

Magic of chess

Few statistical answers

Introduction

Chess is a two-player strategy board game played on a checkered board with 64 squares arranged in an 8×8 grid. The game is played by millions of people worldwide. Play involves no hidden information. Each player begins with 16 pieces: one king, one queen, two rooks, two knights, two bishops, and eight pawns. Each piece type moves differently, with the most powerful being the queen and the least powerful the pawn. The objective is to checkmate the opponent's king by placing it under an inescapable threat of capture. To this end, a player's pieces are used to attack and capture the opponent's pieces, while supporting each other. During the game, play typically involves exchanging pieces for the opponent's similar pieces, and finding and engineering opportunities to trade advantageously or to get a better position. In addition to checkmate, a player wins the game if the opponent resigns, or (in a timed game) runs out of time. There are also several ways that a game can end in a draw. Online chess has opened amateur and professional competition to a wide and varied group of players.



Since the second half of the 20th century, chess engines have been programmed to play with increasing success, to the point where the strongest programs play at a higher level than the best human players. Since the 1990s, computer analysis has contributed significantly to chess theory, particularly in the endgame. The IBM computer Deep Blue was the first machine to overcome a reigning World Chess Champion in a match when it defeated Garry Kasparov in 1997.

The game structure and nature of chess are related to several branches of mathematics. Many combinatorial and topological problems connected to chess have been known for hundreds of years. Interests in chess games have been grown up by few years because of need in testing the modern technologies based on artificial intelligence. In particular chess is a well-known pattern game and for that it offers a theoretical illimited research field for statistical studies and machine learning.

There are many variants of chess that utilize different rules, pieces, or boards. One of these, Fischer Random Chess, has gained widespread popularity and official FIDE recognition.

Dataset Description

This is a set of over 20,000 games collected from a selection of users on the site Lichess.org. This set contains the following attributes:

Names	Type	Levels
ID	Categorical	//
Rated	Dicotomic	True, False
Turns	Discrete	Int ≥ 0
Created at	Continuous	Float ≥ 0
Last move at	Continuous	Float ≥ 0
White rating	Discrete	Int ≥ 0
Black rating	Discrete	Int ≥ 0
White id	Categorical	//
Black id	Categorical	//
Winner	Categorical	White, Black, Draw
Increment code	Categorical	Point: (Int ≥ 0 , Int ≥ 0)
Victory status	Categorical	Mate, OFT, Draw, Resign
Moves	Categorical	All possible move successions
Opening eco	Categorical	Specified in https://www.365chess.com/eco.php
Opening Name	Categorical	Specified in https://www.365chess.com/eco.php
Opening ply	Discrete	Int ≥ 0

Table 1: Description of character by the type and levels.

For each of these separate games from Lichess. Data were collected using the Lichess API, which enables collection of any given users game history. For the statistical analysis I used RStudio (based on R version 3.6.1) in Windows environment.

Follows the description of these attributes (table 2).

Names	Description
ID	Identifier of the game (primary key)
Rated	Boolean variable: indicate if the game is or not a competition in which, in general, one player lose rating and the other one gain some of it
Turns	Number of total moves in game
Created at	Non specified character by the author
Last move at	Non specified character by the author
White rating	Kind of rank of a player, his capability to win against an opponent. In this case referred to the white player
Black rating	As above, referred to the black player
White id	Identifier of the user plays with white pieces
Black id	Identifier of the user plays with black pieces
Winner	One of the 3 possible outcomes: white wins, black wins or draw
Increment code	Control time: the first number indicates the nominal amount of time that every player has (in minutes), while the second one indicates a quantity of time that every player gains after a move (in seconds)
Victory status	One of the 4 possible ends: one player resign, deliver a checkmate to the opponent, runs out of time or the two players tie
Moves	Series of moves the game is made up of
Opening eco	Identifier of a chess opening. A chess opening is a series of standard moves every player can follow at the start of the game and that is known for its correctness
Opening Name	Name of the respective opening
Opening ply	Number of moves of the respective opening

Table 2: Description of characters in their specificity.

In this work have been used all these attributes except: “ID”, “created at” and “last move at”, mainly because of the lack of informations. Moreover, the interval of time for the data connection it’s been supposed to be few days considering the recurrence of the same players among the dataset and the typical affluence of players into Lichess.org site.

A clarification is needed: what I intend for population is a collective of games disputed between players in their exactly rating classes, played in a reasonable number of turns. All the work will revolve around this assumption and for this scope a minimum of data cleaning must be made.

First, an exploratory analysis was carried out.

Explorative analysis

- Frequency distribution for numerical variables:

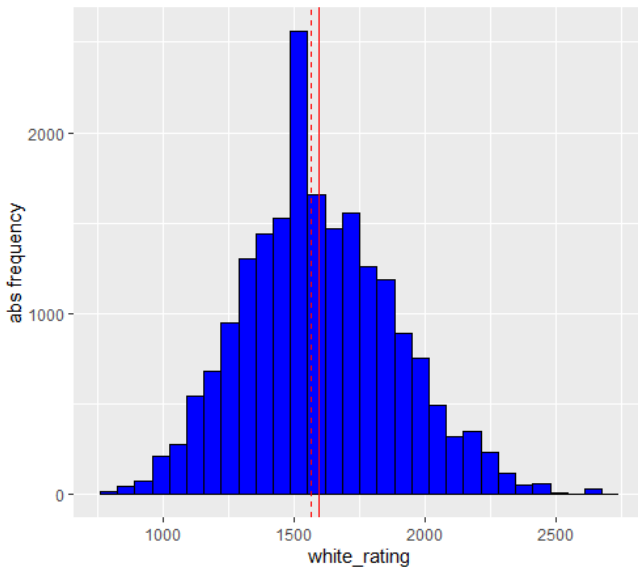


Figure 1: Distribution of absolute frequency of white rating.

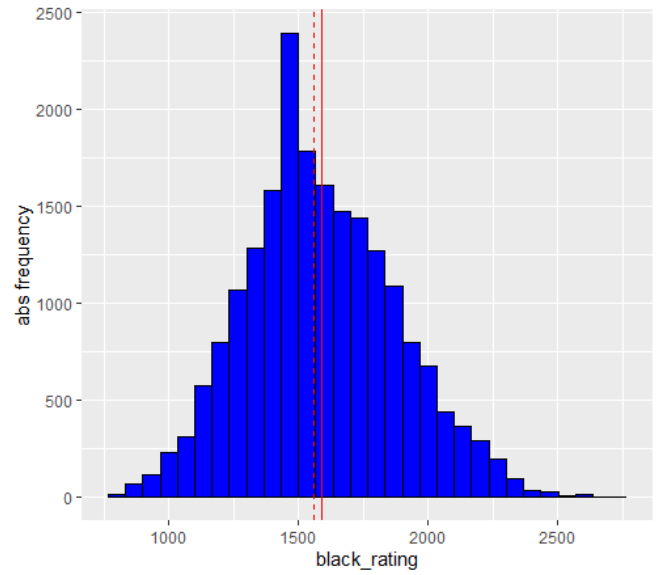


Figure 2: Distribution of absolute frequency of black rating.

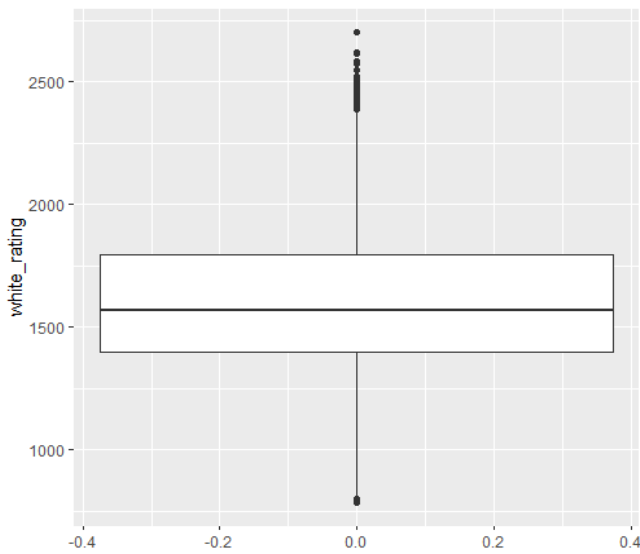


Figure 3: Boxplot of white rating distribution.

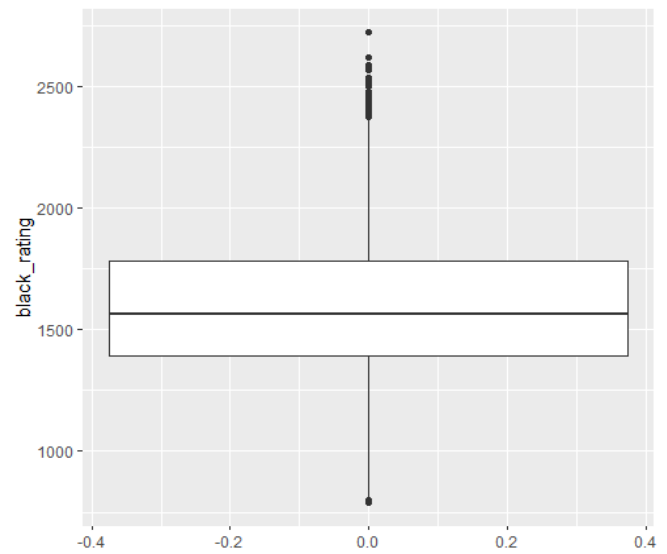


Figure 4: Boxplot of black rating distribution.

First thing I've noticed into these histogram (fig1, fig2) referred to white and black rating is the distribution's anomaly in correspondence of the value 1500, probably caused by the artificial distribution of games played by new users: when a new player registers into Lichess, the site assigns him approximately the average rating of Lichess players, in this case 1500.

The second perceptible element is the slight difference between mean and median of the distributions.

Defining these statistical index: $\Delta M = \text{mean} - \text{median}$, $M_r = \frac{\Delta M}{\text{mean}}$,

$$\text{obtain: } \textit{White} \rightarrow \begin{cases} \Delta M = 1597 - 1567 = 30 \\ M_r = +1.88\% \end{cases} \quad \textit{Black} \rightarrow \begin{cases} \Delta M = 1589 - 1562 = 27 \\ M_r = +1.70\% \end{cases}$$

The result is a slight asymmetry of the distribution to the right, same behavior that the boxplots (fig3, fig4) highlighted.

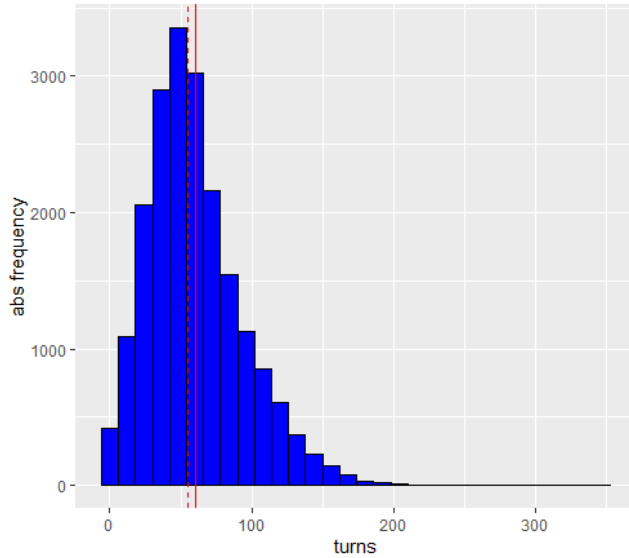


Figure 5: Distribution of absolute frequency of game turns.

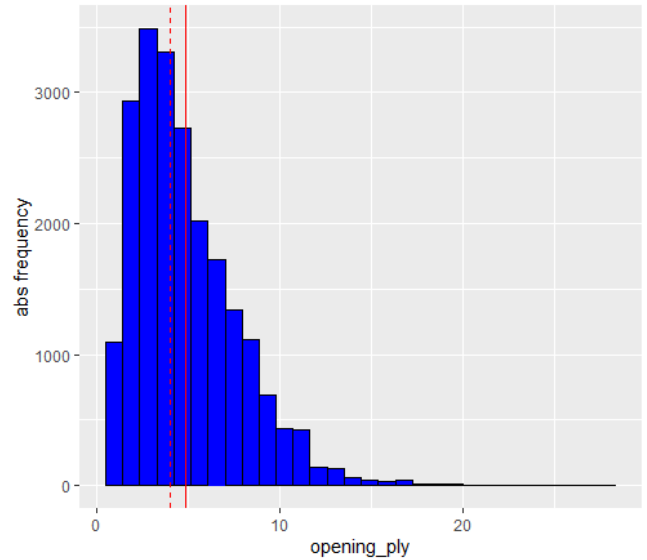


Figure 6: Distribution of absolute frequency of opening ply.

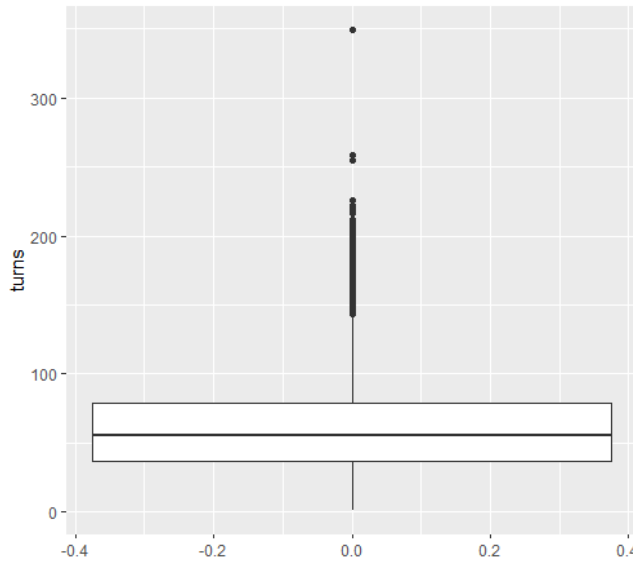


Figure 7: Boxplot of game turns distribution.

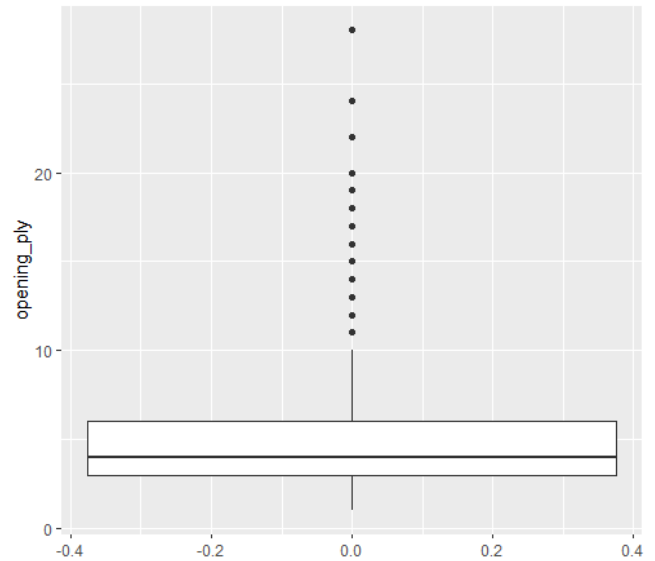


Figure 8: Boxplot of opening ply distribution.

Regarding the attributes 'turns' and 'opening ply' the situation is very different. Being distributions close to the zero, is very difficult to imagine the normal hypothesis could fit them. Moreover there is a more accentuated difference between mean and median:

$$\textit{Turns} \rightarrow \begin{cases} \Delta M = 66 - 55 = 5 \\ M_r = +8.33\% \end{cases} \quad \textit{Opening ply} \rightarrow \begin{cases} \Delta M = 4.8 - 4 = 0.8 \\ M_r = +17\% \end{cases}$$

- Frequency distribution for categorical variables:

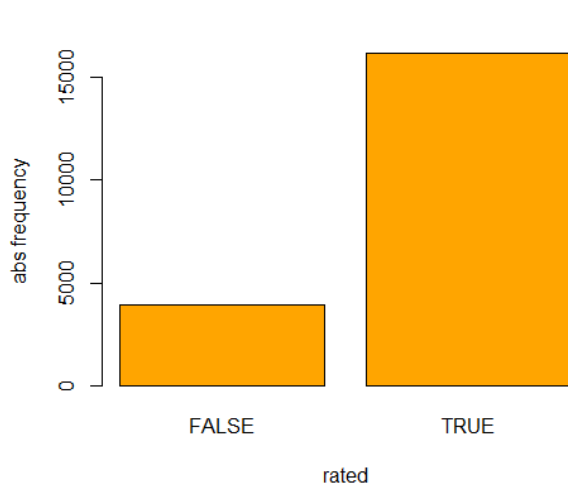


Figure 9: Boxplot of game turns distribution.

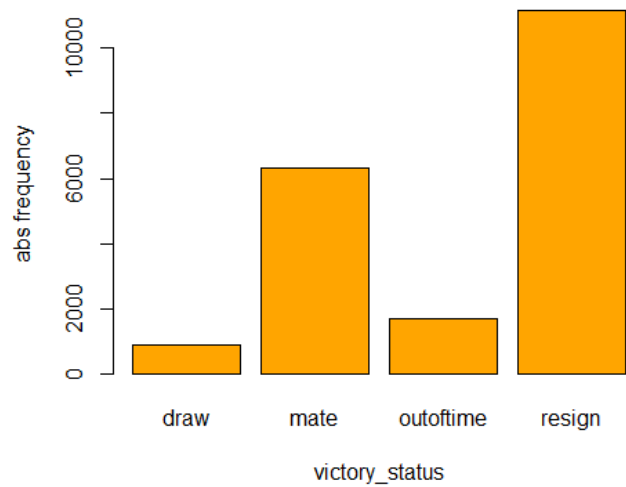


Figure 10: Boxplot of opening ply distribution.

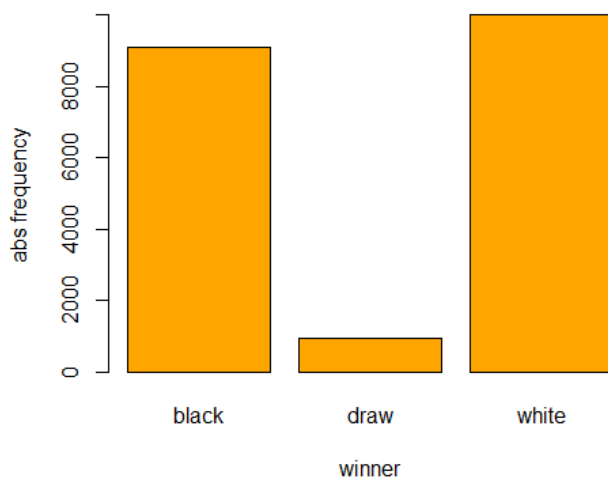


Figure 11: Boxplot of game turns distribution.

Given the large sample size, the 'turns' and the 'opening ply' distributions (fig9, fig10) allows us to infer that the major part of games are disputed as competitive ones as well as they ends with mate or resign of one player.

Last histogram (fig11) is more difficult to interpret. The two bins for black and white player assume almost the same value; in particular the white one seems slightly higher than the black one but it's legitimate to think of a statistical fluctuation. Certainly there are much less cases of draw.

The explorative analysis is based on the entire dataset that is composed of 16 attributes and 20058 statistical units.

Below a summary:

	turns	white rating	black rating	opening ply
Minimum:	1	784	789	1
1st Quartile:	37	1398	1391	3
Median:	55	1567	1562	4
Mean:	60	1597	1589	4.8
3rd Quartile:	79	1793	1784	6
Maximum:	349	2700	2723	28

rated	False	True
	3903	16155

victory Status	Draw	Mate	Out_of_Time	Resign
	906	6325	1680	11147



winner	Black	White	Draw
	9107	10001	950

Table 3: Summary of the explorative analysis.

R implementation for the explorative analysis:

```
dataset <- read.csv("games.csv") #Import dataset

#Definitions of discrete and categorical variables of interest
discrete_variables <- c('turns', 'white_rating', 'black_rating', 'opening_ply')
categorical_variables <- c('rated', 'victory_status', 'winner')

for (ch in discrete_variables) {} #prints the discrete variables histograms
for (ch in categorical_variables) {} #prints the discrete variables histograms
```

```
for (ch in discrete_variables) {
  # Shows frequency histograms and boxplots of all the discrete variables
  sub_set <- dataset[[ch]];
  plot <- ggplot(data = dataset[ch], mapping = aes(x=sub_set)) +
    geom_histogram(color = "black", fill = "blue") +
    xlab(ch) + ylab('abs frequency')+
    geom_vline(xintercept = mean(sub_set), linetype = "solid", color = "red", size = 0.3)+
    geom_vline(xintercept = median(sub_set), linetype = "dashed", color = "red", size = 0.3);
  print(plot)
  boxplot <- ggplot(data = dataset[ch], mapping = aes(y=sub_set)) + geom_boxplot() + ylab(ch)
  print(boxplot)
  print(summary(dataset[ch]))
} #prints the discrete variables histograms
for (ch in categorical_variables) {
  # Shows frequency histograms and boxplots of all the categorical variables
  sub_set <- dataset[[ch]];
  plot <- barplot(table(sub_set), col = 'orange', ylab = 'abs frequency', xlab = ch, horiz = FALSE)
  print(plot)
  print(table(sub_set))
} #prints the discrete variables histograms
```

```
sum <- dataset[1]
for(ch in discrete_variables) sum[length(sum)+1] <- dataset[ch]
for(ch in categorical_variables) sum[length(sum)+1] <- dataset[ch]
summary(sum)
```

Output:

```
> sum <- dataset[1]
> for(ch in discrete_variables) sum[length(sum)+1] <- dataset[ch]
> for(ch in categorical_variables) sum[length(sum)+1] <- dataset[ch]
> summary(sum)
  id      turns  white_rating  black_rating  opening_ply   rated  victory_status  winner
XRuQPSzH:  5  Min.   : 1.00   Min.   : 784   Min.   : 789   Min.   : 1.000  False:2048  draw   : 906  black: 9107
OgTDO6Av:  4  1st Qu.: 37.00   1st Qu.:1398  1st Qu.:1391  1st Qu.: 3.000  FALSE:1855  mate    : 6325  draw : 950
1b0kpInt:  4  Median : 55.00   Median :1567  Median :1562  Median : 4.000  True :8723  outoftime: 1680  white:10001
CvakmVNB:  4  Mean    : 60.47   Mean    :1597  Mean    :1589  Mean    : 4.817  TRUE :7432  resign  :11147
CxQlHSnq:  4  3rd Qu.: 79.00   3rd Qu.:1793  3rd Qu.:1784  3rd Qu.: 6.000
dFQ5D7CS:  4  Max.    :349.00   Max.    :2700  Max.    :2723  Max.    :28.000
(Other) :20033
```


New user's removal

“Is this dataset a sample of the population I’m referring to?”. The answer to this question is not that easy to give. As mentioned, the population on which the work is based would be a collective of games disputed by the group of chess players, amateurs or professionals, that are however quite present on the platform. Peaks in white-black histograms (fig1, fig2) raise few doubts.

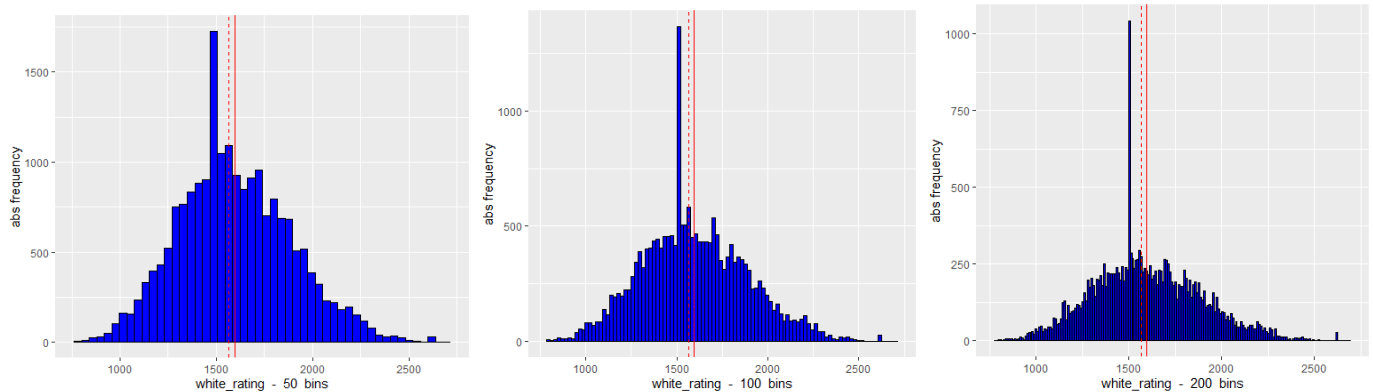


Figure 12: Distribution of absolute frequency of white rating (bins progression from 50 up to 200).

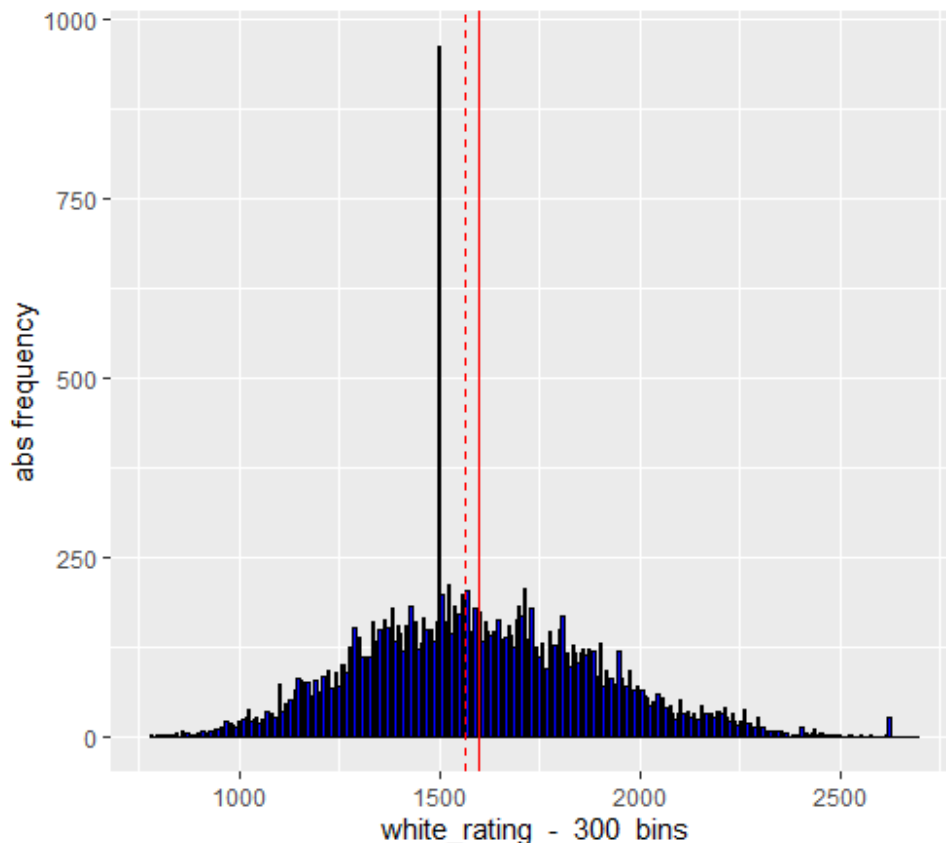


Figure 13: Distribution of absolute frequency of white rating (300 bins).

Looking at details of these distributions and increasing the number of bins (fig13) appears evident the singularity at the value of 1500. Something similar happens in the other distribution of black rating.

Height of that bin would justify a simple removal of the uncomfortable value, although this practise doesn't consider the very probable fact that most of these players have some other games in the dataset. This means there are many not representative games of the population, games in which the user's rating is swinging while finding the right class of belonging. For this reason two criteria have been devised aiming an appropriate removal:

- Foreach user nominated in the dataset, if the user's rating has been 1500 among all the matches he desputed, he will be classified as a new user. More than the 80% of players removed with this rule are almost certainly new users (look at bin in fig13).
- Foreach user nominated in the dataset, if the user's rating jumped by a step higher than 60 rating points at least once, he will be classified as a new user. This rule considers all players not respecting the classical rating steps. Players removed in this phase are those who are still finding their appropriate rating but didn't passed for the critical value of 1500.

The result after previous cuts are showed in figure 14 and figure 15.

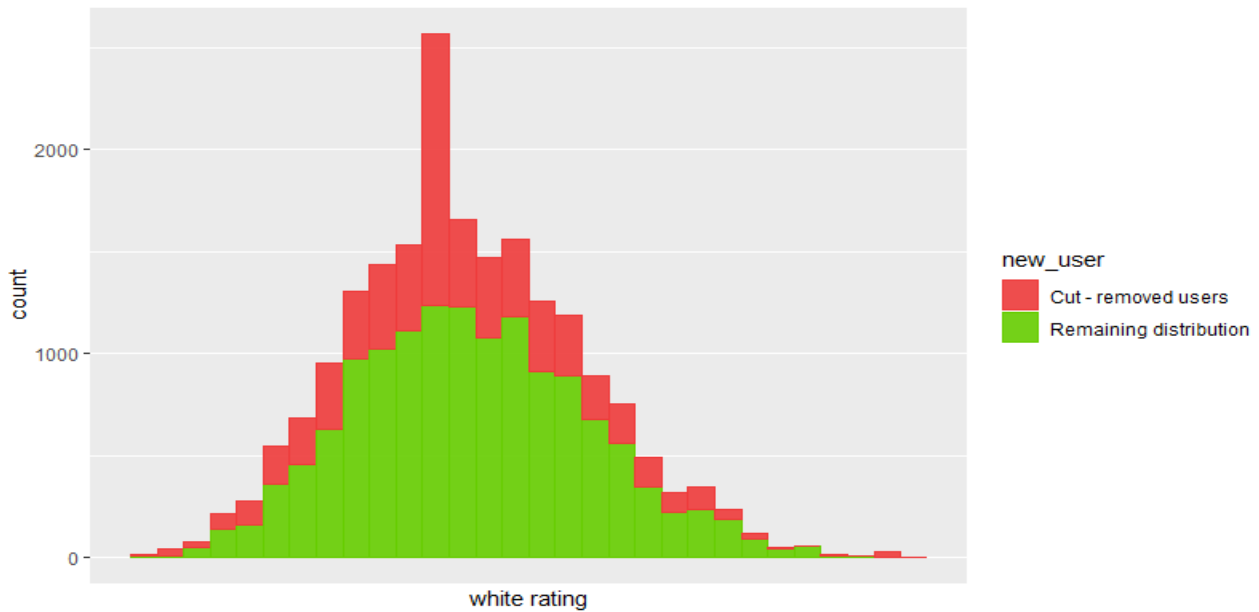


Figure 14: Distribution of absolute frequency of white rating with/without new user's removal.



Figure 15: Distribution of absolute frequency of black rating with/without new user's removal.

Short game's removal

By the same principle I applied another cut on the turns character. Games with less than 5 turns (less than 3 move per player) have been considered much less significant of the rest of matches. In figure 16 is shown the game status distribution for removed data.

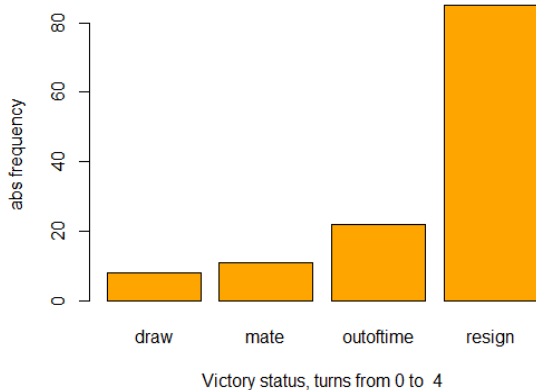


Figure 15: Distribution of absolute frequency of victory status only for games with turns between 0 and 4.

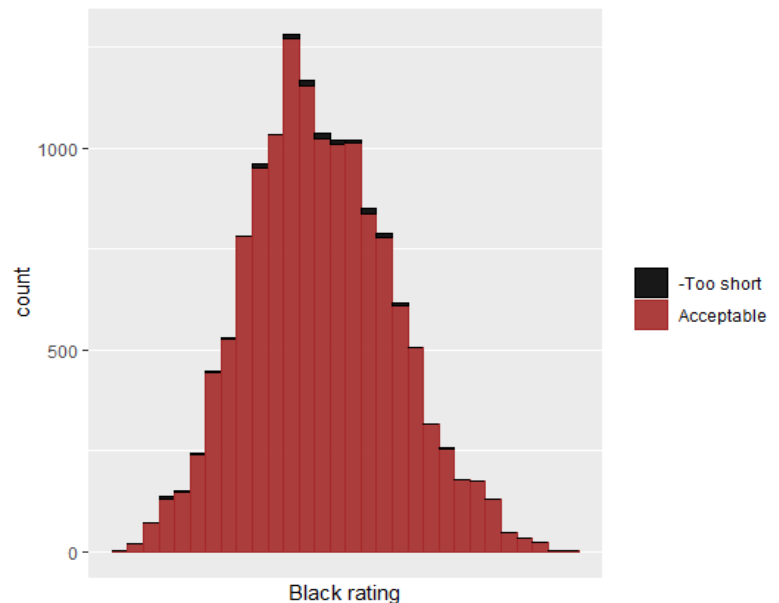


Figure 16: Distribution of absolute frequency of black rating before (brown+black) and after (brown) the short game removal.

There is a huge difference between this distribution (fig 15) and the initial one (fig 10) especially for the 'mate' bin which seems to be of the same order of magnitude as the 'draw' bin. At this range of turns, matches are mostly solved in a resignation, in fact before the games gets to what we could define as their cores. Probably this behavior is caused by lack of time or maybe a poor connection from one opponent.

In any case, these kind of games have been removed from this work for a cleaner analysis.

Black rating histogram (fig 16) shows a small change in distribution, especially for central values. The same effect is been obtained for the white rating distribution.

R implementation for removals:

```
dataset_BR <- preparation(dataset) #Dataset before Removal
# into this method there are few changes of dataset, useful
# for the continuation of the analysis

printHisto_byBins(mydataset, 'white_rating', 50) # |
printHisto_byBins(mydataset, 'white_rating', 100) # | progressively increasing
printHisto_byBins(mydataset, 'white_rating', 200) # | the number of bins
printHisto_byBins(mydataset, 'white_rating', 300) # v

dataset_newUsersR <- filter_newUsers(dataset_BR, max_difference = 60,
                                     excluded_values = c(1500))

#from now on will work with this dataset
mydataset <- filter_shortgames(dataset_newUsersR, min_turns = 4)
```

```

filter_newUsers <- function(dataset, max_difference, excluded_values){

  new_dataset <- dataset
  users <- c()
  for (name in dataset$white_id){
    a <- c()
    for (rating in dataset$white_rating[dataset$white_id == name]){
      a <- c(a, rating)
    }
    for (rating in dataset$black_rating[dataset$black_id == name]){
      a <- c(a, rating)
    }
    a <- sort(a)
    if(length(a) > 1){
      elem <- a[1]
      for (rat in a){
        if(rat - elem > max_difference) {users <- c(users, name); break; }
        excluded = FALSE
        for (value in excluded_values){
          if(elem == value) {users <- c(users, name); excluded = TRUE; }
          if(excluded) break;
        }
        elem <- rat
      }
    }
    else if (a[1] == 1500) {users <- c(users, name);}
  }

  dataset['new_user'] = 'Remaining distribution'
  for (name in users){
    dataset$new_user[dataset$white_id == name | dataset$black_id == name] <- 'Cut - removed users'
  }
  new_dataset <- dataset %>% filter (new_user == 'Remaining distribution')

  plot <- ggplot(data = new_dataset, mapping = aes(x=new_dataset[['white_rating']])) +
    geom_histogram(color = "black", fill = "blue") +
    xlab('white rating (new users removal)') + ylab('abs frequency')+
    geom_vline(xintercept = mean(new_dataset[['white_rating']]), linetype = "solid", color = "red",
    geom_vline(xintercept = median(new_dataset[['white_rating']]), linetype = "dashed", color = "red"
  #print(plot)
  plot <- ggplot(data = new_dataset, mapping = aes(x=new_dataset[['black_rating']])) +
    geom_histogram(color = "black", fill = "blue") +
    xlab('black rating (new users removal)') + ylab('abs frequency')+
    geom_vline(xintercept = mean(new_dataset[['black_rating']]), linetype = "solid", color = "red",
    geom_vline(xintercept = median(new_dataset[['black_rating']]), linetype = "dashed", color = "red"
  #print(plot)
  print(paste('Dataset with new-users removal length: ', length(new_dataset[[1]])))

  palette <- c("brown2", "chartreuse3")
  plot <- ggplot(dataset, aes(x=white_rating, fill=new_user, color = new_user)) +
    geom_histogram(alpha=0.9, position = "stack") + xlab('white rating') +
    scale_x_continuous(breaks = seq(-10, 10, by = 1))+scale_fill_manual(values=palette)+
    scale_color_manual(values=palette)
  print(plot)
  plot <- ggplot(dataset, aes(x=black_rating, fill=new_user, color = new_user)) +
    geom_histogram(alpha=0.9, position = "stack") + xlab('black rating') +
    scale_x_continuous(breaks = seq(-10, 10, by = 1))+scale_fill_manual(values=palette)+
    scale_color_manual(values=palette)
  print(plot)

  return(new_dataset)
}

```

```

filter_shortgames <- function(act_dataset, min_turns){
  new_dataset <- act_dataset
  new_dataset['not_short'] = '-Too short'
  new_dataset$not_short[new_dataset$turns > min_turns] <- 'Acceptable'
  palette = c('black', 'brown')
  print_coloredHistogram(new_dataset, 'not_short', 'white_rating', 'White rating', palette)
  print_coloredHistogram(new_dataset, 'not_short', 'black_rating', 'Black rating', palette)

  sub_set <- new_dataset %>% filter (not_short == '-Too short')
  sub_set_vec <- sub_set[['victory_status']];
  print(barplot(table(sub_set_vec), col = 'orange', ylab = 'abs frequency',
    xlab = paste('Victory status, turns from 0 to ', min_turns), horiz = FALSE))
  new_dataset <- new_dataset %>% filter (turns > min_turns)
  print(paste('Dataset with short-games removal length: ', length(new_dataset[[1]])))
  return(new_dataset)
}

```

```

preparation <- function(dataset){
  new_dataset <- dataset
  levels(new_dataset$rated) <- c('FALSE', 'FALSE', 'TRUE', 'TRUE') #Reassign levels by semantic
  new_dataset['game_time'] <- new_dataset[['turns']]
  new_dataset['increment_time'] <- new_dataset[['turns']]
  i=0
  for(ch in new_dataset$increment_code){
    i = i+1
    ch1 <- str_replace_all(ch, "[^[:alnum:]]", " ")
    ch2 <- strsplit(ch1, " ")
    ch3 <- unlist(ch2)
    new_dataset[i, 'game_time'] <- ch3[[1]]
    new_dataset[i, 'increment_time'] <- ch3[[2]]
  }

  new_dataset <- rbind_rating_difference(new_dataset, 'winner')
  new_dataset['winner_int'] <- new_dataset[['winner']]
  levels(new_dataset$winner_int) <- c(-1, 0, 1)

  new_dataset['group'] <- 'not expected'

  for(i in 1:length(new_dataset[[1]])){
    if(new_dataset[i, 'winner_int'] == 1 & new_dataset[i, 'rating_difference'] < 0)
    {dataset[i, 'group'] <- 'not expected'}
    else if(new_dataset[i, 'winner_int'] == -1 & new_dataset[i, 'rating_difference'] < 0)
    {new_dataset[i, 'group'] <- 'expected'}
    else if(new_dataset[i, 'winner_int'] == 1 & new_dataset[i, 'rating_difference'] > 0)
    {new_dataset[i, 'group'] <- 'expected'}
    else if(new_dataset[i, 'winner_int'] == -1 & new_dataset[i, 'rating_difference'] > 0)
    {new_dataset[i, 'group'] <- 'not expected'}
    else if(new_dataset[i, 'rating_difference'] == 0) {new_dataset[i, 'group'] <- 'uncertain'}

    if(new_dataset[i, 'rating_difference'] > -100 & new_dataset[i, 'rating_difference'] < 100)
    {new_dataset[i, 'group'] <- 'uncertain'}
  }

  new_dataset['rating_difference_div100'] <- as.integer(new_dataset[['rating_difference']] / 100)
  new_dataset['game_result'] <-
    strtoi(new_dataset[['winner_int']], base = 16L) / new_dataset[['turns']]

  return(new_dataset)
}

print_coloredHistogram <- function(mydataset, color, x, xlab, palette){
  plot <- ggplot(mydataset, aes(x=mydataset[[x]], fill=mydataset[[color]], color = mydataset[[color]])
  geom_histogram(alpha=0.9, position = "stack") + xlab(xlab)+ theme(legend.title = element_blank())
  scale_x_continuous(breaks = seq(-10, 10, by = 1))+scale_fill_manual(values=palette)+
  scale_color_manual(values=palette)
  print(plot)
  rm(plot)
}

```

Below a brief summary of the changes applied to the dataset:

Phase	Operations	Units
Preparation	renaming of 'rated' observed mode into only two univocal values (True of False) splitting the 'increment code' into two separate numerical variables creating character 'winner_int' which mask the 'winner' variable with values: -1,0,1 .. 'group' with levels: 'expected' (or not) and 'uncertain' .. 'rating_difference' given by white minus black rating .. 'rating_difference_div100' that represent a subdivision in classes of rating_difference .. 'game_result' given by 'rating_difference' divided by 'winner_int'	20058
Filtering	New user's removal Short game's removal	13851 13725

Table 4: Despriction of all the operations performed on the dataset.

Hypothesis of normality in rating distributions

It's very important in view of a statistical analysis to know what kind of distributions we are facing. Among the most recognizable distributions there is the normal one: white and black rating distributions seems to fit it pretty good and for this reason I considered it appropriate to investigate better.

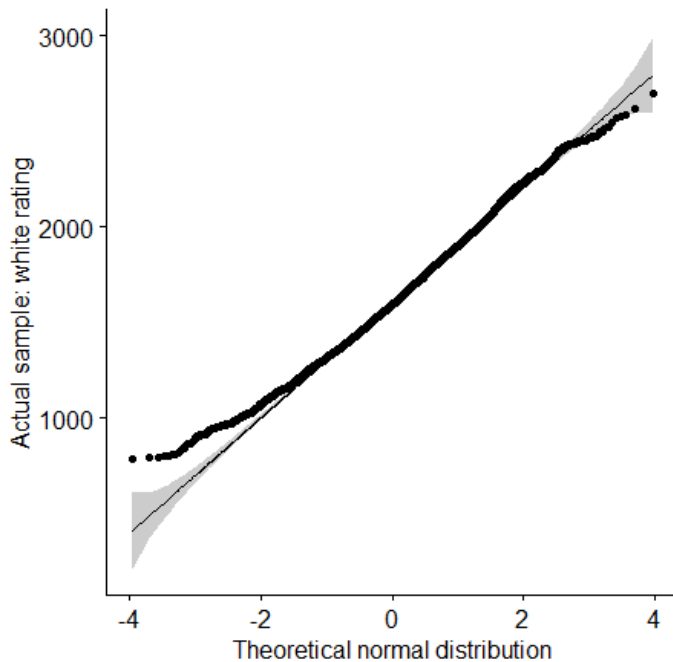


Figure 17: Q-Q plot of white rating (after removals).

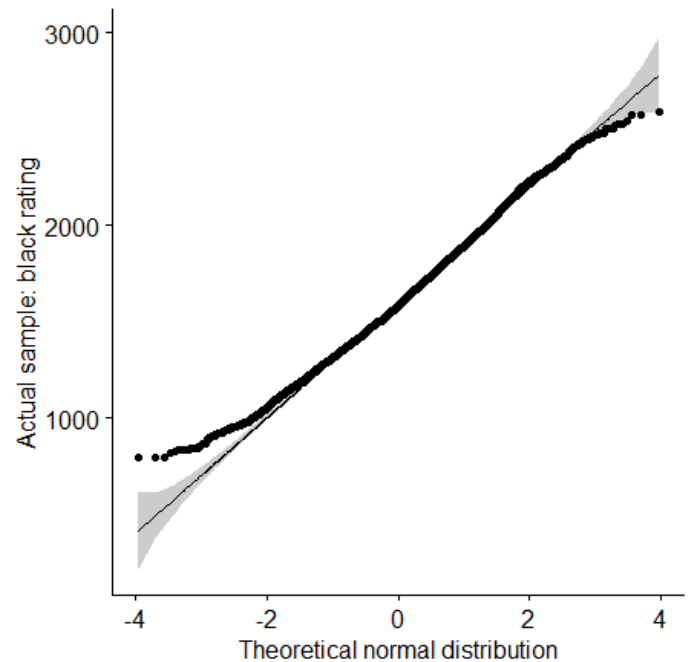


Figure 18: Q-Q plot of black rating (after removals).

Even if rating distributions look like normal distributions at first sight, the large deviation between the sample and theoretical distribution combined with the large sample size (fig17, fig18), does not lay well.

For greater understanding of the phenomenon I did a Shapiro-Wilk test of normality highlighting the truthfulness of the not normality suspect. Technically, the dataset has been splitted in three groups with less than 5000 statistical units and I've obtained the following values:

	N. subset	W- statistic	p-value
White	1	0.987	1.93×10^{-19}
	2	0.998	2.28×10^{-6}
	3	0.999	0.050
Black	1	0.989	1.99×10^{-18}
	2	0.999	2.49×10^{-8}
	3	0.997	0.013

Table 5: Result of the Shapiro-Wilk test performed on 3 subset. It is shown the relative W-statistics and p-values.

From these data it seems that only 3° subset shows a compatibility with the normal distribution within a statistical significance of 5%. Even though, the null hypothesis of normality for rating distributions has been rejected because of the incredible low p-value in the first two subsets.

R implementation for the hypothesis of normality:

```
ggqqplot(mydataset, x= "white_rating") #Distribution is normally distributed?
ggqqplot(mydataset, x= "black_rating")
shapiroTest_3groups(mydataset)
```

```
shapiroTest_3groups <- function(dataset){
  n <- length(dataset[[1]])
  groups <- 3
  step <- as.integer(n/groups)
  minus_dataset1 <- dataset[1:step, ]
  minus_dataset2 <- dataset[step+1:step*2, ]
  minus_dataset3 <- dataset[step*2 + 1:step*3, ]
  a <- shapiro.test(minus_dataset1[['white_rating']])
  b <- shapiro.test(minus_dataset1[['black_rating']])
  c <- shapiro.test(minus_dataset2[['white_rating']])
  d <- shapiro.test(minus_dataset2[['black_rating']])
  e <- shapiro.test(minus_dataset3[['white_rating']])
  f <- shapiro.test(minus_dataset3[['black_rating']])
  print(paste('Dataset1 - White', ' W-statistic: W =', a[[1]], ' p-value: ', a[2]))
  print(paste('Dataset1 - Black', ' W-statistic: W =', b[[1]], ' p-value: ', b[2]))
  print(paste('Dataset2 - White', ' W-statistic: W =', c[[1]], ' p-value: ', c[2]))
  print(paste('Dataset2 - Black', ' W-statistic: W =', d[[1]], ' p-value: ', d[2]))
  print(paste('Dataset3 - White', ' W-statistic: W =', e[[1]], ' p-value: ', e[2]))
  print(paste('Dataset3 - Black', ' W-statistic: W =', f[[1]], ' p-value: ', f[2]))
}
```

Output:

```
> shapiroTest_3groups(mydataset)
[1] "Dataset1 - White W-statistic: W = 0.987825017470463 p-value: 1.93461383247452e-19"
[1] "Dataset1 - Black W-statistic: W = 0.988983113568968 p-value: 1.99451297762181e-18"
[1] "Dataset2 - White W-statistic: W = 0.997696897028605 p-value: 2.28394634386815e-06"
[1] "Dataset2 - Black W-statistic: W = 0.996790872708274 p-value: 2.4939567803058e-08"
[1] "Dataset3 - White W-statistic: W = 0.997929447587918 p-value: 0.0504499742735996"
[1] "Dataset3 - Black W-statistic: W = 0.997398127911836 p-value: 0.0133568781459076"
```

The explanation for the not normality of distributions that seems to should arrange naturally as Gaussian ones, is to be found in effects of other factors in this work have not been considered. Some of them could be again the default assignation from Lichess to the 1500 value that produces an irreducible background noise around that value. This factor should increase the mass distribution on central region and decrease the same on tails. Another factor could be found in the lost of motivation from low rated players which tend to play less than the theoretical distribution would suggest. This factor may find it's realization into the subpopulation on the left tail of distribution.

Hypothesis of equality between white and black

In this section will be tested the hypothesis according to which being white or black doesn't count towards the outcome of the game. For this purpose it's been analyzed a subset containing all games that see or white or black victorious.

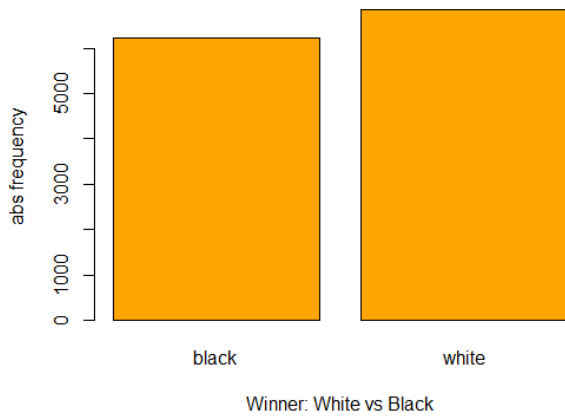


Figure 19: Histogram of the dichomous variable 'winner'.

	Black	White
Numerosity	6237	6845
Percentage	47.8%	52.2%

Table 6: Numerosity and percentage of white and black distributions within the subset composed by these only two outcomes.

The histogram in figure 19 shows the distributional absolute frequency of the two possible outcomes (from the 13725 original samples have been removed 643 games ended with a draw) while the table 6 highlights the slight asymmetry of victory percentage between white and black.

To check if this difference is significant I opted for another test: being these two attributes the only two outcome of a game (the draw games have been excluded) it is legitimate to think of the variable 'winner' as one that is distributed as a Binomial and for that I used a Binomial test.

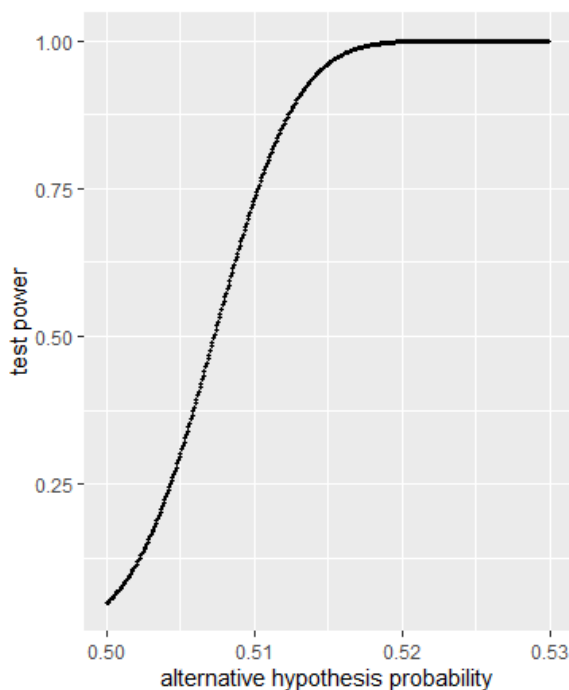


Figure 20: Test power as a function of alternative hypothesis probability into a Binomial test with null hypothesis = 0.522 and n = 13082.

Null Hypothesis	P-value
0.510	0.001
0.513	0.01
0.516	0.05

Table 7: p-value with null hypothesis into a Binomial test with observed value of 0.522 and n = 13082.

Test has been conducted fixing the null hypothesis at the value of 50% while the alternative one increases progressively from that value (one-tailed), obtaining the beta power as in figure 20.

Subsequently, have been found the values of null hypothesis at specified p-values confirming the meaningfulness of the empirical finding that white has an advantage on the black. In particular it can be said that with confidence intervals of 99.9%, 99.0% and 95.0% the probability for white to win on black is greater than 51.0%, 51.3% and 51.6% respectively (table7).

R implementation for the hypothesis of equity:

```
print_whitevsblackHistogram(mydataset)

print(binomialTest(mydataset, null_hyp_prob = 0.5))
print(binomialTest(mydataset, null_hyp_prob = 0.513))
print(binomialTest(mydataset, null_hyp_prob = 0.516))
```

```
print_whitevsblackHistogram <- function(dataset){
  new_dataset <- dataset %>% filter(winner != 'draw')
  levels(new_dataset$winner) <- c('black', 'white', 'white') #remove draw character
  sub_set_vec <- new_dataset[['winner']];
  print(barplot(table(sub_set_vec), col = 'orange', ylab = 'abs frequency',
                    xlab = paste('Winner: White vs Black'), horiz = FALSE))
  print(summary(sub_set_vec))
}
```

```
binomialTest <- function(dataset, null_hyp_prob, alt_hyp_prob = 1, conf_level = 0.95){
  new_dataset <- dataset %>% filter(dataset$winner != 'draw')
  n = length(new_dataset[[1]])
  m <- new_dataset %>% filter(new_dataset$winner == 'white')
  success = length(m[[1]])
  test <- binom.test(c(success,n-success), p = null_hyp_prob,
                    alternative = c("greater"),
                    conf.level = conf_level)

  print(test)

  if(test[3]<1-conf_level){ #when the null hypothesis is refuted
    prev_i <-0
    for(i in as.integer(n/2):success){
      act_i <- pbinom(i, n, null_hyp_prob)
      if(prev_i <= conf_level & act_i >= conf_level) break;
      prev_i <- act_i
    }
    beta <- pbinom(i, n, alt_hyp_prob)
    options(scipen = 0)
    options(digits = 2)
    result = 1-beta
    if(result < 0.0001) result <- formatC(1-beta, format = "e", digits = 3)
    else result <- formatC(1-beta, digits = 3)
    print(paste('Test power -> ', result))
    j <- i
    x <- seq(as.integer(n*null_hyp_prob), as.integer(n*0.53), by=1)
    power <- c()
    for( j in x){
      power <- c(power, 1-pbinom(i, n, j/n))
    }
    for(i in 1:length(x)) x[[i]]<- x[[i]]/n
    plot <- ggplot(mapping = aes(x=x, y = power)) + geom_point(size = 0.4) +
      xlab('alternative hypothesis probability') + ylab('test power')
    rm(n); rm(m); rm(new_dataset); rm(success); rm(test)
    return(plot)
  }
}
```

Descriptive analysis

To better analyze the dataset, have been included in it some other numerical variables as specified in the table 4. Some of them are just a numerical mask to categorical variables such as 'winner_int' that convert the levels 'white', 'draw' and 'black' into the integers '1', '0', '-1'. Other such as 'rating_difference' and 'game_result' have been created to clarify the relation between the speed of victory (or defeat) for the first opponent and the gap of rating between him and the second one. The 'game_result' variable in particular is defined by the relation: $game_result = \frac{winner_int}{turns}$.

In figure 21 is displayed the graphic matrix of correlations of numerical variables in the dataset.

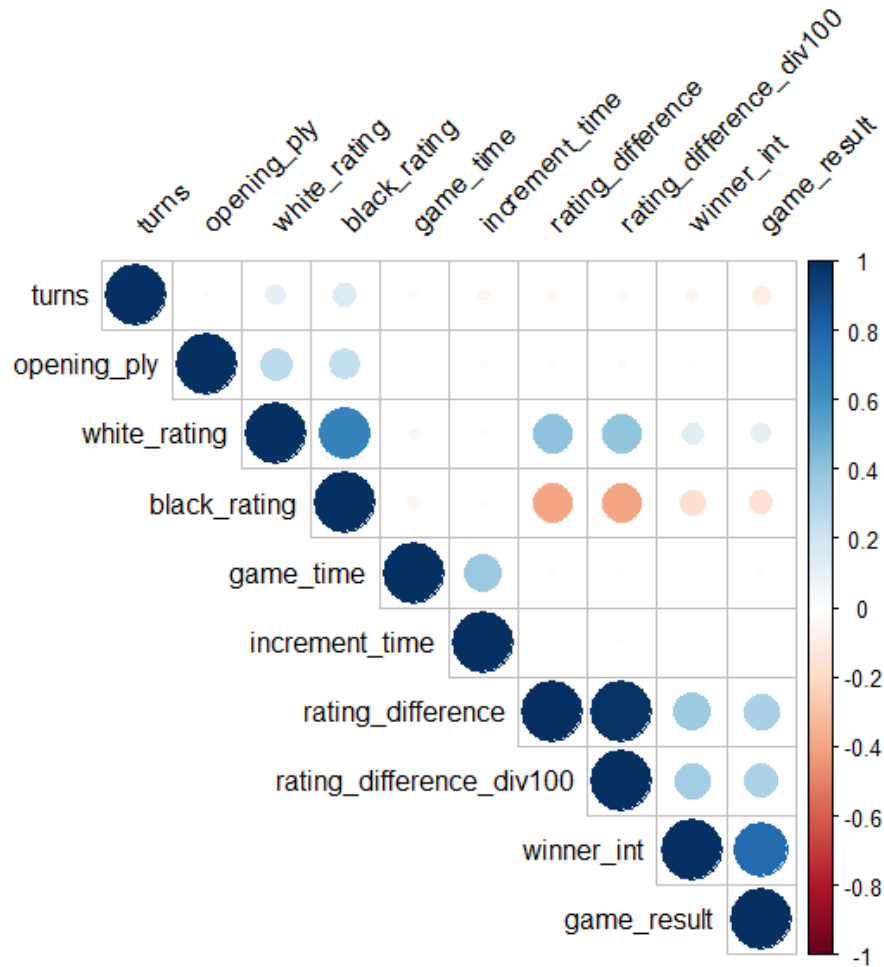


Figure 21: Graphic matrix of correlations between numerical variables of the dataset.

Each correlation is associated with a p-value, that is the probability for the observed correlation to fall within the confidence interval defined by $1 - pvalue$, based on the null hypothesis of no correlation.

Deepening some of these correlations:

- It's obvious the bond between `rating_difference_div100` and `rating_difference` cause of the simple linear relation: $rating_difference_div100 = \frac{rating_difference}{100}$.
- More interesting is the relation between 'white_rating' and 'black_rating' attributes. This correlation could be explained in terms of Lichess policies: when a user finds a new game without specify any conditions, the system try to find for him a similar-rating opponent. So it's legitimate to think of some kind of linear relation as: $white_rating = black_rating$. Value of linear correlation index is equal to 0.67 with a p-value ~ 0.

- Last strong and expected correlation is between variables 'game_result' and 'winner_int' again because of the definition of the first one.
- Another correlation it seemed me legitimate to think of is the bond between 'game_result' and 'rating_difference'. The aim of the definition of the rating is precisely the enstablishment of a relationship of order between the users and for this reason i expect to see higher rated player win, as much quickly as their rating is greater than the opponents one. In this case, the value of linear correlation index is equal to 0.33 with a p-value ~ 0.

R implementation for the matrix of correlations:

```
num_dataset <- Numerical_Dataset(mydataset)
rcorr(as.matrix(num_dataset))
corrplot(cor(num_dataset), type = "upper", order = "hclust", tl.col = "black", tl.srt = 45)
```

```
Numerical_Dataset <- function(mydataset){
  numerical_dataset <- mydataset['turns']
  numerical_dataset <- cbind(numerical_dataset, mydataset['white_rating'])
  numerical_dataset <- cbind(numerical_dataset, mydataset['black_rating'])
  numerical_dataset <- cbind(numerical_dataset, mydataset['opening_ply'])
  numerical_dataset <- cbind(numerical_dataset, mydataset['game_time'])
  numerical_dataset <- cbind(numerical_dataset, mydataset['increment_time'])
  numerical_dataset <- cbind(numerical_dataset, mydataset['rating_difference'])
  numerical_dataset <- cbind(numerical_dataset, mydataset['winner_int'])
  numerical_dataset <- cbind(numerical_dataset, mydataset['rating_difference_div100'])
  numerical_dataset <- cbind(numerical_dataset, mydataset['game_result'])
  return (numerical_dataset)
}
```

Output:

	turns	white_rating	black_rating	opening_ply	game_time	increment_time	rating_difference	winner_int	rating_difference_div100	game_result
turns	1.00	0.12	0.14	0.01	-0.03	-0.04	-0.03	-0.05	-0.03	-0.09
white_rating	0.12	1.00	0.67	0.26	-0.04	-0.01	0.41	0.13	0.40	0.11
black_rating	0.14	0.67	1.00	0.25	-0.05	-0.02	-0.40	-0.16	-0.39	-0.16
opening_ply	0.01	0.26	0.25	1.00	0.01	0.02	0.02	0.02	0.02	0.00
game_time	-0.03	-0.04	-0.05	0.01	1.00	0.38	0.01	0.00	0.02	0.02
increment_time	-0.04	-0.01	-0.02	0.02	0.38	1.00	0.01	0.00	0.01	0.00
rating_difference	-0.03	0.41	-0.40	0.02	0.01	0.01	1.00	0.37	0.98	0.33
winner_int	-0.05	0.13	-0.16	0.02	0.00	0.00	0.37	1.00	0.35	0.77
rating_difference_div100	-0.03	0.40	-0.39	0.02	0.02	0.01	0.98	0.35	1.00	0.32
game_result	-0.09	0.11	-0.16	0.00	0.02	0.00	0.33	0.77	0.32	1.00

n= 13725

P	turns	white_rating	black_rating	opening_ply	game_time	increment_time	rating_difference	winner_int	rating_difference_div100	game_result
turns	0.0000	0.0000	0.1596	0.0007	0.0000	0.0001	0.0000	0.0000	0.0000	0.0000
white_rating	0.0000	0.0000	0.0000	0.0000	0.1625	0.0000	0.0000	0.0000	0.0000	0.0000
black_rating	0.0000	0.0000	0.0000	0.0000	0.0000	0.0226	0.0000	0.0000	0.0000	0.0000
opening_ply	0.1596	0.0000	0.0000	0.2918	0.0136	0.0112	0.0377	0.0120	0.0120	0.6161
game_time	0.0007	0.0000	0.0000	0.2918	0.0000	0.1594	0.5701	0.0692	0.0212	0.0212
increment_time	0.0000	0.1625	0.0226	0.0136	0.0000	0.2776	0.9904	0.1346	0.9799	0.9799
rating_difference	0.0001	0.0000	0.0000	0.0112	0.1594	0.2776	0.0000	0.0000	0.0000	0.0000
winner_int	0.0000	0.0000	0.0000	0.0377	0.5701	0.9904	0.0000	0.0000	0.0000	0.0000
rating_difference_div100	0.0000	0.0000	0.0000	0.0120	0.0692	0.1346	0.0000	0.0000	0.0000	0.0000
game_result	0.0000	0.0000	0.0000	0.6161	0.0212	0.9799	0.0000	0.0000	0.0000	0.0000

Linear regressions

Scatter plot in figure 22 shows the relation between 'white_rating' and 'black_rating' attributes.

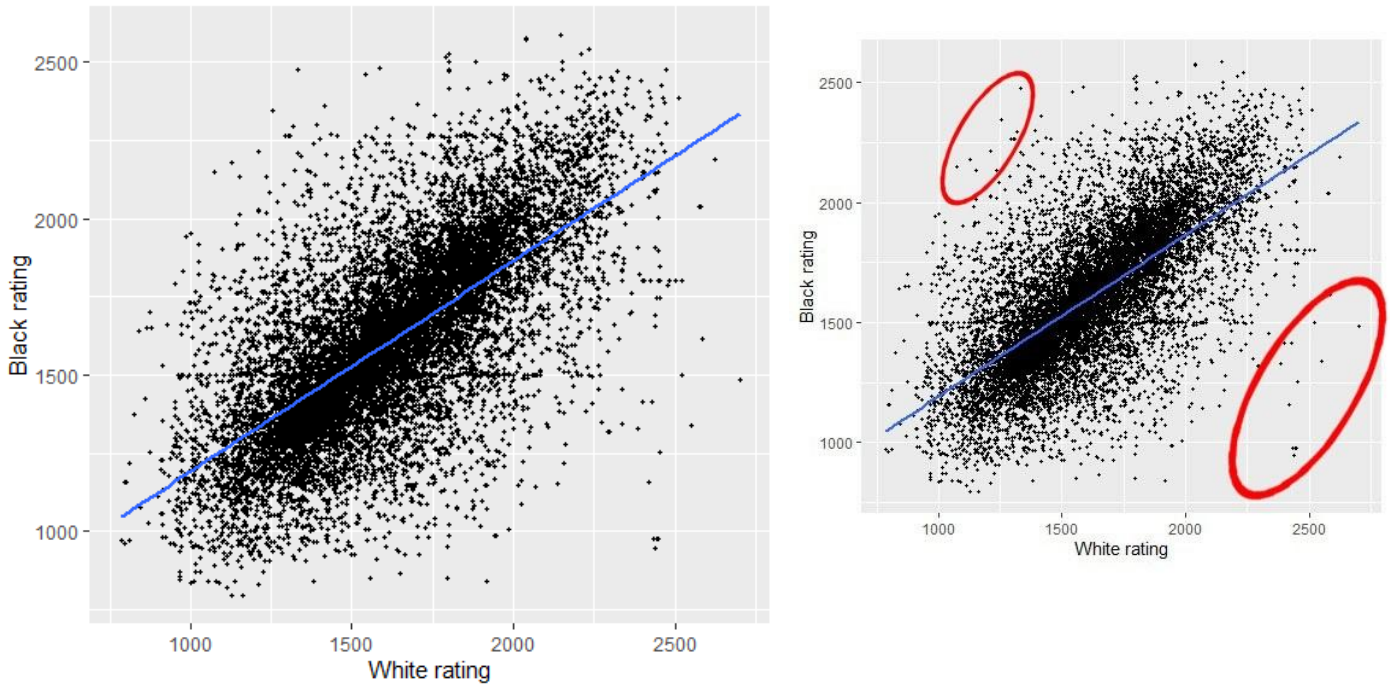


Figure 22: Scatter plot of black rating with white rating. To the right there are highlighted the outliers.

Despite the correlation is evident, there are some issues. First of all we have an angular coefficient which is not compatible with 1. Second, there is the presence of heavy outliers which probably pull the angular coefficient under the unitary value while pushing up the intercept one. Finally we see a distribution characterized by a substantial internal variability which explains the relatively low R^2 value of 0.455.

```
print_coloredScatterplot(mydataset, color = '', x = 'white_rating', y = 'black_rating',
                        xlab = 'White rating', ylab = 'Black rating');
```

```
print_coloredScatterplot <- function(actual_dataset, color, x, y, xlab, ylab){
  plot <- ggplot(data = actual_dataset, mapping = aes(x=mydataset[[x]], y = mydataset[[y]],
                                                    color = mydataset[[color]])) +
    labs(color = "    Result") + geom_point(size = 0.8) +
    xlab(xlab) + ylab(ylab) + theme(legend.title = element_text(color = "blue", size = 10)) +
    geom_smooth(aes(color = NULL), method = lm, se = FALSE)
  print(plot)
  regression = lm(actual_dataset[[y]]~actual_dataset[[x]]);
  print(summary(regression));
  rm(plot); rm(regression);
}
```

Output of linear regression applied on 'black_rating' ~ 'white_rating' relation:

```
Call:
lm(formula = actual_dataset[[y]] ~ actual_dataset[[x]])

Residuals:
    Min       1Q   Median       3Q      Max
-1213.1  -109.3    -3.7   110.5  1058.0

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.203e+02  1.026e+01   50.7   <2e-16 ***
actual_dataset[[x]] 6.722e-01  6.275e-03  107.1   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 213.2 on 13723 degrees of freedom
Multiple R-squared:  0.4554,    Adjusted R-squared:  0.4554
F-statistic: 1.148e+04 on 1 and 13723 DF,  p-value: < 2.2e-16
```

To show the bond between outcome of the games and rating differences I divided the ‘rating_difference’ attribute into classes according to the following criterion:

- the class 0 includes all games with a rating difference between -100 and 100 (games I tagged as ‘uncertain’);
- all classes $n > 0$ include the games with a rating difference between $n \cdot 100$ and $(n + 1) \cdot 100$
- all classes $n < 0$ include the games with a rating difference between $(n - 1) \cdot 100$ and $n \cdot 100$

Games included into positive classes which see white victorious have been tagged with ‘expected’ as well as games included into negative classes which see black victorious. Remaining games have been tagged with ‘unexpected’.

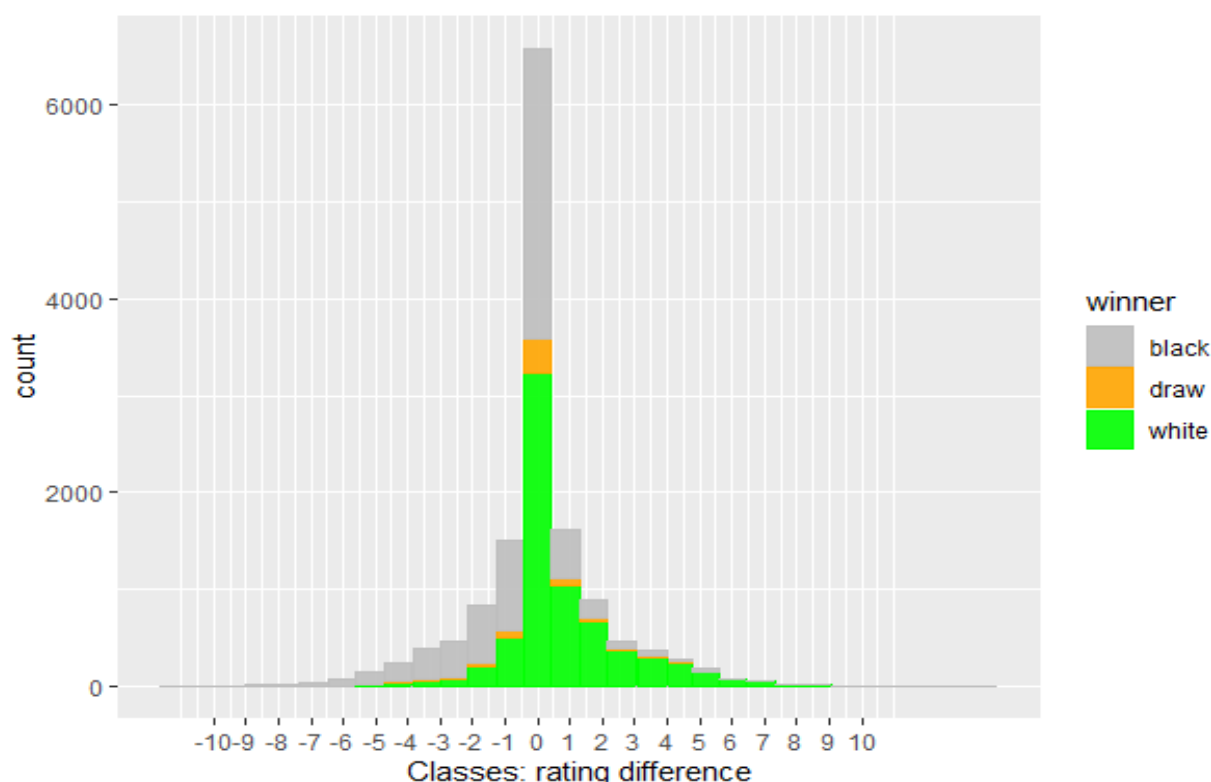


Figure 23: Histogram of absolute frequency with rating differences splitted into classes.

From the histogram in figure 23 we can see a predominance of white within positive classes and a predominance of black within negative ones.

	≤ -1	0	≥ 1			≤ -1	0	≥ 1			Truth	≤ -1	≥ 1
white	875	3230	2675	6780	white	6,4%	23,7%	19,6%	49,8%		white	13%	40%
draw	133	348	153	634	draw	1,0%	2,6%	1,1%	4,7%		black	35%	12%
black	2376	3004	826	6206	black	17,4%	22,1%	6,1%	45,6%				
	3384	6582	3654	13620		24,8%	48,3%	26,8%	100,0%				

Table 8: absolute (left) and relative (center) frequency table of winner with rating_difference_div100. To the right, the truth table between white / black victory and their belonging to the two cumulative levels $\leq -1 / \geq 1$

Truth table in table 8 shows quantitatively this behaviour. In particular it implies that 75% of games in which or white or black (having at least 100 rating point of gap) wins, will be correctly classified as ‘expected’. To test if this result could be inferred to the population I performed once again a binomial test.

Supposing that the dichotomous variable ‘expected’ – ‘not expected’ is distributed as a binomial having the mean value at the true probability for a correct classification into the ‘expected’ group, I had the following result shown in table 9.

Null Hypothesis	P-value
0.731	0.001
0.736	0.01
0.739	0.05

Table 9: p-value with null hypothesis into a Binomial test with observed value of 0.748 and n = 6752.

Test has been conducted fixing the p-value at first species errors of 0.001, 0.01 and 0.5 obtaining the corrisponging values of null hypothesis (rejected). With confidence intervals of 99.9%, 99.0% and 95.0% the probability for a correct classification therefore is greater than 73.1%, 73.6% and 73.9% respectively .

R implementation:

```
binom.test(c(5051, 1701), p = 0.7314, alternative = c("greater"),conf.level = 0.999)
binom.test(c(5051, 1701), p = 0.7356, alternative = c("greater"),conf.level = 0.99)
binom.test(c(5051, 1701), p = 0.7392, alternative = c("greater"),conf.level = 0.95)
```

Output:

```
Exact binomial test

data:  c(5051, 1701)
number of successes = 5051, number of trials = 6752, p-value = 0.04938
alternative hypothesis: true probability of success is greater than 0.7392
95 percent confidence interval:
 0.7392322 1.0000000
sample estimates:
probability of success
      0.7480746
```

As mentioned above, a possible linear relation could be found between ‘game_result’ and ‘rating_difference’ attributes.

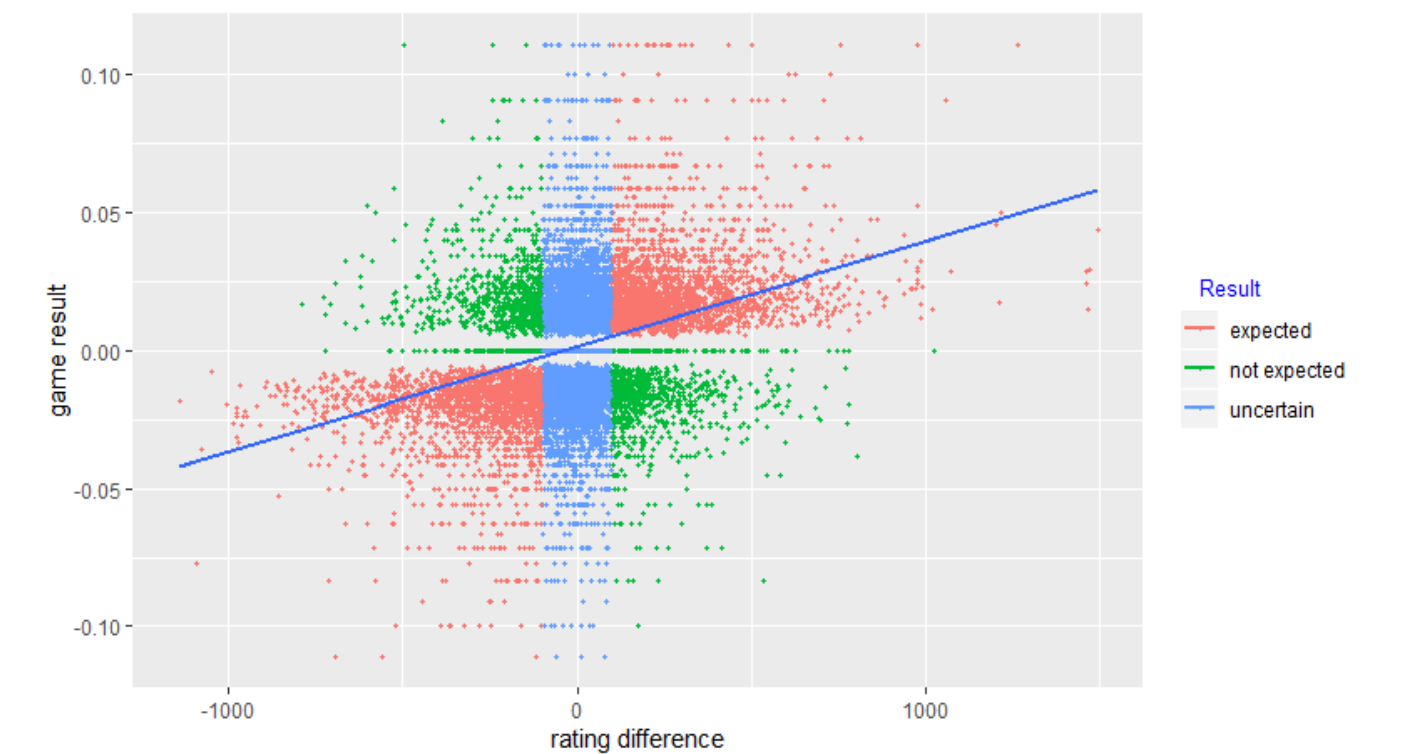


Figure 24: Scatter plot of game result with rating difference.

Even in the scatter plot (fig 24) of game result with the rating difference the asymmetry between the ‘expected’ and ‘not expected’ populations is evident. However, despite the visible correlation it’s quite difficult to imagine a linear relationship because of the huge variability.

In fact, a linear regression has been conducted obtaining an incredibly low R^2 value of 0.108 .

R implementation of descriptive analysis and linear regressions:

```
print(make_classes(mydataset, 'winner_int', 'rating_difference_div100'));
print_coloredHistogram(mydataset, color = 'winner', x = 'rating_difference_div100',
                        xlab = 'Classes: rating difference', palette = c("gray", "orange", "green"));
```

```
make_classes <- function(mydataset, group1, group2){
  matr <- as.matrix(table(mydataset[[group1]], mydataset[[group2]]))
  inf <- 0; sup <- 0;
  i=0; for (name in colnames(matr)) {i=i+1; if(name=="-1") inf <- i; if(name=="1") sup <- i;}
  for(i in 1:3){matr[i,inf] <- sum(matr[i,1:inf]); matr[i,1:(inf-1)] <- 0 }
  for(i in 1:3){matr[i,sup] <- sum(matr[i,sup:length(matr[i,])]); matr[i,(sup+1):length(matr[i,])] <- 0 }
  matr <- matr[ , inf:sup]
  return(matr)
}
```

```
print_coloredScatterplot(mydataset, color = 'group', x = 'rating_difference', y = 'game_result',
                        xlab = 'rating difference', ylab = 'game result');
```

```
print_coloredHistogram <- function(mydataset, color, x, xlab, palette){
  plot <- ggplot(mydataset, aes(x=mydataset[[x]], fill=mydataset[[color]], color = mydataset[[color]])) +
  geom_histogram(alpha=0.9, position = "stack") + xlab(xlab)+ theme(legend.title = element_blank()) +
  scale_x_continuous(breaks = seq(-10, 10, by = 1))+scale_fill_manual(values=palette)+
  scale_color_manual(values=palette)
  print(plot)
  rm(plot)
}
```

Output of linear regression applied on ‘game_result’ ~ ‘rating_difference’ relation:

```
Call:
lm(formula = actual_dataset[[y]] ~ actual_dataset[[x]])

Residuals:
    Min       1Q   Median       3Q      Max
-0.207007 -0.016094 -0.000044  0.014369  0.200301

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.822e-03  2.335e-04   7.805 6.36e-15 ***
actual_dataset[[x]] 4.082e-05  9.996e-07  40.841 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.02734 on 13723 degrees of freedom
Multiple R-squared:  0.1084,    Adjusted R-squared:  0.1083
F-statistic: 1668 on 1 and 13723 DF,  p-value: < 2.2e-16
```

Conclusions

In this work I tried to answer to some interesting questions everyone that plays at chess could make, especially for those who plays on Lichess platform.

Some of these answers can be summarized:

- Users rating distributions are simil-Gaussian but even after a simple attempt of data cleaning this distribution is characterized by an higher kurtosis index (>3) when the Shapiro-Wilk normality test fails.
- It's preferred to be a white player because statistically white beats black more than 51% of the time.
- There is a significant linear correlation between white and black rating with an R^2 value of 0.455 .
- In a match in which the rating gap between two players is greater than 100 everyone is wrong less than 26.9% in betting against the underdog.

I found out a correlation between rating difference and the speed of the game ('game_result') but an univariate relation doesn't seem sufficient. Still, the hypothesis of independence between the statistical units has been taken for granted but it is possible that time series consisting of successions of matches from the same users may have interfered with the analysis.

For future purposes I'll investigate this better together with every information i could retrieve from the time controls and from the actual moves in the game. Moreover, I'll study more efficient ways to generate appropriate criteria for the data cleaning.