

**ANL252**

**Python for Data Analytics**

****

**Group-Based Assignment**

**July 2023 Presentation  
Group 5**

|  |  |
| --- | --- |
| **Name** | **PI. Number** |
| ANG XIU ZHI | J1771750 |
| NAOMI TINA GAN MIN | Z2210445 |
| TAN MEI YI RACHEL | Q2081315 |

**Tutorial Group:** T03

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

**Submission Date:** 12/10/2023

**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 |
| Ang Xiu Zhi (Team Lead) | I did questions 1(a)/(b) |  |
| Naomi Tina Gan Min | I did 1(c) |  |
| Tan Mei Yi Rachel | I did 1(d) |  |

**Question 1**

**(a)**

import pandas as pd

# reading the dataset

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

# dimensions of the dataset

print('Number of rows and columns: ', df.shape)

To read the data set using python, we first have to import the Pandas library. Next, the function ‘pd.read\_cs’ will be used in order to create a Pandas Dataframe termed ‘df’. This dataframe will contain the data from the csv indicated in the code. The dataset we are using in this case is the GBA\_HDB.csv file.

After creating the dataframe, we will then identify the dimensions of the dataset using python. The dimensions of Pandas Dataframe are the number of rows and columns within the dataframe. To identify the dimensions, the shape attribute of the dataframe will be used. The ‘shape’ function will return a tuple which contains the number of rows and columns in the dataframe.

The print function is also used to print the number of rows and columns to the console. The output received for the dataset GBA\_HDB.csv is as follows:

Number of rows and columns: (1250, 11)

**(b)**

# identifying variables with missing values

print('Number of missing values in each column:')

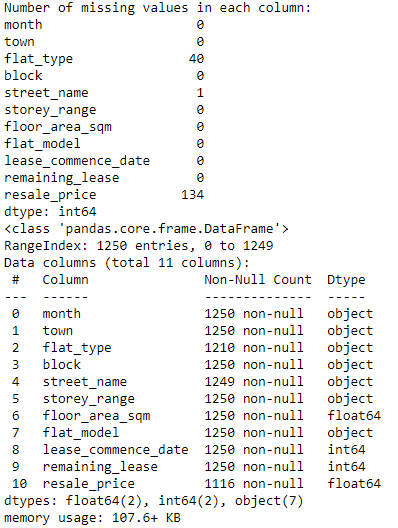
print(df.isna().sum())

# summary of the dataset

print(df.info())

To identify the variables with missing value, the ‘isna()’ function is used. This function will detect if there are any missing values for an array-like object and returns a boolean series which indicates the variable that has missing value within the dataframe. The ‘sum() ‘ function is then used to count the total number of missing values for each variable, followed by the ‘info()’ function which helps display an overall summary of the dataframe with the total number of non-null values for each variable.

It is necessary to handle missing values in a dataset for several reasons. Having missing values in a dataset can lead to the introduction of biasness into the analysis. A dataset with multiple missing values in the resale\_price column might cause the analysis to be skewed towards the lower or higher ends of the price spectrum, affecting the accuracy and reliability of the analysis. Missing values can also cause errors when algorithms are applied to it since most are unable to deal with such values. However, outright elimination of missing values can also cause issues such as smaller datapoints and data loss. Therefore, missing values in a dataset needs to be handled with an appropriate method.



**(c)**

Here are some of the common methods and rationales behind the choices.

1. Imputation with Mean, Median or Mode
   1. For missing numeric data such as “floor\_area\_sqm” and “remaining\_lease”, it can be replaced with the values of mean (normally distributed data) or median (skewed data). For categorical data such as “flat\_type” and “flat\_model”, missing values can be replaced with mode (most used category).

import pandas as pd

# Replace missing values in 'floor\_area\_sqm' with the mean

df['floor\_area\_sqm'].fillna(df['floor\_area\_sqm'].mean(), inplace=True)

# Replace missing values in 'flat\_type' with the mode

df['flat\_type'].fillna(df['flat\_type'].mode()[0], inplace=True)

1. Forward/Backward Fill
   1. With time-series and ordered data such as “lease\_commence\_date”, forward-fill or backward-fill can be used to carry the least observed values forward or backward in the series.

import pandas as pd

# Forward-fill missing values in 'lease\_commence\_date'

df['lease\_commence\_date'].fillna(method='ffill', inplace=True)

1. Drop
   1. If the data missing do not impact the analysis significantly, one can choose to drop the rows and columns.

import pandas as pd

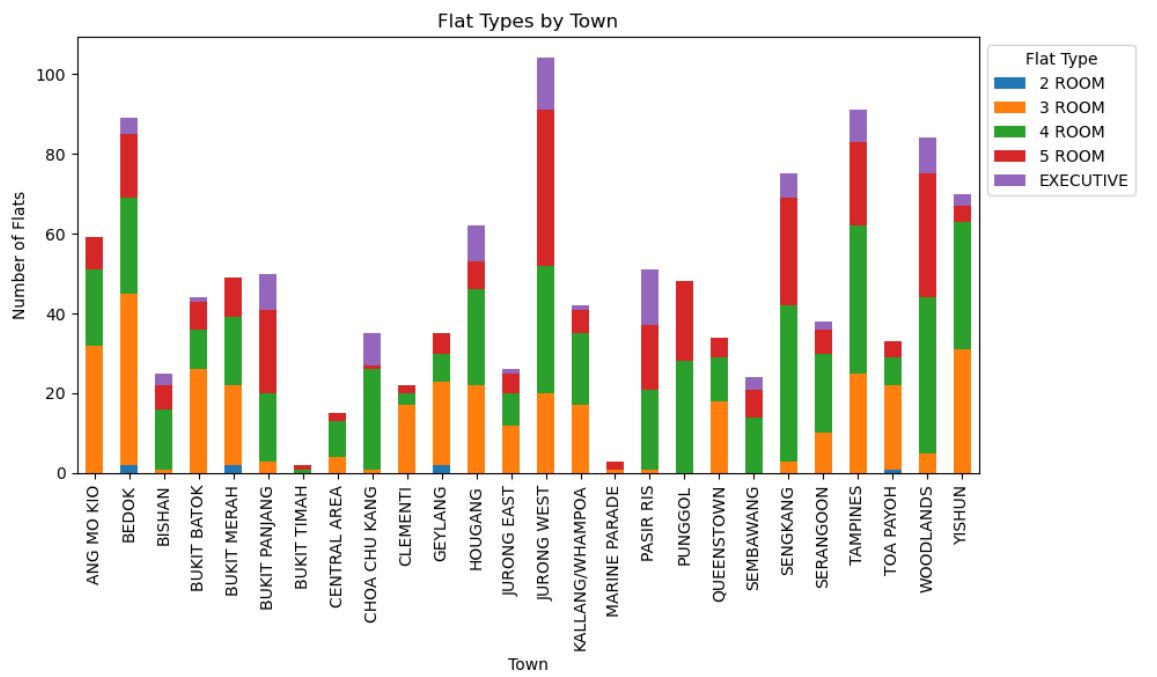
# Drop the rows with missing values

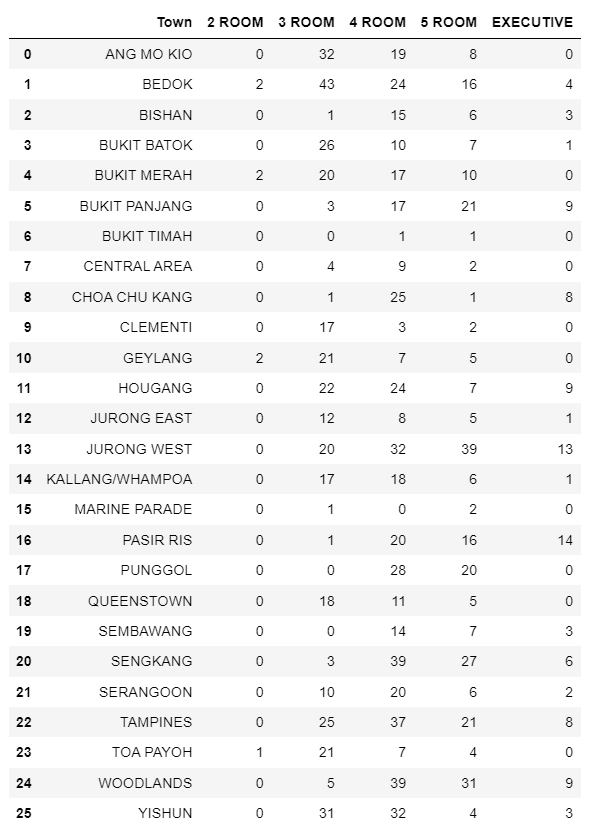
df.dropna(inplace=True)

The choice of replacement for missing data is specific to the data’s nature and its impact on the analysis.

**(d)**

**Chart 1**

****

****

**Code:**

import pandas as pd

import matplotlib.pyplot as plt

df = pd.read\_csv("GBA\_HDB.csv")

flat\_type\_counts = df.groupby(['town', 'flat\_type']).size().unstack(fill\_value=0)

fig, ax = plt.subplots(figsize=(10, 6))

ax = flat\_type\_counts.plot(kind='bar', stacked=True, ax=ax)

plt.xlabel('Town')

plt.ylabel('Number of Flats')

plt.title('Flat Types by Town')

plt.xticks(rotation=90)

labels = ['2 ROOM', '3 ROOM','4 ROOM','5 ROOM', 'EXECUTIVE']

ax.legend(labels, title='Flat Type')

plt.legend(title='Flat Type', loc='upper left', bbox\_to\_anchor=(1, 1))

plt.tight\_layout()

plt.show()

flat\_type\_by\_town\_table = pd.DataFrame(flat\_type\_counts)

flat\_type\_by\_town\_table.reset\_index(inplace=True)

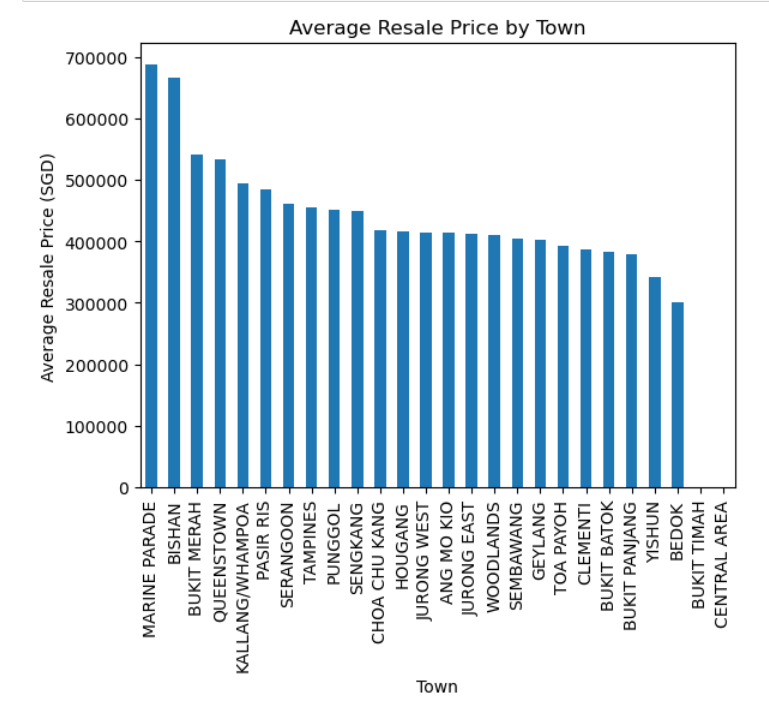
flat\_type\_by\_town\_table.columns = ['Town','2 ROOM', '3 ROOM','4 ROOM','5 ROOM', 'EXECUTIVE']

flat\_type\_by\_town\_table

**Insights:**

The chart illustrates the distribution of flat types across towns, aiding potential buyers in assessing housing options. For instance, lower-income households might seek 2-room flats, while young couples prefer 3 or 4-room units, and larger families opt for 5-room or executive flats. Therefore, they can filter out towns based on their preferences. Typically, most of the flats in Singapore are 3 and 4-room flats, with a substantial amount of 5-room flats. Executive and 2-room flats are much more limited. It can also be seen that the town Jurong West offers diverse options and forms a majority of the dataset.

**Chart 2**





**Code:**

town\_mean\_price = df.groupby('town')['resale\_price'].mean().sort\_values(ascending=False)

town\_mean\_price.plot(kind='bar')

plt.title('Average Resale Price by Town')

plt.xlabel('Town')

plt.ylabel('Average Resale Price (SGD)')

plt.xticks(rotation=90)

plt.show()

town\_mean\_price\_table = pd.DataFrame(town\_mean\_price)

town\_mean\_price\_table.reset\_index(inplace=True)

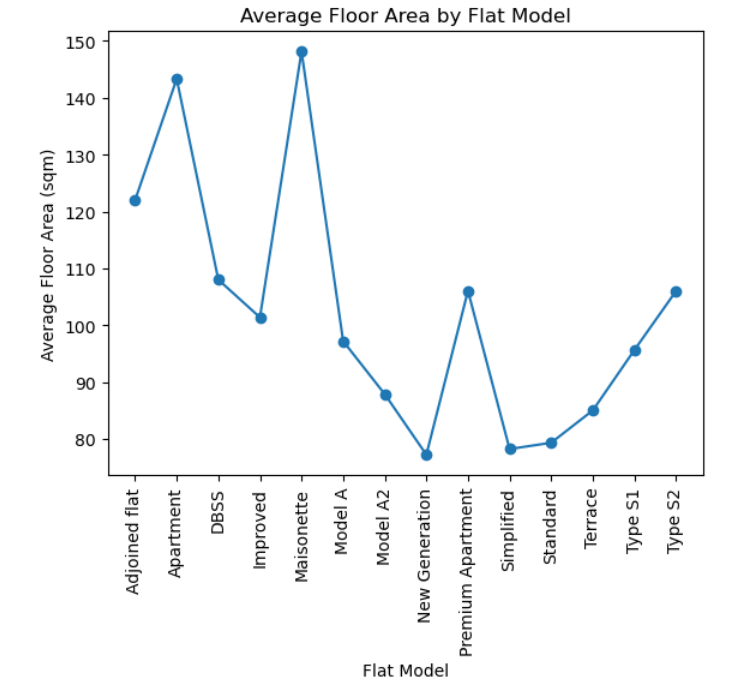
town\_mean\_price\_table.columns = ['Town', 'Average Resale Price (SGD)']

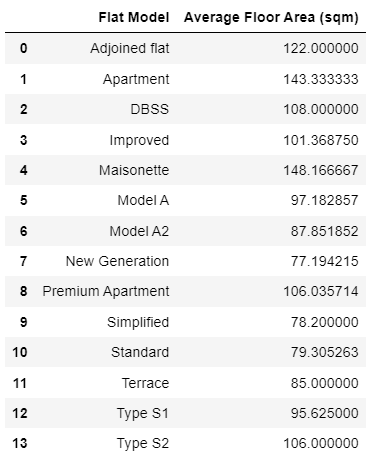
town\_mean\_price\_table

**Insights:**

The majority of average resale prices range between $380,000 to $420,000. Marine Parade and Bishan have the highest resale prices ranging from $600,000 to $700,0000, while Bedok has the lowest, at $300,000. This chart helps HDB sellers to understand the baseline prices in the market and set realistic expectations for their own listings. HDB buyers may also make more informed decisions by filtering out towns based on their budget limitations.

**Chart 3:**

****

****

**Code:**

average\_floor\_area\_by\_flat\_model = df.groupby('flat\_model')['floor\_area\_sqm'].mean()

plt.plot(average\_floor\_area\_by\_flat\_model.index, average\_floor\_area\_by\_flat\_model.values, marker='o')

plt.xlabel('Flat Model')

plt.ylabel('Average Floor Area (sqm)')

plt.title('Average Floor Area by Flat Model')

plt.xticks(rotation=90)

plt.show()

flat\_model\_area = pd.DataFrame({'Flat Model': average\_floor\_area\_by\_flat\_model.index, 'Average Floor Area (sqm)': average\_floor\_area\_by\_flat\_model.values})

flat\_model\_area

**Insights:**

This chart assists potential buyers who are unfamiliar with the types of flat model available and are unsure which one best matches their housing needs and preferences, particularly in terms of the desired size.

Standard flats occupy the least average floor space, whereas Maisonettes, the most.

Almost similar spaces (in sqm) are occupied by the following sets of flat models:

New generation, Standard, Simplified

Maisonettes, Apartments

Terrace, Model A2

Type S1, Model A, Improved

Type S2, DBSS

Hence, using this, potential flat buyers can evaluate additional qualities, like the availability of such flats in towns of their choice, and their resale prices.

**References**

Wu, K. Y., & Zhu, S. (2023). ANL252 Python for data analytics. Singapore University of

Social Sciences.