Analysis and Systems of Big Data Practise

Lab - 2

Sreepathy Jayanand
CED17I038

Q1

On New Year's Eve, Tina walked into a random shop and surprised to see a huge crowd there. She is interested to find what kind of products they sell the most, for which she needs the age distribution of customers. Help her to find out the same using histogram. The age details of the customers are given below 7, 9, 27, 28, 55, 45, 34, 65, 54, 67, 34, 23, 24, 66, 53, 45, 44, 88, 22, 33, 55, 35, 33, 37, 47, 41,31, 30, 29, 12. Identify the type of histogram (eg. Bimodal, Multimodal, Skewed..etc). Use different bin sizes.

Logic : The age of the customers are plotted with values for the bins (10, 20, 30)

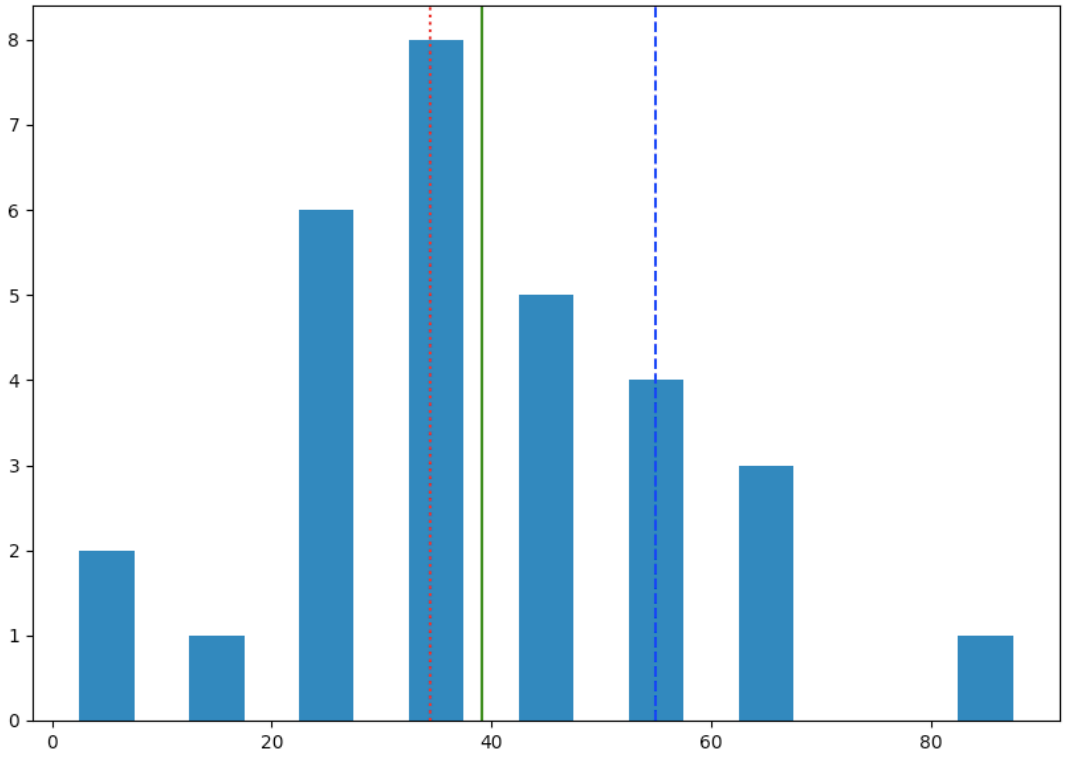
Libraries used:

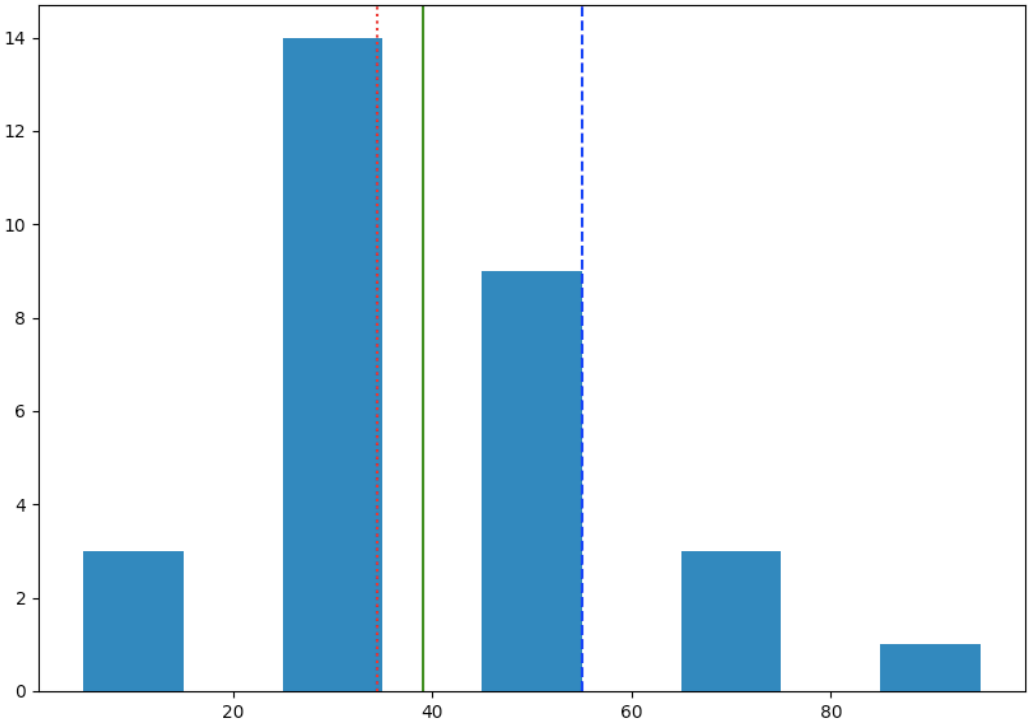
- Statistics - For mean, median, mode, standard deviation
- Matplotlib - For constructing the histogram

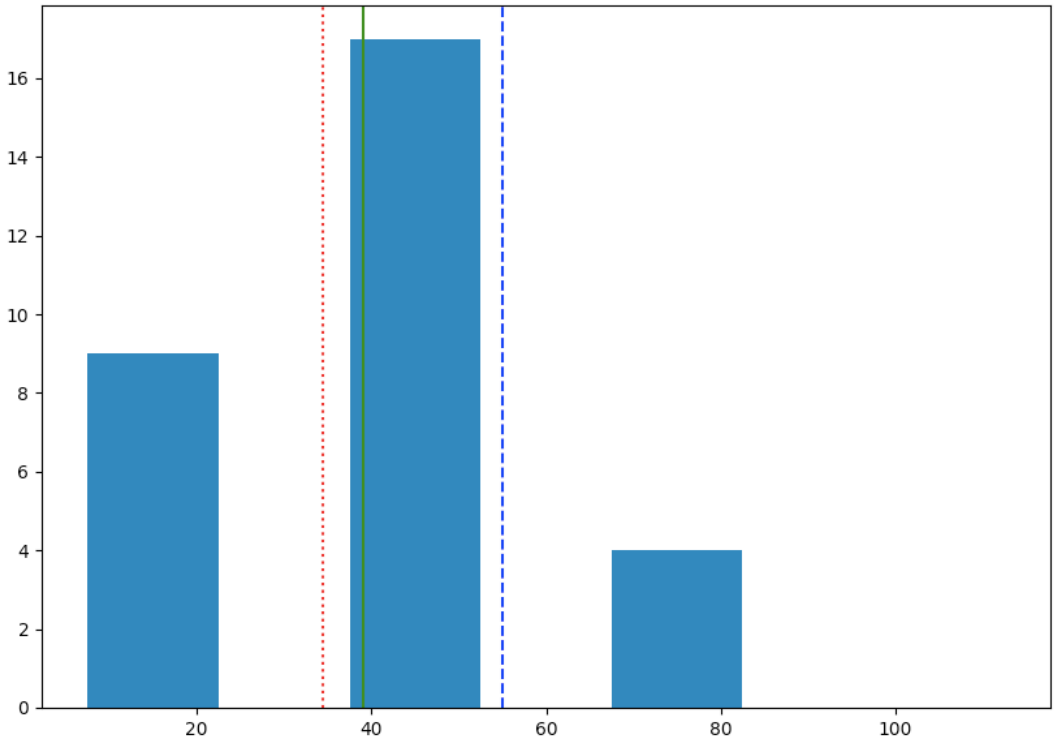
Code:

```
import statistics
from matplotlib import pyplot as plt
x = [7, 9, 27, 28, 55, 45, 34, 65, 54, 67, 34, 23, 24, 66, 53, 45, 44, 88,
 22, 33, 55, 35, 33, 37, 47, 41,31, 30, 29, 12]
x_mean = statistics.mean(x)
x_median = statistics.median(x)
x_mode = statistics.mode(x)
x_stdev = statistics.stdev(x)
print("The mean is", x_mean)
print("The median and mode is", x_median, "\b, ", x_mode)
print("The standard deviation is", x_stdev)
print("The measure of skewness is", (x_mean - x_mode) / x_stdev)
fig, ax = plt.subplots(figsize =(10, 7))
ax.hist(x, bins = [0, 30, 60, 90, 120], histtype = 'bar', rwidth = .5)
plt.axvline(x_median, color = 'r', linestyle = ':')
plt.axvline(x_mode, color = 'b', linestyle = '--')
plt.axvline(x_mean, color = 'g', linestyle = '-')
plt.show()
```

Output:


0
1
2
3
4
5
6
7
8
0
20
40
60
80


0
2
4
6
8
10
12
14
20
40
60
80



We can see that this distribution is a unimodal distribution.

Q2 : A Coach tracked the number of points that each of his 30 players on the team had in one game. The points scored by each player is given below. Visualize the data using ordered stem-leaf plot and also detect the outliers and shape of the distribution.22, 21, 24, 19, 27, 28, 24, 25, 29, 28, 26, 31, 28, 27, 22, 39, 20, 10, 26, 24, 27, 28, 26, 28, 18, 32, 29, 25, 31, 27.

Logic : The points are plotted using box plot

Libraries used:

- Pandas - To make a series from the array
- Stemographic - To construct the stem and leaf plot

Code:

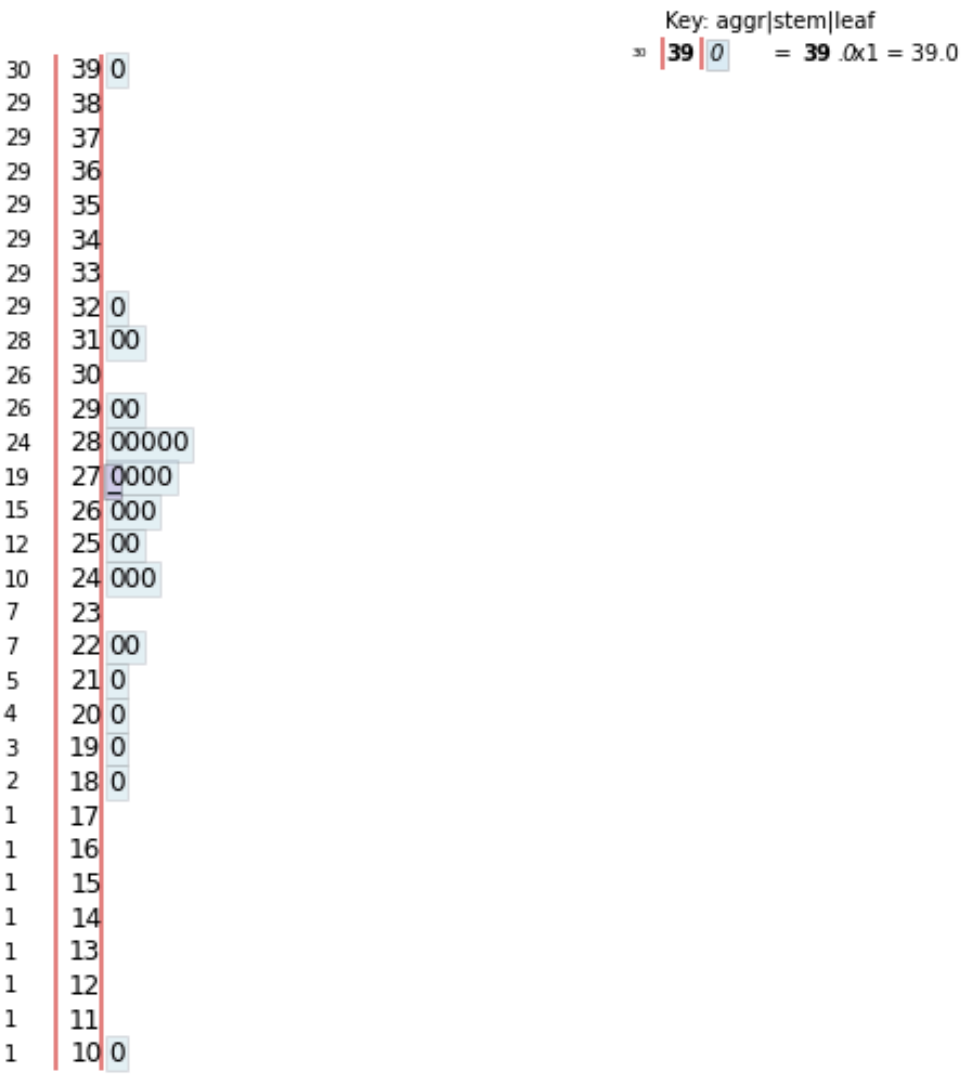
```
import pandas as pd
import stemgraphic

x = [22, 21, 24, 19, 27, 28, 24, 25, 29, 28, 26, 31, 28, 27,
22, 39, 20, 10, 26, 24, 27, 28, 26, 28, 18, 32, 29, 25, 31, 27]
y = pd.Series(x)

fig, ax = stemgraphic.stem_graphic(y)

input('Press ENTER to exit')
```

Output:



The outliers can be clearly seen and are 10 and 39. The distribution is a normal distribution following from its bulged centre and tapering ends.

Q3 : For a sample space of 15 people, a statistician wanted to know the consumption of water and other beverages. He collected their average consumption of water and beverages for 30 days (in litres). Help him to visualize the data using density plot, rug plot and identify the mean, median, mode and skewness of the data from the plot.

Logic : The plots for the graphs using the density plot and mean, median, skewness are calculated and drawn in the plot.

Libraries used:

- Matplotlib - For the various plots.
- Numpy - For the handling of arrays
- Seaborn - For the distplot()
- Statistics - For the mean, median, mode and skewness

Code:

```
from matplotlib import pyplot as plt
import numpy as np
import seaborn as sns
import statistics

a =[3.2, 3.5, 3.6, 2.5, 2.8, 5.9, 2.9, 3.9, 4.9, 6.9, 7.9, 8.0, 3.3, 6.6, 4.4]
b=[2.2, 2.5, 2.6, 1.5, 3.8, 1.9, 0.9, 3.9, 4.9, 6.9, 0.1, 8.0, 0.3, 2.6, 1.4]

mean_a=statistics.mean(a)
median_a=statistics.median(a)
mode_a=statistics.mode(a)
stdev_a=statistics.stdev(a)
skewness_a=(mean_a - mode_a)/stdev_a
mean_b=statistics.mean(b)
median_b=statistics.median(b)
mode_b=statistics.mode(b)
stdev_b=statistics.stdev(b)
skewness_b=(mean_b - mode_b)/stdev_b


sns.set_style('whitegrid')
sns.distplot(np.array(a), hist=False, kde=True,bins=10,
    color ='red',kde_kws = {'linewidth': 2},
    label='Water',rug=True, rug_kws={'color':'red'})
sns.distplot(np.array(b), hist=False, kde=True,bins=10,
    color ='blue',kde_kws = {'linewidth': 2},
    label='Beverages',rug=True, rug_kws={'color':'blue'})
plt.axvline(mean_a, color='g', linestyle=':')
plt.axvline(median_a, color='b', linestyle=':')
plt.axvline(mode_a, color='y', linestyle=':')
plt.axvline(skewness_a, color='r', linestyle=':')
plt.axvline(mean_b, color='g', linestyle='-')
plt.axvline(median_b, color='b', linestyle='-')
plt.axvline(mode_b, color='y', linestyle='-')
plt.axvline(skewness_b, color='r', linestyle='-')
plt.title('Density plot and rug plot')
plt.xlabel('Consumption')
plt.ylabel('Density')
plt.show()
```

Output:

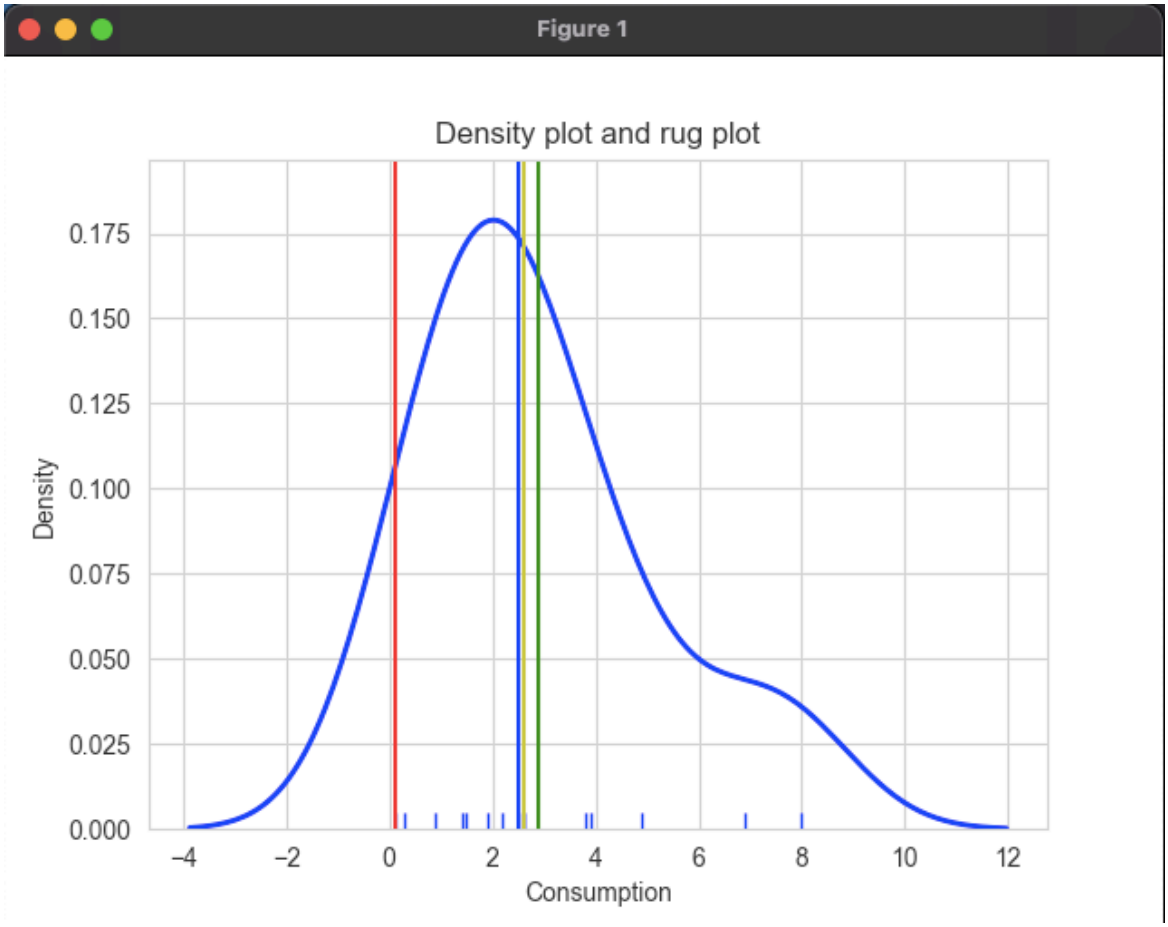


Figure 1
Density plot and rug plot
Density
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
−4
−2
0
2
4
6
8
10
12
Consumption

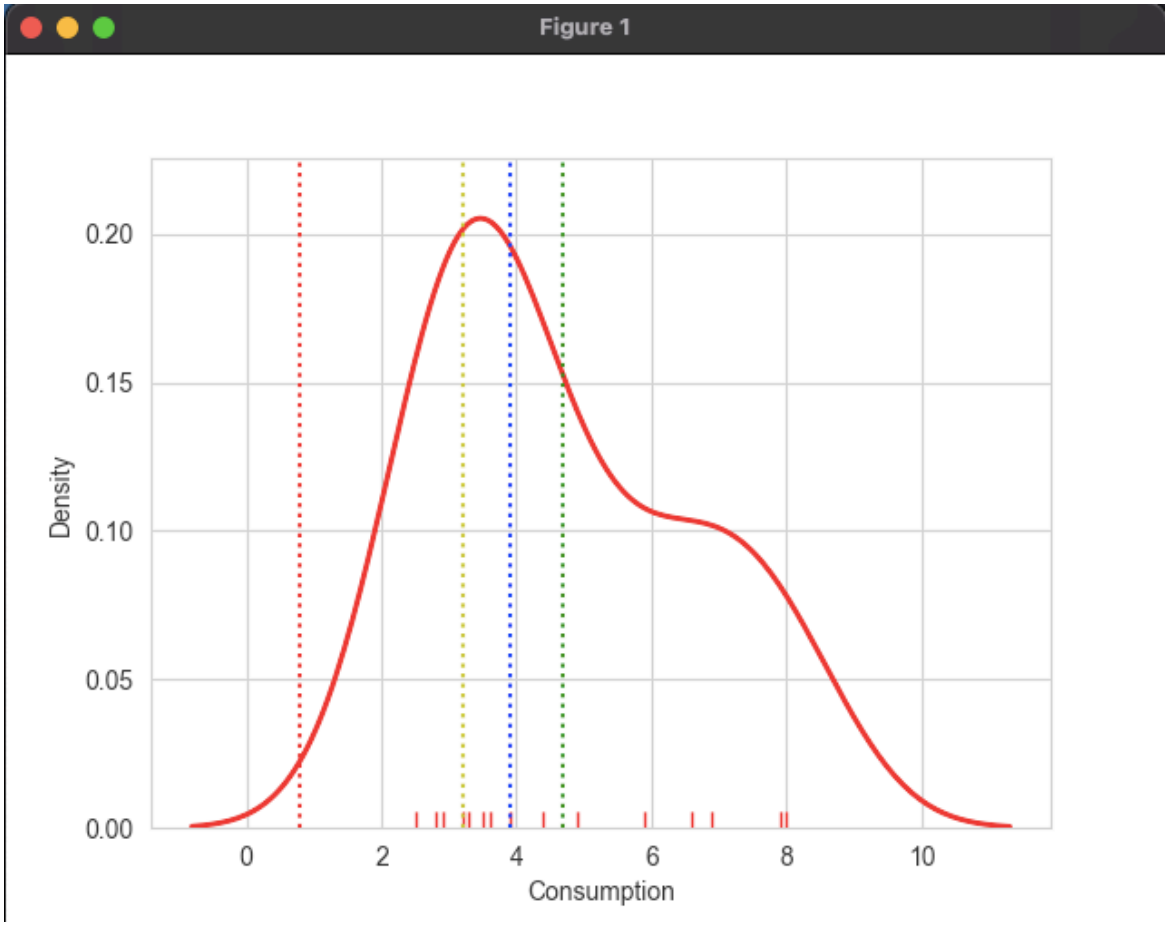


Figure 1
0.20
0.15
0.10
0.05
0.00
Density
0
2
4
6
8
10
Consumption

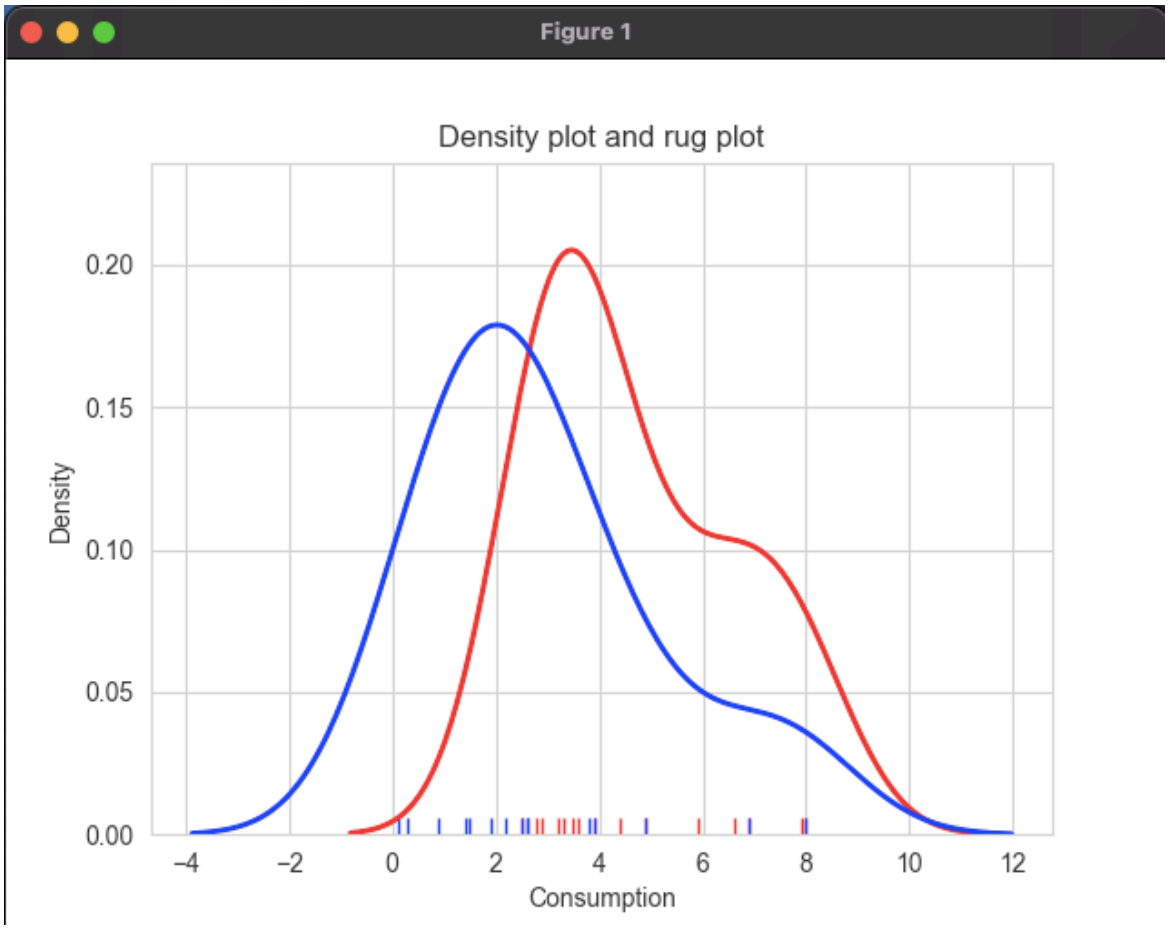


Figure 1
Density plot and rug plot
0.20
0.15
0.10
0.05
0.00
Density
−4
−2
0
2
4
6
8
10
12
Consumption

Q4 : A car company wants to predict how much fuel different cars will use based on their masses. They took a sample of cars, drove each car 100km, and measured how much fuel was used in each case (in litres). Visualize the data using scatterplot and also find co-relation between the 2 variables (eg. Positive/ /Negative, Linear/ Non-linear co-relation) The data is summarized in the table below. (Use a reasonable scale on both axes and put the explanatory variable on the x-axis.)

Logic : The plot fuel v/s mass is plotted using scatter plot.

Libraries used:

- Matplotlib - For the scatter plot.

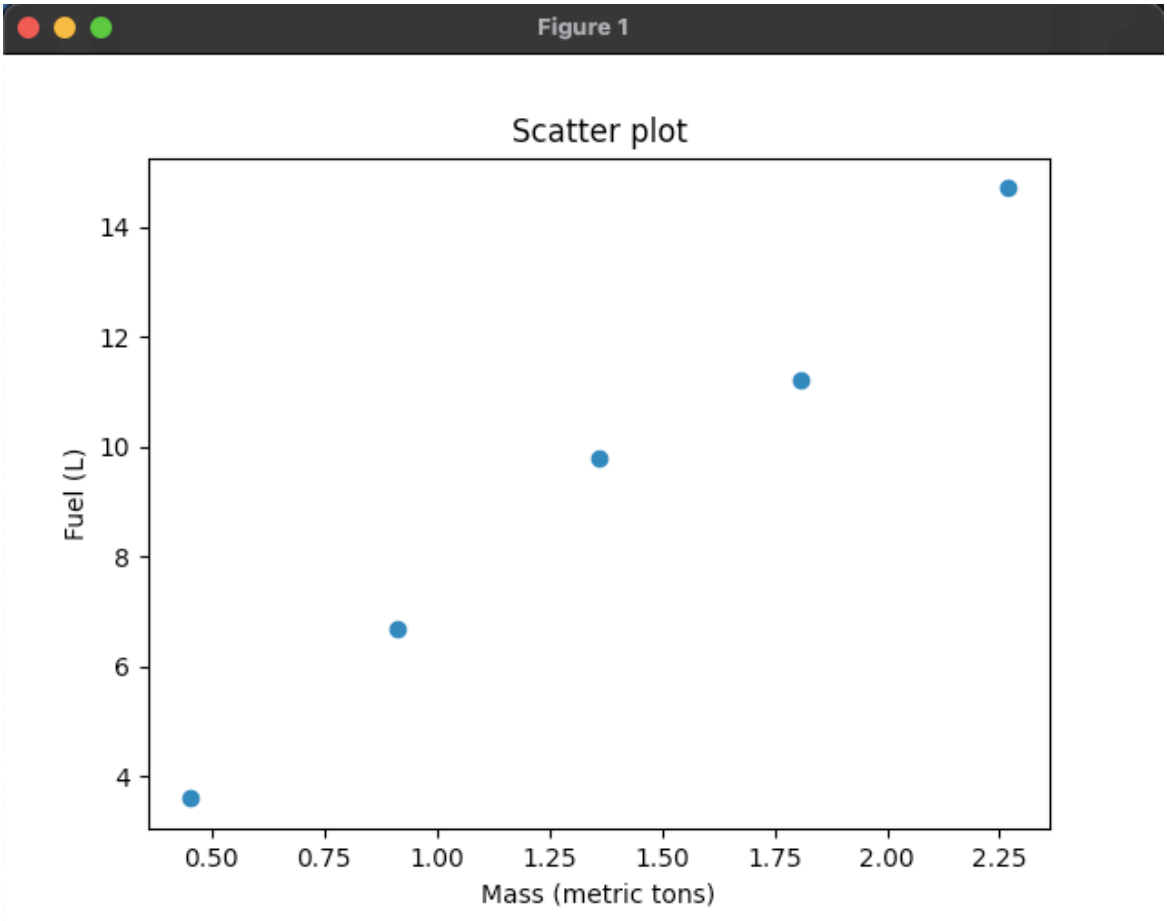
Code:

```
import matplotlib.pyplot as plt
from matplotlib.pyplot import figure

Fuel = [3.6, 6.7, 9.8, 11.2, 14.7]
Mass = [0.45, 0.91, 1.36, 1.81, 2.27]

plt.scatter(Mass, Fuel)
plt.xlabel("Mass (metric tons)")
plt.ylabel("Fuel (L)")
plt.title("Scatter plot")
plt.show()
```

Output:



There is a linear positive correlation between fuel and mass.

Q5 : The data below represents the number of chairs in each class of a government high school. Create a box plot and swarm plot (add jitter) and find the number of data points that are outliers. 35, 54, 60, 65, 66, 67, 69, 70, 72, 73, 75, 76, 54, 25, 15, 60, 65, 66, 67, 69, 70, 72, 130, 73, 75, 76

Logic : The data is plotted using inbuilt functions to get a box and scatter plot.

Libraries used:

- Matplotlib - For the boxplot
- Seaborn - For the scatterplot

Code:

```
import matplotlib.pyplot as plt
from matplotlib.pyplot import figure
import seaborn as sns
figure(num=None, figsize=(8, 6), dpi=80, facecolor='w', edgecolor='k')
data=[35, 54, 60, 65, 66, 67, 69, 70, 72, 73, 75, 76, 54, 25,
15, 60, 65, 66, 67, 69, 70, 72, 130, 73, 75, 76]

plt.boxplot(data)

sns.stripplot(data,jitter=1)
plt.show()
```

Output:

From the boxplot we can observe that there are 4 outliers in the data given.

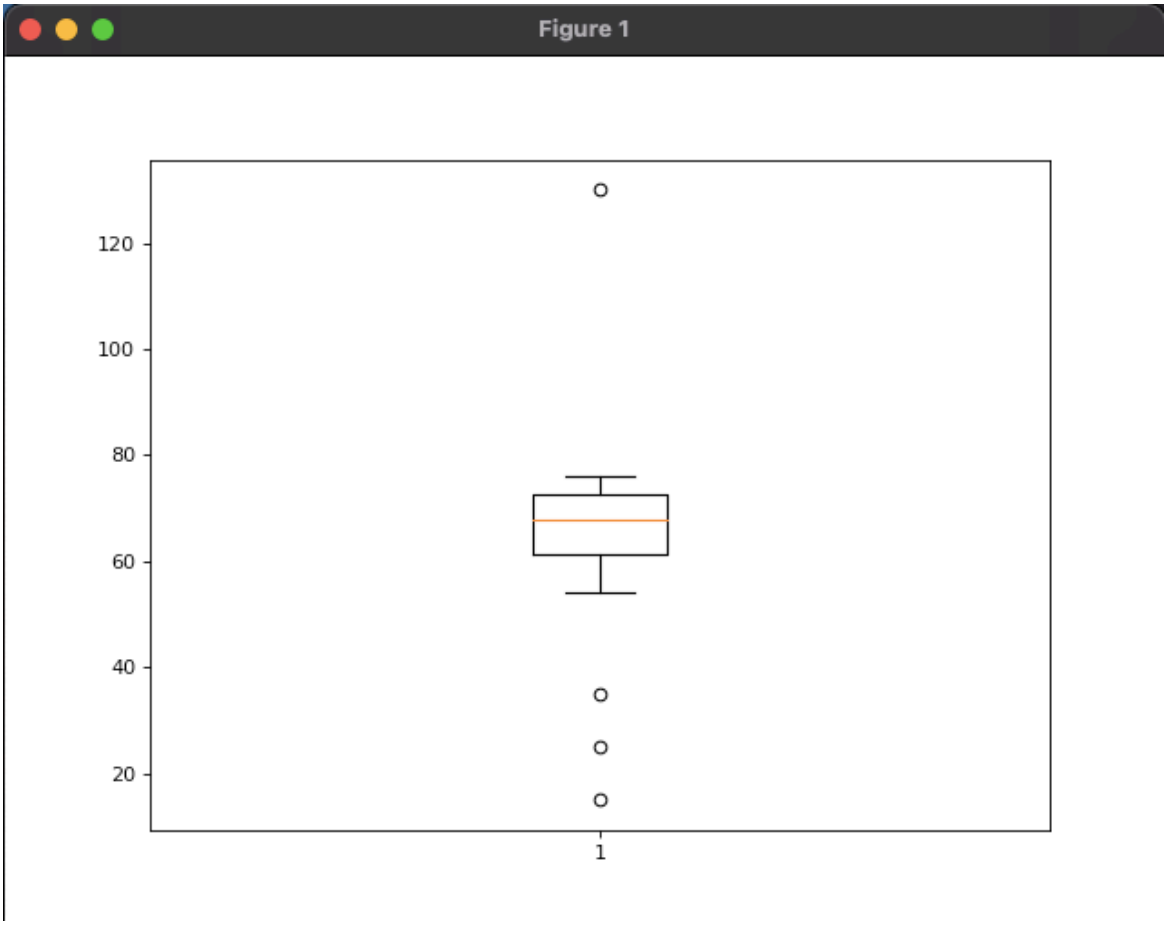


Figure 1
120
100
80
60
40
20
1

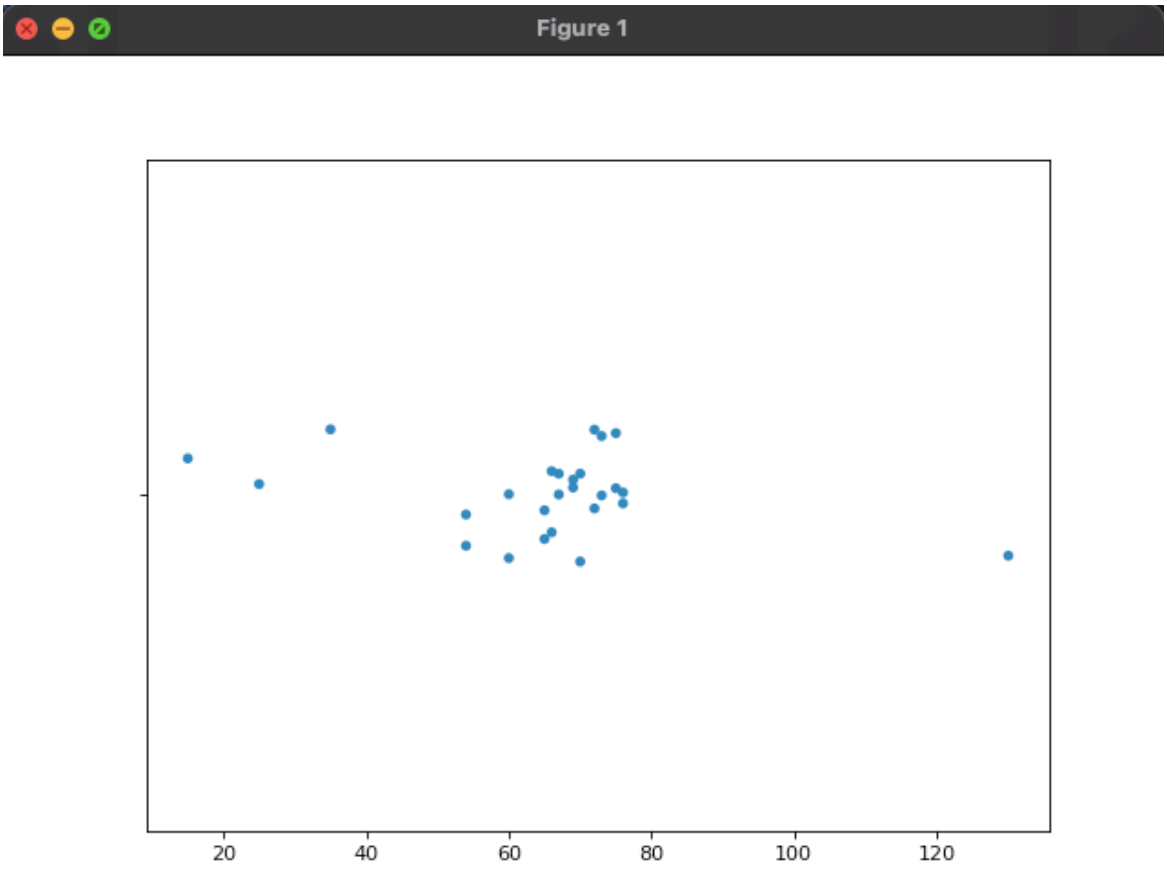


Figure 1
20
40
60
80
100
120

Q6 : Generate random numbers from the following distribution and visualize the data using violin plot.

- (i) Standard-Normal distribution.
- (ii) Log-Normal distribution.

Logic : Random numbers are generated from the given distributions and are plotted.

Libraries used:

- Matplotlib - For the plots
- Numpy - For generating and storing the random array

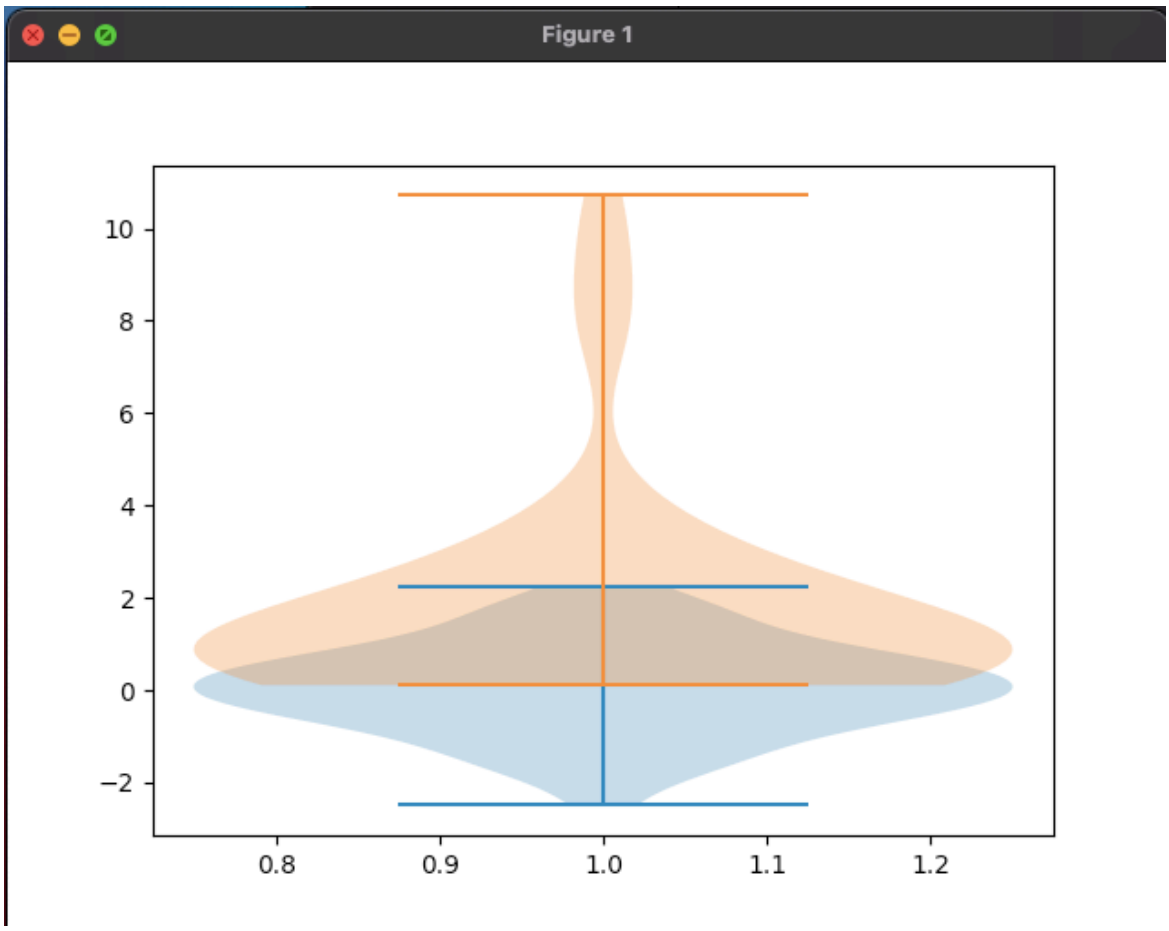
Code:

```
import matplotlib.pyplot as plt
import numpy as np

standard_normal = np.random.normal(0,1,50)
lognormal=np.random.lognormal(0,1,50)

fig, ax = plt.subplots()
ax.violinplot(standard_normal)
ax.violinplot(lognormal)
plt.show()
```

Output:



Q7 : An Advertisement agency develops new ads for various clients (like Jewellery shops, Textile shops). The Agency wants to assess their performance, for which they want to know the num- ber of ads they developed in each quarter for different shop category. Help them to visualize data using radar/spider charts.

Logic : The data is plotted with the help of inbuilt functions.

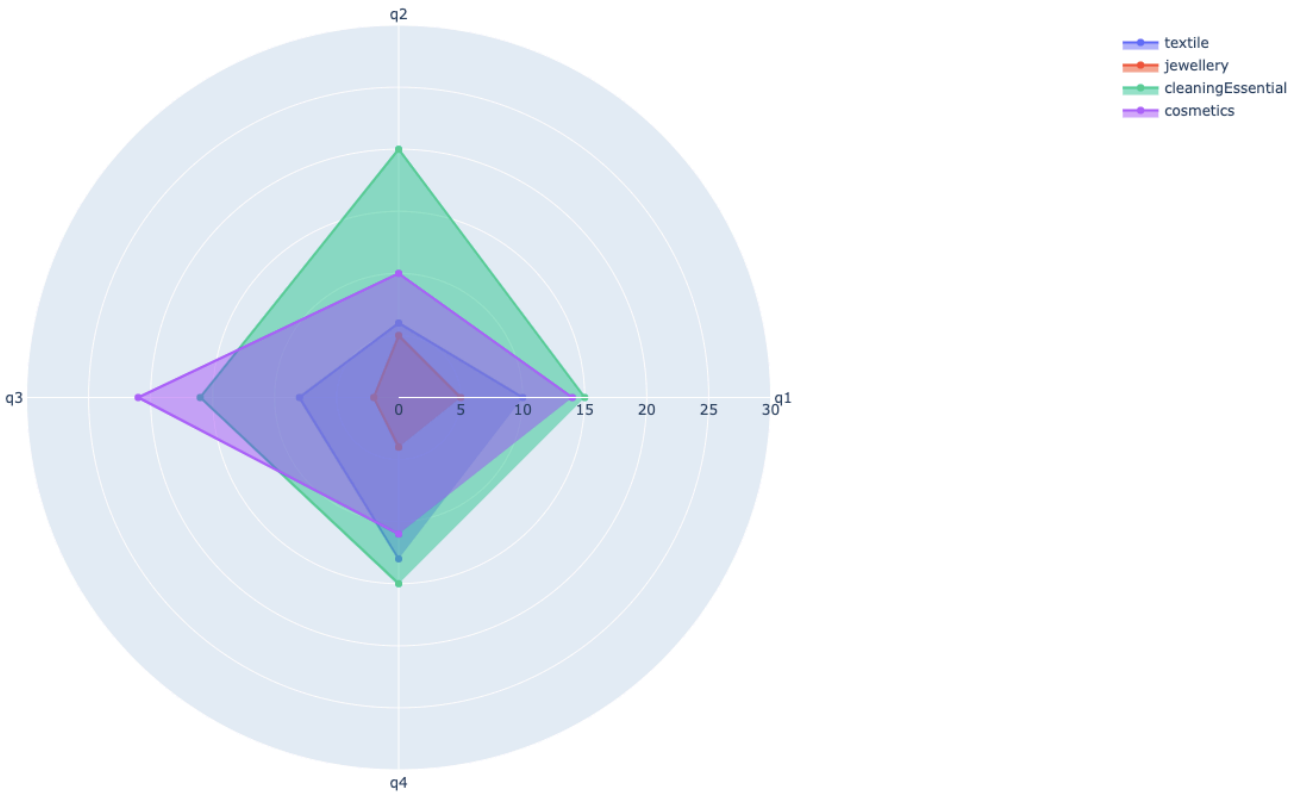
Libraries used:

- Plotly - For the radar chart

Code:

```
import plotly.graph_objects as go
quaters = ['q1','q2','q3','q4']
fig = go.Figure()
fig.add_trace(go.Scatterpolar(
                r=[10,6 ,8 ,13],
                theta=quaters,
                fill='toself',
                name='textile',
))
fig.add_trace(go.Scatterpolar(
                r=[5, 5, 2, 4],
                theta=quaters,
                fill='toself',
                name='jewellery'
))
fig.add_trace(go.Scatterpolar(
                r=[15, 20, 16, 15],
                theta=quaters,
                fill='toself',
                name='cleaningEssential',
))
fig.add_trace(go.Scatterpolar(
                r=[14, 10, 21, 11],
                theta=quaters,
                fill='toself',
                name='cosmetics'
))
fig.update_layout(polar=dict(radialaxis=dict(visible=True,range=[0, 30]),),showlegend=True)
fig.show()
```

Output:


q2
q3
q1
q4
0
5
10
15
20
25
30
textile
jewellery
cleaningEssential
cosmetics

Q8 : An organization wants to calculate the % of time they spent on each process for their product development. Visualize the data using funnel chart with the data given below.

Logic : The total time spent and percentage of time spent for each development steps is calculated.The percentage of time spent for each development steps is plotted against the product development steps using inbuilt functions.

Libraries used:

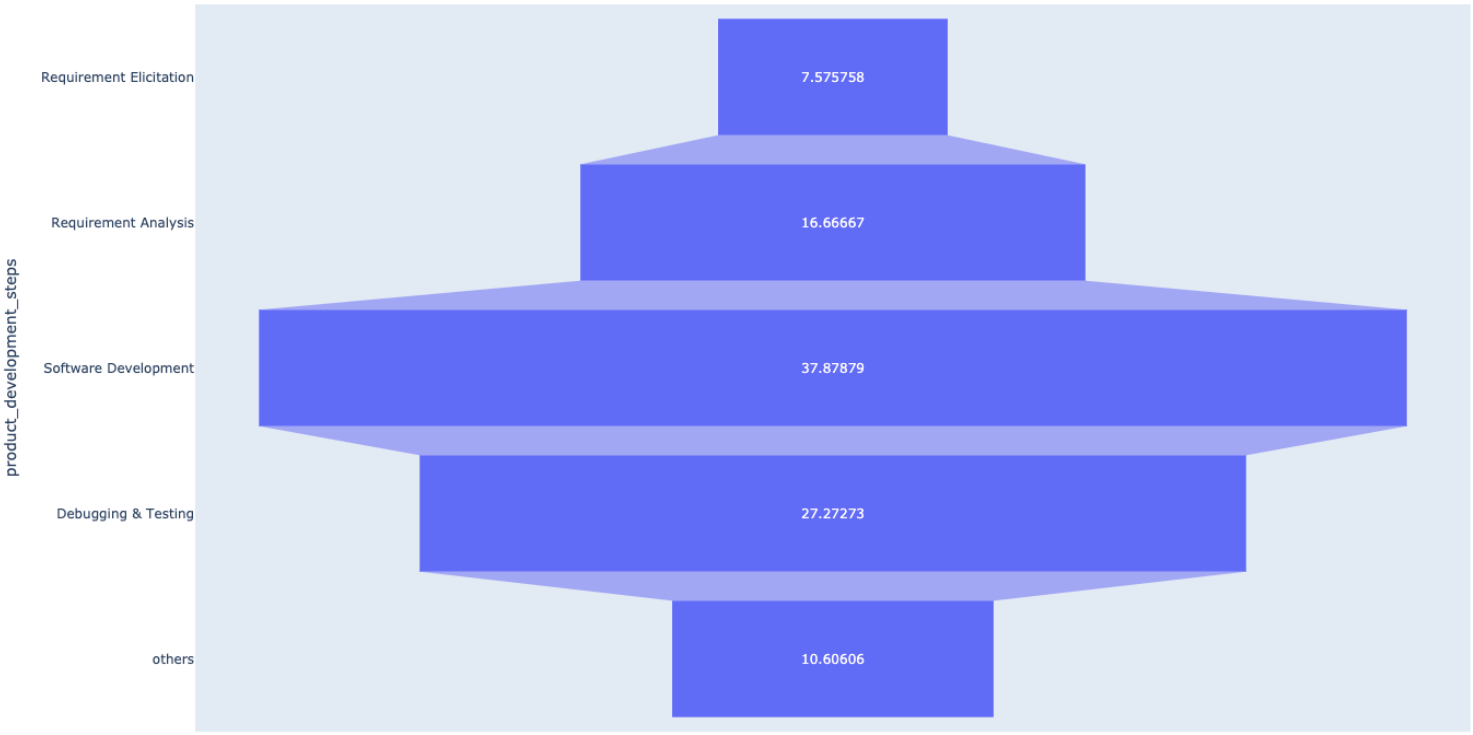
- Plotly - For the funnel chart.

Code:

```
import plotly.express as px
time_spent=[50,110,250,180,70]
total_time_spent=0
for i in range(5):
    total_time_spent= total_time_spent+time_spent[i]
percent_time_spent=[]
for i in range(5):
    percent_time_spent.append((time_spent[i]/total_time_spent)*100)
data=dict(percentage_time_spent=percent_time_spent,
product_development_steps=["Requirement Elicitation",
                           "Requirement Analysis",
                           "Software Development",
                           "Debugging & Testing",
                           "others"])

fig=px.funnel(data,x='percentage_time_spent',y='product_development_steps')
fig.show()
```

Output:


product_development_steps
Requirement Elicitation
Requirement Analysis
Software Development
Debugging & Testing
others
7.575758
16.66667
37.87879
27.27273
10.60606

Q9 : Let's say you are the new owner of a small ice-cream shop in a little village near the beach. You noticed that there was more business in the warmer months than the cooler months. Before you alter your purchasing pattern to match this trend, you want to be sure that the relationship is real. Help him to find the correlation between the data given.

Logic : The correlation co-efficient is found using inbuilt functions

Libraries used:

- Scipy - For the correlation calculation

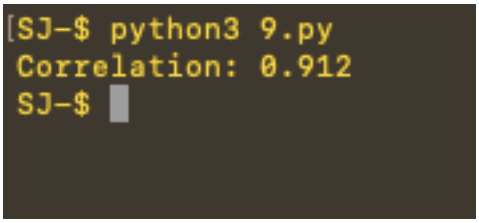
Code:

```
from scipy.stats import pearsonr

temperature = [98, 87, 90, 85, 95, 75]
num_customers = [15, 12, 10, 10, 16, 7]

corr, _ = pearsonr(temperature, num_customers)
print('Correlation: %.3f' % corr)
```

Output:



We can see that the correlation coefficient is very close to +1 indicating a very strong positive correlation.