Script

Script:

(intro.ipynb)

Hello everyone? For the end of the phase project, I will be analyzing the Magnus Carlsen Lichess Games Dataset that I found on Kaggle. Lichess is a website where people can play chess online. Before I start getting into the cleaning and the analysis, I want to give some background information about chess.

Chess is an abstract strategy board game with two players. One controls the white pieces, the other controls the black pieces. Throughout this presentation, I'll be referring to the two players as 'white' and 'black'. The current form of chess appeared in the second half of the 15th century and it is currently one of the most popular games in the world. It is played on a square chessboard with 64 squares arranged in an 8 by 8 grid. Each square has its own algebraic chess notation.

(Might cut this out to save time) (Highlight squares with mouse) For example, this square is e4 and this square is c5. (Highlight pieces with mouse) This is what the starting position looks like. Each player starts with 16 pieces, which include 8 pawns, 1 for each column, 2 rooks that start on the 'a' and 'h' columns, 2 knights that start on 'b' and 'g', 2 bishops that start on 'c' and 'f', a queen that starts on 'd' and the king which starts on 'e'.

Magnus Carlsen has been the highest rated chess player since July 2011. I will be analyzing his games from Lichess.com. There are also other terms I'll be using throughout this presentation. One of them is the elo rating system. It's used to calculate the relative skill level of players or teams in board games like chess, esports like League of Legends, and sports like table tennis.

Elo was created in 1960. The international chess federation uses the elo system. In my code, I use elo and rating interchangeably even though elo is a form of rating system.

Time control is the time given to each player. If a player runs out of time, that player loses. Time control is usually written in the form 'x+y' where x is the amount of minutes and y is the amount of seconds given to a player after each move they make. Magnus Carlsen usually plays 1 minute games with no extra time.

Fractional score is the main metric I'll be exploring in my analysis. It represents the amount of points a player has scored out of the total amount of games they play. Wins are 1 point and draws are half a point. Draws are pretty common in chess, especially between higher rated players that have at least one hour to make their moves. I'll be using the term fractional score a lot

(Cleaning.ipynb)

The main question I wanted to explore with this Magnus Carlsen Lichess Games dataset is which variables affect his fractional score. These variables include the color of his pieces, the time control, the first move he makes, and the rating of his opponents.

Script

Before my analysis, I cleaned the original dataset with this question in mind. (Scroll cg.head) This is what carlsen_games.csv looks like originally. It has 13442 rows, one for each game, and 31 columns. I use the variable name cg, short for clean games, to represent the clean version of games.csv. I end up dropping most of these columns because they were not useful in my analysis. The columns that are remaining include: game_id, white, black, result, white_elo, black_elo, time_control, magnus_color, magnus_result, datetime. I slightly rearranged the columns so that white and white_elo are together and black and black_elo are together.

Next thing I did was change almost all of the lichess usernames to the player's real name. I do this using the user_names.csv. I mainly did this to easily identify which player was Magnus in each game. The name Magnus Carlsen appears in every row. (usernames) This is what the usernames.csv looks like, 543 rows and 2 columns.

Each move that Magnus has played on lichess is stored in another csv called Carlsen_games_moves. It has over 1.1 million rows and 64 columns. (Scroll to the 4 column csv) I only need 4 of these columns. The game_id column which allows me to merge to this csv with the games_csv. The move_number_pair column which shows the nth move for each player. For example, Magnus's 2nd move in this game was e6. The player column which has the real name of the players after I used .replace with the dictionary. And the notation column which has the notation of all the moves. The notation shows which piece, shown by a capital letter, goes to which square. If there's no capital letter, that means it was a pawn move.

(Highlight .loc text) After that, I find all of Magnus's first moves using .loc. Then, I merge the first moves dataframe with the cg dataframe on the game_id column. Now each row has the first move made by Magnus. Then, I sorted cg chronologically by the datetime column.

(Highlight first row) The magnus_color and magnus_result columns from the original csv had incorrect values so I fixed them next. To fix magnus_color, I created a function that returns white when Magnus Carlsen's name is in the white column and returns black otherwise. I used .apply to apply that function to the magnus_color column. Then, I did something similar to fix magnus_result. I could figure out whether magnus won, drew, or lost based on the result column and which color Magnus was playing.

Then I changed the time_control column to show the time in minutes+seconds because that's the convention. Finally, I drop two unneeded columns from the moves and cg dataframe merge from earlier. '%store cg' is used to store the dataframe to be used in another notebook

Next I create another dataframe where Magnus has his own column and the opponent has their own column. I mainly do this to keep track of the opponent's rating. These are the functions I use to do this. For Magnus's elo, I return white_elo if Magnus is white, otherwise return black_elo. Finally, I create a dataframe called mg which is very similar to cg except it has the columns magnus_elo and opponent_elo instead of white, white_elo, black, black_elo.

Script

(Fractional Score Analysis)

Now I'm done with the cleaning, and I can move onto the analysis. I use '%store -r' to basically import my dataframes from my cleaning notebook. I have a function that calculates fractional score from a series like this (highlight value count example). The method value_counts() show number of wins, losses, and draws. The other 2 functions are just used to shorten my code since I use magnus_result.value_counts() a lot. So, initially I created multiple bar charts showing the fractional score of different variables. (opp rating chart) This bar chart shows the fractional score based on the opponent's rating. The bar chart makes it clear that fractional score is negatively correlated with opponent rating. This makes sense since most people would have a lower fractional score against opponents with high ratings. But, since the correlation between fractional score and opponent rating is so strong, it's hard to draw conclusions from the bar charts below. (Scroll to year chart) For example, Magnus's fractional score was a lot lower in 2020 than in 2019. I cannot tell whether this is because Magnus underperformed in 2020 or he faced a lot of higher rated opponents in 2020. ~~(Scroll to bottom) I created a scatterplot and calculated the correlation coefficient between fractional score and opponent rating to make sure my assumption was correct. The x value is average opponent rating. This specifically means the average rating of all opponents in the games that meet the given condition. The correlation coefficient between those two variables was -0.95, meaning they have a very strong negative correlation.~~

(To save time I'm skipping the 'elo analysis.ipynb' file)

(regression analysis.ipynb)

To deal with this issue, I decided to make a scatterplot where I plot the average opponent rating and fractional score for each variable that I am analyzing. The average opponent rating is specifically the average rating of all opponents in the games that meet a given condition (highlight a condition). So I plotted the years 2018 through 2021, black, white, opponent rating ranges, time control, 85% of the games in the dataset were 1 minute games so I split the games up like this, the 4 most common first moves he has played as black, which is d5, Nf6, c5, and g6, and the 4 most common first moves he has played as white, which is e4, d4, c4, Nf3. (Scroll back up and show chessboard(Ctrl+K Ctrl+C) This is d5 Nf6... etc. The first move for black puts a piece in row 5 and 6. The first move for white puts a piece in row 3 and 4. I separated the first moves for white and black because white moves first and has a slight advantage overall.

(Scroll back down to plot equation) I used np.ployfit to create line of best fit, annotated every point and here is what the scatterplot looks like (zoom out if necessary).

Then I use seaborn to plot the residuals (show residual plot). A positive residual means that the fractional score is better than expected, and a negative residual means the fractional score is

worse than expected. With this plot, I can make various conclusions. G6 is Magnus's best first move as black, d4 is Magnus's best first move as white. 2019 was his best performing year. …