Data Analysis of the Paper Design and Evaluation of a Handheld-based 3D User Interface for Collaborative Object Manipulation

Jerônimo G. Grandi

September 26, 2016

Contents

1 Data analyisis

This is the analysis of the data collected in the user experiment performed for the ACM CHI Conference on Human Factors in Computing Systems

2 Design and Procedure

We aim to investigate the relationship between group sizes and the time and accuracy to complete the tasks. Furthermore, we intend to understand the influence of work distribution balance and work division in the performance of each group combination. Thus, the experiment follows a between subject design with Group size as the only independent variable, with one, two, three or four participants. Dependent variables collected were time to complete the task and accuracy of thegroup, and transformation actions (translation, rotation, scale or camera rotation), including duration and magnitude of the action performed by each individual subject. The accuracy is measured as described before in Section Collaborative 3D Manipulation Assessment.

2.1 Task

We used the obstacle crossing game with three wall configurations. The training sessions consist of the first two walls. The test session is formed by one trial for each practice wall and two trials for the tunnel.

The results reported here only use the two trials in the tunnel task for the statistical analysis.

2.2 Subjects

Sixty subjects participated voluntarily in this experiment (nine female), aged 24 years in average (SD=3.6). They were all Computer Science students with no movement restrictions on wrists and arms. Thirteen of the individuals had never used gestural interactions with Kinect, Wiimote or mobile devices. We arranged the participants in 5 groups of one, 7 groups of two, 7 groups of three and 5 groups of four individuals.

2.3 Hypotesis

- H1. Groups with more than one member complete the tasks faster
- H2. Groups with more than one member complete the tasks with more accuracy
- H3. For the tested group size range, if groups increase in members, the time to complete tasks drops proportionally
- H4. For the tested group size range, if groups increase in members, the accuracy to complete tasks increase proportionally

3 Statistical Analysis Results

3.1 Data Summary

```
#sourceDataGroups <- read.csv("errorTimeAndVarPerTeam.csv",header = TRUE,  sep=";", dec = ",")
#sourceDataGroups <- read.csv("errorTimeAndVarPerTeamOnlyTask3and4.csv",header = TRUE,  sep="\t")
sourceDataGroups <- read.csv("errorMedianAndTimePerTeamOnlyTask3and4.csv", header = FALSE,  sep="\t")
sourceDataGroups
```

```
##          V1        V2       V3        V4       V5       V6       V7
## 1   44.5060  86.01300 127.2309 162.96200 7.861463 2.260774 2.711916
## 2   56.6990  90.59442  82.5638  87.06737 9.248310 2.767702 3.181414
## 3  129.5357  87.26190 120.6714  74.45340 5.619129 4.420655 4.090105
## 4   88.6565 113.11968  91.1117  71.80547 3.876767 3.795666 3.396920
## 5  130.7602  69.36490  76.8215  56.40480 6.004525 5.324064 4.424443
## 6   63.4430  58.41970  55.2902  63.90760 5.216207 3.984116 4.767183
## 7  281.5956 139.37750  86.8132 127.17520 2.660728 5.078261 3.269525
## 8  225.2061 111.04040  61.2443  60.76750 2.895181 2.542164 4.584655
## 9  105.6555 109.53340 283.4138 212.14000 7.370132 5.102038 1.681474
## 10 139.6341  83.16500 192.7240 113.40100 6.931314 6.086377 2.685176
## 11       NA  99.48090  41.2966        NA       NA 4.205052 4.419264
## 12       NA  93.11320  38.1163        NA       NA 4.758173 4.131297
##          V8
## 1  2.509332
## 2  1.978163
```

```
## 3  2.854236
## 4  1.911809
## 5  2.344555
## 6  2.784301
## 7  1.736374
## 8  1.853528
## 9  2.504749
## 10 1.641644
## 11       NA
## 12       NA
```

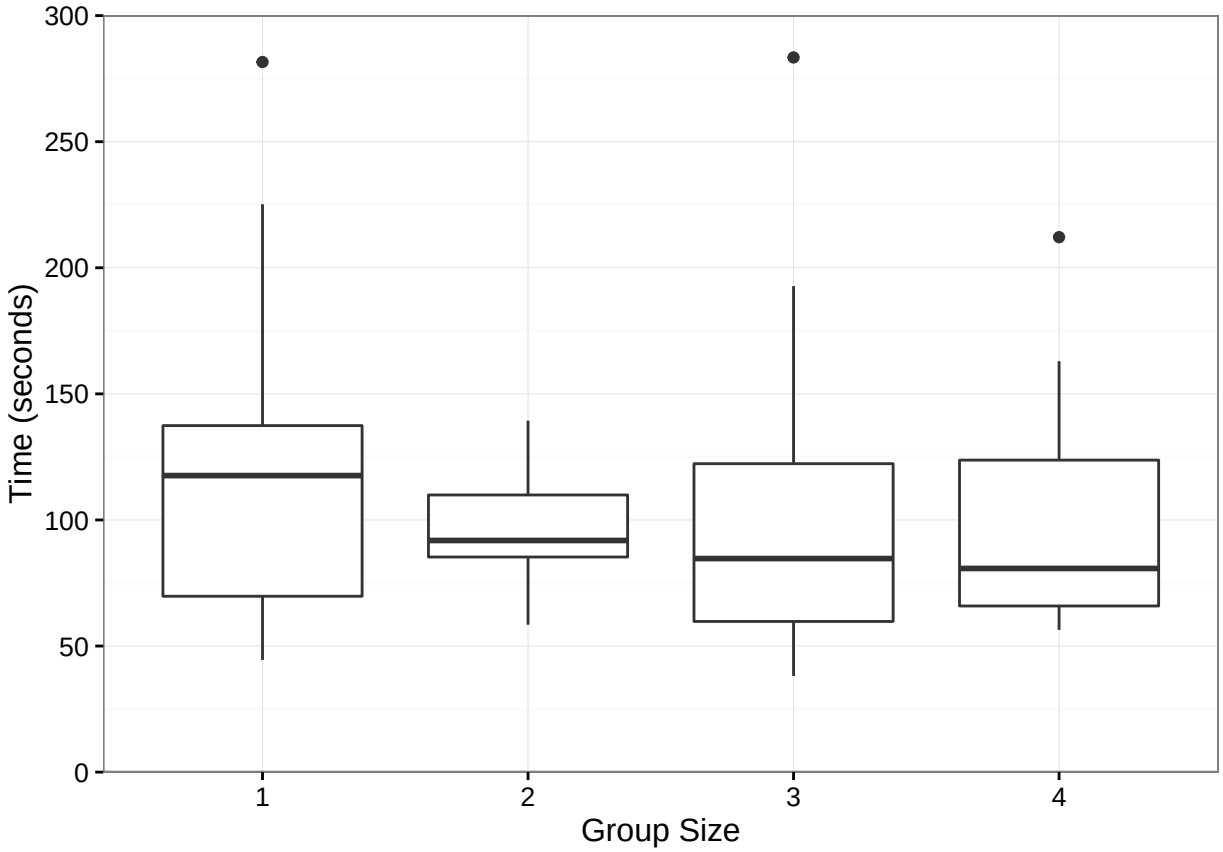
```
#sourceDataTasks <- read.csv("errorAndTimePerTask.csv",header = TRUE,  sep="\t")
sourceDataTasks <- read.csv("errorMedianAndTimePerTaskOnlyTask3and4.csv",header = TRUE,  sep="\t")
sourceDataTasks
```

```
##    Members.Task Time.task.3 Time.task.4 Error.task.3 Error.task.4
## 1             1     44.5060    56.69900     7.861463     9.248310
## 2             1    129.5357    88.65650     5.619129     3.876767
## 3             1    130.7602    63.44300     6.004525     5.216207
## 4             1    281.5956   225.20610     2.660728     2.895181
## 5             1    105.6555   139.63410     7.370132     6.931314
## 6             2     86.0130    90.59442     2.260774     2.767702
## 7             2     87.2619   113.11968     4.420655     3.795666
## 8             2     69.3649    58.41970     5.324064     3.984116
## 9             2    139.3775   111.04040     5.078261     2.542164
## 10            2    109.5334    83.16500     5.102038     6.086377
## 11            2     99.4809    93.11320     4.205052     4.758173
## 12            3    127.2309    82.56380     2.711916     3.181414
## 13            3    120.6714    91.11170     4.090105     3.396920
## 14            3     76.8215    55.29020     4.424443     4.767183
## 15            3     86.8132    61.24430     3.269525     4.584655
## 16            3    283.4138   192.72400     1.681474     2.685176
## 17            3     41.2966    38.11630     4.419264     4.131297
## 18            4    162.9620    87.06737     2.509332     1.978163
## 19            4     74.4534    71.80547     2.854236     1.911809
## 20            4     56.4048    63.90760     2.344555     2.784301
## 21            4    127.1752    60.76750     1.736374     1.853528
## 22            4    212.1400   113.40100     2.504749     1.641644
```

And ploted:

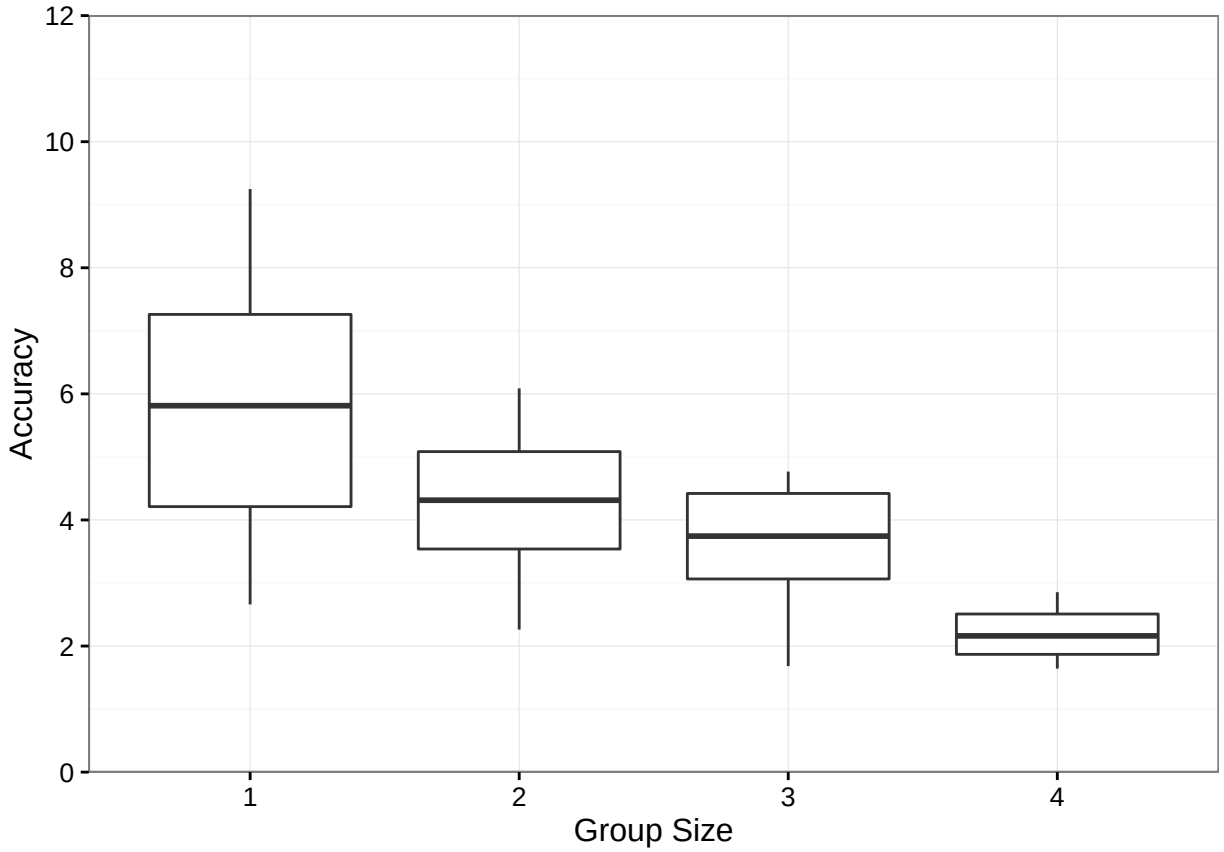
```
time <- gather(sourceDataGroups, "group", "time", 1:4)

ggplot(time, aes(x=group, y=time)) + geom_boxplot()+labs(x="Group Size", y = "Time (seconds)")+theme_bw
```

```
## Warning: Removed 4 rows containing non-finite values (stat_boxplot).
```




```
#boxplot(sourceDataGroups[5:8],xlab="Team members",ylab="Error",main="Error X Groups")
error <- gather(sourceDataGroups, "group", "error", 5:8)
ggplot(error, aes(x=group, y=error)) + geom_boxplot()+labs(x="Group Size", y = "Accuracy")+theme_bw()+ s
```

```
## Warning: Removed 4 rows containing non-finite values (stat_boxplot).
```




```
#variance <- gather(sourceDataGroups, "group", "variance", 9:12)
#ggplot(variance, aes(x=group, y=variance)) + geom_boxplot()+labs(title="Group size vs. Work Distributi
```

3.1.1 Correlation Analysis

The Pearson product-moment correlation coefficient is a measure of the linear correlation between two variables X and Y, giving a value between +1 and -1 inclusive, where 1 is total positive correlation, 0 is no correlation, and -1 is total negative correlation. It is widely used in the sciences as a measure of the degree of linear dependence between two variables.

First we need to take the average of time and error of each team:

```
#sourceDataTasks$MeanTime <- rowMeans(subset(sourceDataTasks, select = c(2,3,4,5)), na.rm = TRUE)
#sourceDataTasks$MeanError <- rowMeans(subset(sourceDataTasks, select = c(6,7,8,9)), na.rm = TRUE)
sourceDataTasks$MeanTime <- rowMeans(subset(sourceDataTasks, select = c(2,3)), na.rm = TRUE)
sourceDataTasks$MeanError <- rowMeans(subset(sourceDataTasks, select = c(4,5)), na.rm = TRUE)
```

Now we can make the pearson correlation analysis. It will tell us if there is a correlation between team members and time to solve the tasks, and team members and errors performed during the tasks.

First, the team members vs. time:

```
#teamTimeCorr <- sourceDataTasks[,c(1,10)]
teamTimeCorr <- sourceDataTasks[,c(1,6)]
teamTimeCorr <- rcorr(as.matrix(teamTimeCorr))
teamTimeCorr
```

```
##              Members.Task MeanTime
```

```
## Members.Task         1.00    -0.12
## MeanTime            -0.12     1.00
##
## n= 22
##
##
## P
##              Members.Task MeanTime
## Members.Task              0.5946
## MeanTime     0.5946
```

Team members vs. error:

```
#teamErrorCorr <- sourceDataTasks[,c(1,11)]
teamErrorCorr <- sourceDataTasks[,c(1,7)]
teamErrorCorr <- rcorr(as.matrix(teamErrorCorr))
teamErrorCorr
```

```
##              Members.Task MeanError
## Members.Task         1.00     -0.72
## MeanError           -0.72      1.00
##
## n= 22
##
##
## P
##              Members.Task MeanError
## Members.Task              2e-04
## MeanError    2e-04
```

As the is greater than 0.05, we assume that there is no correlation between team members and time to complete the tasks. In the other hand, is < 0.0001, so there is a strong correlation between team members and errors performed. At this point we don't know if the errors grow with more team members or in the other way around.

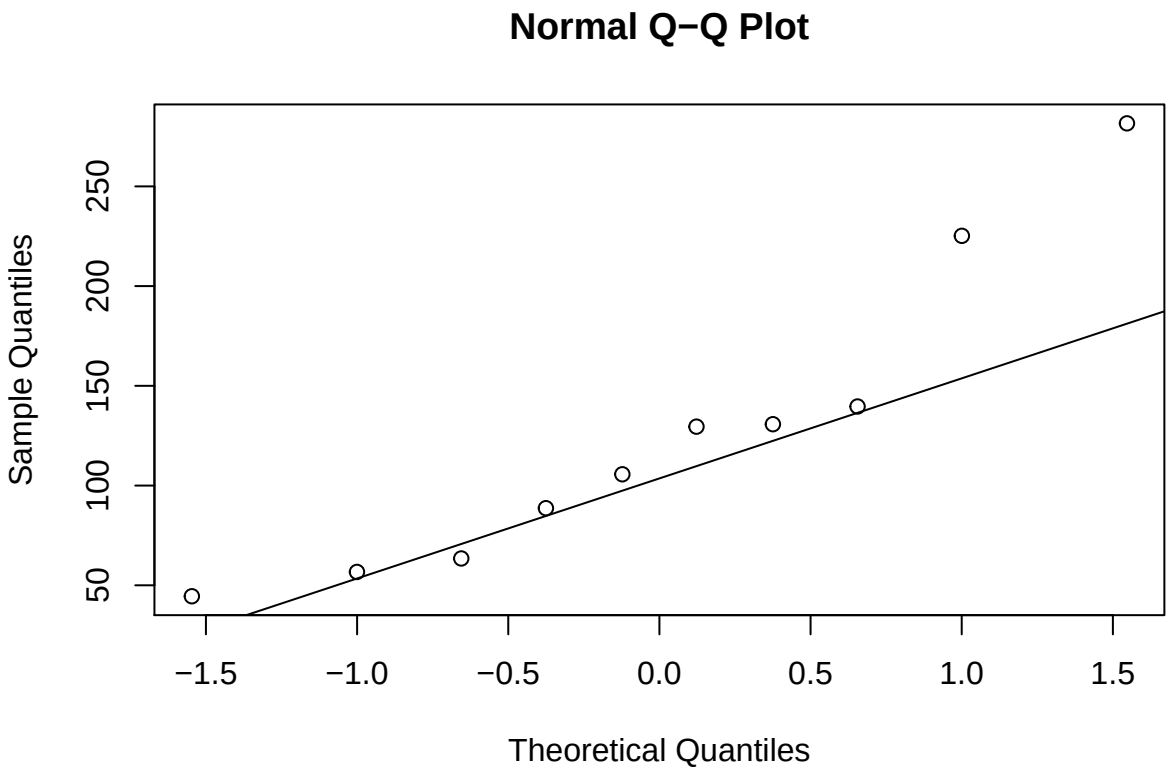
3.1.2 Shapiro-Wilk test

Before we conduce a variance test on the data to evaluate if more or less members cause more errors, we need to check if the data is normally distributed. For this test, we use the Shapiro-wilk test for each group of team members.

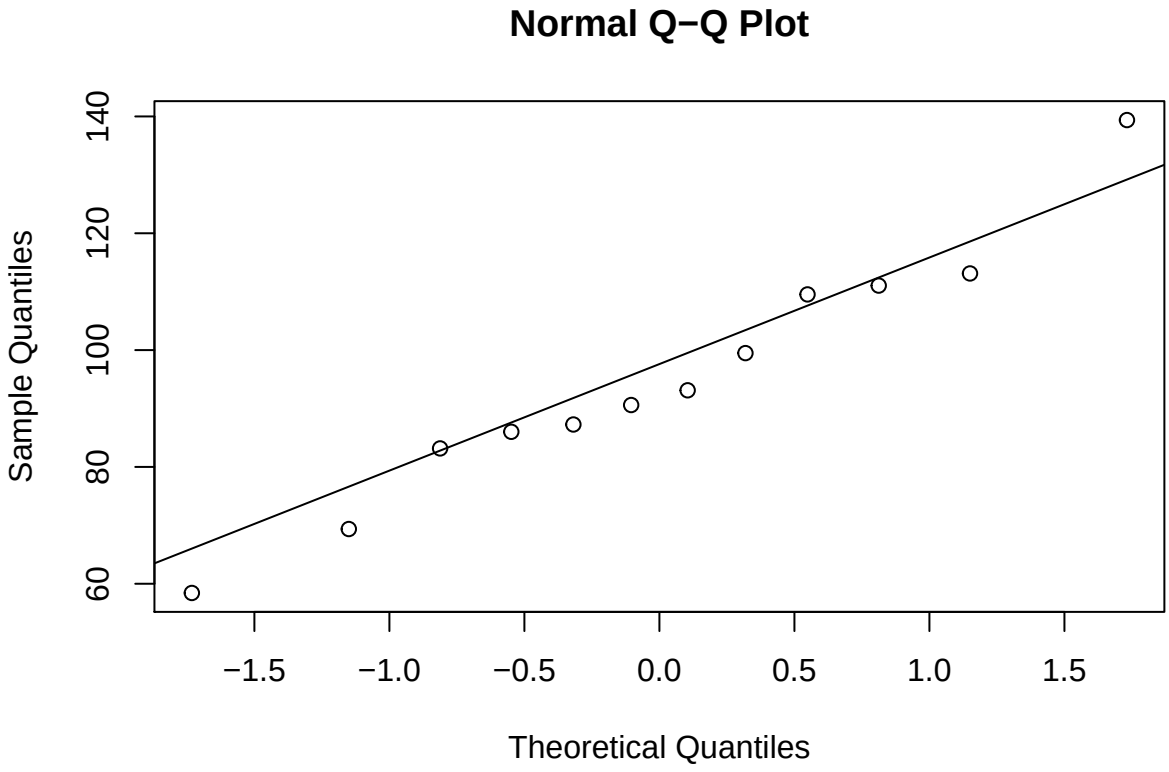
```
sapply(lapply(sourceDataGroups[1:8], shapiro.test), `[`, c("statistic","p.value"))
```

```
##           V1        V2        V3         V4         V5        V6
## statistic 0.8896474 0.9732635 0.8202519  0.8516173  0.9650849 0.9533945
## p.value   0.1680326 0.9418068 0.01606903 0.06071928 0.8419151 0.687011
##           V7        V8
## statistic 0.9272516 0.9160585
## p.value   0.3518895 0.3252503
```

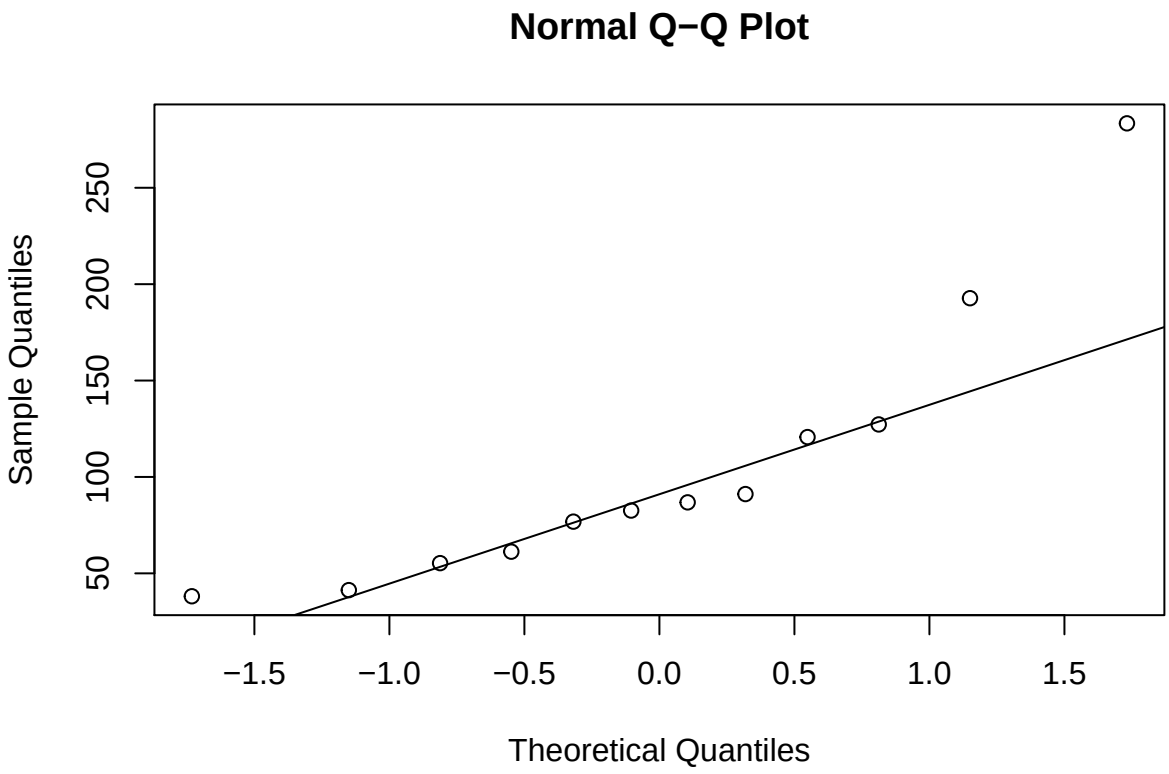
```
qqnorm(sourceDataGroups$V1)
qqline(sourceDataGroups$V1)
```



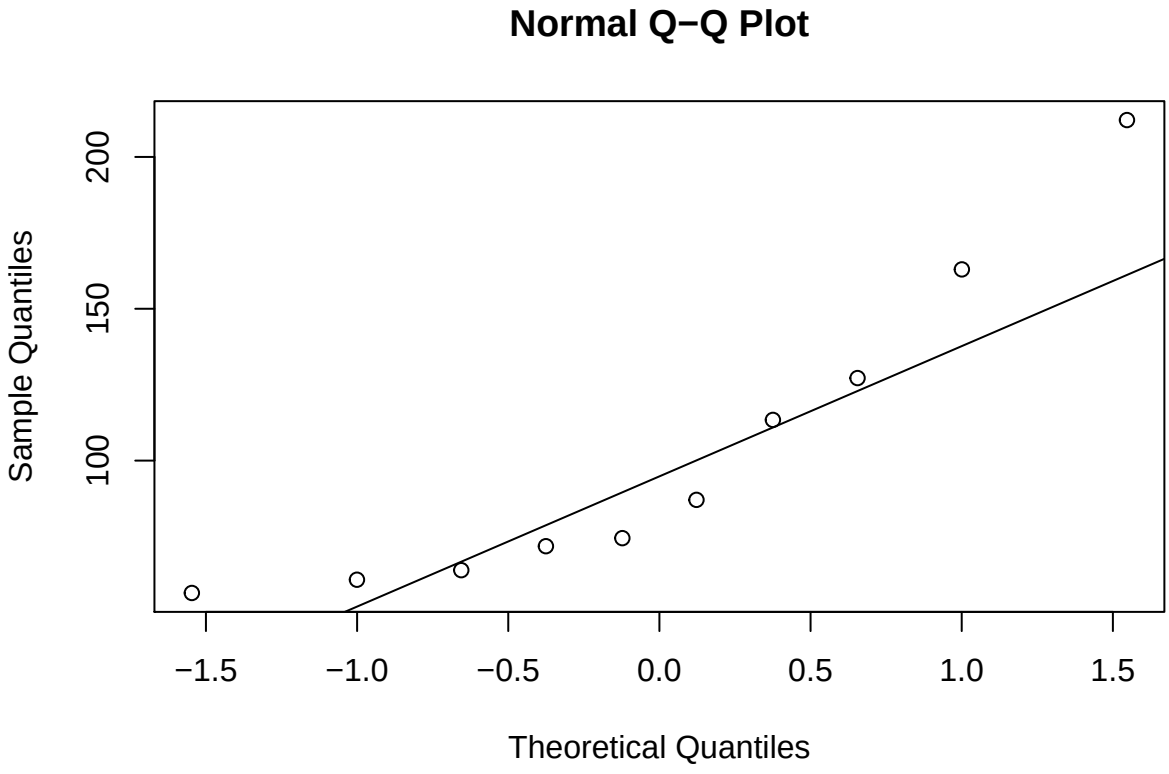

```
qqnorm(sourceDataGroups$V2)
qqline(sourceDataGroups$V2)
```



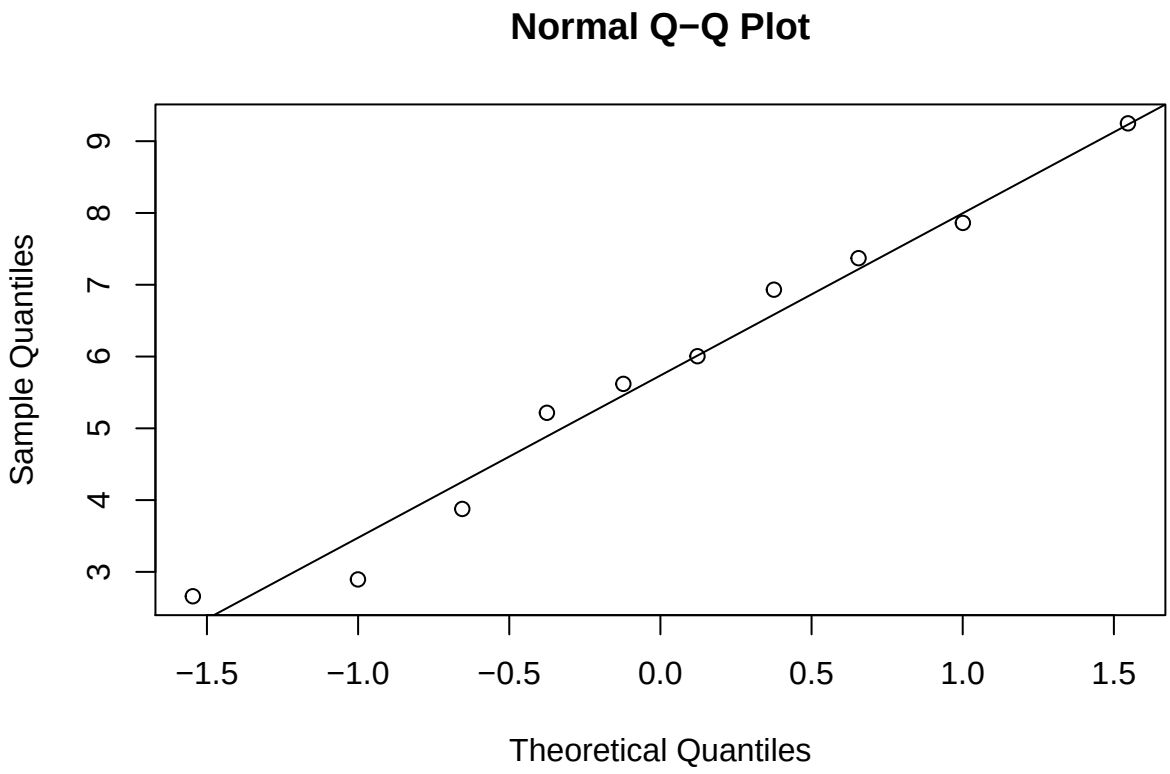

```
qqnorm(sourceDataGroups$V3)
qqline(sourceDataGroups$V3)
```



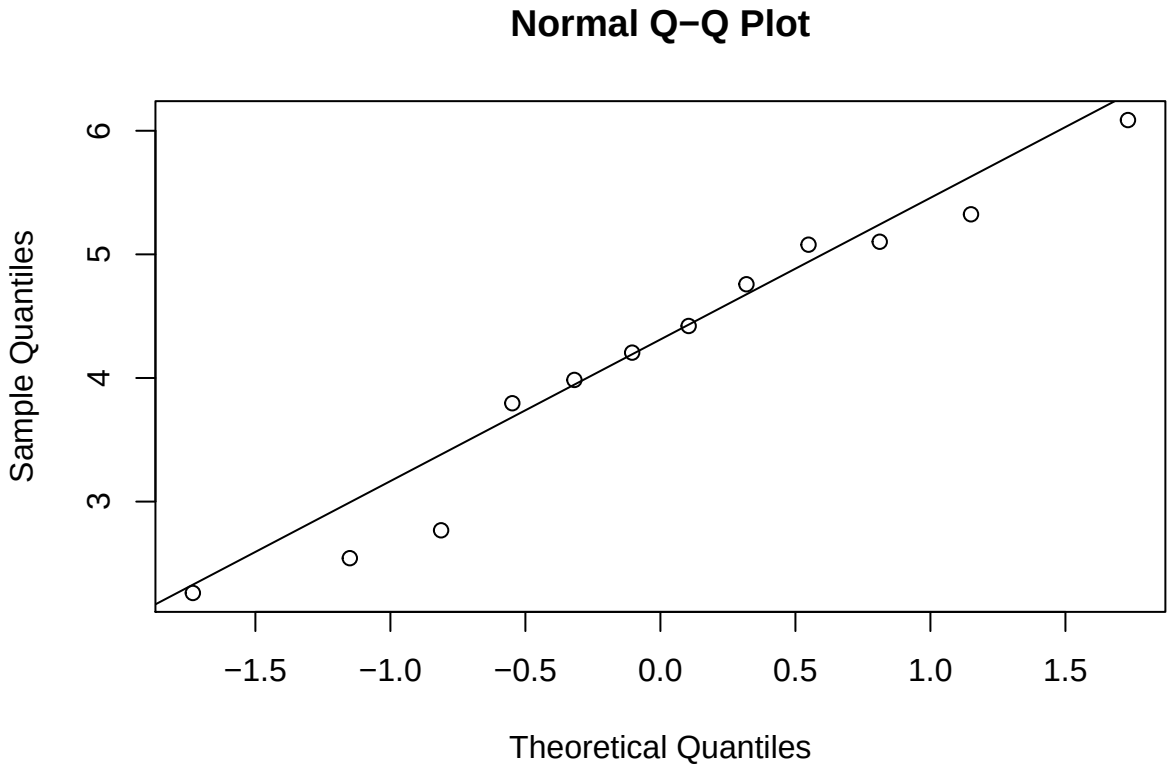

```
qqnorm(sourceDataGroups$V4)
qqline(sourceDataGroups$V4)
```



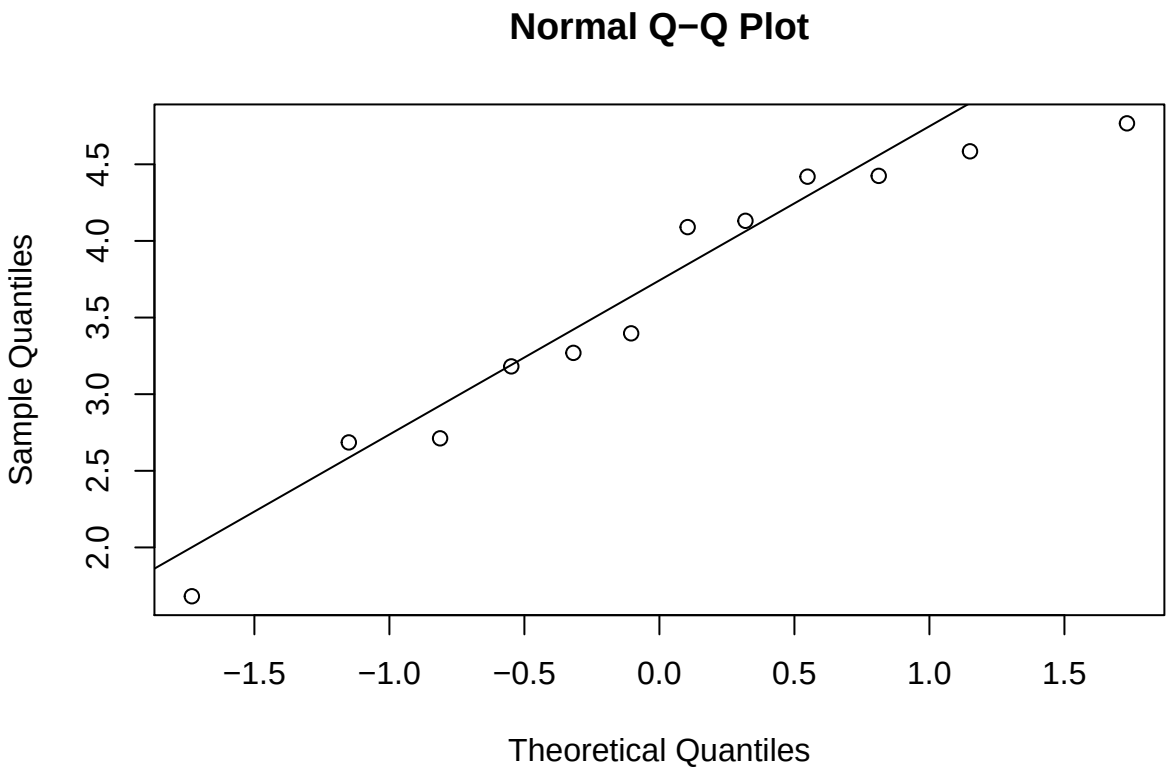

```
qqnorm(sourceDataGroups$V5)
qqline(sourceDataGroups$V5)
```



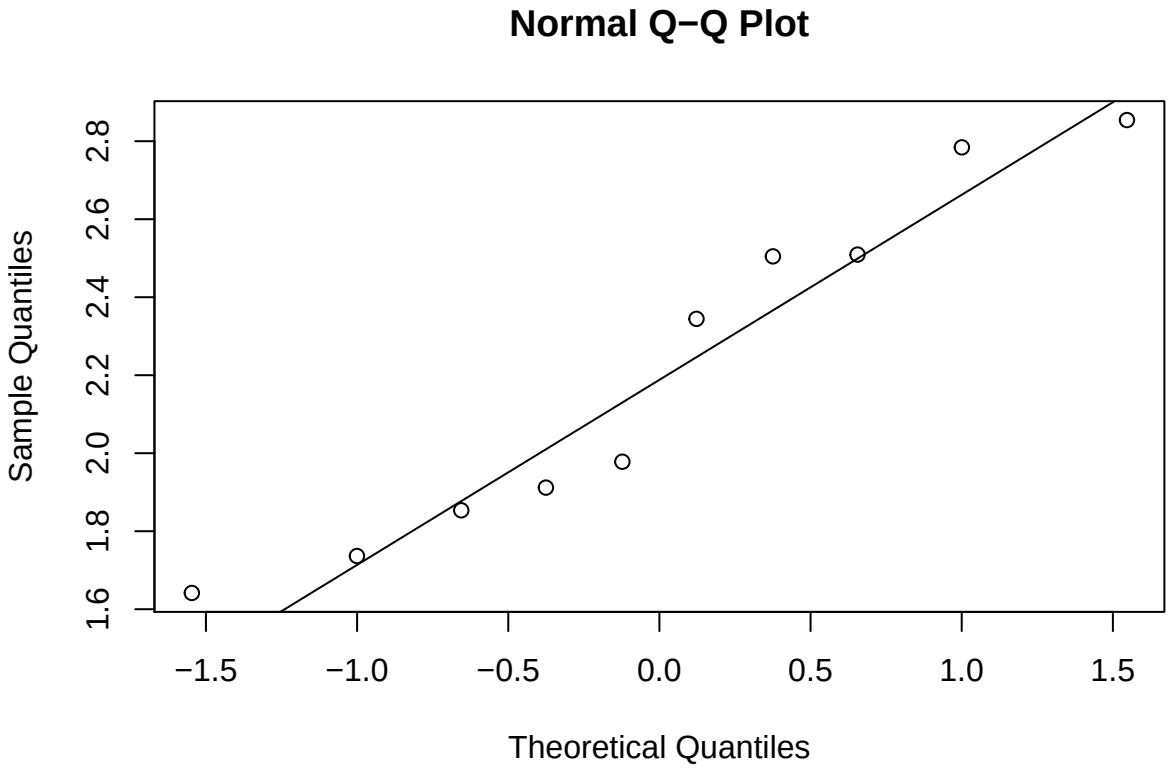

```
qqnorm(sourceDataGroups$V6)
qqline(sourceDataGroups$V6)
```




```
qqnorm(sourceDataGroups$V7)
qqline(sourceDataGroups$V7)
```




```
qqnorm(sourceDataGroups$V8)
qqline(sourceDataGroups$V8)
```



3.1.3 Kruskal-Wallis

We perform the Kruskal test with Dunn posthoc with the time data:

```
library(dunn.test)
dunn.test(error$error,error$group, kw=TRUE, method="holm")
```

```
##   Kruskal-Wallis rank sum test
##
## data: x and group
## Kruskal-Wallis chi-squared = 21.3522, df = 3, p-value = 0
##
##
##                             Comparison of x by group
##                                      (Holm)
## Col Mean-|
## Row Mean |         V5         V6         V7
## ---------+---------------------------------
##       V6 |   1.366666
##          |     0.1717
##          |
##       V7 |   2.139393   0.810443
##          |     0.0486     0.2088
##          |
##       V8 |   4.473795   3.306060   2.533333
##          |    0.0000*    0.0024*    0.0226*
```

... and with the error data:

```
dunn.test(time$time,time$group, kw=TRUE, method="holm")
```

```
##   Kruskal-Wallis rank sum test
##
## data: x and group
## Kruskal-Wallis chi-squared = 2.3449, df = 3, p-value = 0.5
##
##
##                             Comparison of x by group
##                                      (Holm)
## Col Mean-|
## Row Mean |         V1         V2         V3
## ---------+---------------------------------
##       V2 |  -1.506060
##          |     0.3962
##          |
##       V3 |  -1.006060   0.524404
##          |     0.7860     0.6000
##          |
##       V4 |  -0.713718   0.760606   0.260606
##          |     0.7131     0.8938     0.3972
```

```
4 {r dunn3 posthoc} #dunn.test(variance$variance,variance$group,
  kw=TRUE, method="holm") #

5 {r dunn4 posthoc} #variance2 <- gather(sourceDataGroups,
  "group", "variance2", 10:12) #dunn.test(variance2$variance2,variance2
  kw=TRUE, method="holm") #
```

6 Comparison between tasks

6.1 Hypotesis

- H1. Task 1 and 2 are easier than 3 and 4;
- H2. Task 4 is performed with less errors than task 3;
- H3. Task 4 is performed in less time than task 3;

If H2 or/and H3 is confirmed, we can say that the teams have improved their performance with only one trainning.

6.2 Summary

The data is arranged by Task vs. Team members. Collumns are organized as follows:

- Members: Number of users in the team;
- T1, T2, T3, T4: Time to complete task 1 to 4;
- E1, E2, E3, E4: Errors in task 1 to 4.

Below the data is summarized:

```
describe(sourceDataTasks[2:3])
```

```
## sourceDataTasks[2:3]
##
##  2  Variables      22  Observations
## ---------------------------------------------------------------------------
## Time.task.3
##       n missing  unique    Info    Mean     .05     .10     .25     .50
##      22       0      22       1   120.6   45.10   57.70   79.12  107.59
##     .75     .90     .95
##  130.45  207.22  278.12
##
## lowest :  41.30  44.51  56.40  69.36  74.45
## highest: 139.38 162.96 212.14 281.60 283.41
## ---------------------------------------------------------------------------
## Time.task.4
##       n missing  unique    Info    Mean     .05     .10     .25     .50
##      22       0      22       1   92.78   55.36   56.87   61.79   85.12
##     .75     .90     .95
##  106.56  137.01  190.07
##
## lowest :  38.12  55.29  56.70  58.42  60.77
## highest: 113.12 113.40 139.63 192.72 225.21
## ---------------------------------------------------------------------------
```

```
describe(sourceDataTasks[4:5])
```

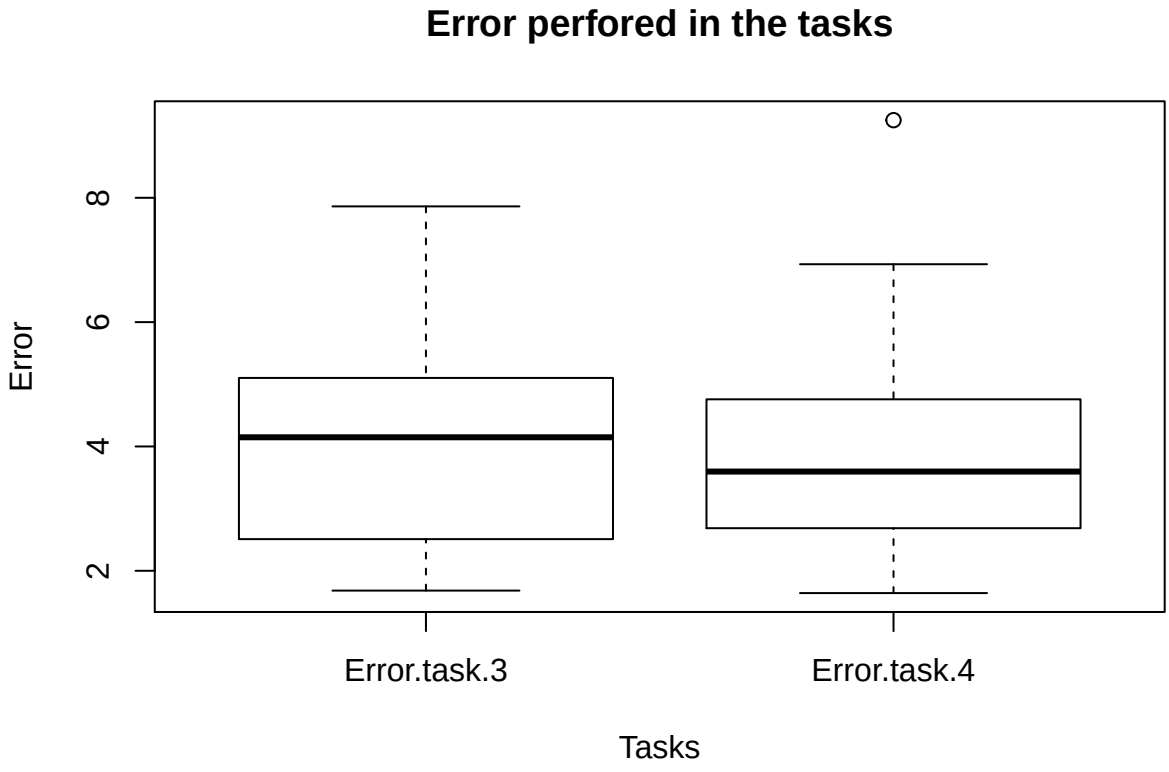
```
## sourceDataTasks[4:5]
##
##  2  Variables      22  Observations
## ---------------------------------------------------------------------------
## Error.task.3
##       n missing  unique    Info    Mean     .05     .10     .25     .50
##      22       0      22       1   4.021   1.763   2.269   2.547   4.148
##     .75     .90     .95
##   5.096   5.966   7.302
##
## lowest : 1.681 1.736 2.261 2.345 2.505
## highest: 5.324 5.619 6.005 7.370 7.861
## ---------------------------------------------------------------------------
## Error.task.4
##       n missing  unique    Info    Mean     .05     .10     .25     .50
##      22       0      22       1   3.864   1.856   1.918   2.706   3.596
##     .75     .90     .95
##   4.715   5.999   6.889
##
## lowest : 1.642 1.854 1.912 1.978 2.542
## highest: 4.767 5.216 6.086 6.931 9.248
## ---------------------------------------------------------------------------
```

6.2.1 Plots

Time of task completion vs. Task for all combinations of teams: The code used to generate the charts is:

```
boxplot(sourceDataTasks[2:3],xlab="Tasks",ylab="Time",main="Time to complete the tasks")
```


```
boxplot(sourceDataTasks[4:5],xlab="Tasks",ylab="Error",main="Error perfored in the tasks")
```



6.3 Analysis of time of completion per task

6.3.1 Shapiro

First, we perform the Shapiro normality test. This test determine if the data is normally distributed. It is important to determine if the data is normally distributed to conduce posterior tests.

```
sapply(lapply(sourceDataTasks[2:5], shapiro.test), `[`, c("statistic","p.value"))
```

```
##           Time.task.3 Time.task.4 Error.task.3 Error.task.4
## statistic 0.8505394   0.8207751   0.9354413    0.8974343
## p.value   0.003470118 0.001077677 0.1592539    0.02642102
```

As we can see, the $p - value$ of most Shapiro tests reveled that the data are not normally distributed. Since in this test the comparisons are made with the same subjects and we are varying the tasks, the next step is to perform a Friedman analysis.

6.3.2 Analysis of learning between tasks 3 and 4 (Wilcoxon signed rank test)

The hypotheses for the comparison across repeated measures are:

- H0: The distributions (whatever they are) are the same across repeated measures
- H1: The distributions across repeated measures are different

```
wilcox.test(sourceDataTasks$Time.task.3,sourceDataTasks$Time.task.4,paired=TRUE)
```

```
##
##  Wilcoxon signed rank test
##
## data:  sourceDataTasks$Time.task.3 and sourceDataTasks$Time.task.4
## V = 214, p-value = 0.003239
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(sourceDataTasks$Error.task.3,sourceDataTasks$Error.task.4,paired=TRUE)
```

```
##
##  Wilcoxon signed rank test
##
## data:  sourceDataTasks$Error.task.3 and sourceDataTasks$Error.task.4
## V = 144, p-value = 0.5879
## alternative hypothesis: true location shift is not equal to 0
```

summary(tasksTime) tasksTime <- gather(sourceDataTasks, “task”, “time”, 2:3) ggplot(tasksTime, aes(x=as.character(task), y=time)) + geom_boxplot()+labs(x=“Tasks”, y = “Time (seconds)”)+theme_bw()+scale_x_discret “Time.task.4”),labels=c(“First Trial”, “Second Trial”))+ scale_y_continuous(limits = c(0, 300), expand = c(0, 0),breaks = c(0,50,100,150,200,250,300))

summary(tasksError) tasksError <- gather(sourceDataTasks, “task”, “error”, 4:5) ggplot(tasksError, aes(x=as.character(task), y=error)) + geom_boxplot()+labs(x=“Tasks”, y = “Error”)+theme_bw()+scale_x_discrete(breaks= “Error.task.4”),labels=c(“First Trial”, “Second Trial”))+ scale_y_continuous(limits = c(0, 9), expand = c(0, 0),breaks = c(0,1.5,3,4.5,6,7.5,9))

6.4 Analysis of user role change

6.4.1 Correlation between Groups and user role change

```
#sourceDataRoles <- read.csv("rolesPerTeam.csv",header = FALSE,  sep=";") #it is not working, but group
sourceDataRoles <- read.csv("groupsRoles.csv",header = FALSE,  sep="\t")
sourceDataRoles
```

```
##     V1 V2
## 1    1 21
## 2    1 24
## 3    1 32
## 4    1 49
## 5    1 19
## 6    1 21
## 7    1 23
## 8    1 21
## 9    1 45
## 10   1 33
## 11   2 17
## 12   2 24
## 13   2 15
## 14   2 15
## 15   2 14
## 16   2 19
## 17   2 41
## 18   2 11
## 19   2 17
## 20   2 29
## 21   2 11
## 22   2 38
## 23   2  9
## 24   2 14
## 25   2 37
## 26   2 19
## 27   3  8
## 28   3 34
## 29   3 19
## 30   3 18
## 31   3 14
## 32   3  4
## 33   3  9
## 34   3  5
## 35   3 11
## 36   3 15
## 37   3  9
## 38   3  7
## 39   3 20
## 40   3 20
## 41   3  1
## 42   3  1
## 43   3 10
## 44   3  7
## 45   3 12
```

```
## 46   3 18
## 47   3  9
## 48   3  8
## 49   3  9
## 50   3  1
## 51   3  8
## 52   3  5
## 53   3 12
## 54   3 12
## 55   3 19
## 56   3  3
## 57   3 17
## 58   3 21
## 59   3  1
## 60   3  3
## 61   3  9
## 62   3  7
## 63   4 18
## 64   4 13
## 65   4  4
## 66   4  7
## 67   4  2
## 68   4  7
## 69   4  4
## 70   4  8
## 71   4 14
## 72   4 11
## 73   4 17
## 74   4  3
## 75   4  1
## 76   4  4
## 77   4 15
## 78   4  3
## 79   4  7
## 80   4  3
## 81   4  3
## 82   4  4
## 83   4  9
## 84   4 15
## 85   4 20
## 86   4 11
## 87   4  6
## 88   4  1
## 89   4  3
## 90   4 10
## 91   4  7
## 92   4 23
## 93   4 22
## 94   4  9
## 95   4  1
## 96   4  1
## 97   4 10
## 98   4  1
## 99   4  1
```

```
## 100  4  4
## 101  4  5
## 102  4  1
```

```
sourceGroupsErrorRoles <- read.csv("members_error_roles.csv",header = FALSE,  sep=";")
sourceGroupsErrorRoles
```

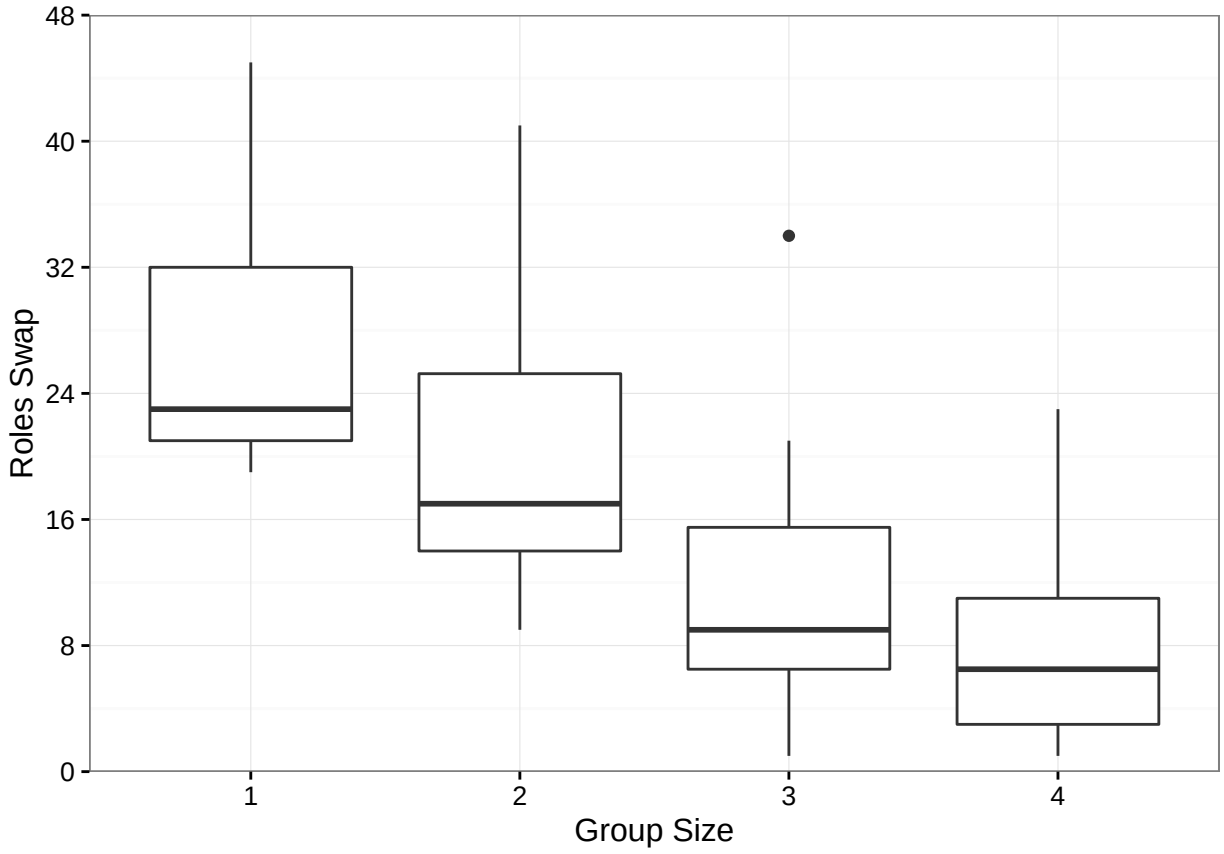
```
##     V1       V2 V3
## 1    1 9.070060 21
## 2    1 7.381541 24
## 3    1 6.207668 32
## 4    1 3.341505 49
## 5    1 6.817599 19
## 6    2 3.251326 17
## 7    2 3.251326 24
## 8    2 4.849439 15
## 9    2 4.849439 15
## 10   2 4.726371 14
## 11   2 4.726371 19
## 12   2 4.699787 41
## 13   2 4.699787 11
## 14   3 2.972291  8
## 15   3 2.972291 34
## 16   3 2.972291 19
## 17   3 3.978512 18
## 18   3 3.978512 14
## 19   3 3.978512  4
## 20   3 4.150805  9
## 21   3 4.150805  5
## 22   3 4.150805 11
## 23   3 3.179116 15
## 24   3 3.179116  9
## 25   3 3.179116  7
## 26   3 2.652203 20
## 27   3 2.652203 20
## 28   3 2.652203  1
## 29   3 4.778647  1
## 30   3 4.778647 10
## 31   3 4.778647  7
## 32   4 2.389965 18
## 33   4 2.389965 13
## 34   4 2.389965  4
## 35   4 2.389965  7
## 36   4 3.337539  2
## 37   4 3.337539  7
## 38   4 3.337539  4
## 39   4 3.337539  8
## 40   4 3.023767 14
## 41   4 3.023767 11
## 42   4 3.023767 17
## 43   4 3.023767  3
## 44   4 1.776181  1
## 45   4 1.776181  4
## 46   4 1.776181 15
## 47   4 1.776181  3
```

```
## 48   4 2.376898  7
## 49   4 2.376898  3
## 50   4 2.376898  3
## 51   4 2.376898  4
## 52   1 8.529819 21
## 53   1 5.705525 23
## 54   1 5.088869 21
## 55   1 3.299659 45
## 56   1 8.455383 33
## 57   2 4.512303 17
## 58   2 4.512303 29
## 59   2 3.707839 11
## 60   2 3.707839 38
## 61   2 3.707839  9
## 62   2 3.707839 14
## 63   2 4.160006 37
## 64   2 4.160006 19
## 65   3 3.701374 12
## 66   3 3.701374 18
## 67   3 3.701374  9
## 68   3 3.053629  8
## 69   3 3.053629  9
## 70   3 3.053629  1
## 71   3 4.628316  8
## 72   3 4.628316  5
## 73   3 4.628316 12
## 74   3 4.247869 12
## 75   3 4.247869 19
## 76   3 4.247869  3
## 77   3 2.953497 17
## 78   3 2.953497 21
## 79   3 2.953497  1
## 80   3 4.583976  3
## 81   3 4.583976  9
## 82   3 4.583976  7
## 83   4 2.110792  9
## 84   4 2.110792 15
## 85   4 2.110792 20
## 86   4 2.110792 11
## 87   4 2.288988  6
## 88   4 2.288988  1
## 89   4 2.288988  3
## 90   4 2.288988 10
## 91   4 3.257560  7
## 92   4 3.257560 23
## 93   4 3.257560 22
## 94   4 3.257560  9
## 95   4 1.903549  1
## 96   4 1.903549  1
## 97   4 1.903549 10
## 98   4 1.903549  1
## 99   4 1.918039  1
## 100  4 1.918039  4
## 101  4 1.918039  5
```

```
## 102  4 1.918039  1
```

```
#rolesGroups <- gather(sourceDataRoles, "group", "roles", 1:4)
#rolesGroups <- rolesGroups[complete.cases(rolesGroups),]
#ggplot(rolesGroups, aes(x=group, y=roles)) + geom_boxplot()+labs(title="Group size vs. Team members ro
#dunn.test(rolesGroups$roles,rolesGroups$group, kw = TRUE, method="holm")

ggplot(sourceDataRoles, aes(x=as.character(V1), y=V2)) + geom_boxplot()+labs(x="Group Size", y = "Roles
```

```
## Warning: Removed 1 rows containing non-finite values (stat_boxplot).
```




```
dunn.test(sourceDataRoles$V2,sourceDataRoles$V1, kw = TRUE, method="holm")
```

```
##   Kruskal-Wallis rank sum test
##
## data: x and group
## Kruskal-Wallis chi-squared = 40.1615, df = 3, p-value = 0
##
##
##                            Comparison of x by group
##                                    (Holm)
## Col Mean-|
## Row Mean |          1          2          3
## ---------+----------------------------------
##        2 |   1.410779
##          |     0.0792
##          |
##        3 |   4.182047   3.082619
```

```
##           |    0.0001*    0.0031*
##           |
##         4 |   5.329473   4.447370   1.694733
##           |    0.0000*    0.0000*     0.0901
```

```
sapply(lapply(sourceGroupsErrorRoles[2:3], shapiro.test), `[`, c("statistic","p.value"))
```

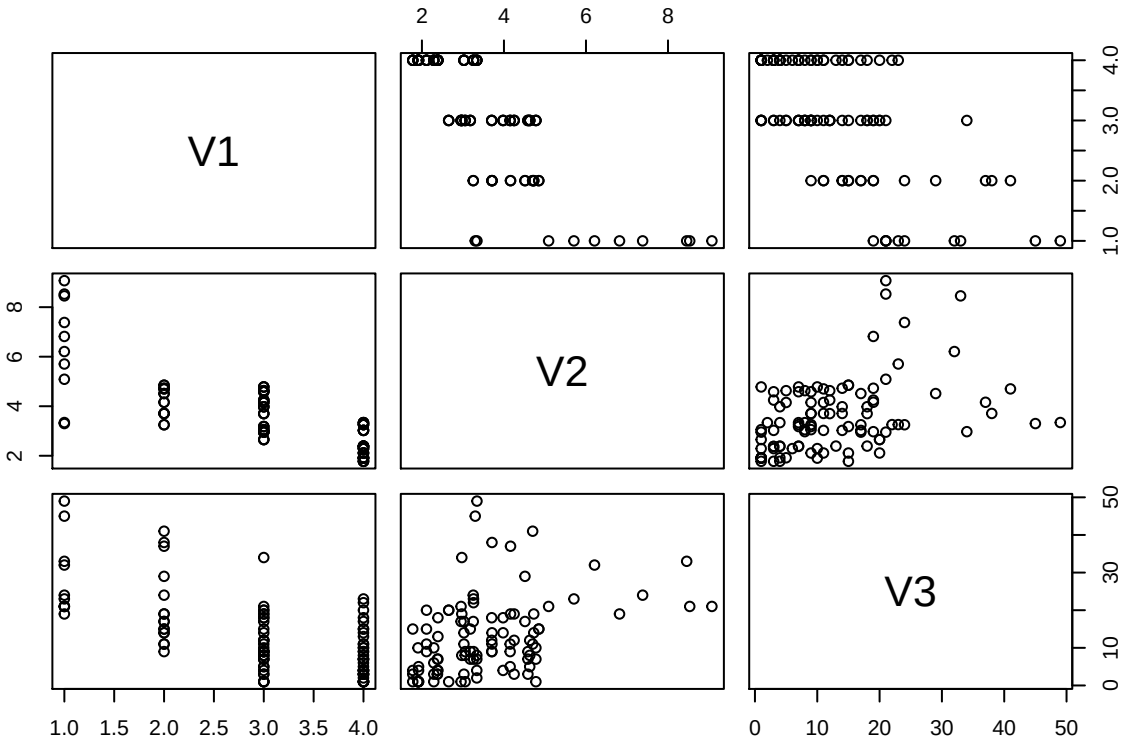
```
##           V2           V3
## statistic 0.8709327    0.8906971
## p.value   6.068221e-08 4.276405e-07
```

```
cor.test(sourceGroupsErrorRoles$V2,sourceGroupsErrorRoles$V3, alternative = "greater",method = "spearma
```

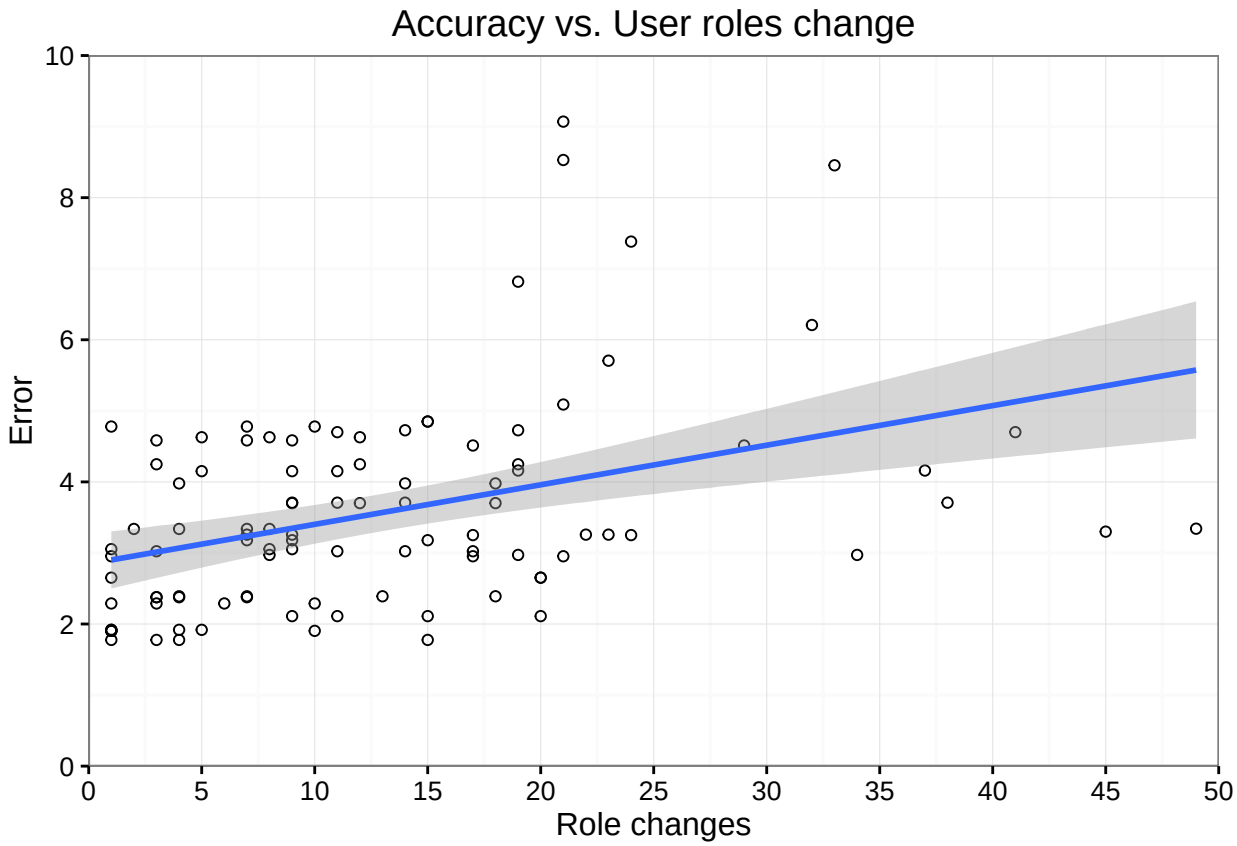
```
## Warning in cor.test.default(sourceGroupsErrorRoles$V2,
## sourceGroupsErrorRoles$V3, : Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  sourceGroupsErrorRoles$V2 and sourceGroupsErrorRoles$V3
## S = 99854, p-value = 2.402e-06
## alternative hypothesis: true rho is greater than 0
## sample estimates:
##       rho
## 0.4353777
```

```
pairs(sourceGroupsErrorRoles)
```




```
lm_out <- lm(V2~V3, data=sourceGroupsErrorRoles)
ggplot(sourceGroupsErrorRoles, aes(x=V3, y=V2)) + geom_point(shape=1) + geom_smooth(method=lm) +labs(ti
```


Accuracy vs. User roles change
Error
Role changes
0
2
4
6
8
10
0
5
10
15
20
25
30
35
40
45
50