

# **ANL252**

# **PYTHON FOR DATA ANALYTICS**



**Group-based Assignment**

# **July 2023 Presentation**



**Submitted by:**

|  |  |
| --- | --- |
| **Name** | **PI No.** |
| **TEA ICH NGY** | **Y2210678** |
| **MUHAMMAD FITHRI BIN FADILIAH** | **Z1910329** |
| **YEO JIA JIE JACKSON** | **N2770191** |
| **TEH YI LIN** | **B2081892** |

**Tutorial Group: ­­­­­­­­­­­­ T03**

**Instructor’s Name: Prof. KUMAR MUNISH**

**Submission Date: 13/10/2023**

**Declaration Page**

We, members of group \_\_\_\_\_\_\_6\_\_\_\_\_ , do hereby declare that we each contributed to this assignment and that we collectively agree to a shared grade.

|  |  |  |
| --- | --- | --- |
| Name | Contribution | Signature |
| Yeo Jia Jie Jackson | I did question 1d histogram |  |
| Muhammad Fithri Bin Fadiliah | I did question 1a, 1b, and 1c |  |
| Tea Ich Ngy | I did 1c (partly), and 1d bar graph |  |
| Teh Yi Lin | I did 1d scatter plot |  |

**Question 1(a)**

# Import Python Data Analysis library “pandas”, under alias “pd”.

import pandas as pd

# Execute 'pd.read\_csv' to load and read GBA\_HDB.csv, with variable name 'hdb\_df'.

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

# Input .shape function to identify dimensions of hdb\_df; expected output - (1250, 11), signifying

# 1250 rows and 11 columns.

hdb\_df.shape

**Question 1(b)**

# White spaces "" represent missing values in HDB\_GBA.csv file, changed to "NaN" in pandas by default.

# Execute .isnull().sum() function to identify variables (columns) with missing values

# .isnull() function involves checking cells, returning True for NaN values - giving value 1 for True, and 0 for False.

# .sum(axis = 0) function involves checking the sum of all cells; "axis = 0" parameter returning values for columns.

hdb\_df.isnull().sum(axis = 0)

# From output, variables "flat\_type" has 40 missing values,

# "street\_name" has 1 missing value,

# and "resale price" has 134 missing values.

Reasons to handle missing data (Premanand, 2023):

* Improper handling of missing values may cause bias and imprecision in statistical analysis, leading to incorrect results.
* Important pre-processing procedure as many algorithms cannot process data where there are missing values.
* Datasets with many missing values can impact machine learning model training, heavily impairing its quality as missing values negatively affect its robustness.

**Question 1(c):**

Methods of handling missing dataset values:

1. Deleting of rows containing missing values via .dropna() (Satyam, 2020) -

* Pro: Efficient in removing null value observations.
* Con: Other information in deleted row are lost; highly problematic if large proportion of variable have missing values.

# To preserve raw data of HDB\_GBA.csv, new variable name "hdb\_df\_drop" given to represent dataset with

# dropped observations.

# .dropna(axis = 0) function to remove rows that hold cells with missing values.

# "any" value in how parameter involves deleting rows holding minimally one missing value under any column.

hdb\_df\_drop = hdb\_df.dropna(axis = 0, how = "any")

2. Replacing missing values via .fillna with pre-established values (e.g., 0, unknown, mean, median, mode) (Satyam, 2020)

* Pro #1: Data loss as in .dropna() reduced, preserving other information.
* Pro #2: Different data types in different variables can be treated differently (e.g, name vs. numericals).
* Con: Replaced values act as proxies; true values unknown and require data collection.

# .fillna() function to replace missing values.

# For variables requiring integer values like, .fillna(), add parameter "value = 0" to replace missing value with 0,

hdb\_df["resale\_price"] = hdb\_df["resale\_price"].fillna(value = 0)

# For variables requiring string values, parameter is "value = 'Unknown'".

hdb\_df[["flat\_type","street\_name"]] = hdb\_df[["flat\_type","street\_name"]].fillna(value = "Unknown")

# Finding and removing outliers in hdb\_df; given new variable treated\_gba\_hdb

q1 = hdb\_df["resale\_price"].quantile(q=.25)

q3 = hdb\_df["resale\_price"].quantile(q=.75)

iqr = q3-q1

treated\_gba\_hdb = hdb\_df[~((hdb\_df["resale\_price"]<q1-1.5\*iqr) | (hdb\_df["resale\_price"]>q3+1.5\*iqr))]

**Question 1(d):**

**Scatter Plot:**

**[Code]**

import matplotlib.pyplot as plt

import seaborn as sns

# Set the aesthetic style of the plots

sns.set(style="whitegrid")

# Create scatterplot charts

fig, axes = plt.subplots(1, 2, figsize=(15, 6))

# Scatterplot for floor\_area\_sqm vs resale\_price

sns.scatterplot(x='floor\_area\_sqm', y='resale\_price', data=dataset, ax=axes[0])

axes[0].set\_title('Floor Area (sqm) vs Resale Price')

# Scatterplot for remaining\_lease vs resale\_price

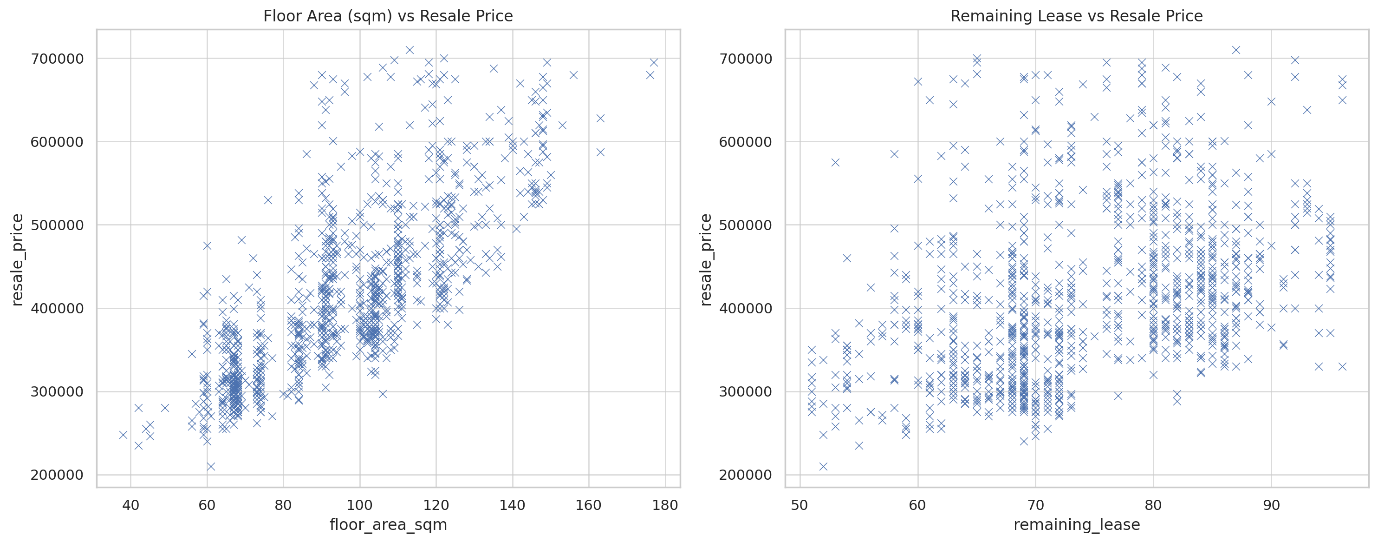
sns.scatterplot(x='remaining\_lease', y='resale\_price', data=dataset, ax=axes[1])

axes[1].set\_title('Remaining Lease vs Resale Price')

plt.tight\_layout()

plt.show()

**[Diagrams]**

****

**Floor Area (sqm) vs Resale Price**

The scatterplot indicates a positive correlation between the floor area in square meters and the resale price. As the floor area increases, the resale price also tends to rise. This relationship is expected, as larger flats generally command higher prices.

**Remaining Lease vs Resale Price**

The second scatterplot doesn't show a clear trend between the remaining years on the lease and the resale price. The data points are dispersed, suggesting that the remaining lease duration doesn't have a straightforward impact on resale price.

**Observations**

Floor Area as a Price Determinant: The floor area of a flat appears to be a significant factor in determining its resale price. This is useful for both buyers and sellers as an initial gauge for pricing.

Ambiguity in Lease Impact :The impact of the remaining lease years on the resale price is not as evident. This could mean that other variables, like the flat's condition or location, might be more influential in this context.

(Words: 165)

**Bar Chart:**

**[Codes]**

import pandas as pd

import matplotlib.pylab as plt

import seaborn as sns

# Setting graph style from seaborn

sns.set\_style('darkgrid')

# Choosing bar colour

colour = "#8bbdd9"

#Plotting the bar chart

ax = sns.barplot(treated\_hdb\_df, x="remaining\_lease", y="town", estimator="mean", color=colour, ci=None)

ax.bar\_label(ax.containers[0], fontsize=10);

# Setting bar chart title. x label and y label

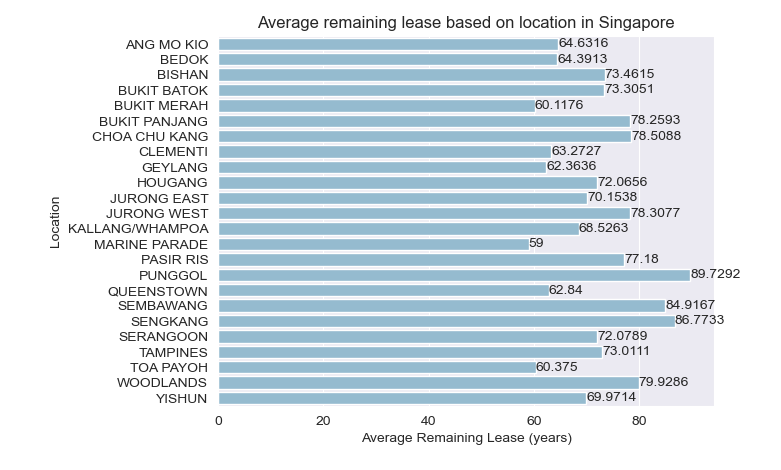
plt.xlabel("Average Remaining Lease (years)")

plt.ylabel("Location")

plt.title("Average remaining lease based on location in Singapore")

plt.show()

**[Diagram]**

****

Bar chart are usually used to show useful statistical relations between numerical and non-numerical factors. In this case, our group had chosen to plot the relationship between town location in Singapore and the average remaining lease.

The bar chart shows the average remaining lease in each location/town in Singapore, ranging from 59 years to 90 years. This shows us that for locations with higher average remaining lease, there are newer HDB for sale in that area, which means these HDBs are “younger”. From the chart, some of those areas that have the highest average remaining lease are Punggol (89.7 years), Sengkang (86.7 years) and Sembawang (84.9 years). This means that resale-buyers who are interested in buying newer HDBs can look into these areas straight away.

In contrast, areas such as Marine Parade (59 years). Bukit Merah (60.7 years) and Toa Payoh (60.3years) are among the areas with the lowest average remaining lease. We can induce from this information alone that these are matured estates. (170 words)

**Histogram Plot:**

**[Code]**

# import relevant libraries for histogram plot

import pandas as pd

import matplotlib.pyplot as plt

# create df from csv file

treated\_hdb\_df = pd.read\_csv("Treated\_GBA\_HDB.csv")

# Plot a histogram of a resale price column

treated\_hdb\_df.hist(column = 'resale\_price')

# Set the title and axis labels

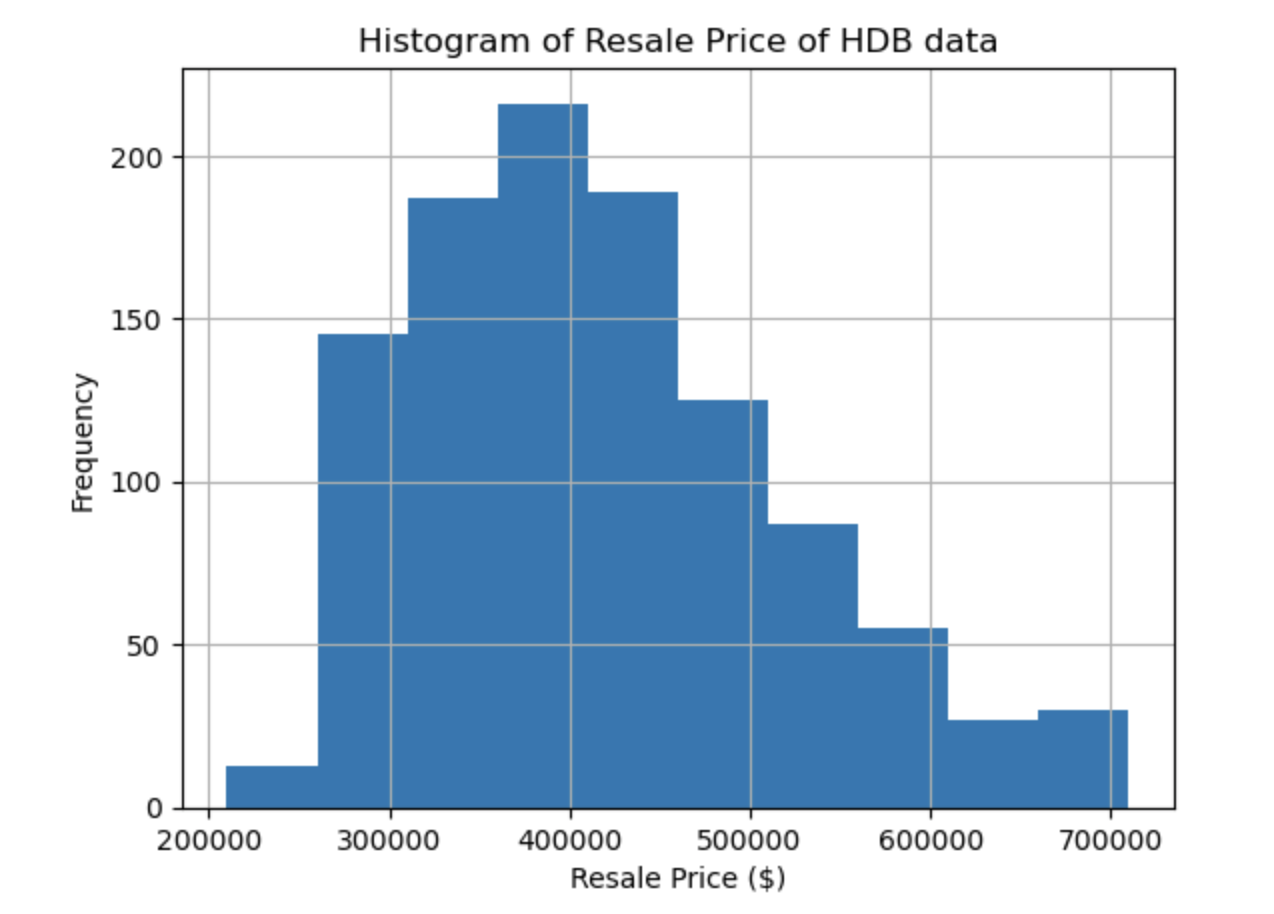
plt.title('Histogram of Resale Price of HDB data')

plt.xlabel('Resale Price ($)')

plt.ylabel('Frequency')

plt.show()

**[Diagram]**

****

From the histogram plotted, it can be seen that most HDBs are transacted between $350,000 to $410,000 range. There are very few HDBs sold at the lower end of the resale prices close to $200,000 and also not many are sold at the higher end of the resale prices close to $700,000. The histogram is skewed to the left. It can be seen that the resale prices for the period that the data was collected are relatively affordable. (78 words)

References:

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

*Shaw, Z. A. (2017). Learn Python 3 the hard way: A Very Simple Introduction to the Terrifyingly Beautiful World of Computers and Code. Addison-Wesley Professional.*

*seaborn.barplot — seaborn 0.13.0 documentation*. (n.d.). <https://seaborn.pydata.org/generated/seaborn.barplot.html>

*seaborn.color\_palette — seaborn 0.13.0 documentation*. (n.d.). <https://seaborn.pydata.org/generated/seaborn.color_palette.html#seaborn.color_palette>

*seaborn.set\_style — seaborn 0.13.0 documentation*. (n.d.). <https://seaborn.pydata.org/generated/seaborn.set_style.html>

*Satyam, K. (2020, July 24). 7 Ways to Handle Missing Values in Machine Learning. Medium. https://towardsdatascience.com/7-ways-to-handle-missing-values-in-machine-learning-1a6326adf79e*

*Premanand, S. (2023, February 27). A complete guide to dealing with missing values in python. https://www.analyticsvidhya.com/blog/2021/10/a-complete-guide-to-dealing-with-missing-values-in-python/*