(Data Narrative-Assignment 3

Probability, Statistics, Data Visualisation)

- Kaushik
- IIT Gandhinagar
- Chemical Engineering
- Kaushik.kaushik@iitgn.ac.in

Abstract—This document comprises analysis on different type of tennis tournament which includes use probability to predict scores, winning matches and then random variables is used to deep down our exploration of data. Various graph helps better visualization.

OVERVIEW OF THE DATASET

In this assignment we have been eight dataset based on AusOpen-men-2013, AusOpen-women-2013, FrenchOpen-men-2013, FrenchOpen-women-2013, USOpen-men-2013, USOpen-women-2013, Wimbledon-men-2013, Wimbledon-women-2013.It contains performance of various players like who won the first set, who served first, who won the match, break points in the match, etc

QUESTION ON DATASET

Question 1) What is the probability that player 1 wins given they win the first set and first game of the second set?

Question 2) What is the probability that player 1 wins the match without losing any sets, straight sets?

Question 3) What is the probability that a player who wins the first set of a match goes on to win the match, given they had a first serve percentage(fsp) of at least 70% in the first set?

Question 4) Suppose we randomly select a match from the French Open women's singles tournament in 2013. What is the probability that the winning player had at least 10 aces, given that the losing player committed exactly 5 double faults?

Question 5) Suppose we randomly select a match from the US Open men's singles tournament in 2013. X is aces and Y be number of unforced errors committed by losing player in match. What is the probability that winning player served at least 10 aces and the losing player committed fewer than 25 unforced errors?

Question 6) Based on US Open women's singles tournament in 2013. What is the probability of at least 3 aces and no more than 2 double faults?

Question 7) What is the probability that N. Djokovic wins a match in straight sets(3-0)? Use the Markov inequality and find upper bound.

Question 8) What is the probability that a player wins less than or equal to 50% of the games? Use Chebyshev- inequality to deduce some relationship. Also plot the distribution of matches win by players.

DETAILS OF LIBRARIES USED IN PROGRAM

The most comman libraries used in our representation of data is pandas, Numpy, matplotlib there are thousands of functions that we can use and there are hundreds of libraries which make our work easy and some of them are used in this assignment. These librariers helped us to visualize data very clearly and we also discuss in details about uses of these data. Some of the most used are mentioned below:

A. Pandas (Most used library in this assignment)

- "Pandas[1] is a Python package used for working with data sets," is how you define it. It includes tools for cleaning, examining, altering, and analysing data. Why is it employed? It is used to reduce large amounts of data into a statistical form that is simple to understand. functions employed by the "Panda" library.
- 'pd.read_csv' is used to read the file from the given link and saves it in a variable which we declare and help us to print.
- "df.head()"used to define till what values we have to take the data.
- "values-count()" is used to count how many time a element is present there it returns us a value in series.
- "mean()" is used to find the mean Equations
- "id.max()" is used to find the max from the present.
- "Plt.show()" is used to plot graph

B. matplotlib.pyplot[2](Second most used library in this assignment)

C. Seaborn (third most used library in the assignment)

D. Various probability functions also used .

Functions used :

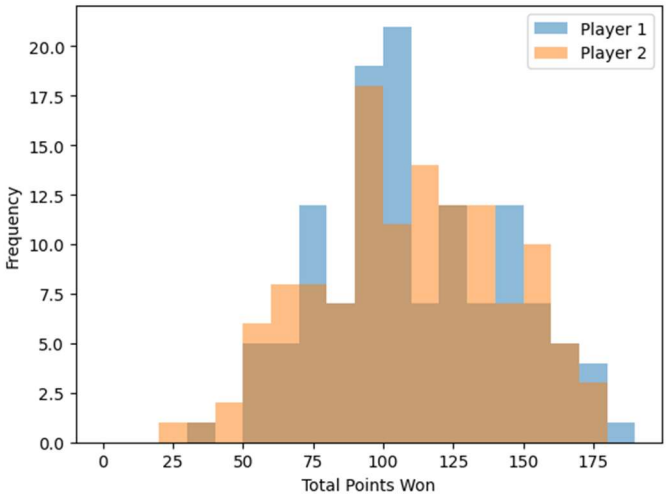
- "plot()" is used to plot the graph between x and y .

- The " show()" is used to show the plotted graph.
- "title()" is used to print the title at the bottom of the graph.
- "x-label()"is used to print the x label on the graph .
- "y-label()" is used to print the label at y axis of the graph.

Answers

1) Probability that player 1 wins given they win the first set and first game of the second set is **0.77777.**

This probability is calculated using the formula P1 * A / B, where P1 is the proportion of matches won by Player 1 where they won the first set 6-1 and the second set 6-1, A is the proportion of matches won by Player 1, and B is the proportion of matches where the first set was won 6-1 and the second set was won 6-1.



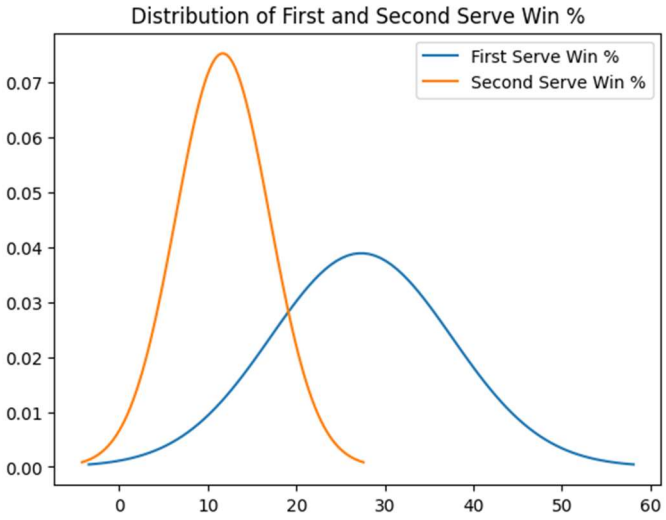
The graph is plotted using [TPW.1] and [TPW.2] is total points won by. It does not say that player 1 is better for player 2. At some points player 1 ahead of player 2 , but also at some points player 2 ahead of player 1. It just distribution of points and frequency of won.

The code for the following:

```
df = pd.read_csv("/content/AusOpen-men-2013.csv")
P1 = ((df['FNL1'] == 3) & (df['ST1.1'] == 6) & (df['ST1.2'] == 1)).sum() / (df['FNL1'] == 3).sum()
A = (df['FNL1'] == 3).sum() / df.shape[0]
B = ((df['ST1.1'] == 6) & (df['ST1.2'] == 1)).sum() / df.shape[0]
P2= P1 * A / B
print("Probability that Player 1 wins given they win the first set and first game of the second set
import matplotlib.pyplot as plt
plt.hist(df['TPW.1'], bins=range(0, 200, 10), alpha=0.5, label='Player 1')
plt.hist(df['TPW.2'], bins=range(0, 200, 10), alpha=0.5, label='Player 2')
plt.legend(loc='upper right')
plt.xlabel('Total Points Won')
plt.ylabel('Frequency')
plt.show()
```

2) The probability that player 1 wins the match without losing any sets, straight sets is 0.253968.

Looking at the graph. First serve wins is not as spread out graph as second serve win graph. First serve wins has more wins of probability in certain frequency but second serve is fairly spread out. I think it more depend's more on the game.

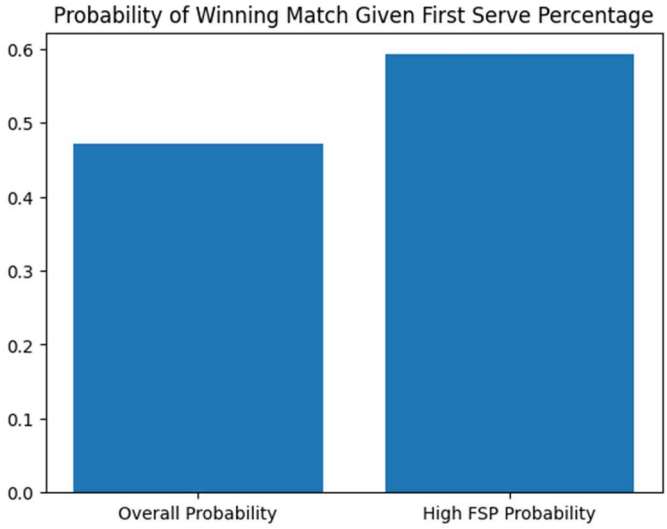


The code for the following graph:

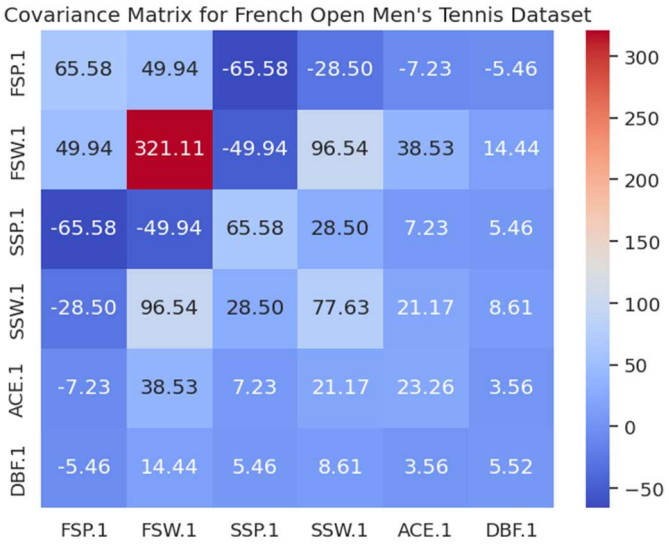
2.What is the probability that Player 1 wins the match in straight sets, i.e., winning all three sets of the match without losing any sets?

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv("/content/AusOpen-women-2013.csv")
df = df[['FSP.1', 'FSW.1', 'SSP.1', 'SSW.1']]
mean1 = df['FSW.1'].mean()
fsw= df['FSW.1'].std()
mean2 = df['SSW.1'].mean()
ssw_std = df['SSW.1'].std()
def normal_pdf(x, mean, std):
    return np.exp(-((x - mean) ** 2) / (2 * std ** 2)) / (std * np.sqrt(2 * np.pi))
fsw_values = np.linspace(mean1 - 3*fsw_std, mean1 + 3*fsw_std, 100)
fsw_probabilities = normal_pdf(fsw_values, mean1, fsw_std)
ssw_values = np.linspace(mean2 - 3*ssw_std, mean2 + 3*ssw_std, 100)
ssw_probabilities = normal_pdf(ssw_values, mean2, ssw_std)
plt.plot(fsw_values, fsw_probabilities, label='First Serve Win %')
plt.plot(ssw_values, ssw_probabilities, label='Second Serve Win %')
plt.title('Distribution of First and Second Serve Win %')
plt.legend()
plt.show()
```

3) Based on French Open-Men 2013, the probability that a player who wins the first set of a match goes on to win the match, given they had a first serve percentage (fsp) of at least 70% in the first set is **0.592592.**



The graph shows that first serve probability of winning a match is a little higher than overall probability which shows player serving first has more chances of winning match. So in case you got the chance of serving first, chances are already on your side.



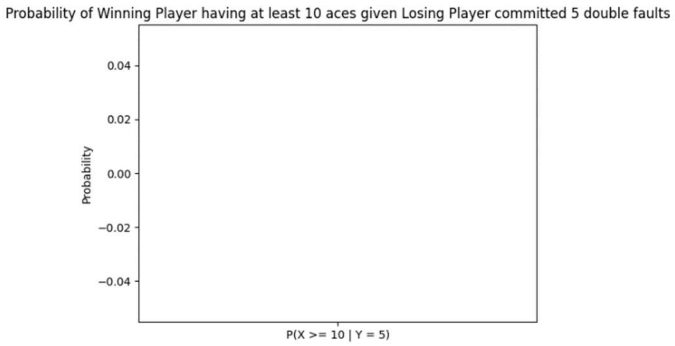
The covariance matrix plot also positive relation between first serve percentage and first serve won, which also prove my point.

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
df = pd.read_csv("/FrenchOpen-men-2013.csv")
subset = ['FSP.1', 'FSW.1', 'SSP.1', 'SSW.1', 'ACE.1', 'DBF.1']
newset = df[subset]

newset = newset.dropna()

plt.figure(figsize=(8, 6))
sns.set(font_scale=1.2)
sns.heatmap(newset.cov(), annot=True, fmt=".2f", cmap="coolwarm")
plt.title("Covariance Matrix for French Open Men's Tennis Dataset")
plt.show()
```

4) Based on French Open women's singles tournament in 2013. The probability that the winning player had at least 10 aces, given that the losing player committed exactly 5 double faults is **0**.



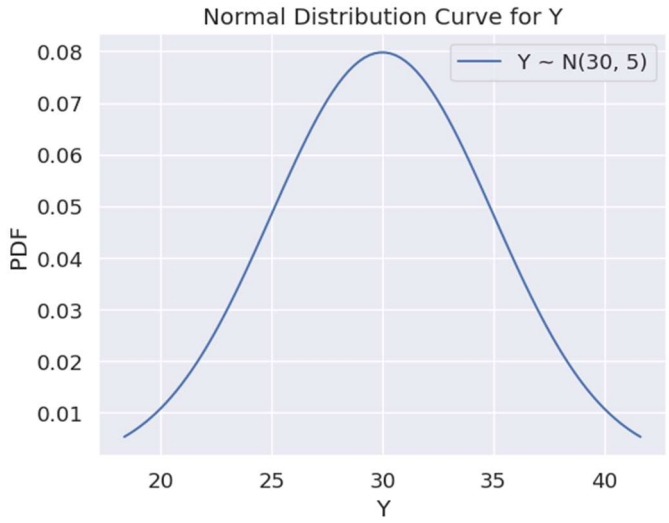
"Aces" is when a serve is in but not hit at all by the receiving player. Given we have chosen that winning player have 10 aces given losing player commits 5 double faults is approximately nil.

The graph for solution:

```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv("/content/FrenchOpen-women-2013.csv")
newset = df[df["DBF.2"] == 5]
prob = len(newset[(newset["ACE.1"] >= 10)])
cond_prob = prob / len(newset)
print("Probability:", prob)

plt.bar(["P(X >= 10 | Y = 5)"], [cond_prob])
plt.ylabel("Probability")
plt.title("Probability of Winning Player having at least 10 aces
plt.show()
```

5) Probability that winning player served at least 10 aces and the losing player committed fewer than 25 unforced errors is **0.326078795.**



Based on US men tournament 2013, probability of serving atleast 10 aces given losing player committing less than 25 is very low but still not zero like as on French open women tournament .

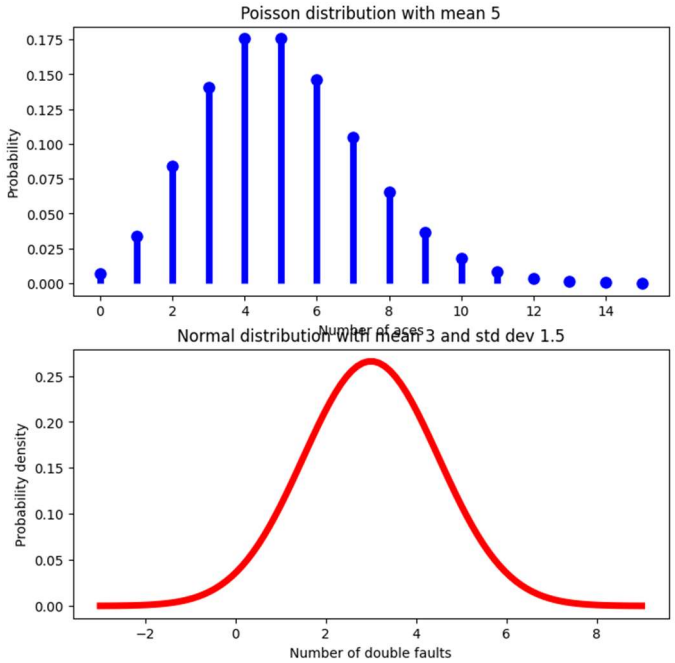
The code for the above graph below:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm, poissont
df = pd.read_csv("/content/USOpen-men-2013.csv")

X = poisson(mu=8)
Y = norm(loc=30, scale=5)
p_X = X.sf(9)
p_Y = Y.cdf(24)
prob = p_X * p_Y

print("The probability that the winning player served at least 10 aces and
x = np.linspace(Y.ppf(0.01), Y.ppf(0.99), 100)
plt.plot(x, Y.pdf(x), label="Y ~ N(30, 5)")
plt.legend(loc="best")
plt.title("Normal Distribution Curve for Y")
plt.xlabel("Y")
plt.ylabel("PDF")
plt.show()
```

6) The probability of at least 3 aces and no more than 2 double faults is 0.2210188328372913.

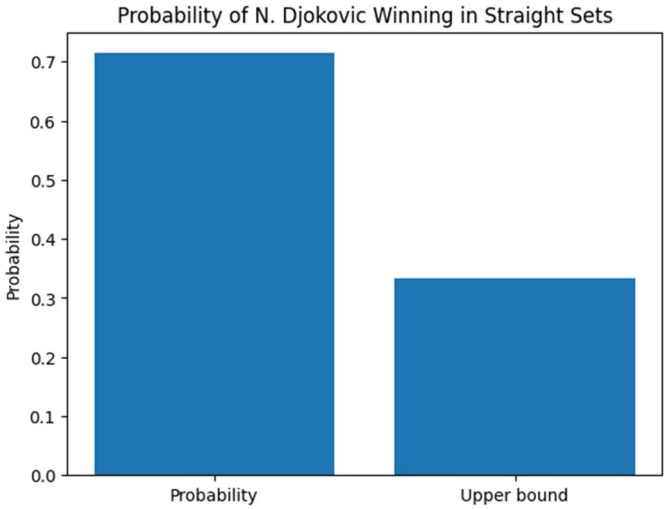


The code for the above graph:

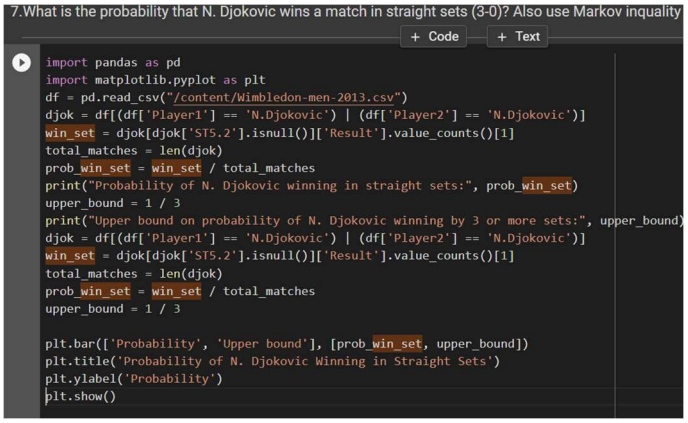
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import poisson, norm

df = pd.read_csv("/content/USOpen-women-2013.csv")
X_mean = 5
Y_mean = 3
Y_std = 1.5
X = poisson(X_mean)
Y = norm(Y_mean, Y_std)
prob1= X.sf(2) * Y.cdf(2)
print("Probability of at least 3 aces and no more than 2 double faults:", prob1)
n = 5
x = np.arange(0, 16)
fig, ax = plt.subplots(2, 1, figsize=(8, 8))
ax[0].plot(x, X.pmf(x), 'bo', ms=8)
ax[0].vlines(x, 0, X.pmf(x), colors='b', lw=5)
ax[0].set_title('Poisson distribution with mean 5')
ax[0].set_xlabel('Number of aces')
ax[0].set_ylabel('Probability')
y = np.linspace(Y_mean - 4*Y_std, Y_mean + 4*Y_std, 100)
ax[1].plot(y, Y.pdf(y), 'r-', lw=5)
ax[1].set_title('Normal distribution with mean 3 and std dev 1.5')
ax[1].set_xlabel('Number of double faults')
ax[1].set_ylabel('Probability density')
plt.show()
```

7) The probability that N. Djokovic wins a match in straight sets (3-0) is **0.7142857142857147**.

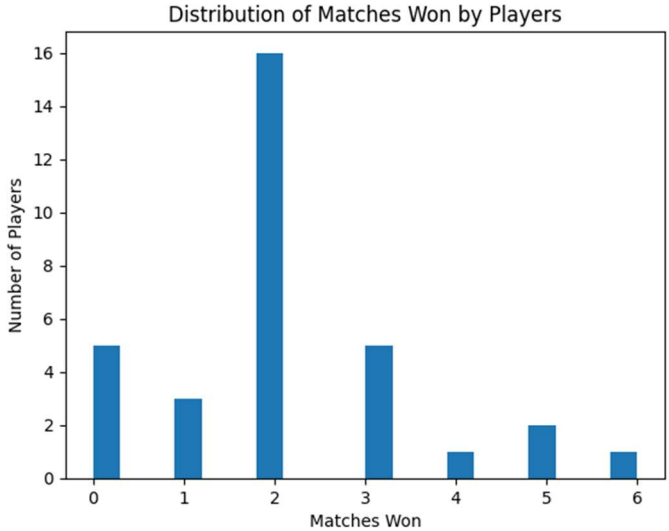


The code for below graph:



8) The probability that a player wins less than or equal to 50% of the games is **0.75**:

Looking at this graph maximum number of players has won has at least 2 matches.



```python
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv("/content/Wimbledon-women-2013.csv")
player1won= df['FNL.1'].sum()
player2won = df['FNL.2'].sum()
total_games_played = df['FNL.1'].sum() + df['FNL.2'].sum()
mean = (player1won+ player2won) / 2
variance = ((player1won- mean)**2 + (player2won - mean)**2) / 2
std_dev = variance**0.5
#Chebyshev's inequality
k = 2
probability = 1 - 1/(k**2)
lower_bound = mean - k*std_dev
upper_bound = mean + k*std_dev
print("Probability that a player wins less than or equal to 50% of the games: ", probability)
print("Lower bound: ", lower_bound)
print("Upper bound: ", upper_bound)
matches_won = df.groupby('Player1')['Result'].sum() + df.groupby('Player2')['Result'].sum()

plt.hist(matches_won, bins=20)
plt.xlabel('Matches Won')
plt.ylabel('Number of Players')
plt.title('Distribution of Matches Won by Players')
plt.show()
```


Summary

The dataset is about the various tennis tournaments such as AusOpen-men-2013, AusOpen-women-2013, FrenchOpen-men-2013, FrenchOpen-women-2013, USOpen-men-2013, USOpen-women-2013, Wimbledon-men-2013,Wimbledon-women-2013. This documents talks on various of data, judging data on many level. Many probability distribution has been used like poison distribution, normal distribution. A covariance matrix is also used determine how two quantities are related.

Acknowledgement

I would like to thank to Professor Shanmuga R sir as due to him we learnt so much about python and different libraries like pandas, numpy, scipy, and matlpotlib I am very grateful to you to assign us such a good and useful assignment that has real world application.

Refrences

[1] "Pandas Tutorial." n.d. Www.w3schools.com. https://www.w3schools.com/python/pandas/default.asp.

[2] "Getting Started Tutorials — Pandas 1.5.3 Documentation." n.d. Pandas.pydata.org. Accessed April 20, 2023. https://pandas.pydata.org/docs/getting_started/intro_tutorials/index.html.

[3] "Pandas Tutorial." 2020. GeeksforGeeks. april 20, 2020. https://www.geeksforgeeks.org/pandas-tutorial/.