**Course Code: ANL252 Python for Data Analytics**

**Tutor-Marked Assignment**

**July 2022 Presentation**

**SUSS PI No: M2170763**

**Name: Reta D/O Rajakumar**

**Submission Date: 14 August 2022**

**Question 1 (a)**

**Image 1**

*Summarised PivotTable to Derive Data for Scatterplot for Chart 1*

**Graphical user interface, application, table, Excel

Description automatically generated**

**Image 2**

*Using summarised PivotTable to Plot Scatterplot as seen in Chart 1*

**Chart, scatter chart

Description automatically generated**

**Chart 1**

*Scatter Plot of Absentees vs Number of Absence Days In Previous Years*

As shown in Chart 1, there is no correlation or relationship between the number of Absentees and number of Absence days in the previous year. The data points are spread out and there is no trend. This suggests that the number of absence days is not related to the total number of absentees, and it is random.

**Image 3**

*Summarised PivotTable to Derive Data for Group Bar chart for Chart 2*

*Chart, waterfall chart

Description automatically generated*

**Chart 2**

*Group Bar Chart of Absentees vs Number of Absence Days In Previous Years*

*Note*. Bins used is 4.

Chart 2 show the frequency of performance type vs the number of absence days took in previous year. The highest frequency of performance type is “Meet” and the lowest is “PIP”. “Meet” performance is significantly higher than all other types. The means that majority of the employees meets their performance KPI, neither exceling or underperforming.

As seen from the orange “Meet” bars, there is a slight downward trend, it means that the higher number of absence days taken in previous years, the lesser number of employee with a “Meet” performance.

There is a slight upward curve trend with yellow “To improve” bars, which means the higher number of absence days taken in previous years, the more number of employee with a “To improve” performance.

Blue “Exceed” bars seems random, meaning there is no trend between absence and excellent performance.

**Question 1 (b)**

Below is the python code for creating Chart 1 as above.

**# CHART 1 , SCATTERPLOT**

# using pd as an alias for pandas library

import pandas as pd

# using plt as an alias for matplotlib.pyplot

import matplotlib.pyplot as plt

# read and show head of the dataset

df = pd.read\_csv("TMA\_data.csv")

df.head()

# counts of Absence frequency

value\_counts = df['Absence'].value\_counts()

# converting to df and assigning new names to the columns

df\_value\_counts = pd.DataFrame(value\_counts)

df\_value\_counts = df\_value\_counts.reset\_index()

df\_value\_counts.columns = ['Number of Absence days in previous year', 'Absentees'] # change column names

df\_value\_counts

df1 = df\_value\_counts.sort\_values(by=['Number of Absence days in previous year'], ascending=True)

print(df1)

# plot scatterplot

df1.plot.scatter(x = 'Number of Absence days in previous year', y = 'Absentees', )

plt.grid()

plt.xlim(0 , 25)

plt.ylim(0 , 25)

**Images 4 to 6**

*Python code for Chart 1, Scatterplot*

Graphical user interface, text

Description automatically generated with medium confidence

A picture containing table

Description automatically generated

Chart, scatter chart

Description automatically generated

Below is the python code for creating Chart 2 as above.

**# CHART 2 , GROUP BAR CHART PLOT**

dfgbc = pd.read\_csv("TMA\_data.csv")

dfgbc.head()

# bin the Absence group

dfgbc['Absence group'] = pd.cut(x=dfgbc['Absence'], bins=[0,5,10,15,20], labels=['1-5','6-10','11-15','16-20'])

# tabulation of Absence and PerformanceScore columns

dfgbc2 = pd.crosstab(dfgbc['Absence group'], dfgbc['PerformanceScore'])

# display(dfgbc2)

print(dfgbc2)

# plot bar chart

colors = ['royalblue','darkorange','grey','gold']

dfgbc2.plot(kind='bar',

rot=0,

stacked=False,

xlabel='Absence days in previous year',

ylabel='Performance',

fontsize=12,

width=1,

color = colors,

title='Performance vs Absence days in previous year')

**Images 7 to 9**

*Python code for Chart 2, Grouped Bar Chart*

Table

Description automatically generated

A picture containing table

Description automatically generated

Chart

Description automatically generated

**Question 1(c)**

import datetime

import numpy as np

import pandas as pd

# import csv file and see data type

dfdate = pd.read\_csv("TMA\_data.csv")

dfdate.info() # Row 7 and 8 are not in date format

# replace empty LeftDate with 1 May 2022

defaultdate = pd.to\_datetime('2022-5-1')

dfdate['LeftDate'] = dfdate['LeftDate'].fillna(defaultdate)

# convert JoinDate and LeftDate column to datetime pandas

dfdate['JoinDate'] = pd.to\_datetime(dfdate['JoinDate'])

dfdate['LeftDate'] = pd.to\_datetime(dfdate['LeftDate'])

# to check row 7 and 8 successful conversion to datetime

dfdate.info()

# Calculates the difference between two dates

dfdate['diff'] = dfdate['LeftDate'] - dfdate['JoinDate']

dfdate['ServiceLength'] = dfdate['diff'] / np.timedelta64(1, 'D')

dfdate.head()

dfdate['ServiceLength']= dfdate['ServiceLength']/365

minvalue = dfdate['ServiceLength'].min()

maxvalue = dfdate['ServiceLength'].max()

avgvalue = dfdate['ServiceLength'].mean()

minvalue = round(minvalue,1)

maxvalue =round(maxvalue,1)

avgvalue = round(avgvalue,1)

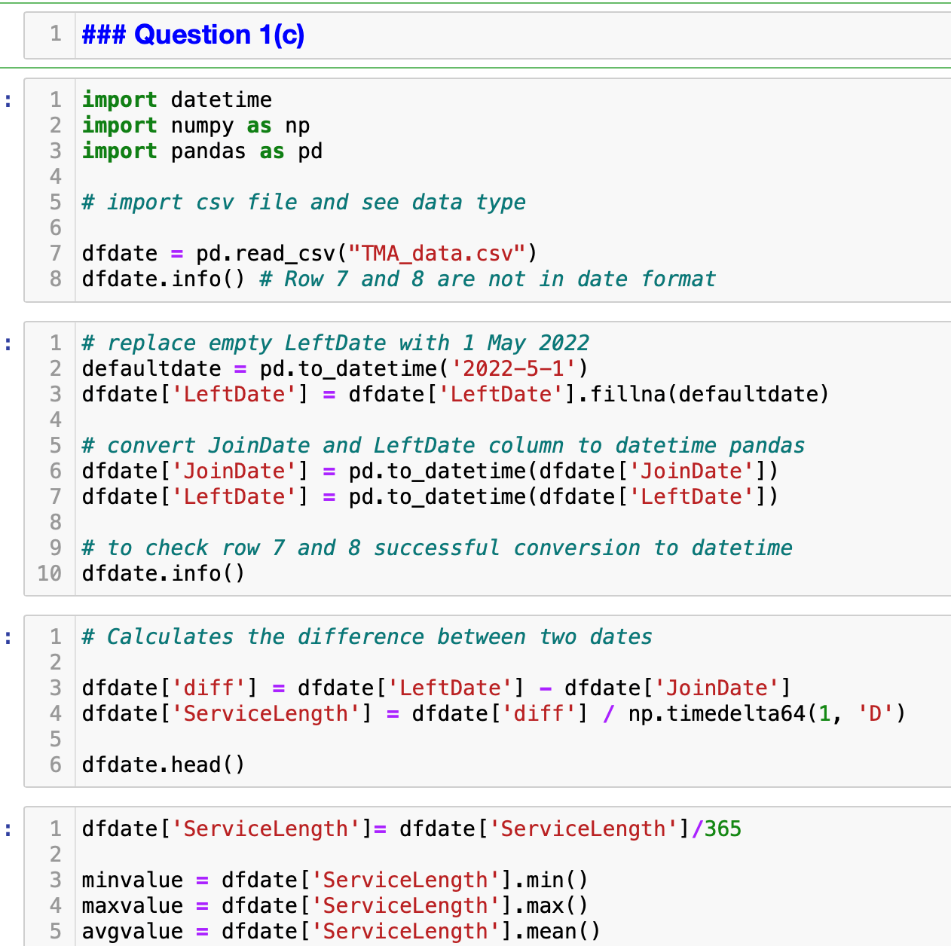
print(minvalue,maxvalue,avgvalue)

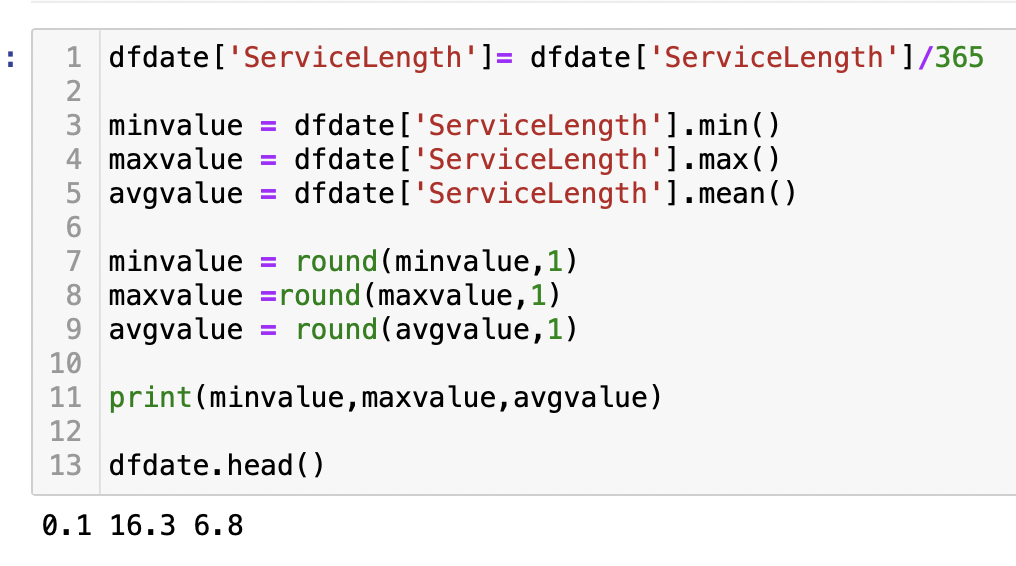
dfdate.head()

The minimum, maximum and average is 0.1, 16.3 and 6.8 respectively.

**Images 10 to 11**

*Python code for Question 1(c)*

****

****

**Question 1(d)**

import pandas as pd

# import csv file

data = pd.read\_csv("TMA\_data.csv")

# extract Staff column as list

namelist = data['Staff'].tolist()

# Start a loop that will run until the user enters 'quit'.

name = str(input("Please enter staff full name for query:"))

while name != 'quit':

print("Staff name is: " + name)

if name in namelist:

print("Staff is/was employed here!")

else:

print("Staff does not exist.")

if name != 'quit':

name = input("Continue to enter staff full name for query, or 'quit' to end: ")

if name == 'quit':

print("Goodbye")

**Images 12**

*Python code for Question 1(d)*

**Graphical user interface, text, application

Description automatically generated**