UNIVERSITY OF KARACHI


# REAL WORLD DATA ANALYSIS REPORT

# FIFA WORLD CUP 2018

**Dated:06-02-2022**

| | |
|---|---|
| **Degree program** | **BS Financial Mathematics (Second year G-2)** |
| **Name of project** | **Data analysis and visualization on Fifa World cup 2018** |
| **Submitted by** | • **Sayyaf Hussain**<br>• **Zainab Sukhera**<br>• **M.Irfan Sagar** |
| **Submitted to** | **Sir Syed Umaid Ahmed** |

# ABSTRACT

Data is the new electricity. We are living in the age of the fourth industrial revolution. This is the era of Artificial Intelligence and Big Data. There is a massive data explosion that has resulted in the culmination of new technologies and smarter products. Around 2.5 Exabyte's of Data is created each day. We need to keep the record of the data to do its analysis and visualization. Today the world is producing the data in zeta byte which is not handled by the normal programming languages, so many advanced high-level programming languages introduced like Python.

In this project, I combined my love for football and data sciences to do the analysis of Fifa World cup 2018 using Python (Jupyter notebook). In this, we discovered many insights that we may not usually noticed while watching a game. The main goal of this project is to go through the analysis of this data taken from the website Kaggle, along with its graphical representation of many attributes. It will increase the sense of awareness amongst the viewers about Fifa World cup 2018 in many aspects.

# Table of Contents

# Article I. CHAPTER # 1

## Section 1.01 INTRODUCTION

**Football is the most important of the less important things in the world.**

**-CARLO ANCELOTTI**

More than four out of ten people consider themselves a football fans and making is a world's popular sport. It is estimated that there are around 250 million football players all around the world, including 240 million amateurs. Our data analysis also based on FIFA WORLD CUP 2018.

The 2018 FIFA World Cup was an international football tournament contested by men's national teams that took place between 14 June and 15 July 2018 in Russia. It was the 21st FIFA World Cup, a worldwide football tournament held once every four years. It was the eleventh time the championships had been held in Europe, and the first time they were held in Eastern Europe. At an estimated cost of over $14.2 billion, it was the most expensive World Cup to date.

In this project, we are analyzing the data taken from Kaggle , for giving the visualization of different characteristics and to answer the statements:

- Number of matches held in each Russian city.
- Number of matches held in each stadium.
- Number of goals are scored in each day
- Number of matches held in each day
- Number of matches held in each hour
- Total goals scored by each team

For this, we do the following analysis because as we all know that:

**"If you torture the data long enough, it will confess to anything"**

**-RONALD H. COASE**

# Article II. CHAPTER # 2

## Section 2.01 DATA ANALYSIS AND VISUALIZATION

### (a) DATA ANALYSIS:

Data analysis is defined as a process of cleaning, transforming, and modeling data to discover useful information for business decision-making. The purpose of Data Analysis is to extract useful information from data and taking the decision based upon the data analysis.

Its life cycle based upon the following steps:

## DATA ANALYSIS AND VISUALIZATION

For analyzing and visualizing this data, we use different libraries:

- Pandas
- Numpy
- Matplotlib
- Seaborn
- Pycountry
- OS
- Datetime
- Warning

### (i) Pandas:

Pandas is a Python library for data analysis. t has functions for analyzing, cleaning, exploring, and manipulating data.

### (ii) NumPy:

NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices. NumPy stands for Numerical Python.

### (iii) Matplotlib:

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.

### (iv) Seaborn:

Seaborn is a library for making statistical graphics in Python. It helps you explore and understand your data.

### (v) Py-country:

A Python library to access ISO country, subdivision, language, currency and script definitions and their translations.

### (vi) OS:

The OS module in Python provides functions for creating and removing a directory (folder), fetching its contents, changing and identifying the current directory, etc.

### (vii) DateTime:

Python Datetime module supplies classes to work with date and time. These classes provide a number of functions to deal with dates, times and time intervals.

### (viii) Warning:

Warning messages are typically issued in situations where it is useful to alert the user of some condition in a program, where that condition (normally) doesn't warrant raising an exception and terminating the program.

## DATA ANALYSIS AND VISUALIZATION

### (ix) CODING SECTION:

**In this section we are briefly describing the coding which is done in this project step by step:**

#### (x) Step #1: Importing libraries and modules:

Here we are importing libraries and modules which are helpful in data analysis and its plotting:

```
In [2]: import numpy as np # linear algebra
        import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
        import matplotlib.pyplot as plt
        import seaborn as sns
        sns.set_style('whitegrid')
        from datetime import datetime
        import warnings
        warnings.filterwarnings('ignore')
        import os
        for dirname, _, filenames in os.walk('/kaggle/input'):
            for filename in filenames:
                print(os.path.join(dirname, filename))
        import pycountry # we uses pycountry for converting the country name as int form it help us to count the country or city when we
```

## DATA ANALYSIS AND VISUALIZATION

### (xi) Step #2: Uploading the CSV file:

After importing the libraries, we upload the data set of Fifa World cup 2018 by using pandas library.

```
In [3]: # by using the library of pandas we uploaded the csv file
        cup = pd.read_csv('worldcup.csv')
        cup.head(4)
```

Out[3]:

| | Year | Datetime | Stage | Stadium | City | Home Team Name | Home Team Goals | Away Team Goals | Away Team Name | Referee | Assistant 1 | Assistant 2 | MatchID | Attendance |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2018 | 14 Jun 2018 - 18:00 | Group A | Luzhniki Stadium, | Moscow | Russia | 5 | 0 | Saudi Arabia | PITANA Nestor (ARG) | MAIDANA Hernan (ARG) | BELATTI Juan Pablo (ARG) | 300331503 | 78011 |
| 1 | 2018 | 15 Jun 2018 - 17:00 | Group A | Ekaterinburg Arena, | Ekaterinburg | Egypt | 0 | 1 | Uruguay | KUIPERS Bjorn (NED) | VAN ROEKEL Sander (NED) | ZEINSTRA Erwin (NED) | 300353632 | 27015 |
| 2 | 2018 | 15 Jun 2018 - 18:00 | Group B | Saint Petersburg Stadium, | St. Petersburg | Morocco | 0 | 1 | IR Iran | CAKIR Cuneyt (TUR) | DURAN Bahattin (TUR) | ONGUN Tarik (TUR) | 300331526 | 62548 |
| 3 | 2018 | 15 Jun 2018 - 21:00 | Group B | Fisht Stadium, | Sochi | Portugal | 3 | 3 | Spain | ROCCHI Gianluca (ITA) | DI LIBERATORE Elenito (ITA) | TONOLINI Mauro (ITA) | 300331524 | 43866 |

**“Nothing is perfect” as rightly said by Hugh Mackay, in this case, it relates to our data, the data we have many unnecessary columns, complex names which we are making easy in the further steps.**

### (xii) Step #3: Dropping unnecessary columns:

We are removing the unnecessary columns by using the df. drop command:

```
In [4]: # by using drop so we could drop some coloumn that we dont need it.
        cup.drop(['Year', 'MatchID','Referee','Assistant 1','Assistant 2'],axis=1, inplace=True)
        cup.head(3)
```

Out[4]:

| | Datetime | Stage | Stadium | City | Home Team Name | Home Team Goals | Away Team Goals | Away Team Name | Attendance |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 14 Jun 2018 - 18:00 | Group A | Luzhniki Stadium, | Moscow | Russia | 5 | 0 | Saudi Arabia | 78011 |
| 1 | 15 Jun 2018 - 17:00 | Group A | Ekaterinburg Arena, | Ekaterinburg | Egypt | 0 | 1 | Uruguay | 27015 |
| 2 | 15 Jun 2018 - 18:00 | Group B | Saint Petersburg Stadium, | St. Petersburg | Morocco | 0 | 1 | IR Iran | 62548 |

### (xiii) Step #4: Simplifying the data for further analyzation and visualization:

At this point, it is important to apply smaller transformations that help to specify the data set. The headings are too long so we change it into simpler form.

```
In [5]: # the headings are to long to write so we are renaming it because for further working it help us to use in coding.
        cup.rename(columns={'Home Team Name': 'Home_team',
                            'Away Team Name': 'Away_team',
                            'Home Team Goals': 'Home_goals',
                            'Away Team Goals': 'Away_goals'
                            }, inplace=True)

In [6]: cup.head(3)
```

Out[6]:

| | Datetime | Stage | Stadium | City | Home_team | Home_goals | Away_goals | Away_team | Attendance |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 14 Jun 2018 - 18:00 | Group A | Luzhniki Stadium, | Moscow | Russia | 5 | 0 | Saudi Arabia | 78011 |
| 1 | 15 Jun 2018 - 17:00 | Group A | Ekaterinburg Arena, | Ekaterinburg | Egypt | 0 | 1 | Uruguay | 27015 |
| 2 | 15 Jun 2018 - 18:00 | Group B | Saint Petersburg Stadium, | St. Petersburg | Morocco | 0 | 1 | IR Iran | 62548 |

## (b) DATA VISUALIZATION:

Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data.

**In this section we are giving the answers of the questions mention above in introduction by visualizing the data:**

### (i) Step #5: Number of matches held in each Russian city:

Here comes the tricky part, for plotting the graph we have to make new column of hours by separating Date and Time:

```
In [7]: # we are separating the date and time here and adding another column for hours.
        # Further it will be use for graph plotting
        cup['Hour'] = cup.Datetime.apply(lambda x: x.split(' - ')[1])
        cup.Datetime = cup.Datetime.apply(lambda x: x.split(' - ')[0])
        cup.head(3)
```

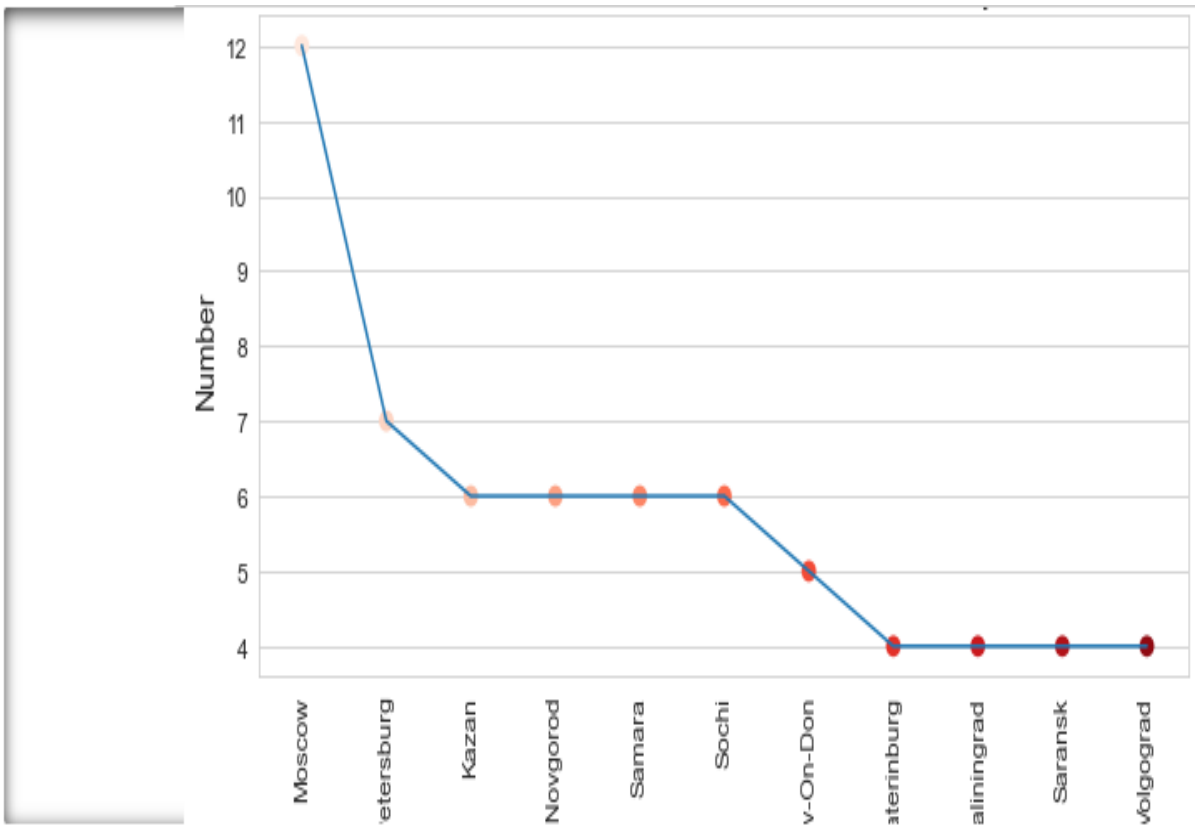
Out[7]:

| | Datetime | Stage | Stadium | City | Home_team | Home_goals | Away_goals | Away_team | Attendance | Hour |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 14 Jun 2018 | Group A | Luzhniki Stadium, | Moscow | Russia | 5 | 0 | Saudi Arabia | 78011 | 18:00 |
| 1 | 15 Jun 2018 | Group A | Ekaterinburg Arena, | Ekaterinburg | Egypt | 0 | 1 | Uruguay | 27015 | 17:00 |
| 2 | 15 Jun 2018 | Group B | Saint Petersburg Stadium, | St. Petersburg | Morocco | 0 | 1 | IR Iran | 62548 | 18:00 |

## DATA ANALYSIS AND VISUALIZATION

Now, we are using the library Seaborne to plot a graph of “Number of Matches held in each Russian City”:

```
In [18]: plt.figure(figsize=(10,5)) # it help us to draw a figure in size.
         by_city = cup.groupby('City').count().reset_index()[['City','Datetime']].sort_values('Datetime', ascending=False)# this line help
         sns.lineplot(by_city.City, by_city.Datetime, palette='vlag')# here we are importing the seaborn library for making line in the gr
         sns.pointplot(by_city.City, by_city.Datetime, palette='Reds')#using seaborn for point plotting
         # tick is basically used for the specific point on the coordinate axis.
         plt.xticks(rotation=90, fontsize=12)
         plt.yticks(fontsize=12)
         plt.title('Number of Matches held in each Russian City', fontsize=16)
         plt.xlabel('City Name', fontsize=14)
         plt.ylabel('Number', fontsize=14)
```

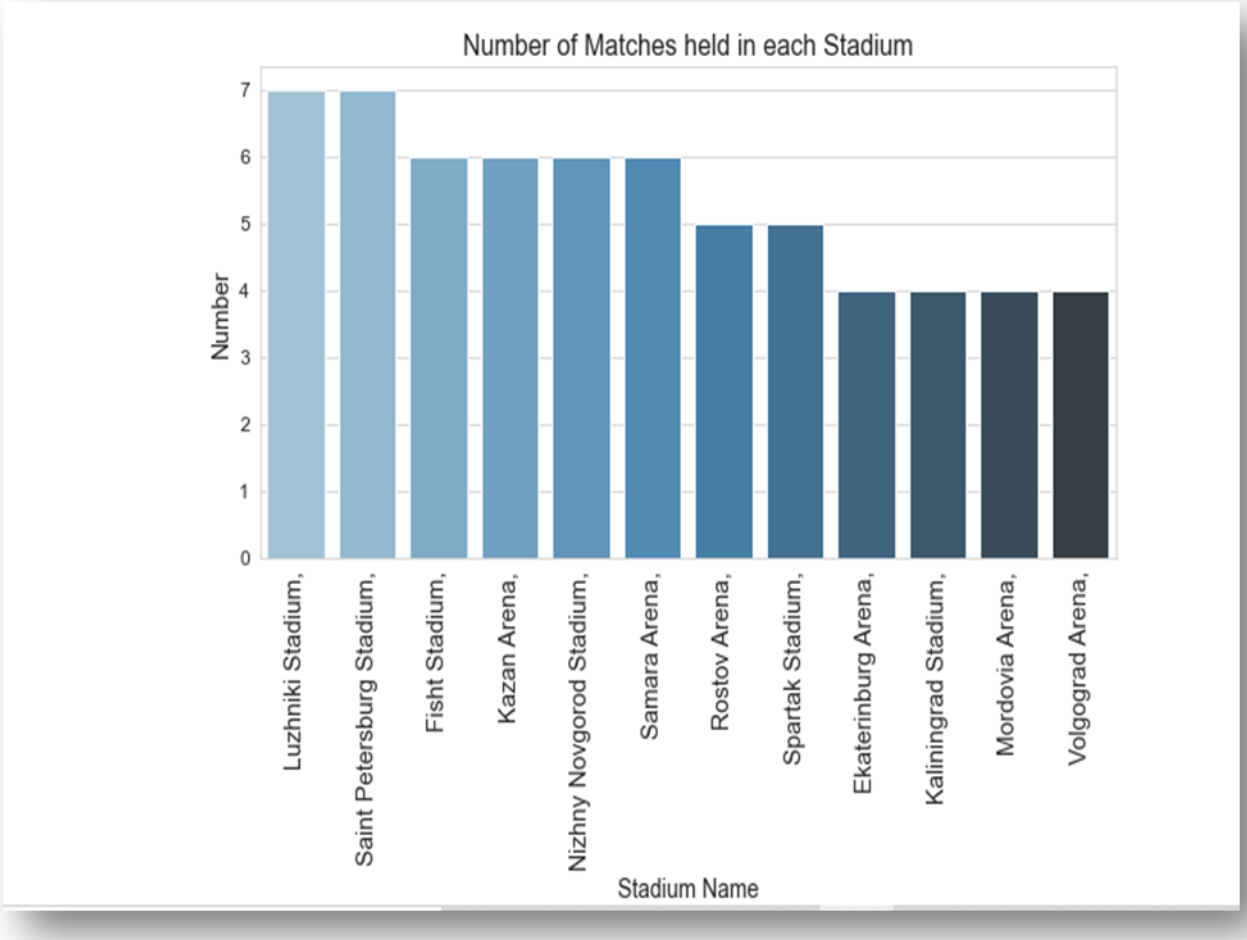


### (ii) Step #6: Number of matches held in each stadium:

By following the same pattern as the above graph, we are plotting the graph of "Number of matches held in each stadium":

```
In [9]: # same pattern that we follow from the previous graph of "Russia city matches"
        plt.figure(figsize=(10,5))
        by_city = cup.groupby('Stadium').count().reset_index()[['Datetime','Stadium']].sort_values('Datetime', ascending=False)
        sns.barplot(by_city.Stadium, by_city.Datetime, palette='Blues_d')
        plt.xticks(rotation=90, fontsize=14)
        plt.yticks(fontsize=12)
        plt.title('Number of Matches held in each Stadium', fontsize=16)
        plt.xlabel('Stadium Name', fontsize=14)
        plt.ylabel('Number', fontsize=14)

Out[9]: Text(0, 0.5, 'Number')
```



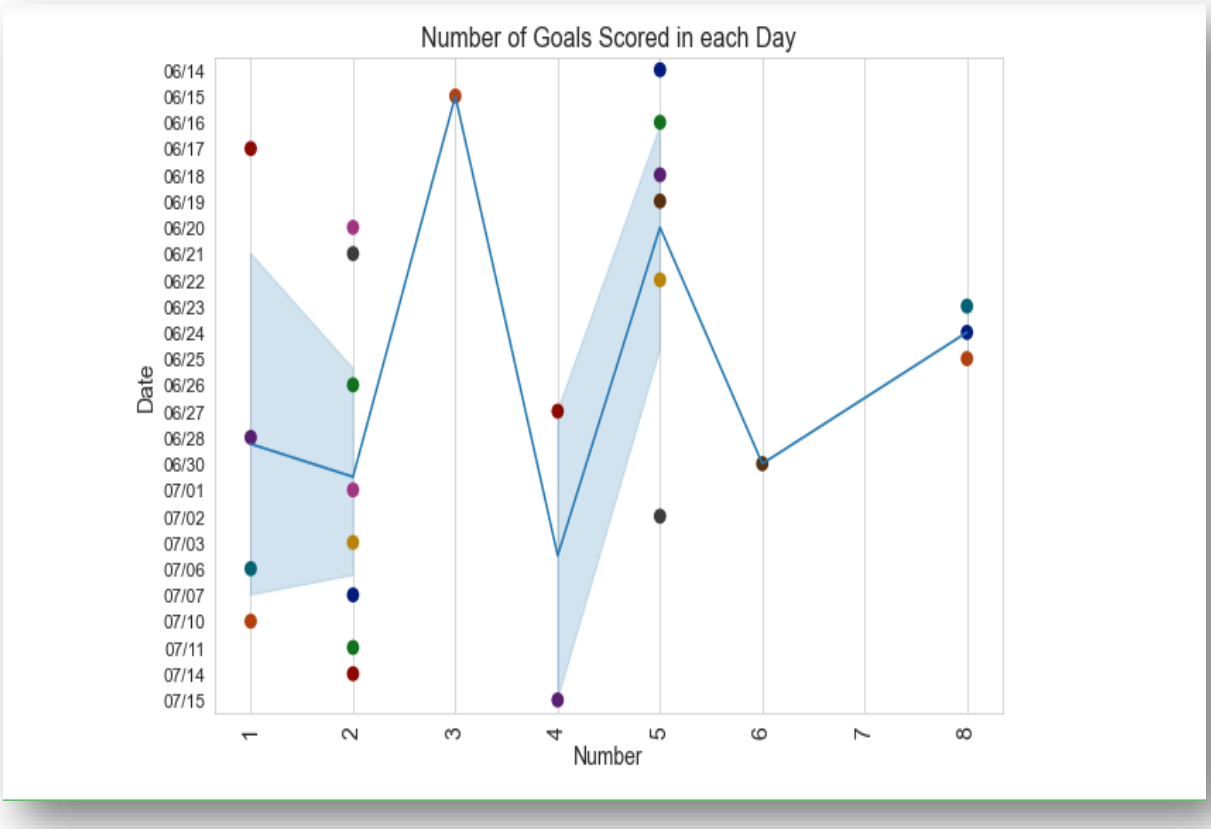
### (iii) Step #7: Numbers of goals scored each day:

By splitting the data into two groups, and then adding them we are making the graph of “Number of goals scored each day”:

```
In [10]: plt.figure(figsize=(10,7))
         #we use groupby for split the data  into groups and we uses sum for adding the the numerical values which help us to plotthe grap
         # reset _index is used to generate a new datafaram with out adding the column in your csv.
         # it form sets a list of integer ranging from 0 to lenght of data as index.
         by_goals = cup.groupby('Datetime').sum().reset_index()[['Home_goals','Datetime']]
         # %d is used for specify the integer values, decimals or numbers.
         # %b is used for convert the date into human readable form.
         by_goals.Datetime = pd.to_datetime(by_goals.Datetime, format='%d %b %Y')
         # than we create a function from which we use the datetime.
         by_goals.Datetime = by_goals.Datetime.apply(lambda x: datetime.strftime(x, '%m/%d'))
         # here we sort the values from our variable that we created.
         by_goals = by_goals.sort_values('Datetime')

         # the process is same as above.
         sns.lineplot(by_goals.Home_goals, by_goals.Datetime, palette='cool_d')
         sns.pointplot(by_goals.Home_goals, by_goals.Datetime, palette='dark')
         plt.xticks(rotation=90, fontsize=14)
         plt.yticks(fontsize=12)
         plt.title('Number of Goals Scored in each Day', fontsize=16)
         plt.xlabel('Number', fontsize=14)
         plt.ylabel('Date', fontsize=14)

Out[10]: Text(0, 0.5, 'Date')
```

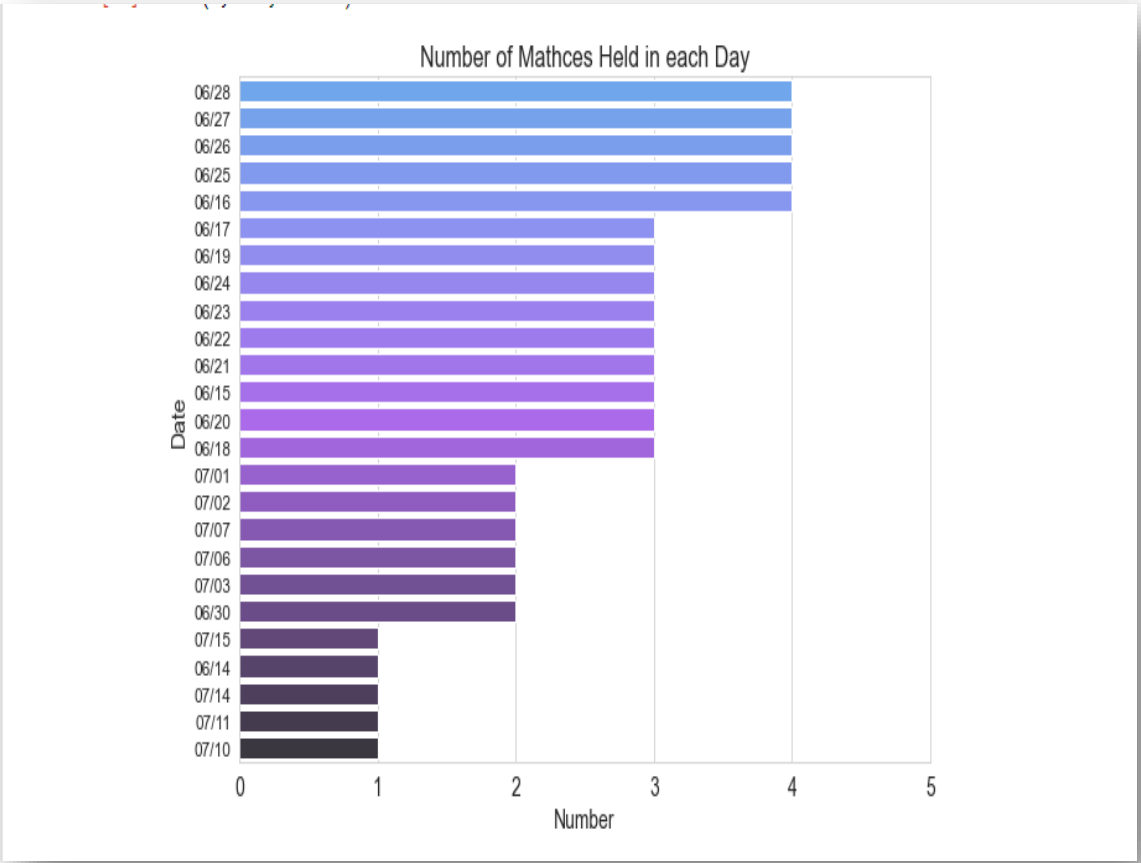


### (iv) Step #9: Number of matches held in each day:

Following is the graph of “Number of matches held each day”:

```
In [28]: plt.figure(figsize=(10,7))
         games_per_day = cup.groupby('Datetime').count().reset_index()[['Datetime', 'Stadium']].sort_values('Stadium', ascending=False)
         #Now here we uses pd.to_datetime which convert string date time into python datetime object..
         games_per_day.Datetime = pd.to_datetime(games_per_day.Datetime, format='%d %b %Y')
         #we uses a function below in and in the function we uses strftime, strftime convert the datetime into string Repersentation
         games_per_day.Datetime = games_per_day.Datetime.apply(lambda x: datetime.strftime(x, '%m/%d'))
         #The process is same as above graph, accept here we introduce range, range is use for return a sequence of numbers starting from
         sns.barplot(games_per_day.Stadium, games_per_day.Datetime, palette='cool_d')
         plt.xticks(range(6), fontsize=14)
         plt.yticks(fontsize=12)
         plt.title('Number of Mathces Held in each Day', fontsize=16)
         plt.xlabel('Number', fontsize=14)
         plt.ylabel('Date', fontsize=14)

Out[28]: Text(0, 0.5, 'Date')
```

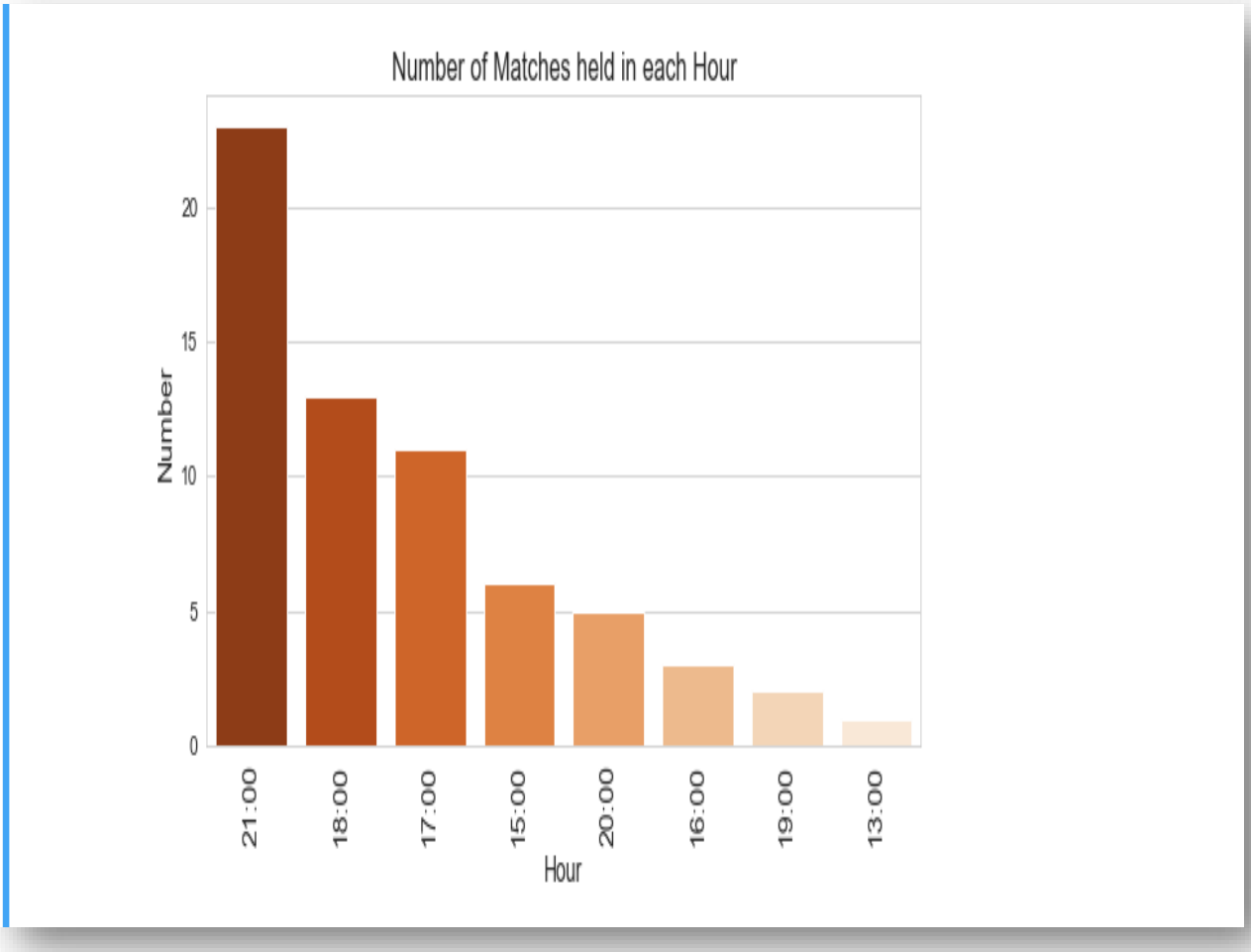


### (v) Step #10: Number of matches held in each hour:

As the above process, we give the graphical representation of “Number of matches held in each Hour”:

```
In [29]: # here we plot number of matches held in each hours the process is same..
         plt.figure(figsize=(10,5))
         games_by_hour = cup.Hour.value_counts().to_frame().reset_index()
         games_by_hour.columns = ['Hour', 'Number of Mathces']
         sns.barplot(games_by_hour.Hour, games_by_hour['Number of Mathces'], palette='Oranges_r')
         plt.xticks(rotation=90, fontsize=14)
         plt.yticks(fontsize=12)
         plt.title('Number of Matches held in each Hour', fontsize=16)
         plt.xlabel('Hour', fontsize=14)
         plt.ylabel('Number', fontsize=14)

Out[29]: Text(0, 0.5, 'Number')
```



### (vi) Step #11: Total goals scored by team:

By making a new column of “Total goals”, we visualize the data of “Total goals scored by each team”:

```
In [49]: # here we create a column which is going to add in our CSV file and in this columns we are going to add the values of Home_goals
         # and Away_goals
         cup['Total_goals'] = cup.Home_goals + cup.Away_goals
         #here we uses sum to adds up all the numerical values in an iterable, such as a list and return the total of those values.
         goals_by_day = cup.groupby('Datetime').Total_goals.sum().to_frame().reset_index()
         # we are creating the columns but these columns are not going to prsent in the CSV file the work in the backend of the terminal
         goals_by_day.columns = ['Datetime2', 'Total_Goals']
         goals_by_day.Datetime2 = pd.to_datetime(goals_by_day.Datetime2, format='%d %b %Y')
         goals_by_day.Datetime2 = goals_by_day.Datetime2.apply(lambda x: datetime.strftime(x, '%m/%d'))
         # we use pd.concat for separating the overlapping
         goal_ratio = pd.concat([goals_by_day, games_per_day],axis=1).drop('Datetime', axis=1)
         goal_ratio.columns = ['Datetime', 'Total_Goals', 'num_of_Matches']
         goal_ratio['Ratio'] = goal_ratio.Total_Goals / goal_ratio.num_of_Matches
         goal_ratio = goal_ratio.sort_values('Ratio', ascending=False)
         cup.head(5)
```

Out[49]:

| | Datetime | Stage | Stadium | City | Home_team | Home_goals | Away_goals | Away_team | Attendance | Hour | Total_goals |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 14 Jun 2018 | Group A | Luzhniki Stadium, | Moscow | Russia | 5 | 0 | Saudi Arabia | 78011 | 18:00 | 5 |
| 1 | 15 Jun 2018 | Group A | Ekaterinburg Arena, | Ekaterinburg | Egypt | 0 | 1 | Uruguay | 27015 | 17:00 | 1 |
| 2 | 15 Jun 2018 | Group B | Saint Petersburg Stadium, | St. Petersburg | Morocco | 0 | 1 | IR Iran | 62548 | 18:00 | 1 |
| 3 | 15 Jun 2018 | Group B | Fisht Stadium, | Sochi | Portugal | 3 | 3 | Spain | 43866 | 21:00 | 6 |
| 4 | 16 Jun 2018 | Group C | Kazan Arena, | Kazan | France | 2 | 1 | Australia | 41279 | 13:00 | 3 |

## DATA ANALYSIS AND VISUALIZATION

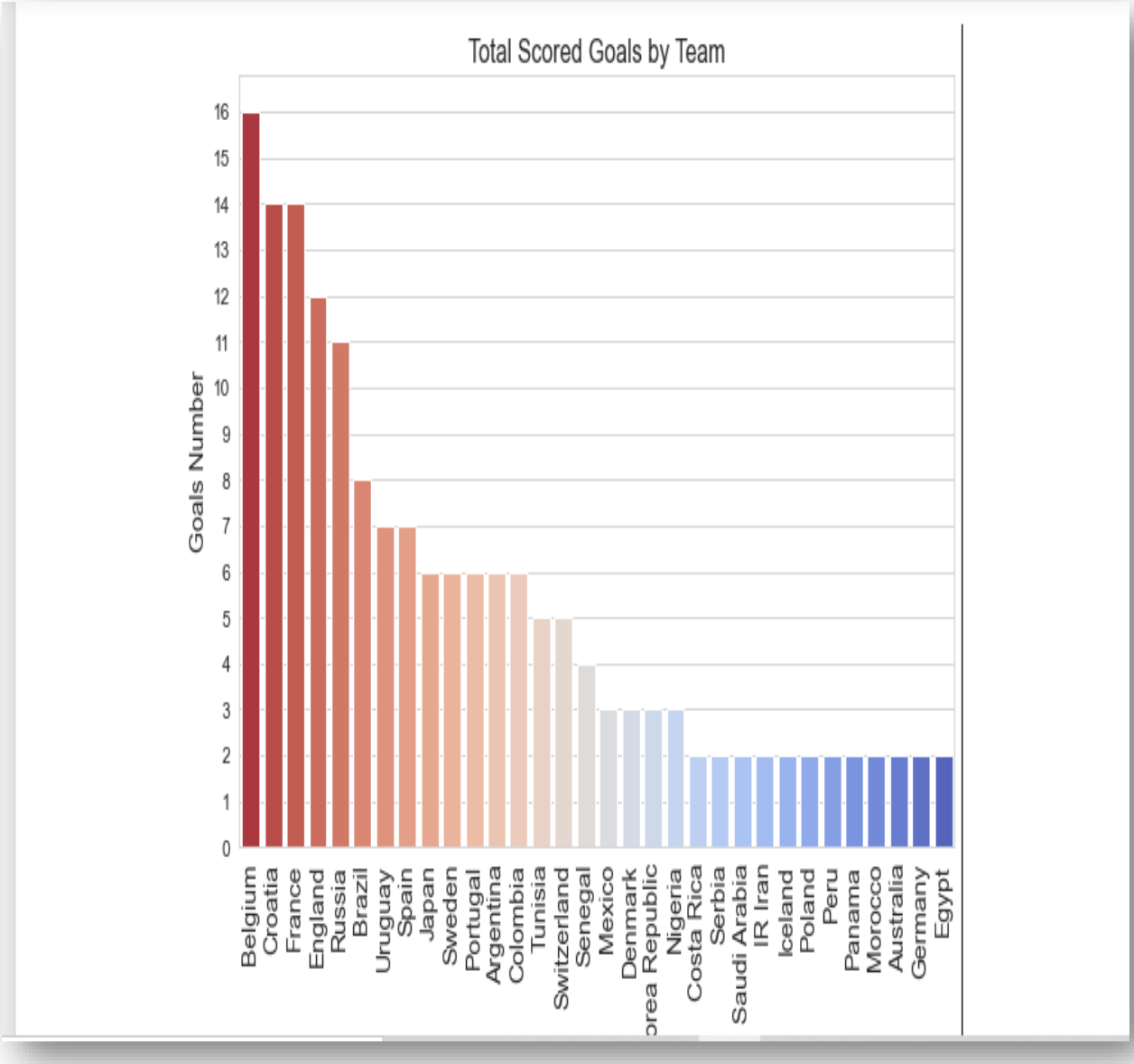
Then we analyze the score of each teams, and make a new column of goal diff, for plotting the graph:

In [50]:
```
# In here we are analysing the score of each team, how many goals that scored by each teams, how many they received and
#the difference of it.
goals_by_home = cup.groupby('Home_team').sum()[['Home_goals', 'Away_goals']].reset_index()
goals_by_away = cup.groupby('Away_team').sum()[['Home_goals', 'Away_goals']].reset_index()
goals_total = pd.concat([goals_by_home, goals_by_away],axis=1)
goals_total.columns = ['Home_team','Home_Scored', 'Home_Received', 'Away_team', 'Away_Received', 'Away_Scored']
goals_total['Scored'] = goals_total.Home_Scored + goals_total.Away_Scored
goals_total['Received'] = goals_total.Home_Received + goals_total.Away_Received
goals_total = goals_total.drop(['Home_Scored', 'Home_Received', 'Away_team', 'Away_Scored', 'Away_Received'], axis=1)
goals_total['Goal_diff'] = goals_total.Scored - goals_total.Received
goals_total = goals_total.rename(columns={'Home_team': 'Team_name'})
goals_total.head(7)
```

Out[50]:

| | Team_name | Scored | Received | Goal_diff |
|---|---|---|---|---|
| 0 | Argentina | 6 | 9 | -3 |
| 1 | Australia | 2 | 5 | -3 |
| 2 | Belgium | 16 | 6 | 10 |
| 3 | Brazil | 8 | 3 | 5 |
| 4 | Colombia | 6 | 3 | 3 |
| 5 | Costa Rica | 2 | 5 | -3 |
| 6 | Croatia | 14 | 9 | 5 |

## DATA ANALYSIS AND VISUALIZATION



# Article III. CHAPTER # 3

## Section 3.01 CONCLUSION

In recent times, Python comes out as an emerging programming language. Due to its advantages and easy syntax, it provides us to do data analysis in a very simple and easy way.

For a data analyst, it is a very engaging to do such type of analysis which is captivating and also a learn full experience at the same time. This analysis, cleared up many concepts which we didn't understand before. This visualization is designed for non-experts who are die heart fan of football, to know about the Fifa World cup 2018.

In doing so, we encountered many difficulties but we managed to solved them calmly. We take help from our teacher and internet and make this analysis a worth doing. But our main goal is to turn data into information and information into insight, and using python (Jupyter notebook) we analyzed this dataset of FIFA WORLD CUP 2018 and make many things visualized i.e.no of goals by each team, goals per day, goals in each hour etc.

In today world, data sciences and visualization is very important for everyone. According to Radi (Data analyst at Centogene):

**Data analytics is a future, and the future is NOW! Everything is about data these days. Data is information and information is power.**

**-Radi**

# Article IV. REFERENCES

https://www.kaggle.com/shivan118/fifa-world-cup-data-analysis

https://en.wikipedia.org/wiki/2018_FIFA_World_Cup