





CA2 PDS PRESENTATION

Presented by Soh Jian Min



AGENDA



- 
- 
- ① INTRODUCTION
 - ② DATA SOURCES & DATASETS
 - ③ TRANSPOSE AND TEXT ANALYSIS
 - ④ CLEANING, FILTERING & TRANSFORMING DATASETS
 - ⑤ QUESTION, GRAPH & ANALYSIS
 - ⑥ CONCLUSION & RECOMMENDATION



INTRODUCTION





Title of Analysis:

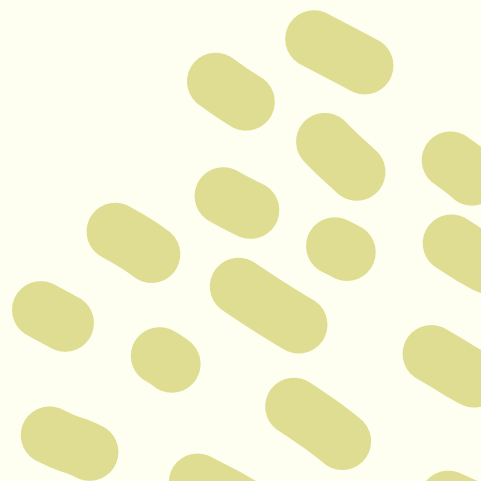
Market Trends and Insights for Housing Agents in Singapore:
Analyzing Popular Flat Types, Sales, and Economic Influences

Story Line:

As a data analyst, I aim to assist a housing agent company named "houseNow" in Singapore in understanding market trends and identifying the most popular types of flats or sales. This is crucial for advising clients and making informed decisions.



DATA SOURCES & DATASETS



DATA SOURCES



Data Source

1. https://beta.data.gov.sg/datasets/d_7c69c943d5f0d89d6a9a773d2b51f337/view (Private Residential Property Transactions in the Whole of Singapore, Quarterly)
2. https://beta.data.gov.sg/datasets/d_02aa4bb51bc674f3a2d0b9bb6911d934/view (Number of Resale Applications Registered by Flat Type)
3. https://beta.data.gov.sg/datasets/d_14f63e595975691e7c24a27ae4c07c79/view (Housing And Development Board Resale Price Index (1Q2009 = 100), Quarterly)
4. https://beta.data.gov.sg/datasets/d_97f8a2e995022d311c6c68cfda6d034c/view (Private Residential Property Price Index (Base Quarter 2009-Q1 = 100), Quarterly)
5. https://beta.data.gov.sg/datasets/d_dce7d88c668273bb8c1291027e63325a/view (Per Capita GNI And Per Capita GDP At Current Prices, Annual)

LOAD DATA

```
def load_and_check_data(file_path, var_name):  
    # Load the dataset  
    df = pd.read_csv(file_path)  
    print("Successfully loaded dataset " + file_path)  
    # To check if the data needs to be transposed  
    if df.shape[0] < df.shape[1]:  
        print(var_name + ' data needs to be transposed')  
  
    return df
```

Function to load and
check data whether is
there a need to transpose

Use function

```
# Load datasets  
private_sales = load_and_check_data('PrivateResidentialPropertyTransactionsintheWholeofSingaporeQuarterly.csv', 'private_sales')  
hdb_resale = load_and_check_data('NumberofResaleApplicationsRegisteredbyFlatType.csv', 'hdb_resale')  
hdb_resale_price_index = load_and_check_data('HousingAndDevelopmentBoardResalePriceIndex1Q2009100Quarterly.csv', 'hdb_resale_price_index')  
private_residential_price_index = load_and_check_data('PrivateResidentialPropertyPriceIndexBaseQuarter2009Q1100Quarterly.csv', 'private_residential_price_index')  
gdp_data = load_and_check_data('PerCapitaGNIAndPerCapitaGDPAAtCurrentPricesAnnual.csv', 'gdp_data')
```

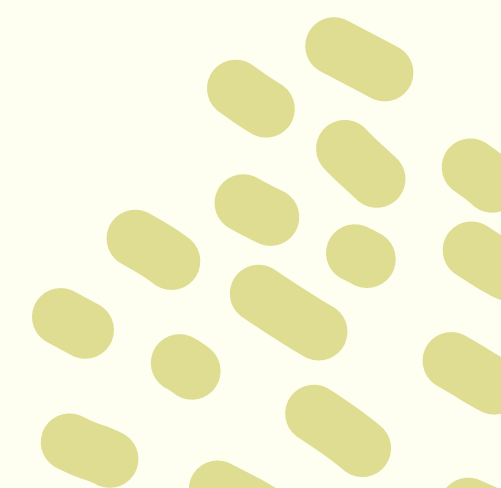


OUTPUT OF LOAD

```
Successfully loaded dataset PrivateResidentialPropertyTransactionsintheWholeofSingaporeQuarterly.csv  
Successfully loaded dataset NumberofResaleApplicationsRegisteredbyFlatType.csv  
Successfully loaded dataset HousingAndDevelopmentBoardResalePriceIndex1Q2009100Quarterly.csv  
Successfully loaded dataset PrivateResidentialPropertyPriceIndexBaseQuarter2009Q1100Quarterly.csv  
Successfully loaded dataset PerCapitaGNIAndPerCapitaGDPAAtCurrentPricesAnnual.csv  
gdp_data data needs to be transposed
```






TRANSPOSE & TEXT ANALYSIS





TRANSPOSE DATA

```
# Transpose gdp data (horizontal to vertical)
gdp_data = gdp_data.transpose()
gdp_data.columns = gdp_data.iloc[0] # Set the first row as the header
gdp_data = gdp_data.drop(gdp_data.index[0]) # Drop the first row
gdp_data = gdp_data.reset_index()
gdp_data.columns.name = None # Remove the column name
gdp_data.rename(columns={'index': 'year'}, inplace=True)
```



TEXT ANALYSIS

```
# Do a Text-Based Analysis on the datasets
def text_based_analysis(df, var_name, col_for_descriptive_stats=[]):
    # Make sure that 'na' is replaced with pd.NA (correct format for missing values)
    df.replace('na', pd.NA, inplace=True)
    print("Text-Based Analysis for " + var_name)
    print()

    # Shape of the dataset
    print('This is the shape of the dataset:')
    print(df.shape)
    print()

    # Index of the dataset
    print('This is the index of the dataset:')
    print(df.index)
    print()

    # Columns of the dataset
    print('These are the columns of the dataset:')
    print(df.columns)
    print()

    # Non-NA values of each column in the dataset
    print('The total number of non-NA values in the dataset is:')
    print(df.count())
    print()

    # NA values of each column in the dataset
    print("The total number of NA values in the dataset is:")
    print(df.isnull().sum())
    print("\n")

    # Summary of the dataset
    print("A summary of this dataset is shown below:")
    print(df.info())
```

```
# Descriptive statistics summary of the dataset based on the col_for_descriptive_stats specified
print("A descriptive statistics summary of this dataset is shown below:")
for col in col_for_descriptive_stats:
    if col in df.columns:
        print(f"A descriptive statistical summary of this dataset is shown below for: {col}")
        if df[col].dtype in ['float64', 'int64']:
            print(df[col].describe())
        else:
            print(df[col].astype('float64').describe())
        print()
    else:
        print(f"Column '{col}' not found in the dataset.")
        print()
        print()

print("=====")
print()
```

Function for text analysis

Using the function

```
text_based_analysis(private_sales, 'private_sales', ["units"])
text_based_analysis(hdb_resale, 'hdb_resale', ["no_of_resale_applications"])
text_based_analysis(hdb_resale_price_index, 'hdb_resale_price_index', ["index"])
text_based_analysis(private_residential_price_index, 'private_residential_price_index', ["index"])
text_based_analysis(gdp_data, 'gdp_data', ["Per Capita GNI", "Per Capita GDP", "Per Capita GNI (US Dollar)", "Per Capita GDP (US Dollar)"])
```

TEXT ANALYSIS

Text-Based Analysis for private_sales

This is the shape of the dataset:
(392, 4)

This is the index of the dataset:
RangeIndex(start=0, stop=392, step=1)

These are the columns of the dataset:
Index(['quarter', 'type_of_sale', 'sale_status', 'units'], dtype='object')

The total number of non-NA values in the dataset is:

```
quarter      392
type_of_sale  392
sale_status   196
units         392
dtype: int64
```

The total number of NA values in the dataset is:

```
quarter      0
type_of_sale  0
sale_status   196
units         0
dtype: int64
```

A summary of this dataset is shown below:

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 392 entries, 0 to 391

Data columns (total 4 columns):

#	Column	Non-Null Count	Dtype
0	quarter	392 non-null	object
1	type_of_sale	392 non-null	object
2	sale_status	196 non-null	object
3	units	392 non-null	int64

dtypes: int64(1), object(3)

memory usage: 12.4+ KB

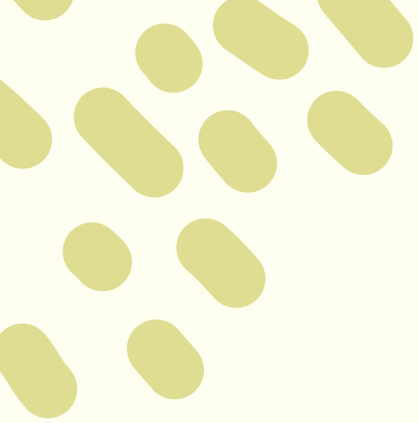
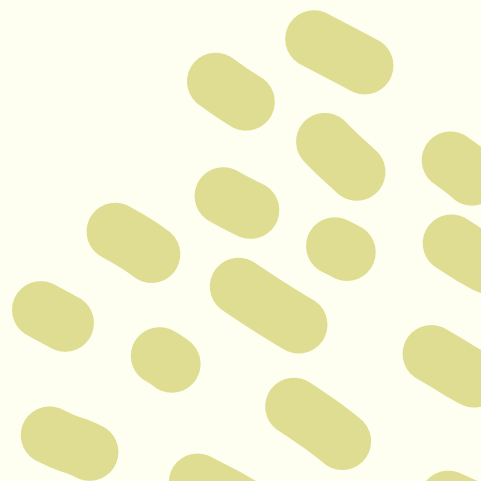
None

A descriptive statistical summary of this dataset is shown below for the column: units

```
count      392.000000
mean       1374.306122
std        1513.903455
min         12.000000
25%        130.000000
50%        824.000000
75%        2137.000000
max        7776.000000
Name: units, dtype: float64
```

=====

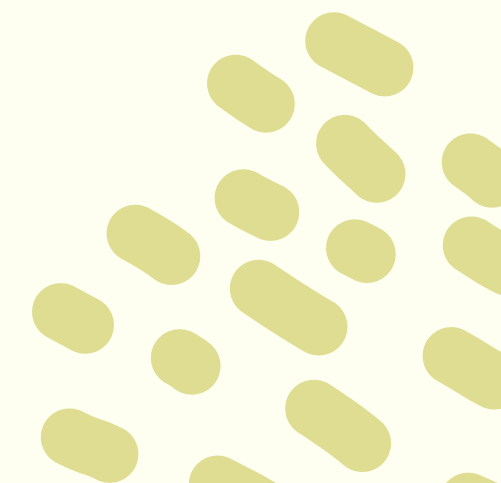
CLEANING, FILTERING & TRANSFORMING DATASETS



CLEAN DATA



```
# Replace NaN values with 'Total Applications' in the 'sale_status' column  
private_sales['sale_status'] = private_sales['sale_status'].fillna('Applied')
```



TRANSFORM & FILTER DATA

```
# Transforming the data
# private_sales
# Pivot the DataFrame
pivot_df = private_sales.pivot_table(index='quarter', columns=['type_of_sale', 'sale_status'], values='units', aggfunc='sum')
# Flatten the MultiIndex columns
pivot_df.columns = ['private_' + '_'.join(col).lower().strip() if isinstance(col, tuple) else col.lower() for col in pivot_df.columns]
# Reset the index to turn the 'quarter' index back into a column
pivot_df = pivot_df.reset_index()
# Expected columns
expected_columns = ['quarter', 'private_new sale_completed', 'private_new sale_uncompleted', 'private_resale_applied', 'private_sub sale_applied']
# Reindex to ensure all expected columns are present, filling missing columns with 0
pivot_df = pivot_df.reindex(columns=expected_columns, fill_value=0)
# Assign the pivoted DataFrame back to the original DataFrame
private_sales=pivot_df
```



TRANSFORM & FILTER DATA

```
# hdb_resale
# Pivot the DataFrame
pivot_df = hdb_resale.pivot_table(index='quarter', columns='flat_type', values='no_of_resale_applications', aggfunc='sum')
# Flatten the MultiIndex columns and convert to lowercase
pivot_df.columns = ['hdb_' + col.replace("-", "_").lower() for col in pivot_df.columns]
# Reset the index to turn the 'quarter' index back into a column
pivot_df = pivot_df.reset_index()
# Expected columns
expected_columns = ['quarter', 'hdb_1_room', 'hdb_2_room', 'hdb_3_room', 'hdb_4_room', 'hdb_5_room', 'hdb_executive']
# Reindex to ensure all expected columns are present, filling missing columns with 0
pivot_df = pivot_df.reindex(columns=expected_columns, fill_value=0)
# Rename pivot_df back to hdb_resale
hdb_resale = pivot_df
```




TRANSFORM & FILTER DATA

```
# private_residential_price_index  
# Group by 'quarter' and calculate the mean of 'index'  
private_residential_price_index= private_residential_price_index.groupby('quarter')['index'].mean().reset_index()  
# Round the index values to 1 decimal place  
private_residential_price_index['index'] = private_residential_price_index['index'].round(1)
```



TRANSFORM & FILTER DATA

```
# Filtering and Joining the Datasets  
# Merge datasets on quarter (inner join to keep only rows with matching quarters)  
merged_data = private_sales.merge(hdb_resale, on='quarter', how='inner')  
merged_data = merged_data.merge(hdb_resale_price_index, on='quarter', how='inner')  
merged_data = merged_data.merge(private_residential_price_index, on='quarter', how='inner', suffixes=('_hdb', '_private'))
```

```
# Extract year from quarter for GDP data merging  
merged_data['year'] = merged_data['quarter'].str[:4].astype(int)  
# Ensure the year column is of the same type in both DataFrames  
gdp_data['year'] = gdp_data['year'].astype(str)  
merged_data['year'] = merged_data['year'].astype(str)  
# Merge GDP data based on year (left join to keep all rows in merged_data)  
merged_data = merged_data.merge(gdp_data, on='year', how='left')  
# Drop the 'year' column as it's no longer needed  
merged_data = merged_data.drop(columns=['year'])
```

TRANSFORM & FILTER DATA

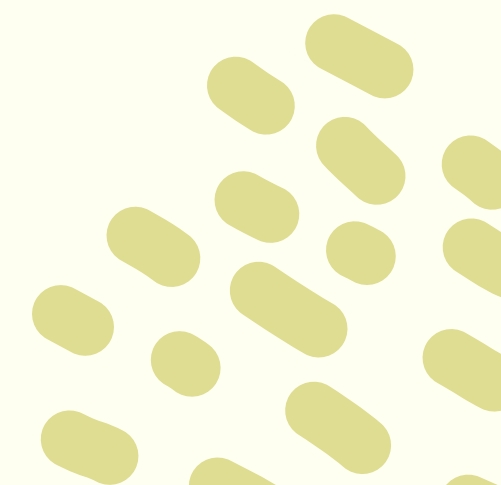
```
# Convert columns to appropriate numeric types
columns_to_numeric = [
    'private_new sale_completed', 'private_new sale_uncompleted', 'private_resale_applied', 'private_sub sale_applied',
    'hdb_1_room', 'hdb_2_room', 'hdb_3_room', 'hdb_4_room', 'hdb_5_room', 'hdb_executive',
    'index_hdb', 'index_private', 'Per Capita GNI', 'Per Capita GDP',
    'Per Capita GNI (US Dollar)', 'Per Capita GDP (US Dollar)'
]

for col in columns_to_numeric:
    merged_data[col] = pd.to_numeric(merged_data[col], errors='coerce')

# Drop rows with any NaN values
merged_data.dropna(inplace=True)
```



QUESTION, GRAPH & ANALYSTS

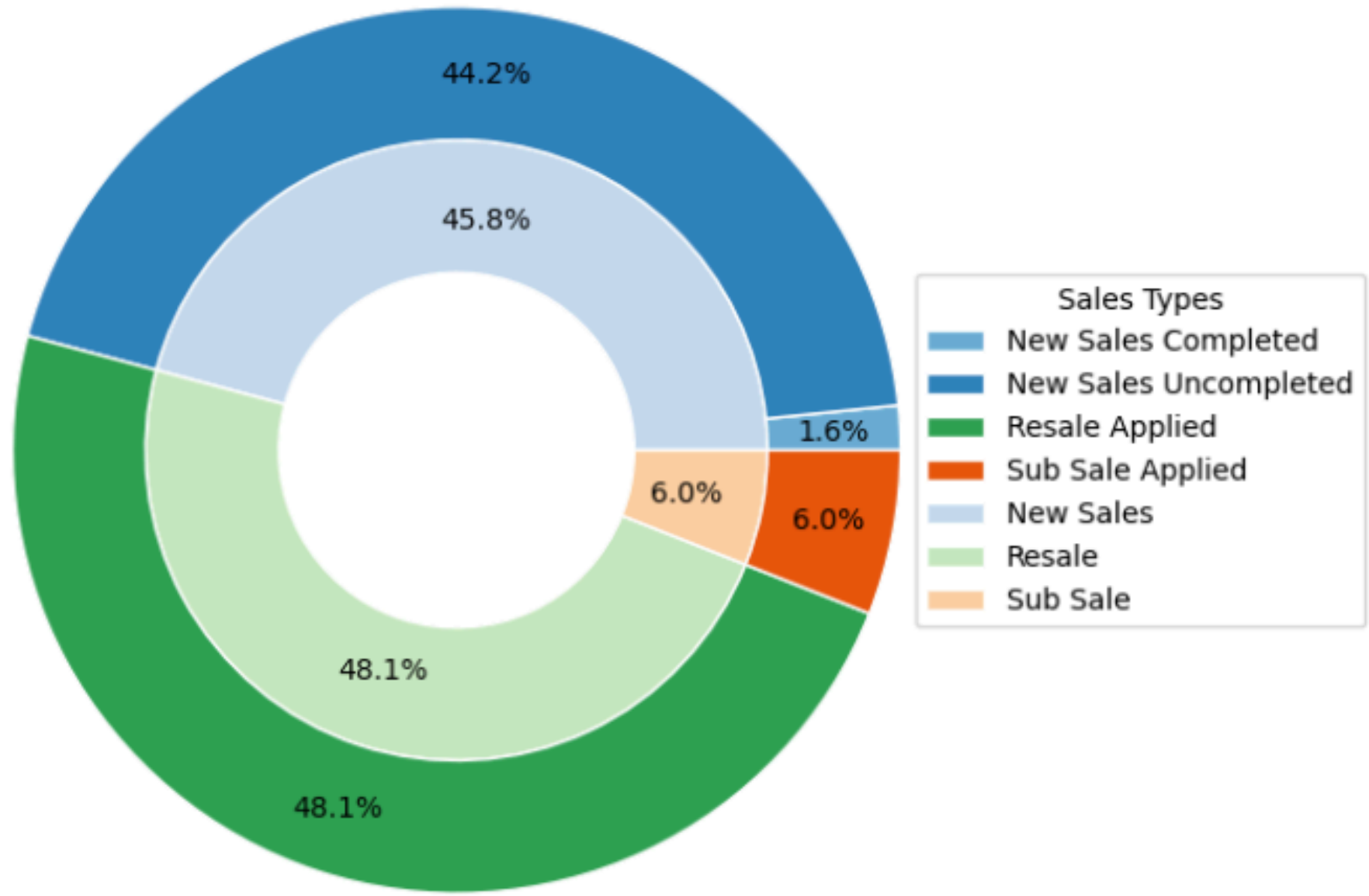




QUESTION 1

What types of sales are most popular in the private residential property market?

Types of Sales in the Private Residential Property Market



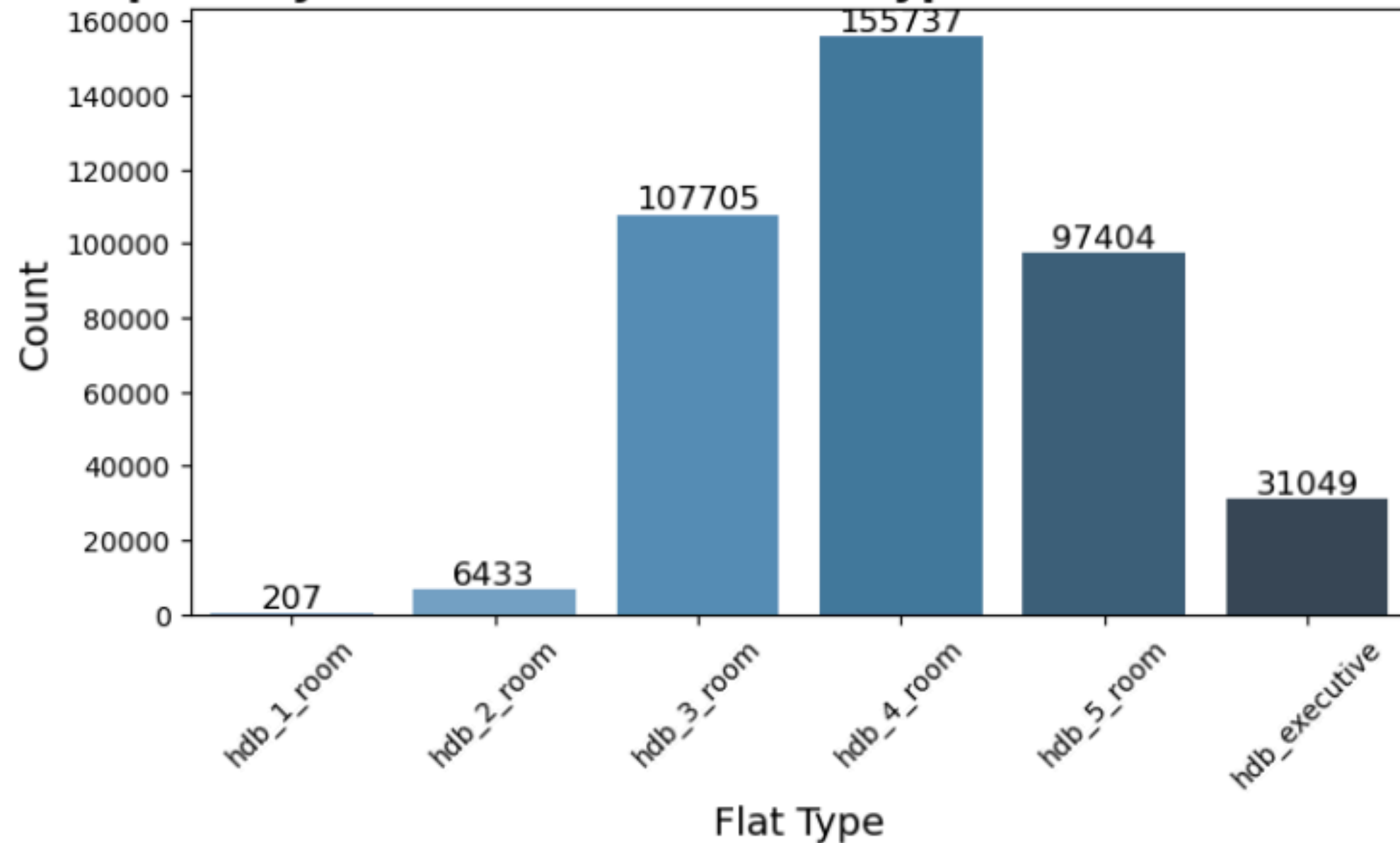
- Focus marketing on popular sales types
- Tailor recommendations to buyers' needs and budgets
- Optimize sales strategies to meet client demands



QUESTION 2

Which flat types are most popular in the HDB resale market?

Popularity of Different HDB Flat Types in the Resale Market



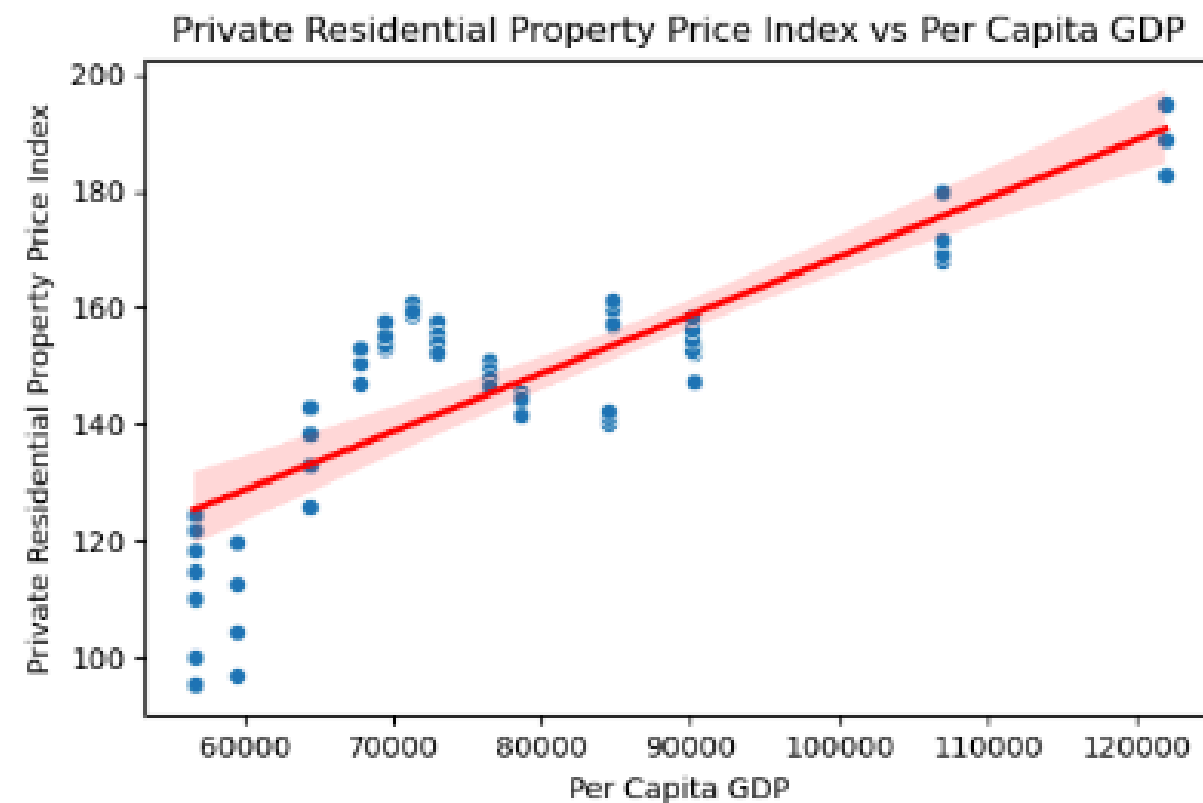
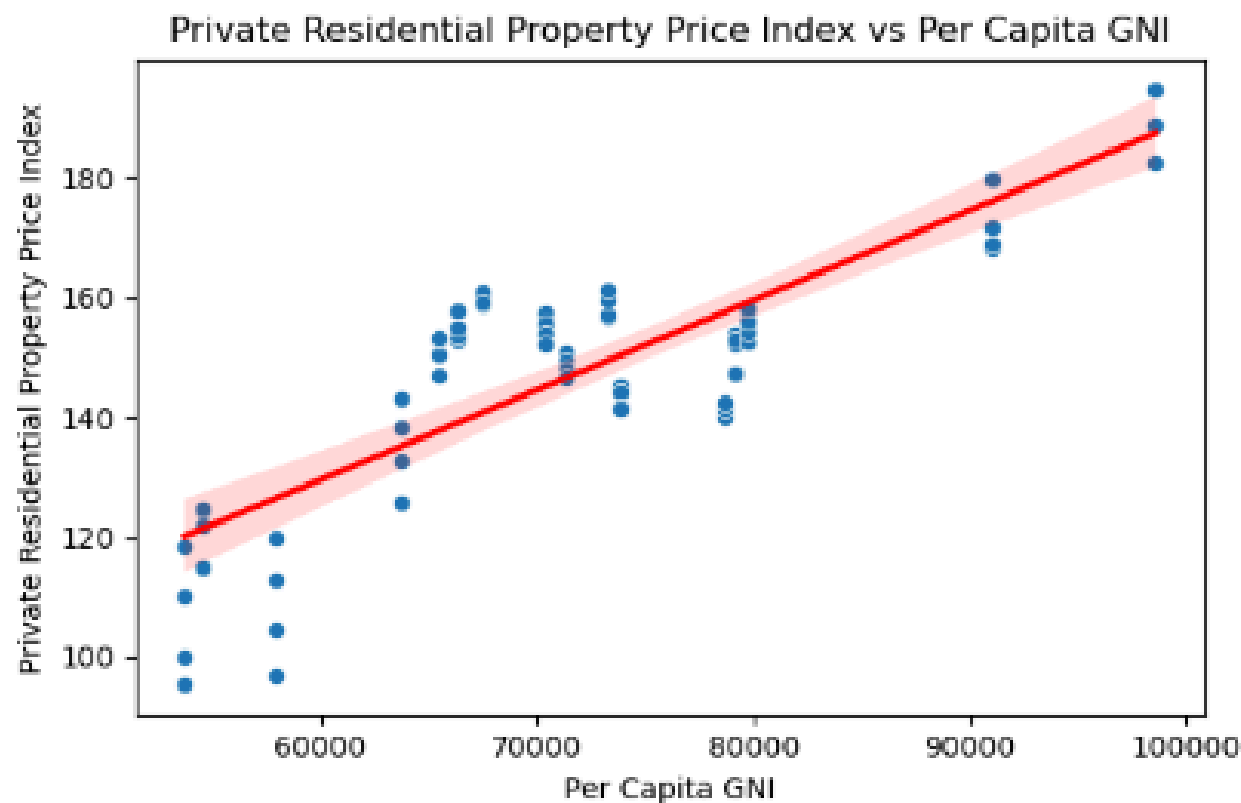
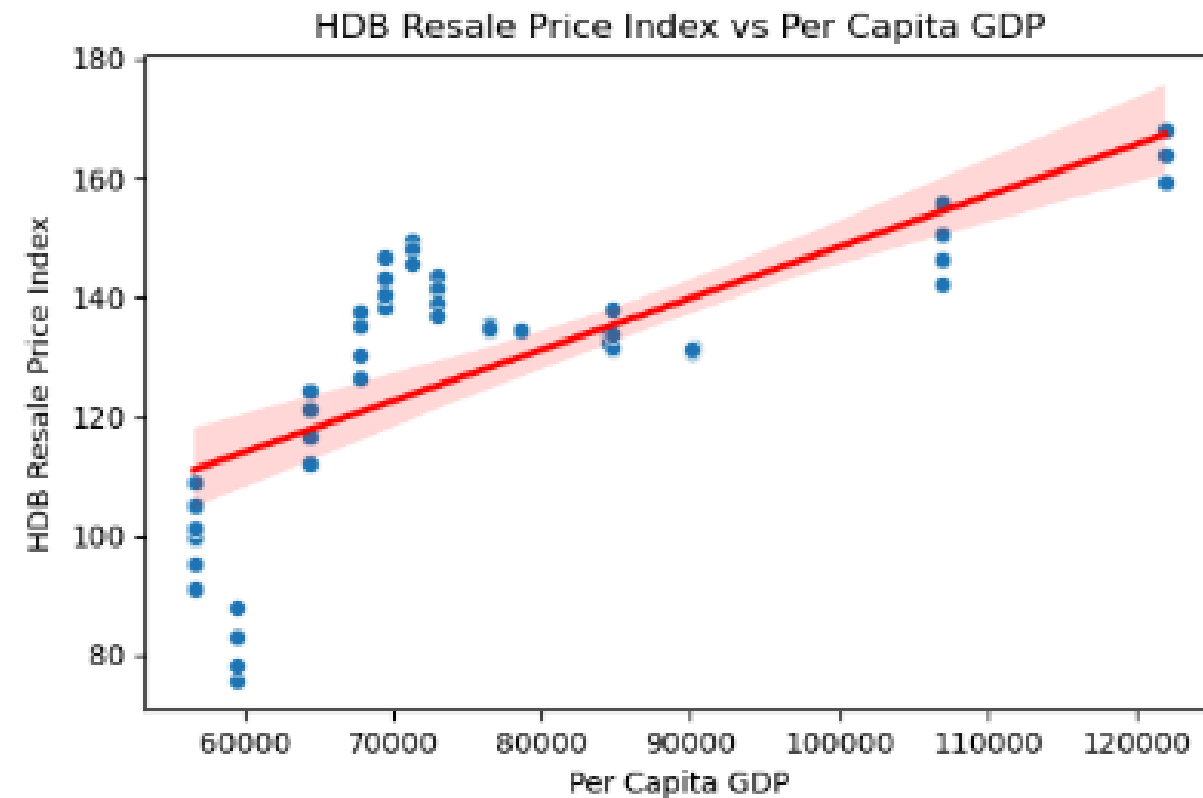
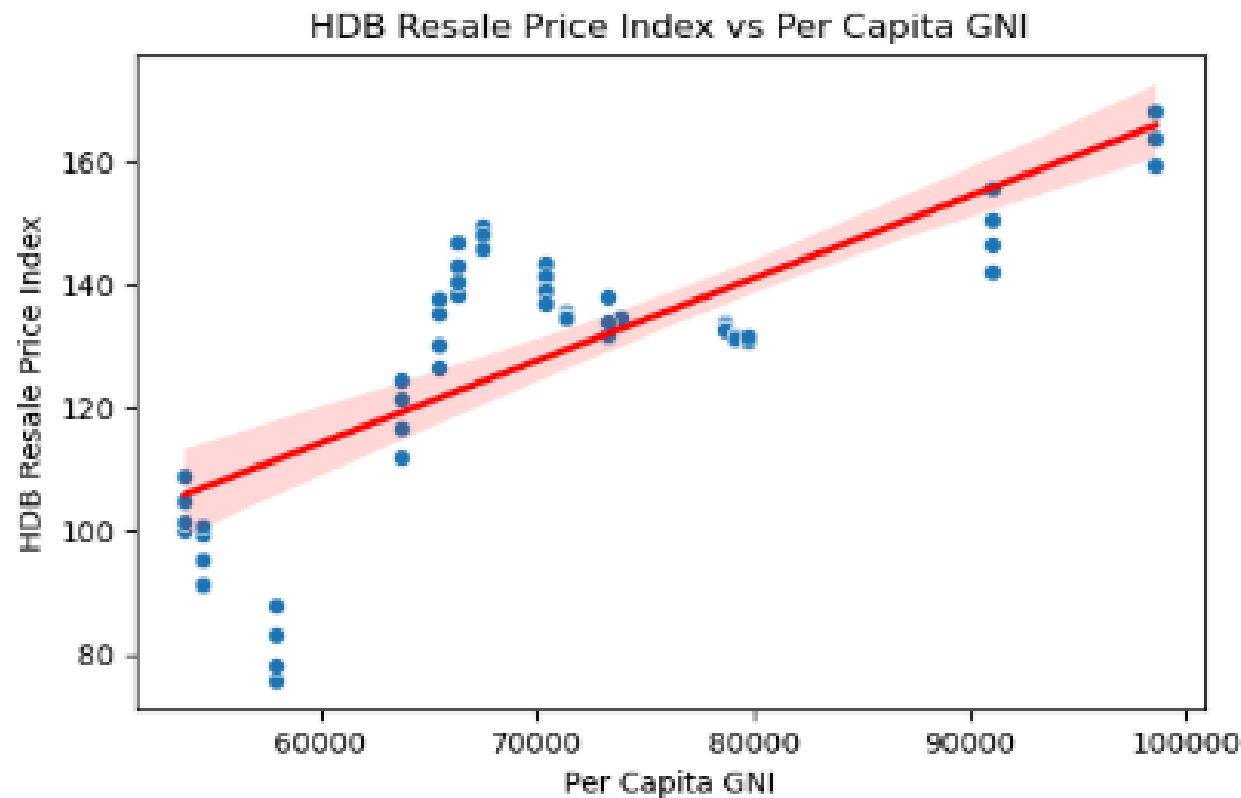
- 3-room, 4-room, and 5-room HDB flats are the most popular, housing agents should focus their efforts on these flat types
- Tailor recommendations based on buyer needs (family size), and budget
- Strategic Marketing (Highlight Popularity, Detailed Listings and, Focus on High-Demand Areas)



QUESTION 3

Does GDP and GNI affect the price index of
HDB Resale and Private Residential
Property?






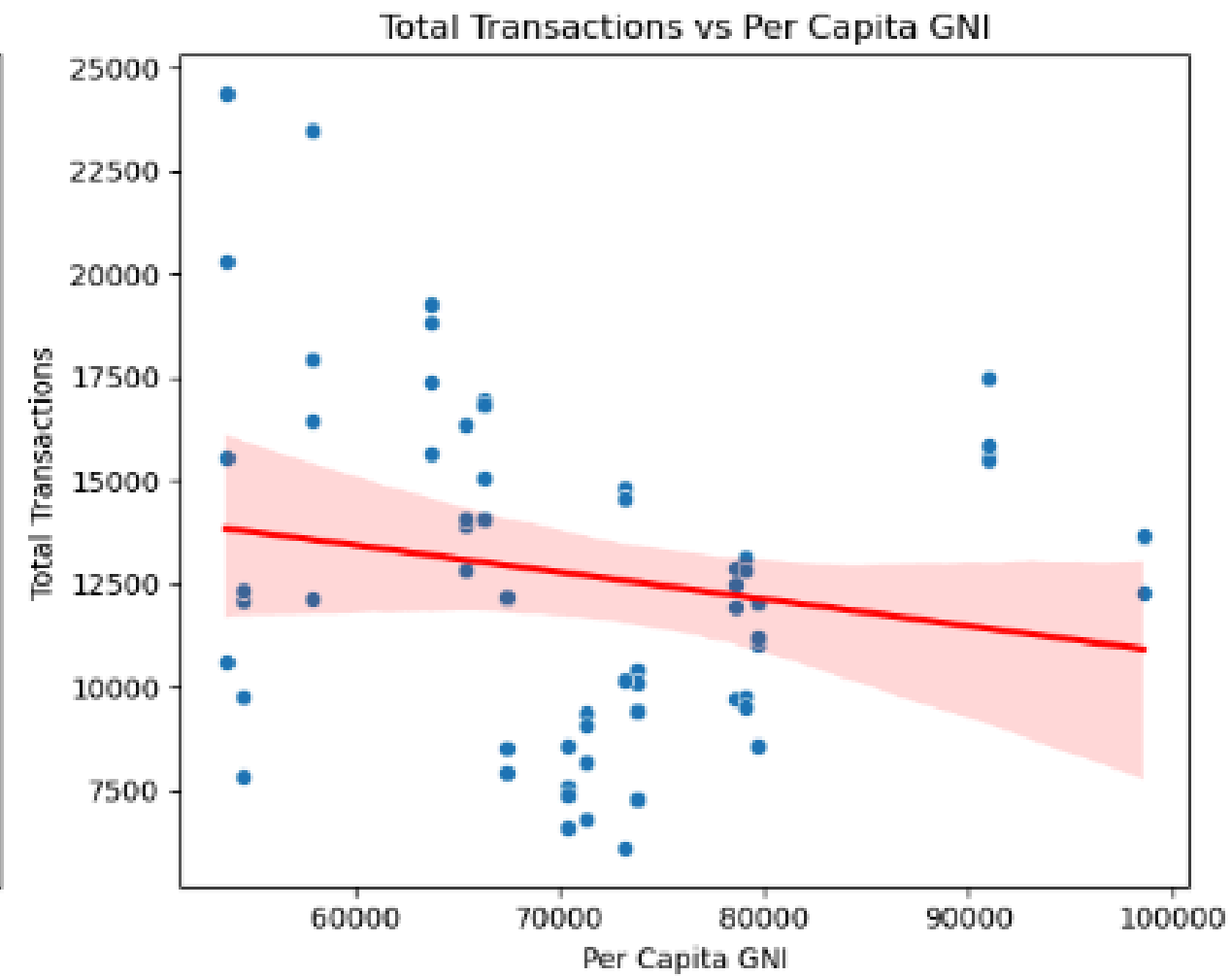
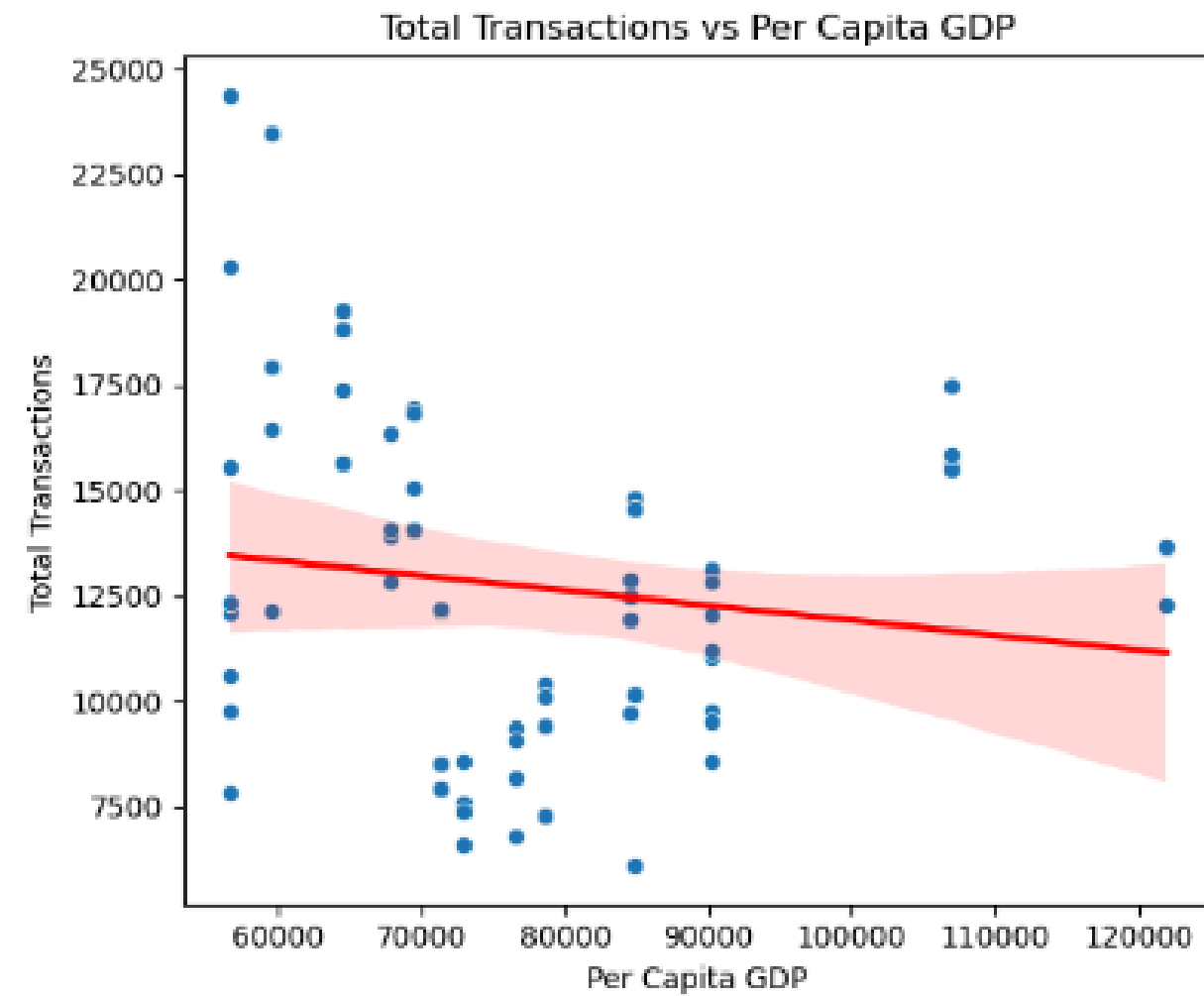
- Both HDB and private residential property price indices increase with higher GDP and GNI
- Advise Clients on Timing, encourage clients to make purchases when GDP and GNI are lower.
- Monitor Economic Trends, to identify the best buying periods and adjust marketing strategies accordingly
- Provide Strategic Insights: Use predictive analysis to offer clients timely advice.
- Consider Budget and Income Levels, For private properties, ensure that buyers are aware of the higher price indexes and recommend options that fit their budget



QUESTION 4

Does GDP and GNI affect the
Applications/Transactions of HDB Resale
and Private Residential Property?





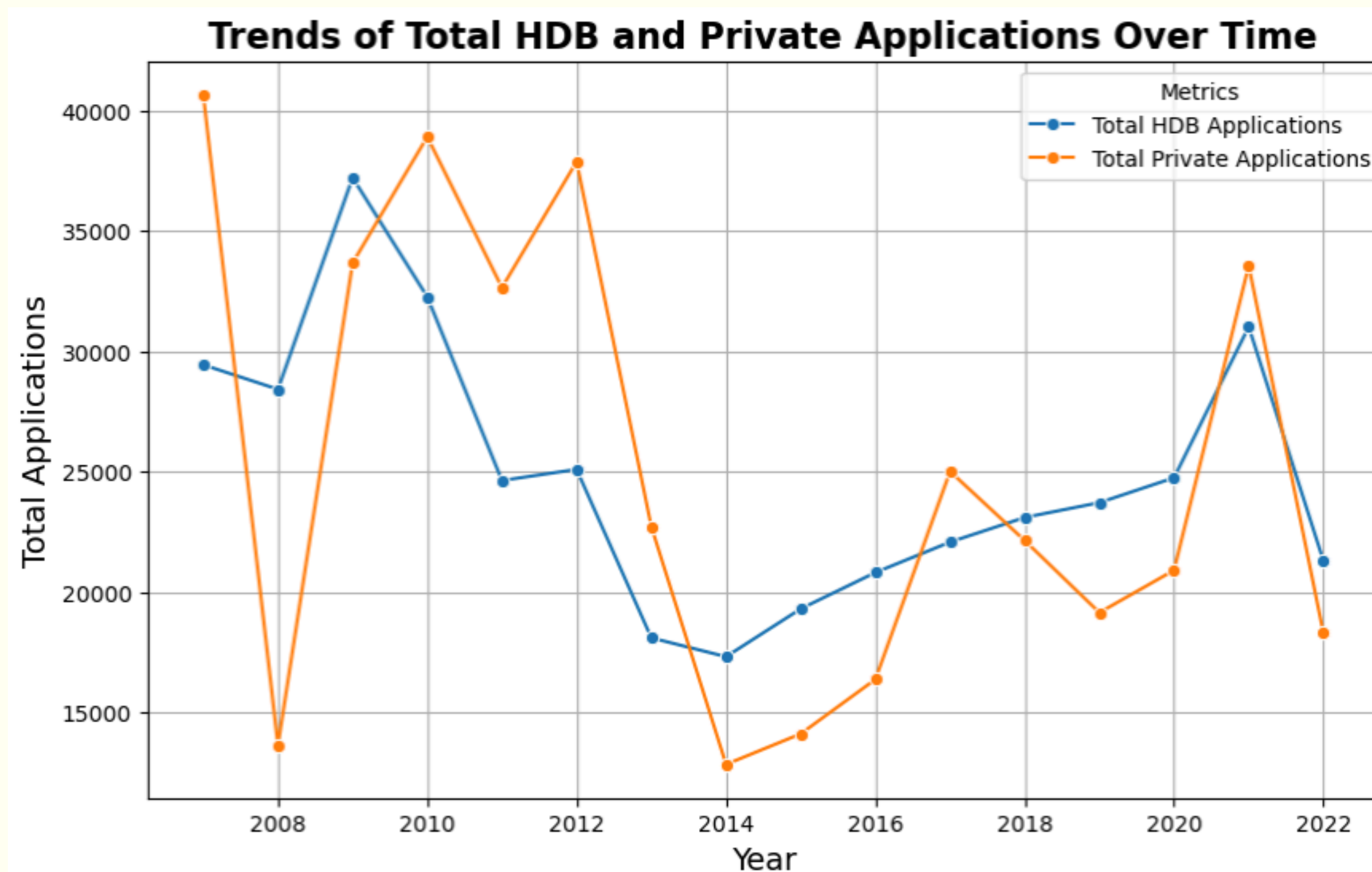
- Both Graph shows negative trend.
- Higher GDP and GNI are associated with fewer housing transactions
- Focus on marketing strategies during periods of lower GDP and GNI, as the number of transaction will be relatively higher.



QUESTION 5 ✨

Should housing agents focus more on HDB
Resale or Private Residential Property?





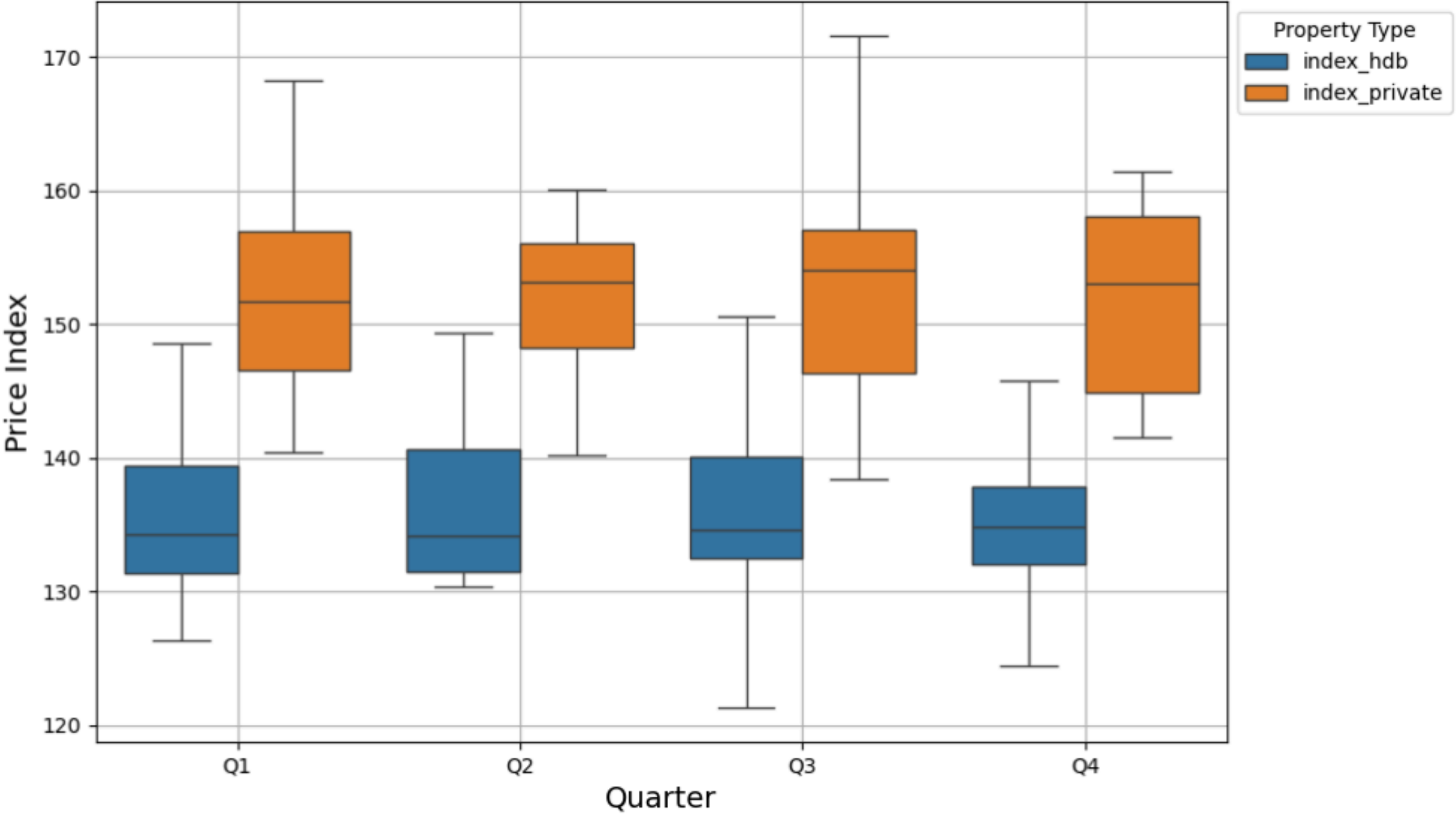
- **Focus on Both:** Both HDB Resale and Private Residential Property applications have shown fluctuations and recoveries. Housing agents should focus on both, adjusting their strategy based on market conditions and economic factors.
- **HDB Resale Applications:** HDB applications tend to be more stable and predictable. Agents preferring stability might focus more on HDB flats.
- **Private Residential Property Applications:** Private applications are more unpredictable but can hit higher peaks, offering the chance for higher returns during good market times.



QUESTION 6 ✨

How do the prices of private residential property and HDB resale properties compare across different quarters?

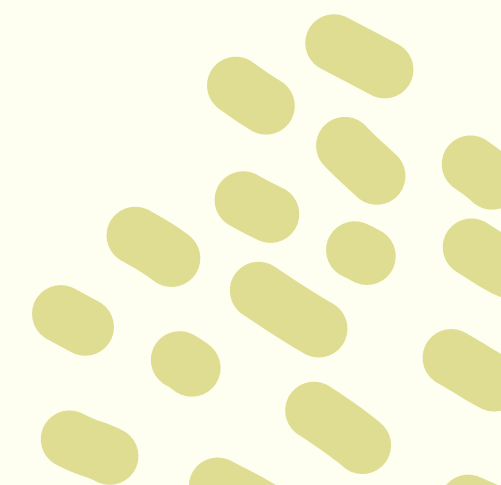
Comparison of Private and HDB Resale Property Price Index Across Different Quarters



- Private Property Price Index is generally higher than HDB Resale Property Price Index (High Income - Private Property, Middle Income - HDB Resale Property)
- During Economic Growth, can focus on private properties as premium housing have higher return while during Economic Uncertainty suggest HDB resale properties due to their affordability and stable price.
- Q2 and Q4 is more stable for Private Property
- Q4 is more stable for HDB Resale Property



CONCLUSION



Dominance of New and Resale Sales

- Resales sales type is the most popular in the private residential market
- Also shows a strong preference for new sales but large percentage being uncompleted

Price Stability

- HDB prices are more stable.
- Private property prices are higher and more unpredictable.

Price Variability

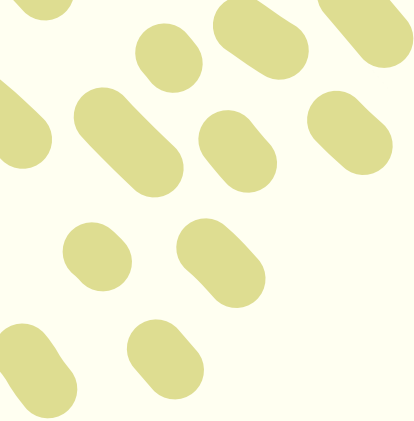
- Private property prices are higher and more variable.
- HDB resale prices are stable, especially in Q4.

Economic Impact

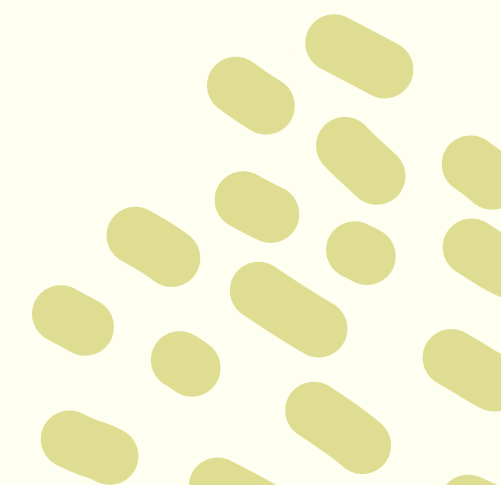
- GDP/GNI influence prices and transaction volumes.
- Higher economic indicators lead to higher prices, fewer transactions.

Market Preferences

- Higher preference for resale followed by new sales in private market.
- Strong preference for 4-room, 3-room, and 5-room flats in HDB Resale market.



RECOMMENDATIONS



High-Income Clients

- Focus on private properties.
- Emphasize Q2 and Q4 for stability.
- Resale for immediate occupancy.
- New Sales for investment.

Focus on Popular Flat Types/Sales Type

- Prioritize 4-room, 3-room, and 5-room HDB Resale flats.
- Prioritize Resale and New Sales in private residential property market.

Middle-Income Clients

- Emphasize HDB resale properties.
- Promote 4-room, 3-room, and 5-room flats.
- Highlight Q4 for stability.

Client Education and Advisory

- Educate clients on economic impact on prices.
- Use trends to guide purchase timing.
- Leverage GDP and GNI predictions for advice.

Timing and Market Conditions:

- Recommend private property purchases in Q2 and Q4.
- Suggest HDB purchases in Q4 for lower risk.
- Monitor GDP and GNI for best buying time for buyers and adjust marketing efforts.

Long-Term Investment Strategy

- Encourage HDB resale properties for steady growth.
- Suggest selective private properties during stable quarters for higher returns.

Continuous Market Monitoring

- Monitor market trends and GDP/GNI.
- Adjust marketing for based on popular flat and sales types.

THANK
YOU

