

**ANL252**

**Python For Data Analytics**

**Tutor-Marked Assignment**

**Submitted by:**

|  |  |
| --- | --- |
| **Name** | **PI No.** |
| Shawn Tan Jin Chuan | J2172224 |

**Tutorial Group:** T09

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

**Submission Date:** 14 August 2022

# Question 1 (a)

The TMA Data provided contains 15 columns of variables and 250 records of employees within a company.

The 1st column is the ID column. This tells us the unique ID number being generated and assigned to each employee. This helps to better retrieve information of certain employees just by stating the ID number.

The 2nd column is the Staff column. This column displays the name of every employee within the organization. This could be potentially renamed as ‘Staff Name’ to ensure clarity of what data is being displayed and stored under the column.

The 3rd column is the Birth Year column. This column displays the date of birth of all the employees. Since it is stored as a whole date format (mm/dd/yyyy), the column could be renamed as ‘Birth Date’ to ensure accuracy of data being displayed.

The 4th column is the Gender column. This shows the employees’ gender with M representing Male, and F representing Female.

The 5th column is the Marital column. This shows the marital status of employees.

The 6th column is the Citizenship column. This tells us whether the employee holds a citizenship, is a permanent resident residing in the country, or a non-resident.

The 7th column is the Minority column. This is a yes or no column with yes representing that the employee belongs to the minority, and no representing the employee does not belong to the minority.

The 8th column is the Join Date column. This tells us when every employee first joined the company.

The 9th column is the Left Date column. This tells us when every employee left the company. If the field is empty, it meant that the employee is still in the company and has not left.

The 10th column is the Unit column. This tells us which department or unit the employee is from within the organization.

The 11th column is the Salary column. This shows the annual salary of all the employees within the company. To prevent confusion, the column could specify whether or not the figures are referring to monthly or annual salary.

The 12th column is the Performance column. This tells us how well each employee performed. Employees are being categorized under Exceed, Meet, To Improve, and PIP for those that are under the Performance Improvement Plan.

The 13th column is the Survey column. This shows the engagement survey results of each employee being measured on a scale of 1 to 5.

The 14th column is the Satisfaction column. This shows the satisfactory level of employees working within the organization and is being measured on a scale of 1 to 5.

The last column is the Absence column. This shows the employees’ total number of days absent in the previous year.

Bar Chart

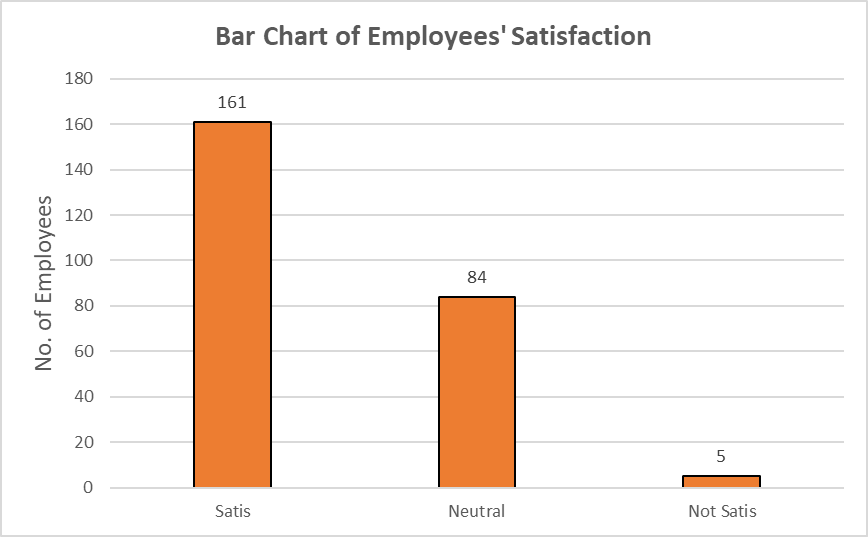
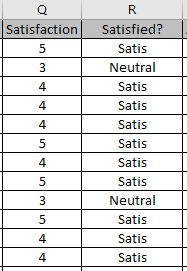


Figure 1: Bar Chart

The bar chart was created based on the satisfaction level of employees. This is under the assumption that the survey was designed in a way where a score level of 1 or 2 means that the employee is not satisfied, a score level of 3 means that the employee is neutral on the statement, and a score level of 4 or 5 means that the employee is satisfied. A new column was created to state the satisfactory status of each employee. Rather than looking at 5 different score numbers, it is easier to group the employees based on the score that they gave.



The formula used:

Figure : New Satisfactory Status Column

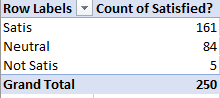


Figure 3: Summarized Table for Bar Chart

The bar chart shows that out of 250 employees, 161 are satisfied, 84 are neutral, and 5 are not satisfied. With this chart, it is easy to identify the group of employees who are unsatisfied. The organization could work on understanding these employees as a priority to help improve their experience. They could also shift their focus to boost the morale of the 84 employees who gave a neutral score. This could improve the overall performance of the company if the employees are satisfied working in the organization.

Histogram

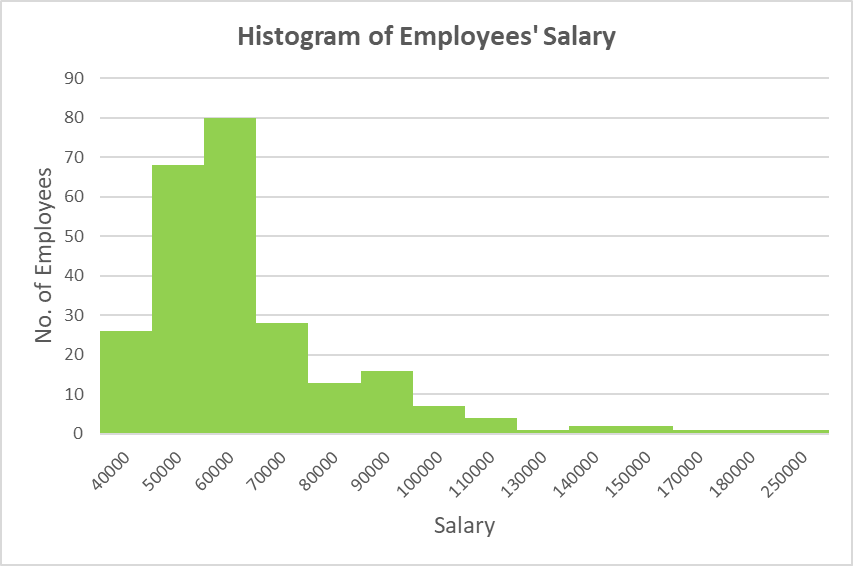


Figure 4: Histogram

The histogram was created from the salary of all the employees within the organization. It shows the summary of all the employees’ annual salary range at one glance. To plot this histogram, a new column called Rounded Salary was introduced. The salary was being rounded off to the nearest 10,000 to ensure accurate binning of the data.

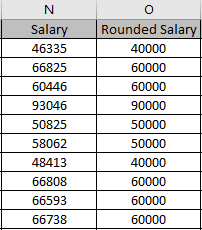
  
The formula used:

Figure : New Rounded Off Salary Column

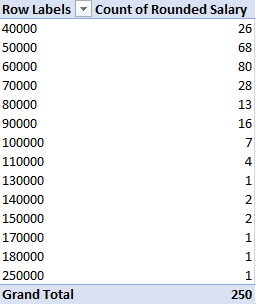


Figure 6: Summarized Table for Histogram

The histogram will be able to tell the organization the highest frequency of employees’ salary range and gauge better how much are money is going out of the company. From figure 4, it is evident that the salary distribution is right skewed. This meant that a higher frequency of the employees is drawing a salary nearer to the left side (lower salary). Most of the employees are currently earning between $50000 to $69999 as seen from the histogram.

# Question 1 (b)

Bar Chart

#import libraries

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

#read data

tmadf = pd.read\_csv('TMA\_Data.csv')

#print columns & datatypes

print(tmadf.columns)

tmadf.dtypes

#replace and group scores to status

tmadf['Satisfaction'] = tmadf['Satisfaction'].apply(str)

tmadf['Satisfaction'].replace("1", "Not Satis", inplace = True)

tmadf['Satisfaction'].replace("2", "Not Satis", inplace = True)

tmadf['Satisfaction'].replace("3", "Neutral", inplace = True)

tmadf['Satisfaction'].replace("4", "Satis", inplace = True)

tmadf['Satisfaction'].replace("5", "Satis", inplace = True)

#create x axis

satis = ['Satis', 'Neutral', 'Not Satis']

x\_axis = np.arange(len(satis))

print(x\_axis)

#assign status

y\_satis = tmadf['Satisfaction'] == 'Satis'

y\_neutral = tmadf['Satisfaction'] == 'Neutral'

y\_notsatis = tmadf['Satisfaction'] == 'Not Satis'

#create y axis

y\_axis = [y\_satis.sum(),y\_neutral.sum(),y\_notsatis.sum()]

y\_axis

#plot bar chart

plt.bar(x\_axis,y\_axis,color="tan", edgecolor="black")

plt.title("Bar Chart of Employees' Satisfaction \n", fontsize = 20)

plt.xticks(x\_axis,satis)

plt.ylabel("No. of Employees", fontsize = 14)

for x,y in zip(x\_axis,y\_axis):

label = y

plt.annotate(label, #text

(x,y), #coordinates to position the label

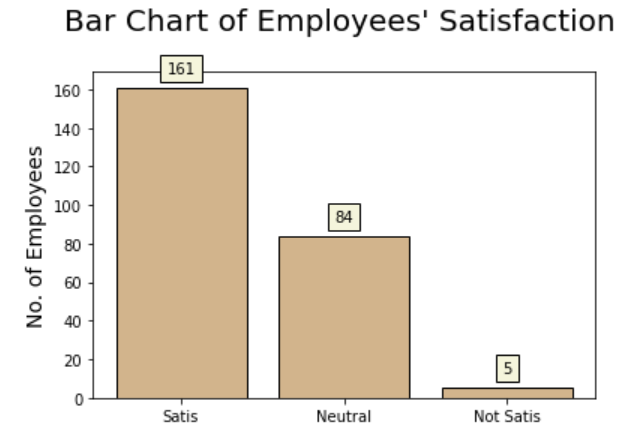
textcoords="offset points", #position text

xytext=(0,10), #distance from text to points (x,y)

ha='center',bbox = dict(facecolor = 'beige'))

plt.show()

**RESULTS:**



Histogram:

#check salary data

salary\_data = list(tmadf['Salary'])

print(salary\_data)

#print min, max, mean salary

print('Lowest Annual Salary: ', tmadf['Salary'].min())

print('Highest Annual Salary: ', tmadf['Salary'].max())

print('Average Annual Salary: ', tmadf['Salary'].mean())

#create bin

salary\_bin = pd.cut(tmadf['Salary'], bins=np.linspace(0, 280000, 29),labels=[0,10000,20000,30000,40000,50000,60000,70000,80000,90000,100000,110000,120000,130000,140000,150000,160000,170000,180000,190000,200000,210000,220000,230000,240000,250000,260000,270000])

print(salary\_bin)

#create histogram frequency table

import collections

from collections import Counter

c=Counter(salary\_bin)

hist\_data = c.values()

hist\_bin = c.keys()

hist\_freq = pd.DataFrame({'Bins':hist\_bin, 'Frequency':hist\_data})

print(hist\_freq)

#plot histogram

bin\_size = np.arange(0, 280000, 10000)

plt.figure(figsize=(8,8), dpi=90)

plt.hist(salary\_bin, bins=bin\_size, align="mid", color="yellowgreen")

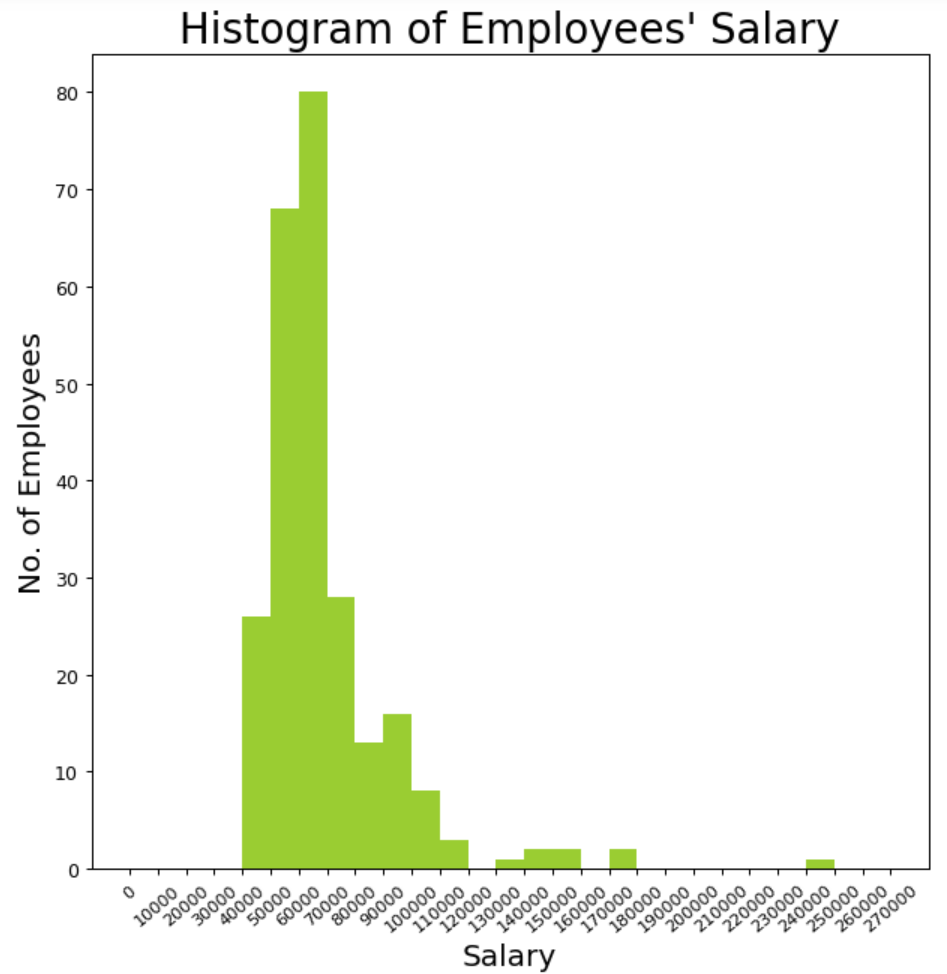
plt.title("Histogram of Employees' Salary", fontsize=22)

plt.xlabel("Salary", fontsize=16)

plt.ylabel("No. of Employees", fontsize=16)

plt.xticks(bin\_size,labels=bin\_size, rotation=40, size=9)

plt.show()

**RESULTS:**  


# Question 1 (c)

#print df

print(tmadf['JoinDate'])

print('---------------')

print(tmadf['LeftDate'])

#fill empty rows with 1st May 2022

tmadf['LeftDate'] = tmadf['LeftDate'].fillna('5/1/2022')

print(tmadf['LeftDate'])

#convert object to datetime format

tmadf[['JoinDate','LeftDate']] = tmadf[['JoinDate','LeftDate']].apply(pd.to\_datetime)

#calculate difference between dates in days and assign to Service Lenth df

#tmadf['ServiceLength'] = round(((tmadf['LeftDate'] - tmadf['JoinDate']) / np.timedelta64(1, 'D'))/365,1)

tmadf['ServiceLength'] = ((tmadf['LeftDate'] - tmadf['JoinDate']) / np.timedelta64(1, 'D'))/365

#view updated df

print(tmadf['ServiceLength'])

#round off values

min\_sl = round(tmadf['ServiceLength'].min(),1)

max\_sl = round(tmadf['ServiceLength'].max(),1)

avg\_sl = round(tmadf['ServiceLength'].mean(),1)

print('Minimum Service Length: ', min\_sl)

print('Maximum Service Length: ', max\_sl)

print('Average Service Length: ', avg\_sl)

#answer

print(f'(c) For length of service of all the staff, the minimum is {min\_sl} year(s), the maximum is {max\_sl} year(s), and the average is {avg\_sl} year(s).')

# Question 1 (d)

#convert df to list

stafflist = tmadf['Staff'].values.tolist()

print(stafflist)

#check staff program

valid\_input = False

while valid\_input == False:

try:

check\_staff = str(input("Check Staff (Y/N):"))

if check\_staff == "Y" or check\_staff == "y":

enter\_name = str(input("Please Enter Name: "))

if enter\_name in stafflist:

print(f"{enter\_name} is a staff!")

elif enter\_name not in stafflist:

print(f"{enter\_name} is not a staff!")

elif check\_staff == "N" or check\_staff == "n":

valid\_input = True

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

print("Thanks! Exiting Program.")

else:

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

print("Please Try Again")

except ValueError:

print("Wrong")