

**ANL252**

**Python for Data Analytics**

**Group Based Assignment**

**Jul 2023**

**Submitted by:**

|  |  |
| --- | --- |
| **Name** | **PI Number** |
| Ong Sheng Da | Z2281575 |
| Chua Yu Zhe | J2311231 |
| Gwendolyn Yeo Zi Hui | Y2310224 |
| Villanueva Van Hally Borines | H2072822 |

**Tutorial Group**: T03 Group 11

**Instructor’s Name**: Professor Kumar Munish

**Submission Date**: 13/10/23

**Declaration Page**

We, members of group 11, do hereby declare that we each contributed to this assignment and that we collectively agree to a shared grade.

|  |  |  |
| --- | --- | --- |
| Name | Contribution | Signature |
| Chua Yu Zhe (Team Lead) | I did questions 1 (a), (b), (c), (d) | Yu Zhe |
| Gwendolyn Yeo Zi Hui | I did questions 1 (a), (b), (c), (d) | Gwendolyn |
| Villanueva Van Hally Borines | I did questions 1 (a), (b), (c), (d) | Villanueva |
| Ong Sheng Da | I did questions 1 (a), (b), (c), (d) | Sheng Da |

**Question 1 (a)**

To read the dataset in Python, we will perform the following steps:

Step 1: Upload ‘GBA\_HDB’ to Jupyter

Step 2: Identify type of file, in this case it is a csv file.

Step 3: Use the correct read function. “pd.read\_csv”

Step 4: to identify the dimensions of the dataset, we will use “hdb.shape”

Step 5: to identify the variables of the dataset, we will use “hdb.columns”

Doing so we are able to get Python to read the dataset in “GBA\_HDB” and identify the dimensions of the dataset. The code is as follows:

import pandas as pd

# Read file

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

print(hdb.shape)

print(hdb.columns)

In this case, “GBA\_HDB” has **1250 rows and 11 columns**. We also know the types of variables in this data set.

**Question 1 (b)**

To identify the variables with missing values we used the following code.

import pandas as pd

# Read file

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

# Locate missing values

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

hdb.isnull().sum(axis = 1)

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

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

clean\_hdb = hdb.isnull().any()

clean\_hdb[clean\_hdb == True].index

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

Using the code above we know that there are 175 missing values in the dataset, “flat\_type” has 40, “street\_name” has 1 and “resale\_price” has 134.

It is important to properly handle missing values from our data set. Missing values may lead to inaccuracy in our findings. For example, if we calculate the mean of a variable with missing values, the result may be skewed and not an accurate represent the population (Lee, 2022).

Additionally, inaccurate findings can lead to user questioning the credibility and reliability of our data. Users will be less likely to trust information which is inaccurate (Lee, 2022).

Incomplete data can affect our data quality. A reduction in sample size can make it difficult to detect meaningful information from our data (Lee, 2022).

In a business setting, bad data can lead to loss of income, potentially fines and penalties as well as loss of customers (Sarangdhar, 2022).

Therefore, it is important for use to ensure that missing values are addressed appropriately to ensure that our users can make meaningful decisions and find meaningful insights using our data.

**Question 1 (c)**

There are several ways to treat missing data. Firstly, we could eliminate all records for which has missing attributes. Secondly, we could encode and identify missing data such as replacing them with “NIL”. Thirdly, we could substitute the missing data with another value such as the mean or mode for the variable. Fourthly, we could seek the help of experts to recommend possible values to replace missing data. Finally, we can use regression, it involves the use of other attributes to predict the missing values. For example, we could use the “floor\_area\_sqm” of the property to determine the flat\_type.

In this case, we choose two methods namely elimination and regression. We decided to eliminate the record with the missing “street\_name” and “flat\_type” as eliminating 41 records from the 1250 records data set will not have little effect on our findings. However, the remaining 134 missing values from “resale\_price” make up 10.72% of the entire data set, a substantial amount. Therefore, we choose to use regression as it can effectively predict the missing values in “resale\_price” using “floor\_area\_sqm”, “flat\_type”, “town”, “flat\_model” and “storey\_range”. The new dimensions are **1209 rows and 14 columns**.

import pandas as pd

import matplotlib.pyplot as plt

import numpy as np

from sklearn.linear\_model import LinearRegression

# Read file

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

# Assign a number to each variable

def generate\_numerical\_identifier(elements):

identifier\_mapping = {}

for i, element in enumerate(elements):

identifier\_mapping[element] = i

return identifier\_mapping

town\_mapping = generate\_numerical\_identifier(hdb.town.unique())

flat\_type\_mapping = generate\_numerical\_identifier(hdb.flat\_type.unique())

flat\_model\_mapping = generate\_numerical\_identifier(hdb.flat\_model.unique())

hdb['town\_encoded'] = hdb.town.map(town\_mapping)

hdb['flat\_type\_encoded'] = hdb.flat\_type.map(flat\_type\_mapping)

hdb['flat\_model\_encoded'] = hdb.flat\_model.map(flat\_model\_mapping)

# Find middle of 'storey\_range'

def convert\_storey\_range(storey\_range):

storey\_start, \_, storey\_end = storey\_range.split(' ')

return (int(storey\_start) + int(storey\_end)) / 2

hdb['storey\_range\_numeric'] = hdb.storey\_range.apply(convert\_storey\_range)

hdb\_missing = hdb[hdb.resale\_price.isnull()]

hdb\_complete = hdb.dropna(subset=['resale\_price'])

# Create a linear regression model

model = LinearRegression()

# Independent variables

X = hdb\_complete[['town\_encoded', 'flat\_type\_encoded', 'flat\_model\_encoded', 'storey\_range\_numeric', 'floor\_area\_sqm']]

# Dependent variable

y = hdb\_complete['resale\_price']

model.fit(X, y)

# Predict missing 'resale\_price' values

X\_missing = hdb\_missing[['town\_encoded', 'flat\_type\_encoded', 'flat\_model\_encoded', 'storey\_range\_numeric', 'floor\_area\_sqm']]

predicted\_prices = model.predict(X\_missing)

# Replace missing values with predicted values

hdb.loc[hdb['resale\_price'].isnull(), 'resale\_price'] = predicted\_prices

hdb\_clean = hdb.dropna()

hdb\_clean

**Question 1 (d)**

The Python code is as follows:

import pandas as pd

import matplotlib.pyplot as plt

import numpy as np

# Read file

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

# Create table for storey\_range and resale price

storey\_price = hdb\_clean.groupby('storey\_range')['resale\_price'].mean().reset\_index()

# Create line chart storey\_range vs resale price

plt.plot (storey\_price.storey\_range, storey\_price.resale\_price, color='b', marker='.',linewidth=2, markersize=12)

plt.xlabel("Storey Range")

plt.ylabel("Average Resale Price")

plt.title("Storey Range vs Resale Price")

plt.xticks(rotation=45)

# Show chart and table

print(storey\_price)

plt.show()

# Create table for flat\_model, resale price and floor area

model\_price\_floor = hdb\_clean.groupby('flat\_model').agg({'resale\_price': 'mean', 'floor\_area\_sqm': 'mean'}).reset\_index()

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

# Create bar and line chart

# 1st y-axis for average resale price

plt.bar(model\_price\_floor.flat\_model, model\_price\_floor.resale\_price, color='b', alpha=0.2, label='Avg Resale Price')

plt.xlabel('Flat Model')

plt.ylabel('Average Resale Price', color='b')

plt.tick\_params(axis='y', labelcolor='b')

plt.xticks(rotation=90)

plt.legend(loc='best')

# 2nd y-axis for average floor area sqm

plt.twinx()

plt.plot(model\_price\_floor.flat\_model, model\_price\_floor.floor\_area\_sqm, color='r', marker='o', label='Avg Floor Area sqm')

plt.ylabel('Average Floor Area sqm', color='r')

plt.tick\_params(axis='y', labelcolor='r')

plt.legend(loc='best')

# Show chart and table

print(model\_price\_floor)

plt.tight\_layout()

plt.show()

# Create table for town, flat model, resale price

heatmap\_data = hdb\_clean.pivot\_table(index='town', columns='flat\_model', values='resale\_price')

# Get the unique flat types and flat models

towns = heatmap\_data.index

flat\_models = heatmap\_data.columns

# Convert the pivot table to a NumPy array

heatmap\_array = heatmap\_data.values

# Create the heatmap

plt.imshow(heatmap\_array, cmap='coolwarm', interpolation='nearest')

plt.title('Heatmap of Resale Price by Town and Flat Model')

plt.xlabel('Flat Model')

plt.ylabel('Town')

plt.xticks(np.arange(len(flat\_models)), flat\_models, rotation=90)

plt.yticks(np.arange(len(towns)), towns)

# Add a colorbar

cbar = plt.colorbar()

cbar.set\_label('Resale Price')

# Show chart and table

print(heatmap\_data)

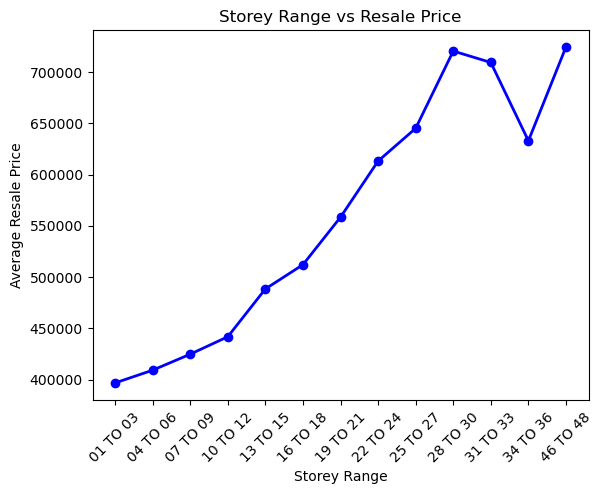
plt.show()

**Line chart of storey range vs average resale price**

From the line chart below, we can see an upwards trend in term of the average resale value of the properties based on the storey range. This aligns with the basic assumption that the higher the property the more expensive it is. However, this trend stops from storey range 31 to 33 and 34 to 36 before rising again.

Therefore, if you were looking for a property between the price range of $60,000 to $70,000, you could consider purchasing a property of storey 31 to 36 rather than 22 to 30 as it could be considered a steal.

storey\_range resale\_price  
0 01 TO 03 396661.532510  
1 04 TO 06 409118.079071  
2 07 TO 09 424721.716284  
3 10 TO 12 441759.847095  
4 13 TO 15 488397.065062  
5 16 TO 18 512052.117170  
6 19 TO 21 558426.391190  
7 22 TO 24 613167.418726  
8 25 TO 27 645125.502152  
9 28 TO 30 720504.609906  
10 31 TO 33 709351.769696  
11 34 TO 36 632945.461102  
12 46 TO 48 724515.555414



**Line and bar chart for flat model, average resale price and average floor area**

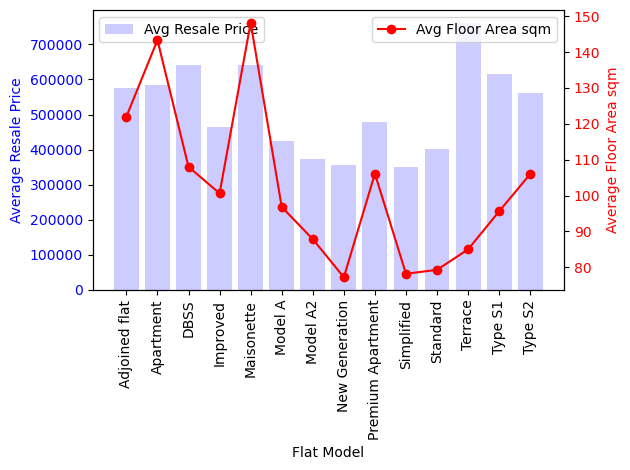
The chart below shows the average resale price and average floor area for each type of flat model. New generation (77.23 sqm) and simplified (78.20 sqm) flat models have the smallest floor area, they thus have the lowest resale prices of $356,645.04 and $350,806.21 respectively. Maisonette (148.17 sqm) and apartment (143.33 sqm) have the greatest floor area and as expected it would have a high resale price of $640,063.20 and $584,547.05 respectively.

What is interesting is that the terrace has one of the lowest average floor areas (85 sqm) but the highest average resale price of all the properties $760,000. This could be due to the fact that it is a landed property and therefore has the highest $/sqm of $8941.17 per sqm.

If we look at the ratio between the average resale price and the floor area, apartments have the smallest $/sqm of $4078.23 per sqm. This means if the buyer can afford to stay in an apartment, it is the most worthwhile property.

However, if the buyer finds the apartment expensive, they could also look at model A2 flats where the $/sqm is $4253.75 per sqm. The $/sqm is slightly higher, but the overall property is a lot cheaper.

flat\_model resale\_price floor\_area\_sqm price\_per\_sqm  
0 Adjoined flat 575000.000000 122.000000 4713.114754  
1 Apartment 584547.047996 143.333333 4078.235219  
2 DBSS 642000.000000 108.000000 5944.444444  
3 Improved 465576.405357 100.610390 4627.518163  
4 Maisonette 640063.203923 148.166667 4319.886641  
5 Model A 424228.053689 96.825153 4381.382718  
6 Model A2 373699.555556 87.851852 4253.747049  
7 New Generation 356645.044196 77.232365 4617.818495  
8 Premium Apartment 479304.106054 106.092593 4517.790492  
9 Simplified 350806.207379 78.200000 4486.012882  
10 Standard 400607.520821 79.305263 5051.461969  
11 Terrace 760000.000000 85.000000 8941.176471  
12 Type S1 616101.676386 95.625000 6442.893348  
13 Type S2 560451.309303 106.000000 5287.276503



**Heatmap of town, flat model and average resale price**

The chart below shows a heat map for the flat models, town and the resale price. At a glance we can tell that the Maisonette in Bishan is the most expensive property followed by the standard flat in marine parade. The cheapest property is the standard property in Geylang. We can tell that the location of the property has a profound affect on the resale price. As seen from the standard flats where the price range is significant.

We can also see that new generation and model 2A flats have a relatively low price no matter where it is located. What I found interesting is that central area properties price is around the middle range. I assumed that central area considered of prime real estate and therefore would have a higher resale price.

A graph of different colored squares

Description automatically generated with medium confidence

|  |  |
| --- | --- |
| flat\_model Adjoined flat Apartment DBSS Improved \  town  ANG MO KIO NaN NaN NaN 606857.142857  BEDOK NaN 637389.833456 NaN 407835.693464  BISHAN NaN NaN NaN 742377.600000  BUKIT BATOK NaN NaN NaN 527269.714286  BUKIT MERAH NaN NaN NaN 632666.500000  BUKIT PANJANG NaN NaN NaN 498486.576214  BUKIT TIMAH NaN NaN NaN NaN  CENTRAL AREA NaN NaN NaN 388099.675143  CHOA CHU KANG NaN 567000.000000 NaN 480000.000000  CLEMENTI NaN NaN NaN 635000.000000  GEYLANG 575000.0 NaN NaN 421820.888889  HOUGANG NaN 638500.000000 NaN 457288.800000  JURONG EAST NaN NaN NaN 516177.600000  JURONG WEST NaN 554300.000000 NaN 420143.764706  KALLANG/WHAMPOA NaN NaN NaN 441285.714286  MARINE PARADE NaN NaN NaN 375000.000000  PASIR RIS NaN 585000.000000 NaN 476450.000000  PUNGGOL NaN NaN NaN 469388.800000  QUEENSTOWN NaN NaN NaN 515440.000000  SEMBAWANG NaN 519000.000000 NaN 440833.333333  SENGKANG NaN 570000.000000 NaN 462302.800000  SERANGOON NaN NaN NaN 504714.285714  TAMPINES NaN 670000.000000 642000.0 534923.076923  TOA PAYOH NaN NaN NaN 366638.875000  WOODLANDS NaN 615083.333333 NaN 419876.695652  YISHUN NaN 538000.000000 NaN 378333.333333 | flat\_model Maisonette Model A Model A2 New Generation \  town  ANG MO KIO NaN 590444.000000 NaN 373679.102041  BEDOK 609292.199115 421963.258637 NaN 361428.940233  BISHAN 890000.000000 559052.917453 NaN 438384.085318  BUKIT BATOK 535000.000000 406000.000000 400000.000000 304421.052632  BUKIT MERAH NaN 483321.429661 NaN 429868.080200  BUKIT PANJANG 597690.492413 355492.000000 NaN NaN  BUKIT TIMAH NaN 496242.883480 NaN NaN  CENTRAL AREA NaN NaN NaN NaN  CHOA CHU KANG 581333.333333 355545.000289 365000.000000 333333.333333  CLEMENTI NaN NaN NaN 362190.000000  GEYLANG NaN 479500.000000 NaN 416564.666667  HOUGANG 632000.000000 408154.222222 NaN 319583.333333  JURONG EAST 678000.000000 463750.000000 NaN 347025.866667  JURONG WEST 545000.000000 364373.147059 356600.000000 309750.000000  KALLANG/WHAMPOA 737000.000000 606076.923077 NaN 399625.000000  MARINE PARADE NaN NaN NaN NaN  PASIR RIS 614333.333333 425908.173913 NaN NaN  PUNGGOL NaN 408000.000000 NaN NaN  QUEENSTOWN NaN 686987.555556 NaN NaN  SEMBAWANG NaN 369555.555556 349629.333333 NaN  SENGKANG NaN 393484.400000 399363.636364 NaN  SERANGOON 672500.000000 526937.500000 NaN 415307.692308  TAMPINES 644285.714286 410284.153846 NaN 377250.000000  TOA PAYOH NaN 570600.000000 NaN 443000.000000  WOODLANDS 581750.000000 365038.608696 350000.000000 307333.333333  YISHUN 537944.000000 353280.380952 NaN 320230.769231 |
| flat\_model Premium Apartment Simplified Standard Terrace \  town  ANG MO KIO NaN NaN 650000.000000 NaN  BEDOK 561405.490104 405982.656861 566366.175501 NaN  BISHAN 588000.000000 479595.003973 NaN NaN  BUKIT BATOK NaN 298000.000000 NaN NaN  BUKIT MERAH NaN 382319.051510 355220.606373 NaN  BUKIT PANJANG 514575.893586 321500.000000 NaN NaN  BUKIT TIMAH NaN NaN 521724.204316 NaN  CENTRAL AREA NaN NaN NaN NaN  CHOA CHU KANG 407222.000000 NaN NaN NaN  CLEMENTI NaN NaN NaN NaN  GEYLANG NaN NaN 255000.000000 NaN  HOUGANG NaN 352285.714286 NaN NaN  JURONG EAST NaN NaN 390000.000000 NaN  JURONG WEST 494166.666667 318333.333333 418000.000000 NaN  KALLANG/WHAMPOA NaN 370000.000000 413750.000000 760000.0  MARINE PARADE NaN NaN 844000.000000 NaN  PASIR RIS NaN NaN NaN NaN  PUNGGOL 453156.606061 NaN NaN NaN  QUEENSTOWN NaN NaN 331600.000000 NaN  SEMBAWANG 414722.000000 NaN NaN NaN  SENGKANG 498238.844000 NaN NaN NaN  SERANGOON NaN 377875.000000 NaN NaN  TAMPINES 532500.000000 375888.888889 NaN NaN  TOA PAYOH NaN NaN 277666.666667 NaN  WOODLANDS 456490.666667 334600.000000 398000.000000 NaN  YISHUN NaN 315957.411765 NaN NaN | lat\_model Type S1 Type S2  town  ANG MO KIO NaN NaN  BEDOK NaN NaN  BISHAN NaN NaN  BUKIT BATOK NaN NaN  BUKIT MERAH NaN NaN  BUKIT PANJANG NaN NaN  BUKIT TIMAH NaN NaN  CENTRAL AREA 602656.944627 546671.24983  CHOA CHU KANG NaN NaN  CLEMENTI NaN NaN  GEYLANG NaN NaN  HOUGANG NaN NaN  JURONG EAST NaN NaN  JURONG WEST NaN NaN  KALLANG/WHAMPOA NaN NaN  MARINE PARADE NaN NaN  PASIR RIS NaN NaN  PUNGGOL NaN NaN  QUEENSTOWN NaN NaN  SEMBAWANG NaN NaN  SENGKANG NaN NaN  SERANGOON NaN NaN  TAMPINES NaN NaN  TOA PAYOH NaN NaN  WOODLANDS NaN NaN  YISHUN NaN NaN |

**Reference**

Lee, K. H. C. (2022). *ANL303 Fundamentals of data mining*. Singapore: Singapore

University of Social Sciences.

Sarangdhar, S. (2022, August 3). *The impact of poor data quality | ZoomInfo*. Pipeline Blog. https://pipeline.zoominfo.com/operations/poor-data-quality-impact