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
```

I. Data Description

1.1 Import data

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

1.2 Summarization

```r
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
```

```r
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 ...
```

```r
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
```

```r
# 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
```

```r
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
```

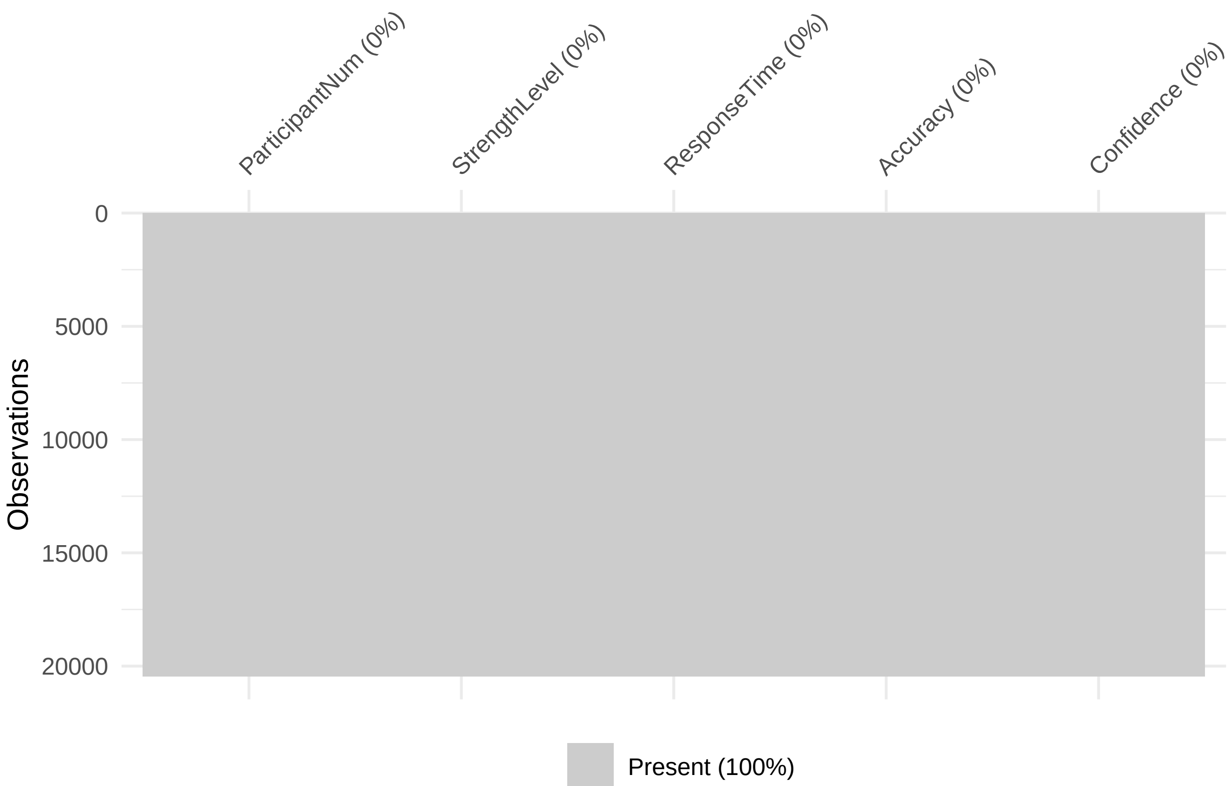
1.3 Factorization

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

II. Explotary Data Analysis

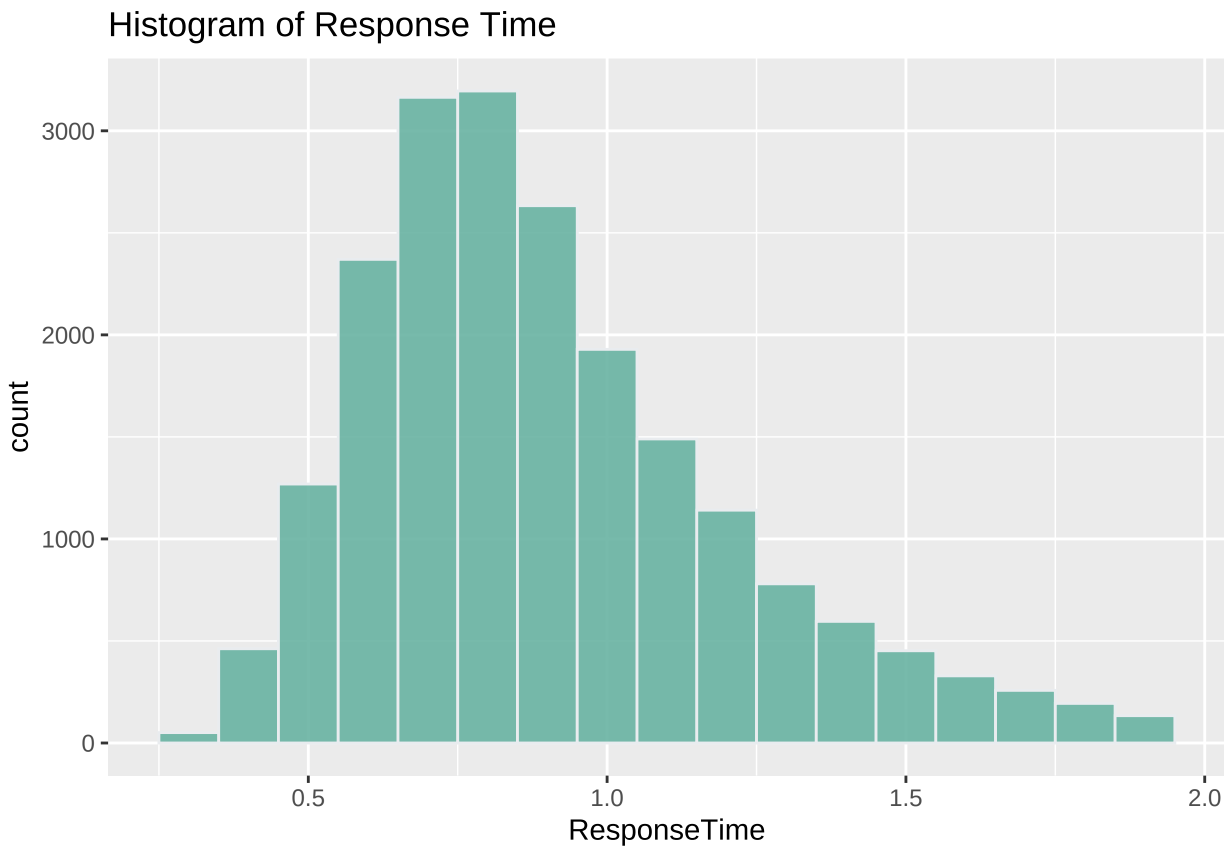
2.1 Missing Data

```r
vis_miss(Behave)
```



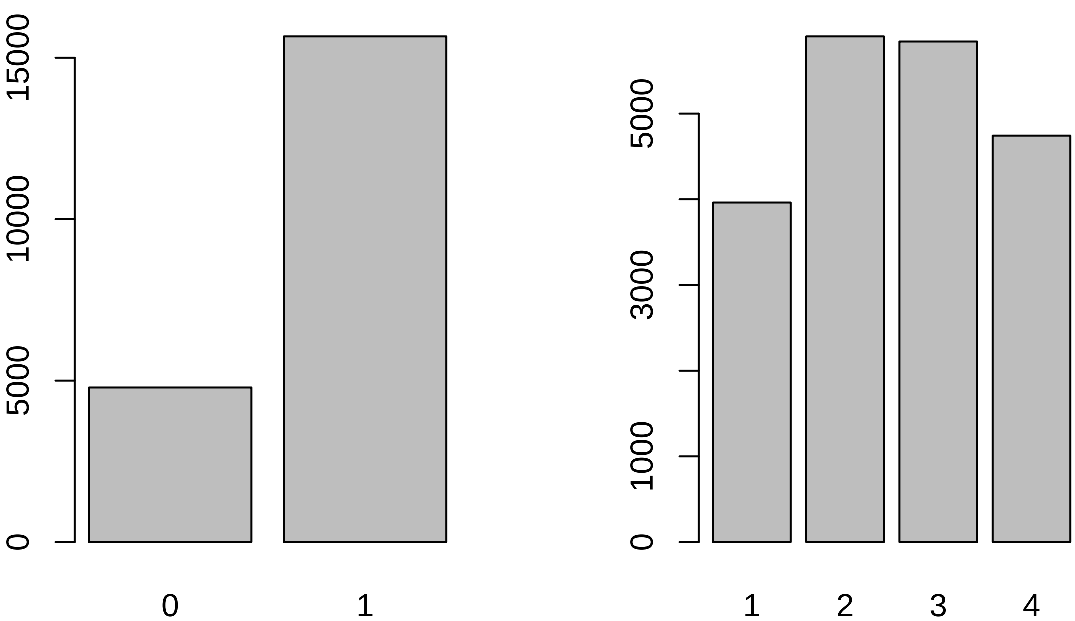
2.2 Potential Outliers for Response Time

```r
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

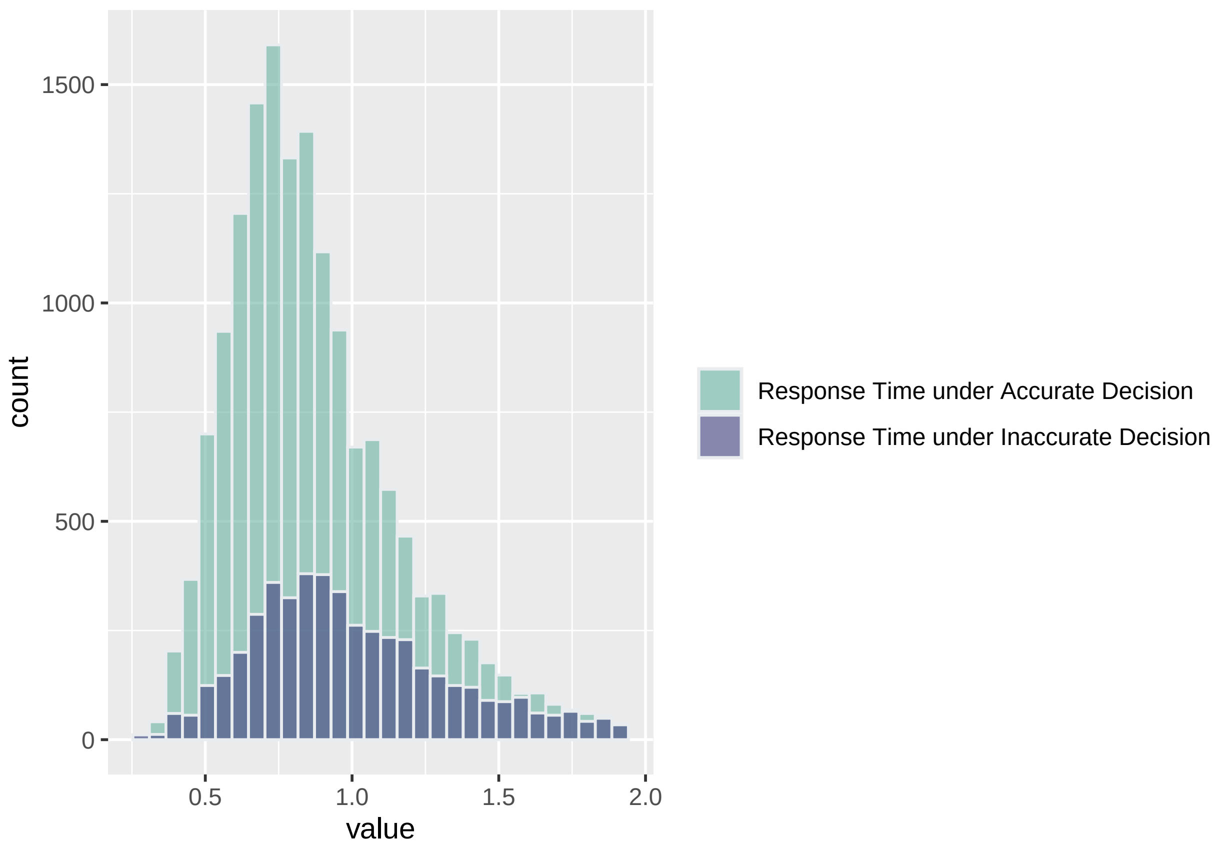
```
par(mfrow = c(1,2))
plot(as.factor(Behave$Accuracy))
plot(as.factor(Behave$Confidence))
```



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

```r
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

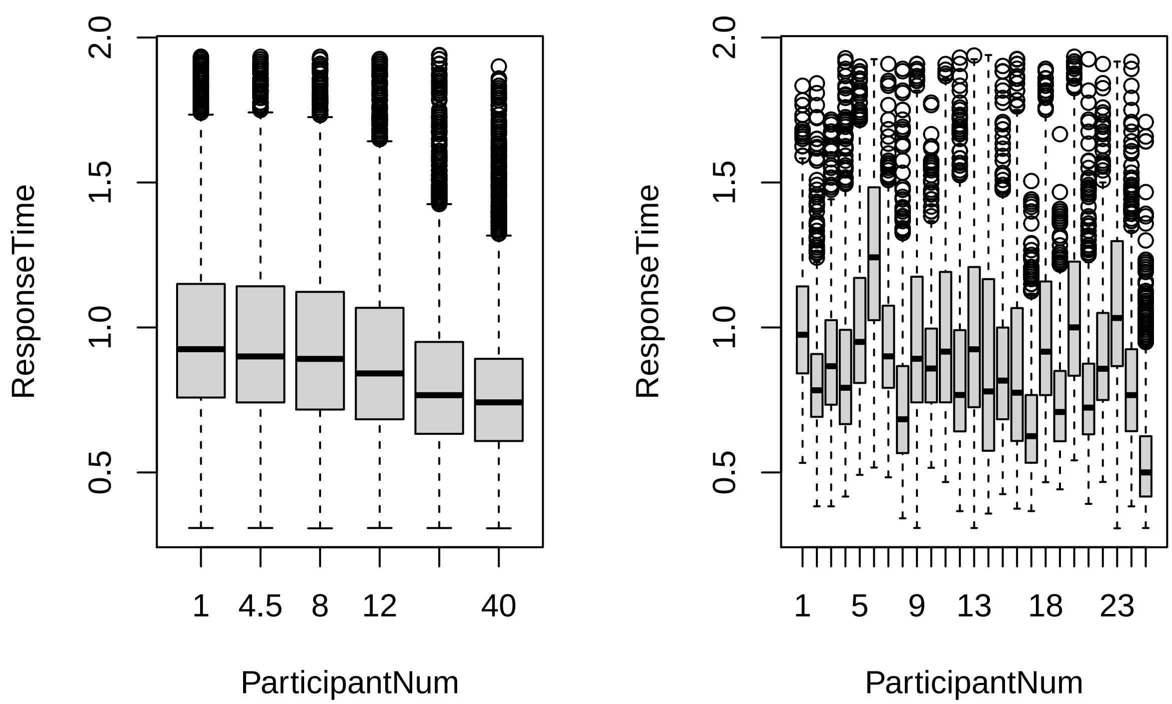
```r
table(Behave$StrengthLevel, Behave$ParticipantNum)
```

```
##
##         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

```r
par(mfrow = c(1,2))
boxplot(ResponseTime~StrengthLevel, data = Behave, xlab="ParticipantNum", ylab="ResponseTime")
boxplot(ResponseTime~ParticipantNum, data = Behave, xlab="ParticipantNum", ylab="ResponseTime")
```



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

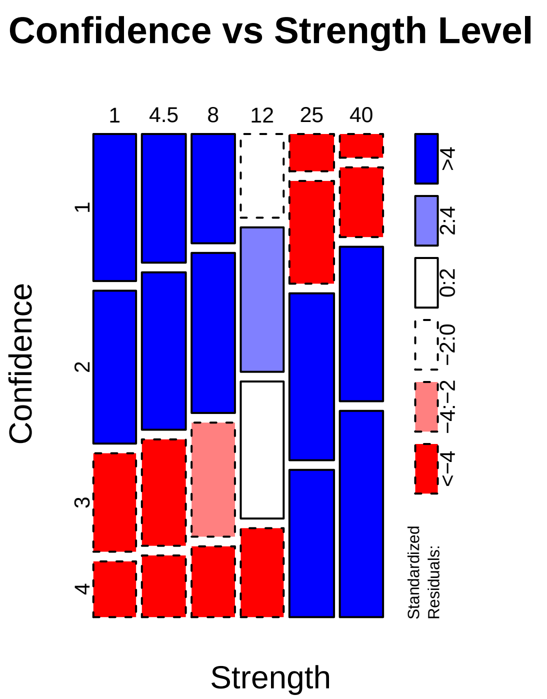
```r
par(mfrow = c(1,2))
tb_Str_Conf = table(Behave$StrengthLevel, Behave$Confidence); tb_Str_Conf
```

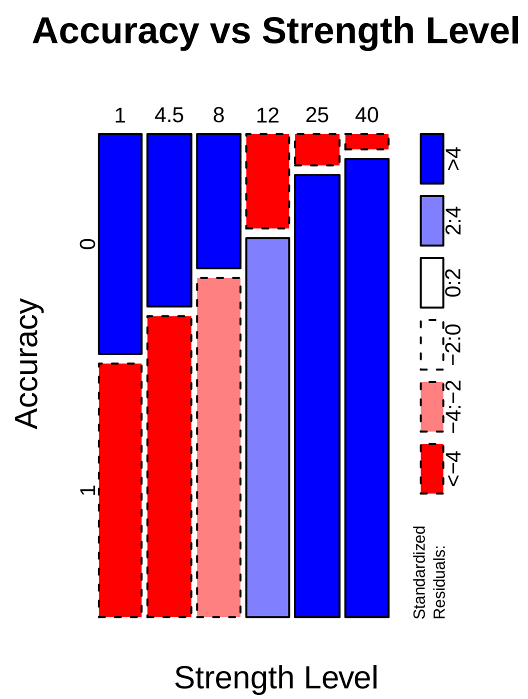
```
##
##           1    2    3    4
##   1    1084 1127  726  411
##   4.5   976 1194  807  468
##   8     818 1198  854  530
##   12    620 1070 1015  659
##   25    287  791 1285 1134
##   40    177  521 1154 1541
```

```r
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
```

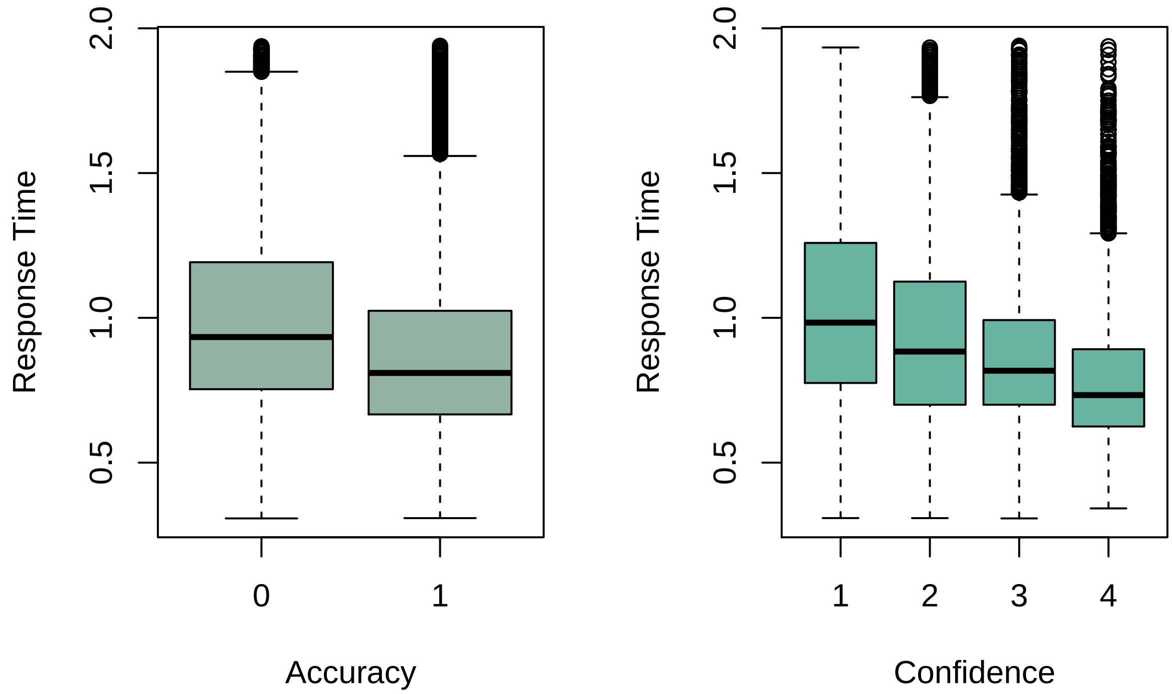
```r
mosaicplot(tb_Stren_Acc, shade = TRUE, main="Accuracy vs Strength Level", xlab="Strength Level", ylab="/
```





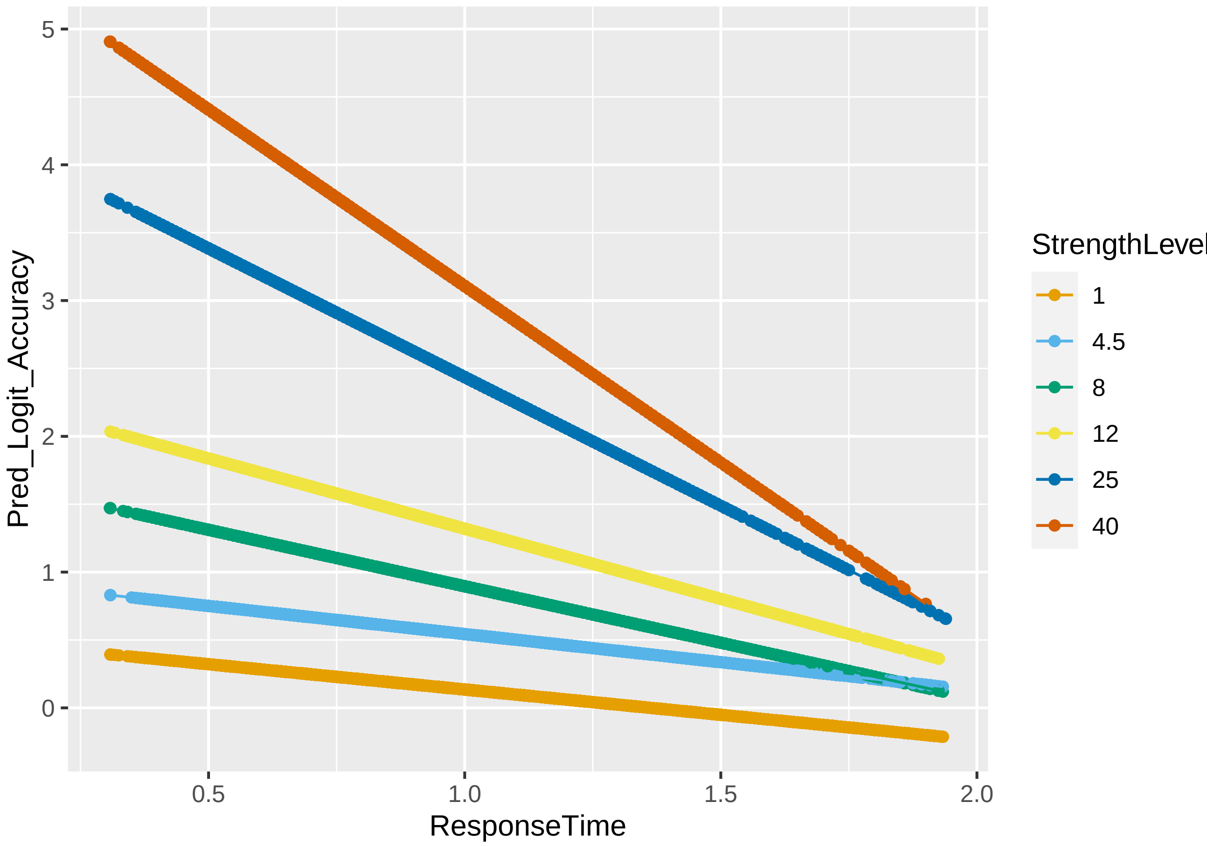
2.8 Box Plots of Reponse time Vs Accuracy and Reponse time Vs Confidence

```r
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
```



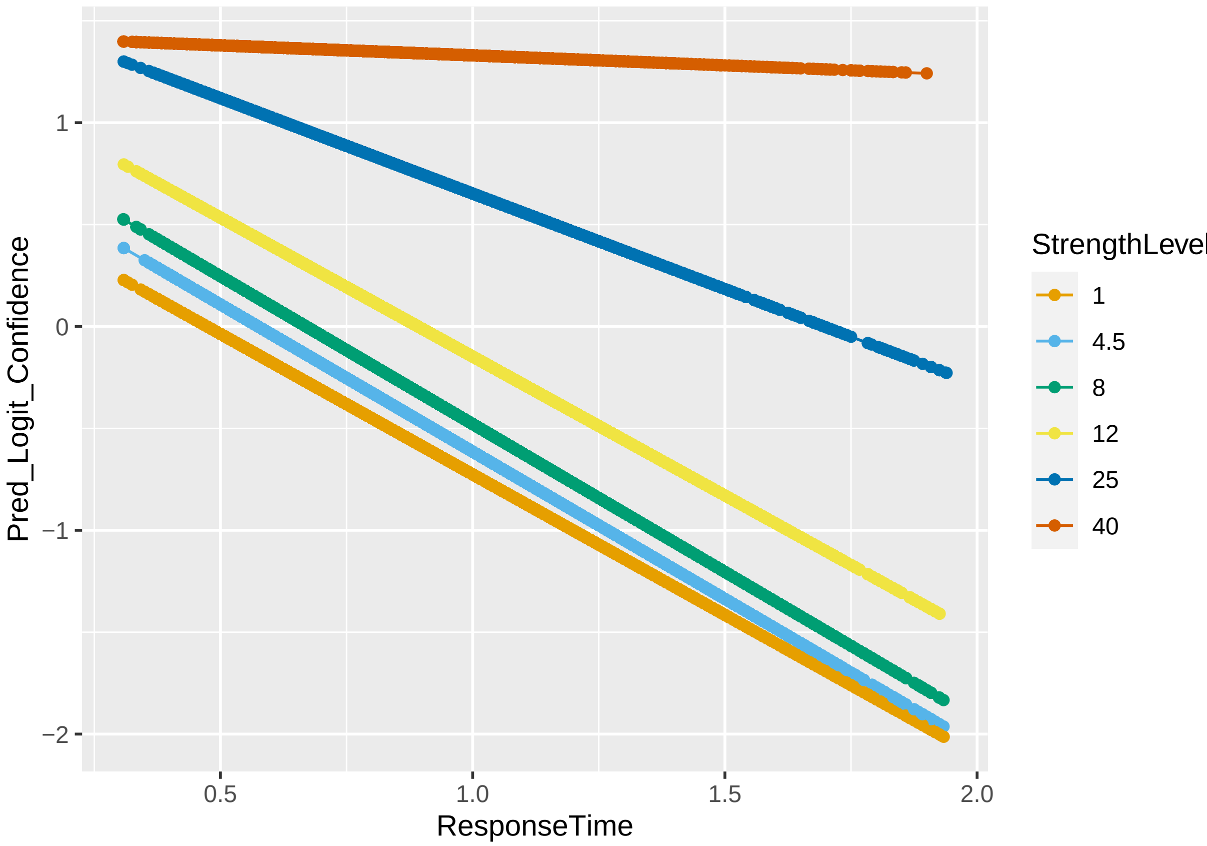
2.9 Interactions between Strength Level and Response Time

```
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

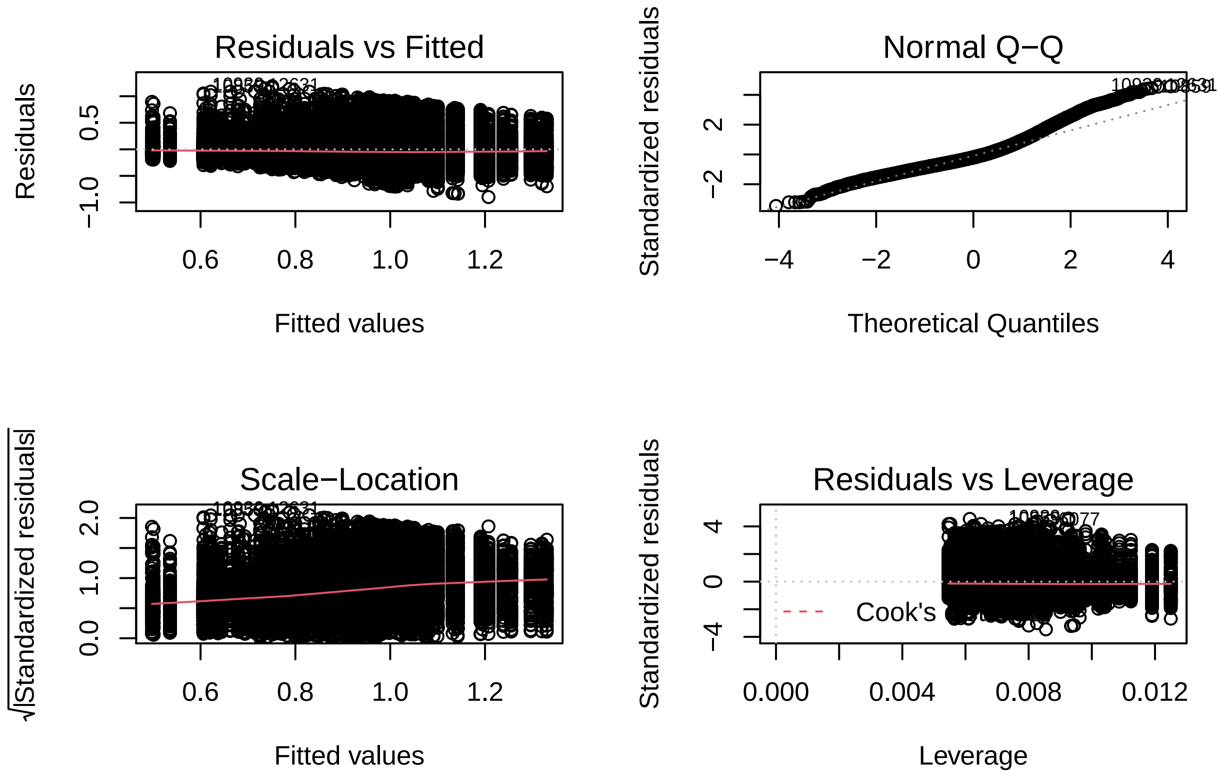
3.1 Build ANOVA Model with Interaction

```
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  311.63 <2e-16 ***
## ParticipantNum                  24  396.5  16.520  242.31 <2e-16 ***
## StrengthLevel:ParticipantNum   120   34.4   0.287    4.21 <2e-16 ***
## Residuals                    20297 1383.8   0.068
## ---
## 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  26.681 360.628 <2e-16 ***
## ParticipantNum                  24  566.1  23.586 318.796 <2e-16 ***
## StrengthLevel:ParticipantNum   120   39.6   0.330   4.457 <2e-16 ***
## Residuals                    20297 1501.7   0.074
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

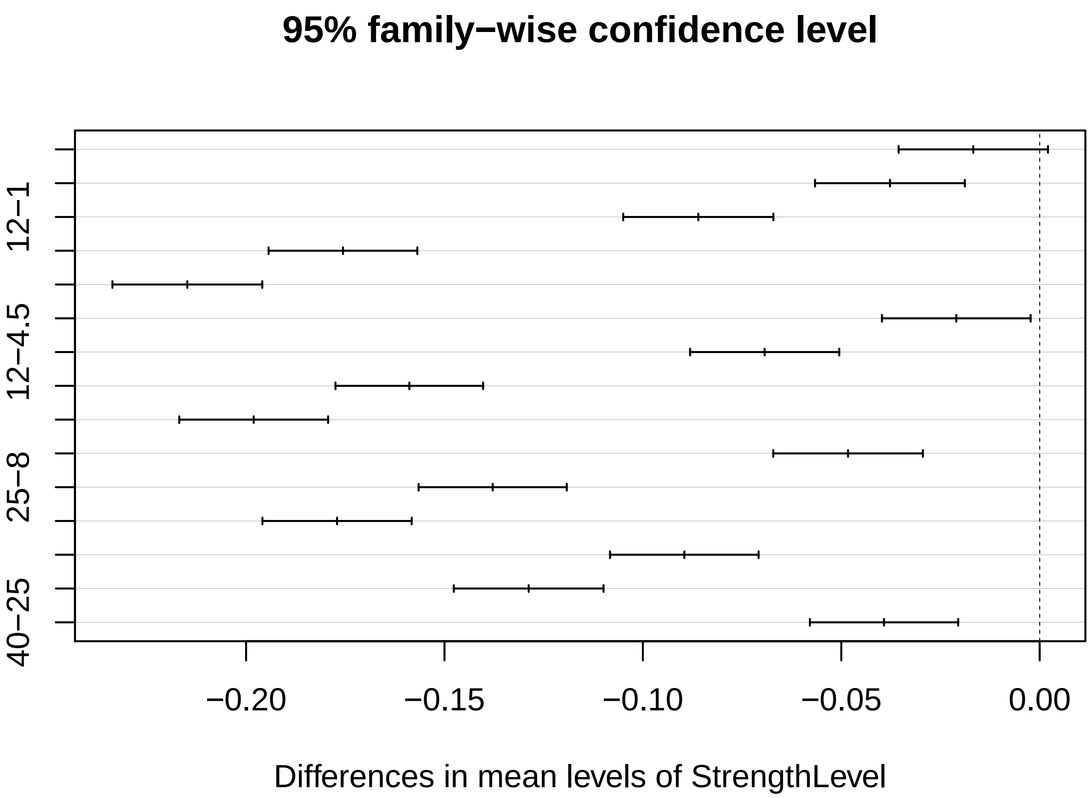
3.4 Tukey Method for Pairwise Comparisons

```
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 -0.03553782  0.002087957 0.1144337
## 8-1    -0.03773623 -0.05661038 -0.018862082 0.0000002
## 12-1   -0.08603296 -0.10495715 -0.067108777 0.0000000
## 25-1   -0.17558830 -0.19433212 -0.156844478 0.0000000
```

```
## 40-1   -0.21481107 -0.23369487 -0.195927266 0.0000000
## 8-4.5  -0.02101130 -0.03975109 -0.002271512 0.0175580
## 12-4.5 -0.06930803 -0.08809822 -0.050517850 0.0000000
## 25-4.5 -0.15886337 -0.17747189 -0.140254849 0.0000000
## 40-4.5 -0.19808614 -0.21683565 -0.179336628 0.0000000
## 12-8   -0.04829674 -0.06714825 -0.029445217 0.0000000
## 25-8   -0.13785207 -0.15652252 -0.119181620 0.0000000
## 40-8   -0.17707484 -0.19588582 -0.158263863 0.0000000
## 25-12  -0.08955534 -0.10827637 -0.070834300 0.0000000
## 40-12  -0.12877811 -0.14763929 -0.109916919 0.0000000
## 40-25  -0.03922277 -0.05790298 -0.020542556 0.0000000
```

```
plot(CIs_strLevel)
```



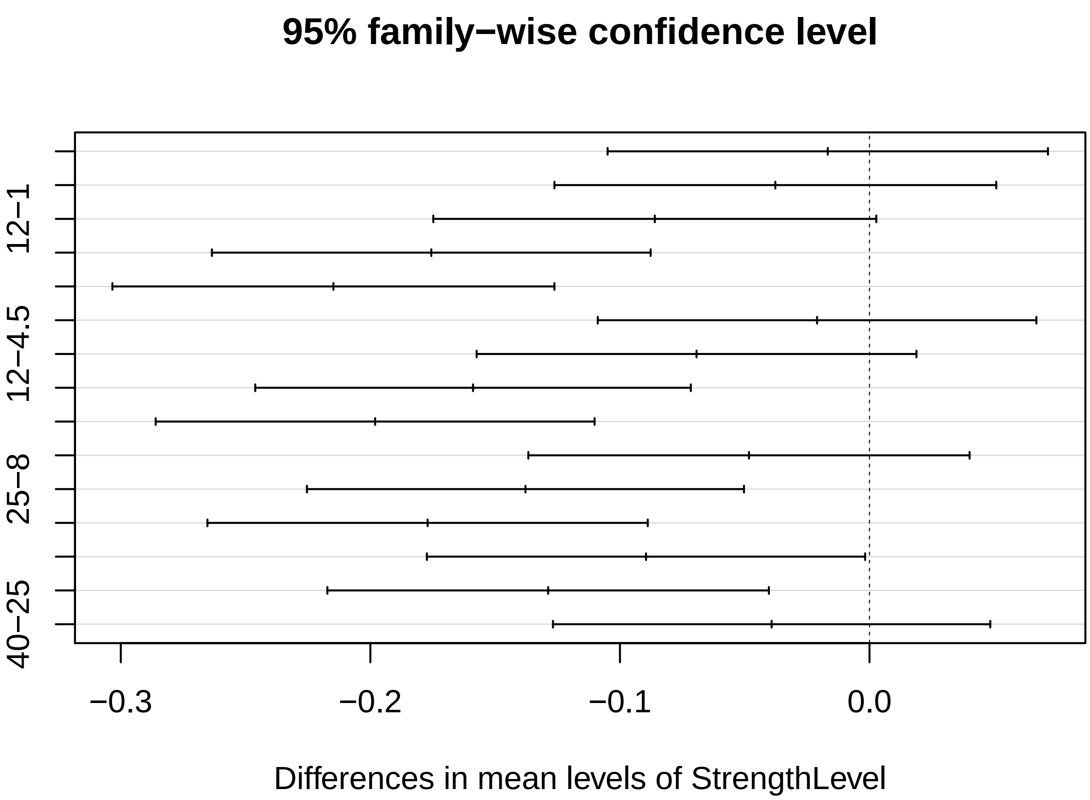
3.5 Scheffe's Method for Pairwise Comparisons

```
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 -0.1049482  0.071498386 1.0000
## 8-1    -0.03773623 -0.1262468  0.050774367 1.0000
## 12-1   -0.08603296 -0.1747782  0.002712295 0.1392
## 25-1   -0.17558830 -0.2634877 -0.087688863 <2e-16 ***
## 40-1   -0.21481107 -0.3033670 -0.126255188 <2e-16 ***
```

```
## 8-4.5  -0.02101130 -0.1088918  0.066869210 1.0000
## 12-4.5 -0.06930803 -0.1574249  0.018808817 0.9920
## 25-4.5 -0.15886337 -0.2461283 -0.071598433 <2e-16 ***
## 40-4.5 -0.19808614 -0.2860123 -0.110160020 <2e-16 ***
## 12-8   -0.04829674 -0.1367012  0.040107741 1.0000
## 25-8   -0.13785207 -0.2254074 -0.050296710 <2e-16 ***
## 40-8   -0.17707484 -0.2652892 -0.088860474 <2e-16 ***
## 25-12  -0.08955534 -0.1773479 -0.001762757 0.0222 *
## 40-12  -0.12877811 -0.2172279 -0.040328288 <2e-16 ***
## 40-25  -0.03922277 -0.1268239  0.048378373 1.0000
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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



IV. Logistic Regression Model for Accuracy

4.1 Train-test Split

```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)
```

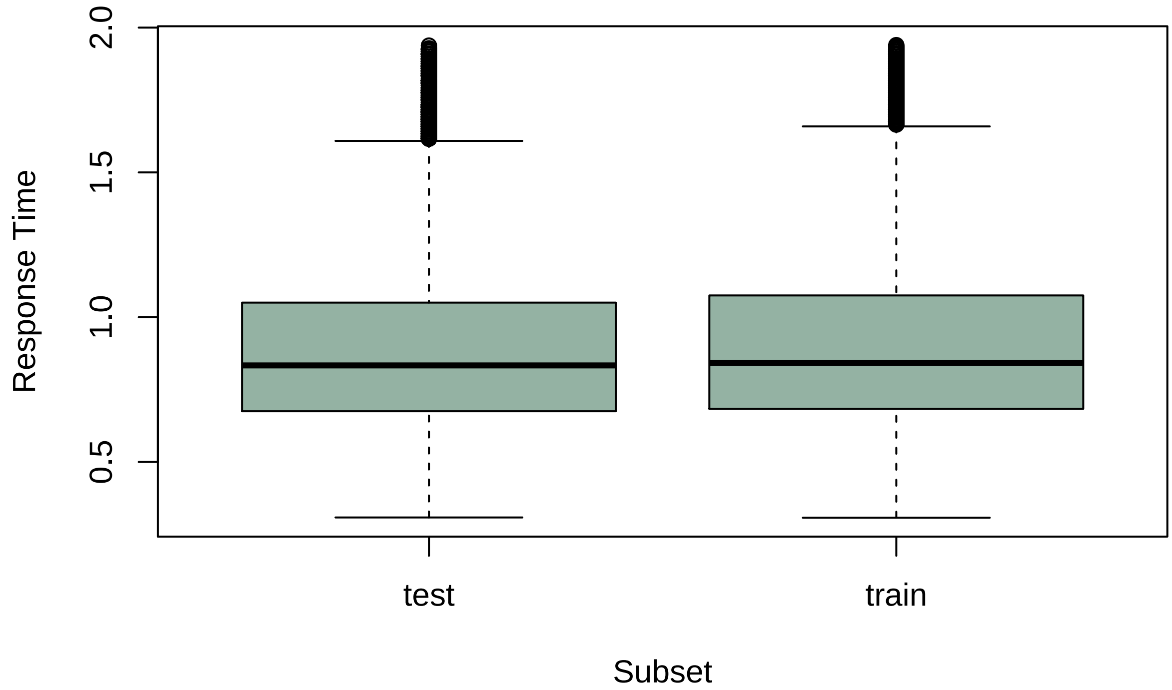
```
## Joining, by = c("ParticipantNum", "StrengthLevel", "ResponseTime", "Accuracy", "Confidence")
```

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

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

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

```r
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.
```



Kolomogorov-Smirnov tests (for continuous response time) (evaluate their similarity by measuring the differences between the ECDFs)

```r
ks.test(train$ResponseTime, test$ResponseTime)
```

```
## Warning in ks.test(train$ResponseTime, test$ResponseTime): p-value will be
## approximate in the presence of ties

##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  train$ResponseTime and test$ResponseTime
## D = 0.020781, p-value = 0.02768
## alternative hypothesis: two-sided
```

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

```r
acc_fit0 = glmer(Accuracy~StrengthLevel*ResponseTime+(1|ParticipantNum), data = train, family = binomia
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
##  11130.4  11167.4  -5560.2  11120.4    11995
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -15.7384   0.0863   0.2885   0.6294   1.7146
##
## Random effects:
##  Groups         Name        Variance Std.Dev.
##  ParticipantNum (Intercept) 0.128    0.3577
## Number of obs: 12000, groups:  ParticipantNum, 25
##
## Fixed effects:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)                0.663162   0.133851   4.954 7.25e-07 ***
## StrengthLevel              0.134244   0.008955  14.991  < 2e-16 ***
## ResponseTime              -0.495258   0.109756  -4.512 6.41e-06 ***
## StrengthLevel:ResponseTime -0.048223   0.008647  -5.577 2.45e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) StrngL RspnsT
## StrengthLvl -0.583
## ResponseTim -0.808  0.650
## StrngthL:RT  0.549 -0.944 -0.679
```

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

```r
acc_fit1 = glmer(Accuracy~StrengthLevel+ResponseTime+(1|ParticipantNum), data = train, family = binomia
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
##  11158.3  11187.9  -5575.2  11150.3    11996
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -9.8091  0.1131  0.2914  0.6174  1.9591
##
## Random effects:
##  Groups         Name        Variance Std.Dev.
##  ParticipantNum (Intercept) 0.1344   0.3666
## Number of obs: 12000, groups:  ParticipantNum, 25
##
```

```
## Fixed effects:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)    1.054894   0.114571   9.207   <2e-16 ***
## StrengthLevel  0.088869   0.002923  30.402   <2e-16 ***
## ResponseTime  -0.902597   0.081627 -11.058   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) StrngL
## StrengthLvl -0.276
## ResponseTim -0.712  0.102
```

4.5 Output the Table

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

```
##
## \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]
##  StrengthLevel & 0.134$^{***}$ & 0.089$^{***}$ \\
##   & (0.117, 0.152) & (0.083, 0.095) \\
##   & & \\
##  ResponseTime & $-$0.495$^{***}$ & $-$0.903$^{***}$ \\
##   & ($-$0.710, $-$0.280) & ($-$1.063, $-$0.743) \\
##   & & \\
##  StrengthLevel:ResponseTime & $-$0.048$^{***}$ &  \\
##   & ($-$0.065, $-$0.031) &  \\
##   & & \\
##  Constant & 0.663$^{***}$ & 1.055$^{***}$ \\
##   & (0.401, 0.926) & (0.830, 1.279) \\
##   & & \\
## \hline \\[-1.8ex]
## Observations & 12,000 & 12,000 \\
## Log Likelihood & $-$5,560.221 & $-$5,575.172 \\
## Akaike Inf. Crit. & 11,130.440 & 11,158.340 \\
## Bayesian Inf. Crit. & 11,167.410 & 11,187.910 \\
## \hline
## \hline \\[-1.8ex]
## \textit{Note:}  & \multicolumn{2}{r}{$^{*}$p$<$0.1; $^{**}$p$<$0.05; $^{***}$p$<$0.01} \\
## \end{tabular}
## \end{table}
```

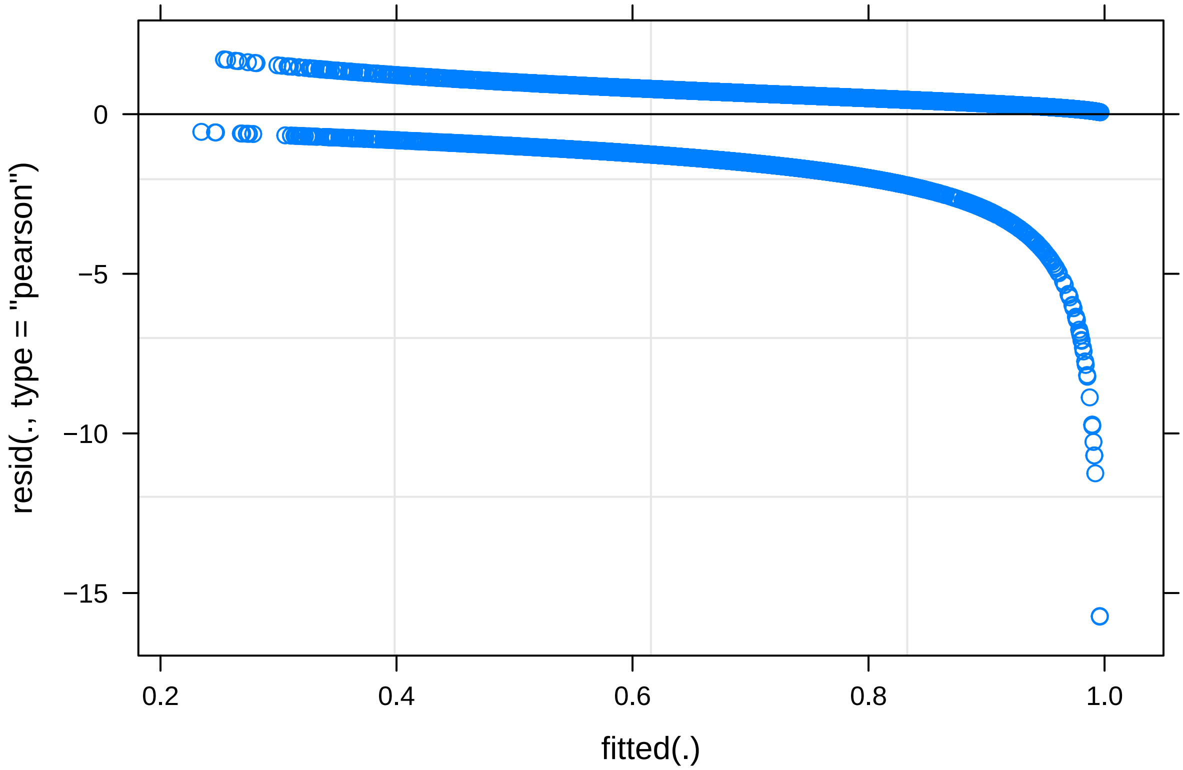
4.6 Likelihood Ratio Tests (Goodness-of-fit)

```
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   5 -5560.2
## 2   4 -5575.2 -1 29.901  4.547e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

4.7 Diagostics

```
plot(acc_fit0)
```



4.8 Predictions

```
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] 0.2283549
```

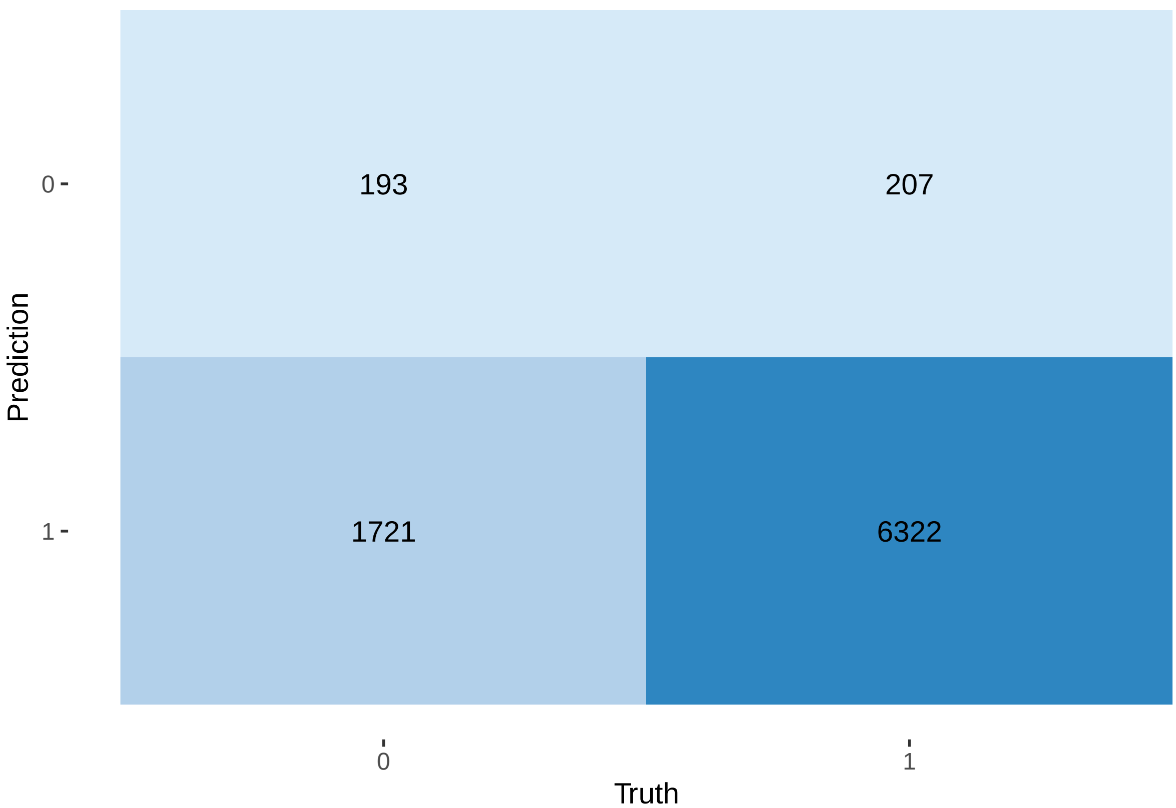
```
confusion_matrix_acc = table(Behave_fit_pred, test$Accuracy)
confusion_matrix_acc
```

Confusion Matrix

```
##
## Behave_fit_pred    0    1
##               0  193  207
##               1 1721 6322
```

```
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.
```




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

AUROC

```
## [1] 0.7676048
```

```r
sensitivity(confusion_matrix_acc)
```

Sensitivity and Specificity

```
## [1] 0.1008359
```

```r
specificity(confusion_matrix_acc)
```

```
## [1] 0.9682953
```

V. Logistic Regression Model for Confidence

5.1 Map the Confidence into 2 Categories

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

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

```r
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
##  11174.6  11211.6  -5582.3  11164.6    11995
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -7.9125 -0.5439  0.1481  0.5249 10.7309
##
## Random effects:
##  Groups         Name        Variance Std.Dev.
##  ParticipantNum (Intercept) 2.299    1.516
## Number of obs: 12000, groups:  ParticipantNum, 25
##
## Fixed effects:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 1.692034   0.328744   5.147 2.65e-07 ***
## StrengthLevel               0.039644   0.006241   6.353 2.12e-10 ***
## ResponseTime               -2.704387   0.129905 -20.818  < 2e-16 ***
## StrengthLevel:ResponseTime  0.031015   0.006758   4.590 4.44e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
```

```
##             (Intr) StrngL RspnsT
## StrengthLvl -0.273
## ResponseTim -0.369  0.672
## StrngthL:RT  0.252 -0.946 -0.690
```

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

```
conf_fit1 = glmer(Conf~StrengthLevel+ResponseTime+(1|ParticipantNum), data = behave_train, family = bin
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
##  11193.8  11223.4  -5592.9  11185.8    11996
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -8.6113 -0.5458  0.1393  0.5289 11.0075
##
## Random effects:
##  Groups         Name        Variance Std.Dev.
##  ParticipantNum (Intercept) 2.335    1.528
## Number of obs: 12000, groups:  ParticipantNum, 25
##
## Fixed effects:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)    1.313316   0.320119   4.103 4.09e-05 ***
## StrengthLevel  0.067019   0.002033  32.972  < 2e-16 ***
## ResponseTime  -2.298765   0.092864 -24.754  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) StrngL
## StrengthLvl -0.105
## ResponseTim -0.274  0.070
```

5.4 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]
##  StrengthLevel & 0.040$^{***}$ & 0.067$^{***}$ \\
##   & (0.027, 0.052) & (0.063, 0.071) \\
##   & & \\
##  ResponseTime & $-$2.704$^{***}$ & $-$2.299$^{***}$ \\
##   & ($-$2.959, $-$2.450) & ($-$2.481, $-$2.117) \\
##   & & \\
##  StrengthLevel:ResponseTime & 0.031$^{***}$ &  \\
##   & (0.018, 0.044) &  \\
##   & & \\
##  Constant & 1.692$^{***}$ & 1.313$^{***}$ \\
##   & (1.048, 2.336) & (0.686, 1.941) \\
##   & & \\
## \hline \\[-1.8ex]
## Observations & 12,000 & 12,000 \\
## Log Likelihood & $-$5,582.294 & $-$5,592.908 \\
## Akaike Inf. Crit. & 11,174.590 & 11,193.820 \\
## Bayesian Inf. Crit. & 11,211.550 & 11,223.390 \\
## \hline
## \hline \\[-1.8ex]
## \textit{Note:}  & \multicolumn{2}{r}{$^{*}$p$<$0.1; $^{**}$p$<$0.05; $^{***}$p$<$0.01} \\
## \end{tabular}
## \end{table}
```

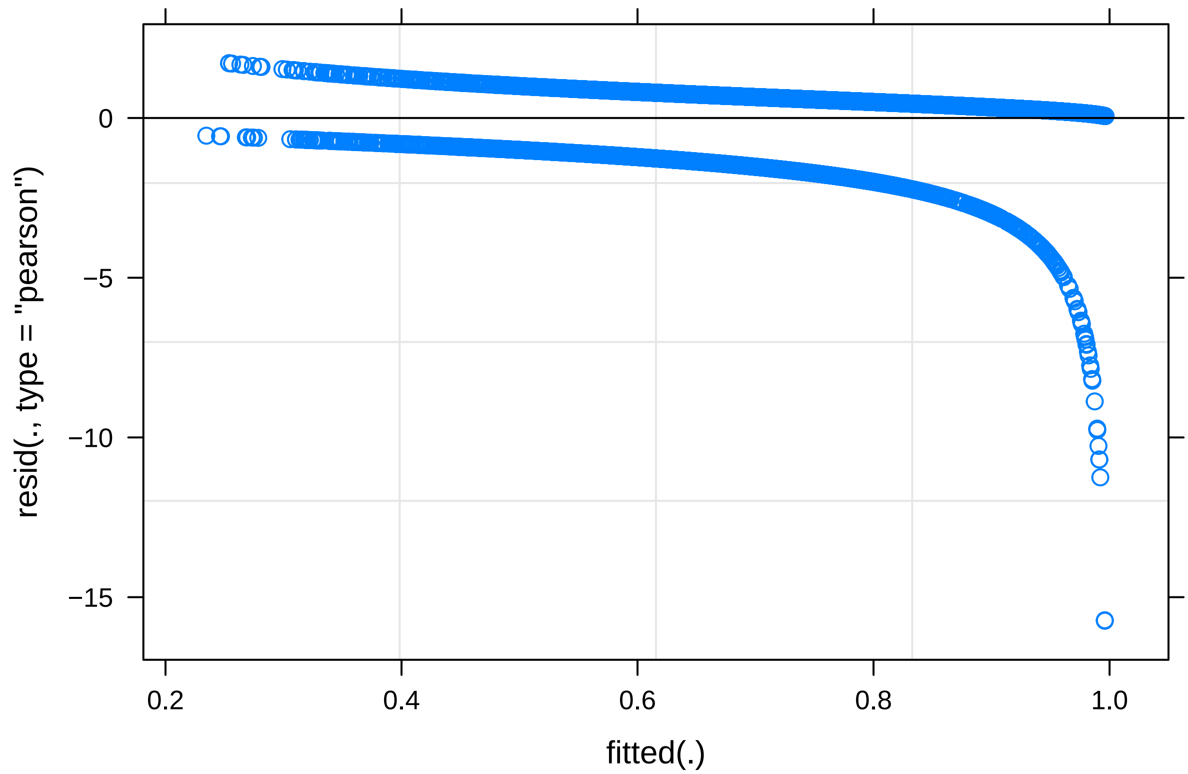
5.5 Likelihood Ratio Tests (Goodness-of-fit)

```
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   4 -5592.9
## 2   5 -5582.3  1 21.227   4.08e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5.6 Diagostics

```
plot(acc_fit0)
```



5.7 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] 0.2111809
```

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

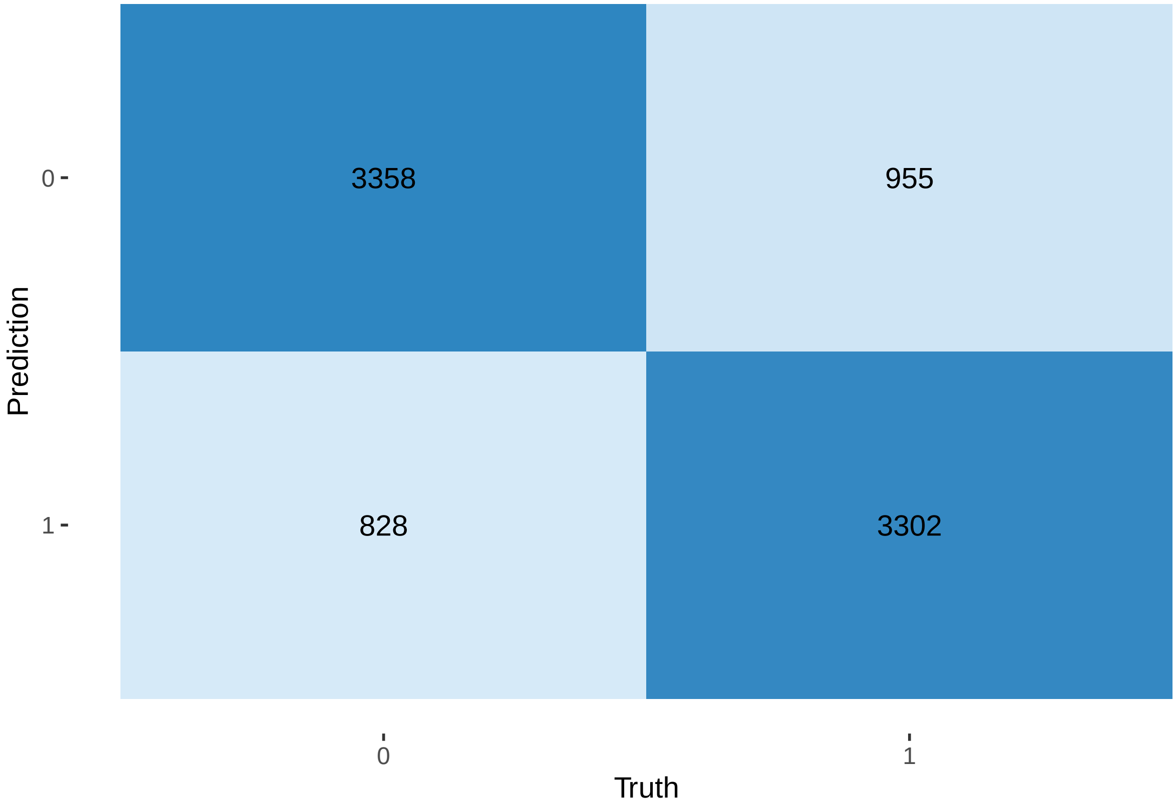
Confusion Matrix

```
##
## Behave_fit_pred    0    1
##               0 3358  955
##               1  828 3302
```

```
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.
```




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

AUROC

```
## [1] 0.8601211
```

```r
sensitivity(confusion_matrix_acc)
```

Sensitivity and Specificity

```
## [1] 0.8021978
```

```r
specificity(confusion_matrix_acc)
```

```
## [1] 0.7756636
```