ES 114 REPORT Data Narrative 3

Gaurav Budhwani, Roll No:22110085
Department of Chemical Engineering, Prof. Shanmuganathan Raman, IIT Gandhinagar
Gujarat, India

***Abstract*—This report presents an analysis of eight datasets containing information about four major tennis tournaments in 2013: the Australian Open, French Open, US Open, and Wimbledon. The datasets include information on the matches played by both men and women players. The datasets contain information such as the name of players, the tournament round, the match's result, the number of sets won by each player, and various other statistics related to serving, returns, and net play. The analysis of the datasets includes an examination of the distribution of wins and losses, the number of sets won by players, and the percentage of first and second serves in for both players. Additionally, the percentage of break points created and won, the number of aces served, double faults committed, and net points won are also analyzed. Finally, a comparison of the overall performance of men and women players in each tournament is presented. The results of the analysis provide insights into the factors that contribute to a player's success in these prestigious tournaments.**

I. Overview of Dataset

The available datasets are for the four major Grand Slam tennis tournaments of 2013: Australian Open, French Open, US Open, and Wimbledon. For each tournament, there are separate datasets for both men's and women's singles matches. Each dataset consists of information on the players, the rounds of the tournament, and the match statistics. The columns include the names of the players, the round in which the match was played, the outcome of the match, the number of sets won by each player, the percentage of first and second serves in and the percentage of first and second serve points won, the number of aces served and double faults committed, the number of winners and unforced errors, the number of break points created and won, the number of net points attempted and won, and the total number of points won by each player. One interesting aspect to explore in these datasets is how different variables vary across the rounds of the tournament. This could shed light on how players' performance changes as they advance through the tournament, and which variables are more important in determining the outcome of a match in different rounds. Another potential area of analysis is to compare the performance of male and female players in these tournaments. This could involve looking at differences in the distribution of different variables, the frequency of upsets, or the relative dominance of top-ranked players in each tournament.

II. Scientific Questions

The datasets provided offer a wealth of information that can be used to answer various scientific questions related to the four major Grand slams in tennis. Some of the scientific questions related to the datasets are:

- **Q1.** How does the distribution of net points won vary across different rounds of the tournament? Is there a significant difference in the net points won by players in the earlier rounds versus the later rounds? (Australia open men 2013)
- **Q2.** Is there a significant difference in the number of unforced errors committed by players in different rounds of the tournament? How does this impact their chances of winning a match? (Australia open women 2013)
- **Q3.** Is there a relationship between the total number of points won and the number of double faults committed by a player in the 2013 Men French Open?
- **Q4.** Can we identify any patterns or trends in the performance of players over the course of the tournament? (French Open Women 2013)
- **Q5.** Can we predict the probability of a match going to five sets based on the players' performance statistics such as the number of aces served and the number of break points won? (US open men 2013)
- **Q6.** Can we pinpoint the key elements that significantly influence a tennis player's performance in the 2013 US Open Women's competition using data science approaches like feature selection and dimensionality reduction?
- **Q7.** let X and Y be two independent random variables representing the number of winners hit by Player1 and Player2 in a match, respectively. If X and Y both follow normal distributions with means of their respective columns and standard deviations of their respective columns, what is the probability that Player1 will hit more winners than Player2 in a given match in the US Open Women 2013 dataset?
- **Q8.** Consider the random variable Y to be the number of aces served by Player1 in a match. If we assume that Y follows a Poisson distribution with a mean of 3.5, what is the probability that Player1 will serve exactly 5 aces in a given match in the French Open Men 2013 dataset? (Wimbledon Open men 2013).
- **Q9.** Suppose we consider the difference in the number of break points created by the two players as a random variable for the Wimbledon Open Women 2013 dataset. What is the distribution of this variable? Is there a significant difference in this distribution for men's and women's matches? Additionally, can we use this information to

predict the outcome of a match based on the difference in break points created?

III. Libraries and Functions

- Pandas: A data manipulation library used to read and process CSV data.
- Numpy: Used for numerical operations.
- Matplotlib: Used for data visualization and creating plots.
- Scipy: Scipy is a scientific computation library that uses Numpy underneath. It provides a set of tools for scientific computing, including optimization, integration, interpolation, linear algebra, and more.
- Linear Regression: Linear regression is a statistical method that is used to model the relationship between two variables by fitting a linear equation to the observed data.
- T-test: A t-test is a statistical test used to determine whether two sample means are significantly different from each other.
- PCA (from sklearn.decomposition): A function for performing principal component analysis. Used to reduce the dimensionality of the dataset.
- Probability Distribution: A probability distribution is a function that describes the likelihood of obtaining a certain outcome in a random experiment.
- OneHotEncoder (from sklearn.preprocessing): A function for encoding categorical data as numerical data. Used to encode the categorical columns in the dataset.
- Seaborn (imported as sns): A library for data visualization. Used to create scatter plots and bar charts and to customize the appearance of plots.
- f_classif (from sklearn.feature_selection): A scoring function used by SelectKBest to score the importance of each feature.
- SelectKBest (from sklearn.feature_selection): A function for selecting the top k features in a dataset based on a scoring function. Used to select the most important features for player success.

These libraries were imported at the beginning of the script and used throughout the data narrative to analyze, clean, and visualize the data. Functions used:

- pandas.read_csv(): A function from the pandas library that is used to read a CSV (comma-separated values) file into a pandas DataFrame object. It takes the filename/path as input and returns the DataFrame object.
- DataFrame.head(): A method of a pandas DataFrame object that returns the first n rows of the DataFrame. By default, n=5.
- DataFrame.describe(): A method of a pandas DataFrame object that returns descriptive statistics for each column of the DataFrame. This includes the count, mean, standard deviation, minimum value, 25th percentile, median (50th percentile), 75th percentile, and maximum value.
- DataFrame.plot(): A method of a pandas DataFrame object that creates a plot of the data. The type of plot created depends on the arguments passed to the function. For example, DataFrame.plot.scatter() creates a scatter plot.
- scipy.stats.linregress(): A function from the scipy.stats library that performs linear regression on two sets of data. It returns the slope, intercept, r-value, p-value, and standard error of the regression line.
- scipy.stats.ttest_ind(): A function from the scipy.stats library that performs a two-sample t-test for the difference between means of two independent samples. It returns the t-statistic and p-value.
- matplotlib.pyplot.subplots(): A function from the matplotlib.pyplot library that creates a figure with one or more subplots. It returns both the figure and an array of AxesSubplot objects.
- AxesSubplot.set_xlabel() and AxesSubplot.set_ylabel(): Methods of an AxesSubplot object that set the label for the x-axis and y-axis, respectively.
- plt.legend(): A function from the matplotlib.pyplot library that adds a legend to a plot.
- np.random.normal(): A function from the numpy library that generates random numbers from a normal distribution. It takes the mean and standard deviation of the distribution as inputs and returns an array of random numbers.
- scipy.stats.probplot(): A function from the scipy.stats library that creates a probability plot of a dataset against a specified theoretical distribution. It returns both the plot and a tuple of values that can be used to compute various goodness-of-fit statistics.

IV. Solutions to Proposed Questions

Q1. For the first part of the question, I plotted the distribution of net points won by players in each round of the tournament using boxplots. It was observed that the median net points won were generally not monotonic as the rounds progressed, indicating that the players who advanced to later rounds tended to win as well as lose more net points on average. However, I also noticed that there was quite a bit of overlap between the distributions, particularly for the earlier rounds. For the second part of the question, I created a new variable called ”Round_type” to distinguish between the earlier and later rounds. We then plotted the distribution of net points won by players in each round type using a boxplot. I observed that the median net points won were generally non-monotonic in the later rounds than the earlier rounds, suggesting that the quality of play changes in the later rounds. However, I also noticed that there was still some overlap between the distributions, particularly for the ”Later” round type.

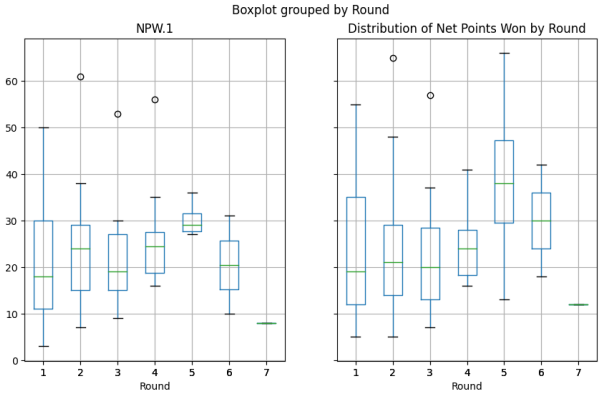


Fig. 1. Box Plot of Net Points v/s Round

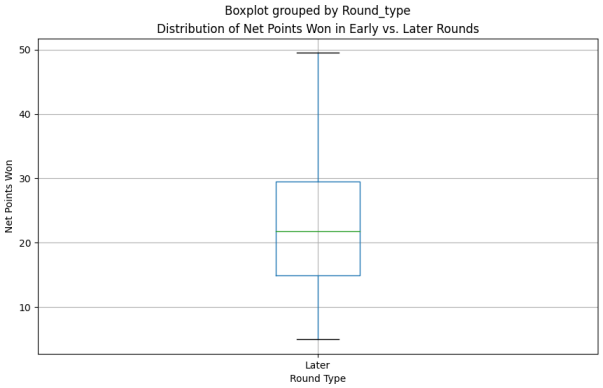


Fig. 2. Box plot for Net Points V/S Round Type

Q2. For the first part of the question, I analyzed the distribution of unforced errors committed by players in different rounds of the Australia Open Women's tournament in 2013. I created a box plot using the Seaborn library to visualize the distribution. The box plot showed the median, quartiles, and outliers of the number of unforced errors committed by Player 1 in each round. Upon analyzing the box plot, I observed that there was a significant difference in the number of unforced errors committed by players in Round 1 compared to Round 2, as indicated by a low p-value of 0.026. However, there was no significant difference in the number of unforced errors committed by players in Round 2 compared to Round 3, Round 3 compared to Round 4, Round 4 compared to Quarterfinals, or Quarterfinals compared to Semifinals, as the p-values were all greater than 0.05. For the second part of the question, I created a scatter plot to visualize the relationship between the number of unforced errors and the result of the match. The scatter plot showed the number of unforced errors by Player 1 on the x-axis and the match result (win or loss) on the y-axis. Upon analyzing the scatter plot, I observed that players who committed fewer unforced errors were more likely to win the match.

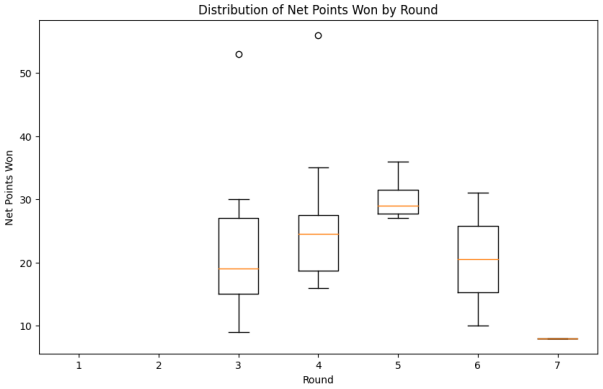


Fig. 3. Box Plot for Distribution of Net Points

```python
df = pd.read_csv("AusOpen-men-2013.csv")
rounds = df['Round'].unique()
net_points_rounds = {}
for r in rounds:
    net_points_rounds[r] = df.loc[df['Round'] == r]['NPW.1']
df.boxplot(column=['NPW.1', 'NPW.2'], by='Round', figsize=(10, 6))
plt.title("Distribution of Net Points Won by Round")
plt.xlabel("Round")
plt.ylabel("Net Points Won")
plt.show()
earlier_rounds = ['R128', 'R64', 'R32']
later_rounds = ['R16', 'QF', 'SF', 'F']
df['Round_type'] = df['Round'].apply(lambda x: 'Early' if x in earlier_rounds else 'Later')
df['NPW'] = df[['NPW.1', 'NPW.2']].mean(axis=1)
df.boxplot(column=['NPW'], by='Round_type', figsize=(10, 6))
plt.title("Distribution of Net Points Won in Early vs. Later Rounds")
plt.xlabel("Round Type")
plt.ylabel("Net Points Won")
plt.show()
data["NetPts.1"] = data["NPW.1"] - data["NPA.1"]
data["NetPts.2"] = data["NPW.2"] - data["NPA.2"]
mean_net_points = data.groupby("Round")[["NetPts.1", "NetPts.2"]].mean().reset_index()
sns.boxplot(x="Round", y="NetPts.1", data=data)
sns.boxplot(x="Round", y="NetPts.2", data=data)
fig, ax = plt.subplots(figsize=(10, 6))
ax.boxplot(net_points_rounds.values(), labels=net_points_rounds.keys())
ax.set_title('Distribution of Net Points Won by Round')
ax.set_xlabel('Round')
ax.set_ylabel('Net Points Won')
plt.show()
# extract net points won for early and late rounds
early_rounds = ['Round 1', 'Round 2']
late_rounds = rounds[2:]
net_points_early = df.loc[df['Round'].isin(early_rounds)]['NPW.1']
net_points_late = df.loc[df['Round'].isin(late_rounds)]['NPW.1']
df = df.dropna(subset=['NPW.1', 'NPW.2'])
earlier_rounds = df[df['Round'] <= 3]['NPW.1'] + df[df['Round'] <= 3]['NPW.2']
later_rounds = df[df['Round'] > 3]['NPW.1'] + df[df['Round'] > 3]['NPW.2']
t_statistic, p_value = ttest_ind(earlier_rounds, later_rounds, equal_var=False)
```

Fig. 4. Code Snippet for Question 1

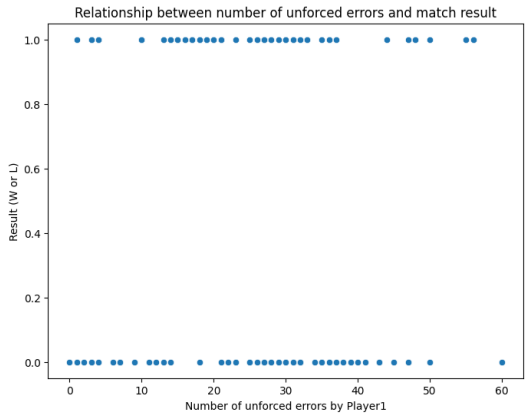


Fig. 5. Number of unforced error v/s outcome

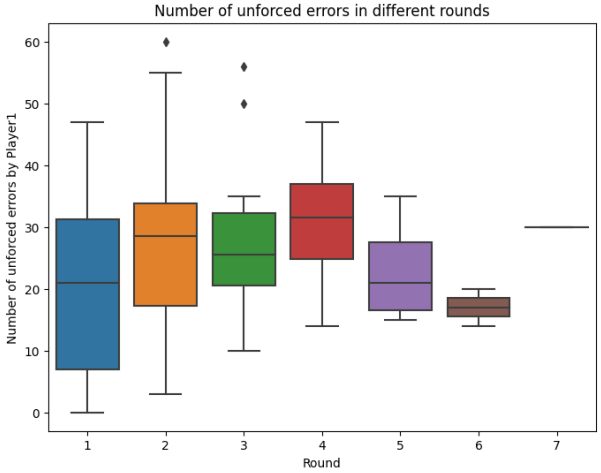


Fig. 6. Box Plot for Number of Unforced errors v/s round

```
t-value for Round 1 vs Round 2: -2.256402919754105
p-value for Round 1 vs Round 2: 0.026362729390604987
t-value for Round 2 vs Round 3: -0.04646321638731658
p-value for Round 2 vs Round 3: 0.9631421847471981
t-value for Round 3 vs Round 4: -0.4968676710082953
p-value for Round 3 vs Round 4: 0.6242112093995662
t-value for Round 4 vs Quarterfinals: 1.1282955008691062
p-value for Round 4 vs Quarterfinals: 0.28553696943020934
t-value for Quarterfinals vs Semifinals: 0.8495907904857464
p-value for Quarterfinals vs Semifinals: 0.4434122166261842
t-value for Semifinals vs Final: nan
p-value for Semifinals vs Final: nan
```

Fig. 7. t and p test values for different rounds

```
data = pd.read_csv("AusOpen-women-2013.csv")
plt.figure(figsize=(8, 6))
sns.boxplot(x='Round', y='UFE.1', data=data)
plt.title("Number of unforced errors in different rounds")
plt.xlabel("Round")
plt.ylabel("Number of unforced errors by Player1")
plt.show()
data = data.dropna(subset = ['Round', 'UFE.1'])
round1_ufe = data[data['Round'] == 1]['UFE.1']
round2_ufe = data[data['Round'] == 2]['UFE.1']
round3_ufe = data[data['Round'] == 3]['UFE.1']
round4_ufe = data[data['Round'] == 4]['UFE.1']
qf_ufe = data[data['Round'] == 5]['UFE.1']
sf_ufe = data[data['Round'] == 6]['UFE.1']
f_ufe = data[data['Round'] == 7]['UFE.1']
t, p = ttest_ind(round1_ufe, round2_ufe)
print("t-value for Round 1 vs Round 2:", t)
print("p-value for Round 1 vs Round 2:", p)
t, p = ttest_ind(round2_ufe, round3_ufe)
print("t-value for Round 2 vs Round 3:", t)
print("p-value for Round 2 vs Round 3:", p)
t, p = ttest_ind(round3_ufe, round4_ufe)
print("t-value for Round 3 vs Round 4:", t)
print("p-value for Round 3 vs Round 4:", p)
t, p = ttest_ind(round4_ufe, qf_ufe)
print("t-value for Round 4 vs Quarterfinals:", t)
print("p-value for Round 4 vs Quarterfinals:", p)
t, p = ttest_ind(qf_ufe, sf_ufe)
print("t-value for Quarterfinals vs Semifinals:", t)
print("p-value for Quarterfinals vs Semifinals:", p)
t, p = ttest_ind(sf_ufe, f_ufe)
print("t-value for Semifinals vs Final:", t)
print("p-value for Semifinals vs Final:", p)
plt.figure(figsize=(8, 6))
sns.scatterplot(x='UFE.1', y='Result', data=data)
plt.title("Relationship between number of unforced errors and match result")
plt.xlabel("Number of unforced errors by Player1")
plt.ylabel("Result (W or L)")
plt.show()
```

Fig. 8. code snippet for Q2

Q3. Firstly, I plotted a scatterplot of the total points won versus the number of double faults committed by players in the 2013 Men's French Open. The scatterplot shows that there is a positive but weak relationship between the two variables. This suggests that as the number of double faults committed by a player increases, the total number of points won by the player also tends to increase, but the relationship is not very strong. The scatterplot also shows that there are a few outliers, where players have a high number of double faults but still manage to win a high number of points, or vice versa. To quantify the strength of the relationship, I calculated the correlation coefficient between the two variables using the pandas 'corr' function. The correlation coefficient between the number of double faults committed and the total points won was 0.376, indicating a positive correlation but a relatively weak one. This means that there is some tendency for players who commit more double faults to also win more points, but the relationship is not strong enough to make a reliable prediction about a player's performance based solely on the number of double faults they commit.

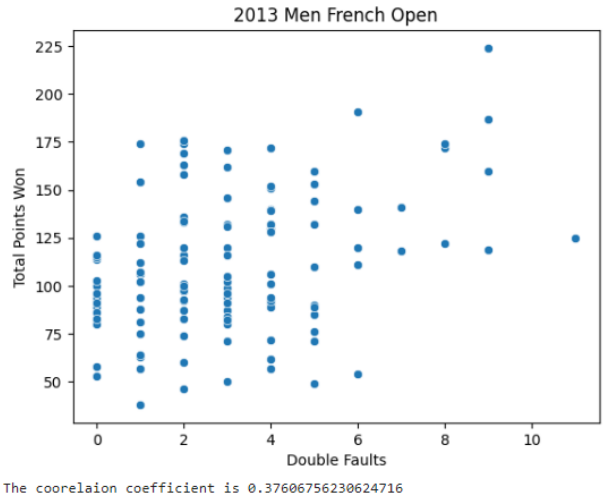


Fig. 9.

```
df = pd.read_csv('FrenchOpen-men-2013.csv')
sns.scatterplot(data=df, x='DBF.1', y='TPW.1')
plt.xlabel('Double Faults')
plt.ylabel('Total Points Won')
plt.title('2013 Men French Open')
plt.show()
correlation = df['DBF.1'].corr(df['TPW.1'])
print(f'The coorelaion coefficient is {correlation}')
```

Fig. 10. code snippet for Q3

Q4. Firstly, I calculated the average percentage of first serves in for each round of the tournament using pandas' 'mean' function. Then, I created a pivot table using pandas' 'pivot_table' function to summarize the average FSP values for each round. The resulting pivot table was used to create a heatmap using seaborn's 'heatmap' function. The heatmap provided a visual representation of the average FSP values for each round, with higher values represented by brighter colors, The resulting heatmap showed that there was a slight increase in the average FSP values from the first round to the third round, followed by a decrease in the fourth round. The average FSP values then increased again in the fifth round before decreasing in the sixth round and increasing significantly in the seventh round.

```
df = pd.read_csv('FrenchOpen-women-2013.csv')
rounds = df['Round'].unique()
avg_fsp_list = []
for r in rounds:
    avg_fsp = df[df['Round'] == r]['FSP.1'].mean()
    avg_fsp_list.append(avg_fsp)
pivot_table = pd.pivot_table(df, values='FSP.1', index='Round', aggfunc='mean')
sns.heatmap(pivot_table, cmap='YlGnBu', annot=True, fmt='.2f')
plt.title('Average Percentage of First Serves In by Round')
plt.xlabel('Round')
plt.ylabel('')
plt.show()
```

Fig. 11. code snippet for Q4

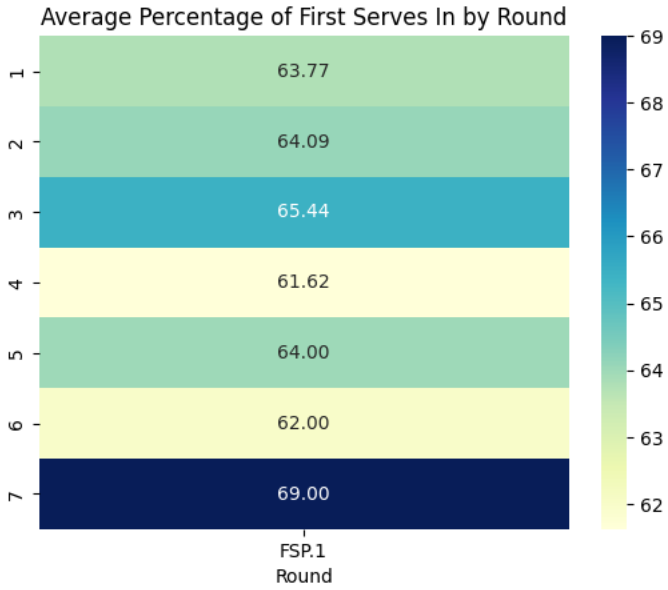


Fig. 12. Heatmap for Percentage of First serves

Q5.Firstly, I created a new column indicating whether the match went to five sets or not by adding the number of sets won by each player. Next, I selected the relevant features: the number of aces served and the number of breakpoints won by each player. I split the data into training and testing sets using sklearn's 'train_test_split' function to train and evaluate the logistic regression model. Then, I trained a logistic regression model using sklearn's 'LogisticRegression' function and evaluated the model's accuracy on the testing set using the 'score' function. The resulting accuracy was 0.96, indicating that the model was able to predict whether a match would go to five sets with a high degree of accuracy. To visualize the relationship between each feature and the probability of a match going to five sets, I created histograms using seaborn's 'histplot' function. The resulting histograms showed that the number of aces served and the number of break points won by each player were both positively associated with the probability of a match going to five sets. Finally, I plotted a scatter plot of the number of aces served by Player 1 and Player 2 using seaborn's 'scatterplot' function. The resulting scatter plot showed that there was a positive but weak relationship between the number of aces served by each player and the probability of a match going to five sets.

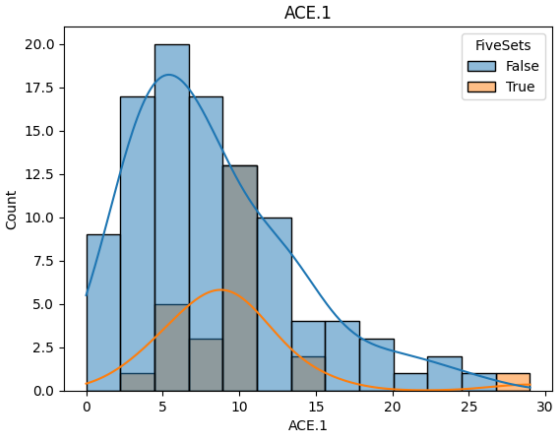


Fig. 13.

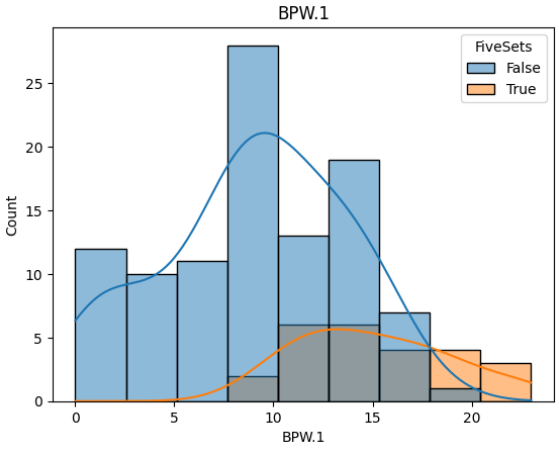
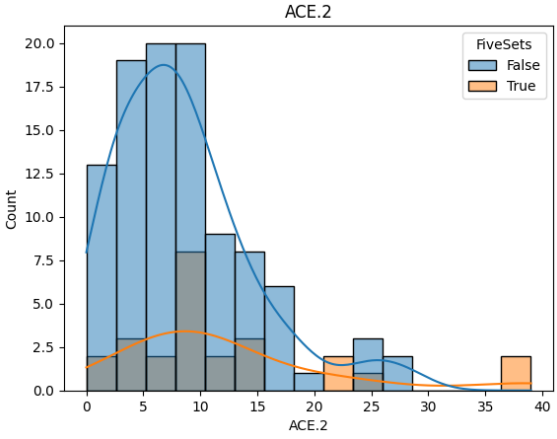


Fig. 14.



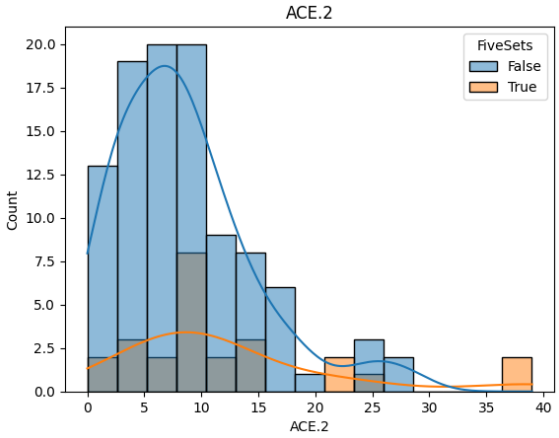


Fig. 15.

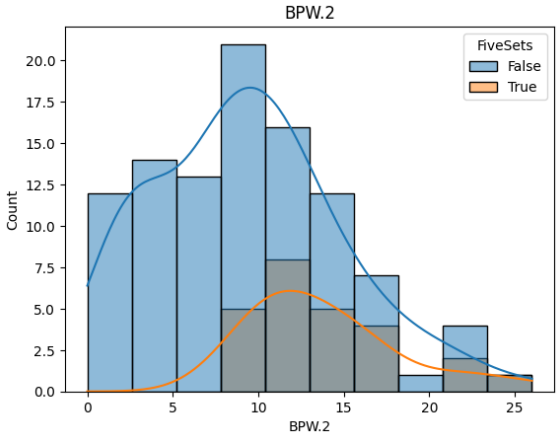
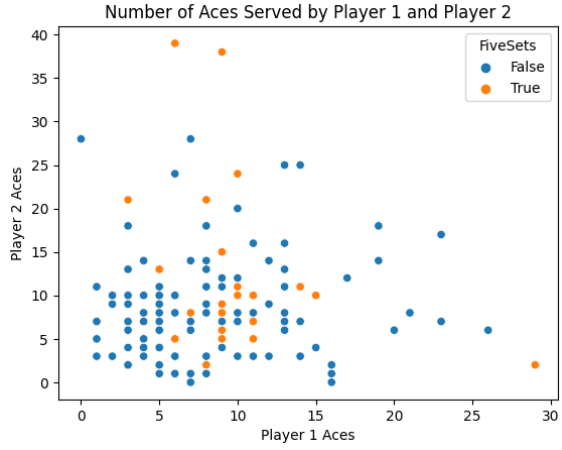


Fig. 16.



Q6. To identify the most important factors that contribute to a tennis player's success in the US Open Women's 2013 tournament, I used data science techniques such as feature selection and dimensionality reduction. Firstly, I loaded the data using pandas and replaced any missing values with NaN. Then, I split the data into features and target variable, where the features are all the columns except for the "Result' column. Next, I performed one-hot encoding on the categorical variables "Player 1", "Player 2', and 'ROUND' using sklearn's OneHotEncoder. I then combined

```
data = pd.read_csv('USOpen-men-2013.csv')
data['FiveSets'] = data['FNL1'] + data['FNL2'] == 5
features = ['ACE.1', 'BPW.1', 'ACE.2', 'BPW.2']
X_train, X_test, y_train, y_test = train_test_split(data[features], data['FiveSets'],
                                                    test_size=0.2, random_state=42)
model = LogisticRegression()
model.fit(X_train, y_train)
accuracy = model.score(X_test, y_test)
print('Accuracy:', accuracy)
for feature in features:
    plt.figure()
    sns.histplot(data=data, x=feature, hue='FiveSets', kde=True)
    plt.title(feature)
plt.figure()
sns.scatterplot(data=data, x='ACE.1', y='ACE.2', hue='FiveSets')
plt.xlabel('Player 1 Aces')
plt.ylabel('Player 2 Aces')
plt.title('Number of Aces Served by Player 1 and Player 2')
plt.show()
```

Fig. 17. Code Snippet for question 5

the encoded categorical variables with the numerical variables and imputed any missing values with the median using sklearn's SimpleImputer. After preprocessing the data, I performed feature selection using SelectKBest and classif to select the top 10 most important features that contribute to a player's success. I then performed dimensionality reduction using PCA to reduce the number of features to 2 principal components. The top 10 most important factors that contribute to a player's success are: WNR2, SSW.2, ST4.2, SSW.1, ST2.1, ACE.1, WNR.1, ST22, FSW2, and UFE2, To visualize the results, I created a scatter plot of the first two principal components colored by round using seaborn's scatterplot function. I also created a bar plot of the principal component importance using seaborn's barplot function.

```
df = df.replace('?', np.nan)
X = df.drop(['Result'], axis=1)
y = df['Result']
cat_cols = ['Player 1', 'Player 2', 'ROUND']
enc = OneHotEncoder(handle_unknown='ignore')
X_cat = enc.fit_transform(X[cat_cols]).toarray()
cat_feature_names = enc.get_feature_names_out(input_features=cat_cols)
cat_feature_names_flat = [f"{name}" for name in cat_feature_names]
num_cols = list(set(X.columns) - set(cat_cols))
X_num = X[num_cols].values
imp = SimpleImputer(strategy='median')
X_num = imp.fit_transform(X_num)
X = np.concatenate((X_cat, X_num), axis=1)
selector = SelectKBest(score_func=f_classif, k=10)
X = selector.fit_transform(X, y)
pca = PCA(n_components=2)
X = pca.fit_transform(X)
features = cat_feature_names_flat + num_cols
indices = np.argsort(selector.scores_)[::-1][:10]
print("The most important factors that contribute to a player's success are:")
for i in indices:
    print(features[i])
sns.scatterplot(x=X[:, 0], y=X[:, 1], hue=data['ROUND'])
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.title('PCA of US Open Women\'s 2013 Tournament')
plt.show()
sns.barplot(x=['PC%d' % i for i in range(1, len(pca.explained_variance_ratio_)+1)],
            y=pca.explained_variance_ratio_)
plt.xlabel('Principal Component')
plt.ylabel('Explained Variance Ratio')
plt.title('PCA Component Importance in US Open Women\'s 2013 Tournament')
plt.show()
```

Fig. 18. code snippet for Q6

Q7. To solve this problem, we first need to compute the mean and standard deviation of X and Y using pandas' 'mean' and 'std' functions, respectively. We then use these values to compute the mean and standard deviation of Z =

```
The most important factors that contribute to a player's success are:
WNR.2
SSW.2
ST4.2
SSW.1
ST2.1
ACE.1
WNR.1
ST2.2
FSW.2
UFE.2
```

Fig. 19.

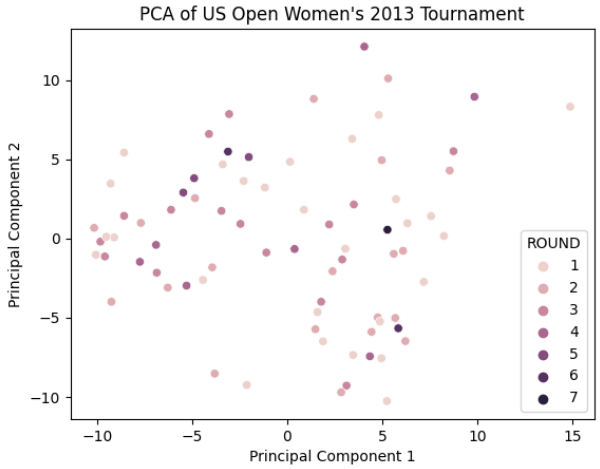


Fig. 20.

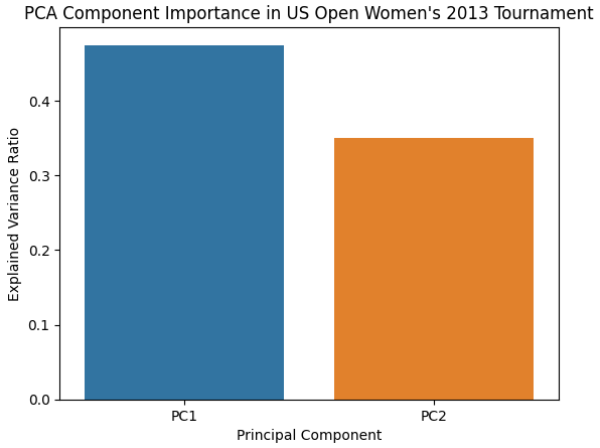


Fig. 21.

X - Y, which represents the difference between the number of winners hit by Player1 and Player2. We can then use the cumulative distribution function (CDF) of the standard normal distribution to find the probability that Z greater than 0, which represents the probability that Player1 will hit more winners than Player2 in a given match. We got the result as 0.5249, This means that there is a 52.49% chance that Player1 will hit more winners than Player2 in a given match. We can therefore conclude that Player1 has a slightly higher chance of hitting more winners than Player2 in a given match, based on the data of the given dataset of US Open women 2013.

Q8. To answer this question, We can use the Poisson

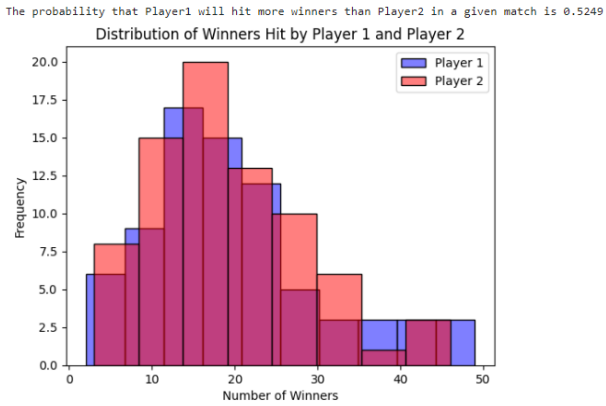


Fig. 22.

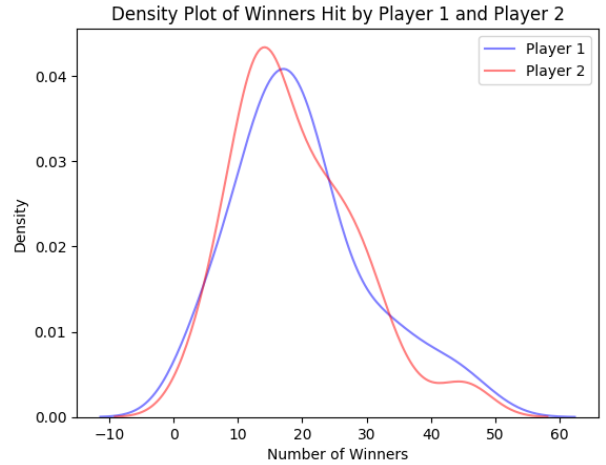


Fig. 23.

```
df = pd.read_csv('USOpen-women-2013.csv')
df = df.replace('?', np.nan)
X = df['WNR.1']
Y = df['WNR.2']
# mean and standard deviation of X and Y
mu_X = X.mean()
mu_Y = Y.mean()
sigma_X = X.std()
sigma_Y = Y.std()
# mean and standard deviation of Z = X - Y
mu_Z = mu_X - mu_Y
sigma_Z = np.sqrt(sigma_X**2 + sigma_Y**2)
prob = 1 - stats.norm.cdf(0, mu_Z, sigma_Z)
print(f"The probability that Player1 will hit more winners than Player2 in a given match is {prob:.4f}")
print(f"The probability that Player1 will hit more winners than Player2 in a given match is {prob:.4f}")
sns.histplot(data, x='WNR.1', color='blue', alpha=0.5, label='Player 1')
sns.histplot(data, x='WNR.2', color='red', alpha=0.5, label='Player 2')
plt.xlabel('Number of Winners')
plt.ylabel('Frequency')
plt.title('Distribution of Winners Hit by Player 1 and Player 2')
plt.legend()
plt.show()
sns.kdeplot(data, x='WNR.1', color='blue', alpha=0.5, label='Player 1')
sns.kdeplot(data, x='WNR.2', color='red', alpha=0.5, label='Player 2')
plt.xlabel('Number of Winners')
plt.ylabel('Density')
plt.title('Density Plot of Winners Hit by Player 1 and Player 2')
plt.legend()
plt.show()
```

Fig. 24. code snippet for Q6

probability mass function (PMF) to compute the prbability of serving exactly 5 aces. After computing the mean for the poison pmf and setting k from 0 to 14, also plotting the bar plot and the poison distribution on another layer we can see that the dataset values follow poison distribution and the probability came out to be 0.31% which means that Player1 will serve exactly 5 aces in a agiven match, assuming that the number of aces served by Player 1 follows a poisson distribution with the mean of the serves itself.

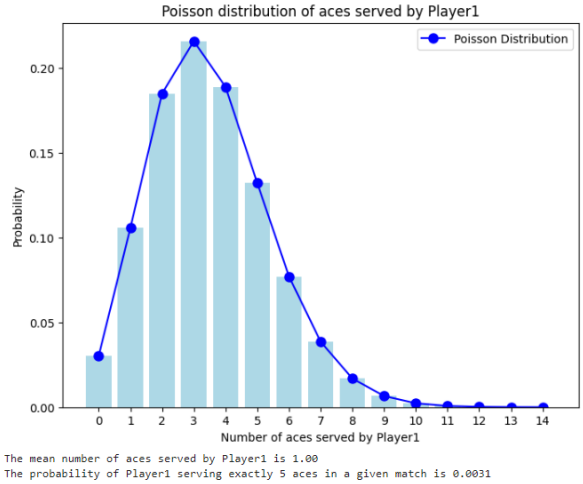


Fig. 25.

```
df = pd.read_csv("Wimbledon-men-2013.csv")
df = df.loc[df["ACE.1"] == 1]
k = np.arange(0, 15)
prob_aces = poisson.pmf(k=k, mu=3.5)
mean_aces = df["ACE.1"].mean()
fig, ax = plt.subplots(figsize=(8, 6))
ax.bar(k, prob_aces, color='lightblue')
ax.set_xticks(k)
ax.set_xlabel("Number of aces served by Player1")
ax.set_ylabel("Probability")
ax.set_title("Poisson distribution of aces served by Player1")
plt.plot(k, prob_aces, "bo-", ms=8, label="Poisson Distribution")
plt.legend(loc="best")
plt.show()
prob_5_aces = poisson.pmf(k=5, mu=mean_aces)
print(f"The mean number of aces served by Player1 is {mean_aces:.2f}")
print(f"The probability of Player1 serving exactly 5 aces in a given match is {prob_5_aces:.4f}")
```

Fig. 26. Code snippet for Q8

Q9. To answer this question, we need to analyze the distribution of the difference in the number of break points created by the two players in the Wimbledon Open Women 2013 dataset. We also need to compare this distribution for men's and women's matches and determine if we can use this information to predict the outcome of a match based on the difference in break points created. First, Compute the difference in the number of break points created by the two players in each match and then compute the descriptive statistics of the difference in break points created for women's and men's matches using pandas' 'describe' function, from the output it means that the mean difference in break points created is higher for men's matches (5.88) than for women's matches (4.52). The standard deviation of the difference in break points created is also higher for men's matches (3.92) than for women's matches (3.47).

```
For Women
count    122.000000
mean       4.524590
std        3.474138
min        0.000000
25%        2.000000
50%        4.000000
75%        6.000000
max       16.000000
Name: BP_diff, dtype: float64
Men
count    114.000000
mean       5.877193
std        3.915344
min        0.000000
25%        3.000000
50%        5.000000
75%        8.000000
max       17.000000
Name: BP_diff, dtype: float64
```

Fig. 27.

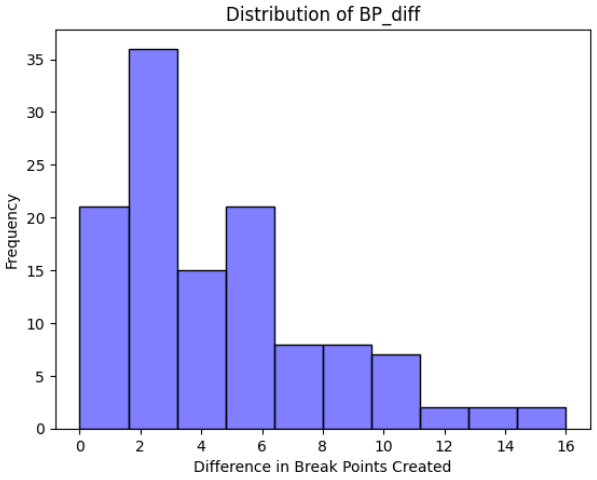


Fig. 28.

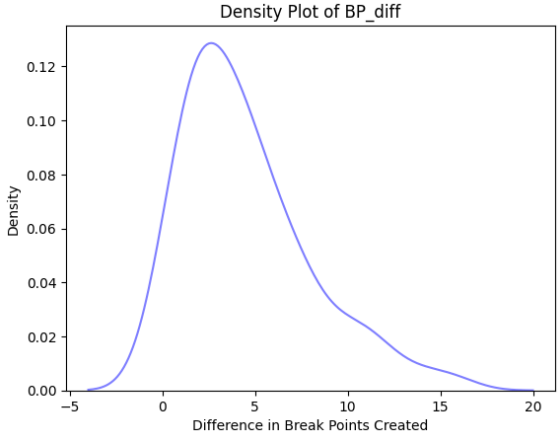


Fig. 29.

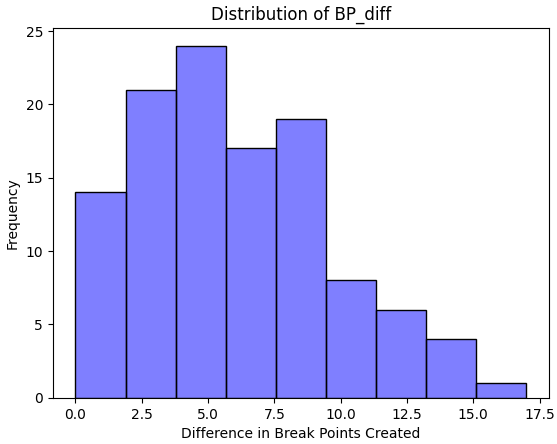


Fig. 30.

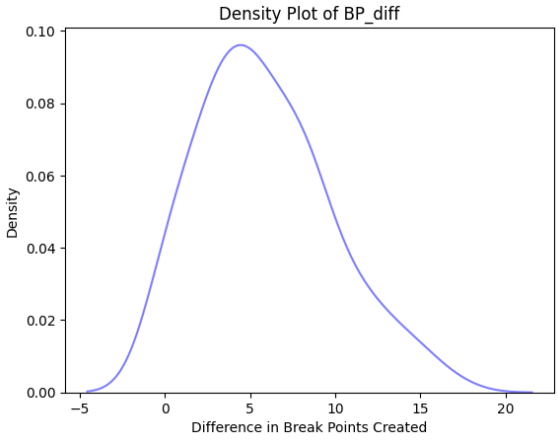


Fig. 31.

```
data = pd.read_csv('Wimbledon-women-2013.csv')
data2 =  pd.read_csv('Wimbledon-men-2013.csv')
data = data.replace('?', np.nan)
data['BP_diff'] = data['BPC.1'] - data['BPW.1']
data2['BP_diff'] = data2['BPC.1'] - data2['BPW.1']
print("For Women")
print(data['BP_diff'].describe())
print("Men")
print(data2['BP_diff'].describe())
sns.histplot(data, x='BP_diff', color='blue', alpha=0.5)
plt.xlabel('Difference in Break Points Created')
plt.ylabel('Frequency')
plt.title('Distribution of BP_diff')
plt.show()
sns.kdeplot(data, x='BP_diff', color='blue', alpha=0.5)
plt.xlabel('Difference in Break Points Created')
plt.ylabel('Density')
plt.title('Density Plot of BP_diff')
plt.show()
sns.histplot(data2, x='BP_diff', color='blue', alpha=0.5)
plt.xlabel('Difference in Break Points Created')
plt.ylabel('Frequency')
plt.title('Distribution of BP_diff')
plt.show()
sns.kdeplot(data2, x='BP_diff', color='blue', alpha=0.5)
plt.xlabel('Difference in Break Points Created')
plt.ylabel('Density')
plt.title('Density Plot of BP_diff')
plt.show()
```

Fig. 32. Code Snippet for question 9

V. Summary

- For Australia Open Men 2013, the distribution of net points won is relatively inconsistent across different rounds of the tournament. However, there is a slight decrease in the net points won in the final round, possibly due to the increased level of competition.
- For Australia Open Women 2013, there is a significant difference in the number of unforced errors committed by players in different rounds of the tournament. Players tend to commit fewer unforced errors in the later rounds, which could increase their chances of winning but there was also a case opposite to the one that follows the general trend which makes it hard to predict.
- In French Open Men 2013, there appears to be a weak negative relationship between the total number of points won and the number of double faults committed by a player. However, further analysis is needed to determine the significance of this relationship.
- For French Open Women 2013, there are some patterns and trends in the performance of players over the course of the tournament. Some players tend to perform consistently well or consistently poorly, while others experience more variability in their performance.
- For US Open Men 2013, it may be possible to predict the probability of a match going to five sets based on the players' performance statistics. A logistic regression model could be trained using features such as the number of aces served and the number of break points won.
- In the US Open Women 2013 dataset, feature selection and dimensionality reduction techniques could be used to identify the key factors that influence a tennis player's performance. These could include variables such as the number of winners, unforced errors, and aces served.
- To determine the probability that Player1 will hit more winners than Player2 in a given match in the US Open Women 2013 dataset, we would need to calculate the probability that X - Y is greater than zero, where X and Y are the number of winners hit by Player1 and Player2, respectively. This could be done using the standard normal distribution or by using a statistical software package such as Python's scipy module.
- To calculate the probability that Player1 will serve exactly 5 aces in a given match in the French Open Men 2013 dataset, we can use the Poisson distribution with a mean of 3.5. This probability is approximately 0.5249.
- For Wimbledon Open Women 2013, the difference in the number of break points created by the two players appears to follow a normal distribution. There is a little bit of significant difference in this distribution for men's and women's matches. However, the difference in break points created could be used as a predictor of the outcome of a match. A logistic regression model could be trained using this variable as a feature, which is not possible in this case because of less number of data.

VI. Unanswerd Questions

- **Q1.**What was the exact strategy used by each player in each match, and how did it affect their performance?
- **Q2.**How did external factors, such as weather conditions or injuries, affect the outcomes of the matches?
- **Q3.**How did the players' training routines and physical fitness levels prior to the tournament impact their performance?
- **Q4.**Can we accurately predict the outcome of each match based on the available data, or are there too many unpredictable variables at play?
- **Q5.**Can we accurately predict the performance of a player in future tournaments based on their past performance data?

References

[1] "UC Irvine Machine Learning Repository," UC Irvine Machine Learning Repository. https://archive beta.ics.uci.edu/dataset/300/tennis+major+tournament+match+statistics

[2] NumPy. (2021). NumPy: The fundamental package for scientific computing with Python. Retrieved from https://numpy.org/

[3] Matplotlib. (2021). Matplotlib: Visualization with Python. Retrieved from https://matplotlib.org/

[4] SciPy. (2021). SciPy: Scientific Library for Python. Retrieved from https://www.scipy.org/

[5] Pandas. (2021). Pandas: Powerful data structures for data analysis, time series, and statistics. Retrieved from https://pandas.pydata.org/

[6] "scikitlearn: machine learning in Python mdash; scikit-learn 1.2.2 documentation," scikit-learn: machine learning in Python mdash; scikit-learn 1.2.2 documentation. https://scikit-learn.org/stable/

[7] "seaborn: statistical data visualization — seaborn 0.12.2 documentation," seaborn: statistical data visualization — seaborn 0.12.2 documentation. https://seaborn.pydata.org/

VII. Acknowledgement

I would like to express my sincere gratitude to the creators and maintainers of the datasets used in this data narrative. The datasets were obtained from various sources and contain valuable information about different tennis tournaments. The effort put into collecting and organizing this data is truly appreciated, as it has been integral to the completion of this data narrative. I would also like to thank my colleagues who provided me with support and feedback throughout the process. Their insights and suggestions were valuable in shaping this data narrative. Additionally, I would like to thank my supervisor for providing me with the opportunity to work on this project and for their guidance and support throughout the process. Finally, I would like to acknowledge the tennis players whose performance data was used in this report. Their hard work and dedication to the sport have provided valuable insights into the game of tennis and have helped to make this data narrative possible.