

PROFESSIONAL TRAINING REPORT

at

Sathyabama Institute of Science and Technology

(Deemed to be University)

Submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Information Technology.

By

S.SAIHANEESH

(Reg.no. 39120098)



DEPARTMENT OF INFORMATION TECHNOLOGY

SCHOOL OF COMPUTING

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY

JEPPIAAR NAGAR, RAJIV GANDHI SALAI,

CHENNAI – 600119, TAMILNADU

NOVEMBER - 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC

(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI– 600119

www.sathyabamauniversity.ac.in



DEPARTEMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of S.SAIHANEESH (39120098) who carried out the project entitled “**ODI MATCHES VISUALIZATION USING PYTHON**” under my supervision from August 2021 to October 2021.

INTERNAL GUIDE:

Name: Dr.Mary Gladence.L

Submitted for Viva Voice Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I S.SAIHANEESH hereby declare that the Project Report entitled **ODI MATCHES VISUALIZATION USING PYTHON** done by me under the guidance of **Dr.Mary Gladence.L** at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Information Technology.

DATE:

PLACE: CHENNAI

SIGNATURE OF THECANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.Mary Gladence.L** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

CERTIFICATE OF COMPLETION



THIS IS TO ACKNOWLEDGE THAT

SANGARAJU SAIHANEESH

Is Certified in

PYTHON USING DATA SCIENCE (B Code: 210923-PDS-1)

**And has successfully completed all requirements and criteria for said certification through
examination administered by irms**

This Certification Earned on

October 2021


Controller


Authorised Signature

ABSTRACT

In this blog post, I will make a short analysis of India vs. Australia One Day matches of past 35 years. The dataset has been taken from Kaggle. My python code for the following analysis and data visualizations

Dataset contain match records of 27 teams that have ever played One Day International cricket which include 24 countries and 3 tournament teams: - ICC World XI, Asia XI and Africa XI. I have used the data of India vs. Pakistan ODI matches for the analysis.

TABLE OF CONTENTS:

Chapter No.	Title.	Page No.
	Abstract	
	List of Figures	
	List of Abbreviations	
1.	INTRODUCTION	
	1.1. INTRODUCTION TO THE DATA VISULIZATION	1
	1.2. NEED FOR THE DATA VISULIZATION	3
	1.3. INTRODUCTION TO PYTHON	3
2.	AIM AND SCOPE OF THE PRESENT INVESTIGATION	6
3.	EXPERIMENTS OR REQUIRMENTS AND METHODS USED	7
	3.1. PROJECT DESCRIPTION	7
	3.2 METHODS USED	8
4.	RESULTS AND DISCUSION, PERFORMANCE ANALYSIS	9
5	SUMMARY AND CONCLUSION	10
	APPENDIX	11
	REFERENCES	23
	A. SOURCE CODE	
	B. SCREEN SHOTS	

LIST OF FIGURES

FIGURE NO.	FIGURE NAME.	PAGE NO.
5.1	GROUNDS	13
5.2	WON AND LOST PERCENTAGE	14
5.3	COUNT AND VENUE 1	15
5.4	COUNT AND VENUE 2	15
5.5	INDIA WON AND LOST PERCENTAGE	16
5.6	RUN RATE FIRST INNINGS	17
5.7	YEARWISE RESULTS FOR INDIA	18
5.8	MATCHES PLAYED BY YEAR	19
5.9	WIN ON RUN RATE TEAM WISE	21
5.10	NUMBER OF MATCHES PLAYED IN THIS GROUND	22

CHAPTER 1

1.1INTRODUCTION

1.1. Introduction to the DATA VISUALIZATION:

Data visualization is the discipline of trying to understand data by placing it in a visual Context so that pattern, trends and correlations that might not otherwise be detected can be exposed.

Python offers multiple great graphing libraries that come packed with lots of different features. No matter if you want to create interactive, live or highly customized plots python has an excellent library for you.

To get a little overview here are a few popular plotting libraries:

- **MATPLOTLIB:** It is a multi-platform data visualization library built on NumPy arrays, and designed to work with the broader SciPy stack. It was conceived by John Hunter in 2002, originally as a patch to I Python for enabling interactive MATLAB-style plotting via gnu plot from the I Python command line.
- **PANDAS VISULIZATION:** Pandas is an open-source python library that is used for data manipulation and analysis. ... The two primary data structures are Series which is 1 dimensional and Data Frame which is 2 dimensional. It is one of the most important and useful tools in the arsenal of a Data Scientist and a Data Analyst.
- **SEABORN:** Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.
- **PLOTLY:** Plotly is the Python Library for interactive data visualizations. Plotly allows you to plot superior interactive graphs than either Matplotlib or Seaborn.

The program is designed in such a way that the user can analyze .csv files for analysis. Using Data visualization technique we are going to visualize ODI Cricket matches. For example, we are going to compare and visualize INDIA vs. PAKISTAN matches and performances since past 35 years(1985 – 2020).

1.2 Need for DATA VISULIZATION:

Data visualization is the practice of translating information into a visual context, such as a map or graph, to make data easier for the human brain to understand and pull insights from. The main goal of data visualization is to make it easier to identify patterns, trends and outliers in large data sets.

Data visualization is one of the steps of the data science process, which states that after data has been collected, processed and modeled, it must be visualized for conclusions to be made. Data visualization is also an element of the broader data presentation architecture (DPA) discipline, which aims to identify, locate, manipulate, format and deliver data in the most efficient way possible.

1.3 INTRODUCTION TO PYTHON

Python is a high level, dynamic programming language. Python3.4 version was used as it is a mature, versatile and robust programming language. It is an interpreted language which makes the testing and debugging extremely quickly as there is no compilation step. There are extensive open-source libraries available for this version of python and a large community of users. Python is simple yet powerful, interpreted and dynamic programming language, which is well known for its functionality of processing natural language data, i.e., spoken English using NLTK. Other high level programming languages such as –R and –Matlab were considered because they have many benefits such as ease of use but they do not offer the same flexibility and freedom that Python can deliver.

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. The name Pandas is derived from the word Panel Data – an Econometrics from Multidimensional data. In 2008, developer Wes McKinney started developing pandas when in need of high performance, flexible tool for analysis of data. Prior to Pandas, Python was majorly used for data munging and preparation. It had very less contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data—load, prepare, manipulate, model

and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Key Features of Pandas:

- Fast and efficient Data Frame object with default and customized indexing.
- Tools for loading data into in-memory data objects from different file formats.
- Data alignment and integrated handling of missing data.
- Reshaping and pivoting of date sets.
- Label-based slicing, indexing and subsetting of large data sets.
- Columns from a data structure can be deleted or inserted.
- Group by data for aggregation and transformations.
- High performance merging and joining of data.
- Time Series functionality.

Matplotlib is the most popular python plotting library. It is a low-level library with a Matlab like interface which offers lots of freedom at the cost of having to write more code.

Key features of Matplotlib:

- Semantic way to generate complex, subplot grids.
- Settings the aspect ratio of the axes box.
- Colored labels in legends.
- Ticks and labels.
- RcParams can be passed as Decorators.
- 3D plots now support minor ticks.

Seaborn is a Python data visualization library based on Matplotlib. It provides a high-level interface for creating attractive graphs.

Seaborn has a lot to offer. You can create graphs in one line that would take you multiple tens of lines in Matplotlib. Its standard designs are awesome and it also has a nice interface for working with pandas data frames.

Key features in Seaborn:

- Built in themes for styling matplotlib graphics.
- Visualizing univariate and bivariate data.
- Fitting in and visualizing linear regression models.
- Plotting statistical time series data.
- Seaborn works well with NumPy and Pandas data structures.

Plotly allows users to import, copy and paste, or stream data to be analyzed and visualized. For analysis and styling graphs, Plotly offers a Python sandbox (NumPy supported), datagrid, and GUI. Python scripts can be saved, shared, and collaboratively edited in Plotly.

Key features of Plotly:

- Charts
- Dashboards
- File Export
- App Manager
- Kubernetes Authentication
- Jobs Queue
- Snapshot Engine

CHAPTER 2

AIM AND SCOPE OF THE PRESENT INVESTIGATION

AIM: To visualize the data using .csv or .tsv files and by using graphs to support our analysis from a data set.

SCOPE: Our brain too interprets information fast and best when received through graphical processing. It makes complicated datasets easily understandable and comprehensible through the medium of visual communication. The strength and scope of data visualization is immense. To get full advantage of data visualization, first you should know about the various modern approaches on scope of data visualization in the related fields.

In this project we are going to use a dataset named ODI.csv file which contains the match data of all the countries which played the one day international match from 1985 to 2020(35 years). We are going to compare and contrast matches between two teams INDIA and PAKISTAN. We will analyze the match wins, lost and performance between these two teams in the form of Data visualization. And we analyzes based upon runrates.

CHAPTER 3

EXPERIMENTAL OR REQUIREMENTS AND METHODS

3. SYSTEM SPECIFICATION

Hardware Requirements:

1. Processor – Intel Core processors or any AMD chips.
2. RAM – 1 GB
3. Hard Disk – 40GB
4. Mouse – Standard Mouse
5. Keyboard – Logitech Keyboard
6. Processor Speed – 2.4GHZ

Display Mode:

1. Color Quality – Highest[32bit]
2. Screen Resolution – 1024 by 768Pixels

Software requirements:

1. Jupyter notebook (anaconda 3) or Google collab.
2. Python 3 or latest.

3.1 PROJECT DESCRIPTION:

We are going to analyze the data of One Day International(ODI) cricket matches. The dataset we are using contains the information about when the match took place, where the match took place which team won the match, which innings the team batted, etc. We have got this dataset from Kaggle. We are going to analyze the information in this dataset and try to present in a visual manner using different libraries in python programming language like numpy, pandas, matplotlib and seaborn.

3.2 METHODS USED:

1. Import Matplotlib
2. Plotly
3. Pandas
4. Numpy
5. Seaborn
6. Using techniques like data cleaning.

CHAPTER 4

RESULTS, PERFORMANCE ANALYSIS

RESULT: In this project we have successfully compared and contrasted between two different teams, their performance, winning probability based upon their run rate and team performance using the data visualization techniques.

We have plotted the graphs in different formats like bar, pie, scatter plots.

PERFORMANCE ANALYSIS:

Performance analysis is the technique of studying or comparing the performance of a specific situation in contrast to the aim and yet executed. In Human Resource, performance analysis can help to review an employee's contribution towards a project or assignment, which they allotted him or her.

We distinguish three basic steps in the performance analysis process: data collection, data transformation, and data visualization. Data collection is the process by which we get data about program performance from an executing program.

I had used countplot technique to count the values from data and plot the graph, and used stripplot techniques to plot the graph, and stripplot techniques to plot the graph in scatter graph.

The xticks is function used to do rotation techniques which will mention the data to rotate the data to rotate the date ticks on the x axis by a specifies amount of degrees.

Save fig is also a method to save the plotted graphs to our files.

Mean() function can be used to calculate mean/average of a given list of number. It returns mean of the data set passed as parameter.

Figsizes can be used to present the graph in a particular size we need

CHAPTER 5

SUMMARY AND CONCLUSION

To summarize the project DATA VISUALIZATION USING PYTHON, we have used all type of python techniques and imported different type of libraries from python and used the jupyter platform to implement the libraries and packages. I have selected the data set from Kaggle which contains data related ODI cricket matches and we have detailed the comparison between the two different teams INDIA and PAKISTAN and their performances in the whole league. We had even visualized and compared all the teams winning and loosing rates. These all visualizations are done using python and python imported packages We have used matplotlib, seaborn, plotly, numpy, pandas and many other types of visualization techniques using **Jupyter Notebook** (Anaconda 3) Software.

In this project we can conclude that all the different type of visualizations are shown through Python techniques. We have illustrated the .csv file which contains the data of ODI cricket matches that are played from past 35years in different forms of graphs and visualizations. We have used different type of packages that are imported from Python and implemented them to visualize the data set. We can use any type of data set and even .tsv files to data visualize the information in the formats we like using Python.

“Good data visualization should communicate a data set clearly and effectively by using graphics”.

“The best visualizations make it easy to comprehend data at a glance”.

APPENDIX

CODING:

CODE:

```
!pip install plotly
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.graph_objects as go
from plotly.subplots import make_subplots
import pandas as pd
df = pd.read_csv('ODIMatchResults_Inn.csv')
df
df.head(50)
df. Ground
df.team1
type(df)
df.Ground.value_counts()
df.Ground.value_counts().plot(kind='bar',figsize=[30,30])
df[df.runs >= 400]
len(df[df.runs >= 400])
df[df.runs >= 300].result.value_counts()
df.Ground.head(20).value_counts().plot(kind='bar',figsize=[30,30])
df['Year'] = df['Date'].apply(lambda date : date.split(" ")[2])
df['Month'] = df['Date'].apply(lambda date : date.split(" ")[1])
df.drop('Date',axis=1,inplace=True)
df.head()
```

```

ind_pak = df[((df['team1'] == 'Pakistan') & (df['team2'] == 'India')) | ((df['team2'] ==
'Pakistan') & (df['team1'] == 'India'))]
ind_pak
ind_df = ind_pak[ind_pak['team1'] == 'India']
pak_df = ind_pak[ind_pak['team1'] == 'Pakistan']
ind_pak_inn1 = ind_pak[ind_pak['Inn'] == 1]
ind_pak_chase = ind_pak[ind_pak['Inn'] == 2]
ind_df_1st = ind_df[ind_df['Inn'] == 1]
ind_df_2nd = ind_df[ind_df['Inn'] == 2]
pak_df_1st = pak_df[pak_df['Inn'] == 1]
pak_df_2nd = pak_df[pak_df['Inn'] == 2]
plt.figure(figsize=(25,10))
plt.xticks(rotation = 30)
sns.countplot('Ground', hue = 'result', data = ind_pak_inn1, palette = 'Set1' )
plt.xlabel('Venue', fontsize = 30);
plt.ylabel('Count', fontsize = 30);
plt.figure(figsize=(25,10))
plt.xticks(rotation = 30)
sns.countplot('Ground', hue = 'result', data = ind_pak_chase, palette = 'Set2' )
plt.xlabel('Venue', fontsize = 30);
plt.ylabel('Count', fontsize = 30);
fig = px.pie(ind_df_1st, values = 'Inn', names = 'result')
fig.show()
fig = px.pie(ind_df_2nd, values = 'Inn', names = 'result')
fig.show()
plt.figure(figsize=(8,6))
sns.stripplot(x = 'team1', y = 'rpo', data = ind_pak_inn1);

```

```

plt.xlabel('Team',fontsize = 15)
plt.ylabel('Run Rate bating 1st',fontsize = 15) ;
plt.figure(figsize=(8,6))
sns.stripplot(x = 'team1',y = 'rpo',data = ind_pak_chase);
plt.xlabel('Team',fontsize = 15)
plt.ylabel('Run Rate batting 2nd',fontsize = 15) ;
ind_pak['Year'] = ind_pak['Year'].astype('int')
last_10_year = ind_pak[ind_pak['Year']>=1990]
match_1_2 = last_10_year[last_10_year['Ground']=='Dubai (DSC)']
match_1_2
match_1_2[(match_1_2['result']=='won')&(match_1_2['Inn']==1)]['runs'].mean()
match_1_2[(match_1_2['result']=='won')&(match_1_2['Inn']==2)]['runs'].mean()
plt.figure(figsize=(25,10))
plt.xticks(rotation = 30)
sns.countplot('Year',hue = 'result',data = ind_df,palette = 'rainbow' )
plt.xlabel('Yearwise results for India',fontsize = 30)
plt.ylabel('Count',fontsize = 30);
df['Ground'].value_counts().plot(kind='pie', title='Number of matches played in this
ground', figsize=(10, 10));
df['Year'].value_counts().sort_index().plot(kind='bar', title='Matches played by
year', figsize=(25, 25))
plt.xticks(rotation=0);
df['team1'].unique()
len(df[df.rpo >=6.0])
df[df.rpo >=6.0].result.value_counts()
df[df.rpo >=6.0].result.value_counts().plot(kind='pie',figsize=[8,8], autopct='%0.2f')
df[df.rpo >=6.0].team1.value_counts()

```

```
df[df.rpo >=6.0].team1.value_counts().plot(kind='pie',figsize=[10,10],  
autopct='%0.2f')  
df[df.rpo >=6.0].Ground.value_counts().plot(kind='pie',figsize=[20,20],  
autopct='%0.2f')
```

OUTPUTS AND GRAPHS:

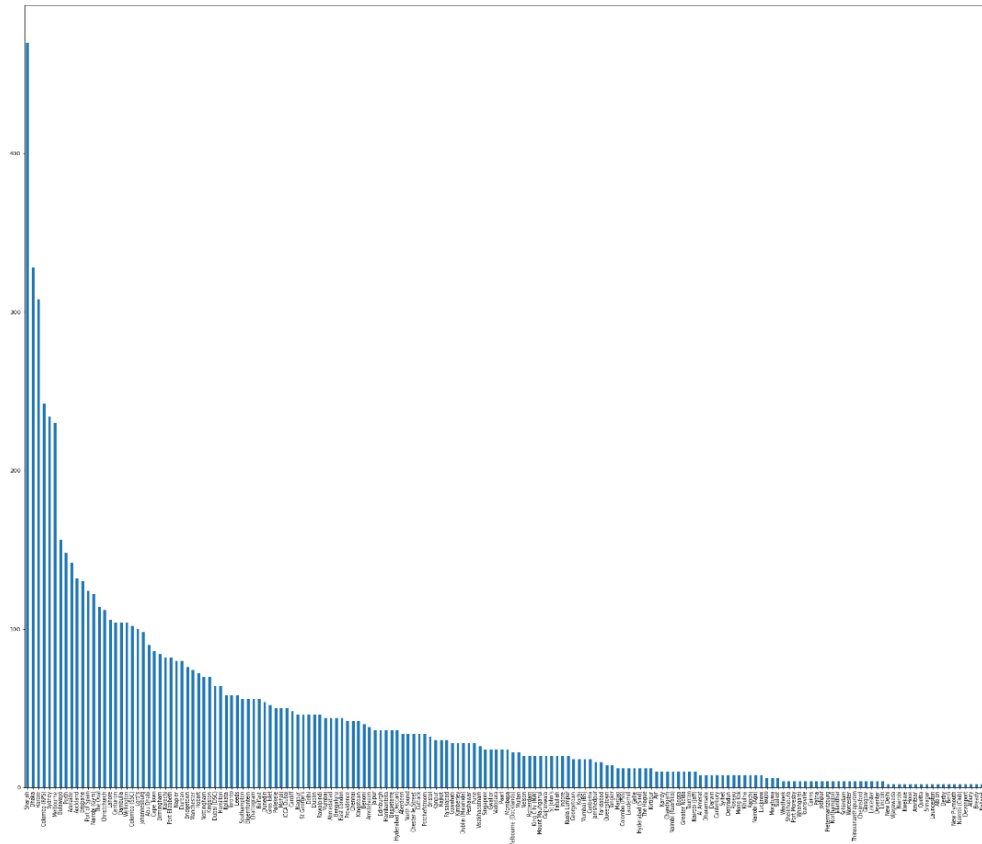


Fig 5.1 GROUNDS

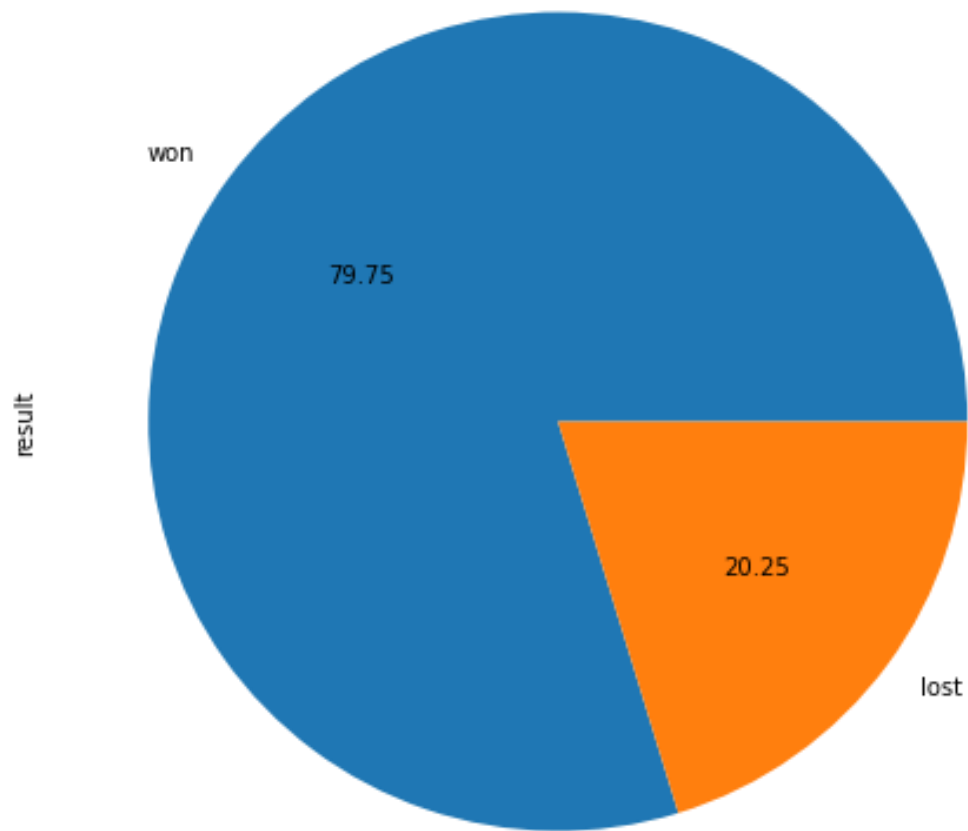


Fig.5.2 WON AND LOST PERCENTAGE

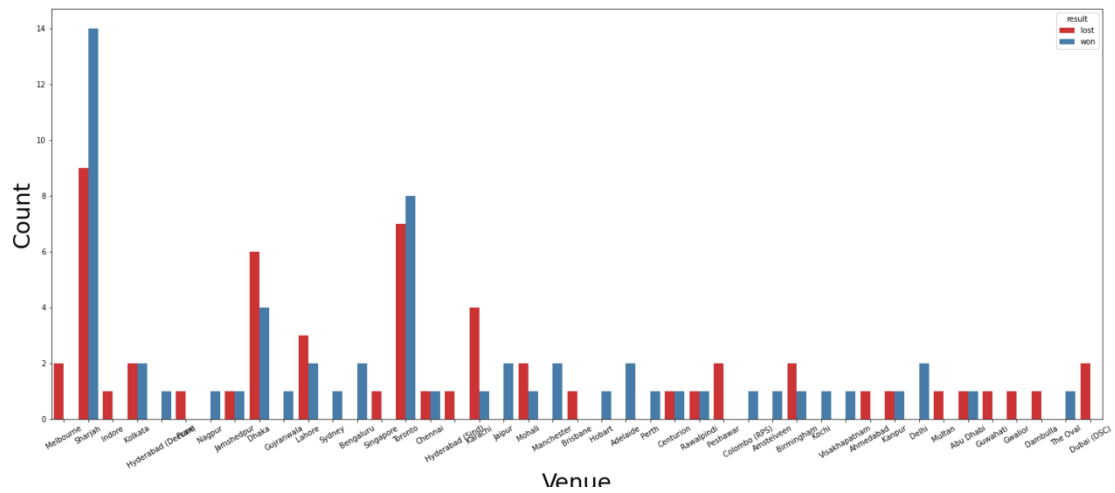


FIG 5.3 COUNT AND VENUES 1

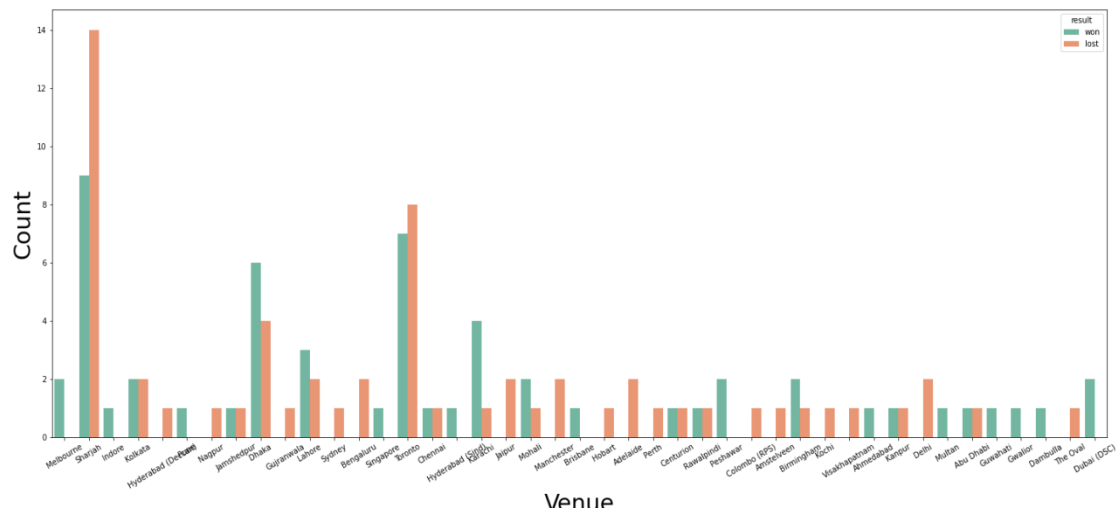


FIG 5.4 COUNT AND VENUE 2

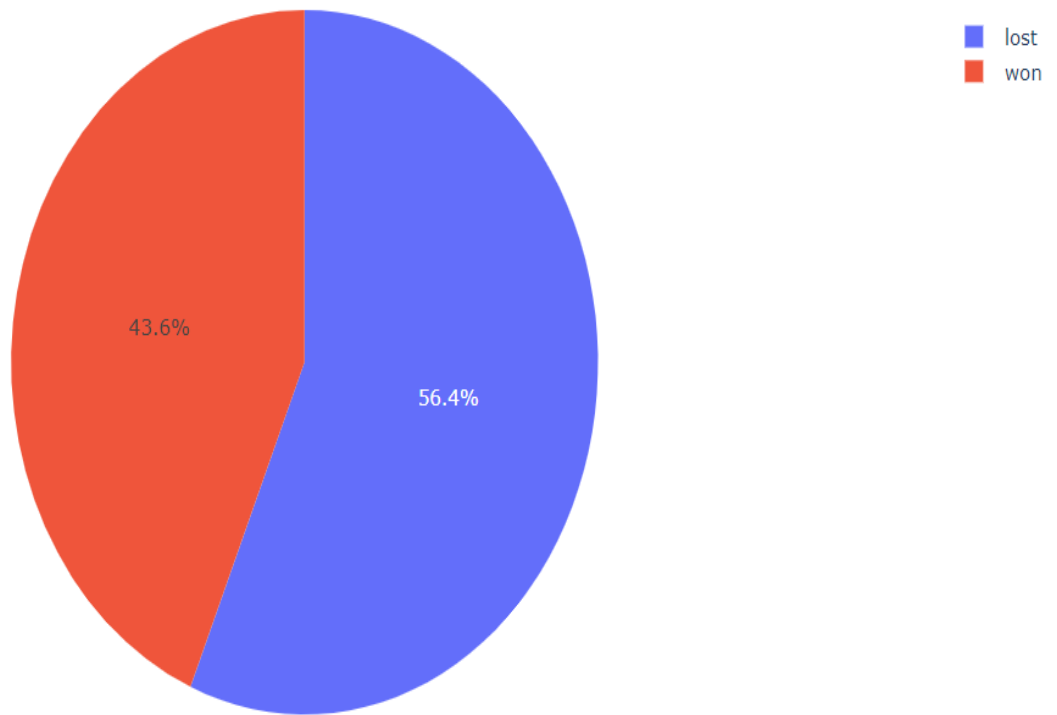


FIG 5.5 INDIA WON AND LOST (PERCENTAGE)

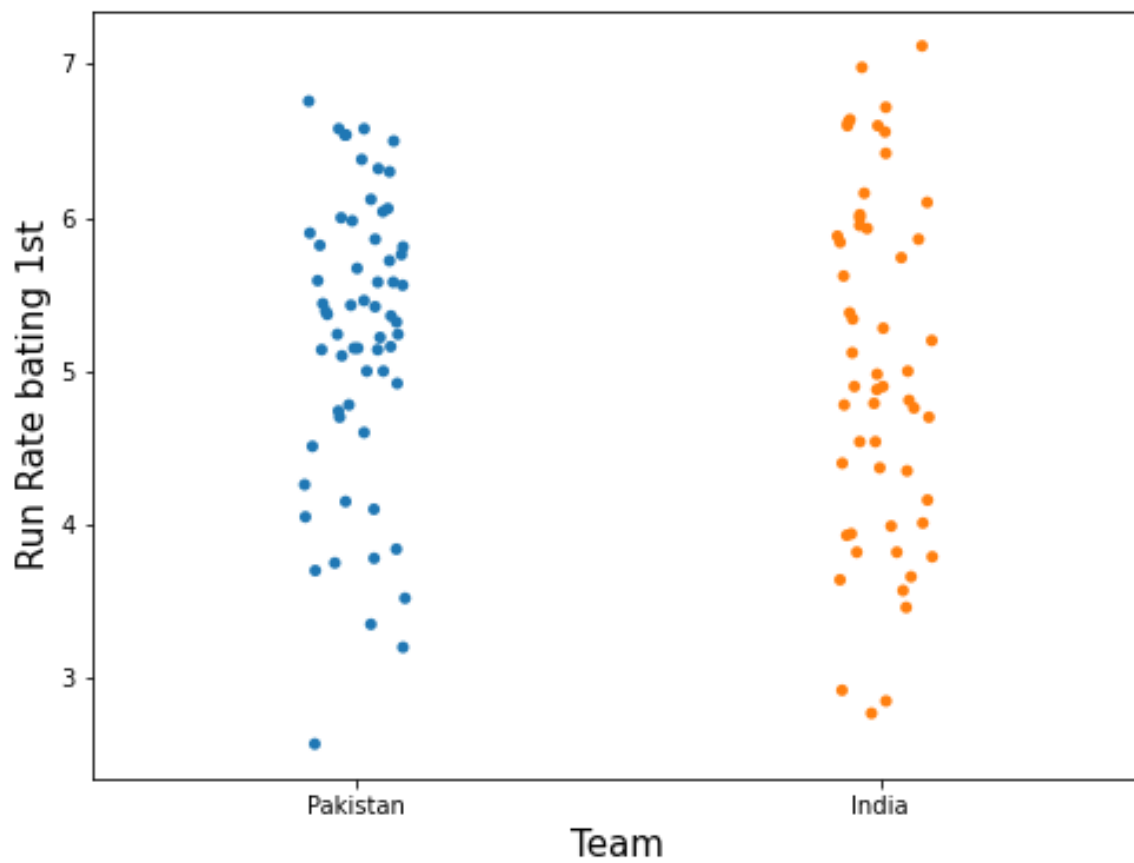


FIG 5.6: RUN RATE IN FIRST INNINGS

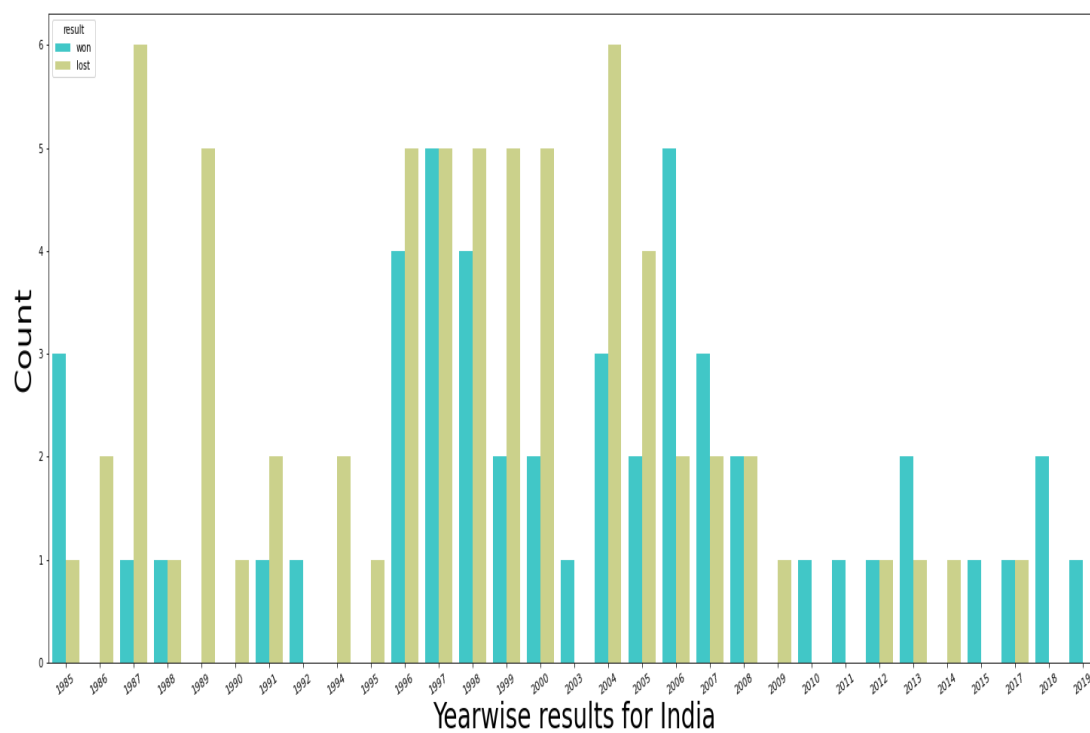


FIG 5.7 YEARWISE RESULTS FOR INDIA

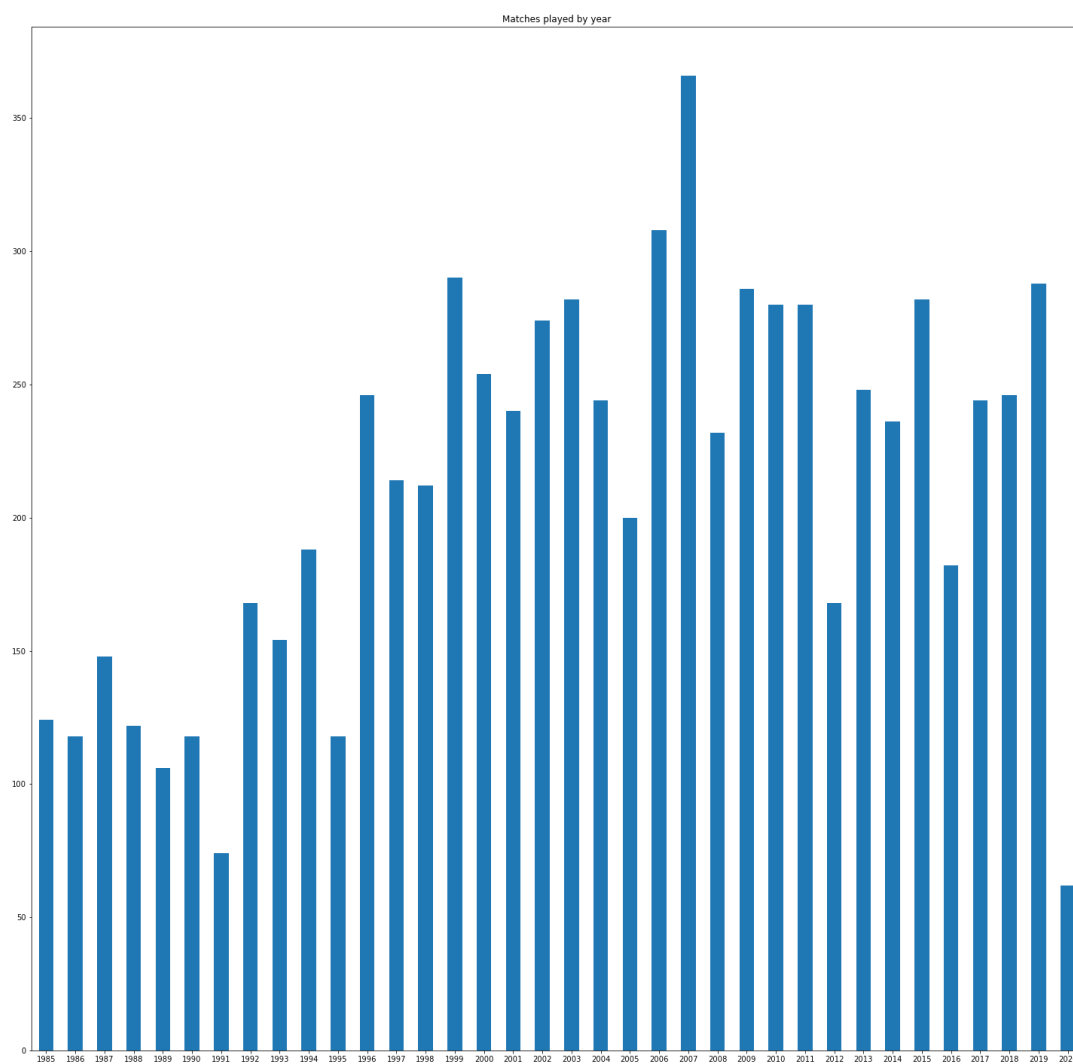


FIG 5.8 MATCHED PLAYED BY YEAR

REFERENCES

[1] Kaggle website for obtaining the datasets (.csv) files.

[2] Jupyter software.

[3] Python 3 for importing packages.