Data Preparation & Cleaning

- Removed redundant or sparse columns such as Area (SQM), Unit Price ($ PSM) and Nett Price ($).
- Standardized pricing using unit Price = Transacted Price/No. of Unit, instead of total Transacted Price as there may be multiple units transacted in 1 transaction.
- Extracted Lease Start Year, Lease Duration to calculate Remaining Lease Years. Assumed that all freehold properties have 999 yrs Lease Duration.

```
# Import necessary libraries
import numpy as np
import pandas as pd
import re
from datetime import datetime
import json
import seaborn as sb
import matplotlib.pyplot as plt

private_data = "../datasets/original/private_with_api.csv"

df = pd.read_csv(private_data, quotechar='"', escapechar='\\',
thousands=',')

# Cleaning private data
df = df.drop(['Area (SQM)','Unit Price ($ PSM)', 'Nett Price($)'],
axis = 1)
df['Transacted Price ($)'] = df['Transacted Price ($)'].astype(float)
df['Number of Units'] = df['Number of Units'].astype(float)
df["Price"] = ((df['Transacted Price ($)'])/(df['Number of
Units'])).astype(int)

# Calculate Remaining Lease Years
df["Sale Date"] = pd.to_datetime(df["Sale Date"], format="%b-%y")
df["Sale Year"] = df["Sale Date"].dt.year
df["Lease Duration"] = df["Tenure"].str.extract(r"(\
d{2,6})").astype(float)
df["Lease Start Year"] = df["Tenure"].str.extract(r"(\
d{4})").astype(float)

def calculate_remaining_lease(row):
    tenure_text = row["Tenure"].lower()
    lease_duration = row["Lease Duration"]
    start_year = row["Lease Start Year"]
    sale_year = row["Sale Year"]

    if "freehold" in tenure_text or (lease_duration and lease_duration
> 900):
        return 999
    if "leasehold" in tenure_text:
        return lease_duration

    if pd.notna(start_year):
        return max(lease_duration - (sale_year - start_year), 0)

    return None

df["Remaining Lease Years"] = df.apply(calculate_remaining_lease,
axis=1)

def lease_start_year(row):
    type_of_sale = row["Type of Sale"]
    sale_year = row['Sale Year']
    lease = row["Lease Start Year"]

    if "New Sale" in type_of_sale:
        return sale_year
    else:
        return lease

def lease_duration(row):
    tenure_text = row["Tenure"].lower()
    lease = row["Lease Duration"]

    if "freehold" in tenure_text:
        return 999
    else:
        return lease

df["Lease Start Year"] = df.apply(lease_start_year, axis =
1).astype('Int64')
df["Lease Duration"] = df.apply(lease_duration, axis =
1).astype('Int64')
df["Remaining Lease Years"] = df["Remaining Lease
Years"].astype('Int64')

df.head()
df.to_csv("cleaned_private.csv", index=False)
```