DATA ANALYSIS AND FORECASTING FOR HOUSE PRICING DATA

Elif DOĞAN DAR
December, 2021

DATA

Housing prices data consists of 3 tables: unit_master, unit_rent_master and rental master.

1) unit_master is a data frame with 1100 observations and 5 variables.

A - UnitNumber (int): Unique number given to the rental.

B - UnitType (factor with 3 levels): Studio Apartment (STD) , 1 Bedroom Apartment(1BR) and 2 Bedroom Apartment(2BR)

C - UnitPlan (factor with 11 levels): UnitType is further divided into subcategories. There are 100 observations per UnitPlan.

STD: STD-L STD-M&A STD-M&B STD-S&A STD-S&B STD-S&C

1BR: 1BR-L&A 1BR-L&B 1BR-S&A 1BR-S&B

2BR: 2BR-S

D - Sqft(int): Area of the rental. It has 6 different values

450 for STD-S&A, STD-S&B, STD-S&C

650 for STD-M&A STD-M&B

850 for STD-L

750 for 1BR-S&A 1BR-S&B

1050 for 1BR-L&A 1BR-L&B

1250 for 2BR-S

E - Floor(int): Floor of the rental taking values between 1 to 10, it does not affect the pricing directly.

2) rental_master is a data frame with 7236 observations and 5 variables.

LeaseNo (int) : Unique number given to lease of the rental.

CustomerNo (int) : Unique number given to the customer.

UnitNumber (int) : It connects rental_master to unit_master.

StartDate (Date) : Start date of the lease. Values are between 2012-03-08 and 2021-03-06.

EndDate (Date) : End date of the lease. Values are between 2014-12-01 and 2021-03-06. If the lease didn't end till 2021-03-06, then this value is NA. Therefore, missing values here are not random, we know that they should be after 2021-03-06.

3) unit_rent_master is a data frame with 836 observations of 4 variables. It gives rise to 11 time series with a window of 76 months.

UnitPlan (factor with 11 levels): It connects unit_rent_master to unit_master.

StartDate (Date) : Start date of the particular month of the pricing.

Values are between 2014-12-01 and 2021-03-01.

EndDate (Date) : End date of the particular month of the pricing.

Values are between 2014-12-31 and 2021-03-31.

RentRate (int) : Rent rate in dollars for the particular month.

We have to fit and forecast the time series of the rentals for each level of the UnitPlan.

TIME SERIES

We are given 11 time series consisting of rental rates by UnitPlan, however UnitPlans are aggregated according to sqft and therefore some information is redundant. There are 6 different time series (Figure 1): STD-S, STD-M, STD-L, 1BR-S, 1BR-L and 2BR.

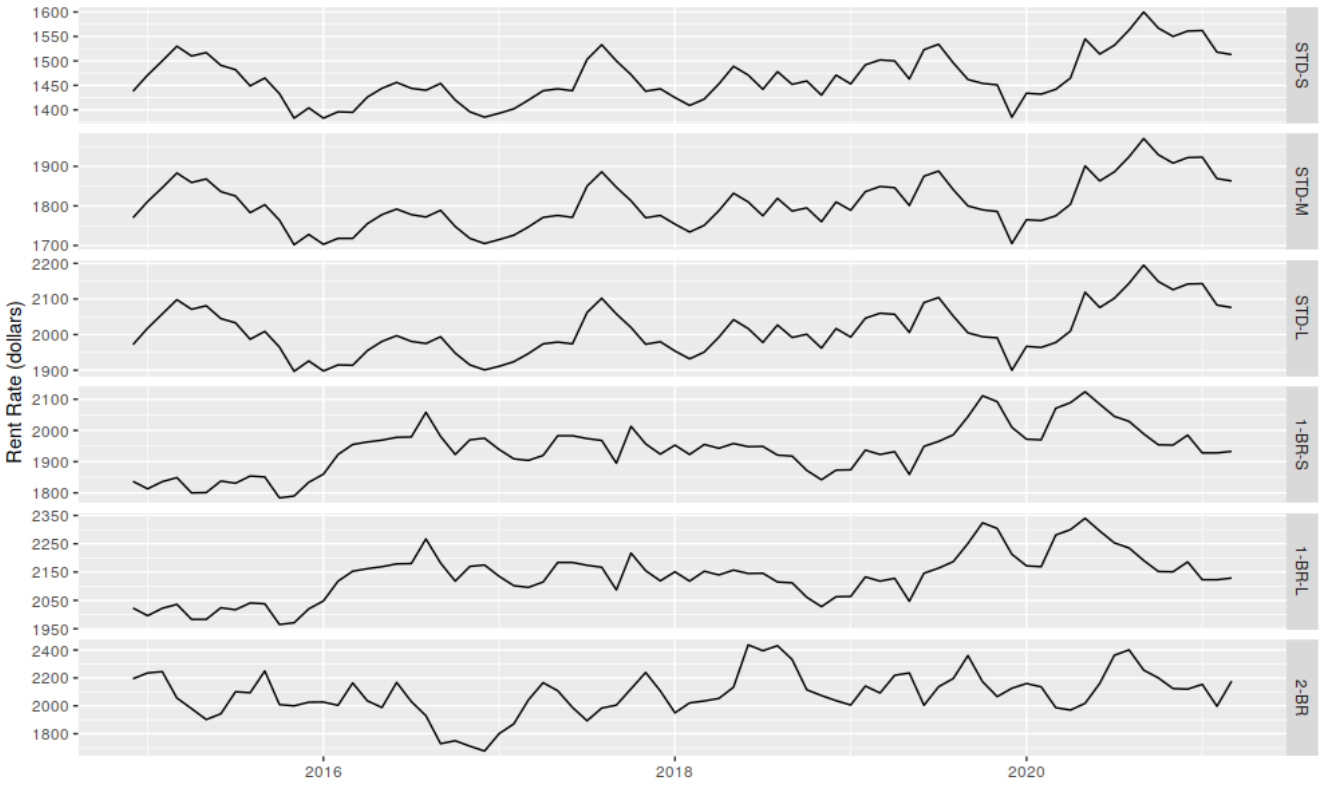

Figure 1 Time series of the apartments according to unit plans.

From here onwards, the analysis will be based on studio apartments, we will refer to it as std. You can find analyses for one bedroom and two-bedroom apartments in the appendix.

Relation among time series with the same UnitType:

If we look at the plots (Figure 1), STD series and 1BR series have almost the same shape. This phenomenon is resulting from the fact that all STD series are getting affected by the scarcity of the aggregated STD products. First, it seems like there is a shift between them, meaning that the relationship between sqft and rent rate is linear. But when we look at the plot of their differences, we see that the change is not constant, it has the same shape like series themselves. Therefore, we can conclude that the effect of area change is proportional to the prices. I tried a few different relations and found that there is almost a perfect fit between STD-M prices and (650/450)*0.852*STD-S prices (Figure 2). Therefore, when we know or forecast the

prices for one series, we can deduct the prices of the other. This will leave us with 3 time series to work with, namely STD, 1BR and 2BR.

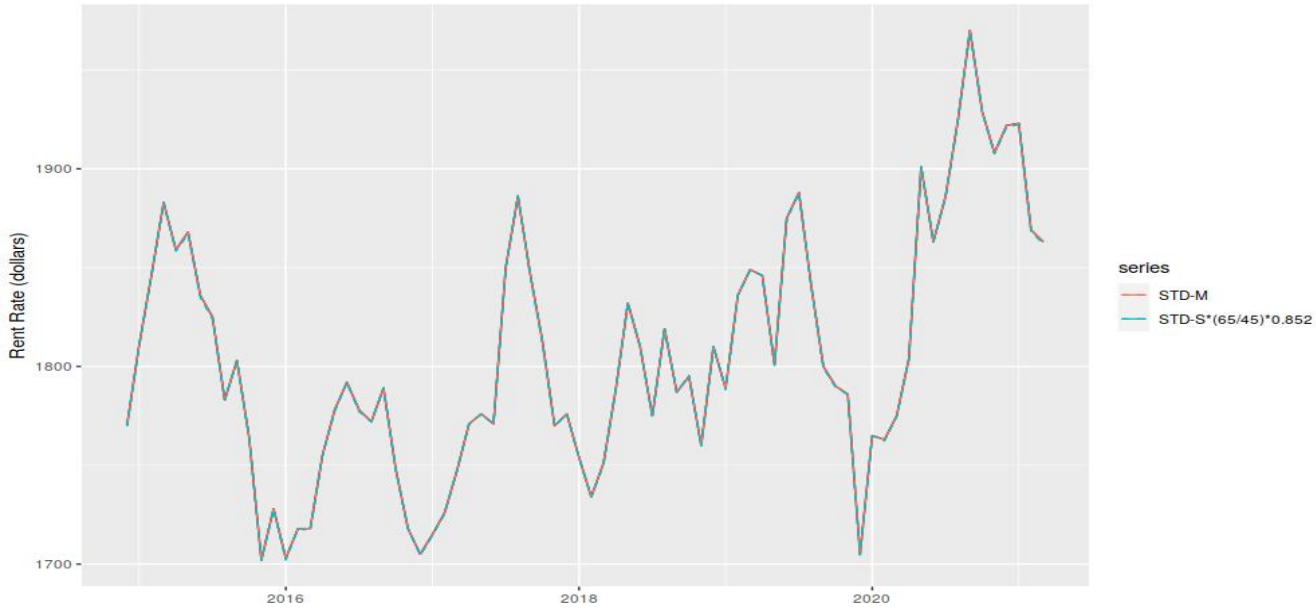

Figure 2 Relation between the series for STD-S and STD-M.

We can also label local maximas (peaks) and minimas (valleys) to have an easier inspection (Figure 3).

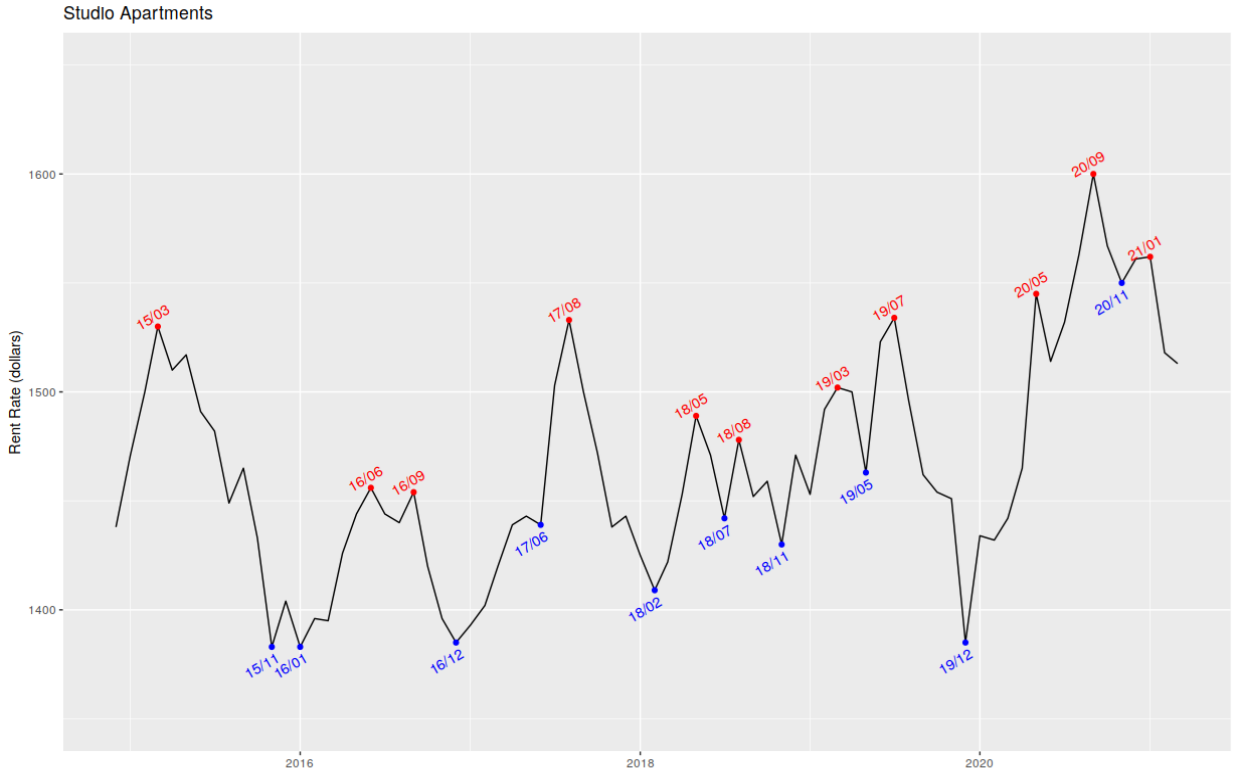

Figure 3 Local minima and maximas for the studio apartment prices.

Moving Averages and Moving Standard Deviations:

To have a better understanding of the data, we can look at the moving averages and moving standard deviations of the series. As can be seen in the Figure 4, moving averages of a window 4 smoothens the curve and tells us that there is a seasonality in our data, every year around summer months prices increase and decrease afterwards. Only exception is 2015, where peak seems to occur earlier than summer months. When we change the window to 12 in moving averages, we get rid of the seasonality component of the data, this is nothing but the trend of the data. We will see it shortly after using decomposition of the time series.

When we apply the moving standard deviations with a window 12, we see the volatility of the series stripped from the seasonality effect. As can be seen in the Figure 4, volatility changes over the years. Volatility is lower in 2016, 2018 and the first half of 2019, compared to the higher volatility in other years.

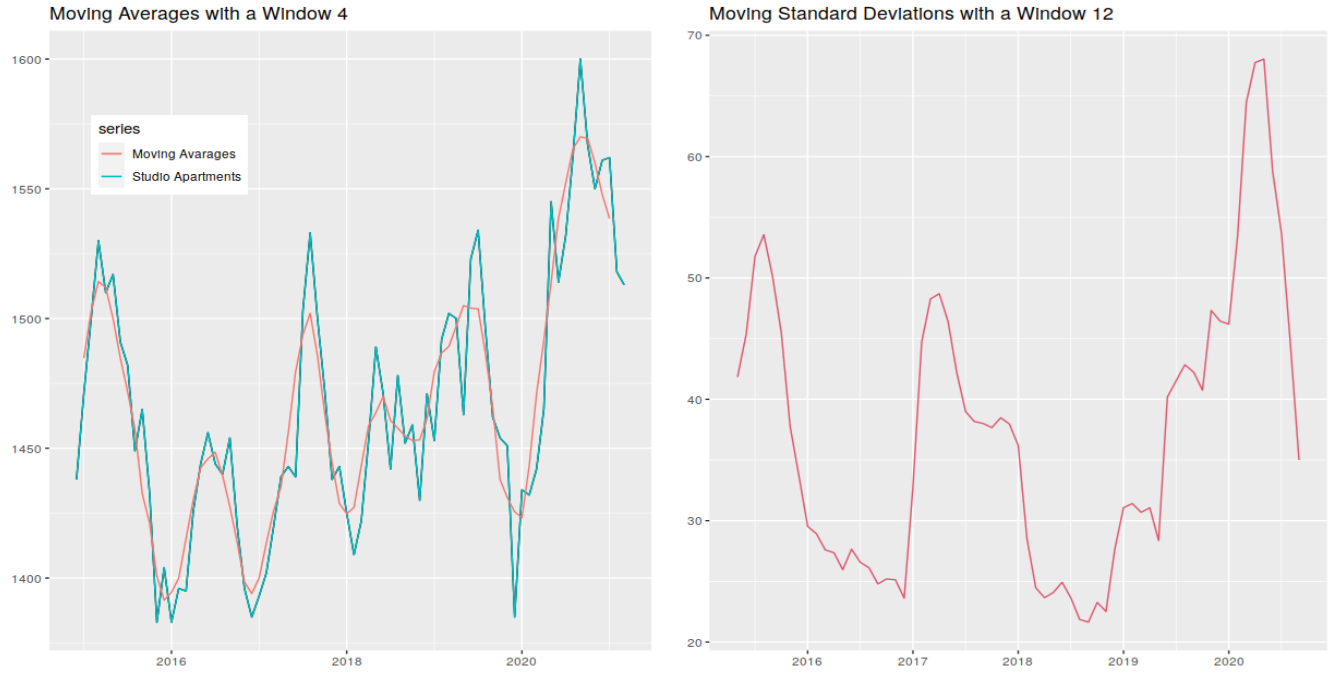

Figure 4 Moving averages and moving standard deviation for studio apartments.

Decomposition:

To have a general overview of the data, we can look at the decomposition of the data into 3 parts: trend, seasonality, and remainder part(Figure 6). This method has the advantage of giving us a more unified view instead of looking at the parts separately. Trend component shows that overall prices decreased until 2017, increased afterwards until the second half of the 2018 and almost got stabilized afterwards. Seasonality component shows that the prices increase around summer months and decrease afterwards. Height of the peaks tend to increase over time. There is a need for further investigations to see if this change is statistically significant. Seasonality plot where time series is plotted against the years makes it easier to see the differences between years (Figure 5). When we look at the remainder part, we see that there is still some pattern. This shows us that using only trend and seasonality of the time series is not enough to capture the whole mechanism behind. External variables or more complicated models are needed.

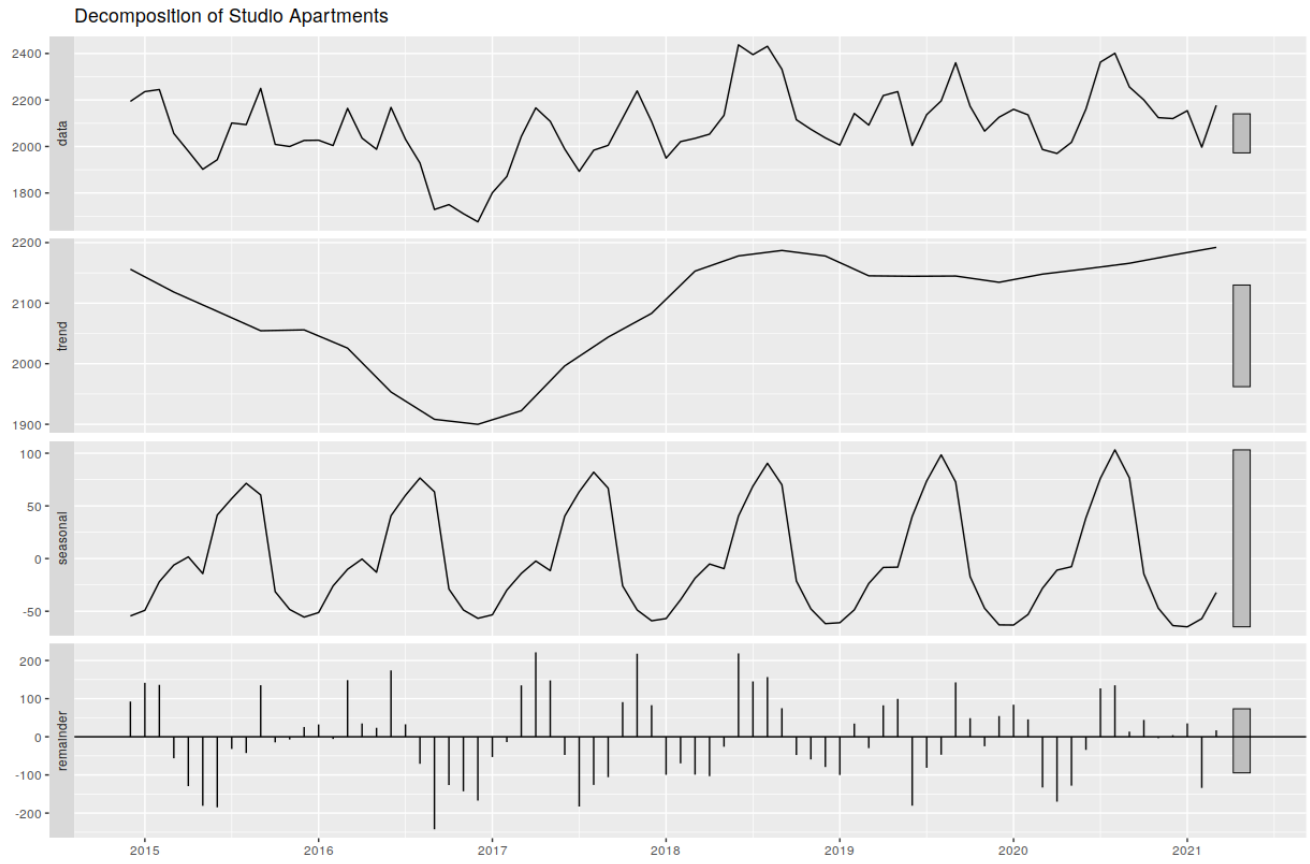

Figure 6 Decomposition of the time series into trend, seasonality and the remainder.

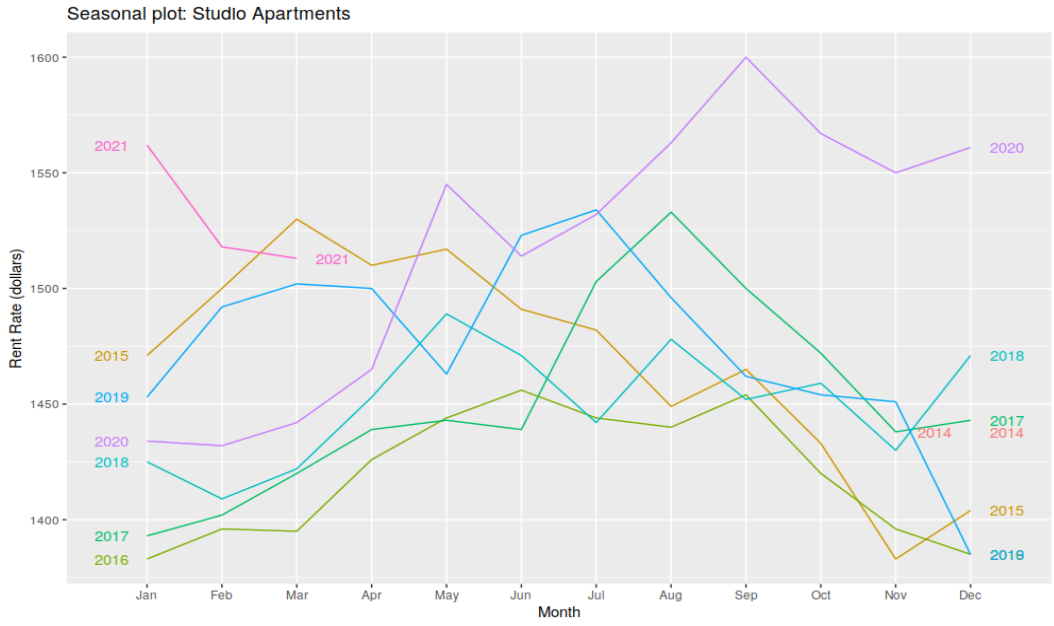

Figure 5 Season plot of the time series.

Lag Plots:

In time series data, most of the times data at the future points are correlated with the past. Most of the models are built on this assumption. To have an overall view of this, we can look at the ACF (Autocorrelation Function) and PACF (Partial Autocorrelation Function) plots (Figure 7). ACF values at lag j, gives the correlation coefficient between the time series and the series itself j step before, called lag j. So, it shows how much the data j unit before affects the series at a given point. Points outside the blue lines are considered to be statistically different than zero. We can see from the ACF plot that there is significant autocorrelation up to 4 lags. And wavy shape also suggests seasonality.

On the other hand, ACF doesn't tell the whole story. When we look at the effect of lag j to the current point, we should consider the indirect effect of lags in between too. For example, is lag 2 affecting today directly, or it is only because it affects lag 1 and therefore have an indirect affect? PACF handles that consideration for us. PACF value at lag j is the direct effect of lag j stripped from earlier lags. When we look at the PACF plot, we see that lag1, lag 5 and lag 24 are significant. However, it is highly improbable that a time point that far directly affects the value today. Since we are applying a statistical test with an expected error of 0.05, out of 20 tests one will give wrong results. Therefore, we might ignore lag 24. Only explanation I can come up with on this issue is that there is a big event in every 2 years which affects the housing prices. However, we don't have this knowledge and our time series has only 5 years. Therefore, we cannot deduce the importance by just examining the data.

Although ACF and PACF plots are highly informative, we need to approach them with caution. They are based on Pearson correlation coefficient; therefore, they are only measuring linear relations. If there is a strong non-linear relationship at lag j, ACF won't be able to give us that information. When the model that we are implementing can only capture linear relationships (such as ARIMA), this is not an issue. But for more complicated models, we might consider different lags not suggested by these plots too. One way to deduct non-linear relations might be making a visual inspection. We can use Lag Plots for this purpose (Figure 8). For a strong linear relationship, we expect the lines to gather around the diagonal. As can be seen in the Figure 8, there is a significant linear relation at lag 1 and strength of the relation decreases over time.

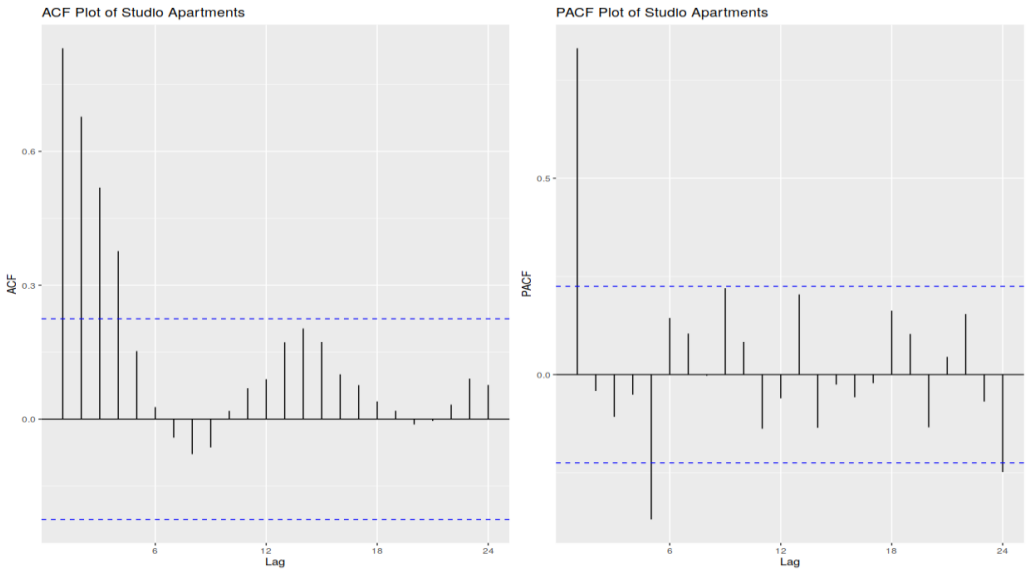

Figure 7 ACF and PACF plots of the time series.

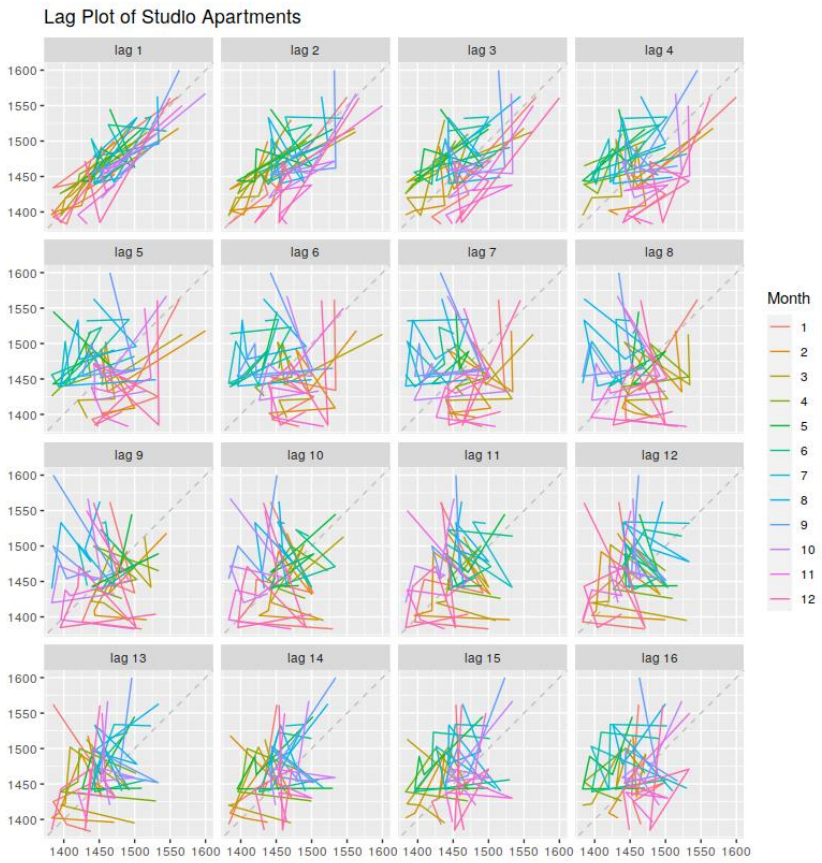

Figure 8 Lag plot of the time series.

Stationarity:

In most of the models we use, we assume the stationarity of the time series. Data should have constant mean and standard deviation over time. There shouldn't be autocorrelation and there shouldn't be seasonality. If series at hand is not stationary, we take the first differences to achieve stationarity. If it doesn't work, we can take second differences. If the data is seasonal, taking seasonal difference might help. There are visual inspection methods as well as formal statistical tests to check stationarity.

Since all statistical tests have different powers and probability of giving a wrong decision with 0.05, we applied multiple tests. Augmented Dickey–Fuller Test (ADF), Phillips–Perron Unit Root Test and Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test. We expect a p-value lower than 0.05 in the first two tests, and a p-value higher than 0.05 in the KPSS test. Results of the tests for the studio apartments are stationary, non-stationary and non-stationary respectively. Therefore, we can assume that the series is non-stationary and apply first differences. This time, all tests unanimously suggest stationarity. For a visual inspection, I used some built-in functions to check the change in the mean, variance and the structure of the series and the first differences (Figure 9). As can be seen in the Figure 9, differencing removed the changes in the structure.

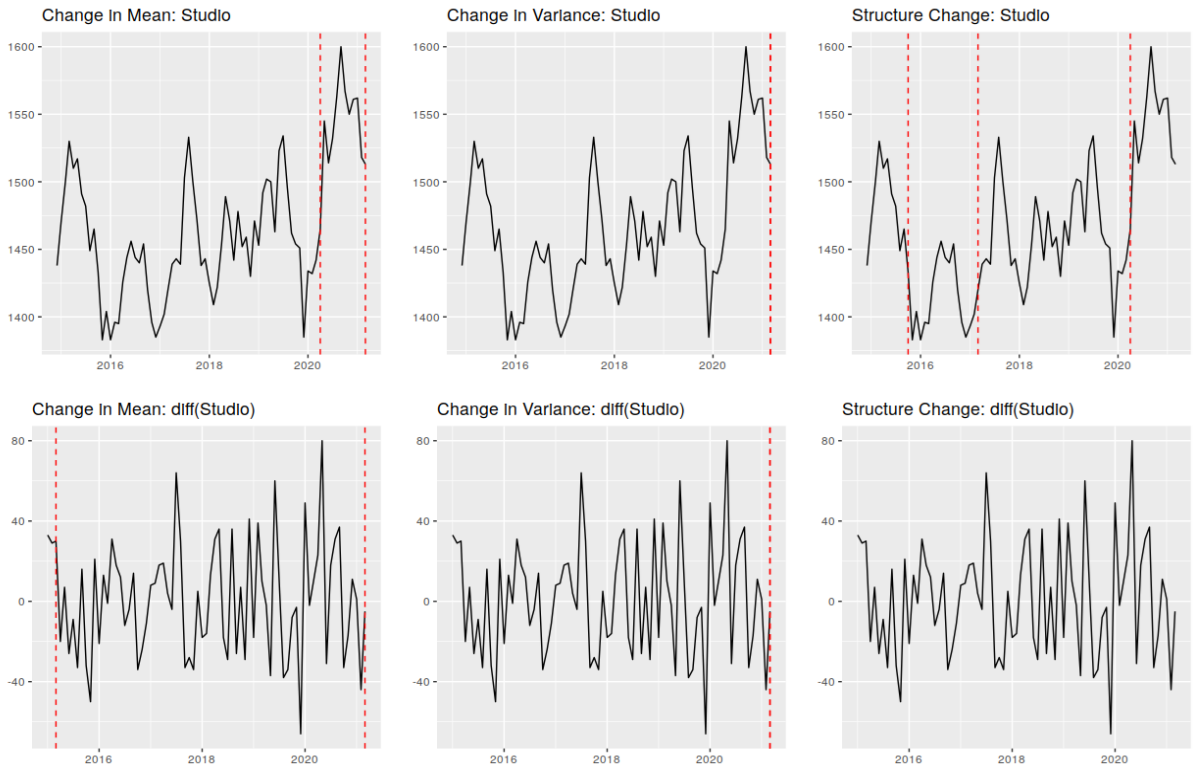

Figure 9 Change in mean, variance and the structure for both std and diff(std) series.

Outlier Detection

When there are obvious outliers in data, our models might not function as expected and behave poorly. If our method is not robust to outliers we might need to treat the outliers before the analysis. Here, we have to be cautious. Is the outlier results from an outlier in the regressor? Then, we might not need to remove or replace the outlier. Do outliers have a repeating pattern, such as every year in August prices are doubled? It might be resulting from, let's say an event being held in every August. In that case, we would want to keep this information in the data and find models which can incorporate those points.

If outliers result from an error or a process which won't be seen in the future again, we can treat those outliers with various methods. We can replace the outlier with the mean or median of other data points. For seasonal data such as monthly data, we can take the mean or median of the corresponding month. Alternatively, we can predict the outliers using other data points with more complicated algorithms.

In our case, none of the series has any outliers.

SUMMARY STATISTICS AND EXPLORATORY VARIABLES

Move-In/ Move-Out/ Renewals:

The time series of the number of customers moved-in, moved-out or renewed the lease at particular months give valuable information for pricing decisions.

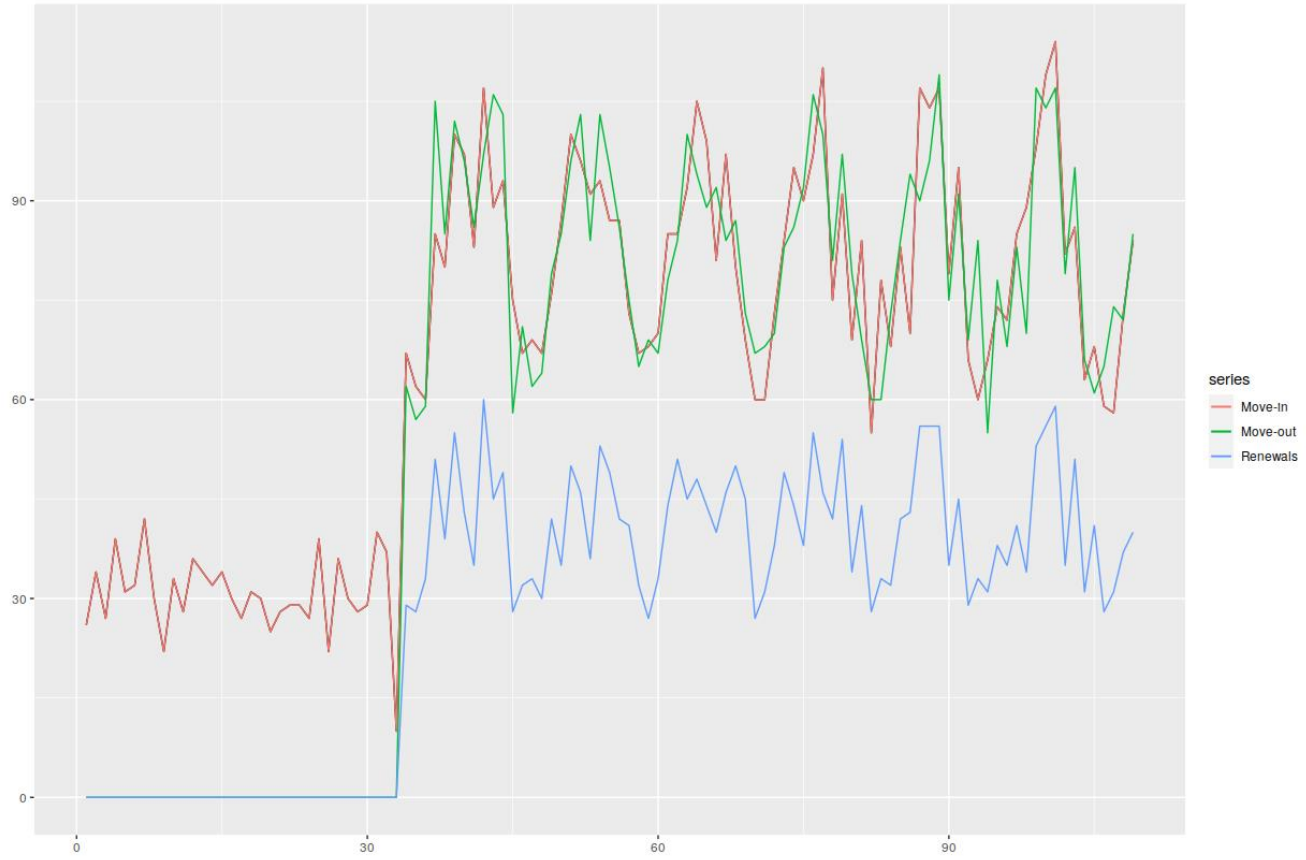

Figure 10 Monthly move-in move-out and renewal numbers for std.

Occupancy:

Occupancy can be seen as the overall effect of move-in, move-out and renewals (Figure 10). Pricing of the rentals is directly proportional to the scarcity of the product. As occupancy gets higher, prices increase to not leave possible profit on the table. As occupancy gets lower, prices decrease to attract more customers. To utilize this information, I calculated occupancy rates of studio, one bedroom and two-bedroom apartments separately for each month at the time series as well as overall occupancy. Occupancy of the studio apartments doesn't follow the overall occupancy exactly although there are certain similarities. It shows that segmentation at the Unit Type level was a good choice. Occupancy of studio apartments exhibits similar shape with the corresponding time series (Figure 12) and the linear relationship between them can also be seen in the scatter plot provided (Figure 11).

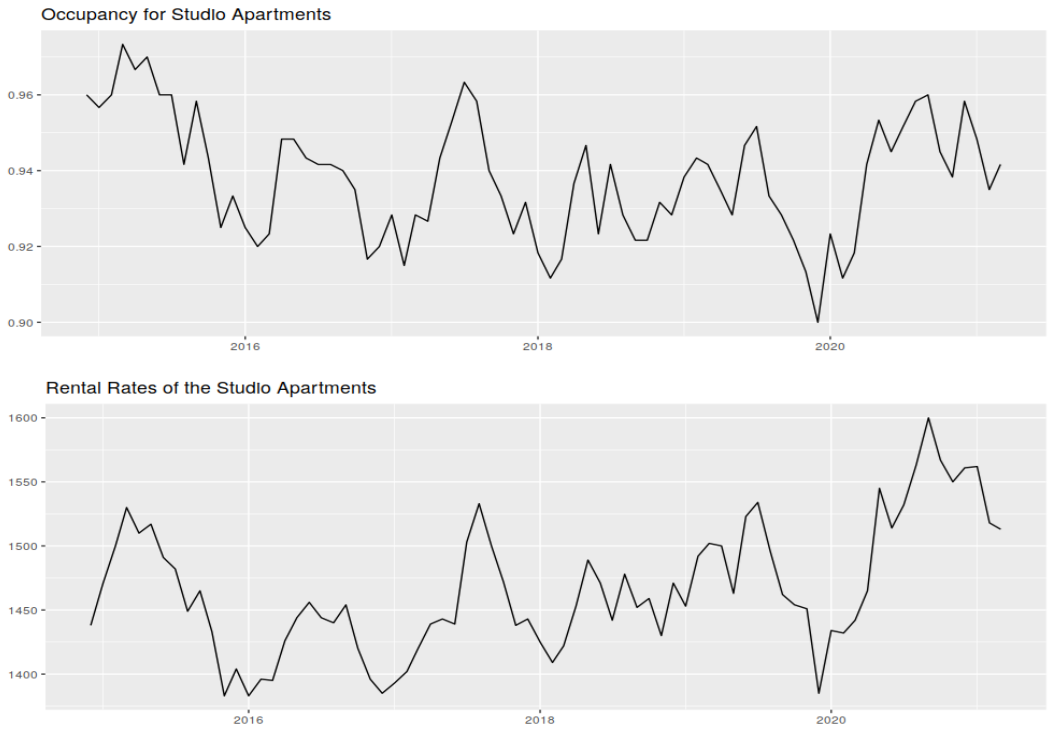

Figure 12 Occupancy and the rental rates of the std.

One observation is that similarity is much more prominent in later years than the earlier two years. And there seems to be a much prominent lag in earlier years. First observation can be checked visually by giving a different color to earlier years (Figure 11). As can be seen in the Figure 11, earlier years are gathered at the lower right side of the plot, which gives evidence in favor of the claim. There can be two reasons that I can think of. One explanation is that, in earlier years, information of occupancy was reaching the pricing decision makers later. Another explanation could be that there were other variables which

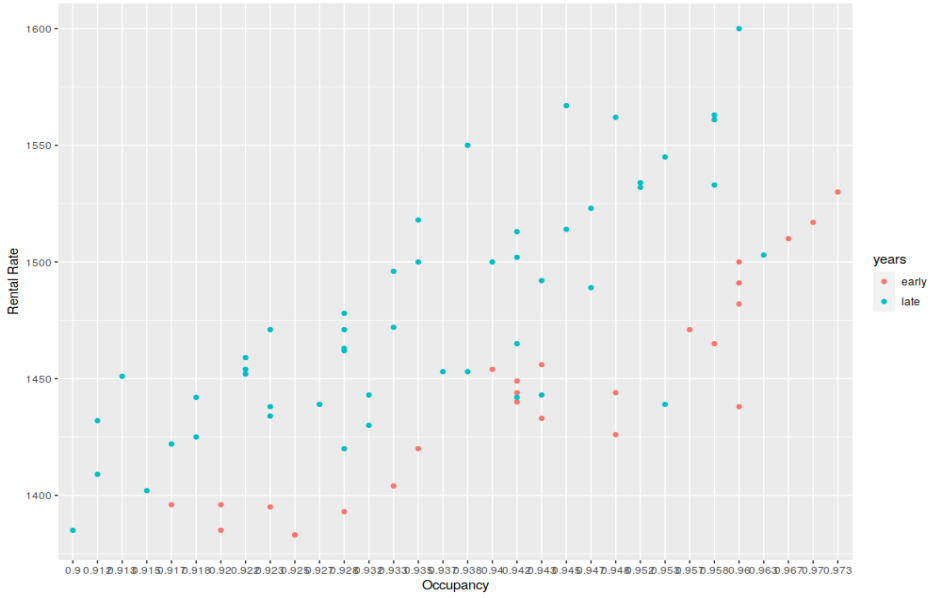

Figure 11 Scatter plot of occupancy versus rental rate divided by the early and later years.

were being taken into consideration in earlier years, which was found to be not as useful as occupancy later on. Therefore, more weight was given to occupancy in the following years.

In addition, as can be seen in the Figure 13 occupancy has a more symmetric distribution while rental rate is more right skewed. It raises the question that, if we had utilized occupancy variable better, would we have a more normal or left skewed rental rate distribution?

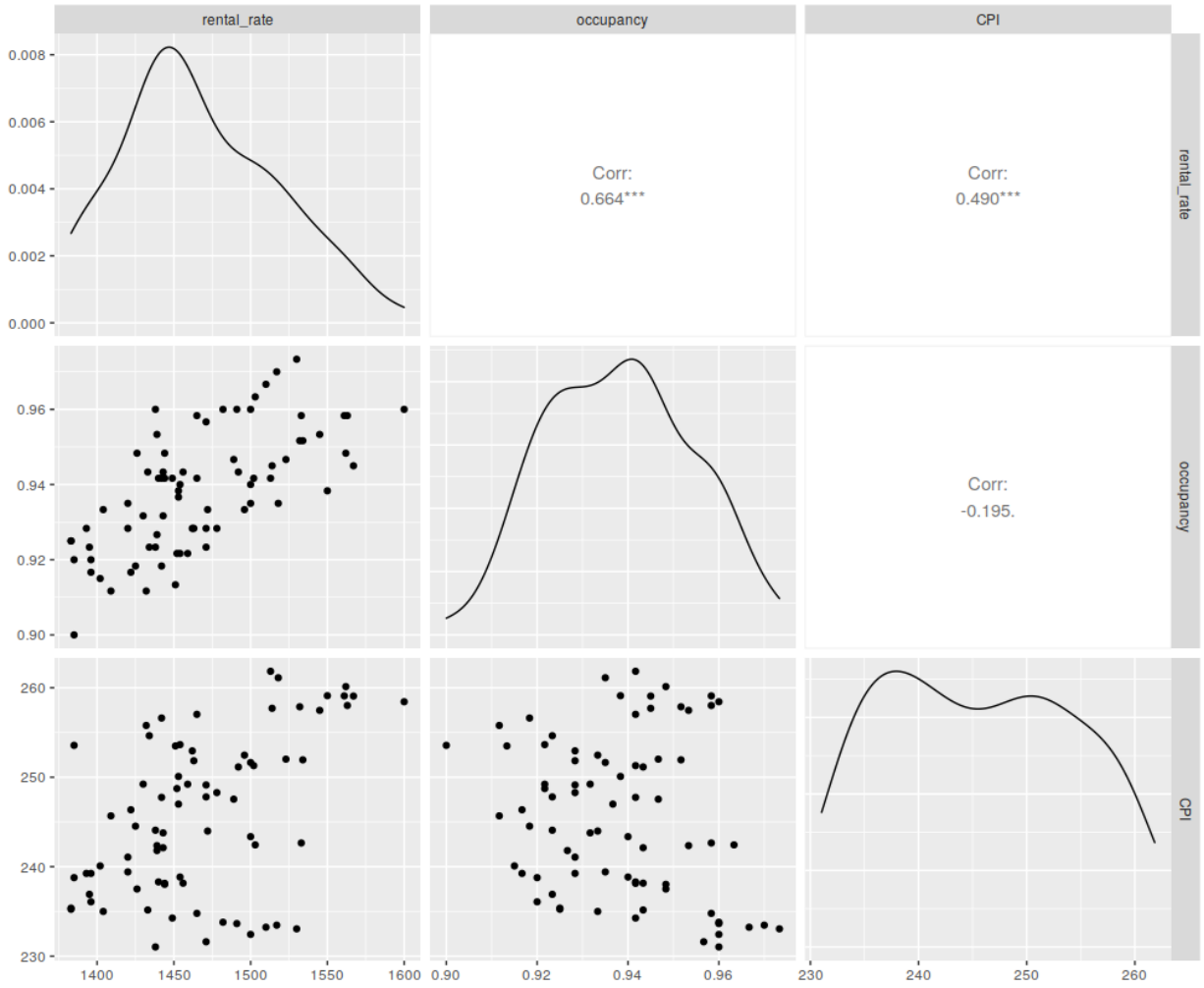

Figure 13 Relations between occupancy, CPI and rental rates.

You can find the yearly occupancy values for the studio apartments in the following table. We have to be careful about comparisons in this table, because number of months at each year differs. There are only 10 months in 2012 and 3 months in 2021. Therefore we should use the adjusted rate values seen in the second row for the comparisons.

	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021
Occupancy	1022	3507	5795	7155	6972	7015	6964	6975	7058	1746
Rate	102.2	292.25	482.9167	596.25	581	584.5833	580.33	581.25	588.1667	582

Customer Duration:

I also calculated the customer durations. As can be seen in the Figure 13, customers tend to stay in the multiples of 12 months. This phenomenon occurs because leases are renewed every year.

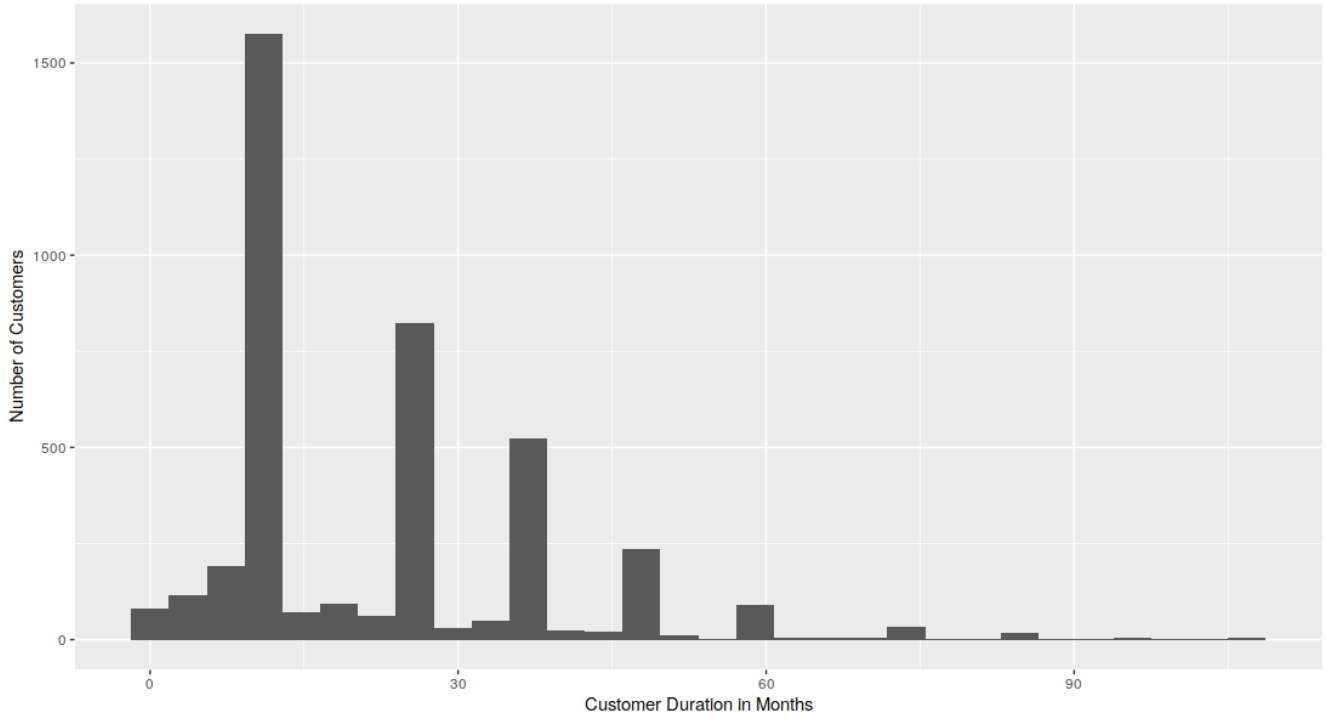

Figure 14 Customer Duration in months.

CPI:

Economic conditions of the region also affect pricing. If the prices of the products are increasing overall, then we would expect a rise in the rental rates too. To be able incorporate this information into the model, I used Consumer Price Index data from the website of U.S. Bureau of Labor Statistics. I used statistics for all urban consumers assuming that our rentals are in urban regions.

You can divide trend into two parts, one is coming from economics of the region, other is specific to our case, housing industry. It would be a wiser idea to use CPI for rentals in the region. Unfortunately, I didn't have this information.

Furthermore, CPI series is quite stable in our case with some small seasonal drops at the end of each year (Figure 15). However, if we were at a country with an unstable economy, this variable would be much more relevant. Prices would need to react to the fluctuations in the economy.

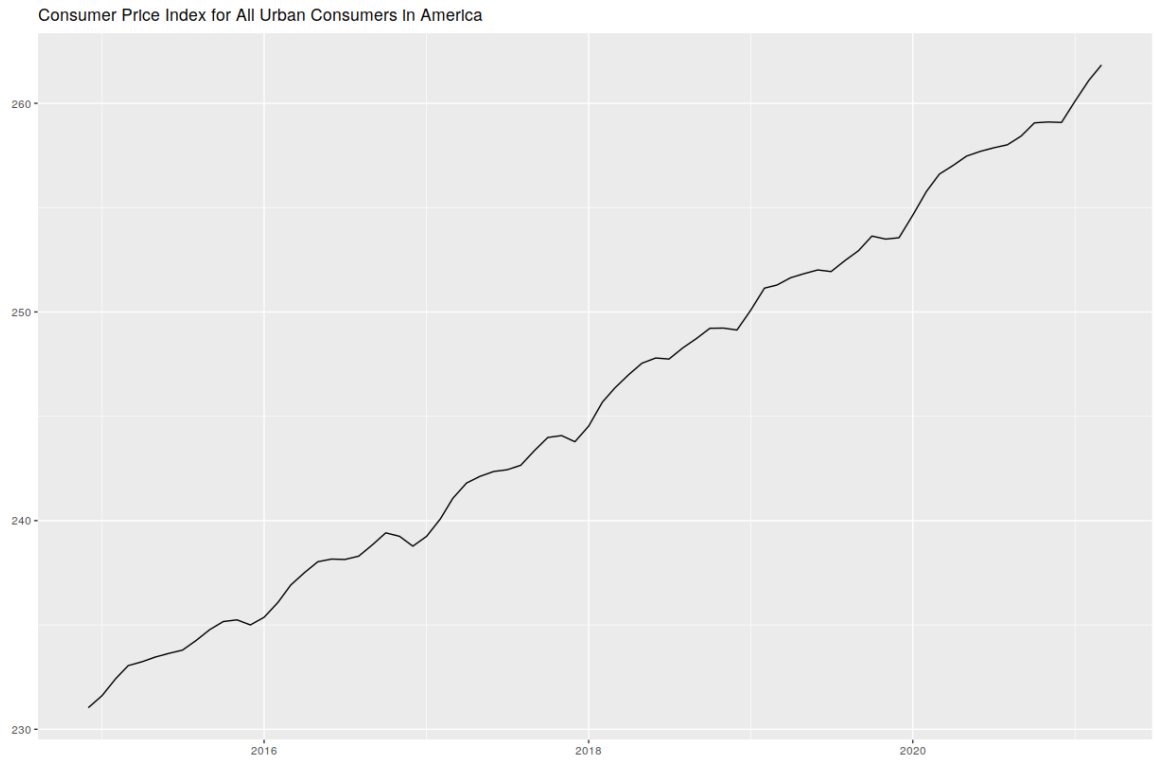

Figure 15 CPI for all urban customers in America.

MODELS TRAINED WITHOUT REGRESSORS

First I divided the data into training and test sets, I will use the last 4 points as test data.

Exponential smoothing techniques assume that the effect of previous lags decay exponentially. Some of the models I applied are (Figure 16):

Holt's Linear Trend Method:

It forecasts utilizing the estimated trend of the data at given point in an additive fashion. However, it doesn't take seasonality into account. As can be seen in Figure 18, just follows the series from one step back. Therefore it will produce one single forecast for the all future points.

Holt's Damped Trend Method:

Since Holt's method assigns a constant trend into the future forecasts, it tends to over-forecast values. This method dampens this trend to fix this issue. Again, no seasonality is considered.

Holt- Winter's Seasonal Method (Additive and multiplicative):

This is a generalization of the earlier method with an addition of the seasonal component. Seasonal component can be additive or multiplicative. The additive method is preferred when the seasonal variations are roughly constant through the series, while the multiplicative method is preferred when the seasonal variations are changing proportional to the level of the series. There are also variations of this method with and without damping. As can be seen in Figure 17, this methods takes both seasonality and the trend into account and therefore fits better.

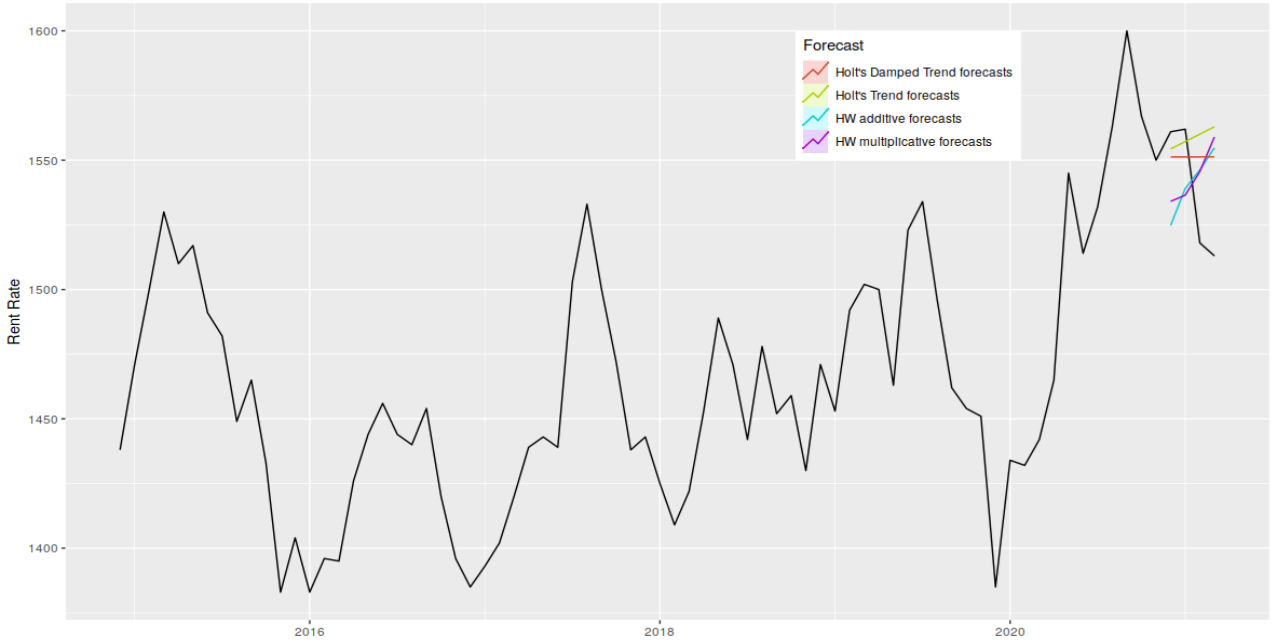

Figure 16 Forecasts from exponential smoothing methods.

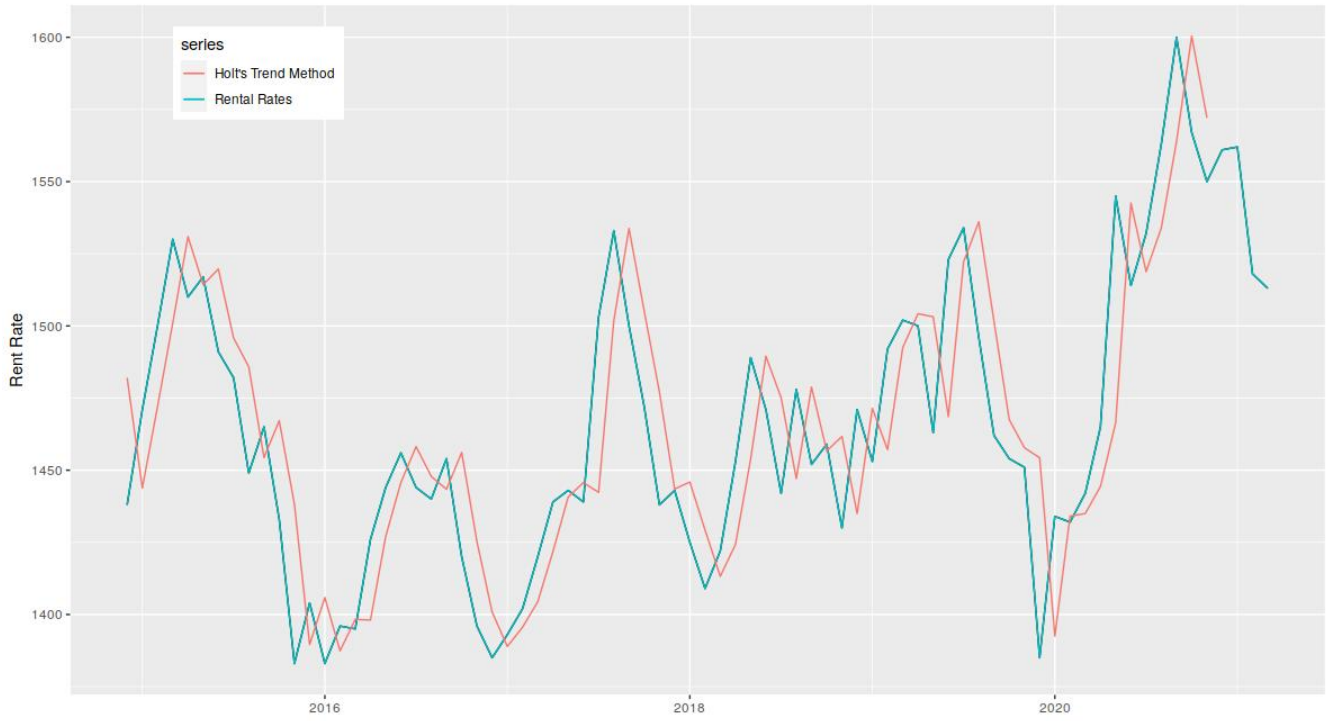

Figure 18 Fitting of Holt's Trend method.

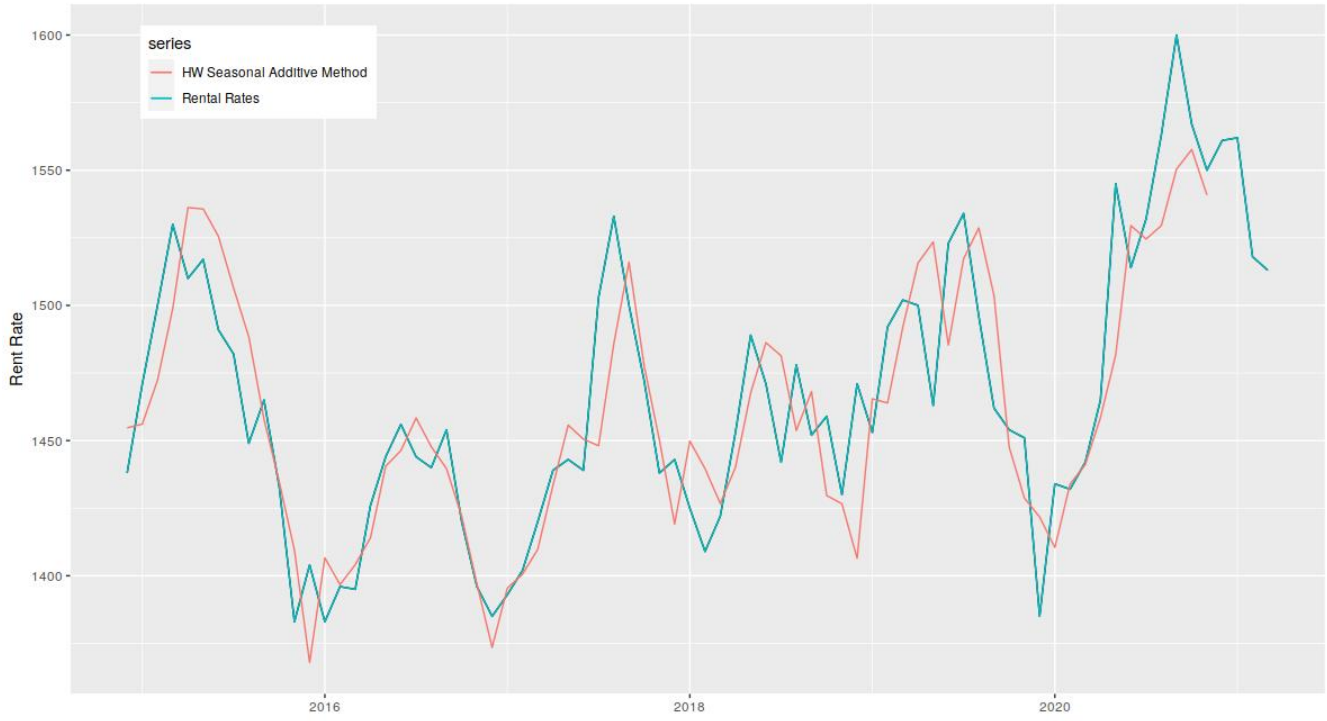

Figure 17 Fitting of Holt Winters additive seasonal method.

SARIMA (Seasonal Autoregressive Integrated Moving Average Method):

ARIMA models utilize the correlation between lags as well as the error coming from past forecast errors. Since we have seasonality in our data, we will apply seasonal version of ARIMA models, SARIMA(p,d,q)(P,D,Q)$_m$. To be able to apply this method, we need stationarity of the series. Our studio apartment series is not stationary as discussed before; we have to take first differences. This gives us the parameter d=1. Because of the seasonality, we could also have to take seasonal differences t_{12}-t_1 , D=1. But since our data is already stationary after first differencing there is no need for this. p variable determines how many lags we consider putting into the model (autoregression part). q variable determines how many errors back we will go in the model (moving averages part). P and Q are the seasonal counterparts of p and q. We train many models with changing hyper parameters p, q, P, D, Q and choose the model which gives the smallest BIC value (Figure 19). Although, our model has seasonality, simple ARIMA (0,1,0) gave the best BIC value. It means that a random walk where our current state is only affected by the earlier state gives the best result. However, to check if the model fits well, we need to check residuals. A Box-Ljung test on the residuals give a p value smaller than 0.05, which means residuals are not coming from white noise. Also, a significant ACF value at lag 5 suggests that autocorrelation exists in our residual series (Figure 20). Therefore, this model will not be used to fit the data further.

```
ARIMA(0,1,4)                           : 691.709
ARIMA(0,1,4)            with drift     : 695.831
ARIMA(0,1,4)(0,0,1)[12]                : 693.344
ARIMA(0,1,4)(0,0,1)[12] with drift     : 697.4637
ARIMA(0,1,4)(1,0,0)[12]                : 692.2382
ARIMA(0,1,4)(1,0,0)[12] with drift     : 696.3634
ARIMA(0,1,5)                           : 693.9006
ARIMA(0,1,5)            with drift     : Inf
ARIMA(1,1,0)                           : 688.071
ARIMA(1,1,0)            with drift     : 692.108
ARIMA(1,1,0)(0,0,1)[12]                : 690.4376
```

Figure 19 Some steps while searching for the optimal ARIMA model.

In addition to ARIMA, I also applied neural network models and TBATS. Same lack of fit problem raised. Therefore, I moved forward with adding regressors to my models.

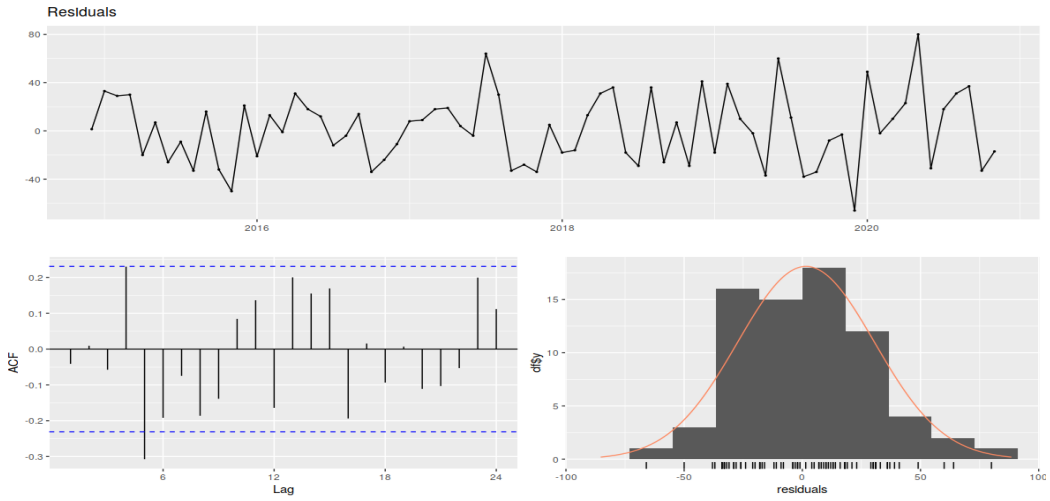

Figure 20 Residual checks for the ARIMA(0,1,0) model.

MODELS TRAINED WITH REGRESSORS

Dynamic Harmonic Regression:

In these models, seasonal pattern is modeled using Fourier terms. These Fourier terms are given to ARIMA models as regressors. The number of Fourier sin and cos pairs, K (controlling smoothness of the seasonal pattern) is a hyper parameter. I applied the model and found the following models and BIC values for different K values.

K=1: ARIMA(0,1,1)(1,0,0)[12] gaveRMSE=23.19812 and MAE= 18.29285.

K=2: ARIMA(0,1,1)(0,0,1)[12] gave RMSE= 20.67676 and MAE= 16.45959

After K=2, models showed lack of fit, therefore, I chose the second model to move forward.

ARIMA models with occupancy and CPI as regressors:

ARIMA(2,0,0) with occupancy and cpi, bic=677.56, RMSE = 22.33245, MAE = 18.35579

ARIMA(0,1,1) with occupancy and cpi, d is enforced, bic= 667.25, RMSE= 23.34762, MAE= 18.17315

ARIMA(0,1,1) with occupancy, bic = 664.37 , RMSE= 23.58078, MAE = 18.03673

Vector Autoregression Models:

In the earlier models, we assumed that regressors are affecting the series not vica versa. However, we have a bidirectional relationship, especially with occupancy. Occupancy affects the prices and prices affect the occupancy in return. Therefore, I thought it might be a good idea to apply vector auto regression models which takes this phenomenon into account. I applied the model with different hyperparameter values, p(number of lags). The ones which passed the lack of fit tests are the following:

VAR(4) MAE =40.08055

VAR(5) MAE = 40.96711

Since MAE values are significantly higher, I discarded these models.

FINAL MODEL INTERPRETATION

As the final model, I chose ARIMA(0,1,1) with the regressor occupancy. Residuals give a Ljung-Box test p- value of 0.1551, which tells that we don't have a lack of fit problem. Also, normality and the non-existence of autocorrelation can been seen in Figure 22. Although dynamic harmonic regression model with K=2 gave smaller accuracy values, we should prefer the model with an external regressor. Let's say we made a bad forecast at some point, external variable coming next month can help us to get back on track. As can be seen in the forecasting plot (Figure 21), it also forecasts well with a test MAE error 17.90125.

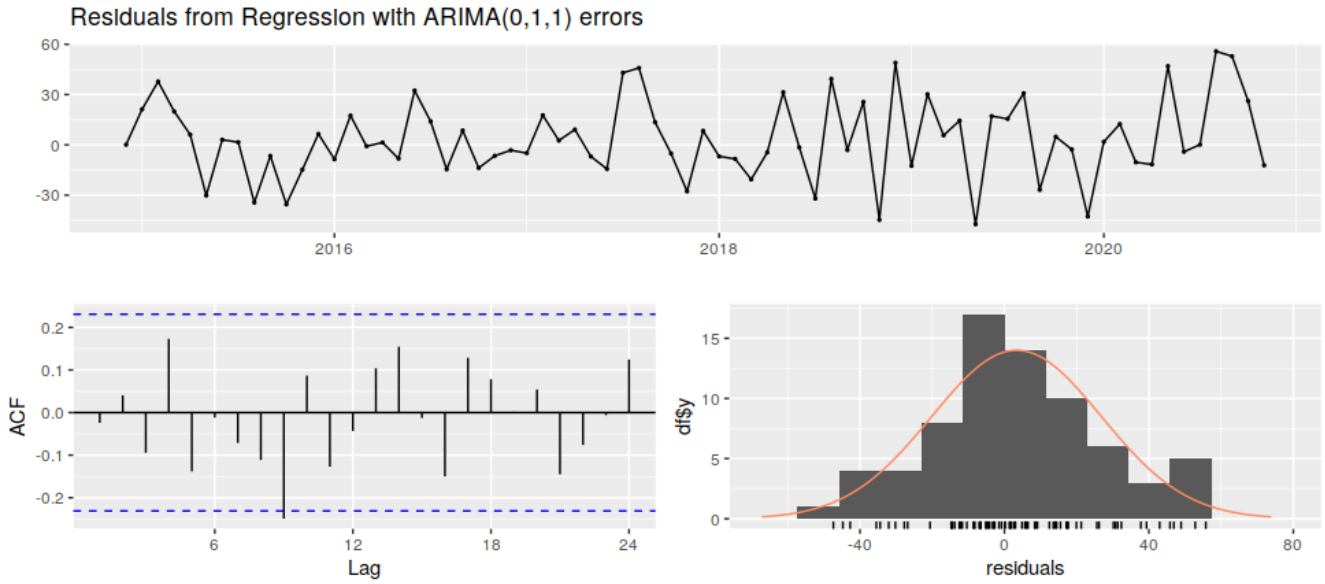

Figure 22 Residual checks for the final model.

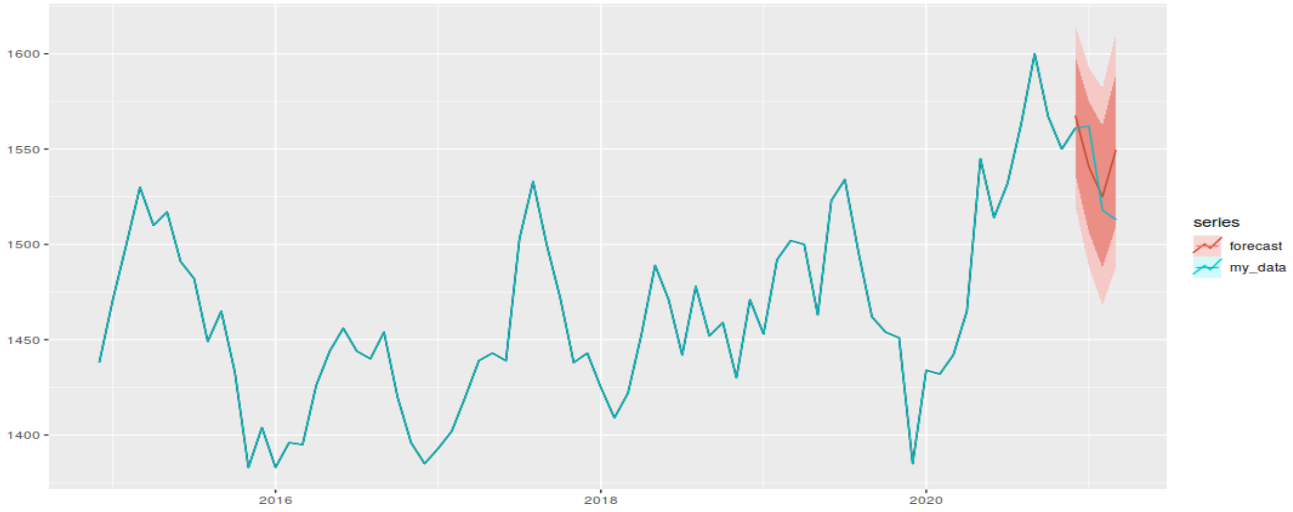

Figure 21 Forecast using the ARIMA(0,1,1) model with occupancy as regressor.

FUTURE IMPROVEMENTS

Here are some suggestion for future improvements.

- We can use lagged regressors instead of only the regressors themselves since the effect can be lagged. For example, rent rate of today might be affected by occupancy of yesterday instead of today.
- To avoid overfitting we can apply cross validation on the training set. But when we have a short time series, it might make us lose importamt information and make the forecasts meaningless.
- Instead of using only one model, we can use a combination of them. Utilizing the wisdom of the crowd might help us with both overfitting and biasedness.
- External variables are important, when we have to decide between the models with and without external variables where accuracy values are similar, we might choose the one with the external variables. Let's say we made a bad forecast at some point, external variable coming next month can help us to get back on track.
- Instead of segregating the data at only UnitType level, we can segregate according to the sensitivity of the customers. This might help us to not lose sensitive customers, and gain more profit from the customers with low sensitivity.
- Outlier detection is important. Model should be able to detect sudden changes so that we can gather more information. Is a park or metro station added to the neighborhood? Or there was a sudden drop in the occupancy because of a natural disaster?
- We can apply non-linear time series models such as STAR, ESTAR or LSTAR.
- When we don't have enough data, we can use bootstrapping and bagging to get better results.