204_Behave_Project

Wei Dou, Jacobo Pereira-Pacheco and Yu Zhu

12/9/2020

```r
library(readxl)
library(naniar)
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(hrbrthemes)
library(vcd)
```

```
## Loading required package: grid
```

```r
library(DescTools)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v tibble  3.0.4     v purrr   0.3.4
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(lme4)
```

```
## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```r
library(stargazer)
```

```
##
```

```
## Please cite as:
##  Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics Tables.
##  R package version 5.2.2. https://CRAN.R-project.org/package=stargazer
```

```r
library(lmtest)
```

```
## Loading required package: zoo
##
## Attaching package: 'zoo'
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
library(ROCR)
library(yardstick)
```

```
## For binary classification, the first factor level is assumed to be the event.
## Use the argument `event_level = "second"` to alter this as needed.
##
## Attaching package: 'yardstick'
## The following object is masked from 'package:readr':
##
##     spec
```

```r
library(caret)
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
## The following objects are masked from 'package:yardstick':
##
##     precision, recall, sensitivity, specificity
## The following object is masked from 'package:purrr':
##
##     lift
## The following objects are masked from 'package:DescTools':
##
##     MAE, RMSE
```

```r
library(arm)
```

```
## Loading required package: MASS
##
## Attaching package: 'MASS'
## The following object is masked from 'package:dplyr':
##
##     select
##
## arm (Version 1.11-2, built: 2020-7-27)
## Working directory is /Users/zoeyyuzhu/Desktop/ucsc/courses/stat 204/project/STAT_204_Project
```

I. Data Description

1.1 Import data

```
Behave <- read_excel("~/Desktop/ucsc/courses/stat 204/project/ProjectDataset_behave.xlsx")
```

1.2 Summarization

```
summary(Behave)
```

```
##  ParticipantNum  StrengthLevel    ResponseTime       Accuracy
##  Min.   : 1.00   Min.   : 1.00   Min.   :0.3074   Min.   :0.0000
##  1st Qu.: 6.00   1st Qu.: 4.50   1st Qu.:0.6832   1st Qu.:1.0000
##  Median :13.00   Median :12.00   Median :0.8416   Median :1.0000
##  Mean   :12.84   Mean   :15.14   Mean   :0.8995   Mean   :0.7659
##  3rd Qu.:19.00   3rd Qu.:25.00   3rd Qu.:1.0666   3rd Qu.:1.0000
##  Max.   :25.00   Max.   :40.00   Max.   :1.9395   Max.   :1.0000
##    Confidence
##  Min.   :1.000
##  1st Qu.:2.000
##  Median :3.000
##  Mean   :2.556
##  3rd Qu.:3.000
##  Max.   :4.000
```

```
str(Behave)
```

```
## tibble [20,447 x 5] (S3: tbl_df/tbl/data.frame)
##  $ ParticipantNum: num [1:20447] 1 1 1 1 1 1 1 1 1 1 ...
##  $ StrengthLevel : num [1:20447] 8 4.5 4.5 1 4.5 25 4.5 4.5 25 40 ...
##  $ ResponseTime  : num [1:20447] 0.904 1.049 1.224 0.905 1.279 ...
##  $ Accuracy      : num [1:20447] 1 1 1 0 1 1 0 0 1 1 ...
##  $ Confidence    : num [1:20447] 3 1 2 2 1 3 1 1 2 3 ...
```

```
head(Behave)
```

```
## # A tibble: 6 x 5
##   ParticipantNum StrengthLevel ResponseTime Accuracy Confidence
##            <dbl>         <dbl>        <dbl>    <dbl>      <dbl>
## 1              1           8          0.904        1          3
## 2              1           4.5        1.05         1          1
## 3              1           4.5        1.22         1          2
## 4              1           1          0.905        0          2
## 5              1           4.5        1.28         1          1
## 6              1          25          0.792        1          3
```

```
# Show the frequency of the variables
table(Behave$ParticipantNum)
```

```
##
##    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16
##  996  929  733 1015  896  582  867  700  883  903  621  766  934  744  745  877
##   17   18   19   20   21   22   23   24   25
##  848  789  716  903  775  637  728  854 1006
```

```
table(Behave$Confidence)
```

```
##
```

```
##    1    2    3    4
## 3962 5901 5841 4743
```

```r
table(Behave$StrengthLevel)
```

```
##
##    1  4.5    8   12   25   40
## 3348 3445 3400 3364 3497 3393
```

```r
table(Behave$Accuracy)
```

```
##
##     0     1
##  4787 15660
```

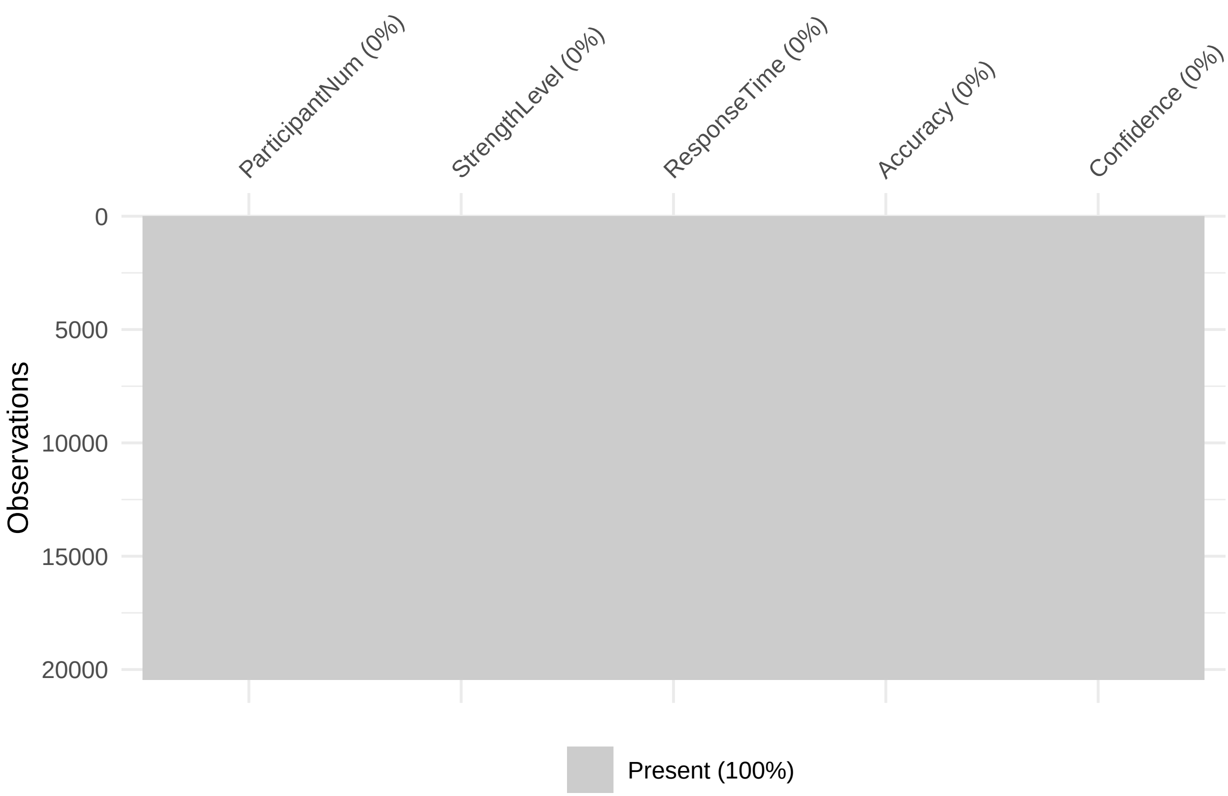
1.3 Factorization

```r
Behave$ParticipantNum = as.factor(Behave$ParticipantNum)
Behave$Confidence = as.factor(Behave$Confidence)
Behave$StrengthLevel = as.factor(Behave$StrengthLevel)
```

II. Explotary Data Analysis

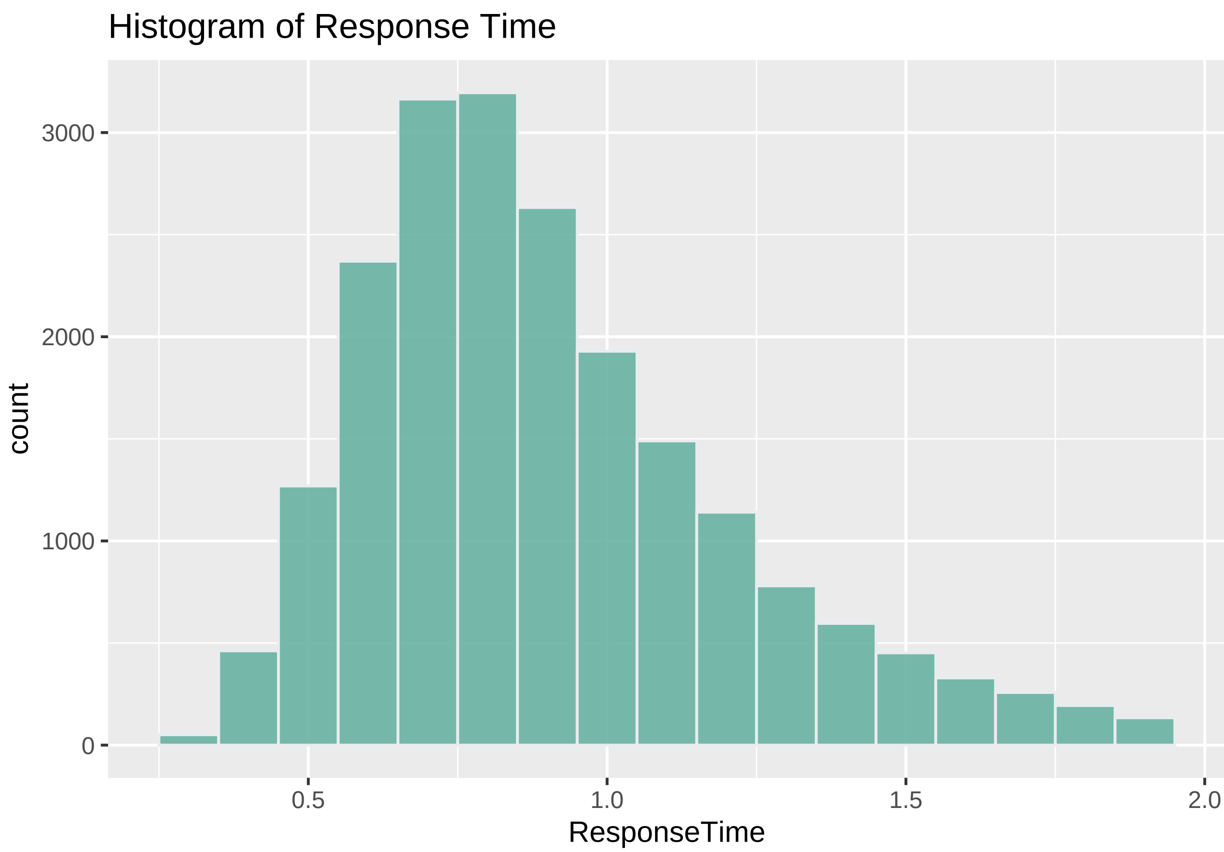
2.1 Missing Data

```r
vis_miss(Behave)
```



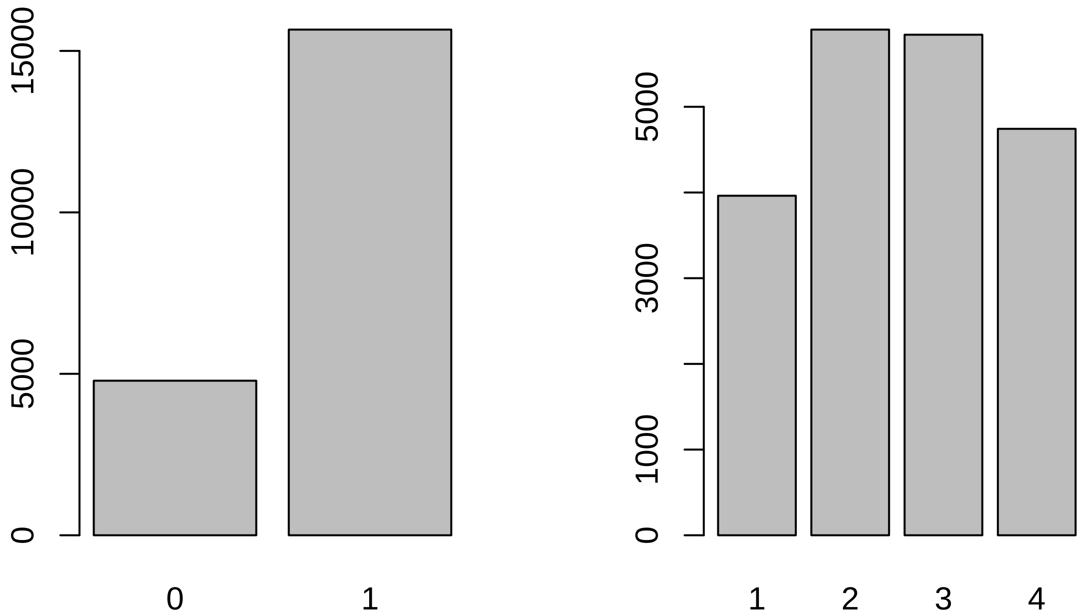
2.2 Potential Outliers for Response Time

```
p <- ggplot(Behave, aes(x=ResponseTime)) +
  geom_histogram(binwidth=0.1, alpha=0.9, fill="#69b3a2", color="#e9ecef") +
  ggtitle("Histogram of Response Time")
p
```



2.3 Visualize Accuracy and Confidence

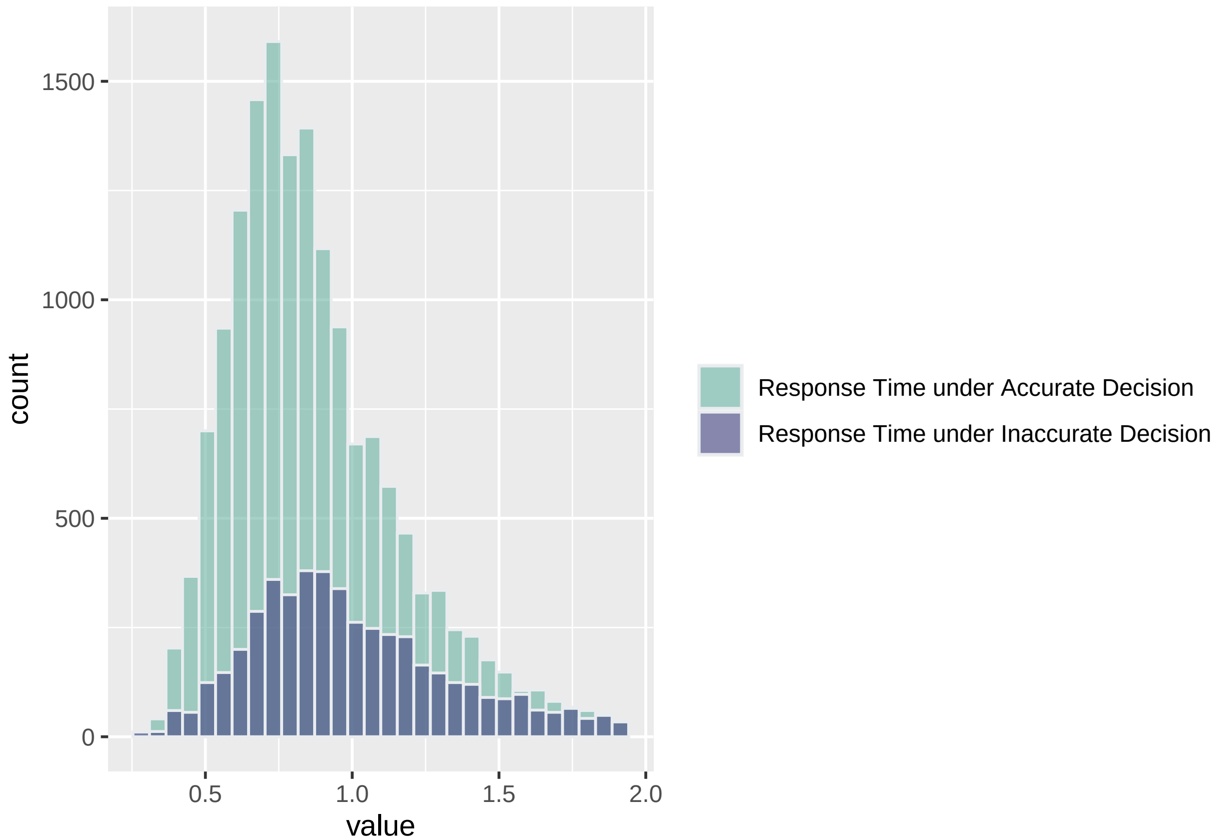
```
par(mfrow = c(1,2))
plot(as.factor(Behave$Accuracy))
plot(as.factor(Behave$Confidence))
```



2.4 The Histograms for Response Time under Accuracy = 1 and 0 Separately

```
Accurate_RT = Behave$ResponseTime[Behave$Accuracy == 1]
Inaccurate_RT = Behave$ResponseTime[Behave$Accuracy == 0]
acc_data <- data.frame(
  type = c( rep("Response Time under Accurate Decision", length(Accurate_RT)), rep("Response Time under
  value = c( Accurate_RT, Inaccurate_RT )
)

p <- acc_data %>%
  ggplot( aes(x=value, fill=type)) +
    geom_histogram( color="#e9ecef", alpha=0.6, position = 'identity') +
    scale_fill_manual(values=c("#69b3a2", "#404080")) +
    labs(fill="")
p
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



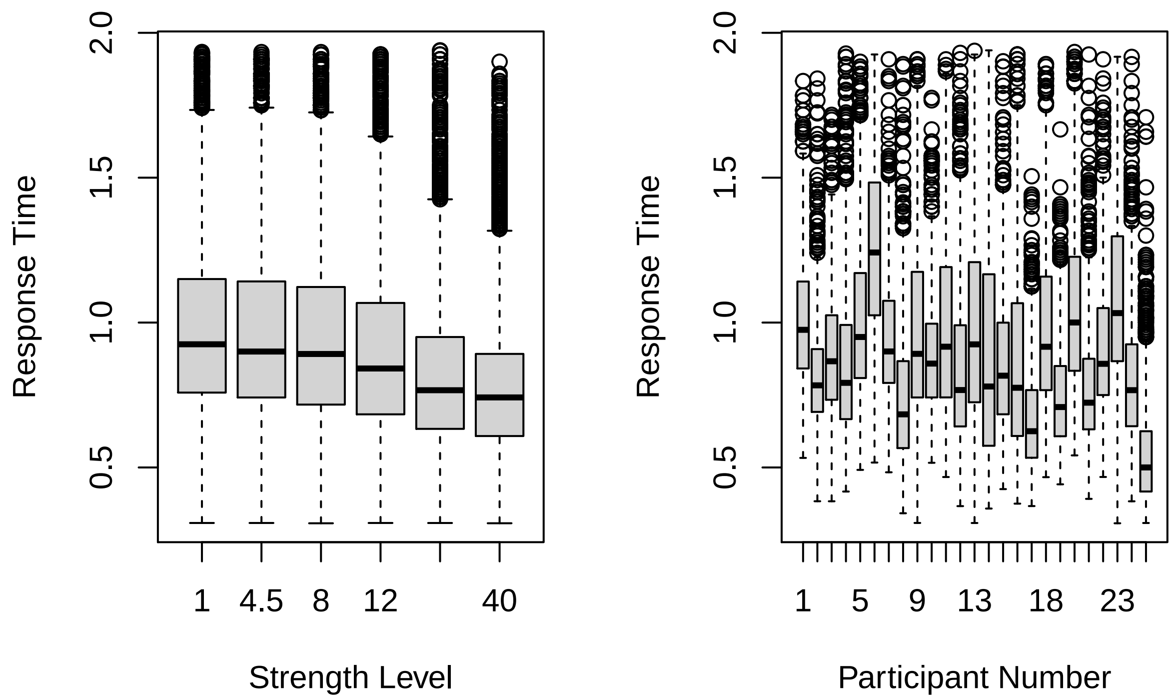
2.5 Contingency Table of Strength level VS Participant Number

```
table(Behave$StrengthLevel, Behave$ParticipantNum)
```

```
##
##          1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##   1    166 159 138 169 158  84 140 119 121 145 107 123 164 130 144 136 142 120
##   4.5  172 154 133 166 167  80 147 116 167 164  98 131 171 118 121 127 126 135
##   8    162 147 116 174 150  96 153 102 144 159 105 139 177 126 129 137 136 138
##   12   168 129 131 177 161 111 132 115 159 124  97 128 147 130  96 160 150 126
##   25   157 165 120 181 127 107 151 136 159 164  97 122 135 108 138 163 146 137
##   40   171 175  95 148 133 104 144 112 133 147 117 123 140 132 117 154 148 133
##
##         19  20  21  22  23  24  25
##   1    111 144 127 116 107 120 158
##   4.5  133 140 141 112 117 149 160
##   8    122 147 121  89 106 142 183
##   12   112 140 142 109 125 123 172
##   25   129 174 122  92 141 167 159
##   40   109 158 122 119 132 153 174
```

2.6 Box Plots for Response Time vs Strength Level and Response Time vs Participants

```
par(mfrow = c(1,2))
boxplot(ResponseTime~StrengthLevel, data = Behave, xlab="Strength Level", ylab="Response Time")
boxplot(ResponseTime~ParticipantNum, data = Behave, xlab="Participant Number", ylab="Response Time")
```



2.7 Mosaic Plots of Confidence vs Strength Level and Accuracy VS Strength Level

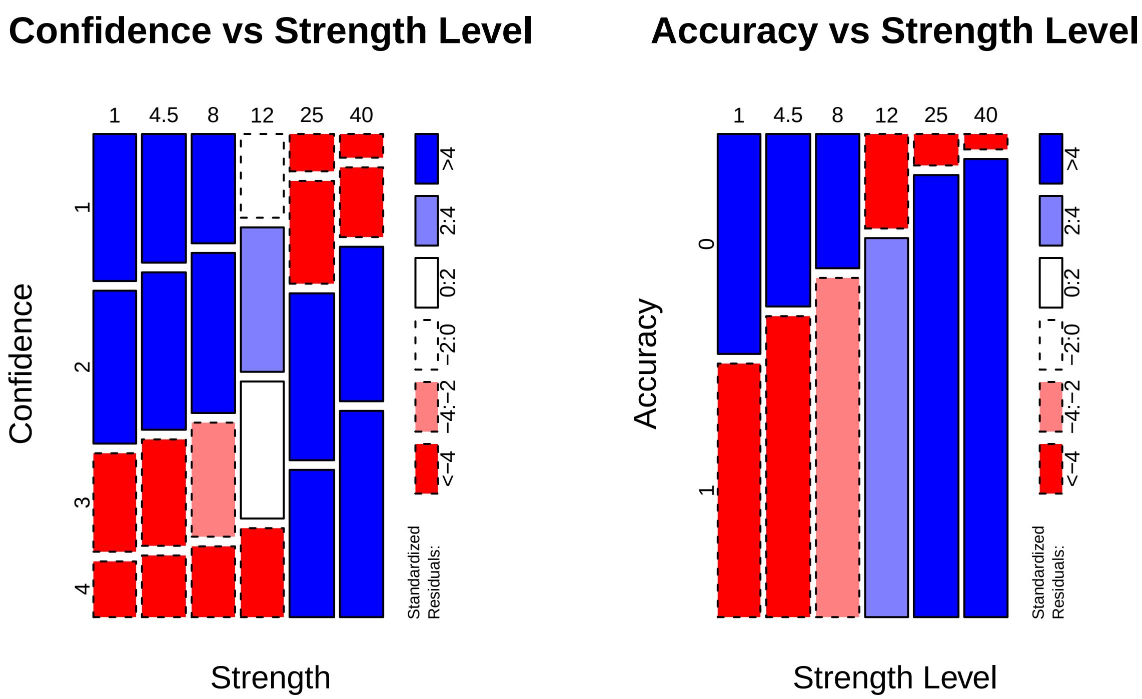
```
par(mfrow = c(1,2))
tb_Str_Conf = table(Behave$StrengthLevel, Behave$Confidence); tb_Str_Conf
```

```
##
##          1    2    3    4
##   1   1084 1127  726  411
##   4.5  976 1194  807  468
##   8    818 1198  854  530
##   12   620 1070 1015  659
##   25   287  791 1285 1134
##   40   177  521 1154 1541
```

```
mosaicplot(tb_Str_Conf, main="Confidence vs Strength Level", xlab="Strength", ylab="Confidence",shade=TI

tb_Stren_Acc = table(Behave$StrengthLevel, Behave$Accuracy); tb_Stren_Acc
```

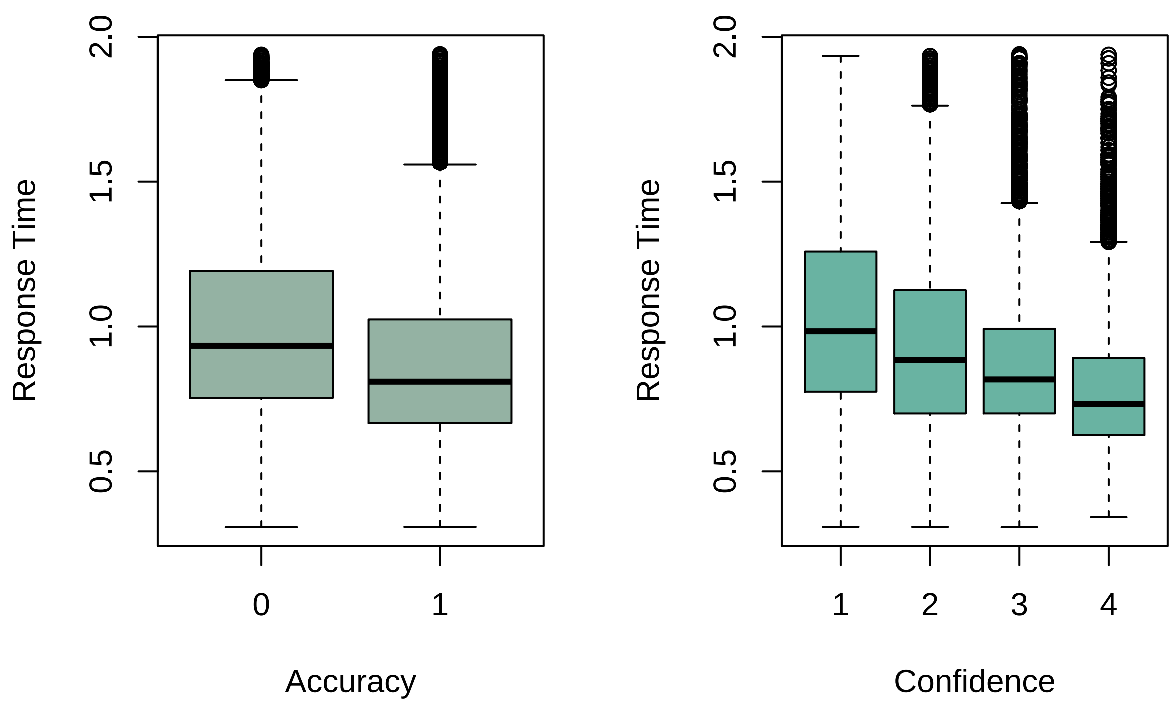
```
##
##          0    1
##   1   1555 1793
##   4.5 1255 2190
##   8    964 2436
##   12   671 2693
##   25   232 3265
##   40   110 3283
```

```
mosaicplot(tb_Stren_Acc, shade = TRUE, main="Accuracy vs Strength Level", xlab="Strength Level", ylab="/
```



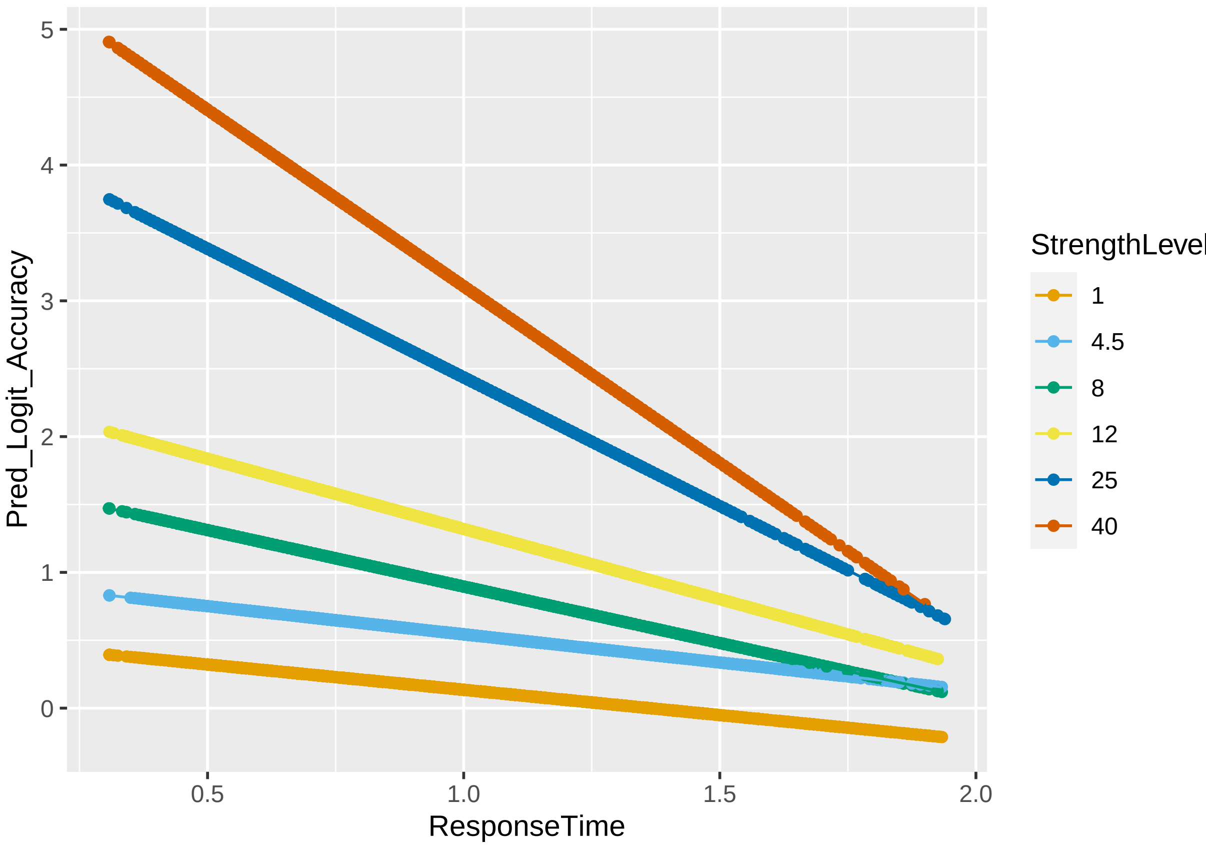
2.8 Box Plots of Reponse time Vs Accuracy and Reponse time Vs Confidence

```
par(mfrow = c(1,2))
boxplot(Behave$ResponseTime~Behave$Accuracy, xlab = "Accuracy", ylab = "Response Time", col=rgb(0.3,0.5
boxplot(Behave$ResponseTime~Behave$Confidence, xlab = "Confidence", ylab = "Response Time", col="#69b3a2
```



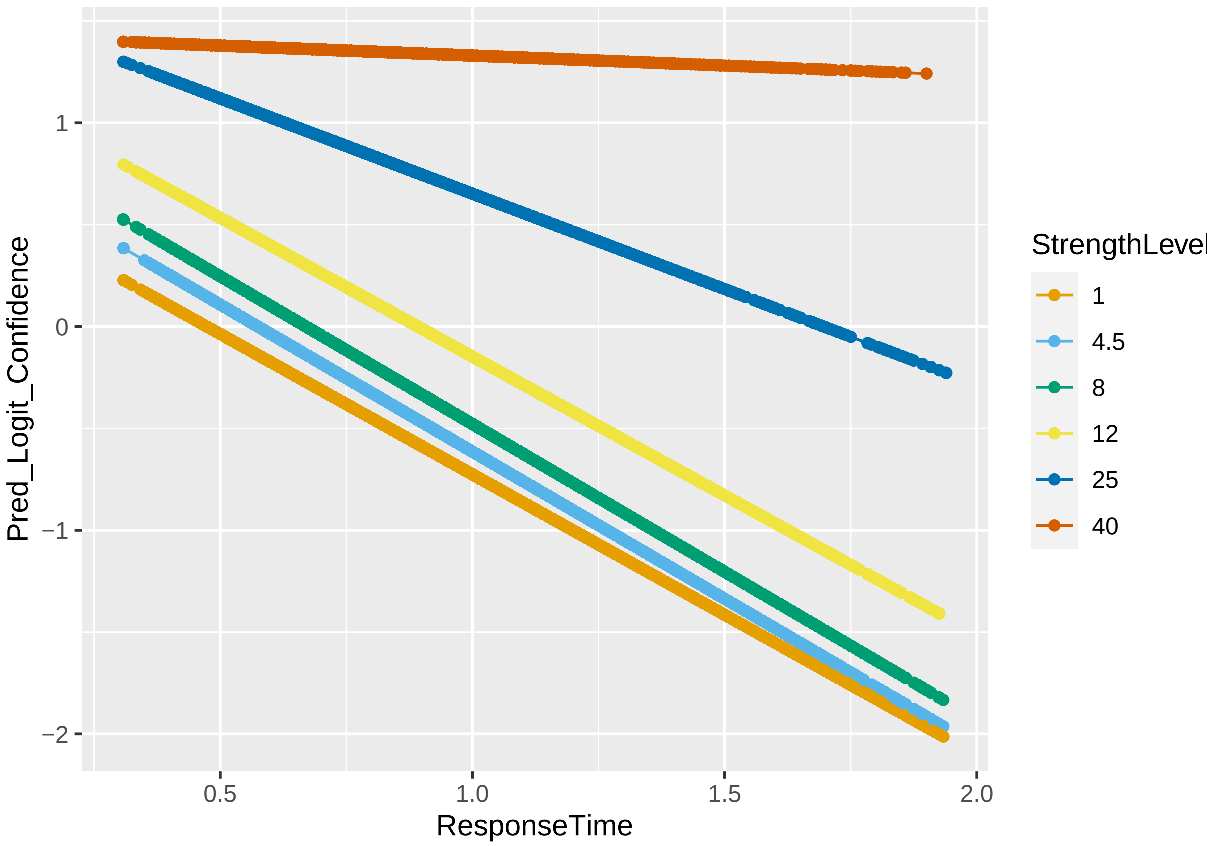
2.9 Interactions between Strength Level and Response Time

```r
glm1 <- glm(Accuracy~StrengthLevel*ResponseTime, data = Behave,
            family = "binomial")
newdf = cbind(Behave, Pred_Logit_Accuracy = predict(glm1))

ggplot(newdf) +
  aes(x = ResponseTime, y = Pred_Logit_Accuracy, color = StrengthLevel) +
  geom_point() +
  geom_line() +
  scale_color_manual(
        values=c("#E69F00", "#56B4E9", "#009E73", "#F0E442", "#0072B2", "#D55E00"))
```




```r
behave <- Behave %>%
  mutate(Conf = ifelse(as.numeric(Confidence) > 2, 1, 0))
```

```r
glm2 <- glm(Conf~StrengthLevel*ResponseTime, data = behave,
            family = "binomial")
newdf2 = cbind(Behave, Pred_Logit_Confidence = predict(glm2))

ggplot(newdf2) +
  aes(x = ResponseTime, y = Pred_Logit_Confidence, color = StrengthLevel) +
  geom_point() +
  geom_line() +
  scale_color_manual(
        values=c("#E69F00", "#56B4E9", "#009E73", "#F0E442", "#0072B2", "#D55E00"))
```



III. Randomized Block Model

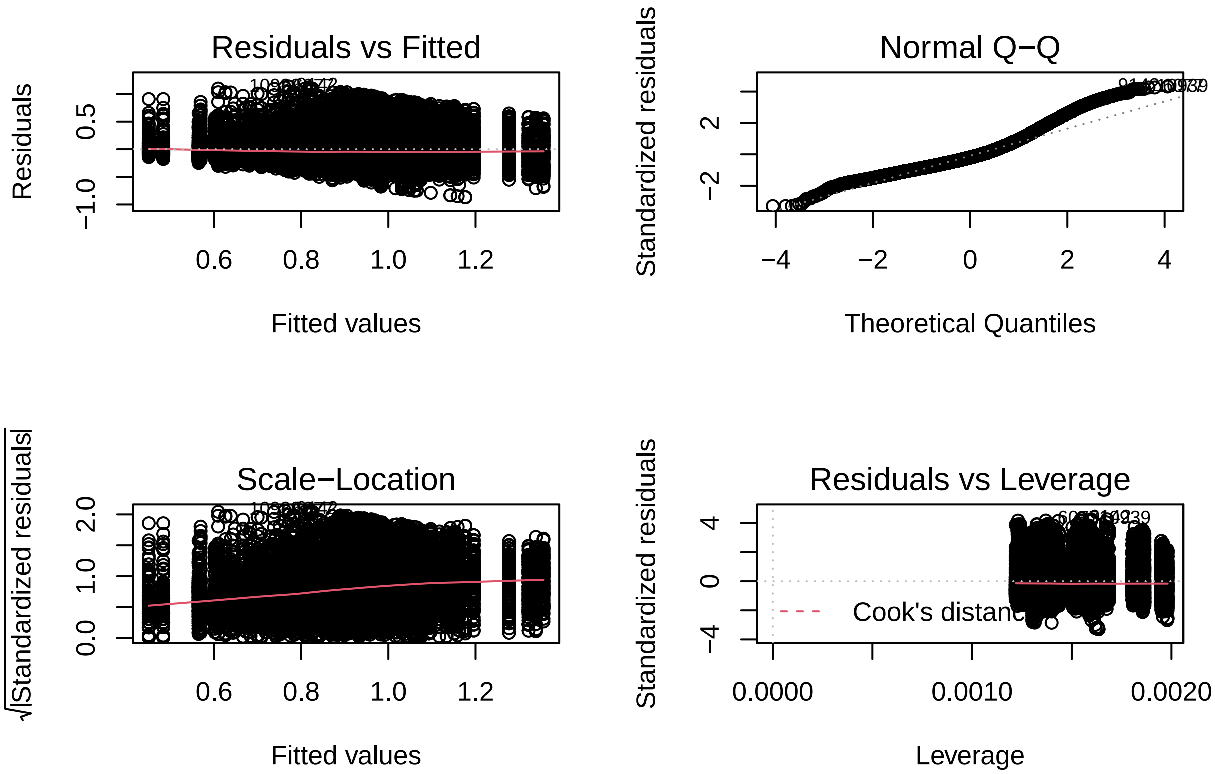
3.1 Build ANOVA Model with Interaction

```
L_aov = aov(ResponseTime~StrengthLevel+ParticipantNum, data = Behave)
summary(L_aov)
```

```
##                   Df Sum Sq Mean Sq F value Pr(>F)
## StrengthLevel      5  106.2  21.246   305.9 <2e-16 ***
## ParticipantNum    24  396.5  16.520   237.8 <2e-16 ***
## Residuals      20417 1418.2   0.069
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

3.2 Diagnostics

```
par(mfrow=c(2,2))
plot(L_aov)
```

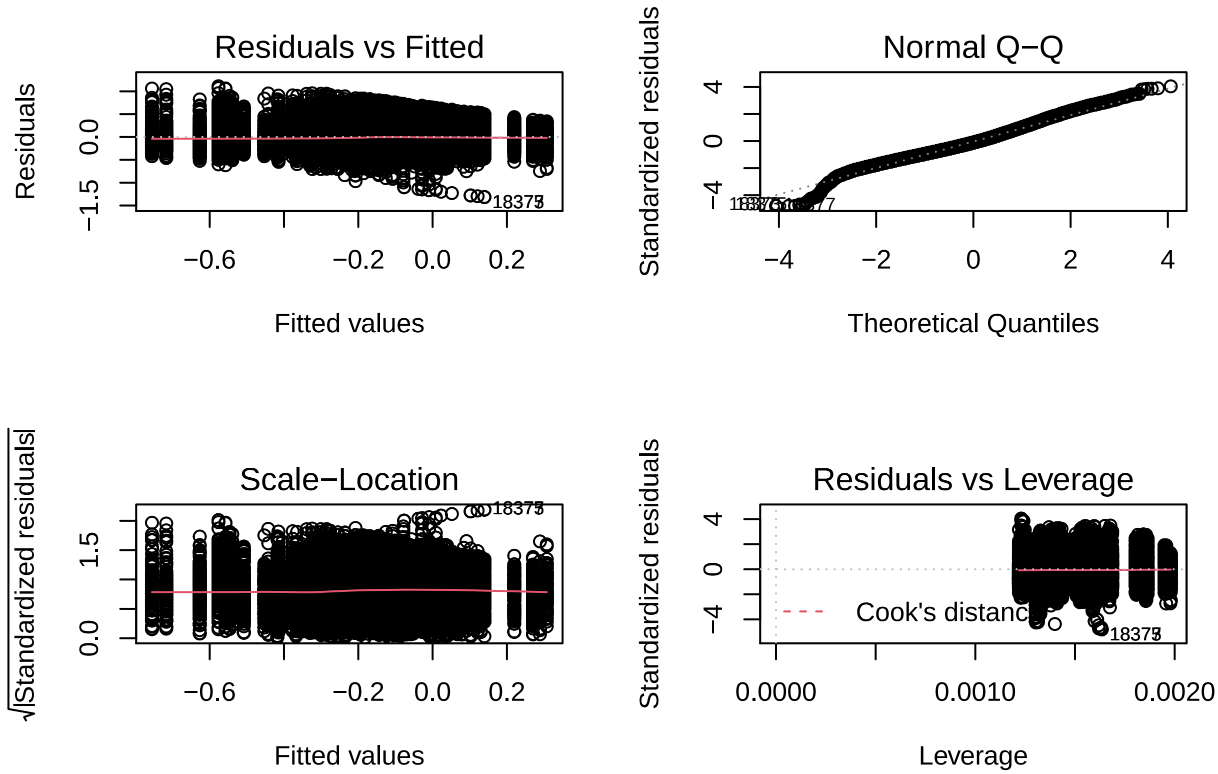


3.3 Log-transformation for Response Time

```
L_aov_log = aov(log(ResponseTime)~StrengthLevel+ParticipantNum, data = Behave)
summary(L_aov_log)
```

```
##                   Df Sum Sq Mean Sq F value Pr(>F)
## StrengthLevel      5  133.4  26.681   353.4 <2e-16 ***
## ParticipantNum    24  566.1  23.586   312.4 <2e-16 ***
## Residuals      20417 1541.2   0.075
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
par(mfrow=c(2,2))
plot(L_aov_log)
```

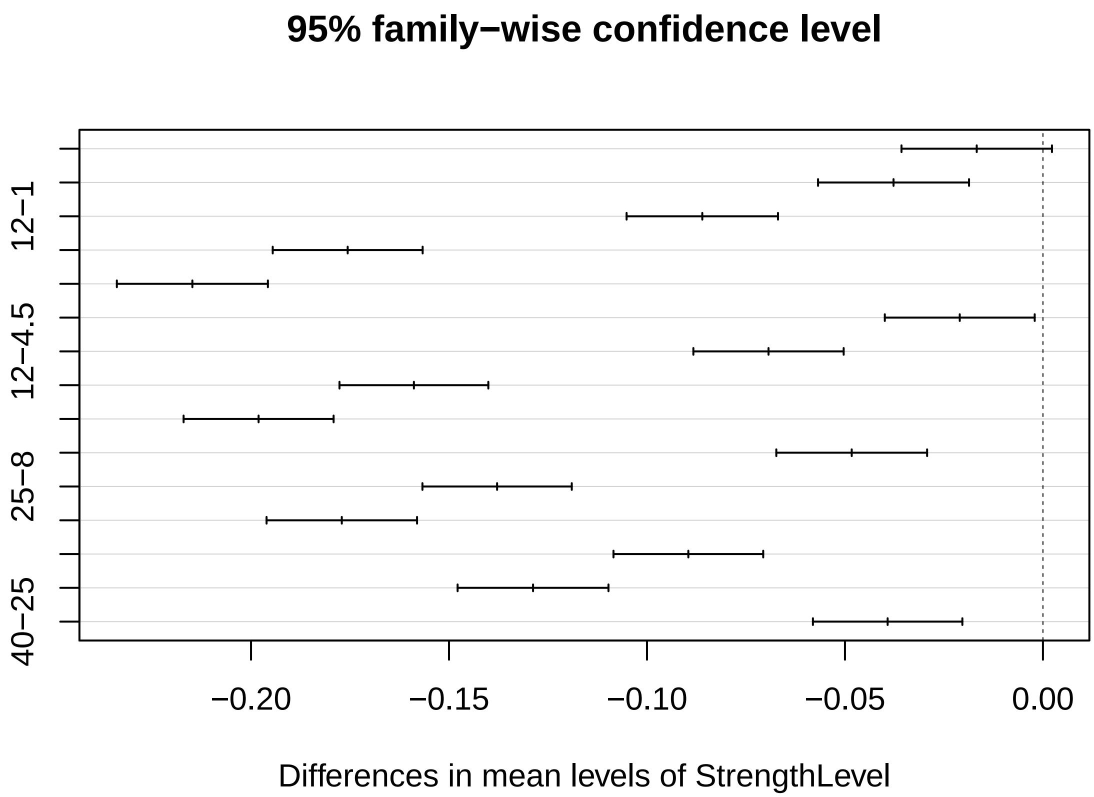


3.4 Tukey Method for Pairwise Comparisons

```
CIs_strLevel = TukeyHSD(L_aov_log, which = 1)
CIs_strLevel
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = log(ResponseTime) ~ StrengthLevel + ParticipantNum, data = Behave)
##
## $StrengthLevel
##               diff         lwr          upr     p adj
## 4.5-1  -0.01672493 -0.03572796  0.002278099 0.1216025
## 8-1    -0.03773623 -0.05680114 -0.018671321 0.0000002
## 12-1   -0.08603296 -0.10514842 -0.066917510 0.0000000
## 25-1   -0.17558830 -0.19452157 -0.156655034 0.0000000
## 40-1   -0.21481107 -0.23388573 -0.195736408 0.0000000
## 8-4.5  -0.02101130 -0.03994049 -0.002082109 0.0194605
## 12-4.5 -0.06930803 -0.08828813 -0.050327938 0.0000000
## 25-4.5 -0.15886337 -0.17765997 -0.140066773 0.0000000
## 40-4.5 -0.19808614 -0.21702515 -0.179147126 0.0000000
## 12-8   -0.04829674 -0.06733879 -0.029254685 0.0000000
## 25-8   -0.13785207 -0.15671123 -0.118992918 0.0000000
## 40-8   -0.17707484 -0.19607594 -0.158073740 0.0000000
## 25-12  -0.08955534 -0.10846559 -0.070645086 0.0000000
## 40-12  -0.12877811 -0.14782992 -0.109726289 0.0000000
## 40-25  -0.03922277 -0.05809179 -0.020353755 0.0000000
```

```r
plot(CIs_strLevel)
```



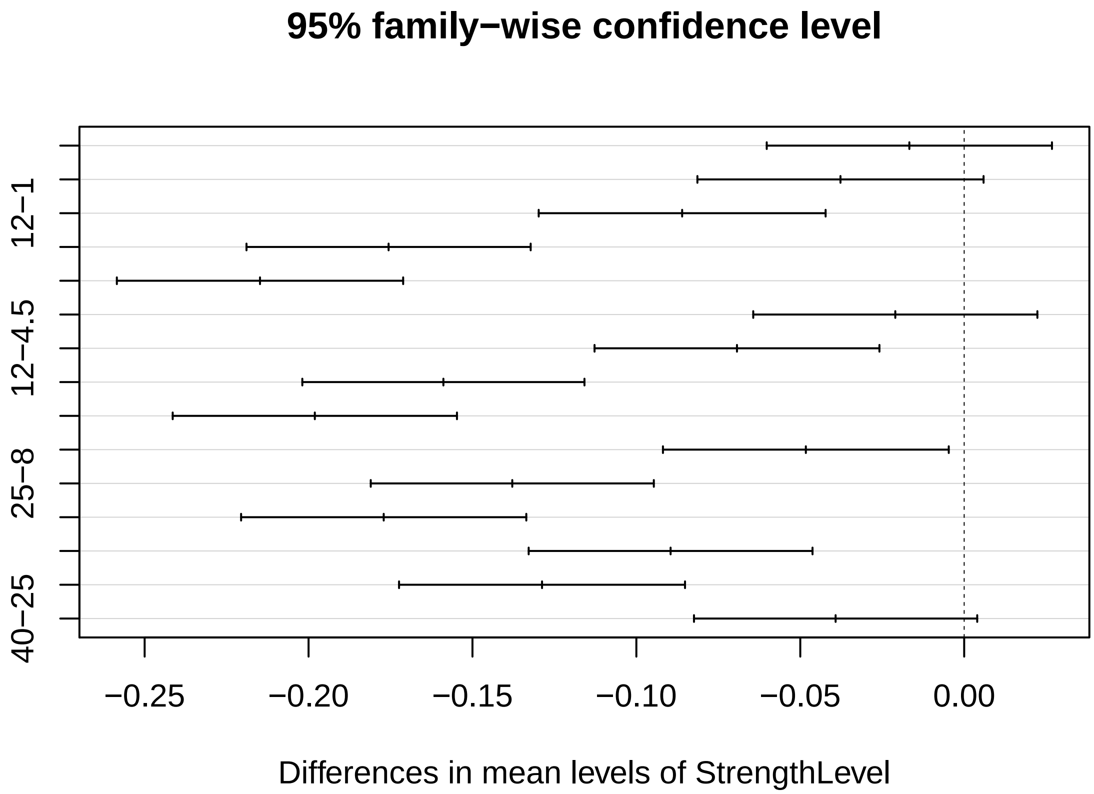
3.5 Scheffe's Method for Pairwise Comparisons

```r
ScheffeTest(x=L_aov_log, which="StrengthLevel")
```

```
##
##   Posthoc multiple comparisons of means: Scheffe Test
##     95% family-wise confidence level
##
## $StrengthLevel
##               diff      lwr.ci       upr.ci    pval
## 4.5-1  -0.01672493 -0.06023098  0.026781116  1.0000
## 8-1    -0.03773623 -0.08138394  0.005911485  0.3279
## 12-1   -0.08603296 -0.12979640 -0.042269530 < 2e-16 ***
## 25-1   -0.17558830 -0.21893463 -0.132241971 < 2e-16 ***
## 40-1   -0.21481107 -0.25848112 -0.171141024 < 2e-16 ***
## 8-4.5  -0.02101130 -0.06434829  0.022325697  0.9996
## 12-4.5 -0.06930803 -0.11276158 -0.025854490 4.8e-11 ***
## 25-4.5 -0.15886337 -0.20189681 -0.115829935 < 2e-16 ***
## 40-4.5 -0.19808614 -0.24144563 -0.154726652 < 2e-16 ***
## 12-8   -0.04829674 -0.09189212 -0.004701353  0.0052 **
## 25-8   -0.13785207 -0.18102873 -0.094675418 < 2e-16 ***
## 40-8   -0.17707484 -0.22057648 -0.133573208 < 2e-16 ***
## 25-12  -0.08955534 -0.13284897 -0.046261702 < 2e-16 ***
## 40-12  -0.12877811 -0.17239585 -0.085160364 < 2e-16 ***
## 40-25  -0.03922277 -0.08242200  0.003976460  0.2016
##
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
plot(ScheffeTest(x=L_aov_log, which="StrengthLevel"))
```



IV. Train-test Split

4.1 Train-test Split and Factorization

```r
set.seed(1234)
train = Behave %>% group_by(ParticipantNum, StrengthLevel) %>% sample_n(80)
# write.csv(train,"~/Desktop/ucsc/courses/stat 204/project/Accuracy_Training.csv", row.names = FALSE)

test = dplyr::anti_join(Behave, train)
```

```
## Joining, by = c("ParticipantNum", "StrengthLevel", "ResponseTime", "Accuracy", "Confidence")
```

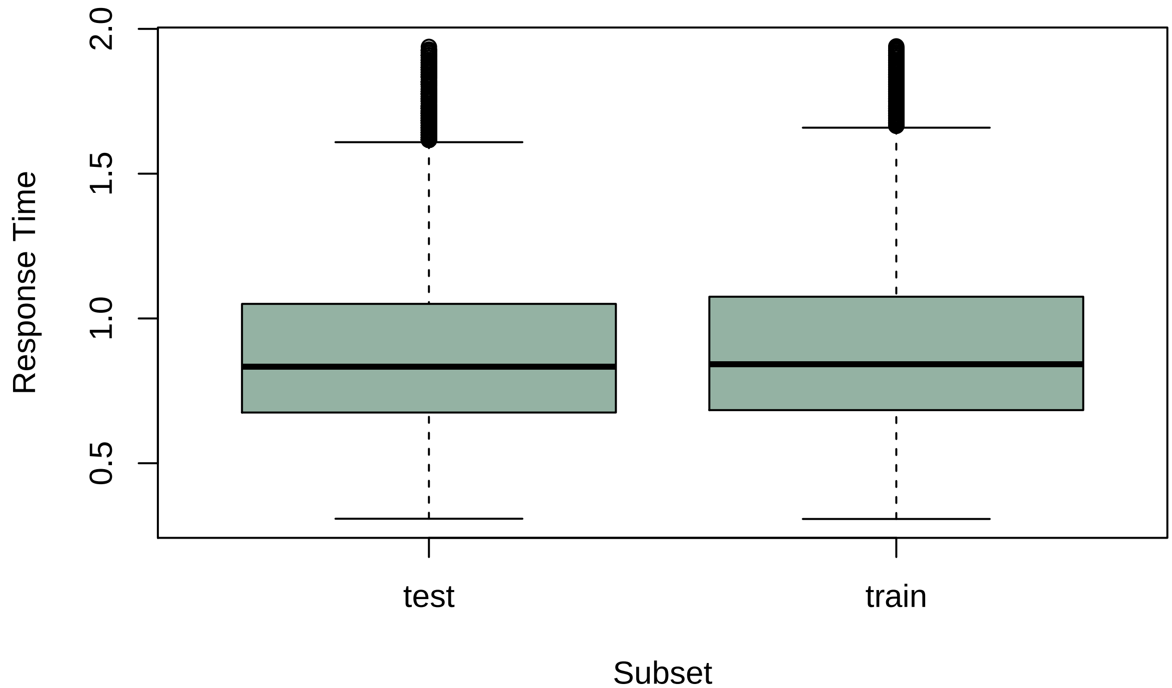
```r
# write.csv(test,"~/Desktop/ucsc/courses/stat 204/project/Accuracy_Testing.csv", row.names = FALSE)

### import the exported training and testing data
train = read.csv("./data/Accuracy_Training.csv")
test = read.csv("./data/Accuracy_Testing.csv")
```

```r
# Factorization
train$ParticipantNum = as.factor(train$ParticipantNum)
train$Confidence = as.factor(train$Confidence)
train$StrengthLevel = as.factor(train$StrengthLevel)

test$ParticipantNum = as.factor(test$ParticipantNum)
test$Confidence = as.factor(test$Confidence)
```

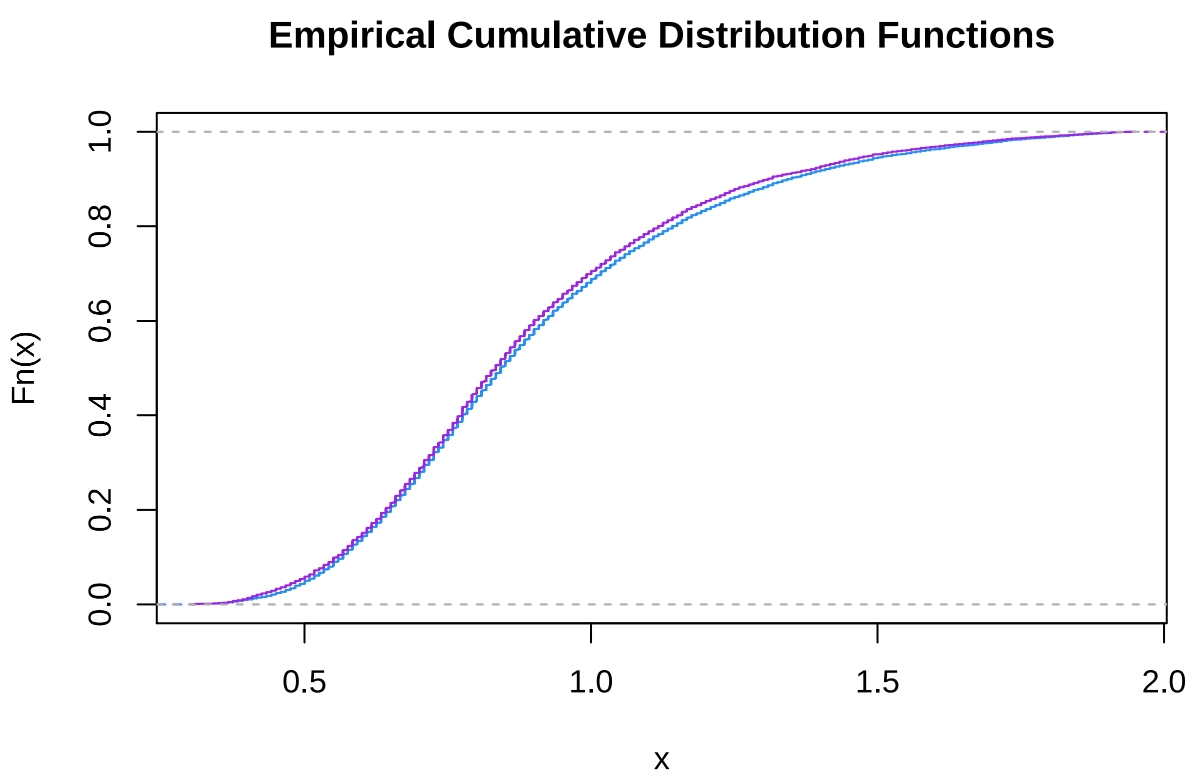
```
test$StrengthLevel = as.factor(test$StrengthLevel)
```

4.2 Brief Check for the Distributions Similarity of Train and Test Sets

```
box_train_rt = data.frame(ResponseTime = train$ResponseTime, set = rep("train", length(train$ResponseTi
box_test_rt = data.frame(ResponseTime = test$ResponseTime, set = rep("test", length(test$ResponseTime))
box_rt = rbind(box_train_rt,box_test_rt)
boxplot(box_rt$ResponseTime~box_rt$set, xlab = "Subset", ylab = "Response Time", col=rgb(0.3,0.5,0.4,0.
```




```
plot(ecdf(train$ResponseTime), xlim=range(c(train$ResponseTime, test$ResponseTime)), lty=2, col="dodger
plot(ecdf(test$ResponseTime), add=TRUE, lty=2, col="purple")
```



V. Logistic Regression Model for Accuracy

5.1 Fit the Training Data for the Full Model M1 With Interaction

```
acc_fit0 = glmer(Accuracy~StrengthLevel*ResponseTime+(1|ParticipantNum), data = train, family = binomial
summary(acc_fit0)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Accuracy ~ StrengthLevel * ResponseTime + (1 | ParticipantNum)
##    Data: train
##
##      AIC      BIC   logLik deviance df.resid
##  11109.2  11205.3  -5541.6  11083.2    11987
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -10.7654   0.1196   0.2824   0.6228   1.8051
##
## Random effects:
##  Groups         Name        Variance Std.Dev.
##  ParticipantNum (Intercept) 0.1294   0.3597
## Number of obs: 12000, groups:  ParticipantNum, 25
##
## Fixed effects:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                    0.650314   0.169945   3.827 0.000130 ***
## StrengthLevel4.5               0.432041   0.215104   2.009 0.044588 *
## StrengthLevel8                 1.167939   0.219178   5.329 9.89e-08 ***
```

```
## StrengthLevel12               2.053118   0.231709   8.861  < 2e-16 ***
## StrengthLevel25               3.449557   0.299961  11.500  < 2e-16 ***
## StrengthLevel40               4.501183   0.400932  11.227  < 2e-16 ***
## ResponseTime                 -0.513388   0.149877  -3.425 0.000614 ***
## StrengthLevel4.5:ResponseTime -0.006717   0.208660  -0.032 0.974321
## StrengthLevel8:ResponseTime   -0.425159   0.212446  -2.001 0.045365 *
## StrengthLevel12:ResponseTime  -0.811667   0.224944  -3.608 0.000308 ***
## StrengthLevel25:ResponseTime  -1.183803   0.294563  -4.019 5.85e-05 ***
## StrengthLevel40:ResponseTime  -1.508629   0.396081  -3.809 0.000140 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) StL4.5 StrnL8 StrL12 StrL25 StrL40 RspnsT SL4.5: SL8:RT
## StrngthL4.5 -0.612
## StrngthLvl8 -0.602  0.474
## StrngthLv12 -0.570  0.449  0.440
## StrngthLv25 -0.445  0.347  0.341  0.323
## StrngthLv40 -0.334  0.260  0.255  0.242  0.189
## ResponseTim -0.865  0.641  0.631  0.598  0.467  0.351
## StrnL4.5:RT  0.584 -0.952 -0.452 -0.428 -0.331 -0.249 -0.675
## StrngtL8:RT  0.575 -0.452 -0.951 -0.420 -0.325 -0.243 -0.665  0.475
## StrngL12:RT  0.543 -0.428 -0.420 -0.948 -0.308 -0.231 -0.627  0.450  0.442
## StrngL25:RT  0.418 -0.327 -0.321 -0.304 -0.944 -0.179 -0.483  0.344  0.338
## StrngL40:RT  0.312 -0.244 -0.239 -0.227 -0.178 -0.943 -0.360  0.257  0.252
##             SL12:R SL25:R
## StrngthL4.5
## StrngthLvl8
## StrngthLv12
## StrngthLv25
## StrngthLv40
## ResponseTim
## StrnL4.5:RT
## StrngtL8:RT
## StrngL12:RT
## StrngL25:RT  0.320
## StrngL40:RT  0.239  0.186
```

5.2 Fit the Training Data for the Additive Model M2 With Interaction

```r
acc_fit1 = glmer(Accuracy~StrengthLevel+ResponseTime+(1|ParticipantNum), data = train, family = binomial
summary(acc_fit1)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Accuracy ~ StrengthLevel + ResponseTime + (1 | ParticipantNum)
##    Data: train
##
##      AIC      BIC   logLik deviance df.resid
##  11136.9  11196.1  -5560.5  11120.9    11992
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -7.7752  0.1423  0.2829  0.6057  2.0782
##
## Random effects:
##  Groups         Name        Variance Std.Dev.
##  ParticipantNum (Intercept) 0.1363   0.3692
## Number of obs: 12000, groups:  ParticipantNum, 25
##
## Fixed effects:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)        1.03285    0.11848   8.718  < 2e-16 ***
## StrengthLevel4.5   0.42803    0.06598   6.487 8.74e-11 ***
## StrengthLevel8     0.75191    0.06801  11.055  < 2e-16 ***
## StrengthLevel12    1.26623    0.07368  17.185  < 2e-16 ***
## StrengthLevel25    2.33495    0.09824  23.767  < 2e-16 ***
## StrengthLevel40    3.11216    0.13204  23.571  < 2e-16 ***
## ResponseTime      -0.90327    0.08206 -11.008  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) StL4.5 StrnL8 StrL12 StrL25 StrL40
## StrngthL4.5 -0.261
## StrngthLvl8 -0.260  0.468
## StrngthLv12 -0.262  0.432  0.421
## StrngthLv25 -0.227  0.324  0.317  0.297
## StrngthLv40 -0.185  0.241  0.236  0.223  0.173
## ResponseTim -0.680 -0.011  0.000  0.032  0.069  0.075
```

5.3 Output the Table

```
stargazer(acc_fit0, acc_fit1, title='Logistic Regression Model',header = FALSE,label="tab:02005",ci=TRU
```

```
##
## \begin{table}[!htbp] \centering
##   \caption{Logistic Regression Model}
##   \label{tab:02005}
## \begin{tabular}{@{\extracolsep{5pt}}lcc}
## \\[-1.8ex]\hline
## \hline \\[-1.8ex]
##  & \multicolumn{2}{c}{\textit{Dependent variable:}} \\
## \cline{2-3}
## \\[-1.8ex] & \multicolumn{2}{c}{Accuracy} \\
## \\[-1.8ex] & (1) & (2)\\
## \hline \\[-1.8ex]
##  StrengthLevel4.5 & 0.432$^{**}$ & 0.428$^{***}$ \\
##   & (0.010, 0.854) & (0.299, 0.557) \\
##   & & \\
##  StrengthLevel8 & 1.168$^{***}$ & 0.752$^{***}$ \\
##   & (0.738, 1.598) & (0.619, 0.885) \\
##   & & \\
##  StrengthLevel12 & 2.053$^{***}$ & 1.266$^{***}$ \\
##   & (1.599, 2.507) & (1.122, 1.411) \\
##   & & \\
##  StrengthLevel25 & 3.450$^{***}$ & 2.335$^{***}$ \\
```

```
##   & (2.862, 4.037) & (2.142, 2.528) \\
##   & & \\
##  StrengthLevel40 & 4.501$^{***}$ & 3.112$^{***}$ \\
##   & (3.715, 5.287) & (2.853, 3.371) \\
##   & & \\
##  ResponseTime & $-$0.513$^{***}$ & $-$0.903$^{***}$ \\
##   & ($-$0.807, $-$0.220) & ($-$1.064, $-$0.742) \\
##   & & \\
##  StrengthLevel4.5:ResponseTime & $-$0.007 &  \\
##   & ($-$0.416, 0.402) &  \\
##   & & \\
##  StrengthLevel8:ResponseTime & $-$0.425$^{**}$ &  \\
##   & ($-$0.842, $-$0.009) &  \\
##   & & \\
##  StrengthLevel12:ResponseTime & $-$0.812$^{***}$ &  \\
##   & ($-$1.253, $-$0.371) &  \\
##   & & \\
##  StrengthLevel25:ResponseTime & $-$1.184$^{***}$ &  \\
##   & ($-$1.761, $-$0.606) &  \\
##   & & \\
##  StrengthLevel40:ResponseTime & $-$1.509$^{***}$ &  \\
##   & ($-$2.285, $-$0.732) &  \\
##   & & \\
##  Constant & 0.650$^{***}$ & 1.033$^{***}$ \\
##   & (0.317, 0.983) & (0.801, 1.265) \\
##   & & \\
## \hline \\[-1.8ex]
## Observations & 12,000 & 12,000 \\
## Log Likelihood & $-$5,541.612 & $-$5,560.464 \\
## Akaike Inf. Crit. & 11,109.220 & 11,136.930 \\
## Bayesian Inf. Crit. & 11,205.330 & 11,196.070 \\
## \hline
## \hline \\[-1.8ex]
## \textit{Note:}  & \multicolumn{2}{r}{$^{*}$p$<$0.1; $^{**}$p$<$0.05; $^{***}$p$<$0.01} \\
## \end{tabular}
## \end{table}
```

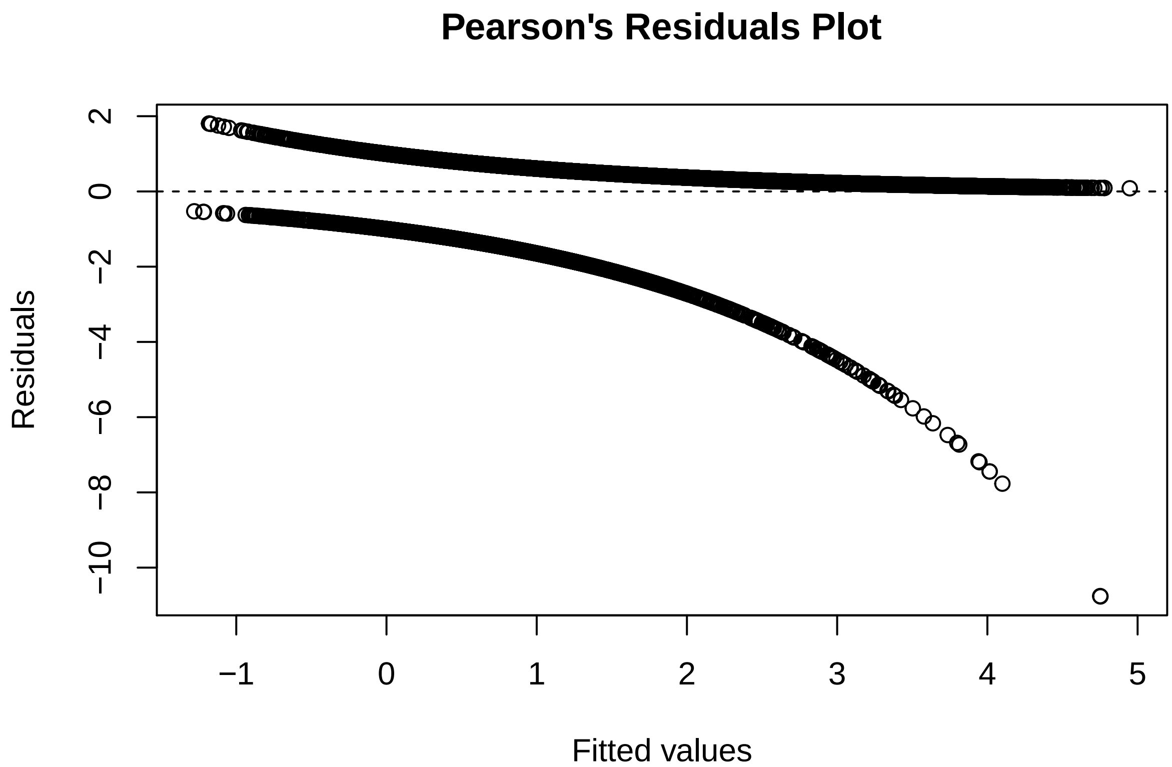
5.4 Likelihood Ratio Tests (Goodness-of-fit)

```r
lrtest(acc_fit0, acc_fit1)
```

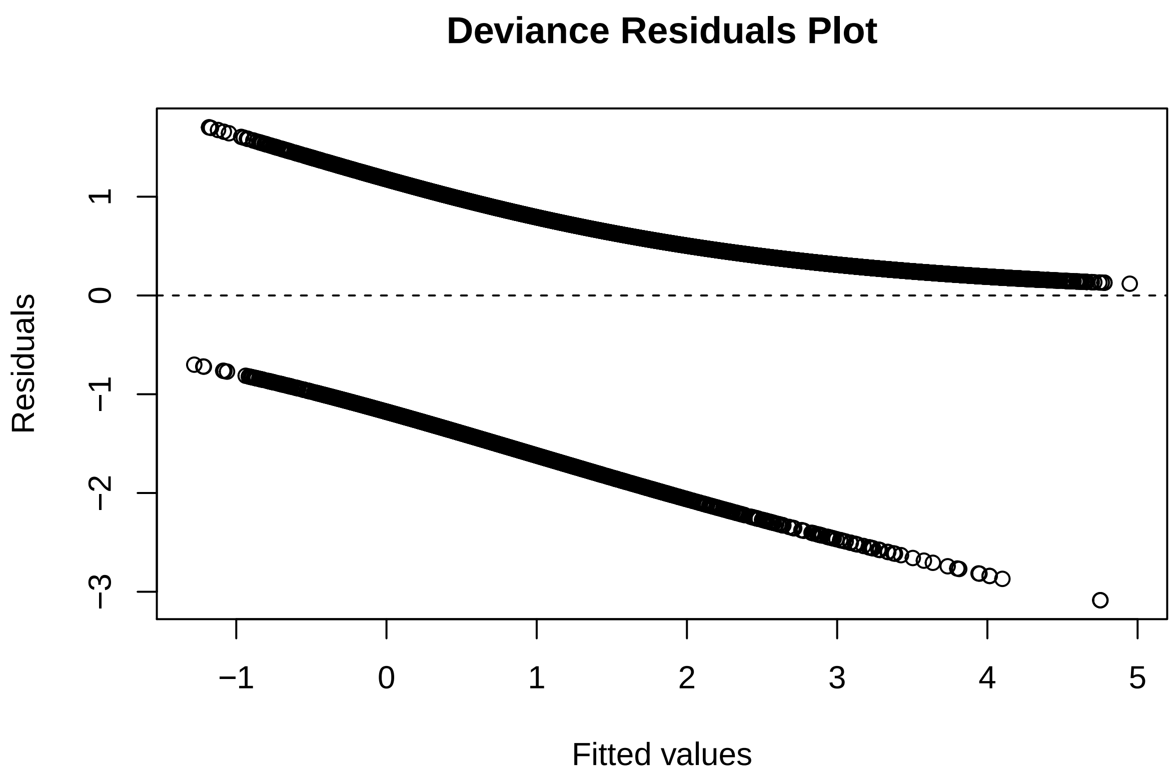
```
## Likelihood ratio test
##
## Model 1: Accuracy ~ StrengthLevel * ResponseTime + (1 | ParticipantNum)
## Model 2: Accuracy ~ StrengthLevel + ResponseTime + (1 | ParticipantNum)
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1  13 -5541.6
## 2   8 -5560.5 -5 37.704  4.326e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5.5 Diagostics

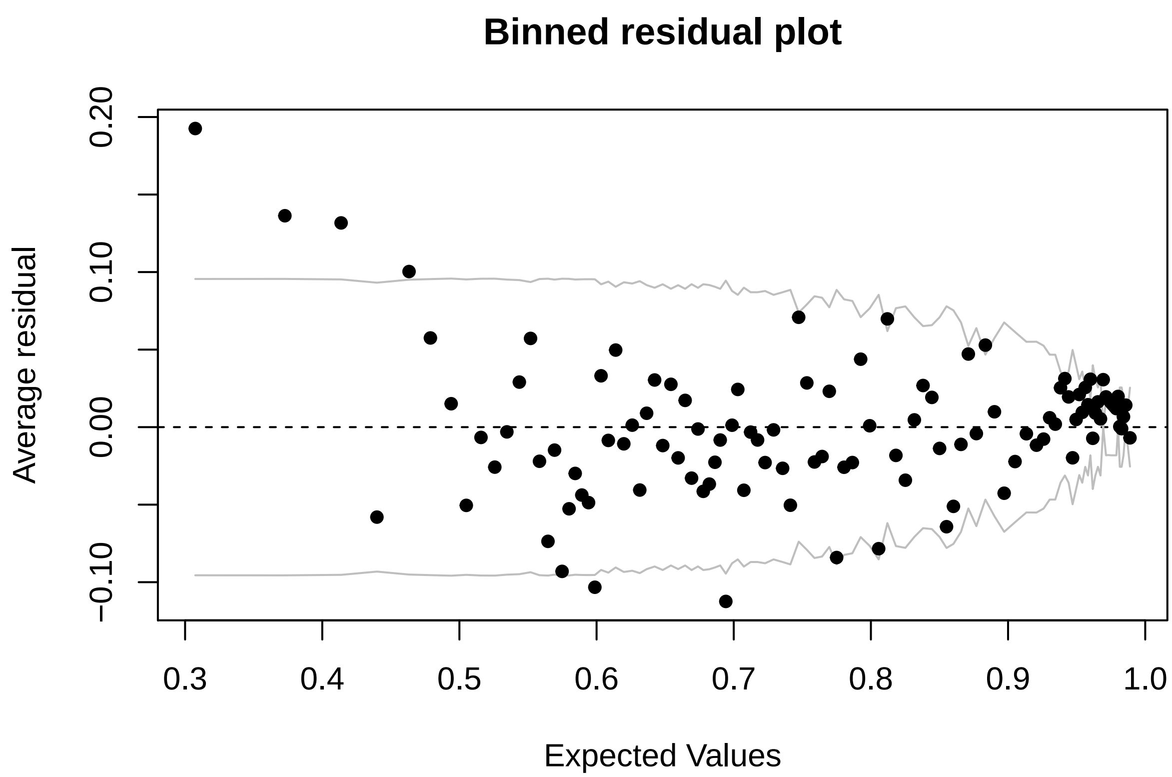
```
# Pearson's residuals plot
res_pear = residuals(acc_fit0, type="pearson")
plot(predict(acc_fit0), res_pear, xlab="Fitted values", ylab = "Residuals", main="Pearson's Residuals Pl
abline(h = 0, lty = 2)
```




```
# Deviance residuals plot
res_dev = residuals(acc_fit0, type="deviance")
plot(predict(acc_fit0), res_dev, xlab="Fitted values", ylab = "Residuals", main="Deviance Residuals Plot
abline(h = 0, lty = 2)
```




```r
# Binned residuals plot
binnedplot(fitted(acc_fit0), residuals(acc_fit0, type = "response"))
```



5.6 Predictions

```r
prob <- predict(acc_fit0, newdata=test, type="response")
pred <- prediction(prob, test$Accuracy)
Behave_fit_pred = rep(0, dim(test)[1])
Behave_fit_pred[prob > 0.5] = 1

test_error = mean(Behave_fit_pred != test$Accuracy)
test_error
```

Test Error

```
## [1] 0.2303684
```

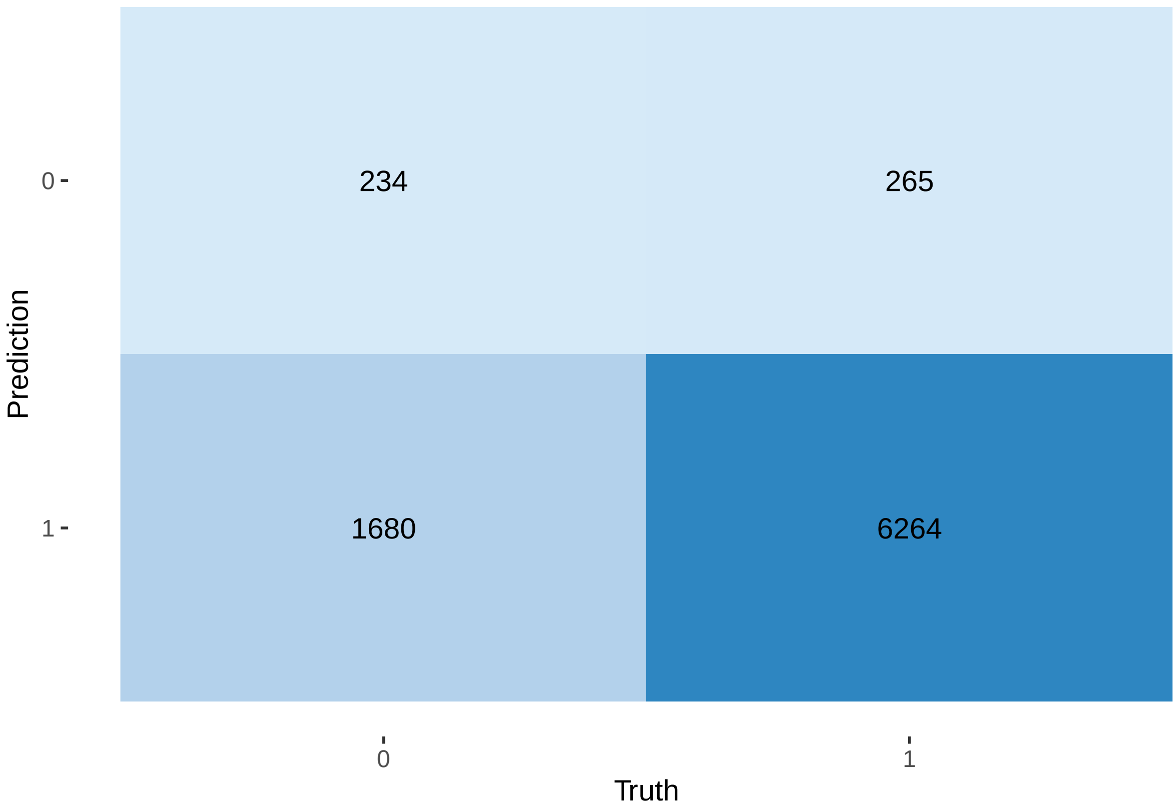
```r
confusion_matrix_acc = table(Behave_fit_pred, test$Accuracy)
confusion_matrix_acc
```

Confusion Matrix

```
##
## Behave_fit_pred    0    1
##               0  234  265
##               1 1680 6264
```

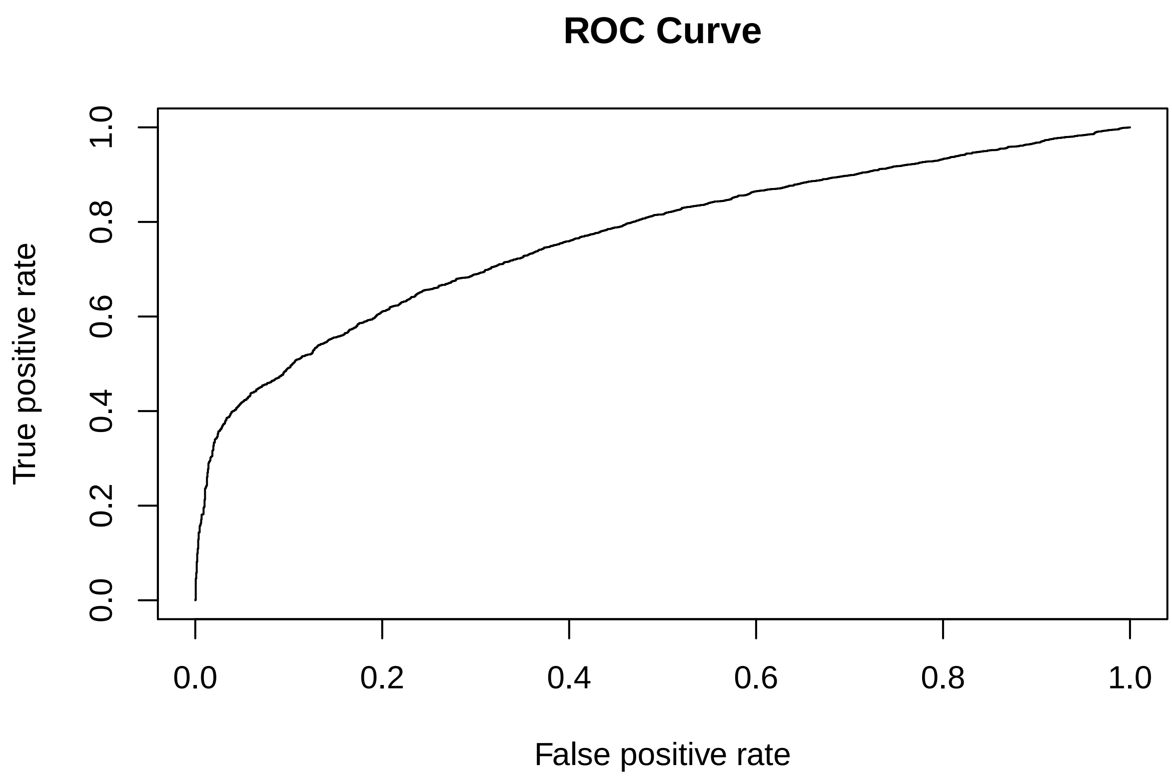
```r
df_cm = data.frame(test$Accuracy, Behave_fit_pred)
df_cm$obs = as.factor(df_cm$test.Accuracy)
df_cm$pred = as.factor(df_cm$Behave_fit_pred)
cm = conf_mat(df_cm, obs, pred)
autoplot(cm, type = "heatmap") +
  scale_fill_gradient(low="#D6EAF8",high = "#2E86C1")
```

```
## Scale for 'fill' is already present. Adding another scale for 'fill', which
## will replace the existing scale.
```



ROC curve and AUROC \begin{figure}[p]

```
perf <- performance(pred, measure = "tpr", x.measure = "fpr")
plot(perf, main = "ROC Curve")
```



\end{figure}

```
auc <- performance(pred, measure = "auc")
auc <- auc@y.values[[1]]
auc
```

```
## [1] 0.7680455
```

```
234/(234+1680)   # Specificity
```

Sensitivity and Specificity

```
## [1] 0.1222571
```

```
6264/(6264+265) # Sensitivity
```

```
## [1] 0.9594119
```

VI. Logistic Regression Model for Confidence

6.1 Map the Confidence into 2 Categories

```
behave_train <- train %>%
  mutate(Conf = ifelse(as.numeric(Confidence) > 2, 1, 0))
behave_test <- test %>%
  mutate(Conf = ifelse(as.numeric(Confidence) > 2, 1, 0))
```

6.2 Fit the Training Data for the Full Model M1 With Interaction

```
conf_fit0 = glmer(Conf~StrengthLevel*ResponseTime+(1|ParticipantNum), data = behave_train, family = bin
summary(conf_fit0)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Conf ~ StrengthLevel * ResponseTime + (1 | ParticipantNum)
##    Data: behave_train
##
##      AIC      BIC   logLik deviance df.resid
##  11164.4  11260.5  -5569.2  11138.4    11987
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -7.3746 -0.5349  0.1490  0.5319 10.5827
##
## Random effects:
##  Groups         Name        Variance Std.Dev.
##  ParticipantNum (Intercept) 2.316    1.522
## Number of obs: 12000, groups:  ParticipantNum, 25
##
## Fixed effects:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                    1.3973     0.3690   3.786 0.000153 ***
## StrengthLevel4.5               0.4355     0.2901   1.501 0.133384
## StrengthLevel8                 0.7143     0.2842   2.514 0.011941 *
## StrengthLevel12                0.8978     0.2756   3.258 0.001121 **
## StrengthLevel25                1.5253     0.2785   5.478 4.31e-08 ***
## StrengthLevel40                1.6956     0.2916   5.815 6.05e-09 ***
## ResponseTime                  -2.3434     0.2087 -11.226  < 2e-16 ***
## StrengthLevel4.5:ResponseTime  -0.1759     0.2897  -0.607 0.543702
## StrengthLevel8:ResponseTime    -0.3249     0.2862  -1.135 0.256358
## StrengthLevel12:ResponseTime   -0.1027     0.2792  -0.368 0.712971
## StrengthLevel25:ResponseTime    0.3862     0.2879   1.341 0.179783
## StrengthLevel40:ResponseTime    0.9251     0.3113   2.972 0.002959 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) StL4.5 StrnL8 StrL12 StrL25 StrL40 RspnsT SL4.5: SL8:RT
## StrngthL4.5 -0.397
## StrngthLvl8 -0.404  0.519
## StrngthLv12 -0.417  0.536  0.548
## StrngthLv25 -0.412  0.532  0.544  0.566
## StrngthLv40 -0.392  0.509  0.521  0.543  0.558
## ResponseTim -0.543  0.672  0.683  0.704  0.693  0.658
## StrnL4.5:RT  0.381 -0.961 -0.497 -0.514 -0.510 -0.488 -0.700
## StrngtL8:RT  0.385 -0.493 -0.959 -0.521 -0.517 -0.495 -0.705  0.513
## StrngL12:RT  0.395 -0.506 -0.517 -0.956 -0.534 -0.513 -0.724  0.527  0.533
## StrngL25:RT  0.383 -0.492 -0.502 -0.523 -0.952 -0.517 -0.701  0.512  0.518
## StrngL40:RT  0.352 -0.456 -0.466 -0.486 -0.501 -0.948 -0.644  0.475  0.481
##             SL12:R SL25:R
## StrngthL4.5
## StrngthLvl8
## StrngthLv12
## StrngthLv25
## StrngthLv40
```

```
## ResponseTim
## StrnL4.5:RT
## StrngtL8:RT
## StrngL12:RT
## StrngL25:RT  0.537
## StrngL40:RT  0.499  0.507
```

6.3 Fit the Training Data for the Additive Model M2 With Interaction

```r
conf_fit1 = glmer(Conf~StrengthLevel+ResponseTime+(1|ParticipantNum), data = behave_train, family = bino
summary(conf_fit1)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Conf ~ StrengthLevel + ResponseTime + (1 | ParticipantNum)
##    Data: behave_train
##
##      AIC      BIC   logLik deviance df.resid
##  11176.6  11235.8  -5580.3  11160.6    11992
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -7.9679 -0.5390  0.1402  0.5318 10.8764
##
## Random effects:
##  Groups         Name        Variance Std.Dev.
##  ParticipantNum (Intercept) 2.348    1.532
## Number of obs: 12000, groups:  ParticipantNum, 25
##
## Fixed effects:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)       1.34212    0.32393   4.143 3.42e-05 ***
## StrengthLevel4.5  0.26548    0.08040   3.302  0.00096 ***
## StrengthLevel8    0.40814    0.08002   5.101 3.39e-07 ***
## StrengthLevel12   0.80425    0.08023  10.025  < 2e-16 ***
## StrengthLevel25   1.86960    0.08485  22.034  < 2e-16 ***
## StrengthLevel40   2.47980    0.09084  27.299  < 2e-16 ***
## ResponseTime     -2.29075    0.09306 -24.616  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) StL4.5 StrnL8 StrL12 StrL25 StrL40
## StrngthL4.5 -0.125
## StrngthLvl8 -0.131  0.513
## StrngthLv12 -0.136  0.514  0.520
## StrngthLv25 -0.136  0.490  0.499  0.514
## StrngthLv40 -0.134  0.459  0.468  0.486  0.501
## ResponseTim -0.270 -0.008  0.011  0.024  0.039  0.053
```

6.4 Output the Table

```
stargazer(conf_fit0, conf_fit1, title='Logistic Regression Model',header = FALSE,label="tab:02005",ci=T
```

```
##
## \begin{table}[!htbp] \centering
##   \caption{Logistic Regression Model}
##   \label{tab:02005}
## \begin{tabular}{@{\extracolsep{5pt}}lcc}
## \\[-1.8ex]\hline
## \hline \\[-1.8ex]
##  & \multicolumn{2}{c}{\textit{Dependent variable:}} \\
## \cline{2-3}
## \\[-1.8ex] & \multicolumn{2}{c}{Conf} \\
## \\[-1.8ex] & (1) & (2)\\
## \hline \\[-1.8ex]
##  StrengthLevel4.5 & 0.435 & 0.265$^{***}$ \\
##   & ($-$0.133, 1.004) & (0.108, 0.423) \\
##   & & \\
##  StrengthLevel8 & 0.714$^{**}$ & 0.408$^{***}$ \\
##   & (0.157, 1.271) & (0.251, 0.565) \\
##   & & \\
##  StrengthLevel12 & 0.898$^{***}$ & 0.804$^{***}$ \\
##   & (0.358, 1.438) & (0.647, 0.961) \\
##   & & \\
##  StrengthLevel25 & 1.525$^{***}$ & 1.870$^{***}$ \\
##   & (0.980, 2.071) & (1.703, 2.036) \\
##   & & \\
##  StrengthLevel40 & 1.696$^{***}$ & 2.480$^{***}$ \\
##   & (1.124, 2.267) & (2.302, 2.658) \\
##   & & \\
##  ResponseTime & $-$2.343$^{***}$ & $-$2.291$^{***}$ \\
##   & ($-$2.753, $-$1.934) & ($-$2.473, $-$2.108) \\
##   & & \\
##  StrengthLevel4.5:ResponseTime & $-$0.176 &  \\
##   & ($-$0.744, 0.392) &  \\
##   & & \\
##  StrengthLevel8:ResponseTime & $-$0.325 &  \\
##   & ($-$0.886, 0.236) &  \\
##   & & \\
##  StrengthLevel12:ResponseTime & $-$0.103 &  \\
##   & ($-$0.650, 0.444) &  \\
##   & & \\
##  StrengthLevel25:ResponseTime & 0.386 &  \\
##   & ($-$0.178, 0.951) &  \\
##   & & \\
##  StrengthLevel40:ResponseTime & 0.925$^{***}$ &  \\
##   & (0.315, 1.535) &  \\
##   & & \\
##  Constant & 1.397$^{***}$ & 1.342$^{***}$ \\
##   & (0.674, 2.121) & (0.707, 1.977) \\
##   & & \\
## \hline \\[-1.8ex]
## Observations & 12,000 & 12,000 \\
```

```
## Log Likelihood & $-$5,569.218 & $-$5,580.312 \\
## Akaike Inf. Crit. & 11,164.440 & 11,176.620 \\
## Bayesian Inf. Crit. & 11,260.540 & 11,235.760 \\
## \hline
## \hline \\[-1.8ex]
## \textit{Note:}  & \multicolumn{2}{r}{$^{*}$p$<$0.1; $^{**}$p$<$0.05; $^{***}$p$<$0.01} \\
## \end{tabular}
## \end{table}
```

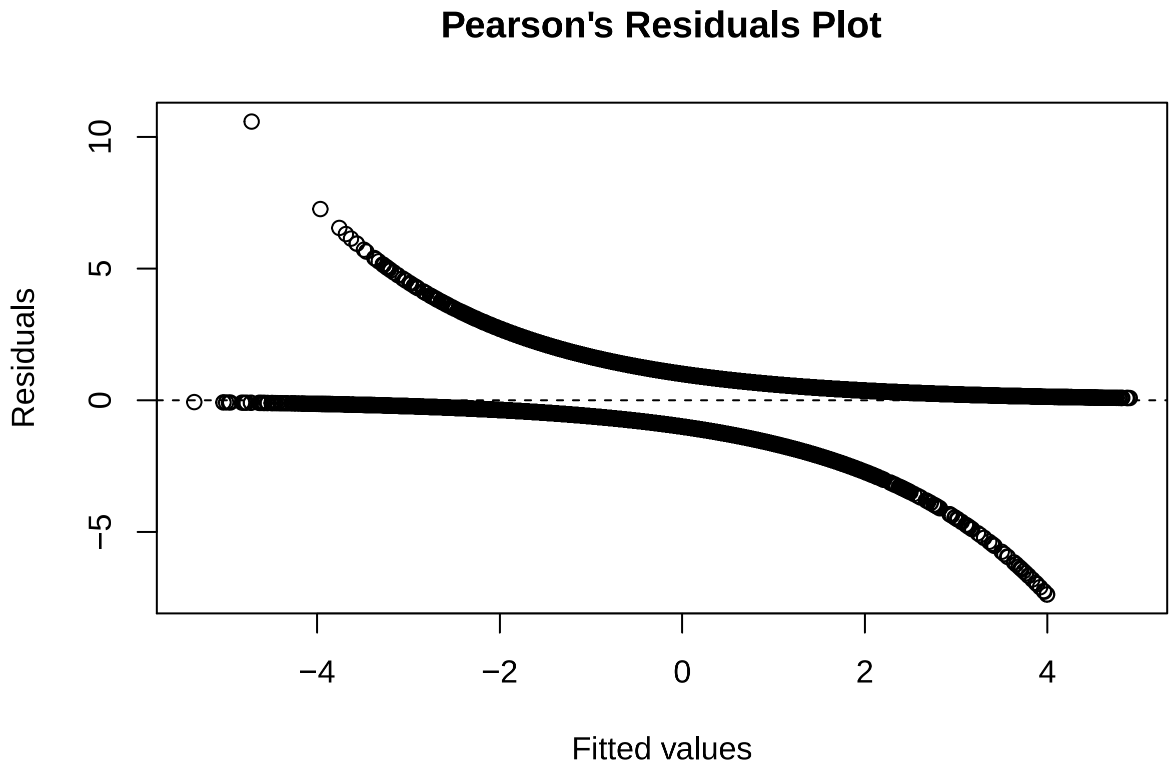
6.5 Likelihood Ratio Tests (Goodness-of-fit)

```r
lrtest(conf_fit1, conf_fit0)
```

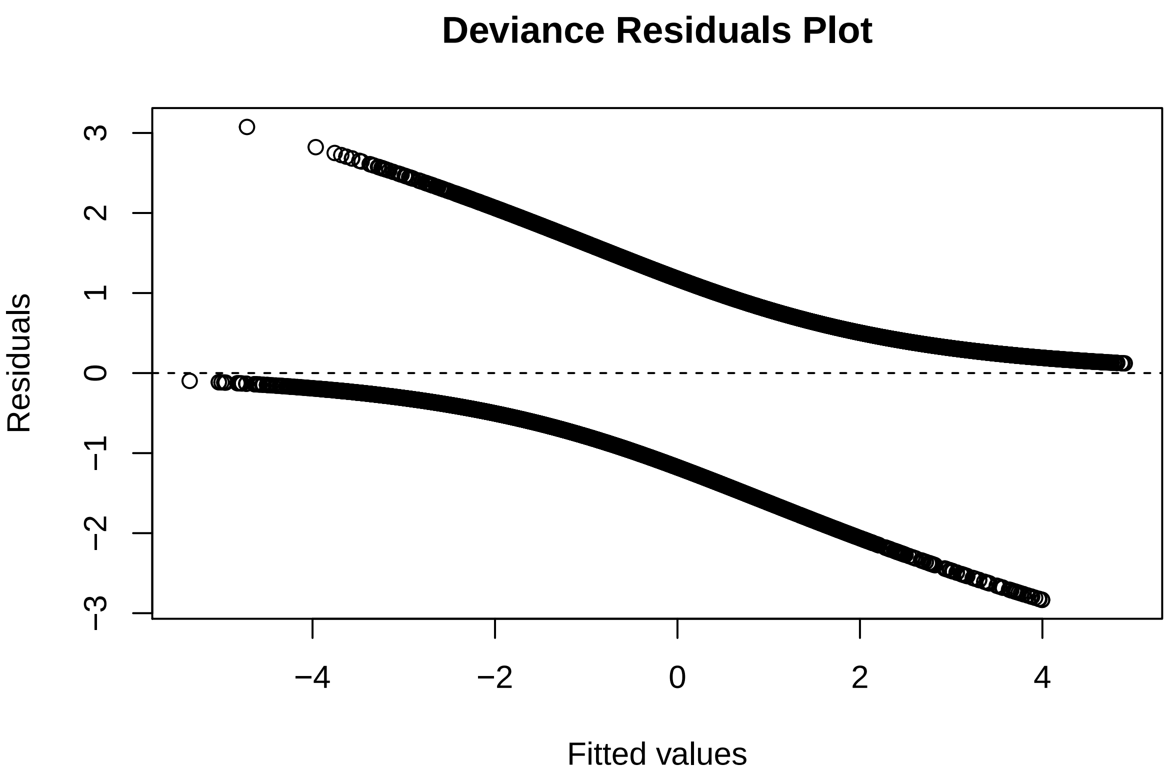
```
## Likelihood ratio test
##
## Model 1: Conf ~ StrengthLevel + ResponseTime + (1 | ParticipantNum)
## Model 2: Conf ~ StrengthLevel * ResponseTime + (1 | ParticipantNum)
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   8 -5580.3
## 2  13 -5569.2  5 22.188  0.0004822 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

6.6 Diagostics

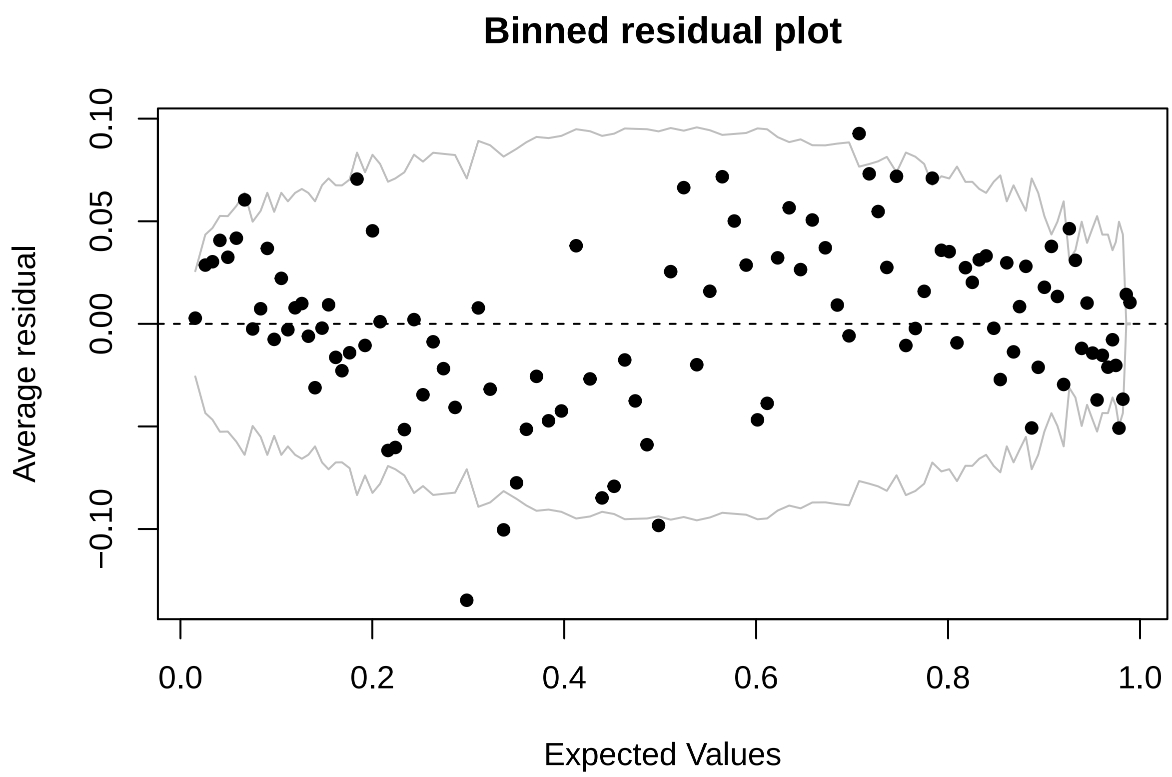
```r
# Pearson's residuals plot
res_pear = residuals(conf_fit0, type="pearson")
plot(predict(conf_fit0), res_pear, xlab="Fitted values", ylab = "Residuals", main="Pearson's Residuals P
abline(h = 0, lty = 2)
```



```r
# Deviance residuals plot
res_dev = residuals(conf_fit0, type="deviance")
plot(predict(conf_fit0), res_dev, xlab="Fitted values", ylab = "Residuals", main="Deviance Residuals Plo
abline(h = 0, lty = 2)
```




```r
# Binned residuals plot
binnedplot(fitted(conf_fit0), residuals(conf_fit0, type = "response"))
```



6.7 Predictions

```
prob <- predict(conf_fit0, newdata=behave_test, type="response")
pred <- prediction(prob, behave_test$Conf)
Behave_fit_pred = rep(0, dim(behave_test)[1])
Behave_fit_pred[prob > 0.5] = 1

test_error = mean(Behave_fit_pred != behave_test$Conf)
test_error
```

Test Error

```
## [1] 0.2114177
```

```
confusion_matrix_conf = table(Behave_fit_pred, behave_test$Conf)
confusion_matrix_conf
```

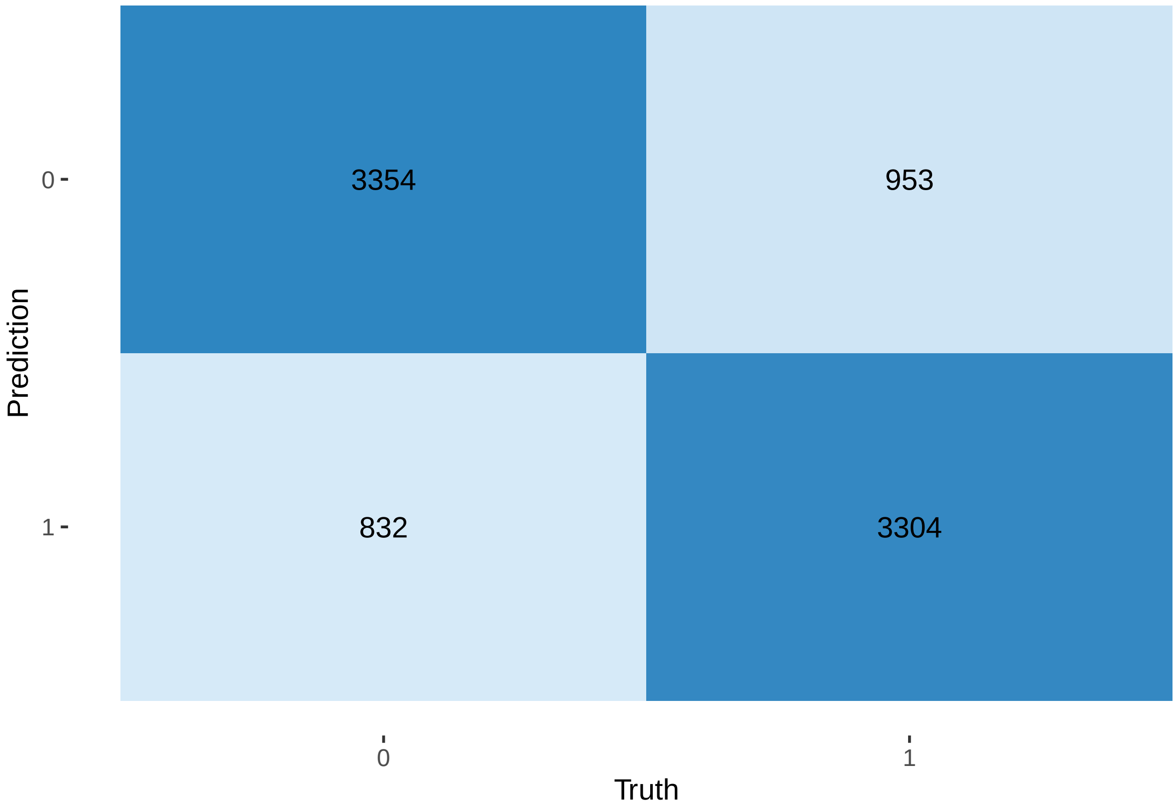
Confusion Matrix

```
##
## Behave_fit_pred    0    1
##               0 3354  953
##               1  832 3304
```

```
df_cm = data.frame(behave_test$Conf, Behave_fit_pred)
df_cm$obs = as.factor(df_cm$behave_test.Conf)
df_cm$pred = as.factor(df_cm$Behave_fit_pred)
cm = conf_mat(df_cm, obs, pred)
autoplot(cm, type = "heatmap") +
```

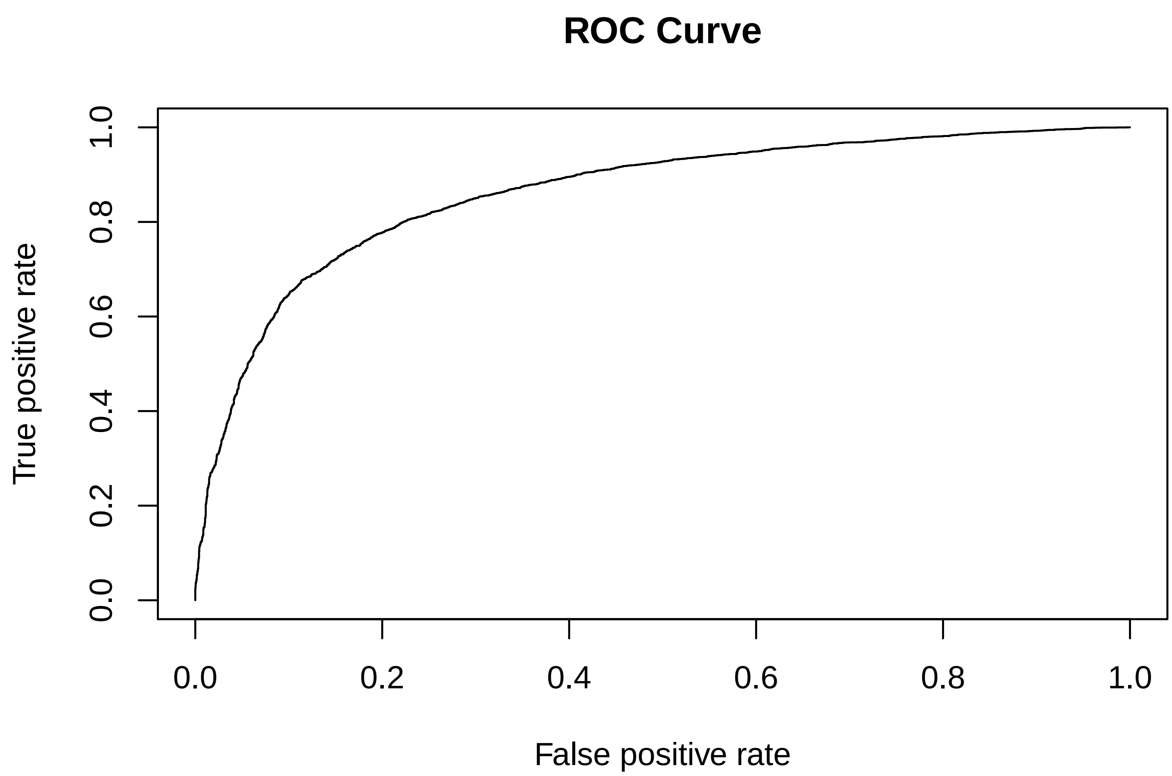
```r
  scale_fill_gradient(low="#D6EAF8",high = "#2E86C1")
```

```
## Scale for 'fill' is already present. Adding another scale for 'fill', which
## will replace the existing scale.
```



ROC curve and AUROC \begin{figure}[p]

```r
perf <- performance(pred, measure = "tpr", x.measure = "fpr")
plot(perf, main = "ROC Curve")
```



\end{figure}

```
auc <- performance(pred, measure = "auc")
auc <- auc@y.values[[1]]
auc
```

```
## [1] 0.8611944
```

```

3354/(3354+832) # specificity
```

Sensitivity and Specificity

```
## [1] 0.8012422
```

```
3304/(3304+953) # sensitivity
```

```
## [1] 0.7761334
```