Lab 1

1. Get acquainted with data science tools and perform statistical analysis

1.1 Objective

- To familarize yourself with essential tools and libraries used in data science.
- To learn and to describe data using descriptive statistics.

1.2 Theory

Python

Python is a versatile and widely-used programming language for data science due to its simplicity and rich ecosystem of libraries. Its popular libraries include Pandas, NumPy, Scikit-learn, Matplotlib, Seaborn, etc. Python is used for data manipulation, visualization, machine learning, and automation.

Anaconda

Anaconda is a distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment.

Jupyter Notebook

The Jupyter Notebook is an open-source web application that allows you to create and share documents taht contain live, code, equations, visualizations, and narrative text. Its uses include data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

SciPy

SciPy is a scientific computation library that uses NumPy underneath. SciPy stands for Scientific Python. It provides more utility functions for optimization, stats and sginal processing. Like NumPy, SciPy is open source so we can use it freely. SciPy, stands for Scientific Python, is used for numerical computations in Python. Both these packages provide extended functionality to work with Python.

Stats models

Stats models is a popular library in Python that enables us to estimate and analyze various statistical models. It is built on numeric and scientific libraries like NumPy and SciPy. It includes various models of linear regression like ordinary least squares, generalized least squares, weighted least squares, etc

Pandas

Pandas are really powerful. They provide you with a huge set of important commands and features which are used to easily analyze your data. We can use Pandas to perform various tasks like filtering your data according to certain conditions, or segmenting and segregating the data according to preference, etc.

Matplotlib

Matplotlib is a Python library used for creating static, interactive, and animated visualizations. It provides a variety of plotting options, such as line plots, bar charts, scatter plots, and histograms. Matplotlib is highly customizable and serves as the foundation for other visualization libraries like Seaborn.

Seaborn

Seaborn is a Python library built on Matplotlib, designed to simplify the creation of visually appealing and informative statistical graphics. It includes features like heatmaps, violin plots, pair plots, and categorical plots. Seaborn integrates seamlessly with Pandas, making it ideal for exploratory data analysis.

Scikit-learn

Scikit-learn is a powerful machine-learning library in Python that provides tools for supervised and unsupervised learning, such as regression, classification, clustering, and dimensionality reduction. It also includes utilities for model selection, preprocessing, and evaluation. Scikit-learn is built on NumPy, SciPy, and Matplotlib.

Google Colab

Google Colab is a cloud-based platform for running Python code, offering free GPU and TPU resources. It supports popular data science libraries like TensorFlow, PyTorch, Pandas, and NumPy. Google Colab is commonly used for machine learning, data analysis, and collaborative projects.

Descriptive Statistics

Descriptive statistics is a branch of statistics that focuses on summarizing and organizing data to uncover patterns, relationships, and trends. It provides a foundation for data analysis by offering methods to describe and present data meaningfully.

Types of Descriptive statistics:

1. **Measures of Central Tendency:** Central tendency provides a single value that represents the center of typical value of a dataset.
   - **Mean**(Average): The sum of all data points divided by the total number of points.
   - **Median**: The middle vlaue of an ordered dataset. It is less sensitive to outliers.
   - **Mode**: The value that appears the most frequently in dataset.
2. **Measures of Dispersion (Spread):** Dispersion measures how spread out the data values are around the central tendency.
   - **Range:** The difference between the maximum and minimum values.
   - **Variance:** The average squared deviation from the mean.

   - **Standard Deviation:** The square root of the variance, showing how data points deviate from the mean.
   - **Interquartile Range (IQR): The range of the middle 50% of the data, calculated as Q3-Q1.
3. **Measures of Shape:** These metrics descrbe the distribution and symmetry of the data.
   - **Skewness:** Measures asymmetry. Positive skew indicates a longer tail on the right; negative skew indicates a longer tail on the left.
   - **Kurtosis:** Measure the "tailedness" of the distribution. High kurtosis means heavy tails; low kurtosis means light tails.

1.3 Installation Process

Download Anaconda:

Step 1: Type "Anaconda Download" in Google Chrome. Visit the site "http://www.anaconda.com". Step2: Provide email to get a link for downloading or You can skip registration to download.

Step3: Scroll the anaconda products website below. Click the 64 bit installer as per your device.Let the download finish.

Step4: Click the downloaded.exe file and install Anaconda.

Step5: Agree terms and conditions and Click the Justme option.

Step6: Use default path to install in C drive.

Step7: Use default options...install it and after installation open the anaconda navigator.

Step8: Search for a Jupyter Notebook by scrolling.. If not installed , install it and Launch it.

Step9: Jupyter Notebook will be launched on your browser. Then Click on “New” and Python3.

Step10: Write your code and run it by shortcut... SHIFT+ENTER.

Installing Packages:

Step 1: Go to the folder anaconda on your start menu and open “ Anaconda Prompt”.
Step 2: Installing

- **Confirm Python is instlaled correctly, by typing:**

  ```
  python -v
  ```

- **Confirm conda is intalled correctly, by typing:**

  ```
  conda -v
  ```

- **You can install packages using pip or conda.**

  ```
  pip install <package_name>
  ```

or,

  ```
  conda install <package_name>
  ```

- **For installing a particular version of the package.**

  ```
  pip install <package_name>=1.11.0
  ```

- **For instaling multiple package all at once.**

  ```
  pip install <package_name1> <package_name2> <package_name3>
  ```

- **For updating package**

  ```
  pip install --upgrade <package_name>
  ```

- **For installing libraries for data-science**

  ```
  pip install numpy scipy statsmodel pandas seaborn matplotlib scikit-learn
  ```

or,

  ```
  conda install numpy scipy statsmodels pandas seaborn matplotlib scikit-learn
  ```

1.4 Code And Result

Installing Packages

In [26]:

```
!pip install numpy pandas matplotlib seaborn
```

```
Requirement already satisfied: numpy in c:\users\kaush.kaushal\anaconda3\lib\site-packages (1.26.4)
Requirement already satisfied: pandas in c:\users\kaush.kaushal\anaconda3\lib\site-packages (2.2.2)
Requirement already satisfied: matplotlib in c:\users\kaush.kaushal\anaconda3\lib\site-packages (3.9.2)
Requirement already satisfied: seaborn in c:\users\kaush.kaushal\anaconda3\lib\site-packages (0.13.2)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\kaush.kaushal\anaconda3\lib\site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\kaush.kaushal\anaconda3\lib\site-packages (from pandas) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in c:\users\kaush.kaushal\anaconda3\lib\site-packages (from pandas) (2023.3)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\kaush.kaushal\anaconda3\lib\site-packages (from matplotlib) (1.2.0)
Requirement already satisfied: cycler>=0.10 in c:\users\kaush.kaushal\anaconda3\lib\site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\kaush.kaushal\anaconda3\lib\site-packages (from matplotlib) (4.51.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\kaush.kaushal\anaconda3\lib\site-packages (from matplotlib) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\kaush.kaushal\anaconda3\lib\site-packages (from matplotlib) (24.1)
Requirement already satisfied: pillow>=8 in c:\users\kaush.kaushal\anaconda3\lib\site-packages (from matplotlib) (10.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\kaush.kaushal\anaconda3\lib\site-packages (from matplotlib) (3.1.2)
Requirement already satisfied: six>=1.5 in c:\users\kaush.kaushal\anaconda3\lib\site-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
```

In [27]:

```python
import numpy as np
import pandas as pd
import seaborn as sb
import matplotlib.pyplot as plt
```

Loading CSV

In [29]:

```python
df_csv = pd.read_csv("FIFA - World Cup Summary.csv")
```

Viewing the top and bottom 5 elements

In [31]:

```python
df_csv.head()
```

Out[31]:

	YEAR	HOST	CHAMPION	RUNNER UP	THIRD PLACE	TEAMS	MATCHES PLAYED	GOALS SCORED	AVG GOALS PER GAME
0	1930	Uruguay	Uruguay	Argentina	United States	13	16	70	3.6
1	1934	Italy	Italy	Czechoslovakia	Germany	16	17	70	4.1
2	1938	France	Italy	Hungary	Brazil	15	18	84	4.7
3	1950	Brazil	Uruguay	Brazil	Sweden	13	22	88	4.0
4	1954	Switzerland	West Germany	Hungary	Austria	16	26	140	5.4

In [32]:

```python
df_csv.tail()
```

Out[32]:

	YEAR	HOST	CHAMPION	RUNNER UP	THIRD PLACE	TEAMS	MATCHES PLAYED	GOALS SCORED	AVG GOALS PER GAME
17	2006	Germany	Italy	France	Germany	32	64	147	2.3
18	2010	South Africa	Spain	Netherlands	Germany	32	64	145	2.3
19	2014	Brazil	Germany	Argentina	Netherlands	32	64	171	2.7
20	2018	Russia	France	Croatia	Belgium	32	64	169	2.6
21	2022	Qatar	Argentina	France	Croatia	32	64	172	2.7

Shape of the dataframe

In [34]:

```python
df_csv.shape
```

Out[34]:

```
(22, 9)
```

This shows that the dataframe has 9 columns and 22 rows.

Checking the properties of the data columns:

In [37]:

```python
df_csv.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 22 entries, 0 to 21
Data columns (total 9 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   YEAR                22 non-null     int64  
 1   HOST                22 non-null     object 
 2   CHAMPION            22 non-null     object 
 3   RUNNER UP           22 non-null     object 
 4   THIRD PLACE         22 non-null     object 
 5   TEAMS               22 non-null     int64  
 6   MATCHES PLAYED      22 non-null     int64  
 7   GOALS SCORED        22 non-null     int64  
 8   AVG GOALS PER GAME  22 non-null     float64
dtypes: float64(1), int64(4), object(4)
memory usage: 1.7+ KB
```

This shows that the dataframe has 4 columns with integer data type, 4 objects, and 1 float.

Checking for null elements

In [40]:

```python
df_csv.isnull()
```

Out[40]:

	YEAR	HOST	CHAMPION	RUNNER UP	THIRD PLACE	TEAMS	MATCHES PLAYED	GOALS SCORED	AVG GOALS PER GAME
0	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False
5	False	False	False	False	False	False	False	False	False
6	False	False	False	False	False	False	False	False	False
7	False	False	False	False	False	False	False	False	False
8	False	False	False	False	False	False	False	False	False
9	False	False	False	False	False	False	False	False	False
10	False	False	False	False	False	False	False	False	False
11	False	False	False	False	False	False	False	False	False
12	False	False	False	False	False	False	False	False	False
13	False	False	False	False	False	False	False	False	False
14	False	False	False	False	False	False	False	False	False
15	False	False	False	False	False	False	False	False	False
16	False	False	False	False	False	False	False	False	False
17	False	False	False	False	False	False	False	False	False
18	False	False	False	False	False	False	False	False	False
19	False	False	False	False	False	False	False	False	False
20	False	False	False	False	False	False	False	False	False
21	False	False	False	False	False	False	False	False	False

In [41]:

```
df_csv.isnull().sum()
```

Out[41]:

```
YEAR                  0
HOST                  0
CHAMPION              0
RUNNER UP             0
THIRD PLACE           0
TEAMS                 0
MATCHES PLAYED        0
GOALS SCORED          0
AVG GOALS PER GAME    0
dtype: int64
```

Since the sum of null elements is zero, there are no null elements in the data frame.

Describing the dataframe

In [44]:

```
df_csv.describe()
```

Out[44]:

	YEAR	TEAMS	MATCHES PLAYED	GOALS SCORED	AVG GOALS PER GAME
count	22.000000	22.000000	22.000000	22.000000	22.000000
mean	1978.909091	22.227273	43.727273	123.636364	3.059091
std	27.738419	7.602830	17.776876	34.841882	0.831327
min	1930.000000	13.000000	16.000000	70.000000	2.200000
25%	1959.000000	16.000000	32.000000	90.500000	2.600000
50%	1980.000000	20.000000	45.000000	129.000000	2.700000
75%	2001.000000	32.000000	64.000000	146.750000	3.450000
max	2022.000000	32.000000	64.000000	172.000000	5.400000

Here, we can clearly see that the most amount of goals scored in a single tournament was 172 goals and highest goals per game was 5.4 whereas the least amount of goals scored in a tournament was 70 and average goals per game was 2.2. We also get the data about the mean, median, and quartiles of the goals scored and goals per game.

In [46]:

```
df_csv.describe(include="object")
```

Out[46]:

	HOST	CHAMPION	RUNNER UP	THIRD PLACE
count	22	22	22	22
unique	18	9	11	15
top	France	Brazil	Argentina	Germany
freq	2	5	3	3

By analyzing the objects we found out that Brazil has been the champion most no. of times(5) whereas Argentina and Germany have been Runner Up and Third respectively(3 each). France has hosted the most no. of world cups.

Describe all data types at once

In [49]:
```
df_csv.describe(include="all")
```

Out[49]:

	YEAR	HOST	CHAMPION	RUNNER UP	THIRD PLACE	TEAMS	MATCHES PLAYED	GOALS SCORED	AVG GOALS PER GAME
count	22.000000	22	22	22	22	22.000000	22.000000	22.000000	22.000000
unique	NaN	18	9	11	15	NaN	NaN	NaN	NaN
top	NaN	France	Brazil	Argentina	Germany	NaN	NaN	NaN	NaN
freq	NaN	2	5	3	3	NaN	NaN	NaN	NaN
mean	1978.909091	NaN	NaN	NaN	NaN	22.227273	43.727273	123.636364	3.059091
std	27.738419	NaN	NaN	NaN	NaN	7.602830	17.776876	34.841882	0.831327
min	1930.000000	NaN	NaN	NaN	NaN	13.000000	16.000000	70.000000	2.200000
25%	1959.000000	NaN	NaN	NaN	NaN	16.000000	32.000000	90.500000	2.600000
50%	1980.000000	NaN	NaN	NaN	NaN	20.000000	45.000000	129.000000	2.700000
75%	2001.000000	NaN	NaN	NaN	NaN	32.000000	64.000000	146.750000	3.450000
max	2022.000000	NaN	NaN	NaN	NaN	32.000000	64.000000	172.000000	5.400000

View all the columns of dataset

In [51]:
```
df_csv.columns
```

Out[51]:
```
Index(['YEAR', 'HOST', 'CHAMPION', 'RUNNER UP', 'THIRD PLACE', 'TEAMS',
       'MATCHES PLAYED', 'GOALS SCORED', 'AVG GOALS PER GAME'],
      dtype='object')
```

Store the columns to analyze into their own variables

In [53]:
```
hosts = df_csv["HOST"]
champions = df_csv["CHAMPION"]
goals = df_csv["GOALS SCORED"]
goals_per_game = df_csv["AVG GOALS PER GAME"]
```

View all unique hosts and count no. of times hosted

In [55]:
```
hosts.unique()
```

Out[55]:
```
array(['Uruguay', 'Italy', 'France', 'Brazil', 'Switzerland', 'Sweden',
       'Chile', 'England', 'Mexico', 'West Germany', 'Argentina', 'Spain',
       'United States', 'South Korea, Japan', 'Germany', 'South Africa',
       'Russia', 'Qatar'], dtype=object)
```

In [56]:
```
hosts.value_counts()
```

Out[56]:
```
HOST
France                 2
Brazil                 2
Mexico                 2
Italy                  2
Uruguay                1
Spain                  1
Russia                 1
South Africa           1
Germany                1
South Korea, Japan     1
United States          1
West Germany           1
Argentina              1
England                1
Chile                  1
Sweden                 1
Switzerland            1
Qatar                  1
Name: count, dtype: int64
```

Analyze the champions

In [58]:
```
champions.unique()
```

```
Out[58]: array(['Uruguay', 'Italy', 'West Germany', 'Brazil', 'England',
                'Argentina', 'France', 'Spain', 'Germany'], dtype=object)
```

```
In [59]: champions.value_counts()
```

```
Out[59]: CHAMPION
         Brazil          5
         Italy           4
         West Germany    3
         Argentina       3
         Uruguay         2
         France          2
         England         1
         Spain           1
         Germany         1
         Name: count, dtype: int64
```

Here we can see that Brazil have been most succesful teams with 5 trophies followed by Italy and Germany with 4 titles each and Argentina with 3 titles.

Computation measures of central tendency

As the no. of teams and no. of matches played varies in different world cup, analyzing the total no. of goals won't be effective. So, we will be analyzing the average no. of goals scored per match

```
In [62]: mean = goals_per_game.mean()
         median = goals_per_game.median()
         mode = goals_per_game.mode()

         print(f"Mean: {mean}\nMedian: {median}\nMode: {mode}")
```

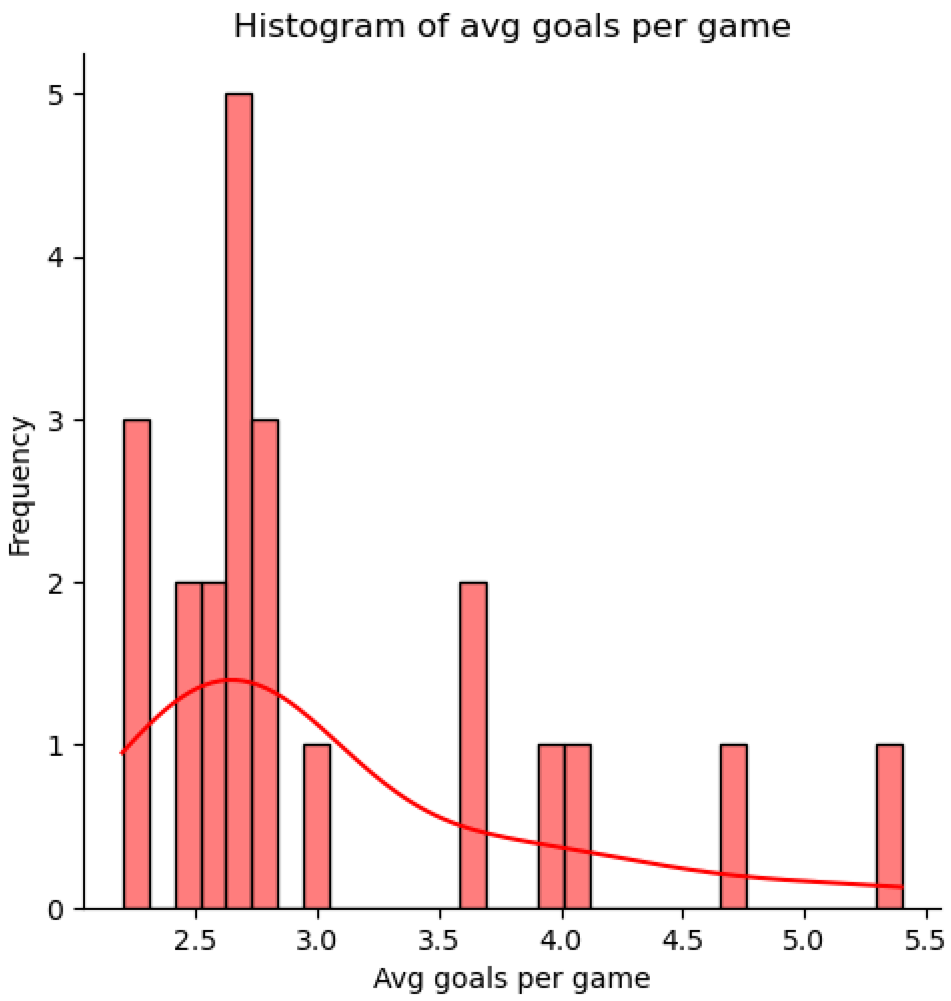
```
Mean: 3.059090909090909
Median: 2.7
Mode: 0    2.7
Name: AVG GOALS PER GAME, dtype: float64
```

Observation

Here, Mean > Median = Mode.

Plotting distribution

```
In [65]: sb.displot(goals_per_game, kde=True, bins=30, kind="hist", color="red")
         plt.title("Histogram of avg goals per game")
         plt.xlabel("Avg goals per game")
         plt.ylabel("Frequency")
         plt.show()
```



Computation measures of dispersion or variability

Minimum Value

```
In [67]: min = goals_per_game.min()
         min
```

```
Out[67]: 2.2
```

Maximum Value

```
In [69]: max = goals_per_game.max()
         max
```

```
Out[69]: 5.4
```

Range

```
In [71]: range = max - min
         range
```

```
Out[71]: 3.2
```

Variance

```
In [73]: goals_per_game.var()
```

```
Out[73]: 0.6911038961038959
```

Standard Deviation

```
In [75]: goals_per_game.std()
```

```
Out[75]: 0.8313265881131771
```

First Quratile

```
In [77]: Q1 = goals_per_game.quantile(0.25)
         Q1
```

```
Out[77]: 2.6
```

Second Quartile

```
In [79]: Q2 = goals_per_game.quantile(0.5)
         Q2
```

```
Out[79]: 2.7
```

Third Quratile

```
In [81]: Q3 = goals_per_game.quantile(0.75)
         Q3
```

```
Out[81]: 3.45
```

Interquartile Range

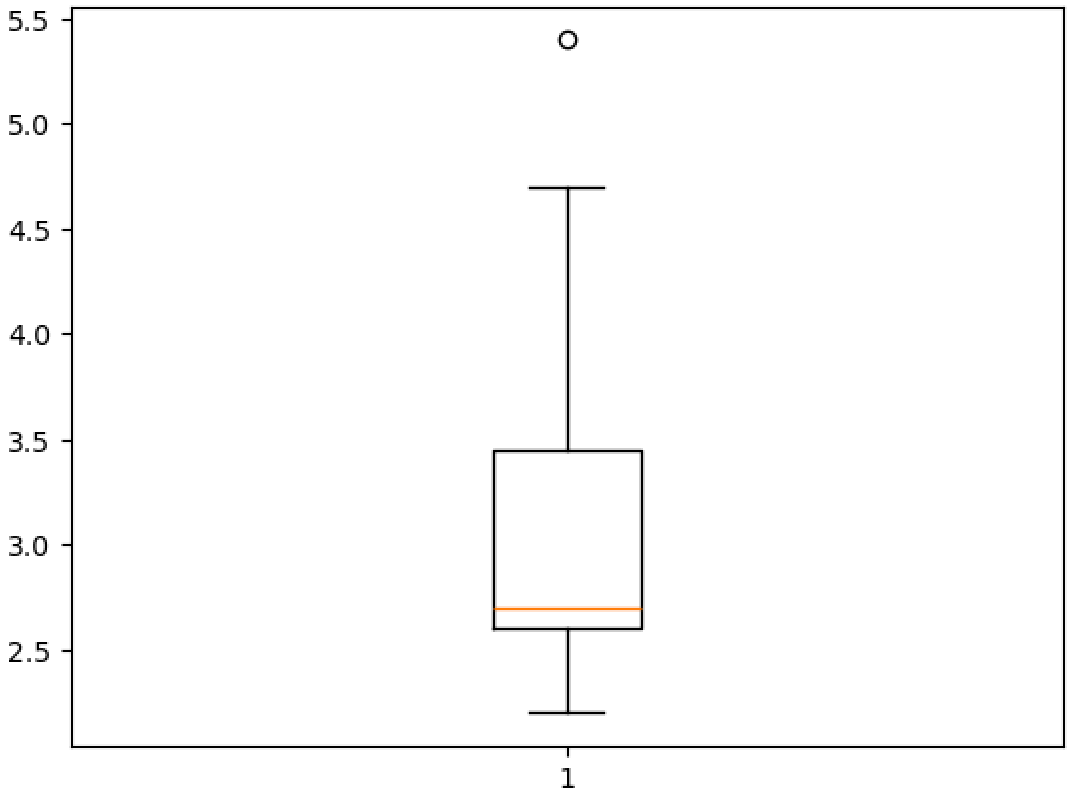
```
In [83]: IQR = Q3 - Q1
         IQR
```

```
Out[83]: 0.8500000000000001
```

Draw BoxPlot

```
In [85]: plt.boxplot(goals_per_game)
         plt.show
```

```
Out[85]: <function matplotlib.pyplot.show(close=None, block=None)>
```



Computation measures of shape or distribution

Skewness

In [88]:

```
goals_per_game.skew()
```

Out[88]:

```
1.5670229988341382
```

Kurtosis

In [90]:

```
goals_per_game.kurt()
```

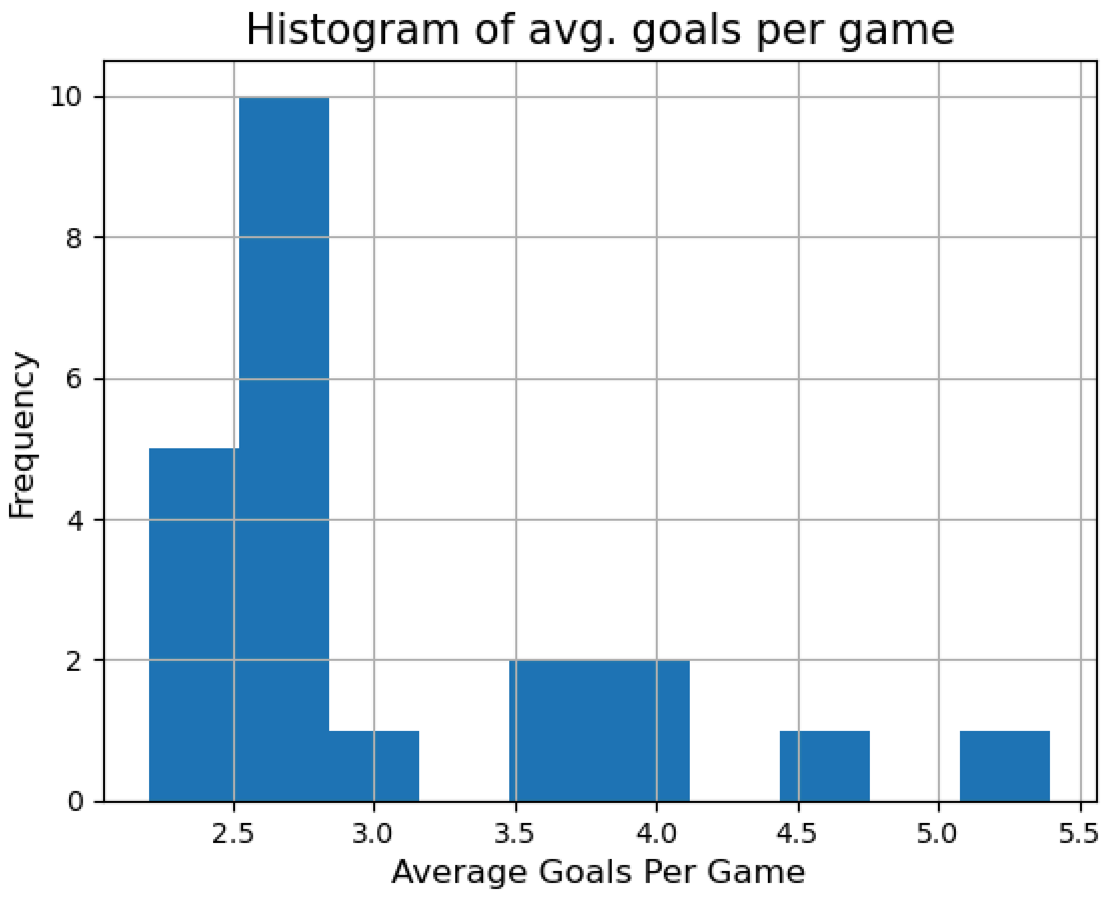
Out[90]:

```
1.9611615006183198
```

Histogram using Matplotlib

In [92]:

```
plt.hist(goals_per_game)
plt.title("Histogram of avg. goals per game", fontsize=15)
plt.xlabel("Average Goals Per Game", fontsize=12)
plt.ylabel("Frequency", fontsize=12)
plt.grid(True)
plt.show()
```



1.5 Conclusion

Hence, Anaconda Navigator and Jupyter Notebook, along with Python packages such as NumPy, Pandas, Matplotlib, and Seaborn, have been successfully downloaded and installed. Also, these data science packages were explored in Google Colab to perform descriptive statistical analysis.