

ANL252 PYTHON FOR DATA ANALYTICS

Group Based Assignment

DAMIAN SIM DING XIAN, E2281574

TAN JIA YI, SHANNON, M2210936

ANG YONG XIANG, B2271127

19 FEBRUARY 2023

**Declaration Page**

We, members of group 5, do hereby declare that we each contributed to this assignment and that we collectively agree to a shared grade.

|  |  |  |
| --- | --- | --- |
| Name | Contribution | Signature |
| [DAMIAN SIM DING XIAN](https://canvas.suss.edu.sg/groups/87722/users/99609) (Team Lead) | 1a,1b,1c | Damian Sim |
| [TAN JIA YI, SHANNON](https://canvas.suss.edu.sg/groups/87722/users/74763) | 1a,1b,1c | Shannon Tan |
| [ANG YONG XIANG](https://canvas.suss.edu.sg/groups/87722/users/100773) | 1a,1b,1c | ANG |

Contents

Table of Contents

[Question 1 (a) 4](#_Toc127742733)

[Chart 1: Bar Chart 4](#_Toc127742734)

[Chart 2: Box Plot 5](#_Toc127742735)

[Chart 3: Scatter Plot 7](#_Toc127742736)

[Chart 4: Histogram 8](#_Toc127742737)

[Question 1 (b) 10](#_Toc127742738)

[Question 1 (c) 14](#_Toc127742739)

[Appendix - Q1 (b) Source Codes 16](#_Toc127742740)

[Chart 1 - Bar Chart 16](#_Toc127742741)

[Table 1 - Summary Table for Chart 1 18](#_Toc127742742)

[Chart 2 - Box Plot 19](#_Toc127742743)

[Table 2 - Summary Table for Box Plot 21](#_Toc127742744)

[Chart 3 - Scatter Plot 22](#_Toc127742745)

[Table 3 - Regression Statistic Table for Chart 3 23](#_Toc127742746)

[Chart 4 - Histogram 24](#_Toc127742747)

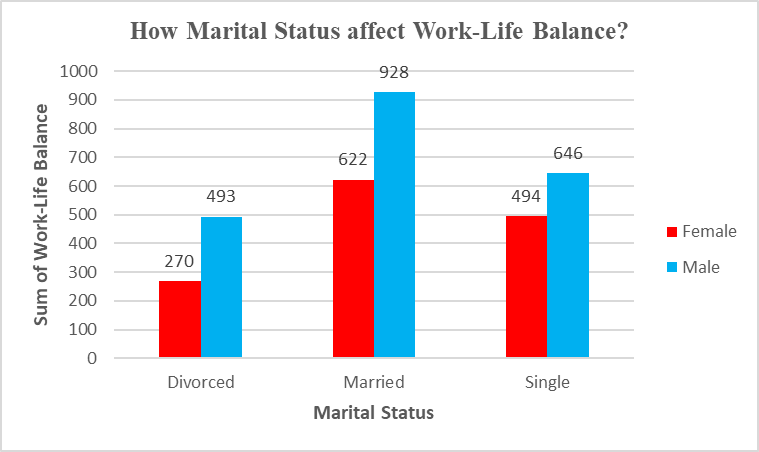
[Table 4 - Frequency Table and Summary Statistic for Histogram 26](#_Toc127742748)

[Appendix - Q1 (c) Source Codes 28](#_Toc127742749)

## Question 1 (a)

Work-life balance is a variable that shows the chances that an employee has to have a healthy, balanced between professional and personal life. It shows how you spent your time outside of work, be it doing things you enjoy or being with the people you care a lot for.

## Chart 1: Bar Chart



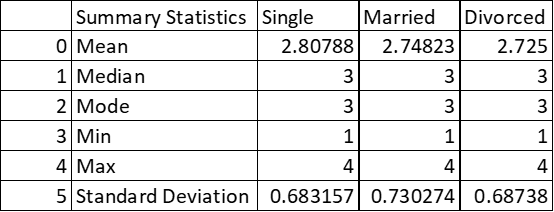
**Chart 1:** Bar Graph to determine relationship between Marital Status and Work-Life Balance

**Chart 1** above shows a comparative bar chart. We have decided to create a bar chart as it is often used to show the visual comparisons between variables very easily, and it is easy to understand. After analysing the dataset given, we have thought of a question whereby everyone would be curious about during their working life. For example, “Does Marital Status affect Work-Life Balance?”.

The bar chart’s x-axis consists of a variable “Marital Status”, it has 3 different categories  Divorced, Married and Single. Each category is split by gender, female and male. The y-axis consists of the sum values of the variable “Work-life Balance”, it shows the employee rating of work-life balance (1-4, where 4 is highest).

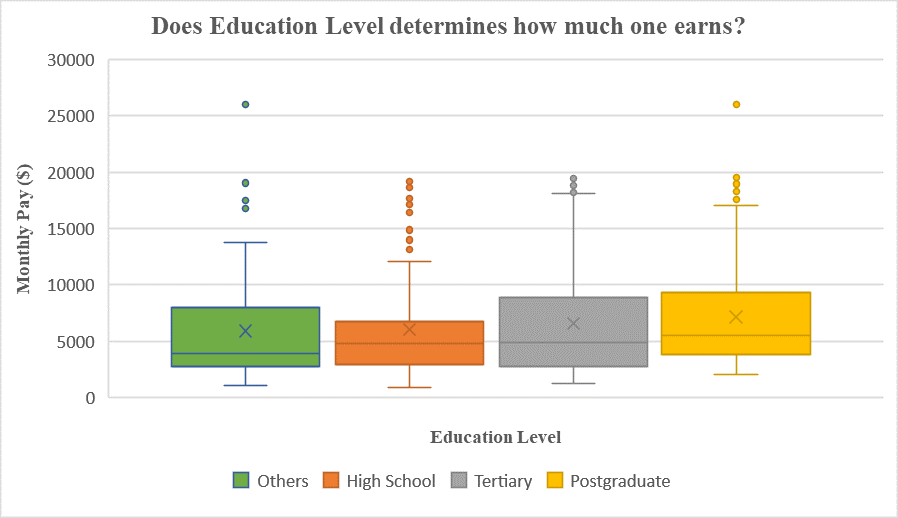
Work-life balance is having the ability to have a healthy, balanced relationship between professional and personal life. It shows how you spent your time outside of work, be it doing things you enjoy or being with the people you care a lot for.

With the data shown in **chart 1**, it can be seen that marital status does affect work-life balance. It can be seen that being married has the highest work-life balance compared to being single and divorced. This could be because being married brings most people happiness and joy, which helps them to increase their motivation in life and do everything at their very best. It also can be seen that the employees who are divorced have the lowest work-life balance. This could be because they might be feeling down and stressed out all the time due to the problems they faced, in both their professional and personal life. Hence, their work-life balance is not good and they would most probably have low job satisfaction as well.

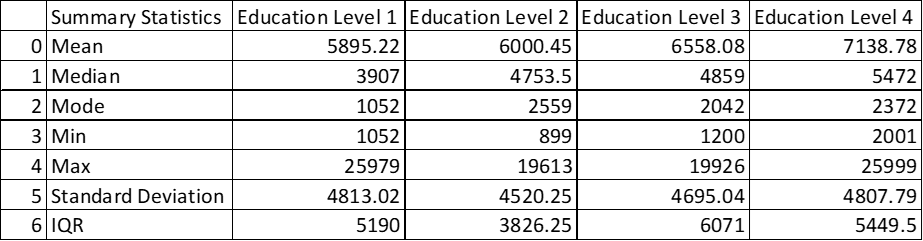


**Table 1:** Summary Statistics of Work Life Balance for each Marital Status

## Chart 2: Box Plot

****

**Chart 2:** Box Plot to show relationship between MonthlyPay and EducationLevel

****

**Table 2:** Summary Statistics of Monthly Pay for each Education Level

**Chart 2** above shows a box plot which separates all of the employees’ monthly pay based on their own education level. This chart is to make comparisons of their income between the 4 types of education levels where (1: Others, 2: High School, 3: Tertiary, 4: Postgraduate) by looking at their location and distribution measures. After comparing, we would also hypothesise whether does Education Level determines how much an employee earns in the organisation.

A box plot is a graph which visualises graphical information the measures of location, dispersion and skewness of the data. The measures which can be seen in a box plot are:

1. Mean
2. Median
3. Minimum and Maximum and,
4. Interquartile Range

**Comparison of location:** It can be seen that ‘Postgraduate’ level 4 has the highest mean, median, minimum and maximum. This shows that the Postgraduate employees earn higher income than the rest of the Education Levels.

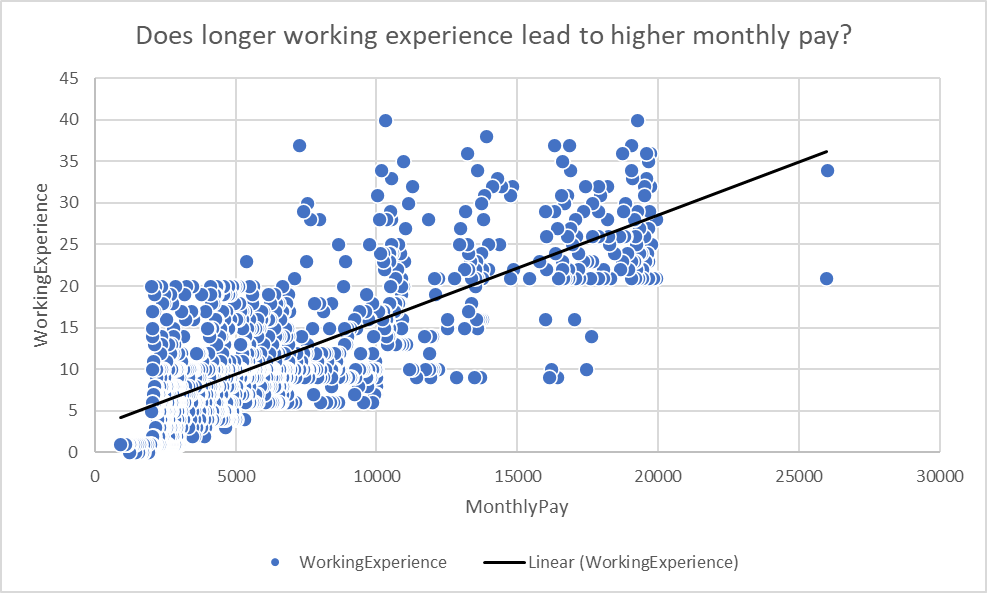
**Comparison of dispersion:** The interquartile ranges for Levels 1, 3 and 4 are reasonably similar as shown by the lengths of their boxes. However, it can be seen that the overall range of the income is much greater for the ‘Postgraduate’ level 4.

**Comparison of skewness:** Based on the box plot above, all of the income for each education levels are positively skewed and distributed, as the whisker and half-box are longer on top of the median than the bottom.

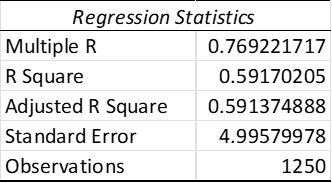
**Comparison of outliers:** It can be seen from the box plot that even though Level 2 which is the ‘High School’ level is considered the lowest education level compared to the rest, there are employees (so-called ‘outliers’) who are earning significantly higher income than most of the employees who have higher education levels than them. Some probable causes could affect these outliers are longer working experience, commitment to the organisation or better job performance which could get them pay bonuses and many more.

Overall, we can conclude that Education Level does not affect employees’ monthly pay as all four boxplot graphs shows closely related income ranges despite their education levels.

## Chart 3: Scatter Plot



**Chart 3:** Scatterplot graph of the relationship between WorkingExperience and MonthlyPay



**Table 3:** Regression Statistics Table between Working Experience and Monthly Pay

**Chart 3** shows a scatterplot chart which shows the correlation between the MonthlyPay of the employees as compared to the Working Experience of each worker. Scatterplots are mainly used to observe the relationships between two numeric variables. It helps us to understand if these variables have a positive/negative correlation.

From this chart, we can understand that there is a positive correlation between both variables and conclusive proof that longer working experience will lead to better monthly pay. The lowest of which, having no experience at all, earned a mere $1,200 monthly as compared to the longest working experience, earning about $19,246 monthly. Even though there is a positive correlation between the variables, the relationship seems moderate as the scatter of the data seems large. We can also tell by the scatterplot that the majority of the data points lie between the range of 0-$20,000 which indicates most of the employees receive that amount monthly.

However, we can see that there are 2 data outliers which are out of the normal distribution earning more than $25000 monthly. These data points could be out of the norm due to various other factors such as their JobRole or FieldOfStudy which commands them a higher MonthlyPay as compared to the normal distribution.

From this Chart, we can conclude that in the normal distribution of the data points, the majority of the monthly payment of the employees is determined by the amount of working experience they have. Therefore, we can say that the longer working experience the employees have, the higher income they could earn.

## Chart 4: Histogram

Chart, histogram

Description automatically generated

**Chart 4:** Histogram chart showing the Income Range for all employees

Table

Description automatically generated

**Table 4a:** Frequency Table of the employees under each bin (Income Range) created

Shape

Description automatically generated with medium confidence

**Table 4b:** Summary Statistics Table for Histogram

**Chart 4** above shows a histogram which shows the range of the income of all the 1250 employees. We created a histogram as it helps us to observe and understand which income range the employees fall into. The histogram’s x-axis consists of 6 bins we have created to group the employees according to their income range starting at $5,000 and increasing every $5,000 interval until it reaches $30,000. The y-axis consists of the number of employees who participated in the excel file which is 1250.

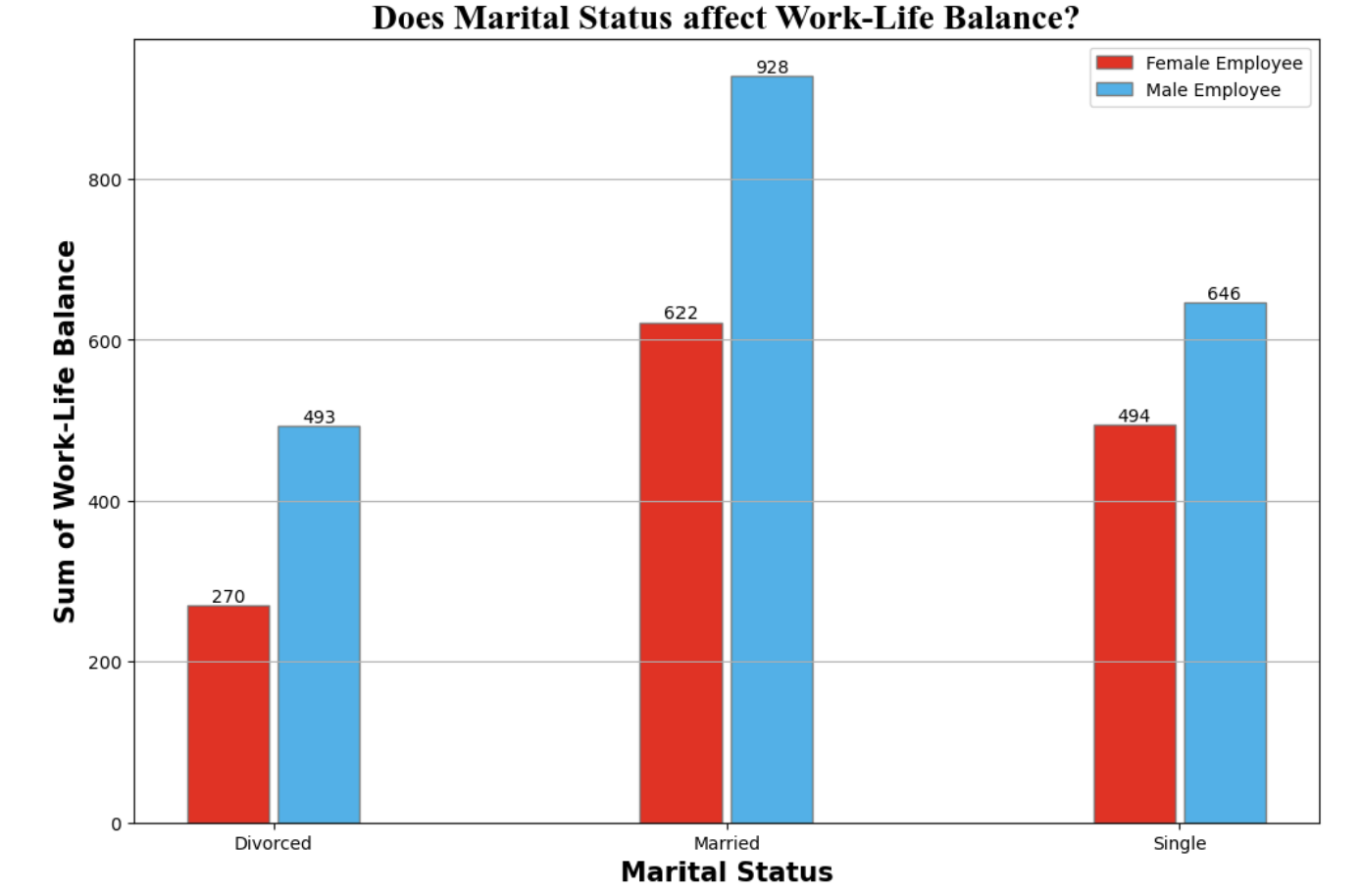
Based on the chart above, we can see that there are 625 employees, which is exactly half of the organisation’s staff, are earning in the income range of $5,000, 385 employees earning in the range of $10,000, 128 employees earning in the $15,000 income range, 110 earning in $20,000 range, and only 2 earning in the $30,000 income range.

Overall, the histogram distribution of the income for the employees is positively skewed as it shows that the values are skewed to the right where it can be seen that the ‘tail’ of the histogram (where the bars are getting shorter) is to the right. Hence, this means that most employees in this organisation are earning an income range of $0-$5,000.

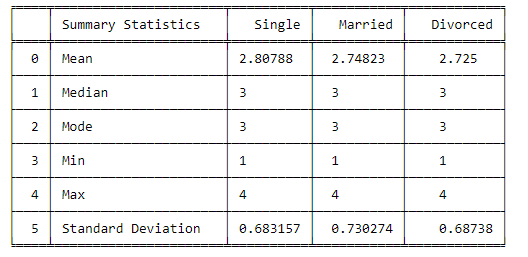
With these observations above, we could help the organisation to determine if their compensation for their employees are comparable to the market rate of their industry. This would help to ensure that every employee has a fair compensation of income determined by the organisation’s standards such as work performance, job role, skills, working attitude etc. This chart would help the organisation to control their pay expenses and ensure every employee is paid fairly.

Thus, we are able to show the number of high income earners in the organisation as well as their range to determine if a pay adjustment is needed for the organisation to ensure a fair compensation throughout their employees.

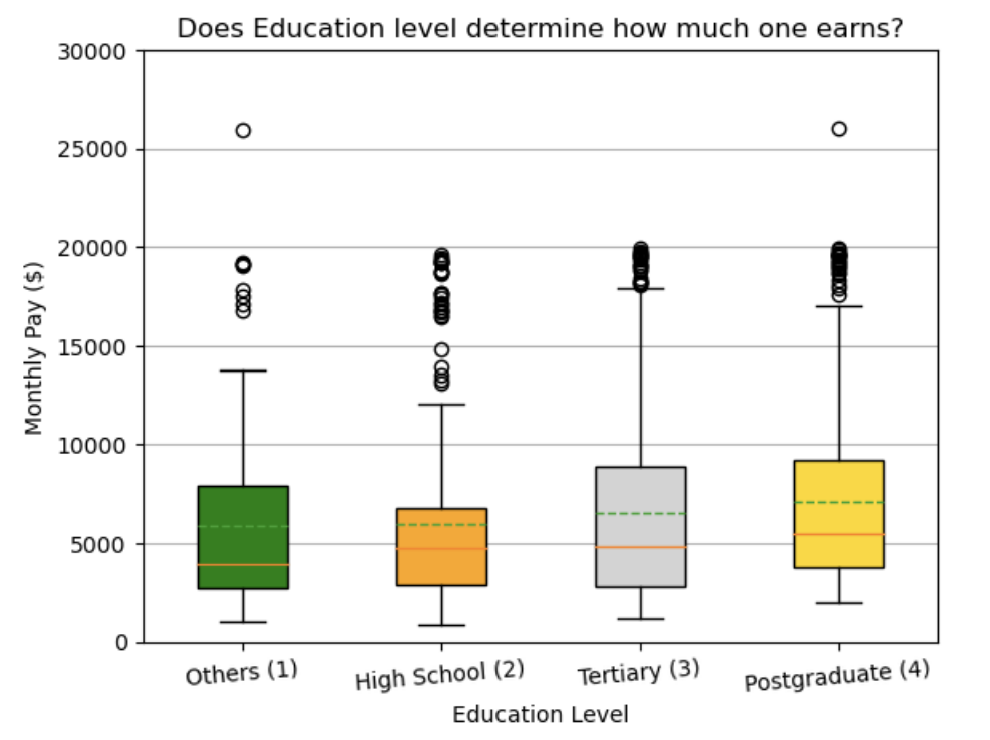
## Question 1 (b)



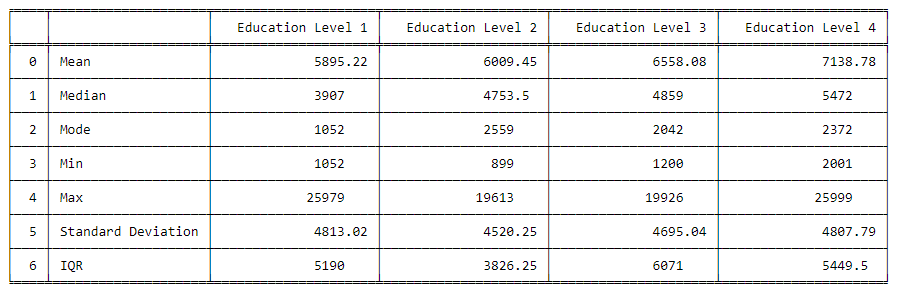
**Bar Chart:** Python output for Chart 1



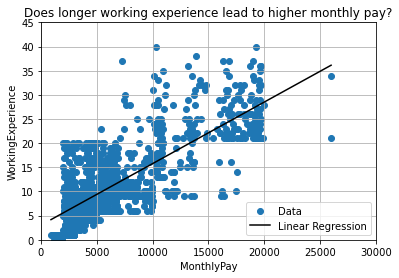
**Table 1:** Summary Statistics Table for Chart 1 (Python)



**Box Plot:** Python output for Chart 2



**Table 2:** Summary Statistics for Chart 2 (Python)



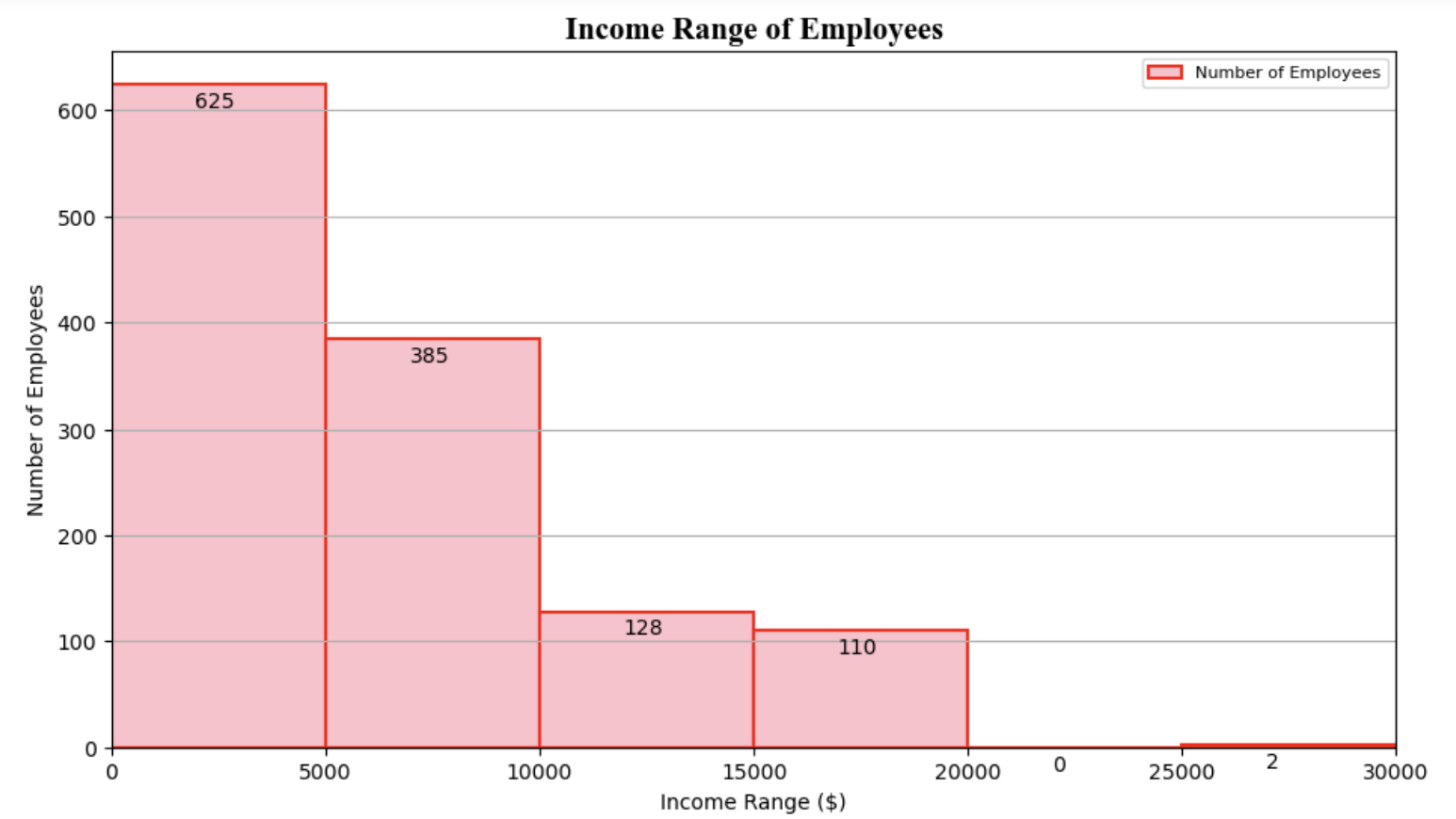
**Scatter Plot:** Python output for Chart 3

The above Graph 3 is the Python output for Chart 3 which shows the positive relationship between ‘WorkingExperience’ and ‘MonthlyPay’.

Table

Description automatically generated

**Table 2:** Summary Statistics for Chart 3 (Python)



**Histogram:** Python output for Chart 4

Table

Description automatically generated

**Table 3**: Frequency Table for Chart 4 (Python)

Table

Description automatically generated

**Table 3**: Frequency Table for Chart 4 (Python)

## Question 1 (c)

**Output of An Interactive Program:**

An User Interactive Program

 Input an option 1 to 4 to view the employee information of an organization.

 1: To view current total staff strength, and a breakdown of staff strength by business unit and gender

 2: To view information of the current total annual staff remuneration expenditure, and the current median monthly salary

 3: To view information of the employees (EmpID), job role and age, ordered by descending age, with 35 or more years of service

 4: To exit the program

---------------------------------------------------------

Please choose your option: 1

---------------------------------------------------------

Total Staff Strength: 1250

Business Development:

Total Number of Female Employees = 159

Total Number of Male Employees = 218

HR:

Total Number of Female Employees = 17

Total Number of Male Employees = 34

Product Development:

Total Number of Female Employees = 327

Total Number of Male Employees = 495

Please choose your option: 2

---------------------------------------------------------

Current total annual staff remuneration expenditure: $8185598.0

Current median monthly salary: $5001.0

Please choose your option: 3

---------------------------------------------------------

      EmpID             JobRole  Age  WorkingExperience

295    1295     Account Manager   59                 38

698    1698     Section Manager   58                 37

711    1711   Product Executive   58                 40

943    1943        Product Head   58                 40

1242   2242     Section Manager   57                 36

120    1120  Manufacturing Head   56                 37

821    1821     Account Manager   56                 36

1226   2226     Section Manager   56                 36

382    1382        Product Head   55                 36

466    1466     Section Manager   55                 37

934    1934     Section Manager   55                 37

1047   2047     Section Manager   55                 35

353    1353        Product Head   53                 35

879    1879     Account Manager   53                 35

Please choose your option: p

You have entered an invalid option

Please re-input your option: 9

---------------------------------------------------------

Please re-input your option as you have entered an invalid option: 2

Current total annual staff remuneration expenditure: $8185598.0

Current median monthly salary: $5001.0

---------------------------------------------------------

Please choose your option: 4

---------------------------------------------------------

This program has ended.

## Appendix - Q1 (b) Source Codes

## Chart 1 - Bar Chart

# Import library

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

# Import CSV file and load into dataframe

dataFrame = pd.read\_csv('GBA.csv')

# Create temporary data frame

temporary\_dataFrame = dataFrame[['Gender','MaritalStatus','WorkLifeBalance']]

# set width of bar

barWidth = 0.2

fig,axis = plt.subplots(figsize =(12, 8))

# To get the sum of work-Life balance for each categories

femaleDivorced = temporary\_dataFrame[(temporary\_dataFrame['Gender'] == 'Female') & (temporary\_dataFrame['MaritalStatus'] == 'Divorced')]

femaleMarried = temporary\_dataFrame[(temporary\_dataFrame['Gender'] == 'Female') & (temporary\_dataFrame['MaritalStatus'] == 'Married')]

femaleSingle = temporary\_dataFrame[(temporary\_dataFrame['Gender'] == 'Female') & (temporary\_dataFrame['MaritalStatus'] == 'Single')]

maleDivorced = temporary\_dataFrame[(temporary\_dataFrame['Gender'] == 'Male') & (temporary\_dataFrame['MaritalStatus'] == 'Divorced')]

maleMarried = temporary\_dataFrame[(temporary\_dataFrame['Gender'] == 'Male') & (temporary\_dataFrame['MaritalStatus'] == 'Married')]

maleSingle = temporary\_dataFrame[(temporary\_dataFrame['Gender'] == 'Male') & (temporary\_dataFrame['MaritalStatus'] == 'Single')]

# Set height of bar

fm = [sum(femaleDivorced['WorkLifeBalance']),sum(femaleMarried['WorkLifeBalance']), sum(femaleSingle['WorkLifeBalance'])]

m = [sum(maleDivorced['WorkLifeBalance']), sum(maleMarried['WorkLifeBalance']), sum(maleSingle['WorkLifeBalance'])]

# Set position of bar on X axis

bar1 = np.arange(len(fm))

bar2 = [x + barWidth for x in bar1 - 0.1]

bar3 = [x + barWidth for x in bar2]

# Make the plot

plt.bar(bar2, fm, color ='#F40C0C', width=barWidth - 0.02, edgecolor ='grey', label ='Female Employee')

plt.bar(bar3, m, color ='#14B3EC', width=barWidth-0.02, edgecolor ='grey', label ='Male Employee')

# Annotate, insert lable value for each bar

axis.bar\_label(axis.containers[0], label\_type='edge')

axis.bar\_label(axis.containers[1], label\_type='edge')

# Adding Xticks

plt.title("Does Marital Status affect Work-Life Balance?",fontsize = 20,fontname="Times New Roman", fontweight="bold")

plt.xlabel('Marital Status', fontweight ='bold', fontsize = 15)

plt.ylabel('Sum of Work-Life Balance', fontweight ='bold', fontsize = 15)

plt.xticks([r + barWidth for r in range(len(fm))], ['Divorced', 'Married', 'Single'])

# Show horizontal Grid line

plt.grid(axis='y')

# Show legend

plt.legend()

# Show Chart

plt.show()

## Table 1 - Summary Table for Chart 1

import pandas as pd

from tabulate import tabulate

# Load the data from a CSV file

data = pd.read\_csv('GBA.csv')

# Group the data by MaritalStatus

grouped\_data = data.groupby('MaritalStatus')

# Compute the summary statistics for each group

mean = df.groupby('MaritalStatus')['WorkLifeBalance'].mean()

median = df.groupby('MaritalStatus')['WorkLifeBalance'].median()

mode = df.groupby('MaritalStatus')['WorkLifeBalance'].apply(lambda x: x.mode()[0])

min1 = df.groupby('MaritalStatus')['WorkLifeBalance'].min()

max1 = df.groupby('MaritalStatus')['WorkLifeBalance'].max()

std = df.groupby('MaritalStatus')['WorkLifeBalance'].std()

# Put the summary statistics into a Pandas DataFrame

summary = pd.DataFrame({

    'Statistic': ['Mean', 'Median', 'Mode', 'Min', 'Max', 'Standard Deviation'],

    'Single': [mean.loc['Single'], median.loc['Single'], mode.loc['Single'], min1.loc['Single'], max1.loc['Single'], std.loc['Single']],

    'Married': [mean.loc['Married'], median.loc['Married'], mode.loc['Married'], min1.loc['Married'], max1.loc['Married'], std.loc['Married']],

    'Divorced': [mean.loc['Divorced'], median.loc['Divorced'], mode.loc['Divorced'], min1.loc['Divorced'], max1.loc['Divorced'], std.loc['Divorced']],

})

# Display the summary table using tabulate

print(tabulate(summary, headers=["Summary Statistics", "Single", "Married", "Divorced"], tablefmt='fancy\_grid'))

## Chart 2 - Box Plot

import pandas as pd

import matplotlib.pyplot as plt

# Load the data from a CSV file

df = pd.read\_csv('GBA.csv')

temp\_df = df[['Education','MonthlyPay']]

# Filter the temp\_df by education level (1: Others, 2: High School, 3: Tertiary, 4: Postgraduate)

df\_Others = temp\_df[temp\_df['Education'] == 1]

df\_HighSch = temp\_df[temp\_df['Education'] == 2]

df\_Tertiary = temp\_df[temp\_df['Education'] == 3]

df\_Postgraduate = temp\_df[temp\_df['Education'] == 4]

# Generating Box Plot

fig, ax = plt.subplots()

columns = [df\_Others['MonthlyPay'], df\_HighSch['MonthlyPay'], df\_Tertiary['MonthlyPay'], df\_Postgraduate['MonthlyPay']]

box = ax.boxplot(columns, patch\_artist = True,meanline=True, showmeans=True)

plt.xticks([1, 2, 3, 4], ["Others (1)", "High School (2)", "Tertiary (3)", "Postgraduate (4)"], rotation=5)

# Changing colours of the box plot

colors = ['green','orange','lightgrey','gold']

for patch, color in zip(box['boxes'], colors):

    patch.set\_facecolor(color)

# Set the chart title and axis labels

plt.title('Does Education level determine how much one earns?')

plt.xlabel('Education Level')

plt.ylabel('Monthly Pay ($)')

# Include horizontal Gride line

plt.grid(axis = 'y')

#Changing the limit for the x and y axis to be higher

plt.ylim(0, 30000)

# Display boxplot

plt.show()

## Table 2 - Summary Table for Box Plot

import pandas as pd

from tabulate import tabulate

# Load the data from a CSV file

df = pd.read\_csv('GBA.csv')

# Calculate summary statistics for each education level

mean\_vals = df.groupby('Education')['MonthlyPay'].mean()

median\_vals = df.groupby('Education')['MonthlyPay'].median()

mode\_vals = df.groupby('Education')['MonthlyPay'].apply(lambda x: x.mode()[0])

min\_vals = df.groupby('Education')['MonthlyPay'].min()

max\_vals = df.groupby('Education')['MonthlyPay'].max()

std\_vals = df.groupby('Education')['MonthlyPay'].std()

q1\_val = df.groupby('Education')['MonthlyPay'].quantile(0.25)

q3\_val = df.groupby('Education')['MonthlyPay'].quantile(0.75)

iqr\_val = q3\_val - q1\_val

# Put the summary statistics into a Pandas DataFrame

summary = pd.DataFrame({

    'Statistic': ['Mean', 'Median', 'Mode', 'Min', 'Max', 'Standard Deviation', 'IQR'],

    'Education Level 1': [mean\_vals[1], median\_vals[1], mode\_vals[1], min\_vals[1], max\_vals[1], std\_vals[1], iqr\_val[1]],

    'Education Level 2': [mean\_vals[2], median\_vals[2], mode\_vals[2], min\_vals[2], max\_vals[2], std\_vals[2], iqr\_val[2]],

    'Education Level 3': [mean\_vals[3], median\_vals[3], mode\_vals[3], min\_vals[3], max\_vals[3], std\_vals[3], iqr\_val[3]],

    'Education Level 4': [mean\_vals[4], median\_vals[4], mode\_vals[4], min\_vals[4], max\_vals[4], std\_vals[4], iqr\_val[4]],

})

# Display the summary table using tabulate

print(tabulate(summary, headers=["Summary Statistics", "Others (1)", "High School (2)", "Tertiary (3)", "Postgraduate (4)"], tablefmt='fancy\_grid'))

## Chart 3 - Scatter Plot

# Import pandas to use the dataframe in the library

import pandas as pd

import matplotlib.pyplot as plt

import numpy as np

from sklearn.linear\_model import LinearRegression

# Read the data from a CSV file

df = pd.read\_csv('GBA.csv')

# Extract the columns we need

monthly\_pay = df['MonthlyPay']

working\_exp = df['WorkingExperience']

# Create a scatter plot

plt.scatter(monthly\_pay, working\_exp, label='Data')

# Add a linear regression line to the scatter plot

x = np.array(monthly\_pay).reshape((-1, 1))

y = np.array(working\_exp)

model = LinearRegression().fit(x, y)

y\_pred = model.predict(x)

plt.plot(x, y\_pred, color='black', label='Linear Regression')

#Changing the limit for the x and y axis to be higher

plt.xlim(0, 30000)

plt.ylim(0, 45)

#Adding the descriptions for scatterplot

plt.title('Does longer working experience lead to higher monthly pay?')

plt.xlabel('Monthly Pay')

plt.ylabel('Working Experience')

plt.legend()

# Show the grid lines

plt.grid()

# Show the plot

plt.show()

## Table 3 - Regression Statistic Table for Chart 3

import pandas as pd

import numpy as np

from tabulate import tabulate

from scipy.stats import pearsonr

import statsmodels.api as sm

# Load the data from a CSV file

df = pd.read\_csv('GBA.csv')

# Group the data by WorkingExperience

grouped\_data = df.groupby('MonthlyPay')

# Calculate the correlation coefficient and p-value

r, p = pearsonr(df['MonthlyPay'], df['WorkLifeBalance'])

# Fit a linear regression model to the data

X = sm.add\_constant(df['WorkingExperience'])

model = sm.OLS(df['MonthlyPay'], X)

results = model.fit()

# Print the regression results summary

print(results.summary())

## Chart 4 - Histogram

# Importing required libraries

from matplotlib import pyplot as plt

import pandas as pd

import numpy as np

# Import CSV file and load into dataframe

dataFrame = pd.read\_csv('GBA.csv')

dataFrame[["MonthlyPay"]]

# A dataset of 1250 Employees

monthlyPay = dataFrame[["MonthlyPay"]]

# Creating Histogram

fig, axis = plt.subplots(figsize =(11, 6))

axis = plt.gca()

counts, \_ , patches = axis.hist(monthlyPay, bins = [0, 5000, 10000, 15000,20000,25000,30000], histtype= 'bar', align='mid', color='pink', edgecolor='red', linewidth=1.5)

# To remove the margin space

axis.margins(x=0)

# To display the total number of employees of each bin

for count, patch in zip(counts,patches):

    axis.annotate(str(int(count)), xy=(patch.get\_x(), patch.get\_height()), textcoords='offset points', xytext=(40, -11))

# Displaying the title, x-label and y-label

plt.title("Income Range of Employees",fontsize = 15,fontname="Times New Roman", fontweight="bold")

plt.xlabel("Income Range ($)")

plt.ylabel("Number of Employees")

# Function add a legend

plt.legend(["Number of Employees"], loc ="upper right",prop={'size': 8})

# Show Horizontal Grid line

plt.grid(axis='y')

# Show the histogram

plt.show()

## Table 4 - Frequency Table and Summary Statistic for Histogram

# Importing required libraries

from matplotlib import pyplot as plt

import pandas as pd

from tabulate import tabulate

# Import CSV file and load into dataframe

data\_frame = pd.read\_csv('GBA.csv')

# Number of Employees earning less than $4999 - Bin row 1

binRow1 = data\_frame[(data\_frame['MonthlyPay'] <= 4999)]

total\_binRow1 = len(binRow1)

# Number of Employees earning $5000 to $9999 - Bin row 2

binRow2 = data\_frame[(data\_frame['MonthlyPay'] > 4999) & (data\_frame['MonthlyPay'] <= 9999)]

total\_binRow2 = len(binRow2)

# Number of Employees earning $10000 to $14999 - Bin row 3

binRow3 = data\_frame[(data\_frame['MonthlyPay'] > 9999) & (data\_frame['MonthlyPay'] <= 14999)]

total\_binRow3 = len(binRow3)

# Number of Employees earning $15000 to $19999 - Bin row 4

binRow4 = data\_frame[(data\_frame['MonthlyPay'] > 14999) & (data\_frame['MonthlyPay'] <= 19999)]

total\_binRow4 = len(binRow4)

# Number of Employees earning $20000 to $24999 - Bin row 5

binRow5 = data\_frame[(data\_frame['MonthlyPay'] > 19999) & (data\_frame['MonthlyPay'] <= 24999)]

total\_binRow5 = len(binRow5)

# Number of Employees earning $25000 to $29999 - Bin row 6

binRow6 = data\_frame[(data\_frame['MonthlyPay'] > 24999) & (data\_frame['MonthlyPay'] <= 29999)]

total\_binRow6 = len(binRow6)

# Generating of data table for histogram

df\_Bins = pd.DataFrame({

    'Bins (Income Range ($))': ['0 to 4999','5000 to 9999','10000 to 14999','15000 to 19999','20000 to 24999','25000 to 29999'],

    'Number of Employees': [total\_binRow1,total\_binRow2,total\_binRow3,total\_binRow4,total\_binRow5,total\_binRow6],

})

print("Frequency Table for Histogram")

print("--------------------------------------")

print(tabulate(df\_Bins, headers=['Bins (Income Range ($))','Number of Employees'], tablefmt='fancy\_grid'))

# To generate analysis table

mean\_vals = data\_frame['MonthlyPay'].mean()

median\_vals = data\_frame['MonthlyPay'].median()

mode\_vals = df\_Bins["Number of Employees"].max()

min\_vals = data\_frame['MonthlyPay'].min()

max\_vals = data\_frame['MonthlyPay'].max()

if mode\_vals == 625:

    mode\_vals = "0 to 4999"

elif mode\_vals == 385:

    mode\_vals = "4999 to 9999"

elif mode\_vals == 128:

    mode\_vals = "15000 to 19999"

elif mode\_vals == 110:

    mode\_vals = "20000 to 24999"

else:

    mode\_vals = "25000 to 29999"

# Generating of analysing table for histogram

df\_SummaryStatistics = pd.DataFrame({

    'Analysis': ['Mean','Median','Mode','Min','Max'],

    'Income ($)': [mean\_vals,median\_vals,mode\_vals,min\_vals,max\_vals],

})

print("\nSummary Statistics for Histogram")

print("--------------------------------------")

print(tabulate(df\_SummaryStatistics, headers=['Summary Statistics','Income ($)'], tablefmt='fancy\_grid'))

## Appendix - Q1 (c) Source Codes

# Import pandas to use the dataframe in the library

import pandas as pd

# Import CSV file and load into dataframe

dataFrame = pd.read\_csv('GBA.csv')

# Functions to cater user selection of employee information of an organization

def selectionOfInfo(int\_option):

    if(int\_option == 1):

        # Start of Option 1:

        # Provide information of the current total staff strength, and a breakdown of staff strength

        # by business unit and gender.

        # To retrieve the total staff strength

        rows =  dataFrame.shape[0]

        # Extract selected Data Columns

        temporary\_dataFrame = dataFrame[["BusinessUnit", "Gender"]]

        # Display current total staff strength

        print(f"\nTotal Staff Strength: " + str(rows))

        # To get staff strength by business units (Business Development, HR, Product Development) and gender

        buinessDev\_Female = temporary\_dataFrame[(temporary\_dataFrame['Gender'] == 'Female') & (temporary\_dataFrame['BusinessUnit'] == 'Business Development')]

        hR\_Female = temporary\_dataFrame[(temporary\_dataFrame['Gender'] == 'Female') & (temporary\_dataFrame['BusinessUnit'] == 'HR')]

        productDev\_Female = temporary\_dataFrame[(temporary\_dataFrame['Gender'] == 'Female') & (temporary\_dataFrame['BusinessUnit'] == 'Product Development')]

        buinessDev\_Male = temporary\_dataFrame[(temporary\_dataFrame['Gender'] == 'Male') & (temporary\_dataFrame['BusinessUnit'] == 'Business Development')]

        hR\_Male = temporary\_dataFrame[(temporary\_dataFrame['Gender'] == 'Male') & (temporary\_dataFrame['BusinessUnit'] == 'HR')]

        productDev\_Male = temporary\_dataFrame[(temporary\_dataFrame['Gender'] == 'Male') & (temporary\_dataFrame['BusinessUnit'] == 'Product Development')]

        # Display a breakdown of staff strength by business unit and gender

        print("")

        print(f"\nBusiness Development:")

        print(f"\nTotal Number of Female Employees = {len(buinessDev\_Female)}")

        print(f"\nTotal Number of Male Employees = {len(buinessDev\_Male)}")

        print("")

        print(f"\nHR:")

        print(f"\nTotal Number of Female Employees = {len(hR\_Female)}")

        print(f"\nTotal Number of Male Employees = {len(hR\_Male)}")

        print("")

        print(f"\nProduct Development:")

        print(f"\nTotal Number of Female Employees = {len(productDev\_Female)}")

        print(f"\nTotal Number of Male Employees = {len(productDev\_Male)}")

        # End of Option 1:

    elif(int\_option == 2):

        # Start of Option 2:

        # Provide information of the current total annual staff remuneration expenditure, and

        # the current median monthly salary.

        # To retrieve the current total annual staff remuneration expenditure

        totalRemuneration = dataFrame['MonthlyPay'].sum()

        print(f"\nCurrent total annual staff remuneration expenditure: ${float(totalRemuneration)}")

        # To retrieve the current median monthly salary

        medianSalary = dataFrame['MonthlyPay'].median()

        print(f"\nCurrent median monthly salary: ${medianSalary}")

        # End of Option 2:

    elif(int\_option == 3):

        # End of Option 3:

        # Provide output information of the employees (EmpID), job role and age, ordered by descending age, with 35 or more years of service.

        # Selecting information such as the employees (EmpID), job role and age

        modifiedDataFrame = dataFrame[["EmpID","JobRole","Age","WorkingExperience"]]

        # Filtering data frame based on condition

        filtered\_df = modifiedDataFrame[modifiedDataFrame['WorkingExperience'] >= 35]

        # Sorting by column "Age"

        sortedDataFrame = filtered\_df.sort\_values(by=['Age'], ascending = False)

        # Display sortedDataFrame

        print("")

        print(sortedDataFrame[["EmpID","JobRole","Age","WorkingExperience"]])

        # End of Option 3:

# Start of Main program

print("\n \033[95mAn User Interactive Program")

print("\n Input an option 1 to 4 to view the employee information of an organization.")

print("\n 1: To view current total staff strength, and a breakdown of staff strength by business unit and gender")

print("\n 2: To view information of the current total annual staff remuneration expenditure, and the current median monthly salary")

print("\n 3: To view information of the employees (EmpID), job role and age, ordered by descending age, with 35 or more years of service")

print("\n 4: To exit the program")

print("\n---------------------------------------------------------")

bool\_toContinue = True

int\_option = 0

# Prompt user for user input if user chose option 4,the program will stop running

while bool\_toContinue == True and int\_option != 4:

    try:

        print("")

        int\_option = int(input(f"\nPlease choose your option: "))

        print("\n---------------------------------------------------------")

        # To end the program if user choose option 4

        if(int\_option == 4):

            bool\_toContinue = False

            print("This program has ended.")

            break

        elif (int\_option < 1 or int\_option > 4):

            int\_option = int(input(f"\nPlease re-input your option as you have entered an invalid option: "))

            selectionOfInfo(int\_option)

            print("\n---------------------------------------------------------")

        else:

            selectionOfInfo(int\_option)

    except ValueError:

            print(f"\nYou have entered an invalid option")

            int\_option = int(input(f"\nPlease re-input your option: "))

            print("\n---------------------------------------------------------")