ANALYTICS AND SYSTEMS OF BIG DATA PRACTICE PROBLEM SET 2

January 26, 2021

Bhaavanaa Thumu - CED17I021

QUESTION 1.

PROBLEM STATEMENT

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 USED

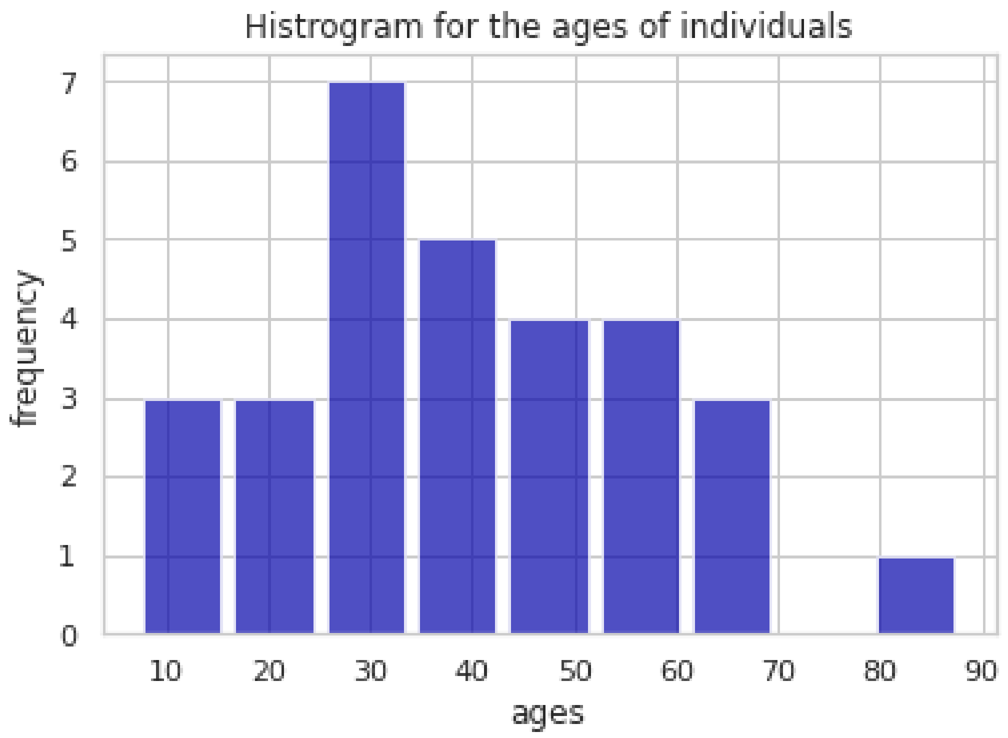
Plot the histogram for the given ages of customers using matplotlib library by varying the bin size and the type of histogram. The histogram is unimodal since it has only one peak.

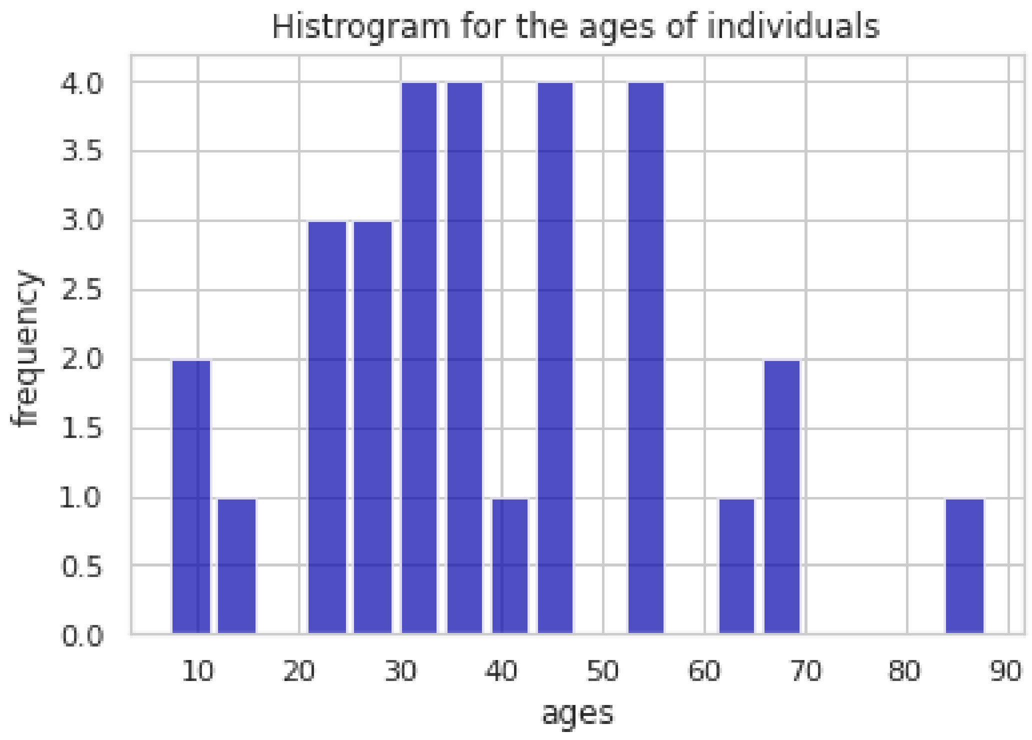
PACKAGES USED

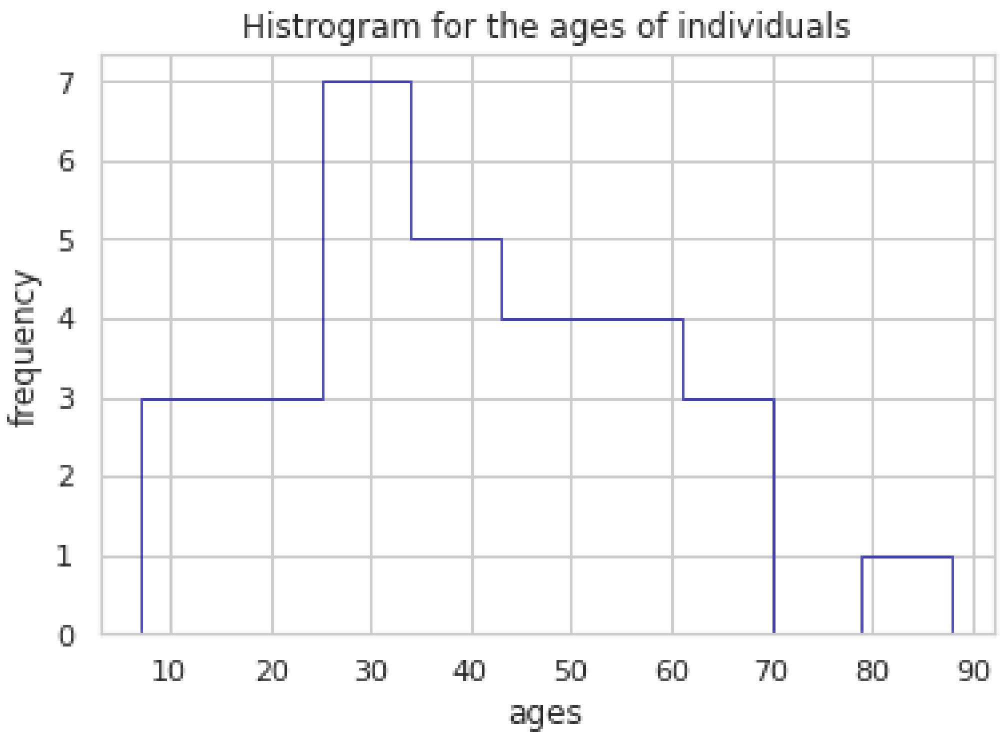
matplotlib - for plotting the histogram

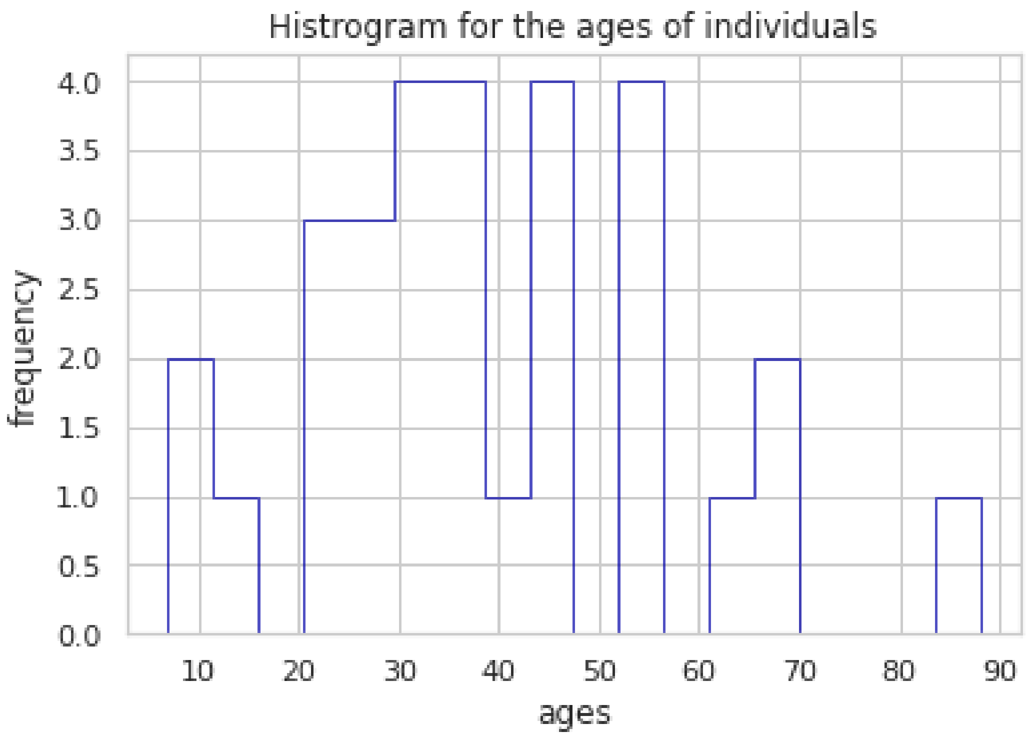
[82]:
```
from matplotlib import pyplot as plt                                # import the␣
 ↪required library

ages = [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]                     # store the ages␣
 ↪of the individuals in a list

# barstacked histogram with 9 bins i.e. binsize=10
plt.hist(ages, bins=9, color='#0504aa', alpha=0.7, rwidth=0.85,␣
 ↪histtype='barstacked')
plt.xlabel("ages")                                                  # set the x axis␣
 ↪label as ages
plt.ylabel("frequency")                                             # set the y axis␣
 ↪label as frequency
plt.title("Histrogram for the ages of individuals")                 # give a title␣
 ↪to the plot
```

```
plt.show()                                                    # display the␣
↪plot

# barstacked histogram with 18 bins i.e. binsize=5
plt.hist(ages, bins=18, color='#0504aa', alpha=0.7, rwidth=0.85,␣
↪histtype='barstacked')
plt.xlabel("ages")                                            # set the x axis␣
↪label as ages
plt.ylabel("frequency")                                       # set the y axis␣
↪label as frequency
plt.title("Histrogram for the ages of individuals")           # give a title␣
↪to the plot
plt.show()                                                    # display the␣
↪plot

# step histogram with 9 bins i.e. binsize=10
plt.hist(ages, bins=9, color='#0504aa', alpha=0.7, rwidth=0.85, histtype='step')
plt.xlabel("ages")                                            # set the x axis␣
↪label as ages
plt.ylabel("frequency")                                       # set the y axis␣
↪label as frequency
plt.title("Histrogram for the ages of individuals")           # give a title␣
↪to the plot
plt.show()                                                    # display the␣
↪plot

# step histogram with 18 bins i.e. binsize=10
plt.hist(ages, bins=18, color='#0504aa', alpha=0.7, rwidth=0.85, histtype='step')
plt.xlabel("ages")                                            # set the x axis␣
↪label as ages
plt.ylabel("frequency")                                       # set the y axis␣
↪label as frequency
plt.title("Histrogram for the ages of individuals")           # give a title␣
↪to the plot
plt.show()                                                    # display the␣
↪plot
```


Histrogram for the ages of individuals
frequency
ages
0
1
2
3
4
5
6
7
10
20
30
40
50
60
70
80
90


Histrogram for the ages of individuals
frequency
ages
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
10
20
30
40
50
60
70
80
90


Histrogram for the ages of individuals
frequency
ages
0
1
2
3
4
5
6
7
10
20
30
40
50
60
70
80
90


Histrogram for the ages of individuals
frequency
ages
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
10
20
30
40
50
60
70
80
90

QUESTION 2.

PROBLEM STATEMENT

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 USED

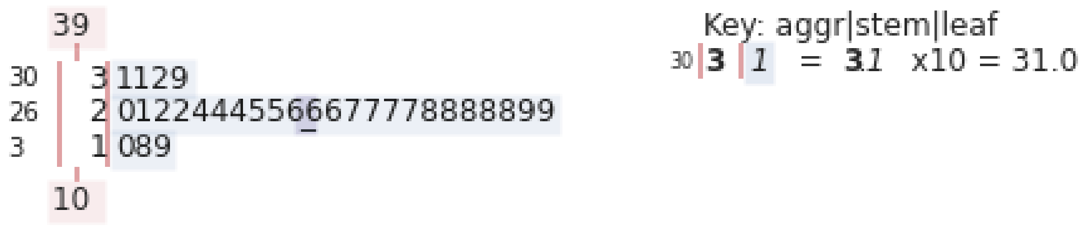
Use the stemgraphic library to plot the stem-leaf plot for the points scored. The outliers can be detected using the plot obtained. They are 10 and 39, in this case. The distribution looks like a normal distribution since the no. of leaves is greater in the middle and less towards both the extremes.

PACKAGES USED

stemgraphic - for plotting the stem-leaf plot and obtaining the outliers

[83]:
```
import stemgraphic                                  # import the required library

points = [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]     # store the points scored in␣
 ↪a list

stemgraphic.stem_graphic(points, scale = 10)        # call stem_graphic with␣
 ↪required parameters, points and scale
```

[83]:
```
(<Figure size 540x108 with 1 Axes>,
 <matplotlib.axes._axes.Axes at 0x7fdd746486d8>)
```



QUESTION 3.

PROBLEM STATEMENT

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. WATER 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 BEVERAGES 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

LOGIC USED

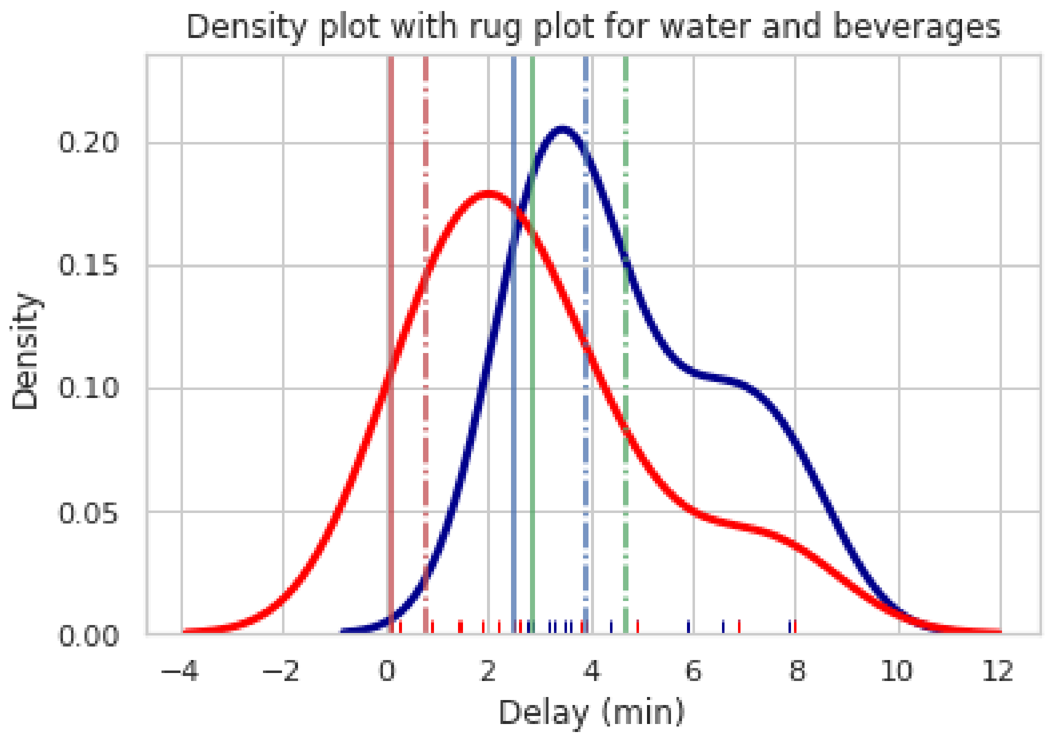
Use the matplotlib and the seaborn library to plot the density and rug plot for water and other beverages. Compute the median, mean using statistics library and skewness using the formula and plot them in the graph. water - blue , beverages - red median - blue, mean - green, skewness - red water - '-.' , beverages - '-', for median, mean and skewness representation

PACKAGES USED

matplotlib - to plot the graphs seaborn - to identify the mean, median and skewness and to plot the density and rug plot statistics - to compute the median and mean

[84]:
```
import matplotlib.pyplot as plt                                           # import␣
 ↪the required libraries
import seaborn as sns
import statistics

water = [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]      # save the data in lists
beverages = [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]

meanw = statistics.mean(water)                                  # compute the mean␣
 ↪using statistics lib
medianw = statistics.median(water)                              # compute the median␣
 ↪using statistics lib
modew = water[0]                     # compute the mode using statistics lib
std_devw = statistics.stdev(water)                              # compute the std dev.
 ↪ using statistics lib
skewnessw = (meanw - modew)/std_devw                            # compute the␣
 ↪skewness=(Mean-Mode)/standard deviation

meanb = statistics.mean(beverages)                              # compute the mean␣
 ↪using statistics lib
medianb = statistics.median(beverages)                          # compute the median␣
 ↪using statistics lib
modeb = statistics.mode(beverages)                              # compute the mode␣
 ↪using statistics lib
std_devb = statistics.stdev(beverages)                          # compute the std dev.
 ↪ using statistics lib
skewnessb = (meanb - modeb)/std_devb                            # compute the␣
 ↪skewness=(Mean-Mode)/standard deviation
```

```
sns.distplot(water, hist = False, kde = True, rug = True,           # density␣
 ↪plot with rug plot for water
             color = 'darkblue',
             kde_kws={'linewidth': 3},
             rug_kws={'color': 'darkblue'})

sns.distplot(beverages, hist = False, kde = True, rug = True,       # density␣
 ↪plot with rug plot for beverages
             color = 'red',
             kde_kws={'linewidth': 3},
             rug_kws={'color': 'red'})

plt.axvline(medianw, color='b', linestyle='-.')
plt.axvline(meanw, color='g', linestyle='-.')
plt.axvline(skewnessw, color='r', linestyle='-.')

plt.axvline(medianb, color='b', linestyle='-')
plt.axvline(meanb, color='g', linestyle='-')
plt.axvline(skewnessb, color='r', linestyle='-')

plt.title('Density plot with rug plot for water and beverages')      # give a␣
 ↪title to the plot
plt.xlabel('Delay (min)')                                            # give a␣
 ↪label to the x axis
plt.ylabel('Density')                                                # give a␣
 ↪label to the y axis
plt.show()
```



QUESTION 4.

PROBLEM STATEMENT

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 correlation). The data is summarized in the table below. (Use a reasonable scale on both axes and put the explanatory variable on the x-axis.) Fuel used (L) 3.6 6.7 9.8 11.2 14.7 Mass (metric tons) 0.45 0.91 1.36 1.81 2.27

LOGIC USED

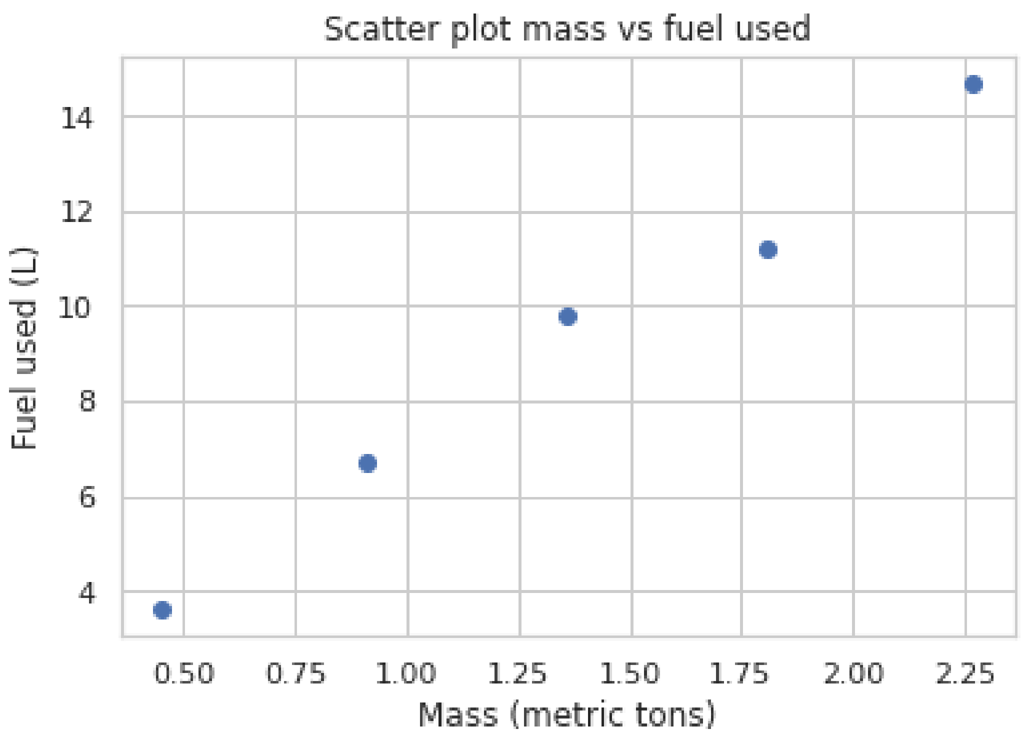
Plot the scatter plot using the matplotlib library, with mass on the x axis as it is the independent variable, and fuel on the y axis as it is the dependent variable. From the graph, we can see that it is a postive and linear correlation.

PACKAGES USED

matplotlib - to plot the scatter plot

[85]:
```
import matplotlib.pyplot as plt                   # import the required libraries

fuel = [3.6, 6.7, 9.8, 11.2, 14.7]                # store the given data as lists
mass = [0.45, 0.91, 1.36, 1.81, 2.27]
```

```
plt.scatter(mass, fuel)                              # plot the scatter plot

plt.title('Scatter plot mass vs fuel used')          # give a title to the plot
plt.xlabel('Mass (metric tons)')                     # give a label to the x axis
plt.ylabel('Fuel used (L)')                          # give a label to the y axis
plt.show()                                           # display the plot
```



QUESTION 5.

PROBLEM STATEMENT

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.

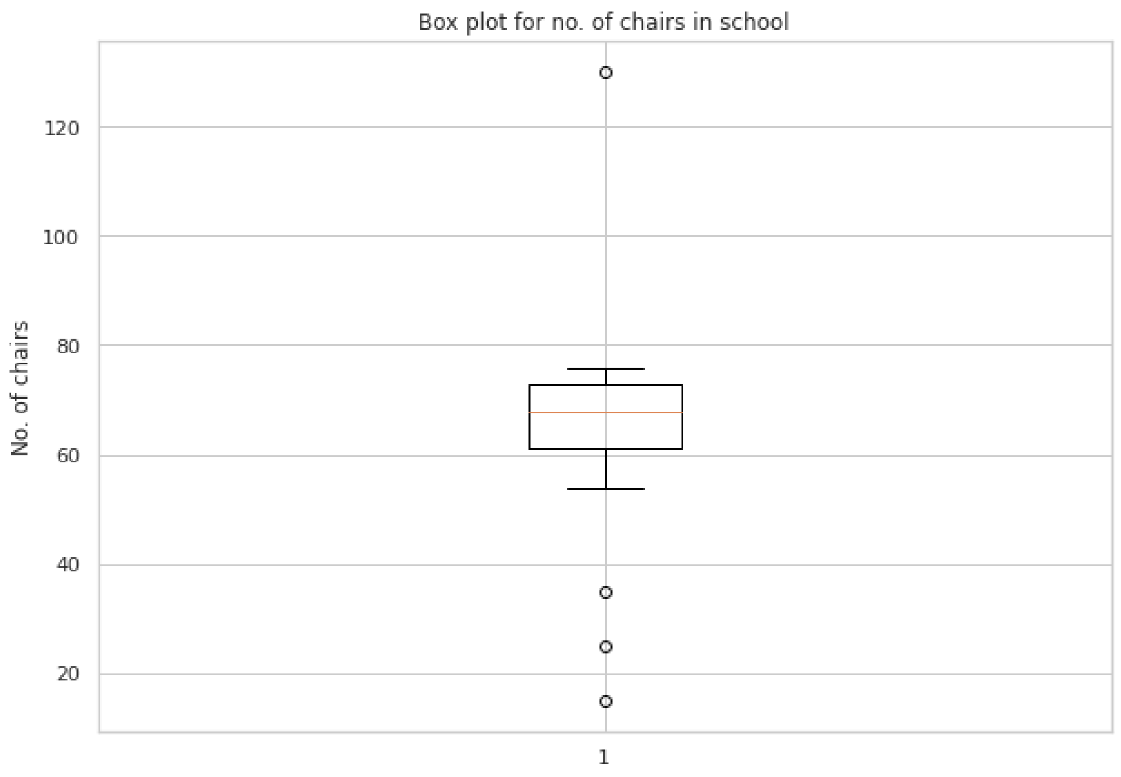
LOGIC USED

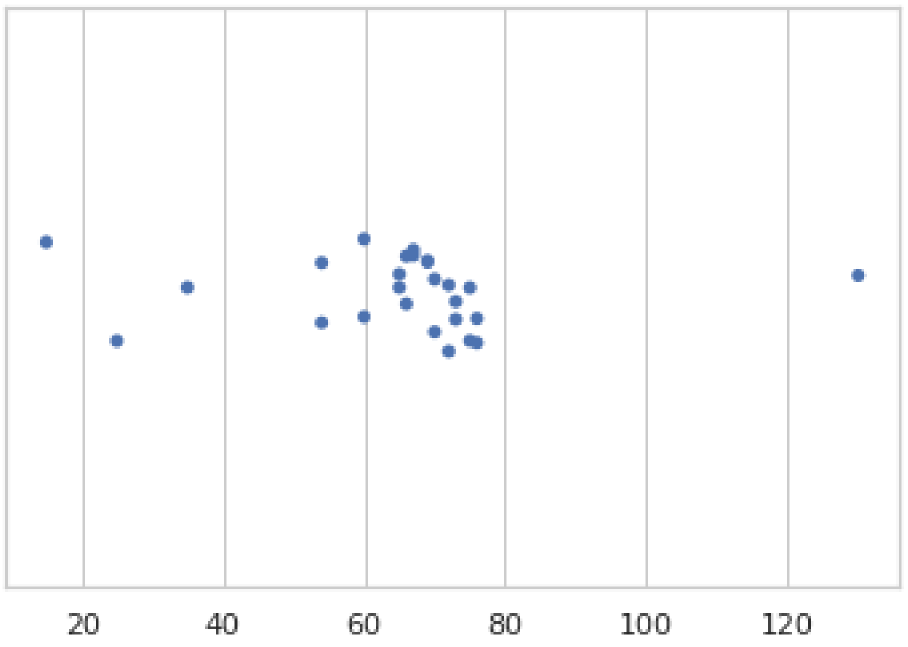
Plot the box plot using the mtplotlib library, and the swarmplot using the seaborn library, and initialize jitter to True/1. From the graph obtained, we can clearly see that the number of outliers is 4, in this case.

PACKAGES USED

matplotlib - for plotting the boxplot seaborn - for plotting the swarmplot

```
[86]: import matplotlib.pyplot as plt                          # import the required␣
       ↪libraries
      import seaborn as sns

      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]
                                                               # represent the given␣
       ↪data in a list
      fig = plt.figure(figsize =(10, 7))                        # mention the size of␣
       ↪the plot
      plt.boxplot(data)                                         # create the box plot

      plt.title('Box plot for no. of chairs in school')         # give a title to the␣
       ↪plot
      plt.ylabel('No. of chairs')                               # give a label to the y␣
       ↪axis
      plt.show()                                                # display the plot

      sns.set_theme(style="whitegrid")
      ax = sns.stripplot(x=data, jitter=1)
```





QUESTION 6.

PROBLEM STATEMENT

Generate random numbers from the following distribution and visualize the data using violin plot. (i) Standard-Normal distribution. (ii) Log-Normal distribution.

LOGIC USED

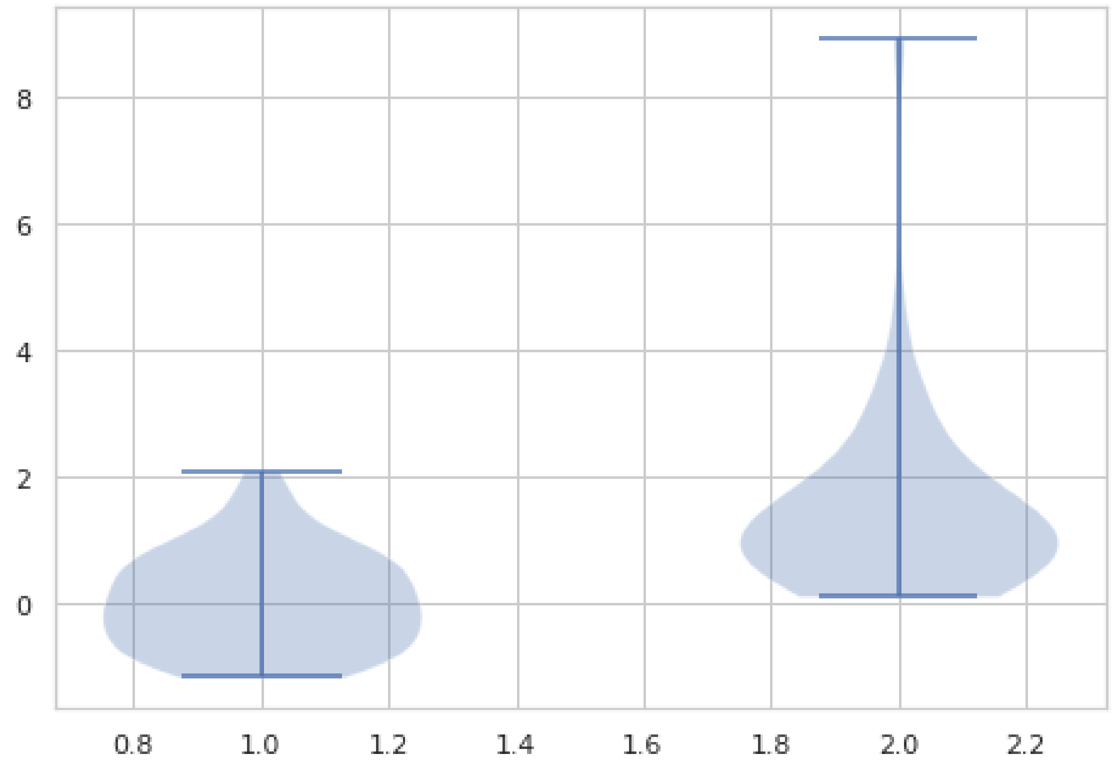
Genearte the random numbers from the distributions mentioned using the numpy library. Plot the violin plot for the random numbers using the matplotlib library

PACKAGES USED

numpy - for generating the random numbers from the required distribution matplotlib - for plotting the violinplot

[87]:
```
import numpy as np                                          # import the required␣
 ↪libraries
from matplotlib import pyplot as plt

random_stdnormal = np.random.normal(0.0, 1.0, 50)          # genearte rand nos.␣
 ↪from std. normal distribution
random_lognormal = np.random.lognormal(0.0, 1.0, 50)       # generate rand nos.␣
 ↪from log normal distribution

data_to_plot = [random_stdnormal, random_lognormal]        # combine the data in␣
 ↪one list
```

```
fig = plt.figure()                                      # create a figure␣
 ↪instance
ax = fig.add_axes([0,0,1,1])                            # create an axes instance
bp = ax.violinplot(data_to_plot)                        # plot the violinplot
plt.show()                                              # display the violin plot
```



QUESTION 7.

PROBLEM STATEMENT

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 number of ads they developed in each quarter for different shop category. Help them to visualize data using radar/spider charts.

Shop Category Quarter 1 Quarter 2 Quarter 3 Quarter 4 Textile 10 6 8 13 Jewellery 5 5 2 4 Cleaning Essentials 15 20 16 15 Cosmetics 14 10 21 11

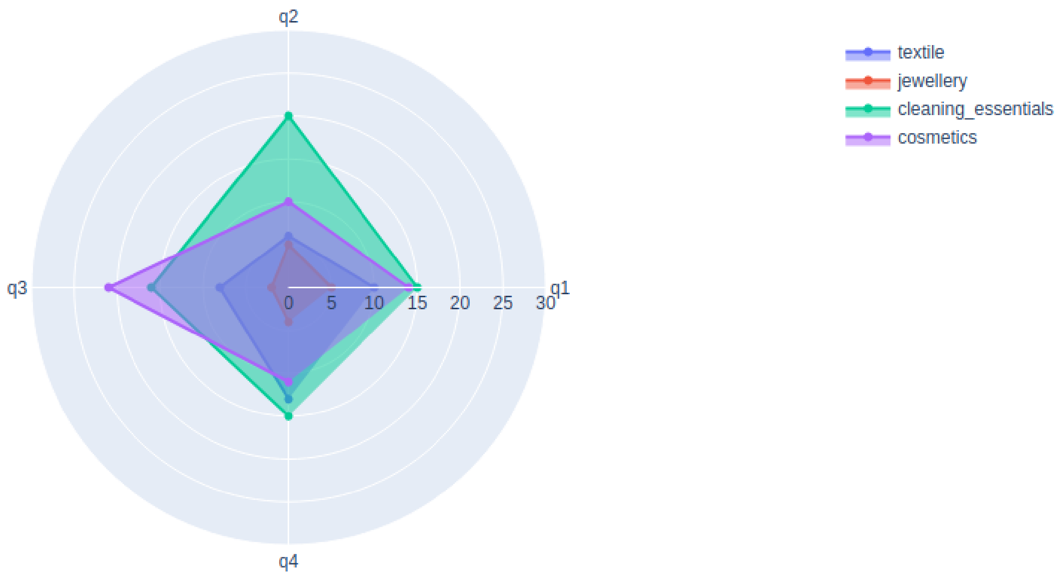
LOGIC USED

Plot the radar chart using the plotly library, along with the required values and labels.

PACKAGES USED

plotly - for plotting the radar chart

```
[88]: import plotly.graph_objects as go              # import th required library

quarters = ['q1','q2','q3','q4']                   # store the quarters in a list

fig = go.Figure()
fig.add_trace(go.Scatterpolar(r=[10,6,8,13], theta=quarters, fill='toself',␣
 ↪name='textile'))
fig.add_trace(go.Scatterpolar(r=[5,5,2,4], theta=quarters, fill='toself',␣
 ↪name='jewellery'))
fig.add_trace(go.Scatterpolar(r=[15,20,16,15], theta=quarters, fill='toself',␣
 ↪name='cleaning_essentials'))
fig.add_trace(go.Scatterpolar(r=[14,10,21,11], theta=quarters, fill='toself',␣
 ↪name='cosmetics'))

fig.update_layout(polar=dict(radialaxis=dict(visible=True, range=[0, 30])),␣
 ↪showlegend=True)

fig.show()                                         # display the plot
```



QUESTION 8.

PROBLEM STATEMENT

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.

Product Development steps Time spent (in hours) Requirement Elicitation 50 Requirement Analysis 110 Software Development 250 Debugging & Testing 180 Others 70

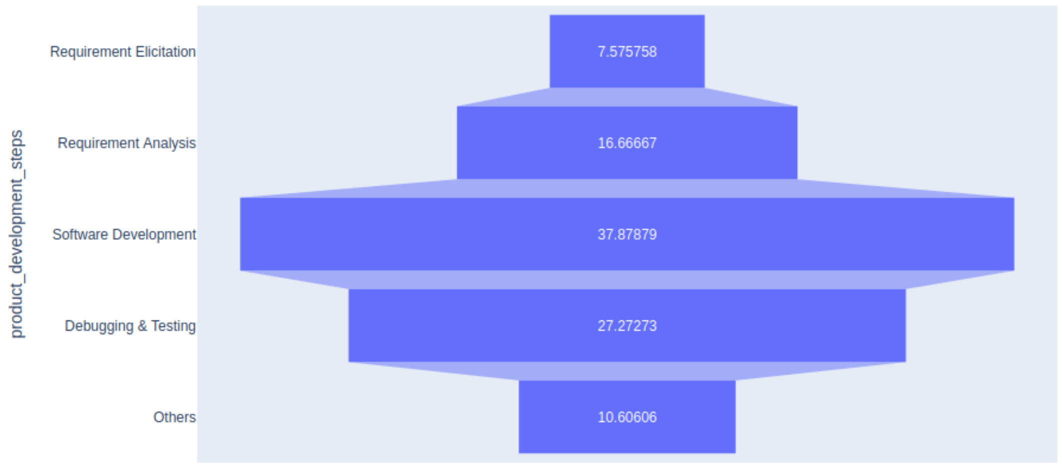
LOGIC USED

Compute the percentage of time spent on each step. Using the percentage of time and the step name, plot the funnel chart using plotly library.

PACKAGES USED

plotly - for plotting the funnel chart

[89]:
```
import plotly.express as px                             # import the required␣
 ↪library

time_spent = [50, 110, 250, 180, 70]                    # store the time spent in␣
 ↪a list
time_spent_percent = []
for i in time_spent:                                    # calculate percentage␣
 ↪time spent for each step
    time_spent_percent.append(i*100/sum(time_spent))

data = dict(                                            # store the required info␣
 ↪in a dictionary
    time_spent_percentage=list(time_spent_percent),
    product_development_steps=["Requirement Elicitation", "Requirement␣
 ↪Analysis", "Software Development",
           "Debugging & Testing", "Others"])

fig = px.funnel(data, x='time_spent_percentage', y='product_development_steps') ␣
 ↪   # plot the funnel chart
fig.show()                                              # display the funnel chart
```



QUESTION 9.

PROBLEM STATEMENT

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.

Temperature Number of Customers 98 15 87 12 90 10 85 10 95 16 75 7

LOGIC USED

Calculate the correlation value using Pearsons correlation. Since the correlation value obtained is high i.e. 0.912, it is positive high correlation.

PACKAGES USED

scipy - for obtaining the pearson correlation

[91]:
```python
from scipy.stats import pearsonr                # import the required library

temp = [98, 87, 90, 85, 95, 75]                 # store the data in a list
no_of_customers = [15, 12, 10, 10, 16, 7]

corr, _ = pearsonr(temp, no_of_customers)       # calculate Pearson's correlation

print('Pearsons correlation: %.3f' % corr)      # print the correlation value
```

```
Pearsons correlation: 0.912
```