Housing Prices in Singapore

Annabel Lee [Dec 2020]

Introduction

Global Living: Top 10 highest value locations

Rank	City	Average property price ($US)	Average price per square foot ($US)
1	Hong Kong	1,235,220	2,091
2	Singapore	874,372	1,063
3	Shanghai	872,555	714
4	Vancouver	815,322	n/a
5	Shenzhen	680,283	726
6	Los Angeles	679,220	466
7	New York	674,500	526
8	London	646,973	776
9	Beijing	629,276	575
10	Paris	625,299	985

Source: CBRE

- In land-scarce Singapore, real estate properties prices are sky-high; in fact Singapore ranks second among the most expensive residential property markets worldwide, after Hong Kong.
- As such, housing prices may not be so accessible and house shopping is a big decision indeed. Hence we want to be wise when buying a property, be it for own stay or future investment.
- We want to investigate different drivers that may affect Singapore's HDB prices. Some possible drivers may be town locations of the HDB flats, proximity to public transport MRT train stations, distance from Central (Orchard). This analysis will hopefully provide some insights as well as consideration factors when buying or selling your properties.
- For the purpose of this project I have opted to do analysis and modelling on HDB flats (public housing) as the data is public and readily available, without having to spend a hefty sum on the private property sites for data but the techniques and analysis will follow in the similar fashion.

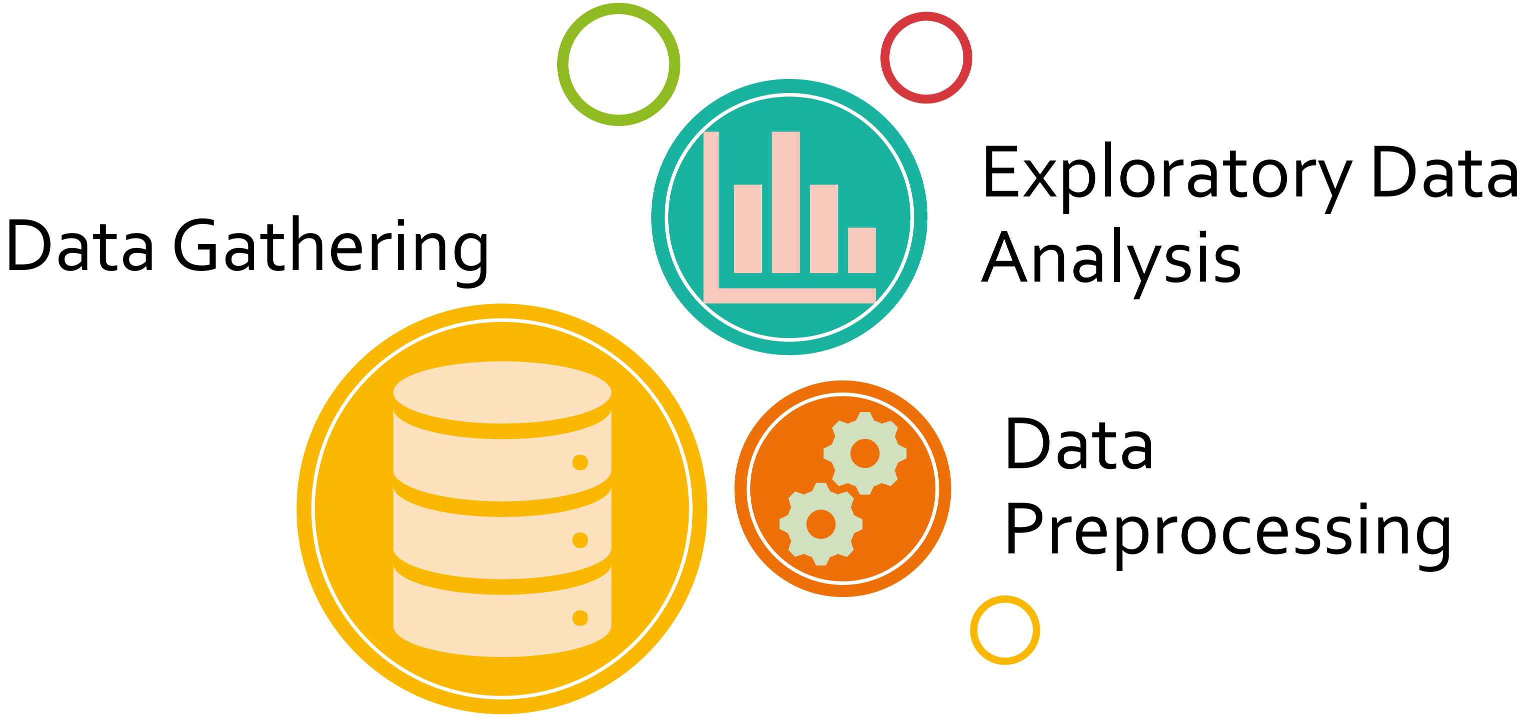
Data

- Resale HDB flat prices, as a proxy of property prices - this data set will include useful information including
  - Town
  - flat type
  - Size
  - street name
  - resale price
  - remaining lease
  - lease commencement date
- From these, we will also be able to derive price per area, address, number of years remaining in the lease etc, which after some exploratory data analysis, we can use to determine which will be features in our model.
- * Public transport - we will use a static list of public train service MRT stations for a simplified view.
- * List of retail malls
- * Geolocation coordinates for the above mentioned including HDB flats, MRT stations, retail malls

Sources

- Resale flat prices - https://data.gov.sg/
- Geolocation data - https://docs.onemap.sg/
- MRT Stations - https://en.wikipedia.org/wiki/List_of_Singapore_MRT_stations_by_planning_area
- Retail Shopping Malls - https://en.wikipedia.org/wiki/List_of_shopping_malls_in_Singapore

Methodology



Data Gathering

1. Get list of HDB data via API
2. Get list of MRT stations
3. Get list of Retail Shopping Malls
4. Get geolocation coordinates for the above 3 datasets
5. Calculate minimum distances from Orchard, from Nearest MRT, from Nearest Retail Mall, price per sqm
6. Combine to get a master dataset

Data Gathering

We first get a list of resale price data via api call.

[3]:
```
LIMIT = 5000 ## we dont want to overrload the api call

query_string='https://data.gov.sg/api/action/datastore_search?resource_id=42ff9cfe-abe5-4b54-beda-c88f9bb438ee&limit='+str(LIMIT)
resp = requests.get(query_string)
data = json.loads(resp.content)
len(data['result']['records'])
```

[3]: 5000

[8]:

	MRT	Latitude	Longitude
0	Admiralty MRT Station	1.44034337155075	103.800984160903
1	Aljunied MRT Station	1.31623848507354	103.882496650859
2	Ang Mo Kio MRT Station	1.36993284962264	103.849558091776
3	Bartley MRT Station	1.34244543829251	103.88019708711701
4	Bayfront MRT Station	1.28283490852293	103.85959687246401

[9]:

	Mall	RoadName	Latitude	Longitude
0	100 AM	TRAS STREET	1.27458821795427	103.84347073660999
1	112 Katong	EAST COAST ROAD	1.30508681845447	103.905098915055
2	313@Somerset	ORCHARD ROAD	1.30100656917243	103.838246592796
3	321 Clementi	CLEMENTI AVENUE 3	1.3120249182444	103.764960537008
4	888 Plaza	WOODLANDS DRIVE 50	1.4371305244487	103.795289911954

```
Preview of data:
```

	latitude	longitude	blk_no	road_name	postal_code	address	min_dist_mrt	min_dist_mall	orchard_dist	town	...	street_name	resale_price	month	remaining_lease	lease_commence_date	storey_range	_id	block	lease_remain_years	price_per_sqm
0	1.33908506817906	103.74705803294	283	TOH GUAN ROAD	600283	283 TOH GUAN RD	816.093559	464.334108	10166.435956	JURONG EAST	...	TOH GUAN RD	398000.0	2017-01	80 years 10 months	1998	01 TO 03	1	283	22	4422.222222
1	1.33908506817906	103.74705803294	283	TOH GUAN ROAD	600283	283 TOH GUAN RD	816.093559	464.334108	10166.435956	JURONG EAST	...	TOH GUAN RD	398000.0	2017-01	80 years 10 months	1998	01 TO 03	951	283	22	4422.222222
2	1.33908506817906	103.74705803294	283	TOH GUAN ROAD	600283	283 TOH GUAN RD	816.093559	464.334108	10166.435956	JURONG EAST	...	TOH GUAN RD	423000.0	2017-02	80 years 09 months	1998	04 TO 06	2328	283	22	4548.387097

3 rows × 23 columns

Data Preprocessing

- We need to clean the master data set before any meaningful analysis
- Check for
  - Nulls
  - Duplicates
  - Data types

[26]:
```
# generate preview of entries with null values
if len(combined[combined.isnull().any(axis=1)] != 0):
    print("\nPreview of data with null values:")
    display(combined[combined.isnull().any(axis=1)].head(3))
    missingno.matrix(combined)
    plt.show()
else:
    print("\nNo null values found")
```

```
No null values found
```

For the current dataset, we don't have any null values; else we will need to review the data for some clean-up before proceeding with analysis.

[27]:
```
# generate count statistics of duplicate entries
if len(combined[combined.duplicated()]) > 0:
    print("\n***Number of duplicated entries: ", len(combined[combined.duplicated()]))
    display(combined[combined.duplicated(keep=False)].sort_values(by=list(combined.columns)).head())
else:
    print("\nNo duplicated entries found")
```

```
No duplicated entries found
```

In this case, we don't have any duplicate entries; otherwise we will need to deduplicate the dataset. See the following code:

[28]:
```
combined.drop_duplicates(inplace=True)
```

From the data, we want to clean it up for further analysis.

[25]:
```
## clean up dataset

# set as numerical
#combined['resale_price'] = combined['resale_price'].astype('float64')
combined['floor_area_sqm'] = combined['floor_area_sqm'].astype('float64')
combined['lease_commence_date'] = combined['lease_commence_date'].astype('int64')


# set categorical data
combined['town'] = combined['town'].astype('category')
combined['flat_type'] = combined['flat_type'].astype('category')
combined['storey_range'] = combined['storey_range'].astype('category')

# set datetime data
combined['month'] = pd.to_datetime(combined['month'])


# set as string
combined['street_name'] = combined['street_name'].astype('str')
combined['remaining_lease'] = combined['remaining_lease'].astype('str')

## set as string to exclude from numerical analysis
combined['_id'] = combined['_id'].astype('str')
combined['resale_price'] = combined['resale_price'].astype('str')
combined['flat_model'] = combined['flat_model'].astype('str')


print(combined.info())
```

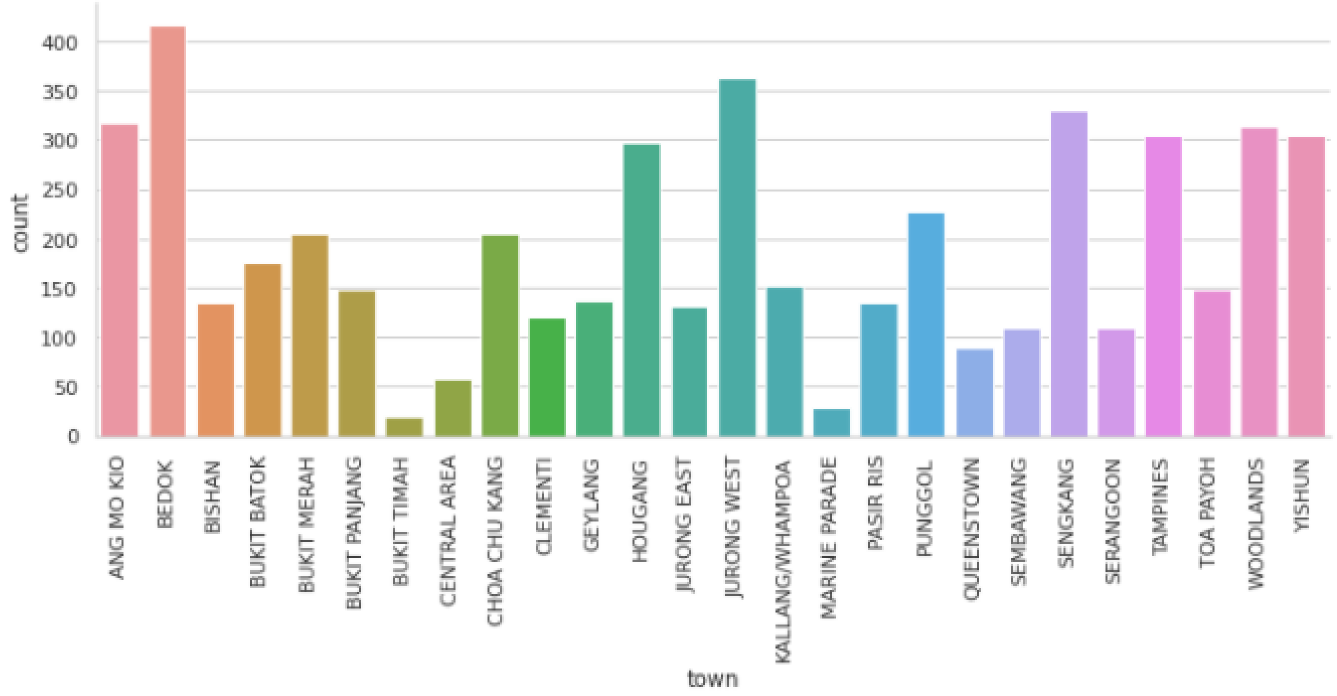
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 4990 entries, 0 to 4989
Data columns (total 23 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   latitude             4990 non-null   object
 1   longitude            4990 non-null   object
 2   blk_no               4990 non-null   object
 3   road_name            4990 non-null   object
 4   postal_code          4990 non-null   object
 5   address              4990 non-null   object
 6   min_dist_mrt         4990 non-null   float64
 7   min_dist_mall        4990 non-null   float64
 8   orchard_dist         4990 non-null   float64
 9   town                 4990 non-null   category
 10  flat_type            4990 non-null   category
 11  flat_model           4990 non-null   object
 12  floor_area_sqm       4990 non-null   float64
 13  street_name          4990 non-null   object
 14  resale_price         4990 non-null   object
 15  month                4990 non-null   datetime64[ns]
 16  remaining_lease      4990 non-null   object
 17  lease_commence_date  4990 non-null   int64
```

Exploratory Data Analysis

- Top unique counts
- Distribution Visualization
- Violin Plots
- Correlation Plots
- Machine Learning Models

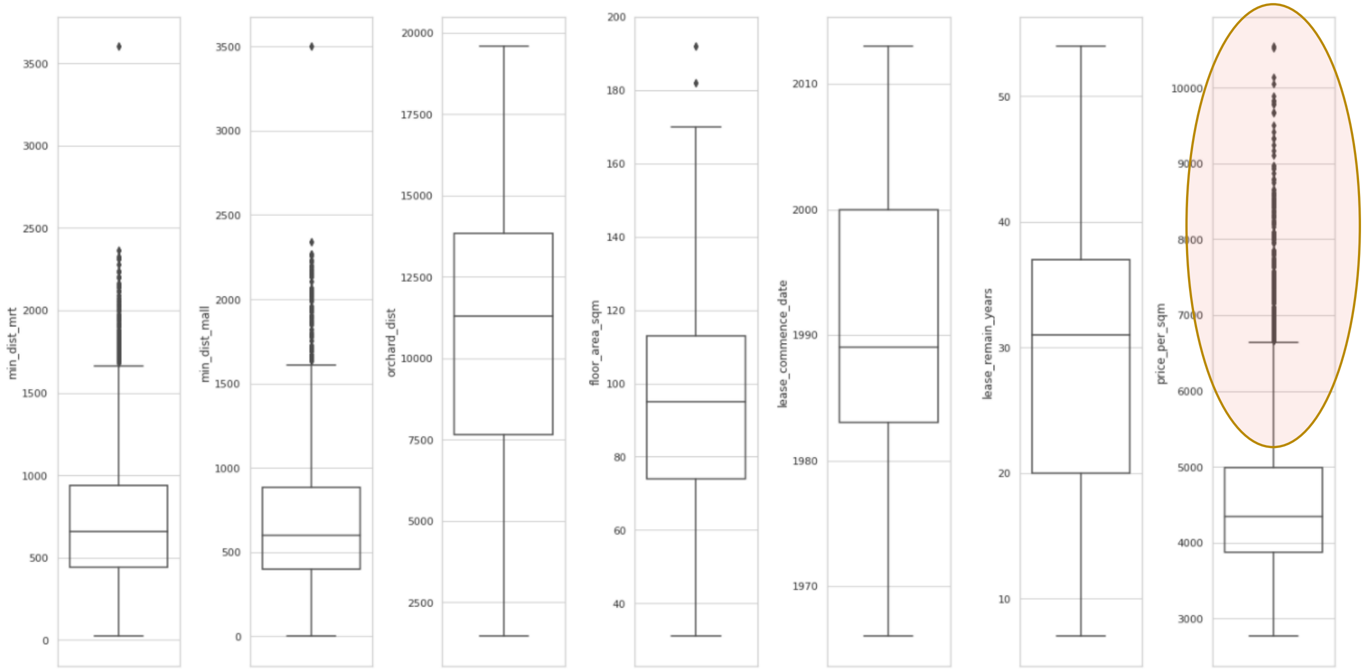
[31]:
```python
# Plot count distribution of categorical data

sns.set(style="whitegrid")
for col in combined.select_dtypes(include=['category']).columns:
    fig = sns.catplot(x=col, kind="count", data=combined, hue=None)
    fig.set_xticklabels(rotation=90)
    fig = plt.gcf()
    fig.set_size_inches(12,4)
    plt.show()
```



Exploratory Data Analysis

- Generally, numbers look good, no negative; if not we will need to do some investigation, and further cleaning since these numberical values of Resale Prices and Floor Area cannot possibly be negative.
- Look at the Resale Price's max value - it is way over the 75 percentile (2.5x more), implying outliers are highly likely to be present.
- The boxplot of Resale Price further confirms outliers, although we do know in real estate, there may be volatile and wide swing of prices due to many factors.

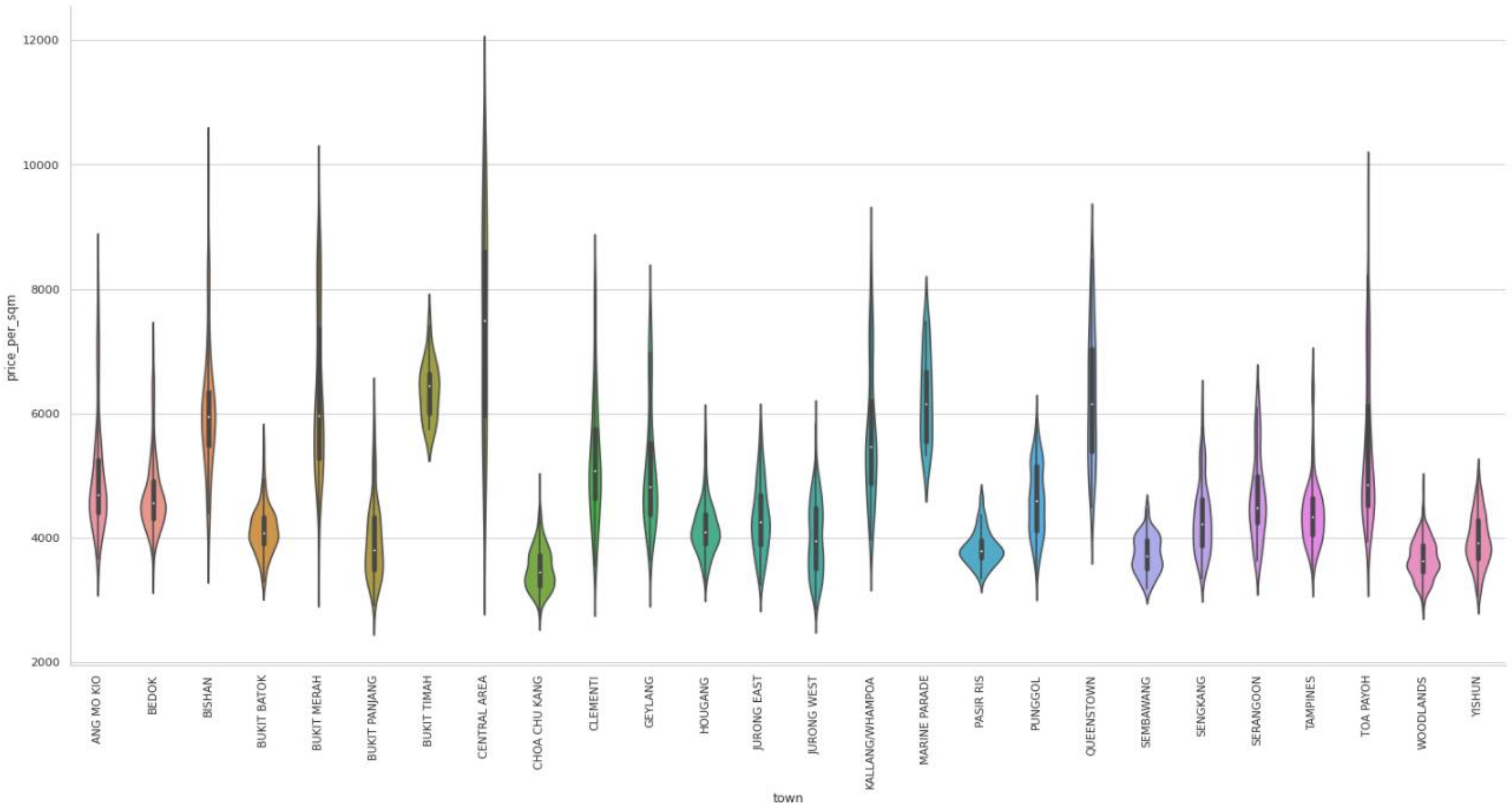



```
To check:
Distribution of numeric data
```

	count	mean	std	min	25%	50%	75%	max
min_dist_mrt	4990.0	743.753814	419.991431	21.879464	440.163441	657.286917	935.855968	3605.171617
min_dist_mall	4990.0	677.339842	388.264828	0.000030	396.522468	600.860068	885.484833	3502.101533
orchard_dist	4990.0	10626.123120	3901.780958	1456.287304	7648.795903	11285.010588	13833.548088	19593.855652
floor_area_sqm	4990.0	96.715551	24.305697	31.000000	74.000000	95.000000	113.000000	192.000000
lease_commence_date	4990.0	1991.302204	11.442364	1966.000000	1983.000000	1989.000000	2000.000000	2013.000000
lease_remain_years	4990.0	28.697796	11.442364	7.000000	20.000000	31.000000	37.000000	54.000000
price_per_sqm	4990.0	4587.788525	1096.576125	2760.683761	3866.666667	4345.974882	4983.606557	10552.380952

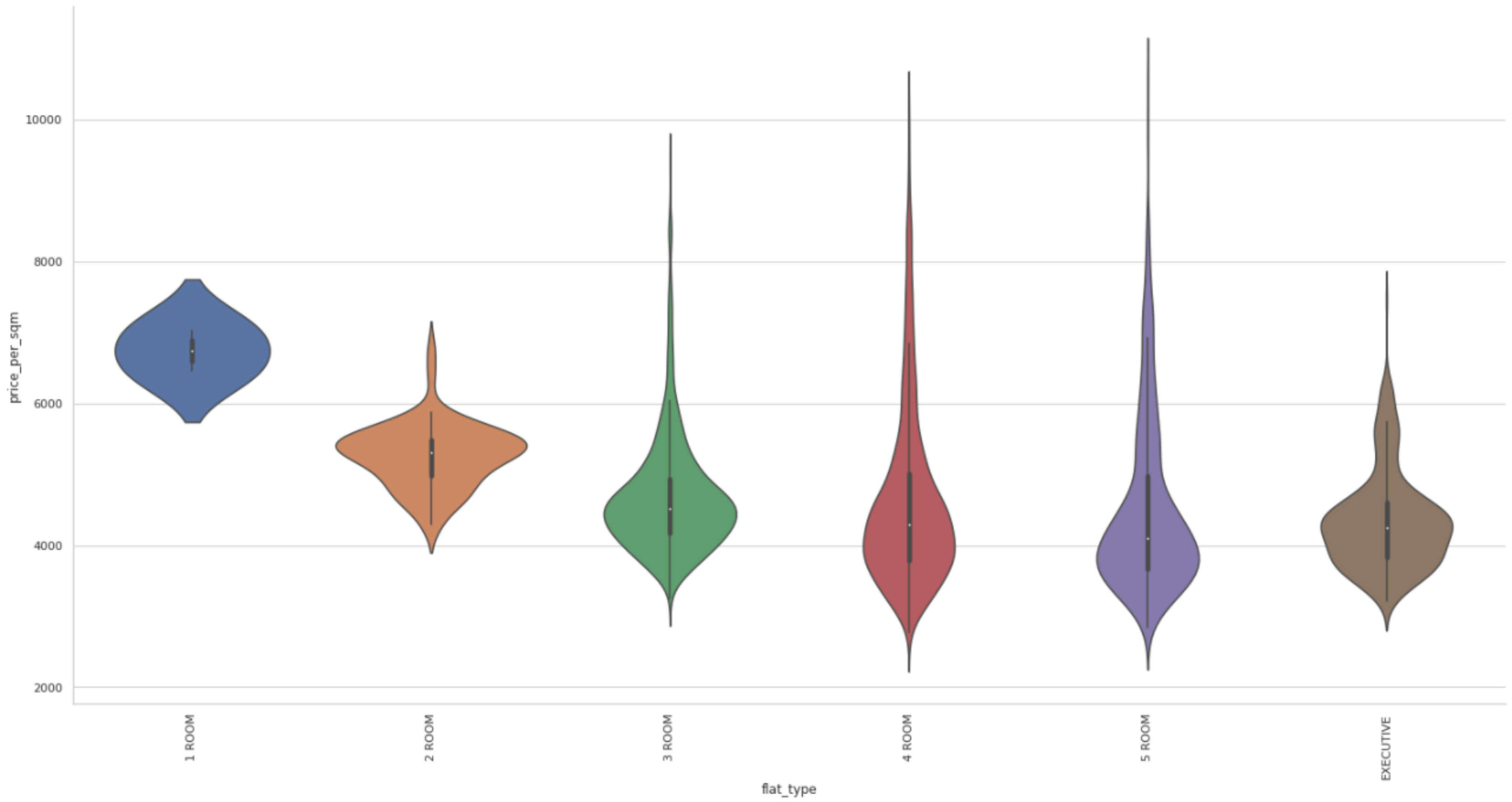
Exploratory Data Analysis

- The above violin plots for price per sqm for each town highlights the Central Area has a very large variance in pricing, and a higher than average median price, with low probability throughout the price range.
- On the other end, we can see towns such as Pasir Ris and Choa Chu Kang have lower variance and much lower median prices which higher probablity around the media prices.

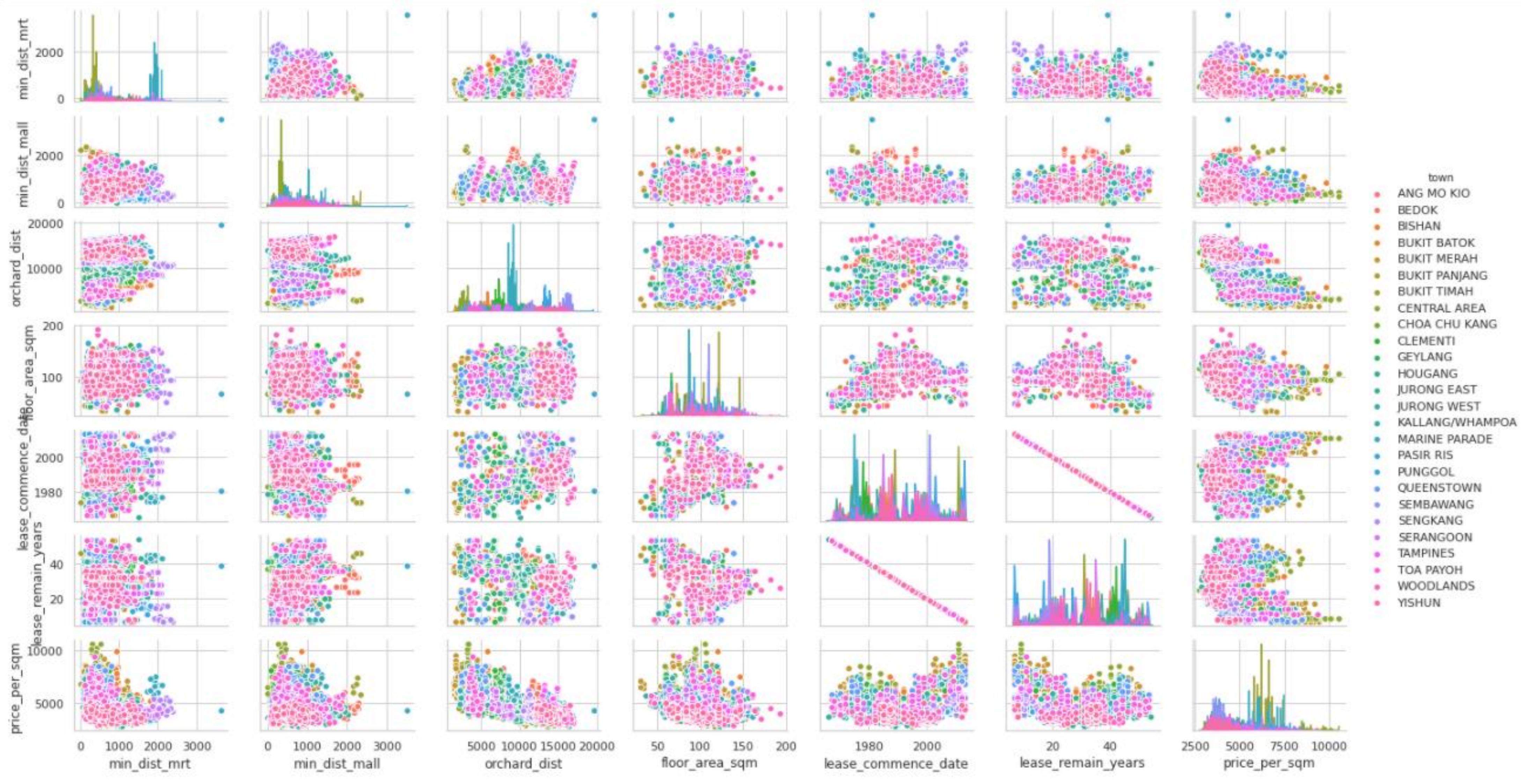


Exploratory Data Analysis

- Most outliers in 5 rooms flats wherre they are view as premium and are typically thriving in a seller's market.
- The converse is true in the case of 2-Room flat. The variance of the price is close to the median.
- In the case of the 2 Room flats, the median for prices per sqm is much higher than the others, presumably due to the much smaller overall quantum.
- The Median price per sqm for the other flat types hover around the same levels.



Exploratory Data Analysis



From the above, we can see there may be a correlation (negative) between price per sqm vs distance from Orchard (Central) area, and remaining years on the lease. To a less extent, there seems to be some correlation between the price as well as proximity to MRTs and Retail malls as well.

Exploratory Data Analysis

[43]:
```python
# corr matrix

corrMatrix = df_numerical_cols.corr()
sns.heatmap(corrMatrix,
        xticklabels=corrMatrix.columns,
        yticklabels=corrMatrix.columns,
        cmap='RdPu',
        annot=True)
```

[43]: <AxesSubplot:>

From the above we can see some good features, or predictors of price per sqm will be

- distance from orchard
- distance from MRT

While distance from malls, or number of remaining lease in years does not seem to play a huge part in the property prices.

Results

```python
[51]: X = df_reg[['min_dist_mrt', 'min_dist_mall',
             'orchard_dist', 'floor_area_sqm', 'lease_remain_years',
             'flat_type_mapped', 'storey_mean']]
      y = df_reg["price_per_sqm"]
      X = sm.add_constant(X)
      model = sm.OLS(y, X).fit()
      predictions = model.predict(X)
      # Print out the statistics
      model.summary()
```

[51]: OLS Regression Results

Dep. Variable:	price_per_sqm	R-squared:	0.661
Model:	OLS	Adj. R-squared:	0.660
Method:	Least Squares	F-statistic:	1385.
Date:	Sat, 12 Dec 2020	Prob (F-statistic):	0.00
Time:	18:44:29	Log-Likelihood:	-39313.
No. Observations:	4990	AIC:	7.864e+04
Df Residuals:	4982	BIC:	7.870e+04
Df Model:	7		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	7814.8855	67.978	114.961	0.000	7681.618	7948.153
min_dist_mrt	-0.5681	0.022	-25.948	0.000	-0.611	-0.525
min_dist_mall	-0.0351	0.024	-1.436	0.151	-0.083	0.013
orchard_dist	-0.1920	0.003	-72.888	0.000	-0.197	-0.187
floor_area_sqm	-11.4823	1.056	-10.871	0.000	-13.553	-9.412
lease_remain_years	-31.3771	0.945	-33.204	0.000	-33.230	-29.525
flat_type_mapped	207.6474	22.770	9.119	0.000	163.009	252.286
storey_mean	48.6296	1.794	27.102	0.000	45.112	52.147

Omnibus:	643.737	Durbin-Watson:	0.445
Prob(Omnibus):	0.000	Jarque-Bera (JB):	1260.245
Skew:	0.813	Prob(JB):	2.19e-274
Kurtosis:	4.848	Cond. No.	8.53e+04

Initial Model Analysis

- R-Squared — this the percentage of explained variance of the predictions. Our model can explain only about 0.661 or 66.1% of the variance in our data
- Or, the model is about $638.75 off in possibly predicting the prices.
- This means the margin of error is about 13.9%

This results are not really desirable or helpful in predicting the prices. How can the model be further improved? We want to try to improve our model's explanatory power by introducing our categorical variables into the regression model.

Conclusions

	coef	std err	t	P>|t|	[0.025	0.975]
const	7938.0706	82.669	96.022	0.000	7776.003	8100.138
min_dist_mrt	-0.5541	0.018	-30.994	0.000	-0.589	-0.519
min_dist_mall	-0.2788	0.020	-14.096	0.000	-0.318	-0.240
orchard_dist	-0.1067	0.008	-13.246	0.000	-0.122	-0.091
floor_area_sqm	-14.9737	0.798	-18.754	0.000	-16.539	-13.408
lease_remain_years	-45.2852	0.784	-57.792	0.000	-46.821	-43.749
flat_type_mapped	274.5145	16.862	16.280	0.000	241.457	307.572
storey_mean	44.2493	1.302	33.981	0.000	41.696	46.802
town-BEDOK	122.3183	45.225	2.705	0.007	33.658	210.979
town-BISHAN	631.9138	49.768	12.697	0.000	534.346	729.482
town-BUKIT BATOK	-683.5672	49.225	-13.887	0.000	-780.069	-587.065
town-BUKIT MERAH	462.5869	55.319	8.362	0.000	354.137	571.037
town-BUKIT PANJANG	-1228.4899	54.984	-22.343	0.000	-1336.282	-1120.698
town-BUKIT TIMAH	1212.9507	109.012	11.127	0.000	999.239	1426.662
town-CENTRAL AREA	1036.8098	76.163	13.613	0.000	887.497	1186.123
town-CHOA CHU KANG	-1400.3631	62.833	-22.287	0.000	-1523.544	-1277.182
town-CLEMENTI	270.9682	49.056	5.524	0.000	174.798	367.139
town-GEYLANG	-30.7388	47.633	-0.645	0.519	-124.120	62.642
town-HOUGANG	-783.4477	41.260	-18.988	0.000	-864.336	-702.559
town-JURONG EAST	-264.2369	54.905	-4.813	0.000	-371.876	-156.598
town-JURONG WEST	-637.2774	68.355	-9.323	0.000	-771.283	-503.272
town-KALLANG/WHAMPOA	-4.1033	53.661	-0.076	0.939	-109.302	101.096
town-MARINE PARADE	2405.5243	93.264	25.793	0.000	2222.685	2588.363
town-PASIR RIS	-284.7167	78.478	-3.628	0.000	-438.568	-130.866
town-PUNGGOL	-921.6586	64.413	-14.309	0.000	-1047.936	-795.381
town-QUEENSTOWN	636.3560	61.042	10.425	0.000	516.687	756.025
town-SEMBAWANG	-1338.1697	87.055	-15.372	0.000	-1508.835	-1167.504
town-SENGKANG	-1192.0827	51.404	-23.191	0.000	-1292.857	-1091.308
town-SERANGOON	-82.2967	51.576	-1.596	0.111	-183.409	18.815
town-TAMPINES	-36.1559	64.150	-0.564	0.573	-161.919	89.607
town-TOA PAYOH	37.8340	52.382	0.722	0.470	-64.857	140.525
town-WOODLANDS	-944.3048	73.274	-12.887	0.000	-1087.953	-800.656
town-YISHUN	-576.1741	60.050	-9.595	0.000	-693.898	-458.450

Omnibus:	579.334	Durbin-Watson:	0.781
Prob(Omnibus):	0.000	Jarque-Bera (JB):	1927.616
Skew:	0.582	Prob(JB):	0.00
Kurtosis:	5.814	Cond. No.	3.70e+05

[58]: OLS Regression Results

Dep. Variable:	price_per_sqm	R-squared:	0.828
Model:	OLS	Adj. R-squared:	0.827
Method:	Least Squares	F-statistic:	747.2
Date:	Sat, 12 Dec 2020	Prob (F-statistic):	0.00
Time:	18:49:07	Log-Likelihood:	-37614.
No. Observations:	4990	AIC:	7.529e+04
Df Residuals:	4957	BIC:	7.551e+04
Df Model:	32		
Covariance Type:	nonrobust		

- **The new model's Mean Absolute Error, 'Mean Squared Error, as well as Root Mean Squared Error, have all dropped, improving the accuracy of the model.**
- **R-Squared — this the percentage of explained variance of the predictions. Our model can explain up to 0.828 or 82.8% of the variance in our data**
- **Model's prediction error is up to $454.37 per sqm, or 9.9%. This is much better than the previous 13.9% error margin.**
- **The location area / town in which the property sits on, has a large and significant impact on its price per sqm. Bukit Timah, Central Area, Marine Parade are priced highed at premiums vis a vis estates in Geylang and Pasir Ris**

Improvements

- How this analysis can be further improved:
  - Due to limited api calls, our dataset is only limited to less than 5k data.
  - If we can have more data 100k, definitely the modelling and predictions may be further improved
  - As a further extension of the project, if there are more data for private properties, we can draw further insights.

Thank you for reading!