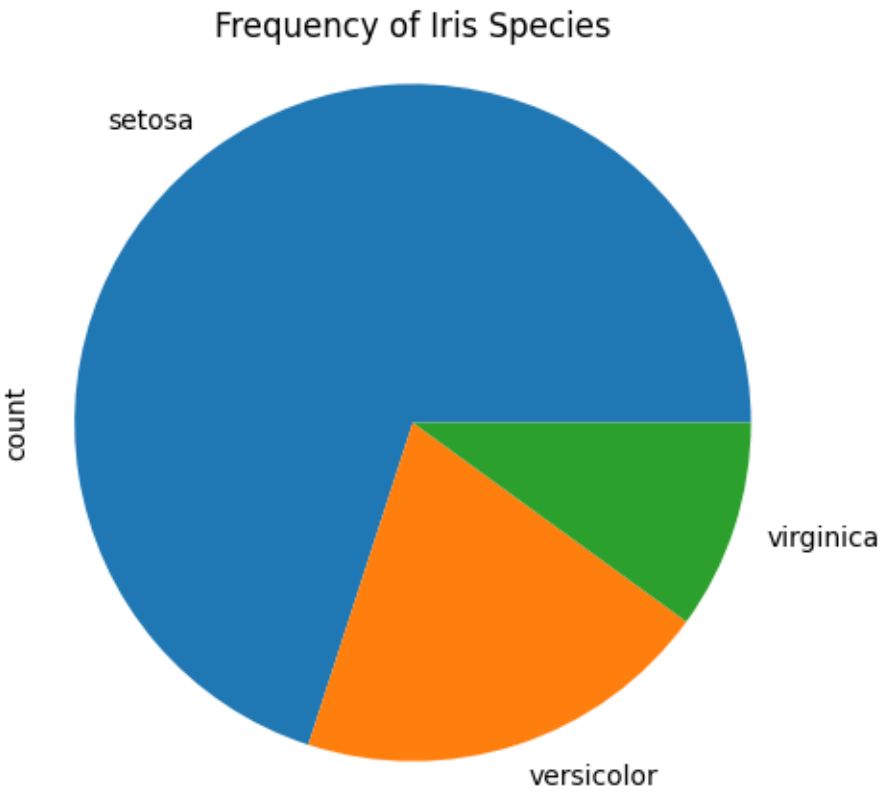
slip1

October 15, 2024

[4]:
```
#Write a Python program to create a Pie plot to get the frequency
#of the three species of the Iris data (Use 1ris.csv)

# (slip1 and slip11 are same)

import pandas as pd
import matplotlib.pyplot as plt

d = pd.read_csv('Iris.csv')

ax = plt.figure(figsize=(5, 5))
d['Species'].value_counts().plot.pie()

plt.title('Frequency of Iris Species')
plt.axis('equal')
plt.show()
```



[3]:
```
# write a Python program to view basic statistical details of the data
#.(Use wineequality-red.csv)
import pandas as pd

wine_data = pd.read_csv('winequality-red.csv')
print(wine_data.describe())
```

```
       fixed acidity  volatile acidity  citric acid  residual sugar  \
count       5.000000          5.000000     5.000000        5.000000
mean        8.420000          0.492000     0.128000        1.940000
std         1.565886          0.350885     0.242322        0.427785
min         7.400000          0.000000     0.000000        1.400000
25%         7.800000          0.280000     0.000000        1.900000
50%         7.800000          0.600000     0.040000        1.900000
75%         7.900000          0.700000     0.040000        1.900000
max        11.200000          0.880000     0.560000        2.600000

       chlorides  free sulfur dioxide  total sulfur dioxide   density  \
count   5.000000              5.00000               5.00000  5.000000
mean    0.074800             17.00000              55.00000  0.997280
std     0.003633              5.09902              12.60952  0.000576
min     0.069000             11.00000              34.00000  0.996800
```

```
25%     0.075000             15.00000              54.00000  0.996800
50%     0.075000             17.00000              60.00000  0.997000
75%     0.076000             17.00000              60.00000  0.997800
max     0.079000             25.00000              67.00000  0.998000
```

```
             pH  sulphates   alcohol   quality
count  5.000000     5.0000  5.000000  5.000000
mean   3.258000     0.6220  9.720000  5.200000
std    0.146697     0.0502  0.178885  0.447214
min    3.160000     0.5600  9.400000  5.000000
25%    3.160000     0.5800  9.800000  5.000000
50%    3.200000     0.6400  9.800000  5.000000
75%    3.260000     0.6500  9.800000  5.000000
max    3.510000     0.6800  9.800000  6.000000
```

[ ]:

slip2

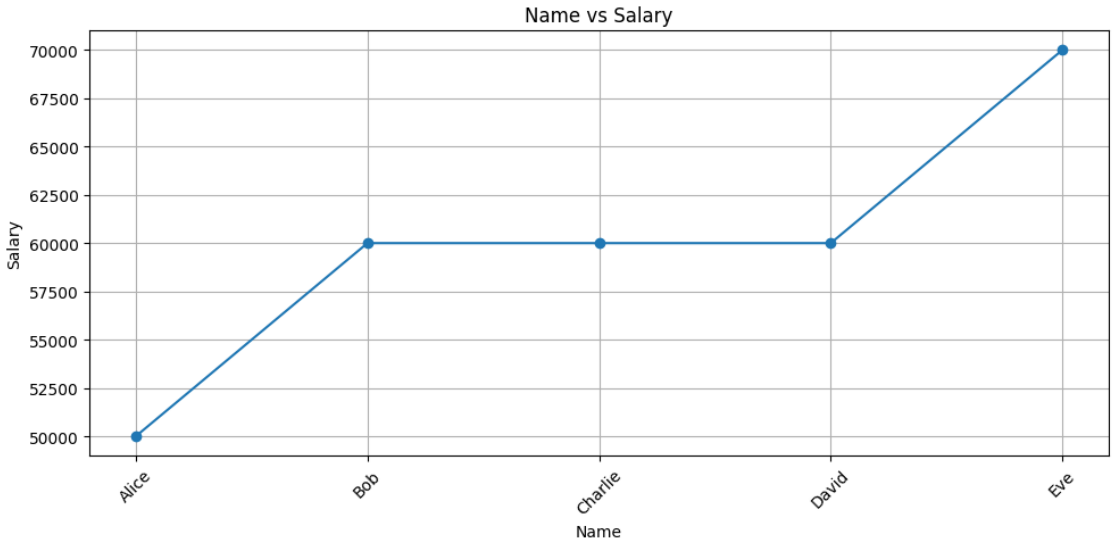
October 15, 2024

[12]:
```
# write a Python program for Handling Missing Value. Replace missing
# value of salary, age column with mean of that column.
# (Use Data.csv file)
# slip 2 and slip6 are same

import pandas as pd

df = pd.read_csv('Data.csv')
print("Original DataFrame:")
print(df)

df['salary'] = df['salary'].fillna(df['salary'].mean())
df['age'] = df['age'].fillna(df['age'].mean())

print("\nDataFrame after handling missing values:")
print(df)
```

```
Original DataFrame:
      name   salary   age
0    Alice  50000.0  30.0
1      Bob      NaN  25.0
2  Charlie  60000.0   NaN
3    David      NaN  35.0
4      Eve  70000.0  28.0

DataFrame after handling missing values:
      name   salary   age
0    Alice  50000.0  30.0
1      Bob  60000.0  25.0
2  Charlie  60000.0  29.5
3    David  60000.0  35.0
4      Eve  70000.0  28.0
```

[4]:
```
#  Write a Python program to generate a line plot of name Vs salary
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('Data_filled.csv')
```

```
# Generate a line plot for name vs salary
plt.figure(figsize=(10, 5))
plt.plot(df['name'], df['salary'], marker='o')
plt.title('Name vs Salary')
plt.xlabel('Name')
plt.ylabel('Salary')
plt.xticks(rotation=45)
plt.grid()
plt.tight_layout()
plt.show()
```



[10]:
```
# Download the heights and weights dataset and load the dataset from
# a given csv file into a dataframe. Print the first, last 10 rows
# and random 20 rows also display shape of the dataset.
import pandas as pd

# Load the dataset from the CSV file
df = pd.read_csv('short_heights_weights.csv')

# Print the first 10 rows
print("First 10 rows:")
print(df.head(10))

# Print the last 10 rows
print("\nLast 10 rows:")
print(df.tail(10))
```

```python
# Print 20 random rows
print("\nRandom 20 rows:")
print(df.sample(20))

# Display the shape of the dataset
print("\nShape of the dataset:", df.shape)
```

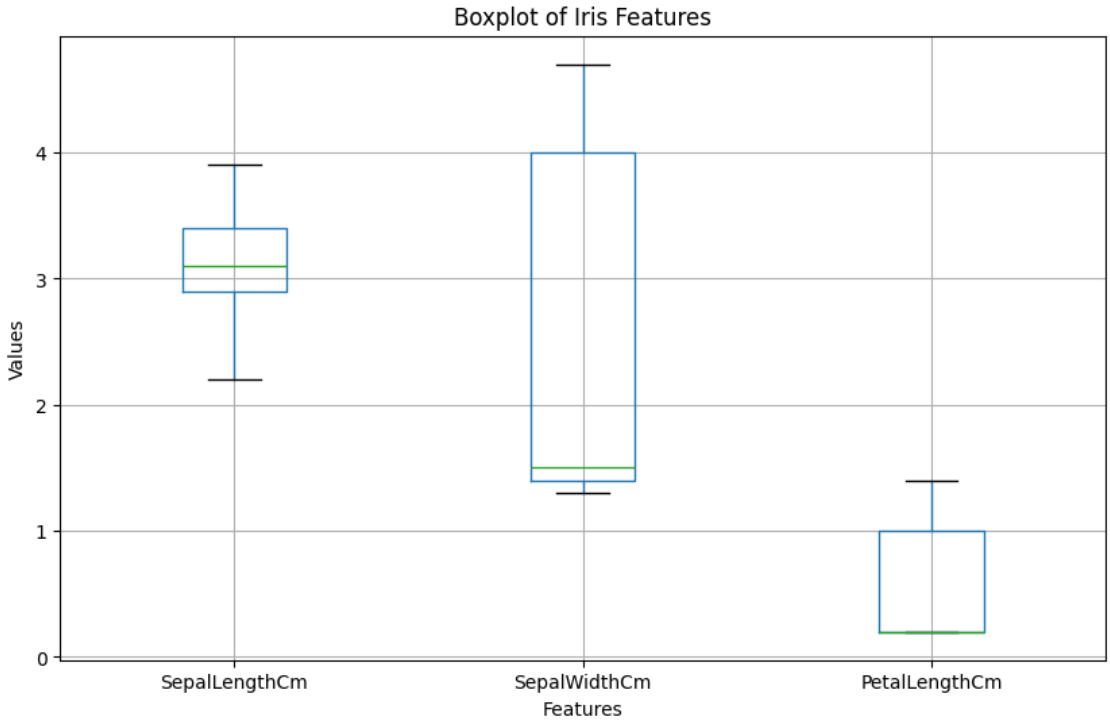
```
First 10 rows:
   Gender  Height(Inches)  Weight(Pounds)
0    Male              65             150
1  Female              62             120
2    Male              70             180
3  Female              58             110
4    Male              66             160
5  Female              64             130
6    Male              65             150
7  Female              62             120
8    Male              70             180
9  Female              58             110

Last 10 rows:
    Gender  Height(Inches)  Weight(Pounds)
33  Female              62             120
34    Male              70             180
35  Female              58             110
36    Male              66             160
37  Female              64             130
38    Male              65             150
39  Female              62             120
40    Male              70             180
41  Female              58             110
42    Male              66             160

Random 20 rows:
    Gender  Height(Inches)  Weight(Pounds)
3   Female              58             110
8     Male              70             180
1   Female              62             120
5   Female              64             130
9   Female              58             110
0     Male              65             150
19  Female              62             120
13  Female              62             120
37  Female              64             130
10    Male              66             160
23  Female              64             130
42    Male              66             160
```

```
33  Female              62              120
17  Female              64              130
31  Female              64              130
38    Male              65              150
20    Male              70              180
12    Male              65              150
28    Male              66              160
25  Female              62              120

Shape of the dataset: (43, 3)
```

slip3

October 15, 2024

[4]:
```
# Write a Python program to create box plots to see how each feature i.e. Sepal␣
 ↪Length,
# Sepal Width, Petal Length, Petal Width are distributed across the three␣
 ↪species. (Use iris.csv
# dataset)

import pandas as pd
import matplotlib.pyplot as plt

d = pd.read_csv('iris3.csv')
new_data = d[["SepalLengthCm", "SepalWidthCm", "PetalLengthCm", "PetalWidthCm"]]

plt.figure(figsize=(10, 6))
new_data.boxplot()
plt.title("Boxplot of Iris Features")
plt.xlabel("Features")
plt.ylabel("Values")
plt.show()
```



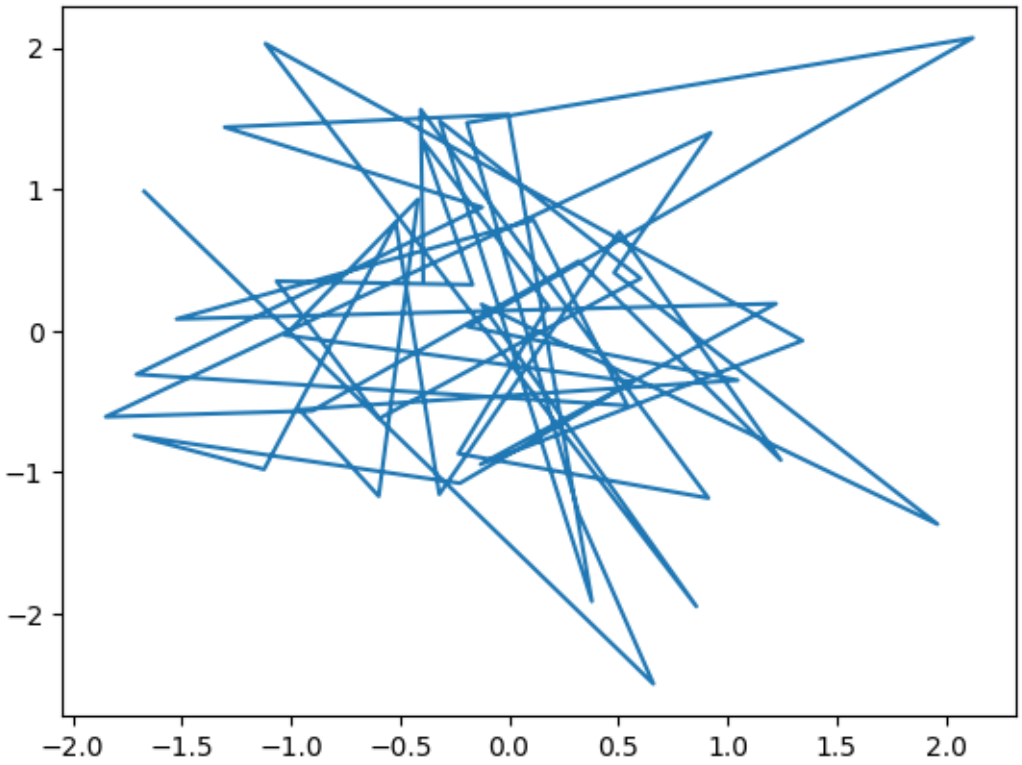
[5]:
```
#  Write a Python program to view basic statistical details of the data (Use
 ↪Heights and
# Weights Dataset
# (similer as slip1 of 2)

import pandas as pd

df = pd.read_csv('hw3.csv')
stats = df.describe()
print(stats)
```

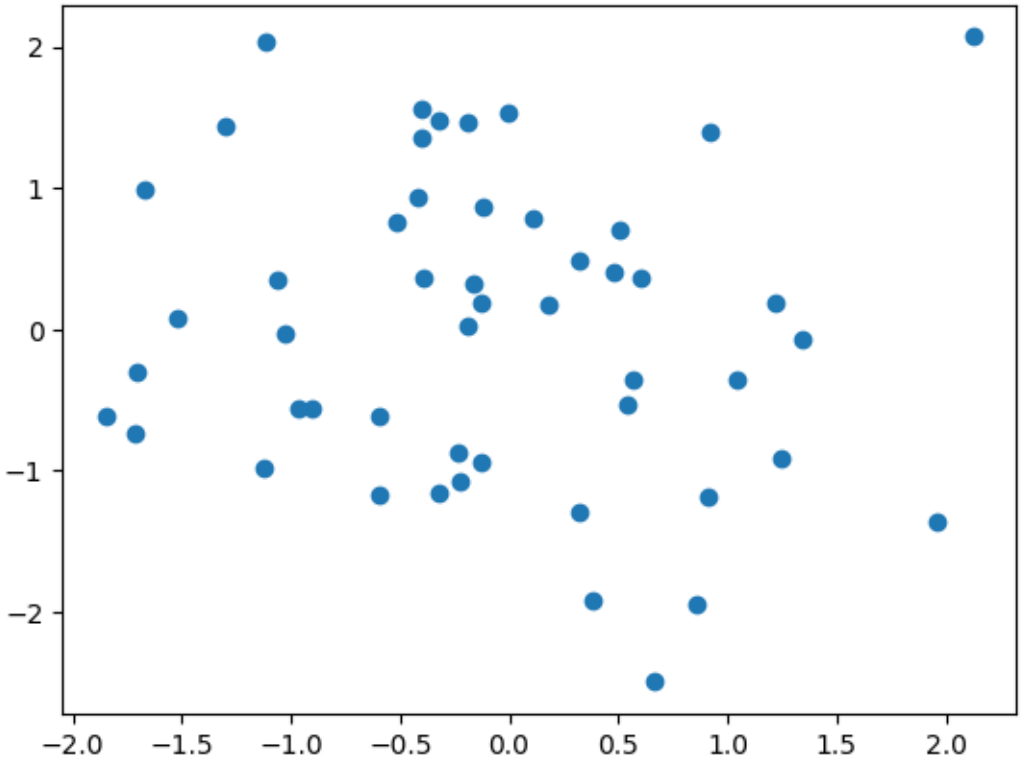
```
         HeightCm   WeightKg
count    5.000000   5.000000
mean   170.000000  70.000000
std     15.811388  15.811388
min    150.000000  50.000000
25%    160.000000  60.000000
50%    170.000000  70.000000
75%    180.000000  80.000000
max    190.000000  90.000000
```

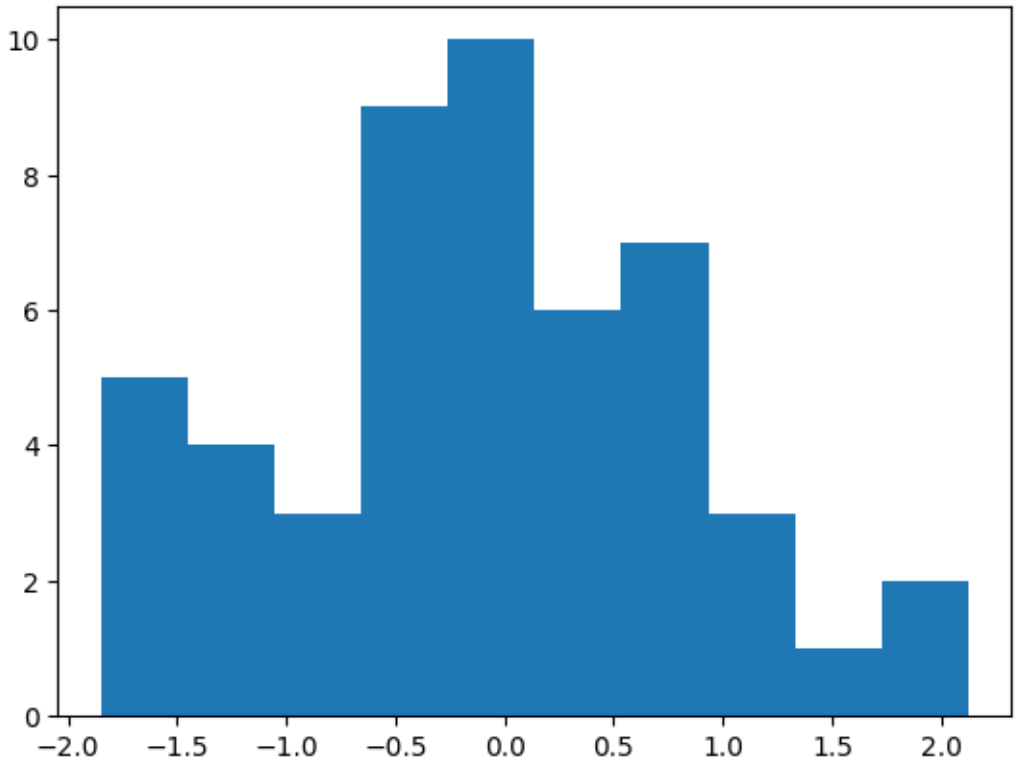
slip4

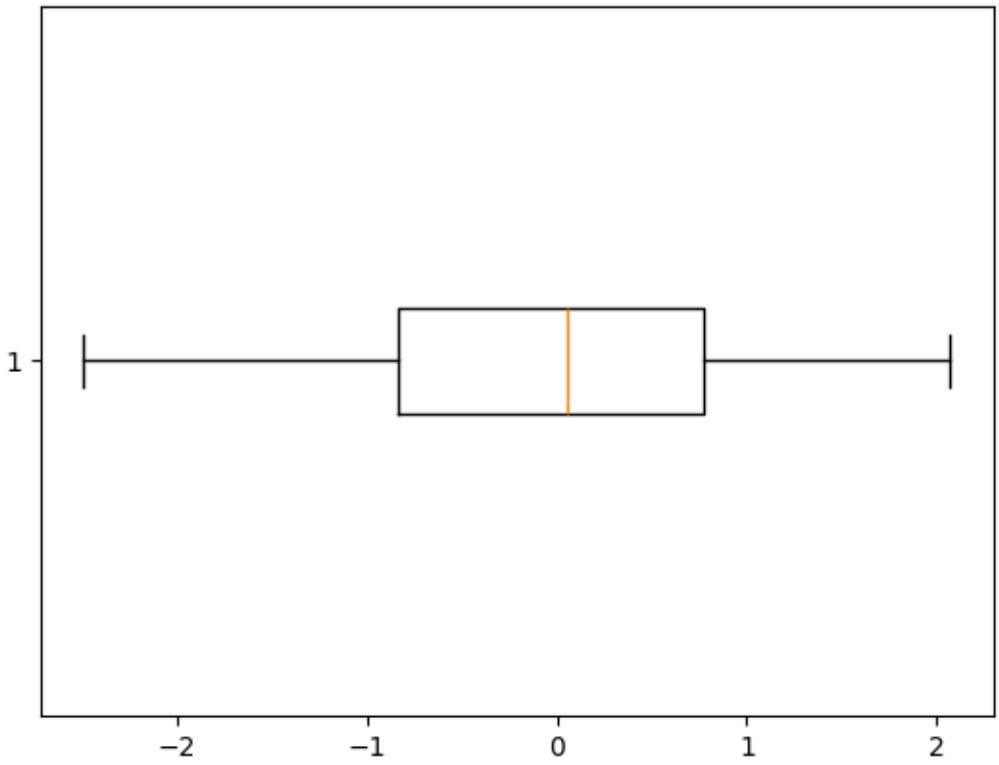
October 15, 2024

[1]:
```
# Generate a random array of 50 integers and display them using a line chart,␣
 ↪scatter plot,
# histogram and box plot. Apply appropriate color, labels and styling options

# slip4 and slip 5 is same

import matplotlib.pyplot as plt
import numpy as np

x=np.random.randn(50)
y=np.random.randn(50)

plt.plot(x,y)
plt.show()
plt.scatter(x,y)
plt.show()
plt.hist(x)
plt.show()
plt.boxplot(y,vert=False)
plt.show()
```


2
1
0
−1
−2
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0


2
1
0
−1
−2
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0


10
8
6
4
2
0
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0



[2]:
```
# Write a Python program to print the shape, number of rows-columns, data
 ↪types, feature
# names and the description of the data(Use User_Data.csv)
# (this program also in slip19 of 2 )

import pandas as pd
df = pd.read_csv('Data_filled.csv')
print("Shape of the dataset:", df.shape)
print("features :",df.info())
print("\nDescription of the Data:\n", df.describe())
print("\nData Types:\n", df.dtypes)
```

```
Shape of the dataset: (5, 3)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   name    5 non-null      object
 1   salary  5 non-null      float64
 2   age     5 non-null      float64
dtypes: float64(2), object(1)
```

```
memory usage: 252.0+ bytes
features : None

Description of the Data:
              salary        age
count      5.000000   5.000000
mean   60000.000000  29.500000
std     7071.067812   3.640055
min    50000.000000  25.000000
25%    60000.000000  28.000000
50%    60000.000000  29.500000
75%    60000.000000  30.000000
max    70000.000000  35.000000

Data Types:
 name       object
salary    float64
age       float64
dtype: object
```

slip7

October 15, 2024

[3]:
```
# Write a Python program to perform the following tasks :
# a. Apply OneHot coding on Country column.
# b. Apply Label encoding on purchased column
# (Data.csv have two categorical column the country column, and the purchased column).

import pandas as pd
from sklearn.preprocessing import LabelEncoder

d = pd.read_csv('s7.csv')
# Apply Label encoding on the 'Purchased' column
le = LabelEncoder()
d['Purchased'] = le.fit_transform(d['Purchased'])
# Apply OneHot encoding on the 'Country' column
ohed = pd.get_dummies(d, columns=['Country'])
print(ohed)
```

```
   Purchased  Country_Canada  Country_UK  Country_USA
0          1           False       False         True
1          0            True       False        False
2          1           False       False         True
3          0           False        True        False
```

slip8

October 15, 2024

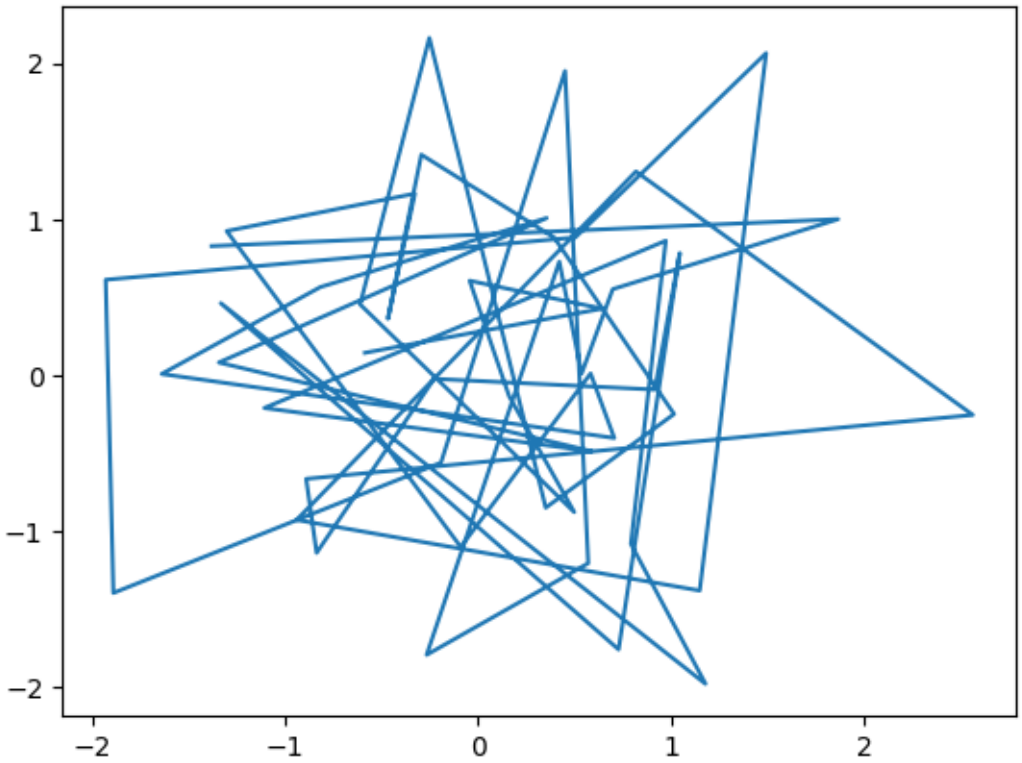
[1]:
```
# Write a program in python to perform following task : [15] Standardizing Data
 ↪(transform
# them into a standard Gaussian distribution with a mean of 0 and a standard
 ↪deviation of 1) (Use
# winequality-red.csv)

import pandas as pd
from sklearn.preprocessing import StandardScaler

df = pd.read_csv('winequality-red.csv')
num_cols = df.select_dtypes(include=['float64', 'int64']).columns

scaler = StandardScaler()
std_data = scaler.fit_transform(df[num_cols])

std_df = pd.DataFrame(std_data, columns=num_cols)
print(std_df.head())
```

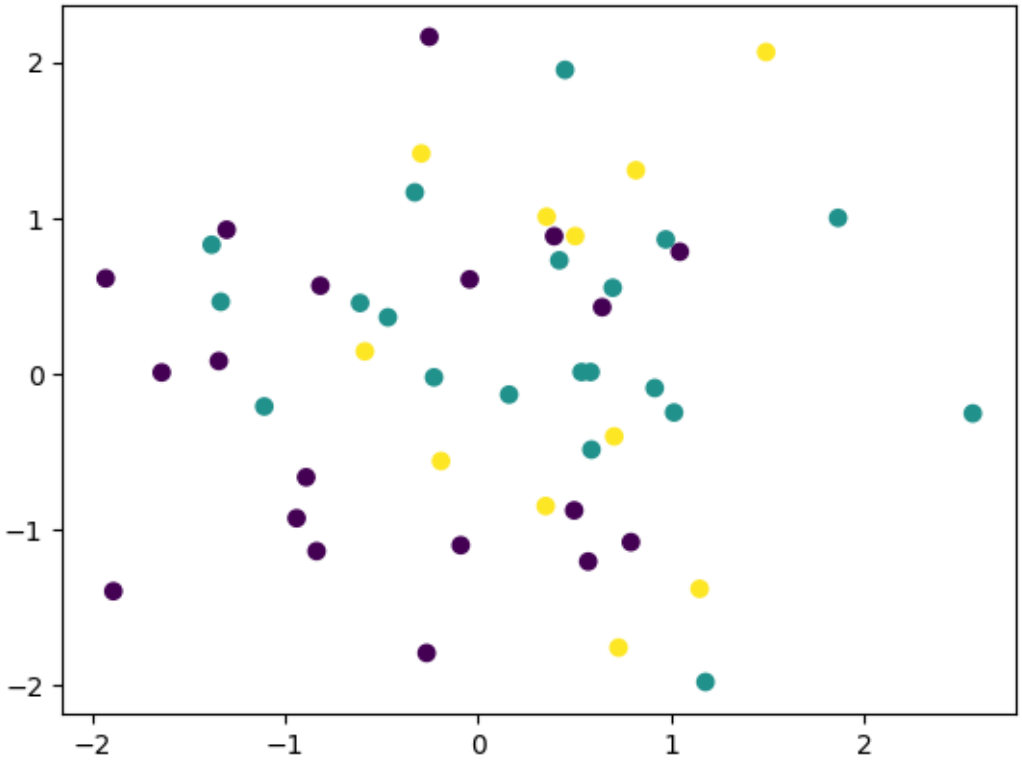
```
   fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
0      -0.728274          0.662757    -0.590571       -0.104542   0.369274
1      -0.442676          1.236296    -0.590571        1.724938   0.061546
2      -0.442676         -1.567674    -0.406017       -0.104542  -1.784827
3       1.984904         -0.675502     1.993176       -0.104542   0.061546
4      -0.371277          0.344124    -0.406017       -1.411313   1.292461

   free sulfur dioxide  total sulfur dioxide   density        pH  sulphates  \
0            -1.315587             -1.861983  1.008996  1.920589  -1.380850
1             1.754116              1.063990 -0.931381 -0.442040   1.291763
2            -0.438529             -0.088666 -0.543305  0.015243   0.623610
3             0.000000              0.443329  1.397071 -0.746896  -0.935414
4             0.000000              0.443329 -0.931381 -0.746896   0.400892

   alcohol  quality
0     -2.0     -0.5
1      0.5     -0.5
2      0.5     -0.5
3      0.5      2.0
4      0.5     -0.5
```

slip9

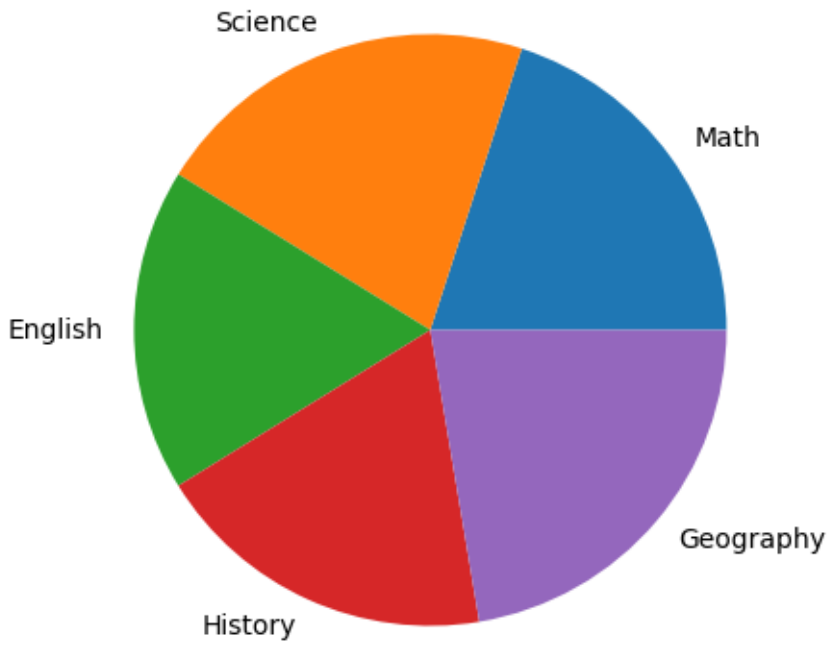
October 15, 2024

[1]:
```
# Generate a random array of 50 integers and display them using a line chart,␣
 ↪scatter plot.
# Apply appropriate color, labels and styling options
# slip 9 and slip 15 are same

import matplotlib.pyplot as plt
import numpy as np
from matplotlib import colors
from matplotlib.ticker import PercentFormatter

no_of_balls = 50
x = np.random.randn(50)
y= np.random.randn(50)

colors = [np.random.randint(1,4) for i in range(no_of_balls)]
plt.plot(x,y)
plt.show()
plt.scatter(x,y,c=colors)
plt.show()
```


2
1
0
−1
−2
−2
−1
0
1
2


2
1
0
−1
−2
−2
−1
0
1
2

[3]:
```
# Create two lists, one representing subject names and the other representing␣
 ↪marks
# obtained in those subjects. Display the data in a pie chart

import matplotlib.pyplot as plt
import numpy as np

subjects = ['Math', 'Science', 'English', 'History', 'Geography']
marks = [85, 90, 75, 80, 95]

fig = plt.figure(figsize=(5, 5))
plt.pie(marks,labels=subjects)
plt.show()
```



[5]:
```
# Write a program in python to perform following task (Use winequality-red.csv␣
 ↪)  Import
# Dataset and do the followings: a) Describing the dataset b) Shape of the␣
 ↪dataset c) Display first
# 3 rows from dataset
```

```python
# (similer slip4 of 2)

import pandas as pd

df = pd.read_csv('Data_filled.csv')

print("Dataset Description:")
print(df.describe())

print("\nShape of the dataset:", df.shape)

print("\nFirst 3 rows of the dataset:")
print(df.head(3))
```

```
Dataset Description:
             salary        age
count      5.000000   5.000000
mean   60000.000000  29.500000
std     7071.067812   3.640055
min    50000.000000  25.000000
25%    60000.000000  28.000000
50%    60000.000000  29.500000
75%    60000.000000  30.000000
max    70000.000000  35.000000

Shape of the dataset: (5, 3)

First 3 rows of the dataset:
      name   salary   age
0    Alice  50000.0  30.0
1      Bob  60000.0  25.0
2  Charlie  60000.0  29.5
```

slip10

October 15, 2024

[2]:
```
#  Write a python program to Display column-wise mean, and median for the
# SOCR HeightWeight dataset.

import pandas as pd

df = pd.read_csv('hw3.csv')
print("mean is \n",df.mean())
print("median is \n",df.median())
```

```
mean is
 HeightCm    170.0
WeightKg     70.0
dtype: float64
median is
 HeightCm    170.0
WeightKg     70.0
dtype: float64
```

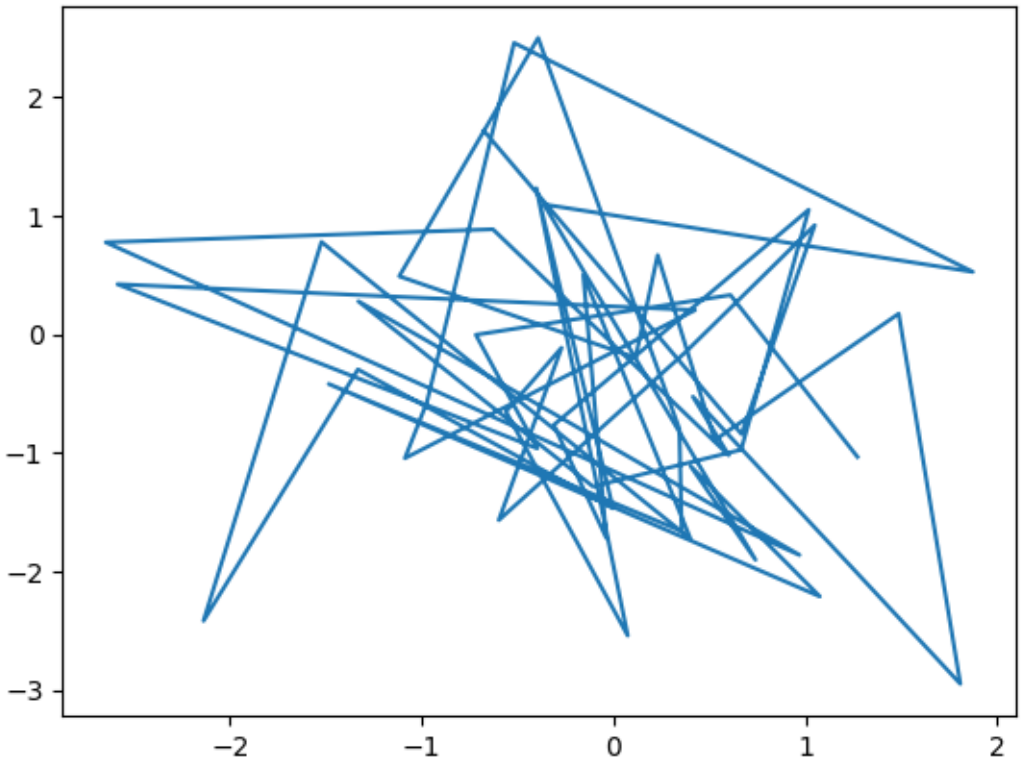
[3]:
```
# Write a python program to compute sum of Manhattan distance between all pairs␣
 ↪of
# points.

points = [(1, 2), (3, 4), (5, 6)]
total_distance = 0

for i in range(len(points)):
    for j in range(i + 1, len(points)):
        total_distance += abs(points[i][0] - points[j][0]) + abs(points[i][1] -␣
 ↪points[j][1])

print("Sum of Manhattan distances:", total_distance)
```

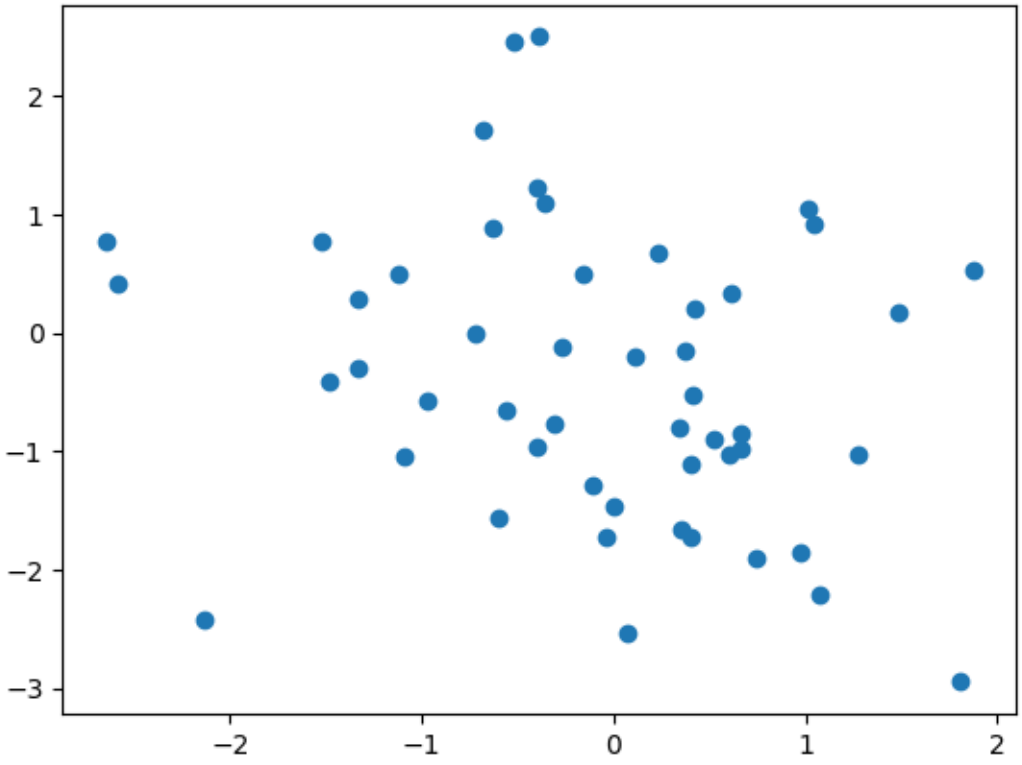
```
Sum of Manhattan distances: 16
```

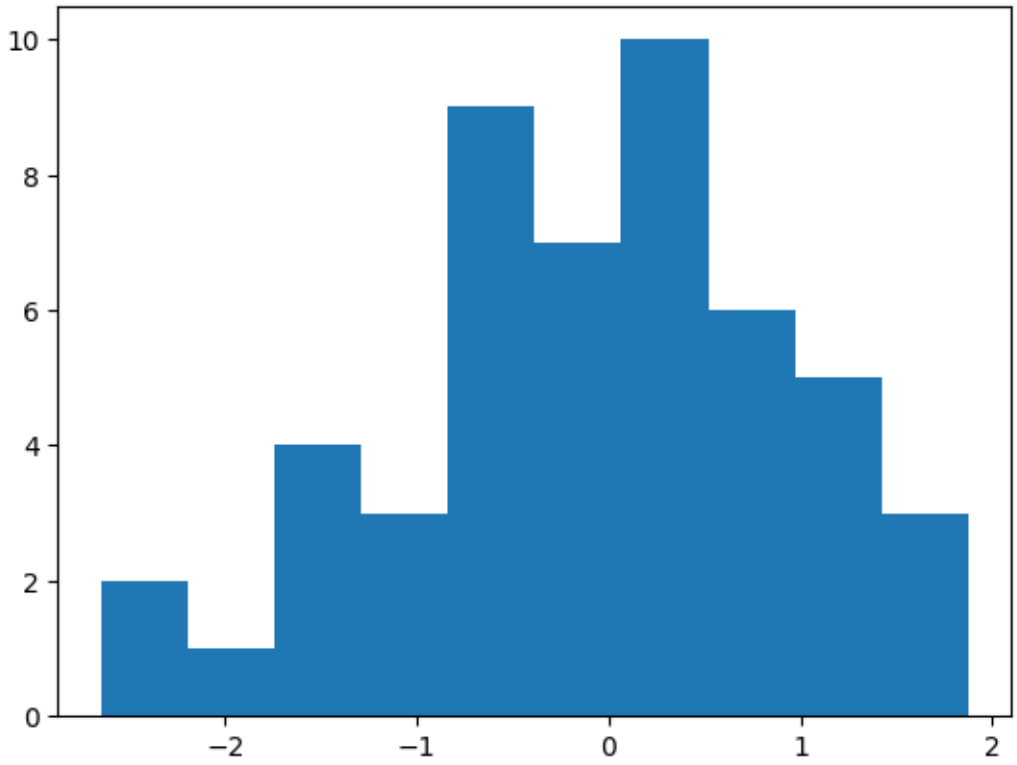
slip12

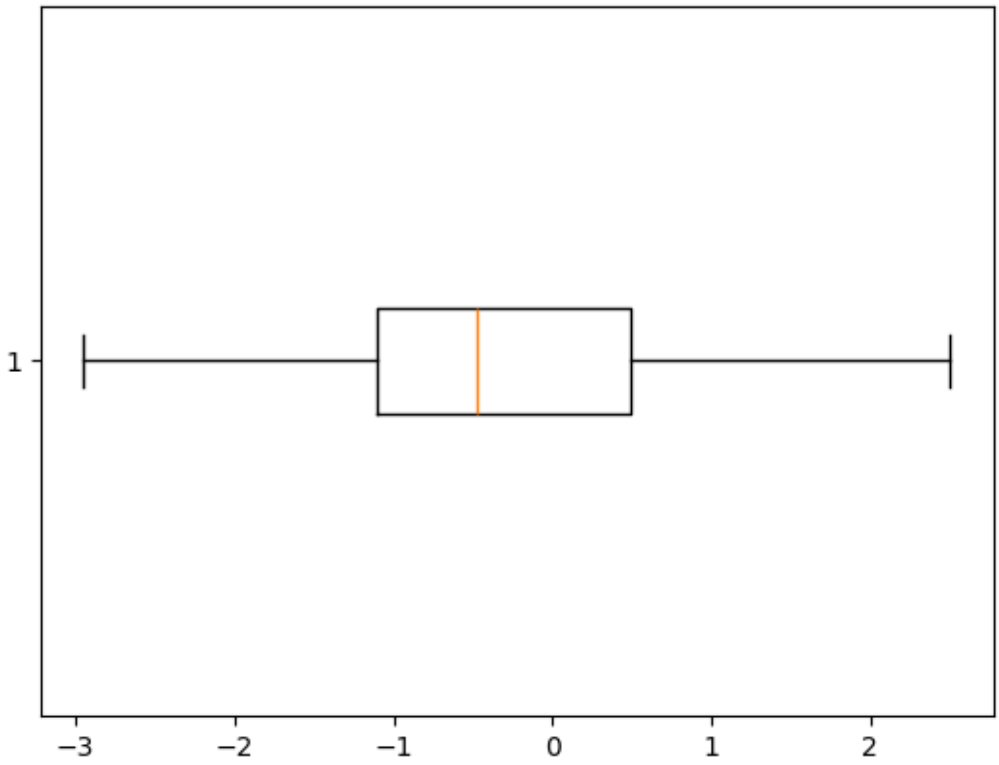
October 15, 2024

[1]:
```
# Generate a random array of 50 integers and display them using a line chart,␣
 ↪scatter plot,
# histogram and box plot. Apply appropriate color, labels and styling options.

# (this program is same as slip4 of 1 same code)

import matplotlib.pyplot as plt
import numpy as np

x=np.random.randn(50)
y=np.random.randn(50)

plt.plot(x,y)
plt.show()
plt.scatter(x,y)
plt.show()
plt.hist(x)
plt.show()
plt.boxplot(y,vert=False)
plt.show()
```


2
1
0
−1
−2
−3
−2
−1
0
1
2


2
1
0
−1
−2
−3
−2
−1
0
1
2


10
8
6
4
2
0
−2
−1
0
1
2



[2]:
```
# Write a Python program to create data frame containing column name, salary,
# department add 10 rows with some missing and duplicate values to the data␣
 ↪frame. Also drop all
# null and empty values. Print the modified data frame

import pandas as pd
import numpy as np

# Create a DataFrame with some missing and duplicate values
data = {
    'Name': ['Alice', 'Bob', 'Charlie', 'David', 'Eve', 'Frank', 'Grace',␣
 ↪'Alice', None, 'Hannah'],
    'Salary': [70000, 80000, None, 60000, 90000, 75000, None, 80000, 95000,␣
 ↪50000],
    'Department': ['HR', 'IT', 'Finance', None, 'Marketing', 'IT', 'HR', 'IT',␣
 ↪'Finance', None]
}

df = pd.DataFrame(data)

# Display the original DataFrame
print("Original DataFrame:")
```

```python
print(df)

# Drop all null and empty values
df_cleaned = df.dropna()

# Display the modified DataFrame
print("\nModified DataFrame (after dropping nulls):")
print(df_cleaned)
```

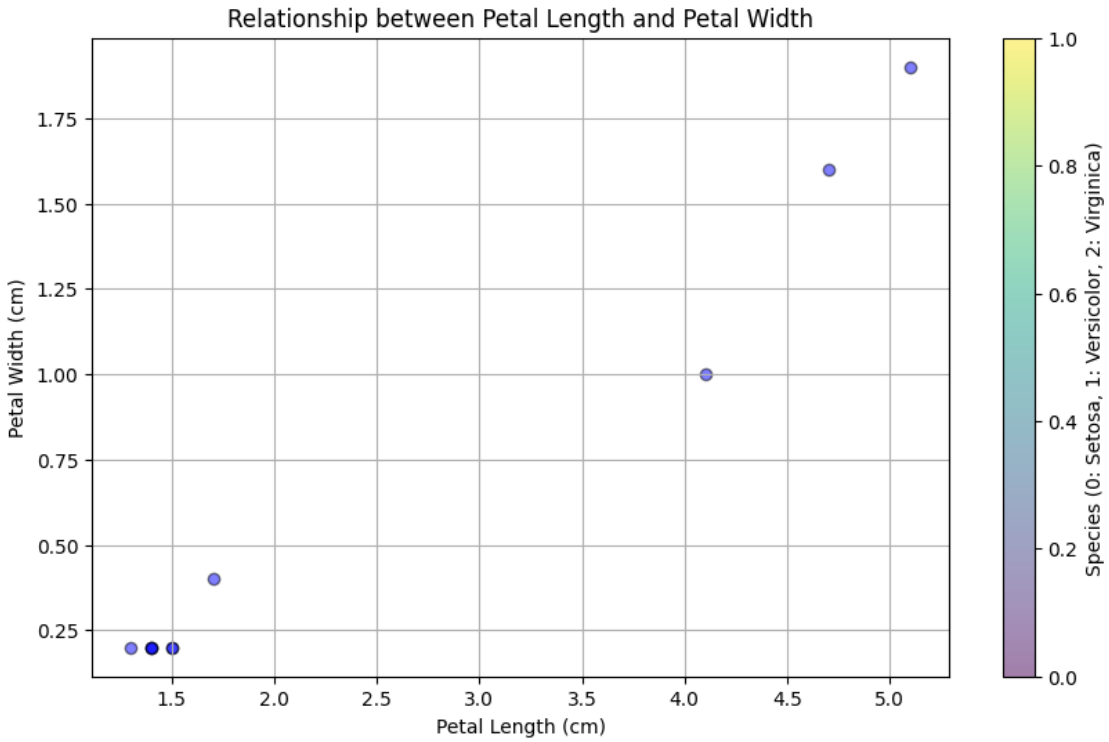
```
Original DataFrame:
      Name   Salary Department
0    Alice  70000.0         HR
1      Bob  80000.0         IT
2  Charlie      NaN    Finance
3    David  60000.0       None
4      Eve  90000.0  Marketing
5    Frank  75000.0         IT
6    Grace      NaN         HR
7    Alice  80000.0         IT
8     None  95000.0    Finance
9   Hannah  50000.0       None

Modified DataFrame (after dropping nulls):
    Name   Salary Department
0  Alice  70000.0         HR
1    Bob  80000.0         IT
4    Eve  90000.0  Marketing
5  Frank  75000.0         IT
7  Alice  80000.0         IT
```

slip13

October 15, 2024

[4]:
```
# Write a Python program to create a graph to find relationship between the␣
 ↪petal length
# and petal width.(Use iris.csv dataset)

import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('Iris.csv')

print(df.head())

plt.figure(figsize=(10, 6))
plt.scatter(df['PetalLength'], df['PetalWidth'], c='blue', alpha=0.5,␣
 ↪edgecolor='k')
plt.colorbar(label='Species (0: Setosa, 1: Versicolor, 2: Virginica)')
plt.title('Relationship between Petal Length and Petal Width')
plt.xlabel('Petal Length (cm)')
plt.ylabel('Petal Width (cm)')
plt.grid()
plt.show()
```

```
   SepalLength  SepalWidth  PetalLength  PetalWidth Species
0          5.1         3.5          1.4         0.2  setosa
1          4.9         3.0          1.4         0.2  setosa
2          4.7         3.2          1.3         0.2  setosa
3          4.6         3.1          1.5         0.2  setosa
4          5.0         3.6          1.4         0.2  setosa
```



[5]:
```python
# Write a Python program to find the maximum and minimum value of a given flattened
# array

import numpy as np

array = np.array([3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5])
max_value = np.max(array)
min_value = np.min(array)

print(f"Flattened Array: {array}")
print(f"Maximum Value: {max_value}")
print(f"Minimum Value: {min_value}")
```

```
Flattened Array: [3 1 4 1 5 9 2 6 5 3 5]
Maximum Value: 9
Minimum Value: 1
```

slip14

October 15, 2024

[1]:
```
# Write a Python NumPy program to compute the weighted average along the␣
 ↪specified
# axis of a given flattened array.

import numpy as np

data = np.array([10, 20, 30, 40, 50])
weights = np.array([1, 2, 3, 4, 5])
weighted_average = np.average(data, weights=weights)

print(f"Flattened Array: {data}")
print(f"Weights: {weights}")
print(f"Weighted Average: {weighted_average}")
```

```
Flattened Array: [10 20 30 40 50]
Weights: [1 2 3 4 5]
Weighted Average: 36.666666666666664
```

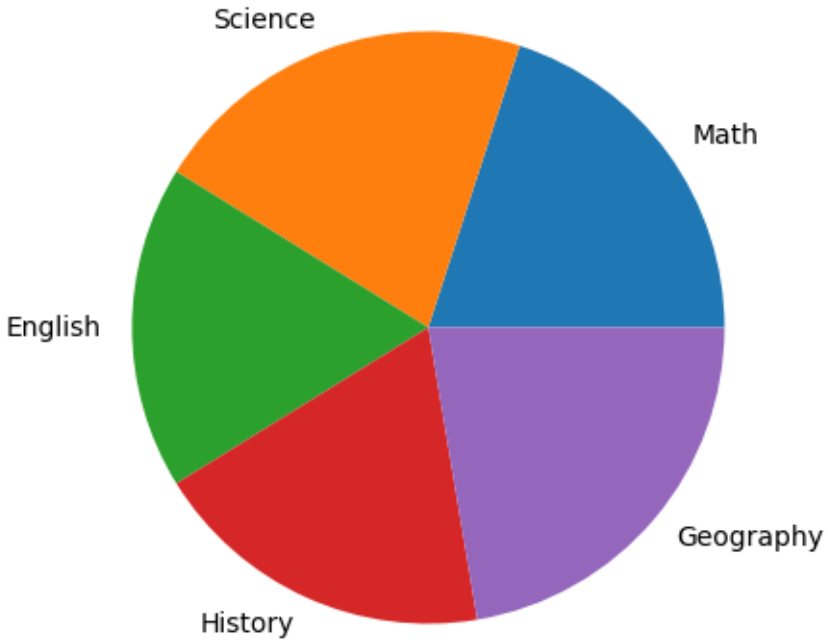
[1]:
```
#  Write a Python program to view basic statistical details of the data (Use␣
 ↪advertising.csv)
# (similer as slip1 of 2)

import pandas as pd

df = pd.read_csv('s7.csv')
basic_stats = df.describe()
print("Basic Statistical Details:")
print(basic_stats)
```

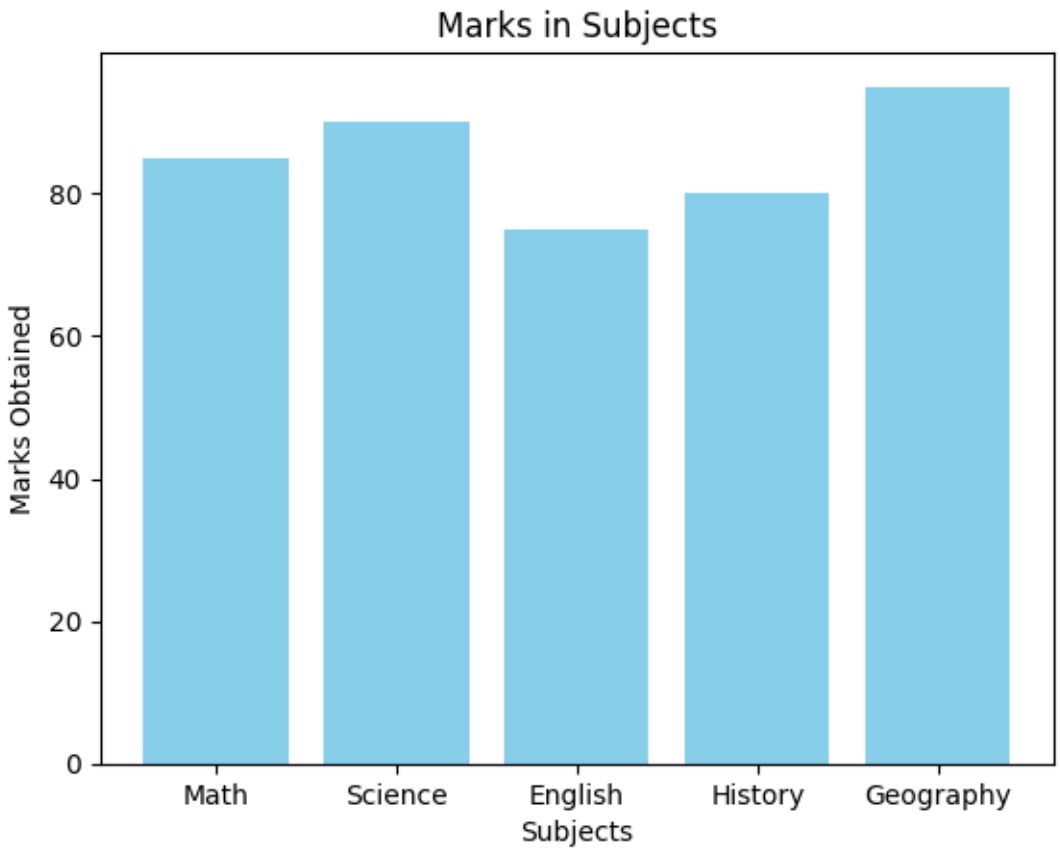
```
Basic Statistical Details:
       Country Purchased
count        4         4
unique       3         2
top        USA       Yes
freq         2         2
```

slip16

October 15, 2024

[3]:
```
# Write a python program to create two lists, one representing subject names␣
 ↪and the other
# representing marks obtained in those subjects. Display the data in a pie␣
 ↪chart and bar chart
# (similer program as slip 9 of 2)

import matplotlib.pyplot as plt
import numpy as np

subjects = ['Math', 'Science', 'English', 'History', 'Geography']
marks = [85, 90, 75, 80, 95]

fig = plt.figure(figsize=(5, 5))
plt.pie(marks,labels=subjects)
plt.show()

plt.subplot(1, 1, 1)
plt.bar(subjects, marks, color='skyblue')
plt.xlabel('Subjects')
plt.ylabel('Marks Obtained')
plt.title('Marks in Subjects')
plt.show()
```


Science
Math
English
Geography
History



[4]:
```
#  Write a python program to create a data frame for students' information such as name,
# graduation percentage and age. Display average age of students, average of graduation
# percentage

import pandas as pd

data = {
    'Name': ['Alice', 'Bob', 'Charlie', 'David', 'Eva'],
    'Graduation Percentage': [85.5, 78.0, 92.5, 88.0, 80.0],
    'Age': [21, 22, 20, 23, 21]
}

# Create a DataFrame from the dictionary
df = pd.DataFrame(data)

# Calculate the average age of students
average_age = df['Age'].mean()
```

```python
# Calculate the average graduation percentage
average_graduation_percentage = df['Graduation Percentage'].mean()

print("Student Information:")
print(df)
print(f"\nAverage Age of Students: {average_age:.2f}")
print(f"Average Graduation Percentage: {average_graduation_percentage:.2f}")
```

```
Student Information:
      Name  Graduation Percentage  Age
0    Alice                   85.5   21
1      Bob                   78.0   22
2  Charlie                   92.5   20
3    David                   88.0   23
4      Eva                   80.0   21

Average Age of Students: 21.40
Average Graduation Percentage: 84.80
```

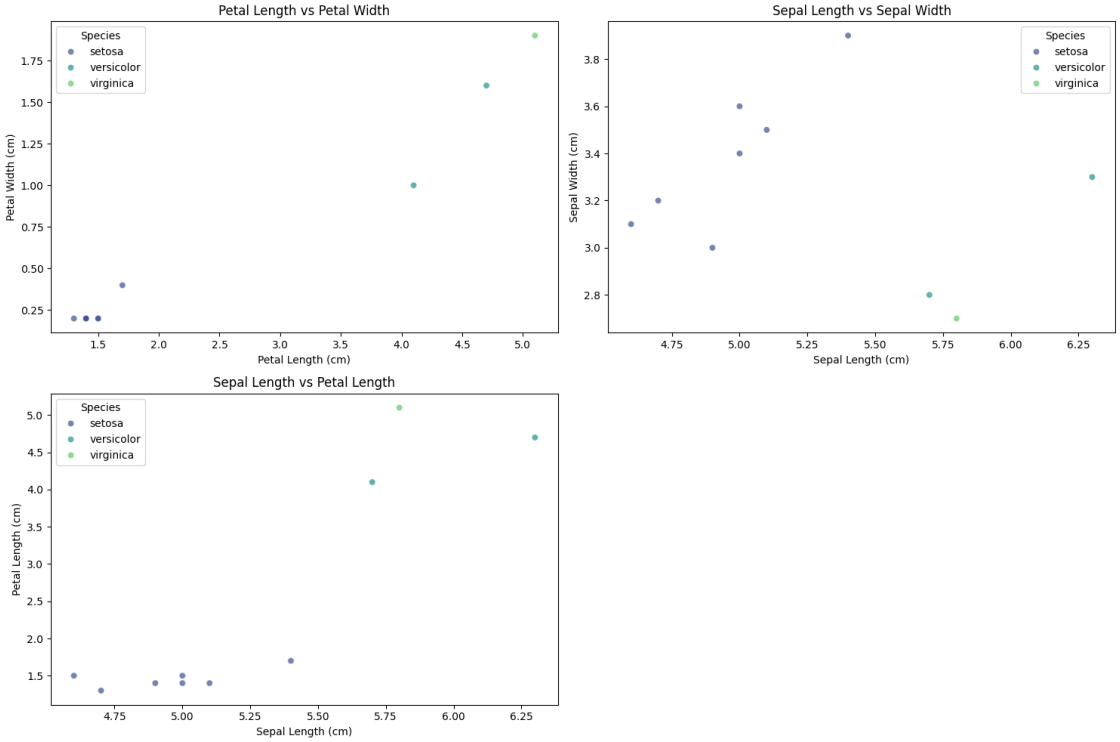
slip17

October 15, 2024

[8]:
```
# Write a Python program to draw scatter plots to compare two features of the␣
 ↪iris dataset
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load the iris dataset
iris = pd.read_csv("Iris.csv")

# Display the first few rows of the dataset
print(iris.head())

# Set up the matplotlib figure
plt.figure(figsize=(15, 10))

# Create scatter plot for Petal Length vs Petal Width
plt.subplot(2, 2, 1)
sns.scatterplot(data=iris, x='PetalLength', y='PetalWidth', hue='Species',␣
 ↪palette='viridis', alpha=0.7)
plt.title("Petal Length vs Petal Width")
plt.xlabel("Petal Length (cm)")
plt.ylabel("Petal Width (cm)")

# Create scatter plot for Sepal Length vs Sepal Width
plt.subplot(2, 2, 2)
sns.scatterplot(data=iris, x='SepalLength', y='SepalWidth', hue='Species',␣
 ↪palette='viridis', alpha=0.7)
plt.title("Sepal Length vs Sepal Width")
plt.xlabel("Sepal Length (cm)")
plt.ylabel("Sepal Width (cm)")

# Create scatter plot for Sepal Length vs Petal Length
plt.subplot(2, 2, 3)
sns.scatterplot(data=iris, x='SepalLength', y='PetalLength', hue='Species',␣
 ↪palette='viridis', alpha=0.7)
plt.title("Sepal Length vs Petal Length")
plt.xlabel("Sepal Length (cm)")
```

```
plt.ylabel("Petal Length (cm)")

# Adjust layout
plt.tight_layout()

# Show the plot
plt.show()
```

```
   SepalLength  SepalWidth  PetalLength  PetalWidth Species
0          5.1         3.5          1.4         0.2  setosa
1          4.9         3.0          1.4         0.2  setosa
2          4.7         3.2          1.3         0.2  setosa
3          4.6         3.1          1.5         0.2  setosa
4          5.0         3.6          1.4         0.2  setosa
```



[9]:
```
# Write a Python program to create a data frame containing columns name, age ,␣
↪salary,
# department . Add 10 rows to the data frame. View the data frame

import pandas as pd

data = {
```

```
    'name': ['Alice', 'Bob', 'Charlie', 'David', 'Eva', 'Frank', 'Grace',␣
 ↪'Hannah', 'Ian', 'Judy'],
    'age': [25, 30, 22, 28, 26, 35, 29, 31, 24, 27],
    'salary': [50000, 60000, 45000, 70000, 52000, 65000, 58000, 62000, 48000,␣
 ↪55000],
    'department': ['HR', 'IT', 'Finance', 'IT', 'HR', 'Sales', 'Marketing',␣
 ↪'Finance', 'Sales', 'Marketing']
}

df = pd.DataFrame(data)
print(df)
```

```
      name  age  salary department
0    Alice   25   50000         HR
1      Bob   30   60000         IT
2  Charlie   22   45000    Finance
3    David   28   70000         IT
4      Eva   26   52000         HR
5    Frank   35   65000      Sales
6    Grace   29   58000  Marketing
7   Hannah   31   62000    Finance
8      Ian   24   48000      Sales
9     Judy   27   55000  Marketing
```

slip19

October 15, 2024

[4]:
```
# Write a Python program To create a dataframe containing columns name, age and
# percentage. Add 10 rows to the dataframe. View the dataframe.

import pandas as pd

data = {
    'name': ['Alice', 'Bob', 'Charlie', 'David', 'Eva', 'Frank', 'Grace',␣
 ↪'Hannah', 'Ian', 'Judy'],
    'age': [20, 22, 21, 23, 24, 25, 22, 26, 27, 19],
    'percentage': [85.5, 90.2, 78.0, 88.5, 92.0, 80.0, 84.5, 75.5, 89.0, 91.0]
}

df = pd.DataFrame(data)
print(df)

# # below code for slip19 of 2
# print("Shape of the dataset:", df.shape)
# print("features :",df.info())
# print("\nDescription of the Data:\n", df.describe())
# print("\nData Types:\n", df.dtypes)
```

```
      name  age  percentage
0    Alice   20        85.5
1      Bob   22        90.2
2  Charlie   21        78.0
3    David   23        88.5
4      Eva   24        92.0
5    Frank   25        80.0
6    Grace   22        84.5
7   Hannah   26        75.5
8      Ian   27        89.0
9     Judy   19        91.0
```

[2]:
```
# slip 19 of 2 program in slip 4 of 2 (program is same )

import pandas as pd
import numpy as np
```

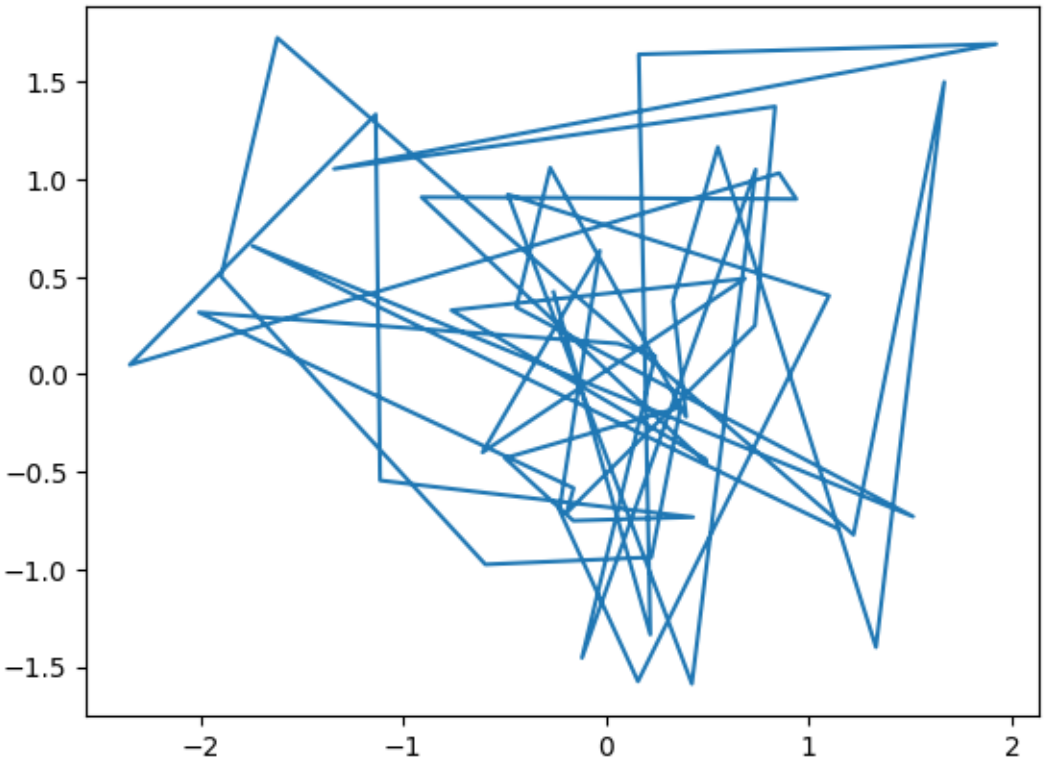
```python
# Initial data
data = {
    'name': ['Alice', 'Bob', 'Charlie', 'David', 'Eva', 'Frank', 'Grace',␣
 ↪'Hannah', 'Ian', 'Judy'],
    'age': [20, 22, 21, 23, 24, 25, 22, 26, 27, 19],
    'percentage': [85.5, 90.2, 78.0, 88.5, 92.0, 80.0, 84.5, 75.5, 89.0, 91.0]
}

# Create DataFrame
df = pd.DataFrame(data)

# Add duplicate rows
duplicates = pd.DataFrame({
    'name': ['Alice', 'Bob', 'Charlie', 'Charlie', 'Eva'],
    'age': [20, 22, 21, np.nan, 24],
    'percentage': [85.5, 90.2, 78.0, 88.5, 92.0]
})

# Concatenate the original DataFrame and the duplicates
df = pd.concat([df, duplicates], ignore_index=True)

# Add a new column 'remarks' with empty values
df['remarks'] = ''

# Display the DataFrame
print(df)
```

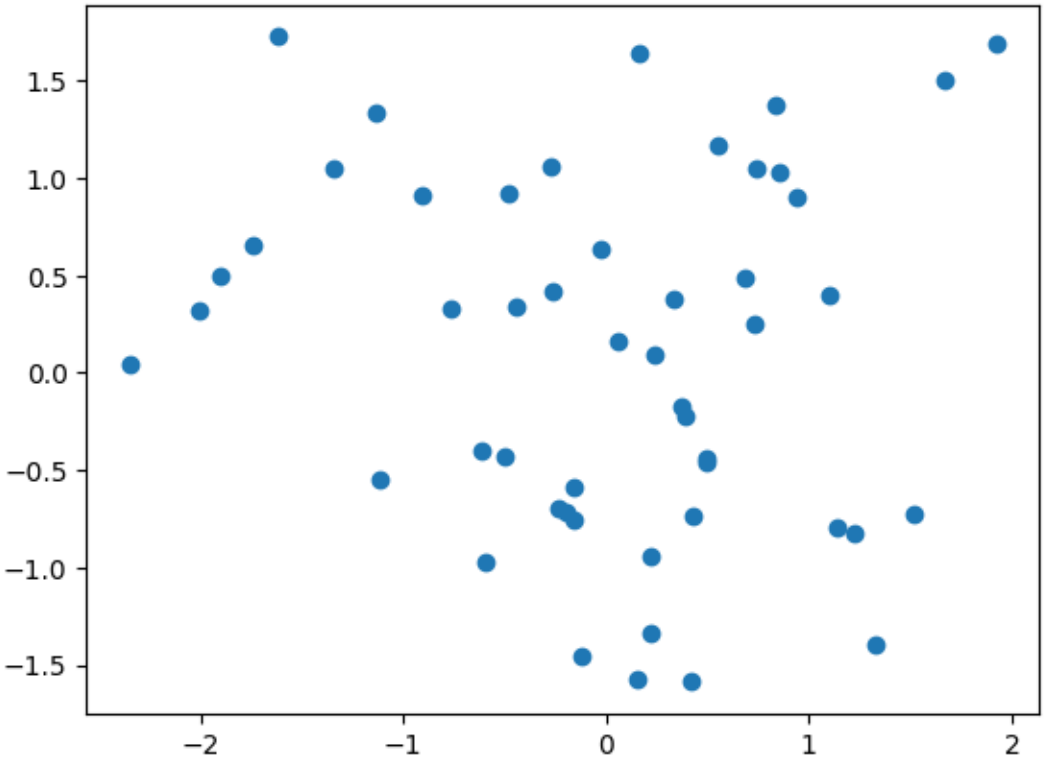
```
       name   age  percentage remarks
0     Alice  20.0        85.5
1       Bob  22.0        90.2
2   Charlie  21.0        78.0
3     David  23.0        88.5
4       Eva  24.0        92.0
5     Frank  25.0        80.0
6     Grace  22.0        84.5
7    Hannah  26.0        75.5
8       Ian  27.0        89.0
9      Judy  19.0        91.0
10    Alice  20.0        85.5
11      Bob  22.0        90.2
12  Charlie  21.0        78.0
13  Charlie   NaN        88.5
14      Eva  24.0        92.0
```

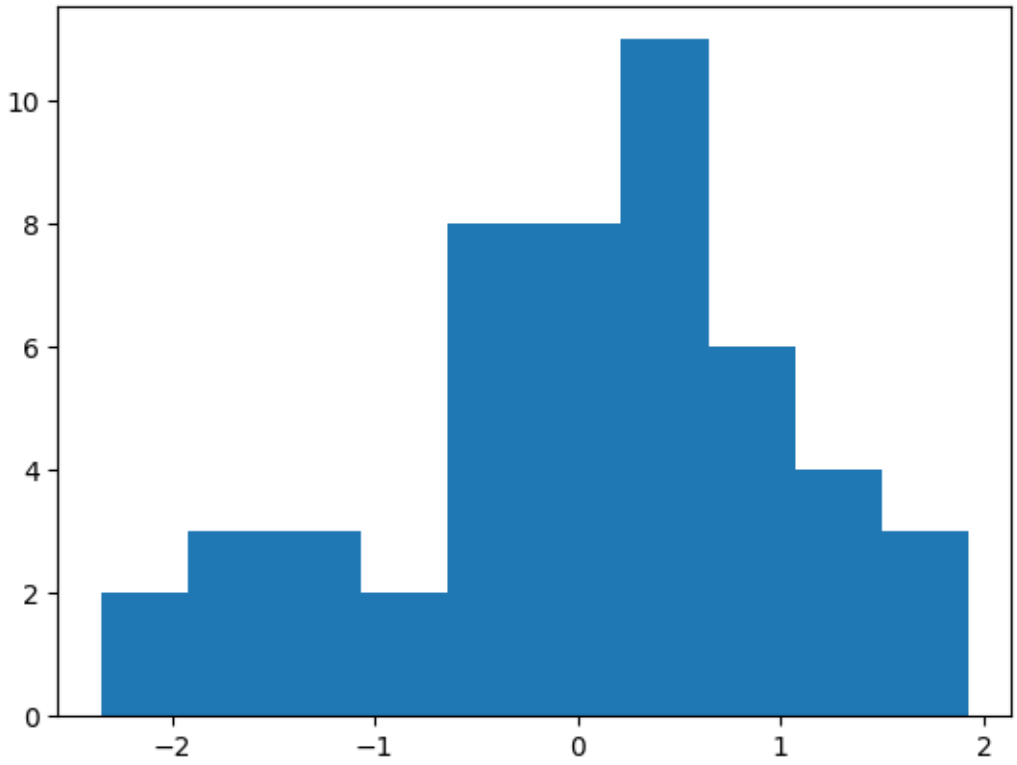
slip20

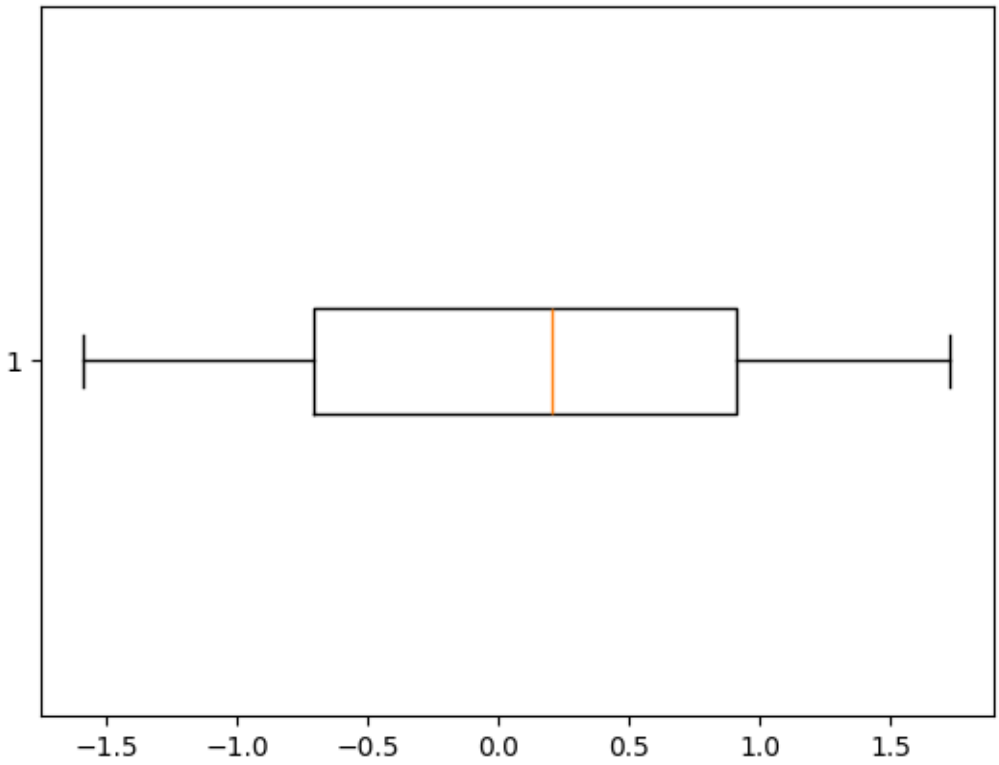
October 15, 2024

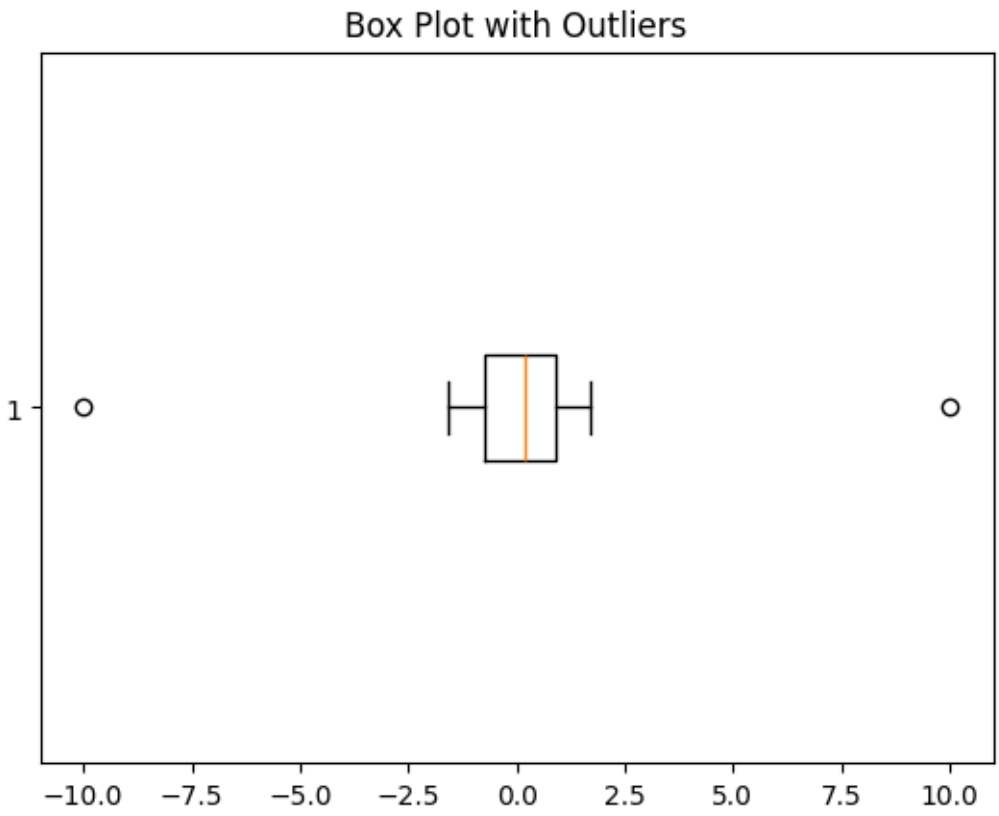
[2]:
```
# slip20-1 : Generate a random array of 50 integers and display them using a line chart, scatter plot,
# histogram and box plot. Apply appropriate color, labels and styling options.



import matplotlib.pyplot as plt
import numpy as np

x=np.random.randn(50)
y=np.random.randn(50)

plt.plot(x,y)
plt.show()
plt.scatter(x,y)
plt.show()
plt.hist(x)
plt.show()
plt.boxplot(y,vert=False)
plt.show()

# slip20-2: Add two outliers to the above data and display the box plot.

outliers = np.append(y, [10, -10])  # Adding two outliers

# Display the box plot with outliers
plt.boxplot(outliers, vert=False)
plt.title("Box Plot with Outliers")
plt.show()
```


1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2
−1
0
1
2


1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2
−1
0
1
2


10
8
6
4
2
0
−2
−1
0
1
2

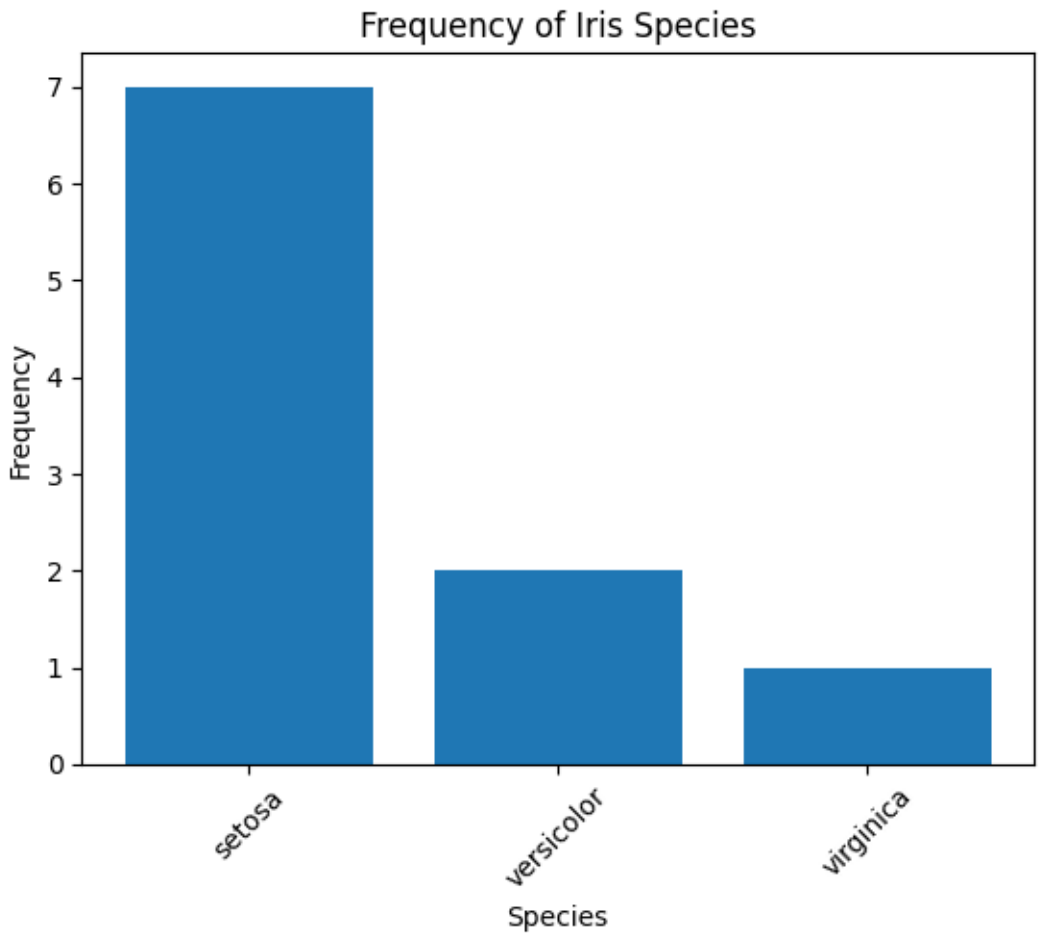

1
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5


Box Plot with Outliers
1
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

slip21

October 15, 2024

[2]:
```
#  Import dataset "iris.csv". Write a Python program to create a Bar plot to␣
 ↪get the frequency
# of the three species of the Iris data.

import pandas as pd
import matplotlib.pyplot as plt

# Import the dataset
df = pd.read_csv('Iris.csv')

# Count the frequency of each species
species_counts = df['Species'].value_counts()

# Create a bar plot
plt.bar(species_counts.index, species_counts.values)
plt.title('Frequency of Iris Species')
plt.xlabel('Species')
plt.ylabel('Frequency')
plt.xticks(rotation=45)
plt.show()
```



[14]:
```
# Write a Python program to create a histogram of the three species of the Iris data
import pandas as pd
import matplotlib.pyplot as plt

d = pd.read_csv("Iris.csv")

plt.figure(figsize=(10, 6))

d[d.Species == 'setosa'].plot.hist(x='PetalLength', y='PetalWidth', alpha=0.7, color='orange', label='setosa', bins=10, ax=plt.gca())
d[d.Species == 'versicolor'].plot.hist(x='PetalLength', y='PetalWidth', alpha=0.7, color='blue', label='versicolor', bins=10, ax=plt.gca())
d[d.Species == 'virginica'].plot.hist(x='PetalLength', y='PetalWidth', alpha=0.7, color='green', label='virginica', bins=10, ax=plt.gca())

# Set labels and title
```

```
plt.xlabel("Petal Length (cm)")
plt.ylabel("Frequency")
plt.title("Histogram of Petal Length by Species")
plt.legend()
plt.show()
```