Collaborative 3DUI Data Analysis

Jerônimo G. Grandi

August 22, 2016

Data analyisis

This is a on going analysis of the data collected in the user experiment performed. . . .

Comparisons between groups

Hypotesis

- Groups with more than one member perform the tasks with less errors.
- Groups with two and three members perform the tasks with less errors than with one and four.
- Groups with more than one member perform the tasks faster.
- Groups with two and three members perform the tasks faster than with one and four.
- Task one and two are easier than three and four.

Data Summary

```
errorXGroups <- read.csv("ErrorXGroups.csv",header = FALSE)
errorXGroups
```

```
##           V1        V2       V3       V4
## 1   4.280380  1.923633 0.867342 1.554914
## 2  16.953010 11.753480 3.243309 3.538198
## 3   9.070060  3.251326 2.972291 2.389965
## 4   8.529819  3.595758 3.701374 2.110792
## 5   3.869670  3.004579 2.359320 1.402203
## 6   6.642847  4.069859 4.060901 4.501410
## 7   7.381541  4.849439 3.978512 3.337539
## 8   5.705525  4.512303 3.053629 2.288988
## 9   1.317529  2.192855 2.259253 3.145458
## 10  9.220284  5.547572 3.317118 6.065219
## 11  6.207668  6.037892 4.150805 3.023767
## 12  5.088869  4.339551 4.628316 3.257560
## 13  4.024921  2.576868 1.300738 1.623129
## 14  8.576299  8.645294 3.357939 2.768127
## 15  3.341505  4.726371 3.179116 1.776181
## 16  3.299659  3.707839 4.247869 1.903549
## 17  4.456124  4.601643 2.329338       NA
## 18  9.002918  8.891509 4.309965       NA
## 19  6.817599  4.380987 2.652203       NA
## 20  8.455383  5.540681 2.953497       NA
## 21        NA  1.390942 2.187007       NA
## 22        NA  4.142436 8.903512       NA
## 23        NA  4.699787 4.778647       NA
```

```
## 24        NA  4.160006 4.583976       NA
```

```r
timeXGroups <- read.csv("TimeXGroups.csv", header = FALSE)
timeXGroups
```

```
##          V1        V2        V3        V4
## 1   44.6010  19.92100  58.66863  47.19800
## 2   27.7300  23.75830  16.39880  21.40800
## 3   44.5060  86.01300 127.23090 162.96200
## 4   56.6990  90.59442  82.56380  87.06737
## 5   86.2698  13.64980  29.00930  22.00295
## 6   26.6007  15.95330  49.16030  16.25100
## 7  129.5357  87.26190 120.67140  74.45340
## 8   88.6565 113.11968  91.11170  71.80547
## 9   37.2235  16.45030  36.21300  21.09751
## 10  32.5000  14.49990  22.41350  16.95763
## 11 130.7602  69.36490  76.82150  56.40480
## 12  63.4430  58.41970  55.29020  63.90760
## 13  71.4288  17.20950  21.44110  54.20073
## 14  55.8885  29.62760  17.18680  37.12230
## 15 281.5956 139.37750  86.81320 127.17520
## 16 225.2061 111.04040  61.24430  60.76750
## 17  32.6565  59.33450  20.20286        NA
## 18  41.4964  16.48410  25.70100        NA
## 19 105.6555 109.53340 283.41380        NA
## 20 139.6341  83.16500 192.72400        NA
## 21       NA  35.49839  24.40880        NA
## 22       NA  34.55538  42.26010        NA
## 23       NA  99.48090  41.29660        NA
## 24       NA  93.11320  38.11630        NA
```

Below the data is summarized:

```r
describe(errorXGroups)
```

```
##    vars  n mean   sd median trimmed  mad  min   max range skew kurtosis
## V1    1 20 6.61 3.36   6.43    6.34 3.18 1.32 16.95 15.64 1.17     2.02
## V2    2 24 4.69 2.32   4.36    4.43 1.39 1.39 11.75 10.36 1.33     1.73
## V3    3 24 3.47 1.55   3.28    3.38 1.33 0.87  8.90  8.04 1.48     3.97
## V4    4 16 2.79 1.22   2.58    2.66 1.07 1.40  6.07  4.66 1.11     0.74
##      se
## V1 0.75
## V2 0.47
## V3 0.32
## V4 0.31
```

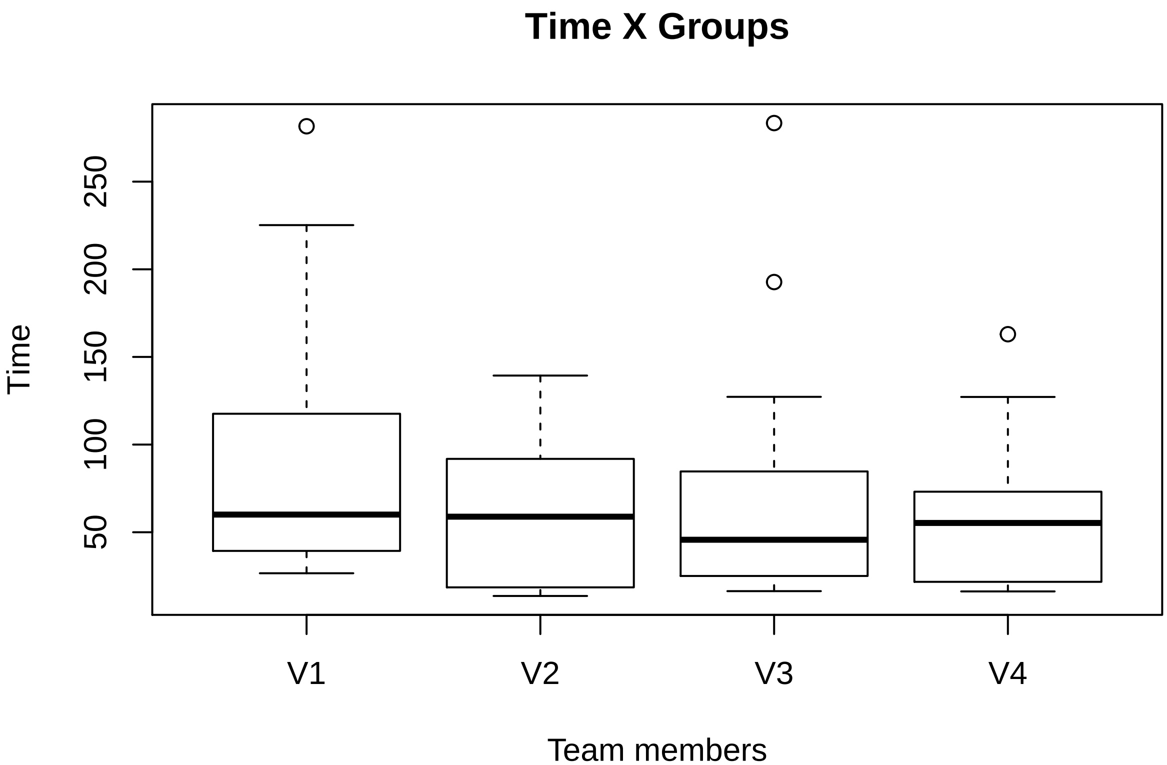
```r
describe(timeXGroups)
```

```
##    vars  n  mean    sd median trimmed   mad   min    max  range skew
## V1    1 20 86.10 67.95  60.07   72.56 40.76 26.60 281.60 254.99 1.49
## V2    2 24 59.89 39.94  58.88   57.84 54.91 13.65 139.38 125.73 0.26
## V3    3 24 67.52 62.67  45.71   55.53 35.26 16.40 283.41 267.01 1.96
## V4    4 16 58.80 40.93  55.30   54.40 37.74 16.25 162.96 146.71 1.06
##    kurtosis    se
## V1     1.48 15.19
## V2    -1.43  8.15
## V3     3.69 12.79
```

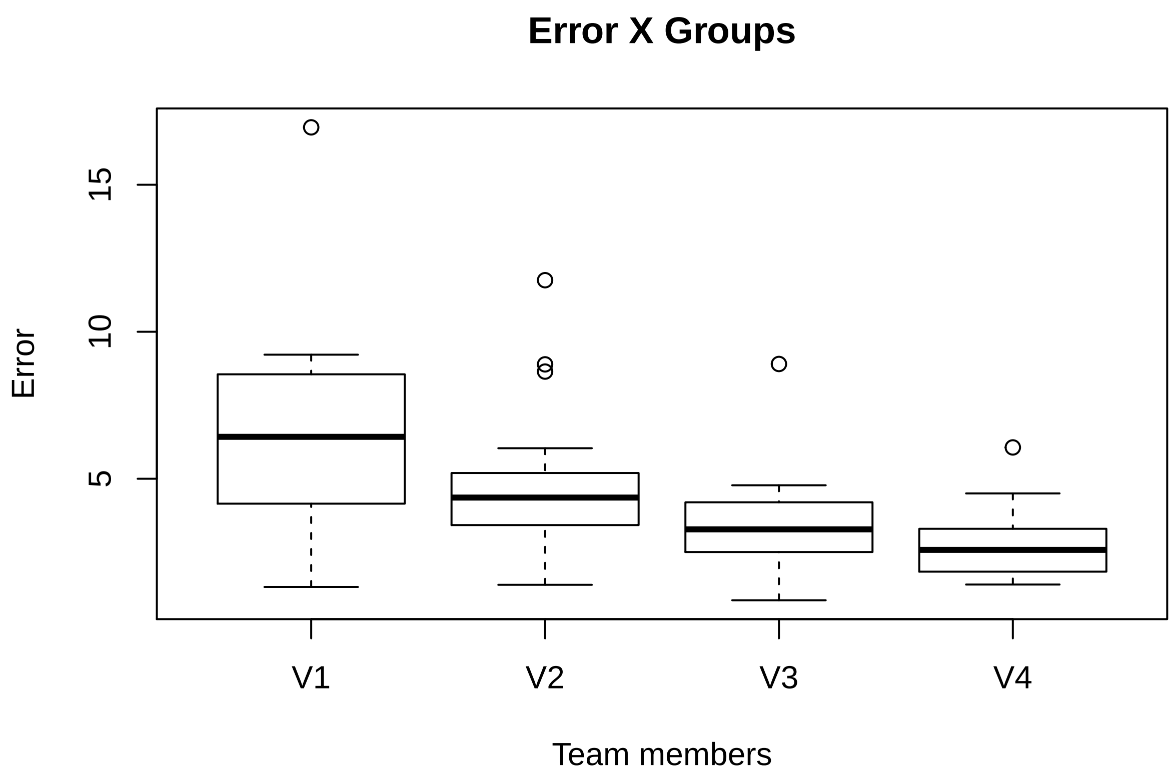
```
## V4      0.40 10.23
```

And ploted:

```
boxplot(timeXGroups,xlab="Team members",ylab="Time",main="Time X Groups")
```




```
boxplot(errorXGroups,xlab="Team members",ylab="Error",main="Error X Groups")
```



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.

Correlacao Qtd de Pessoas – Tempo Medio	
Pearson	
	Colunas 1 e 10
n (pares) =	23
r (Pearson) =	-0,3321
IC 95% =	-0,65 a 0,09
IC 99% =	-0,73 a 0,23
R2 =	0,1103
t =	-1,6137
GL =	21
(p) =	0,1214
Poder 0.05 =	0,4598
Poder 0.01 =	0,2168

Correlacao Qtd de Pessoas – Erro Medio	
Pearson	
	Colunas 1 e 11
n (pares) =	23
r (Pearson) =	-0,7719
IC 95% =	-0,90 a -0,53
IC 99% =	-0,92 a -0,42
R2 =	0,5959
t =	-5,5649
GL =	21
(p) =	< 0.0001
Poder 0.05 =	0,9984
Poder 0.01 =	0,9880

Kruskal-Wallis

A collection of data samples are independent if they come from unrelated populations and the samples do not affect each other. Using the Kruskal-Wallis Test, we can decide whether the population distributions are identical without assuming them to follow the normal distribution.

Erro por Qnt de Pessoas

Kruskal-Wallis				
H =	25.9856			
Graus de liberdade =	3			
(p) Kruskal-Wallis =	< 0.0001			
R 1 =	1227			
R 2 =	1157			
R 3 =	808			
R 4 =	378			
R 1 (posto médio) =	61.35			
R 2 (posto médio) =	48.2083			
R 3 (posto médio) =	33.6667			
R 4 (posto médio) =	23.625			
Comparações (métod	Dif. Postos	z calculado	z crítico	p
Postos médios 1 e 2	13.1417	1.7795	2.635	ns
Postos médios 1 e 3	27.6833	3.7485	2.635	< 0.05
Postos médios 1 e 4	37.725	4.611	2.635	< 0.05
Postos médios 2 e 3	14.5417	2.0651	2.635	ns
Postos médios 2 e 4	24.5833	3.1226	2.635	< 0.05
Postos médios 3 e 4	10.0417	1.2755	2.635	ns

Analise de Erro em cada checkpoint por Qtd de Pessoas				
Shapiro				
Kruskal-Wallis				
	Resultados			
H =	90,0959			
Graus de liberdade =	3			
(p) Kruskal-Wallis =	< 0.0001			
R 1 =	27466,0000			
R 2 =	27680,0000			
R 3 =	23579,0000			
R 4 =	8846,0000			
R 1 (posto médio) =	289,1158			
R 2 (posto médio) =	225,0407			
R 3 (posto médio) =	188,6320			
R 4 (posto médio) =	117,9467			
Dunn (posHoc)				
Comparações (método de	Dif. Postos	z calculado	z crítico	p
Postos médios E1 e E2	64,0751	3,8830	2635	< 0.05
Postos médios E1 e E3	100,4838	6,1108	2635	< 0.05
Postos médios E1 e E4	171,1691	9,1725	2635	< 0.05
Postos médios E2 e	36,4087	2,3729	2635	ns
Postos médios E2 e E4	107,0940	6,0508	2635	< 0.05
Postos médios E3 e E4	70,6853	4,0059	2635	< 0.05

Comparações Student-Ne	Dif. Postos	p-valor
Grupos (1 e 2) =	13.1417	0.0752
Grupos (1 e 3) =	27.6833	0.0002
Grupos (1 e 4) =	37.725	< 0.0001
Grupos (2 e 3) =	14.5417	0.0389
Grupos (2 e 4) =	24.5833	0.0018
Grupos (3 e 4) =	10.0417	0.2021

Comparisons between tasks

Hypotesis

- Task 1 and 2 are easier than 3 and 4;
- Task 4 is performed better than task 3

Summary

First we set the environment, load and show the raw data.

```r
setwd("/home/jeronimo/Documents/DataAnalysis3DController/Analysis/R")
dAvgPerTask <- read.csv("SummaryToR.csv")
dAvgPerTask
```

```
##    Members       T1       T2       T3        T4       E1        E2
## 1        1 44.60100 27.73000  44.5060  56.69900 4.280380 16.953010
## 2        1 86.26980 26.60070 129.5357  88.65650 3.869670  6.642847
## 3        1 37.22350 32.50000 130.7602  63.44300 1.317529  9.220284
## 4        1 71.42880 55.88850 281.5956 225.20610 4.024921  8.576299
## 5        1 32.65650 41.49640 105.6555 139.63410 4.456124  9.002918
## 6        2 19.92100 23.75830  86.0130  90.59442 1.923633 11.753480
## 7        2 13.64980 15.95330  87.2619 113.11968 3.004579  4.069859
## 8        2 16.45030 14.49990  69.3649  58.41970 2.192855  5.547572
## 9        2 17.20950 29.62760 139.3775 111.04040 2.576868  8.645294
## 10       2 59.33450 16.48410 109.5334  83.16500 4.601643  8.891509
## 11       2 35.49839 34.55538  99.4809  93.11320 1.390942  4.142436
## 12       3 58.66863 16.39880 127.2309  82.56380 0.867342  3.243309
## 13       3 29.00930 49.16030 120.6714  91.11170 2.359320  4.060901
## 14       3 36.21300 22.41350  76.8215  55.29020 2.259253  3.317118
## 15       3 21.44110 17.18680  86.8132  61.24430 1.300738  3.357939
## 16       3 20.20286 25.70100 283.4138 192.72400 2.329338  4.309965
## 17       3 24.40880 42.26010  41.2966  38.11630 2.187007  8.903512
## 18       4 47.19800 21.40800 162.9620  87.06737 1.554914  3.538198
## 19       4 22.00295 16.25100  74.4534  71.80547 1.402203  4.501410
## 20       4 21.09751 16.95763  56.4048  63.90760 3.145458  6.065219
## 21       4 54.20073 37.12230 127.1752  60.76750 1.623129  2.768127
##          E3       E4
## 1  9.070060 8.529819
## 2  7.381541 5.705525
## 3  6.207668 5.088869
## 4  3.341505 3.299659
## 5  6.817599 8.455383
## 6  3.251326 3.595758
## 7  4.849439 4.512303
## 8  6.037892 4.339551
## 9  4.726371 3.707839
## 10 4.380987 5.540681
## 11 4.699787 4.160006
## 12 2.972291 3.701374
## 13 3.978512 3.053629
## 14 4.150805 4.628316
## 15 3.179116 4.247869
## 16 2.652203 2.953497
## 17 4.778647 4.583976
## 18 2.389965 2.110792
## 19 3.337539 2.288988
## 20 3.023767 3.257560
## 21 1.776181 1.903549
```

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.

We split the data for better manipulation:

```
dAvgTime <- subset(dAvgPerTask, select = c(2,3,4,5))
dAvgError <- subset(dAvgPerTask, select = c(6,7,8,9))
```

Below the data is summarized:

```
describe(dAvgTime)
```

```
##    vars  n   mean    sd median trimmed   mad   min    max  range skew
## T1    1 21  36.60 19.92  32.66   34.17 18.46 13.65  86.27  72.62 0.87
## T2    2 21  27.81 11.90  25.70   26.38 13.13 14.50  55.89  41.39 0.77
## T3    3 21 116.21 63.86 105.66  105.27 37.22 41.30 283.41 242.12 1.46
## T4    4 21  91.79 45.68  83.17   83.31 32.50 38.12 225.21 187.09 1.57
##    kurtosis    se
## T1    -0.25  4.35
## T2    -0.50  2.60
## T3     1.64 13.93
## T4     1.84  9.97
```

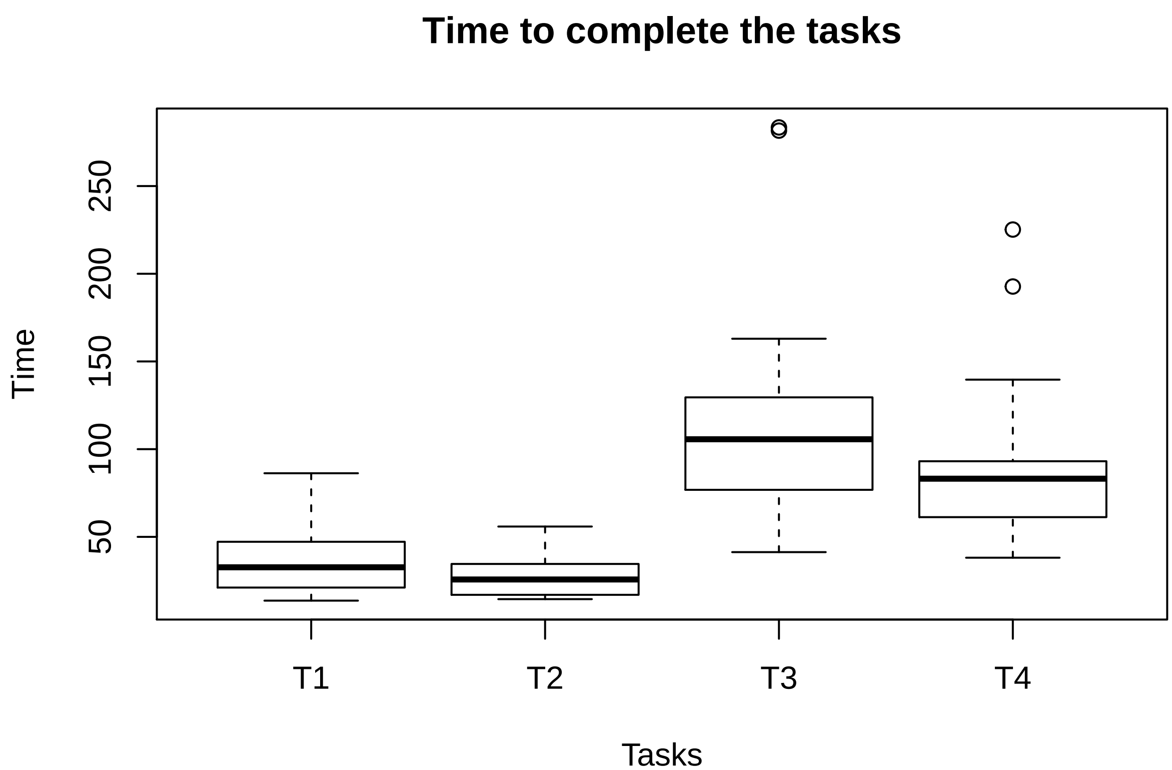
```
describe(dAvgError)
```

```
##    vars  n mean   sd median trimmed  mad  min   max range skew kurtosis
## E1    1 21 2.51 1.15   2.26    2.44 1.27 0.87  4.60  3.73 0.50    -1.15
## E2    2 21 6.55 3.55   5.55    6.05 3.31 2.77 16.95 14.18 1.17     1.04
## E3    3 21 4.43 1.82   4.15    4.26 1.44 1.78  9.07  7.29 0.83    -0.04
## E4    4 21 4.27 1.74   4.16    4.04 1.34 1.90  8.53  6.63 1.07     0.71
##      se
## E1 0.25
## E2 0.77
## E3 0.40
## E4 0.38
```

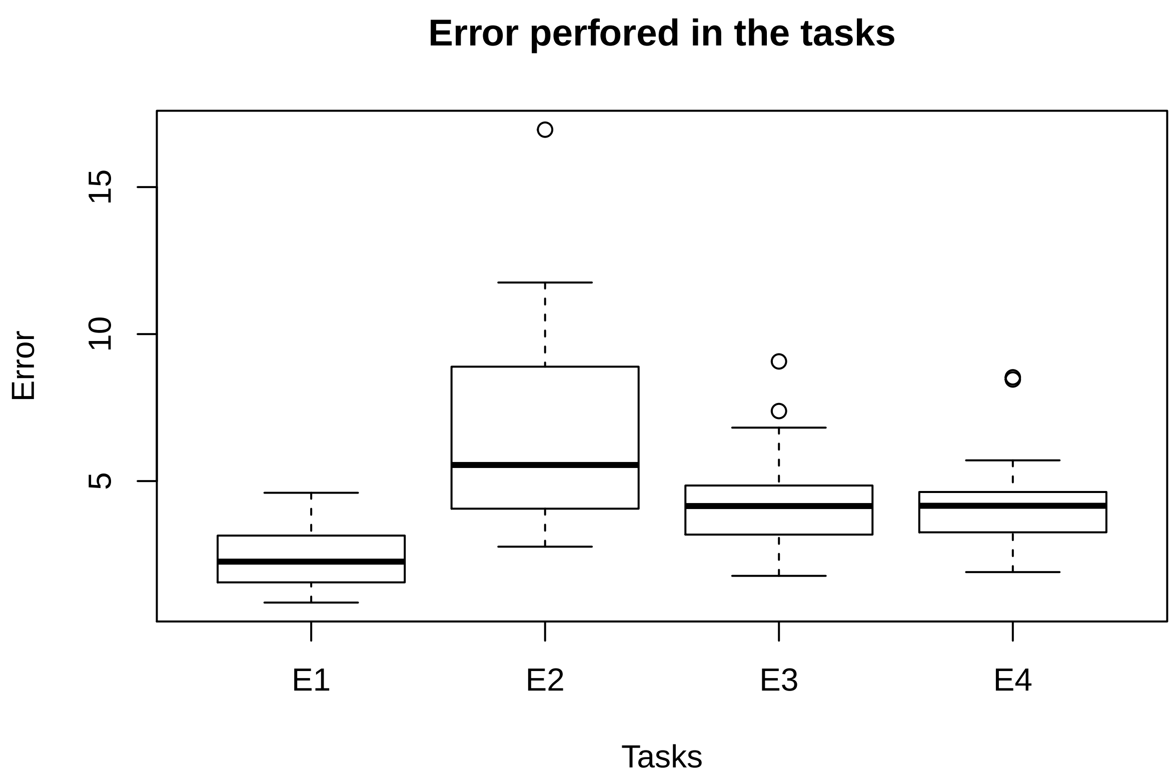
Plots

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

```
boxplot(dAvgTime,xlab="Tasks",ylab="Time",main="Time to complete the tasks")
```




```
boxplot(dAvgError,xlab="Tasks",ylab="Error",main="Error perfored in the tasks")
```



Analysis of time of completion per task

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.

```
shap_dT <- lapply(dAvgTime, shapiro.test)
shap_dE <- lapply(dAvgError, shapiro.test)
res_shap_dT <- sapply(shap_dT, `[`, c("statistic","p.value"))
res_shap_dE <- sapply(shap_dE, `[`, c("statistic","p.value"))

res_shap_dT
```

```
##           T1         T2        T3          T4
## statistic 0.8974621  0.9017898 0.812624    0.8037059
## p.value   0.03125427 0.0379083 0.001029792 0.0007478811
```

```
res_shap_dE
```

```
##           E1         E2          E3        E4
## statistic 0.9139373  0.8525865   0.9313148 0.8860365
## p.value   0.06570504 0.004708184 0.1462635 0.0189353
```

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.

Friedman

Friedman test is a non-parametric randomized block analysis of variance. Which is to say it is a non-parametric version of a one way ANOVA with repeated measures. That means that while a simple ANOVA test requires the assumptions of a normal distribution and equal variances (of the residuals), the Friedman test is free from those restriction. The price of this parametric freedom is the loss of power (of Friedman's test compared to the parametric ANOVa versions).

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

The test statistic for the Friedman's test is a Chi-square with [(number of repeated measures)-1] degrees of freedom. A detailed explanation of the method for computing the Friedman test is available on Wikipedia.

Friedman				
	T1	T2	T3	T4
Soma dos R	40,0000	32,0000	84,0000	74,0000
Mediana =	35,4984	25,7010	127,1752	88,6565
Média dos R	1,7391	1,3913	3,6522	3,2174
Média dos v	37,0001	27,8165	131,2051	100,7838
Desvio padr	19,0349	11,3654	67,6085	46,0546
Friedman (F	50,3739			
Graus de libe	3			
(p) =	< 0.0001			
PostHoc				
Comparaçõe	Diferença	(p)		
Ranks T1 e	8	ns		
Ranks T1 e	44	< 0.05		
Ranks T1 e	34	< 0.05		
Ranks T2 e	52	< 0.05		
Ranks T2 e	42	< 0.05		
Ranks T3 e	10	ns		

Friedman				
	E1	E2	E3	E4
Soma dos Ra	28,0000	79,0000	62,0000	61,0000
Mediana =	2,2593	5,5476	4,0136	4,0136
Média dos Ra	1,2174	3,4348	2,6957	2,6522
Média dos va	2,4973	6,3589	4,3564	4,2113
Desvio padrã	1,1008	3,5407	1,7560	1,6780
Friedman (Fr	35,6087			
Graus de libe	3			
(p) =	< 0.0001			
PostHoc				
Comparações	Diferença	(p)		
Ranks E1 e	51	< 0.05		
Ranks E1 e	34	< 0.05		
Ranks E1 e	33	< 0.05		
Ranks E2 e	17	ns		
Ranks E2 e	18	ns		
Ranks E3 e	1	ns		