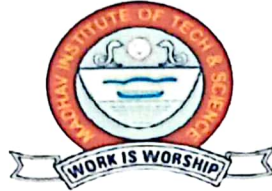


Madhav Institute of Technology and Science

A Govt. Aided UGC Autonomous Institute, Affiliated To R.G.P.V. BHOPAL (M.P.), INDIA NAAC
Accredited with A++ Grade



A Minor Project

Submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

In

ENGINEERING MATHEMATICS & COMPUTING

Submitted by

**Aryan Bhadoriya(0901MC211030), Harsh Kushwaha(0901MC211030), Harshraj singh
Dangi(0901MC211031), Hradyansh Shakya(0901MC211032), Jitendra singh
Parihar(0901MC211033), Khushi Soni (0901MC211036)**

Under the Guidance of

**Dr. Atul Kumar Ray
Assistant Professor**

DEPARTMENT OF ENGINEERING MATHEMATICS & COMPUTING

MADHAV INSTITUTE OF TECHNOLOGY AND SCIENCE

GWALIOR, 474005 M.P., INDIA

PAPER NAME

hradyansh for plagrism.pdf

WORD COUNT

1767 Words

CHARACTER COUNT

9626 Characters

PAGE COUNT

24 Pages

FILE SIZE

2.9MB

SUBMISSION DATE

Dec 14, 2023 2:51 PM GMT+5:30

REPORT DATE

Dec 14, 2023 2:51 PM GMT+5:30

● 6% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 4% Internet database
- 0% Publications database
- Crossref database
- Crossref Posted Content database
- 5% Submitted Works database

● Excluded from Similarity Report

- Bibliographic material

ARYAN BHADORIA

HARSH KUSHWAHA

HARSH RAJ Singh Dangar

HRADYANSH SHARMA

Jitendra Singh PARIHAR

Khushi Soni

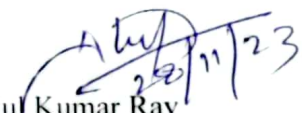
20/12/23.

Summary

MADHAV INSTITUTE OF TECHNOLOGY AND SCIENCE GWALIOR

Certificate

This to certify that Minor Project entitled "OLYMPIC DATA SET" which is being submitted by Aryan Bhadoriya (0901MC211016) for the award of degree of **Bachelor of Technology** degree in Engineering Mathematics and Computing, MITS Gwalior is a record of benefited work carried out by him under my supervision. This dissertation has reached the standard fulfilling the requirements of the regulations relating to the degree.



Dr. Atul Kumar Ray
Assistant Professor

Department of Engineering Mathematics and Computing
Madhav Institute of Technology and Science Gwalior

MADHAV INSTITUTE OF TECHNOLOGY AND SCIENCE GWALIOR

Certificate

This to certify that Minor Project entitled "OLYMPIC DATA SET" which is being submitted by Harsh Kushwaha (0901MC211030) for the award of degree of **Bachelor of Technology** degree in Engineering Mathematics and Computing. MITS Gwalior is a record of benefited work carried out by him under my supervision. This Minor Project has reached the standard fulfilling the requirements of the regulations relating to the degree.

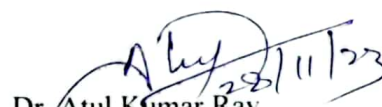

Dr. Atul Kumar Ray
Assistant Professor

Department of Engineering Mathematics and Computing
Madhav Institute of Technology and Science Gwalior

MADHAV INSTITUTE OF TECHNOLOGY AND SCIENCE GWALIOR

Certificate

This to certify that Minor Project entitled "OLYMPIC DATA SET" which is being submitted by Harshraj Singh Dangi (0901MC211031) for the award of degree of **Bachelor of Technology** degree in Engineering Mathematics and Computing, MITS Gwalior is a record of benefited work carried out by him under my supervision. This dissertation has reached the standard fulfilling the requirements of the regulations relating to the degree.



Dr. Atul Kumar Ray
Assistant Professor

Department of Engineering Mathematics and Computing
Madhav Institute of Technology and Science Gwalior

MADHAV INSTITUTE OF TECHNOLOGY AND SCIENCE GWALIOR

Certificate

This to certify that Minor Project entitled "OLYMPIC DATA SET" which is being submitted by Hradyansh Shakya (0901MC211032) for the award of degree of **Bachelor of Technology** degree in Engineering Mathematics and Computing, MITS Gwalior is a record of benefited work carried out by him under my supervision. This dissertation has reached the standard fulfilling the requirements of the regulations relating to the degree.



Dr. Atul Kumar Ray
Assistant Professor

Department of Engineering Mathematics and Computing
Madhav Institute of Technology and Science Gwalior

MADHAV INSTITUTE OF TECHNOLOGY AND SCIENCE GWALIOR

Certificate

This to certify that Minor Project entitled "OLYMPIC DATA SET" which is being submitted by Jitendra Singh Parihar (0901MC211033) for the award of degree of Bachelor of Technology degree in Engineering Mathematics and Computing, MITS Gwalior is a record of benefited work carried out by him under my supervision. This dissertation has reached the standard fulfilling the requirements of the regulations relating to the degree.



Dr. Atul Kumar Ray
Assistant Professor

Department of Engineering Mathematics and Computing
Madhav Institute of Technology and Science Gwalior

MADHAV INSTITUTE OF TECHNOLOGY AND SCIENCE GWALIOR

Certificate

This to certify that Minor Projectentitled "OLYMPIC DATA SET" which is being submitted by Khushi Soni (0901MC211036) for the award of degree of Bachelor of Technology degree in Engineering Mathematics and Computing, MITS Gwalior is a record of benefited work carried out by her under my supervision. This dissertation has reached the standard fulfilling the requirements of the regulations relating to the degree.


Dr. Atul Kumar Ray
Assistant Professor

Department of Engineering Mathematics and Computing
Madhav Institute of Technology and Science Gwalior

Acknowledgement

It is our great pleasure to express sincere gratitude to my supervisor, **Dr. Atul Kumar Ray** for his expert guidance and constant encouragement. We acknowledge that it is because of his interest that we enjoyed working on this project and express my earnest and heartfelt thanks to him for his time, support and efforts.

We are also thankful to all the faculties of the **Department of Engineering Mathematics and Computing** for their encouragement, who had invested their valuable time in providing their feedback with a lot of useful suggestions.

We are highly obliged to all my friends for their encouragement and for helping me at the points where I got stuck. I am deeply indebted to all of them for always helping and inspiring me.



Aryan Bhadoriya
(0901MC211016)



Harsh Kushwaha
(0901MC211030)



Harshraj Singh Dangi
(0901MC211031)



Hradyansh Shakya
(0901MC211032)



Jitendra Singh Parihar
(0901MC211033)



Khushi Soni
(0901MC211036)

INDEX

1. Introduction	
1.1. Purpose	
1.2. Overview	
1.3. Research Gaps	
2. System Design and Formulation	
2. Functional Requirement	
2.1. Pandas	
2.1.1 Introduction to Pandas	
2.1.2 Uses of Pandas	
2.2. Jupyter Notebook	
2.2.1 Introduction to Jupyter Notebook	
2.2.2 Uses of Jupyter Notebook	
2.3. Numpy	
2.3.1 Introduction to Numpy	
2.3.2 Uses of Numpy	
2.4. Streamlit	
2.5. Seaborn	
3. Methodology	
3.2. Coding	
3.3. Software Testing	

4. Result and Conclusion

4.1. Result.....

4.2. Conclusion

5. Future Scope **40**

Reference.....

INTRODUCTION

1. Introduction

The Winter Olympic Games, or simply Olympic games, is an international sports contest consisting of winter and summer sports contest, in which many athletes from all over the world participates in various contests held. The Olympics are said to be the most important sports contest in the world, gaining international recognition with the participation of more than 200 countries. There are a number of data sets related to the Olympics that provide information on various aspects of the Games, including historical results, athlete profiles, and more. These data sets are valuable for conducting various analyses, research, and data visualizations related to the Olympic Games.

1.1 Purpose

We used a variety of methods here to analyze Olympic data, including collecting data from standards such as medal count, athlete overall performance and Olympic stats and then using different analysis techniques to uncover patterns or patterns of collected dataset.

1.2 Overview

The Olympics are undoubtedly one of the most talked about and watched sports in the sports world. Every five years, athletes of various sports all around the world assemble. These activities date of each and every faceoff have undergone changes and modifications with time. In previous years there had been the introduced use of that data and analysis to acknowledged the performance of countries and technique of facing events

1.3 Research Gap

In this article, our ambition is to found statergy of data from last 60 years regarding the event. Gathered data on trends such as the number of medals, individual sports, and social media. We have analyzed the collected data using the scientific EDA technique. This research provides a analysis of Olympic data, shedding light on the evolution of this major event. We will focus on a country's statistical results, a comprehensive analysis of Olympic events, analysis of a country's past Olympic medals and achievements, identifying athletes and finally providing news about our Olympic Games. We firmly trust that this analysis will give or demonstrate a deep insight of games and sports that had been most successful competitions over the years.

CHAPTER 2

System Design and Formulation

2. Functional Requirement

2.1 Pandas

2.1.1 Introduction to pandas:

It is the main Python library to work with data. It has functions such as examining, maintaining, searching and managing data. This is the powerful python library for data manipulation and analysis. It offers data structure like series and data-frames, making it easier to go with labelled or rational data work.

2.1.2 Uses of Pandas:

Pandas therefore allows us to analyze large data sets and draw meaningful conclusions based on statistical studies. Interestingly, Pandas can clean up complex files and make them easier to read and use. Correlation is very important in data science.

Pandas Library c remove rows that are not relevant to the data or contain invalid values such as empty or NULL values. This is technically called data sanitization.

2.2 Jupyter Notebook



Jupyter_Notebook is the first Web_Application for creating and sharing computational data. It provides a simple, fluid and data-centric experience

2.2.1 Introduction to Jupyter Notebook:

Jupyter Notebook is an open-source web application that allowing us to create and share documents with containing descriptive information as well as live numbers, equations, calculation outputs, visualizations, and other multimedia tools. Write it on a piece of paper. Interestingly, you can use Jupyter Notebooks to perform a variety of task, including transformation and data cleaning, numerical simulations, data_analytics and visualization of data, Deep_Learning & ML etc.

Jupyter Handbook provides you with an easy-to-use GUI that can be used not only as an integrated development environment (IDE) but also as a learning and presentation tool. This is a way to use Python in a virtual "notebook" or it is very popular in data scientists because of its simplicity. It gives you the opportunity to collect numbers, pictures, plans, descriptions, and more, according to the steps of the Research Process. Additionally, it is an interactive form of analysis; It is an environment where users run code, see what happens, and then modify it, creating a dialogue between research articles and archives. Data scientists can also use the guide to create manuals or interactive guides for their own versions of the software.

It has two features: Data scientists enter numbers and text into rectangular "cells" on the front page of the web page. Browser passes this program to the final "kernel" which then compiles the program and gives the results. Many Jupyter kernels have been developed that support many programming languages. The kernel does not need to be on the data scientist's computer. These books can also be run in a cloud environment such as Google Collaboration Projects (commonly known as Google Colab). You can run Jupyter on your own computer without having to log into the network and do your work very well locally.

2.2.2 Uses of Jupyter Notebooks

Jupyter is being used to do Python machine learning work. It is a great environment which can be used to communicate results and develop programs.

The name Jupyter was chosen mainly to promote scientific ideas, leadership and research methods. Additionally, the main languages accepted by Jupyter Notebook are R, Python and Julia. Although the name Jupyter is not an abbreviation of these words Julia (Ju), python (Py) and R. It creates a strong association with them.

Summarizing the Pros and Cons:

We can conclude that these Notebooks are absolutely amazing in performing tasks related to visualizations, cleansing of data, and any projects related to data science or Python in general.

Pros:

- Best Platform for getting started with data science.
- It is Easy to share the notebooks and visualizations.
- Easy Availability of mark-downs and other additional functionalities.

Cons:

- Lack of powerful features which are included in some IDE's.

2.3 NumPy

2.3.1 Introduction to Numpy

NumPy: A widely used library for manipulating arrays. It possess properties/functions that deal with linear algebra, Fourier transforms, and matrix fields. Creation of NumPy was done by Travis Oliphant in 2005. It is free to use and open source library.



2.3.2 Uses of Numpy :

- Data manipulation and analysis .
- Scientific computing and simulation.
- Image processing and computer vision.
- Signal processing and audio analysis.
- Data visualization and graphics.
- Educations and research.



Streamlit

2.4 WHAT IS STREAMLIT?

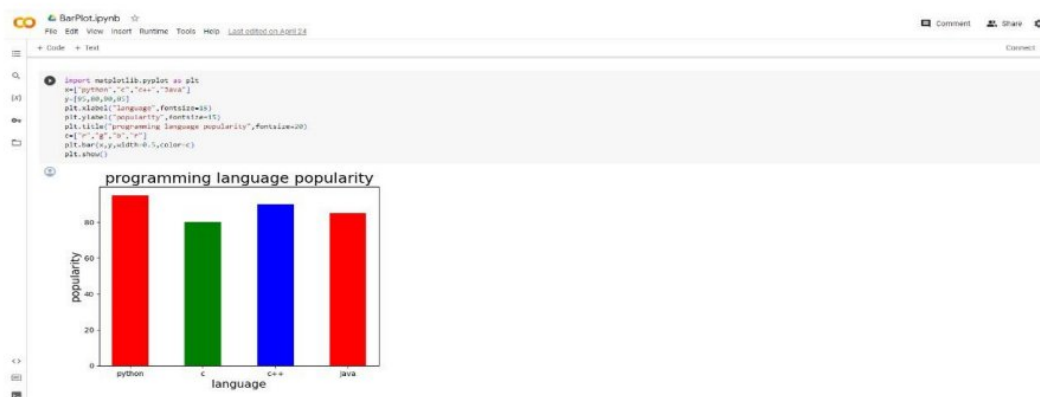
Streamlit: A python library to allows a data scientist to easily create free educational applications. Streamlit allows us to view descriptions and output models, see data and performance models, and use the sidebar to change input models from the GUI.

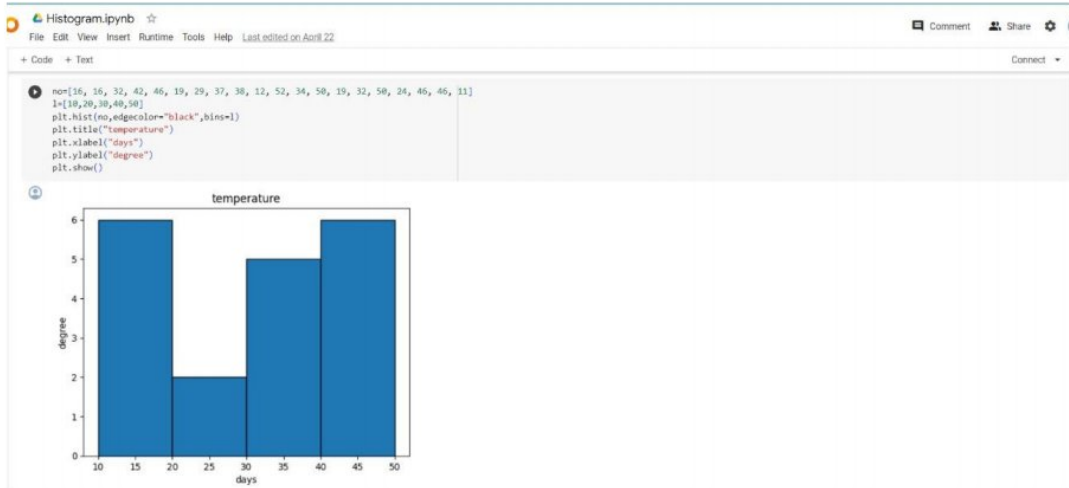
Matplotlib: The Visualization using Python

Matplotlib is a library used for creating multiple plots & creating static, dynamic & interactive plots using Python. While Matplotlib is used to make things easier for seeing, it also allows things that are hard to see to be seen.

- Create the best images.
- Create interactive charts that can be zoomed, panned and resized.
- Customize images, style and layout.
- Can be exported to various file formats.
- Integrate Jupyter-Lab and Graphical User Interface (GUI).
- It is a great Python library for Plotting two-dimensional array. It is a multi_data visualization library based on NumPy arrays, designed for comprehensive Sci-Py stack. It was proposed by John Hunter in 2002.

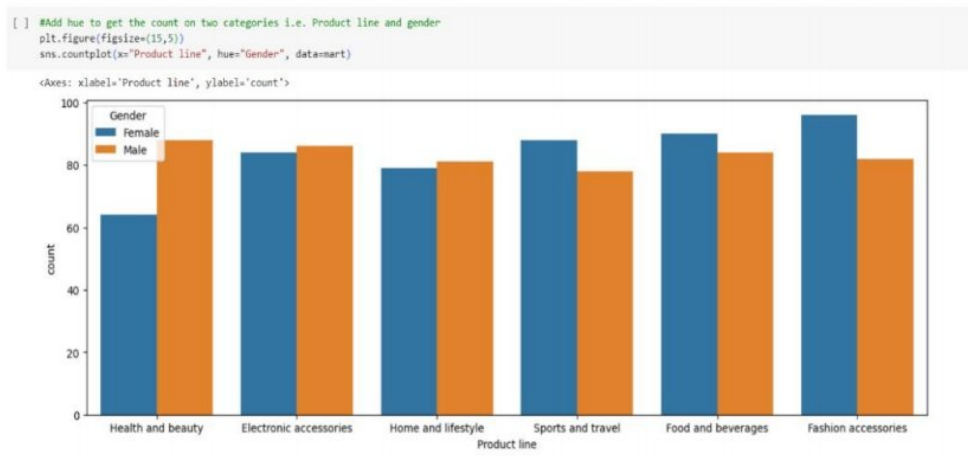
One of the biggest benefits of this visual library is that it allows us to understand a lot of information with easy understand texts. LineCharts, bar_charts, Scatter Plots, histogram, etc. in it. There may be graphics such as.





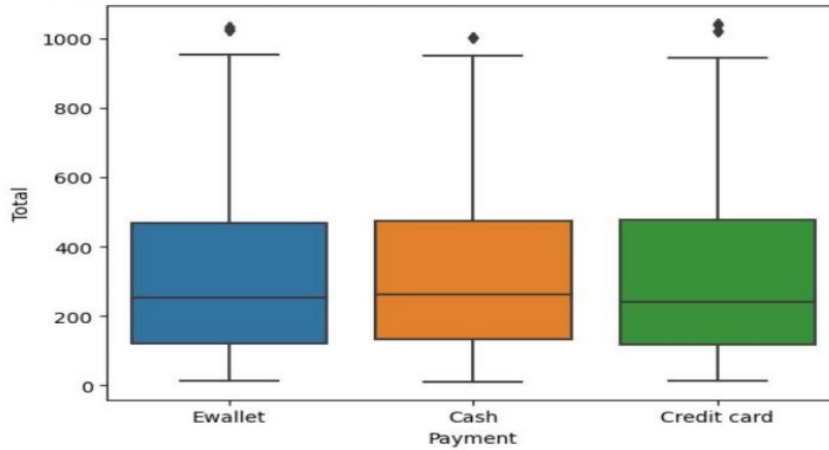
2.5 SEABORN:

Seaborn is a Python data visualization library built on top of Matplotlib. It provides a high-level interface for creating attractive and informative statistical graphics. Seaborn simplifies the process of generating complex visualization by offering function for various plot types such as scatter plots, bar plots, histogram, heatmaps and more



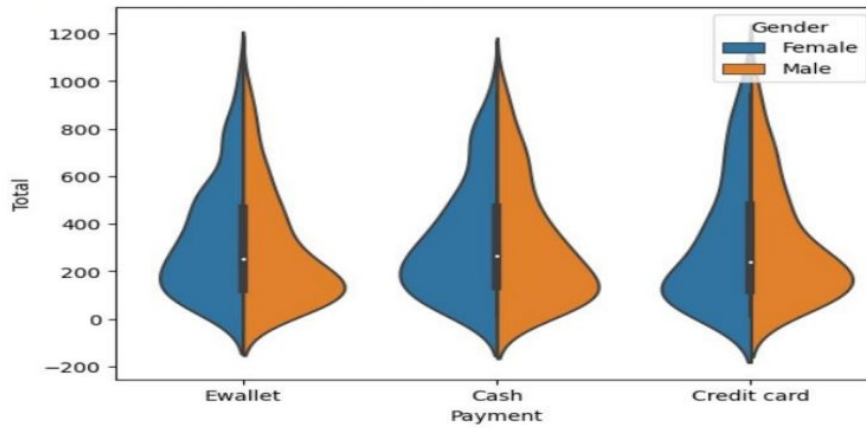
```
#create a Box plot on one of the numeric variable by a categorical value
sns.boxplot(x='Payment', y='Total', data=mart)
```

```
<Axes: xlabel='Payment', ylabel='Total'>
```



```
#create a VIOLINPLOT on two categorical and one numeric variable and use split
sns.violinplot(x='Payment', y='Total', hue='Gender', data=mart, split=True)
```

```
<Axes: xlabel='Payment', ylabel='Total'>
```



CHAPTER 3

CODING

APP.py

```
Project v
> olympics-analysis-web-app C:\Users\Acer\ML Projects
> External Libraries
Scratches and Consoles

helper.py app.py x main.py preprocessor.py
1 import streamlit as st
2 import pandas as pd
3 import numpy as np
4 import preprocessor
5 import helper
6 import plotly.express as px
7 import matplotlib.pyplot as plt
8 import seaborn as sns
9 import plotly.figure_factory as ff
10 from plotly.figure_factory import create_distplot
11
12 df = pd.read_csv('athlete_events.csv')
13 region_df = pd.read_csv('noc_regions.csv')
14
15 df = preprocessor.preprocess(df, region_df)
16
17 st.sidebar.title("Olympics Analysis")
18 user_menu = st.sidebar.radio(
19     'Select an Option',
20     ('Medal Tally', 'Overall Analysis', 'Country-wise Analysis', 'Athlete wise Analysis')
21 )
22 st.title("Olympics Data Model")
23 st.dataframe(df)
24
```

```
Project v
> olympics-analysis-web-app C:\Users\Acer\ML Projects
> External Libraries
Scratches and Consoles

helper.py app.py x main.py preprocessor.py
25 if user_menu == 'Medal Tally':
26     st.sidebar.header("Medal Tally")
27     years, country = helper.country_year_list(df)
28
29     selected_year = st.sidebar.selectbox("Select Year", years)
30     selected_country = st.sidebar.selectbox("Select Country", country)
31
32     medal_tally = helper.fetch_medal_tally(df, selected_year, selected_country)
33     if selected_year == 'Overall' and selected_country == 'Overall':
34         st.title("Overall Tally")
35     if selected_year != 'Overall' and selected_country == 'Overall':
36         st.title("Medal Tally in " + str(selected_year) + " Olympics")
37     if selected_year == 'Overall' and selected_country != 'Overall':
38         st.title(selected_country + "Overall Performance")
39     if selected_year != 'Overall' and selected_country != 'Overall':
40         st.title(selected_country + " performance in " + str(selected_year) + " Olympics")
41     st.table(medal_tally)
42
```

```

Project
  olympics-analysis-web-app C:\Users\Acer\ML Projects
  External Libraries
  Scratches and Consoles

helper.py  app.py  main.py  preprocessor.py
43
44 if user_menu == 'Overall Analysis':
45     editions = df['Year'].unique().shape[0]-1
46     cities = df['City'].unique().shape[0]
47     sports = df['Sport'].unique().shape[0]
48     events = df['Event'].unique().shape[0]
49     athletes = df['Name'].unique().shape[0]
50     nations = df['region'].unique().shape[0]
51
52     st.title("Top Statistics")
53     col1,col2,col3 = st.columns(3)
54     with col1:
55         st.header("Editions")
56         st.title(editions)
57     with col2:
58         st.header("Host")
59         st.title(cities)
60     with col3:
61         st.header("Sports")
62         st.title(sports)
44

```

```

Project
  olympics-analysis-web-app C:\Users\Acer\ML Projects
  External Libraries
  Scratches and Consoles

helper.py  app.py  main.py  preprocessor.py
64 col1, col2, col3 = st.columns(3)
65 with col1:
66     st.header("Events")
67     st.title(events)
68 with col2:
69     st.header("Nations")
70     st.title(nations)
71 with col3:
72     st.header("Athletes")
73     st.title(athletes)
74
75

```

```

Project
  olympics-analysis-web-app C:\Users\Acer\ML Projects
  External Libraries
  Scratches and Consoles

helper.py  app.py  main.py  preprocessor.py
76 nations_over_time = helper.data_over_time(df, col: 'region')
77 st.title("Participating nations over the Year")
78 fig = px.line(nations_over_time, x="Edition", y="region")
79 st.plotly_chart(fig)
80
81 events_over_time = helper.data_over_time(df, col: 'Event')
82 st.title("Events over the Year")
83 fig = px.line(events_over_time, x="Edition", y="Event")
84 st.plotly_chart(fig)
85
86 athlete_over_time = helper.data_over_time(df, col: 'Event')
87 st.title("Events over the Year")
88 fig = px.line(athlete_over_time, x="Edition", y="Event")
89 st.plotly_chart(fig)
90
91 st.title("No. of Events Over time(Every Sport)")
92 fig, ax = plt.subplots(figsize=(20, 20))
93 x = df.drop_duplicates(['Year', 'Sport', 'Event'])
94 ax = sns.heatmap(x.pivot_table(index='Sport', columns='Year', values='Event', aggfunc=
95     annot=True)
96 st.pyplot(fig)
97

```

```

Project
  olympics-analysis-web-app C:\Users\Acer\ML Projects
  External Libraries
  Scratches and Consoles

helper.py  app.py  main.py  preprocessor.py
106
107 if user_menu == 'Country-wise Analysis':
108
109     st.sidebar.title('Country-wise Analysis')
110     country_list = df['region'].dropna().unique().tolist()
111     country_list.sort()
112
113     selected_country = st.sidebar.selectbox('Select a Country', country_list)
114
115     country_df = helper.yearwise_medal_tally(df, selected_country)
116
117     fig = px.line(country_df, x="Year", y="Medal")
118     st.title(selected_country + "Medal Tally over the years")
119     st.plotly_chart(fig)
120
121     st.title(selected_country + " excels in the following sports")
122     pt = helper.country_event_heatmap(df, selected_country)
123     fig, ax = plt.subplots(figsize=(20, 20))
124     ax = sns.heatmap(pt, annot=True)
125     st.pyplot(fig)
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157

```

```

Project
  olympics-analysis-web-app C:\Users\Acer\ML Projects
  External Libraries
  Scratches and Consoles

helper.py  app.py  main.py  preprocessor.py
127
128 if user_menu == 'Athlete wise Analysis':
129     athlete_df = df.drop_duplicates(subset=['Name', 'region'])
130
131     x1 = athlete_df['Age'].dropna()
132     x2 = athlete_df[athlete_df['Medal'] == 'Gold']['Age'].dropna()
133     x3 = athlete_df[athlete_df['Medal'] == 'Silver']['Age'].dropna()
134     x4 = athlete_df[athlete_df['Medal'] == 'Bronze']['Age'].dropna()
135
136     fig = ff.create_distplot(hist_data=[x1, x2, x3, x4], group_labels=['Overall Age', 'Gold Me
137     fig.update_layout(autosize=False, width=1000, height=1000)
138     st.title("Distribution of Age")
139     st.plotly_chart(fig)
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157

```

```

Project
  olympics-analysis-web-app C:\Users\Acer\ML Projects
  External Libraries
  Scratches and Consoles

helper.py  app.py  main.py  preprocessor.py
139
140 st.plotly_chart(fig)
141
142 x = []
143 name = []
144 famous_sports = ['Basketball', 'Judo', 'Football', 'Tug-Of-War', 'Athletics', 'Swimming
145                 'Gymnastics', 'Art Competitions', 'Handball', 'Rowing', 'Fencing', 'S
146                 'Taekwondo', 'Cycling', 'Diving', 'Canoeing', 'Tennis', 'Golf', 'Soft
147                 'Volleyball', 'Synchronized Swimming', 'Table Tennis', 'Baseball', 'R
148                 'Rugby Sevens', 'Beach Volleyball', 'Triathlon', 'Rugby', 'Polo', 'Ic
149
150 for sport in famous_sports:
151     temp_df = athlete_df[athlete_df['Sport'] == sport]
152     x.append(temp_df[temp_df['Medal'] == 'Gold']['Age'].dropna())
153     name.append(sport)
154
155 fig = ff.create_distplot(x, name, show_hist=False, show_rug=False)
156 fig.update_layout(autosize=False, width=1000, height=600)
157 st.title("Distribution of Age wrt Sports(Gold Medalist)")
158 st.plotly_chart(fig)
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177

```

```

158
159 sport_list = df['Sport'].unique().tolist()
160 sport_list.sort()
161 sport_list.insert(0, 'Overall')
162
163 selected_sport = st.selectbox('Select a Sport', sport_list)
164 temp_df = helper.weight_v_height(df, selected_sport)
165 fig, ax = plt.subplots()
166 ax = sns.scatterplot(temp_df['Weight'])# hue=temp_df['Medal'], style=temp_df['Sex'],
167 st.pyplot(fig)
168
169 st.title("Men Vs Women Participation over the Year")
170 final = helper.men_vs_women(df)
171 fig = px.line(final, x="Year", y=["Male", "Female"])
172 fig.update_layout(autosize=False, width=1000, height=600)
173 st.plotly_chart(fig)
174

```

PREPROCESSOR.py

```

Project | helper.py | app.py | main.py | preprocessor.py x
olympics-analysis-web-app C:\Users\Acer\ML Projects
External Libraries
Scratches and Consoles
1 import pandas as pd
2
3 usage
4 def preprocess(df, region_df):
5     #filtering the summer olympics
6     df = df[df['Season'] == 'Summer']
7     #merge with region_df
8     df = df.merge(region_df, on='NOC', how='left')
9     #dropping duplicates
10    df.drop_duplicates(inplace=True)
11    #one hot encoding medals
12    df = pd.concat([df, pd.get_dummies(df['Medal'])], axis=1)
13    return df

```

HELPER.py

```

Project | helper.py x | app.py | main.py | preprocessor.py
olympics-analysis-web-app C:\Users\Acer\ML Projects
External Libraries
Scratches and Consoles
70 usage
71 def yearwise_medal_tally(df, country):
72     temp_df = df.dropna(subset=['Medal'])
73     temp_df.drop_duplicates(subset=['Team', 'NOC', 'Games', 'Year', 'City', 'Sport', 'Event'])
74     new_df = temp_df[temp_df['region'] == country]
75     final_df = new_df.groupby('Year').count()['Medal'].reset_index()
76     return final_df
77
78 usage
79 def country_event_heatmap(df, country):
80     temp_df = df.dropna(subset=['Medal'])
81     temp_df.drop_duplicates(subset=['Team', 'NOC', 'Games', 'Year', 'City', 'Sport', 'Event'])
82     new_df = temp_df[temp_df['region'] == country]
83
84     pt = new_df.pivot_table(index='Sport', columns='Year', values='Medal', aggfunc='count')
85     return pt
86

```

```

Project > olympics-analysis-web-app C:\Users\Acer\ML Projects
> External Libraries
Scratches and Consoles

helper.py x app.py main.py preprocessor.py
1 import numpy as np
2
3 1 usage
4 def fetch_medal_tally(df, year, country):
5     medal_df = df.drop_duplicates(subset=['Team', 'NOC', 'Games', 'Year', 'City', 'Sport', 'Event', 'Medal'])
6     flag = 0
7     if year == 'Overall' and country == 'Overall':
8         temp_df = medal_df
9     if year == 'Overall' and country != 'Overall':
10        flag = 1
11        temp_df = medal_df[medal_df['region'] == country]
12    if year != 'Overall' and country == 'Overall':
13        temp_df = medal_df[medal_df['Year'] == int(year)]
14    if year != 'Overall' and country != 'Overall':
15        temp_df = medal_df[(medal_df['Year'] == year) & (medal_df['region'] == country)]
16

```

```

Project > olympics-analysis-web-app C:\Users\Acer\ML Projects
> External Libraries
Scratches and Consoles

helper.py x app.py main.py preprocessor.py
15
16
17 if flag==1:
18     x = temp_df.groupby('Year').sum()[['Gold', 'Silver', 'Bronze']].sort_values('Year')
19 else:
20     x = temp_df.groupby('region').sum()[['Gold', 'Silver', 'Bronze']].sort_values('Gold')
21 x['total'] = x['Gold'] + x['Silver'] + x['Bronze']
22
23 x['Gold'] = x['Gold'].astype('int')
24 x['Silver'] = x['Silver'].astype('int')
25 x['Bronze'] = x['Silver'].astype('int')
26 x['total'] = x['total'].astype('int')
27
28 return x
29

```

```

Project > olympics-analysis-web-app C:\Users\Acer\ML Projects
> External Libraries
Scratches and Consoles

helper.py x app.py main.py preprocessor.py
30 def medal_tally(df):
31     medal_tally = df.drop_duplicates(subset=['Team', 'NOC', 'Games', 'Year', 'City', 'Sport', 'Event', 'Medal'])
32     medal_tally = medal_tally.groupby('region').sum()[['Gold', 'Silver', 'Bronze']].sort_values('Gold')
33
34     medal_tally['total'] = medal_tally['Gold'] + medal_tally['Silver'] + medal_tally['Bronze']
35
36     medal_tally['Gold'] = medal_tally['Gold'].astype('int')
37     medal_tally['Silver'] = medal_tally['Silver'].astype('int')
38     medal_tally['Bronze'] = medal_tally['Silver'].astype('int')
39     medal_tally['total'] = medal_tally['total'].astype('int')
40
41     return medal_tally
42

```

```

Project > olympics-analysis-web-app C:\Users\Acer\ML Projects
> External Libraries
Scratches and Consoles

helper.py x app.py main.py preprocessor.py
53
54 3 usages
55 def data_over_time(df, col):
56     #df.rename(columns={'Year': 'index', 'count': 'Year'}, inplace=True)
57     nations_over_time = df.drop_duplicates(['Year', col])[['Year']].value_counts().reset_index()
58     nations_over_time.rename(columns={'Year': 'Edition', 'count': col}, inplace=True)
59     return nations_over_time
60
61 def most_successful(df, sport):
62     temp_df = df.dropna(subset=['Medal'])
63
64     if sport != 'Overall':
65         temp_df = temp_df[temp_df['Sport'] == sport]
66
67     x = temp_df['Name'].value_counts().reset_index().head(15).merge(df, left_on='Year', right_on='Year')
68     x.rename(columns={'Year': 'Name', 'Name_x': 'Medals'}, inplace=True)
69     return x
70

```

```
Project v
olympics-analysis-web-app C:\Users\Acer\ML Projects
  External Libraries
  Scratches and Consoles
helper.py x app.py main.py preprocessor.py
42 1 usage
43 def country_year_list(df):
44     years = df['Year'].unique().tolist()
45     years.sort()
46     years.insert(0, 'Overall')
47
48     country = np.unique(df['region'].dropna().values).tolist()
49     country.sort()
50     country.insert(0, 'Overall')
51
52     return years, country
53
```

```
Project v
olympics-analysis-web-app C:\Users\Acer\ML Projects
  External Libraries
  Scratches and Consoles
helper.py x app.py main.py preprocessor.py
86 1 usage
87 def weight_v_height(df, sport):
88     athlete_df = df.drop_duplicates(subset=['Name', 'region'])
89     athlete_df['Medal'].fillna('No Medal', inplace=True)
90     if sport != 'Overall':
91         temp_df = athlete_df[athlete_df['Sport'] == sport]
92         return temp_df
93     else:
94         return athlete_df
95
96 1 usage
97 def men_vs_women(df):
98     athlete_df = df.drop_duplicates(subset=['Name', 'region'])
99
100     men = athlete_df[athlete_df['Sex'] == 'M'].groupby('Year').count()['Name'].reset_index
101     women = athlete_df[athlete_df['Sex'] == 'F'].groupby('Year').count()['Name'].reset_index
102
103     final = men.merge(women, on='Year', how='left')
104     final.rename(columns={'Name_x': 'Male', 'Name_y': 'Female'}, inplace=True)
105
106     final.fillna(0, inplace=True)
107
108     return final
```

STEPWISE IMPLEMENTATION

Olympics Analysis

Select an Option

Medal Tally

Overall Analysis

Country-wise Analysis

Athlete wise Analysis

Medal Tally

Select Year

Overall

Select Country

Overall

Olympics Data Model

ID	Name	Sex	Age	Height	Weight	Team	NOC	
0	1	A Dijiang	M	24	180	80	China	CHN
1	2	A Lamusi	M	23	170	60	China	CHN
2	3	Gunnar Nielsen Aaby	M	24	None	None	Denmark	DEN
3	4	Edgar Lindenau Aabye	M	34	None	None	Denmark/Sweden	DEN
4	8	Cornelia "Cor" Aalten (-Strannood)	F	18	168	None	Netherlands	NED
5	8	Cornelia "Cor" Aalten (-Strannood)	F	18	168	None	Netherlands	NED
6	10	Einar Ferdinand "Einari" Aalto	M	26	None	None	Finland	FIN
7	12	Jyri Tapani Aalto	M	31	172	70	Finland	FIN
8	13	Minna Maarit Aalto	F	30	159	55.5	Finland	FIN
9	13	Minna Maarit Aalto	F	34	159	55.5	Finland	FIN

MEDAL TALLY

×

Olympics Analysis

Select an Option

Medal Tally

Overall Analysis

Country-wise Analysis

Athlete wise Analysis

Medal Tally

Select Year

Overall ▾

Select Country

India ▾

India Overall Performance

	Year	Gold	Silver	Bronze	total
0	1900	0	2	2	2
1	1920	0	0	0	0
2	1924	0	0	0	0
3	1928	1	0	0	1
4	1932	1	0	0	1
5	1936	1	0	0	1
6	1948	1	0	0	1
7	1952	1	0	0	2
8	1956	1	0	0	1
9	1960	0	1	1	1
10	1964	1	0	0	1
11	1968	0	0	0	1
12	1972	0	0	0	1
13	1976	0	0	0	0
14	1980	1	0	0	1
15	1984	0	0	0	0
16	1988	0	0	0	0
17	1992	0	0	0	0
18	1996	0	0	0	1
19	2000	0	0	0	1

×

Olympics Analysis

Select an Option

Medal Tally

Overall Analysis

Country-wise Analysis

Athlete wise Analysis

Medal Tally

Select Year

Overall ▾

Select Country

Australia ▾

Australia Overall Performance

	Year	Gold	Silver	Bronze	total
0	1896	2	0	0	3
1	1900	3	0	0	6
2	1904	0	3	3	4
3	1906	0	0	0	3
4	1908	1	2	2	5
5	1912	2	2	2	7
6	1920	0	2	2	3
7	1924	3	1	1	6
8	1928	1	2	2	4
9	1932	3	1	1	5
10	1936	0	0	0	1
11	1948	2	6	6	13
12	1952	6	2	2	11
13	1956	13	8	8	35
14	1960	8	8	8	22
15	1964	6	2	2	18
16	1968	5	7	7	17
17	1972	8	7	7	17
18	1976	0	1	1	5
19	1980	2	2	2	9

Olympics Analysis

Select an Option

- Medal Tally
- Overall Analysis
- Country-wise Analysis
- Athlete wise Analysis

Medal Tally

Select Year

Overall

Select Country

China

China Overall Performance

	Year	Gold	Silver	Bronze	total
0	1932	0	0	0	0
1	1936	0	0	0	0
2	1948	0	0	0	0
3	1952	0	0	0	0
4	1956	0	0	0	0
5	1960	0	0	0	0
6	1964	0	0	0	0
7	1968	0	0	0	0
8	1972	0	0	0	0
9	1976	0	0	0	0
10	1984	15	8	8	32
11	1988	5	11	11	28
12	1992	16	22	22	53
13	1996	17	22	22	51
14	2000	28	16	16	58
15	2004	32	18	18	64
16	2008	51	21	21	100
17	2012	38	27	27	89
18	2016	26	18	18	70

OVERALL ANALYSIS

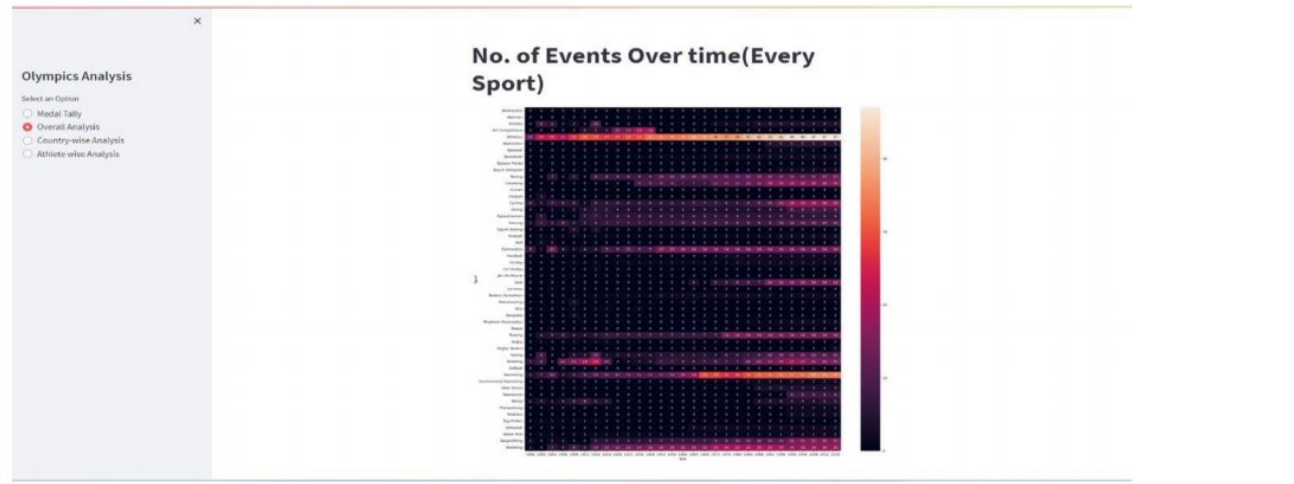
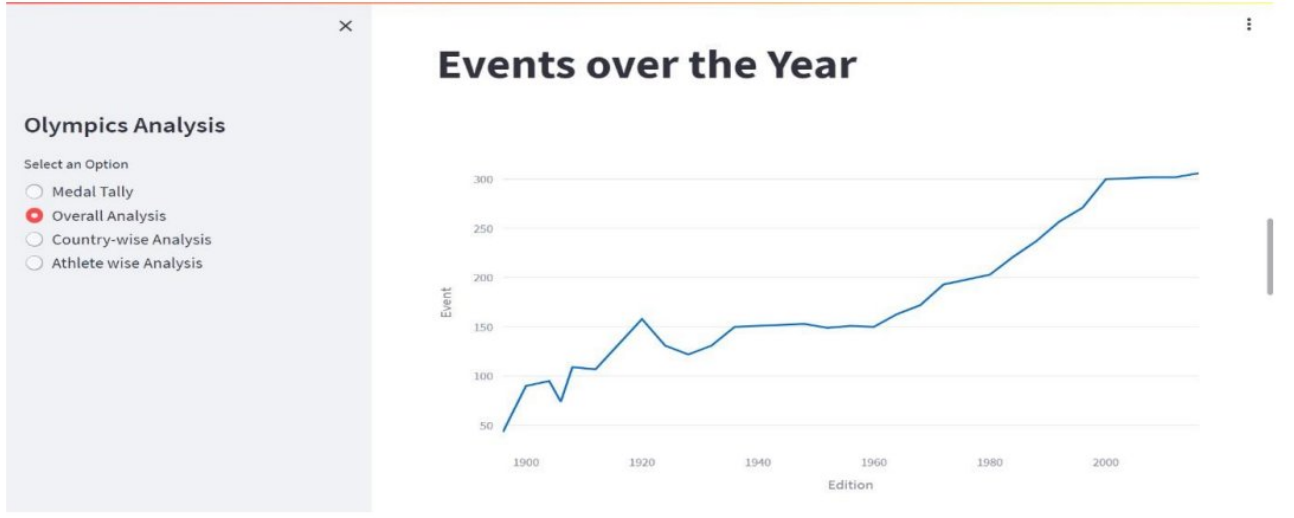
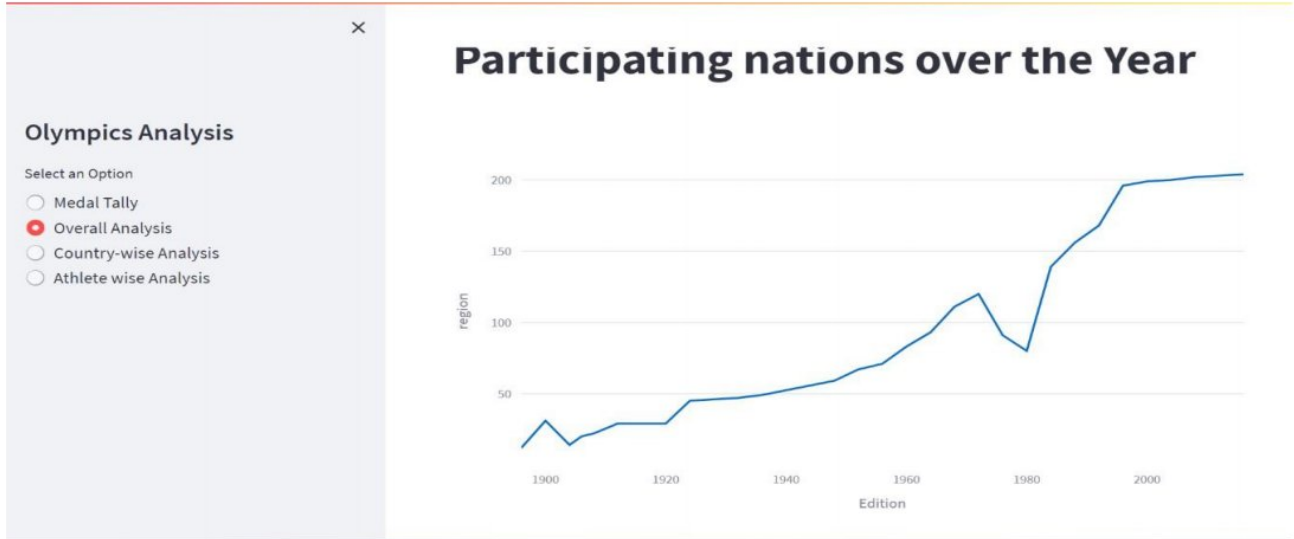
Olympics Analysis

Select an Option

- Medal Tally
- Overall Analysis
- Country-wise Analysis
- Athlete wise Analysis

Top Statistics

Editions	Host	Sports
28	23	52
Events	Nations	Athletes
651	206	116122



Olympics Analysis

Select an Option

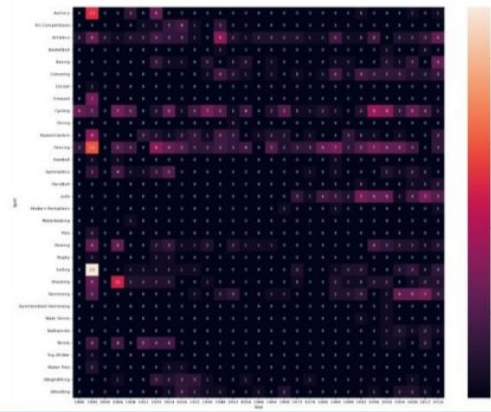
- Medal Tally
- Overall Analysis
- Country-wise Analysis
- Athlete wise Analysis

Country-wise Analysis

Select a Country

France

France excels in the following sports



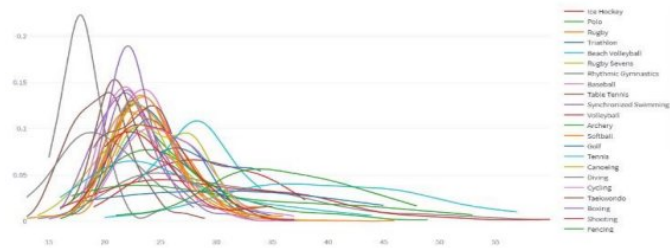
ATHELETE-WISE ANALYSIS

Olympics Analysis

Select an Option

- Medal Tally
- Overall Analysis
- Country-wise Analysis
- Athlete wise Analysis

Distribution of Age wrt Sports (Gold Medalist)



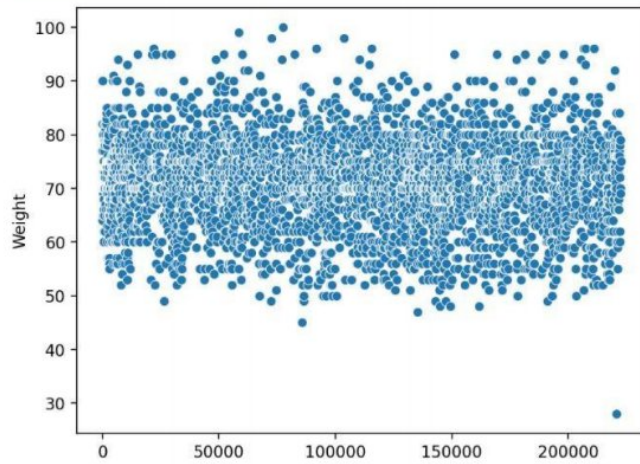
Olympics Analysis

Select an Option

- Medal Tally
- Overall Analysis
- Country-wise Analysis
- Athlete wise Analysis

Select a Sport

Football

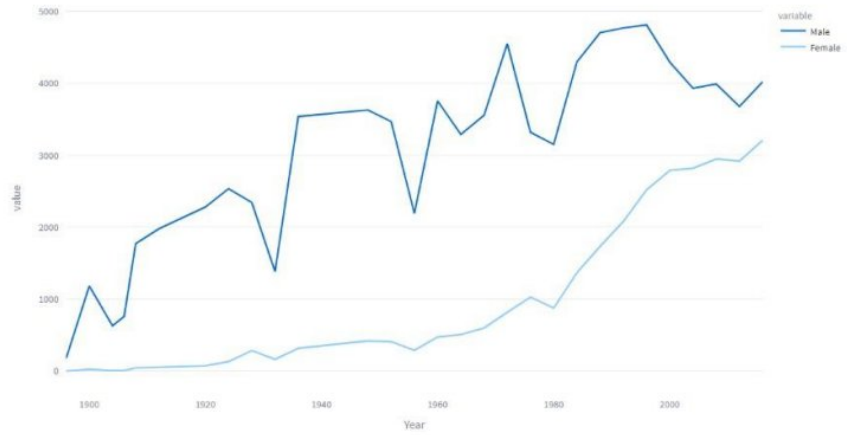


Olympics Analysis

Select an Option

- Medal Tally
- Overall Analysis
- Country-wise Analysis
- Athlete wise Analysis

Men Vs Women Participation over the Year



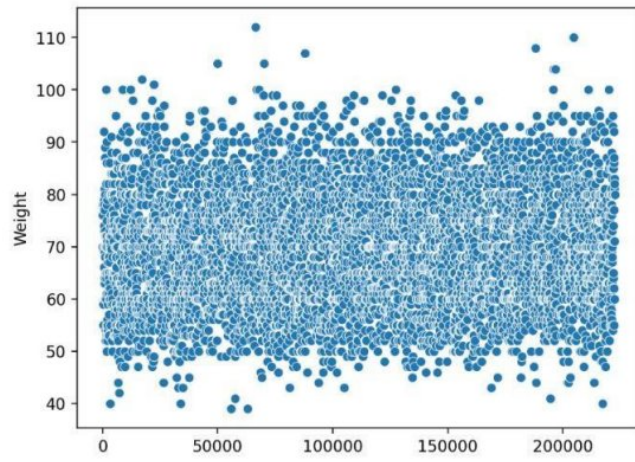
Olympics Analysis

Select an Option

- Medal Tally
- Overall Analysis
- Country-wise Analysis
- Athlete wise Analysis

Select a Sport

Swimming



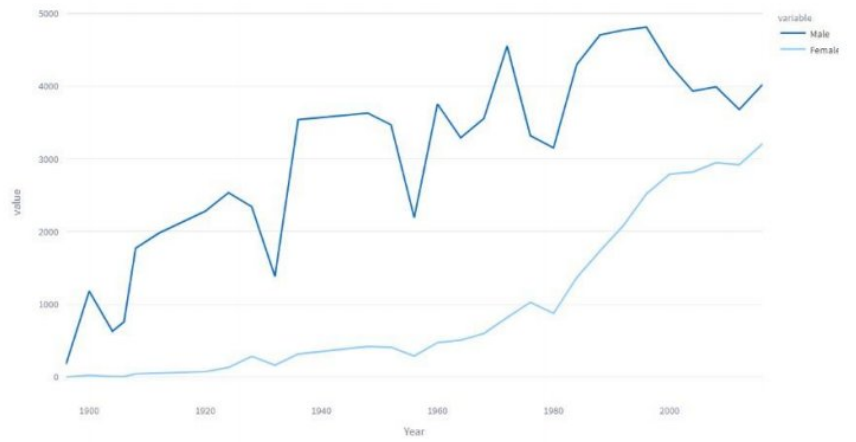
Olympics Analysis

Select an Option

- Medal Tally
- Overall Analysis
- Country-wise Analysis
- Athlete wise Analysis

Start

Men Vs Women Participation over the Year



CHAPTER 4

RESULT AND CONCLUSION

4.1 Result

Data Analysis of the Olympic dataset has lightened us on the valuable insights about the trends and patterns that have occurred throughout the historical Olympic Games. Upon thoroughly examining and Statistical Analysis, the key modules or points include the distribution of medals across different countries, the evolution of participation in the Olympic Sports over the years, and yes the impact of various factors on athletic performance of athletes.

4.2 Conclusion

In conclusion, this minor project on the data analysis of the Olympic dataset has provided a effective understanding of the dynamics carried throughout the Olympic Games. Several trends have been identified and the trends identified are very helpful and crucial for stakeholders, policymakers, and yes specially sports enthusiasts as it helps them in making informed decisions for forthcoming Olympics. Moving Further, the project highlights the significance of Use data analysis to extract important information from large data sets, contributing to the growing field of sports analytics. Furthermore , the future research could dig deeper into specific aspects revealed in this analysis, opening avenues for more targeted investigations into the world of Olympic sports.

CHAPTER 5

Future Scope

5. Future Scope

- The Minor Project on the "Data Analysis of Olympic Dataset" lays down the foundation for several potential avenues of future explorations:
- Predictive Modeling: Implement predictive models to forecast medal outcomes based on historical data, incorporating variables such as economic indicators, host country influence, and advancements in sports science.
- Athlete Performance Analysis: Conduct a detailed analysis of individual athlete performance trends, considering factors like age, training methodologies, and the impact of changing sports technologies.
- Country-Specific Analysis: Explore in-depth analyses for a specific country, examining its performance over a course of time, investments made in sports infrastructure, and yes, also the socio-economic factors influencing their success in the Olympic Games.
- Impact of Host City on Participation: Investigating the influence of the host city on athlete's participation and the athletic performance, taking into account geographical, cultural, and logistical factors.
- Incorporate Qualitative Data: Integrate qualitative data, such as athlete interviews, coach feedback, and crowd sentiments, to provide a more holistic understanding of the Olympic experience.
- Longitudinal Study: Extend the analysis over a more extended period to identify long-term trends and patterns, allowing for a more comprehensive understanding of the game.

REFERENCES

1. Analyzing Evolution of the Olympics by Exploratory Data Analysis using R Rahul Pradhan1, Kartik Agrawal and Anubhav Nag1
2. Leonardo De Marchi, "Data mining of Sports performance data".
3. <https://www.kaggle.com/search?q=+dataset+o>

