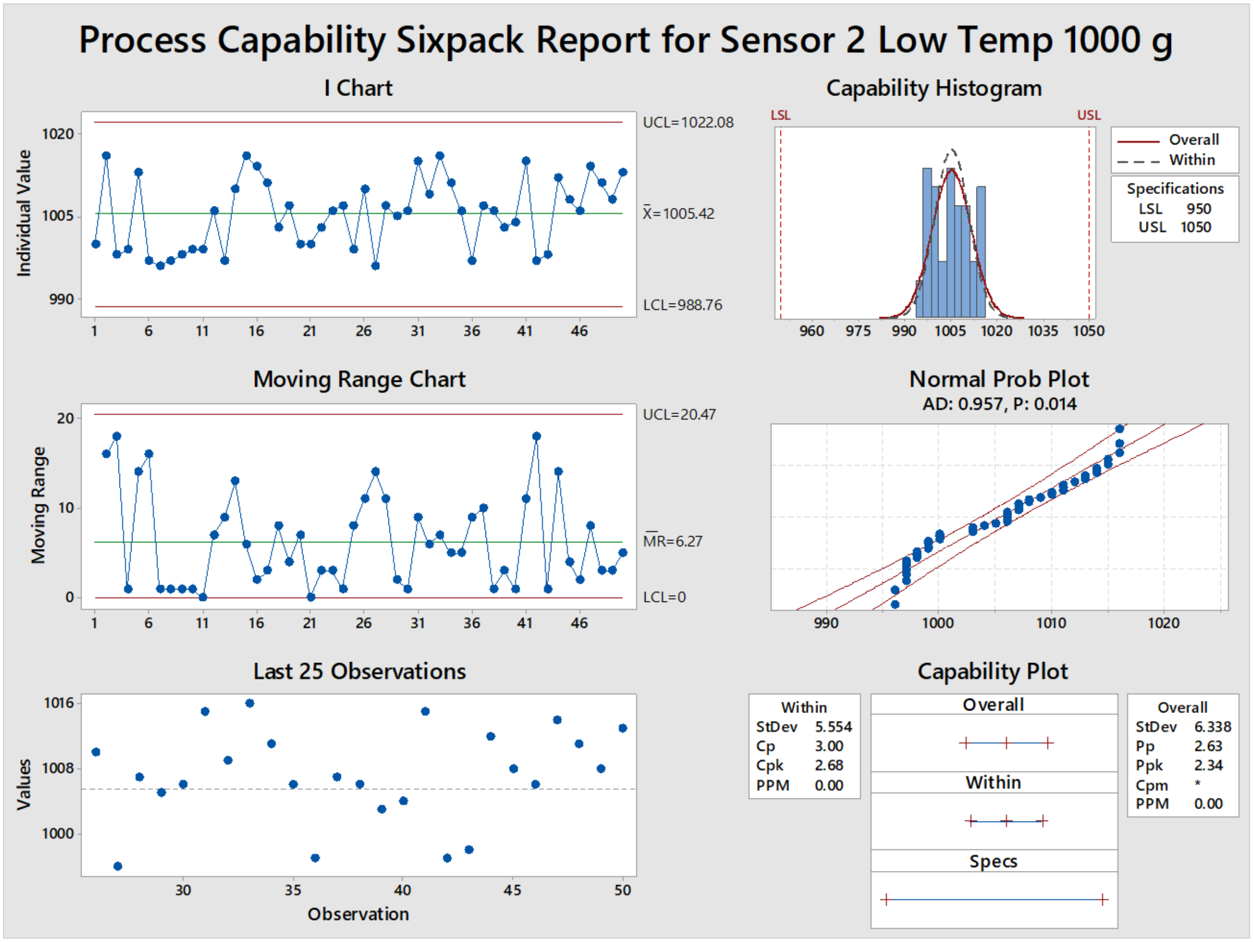

Process Capability Sixpack Report for Sensor 2 High Temp 500 g
I Chart
Individual Value
525
510
495
UCL=523.13
X̄=505.06
LCL=486.99
1 6 11 16 21 26 31 36 41 46
Capability Histogram
LSL
USL
Overall
Within
Specifications
LSL 470
USL 530
472 480 488 496 504 512 520 528
Moving Range Chart
Moving Range
20
10
0
UCL=22.20
MR̄=6.80
LCL=0
Normal Prob Plot
AD: 1.365, P: < 0.005
490 500 510 520
Last 25 Observations
Values
512
504
496
30 35 40 45 50
Observation
Capability Plot
Within
StDev 6.025
Cp 1.66
Cpk 1.38
PPM 17.40
Overall
Within
Specs
Overall
StDev 5.698
Pp 1.76
Ppk 1.46
Cpm *
PPM 6.01


Process Capability Sixpack Report for Sensor 2 Low Temp 500 g
I Chart
Individual Value
UCL=517.48
X̄=504.94
LCL=492.40
Capability Histogram
LSL
USL
Overall
Within
Specifications
LSL 470
USL 530
Moving Range Chart
Moving Range
UCL=15.40
M̄R=4.71
LCL=0
Normal Prob Plot
AD: 0.743, P: 0.050
Last 25 Observations
Values
Observation
Capability Plot
Within
StDev 4.179
Cp 2.39
Cpk 2.00
PPM 0.00
Overall
Within
Specs
Overall
StDev 4.863
Pp 2.06
Ppk 1.72
Cpm *
PPM 0.13

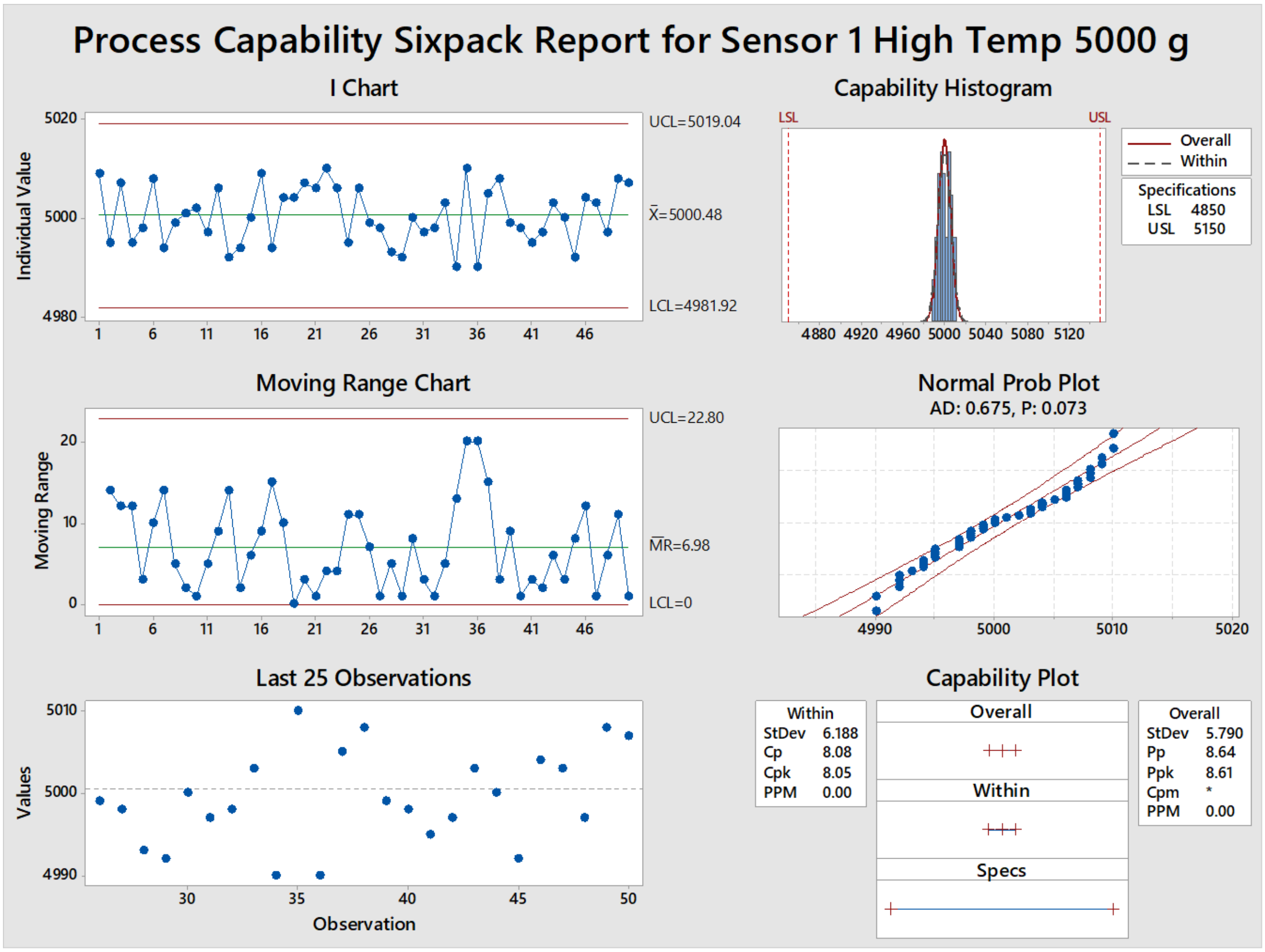

Process Capability Sixpack Report for Sensor 1 High Temp 1000 g
I Chart
UCL=1024.67
X̄=1006.16
LCL=987.65
Individual Value
Capability Histogram
LSL
USL
Overall
Within
Specifications
LSL 950
USL 1050
Moving Range Chart
UCL=22.74
MR̄=6.96
LCL=0
Moving Range
Normal Prob Plot
AD: 1.382, P: < 0.005
Last 25 Observations
Values
Observation
Capability Plot
Within
StDev 6.169
Cp 2.70
Cpk 2.37
PPM 0.00
Overall
Within
Specs
Overall
StDev 5.801
Pp 2.87
Ppk 2.52
Cpm *
PPM 0.00


Process Capability Sixpack Report for Sensor 1 Low Temp 1000 g
I Chart
Individual Value
UCL=1023.32
X̄=1004.76
LCL=986.20
Capability Histogram
LSL
USL
Overall
Within
Specifications
LSL 950
USL 1050
Moving Range Chart
Moving Range
UCL=22.80
MR̄=6.98
LCL=0
Normal Prob Plot
AD: 0.736, P: 0.051
Last 25 Observations
Values
Observation
Capability Plot
Within
StDev 6.188
Cp 2.69
Cpk 2.44
PPM 0.00
Overall
Within
Specs
Overall
StDev 6.143
Pp 2.71
Ppk 2.45
Cpm *
PPM 0.00

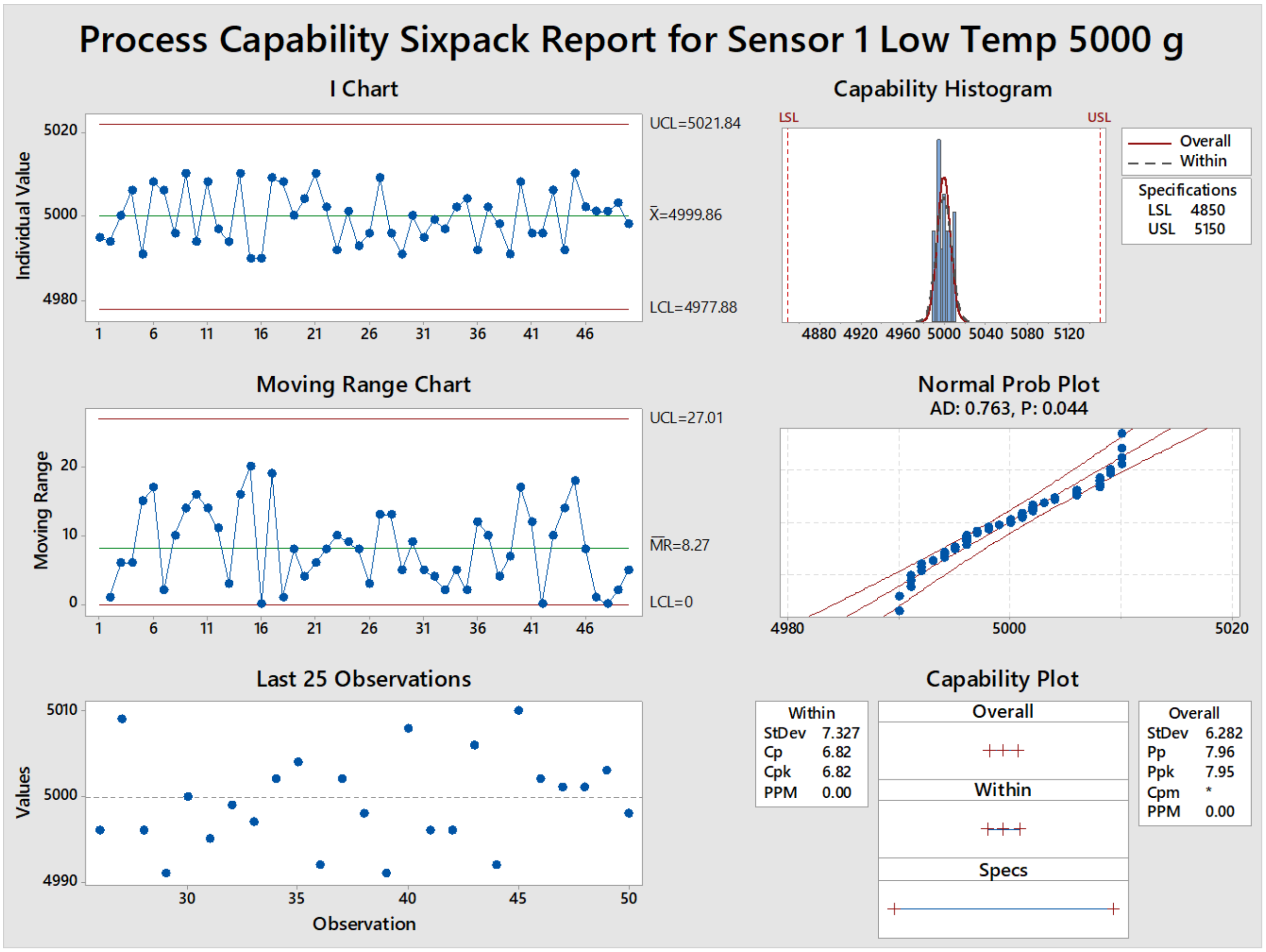

Process Capability Sixpack Report for Sensor 2 High Temp 1000 g
I Chart
UCL=1023.14
X̄=1004.2
LCL=985.26
Individual Value
Capability Histogram
LSL
USL
Overall
Within
Specifications
LSL 950
USL 1050
Moving Range Chart
UCL=23.27
MR̄=7.12
LCL=0
Moving Range
Normal Prob Plot
AD: 1.277, P: < 0.005
Last 25 Observations
Values
Observation
Capability Plot
Within
StDev 6.314
Cp 2.64
Cpk 2.42
PPM 0.00
Overall
Within
Specs
Overall
StDev 6.308
Pp 2.64
Ppk 2.42
Cpm *
PPM 0.00

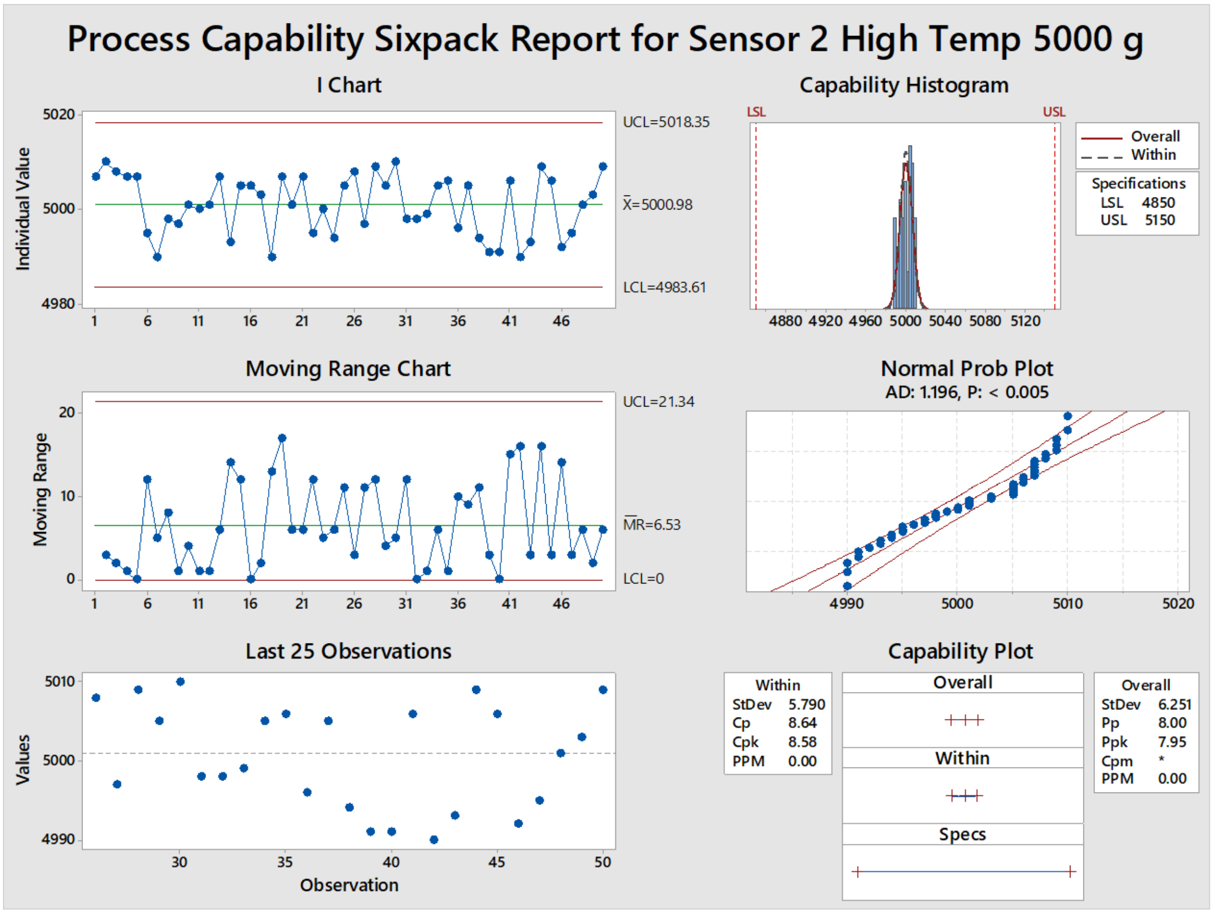


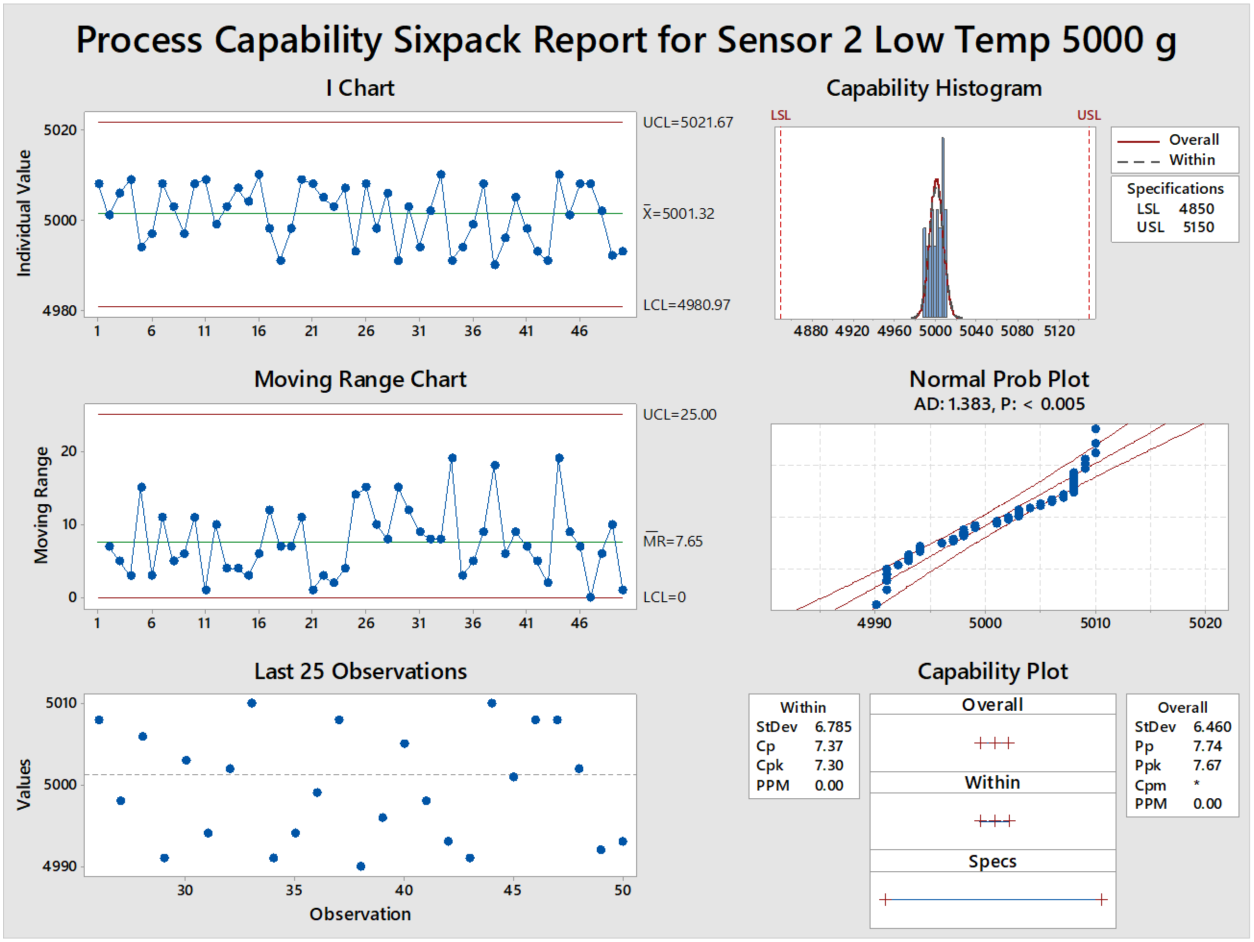
Process Capability Sixpack Report for Sensor 2 Low Temp 1000 g
I Chart
Individual Value
UCL=1022.08
X̄=1005.42
LCL=988.76
Capability Histogram
LSL
USL
Overall
Within
Specifications
LSL 950
USL 1050
Moving Range Chart
Moving Range
UCL=20.47
MR̄=6.27
LCL=0
Normal Prob Plot
AD: 0.957, P: 0.014
Last 25 Observations
Values
Observation
Capability Plot
Within
StDev 5.554
Cp 3.00
Cpk 2.68
PPM 0.00
Overall
Within
Specs
Overall
StDev 6.338
Pp 2.63
Ppk 2.34
Cpm *
PPM 0.00

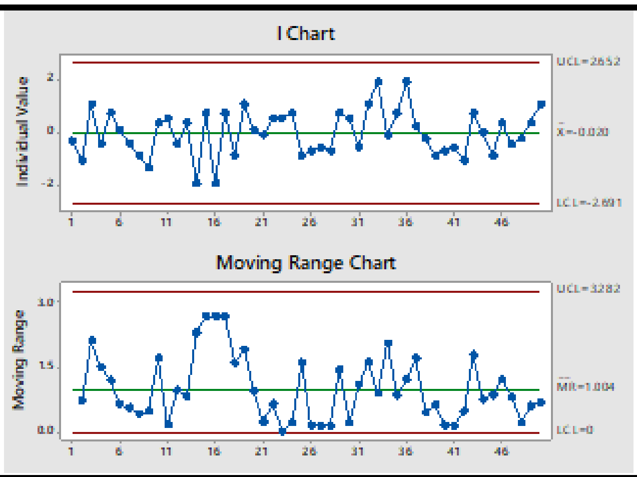
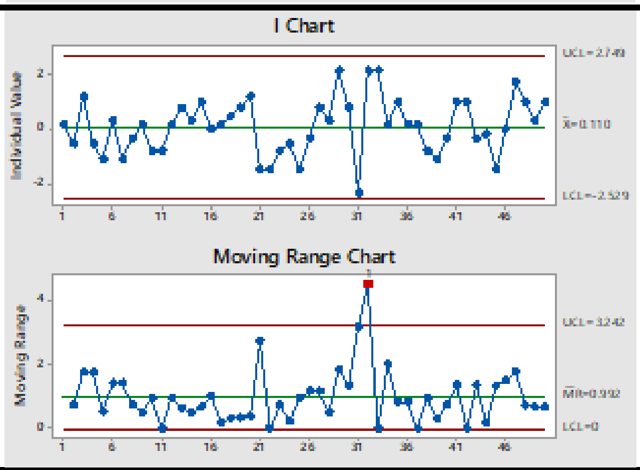
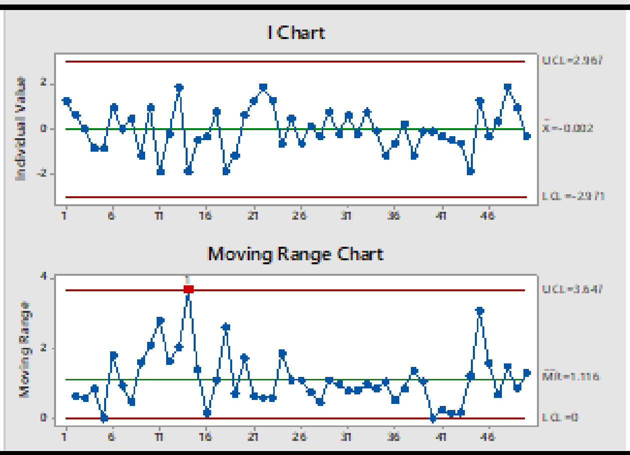
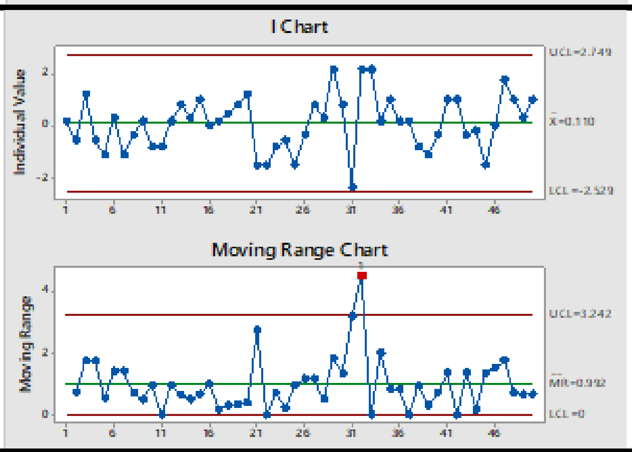


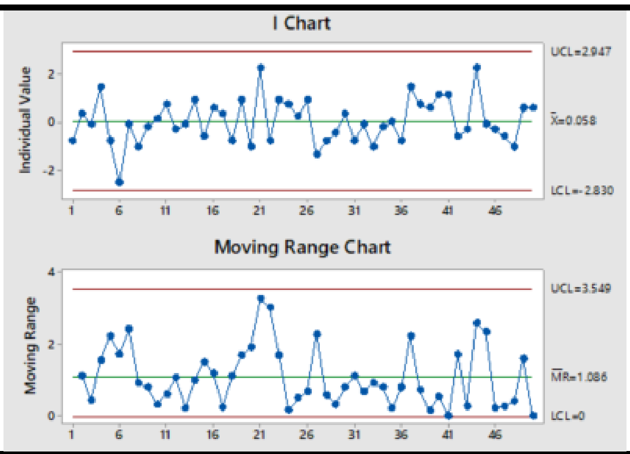
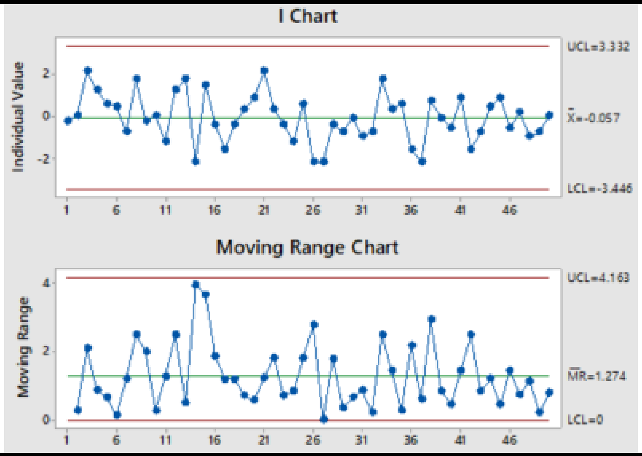
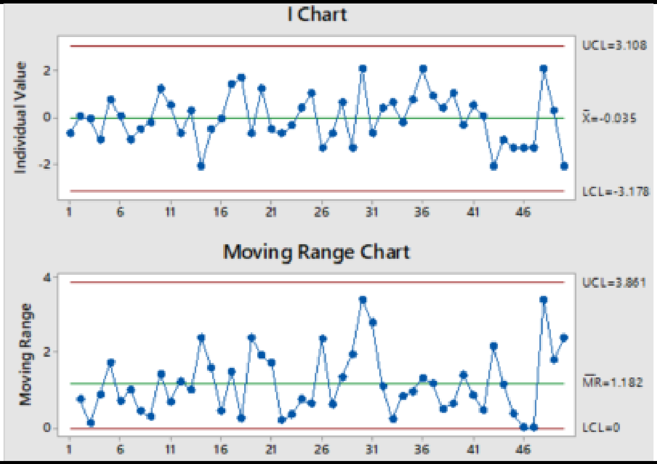
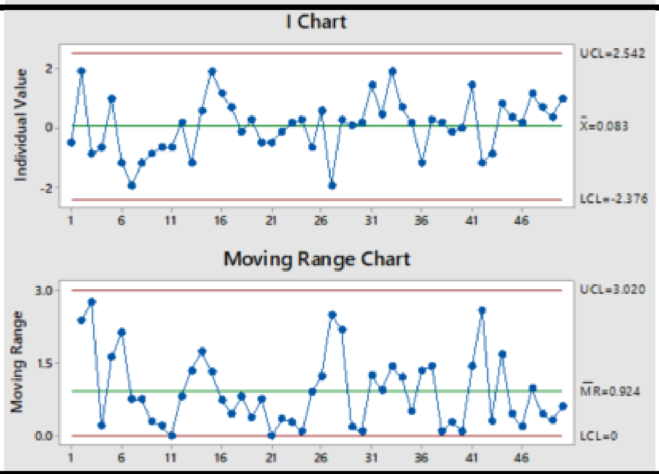
Process Capability Sixpack Report for Sensor 1 High Temp 5000 g
I Chart
Individual Value
UCL=5019.04
X̄=5000.48
LCL=4981.92
Capability Histogram
LSL
USL
Overall
Within
Specifications
LSL 4850
USL 5150
Moving Range Chart
Moving Range
UCL=22.80
M̄R=6.98
LCL=0
Normal Prob Plot
AD: 0.675, P: 0.073
Last 25 Observations
Values
Observation
Capability Plot
Within
StDev 6.188
Cp 8.08
Cpk 8.05
PPM 0.00
Overall
Within
Specs
Overall
StDev 5.790
Pp 8.64
Ppk 8.61
Cpm *
PPM 0.00


Process Capability Sixpack Report for Sensor 1 Low Temp 5000 g
I Chart
Individual Value
UCL=5021.84
X̄=4999.86
LCL=4977.88
Capability Histogram
LSL
USL
Overall
Within
Specifications
LSL 4850
USL 5150
Moving Range Chart
Moving Range
UCL=27.01
MR̄=8.27
LCL=0
Normal Prob Plot
AD: 0.763, P: 0.044
Last 25 Observations
Values
Observation
Capability Plot
Within
StDev 7.327
Cp 6.82
Cpk 6.82
PPM 0.00
Overall
Within
Specs
Overall
StDev 6.282
Pp 7.96
Ppk 7.95
Cpm *
PPM 0.00

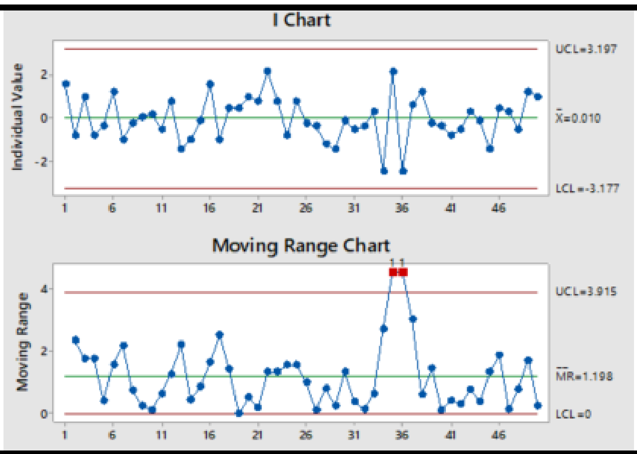
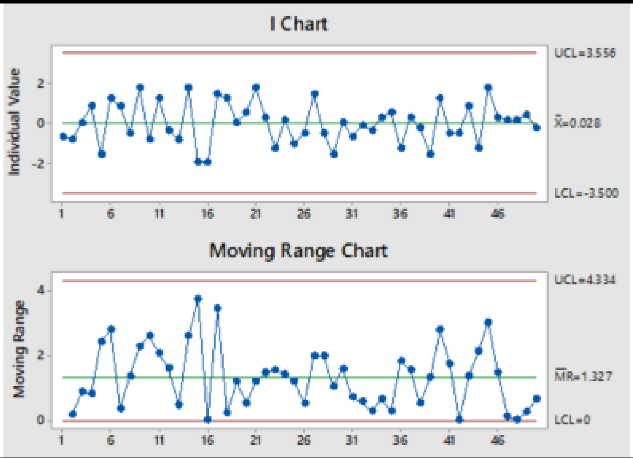
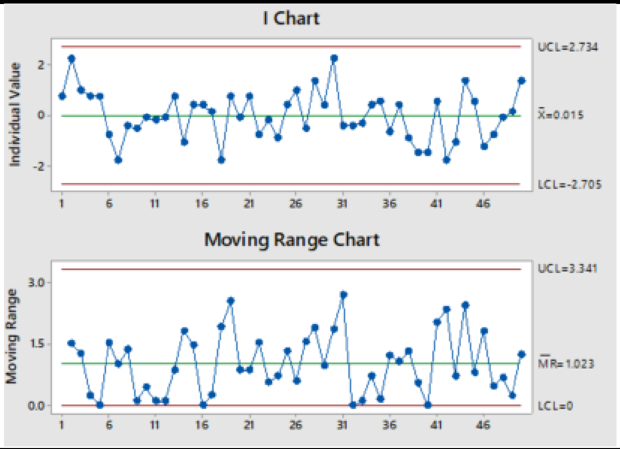
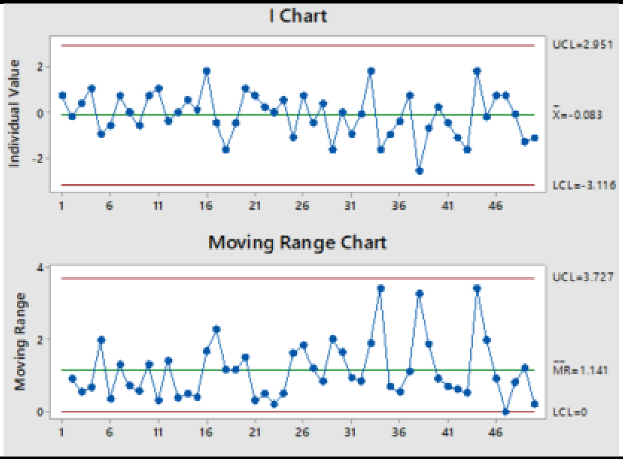

Process Capability Sixpack Report for Sensor 2 High Temp 5000 g
I Chart
Individual Value
UCL=5018.35
X̄=5000.98
LCL=4983.61
Capability Histogram
LSL
USL
Overall
Within
Specifications
LSL 4850
USL 5150
Moving Range Chart
Moving Range
UCL=21.34
MR̄=6.53
LCL=0
Normal Prob Plot
AD: 1.196, P: < 0.005
Last 25 Observations
Values
Observation
Capability Plot
Within
StDev 5.790
Cp 8.64
Cpk 8.58
PPM 0.00
Overall
Within
Specs
Overall
StDev 6.251
Pp 8.00
Ppk 7.95
Cpm *
PPM 0.00

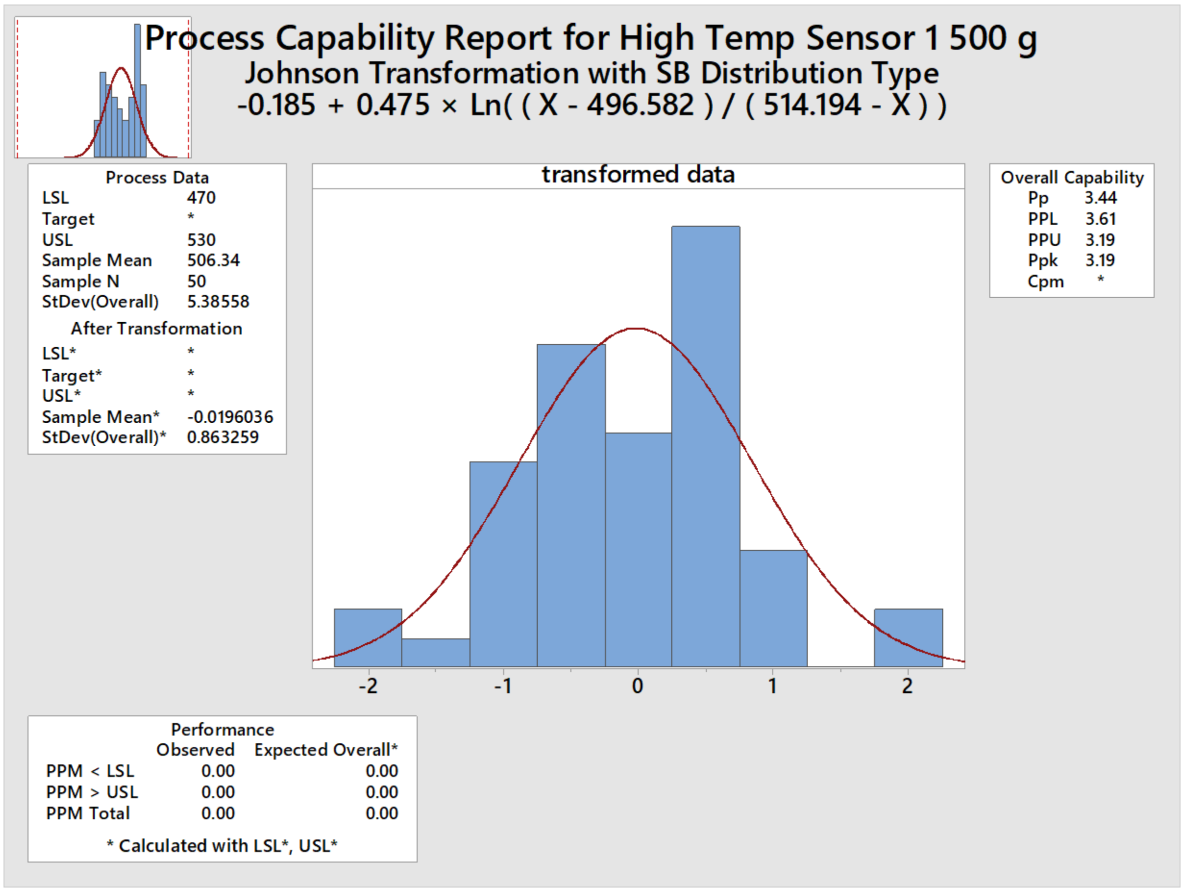

Process Capability Sixpack Report for Sensor 2 Low Temp 5000 g
I Chart
Individual Value
UCL=5021.67
X̄=5001.32
LCL=4980.97
Capability Histogram
LSL
USL
Overall
Within
Specifications
LSL 4850
USL 5150
Moving Range Chart
Moving Range
UCL=25.00
MR̄=7.65
LCL=0
Normal Prob Plot
AD: 1.383, P: < 0.005
Last 25 Observations
Values
Observation
Capability Plot
Within
StDev 6.785
Cp 7.37
Cpk 7.30
PPM 0.00
Overall
Within
Specs
Overall
StDev 6.460
Pp 7.74
Ppk 7.67
Cpm *
PPM 0.00

500 (g)- Transformed	Temperature	
Sensor 1	High	
	Low	
Sensor 2	High	
	Low	
1000 (g)- Transformed	**Temperature**	

Sensor 1	High	
	Low	
Sensor 2	High	
	Low	

	Temperature	

5000 (g)-Transformed		
Sensor 1	High	
	Low	
Sensor 2	High	
	Low	



Process Capability Report for High Temp Sensor 1 500 g

Johnson Transformation with SB Distribution Type

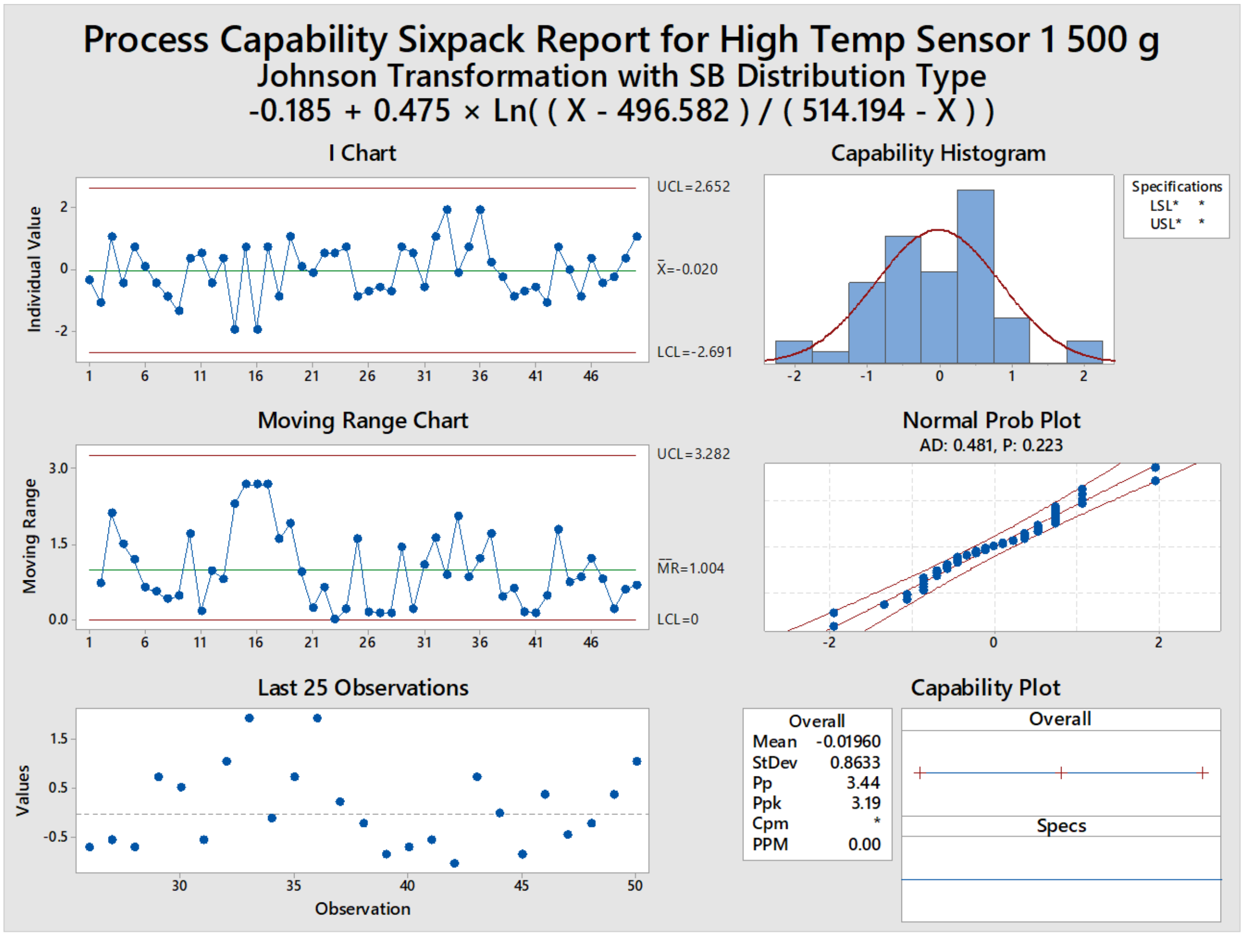
-0.185 + 0.475 × Ln((X - 496.582) / (514.194 - X))

Process Data	
LSL	470
Target	*
USL	530
Sample Mean	506.34
Sample N	50
StDev(Overall)	5.38558
After Transformation	
LSL*	*
Target*	*
USL*	*
Sample Mean*	-0.0196036
StDev(Overall)*	0.863259

Overall Capability	
Pp	3.44
PPL	3.61
PPU	3.19
Ppk	3.19
Cpm	*

Performance	Observed	Expected Overall*
PPM < LSL	0.00	0.00
PPM > USL	0.00	0.00
PPM Total	0.00	0.00

* Calculated with LSL*, USL*



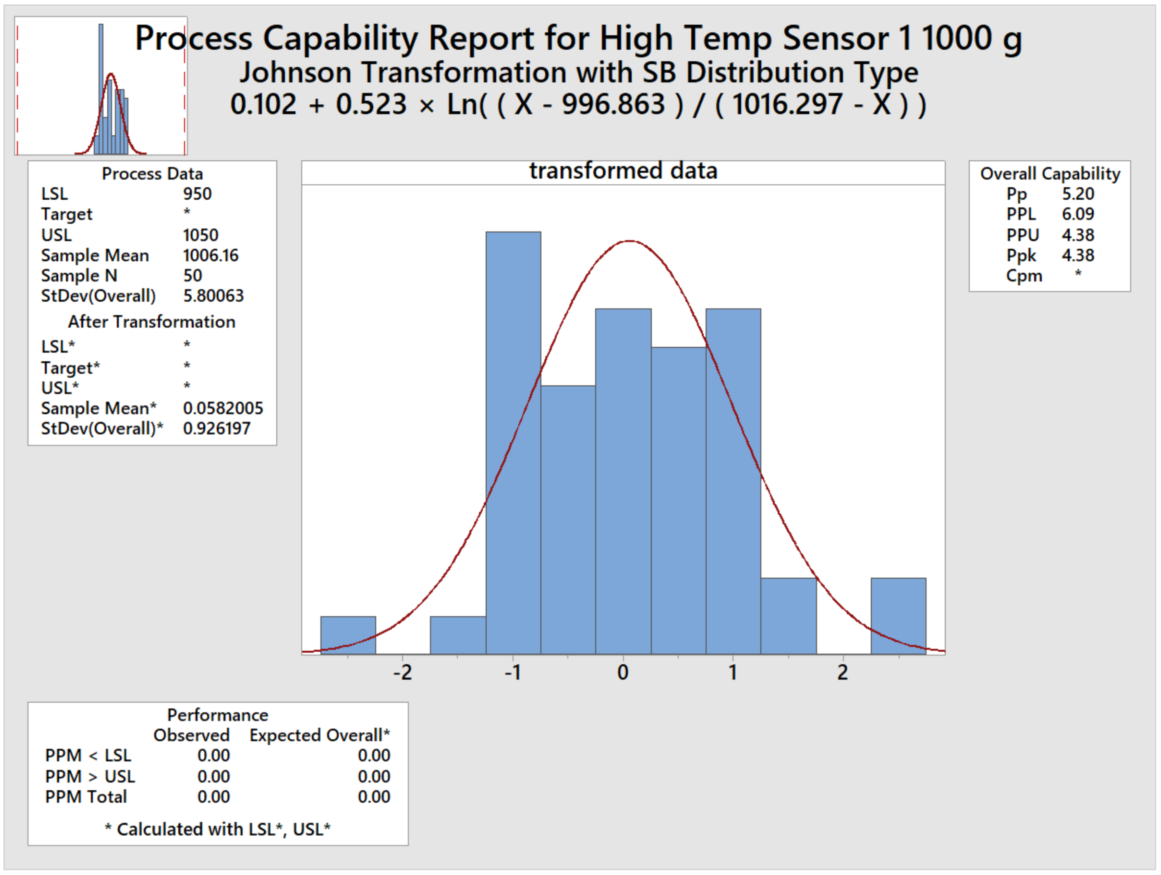
Process Capability Sixpack Report for High Temp Sensor 1 500 g

Johnson Transformation with SB Distribution Type

-0.185 + 0.475 × Ln((X - 496.582) / (514.194 - X))

Specifications	
LSL*	*
USL*	*

Overall	
Mean	-0.01960
StDev	0.8633
Pp	3.44
Ppk	3.19
Cpm	*
PPM	0.00



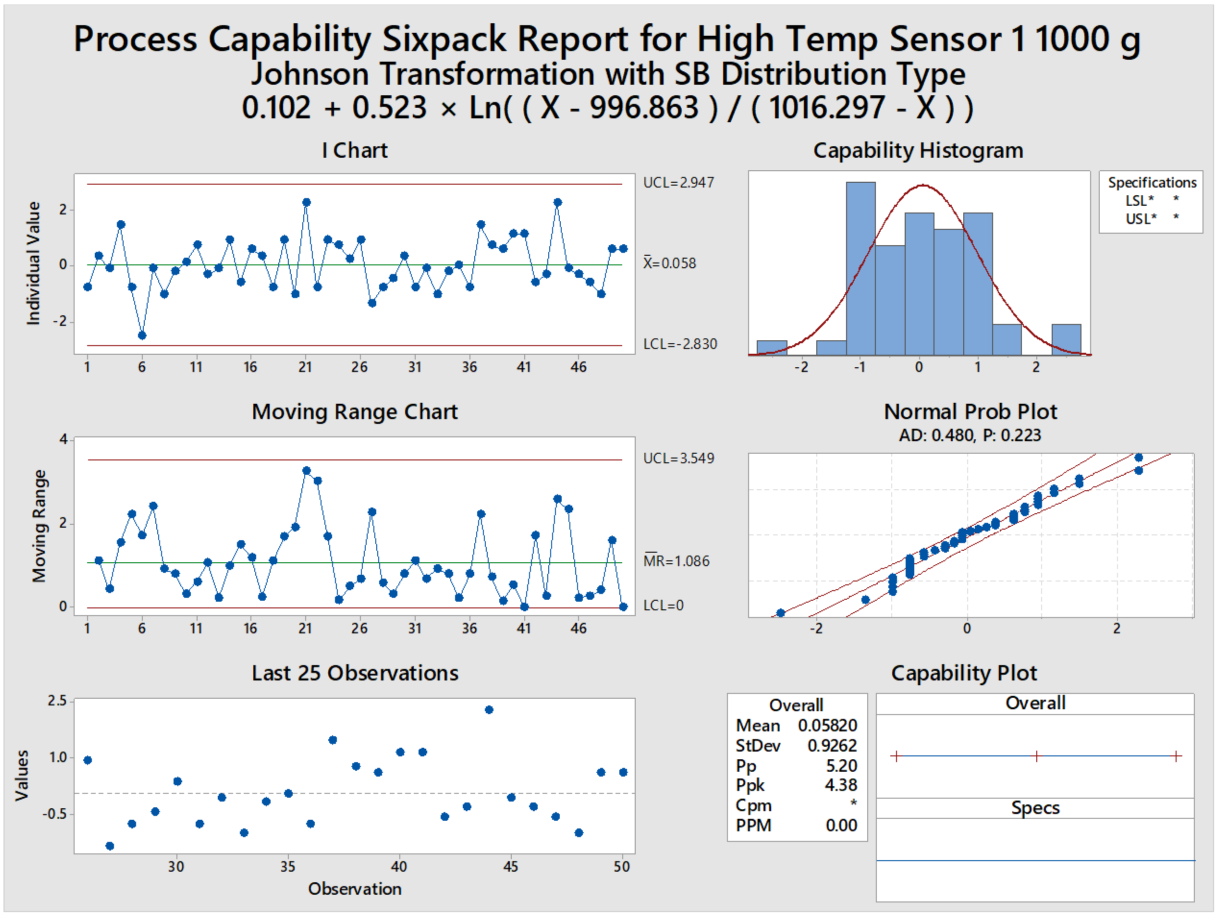
Process Capability Report for High Temp Sensor 1 1000 g
Johnson Transformation with SB Distribution Type
0.102 + 0.523 × Ln((X - 996.863) / (1016.297 - X))

Process Data	
LSL	950
Target	*
USL	1050
Sample Mean	1006.16
Sample N	50
StDev(Overall)	5.80063
After Transformation	
LSL*	*
Target*	*
USL*	*
Sample Mean*	0.0582005
StDev(Overall)*	0.926197

Overall Capability	
Pp	5.20
PPL	6.09
PPU	4.38
Ppk	4.38
Cpm	*

Performance	Observed	Expected Overall*
PPM < LSL	0.00	0.00
PPM > USL	0.00	0.00
PPM Total	0.00	0.00

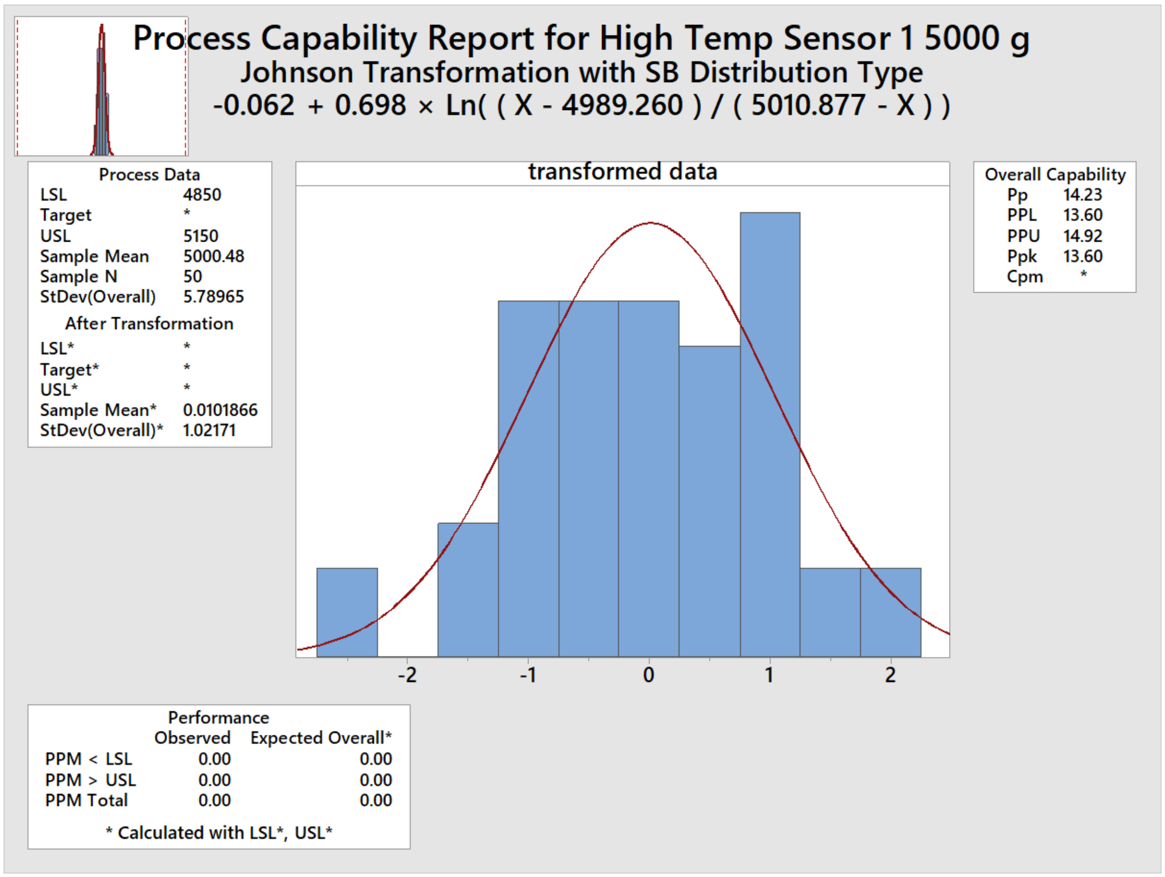
* Calculated with LSL*, USL*



Process Capability Sixpack Report for High Temp Sensor 1 1000 g
Johnson Transformation with SB Distribution Type
0.102 + 0.523 × Ln((X - 996.863) / (1016.297 - X))

Specifications	
LSL*	*
USL*	*

Overall	
Mean	0.05820
StDev	0.9262
Pp	5.20
Ppk	4.38
Cpm	*
PPM	0.00



Process Data

LSL	4850
Target	*
USL	5150
Sample Mean	5000.48
Sample N	50
StDev(Overall)	5.78965

After Transformation

LSL*	*
Target*	*
USL*	*
Sample Mean*	0.0101866
StDev(Overall)*	1.02171

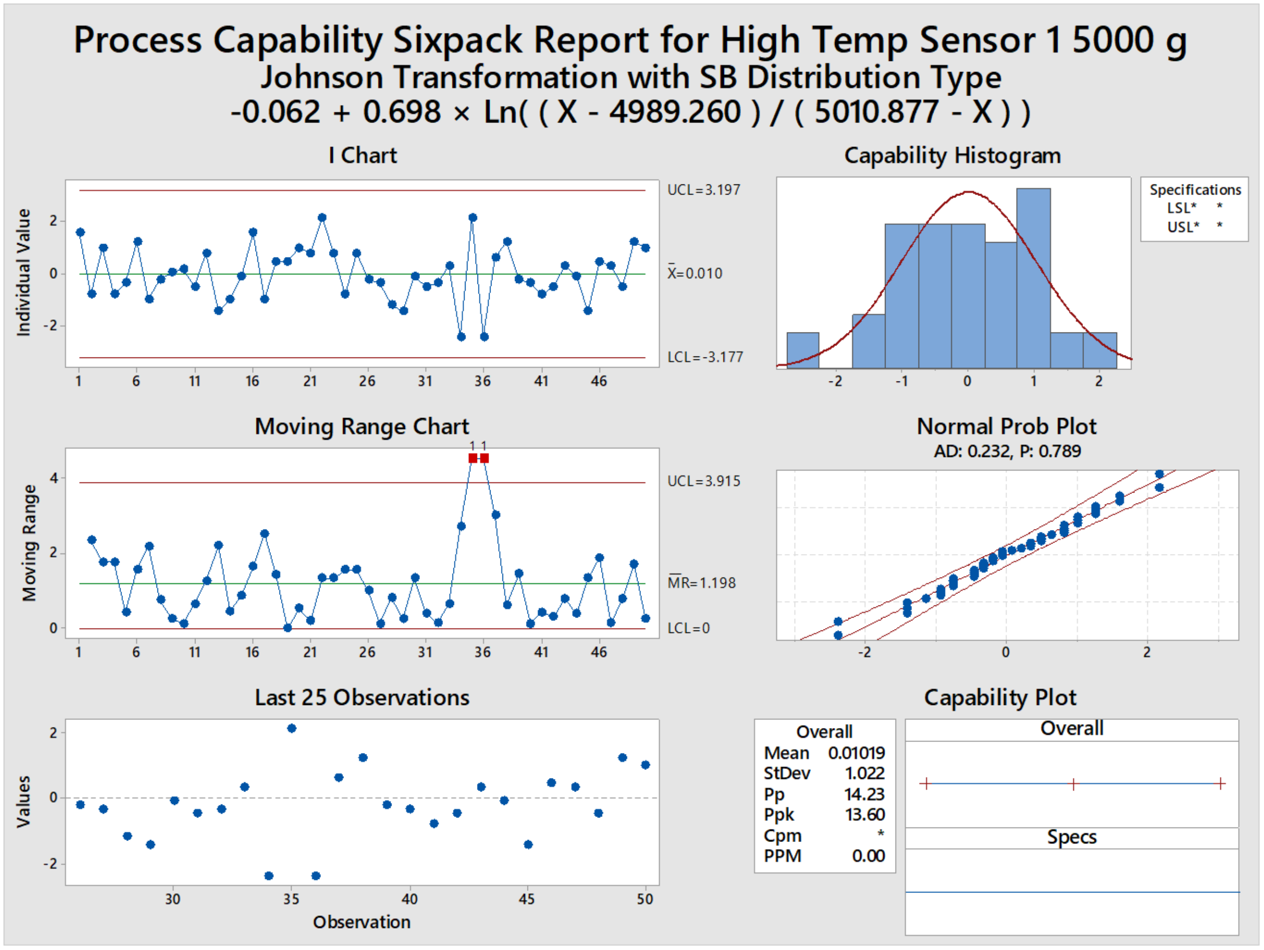
Overall Capability

Pp	14.23
PPL	13.60
PPU	14.92
Ppk	13.60
Cpm	*

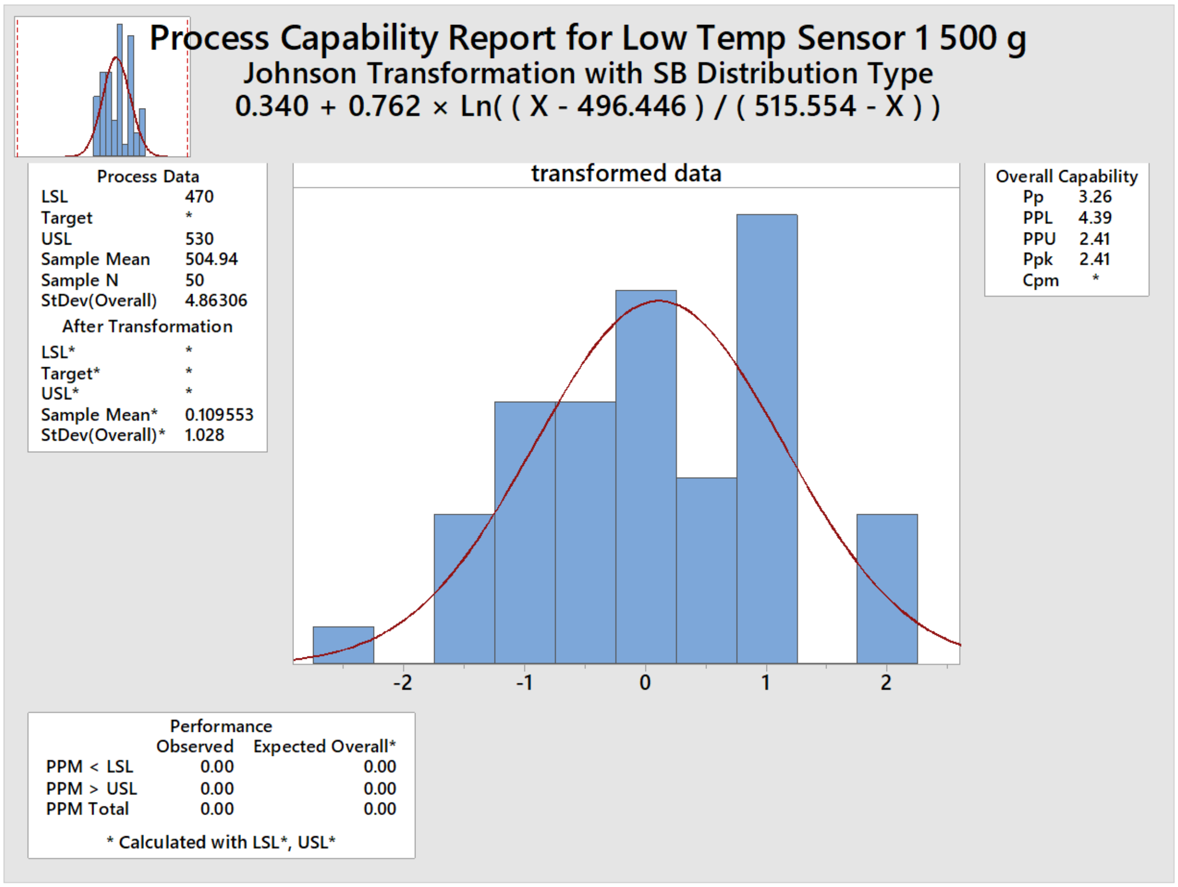
Performance

	Observed	Expected Overall*
PPM < LSL	0.00	0.00
PPM > USL	0.00	0.00
PPM Total	0.00	0.00

* Calculated with LSL*, USL*



Overall	
Mean	0.01019
StDev	1.022
Pp	14.23
Ppk	13.60
Cpm	*
PPM	0.00



Process Data

LSL	470
Target	*
USL	530
Sample Mean	504.94
Sample N	50
StDev(Overall)	4.86306

After Transformation

LSL*	*
Target*	*
USL*	*
Sample Mean*	0.109553
StDev(Overall)*	1.028

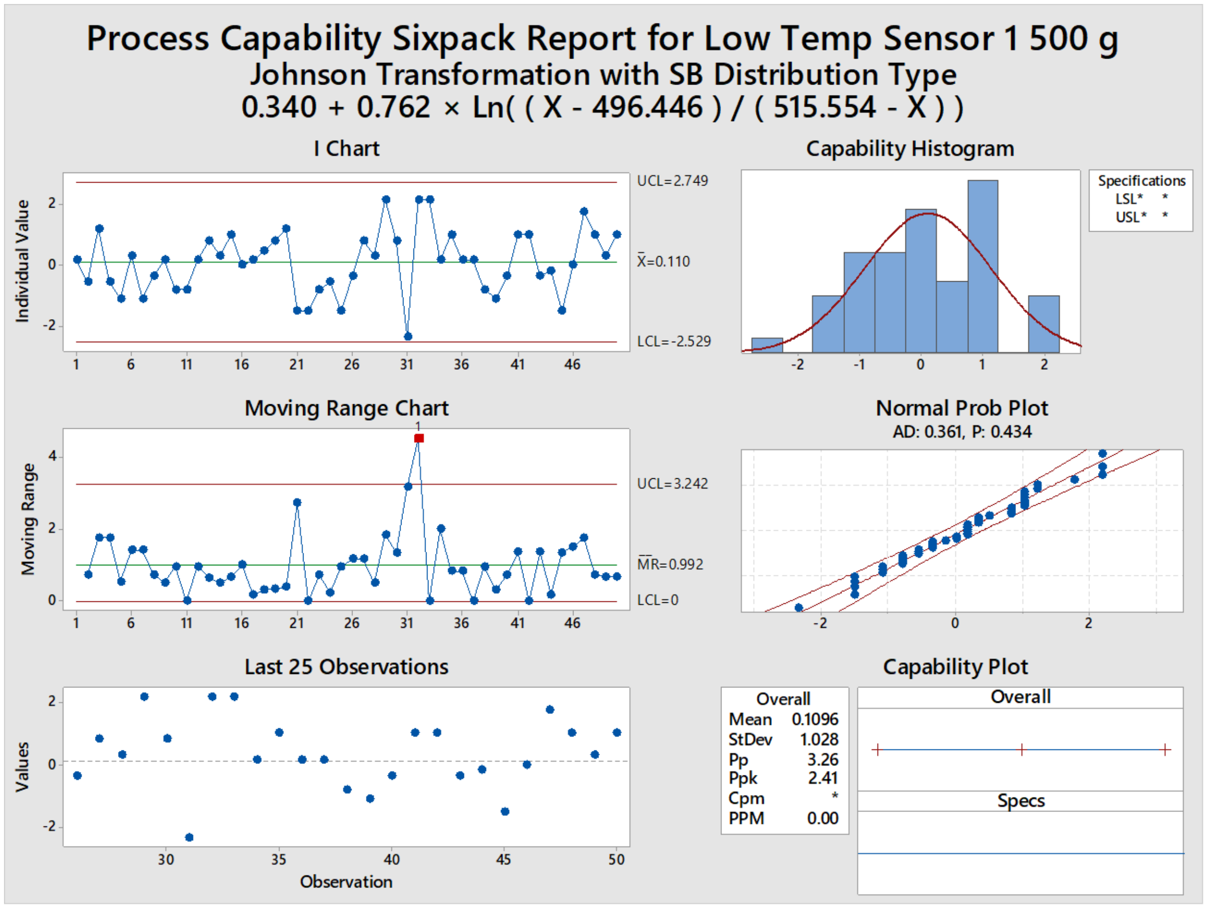
Overall Capability

Pp	3.26
PPL	4.39
PPU	2.41
Ppk	2.41
Cpm	*

Performance

	Observed	Expected Overall*
PPM < LSL	0.00	0.00
PPM > USL	0.00	0.00
PPM Total	0.00	0.00

* Calculated with LSL*, USL*



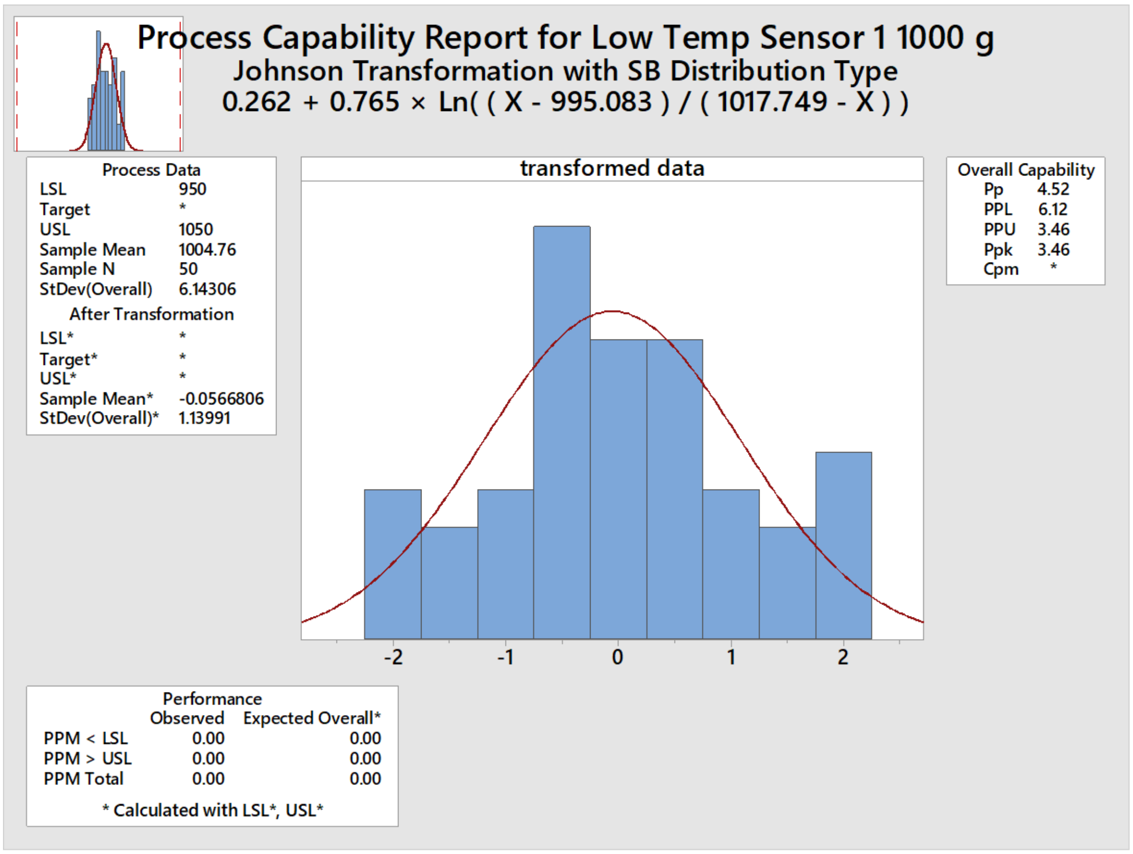
Specifications

LSL*	*
USL*	*

Overall

Mean	0.1096
StDev	1.028
Pp	3.26
Ppk	2.41
Cpm	*
PPM	0.00

C

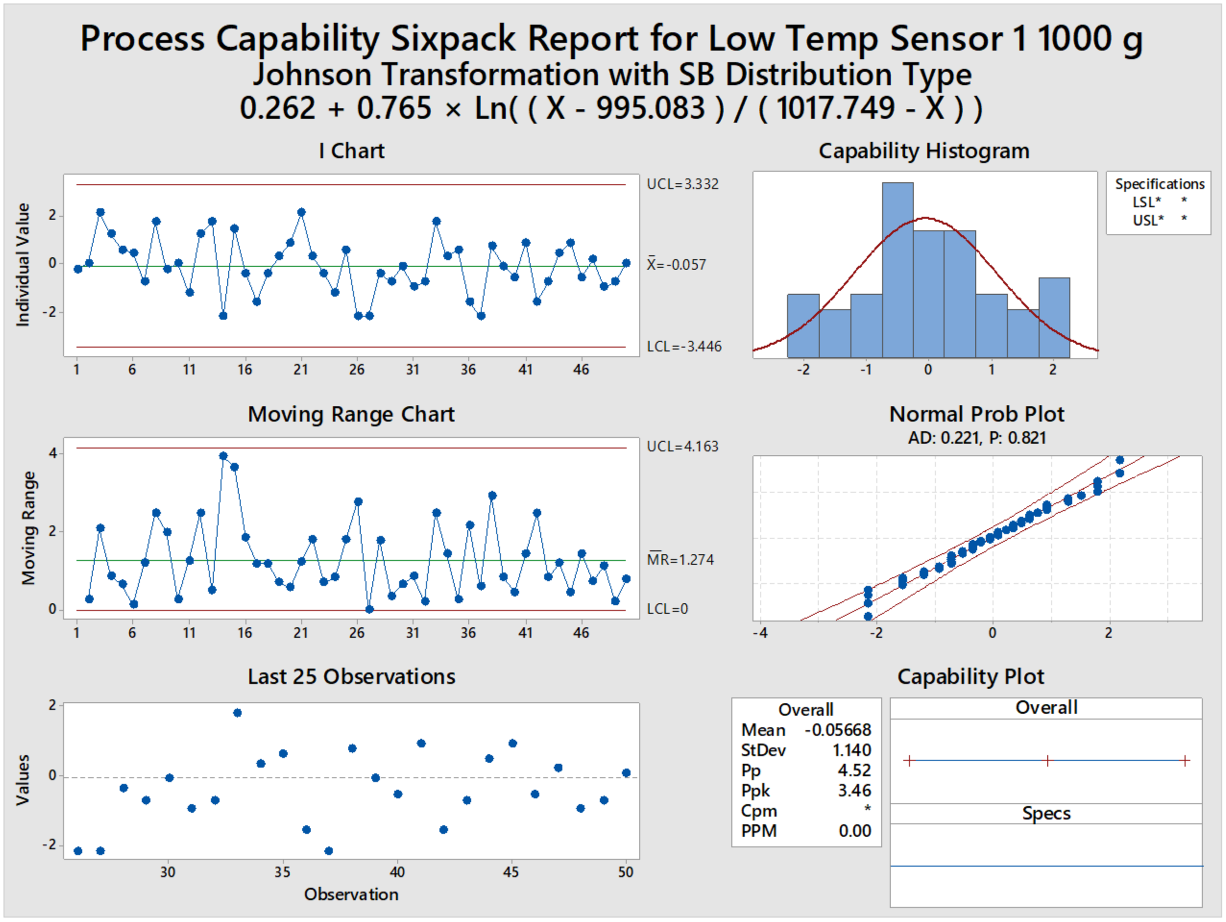


Process Data	
LSL	950
Target	*
USL	1050
Sample Mean	1004.76
Sample N	50
StDev(Overall)	6.14306
After Transformation	
LSL*	*
Target*	*
USL*	*
Sample Mean*	-0.0566806
StDev(Overall)*	1.13991

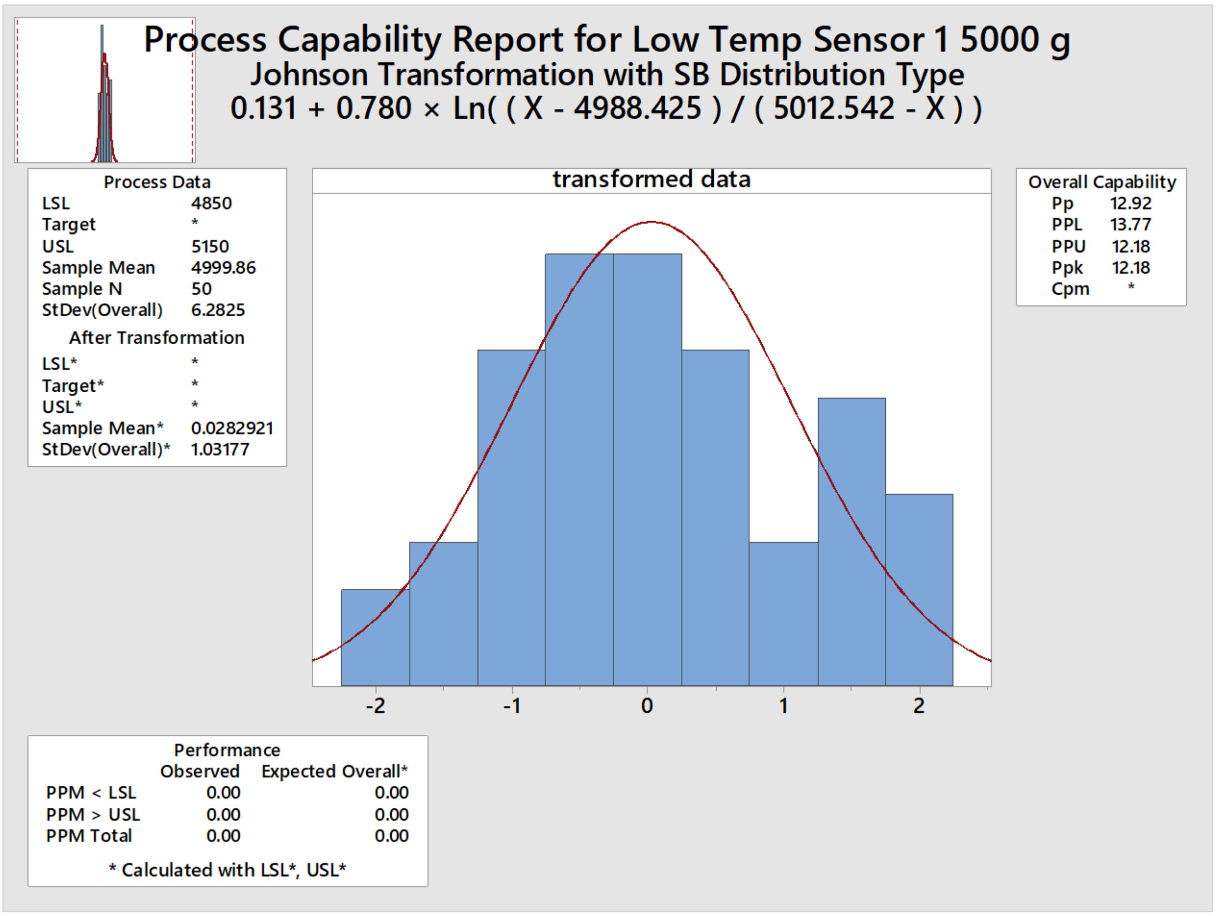
Overall Capability	
Pp	4.52
PPL	6.12
PPU	3.46
Ppk	3.46
Cpm	*

Performance	Observed	Expected Overall*
PPM < LSL	0.00	0.00
PPM > USL	0.00	0.00
PPM Total	0.00	0.00

* Calculated with LSL*, USL*



Overall	
Mean	-0.05668
StDev	1.140
Pp	4.52
Ppk	3.46
Cpm	*
PPM	0.00



Process Capability Report for Low Temp Sensor 1 5000 g
Johnson Transformation with SB Distribution Type
0.131 + 0.780 × Ln((X - 4988.425) / (5012.542 - X))

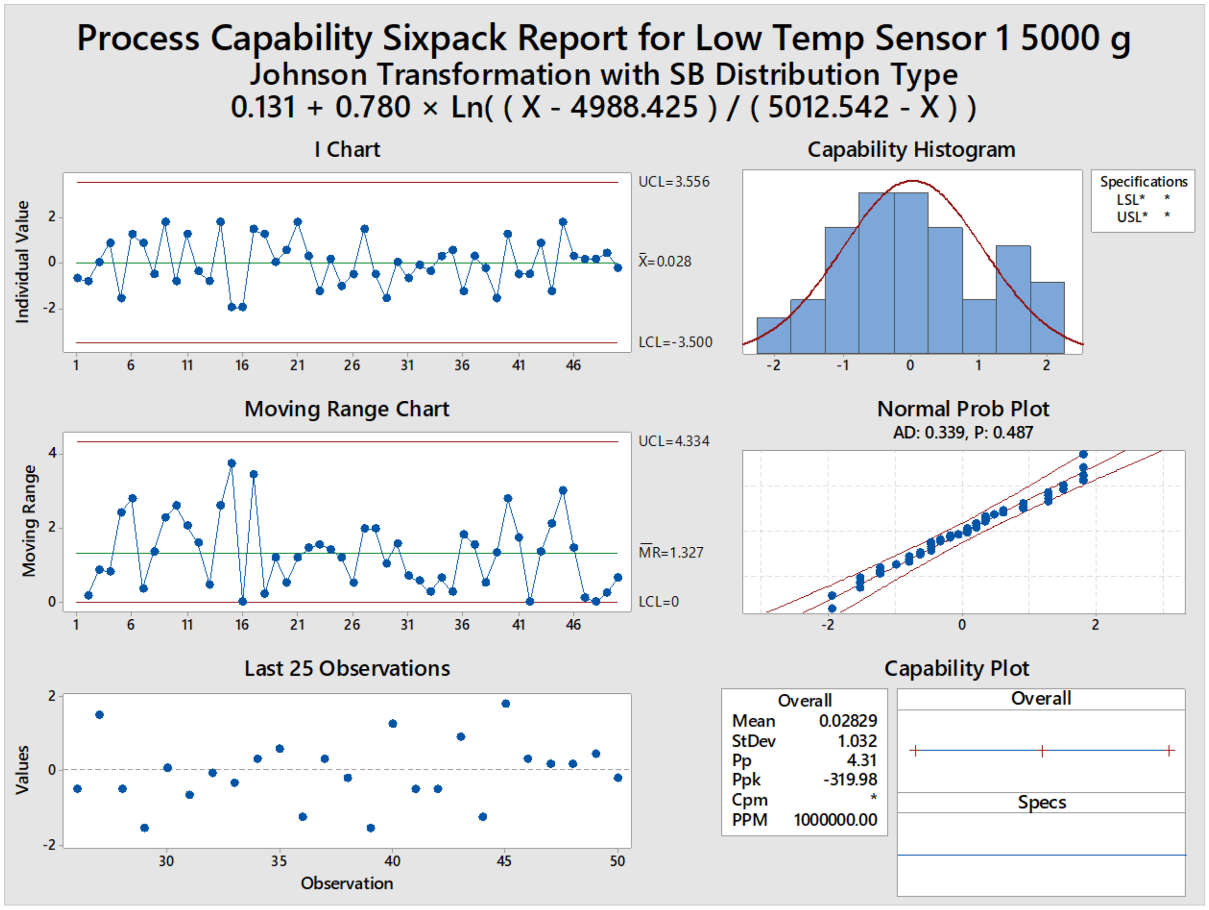
Process Data	
LSL	4850
Target	*
USL	5150
Sample Mean	4999.86
Sample N	50
StDev(Overall)	6.2825
After Transformation	
LSL*	*
Target*	*
USL*	*
Sample Mean*	0.0282921
StDev(Overall)*	1.03177

Overall Capability	
Pp	12.92
PPL	13.77
PPU	12.18
Ppk	12.18
Cpm	*

Performance

	Observed	Expected Overall*
PPM < LSL	0.00	0.00
PPM > USL	0.00	0.00
PPM Total	0.00	0.00

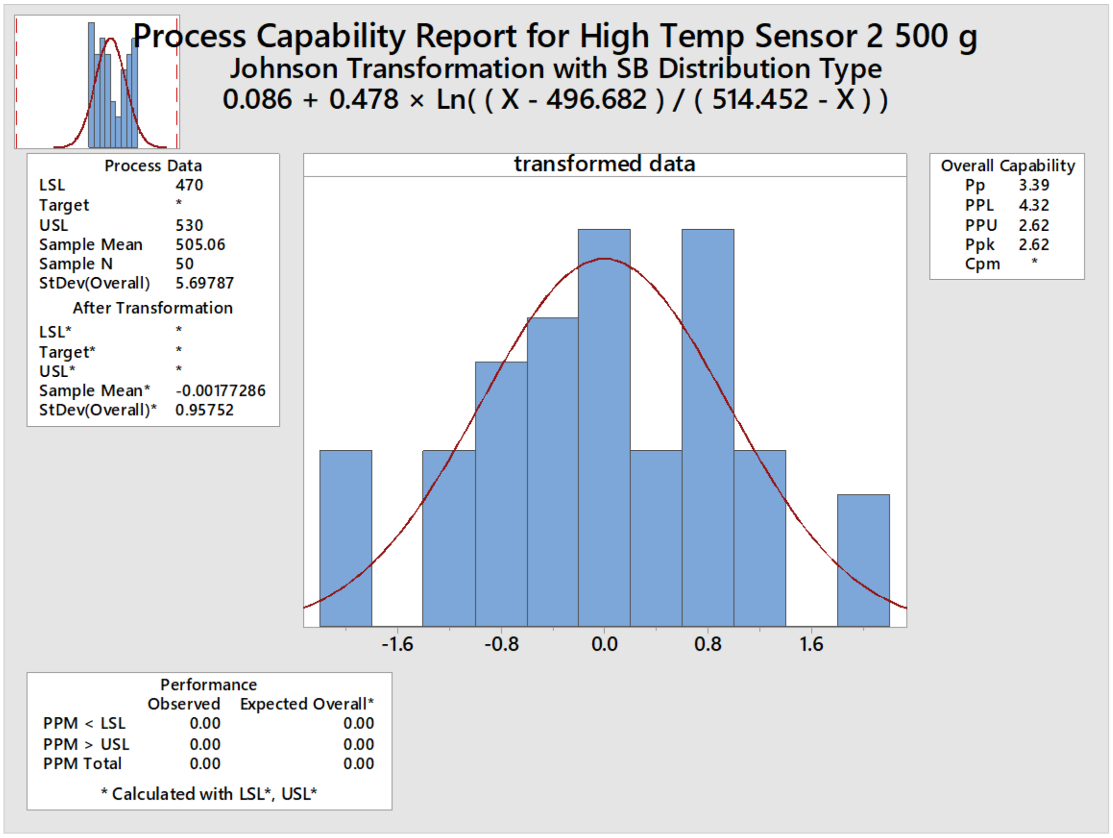
* Calculated with LSL*, USL*



Process Capability Sixpack Report for Low Temp Sensor 1 5000 g
Johnson Transformation with SB Distribution Type
0.131 + 0.780 × Ln((X - 4988.425) / (5012.542 - X))

Overall	
Mean	0.02829
StDev	1.032
Pp	4.31
Ppk	-319.98
Cpm	*
PPM	1000000.00

Specifications	
LSL*	*
USL*	*



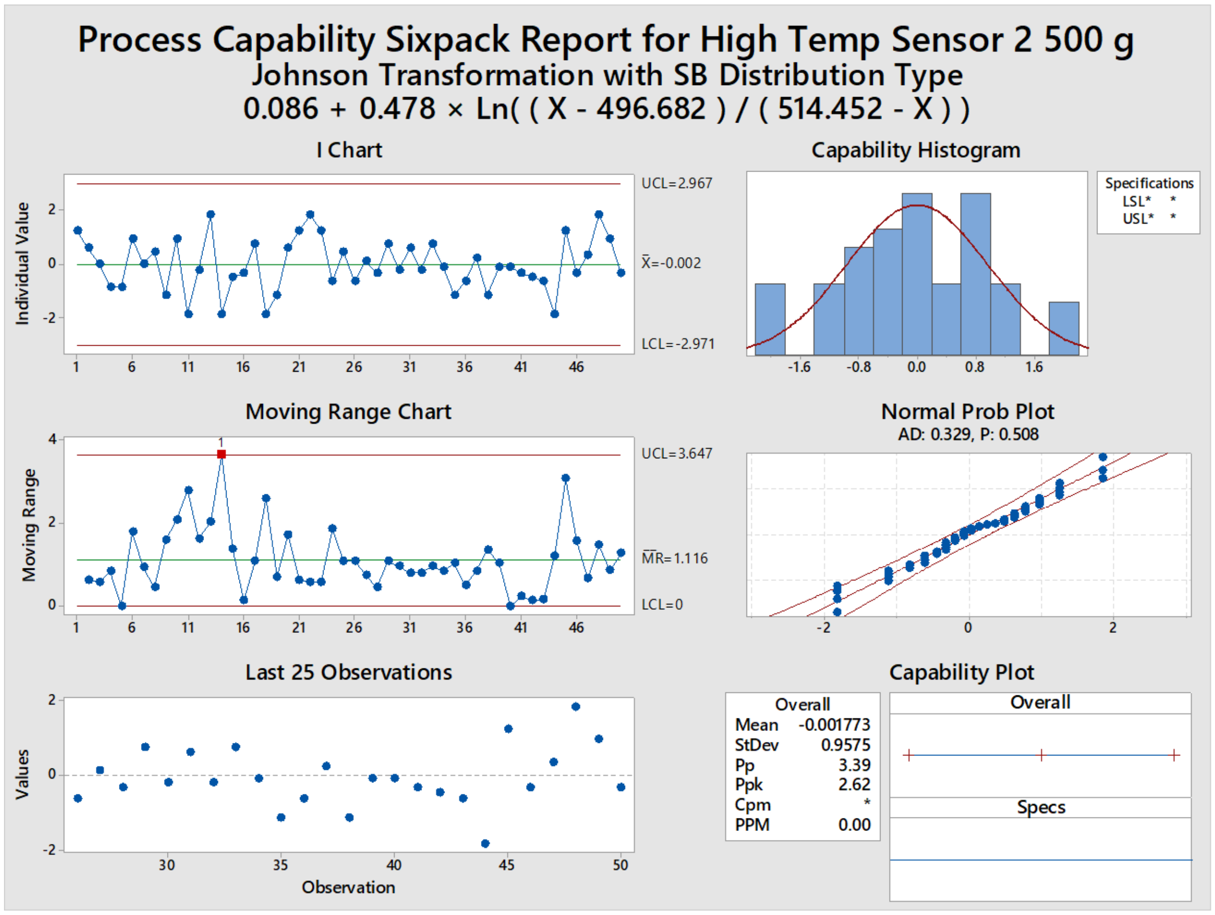
Process Data	
LSL	470
Target	*
USL	530
Sample Mean	505.06
Sample N	50
StDev(Overall)	5.69787
After Transformation	
LSL*	*
Target*	*
USL*	*
Sample Mean*	-0.00177286
StDev(Overall)*	0.95752

Overall Capability	
Pp	3.39
PPL	4.32
PPU	2.62
Ppk	2.62
Cpm	*

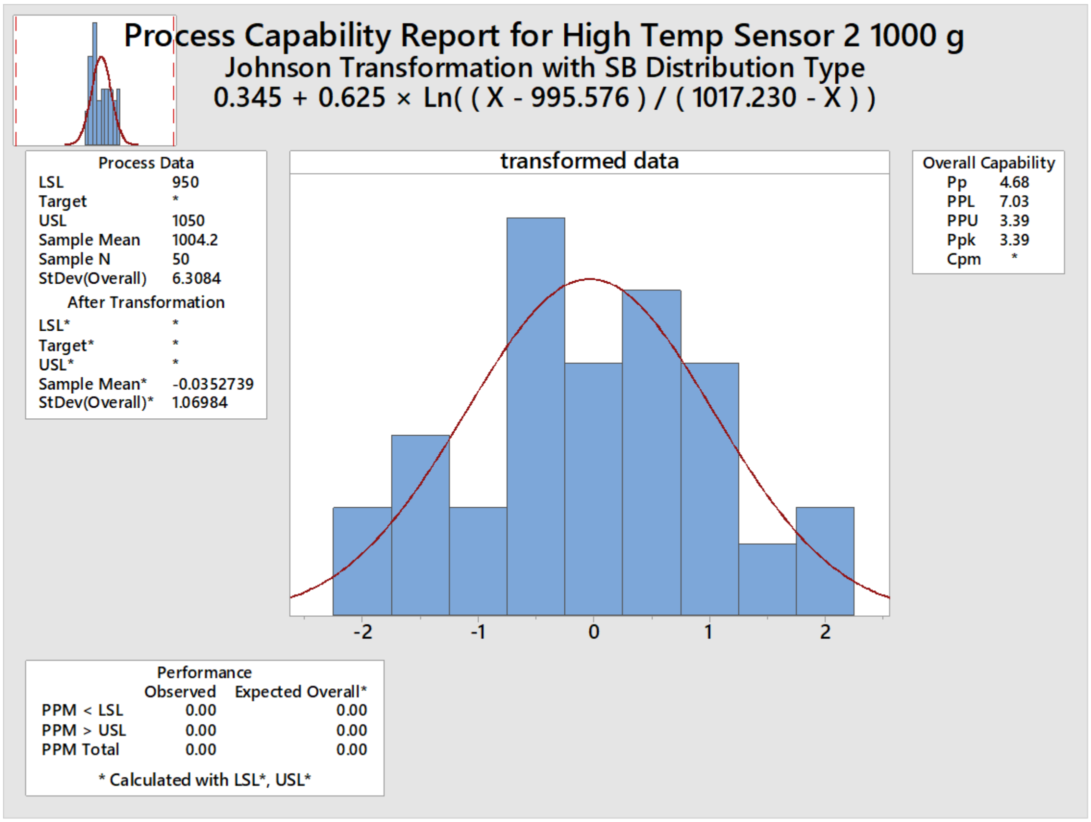
Performance

	Observed	Expected Overall*
PPM < LSL	0.00	0.00
PPM > USL	0.00	0.00
PPM Total	0.00	0.00

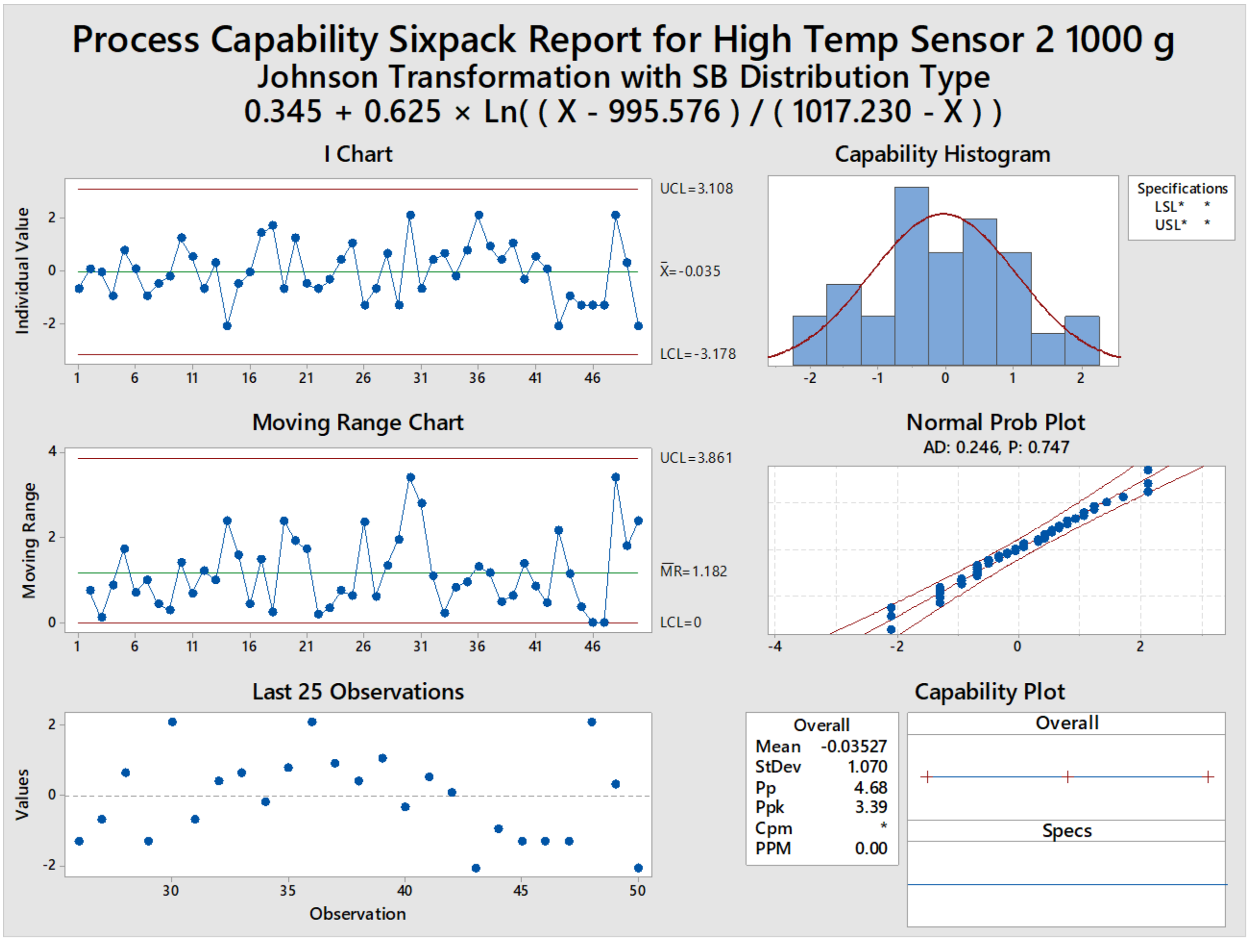
* Calculated with LSL*, USL*



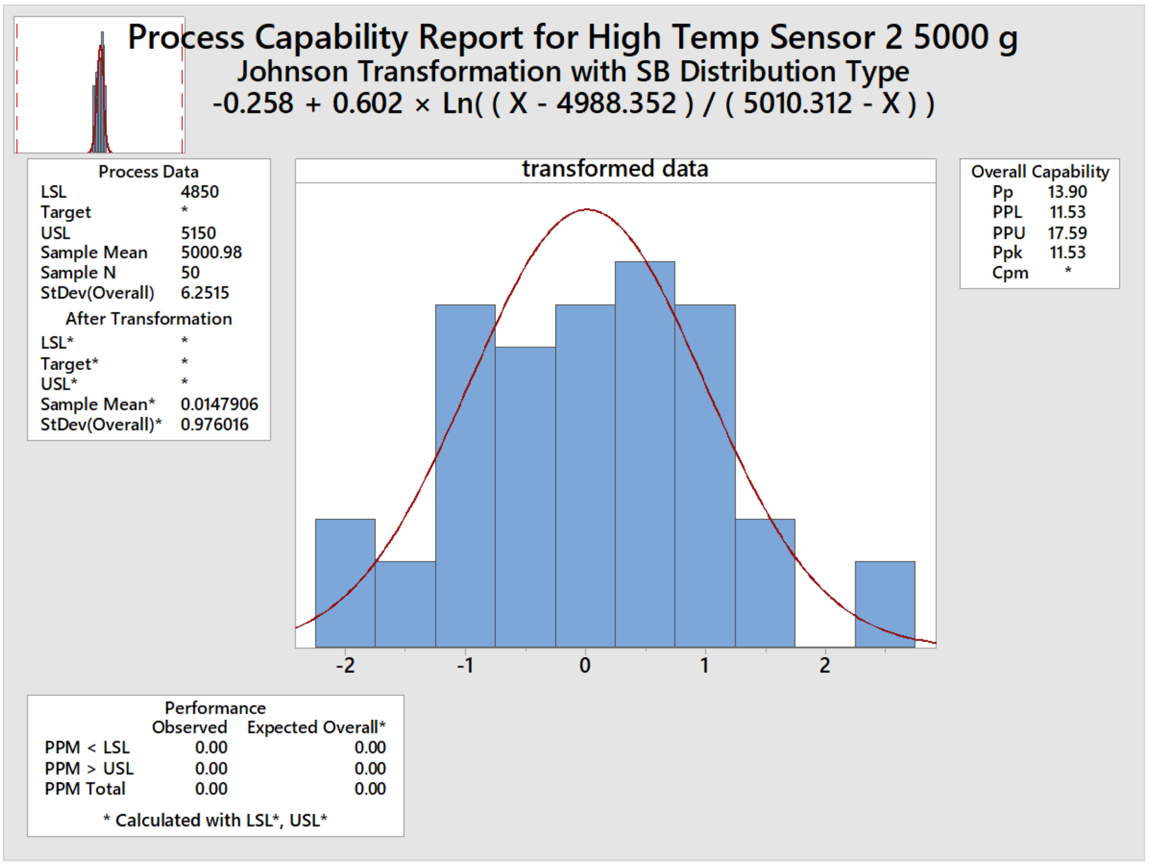
Overall	
Mean	-0.001773
StDev	0.9575
Pp	3.39
Ppk	2.62
Cpm	*
PPM	0.00



Process Capability Report for High Temp Sensor 2 1000 g
Johnson Transformation with SB Distribution Type
0.345 + 0.625 × Ln((X - 995.576) / (1017.230 - X))
Process Data
LSL 950
Target *
USL 1050
Sample Mean 1004.2
Sample N 50
StDev(Overall) 6.3084
After Transformation
LSL* *
Target* *
USL* *
Sample Mean* -0.0352739
StDev(Overall)* 1.06984
transformed data
Overall Capability
Pp 4.68
PPL 7.03
PPU 3.39
Ppk 3.39
Cpm *
Performance
Observed Expected Overall*
PPM < LSL 0.00 0.00
PPM > USL 0.00 0.00
PPM Total 0.00 0.00
* Calculated with LSL*, USL*



Process Capability Sixpack Report for High Temp Sensor 2 1000 g
Johnson Transformation with SB Distribution Type
0.345 + 0.625 × Ln((X - 995.576) / (1017.230 - X))
I Chart
UCL=3.108
X̄=-0.035
LCL=-3.178
Individual Value
Capability Histogram
Specifications
LSL* *
USL* *
Moving Range Chart
UCL=3.861
MR̄=1.182
LCL=0
Moving Range
Normal Prob Plot
AD: 0.246, P: 0.747
Last 25 Observations
Values
Observation
Capability Plot
Overall
Mean -0.03527
StDev 1.070
Pp 4.68
Ppk 3.39
Cpm *
PPM 0.00
Overall
Specs

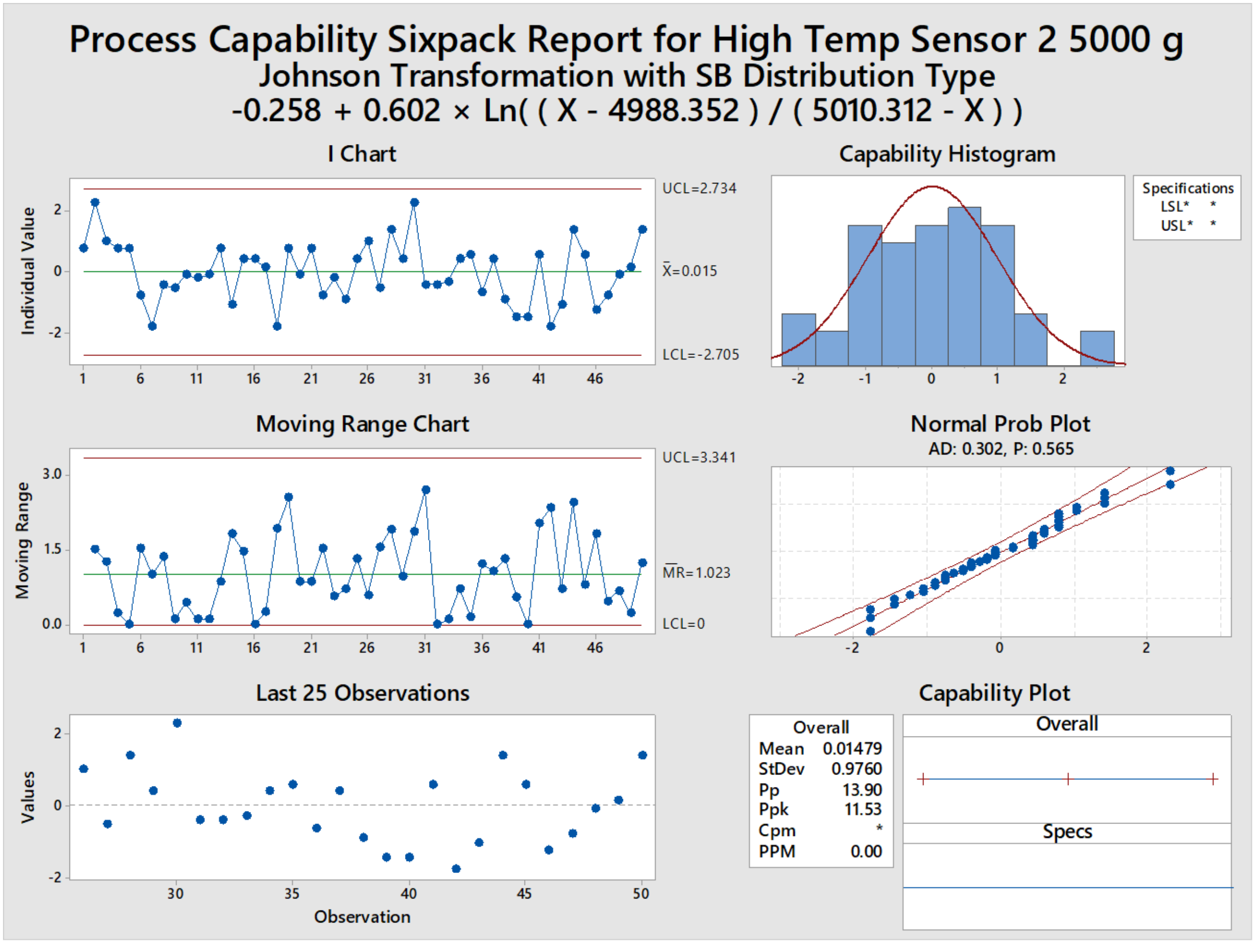

Process Capability Report for High Temp Sensor 2 5000 g
Johnson Transformation with SB Distribution Type
-0.258 + 0.602 × Ln((X - 4988.352) / (5010.312 - X))

Process Data	
LSL	4850
Target	*
USL	5150
Sample Mean	5000.98
Sample N	50
StDev(Overall)	6.2515
After Transformation	
LSL*	*
Target*	*
USL*	*
Sample Mean*	0.0147906
StDev(Overall)*	0.976016

Overall Capability	
Pp	13.90
PPL	11.53
PPU	17.59
Ppk	11.53
Cpm	*

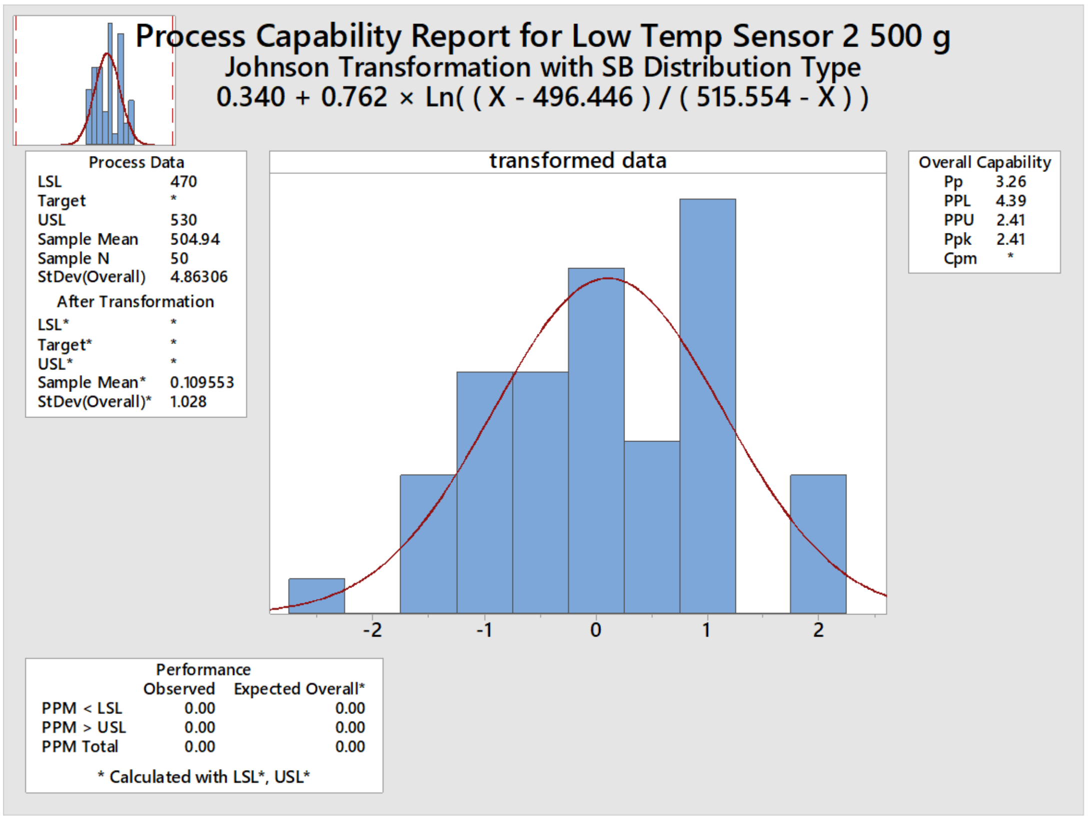
Performance	Observed	Expected Overall*
PPM < LSL	0.00	0.00
PPM > USL	0.00	0.00
PPM Total	0.00	0.00

* Calculated with LSL*, USL*


Process Capability Sixpack Report for High Temp Sensor 2 5000 g
Johnson Transformation with SB Distribution Type
-0.258 + 0.602 × Ln((X - 4988.352) / (5010.312 - X))

Specifications	
LSL*	*
USL*	*

Overall	
Mean	0.01479
StDev	0.9760
Pp	13.90
Ppk	11.53
Cpm	*
PPM	0.00



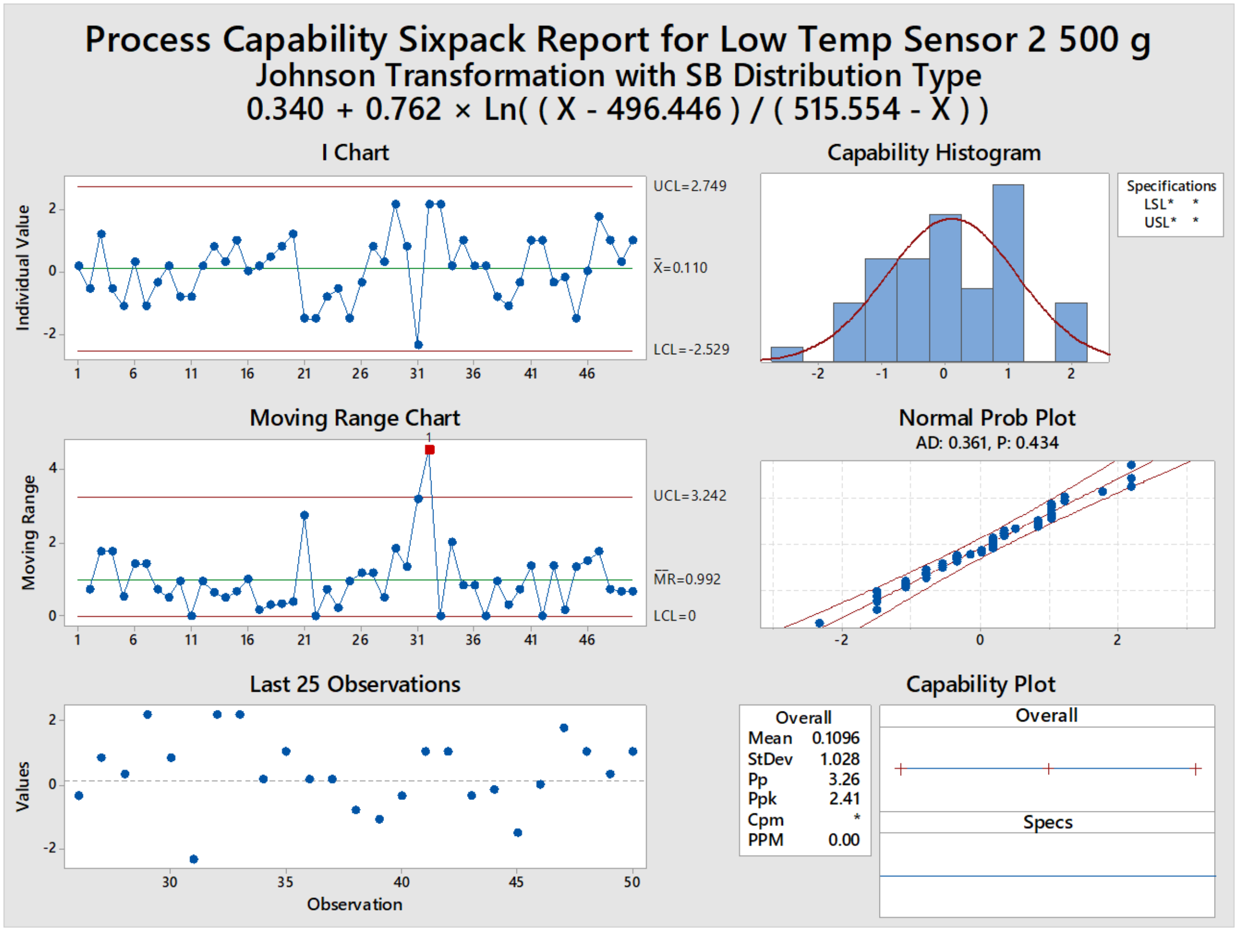
Process Capability Report for Low Temp Sensor 2 500 g
Johnson Transformation with SB Distribution Type
0.340 + 0.762 × Ln((X - 496.446) / (515.554 - X))

Process Data	
LSL	470
Target	*
USL	530
Sample Mean	504.94
Sample N	50
StDev(Overall)	4.86306
After Transformation	
LSL*	*
Target*	*
USL*	*
Sample Mean*	0.109553
StDev(Overall)*	1.028

Overall Capability	
Pp	3.26
PPL	4.39
PPU	2.41
Ppk	2.41
Cpm	*

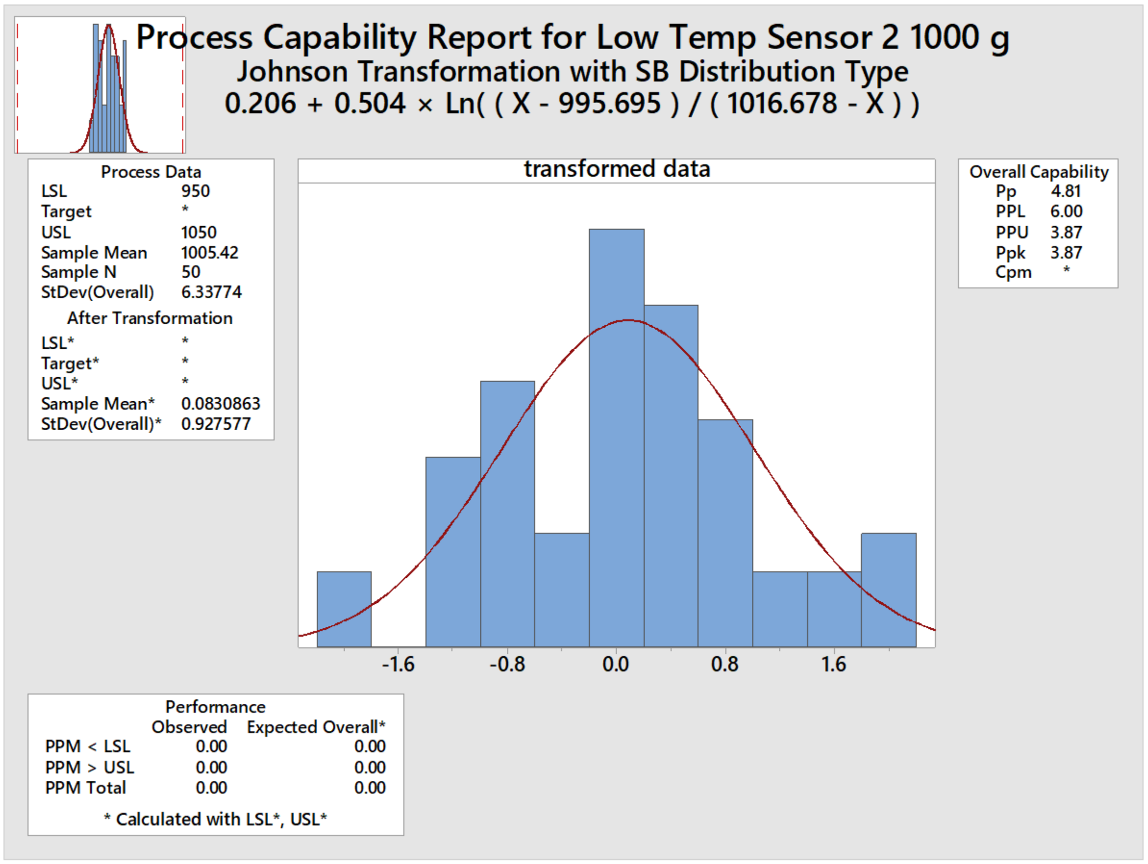
Performance	Observed	Expected Overall*
PPM < LSL	0.00	0.00
PPM > USL	0.00	0.00
PPM Total	0.00	0.00

* Calculated with LSL*, USL*



Process Capability Sixpack Report for Low Temp Sensor 2 500 g
Johnson Transformation with SB Distribution Type
0.340 + 0.762 × Ln((X - 496.446) / (515.554 - X))

Overall	
Mean	0.1096
StDev	1.028
Pp	3.26
Ppk	2.41
Cpm	*
PPM	0.00

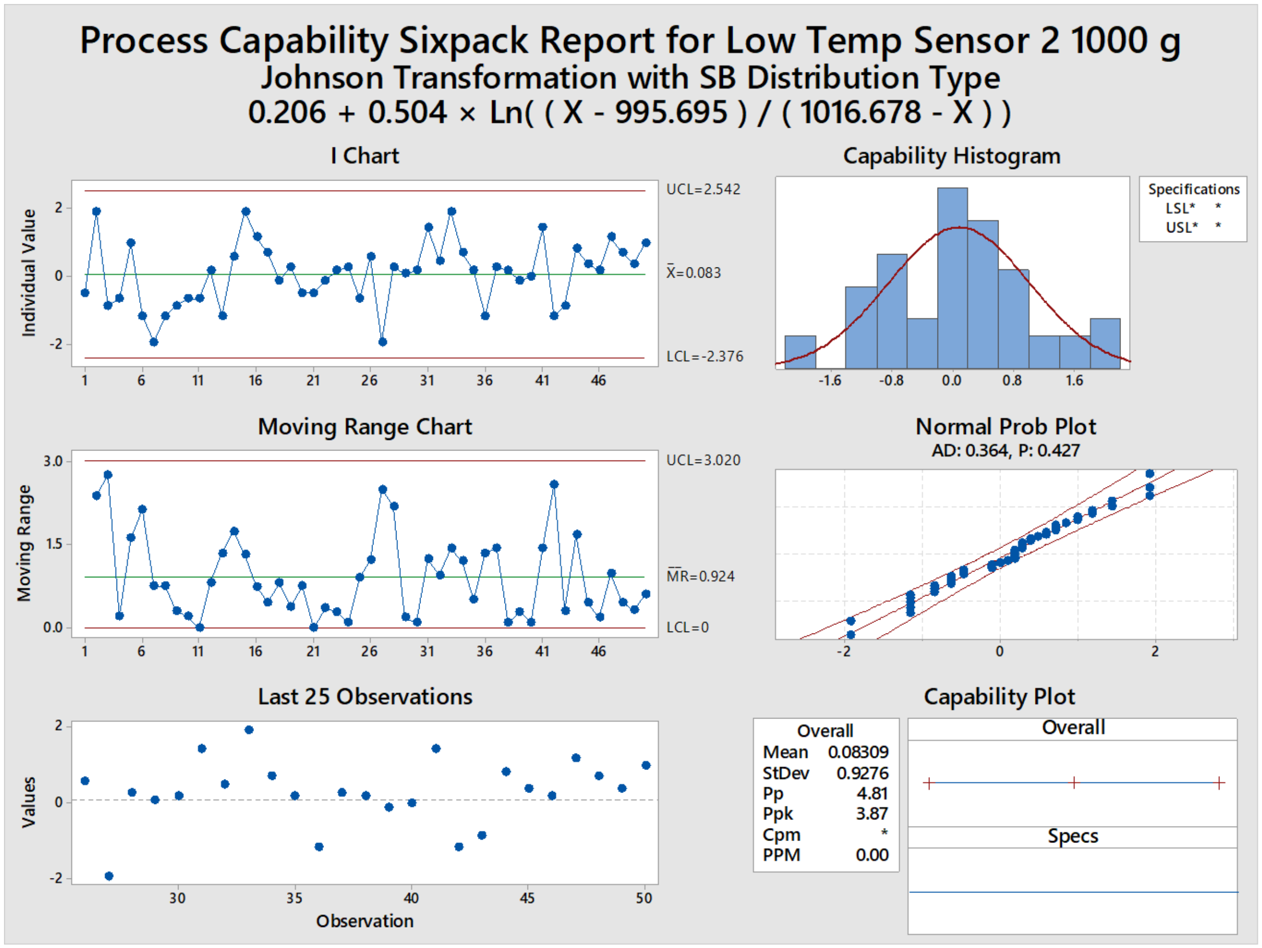


Process Data	
LSL	950
Target	*
USL	1050
Sample Mean	1005.42
Sample N	50
StDev(Overall)	6.33774
After Transformation	
LSL*	*
Target*	*
USL*	*
Sample Mean*	0.0830863
StDev(Overall)*	0.927577

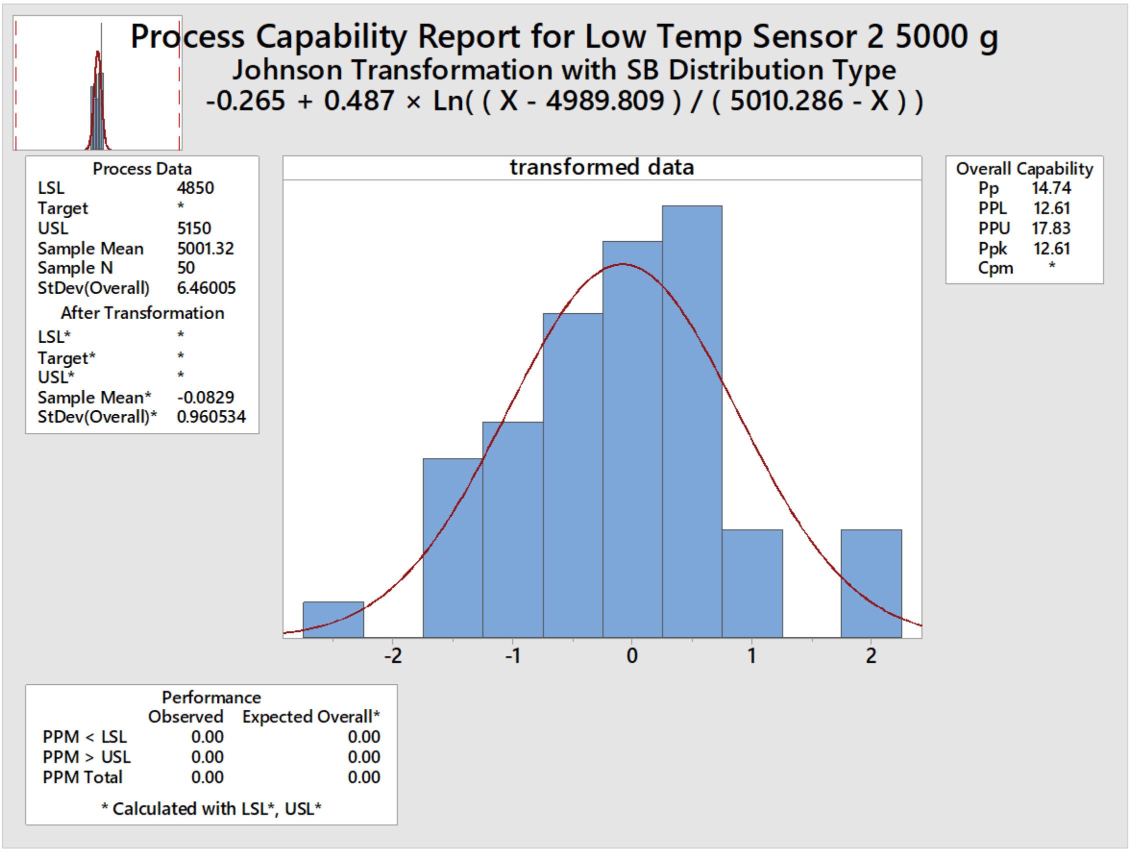
Overall Capability	
Pp	4.81
PPL	6.00
PPU	3.87
Ppk	3.87
Cpm	*

Performance	Observed	Expected Overall*
PPM < LSL	0.00	0.00
PPM > USL	0.00	0.00
PPM Total	0.00	0.00

* Calculated with LSL*, USL*



Overall	
Mean	0.08309
StDev	0.9276
Pp	4.81
Ppk	3.87
Cpm	*
PPM	0.00



Process Capability Report for Low Temp Sensor 2 5000 g

Johnson Transformation with SB Distribution Type

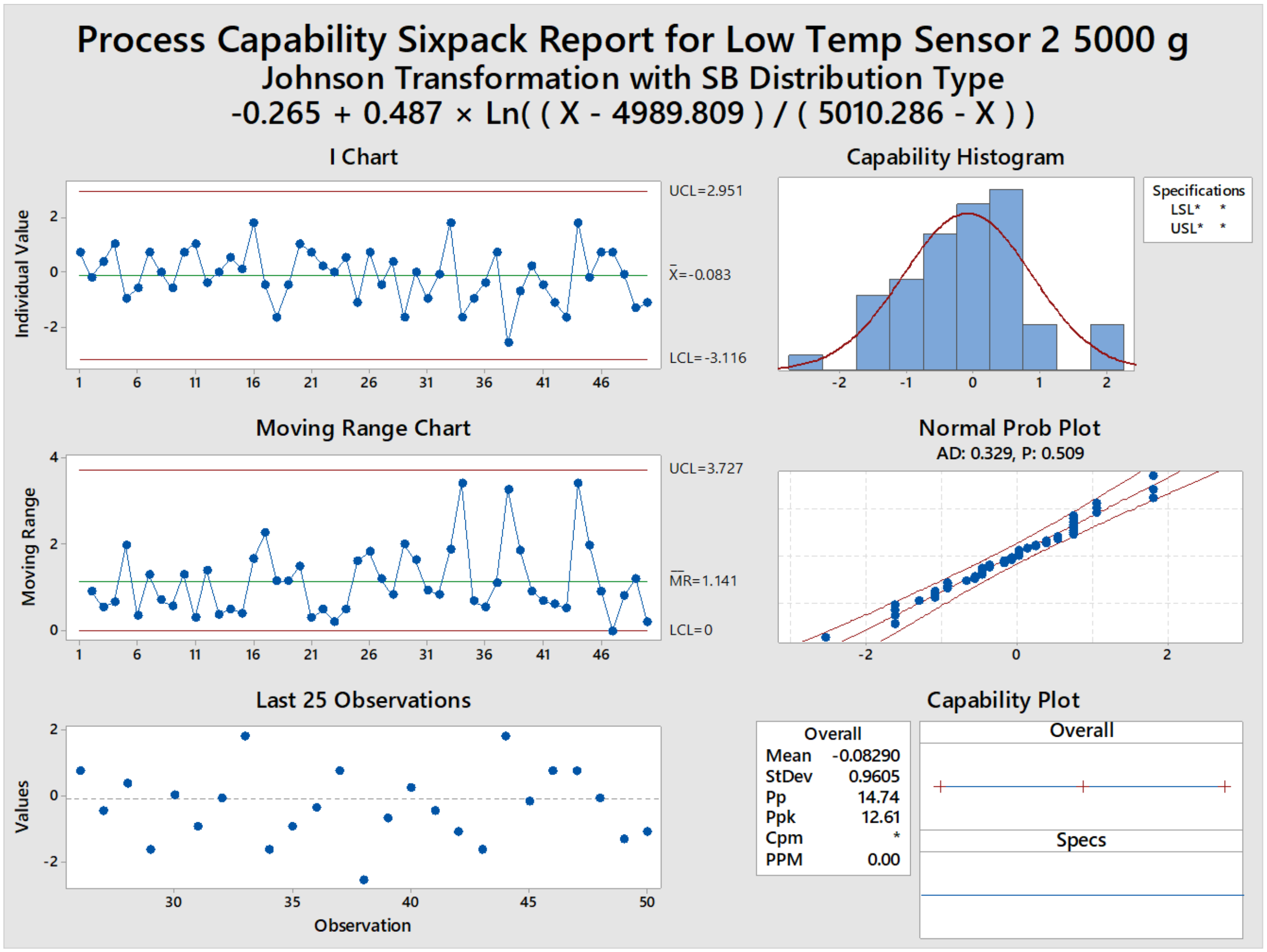
-0.265 + 0.487 × Ln((X - 4989.809) / (5010.286 - X))

Process Data	
LSL	4850
Target	*
USL	5150
Sample Mean	5001.32
Sample N	50
StDev(Overall)	6.46005
After Transformation	
LSL*	*
Target*	*
USL*	*
Sample Mean*	-0.0829
StDev(Overall)*	0.960534

Overall Capability	
Pp	14.74
PPL	12.61
PPU	17.83
Ppk	12.61
Cpm	*

Performance	Observed	Expected Overall*
PPM < LSL	0.00	0.00
PPM > USL	0.00	0.00
PPM Total	0.00	0.00

* Calculated with LSL*, USL*



Process Capability Sixpack Report for Low Temp Sensor 2 5000 g

Johnson Transformation with SB Distribution Type

-0.265 + 0.487 × Ln((X - 4989.809) / (5010.286 - X))

Specifications	
LSL*	*
USL*	*

Overall	
Mean	-0.08290
StDev	0.9605
Pp	14.74
Ppk	12.61
Cpm	*
PPM	0.00