MIT World Peace University

Data Science for Cybersecurity and Forensics
Third Year B. Tech, Semester 6

Statistical Approaches in Data Science

Assignment 3

Prepared By

Krishnaraj Thadesar
Cyber Security and Forensics
Batch A1, PA 10

April 16, 2024

Contents

1 Aim

Learning some statistical approaches that are used in data science.

2 Objectives

1. Write a Python program to implement central tendency for housing data.
2. Using python compute variance in the weather.
3. Compute variance in the weather to find best time to visit New Delhi(or any city).
4. Using histogram find the best time to visit Delhi (or any)s on any dataset.

3 Theory

3.1 Types of Statistics

Statistics can be broadly categorized into two main types: descriptive statistics and inferential statistics.

3.2 Descriptive Statistics

Descriptive statistics involve methods for summarizing and describing the features of a dataset. It provides insights into the central tendency, variability, and distribution of the data.

3.2.1 Measures of Central Tendency

Measures of central tendency are statistics that describe the center or average of a dataset. Common measures of central tendency include the mean, median, and mode.

- **Mean**: The arithmetic average of a set of values, calculated by summing all the values and dividing by the number of observations.

  Formula:
  $$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$$

- **Median**: The middle value in a dataset when the values are arranged in ascending order. It divides the dataset into two equal halves.

  Formula:
  $$\text{Median} = \begin{cases} x_{(n+1)/2} & \text{if } n \text{ is odd} \\ \frac{x_{n/2} + x_{n/2+1}}{2} & \text{if } n \text{ is even} \end{cases}$$

- **Mode**: The value that appears most frequently in a dataset.

  Formula:
  $$\text{Mode} = \text{value with highest frequency}$$

3.2.2 Measures of Dispersion

Measures of dispersion quantify the spread or variability of the data points in a dataset. They provide information about how the data is distributed around the central tendency.

- **Range**: The difference between the maximum and minimum values in a dataset.

  Formula:

  $$\text{Range} = \text{Maximum value} - \text{Minimum value}$$

- **Variance**: The average of the squared differences from the mean. It measures the average distance of each data point from the mean.

  Formula:

  $$\text{Variance} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n}$$

- **Standard Deviation**: The square root of the variance. It provides a measure of the dispersion of data points around the mean.

  Formula:

  $$\text{Standard Deviation} = \sqrt{\text{Variance}}$$

3.3 Inferential Statistics

Inferential statistics involve methods for making predictions or inferences about a population based on a sample of data. It uses probability theory to draw conclusions about the population parameters.

3.3.1 Hypothesis Testing

Hypothesis testing is a statistical method used to determine whether there is enough evidence to reject a null hypothesis in favor of an alternative hypothesis. It involves setting up a null hypothesis and an alternative hypothesis, collecting data, and using statistical tests to make a decision.

3.3.2 Regression Analysis

Regression analysis is a statistical technique used to model the relationship between a dependent variable and one or more independent variables. It helps in understanding how the value of the dependent variable changes when one or more independent variables are varied.

3.3.3 Correlation Analysis

Correlation analysis is a statistical method used to measure the strength and direction of the relationship between two variables. It helps in understanding how changes in one variable are associated with changes in another variable.

4 Platform

Operating System: Windows 11
IDEs or Text Editors Used: Visual Studio Code
Compilers or Interpreters: Python 3.10.1

5 Requirements

```
python==3.10.1
matplotlib==3.8.3
numpy==1.26.4
pandas==2.2.2
seaborn==0.13.2
```

6 Code

[26]:
```
# import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

sns.set_theme(style="darkgrid")
```

[7]:
```
# load dataset
df = pd.read_csv("data.csv")
df.head()
```

[7]:

	Unnamed: 0	Zone	State	City	Name	Type	Establishment Year	time needed to visit in hrs	Google review rating	Entrance Fee in INR	Airport with 50km Radius	Weekly Off	Significance	DSLR Allowed	Number of google review in lakhs	Best Time to visit
0	0	Northern	Delhi	Delhi	India Gate	War Memorial	1921	0.5	4.6	0	Yes	NaN	Historical	Yes	2.60	Evening
1	1	Northern	Delhi	Delhi	Humayun's Tomb	Tomb	1572	2.0	4.5	30	Yes	NaN	Historical	Yes	0.40	Afternoon
2	2	Northern	Delhi	Delhi	Akshardham Temple	Temple	2005	5.0	4.6	60	Yes	NaN	Religious	No	0.40	Afternoon
3	3	Northern	Delhi	Delhi	Waste to Wonder Park	Theme Park	2019	2.0	4.1	50	Yes	Monday	Environmental	Yes	0.27	Evening
4	4	Northern	Delhi	Delhi	Jantar Mantar	Observatory	1724	2.0	4.2	15	Yes	NaN	Scientific			

```
4          Yes                                0.31            Morning
```

6.1 Pre Processing

[8]:
```
# lets remove the unnamed column
df = df.drop("Unnamed: 0", axis=1)
df.columns
```

[8]:
```
Index(['Zone', 'State', 'City', 'Name', 'Type', 'Establishment Year',
       'time needed to visit in hrs', 'Google review rating',
       'Entrance Fee in INR', 'Airport with 50km Radius', 'Weekly Off',
       'Significance', 'DSLR Allowed', 'Number of google review in lakhs',
       'Best Time to visit'],
      dtype='object')
```

[19]:
```
# lets remove null values
print(df.isnull().sum())
```

```
Zone                                0
State                               0
City                                0
Name                                0
Type                                0
Establishment Year                  0
time needed to visit in hrs         0
Google review rating                0
Entrance Fee in INR                 0
Airport with 50km Radius            0
Weekly Off                        293
Significance                        0
DSLR Allowed                        0
Number of google review in lakhs    0
Best Time to visit                  0
dtype: int64
```

[30]:
```
# lets find outliers for ratings column, using z score
from scipy import stats

z = np.abs(stats.zscore(df["Google review rating"]))

# lets remove outliers
df = df[(z < 3)]
df.shape
```

[30]:
```
(324, 14)
```

[20]:
```
# given weekly off is mostly empty, lets remove it
df = df.drop("Weekly Off", axis=1)
```

```
df.head()
```

[20]:

	Zone	State	City	Name	Type	Establishment Year	time needed to visit in hrs	Google review rating	Entrance Fee in INR	Airport with 50km Radius	Significance	DSLR Allowed	Number of google review in lakhs	Best Time to visit
0	Northern	Delhi	Delhi	India Gate	War Memorial	1921	0.5	4.6	0	Yes	Historical	Yes	2.60	Evening
1	Northern	Delhi	Delhi	Humayun's Tomb	Tomb	1572	2.0	4.5	30	Yes	Historical	Yes	0.40	Afternoon
2	Northern	Delhi	Delhi	Akshardham Temple	Temple	2005	5.0	4.6	60	Yes	Religious	No	0.40	Afternoon
3	Northern	Delhi	Delhi	Waste to Wonder Park	Theme Park	2019	2.0	4.1	50	Yes	Environmental	Yes	0.27	Evening
4	Northern	Delhi	Delhi	Jantar Mantar	Observatory	1724	2.0	4.2	15	Yes	Scientific	Yes	0.31	Morning

6.2 EDA

[31]:

```
# lets see the data types of the columns
df.dtypes
```

[31]:

```
Zone                            object
State                           object
City                            object
Name                            object
Type                            object
Establishment Year              object
time needed to visit in hrs    float64
Google review rating           float64
Entrance Fee in INR              int64
Airport with 50km Radius        object
Significance                    object
```

```
DSLR Allowed                          object
Number of google review in lakhs     float64
Best Time to visit                    object
dtype: object
```

[32]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 324 entries, 0 to 324
Data columns (total 14 columns):
 #   Column                            Non-Null Count  Dtype
---  ------                            --------------  -----
 0   Zone                              324 non-null    object
 1   State                             324 non-null    object
 2   City                              324 non-null    object
 3   Name                              324 non-null    object
 4   Type                              324 non-null    object
 5   Establishment Year                324 non-null    object
 6   time needed to visit in hrs       324 non-null    float64
 7   Google review rating              324 non-null    float64
 8   Entrance Fee in INR               324 non-null    int64
 9   Airport with 50km Radius          324 non-null    object
 10  Significance                      324 non-null    object
 11  DSLR Allowed                      324 non-null    object
 12  Number of google review in lakhs  324 non-null    float64
 13  Best Time to visit                324 non-null    object
dtypes: float64(3), int64(1), object(10)
memory usage: 38.0+ KB
```

[33]:
```
df.describe()
```

[33]:
```
       time needed to visit in hrs  Google review rating  Entrance Fee in␣
↪INR  \
count                   324.000000            324.000000           324.000000
mean                      1.797840              4.495679           112.620370
std                       0.956497              0.214591           528.554154
min                       0.500000              3.700000             0.000000
25%                       1.000000              4.400000             0.000000
50%                       1.500000              4.500000             0.000000
75%                       2.000000              4.600000            36.250000
max                       7.000000              4.900000          7500.000000

       Number of google review in lakhs
count                        324.000000
mean                           0.406767
std                            0.646965
min                            0.010000
25%                            0.059000
```

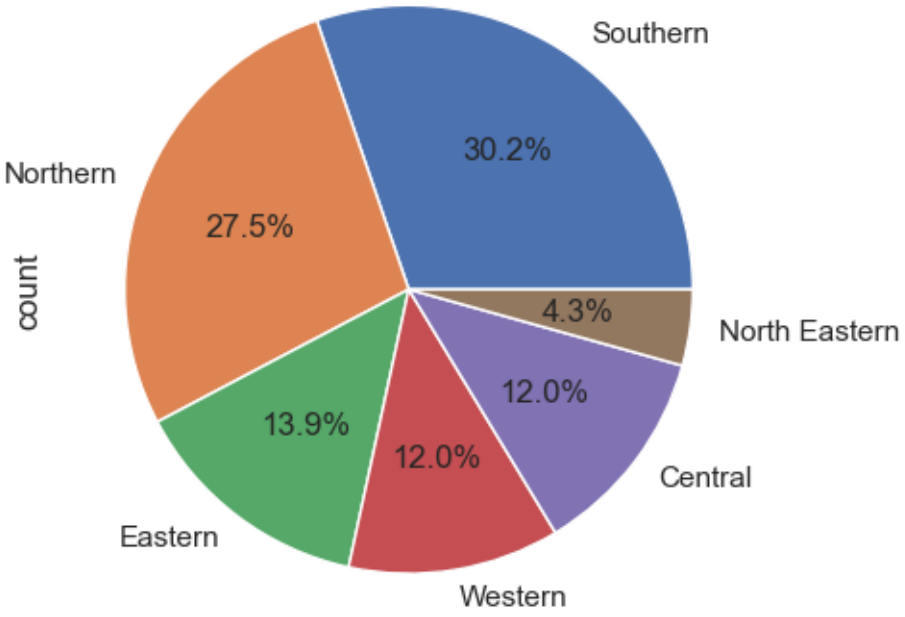
```
50%                          0.165000
75%                          0.492500
max                          7.400000
```

[34]:
```
# lets see the highest rated places to visit, sorting by Google Review Rating
df.sort_values("Google review rating", ascending=False).head()
```

[34]:
```
            Zone           State           City                                ␣
↪Name  \
92    Northern          Punjab       Amritsar  Golden Temple (Harmandir Sahib)
196   Northern          Ladakh            Leh                      Pangong Tso
72     Western         Gujarat  Rann of Kutch                       Rann Utsav
71     Western         Gujarat        Somnath                   Somnath Temple
145    Central  Madhya Pradesh         Orchha                      Orchha Fort

                Type Establishment Year  time needed to visit in hrs  \
92    Religious Site               1604                          1.5
196             Lake            Unknown                          2.0
72          Cultural            Unknown                          3.0
71            Temple               1951                          2.0
145             Fort               1500                          1.5

      Google review rating  Entrance Fee in INR Airport with 50km Radius  \
92                     4.9                    0                       Yes
196                    4.9                   20                       Yes
72                     4.9                 7500                       Yes
71                     4.8                    0                        No
145                    4.8                   10                        No

     Significance DSLR Allowed  Number of google review in lakhs  \
92      Spiritual          Yes                              1.90
196        Nature          Yes                              0.15
72       Cultural          Yes                              0.10
71      Religious           No                              0.39
145    Historical          Yes                              0.10

     Best Time to visit
92                  All
196             Morning
72              Evening
71              Morning
145           Afternoon
```

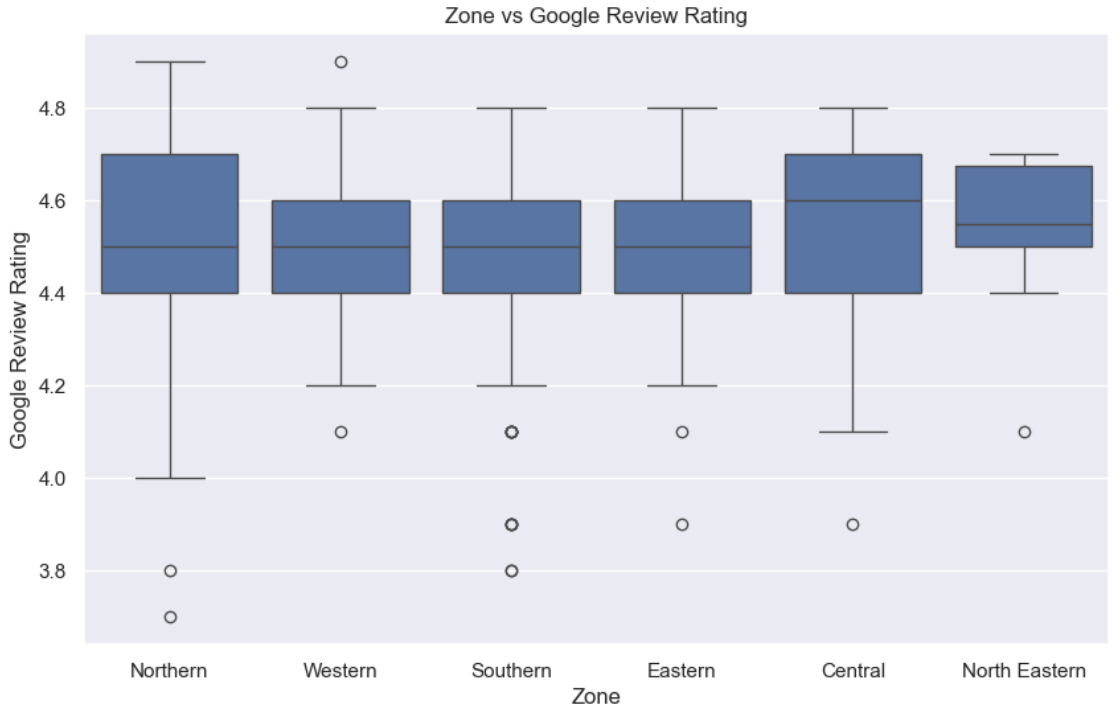
[39]:
```
# lets see which zone of India is most rated, being categorical, lets make a␣
↪pie chart to see which has the most number of ratings
df["Zone"].value_counts().plot.pie(autopct="%1.1f%%")
```

[39]:
```
<Axes: ylabel='count'>
```



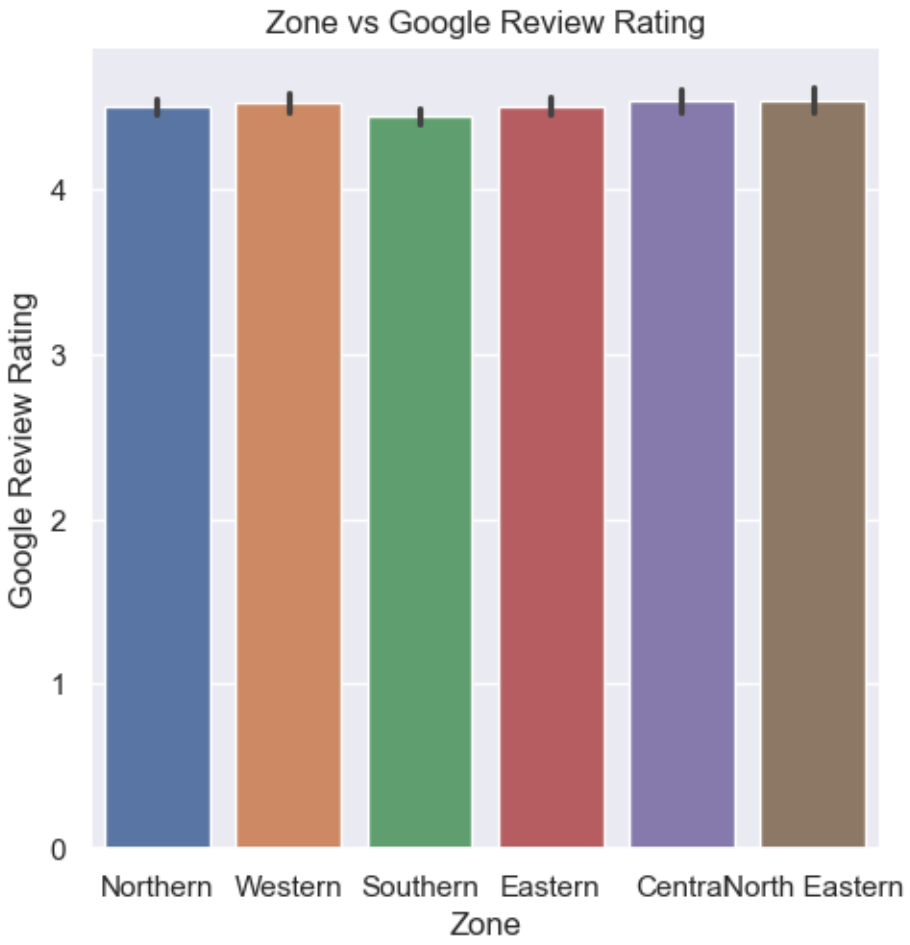
[46]:
```
# lets plot zone vs rating
plt.figure(figsize=(10, 6))
sns.boxplot(x="Zone", y="Google review rating", data=df, ax=plt.gca())
# title and labels
plt.title("Zone vs Google Review Rating")
plt.xlabel("Zone")
plt.ylabel("Google Review Rating")
```

[46]:
```
Text(0, 0.5, 'Google Review Rating')
```



[47]:
```
    sns.catplot(data=df, x="Zone", y="Google review rating", kind="bar",␣
↪hue="Zone")
    # title and labels
    plt.title("Zone vs Google Review Rating")
    plt.xlabel("Zone")
    plt.ylabel("Google Review Rating")
```

[47]:
```
    Text(25.319444444444443, 0.5, 'Google Review Rating')
```



so we see that most of the zones are almost equally rated and there isnt much difference there.

we can then perform a hypothesis test to see if the ratings are significantly different or not

[50]:
```
# null hypothesis: the ratings are not significantly different
# alternate hypothesis: the ratings are significantly different

from scipy.stats import f_oneway  # anova test

# lets get the ratings for each zone
north = df[df["Zone"] == "Northern"]["Google review rating"]
south = df[df["Zone"] == "Southern"]["Google review rating"]
east = df[df["Zone"] == "Eastern"]["Google review rating"]
west = df[df["Zone"] == "Western"]["Google review rating"]
central = df[df["Zone"] == "Central"]["Google review rating"]
north_east = df[df["Zone"] == "North Eastern"]["Google review rating"]

# lets perform the test
```

```
    f_oneway(north, south, east, west, central, north_east)
```

[50]:
```
    F_onewayResult(statistic=1.598487683716663, pvalue=0.16009417988080096)
```

since p value is more than 0.05, we fail to reject the null hypothesis, which means the ratings are not significantly different

[52]:
```
    df.head()
```

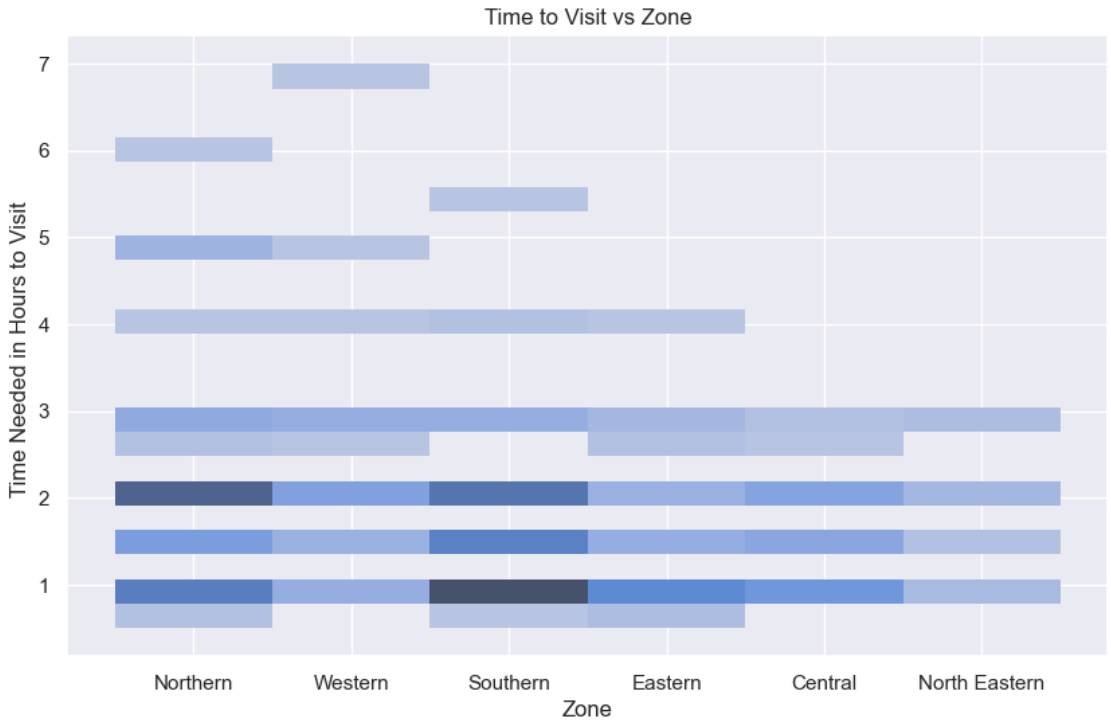
[52]:

	Zone	State	City	Name	Type	Establishment Year	time needed to visit in hrs	Google review rating	Entrance Fee in INR	Airport with 50km Radius	Significance	DSLR Allowed	Number of google review in lakhs	Best Time to visit
0	Northern	Delhi	Delhi	India Gate	War Memorial	1921	0.5	4.6	0	Yes	Historical	Yes	2.60	Evening
1	Northern	Delhi	Delhi	Humayun's Tomb	Tomb	1572	2.0	4.5	30	Yes	Historical	Yes	0.40	Afternoon
2	Northern	Delhi	Delhi	Akshardham Temple	Temple	2005	5.0	4.6	60	Yes	Religious	No	0.40	Afternoon
3	Northern	Delhi	Delhi	Waste to Wonder Park	Theme Park	2019	2.0	4.1	50	Yes	Environmental	Yes	0.27	Evening
4	Northern	Delhi	Delhi	Jantar Mantar	Observatory	1724	2.0	4.2	15	Yes	Scientific	Yes	0.31	Morning

[66]:
```
    # lets see what places take the most time to visit
    df.sort_values("time needed to visit in hrs", ascending=False).head()

    # plottin a simple hist plot to see the relationship between time to visit␣
↪and google review rating
    plt.figure(figsize=(10, 6))
    sns.histplot(y="time needed to visit in hrs", x="Zone", data=df, ax=plt.
↪gca())
    # title and labels
    plt.title("Time to Visit vs Zone")
```

```
    plt.xlabel("Zone")
    plt.ylabel("Time Needed in Hours to Visit ")
```

[66]:
```
    Text(0, 0.5, 'Time Needed in Hours to Visit ')
```

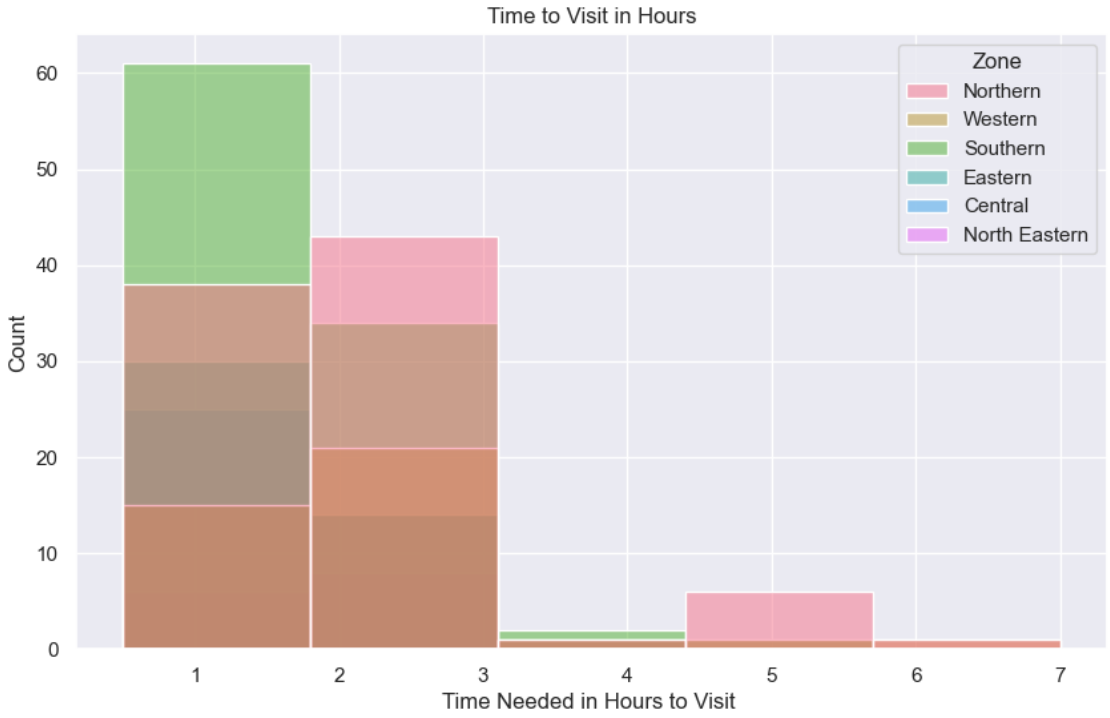


above graphs shows us that north and south often take around 2 hours, south takes the least time, while eastern, celtral and north eastern can be visited mostly in half to 3 hours.

[77]:
```
    # lets see this better by binning the time needed to visit between 0 and 1,␣
↪1 and 2, 2 and 3, 3 and 4, 4 and 5
    df["time needed to visit in hrs"].value_counts(bins=5)
    # lets plot
    plt.figure(figsize=(10, 6))
    sns.histplot(
        x="time needed to visit in hrs",
        data=df,
        bins=5,
        ax=plt.gca(),
        hue="Zone",
        palette=sns.color_palette("husl", 6),
    )
    # title and labels
    plt.title("Time to Visit in Hours")
    plt.xlabel("Time Needed in Hours to Visit ")
```

```
    plt.ylabel("Count")
```

[77]:
```
    Text(0, 0.5, 'Count')
```



this shows us that most places can be visted in half to 2 hours, while only some may take more. Places in south take the least time

[78]:
```
    df.head()
```

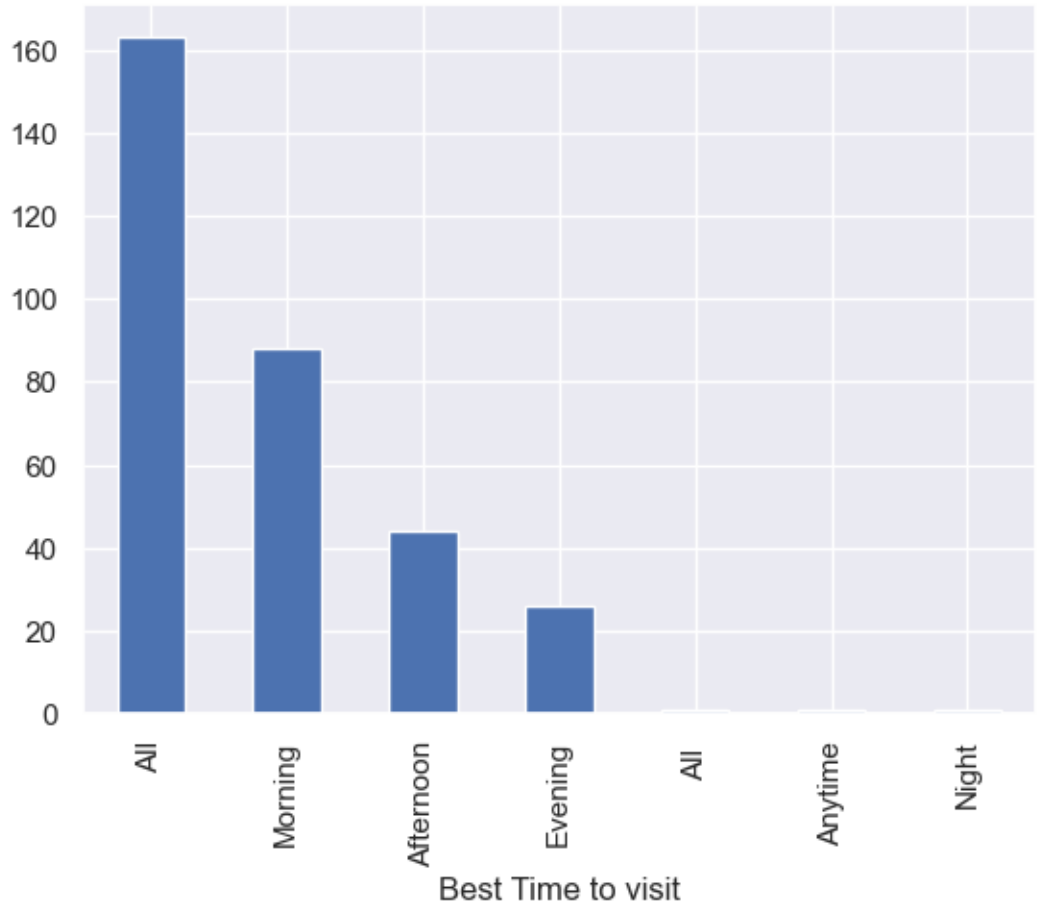
[78]:

	Zone	State	City	Name	Type	Establishment Year	time needed to visit in hrs	Google review rating
0	Northern	Delhi	Delhi	India Gate	War Memorial	1921	0.5	4.6
1	Northern	Delhi	Delhi	Humayun's Tomb	Tomb	1572	2.0	4.5
2	Northern	Delhi	Delhi	Akshardham Temple	Temple	2005	5.0	4.6
3	Northern	Delhi	Delhi	Waste to Wonder Park	Theme Park	2019	2.0	4.1
4	Northern	Delhi	Delhi	Jantar Mantar	Observatory	1724	2.0	4.2

```
    Entrance Fee in INR Airport with 50km Radius   Significance DSLR Allowed ␣
↪\
0                     0                      Yes     Historical          Yes
1                    30                      Yes     Historical          Yes
2                    60                      Yes      Religious           No
3                    50                      Yes  Environmental          Yes
4                    15                      Yes     Scientific          Yes

   Number of google review in lakhs Best Time to visit
0                              2.60            Evening
1                              0.40          Afternoon
2                              0.40          Afternoon
3                              0.27            Evening
4                              0.31            Morning
```

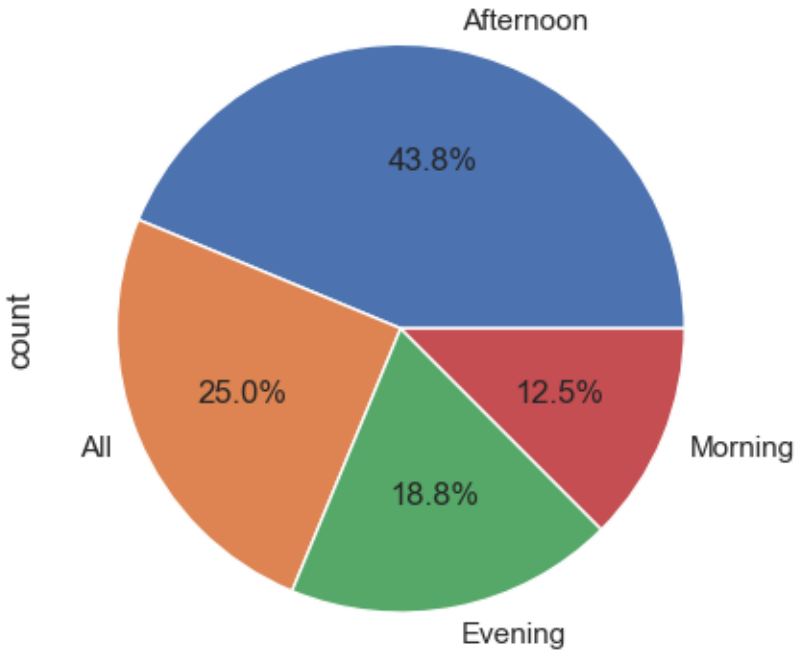
[82]:
```
# lets see what the best time to visit is with a bar chart
df["Best Time to visit"].value_counts().plot.bar()
# df["Best Time to visit"].value_counts().plot.pie(autopct="%1.1f%%")
```

[82]:
```
<Axes: xlabel='Best Time to visit'>
```



[84]:
```
# lets see what the best time to visit places in delhi are
df[df["City"] == "Delhi"]["Best Time to visit"].value_counts().plot.pie(
    autopct="%1.1f%%"
)
```

[84]:
```
<Axes: ylabel='count'>
```



[85]:
```
df.head()
```

[85]:

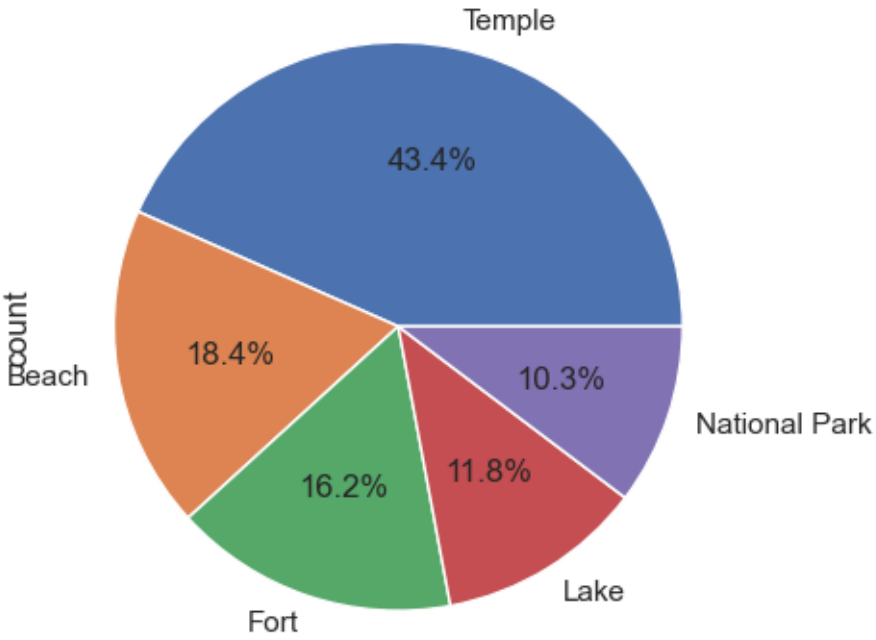
	Zone	State	City	Name	Type	Establishment Year	time needed to visit in hrs	Google review rating
0	Northern	Delhi	Delhi	India Gate	War Memorial	1921	0.5	4.6
1	Northern	Delhi	Delhi	Humayun's Tomb	Tomb	1572	2.0	4.5
2	Northern	Delhi	Delhi	Akshardham Temple	Temple	2005	5.0	4.6
3	Northern	Delhi	Delhi	Waste to Wonder Park	Theme Park	2019	2.0	4.1
4	Northern	Delhi	Delhi	Jantar Mantar	Observatory			

```
4                1724                          2.0                    4.2

   Entrance Fee in INR Airport with 50km Radius   Significance DSLR Allowed ␣
↪\
0                     0                      Yes     Historical          Yes
1                    30                      Yes     Historical          Yes
2                    60                      Yes      Religious           No
3                    50                      Yes  Environmental          Yes
4                    15                      Yes     Scientific          Yes

   Number of google review in lakhs Best Time to visit
0                              2.60            Evening
1                              0.40          Afternoon
2                              0.40          Afternoon
3                              0.27            Evening
4                              0.31            Morning
```

[103]:
```
top_5_types = df["Type"].value_counts().head()
# lets filter df by these
df_top_5 = df[df["Type"].isin(top_5_types.index)]
```

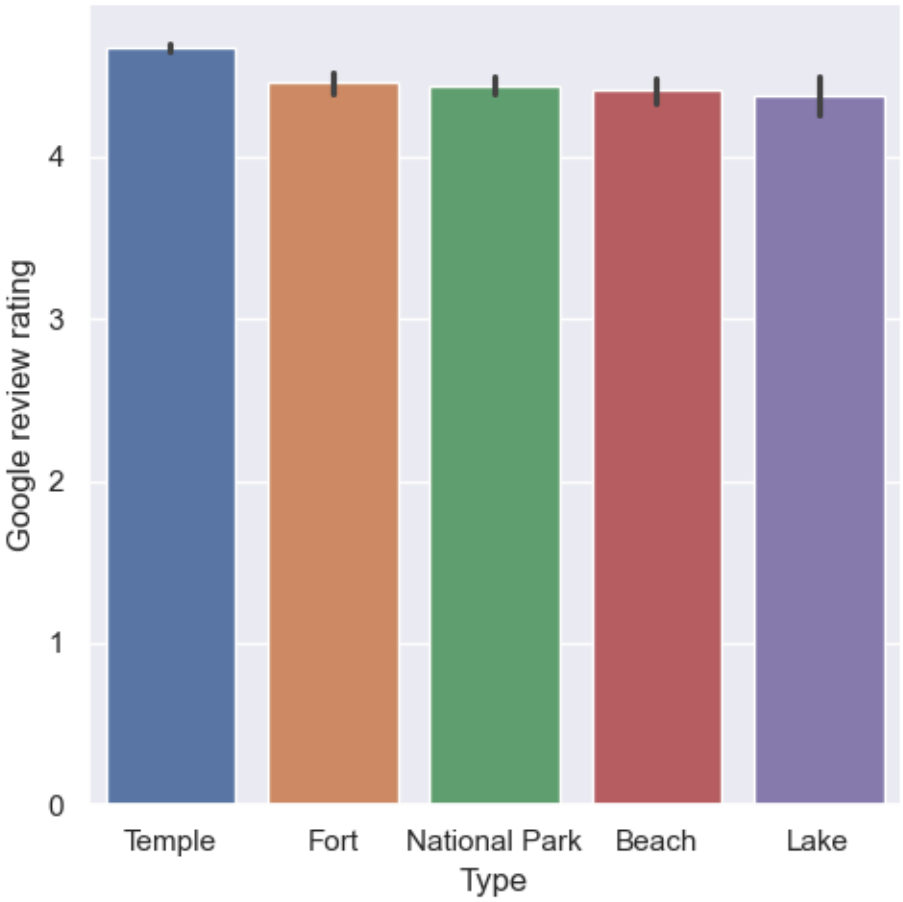
[104]:
```
# lets see the top 5 kinds of places by making a pie chart
df_top_5["Type"].value_counts().plot.pie(autopct="%1.1f%%")
```

[104]:
```
<Axes: ylabel='count'>
```



[105]:
```
    # lets now see which are the most highly rated
    sns.catplot(data=df_top_5, x="Type", y="Google review rating", kind="bar",␣
↪hue="Type")
```

[105]:
```
    <seaborn.axisgrid.FacetGrid at 0x1d85df57100>
```



[ ]:

7 FAQs

7.1 Question 1

1. **What do you understand by Statistics for Data science?**
   Statistics for data science involves the application of statistical methods and techniques to analyze, interpret, and derive insights from data. It encompasses a wide range of methods, including descriptive statistics, inferential statistics, and predictive modeling, to explore patterns, relationships, and trends within datasets. In data science, statistics plays a crucial role in data preprocessing, exploratory data analysis, hypothesis testing, and model evaluation, enabling data scientists to make informed decisions and derive actionable insights from data.

7.2 Question 2

1. **Do we need preprocessing to perform statistics for Data science? Justify, your answer**
   Yes, preprocessing is essential for performing statistics in data science. Preprocessing involves cleaning, transforming, and preparing raw data to make it suitable for statistical analysis. Without preprocessing, raw data may contain missing values, outliers, inconsistencies, or other irregularities that can affect the accuracy and reliability of statistical results. Preprocessing techniques such as handling missing values, outlier detection, data normalization, and feature engineering help ensure that the data meets the assumptions and requirements of statistical methods. By preprocessing the data, data scientists can improve the quality of statistical analysis, enhance the performance of models, and derive more accurate and meaningful insights from the data.

7.3 Question 3

1. **Describe the different Statistical approaches in Data science using Python?**
   In data science, various statistical approaches are used to analyze and model data using Python. Some common statistical approaches include:

   - Descriptive Statistics: Summarizing and describing the features of a dataset using measures of central tendency, dispersion, and visualization techniques.
   - Inferential Statistics: Making inferences and predictions about populations based on sample data using hypothesis testing, confidence intervals, and regression analysis.
   - Predictive Modeling: Building predictive models to forecast future outcomes or classify data into different categories using techniques such as linear regression, logistic regression, decision trees, and ensemble methods.
   - Time Series Analysis: Analyzing and forecasting time-series data using methods like autoregressive integrated moving average (ARIMA), seasonal decomposition, and exponential smoothing.

   Python provides a wide range of libraries and tools for implementing these statistical approaches, including NumPy, pandas, SciPy, scikit-learn, and Statsmodels, making it a popular choice for statistical analysis in data science.

8 Conclusion

In this assignment, we learned about various statistical approaches used in data science, including measures of central tendency, dispersion, hypothesis testing, regression analysis, and correlation analysis. We implemented these statistical concepts using Python and explored how they can be applied to analyze and interpret data. By understanding and applying statistical methods, data scientists can gain valuable insights from data, make informed decisions, and build predictive models to solve real-world problems.