

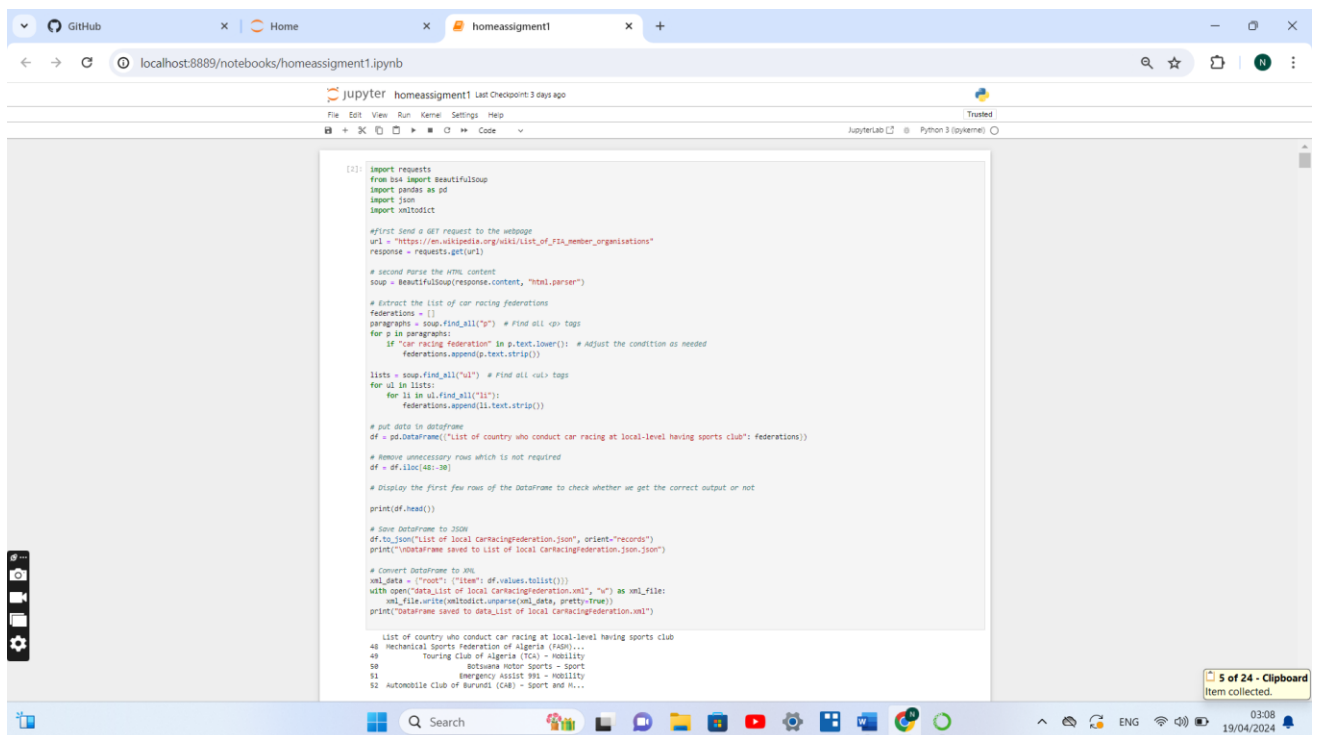
Report for HomeAssignment1

The code is written in Jupiter's notebook. Here is a brief of what work has been performed in the code

1. In the initial stage as per the requirement of the task data has been collected from the “ **list of FIA member organizations**”. “The FIA is the governing body for world motor sport and the federation of the world's leading motoring organizations” The list contains all the local federation and clubs of a different country that comes under this or are members of this all listed down so the dataset taken from the source:-

https://en.wikipedia.org/wiki/List_of_FIA_member_organisation

And results have been stored based on the requirement in List of local CarRacingFederation.json and data_List of local CarRacingFederation.xml few rows just for checking are also printed on the screen here in this both are provided.



```
[1]: import requests
from bs4 import BeautifulSoup
import pandas as pd
import json
import xmltodict

#first Send a GET request to the webpage
url = "https://en.wikipedia.org/wiki/List_of_FIA_member_organisations"
response = requests.get(url)

# second Parse the HTML content
soup = BeautifulSoup(response.content, "html.parser")

# Extract the list of car racing federations
federations = []
for p in paragraphs:
    if "car racing federation" in p.text.lower(): # Adjust the condition as needed
        federations.append(p.text.strip())

lists = soup.find_all("ul") # Find all <ul> tags
for ul in lists:
    for li in ul.find_all("li"):
        federations.append(li.text.strip())

# put data in dataframe
df = pd.DataFrame({"List of country who conduct car racing at local-level having sports club": federations})

# Remove unnecessary rows which is not required
df = df.iloc[48:98]

# Display the first few rows of the Dataframe to check whether we get the correct output or not
print(df.head())

# Save Dataframe to JSON
df.to_json("List of local CarRacingFederation.json", orient="records")
print("Dataframe saved to List of local CarRacingFederation.json")

# Convert Dataframe to XML
xml_data = ("root": [{"item": df.values.tolist()}])
with open("data_list of local CarRacingFederation.xml", "w") as xml_file:
    xml_file.write(xmltodict.unparse(xml_data, pretty=True))
print("Dataframe saved to data_List of local CarRacingFederation.xml")

List of country who conduct car racing at local-level having sports club
48 Mechanical Sports Federation of Algeria (FASB)...
49 Touring Club of Algeria (TCA) - Mobility
50 Botswana Motor Sports - Sport
51 Emergency Assist 991 - Mobility
52 automobile club of Burundi (CAB) - Sport and M...
```

Steps used in this are

#first Send a GET request to the webpage

second Parse the HTML content

Extract the list of car racing federations

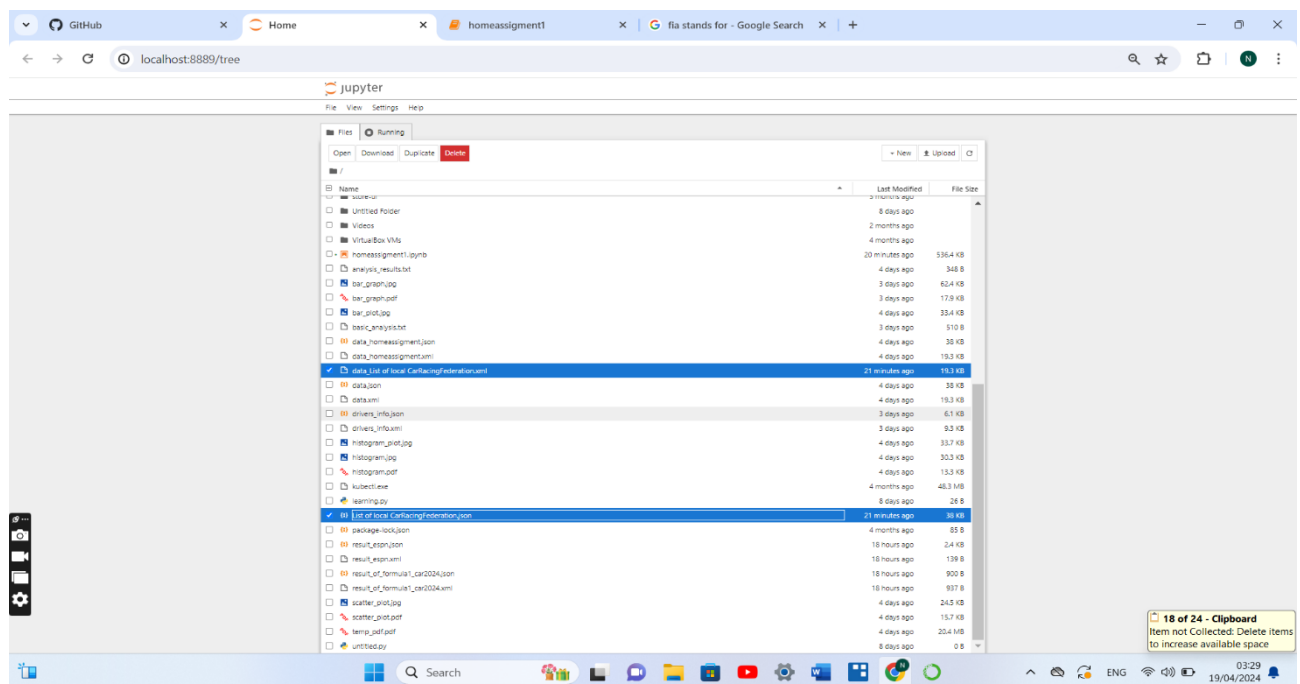
put data in dataframe

Remove unnecessary rows which is not required

Save DataFrame to JSON

Convert DataFrame to XML

These are also commented in the main program



2. next to full fill the other requirement. The extracted from next source which is

https://www.fia.com/sites/default/files/brochure_sport_grant_programme_a4_v8_web.pdf which was pdf it contains information of grant

and funds by FIA in 2020 and 2021 based on region and based on pillar division to local car racing federation and then this data first extracted and processed for further analysis here mean, median , mode correlation function have been performed data of 2020 and 2021 first merged based on region and also based on pillar division then comparison made this analysis for visual representation stored in graph such as Bar Graph, histogram, scatter plot is done this visual analysis is stored both in jpg and pdf format with the following name

“bar_plot_funding based on region 2020 and 2021.jpg”,

“histogram_plot_funding based on region 2020 and 2021.jpg”,

“scatter_plot_funding based on region 2020 and 2021”.jpg,
“bar_plot_funding based on region 2020 and 2021”.pdf,
“histogram_plot_funding based on region 2020 and 2021”.pdf,
“scatter_plot_funding based on region 2020 and 2021.jpg” have been stored.

Basic analysis such as mean, median, mode, correlation, and statistical analysis are stored in a text file with the “basic analysis.txt” For these all data extracted from pdf this pdf is also stored which was extracted from the source with the name “temp.pdf”.

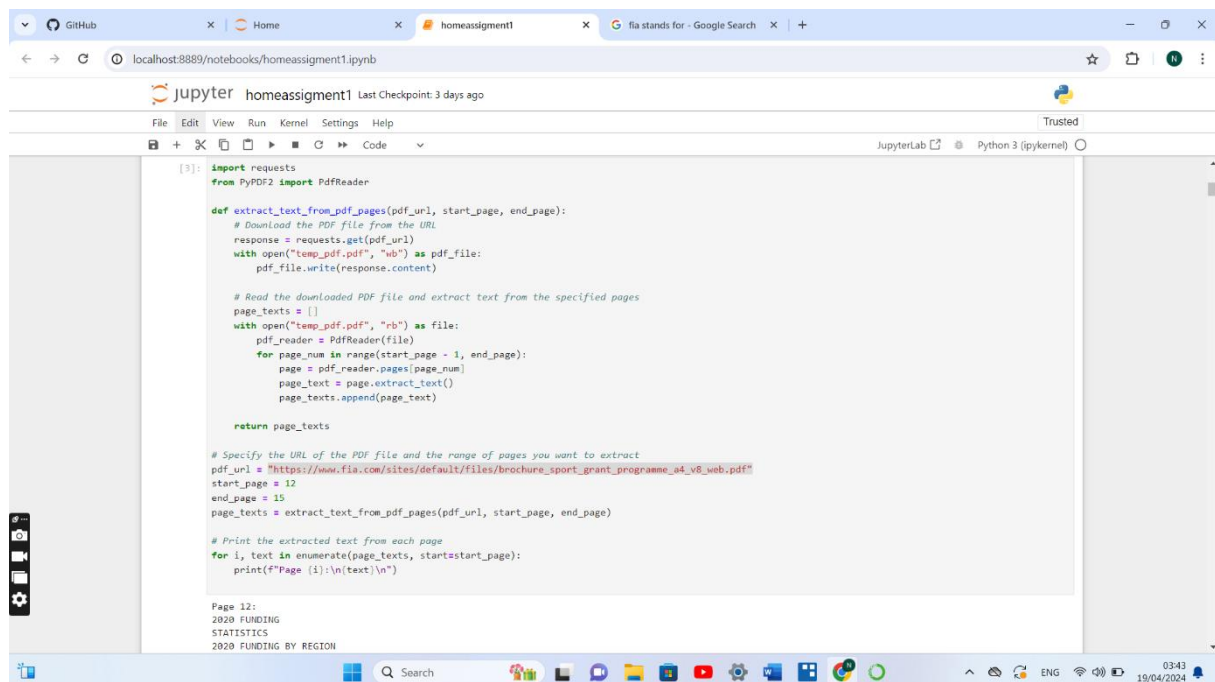
The steps used for this are

data extracted from the source which was pdf

Then download the PDF file from the URL

Read the downloaded PDF file and extract text from the specified pages

Print the extracted text from each page



```
[3]: import requests
from PyPDF2 import PdfReader

def extract_text_from_pdf_pages(pdf_url, start_page, end_page):
    # Download the PDF file from the URL
    response = requests.get(pdf_url)
    with open("temp_pdf.pdf", "wb") as pdf_file:
        pdf_file.write(response.content)

    # Read the downloaded PDF file and extract text from the specified pages
    page_texts = []
    with open("temp_pdf.pdf", "rb") as file:
        pdf_reader = PdfReader(file)
        for page_num in range(start_page - 1, end_page):
            page = pdf_reader.pages[page_num]
            page_text = page.extract_text()
            page_texts.append(page_text)

    return page_texts

# Specify the URL of the PDF file and the range of pages you want to extract
pdf_url = "https://www.fia.com/sites/default/files/brochure_sport_grant_programme_04_v8_web.pdf"
start_page = 12
end_page = 15
page_texts = extract_text_from_pdf_pages(pdf_url, start_page, end_page)

# Print the extracted text from each page
for i, text in enumerate(page_texts, start=start_page):
    print(f"Page {i}: \n{text}\n")

Page 12:
2020 FUNDING
STATISTICS
2020 FUNDING BY REGION
```


The screenshot shows a JupyterLab window titled 'homeassignment1'. The code cell contains the following code:

```
# Print the table
print(table1)
```

The output of the code is a table titled 'Funding by Region in 2020 by FIA'.

	Region	Funding Amount (€)	Percentage (%)
0	Americas	466220	15.4
1	Asia Pacific	600426	19.8
2	Europe	877256	29.0
3	MENA	706160	23.4
4	Saharan Africa	375623	12.4

Below the table, the code cell shows the command to import pandas:

```
[13]: import pandas as pd
```

After this 2020 and 2021 data was merged for analysis with this preprocessing is also done.

The screenshot shows a JupyterLab window titled 'homeassignment1'. The code cell contains the following code:

```
import numpy as np # Import NumPy for handling NaN values
import matplotlib.pyplot as plt
import seaborn as sns
from fpdf import FPDF

# Set use_inf_as_na option explicitly
pd.set_option('mode.use_inf_as_na', True)

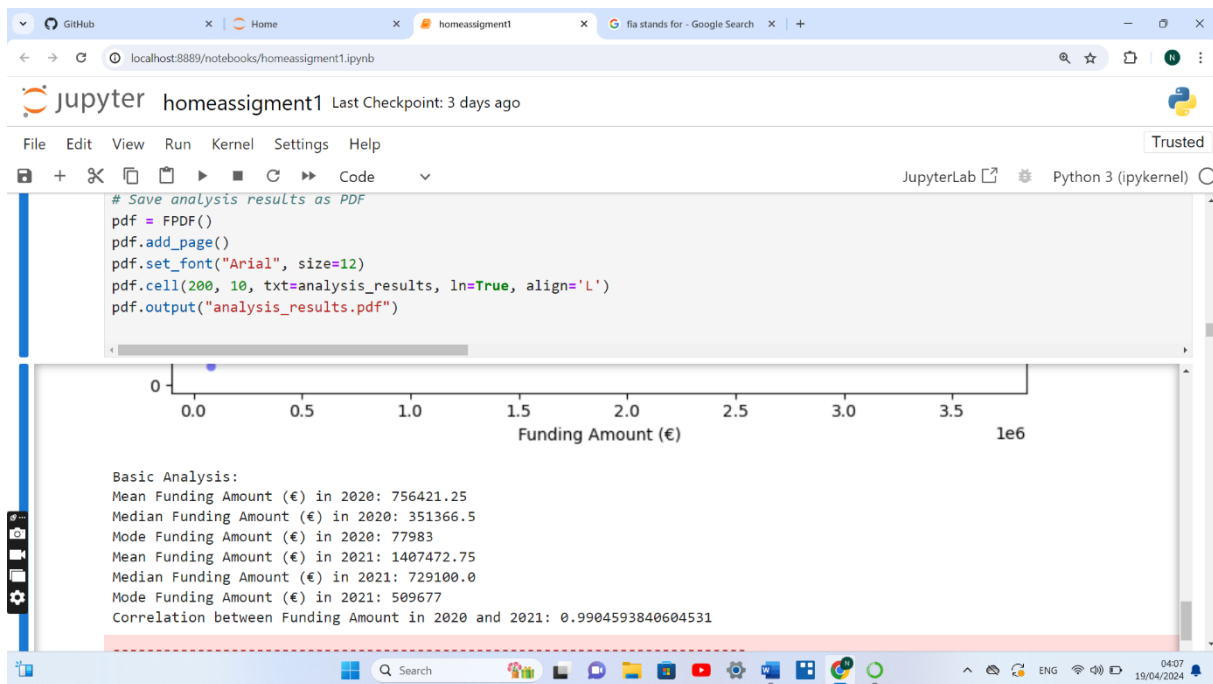
# Define the data for both tables
data_2020 = {
    'Region': ['Americas', 'Asia Pacific', 'Europe', 'MENA', 'Saharan Africa'],
    'Funding Amount (€)': [466220, 600426, 877256, 706160, 375623],
    'Percentage (%)': [15.4, 19.8, 29.0, 23.4, 12.4]
}

data_2021 = {
    'Region': ['Americas', 'Asia Pacific', 'Europe', 'MENA', 'Sub-Saharan Africa'],
    'Funding Amount (€)': [933493, 846304, 1807964, 1126642, 915488],
    'Percentage (%)': [16.6, 15.0, 32.1, 20.0, 16.3]
}

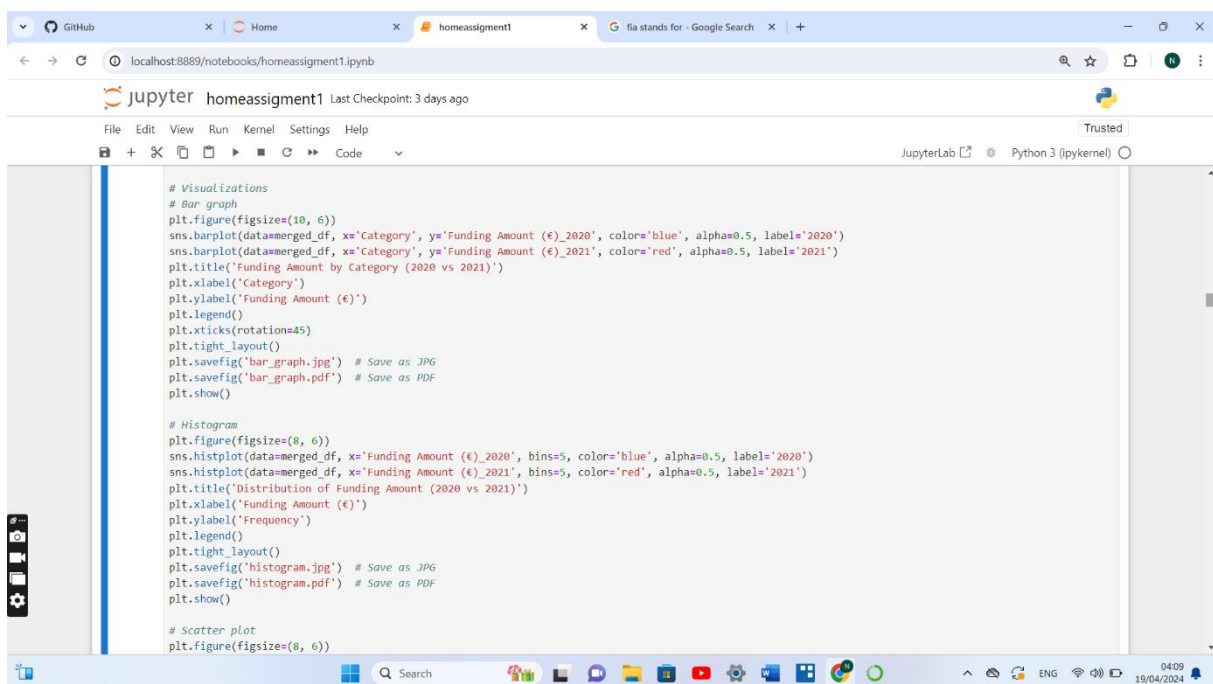
# Convert data to DataFrames
df_2020 = pd.DataFrame(data_2020)
df_2021 = pd.DataFrame(data_2021)

# Merge the two tables
merged_df = pd.merge(df_2020, df_2021, on='Region', suffixes=('_2020', '_2021'))

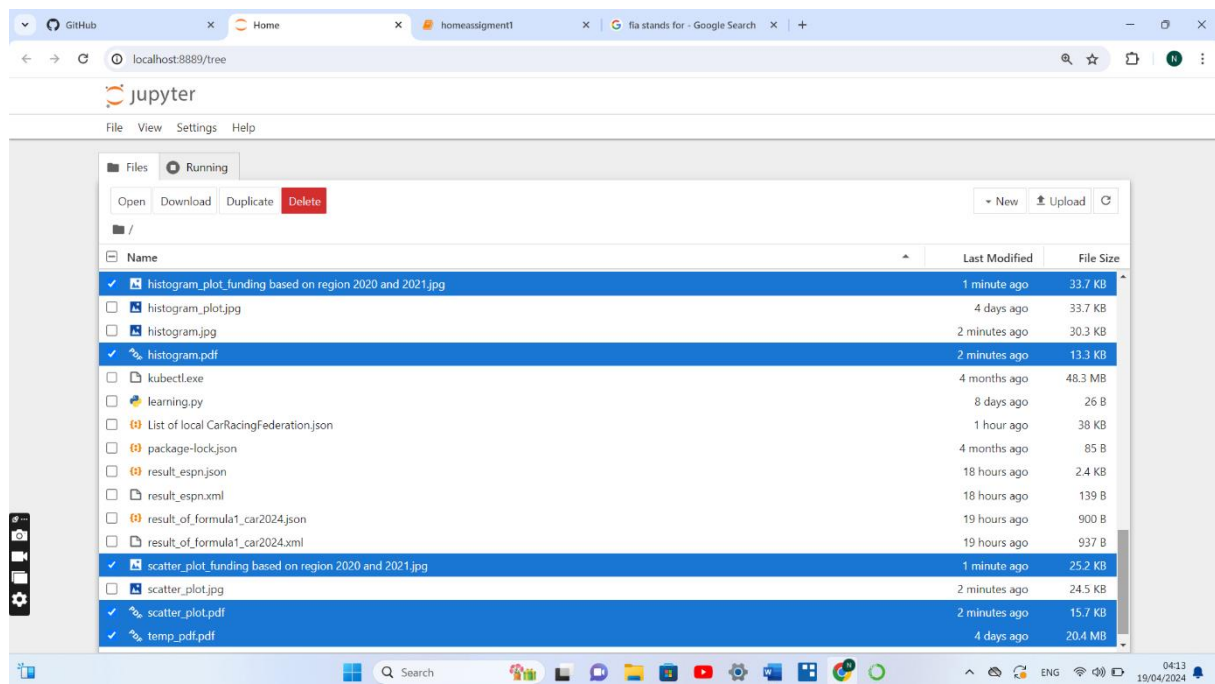
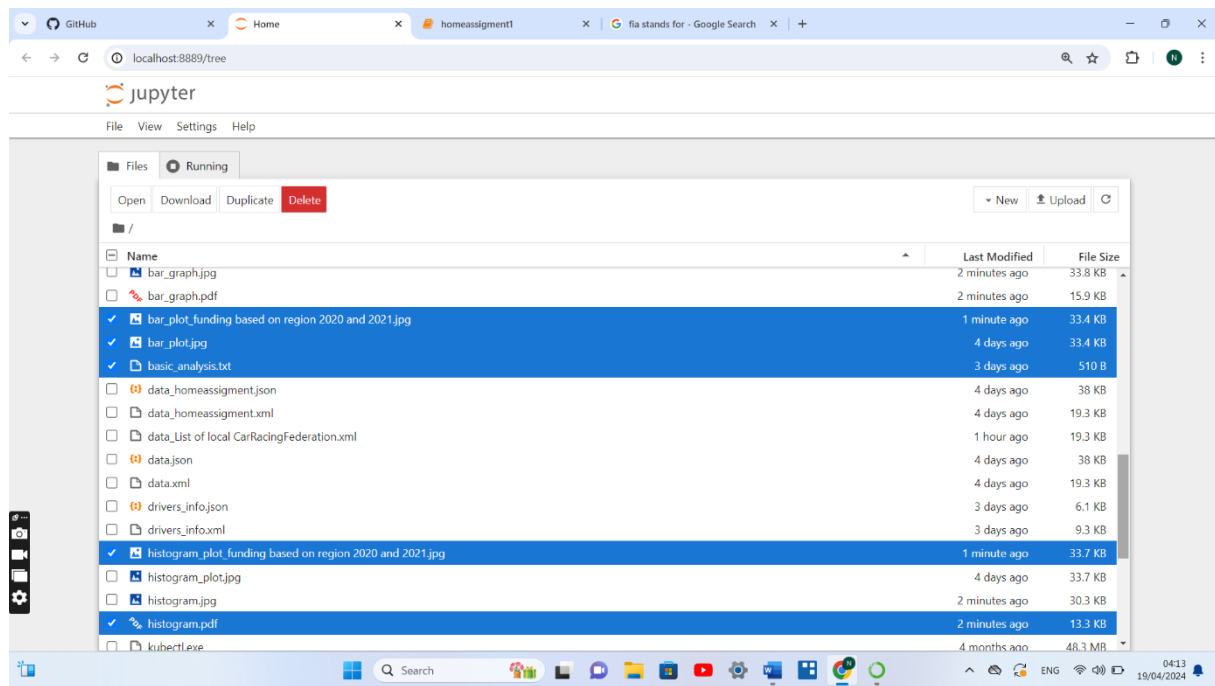
# Replace infinite values with NaN
merged_df.replace([np.inf, -np.inf], np.nan, inplace=True)
```



After this data visualisation task was performed



Results are stored in above mentioned file in both format

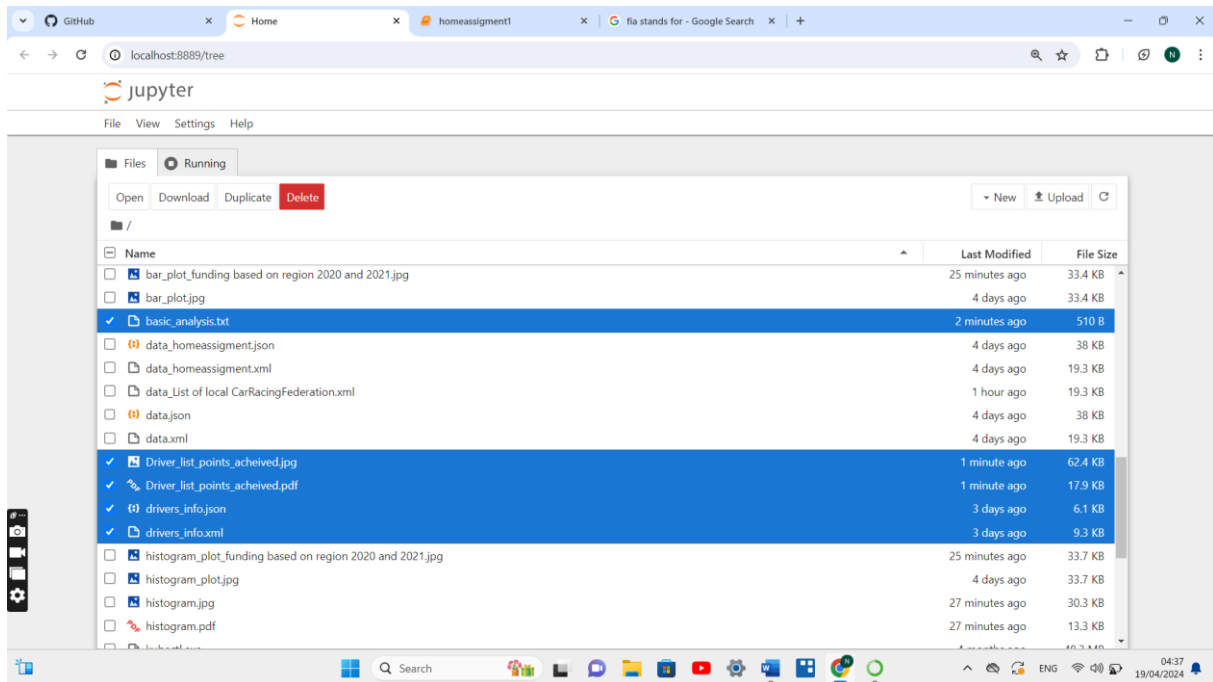


3. then data extracted from 3rd data source list of drivers for participate in local or internation car racing race in different races their performance are first stored in data frame to make it more representable data stored in table then for analysis graph have been formed this data stored in both xml and json format.

Source:- <https://www.motorsport.com/f1/results/2024/japanese-gp-639931/>

Graph stored with name:-Driver_list_points_acheived.jpg,
Driver_list_points_acheived.pdf

Data extracted stored in json and xml with the name :-
drivers_info.xml", drivers_info.json



4.Then at the end of the assignment two data set is extracted from two sources for results of international car racing “Formula 1 “ 2024 and other car racing results such as “Esn”which is supported and govern at local level the data was extracted and stored both as xml and json format to understand better data extracted converted into data frame for better analysis.

Source:- <https://www.formula1.com/en/results.html>

Data stored with the file name:- result_of_formula1_car2024.json,
result_of_formula1_car2024.xml.

The screenshot shows a JupyterLab interface with a notebook titled 'homeassignment1'. The code in the notebook is as follows:

```
# Initialize a list to store the data
data = []

# Loop through each row and extract the text content of each cell
for row in rows:
    cells = row.find_all(['th', 'td'])
    row_data = [cell.get_text(strip=True) for cell in cells]
    data.append(row_data)

# Save data to JSON file
with open('result_of_formula1_car2024.json', 'w') as json_file:
    json.dump(data, json_file, indent=4)

# Create XML structure
root = ET.Element("Formula1Results")
for row_data in data:
    row_element = ET.SubElement(root, "Row")
    for index, cell_data in enumerate(row_data):
        cell_element = ET.SubElement(row_element, f"cell{index}")
        cell_element.text = cell_data

# Save data to XML file
tree = ET.ElementTree(root)
tree.write("result_of_formula1_car2024.xml")

print("Data successfully in JSON and XML formats.")
else:
    print('Table not found.')
print('Failed to retrieve webpage. Status code:', response.status_code)
```

Other data source as mentioned above

Source:-“ <https://www.espn.com/racing/results>

Data stored :- result_espn.json, result_espn.xml

The screenshot shows a JupyterLab interface with a notebook titled 'homeassignment1'. The code in the notebook is as follows:

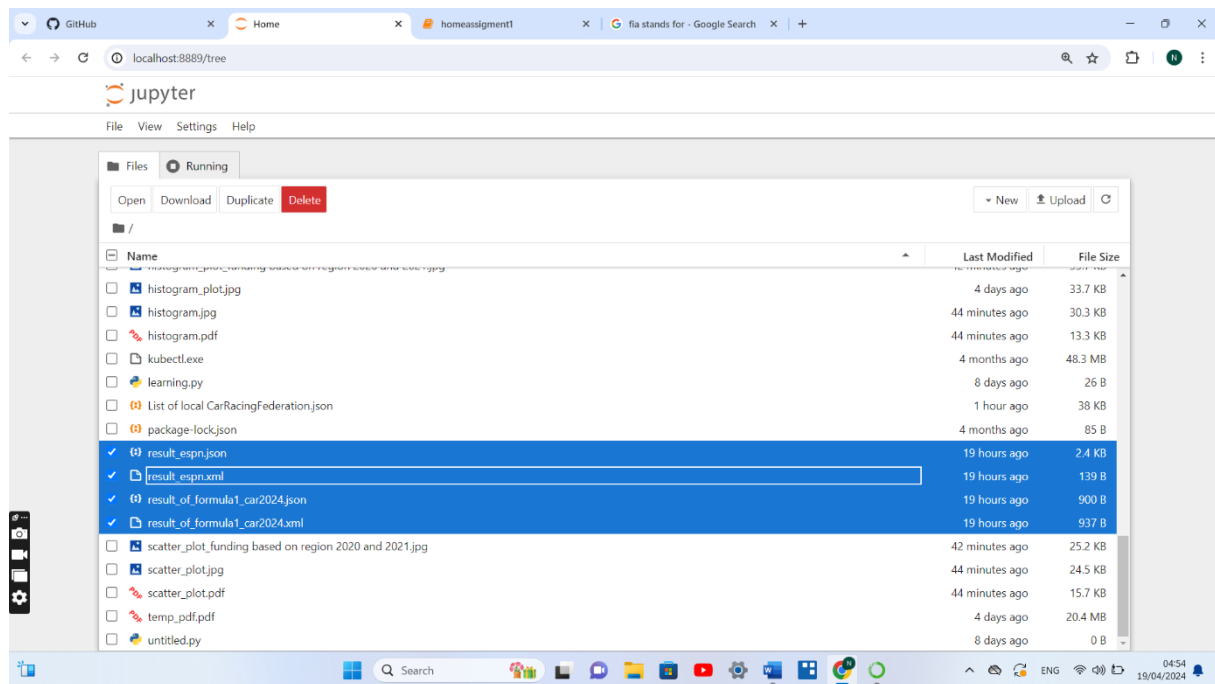
```
headers = [th.text.strip() for th in table.find_all("th")]

# Extract the rows
rows = []
for tr in table.find_all("tr"):
    row = [td.text.strip() for td in tr.find_all("td")]
    if row:
        rows.append(row)

# Display the result
for row in rows:
    print(row)

# Store the data in JSON format
data_json = {"headers": headers, "data": rows}
with open("result_espn.json", "w") as json_file:
    json.dump(data_json, json_file, indent=4)
print("Data stored in JSON format.")

# Store the data in XML format
root = ET.Element("data")
for row in rows:
    entry = ET.SubElement(root, "entry")
    for i, header in enumerate(headers):
        ET.SubElement(entry, header).text = row[i]
tree = ET.ElementTree(root)
tree.write("result_espn.xml")
print("Data stored in XML format.")
else:
    print("Table with class 'tablehead' not found.")
```



Overall there were 5 data sources were collected and data were analyzed from different data sources as per the requirement and stored in different file formats as required to represent different features of analysis, such as funds driver information local and international car racing results or a list of international and local car racing federation and clubs are represented in the task which was performed.