FEU INSTITUTE OF TECHNOLOGY

Case Study 2

Name: Joaquin Alfonso Louis S. Cruz	Date: March 5, 2025
Activity: Case Study 2	Instructor: Ms. Geliza Marie Alcobar

General Instructions:

Put your data skills to the test! This case study presents a real-world scenario where you'll navigate the complexities of data collection, wrangle diverse datasets, create impactful visualizations, and ultimately, extract meaningful interpretations to drive informed decisions. Do this task individually.

First Step:

Select what type of analytics to focus on:

Customer Analytics	User Behavior Analytics
Customer Intelligence Analysis	Spatial Analytics
Temporal Analysis	Profiling

Your analytics of choice is: Profiling

Title: Rivalry: Lewis VS Max Last 50 Race Stat Comparison

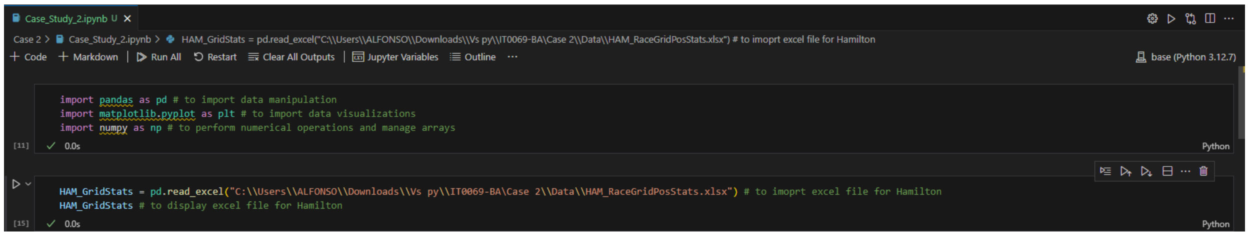
Reference: https://www.motorsportstats.com/

Second Step:

Find an actual dataset **with at least 50 rows** about the analytics of your choice.

Perform data preparation using Python. Please include #comment for each line of code explaining the syntax and results.

Data collection codes and results here.

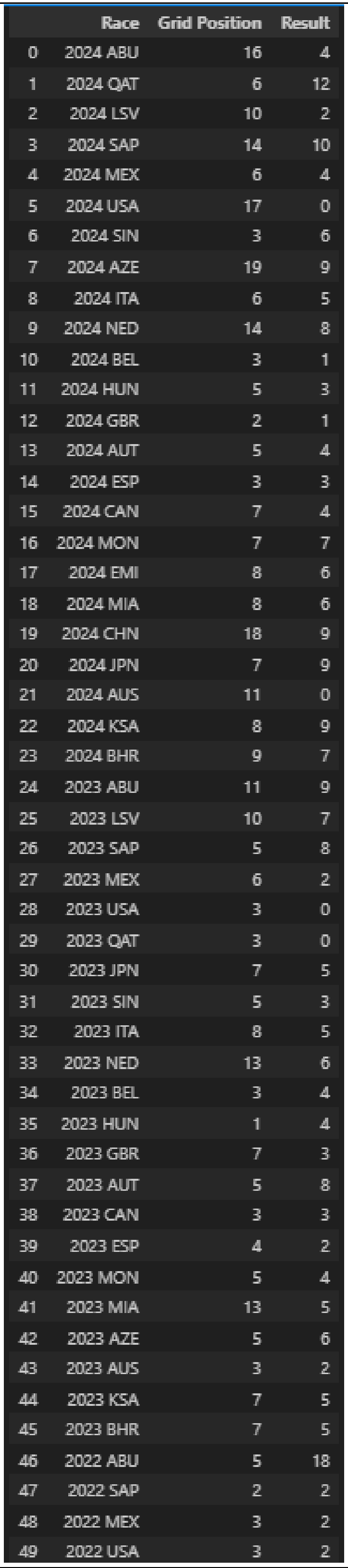

Case_Study_2.ipynb U
Case 2 > Case_Study_2.ipynb > HAM_GridStats = pd.read_excel("C:\\Users\\ALFONSO\\Downloads\\Vs py\\IT0069-BA\Case 2\\Data\\HAM_RaceGridPosStats.xlsx") # to imoprt excel file for Hamilton
+ Code
+ Markdown
Run All
Restart
Clear All Outputs
Jupyter Variables
Outline
base (Python 3.12.7)

```
import pandas as pd # to import data manipulation
import matplotlib.pyplot as plt # to import data visualizations
import numpy as np # to perform numerical operations and manage arrays
```

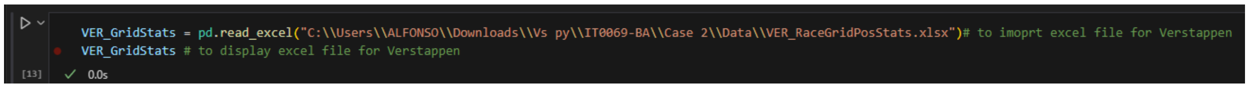
[11] ✓ 0.0s Python

```
HAM_GridStats = pd.read_excel("C:\\Users\\ALFONSO\\Downloads\\Vs py\\IT0069-BA\Case 2\\Data\\HAM_RaceGridPosStats.xlsx") # to imoprt excel file for Hamilton
HAM_GridStats # to display excel file for Hamilton
```

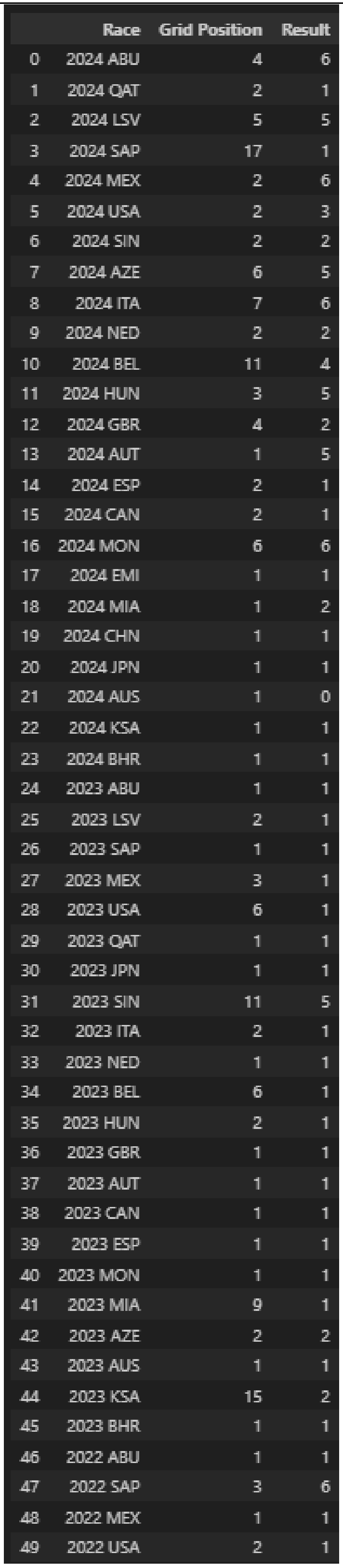
[15] ✓ 0.0s Python



	Race	Grid Position	Result
0	2024 ABU	16	4
1	2024 QAT	6	12
2	2024 LSV	10	2
3	2024 SAP	14	10
4	2024 MEX	6	4
5	2024 USA	17	0
6	2024 SIN	3	6
7	2024 AZE	19	9
8	2024 ITA	6	5
9	2024 NED	14	8
10	2024 BEL	3	1
11	2024 HUN	5	3
12	2024 GBR	2	1
13	2024 AUT	5	4
14	2024 ESP	3	3
15	2024 CAN	7	4
16	2024 MON	7	7
17	2024 EMI	8	6
18	2024 MIA	8	6
19	2024 CHN	18	9
20	2024 JPN	7	9
21	2024 AUS	11	0
22	2024 KSA	8	9
23	2024 BHR	9	7
24	2023 ABU	11	9
25	2023 LSV	10	7
26	2023 SAP	5	8
27	2023 MEX	6	2
28	2023 USA	3	0
29	2023 QAT	3	0
30	2023 JPN	7	5
31	2023 SIN	5	3
32	2023 ITA	8	5
33	2023 NED	13	6
34	2023 BEL	3	4
35	2023 HUN	1	4
36	2023 GBR	7	3
37	2023 AUT	5	8
38	2023 CAN	3	3
39	2023 ESP	4	2
40	2023 MON	5	4
41	2023 MIA	13	5
42	2023 AZE	5	6
43	2023 AUS	3	2
44	2023 KSA	7	5
45	2023 BHR	7	5
46	2022 ABU	5	18
47	2022 SAP	2	2
48	2022 MEX	3	2
49	2022 USA	3	2



```
VER_GridStats = pd.read_excel("C:\\Users\\ALFONSO\\Downloads\\Vs py\\IT0069-BA\\Case 2\\Data\\VER_RaceGridPosStats.xlsx")# to imoprt excel file for Verstappen
VER_GridStats # to display excel file for Verstappen
```

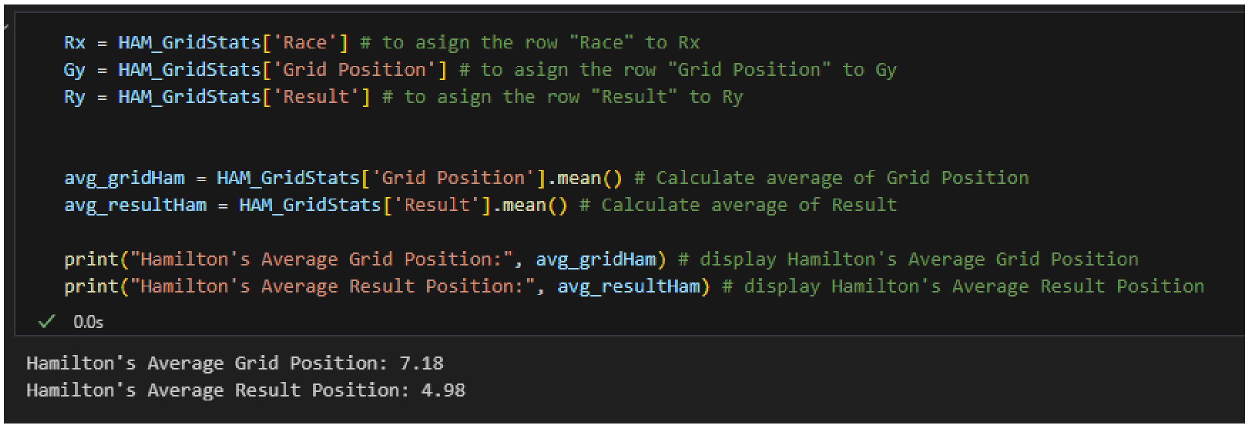


	Race	Grid Position	Result
0	2024 ABU	4	6
1	2024 QAT	2	1
2	2024 LSV	5	5
3	2024 SAP	17	1
4	2024 MEX	2	6
5	2024 USA	2	3
6	2024 SIN	2	2
7	2024 AZE	6	5
8	2024 ITA	7	6
9	2024 NED	2	2
10	2024 BEL	11	4
11	2024 HUN	3	5
12	2024 GBR	4	2
13	2024 AUT	1	5
14	2024 ESP	2	1
15	2024 CAN	2	1
16	2024 MON	6	6
17	2024 EMI	1	1
18	2024 MIA	1	2
19	2024 CHN	1	1
20	2024 JPN	1	1
21	2024 AUS	1	0
22	2024 KSA	1	1
23	2024 BHR	1	1
24	2023 ABU	1	1
25	2023 LSV	2	1
26	2023 SAP	1	1
27	2023 MEX	3	1
28	2023 USA	6	1
29	2023 QAT	1	1
30	2023 JPN	1	1
31	2023 SIN	11	5
32	2023 ITA	2	1
33	2023 NED	1	1
34	2023 BEL	6	1
35	2023 HUN	2	1
36	2023 GBR	1	1
37	2023 AUT	1	1
38	2023 CAN	1	1
39	2023 ESP	1	1
40	2023 MON	1	1
41	2023 MIA	9	1
42	2023 AZE	2	2
43	2023 AUS	1	1
44	2023 KSA	15	2
45	2023 BHR	1	1
46	2022 ABU	1	1
47	2022 SAP	3	6
48	2022 MEX	1	1
49	2022 USA	2	1

FEU INSTITUTE OF TECHNOLOGY

Third Step:

Numerical reports using Python. Please include #comment for each line of code explaining the syntax and results.



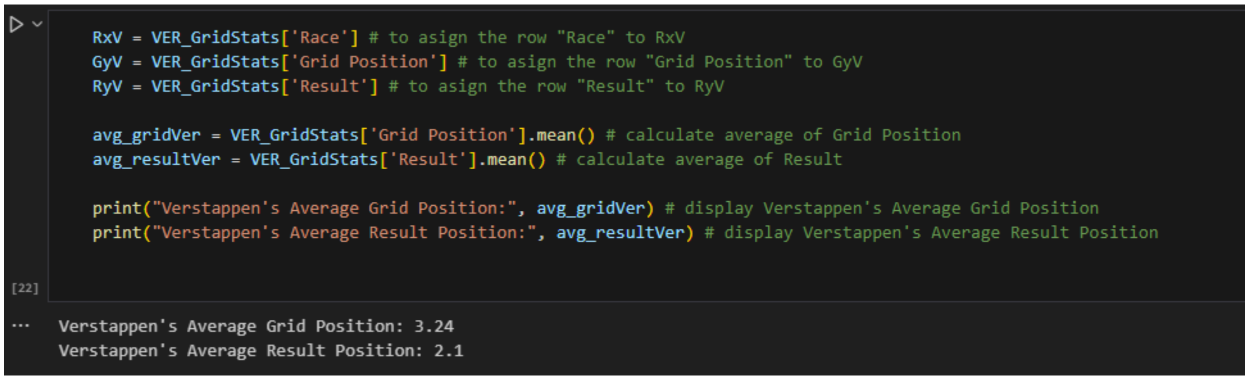
```
Rx = HAM_GridStats['Race'] # to asign the row "Race" to Rx
Gy = HAM_GridStats['Grid Position'] # to asign the row "Grid Position" to Gy
Ry = HAM_GridStats['Result'] # to asign the row "Result" to Ry


avg_gridHam = HAM_GridStats['Grid Position'].mean() # Calculate average of Grid Position
avg_resultHam = HAM_GridStats['Result'].mean() # Calculate average of Result

print("Hamilton's Average Grid Position:", avg_gridHam) # display Hamilton's Average Grid Position
print("Hamilton's Average Result Position:", avg_resultHam) # display Hamilton's Average Result Position
```

✓ 0.0s

```
Hamilton's Average Grid Position: 7.18
Hamilton's Average Result Position: 4.98
```



```
RxV = VER_GridStats['Race'] # to asign the row "Race" to RxV
GyV = VER_GridStats['Grid Position'] # to asign the row "Grid Position" to GyV
RyV = VER_GridStats['Result'] # to asign the row "Result" to RyV

avg_gridVer = VER_GridStats['Grid Position'].mean() # calculate average of Grid Position
avg_resultVer = VER_GridStats['Result'].mean() # calculate average of Result

print("Verstappen's Average Grid Position:", avg_gridVer) # display Verstappen's Average Grid Position
print("Verstappen's Average Result Position:", avg_resultVer) # display Verstappen's Average Result Position
```

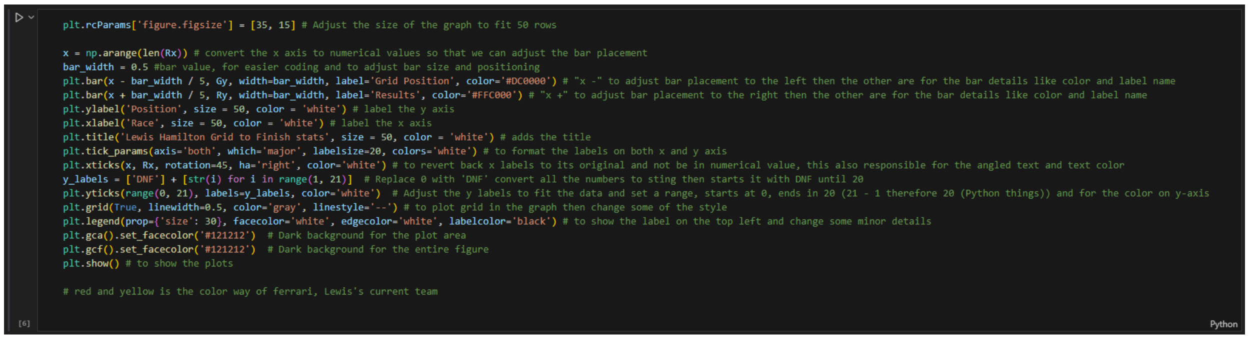
[22]

```
Verstappen's Average Grid Position: 3.24
Verstappen's Average Result Position: 2.1
```

FEU INSTITUTE OF TECHNOLOGY

Fourth Step:

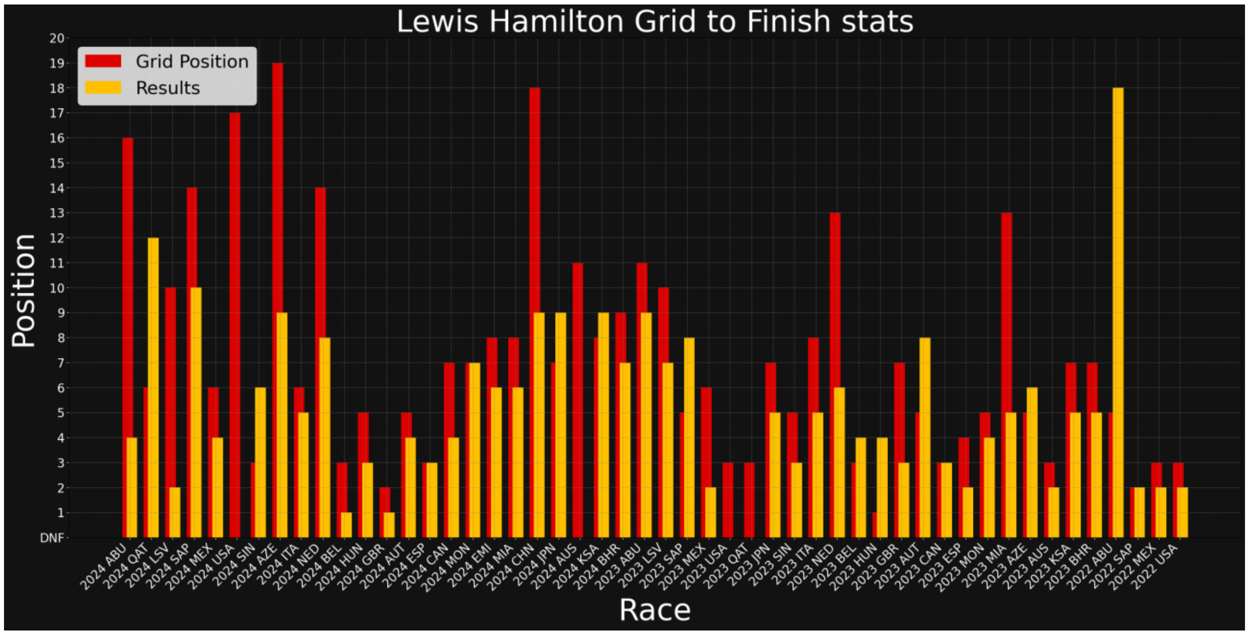
Visual reports using Python. Please include #comment for each line of code explaining the syntax and results.

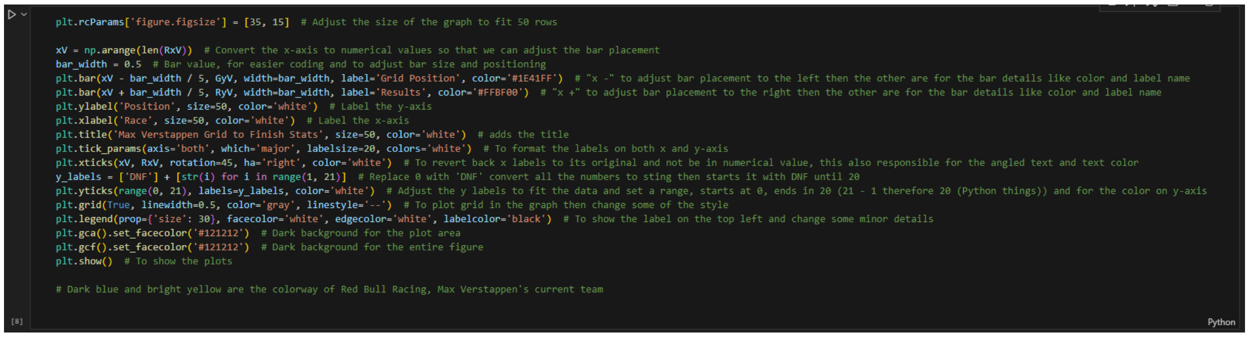


```python
plt.rcParams['figure.figsize'] = [35, 15] # Adjust the size of the graph to fit 50 rows

x = np.arange(len(Rx)) # convert the x axis to numerical values so that we can adjust the bar placement
bar_width = 0.5 #bar value, for easier coding and to adjust bar size and positioning
plt.bar(x - bar_width / 5, Gy, width=bar_width, label='Grid Position', color='#DC0000') # "x -" to adjust bar placement to the left then the other are for the bar details like color and label name
plt.bar(x + bar_width / 5, Ry, width=bar_width, label='Results', color='#FFC000') # "x +" to adjust bar placement to the right then the other are for the bar details like color and label name
plt.ylabel('Position', size = 50, color = 'white') # label the y axis
plt.xlabel('Race', size = 50, color = 'white') # label the x axis
plt.title('Lewis Hamilton Grid to Finish stats', size = 50, color = 'white') # adds the title
plt.tick_params(axis='both', which='major', labelsize=20, colors='white') # to format the labels on both x and y axis
plt.xticks(x, Rx, rotation=45, ha='right', color='white') # to revert back x labels to its original and not be in numerical value, this also responsible for the angled text and text color
y_labels = ['DNF'] + [str(i) for i in range(1, 21)]  # Replace 0 with 'DNF' convert all the numbers to sting then starts it with DNF until 20
plt.yticks(range(0, 21), labels=y_labels, color='white')  # Adjust the y labels to fit the data and set a range, starts at 0, ends in 20 (21 - 1 therefore 20 (Python things)) and for the color on y-axis
plt.grid(True, linewidth=0.5, color='gray', linestyle='--') # to plot grid in the graph then change some of the style
plt.legend(prop={'size': 30}, facecolor='white', edgecolor='white', labelcolor='black') # to show the label on the top left and change some minor details
plt.gca().set_facecolor('#121212')  # Dark background for the plot area
plt.gcf().set_facecolor('#121212')  # Dark background for the entire figure
plt.show() # to show the plots

# red and yellow is the color way of ferrari, Lewis's current team
```

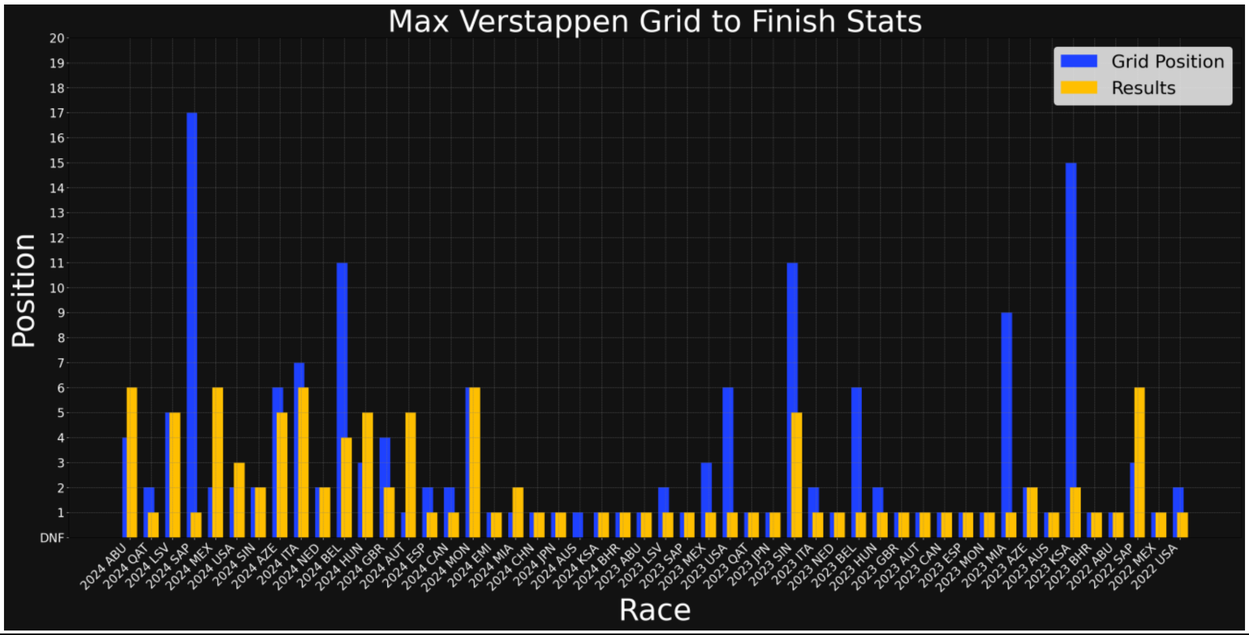


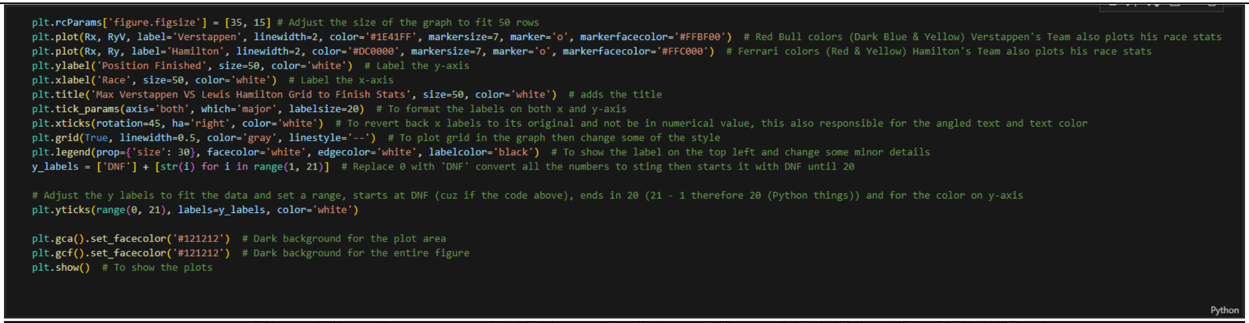


```python
plt.rcParams['figure.figsize'] = [35, 15]  # Adjust the size of the graph to fit 50 rows

xV = np.arange(len(RxV))  # Convert the x-axis to numerical values so that we can adjust the bar placement
bar_width = 0.5  # Bar value, for easier coding and to adjust bar size and positioning
plt.bar(xV - bar_width / 5, GyV, width=bar_width, label='Grid Position', color='#1E41FF')  # "x -" to adjust bar placement to the left then the other are for the bar details like color and label name
plt.bar(xV + bar_width / 5, RyV, width=bar_width, label='Results', color='#FFBF00')  # "x +" to adjust bar placement to the right then the other are for the bar details like color and label name
plt.ylabel('Position', size=50, color='white')  # Label the y-axis
plt.xlabel('Race', size=50, color='white')  # Label the x-axis
plt.title('Max Verstappen Grid to Finish Stats', size=50, color='white')  # adds the title
plt.tick_params(axis='both', which='major', labelsize=20, colors='white')  # To format the labels on both x and y-axis
plt.xticks(xV, RxV, rotation=45, ha='right', color='white')  # To revert back x labels to its original and not be in numerical value, this also responsible for the angled text and text color
y_labels = ['DNF'] + [str(i) for i in range(1, 21)]  # Replace 0 with 'DNF' convert all the numbers to sting then starts it with DNF until 20
plt.yticks(range(0, 21), labels=y_labels, color='white')  # Adjust the y labels to fit the data and set a range, starts at 0, ends in 20 (21 - 1 therefore 20 (Python things)) and for the color on y-axis
plt.grid(True, linewidth=0.5, color='gray', linestyle='--')  # To plot grid in the graph then change some of the style
plt.legend(prop={'size': 30}, facecolor='white', edgecolor='white', labelcolor='black')  # To show the label on the top left and change some minor details
plt.gca().set_facecolor('#121212')  # Dark background for the plot area
plt.gcf().set_facecolor('#121212')  # Dark background for the entire figure
plt.show()  # To show the plots

# Dark blue and bright yellow are the colorway of Red Bull Racing, Max Verstappen's current team
```



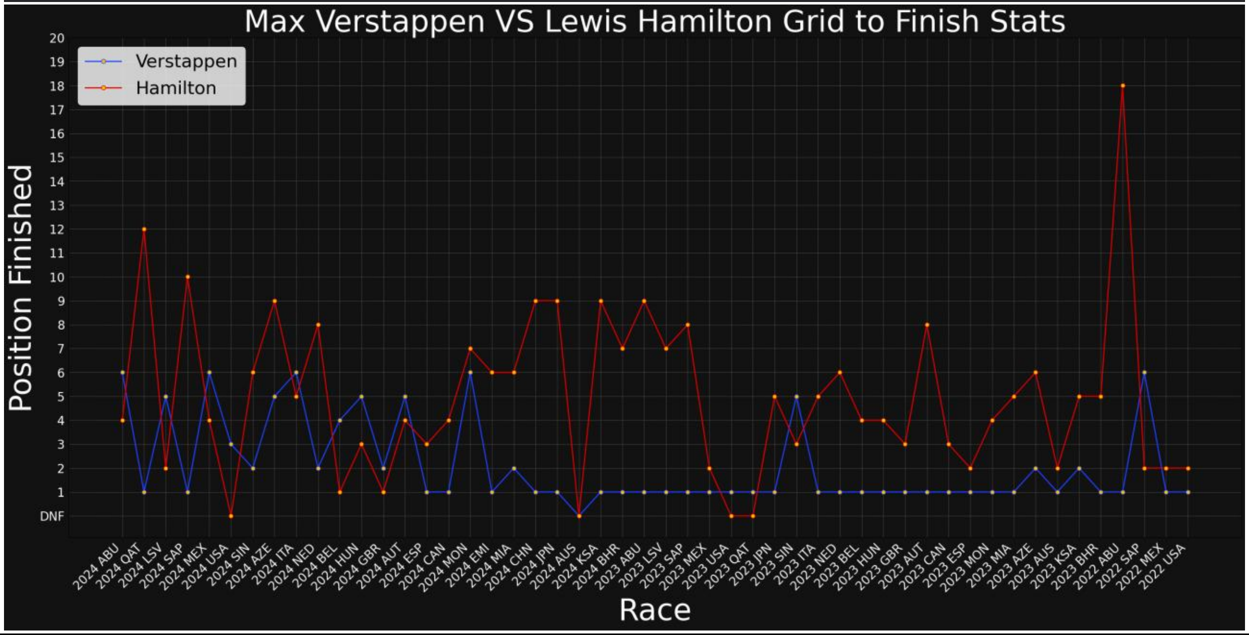


```python
plt.rcParams['figure.figsize'] = [35, 15] # Adjust the size of the graph to fit 50 rows
plt.plot(Rx, RyV, label='Verstappen', linewidth=2, color='#1E41FF', markersize=7, marker='o', markerfacecolor='#FFBF00')  # Red Bull colors (Dark Blue & Yellow) Verstappen's Team also plots his race stats
plt.plot(Rx, Ry, label='Hamilton', linewidth=2, color='#DC0000', markersize=7, marker='o', markerfacecolor='#FFC000')  # Ferrari colors (Red & Yellow) Hamilton's Team also plots his race stats
plt.ylabel('Position Finished', size=50, color='white')  # Label the y-axis
plt.xlabel('Race', size=50, color='white')  # Label the x-axis
plt.title('Max Verstappen VS Lewis Hamilton Grid to Finish Stats', size=50, color='white')  # adds the title
plt.tick_params(axis='both', which='major', labelsize=20)  # To format the labels on both x and y-axis
plt.xticks(rotation=45, ha='right', color='white')  # To revert back x labels to its original and not be in numerical value, this also responsible for the angled text and text color
plt.grid(True, linewidth=0.5, color='gray', linestyle='--')  # To plot grid in the graph then change some of the style
plt.legend(prop={'size': 30}, facecolor='white', edgecolor='white', labelcolor='black')  # To show the label on the top left and change some minor details
y_labels = ['DNF'] + [str(i) for i in range(1, 21)]  # Replace 0 with 'DNF' convert all the numbers to sting then starts it with DNF until 20

# Adjust the y labels to fit the data and set a range, starts at DNF (cuz if the code above), ends in 20 (21 - 1 therefore 20 (Python things)) and for the color on y-axis
plt.yticks(range(0, 21), labels=y_labels, color='white')

plt.gca().set_facecolor('#121212')  # Dark background for the plot area
plt.gcf().set_facecolor('#121212')  # Dark background for the entire figure
plt.show()  # To show the plots
```

Python



In Conclusion Max Verstappen was more consistent and faster than Lewis Hamilton in the last 50 races that span 3 seasons. Furthermore, Verstappen was more efficient than Hamilton, as the data charts show that Hamilton has more scattered values compared to Verstappen.

Fifth step:

Presentation.

Present this data to me personally at a given time.

FEU INSTITUTE OF TECHNOLOGY

Book a schedule for presentation here: https://calendar.app.google/K2TBgL5Ly9CSy5gG8

RUBRICS:

Criteria	**Percentage**
Data Quality (handling of data, cleaning of data, transformation)	15%
Numerical reports (relevancy and accuracy of results)	20%
Visual reports (readability, quality)	30%
Storytelling	25%
Completeness + Timeliness	10%
Total	**100%**