project_EDA

Yu Zhu

10/26/2020

Behavior Data

I. Data Description

1.1 Import data

```
library(readxl)
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
```

1.3 Check Missing Data

```
which(is.na(Behave))
```

```
## integer(0)
```

No missing data found. Great!

1.4 Factorization

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

```
# 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
```

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

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

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

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

II. Explotary Data Analysis

2.1 Contingency table of Acc VS Participant Number

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

```
##
##       1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19
##   0 228 173 163 215 209 144 162 123 184 185 257 223 343 129 163 184 235 148 133
##   1 768 756 570 800 687 438 705 577 699 718 364 543 591 615 582 693 613 641 583
##
##      20  21  22  23  24  25
```

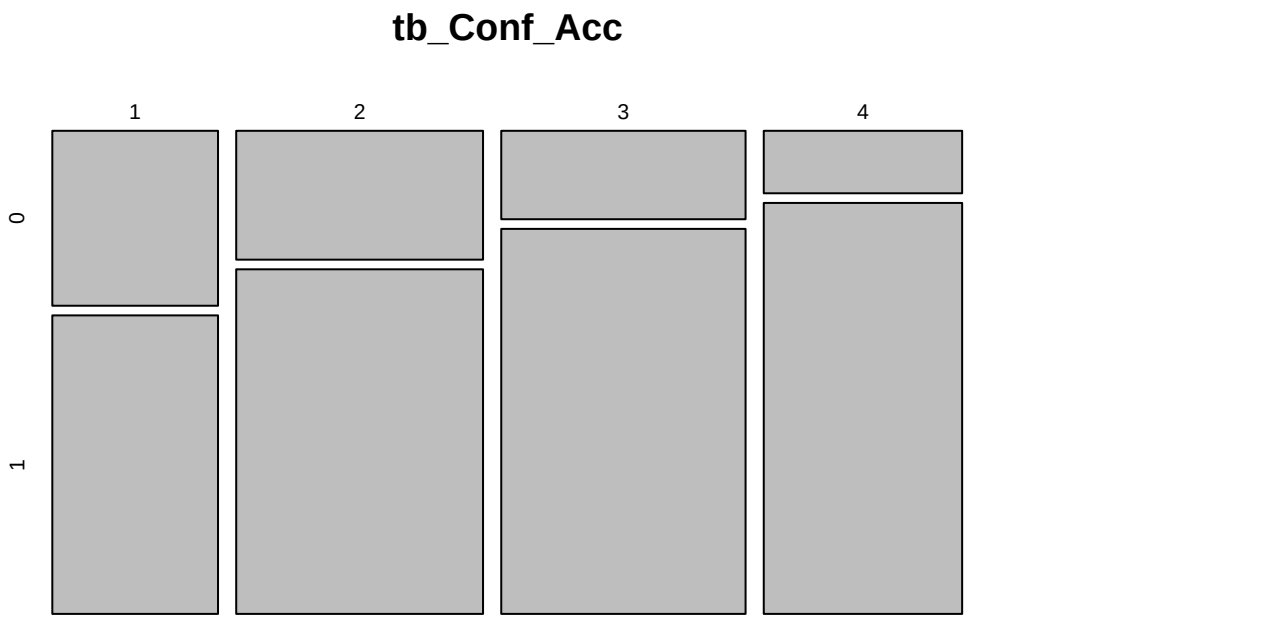
```
##   0 209 195 174 150 217 241
##   1 694 580 463 578 637 765
```

2.2 Contingency table of Acc VS Confidence

```
tb_Conf_Acc = table(Behave$Confidence, Behave$Accuracy); tb_Conf_Acc
```

```
##
##        0    1
##   1 1464 2498
##   2 1606 4295
##   3 1091 4750
##   4  626 4117
```

```
plot(tb_Conf_Acc)
```




```
chisq.test(tb_Conf_Acc) ### Perform independence test in detail later
```

```
##
##  Pearson's Chi-squared test
##
## data:  tb_Conf_Acc
## X-squared = 801.59, df = 3, p-value < 2.2e-16
```

Obviously, the larger the confidence, the more accurate the experiment.

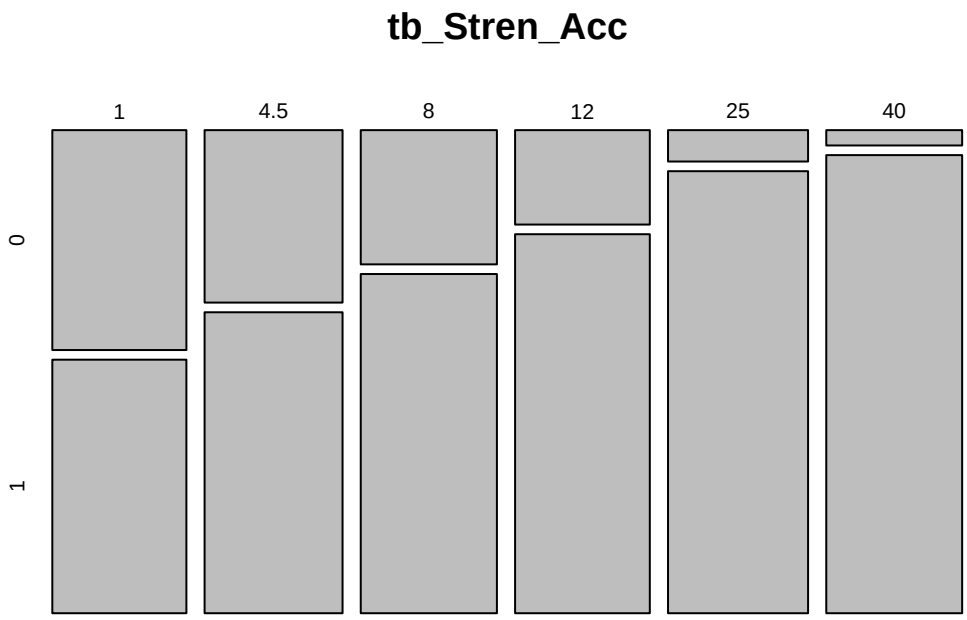
2.3 Contingency table of Acc VS Strength Level

```
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
```

```
plot(tb_Stren_Acc)
```




```
chisq.test(tb_Stren_Acc) ### Perform independence test in detail later
```

```
##
##  Pearson's Chi-squared test
##
## data:  tb_Stren_Acc
## X-squared = 2703.9, df = 5, p-value < 2.2e-16
```

Obviously, the larger the strength level, the more accurate the experiment.

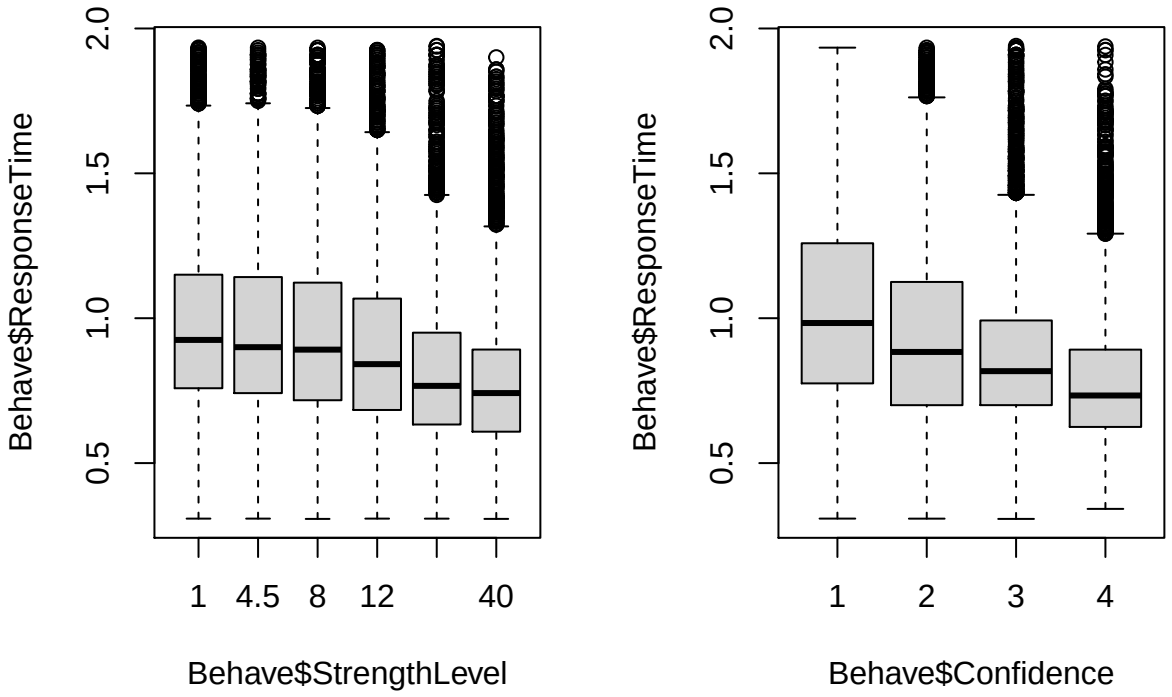
```
Behave_1 = data.frame(Behave$Confidence,Behave$StrengthLevel, Behave$Accuracy)
ftable(table(Behave_1))
```

```
##                                      Behave.Accuracy    0    1
## Behave.Confidence Behave.StrengthLevel
## 1                 1                                  530  554
##                   4.5                                379  597
##                   8                                  290  528
##                   12                                 191  429
##                   25                                  54  233
##                   40                                  20  157
## 2                 1                                  519  608
##                   4.5                                433  761
##                   8                                  336  862
##                   12                                 219  851
##                   25                                  71  720
##                   40                                  28  493
## 3                 1                                  335  391
##                   4.5                                281  526
##                   8                                  226  628
##                   12                                 161  854
##                   25                                  61 1224
```

```
##                 40                                  27 1127
## 4               1                                  171  240
##               4.5                                  162  306
##                 8                                  112  418
##                12                                  100  559
##                25                                   46 1088
##                40                                   35 1506
```

2.4 Boxplot of Reponse time Vs StrengthLevel and Confidence

```
par(mfrow = c(1,2))
boxplot(Behave$ResponseTime~Behave$StrengthLevel)
boxplot(Behave$ResponseTime~Behave$Confidence)
```



From the boxplots, there shows the pattern that the response time decrease along with the increasing of strength level and confidence.

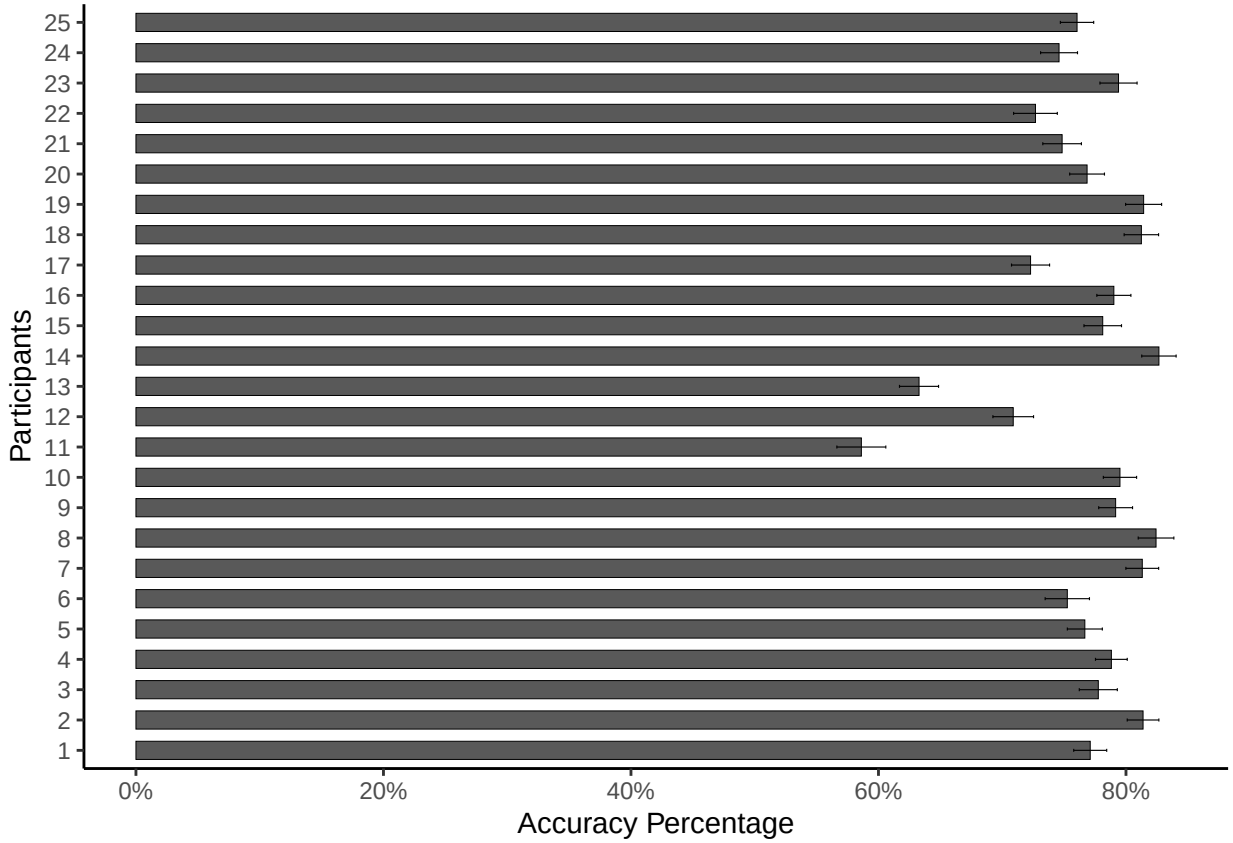
2.5 Accuracy rate

Barplot Here I tried to transfer ACC into accuracy rate, taking means for each participant

```
agg.ParticipantNum = aggregate(Accuracy~ParticipantNum, data = Behave,  mean)
agg.ParticipantNum_len = aggregate(Accuracy~ParticipantNum,
                        data = Behave,  length)
agg.ParticipantNum_sd = aggregate(Accuracy~ParticipantNum,
                           data = Behave, function(x) sd(x))
agg.ParticipantNum_se = agg.ParticipantNum_sd$Accuracy/sqrt(agg.ParticipantNum_len$Accuracy)
agg.ParticipantNum$se = agg.ParticipantNum_se
```

```
library(ggplot2)
gp <- ggplot(agg.ParticipantNum , aes(x=ParticipantNum, y=Accuracy)) +
  geom_col( size=.05, color = "black",show.legend = FALSE,width = 0.6) +
  scale_fill_grey()+
```

```
  geom_errorbar(aes(ymax=Accuracy+se, ymin=Accuracy-se), width=.1,size = 0.2, color = 'black')+
  theme_classic() +
  scale_y_continuous(labels = scales::percent) +
  xlab("Participants") +
  ylab("Accuracy Percentage")
gp+coord_flip()
```



The above barplot shows the mean accuracy rate of different participants.

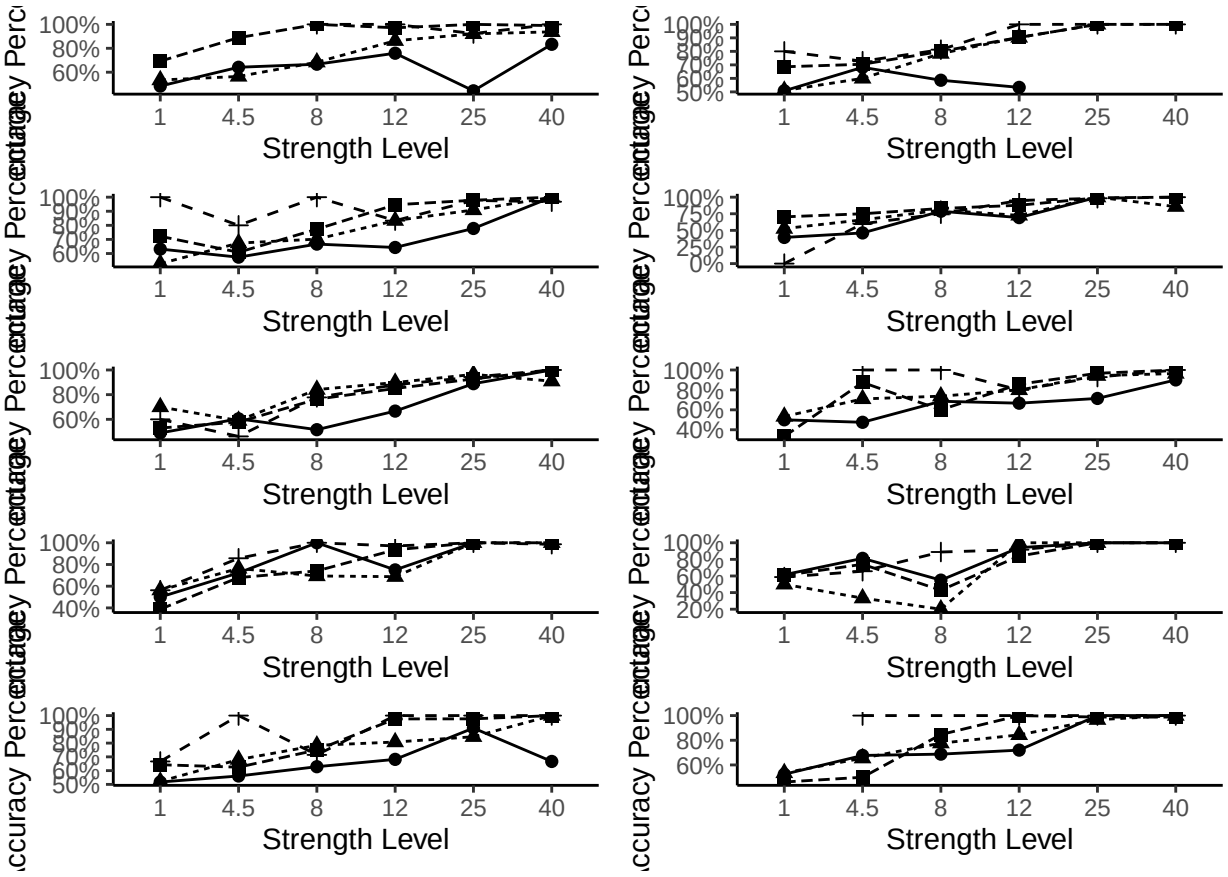
Interaction plot Here I want to plot interaction plot to show interactions for strength level and confidence.

```
## FOR DIFFERENT Confidence
plot_F_interaction = function(Behave) {
  gp = list()
  # pick first 10 participants
  for (i in 1:10) {
    dataset.par= Behave[which(Behave$ParticipantNum == unique(Behave$ParticipantNum)[i]),]
    ### interaction plots
    agg.stren.conf = aggregate(Accuracy~StrengthLevel + Confidence,
                               data = dataset.par,  mean)

    gp[[i]] <- ggplot(agg.stren.conf, aes(x=StrengthLevel, y=Accuracy, colour=Confidence, group=Confider
      geom_line(aes(linetype=Confidence), size=.5, color = "black",show.legend = FALSE) +
      geom_point(aes(shape=Confidence), size=2, color = 'black',show.legend = FALSE) +
     # geom_errorbar(aes(ymax=recovery+se, ymin=recovery-se), width=.1,size = 0.2, color = 'black')+
      theme_classic() +
      scale_y_continuous(labels = scales::percent) +
      xlab("Strength Level") +
```

```
      ylab("Accuracy Percentage")
  }
  library("gridExtra")
  grid.arrange(gp[[1]], gp[[2]],gp[[3]],gp[[4]],gp[[5]],gp[[6]],gp[[7]],gp[[8]],gp[[9]], gp[[10]], nrow
}
```

```
plot_F_interaction(Behave)
```



There seems to have mild interactions between strength level and confidence. So when we fit model, we could consider to add interaction terms.

III. Models

3.1 GLM Model

Fist we want to fit logistic regression model here with response variable Acc.

```
fit1 = glm(Accuracy~., data = Behave, family = binomial) # additive model
summary(fit1)
```

```
##
## Call:
## glm(formula = Accuracy ~ ., family = binomial, data = Behave)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8765   0.1858   0.3790   0.7812   2.0482
##
```

```
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)        0.903875   0.116755   7.742 9.81e-15 ***
## ParticipantNum2    0.004375   0.123110   0.036  0.97165
## ParticipantNum3   -0.037437   0.125640  -0.298  0.76572
## ParticipantNum4   -0.062213   0.117186  -0.531  0.59549
## ParticipantNum5   -0.067083   0.118450  -0.566  0.57116
## ParticipantNum6   -0.024292   0.133895  -0.181  0.85603
## ParticipantNum7    0.039192   0.125564   0.312  0.75494
## ParticipantNum8   -0.121380   0.138962  -0.873  0.38240
## ParticipantNum9    0.047108   0.121261   0.388  0.69766
## ParticipantNum10   0.048411   0.120550   0.402  0.68799
## ParticipantNum11 -1.412392   0.128216 -11.016  < 2e-16 ***
## ParticipantNum12 -0.877692   0.126110  -6.960 3.41e-12 ***
## ParticipantNum13 -0.828562   0.111258  -7.447 9.53e-14 ***
## ParticipantNum14   0.200521   0.134345   1.493  0.13555
## ParticipantNum15 -0.348858   0.130976  -2.664  0.00773 **
## ParticipantNum16 -0.321219   0.126445  -2.540  0.01107 *
## ParticipantNum17 -0.541996   0.119452  -4.537 5.70e-06 ***
## ParticipantNum18   0.260032   0.127476   2.040  0.04136 *
## ParticipantNum19   0.037916   0.132021   0.287  0.77396
## ParticipantNum20 -0.038183   0.119182  -0.320  0.74868
## ParticipantNum21 -0.608340   0.124527  -4.885 1.03e-06 ***
## ParticipantNum22 -0.607646   0.130007  -4.674 2.95e-06 ***
## ParticipantNum23   0.020401   0.129916   0.157  0.87522
## ParticipantNum24 -0.354272   0.119708  -2.959  0.00308 **
## ParticipantNum25 -0.501030   0.117793  -4.253 2.10e-05 ***
## StrengthLevel4.5   0.392838   0.050739   7.742 9.76e-15 ***
## StrengthLevel8     0.754145   0.052833  14.274  < 2e-16 ***
## StrengthLevel12    1.158294   0.057004  20.319  < 2e-16 ***
## StrengthLevel25    2.277537   0.078767  28.915  < 2e-16 ***
## StrengthLevel40    2.976192   0.105958  28.088  < 2e-16 ***
## ResponseTime      -0.784543   0.067164 -11.681  < 2e-16 ***
## Confidence2        0.198144   0.048696   4.069 4.72e-05 ***
## Confidence3        0.461658   0.057476   8.032 9.58e-16 ***
## Confidence4        0.672325   0.074119   9.071  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 22255  on 20446  degrees of freedom
## Residual deviance: 18561  on 20413  degrees of freedom
## AIC: 18629
##
## Number of Fisher Scoring iterations: 6
```

We can consider to use AIC criterion to fit the best model.

```
summary(step(fit1))
```

```
## Start:  AIC=18629.2
## Accuracy ~ ParticipantNum + StrengthLevel + ResponseTime + Confidence
##
##                  Df Deviance   AIC
```

```
## <none>                 18561 18629
## - Confidence      3    18659 18721
## - ResponseTime    1    18697 18763
## - ParticipantNum 24    18979 18999
## - StrengthLevel   5    20582 20640
```

```
##
## Call:
## glm(formula = Accuracy ~ ParticipantNum + StrengthLevel + ResponseTime +
##     Confidence, family = binomial, data = Behave)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8765   0.1858   0.3790   0.7812   2.0482
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)       0.903875   0.116755   7.742 9.81e-15 ***
## ParticipantNum2   0.004375   0.123110   0.036  0.97165
## ParticipantNum3  -0.037437   0.125640  -0.298  0.76572
## ParticipantNum4  -0.062213   0.117186  -0.531  0.59549
## ParticipantNum5  -0.067083   0.118450  -0.566  0.57116
## ParticipantNum6  -0.024292   0.133895  -0.181  0.85603
## ParticipantNum7   0.039192   0.125564   0.312  0.75494
## ParticipantNum8  -0.121380   0.138962  -0.873  0.38240
## ParticipantNum9   0.047108   0.121261   0.388  0.69766
## ParticipantNum10  0.048411   0.120550   0.402  0.68799
## ParticipantNum11 -1.412392   0.128216 -11.016  < 2e-16 ***
## ParticipantNum12 -0.877692   0.126110  -6.960 3.41e-12 ***
## ParticipantNum13 -0.828562   0.111258  -7.447 9.53e-14 ***
## ParticipantNum14  0.200521   0.134345   1.493  0.13555
## ParticipantNum15 -0.348858   0.130976  -2.664  0.00773 **
## ParticipantNum16 -0.321219   0.126445  -2.540  0.01107 *
## ParticipantNum17 -0.541996   0.119452  -4.537 5.70e-06 ***
## ParticipantNum18  0.260032   0.127476   2.040  0.04136 *
## ParticipantNum19  0.037916   0.132021   0.287  0.77396
## ParticipantNum20 -0.038183   0.119182  -0.320  0.74868
## ParticipantNum21 -0.608340   0.124527  -4.885 1.03e-06 ***
## ParticipantNum22 -0.607646   0.130007  -4.674 2.95e-06 ***
## ParticipantNum23  0.020401   0.129916   0.157  0.87522
## ParticipantNum24 -0.354272   0.119708  -2.959  0.00308 **
## ParticipantNum25 -0.501030   0.117793  -4.253 2.10e-05 ***
## StrengthLevel4.5  0.392838   0.050739   7.742 9.76e-15 ***
## StrengthLevel8    0.754145   0.052833  14.274  < 2e-16 ***
## StrengthLevel12   1.158294   0.057004  20.319  < 2e-16 ***
## StrengthLevel25   2.277537   0.078767  28.915  < 2e-16 ***
## StrengthLevel40   2.976192   0.105958  28.088  < 2e-16 ***
## ResponseTime     -0.784543   0.067164 -11.681  < 2e-16 ***
## Confidence2       0.198144   0.048696   4.069 4.72e-05 ***
## Confidence3       0.461658   0.057476   8.032 9.58e-16 ***
## Confidence4       0.672325   0.074119   9.071  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 22255  on 20446  degrees of freedom
## Residual deviance: 18561  on 20413  degrees of freedom
## AIC: 18629
##
## Number of Fisher Scoring iterations: 6
```

So this additive model has the best AIC.

Response time is significant here, but we can still exclude response time adn fit ANOVA to see what will happen.

```
fit2 = glm(Accuracy~ParticipantNum + Confidence + StrengthLevel, data = Behave, family = binomial) # ad
summary(fit2)
```

```
##
## Call:
## glm(formula = Accuracy ~ ParticipantNum + Confidence + StrengthLevel,
##     family = binomial, data = Behave)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8316   0.1862   0.3841   0.7885   1.8378
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)       0.027270   0.089478   0.305  0.76054
## ParticipantNum2   0.078437   0.122957   0.638  0.52353
## ParticipantNum3   0.022078   0.125625   0.176  0.86049
## ParticipantNum4  -0.025023   0.116727  -0.214  0.83025
## ParticipantNum5  -0.122935   0.118241  -1.040  0.29848
## ParticipantNum6  -0.231845   0.132682  -1.747  0.08057 .
## ParticipantNum7  -0.003792   0.125445  -0.030  0.97589
## ParticipantNum8  -0.076354   0.138502  -0.551  0.58144
## ParticipantNum9   0.009837   0.120974   0.081  0.93519
## ParticipantNum10  0.117138   0.120356   0.973  0.33042
## ParticipantNum11 -1.511709   0.127825 -11.826  < 2e-16 ***
## ParticipantNum12 -0.905086   0.125980  -7.184 6.75e-13 ***
## ParticipantNum13 -0.835407   0.111194  -7.513 5.78e-14 ***
## ParticipantNum14  0.140468   0.133084   1.055  0.29121
## ParticipantNum15 -0.389592   0.130610  -2.983  0.00286 **
## ParticipantNum16 -0.403415   0.125514  -3.214  0.00131 **
## ParticipantNum17 -0.258169   0.116870  -2.209  0.02717 *
## ParticipantNum18  0.237645   0.127199   1.868  0.06172 .
## ParticipantNum19  0.215079   0.131014   1.642  0.10066
## ParticipantNum20 -0.127446   0.118540  -1.075  0.28232
## ParticipantNum21 -0.529821   0.124290  -4.263 2.02e-05 ***
## ParticipantNum22 -0.635773   0.129948  -4.893 9.95e-07 ***
## ParticipantNum23 -0.116231   0.128929  -0.902  0.36731
## ParticipantNum24 -0.185631   0.118829  -1.562  0.11825
## ParticipantNum25 -0.168322   0.114136  -1.475  0.14028
## Confidence2       0.265347   0.048160   5.510 3.59e-08 ***
## Confidence3       0.612710   0.055887  10.963  < 2e-16 ***
## Confidence4       0.921494   0.070987  12.981  < 2e-16 ***
## StrengthLevel4.5  0.393987   0.050513   7.800 6.20e-15 ***
```

```
## StrengthLevel8    0.755990   0.052585  14.376  < 2e-16 ***
## StrengthLevel12   1.177464   0.056747  20.749  < 2e-16 ***
## StrengthLevel25   2.319228   0.078473  29.554  < 2e-16 ***
## StrengthLevel40   3.019729   0.105644  28.584  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 22255  on 20446  degrees of freedom
## Residual deviance: 18697  on 20414  degrees of freedom
## AIC: 18763
##
## Number of Fisher Scoring iterations: 6
```

ANOVA:

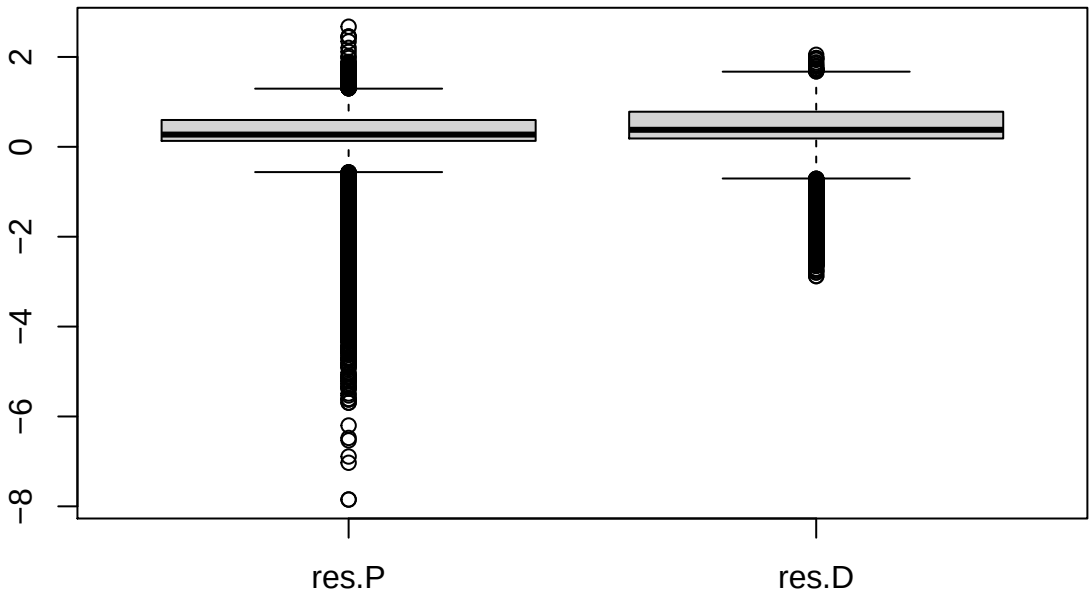
```
anova(fit2, test = 'Chi')
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Accuracy
##
## Terms added sequentially (first to last)
##
##
##                Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                           20446      22255
## ParticipantNum 24   305.03     20422      21950 < 2.2e-16 ***
## Confidence      3  1115.15     20419      20834 < 2.2e-16 ***
## StrengthLevel   5  2137.27     20414      18697 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

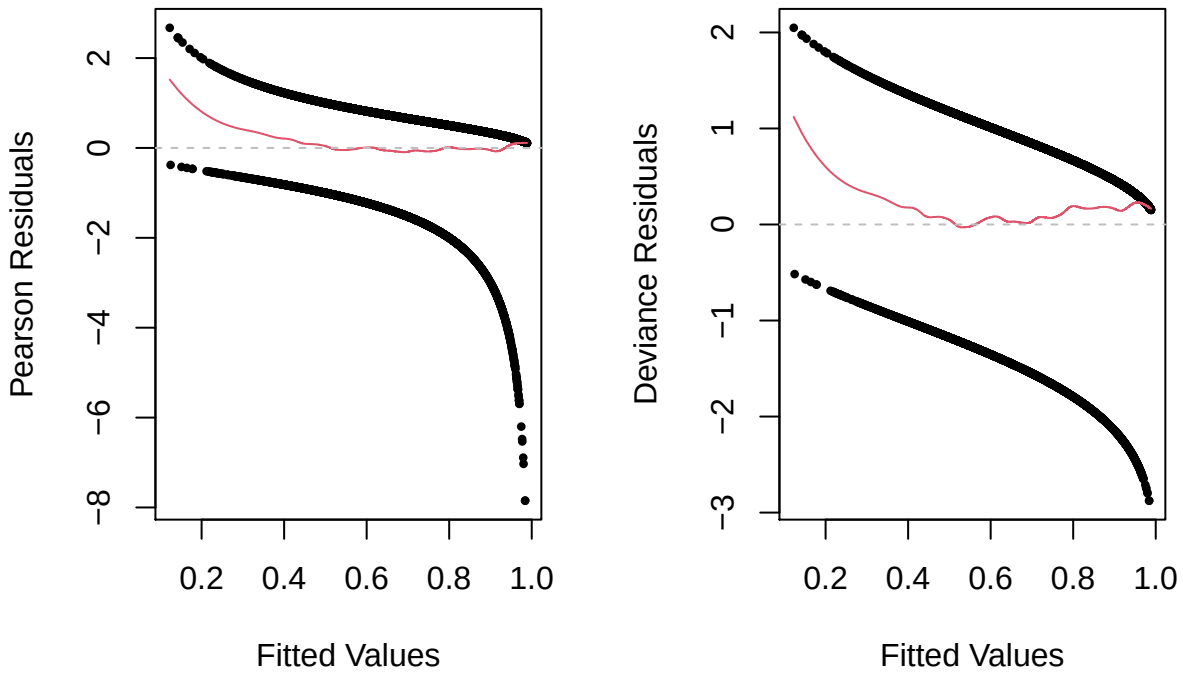
All are significant.

Check goodness of fit:

```
res.P = residuals(fit1, type="pearson")
res.D = residuals(fit1, type="deviance") #or residuals(fit), by default
boxplot(cbind(res.P, res.D), labels = c("Pearson", "Deviance"))
```




```
par(mfrow=c(1,2))
plot(fit1$fitted.values, res.P, pch=16, cex=0.6, ylab='Pearson Residuals', xlab='Fitted Values')
lines(smooth.spline(fit1$fitted.values, res.P, spar=0.9), col=2)
abline(h=0, lty=2, col='grey')
plot(fit1$fitted.values, res.D, pch=16, cex=0.6, ylab='Deviance Residuals', xlab='Fitted Values')
lines(smooth.spline(fit1$fitted.values, res.D, spar=0.9), col=2)
abline(h=0, lty=2, col='grey')
```



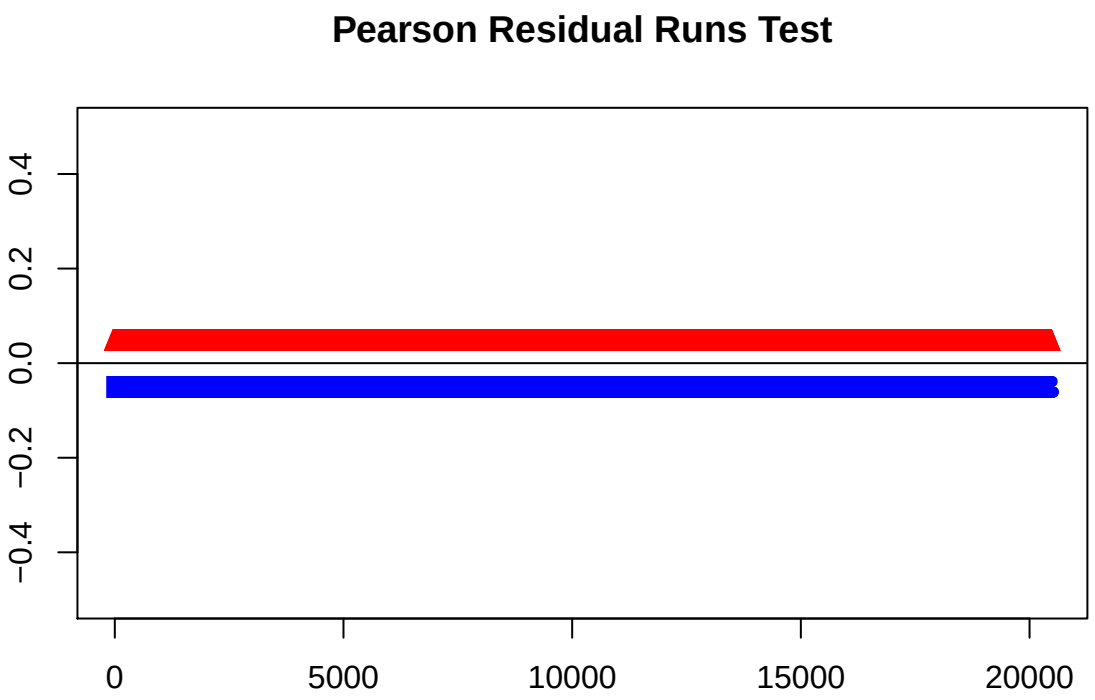
Run tests:

```
library(lawstat)
runs.test(y = res.P, plot.it = TRUE)
```

```
##
```

```
##  Runs Test - Two sided
##
## data:  res.P
## Standardized Runs Statistic = -1.3917, p-value = 0.164
```

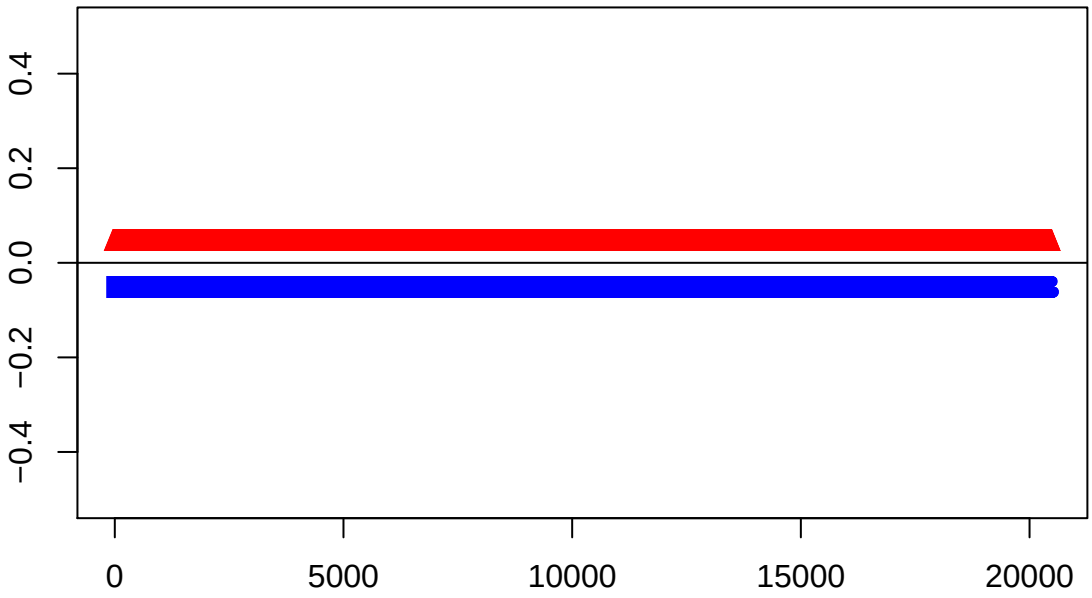
```
title(main='Pearson Residual Runs Test')
```




```
runs.test(y = res.D, plot.it = TRUE)
```

```
##
##  Runs Test - Two sided
##
## data:  res.D
## Standardized Runs Statistic = -1.3917, p-value = 0.164
```

```
title(main='Deviance Residual Runs Test')
```



Sheffe's Pariwise Comparisions:

```
library(multcomp)
```

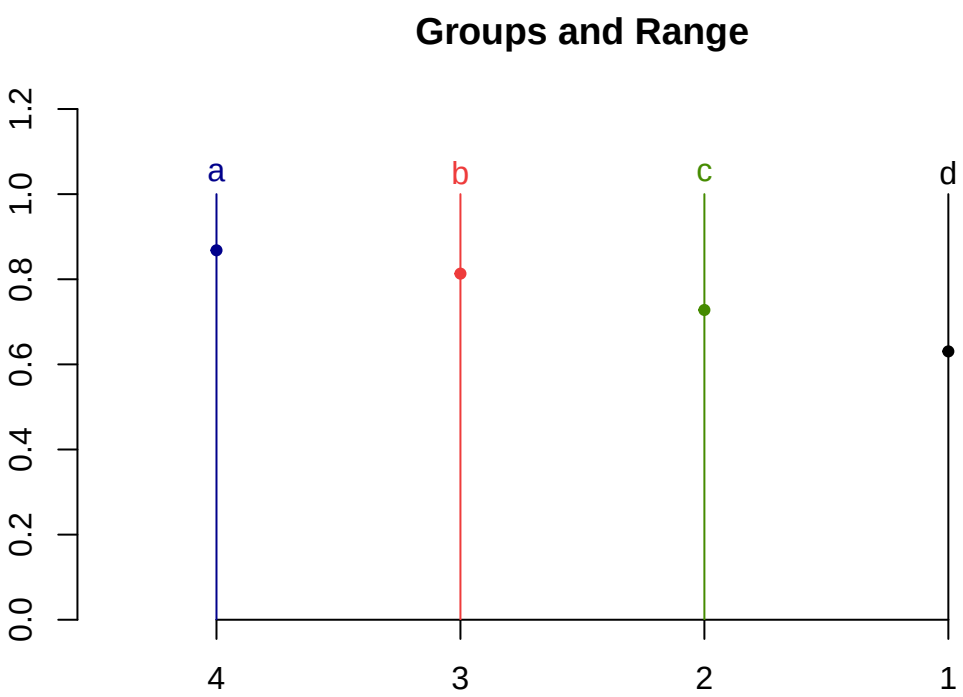
```
## Loading required package: mvtnorm
## Loading required package: survival
## Loading required package: TH.data
## Loading required package: MASS
##
## Attaching package: 'TH.data'
## The following object is masked from 'package:MASS':
##
##     geyser
```

```
library(agricolae)
par(mfrow = c(1,1))
sheffetest.1 = scheffe.test(fit1,'Confidence')
print(sheffetest.1)
```

```
## $statistics
##     MSerror    Df        F      Mean       CV
##   0.9092832 20413 2.605344 0.7658825 124.5052
##
## $parameters
##      test     name.t ntr alpha
##   Scheffe Confidence   4  0.05
##
## $means
##    Accuracy       std    r Min Max Q25 Q50 Q75
## 1 0.6304897 0.4827331 3962   0   1   0   1   1
## 2 0.7278427 0.4451081 5901   0   1   0   1   1
```

```
## 3 0.8132169 0.3897707 5841   0   1   1   1   1
## 4 0.8680160 0.3385090 4743   0   1   1   1   1
##
## $comparison
## NULL
##
## $groups
##    Accuracy groups
## 4 0.8680160      a
## 3 0.8132169      b
## 2 0.7278427      c
## 1 0.6304897      d
##
## attr(,"class")
## [1] "group"
```

```
plot(sheffetest.1)
```




```
sheffetest.2 = scheffe.test(fit1,'StrengthLevel')
print(sheffetest.2)
```

```
## $statistics
##     MSerror    Df        F      Mean       CV
##   0.9092832 20413 2.214537 0.7658825 124.5052
##
## $parameters
##      test        name.t ntr alpha
##   Scheffe StrengthLevel   6  0.05
##
## $means
##      Accuracy       std    r Min Max Q25 Q50 Q75
## 1   0.5355436 0.4988096 3348   0   1   0   1   1
## 12  0.8005351 0.3996575 3364   0   1   1   1   1
```

```
## 25  0.9336574 0.2489156 3497   0   1   1   1   1
## 4.5 0.6357039 0.4813021 3445   0   1   0   1   1
## 40  0.9675803 0.1771381 3393   0   1   1   1   1
## 8   0.7164706 0.4507774 3400   0   1   0   1   1
##
## $comparison
## NULL
##
## $groups
##      Accuracy groups
## 40  0.9675803      a
## 25  0.9336574      a
## 12  0.8005351      b
## 8   0.7164706      c
## 4.5 0.6357039      d
## 1   0.5355436      e
##
## attr(,"class")
## [1] "group"
```

```
plot(sheffetest.2)
```