Hrishikesh Rajesh Menon COE18B024 Problem-Set 2

QUESTION 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.

Modules Used:-

Matplotlib

Approach:-

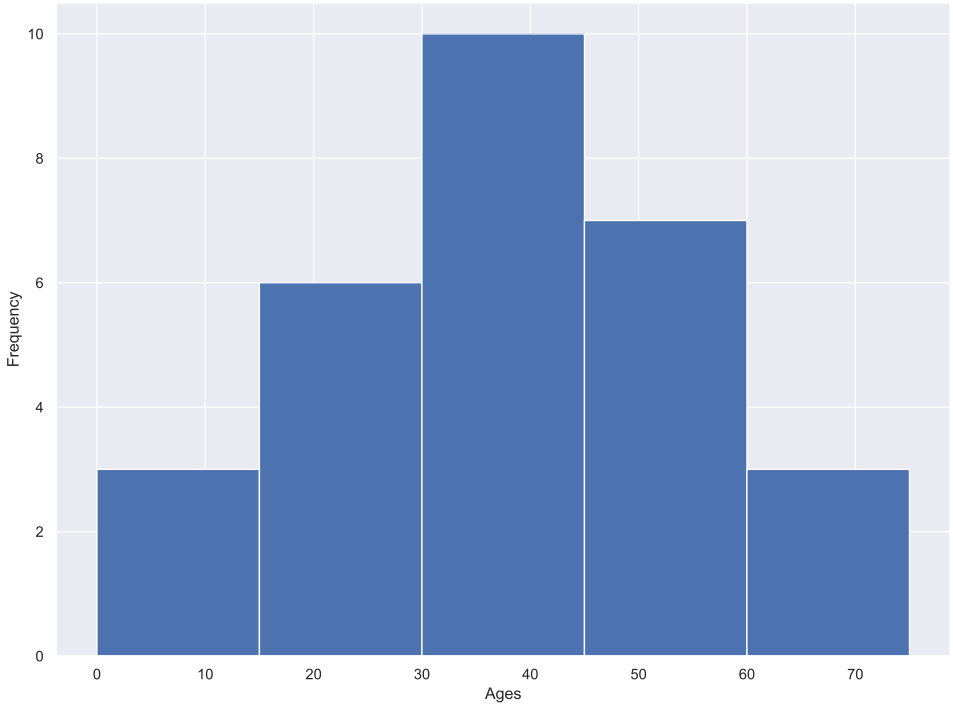
- Input the age details into a list
- Preprocessing data
- Using a matplotlib to create the histogram

In [118…

```
import pandas as pan
import numpy as nps
import random
import matplotlib.pyplot as plt

Age_samps=[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]
```

In [131…

```
# Using Matplotlib default histogram
plt.hist(Age_samps,  bins=range(0,max(Age_samps), 15) )
plt.xlabel("Ages")
plt.ylabel("Frequency")
plt.show()
```



Conclusion :- Bell Shaped Histogram

QUESTION 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.

Modules Used:-

Stemgraphic

Approach:-

- Input the age details into a list
- Preprocessing data
- Using a matplotlib to create the stem-leaf plot

In [137…
```
import stemgraphic

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]

stemgraphic.stem_graphic(points,scale=5)
```

Out[137…
```
(<Figure size 540x162 with 1 Axes>,
 <matplotlib.axes._axes.Axes at 0x270167ac7c0>)
```

Conclusion

The Outliers are 39 and 10 (7.8 x 5 = 39.0 and 2.0 x 5 =10)

Shape - Unimodal

QUESTION 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

Modules Used:-

Seaborn

Numpy

Scipy

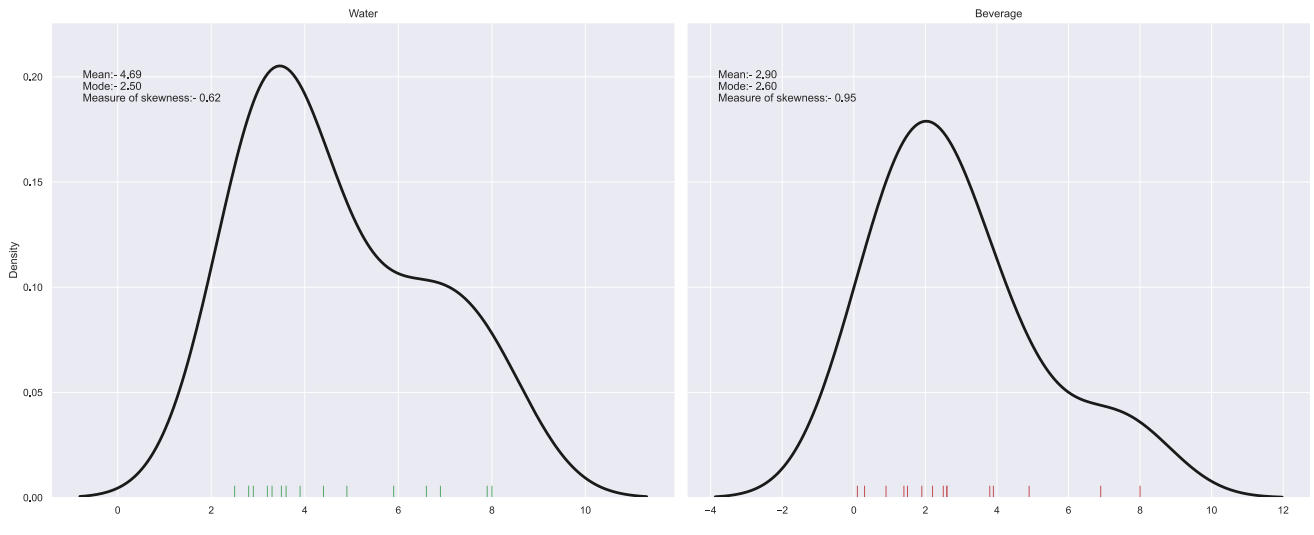
Matplotlib

Approach:-

- Input the values into a numpy array
- Preprocessing data
- Using a matplotlib to create the density and rug plot
- Using Numpy and Scipy to calculate mean ,mode, and skew respectively

In [148…

```
import seaborn as sns
import scipy.stats as st


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]
Bevr= [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]


watarr=nps.array(Water)
bevarr=nps.array(Bevr)

fig, (ax1,ax2) = plt.subplots(nrows=1, ncols=2 , figsize=(20,8) , sharey=True ,tight_layout = True)
sns.distplot(watarr, rug=True, hist=False, rug_kws={"color": "g"},
    kde_kws={"color": "k", "lw": 3},ax=ax1)
sns.distplot(bevarr, rug=True, hist=False, rug_kws={"color": "r"},
    kde_kws={"color": "k", "lw": 3},ax=ax2)

#MEAN,MODE,AND SKEWNESS of WATER

# print(st.mode(watarr)[0])
ax1.text(0.05, 0.9, "Mean:- {:.2f} \nMode:- {:.2f} \nMeasure of skewness:- {:.2f}".format(nps.mean(watarr),st.mode(watarr)[0][0],st.skew(watarr) ) ,horizontalalignment='left', verticala

ax2.text(0.05, 0.9, "Mean:- {:.2f} \nMode:- {:.2f} \nMeasure of skewness:- {:.2f}".format(nps.mean(bevarr),st.mode(bevarr)[0][0],st.skew(bevarr) ) ,horizontalalignment='left', verticala


ax1.set_title("Water")
ax2.set_title("Beverage")

plt.show()
```



QUESTION 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

Modules Used:-

Numpy

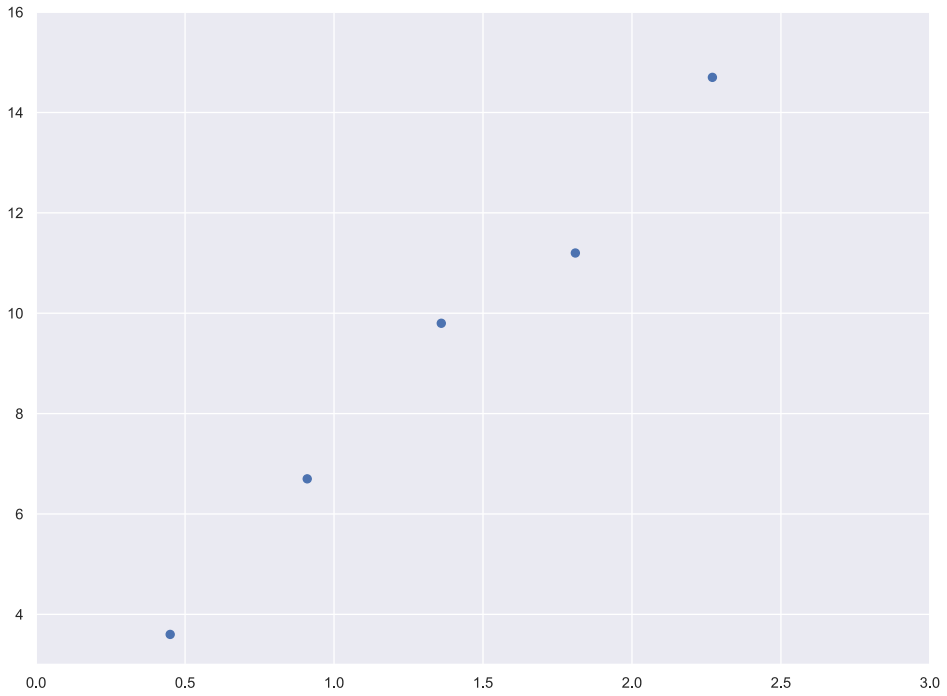
Matplotlib

Approach:-

- Input the age details into a list
- Preprocessing data
- Using a matplotlib to create the scatterplot
- Using Numpy to calculate correlation

In [122...

```
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.xlim(0,3)
plt.ylim(3,16)

plt.text(0,1,"Correlation Coefficient:-{:.5f} ".format(nps.corrcoef(fuel,mass)[0][1]))
plt.show()
```



Correlation Coefficient:-0.99387

Conclusion :- Positive and Linear Correlation

QUESTION 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

Modules Used:-

Seaborn

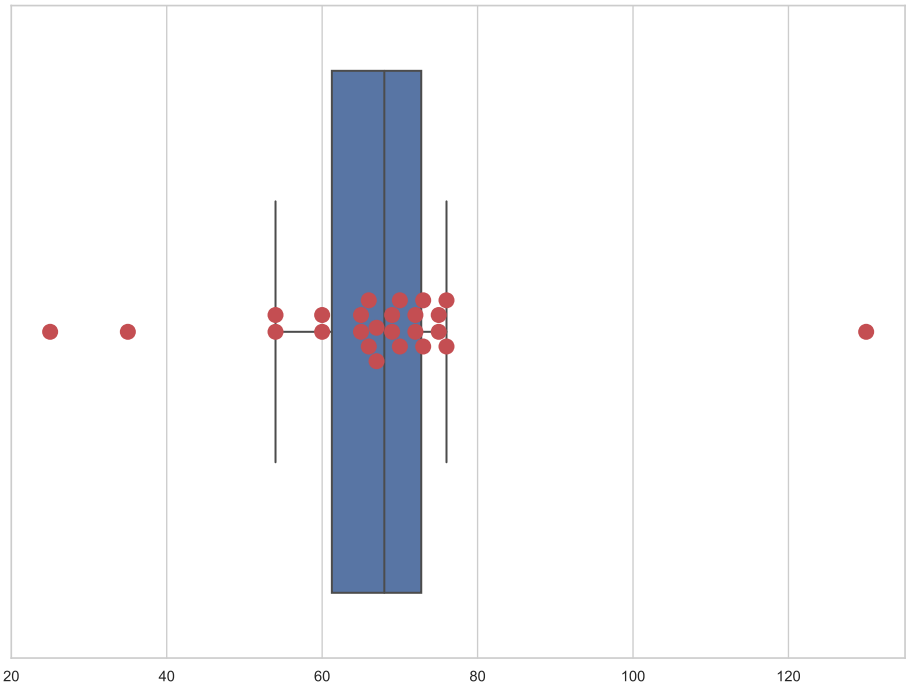
Matplotlib

Approach:-

- Input the age details into a list
- Using a matplotlib and seaborn to create the boxplot and swarmplot

In [149...

```
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]

sns.set_theme(style="whitegrid")

ax = sns.boxplot(x=chairs)
ax.set(xlim=(20,135))
sns.set(rc={'figure.figsize':(12,9)})
ax = sns.swarmplot(x=chairs,color='r',size=12)
```



Conlusion - 3 Outliers

QUESTION 6

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

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

Modules Used:-

Numpy

Seaborn

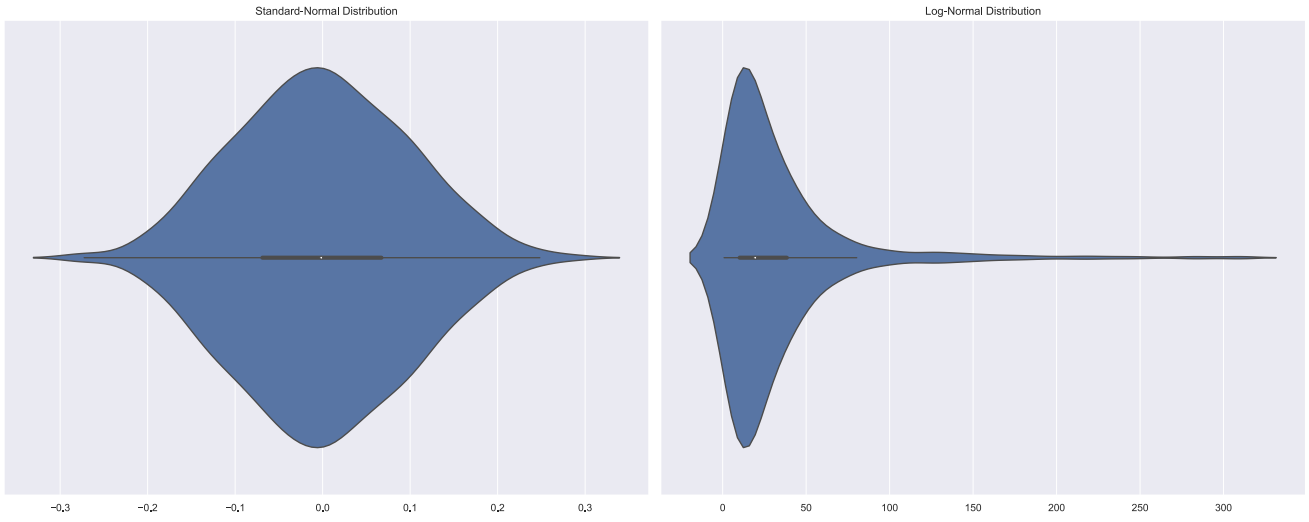
Matplotlib

Approach:-

- Using numpy we can generate random numbers using normal and log normal
- Using a matplotlib and seaborn to create the violin plot

In [124…

```
mu, sigma = 0, 0.1 # mean and standard deviation for normal distribution

norm = nps.random.normal(mu, sigma, 1000)

mu, sigma = 3., 1. # mean and standard deviation for log normal distrubution
lognorm = nps.random.lognormal(mu, sigma, 1000)


fig, (ax1,ax2) = plt.subplots(nrows=1, ncols=2 , figsize=(20,8) , sharey=True ,tight_layout = True)
sns.violinplot(ax=ax1,x=norm)
sns.violinplot(ax=ax2,x=lognorm)

ax1.set_title("Standard-Normal Distribution")
ax2.set_title("Log-Normal Distribution")

fig.show()
```



QUESTION 7

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.

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

Modules Used:-

Pandas

Matplotlib

Approach:-

- Using pandas we can create the table as a Data frame
- Using a matplotlib we can create the background of the radar charts
- And then by calculating the angles we can draw each radar chart

In [125…

```
from math import pi

# Set data
df = pan.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 [126…

```
# 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()
```


Q1
20
16
12
8
4
0
Q2
Q3
Q4
Textile
Jewellery
Cleaning Essentials
Cosemtics

QUESTION 8

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

Modules Used:-

Matplotlib

Plotly

Approach:-

- Create a Dictionary which stores percentage of time and the step
- Using a matplotlib and plotly we can create the funnel chart

In [127…

```
import plotly.express as px

#Setting up data

time = [50,110,250,180,70]

tottime = sum(time)

time_perc = [time[i]/tottime * 100 for i in range(len(time))]

#round off to 2 decimal places

time_percent= [round(time_perc[i],2) for i in range(len(time_perc))]

data = dict(values=time_percent,
            labels=['Requirement Elicitation','Requirement Analysis ','Software Development ','Debugging & Testing ','Others'])

fig = px.funnel(data, y='labels', x='values')

fig.update_layout(
    title="Product Development",
    yaxis_title="Step",
    font=dict(
        size=12,
        color="#7f7f7f"
    )
)

fig.show()
```

QUESTION 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

Modules Used:-

Numpy

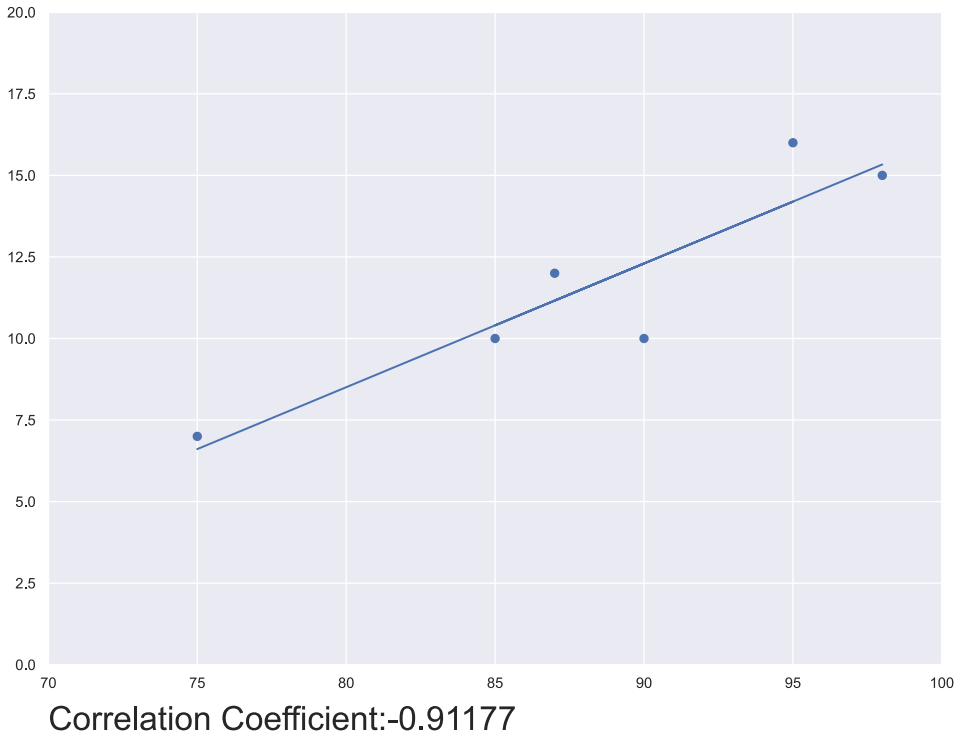
Matplotlib

Approach:-

- Create matplotlib to create a scatterplot of the two variables and using numpy to find line of best fit
- Using numpy to find the correlation between the two variables

In [150…

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

#scatter plot
plt.scatter(temp,customers)
plt.xlim(70,100)
plt.ylim(0,20)

#finding line of best fit
tempnp=nps.array(temp)
custnp=nps.array(customers)
m , b = nps.polyfit(tempnp,custnp,1)

#plot line of best fit
plt.plot(tempnp, m * tempnp + b)
plt.rcParams.update({'font.size': 25})
plt.text(70,-2,"Correlation Coefficient:-{:.5f} ".format(nps.corrcoef(temp,customers)[0][1]) )
# plt.rcParams.update({'font.size': 25})
plt.show()
```



Conclusion :- Postive Correlation , Proving that there is more business in the warmer months than cooler months

---