python_practical_quetions

October 19, 2024

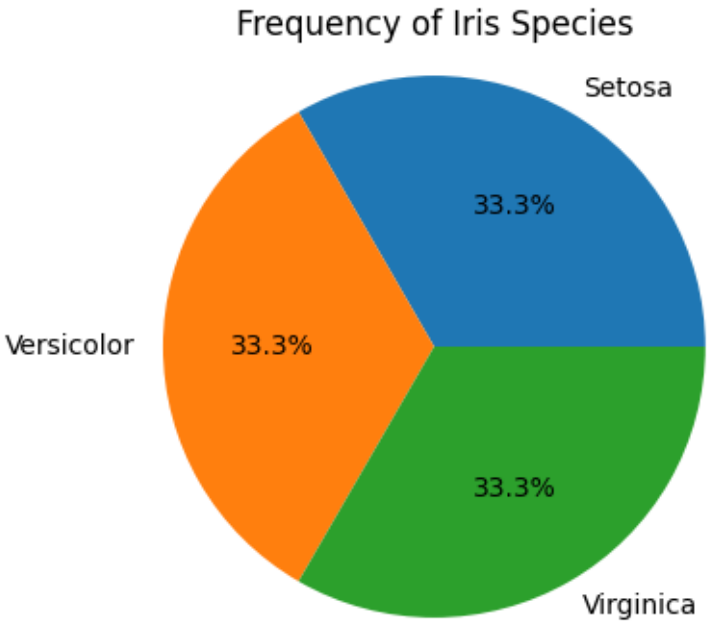
slip 1

A) Write a Python program to create a Pie plot to get the frequency of the three species of the Iris data (Use iris.csv)

[65]:
```
import pandas as pd
import matplotlib.pyplot as plt

iris_data = pd.read_csv("/home/ajay/Downloads/FDS_TEST_DATA/iris.csv")

print(pd.DataFrame(iris_data))

species_count = iris_data['variety'].value_counts()
print("species count :\n",species_count)

plt.figure(figsize=(4, 4))
plt.pie(species_count, labels=species_count.index, autopct='%1.1f%%')
plt.title('Frequency of Iris Species')
plt.axis('equal')
plt.show()
```

```
     sepal.length  sepal.width  petal.length  petal.width    variety
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
..            …            …             …            …         …
145           6.7          3.0           5.2          2.3  Virginica
146           6.3          2.5           5.0          1.9  Virginica
147           6.5          3.0           5.2          2.0  Virginica
148           6.2          3.4           5.4          2.3  Virginica
149           5.9          3.0           5.1          1.8  Virginica

[150 rows x 5 columns]
species count :
 variety
Setosa        50
Versicolor    50
```

```
Virginica     50
Name: count, dtype: int64
```



B) Write a Python program to view basic statistical details of the data.(Use wineequality-red.csv)

[66]:
```
import pandas as pd

wine_data = pd.read_csv("/home/ajay/Downloads/FDS_TEST_DATA/wineequality-red.
 ↪csv")

print("First few rows of the dataset:")
print(wine_data.head())

print("\nBasic Statistical Details:")
print(wine_data.describe())
```

```
First few rows of the dataset:
   fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
0            7.4              0.70         0.00             1.9      0.076
1            7.8              0.88         0.00             2.6      0.098
2            7.8              0.76         0.04             2.3      0.092
3           11.2              0.28         0.56             1.9      0.075
4            7.4              0.70         0.00             1.9      0.076

   free sulfur dioxide  total sulfur dioxide  density    pH  sulphates  \
0                 11.0                  34.0   0.9978  3.51       0.56
1                 25.0                  67.0   0.9968  3.20       0.68
```

```
2                15.0                 54.0   0.9970  3.26       0.65
3                17.0                 60.0   0.9980  3.16       0.58
4                11.0                 34.0   0.9978  3.51       0.56

   alcohol  quality
0      9.4        5
1      9.8        5
2      9.8        5
3      9.8        6
4      9.4        5

Basic Statistical Details:
       fixed acidity  volatile acidity  citric acid  residual sugar  \
count    1599.000000       1599.000000  1599.000000     1599.000000
mean        8.319637          0.527821     0.270976        2.538806
std         1.741096          0.179060     0.194801        1.409928
min         4.600000          0.120000     0.000000        0.900000
25%         7.100000          0.390000     0.090000        1.900000
50%         7.900000          0.520000     0.260000        2.200000
75%         9.200000          0.640000     0.420000        2.600000
max        15.900000          1.580000     1.000000       15.500000

         chlorides  free sulfur dioxide  total sulfur dioxide      density  \
count  1599.000000          1599.000000           1599.000000  1599.000000
mean      0.087467            15.874922             46.467792     0.996747
std       0.047065            10.460157             32.895324     0.001887
min       0.012000             1.000000              6.000000     0.990070
25%       0.070000             7.000000             22.000000     0.995600
50%       0.079000            14.000000             38.000000     0.996750
75%       0.090000            21.000000             62.000000     0.997835
max       0.611000            72.000000            289.000000     1.003690

                pH    sulphates      alcohol      quality
count  1599.000000  1599.000000  1599.000000  1599.000000
mean      3.311113     0.658149    10.422983     5.636023
std       0.154386     0.169507     1.065668     0.807569
min       2.740000     0.330000     8.400000     3.000000
25%       3.210000     0.550000     9.500000     5.000000
50%       3.310000     0.620000    10.200000     6.000000
75%       3.400000     0.730000    11.100000     6.000000
max       4.010000     2.000000    14.900000     8.000000
```

---

slip 2

A) Write a Python program for Handling Missing Value. Replace missing value of salary, age column with mean of that column.(Use Data.csv file).

[67]:
```
import pandas as pd

data = pd.read_csv("/home/ajay/Downloads/FDS_TEST_DATA/Data.csv")

print("Original Dataset:")
print(data.head())

print("\nMissing Values Before Replacement:")
print(data.isnull().sum())

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

print("\nDataset After Replacing Missing Values with Mean:")
print(data.head())

print("\nMissing Values After Replacement:")
print(data.isnull().sum())
```

```
Original Dataset:
   name   age   salary department
0  John  28.0  50000.0         HR
1  Jane  32.0      NaN    Finance
2  Mike  45.0  70000.0         IT
3  Kate  29.0  60000.0  Marketing
4   Tom   NaN  45000.0    Finance

Missing Values Before Replacement:
name          0
age           1
salary        2
department    0
dtype: int64

Dataset After Replacing Missing Values with Mean:
   name   age   salary department
0  John  28.0  50000.0         HR
1  Jane  32.0  55400.0    Finance
2  Mike  45.0  70000.0         IT
3  Kate  29.0  60000.0  Marketing
4   Tom  33.0  45000.0    Finance

Missing Values After Replacement:
name          0
age           0
salary        0
department    0
dtype: int64
```

B) Write a Python program to generate a line plot of name Vs salary

[68]:
```
import pandas as pd
import matplotlib.pyplot as plt

data = pd.read_csv("/home/ajay/Downloads/FDS_TEST_DATA/Data.csv")

print("Dataset:")
print(data.head())

plt.figure(figsize=(8, 5))
plt.plot(data['name'], data['salary'], marker='o', color='b', linestyle='-',␣
 ↪markersize=5)

plt.xlabel('Name')
plt.ylabel('Salary')
plt.title('Name vs Salary Line Plot')

plt.xticks(rotation=45)

plt.tight_layout()
plt.show()
```

```
Dataset:
   name   age   salary department
0  John  28.0  50000.0         HR
1  Jane  32.0      NaN    Finance
2  Mike  45.0  70000.0         IT
3  Kate  29.0  60000.0  Marketing
4   Tom   NaN  45000.0    Finance
```

C) Download the heights and weights dataset and load the dataset froma given csv file into a dataframe. Print the first, last 10 rows and random 20 rows also display shape of the dataset.

[69]:
```
import pandas as pd

file_path = '/home/ajay/Downloads/FDS_TEST_DATA/heights_weights.csv'
data = pd.read_csv(file_path)

print("Shape of the Dataset:", data.shape)

print("\nFirst 10 Rows:")
print(data.head(10))

print("\nLast 10 Rows:")
print(data.tail(10))

print("\nRandom 20 Rows:")
print(data.sample(20))
```

```
Shape of the Dataset: (20, 3)

First 10 Rows:
      Name  Height  Weight
0    Alice      65     150
1      Bob      70     180
2  Charlie      68     165
```

```
3    David      72     190
4      Eva      64     130
5    Frank      75     200
6    Grace      62     120
7   Hannah      67     140
8      Ian      74     185
9     Jack      69     170

Last 10 Rows:
      Name  Height  Weight
10    Liam      66     155
11     Mia      71     175
12    Noah      63     135
13  Olivia      68     160
14    Paul      76     210
15   Quinn      65     145
16    Rita      73     195
17   Steve      70     185
18    Tina      66     150
19     Uma      64     130

Random 20 Rows:
       Name  Height  Weight
0     Alice      65     150
1       Bob      70     180
15    Quinn      65     145
19      Uma      64     130
2   Charlie      68     165
7    Hannah      67     140
5     Frank      75     200
11      Mia      71     175
4       Eva      64     130
12     Noah      63     135
13   Olivia      68     160
3     David      72     190
16     Rita      73     195
18     Tina      66     150
8       Ian      74     185
10     Liam      66     155
6     Grace      62     120
17    Steve      70     185
9      Jack      69     170
14     Paul      76     210
```

---

slip 3

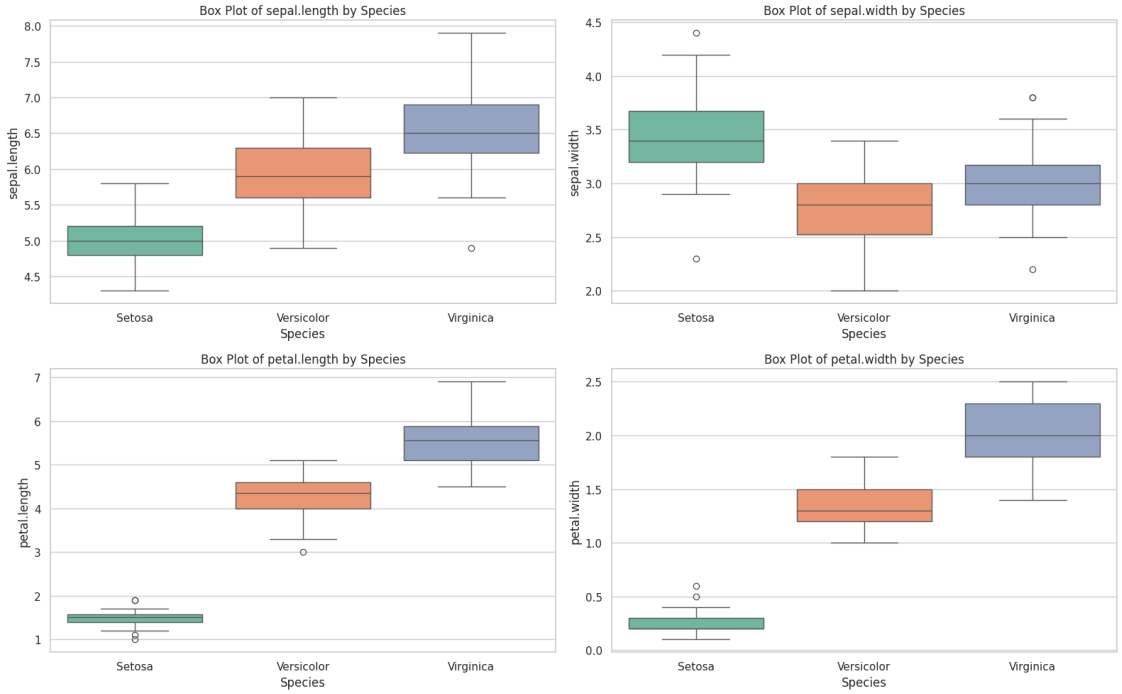
A)Write a Python program to create box plots to see how each feature i.e. Sepal Length, Sepal

Width, Petal Length, Petal Width are distributed across the three species.

[70]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

iris_data = pd.read_csv("/home/ajay/Downloads/FDS_TEST_DATA/iris.csv")

print("First few rows of the dataset:")
print(iris_data.head())

sns.set(style="whitegrid")

features = ['sepal.length', 'sepal.width', 'petal.length', 'petal.width']
species = iris_data['variety'].unique()

plt.figure(figsize=(16, 10))

for i, feature in enumerate(features):
    plt.subplot(2, 2, i + 1)
    sns.boxplot(x='variety', y=feature, data=iris_data, palette="Set2")
    plt.title(f'Box Plot of {feature} by Species')
    plt.xlabel('Species')
    plt.ylabel(feature)

plt.tight_layout()
plt.show()
```

```
First few rows of the dataset:
   sepal.length  sepal.width  petal.length  petal.width variety
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

/tmp/ipykernel_17471/3684776015.py:19: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.boxplot(x='variety', y=feature, data=iris_data, palette="Set2")
/tmp/ipykernel_17471/3684776015.py:19: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.
```

```
  sns.boxplot(x='variety', y=feature, data=iris_data, palette="Set2")
/tmp/ipykernel_17471/3684776015.py:19: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.boxplot(x='variety', y=feature, data=iris_data, palette="Set2")
/tmp/ipykernel_17471/3684776015.py:19: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.boxplot(x='variety', y=feature, data=iris_data, palette="Set2")
```



B) Write a Python program to view basic statistical details of the data (Use Heights and Weights Dataset)

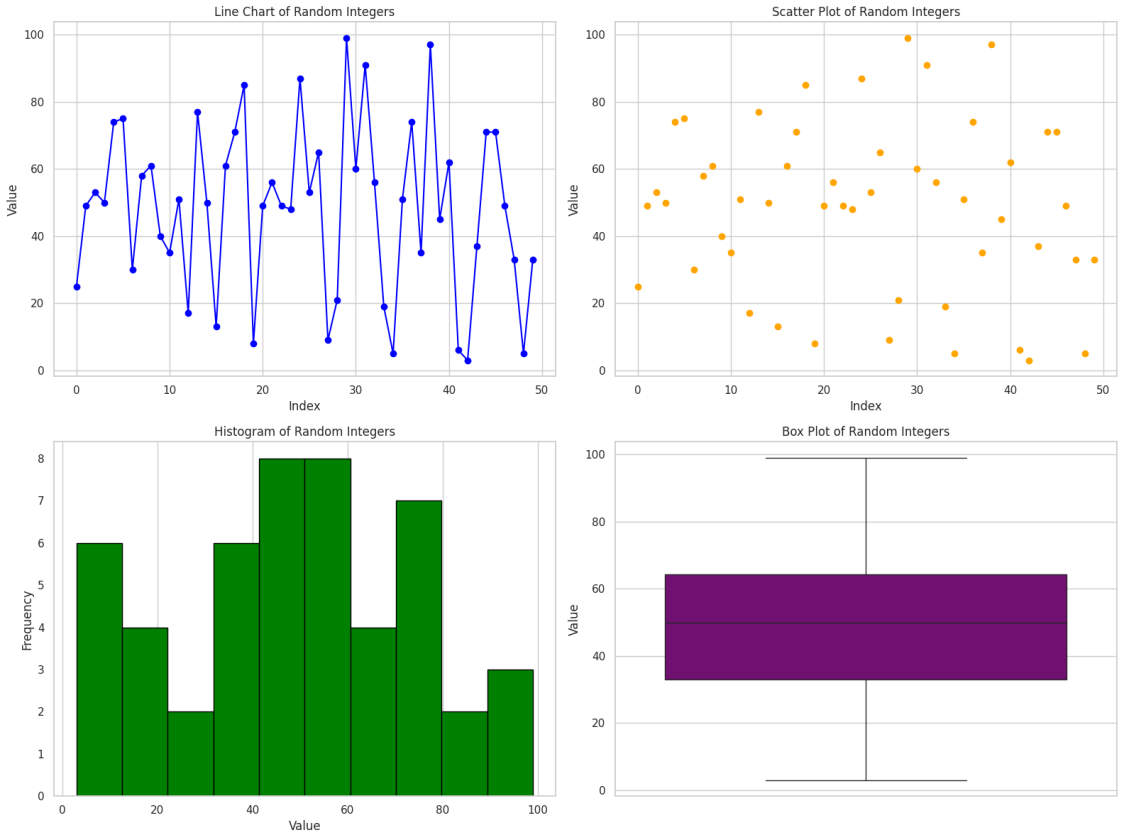
[71]:
```
# in slip 1 Q.2) B)
```

---

slip 4

A) 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.

[72]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

random_integers = np.random.randint(1, 101, size=50)

data = pd.DataFrame(random_integers, columns=['Random Integers'])

sns.set(style="whitegrid")

plt.figure(figsize=(16, 12))

plt.subplot(2, 2, 1)
plt.plot(data['Random Integers'], color='blue', marker='o', linestyle='-')
plt.title('Line Chart of Random Integers')
plt.xlabel('Index')
plt.ylabel('Value')
plt.grid(True)

plt.subplot(2, 2, 2)
plt.scatter(data.index, data['Random Integers'], color='orange')
plt.title('Scatter Plot of Random Integers')
plt.xlabel('Index')
plt.ylabel('Value')
plt.grid(True)

plt.subplot(2, 2, 3)
plt.hist(data['Random Integers'], bins=10, color='green', edgecolor='black')
plt.title('Histogram of Random Integers')
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.grid(axis='y')

plt.subplot(2, 2, 4)
sns.boxplot(y=data['Random Integers'], color='purple')
plt.title('Box Plot of Random Integers')
plt.ylabel('Value')

plt.tight_layout()
plt.show()
```



B) 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)

[73]:
```
import pandas as pd

file_path = '/home/ajay/Downloads/FDS_TEST_DATA/User_Data.csv'
user_data = pd.read_csv(file_path)

print("Shape of the dataset (rows, columns):")
print(user_data.shape)

num_rows, num_columns = user_data.shape
print(f"Number of rows: {num_rows}")
print(f"Number of columns: {num_columns}")

print("\nData types of each feature:")
print(user_data.dtypes)

print("\nFeature names:")
print(user_data.columns.tolist())
```

```
print("\nDescription of the dataset:")
print(user_data.describe(include='all'))
```

```
Shape of the dataset (rows, columns):
(10, 5)
Number of rows: 10
Number of columns: 5

Data types of each feature:
UserID              int64
Name               object
Age                 int64
Email              object
RegistrationDate   object
dtype: object

Feature names:
['UserID', 'Name', 'Age', 'Email', 'RegistrationDate']

Description of the dataset:
          UserID      Name        Age                Email RegistrationDate
count   10.00000        10  10.000000                   10               10
unique       NaN        10        NaN                   10               10
top          NaN  John Doe        NaN  johndoe@example.com       2022-01-15
freq         NaN         1        NaN                    1                1
mean     5.50000       NaN  32.800000                  NaN              NaN
std      3.02765       NaN   6.908931                  NaN              NaN
min      1.00000       NaN  22.000000                  NaN              NaN
25%      3.25000       NaN  28.250000                  NaN              NaN
50%      5.50000       NaN  32.000000                  NaN              NaN
75%      7.75000       NaN  37.250000                  NaN              NaN
max     10.00000       NaN  45.000000                  NaN              NaN
```

---

slip 5

A) 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.

[74]:
```
# in slip 4 Q.2) A)
```

B) 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)

[75]:
```
# in slip 4 Q.2) B)
```

---

slip 6

A) Write a Python program for Handling Missing Value. Replace missing value of salary, age column with mean of that column.(Use Data.csv file).

[76]:
```
# in slip 2 Q.2) A)
```

B) Write a Python program to generate a line plot of name Vs salary

[77]:
```
# in slip 2 Q.2) B)
```

C) Download the heights and weights dataset and load the dataset froma given csv file into a dataframe. Print the first, last 10 rows and random 20 rows also display shape of the dataset.

[78]:
```
# in slip 2 Q.2) C)
```

---

slip 7

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).

[79]:
```
import pandas as pd
from sklearn.preprocessing import LabelEncoder

data = pd.read_csv("/home/ajay/Downloads/FDS_TEST_DATA/Data7.csv")

print("Original Dataset:")
print(data)

data_onehot = pd.get_dummies(data, columns=['Country'], drop_first=True)

label_encoder = LabelEncoder()
data_onehot['Purchased'] = label_encoder.fit_transform(data_onehot['Purchased'])

print("\nDataset after One-Hot Encoding and Label Encoding:")
print(data_onehot)
```

```
Original Dataset:
  Country Purchased
0     USA       Yes
1  Canada        No
2      UK       Yes
3  Canada       Yes
4     USA        No
5      UK        No
6     USA       Yes
7      UK       Yes
8  Canada        No
9     USA       Yes
```

```
Dataset after One-Hot Encoding and Label Encoding:
   Purchased  Country_UK  Country_USA
0          1       False         True
1          0       False        False
2          1        True        False
3          1       False        False
4          0       False         True
5          0        True        False
6          1       False         True
7          1        True        False
8          0       False        False
9          1       False         True
```

---

slip 8

Write a program in python to perform following task : 1. Import Dataset from above link. 2. Standardizing Data (transform them into a standard Gaussian distribution with a mean of 0 and a standard deviation of 1) (Use winequality-red.csv)

[80]:
```
import pandas as pd
from sklearn.preprocessing import StandardScaler

data = pd.read_csv("/home/ajay/Downloads/FDS_TEST_DATA/wineequality-red.csv")

print("Original Dataset:")
print(data.head())

scaler = StandardScaler()

features = data.columns[:-1]

data_standardized = scaler.fit_transform(data[features])

data_standardized_df = pd.DataFrame(data_standardized, columns=features)

print("\nStandardized Dataset:")
print(data_standardized_df.head())
```

```
Original Dataset:
   fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
0            7.4              0.70         0.00             1.9      0.076
1            7.8              0.88         0.00             2.6      0.098
2            7.8              0.76         0.04             2.3      0.092
3           11.2              0.28         0.56             1.9      0.075
4            7.4              0.70         0.00             1.9      0.076

   free sulfur dioxide  total sulfur dioxide  density    pH  sulphates  \
0                 11.0                  34.0   0.9978  3.51       0.56
```

```
1                25.0                 67.0   0.9968  3.20       0.68
2                15.0                 54.0   0.9970  3.26       0.65
3                17.0                 60.0   0.9980  3.16       0.58
4                11.0                 34.0   0.9978  3.51       0.56

   alcohol  quality
0      9.4        5
1      9.8        5
2      9.8        5
3      9.8        6
4      9.4        5

Standardized Dataset:
   fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
0      -0.528360          0.961877    -1.391472       -0.453218  -0.243707
1      -0.298547          1.967442    -1.391472        0.043416   0.223875
2      -0.298547          1.297065    -1.186070       -0.169427   0.096353
3       1.654856         -1.384443     1.484154       -0.453218  -0.264960
4      -0.528360          0.961877    -1.391472       -0.453218  -0.243707

   free sulfur dioxide  total sulfur dioxide   density        pH  sulphates  \
0            -0.466193             -0.379133  0.558274  1.288643  -0.579207
1             0.872638              0.624363  0.028261 -0.719933   0.128950
2            -0.083669              0.229047  0.134264 -0.331177  -0.048089
3             0.107592              0.411500  0.664277 -0.979104  -0.461180
4            -0.466193             -0.379133  0.558274  1.288643  -0.579207

    alcohol
0 -0.960246
1 -0.584777
2 -0.584777
3 -0.584777
4 -0.960246
```

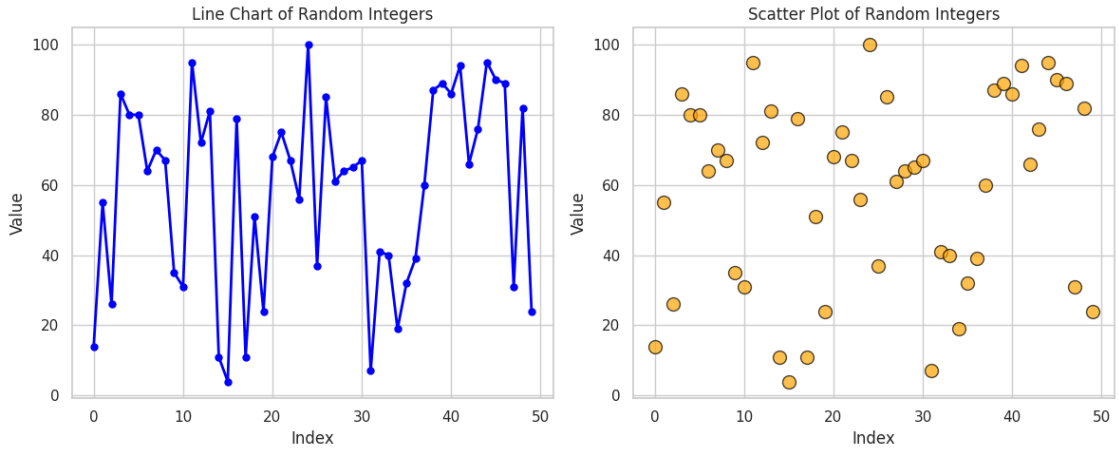
---

slip 9

A) Generate a random array of 50 integers and display them using a line chart, scatter plot. Apply appropriate color, labels and styling options.

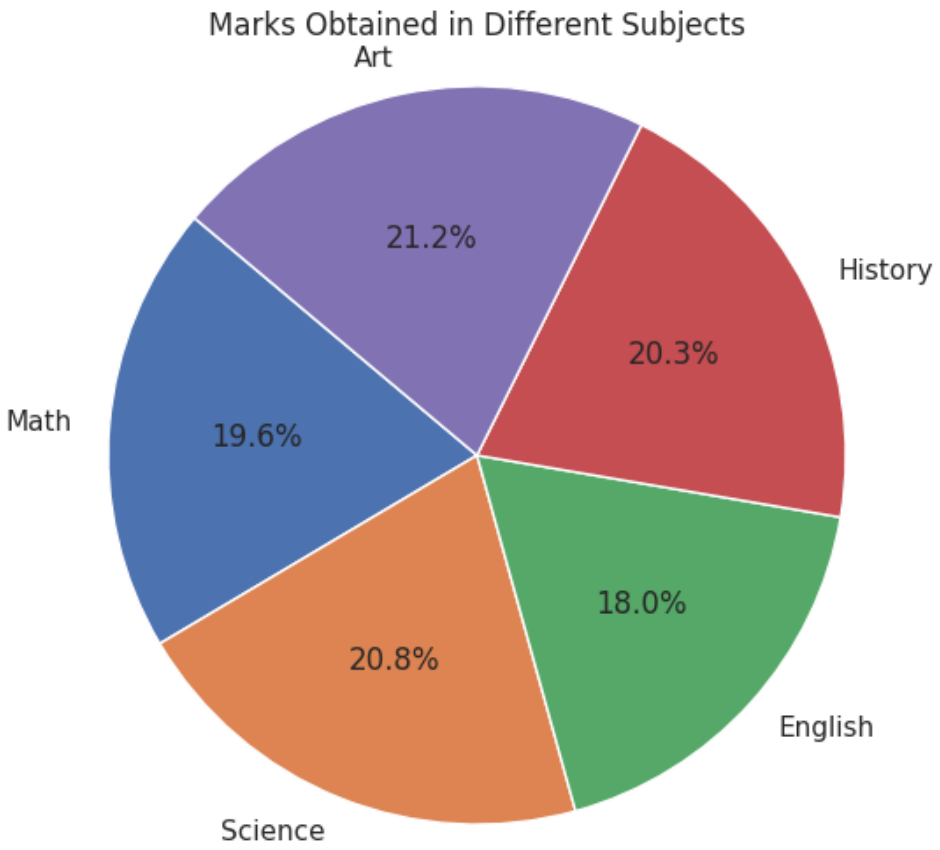
[81]:
```
import numpy as np
import matplotlib.pyplot as plt

random_integers = np.random.randint(1, 101, size=50)

plt.figure(figsize=(12, 5))

plt.subplot(1, 2, 1)
```

```
plt.plot(random_integers, color='blue', marker='o', linestyle='-', linewidth=2,␣
 ↪markersize=5)
plt.title('Line Chart of Random Integers')
plt.xlabel('Index')
plt.ylabel('Value')
plt.grid(True)

plt.subplot(1, 2, 2)
plt.scatter(range(len(random_integers)), random_integers, color='orange',␣
 ↪s=100, alpha=0.7, edgecolors='black')
plt.title('Scatter Plot of Random Integers')
plt.xlabel('Index')
plt.ylabel('Value')
plt.grid(True)

plt.tight_layout()
plt.show()
```



B) Create two lists, one representing subject names and the other representing marks obtained in those subjects. Display the data in a pie chart.

[82]:
```
import matplotlib.pyplot as plt

subjects = ['Math', 'Science', 'English', 'History', 'Art']
marks = [85, 90, 78, 88, 92]

plt.figure(figsize=(8, 6))
plt.pie(marks, labels=subjects, autopct='%1.1f%%', startangle=140)
plt.title('Marks Obtained in Different Subjects')
plt.axis('equal')
plt.show()
```



C) 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

[83]:
```
import pandas as pd

data = pd.read_csv('/home/ajay/Downloads/FDS_TEST_DATA/wineequality-red.csv')

description = data.describe()
print("Dataset Description:")
print(description)

shape = data.shape
print("\nShape of the dataset:")
print(f"Number of rows: {shape[0]}, Number of columns: {shape[1]}")

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

```
print(data.head(3))
```

```
Dataset Description:
       fixed acidity  volatile acidity  citric acid  residual sugar  \
count    1599.000000       1599.000000  1599.000000     1599.000000
mean        8.319637          0.527821     0.270976        2.538806
std         1.741096          0.179060     0.194801        1.409928
min         4.600000          0.120000     0.000000        0.900000
25%         7.100000          0.390000     0.090000        1.900000
50%         7.900000          0.520000     0.260000        2.200000
75%         9.200000          0.640000     0.420000        2.600000
max        15.900000          1.580000     1.000000       15.500000

         chlorides  free sulfur dioxide  total sulfur dioxide      density  \
count  1599.000000          1599.000000           1599.000000  1599.000000
mean      0.087467            15.874922             46.467792     0.996747
std       0.047065            10.460157             32.895324     0.001887
min       0.012000             1.000000              6.000000     0.990070
25%       0.070000             7.000000             22.000000     0.995600
50%       0.079000            14.000000             38.000000     0.996750
75%       0.090000            21.000000             62.000000     0.997835
max       0.611000            72.000000            289.000000     1.003690

                pH    sulphates      alcohol      quality
count  1599.000000  1599.000000  1599.000000  1599.000000
mean      3.311113     0.658149    10.422983     5.636023
std       0.154386     0.169507     1.065668     0.807569
min       2.740000     0.330000     8.400000     3.000000
25%       3.210000     0.550000     9.500000     5.000000
50%       3.310000     0.620000    10.200000     6.000000
75%       3.400000     0.730000    11.100000     6.000000
max       4.010000     2.000000    14.900000     8.000000

Shape of the dataset:
Number of rows: 1599, Number of columns: 12

First 3 rows of the dataset:
   fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
0            7.4              0.70         0.00             1.9      0.076
1            7.8              0.88         0.00             2.6      0.098
2            7.8              0.76         0.04             2.3      0.092

   free sulfur dioxide  total sulfur dioxide  density    pH  sulphates  \
0                 11.0                  34.0   0.9978  3.51       0.56
1                 25.0                  67.0   0.9968  3.20       0.68
2                 15.0                  54.0   0.9970  3.26       0.65

   alcohol  quality
```

```
0      9.4        5
1      9.8        5
2      9.8        5
```

---

slip 10

A) Write a python program to Display column-wise mean, and median for SOCR- HeightWeight dataset.

[84]:
```
import pandas as pd

data = pd.read_csv("/home/ajay/Downloads/FDS_TEST_DATA/heights_weights.csv")

print("First few rows of the dataset:")
print(data.head())

print("\nData Types of Each Column:")
print(data.dtypes)

numeric_columns = data.select_dtypes(include='number')

mean_values = numeric_columns.mean()
median_values = numeric_columns.median()

# Step 5: Display the results
print("\nColumn-wise Mean:")
print(mean_values)

print("\nColumn-wise Median:")
print(median_values)
```

```
First few rows of the dataset:
      Name  Height  Weight
0    Alice      65     150
1      Bob      70     180
2  Charlie      68     165
3    David      72     190
4      Eva      64     130

Data Types of Each Column:
Name      object
Height     int64
Weight     int64
dtype: object

Column-wise Mean:
Height     68.4
Weight    163.5
```

```
dtype: float64

Column-wise Median:
Height     68.0
Weight    162.5
dtype: float64
```

B) Write a python program to compute sum of Manhattan distance between all pairs of points.

[85]:
```python
import pandas as pd
from itertools import combinations

data = pd.DataFrame({
    'x': [1, 2, 3],
    'y': [4, 5, 6]
})

print("Dataset:")
print(data)

def manhattan_distance(point1, point2):
    return abs(point1.iloc[0] - point2.iloc[0]) + abs(point1.iloc[1] - point2.
 ↪iloc[1])

total_distance = 0

for (i, point1), (j, point2) in combinations(data.iterrows(), 2):
    distance = manhattan_distance(point1[['x', 'y']], point2[['x', 'y']])
    total_distance += distance
    print(f"Distance between point {i} and point {j}: {distance}")

print("\nTotal Sum of Manhattan Distances:", total_distance)
```

```
Dataset:
   x  y
0  1  4
1  2  5
2  3  6
Distance between point 0 and point 1: 2
Distance between point 0 and point 2: 4
Distance between point 1 and point 2: 2

Total Sum of Manhattan Distances: 8
```

---

slip 11

A) Write a Python program to create a Pie plot to get the frequency of the three species of the Iris data (Use iris.csv)

[86]:
```
# in slip 1 Q.2) A)
```

B) Write a Python program to view basic statistical details of the data.(Use wineequality-red.csv)

[87]:
```
# in slip 1 Q.2) B)
```

---

slip 12

A) 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.

[88]:
```
# in slip 4 Q.2) A)
```

B) 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.

[89]:
```
import pandas as pd
import numpy as np

data = {
    'Name': ['Alice', 'Bob', 'Charlie', 'David', 'Eva',
             'Frank', 'Grace', 'Hannah', 'Ian', 'Bob'],
    'Salary': [70000, 80000, 120000, np.nan, 95000,
               70000, 100000, np.nan, 85000, 80000],
    'Department': ['HR', 'Finance', 'IT', 'HR', 'Marketing',
                   'IT', 'HR', 'Finance', None, 'Marketing']
}

df = pd.DataFrame(data)

print("Original DataFrame:")
print(df)

df.dropna(inplace=True)

df.drop_duplicates(inplace=True)

print("\nModified DataFrame after dropping null and duplicate values:")
print(df)
```

```
Original DataFrame:
      Name    Salary Department
0    Alice   70000.0         HR
1      Bob   80000.0    Finance
2  Charlie  120000.0         IT
3    David       NaN         HR
4      Eva   95000.0  Marketing
```

```
5    Frank   70000.0         IT
6    Grace  100000.0         HR
7   Hannah       NaN    Finance
8      Ian   85000.0       None
9      Bob   80000.0  Marketing

Modified DataFrame after dropping null and duplicate values:
      Name    Salary Department
0    Alice   70000.0         HR
1      Bob   80000.0    Finance
2  Charlie  120000.0         IT
4      Eva   95000.0  Marketing
5    Frank   70000.0         IT
6    Grace  100000.0         HR
9      Bob   80000.0  Marketing
```

---

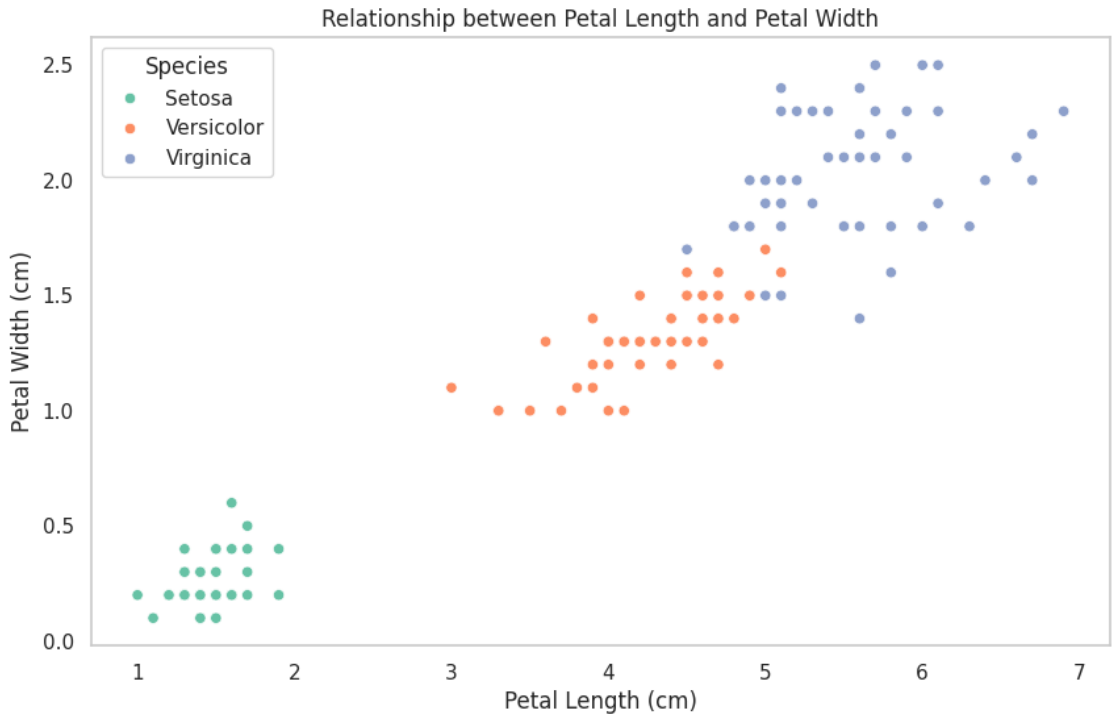
slip 13

A) Write a Python program to create a graph to find relationship between the petal length and petal width.(Use iris.csv dataset)

[90]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

iris_data = pd.read_csv("/home/ajay/Downloads/FDS_TEST_DATA/iris.csv")

print("First few rows of the dataset:")
print(iris_data.head())

plt.figure(figsize=(10, 6))
sns.scatterplot(data=iris_data, x='petal.length', y='petal.width',␣
 ↪hue='variety', palette='Set2')

plt.title('Relationship between Petal Length and Petal Width')
plt.xlabel('Petal Length (cm)')
plt.ylabel('Petal Width (cm)')

plt.legend(title='Species')
plt.grid()
plt.show()
```

```
First few rows of the dataset:
   sepal.length  sepal.width  petal.length  petal.width variety
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
```



B) Write a Python program to find the maximum and minimum value of a given flattened array.

[91]:
```
import numpy as np

array_2d = np.array([[10, 20, 30],
                     [40, 50, 60],
                     [70, 80, 90]])

flattened_array = array_2d.flatten()

max_value = np.max(flattened_array)
min_value = np.min(flattened_array)

print("Flattened Array:", flattened_array)
print("Maximum Value:", max_value)
print("Minimum Value:", min_value)
```

```
Flattened Array: [10 20 30 40 50 60 70 80 90]
Maximum Value: 90
Minimum Value: 10
```

slip 14

A) Write a Python NumPy program to compute the weighted average along the specified axis of a given flattened array.

[92]:
```
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("Data Array:", data)
print("Weights:", weights)
print("Weighted Average:", weighted_average)
```

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

B) Write a Python program to view basic statistical details of the data (Use advertising.csv)

[93]:
```
import pandas as pd

advertising_data = pd.read_csv("/home/ajay/Downloads/FDS_TEST_DATA/advertising.
 ↪csv")

print("First few rows of the dataset:")
print(advertising_data.head())

print("\nBasic Statistical Details:")
print(advertising_data.describe())
```

```
First few rows of the dataset:
   RND     TV  Radio  Newspaper  Sales
0    1  230.1   37.8       69.0   22.1
1    2   44.5   39.3       45.0   10.4
2    3   17.2   45.9       69.0   12.0
3    4  151.5   41.3       58.0   16.5
4    5  180.8   10.8       58.0   17.0

Basic Statistical Details:
            RND          TV       Radio  Newspaper       Sales
count  50.00000   50.000000  50.000000  50.000000  50.000000
mean   25.50000   97.412000  26.762000  34.880000  14.340000
std    14.57738   52.712658   9.672893  21.149265   4.471314
min     1.00000    5.500000   8.000000   2.000000   4.000000
25%    13.25000   63.425000  20.050000  20.000000  11.050000
50%    25.50000   90.000000  25.500000  29.500000  14.300000
```

```
75%     37.75000  139.825000  34.175000  48.000000  17.900000
max     50.00000  230.100000  48.900000  94.000000  22.500000
```

---

slip 15

A) 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.

[94]:
```
# in slip 4 Q.2) A)
```

B) Create two lists, one representing subject names and the other representing marks obtained in those subjects. Display the data in a pie chart.

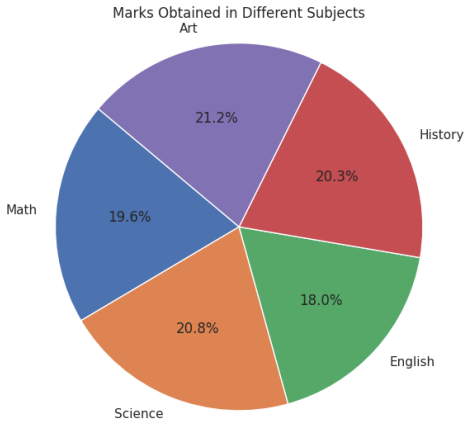
[95]:
```
# in slip 9 Q.2) B)
```

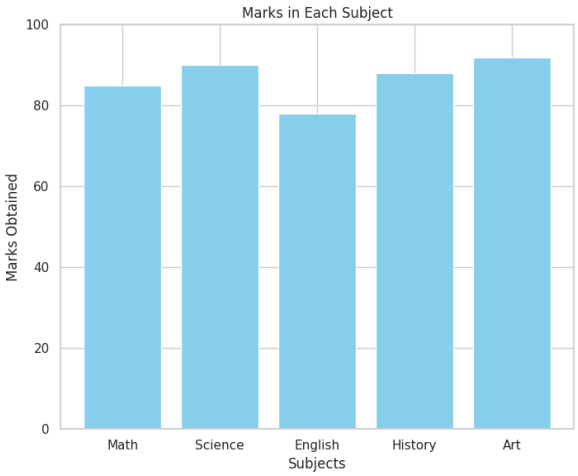
---

slip 16

A) 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.

[96]:
```
import matplotlib.pyplot as plt

subjects = ['Math', 'Science', 'English', 'History', 'Art']
marks = [85, 90, 78, 88, 92]

plt.figure(figsize=(14, 6))

plt.subplot(1, 2, 1)
plt.pie(marks, labels=subjects, autopct='%1.1f%%', startangle=140)
plt.title('Marks Obtained in Different Subjects')
plt.axis('equal')

plt.subplot(1, 2, 2)
plt.bar(subjects, marks, color='skyblue')
plt.title('Marks in Each Subject')
plt.xlabel('Subjects')
plt.ylabel('Marks Obtained')
plt.ylim(0, 100)

plt.tight_layout()
plt.show()
```





B) 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.

[97]:
```
import pandas as pd

data = {
    'Name': ['Alice', 'Bob', 'Charlie', 'David', 'Eva'],
    'Graduation Percentage': [85.5, 78.0, 92.0, 88.5, 90.0],
    'Age': [22, 23, 21, 24, 22]
}

students_df = pd.DataFrame(data)

print("Students Information:")
print(students_df)

average_age = students_df['Age'].mean()
average_graduation_percentage = students_df['Graduation Percentage'].mean()

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

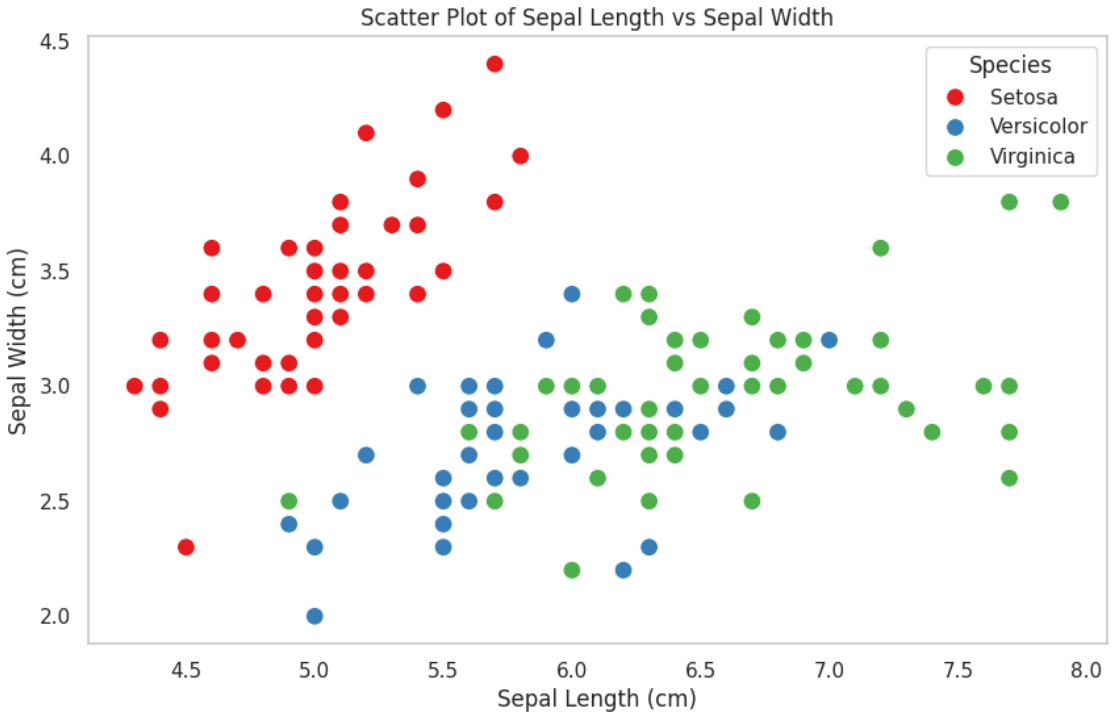
```
Students Information:
      Name  Graduation Percentage  Age
0    Alice                   85.5   22
1      Bob                   78.0   23
2  Charlie                   92.0   21
3    David                   88.5   24
4      Eva                   90.0   22

Average Age of Students: 22.40 years
Average Graduation Percentage: 86.80%
```

slip 17

A) Write a Python program to draw scatter plots to compare two features of the iris dataset

[98]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

file_path = '/home/ajay/Downloads/FDS_TEST_DATA/iris.csv'
iris_data = pd.read_csv(file_path)

print("First few rows of the dataset:")
print(iris_data.head())

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

sns.scatterplot(x='sepal.length', y='sepal.width', hue='variety',␣
 ↪data=iris_data, palette='Set1', s=100)

plt.title('Scatter Plot of Sepal Length vs Sepal Width')
plt.xlabel('Sepal Length (cm)')
plt.ylabel('Sepal Width (cm)')
plt.grid()

plt.legend(title='Species')
plt.show()
```

```
First few rows of the dataset:
   sepal.length  sepal.width  petal.length  petal.width variety
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
```



B) 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.

[99]:
```
import pandas as pd

data = {
    'Name': [
        'Alice', 'Bob', 'Charlie', 'David', 'Eva',
        'Frank', 'Grace', 'Hannah', 'Ian', 'Jack'
    ],
    'Age': [28, 34, 29, 42, 36, 30, 27, 40, 38, 33],
    'Salary': [70000, 80000, 75000, 120000, 95000,
               60000, 85000, 90000, 65000, 72000],
    'Department': [
        'HR', 'Finance', 'IT', 'Marketing', 'IT',
        'HR', 'Finance', 'Marketing', 'IT', 'HR'
    ]
}

employees_df = pd.DataFrame(data)

print("Employee DataFrame:")
print(employees_df)
```

```
Employee DataFrame:
```

```
      Name  Age  Salary Department
0    Alice   28   70000         HR
1      Bob   34   80000    Finance
2  Charlie   29   75000         IT
3    David   42  120000  Marketing
4      Eva   36   95000         IT
5    Frank   30   60000         HR
6    Grace   27   85000    Finance
7   Hannah   40   90000  Marketing
8      Ian   38   65000         IT
9     Jack   33   72000         HR
```

---

slip 18

A) 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)

[100]:
```
# in slip 3 Q.2) A)
```

B) Use the heights and weights dataset and load the dataset from a given csv file into a dataframe. Print the first, last 5 rows and random 10 row

[101]:
```
import pandas as pd

file_path = '/home/ajay/Downloads/FDS_TEST_DATA/heights_weights.csv'
data = pd.read_csv(file_path)

print("First 5 rows of the dataset:")
print(data.head())

print("\nLast 5 rows of the dataset:")
print(data.tail())

print("\nRandom 10 rows of the dataset:")
print(data.sample(n=10))
```

```
First 5 rows of the dataset:
      Name  Height  Weight
0    Alice      65     150
1      Bob      70     180
2  Charlie      68     165
3    David      72     190
4      Eva      64     130

Last 5 rows of the dataset:
     Name  Height  Weight
15  Quinn      65     145
```

```
16   Rita      73     195
17  Steve      70     185
18   Tina      66     150
19    Uma      64     130

Random 10 rows of the dataset:
       Name  Height  Weight
10     Liam      66     155
14     Paul      76     210
11      Mia      71     175
0     Alice      65     150
4       Eva      64     130
2   Charlie      68     165
19      Uma      64     130
18     Tina      66     150
8       Ian      74     185
9      Jack      69     170
```

---

slip 19

Write a Python program 1. To create a dataframe containing columns name, age and percentage. Add 10 rows to the dataframe. View the dataframe. 2. To print the shape, number of rows-columns, data types, feature names and the description of the data 3. To Add 5 rows with duplicate values and missing values. Add a column 'remarks' with empty values. Display the data.

[102]:
```python
import pandas as pd
import numpy as np

data = {
    'Name': [
        'Alice', 'Bob', 'Charlie', 'David', 'Eva',
        'Frank', 'Grace', 'Hannah', 'Ian', 'Jack'
    ],
    'Age': [24, 30, 22, 29, 31, 25, 28, 27, 35, 26],
    'Percentage': [85.5, 90.0, 78.5, 88.0, 92.0,
                   80.0, 95.5, 82.0, 89.5, 76.0]
}

df = pd.DataFrame(data)

print("DataFrame with Name, Age, and Percentage:")
print(df)

print("\nShape of the DataFrame:", df.shape)
print("Data Types:")
print(df.dtypes)
print("Feature Names:")
```

```
print(df.columns)
print("\nDescription of the Data:")
print(df.describe())

duplicate_rows = pd.DataFrame({
    'Name': ['Alice', 'Bob', 'Charlie', 'David', None],
    'Age': [24, 30, 22, 29, None],
    'Percentage': [85.5, 90.0, 78.5, 88.0, None]
})

df = pd.concat([df, duplicate_rows], ignore_index=True)

df['Remarks'] = ''

print("\nUpdated DataFrame with duplicates and missing values:")
print(df)
```

```
DataFrame with Name, Age, and Percentage:
      Name  Age  Percentage
0    Alice   24        85.5
1      Bob   30        90.0
2  Charlie   22        78.5
3    David   29        88.0
4      Eva   31        92.0
5    Frank   25        80.0
6    Grace   28        95.5
7   Hannah   27        82.0
8      Ian   35        89.5
9     Jack   26        76.0

Shape of the DataFrame: (10, 3)
Data Types:
Name           object
Age             int64
Percentage    float64
dtype: object
Feature Names:
Index(['Name', 'Age', 'Percentage'], dtype='object')

Description of the Data:
             Age  Percentage
count  10.000000   10.000000
mean   27.700000   85.700000
std     3.772709    6.377913
min    22.000000   76.000000
25%    25.250000   80.500000
50%    27.500000   86.750000
75%    29.750000   89.875000
```

```
max    35.000000   95.500000

Updated DataFrame with duplicates and missing values:
       Name   Age  Percentage Remarks
0     Alice  24.0        85.5
1       Bob  30.0        90.0
2   Charlie  22.0        78.5
3     David  29.0        88.0
4       Eva  31.0        92.0
5     Frank  25.0        80.0
6     Grace  28.0        95.5
7    Hannah  27.0        82.0
8       Ian  35.0        89.5
9      Jack  26.0        76.0
10    Alice  24.0        85.5
11      Bob  30.0        90.0
12  Charlie  22.0        78.5
13    David  29.0        88.0
14     None   NaN         NaN
```

---

slip 20

A) 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.

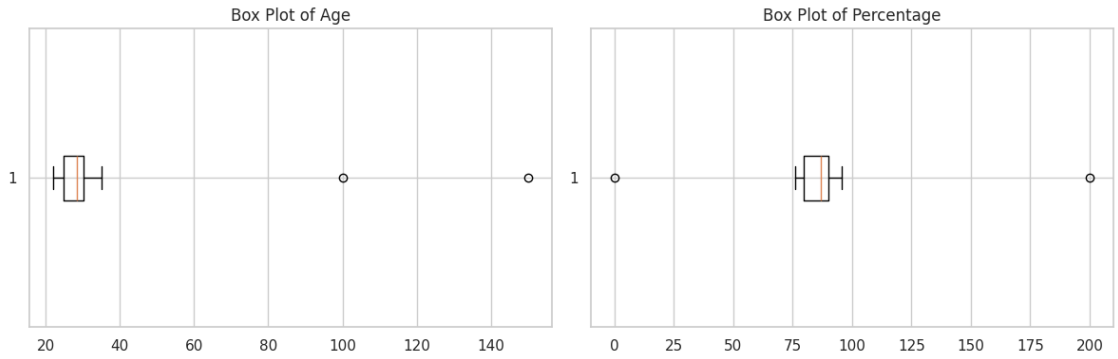
[103]:
```
# in slip 4 Q.2) A)
```

B) Add two outliers to the above data and display the box plot.

[104]:
```
import pandas as pd
import matplotlib.pyplot as plt

data = {
    'Name': ['Alice', 'Bob', 'Charlie', 'David', 'Eva', 'Frank', 'Grace',␣
 ↪'Hannah', 'Ian', 'Jack'],
    'Age': [24, 30, 22, 29, 31, 25, 28, 27, 35, 26],
    'Percentage': [85.5, 90.0, 78.5, 88.0, 92.0, 80.0, 95.5, 82.0, 89.5, 76.0]
}

df = pd.DataFrame(data)

df = pd.concat([
    df,
    pd.DataFrame({'Name': ['Alice', 'Bob', 'Charlie', 'David', None],
                  'Age': [24, 30, 22, 29, None],
                  'Percentage': [85.5, 90.0, 78.5, 88.0, None]}),
    pd.DataFrame({'Name': ['Outlier1', 'Outlier2'],
```

```
                    'Age': [100, 150],
                    'Percentage': [0, 200]})
], ignore_index=True)

print("Updated DataFrame:")
print(df)

plt.figure(figsize=(12, 4))
plt.subplot(1, 2, 1)
plt.boxplot(df['Age'].dropna(), vert=False)
plt.title('Box Plot of Age')

plt.subplot(1, 2, 2)
plt.boxplot(df['Percentage'].dropna(), vert=False)
plt.title('Box Plot of Percentage')

plt.tight_layout()
plt.show()
```

```
Updated DataFrame:
         Name    Age  Percentage
0       Alice   24.0        85.5
1         Bob   30.0        90.0
2     Charlie   22.0        78.5
3       David   29.0        88.0
4         Eva   31.0        92.0
5       Frank   25.0        80.0
6       Grace   28.0        95.5
7      Hannah   27.0        82.0
8         Ian   35.0        89.5
9        Jack   26.0        76.0
10      Alice   24.0        85.5
11        Bob   30.0        90.0
12    Charlie   22.0        78.5
13      David   29.0        88.0
14       None    NaN         NaN
15   Outlier1  100.0         0.0
16   Outlier2  150.0       200.0
```

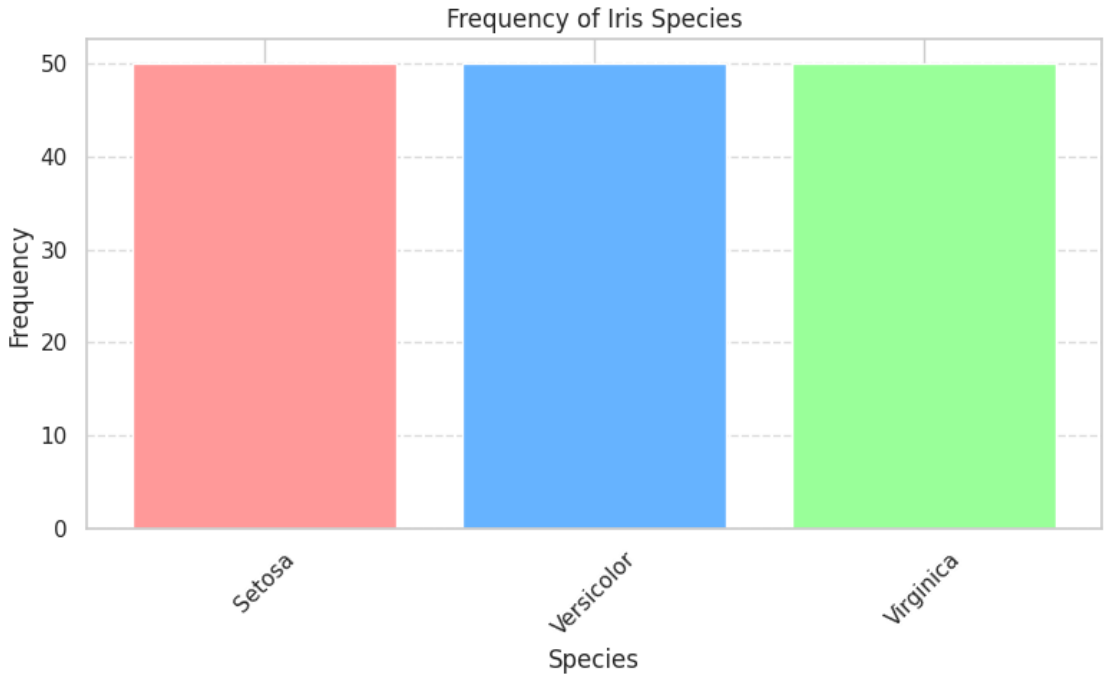


---

slip 21

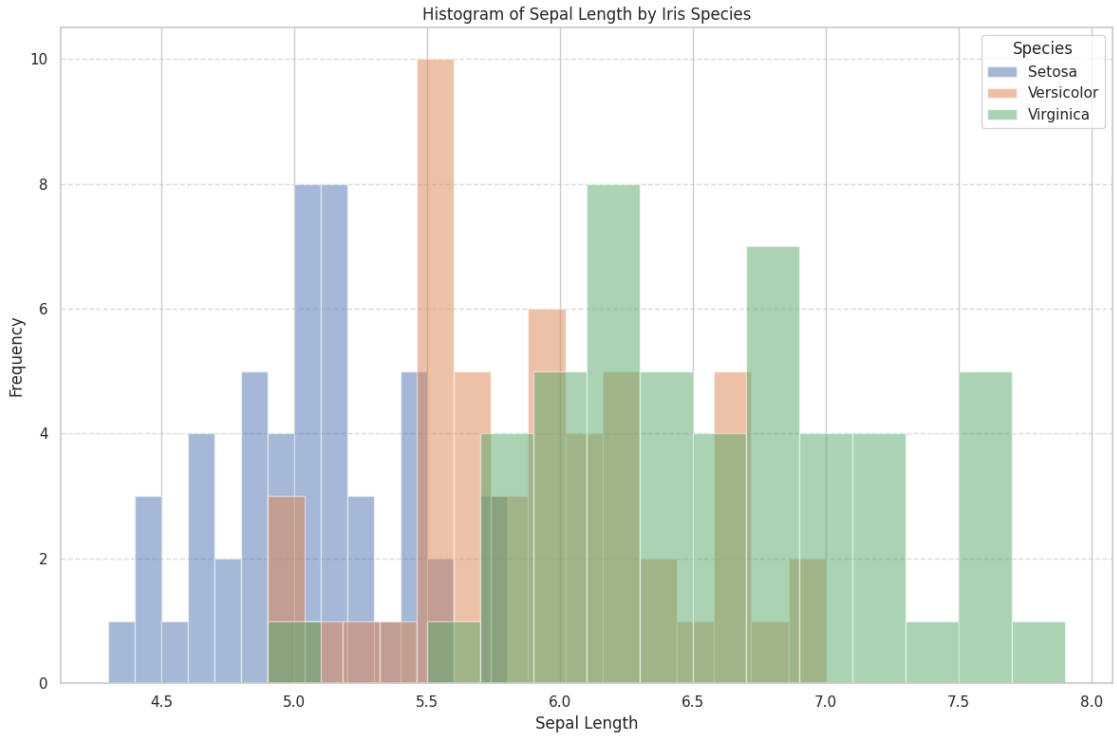
A) Import dataset “iris.csv”. Write a Python program to create a Bar plot to get the frequency of the three species of the Iris data.

[105]:
```
import pandas as pd
import matplotlib.pyplot as plt

iris_data = pd.read_csv("/home/ajay/Downloads/FDS_TEST_DATA/iris.csv")

species_count = iris_data['variety'].value_counts()

plt.figure(figsize=(8, 5))
plt.bar(species_count.index, species_count.values, color=['#ff9999', '#66b3ff',␣
↪'#99ff99'])
plt.title('Frequency of Iris Species')
plt.xlabel('Species')
plt.ylabel('Frequency')
plt.xticks(rotation=45)
plt.grid(axis='y', linestyle='--', alpha=0.7)

plt.tight_layout()
plt.show()
```



B)Write a Python program to create a histogram of the three species of the Iris data.

[106]:
```
import pandas as pd
import matplotlib.pyplot as plt

iris_data = pd.read_csv("/home/ajay/Downloads/FDS_TEST_DATA/iris.csv")

plt.figure(figsize=(12, 8))

for species in iris_data['variety'].unique():
    subset = iris_data[iris_data['variety'] == species]
    plt.hist(subset['sepal.length'], bins=15, alpha=0.5, label=species)

plt.title('Histogram of Sepal Length by Iris Species')
plt.xlabel('Sepal Length')
plt.ylabel('Frequency')
plt.legend(title='Species')
plt.grid(axis='y', linestyle='--', alpha=0.7)

plt.tight_layout()
plt.show()
```



---

slip 22

Dataset Name: winequality-red.csv Write a program in python to perform following tasks a. Rescaling: Normalised the dataset using MinMaxScaler class b. Standardizing Data (transform them into a standard Gaussian distribution with a mean of 0 and a standard deviation of 1) c. Normalizing Data (rescale each observation to a length of 1 (a unit norm). For this, use the Normalizer class.)

[107]:
```
import pandas as pd
from sklearn.preprocessing import MinMaxScaler, StandardScaler, Normalizer

wine_data = pd.read_csv("/home/ajay/Downloads/FDS_TEST_DATA/wineequality-red.
 ↪csv")

print("Original Dataset:")
print(wine_data.head())

minmax_scaler = MinMaxScaler()
wine_data_minmax = minmax_scaler.fit_transform(wine_data)

print("\nNormalized Dataset using MinMaxScaler:")
print(pd.DataFrame(wine_data_minmax, columns=wine_data.columns).head())

standard_scaler = StandardScaler()
```

```
wine_data_standardized = standard_scaler.fit_transform(wine_data)

print("\nStandardized Dataset:")
print(pd.DataFrame(wine_data_standardized, columns=wine_data.columns).head())

normalizer = Normalizer()
wine_data_normalized = normalizer.fit_transform(wine_data)

print("\nNormalized Dataset using Normalizer:")
print(pd.DataFrame(wine_data_normalized, columns=wine_data.columns).head())
```

```
Original Dataset:
   fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
0            7.4              0.70         0.00             1.9      0.076
1            7.8              0.88         0.00             2.6      0.098
2            7.8              0.76         0.04             2.3      0.092
3           11.2              0.28         0.56             1.9      0.075
4            7.4              0.70         0.00             1.9      0.076

   free sulfur dioxide  total sulfur dioxide  density    pH  sulphates  \
0                 11.0                  34.0   0.9978  3.51       0.56
1                 25.0                  67.0   0.9968  3.20       0.68
2                 15.0                  54.0   0.9970  3.26       0.65
3                 17.0                  60.0   0.9980  3.16       0.58
4                 11.0                  34.0   0.9978  3.51       0.56

   alcohol  quality
0      9.4        5
1      9.8        5
2      9.8        5
3      9.8        6
4      9.4        5

Normalized Dataset using MinMaxScaler:
   fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
0       0.247788          0.397260         0.00        0.068493   0.106845
1       0.283186          0.520548         0.00        0.116438   0.143573
2       0.283186          0.438356         0.04        0.095890   0.133556
3       0.584071          0.109589         0.56        0.068493   0.105175
4       0.247788          0.397260         0.00        0.068493   0.106845

   free sulfur dioxide  total sulfur dioxide   density        pH  sulphates  \
0             0.140845              0.098940  0.567548  0.606299   0.137725
1             0.338028              0.215548  0.494126  0.362205   0.209581
2             0.197183              0.169611  0.508811  0.409449   0.191617
3             0.225352              0.190813  0.582232  0.330709   0.149701
4             0.140845              0.098940  0.567548  0.606299   0.137725
```

```
    alcohol  quality
0  0.153846      0.4
1  0.215385      0.4
2  0.215385      0.4
3  0.215385      0.6
4  0.153846      0.4

Standardized Dataset:
   fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
0      -0.528360          0.961877    -1.391472       -0.453218  -0.243707
1      -0.298547          1.967442    -1.391472        0.043416   0.223875
2      -0.298547          1.297065    -1.186070       -0.169427   0.096353
3       1.654856         -1.384443     1.484154       -0.453218  -0.264960
4      -0.528360          0.961877    -1.391472       -0.453218  -0.243707

   free sulfur dioxide  total sulfur dioxide   density        pH  sulphates  \
0            -0.466193             -0.379133  0.558274  1.288643  -0.579207
1             0.872638              0.624363  0.028261 -0.719933   0.128950
2            -0.083669              0.229047  0.134264 -0.331177  -0.048089
3             0.107592              0.411500  0.664277 -0.979104  -0.461180
4            -0.466193             -0.379133  0.558274  1.288643  -0.579207

    alcohol   quality
0 -0.960246 -0.787823
1 -0.584777 -0.787823
2 -0.584777 -0.787823
3 -0.584777  0.450848
4 -0.960246 -0.787823

Normalized Dataset using Normalizer:
   fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
0       0.193478          0.018302     0.000000        0.049677   0.001987
1       0.106989          0.012071     0.000000        0.035663   0.001344
2       0.134949          0.013149     0.000692        0.039793   0.001592
3       0.173611          0.004340     0.008681        0.029452   0.001163
4       0.193478          0.018302     0.000000        0.049677   0.001987

   free sulfur dioxide  total sulfur dioxide   density        pH  sulphates  \
0             0.287602              0.888952  0.026088  0.091771   0.014642
1             0.342913              0.919006  0.013673  0.043893   0.009327
2             0.259517              0.934261  0.017249  0.056402   0.011246
3             0.263517              0.930059  0.015470  0.048983   0.008991
4             0.287602              0.888952  0.026088  0.091771   0.014642

    alcohol   quality
0  0.245769  0.130728
1  0.134422  0.068583
2  0.169551  0.086506
```

```
3  0.151910  0.093006
4  0.245769  0.130728
```

Dataset Name: winequality-red.csv Write a program in python to perform following task a. Rescaling: Normalised the dataset using MinMaxScaler class b. Standardizing Data (transform them into a standard Gaussian distribution with a mean of 0 and a standard deviation of 1) c. Binarizing Data using we use the Binarizer class (Using a binary threshold, it is possible to transform our data by marking the values above it 1 and those equal to or below it, 0)

[108]:
```
import pandas as pd
from sklearn.preprocessing import MinMaxScaler, StandardScaler, Binarizer

wine_data = pd.read_csv("/home/ajay/Downloads/FDS_TEST_DATA/wineequality-red.
 ↪csv")

print("Original Dataset:")
print(wine_data.head())

minmax_scaler = MinMaxScaler()
wine_data_minmax = minmax_scaler.fit_transform(wine_data)

print("\nNormalized Dataset using MinMaxScaler:")
print(pd.DataFrame(wine_data_minmax, columns=wine_data.columns).head())

standard_scaler = StandardScaler()
wine_data_standardized = standard_scaler.fit_transform(wine_data)

print("\nStandardized Dataset:")
print(pd.DataFrame(wine_data_standardized, columns=wine_data.columns).head())

threshold = 0.5
binarizer = Binarizer(threshold=threshold)
wine_data_binarized = binarizer.fit_transform(wine_data)

print("\nBinarized Dataset:")
print(pd.DataFrame(wine_data_binarized, columns=wine_data.columns).head())
```

```
Original Dataset:
   fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
0            7.4              0.70         0.00             1.9      0.076
1            7.8              0.88         0.00             2.6      0.098
2            7.8              0.76         0.04             2.3      0.092
3           11.2              0.28         0.56             1.9      0.075
4            7.4              0.70         0.00             1.9      0.076

   free sulfur dioxide  total sulfur dioxide  density    pH  sulphates  \
0                 11.0                  34.0   0.9978  3.51       0.56
```

```
1                25.0                 67.0   0.9968  3.20       0.68
2                15.0                 54.0   0.9970  3.26       0.65
3                17.0                 60.0   0.9980  3.16       0.58
4                11.0                 34.0   0.9978  3.51       0.56

   alcohol  quality
0      9.4        5
1      9.8        5
2      9.8        5
3      9.8        6
4      9.4        5

Normalized Dataset using MinMaxScaler:
   fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
0       0.247788          0.397260         0.00        0.068493   0.106845
1       0.283186          0.520548         0.00        0.116438   0.143573
2       0.283186          0.438356         0.04        0.095890   0.133556
3       0.584071          0.109589         0.56        0.068493   0.105175
4       0.247788          0.397260         0.00        0.068493   0.106845

   free sulfur dioxide  total sulfur dioxide   density        pH  sulphates  \
0             0.140845              0.098940  0.567548  0.606299   0.137725
1             0.338028              0.215548  0.494126  0.362205   0.209581
2             0.197183              0.169611  0.508811  0.409449   0.191617
3             0.225352              0.190813  0.582232  0.330709   0.149701
4             0.140845              0.098940  0.567548  0.606299   0.137725

    alcohol  quality
0  0.153846      0.4
1  0.215385      0.4
2  0.215385      0.4
3  0.215385      0.6
4  0.153846      0.4

Standardized Dataset:
   fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
0      -0.528360          0.961877    -1.391472       -0.453218  -0.243707
1      -0.298547          1.967442    -1.391472        0.043416   0.223875
2      -0.298547          1.297065    -1.186070       -0.169427   0.096353
3       1.654856         -1.384443     1.484154       -0.453218  -0.264960
4      -0.528360          0.961877    -1.391472       -0.453218  -0.243707

   free sulfur dioxide  total sulfur dioxide   density        pH  sulphates  \
0            -0.466193             -0.379133  0.558274  1.288643  -0.579207
1             0.872638              0.624363  0.028261 -0.719933   0.128950
2            -0.083669              0.229047  0.134264 -0.331177  -0.048089
3             0.107592              0.411500  0.664277 -0.979104  -0.461180
4            -0.466193             -0.379133  0.558274  1.288643  -0.579207
```

```
    alcohol   quality
0 -0.960246 -0.787823
1 -0.584777 -0.787823
2 -0.584777 -0.787823
3 -0.584777  0.450848
4 -0.960246 -0.787823

Binarized Dataset:
   fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
0            1.0               1.0          0.0             1.0        0.0
1            1.0               1.0          0.0             1.0        0.0
2            1.0               1.0          0.0             1.0        0.0
3            1.0               0.0          1.0             1.0        0.0
4            1.0               1.0          0.0             1.0        0.0

   free sulfur dioxide  total sulfur dioxide  density   pH  sulphates  \
0                  1.0                   1.0      1.0  1.0        1.0
1                  1.0                   1.0      1.0  1.0        1.0
2                  1.0                   1.0      1.0  1.0        1.0
3                  1.0                   1.0      1.0  1.0        1.0
4                  1.0                   1.0      1.0  1.0        1.0

   alcohol  quality
0      1.0      1.0
1      1.0      1.0
2      1.0      1.0
3      1.0      1.0
4      1.0      1.0
```

---

slip 24

A) 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.

[109]:
```
# in slip 21 Q.2) A)
```

B) Write a Python program to create a histogram of the three species of the Iris data.

[110]:
```
# in slip 21 Q.2) B)
```

---

slip 25

A) 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.

[111]:
```
# in slip 4 Q.2) A)
```

B) Create two lists, one representing subject names and the other representing marks obtained in those subjects. Display the data in a pie chart.

[112]:
```
# in slip 9 Q.2) B)
```

---

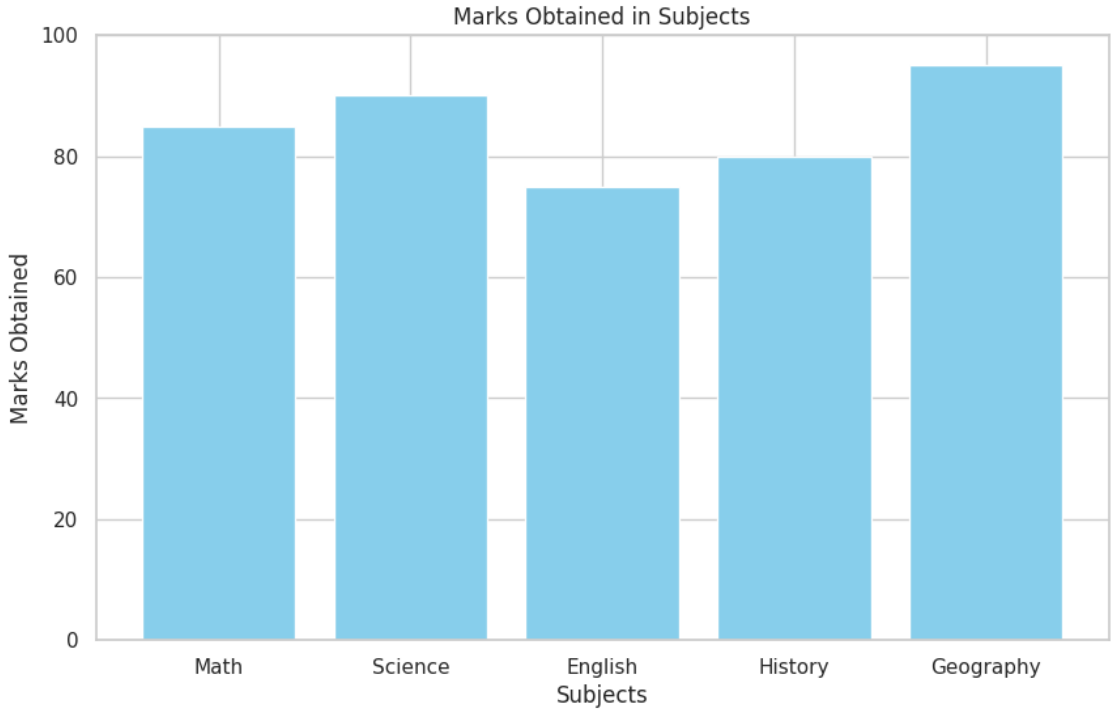
slip 26

A) 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.

[113]:
```
# in slip 4 Q.2) A)
```

B) Create two lists, one representing subject names and the other representing marks obtained in those subjects. Display the data in bar chart

[121]:
```
import matplotlib.pyplot as plt

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

plt.figure(figsize=(10, 6))
plt.bar(subjects, marks, color='skyblue')

plt.xlabel('Subjects')
plt.ylabel('Marks Obtained')
plt.title('Marks Obtained in Subjects')
plt.ylim(0, 100)

plt.show()
```



---

slip 27

Create a dataset data.csv having two categorical column (the country column, and the purchased column). a. Apply OneHot coding on Country column. b. Apply Label encoding on purchased column

[115]:
```
# in slip 7 Q.2)
```

---

slip 28

Write a Python program 1. To create a dataframe containing columns name, age and percentage. Add 10 rows to the dataframe. View the dataframe. 2. To print the shape, number of rows-columns, data types, feature names and the description of the data. 3. To view basic statistical details of the data. 4. To Add 5 rows with duplicate values and missing values. Add a column 'remarks' with empty values. Display the data.

[116]:
```
# in slip 19 Q.2)
```

---

slip 29

Create a dataset data.csv having two categorical column (the country column, and the purchased column). 1. Apply OneHot coding on Country column. 2. Apply Label encoding on purchased

column

[117]:
```
# in slip 7 Q.2)
```

---

slip 30

a. 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.

[118]:
```
# in slip 4 Q.2) A)
```

b. Create two lists, one representing subject names and the other representing marks obtained in those subjects. Display the data in bar chart.

[123]:
```
# in slip 26 Q.2) B)
```

[ ]:
```

```