Housing Prices Prediction and Seasonality Analysis

Introduction

Accurate forecasting of housing prices plays a crucial role in various sectors, including real estate investment, urban planning, and policy-making. The housing market is influenced by a wide range of factors, making it a complex and dynamic system to model. Time series forecasting methods are valuable tools that can be used for understanding and predicting market trends.

In this report, we explore the application of the ARIMA (AutoRegressive Integrated Moving Average) model to forecast housing prices. In addition to the standard ARIMA approach, we incorporate a seasonal component using the SARIMA (Seasonal ARIMA) model to account for recurring patterns throughout the year. This allows us to remove seasonal effects from the price data such as the consistent price increases during certain periods (e.g., spring and summer months) and focus on the underlying trends, enabling a more accurate prediction of housing values across different times of the year.The family of ARIMA models was selected to model this problem because it has proven effective in various real-world forecasting applications. The goal of the project is to find the most suitable model for our data set and predict future values on the training part of the data.

Applying the ARIMA model to the data set effectively requires several key steps. First, the raw data must be preprocessed to ensure it is suitable for modeling. This includes verifying and, if necessary, transforming the data to achieve stationarity. Once the data is stationary, the next step is to identify and select the appropriate model parameters. These steps will be outlined in the following sections of the report, followed by testing and evaluation of the model’s performance.

Data

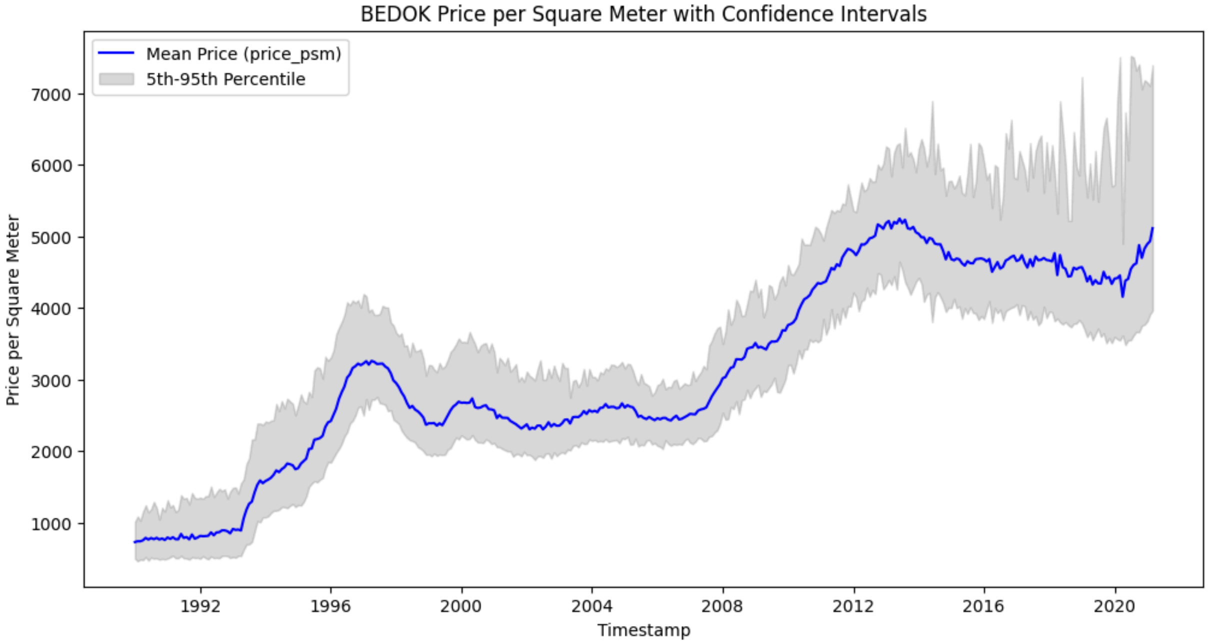
The dataset used in this analysis comprises detailed records of property transactions across various regions of Singapore. For each transaction, it provides the price per square meter in USD, as well as the area in which the property is located. Every entry is timestamped at the monthly level, indicating both the month and year when the sale occurred. Spanning from 1990 to 2021, this dataset offers a comprehensive view of the real estate market over more than three decades, allowing for an in-depth examination of long-term trends and regional variations in property prices. We concluded our analysis on the district called Bedok, but the code could be used for any other district from the data set.

Preprocessing the Data

Preprocessing the data that is going to be used is a crucial step when applying ARIMA models, as these models assume a clean and structured input with consistent time intervals. Any missing values, irregular frequencies, or irrelevant information can lead to inaccurate forecasts or model errors.

Our script begins by filtering the entries to only include transactions in the town of Bedok. It then converts the "month" column into a proper datetime format and sets it as the index for the DataFrame. The relevant columns (date and the price per square meter) are then selected. Since ARIMA is designed to model a univariate time series, meaning it expects a single value for each time period, the data is resampled to calculate the average monthly price to ensure that the data is at a consistent monthly frequency. We also extracted 5th and 95th percentile of the monthly data prices, so we could plot it together with mean values to obtain a broader picture of our data.

We plotted the dataset of monthly Bedok price per square meter over the observed time span, with the mean price (blue line) and its 5th--95th percentile range (shaded area). The x-axis represents the chronological progression from the early 1990s to around 2020, and the y-axis measures the price per square meter. The shaded region highlights the spread of the data, indicating increased volatility in more recent years. Overall, the chart reveals a steady upward trend, suggesting long-term growth in housing prices in the Bedok area.



Stationarity

The need for stationary data

Stationarity means that statistical properties like the mean, variance, and autocorrelation, stay constant over time. In simple terms, the series behaves consistently and predictably, without trends or changing patterns that evolve as time goes on. It is often hard to assume a strict stationarity, (when the joint distribution of $(r_{t_1}, \ldots, r_{t_k})$ is identical to that of $r_{t_1+t}, \ldots, r_{t_k+t})$ hence many time series models assume \textit{weak stationarity} if moth mean of r_t and covariance between r_t and $r_{t-\ell}$.The ARIMA model assumes stationarity in the data, and hence it is critical for model performance, and predictions and estimates can become unreliable and misleading.

Transformation of our data set

After preprocessing the data, our data set consisted of monthly prices in absolute values in US dollars per square meter, which is often not stationary and that was clearly the observable in the plot of our data. We tackled this problem by using the log transformation on the data and then taking the difference of it. With that technique we got an approximation of the relative (percentage) change from one period to the next. Mathematically explained, we used the formula:

$$\Delta \log(P_t) = \log(P_t) - \log(P_{t-1}) = \log\left(\frac{P_t}{P_{t-1}}\right)$$

After this transformation, we suspected that our data became stationary, so we used a statistical test to check our assumptions.

Testing for stationarity

The test performed is the Augmented Dickey-Fuller test (ADF-test), which check for unit roots in time series, which indicates nonstationarity.

The test uses the test statistic:

$$x_t = c_t + \beta x_{t-1} + \sum_{i=1}^{p-1} \phi_i \Delta x_{t-i} + e_t$$

Where the test hypotheses are:

$$H_0 : \phi = 1 \Leftrightarrow \beta = 0 \rightarrow \text{non-stationary}$$

$$H_A : \phi < 1 \Leftrightarrow \beta \neq 0 \rightarrow \text{stationary}$$

Considering the results of the ADF-test we see that, the ADF test statistic equals -3.7515, with the corresponding p-Value of 0.0034 which leads us to reject the null hypothesis, and hence we believe the data is stationary.

In the case where we would not have received the result that the data was stationary, the data would have been transformed until the results would have implied stationarity. This would have been done using different transformations or using more diffrences, to stabilize the variance and hence help with stationarity.

```
ADF Statistic: -3.7515
The data is stationary. The p value equals 0.00344493308440573
```

Determining model parameters

Now that it is ensured that the data is stationary, tests can be performed to find the optimal parameters for the ARIMA model. The model consists of `p`, `q` and `d`

The `p` is the AR-term, which tells how many of the last `p` variables should be included for prediction \citep[p. 38]{tsay2010analysis}. The determination of `p` will be done through with PACF (partial autocorrelation function). The `q` is the MA-term, which helps determine the number of lags in the model. This is done with ACF (Autocorrelation function). The `d` is the differencing, which tells how many times the non-stationary series had to be differenced to become stationary. As described in Section , the data was only log transformed once, before receiving sufficiently good results on the tests. Hence, the value for `d` in the ARIMA model, should be at least 1, but could be also more since adding more differences keeps data stationary. We decided to test the models also for second differences, so when `d` equalls 2.

ACF - Determining q

ACF is the autocorrelation function which measure correlation between a time series and its past values - the lags. This helps identify seasonal patterns or trends. The correlation between r_t and $r_{t-\ell}$ is the lag-ℓ autocorrelation of r_t, denoted ρ_ℓ, and is defined as:

$$p_\ell = \frac{Cov(r_t, r_{t-\ell)}}{\sqrt{Var(r_t)Var(r_{t-\ell})}} = \frac{\mathrm{Cov}(r_t, r_{t-\ell})}{\mathrm{Var}(r_t)} = \frac{\gamma_\ell}{\gamma_0}$$

In the next Figure we can see the plot of ACF for our transformed data set.

The value of q is implied in this plot by where the plot "cuts off" the consecutive spikes beyond in the plot. This implies an early cut-off already at q = 1, but there are also significant spikes at 2 and 4, though these are not consecutive. Hence this implies that `q` equals 1, but should also be tested for values up to 4.

PACF - Determining p

Through PACF one receive information regarding how much a past value (lag) helps predict the present. For instance, how much r_{t-2} helps predict r_t or if r_{t-1} is sufficient. To figure out which lags to include, AR models of different orders are considered:

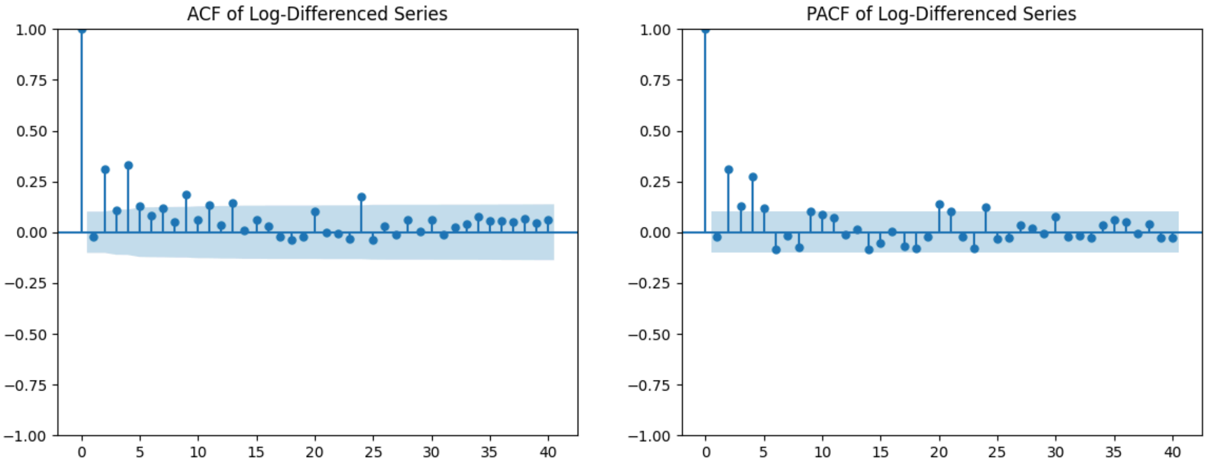
$$r_t = \phi_{0,1} + \phi_{1,1} r_{t-1} + e_{1t}$$

$$r_t = \phi_{0,2} + \phi_{1,2} r_{t-1} + \phi_{2,2} r_{t-2} + e_{2t}$$

$$r_t = \phi_{0,3} + \phi_{1,3} r_{t-1} + \phi_{2,3} r_{t-2} + \phi_{3,3} r_{t-3} + e_{3t}$$

Considering this, each $\phi_{j,j}$ is the PACF at lag j, which explains how much "only" the considered lag j helps predict, once earlier lags already are accounted for. Hence, in an AR(p) model the PACF will drop close to 0 after lag `p` (it will cut off). This is the `p` to use in the ARIMA model. Hence we consider the plot of PACF below, to determine the parameter.

As mentioned, where the PACF plot drops, is where the lags no longer contribute to the prediction. In the plot, the value drops to 0 on the first lag, before becoming high until including the fifth lag. This implies that the optimal `p`-value would be somewhere between 1 and 5.



AIC and BIC testing

In addition to ACF and PACF testing, the model selection criterias Akaike information criterion (AIC) and Schwarz-Bayesian information criterion (BIC) were also used to choose the best group of the assumed parameters. We used a loop to check all the models with the potentially best parameter choices and compared them to each other.

AIC is done using the formula

$$AIC(k) = ln(\hat{\sigma}^2) + \frac{2k}{T}$$

BIC is performed using the formula

$$BIC(k) = ln(\hat{\sigma}^2) + \frac{k}{T} ln(T)$$

The same selection rule applies to both of these testing models; one computes the AIC and BIC for the set of different parameters and then compares them to each other. Then one should choose the model with set of parameters with the smallest value for AIC and BIC.

The output from running the python code for AIC and BIC testing is shown below, and has been performed on the ARIMA model for both `p`, `d` and `q`. This results hence shows that the best parameters for the ARIMA model are `p` = 1, `d` = 2 and `q` = 1 if we follow the BIC criterium, which often prefers simpler models. If we would choose regarding the AIC criterium, we would choose the more complex `p` = 5, `d` = 2 and `q` = 4 model. We decided use the more complex model, proposed by AIC in the upcoming analysis.

Ther results of the some of the best models are shown in the table:

p	d	q	AIC	BIC
5	2	4	4107.893273	4147.109057
3	1	4	4112.561051	4143.955097
5	1	3	4115.095460	4150.413763
2	2	4	4118.010553	4145.461602
5	2	3	4120.520970	4155.815176
5	2	1	4121.524688	4148.975737
5	2	2	4121.946664	4153.319291
4	2	4	4122.422867	4157.717072
3	2	4	4122.900572	4154.273200
1	2	1	4122.981228	4134.745963

Prediction and evaluation

Evaluation metrics

To evaluate the forecasting performance of our model, the dataset was split into a training set (the first 90% of the observations) and a testing set (the last 10%). This approach enabled us to assess the models' ability to predict unseen data. We concluded two comparable forecasts, one with ARIMA and other one with SARIMA model on the same data and so we could directly compare their performance.

To assess the performance of our forecasting models, we use two common error metrics: the Mean Squared Error (MSE) and the Mean Absolute Error (MAE). The MSE is defined as

$$\mathrm{MSE} = \frac{1}{n} \sum_{t=1}^{n} (\hat{y}_t - y_t)^2,$$

where (\hat{y}_t) are the predicted values and (y_t) are the actual observations. By squaring the errors, MSE places a greater penalty on larger discrepancies, making it particularly useful for highlighting significant forecasting errors. In contrast, the MAE is given by

$$\mathrm{MAE} = \frac{1}{n} \sum_{t=1}^{n} \left| \hat{y}_t - y_t \right|,$$

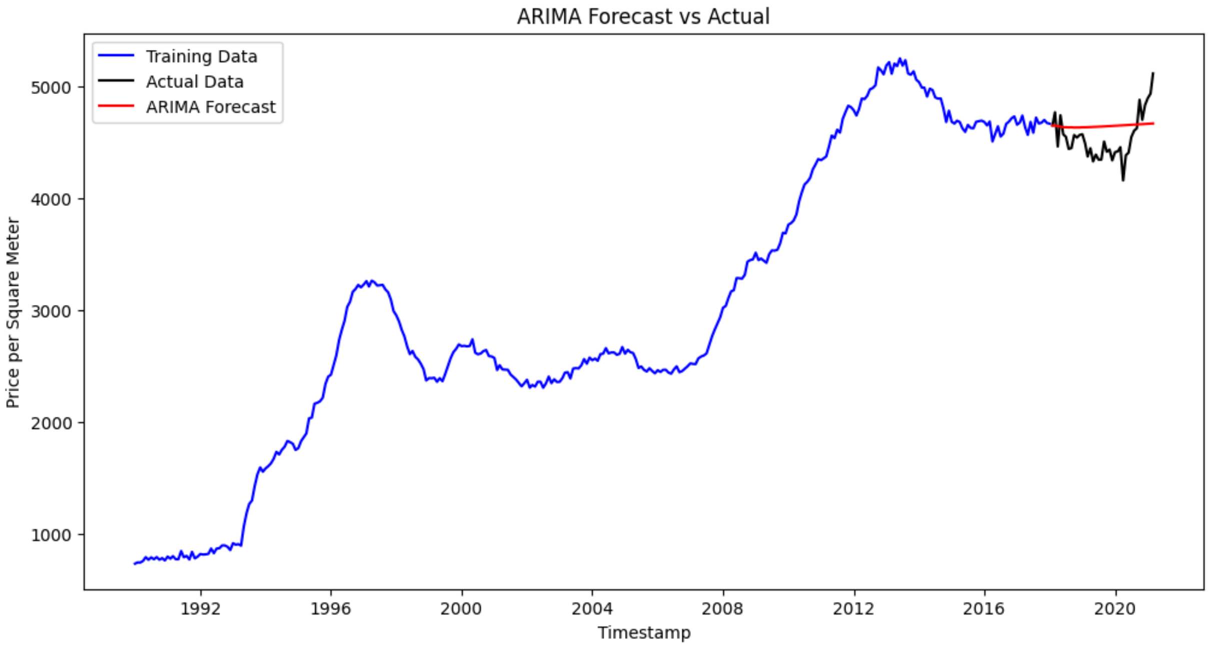
which calculates the average absolute difference between predictions and observations, thus providing a straightforward measure of average error in the same units as the data. Employing both MSE and MAE allows us to capture a more comprehensive picture of model accuracy, balancing the sensitivity to large errors (MSE) with the interpretability of average error magnitude (MAE).

To further examine the models we will also analyze the diagnostic plots, which serve as a comprehensive check on the model assumptions. For the standardized residuals plot, we expect to see the residuals fluctuating randomly around zero without any apparent trend or pattern, which would indicate that the model has successfully captured the data's dynamics. In the histogram of residuals, when compared with a normal density curve, a bell-shaped distribution is desirable, suggesting that the residuals follow a normal distribution. The Normal Q-Q plot further confirms this by displaying the quantiles of the residuals against the theoretical quantiles of a normal distribution; a linear pattern along the 45-degree line signifies normality. Finally, the autocorrelation function (ACF) plot of the residuals should show no significant spikes beyond the confidence intervals, indicating that there is no remaining serial correlation. Together, these plots affirm that the residuals behave like white noise, a key requirement for the validity of the ARIMA and SARIMA models.

ARIMA Model Evaluation

The ARIMA model was applied on the log-transformed and second-differenced series with parameters $(p, d, q) = (5, 2, 4)$, as suggested by the ACF/PACF analysis and confirmed by AIC/BIC criteria. Forecasts were generated for the testing period and subsequently back-transformed to the original scale.

One can also visually inspect the plot of the prediction and compare it to the original values of the data. We can see that the prediciton does not capture the volatility of the data very good and that the prediction seems almost constant, but overall the prediction seems reasonable close to the actual values.

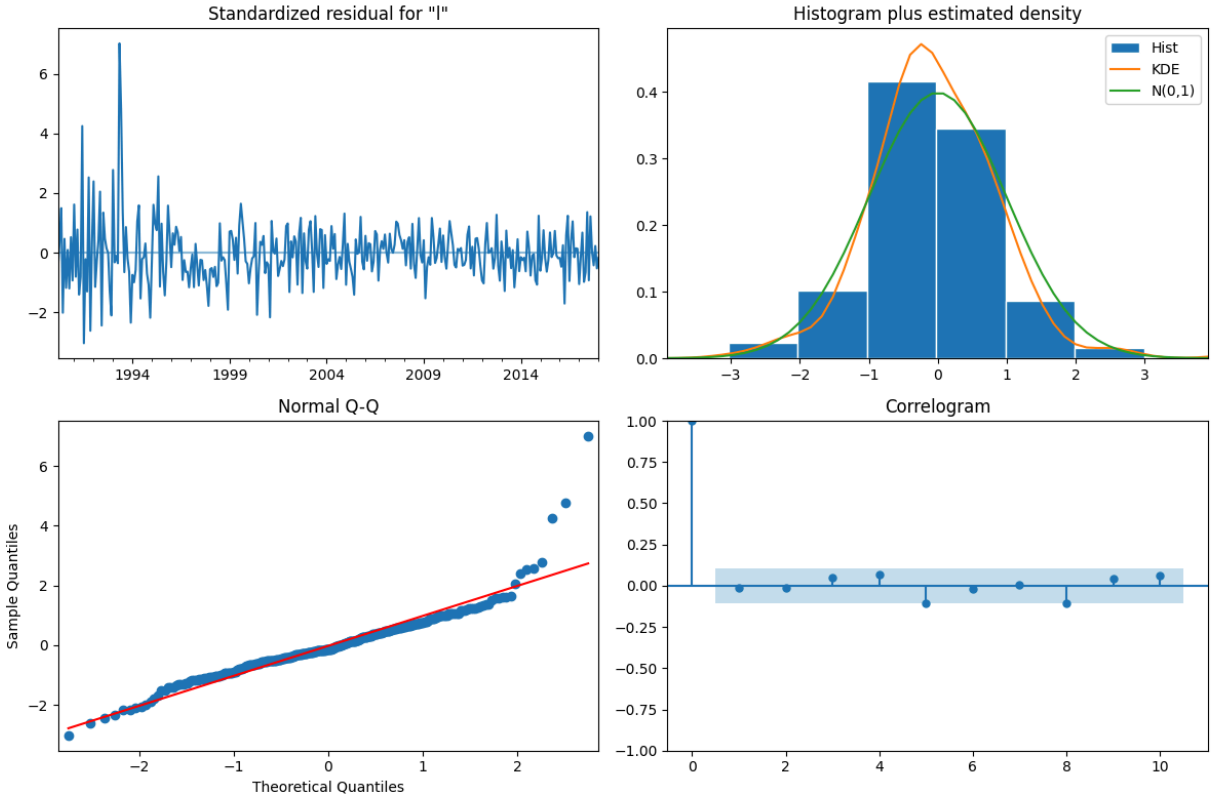


The performance of the model was quantified above descripted MSE and MAE metrics. Next table shows the computed metrics:

Metric	Value
MSE	46516.09
MAE	186.29

```
ARIMA Model - MSE: 46516.0907, MAE: 186.2929
```

We also inspected the diagnostic plots of the model, which can be seen in the next output.. On the plots, in the top-left panel we can see, the standardized residuals fluctuate around zero with no evident pattern for most of the dataset, only a few spikes are spotted in the beggining of the data which is completely normal, still this suggests that the model has adequately captured the primary time-dependent structure. The top-right panel shows the residual histogram and its estimated density, both of which indicate a distribution close to normal, albeit with slightly heavier tails and slightly negative mean. The bottom-left panel is the Normal Q--Q plot, where most points align well with the theoretical line, reinforcing the notion of approximate normality but also showing some deviation in the upper tail. Finally, the bottom-right panel, the autocorrelation (correlogram) of the residuals, does exhibit one significant lags, implying there is no more considerable autocorrelation present. Collectively, these diagnostics suggest that the chosen model is reasonably specified for the data.

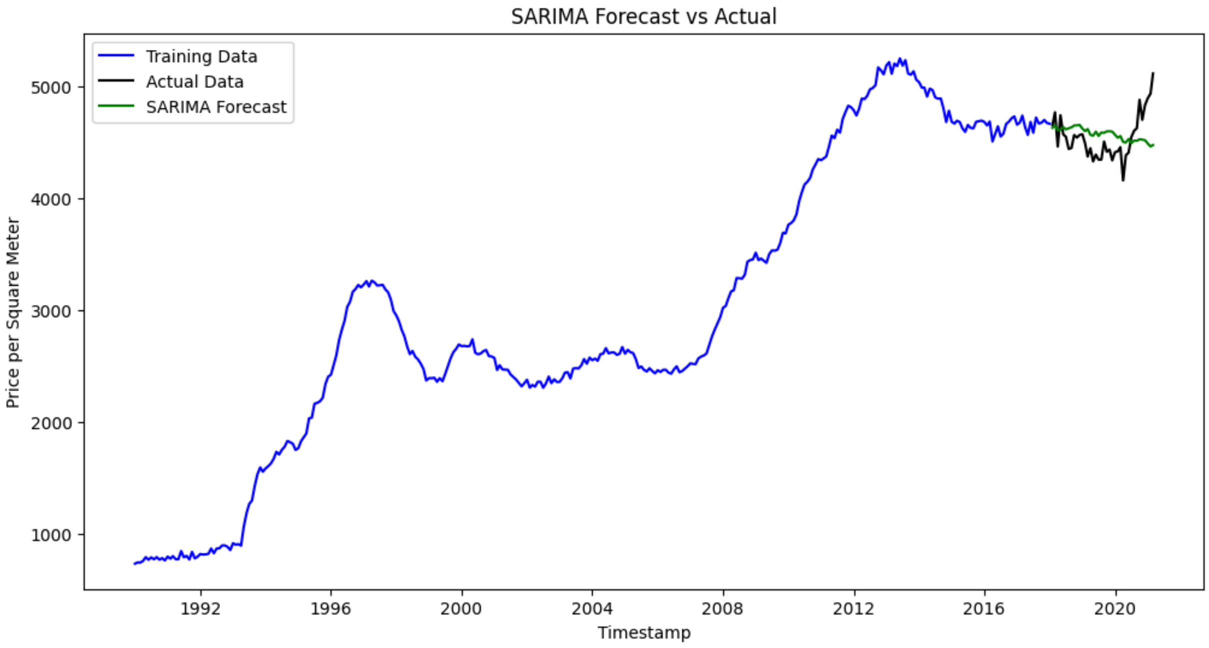


SARIMA model evaluation

Next step in our analysis was to try to improve the predictions of the simple ARIMA model. We wanted to capture the seasonal patterns, which we speculate occur on the yearly level (meaning once on every 12 measurements in our data set), so a SARIMA model was implemented by incorporating seasonal parameters $(P, D, Q, s) = (5, 2, 4, 12)$. Similar to the ARIMA approach, the SARIMA model was trained on the first 90% of the dataset and used to forecast the remaining 10%. The forecasts were then back-transformed, and the corresponding error metrics are summarized in the next table:

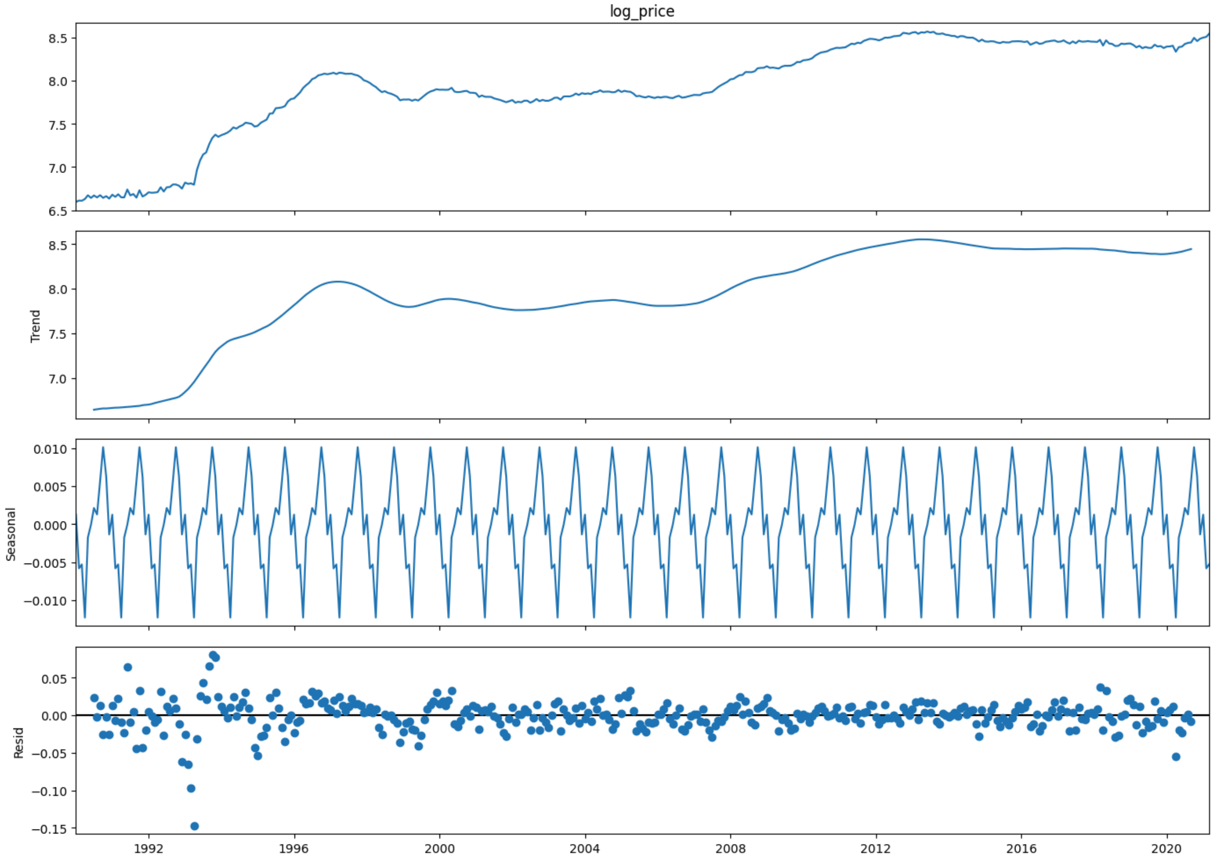
Metric	**Value**
MSE	48569.40
MAE	180.70

We can also see the visualization of the prediction below. In this case one can spot a slightly declining prediction which for the big part of the test set corresponds very well with the acctual data set, but then our prediction fails to recognize the change in the trend and turn upwards. This could be a consequence that we are trying to predict prices for quite a few years ahead which is too dificult for models like this. Overall we are very satisfied with the model, specially in short term predictions.

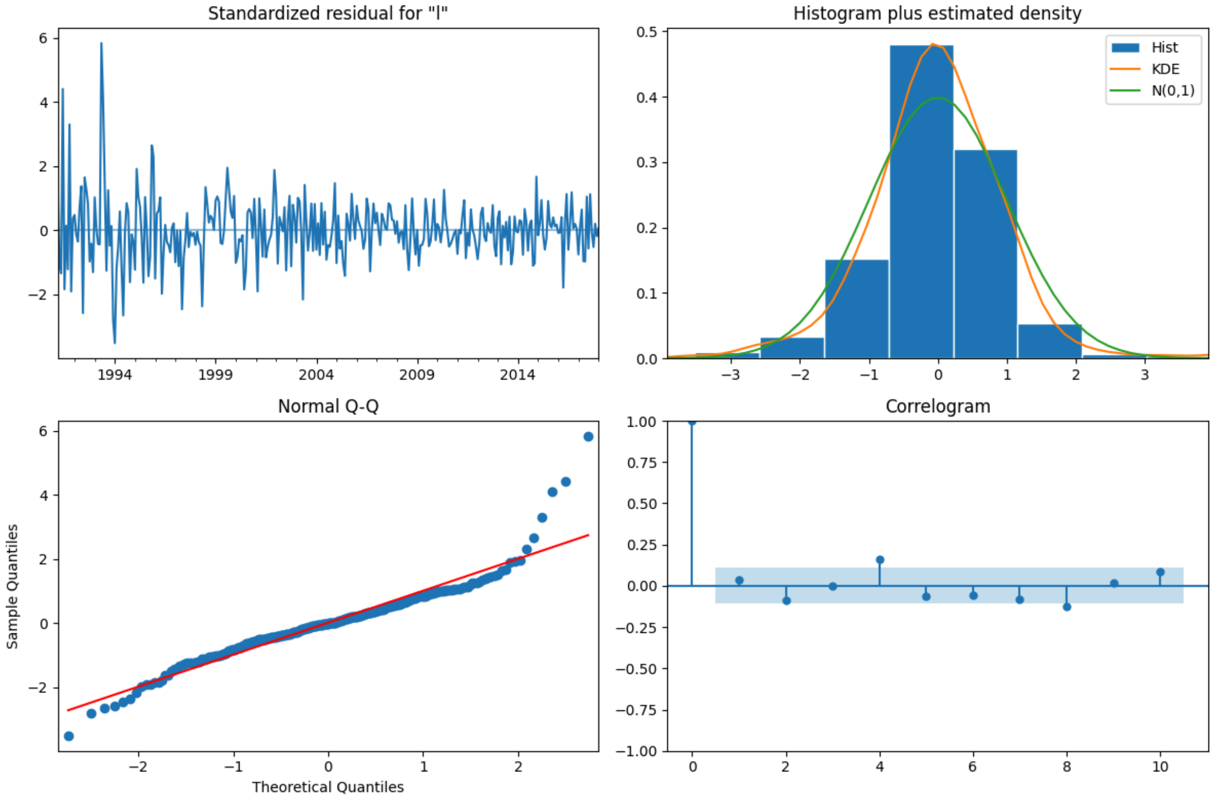


```
SARIMA Model - MSE: 48569.4010, MAE: 180.7027
```

Next figure shows the SARIMA decomposition displays the decomposition of the log-transformed series into trend, seasonal, and residual components. The top panel shows the smoothed $log\ price$, while the second panel illustrates a gently evolving trend over time. In the third panel, a clear yearly seasonality emerges, reflecting a recurring pattern each year. Finally, the bottom panel contains the residuals, which appear centered around zero. This decomposition confirms the strong seasonal dynamics in the data, justifying the use of a SARIMA model with explicit seasonal terms.



Again, on diagnostic plot also for this model (top-left panel) shows the standardized residuals from the SARIMA model fluctuating around zero without noticeable patterns, indicating that the main temporal structure has been captured. The top-right panel compares the residual histogram with a fitted density, suggesting approximate normality. The Normal Q--Q plot (bottom-left) further supports this, though some slight deviation is visible in the tails. Lastly, the correlogram (bottom-right) exhibits one significant autocorrelation, which we do not know the cause of. Still, all together we can confirm that also this model fits well to the problem at hand.



Conclusion

Both of the models that we developed showed to be very comparable to each other. While ARIMA model had lower MSE value on the test set, SARIMA model had lower MAE value so it hard to explicitly say which model is more suitable for predicting housing prices. But we are satisfied with performances of both models, specifically when looking at the visualization of the price prediction where both models showed decent performance. We are also very satisfied with the decomposition of seasonality in SARIMA model which is clearly visible and confirms that seasonality in housing market exists at least in the observed city in Asia.

In our opinion these models offer significant advantages in the real-world prediction of housing prices by providing actionable insights into both the general market trend and seasonal fluctuations. Moreover, their flexibility makes them easily adaptable for use in other cities and real estate markets. By adjusting the model parameters to local market characteristics, urban planners, investors, and policymakers can leverage these forecasting tools to make more informed decisions. Our code provides a very good tool for analyzing this seasonality as it can be made to work on a lot of different data sets with minimal need of preprocessing the data.

In short, the ARIMA and SARIMA models not only provide robust forecasts for Bedok but we believe models like these also serve as valuable, general-purpose tools for housing market analysis across different urban settings.

Use of LLM

During the work on this project we used LLM, specifically ChatGPT to help with writing the programming code. It turned out to be useful as it was able to write big chunks of code fast, but it struggled with understanding of a broader sense of our code and needed a lot of guidance and very specific instructions on what we really want to achieve. It was very useful especially for preprocesing of the data and generating plots. The biggest disadvantage of it was interpretation of the results where he was usually unable to provide any useful information.

This notebook was converted with convert.ploomber.io