

**ANL252**

**Python for Data Analytics**

**Group-based Assignment**

**JANUARY 2023 Presentation**

**Submitted by:**

|  |  |
| --- | --- |
| **Name** | **PI No.** |
| Brandan Chia Jian Hao | Y2172066 |
| Ang Ding Yi | B2211211 |
| Janelle Lim Sock Hui | E2181690 |
| Leona Inez Tan Sheng Guat | J2082686 |

Tutorial Group: T05

Instructor’s Name: Prof. Munish Kumar

Submission Date: 19 Feb 2023

Declaration Page

We, members of group 10, do hereby declare that we each contributed to this assignment and that we collectively agree to a shared grade.

|  |  |  |
| --- | --- | --- |
| Name | Contribution | Signature |
| Ang Ding Yi (Team Lead) | I did 1a | Ang Ding Yi |
| Leona Inez Tan Sheng Guat | I did 1b | Leona Inez Tan |
| Janelle Lim Sock Hui | I did 1b | Janelle |
| Brandan Chia Jian Hao | I did 1c | Brandan |

# 

**Table of Contents**

[**Question 1a**](#_15nj88cdkmfv) **3**

[Observation 1](#_mgmh5zeo3hfh) 3

[Chart 1a (created using Excel)](#_ipwtx9pnwhsd) 3

[Observation 2](#_h2dcmw56wly9) 4

[Chart 2a (created using Excel)](#_jtw2t0e4yu9p) 4

[Table 2a (created using Excel)](#_awzgfdrbsg99) 4

[Observation 3](#_pktlg6rn06i5) 5

[Chart 3a (created using Excel)](#_crx1brazwz25) 5

[Table 3a.1 (created using Excel)](#_6cyqzg84s28y) 5

[Table 3a.2 (created using Excel)](#_fniwvptuz05y) 6

[Observation 4](#_fe4euy6zftc2) 6

[Chart 4a (created using Excel)](#_2mmfymlz8xxb) 6

[Table 4a.1 (created using Excel)](#_5pk86ayt4c5a) 7

[Table 4a.2 (created using Excel)](#_nncdwuygv4am) 7

[**Question 1b**](#_phbxbune7r3p) **8**

[Chart 1b (created using Python)](#_oljahykk4ofe) 8

[Table 1b (created using Python)](#_tlckgylfqxf4) 8

[Chart 2b (created using Python)](#_9qast1kh7obv) 9

[Table 2b (created using Python)](#_affageymr9i4) 9

[Chart 3b (created using Python)](#_81xx5ddsjcur) 10

[Table 3b.1 (created using Python)](#_rqekrppn0xcf) 10

[Table 3b.2 (created using Python)](#_w9leal5sded5) 11

[Chart 4b (created using Python)](#_dnwhnufm5y0a) 11

[Table 4b.1 (created using Python)](#_acsy2yw7w92l) 11

[Table 4b.2 (created using Python)](#_e8v1ibhr17l1) 12

[**Question 1c**](#_i09qfdw70zj) **12**

[**Appendix**](#_l9dif5z0dgja) **15**

[Code for 1b](#_j64c7xjha48r) 15

[Code for Observation 1](#_1mbnmo727zj2) 15

[Code for Observation 2](#_mua4s0opc63u) 18

[Code for Observation 3](#_pkgueifgxy7m) 19

[Code for Observation 4](#_3iwzz3dzjrkx) 22

[Code for 1c](#_ktlu2ijf1jo2) 25

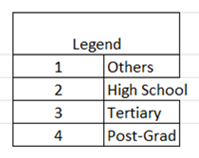
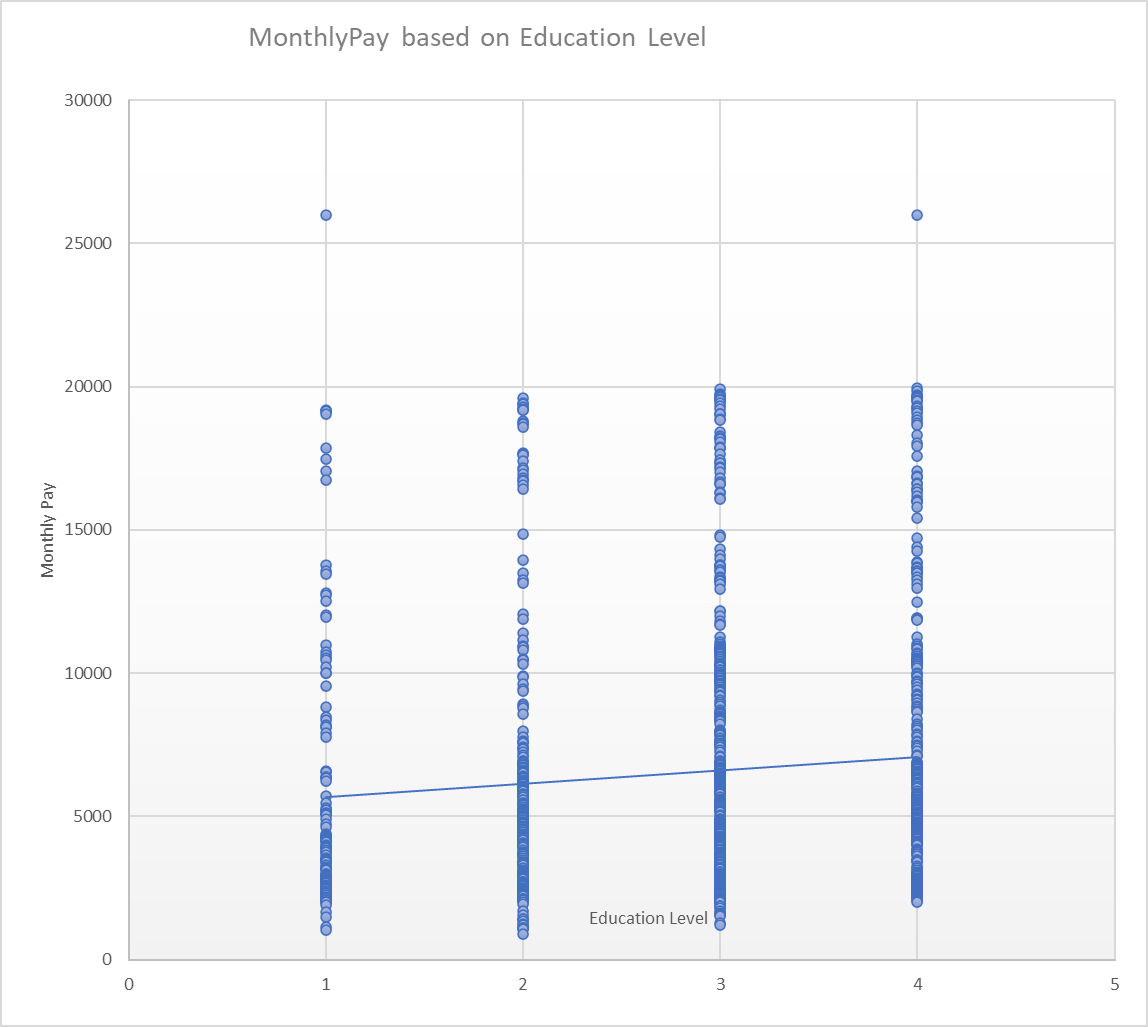
[**References**](#_aateeyz68n1h) **29**

# 

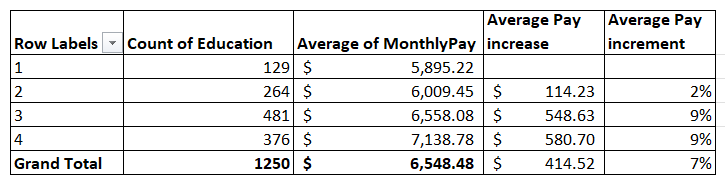
# **Question 1a**

## **Observation 1**

### **Chart 1a (created using Excel)**

Monthly Pay based on Education Level  


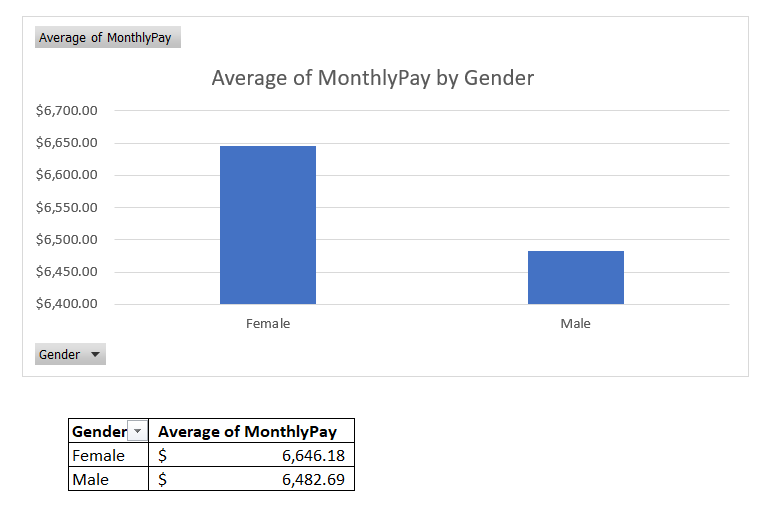
**Table 1a (created using Excel)**

Monthly Pay based on Education Level  


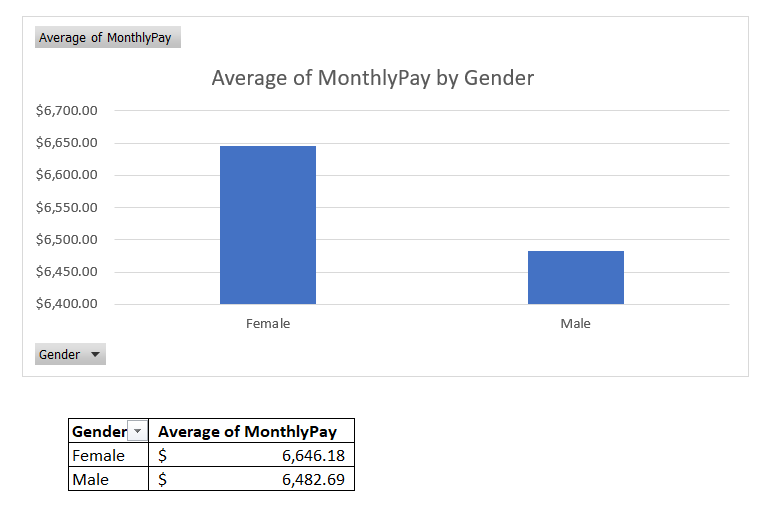
Every Advancement in Academic Achievement gives an Average pay increase of 7%. The Data shows that the average pay increase per increase in academic achievement is 7% with the “others” to “High School” showing the least at a 2% increase ($114.23) and “high school” to “tertiary” and “tertiary” to “Post Grad” showing an average of 9% at $548.63 and $580.70 respectively. This is of stark contrast to Singapore, where degree holders (post-grad) earn 72.2% more than Diploma holders (tertiary) (Lim, Average Median Income/Salary in Singapore by Age, 2022). This has proven that education does on average lead to a higher monthly average salary. However, the table showed that education is not the limiting and deciding factor in achieving a high salary. All education levels from “other” to “post-grad” have samples of salary over $15,000.

## **Observation 2**

### **Chart 2a (created using Excel)**

Average of Monthly Pay by Gender  


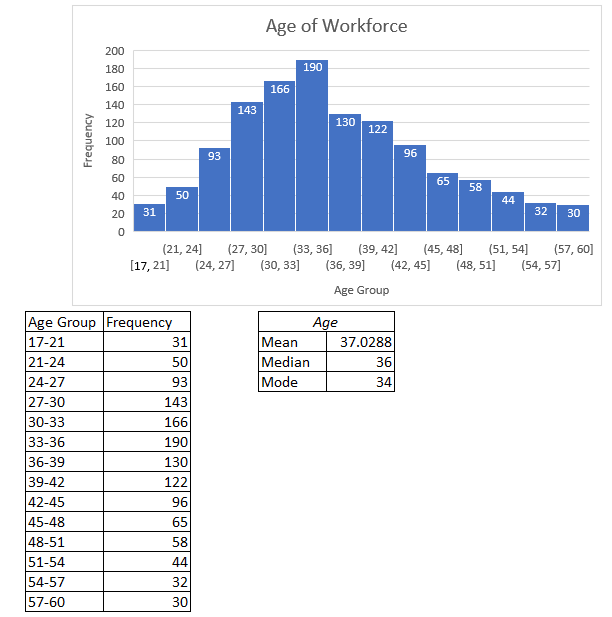
### **Table 2a (created using Excel)**

Average of Monthly Pay by Gender  


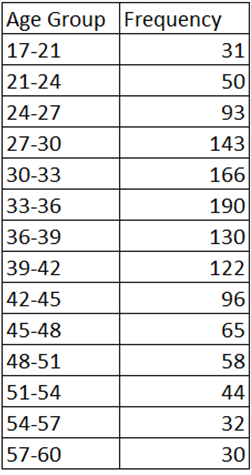
Females earn 2.5% more than Males on average. Females on average earn $163.49 or 2.5% more than their male counterparts. This is a stark contrast when compared to Singapore, where Males earn 25% more in 2021 (Statista, 2022) and 10% more when comparing the median income (Lim, Average Median Income/Salary in Singapore by Age, 2022). This can provide helpful insight into the company’s HR policy i.e. where pay equality is observed and overall team composition i.e. the possibility that high-income females hold top positions.

## **Observation 3**

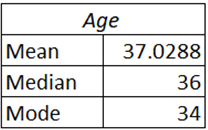
### **Chart 3a (created using Excel)**

Age of Workforce  


### **Table 3a.1 (created using Excel)**

Frequency Table  


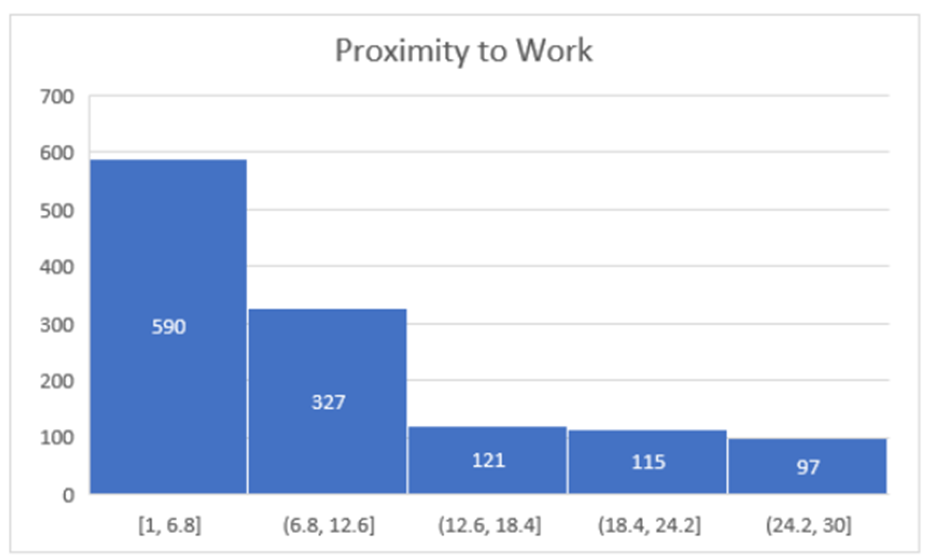
### **Table 3a.2 (created using Excel)**

Statistics Summary  


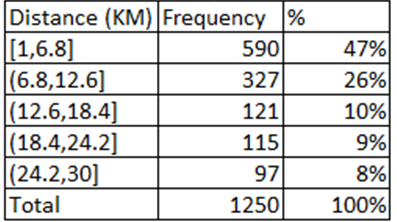
Younger Workers in the Company Compared to US National Average. The age distribution of the workforce in the company is normally distributed. It is slightly skewed to the right, with a mean age of 37 and a median of 36, indicating that the workforce is young. The median age in the company is 36. It is lower than the US median working age of 42 in 2022 (Bureau of Labor Statistics, 2022). The oldest worker is 60 as compared to retire in the US which is 67 (Ue.S. Social Security, 2023) indicating that the company has a relatively young workforce.

## **Observation 4**

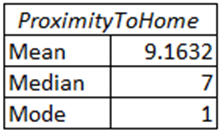
### **Chart 4a (created using Excel)**

Employees’ Proximity to Work  


### **Table 4a.1 (created using Excel)**

Frequency Table  


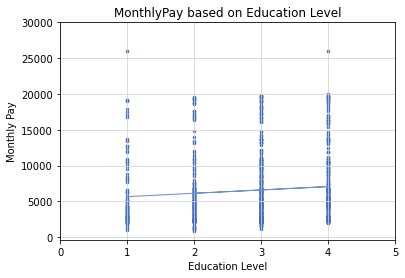
### **Table 4a.2 (created using Excel)**

Statistics Summary  


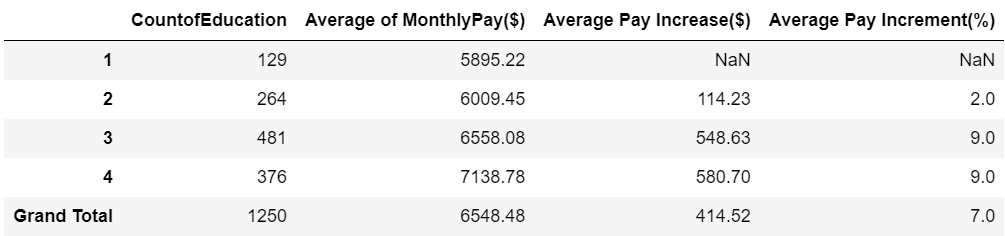
Talent Pool Referencing Proximity to Work. Talent Pool Referencing Distance to Work. 73% of the workforce stays within 12.6km of the workplace. With 47% staying within 6.8km and a median of 7km. This highlights the importance of the location of the workplace in the context of talent acquisition. From the data, it was also shown that the maximum distance an employee would travel is 29km. This, therefore, shows that if the company wants to have a rich talent pool to select from, it must be located near the talent pool and make itself more attractive compared to other competing companies.

# **Question 1b**

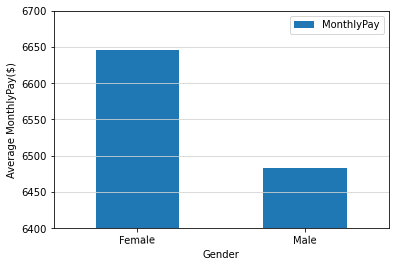
## **Chart 1b (created using Python)**

Monthly Pay based on Education Level  


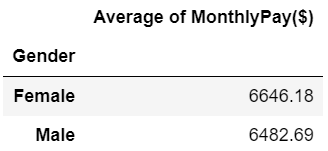
## **Table 1b (created using Python)**

Monthly Pay based on Education Level  


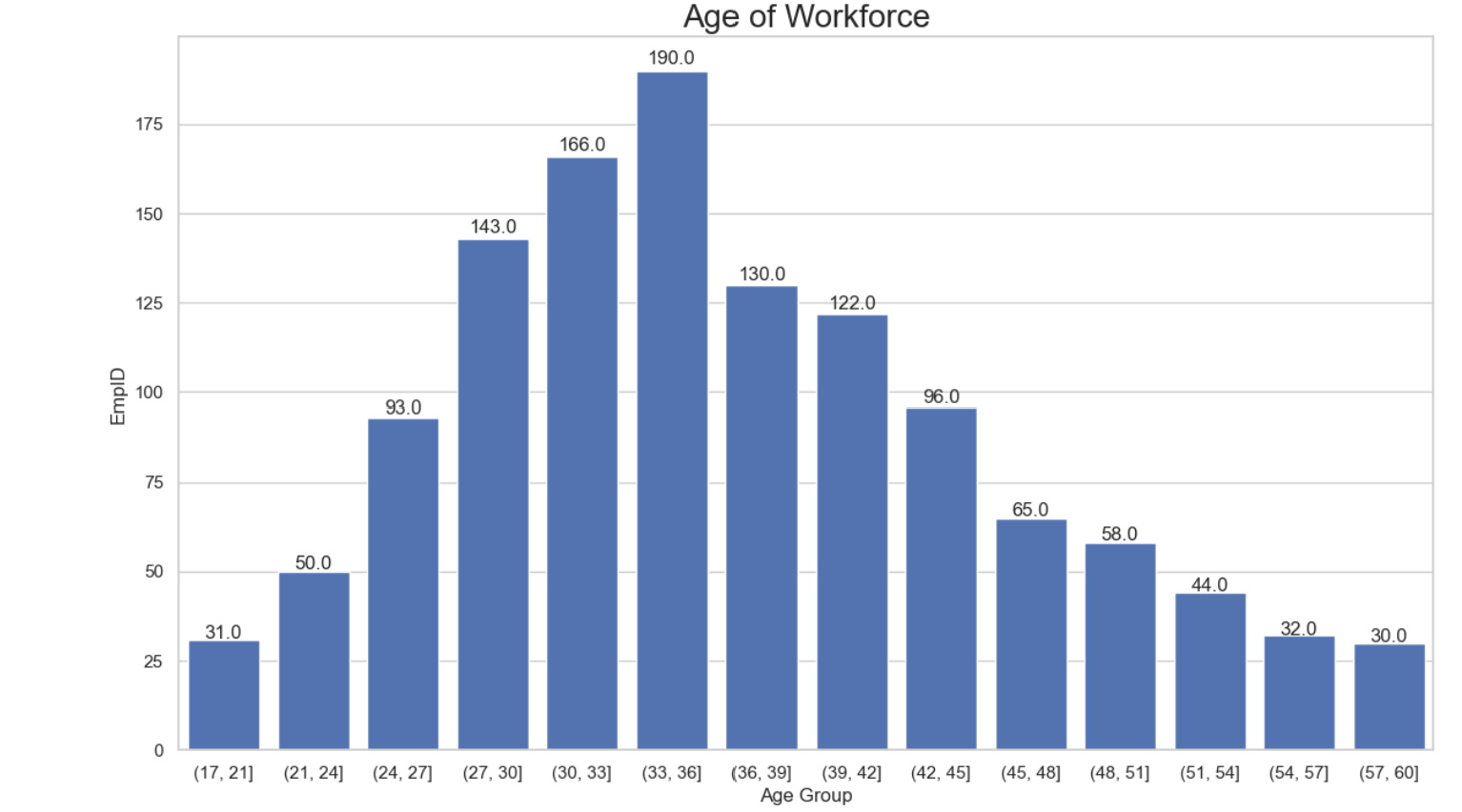
## **Chart 2b (created using Python)**

Average of Monthly Pay by Gender  


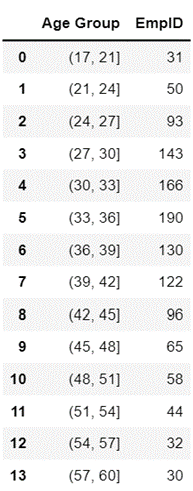
## **Table 2b (created using Python)**

Average of Monthly Pay by Gender ****

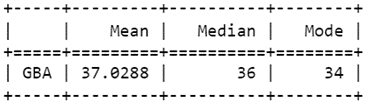
## **Chart 3b (created using Python)**

Age of Workforce  
****

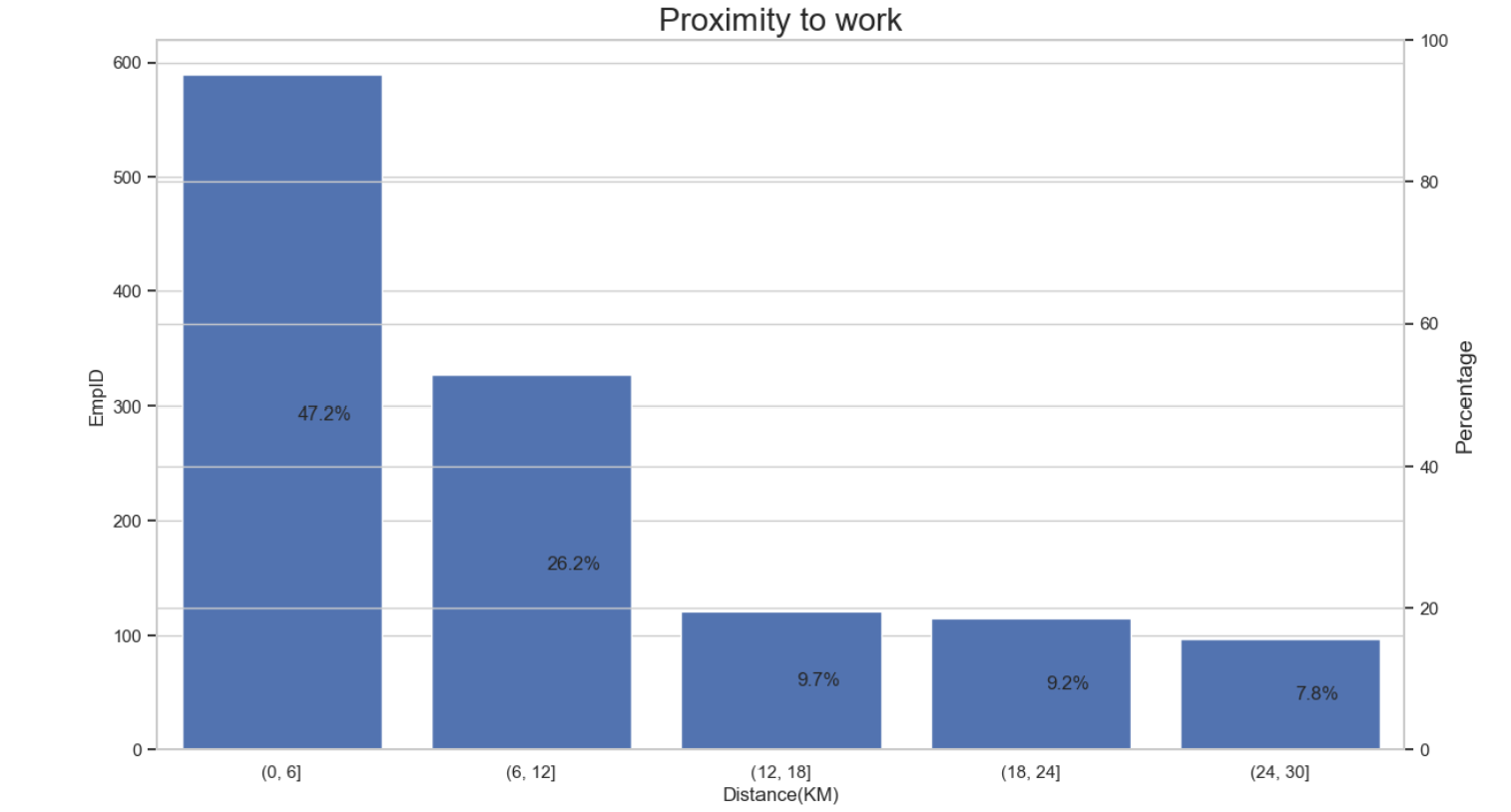
## **Table 3b.1 (created using Python)**

Frequency Table ****

## **Table 3b.2 (created using Python)**

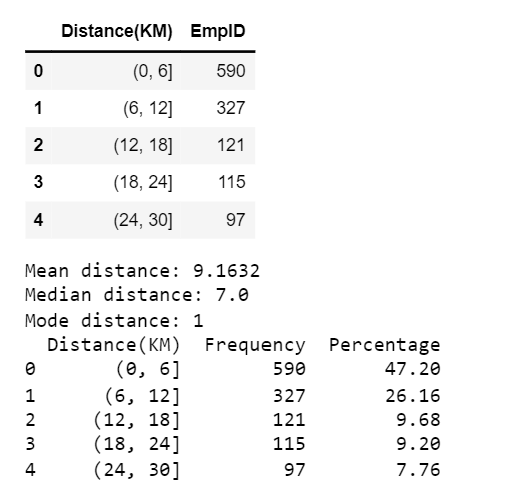
Statistics Summary  
****

## **Chart 4b (created using Python)**

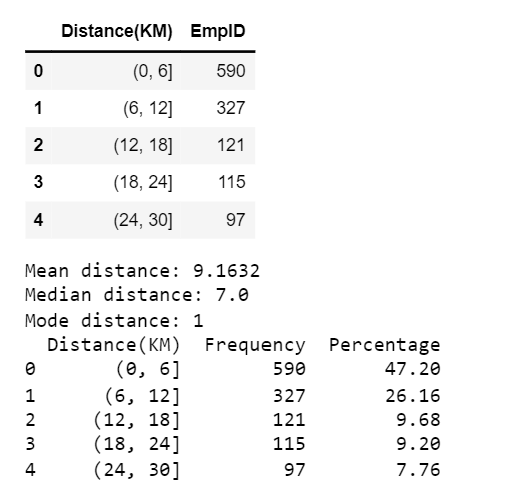
Employees’ Proximity to Work  
****

## **Table 4b.1 (created using Python)**

Frequency Table

****

## **Table 4b.2 (created using Python)**

Statistics Summary  
****

# **Question 1c**

**Output 1 - Current staff strength**

Welcome. What would you like to do?

[1] Enter 1 to see the Current staff strength.

[2] Enter 2 to see Current staff annual renumeration.

[3] Enter 3 to see Eligible long service award recipients.

[4] Enter 4 to Exit.

What would you like to do? 1

The current staff strength is 1057

There are 495 Product Development who are Male

There are 327 Product Development who are Female

There are 218 Business Development who are Male

There are 159 Business Development who are Female

There are 34 HR who are Male

There are 17 HR who are Female

[1] Enter 1 to see the Current staff strength.

[2] Enter 2 to see Current staff annual renumeration.

[3] Enter 3 to see Eligible long service award recipients.

[4] Enter 4 to Exit.

What would you like to do? (Textbox for input option 1-4)

**Output 2 - Current staff annual remuneration**

Welcome. What would you like to do?

[1] Enter 1 to see the Current staff strength.

[2] Enter 2 to see Current staff annual renumeration.

[3] Enter 3 to see Eligible long service award recipients.

[4] Enter 4 to Exit.

What would you like to do? 2

The current total annual staff renumeration is $ 7265209

The current median monthly salary is $ 5238

[1] Enter 1 to see the Current staff strength.

[2] Enter 2 to see Current staff annual renumeration.

[3] Enter 3 to see Eligible long service award recipients.

[4] Enter 4 to Exit.

What would you like to do? (Textbox for input option 1-4)

**Output 3 - Eligible long service award recipients**

Welcome. What would you like to do?

[1] Enter 1 to see the Current staff strength.

[2] Enter 2 to see Current staff annual renumeration.

[3] Enter 3 to see Eligible long service award recipients.

[4] Enter 4 to Exit.

What would you like to do? 3

EmpID JobRole Age

295 1295 Account Manager 59

896 1896 Section Manager 59

1242 2242 Section Manager 57

466 1466 Section Manager 55

[1] Enter 1 to see the Current staff strength.

[2] Enter 2 to see Current staff annual renumeration.

[3] Enter 3 to see Eligible long service award recipients.

[4] Enter 4 to Exit.

What would you like to do? (Textbox for input option 1-4)

**Output 4 - Exit**

Welcome. What would you like to do?

[1] Enter 1 to see the Current staff strength.

[2] Enter 2 to see Current staff annual renumeration.

[3] Enter 3 to see Eligible long service award recipients.

[4] Enter 4 to Exit.

What would you like to do? 4

You have selected Exit.

Thanks again, goodbye for now.

# **Appendix**

## **Code for 1b**

### **Code for Observation 1**

|  |
| --- |
| # to import the libraries  import pandas as pd  import matplotlib.pyplot as plt  import numpy as np  import statistics as stat  ## creating chart  # to read GBA csv file  df\_EmployeeInfo = pd.read\_csv("GBA.csv")  # to extract columns required for scatter chart  df\_EmployeePaybyEducation = df\_EmployeeInfo [['Education','MonthlyPay']]  display(df\_EmployeePaybyEducation) # show columns required  # to set x and y values for the plot  x\_axis = df\_EmployeePaybyEducation['Education']  y\_axis = df\_EmployeePaybyEducation['MonthlyPay']  # initiating the plot  plt.scatter(x\_axis, y\_axis, s = 6, color = "#426fc0")  plt.title('MonthlyPay based on Education Level')  # to calculate equation for trendline  z = np.polyfit(x\_axis, y\_axis, 1)  p = np.poly1d(z)  # to add trendline to plot  plt.plot(x\_axis,p(x\_axis),"-",color = "#6d8dcd", linewidth = 1) # adding hex color codes to trendline  # to set the x and y axis labels for plot  plt.xlabel('Education Level')  plt.ylabel('Monthly Pay')  # to set x and y ticks label  plt.xticks(ticks = range(0,6,1))  plt.yticks(ticks = range(0,35000,5000))  # to create background grids  plt.grid(color = 'lightgrey', linestyle = '-')  # to show the plot  plt.show()  ## creating table  # column 2: computing number of employees for each education level  CountofEducation = df\_EmployeePaybyEducation.groupby('Education').count()  # column 2 total: computing total number of employees  TotalEmployees = df\_EmployeePaybyEducation.groupby('Education').count().sum()  # column 3: computing average monthly pay for each education level  AveragePaybyEducation = df\_EmployeePaybyEducation.groupby('Education').mean().round(2)  # column 3 total: computing average monthly pay for all employees  AveragePay = round(df\_EmployeePaybyEducation['MonthlyPay'].sum()/len(df\_EmployeePaybyEducation['MonthlyPay']),2)  # column 4: computing average pay increase  AveragePayIncrease = df\_EmployeePaybyEducation.groupby('Education').mean().round(2).diff()  # column 4 total: computing mean of average pay increase  MeanAveragePayIncrease = df\_EmployeePaybyEducation.groupby('Education').mean().round(2).diff().iloc[1:].sum()/3  # column 5: computing average pay increase in percentage  AveragePayIncreasePctChange = df\_EmployeePaybyEducation.groupby('Education').mean().round(2).pct\_change().mul(100).round(0)  #column 5 total: computing mean of average pay increase in percentage  result\_df = df\_EmployeePaybyEducation.groupby('Education').mean().round(2).pct\_change().mul(100).round(0)  result\_df = result\_df.apply(pd.to\_numeric, errors='coerce')  MeanAveragePayIncreasePctChange = (result\_df.sum().sum()/3).round()  # existing DataFrame  frame = [CountofEducation, AveragePaybyEducation, AveragePayIncrease, AveragePayIncreasePctChange]  result = pd.concat((frame), axis=1, join='inner')  result.set\_axis(['CountofEducation', 'Average of MonthlyPay($)', 'Average Pay Increase($)', 'Average Pay Increment(%)'], axis='columns', inplace=True)  new\_row = {'CountofEducation': TotalEmployees.values[0], 'Average of MonthlyPay($)': AveragePay, 'Average Pay Increase($)': MeanAveragePayIncrease.values[0], 'Average Pay Increment(%)': MeanAveragePayIncreasePctChange}  # create a new DataFrame from the new row  new\_df = pd.DataFrame(new\_row, index=[0])  # concatenate the existing DataFrame with the new DataFrame  result = pd.concat([result, new\_df], ignore\_index=True)  # set the index to 1, 2, 3, 4, and 'Grand Total'  index\_values = [1, 2, 3, 4, 'Grand Total']  result = result.set\_index(pd.Index(index\_values))  # set the name of the index  result.index.name = 'Education Level'  # display the updated table  display(result) |

### **Code for Observation 2**

|  |
| --- |
| # to import the libraries  import pandas as pd  import numpy as np  import matplotlib.pyplot as plt  ## creating chart  # to read GBA csv file  df\_EmployeeInfo = pd.read\_csv("GBA.csv")  # to extract columns required for bar chart  df\_EmployeePaybyGender = df\_EmployeeInfo [['Gender','MonthlyPay']]  display(df\_EmployeePaybyGender) # show columns required  ## creating table  # to find average monthly pay by gender, rounded to 2 decimal places  df\_AverageMonthlyPaybyGender = df\_EmployeePaybyGender.groupby('Gender').mean().round(2)  # renaming column  renamed\_table = df\_AverageMonthlyPaybyGender.rename(columns={'MonthlyPay': 'Average of MonthlyPay($)'})  display(renamed\_table)  ## creating chart  # initiating the plot  ax = df\_AverageMonthlyPaybyGender.plot.bar(y='MonthlyPay', rot=1)  # to set y-label  plt.ylabel('Average MonthlyPay($)')  # to set y-axis limits  plt.ylim(6400, 6700)  # to create background grids  plt.grid(color='lightgrey', axis='y')  # to show the plot  plt.show() |

## 

### **Code for Observation 3**

|  |
| --- |
| import seaborn as sb  import pandas as pd  import numpy as np  import matplotlib.pyplot as plt  from tabulate import tabulate as tb  from collections import Counter  gbadata = pd.read\_csv("GBA.csv")  chart3\_df = gbadata.copy()  #creating bins  bins=[17,21,24,27,30,33,36,39,42,45,48,51,54,57,60]  chart3\_df["Age Group"]=pd.cut(chart3\_df["Age"], bins)  chart3\_df=chart3\_df.groupby("Age Group")["EmpID"].count()  chart3\_df=chart3\_df.reset\_index()  display(chart3\_df)  # Create the table as a DataFrame  mean\_age = gbadata["Age"].mean()  median\_age = gbadata["Age"].median()  def my\_mode(sample):  c = Counter(sample)  return [k for k, v in c.items() if v == c.most\_common(1)[0][1]]  mode\_age = my\_mode(gbadata["Age"])  data = {"Mean Age": [mean\_age], "Median Age": [median\_age], "Mode Age": [mode\_age[0]]}  table = pd.DataFrame(data, index=["GBA"])  # Print the table with lines  print(tb(table, headers=["Mean", "Median","Mode"], tablefmt="grid"))  #show values in barplot  def show\_values(axs, orient="v", space=.02):  def \_single(ax):  if orient == "v": #if orientation is vertical  for p in ax.patches:  \_x = p.get\_x() + p.get\_width() / 2  \_y = p.get\_y() + p.get\_height() + (p.get\_height()\*0.01)  value = '{:.1f}'.format(p.get\_height())  ax.text(\_x, \_y, value, ha="center")  elif orient == "h": #if orientation is horizontal  for p in ax.patches:  \_x = p.get\_x() + p.get\_width() + float(space)  \_y = p.get\_y() + p.get\_height() - (p.get\_height()\*0.5)  value = '{:.1f}'.format(p.get\_width())  ax.text(\_x, \_y, value, ha="left")  if isinstance(axs, np.ndarray): #enumerating  for idx, ax in np.ndenumerate(axs):  \_single(ax)  else:  \_single(axs)  sb.set\_theme(style="whitegrid")  f, ax=plt.subplots(figsize=(14,8))  barplot2 = sb.barplot(data=chart3\_df, x="Age Group", y="EmpID", color="#426fc0")  barplot2.set\_title("Age of Workforce", fontsize=20)  show\_values(barplot2, "v", space=0) |

## 

### **Code for Observation 4**

|  |
| --- |
| # to import the libraries  import seaborn as sb  import pandas as pd  import numpy as np  import matplotlib.pyplot as plt  from tabulate import tabulate as tb  from collections import Counter  #importing excel file  gbadata = pd.read\_csv("GBA.csv")  chart4\_df = gbadata.copy()  #creating bins  bins=[0,6,12,18,24,30]  chart4\_df["Distance(KM)"]=pd.cut(chart4\_df["ProximityToHome"], bins)  chart4\_df=chart4\_df.groupby("Distance(KM)")["EmpID"].count()  chart4\_df=chart4\_df.reset\_index()  display(chart4\_df)  # Create a copy of the original data and extract the distance values  distance\_data = gbadata.copy()["ProximityToHome"]  # Calculate the mean, median, and mode of the distance data  distance\_mean = np.mean(distance\_data)  distance\_median = np.median(distance\_data)  distance\_mode = Counter(distance\_data).most\_common(1)[0][0]  # Print the results  print("Mean distance:", distance\_mean)  print("Median distance:", distance\_median)  print("Mode distance:", distance\_mode)  ##creating the table with percentages  import pandas as pd  # create a copy of the original data and group by distance bins  chart4\_df = gbadata.copy()  bins = [0, 6, 12, 18, 24, 30]  chart4\_df["Distance(KM)"] = pd.cut(chart4\_df["ProximityToHome"], bins)  chart4\_df = chart4\_df.groupby("Distance(KM)")["EmpID"].count()  chart4\_df = chart4\_df.reset\_index()  # Define the frequency of each value as a list using the chart4\_df dataframe  freq = chart4\_df["EmpID"].tolist()  # Calculate the total frequency  total\_freq = sum(freq)  # Calculate the percentage of frequency for each value  perc\_freq = [f / total\_freq \* 100 for f in freq]  # Create a pandas DataFrame from the frequency and percentage data  data = {"Distance(KM)": chart4\_df["Distance(KM)"].tolist(), "Frequency": freq, "Percentage": perc\_freq}  df = pd.DataFrame(data)  # Print the DataFrame to the console  print(df)  ##creating the chart  # Calculate the percentage of each distance category  df = chart4\_df.copy()  # create a copy of the original data and group by distance bins  chart4\_df = gbadata.copy()  bins = [0, 6, 12, 18, 24, 30]  chart4\_df["Distance(KM)"] = pd.cut(chart4\_df["ProximityToHome"], bins)  chart4\_df = chart4\_df.groupby("Distance(KM)")["EmpID"].count()  chart4\_df = chart4\_df.reset\_index()  # Define the frequency of each value as a list using the chart4\_df dataframe  freq = chart4\_df["EmpID"].tolist()  # Calculate the total frequency  total\_freq = sum(freq)  # Calculate the percentage of frequency for each value  df['%'] = [f / total\_freq \* 100 for f in freq]  # Plot the bar chart  sb.set\_theme(style="whitegrid")  f, ax = plt.subplots(figsize=(14,8))  barplot3 = sb.barplot(data=df, x="Distance(KM)", y="EmpID", color="#426fc0")  barplot3.set\_title("Proximity to work", fontsize=20)  # Create a second y-axis for the percentage values  ax2 = ax.twinx()  ax2.set\_ylim(0, 100)  ax2.set\_ylabel('Percentage', fontsize=14)  # Plot the percentage values on the second y-axis  for index, row in df.iterrows():  ax2.annotate(str(round(row['%'], 1)) + '%', xy=(row.name, row['%']), xytext=(10, 0),  textcoords='offset points', ha='left', va='center', fontsize=12)    plt.show() |

## **Code for 1c**

|  |
| --- |
| # import libraries  import pandas as pd  import numpy as np  # read excel file located on my desktop  xl\_file = pd.read\_excel("C:\\Users\\shado\\OneDrive\\Desktop\\GBA.xlsx")  # count LeftCompany by category  xl\_file['LeftCompany'].value\_counts()  # count current total staff strength  ss = xl\_file.LeftCompany.value\_counts().No  # count BusinessUnit by category  xl\_file['BusinessUnit'].value\_counts()  # count Gender by category  xl\_file['Gender'].value\_counts()  # count Product Development & Male and assign a variable pdm  pdm = len(xl\_file[(xl\_file.BusinessUnit == 'Product Development') & (xl\_file.Gender == 'Male') ])  # count Product Development & Female and assign a variable pdf  pdf = len(xl\_file[(xl\_file.BusinessUnit == 'Product Development') & (xl\_file.Gender == 'Female') ])  # count Business Development & Male and assign a variable bdm  bdm = len(xl\_file[(xl\_file.BusinessUnit == 'Business Development') & (xl\_file.Gender == 'Male') ])  # count Business Development & Female and assign a variable bdf  bdf = len(xl\_file[(xl\_file.BusinessUnit == 'Business Development') & (xl\_file.Gender == 'Female') ])  # count HR & Male and assign a variable hrm  hrm = len(xl\_file[(xl\_file.BusinessUnit == 'HR') & (xl\_file.Gender == 'Male') ])  # count HR & Female and assign a variable hrf  hrf = len(xl\_file[(xl\_file.BusinessUnit == 'HR') & (xl\_file.Gender == 'Female') ])  #Current total annual staff remuneration expenditure & returning whole number  e = xl\_file.loc[xl\_file['LeftCompany'] == 'No', 'MonthlyPay'].sum()  re = ("%d" % e)  #Current median monthly salary & returning whole number  m = xl\_file.loc[xl\_file['LeftCompany'] == 'No', 'MonthlyPay'].median()  rm = ("%d" % m)  #35 or more years of service and still in the company  xl\_file.loc[(xl\_file["LeftCompany"] == 'No') & (xl\_file["WithCompany"] >=35)]  # constrain with 35 or more years of service and assign a variable withcompany  withcompany = (xl\_file["WithCompany"]>=35)  # constrain with staff that did not left company and assign a variable noleft  noleft = (xl\_file["LeftCompany"]=="No")  #output information with all the required constraints  optionthree = xl\_file.loc[withcompany & noleft,['EmpID', 'JobRole', 'Age']].sort\_values(by='Age', ascending = False)  #output continue to run so long as option 4 has not been selected.  def current\_staff\_strength():  print('The current staff strength is', ss,  '\nThere are', pdm, 'Product Development who are Male',  '\nThere are', pdf, 'Product Development who are Female',  '\nThere are', bdm, 'Business Development who are Male',  '\nThere are', bdf, 'Business Development who are Female',  '\nThere are', hrm, 'HR who are Male',  '\nThere are', hrf, 'HR who are Female',)    def current\_staff\_annual\_renumeration():  print('The current total annual staff renumeration is $',re,  '\nThe current median monthly salary is $',rm)    def eligible\_long\_service\_award\_recipients():  print(optionthree)  # Give the user some context.  print("\nWelcome. What would you like to do?")  # Set an initial value for choice other than the value for 'quit'.  choice = ''  # Start a loop that runs until the user enters the value for 'quit'.  while choice != '4':  # Give all the choices in a series of print statements.  print("\n[1] Enter 1 to see the Current staff strength.")  print("[2] Enter 2 to see Current staff annual renumeration.")  print("[3] Enter 3 to see Eligible long service award recipients.")  print("[4] Enter 4 to Exit.")    # Ask for the user's choice.  choice = input("\nWhat would you like to do? ")    # Respond to the user's choice.  if choice == '1':  current\_staff\_strength()  elif choice == '2':  current\_staff\_annual\_renumeration()  elif choice == '3':  eligible\_long\_service\_award\_recipients()  elif choice == '4':  print("\nYou have selected Exit.\n")  else:  print("\nI don't understand that choice, please Select an option 1-4.\n")    # Print a message that we are all finished.  print("Thanks again, goodbye for now.") |

## 

# **References**

Bureau of Labor Statistics. (2022, Sep 8). Median age of the labor force, by sex, race and ethnicity. Retrieved from BLS: https://www.bls.gov/emp/tables/median-age-labor-force.htm).

Lim, A. (2022, Feb 14). Average Median Income/Salary in Singapore by Age. Retrieved from Smartwealth: https://smartwealth.sg/average-income-salary-singapore/).

Statista. (2022, Jun). Average mean monthly earnings per employee in Singapore from 2012 to 2021, by gender. Retrieved from Statista: https://www.statista.com/statistics/959519/singapore-mean-monthly-earnings-per-employee-by-gender/

Wu, K. Y. (2022). ANL252 Python for data analytics (study guide). Singapore University of Social Sciences.

U.S. Social Security. (2023, Jan 4). Full Retirement Age in US. Retrieved from U.S. Social Security: https://faq.ssa.gov/en-us/Topic/article/KA-01885#:~:text=The%20current%20full%20retirement%20age,Of%20Birth%20for%20more%20information