DATA ANALYSIS AND VISUALIZATION REPORT

DATASET: RAINFALL IN PAKISTAN

GROUP MEMBERS:

SYEDA ALIZA

RUBA NIZAM

NARMEEN

DEPARTMENT OF MATHEMATICS

(BS FINANCIAL MATHEMATICS)

(SECOND YEAR 4TH SEMESTER)

PROJECT SUBMITTED TO:

(MR SYED UMAID AHMED)

CONTENTS

INTRODUCTION TO DATA SCIENCES:

Data sciences is a field that uses scientific methods, algorithms and system to extract an insight and knowledge from the unstructured data and apply knowledge and actionable insights from data across a broad range of application domain. Data sciences encompasses preparing for data including cleaning, aggregating and manipulating the data. Data sciences is one of the most exciting fields out here today in modern world.

USES OF DATA SCIENCES IN DAILY LIFE:

Data sciences is one of the most common and famous field in today's world. It is a career field that stems from multiple disciplines following are some main and common applications of the data sciences given below

1) Banking

2) E-Commerce

3) Finance

4) Manufacturing

5) Transport

6) Healthcare

7) Predictive models for diagnosis

LIBRARIES USED IN PROJECT DATASET:

Following are the libraries that we have used in our data analysis project:

1) NumPy

2) Pandas

3) Matplotlib

4) Seaborn

1) NUMPY:

NumPy is a python library used for working with arrays. It also has a function of working in domain of linear algebra and matrices. It was created in 2005 by Travis Oliphant. It is an open-source project which we can use freely. NumPy stands for numerical python. NumPy facilitate advanced mathematical operations on a large scale of data.

2) MATPLOTLIB:

Matplotlib is a plotting library in python programming language and its numerical extension NumPy. A python Matplotlib script is

structured so that a few lines of code all are that is required in most instances to generate a visual data plot. Matplotlib is developed by Michael Droettboom in 2003.

3) PANDAS:

Pandas is a python library providing fast, flexible and expressive data structure designed to make working with relational or labelled data easily. Is is used to perform machine learning task efficiently. Pandas makes it simple to do many of the time-consuming tasks efficiently which includes data cleansing, data fill, data normalization, merges and joins, data visualization, loading and saving data, data inspection, statistical analysis and many more.

4) SEABORN:

Seaborn is a data visualization library built on the top of Matplotlib. Visualization is the central part of Seaborn which helps in exploration and understanding of a data. This library is used to make statistical graphs in python. Its plotting function operate on data frames and arrays containing whole dataset and perform the necessary mapping to produce informative plots.

INRTODUCTION TO DATASET USED IN A PROJECT:

The selected dataset contains rainfall data of Pakistan. The parameter considered for the evaluation of the performance and the efficiency of the given rainfall prediction model. Following are the steps which we are going to discuss in our project dataset

1) Data Cleaning

2) Data Analysis

3) Forecast and prediction using the dataset

4) Graph plotting of given dataset

INPUT NO:1

IMPORTING LIBRARIES:

First of all, we have to import the libraries as show below.

In [2]:

```
import matplotlib.pyplot as plt
import seaborn as sns
```

INPUT NO: 2

READ THE FILE:

Now we are reading the file on jupyter notebook on which we have to complete our data analysis.

In [3]:

```
df=pd.read_csv('../input/rainfall-in-pakistan/Rainfall_1901_2016_PAK.csv')
df.head()
```

INPUT NO; 3

CHECKING THE COLUMNS AND HEADS:

Now we will check the head and columns of the data using the following inputs.

In [4]:

```
df.columns
```

Out[4]:

```
Index(['Rainfall - (MM)', ' Year', 'Month'], dtype='object')
```

In [5]:

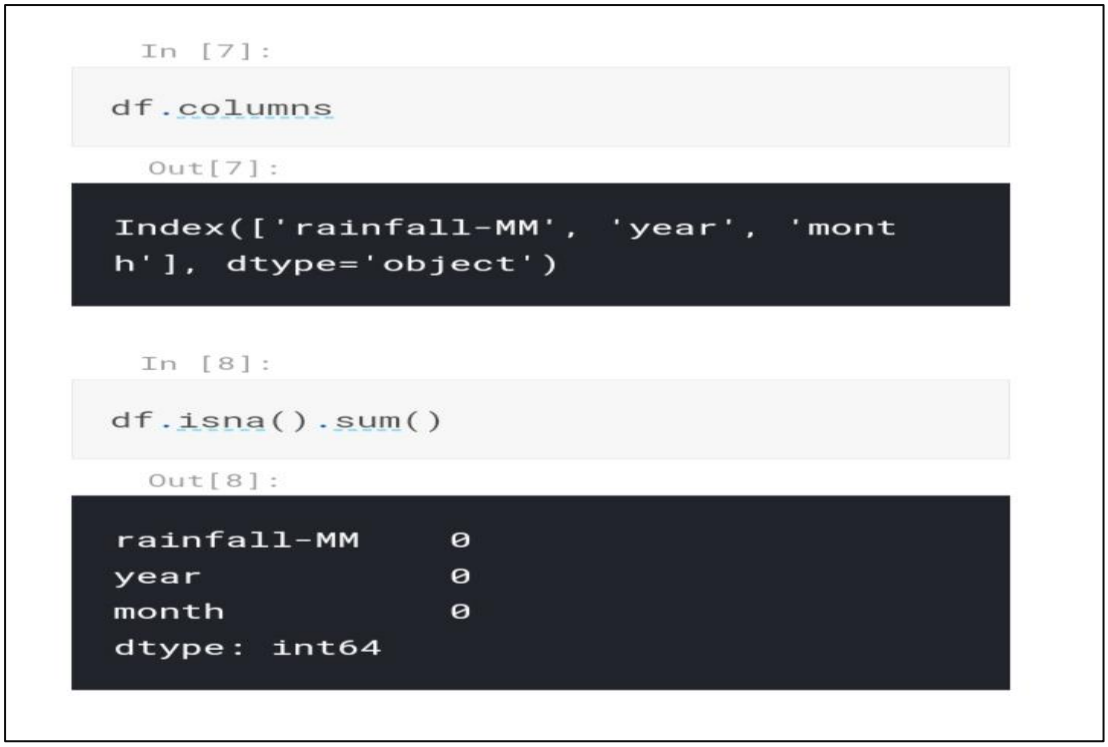
```
# Renaming columns
df.rename(columns={'Rainfall - (M
M)':'rainfall-MM',' Year':'year','Mont
h':'month'}, inplace=True)
```

In [6]:

```
df.head()
```

Out[6]:

	rainfall-MM	year	month
0	40.4258	1901	January
1	12.3022	1901	February
2	25.5119	1901	March
3	14.2942	1901	April
4	38.3046	1901	May



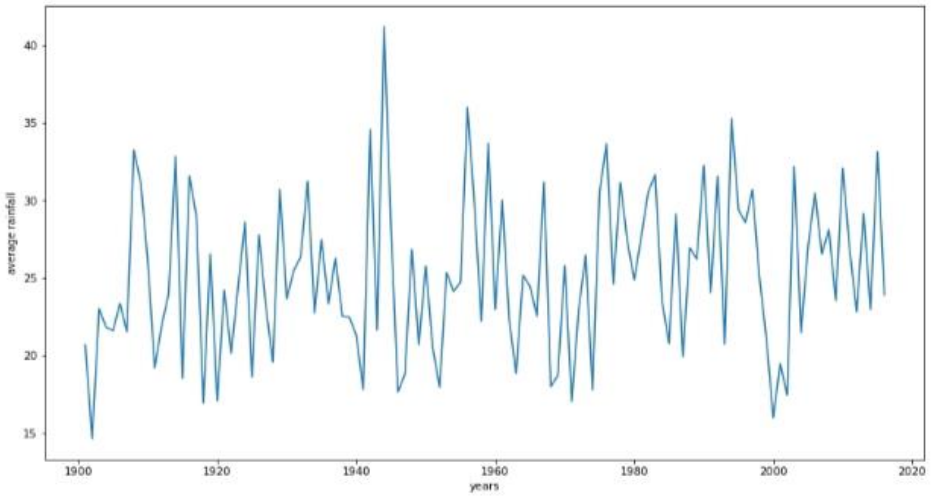
INPUT NO: 4:

PLOTTING OF GRAPH:

Now we will plot the graph of average rainfall throughout the data.

In [11]:

```
plt.figure(figsize=(14,8))
plt.plot(df.groupby(['year']).mean())
plt.xlabel('years')
plt.ylabel('average rainfall')
plt.show()
```



INPUT NO: 5

FINDING MEAN:

Now we are going to find the average of rainfall from our dataset.

In [12]:

```
df[df['year']==2016].mean()['rainfall-MM']
```

Out[12]:

```
23.913654999999995
```

INPUT NO: 6

FINDING MONTHLY AVERAGE:

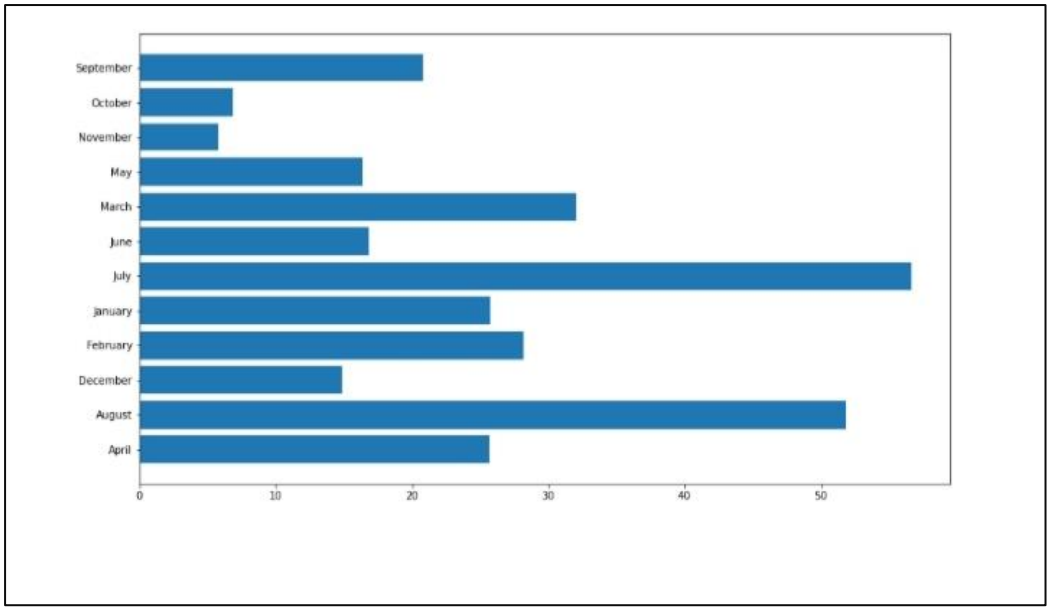
Now we are going to find monthly average of rainfall.

In [13]:

```
avg_rainfall_by_month=df.groupby(df.mo
nth)['rainfall-MM'].mean()
```

In [14]:

```
plt.figure(figsize=(14,8))
plt.barh(avg_rainfall_by_month.index,a
vg_rainfall_by_month)
plt.show()
```



INPUT NO: 7

MONTHLY AVERAGE:

In the given input we will find the monthly average of our rainfall dataset.

In [15]:

```
avg_rainfall_across_years=df.groupby(df.year)['rainfall-MM'].mean()
```

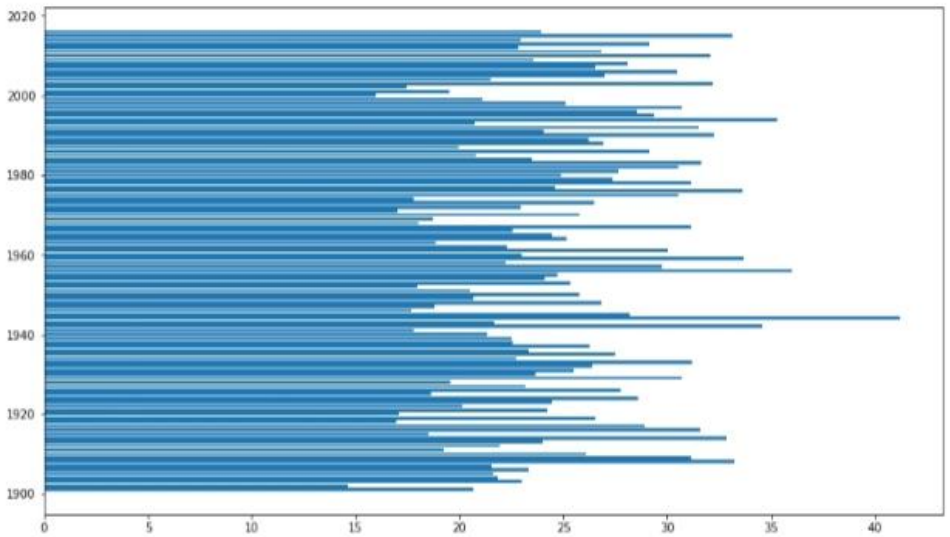
INPUT NO: 7

GRAPH PLOTTING:

For now, we will plot the graph of our monthly average of rainfall obtained above.

In [16]:

```
plt.figure(figsize=(14,8))
plt.barh(avg_rainfall_across_years.index,avg_rainfall_across_years)
plt.show()
```



INPUT NO: 8

DATA SORTING:

Now we are going to sort the data so that we can easily predict the top most rainfall year.

In [17]:

```
avg_rainfall_across_years.sort_values
(ascending=False)
```

Out[17]:

```
year
1944    41.197529
1956    35.999497
1994    35.272982
1942    34.558610
1959    33.684169
           ...
1920    17.075132
1971    17.028849
1918    16.945073
2000    15.983080
1902    14.635436
Name: rainfall-MM, Length: 116, dtype:
float64
```

INPUT NO: 9

FILTERING THE DATA:

In [18]:

```
#dataframe copy
df_after_2k=df.copy()
```

In [19]:

```
#Filtering Years
df_after_2k=df_after_2k[df_after_2k['year']>=2000]
df_after_2k.head()
```

Out[19]:

	rainfall-MM	year	month
1188	19.31100	2000	January
1189	14.23570	2000	February
1190	8.70368	2000	March
1191	3.96081	2000	April
1192	7.20656	2000	May

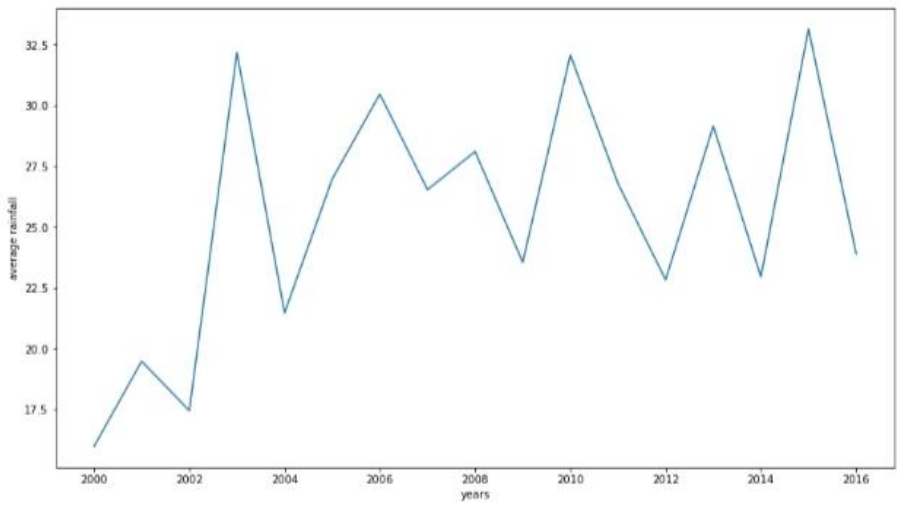
INPUT NO: 10

PLOTTING OF A GRAPH:

Now going to plot the graph of rainfall data on yearly basis.

In [20]:

```
#plotting the data by taking mean acros
s the year
plt.figure(figsize=(14,8))
plt.plot(df_after_2k.groupby(['yea
r']).mean())
plt.xlabel('years')
plt.ylabel('average rainfall')
plt.show()
```



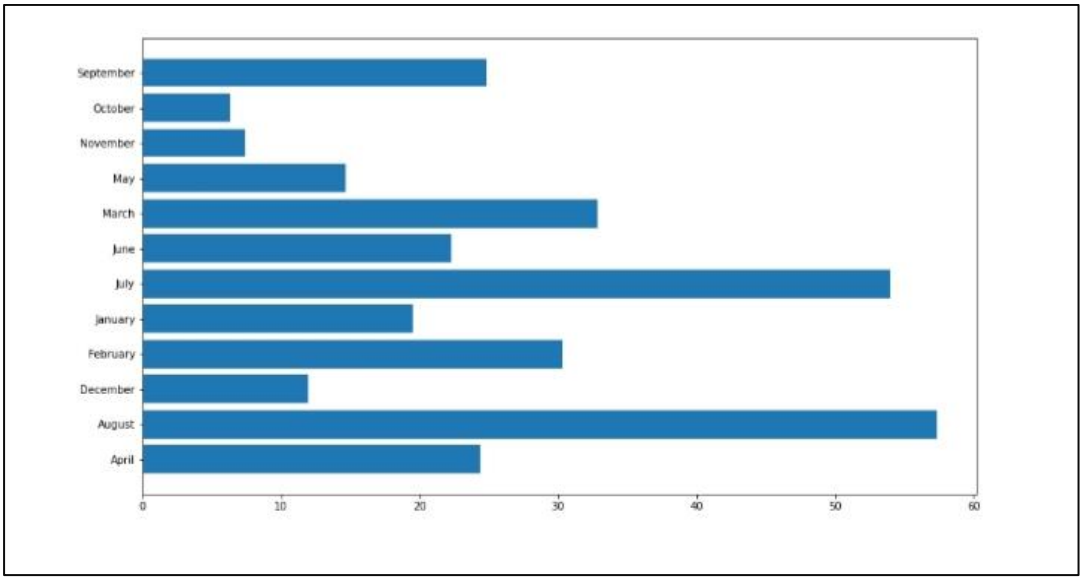
INPUT NO: 11

GRAPH PLOTTING:

Plotting the graph again of average rainfall occurred in a month after a year of 2020.

In [22]:

```
#Plotting average rainfall in a month a
fter 2020
plt.figure(figsize=(14,8))
plt.barh(avg_rainfall_by_month_after_2
k.index,avg_rainfall_by_month_after_2
k)
plt.show()
```



INPUT NO: 11

NUMERAL VALUE OF AVERAGE RAINFALL:

We will find the numeral values of average rainfall occurred from year 2000 to 2016.

In [23]:

```
avg_rainfall_across_years_after_2k=df_
after_2k.groupby(df_after_2k.year)['ra
infall-MM'].mean()
```

In [24]:

```
avg_rainfall_across_years_after_2k
```

Out[24]:

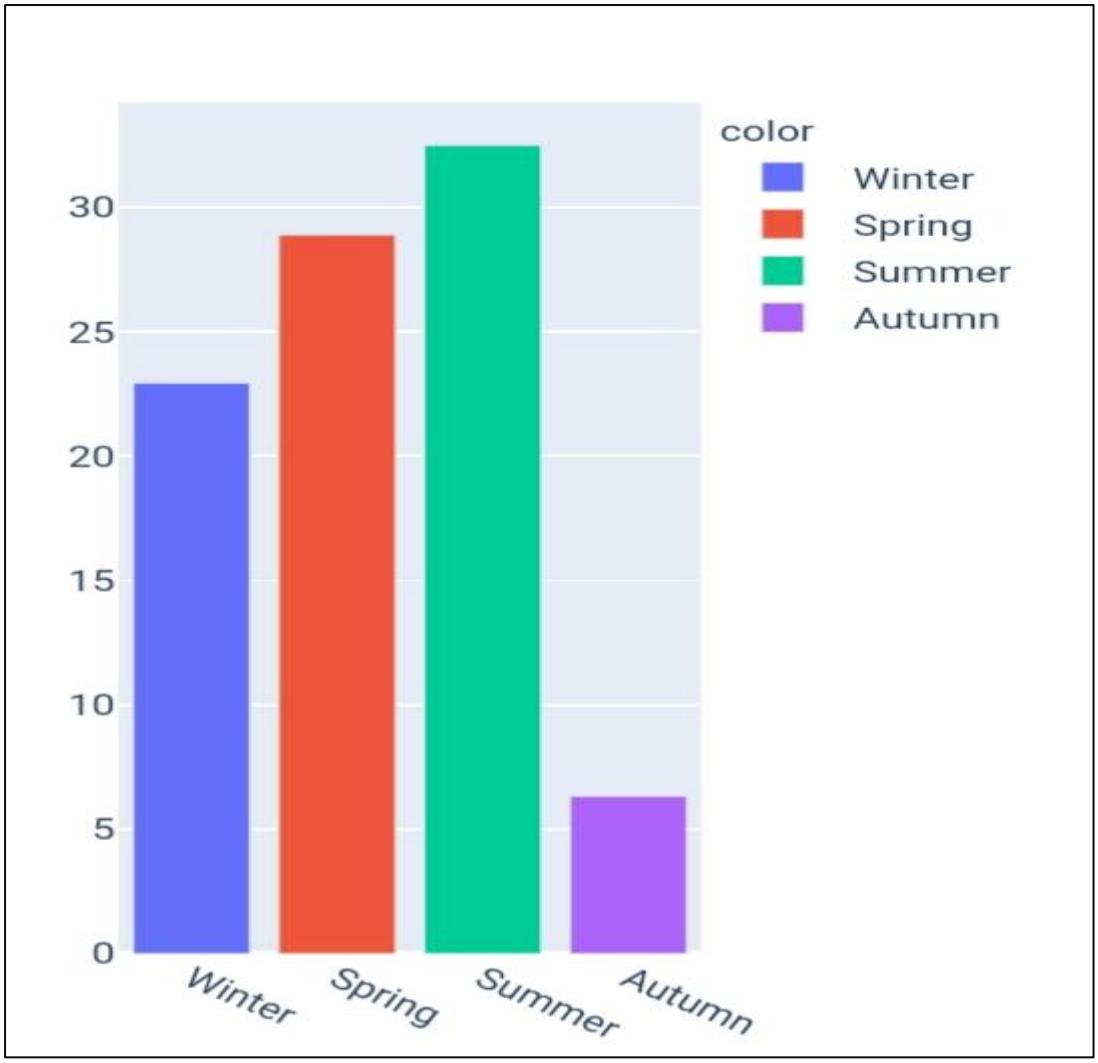
```
year
2000    15.983080
2001    19.488182
2002    17.455713
2003    32.173924
2004    21.477698
2005    26.974240
2006    30.470729
2007    26.539996
2008    28.102989
2009    23.550433
2010    32.079900
2011    26.790030
2012    22.835234
2013    29.153537
2014    22.957235
2015    33.140839
2016    23.913655
Name: rainfall-MM, dtype: float64
```

INPUT NO: 12

SEASON WISE GRAPH PLOTTING:

Now we will plot the graph season wise which will show that which season receives highest rainfall.

```
y=data.mean()
x=data.columns
fig = px.bar(x=x,y=y,color=x,title='Se
ason wise Rainfall in Pakistan')
fig.update(layout=dict(title=dict(x=0.
5)))
fig.show()
```


color
Winter
Spring
Summer
Autumn
30
25
20
15
10
5
0
Winter
Spring
Summer
Autumn