

**ANL252**

**Python for Data Analytics**

**Tutor-Marked Assignment**

**July 2022**

**Submitted by:**

|  |  |
| --- | --- |
| **Name** | **PI No.** |
| **Vivian Wee Si Min** | **E2010380** |

**Tutorial Group: ­­­­­­­­­­­­ T 09**

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

**Submission Date: 14/08/2022**

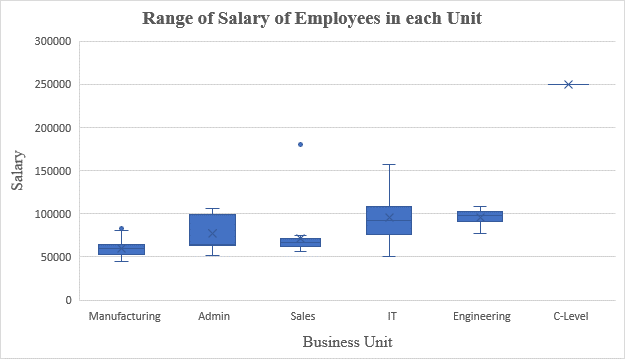
**Question 1 (a)**

The given data contains a record of 250 staffs from a company. They were analyzed, and two charts were created using Excel. The first chart drawn up compares the amount of salary the staffs from each business unit is earning, a box plot was chosen in this case as box plots can give a good indication of how the values in the dataset are spread out. The salary of the staffs in the company is a continuous and quantitative data. Continuous data are data that can be measured on an infinite scale and can be any value between two numbers no matter how small or big; and it can also be in decimal numbers *(Continuous Data, n.d.)*.

The first chart drawn up is a box plot that shows a summary of salaries of staff in each unit.

*\*Assumption: We will assume that the salaries given in the excel is based on an annual basis.*

First Chart



*Figure 1: Box plot of Salary in each business unit*

Figure 1 suggests that the range of salary is the highest in the IT unit and the staff earning the lowest salary is in the manufacturing unit. The interquartile range of salary (IQR) in IT unit is higher than the Sales unit and Manufacturing unit. It can be noted from the chart that the Sales unit has one outlier. Salaries in Sales unit is less than the salaries in IT and Engineering unit but one staff is getting much higher salary than the IT and Engineering unit. The data is not labeled in the box plot as it would make the box plot look messy. Therefore, the data is not labeled in the chart. It is also noted from the chart that the salaries in the Admin unit is higher than the Manufacturing unit and the Sales unit but comparable to that in IT and Engineering unit. The IQR for salaries in Engineering unit is narrow suggesting that there is not much variation in salaries in Engineering unlike the IT and Admin units.

Corresponding table for Figure 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Unit** | **Min Salary** | **1st Quartile** | **2nd Quartile** | **3rd Quartile** | **Max Salary** |
| Admin | 51920 | 63001.5 | 64520 | 96198.5 | 106367 |
| C-Level | 250000 | 250000 | 250000 | 250000 | 250000 |
| Engineering | 77692 | 93090.75 | 97470 | 101003.25 | 108987 |
| IT | 50178 | 76029 | 92328 | 107226 | 178000 |
| Manufacturing | 45046 | 53018 | 59472 | 64246 | 83667 |
| Sales | 55875 | 61817.75 | 67029.5 | 71615 | 180000 |

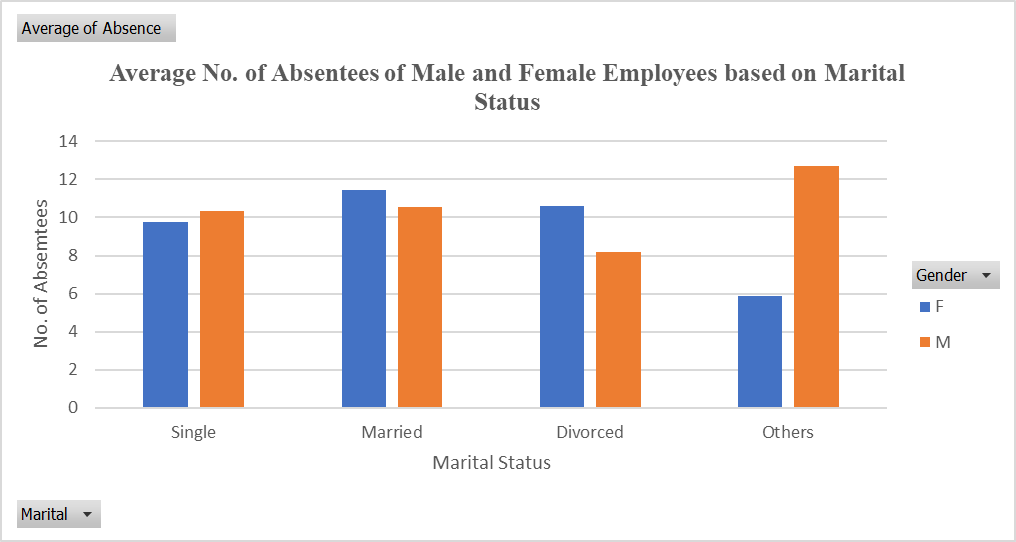
*Table 1*

Table 1 shows the corresponding table for the above box plot. From the table, we can see that the staff with the lowest salary comes from the Manufacturing unit; earning about $45,046 a year, and the staff earning the highest salary is from the C-level unit, earning about $250,000 a year. We can also see that there is one outlier in the Sales unit earning about $180,000 a year. The IQR for the Admin and IT unit is also much bigger as compared to the rest of the units with little variations in salary.

Second Chart

The second chart that is drawn up is a clustered bar graph; comparing the numbers of absentees based on the marital status of the staffs. Here, I am trying to see if the different marital status and gender impacts the number of times the staffs have been absent from work.

The number of absentees here is a discrete data. A discrete data is a count that involves only integers and cannot be divided into smaller parts *(Discrete vs Continuous Data, n.d.)*



*Figure 2: Clustered bar of No. of Absentees based on Marital Status*

Figure 2 above shows the average number of absentees based on the different marital statuses and gender of the company. The graph shows that there have been more married females that have been absent from work as compared to other females of other marital statuses. It also shows that there are more males under the “Others” status that have been absent from work as compared to other males from other marital status.

We can see that females classified under “Others” have been the least absent from work and the male divorcees have been the least absent from work. However, if we were to look at the average number of staffs that have been absent from work the most (from both genders), we see that the married couples in the company are absent the most. I am assuming that they may have family commitments and had to take more time off work as compared to the single and divorced staffs.

Corresponding table for Figure 2

|  |  |  |
| --- | --- | --- |
| **Row Labels** | **F** | **M** |
| Single | 9.774194 | 10.346154 |
| Married | 11.448276 | 10.527778 |
| Divorced | 10.600000 | 8.181818 |
| Others | 5.888889 | 12.714286 |

*Table 2*

From the above table, we can see that the female classified under “Others” have been the least absent from work at an average of 5.9 times and that the male divorcees have been the least absent from work at an average of 8.2 times. More married females have been absent from work at an average of 11.4 times as compared to other females, and more males under “Others” have been absent from work at an average of 12.7 times.

**Question 1 (b)**

Two charts shown in part(a) using python:

Chart 1 – Bot Plot:

Chart, box and whisker chart

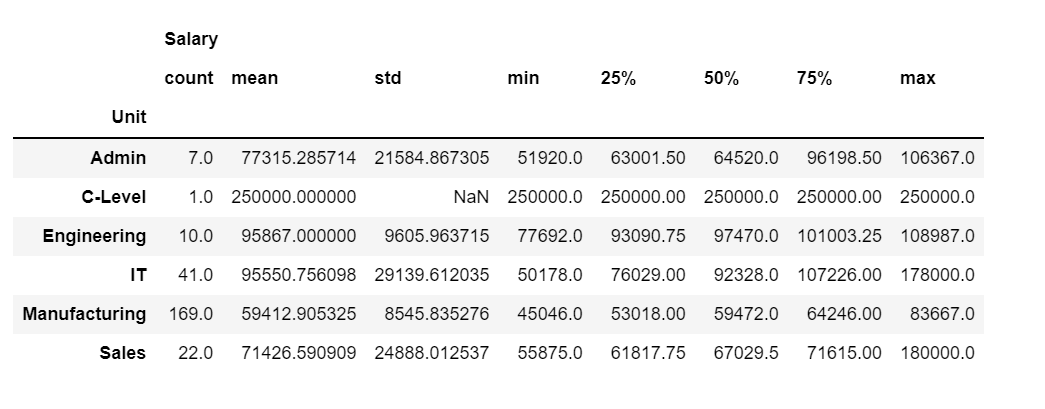
Description automatically generated

*Figure 3: Box plot of Salary in each business unit*

The codes for the above charts are given below:

|  |
| --- |
| # Import numpy library as np  import numpy as np  # Import pandas library as pd  import pandas as pd  # Import matplotlib import pyplot as plt  from matplotlib import pyplot as plt  # Import the library seaborn as snn  import seaborn as sns  # Import matplotlib as mtpl  import matplotlib as mtpl  # Adopt plotting style. the seaborn style looks good so I am using seaborn  plt.style.use('seaborn')  # Also make grid with white background  sns.set\_theme(style="whitegrid")  # Adopt font for the text in chart. I like Times New Roman so I am using "serif" as the font.family.  mtpl.rcParams['font.family'] = 'serif'  givenData = pd.read\_csv(r'C:\Users\vivia\Desktop\ANL252\TMA\TMA\_Data.csv')  # Let's look at some rows of the data using head() function in pandas  givenData.head()  ## Create a blank chart space with some specified width and height. I like width 12 pixel and height 8 pixel.  plt.figure(figsize=(12,8))  ## pyplot.figure() just create a blank space for chart and figsize(12,8) tells python that width and height should be  ## 12 and 8 pixels respectively.  ax = sns.boxplot(x="Unit", y="Salary", data=givenData, color="#0073e6")  ## seaborn.boxplot() makes a boxplot for categorical data and numerical data.  ## x = " categorical column", y=" numerical column" and data=" given data frame that we have imported",    ## I tried to match the color from excel using the color code : "#0073e6"  plt.ylabel("Salary", fontsize=12) # specify the y-axis label  plt.xlabel("Business Unit", fontsize=12) # specify the x-axis label  plt.title("Range of Salary of Employees in each Unit",fontdict= {'fontsize':16, 'fontweight':"bold"}) # Give some title to chart  plt.show() ## Show the plot |

Table for the chart can be seen in the summary of each category:



Codes for the above table:

|  |
| --- |
| data4table= givenData[['Unit', 'Gender', 'Salary']]  # Table for the above plot  data4table.groupby(['Unit']).describe() |

Chart 2 – Clustered Bar Graph

Chart, bar chart

Description automatically generated

*Figure 4: Clustered bar of No. of Absentees based on Marital Status*

Codes for the above chart:

|  |
| --- |
| # Create an order list so that bins can be arranged as we obtained in Excel  odrLst=["Single","Married","Divorced", "Others"]  # Create blank chart of width 12 and height 8 pixels  plt.figure(figsize=(12,8))  # Using seaborn.barplot() method we can create a cluster bar plot for the given data.  ax = sns.barplot(x="Marital",y="Absence", data=givenData, hue="Gender", order=odrLst, estimator=np.mean, ci=0)  # In the above code, the argument x= " categorical column in our data", y=" Numerical column in given data"  # data=" the variable in which data is stored in Jupyter", hue= " Legend"  # We need mean or average so I used nump.mean to average of salary in each category  # ci= confidence interval but I don't need confidence interval so I just put ci = 0  plt.legend(loc="center left", bbox\_to\_anchor=(1,0.5)) # position of legend on the chart. I like center left  plt.ylabel("Average No. of Absentees", fontsize=12) # y-axis label  plt.xlabel("Marital Status", fontsize=12) # x-axis label  plt.title("Average No. of Absentees of Male and Female Employees based on Marital Status",fontdict= {'fontsize':15, 'fontweight':"bold"}) # title  plt.show() # show the plot |

Corresponding Table

Table

Description automatically generated

Codes for the table:

|  |
| --- |
| #Capturing the data for the second chart  data4table= givenData[['Marital', 'Gender', 'Absence']]  # Table for the below plot  data4table.groupby(['Marital']).describe()  # We can use groupby() method from pandas to get the mean of each category and reshape the table using unstack() method  T1=data4table.groupby(['Marital', 'Gender']).mean().unstack()['Absence']  T1 |

**Question 1 (c)**

The codes to find the length of service of each staff is given below:

|  |
| --- |
| #We are given the data with some missing values of LeftDates. In the python the missing value is just nan  # and its data-type is float. Length of service of a staff can be calculated using the basic algebra.  # Length of Srvice = date of Joining - date of Leaving the company.  # we can easily subtract integres and floats but here the data is date and time.  # It is also mentioned that the Left date where it is missing, the date 1st May 2022 or 5/1/2022  # is to be considered as the leftDate.  # Now the missing values in the leftDate column must be replaced by 5/1/2022  # Let us replace the missing values in the givenData  updatedData = givenData.iloc[:].replace(np.nan, str("5/1/2022"))  # The date is in the string format. to make algebraic calculation, let us modified the column  # joinDate and leftDate as datetime  updatedData['JoinDate'] = pd.to\_datetime( updatedData['JoinDate']).dt.date  updatedData['LeftDate'] = pd.to\_datetime(updatedData ['LeftDate']).dt.date  # It is the time to make calculation. The updatedData has JoinDate and LeftDate as datetime not string  updatedData['Length\_of\_Service']=(updatedData['LeftDate']-updatedData['JoinDate']).astype("timedelta64[D]")/365  #Now we have a new column Length\_of\_Service. It is the time to print minimum, maximum and average length of service  # from the column Length\_of\_Service  print(f"Result: The minimum duration of service is {round(updatedData['Length\_of\_Service'].min(),1)} year; "  + f"the maximum duration of service is {round(updatedData['Length\_of\_Service'].max(),1)} years and "  +f" the average duration of service is {round(updatedData['Length\_of\_Service'].mean(),1)} years. ") |

**Output:**

Result: The minimum duration of service is 0.1 year; the maximum duration of service is 16.3 years and the average duration of service is 6.8 years.

**Question 1 (d)**

The codes to create and interactive user interface to check whether the given full name is in the staff record are written below:

|  |
| --- |
| # For this part we have to make an infinite loop that runs until the user decide to exit the program.  ## We can make an infinite loop using for and while loop or try except statement. But we will use while  state = 1 # This is the condition to stop the program or continue the program for another task.  while state ==1: # The loop will run until state is not equal to 1  # Now we have to save input from the user. We have syntax "variable= input()"  # The function input takes value from the keyboard and save it in the variable.  user\_input = input("\nPlease enter the full name of the staff to be found in record: ")  ## Now we have the value to check in the record of column Staff. To do this we can use for or while loop.  check=0 # condition to use the print on the screen. We have many options like a record can be found, not found,  # found but left the company or found and still working.  for name\_of\_staff in givenData["Staff"]: # Here name\_of\_Staff is a variable to get the value from data and check  # whether it matches the user\_input  if name\_of\_staff == user\_input:  state=1  break;  else: state =0  if state ==1: # If record found than we have to take join date and left date from the data  record\_check = givenData.loc[givenData["Staff"]==user\_input]["LeftDate"].values[0]  join\_date= givenData.loc[givenData["Staff"]==user\_input]["JoinDate"].values[0]  if type(record\_check)!=str: # Missing value is float and date is string so we check if value found not string  # it means the employee has not left the organization. And he/she is still working.  print(f"\n{user\_input} joined the organization on {join\_date} and is still working.")  else: print(f"\n{user\_input} joined the organization on {join\_date} and left on {record\_check}.")  else: print(f"\n{user\_input} is not found in the record.")  # Now we have print the finding on the screen. we need to continue until user decide to quit.  # for this purpose, we will use while loop and try inside the while loop.  while True:  try:  state=int(input("\nDo you want to find record for other staff? "  +"Please type 1 for Yes and any other number to Exit: "))  # To be successfully run the program, we to just update the value in state.  break  except ValueError: print("\n !!Invalid input!! Please try again.")  if state != 1:  print("\nIt was a pleasure to help. \nThank you.") |

**Output:**

Please enter the full name of the staff to be found in record: Ace Potter

Ace Potter joined the organization on 7/11/2011 and left on 9/6/2020.

Do you want to find record for other staff? Please type 1 for Yes and any other number to Exit: Aamna Howell

!!Invalid input!! Please try again.

Do you want to find record for other staff? Please type 1 for Yes and any other number to Exit: 1

Please enter the full name of the staff to be found in record: Aamna Howell

Aamna Howell joined the organization on 9/29/2014 and is still working.

Do you want to find record for other staff? Please type 1 for Yes and any other number to Exit: 1

Please enter the full name of the staff to be found in record: Stefan McMillan

Stefan McMillan is not found in the record.

Do you want to find record for other staff? Please type 1 for Yes and any other number to Exit: 1

Please enter the full name of the staff to be found in record: Aaminah Mcmillan

Aaminah Mcmillan joined the organization on 5/12/2014 and is still working.

Do you want to find record for other staff? Please type 1 for Yes and any other number to Exit: 2

It was a pleasure to help.

Thank you.

**References**

1. *Continuous Data*. iSixSigma. (n.d.). Retrieved August 10, 2022, from https://www.isixsigma.com/dictionary/continuous-data/
2. *Discrete vs Continuous Data*. Intellspot. (n.d.). Retrieved August 10, 2022, from https://www.intellspot.com/discrete-vs-continuous-data/
3. How to create a grouped bar plot in Seaborn (step-by-step). Statology. (2021, October 28). Retrieved August 8, 2022, from <https://www.statology.org/seaborn-grouped-bar-plot/>
4. Pandas.dataframe.boxplot. pandas.DataFrame.boxplot - pandas 1.4.3 documentation. (n.d.). Retrieved August 10, 2022, from <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.boxplot.html>
5. Python Looping Techniques. Programiz. (n.d.). Retrieved August 13, 2022, from https://www.programiz.com/python-programming/looping-technique#:~:text=We%20can%20create%20an%20infinite,we%20get%20an%20infinite%20loop.