VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY

UNIVERSITY OF TECHNOLOGY

OFIICE FOR INTERNATIONAL STUDY PROGRAM

PROBABILITY AND STATISTICS (MT2013)

Assignment

Analyze the data in the data set using R Studio

Lecturer: Nguyễn Tiến Dũng

HO CHI MINH CITY, MAY 2022

Contents

1 Member list & Workload 2

2 Data description 3

3 Theory 4

3.1 ANOVA (Analysis of variance) 4

3.1.1 Definition 4

3.2 Tukey's HSD (honestly significant difference) 5

3.2.1 Definition 5

3.2.2 Equation 5

3.3 Simple Linear Regression 6

3.3.1 Definition 6

3.3.2 Equation 6

4 Question 1 6

5 Question 2 7

6 Question 3 7

6.1 Descriptive data 7

6.2 Graph 9

7 Question 4 13

8 Question 5 17

9 Conclusion 18

1 Member list & Workload

No.	Full name	Student ID	Task	Contribution
1	Hoàng Duy Tân	2053422		20%
2	Cao Đức Nam	1952856		0%
3	Nguyễn Tôn Minh	2052600		40%
4	Đái Ngọc Quốc Trung	2053537		20%
5	Ngô Trương Trọng Nghĩa	2053264		20%

2 Data description

The data is stored in file game.csv. It contains the number of hours people of different age groups spent in different games per days, per weeks and the total hours spent and the actions per minutes, spanning from 18 to 24 years old.

Data Set Information:

- We aggregated screen movements into screen-fixations using a Salvucci & Goldberg (2000) dispersion-threshold algorithm, and defined Perception Action Cycles (PACs) as fixations with at least one action.
- Time is recorded in terms of timestamps in the StarCraft 2 replay file. When the game is played on 'faster', 1 real-time second is equivalent to roughly 88.5 timestamps.
- List of possible game actions is discussed in Thompson, Blair, Chen, & Henrey (2013)

Attribute infomation:

1. GameID: Unique ID number for each game (integer)
2. LeagueIndex: Bronze, Silver, Gold, Platinum, Diamond, Master, GrandMaster, and Professional leagues coded 1-8 (Ordinal)
3. Age: Age of each player (integer)
4. HoursPerWeek: Reported hours spent playing per week (integer)
5. TotalHours: Reported total hours spent playing (integer)
6. APM: Action per minute (continuous)

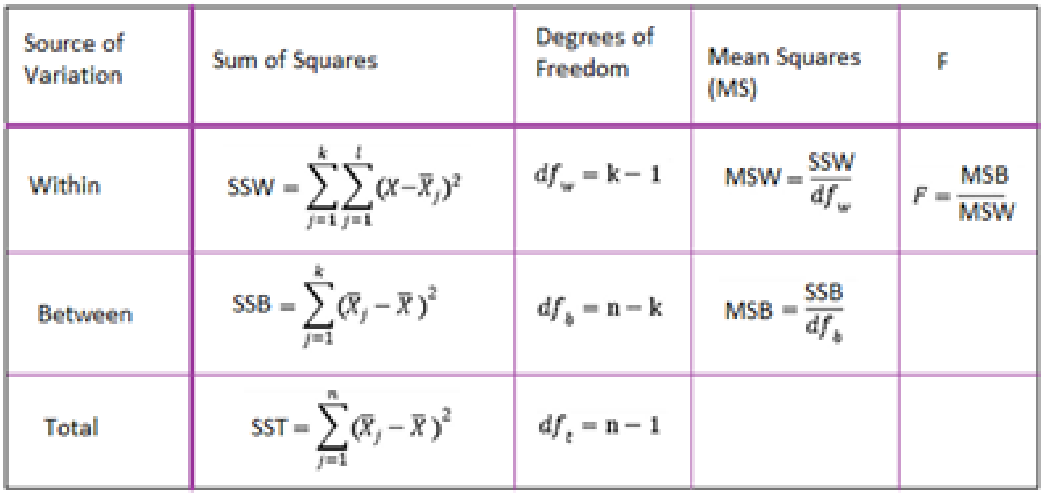
3 Theory

3.1 ANOVA (Analysis of variance)

3.1.1 Definition

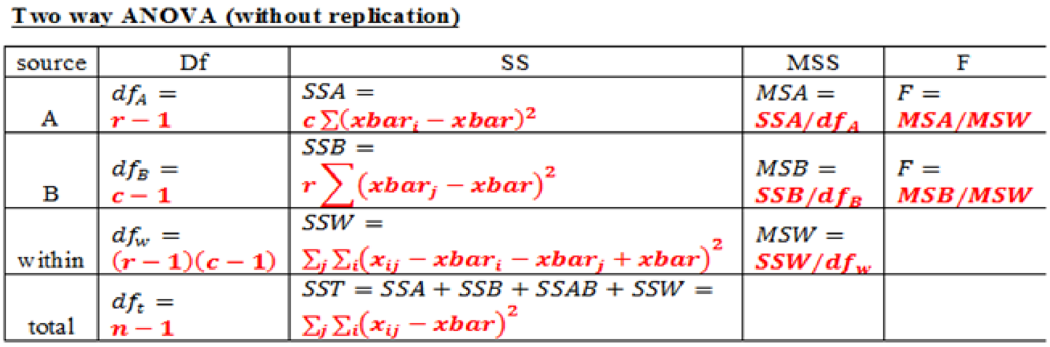
There are two types of ANOVA: Analysis of variance (ANOVA) is a collection of statistical models and the estimating processes that go with them. (such as the "variation" among and between groups) used to analyze the differences among means. ANOVA was developed by the statistician Ronald Fisher.

One-way ANOVA: The one-way ANOVA compares the means between the groups you are interested in and determines whether any of those means are statistically significantly different from each other. The null hypothesis (H0) is the homogeneity in all groups's means while the alternative hypothesis (H1) is a difference in at least one mean.



Source of Variation	Sum of Squares	Degrees of Freedom	Mean Squares (MS)	F
Within	$SSW = \sum_{j=1}^{k}\sum_{j=1}^{l}(X-\bar{X}_j)^2$	$df_w = k - 1$	$MSW = \frac{SSW}{df_w}$	$F = \frac{MSB}{MSW}$
Between	$SSB = \sum_{j=1}^{k}(\bar{X}_j - \bar{X})^2$	$df_b = n - k$	$MSB = \frac{SSB}{df_b}$	
Total	$SST = \sum_{j=1}^{n}(\bar{X}_j - \bar{X})^2$	$df_t = n - 1$		

Two-way ANOVA: When we want to see how two independent variables affect a dependent factor, we apply this technique.



Two way ANOVA (without replication)

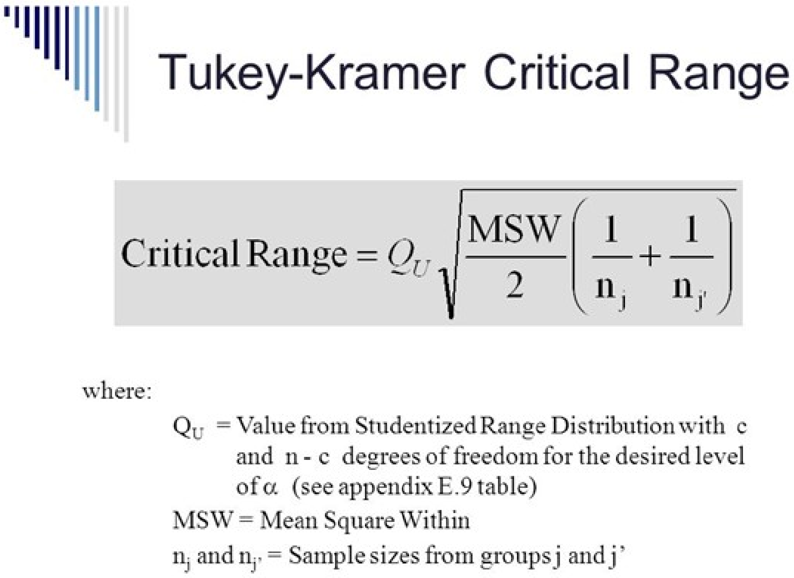
source	Df	SS	MSS	F
A	$df_A = r-1$	$SSA = c\sum(xbar_i - xbar)^2$	$MSA = SSA/df_A$	$F = MSA/MSW$
B	$df_B = c-1$	$SSB = r\sum(xbar_j - xbar)^2$	$MSB = SSB/df_B$	$F = MSB/MSW$
within	$df_w = (r-1)(c-1)$	$SSW = \sum_j\sum_i(x_{ij} - xbar_i - xbar_j + xbar)^2$	$MSW = SSW/df_w$	
total	$df_t = n-1$	$SST = SSA + SSB + SSAB + SSW = \sum_j\sum_i(x_{ij} - xbar)^2$		

3.2 Tukey's HSD (honestly significant difference)

3.2.1 Definition

The Tukey Test (or Tukey procedure), also called Tukey's Honest Significant Difference test, is a post-hoc test based on the studentized range distribution. An ANOVA test can tell you if your results are significant overall, but it won't tell you exactly where those differences lie. After you have run an ANOVA and found significant results, then you can run Tukey's HSD to find out which specific groups's means (compared with each other) are different. The test compares all possible pairs of means.

3.2.2 Equation



Tukey-Kramer Critical Range

$$\text{Critical Range} = Q_U\sqrt{\frac{\text{MSW}}{2}\left(\frac{1}{n_j}+\frac{1}{n_{j'}}\right)}$$

where:

Q_U = Value from Studentized Range Distribution with c and n - c degrees of freedom for the desired level of α (see appendix E.9 table)

MSW = Mean Square Within

n_j and $n_{j'}$ = Sample sizes from groups j and j'

3.3 Simple Linear Regression

3.3.1 Definition

In statistics, **simple linear regression** is a linear regression model with a single explanatory variable. That is, it concerns two-dimensional sample points with one independent variable and one dependent variable (conventionally, the x and y coordinates in a Cartesian coordinate system) and finds a linear function (a non-vertical straight line) that, as accurately as possible, predicts the dependent variable values as a function of the independent variable. The adjective simple refers to the fact that the outcome variable is related to a single predictor.

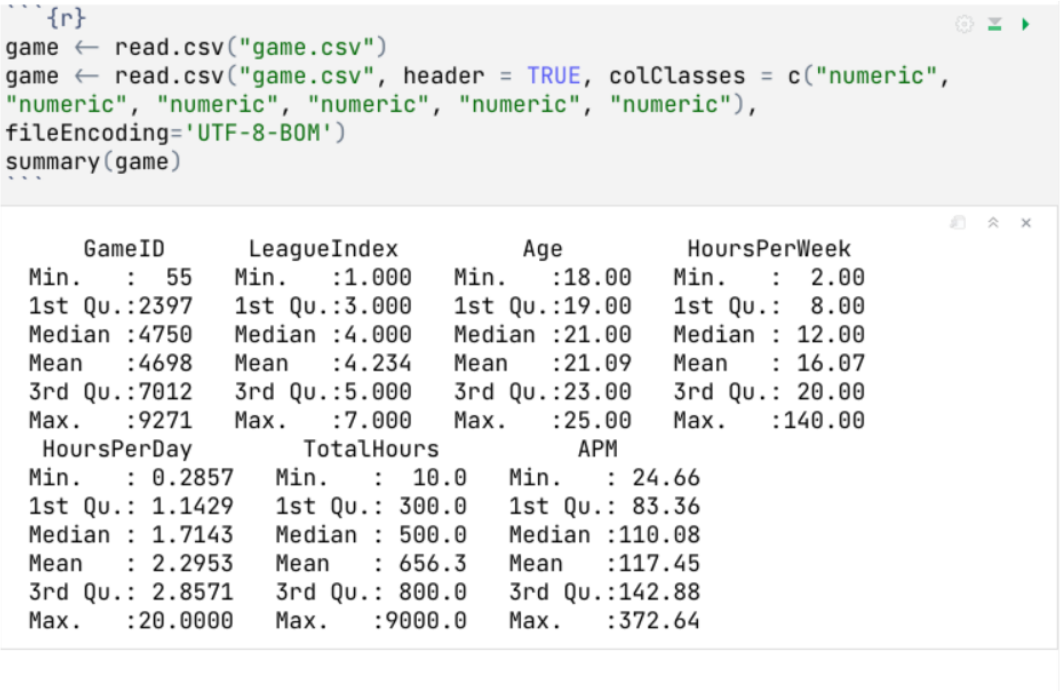
3.3.2 Equation

We assume that each observation, Y, can be described by the model:

$y_i = \alpha + \beta x_i + \epsilon_i$

4 Question 1

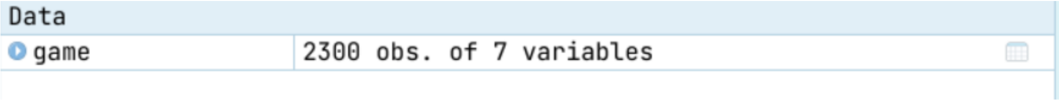
We need to import data into R. We will use <- operator to import data to "game" object. After that, we use "summary()" to get a summary of data



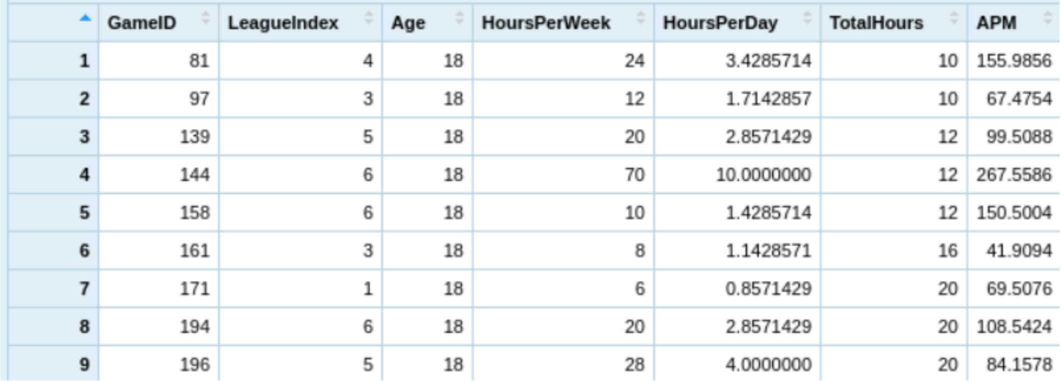
~~~
```{r}
game ← read.csv("game.csv")
game ← read.csv("game.csv", header = TRUE, colClasses = c("numeric",
"numeric", "numeric", "numeric", "numeric", "numeric"),
fileEncoding='UTF-8-BOM')
summary(game)
```
~~~

```
     GameID      LeagueIndex         Age          HoursPerWeek   
 Min.   :  55   Min.   :1.000   Min.   :18.00   Min.   :  2.00  
 1st Qu.:2397   1st Qu.:3.000   1st Qu.:19.00   1st Qu.:  8.00  
 Median :4750   Median :4.000   Median :21.00   Median : 12.00  
 Mean   :4698   Mean   :4.234   Mean   :21.09   Mean   : 16.07  
 3rd Qu.:7012   3rd Qu.:5.000   3rd Qu.:23.00   3rd Qu.: 20.00  
 Max.   :9271   Max.   :7.000   Max.   :25.00   Max.   :140.00  
  HoursPerDay        TotalHours          APM        
 Min.   : 0.2857   Min.   :  10.0   Min.   : 24.66  
 1st Qu.: 1.1429   1st Qu.: 300.0   1st Qu.: 83.36  
 Median : 1.7143   Median : 500.0   Median :110.08  
 Mean   : 2.2953   Mean   : 656.3   Mean   :117.45  
 3rd Qu.: 2.8571   3rd Qu.: 800.0   3rd Qu.:142.88  
 Max.   :20.0000   Max.   :9000.0   Max.   :372.64  
```

After that, the environment tab will display like this



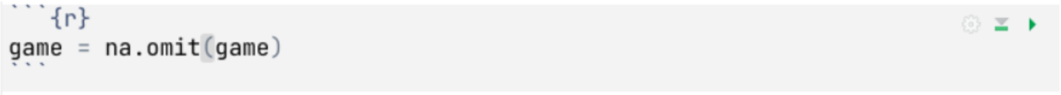
```
Data
game        2300 obs. of 7 variables
```



	GameID	LeagueIndex	Age	HoursPerWeek	HoursPerDay	TotalHours	APM
1	81	4	18	24	3.4285714	10	155.9856
2	97	3	18	12	1.7142857	10	67.4754
3	139	5	18	20	2.8571429	12	99.5088
4	144	6	18	70	10.0000000	12	267.5586
5	158	6	18	10	1.4285714	12	150.5004
6	161	3	18	8	1.1428571	16	41.9094
7	171	1	18	6	0.8571429	20	69.5076
8	194	6	18	20	2.8571429	20	108.5424
9	196	5	18	28	4.0000000	20	84.1578

5 Question 2

We use "na.omit()" to clean up the data



~~~
```{r}
game = na.omit(game)
```
~~~

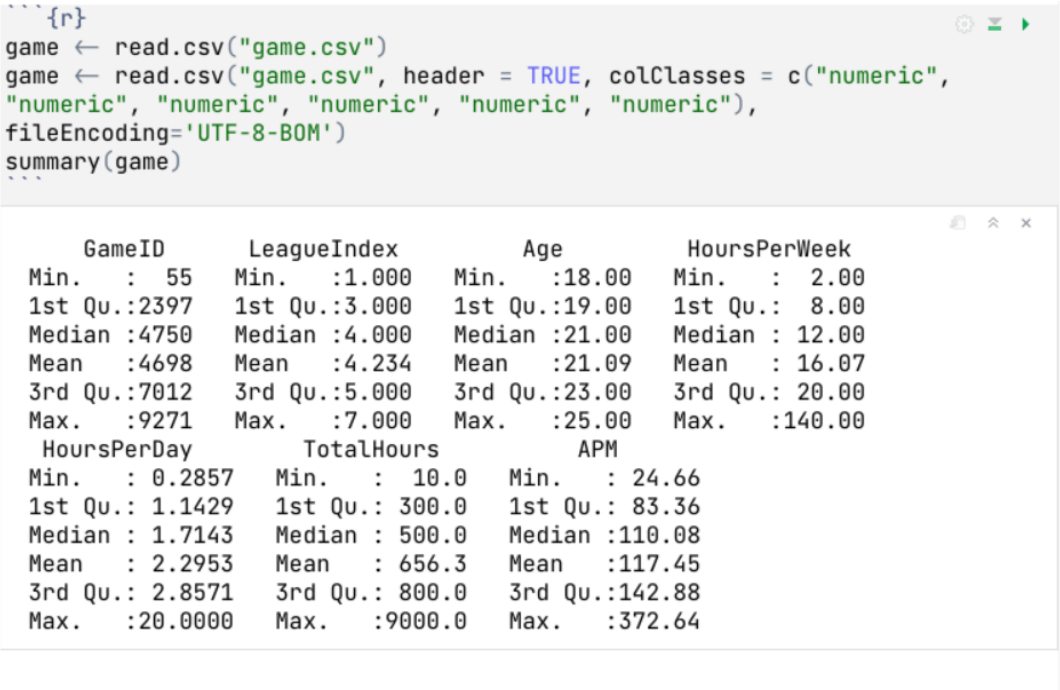
After cleaning, the dataset remains the same, which means the dataset is already clean

6 Question 3

We will do data visualization

6.1 Descriptive data

To get a descriptive statistic for each variables, we will use "summary()"

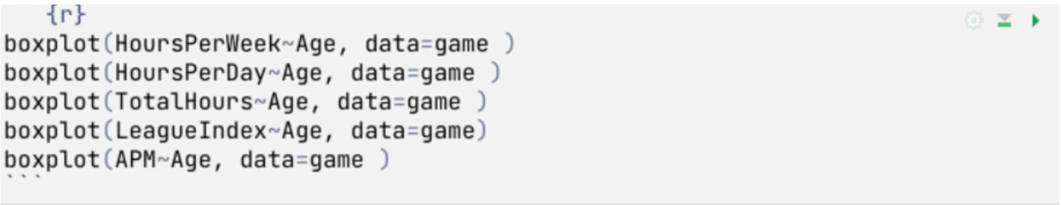


~~~
```{r}
game ← read.csv("game.csv")
game ← read.csv("game.csv", header = TRUE, colClasses = c("numeric",
"numeric", "numeric", "numeric", "numeric", "numeric"),
fileEncoding='UTF-8-BOM')
summary(game)
```
~~~

```
     GameID       LeagueIndex         Age         HoursPerWeek
 Min.   :  55   Min.   :1.000   Min.   :18.00   Min.   :  2.00
 1st Qu.:2397   1st Qu.:3.000   1st Qu.:19.00   1st Qu.:  8.00
 Median :4750   Median :4.000   Median :21.00   Median : 12.00
 Mean   :4698   Mean   :4.234   Mean   :21.09   Mean   : 16.07
 3rd Qu.:7012   3rd Qu.:5.000   3rd Qu.:23.00   3rd Qu.: 20.00
 Max.   :9271   Max.   :7.000   Max.   :25.00   Max.   :140.00
  HoursPerDay         TotalHours          APM
 Min.   : 0.2857   Min.   :  10.0   Min.   : 24.66
 1st Qu.: 1.1429   1st Qu.: 300.0   1st Qu.: 83.36
 Median : 1.7143   Median : 500.0   Median :110.08
 Mean   : 2.2953   Mean   : 656.3   Mean   :117.45
 3rd Qu.: 2.8571   3rd Qu.: 800.0   3rd Qu.:142.88
 Max.   :20.0000   Max.   :9000.0   Max.   :372.64
```

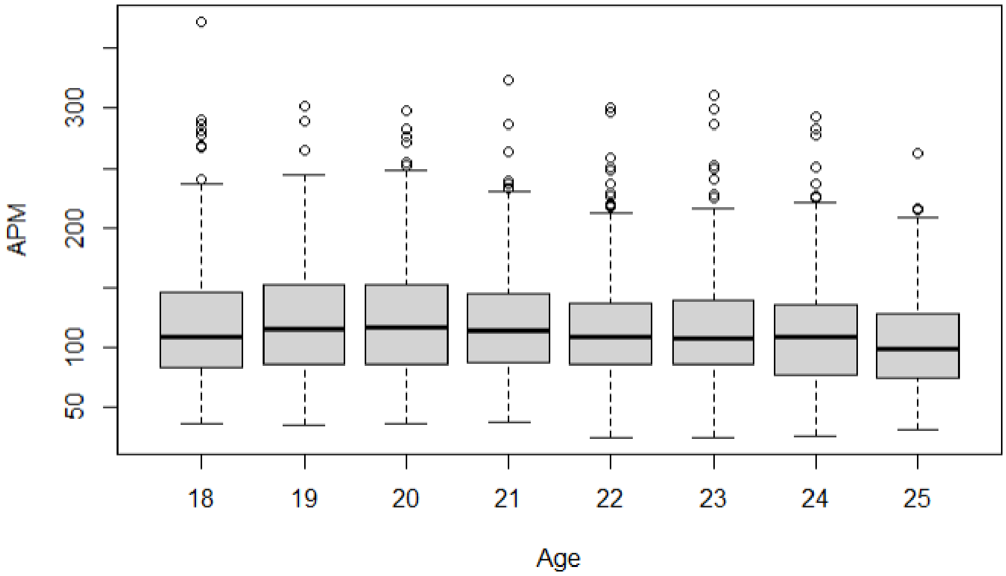
6.2 Graph

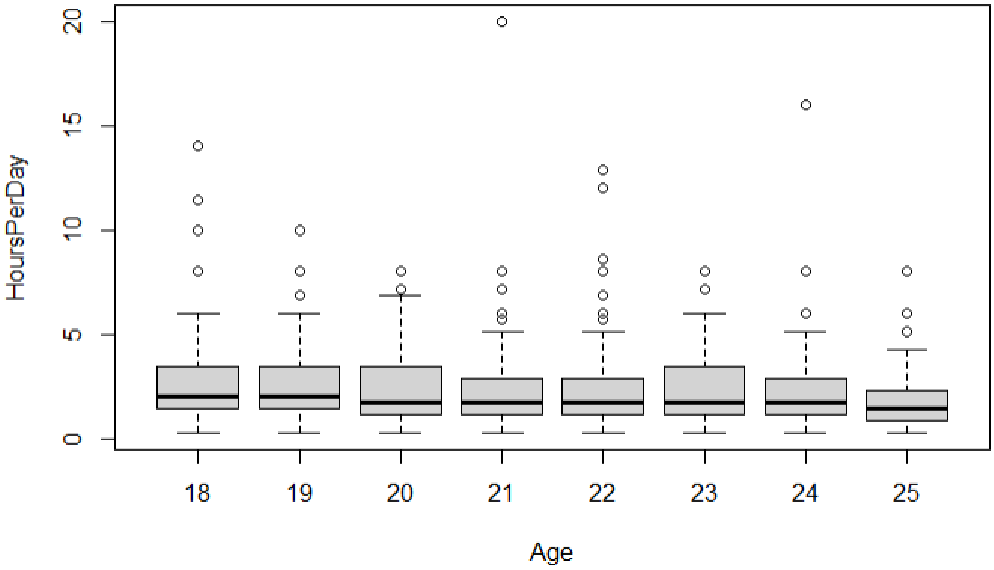
To draw box graph, we will use the "boxplot()" function

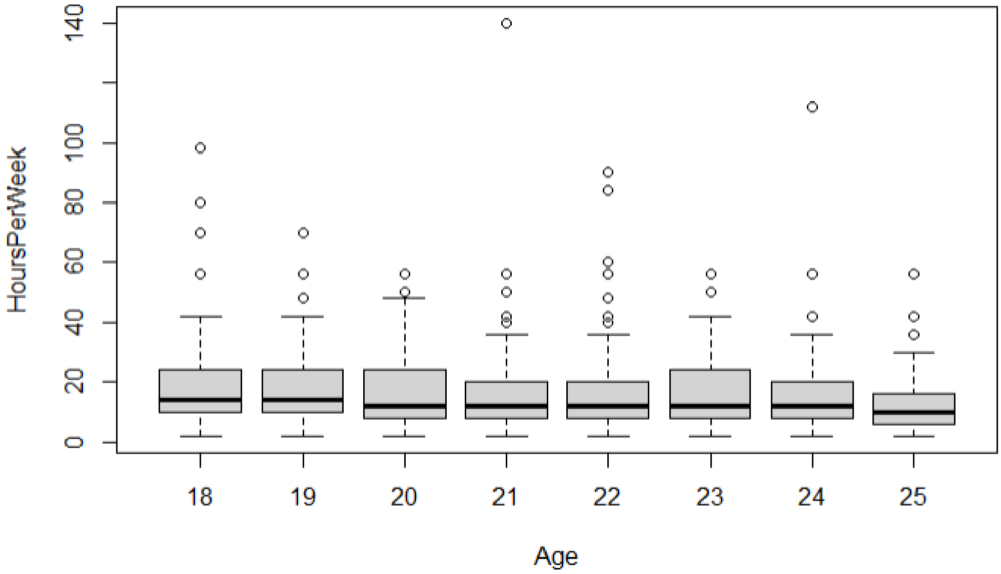


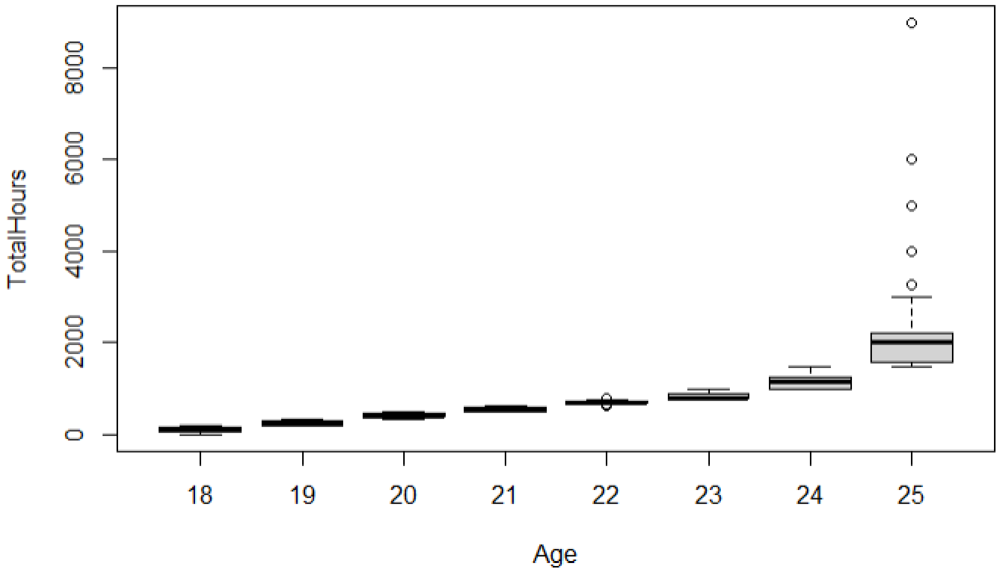
~~~
   {r}
boxplot(HoursPerWeek~Age, data=game )
boxplot(HoursPerDay~Age, data=game )
boxplot(TotalHours~Age, data=game )
boxplot(LeagueIndex~Age, data=game)
boxplot(APM~Age, data=game )
```
~~~

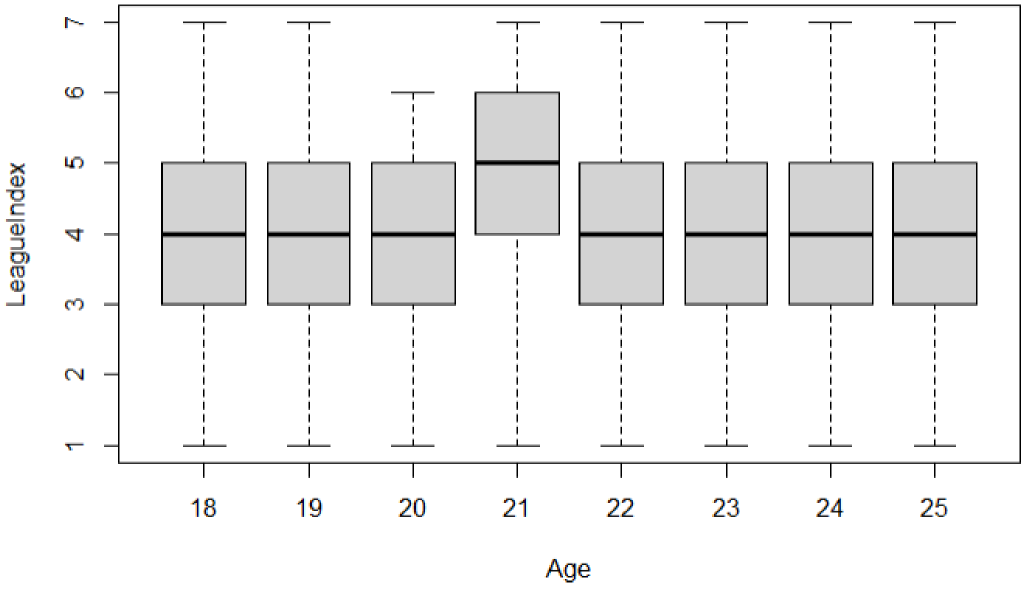
The box graph visualize the relationship between variable. The rectangular box is the interquartile range. The line cross the box shows the maximum and minimum value of the data



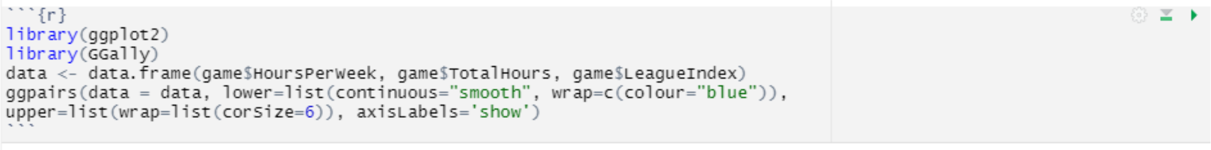

HoursPerDay
0
5
10
15
20
18
19
20
21
22
23
24
25
Age


HoursPerWeek
0
20
40
60
80
100
140
18
19
20
21
22
23
24
25
Age

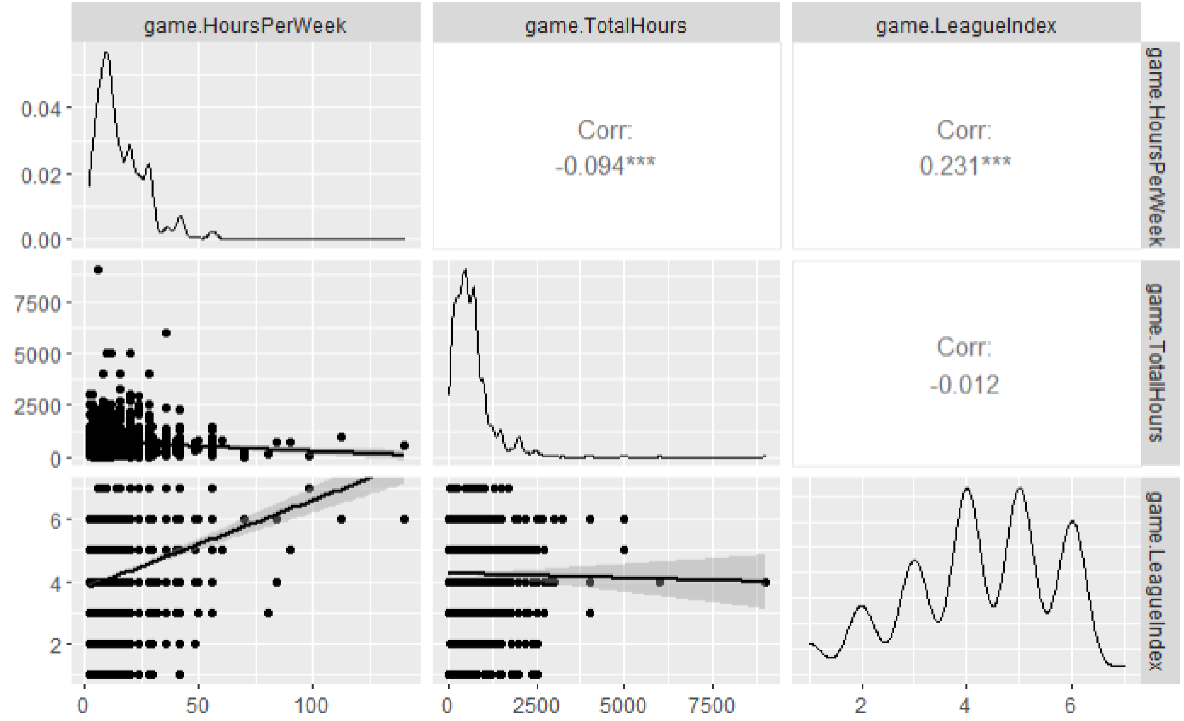




To draw a pair we use ggpairs()

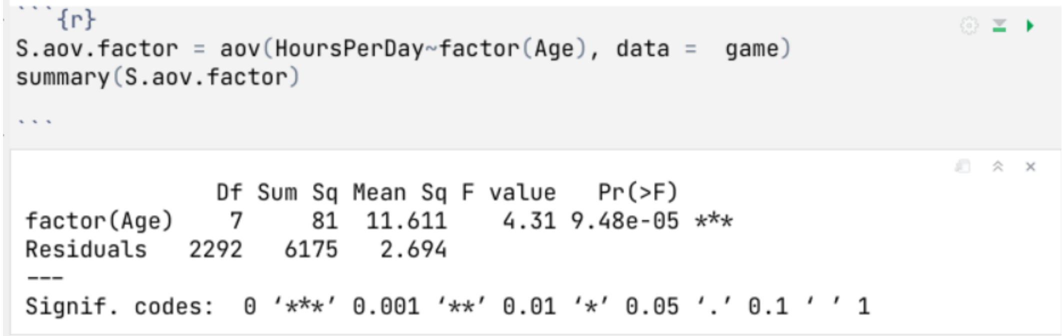


~~~
```{r}
library(ggplot2)
library(GGally)
data <- data.frame(game$HoursPerWeek, game$TotalHours, game$LeagueIndex)
ggpairs(data = data, lower=list(continuous="smooth", wrap=c(colour="blue")),
upper=list(wrap=list(corSize=6)), axisLabels='show')
```
~~~


game.HoursPerWeek
game.TotalHours
game.LeagueIndex
Corr:
-0.094***
Corr:
0.231***
Corr:
-0.012
0.04
0.02
0.00
7500
5000
2500
0
6
4
2
0
50
100
0
2500
5000
7500
2
4
6

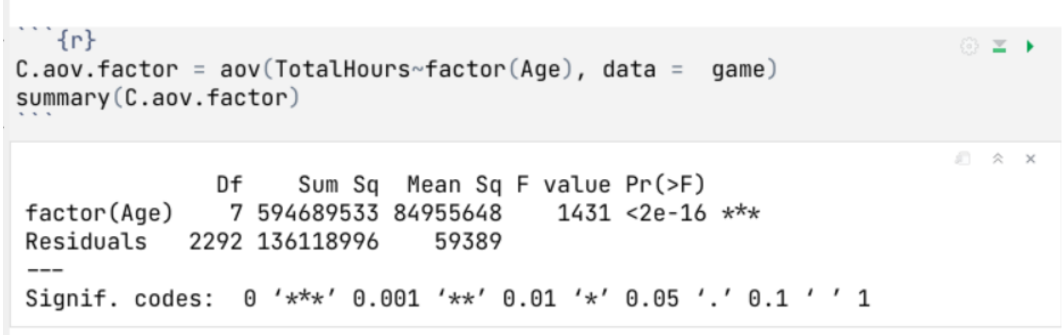
7 Question 4

We carry out the ANOVA test with the "aov" function



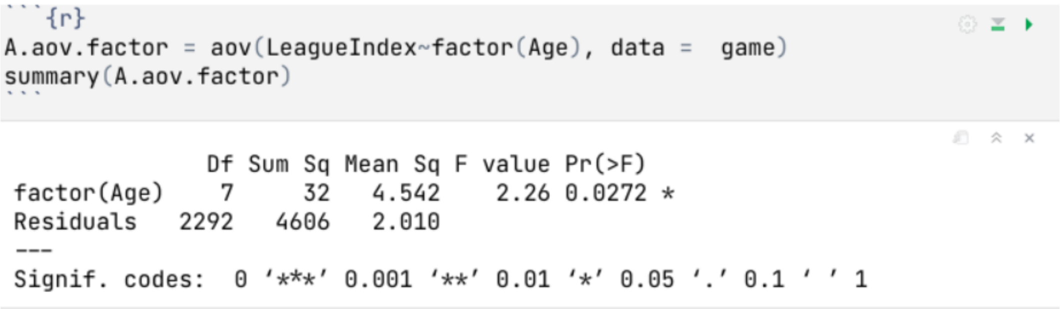
~~~
```{r}
S.aov.factor = aov(HoursPerDay~factor(Age), data =  game)
summary(S.aov.factor)

```
~~~

```
               Df Sum Sq Mean Sq F value   Pr(>F)
factor(Age)     7     81  11.611    4.31 9.48e-05 ***
Residuals    2292   6175   2.694
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```



~~~
```{r}
C.aov.factor = aov(TotalHours~factor(Age), data =  game)
summary(C.aov.factor)
```
~~~

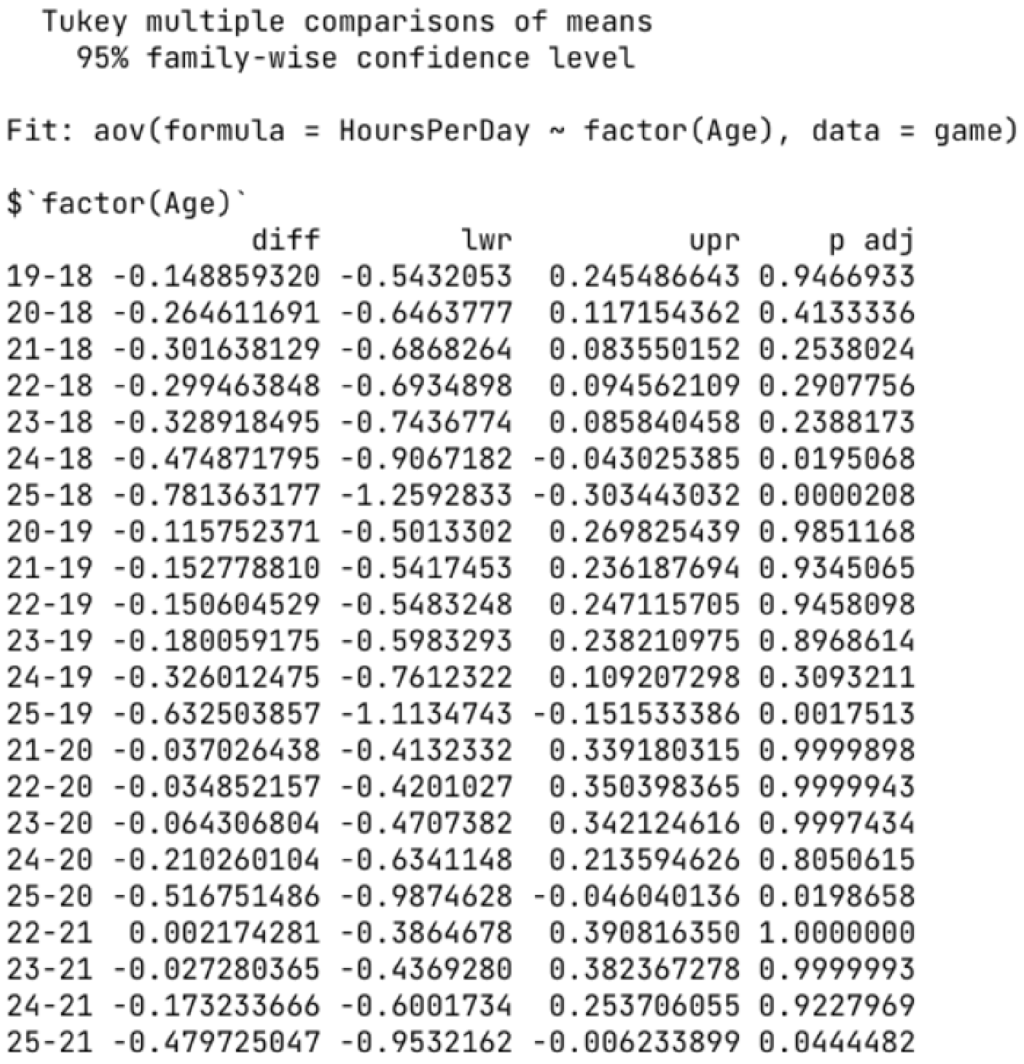
```
               Df    Sum Sq  Mean Sq F value Pr(>F)
factor(Age)     7 594689533 84955648    1431 <2e-16 ***
Residuals    2292 136118996    59389
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```



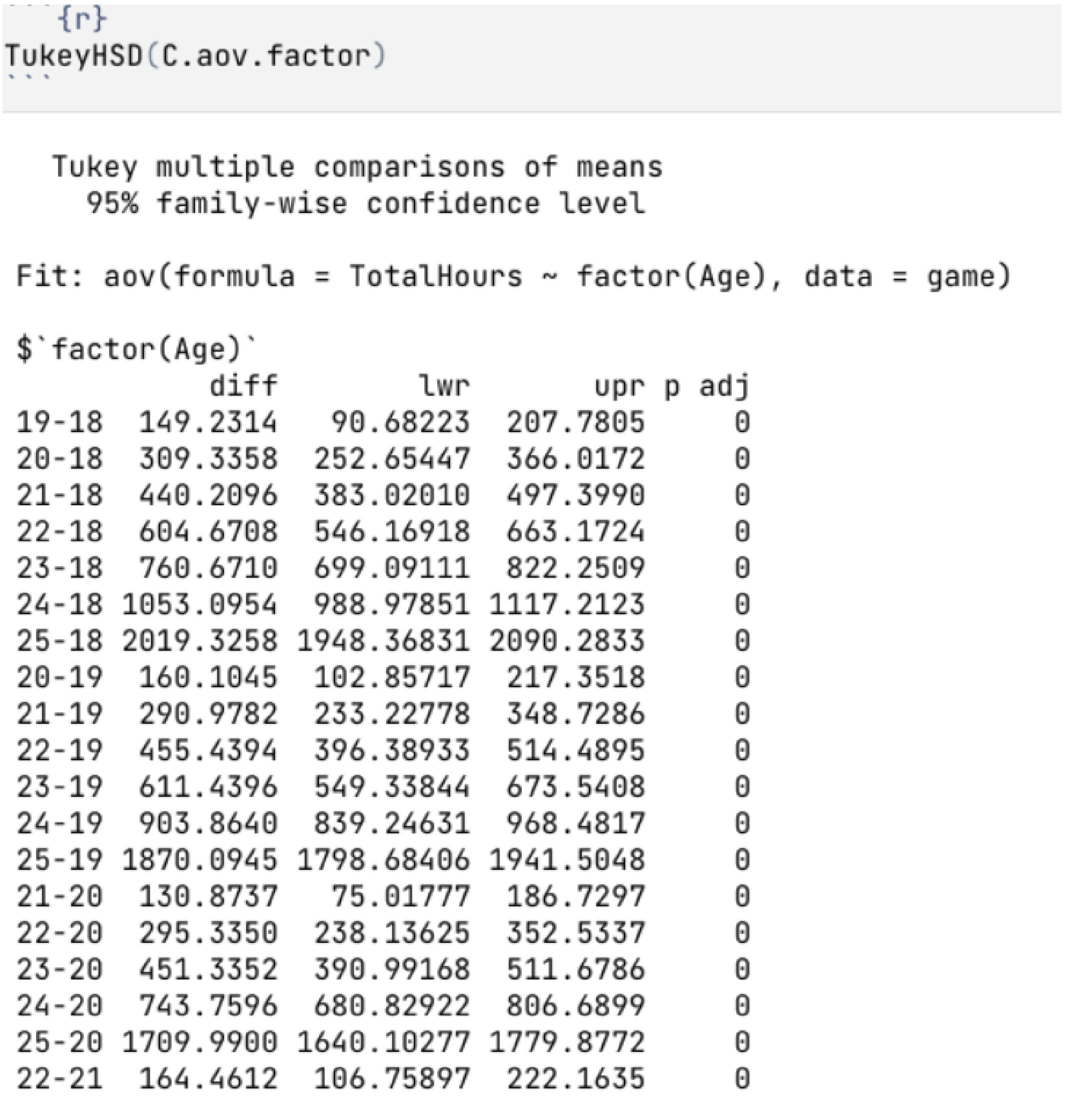
~~~
```{r}
A.aov.factor = aov(LeagueIndex~factor(Age), data =  game)
summary(A.aov.factor)
```
~~~

```
               Df Sum Sq Mean Sq F value Pr(>F)
factor(Age)     7     32   4.542    2.26 0.0272 *
Residuals    2292   4606   2.010
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Then we will carry out the Tukey's HSD with the "TukeyHSD()" function

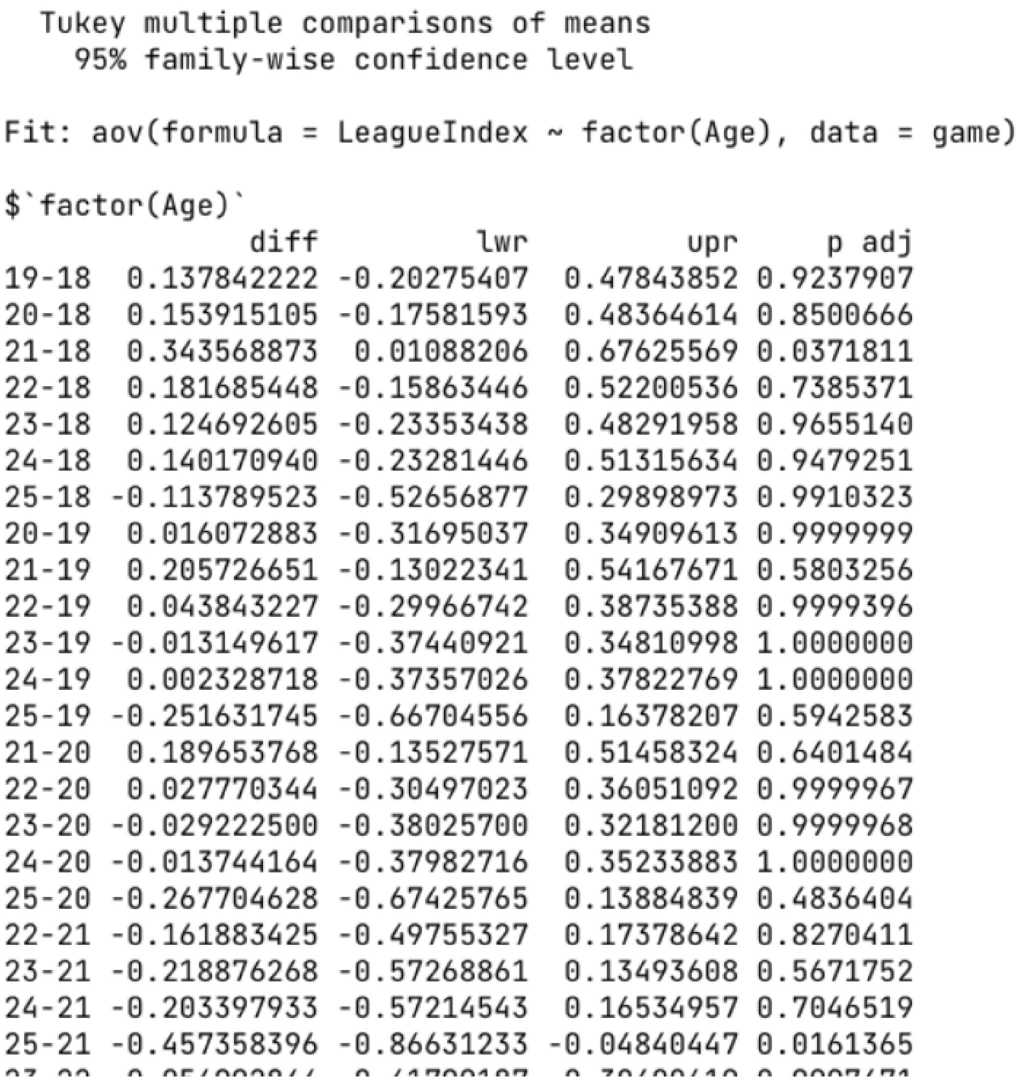


```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = HoursPerDay ~ factor(Age), data = game)

$`factor(Age)`
              diff        lwr          upr     p adj
19-18 -0.148859320 -0.5432053  0.245486643 0.9466933
20-18 -0.264611691 -0.6463777  0.117154362 0.4133336
21-18 -0.301638129 -0.6868264  0.083550152 0.2538024
22-18 -0.299463848 -0.6934898  0.094562109 0.2907756
23-18 -0.328918495 -0.7436774  0.085840458 0.2388173
24-18 -0.474871795 -0.9067182 -0.043025385 0.0195068
25-18 -0.781363177 -1.2592833 -0.303443032 0.0000208
20-19 -0.115752371 -0.5013302  0.269825439 0.9851168
21-19 -0.152778810 -0.5417453  0.236187694 0.9345065
22-19 -0.150604529 -0.5483248  0.247115705 0.9458098
23-19 -0.180059175 -0.5983293  0.238210975 0.8968614
24-19 -0.326012475 -0.7612322  0.109207298 0.3093211
25-19 -0.632503857 -1.1134743 -0.151533386 0.0017513
21-20 -0.037026438 -0.4132332  0.339180315 0.9999898
22-20 -0.034852157 -0.4201027  0.350398365 0.9999943
23-20 -0.064306804 -0.4707382  0.342124616 0.9997434
24-20 -0.210260104 -0.6341148  0.213594626 0.8050615
25-20 -0.516751486 -0.9874628 -0.046040136 0.0198658
22-21  0.002174281 -0.3864678  0.390816350 1.0000000
23-21 -0.027280365 -0.4369280  0.382367278 0.9999993
24-21 -0.173233666 -0.6001734  0.253706055 0.9227969
25-21 -0.479725047 -0.9532162 -0.006233899 0.0444482
```



~~~
```{r}
TukeyHSD(C.aov.factor)
```
~~~

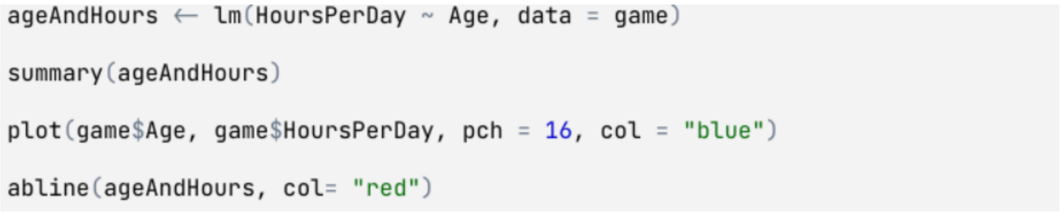
```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = TotalHours ~ factor(Age), data = game)

$`factor(Age)`
           diff        lwr       upr p adj
19-18  149.2314   90.68223  207.7805     0
20-18  309.3358  252.65447  366.0172     0
21-18  440.2096  383.02010  497.3990     0
22-18  604.6708  546.16918  663.1724     0
23-18  760.6710  699.09111  822.2509     0
24-18 1053.0954  988.97851 1117.2123     0
25-18 2019.3258 1948.36831 2090.2833     0
20-19  160.1045  102.85717  217.3518     0
21-19  290.9782  233.22778  348.7286     0
22-19  455.4394  396.38933  514.4895     0
23-19  611.4396  549.33844  673.5408     0
24-19  903.8640  839.24631  968.4817     0
25-19 1870.0945 1798.68406 1941.5048     0
21-20  130.8737   75.01777  186.7297     0
22-20  295.3350  238.13625  352.5337     0
23-20  451.3352  390.99168  511.6786     0
24-20  743.7596  680.82922  806.6899     0
25-20 1709.9900 1640.10277 1779.8772     0
22-21  164.4612  106.75897  222.1635     0
```



```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = LeagueIndex ~ factor(Age), data = game)

$`factor(Age)`
              diff         lwr         upr     p adj
19-18  0.137842222 -0.20275407  0.47843852 0.9237907
20-18  0.153915105 -0.17581593  0.48364614 0.8500666
21-18  0.343568873  0.01088206  0.67625569 0.0371811
22-18  0.181685448 -0.15863446  0.52200536 0.7385371
23-18  0.124692605 -0.23353438  0.48291958 0.9655140
24-18  0.140170940 -0.23281446  0.51315634 0.9479251
25-18 -0.113789523 -0.52656877  0.29898973 0.9910323
20-19  0.016072883 -0.31695037  0.34909613 0.9999999
21-19  0.205726651 -0.13022341  0.54167671 0.5803256
22-19  0.043843227 -0.29966742  0.38735388 0.9999396
23-19 -0.013149617 -0.37440921  0.34810998 1.0000000
24-19  0.002328718 -0.37357026  0.37822769 1.0000000
25-19 -0.251631745 -0.66704556  0.16378207 0.5942583
21-20  0.189653768 -0.13527571  0.51458324 0.6401484
22-20  0.027770344 -0.30497023  0.36051092 0.9999967
23-20 -0.029222500 -0.38025700  0.32181200 0.9999968
24-20 -0.013744164 -0.37982716  0.35233883 1.0000000
25-20 -0.267704628 -0.67425765  0.13884839 0.4836404
22-21 -0.161883425 -0.49755327  0.17378642 0.8270411
23-21 -0.218876268 -0.57268861  0.13493608 0.5671752
24-21 -0.203397933 -0.57214543  0.16534957 0.7046519
25-21 -0.457358396 -0.86631233 -0.04840447 0.0161365
```

8 Question 5

We will generalize linear model with the following code

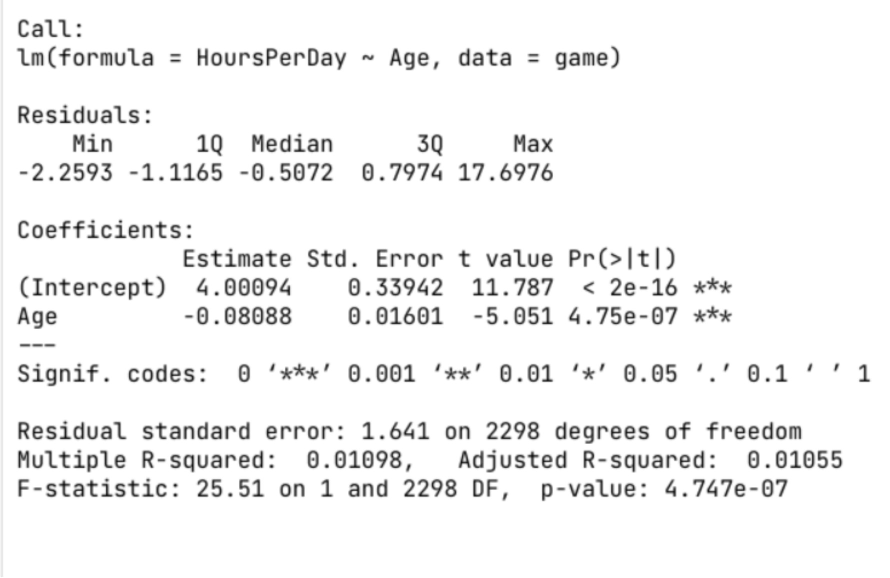


```
ageAndHours <- lm(HoursPerDay ~ Age, data = game)

summary(ageAndHours)

plot(game$Age, game$HoursPerDay, pch = 16, col = "blue")

abline(ageAndHours, col= "red")
```

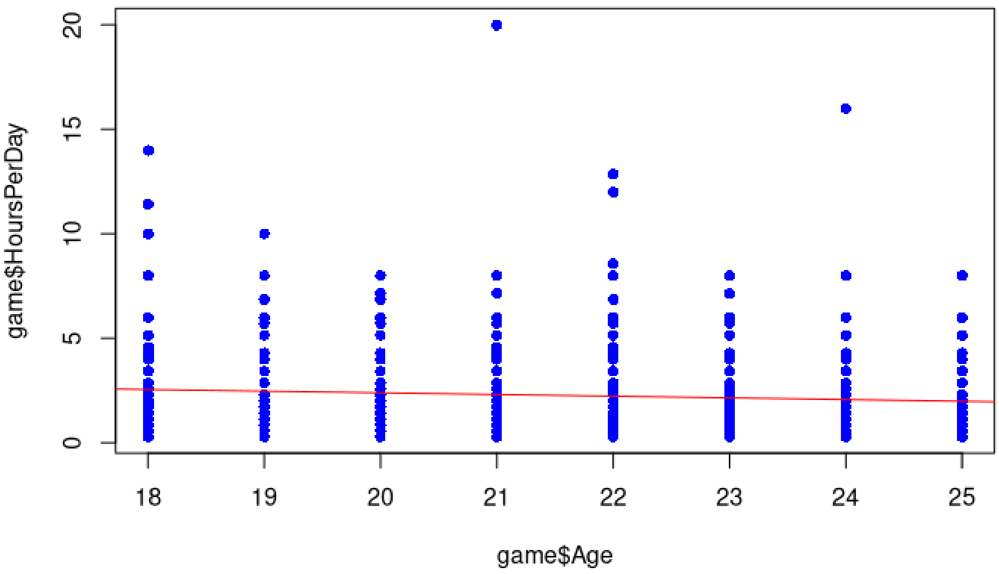
Explanation

- "lm()" limits the data set to only hoursPerDay and age
- "summary()" will summarize the data
- "plot()" plot the data
- "abline" will draw the linear regression graph

The results



```
Call:
lm(formula = HoursPerDay ~ Age, data = game)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.2593 -1.1165 -0.5072  0.7974 17.6976 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  4.00094    0.33942  11.787  < 2e-16 ***
Age         -0.08088    0.01601  -5.051 4.75e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.641 on 2298 degrees of freedom
Multiple R-squared:  0.01098,	Adjusted R-squared:  0.01055 
F-statistic: 25.51 on 1 and 2298 DF,  p-value: 4.747e-07
```



9 Conclusion

From the analysis we can conclude that

- On average people spent similar amount of time peer weeks and peer day
- Older group spent more time to play games
- On average, they have similar APM
- Young people can score higher APM score.