Part 1

For this project I will be analyzing a chess dataset from kaggle published in 2017 more specifically I will be analyzing based on data in columns **white_rating, turns and increment_code** to answer some interesting questions**.**

Overall Dataset:

- There are **20058** rows and **16** columns

```
> data = read.csv("C:\\Users\\anish\\Desktop\\games.csv")
> dim(data)
[1] 20058    16
```

- There are no empty entries (na) in the entire dataset

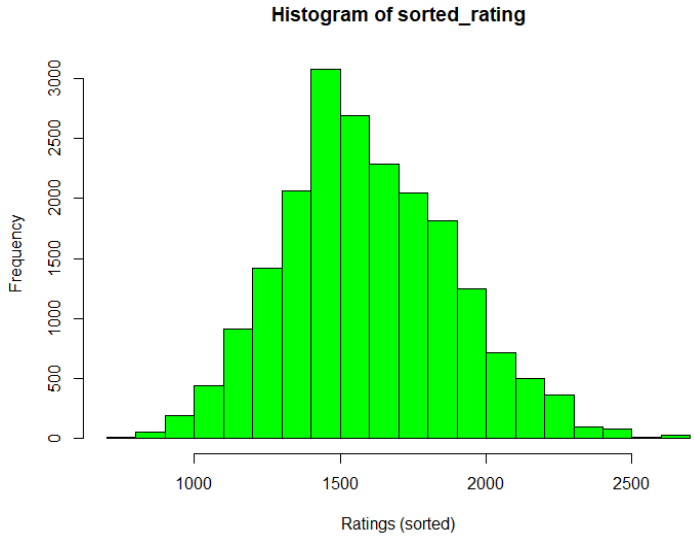
```
> data = read.csv("C:\\Users\\anish\\Desktop\\games.csv")
> check = sapply(data, is.na)
> length(which(check=="TRUE"))>0
[1] FALSE
```

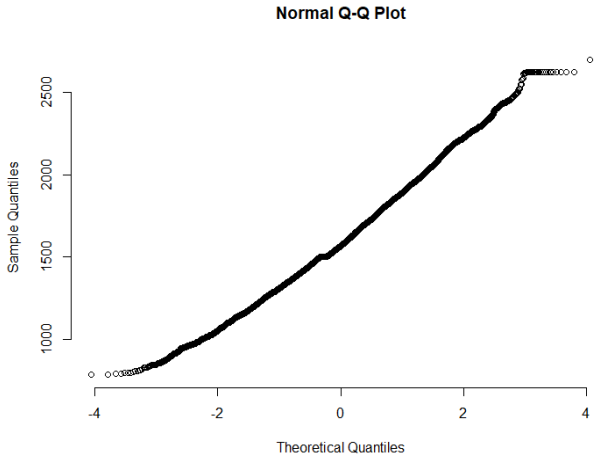
Below are the descriptions, data type, histograms and summary statistics for each of the three columns of interest to get an idea of the data.

This dataset was made by the lichess api, as an amateur chess player it is really interesting to apply data science to this area.

Columns:

- **White_Rating**: Contains the online chess ratings (integers) of the players playing as white for the associated game.
    - As expected it follows a normal distribution shown below





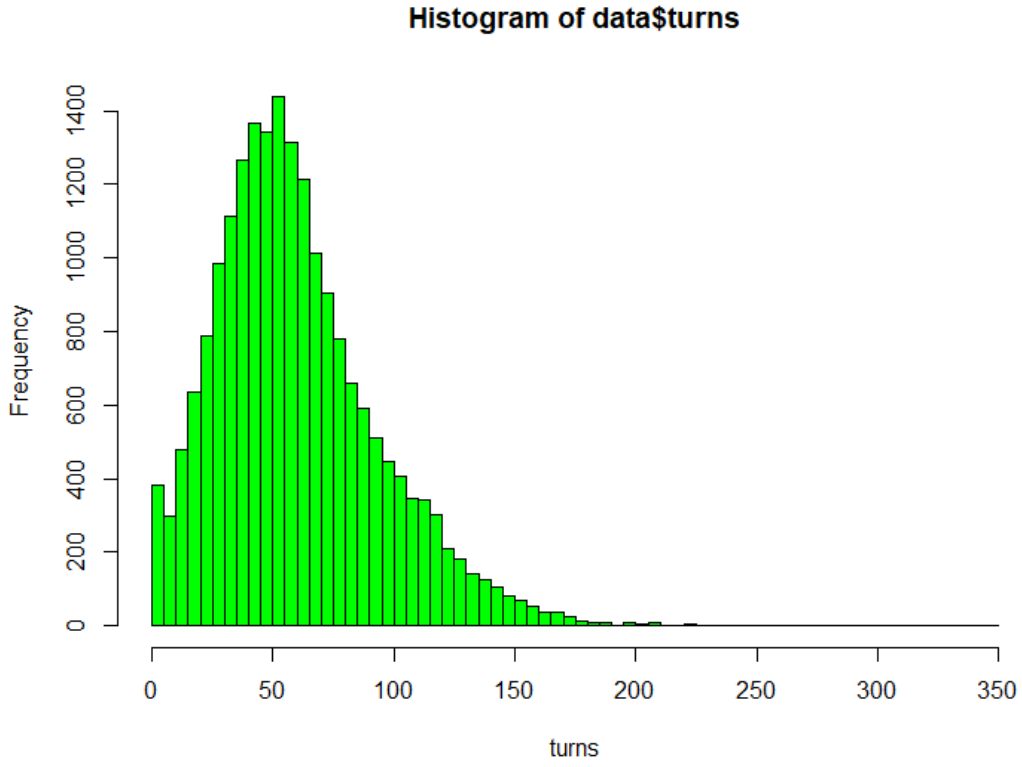
Code (showing normal):

```
data = read.csv("C:\\Users\\anish\\Desktop\\games.csv")
sorted_rating = sort(data$white_rating)
hist(sorted_rating, breaks=20, col="green",xlab="Ratings (sorted)")
qqnorm(sorted_rating, pch = 1, frame = FALSE)
```

Summary stats for white_rating column:

```
> summary(sorted_rating)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    784    1398    1567    1597    1793    2700
```

- **turns:** Contains integers representing the amount of moves for the associated game.



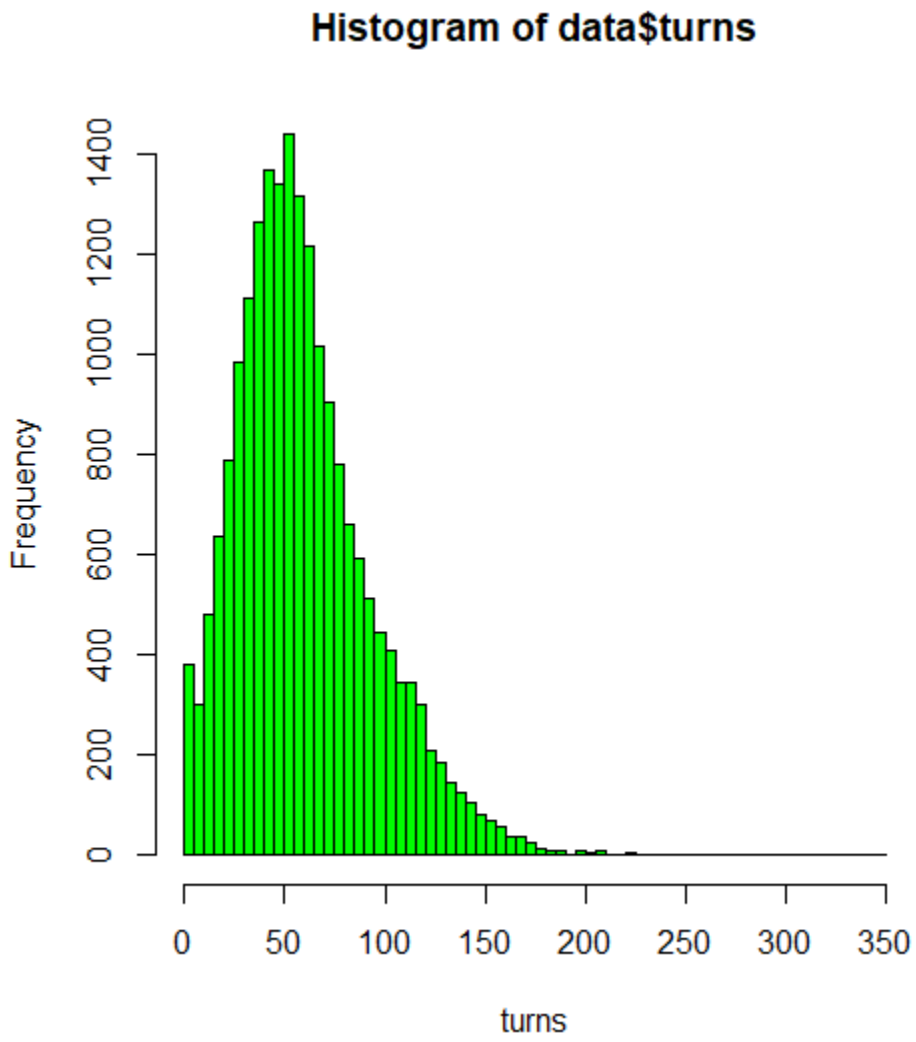
Code:

```
data = read.csv("C:\\Users\\anish\\Desktop\\games.csv")
sorted_rating = sort(data$white_rating)
hist(data$turns, breaks=60, col="green",xlab="turns")
```

Summary statistics for turns:

```
> summary(data$turns)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1.00   37.00   55.00   60.47   79.00  349.00
```

- **Increment_code:** Contains the time format of the chess game (in the form a or a:b, where a is a number describing minutes and b is a number describing seconds incremented per move.
- Overall the data type is shown as a character however.
- .Histogram



- Summary stats for increment code:

```
> summary(data$increment_code)
   Length     Class      Mode
    20058 character character
```

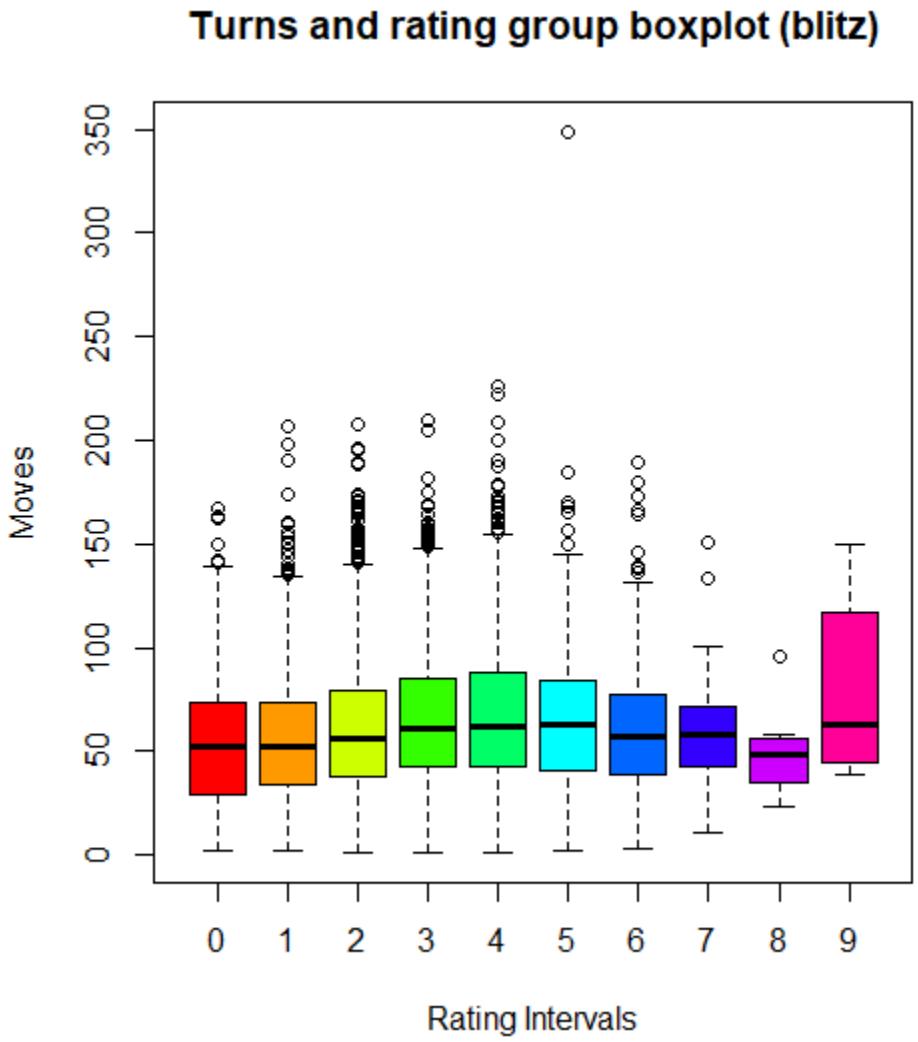
Part 2: Questions and analysis

Question 1: Do different rating groups have a similar number of moves played in blitz (10 minute) games?

- **Interest:**
  - I don't believe there is any theory on how the number of games if related to rating, the results suggested by the data could yield some interesting speculation.
- **Method:** Anova
  - **Reasoning:** The means for the number of moves will be compared across a variety (more than 2) of rating intervals (across groups).
  - **Assumptions:**
    - Each rating interval's number of moves are normally distributed.
    - Each rating interval's sample size is sufficient and approximately equal.
    - Each rating interval is independent.
    - White's rating is the same as black's rating
    - We will follow the ELO rating system to make a new column of rating intervals as shown below (marked 0-10, with 0 being novice)

Rating range	Category
2700+	No formal title, but sometimes informally called "super grandmasters"[5]
2500–2700	most Grandmasters (GM)
2400–2500	most International Masters (IM) and some Grandmasters (GM)
2300–2400	most FIDE Masters (FM) and some International Masters (IM)
2200–2300	FIDE Candidate Masters (CM), most national masters (NM)
2000–2200	Candidate masters (CM)
1800–2000	Class A, category 1
1600–1800	Class B, category 2
1400–1600	Class C, category 3
1200–1400	Class D, category 4
below 1200	novices

- **Results:**
  - After using lapply to assign each white_rating an interval number (0-10], we can see the boxplot of the number of turns for each interval group below.



From the boxplot graph above the means look to be relatively similar. Therefore I expect the anova result comparing turns (moves) and rating intervals to have a p value >= 0.05 for blitz games.

Code:

```
# ratings to be used in calculating intervals
tenratings <- data[ which(data$increment_code=='10+0'),'white_rating']
# turns (moves) to be used in anova
tenturns <- data[ which(data$increment_code=='10+0'),'turns']
# change the ratings to a discrete interval form
i = lapply(tenratings,intervals)
groups = unlist(i)
#results
boxplot(tenturns~groups,col=rainbow(length(unique(groups))),xlab="Rating Intervals",ylab="Moves",main="Turns and rating group boxplot (blitz)")
```

However, when I actually do run anova with the intervals and moves for blitz games. I get a super tiny p value shown below.

```
> anova(lm(tenturns~groups))
Analysis of Variance Table

Response: tenturns
            Df  Sum Sq Mean Sq F value    Pr(>F)
groups       1  124195  124195  109.36 < 2.2e-16 ***
Residuals 7719 8766426    1136
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
>
```

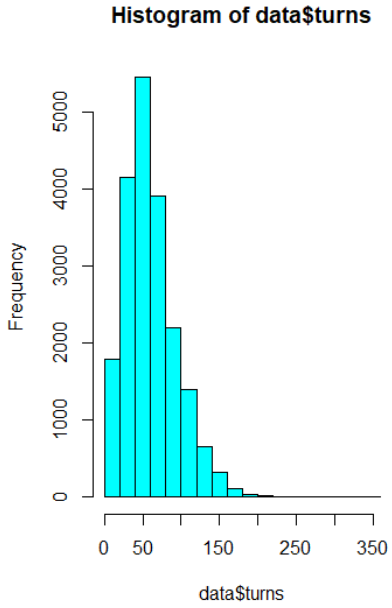
- The p value is basically 0 which is < 0.05. Therefore we reject the null hypothesis, that the number of moves across different rating levels is the same.

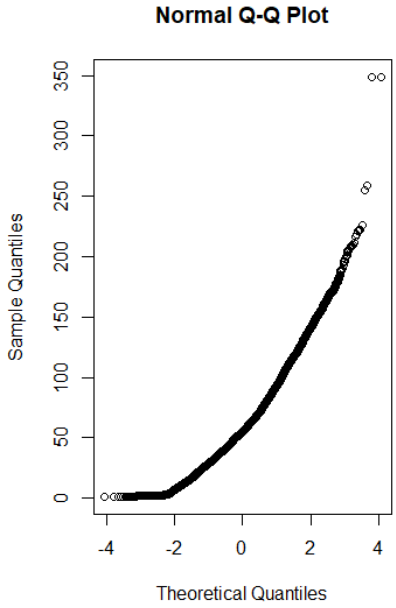
- **Interpretation:**
  - A possible explanation for this is that at different rating levels the average number of moves is different due to the understanding of the game.
  - However , this is very surprising to me as the boxplot had shown the mean number of moves to be quite similar.
  - It is likely that I made an incorrect assumption, making anova unreliable but I am pretty sure the boxplot is correct in the R code.

Question 2: As ratings increase, does the number of moves per game increase as well?

- **Interest:**
  - I am not sure ratings are related to the number of moves per game, so it will be interesting to see if there is any relation at all.
- **Method:** Correlation
  - **Reasoning:** By using correlation statistics we can determine if there is a positive/negative or no relationship between rating and the number of moves.

  - **Assumptions:**
    - Ratings are shown on the second page to be relatively normal.
    - Number of moves per game is relatively normal as well shown below





Code:

```
#Assumptions
hist(data$turns,col='cyan')
qqnorm(data$turns)
```

- We also assume that white's rating is approximately the same as black's rating (so the game rating is white's rating).

- **Results:**

```
> cor.test(data$white_rating,data$turns)

        Pearson's product-moment correlation

data:  data$white_rating and data$turns
t = 18.532, df = 20056, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.1161220 0.1433344
sample estimates:
      cor
0.1297526
```

- As shown above there is approximately a 0.13 correlation coefficient with white's rating and the number of turns.

- **Interpretation:**
  - A correlation coefficient value of around 0.13 can be interpreted as a very weak positive correlation between white's rating and the number of moves played.
  - Therefore there is a little to no connection between rating and the number of moves.

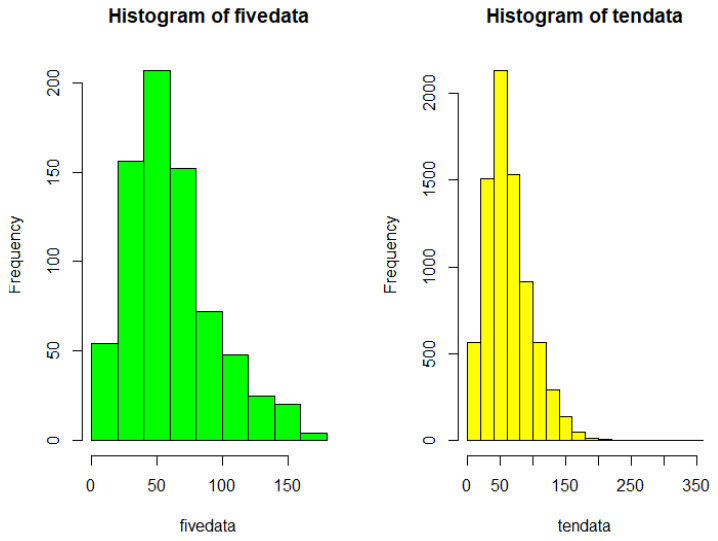
Question 3: Do 10 minute games and 5+5 minute games have a similar number of moves?

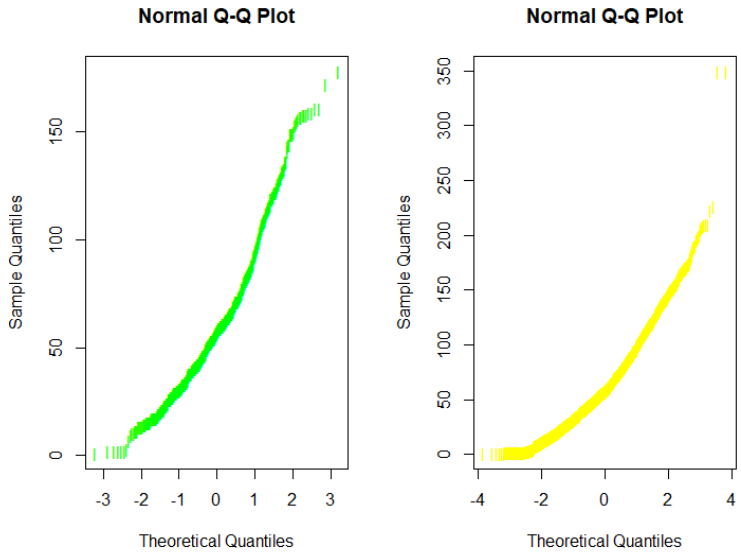
* 5+5 = 5 minutes with 5 extra seconds (for that player) each move.

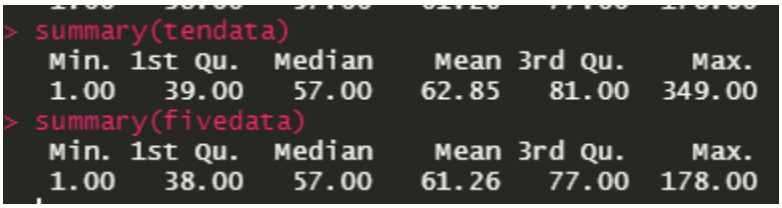
- **Interest:**
  - I would guess that a lower time control would mean less moves as the games would end faster due to poorer play, on average. However I am not sure how far the time controls should be compared for this to happen, which makes this question interesting.
- **Method:** two-sided t-test
  - **Data Processing:**
    -
```
data = read.csv("C:\\Users\\anish\\Desktop\\games.csv")
tendata <- data[ which(data$increment_code=='10+0'),'turns']
fivedata <- data[ which(data$increment_code=='5+5'),'turns']
```
    - Select the **turn** column where increments are 10+0 or 5+5 respectively.
    - Save them as fivedata and tendata
  - **Reasoning:**
    - Comparing two means (number of turns) to see if there is a significant difference (10 / 5+5 minute games) is best suited for t-tests.
    - In this case it should be two-sided as we are trying to determine whether 10 or 5+5 minute games have a significantly different number of tests.
  - **Assumptions:**
    - **T-tests**
      - **N is sufficiently large = 20058**
      - There are significantly more 10 minute games (7721) than 5+5 minute games (738). This is not ideal and could cause a lot of errors in the results. We are assuming that (738) 5+5 games represent as much as (7721) 10 minute games which is a

big flaw. For simplicity's sake we will assume extrapolating like this is ok.

- **Same variance:**
  -
    ```
    > sd(fivedata)
    [1] 32.99613
    > sd(tendata)
    [1] 33.93574
    ```
  - Standard deviations are approximately equal, as n is the same variance is also approximately equal.
- **Distributions are relatively normal when plotted.**





- **Results:**
  - 

    - We can expect H0 to be accepted as the means are very similar between 10 and 5+5 minute games.

```
> t.test(fivedata,tendata,var.equal = T)

        Two Sample t-test

data:  fivedata and tendata
t = -1.2213, df = 8457, p-value = 0.222
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -4.1501188  0.9638292
sample estimates:
mean of x mean of y
 61.25610  62.84924
```

  - As shown above the p-value is significantly greater than the threshold value (0.222>>0.05) so we can accept H0, the hypothesis that the mean number of moves (turns) in 10 minute games and 5+5 minute games are equal.

- **Interpretation:**
  - This shows how the number of moves in 10 minute and 5+5 minute games are significantly similar. I hypothesized that 5+5 minute games would have less moves because there would be more mistakes with less time but that does not seem to be the case.
  - I think the reason that H0 was accepted was because 5 minutes and 5 extra seconds per move is close enough to 10 minutes with the 5 second increments. It would be

interesting to do it again with more extreme time controls like one minute vs thirty minutes.

- This makes me wonder what if there is an exact cutoff time control where the number of moves would be reduced due to playing worse with constrained time.