Chess Study

Briksam Kasımoğlu 2102969
Firas Elbayoumi 2100503
Yaser Z. K. Shoshaa 2104449

INE2002 PROJECT

Table of Contents

Chess Study..1
1.Study Aim:..4
2.Data Description:..4
2.1.Requirements: 4
2.2.Data collection and sampling:..4
2.3.Reading the Data:..4
2.4.First 5 instances of the population:..5
2.5.New column:..5
3.Data visualization 6
3.1. Sample 6
3.2. Population 7
4.Normality tests 8
4.1.Pearson Coefficient and qqplot...8
5. Point estimations and confidence intervals...9
5.1.Point estimations 9
5.2.Confidence Intervals..9
6.Hypothesis Testing. 11
7.Linear Regression. 13
7.1.First model. 14
7.2.Second model: 16
8.Chi Square Tests. 17
8.1.Goodness of fit 17
8.2.Independence test 18
8.3. Homogeneity test 19
9.Anova..20
9.1. One way Anova 20

9.2.Two way Anova....................21
10.Non-parametric tests. 22
10.1. Sign test. 22
10.2.Wilcox rank sum test. 23
10.3.Spearman correlation. 23
10.4.Runs test 24
References 24

1.Study Aim:

The aim of this study is to analyze the effects of the number of moves, white player rating, black player rating, average rating, victory-status, and winner on each other and on the game's outcome as a whole, to achieve this a set of statistical tools are going to be applied.

2.Data Description:

2.1.Requirements:

```r
library(tidyr)
library(ggplot2)
library(dplyr)
library(stringr)
library(DescTools)
library(corrplot)
library("BSDA")
library(randtests)
```

2.2.Data collection and sampling:

The dataset we used can be found online on the Kaggle website.
It is collected from a chess platform called Lichess with the help of the platform's own API.
A sample of 150 from a population of 10,759 (instead of 20,058) was taken because some rows were dropped, the sampling technique applied is systematic sampling. The dataset has 16 columns not all of them are used in this study, here is a quick overview of the used columns:
rated (true/false), start time, end time, turns, victory status,
winner, white player rating,
black player rating, opening ply (number of moves in the opening part).

2.3.Reading the Data:

```r
games = read.csv('games.csv')
View(games)
```

2.4.First 5 instances of the population:

id	rated	created_at	last_move_at	turns	victory_status	winner	increment_code	white_id	white_rating	black_id
TZJHLljE	FALSE	1.50421e+12	1.50421e+12	13	outoftime	white	15+2	bourgris	1500	a-00
l1NXvwaE	TRUE	1.50413e+12	1.50413e+12	16	resign	black	5+10	a-00	1322	skinnerua
mIICvQHh	TRUE	1.50413e+12	1.50413e+12	61	mate	white	5+10	ischia	1496	a-00
kWKvrqYL	TRUE	1.50411e+12	1.50411e+12	61	mate	white	20+0	daniamurashov	1439	adivanov2009
9tXo1AUZ	TRUE	1.50403e+12	1.50403e+12	95	mate	white	30+3	nik221107	1523	adivanov2009

black_rating	moves	opening_eco	opening_name	opening_ply
1191	d4 d5 c4 c6 cxd5 e6 dxe6 fxe6 Nf3 Bb4+ Nc3 Ba5 Bf4	D10	Slav Defense: Exchange Variation	5
1261	d4 Nc6 e4 e5 f4 f6 dxe5 fxe5 fxe5 Nxe5 Qd4 Nc6 Qe5+ Nxe...	B00	Nimzowitsch Defense: Kennedy Variation	4
1500	e4 e5 d3 d6 Be3 c6 Be2 b5 Nd2 a5 a4 c5 axb5 Nc6 bxc6 Ra6...	C20	King's Pawn Game: Leonardis Variation	3
1454	d4 d5 Nf3 Bf5 Nc3 Nf6 Bf4 Ng4 e3 Nc6 Be2 Qd7 O-O O-O-...	D02	Queen's Pawn Game: Zukertort Variation	3
1469	e4 e5 Nf3 d6 d4 Nc6 d5 Nb4 a3 Na6 Nc3 Be7 b4 Nf6 Bg5 O...	C41	Philidor Defense	5

2.5.New column:

A decision was made to curate a new column called average rating this column is the average of the white player rating and the black player rating for each game.

```
games$avg_rating = (games$white_rating + games$black_rating) / 2
```

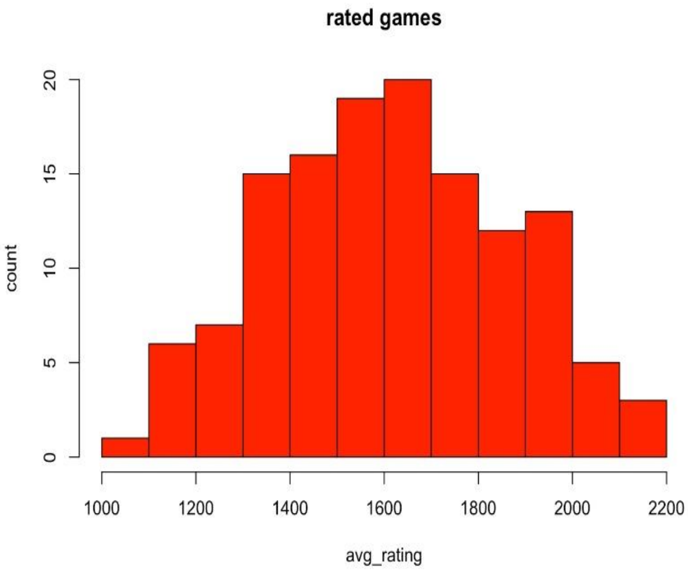
In addition to creating a new column called game_length where start time and end time where used to calculate it. After the calculation of game_length, it was realized that some datapoints had no game_length (no time was calculated) and so we dropped those rows.

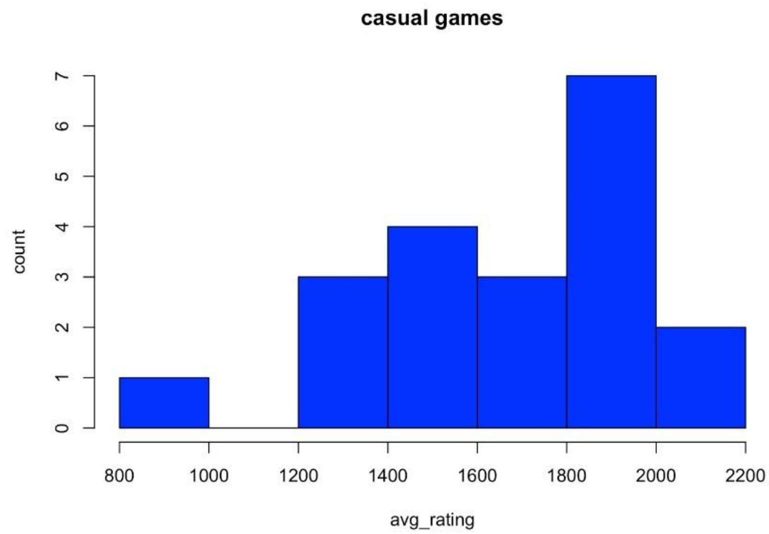
```
games$game_length = games$last_move_at/1000 - games$created_at/1000
games = games[9300:nrow(games),]
```

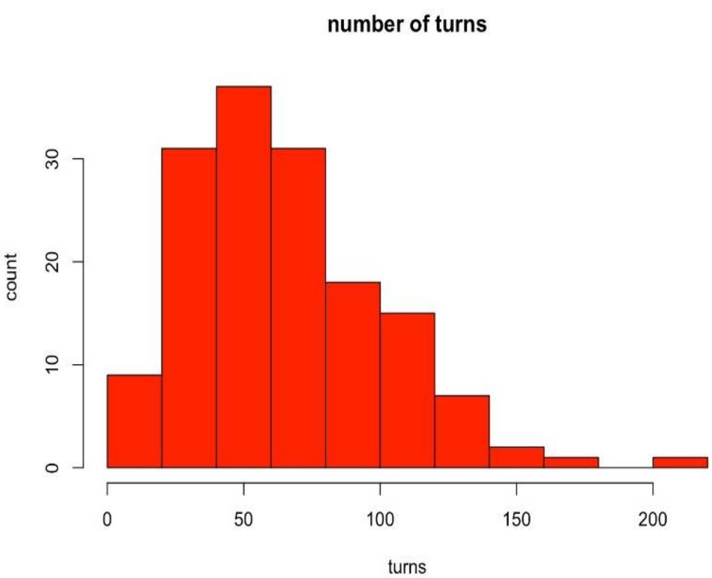
3.Data visualization

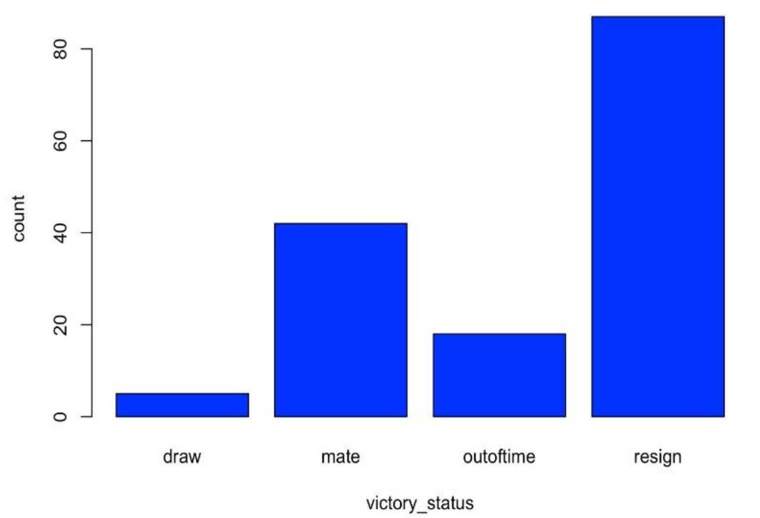
3.1. Sample

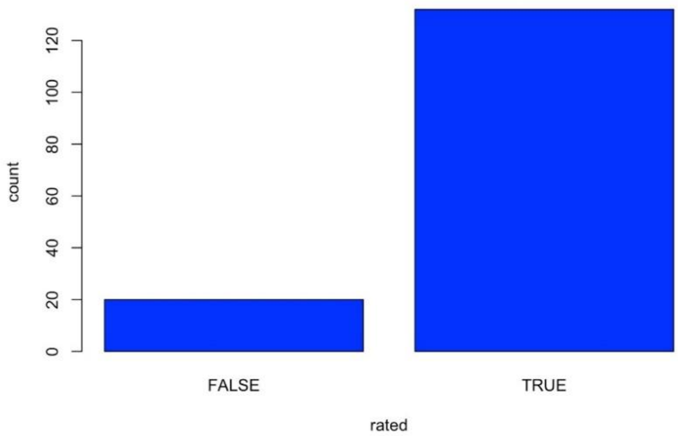
(Casual means unrated)

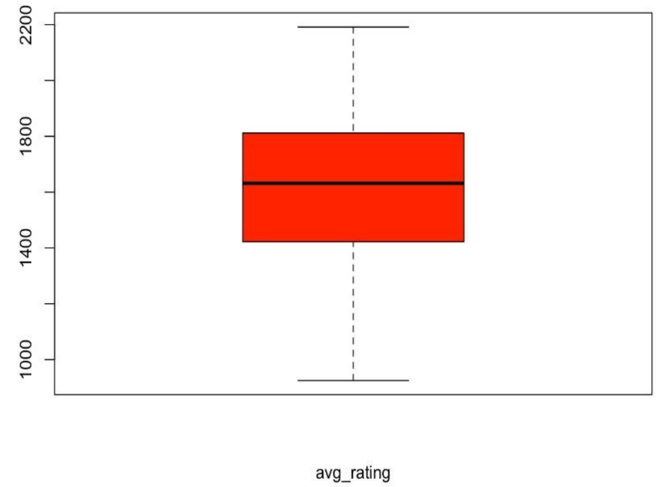


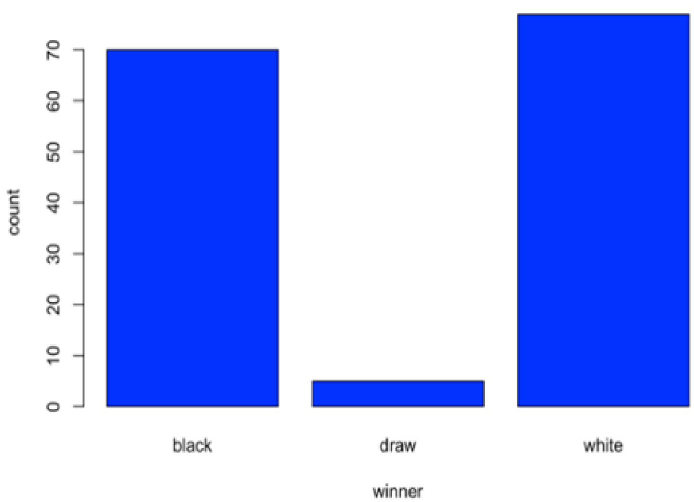


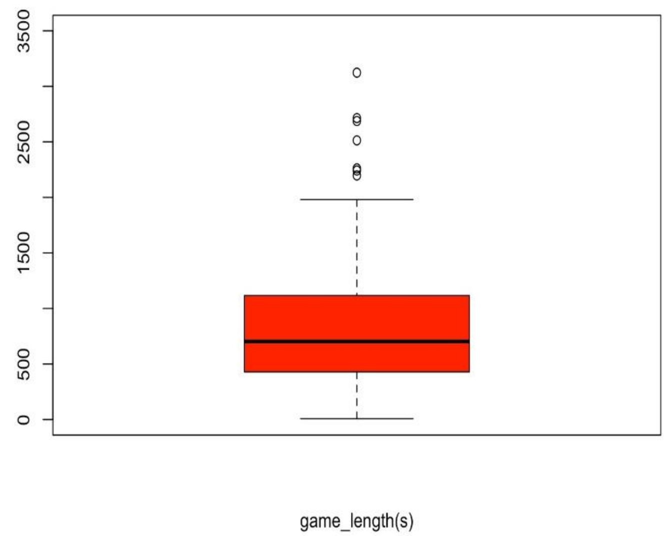




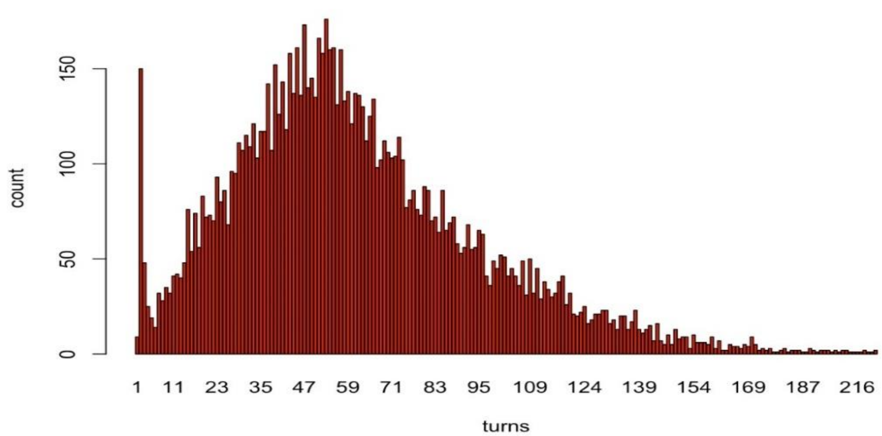








3.2. Population

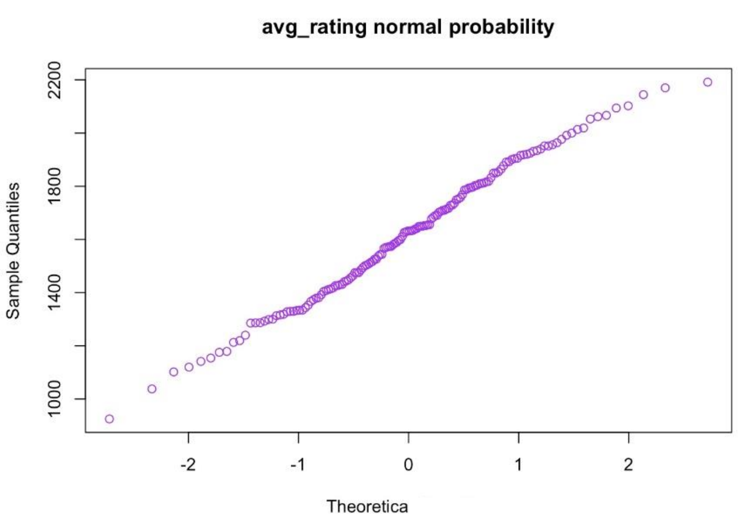


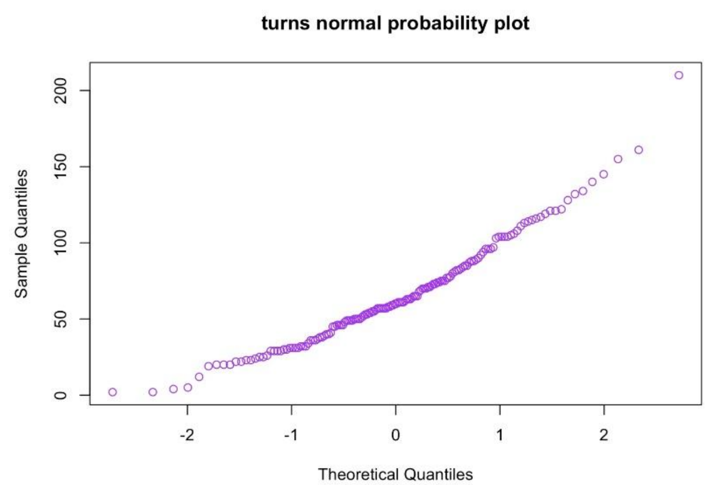
4.Normality tests

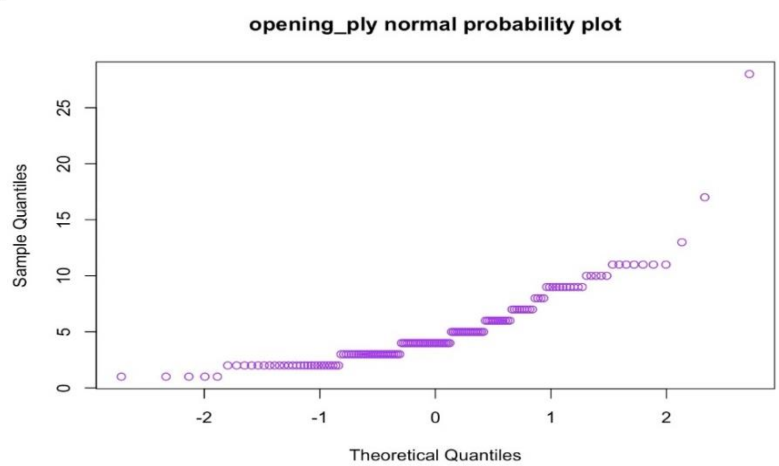
4.1.Pearson Coefficient and qqplot.

To test for normality, the function qqplot (quantile-quantile plot) was used where it plots the relation between a sample and the normal distribution. Another test for normality was also applied called The Pearson Coefficient (PC) of Skewness the results for this test can be evaluated as such:
If the index is greater than or equal to +1 or less than or equal to –1, it can be concluded that the data is significantly skewed (normality is questionable).

```
> qqnorm(gamesSample$avg_rating,col='purple', main='avg_rating normal probability plot')
> 3*(mean(gamesSample$avg_rating) - median(gamesSample$avg_rating)) / sd(gamesSample$avg_rating)
[1] -0.1424546
> qqnorm(gamesSample$turns,col='purple', main='turns normal probability plot')
> 3*(mean(gamesSample$turns) - median(gamesSample$turns)) / sd(gamesSample$turns)
[1] 0.4704973
> qqnorm(gamesSample$opening_ply,col='purple', main='opening_ply normal probability plot')
> 3*(mean(gamesSample$opening_ply) - median(gamesSample$opening_ply)) / sd(gamesSample$opening_ply)
[1] 0.9664849
```







As seen in the scores of PC and the plots the variables avg_rating and turns are approximately normally distributed and opening_ply is significantly skewed.

5. Point estimations and confidence intervals.

5.1.Point estimations

Sample means and sample standard deviations were calculated and compared with their population counterparts. In order to calculate population std a function was curated as well.

```
> pop_sd = function(data){
+   return(sqrt(var(data) * ((length(data) - 1) / length(data))))
+ }
> cat("mean population opening_ply:", mean(games$opening_ply), ", mean sample opening_ply:",mean(gamesSample$opening_ply))
mean population opening_ply: 4.926852 , mean sample opening_ply: 5.118421
> cat("std population opening_ply:", pop_sd(games$opening_ply), ", std sample opening_ply:",sd(gamesSample$opening_ply))
std population opening_ply: 2.909986 , std sample opening_ply: 3.471615
> cat("mean population avg_rating:", mean(games$avg_rating), ", mean sample avg_rating:",mean(gamesSample$avg_rating))
mean population avg_rating: 1614.242 , mean sample avg_rating: 1619.339
> cat("std population avg_rating:", pop_sd(games$avg_rating), ", std sample avg_rating:",sd(gamesSample$avg_rating))
std population avg_rating: 260.5032 , std sample avg_rating: 261.3713
> cat("mean population turns:", mean(games$turns), ", mean sample turns:",mean(gamesSample$turns))
mean population turns: 61.2727 , mean sample turns: 65.49342
> cat("std population turns:", pop_sd(games$turns), ", std sample turns:",sd(gamesSample$turns))
std population turns: 34.22212 , std sample turns: 35.02733
```

5.2.Confidence Intervals

```
> turns_sample_mean = mean(gamesSample$turns)
> turns_sample_sd = sd(gamesSample$turns)
> turns_sample_se = turns_sample_sd / sqrt(SAMPLE_SIZE)
> t_score = qt(p=1-alpha/2, df=SAMPLE_SIZE-1)
> margin_error <- t_score * turns_sample_se
> lower_bound <- turns_sample_mean - margin_error
> upper_bound <- turns_sample_mean + margin_error
> cat('cl=',1-alpha,lower_bound,'< turns mean < ', upper_bound)
cl= 0.95 59.84208 < turns mean <  71.14476
```

CI with t-test for the turns' mean: with 95% confidence the population mean falls in the confidence interval shown above.

```
> avg_rating_sample_mean = mean(gamesSample$avg_rating)
> avg_rating_sample_sd = sd(gamesSample$avg_rating)
> avg_rating_sample_se = turns_sample_sd / sqrt(SAMPLE_SIZE)
> t_score = qt(p=1-alpha/2, df=SAMPLE_SIZE-1,)
> margin_error <- t_score * avg_rating_sample_se
> lower_bound <- avg_rating_sample_mean - margin_error
> upper_bound <- avg_rating_sample_mean + margin_error
> cat('cl=',1-alpha,lower_bound,'< avg_rating< ', upper_bound)
cl= 0.95 1613.687 < avg_rating<  1624.99
```

CI with t-test for the average ratings' mean: with 95% confidence the population mean falls in the confidence interval shown above.

```
> p_hat = table(gamesSample['rated'])[[1]] / SAMPLE_SIZE
> q_hat = 1 - p_hat
> SE = sqrt(p_hat*q_hat/SAMPLE_SIZE)
> moe = qnorm(1-alpha/2) * SE
> cat('cl=',1-alpha,p_hat - moe,' < proportion of rated games < ',p_hat + moe)
cl= 0.95 0.07893346  < proportion of rated games <  0.1877332
```

CI with z-test for the proportion of rated games: with 95% confidence the population proportion falls in the confidence interval shown above.

```
> df = SAMPLE_SIZE - 1
> chi_right = qchisq(1-alpha/2,df)
> chi_left = qchisq(alpha/2,df)
> cat('cl=',1-alpha,sqrt(df/chi_right) * sd(games$white_rating),' < sigma < ',sqrt(df/chi_left) * sd(games$white_rating))
cl= 0.95 259.8109  < sigma <  326.2878
```

CI with chi-square-test for the standard deviation of the white player rating: with 95% confidence the population proportion falls in the confidence interval shown above.

6.Hypothesis Testing.

```
> rated_false = systematic_sampling(games[games$rated == FALSE,], SAMPLE_SIZE)
> rated_true= systematic_sampling(games[games$rated == TRUE,], SAMPLE_SIZE)
> t.test(rated_false$turns, rated_true$turns)

	Welch Two Sample t-test

data:  rated_false$turns and rated_true$turns
t = -2.5675, df = 290.68, p-value = 0.01075
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -18.528984  -2.448247
sample estimates:
mean of x mean of y
 53.34247  63.83108
```

First test: with alpha=**0.05**
H0: unrated games have the same number of turns with rated games.
Ha: the number of turns is not equal for rated and unrated games.
A t-test was used to evaluate the hypothesis and resulted in a p-value of **0.01075**
Which is significant to **reject** the null hypothesis in favor of the alternative hypothesis.

```
> t.test(gamesSample$turns, mu = 80, alternative = 'less')

	One Sample t-test

data:  gamesSample$turns
t = -6.7365, df = 149, p-value = 1.653e-10
alternative hypothesis: true mean is less than 80
95 percent confidence interval:
    -Inf 65.4822
sample estimates:
mean of x
 60.75333
```

Referencing an online blog of (Alex Crompton)[1] where it is mentioned that the average number of turns is 80.
Second test: with alpha=**0.05**
H0: the mean number of turns **equals 80.**
Ha: the mean number of turns is **less than 80**.
A t-test was used to evaluate the hypothesis and resulted in a highly **significant** p-value so, the null hypothesis was **rejected** in favor of the alternative hypothesis.

```
> mate_sample = systematic_sampling(games[games$victory_status == 'mate',], SAMPLE_SIZE)
> resign_sample = systematic_sampling(games[games$victory_status == "resign",], SAMPLE_SIZE)
> t.test(mate_sample$turns, resign_sample$turns)

        Welch Two Sample t-test

data:  mate_sample$turns and resign_sample$turns
t = 3.4961, df = 290.38, p-value = 0.0005459
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
  5.51145 19.71049
sample estimates:
mean of x mean of y
 66.75862  54.14765
```

Third test: with alpha=**0.05**
H0: the games with the outcome 'mate' and the games with the outcome 'resign' have the **same** number of turns.
Ha: the number of turns is **not equal** for 'mate' and 'resign' games.
A t-test was used to evaluate the hypothesis and resulted in a p-value of **0.005459.** Which is significant to **reject** the null hypothesis in favor of the alternative hypothesis.

```
> white_turns = systematic_sampling(games[games$winner == 'white',], SAMPLE_SIZE)
> black_turns = systematic_sampling(games[games$winner == "black",], SAMPLE_SIZE)
> var.test(white_turns$turns, black_turns$turns)

        F test to compare two variances

data:  white_turns$turns and black_turns$turns
F = 0.77688, num df = 149, denom df = 146, p-value = 0.1261
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.5617114 1.0738948
sample estimates:
ratio of variances
         0.7768817
```

Fourth test: with alpha=**0.05**
H0: the standard deviations of games with white as the winner and games with black as the winner are equal.
Ha: the standard deviations of games with black winner and games with white winner are not equal.
A F-test was used to evaluate the hypothesis and resulted in a p-value of **0.1261.** Which is not significant, so we **fail to reject** the null hypothesis.

```
> white_wins = sum(gamesSample$winner == "white")
> black_wins = sum(gamesSample$winner == "black")
> prop.test(x = white_wins, n = (white_wins+black_wins), p = 0.5)

	1-sample proportions test with continuity correction

data:  white_wins out of (white_wins + black_wins), null probability 0.5
X-squared = 0.35, df = 1, p-value = 0.5541
alternative hypothesis: true p is not equal to 0.5
95 percent confidence interval:
 0.3871699 0.5572783
sample estimates:
        p 
0.4714286
```

Fifth test: with alpha=**0.05**
H0: the proportion of white wining is **equal to 0.5.**
Ha: the proportion of white winning is **not equal to 0.5.**
A z-test was used to evaluate the hypothesis and resulted in a p-value of **0.5541.** Which is not significant, so we **fail to reject** the null hypothesis.

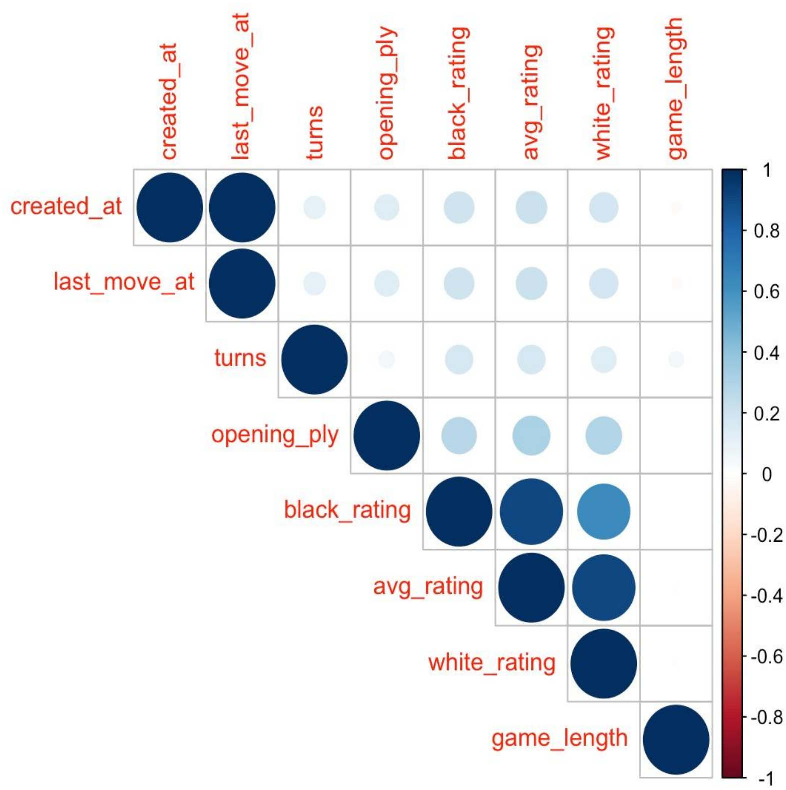
7.Linear Regression.

First a correlation plot was curated to have a deeper insight about linear regression model candidates.

It was realized that not many variables were correlated leaving us with two candidates only that needed further analysis and visualizations:

1-game_length and turns.

2-black and white ratings.



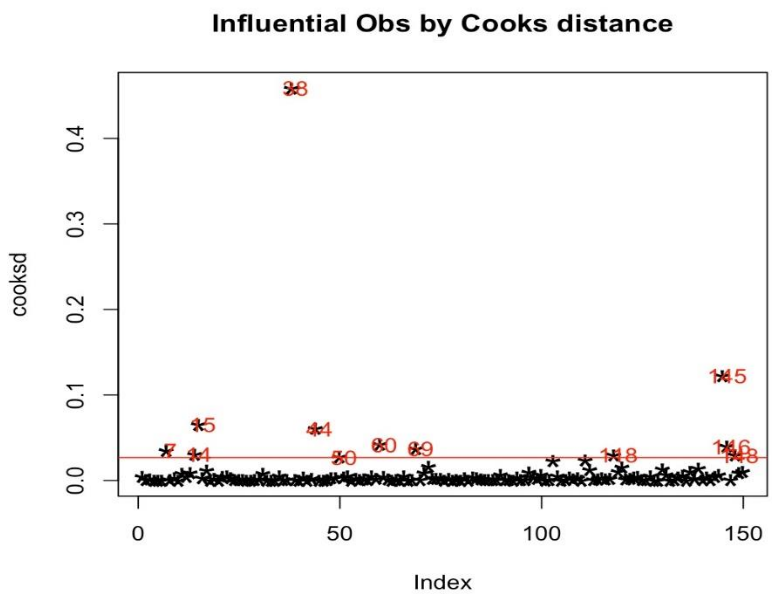
7.1.First model.

The correlation value is:

```
> cor(gamesSample$white_rating,gamesSample$black_rating)
[1] 0.6186693
```

```
model = lm(gamesSample$white_rating~gamesSample$black_rating)
cooksd <- cooks.distance(model)
plot(cooksd, pch="*", cex=2, main="Influential Obs by Cooks distance")  # plot cook's distance
abline(h = 4/SAMPLE_SIZE, col="red")  # add cutoff line
text(x=1:length(cooksd)+1, y=cooksd, labels=ifelse(cooksd>4/SAMPLE_SIZE, names(cooksd),""), col="red")
influential <- as.numeric(names(cooksd)[(cooksd > (4/SAMPLE_SIZE))])
new_gamesSample <- gamesSample[-influential, ]
```

the data was fitted to the model and the influential points affecting the correlation magnitude of the model were removed using the cooks.distance() methods a method that is used commonly to detect outliers and influential points when it comes to least squares regression.



The new correlation value after influential points removal:

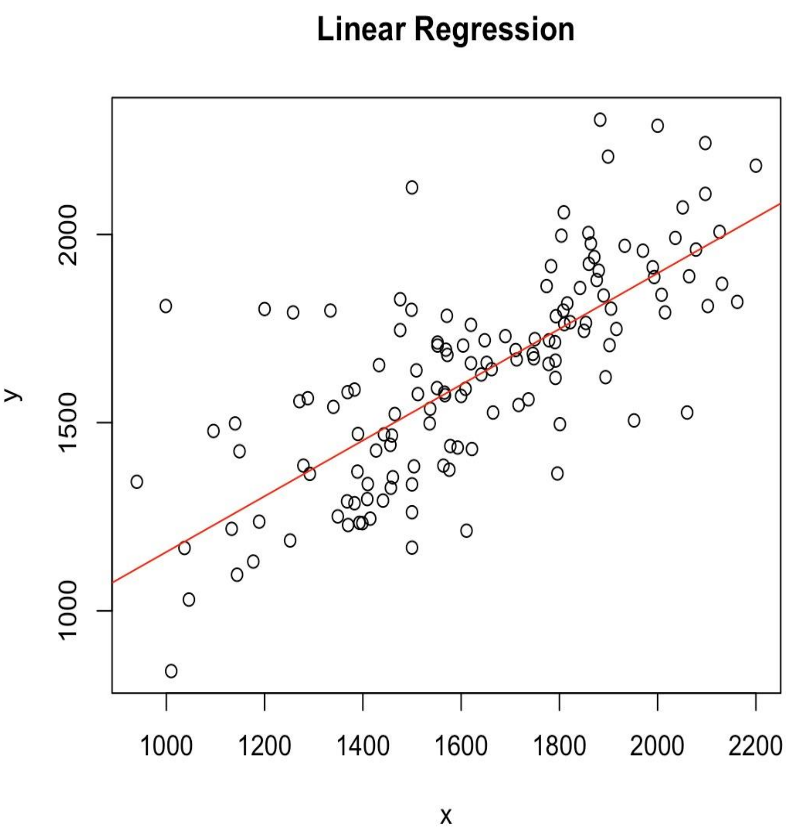
```
> cor(new_gamesSample$white_rating,new_gamesSample$black_rating)
[1] 0.7204316
```

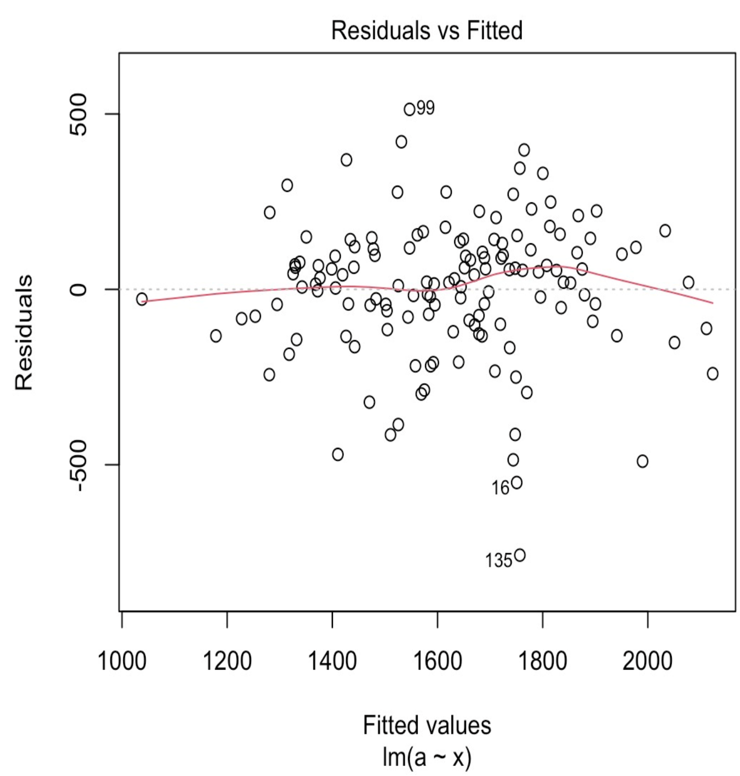
Updated model:

```
> new_model = lm(new_gamesSample$white_rating~new_gamesSample$black_rating)
> a <- new_gamesSample$white_rating
> x <- new_gamesSample$black_rating
> new_model = lm(a~x)
> predict(new_model, data.frame(x=c(1200, 2000, 1500)))
       1        2        3
1304.579 1897.312 1526.854
```

A few predictions were made using the model as seen above.

Visualizations:





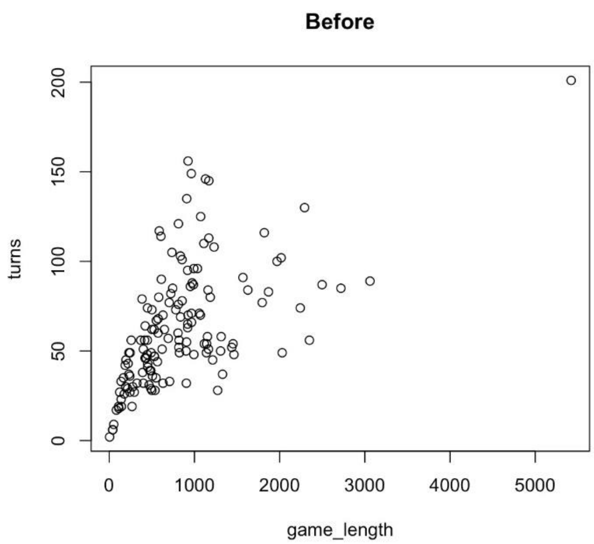
7.2.Second model:

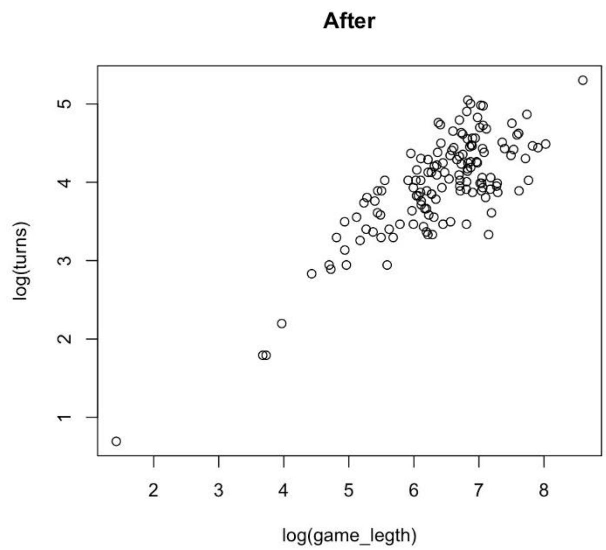
The original correlation value:

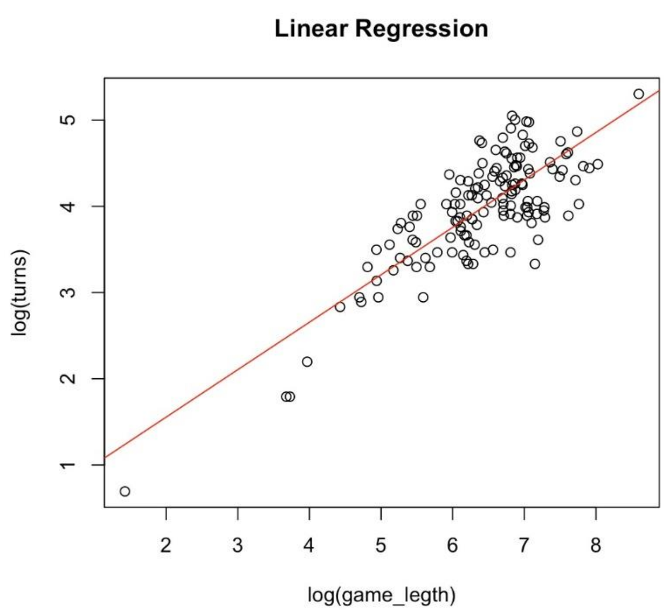
```
> cor(gamesSample$turns, gamesSample$game_length)
[1] 0.4952019
```

It was noticed that making a log transformation increased linearity.
Correlation value after transformation:

```
> cor(log(gamesSample$turns),log(gamesSample$game_length))
[1] 0.8555455
```







Updated model:
A few predictions were made using the model as seen below.

```
> a <- log(new_gamesSample$turns)
> x <- log(new_gamesSample$game_length)
> model2 = lm(a~x)
> exp(predict(model2, data.frame(x=c(log(1500), log(3000), log(5000)))))
        1         2         3 
 88.11238 129.02394 170.89776 
```

8.Chi Square Tests.

8.1.Goodness of fit

According to an article on 'chessbase'[2] the percentages of game outcomes are the following:
White winning 28.85%, **Black** winning 18.00%, **Draw** 53.15%. to test those percentages (expected values) a goodness of fit test was preformed.

```
> # Godness-of-fit
> #observed
> table(gamesSample$winner)

black  draw white 
   74    10    66 
> b = table(gamesSample$winner)[[1]]
> d = table(gamesSample$winner)[[2]]
> w = table(gamesSample$winner)[[3]]
> o = c(w,b,d)
> #expected
> p = c(0.2885,0.18,0.5315)
```

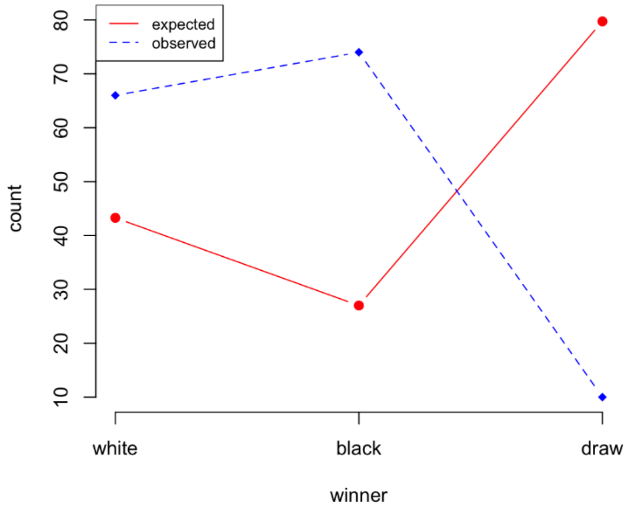
```
> result = chisq.test(o,p=p)
> result

	Chi-squared test for given probabilities

data:  o
X-squared = 154.73, df = 2, p-value < 2.2e-16

> result$expected
[1] 43.275 27.000 79.725
> result$observed
[1] 66 74 10
```

Ho: the percentages in our data **match** the expected values.
Ha: the percentages **don't match** the expected values.
The observed values were collected and both the expected and observed were plugged into chisq.test() method. Which resulted in a highly **significant** p-value, so the null hypothesis was **rejected** in favor of the alternative hypothesis.
(It might be due to the fact that the percentages found on the website belong to more professional chess games, and so more draws.)



The difference between expected and observed is as plotted.

8.2.Independence test

```
> # independence test
> table(gamesSample$rated, gamesSample$victory_status)

        outoftime resign mate draw
  FALSE         3     17    8    2
  TRUE         11     77   24    8
> chisq.test(table(gamesSample$rated, gamesSample$victory_status))

        Pearson's Chi-squared test

data:  table(gamesSample$rated, gamesSample$victory_status)
X-squared = 0.73328, df = 3, p-value = 0.8653
```

Ho: The categorical variables rated and victory status are **independent.**

Ha: The categorical variables are **dependent**.

A contingency table of both variables was formed and plugged into chisq.test() method. It resulted in a **non-significant** p-value, so we **fail to reject** the null hypothesis.

8.3. Homogeneity test

```
> sample1 = systematic_sampling(games[games$rated == FALSE, ], SAMPLE_SIZE)
> sample2 = systematic_sampling(games[games$rated == TRUE, ], SAMPLE_SIZE)
> sample3 = rbind(sample1, sample2)
> table(sample3$rated, sample3$winner)

        black draw white
  FALSE    58   12    76
  TRUE     67    5    76
> chisq.test(table(sample3$rated, sample3$winner))

        Pearson's Chi-squared test

data:  table(sample3$rated, sample3$winner)
X-squared = 3.5169, df = 2, p-value = 0.1723
```

Two samples were taken from the dataset where sample1 contained not-rated games and sample two contained rated games. The samples were then merged and a contingency table was formed between the rated and winner variables of the merged sample, and plugged into the chisq.test() method.

H0: The rated and not-rated games are **homogeneous** when it comes to the **categories** of the **winner** variable.

Ha: There is a **difference** when it comes to the **categories** of the variable **winner** between rated and not rated games.

The resulting p-value was **not significant enough** to reject the null hypothesis. We **fail to reject** the null hypothesis.

9.Anova

9.1. One way Anova

```
> # one-way
> gamesSample$victory_status = ordered(gamesSample$victory_status, levels = c("outoftime", "resign", "mate", "draw"))
> anova_t1 = aov(turns~victory_status, data=gamesSample)
> summary(anova_t1)
                Df Sum Sq Mean Sq F value  Pr(>F)
victory_status   3  17498    5833   5.163 0.00202 **
Residuals      146 164942    1130
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Ho: The turns’ mean is the same for all victory-status categories.

Ha: The turns' mean is the different for at least one victory-status category.

The test resulted in a **highly significant** p-value, so the null hypothesis was **rejected** in favor of the alternative hypothesis.

To see which category/categories were different the Scheffé test was applied.

```
> #perform Scheffe's test
> ScheffeTest(anova_t1)

  Posthoc multiple comparisons of means: Scheffe Test
    95% family-wise confidence level

$victory_status
                       diff     lwr.ci    upr.ci   pval
resign-outoftime -29.784195 -57.018107 -2.550282 0.0256 *
mate-outoftime   -12.446429 -42.908962 18.016104 0.7211
draw-outoftime    -7.471429 -46.832572 31.889715 0.9621
mate-resign       17.337766  -2.119081 36.794613 0.1005
draw-resign       22.312766  -9.308470 53.934002 0.2677
draw-mate          4.975000 -29.466001 39.416001 0.9827

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

9.2.Two way Anova

```
> #two-way
> anova_t2 = aov(turns~winner*rated, data=gamesSample)
> summary(anova_t2)
              Df Sum Sq Mean Sq F value Pr(>F)
winner         2   2205  1102.7   0.895  0.411
rated          1   1586  1585.6   1.287  0.258
winner:rated   2   1291   645.4   0.524  0.593
Residuals    144 177358  1231.7
```

Three hypotheses:

H01: There is **no difference** between the means of turns for winner categories.
Ha1: There is **a difference** between the means of turns for winner categories.

H02: There is **no difference** between the means of turns for rated categories.
Ha2: There is **a difference** between the means of turns for rated categories.

H03: There is **no interaction affect** between the **winner** and the **rated categories on** the **turns' means.**
Ha3: There is **an interaction affect** between the **winner** and the **rated categories on** the **turns' means.**

All the resulting p-values were **not significant** enough to reject the null hypotheses.

10.Non-parametric tests.

10.1. Sign test.

Opening_ply variable was chosen due to its skewed distribution.

```
> s2 = systematic_sampling(games, 20)
> SIGN.test(s2$opening_ply ,md=8, alternative = 'less')

        One-sample Sign-Test

data:  s2$opening_ply
s = 1, p-value = 3.815e-05
alternative hypothesis: true median is less than 8
95 percent confidence interval:
 -Inf    4
sample estimates:
median of x
          3

Achieved and Interpolated Confidence Intervals:

                  Conf.Level L.E.pt U.E.pt
Lower Achieved CI     0.9423   -Inf      4
Interpolated CI       0.9500   -Inf      4
Upper Achieved CI     0.9793   -Inf      4
```

Based on chess.com forums[3] the average opening-ply is 8.

H0: The **median** of opening_ply **equals 8**.
Ha: The **median** of opening_ply is **less than 8**.

The sign test resulted in a **highly significant** p-value, so the null hypothesis was **rejected** in favor of the alternative hypothesis.

10.2.Wilcox rank sum test.

```
> wilcox.test(s2$opening_ply[s2$rated == TRUE],s2$opening_ply[s2$rated == FALSE])

	Wilcoxon rank sum test with continuity correction

data:  s2$opening_ply[s2$rated == TRUE] and s2$opening_ply[s2$rated == FALSE]
W = 54, p-value = 0.3224
alternative hypothesis: true location shift is not equal to 0
```

Ho: The **median** of opening_ply for **rated and not rated** games is the **same**.
Ha: The **median** of opening_ply for **rated and not rated** games is **different**.

The Wilcoxon rank sum test resulted in a **non-significant** p-value, so we **fail to reject** the null hypothesis.

10.3.Spearman correlation.

```
> cor.test(s2$opening_ply, s2$game_length , method = "spearman", exact = FALSE)

	Spearman's rank correlation rho

data:  s2$opening_ply and s2$game_length
S = 1210, p-value = 0.7053
alternative hypothesis: true rho is not equal to 0
sample estimates:
       rho
0.09020621
```

The cor.test() was applied to see if there was correlation between opening_ply and game_length.

Ho: There is **no correlation** between opening_ply and game_length.
Ha: There is **a correlation** between opening_ply and game_length.

The test resulted in a **non-significant** p-value, so we fail to reject the null hypothesis.

10.4.Runs test

```
> runs_test_sample = games$opening_ply[1:20] %% 2
> runs_test_sample
 [1] 0 1 0 1 1 0 0 1 0 0 1 1 1 0 1 0 0 1 0 1
> runs.test(runs_test_sample)

	Runs Test

data:  runs_test_sample
statistic = 1.3784, runs = 14, n1 = 10, n2 = 10, n = 20, p-value = 0.1681
alternative hypothesis: nonrandomness
```

The method run.test() was applied on the variable opening_ply to check for randomness.

H0: The sequence is **random**.
Ha: The sequence is **not random**.

The p-value resulting from the test is **not significant** enough to reject the null hypothesis, so we **fail to reject**.

References

[1] https://www.alexcrompton.com/blog/time-thoughts-chess#:~:text=will%20be%20straightforward.-,Average%20moves%20per%20game,chess%20skill%20%2D%20on%20the%20table.

[2] https://en.chessbase.com/post/has-the-number-of-draws-in-chess-increased

[3] https://www.chess.com/forum/view/chess-openings/how-many-moves-does-an-opening-consist-of