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("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("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
```

3.2 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$MeanTime <- rowMeans(subset(sourceDataTasks, select = c(2,3)), na.rm = TRUE)
sourceDataTasks$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.

The team members vs. time:

```
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(1,7)]
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 time 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, accuracy is < 0.0001, so there is a strong correlation between team members and errors performed. We have to conduct further analysis to understand the behavior of the accuracy increase. At this point we don't know if the errors grow with more team members or in the other way around.

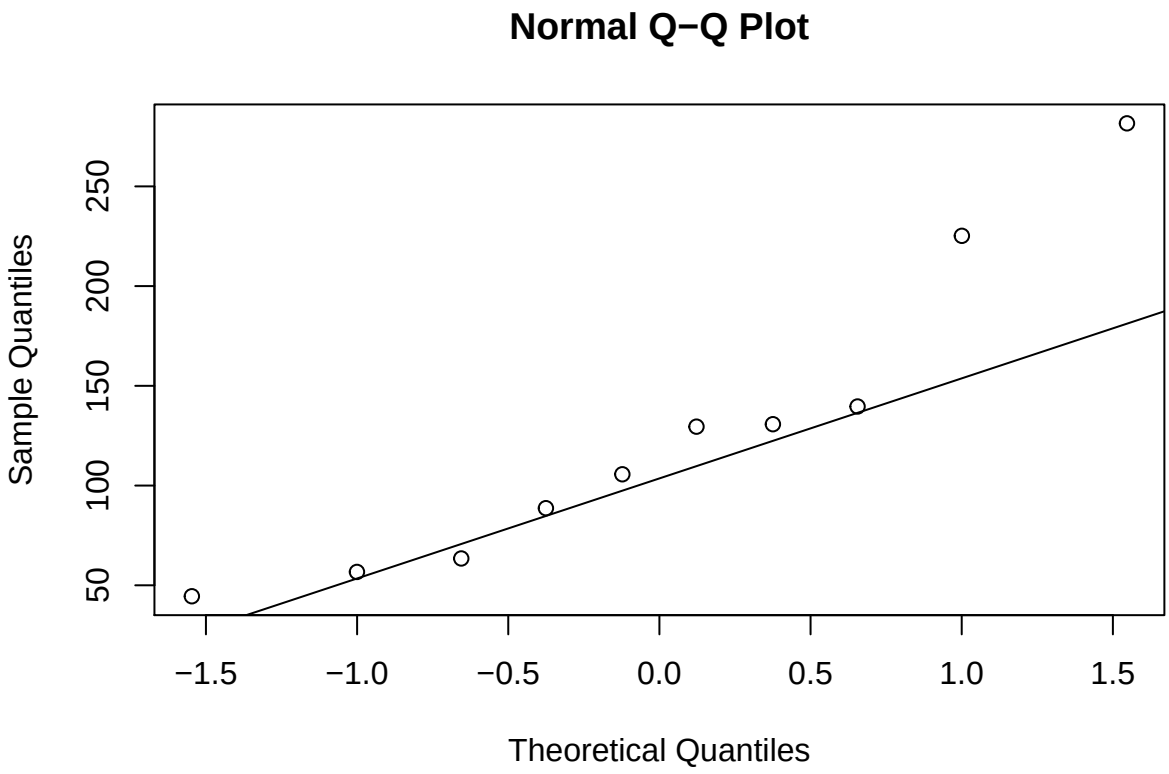
3.3 Shapiro-Wilk test

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.

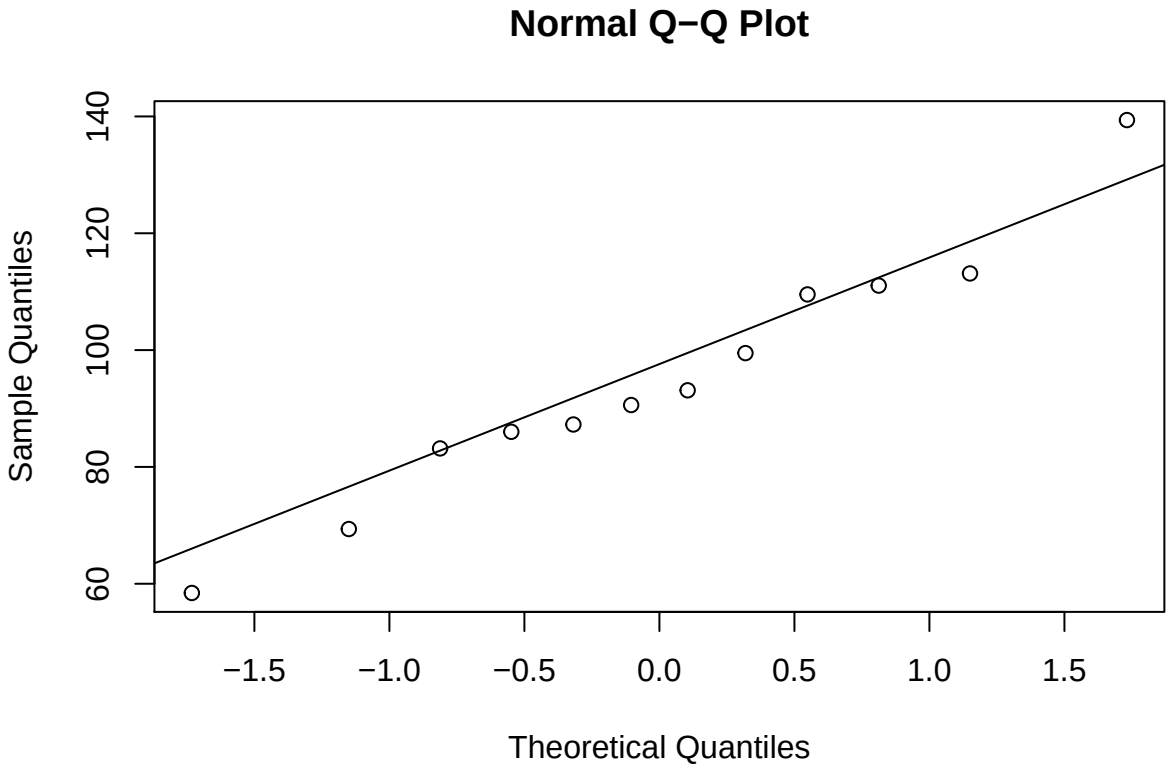
```
sapply(lapply(sourceDataGroups[1:8], shapiro.test), `[`, c("statistic","p.value"))
```

```
##           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
```

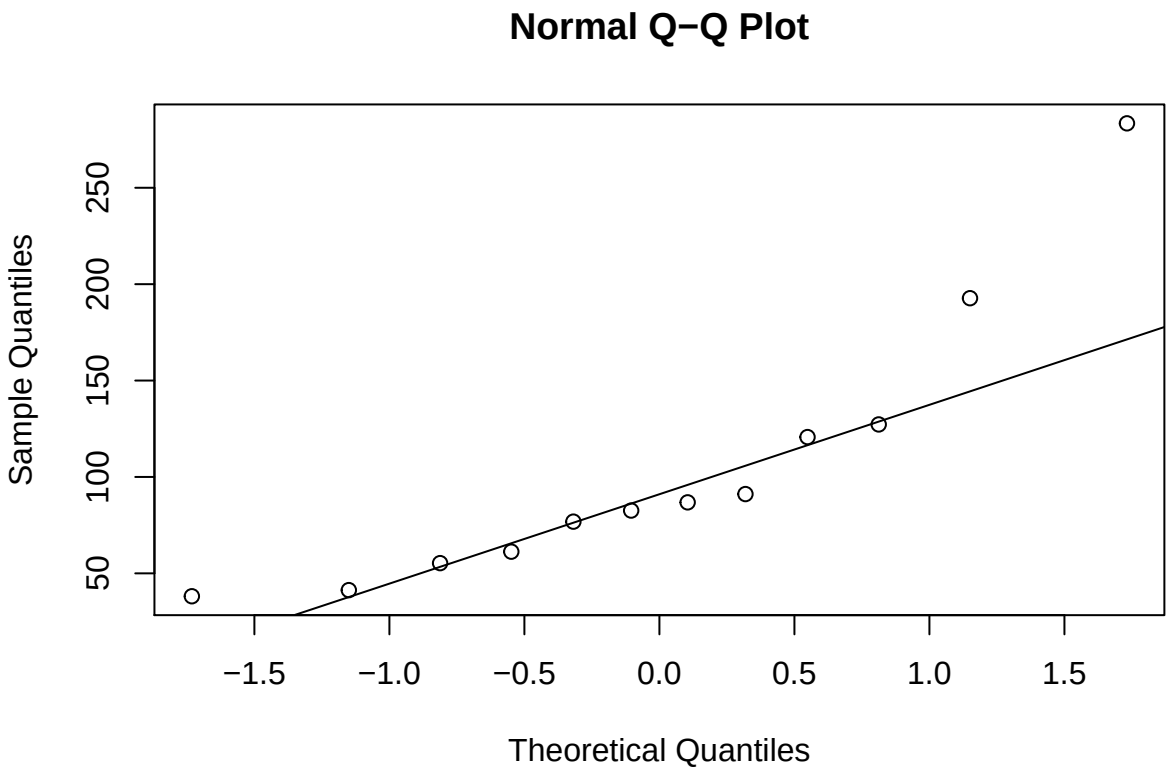
```
qqnorm(sourceDataGroups$V1)
qqline(sourceDataGroups$V1)
```



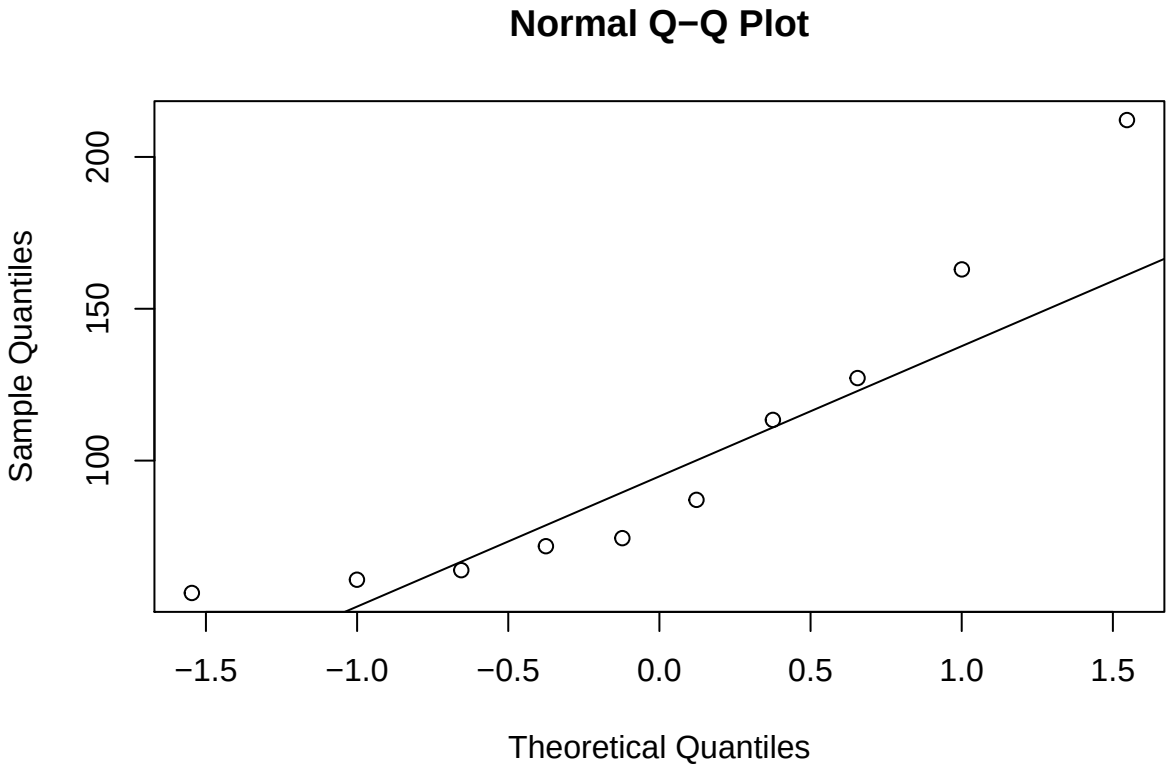

```
qqnorm(sourceDataGroups$V2)
qqline(sourceDataGroups$V2)
```



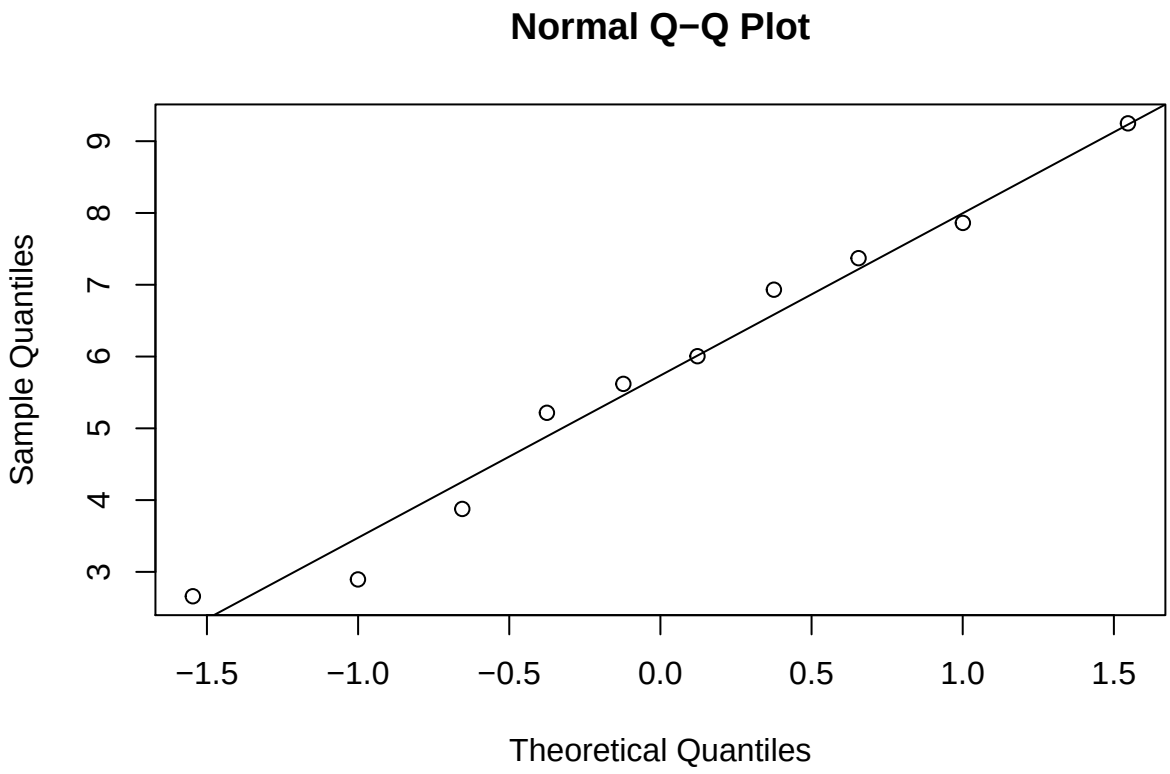

```
qqnorm(sourceDataGroups$V3)
qqline(sourceDataGroups$V3)
```



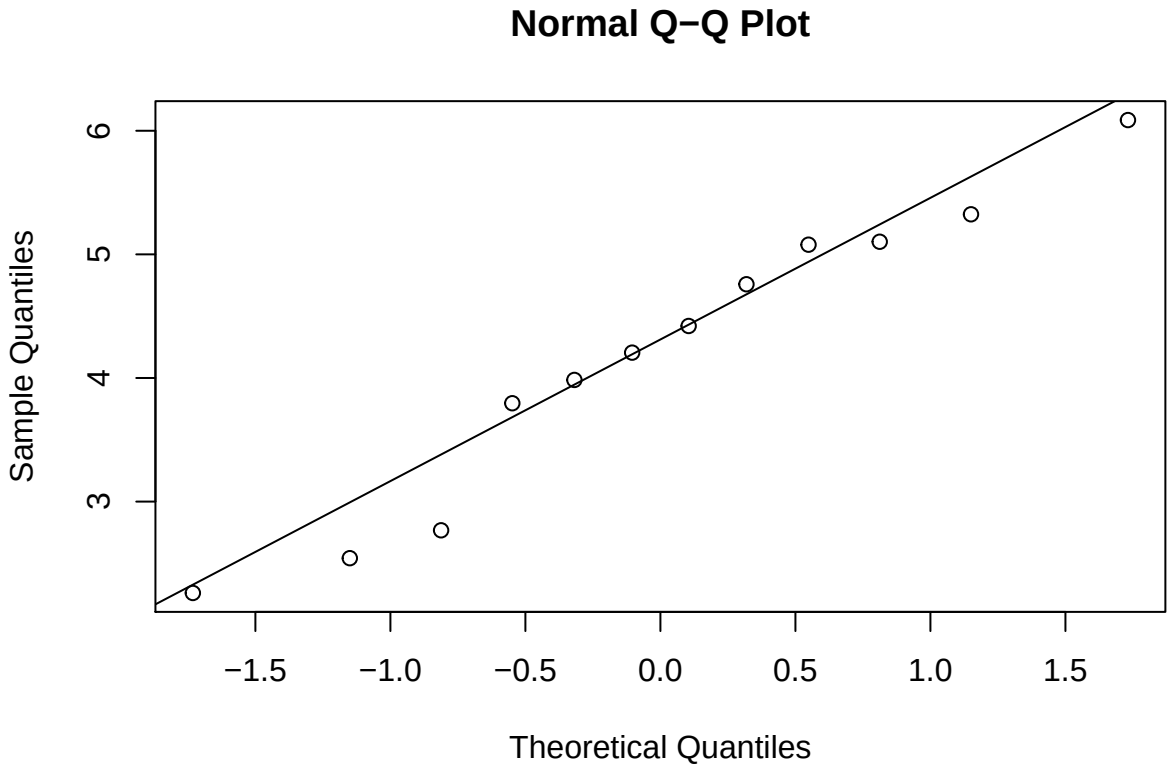

```
qqnorm(sourceDataGroups$V4)
qqline(sourceDataGroups$V4)
```



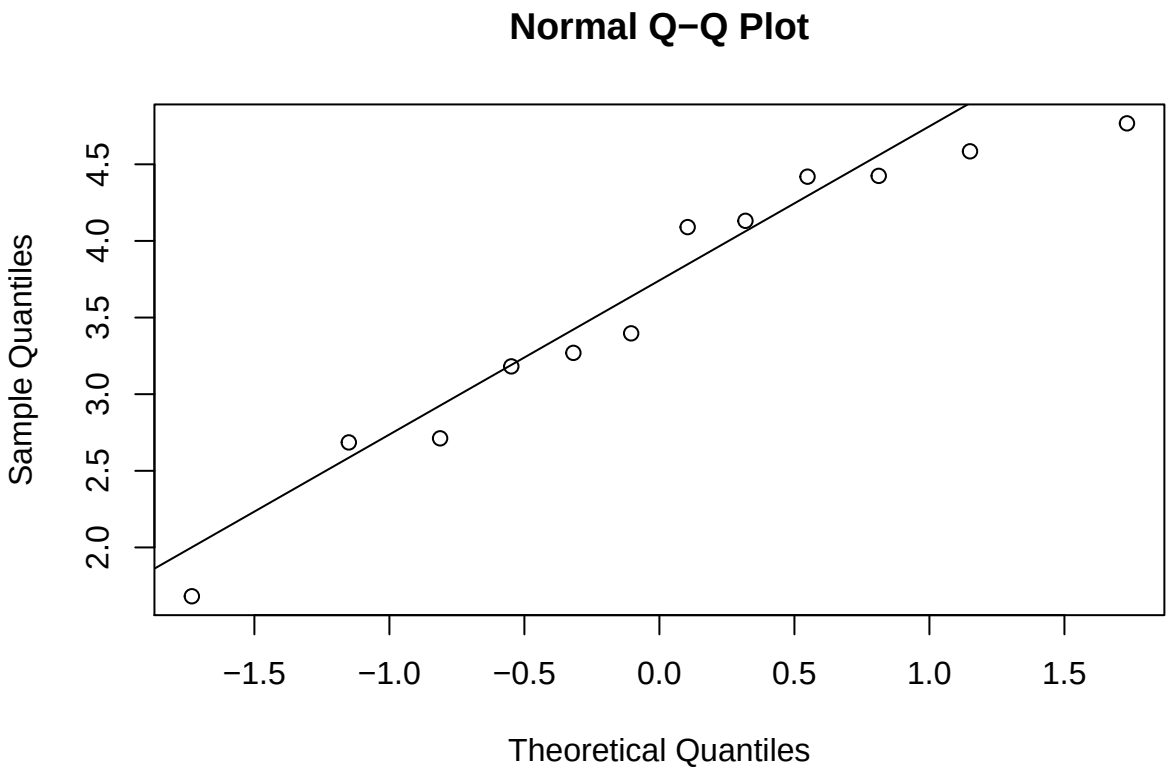

```
qqnorm(sourceDataGroups$V5)
qqline(sourceDataGroups$V5)
```



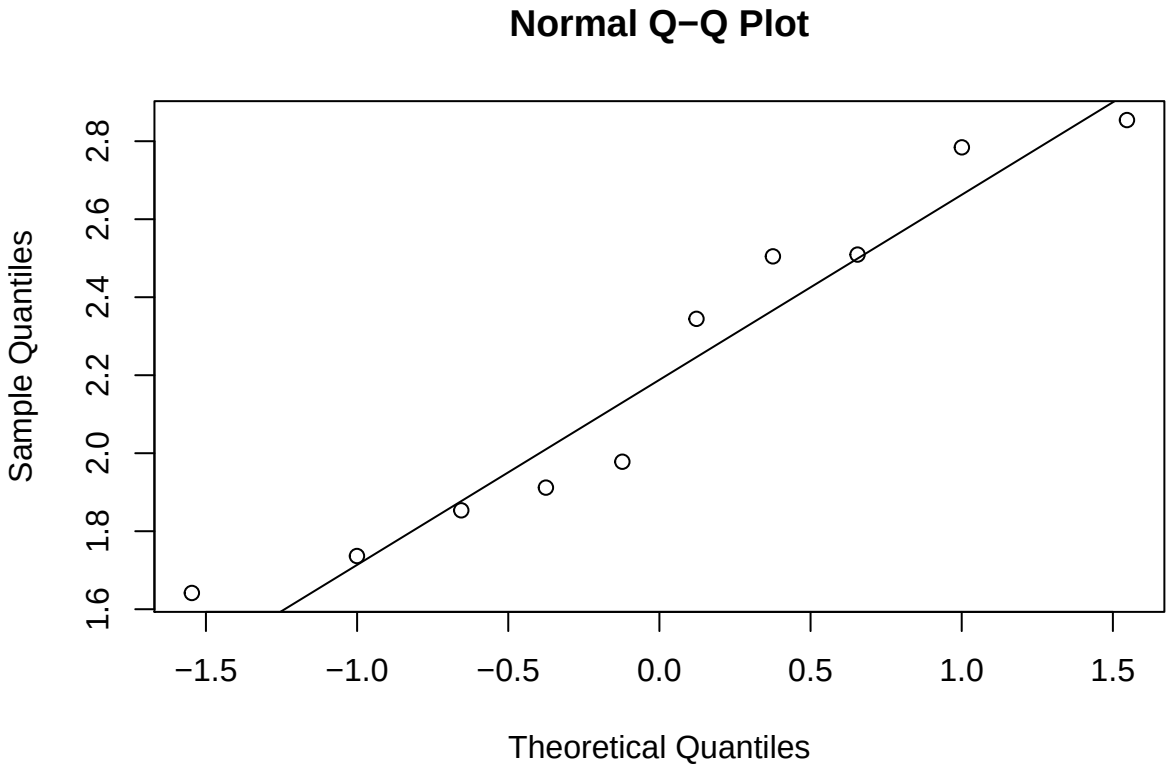

```
qqnorm(sourceDataGroups$V6)
qqline(sourceDataGroups$V6)
```




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




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

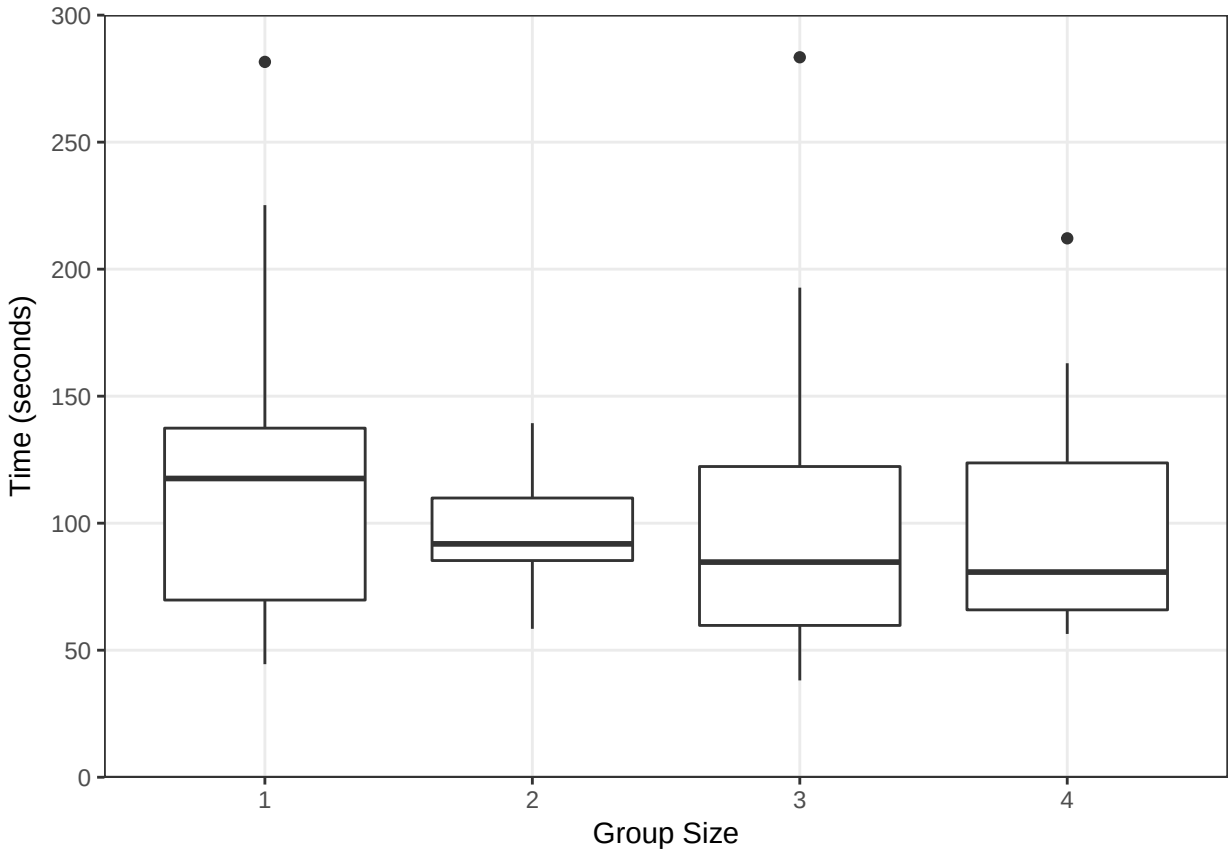


3.4 Group Size vs. Task Completion Time

We plotted the chart with group size vs. Time and performed the Kruskal test with paired Dunn tests with Holm-Bonferroni correction:

```
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).
```




```
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
```

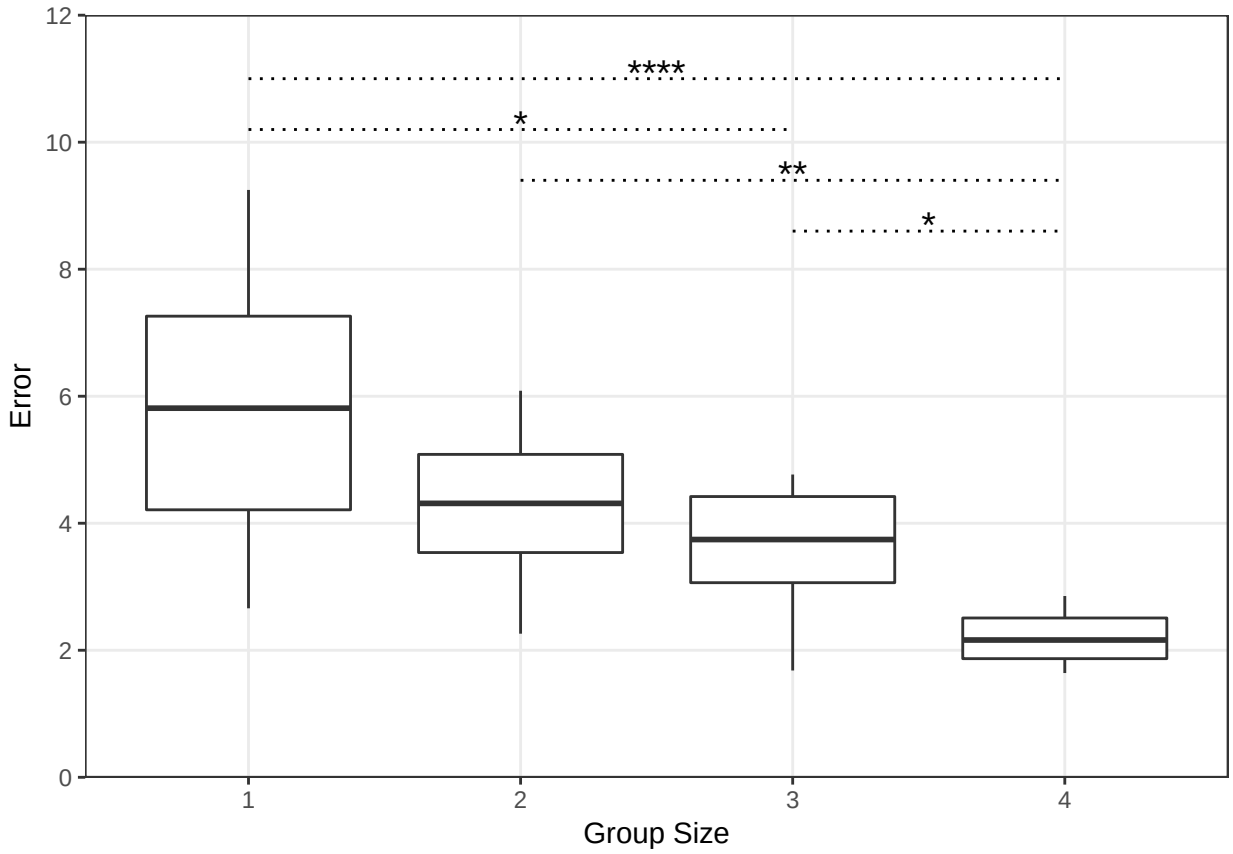
```
##           |
##        V4 |  -0.713718   0.760606   0.260606
##           |     0.7131     0.8938     0.3972
```

The Kruskal-Wallis test failed to reject equality of medians across different group sizes for task completion time (H(3) = 2.1834, p = 0.54), thus we reject H1 and H3.

3.5 Group Size vs. Task Accuracy

```
error <- gather(sourceDataGroups, "group", "error", 5:8)
errGr <- ggplot(error, aes(x=group, y=error)) + geom_boxplot()+labs(x="Group Size", y = "Error")+theme_l
df1 <- data.frame(a = c(1,1:4,4), b = c(11, 11, 11, 11,11, 11))
df2 <- data.frame(a = c(1,1:3,3), b = c(10.2, 10.2, 10.2, 10.2, 10.2))
df3 <- data.frame(a = c(2,2:4,4), b = c(9.4, 9.4, 9.4, 9.4, 9.4))
df4 <- data.frame(a = c(3,3:4,4), b = c(8.6, 8.6, 8.6, 8.6))
errGr + geom_line(data = df1, aes(x = a, y = b),linetype="dotted") + annotate("text", x = 2.5, y = 11.1
     geom_line(data = df2, aes(x = a, y = b),linetype="dotted") + annotate("text", x = 2, y = 10.3, labe
     geom_line(data = df3, aes(x = a, y = b),linetype="dotted") + annotate("text", x = 3, y = 9.5, label
     geom_line(data = df4, aes(x = a, y = b),linetype="dotted") + annotate("text", x = 3.5, y = 8.7, lab
```

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




```
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*
```

The Kruskal-Wallis test revealed significant effect of group size on task accuracy median ($H(3) = 21.3522$, $p < 0.0001$). The post-hoc Dunn test indicate that significant accuracy increase occurs between group sizes 1 and 3 ($p = 0.0486$), 1 and 4 ($p < 0.0001$), 2 and 4 ($p = 0.0024$) and 3 and 4 ($p = 0.0226$). No significant difference exists between group sizes 1 and 2 and between group sizes 2 and 3. This result confirms H4.

3.6 Groups vs. User Workload

```
usersWorkload <- read.csv("workload.csv", header = FALSE,  sep=",")
usersWorkload
```

```
##          V1        V2        V3         V4
## 1  1.000000 0.8486486 0.2398884 0.42071197
## 2  1.007735 0.3351351 0.7740586 0.32524272
## 3  1.000000 0.3594470 0.2370990 0.35113269
## 4  1.001961 0.8341014 0.4810127 0.37540453
## 5  1.000000 0.5319728 0.2531646 0.40140845
## 6  1.003135 0.7945578 0.6025316 0.43309859
## 7  1.003876 0.8385542 0.2000000 0.51056338
## 8  1.003636 0.2602410 0.7180328 0.11971831
## 9  1.000000 0.7364017 0.2655738 0.42537313
## 10 1.000000 0.5857741 0.3949045 0.30597015
## 11 1.008190 0.6855670 0.7006369 0.60447761
## 12 1.002079 0.8814433 0.2579618 0.19029851
## 13 1.000000 0.7396694 0.1146245 0.47951807
## 14 1.000875 0.6818182 0.2450593 0.14457831
## 15 1.001959 0.6666667 0.8142292 0.37228916
## 16 1.001543 0.4560185 0.7745358 0.59518072
## 17 1.001359 0.4277551 0.1432361 0.18047340
## 18 1.000000 0.8742857 0.4880637 0.49704142
## 19 1.001999 0.6045504 0.5809129 0.80473370
## 20 1.000000 0.6435536 0.7219917 0.00000000
## 21       NA 0.3901099 0.1784232 0.46666667
## 22       NA 0.8168498 0.5118397 0.22424242
## 23       NA 0.8901210 0.5409836 0.45454545
## 24       NA 0.2368952 0.2659381 0.26363636
## 25       NA 0.3833819 0.2421525 0.26724138
```

```
## 26       NA 0.8126822 0.6591928 0.50862069
## 27       NA 0.5645806 0.2959641 0.39224138
## 28       NA 0.7683089 0.4377224 0.25862069
## 29       NA 0.4294069 0.6512456 0.51171875
## 30       NA 0.8604845 0.2669039 0.27343750
## 31       NA 0.8459484 0.2538700 0.51562500
## 32       NA 0.3027605 0.2507740 0.22265625
## 33       NA        NA 0.7399381 0.56530214
## 34       NA        NA 0.7176080 0.08771930
## 35       NA        NA 0.2375415 0.16569201
## 36       NA        NA 0.4667774 0.51267057
## 37       NA        NA 0.4025596 0.05921053
## 38       NA        NA 0.6788831 0.26754386
## 39       NA        NA 0.2646888 0.64692980
## 40       NA        NA 0.6886314 0.30701750
## 41       NA        NA 0.3200579 0.28886169
## 42       NA        NA 0.2932657 0.38514892
## 43       NA        NA 0.3638470 0.54385965
## 44       NA        NA 0.5469293 0.23133415
## 45       NA        NA 0.3012746 0.23067332
## 46       NA        NA 0.5012658 0.61720698
## 47       NA        NA 0.5189873 0.25311721
## 48       NA        NA 0.3240506 0.12718204
## 49       NA        NA 0.3215138 0.36289222
## 50       NA        NA 0.4091374 0.25375171
## 51       NA        NA 0.4616434 0.52387449
## 52       NA        NA 0.7004754 0.28922237
## 53       NA        NA 0.3058637 0.62378765
## 54       NA        NA 0.4659271 0.11026034
## 55       NA        NA 0.3510638 0.40939255
## 56       NA        NA 0.7398453 0.30934150
## 57       NA        NA 0.1885880 0.09276665
## 58       NA        NA 0.5855589 0.26358807
## 59       NA        NA 0.2324431 0.31426237
## 60       NA        NA 0.3837784 0.51409889
## 61       NA        NA 0.3344426 0.38775510
## 62       NA        NA 0.4309484 0.26267281
## 63       NA        NA 0.4475874 0.71099408
## 64       NA        NA 0.3902116 0.32060566
## 65       NA        NA 0.6825397 0.22325581
## 66       NA        NA 0.3082011 0.56627907
## 67       NA        NA 0.3167116 0.27441860
## 68       NA        NA 0.4541779 0.19418605
## 69       NA        NA 0.4541779 0.44656918
## 70       NA        NA 0.7440678 0.46681665
## 71       NA        NA 0.2084746 0.09448819
## 72       NA        NA 0.4542373 0.57930259
## 73       NA        NA        NA 0.51364522
## 74       NA        NA        NA 0.10233918
## 75       NA        NA        NA 0.49220273
## 76       NA        NA        NA 0.39376218
## 77       NA        NA        NA 0.14591291
## 78       NA        NA        NA 0.43009931
## 79       NA        NA        NA 0.37433156
```

```
## 80       NA        NA        NA 0.32314744
```

We extracted the time each participant took to perform each transformation action (translation, rotation, scale and camera) during the task and normalized by the group worked time. Then, we added up all transformations into the final worked time of the participant. We performed an analysis to assess the workload of each team members for all group size.

```
workload <- gather(usersWorkload, "group", "workload", 1:4)
dunn.test(workload$workload,workload$group, kw=TRUE, method="holm")
```

```
##   Kruskal-Wallis rank sum test
##
## data: x and group
## Kruskal-Wallis chi-squared = 79.0784, df = 3, p-value = 0
##
##
##                            Comparison of x by group
##                                      (Holm)
## Col Mean-|
## Row Mean |         V1         V2         V3
## ---------+-----------------------------------
##       V2 |   3.435690
##          |    0.0006*
##          |
##       V3 |   6.824663   3.509835
##          |    0.0000*    0.0007*
##          |
##       V4 |   8.119165   5.022202   1.876138
##          |    0.0000*    0.0000*     0.0303
```

The Kruskal-Wallis test revealed a significant effect in the workload when group sizes vary ($H(3) = 79.0784$,$p < 0.0001$). The Dunn post-hoc indicates significant decrease in workload between groups size 1 and 2 ($p < 0.0006$), 1 and 3 ($p < 0.0001$), 1 and 4 ($p = 0.0031$), 2 and 3 ($p < 0.0007$), 2 and 4 ($p < 0.0001$). There was no significance difference effect between group sizes 3 and 4 ($p = 0.0303$).

3.7 Groups vs. Work Distibution Balance

```
workDistribution <- read.csv("work_distribution_balance.csv", header = FALSE,  sep="\t")
workDistribution
```

```
##             V1          V2          V3
## 1   Var Team 2  Var Team 3  Var Team 4
## 2  0.263696129 0.286835548  0.00659468
## 3  0.022688678 0.238683907  0.06372513
## 4  0.199389588 0.133459968 0.074708292
## 5  0.080684134 0.240716977 0.160463115
## 6  0.225296778 0.094369492 0.116412088
## 7   0.03836752 0.068483515 0.055953973
## 8  0.001521254 0.146439131 0.183055972
## 9  0.041505246 0.094154215 0.117066883
## 10   0.0689509 0.238688525 0.125208844
## 11 0.003346766 0.154376722  0.09580485
## 12 0.182106965 0.048890327  0.05751343
## 13 0.185827978 0.011195982 0.176105237
## 14 0.334446219 0.154093472  0.14721198
```

```
## 15 0.044372642 0.111244792  0.23253372
## 16 0.426703954 0.034859798 0.182558512
## 17 0.295053083 0.116155413 0.143538626
## 18             0.415207236 0.503935974
## 19             0.237776649 0.237085748
## 20             0.015035532 0.120339786
## 21             0.018896986 0.060505875
## 22             0.299755856
## 23             0.172964426
## 24             0.118175813
## 25             0.215630566
```

We computed the work distribution balance. We calculated the group variance using each individual workload. We adjusted the results by multiplying the variance by the group size. In this test, only groups with two, three and four members were evaluated, since the work distribution in groups with one participant is zero.

```
workdistr <- gather(workDistribution, "group", "workdistr", 1:3)
```

```
## Warning: attributes are not identical across measure variables; they will
## be dropped
```

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

```
##   Kruskal-Wallis rank sum test
##
## data: x and group
## Kruskal-Wallis chi-squared = 4.6289, df = 2, p-value = 0.1
##
##
##                            Comparison of x by group
##                                      (Holm)
## Col Mean-|
## Row Mean |         V1         V2
## ---------+----------------------
##       V2 |  -2.145691
##          |     0.0478
##          |
##       V3 |  -0.936301   1.209389
##          |     0.1746     0.2265
```

The Kruskal-Wallis test failed to reject equality of medians across different group sizes for work distribution balance ($H(2) = 4.628$, $p = 0.1$)

3.8 Correlation between Groups and user role change

```
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
```

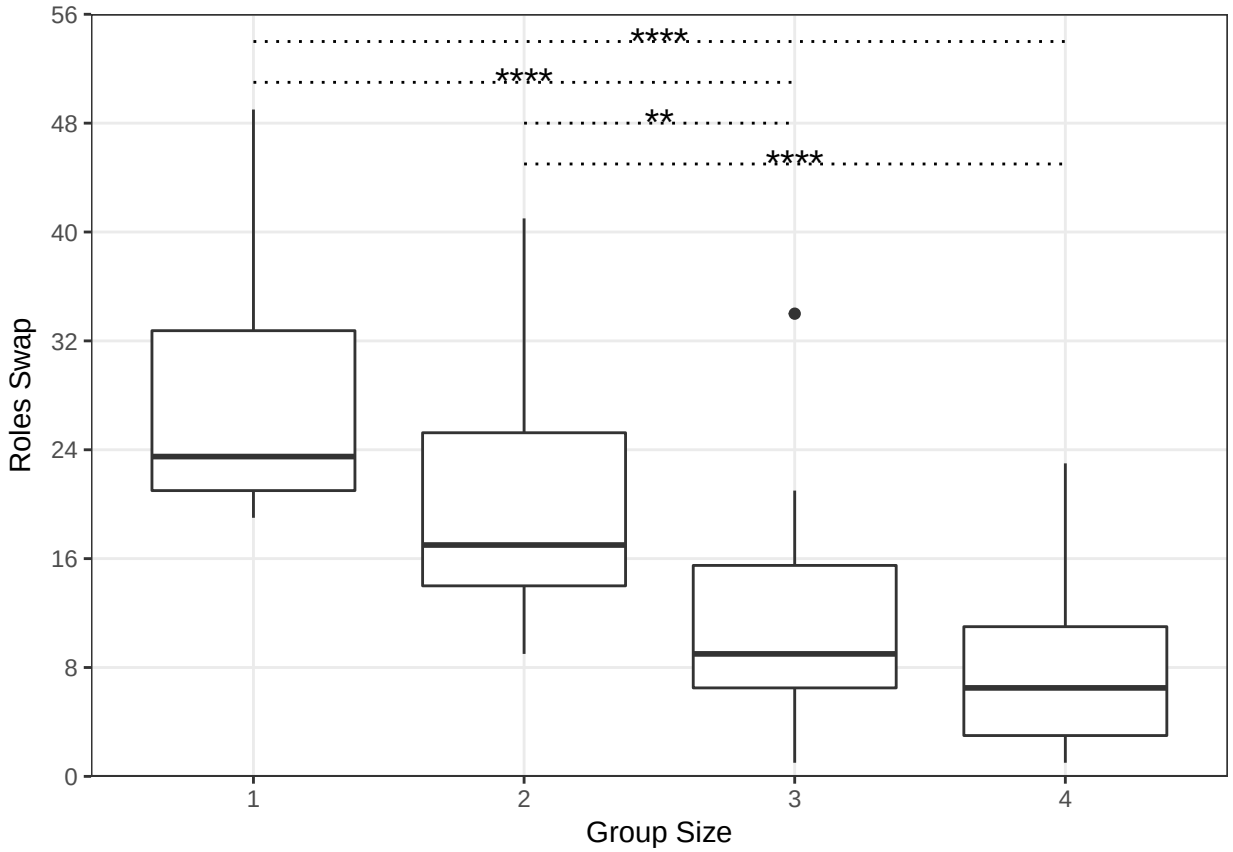
```
##     V1       V2 V3
## 1    1 9.070060 21
## 2    1 7.381541 24
## 3    1 6.207668 32
## 4    1 3.341505 49
## 5    1 6.817599 19
## 6    2 3.251326 17
## 7    2 3.251326 24
## 8    2 4.849439 15
```

```
## 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
```

To extract the division of tasks between the groups, we identified the frequency users swap between transformations (translation, rotation, scale). We call this swap a role change.

```
rolesGr <- ggplot(sourceDataRoles, aes(x=as.character(V1), y=V2)) + geom_boxplot()+labs(x="Group Size",
df1 <- data.frame(a = c(1,1:4,4), b = c(54, 54, 54, 54,54,54))
df2 <- data.frame(a = c(1,1:3,3), b = c(51, 51, 51, 51, 51))
df3 <- data.frame(a = c(2,2:3,3), b = c(48, 48, 48, 48))
df4 <- data.frame(a = c(2,2:4,4), b = c(45, 45, 45, 45,45))
rolesGr + geom_line(data = df1, aes(x = a, y = b),linetype="dotted") + annotate("text", x = 2.5, y = 54
  geom_line(data = df2, aes(x = a, y = b),linetype="dotted") + annotate("text", x = 2, y = 51.1, label =
  geom_line(data = df3, aes(x = a, y = b),linetype="dotted") + annotate("text", x = 2.5, y = 48.1, label
  geom_line(data = df4, aes(x = a, y = b),linetype="dotted") + annotate("text", x = 3, y = 45.1, label =
```




```
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
```

The Kruskal-Wallis variance analysis revealed significant effect between groups and user roles changes ($H(3) = 40.1615$, $p < 0.0001$). The post-hoc Dunn test indicates that significant work division occurs between groups size 1 and 3 ($p < 0.0001$), 1 and 4 ($p < 0.0001$), 2 and 3 ($p = 0.0031$) and 2 and 4 ($p < 0.0001$).

3.9 Accuracy vs. User Roles

Since the accuracy and the work division have similar behaviors, we hypothesize that the two variables were related.

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

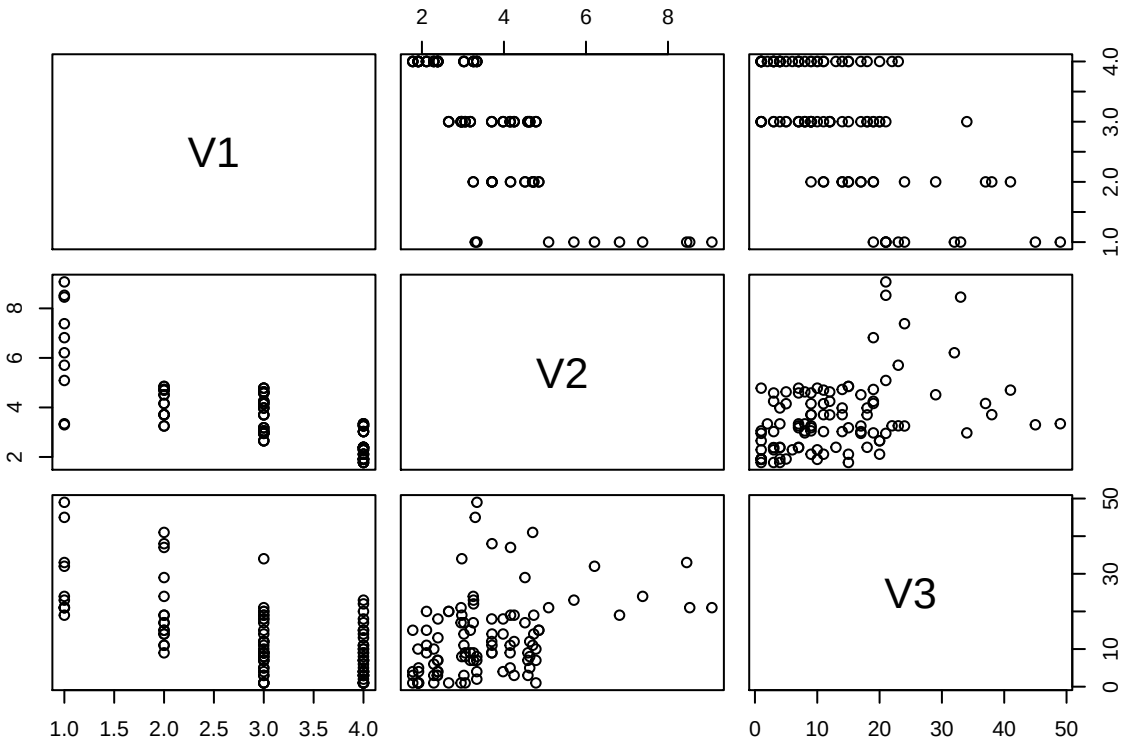
```
##           V2           V3
## statistic 0.8709327    0.8906971
## p.value   6.068221e-08 4.276405e-07
```

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

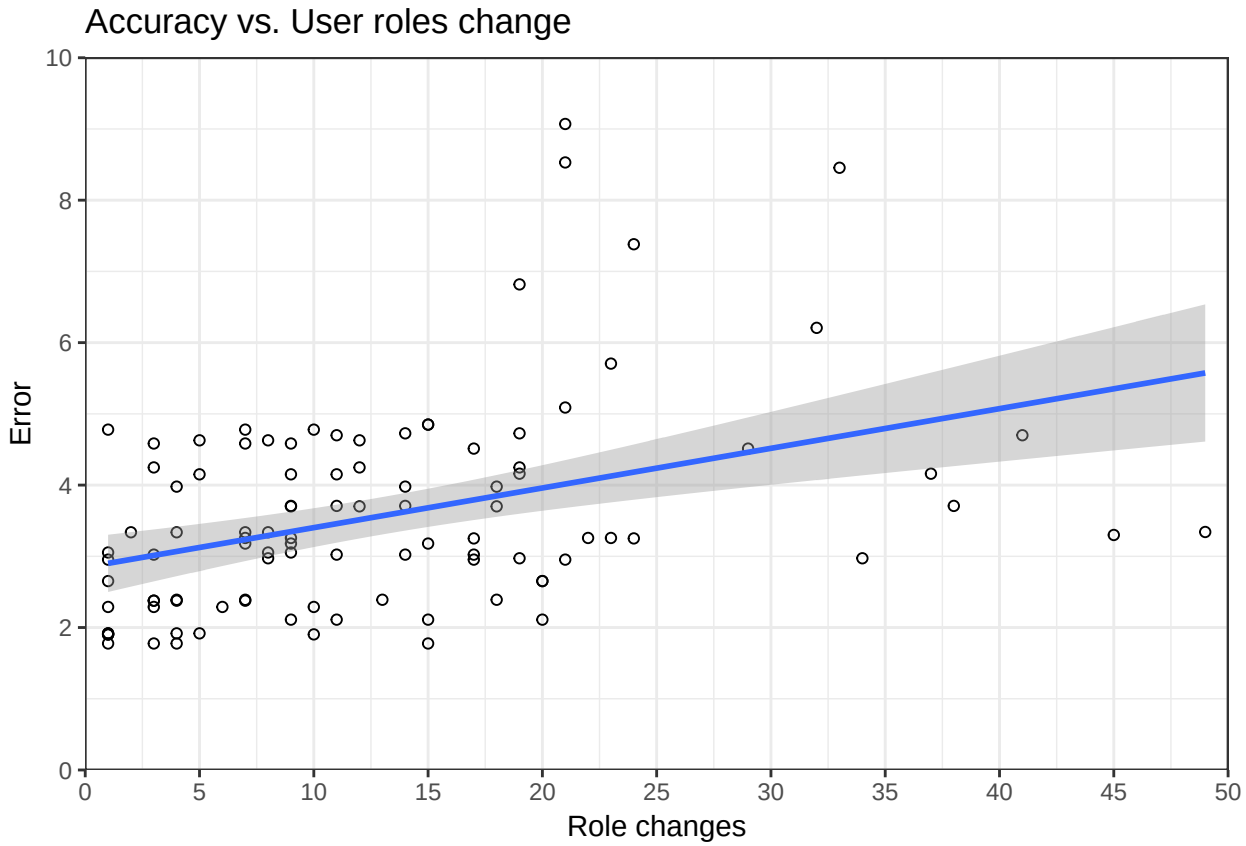
```
## Warning in cor.test.default(sourceGroupsErrorRoles$V2,
## sourceGroupsErrorRoles$V3, : Cannot compute exact p-value with ties
```

```
##
##  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
```



The Pearson correlation revealed a significant effect between accuracy and user roles changes (r = -0.3957,p = <0.0001).

3.10 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
```

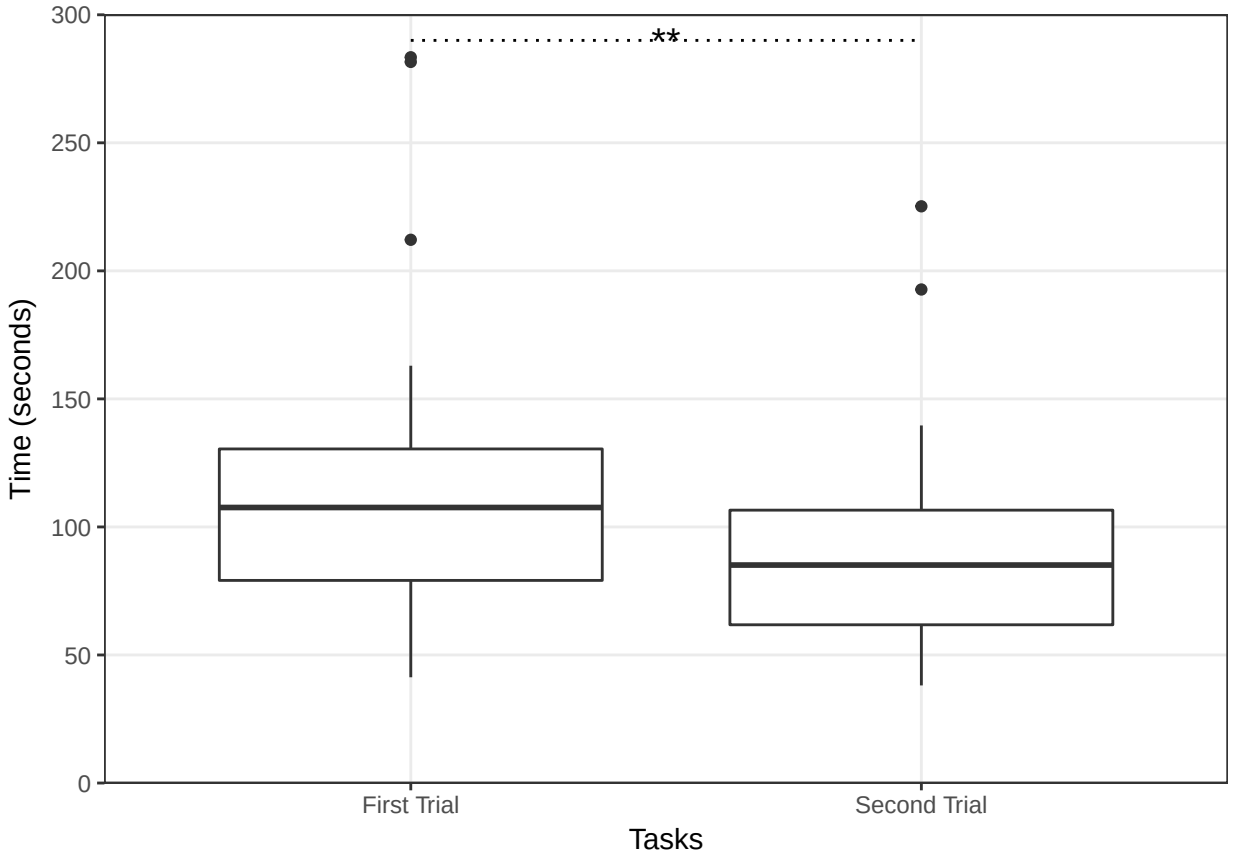
```
wilcox.test(sourceDataTasks$Error.task.3,sourceDataTasks$Error.task.4,paired=TRUE)
```

```
##
##  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
```

```
tasksTime <- gather(sourceDataTasks, "task", "time", 2:3)
summary(tasksTime)
```

```
##   Members.Task  Error.task.3    Error.task.4      MeanTime
##  Min.   :1.0   Min.   :1.681   Min.   :1.642   Min.   : 39.71
##  1st Qu.:2.0   1st Qu.:2.509   1st Qu.:2.685   1st Qu.: 73.13
##  Median :2.5   Median :4.148   Median :3.596   Median : 96.73
##  Mean   :2.5   Mean   :4.021   Mean   :3.864   Mean   :106.67
##  3rd Qu.:3.0   3rd Qu.:5.102   3rd Qu.:4.758   3rd Qu.:122.64
##  Max.   :4.0   Max.   :7.861   Max.   :9.248   Max.   :253.40
##    MeanError         task                time
##  Min.   :1.795   Length:44          Min.   : 38.12
##  1st Qu.:2.514   Class :character   1st Qu.: 68.00
##  Median :3.869   Mode  :character   Median : 89.63
##  Mean   :3.943                      Mean   :106.67
##  3rd Qu.:4.654                      3rd Qu.:127.19
##  Max.   :8.555                      Max.   :283.41
```

```
taskT <- ggplot(tasksTime, aes(x=as.character(task), y=time)) + geom_boxplot()+labs(x="Tasks", y = "Tim
df1 <- data.frame(a = c(1,1:2,2), b = c(290, 290, 290, 290))
taskT + geom_line(data = df1, aes(x = a, y = b),linetype="dotted") + annotate("text", x = 1.5, y = 290.
```




```
tasksError <- gather(sourceDataTasks, "task", "error", 4:5)
summary(tasksError)
```

```
##   Members.Task  Time.task.3      Time.task.4        MeanTime
##  Min.   :1.0   Min.   : 41.30   Min.   : 38.12   Min.   : 39.71
```

```
##  1st Qu.:2.0   1st Qu.: 76.82   1st Qu.: 61.24   1st Qu.: 73.13
##  Median :2.5   Median :107.59   Median : 85.12   Median : 96.73
##  Mean   :2.5   Mean   :120.57   Mean   : 92.78   Mean   :106.67
##  3rd Qu.:3.0   3rd Qu.:130.76   3rd Qu.:111.04   3rd Qu.:122.64
##  Max.   :4.0   Max.   :283.41   Max.   :225.21   Max.   :253.40
##    MeanError         task               error
##  Min.   :1.795   Length:44          Min.   :1.642
##  1st Qu.:2.514   Class :character   1st Qu.:2.631
##  Median :3.869   Mode  :character   Median :3.836
##  Mean   :3.943                      Mean   :3.943
##  3rd Qu.:4.654                      3rd Qu.:4.845
##  Max.   :8.555                      Max.   :9.248
```

```
ggplot(tasksError, aes(x=as.character(task), y=error)) + geom_boxplot()+labs(x="Tasks", y = "Error")+th
```

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