

**ANL252**

**Python For Data Analytics**

**Tutor-Marked Assignment**

**July 2022 Presentation**

Muhammad Azfar Bin Razak (H2170546)

Submission Date: 11th August 2022

Contents

[Question 1(a) 3](#_Toc111152731)

[Table.1: Summary of Count, Average Salary, Satisfaction Score 3](#_Toc111152732)

[Figure.1: Count of Gender Graph 3](#_Toc111152733)

[Table.2: Summary of Minority Count, Average Salary, Satisfaction Score 4](#_Toc111152734)

[Figure.2: Count of Minority Graph 4](#_Toc111152735)

[Question 1(b) 5](#_Toc111152736)

[Code 5](#_Toc111152737)

[**Output Cell 3:** Table 1 - Corresponding summarised table 8](#_Toc111152738)

[**Output Cell 4:** Pie Chart - Count of Gender 8](#_Toc111152739)

[**Output Cell 6:** Table 1 - Corresponding summarised table 9](#_Toc111152740)

[**Output Cell 7:** Bar Chart - Count of Minority Graph 9](#_Toc111152741)

[Question 1(c) 10](#_Toc111152742)

[Code 10](#_Toc111152743)

[**Output Cell 1:** Result of minimum, maximum and average length of service 11](#_Toc111152744)

[Question 1(d) 12](#_Toc111152745)

[Code 12](#_Toc111152746)

[**Output Cell 1:** Interactive user input 13](#_Toc111152747)

[References 14](#_Toc111152748)

# Question 1(a)

Analyzing the given data reveals that some cells have missing values and other cells have incorrect word format; hence, there are codes in Python programming to work with the aforementioned. Based on the given data, we divide the analysis into two sections, firstly with gender and then followed by minority.

The following data was extracted when the analysis was done with respect to gender.

|  |  |  |  |
| --- | --- | --- | --- |
| Table\_1 | | | |
| **Gender** | **Count** | **Average Salary** | **Satisfaction** |
| F | 144 | 67686.5 | 3.99 |
| M | 106 | 71063.8 | 3.88 |

## Table.1: Summary of Count, Average Salary, Satisfaction Score

## Figure.1: Count of Gender Graph

Figure.1 and table.1 shows a comparison between the gender of male and female by its count, average salary and satisfaction score. From figure 1, It shows that the given data has the most female staff members, and clearly shows by its visual separation that there is more than 50% of all the staff members compromises of female staff members.

From table 1, female staff members have a higher average satisfaction score than male staff members by 0.11. It is also worth noting that the average salary of male staff members is higher than that of female staff members by 3,377.30.

In conclusion, based the data presented in both table 1 and figure 1, having higher proportion of female to male staff ratio does not mean that the average salary would be higher but it is worth noting that even though female staff member has lower average salary, they still have the higher average satisfaction score

The following data was extracted when the analysis was done with respect to minority.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Table\_2 | | | | | |
| **Minority** | **Count** | **Female** | **Male** | **Average Salary** | **Satisfaction** |
| Yes | 25 | 12 | 13 | 73478.1 | 4 |
| No | 225 | 132 | 93 | 68634.1 | 3.94 |

## Table.2: Summary of Minority Count, Average Salary, Satisfaction Score

## Figure.2: Count of Minority Graph

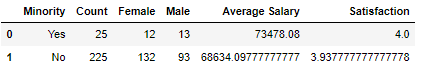
Figure 2 and table 2 shows the comparison of a minority and a non-minority by its gender, average salary and average satisfaction score. From figure 2, with its visual representation on the volume and size of the bar and that the given data, it has extremely few staff members that are members of a minority group.

From table 2, male staff members have a substantially higher minority ratio (13/106) than female staff members (12/144). It also shows that minorities are paid more than non-minorities by 4,844, and that minorities are also better pleased on average by 0.06.

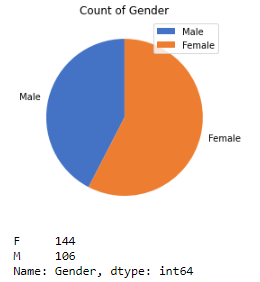
In conclusion, even though the proportion between both minority and non-minority group has a big difference, it is worth nothing that the satisfaction score is higher and has the higher average salary between them.

# Question 1(b)

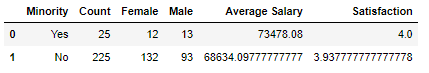
|  |  |
| --- | --- |
| Cell | Code |
| 1 | # Importing the necessary packages  import numpy as np  import pandas as pd  import matplotlib.pyplot as plt  # Reading csv file into df(Data Frame) and displaying the dataframe  df = pd.read\_csv("TMA\_Data.csv")  df |
| 2 | # Question 1(b) First Chart: Male vs Female Count + Average salary + satisfaction  # Converting the dataframe into numppy array  data = df.to\_numpy()  #Initializing variables to zero first in order to store stats  male\_count = 0; male\_salary = 0; male\_satisfaction = 0;  female\_count = 0; female\_salary = 0; female\_satisfaction = 0;  # Iterating over all of the rows(.shape[0]) of dataset using '0' for rows  for x in range(data.shape[0]):        if data[x,3] == 'M ': # if current record is male staff. \*Space after the letter 'M ' in order to match the data in csv file          male\_count += 1 # Increment variable male\_count by 1          male\_salary += data[x,10] # add salary of current male staff to variable male\_salary          male\_satisfaction += data[x,13] # add satisfaction score of current male staff to variable male\_satisfaction        elif data[x,3] == 'F': # Else if the current record belong to female staff instead          female\_count += 1 # Increment variable female\_count by 1          female\_salary += data[x,10] # add salary of current female staff to variable male\_salary          female\_satisfaction += data[x,13] # add satisfaction score of current female staff to variable male\_satisfaction  # Storing the table\_1 data and calculating the Average Salary and Average Satisfaction into list of list  table\_1 = [['M', male\_count, male\_salary/male\_count, male\_satisfaction/male\_count ],             ['F', female\_count, female\_salary/female\_count, female\_satisfaction/female\_count ]] |
| 3 | # Converting table\_1 into dataframe and displaying the corresponding summarised table  pd.DataFrame(table\_1, columns=['Gender', 'Count', 'Average Salary', 'Average Satisfaction']) |
| 4 | # Plotting and displaying the pie chart which includes the start angle, colours, labels and values  mylabel\_table1 = ['Male', 'Female']  mycolours\_table1 = ['#4472C4', '#ED7D31']  values = df['Gender'].value\_counts()  plt.pie([male\_count, female\_count], labels = mylabel\_table1, startangle = 90,         colors = mycolours\_table1)  plt.title('Count of Gender')  plt.legend()  plt.show()  print(values) |
| 5 | # Question 1(b) Second Chart: Minority Male + Female Count + Average salary + satisfaction  # Converting the dataframe into numppy array  data = df.to\_numpy()  #Initializing variables to zero first in order to store stats  minority\_yes\_count = 0; minority\_yes\_salary = 0; minority\_yes\_satisfaction = 0; minority\_yes\_female = 0; minority\_yes\_male = 0;  minority\_no\_count = 0; minority\_no\_salary = 0; minority\_no\_satisfaction = 0; minority\_no\_female = 0; minority\_no\_male = 0;  # Iterating over all of the rows(.shape[0]) of dataset using '0' for rows  for x in range(data.shape[0]):        if data[x,6].lower() == 'yes': # If current record is a minority staff. There is one 'yes' which is in lower case, workaround is using .lower().            if data[x,3] == 'F': # if current record is a minority & female staff              minority\_yes\_female += 1 # Increment female minority count by 1            elif data[x,3] == 'M ': # else if current record is a minority & male staff. \*Space after the letter 'M ' in order to match the data in csv file              minority\_yes\_male += 1 # Increment male minority count by 1            minority\_yes\_count += 1 # Increment minority yes count by 1          minority\_yes\_salary += data[x,10] # Add salary of current minority staff to variable minority\_yes\_salary          minority\_yes\_satisfaction += data[x,13] # Add satisfaction score of current minority staff to variable minority\_yes\_satisfaction        elif data[x,6].lower() == 'no': # else if current record is not a minority staff. \*There is one 'no' which is in lower case, workaround is using .lower().            if data[x,3] == 'F': # if current record is a not minority & female staff              minority\_no\_female += 1 # Increment female non-minority count by 1            elif data[x,3] == 'M ': # else if current record is not a minority & male staff. \*Space after the letter 'M ' in order to match the data in csv file              minority\_no\_male += 1 # Increment male non-minority count by 1            minority\_no\_count += 1 # Increment non-minority count by 1          minority\_no\_salary += data[x,10] # Add salary of current non-minority staff to variable minority\_no\_salary          minority\_no\_satisfaction += data[x,13] # Add satisfaction score of current non-minority staff to variable minority\_no\_satisfaction    # Storing the table\_2 data and calculating the minority and non-minority count by gender, calculation of it's average salary and average satisfaction score into list of list  table\_2 = np.array( [np.array(['Yes', minority\_yes\_count, minority\_yes\_female, minority\_yes\_male, minority\_yes\_salary/minority\_yes\_count, minority\_yes\_satisfaction/minority\_yes\_count ]),              np.array(['No', minority\_no\_count, minority\_no\_female, minority\_no\_male, minority\_no\_salary/minority\_no\_count, minority\_no\_satisfaction/minority\_no\_count ])]) |
| 6 | # Converting table\_2 into dataframe and displaying the corresponding summarised table  pd.DataFrame(table\_2, columns=['Minority', 'Count', 'Female', 'Male', 'Average Salary', 'Average Satisfaction']) |
| 7 | # Ploting and Displaying bar chart which includes labels, titles, font size, colour  plt.bar(['Yes', 'No'], [minority\_yes\_count, minority\_no\_count], width = 0.2, color = '#4472C4', linestyle = '--')  font1 = {'color':'blue', 'size':20}  font2 = {'color':'darkred', 'size':15}  plt.title('Count of Minority', fontdict = font1)  plt.xlabel("Minority Yes or No", fontdict = font2)  plt.ylabel("Count of Both Gender", fontdict = font2)  plt.grid(axis = 'y')  plt.show() |

****

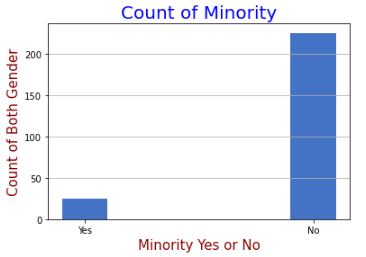
## **Output Cell 3:** Table 1 - Corresponding summarised table

****

## **Output Cell 4:** Pie Chart - Count of Gender



## **Output Cell 6:** Table 1 - Corresponding summarised table



## **Output Cell 7:** Bar Chart - Count of Minority Graph

# Question 1(c)

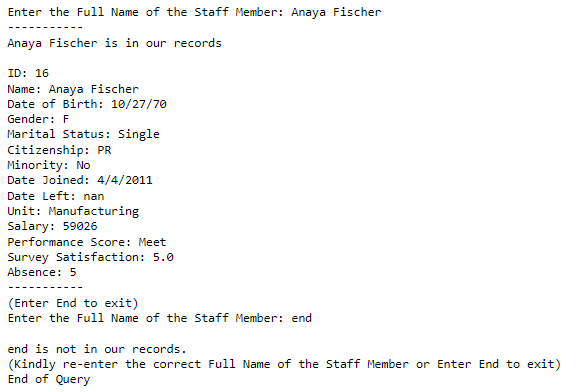
|  |  |
| --- | --- |
| Cell | Code |
| 1 | # Importing the necessary packages  import numpy as np  import pandas as pd  from datetime import date  import matplotlib.pyplot as plt  # Reading csv file into df(Data Frame)  df = pd.read\_csv("TMA\_Data.csv")  # defining a method to take two string dates to subtracts them and return the difference  def subtract\_dates(d1,d2):      d1 = d1.split('/') # splitting the first date of day, month, year      d2 = d2.split('/') # splitting the Second date of day, month, year      dt1 = date(int(d1[2]), int(d1[0]), int(d1[1])) # creating new date object for first date      dt2 = date(int(d2[2]), int(d2[0]), int(d2[1])) # creating new date object for second date      alpha = dt1 - dt2 # Formula to subtract both of the date object      return alpha.days/365 # Converting days to years and thereafter returning them to a calling function  # Filling up the missing Left Date records with 1/5/2022  df['LeftDate'] = df['LeftDate'].replace(np.nan, '1/5/2022')  new\_data = df.to\_numpy()  # Creating a list to store the length of service for all of the staff in records  ServiceDuration = []  for i in range(new\_data.shape[0]): # Iterating over all of the rows(.shape[0]) of dataset using '0' for rows      Join\_Date = new\_data[i,7] # Reading Join date of the current record      Left\_Date = new\_data[i,8] # Reading Left date of the current record after filling up missing Left Date      ServiceLength = subtract\_dates( Left\_Date, Join\_Date) # Calling the function of subtract\_dates method to subtract both the dates      ServiceDuration.append(ServiceLength) # Adding the calculated durations to the list    ServiceDuration = np.array(ServiceDuration) # converting to numpy array  # Displaying the minimum, maximum and average length of service, expressing in years, rounded to 1 decimal place.  print("The minimum length of service is ","{:.1f}".format(np.min(ServiceDuration)))  print("The maximum length of service is ","{:.1f}".format(np.max(ServiceDuration)))  print("The average length of service is ","{:.1f}".format(np.mean(ServiceDuration)))  # If required, peform the following code to produce the updated csv file which includes the Left Date that was added  # df.to\_csv('output.csv') |

****

## **Output Cell 1:** Result of minimum, maximum and average length of service

# Question 1(d)

|  |  |
| --- | --- |
| Cell | Code |
| 1 | # Importing the necessary packages  import numpy as np  import pandas as pd  from datetime import date  import matplotlib.pyplot as plt  # Reading csv file into df(Data Frame)  df = pd.read\_csv("TMA\_Data.csv")  data = df.to\_numpy() # Converting the dataframe into numppy array  user\_input = '' # Initializing an empty string to store user input when query  while user\_input.lower() != 'end': # Loops until user enters End to end the query        user\_input = input('Enter the Full Name of the Staff Member: ') # Taking user input if a particular person is in the organization      check\_name = False; # the variable was define as false to keep track if the record was found in the records data or not        for i in range(data.shape[0]): # Iterating over all of the rows(.shape[0]) of dataset using '0' for rows          if data[i,1] == user\_input: # If staff name in the record is similar to the user's input              # Displaying the query output of the records if it is similar user's input              print('-----------')              print(user\_input, 'is in our records')              print()              print("ID:", data[i,0],"\nName:",data[i,1] ,"\nDate of Birth:",data[i,2],"\nGender:",data[i,3] ,"\nMarital Status:",data[i,4] ,"\nCitizenship:",data[i,5], "\nMinority:",data[i,6],"\nDate Joined:",data[i,7],"\nDate Left:",data[i,8],"\nUnit:",data[i,9],"\nSalary:",data[i,10],"\nPerformance Score:",data[i,11],"\nSurvey Satisfaction:",data[i,12],"\nAbsence:",data[i,13])              print('-----------')              print('(Enter End to exit)')              check\_name = True # A setting check variable equals to true        if check\_name == False: # If user's input was not found in the records, print the following          # Displaying the query output of the records if it is not similar user's input          print()          print(user\_input, 'is not in our records.')          print('(Kindly re-enter the correct Full Name of the Staff Member or Enter End to exit)')    print('End of Query') |

****

## **Output Cell 1:** Interactive user input

# References

GEEKSFORGEEKS. (n.d). Creating Pandas dataframe using list of lists

Available: https://www.geeksforgeeks.org/creating-pandas-dataframe-using-list-of-lists/amp/

Stackoverflow. (n.d) Convert pandas dataframe to NumPy array

Available: https://stackoverflow.com/questions/13187778/convert-pandas-dataframe-to-numpy-array

Stackoverflow. (n.d). How to calculate number of days between two given dates

Available: https://stackoverflow.com/questions/151199/how-to-calculate-number-of-days-between-two-given-dates

w3schools. (n.d). Matplotlib Bars

Available: https://www.w3schools.com/python/matplotlib\_bars.asp

w3schools. (n.d). Matplotlib Pie Charts

Available: https://www.w3schools.com/python/matplotlib\_pie\_charts.asp

w3schools. (n.d). NumPy Array Shape

Available: https://www.w3schools.com/python/numpy/numpy\_array\_shape.asp

w3schools. (n.d). Pandas Read CSV

Available: https://www.w3schools.com/python/pandas/pandas\_csv.asp

w3schools. (n.d). Python Datetime

Available: https://www.w3schools.com/python/python\_datetime.asp

w3schools. (n.d). Python While Loops

Available: https://www.w3schools.com/python/python\_while\_loops.asp