

**ANL252**

**Python for Data Analytics**

# **Tutor-Marked Assignment**

# **July 2022 Presentation**

|  |  |
| --- | --- |
| **Name:** | **Tng Ting Xu** |
| **PI Number:** | **K2181584** |
| **Date of Submission:** | **14 August 2022** |

**(a) Chart 1 (Excel Output)**

|  |  |  |
| --- | --- | --- |
| **Category** | **Female / Married / Majority** | **Male / Single / Minority** |
| **Gender** | $ 67,686.53 | $ 71,063.81 |
| **Marital Status** | $ 71,827.85 | $ 68,702.28 |
| **Racial / Ethnic** | $ 68,634.10 | $ 73,478.08 |

**(a) Chart 1 (Excel Output)**

**Insights:**

Analysing the data from the perspective of a Human Resource (HR) manager, the results presented above can be helpful in determining if the organisation practices income equality or if income inequality through discriminatory practices exist in the workplace. From the results, we can observe that certain inequalities exist.

Firstly, although the organisation has more female staff (Female: 144, Male: 106), the mean salary for male staff ($71,063.81) is significantly higher than that of female staff ($67,686.53). This is despite a high outlying salary amount of $250,000 paid to a female staff. This provides evidence that there is a gender pay gap prevalent within the organisation which could be due to various factors such as discriminatory hiring/salary practices or the need for females to take time out of the workforce to start a family which could negatively impact their career progression when they return eventually.

Secondly, there is also a slight difference in the mean salary between Married ($71,827.85) and Single ($68,702.28) staff. This difference can be attributed to the fact that married individuals tend to be older and have more working experience and thus command a higher salary as compared to single individuals who are likely to have just started their career and thus command a lower salary.

Lastly, there is a significant difference in the mean salary between staff belonging to the Racial /Ethnic Majority ($68,634.10) and staff belonging to the Racial/Ethnic Minority ($73,478.08). The organisation has a huge disparity between majorities and minorities (Majority: 225, Minority: 25). A plausible reason for this could be due to people having to immigrate out from their home country in search for better paying jobs and opportunities. These people might become part of the minority in the country that they are working in, despite commanding a higher average salary as an expatriate.

**(b) Chart 1 (Python Codes)**

*# Import Libraries*

import pandas as pd

import numpy as np

from matplotlib import pyplot as plt

*# States the variables on the X and Y axis*

Labels = ['Gender','Marital Status','Racial/Ethnic']

Yblue = [67686, 71827, 68634]

Yorange = [71063, 68702, 73478]

*# Expands chart size*

fig, ax = plt.subplots(1, figsize=(16, 8))

*# Creates a range or array of values on the x axis*

X\_axis = np.arange(len(Labels))

plt.xticks(X\_axis, Labels)

*# Plots a bar chart for both Bar Charts X and Y and plots legend*

plt.bar(X\_axis - 0.2, Yblue, 0.4, label = 'Female / Married / Majority')

plt.bar(X\_axis + 0.2, Yorange, 0.4, label = 'Male / Single / Minority')

*# Change the current limits of both x and y axis*

plt.ylim(61000,76000)

plt.xlim(-0.7, 2.7)

*# Creates the data labels*

plt.ylabel("Mean Salary ($)", fontweight='bold', fontsize=13, labelpad=10, verticalalignment='center')

plt.xlabel("Categories", fontweight='bold', fontsize=13, labelpad=10, horizontalalignment='center')

plt.title("Salary Equality", fontweight='bold', fontsize=15, horizontalalignment='center')

*# Creates Grids*

ax.set\_axisbelow(True)

ax.yaxis.grid(color='grey', linestyle='dashed', alpha=0.8)

*# Show the legend and graph*

plt.legend(ncol = 4, bbox\_to\_anchor=([1, 1.05, 0, 0]), frameon = False)

plt.show()

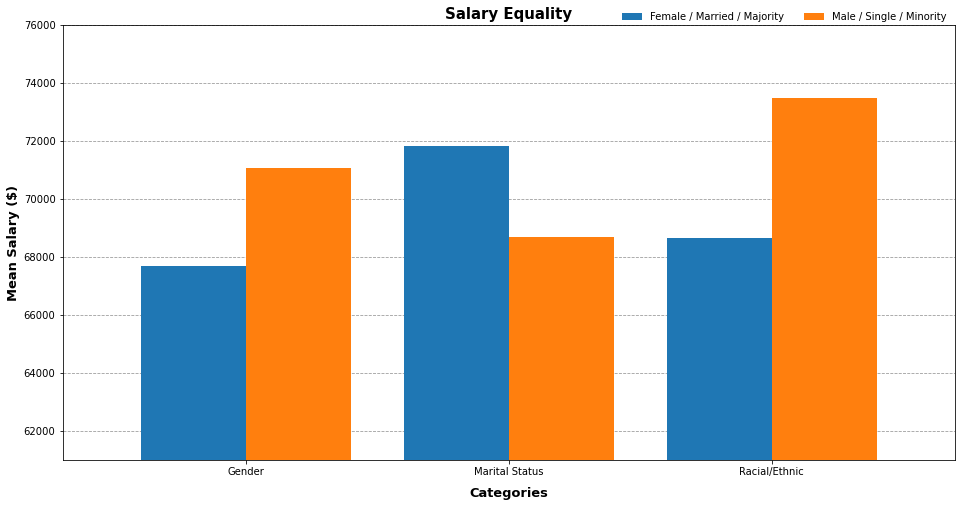
*# Next we create the datatable by importing tabulate*

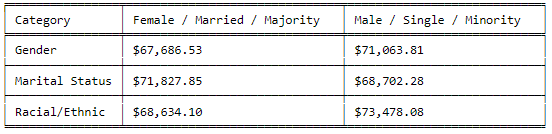
from tabulate import tabulate

*# We create the table using a dictionary and print it*

info = {'Category': ['Gender', 'Marital Status', 'Racial/Ethnic'], 'Female / Married / Majority': ['$67,686.53', '$71,827.85', '$68,634.10'], 'Male / Single / Minority': ['$71,063.81', '$68,702.28', '$73,478.08']}

print(tabulate(info, headers='keys', tablefmt='fancy\_grid'))

**(b) Chart 1 (Python Chart Output)**



**(a) Chart 2 (Excel Output)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Department** | **Minimum** | **Midpoint** | **Maximum** | **Absence** |
| Manufacturing | $ 45,046.00 | $ 64,356.50 | $ 83,667.00 | 10 |
| Admin | $ 51,920.00 | $ 79,143.50 | $ 106,367.00 | 8 |
| Engineering | $ 77,692.00 | $ 93,339.50 | $ 108,987.00 | 9 |
| IT | $ 50,178.00 | $ 114,089.00 | $ 178,000.00 | 11 |
| Sales | $ 55,875.00 | $ 117,937.50 | $ 180,000.00 | 13 |

**(a) Chart 2 (Excel Output)**

**Insights:**

The data plot presents the salary range of staff from the five different departments and the average days of absence per department.

Analysis of the data shows that staff from both the IT and Sales Departments generally enjoy higher salaries and have the widest salary range, from $50,178 - $178,000 and $55,875 - $180,000 respectively. This suggest that various roles / levels commanding different paygrades exist within these departments. For example, experienced roles such as Software Engineers vs entry level roles such as IT Technician, and that staff from the Sales department have commission-based salaries.

Conversely, staff from both the Manufacturing and Admin Departments have generally lower salaries, ranging from $45,046 - $83,667 and $51,920 - $106,367 respectively. The Manufacturing, Admin and Engineering Departments also have a narrower salary range, indicating that staff in these departments command around the same salaries. This could indicate the lack of specialised roles for career progression within these departments or that roles within these departments are highly similar. For example, Manual workers typically command a lower paygrade and makes up the majority of staff in a manufacturing department.

Additionally based on the line chart, we can observe that there is a trend of increasing days of absence with generally higher salary range. This is except for the Manufacturing department where it has an unusually high average days of absence of 10 days despite having the lowest salary range. This high rate of absence could be attributed to the need for manual labour during the manufacturing process which can cause fatigue, resulting to higher rate of absence.

**(b) Chart 2 (Python Codes)**

*# Import libraries*

import pandas as pd

import matplotlib.pyplot as plt

import numpy as np

*# Reads data from file*

PATH = r"C:\Users\Tng Ting Xu\Desktop\SUSS\Y1S2\ANL252\TMA\TMA\_Data.csv"

DF = pd.read\_csv(PATH)

*# Defines the different Departments*

Units = ['Manufacturing', 'Admin ', 'Engineering', 'IT', 'Sales']

*# Convert lists to dict*

my\_dict = {'Unit': Units,

'maximum\_salary' : [],

'midpoint\_salary': [],

'minimum\_salary' : [],

'Absence': [],

'bar\_val' : []

}

*# Manipulate/calculate values and store into dictionary*

for unit in Units:

df\_u = DF[DF["Unit"]==unit]

max\_val = df\_u["Salary"].max()

min\_val = df\_u["Salary"].min()

full\_bar = max\_val-min\_val

half\_bar = int(round((max\_val-min\_val)/2,0))

midpoint\_val = min\_val + half\_bar

avg\_absence = int(round(df\_u["Absence"].mean(),0))

my\_dict["maximum\_salary"] = my\_dict["maximum\_salary"] + [max\_val]

my\_dict["midpoint\_salary"] = my\_dict["midpoint\_salary"] + [min\_val]

my\_dict["minimum\_salary"] = my\_dict["minimum\_salary"] + [midpoint\_val]

my\_dict["Absence"] = my\_dict["Absence"] + [avg\_absence]

my\_dict["bar\_val"] = my\_dict["bar\_val"] + [half\_bar]

*# Converts dict to Dataframe / df*

result\_df = pd.DataFrame(my\_dict)

*# Creates a blank canvas*

fig, ax = plt.subplots(1, figsize=(16, 8))

*# Next, we plot 2 bars, first to offset the midpoint salary, the second to offset the minimum salary*

bar1 = plt.bar(result\_df.index, result\_df['bar\_val'], bottom = result\_df['midpoint\_salary'], color = '#4472C4', edgecolor = "black", linewidth = 2, width = 0.2)

bar2 = plt.bar(result\_df.index, result\_df['bar\_val'], bottom = result\_df['minimum\_salary'], color = '#4472C4', edgecolor = "black", linewidth = 2, width = 0.2)

*# Creates the axis ticks by using index to return the position of the element*

plt.xticks(result\_df.index , labels = Units)

plt.yticks([0]+list(np.linspace(20000,200000,10)))

*# Creates the data labels*

plt.ylabel("Salary ($)", fontweight='bold', fontsize=13, labelpad=10, verticalalignment='center')

plt.xlabel("Departments", fontweight='bold', fontsize='13', labelpad=10, horizontalalignment='center')

plt.title("Salary Range & Average Days of Absence Across Departments", fontweight='bold', fontsize='15', horizontalalignment='center')

*# Creates the secondary axis*

*# Clone the x-axis*

ax2 = ax.twinx()

*# Plots line and points*

ax2.plot(Units,my\_dict["Absence"], color='orange', linewidth=3)

ax2.scatter(Units,my\_dict["Absence"], color='orange')

*# Define the y-axis values and y axis*

plt.yticks(list(np.linspace(0,14,8)))

plt.ylabel("Days of Absence", fontweight='bold', fontsize=13, labelpad=10)

*# Creates Grids*

ax.set\_axisbelow(True)

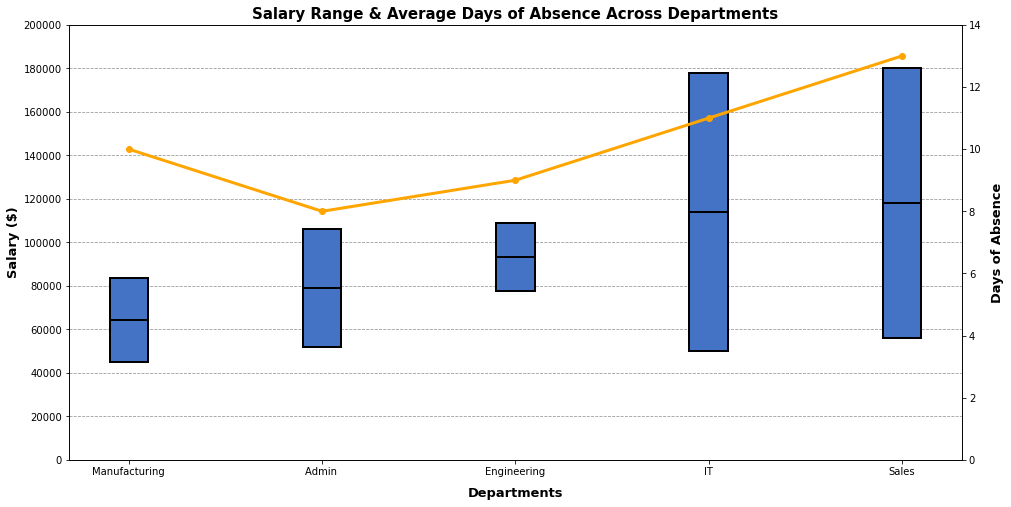
ax.yaxis.grid(color='grey', linestyle='dashed', alpha=0.8)

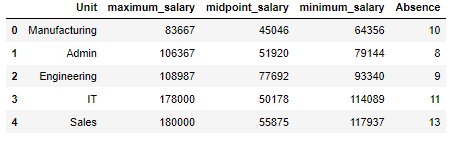
*# Shows the plot*

plt.show()

*# Creates a table to show the results*

result\_df.iloc[:,:-1]**(b) Chart 2 (Python Chart Output)**

****



**(C)**

*# Fills the LeftDate Column with default date*

default\_date = "2022-05-01"

DF["LeftDate"] = DF["LeftDate"].fillna(pd.to\_datetime(default\_date))

*# Converts both columns to datetime format for easier manipulation*

DF["LeftDate"] = pd.to\_datetime(DF["LeftDate"])

DF["JoinDate"] = pd.to\_datetime(DF["JoinDate"])

*# Calculates the length of service*

DF["Length\_of\_service"] = DF["LeftDate"] - DF["JoinDate"]

*# Calculates & prints the results*

print("Length of service info:")

print("Minimum :", round(DF["Length\_of\_service"].min().days/365 ,1), "Years")

print("Maximum :", round(DF["Length\_of\_service"].max().days/365 ,1), "Years")

print("Average :", round(DF["Length\_of\_service"].mean().days/365,1), "Years")

**Answer:**

|  |  |
| --- | --- |
| **Length of Service Info:** | **Years (1 d.p)** |
| Minimum | 0.1 |
| Maximum | 16.3 |
| Average | 6.8 |

d)

*# Import libraries*

import pandas as pd

*# Reads data from file and create dictionary since we are using key-value pairs*

PATH = r"C:\Users\Tng Ting Xu\Desktop\SUSS\Y1S2\ANL252\TMA\TMA\_Data.csv"

df = pd.read\_csv(PATH)

mydict = df.set\_index('Staff')['PerformanceScore'].to\_dict()

*# Checks if user wants to check database*

while True:

input\_check = str(input("Would you like to check the database? (Yes/No): "))

*# For user to input name*

if input\_check == "Yes":

input\_name = str(input("Please enter the name of staff you want to check: "))

*# If user's name is found, prompts second question*

if input\_name in df['Staff'].values:

print(input\_name, "is in the organisation's records!")

input\_perfcheck = str(input("Would you like to know his/her performance score? (Yes/No): "))

*# Pulls value from dictionary using key*

if input\_perfcheck == "Yes":

print(input\_name, "obtained a performance grade of:", mydict.get(input\_name))

input\_checkagain = str(input("Would you like to search another?: "))

if input\_checkagain == "Yes":

print("Please start a new query")

else:

print("Goodbye")

break

*# If second question is denied, break*

else:

print("Goodbye")

break

*# If user's name is not found in dataset*

else:

print(input\_name, "is NOT in the organization's records!")

*# Breaks when user does not want to check database*

if input\_check == "No":

input\_name = print("Goodbye")

break