A logo with text on it

Description automatically generated

ANL252 Python for Data Analytics

Group-Based Assignment

Assignment July 2023

Submission Date:

|  |  |
| --- | --- |
| Name | PI No |
| Loke Kum Wai | M2311263 |
| Ang kia loke | B2310015 |
| Beverlyn Tan Jiamin | E2081428 |
| Muhammad Bin Osman | H2310678 |

## Question 1(a)

An inbuild library called “pandas”, which provides a number of functions for data processing and analysis, can be used to read the dataset in Python. As such we should import the Pandas library first since it enables us to work with structured data and present it to us in a neat way. Since we are trying to read a CSV file, we can use the function “read\_csv” which will enable us to read the dataset as a DataFrame.

The dimensions of the dataset can be identified by using the “Shape” attribute of a DataFrame which produces a tuple indicating the number of rows and columns. Hence, we can use the function “.shape” to identify the dimensions which will give us an output of 1,250 rows and 11 columns.

The relevant codes to read the CSV file and to identify the dimensions of the dataset are as follows:

#To read the CSV file

#Import pandas library

import pandas as pd

#reading dataset

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

hdb\_data

#To identify the dimensions of the dataset

hdb\_data.shape

## Question 1(b)

We can identify the missing values using Python by utilizing “pandas” where we can use the function “.isnull().sum(). This will return a Series object containing the number of missing values in each DataFrame column and the relevant output for missing values are flat\_type (40), street\_name (1) and resale\_price(134).

Handling missing values is essential to allow well-informed decisions to be made. The primary reason would be to ensure data quality since missing values has the potential to skew the results which interferes with data quality and result in inaccurate conclusions (Tamboli, 2023). Missing value handling should be part of data cleansing, which is critical for assuring the quality of the dataset.

Another reason would be because many statistical approaches necessitate complete data for proper analysis. Missing values might have an impact on the dataset's statistical features such as mean and variance. Failure to account for missing data may cause statistical analysis to be skewed and thereby resulting in inaccurate results.

The relevant codes to identify the variables with missing values are as follows:

#To locate missing values

missing\_values = hdb\_data.isnull().sum()

missing\_values

## Question 1(c)

One way to handle the missing data with Python would be to remove the rows or columns with missing values since the number of missing values is negligible in comparison to the total number of values in the dataset. The percentage of missing value is 14% which suggests that removing these rows would not significantly impact the analysis that are to be drawn upon. Thus, we can remove the missing value by using the “dropna()” method for missing values in the rows and “dropna(axis=1)” method for missing values in the columns.

Alternatively, we can use the “fillna()” function to replace the missing value with a specific value or summary statistic like the mean to preserve the central tendency. To replace the missing value with a specific value, we can use interpolation methods such as linear or polynomial interpolation to approximate missing values based on data relationships or data-driven approaches such as k-means clustering or matrix factorization to impute missing values based on data patterns.

## Question 1(d)

Chart 1:Scatterplot of Resale Price vs Remaining Lease

Based on the data, we see a generally increasing trend, where there is a positive correlation between resale price and remaining lease. The data spread is relatively tight, where there is a concentration around 70 years and 80-85 years of remaining lease. There are also outliers that are relatively evenly spread, not affecting the data severely. Based on this data, we can infer that the homes with higher remaining years of lease tend to have higher resale rates. The remaining lease can be an important factor to consider when buyers are purchasing resale homes. Depending on their preferences, some might prefer to purchase houses that are low on lease as it is more affordable or are only planning to stay temporarily. Buyers that want to invest in longer lease houses might find it advantageous for secondary purposes such as property assets or as a form of assurance that the house is still relatively new and well-maintained, not having to worry about frequent utility failures or infrastructure deterioration.

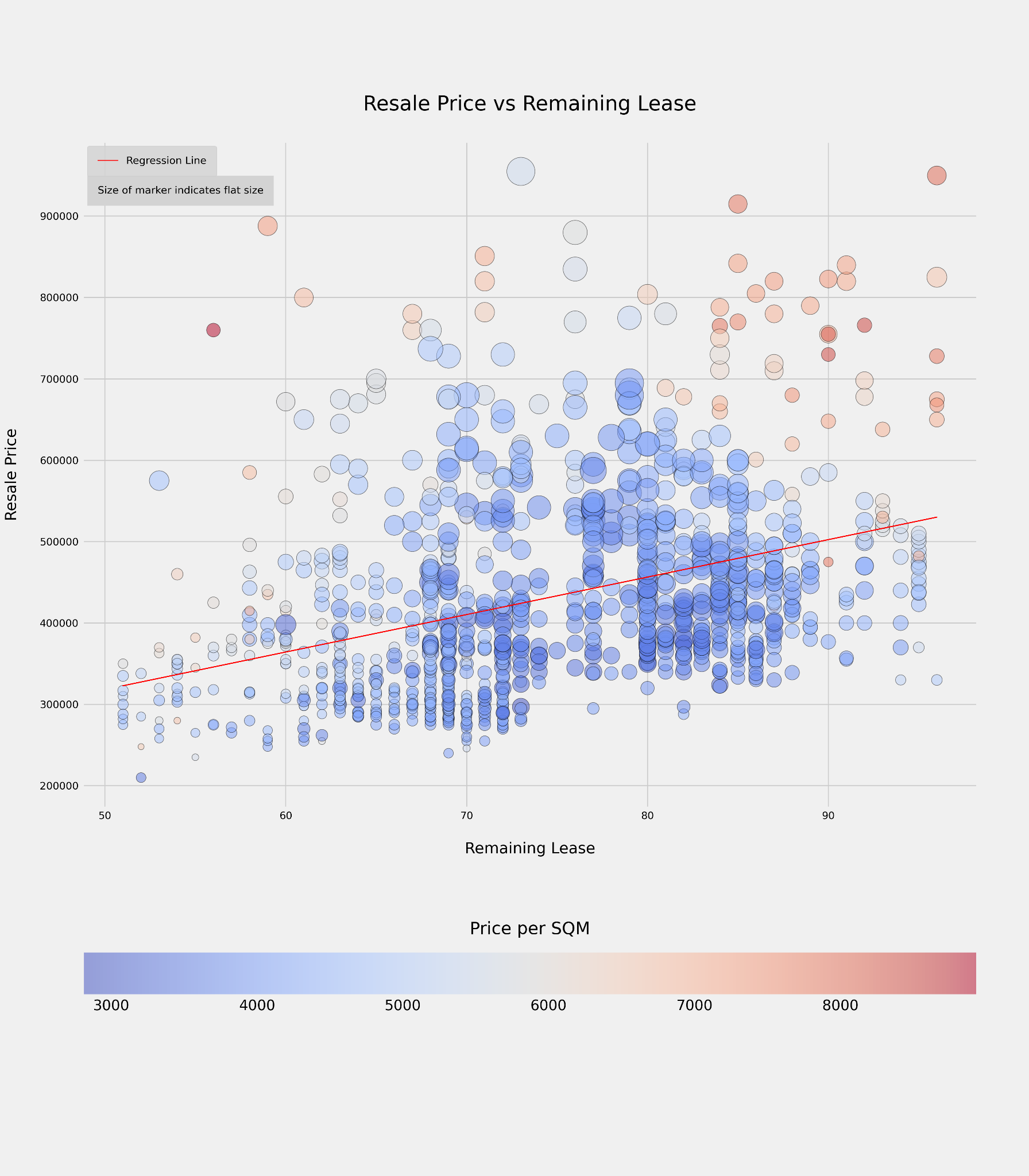


Chart 1 Table:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **remaining\_lease** | **resale\_price** | **price\_per\_sqm** |
| **0** | 70 | 255000.0 | 4250.000000 |
| **1** | 65 | 275000.0 | 4044.117647 |
| **2** | 64 | 285000.0 | 4130.434783 |
| **3** | 63 | 290000.0 | 4264.705882 |
| **4** | 64 | 290000.0 | 4264.705882 |
| **...** | ... | ... | ... |
| **1245** | 69 | 460000.0 | 3432.835821 |
| **1246** | 77 | 500000.0 | 3759.398496 |
| **1247** | 72 | 525888.0 | 3601.972603 |
| **1248** | 72 | 538000.0 | 3788.732394 |
| **1249** | 72 | 550000.0 | 3767.123288 |

Chart 1 Python code:

# Import libraries

import pandas as pd

from matplotlib import pyplot as plt

from sklearn.linear\_model import LinearRegression

# Style of graph

plt.style.use('fivethirtyeight')

# Read CSV

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

# Remove rows with NaN values

df = df.dropna()

# Create new column for resale price in thousands

df['resale\_price\_in\_thousands'] = df['resale\_price']/1000

# Create a new column for price per sqm

df['price\_per\_sqm'] = df['resale\_price'] / df['floor\_area\_sqm']

# Create scatter graph

plt.figure(figsize=(12, 18))

plt.scatter(df['remaining\_lease'], df['resale\_price'], c=df['price\_per\_sqm'], alpha=0.5, cmap='coolwarm', s=(df['floor\_area\_sqm']\*\*2)/40, edgecolors='k')

#color bar settings

cbar = plt.colorbar(label='Price per SQM', orientation='horizontal')

cbar.ax.yaxis.label.set\_fontsize(20)

cbar.ax.xaxis.labelpad = -90

# Graph labels

plt.xlabel('Remaining Lease',fontsize=15, labelpad=20)

plt.ylabel('Resale Price', fontsize=15, labelpad=20)

plt.title('Resale Price vs Remaining Lease', fontsize=20, pad=30)

plt.grid(True)

# Tick labels

plt.xticks(fontsize=10)

plt.yticks(fontsize=10)

# Add Linear regression line

model = LinearRegression()

X = df[['remaining\_lease']]

y = df['resale\_price']

model.fit(X, y)

y\_pred = model.predict(X)

plt.plot(df['remaining\_lease'], y\_pred, color='red', linewidth=1, label='Regression Line')

# Add annotation

plt.annotate(

'Size of marker indicates flat size',

xy=(49.8, 928000), fontsize=10,

bbox=dict(boxstyle='square,pad=1',

edgecolor='none',

facecolor='lightgray'))

# Add legend

legend = plt.legend(facecolor='lightgray', loc='upper left', borderpad=1, fontsize = 10)

#plt.tight\_layout(pad=0)

# Save as image

plt.savefig('Scatterplot\_Resale\_Price\_vs\_Remaining\_Lease', dpi=300)

# Show the plot

plt.show()

Chart 2: Heatmap of Mean Resale Price by Storey Range and Flat Type

Based on the colour scheme, the darker blue hue represents a smaller price while the brighter red hue represents a higher price. From the heatmap, we see 2 positive trends. The first being the higher the storey, the higher the resale price. Secondly, the price is also affected by the type of flat. The more premium the flat, the more expensive the resale price. We see a greater concentration of darker blue between 2-room to 4-room flats, indicating that they are a more budget-friendly option for resale homes. This is also irregardless of storey, with an exception of 4-room flats on the 22nd floor and beyond. Top floors tend to be more expensive due to factors like greater privacy and being less prone to pests or sound pollution. There is an anomaly of a relatively low resale price for the 4-room flat between the 28th-30th storey. External factors such as remaining lease may have skewed the prices. Next, we also observe that the lighter blue and red start to be more prevalent in the 5-room and executive flats. Being a more premium type of flat, prices are naturally much higher than their counterparts. Even the ground-floor units are relatively expensive due to the nature of the flat.

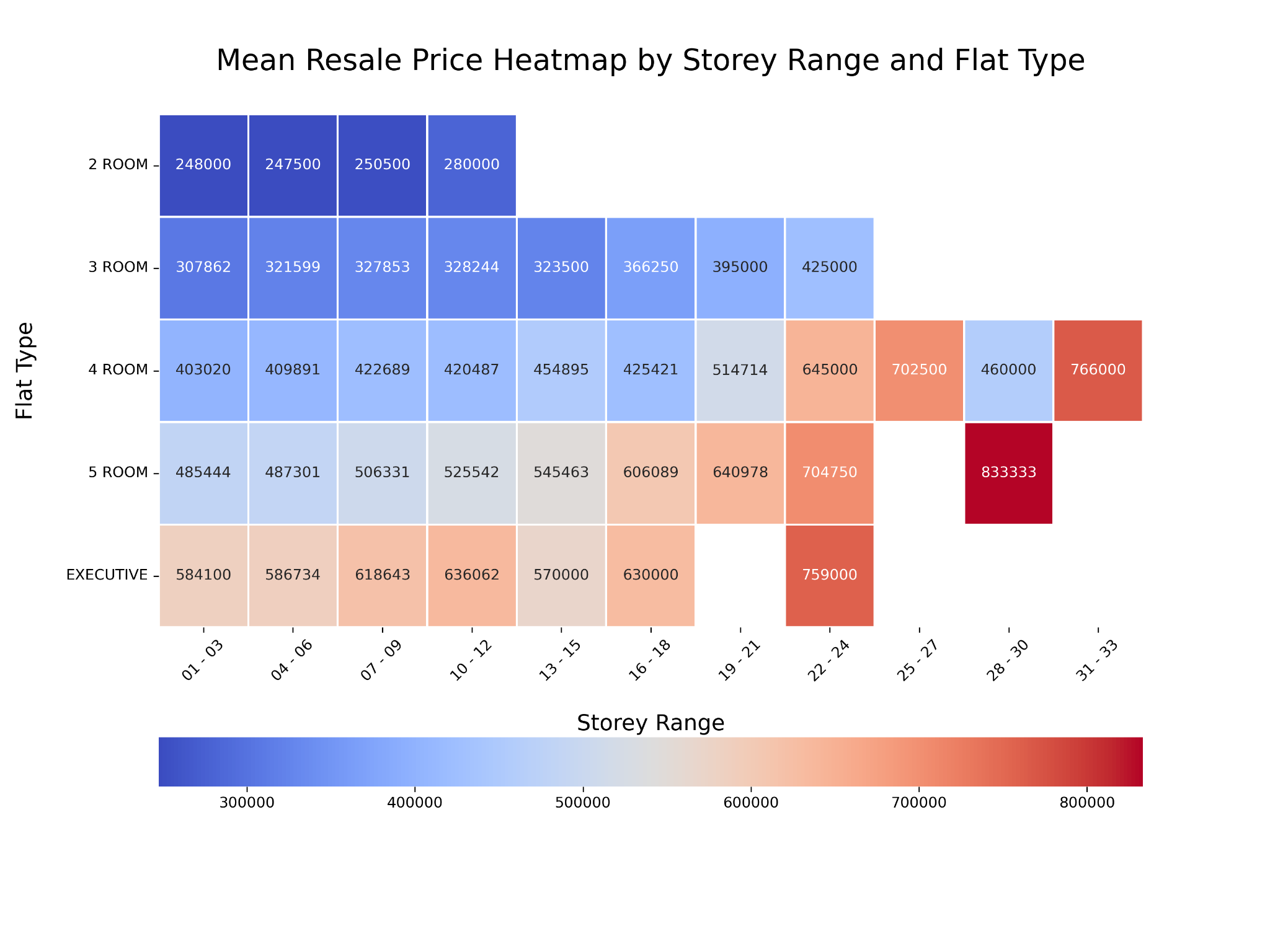


Chart 2 Python code:

import pandas as pd

from matplotlib import pyplot as plt

import seaborn as sns

# Read CSV

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

# Style of graph

#plt.style.use('fivethirtyeight')

# Remove rows with NaN values

df = df.dropna()

#Replace missing values

# Forward fill missing values (replace with the previous valid value)

#df['flat\_type'] = df['flat\_type'].fillna(method='ffill')

#df['street\_name'] = df['street\_name'].fillna(method='ffill')

#To treat resale\_price

#mean\_resale\_price = df['resale\_price'].mean()

#df['resale\_price'].fillna(mean\_resale\_price, inplace=True)

# Create a new column by replacing "TO" with "-"

df['new\_storey\_range'] = df['storey\_range'].str.replace(' TO ', ' - ')

# Group data by storey range and flat type

grouped\_data = df.groupby(['new\_storey\_range', 'flat\_type'])['resale\_price'].mean().reset\_index()

# Sort by mean resale price

grouped\_data = grouped\_data.sort\_values(by='resale\_price', ascending=False)

# Create a heatmap to visualize the results

plt.figure(figsize=(12, 9))

heatmap\_data = grouped\_data.pivot\_table(index='flat\_type', columns='new\_storey\_range', values='resale\_price')

sns.heatmap(heatmap\_data, annot=True, fmt='.0f', cmap='coolwarm', annot\_kws={"size": 10,}, cbar\_kws={'location': 'bottom'}, linewidths=1, linecolor='white')

plt.xlabel('Storey Range', fontsize=15, labelpad=20)

plt.ylabel('Flat Type', fontsize=15, labelpad=20)

plt.title('Mean Resale Price Heatmap by Storey Range and Flat Type', fontsize=20, pad=30)

plt.xticks(rotation=45, fontsize= 10)

plt.yticks(rotation=0, fontsize=10)

# Save as image

plt.savefig('Heatmap\_Mean\_Resale\_Price\_by\_Storey\_range\_and\_flat\_type.png', dpi=300)

plt.tight\_layout()

plt.show()

Chart 2 Table:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **new\_storey\_range** | **01 - 03** | **04 - 06** | **07 - 09** | **10 - 12** | **13 - 15** | **16 - 18** | **19 - 21** | **22 - 24** | **25 - 27** | **28 - 30** | **31 - 33** |
| **flat\_type** |  |  |  |  |  |  |  |  |  |  |  |
| **2 ROOM** | 248000.000000 | 247500.000000 | 250500.000000 | 280000.000000 | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| **3 ROOM** | 307861.761905 | 321598.936170 | 327853.293333 | 328244.000000 | 323500.000000 | 366250.000000 | 395000.000000 | 425000.0 | NaN | NaN | NaN |
| **4 ROOM** | 403019.809524 | 409891.200000 | 422689.018182 | 420487.189189 | 454895.166667 | 425420.571429 | 514714.285714 | 645000.0 | 702500.0 | 460000.000000 | 766000.0 |
| **5 ROOM** | 485444.413793 | 487301.333333 | 506331.259259 | 525541.520000 | 545462.933333 | 606089.333333 | 640977.688000 | 704750.0 | NaN | 833333.333333 | NaN |
| **EXECUTIVE** | 584100.000000 | 586734.260870 | 618642.857143 | 636062.500000 | 570000.000000 | 630000.000000 | NaN | 759000.0 | NaN | NaN | NaN |

Chart 3: Mean Resale Price of Flat Sold by Town and Flat Type

We infer that the most expensive flats sold are located in the central region (Marine parade, Queenstown etc.). Inversely, the more affordable flats are located in the residential regions such as Woodlands. The central tends to be more expensive due to the ease in transportation and access for leisure. Shopping districts, major offices and public transport are easily accessible. Cities' cores are often the most densely populated areas, with little room for further development. Because of this shortage, real estate values have the potential to rise. Such observations can also allude to the socioeconomic status of residents in the region. With the higher concentration of more premium flat-types in the central, it tallies with the heatmap regarding the higher average price for premium flats, regardless of storey.

(500 words)

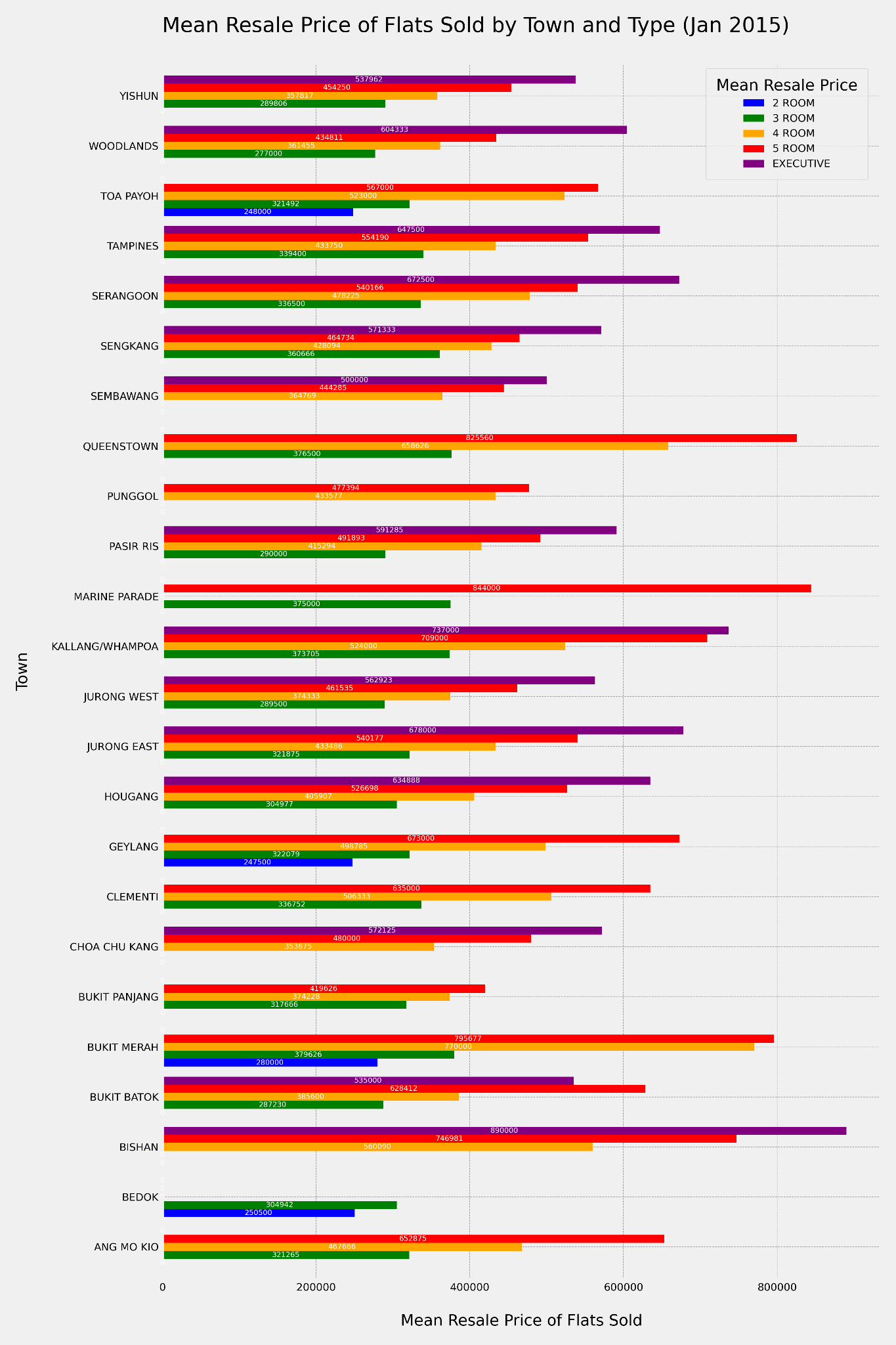


Chart 2 Python code:

# Import libraries

import pandas as pd

import matplotlib.pyplot as plt

# Read CSV

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

# Remove rows with NaN values

df = df.dropna()

# Groupby Flat type

mean\_prices\_by\_town = df.groupby(['flat\_type', 'town'])['resale\_price'].mean().sort\_values(ascending=False)

# Make a copy, to data variable

data = mean\_prices\_by\_town

# Set styling for graph

plt.style.use('fivethirtyeight')

bar\_width = 0.8

graph\_size = (12,18)

# Set colors for bars

colors = {

'2 ROOM': 'blue',

'3 ROOM': 'green',

'4 ROOM': 'orange',

'5 ROOM': 'red',

'EXECUTIVE': 'purple'

}

# Create bar chart

barchart = data.unstack(level=0).plot(kind='barh', figsize=graph\_size, width=bar\_width, color=[colors[col] for col in data.unstack(level=0).columns], )

# Labels and legends of chart

plt.xlabel('Mean Resale Price of Flats Sold', fontsize=15, labelpad=20)

plt.ylabel('Town', fontsize=15, labelpad=20)

plt.title('Mean Resale Price of Flats Sold by Town and Type (Jan 2015)', fontsize=20, pad=30, loc='left')

plt.legend(title='Mean Resale Price', bbox\_to\_anchor=(0.99, 1), loc='upper right', fontsize=10, title\_fontsize=15,borderpad=1)

# Ticks style

plt.xticks(fontsize=10)

plt.yticks(fontsize=10)

# Set grid

plt.grid(True, linestyle='--', linewidth=0.5, color='gray', alpha=0.7)

# Add the mean price in the bar

for bar in barchart.containers:

barchart.bar\_label(bar, fmt='%d', fontsize=7, color='white', label\_type='center')

# Show the chart

plt.tight\_layout()

# Save as image

plt.savefig('Bargraph\_Mean\_Resale\_Price\_of\_Flats\_Sold\_by\_Town\_and\_Type.png', dpi=300)

plt.show()

Chart 3 Table:

flat\_type town

EXECUTIVE BISHAN 890000.0

5 ROOM MARINE PARADE 844000.0

QUEENSTOWN 825560.0

BUKIT MERAH 795677.6

4 ROOM BUKIT MERAH 770000.0

...

2 ROOM BUKIT MERAH 280000.0

3 ROOM WOODLANDS 277000.0

2 ROOM BEDOK 250500.0

TOA PAYOH 248000.0

GEYLANG 247500.0

Name: resale\_price, Length: 83, dtype: float64

# References:

Tamboli, N. (2023, July 14). *Effective Strategies for Handling Missing Values in Data Analysis (Updated 2023)*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/10/handling-missing-value/#:~:text=It%20is%20important%20to%20handle,support%20data%20with%20missing%20values.

# Declaration Page

We, members of group 1, do hereby declare that we each contributed to this assignment and that we collectively agree to a shared grade.

|  |  |  |
| --- | --- | --- |
| Name | Contribution | Signature |
| Loke Kum Wai (Team Lead) | I worked on ideas for part d. I further refined the graphs to be worked on after we chose 3 from everyone’s contribution. |  |
| Ang Kia Loke | I worked on ideas for part d. I suggested answers for part a, b, c. I worked on the insights for the graphs for part d. |  |
| Beverlyn Tan Jiamin | I worked on ideas for part d. I suggested answers for part a, b, c. I further refined and consolidated the answers for part a, b, c. |  |
| Muhammad Bin Osman | I worked on ideas for part d. I suggested answers for part a, b, c. I worked on the insights for the graphs for part d. |  |