****

**ANL252**

**Python for Data Analytics**

**Group-based Assignment**

**July 2023 Presentation**

|  |  |
| --- | --- |
| **Student Name** | **PI Number** |
| Lee Wei Xuan | E2310018 |
| Chua Qi Ying | E2081908 |
| Ong Chiew Ling | B2010698 |
| Ris Nurish | Y2270086 |

**Submission Date: 9 October 2023**

**Question 1(a)**

In order for Python to read the dataset, we would have to import the NumPy and Panda library ‘import numpy as np’ , ‘ import pandas as pd’ we could utilize the Pandas ‘data = pd.read\_csv(“GBA\_HDB.csv”)’ to read a CSV file and convert it into an array.

To easily identify the dimensions of the dataset with Python, we could use the ‘.shape’ attribute and ‘.size’ assuming ‘data’ is my array name:

‘data.shape’ ‘data.size’

For example, for the HDB resale flat prices, upon using .shape and .size , the output would be (1250,11) , (13750) respectively.

**Question 1(b)**

It is crucial to handle missing values in a data set because it would significantly impact our data analysis and modelling of the dataset. Missing values could lead to inaccurate insights and reduce the accuracy of machine learning. In order to use Python to find out the missing variable, we can use .any():

#Assuming ‘data’ is my variable name for my dataset and ‘missing’ is the variable name for my missing variables

data.isnull().any(axis=0)

missing = data.isnull().any(axis=0)

missing[missing==True].index

For the missing values in the HDB resale price, the output would be:

Index(['flat\_type', 'street\_name', 'resale\_price'], dtype='object')

The variables with missing values are flat\_type, street\_name and resale\_price. The following is a detailed table:

month False

town False

flat\_type True

block False

street\_name True

storey\_range False

floor\_area\_sqm False

flat\_model False

lease\_commence\_date False

remaining\_lease False

resale\_price True

dtype: bool

**Question 1(c)**

There are many ways to handle missing data in Python such as ignoring it or replacing it with a pre-defined value using .fillna() method. Another way is to delete the missing data from the rows and columns that have null values. By counting the NaNs in each row and column, we can find the missing data. Missing values exist in a variable if the number of NaNs is larger than zero. Counting the NaNs in the columns ensures the existence of the missing values in each variable.

From the above, input:

missing = data.isnull().any(axis=0)

for x in missing[missing==True].index:

del data[x]

print(data)

lease\_commence\_date remaining\_lease

0 1986 70

1 1981 65

2 1980 64

3 1979 63

4 1980 64

... ... ...

1245 1985 69

1246 1993 77

1247 1988 72

1248 1988 72

1249 1988 72

[1250 rows x 8 columns]

The null data has been deleted with 8 columns remaining.

Alternatively, we can delete row indices with missing values using .drop() method. It drops rows or columns by providing the labels or indices of the items we want to remove.

data.isnull().any(axis=1)

missing = data.isnull().any(axis=1)

missing[missing==True].index

data.drop(axis=0, index = missing[missing==True].index)

Output: 1076 rows × 11 columns

We can obtain the same result using the .dropna() method. It drops rows or columns that contain any missing values based on specified conditions.

data.isnull().any(axis=1)

missing = data.isnull().any(axis=1)

missing[missing==True].index

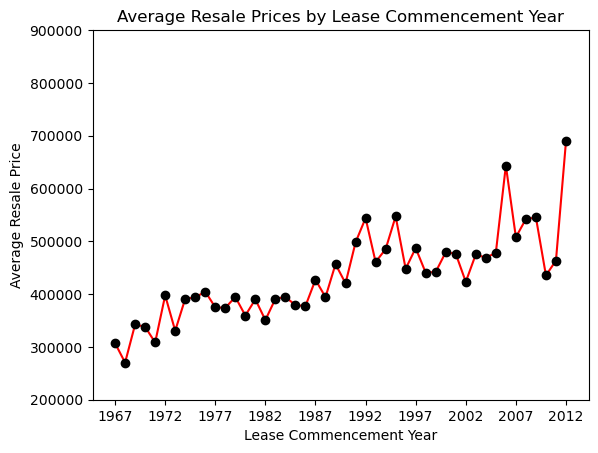
data.dropna(axis=0, how = "any")

Output: 1076 rows × 11 columns

Both ensure that the entire dataset is used for analysis by only keeping those with complete data. Therefore, 174 rows of missing data were deleted.

**Question 1(d)**

**Line Chart**

****

**Figure 1: Average Resale Prices by Lease Commencement Year**

**Python Code**

import pandas as pd

import matplotlib.pyplot as plt

# Read the CSV file

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

# Calculate the average resale price by lease commencement year

avg\_price\_by\_lease\_year = resale\_hdb.groupby(by = 'lease\_commence\_date')['resale\_price'].mean()

# Create a line chart

plt.plot(avg\_price\_by\_lease\_year.index, avg\_price\_by\_lease\_year, color = 'red', marker = 'o', markerfacecolor = 'black', markeredgecolor = 'black')

plt.xlabel('Lease Commencement Year')

plt.ylabel('Average Resale Price')

plt.xticks(ticks = range(1967, 2013, 5), labels = range(1967, 2013, 5))

plt.yticks(ticks = range(200000, 1000000, 100000), labels = range(200000, 1000000, 100000))

plt.title('Average Resale Prices by Lease Commencement Year')

plt.show()

# Display the corresponding table

table = pd.DataFrame(avg\_price\_by\_lease\_year)

print(table)

**Table**

resale\_price

lease\_commence\_date

1967 307125.000000

1968 270250.000000

1969 342625.000000

1970 337952.357143

1971 308400.000000

1972 397571.428571

1973 330577.600000

1974 391200.000000

1975 393636.363636

1976 404794.117647

1977 375055.500000

1978 374034.400000

1979 394547.200000

1980 359507.520000

1981 390099.500000

1982 350836.210526

1983 391085.285714

1984 394682.514286

1985 379062.070588

1986 377433.560976

1987 427239.030303

1988 394283.578947

1989 456744.117647

1990 420535.272727

1991 498000.000000

1992 544043.478261

1993 460876.615385

1994 485166.666667

1995 547229.166667

1996 447910.867925

1997 487268.444444

1998 440480.947368

1999 442159.135135

2000 479343.111111

2001 476459.833333

2002 423225.866667

2003 475364.500000

2004 468734.299130

2005 478461.538462

2006 643461.000000

2007 507750.000000

2008 542250.000000

2009 545142.857143

2010 435452.571429

2011 463586.666667

2012 689428.571429

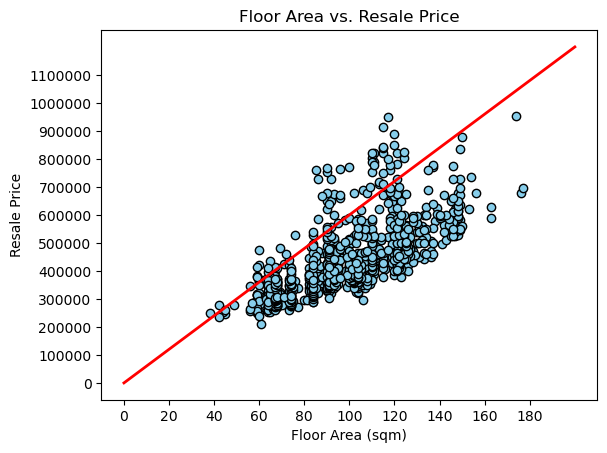
**Insights**

The line chart shows an overall increasing trend in average resale prices, with newer properties having higher average resale prices compared to older ones. This aligns with the common perception that property values typically appreciate over time.

However, there are fluctuations and peaks in average resale prices. From the 1970s to the 1980s, the average resale prices experienced steady and consistent growth, followed by a noticeable increase in average resale prices in the early 1990s. Thereafter, the average resale prices remained relatively stable in subsequent years. In recent years, the slope of the line has been steeper with a significant increase in average resale prices. The variation in price growth could be due to economic growth, changes in market dynamics such as high demand and low supply, as well as government policies.

**Word Count (Insights): 132**

**Scatterplot**

****

**Figure 2: Floor Area vs. Resale Price**

**Python Code**

import pandas as pd

import matplotlib.pyplot as plt

# Read the CSV file

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

# Create a scatter plot of floor area (sqm) vs. resale price

plt.scatter(resale\_hdb['floor\_area\_sqm'], resale\_hdb['resale\_price'], color='skyblue', marker='o', edgecolor='black')

plt.plot([0, 200], [0, 1200000], color = 'red', linewidth = 2)

plt.xlabel('Floor Area (sqm)')

plt.ylabel('Resale Price')

plt.xticks(ticks = range(0, 200, 20), labels = range(0, 200, 20))

plt.yticks(ticks = range(0, 1200000, 100000), labels = range(0, 1200000, 100000))

plt.title('Floor Area vs. Resale Price')

plt.show()

# Display the corresponding table

table = resale\_hdb[['remaining\_lease', 'resale\_price']]

print(table)

**Table**

|  |  |  |
| --- | --- | --- |
|  | remaining\_lease | resale\_price |
| 0 | 70 | 255000.0 |
| 1 | 65 | 275000.0 |
| 2 | 64 | 285000.0 |
| 3 | 63 | 290000.0 |
| 4 | 64 | 290000.0 |
| ... | ... | ... |
| 1245 | 69 | 460000.0 |
| 1246 | 77 | 500000.0 |
| 1247 | 72 | 525888.0 |
| 1248 | 72 | 538000.0 |
| 1249 | 72 | 550000.0 |

[1250 rows x 2 columns]

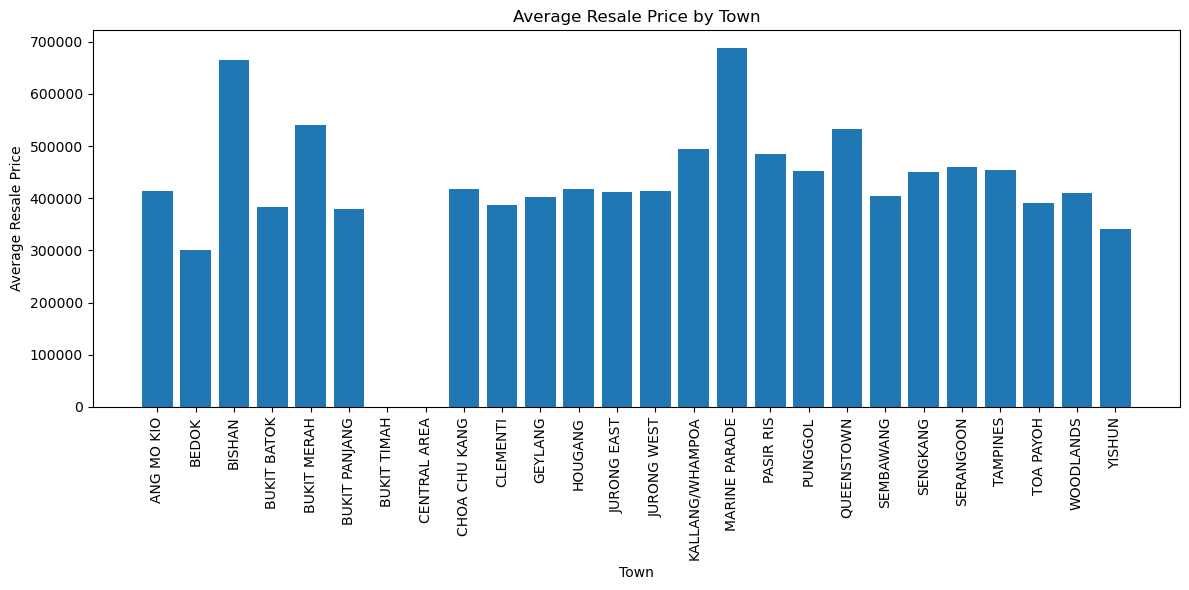
**Insights**

The scatterplot shows a positive linear relationship between floor area and resale price, indicating that larger flats tend to have a higher resale price. This relationship aligns with the common perception that larger floor areas have a higher perceived value and potentially higher resale prices.

However, it is observed that there is a wide variation in resale prices within each floor area range. This highlights the influence of factors such as flat type and town that can significantly impact the resale price. These factors contribute to the presence of outliers with unexpected prices. Additionally, the presence of clusters in the lower-priced range of both floor area and resale price suggests a higher concentration of smaller-sized, more affordable flats. This indicates a strong demand for these smaller-sized flats among buyers who prioritize affordability.

**Word Count (Insights): 132**

**Bar Chart**

****

**Figure 3: Average Resale Price by Town**

**Python Code**

import pandas as pd

import matplotlib.pyplot as plt

# Read the CSV file

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

# Group data by town and calculate the average resale price

town\_prices = resale\_hdb.groupby('town')['resale\_price'].mean().reset\_index()

# Create a bar chart

plt.figure(figsize=(12, 6))

plt.bar(town\_prices['town'], town\_prices['resale\_price'])

plt.xlabel('Town')

plt.ylabel('Average Resale Price')

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

plt.xticks(rotation=90) # Rotate x-axis labels for readability

plt.tight\_layout()

plt.show()

# Display the corresponding table

print(town\_prices)

**Table**

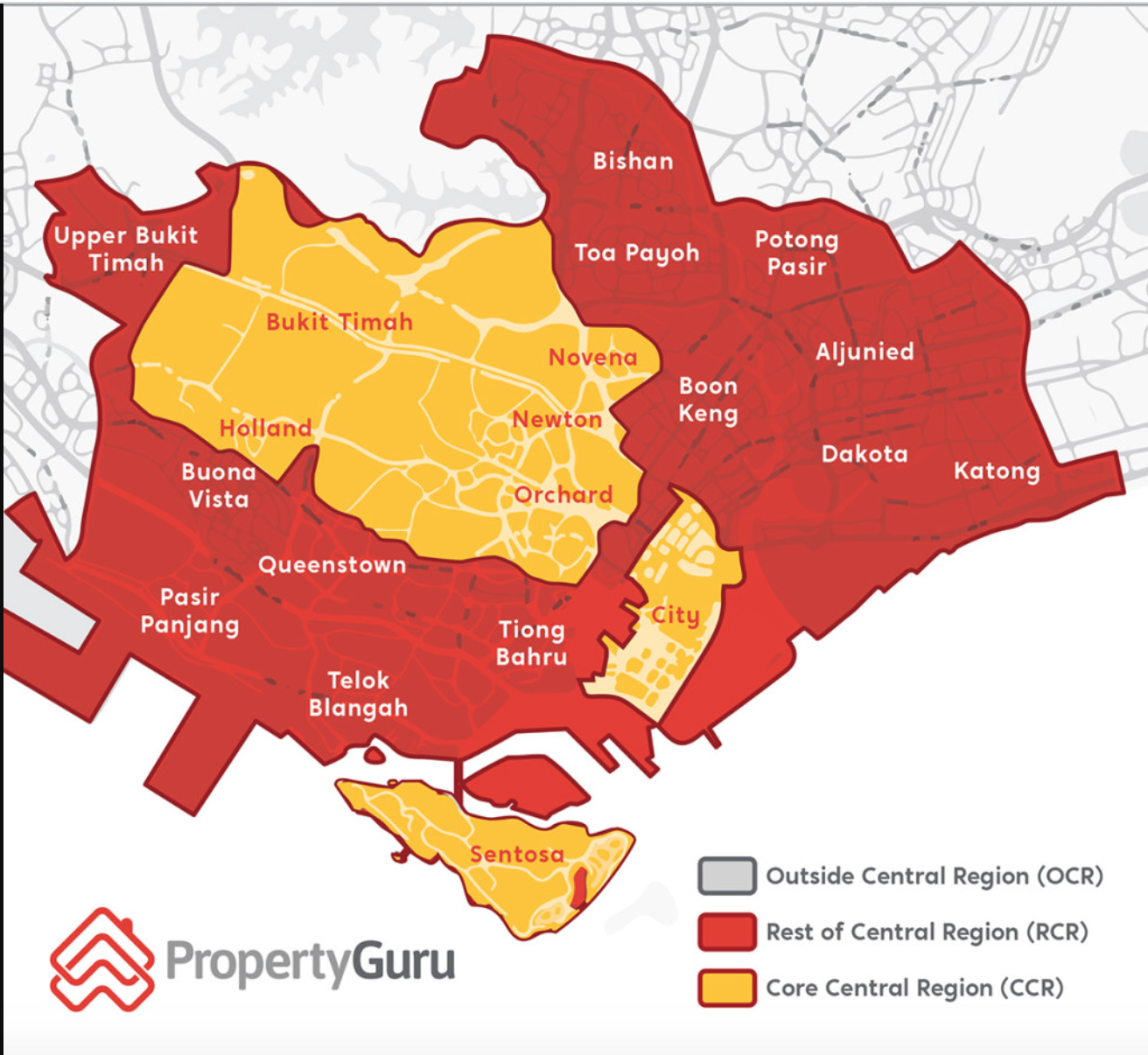
|  |  |  |
| --- | --- | --- |
| **S/N** | **Town** | **Resale Price (round off to nearest dollar)** |
| 0 | ANG MO KIO | 413,376 |
| 1 | BEDOK | 300,209 |
| 2 | BISHAN | 665,644 |
| 3 | BUKIT BATOK | 382,652 |
| 4 | BUKIT MERAH | 540,798 |
| 5 | BUKIT PANJANG | 379,714 |
| 6 | BUKIT TIMAH | NaN |
| 7 | CENTRAL AREA | NaN |
| 8 | CHOA CHU KANG | 417,428 |
| 9 | CLEMENTI | 389,991 |
| 10 | GEYLANG | 403,290 |
| 11 | HOUGANG | 416,970 |
| 12 | JURONG EAST | 411,895 |
| 13 | JURONG WEST | 414,294 |
| 14 | KALLANG/WHAMPOA | 494,667 |
| 15 | MARINE PARADE | 687,667 |
| 16 | PASIR RIS | 485,180 |
| 17 | PUNGGOL | 451,835 |
| 18 | QUEENSTOWN | 533,814 |
| 19 | SEMBAWANG | 404,866 |
| 20 | SENGKANG | 450,047 |
| 21 | SERANGOON | 460,934 |
| 22 | TAMPINES | 454,415 |
| 23 | TOA PAYOH | 391,768 |
| 24 | WOODLANDS | 409,523 |
| 25 | YISHUN | 340,929 |

**Insights**

The data set has a population mean of 429,950.

From the bar chart, we noticed that matured estates that are nearer to town such as Marine Parade, Bishan, Bukit Merah, Kallang/Whampoa and Queenstown have an average selling price higher than the data set mean, capturing 687,667, 665,644, 540,798, 494,667 and 533,814 respectively. Whereas, non-matured estates in the suburban region of Bukit Batok (382,652), Bukit Panjang (379,713), Sembawang (404,865) Woodland (409,523) Yishun (340,929) are lower.

From the above, we infer that resale prices vary by town. HDBs in the Central Region and RCR (URA, 2019) fetch higher prices as represented in the map below:



**Figure 4: Singapore District Map (PropertyGuru, 2023)**

Unfortunately, the resale prices for Bukit Timah and Central Area are missing, thus we cannot verify if HDB resale prices in CCR are higher than those in RCR. We believe they are.

One interesting observation is that Toa Payoh, also in RCR, had a lower-than-average average of $391,768. Our findings in the line chart and scatter plot above show that older and smaller flats have lower resale prices. Therefore, we infer that the HDB units sold in Toa Payoh town are older and/or smaller. While proximity to the city significantly influences HDB resale prices, we concluded that other factors such as floor area and the remaining lease of the flat also play a role.

**Word Count (Insights): 224**

**References**

URA. (2019). *Data Dictionary*. Retrieved from URA website on 28 Sep 2023. <https://www.ura.gov.sg/reis/dataDictionary#:~:text=Core%20Central%20Region%20(CCR)%3A,Rochor%2C%20Orchard%20and%20Downtown%20Core>.

PropertyGuru Editorial Team. (2023, September 8). *Singapore District Map: Defining the CCR, RCR and OCR by the 28 Districts.* Retrieved on 28 Sep 2023. <https://www.propertyguru.com.sg/property-guides/ccr-ocr-rcr-region-singapore-ura-map-21045>