Final Project EDA

AUTHOR
Matheo Dwyer and Sriram Soundar

Exploring the Data

What is your outcome variable(s)? How well does it measure the outcome you are interested? How does it relate to your expectations?

Our main outcome variable is the *winner* of the chess game—either "white", "black", or "draw". This is a pretty direct way to measure success in a game, since it tells us who actually won. It captures the outcome we care about clearly and is simple to analyze. It also fits our expectations well; for example, we expected white to have a slight advantage because they go first, and that's exactly what we see—white wins more of the decisive games (52.34% vs. 47.66%).

What are your key explanatory variables?

Some of our key explanatory variables are the *Elo ratings* of both players and the *difference in rating* between them. These help us understand how player skill affects game outcomes. We also look at who had the higher rating and whether they won or lost, which is captured in the variable `higher_winner`. Additionally, we explored *number of turns* as a variable influenced by Elo difference, and *opening_name* as a possible influencer in strategy or outcome. These variables give us insight into what might give one player an advantage and how skill gaps affect game dynamics.

Data Wrangling and Transformation

What data cleaning did you have to do?

We actually didn't have to do much data cleaning at all. The dataset came in a really clean format with no missing values, which made it easy to jump right into analysis. Everything was already labeled well, and all the variables we needed were available without any obvious errors or inconsistencies.

How did you wrangle the data?

We did a lot of data wrangling throughout the project. We used filtering to focus on just decisive games (excluding draws), grouped and counted wins by color, created subsets for different Elo rating difference thresholds, and used `mutate()` to generate new variables for analysis. The wrangling let us break the data down in multiple ways—by rating, outcome, and opening type—which helped us explore different patterns and questions.

Are you deciding to exclude any observations? If so, why?

We didn't exclude many observations overall, but we did make some decisions to narrow our focus in specific cases. For example, when looking at win percentages, we excluded draws to only focus on decisive games. Similarly, when analyzing rating differences, we filtered by thresholds like ≤100, ≤250, and ≤500 Elo points to see how outcomes shifted across different skill gaps. These exclusions weren't about cleaning bad data—they were more about helping us zoom in on specific comparisons.

Did you have to create any new variables from existing variables? If so, how and why?

Yes, we created a few new variables to help with analysis. One key one was `rating_diff`, which we made using the absolute difference between white and black player ratings. This helped us explore how skill differences affected outcomes and game length. Another one was `higher_winner`, where we labeled whether the higher-rated player or lower-rated player won. Creating these variables made it easier to summarize trends and visualize how rating impacts performance.

Codebook

Variables:

- **Game ID**: Unique identifier for each game.
- **Rated (T/F)**: Whether the game was rated (True) or unrated (False).
- **Start Time**: Timestamp of when the game started.
- **End Time**: Timestamp of when the game ended.
- **Number of Turns**: Total number of moves made in the game.
- **Game Status**: The outcome of the game (e.g., "checkmate", "resign", "draw").
- **Winner**: The winner of the game ("white", "black", or "draw").
- **Time Increment**: The time increment (in seconds) added after each move.
- **White Player ID**: Unique identifier for the white player.
- **White Player Rating**: The rating of the white player at the time of the game.
- **Black Player ID**: Unique identifier for the black player.
- **Black Player Rating**: The rating of the black player at the time of the game.
- **All Moves in Standard Chess Notation**: The sequence of moves made in the game, recorded in standard algebraic notation.
- **Opening Eco**: The ECO code (Encyclopedia of Chess Openings) for the opening used in the game.
- **Opening Name**: The name of the opening used in the game.
- **Opening Ply**: The number of moves in the opening phase of the game.

Data Visualizations

```r
library(ggplot2)
library(dplyr)
```

```

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```r
df <- read.csv("games.csv")

# count how many times each outcome happened (white win, black win, draw)
outcome_counts <- table(df$winner)
print(outcome_counts)
```

```

black  draw white 
 9107   950 10001 
```

```r
# figuring out the percentage of wins for white and black, only looking at games that did
total_decisive_games <- sum(df$winner %in% c("white", "black"))
white_win_percent <- sum(df$winner == "white") / total_decisive_games * 100
black_win_percent <- sum(df$winner == "black") / total_decisive_games * 100

# draw percentage out of all games
draw_percent <- sum(df$winner == "draw") / nrow(df) * 100

cat(sprintf("White wins %.2f%% of decisive games.\n", white_win_percent))
```

```
White wins 52.34% of decisive games.
```

```r
cat(sprintf("Black wins %.2f%% of decisive games.\n", black_win_percent))
```

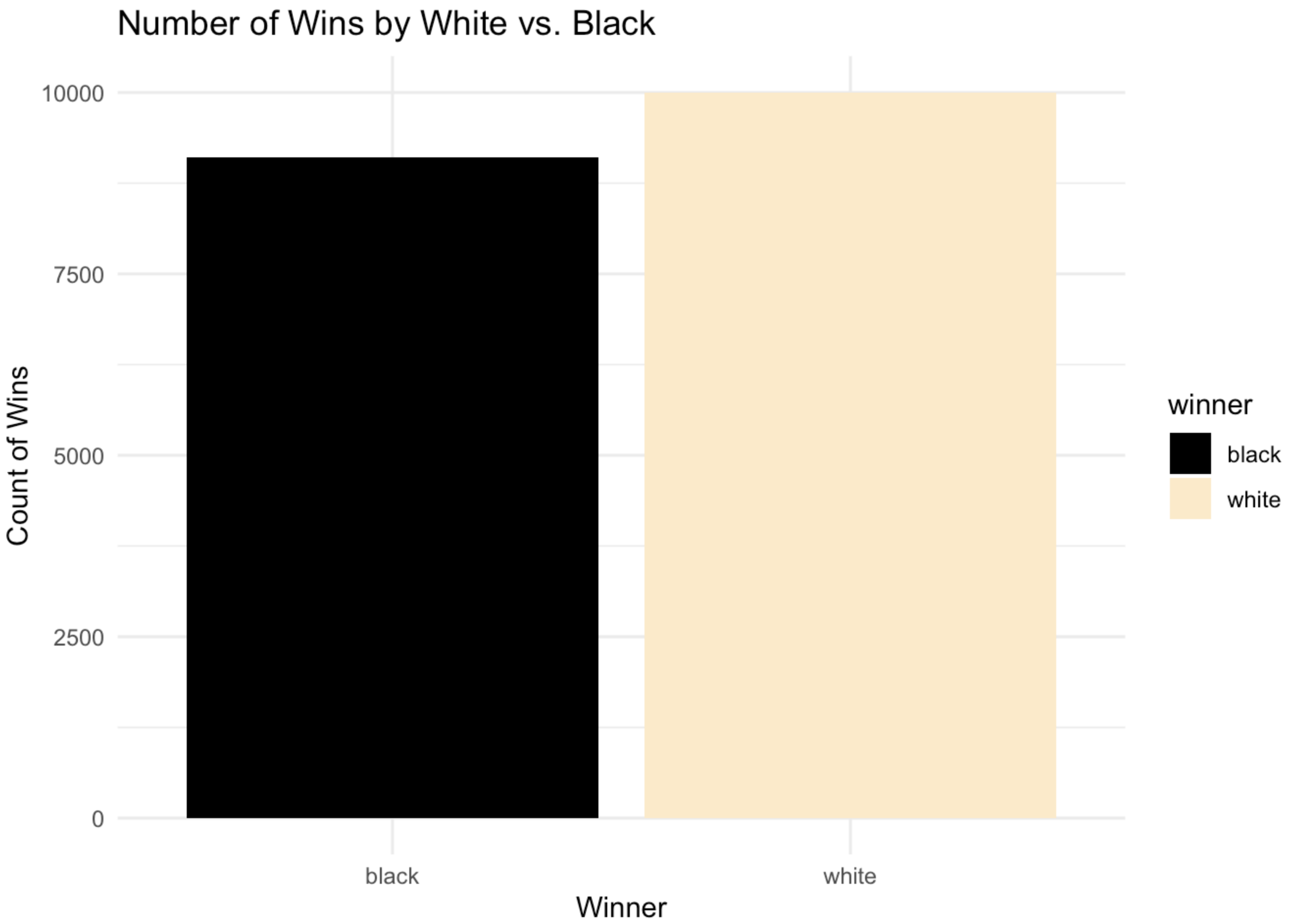
```
Black wins 47.66% of decisive games.
```

```r
cat(sprintf("Draws occur %.2f%% of all games.\n", draw_percent))
```

```
Draws occur 4.74% of all games.
```

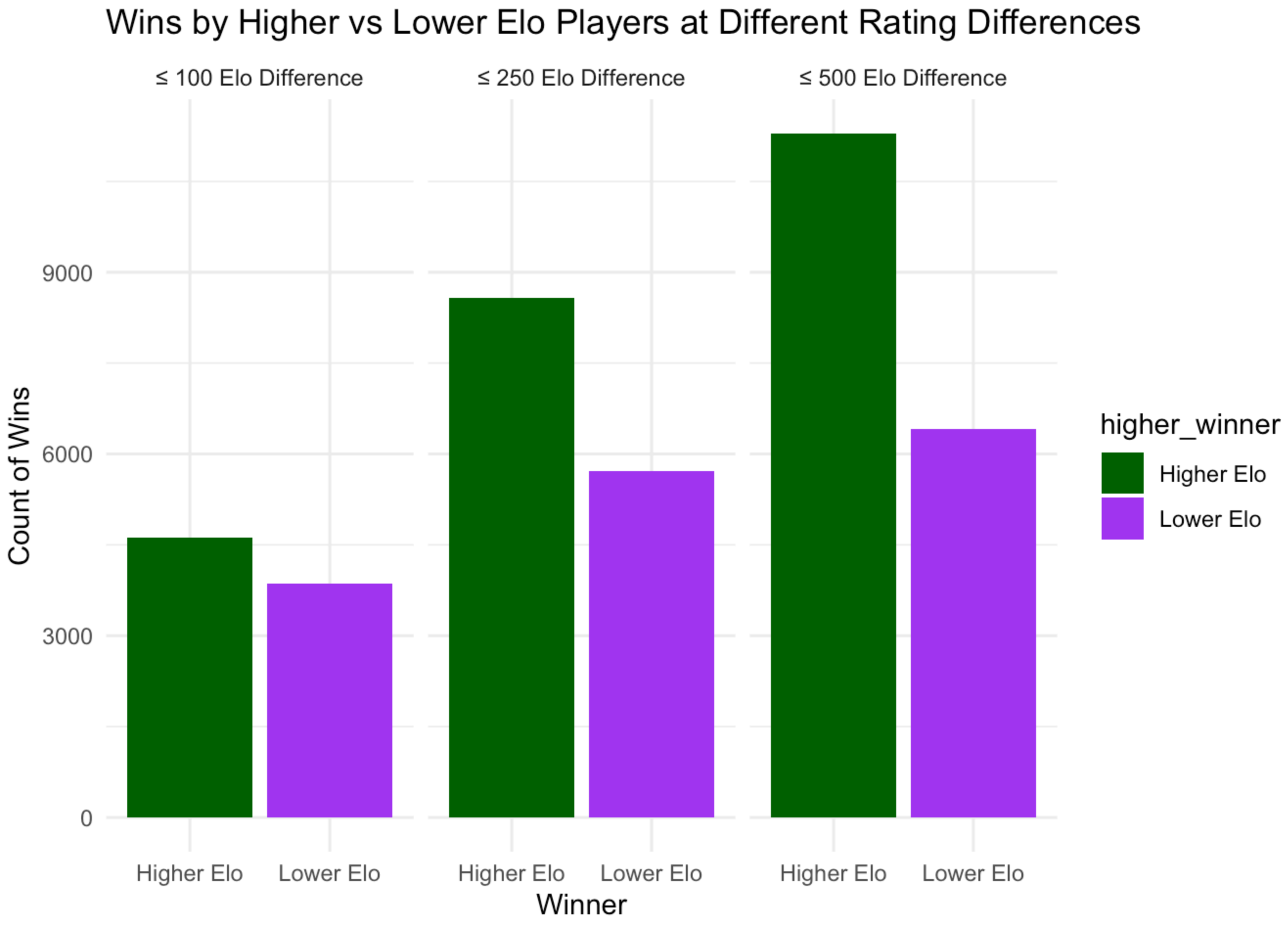
```r
win_counts <- df %>%
  filter(winner %in% c("white", "black")) %>%
  count(winner)
```

```
# bar graph showing how often white and black win
ggplot(win_counts, aes(x = winner, y = n, fill = winner)) +
  geom_bar(stat = "identity") +
  labs(title = "Number of Wins by White vs. Black",
       x = "Winner",
       y = "Count of Wins") +
  theme_minimal() +
  scale_fill_manual(values = c("white" = "blanchedalmond", "black" = "black"))
```



We can see above that white wins more games, with 52.34% vs 47.66%. The term “decisive games” is a reference to all games that ended in a win or a loss exclusively

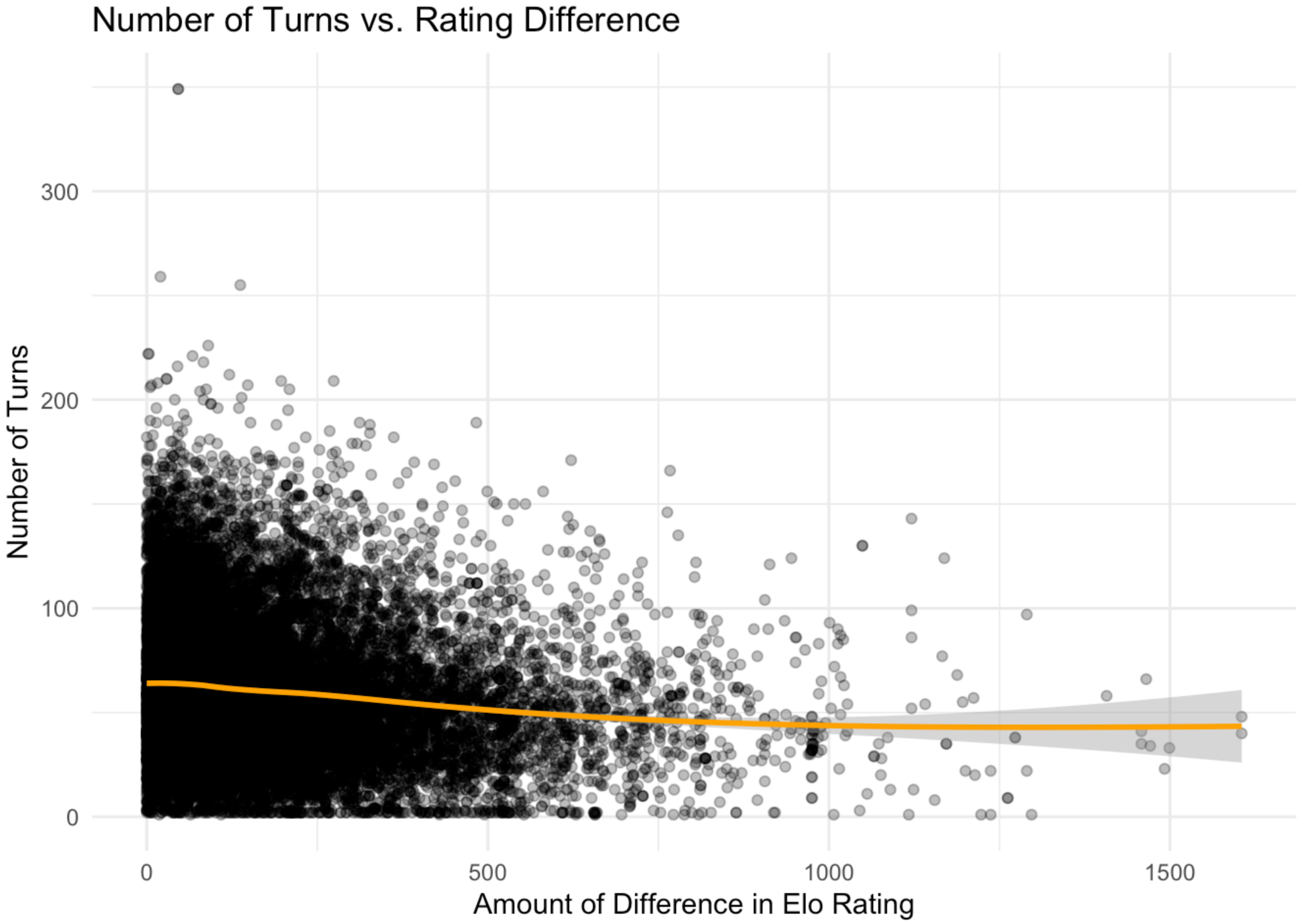
```
library(ggplot2)
library(dplyr)

df <- read.csv("games.csv")

# creating a new variable: rating_diff shows how far apart the two players are in skill
# higher_winner tells us if the higher rated player won the game
df <- df %>%
  mutate(rating_diff = abs(white_rating - black_rating),
         higher_winner = case_when(
           white_rating > black_rating & winner == "white" ~ "Higher Elo",
           black_rating > white_rating & winner == "black" ~ "Higher Elo",
           white_rating > black_rating & winner == "black" ~ "Lower Elo",
           black_rating > white_rating & winner == "white" ~ "Lower Elo",
           TRUE ~ NA_character_ # we leave out draws or equal ratings
         ))

# This function helps us count how often the higher-rated player wins at different rating
get_win_counts <- function(threshold) {
  df %>%
    filter(rating_diff <= threshold, !is.na(higher_winner)) %>%
    count(higher_winner) %>%
    mutate(threshold = paste("≤", threshold, "Elo Difference"))
}

# running the function for different thresholds of Elo difference
win_counts_100 <- get_win_counts(100)
win_counts_250 <- get_win_counts(250)
win_counts_500 <- get_win_counts(500)

# combining all the results into one table so we can plot it
win_counts <- bind_rows(win_counts_100, win_counts_250, win_counts_500)

# bar chart showing who tends to win as rating differences increase
ggplot(win_counts, aes(x = higher_winner, y = n, fill = higher_winner)) +
  geom_bar(stat = "identity", position = "dodge") +
  facet_wrap(~threshold) +  # separate plots for each threshold
  labs(title = "Wins by Higher vs Lower Elo Players at Different Rating Differences",
       x = "Winner",
       y = "Count of Wins") +
  theme_minimal() +
  scale_fill_manual(values = c("Higher Elo" = "darkgreen", "Lower Elo" = "purple"))
```



“Elo” is a numerical score measure someone’s rating and how good they are. Here we can see that the closer the elo’s and rankings are, the more the odds of winning are even. However when we start looking at larger elo gaps, like 250 and 500, higher elo can win up to twice as much as the lower ranked one.

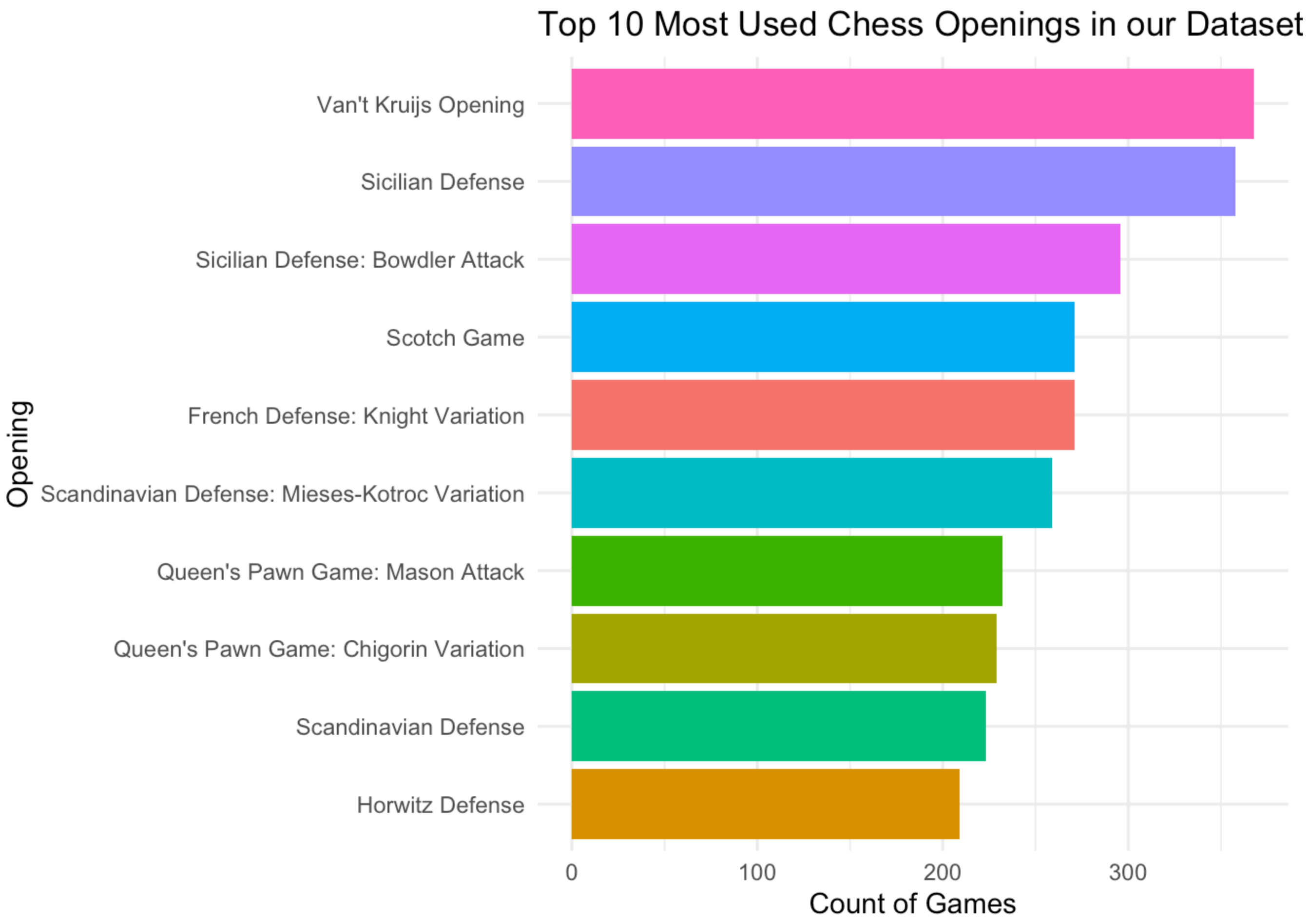
```
# plots the rating difference vs. how long the game lasted (in turns)
ggplot(df, aes(x = abs(white_rating - black_rating), y = turns)) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = "loess", color = "orange") +
  labs(title = "Number of Turns vs. Rating Difference",
       x = "Amount of Difference in Elo Rating",
       y = "Number of Turns") +
  theme_minimal()
```

```
`geom_smooth()` using formula = 'y ~ x'
```



This in our opinion is a very interesting graph, that captures game length in turns relative to difference in elo rating. A conclusion we can draw is that the larger the gap in elo ranking, the faster the games tend to be.

```
# counts how many times each opening shows up, then grabs the top 10 most common ones and
df %>%
  count(opening_name) %>%
  top_n(10, n) %>%
  ggplot(aes(x = reorder(opening_name, n), y = n, fill = opening_name)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(title = "Top 10 Most Used Chess Openings in our Dataset",
       x = "Opening",
       y = "Count of Games") +
  theme_minimal() +
  guides(fill = "none")
```



Above we have the top ten most used openings. We can see that Van’t Kruji and Sicilian are almost tied for first in sheer game count.