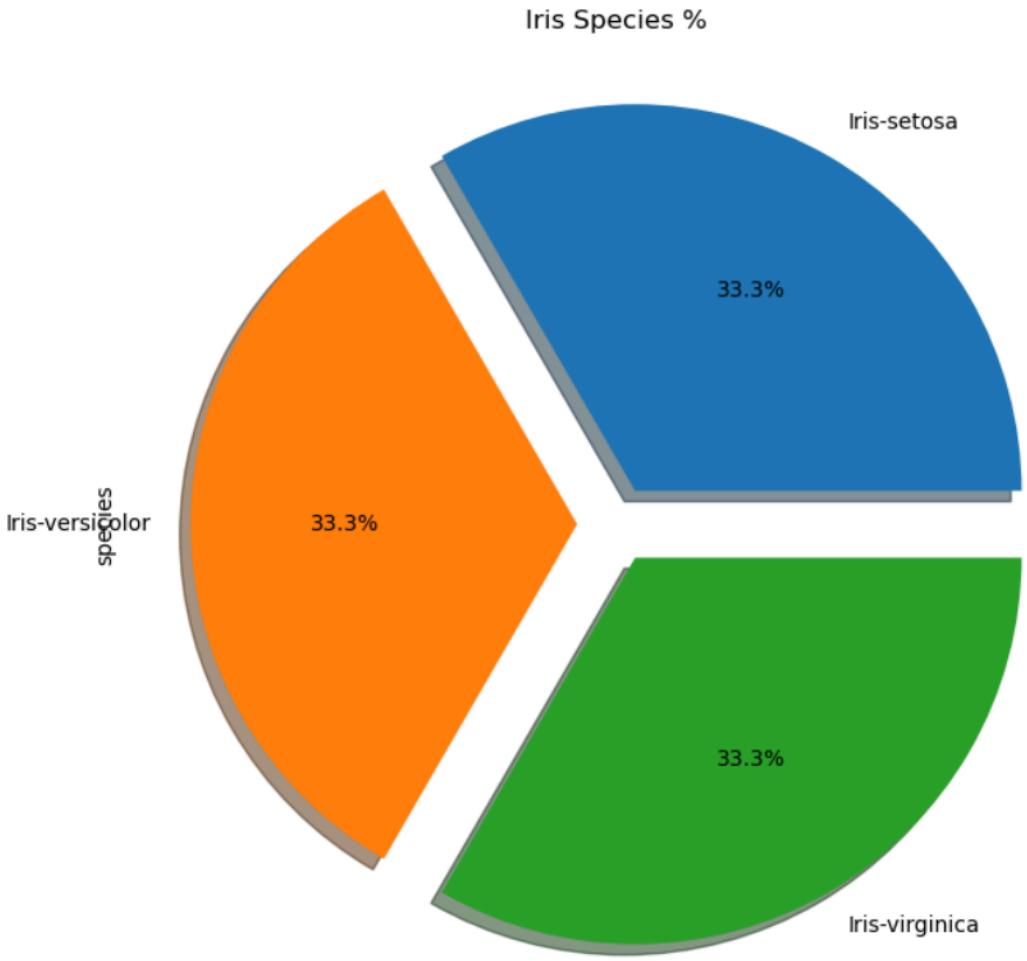
fds-solved-slips

November 26, 2023

[82]:
```
#SLIP 1 A
import pandas as pd
import matplotlib.pyplot as plt
iris = pd.read_csv("IRIS.csv")
ax=plt.subplots(1,1,figsize=(10,8))
iris['species'].value_counts().plot.pie(explode=[0.1,0.1,0.1],
autopct='%1.1f%%',shadow=True,figsize=(10,8))
plt.title("Iris Species %")
plt.show()
```



[23]:
```
#SLIP 1 B
import pandas as pd
data = pd.read_csv("winequality-red.csv ")
print(data.describe())
```

```
       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  \
```

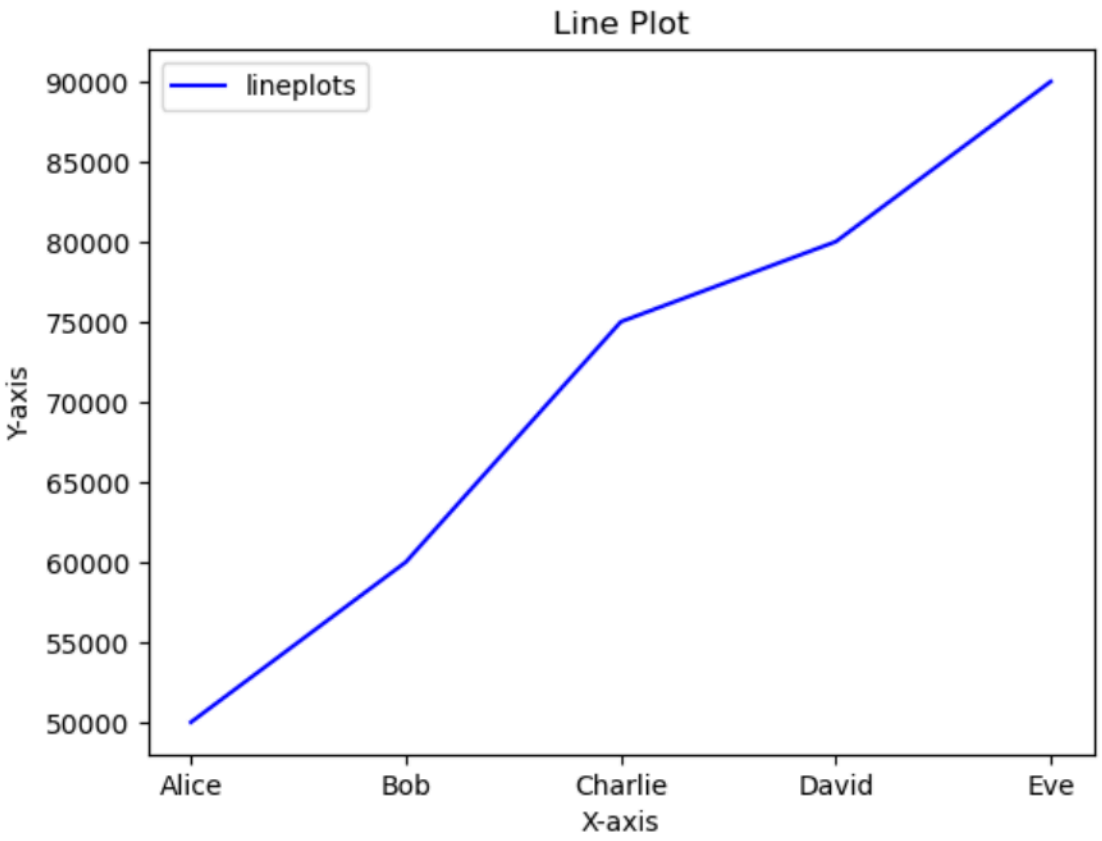
```
count  1599.000000          1599.000000          1599.000000  1599.000000
mean      0.087467            15.874922            46.467792     0.996747
std       0.047065            10.460157            32.895324     0.001887
min       0.012000             1.000000             6.000000     0.990070
25%       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
```

[40]:
```
#SLIP 2 A
import pandas as pd
data = pd.read_csv('Data.csv')
data['Salary'] = data['Salary'].fillna(data['Salary'].mean())
data['Age'] = data['Age'].fillna(data['Age'].mean())
print(data)
```

```
      Country        Age        Salary Purchased
0       India  34.000000  92000.000000       Yes
1   Sri lanka  22.000000  25000.000000       Yes
2       China  31.000000  74000.000000       Yes
3   Sri lanka  29.000000  54857.142857        No
4       China  55.000000  98000.000000       Yes
5       India  24.000000  30000.000000        No
6   Sri lanka  28.000000  40000.000000        No
7       India  33.714286  60000.000000        No
8       China  51.000000  89000.000000       Yes
9       India  44.000000  78000.000000       Yes
10  Sri lanka  21.000000  20000.000000        No
11      China  25.000000  30000.000000       Yes
12      India  33.000000  45000.000000       Yes
13      India  42.000000  65000.000000       Yes
14  Sri lanka  33.000000  22000.000000        No
```

[61]:
```
#SLIP 2 B
import matplotlib.pyplot as plt
name = ["Alice", "Bob","Charlie","David","Eve"]
salary = [50000, 60000, 75000, 80000, 90000]
```

```
plt.plot(name,salary, label='lineplots', color='b')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Line Plot')
plt.legend()
plt.show()


#ALTERNATIVE CODE WITH SAME OUTPUT


import matplotlib.pyplot as plt
names = ["Alice", "Bob", "Charlie", "David", "Eve"]
salaries = [50000, 60000, 75000, 80000, 90000]
plt.figure(figsize=(8, 6))
plt.plot(names, salaries, marker='o', linestyle='-')
plt.title("Salary vs Name")
plt.xlabel("Name")
plt.ylabel("Salary")
plt.grid(True)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```


Line Plot
lineplots
90000
85000
80000
75000
70000
65000
60000
55000
50000
Y-axis
Alice
Bob
Charlie
David
Eve
X-axis

[22]:
```python
#SLIP 2 C
import pandas as pd
file_path = "weight-height.csv"
df = pd.read_csv(file_path)
print("First 10 rows:")
print(df.head(10))
print("\nLast 10 rows:")
print(df.tail(10))
print("\nRandom 20 rows:")
print(df.sample(20))
print("\nDataset shape:", df.shape)
```

```
First 10 rows:
  Gender     Height      Weight
0   Male  73.847017  241.893563
1   Male  68.781904  162.310473
2   Male  74.110105  212.740856
3   Male  71.730978  220.042470
4   Male  69.881796  206.349801
5   Male  67.253016  152.212156
6   Male  68.785081  183.927889
```

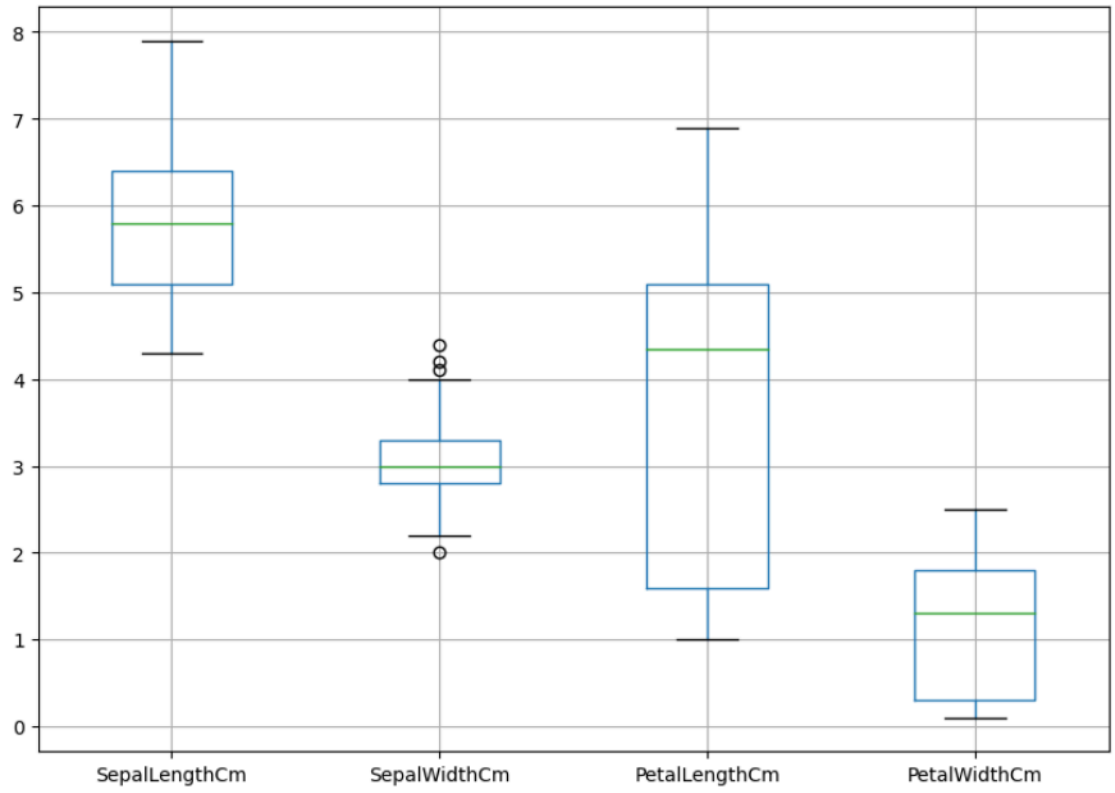
```
7   Male  68.348516  167.971110
8   Male  67.018950  175.929440
9   Male  63.456494  156.399676

Last 10 rows:
      Gender     Height      Weight
9990  Female  63.179498  141.266100
9991  Female  62.636675  102.853563
9992  Female  62.077832  138.691680
9993  Female  60.030434   97.687432
9994  Female  59.098250  110.529686
9995  Female  66.172652  136.777454
9996  Female  67.067155  170.867906
9997  Female  63.867992  128.475319
9998  Female  69.034243  163.852461
9999  Female  61.944246  113.649103

Random 20 rows:
      Gender     Height      Weight
3533    Male  68.146474  192.153701
1852    Male  71.760364  211.561670
4328    Male  67.926385  168.685986
5627  Female  64.742004  146.046996
4620    Male  64.239912  169.850769
5443  Female  64.785551  148.431177
657     Male  68.763464  177.083388
1072    Male  68.322982  178.413189
3754    Male  72.442899  200.857658
4833    Male  69.359458  185.284807
489     Male  64.588782  158.454094
1280    Male  73.438772  226.225700
3082    Male  72.311261  199.751430
6672  Female  63.206830  140.457941
4378    Male  66.433684  173.995007
3593    Male  71.452780  196.142631
9149  Female  65.454875  142.458735
3783    Male  70.265756  192.689382
207     Male  65.931785  174.485295
9258  Female  61.130645  110.605644

Dataset shape: (10000, 3)
```

[46]:
```python
#SLIP 3 A
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_csv("Iriss.csv")
```

```
print (data.head(10))
data.describe()
new_data = data[["SepalLengthCm", "SepalWidthCm",
"PetalLengthCm", "PetalWidthCm"]]
print(new_data.head())
plt.figure(figsize = (10, 7))
new_data.boxplot()
```

```
   Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm      Species
0   1            5.1           3.5            1.4           0.2  Iris-setosa
1   2            4.9           3.0            1.4           0.2  Iris-setosa
2   3            4.7           3.2            1.3           0.2  Iris-setosa
3   4            4.6           3.1            1.5           0.2  Iris-setosa
4   5            5.0           3.6            1.4           0.2  Iris-setosa
5   6            5.4           3.9            1.7           0.4  Iris-setosa
6   7            4.6           3.4            1.4           0.3  Iris-setosa
7   8            5.0           3.4            1.5           0.2  Iris-setosa
8   9            4.4           2.9            1.4           0.2  Iris-setosa
9  10            4.9           3.1            1.5           0.1  Iris-setosa
   SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm
0            5.1           3.5            1.4           0.2
1            4.9           3.0            1.4           0.2
2            4.7           3.2            1.3           0.2
3            4.6           3.1            1.5           0.2
4            5.0           3.6            1.4           0.2
```

[46]: <AxesSubplot:>

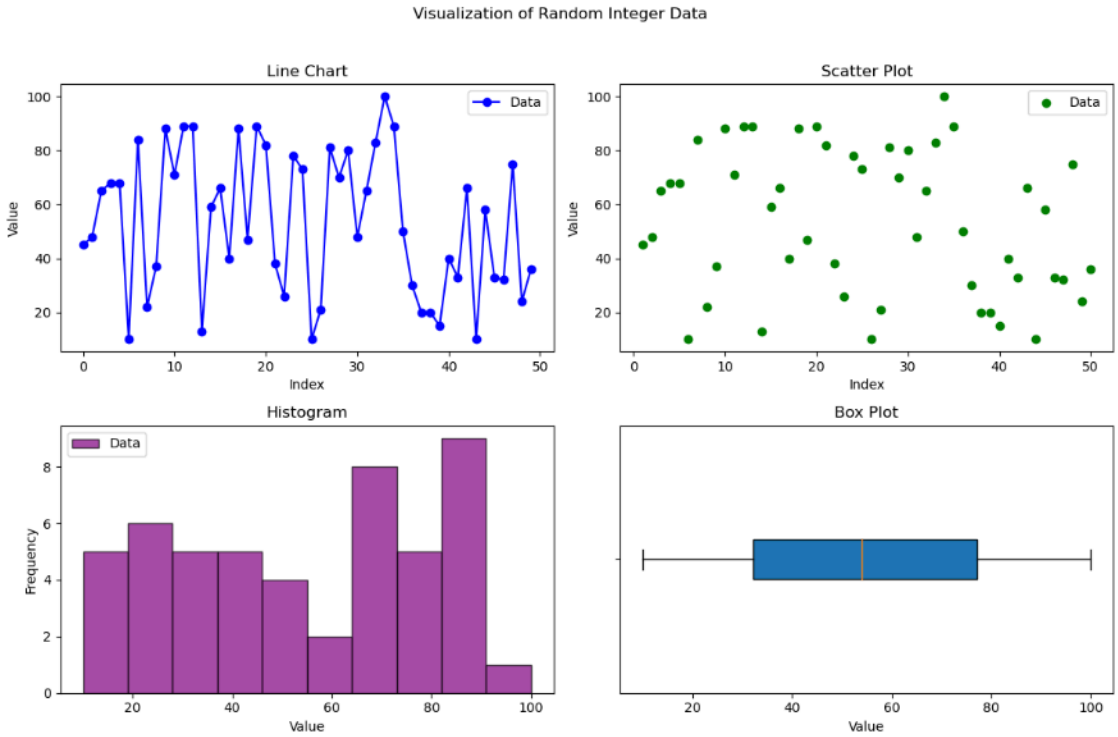


[51]:
```
#SLIP 3 B
import pandas as pd
data = pd.read_csv("weight-height.csv")
print(data.describe())
```

```
              Height        Weight
count  10000.000000  10000.000000
mean      66.367560    161.440357
std        3.847528     32.108439
min       54.263133     64.700127
25%       63.505620    135.818051
50%       66.318070    161.212928
75%       69.174262    187.169525
max       78.998742    269.989699
```

[30]:
```
#SLIP 4 A
import numpy as np
import matplotlib.pyplot as plt
np.random.seed(0)
data = np.random.randint(1, 101, 50)
fig, axs = plt.subplots(2, 2, figsize=(12, 8))
fig.suptitle("Visualization of Random Integer Data")
```

```
# Line chart
axs[0, 0].plot(data, color='blue', marker='o', linestyle='-', label='Data')
axs[0, 0].set_title("Line Chart")
axs[0, 0].set_xlabel("Index")
axs[0, 0].set_ylabel("Value")
axs[0, 0].legend()
# Scatter plot
axs[0, 1].scatter(range(1, 51), data, color='green', label='Data',
marker='o')
axs[0, 1].set_title("Scatter Plot")
axs[0, 1].set_xlabel("Index")
axs[0, 1].set_ylabel("Value")
axs[0, 1].legend()
# Histogram
axs[1, 0].hist(data, bins=10, color='purple', alpha=0.7,
edgecolor='black', label='Data')
axs[1, 0].set_title("Histogram")
axs[1, 0].set_xlabel("Value")
axs[1, 0].set_ylabel("Frequency")
axs[1, 0].legend()
# Box plot
axs[1, 1].boxplot(data, vert=False, patch_artist=True)
axs[1, 1].set_title("Box Plot")
axs[1, 1].set_yticklabels([])
axs[1, 1].set_xlabel("Value")
plt.tight_layout(rect=[0, 0, 1, 0.96])
plt.show()
```

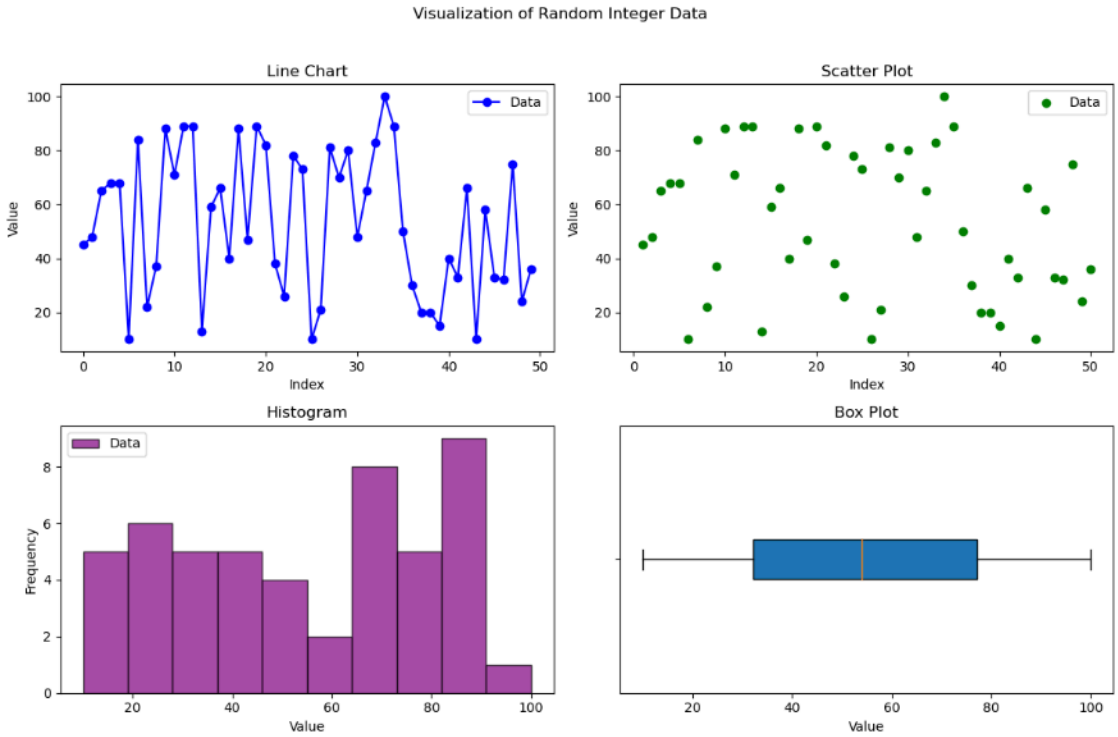


[62]:
```
#SLIP 4 B
import pandas as pd
file_path = "Data.csv"
df = pd.read_csv(file_path)
print("Shape of the DataFrame:", df.shape)
num_rows, num_columns = df.shape
print("Number of rows:", num_rows)
print("Number of columns:", num_columns)
print("Data Types and Feature Names:")
print(df.dtypes)
print("Feature Names:", df.columns.tolist())
print("Data Description:")
print(df.describe())
```

```
Shape of the DataFrame: (15, 4)
Number of rows: 15
Number of columns: 4
Data Types and Feature Names:
Country       object
Age          float64
Salary       float64
Purchased     object
dtype: object
Feature Names: ['Country', 'Age', 'Salary', 'Purchased']
```

```
Data Description:
              Age        Salary
count   14.000000     14.000000
mean    33.714286  54857.142857
std     10.593384  27980.369885
min     21.000000  20000.000000
25%     25.750000  30000.000000
50%     32.000000  52500.000000
75%     40.000000  77000.000000
max     55.000000  98000.000000
```

[64]:
```
#SLIP 5 A
import numpy as np
import matplotlib.pyplot as plt
np.random.seed(0)
data = np.random.randint(1, 101, 50)
fig, axs = plt.subplots(2, 2, figsize=(12, 8))
fig.suptitle("Visualization of Random Integer Data")
# Line chart
axs[0, 0].plot(data, color='blue', marker='o', linestyle='-', label='Data')
axs[0, 0].set_title("Line Chart")
axs[0, 0].set_xlabel("Index")
axs[0, 0].set_ylabel("Value")
axs[0, 0].legend()
# Scatter plot
axs[0, 1].scatter(range(1, 51), data, color='green', label='Data',
marker='o')
axs[0, 1].set_title("Scatter Plot")
axs[0, 1].set_xlabel("Index")
axs[0, 1].set_ylabel("Value")
axs[0, 1].legend()
# Histogram
axs[1, 0].hist(data, bins=10, color='purple', alpha=0.7,
edgecolor='black', label='Data')
axs[1, 0].set_title("Histogram")
axs[1, 0].set_xlabel("Value")
axs[1, 0].set_ylabel("Frequency")
axs[1, 0].legend()
# Box plot
axs[1, 1].boxplot(data, vert=False, patch_artist=True)
axs[1, 1].set_title("Box Plot")
axs[1, 1].set_yticklabels([])
axs[1, 1].set_xlabel("Value")
plt.tight_layout(rect=[0, 0, 1, 0.96])
plt.show()
```



[65]:
```
#SLIP 5 B
import pandas as pd
file_path = "Data.csv"
df = pd.read_csv(file_path)
print("Shape of the DataFrame:", df.shape)
num_rows, num_columns = df.shape
print("Number of rows:", num_rows)
print("Number of columns:", num_columns)
print("Data Types and Feature Names:")
print(df.dtypes)
print("Feature Names:", df.columns.tolist())
print("Data Description:")
print(df.describe())
```

```
Shape of the DataFrame: (15, 4)
Number of rows: 15
Number of columns: 4
Data Types and Feature Names:
Country       object
Age          float64
Salary       float64
Purchased     object
dtype: object
Feature Names: ['Country', 'Age', 'Salary', 'Purchased']
```

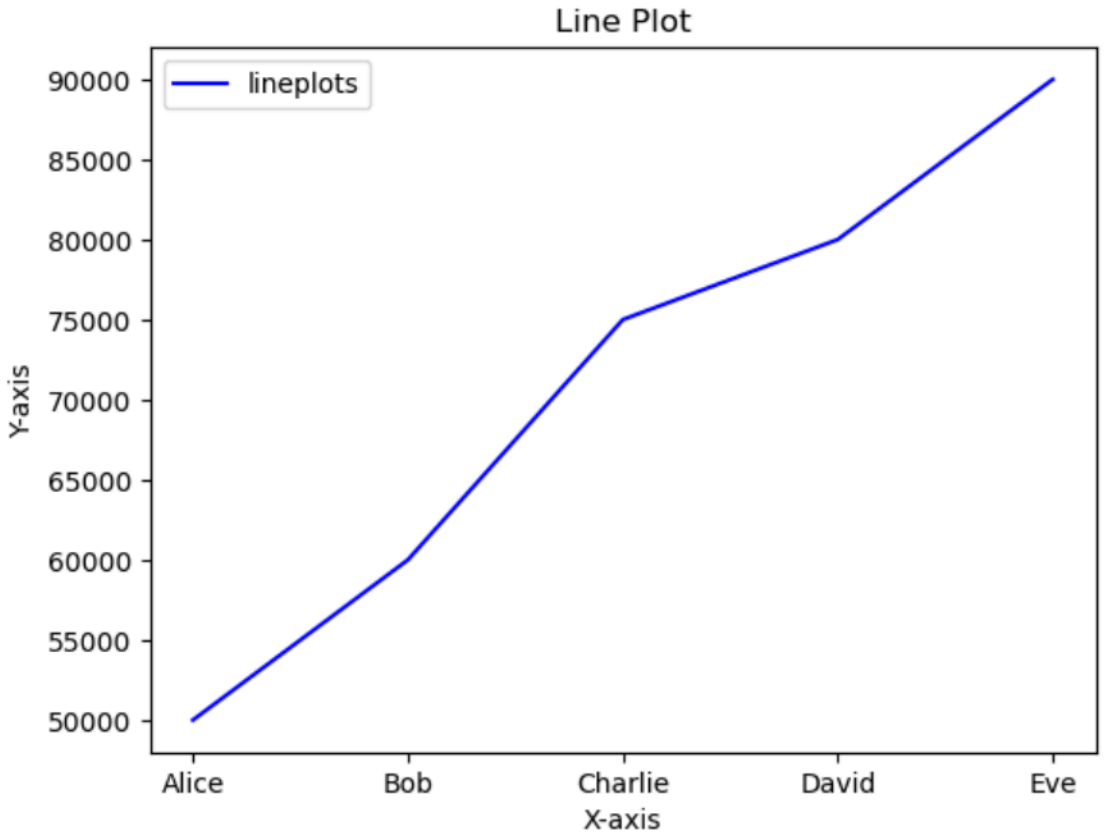
```
Data Description:
              Age        Salary
count   14.000000     14.000000
mean    33.714286  54857.142857
std     10.593384  27980.369885
min     21.000000  20000.000000
25%     25.750000  30000.000000
50%     32.000000  52500.000000
75%     40.000000  77000.000000
max     55.000000  98000.000000
```

[66]:
```
#SLIP 6 A
import pandas as pd
data = pd.read_csv('Data.csv')
data['Salary'] = data['Salary'].fillna(data['Salary'].mean())
data['Age'] = data['Age'].fillna(data['Age'].mean())
print(data)
```

```
      Country        Age        Salary Purchased
0       India  34.000000  92000.000000       Yes
1   Sri lanka  22.000000  25000.000000       Yes
2       China  31.000000  74000.000000       Yes
3   Sri lanka  29.000000  54857.142857        No
4       China  55.000000  98000.000000       Yes
5       India  24.000000  30000.000000        No
6   Sri lanka  28.000000  40000.000000        No
7       India  33.714286  60000.000000        No
8       China  51.000000  89000.000000       Yes
9       India  44.000000  78000.000000       Yes
10  Sri lanka  21.000000  20000.000000        No
11      China  25.000000  30000.000000       Yes
12      India  33.000000  45000.000000       Yes
13      India  42.000000  65000.000000       Yes
14  Sri lanka  33.000000  22000.000000        No
```

[67]:
```
#SLIP 6 B
import matplotlib.pyplot as plt
name = ["Alice", "Bob","Charlie","David","Eve"]
salary = [50000, 60000, 75000, 80000, 90000]
plt.plot(name,salary, label='lineplots', color='b')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Line Plot')
plt.legend()
plt.show()
```

```python
#ALTERNATIVE CODE WITH OUTPUT


import matplotlib.pyplot as plt
names = ["Alice", "Bob", "Charlie", "David", "Eve"]
salaries = [50000, 60000, 75000, 80000, 90000]
plt.figure(figsize=(8, 6))
plt.plot(names, salaries, marker='o', linestyle='-')
plt.title("Salary vs Name")
plt.xlabel("Name")
plt.ylabel("Salary")
plt.grid(True)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



[68]:
```
#SLIP 6 C
import pandas as pd
file_path = "weight-height.csv"
df = pd.read_csv(file_path)
print("First 10 rows:")
print(df.head(10))
print("\nLast 10 rows:")
print(df.tail(10))
print("\nRandom 20 rows:")
print(df.sample(20))
print("\nDataset shape:", df.shape)
```

```
First 10 rows:
  Gender     Height      Weight
0   Male  73.847017  241.893563
1   Male  68.781904  162.310473
2   Male  74.110105  212.740856
3   Male  71.730978  220.042470
4   Male  69.881796  206.349801
5   Male  67.253016  152.212156
6   Male  68.785081  183.927889
```

```
7   Male  68.348516  167.971110
8   Male  67.018950  175.929440
9   Male  63.456494  156.399676

Last 10 rows:
      Gender     Height      Weight
9990  Female  63.179498  141.266100
9991  Female  62.636675  102.853563
9992  Female  62.077832  138.691680
9993  Female  60.030434   97.687432
9994  Female  59.098250  110.529686
9995  Female  66.172652  136.777454
9996  Female  67.067155  170.867906
9997  Female  63.867992  128.475319
9998  Female  69.034243  163.852461
9999  Female  61.944246  113.649103

Random 20 rows:
      Gender     Height      Weight
4969    Male  71.169447  195.330596
7395  Female  64.281598  135.966146
5372  Female  67.803789  162.509491
2087    Male  74.147857  211.237968
9534  Female  65.489716  121.330674
2936    Male  73.143178  217.238225
6224  Female  66.624884  141.187262
9565  Female  67.156819  150.912424
1563    Male  71.690869  188.656335
4379    Male  69.606437  184.644157
9926  Female  63.171535  142.747754
5753  Female  67.763504  154.041759
2097    Male  65.828196  168.421283
8175  Female  61.735826  110.513242
8721  Female  67.408918  170.270948
5790  Female  62.242208  133.324581
9701  Female  63.220229  125.074859
7637  Female  64.433935  135.269942
2670    Male  68.522231  188.053333
8680  Female  62.088826  130.941881

Dataset shape: (10000, 3)
```

[50]:
```python
#SLIP 7 A
import pandas as pd
from sklearn.preprocessing import LabelEncoder
file_path = "Data.csv"
df = pd.read_csv(file_path)
```

```
# Apply One-Hot Encoding to the "Country" column
df = pd.get_dummies(df, columns=['Country'], prefix=['Country'])
print("Modified DataFrame:")
print(df)
```

```
Modified DataFrame:
     Age   Salary Purchased  Country_China  Country_India  Country_Sri lanka
0   34.0  92000.0       Yes              0              1                   0
1   22.0  25000.0       Yes              0              0                   1
2   31.0  74000.0       Yes              1              0                   0
3   29.0      NaN        No              0              0                   1
4   55.0  98000.0       Yes              1              0                   0
5   24.0  30000.0        No              0              1                   0
6   28.0  40000.0        No              0              0                   1
7    NaN  60000.0        No              0              1                   0
8   51.0  89000.0       Yes              1              0                   0
9   44.0  78000.0       Yes              0              1                   0
10  21.0  20000.0        No              0              0                   1
11  25.0  30000.0       Yes              1              0                   0
12  33.0  45000.0       Yes              0              1                   0
13  42.0  65000.0       Yes              0              1                   0
14  33.0  22000.0        No              0              0                   1
```

[51]:
```
#SLIP 7 B
import pandas as pd
from sklearn.preprocessing import LabelEncoder
file_path = "Data.csv"
df = pd.read_csv(file_path)
# Apply Label Encoding to the "Purchased" column
label_encoder = LabelEncoder()
df['Purchased'] = label_encoder.fit_transform(df['Purchased'])
print("Modified DataFrame:")
print(df)
```

```
Modified DataFrame:
      Country   Age   Salary  Purchased
0       India  34.0  92000.0          1
1   Sri lanka  22.0  25000.0          1
2       China  31.0  74000.0          1
3   Sri lanka  29.0      NaN          0
4       China  55.0  98000.0          1
5       India  24.0  30000.0          0
6   Sri lanka  28.0  40000.0          0
7       India   NaN  60000.0          0
8       China  51.0  89000.0          1
9       India  44.0  78000.0          1
10  Sri lanka  21.0  20000.0          0
11      China  25.0  30000.0          1
```

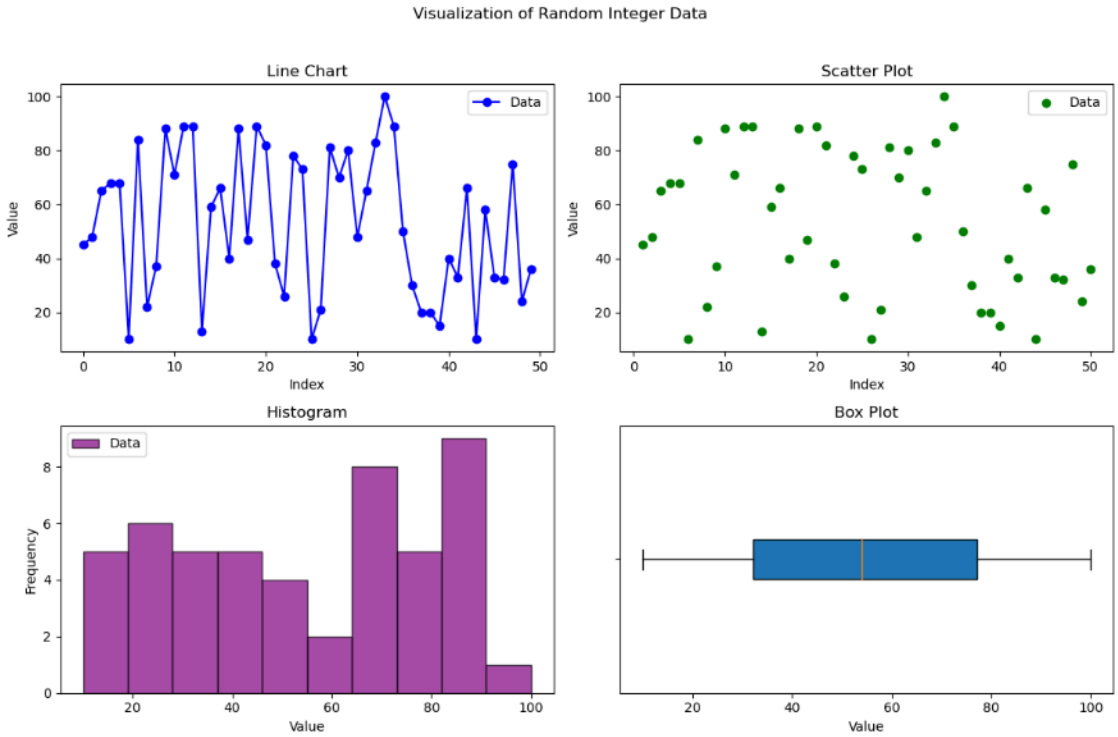
```
12      India  33.0  45000.0          1
13      India  42.0  65000.0          1
14  Sri lanka  33.0  22000.0          0
```

[69]:
```
#SLIP 8
import pandas as pd
from sklearn.preprocessing import StandardScaler
file_path = "winequality-red.csv"
df = pd.read_csv(file_path)
columns_to_standardize = df.columns
scaler = StandardScaler()
df[columns_to_standardize] = scaler.fit_transform(df[columns_to_standardize])
print("Standardized Data:")
print(df)
```

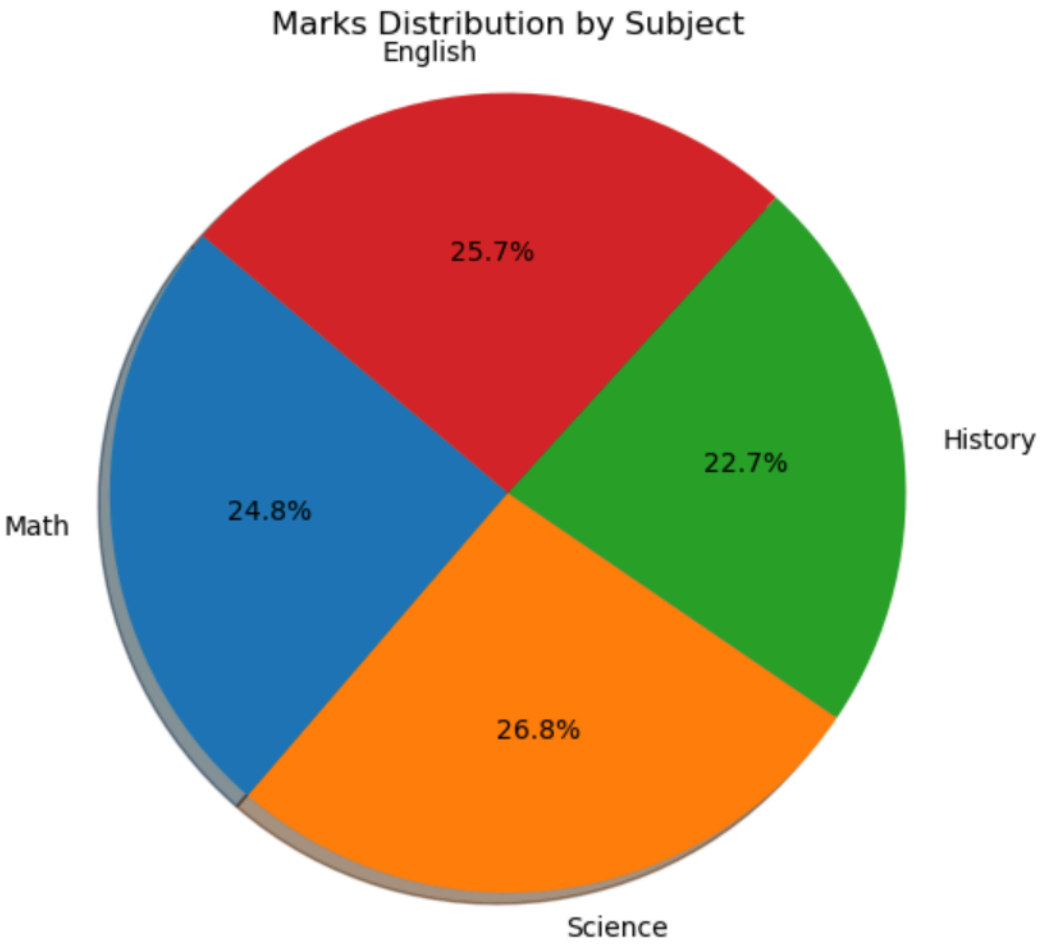
```
Standardized Data:
      fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
0         -0.528360          0.961877    -1.391472       -0.453218  -0.243707
1         -0.298547          1.967442    -1.391472        0.043416   0.223875
2         -0.298547          1.297065    -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
…               …                 …            …               …          …
1594      -1.217796          0.403229    -0.980669       -0.382271   0.053845
1595      -1.390155          0.123905    -0.877968       -0.240375  -0.541259
1596      -1.160343         -0.099554    -0.723916       -0.169427  -0.243707
1597      -1.390155          0.654620    -0.775267       -0.382271  -0.264960
1598      -1.332702         -1.216849     1.021999        0.752894  -0.434990

      free sulfur dioxide  total sulfur dioxide   density        pH  \
0               -0.466193             -0.379133  0.558274  1.288643
1                0.872638              0.624363  0.028261 -0.719933
2               -0.083669              0.229047  0.134264 -0.331177
3                0.107592              0.411500  0.664277 -0.979104
4               -0.466193             -0.379133  0.558274  1.288643
…                     …                     …         …         …
1594             1.542054             -0.075043 -0.978765  0.899886
1595             2.211469              0.137820 -0.862162  1.353436
1596             1.255161             -0.196679 -0.533554  0.705508
1597             1.542054             -0.075043 -0.676657  1.677400
1598             0.203223             -0.135861 -0.666057  0.511130

      sulphates   alcohol   quality
0     -0.579207 -0.960246 -0.787823
1      0.128950 -0.584777 -0.787823
2     -0.048089 -0.584777 -0.787823
3     -0.461180 -0.584777  0.450848
4     -0.579207 -0.960246 -0.787823
```

```
...          ...       ...       ...
1594  -0.461180  0.072294 -0.787823
1595   0.601055  0.729364  0.450848
1596   0.542042  0.541630  0.450848
1597   0.305990 -0.209308 -0.787823
1598   0.010924  0.541630  0.450848

[1599 rows x 12 columns]
```

[70]:
```
#SLIP 9 A
import numpy as np
import matplotlib.pyplot as plt
np.random.seed(0)
data = np.random.randint(1, 101, 50)
fig, axs = plt.subplots(2, 2, figsize=(12, 8))
fig.suptitle("Visualization of Random Integer Data")
# Line chart
axs[0, 0].plot(data, color='blue', marker='o', linestyle='-', label='Data')
axs[0, 0].set_title("Line Chart")
axs[0, 0].set_xlabel("Index")
axs[0, 0].set_ylabel("Value")
axs[0, 0].legend()
# Scatter plot
axs[0, 1].scatter(range(1, 51), data, color='green', label='Data',
marker='o')
axs[0, 1].set_title("Scatter Plot")
axs[0, 1].set_xlabel("Index")
axs[0, 1].set_ylabel("Value")
axs[0, 1].legend()
# Histogram
axs[1, 0].hist(data, bins=10, color='purple', alpha=0.7,
edgecolor='black', label='Data')
axs[1, 0].set_title("Histogram")
axs[1, 0].set_xlabel("Value")
axs[1, 0].set_ylabel("Frequency")
axs[1, 0].legend()
# Box plot
axs[1, 1].boxplot(data, vert=False, patch_artist=True)
axs[1, 1].set_title("Box Plot")
axs[1, 1].set_yticklabels([])
axs[1, 1].set_xlabel("Value")
plt.tight_layout(rect=[0, 0, 1, 0.96])
plt.show()
```



[45]:
```
#SLIP 9 B
import matplotlib.pyplot as plt
subjects = ["Math", "Science", "History", "English"]
marks = [85, 92, 78, 88]
plt.figure(figsize=(6, 6))
plt.pie(marks, labels=subjects, autopct='%1.1f%%', startangle=140,
shadow=True)
plt.title("Marks Distribution by Subject")
plt.axis('equal')
plt.show()
```



[71]:
```
#SLIP 9 C
import pandas as pd
#a
data = pd.read_csv("winequality-red.csv")
print("Describing the dataset:")
print(data.describe())
#b
print("\n Shape of the dataset:")
print(data.shape)
#c
print("\n Display first 3 rows from the dataset:")
print(data.head(3))
```

```
Describing the dataset:
       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:
(1599, 12)

 Display first 3 rows from 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
```

[72]:
```
#SLIP 10 A
import pandas as pd
file_path = "SOCR-HeightWeight.csv"
df = pd.read_csv(file_path)
mean_values = df.mean()
print("Column-wise Mean:")
print(mean_values)
median_values = df.median()
print("\nColumn-wise Median:")
print(median_values)
```

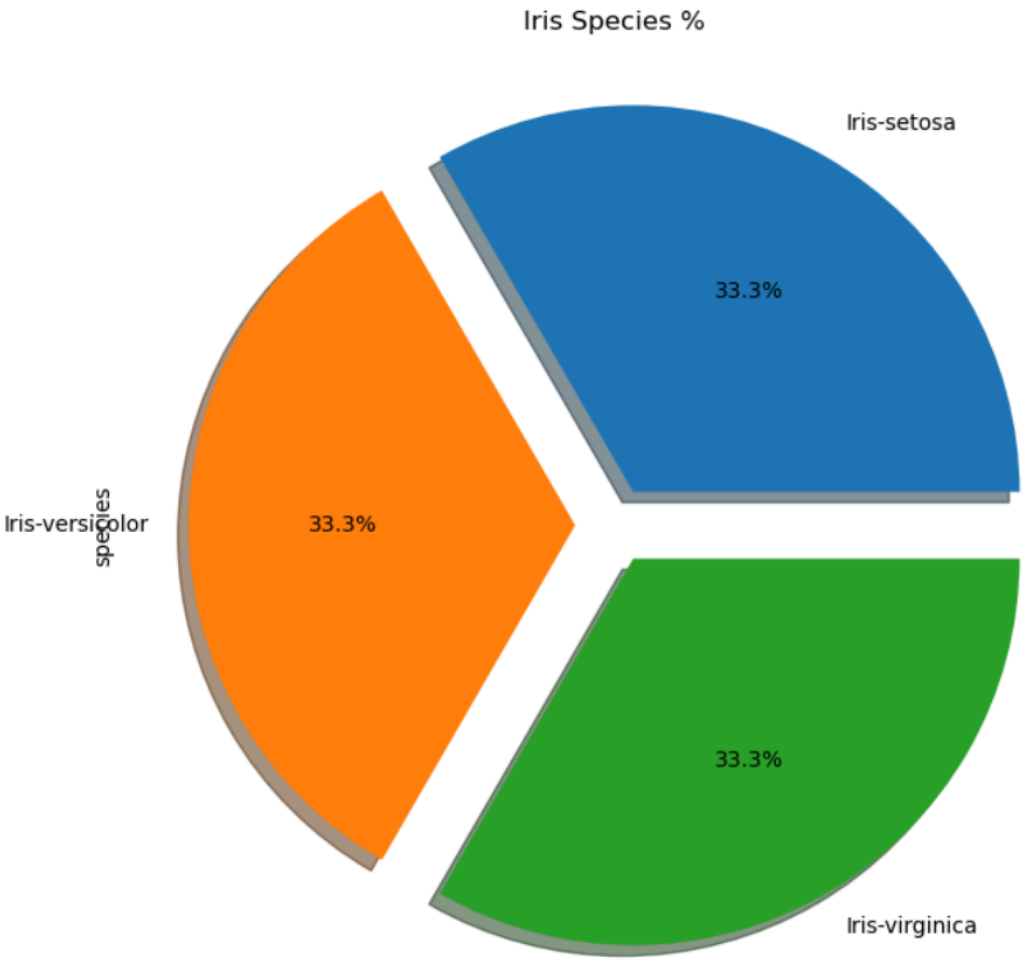
```
Column-wise Mean:
Index              12500.500000
Height(Inches)        67.993114
Weight(Pounds)       127.079421
dtype: float64

Column-wise Median:
Index              12500.50000
Height(Inches)        67.99570
Weight(Pounds)       127.15775
dtype: float64
```

[73]:
```
#SLIP 10 B
def manhattan_distance(point1, point2):
 return abs(point1[0] - point2[0]) + abs(point1[1] - point2[1])
def sum_manhattan_distances(points):
 total_distance = 0
 for i in range(len(points)):
    for j in range(i + 1, len(points)):
        distance = manhattan_distance(points[i], points[j])
        total_distance += distance
 return total_distance
points = [(1, 2), (3, 4), (5, 6), (7, 8)]
result = sum_manhattan_distances(points)
print("Sum of Manhattan Distances:", result)
```

```
Sum of Manhattan Distances: 40
```

[5]:
```
#SLIP 11 A
import pandas as pd
import matplotlib.pyplot as plt
iris = pd.read_csv("IRIS.csv")
ax=plt.subplots(1,1,figsize=(10,8))
iris['species'].value_counts().plot.pie(explode=[0.1,0.1,0.1],autopct='%1.
 ↪1f%%',shadow=True,figsize=(10,8))
plt.title("Iris Species %")
```

```
plt.show()
```



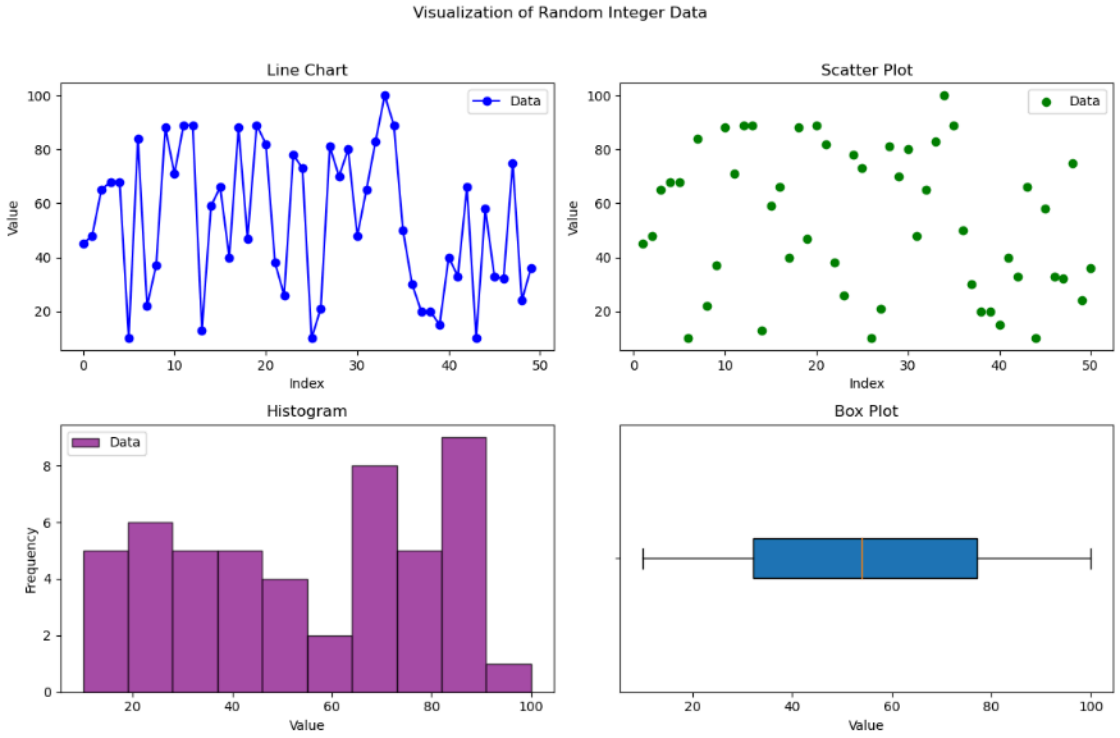
[6]:
```
#SLIP 11 B
import pandas as pd
data = pd.read_csv("winequality-red.csv ")
print(data.describe())
```

```
       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
```

[74]:
```python
#SLIP 12 A
import numpy as np
import matplotlib.pyplot as plt
np.random.seed(0)
data = np.random.randint(1, 101, 50)
fig, axs = plt.subplots(2, 2, figsize=(12, 8))
fig.suptitle("Visualization of Random Integer Data")
# Line chart
axs[0, 0].plot(data, color='blue', marker='o', linestyle='-', label='Data')
axs[0, 0].set_title("Line Chart")
axs[0, 0].set_xlabel("Index")
axs[0, 0].set_ylabel("Value")
axs[0, 0].legend()
# Scatter plot
axs[0, 1].scatter(range(1, 51), data, color='green', label='Data',
marker='o')
axs[0, 1].set_title("Scatter Plot")
axs[0, 1].set_xlabel("Index")
axs[0, 1].set_ylabel("Value")
axs[0, 1].legend()
# Histogram
axs[1, 0].hist(data, bins=10, color='purple', alpha=0.7,
edgecolor='black', label='Data')
axs[1, 0].set_title("Histogram")
axs[1, 0].set_xlabel("Value")
axs[1, 0].set_ylabel("Frequency")
```

```
axs[1, 0].legend()
# Box plot
axs[1, 1].boxplot(data, vert=False, patch_artist=True)
axs[1, 1].set_title("Box Plot")
axs[1, 1].set_yticklabels([])
axs[1, 1].set_xlabel("Value")
plt.tight_layout(rect=[0, 0, 1, 0.96])
plt.show()
```



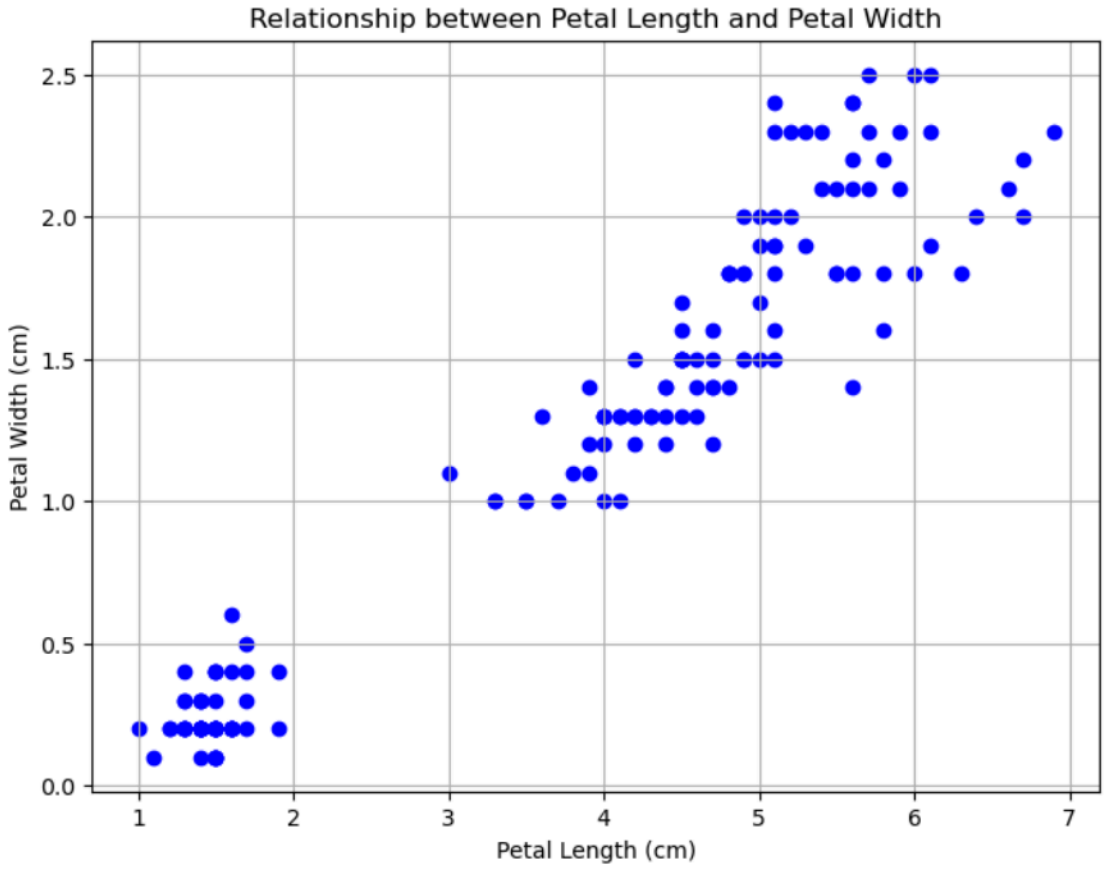
[16]:
```
#SLIP 12 B
import pandas as pd
data = {'name': ['Alice', 'Bob', 'Charlie', 'David', 'Emma', 'Frank', 'Alice',␣
↪'Bob', 'Charlie', 'David'],
        'salary': [50000, 60000, 70000, None, 55000, 60000, 50000, 60000,␣
↪70000, 75000],
        'department': ['HR', 'IT', 'Finance', 'HR', 'IT', 'Finance', 'HR',␣
↪'IT', 'Finance', 'HR']}
df = pd.DataFrame(data)
print("Original DataFrame:")
print(df)
print("\n")
new_data = {'name': ['Alice', 'Bob', 'Eva', 'David', 'Frank', 'Alice', 'Bob',␣
↪'Charlie', 'David', 'Eva'],
```

```
            'salary': [50000, 60000, None, 75000, 60000, None, 60000, 70000,␣
 ↪75000, 80000],
            'department': ['HR', 'IT', 'Marketing', 'HR', 'Finance', 'HR',␣
 ↪'IT', 'Finance', 'HR', 'Marketing']}
df = pd.concat([df, pd.DataFrame(new_data)], ignore_index=True)
print("DataFrame after adding new rows:")
print(df)
print("\n")
df = df.dropna()
print("Modified DataFrame after dropping null and empty values:")
print(df)
```

```
Original DataFrame:
      name   salary department
0    Alice  50000.0         HR
1      Bob  60000.0         IT
2  Charlie  70000.0    Finance
3    David      NaN         HR
4     Emma  55000.0         IT
5    Frank  60000.0    Finance
6    Alice  50000.0         HR
7      Bob  60000.0         IT
8  Charlie  70000.0    Finance
9    David  75000.0         HR


DataFrame after adding new rows:
       name   salary department
0     Alice  50000.0         HR
1       Bob  60000.0         IT
2   Charlie  70000.0    Finance
3     David      NaN         HR
4      Emma  55000.0         IT
5     Frank  60000.0    Finance
6     Alice  50000.0         HR
7       Bob  60000.0         IT
8   Charlie  70000.0    Finance
9     David  75000.0         HR
10    Alice  50000.0         HR
11      Bob  60000.0         IT
12      Eva      NaN  Marketing
13    David  75000.0         HR
14    Frank  60000.0    Finance
15    Alice      NaN         HR
16      Bob  60000.0         IT
17  Charlie  70000.0    Finance
18    David  75000.0         HR
```

```
19      Eva  80000.0  Marketing


Modified DataFrame after dropping null and empty values:
       name   salary department
0     Alice  50000.0         HR
1       Bob  60000.0         IT
2   Charlie  70000.0    Finance
4      Emma  55000.0         IT
5     Frank  60000.0    Finance
6     Alice  50000.0         HR
7       Bob  60000.0         IT
8   Charlie  70000.0    Finance
9     David  75000.0         HR
10    Alice  50000.0         HR
11      Bob  60000.0         IT
13    David  75000.0         HR
14    Frank  60000.0    Finance
16      Bob  60000.0         IT
17  Charlie  70000.0    Finance
18    David  75000.0         HR
19      Eva  80000.0  Marketing
```

[75]:
```python
#SLIP 13 A
import pandas as pd
import matplotlib.pyplot as plt
file_path = "Iriss.csv"
df = pd.read_csv(file_path)
petal_length = df['PetalLengthCm']
petal_width = df['PetalWidthCm']
plt.figure(figsize=(8, 6))
plt.scatter(petal_length, petal_width, color='blue', marker='o')
plt.title("Relationship between Petal Length and Petal Width")
plt.xlabel("Petal Length (cm)")
plt.ylabel("Petal Width (cm)")
plt.grid(True)
plt.show()
```



[76]:
```
#SLIP 13 B
import numpy as np
flattened_array = [3, 8, 12, 5, 2, 15, 9, 4, 10]
array = np.array(flattened_array)
max_value = np.max(array)
min_value = np.min(array)
print("Maximum value:", max_value)
print("Minimum value:", min_value)
```

```
Maximum value: 15
Minimum value: 2
```

[77]:
```
#SLIP 14 A
import numpy as np
flattened_array = np.array([1, 2, 3, 4, 5, 6])
weights = np.array([0.1, 0.2, 0.3, 0.2, 0.1, 0.1])
weighted_average = np.average(flattened_array, weights=weights)
print("Weighted Average:", weighted_average)
```

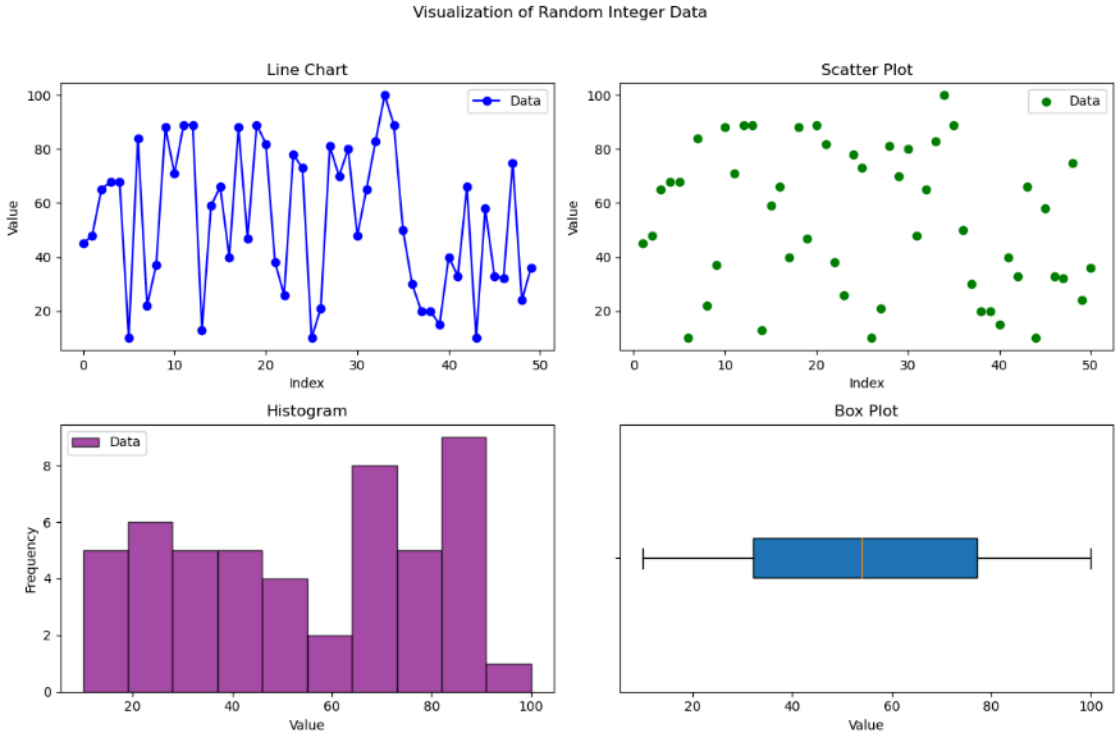
```
Weighted Average: 3.3000000000000003
```

[78]:
```
#SLIP 14 B
import pandas as pd
file_path = "Advertising.csv"
df = pd.read_csv(file_path)
summary = df.describe()
print("Basic Statistical Details of the Data:")
print(summary)
```

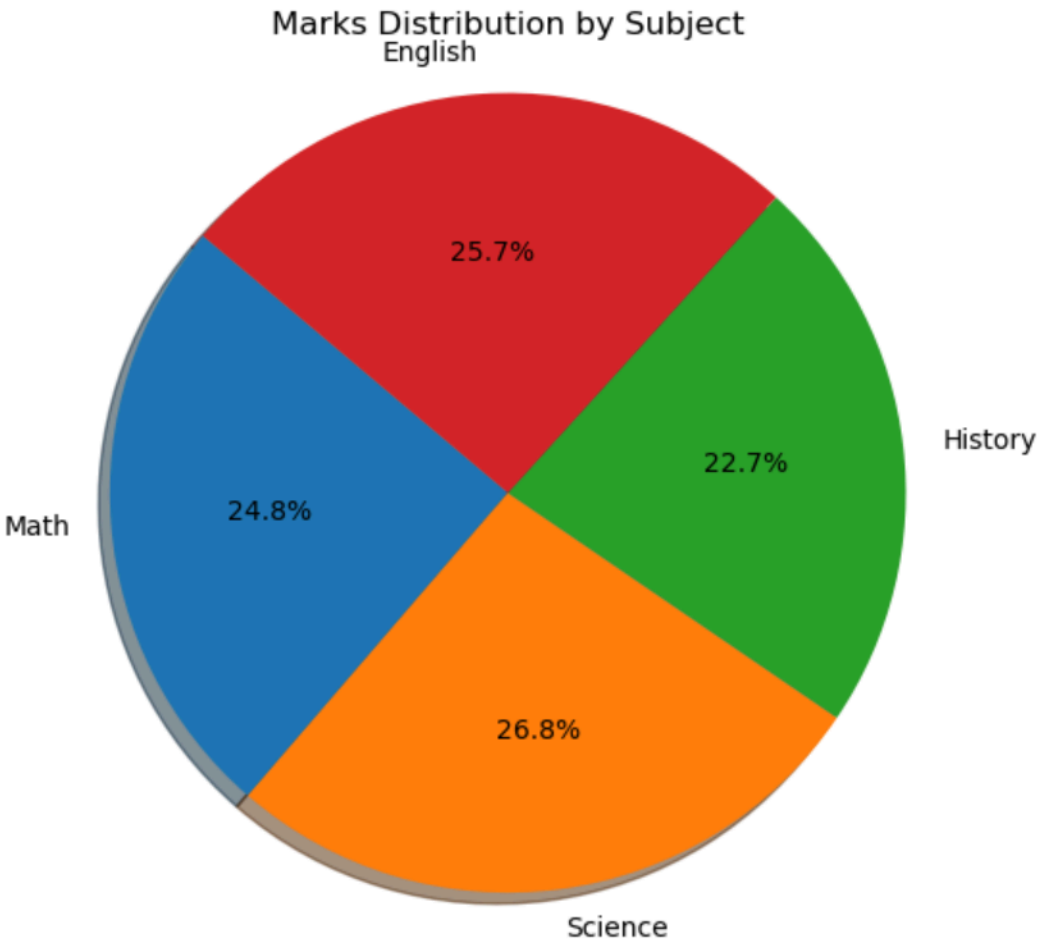
```
Basic Statistical Details of the Data:
       Unnamed: 0          TV       Radio   Newspaper       Sales
count  200.000000  200.000000  200.000000  200.000000  200.000000
mean   100.500000  147.042500   23.264000   30.554000   14.022500
std     57.879185   85.854236   14.846809   21.778621    5.217457
min      1.000000    0.700000    0.000000    0.300000    1.600000
25%     50.750000   74.375000    9.975000   12.750000   10.375000
50%    100.500000  149.750000   22.900000   25.750000   12.900000
75%    150.250000  218.825000   36.525000   45.100000   17.400000
max    200.000000  296.400000   49.600000  114.000000   27.000000
```

[79]:
```
#SLIP 15 A
import numpy as np
import matplotlib.pyplot as plt
np.random.seed(0)
data = np.random.randint(1, 101, 50)
fig, axs = plt.subplots(2, 2, figsize=(12, 8))
fig.suptitle("Visualization of Random Integer Data")
# Line chart
axs[0, 0].plot(data, color='blue', marker='o', linestyle='-', label='Data')
axs[0, 0].set_title("Line Chart")
axs[0, 0].set_xlabel("Index")
axs[0, 0].set_ylabel("Value")
axs[0, 0].legend()
# Scatter plot
axs[0, 1].scatter(range(1, 51), data, color='green', label='Data',
marker='o')
axs[0, 1].set_title("Scatter Plot")
axs[0, 1].set_xlabel("Index")
axs[0, 1].set_ylabel("Value")
axs[0, 1].legend()
# Histogram
axs[1, 0].hist(data, bins=10, color='purple', alpha=0.7,
edgecolor='black', label='Data')
axs[1, 0].set_title("Histogram")
axs[1, 0].set_xlabel("Value")
axs[1, 0].set_ylabel("Frequency")
axs[1, 0].legend()
# Box plot
```

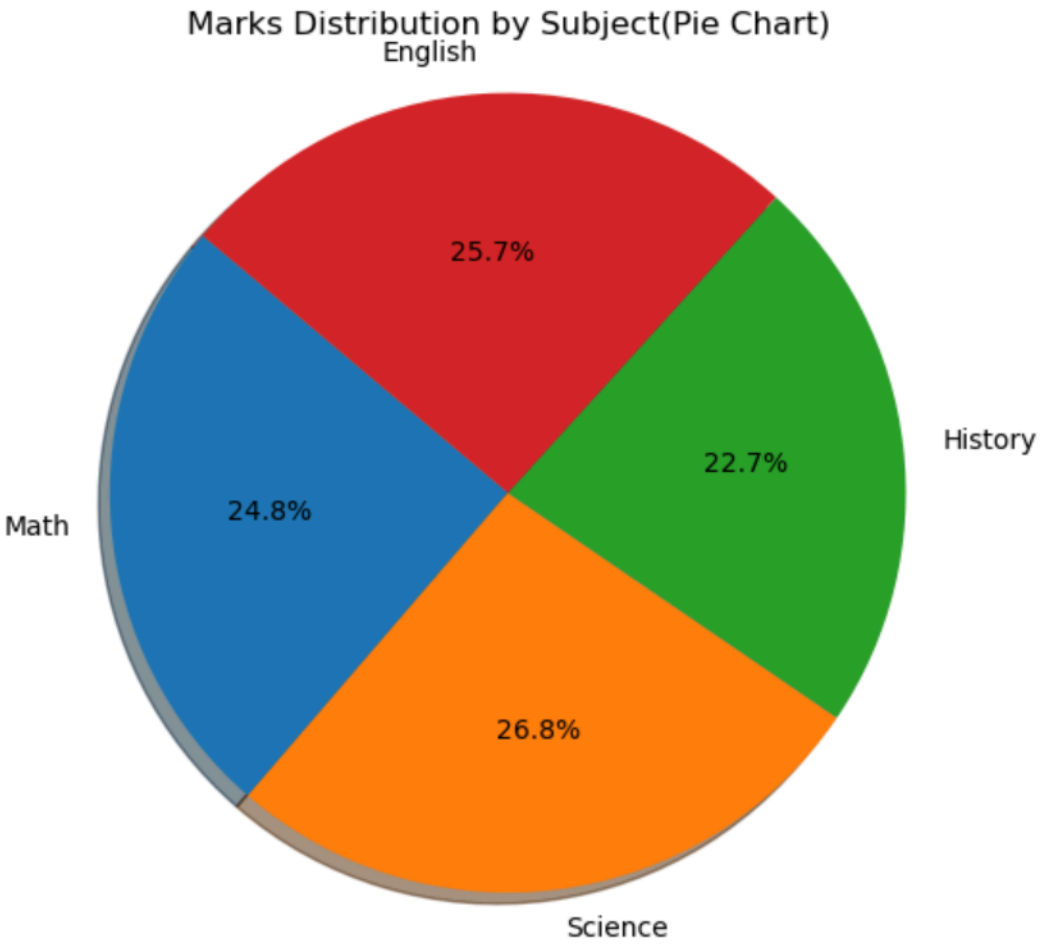
```
axs[1, 1].boxplot(data, vert=False, patch_artist=True)
axs[1, 1].set_title("Box Plot")
axs[1, 1].set_yticklabels([])
axs[1, 1].set_xlabel("Value")
plt.tight_layout(rect=[0, 0, 1, 0.96])
plt.show()
```

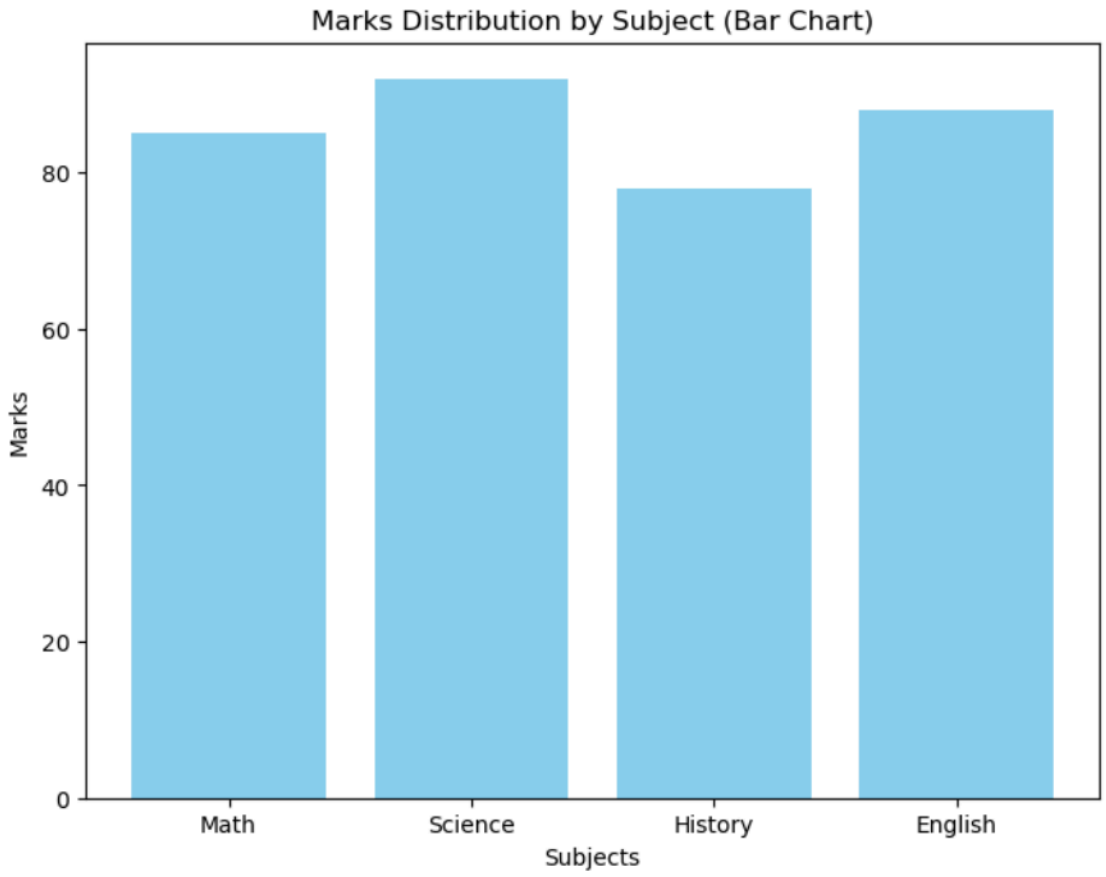


[80]:
```
#SLIP 15 B
import matplotlib.pyplot as plt
subjects = ["Math", "Science", "History", "English"]
marks = [85, 92, 78, 88]
plt.figure(figsize=(6, 6))
plt.pie(marks, labels=subjects, autopct='%1.1f%%', startangle=140,
shadow=True)
plt.title("Marks Distribution by Subject")
plt.axis('equal')
plt.show()
```



[81]:
```
#SLIP 16 A
import matplotlib.pyplot as plt
subjects = ["Math", "Science", "History", "English"]
marks = [85, 92, 78, 88]
plt.figure(figsize=(6, 6))
plt.pie(marks, labels=subjects, autopct='%1.1f%%', startangle=140,
shadow=True)
plt.title("Marks Distribution by Subject(Pie Chart)")
plt.axis('equal')
plt.show()
plt.figure(figsize=(8, 6))
plt.bar(subjects, marks, color='skyblue')
plt.xlabel('Subjects')
plt.ylabel('Marks')
plt.title('Marks Distribution by Subject (Bar Chart)')
plt.show()
```


Marks Distribution by Subject(Pie Chart)
English
25.7%
History
22.7%
Math
24.8%
26.8%
Science

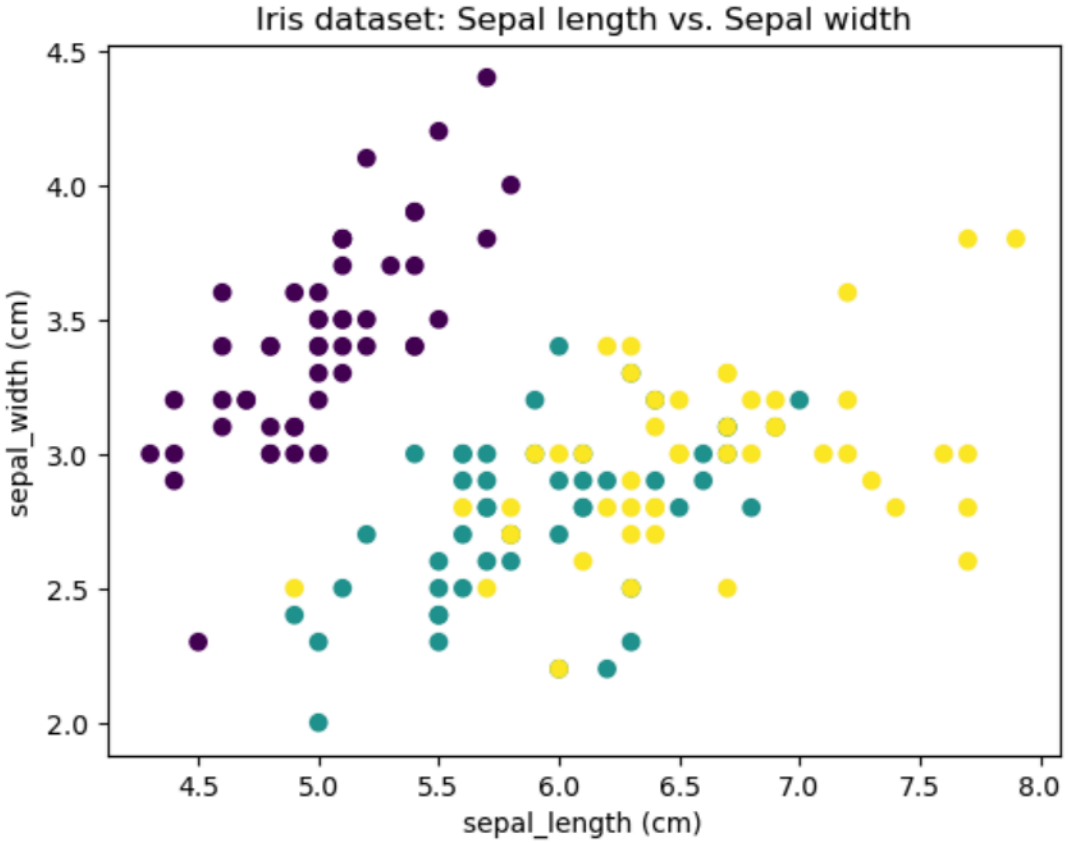


[3]:
```
#SLIP 16 B
import pandas as pd
students_data = {'Name': ['Alice', 'Bob', 'Charlie', 'David'],
    'Graduation Percentage': [85.5, 92.0, 78.3, 89.7],
    'Age': [22, 23, 21, 24]}
students_df = pd.DataFrame(students_data)
average_age = students_df['Age'].mean()
average_percentage = students_df['Graduation Percentage'].mean()
print(f"Average Age of Students: {average_age}")
print(f"Average Graduation Percentage: {average_percentage}")
```

```
Average Age of Students: 22.5
Average Graduation Percentage: 86.375
```

[15]:
```
#SLIP 17 A
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
iris = load_iris()
X = iris.data[:, :2]
```

```
y = iris.target
plt.scatter(X[:, 0], X[:, 1], c=y, cmap='viridis')
plt.xlabel('sepal_length (cm)')
plt.ylabel('sepal_width (cm)')
plt.title('Iris dataset: Sepal length vs. Sepal width')
plt.show()
```



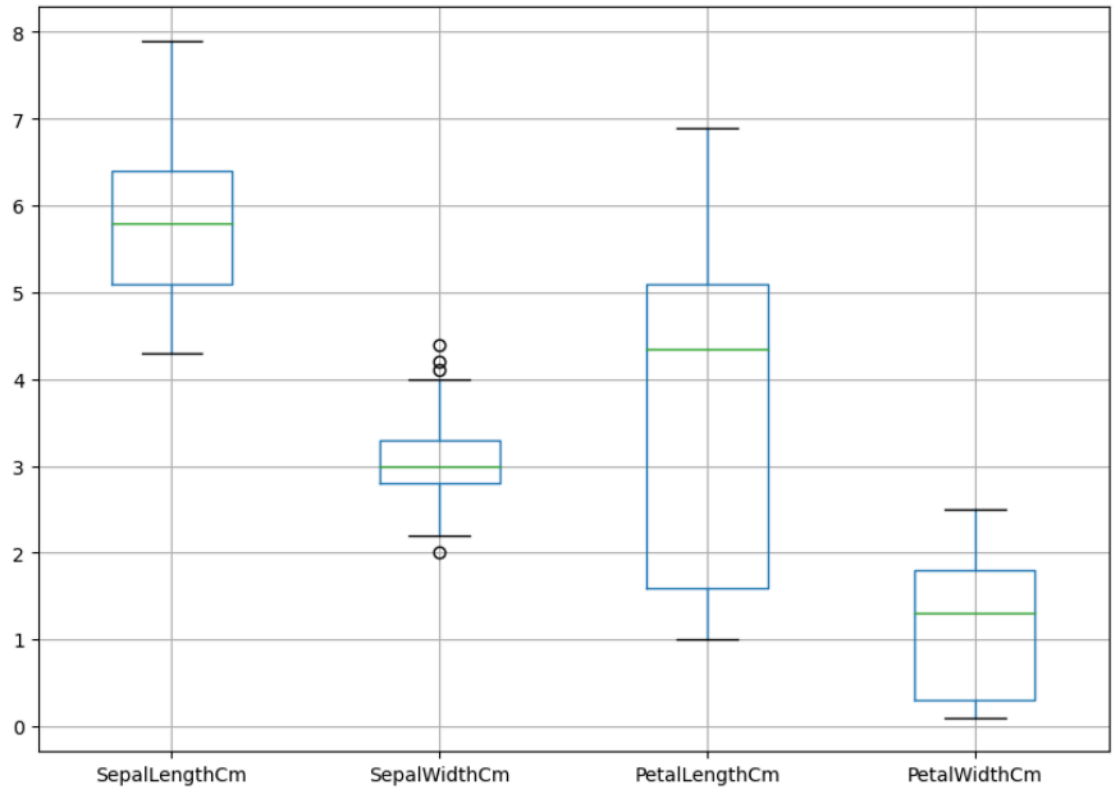
[18]:
```
#SLIP 17 B
import pandas as pd
data = {'Name': ['Alice', 'Bob', 'Charlie', 'David', 'Emma', 'Frank', 'Grace',␣
 ↪'Hank', 'Ivy', 'Jack'],
    'Age': [25, 30, 35, 28, 22, 40, 32, 45, 28, 33],
    'Salary': [50000, 60000, 70000, 55000, 60000, 75000, 65000, 80000, 60000,␣
 ↪70000],
    'Department': ['HR', 'IT', 'Finance', 'HR', 'IT', 'Finance', 'IT',␣
 ↪'Finance', 'HR', 'IT']}
df = pd.DataFrame(data)
print("Data Frame:")
print(df)
```

```
Data Frame:
      Name  Age  Salary Department
0    Alice   25   50000         HR
1      Bob   30   60000         IT
2  Charlie   35   70000    Finance
3    David   28   55000         HR
4     Emma   22   60000         IT
5    Frank   40   75000    Finance
6    Grace   32   65000         IT
7     Hank   45   80000    Finance
8      Ivy   28   60000         HR
9     Jack   33   70000         IT
```

[21]:
```
#SLIP 18 A
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_csv("Iriss.csv")
print (data.head(10))
data.describe()
new_data = data[["SepalLengthCm", "SepalWidthCm",
"PetalLengthCm", "PetalWidthCm"]]
print(new_data.head())
plt.figure(figsize = (10, 7))
new_data.boxplot()
```

```
   Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm      Species
0   1            5.1           3.5            1.4           0.2  Iris-setosa
1   2            4.9           3.0            1.4           0.2  Iris-setosa
2   3            4.7           3.2            1.3           0.2  Iris-setosa
3   4            4.6           3.1            1.5           0.2  Iris-setosa
4   5            5.0           3.6            1.4           0.2  Iris-setosa
5   6            5.4           3.9            1.7           0.4  Iris-setosa
6   7            4.6           3.4            1.4           0.3  Iris-setosa
7   8            5.0           3.4            1.5           0.2  Iris-setosa
8   9            4.4           2.9            1.4           0.2  Iris-setosa
9  10            4.9           3.1            1.5           0.1  Iris-setosa
   SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm
0            5.1           3.5            1.4           0.2
1            4.9           3.0            1.4           0.2
2            4.7           3.2            1.3           0.2
3            4.6           3.1            1.5           0.2
4            5.0           3.6            1.4           0.2
```

[21]: <AxesSubplot:>



[23]:
```
#SLIP 18 B
import pandas as pd
file_path = "weight-height.csv"
df = pd.read_csv(file_path)
print("First 5 rows:")
print(df.head(5))
print("\nLast 5 rows:")
print(df.tail(5))
print("\nRandom 10 rows:")
print(df.sample(10))
```

```
First 5 rows:
  Gender     Height      Weight
0   Male  73.847017  241.893563
1   Male  68.781904  162.310473
2   Male  74.110105  212.740856
3   Male  71.730978  220.042470
4   Male  69.881796  206.349801

Last 5 rows:
      Gender     Height      Weight
9995  Female  66.172652  136.777454
```

```
9996  Female  67.067155  170.867906
9997  Female  63.867992  128.475319
9998  Female  69.034243  163.852461
9999  Female  61.944246  113.649103

Random 10 rows:
      Gender     Height      Weight
53      Male  67.330831  181.407679
697     Male  71.434727  206.594059
569     Male  72.500757  201.655056
5062  Female  64.936556  152.419352
1081    Male  69.903211  221.897619
304     Male  65.244143  159.503220
5308  Female  67.041972  139.993374
5133  Female  65.060126  140.985246
2078    Male  69.010219  208.536195
8561  Female  67.700621  155.726683
```

[27]:
```python
#SLIP 19
import pandas as pd
data = {'Name': ['Alice', 'Bob', 'Charlie', 'David', 'Emma', 'Frank', 'Grace',␣
 ↪'Hank', 'Ivy', 'Jack'],
    'Age': [25, 30, 35, 28, 22, 40, 32, 45, 28, 33],
    'Percentage': [80.5, 92.0, 78.3, 89.7, 75.0, 88.5, 95.2, 70.8, 82.3, 91.1]}
df = pd.DataFrame(data)
print("Data Frame:")
print(df)
print("\n")
print("DataFrame Information:")
print(f"Shape: {df.shape}")
print(f"Number of Rows-Columns: {df.shape[0]}-{df.shape[1]}")
print("\nData Types:")
print(df.dtypes)
print("\nFeature Names:")
print(df.columns)
print("\nData Description:")
print(df.describe())
print("\n")
additional_data = {'Name': ['Alice', 'Bob', 'David', 'Emma', 'Hank', 'Bob',␣
 ↪'Grace', 'Ivy', 'Jack', 'Charlie'],
    'Age': [25, 30, None, 22, 45, 30, 32, None, 33, 35],
    'Percentage': [80.5, 92.0, 89.7, None, 70.8, 92.0, 95.2, 82.3, 91.1, 78.3],
    'Remarks': [''] * 10}
df = pd.concat([df, pd.DataFrame(additional_data)], ignore_index=True)
print("Modified DataFrame:")
print(df)
```

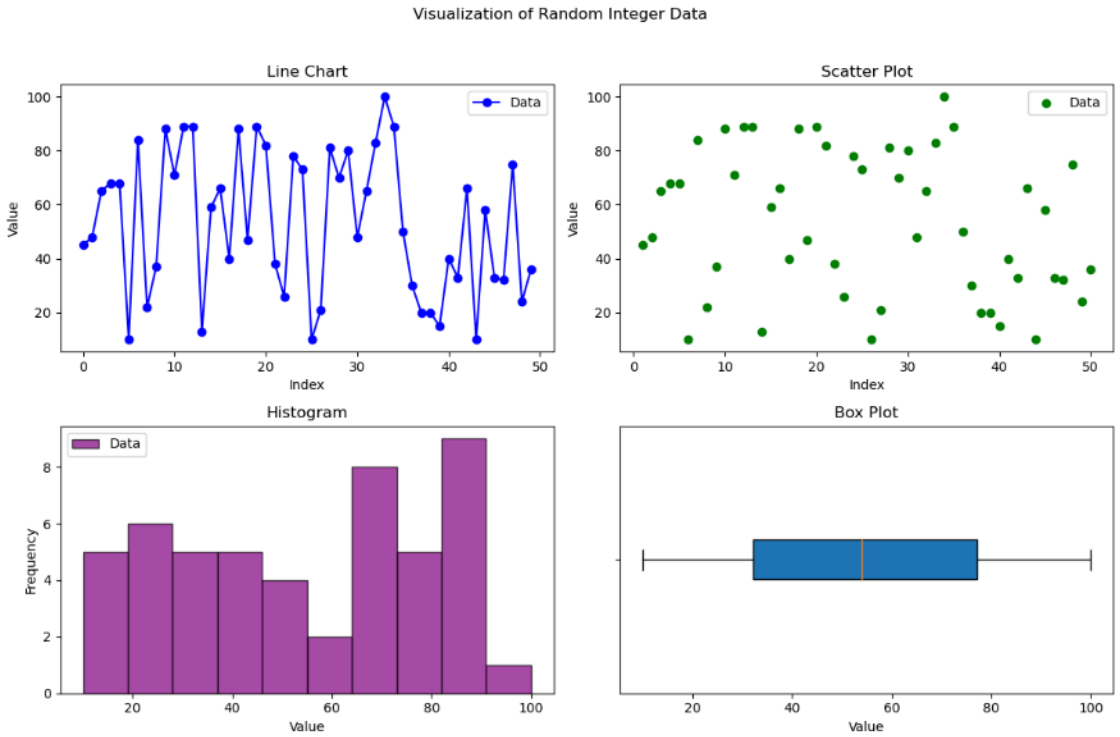
```
Data Frame:
```

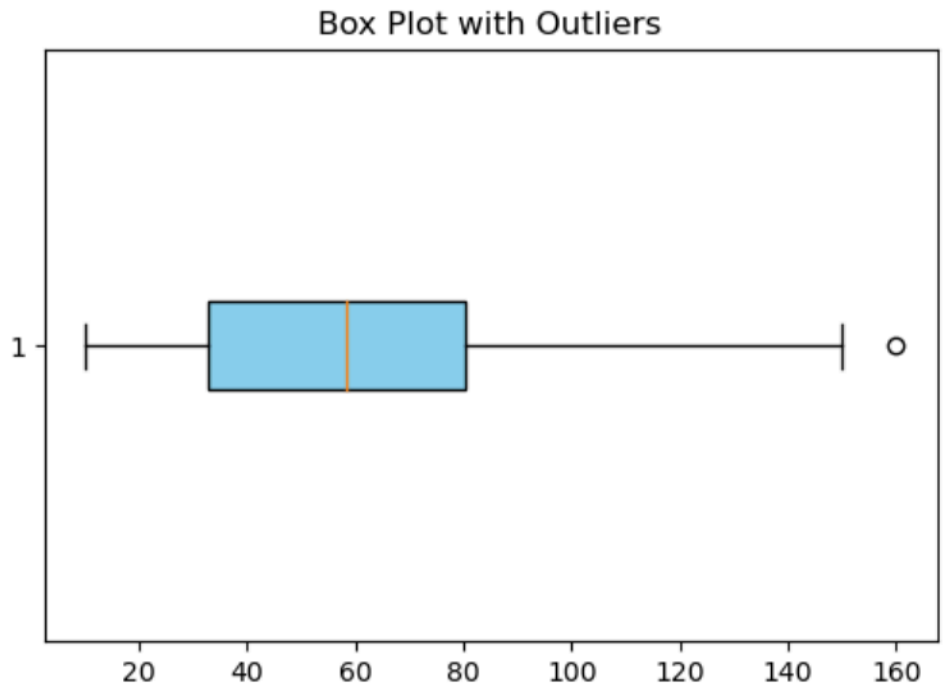
```
      Name  Age  Percentage
0    Alice   25        80.5
1      Bob   30        92.0
2  Charlie   35        78.3
3    David   28        89.7
4     Emma   22        75.0
5    Frank   40        88.5
6    Grace   32        95.2
7     Hank   45        70.8
8      Ivy   28        82.3
9     Jack   33        91.1


DataFrame Information:
Shape: (10, 3)
Number of Rows-Columns: 10-3

Data Types:
Name           object
Age             int64
Percentage    float64
dtype: object

Feature Names:
Index(['Name', 'Age', 'Percentage'], dtype='object')

Data Description:
             Age  Percentage
count  10.00000   10.000000
mean   31.80000   84.340000
std     6.89283    8.126117
min    22.00000   70.800000
25%    28.00000   78.850000
50%    31.00000   85.400000
75%    34.50000   90.750000
max    45.00000   95.200000


Modified DataFrame:
       Name   Age  Percentage Remarks
0     Alice  25.0        80.5     NaN
1       Bob  30.0        92.0     NaN
2   Charlie  35.0        78.3     NaN
3     David  28.0        89.7     NaN
4      Emma  22.0        75.0     NaN
5     Frank  40.0        88.5     NaN
6     Grace  32.0        95.2     NaN
7      Hank  45.0        70.8     NaN
```

```
8       Ivy  28.0        82.3     NaN
9      Jack  33.0        91.1     NaN
10    Alice  25.0        80.5
11      Bob  30.0        92.0
12    David   NaN        89.7
13     Emma  22.0         NaN
14     Hank  45.0        70.8
15      Bob  30.0        92.0
16    Grace  32.0        95.2
17      Ivy   NaN        82.3
18     Jack  33.0        91.1
19  Charlie  35.0        78.3
```

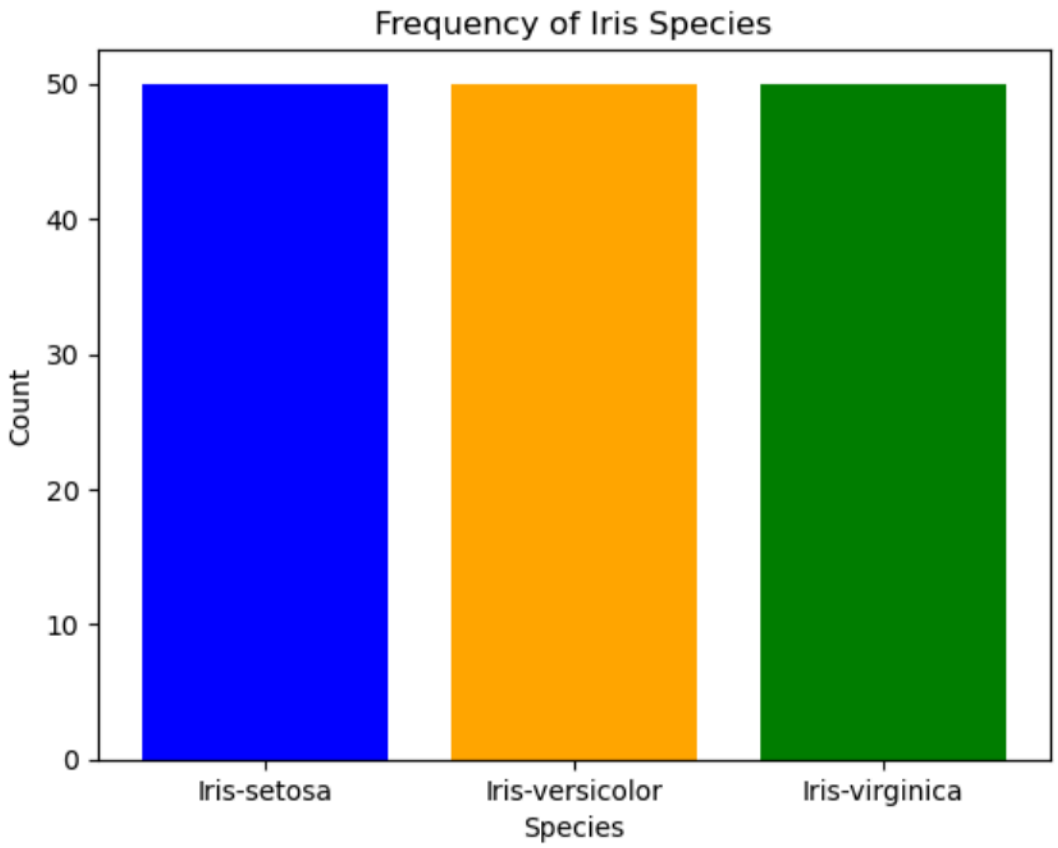
[34]:
```python
#SLIP 20 A
import numpy as np
import matplotlib.pyplot as plt
np.random.seed(0)
data = np.random.randint(1, 101, 50)
fig, axs = plt.subplots(2, 2, figsize=(12, 8))
fig.suptitle("Visualization of Random Integer Data")
# Line chart
axs[0, 0].plot(data, color='blue', marker='o', linestyle='-', label='Data')
axs[0, 0].set_title("Line Chart")
axs[0, 0].set_xlabel("Index")
axs[0, 0].set_ylabel("Value")
axs[0, 0].legend()
# Scatter plot
axs[0, 1].scatter(range(1, 51), data, color='green', label='Data',
marker='o')
axs[0, 1].set_title("Scatter Plot")
axs[0, 1].set_xlabel("Index")
axs[0, 1].set_ylabel("Value")
axs[0, 1].legend()
# Histogram
axs[1, 0].hist(data, bins=10, color='purple', alpha=0.7,
edgecolor='black', label='Data')
axs[1, 0].set_title("Histogram")
axs[1, 0].set_xlabel("Value")
axs[1, 0].set_ylabel("Frequency")
axs[1, 0].legend()
# Box plot
axs[1, 1].boxplot(data, vert=False, patch_artist=True)
axs[1, 1].set_title("Box Plot")
axs[1, 1].set_yticklabels([])
axs[1, 1].set_xlabel("Value")
plt.tight_layout(rect=[0, 0, 1, 0.96])
plt.show()
```

```
#SLIP 20 B


# Add two outliers
outliers = [150, 160]
random_data_with_outliers = np.concatenate([data, outliers])
plt.figure(figsize=(6, 4))
plt.boxplot(random_data_with_outliers, vert=False, patch_artist=True,␣
 ↪boxprops=dict(facecolor='skyblue'))
plt.title('Box Plot with Outliers')
plt.show()
```

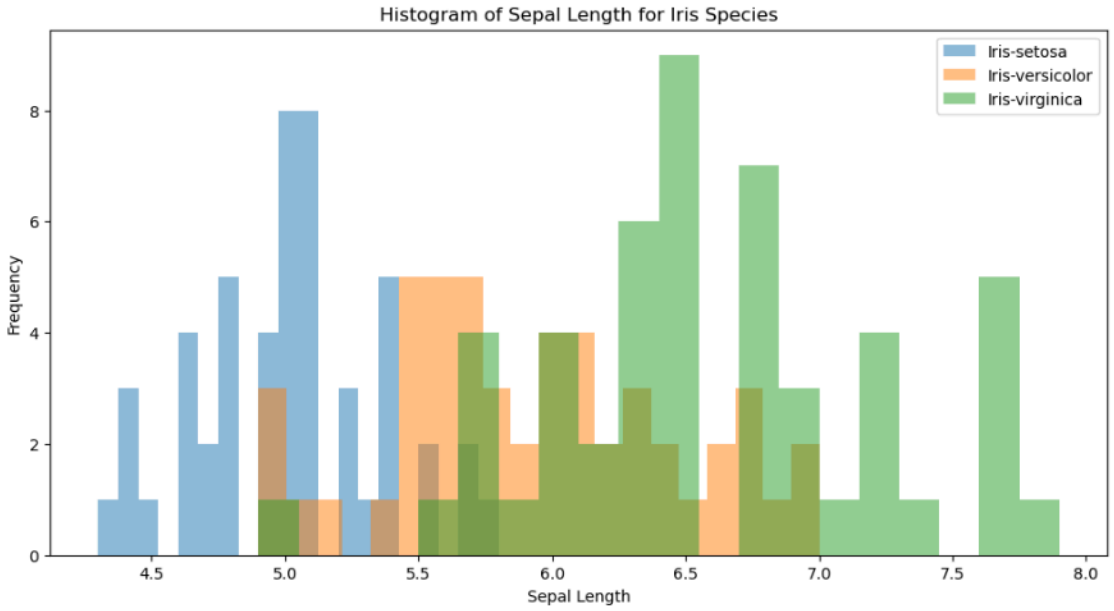




[37]:
```
#SLIP 21 A
import pandas as pd
import matplotlib.pyplot as plt
iris_df = pd.read_csv('IRIS.csv')
species_counts = iris_df['species'].value_counts()
plt.bar(species_counts.index, species_counts.values, color=['blue', 'orange',␣
↪'green'])
plt.title('Frequency of Iris Species')
plt.xlabel('Species')
plt.ylabel('Count')
plt.show()
```



[38]:
```
#SLIP 21 B
import pandas as pd
import matplotlib.pyplot as plt
iris_df = pd.read_csv('IRIS.csv')
plt.figure(figsize=(12, 6))
for species in iris_df['species'].unique():
    species_data = iris_df[iris_df['species'] == species]
    plt.hist(species_data['sepal_length'], bins=20, alpha=0.5, label=species)
plt.title('Histogram of Sepal Length for Iris Species')
plt.xlabel('Sepal Length')
plt.ylabel('Frequency')
plt.legend()
plt.show()
```



[39]:
```python
#SLIP 22 A
import pandas as pd
from sklearn.preprocessing import MinMaxScaler, StandardScaler, Normalizer
dataset = pd.read_csv('winequality-red.csv')
print("Original Dataset:")
print(dataset.head())
print("\n")
X = dataset.drop('quality', axis=1)
# a. Rescaling using MinMaxScaler
minmax_scaler = MinMaxScaler()
rescaled_data = minmax_scaler.fit_transform(X)
rescaled_df = pd.DataFrame(rescaled_data, columns=X.columns)
print("Rescaled Dataset:")
print(rescaled_df.head())
print("\n")
# b. Standardizing Data using StandardScaler
standard_scaler = StandardScaler()
standardized_data = standard_scaler.fit_transform(X)
standardized_df = pd.DataFrame(standardized_data, columns=X.columns)
print("Standardized Dataset:")
print(standardized_df.head())
print("\n")
# c. Normalizing Data using Normalizer
normalizer = Normalizer()
normalized_data = normalizer.fit_transform(X)
normalized_df = pd.DataFrame(normalized_data, columns=X.columns)
print("Normalized Dataset:")
```

```
print(normalized_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


Rescaled Dataset:
   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
0  0.153846
1  0.215385
2  0.215385
3  0.215385
4  0.153846
```

```
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


Normalized Dataset:
   fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
0       0.195153          0.018460     0.000000        0.050107   0.002004
1       0.107241          0.012099     0.000000        0.035747   0.001347
2       0.135457          0.013198     0.000695        0.039942   0.001598
3       0.174367          0.004359     0.008718        0.029580   0.001168
4       0.195153          0.018460     0.000000        0.050107   0.002004

   free sulfur dioxide  total sulfur dioxide   density        pH  sulphates  \
0             0.290092              0.896647  0.026314  0.092566   0.014768
1             0.343722              0.921175  0.013705  0.043996   0.009349
2             0.260494              0.937777  0.017314  0.056614   0.011288
3             0.264664              0.934108  0.015537  0.049196   0.009030
4             0.290092              0.896647  0.026314  0.092566   0.014768

    alcohol
0  0.247896
1  0.134739
2  0.170189
3  0.152571
4  0.247896
```

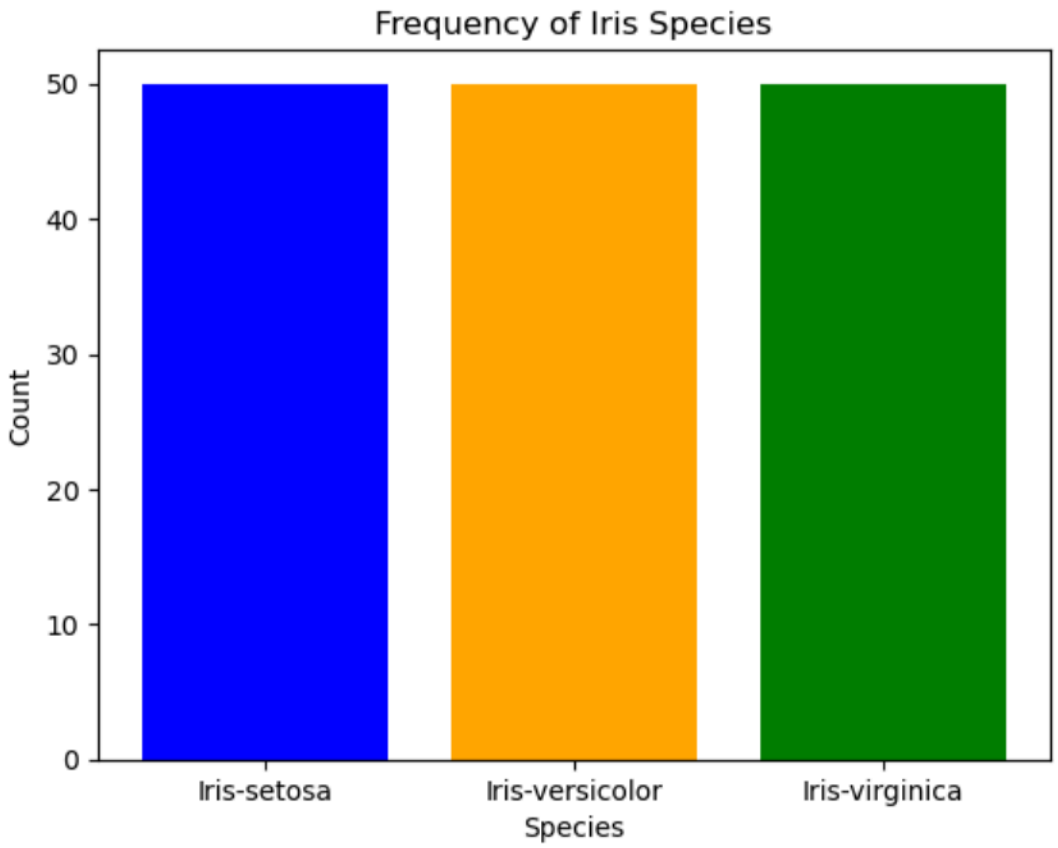
[41]:
```
#SLIP 23
import pandas as pd
from sklearn.preprocessing import MinMaxScaler, StandardScaler, Binarizer
dataset = pd.read_csv('winequality-red.csv')
print("Original Dataset:")
print(dataset.head())
print("\n")
X = dataset.drop('quality', axis=1)
# a. Rescaling using MinMaxScaler
minmax_scaler = MinMaxScaler()
rescaled_data = minmax_scaler.fit_transform(X)
rescaled_df = pd.DataFrame(rescaled_data, columns=X.columns)
print("Rescaled Dataset:")
print(rescaled_df.head())
print("\n")
# b. Standardizing Data using StandardScaler
standard_scaler = StandardScaler()
standardized_data = standard_scaler.fit_transform(X)
standardized_df = pd.DataFrame(standardized_data, columns=X.columns)
print("Standardized Dataset:")
print(standardized_df.head())
print("\n")
# c. Binarizing Data using Binarizer
# binary threshold for example 0.5
binary_threshold = 0.5
binarizer = Binarizer(threshold=binary_threshold)
binarized_data = binarizer.fit_transform(X)

binarized_df = pd.DataFrame(binarized_data, columns=X.columns)
print(f"Binarized Dataset (Threshold: {binary_threshold}):")
print(binarized_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
```

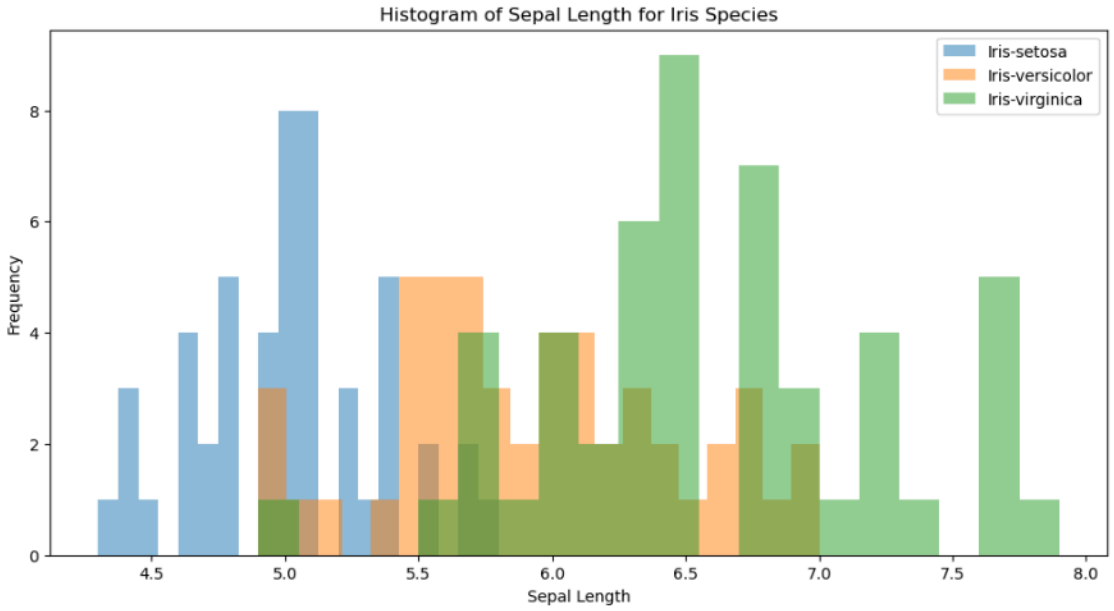
```
   alcohol  quality
0      9.4        5
1      9.8        5
2      9.8        5
3      9.8        6
4      9.4        5


Rescaled Dataset:
   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
0  0.153846
1  0.215385
2  0.215385
3  0.215385
4  0.153846


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


Binarized Dataset (Threshold: 0.5):
   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  alcohol
0                  1.0                   1.0      1.0  1.0        1.0      1.0
1                  1.0                   1.0      1.0  1.0        1.0      1.0
2                  1.0                   1.0      1.0  1.0        1.0      1.0
3                  1.0                   1.0      1.0  1.0        1.0      1.0
4                  1.0                   1.0      1.0  1.0        1.0      1.0
```

[42]:
```
#SLIP 24 A
import pandas as pd
import matplotlib.pyplot as plt
iris_df = pd.read_csv('IRIS.csv')
species_counts = iris_df['species'].value_counts()
plt.bar(species_counts.index, species_counts.values, color=['blue', 'orange',␣
 ↪'green'])
plt.title('Frequency of Iris Species')
plt.xlabel('Species')
plt.ylabel('Count')
plt.show()
```

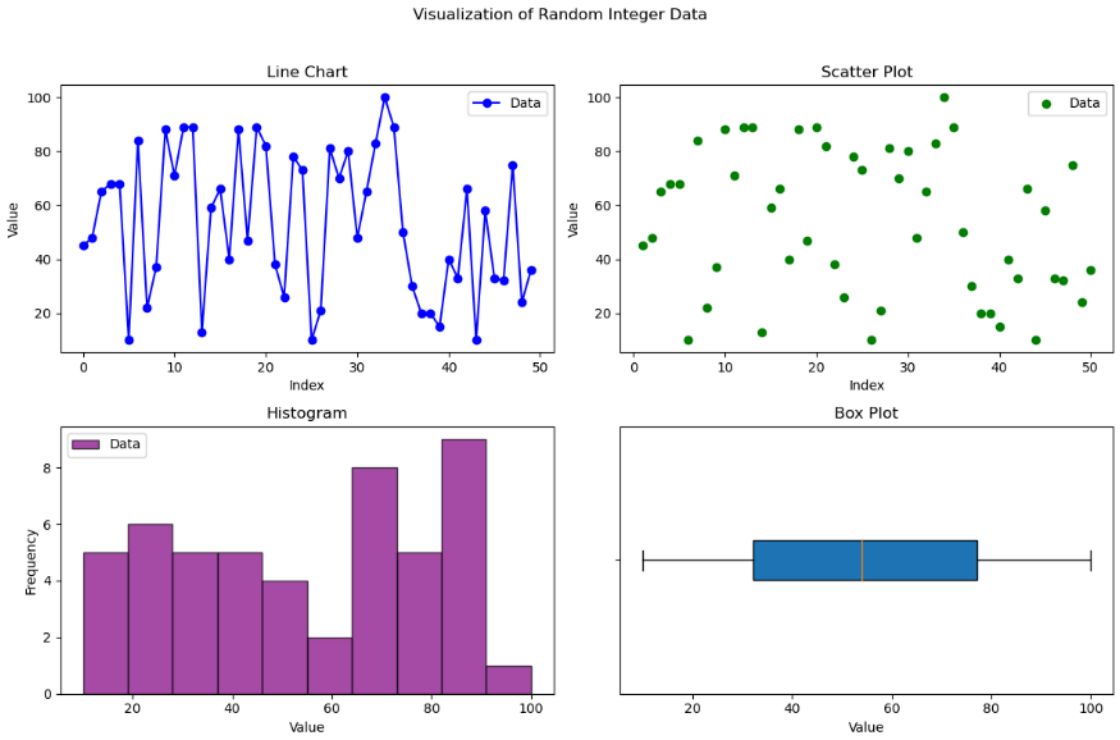


[43]:
```
#SLIP 24 B
import pandas as pd
import matplotlib.pyplot as plt
iris_df = pd.read_csv('IRIS.csv')
plt.figure(figsize=(12, 6))
for species in iris_df['species'].unique():
    species_data = iris_df[iris_df['species'] == species]
    plt.hist(species_data['sepal_length'], bins=20, alpha=0.5, label=species)
plt.title('Histogram of Sepal Length for Iris Species')
plt.xlabel('Sepal Length')
plt.ylabel('Frequency')
plt.legend()
plt.show()
```

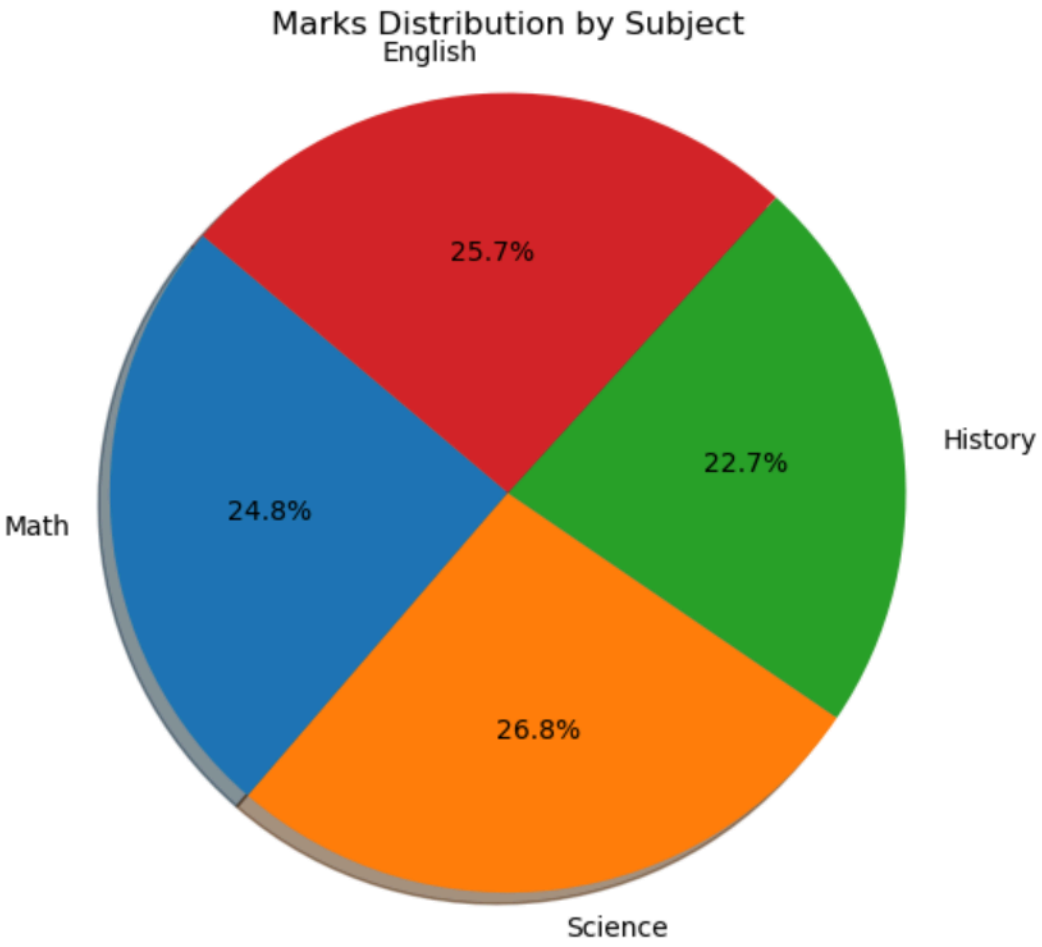


[44]:
```
#SLIP 25 A
import numpy as np
import matplotlib.pyplot as plt
np.random.seed(0)
data = np.random.randint(1, 101, 50)
fig, axs = plt.subplots(2, 2, figsize=(12, 8))
fig.suptitle("Visualization of Random Integer Data")
# Line chart
axs[0, 0].plot(data, color='blue', marker='o', linestyle='-', label='Data')
axs[0, 0].set_title("Line Chart")
axs[0, 0].set_xlabel("Index")
axs[0, 0].set_ylabel("Value")
axs[0, 0].legend()
# Scatter plot
axs[0, 1].scatter(range(1, 51), data, color='green', label='Data',
marker='o')
axs[0, 1].set_title("Scatter Plot")
axs[0, 1].set_xlabel("Index")
axs[0, 1].set_ylabel("Value")
axs[0, 1].legend()
# Histogram
axs[1, 0].hist(data, bins=10, color='purple', alpha=0.7,
edgecolor='black', label='Data')
axs[1, 0].set_title("Histogram")
axs[1, 0].set_xlabel("Value")
axs[1, 0].set_ylabel("Frequency")
axs[1, 0].legend()
```

```
# Box plot
axs[1, 1].boxplot(data, vert=False, patch_artist=True)
axs[1, 1].set_title("Box Plot")
axs[1, 1].set_yticklabels([])
axs[1, 1].set_xlabel("Value")
plt.tight_layout(rect=[0, 0, 1, 0.96])
plt.show()
```

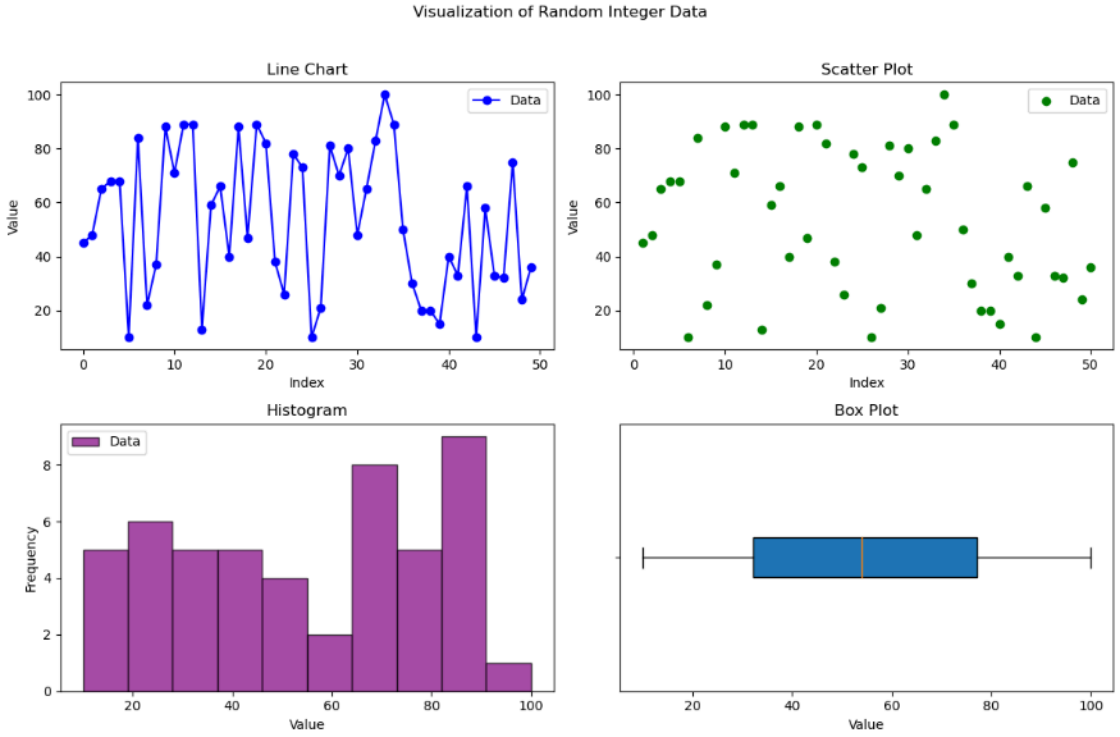


[46]:
```
#SLIP 25 B
import matplotlib.pyplot as plt
subjects = ["Math", "Science", "History", "English"]
marks = [85, 92, 78, 88]
plt.figure(figsize=(6, 6))
plt.pie(marks, labels=subjects, autopct='%1.1f%%', startangle=140,
shadow=True)
plt.title("Marks Distribution by Subject")
plt.axis('equal')
plt.show()
```



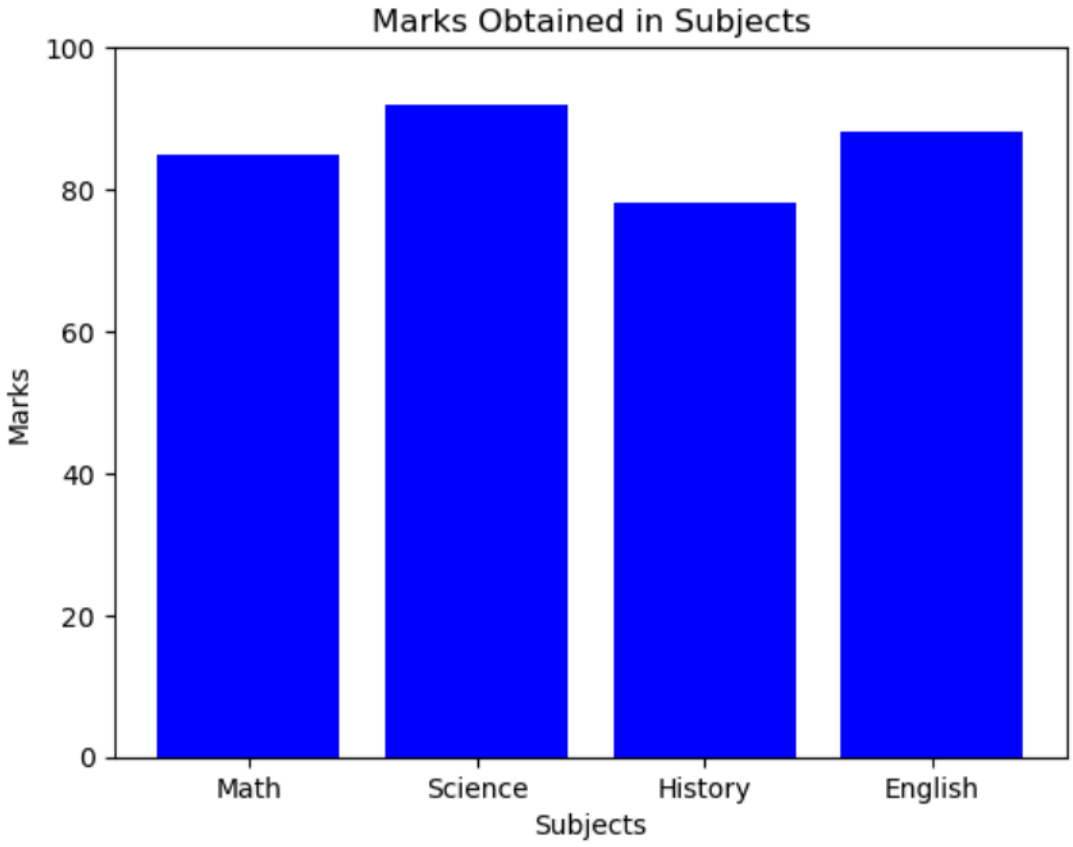
[47]:
```
#SLIP 26 A
import numpy as np
import matplotlib.pyplot as plt
np.random.seed(0)
data = np.random.randint(1, 101, 50)
fig, axs = plt.subplots(2, 2, figsize=(12, 8))
fig.suptitle("Visualization of Random Integer Data")
# Line chart
axs[0, 0].plot(data, color='blue', marker='o', linestyle='-', label='Data')
axs[0, 0].set_title("Line Chart")
axs[0, 0].set_xlabel("Index")
axs[0, 0].set_ylabel("Value")
axs[0, 0].legend()
# Scatter plot
axs[0, 1].scatter(range(1, 51), data, color='green', label='Data',
marker='o')
axs[0, 1].set_title("Scatter Plot")
```

```
axs[0, 1].set_xlabel("Index")
axs[0, 1].set_ylabel("Value")
axs[0, 1].legend()
# Histogram
axs[1, 0].hist(data, bins=10, color='purple', alpha=0.7,
edgecolor='black', label='Data')
axs[1, 0].set_title("Histogram")
axs[1, 0].set_xlabel("Value")
axs[1, 0].set_ylabel("Frequency")
axs[1, 0].legend()
# Box plot
axs[1, 1].boxplot(data, vert=False, patch_artist=True)
axs[1, 1].set_title("Box Plot")
axs[1, 1].set_yticklabels([])
axs[1, 1].set_xlabel("Value")
plt.tight_layout(rect=[0, 0, 1, 0.96])
plt.show()
```



[48]:
```
#SLIP 26 B
import matplotlib.pyplot as plt
subjects = ['Math', 'Science', 'History', 'English']
marks = [85, 92, 78, 88]
plt.bar(subjects, marks, color='blue')
```

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



[49]:
```
#SLIP 27 A
import pandas as pd
from sklearn.preprocessing import LabelEncoder
file_path = "Data.csv"
df = pd.read_csv(file_path)
# Apply One-Hot Encoding to the "Country" column
df = pd.get_dummies(df, columns=['Country'], prefix=['Country'])
print("Modified DataFrame:")
print(df)
```

```
Modified DataFrame:
     Age   Salary Purchased  Country_China  Country_India  Country_Sri lanka
0   34.0  92000.0       Yes              0              1                   0
1   22.0  25000.0       Yes              0              0                   1
```

```
2   31.0  74000.0       Yes              1              0              0
3   29.0      NaN        No              0              0              1
4   55.0  98000.0       Yes              1              0              0
5   24.0  30000.0        No              0              1              0
6   28.0  40000.0        No              0              0              1
7    NaN  60000.0        No              0              1              0
8   51.0  89000.0       Yes              1              0              0
9   44.0  78000.0       Yes              0              1              0
10  21.0  20000.0        No              0              0              1
11  25.0  30000.0       Yes              1              0              0
12  33.0  45000.0       Yes              0              1              0
13  42.0  65000.0       Yes              0              1              0
14  33.0  22000.0        No              0              0              1
```

[52]:
```python
#SLIP 27 B
import pandas as pd
from sklearn.preprocessing import LabelEncoder
file_path = "Data.csv"
df = pd.read_csv(file_path)
# Apply Label Encoding to the "Purchased" column
label_encoder = LabelEncoder()
df['Purchased'] = label_encoder.fit_transform(df['Purchased'])
print("Modified DataFrame:")
print(df)
```

```
Modified DataFrame:
      Country   Age   Salary  Purchased
0       India  34.0  92000.0          1
1   Sri lanka  22.0  25000.0          1
2       China  31.0  74000.0          1
3   Sri lanka  29.0      NaN          0
4       China  55.0  98000.0          1
5       India  24.0  30000.0          0
6   Sri lanka  28.0  40000.0          0
7       India   NaN  60000.0          0
8       China  51.0  89000.0          1
9       India  44.0  78000.0          1
10  Sri lanka  21.0  20000.0          0
11      China  25.0  30000.0          1
12      India  33.0  45000.0          1
13      India  42.0  65000.0          1
14  Sri lanka  33.0  22000.0          0
```

[55]:
```python
#SLIP 28
import pandas as pd
#1
```

```
data = {'Name': ['Alice', 'Bob', 'Charlie', 'David', 'Emma', 'Frank', 'Grace',␣
 ↪'Hank', 'Ivy', 'Jack'],
    'Age': [25, 30, 35, 28, 22, 40, 32, 45, 28, 33],
    'Percentage': [80.5, 92.0, 78.3, 89.7, 75.0, 88.5, 95.2, 70.8, 82.3, 91.1]}
df = pd.DataFrame(data)
print("Data Frame:")
print(df)
print("\n")

#2
print("DataFrame Information:")
print(f"Shape: {df.shape}")
print(f"Number of Rows-Columns: {df.shape[0]}-{df.shape[1]}")
print("\nData Types:")
print(df.dtypes)
print("\nFeature Names:")
print(df.columns)
print("\nData Description:")
print(df.describe())
print("\n")

#3
print("Statistical Details of the Data:")
print(df.describe())
print("\n")

#4
additional_data = {'Name': ['Alice', 'Bob', 'David', 'Emma', 'Hank', 'Bob',␣
 ↪'Grace', 'Ivy', 'Jack', 'Charlie'],
    'Age': [25, 30, None, 22, 45, 30, 32, None, 33, 35],
    'Percentage': [80.5, 92.0, 89.7, None, 70.8, 92.0, 95.2, 82.3, 91.1, 78.3],
    'Remarks': [''] * 10}
df = pd.concat([df, pd.DataFrame(additional_data)], ignore_index=True)
print("Modified DataFrame:")
print(df)
```

```
Data Frame:
      Name  Age  Percentage
0    Alice   25        80.5
1      Bob   30        92.0
2  Charlie   35        78.3
3    David   28        89.7
4     Emma   22        75.0
5    Frank   40        88.5
6    Grace   32        95.2
7     Hank   45        70.8
8      Ivy   28        82.3
```

```
9     Jack   33        91.1


DataFrame Information:
Shape: (10, 3)
Number of Rows-Columns: 10-3

Data Types:
Name           object
Age             int64
Percentage    float64
dtype: object

Feature Names:
Index(['Name', 'Age', 'Percentage'], dtype='object')

Data Description:
             Age  Percentage
count  10.00000   10.000000
mean   31.80000   84.340000
std     6.89283    8.126117
min    22.00000   70.800000
25%    28.00000   78.850000
50%    31.00000   85.400000
75%    34.50000   90.750000
max    45.00000   95.200000


Statistical Details of the Data:
             Age  Percentage
count  10.00000   10.000000
mean   31.80000   84.340000
std     6.89283    8.126117
min    22.00000   70.800000
25%    28.00000   78.850000
50%    31.00000   85.400000
75%    34.50000   90.750000
max    45.00000   95.200000


Modified DataFrame:
       Name   Age  Percentage Remarks
0     Alice  25.0        80.5     NaN
1       Bob  30.0        92.0     NaN
2   Charlie  35.0        78.3     NaN
3     David  28.0        89.7     NaN
4      Emma  22.0        75.0     NaN
5     Frank  40.0        88.5     NaN
```

```
6     Grace  32.0        95.2     NaN
7      Hank  45.0        70.8     NaN
8       Ivy  28.0        82.3     NaN
9      Jack  33.0        91.1     NaN
10    Alice  25.0        80.5
11      Bob  30.0        92.0
12    David   NaN        89.7
13     Emma  22.0         NaN
14     Hank  45.0        70.8
15      Bob  30.0        92.0
16    Grace  32.0        95.2
17      Ivy   NaN        82.3
18     Jack  33.0        91.1
19  Charlie  35.0        78.3
```

[57]:
```python
#SLIP 29 A
import pandas as pd
from sklearn.preprocessing import LabelEncoder
file_path = "Data.csv"
df = pd.read_csv(file_path)
# Apply One-Hot Encoding to the "Country" column
df = pd.get_dummies(df, columns=['Country'], prefix=['Country'])
print("Modified DataFrame:")
print(df)
```

```
Modified DataFrame:
     Age   Salary Purchased  Country_China  Country_India  Country_Sri lanka
0   34.0  92000.0       Yes              0              1                   0
1   22.0  25000.0       Yes              0              0                   1
2   31.0  74000.0       Yes              1              0                   0
3   29.0      NaN        No              0              0                   1
4   55.0  98000.0       Yes              1              0                   0
5   24.0  30000.0        No              0              1                   0
6   28.0  40000.0        No              0              0                   1
7    NaN  60000.0        No              0              1                   0
8   51.0  89000.0       Yes              1              0                   0
9   44.0  78000.0       Yes              0              1                   0
10  21.0  20000.0        No              0              0                   1
11  25.0  30000.0       Yes              1              0                   0
12  33.0  45000.0       Yes              0              1                   0
13  42.0  65000.0       Yes              0              1                   0
14  33.0  22000.0        No              0              0                   1
```

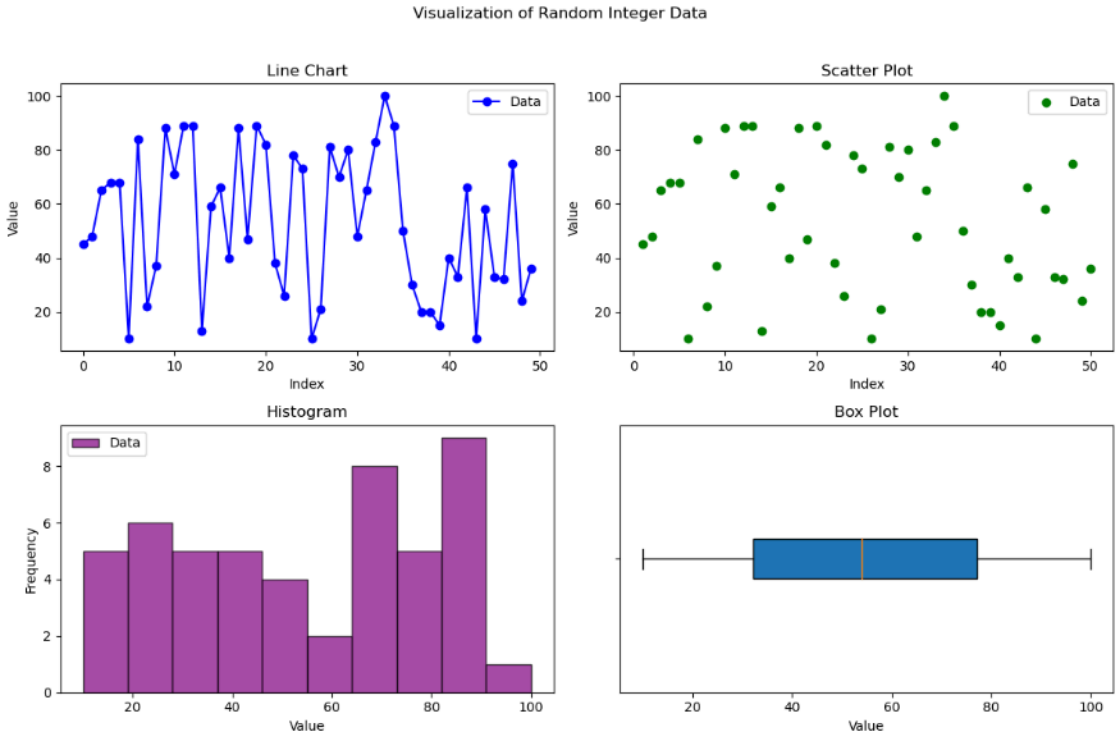
[58]:
```python
#SLIP 29 B
import pandas as pd
from sklearn.preprocessing import LabelEncoder
file_path = "Data.csv"
```

```
df = pd.read_csv(file_path)
# Apply Label Encoding to the "Purchased" column
label_encoder = LabelEncoder()
df['Purchased'] = label_encoder.fit_transform(df['Purchased'])
print("Modified DataFrame:")
print(df)
```

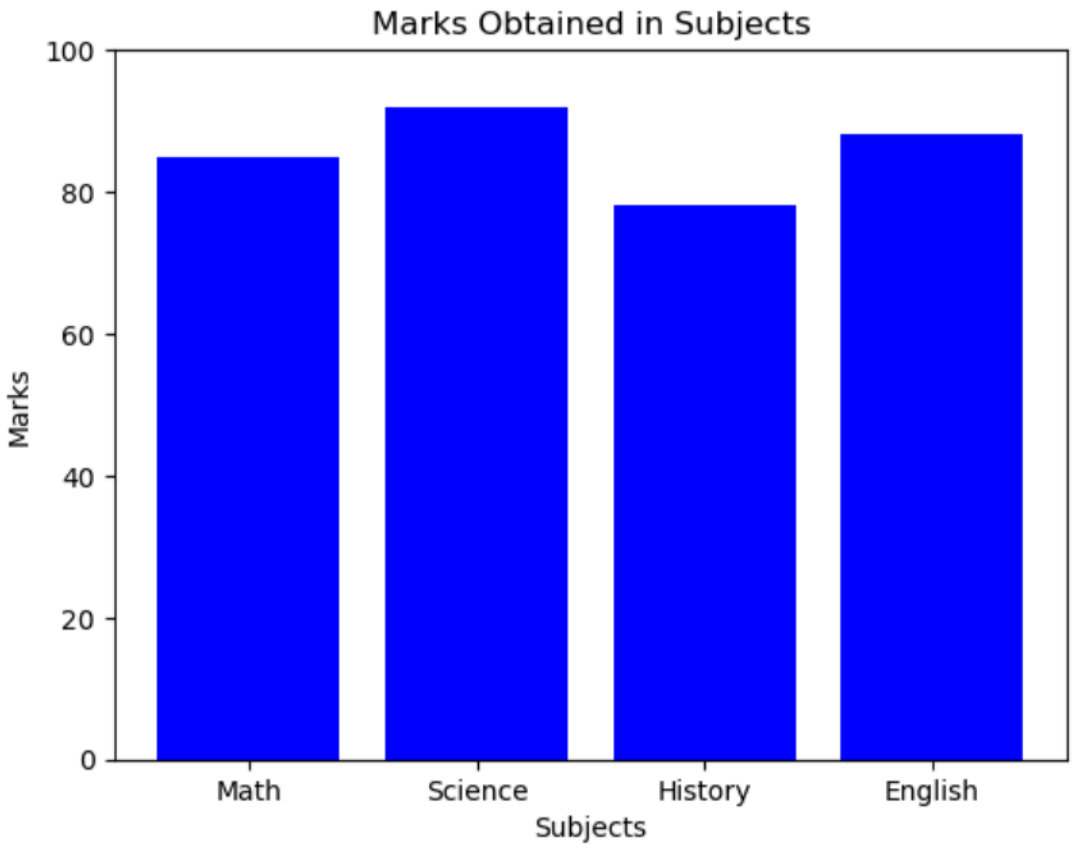
```
Modified DataFrame:
      Country   Age   Salary  Purchased
0       India  34.0  92000.0          1
1   Sri lanka  22.0  25000.0          1
2       China  31.0  74000.0          1
3   Sri lanka  29.0      NaN          0
4       China  55.0  98000.0          1
5       India  24.0  30000.0          0
6   Sri lanka  28.0  40000.0          0
7       India   NaN  60000.0          0
8       China  51.0  89000.0          1
9       India  44.0  78000.0          1
10  Sri lanka  21.0  20000.0          0
11      China  25.0  30000.0          1
12      India  33.0  45000.0          1
13      India  42.0  65000.0          1
14  Sri lanka  33.0  22000.0          0
```

[59]:
```
#SLIP 30 A
import numpy as np
import matplotlib.pyplot as plt
np.random.seed(0)
data = np.random.randint(1, 101, 50)
fig, axs = plt.subplots(2, 2, figsize=(12, 8))
fig.suptitle("Visualization of Random Integer Data")
# Line chart
axs[0, 0].plot(data, color='blue', marker='o', linestyle='-', label='Data')
axs[0, 0].set_title("Line Chart")
axs[0, 0].set_xlabel("Index")
axs[0, 0].set_ylabel("Value")
axs[0, 0].legend()
# Scatter plot
axs[0, 1].scatter(range(1, 51), data, color='green', label='Data',
marker='o')
axs[0, 1].set_title("Scatter Plot")
axs[0, 1].set_xlabel("Index")
axs[0, 1].set_ylabel("Value")
axs[0, 1].legend()
# Histogram
axs[1, 0].hist(data, bins=10, color='purple', alpha=0.7,
```

```
edgecolor='black', label='Data')
axs[1, 0].set_title("Histogram")
axs[1, 0].set_xlabel("Value")
axs[1, 0].set_ylabel("Frequency")
axs[1, 0].legend()
# Box plot
axs[1, 1].boxplot(data, vert=False, patch_artist=True)
axs[1, 1].set_title("Box Plot")
axs[1, 1].set_yticklabels([])
axs[1, 1].set_xlabel("Value")
plt.tight_layout(rect=[0, 0, 1, 0.96])
plt.show()
```



[60]:
```
#SLIP 30 B
import matplotlib.pyplot as plt
subjects = ['Math', 'Science', 'History', 'English']
marks = [85, 92, 78, 88]
plt.bar(subjects, marks, color='blue')
plt.title('Marks Obtained in Subjects')
plt.xlabel('Subjects')
plt.ylabel('Marks')
plt.ylim(0, 100)
plt.show()
```


Marks Obtained in Subjects
100
80
60
40
20
0
Marks
Math
Science
History
English
Subjects

[ ]: