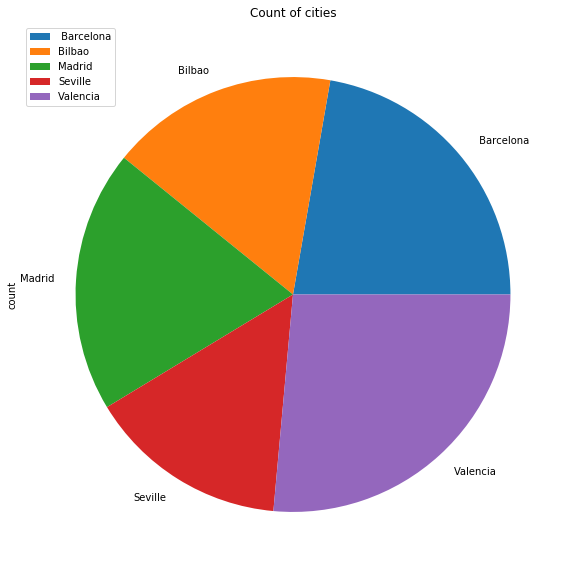
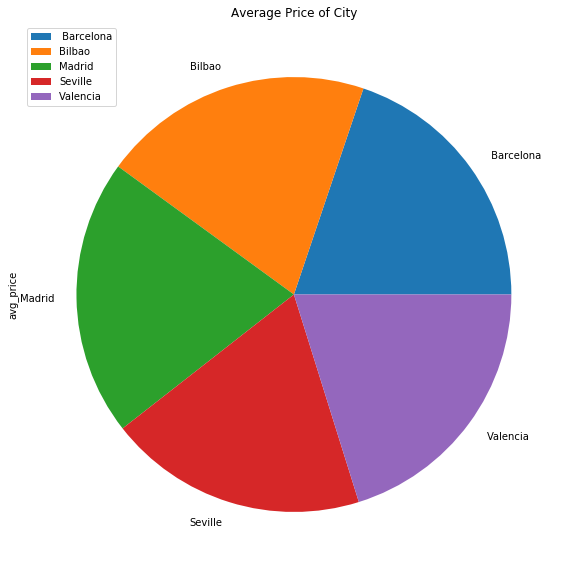
Wes Final Iteration Summary

I did an exploration of the data using Pyspark and Jupyter Notebooks

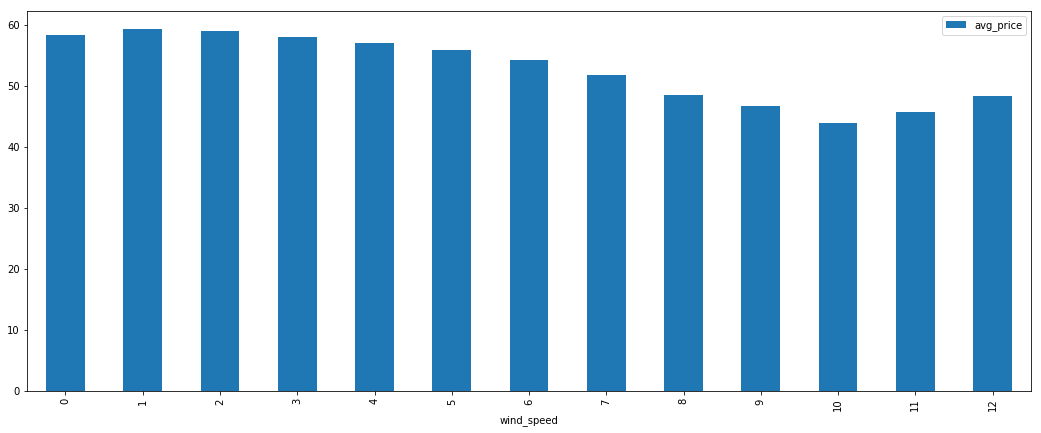
Visualization of the count of cities, to show that each of them are well represented within the dataset. Valencia was the most represented and Seville being the least represented



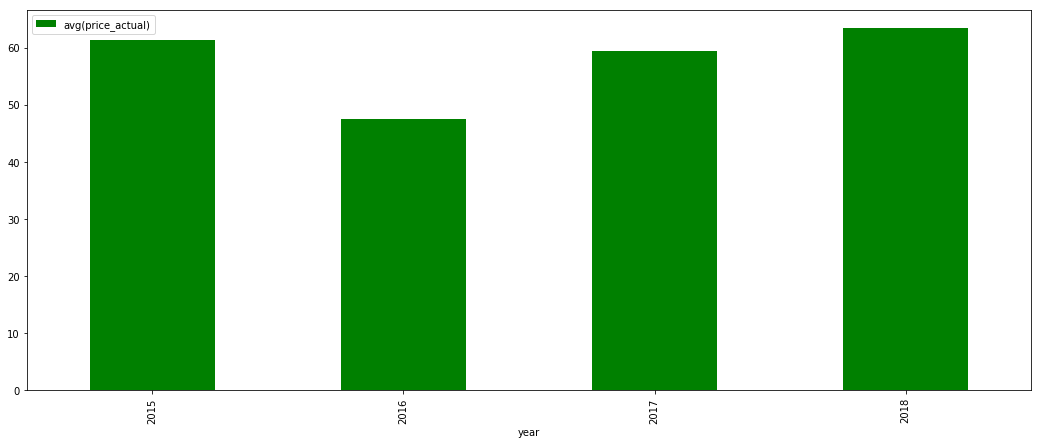


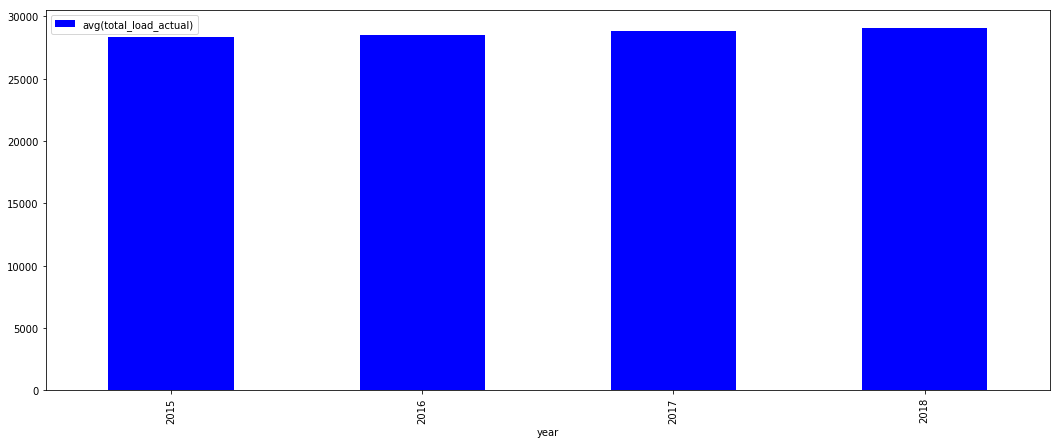
The average price between each city is nearly identical as well. This is something we had hoped to see, as it means the pricing is fair between each city and no one customer is paying more than another just for living in a different city

Exploring more relationships in the data I looked at Wind Speeds and was surprised to a semi-strong correlation between that and price. I don’t have the domain knowledge of the field to know if this is expected behavior or not. It should be noted that higher wind speeds have less data associated with them so those should not be seen as strongly correlated

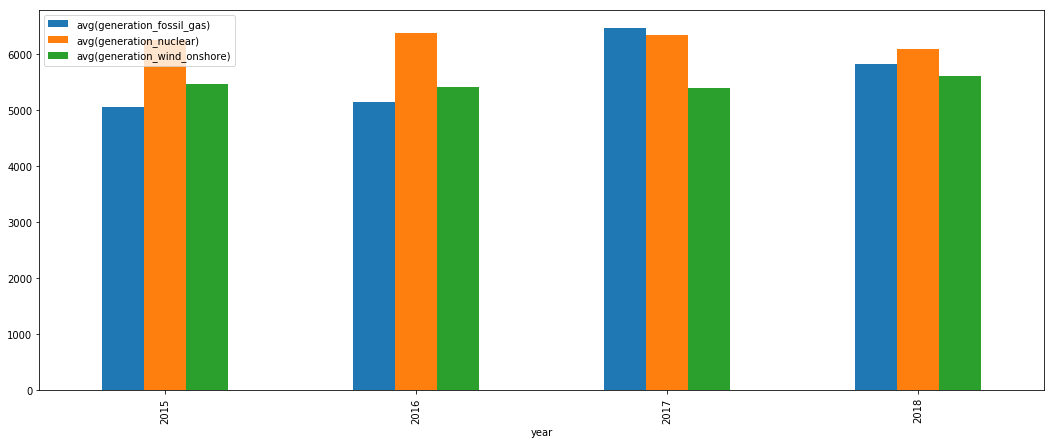


Looking year by year 2016 was a noticeable low point for average price while the load on the system stayed very consistent

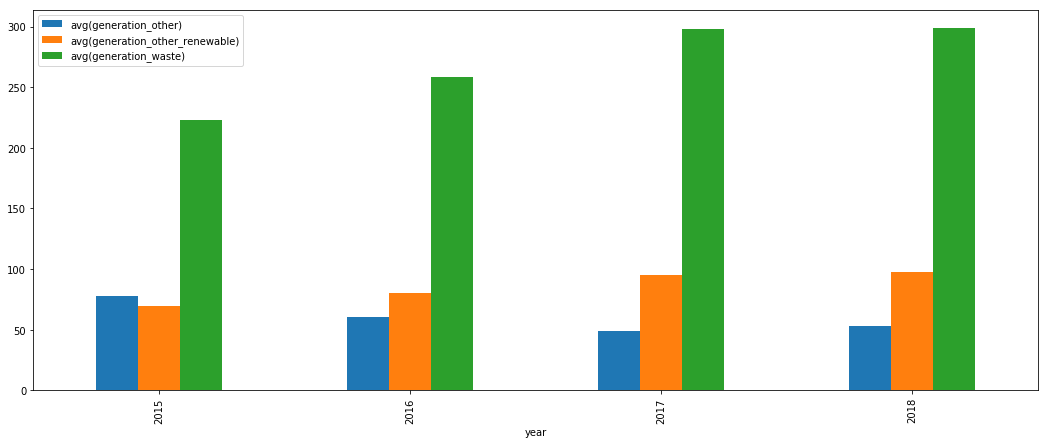




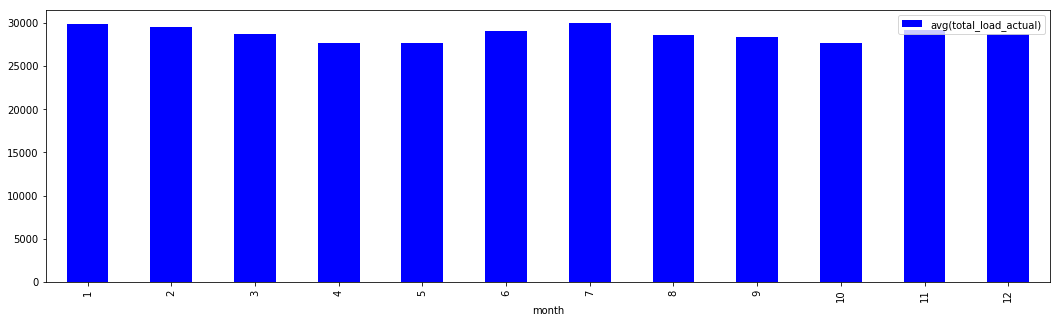
And while the total load on the system remained consistent, the individual energy generations have varied. These are the top three Generations (measured in MegaWatts) and how they varied through the years

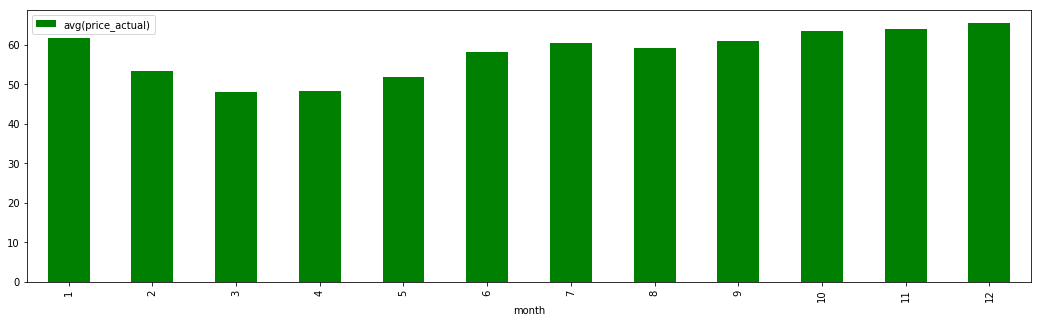


The three lowest (that still follow trends and are not 0) are

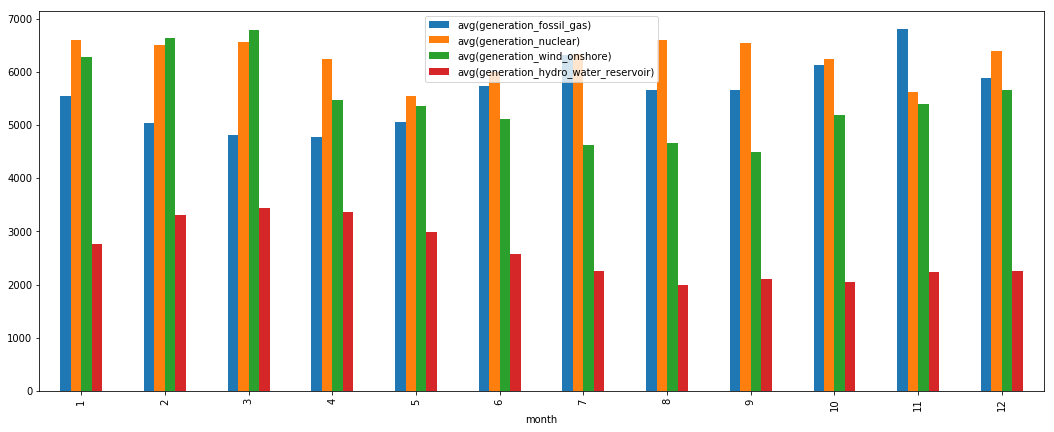


Looking at trends through the different months of the year we can see that the total load is much more consistent than the average price. We can also see that average price is less expensive in the warm months and more expensive in the cold months, as we expect

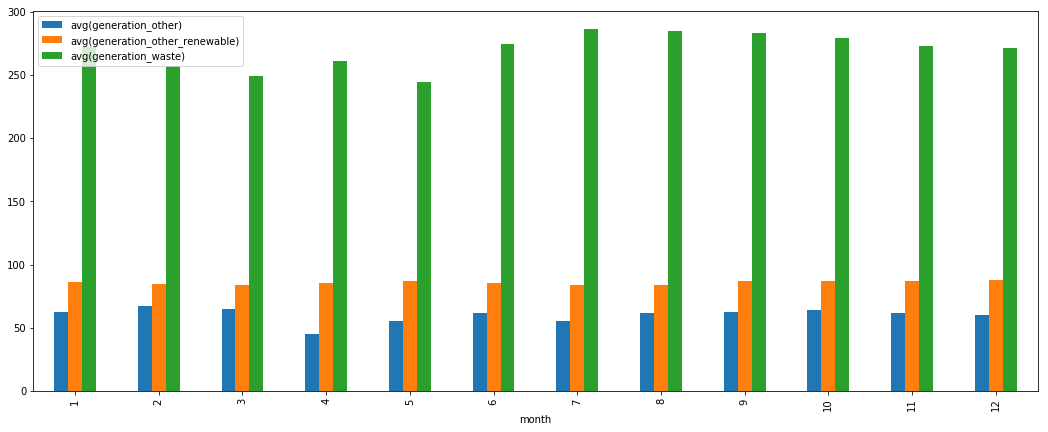




High generation trends by month

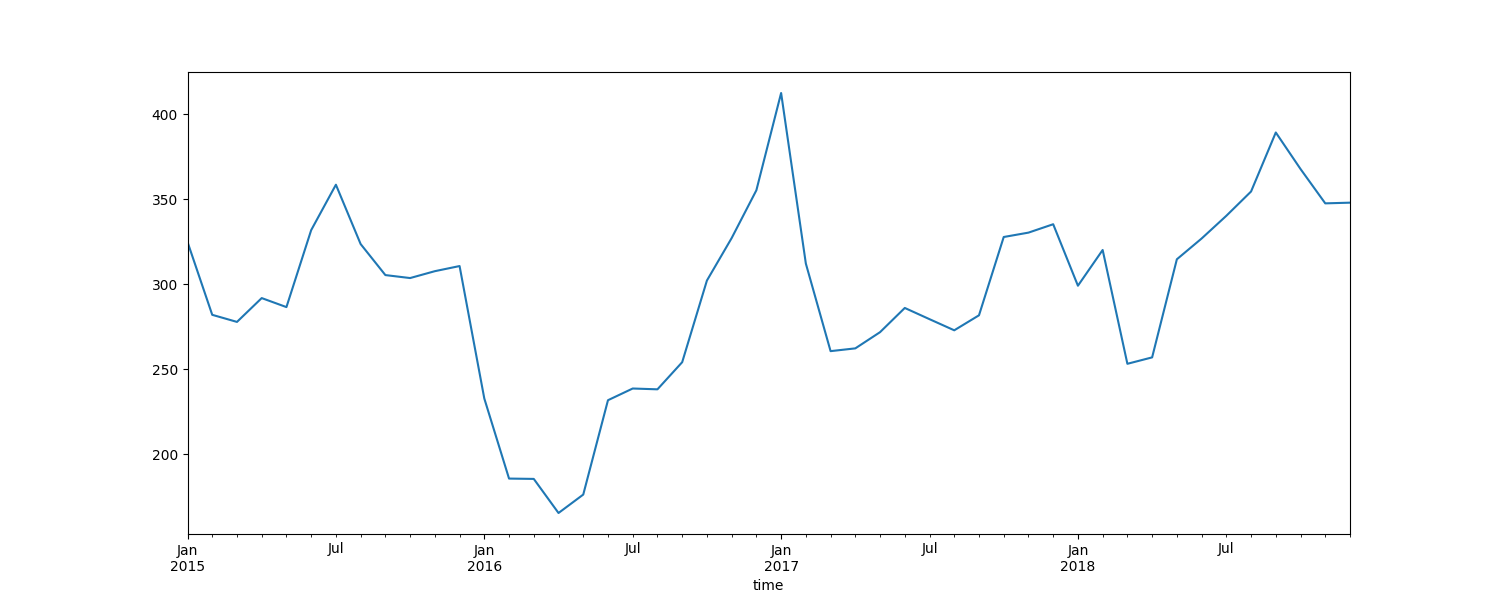


Low generation trends by month

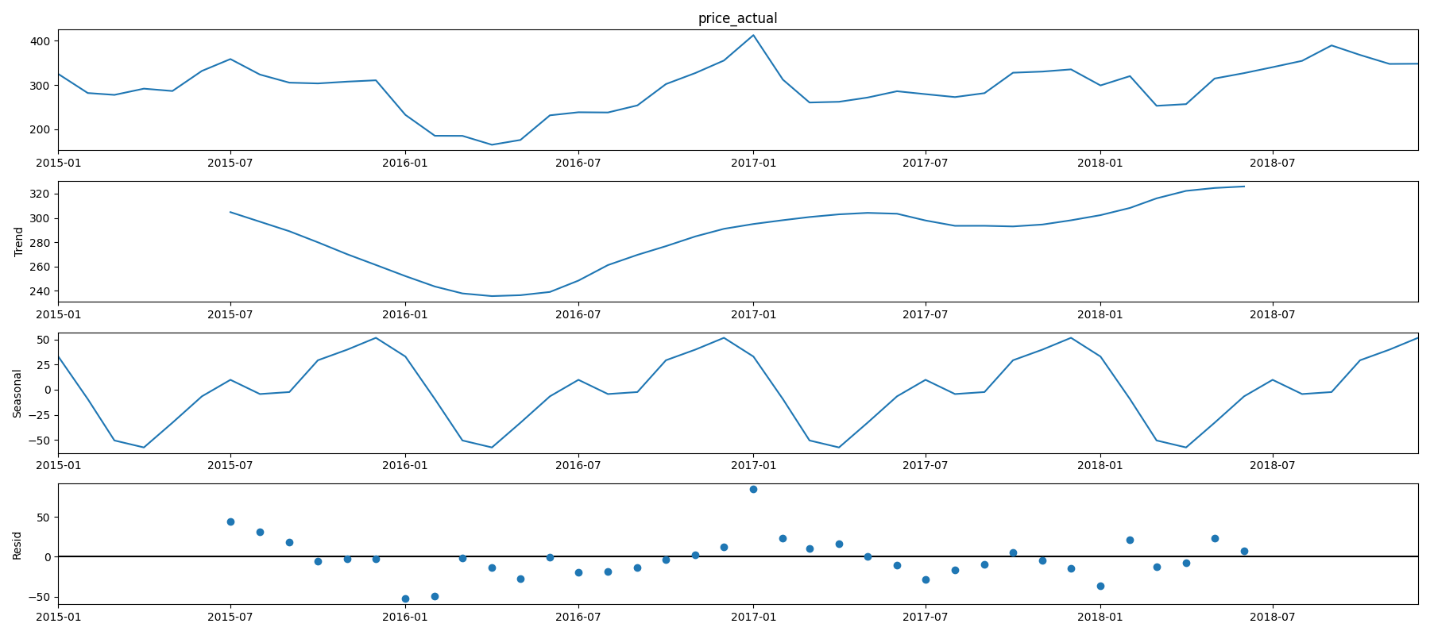


I also did some Time Series analysis using <https://towardsdatascience.com/an-end-to-end-project-on-time-series-analysis-and-forecasting-with-python-4835e6bf050b> as a guide. I wanted to try and implement some simple machine learning / statistical learning techniques.

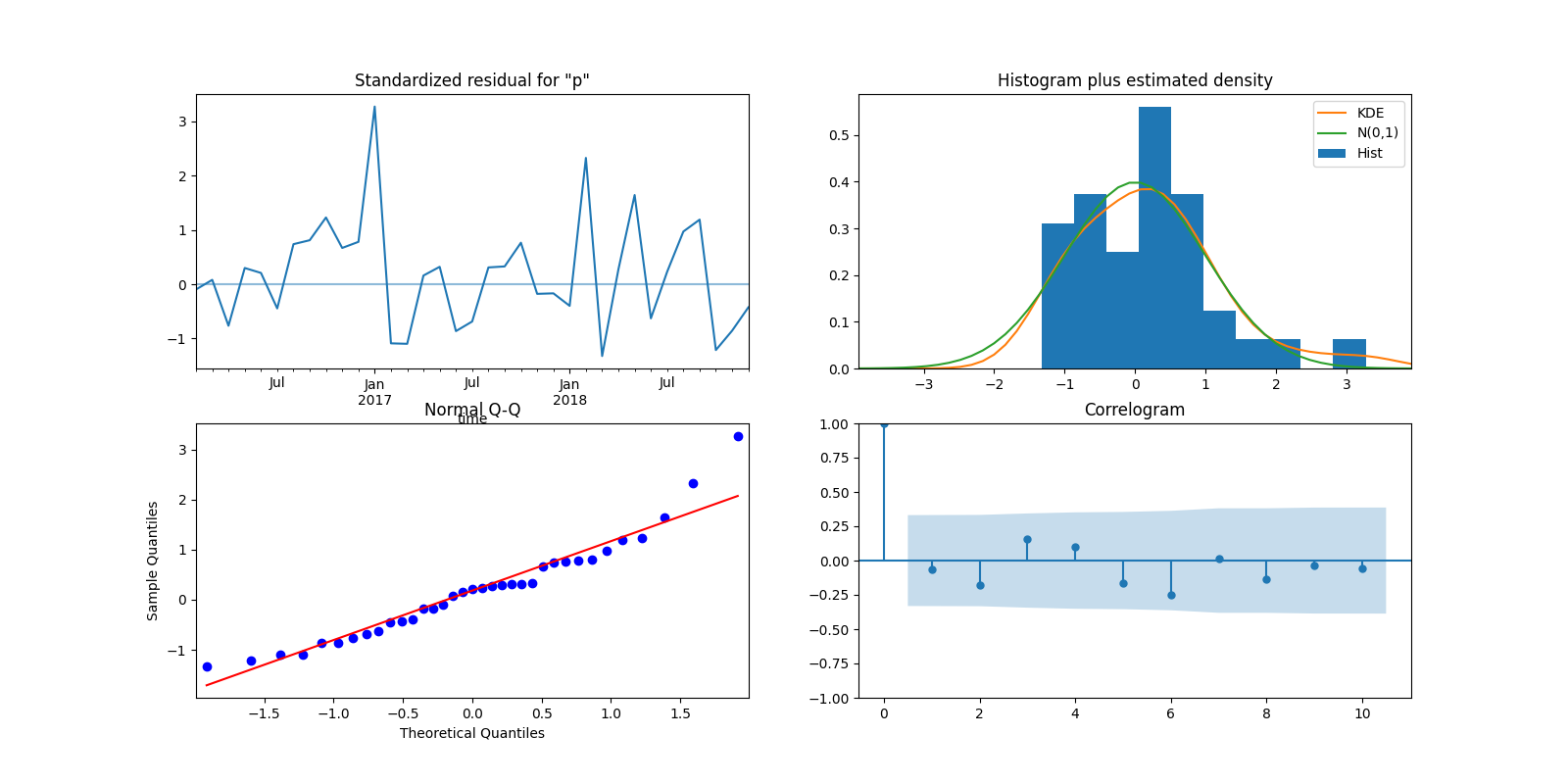
Looking at the actual price data broken down by months, it follows a general trend line like this from 2015-2018



Looking at the Decomposition analysis we can look at the Trend, Seasonals and Residuals. The Trend is essentially the trend line as a quadratic function, simplifying the flow of the data and showing us the general highs, lows and path it traveled. Seasonal is the variation we can expect for a given season, which is why it follows a consistent pattern. Residuals are the timeseries data subtracted by the Trend and the Seasonal data, which we want as close to 0 as possible. The farther away from 0, the more exceptional the data is to be seen at that point.

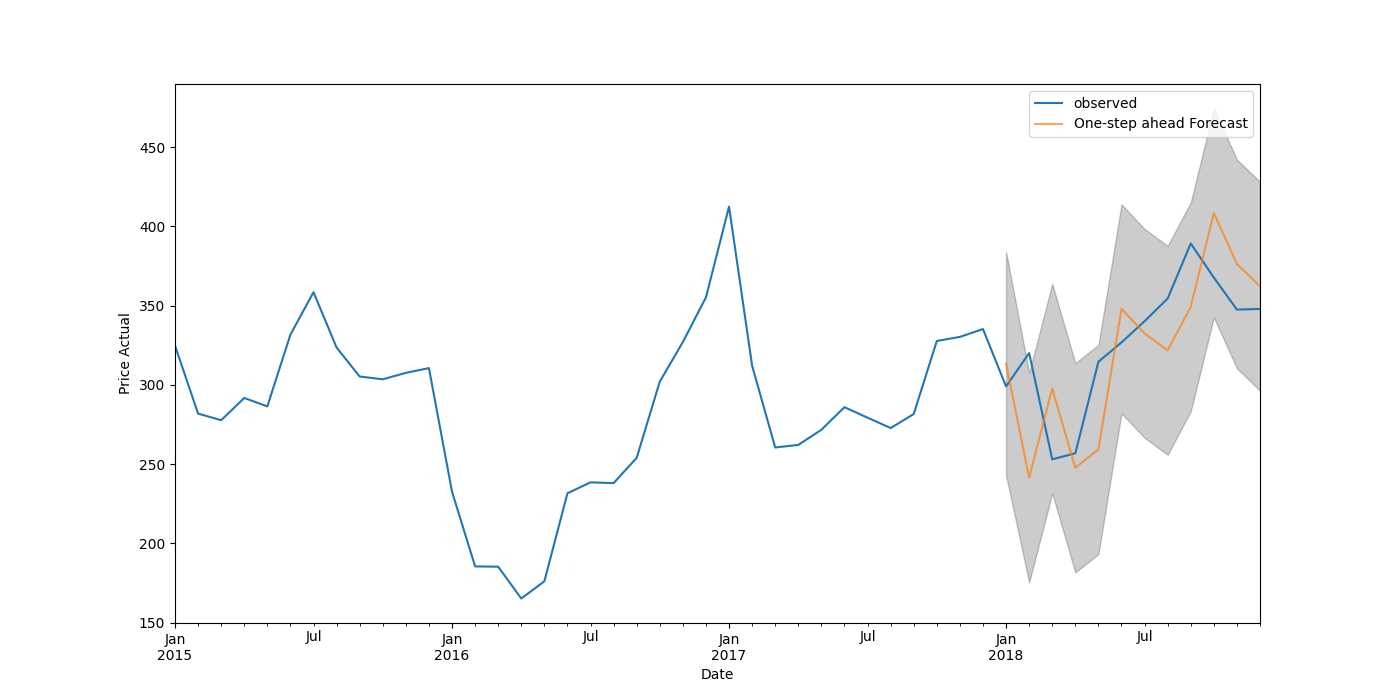


We can see that high point in 2017 was unexpected data, having the highest residual

ARIMA is a commonly used method for time-series analysis. It takes in parameters for the seasonlity, trend and noise within the data. You can decide what values to this by doing an exhaustive search and comparing the AIC and choosing the lowest possible value. Following that methodology and running the diagnostics we get

The data isn’t perfect, as any real world data won’t be, but we can through the Normal Q-Q, Histogram, and Correlogram that the data is normally distributed enough to run some predictive statistics on it

This is a rolling forecast prediction, using the current data to try and predict the tail end of the current data. It’s not perfect but we can see we are lying in the same general range of the expected price



Using the same model we built before but trying to predict the next few years of data, we can see a general trend and confidence interval that we can expect the actual price to lie in between 2019 and 2020

