<https://github.com/facebook/prophet/issues/549>

<https://jolars.github.io/TSAsolutions/index.html>

<https://machinelearningmastery.com/probabilistic-model-selection-measures/>

<https://www.kaggle.com/nholloway/stationarity-smoothing-and-seasonality>

<https://www.epa.gov/sites/production/files/2016-04/documents/2012_aqi_factsheet.pdf>

<https://www.epa.gov/criteria-air-pollutants/naaqs-table>

<http://www.stateair.net/web/post/1/1.HTML>

<https://www.transportpolicy.net/standard/china-air-quality-standards/>

<https://peerj.com/articles/9961/>

<https://github.com/advaitsave/Introduction-to-Time-Series-forecasting-Python/blob/master/Time%20Series%20in%20Python.ipynb>

<https://rstudio-pubs-static.s3.amazonaws.com/523169_3cb207e72e6f41e8a56dfd083c3fcc7e.html>

<https://www.nielsenmark.us/2018/02/21/forecasting-pm2-5-with-forecast-and-prophet/>

<https://www.kaggle.com/robikscube/time-series-forecasting-with-prophet#Simple-Prophet-Model>

<https://nextjournal.com/fb-prophet/facebook-prophet-diagnostics>

<https://datascience.stackexchange.com/questions/69277/how-fbprophet-cross-validation-works>

<https://www.kaggle.com/alexkaggle95/stock-prices-forecast-plotly-prophet>

<https://facebook.github.io/prophet/docs/diagnostics.html>

<https://docs.exploratory.io/analytics/forecasting>

<https://github.com/taflor/CWB-Pollution-Timeseries/blob/main/notebooks/4.0%20CWB%20Modeling.ipynb>

<https://github.com/taeokimeng/pm2.5-forecast/blob/master/forecast.ipynb>

<https://github.com/Nidhi-Rai-Programmer/Forecasting-PM2.5-Level/blob/master/Final_Code_PM25.ipynb>

<https://github.com/LuisErnestoColchado/Learning-Deep-Pollution/blob/master/LSTM_LSTME_XGBOOST.ipynb>

<https://github.com/Dieselmarble/Beijing-PM2.5-Forecasting/blob/master/Latex%26PDF/machine-learning.pdf>

<https://blissair.com/what-is-pm-2-5.htm>

<https://www.kaggle.com/ulrich07/parallel-linear-regression-silver-medal-v1>

<https://www.kaggle.com/shangweichen/pytorch-osic-multiple-quantile-regression-starter>

<https://www.kaggle.com/jagadish13/osic-multiple-quantile-regression-eda>

<https://www.youtube.com/watch?v=JvIzB3hULCo>

<https://www.youtube.com/watch?v=Aw77aMLj9uM&t=882s>

<https://eranraviv.com/quantile-autoregression-in-r/>

<https://www.statsmodels.org/devel/examples/notebooks/generated/quantile_regression.html>

<https://pawarbi.github.io/blog/forecasting/r/python/rpy2/altair/fbprophet/ensemble_forecast/uncertainty/simulation/2020/04/21/timeseries-part2.html#Auto-regression,-AR(p)>:

<https://tanthiamhuat.files.wordpress.com/2015/12/step-by-step-guide-to-forecasting-through-arima-modeling.pdf>

<https://pawarbi.github.io/blog/forecasting/r/python/rpy2/altair/fbprophet/ensemble_forecast/uncertainty/simulation/2020/04/21/timeseries-part2.html>

<https://www.statsmodels.org/devel/examples/notebooks/generated/quantile_regression.html>

The publication referenced there or the monograph by one of the authors Roger Koenker "Quantile Regression" ISBN-13: 978-0521845731 is the ultimate reference on this model, as far as I know.  But I think the two vignettes for the R package that I am attaching (you can get them if you install 'quantreg' in R and then use e.g.  'vignette("rq")' at the command line) are a good, concise source about what quantile regression is.

<https://www.kaggle.com/allunia/m5-sales-uncertainty-prediction>

<https://www.kaggle.com/gloriousc/madrid-s-air-quality-with-arima-forecasting>

<https://www.kaggle.com/dhimananubhav/forecasting-ozone-levels-in-madrid>

<https://plotly.com/python/time-series/>

<https://aqicn.org/faq/2013-09-09/revised-pm25-aqi-breakpoints/>

<https://www.epa.gov/sites/production/files/2020-04/documents/fact_sheet_pm_naaqs_proposal.pdf>

<https://19january2017snapshot.epa.gov/international-cooperation/epa-collaboration-china_.html>

<https://otexts.com/fpp2/residuals.html>

<https://campus.datacamp.com/courses/garch-models-in-python/model-performance-evaluation?ex=8>

<https://towardsdatascience.com/identifying-ar-and-ma-terms-using-acf-and-pacf-plots-in-time-series-forecasting-ccb9fd073db8>

<https://www.statisticshowto.com/ljung-box-test/>

<https://stats.stackexchange.com/questions/64711/ljung-box-statistics-for-arima-residuals-in-r-confusing-test-results>

<https://github.com/Alro10/deep-learning-time-series/blob/master/notebooks/SARIMA.ipynb>

<https://github.com/AileenNielsen/TimeSeriesAnalysisWithPython>

<https://www.kaggle.com/kashnitsky/topic-9-part-1-time-series-analysis-in-python>

<https://towardsdatascience.com/time-series-forecasting-with-a-sarima-model-db051b7ae459>

<https://www.machinelearningplus.com/time-series/kpss-test-for-stationarity/>

<https://www.kaggle.com/sumi25/understand-arima-and-tune-p-d-q>

<https://www.kaggle.com/kmaciver/using-holtwinter-and-arima-models>

<https://www.kaggle.com/berhag/co2-emission-forecast-with-python-seasonal-arima#5)-Natural-gas-CO2-emission-analysis>

<https://www.kaggle.com/freespirit08/time-series-for-beginners-with-arima>

<https://otexts.com/fpp2/complexseasonality.html>

<https://www.kaggle.com/msripooja/hourly-energy-consumption-time-series-rnn-lstm>

<https://www.kaggle.com/thebrownviking20/everything-you-can-do-with-a-time-series>

<https://www.kaggle.com/robinteuwens/forecasting-energy-demand>

<https://datascienceplus.com/time-series-analysis-using-arima-model-in-r/>

<https://www.kaggle.com/raenish/time-series-on-air-quality>

<https://github.com/ageron/handson-ml2/blob/master/16_nlp_with_rnns_and_attention.ipynb>

<https://stackoverflow.com/questions/63691083/pane-layout-in-r-studio-showing-fully-all-four-panes-fully>

We executed both first and seasonal differencing and ran their unit root tests. For both types of differencing, we got similar ADF results from that of the original time series. For the KPSS test we got more significant results with the first difference. **Figure 8** depicts the rolling statistics and autocorrelation plots of the first difference and **Figure 9** depicts that of the seasonal difference. We saw that the rolling statistics for the first difference was more constant about zero than that of the seasonal difference. Subsequently, we achieve a stationary time series after differencing once.

*4.1 Model Selection*

We used various methods to determine the best model. First, we looked at the first differenced auto correlation plot in Figure 8 to determine our time series parameters. The PACF Using the first differenced time series, we

|  |  |
| --- | --- |
| Model | AIC |
| Assumetion: ARIMA(4,1,3) | 15976.804 |
| Auto: ARIMA(0,1,4) | 15975.522 |
| Grid: ARIMA(3,1,4) | 15970.661 |
| Grid: SARIMA(2,1,1)(0,1,1)[12] | 15911.823 |
| Fb: Prophet(5) | 12250.355 |

*5.2 Error Metrics*

We evaluated seven error metrics. R-squared is the proportion of the variance explained by the model. Mean absolute error is the average of the absolute difference between the predicted value yhat and the mean value ybar. Median absolute error is similar except that is used the median value of y instead of the mean. Therefore, it is invariant to outliers.

For the rest for the metrics, we defined errors to be the actual values minus the predicted values. Mean squared error (MSE) is the average of the squared difference of the errors, measuring the variance of the residuals. Mean squared log error (MSLE) measures the relative difference of the errors. This means that it will treat small differences approximately the same as big differences, therefore penalizes underestimates more than overestimates. Mean absolute percentage error (MAPE) is the average of the absolute percentage of the errors. And lastly, root mean square error (RMSE) is the standard deviation of the residuals.

**Table 3. Prophet Model’s Error Metrics**

|  |  |
| --- | --- |
| Metric | Score |
| R-squared | 0.13 |
| Mean squared error (MSE) | 4350.59 |
| Mean absolute percentage error (MAPE) | 136.19 |