Needs fixing

Treatment Evaluation Statistical Tool, TEST 1

08/11/2022

Table of Contents

# 1 TEST Session Summary

|  |  |
| --- | --- |
| Contact Statistician | Cheng Su |
| Objective Type | Confirmatory |
| Researcher |  |
| Studyid | TB21-02 |
| Email |  |
| Program | Exploratory |
| Department | Cardiovascular |
| Project |  |
| Full Path Home | /home/chris/bmrn-test/test\_output/Exploratory/TB21-02/1e50d506-53fa-4cec-8f30-d9bb45cd0fe8 |
| Timestamp | 2022-08-11 19:34:44 |

# 2 Study Background and Objectives

# 3 Statistical Methods

## 3.1 Overview

In a typical study supported by TEST 1, data is collected over time, and the primary interest is to evaluate a treatment in multiple doses. The study typically includes two types of subjects, wild type and non-wild type, and several controls, such as vehicle, negative control, positive control and other comparators. We will denote the Wild Type animals as the Wild Type group and the non-Wild Type groups as vehicle, negative control, positive control, other comparators, and the treatments as doses. Based on the study design and known study objective, TEST 1 defines a set of group comparisons to be evaluated statistically.

TEST 1 implements an algorithm that carries out a multiple-step analysis for statistical comparisons of groups. The statistical comparisons are conducted through mixed effect repeated measure modeling. TEST 1 is designed to automatically determine the best data transformation, check model assumptions (such as normality and homogeneous variance) and the appropriate correlation structure for the repeated measures (data across time points). In addition, TEST 1 generates linear contrasts and carries out multiple testing adjustment for the pre-defined tests.

Explanations of these key steps are described below, and the workflow of TEST 1 is described in appendix (section 6).

## 3.2 Basic Model

We define the basic model as a model that only considers the vehicle and treatment groups. These groups will be used to determine if a transformation is needed and which transformation is most appropriate. If the normality and homogeneous variance assumptions are not met in the basic model then further assistance is required and the consulting statistician will be contacted. The treatment and vehicle groups are used to remove the risk of contamination from the controls or wild type groups.

## 3.3 Mixed Effects Modeling

TEST 1 uses mixed effects models to evaluate statistical differences between multiple experimental (treatment) groups across multiple times points. The model evaluates factors including treatment, time, and the interaction between treatment and time. In addition, TEST 1 determines the appropriate correlation structure to account for the correlation between different time points. Some common correlation structures in consideration include auto-regressive (AR1) correlation structure which assumes that the correlation between time points decays at an exponential rate, compound symmetry (CS) correlation structure assumes that the correlation between time points is constant between any given two time points, and unstructured is the most flexible and has no constraints.

In TEST 1, we also allow mixed effects models can use different variance/covariance structure for each group separately as needed.

## 3.4 Normality Assumption and Transformation

A Shapiro-Wilk test is conducted to determine if the residuals from a linear model with a treatment, time, and treatment and time interaction term are normally distributed. The Shapiro Wilks test assumes that the residuals are normally distributed and looks for evidence for non-normality. If the Shapiro-Wilk test is rejected (there is enough evidence that the residuals do not follow a normal distribution), then a Box-Cox transformation is conducted to suggest an appropriate transformation. Note that the Shapiro Wilks test is rejected if the p-value is less than 0.05. Then the Shapiro-Wilk test is repeated on the residuals of the transformation to confirm that the residuals of the transformed data follow a normal distribution. If the Shapiro-Wilk test is rejected, then we recommend further discussion with a statistician as a transformation did not make data follow the normality assumption.

In experiments that have technical replication, all of the data will be used to check the normality assumption and subsequently determine the most appropriate Box-Cox transformation (treating technical replicates as independent). After this step, the technical replicates will be averaged at each time point for each subject. The resulting dataset will be used for all of the remaining analysis steps.

## 3.5 Checking for Similar Variance between Groups

### 3.5.1 Basic Model

For this application, we require that the variance for each of the groups in the basic model to be similar. To verify this assumption, first the variance is determined for each group and at each time point, and then averaged across the time points. In addition, a likelihood ratio test (LRT) is conducted between a model that estimates one common variance and a model that estimates an individual variance for each group. In this case, LRT helps determines if a more complex model fits the data significantly better than a less complex model. Thus, a significant (p-value < 0.05) LRT provides evidence that it would be beneficial to model the variances of the groups separately. The variance between groups will be considered different if one the following two conditions are met:

1. If the p-value from the LRT is less than 0.05 (statistically significant) and the fold change between any group and the common variance is greater than 2, or
2. If the p-value from the LRT is greater than 0.05 (not statistically significant) and the fold change between any group and the common variance is greater than 3.

If the variance between groups are determined to be different, then no further analysis will be conducted and the consulting statistician will be contacted. Otherwise, the application will move forward to the next modeling step.

### 3.5.2 Controls and Wild Type

Once the variance for groups within basic model is determined to be similar, then a similar procedure is followed as above. A LRT is conducted for a model that has one common variance and a model with a common variance for the groups in the basic model and a different variance for the controls and wild type. A similar set of criteria as above to determine if a group has a different variance than the basic model:

1. If the p-value from the LRT is less than 0.05 and the fold change between any group and the of the basic model is greater than 2, or
2. If the p-value from the LRT is greater than 0.05 and the fold change between any group and variance of the basic model is greater than 3.

If a group is determined to have a difference variance than the basic model then a estimation of the variance/covariance will be separate from the basic model. Otherwise, the application will move forward to fit the final mixed model with one single common variance component.

## 3.6 Selection of Correlation Sturcture

The Akaike Information Criterion (AIC) is used to select the covariance structure that is most appropriate. The AIC strikes a balance between model complexity (number of parameters) and quality of the model fit (LRT). The candidate covariance structures are AR1 (ARH1), CS (CSH), Toeplitz (TOEP), and unstructured (UN) where the AR1 and CS are simplistic structures (only estimates one additional parameter) while UN correlation structure is the most complex (requires the estimation of additional parameters where is the number of time points). AR1H and CSH covariance structure allows for each time point to have a different variance.

## 3.7 Comparison between Experimental Groups

In this type of studies, there could be nine group comparisons:

| Label | Comparison | Purpose of Comparison |
| --- | --- | --- |
| A | Wild Type vs. Vehicle | Verify Disease Model. |
| B | Positive Control vs. Vehicle | Verify the experiment. |
| C | Wild Type vs. Each Dose | Can any doses reverse the disease? |
| D | Vehicle vs. Each Dose | Are any treatments effective? |
| E | Wild Type vs. Positive Control | Does Positive Control reverse disease? |
| F | Positive Control vs. Each Dose | Which doses are similar to Positive Control? |
| G | Each Dose vs. Every Other Dose | Do doses differ from each other? |
| H | Negative Control vs. Vehicle | Rule out matrix effect. |
| I | Negative Control vs. Each Dose | If H is rejected, then show Negative Control is not as good as treatment. |

## 3.8 Multiple Hypothesis Testing

Multiple comparisons arise when a statistical analysis involves multiple simultaneous statistical tests, each of which has a potential to produce a discovery. A stated confidence level generally applies only to each test considered individually, but often it is desirable to have a confidence level for the whole family of simultaneous tests. Controlling the overall Type 1 error rate, as opposed to the Type 1 error rate of each individual test, allows us to avoid rejecting a comparison simply due to random error. For instance, if 100 comparisons were conducted each with 5% Type I error rate, we would expect 5 results to be false positives. P-value adjustment allows us to control the probability of obtaining at least one false positive in a family of tests.

For each one the comparison groups described above, linear contrasts are constructed to allow for simultaneous testing of the groups, as wells as testing the groups at the pre-selected time point and over time course average. To adjust for multiple comparison and preserve family wise Type I error rate, a simulation based method is used to determine the critical values and p-values based on sampling from the multivariate t distribution.([Liu et al. 2007](#ref-Liu)) ([Edwards and Berry 1987](#ref-berry))

For comparisons between doses, and doses and control there are multiple tests. To adjust for multiple comparison and preserve family wise Type I error rate, a simulation based method is used to determine the critical values and p-values based on sampling from the multivariate t distribution.([Liu et al. 2007](#ref-Liu)) ([Edwards and Berry 1987](#ref-berry)) There will be no p-value adjustments for the comparision between controls.

## 3.9 Change from Baseline Analysis

The user will be allowed to select whether the raw/transformed values are to be analysed or to analysis the change of each observation from a subject’s baseline measurement. The normality check and Box-Cox transformation will be conducted ignoring the baseline, then the change from baseline will be the response studied for the remainder of the analysis.

## 3.10 Software

TEST 1 was developed using the R statistical software ([R Core Team 2016](#ref-Rbase)). The nlme::gls ([Pinheiro et al. 2013](#ref-nlme)) (this syntax means) [function library]::[specific function]) was used to construct the mixed effects models, and then the emmeans::emmeans function ([Lenth 2022](#ref-emmeans)) was used to compute the group and pairwise comparison summary statistics, as well as, adjust the p-values for multiple comparisons. R Shiny ([RStudio, Inc 2013](#ref-shiny)) was used to generate the web tool and this reported was generated using RMarkdown ([Allaire et al. 2020](#ref-rmarkdown)). The interactive plots within the web application were generated using the plotly::ggplotly function which converts a plot from ggplot2 ([Wickham 2016](#ref-ggplot2)) to plotly ([Sievert 2020](#ref-plotly)). The prism-style plots were generated using the ggprism([Dawson 2021](#ref-ggprism)) package which is an extension to the ggplot2 package.

# 4 Results

## 4.1 Assumptions and Correlation Structure Results

A Square Root was applied to the data. of the data was determined by the algorithm.

An AR1 was determined by the algorithm.

Wild Type had a different variance from the basic model.

## 4.2 Graphical Summary of Residuals Before and After Transformation

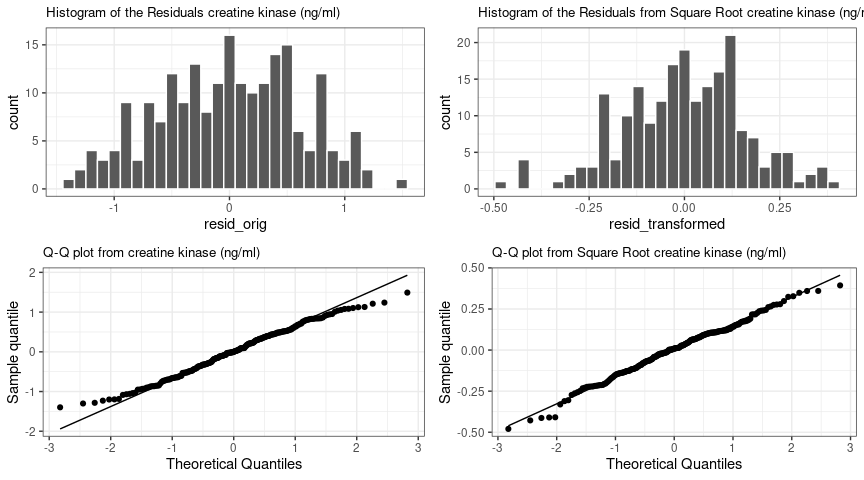


Figure 4.1: Histogram (top) and Q-Q plots (bottom) to assess the normality of the residuals

## 4.3 Plots

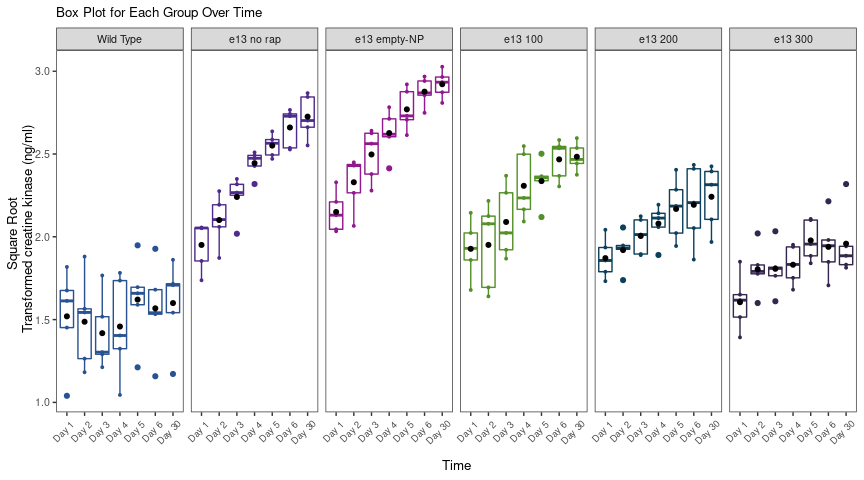


Figure 4.2: Box plot for each group and time.

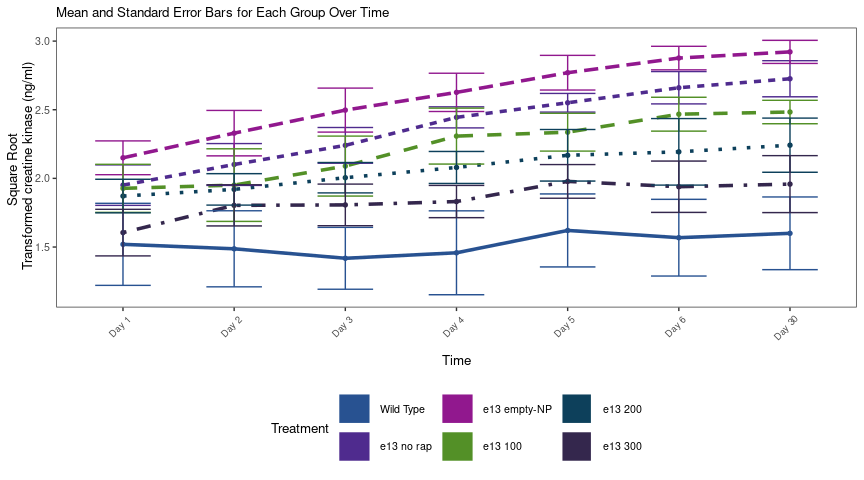


Figure 4.3: Group level trajectory across time where the line corresponds to the group average of Transformed Scale. The vertical bars are standard error bars for each group at each time point.

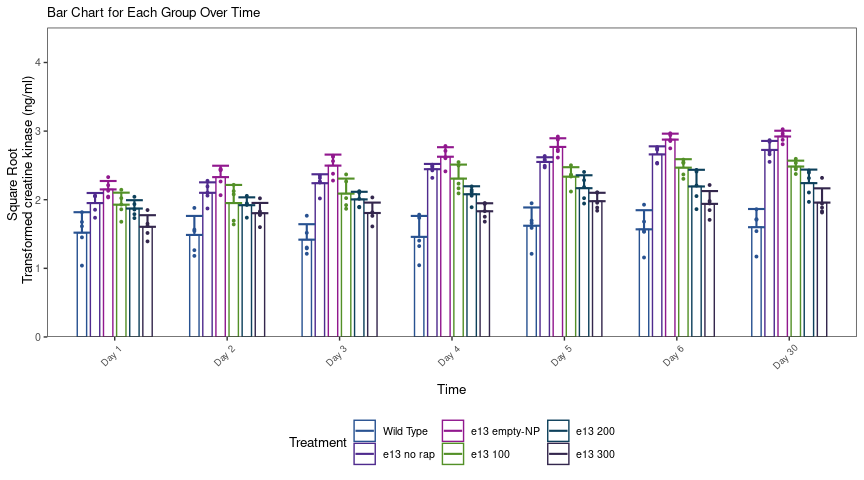


Figure 4.4: Bar plot for each group and time.

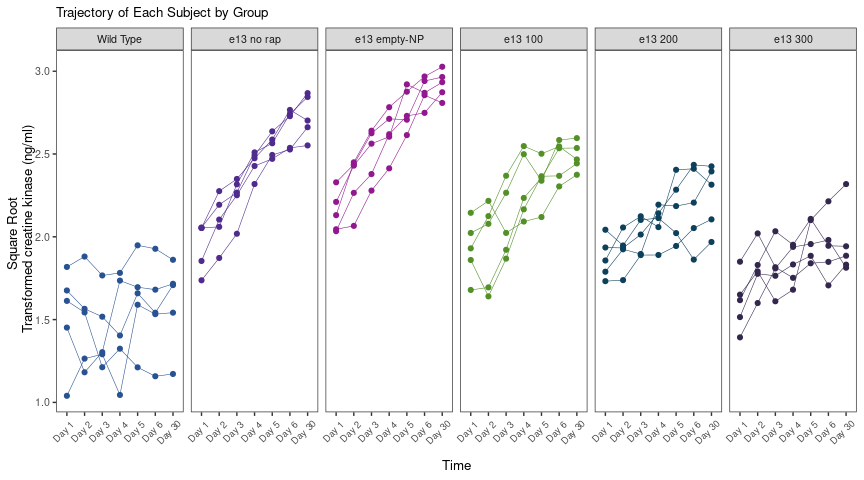


Figure 4.5: Subject level trajectory across time where the line corresponds to the average of Transformed Scale response in the presence of techinical replicates.

## Called from: prism\_plot(data = transformed\_data, tables = data$tables$tables,   
## trt\_sel = levels(transformed\_data$Treatment), time\_sel = time,   
## endpoint = data$plot$endpoint, format = "word", cfb = data$input\_data$changeFromBaseline,   
## power = data$tables$power, num\_groups = length(levels(transformed\_data$Treatment)),   
## type = "bar", inputs = prismInputs)  
## debug at /home/chris/bmrn-test/R/prism\_plot.R#108: data <- data %>% filter(Time == time\_sel, Treatment %in% trt\_sel) %>%   
## group\_by(Treatment) %>% mutate(outlier = is.outlier(Response\_Transformed),   
## Treatment = factor(Treatment, levels = correct\_level\_order)) %>%   
## ungroup()  
## debug at /home/chris/bmrn-test/R/prism\_plot.R#118: data\_max <- data %>% group\_by(Treatment) %>% summarize(max = max(Response\_Transformed),   
## Mean\_Response = mean(Response\_Transformed), sd\_Response = sd(Response\_Transformed)) %>%   
## rename(Response\_Transformed = Mean\_Response) %>% mutate(error = if\_else(Response\_Transformed <   
## 0, Response\_Transformed - sd\_Response, Response\_Transformed +   
## sd\_Response), ymin = if\_else(Response\_Transformed < 0, error,   
## Response\_Transformed), ymax = if\_else(Response\_Transformed <   
## 0, Response\_Transformed, error))  
## debug at /home/chris/bmrn-test/R/prism\_plot.R#132: if (type == "box") {  
## full\_prism <- ggplot(data %>% rename(group1 = Treatment),   
## aes(x = group1, y = Response\_Transformed, color = group1)) +   
## stat\_boxplot(geom = "errorbar", width = 3 \* length(unique(data$Treatment))/num\_groups^2,   
## lwd = 1) + geom\_boxplot(aes(fill = group1), outlier.color = NA,   
## lwd = ifelse(format == "word", 1, 2), fatten = 1, width = 3 \*   
## length(unique(data$Treatment))/num\_groups^2, alpha = 0.5) +   
## geom\_jitter(size = 3, position = position\_dodge(width = 0.2)) +   
## stat\_summary(fun = "mean", color = "black", show.legend = FALSE) +   
## scale\_y\_continuous(limits = c(NA, 3 \* max(data$Response\_Transformed))) +   
## scale\_color\_prism("floral") + scale\_fill\_prism("floral") +   
## guides(y = "prism\_offset\_minor") + theme\_prism(base\_size = ifelse(format ==   
## "word", 16, inputs$fontSize)) + theme(legend.position = "none") +   
## ylab(ylab) + xlab("Treatment")  
## bottom <- full\_prism + scale\_y\_continuous(limits = c(1.1 \*   
## min(0, min(data\_max$ymin)), 1.1 \* max(data\_max$ymax)),   
## expand = expansion(mult = c(0, 0))) + theme(plot.margin = margin(t = -10,   
## r = 0, b = 0, l = 0))  
## } else {  
## full\_prism <- ggplot(data\_max %>% rename(group1 = Treatment),   
## aes(x = group1, y = Response\_Transformed, color = group1)) +   
## geom\_bar(aes(fill = group1), lwd = ifelse(format == "word",   
## 1, 2), stat = "identity", width = 3 \* length(unique(data$Treatment))/num\_groups^2,   
## position = position\_dodge(width = 0.7), alpha = 0.5) +   
## geom\_jitter(data = data %>% rename(group1 = Treatment),   
## aes(y = Response\_Transformed), size = 3, position = position\_dodge(width = 0.2)) +   
## stat\_summary(data = data %>% rename(group1 = Treatment),   
## fun = "mean", color = "black", show.legend = FALSE) +   
## geom\_errorbar(aes(ymin = ymin, ymax = ymax), position = position\_dodge(width = 0.7),   
## width = 3 \* length(unique(data$Treatment))/num\_groups^2,   
## size = ifelse(format == "word", 1, 2)) + guides(y = "prism\_offset\_minor") +   
## scale\_y\_continuous(limits = c(NA, 1.1 \* max(p\_vals$new\_y.position))) +   
## scale\_color\_prism("floral") + scale\_fill\_prism("floral") +   
## theme\_prism(base\_size = ifelse(format == "word", 16,   
## inputs$fontSize)) + theme(legend.position = "none") +   
## ylab(ylab) + xlab("Treatment")  
## bottom <- full\_prism + theme(plot.margin = margin(t = -10,   
## r = 0, b = 0, l = 0)) + scale\_y\_continuous(limits = c(1.1 \*   
## min(0, min(data\_max$ymin)), 1.1 \* max(data\_max$ymax)),   
## expand = expansion(mult = c(0, 0)))  
## }  
## debug at /home/chris/bmrn-test/R/prism\_plot.R#185: full\_prism <- ggplot(data\_max %>% rename(group1 = Treatment),   
## aes(x = group1, y = Response\_Transformed, color = group1)) +   
## geom\_bar(aes(fill = group1), lwd = ifelse(format == "word",   
## 1, 2), stat = "identity", width = 3 \* length(unique(data$Treatment))/num\_groups^2,   
## position = position\_dodge(width = 0.7), alpha = 0.5) +   
## geom\_jitter(data = data %>% rename(group1 = Treatment), aes(y = Response\_Transformed),   
## size = 3, position = position\_dodge(width = 0.2)) + stat\_summary(data = data %>%   
## rename(group1 = Treatment), fun = "mean", color = "black",   
## show.legend = FALSE) + geom\_errorbar(aes(ymin = ymin, ymax = ymax),   
## position = position\_dodge(width = 0.7), width = 3 \* length(unique(data$Treatment))/num\_groups^2,   
## size = ifelse(format == "word", 1, 2)) + guides(y = "prism\_offset\_minor") +   
## scale\_y\_continuous(limits = c(NA, 1.1 \* max(p\_vals$new\_y.position))) +   
## scale\_color\_prism("floral") + scale\_fill\_prism("floral") +   
## theme\_prism(base\_size = ifelse(format == "word", 16, inputs$fontSize)) +   
## theme(legend.position = "none") + ylab(ylab) + xlab("Treatment")  
## debug at /home/chris/bmrn-test/R/prism\_plot.R#223: bottom <- full\_prism + theme(plot.margin = margin(t = -10, r = 0,   
## b = 0, l = 0)) + scale\_y\_continuous(limits = c(1.1 \* min(0,   
## min(data\_max$ymin)), 1.1 \* max(data\_max$ymax)), expand = expansion(mult = c(0,   
## 0)))  
## debug at /home/chris/bmrn-test/R/prism\_plot.R#235: if (format == "word") {  
## bottom <- bottom + theme(axis.text = element\_text(size = 8),   
## axis.title = element\_text(size = 10))  
## }  
## debug at /home/chris/bmrn-test/R/prism\_plot.R#236: bottom <- bottom + theme(axis.text = element\_text(size = 8),   
## axis.title = element\_text(size = 10))  
## debug at /home/chris/bmrn-test/R/prism\_plot.R#241: if (nrow(p\_vals) > 0 & ((as.logical(cfb) == TRUE & y\_axis ==   
## "change\_from\_baseline") | (as.logical(cfb) != TRUE & y\_axis !=   
## "change\_from\_baseline"))) {  
## full\_prism <- full\_prism + add\_pvalue(data = p\_vals, y.position = "new\_y.position",   
## label = "{sig}", tip.length = 0.02, label.size = ifelse(format ==   
## "word", 4, 8), color = "black", size = 2, step.increase = 0.02)  
## top <- full\_prism + scale\_y\_continuous(limits = c(0.9 \* min(p\_vals$new\_y.position),   
## NA), expand = expansion(mult = c(0, 0.1)))  
## if (format == "word") {  
## top = top + theme(line = element\_blank(), axis.title = element\_blank(),   
## axis.text = element\_blank()) + theme(plot.margin = margin(t = 0,   
## r = 0, b = 0, l = 0))  
## }  
## else {  
## top <- full\_prism + scale\_y\_continuous(limits = c(0.9 \*   
## min(p\_vals$new\_y.position), NA), expand = expansion(mult = c(0,   
## 0.1))) + theme(line = element\_blank(), axis.title = element\_blank(),   
## axis.text = element\_blank()) + theme(plot.margin = margin(t = 0,   
## r = 0, b = 0, l = 0))  
## }  
## if (type == "box") {  
## top <- top + ggtitle(paste("Box plot for Treatment Groups at",   
## time\_sel))  
## combined <- grid.arrange(grobs = list(top, bottom), layout\_matrix = if (format ==   
## "word") {  
## rbind(1, 2, 2)  
## }  
## else {  
## rbind(1, 2, 2)  
## })  
## }  
## else {  
## top <- top + ggtitle(paste("Bar Chart for Treatment Groups at",   
## time\_sel))  
## combined <- grid.arrange(grobs = list(top, bottom), layout\_matrix = if (format ==   
## "word") {  
## rbind(1, 2, 2)  
## }  
## else {  
## rbind(1, 2, 2)  
## })  
## }  
## return(combined)  
## } else {  
## if (type == "box") {  
## bottom <- bottom + ggtitle(paste("Box Plot for Treatment Groups at",   
## time\_sel)) + theme(plot.margin = margin(t = 0, r = 0,   
## b = 0, l = 0))  
## }  
## else {  
## bottom <- bottom + ggtitle(paste("Bar Chart for Treatment Groups at",   
## time\_sel)) + theme(plot.margin = margin(t = 0, r = 0,   
## b = 0, l = 0))  
## }  
## return(bottom)  
## }  
## debug at /home/chris/bmrn-test/R/prism\_plot.R#243: full\_prism <- full\_prism + add\_pvalue(data = p\_vals, y.position = "new\_y.position",   
## label = "{sig}", tip.length = 0.02, label.size = ifelse(format ==   
## "word", 4, 8), color = "black", size = 2, step.increase = 0.02)  
## debug at /home/chris/bmrn-test/R/prism\_plot.R#254: top <- full\_prism + scale\_y\_continuous(limits = c(0.9 \* min(p\_vals$new\_y.position),   
## NA), expand = expansion(mult = c(0, 0.1)))  
## debug at /home/chris/bmrn-test/R/prism\_plot.R#260: if (format == "word") {  
## top = top + theme(line = element\_blank(), axis.title = element\_blank(),   
## axis.text = element\_blank()) + theme(plot.margin = margin(t = 0,   
## r = 0, b = 0, l = 0))  
## } else {  
## top <- full\_prism + scale\_y\_continuous(limits = c(0.9 \* min(p\_vals$new\_y.position),   
## NA), expand = expansion(mult = c(0, 0.1))) + theme(line = element\_blank(),   
## axis.title = element\_blank(), axis.text = element\_blank()) +   
## theme(plot.margin = margin(t = 0, r = 0, b = 0, l = 0))  
## }  
## debug at /home/chris/bmrn-test/R/prism\_plot.R#261: top = top + theme(line = element\_blank(), axis.title = element\_blank(),   
## axis.text = element\_blank()) + theme(plot.margin = margin(t = 0,   
## r = 0, b = 0, l = 0))  
## debug at /home/chris/bmrn-test/R/prism\_plot.R#291: if (type == "box") {  
## top <- top + ggtitle(paste("Box plot for Treatment Groups at",   
## time\_sel))  
## combined <- grid.arrange(grobs = list(top, bottom), layout\_matrix = if (format ==   
## "word") {  
## rbind(1, 2, 2)  
## }  
## else {  
## rbind(1, 2, 2)  
## })  
## } else {  
## top <- top + ggtitle(paste("Bar Chart for Treatment Groups at",   
## time\_sel))  
## combined <- grid.arrange(grobs = list(top, bottom), layout\_matrix = if (format ==   
## "word") {  
## rbind(1, 2, 2)  
## }  
## else {  
## rbind(1, 2, 2)  
## })  
## }  
## debug at /home/chris/bmrn-test/R/prism\_plot.R#296: top <- top + ggtitle(paste("Bar Chart for Treatment Groups at",   
## time\_sel))  
## debug at /home/chris/bmrn-test/R/prism\_plot.R#297: combined <- grid.arrange(grobs = list(top, bottom), layout\_matrix = if (format ==   
## "word") {  
## rbind(1, 2, 2)  
## } else {  
## rbind(1, 2, 2)  
## })

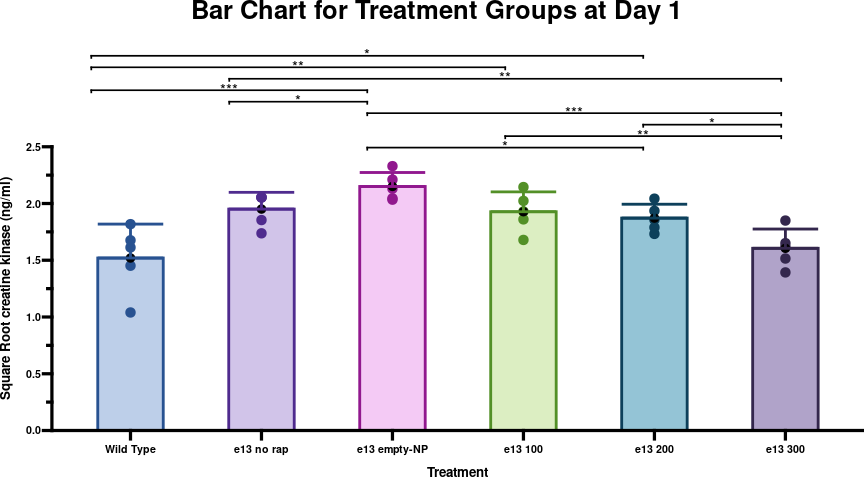


Figure 4.6: Bar chart of Transformed Scale response values at Day 1. The stars above the horizontal bars correspond to the magnitude of the p-value comparing two groups where “\*”, “\*\*”, and “\*\*\*” represent 0.01 <= p-value < 0.05, 0.001 <= p-value < 0.01, and p-value < 0.001, respectively. A p-value summary will only be shown form p-value less than 0.05.

## debug at /home/chris/bmrn-test/R/prism\_plot.R#297: rbind(1, 2, 2)  
## debug at /home/chris/bmrn-test/R/prism\_plot.R#300: return(combined)

## Called from: prism\_plot(data = transformed\_data, tables = data$tables$tables,   
## trt\_sel = levels(transformed\_data$Treatment), time\_sel = time,   
## endpoint = data$plot$endpoint, format = "word", cfb = data$input\_data$changeFromBaseline,   
## power = data$tables$power, num\_groups = length(levels(transformed\_data$Treatment)),   
## type = "box", inputs = prismInputs)  
## debug at /home/chris/bmrn-test/R/prism\_plot.R#108: data <- data %>% filter(Time == time\_sel, Treatment %in% trt\_sel) %>%   
## group\_by(Treatment) %>% mutate(outlier = is.outlier(Response\_Transformed),   
## Treatment = factor(Treatment, levels = correct\_level\_order)) %>%   
## ungroup()  
## debug at /home/chris/bmrn-test/R/prism\_plot.R#118: data\_max <- data %>% group\_by(Treatment) %>% summarize(max = max(Response\_Transformed),   
## Mean\_Response = mean(Response\_Transformed), sd\_Response = sd(Response\_Transformed)) %>%   
## rename(Response\_Transformed = Mean\_Response) %>% mutate(error = if\_else(Response\_Transformed <   
## 0, Response\_Transformed - sd\_Response, Response\_Transformed +   
## sd\_Response), ymin = if\_else(Response\_Transformed < 0, error,   
## Response\_Transformed), ymax = if\_else(Response\_Transformed <   
## 0, Response\_Transformed, error))  
## debug at /home/chris/bmrn-test/R/prism\_plot.R#132: if (type == "box") {  
## full\_prism <- ggplot(data %>% rename(group1 = Treatment),   
## aes(x = group1, y = Response\_Transformed, color = group1)) +   
## stat\_boxplot(geom = "errorbar", width = 3 \* length(unique(data$Treatment))/num\_groups^2,   
## lwd = 1) + geom\_boxplot(aes(fill = group1), outlier.color = NA,   
## lwd = ifelse(format == "word", 1, 2), fatten = 1, width = 3 \*   
## length(unique(data$Treatment))/num\_groups^2, alpha = 0.5) +   
## geom\_jitter(size = 3, position = position\_dodge(width = 0.2)) +   
## stat\_summary(fun = "mean", color = "black", show.legend = FALSE) +   
## scale\_y\_continuous(limits = c(NA, 3 \* max(data$Response\_Transformed))) +   
## scale\_color\_prism("floral") + scale\_fill\_prism("floral") +   
## guides(y = "prism\_offset\_minor") + theme\_prism(base\_size = ifelse(format ==   
## "word", 16, inputs$fontSize)) + theme(legend.position = "none") +   
## ylab(ylab) + xlab("Treatment")  
## bottom <- full\_prism + scale\_y\_continuous(limits = c(1.1 \*   
## min(0, min(data\_max$ymin)), 1.1 \* max(data\_max$ymax)),   
## expand = expansion(mult = c(0, 0))) + theme(plot.margin = margin(t = -10,   
## r = 0, b = 0, l = 0))  
## } else {  
## full\_prism <- ggplot(data\_max %>% rename(group1 = Treatment),   
## aes(x = group1, y = Response\_Transformed, color = group1)) +   
## geom\_bar(aes(fill = group1), lwd = ifelse(format == "word",   
## 1, 2), stat = "identity", width = 3 \* length(unique(data$Treatment))/num\_groups^2,   
## position = position\_dodge(width = 0.7), alpha = 0.5) +   
## geom\_jitter(data = data %>% rename(group1 = Treatment),   
## aes(y = Response\_Transformed), size = 3, position = position\_dodge(width = 0.2)) +   
## stat\_summary(data = data %>% rename(group1 = Treatment),   
## fun = "mean", color = "black", show.legend = FALSE) +   
## geom\_errorbar(aes(ymin = ymin, ymax = ymax), position = position\_dodge(width = 0.7),   
## width = 3 \* length(unique(data$Treatment))/num\_groups^2,   
## size = ifelse(format == "word", 1, 2)) + guides(y = "prism\_offset\_minor") +   
## scale\_y\_continuous(limits = c(NA, 1.1 \* max(p\_vals$new\_y.position))) +   
## scale\_color\_prism("floral") + scale\_fill\_prism("floral") +   
## theme\_prism(base\_size = ifelse(format == "word", 16,   
## inputs$fontSize)) + theme(legend.position = "none") +   
## ylab(ylab) + xlab("Treatment")  
## bottom <- full\_prism + theme(plot.margin = margin(t = -10,   
## r = 0, b = 0, l = 0)) + scale\_y\_continuous(limits = c(1.1 \*   
## min(0, min(data\_max$ymin)), 1.1 \* max(data\_max$ymax)),   
## expand = expansion(mult = c(0, 0)))  
## }  
## debug at /home/chris/bmrn-test/R/prism\_plot.R#133: full\_prism <- ggplot(data %>% rename(group1 = Treatment), aes(x = group1,   
## y = Response\_Transformed, color = group1)) + stat\_boxplot(geom = "errorbar",   
## width = 3 \* length(unique(data$Treatment))/num\_groups^2,   
## lwd = 1) + geom\_boxplot(aes(fill = group1), outlier.color = NA,   
## lwd = ifelse(format == "word", 1, 2), fatten = 1, width = 3 \*   
## length(unique(data$Treatment))/num\_groups^2, alpha = 0.5) +   
## geom\_jitter(size = 3, position = position\_dodge(width = 0.2)) +   
## stat\_summary(fun = "mean", color = "black", show.legend = FALSE) +   
## scale\_y\_continuous(limits = c(NA, 3 \* max(data$Response\_Transformed))) +   
## scale\_color\_prism("floral") + scale\_fill\_prism("floral") +   
## guides(y = "prism\_offset\_minor") + theme\_prism(base\_size = ifelse(format ==   
## "word", 16, inputs$fontSize)) + theme(legend.position = "none") +   
## ylab(ylab) + xlab("Treatment")  
## debug at /home/chris/bmrn-test/R/prism\_plot.R#172: bottom <- full\_prism + scale\_y\_continuous(limits = c(1.1 \* min(0,   
## min(data\_max$ymin)), 1.1 \* max(data\_max$ymax)), expand = expansion(mult = c(0,   
## 0))) + theme(plot.margin = margin(t = -10, r = 0, b = 0,   
## l = 0))  
## debug at /home/chris/bmrn-test/R/prism\_plot.R#235: if (format == "word") {  
## bottom <- bottom + theme(axis.text = element\_text(size = 8),   
## axis.title = element\_text(size = 10))  
## }  
## debug at /home/chris/bmrn-test/R/prism\_plot.R#236: bottom <- bottom + theme(axis.text = element\_text(size = 8),   
## axis.title = element\_text(size = 10))  
## debug at /home/chris/bmrn-test/R/prism\_plot.R#241: if (nrow(p\_vals) > 0 & ((as.logical(cfb) == TRUE & y\_axis ==   
## "change\_from\_baseline") | (as.logical(cfb) != TRUE & y\_axis !=   
## "change\_from\_baseline"))) {  
## full\_prism <- full\_prism + add\_pvalue(data = p\_vals, y.position = "new\_y.position",   
## label = "{sig}", tip.length = 0.02, label.size = ifelse(format ==   
## "word", 4, 8), color = "black", size = 2, step.increase = 0.02)  
## top <- full\_prism + scale\_y\_continuous(limits = c(0.9 \* min(p\_vals$new\_y.position),   
## NA), expand = expansion(mult = c(0, 0.1)))  
## if (format == "word") {  
## top = top + theme(line = element\_blank(), axis.title = element\_blank(),   
## axis.text = element\_blank()) + theme(plot.margin = margin(t = 0,   
## r = 0, b = 0, l = 0))  
## }  
## else {  
## top <- full\_prism + scale\_y\_continuous(limits = c(0.9 \*   
## min(p\_vals$new\_y.position), NA), expand = expansion(mult = c(0,   
## 0.1))) + theme(line = element\_blank(), axis.title = element\_blank(),   
## axis.text = element\_blank()) + theme(plot.margin = margin(t = 0,   
## r = 0, b = 0, l = 0))  
## }  
## if (type == "box") {  
## top <- top + ggtitle(paste("Box plot for Treatment Groups at",   
## time\_sel))  
## combined <- grid.arrange(grobs = list(top, bottom), layout\_matrix = if (format ==   
## "word") {  
## rbind(1, 2, 2)  
## }  
## else {  
## rbind(1, 2, 2)  
## })  
## }  
## else {  
## top <- top + ggtitle(paste("Bar Chart for Treatment Groups at",   
## time\_sel))  
## combined <- grid.arrange(grobs = list(top, bottom), layout\_matrix = if (format ==   
## "word") {  
## rbind(1, 2, 2)  
## }  
## else {  
## rbind(1, 2, 2)  
## })  
## }  
## return(combined)  
## } else {  
## if (type == "box") {  
## bottom <- bottom + ggtitle(paste("Box Plot for Treatment Groups at",   
## time\_sel)) + theme(plot.margin = margin(t = 0, r = 0,   
## b = 0, l = 0))  
## }  
## else {  
## bottom <- bottom + ggtitle(paste("Bar Chart for Treatment Groups at",   
## time\_sel)) + theme(plot.margin = margin(t = 0, r = 0,   
## b = 0, l = 0))  
## }  
## return(bottom)  
## }  
## debug at /home/chris/bmrn-test/R/prism\_plot.R#243: full\_prism <- full\_prism + add\_pvalue(data = p\_vals, y.position = "new\_y.position",   
## label = "{sig}", tip.length = 0.02, label.size = ifelse(format ==   
## "word", 4, 8), color = "black", size = 2, step.increase = 0.02)  
## debug at /home/chris/bmrn-test/R/prism\_plot.R#254: top <- full\_prism + scale\_y\_continuous(limits = c(0.9 \* min(p\_vals$new\_y.position),   
## NA), expand = expansion(mult = c(0, 0.1)))  
## debug at /home/chris/bmrn-test/R/prism\_plot.R#260: if (format == "word") {  
## top = top + theme(line = element\_blank(), axis.title = element\_blank(),   
## axis.text = element\_blank()) + theme(plot.margin = margin(t = 0,   
## r = 0, b = 0, l = 0))  
## } else {  
## top <- full\_prism + scale\_y\_continuous(limits = c(0.9 \* min(p\_vals$new\_y.position),   
## NA), expand = expansion(mult = c(0, 0.1))) + theme(line = element\_blank(),   
## axis.title = element\_blank(), axis.text = element\_blank()) +   
## theme(plot.margin = margin(t = 0, r = 0, b = 0, l = 0))  
## }  
## debug at /home/chris/bmrn-test/R/prism\_plot.R#261: top = top + theme(line = element\_blank(), axis.title = element\_blank(),   
## axis.text = element\_blank()) + theme(plot.margin = margin(t = 0,   
## r = 0, b = 0, l = 0))  
## debug at /home/chris/bmrn-test/R/prism\_plot.R#291: if (type == "box") {  
## top <- top + ggtitle(paste("Box plot for Treatment Groups at",   
## time\_sel))  
## combined <- grid.arrange(grobs = list(top, bottom), layout\_matrix = if (format ==   
## "word") {  
## rbind(1, 2, 2)  
## }  
## else {  
## rbind(1, 2, 2)  
## })  
## } else {  
## top <- top + ggtitle(paste("Bar Chart for Treatment Groups at",   
## time\_sel))  
## combined <- grid.arrange(grobs = list(top, bottom), layout\_matrix = if (format ==   
## "word") {  
## rbind(1, 2, 2)  
## }  
## else {  
## rbind(1, 2, 2)  
## })  
## }  
## debug at /home/chris/bmrn-test/R/prism\_plot.R#292: top <- top + ggtitle(paste("Box plot for Treatment Groups at",   
## time\_sel))  
## debug at /home/chris/bmrn-test/R/prism\_plot.R#293: combined <- grid.arrange(grobs = list(top, bottom), layout\_matrix = if (format ==   
## "word") {  
## rbind(1, 2, 2)  
## } else {  
## rbind(1, 2, 2)  
## })

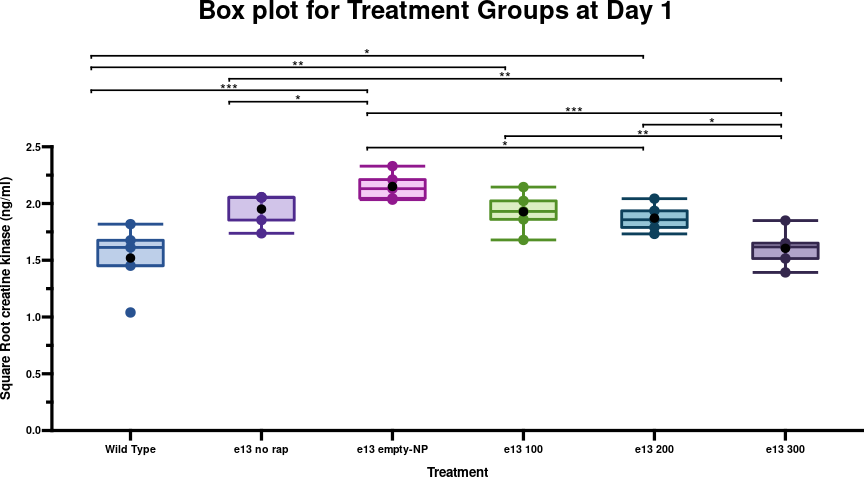


Figure 4.7: Box plot of Transformed Scale response values at Day 1. The stars above the horizontal bars correspond to the magnitude of the p-value comparing two groups where “\*”, “\*\*”, and “\*\*\*” represent 0.01 <= p-value < 0.05, 0.001 <= p-value < 0.01, and p-value < 0.001, respectively. A p-value summary will only be shown form p-value less than 0.05.

## debug at /home/chris/bmrn-test/R/prism\_plot.R#293: rbind(1, 2, 2)  
## debug at /home/chris/bmrn-test/R/prism\_plot.R#300: return(combined)

## 4.4 Tables

### 4.4.1 Table 1: Summary of creatine kinase (ng/ml)

|  | | creatine kinase (ng/ml) | | |
| --- | --- | --- | --- | --- |
| Treatment | Time Points | Mean | Median | SE |
| Wild Type | Day 1 | -0.35 | -0.12 | 0.169 |
| e13 no rap | Day 1 | 1.14 | 1.47 | 0.123 |
| e13 empty-NP | Day 1 | 1.92 | 1.84 | 0.11 |
| e13 100 | Day 1 | 1.02 | 0.99 | 0.128 |
| e13 200 | Day 1 | 0.79 | 0.71 | 0.094 |
| e13 300 | Day 1 | -0.11 | -0.12 | 0.11 |
| Group level summary on the original scale. | | | | |

### 4.4.2 Table 2: Comparison between Controls and Wild Type as to creatine kinase (ng/ml)

|  | | Transformed Scale | | Back Transformed | | vs. e13 empty-NP | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Treatment | Time Points | Mean | SE | Mean | SE | LSMean Diff (95% CI) | p value |
| Wild Type | Day 1 | 1.52 | 0.116 | 2.31 | 0.35 | -0.63 (-0.9, -0.36) | < 0.001\* |
| e13 no rap | Day 1 | 1.95 | 0.069 | 3.81 | 0.27 | -0.2 (-0.394, -0.004) | 0.046\* |
| A Square Root transformation was applied to the data. Difference and CI are estimated using model based LSmean | | | | | | | |

### 4.4.3 Table 3: Comparison between Doses as to creatine kinase (ng/ml)

|  | | Transformed Scale | | Back Transformed | | vs. e13 100 | | vs. e13 200 | | vs. e13 300 | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Treatment | Time Points | Mean | SE | Mean | SE | LSMean Diff (95% CI) | p value | LSMean Diff (95% CI) | p value | LSMean Diff (95% CI) | p value |
| e13 empty-NP | Day 1 | 2.15 | 0.069 | 4.62 | 0.3 | 0.22 (-0.012, 0.457) | 0.067 | 0.28 (0.044, 0.514) | 0.016\* | 0.54 (0.31, 0.78) | < 0.001\* |
| e13 100 | Day 1 | 1.93 | 0.069 | 3.71 | 0.27 |  |  | 0.06 (-0.179, 0.291) | 0.834 | 0.32 (0.088, 0.556) | 0.004\* |
| e13 200 | Day 1 | 1.87 | 0.069 | 3.5 | 0.26 |  |  |  |  | 0.27 (0.032, 0.5) | 0.022\* |
| e13 300 | Day 1 | 1.61 | 0.069 | 2.58 | 0.22 |  |  |  |  |  |  |
| A Square Root transformation was applied to the data. Difference and CI are estimated using model based LSmean | | | | | | | | | | | |

### 4.4.4 Table 4: Comparison Doses and Controls/Wild Type as to creatine kinase (ng/ml)

|  | | vs. Wild Type | | vs. e13 no rap | |
| --- | --- | --- | --- | --- | --- |
| Treatment | Time Points | LSMean Diff (95% CI) | p value | LSMean Diff (95% CI) | p value |
| e13 100 | Day 1 | 0.41 (0.092, 0.724) | 0.009\* | -0.02 (-0.258, 0.211) | 0.99 |
| e13 200 | Day 1 | 0.35 (0.036, 0.667) | 0.026\* | -0.08 (-0.314, 0.155) | 0.753 |
| e13 300 | Day 1 | 0.09 (-0.23, 0.401) | 0.806 | -0.35 (-0.581, -0.11) | 0.002\* |
| A Square Root transformation was applied to the data. Difference and CI are estimated using model based LSmean | | | | | |

# 5 References

Allaire, JJ, Yihui Xie, Jonathan McPherson, Javier Luraschi, Kevin Ushey, Aron Atkins, Hadley Wickham, Joe Cheng, Winston Chang, and Richard Iannone. 2020. *Rmarkdown: Dynamic Documents for r*. <https://github.com/rstudio/rmarkdown>.

Dawson, Charlotte. 2021. *Ggprism: A ’Ggplot2’ Extension Inspired by ’GraphPad Prism’*. <https://doi.org/10.5281/zenodo.4556067>.

Edwards, Don, and Jack J. Berry. 1987. “The Efficiency of Simulation-Based Multiple Comparisons.” *Biometrics* 43 (4): 913–28. <http://www.jstor.org/stable/2531545>.

Lenth, Russell V. 2022. *Emmeans: Estimated Marginal Means, Aka Least-Squares Means*. <https://CRAN.R-project.org/package=emmeans>.

Liu, Wei, Mortaza Jamshidian, Ying Zhang, Frank Bretz, and Xiaoliang Han. 2007. “Some New Methods for the Comparison of Two Linear Regression Models.” *Journal of Statistical Planning and Inference* 137 (1): 57–67. https://doi.org/<https://doi.org/10.1016/j.jspi.2005.09.007>.

Pinheiro, Jose, Douglas Bates, Saikat DebRoy, Deepayan Sarkar, and R Core Team. 2013. *Nlme: Linear and Nonlinear Mixed Effects Models*.

R Core Team. 2016. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. <https://www.R-project.org/>.

RStudio, Inc. 2013. *Easy Web Applications in r.*

Sievert, Carson. 2020. *Interactive Web-Based Data Visualization with r, Plotly, and Shiny*. Chapman; Hall/CRC. <https://plotly-r.com>.

Wickham, Hadley. 2016. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. <https://ggplot2.tidyverse.org>.

# 6 Appendix: Workflow Illustration

