Hans Niemann: Did He Cheat?

An Indepth Look at Chess Performance and Statistical Methods to Determine Cheating

Abstract

In high level chess play, grandmasters must study their opponents, understand complex theories and moves, and be creative in their decision-making process. However, all these aspects rely on the underlying principle of fair and unaided game without cheating. Hans Niemann has been accused of violating these principles in order to rise in the rankings and beat the current World Champion Magnus Carlsen. In this paper, we will demonstrate that accuracy in games is positively correlated to consistency and that as skill increases accuracy increases. Niemann has statistically higher yearly Elo growth, but his growth per game is similar to other professional players. Over the time period examined, World Champion Magnus Carlsen performed worse than six of the top Grand Masters, while Niemann played less accurately than the majority of the other Grand Masters but played with the same consistency. These findings suggest that there is inconclusive proof that Hans Niemann is cheating and that performance is a complex subject that further research needs to explore.

Introduction

In August 2022, Hans Niemann, an up and coming chess prodigy, defeated Magnus Carlsen, the current reigning chess champion, during the FTX Crypto Cup. Shortly thereafter, in September 2022, Niemann once again defeated Carlsen; however, this time, Niemann pulled it off while playing black in classical chess. This became a monumental occurrence. Niemann became the first person in over two years to defeat Carlsen as black in over the board classical chess. The chess champion didn't take this lightly and abruptly dropped out of the Sinquefield Cup after taking an unexpected loss to up Niemann. This withdrawal from the Sinquefield Cup stirred controversy online and a chess cheating controversy swept the chess community. The cheating allegation gained more traction as Carlsen tweeted a cryptic message of football manager Mourinho saying, "I prefer really not to speak. If I speak, I am in big trouble."

Chess cheating has always been a concern in online formats, where players play from their homes on their own computers. However, over the board cheating, though much more rare, has also been caught on several occasions in the past. For example, people have cheated in the past with vibrating Bluetooth devices in their hats, shoes, and wrist watches. As such, we wanted to investigate if the current Magnus Carlsen - Hans Niemann chess cheating allegations held any weight by looking at measurements for a chess player's accuracy and consistency.

Data

Data Collection

Data was collected based on the top 50 chess grandmasters (as of 11/14/2022) according to FIDE, the International Chess Federation (Figure 1).

Rank	Name	Title	Country	Rating	Games	B-Year
1	Carlsen, Magnus	g	NOR	2859	0	1990
2	Ding, Liren	g	CHN	2811	0	1992
3	Nepomniachtchi, Ian	g	RUS	2793	0	1990
4	Firouzja, Alireza	g	FRA	2785	0	2003
5	Nakamura, Hikaru	g	USA	2768	0	1987
6	Caruana, Fabiano	g	USA	2766	0	1992
7	Giri, Anish	g	NED	2764	0	1994
8	So, Wesley	g	USA	2760	0	1993
9	Anand, Viswanathan	g	IND	2754	0	1969
10	Karjakin, Sergey	g	RUS	2747	0	1990
11	Radjabov, Teimour	g	AZE	2747	0	1987
12	Grischuk, Alexander	g	RUS	2745	0	1983
13	Dominguez Perez, Leinier	g	USA	2743	0	1983
14	Mamedyarov, Shakhriyar	g	AZE	2740	2	1985
15	Rapport, Richard	g	ROU	2740	0	1996
16	Vachier-Lagrave, Maxime	g	FRA	2737	3	1990
17	Aronian, Levon	g	USA	2735	0	1982
18	Vidit, Santosh Gujrathi	g	IND	2730	9	1994
19	Duda, Jan-Krzysztof	g	POL	2729	2	1998
20	Andreikin, Dmitry	g	FID	2729	0	1990
21	Yu, Yangyi	g	CHN	2728	9	1994
22	Le, Quang Liem	g	VIE	2728	0	1991
23	Topalov, Veselin	g	BUL	2728	0	1975
24	Gukesh D	g	IND	2725	0	2006
25	Vitiugov, Nikita	g	FID	2723	2	1987
26	Erigaisi Arjun	g	IND	2722	3	2003
27	Wang, Hao	g	CHN	2722	0	1989
28	Wei, Yi	g	CHN	2722	0	1999
29	Maghsoodloo, Parham	g	IRI	2719	7	2000
30	Vallejo Pons, Francisco	g	ESP	2716	0	1982
31	Abdusattorov, Nodirbek	g	UZB	2713	0	2004
32	Sjugirov, Sanan	g	RUS	2712	0	1993
33	Shankland, Sam	g	USA	2710	0	1991
34	Dubov, Daniil	g	RUS	2708	0	1996
35	Eljanov, Pavel	g	UKR	2706	0	1983
36	Harikrishna, Pentala	g	IND	2705	3	1986
37	Robson, Ray	g	USA	2702	0	1994
38	Artemiev, Vladislav	g	RUS	2701	0	1998
39	Deac, Bogdan-Daniel	g	ROU	2700	0	2001
40	Sargissian, Gabriel	g	ARM	2699	0	1983
41	Niemann, Hans Moke	g	USA	2698	9	2003
42	Bu, Xiangzhi	g	CHN	2698	0	1985
43	Keymer, Vincent	g	GER	2696	2	2004
44	Tomashevsky, Evgeny	g	RUS	2694	0	1987
45	Xiong, Jeffery	g	USA	2692	0	2000
46	Van Foreest, Jorden	g	NED	2690	0	1999
47	Cheparinov, Ivan	g	BUL	2688	6	1986
48	Adams, Michael	g	ENG	2688	2	1971
49	Fedoseev, Vladimir	g	FID	2688	0	1995
50	Sevian, Samuel	g	USA	2687	0	2000

Figure 1. **Rankings of the Top 50 Players**. Data was collected for the top 50 players in the world with a range of Elo from 2687-2859. Rank was obtained from the International Chess Federation.

Based on the top 50 chess grandmasters, each respective player's game data was obtained from PGN Mentor, a chess database that contains the Portable Game Notation (PGN) — the most popular standard for representing chess games — for each player. Among the top 50 chess grandmasters, we were able to obtain data for 34 players.

Data Cleaning

The downloaded PGN files have a specific format that isn't easily parseable with standard Python or R. The PGN files for each player contained details about games with the name of the event, the location of the event, the date, players involved and their Elos, the match's result, and every move made in the game. For certain players, their PGN files had games ranging back as far as the late 90s (Figure 2).

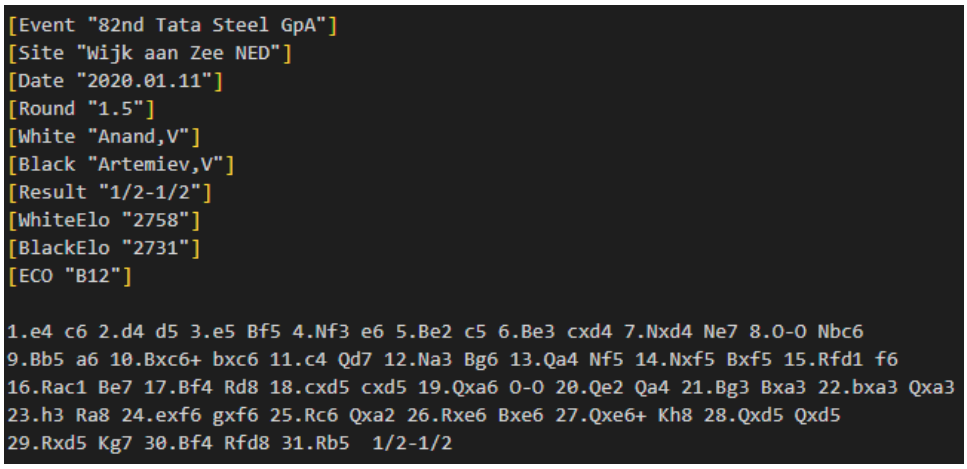


Figure 2. **PGN Format for Chess Games**. PGN format contains the information concerning all aspects of a game from the event and where it was played to the moves played. PGN format is a type of text format for chess games.

In order to convert the PGN files into a usable format, we used the chessdata library in Python to convert every player's PGNs to a dataframe and subsequently a csv containing all of the metadata. Since we are specifically interested in over the board cheating, we removed all online games that were played, and to prepare for the next step (which turned out to be quite significantly computationally costly and time consuming), we decided to keep only the past two years worth of games (Figure 3).

```
for player in players:
    pgn = open(pgns/f"{player}.pgn")
    df = pgn2df(pgn)
    df = df[df['Date'] > '2020.01.01'] ## only keep games after 2020
    df = df[df['Site'].str.contains(".com")==False] ## only keep over-the-board chess (i.e. remove online games)
    df = df[df['Site'].str.contains(".org")==False] ## only keep over-the-board chess (i.e. remove online games)
    df = df[df['Event'].str.contains("Online")==False] ## only keep over-the-board chess (i.e. remove online games)
    print(player)
    print(df.shape)
    df.to_csv(path/f"output/metadata/{player}.csv")
```

Figure 3. **For Loop to Filter for Desired Games**. A for loop was employed to loop through all players. It then removed all online games.

Next, to measure the accuracy and consistency of each player, we relied on a measure of Centipawn Loss (CP). With the advancement of machine learning and neural networks in chess, chess computer engines, such as Stockfish, are able to play the game with more accuracy than the strongest of human chess grandmasters. A chess engine is able to evaluate certain moves in chess given a position by providing every possible move with a Centipawn value, representing 1/100th of a pawn (or point) (Figure 4).

```
for player in players:
    print(player)
    print(datetime.now().strftime('%Y-%m-%d %H:%M:%S'))
    engine = chess.engine.SimpleEngine.popen_uci(stockfish)
    pgn = open(pgns/f"{player}.pgn")
    evals = evaluate_pgn(pgn, engine, limit=chess.engine.Limit(depth=15))
    evals.to_csv(path/f"output/centipawns/{player}.csv")
```

Figure 4. **Employing the Chess Engine**. Chess Engine Stockfish was used to compute the centipawn for each move in a chess game. The for loop applies the centipawn calculation from stockfish onto each player and saves it as csv file.

By using an engine's Centipawn value for moves as the baseline, we are able to calculate the CP (or the amount a player's move deviates from the most optimal suggested move) of every move of every game for every player (Figure 5).

```python
def get_cp_loss(player):
    cp = pd.read_csv(f"output/centipawns/{player}.csv", index_col=0)
    cp = cp.clip(-1000, 1000)
    diffs = cp.diff(axis=1)
    md = pd.read_csv(f"output/metadata/{player}.csv", index_col=0)
    colors = [get_color(player, row) for _, row in md.iterrows()]
    all_diffs = []
    for i in range(cp.shape[0]):
        start = 0 if colors[i] == "Black" else 1
        player_diffs = diffs.iloc[i, :].dropna()[start::2].values
        if colors[i] == "White":
            player_diffs *= -1
        player_diffs = player_diffs.clip(min=0)
        all_diffs.append(player_diffs)
    df = pd.DataFrame(all_diffs)
    df.to_csv(f"output/cp_loss/{player}.csv")
```

Figure 5. **Calculating Centipawn Loss**. Centipawn Loss was read in then loss was calculated in the function above. The output was then saved as a csv file.

About the Data

Based on the metadata gathered from parsing the PGNs as well as the CPs for every player, we created a concatenated data set for each player containing their name, age, years spent playing the game, mean CP, standard deviation in CP, and Elo (Figure 6).

```python
for player in players:
    age = pd.read_csv(f"output/age/{player}.csv", index_col=0)
    time = pd.read_csv(f"output/time/{player}.csv", index_col=0)
    means = pd.read_csv(f"output/mean_cp_loss/{player}.csv", index_col=0)
    stds = pd.read_csv(f"output/std_cp_loss/{player}.csv", index_col=0)
    elo = pd.read_csv(f"output/elo/{player}.csv", index_col=0)
    wl = pd.read_csv(f"output/winloss/{player}.csv", index_col=0)
    whitewl = pd.read_csv(f"output/whitewinloss/{player}.csv", index_col=0)

    df = pd.concat([age, time, means, stds, elo, wl, whitewl], axis=1)
    df['Name'] = player
    df = df.dropna()
    df = df.reset_index(drop=True)

    df.to_csv(f"./data/{player}.csv")
```

Figure 6. **Combining the Data**. After collecting the data, it was then combined to create a single data frame for each player. The data frame was then saved as a csv file.

Exploratory Data Analysis

After the data cleaning and collection, we visualized the information in order to better understand the data. We first plotted a boxplot for mean CP, standard deviation CP, age, player and opponent Elo for all players.

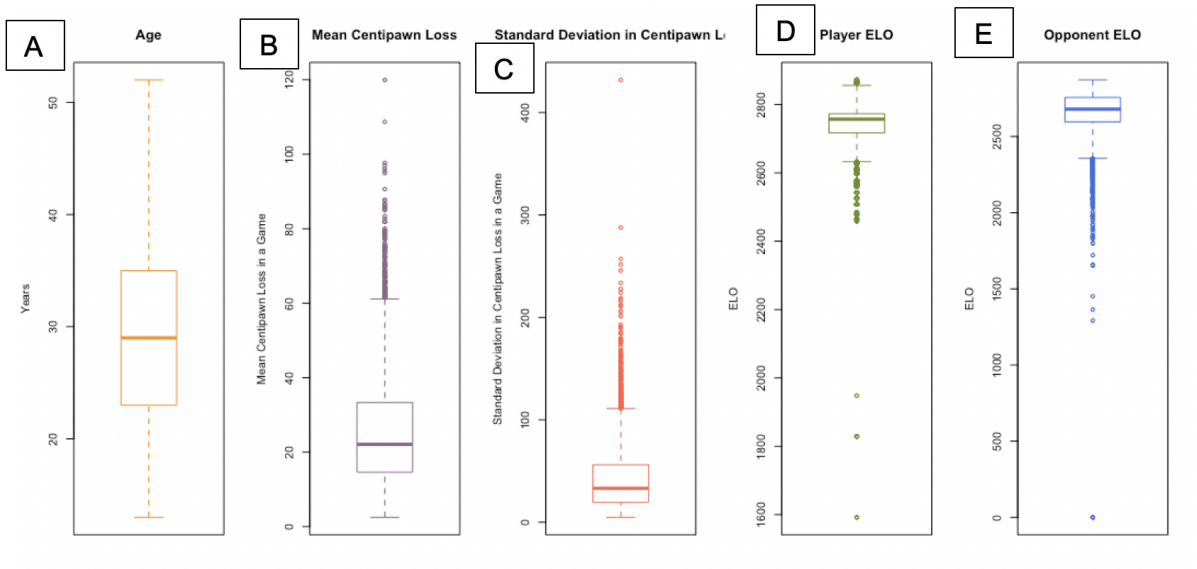


Figure 7. **Histogram of Player Data**. Boxplots were created for mean CP, standard deviation CP, Age, player and opponent Elo across all payers.

The boxplots show that the majority of players had an age between 23 and 35 years of age with the maximum age being 53, demonstrating that the majority of the top-level players were younger than expected (Figure 7a). Additionally, mean CP and standard deviation CP are more right skewed which is expected due to the high level of play observed at the grandmaster skill level (Figure 7b-c). Player and opponent Elo were very similar with more variation in opponent Elo (Figure 7d-e).

Next, we examined performance over time for the key players Magnus Carlsen and Hans Niemann compared to other players. We first plotted mean CP per game and saw that players mean CP is not consistent but increases and decreases between games. Additionally, it appears that Hans Niemann has a similar performance to World Champion Magnus Carlson in the number and degree of change between peaks (Figure 8).

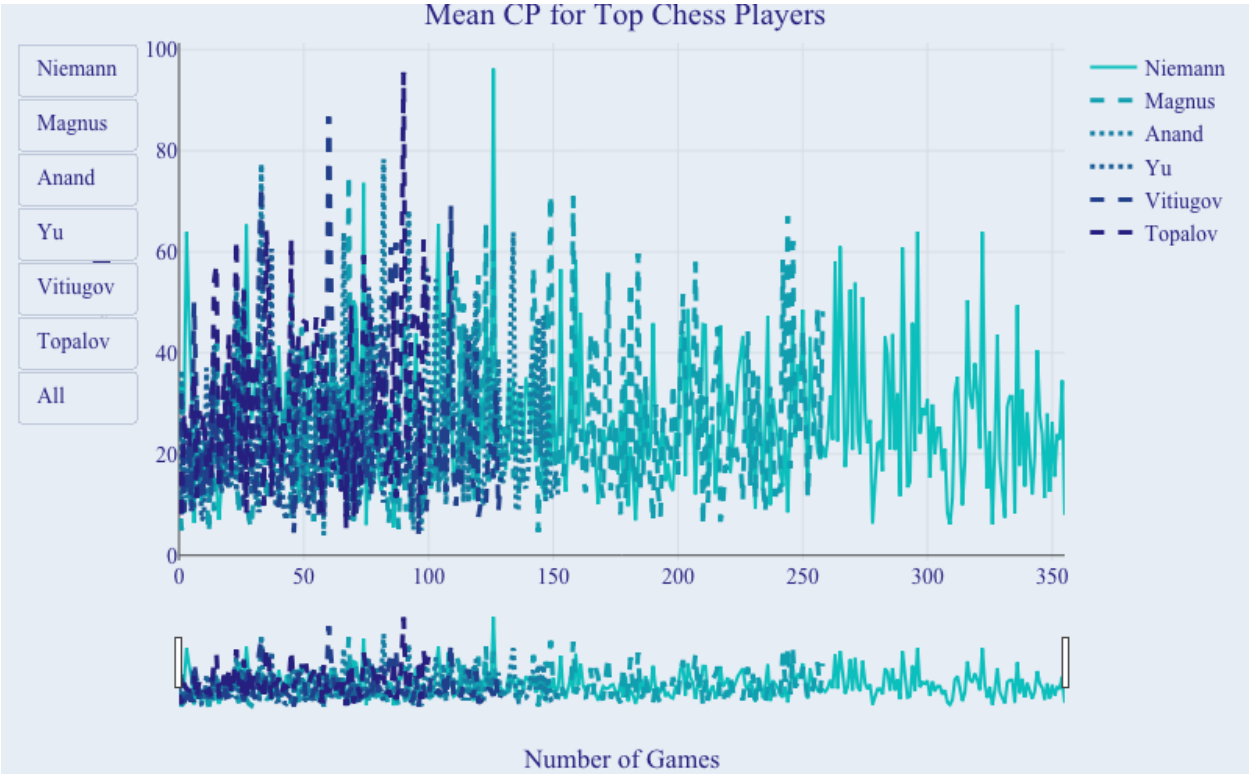


Figure 8. **Mean CP Over Time**. Mean CP was plotted for each game to determine the change over the number of games for players Niemann, Magnus, Anand, Yu, Vitiugov and Topalov.

We then plotted the standard deviation of CP per game. We saw similar trends as with mean CP but higher peaks which may be due to bad games or more blunders in games than normal (Figure 9).

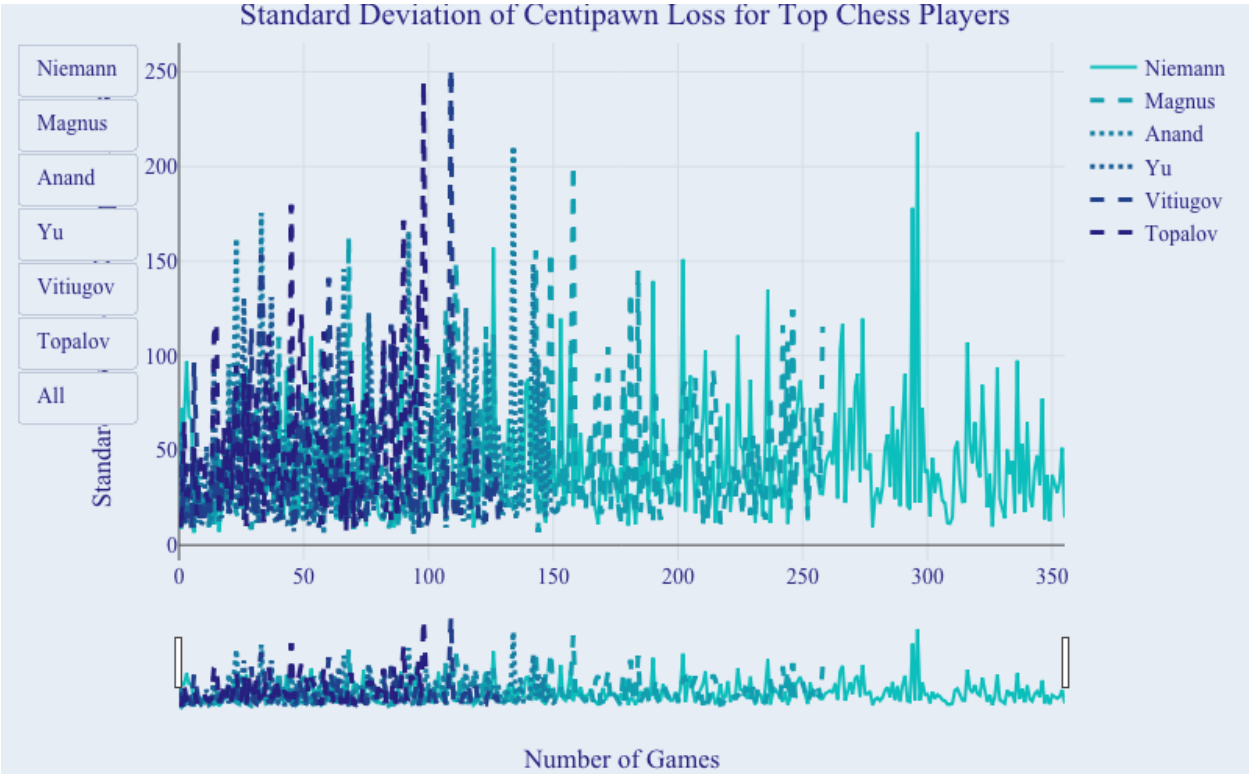


Figure 9. **Standard Deviation CP Over Time**. Standard Deviation CP was plotted for each game to determine the change over the number of games for players Niemann, Magnus, Anand, Yu, Vitiugov and Topalov.

Finally, we plotted Elo per game to determine the change of Elo overtime. We saw that Niemann continued to raise his Elo compared to other players which stayed consistent. This rapid change of Elo may be due to starting at a lower Elo or the result of cheating (Figure 10).

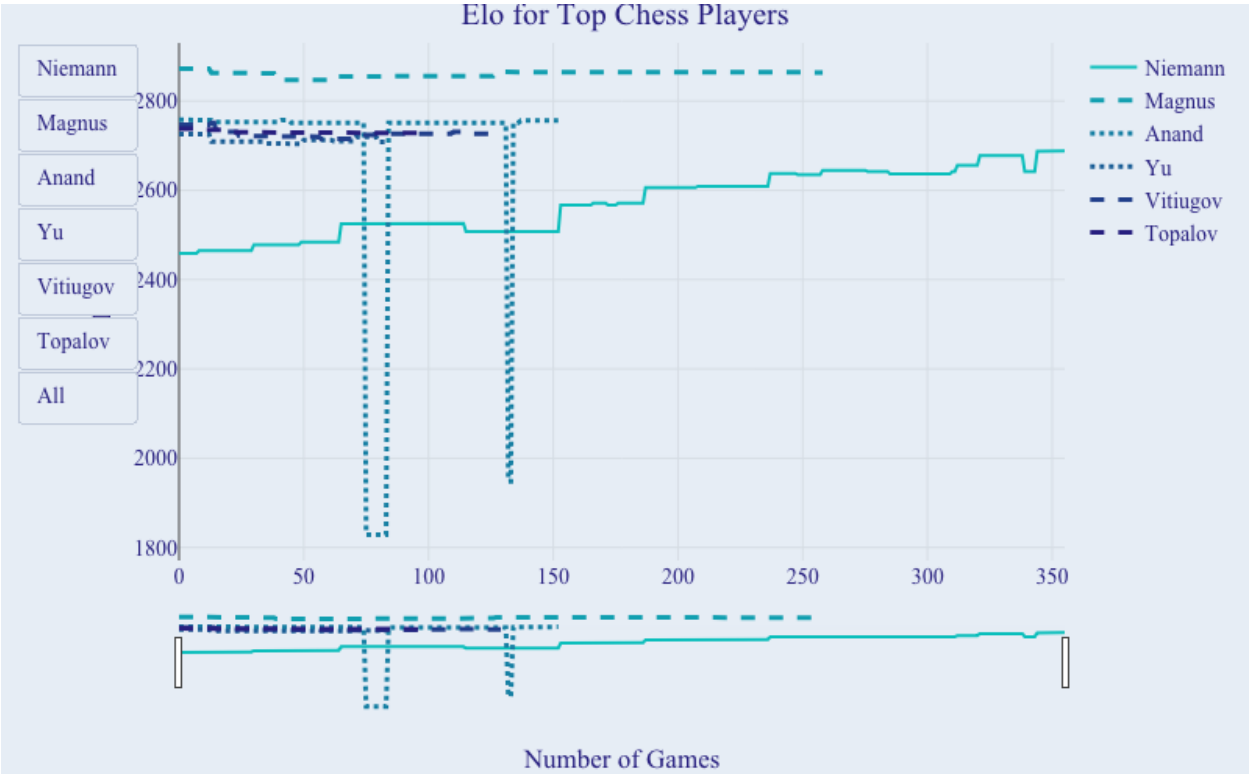


Figure 10. **Change in Elo Overtime**. Elo was plotted for each game to determine the change over the number of games for players Niemann, Magnus, Anand, Yu, Vitiugov and Topalov.

The data showed unique trends such as the right skewed for mean CP and standard deviation CP and Elo change overtime. Thus, in the next sections we will use statistical tests to determine what results in high performance and whether Hans Niemann was cheating.

Question 1

Question: *What are the best predictors for how well and consistently a chess player performs?*
Exploratory Data Analysis:

As we saw in our initial EDA, mean CP and std CP are both right skewed. We can further confirm this by looking at the respective Q-Q plots (Figure 11).

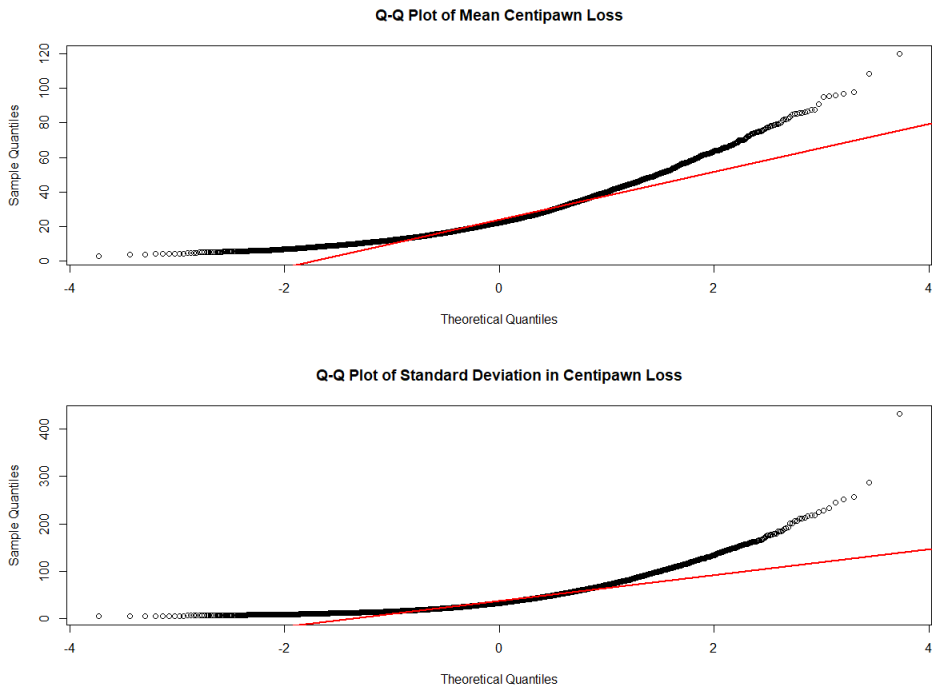


Figure 11. **Q-Q Plot of Centipawn Loss**. A Q-Q plot was used to determine whether mean CP and standard deviation CP follow a normal distribution. The results are plotted above.

The Q-Q plots are further supported by running the Shapiro Wilks Test, which results in p-values that are statistically significant for both mean CP and std CP (i.e. neither distribution is normally distributed) (Figure 12).

```
        Shapiro-Wilk normality test

data:  rand_samp_mean
W = 0.90813, p-value < 2.2e-16
```

```
        Shapiro-Wilk normality test

data:  rand_samp_std
W = 0.82968, p-value < 2.2e-16
```

Figure 12. **Shapiro Wilks Test for Centipawn Loss**. Shapiro Wilks test was applied to both to both mean CP and standard deviation CP to determine if they follow a normal distribution. The results are shown above.

In attempts to address this, we perform a log transformation, resulting in a distribution that appears more normal (Figure 13).

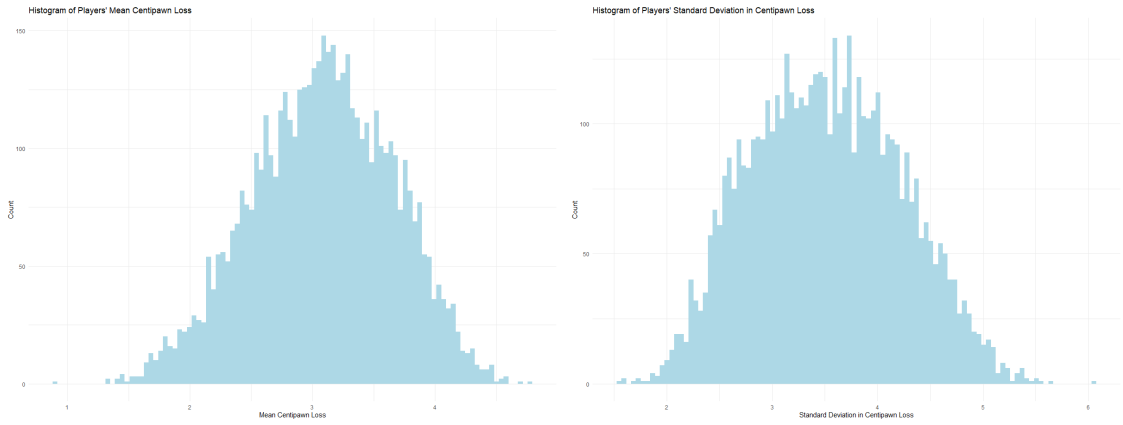


Figure 13. **Histogram of Log Transformation of Centipawn Loss.** Mean CP and standard deviation cp was log transformed and then plotted as a histogram. This was done to determine if the plot followed a normal distribution. The results are plotted above.

The resulting Q-Q plots, although still not looking normal, do appear to confirm normality better than before the log transformation (Figure 14).

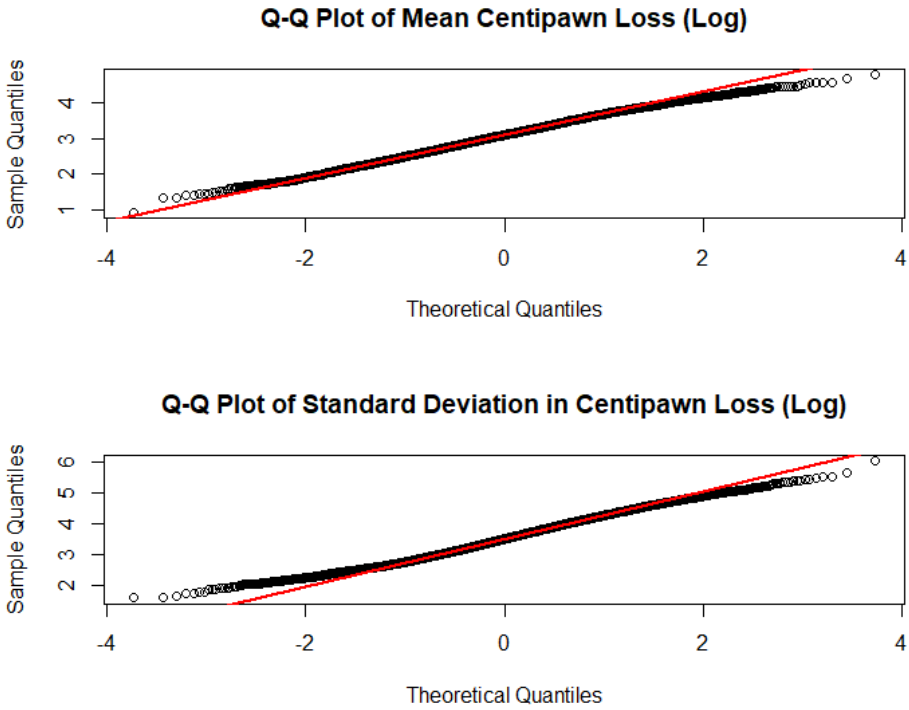


Figure 14. **Q-Q Plot of Log Transformed Centipawn Loss**. A Q-Q plot was used to determine whether the log transformed mean CP and standard deviation CP follow a normal distribution. The results are plotted above.

While the Shapiro-Wilks Test for the log transformed mean CP and std CP still result in p-values that are statistically significant, we can see some reduction in the p-values (Figure 15).

```
        Shapiro-Wilk normality test

data:  rand_samp_mean
W = 0.9966, p-value = 2.817e-09
```

```
        Shapiro-Wilk normality test

data:  rand_samp_std
W = 0.99252, p-value = 1.454e-15
```

Figure 15. **Shapiro Wilks Test for Log Transformed Centipawn Loss**. Shapiro Wilks test was applied both to the log transformed mean CP and standard deviation CP to determine if they follow a normal distribution. The results are shown above.

Although the Shapiro-Wilks Test rejects that the mean CP and std CP after log transformation are normally distributed, the Q-Q plots and histograms appear approximately normal. As such, for our linear regression, we will use the log transformed values for the two response variables.

Linear Regression

For predicting mean CP, we considered age, years spent playing chess (denoted as time), Elo, opponent Elo, and std CP. For predicting std CP, we had the same explanatory variables as the mean CP model, except with std CP swapped for mean CP. When checking for multicollinearity, however, we discover that age and time are correlated (Figure 16).

```
> vif(lm(Std_CP ~ Age + Time + Mean_CP + Elo + OppElo, data = players))
      Age      Time   Mean_CP       Elo    OppElo
38.564878 40.000101  1.007928  1.571785  1.418133
> vif(lm(Mean_CP ~ Age + Time + Std_CP + Elo + OppElo, data = players))
      Age      Time    Std_CP       Elo    OppElo
38.581020 40.019047  1.005146  1.570897  1.417285
```

Figure 16. **Multicollinearity Test**. A multicollinearity test was applied to the linear regression model in R for mean CP and std CP. The results are shown above.

To work around this, we ran separate models, one using time and another using age, and checked once more for multicollinearity. The output below shows that the issue of multicollinearity is addressed when age and time are separated from the models (Figure 17).

```
> vif(lm(Std_CP ~ Age + Mean_CP + Elo + OppElo, data = players))
     Age  Mean_CP      Elo   OppElo
1.143515 1.005387 1.477545 1.414888
> vif(lm(Mean_CP ~ Age + Std_CP + Elo + OppElo, data = players))
     Age   Std_CP      Elo   OppElo
1.143389 1.002137 1.476074 1.413979
> vif(lm(Std_CP ~ Time + Mean_CP + Elo + OppElo, data = players))
    Time  Mean_CP      Elo   OppElo
1.186071 1.005152 1.507732 1.417107
> vif(lm(Mean_CP ~ Time + Std_CP + Elo + OppElo, data = players))
    Time   Std_CP      Elo   OppElo
1.186006 1.001958 1.506326 1.416221
```

Figure 17. **Multicollinearity Test After Removing Age/Time** . A multicollinearity test was applied to the linear regression model in R for mean CP and std CP. The results are shown above.

We first considered the entirety of our data, including outliers. We initially kept outliers because chess matches at the grandmaster levels tend to have "calculated" suboptimal moves made by players in order to throw opponents into uncharted territory by deviating from well-known and well-documented theoretically optimal moves. Additionally, since our dataset consists of any over the board games regardless of time format, we believed that keeping inevitable inaccuracies caused by players because of running low on time was still an important indicator of a player's overall accuracy and consistency.

As such, the following eight models for the linear regression model, and using the olsrr library in R, we were able to find the best explanatory variable subset for each model based on R^2, Adjusted R^2, Predicted R^2, Mallow's Cp, AIC, and degrees of freedom. The subsets highlighted represent the same best subset generated despite having different explanatory variables (Table 1).

Table 1. **Best Features Used in the Linear Regression Models**. Multiple combinations of features were applied to the data for mean CP and standard deviation CP. R^2, Adjusted R^2, Predicted R^2, Mallow's Cp, AIC, and degrees of freedom were then used to evaluate the data with those showing similar performance being highlighted below.

Response Variable	Explanatory Variables	Best Subset
Mean CP	Age + Std CP + Elo + OppElo	Std CP + Elo + OppElo
Mean CP	Time + Std CP + Elo + OppElo	Std CP + Elo + OppElo
Mean CP	$(\text{Age} + \text{Std CP} + \text{Elo} + \text{OppElo})^2$	Age + Std CP + Elo + OppElo + Age*Std CP + Age*OppElo + Std CP*Elo
Mean CP	$(\text{Time} + \text{Std CP} + \text{Elo} + \text{OppElo})^2$	Time + Std CP + Elo + OppElo + Time*Std CP + Std CP*Elo + Std CP*OppElo + Elo*OppElo
Std CP	Age + Mean CP + Elo + OppElo	Mean CP + Elo + OppElo
Std CP	Time + Mean CP + Elo + OppElo	Mean CP + Elo + OppElo
Std CP	$(\text{Age} + \text{Mean CP} + \text{Elo} + \text{OppElo})^2$	Age + Mean CP + Elo + OppElo + Age*Mean CP + Age*OppElo + Mean CP*Elo + Elo*OppElo
Std CP	$(\text{Time} + \text{Mean CP} + \text{Elo} + \text{OppElo})^2$	Time + Mean CP + Elo + OppElo + Time*Mean CP +

		Mean CP*Elo + Elo*OppElo

As the table above shows, we have three optimal subsets for predicting mean CP. We then conducted an 80-20 train-test split and generated a linear regression model for each of the three best explanatory variable subsets. We then evaluated the performance of each of the three models by considering their root mean squared errors (RMSE), Adjusted R^2, Predicted R^2, AIC, and degrees of freedom to find the most optimal linear regression model (Table 2).

Table 2. **Results of the Three Optimal Subsets for Mean CP**. Three different subsets were used as features in the linear regression model. The models were then evaluated using RMSE, Adjusted R^2, Predicted R^2, AIC, and degrees of freedom with the best performing model highlighted.

Models	Explanatory Variables	RMSE	Adj. R^2	Pred. R^2	AIC	DF
1	Std CP + Elo + OppElo	0.219	0.856	0.860	-907.335	5
2	Age + Std CP + Elo + OppElo + Age*Std CP + Age*OppElo + Std CP*Elo	0.219	0.857	0.860	-918.286	9
3	Time + Std CP + Elo + OppElo + Time*Std CP + Std CP*Elo + Std CP*OppElo + Elo*OppElo	0.219	0.857	0.860	-917.962	10

Across all three models, RMSE, Adjusted R^2, and Predicted R^2 are all relatively similar. Most notably, the first model has the greatest AIC with the least degrees of freedom. Since model 1's degrees of freedom are quite lower than the other two models while still being relatively similar in regards to the other four metrics, we considered the first and simplest model to be the most optimal.

The same process was repeated for predicting std CP, and the resulting linear regression models were also evaluated based on the same criteria (Table 3).

Table 3. **Results of the Three Optimal Subsets for Standard Deviation CP**. Three different subsets were used as features in the linear regression model. The models were then evaluated using RMSE, Adjusted R^2, Predicted R^2, AIC, and degrees of freedom with the best performing model highlighted.

Models	Explanatory Variables	RMSE	Adj. R^2	Pred. R^2	AIC	DF
1	Mean CP + Elo + OppElo	0.437	0.856	1.000	888.259	5
2	Age + Mean CP + Elo + OppElo + Age*Mean CP + Age*OppElo +	0.436	0.857	0.998	866.427	10

	Mean CP*Elo + Elo*OppElo					
3	Time + Mean CP + Elo + OppElo + Time*Mean CP + Mean CP*Elo + Elo*OppElo	0.436	0.857	0.998	866.202	9

Across all three models, RMSE, Adjusted R^2, and Predicted R^2 are all relatively similar. Most notably, the first model has the greatest AIC with the least degrees of freedom. Since model 1's degrees of freedom are quite lower than the other two models while still being relatively similar in regards to the other four metrics, we considered the first and simplest model to be the most optimal.

Next, we considered the data without outliers to see if our prior linear regression models would drastically change. The same procedure and evaluation metrics were used to generate the eight models and the respective best explanatory variable subsets. The subsets highlighted represent the same best subset generated despite having different explanatory variables (Table 4).

Table 4. **Best Features Used in the Linear Regression Models for Data without Outliers**. Multiple combinations of features were applied to the data for mean CP and standard deviation CP. R^2, Adjusted R^2, Predicted R^2, Mallow's Cp, AIC, and degrees of freedom were then used to evaluate the data with similar performance being highlighted below.

Response Variable	Explanatory Variables	Best Subset
Mean CP	Age + Std CP + Elo + OppElo	Std CP + Elo + OppElo
Mean CP	Time + Std CP + Elo + OppElo	Std CP + Elo + OppElo
Mean CP	$(\text{Age} + \text{Std CP} + \text{Elo} + \text{OppElo})^2$	Age + Std CP + Elo + OppElo + Age*Std CP + Age*OppElo + Std CP*Elo
Mean CP	$(\text{Time} + \text{Std CP} + \text{Elo} + \text{OppElo})^2$	Time + Elo + OppElo + Time*Std CP + Time*OppElo + Std CP*Elo
Std CP	Age + Mean CP + Elo + OppElo	Mean CP + Elo + OppElo
Std CP	Time + Mean CP + Elo + OppElo	Mean CP + Elo + OppElo
Std CP	$(\text{Age} + \text{Mean CP} + \text{Elo} + \text{OppElo})^2$	Age + Mean CP + Elo + OppElo + Age*Mean CP + Age*Elo + Mean CP*Elo + Mean CP*OppElo + Elo*OppElo
Std CP	$(\text{Time} + \text{Mean CP} + \text{Elo} + \text{OppElo})^2$	Time + Mean CP + Elo +

		OppElo + Time*Mean CP + Time*Elo + Mean CP*Elo + Elo*OppElo

As the table above shows, we have three optimal subsets for predicting mean CP. We then conducted an 80-20 train-test split and generated a linear regression model for each of the three best explanatory variable subsets. We then evaluated the performance of each of the three models by considering their root mean squared errors (RMSE), Adjusted R^2, Predicted R^2, AIC, and degrees of freedom to find the most optimal linear regression model (Table 5).

Table 5. **Results of the Three Optimal Subsets for Mean CP without Outliers**. Three different subsets were used as features in the linear regression model. The models were then evaluated using RMSE, Adjusted R^2, Predicted R^2, AIC, and degrees of freedom with the best performing model highlighted.

Models	Explanatory Variables	RMSE	Adj. R^2	Pred. R^2	AIC	DF
1	Std CP + Elo + OppElo	0.219	0.857	0.856	-944.468	5
2	Age + Std CP + Elo + OppElo + Age*Std CP + Age*OppElo + Std CP*Elo	0.219	0.857	0.855	-959.351	9
3	Time + Elo + OppElo + Time*Std CP + Time*OppElo + Std CP*Elo	0.219	0.857	0.855	-960.152	9

Across all three models, RMSE, Adjusted R^2, and Predicted R^2 are all relatively similar. Most notably, the first model has the greatest AIC with the least degrees of freedom. Since model 1's degrees of freedom are quite lower than the other two models while still being relatively similar in regards to the other four metrics, we considered the first and simplest model to be the most optimal.

The same process was repeated for predicting std CP, and the resulting linear regression models were also evaluated based on the same criteria (Table 6).

Table 6. **Results of the Three Optimal Subsets for Standard Deviation CP without Outliers**. Three different subsets were used as features in the linear regression model. The models were then evaluated using RMSE, Adjusted R^2, Predicted R^2, AIC, and degrees of freedom with the best performing model highlighted.

Models	Explanatory Variables	RMSE	Adj. R^2	Pred. R^2	AIC	DF
1	Mean CP + Elo + OppElo	0.436	0.858	1.000	829.298	5
2	Age + Mean CP + Elo + OppElo +	0.437	0.858	0.998	819.275	11

	Age*Mean CP + Age*Elo + Mean CP*Elo + Mean CP*OppElo + Elo*OppElo					
3	Time + Mean CP + Elo + OppElo + Time*Mean CP + Time*Elo + Mean CP*Elo + Elo*OppElo	0.437	0.858	0.998	816.293	10

Across all three models, RMSE, Adjusted R^2, and Predicted R^2 are all relatively similar. Most notably, the first model has the greatest AIC with the least degrees of freedom. Since model 1's degrees of freedom are quite lower than the other two models while still being relatively similar in regards to the other four metrics, we considered the first and simplest model to be the most optimal.

Results

Looking closer at the optimal linear regression models for both with and without outliers for our response variables of interest, we are left with the following equations (Table 7).

Table 7**. Linear Regression Models for CP with and without Outliers**. Equations for linear Regression Models are selected from best performing models and shown below.

Models	**With Outliers**
1	**Mean CP** = 0.996 + 0.745***Std CP** - 0.0002***Elo**
2	**Std CP** = -0.417 + 1.140***Mean CP** + 0.0001***Elo**
	Without Outliers
3	**Mean CP** = 0.968 + 0.743***Std CP** - 0.00002***Elo**
4	**Std CP** = -0.404 + 1.146***Mean CP**

Based on model 1, we observe that on average, given identical player Elos, with every 1% increase in Std CP, Mean CP increases by $(1.01^{0.745} - 1) * 100$, or 0.744%, and on average, given identical Std CPs, with every point increase in player Elo, Mean CP increases by $(e^{-0.0002} - 1) * 100$, or -0.016% (i.e. Mean CP decreases by 0.016%).

Based on model 2, we observe that on average, given identical player Elos, with every 1% increase in Mean CP, Std CP increases by $(1.01^{1.14} - 1) * 100$, or 1.141%, and on average, given identical Mean CPs, with every 1 point increase in player Elo, Std CP increases by $(e^{0.0001} - 1) * 100$, or 0.012%.

Based on model 3, we observe that on average, given identical player Elos, with every 1% increase in Std CP, Mean CP increases by $(1.01^{0.743} - 1) * 100$, or 0.742%, and on average, given identical Std CPs, with every point increase in player Elo, Mean CP increases by $(e^{-0.00002} - 1) * 100$, or -0.016% (i.e. Mean CP decreases by 0.016%).

Based on model 4, we observe that on average, with every 1% increase in Mean CP, Std CP increases by $(1.01^{1.146} - 1) * 100$, or 1.147%.

Overall, a majority of the models both with and without outliers seem to coincide with having Mean CP or Std CP alongside Elo as important explanatory variables for predicting how well and consistently a player performs.

Question 2

Question: *How does Niemann's growth compare to that of other grandmasters?*

Data Preparation

To answer this question some measure of growth is needed. The Elo rating system is a player rating system used in many zero-sum sports and is particularly popular in chess to give an indicator of a players performance and skill level. The Elo score for a given player is updated after every game and is included in the dataset. For each player, the maximum Elo score was calculated and the first recorded Elo score was subtracted from it. Giving a measure of "growth". It is important to note that all following analysis will be based only on games between a player's first game and their game with the maximum Elo.

Exploratory Data Analysis

To get a preliminary look at how Niemann compares to other players a graph is shown below plotting the Elo over time for Niemann and a random subset of the other grandmasters. Carlsen is included because of his role as the world champion. As seen in the EDA section, Niemann's Elo has significantly increased over the last few years in comparison to his grandmaster counterparts (Figure 10).

After calculating the "growth" of each player we can visualize that as a histogram below. As can be seen most of the players have around 0-25 points of Elo growth over the last few years with a few outliers having more than that. The red line in the histogram indicates Niemann's growth. At nearly 270 points of Elo gain he has drastically outperformed the other players (Figure 18).

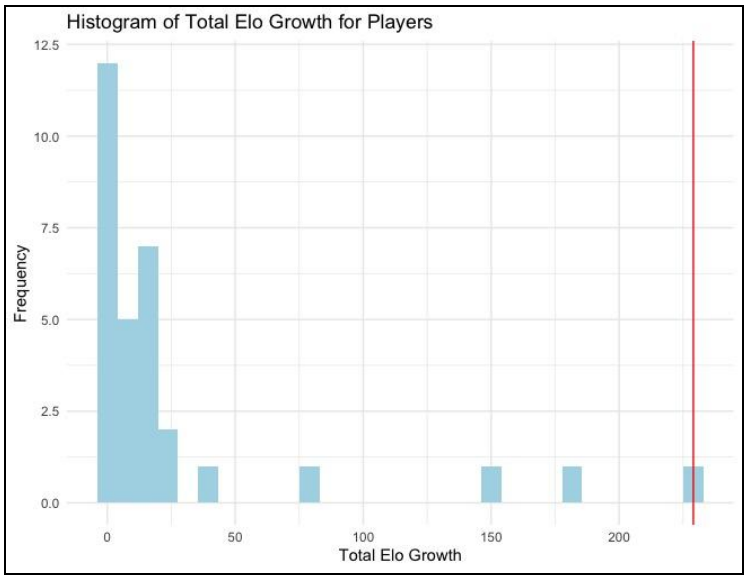


Figure 18. **Histogram of Total Elo Growth**. Total Elo was plotted as a histogram for all players. Majority of players had less than 50 points of growth while Niemann had the highest growth demonstrated by the red line.

When Elo Growth is broken down by average Elo growth per year, a similar histogram is found as shown below where the red line indicates Niemann's Elo growth per year on average (Figure 19).

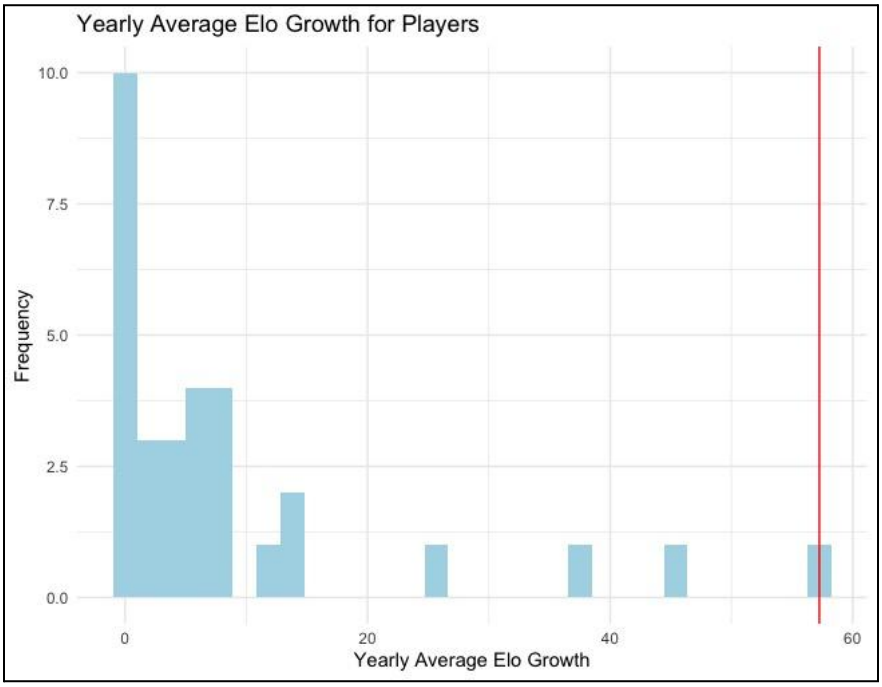


Figure 19. **Histogram of Average Elo Growth**. Total Elo was plotted as a histogram for all players. Majority of players had less than 20 points of growth while Niemann had the highest average growth demonstrated by the red line.

From the first plot it is also clear that Niemann has played significantly more games than the other grandmasters. Plotting Elo Growth vs Number of games it took to attain that growth a less severe story is told. While Niemann has outperformed all other players in Elo growth, he has also played over 300 games while most of the other grandmasters played between 0 and 75 games to achieve their max Elo. The plot also indicates a linear correlation between number of games played and total Elo growth, therefore, perhaps it makes sense that Niemann's Elo change is so much higher than the other grandmasters, given his significant edge in games played. To further solidify this finding, the number of games vs the average Elo change per Year shows a similar finding of a positive linear correlation between games played and Elo change (Figure 20).

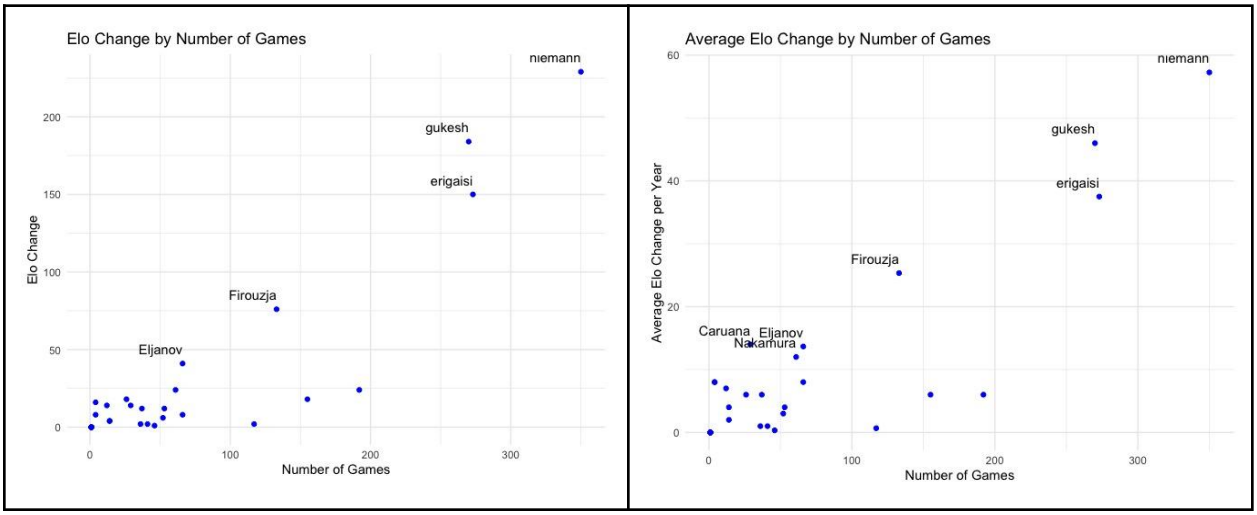


Figure 20. **Scatter Plots of Average Elo Growth and Total Elo Change per Number of Games**. Total Elo and Average Elo was plotted as a scatter plot for all players. The results are shown above.

Bootstrapping/Hypothesis Testing

From the histograms shown above it is evident that the data is not normally distributed and in addition there are few data points (n<35). Therefore, it would be incorrect to perform basic T-tests or other hypothesis tests that assume a normal distribution. To solve this problem of non-normality bootstrapping can be used. To perform bootstrap analysis with this data, 10,000 repeated samples with replacement were taken of the Elo growth variable and the average Elo change per game played variable. This results in a normal distribution of sample means, proven by the central limit theorem. Taking the quartiles of these distributions will inform the 95% confidence intervals that can be used to see if Niemann's growth is significantly different from the average player.

Hypothesis:

$$Null\ Hypothesis\ 1\text{:}\ H_0\text{:}\ mean(player\ growth\ per\ year)\ =\ niemann's\ growth\ per\ year$$

$$Alternative\ Hypothesis\ 1\text{:}\ H_A\text{:}\ mean(player\ growth\ per\ year)\ !=\ niemann's\ growth\ per\ year$$

$Null\ Hypothesis\ 2\colon\ H_0\colon\ mean(player\ growth\ per\ game)\ =\ niemann's\ growth\ per\ game$

$Alternative\ Hypothesis\ 1\colon\ H_A\colon\ mean(player\ growth\ per\ game)\ !=\ niemann's\ growth\ per\ game$

Results

For the first hypothesis tests the histogram of bootstrapped sample means is shown below. Niemann's Elo growth is indicated by the red line. The associating confidence interval for the mean Elo growth for a player is [4.55 to 14.24]. Niemann's growth per year is 57.25 which is well outside of those bounds. Therefore we can reject the null hypothesis of this test and conclude that niemann's growth per year is significantly different from the mean (Figure 21).

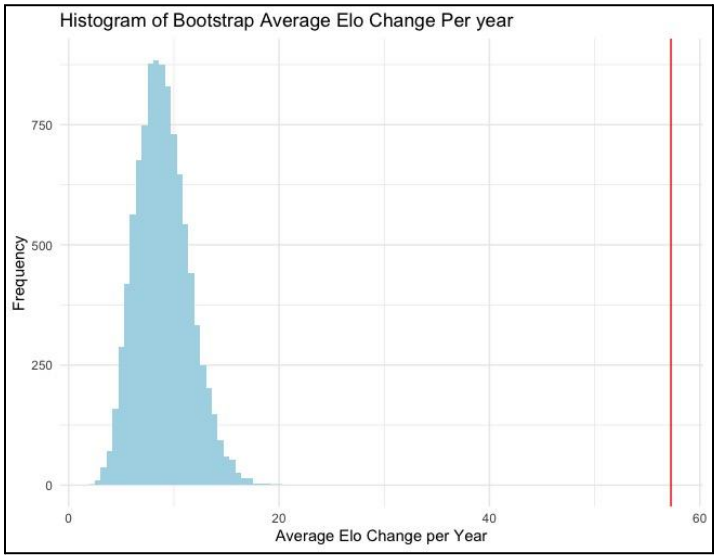


Figure 21. **Bootstrapped Average Elo Change Per Year**. Average Elo change was bootstrapped then plotted above.

For the second hypothesis test our findings are different. The histogram of the bootstrapped sample means of average Elo change per game is shown below. Again the red line indicates Niemann's change per game. Here the associating 95% confidence interval for the mean Elo change per game is [0.211 to 0.745] points of increase per game. Niemann's average increase per game is 0.654 which falls inside of the confidence interval. Therefore we cannot reject the null and have no evidence to suggest that Niemann's Elo change per game is significantly different from the mean (Figure 22)..

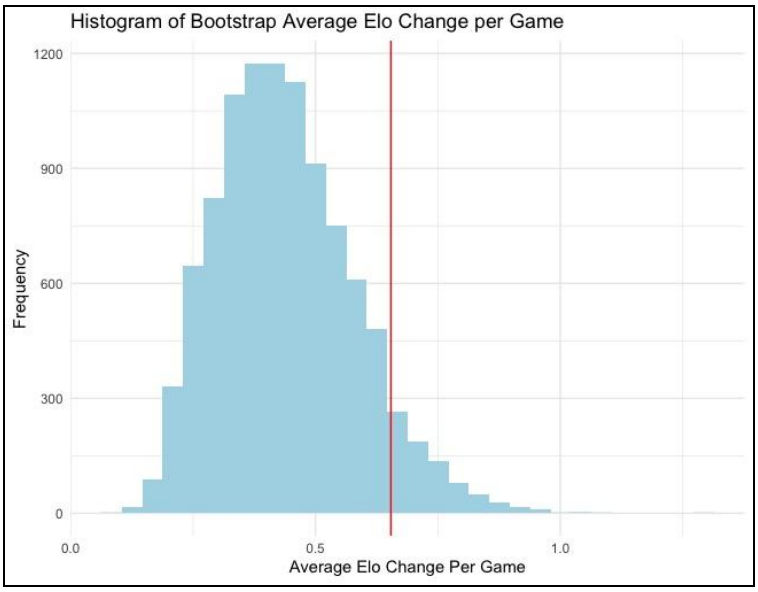


Figure 22. **Bootstrapped Average Elo Change Per Game**. Average Elo change was bootstrapped then plotted above.

Question 3

Question: *How do other Grandmasters perform compared to Magnus Carlsen?*

T-Tests

To compare Carlsen's Mean CP and Std CP against each of the other Grandmasters in the data, we decided that a two sample T-Test would be best. However, as discovered by the QQ Plots and Shapiro Wilks Test in question one, the data is not normally distributed. Therefore, we performed a log transformation before running our T-Tests. The resulting tests showed that, on average, Carlsens Mean CP was less than 85% of players (Table 7).

Table 7. **Mean CP T-Test Results**. T-Tests were performed with Carlsen and each of the other players, with the alternative hypothesis being "greater".

Player	**Metric**	**P-Value**
Bu	Mean CP	0.0003076189
Ding	Mean CP	0.0001958873
Nepo	Mean CP	0.0001694836
Wei	Mean CP	0.0159982436
Yu	Mean CP	0.0371047003

The same procedure was conducted for his Std CP, and the results show that his Std CP was less than 82% of players (Table 8).

Table 8. **Std CP T-Test Results**. T-Tests were performed with Carlsen and each of the other players, with the alternative hypothesis being "greater"

Player	**Metric**	**P-Value**
Andreikin	Std CP	2.524653e-02
Bu	Std CP	6.122359e-04
Ding	Std CP	3.393064e-05
Nepo	Std CP	5.687054e-06
Wei	Std CP	6.322617e-03
Yu	Std CP	7.677776e-03

Kruskall Wallis

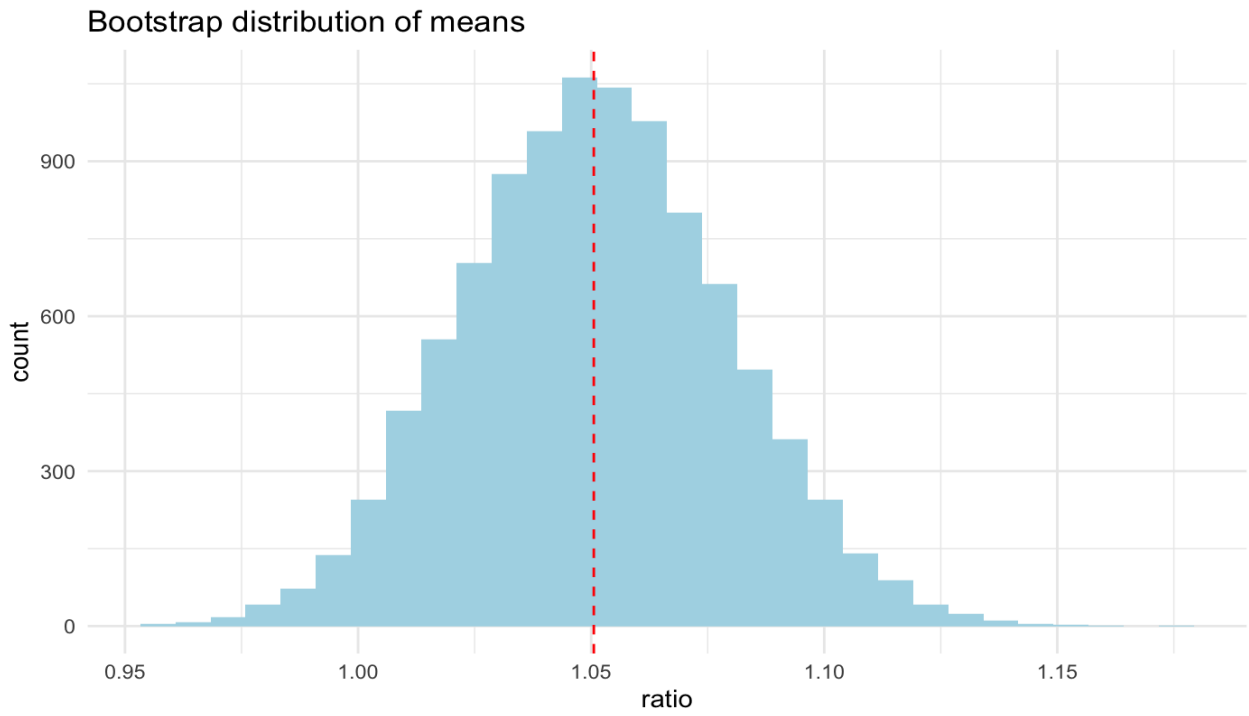
To compare the medians of Mean CP and Std CP, we performed a Kruskall Wallis test with Carlsen both included and excluded from the data. Since this is a non-parametric test, we used a copy of the data we had not performed a log transformation on. The results were not very informative. In both cases, we were able to reject the null hypothesis that the medians were equal.

Question 4

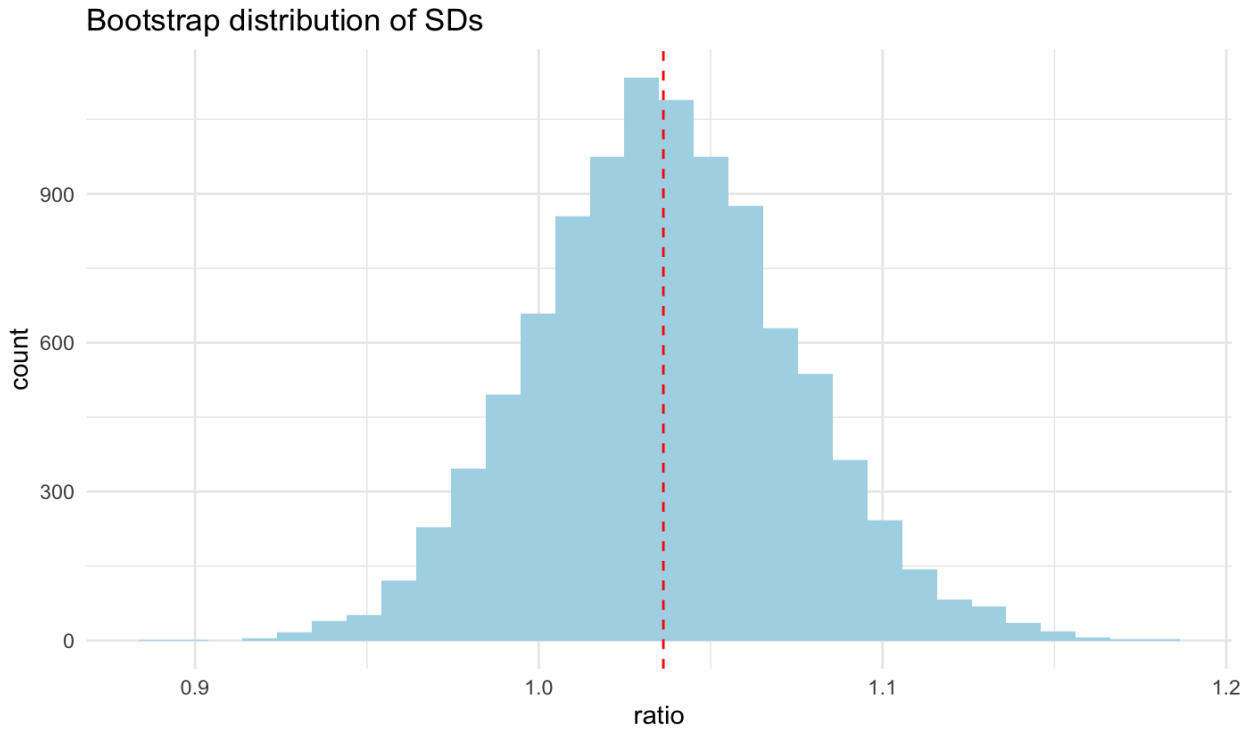
Question: *How do other GMs perform when compared to the rising chess prodigy Niemann?*

Bootstrapping/Hypothesis Testing

In order to first get a rough estimate of whether Niemann's statistics are significantly different from others, we perform a bootstrap analysis. The features we chose are Mean_cp and Std_cp, and by bootstrapping and calculating the ratio of statistics of Niemann over those of others, we have a 95% confidence interval of what the mean ratio would be.



95% Confidence Interval: [1.00 : 1.11]



95% Confidence Interval: [0.97 : 1.11]

Figure 23. **Bootstrapped Ratio of Mean CP and Standard Deviation CP**. Mean and standard deviation CP were bootstrapped then calculated the ratio over Niemann were plotted above.

We can see that the confidence interval of Mean CP falls in a range larger than 1, which suggests that the Mean CP of Niemann is higher than others (Figure 23).

In order to further examine whether these differences are statistically significant, we would perform a T-test.

Table 9. **Mean CP and Std CP T-Test Results**. T-Tests were performed with Niemann and other players as a whole, with the alternative hypothesis being "greater"

	t	**df**	**p-value**
Mean CP	1.8045	428.4	0.03593
Std CP	0.96464	428.9	0.1676

Upon conducting a two-sample T-test, we found that p value for Mean CP is 0.04, and p value for Std CP is 0.17. For Mean CP, since the p value is less than 0.05, we reject the null hypothesis stating that the statistics for other players with Niemann is the same (Table 9). However, we fail to reject the null hypothesis for Std CP, stating that Niemann's Std CP is not different from other players.

In conclusion, since the Mean CP measures the deviation from the optimal steps calculated by the computer, we fail to provide evidence to prove that Niemann is cheating using a computer.

Conclusion

Our linear regression results found that for player's with the same Elo, their accuracy, on average, decreases as their consistency decreases (and vice versa), and for players with the same standard deviation in centipawn loss, a player's accuracy also tends to increase, on average, as their Elo increases.

Using Elo Change as a measure of growth for each player we found some evidence to suggest that Niemann had a significantly greater growth than other players. however the findings were not conclusive when looking at the average growth per game. Indicating a need for further investigation into Niemann's drastic growth, but not providing sufficient evidence for cheating.

Our two sample T-Tests found that, on average, Carlsen's Mean CP was greater than only 15% of players, and his Std CP was greater than only 18%. This makes sense for a skilled Grandmaster because if both Mean CP and Std CP are proxies for how much a player errs from an optimal move, a skilled Grandmasters Mean CP and Std CP would be less than the majority of players (as shown by our T-Tests).

Our two sample T-Test for Niemann and other players' statistics showed that Niemann's Mean CP was higher than other players, and Niemann's Std CP was the same with other players. Since Mean CP measures the deviation from the optimal steps calculated by the computer, it is

highly unlikely that Niemann is cheating by using a computer that can calculate better moves for him, which will result in a smaller Mean CP.

All in all, we must also consider the limitations of our analysis. First, our data has naturally occurring dependencies. Elo is inherently correlated with the player, CP will be correlated with game time format (where shorter game time controls will have less accurate and consistent play than longer, classical formats), and CP calculations will differ based on the strength of the engine (i.e. the version of Stockfish used). Additionally, our dataset only contains over the board games and 31 of the top 50 GMs, so our sample size may not be fully representative of top level GM performance. Finally, our data contained many outliers and influential points, which is both counterintuitive yet expected. We'd expect the top GMs to perform relatively similar to each other in terms of accuracy and consistency, which makes the numerous amount of outliers quite surprising; at the same time, this could be explained by players intentionally making suboptimal moves that escape well-known and well-documented theory to throw opponents off guard and out of their preparation.

For further steps, time series analysis could provide some valuable insight in looking at overall trends over multiple months and years rather than looking at games as single snapshots as we have for this analysis. Furthermore, our dataset lacks game data following Magnus Carlsen's cheating allegations against Hans Niemann; thus, future analysis could include more recent data, which may prove to be insightful.