**ANL252**

**Python for Data Analytics**

**Group Based Assignment**

**July 2023 Presentation**

|  |  |
| --- | --- |
| Name | Student ID |
| Hilmi Bin Ishak | N2181344 |
| Ong Ming Da | B1611465 |
| Tan Pei Ling | W2211367 |
| Pang Shi Da Lawrence | Q2310757 |

|  |  |
| --- | --- |
| Tutorial Group: | T03 |
| Instructor’s Name: | Mr Munish Kumar |
| Submission Date: | 12 Oct 2023 |

Declaration Page

We, members of GBA group 2, do hereby declare that we each contributed to this assignment and that we collectively agree to a shared grade.

|  |  |  |
| --- | --- | --- |
| Name | Contribution | Signature |
| Tan Pei Ling | I did question 1d |  |
| Hilmi Bin Ishak | I did question 1d |  |
| Ong Ming Da | I did question 1a, 1b |  |
| Pang Shi Da Lawrence | I did question 1c |  |

**Question 1a**

#in order to start reading the data, the “pandas” package is imported

Import pandas as pd

#the next step is to load the .csv dataset

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

#viewing the top five rows of the dataset to check if the data has been imported correctly

hdb\_sales.head()

#to get the number of dimensions in the data which will result in 2, meaning the data is in a 2-dimensional array

hdb\_sales.ndim

#to get the number of rows and columns which results in (1250, 11), meaning the data has 1250 rows and 11 columns

hdb\_sales.shape

**Question 1b**

#loading the .csv dataset will convert missing entries in the dataframe to Not a Number, or NaN, by default

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

hdb\_sales

#identifying the number of missing values in the dataframe for each object

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

According to Siying (2023), missing data are not desirable for data analytics since they cannot be included in constructing models, forecasting, etc. It is thus necessary to deal with the missing values as they cannot be read by the statistical functions in pandas. Although the code can be run just fine, the end results may be inaccurate due to the computation of the missing values. As a data analyst, it is important to deal with missing values within the dataframe before they are used.

**Question 1c**

Mean/Median/Mode Imputation

#fill missing values in a numeric column with the mean

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

This works well under the assumption that the missing data are normally distributed or nearly normally. The data's overall trend remains the same.

Predictive Modeling (using scikit-learn)

from sklearn.linear\_model import LinearRegression

#create a model to predict missing values based on other factors

model = LinearRegression()

X = df.dropna().drop('resale\_price', axis=1)

y = df.dropna()['resale\_price']

model.fit(X, y)

#predict the missing values

missing\_data = df[df['resale\_price'].isnull()]

predicted\_values = model.predict(missing\_data.drop('resale\_price', axis=1))

df.loc[df['resale\_price'].isnull(), 'resale\_price'] = predicted\_values

When sufficient data and features are available, predictive modeling can effectively make accurate predictions about unknown variables. It makes use of interrelationships between variables to make inferences about missing information.

*Used* ***CHATGPT***

***Prompted*** *‘Propose ways to treat the missing data with Python and explain rationales of the treatments’*

***Full output:***

***“****To handle missing data in Python, consider the following methods:*

*Mean/Median/Mode Imputation:*

*Method: Replace missing values with the mean (for numeric data), median (for skewed data), or mode (for categorical data) of the respective column.*

*Rationale: This method is suitable when missing data is missing at random and doesn't introduce bias. It maintains the central tendency of the data.*

*Forward Fill/Backward Fill (for time-series data):*

*Method: Fill missing values with the previous (forward fill) or subsequent (backward fill) known value in a time-ordered dataset.*

*Rationale: Time-series data often exhibits temporal dependencies, making this method appropriate for preserving temporal patterns.*

*Interpolation (for time-ordered data):*

*Method: Use interpolation techniques (e.g., linear, polynomial) to estimate missing values based on neighboring data points.*

*Rationale: Interpolation leverages relationships between adjacent data points, making it suitable for data with continuous trends.*

*Predictive Modeling:*

*Method: Train a machine learning model (e.g., regression) on the non-missing data to predict missing values.*

*Rationale: This approach is useful when missing data has a complex relationship with other variables, allowing for accurate estimation.*

*Deletion:*

*Method: Remove rows or columns with missing values.*

*Rationale: Use deletion sparingly when missing data is minimal or removing it won't significantly impact the analysis. However, be cautious to avoid data loss.*

*Choose the method that aligns with your data's characteristics and the research objectives. The goal is to handle missing data in a way that maintains data integrity and minimizes bias in subsequent analyses.****”***

***Part of output used and adapted:***

*Mean/Median/Mode Imputation:*

*Rationale: This method is suitable when missing data is missing at random and doesn't introduce bias. It maintains the central tendency of the data.*

*Predictive Modeling:*

*Rationale: This approach is useful when missing data has a complex relationship with other variables, allowing for accurate estimation.*

**Question 1d**

Bar Chart

#importing of matplotlib

import matplotlib.pyplot as plt

#creating of bar plot with table output

fig, ax = plt.subplots()

hdbbar = cleanedhdb['flat\_type'].value\_counts().plot(ax=ax, kind='bar', table = True)

hdbbar.axes.get\_xaxis().set\_visible(False)

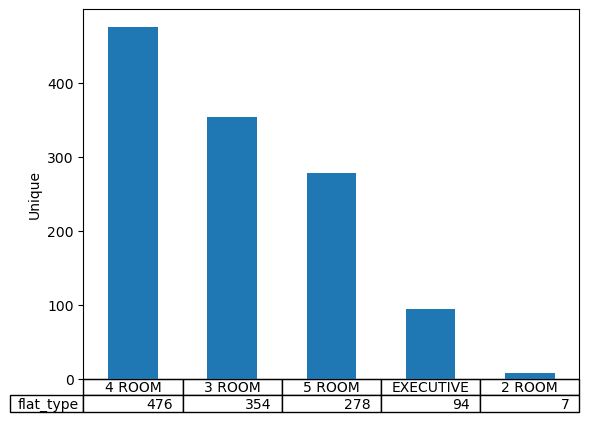
#styling of bar plot

plt.xlabel("Flat Type")

plt.ylabel("Unique")

**Figure 1**

*Distribution of purchase of HDB flat types*



In the bar chart, the most common HDB flat type is the 4-room flat, representing a high 40%. It reflects the ideal choice for young couples and small families. The next most common type is the 3-room flats that have leases starting in the 70s and 80s and were once the popular flat type for small families then. The common 4-room flats can be seen as an improvement in the quality of life for most Singaporeans as average incomes increase, leading to a better lifestyle.

The executive units are mainly bought by wealthier households or larger families. The low number may indicate more homeowners choosing private properties over HDBs as they view private housing to be better investments.

Line Plot

#grouping of data according to columns needed as well as the average of the groupings

unique = cleanedhdb[['lease\_commence\_date', 'resale\_price']]

unique2 = unique.groupby(['lease\_commence\_date'])

unique3 = unique2.mean()

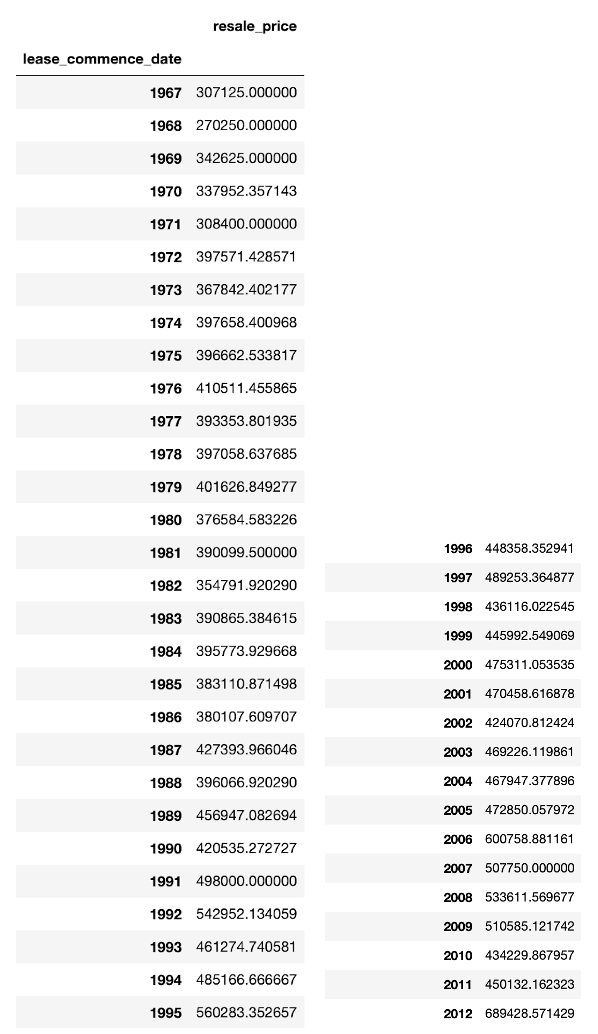
#line plot and table output

unique3.plot.line()

Unique3

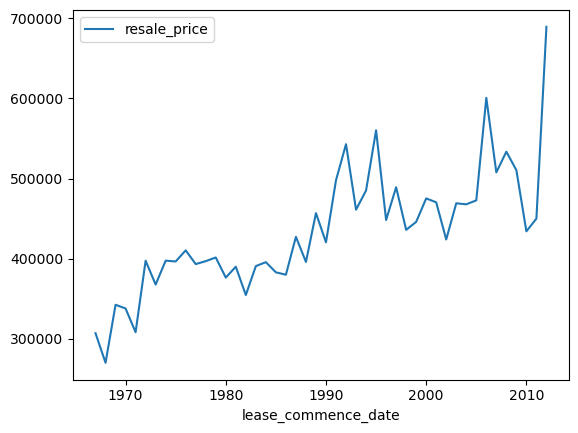
**Figure 2**

*Grouping of resale data according to lease commencement year and the average prices of housing in that year*



**Figure 3**

*Average resale HDB prices against the lease commencement year*



The line plot shows the average resale prices of HDB flats in relation to their lease commencement date over the decades from 1970 to 2010. The older flats built in the 70s and 80s have their average resale prices ranging from $300,000 to $400,000. For flats built from 1990 onwards, the prices saw a significant increase, climbing to more than $500,000 in value, a measure of the appreciating value of HDB houses.

A sharp uptick in the resale prices occurred for those flats built around 2010. It coincided with the remarkable growth in Singapore’s economy with its GDP hitting a record of 14.7% in that year (“Singapore Economy Sees Record Expansion in 2010,” 2011). A better performing economy translates to higher prices for resale flats when demand is greater than the supply, in our land-scarce Singapore.

Scatter Plot

#extracting columns for scatter plot and removal of duplicate values

scatterhdb = cleanedhdb[['floor\_area\_sqm', 'remaining\_lease', 'flat\_type']]

filterscatter = scatterhdb.loc[(scatterhdb.flat\_type == 'EXECUTIVE')]

rdscatterhdb = filterscatter.drop\_duplicates(subset='floor\_area\_sqm')

#scatter plot output

x = rdscatterhdb['floor\_area\_sqm']

y = rdscatterhdb['remaining\_lease']

plt.scatter(x, y)

plt.title('Floor Area per sqm vs Remaining Lease')

plt.xlabel('Floor Area per sqm')

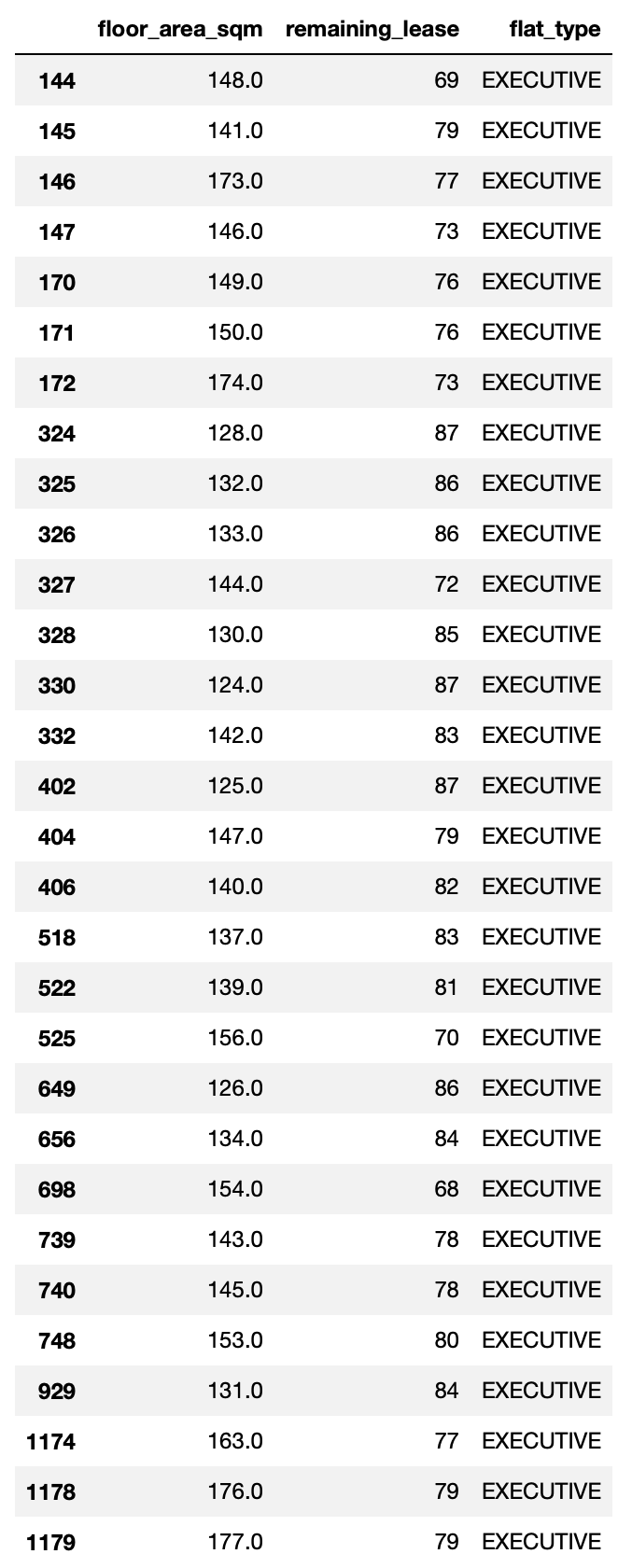
plt.ylabel('Remaining Lease')

#table output

rdscatterhdb

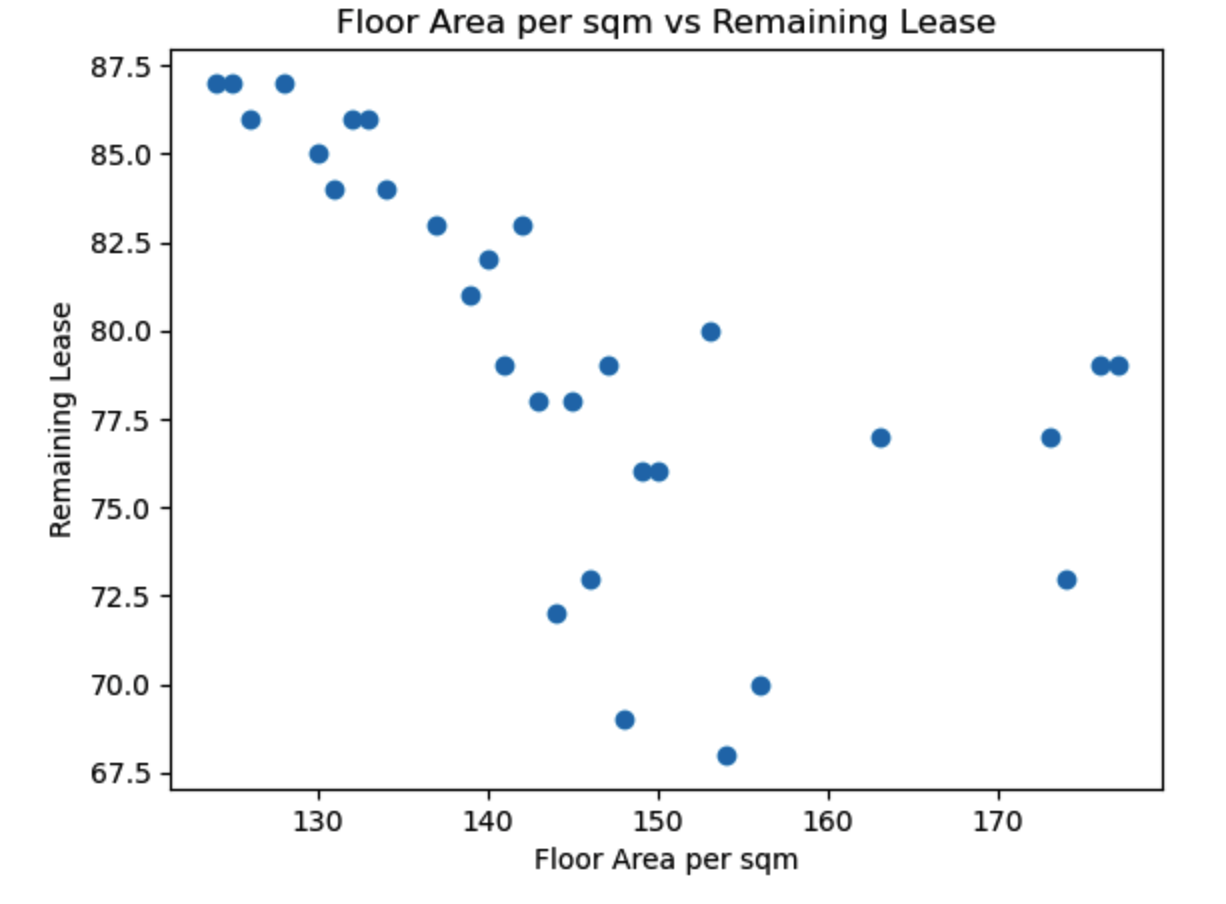
**Figure 4**

*HDB resale Executive flats data with duplicates removed*



**Figure 5**

*Floor Area per sqm against remaining lease of HDB Executive Resale flats*



The scatter plot shows a negative correlation of the floor area of executive flats to the remaining lease. The older the flats, which have shorter remaining leases, the larger the floor area of 150 sqm to slightly more than 170 sqm. The higher remaining lease corresponds to lower floor area, indicating that the newer flats are smaller, with floor area of below 150 sqm. One observable trend could be the shrinking household size over the years, from 4.87 persons in 1980 to 3.30 persons in 2017 (Commentary: The Future of Singapore Housing Is in Bigger HDB Flats, n.d.).

**References**

Kumar, A (2023, March 26). *Python – Replace Missing Values with Mean, Median & Mode*. <https://vitalflux.com/pandas-impute-missing-values-mean-median-mode/>

Pedregosa et al. (2011). Scikit-learn: Machine Learning in Python*. JMLR, 12, 2825-2830.* <https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html>

Sing, T. F. (2018, December 29). *Commentary: The future of Singapore housing is in bigger HDB flats*. CNA. https://www.channelnewsasia.com/commentary/jumbo-hdb-flats-dual-key-condominium-units-multi-generation-915351

Singapore economy sees record expansion in 2010. (2011, January 3). *BBC News*. <https://www.bbc.com/news/business-12106645>

Siying, D.K.W.K.Y.D. Z. (2023). *Python for Data Analytics*. Singapore University of Social Sciences (SUSS) - Publisher. <https://suss.vitalsource.com/books/9789814873918>

‌

‌