A picture containing logo

Description automatically generated

**ANL252**

**Python for Data Analytics:**

**Group-based Assignment**

**JANUARY 2023 Presentation**

|  |  |  |
| --- | --- | --- |
| **Name** | **PI Number** | **Submission Date** |
| Teh Zi Jing | M2211393 | 19 Feburary 2023, 2355Hrs |
| Jansen Edmund Joseph Junior | H2070894 | 19 Feburary 2023, 2355Hrs |
| Prabu S/O Ramasamy | Q2110662 | 19 Feburary 2023, 2355Hrs |
| Tey Jin Hean | K1810665 | 19 Feburary 2023, 2355Hrs |

Declaration Page

We, members of group 6, do hereby declare that we each contributed to this assignment and that we collectively agree to a shared grade.

|  |  |  |
| --- | --- | --- |
| Name | Contribution | Signature |
| Teh Zi Jing | I did questions 1a, 1b, 1c | Teh Zi Jing |
| Jansen Edmund Joseph Junior | I did questions 1c | Jansen Edmund Joseph Junior |
| Prabu S/O Ramasamy | I did questions 1c | Prabu |
| Tey Jin Hean | I did questions 1b | Tey |

**Question 1a)**

The dataset provided has 26 variables and 1250 records. This information allows us to analyse the data according to different decision-making criteria.

**Chart and Table 1 (Number of employees in different business units that stayed in the company)**

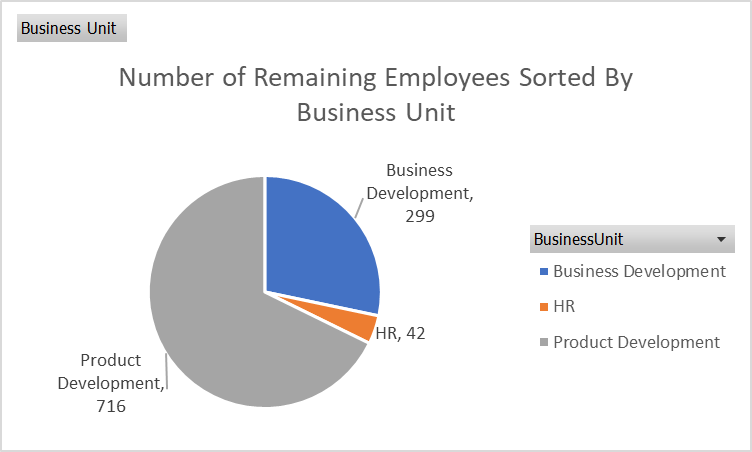
Firstly, we use the excel PivotTable function to generate the number of employees in each business unit and the number of employees who have left the company. With this information, we have created a pie chart to show the number of remaining employees in each business unit.

Graphical user interface, application

Description automatically generated Graphical user interface, text, application

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| **Business Unit** | **Left Company** |  |  |
| **Row Labels** | **No** | **Yes** | **Grand Total** |
| Business Development | 299 | 78 | 377 |
| HR | 42 | 9 | 51 |
| Product Development | 716 | 106 | 822 |
| **Grand Total** | **1057** | **193** | **1250** |



**Insights and Interesting Observations**

Knowing the number of employees in each business unit is crucial information because it allows the organisation to evaluate the labour cost, productivity, workload, job responsibilities, and possible company growth. The pie chart above shows that most employees are in the product development role, followed by business development and HR. The large number of product developers shows that the organisation is highly innovative and may be in a fast-paced industry, such as goods or technology, as they need to keep up with the rapid change in the market. The organisation's high number of business developers also suggests that the organisation focuses on identifying new opportunities, building relationships with partners, and developing sales strategies to increase revenue.

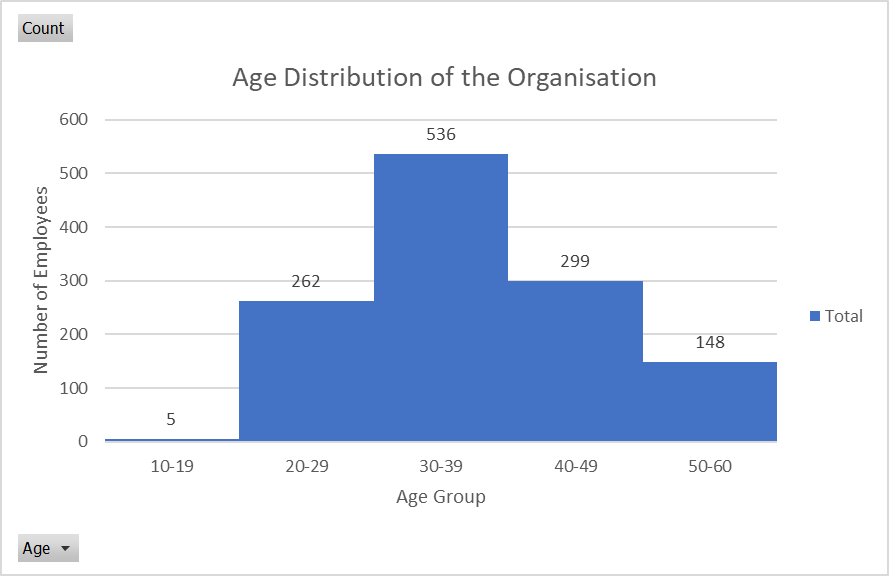
**Chart and Table 2 (Age Distribution of the Organisation)**

An excel PivotTable function was created to find the bin and the number of counts within each bin. The bin, in this case, refers to the different age groups, which are grouped in 10 years intervals. To create this bin, we use the grouping function in excel to help us distribute the age groups accordingly. After that, we use Histogram to show the distribution of employees by age.

**Graphical user interface, application

Description automatically generated**

|  |  |
| --- | --- |
| **Age** | **Count** |
| 10-19 | 5 |
| 20-29 | 262 |
| 30-39 | 536 |
| 40-49 | 299 |
| 50-60 | 148 |
| **Grand Total** | **1250** |

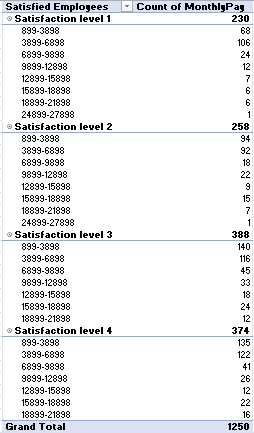
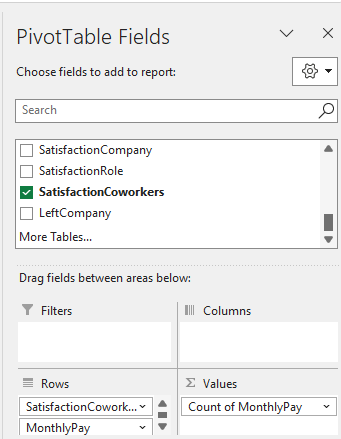


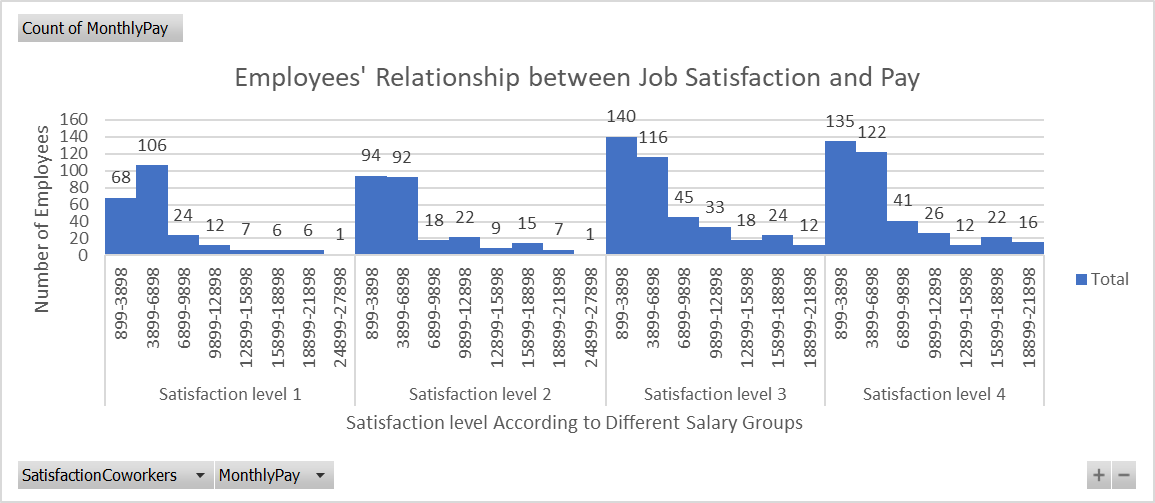
**Insights and Interesting Observations**

The Histogram above shows us the total number of employees in each Age Group. This information is useful as it allows the organisation to analyse and predict the possibility of employees leaving and their reasons for leaving. For example, in this case, we can assume that the 5 employees in the age group 10-19 are working in the organisation as interns and will most likely leave after a few months. The middle-aged group is more likely to leave the organisation to look for better opportunities, and the age group of 50-60 is likely to leave the organisation to retire. By understanding the age of employees, the organisation can do succession planning by ensuring that there are enough employees in the 40-49 age group to take over the roles of those who are more likely to retire, which in this case is the age group 50-60. The age of employees also allows the organisation to know that its workforce is well diversified, which brings in different perspectives and experiences, thus resulting in more informed decisions.

**Chart and Table 3 (Employees’ Relationship between Job Satisfaction and Pay)**

An Excel PivotTable was created to find the employees’ satisfaction level in relation to their monthly pay. We used the grouping function in excel to group the employees with different salaries into 8 groups with $3000 intervals. From there, we can compare whether those employees with lower salaries are less satisfied than those with higher salaries.





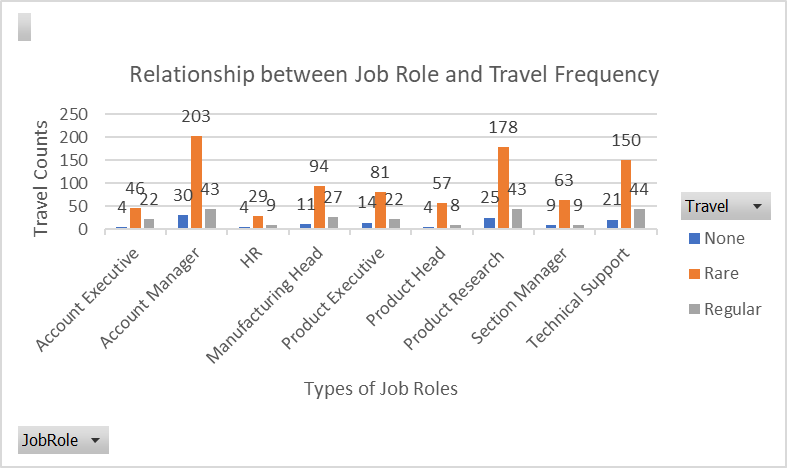
**Insights and Interesting Observations**

Salary may affect an employee’s work satisfaction level, but it is not the only factor affecting it. While a high salary is a strong motivator to employees’ satisfaction, it has to come with other factors such as a good work-life balance and relationship with their superiors and colleagues. This can be seen in the chart above as the 2 employees in the $24899-27898 per month salary range only have a satisfaction level of 1 and 2. Whereas, even though most employees earn between $899-3898 and $3899-6898, there are more counts in satisfaction levels 3 and 4 as compared to levels 1 and 2. The employees’ satisfaction level is correlated to their pay for the rest of the salary range of 6899 to 21898. This can be seen as there are more counts of employees in the level 3 and 4 satisfaction levels for these groups of employees.

**Chart and Table 4** **(Relationship between Job Role and Travel Frequency)**

An Excel Pivot Table was created to find the relationship between the Job Role and how often employees in the Job Role travel. We created a row of Job Roles and different columns of travel frequency and their counts.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Travel Frequency** |  |  |  |
| **Job Role** | **None** | **Rare** | **Regular** | **Grand Total** |
| **Account Executive** | **4** | **46** | **22** | **72** |
| **Account Manager** | **30** | **203** | **43** | **276** |
| **HR** | **4** | **29** | **9** | **42** |
| **Manufacturing Head** | **11** | **94** | **27** | **132** |
| **Product Executive** | **14** | **81** | **22** | **117** |
| **Product Head** | **4** | **57** | **8** | **69** |
| **Product Research** | **25** | **178** | **43** | **246** |
| **Section Manager** | **9** | **63** | **9** | **81** |
| **Technical Support** | **21** | **150** | **44** | **215** |
| **Grand Total** | **122** | **901** | **227** | **1250** |

****

**Insights and Interesting Observations**

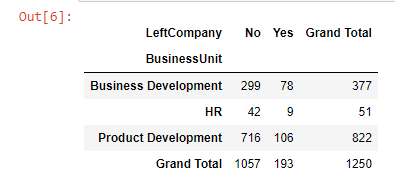
Organisations need to understand which job roles require frequent travelling and allocate travel incentives to increase employees’ satisfaction and reduce stress levels. The organisation can better plan their expenses with more information on the job roles that require travelling. This allows them to know how much to set aside, which helps them manage their funds efficiently to prevent money from idling. It can be seen from the chart that the Technical Support role travel most frequently, followed by Product Research and Account Manager. This is the area where the organisation can look into to determine if travelling is necessary or if there are other alternatives, such as meeting through online platforms like Zoom.

**Question 1b)**

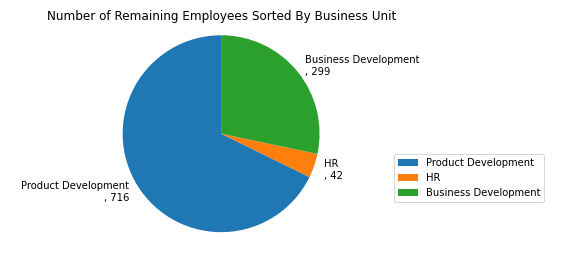
**Chart and Table 1 (Number of employees in different business units that stayed in the company) Python Code:**

|  |  |
| --- | --- |
| In [1]: | #Importing pandas with alias pd  import pandas as pd |
| In [2]: | #Defining gba\_file to the .csv file filled with the GBA data  #r is to prevent python from reading \ as an escape character  gba\_file = r"C:\Users\zijin\OneDrive\Desktop\GBA.csv" |
| In [3]: | #Read the file using Pandas and read\_csv function  df = pd.read\_csv(gba\_file) |
| In [4]: | #Group the dataframe based on the variables 'BusinessUnit' and 'LeftCompany'  #Use .value\_counts() to count the number of employees left in each department  #Use unstack method to prevent the numbers from stacking  #use fill\_value=0 to fill any missing values with the value 0  df\_grouped\_Chart1 = df.groupby(by='BusinessUnit')['LeftCompany'].value\_counts().unstack(fill\_value=0) |
| In [5]: | #Add in a new column after adding up the values of each row and name it 'Grand Total'  df\_grouped\_Chart1['Grand Total'] = df\_grouped\_Chart1.sum(axis=1)  #Add in a new row after adding up the values of each column and name it 'Grand Total'  #Use .loc to locate and access the existing row of 'no' and 'yes'  df\_grouped\_Chart1.loc['Grand Total'] = df\_grouped\_Chart1.sum(axis=0) |
| In [6]: | #Show the pivot table  df\_grouped\_Chart1 |
| Out[6]: |  |
| In [7]: | #import matplotlib  import matplotlib.pyplot as plt  #Show the count of employees who have not left the company  remaining\_employees = df\_grouped\_Chart1.loc[:, 'No']  #Split the employees into different business unit groups: Product Development, HR, and Business Development  #Put it in an order that allows it to show the same chart as the excel chart  remaining\_employees = remaining\_employees[['Product Development', 'HR', 'Business Development']]  # Plot the pie chart  chart, values = plt.pie(remaining\_employees, labels = remaining\_employees.index, startangle=90)  #Adding labels for the pie chart  for i, val in enumerate(remaining\_employees.values):  values[i].set\_text(f'{remaining\_employees.index[i]}\n, {str(val)}')  #Plot the legend and assigning it to be at the bottom right corner  plt.legend(chart, remaining\_employees.index, bbox\_to\_anchor=(1,0.3), loc="center left")  #Plotting the chart title  plt.title('Number of Remaining Employees Sorted By Business Unit')  #Ensure same scale for both x and y axis  plt.axis('equal')  # Display the chart  plt.show() |
| Out [7]: |  |

**Table 1’s output:**



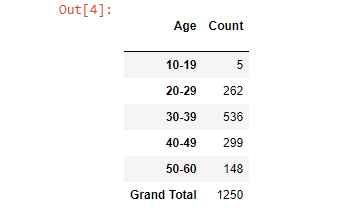
**Table 1’s chart:**

****

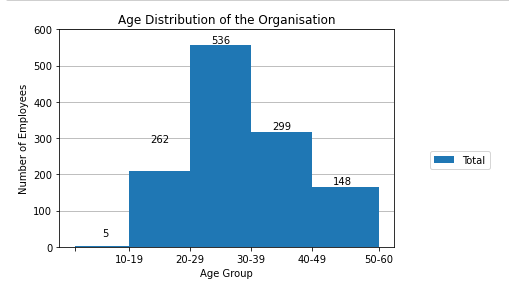
**Chart and Table 2 (Age Distribution of the Organisation) Python Code:**

|  |  |
| --- | --- |
| In [1]: | #Importing pandas with alias pd  import pandas as pd |
| In [2]: | #Defining gba\_file to the .csv file filled with the GBA data  #r is to prevent python from reading \ as an escape character  gba\_file = r"C:\Users\zijin\OneDrive\Desktop\GBA.csv" |
| In [3]: | #Read the file using Pandas and read\_csv function  df = pd.read\_csv(gba\_file) |
| In [4]: | # Create the bins and its intervals under bin  bins = [10, 19, 29, 39, 49, 60]  #Create the bin labels  ranges = ['10-19', '20-29', '30-39', '40-49', '50-60']  #Create new column age\_bin as the column name Age has been used  #Use include\_lowest function to include the lowest value in the bins  df['age\_bin'] = pd.cut(df['Age'], bins=bins, labels=ranges, include\_lowest=True)  # Create the pivot table named table 1 showing the bin and the age count  table1 = pd.pivot\_table(df, index='age\_bin', values='Age', aggfunc='count', margins=True)  # Rename the Age bin column and add a title for the count column  table1 = table1.rename(columns={'Age': 'Count'})  #Rename 'All' to 'Grand Total' to match the excel table  table1 = table1.rename(index={'All': 'Grand Total'})  #Remove the name 'age\_bin'  table1.index.name = ''  #Add Column name 'Age' beside 'Count'  table1.columns.name = 'Age'  # Show the table  table1 |
| Out [4]: |  |
| In [5]: | import matplotlib.pyplot as plt  # Plot histogram with the use of zorder to 'draw' the histogram on top of the gridlines  plt.hist(df['Age'], bins=bins, zorder=2)  # set the range from 0 to 600 with 100 intervals  plt.yticks(range(0, 601, 100))  # set the Y axis number limit to 600  plt.ylim(top=600)  # add gridlines behind the histogram  plt.grid(axis='y', zorder=1)  #Label the x-axis as 'Age Group'  plt.xlabel('Age Group')  #Label the y-axis as 'Number of Employees'  plt.ylabel('Number of Employees')  #Add the Chart title 'Age Distribution of the Organisation'  plt.title('Age Distribution of the Organisation')  #Recreating the set of bins and values to make sure both have the same number of indexes  intervals = [10, 19, 29, 39, 49, 60]  intervals\_values = ['','10-19', '20-29', '30-39', '40-49', '50-60']  plt.xticks(intervals, intervals\_values)  # Remove the Grand Total row so it will not appear in the chart  table1 = table1.drop(index='Grand Total')  #Show the histogram bar chart values  #Locate the 'Count' values and input into the chart  #Align the text in the center and vertical alignment at bottom  #Use count + 20 to shift the values on top of the bar slightly so it wont overlap  for i in range(len(table1)):  count = table1.iloc[i]['Count']  plt.text(intervals[i] + 5, count + 20, str(count), ha='center', va='bottom')    # Add a blue legend showing 'Total'  plt.legend(['Total'], loc='center right', bbox\_to\_anchor=(1.3, 0.4))    #Show the histogram  plt.show() |
| Out [5]: |  |

**Table 2’s Output:**

****

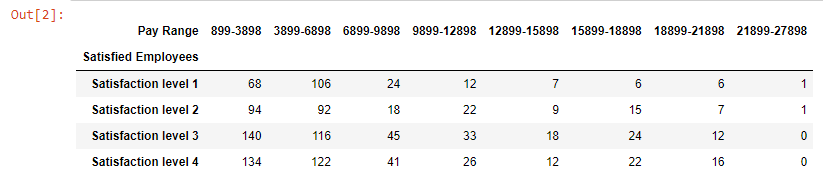
**Table 2’s Chart:**

****

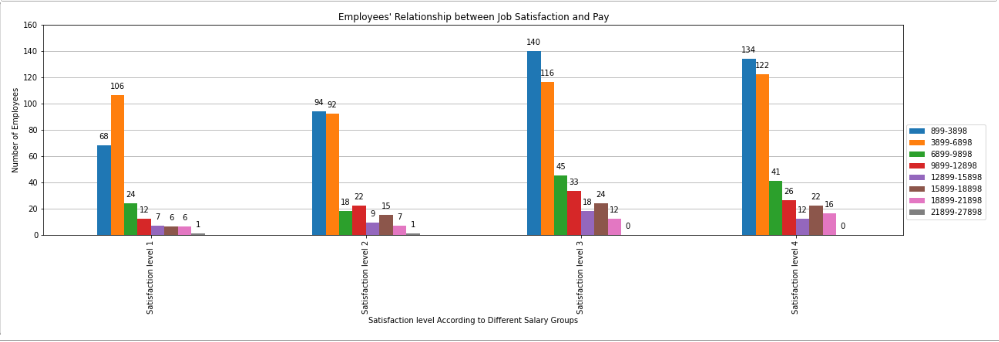
**Chart and Table 3 (Employees’ Relationship between Job Satisfaction and Pay) Python Code:**

|  |  |
| --- | --- |
| In [1]: | #Importing pandas with alias pd  import pandas as pd  #Defining gba\_file to the .csv file filled with the GBA data  #r is to prevent python from reading \ as an escape character  gba\_file = r"C:\Users\zijin\OneDrive\Desktop\GBA.csv"  #Read the file using Pandas and read\_csv function  df = pd.read\_csv(gba\_file) |
| In [2]: | #Define the different bins of monthly salary with an interval of $3000  bins = [899, 3898, 6898, 9898, 12898, 15898, 18898, 21898, 27898]  #Name each bin with labels that matches the excel table  ranges = ['899-3898', '3899-6898', '6899-9898', '9899-12898', '12899-15898', '15899-18898', '18899-21898', '21899-27898']  #Create a new column with variable name PayGroup using pd.cut() function  df['PayGroup'] = pd.cut(df['MonthlyPay'], bins=bins, labels=ranges)  #Rename the SatisfactionCoworkers column to Satisfied Employees and PayGroup column to Pay Range  df = df.rename(columns={'SatisfactionCoworkers': 'Satisfied Employees', 'PayGroup': 'Pay Range'})  #Replace the numbering of 1,2,3 and 4 with strings of satisfaction level 1, satisfaction level 2... level 4 in the Satisfied Employees column  df['Satisfied Employees'] = df['Satisfied Employees'].replace({1: 'Satisfaction level 1', 2: 'Satisfaction level 2', 3: 'Satisfaction level 3', 4: 'Satisfaction level 4'})  #Create a pivot table that shows the number of employees in each Pay Range and their satisfaction levels  table3 = pd.pivot\_table(df, values='MonthlyPay', index='Satisfied Employees', columns='Pay Range', aggfunc='count')  #Show the pivot table 3  table3 |
| Out [2]: |  |
| In [3]: | import matplotlib.pyplot as plt  #Create a bar chart using the pivot table and set the figsize to increase width to 20 inches wide and 5 inches tall  #name the variable chart3  #use of zorder to ensure that bar chart is in front of the gridlines to prevent the gridlines from cutting through it  chart3 = table3.plot(kind='bar', figsize=(20, 5), zorder = 2)  #Add gridlines behind the bar chart  plt.grid(axis='y', zorder=1)  #Set the range from 0 to 160 with 20 intervals  plt.yticks(range(0, 161, 20))  #Set the Y axis number limit to 160  plt.ylim(top=160)  #Create a chart title with name 'Employees' Relationship between Job Satisfaction and Pay'  #Use double quotation marks instead of single quotation mark to let python know that the quotation mark in Employees' is not to close the string  plt.title("Employees' Relationship between Job Satisfaction and Pay")  #Label the x-axis as 'Satisfaction level According to Different Salary Groups'  plt.xlabel('Satisfaction level According to Different Salary Groups')  #Label the y-axis as 'Number of Employees'  plt.ylabel('Number of Employees')  #Get the heights of each bars using .get\_height() method  heights = [bar.get\_height() for bar in chart3.patches]  #Add the values of each bars to the chart  #Use ha='center' to create a horizontal alignment for the values to be on the bar  #height + 5 to put the string slightly above the bars  for i, height in enumerate(heights):  chart3.text(chart3.patches[i].get\_x() + chart3.patches[i].get\_width() / 2, height + 5, str(int(height)), ha='center')    #Create a legend showing the different Pay Range colours  #Use loc 'center right' to shift the legend to the right side of the chart  #use bbox\_to\_anchor to shift the legend to the outside of the chart  chart3.legend(loc='center right', bbox\_to\_anchor=(1.1, 0.3))  #Show the chart  plt.show() |
| Out [3]: | (Clearer picture of the chart below) |

**Table 3’s Output:**

****

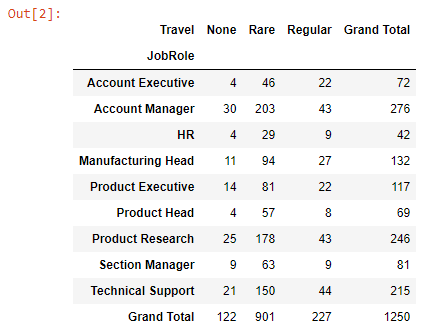
**Table 3’s Chart:**

****

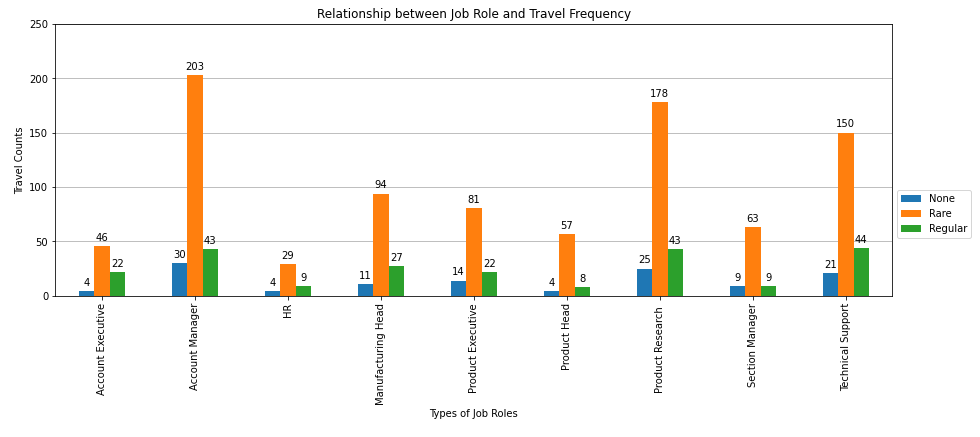
**Chart and Table 4** **(Relationship between Job Role and Travel Frequency) Python Code:**

|  |  |
| --- | --- |
| In [1]: | #Importing pandas with alias pd  import pandas as pd  #Defining gba\_file to the .csv file filled with the GBA data  #r is to prevent python from reading \ as an escape character  gba\_file = r"C:\Users\zijin\OneDrive\Desktop\GBA.csv"  #Read the file using Pandas and read\_csv function  df = pd.read\_csv(gba\_file) |
| In [2]: | #Use groupby function to group the data into JobRole and Travel, and get the count of travel frequency  #Use size() to get the count of each group  #Use unstack() to show a pivot table with 'Jobrole' as rows and 'Travel' as columns  #Lastly fill any missing values with 0  chart4 = df.groupby(['JobRole', 'Travel']).size().unstack(fill\_value=0)  #Add a new row showing the total count of travel frequency for each JobRole  chart4.loc['Grand Total'] = chart4.sum(axis=0)  #Add a new column for the total count of travel frequency  chart4['Grand Total'] = chart4.sum(axis=1)  #Show the table  chart4 |
| Out [2]: |  |
| In [3]: | import matplotlib.pyplot as plt  #Remove both Grand total values from row and column  chart4 = chart4.drop(index=['Grand Total'], columns=['Grand Total'])  #Create the pivot table as a bar chart with zorder = 2 so the gridlines wont block the charts  #Set the figsize to increase width to 15 inches wide and 5 inches tall  ax = chart4.plot(kind='bar', zorder=2, figsize=(15, 5))  #Set the range from 0 to 250 with 50 intervals  plt.yticks(range(0, 251, 50))  #Set the Y axis number limit to 250  plt.ylim(top=250)  #Add gridlines behind the bar chart by using zorder = 1  plt.grid(axis='y', zorder=1)  #Set chart title  plt.title('Relationship between Job Role and Travel Frequency')  #Set the x axis label  plt.xlabel('Types of Job Roles')  #Set the y axis label  plt.ylabel('Travel Counts')  #Get the heights of each bars using .get\_height() method  heights = [bar.get\_height() for bar in ax.patches]  #Add the values of each bars to the chart  #Use ha='center' to create a horizontal alignment for the values to be on the bar  #height + 5 to put the string slightly above the bars  for i, height in enumerate(heights):  ax.text(ax.patches[i].get\_x() + ax.patches[i].get\_width() / 2, height + 5, str(int(height)), ha='center')    #Create a legend showing the different travel frequency  #Use loc 'center right' to shift the legend to the right side of the chart  #Use bbox\_to\_anchor to shift the legend to the outside of the chart  ax.legend(loc='center right', bbox\_to\_anchor=(1.1, 0.3))  #Show the chart  plt.show() |
| Out [3]: | (Clearer picture below) |

**Table 4’s Output:**



**Table 4’s Chart:**

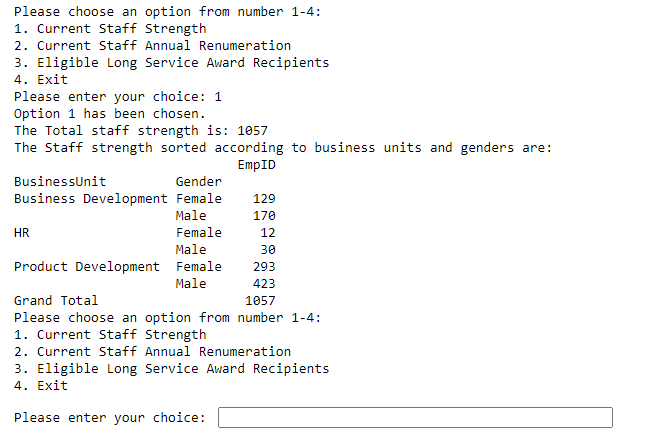


**Question 1c)**

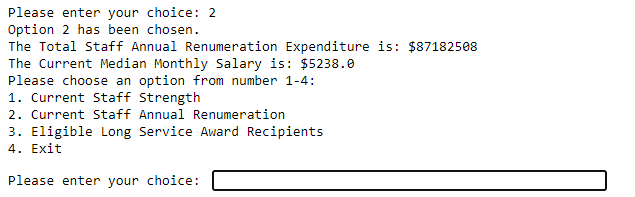
|  |  |
| --- | --- |
| In [1]: | import pandas as pd  import statistics  #Defining gba\_file to the .csv file filled with the GBA data  #r is to prevent python from reading \ as an escape character  gba\_file = r"C:\Users\zijin\OneDrive\Desktop\GBA.csv"  #Read the file using Pandas and read\_csv function  df = pd.read\_csv(gba\_file)  #Printing all the different options for the user to choose from.  while True:  print("Please choose an option from number 1-4: ")  print("1. Current Staff Strength")  print("2. Current Staff Annual Renumeration")  print("3. Eligible Long Service Award Recipients")  print("4. Exit")  # Prompt the user for input  user\_option = input(str("Please enter your choice: "))    #If user\_option is 1, we find the number of employees that stayed first  #Provide information of the current total staff strength, and a breakdown of staff strength by business unit and gender.  if user\_option == "1":  print("Option 1 has been chosen.")  employees\_stayed = df[df["LeftCompany"] == "No"]    #Calculate the total staff strength of those employees who stayed  total\_staff = len(employees\_stayed)    #Calculate staff strength by business unit and gender using pivot\_table method  employees\_grouped\_unit\_gender = employees\_stayed.pivot\_table(index=["BusinessUnit", "Gender"], values="EmpID", aggfunc=len, margins=True, margins\_name="Grand Total")    #Print the results  print(f"The Total staff strength is: {total\_staff}")  print("The Staff strength sorted according to business units and genders are:")  print(employees\_grouped\_unit\_gender)  #If user has chosen option 2, it will show "Option 2 has been chosen", followed by the staff renumeration and median salary  elif user\_option == "2":  print("Option 2 has been chosen.")    # Find the employees who are still in the company  employees\_stayed = df[df["LeftCompany"] == "No"]  #Find the sum of monthlypay with data extracted from excel  sum\_of\_monthlypay = employees\_stayed["MonthlyPay"].sum()  #Find the total yearly pay which is the total staff renumeration by multiplying the total monthly pay by 12  staff\_renumeration = sum\_of\_monthlypay\*12  #Find the median salary using .median() function  median\_salary = employees\_stayed["MonthlyPay"].median()  #Print the the total staff renumeration and median salary  print(f"The Total Staff Annual Renumeration Expenditure is: ${staff\_renumeration}")  print(f"The Current Median Monthly Salary is: ${median\_salary}")  #If user has chosen option 3, it will show "Option 3 has been chosen", followed by employees who are eligible for long service award  elif user\_option == "3":  print("Option 3 has been chosen.")  #Finding the eligible long service award recipients  #Employees must have serviced the company for 35 years or more  #Sort the employees' ages by descending order  #Show only values from column "EmpID", "JobRole", "Age"  long\_service\_employees = df[df["WorkingExperience"] >= 35].sort\_values("Age", ascending=False)[["EmpID", "JobRole", "Age"]]  #Print the employees who are eligible for long service award  print("Congratulations! The following employees are eligible for long service award:")  #Remove the indexing  print(long\_service\_employees.to\_string(index=False))  #Exit the program if option 4 is chosen.  elif user\_option == "4":  print("Option 4 has been chosen. Program terminated.")  break  else:  #Anything that's not option 1,2,3 or 4 will be invalid.  print(f"Invalid option! You have chosen {user\_option} which is not in options 1-4") |

**Outputs when different options are selected:**

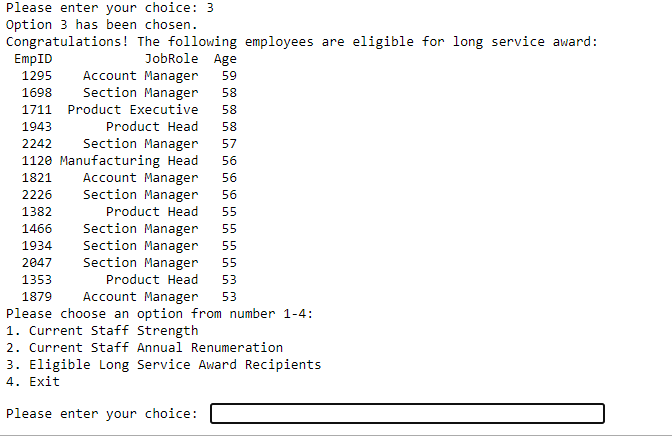
**Option 1:**

****

**Option 2:**

****

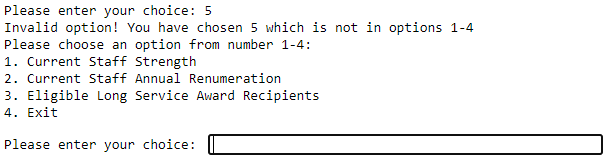
**Option 3:**

****

**Option 4:**

****

**Options not from 1-4:**

****