

**ANL252 (Online)**

**Python for Data Analytics**

# **Group-Based Assignment**

**July 2022 Presentation**

**Submitted by:**

|  |  |
| --- | --- |
| **Name** | **PI No.** |
| **Chan Teck Fong** | **W2070448** |
| **Chang Guo Rui, Joel** | **Q1910355** |
| **Lim Zheng Hui** | **K1711101** |
| **Mako Wang Jun** | **Y2172686** |

**Tutorial Group: ­­­­­­­­­­­­ TV 05**

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

**Submission Date: 27/08/2022**

**Declaration Page**

We, members of group \_\_\_**5**\_\_\_\_\_\_\_\_\_ , do hereby declare that we each contributed to this assignment and that we collectively agree to a shared grade.

|  |  |  |
| --- | --- | --- |
| Name | Contribution | Signature |
| Chan Teck Fong (Team Lead) | I did questions e (2 Python code for 2 charts & insights) |  |
| Chang Guo Rui, Joel | I did d + e (1 chart & 1 insight) |  |
| Lim Zheng Hui | I did a + b |  |
| Mako Wang Jun | I did c |  |

**Question 1**

**a)** Dataset can be read as below Python coding.

import pandas as pd

import numpy as np

from datetime import datetime

import matplotlib.pyplot as plt

#read the dataset in as a Pandas dataframe

df = pd.read\_csv('GBA\_data.csv')

df.head()

**b)** Below Python coding is used to identify variable columns which have missing values.

import pandas as pd

import numpy as np

from datetime import datetime

import matplotlib.pyplot as plt

# To determine which variable columns have missing values

df = pd.read\_csv('GBA\_data.csv')

df.isnull().sum(axis = 0)

# Covert variable columns with missing values into a list

df.columns[df.isna().any()].tolist()

Hence, we can determine that the variable columns with missing values are origin and destination.

# To determine the number of rows with at least one missing value

missrow = df.isnull().any(axis = 1)

df.loc[missrow[missrow == True].index]

# Treatment of missing value by deleting entire row/column

df.drop(index = missrow[missrow == True].index)

# Same result can also be obtained using .dropna() function

df.dropna()

# Alternatively, all missing values can be replaced by value 0

df.fillna(0)

# Replacing all missing values with the string "unknown" is more suitable since origin and destination is odd to have value 0

misscol = df.isnull().any(axis = 0)

df[misscol[misscol == True].index].fillna(value = "unknown")

In conclusion, the missing data can be treated by removing them entirely or replacing them with specific value or string. In this case, since the missing data are all in origin and destination, it would make logical sense to replace the missing data with the string “unknown” as the values are indeed not known.

**c)** Firstly, for the start time/end time column in our data set. The issue we face is that the format is inconsistent throughout, something as simple as time could be entered in multiple different ways. However, it is not easy to understand the raw data without converting it. An example would be that date format could vary from other data format as seen in the data set given. This could prove to be a problem if we do not standardise it across the board because the code would not be able to read it in the way we intend to, and it might prompt error in coding.

A few ways we could treat this issue is to change the start and end time to 24hr format (hh:mm:ss) or add another column to find the ‘time difference’ which is also known as the amount of time taken. As a result, the code will be able to tell us the time difference of the start and end time appropriately from the suggestions we made.

The rationale behind this is because reading raw data with the 24hr format is easier as compared to the current format and there might be some applications which are not able to translate CSV and would corrupt the data upon opening the file, hence making the data unreadable. As a result, we have cleaned up the data in a manner where it is readable, and it allows us to perform functions in the code to calculate the difference between start and end time in hours, minutes and seconds. By doing so, we can complete data preparation and ensure there are no data quality issues moving forward.

Secondly, in the age column, the data is not updated to be in the present year. Based on the dataset, the age calculated in the data seems to be the year 2020 which is not accurate, and this may cause issues when processing the data. An example would be the trend of age group that commutes the most, could be majorly affected and hence representing an outdated trend. As such, this causes issues in data quality.

A suggestion to overcoming this problem would be to rely on the yob column instead of the age and make use of the functions available in python to compute the current age of all the dataset based on the present year. By doing so, we can keep the dataset up to date and produce accurate analysis on the dataset.

The rationale behind this is so that the data is presented in an updated and accurate manner to avoid providing a false narrative when processing such data. Thus, we can ensure that as part of data preparations, we can implement the suggestion mentioned above to treat the data quality issues provided in the dataset.

Lastly, for data with missing values such as ‘-’, ‘--’, and ‘?’ or incomplete field, these values will not be read for data analytics as the data is not recognized and hence the programme would not be able to process it. This may occur due to insufficient information while the data was recorded or at the point of data collection, there is an error, and the system did not register the information. As a result, having missing values is a common occurrence which could be easily dealt with.

For the issue presented above, one suggestion would be to convert all missing values and dashes to facilitate data cleaning and ensure a smooth transition of code. By doing so, the data will be readable, and we would be able to run mathematical equations on our code without errors.

The rationale behind this is so that we can convert the missing or unreadable data into data that can be processed by code which enables us to create functions to apply data analytics on our dataset without bumping into any error and to have a code that runs smoothly to perform our work. Therefore, we could efficiently complete data preparation by using the suggestions above.

**d)** The Python coding is:

import pandas as pd

import numpy as np

from datetime import datetime

import matplotlib.pyplot as plt

df = pd.read\_csv('GBA\_data.csv')

# Convert to 12h clock

df['Time'] = pd.to\_datetime(df['start'], format='%Y-%m-%d %H:%M:%S.%f').dt.strftime('%I %p')

# sorted from highest to lowest

df['Time'].value\_counts()

05 PM 82343

08 AM 78583

06 PM 62396

09 AM 60362

04 PM 51293

07 AM 38490

07 PM 36967

03 PM 33549

12 PM 31661

10 AM 30619

01 PM 29887

02 PM 27911

11 AM 27202

08 PM 23038

09 PM 16097

06 AM 12550

10 PM 10374

11 PM 6103

12 AM 3312

05 AM 3195

01 AM 1598

02 AM 998

04 AM 837

03 AM 644

Name: Time, dtype: int64

Based on the data provided, the time whereby the highest number of commuters start their journey is at **05 PM** followed by **08 AM** and the time whereby there is the least number of commuters is at 03 AM. This shows that the time which has the highest number of commuters starting their journey is at 08 AM in the morning and 05 PM in the late afternoon.

**e)** The dimension of dataset comprises of 10 variables in columns and 670,009 rows. 3 charts were created using Python coding.

i. Visualisation on “Commuter’s Gender by Age”

From the chart shown in Figure 1, there are three types of commuter’s gender which are female, male and other. Commuter’s aged 27-38 take more trips compare with other age groups. The greatest number of commuter’s travel at the age of 32 and least commuter travel age is 71. It could reveal that the age influencing commuter’s travel behaviour. An individual’s phase of life like youth, working age, or retirement related with the traveling frequency (Kalhoro, et al., 2021)

Another factor which affects travel behaviour is gender. Figure 1 shows that majority of commuters are male (493,829) compare with female (165,744) and other (9,909). The greatest number of females commuting at the aged of 31-32 and least at aged of 71. It could be household responsibilities, gender inequality, sensitivity of female to environment associated with frequency of gender commuting.

Chart, histogram

Description automatically generated

Figure 1 – Commuter’s Gender by Age

ii. Visualisation on “Commuter’s Type by Age”

From the chart shown in Figure 2, there are three types of commuter’s pass - ad-hoc, concession, regular. Over 554,881 commuters are travelling with regular pass, 46,572 are with concession pass and 68,030 are ad-hoc commuters. Commuters at the aged of 21-54 with regular pass are working adults who monetarily independent to pay the fees. As for the regular pass holder at the aged of 66-71, they could be occasionally travel, subsequently more worth to use regular pass and ad-hoc pass.

The eligibility aged of concession travel pass should be at the aged of 55 and above. Somehow, a portion of commuters who 55 and above are travelled with ad-hoc and regular pass, it could be the travel frequency are lesser or occasionally use it. Hence more worth to go by ad-hoc and regular pass. There is an uptrend of ad-hoc commuters at 20-32. These aged group of youthful ad-hoc commuters prefer freedom, unstructured and not required detail preparation of traveling.

**Chart, histogram

Description automatically generated**

Figure 2 – Commuter’s Type by Age

iii. Visualisation on “Subscriber by Age”

Based on the visualisation produced by the Python code regarding the breakdown of ‘Subscriber by Age’, majority of the customers did not subscribe. Due to the lack of information provided by the completed raw data, it is assumed that these customers did not want to subscribe to receive marketing promotions from the transport company. The lack of promotional items provided by the company could be another reason which prevents the customers from subscribing. Customers will not be enticed to subscribe if they see no value/offers provided occasionally for taking the transport provided by the company. From Figure 3, the age group majority comes from ages 25-45, this age group belongs to the working class. Being in the working class, it is normal to be travelling frequently for work. Based on the data collected from part d) the peak hours of 8 AM and 5 PM could signify that majority of the working-class commuters travel to their workplace and back home.

Chart, bar chart, histogram

Description automatically generated

Figure 3 – Subscriber by Age

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

df = pd.read\_csv("C:\\Users\\Candice\\Desktop\\Python practice\\GBA\_data.csv", low\_memory=False)

df=df.replace('?',np.NaN)

## Data cleaning is required. Search dataframe for cells with "?" and replace with NaN

df=df.replace({'yob':{'--':np.NaN},

'age':{'--':np.NaN},

'gender':{'-':np.NaN}})

## Search dataframe for cells with "--", "-" and replace with NaN

df=df.dropna()

## Remove all the dataframe for cells with NaN as part of data cleaning

df1 = df.groupby(['age', 'gender'])['gender'].count().unstack().fillna(0).astype(int)

print(df1)

df1.sum()

## i. To get the sum by gender

## To plot "Commuter's Gender by Age"

df1.plot(kind='bar', stacked=True)

plt.title("Commuter's Gender by Age")

plt.xlabel("Age")

plt.ylabel("No. of Commuters")

plt.figure(figsize=(80,60))

plt.rc('xtick', labelsize= 7.5)

plt.rc('ytick', labelsize= 8)

df2 = df.groupby(['age', 'type'])['type'].count().unstack().fillna(0).astype(int)

print(df2)

df2.sum()

## To get the sum by gender

## ii. To plot "Commuter's Type by Age"

df2.plot(kind='bar', stacked=True)

plt.title("Commuter's Type by Age")

plt.xlabel("Age")

plt.ylabel("No. of Commuters")

plt.figure(figsize=(80,60))

plt.rc('xtick', labelsize= 7.5)

plt.rc('ytick', labelsize= 8)

df3 = df.groupby(['age', 'subscriber'])['subscriber'].count().unstack().fillna(0).astype(int)

print(df3)

df3.sum()

## iii. To plot "Subscriber by Age"

df3.plot(kind='bar', stacked=True)

plt.title("Subscriber by Age")

plt.xlabel("Age")

plt.ylabel("No. of Subscriber")

plt.figure(figsize=(80,60))

plt.rc('xtick', labelsize= 7.5)

plt.rc('ytick', labelsize= 8)

plt.show()

**References**

Kalhoro.M, Au Yong.H.N., Ramendran. C. (November 2021). *Commuter’s Age, Gender and Public Transport Usage: A literature Review* https://www.researchgate.net/publication/357516910\_Commuter's\_Age\_Gender\_and\_Public\_Transport\_Usage\_A\_Literature\_Review