Problem Set 2

CED18I039 - Paleti Krishnasai

1.

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.

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import random
```

In [2]:

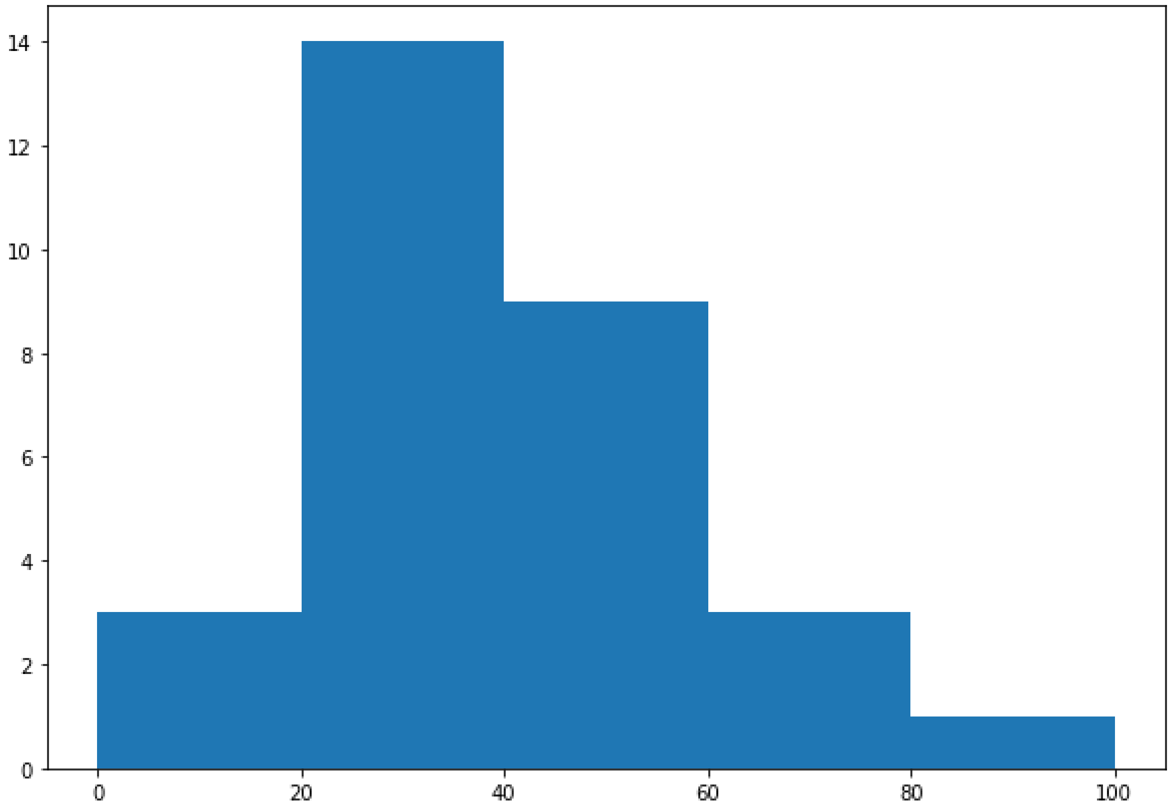
```
Age = [7, 9, 27, 28, 55, 45, 34, 65, 54, 67, 34, 23, 24, 66, 53, 45, 44, 88, 22, 33, 55, 35, 33, 37, 47, 41,31, 30, 29,
```

In [3]:

```
fig, ax = plt.subplots(figsize =(10, 7))
ax.hist(Age, bins = [0, 20, 40, 60, 80, 100])
```

Out[3]:

```
(array([ 3., 14.,  9.,  3.,  1.]),
 array([  0,  20,  40,  60,  80, 100]),
 <BarContainer object of 5 artists>)
```

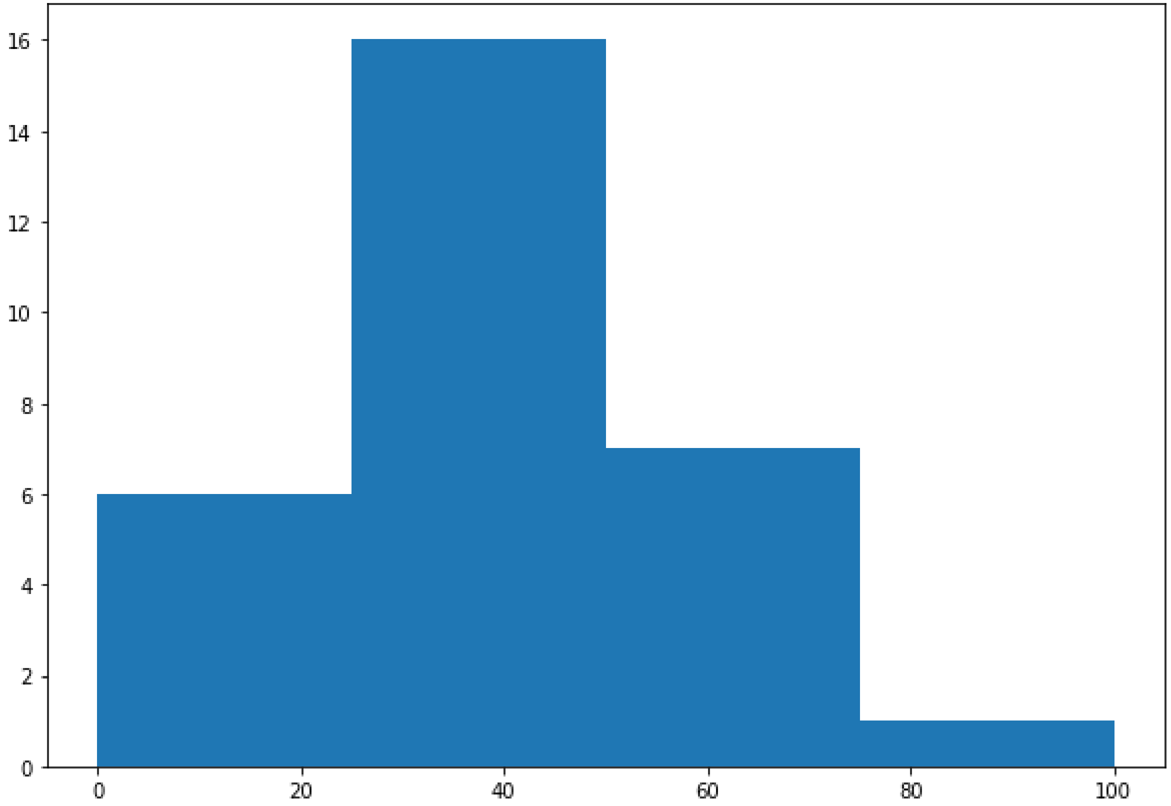


In [4]:

```
fig, ax = plt.subplots(figsize =(10, 7))
ax.hist(Age, bins = [0, 25, 50, 75, 100])
```

Out[4]:

```
(array([ 6., 16.,  7.,  1.]),
 array([  0,  25,  50,  75, 100]),
 <BarContainer object of 4 artists>)
```

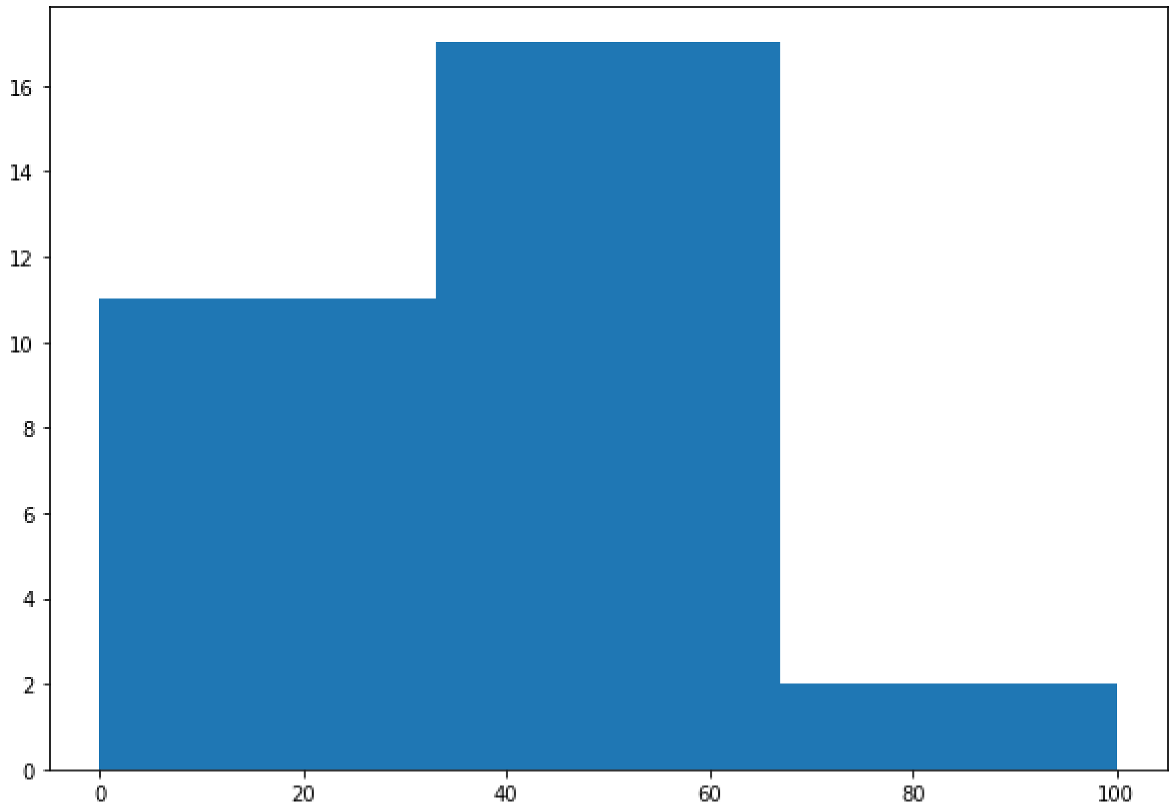


In [5]:

```
fig, ax = plt.subplots(figsize =(10, 7))
ax.hist(Age, bins = [0, 33, 67, 100])
```

Out[5]:

```
(array([11., 17.,  2.]),
 array([  0,  33,  67, 100]),
 <BarContainer object of 3 artists>)
```

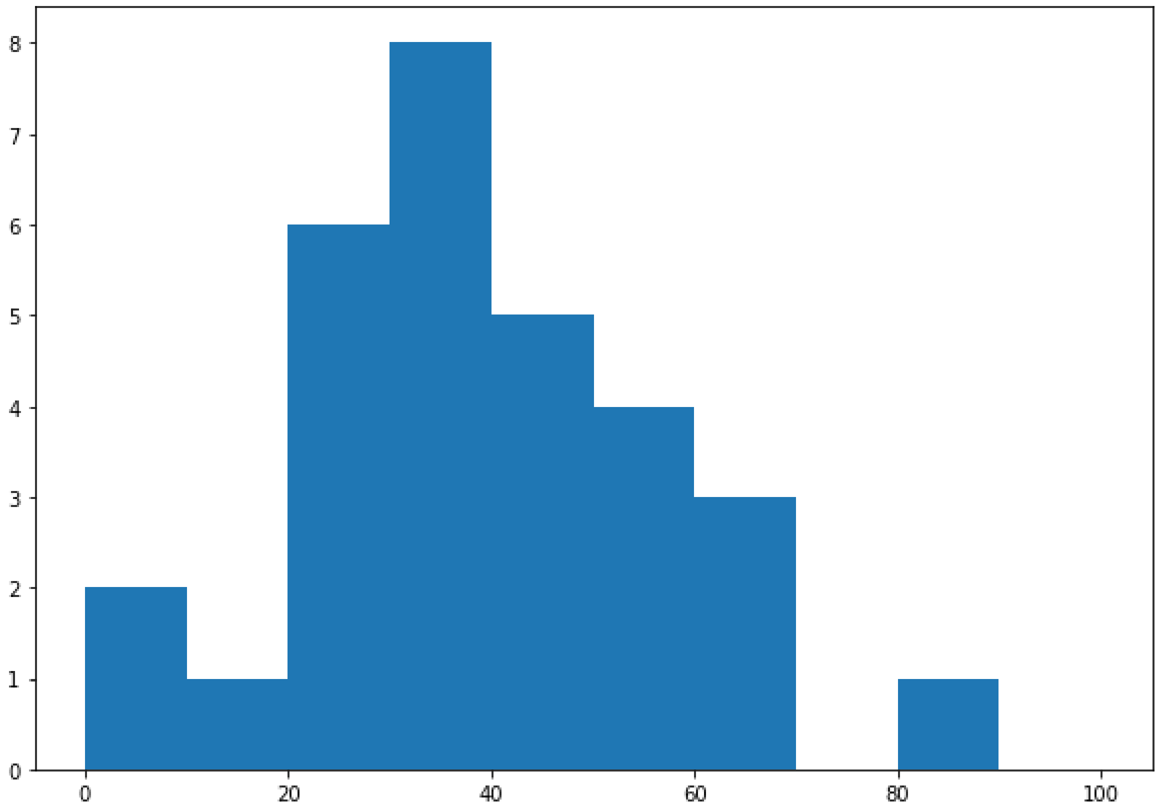


In [6]:

```
fig, ax = plt.subplots(figsize =(10, 7))
ax.hist(Age, bins = [0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100])
```

Out[6]:

```
(array([2., 1., 6., 8., 5., 4., 3., 0., 1., 0.]),
 array([  0,  10,  20,  30,  40,  50,  60,  70,  80,  90, 100]),
 <BarContainer object of 10 artists>)
```



- UNIMODAL histograms with different

2.

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.

In [7]:

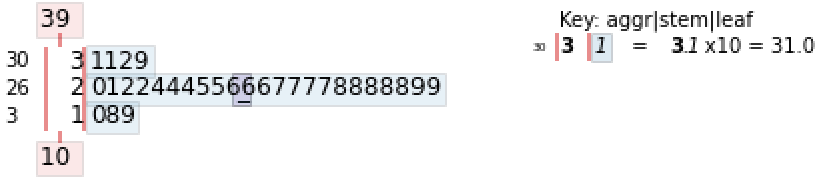
```python
import stemgraphic
```

In [8]:

```python
data = [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,
stemgraphic.stem_graphic(data,scale=10)
```

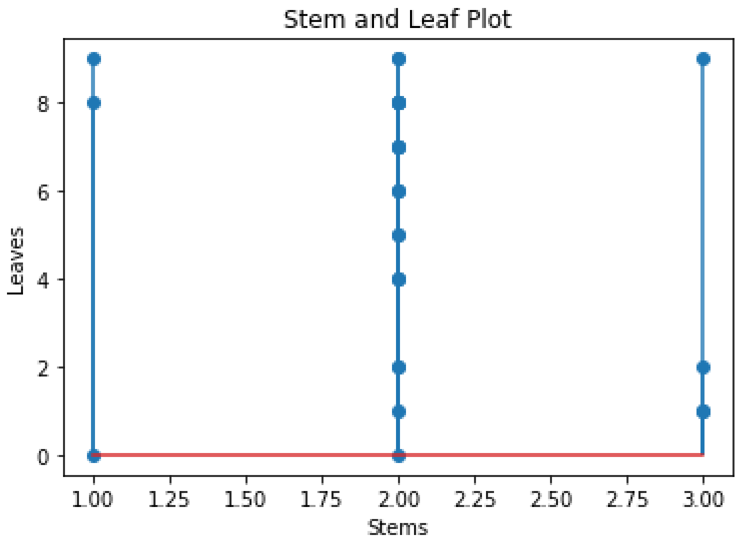
Out[8]:

```
(<Figure size 540x108 with 1 Axes>,
 <matplotlib.axes._axes.Axes at 0x7fa7c4947ac0>)
```



In [9]:

```python
def Generate_Stems_Leaves(data, leaflen):
    leaves = []
    stems = []
    for d in data:
        leaves.append(int(str(d)[(-1*leaflen):]))
        stems.append(int(str(d)[:(-1*leaflen)]))
    return stems, leaves

def GenerateStemPlot(stems, leaves):
    plt.title('Stem and Leaf Plot')
    plt.xlabel('Stems')
    plt.ylabel('Leaves')
    markerline, stemlines, baseline = plt.stem(stems, leaves)
    plt.show()

# Driver Code
data = [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,
stems, leaves = Generate_Stems_Leaves(data, 1)
GenerateStemPlot(stems, leaves)
```



- 10 and 39 are the potential outliers as they seem to be far off from the other data points

---

3.

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

In [10]:

```
import seaborn as sns
```

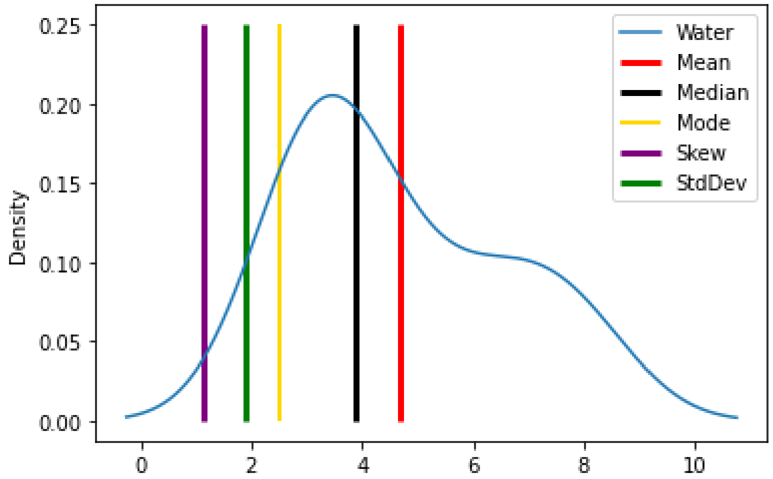
In [11]:

```
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]
```

In [12]:

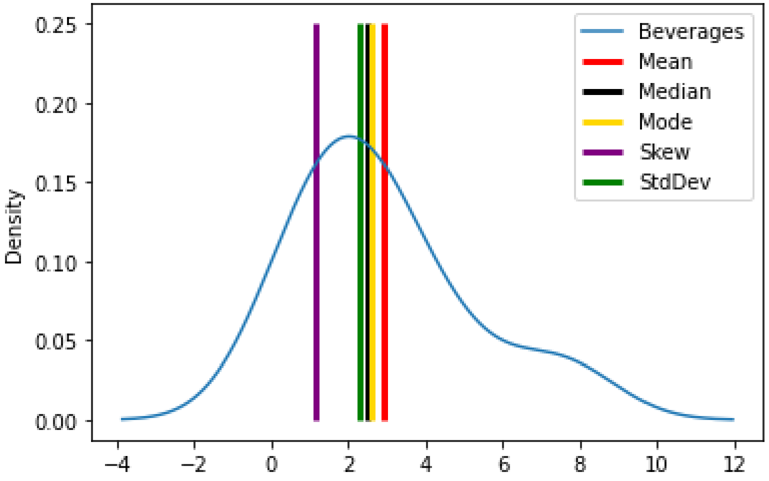
```
df=pd.DataFrame(Water,columns = ['Water'])
df.plot(kind='density')
Mean=np.mean(Water)
```

```
Median=np.median(Water)
Mode = df['Water'].mode()[0]
Stddev=df['Water'].std()
Skew=(Mean-Mode) / Stddev
plt.vlines(Mean, 0, 0.25, color = 'red', linestyle='solid', label = 'Mean', linewidth = 3)
plt.vlines(Median, 0, 0.25, color = 'black', linestyle='solid', label = 'Median', linewidth = 3)
plt.vlines(Mode, 0, 0.25, color = 'gold', linestyle='solid', label = 'Mode', linewidth = 2)
plt.vlines(Skew, 0, 0.25, color = 'purple', linestyle='solid', label = 'Skew', linewidth = 3)
plt.vlines(Stddev, 0, 0.25, color = 'green', linestyle='solid', label = 'StdDev', linewidth = 3)

plt.legend()
plt.show()
```



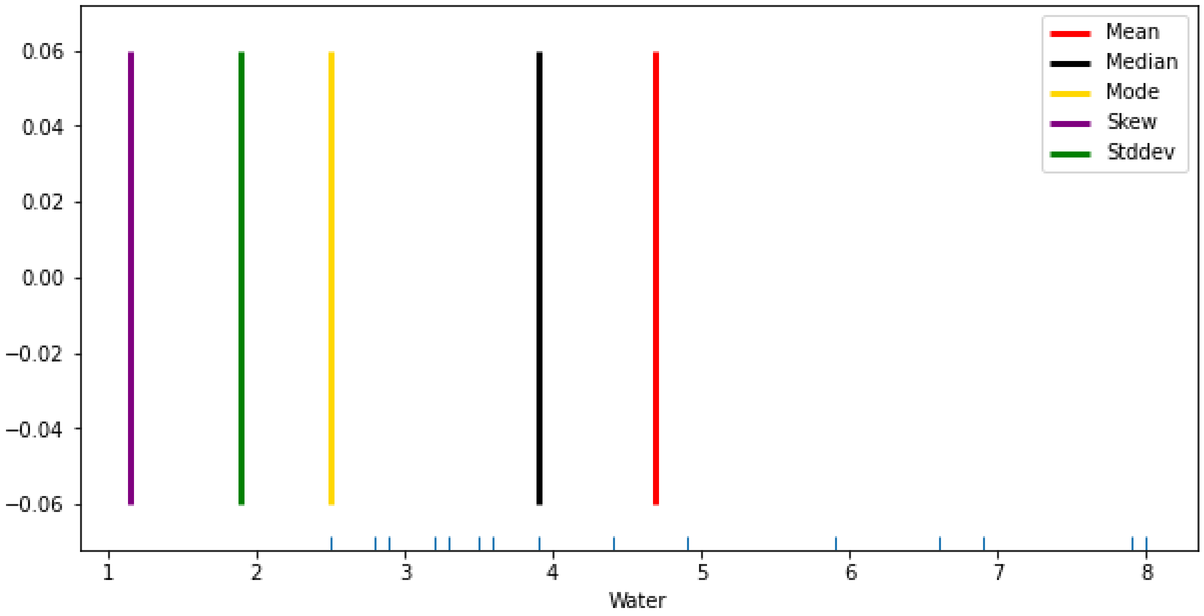
In [13]:

```
df1 = pd.DataFrame(Beverages, columns =  ['Beverages'])
df1.plot(kind = 'density')
Mean1 = np.mean(Beverages)
Median1 = np.median(Beverages)
Mode1 = df1['Beverages'].mode()[0]
Stddev1 = df1['Beverages'].std()
Skew1 = (Mean - Mode)/Stddev
plt.vlines(Mean1, 0, 0.25, color = 'red', linestyle='solid', label = 'Mean', linewidth = 3)
plt.vlines(Median1, 0, 0.25, color = 'black', linestyle='solid', label = 'Median', linewidth = 3)
plt.vlines(Mode1, 0, 0.25, color = 'gold', linestyle='solid', label = 'Mode', linewidth = 3)
plt.vlines(Skew1, 0, 0.25, color = 'purple', linestyle='solid', label = 'Skew', linewidth = 3)
plt.vlines(Stddev1, 0, 0.25, color = 'green', linestyle='solid', label = 'StdDev', linewidth = 3)
plt.legend()
plt.show()
```



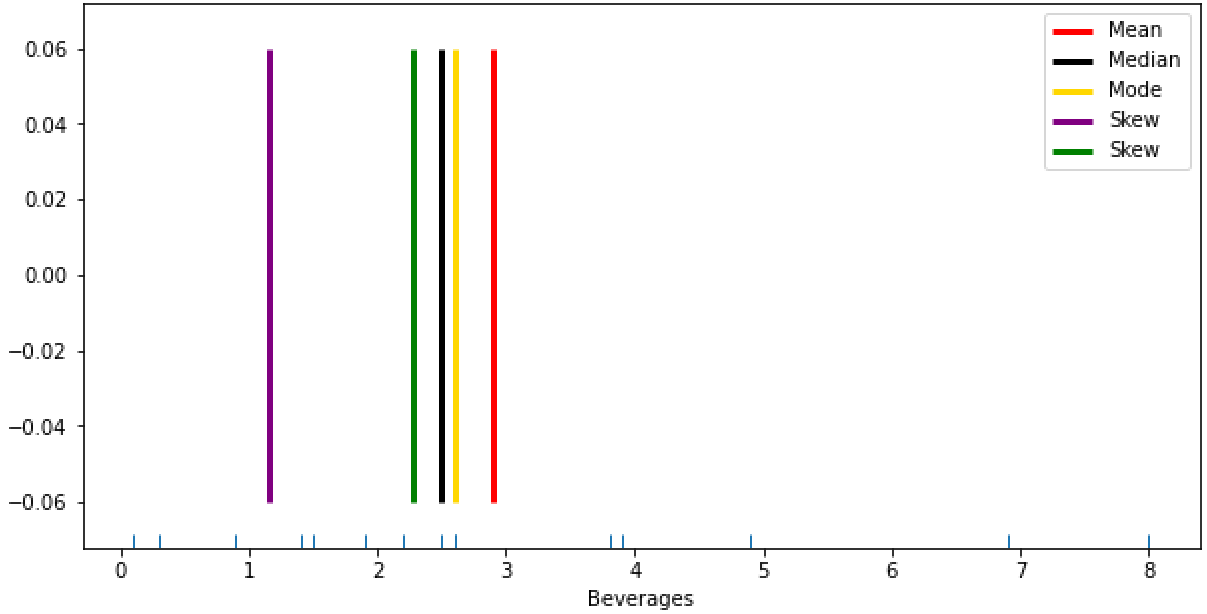
In [14]:

```
plt.figure(figsize=(10,5))
sns.rugplot(data=df, x ="Water")
Mean2 = np.mean(Water)
Median2 = np.median(Water)
Mode2 = df['Water'].mode()[0]
Stddev2 = df['Water'].std()
Skew2 = (Mean - Mode)/Stddev
plt.vlines(Mean2, -0.06, 0.06, color = 'red', linestyle='solid', label = 'Mean', linewidth = 3)
plt.vlines(Median2, -0.06, 0.06, color = 'black', linestyle='solid', label = 'Median', linewidth = 3)
plt.vlines(Mode2, -0.06, 0.06, color = 'gold', linestyle='solid', label = 'Mode', linewidth = 3)
plt.vlines(Skew2, -0.06, 0.06, color = 'purple', linestyle='solid', label = 'Skew', linewidth = 3)
plt.vlines(Stddev2, -0.06, 0.06, color = 'green', linestyle='solid', label = 'Stddev', linewidth = 3)

plt.legend()
plt.show()
```



In [15]:

```
plt.figure(figsize=(10,5))
sns.rugplot(data=df1, x ="Beverages")
Mean3 = np.mean(Beverages)
Median3 = np.median(Beverages)
Mode3 = df1['Beverages'].mode()[0]
Stddev3 = df1['Beverages'].std()
Skew3 = (Mean - Mode)/Stddev
plt.vlines(Mean3, -0.06, 0.06, color = 'red', linestyle='solid', label = 'Mean', linewidth = 3)
plt.vlines(Median3, -0.06, 0.06, color = 'black', linestyle='solid', label = 'Median', linewidth = 3)
plt.vlines(Mode3, -0.06, 0.06, color = 'gold', linestyle='solid', label = 'Mode', linewidth = 3)
plt.vlines(Skew3, -0.06, 0.06, color = 'purple', linestyle='solid', label = 'Skew', linewidth = 3)
plt.vlines(Stddev3, -0.06, 0.06, color = 'green', linestyle='solid', label = 'Skew', linewidth = 3)

plt.legend()
plt.show()
```



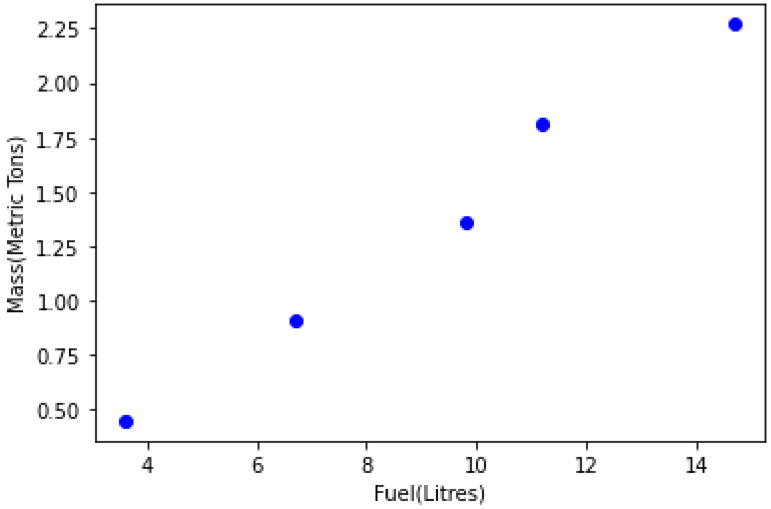
4

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.) 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

In [16]:
```
from scipy.stats import pearsonr
```

In [17]:
```
fuel = [3.6, 6.7, 9.8, 11.2, 14.7]
mass = [0.45, 0.91, 1.36, 1.81, 2.27]
```

In [18]:
```
plt.scatter(fuel, mass, c="blue")
plt.xlabel("Fuel(Litres)")
plt.ylabel("Mass(Metric Tons)")
plt.show()
```



In [19]:

```
Correlation,_= pearsonr(fuel, mass)
Correlation
```

Out[19]:

```
0.9938681082455859
```

The correlation between the 2 variables is positive. As the correlation coefficient is very close to 1, this suggests a highly linear relationship.

5

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

In [20]:

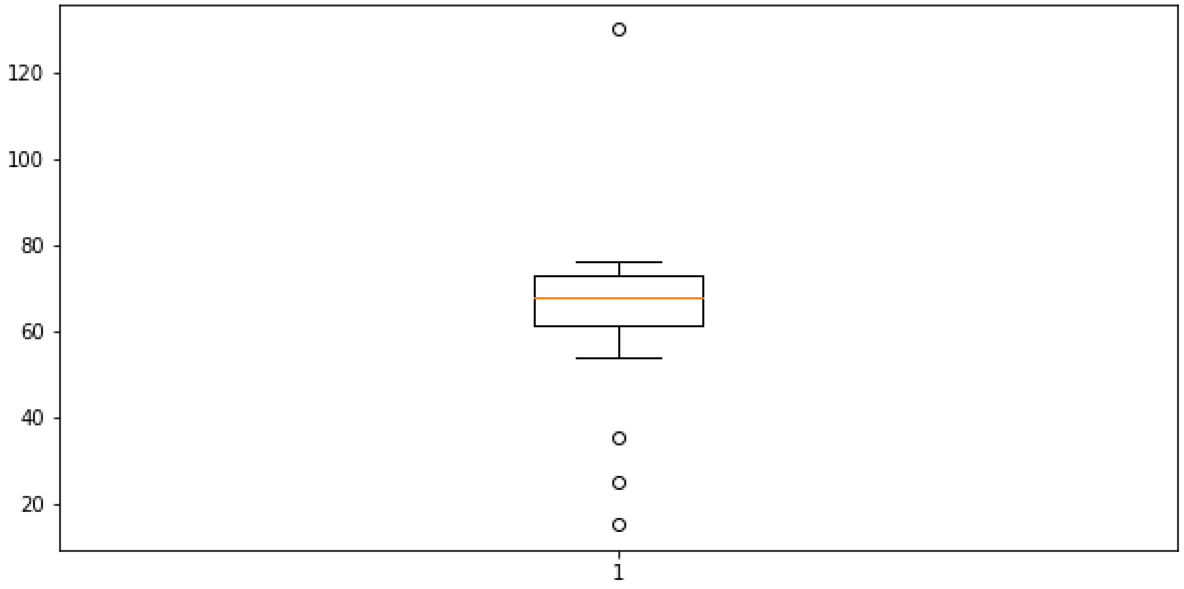
```
chairs = [ 35, 54, 60, 65, 66, 67, 69, 70, 72, 73, 75, 76, 54, 25, 15, 60, 65, 66, 67, 69, 70, 72, 130, 73, 75, 76]
```

In [21]:

```
df = pd.DataFrame(chairs,columns = ['chairs'] )
```

In [22]:

```
fig = plt.figure(figsize=(10, 5))
plt.boxplot(chairs)
plt.show()
```

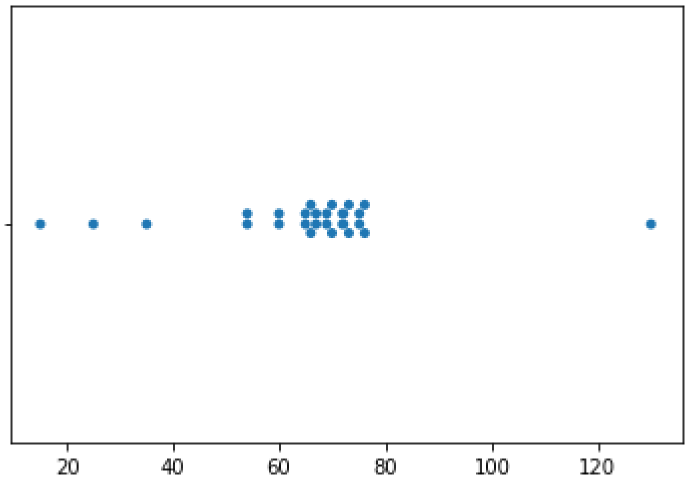


In [23]:

```
sns.swarmplot(x=chairs)
```

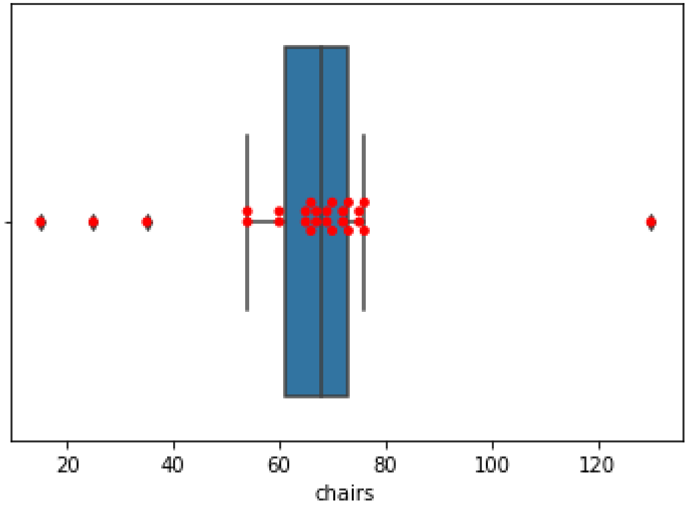
Out[23]:

```
<AxesSubplot:>
```



In [24]:

```
ax = sns.boxplot(x='chairs', data=df)
ax = sns.swarmplot(x='chairs', data=df, color="red")
plt.show()
```



- Conclusion : 4 outliers

6

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

In [25]:

```
sn = np.random.normal(size=30)
ln = np.random.lognormal(3, 1, 30)
```

In [26]:

```
sn
```

Out[26]:

```
array([-0.76401459, -0.21697936,  0.4631075 ,  0.88338833, -1.26161874,
        0.07395699,  0.55953427, -0.98330089,  0.81801783,  0.93138275,
        1.04006755, -0.74149259, -0.33559994,  1.55761845, -1.30012583,
       -0.71671738,  0.16986862, -1.14933215, -0.54094183, -0.96154014,
        1.98048075, -0.08905716, -0.85226255, -0.62852498,  1.3764147 ,
       -1.26660293,  0.17820984, -0.22876266,  0.02143712,  2.23869296])
```

In [27]:

```
ln
```

Out[27]:

```
array([52.64830296,  4.44776895, 68.03526209, 15.26492166,  8.04979778,
       21.42282455,  3.49976131, 14.00481402, 40.85936662, 26.94701878,
        4.08564524,  9.60461719, 33.82539172, 21.37563186, 29.53531179,
        6.92329662, 10.97227828,  5.95593145, 16.60260003, 27.59886931,
        9.23939984, 47.07574548, 23.20345594,  4.66601272,  9.00245117,
        4.388701  ,  9.64647547, 14.10409734,  3.58151431, 22.14493556])
```

In [28]:

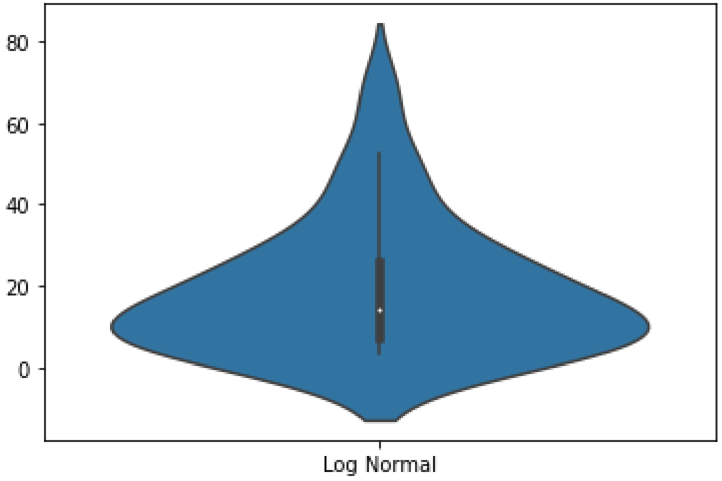
```
df = pd.DataFrame(sn, columns = ['Standard Normal'])
df1 = pd.DataFrame(ln, columns = ['Log Normal'])
```

In [29]:

```
sns.violinplot(data = df1) #lognormal
```

Out[29]:

```
<AxesSubplot:>
```

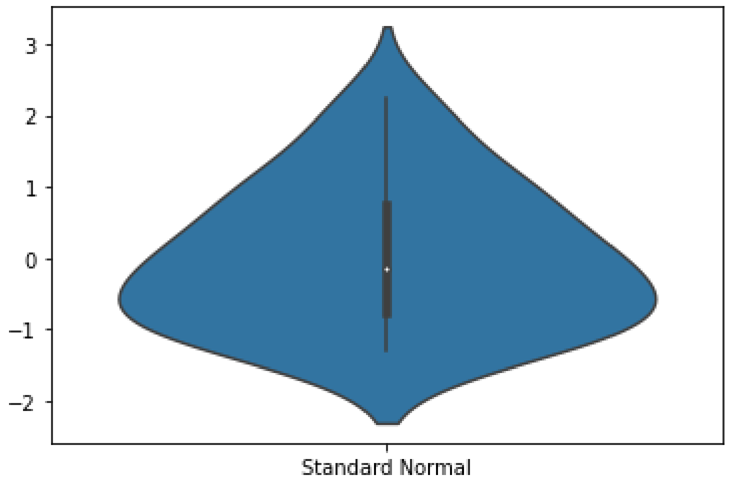


In [30]:

```
sns.violinplot(data = df)
```

Out[30]:

```
<AxesSubplot:>
```



7

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. ShopCategory Quarter 1 Quarter 2 Quarter 3 Quarter 4 Textile 10 6 8 13 Jewellery 5 5 2 4 CleaningEssentials 15 20 16 15 Cosmetics 14 10 21 11

In [31]:

```
from math import pi
```

In [32]:

```
# Set data
df = pd.DataFrame({
'Shop_Cat': [' Textile ','Jewellery', 'Cleaning Essentials','Cosemtics'],
'Q1': [10, 5, 15, 14],
'Q2': [6, 5, 20, 10],
'Q3': [8, 2, 16, 21],
'Q4': [13, 4, 15, 11]
})

print(df.to_string(index=False))
```

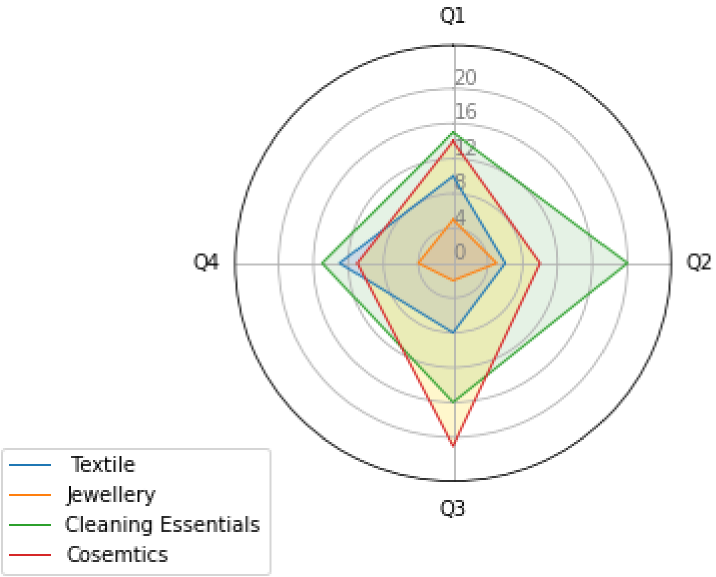
```
            Shop_Cat  Q1  Q2  Q3  Q4
```

```
           Textile   10   6   8  13
         Jewellery    5   5   2   4
Cleaning Essentials   15  20  16  15
          Cosemtics   14  10  21  11
```

In [33]:

```python
# Create radar plot background

# number of variables
categories = list(df)[1:]
N = len(categories)

# What will be the angle of each axis in the plot? (we divide the plot / number of variable)
angles = [n / float(N) * 2 * pi for n in range(N)]
angles += angles[:1]

# Initialise the spider plot
ax = plt.subplot(111, polar=True)

# If you want the first axis to be on top:
ax.set_theta_offset(pi/2)
ax.set_theta_direction(-1)

# Draw one axe per variable + add labels labels yet
plt.xticks(angles[:-1], categories)

# Draw ylabels
ax.set_rlabel_position(0)
plt.yticks(list(range(0,24,4)), list(range(0,24,4)), color="grey", size=10)
plt.ylim(0,25)

#Draw from each shop the number of ads in each quarter

#Shop 1
values=df.loc[0].drop('Shop_Cat').values.flatten().tolist()
values += values[:1]
ax.plot(angles, values, linewidth=1, linestyle='solid', label=df['Shop_Cat'][0])
ax.fill(angles, values, 'b', alpha=0.1)

#Shop 2
values=df.loc[1].drop('Shop_Cat').values.flatten().tolist()
values += values[:1]
ax.plot(angles, values, linewidth=1, linestyle='solid', label=df['Shop_Cat'][1])
ax.fill(angles, values, 'r', alpha=0.1)
```

```
#Shop 3
values=df.loc[2].drop('Shop_Cat').values.flatten().tolist()
values += values[:1]
ax.plot(angles, values, linewidth=1, linestyle='solid', label=df['Shop_Cat'][2])
ax.fill(angles, values, 'g', alpha=0.1)

#Shop 4
values=df.loc[3].drop('Shop_Cat').values.flatten().tolist()
values += values[:1]
ax.plot(angles, values, linewidth=1, linestyle='solid', label=df['Shop_Cat'][3])
ax.fill(angles, values, 'gold', alpha=0.2)


plt.legend(loc='upper right', bbox_to_anchor=(0.1, 0.1))

plt.show()
```



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

In [34]:

```python
import pygal
import cairosvg
```

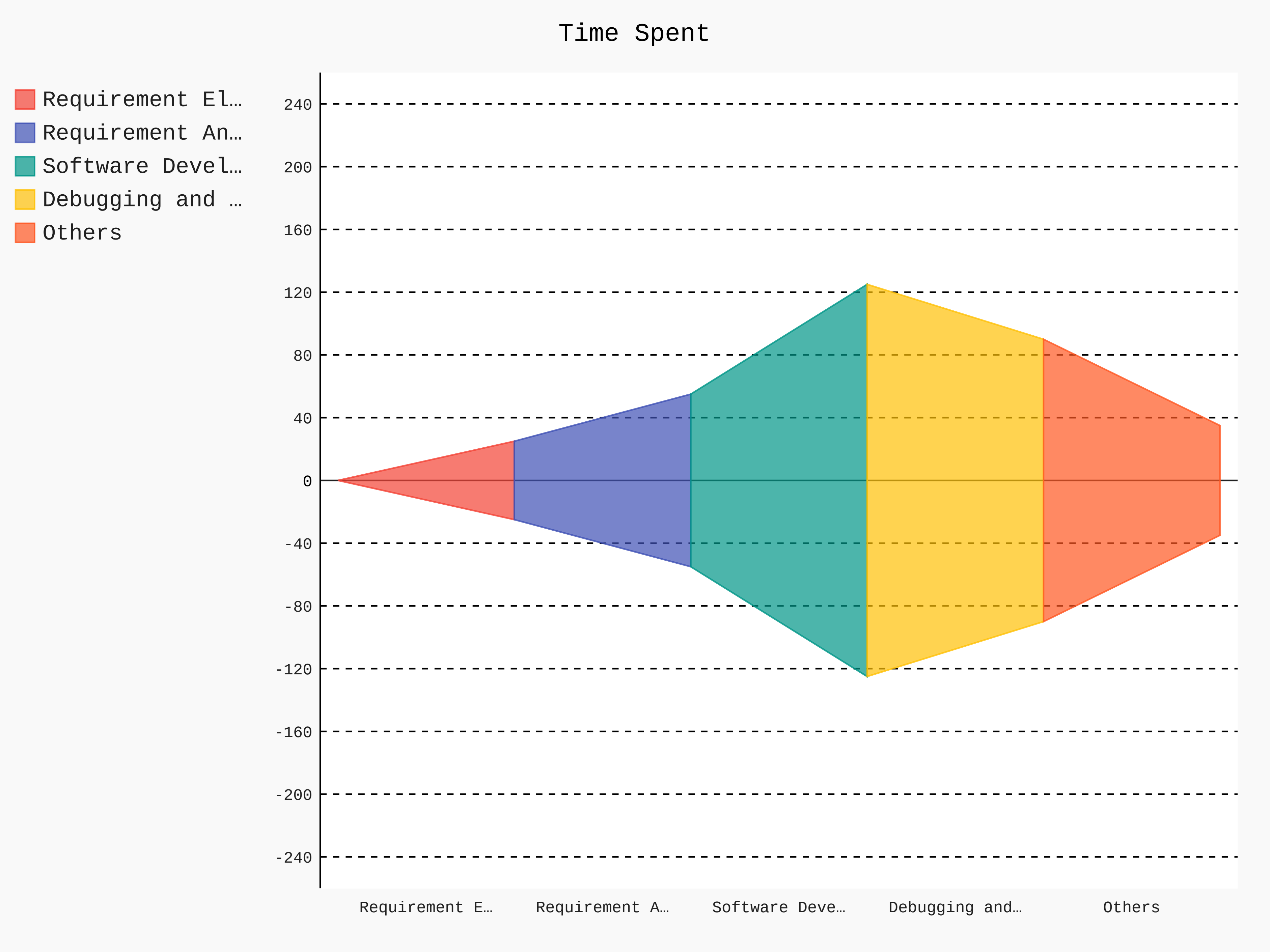
In [35]:

```python
funnel1 = pygal.Funnel()
funnel1.title = 'Time Spent'
```

In [36]:

```python
funnel1.add('Requirement Elicitation', [50])
funnel1.add('Requirement Analysis', [110])
funnel1.add('Software Development', [250])
funnel1.add('Debugging and Testing', [180])
funnel1.add('Others', [70])
```

Out[36]:


Time Spent
Requirement El…
Requirement An…
Software Devel…
Debugging and …
Others
240
200
160
120
80
40
0
-40
-80
-120
-160
-200
-240
Requirement E…
Requirement A…
Software Deve…
Debugging and…
Others

9

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.

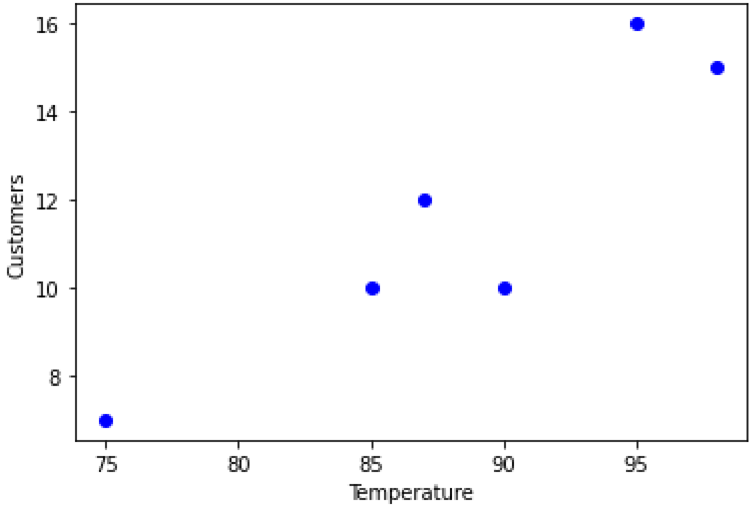
Temperature	Number of Customers
98	15
87	12
90	10
85	10
95	16
75	7

In [37]:

```
Temperature = [98, 87, 90, 85, 95, 75]
Customers = [15, 12, 10, 10, 16, 7]
```

In [38]:

```
plt.scatter(Temperature, Customers, c="blue")
plt.xlabel("Temperature")
plt.ylabel("Customers")
plt.show()
```



In [39]:

```
Correlation, _ = pearsonr(Temperature, Customers)
Correlation
```

Out[39]:

```
0.9117671365080744
```

The correlation between the 2 variables is positive. As the correlation coefficient is very close to 1, this suggests a highly linear and real relationship.

- More business in warmer months.