A red and blue text on a black background

Description automatically generated

**ANL252**

**Python for Data Analytics**

**Group-Based Assignment**

**July 2023 Presentation**

**Declaration Page**

We, members of group 14 , do hereby declare that we each contributed to this assignment and that we collectively agree to a shared grade.

|  |  |  |
| --- | --- | --- |
| Name | Contribution | Signature |
| Yiew Sher Kym Lynette (Y2181412) | I did questions 1(c) | Lynette yiew |
| Nur Ruzainatul Husnaa Binte Abdul Jalil (Q2310679) | I did question 1(d) | Nur |
| Nur Ameerah Binte Nassirudin (Y2170573) | I did question 1(d) | Ameerah |
| Poh Sze Ying (M2170235) | I did question 1(d) | Sze Ying |
| Lee Jay Hoon Jenny (Y2070655) | I did question 1(a), 1(b) | Jenny |

**Q1(a)**

The following piece of code loads the dataset and identifies its dimensions to be (1250, 11) - which is 1250 rows and 11 columns.

|  |
| --- |
| *#to import pandas*  import pandas as pd  *#to import data from excel file*   hdb\_df = pd.read\_csv("GBA\_HDB.csv")  *#to generate the dimensions output*  dimensions = hdb\_df.shape |

The dimensions are obtained by calling the 'shape' of the array.

**Q1(b)**

The missing values can be identified using the following piece of code. The code identifies all the rows that have a missing value in any of its columns.

|  |
| --- |
| *#to import pandas*  import pandas as pd  *#to import data from excel file*   hdb\_df = pd.read\_csv("GBA\_HDB.csv")  *#to identify rows with missing data*  null\_hdb\_df = hdb\_df[hdb\_df.isnull().any(axis=1)]  *#to identify rows with missing resale price data*   null\_resale\_df = null\_hdb\_df[null\_hdb\_df["resale\_price"].isnull()]  *#to identify rows with missing flat type data*   null\_flat\_df = null\_hdb\_df[null\_hdb\_df["flat\_type"].isnull()] |

There are 134 rows with missing resale prices and 40 rows with missing flat types.

It is necessary to handle missing values as it can lead to inaccuracies when performing calculations or analysis on the data. Statistics like the mean, median, and variance, will be affected when missing values are not taken into account or not handled properly. This is especially true for scenarios where the absent information relates to sensitive or critical variables. An example would be customer profiling in marketing, where neglecting missing data on customer preferences or behavior can lead to inaccurate targeting or recommendations, resulting in customer dissatisfaction.

Managing missing values is essential to ensure that these metrics faithfully reflect the accessible information.

Word count: 145 words

**Q1(c)**

Missing data can be treated with the various Pandas functions, such as dropna(), fillna(), interpolate() and ffill().

The functions 'fillna()', 'interpolate()', and 'ffill()' is used to populate the missing data, while dropna() deletes the rows/columns containing them. The context for using each function depends on the importance of the missing data, and how much it skews the analysis we are trying to perform on the data.

If filling in missing data affects the analysis accuracy significantly, removing them would be better. In this case, the missing data are in the "resale\_prices" column (134 occurrences) and "flat\_type" column (40 occurrences). In the group's opinion, the resale prices are more meaningful compared to the flat type. This means that they have a high impact on the analysis we will be conducting. If we simply interpolated the resale data, our analysis would not be accurate and would be skewed significantly. Thus, the rows with missing resale prices should be deleted. For "flat\_type", it can be filled with ffill() since the values are a string.

The following piece of code removes the rows with missing resale prices and fills the missing flat types.

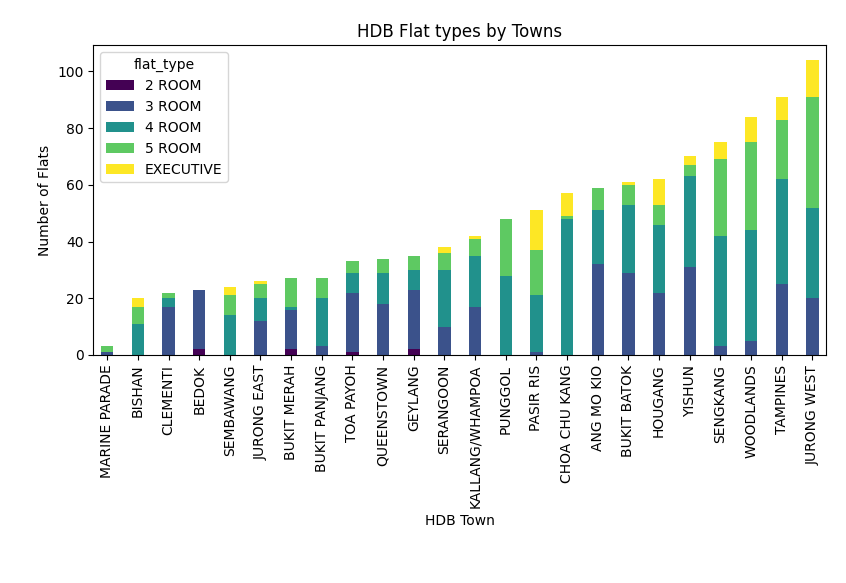
|  |
| --- |
| *#to import pandas*  import pandas as pd  *#to import data from excel file*   hdb\_df = pd.read\_csv("GBA\_HDB.csv")  *#to remove the rows with missing resale price*  no\_na\_df = hdb\_df.dropna(subset = ["resale\_price"])  *#to fill the rows with missing flat types*  no\_na\_df["flat\_type"] = no\_na\_df["flat\_type"].ffill()  *#to print the data after after missing data is managed*  print(no\_na\_df) |

Word count: 189 words

**Q1(d)**

Based on the consensus in 1(c), the group removed rows with missing resale prices to maintain analysis accuracy. Hence, the following analysis and charts were based on this adjusted dataset.

**Chart 1**

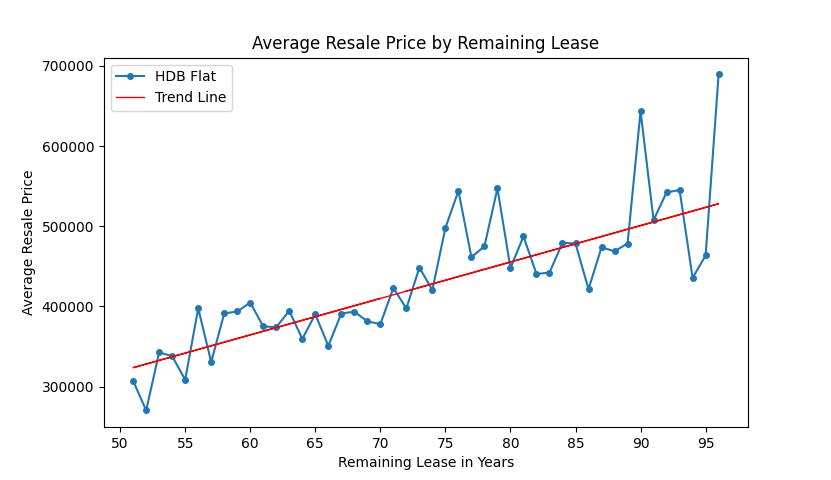


The stacked bar graph displays the sale of the various room types offered across the HDB Towns.

A quick glance reveals that 4-room apartments are more common than the other types of flats. Among the towns, Marine Parade has the lowest number of flat sales, with the majority of these sales being 5-room flats. The top 3 towns with the greatest number of flats sold are Jurong West, Tampines, and Woodlands. None of these 3 towns sold flats are 2-room flats. Below is the code used to generate this stacked bar graph.

|  |
| --- |
| *#to import graphs import matplotlib.pyplot as plt #to import data import pandas as pd  #import data from excel file hdb\_df = pd.read\_csv("GBA\_HDB.csv")  #drop rows with missing data and fill flat\_types. no\_na\_df = hdb\_df.dropna(subset = ["resale\_price"]) no\_na\_df["flat\_type"] = no\_na\_df["flat\_type"].ffill()  # Prepare and sort the data town\_flat\_counts = no\_na\_df.groupby(['town', 'flat\_type']).size().unstack(fill\_value=0) town\_flat\_counts['total'] = town\_flat\_counts.sum(axis=1) town\_flat\_counts\_sorted = town\_flat\_counts.sort\_values(by='total') town\_flat\_counts\_sorted = town\_flat\_counts\_sorted.drop('total', axis=1)  # Display 'town' in string town\_flat\_counts\_sorted = town\_flat\_counts\_sorted.reset\_index()  # Create a stacked bar chart to visualize the counts of flat types in each town. town\_flat\_counts\_sorted.plot(x='town', kind='bar', stacked=True, colormap='viridis', width=0.4)  # plot histogram plt.xlabel("HDB Town") plt.ylabel("Number of Flats") plt.title("HDB Flat types by Towns") plt.tight\_layout()  plt.show()* |

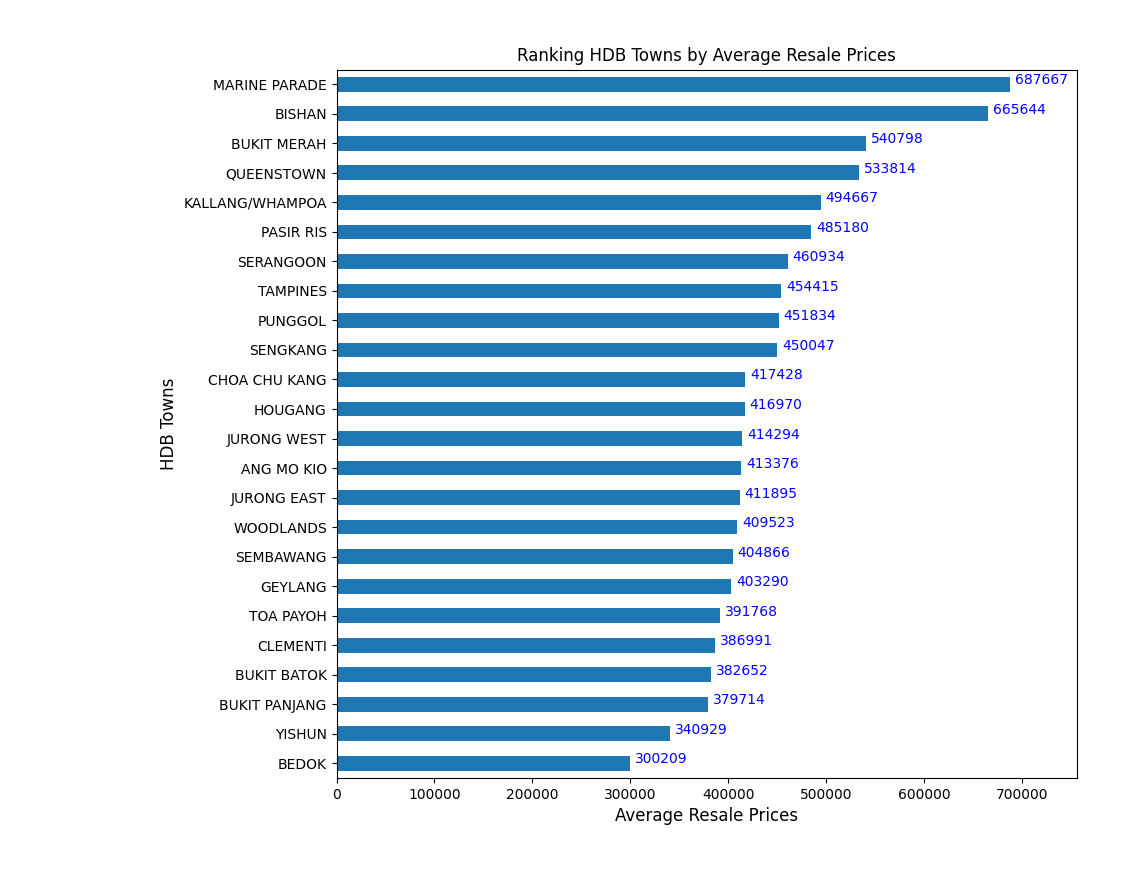
**Chart 2**



This chart compares the relation between the average resale prices of the HDB flats and their remaining leases. By adding **a linear trend line**, we can visualise this relationship better. The graph reveals that, on average, flats with longer remaining leases tend to have higher values and selling prices. Below is the code used to generate this line chart.

|  |
| --- |
| *#to import graphs* import matplotlib.pyplot as plt *#to import data* import pandas as pd import numpy as np  *#import data from excel file* hdb\_df = pd.read\_csv("GBA\_HDB.csv")  *#drop rows with missing data and fill flat\_types.* no\_na\_df = hdb\_df.dropna(subset = ["resale\_price"]) no\_na\_df["flat\_type"] = no\_na\_df["flat\_type"].ffill()  grouped\_data = no\_na\_df.groupby('remaining\_lease')['resale\_price'].mean()  *# Create the line plot* plt.plot(grouped\_data.index, grouped\_data.values, marker='o', markersize='4', label='HDB Flat')  *# Get the data ranges for x and y* x = no\_na\_df['remaining\_lease'] y = no\_na\_df['resale\_price']  *# Customise the graph*  plt.xticks(np.arange(min(x)-1, max(x), 5)) plt.xlabel('Remaining Lease in Years') plt.ylabel('Average Resale Price') plt.title('Average Resale Price by Remaining Lease')  *# Add linear trend line* z = np.polyfit(x,y,1) p = np.poly1d(z) plt.plot(x, p(x), color="red", linewidth=1, label='Trend Line') plt.legend() *# Display the plot* plt.show() |

**Chart 3**



The following horizontal bar graph ranks the HDB Towns by their average resale prices.

The graph reveals that on average, the towns Marine Parade and Bishan have the highest resale prices, compared to Yishun and Bedok. Making a cross-reference to **Chart 1**, it can be observed that though Marine Parade and Bishan have the least number of flats sold, these flats still fetched higher prices compared to Jurong West and Yishun. Below is the code used to generate this horizontal bar graph.

|  |
| --- |
| *#to import graphs* import matplotlib.pyplot as plt *#to import data* import pandas as pd  *#import data from excel file* hdb\_df = pd.read\_csv("GBA\_HDB.csv")  *#drop rows with missing data and fill flat\_types.* no\_na\_df = hdb\_df.dropna(subset = ["resale\_price"]) no\_na\_df["flat\_type"] = no\_na\_df["flat\_type"].ffill()  *# calculate*  town\_prices = no\_na\_df.groupby('town')['resale\_price'].mean()  town\_prices = town\_prices.sort\_values(ascending=True) town\_prices.plot(kind='barh') plt.xlabel("Average Resale Prices", fontsize=12) plt.ylabel("HDB Towns", fontsize=12) plt.title("Ranking HDB Towns by Average Resale Prices")  *# Label bars* for i, v in enumerate(town\_prices):  plt.text(v+5000, i, str("{:.0f}".format(v)), color='blue')  plt.xlim(0,np.max(hdb\_flat)\*1.1) plt.tight\_layout() plt.show() |

Word count: 263 words