Data Narrative III

Name: Saloni Sunil Shinde

Roll No: 22110242

Discipline: CSE

I. OVERVIEW OF THE DATASET

The dataset "Tennis Major Tournament Match Statistics" is a compilation of match statistics from various tennis tournaments, such as the Australian Open, French Open, Wimbledon, and the US Open in 2013. The dataset encompasses data for men's and women's singles matches and men's and women's doubles match.

The data covers crucial information such as the sets each player won, and the number of aces, double faults, and unforced errors committed by each player. It also comprises statistics concerning match length, games played, and the ultimate match result.

The dataset is available in CSV format and holds over 5000 rows. Researchers can use the dataset for various analyses and machine learning tasks, such as forecasting match outcomes, detecting player strengths and weaknesses, and examining patterns and trends in tennis match statistics over time.

II. SCIENTIFIC QUESTIONS / HYPOTHESES

1. AusOpen-men-2013: Is there a significant correlation between a player's first serve percentage and their second serve percentage in tennis matches?
2. AusOpen-women-2013: What is the average number of break points created by players in a tennis match?
3. FrenchOpen-men-2013: Does a higher number of aces won by a player in a tennis match correspond to a higher probability of winning the match?
4. FrenchOpen-women-2013: What is the impact of winning or losing set-1 on the outcome of a tennis match for a player?
5. USOpen-men-2013: How does the number of double faults committed by the loser of a tennis match compare to the average number of double faults in a match?
6. USOpen-women-2013: Is there a correlation between the number of net points attempted and the number of net points won by a player in a tennis match?
7. WimbledonOpen-men-2013: Does the round of a tennis tournament have an impact on the number of winners earned by players in a match?
8. WimbledonOpen-women-2013: Does the outcome of Set 1 have any influence on the outcome of Set 3 in a tennis match?

III. DETAILS OF LIBRARIES AND FUNCTIONS

- pandas [Library]: Pandas is a Python library used for working with data sets. It has functions for analysing, cleaning, exploring, and manipulating data[1].
- read_csv: Read a comma-separated values (csv) file into DataFrame[1].
- head: Return the first *n* rows[1].
- dropna: Remove missing values[1].
- apply: Apply a function along an axis of the DataFrame[1].
- groupby: Group DataFrame using a mapper or by a Series of columns[1].
- astype: Cast a pandas object to a specified dtype dtype[1].
- seaborn[Library]: a Python data visualization library based on matplotlib[4].
- regplot: Plot data and a linear regression model fit[4].
- scatterplot: Draw a scatter plot with possibility of several semantic groupings[4].
- boxplot: Draw a box plot to show distributions with respect to categories[4].
- matplotlib [Library]: library for creating static, animated, and interactive visualizations in Python[3].
- pyplot: mainly intended for interactive plots and simple cases of programmatic plot generation.[3]
- plot: Plot y versus x as lines and/or markers.[3]
- hist: Compute and plot a histogram.[3]
- bar: Make a bar plot.[3]
- xlabel: Labels the x-axis.
- ylabel: Labels the y-axis.
- show: Display all open figures.[3]
- axvline: Add a vertical line across the Axes[3].
- legend: Place a legend on the Axes[3].
- scatter: A scatter plot of *y* vs. *x* with varying marker size and/or color.[3]
- figure: A scatter plot of *y* vs. *x* with varying marker size and/or color[3].
- numpy [Library]: NumPy is a Python library used for working with arrays.[2]
- corrcoef: Return Pearson product-moment correlation coefficients.[2]
- mean: Compute the arithmetic mean along the specified axis.[2]

IV. ANSWERS TO THE QUESTIONS

1. To answer this question, let's visualize the relationship between a player's first serve percentage and their second serve percentage.

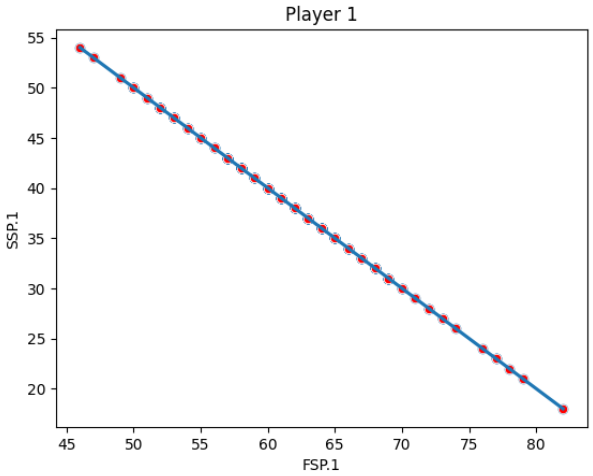
   Here is the python code for it:

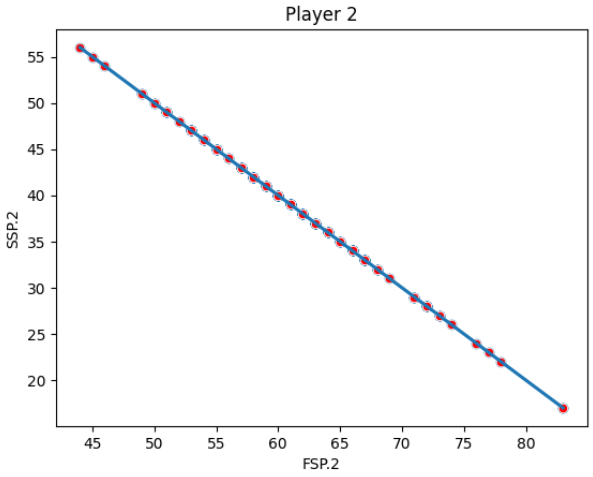
```
sns.regplot(x="FSP.1", y="SSP.1", data=df)
sns.scatterplot(x="FSP.1", y="SSP.1", data=df, color="red")
plt.title("Player 1")
plt.show()

sns.regplot(x="FSP.2", y="SSP.2", data=df)
sns.scatterplot(x="FSP.2", y="SSP.2", data=df, color="red")
plt.title("Player 2")
plt.show()
```

This code uses the Seaborn library to create a scatter plot of the first serve percentage (x-axis) vs. the second serve percentage (y-axis). It also adds a regression line to the plot to show the relationship between the two variables.

This is the graph:





By graph we can say that, there is a correlation between a player's first serve percentage and their second serve percentage as the regression line is steep and the data points are clustered around the line. As the slope of the line is negative, there is negative correlation. The first serve percentage increases, the second serve percentage decreases, and vice versa.

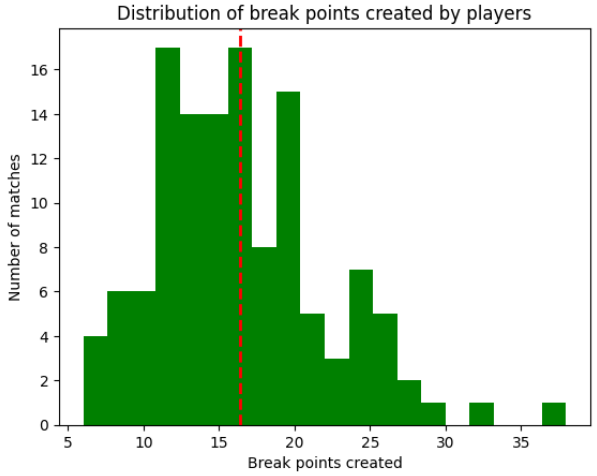
2. To answer this question, we need two columns consisting information about break points created by player 1 and player 2. By that, we can calculate the average number of break points created by all players in the dataset. We can also visualize the distribution of break points created by players using a histogram. This can give us a better understanding of the typical range of break points created by players in a tennis match.

Here is python code for it:

```
total_bpc = data["BPC.1"] + data["BPC.2"]
avg_bpc = total_bpc.mean()
print("Average number of break points created per match:", avg_bpc)

plt.hist(total_bpc, bins=20, color='green')
plt.axvline(avg_bpc, color='red', linestyle='dashed', linewidth=2)
```

This is the graph:



From the histogram, we can see that the majority of matches have a relatively low number of break points created, with the peaks at around 12-20 break points per match. The dashed red line represents the average number of break points created per match, which is around 16 break points.

3. To answer the question, we would need to compare the distribution of aces won by the winner to the distribution of aces won by the loser and analyze the relationship between aces won and match outcome using statistical methods.

Here is the python code for it:

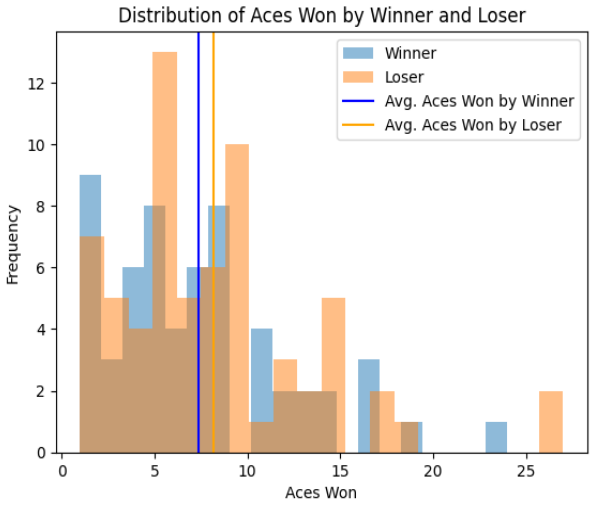
```
winner_aces_df = data.loc[data['Result'] == 1, ['ACE.1']]
loser_aces_df = data.loc[data['Result'] == 0, ['ACE.2']]

avg_aces_winner = winner_aces_df.mean()[0]
avg_aces_loser = loser_aces_df.mean()[0]

plt.hist(winner_aces_df['ACE.1'], bins=20, alpha=0.5, label='Winner')
plt.hist(loser_aces_df['ACE.2'], bins=20, alpha=0.5, label='Loser')
plt.axvline(x=avg_aces_winner, color='blue', label='Avg. Aces Won by Winner')
plt.axvline(x=avg_aces_loser, color='orange', label='Avg. Aces Won by Loser')
```

This code first gets the data for aces won by the winner and loser separately, and then calculates the average number of aces won by each. It then plots two histograms of the distribution of aces won by the winner and loser, and adds vertical lines to show the average number of aces won by each.

This is the resultant histogram:



By looking at the histograms and the vertical lines representing the averages, we can see that the winners tend to win slightly less aces on average than the losers. This suggests that there is no such correlation between the number of aces won and the probability of winning the match.

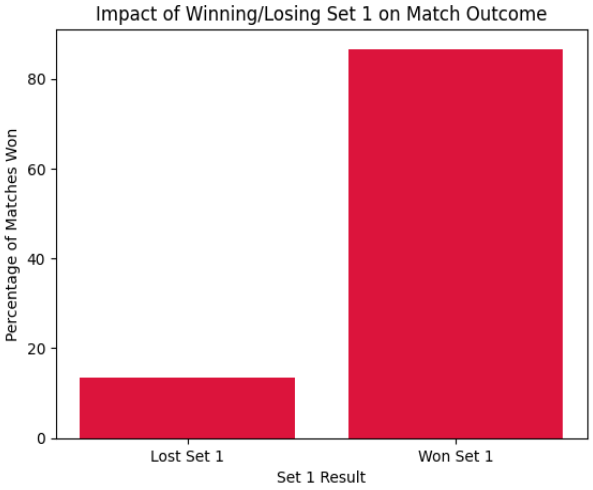
4. To answer this question, we can analyze the match data and determine the percentage of matches won by players who won set-1 compared to those who lost set-1.

Here is the python code for it:

```
set1_df = data[['ST1.1', 'ST1.2', 'Result']].copy()
set1_df.dropna(inplace=True)
set1_df['Set1_winner'] = set1_df.apply(lambda x: x['ST1.1'] if x['Result'] == 1
                                       else x['ST1.2'], axis=1)
set1_df['Set1_loser'] = set1_df.apply(lambda x: x['ST1.2'] if x['Result'] == 1
                                      else x['ST1.1'], axis=1)
set1_df['Set1_winner_win'] = set1_df.apply(lambda x: 1
                                           if x['Set1_winner'] > x['Set1_loser']
                                           else 0, axis=1)
set1_win_percentage = set1_df.groupby('Set1_winner_win').size().div(len(set1_df)).mul(100).round(2)
labels = ['Lost Set 1', 'Won Set 1']
values = set1_win_percentage.values

plt.bar(labels, values, color="#DC143C")
```

This is the bar graph:



The resulting plot is showing us the percentage of matches won by players who won Set 1 and those who lost Set 1. By the graph, we can clearly state that set1 of the match has huge impact on winning or losing the match.

5. To answer the question, we are comparing the distribution of double faults made by the loser of a match to the overall distribution of unforced errors in a match. For this we will need ‘DBF.2’, ‘DBF.1’ and ‘Result’ columns.

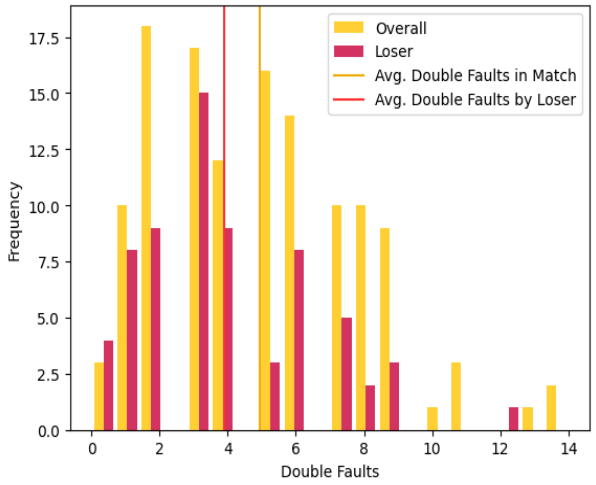
Here is the code for that:

```
loser_df = data[['DBF.2']][data['Result'] == 0]

avg_df_match = data['DBF.1'].mean()
avg_df_loser = loser_df.mean()
colors = ['#FFC300', '#C70039']

plt.hist([data['DBF.1'].dropna(), loser_df['DBF.2'].dropna()], bins=20,
         alpha=0.8, label=['Overall', 'Loser'], color=colors)
plt.axvline(x=avg_df_match, color='#EEAD0E', label='Avg. Double Faults in Match')
plt.axvline(x=avg_df_loser[0], color='#FF3030', label='Avg. Double Faults by Loser')
```

This is the graph:



Based on the visualization showing that the average number of double faults in a match is higher than the number of double faults committed by the losers, we can conclude that double faults are not the sole determining factor in the outcome of a match. It is possible for a player to commit a higher number of double faults and still win the match if they perform well in other areas.

6. To answer this question, we can use a scatter plot to visualize the relationship between the number of net points attempted and the number of net points won by a player in a tennis match.

Here's the Python code to create the scatter plot:

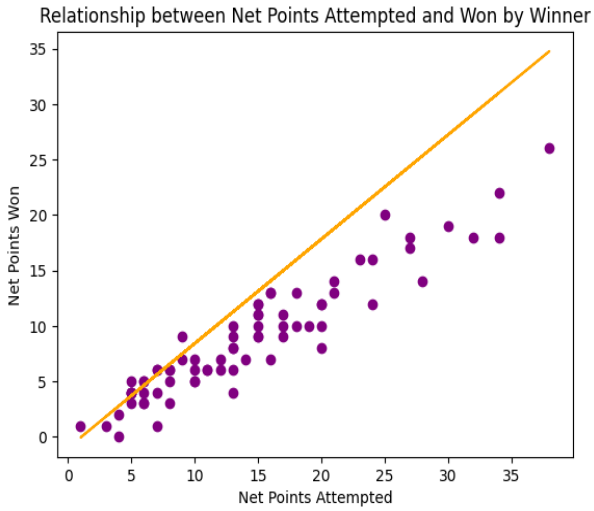
```
winner_net_points = data[['NPW.1', 'NPA.1']]

corr = winner_net_points.corr()

plt.scatter(winner_net_points['NPA.1'], winner_net_points['NPW.1'], color='purple')
plt.plot(winner_net_points['NPA.1'],
         winner_net_points['NPA.1']*corr.iloc[0,1] - corr.iloc[0,0], color='orange')
```

This code will create a scatter plot with the number of net points attempted on the x-axis and the number of net points won on the y-axis. The plot will also include a linear regression line to show the direction and strength of the correlation between the two variables.

This is the scatter plot:



The plot shows a clear upward trend, this indicates that there is a positive correlation between the number of net points attempted and won, meaning that the more net points attempted, the more net points won.

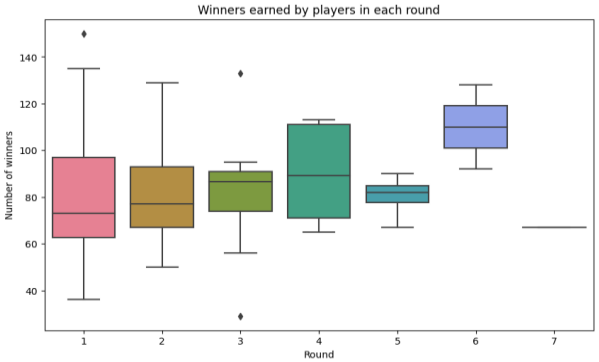
7. To answer this question, we can analyse the data set and create a visual representation of the number of winners earned by players in each round of a tournament. We can use a box plot to compare the distribution of winners across different rounds of the tournament.

Here's the Python code to create the box plot:

```
plt.figure(figsize=(10, 6))
ax = sns.boxplot(x='Round',
                 y=data['WNR.1']+data['WNR.2'], data=data, palette='husl')
plt.title('Winners earned by players in each round')
```

In this code, we are using the Seaborn library to create the box plot. We are selecting the 'Round' and 'WNR.1' & 'WNR.2' columns from the data set and passing them to the sns.boxplot() function. We are also specifying a color palette using the palette parameter.

This is the box plot:



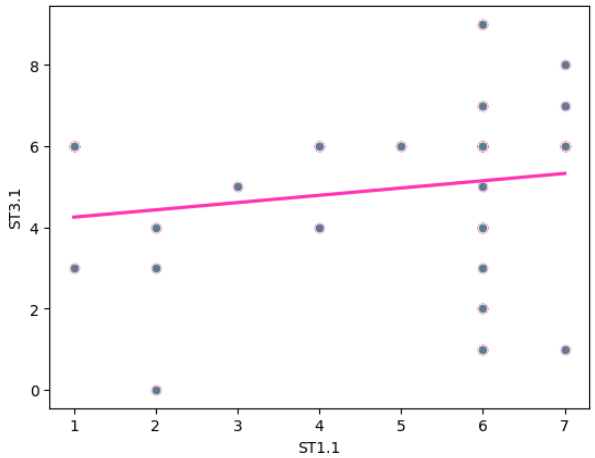
The resulting plot shows the distribution of winners earned by players in each round of the tournament. We can see that the median number of winners is slightly higher in the later rounds of the tournament, indicating that the round of the tournament does have an impact on the number of winners earned by players.

8. To investigate whether the outcome of set-1 has any influence on the outcome of set-3 in a tennis match, we can use a scatter plot to visualize the relationship between the scores of set-1 and set-3. We will also add a regression line to the plot to see if there is any linear correlation between the scores.

Here is the Python code to create the scatter plot with Seaborn library:

```
sns.regplot(x='ST1.1', y='ST3.1', data=tennis_data, ci=None, color="#FF34B3")
sns.scatterplot(x='ST1.1', y='ST3.1', data=tennis_data, color="#607B8B")
plt.show()
```

This is the graph:



A horizontal line indicates that for any given Set 1 set-1 score, the set-3 score can vary widely. The outcome of set-1 does not seem to have a significant influence on the outcome of Set 3. This means that the set-1 score does not provide any information about what the Set 3 score.

V. Summary of the Observations

In AusOpen-men-2013, as the first serve percentage increases for a player, his second serve percentage decreases. In AusOpen-women-2013, the average number of break points created per match is around 16 points. In FrenchOpen-men-2013, no such relation exists between the number of aces won and the probability of winning the match. In FrenchOpen-women-2013, Set 1 of the match has an huge impact on the final result of the match. In USOpen-men-2013, even if a player commits higher number of double faults, still he can win the match. In USOpen-women-2013, there are more chances to win net points if the player attempts more. In WimbledonOpen-men-2013,

the median number of winners is slightly higher in the later rounds of the tournament. In WimbledonOpen-women-2013, Set 1 score doesn't have any impact on the result of Set 3.

VI. References

[1] “Pandas Documentation — Pandas 1.0.1 Documentation.” n.d. Pandas.pydata.org. https://pandas.pydata.org/docs/

[2] “NumPy Documentation.” n.d. Numpy.org. https://numpy.org/doc/.

[3] “Matplotlib: Python Plotting — Matplotlib 3.3.4 Documentation.” n.d. Matplotlib.org. https://matplotlib.org/stable/index.html.

[4] “Seaborn: Statistical Data Visualization — Seaborn 0.9.0 Documentation.” Pydata.org. 2012. https://seaborn.pydata.org/.

VII. Acknowledgements

This analysis of the "Tennis Major Tournament Match Statistics" dataset was only possible due to the hard work of its creators who collected and organized the data. I would like to thank them and many researchers who have contributed to this field, making it possible to get the information from large datasets like this one. Also, I would like to thank Prof. Shanmuga to provide us this dataset.