EXPLORATORY DATA ANALYSIS PROJECT

ON

FIDE CHESS RANKINGS

In partial fulfilment for the requirements of the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COUMPETR SCIENCE AND ENGINEERING

Submitted by:

Name of the Student : LANKA SRINIVASUDU

Registration Number : 12106470

Roll No : RK21URA28

Section : K21UR

submitted to

LOVELY PROFESSIONAL UNIVERSITY

Jalandhar, Punjab, India.

TABLE OF CONTENT

1.Introduction:

Chess is a board game for two players, called White and Black, each controlling an army of chess pieces, with the objective to checkmate the opponent's king. It is sometimes called international chess or Western chess to distinguish it from related games such as xiangqi (Chinese chess) and shogi (Japanese chess). The recorded history of chess goes back at least to the emergence of a similar game, chaturanga, in seventh century India. The rules of chess as they are known today emerged in Europe at the end of the 15th century, with standardization and universal acceptance by the end of the 19th century. Today, chess is one of the world's most popular games played by millions of people worldwide.

Chess is an abstract strategy game that involves no hidden information and no elements of chance. It is played on a chessboard with 64 squares arranged in an 8×8 grid. At the start, each player controls sixteen pieces: one king, one queen, two rooks, two bishops, two knights, and eight pawns. White moves first, followed by Black. The game is won by checkmating the opponent's king, i.e. threatening it with inescapable capture. There are also several ways a game can end in a draw.

Organized chess arose in the 19th century. Chess competition today is governed internationally by FIDE (**The International Chess Federation**). The first universally recognized World Chess Champion, Wilhelm Steinitz, claimed his title in 1886; Ding Liren is the current World Champion. A huge body of chess theory has developed since the game's inception.

The International Chess Federation or World Chess Federation, commonly referred to by its French acronym **FIDE**, is an international organization based in Switzerland that connects the various national chess federations and acts as the governing body of international chess competition. FIDE was founded in Paris, France, on July 20, 1924. Its motto is Gens una sumus, Latin for 'We are one Family'. In 1999, FIDE was recognized by the International Olympic Committee (IOC). As of May 2022, there are 200 member federations of FIDE.

2.Domain Knowledge:

Chess, as we know it today, was born out of the Indian game **Chaturanga** before the 600s AD. The game spread throughout Asia and Europe over the coming centuries, and eventually evolved into what we know as chess around the 16th century. One of the first masters of the game was a Spanish priest named Ruy Lopez. Although he didn't invent the opening named after him, he analyzed it in a book he published in 1561. Chess theory was so primitive back then that Lopez advocated the strategy of playing with the sun in your opponent's eyes!

A chess rating system is a system used in chess to estimate the strength of a player, based on their performance versus other players. They are used by organizations such as FIDE, the US Chess Federation (USCF or US Chess), International Correspondence Chess Federation, and

the English Chess Federation. Most of the systems are used to recalculate ratings after a tournament or match but some are used to recalculate ratings after individual games. Popular online chess sites such as chess.com, Lichees, and Internet Chess Club also implement rating systems. In almost all systems, a higher number indicates a stronger player. In general, players' ratings go up if they perform better than expected and down if they perform worse than expected. The magnitude of the change depends on the rating of their opponents.

The Elo rating system is currently the most widely used.

The first modern rating system was used by the Correspondence Chess League of America in 1939. Soviet player Andrey Khachaturov proposed a similar system in 1946 (Hooper & Whyld 1992:332). The first one that made an impact on international chess was the Ingo system in 1948. The USCF adopted the Harkness system in 1950. Shortly after, the British Chess Federation started using a system devised by Richard.

W. B. Clarke. The USCF switched to the Elo rating system in 1960, which was adopted by FIDE in 1970 (Hooper & Whyld 1992:332).

2.1 Ratings in chess:

2700+	No formal title, but sometimes informally called super grandmasters.
2500–2700	Most Grand Masters
2400–2500	Most International Masters
2300–2400	FIDE masters
1800–2000	Candidate Masters
1600–1800	Class A, Category 1
Below 1000	Novices

2.2 Rules of chess:

Chess is played on a chessboard, a square board divided into a grid of 64 squares (eight-by-eight) of alternating color (similar to the board used in draughts).[1] Regardless of the actual colors of the board, the lighter-colored squares are called "light" or "white", and the darker- colored squares are called "dark" or "black". Sixteen "white" and sixteen "black" pieces are placed on the board at the beginning of the game. The board is placed so that a white square is in each player's near-right corner. Horizontal rows are called ranks, and vertical columns are called files. Each player controls sixteen pieces:

At the beginning of the game, the pieces are arranged as shown in the diagram: for each side one king, one queen, two rooks, two bishops, two knights, and eight pawns. The pieces are placed, one per square, as follows:

- Rooks are placed on the outside corners, right and left edge.
- Knights are placed immediately inside of the rooks.
- Bishops are placed immediately inside of the knights.
- The queen is placed on the central square of the same color of that of the piece: white queen on the white square and black queen on the black square.
- The king takes the vacant spot next to the queen.
- Pawns are placed one square in front of all of the other pieces.

Popular mnemonics used to remember the setup are "queen on her own color" and "white on right". The latter refers to setting up the board so that the square closest to each player's right is white.

If it is not possible to get out of check, the king is checkmated and the game is over (see the next section).
In informal games, it is customary to announce "check" when making a move that puts the opponent's king in check. In formal competitions, however, checks are rarely announced.

3.Why did you choose this dataset?

Here are some reasons, for choosing this dataset:

- I chose FIDE chess rankings dataset because of the respect I have for this game and the history that goes back to centuries.
- Another reason is this dataset is updated thoroughly.
- The FIDE dataset contains a wealth of information on chess games, including player ratings, tournament details, game outcomes, and more. This richness allows for a thorough and detailed analysis.
- The FIDE dataset contians international chess competitions, making the dataset representative of a global chess-playing community. This diversity can lead to interesting findings regarding playing styles, strategies, and trends across different regions.
- This dataset from Kaggle contains 200 rows and 7 columns.
- This dataset is very easy to demonstrate the EDA topics we've learnt during the course EDA 351.

4. Libraries used and approaches:

4.1 Pandas:

Pandas is a Python library for data manipulation and analysis, featuring two main data structures, Series and Data Frame, for handling structured data efficiently. It simplifies tasks such as data cleaning, aggregation, and transformation. Pandas is widely used in data exploration and analysis due to its versatility and ease of use.

4.2 NumPy:

NumPy, short for "Numerical Python," is a fundamental Python library for numerical and mathematical operations. It introduces a powerful N- dimensional array object, allowing efficient handling of large datasets and mathematical computations. NumPy is the cornerstone of scientific computing and data analysis in Python, providing essential tools for array manipulation and mathematical functions.

4.3 Matplotlib:

Matplotlib is a popular Python library for creating static, animated, and interactive visualizations in various formats. It offers a comprehensive set of tools for generating high-quality plots and charts for data exploration and presentation. Matplotlib is widely used in data science, scientific research, and data visualization due to its flexibility and customization options.

4.4 Seaborn:

Seaborn is a Python data visualization library built on top of Matplotlib, designed to create informative and aesthetically pleasing statistical graphics. It simplifies the process of creating complex visualizations with minimal code. Seaborn offers a wide range of plot types, color palettes, and themes, making it a popular choice for data analysts and researchers for data exploration and presentation.

4.5 Scikit-Learn:

For machine learning tasks, if applicable, Scikit-Learn provides a comprehensive suite of tools for data mining and data analysis. It includes modules for classification, regression, clustering, and more.

4.6 Statsmodels:

This library is particularly useful for statistical modelling. It can be

employed for regression analysis, hypothesis testing, and exploring relationships between variables.

5. Approach to solve the problems:

1. Understanding the Problem Statement:

Understanding the problem statement is paramount in any data analysis endeavor. It involves gaining a deep comprehension of project objectives, the nature of the available data, and the specific insights sought. This initial phase serves as a guiding compass, steering the direction of the entire analysis process. It helps in formulating relevant questions, setting clear goals, and determining the appropriate data sources and methods needed to extract valuable insights from the data.

2. Data Collection:

Data collection is a pivotal step in the data analysis process. It entails acquiring and assembling relevant data from various sources, ensuring data quality and consistency. This phase involves selecting appropriate data collection methods, such as surveys, sensors, or databases, to align with the research objectives. Effective data collection forms the bedrock for subsequent analysis, enabling insights, trends, and patterns to be extracted to address the core objectives of the project.

3. Data Cleaning:

Data cleaning is a critical stage in data analysis, focused on refining and preparing the collected data for analysis. It involves identifying and addressing issues like missing values, outliers, and inconsistencies. Data cleaning techniques encompass imputation, data transformation, and outlier handling. Ensuring data integrity and accuracy during this phase is crucial for reliable and meaningful insights. A clean dataset serves as the foundation for accurate analysis and informed decision-making.

4. Data Preprocessing:

Data preprocessing is a vital stage in data

analysis, encompassing a series of transformations and enhancements to optimize data for subsequent analysis. This phase involves tasks like feature scaling, normalization, encoding categorical variables, and handling data imbalances. Data preprocessing aims to improve the quality and compatibility of data, making it suitable for machine learning models and other analytical techniques. By preparing the data effectively, this step contributes to more accurate and meaningful insights.

5. Exploratory Analysis:

Exploratory data analysis (EDA) is a foundational step in data analysis, dedicated to gaining a preliminary understanding of the data's characteristics, relationships, and patterns. Through the use of visualizations, summary statistics, and data exploration techniques, EDA uncovers key insights, identifies trends, and highlights potential anomalies or outliers. EDA sets the stage for more in-depth analysis and helps researchers formulate hypotheses and refine their analytical approach, ensuring a robust foundation for decision-making and further investigation.

5.2 Steps of EDA:

5.2.1 Importing Libraries:

In this initial step, the necessary Python libraries such as numpy, pandas, matplotlib, and seaborn are imported.

5.2.2 6. Reading CSV File:

- ✓ Reads data from a CSV file named "imdb_top_1000.csv" located in the current directory.
- ✓ Uses Pandas' read_csv() function to read the CSV data and stores it in a Data Frame named "df."
- ✓ Data Frames are Pandas' data structures for working with tabular data.
- ✓ Displays the first 5 rows of the Data Frame "df" using the head() method.

6.Data Describe:

6.1 Displaying last 5 rows:

- df.tail() displays the last 5 rows of the Data Frame "df."
- It helps you inspect the tail end of the dataset to check for any patterns or issues present in the data.

6.2 Generating summary statistics for numerical columns:

- df.describe() generates summary statistics for numerical columns in the Data Frame "df."
- This includes statistics like count, mean, standard deviation, minimum, and maximum values, providing insights into the central tendencies and spread of the data.

6.2 Displaying basic information about the data frame, including

data types and non-null counts:

- df.info() provides an overview of the Data Frame's structure.
- It displays information such as the number of non-null values, data types, and memory usage for each column.
- Useful for assessing data types and identifying missing values.

6.4 counting the number of unique values in each column:

- df.nunique() calculates the number of unique values in each column of the Data Frame.
- It helps in understanding the level of variability within each column and identifying potential categorical features.

6.5 Displaying null values of each column:

- df.isnull().sum() counts the number of missing (null) values in each column.
- It's valuable for identifying columns with missing data, which may require data cleaning or imputation.

7.Data Cleaning:

7.1. Filling the null values:

- clean is a new data Frame created by filling missing values (NaN)

in the original datagram "df" with zeros (0).

- This step replaces missing values with a default value to make the data more complete.
-

7.2. Selecting specific columns two a new CSV file:

- Defines a list of column names that you want to focus on in the subsequent steps.
- These columns are selected for further analysis and export. This block creates a new CSV file named "cleaneddata.csv" containing only the specified columns from the "clean" Data Frame.

7.3. Reading data from new CSV file:

- Reading the "cleaneddata.csv" file into a new Data Frame called "cleandf."
- This step allows you to work with the cleaned data in a new Data Frame.
- Displaying the first 5 rows of the "cleandf" Data Frame to examine the cleaned data.

7.4. Detecting and marking duplicate rows:

- Checking for duplicate rows in the "cleandf" Data Frame and returns a Boolean Series indicating whether each row is a duplicate.

7.5. Filtering and displaying rows with null-values:

- Selecting rows in the "cleandf" data Frame where there are still missing values in any column.
- This helps identify any remaining rows with missing data in the cleaned dataset.

8. Outlier Analysis:

8.1. IQR Method:

- Defining a function to find outliers using the IQR Method.
- Calculating the first quartile and third quartile.
- Calculating the interquartile range (IQR).

- Calculating the lower and upper bounds to identify the outliers.
- Returning a Boolean mask indicating which data points are outliers.
- Displaying rows containing potential outliers.

We code to define a function, find_outliers_iqr, it is used to identify outliers in a dataset using the Interquartile Range (IQR) method. This function calculates the first quartile (Q1), third quartile (Q3), and the Interquartile Range (IQR) in our dataset. It then determines lower and upper bounds for identifying outliers as 1.5 times the IQR below Q1 and 1.5 times the IQR above Q3, respectively. The function returns a boolean mask indicating which data points are outliers.

Then the code applies this outlier detection function to the '**age**' column of a DataFrame (df) and stores the result in a boolean DataFrame named outliers. The code then selects rows from the original DataFrame where at least one column has an outlier, based on the boolean mask. The resulting DataFrame, potential_outliers, contains rows with potential outliers in the 'age' column. Finally, it prints and displays these rows with potential outliers.

8.2. Box plot:

- Creating a figure with a specified size (10x6 inches)
- Creating a boxplot with 'genre' on the x-axis and 'rating' on the y-axis.
- Rotate x-axis labels by 45 degrees for better readability.
- Set a label for the y-axis.
- Set a title for the plot.
- Displaying the plot.

We created a boxplot using Seaborn to visualize the distribution of chess player ratings ('ELO') across different age groups ('age') from the DataFrame 'df'. The x-axis represents age, the y-axis represents ELO ratings and also plot includes rotated x-axis labels for readability, axis labels, and a title. The 'tight_layout()' function enhances the presentation by optimizing layout spacing, and the plot is displayed with a specified size of 10x6 inches.

9.Data Visualization:

9.1. Univariate Analysis:

Univariate analysis of my dataset(FIDE CHESS) involves examining individual variables in isolation to understand their distributions, central tendencies, and spread and analysis may include generating summary statistics, visualizations like histograms, distribution plot, boxplots, and exploring key characteristics of variables such as player ratings, tournament frequencies, or game outcomes. Univariate analysis provides a foundational understanding of the individual features within the dataset.

9.1.1. Histogram Plot:

- Setting Figure Size using figsize().
- Creating a histogram using sns.histplot().
- Labeling Axes using xlabel, ylabel.
- Adding a title using plt.title().
- Displaying the plot using plt.show().

Insights from histogram plot:

This is a histogram graph that shows the distribution of ELO ratings for chess players. The x-axis represents the ELO rating and the y-axis represents the frequency of players with that rating. The graph is blue in color with a white background. The majority of players have a rating between 2600 and 2700, with a sharp drop off in frequency as the rating increases. The highest frequency is around 40 players with a rating of 2600, while the lowest frequency is around 0 players with a rating of 2850. The graph is titled “Distribution of ELO”.

9.1.2. Pie chart:

- Setting Figure Size using figsize().
- Calculating genre counts using [].value_counts.
- Exploding slices using explode.
- Creating a pie chart using plt.pie().
- Adding a title using plt.title().

- Equaling the aspect ratio using plt.axis().

- Displaying the plot using plt.show().

Insights from Pie Chart:

This is a pie chart titled “Distribution of Age Genres”. The chart is made up of different colored segments representing different age groups. The age groups are represented by numbers 1 through 17. The percentages for each age group are written on the chart. The chart is in a circular shape with a white background. The highest percentage is 58% for age groups 3 and 15, while the lowest percentage is 25% for age groups 6 and 16.

9.1.3. Box Plot:

- Creating a figure with a specified size (10x6 inches).
- Creating a boxplot with 'genre' on the x-axis and 'rating' on the y-axis.
- Rotate x-axis labels by 45 degrees for better readability.
- Set a label for the y-axis.
- Set a title for the plot.
- Displaying the plot.

Insights from Box Plot:

This is a box plot graph that shows the distribution of ratings by age. The x-axis represents age, and the y-axis represents Elo rating. The graph shows that the ratings are highest for the age group 24-26 and lowest for the age group 30-32. The graph has a title “Rating Distribution by age” and a legend that explains the colors of the boxes. The boxes are colored in different shades of green, blue, purple, and pink. A box plot is a type of chart that depicts a group of numerical data through their quartiles. It is a simple way to visualize the shape of our data. It makes comparing characteristics of data between categories very easy. A box plot gives a five-number summary of a set of data which is- Minimum – It is the minimum value in the dataset excluding the outliers First Quartile (Q1) – 25% of the data lies below the First (lower) Quartile. Median (Q2) – It is the mid-point of the dataset. Half of the values lie below it and half above. Third Quartile (Q3) – 75% of the data lies below the Third (Upper) Quartile. Maximum – It is the maximum value in the dataset excluding the outliers. The area inside the box (50% of the data) is

known as the Inter Quartile Range. The IQR is calculated as – IQR = Q3-Q1 Outliers are the data points below and above the lower and upper limit. The lower and upper limit is calculated as – Lower Limit = Q1 - 1.5IQR Upper Limit = Q3 + 1.5IQR The values below and above these limits are considered outliers and the minimum and maximum values are calculated from the points which lie under the lower and upper limit.

In this box plot, the x-axis represents age, and the y-axis represents Elo rating. The highest ratings are for the age group 24-26, with a median Elo rating of around 2550. The ratings gradually decrease as the age increases. The lowest ratings are for the age group 30-32, with a median Elo rating of around 2450.

9.1.4. Distribution Plot:

- Setting Figure Size using figsize().
- Creating a histogram using sns.histplot().
- Labeling Axes using xlabel, ylabel.
- Adding a title using plt.title().
- Displaying the plot using plt.show().

Insights from Distribution Plot:

This is a distribution plot that shows the distribution of the number of games played by a group of people. The x-axis represents the number of games played and the y-axis represents the frequency of people who played that many games. The graph appears to be positively skewed, meaning that there are more people who played a smaller number of games than those who played a larger number of games. The highest frequency is around 1000 games played, while the lowest frequency is around 4000 games played.

9.2 Bivariate Analysis:

We used Bivariate analysis to examine the relationships between two different variables in the dataset. This analysis aims to understand how changes in one variable may relate to changes in another, bivariate analysis could explore correlations or patterns between two variables,

such as player ratings and tournament performance, age and playing style, or any other pair of relevant factors. It provides insights into how these variables interact and influence each other within the chess context.

9.2.1. Bar plot:

- Grouping by Genre using groupby().
- Sorting by average gross using sort_values().
- Creating a bar plot of size (10,6) using figsize().
- Labeling Axes and title using xticks, yticks, xlabel, ylabel.
- Displaying the plot using plt.show().

Insights from Bar Plot:

The user sent five images, each with a different type of graph. The first image is a histogram graph that shows the distribution of ELO ratings for chess players. The highest frequency of players is around 2600 ELO rating. The second image is a pie chart titled “Distribution of Age Genres”. The highest percentage is 58% for age groups 3 and 15, while the lowest percentage is 25% for age groups 6 and 16. The third image is a box plot graph that shows the distribution of ratings by age. The highest ratings are for the age group 24-26, with a median Elo rating of around 2550. The ratings gradually decrease as the age increases. The lowest ratings are for the age group 30-32, with a median Elo rating of around 2450. The fourth image is a histogram that shows the distribution of the number of games played by a group of people. The highest frequency is around 1000 games played, while the lowest frequency is around 4000 games played. The fifth image is a box plot that shows the average ELO rating by age. The highest average ELO rating is for the age group 40-42, with a median Elo rating of around 2550. The ratings gradually increase with age until around age 40, after which they decrease. The lowest average ELO rating is for the age group 80-82, with a median Elo rating of around 2350.

9.2.2. Scatterplot:

- Setting Figure Size using figsize().
- Creating a scatterplot using sns.scatterplot().
- Labeling Axes using xlabel, ylabel.

- Adding a title using plt.title().
- Adding a legend using plt.legend().
- Displaying the plot using plt.show().

Insights from Scatterplot:

This scatter plot graph that shows the relationship between the gross revenue and release year of movies from different countries. The x-axis represents the release year, and the y-axis represents the gross revenue. Each dot represents a movie, and the color of the dot represents the country the movie is from. The plot shows that the average gross revenue of movies increases with release year until around 2015, after which it decreases. The highest gross revenue is for movies from the United States, followed by movies from the United Kingdom and India

9.2.3. Violin plot:

- Setting Figure Size using figsize().
- Creating a violin plot using sns.violinplot().
- Labeling Axes using xlabel, ylabel.
- Rotating x-axes labels using plt.xticks(rotation).
- Displaying the plot using plt.show().

Insights from Violin plot:

The graph shows that the most popular age group for gamers is 18-34 years old (36%), followed by those under 18 years old (24%). The proportion of gamers decreases with age, but even among the oldest age group (65+ years old), 6% of people still play video games and also graph is consistent with other data on the age distribution of gamers. For example, a 2022 survey by the Entertainment Software Association found that the average age of gamers in the United States is 35.4 years old.

Overall, the graph shows that video games are popular with people of all ages, but the most popular age group for gamers is 18-34 years old.

9.3. Multivariate analysis:

Multivariate analysis involves examining the relationships and patterns among multiple variables simultaneously. This can include studying the interplay between various factors such as player ratings, game outcomes, tournament types, and player demographics. Techniques like multivariate regression, factor analysis, or clustering may be employed to uncover complex associations and dependencies within the chess dataset to provide a comprehensive understanding.

9.3.1. Pair Plot:

- Selecting the specified columns.
- Filtering the data frame.
- Setting seaborn runtime configurations for more space.
- Creating a pair plot using sns.pairplot().
- Adjusting layout for more space between plot and axes titles.

Insights from Pair Plot:

The graph is a pair plot of ELO rating and the number of games played by different countries in FIFA. ELO rating is a method for calculating the relative skill levels of players in zero-sum games such as chess. The number of games played is a measure of a country's experience in international competition.Also graph shows a positive correlation between ELO rating and the number of games played. This means that countries that play more games tend to have higher ELO ratings. There are a few outliers on the graph, such as Qatar, which has a very high ELO rating despite having played relatively few games. This is likely due to the fact that Qatar has invested heavily in football in recent years and has a number of world-class players in its national team.

Overall, the graph suggests that the number of games played is a significant factor in a country's ELO rating. However, other factors such as the quality of the players and the investment of the national football federation can also play a role.

9.3.2. Heat Map:

- Selecting the specified columns.
- Calculating correlation matrix.
- Setting Figure Size using plt.figsize().
- Creating a heat map using sns.heatmap().
- Adjusting layout for more space between plot and axes titles.
- Displaying the title using plt.title().
- Displaying the plot using plt.show().

Insights from Heat Map:

The heatmap shows the correlation between the different variables in a dataset. The correlation between two variables is a measure of how strongly they are related to each other. A correlation of 1 indicates a perfect positive correlation, meaning that the two variables are always moving in the same direction. A correlation of -1 indicates a perfect negative correlation, meaning that the two variables are always moving in opposite directions. A correlation of 0 indicates no correlation, meaning that the two variables are not related to each other. The map is colored to represent the strength and direction of the correlation between each pair of variables. Red indicates a strong positive correlation, blue indicates a strong negative correlation, and white indicates no correlation. The darker the color, the stronger the correlation.

Here are some of the key takeaways from the heatmap:

1. The variables "ELO" and "games" are strongly positively correlated, meaning that countries that play more games tend to have higher ELO ratings.
2. The variables "ELO" and "birth year" are weakly negatively correlated, meaning that older countries tend to have slightly lower ELO ratings.
3. The variables "games" and "age" are weakly negatively correlated, meaning that older players tend to play fewer games.
4. The variables "ELO" and "age" are not significantly correlated.

Overall, the heatmap shows that the number of games played is the strongest factor that is correlated with ELO rating. However, other factors such as birth year and age may also play a role.

10. Distributions:

The distribution helps you identify patterns, outliers, and potential issues in the data. It also guides the selection of appropriate statistical tests and models for further analysis.

10.1. Poisson Distribution of rating:

We performed a analysis on a column ('ELO') from a DataFrame ('df') by visualizing its distribution and fitting a Poisson distribution to the data. Here's a step-by-step explanation:

1. We specified the column name ('ELO') for which the Poisson distribution to be analyzed.
2. We created histogram of the selected column ('ELO') using Seaborn. The histogram includes 30 bins, a Kernel Density Estimate (KDE) for smoothness, and is colored in sky blue. The stat='density' parameter normalizes the histogram for better comparison.
3. Calculated the mean ('mu') of the selected column ('ELO'). Then, it defines values for the x-axis ('x_poisson') and computes the Probability Mass Function (PMF) of the Poisson distribution ('p_poisson') based on the mean.
4. Plots the stems of the Poisson distribution on the same plot using the plt.stem function. The distribution is colored in red ('r-') and labeled for the legend.
5. Sets the plot title, x-axis label, and legend. Finally, displays the combined histogram and Poisson distribution plot.

10.2. Exponential Distribution of rating:

The variable ‘column_name’ is left empty in the code snippet. To provide a meaningful explanation, let's assume a specific column, such as 'rating,' is chosen for the Exponential distribution. Here's a step-by-step:

1. We Specified the column name ('rating') for which the Exponential distribution will be analyzed. You should replace this with the actual column name from dataset.

2. Created a histogram of the selected column ('rating') using Seaborn.

The histogram includes 30 bins, a Kernel Density Estimate (KDE) for smoothness, and is colored in sky blue. The stat='density' parameter normalizes the histogram.

3. Calculated the scale parameter for the Exponential distribution based on the inverse of the mean of the selected column ('rating'). It then defines values for the x-axis ('x_exp') and computes the Probability Density Function (PDF) of the Exponential distribution ('p_exp').
4. Plots the Exponential distribution on the same plot using the plt.plot function. The distribution is colored in green ('g-') and labeled for the legend.
5. Sets the plot title, x-axis label, and legend. Finally, displays the combined histogram and Exponential distribution plot.

10.3. Gamma Distribution of rating:

We analysis specified column ('ELO') in a DataFrame ('df') by visualizing its distribution and fitting a Gamma distribution to the data. Here's a step-by-step:

1. Specified the column name ('ELO') for which the Gamma distribution will be analyzed.
2. We created a histogram of the selected column ('ELO') using Seaborn. The histogram includes 30 bins, a Kernel Density Estimate (KDE) for smoothness, and is colored in sky blue. The **stat='density'**

parameter normalizes the histogram for better comparison.

3. Fited a Gamma distribution to the data using the **gamma.fit** function, which returns the parameters 'a' (shape), 'loc' (location), and 'scale' (scale). It then defines values for the x-axis ('x_gamma') and computes the Probability Density Function (PDF) of the Gamma distribution ('p_gamma').
4. Plots the Gamma distribution on the same plot using the **plt.plot** function. The distribution is colored in blue ('b-') and labeled for the legend.

5. Sets the plot title, x-axis label, and legend. Finally, displays the combined histogram and Gamma distribution plot.

10.4. Normal Distribution of rating:

The code analysised on a specified column ('ELO') in a DataFrame ('df') by visualizing its distribution and fitting a Normal (Gaussian) distribution to the data. Here's a step-by-step:

1. Specified the column name ('ELO') for which the Normal distribution will be analyzed.
2. Created a histogram of the selected column ('ELO') using Seaborn. The histogram includes 30 bins, a Kernel Density Estimate (KDE) for smoothness, and is colored in sky blue. The **stat='density'** parameter normalizes the histogram for better comparison.
3. Fited a Normal distribution to the data using the **norm.fit** function, which returns the mean and standard deviation. It then defines values for the x-axis ('x_norm') and computes the Probability Density Function (PDF) of the Normal distribution ('p_norm').
4. Ploted the Normal distribution on the same plot using the **plt.plot** function. The distribution is represented by a red dashed line ('r--') and labeled for the legend.
5. Sets the plot title, x-axis label, and legend. Finally, displays the combined histogram and Normal distribution plot.

10.5. Combination of All Distributions of rating:

The code analyses on a specified column ('ELO') in a DataFrame ('df') by visualizing its distribution and fitting multiple probability distributions (Normal, Exponential, Gamma, and Poisson) to the data. Here's a step-by-step:

1. Specified the column name ('ELO') for which various distribution fits will be analyzed.
2. Created a histogram of the selected column ('ELO') using Seaborn. The histogram includes 30 bins, a Kernel Density Estimate (KDE)

for smoothness, and is colored in sky blue. The **stat='density'** parameter normalizes the histogram for better comparison.

3. Fited a Normal distribution to the data using the **norm.fit** function, defines values for the x-axis ('x_norm'), and computes the Probability Density Function (PDF) of the Normal distribution ('p_norm'). The Normal distribution is plotted as a red dashed line.
4. Fited an Exponential distribution to the data using the inverse of the mean as the scale parameter. It defines values for the x-axis ('x_exp') and computes the PDF of the Exponential distribution ('p_exp'). The Exponential distribution is plotted as a green line.
5. Fits a Gamma distribution to the data using the **gamma.fit** function. It defines values for the x-axis ('x_gamma') and computes the PDF of the Gamma distribution ('p_gamma'). The Gamma distribution is plotted as a blue line.

6. Fits a Poisson distribution to the data using the mean as the parameter. It defines values for the x-axis ('x_poisson') and computes the Probability Mass Function (PMF) of the Poisson distribution ('p_poisson'). The Poisson distribution is plotted as purple stems.

7. Sets the plot title, x-axis label, and legend. Finally, displays the combined histogram and multiple distribution fits plot.

11. Hypothesis Testing of rating:

Hypothesis testing in the context of Exploratory Data Analysis (EDA) typically involves using statistical tests to make inferences about the characteristics of a population based on data. This code performs an independent two-sample t-test to compare the average number of games played by chess players born in two different years, namely 'birth_year1' (1990) and 'birth_year2' (2000). Here's a step-by-step:

1. Specified two birth years, 'birth_year1' and 'birth_year2,' for which the average number of games played will be compared.
2. Filterd the DataFrame ('cleandf') to obtain two groups of chess players: those born in 'birth_year1' and those born in 'birth_year2.' Specifically, it extracts the 'games' column for each group.
3. We conducted an independent two-sample t-test using the **ttest_ind** function from the **scipy.stats** module. The 'equal_var=False'

parameter indicates that the variances of the two groups are not assumed to be equal. The t-test returns the t-statistic and the p-value.

4. Will be prints the results of the t-test, including the t-statistic and the p-value.
5. Sets a significance level (alpha) commonly used in hypothesis testing. In this case, the common choice is 0.05.
6. Finally, tests the null hypothesis that there is no difference in the average number of games played between the two groups. If the p-value is less than the significance level (0.05), the null hypothesis is rejected, indicating a significant difference in the average number of games played between the two birth years. Otherwise, the null hypothesis is not rejected, suggesting no significant difference. The results are printed accordingly.

Output:

T-statistic: 4.664179309622362
P-value: 0.00042923574058235556
Reject the null hypothesis: There is a significant difference in the number of games played between players born in 1990 and 2000.

12. Limitations:

- While conducting the analysis on the FIDE Top 200 Chess Rankings dataset, it's crucial to acknowledge certain limitations:
- The dataset's accuracy relies on the correctness of the provided information.
- Missing or incomplete data points may impact the comprehensiveness of the analysis.
- Changes in player status, rankings, or other attributes over time are not considered in this snapshot.

13.Recommendations:

Based on the analysis, consider the following recommendations:

- Regularly update the dataset to reflect the most current rankings and player information.
- Enhance data quality by addressing missing or incomplete entries.
- Explore incorporating historical data to analyze trends and player

development over time.

14.Conclusion:

The chess federation dataset shows that the number of games played is the strongest factor correlated with ELO rating. This is likely since countries and players that play more games have more experience and are able to improve their skills more quickly. Other factors such as birth year and age may also play a role, but the effect is less pronounced, findings suggest that countries that want to improve their ELO rating should focus on playing more games, both domestically and internationally. Additionally, federations should invest in developing young players so that they can compete at a high level early in their careers.

Here are some specific recommendations for chess federations:

1. Increase the number of domestic tournaments. This will give players more opportunities to compete and improve their skills.
2. Send players to compete in more international tournaments. This will expose players to the best competition in the world and help them to raise their level of play.
3. Invest in developing young players. This can be done through programs such as coaching camps and scholarships.
4. Promote chess in schools and other youth organizations. This will help to increase the number of people playing chess and create a pipeline of talented young players.

By following these recommendations, chess federations can help to improve the level of play in their countries and produce world-class players.

15.References:

- I taken this dataset from Kaggle.
- Chess Wikipedia
- FEDI Chess Wikipedia
- Online chess competions.

16.Acknowledgment:

“GOD HELPS THOSE WHO HELP THEMSELVES.”

“ARISE! AWAKE! AND STOP NOT UNTIL THE GOAL IS REACHED.”

Success often requires preparation, hard work, and perspiration. The path to success is a long journey that calls for tremendous effort with many bitter and sweet experiences. This can only be achieved by the Graceful Blessing from the Almighty on everybody. I want to submit everything beneath the feet of God.

I want to acknowledge my regards to my teacher, Ms. Shivangini Gupta, for her constant support and guidance throughout my training. I would also like to thank HOD Ms. Harjeet Kaur, School of Computer Science and Engineering for introducing such a great program.

I may be failing in my duties if I do not thank my parents for their constant support, suggestion, inspiration and encouragement and best wishes for my success. I am thankful for their supreme sacrifice, eternal benediction, and ocean-like bowls full of love and affection. We extend our gratitude to FIDE for maintaining and providing the dataset. Special thanks to the developers of Python, Pandas, and Matplotlib for their contributions to data analysis and visualization tool.

Presentation link:
https://docs.google.com/presentation/d/1VY7r87RHu2BRmt_9USpx85NtUl1jvOxQ/edit?usp=drive_link&ouid=102928907848134080440&rtpof=true&sd=true

Project link:
https://drive.google.com/file/d/1zZDAA4gvTh3G1ImSjFEprzNR76kbm4LX/view?usp=drive_link

Dataset link:
https://drive.google.com/file/d/1p2z5y3RwKCsf_5klI_4ass1L1yIZmsf4/view?usp=drive_link