

**ANL252**

**Python for Data Analytics**

# **Group-Based Assignment**

**January 2023**

**Submitted by:**

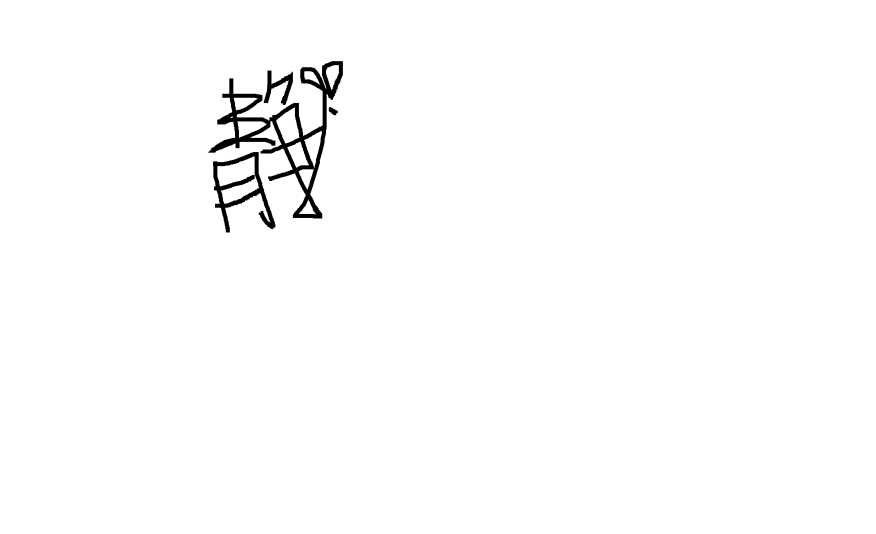
|  |  |
| --- | --- |
| **Name** | **PI No.** |
| **KOH REN YI, DOMINIC** | **E1911548** |
| **CHEN JING** | **K1981651** |
| **MOHAMED NURHAIKAL BIN MOHAMED SHARIFF** | **J2170095** |

**Tutorial Group: ­­­­­­­­­­­­ T05**

**Instructor’s Name: Dr. Munish Kumar**

**Submission Date: 19/02/2023**

*Declaration Page*

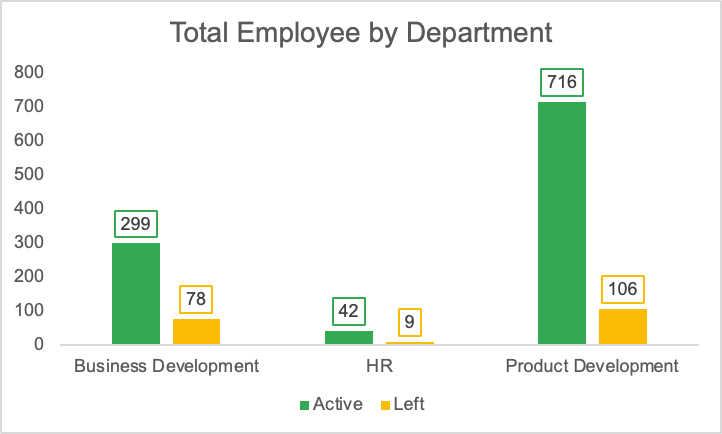
We, members of G**roup 4,** do hereby declare that we each contributed to this assignment and that we collectively agree to a shared grade.

|  |  |  |
| --- | --- | --- |
| **Name** | **Contribution** | **Signature** |
| Koh Ren Yi, Dominic (Team Lead) | I did 1c | DKRY |
| Chen Jing | I did 1a |  |
| Mohamed Nurhaikal Bin Mohamed Shariff | I did 1b |  |

**Question 1(a)**

| **Employment Status** | **Business Development** | **HR** | **Product Development** | **Grand Total** |
| --- | --- | --- | --- | --- |
| Active | 299 | 42 | 716 | 1057 |
| Left | 78 | 9 | 106 | 193 |
| Grand Total | 377 | 51 | 822 | 1250 |

***Table 1: Data for Employment Status by Department***

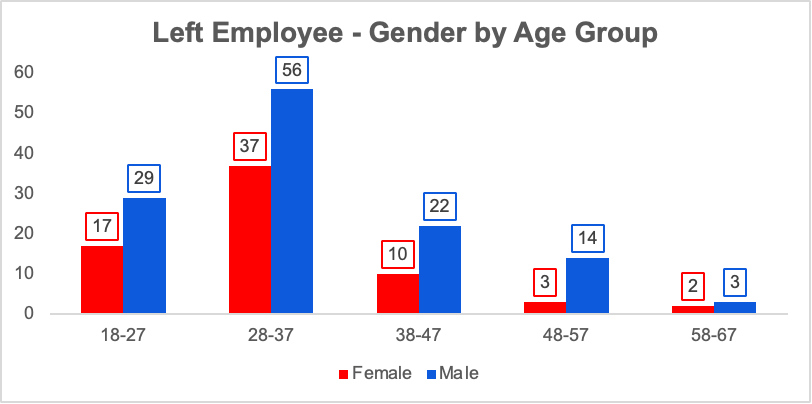


***Figure 1: Total Employment Status by Department***

Based on Figure 1, we can identify that most of the employees that left the company are from the Product Development department. It takes up about 55% of the total number of employees that left the company. As such this department’s management team could strategize to come up with measures to retain its staff.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Age Group** | **18-27** | **28-37** | **38-47** | **48-57** | **58-67** | **Grand Total** |
| Female | 17 | 37 | 10 | 3 | 2 | 69 |
| Male | 29 | 56 | 22 | 14 | 3 | 124 |

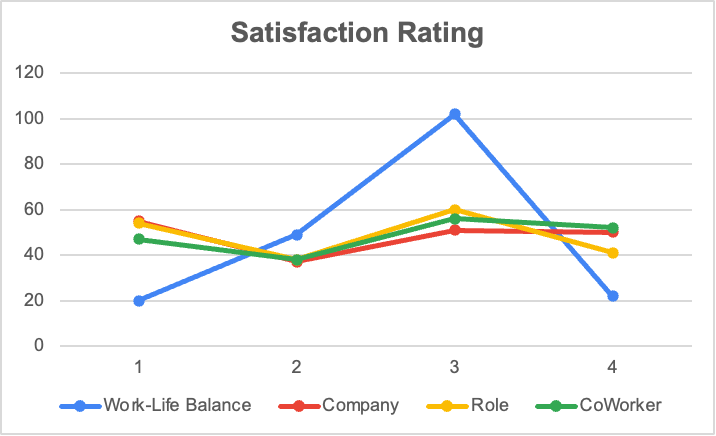
***Table 2: Data of Left Employee Gender by Age Group***

***Figure 2: Left Employee Gender by Age Group***

Based on Figure 2, we can identify that the majority of employees that left belong to the 20s to 30s age group and most of them are male. This highlights the potential issue that the company is no longer attractive to the younger generations and if this issue is left unattended, it could cause detrimental effects to the sustainability of the company. This is because the younger generation plays a significant role in sustaining the business in the long-run, especially when the company is moving on to a digital age.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Satisfaction Rating** | **Work-Life Balance** | **Company** | **Role** | **Co-Worker** |
| 1 | 20 | 55 | 54 | 47 |
| 2 | 49 | 37 | 38 | 38 |
| 3 | 102 | 51 | 60 | 56 |
| 4 | 22 | 50 | 41 | 52 |

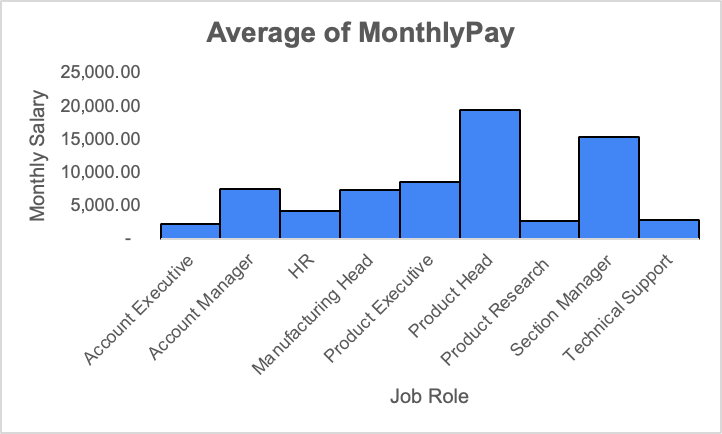
***Table 3: Data of Left Employee Satisfaction Rating***

***Figure 3: Left Employee Satisfaction Rating***

For employees that left the company, most of them are satisfied with work-life balance. However, with regards to satisfaction towards the Company, their Job Role and Co-workers, less than 60% gave a scale of 3 or 4 which represents higher satisfaction. As such, human resource management (HRM) of the company might look into further detail to see what caused the satisfaction towards the company, job role and co-worker. Regarding company and co-worker factors, they are largely affected by the company’s culture. Job roles on the other hand, could be due to a number of reasons, including the salary increment, the work exposure via training and business travels and the promotion rate.

|  |  |  |  |
| --- | --- | --- | --- |
| **Job Role** | **No. of Employee Left** | **Average of Monthly Pay** | **Average of Salary Increment %** |
| Account Executive | 28 | 2313.61 | 9.00 |
| Account Manager | 49 | 7505.84 | 7.47 |
| HR | 9 | 4258.00 | 7.44 |
| Manufacturing Head | 10 | 7365.50 | 6.90 |
| Product Executive | 7 | 8604.86 | 7.57 |
| Product Head | 2 | 19395.50 | 5.00 |
| Product Research | 35 | 2830.49 | 9.43 |
| Section Manager | 2 | 15336.50 | 7.00 |
| Technical Support | 51 | 2883.92 | 8.96 |

***Table 4: Data of Average Monthly Salary and Average Last Salary Increment of Left Employee by Job Role***

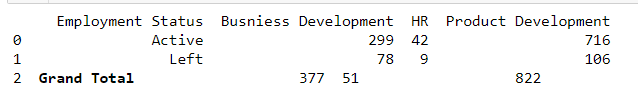


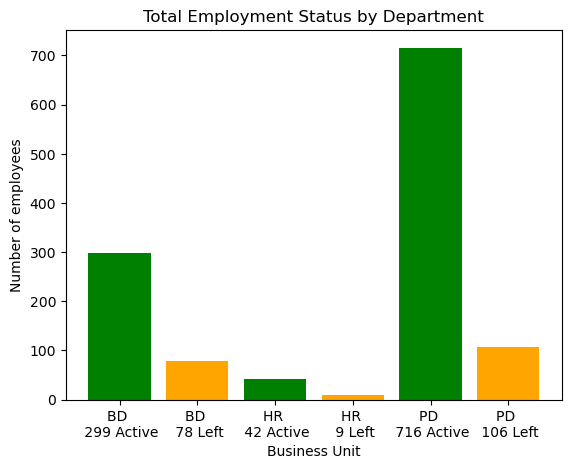
***Figure 4: Average Monthly Salary of Left Employee by Job Role***

Through the distribution of the job roles of employees that left the company, most are from the job roles of “Account Manager” and “Technical Support”. Amongst the two, “Technical Support” is one of the lower paying jobs within the firm. As such, the HRM department might be required to assess through further enquiry with the employees if salary was one of the reasons since the average for the last salary increment of all the job roles was less than 10%.

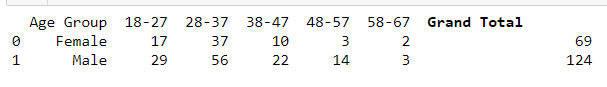
**Question 1(b)**

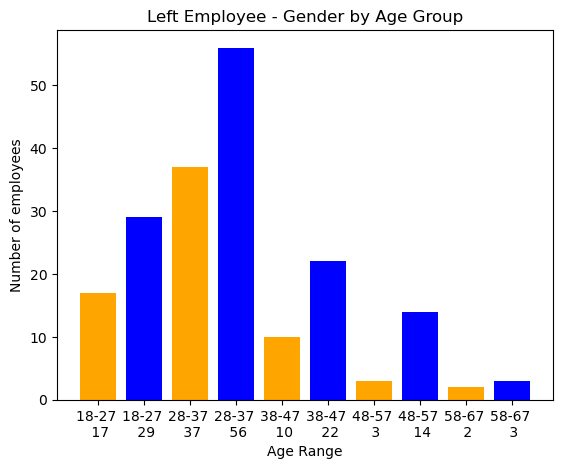
**Figure 1: Table and Chart Output**

****

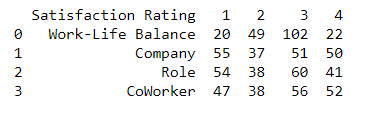
****

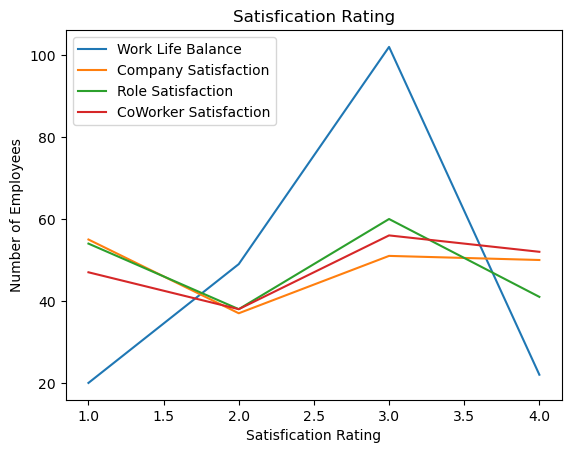
**Figure 2: Table and Chart Output**



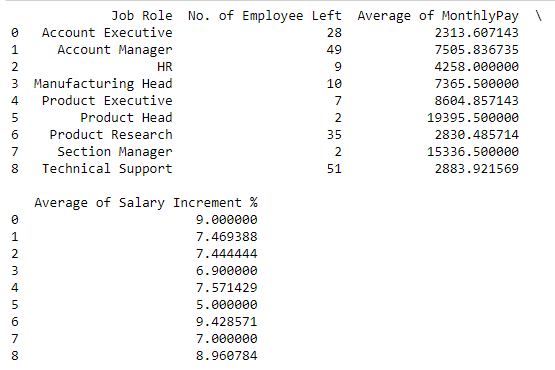


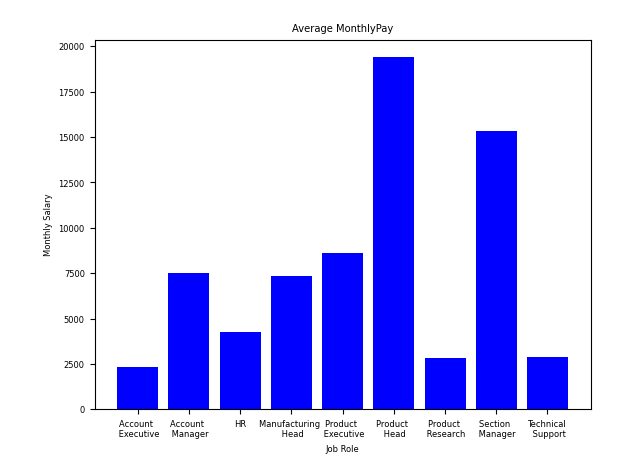
**Figure 3: Table and Chart Output**

****

****

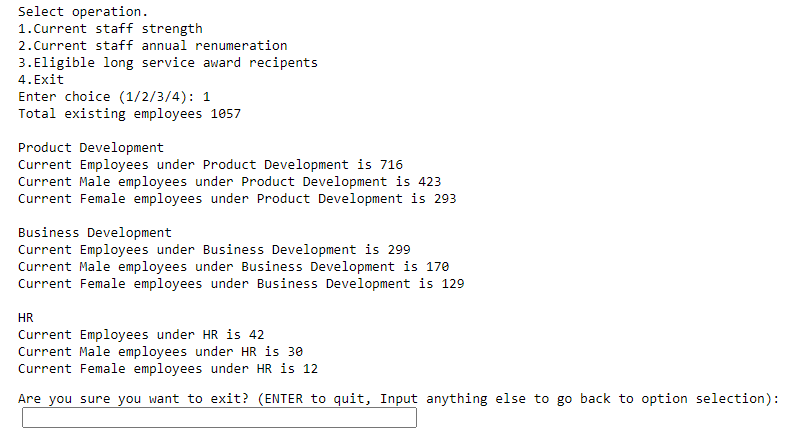
**Figure 4: Table and Chart Output**



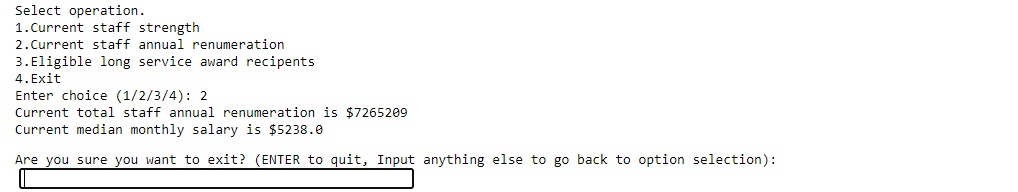
****

**Question 1(c)**

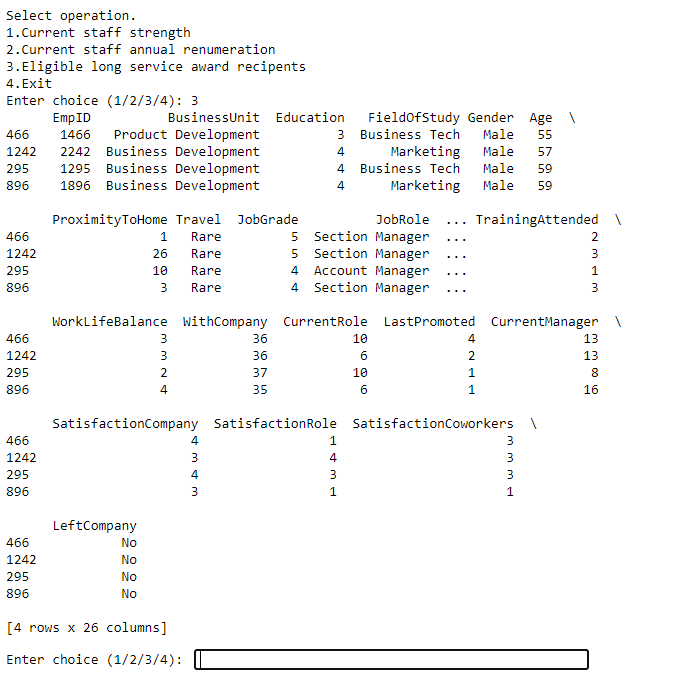
Option 1 Selection:

****

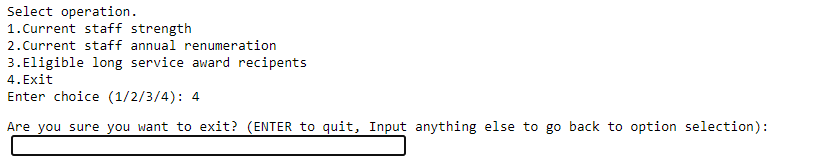
Option 2 Selection:

****

Option 3 Selection:

****

Option 4 Selection:

****

**Appendix 1 for Figure 1.**

# import pandas and matplotlib.pyplot

import pandas as pd

import matplotlib.pyplot as plt

# read the excel file into a pandas dataframe. Change (‘’) accordingly to where the GBA.csv link is at your computer.

dfemployee\_info = pd.read\_csv('GBA.csv')

# filter the dataframe by Business Development employees who have left the company

BusinessUnit\_BusinessDevelopment\_left = dfemployee\_info[

(dfemployee\_info['BusinessUnit'] == 'Business Development')

&

(dfemployee\_info['LeftCompany'] == 'Yes')

]

# filter the dataframe by Business Development employees who are active in the company

BusinessUnit\_BusinessDevelopment\_active = dfemployee\_info[

(dfemployee\_info['BusinessUnit'] == 'Business Development')

&

(dfemployee\_info['LeftCompany'] == 'No')

]

# filter the dataframe by HR employees who have left the company

BusinessUnit\_HR\_left = dfemployee\_info[

(dfemployee\_info['BusinessUnit'] == 'HR')

&

(dfemployee\_info['LeftCompany'] == 'Yes')

]

# filter the dataframe by HR employees who are active the company

BusinessUnit\_HR\_active = dfemployee\_info[

(dfemployee\_info['BusinessUnit'] == 'HR')

&

(dfemployee\_info['LeftCompany'] == 'No')

]

# filter the dataframe by Product Development employees who have left the company

BusinessUnit\_ProductDevelopment\_left = dfemployee\_info[

(dfemployee\_info['BusinessUnit'] == 'Product Development')

&

(dfemployee\_info['LeftCompany'] == 'Yes')

]

# filter the dataframe by Product Development employees who are active the company

BusinessUnit\_ProductDevelopment\_active = dfemployee\_info[

(dfemployee\_info['BusinessUnit'] == 'Product Development')

&

(dfemployee\_info['LeftCompany'] == 'No')

]

# count the number of employees who have left the company

num\_BusinessDevelopment\_left = len(BusinessUnit\_BusinessDevelopment\_left)

num\_BusinessDevelopment\_active = len(BusinessUnit\_BusinessDevelopment\_active)

num\_HR\_left = len(BusinessUnit\_HR\_left)

num\_HR\_active = len(BusinessUnit\_HR\_active)

num\_ProductDevelopment\_left = len(BusinessUnit\_ProductDevelopment\_left)

num\_ProductDevelopment\_active = len(BusinessUnit\_ProductDevelopment\_active)

# Define data for the bar graph

values = [

num\_BusinessDevelopment\_active,

num\_BusinessDevelopment\_left,

num\_HR\_active,

num\_HR\_left,

num\_ProductDevelopment\_active,

num\_ProductDevelopment\_left

]

labels = [

f'BD \n {num\_BusinessDevelopment\_active} Active',

f'BD \n {num\_BusinessDevelopment\_left} Left',

f'HR \n {num\_HR\_active} Active',

f'HR \n {num\_HR\_left} Left',

f'PD \n {num\_ProductDevelopment\_active} Active',

f'PD \n {num\_ProductDevelopment\_left} Left',

]

# create a table using a dictionary of lists

dfemployee\_info = {

'Employment Status': ['Active', 'Left', '\033[1mGrand Total\033[0m'],

'Busniess Development': [num\_BusinessDevelopment\_active, num\_BusinessDevelopment\_left,

(num\_BusinessDevelopment\_active + num\_BusinessDevelopment\_left)],

'HR': [num\_HR\_active, num\_HR\_left,

(num\_HR\_active + num\_HR\_left)],

'Product Development': [num\_ProductDevelopment\_active, num\_ProductDevelopment\_left,

(num\_ProductDevelopment\_active + num\_ProductDevelopment\_left)]

}

dftable = pd.DataFrame(dfemployee\_info)

print(dftable)

# Create bar graph

plt.bar(labels, values)

# create a bar graph with custom colors

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

plt.bar(labels, values, color=colors)

# Add labels and title

plt.xlabel('Business Unit')

plt.ylabel('Number of employees')

plt.title('Total Employment Status by Department')

# Show plot

plt.show()

**Appendix 2 for Figure 2.**

# import pandas and matplotlib.pyplot

import pandas as pd

import matplotlib.pyplot as plt

# read the excel file into a pandas dataframe. Change (‘’) accordingly to where the GBA.csv link is at your computer.

dfemployee\_info = pd.read\_csv('GBA.csv')

# filter the dataframe by female & male employees who have left the company between 18 to 27

Gender\_Female\_left\_grp1 = dfemployee\_info[

(dfemployee\_info['Gender'] == 'Female')

&

(dfemployee\_info['Age'] > 17)

&

(dfemployee\_info['Age'] < 28)

&

(dfemployee\_info['LeftCompany'] == 'Yes')

]

Gender\_Male\_left\_grp1 = dfemployee\_info[

(dfemployee\_info['Gender'] == 'Male')

&

(dfemployee\_info['Age'] > 17)

&

(dfemployee\_info['Age'] < 28)

&

(dfemployee\_info['LeftCompany'] == 'Yes')

]

# filter the dataframe by female & male employees who have left the company between 28 to 37

Gender\_Female\_left\_grp2 = dfemployee\_info[

(dfemployee\_info['Gender'] == 'Female')

&

(dfemployee\_info['Age'] > 27)

&

(dfemployee\_info['Age'] < 38)

&

(dfemployee\_info['LeftCompany'] == 'Yes')

]

Gender\_Male\_left\_grp2 = dfemployee\_info[

(dfemployee\_info['Gender'] == 'Male')

&

(dfemployee\_info['Age'] > 27)

&

(dfemployee\_info['Age'] < 38)

&

(dfemployee\_info['LeftCompany'] == 'Yes')

]

# filter the dataframe by female & male employees who have left the company between 38 to 47

Gender\_Female\_left\_grp3 = dfemployee\_info[

(dfemployee\_info['Gender'] == 'Female')

&

(dfemployee\_info['Age'] > 37)

&

(dfemployee\_info['Age'] < 48)

&

(dfemployee\_info['LeftCompany'] == 'Yes')

]

Gender\_Male\_left\_grp3 = dfemployee\_info[

(dfemployee\_info['Gender'] == 'Male')

&

(dfemployee\_info['Age'] > 37)

&

(dfemployee\_info['Age'] < 48)

&

(dfemployee\_info['LeftCompany'] == 'Yes')

]

# filter the dataframe by female & male employees who have left the company between 48 to 57

Gender\_Female\_left\_grp4 = dfemployee\_info[

(dfemployee\_info['Gender'] == 'Female')

&

(dfemployee\_info['Age'] > 47)

&

(dfemployee\_info['Age'] < 58)

&

(dfemployee\_info['LeftCompany'] == 'Yes')

]

Gender\_Male\_left\_grp4 = dfemployee\_info[

(dfemployee\_info['Gender'] == 'Male')

&

(dfemployee\_info['Age'] > 47)

&

(dfemployee\_info['Age'] < 58)

&

(dfemployee\_info['LeftCompany'] == 'Yes')

]

# filter the dataframe by female & male employees who have left the company between 58 to 67

Gender\_Female\_left\_grp5 = dfemployee\_info[

(dfemployee\_info['Gender'] == 'Female')

&

(dfemployee\_info['Age'] > 57)

&

(dfemployee\_info['Age'] < 68)

&

(dfemployee\_info['LeftCompany'] == 'Yes')

]

Gender\_Male\_left\_grp5 = dfemployee\_info[

(dfemployee\_info['Gender'] == 'Male')

&

(dfemployee\_info['Age'] > 57)

&

(dfemployee\_info['Age'] < 68)

&

(dfemployee\_info['LeftCompany'] == 'Yes')

]

# count the number of employees who have left the company

num\_Gender\_Female\_left\_grp1 = len(Gender\_Female\_left\_grp1)

num\_Gender\_Male\_left\_grp1 = len(Gender\_Male\_left\_grp1)

num\_Gender\_Female\_left\_grp2 = len(Gender\_Female\_left\_grp2)

num\_Gender\_Male\_left\_grp2 = len(Gender\_Male\_left\_grp2)

num\_Gender\_Female\_left\_grp3 = len(Gender\_Female\_left\_grp3)

num\_Gender\_Male\_left\_grp3 = len(Gender\_Male\_left\_grp3)

num\_Gender\_Female\_left\_grp4 = len(Gender\_Female\_left\_grp4)

num\_Gender\_Male\_left\_grp4 = len(Gender\_Male\_left\_grp4)

num\_Gender\_Female\_left\_grp5 = len(Gender\_Female\_left\_grp5)

num\_Gender\_Male\_left\_grp5 = len(Gender\_Male\_left\_grp5)

# create a table using a dictionary of lists

dfemployee\_info = {

'Age Group': ['Female', 'Male'],

'18-27': [num\_Gender\_Female\_left\_grp1, num\_Gender\_Male\_left\_grp1],

'28-37': [num\_Gender\_Female\_left\_grp2, num\_Gender\_Male\_left\_grp2],

'38-47': [num\_Gender\_Female\_left\_grp3, num\_Gender\_Male\_left\_grp3],

'48-57': [num\_Gender\_Female\_left\_grp4, num\_Gender\_Male\_left\_grp4],

'58-67': [num\_Gender\_Female\_left\_grp5, num\_Gender\_Male\_left\_grp5],

'\033[1mGrand Total\033[0m': [(num\_Gender\_Female\_left\_grp1 +

num\_Gender\_Female\_left\_grp2 +

num\_Gender\_Female\_left\_grp3 +

num\_Gender\_Female\_left\_grp4 +

num\_Gender\_Female\_left\_grp5),

(

num\_Gender\_Male\_left\_grp1 +

num\_Gender\_Male\_left\_grp2 +

num\_Gender\_Male\_left\_grp3 +

num\_Gender\_Male\_left\_grp4 +

num\_Gender\_Male\_left\_grp5

)

]

}

dftable = pd.DataFrame(dfemployee\_info)

print(dftable)

# Define data for the bar graph

values = [

num\_Gender\_Female\_left\_grp1,

num\_Gender\_Male\_left\_grp1,

num\_Gender\_Female\_left\_grp2,

num\_Gender\_Male\_left\_grp2,

num\_Gender\_Female\_left\_grp3,

num\_Gender\_Male\_left\_grp3,

num\_Gender\_Female\_left\_grp4,

num\_Gender\_Male\_left\_grp4,

num\_Gender\_Female\_left\_grp5,

num\_Gender\_Male\_left\_grp5

]

labels = [

f'18-27 \n {num\_Gender\_Female\_left\_grp1}',

f'18-27 \n {num\_Gender\_Male\_left\_grp1}',

f'28-37 \n {num\_Gender\_Female\_left\_grp2}',

f'28-37 \n {num\_Gender\_Male\_left\_grp2}',

f'38-47 \n {num\_Gender\_Female\_left\_grp3}',

f'38-47 \n {num\_Gender\_Male\_left\_grp3}',

f'48-57 \n {num\_Gender\_Female\_left\_grp4}',

f'48-57 \n {num\_Gender\_Male\_left\_grp4}',

f'58-67 \n {num\_Gender\_Female\_left\_grp5}',

f'58-67 \n {num\_Gender\_Male\_left\_grp5}',

]

# Create bar graph

plt.bar(labels, values)

# create a bar graph with custom colors

colors = ['orange', 'blue', 'orange', 'blue', 'orange', 'blue', 'orange', 'blue']

plt.bar(labels, values, color=colors)

# Add labels and title

plt.xlabel('Age Range')

plt.ylabel('Number of employees')

plt.title('Left Employee - Gender by Age Group')

# Show plot

plt.show()

**Appendix 3 for Figure 3.**

# import pandas and matplotlib.pyplot

import pandas as pd

import matplotlib.pyplot as plt

# read the excel file into a pandas dataframe. Change (‘’) accordingly to where the GBA.csv link is at your computer.

dfemployee\_info = pd.read\_csv('GBA.csv')

# filter the dataframe by satisfaction rating of employees on work life balance

WorkLifeBalance\_1 = dfemployee\_info[

(dfemployee\_info['WorkLifeBalance'] == 1)

&

(dfemployee\_info['LeftCompany'] == 'Yes')

]

WorkLifeBalance\_2 = dfemployee\_info[

(dfemployee\_info['WorkLifeBalance'] == 2)

&

(dfemployee\_info['LeftCompany'] == 'Yes')

]

WorkLifeBalance\_3 = dfemployee\_info[

(dfemployee\_info['WorkLifeBalance'] == 3)

&

(dfemployee\_info['LeftCompany'] == 'Yes')

]

WorkLifeBalance\_4 = dfemployee\_info[

(dfemployee\_info['WorkLifeBalance'] == 4)

&

(dfemployee\_info['LeftCompany'] == 'Yes')

]

# filter the dataframe by satisfaction rating of employees on company satisfaction

SatisfactionCompany\_1 = dfemployee\_info[

(dfemployee\_info['SatisfactionCompany'] == 1)

&

(dfemployee\_info['LeftCompany'] == 'Yes')

]

SatisfactionCompany\_2 = dfemployee\_info[

(dfemployee\_info['SatisfactionCompany'] == 2)

&

(dfemployee\_info['LeftCompany'] == 'Yes')

]

SatisfactionCompany\_3 = dfemployee\_info[

(dfemployee\_info['SatisfactionCompany'] == 3)

&

(dfemployee\_info['LeftCompany'] == 'Yes')

]

SatisfactionCompany\_4 = dfemployee\_info[

(dfemployee\_info['SatisfactionCompany'] == 4)

&

(dfemployee\_info['LeftCompany'] == 'Yes')

]

# filter the dataframe by satisfaction rating of employees on company role satisfaction

SatisfactionRole\_1 = dfemployee\_info[

(dfemployee\_info['SatisfactionRole'] == 1)

&

(dfemployee\_info['LeftCompany'] == 'Yes')

]

SatisfactionRole\_2 = dfemployee\_info[

(dfemployee\_info['SatisfactionRole'] == 2)

&

(dfemployee\_info['LeftCompany'] == 'Yes')

]

SatisfactionRole\_3 = dfemployee\_info[

(dfemployee\_info['SatisfactionRole'] == 3)

&

(dfemployee\_info['LeftCompany'] == 'Yes')

]

SatisfactionRole\_4 = dfemployee\_info[

(dfemployee\_info['SatisfactionRole'] == 4)

&

(dfemployee\_info['LeftCompany'] == 'Yes')

]

# filter the dataframe by satisfaction rating of employees on company coworkers satisfaction

SatisfactionCoworkers\_1 = dfemployee\_info[

(dfemployee\_info['SatisfactionCoworkers'] == 1)

&

(dfemployee\_info['LeftCompany'] == 'Yes')

]

SatisfactionCoworkers\_2 = dfemployee\_info[

(dfemployee\_info['SatisfactionCoworkers'] == 2)

&

(dfemployee\_info['LeftCompany'] == 'Yes')

]

SatisfactionCoworkers\_3 = dfemployee\_info[

(dfemployee\_info['SatisfactionCoworkers'] == 3)

&

(dfemployee\_info['LeftCompany'] == 'Yes')

]

SatisfactionCoworkers\_4 = dfemployee\_info[

(dfemployee\_info['SatisfactionCoworkers'] == 4)

&

(dfemployee\_info['LeftCompany'] == 'Yes')

]

# count the ratings of the number of employees who have left the company

num\_WorkLifeBalance\_1 = len(WorkLifeBalance\_1)

num\_WorkLifeBalance\_2 = len(WorkLifeBalance\_2)

num\_WorkLifeBalance\_3 = len(WorkLifeBalance\_3)

num\_WorkLifeBalance\_4 = len(WorkLifeBalance\_4)

num\_SatisfactionCompany\_1 = len(SatisfactionCompany\_1)

num\_SatisfactionCompany\_2 = len(SatisfactionCompany\_2)

num\_SatisfactionCompany\_3 = len(SatisfactionCompany\_3)

num\_SatisfactionCompany\_4 = len(SatisfactionCompany\_4)

num\_SatisfactionRole\_1 = len(SatisfactionRole\_1)

num\_SatisfactionRole\_2 = len(SatisfactionRole\_2)

num\_SatisfactionRole\_3 = len(SatisfactionRole\_3)

num\_SatisfactionRole\_4 = len(SatisfactionRole\_4)

num\_SatisfactionCoworkers\_1 = len(SatisfactionCoworkers\_1)

num\_SatisfactionCoworkers\_2 = len(SatisfactionCoworkers\_2)

num\_SatisfactionCoworkers\_3 = len(SatisfactionCoworkers\_3)

num\_SatisfactionCoworkers\_4 = len(SatisfactionCoworkers\_4)

# create a table using a dictionary of lists

dfemployee\_info = {

'Satisfaction Rating': ['Work-Life Balance', 'Company', 'Role', 'CoWorker'],

'1': [num\_WorkLifeBalance\_1, num\_SatisfactionCompany\_1, num\_SatisfactionRole\_1, num\_SatisfactionCoworkers\_1],

'2': [num\_WorkLifeBalance\_2, num\_SatisfactionCompany\_2, num\_SatisfactionRole\_2, num\_SatisfactionCoworkers\_2],

'3': [num\_WorkLifeBalance\_3, num\_SatisfactionCompany\_3, num\_SatisfactionRole\_3, num\_SatisfactionCoworkers\_3],

'4': [num\_WorkLifeBalance\_4, num\_SatisfactionCompany\_4, num\_SatisfactionRole\_4, num\_SatisfactionCoworkers\_4],

}

dftable = pd.DataFrame(dfemployee\_info)

print(dftable)

import seaborn as sns

import pandas as pd

# Load sample data

data = pd.DataFrame({

'x': [1, 2, 3, 4],

'y1': [num\_WorkLifeBalance\_1, num\_WorkLifeBalance\_2, num\_WorkLifeBalance\_3, num\_WorkLifeBalance\_4],

'y2': [num\_SatisfactionCompany\_1, num\_SatisfactionCompany\_2, num\_SatisfactionCompany\_3, num\_SatisfactionCompany\_4],

'y3': [num\_SatisfactionRole\_1, num\_SatisfactionRole\_2, num\_SatisfactionRole\_3, num\_SatisfactionRole\_4],

'y4': [num\_SatisfactionCoworkers\_1, num\_SatisfactionCoworkers\_2, num\_SatisfactionCoworkers\_3, num\_SatisfactionCoworkers\_4]

})

# Plot two lines on the same graph

sns.lineplot(data=data, x='x', y='y1', label='Work Life Balance')

sns.lineplot(data=data, x='x', y='y2', label='Company Satisfaction')

sns.lineplot(data=data, x='x', y='y3', label='Role Satisfaction')

sns.lineplot(data=data, x='x', y='y4', label='CoWorker Satisfaction')

# Labelling the graph

plt.xlabel('Satisfication Rating')

plt.ylabel('Number of Employees')

plt.title('Satisfication Rating')

# Show the combined plot

plt.show()

**Appendix 4 for Figure 4.**

# import pandas and matplotlib.pyplot

import pandas as pd

import matplotlib.pyplot as plt

#Change font size.

plt.rcParams.update({'font.size': 6})

# read the excel file into a pandas dataframe. Change (‘’) accordingly to where the GBA.csv link is at your computer.

dfemployee\_info = pd.read\_csv('C:/Users/User/Desktop/GBA.csv')

# filter the dataframe by satisfaction rating of employees on work life balance

Account\_Executive\_left = dfemployee\_info[

(dfemployee\_info['JobRole'] == 'Account Executive')

&

(dfemployee\_info['LeftCompany'] == 'Yes')

]

Account\_Manager\_left = dfemployee\_info[

(dfemployee\_info['JobRole'] == 'Account Manager')

&

(dfemployee\_info['LeftCompany'] == 'Yes')

]

HR\_left = dfemployee\_info[

(dfemployee\_info['JobRole'] == 'HR')

&

(dfemployee\_info['LeftCompany'] == 'Yes')

]

Manufacturing\_Head\_left = dfemployee\_info[

(dfemployee\_info['JobRole'] == 'Manufacturing Head')

&

(dfemployee\_info['LeftCompany'] == 'Yes')

]

Product\_Executive\_left = dfemployee\_info[

(dfemployee\_info['JobRole'] == 'Product Executive')

&

(dfemployee\_info['LeftCompany'] == 'Yes')

]

Product\_Head\_left = dfemployee\_info[

(dfemployee\_info['JobRole'] == 'Product Head')

&

(dfemployee\_info['LeftCompany'] == 'Yes')

]

Product\_Research\_left = dfemployee\_info[

(dfemployee\_info['JobRole'] == 'Product Research ')

&

(dfemployee\_info['LeftCompany'] == 'Yes')

]

Section\_Manager\_left = dfemployee\_info[

(dfemployee\_info['JobRole'] == 'Section Manager')

&

(dfemployee\_info['LeftCompany'] == 'Yes')

]

Technical\_Support\_left = dfemployee\_info[

(dfemployee\_info['JobRole'] == 'Technical Support')

&

(dfemployee\_info['LeftCompany'] == 'Yes')

]

# count the number of job role of employees who have left the company

num\_Account\_Executive\_left = len(Account\_Executive\_left)

num\_Account\_Manager\_left = len(Account\_Manager\_left)

num\_HR\_left = len(HR\_left)

num\_Manufacturing\_Head\_left = len(Manufacturing\_Head\_left)

num\_Product\_Executive\_left = len(Product\_Executive\_left)

num\_Product\_Head\_left = len(Product\_Head\_left)

num\_Product\_Research\_left = len(Product\_Research\_left)

num\_Section\_Manager\_left = len(Section\_Manager\_left)

num\_Technical\_Support\_left = len(Technical\_Support\_left)

# Calculate the average of the job salary of each employee that left the company

Mean\_MonthlyPay\_Account\_Executive\_left = Account\_Executive\_left['MonthlyPay'].mean()

Mean\_MonthlyPay\_Account\_Manager\_left = Account\_Manager\_left['MonthlyPay'].mean()

Mean\_MonthlyPay\_HR\_left = HR\_left['MonthlyPay'].mean()

Mean\_MonthlyPay\_Manufacturing\_Head\_left = Manufacturing\_Head\_left['MonthlyPay'].mean()

Mean\_MonthlyPay\_Product\_Executive\_left = Product\_Executive\_left['MonthlyPay'].mean()

Mean\_MonthlyPay\_Product\_Head\_left = Product\_Head\_left['MonthlyPay'].mean()

Mean\_MonthlyPay\_Product\_Research\_left = Product\_Research\_left['MonthlyPay'].mean()

Mean\_MonthlyPay\_Section\_Manager\_left = Section\_Manager\_left['MonthlyPay'].mean()

Mean\_MonthlyPay\_Technical\_Support\_left = Technical\_Support\_left['MonthlyPay'].mean()

# Calculate the average of the salary increment of each employee that left the company

Mean\_SalaryIncrement\_Account\_Executive\_left = Account\_Executive\_left['SalaryIncrement'].mean()

Mean\_SalaryIncrement\_Account\_Manager\_left = Account\_Manager\_left['SalaryIncrement'].mean()

Mean\_SalaryIncrement\_HR\_left = HR\_left['SalaryIncrement'].mean()

Mean\_SalaryIncrement\_Manufacturing\_Head\_left = Manufacturing\_Head\_left['SalaryIncrement'].mean()

Mean\_SalaryIncrement\_Product\_Executive\_left = Product\_Executive\_left['SalaryIncrement'].mean()

Mean\_SalaryIncrement\_Product\_Head\_left = Product\_Head\_left['SalaryIncrement'].mean()

Mean\_SalaryIncrement\_Product\_Research\_left = Product\_Research\_left['SalaryIncrement'].mean()

Mean\_SalaryIncrement\_Section\_Manager\_left = Section\_Manager\_left['SalaryIncrement'].mean()

Mean\_SalaryIncrement\_Technical\_Support\_left = Technical\_Support\_left['SalaryIncrement'].mean()

# Create a table using a dictionary of lists

dfemployee\_info = {

'Job Role': [

'Account Executive',

'Account Manager',

'HR',

'Manufacturing Head',

'Product Executive',

'Product Head',

'Product Research',

'Section Manager',

'Technical Support'

],

'No. of Employee Left': [num\_Account\_Executive\_left,

num\_Account\_Manager\_left,

num\_HR\_left,

num\_Manufacturing\_Head\_left,

num\_Product\_Executive\_left,

num\_Product\_Head\_left,

num\_Product\_Research\_left,

num\_Section\_Manager\_left,

num\_Technical\_Support\_left

],

'Average of MonthlyPay': [Mean\_MonthlyPay\_Account\_Executive\_left,

Mean\_MonthlyPay\_Account\_Manager\_left,

Mean\_MonthlyPay\_HR\_left,

Mean\_MonthlyPay\_Manufacturing\_Head\_left,

Mean\_MonthlyPay\_Product\_Executive\_left,

Mean\_MonthlyPay\_Product\_Head\_left,

Mean\_MonthlyPay\_Product\_Research\_left,

Mean\_MonthlyPay\_Section\_Manager\_left,

Mean\_MonthlyPay\_Technical\_Support\_left

],

'Average of Salary Increment %': [Mean\_SalaryIncrement\_Account\_Executive\_left,

Mean\_SalaryIncrement\_Account\_Manager\_left,

Mean\_SalaryIncrement\_HR\_left,

Mean\_SalaryIncrement\_Manufacturing\_Head\_left,

Mean\_SalaryIncrement\_Product\_Executive\_left,

Mean\_SalaryIncrement\_Product\_Head\_left,

Mean\_SalaryIncrement\_Product\_Research\_left,

Mean\_SalaryIncrement\_Section\_Manager\_left,

Mean\_SalaryIncrement\_Technical\_Support\_left

],

}

dftable = pd.DataFrame(dfemployee\_info)

print(dftable)

# Define data for the bar graph

values = [

Mean\_MonthlyPay\_Account\_Executive\_left,

Mean\_MonthlyPay\_Account\_Manager\_left,

Mean\_MonthlyPay\_HR\_left,

Mean\_MonthlyPay\_Manufacturing\_Head\_left,

Mean\_MonthlyPay\_Product\_Executive\_left,

Mean\_MonthlyPay\_Product\_Head\_left,

Mean\_MonthlyPay\_Product\_Research\_left,

Mean\_MonthlyPay\_Section\_Manager\_left,

Mean\_MonthlyPay\_Technical\_Support\_left

]

labels = [

'Account \n Executive',

'Account \n Manager',

'HR',

'Manufacturing \n Head',

'Product \n Executive',

'Product \n Head',

'Product \n Research',

'Section \n Manager',

'Technical \n Support'

]

# Create bar graph

plt.bar(labels, values)

# create a bar graph with custom colors

colors = ['blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue', 'blue']

plt.bar(labels, values, color=colors)

# Add labels and title

plt.xlabel('Job Role')

plt.ylabel('Monthly Salary')

plt.title('Average MonthlyPay')

# Show plot

plt.show()

**Appendix 5 for Question 1(c)**

#Import library "pandas" for data management. "as pd" is just giving pandas a name, it can be anything.

import pandas as pd

#assigning df as the name to read GBA.csv file. Change (‘’) accordingly to where the GBA.csv link is at your computer.

df = pd.read\_csv('C:/Users/User/Desktop/GBA.csv')

#Putting 1,2,3,4 as a condition to provide error message if other than these inputs are being input, will be invalid.

in1 = ["1", "2", "3", "4"]

#Interface layout (4 options)

print("Select operation.")

print("1.Current staff strength")

print("2.Current staff annual renumeration")

print("3.Eligible long service award recipents")

print("4.Exit")

while True:

#Allowing users to input their choice.

choice = input ("Enter choice (1/2/3/4): ")

#Based on the input, the following if,elif,else conditions will take place.

if choice in ('1', '2', '3', '4'):

if choice=='1':

#Display all current working staff (1057 count)

#Product Development (716)

#Male: 423, Female: 293

#Filter out people that left = No, then filter the Business Unit

currentemployees=df.loc[df['LeftCompany']=="No", 'LeftCompany'].count()

currentpd\_overall=len(df[(df["LeftCompany"]=="No") &

(df["BusinessUnit"]=="Product Development")])

#Filter out male employees.

malepd=len(df[(df["LeftCompany"]=="No") &

(df["BusinessUnit"]=="Product Development") &

(df["Gender"]=="Male")])

#Filter out female employees.

femalepd=len(df[(df["LeftCompany"]=="No") &

(df["BusinessUnit"]=="Product Development") &

(df["Gender"]=="Female")])

print (f"Total existing employees {currentemployees}\n")

print (f"Product Development")

print (f"Current Employees under Product Development is {currentpd\_overall}")

print (f"Current Male employees under Product Development is {malepd}")

print (f"Current Female employees under Product Development is {femalepd}\n")

#Business development (299 count)

#Male: 170, Female: 129

#Filter out people that left = No, then filter the Business Unit

currentbd\_overall=len(df[(df["LeftCompany"]=="No") &

(df["BusinessUnit"]=="Business Development")])

#Filter out male employees.

malebd=len(df[(df["LeftCompany"]=="No") &

(df["BusinessUnit"]=="Business Development") &

(df["Gender"]=="Male")])

#Filter out female employees.

femalebd=len(df[(df["LeftCompany"]=="No") &

(df["BusinessUnit"]=="Business Development") &

(df["Gender"]=="Female")])

print (f"Business Development")

print (f"Current Employees under Business Development is {currentbd\_overall}")

print (f"Current Male employees under Business Development is {malebd}")

print (f"Current Female employees under Business Development is {femalebd}\n")

#HR (42 count)

#Male: 30 pax, Female: 12 pax

#Filter out people that left = No, then filter the Business Unit

currenthr\_overall=len(df[(df["LeftCompany"]=="No") &

(df["BusinessUnit"]=="HR")])

#Filter out male employees.

malehr=len(df[(df["LeftCompany"]=="No") &

(df["BusinessUnit"]=="HR") &

(df["Gender"]=="Male")])

#Filter out female employees.

femalehr=len(df[(df["LeftCompany"]=="No") &

(df["BusinessUnit"]=="HR") &

(df["Gender"]=="Female")])

print (f"HR")

print (f"Current Employees under HR is {currenthr\_overall}")

print (f"Current Male employees under HR is {malehr}")

print (f"Current Female employees under HR is {femalehr}")

if choice=='2':

#Filter out employees that left, then sum the total salary for existing workers. Amount should be $7,265,209

TotalSalary=df.loc[df['LeftCompany'] == "No", 'MonthlyPay'].sum()

print (f"Current total staff annual renumeration is ${TotalSalary}")

#Filter out employees that left, then median the total salary for existing workers. Amount should be $5,238

Medianz=df.loc[df['LeftCompany'] == "No", 'MonthlyPay'].median()

print (f"Current median monthly salary is ${Medianz}")

if choice=='3':

#Filter out employees that left.

#Filter out years of service =>35 years under 'WithCompany'

#Sort the age

bonus=df.loc[(df['WithCompany'] >= 35) & (df['LeftCompany']=="No")].sort\_values(by=['Age'])

print (bonus)

else:

exit\_confirm = input ("Are you sure you want to exit? (ENTER to quit, Input anything else to go back to option selection): ")

if exit\_confirm == "":

break

elif choice != 'in1':

print("Invalid Input. Please input one of the choices.")