Analyzing Time Series of Hog and Soy Futures

Ruoting Wang

In financial informatics, analyzing stochastic processes is a useful aspect of data analysis in forecasting. A time series is an example of such a stochastic process. The definition of a time series is simply an ordered stochastic sequence of values spaced at even intervals [1]. The properties of a time series allows for descriptive and predictive analysis to be done on it. In this project, a univariate time series of Hog Futures will be analyzed, and a model will be created and back-tested to determine if it can be used in future trading strategies.

First, a visualization of the time series is necessary:

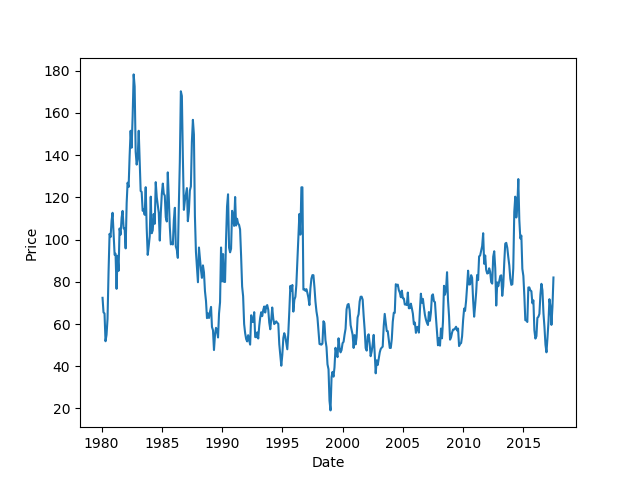
****

Figure 1. Hog Futures from 1980 to 2017.

The three aspects of a time series are trend, seasonality, and noise. An example of trend and seasonality would be seen in Figure 2. The sales data is trending upward with seasonality at the end of every year. Since each data point is a variable xt, each has component of independent variance with.

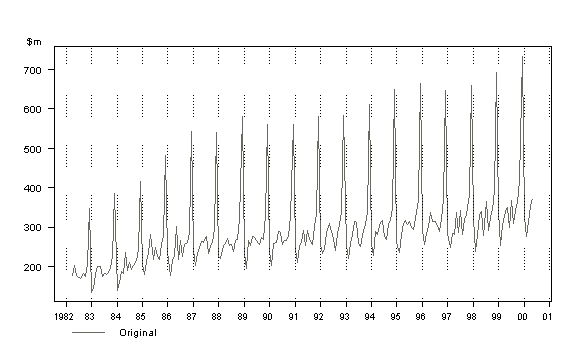
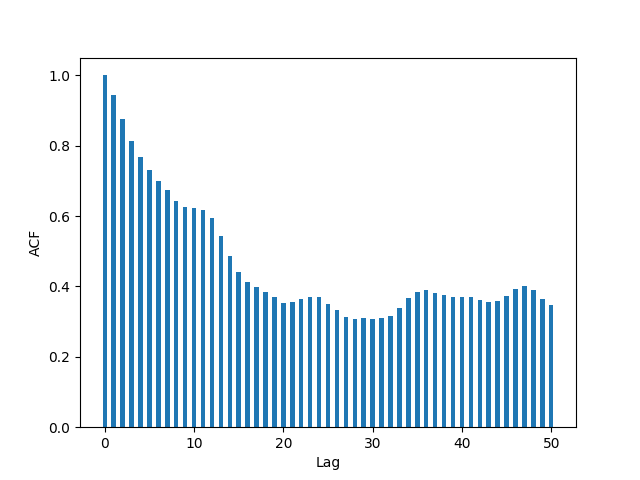


Figure 2. Retail sales in South Wales. An example of trend and seasonality in data [2]

Going back to the data presenting in Figure 1, at a glance, it looks like there is no consistent trend, except from ~2000 to ~2012. Seasonality might exist due to some peaks, but it doesn’t seem to be apparent either compared to the data shown in Figure 2.

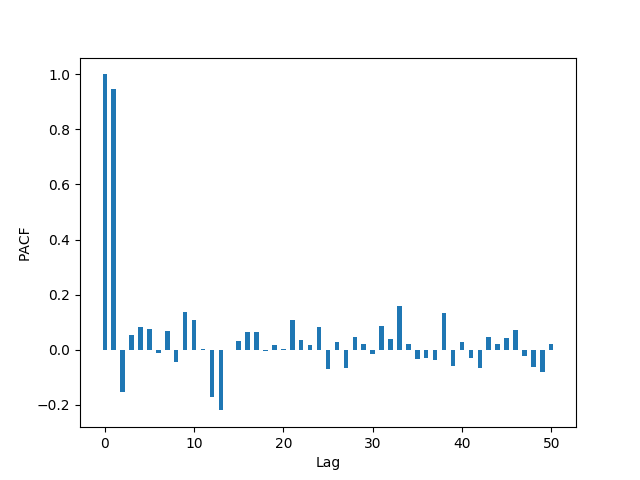
Breaking down a time series into its trend, seasonality, and noise (assuming one or more of the above exist in said series) helps if the series is stationary. The properties of a stationary series are that its mean and variance is the same across all xt, and any autocorrelation between xt and xt-n is the same. In analyzing, a stationary series, one can safely assume that each stochastic variable is independent, something that cannot be done in a non-stationary series. For example, if xt and xt-1 had a relationship, the variables are not independent, and useful properties for independent variables cannot be exploited [3]. One

**ACF**

****

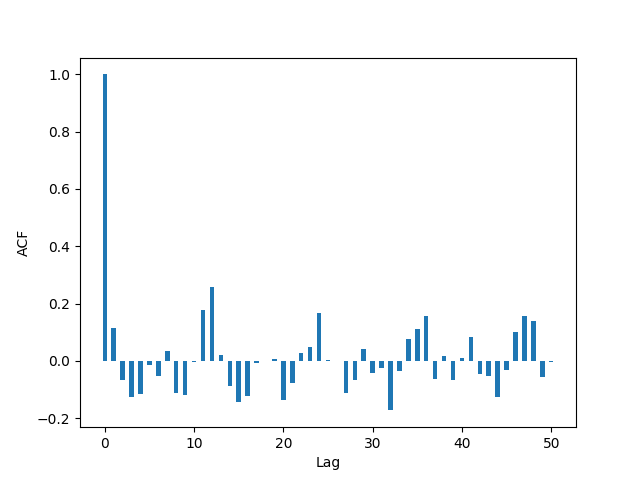
We see there is a tapering always positive for the 0 difference graph. This autocorrelation implies that there is a strong correlation for multiple lags past 1. The xt term is correlated with a lot of later terms. This implies that the process isn’t stationary, since are correlations between a current time point and future time points.

Let’s take a look at the partial ACF. The PACF takes into account xt andxt+k while discounting all other points.

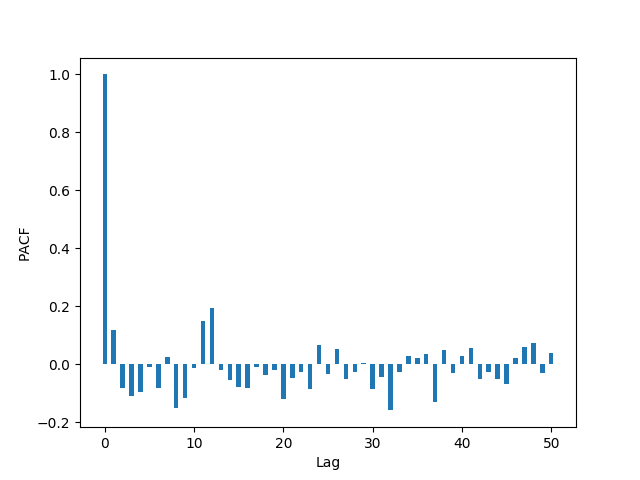


This has two lags and then almost no lags at all. The PACF indicates a strong correlation between xt andxt+1 suggesting an AR(1) model.

Taking a look at the ACF for a 1 diff model, which is essentially an AR(1):



It looks like later lags are not correlated with the 0 lag, which indicates that this is a stationary series.



The partial ACF also tells a similar story in that future lags are not as correlated with the 0 lag. It doesn’t look like we need to account for future lags in our AR(1) model.

**ADF (Augmented Dickey Fuller)**

Null hypothesis is that there’s a unit root, therefore it is a non-stationary process

If Statistic < Critical Value, we reject the null hypothesis and the series is stationary

**KPSS (Kwiatkowski-Phillips-Schmidt-Shin)**

Null hypothesis is that the trend is stationary on the absence of a root

If Statistic > Critical Value, we reject the null hypothesis and the series is non-stationary

Strict stationary - mean, variance, covariance are not a function of time

Trend stationary - series has no unit root. A lack of a unit root means that the series is either trend stationary or series stationary. KPSS classifies stationary as the absence of a root. Therefore if we reject the KPSS null hypothesis, the time series is not a trend stationary trend

Difference stationary - ADF is the difference stationary test

* **Case 1**: Both tests conclude that the series is not stationary -> series is not stationary
* **Case 2**: Both tests conclude that the series is stationary -> series is stationary
* **Case 3**: KPSS = stationary and ADF = not stationary  -> trend stationary, remove the trend to make series strict stationary
* **Case 4**: KPSS = not stationary and ADF = stationary -> difference stationary, use differencing to make series stationary

Hog Time Series

|  |  |
| --- | --- |
| Results of Dickey Test |  |
| Critical Values: |  |
|  | 1%: -3.445 |
|  | 5%: -2.868 |
|  | 10%: -2.570 |
| Test Statistic | -3.590515 |
| p-value | 0.005945 |
| Lags Used | 0 |

Our test statistic is less than our 1% so we reject our null hypothesis that our series is non-stationary

|  |  |
| --- | --- |
| Results of KPSS Test: |  |
| Test Statistic | 12.299752 |
| p-value | 0.010000 |
| Lags Used | 0.000000 |
| Critical Value (10%) | 0.347000 |
| Critical Value (5%) | 0.463000 |
| Critical Value (2.5%) | 0.574000 |
| Critical Value (1%) | 0.739000 |

Our test statistic is greater than our 1% critical value so we reject our null hypothesis that our series is stationary

**Case 4**: KPSS = not stationary and ADF = stationary -> difference stationary, use differencing to make series stationary

Let’s try the same tests on an AR(1) Model (a diff model)

|  |  |
| --- | --- |
| Results of Dickey Test |  |
| Critical Values: |  |
|  | 1%: -3.445 |
|  | 5%: -2.868 |
|  | 10%: -2.570 |
| Test Statistic | -1.875213e+01 |
| p-value | 2.027503e-30 |
| Lags Used | 0.000000e+00 |

Our test statistic is less than our 1% so we reject our null hypothesis that our series is non-stationary

|  |  |
| --- | --- |
| Results of KPSS Test: |  |
| Test Statistic | 0.021948 |
| p-value | 0.010000 |
| Lags Used | 0.000000 |
| Critical Value (10%) | 0.347000 |
| Critical Value (5%) | 0.463000 |
| Critical Value (2.5%) | 0.574000 |
| Critical Value (1%) | 0.739000 |

Our test statistic is less than our 10% critical value so we do not reject our null hypothesis that our series is stationary.

Now we can see that based off these statistics, an AR(1) provides a stationary series.

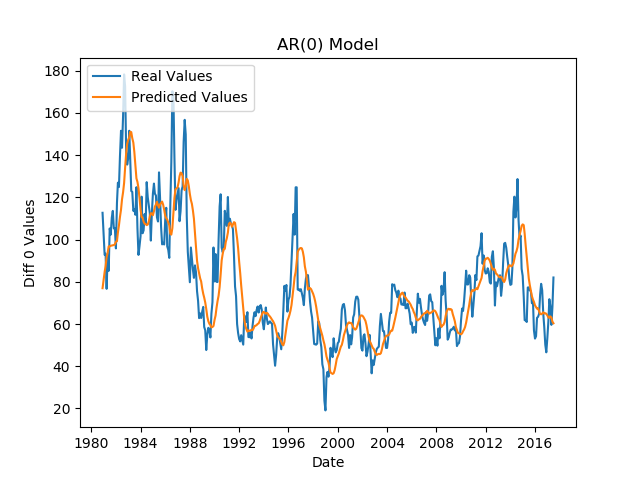
|  |  |
| --- | --- |
| Lags | Root MSE |
| 0 | 326.3551822 |
| 1 | 225.0732931 |
| 2 | 244.7989364 |
| 3 | 346.262086 |
| 4 | 725.731237 |
| 5 | 2331.214312 |
| 6 | 405.2250549 |
| 7 | 264.190726 |

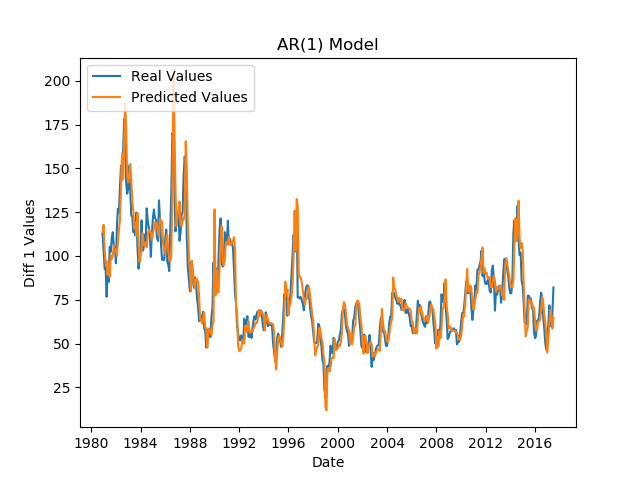
Diff 1 series

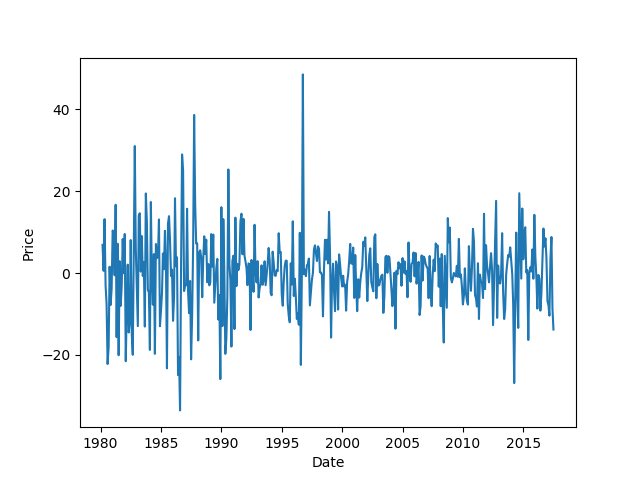
|  |  |
| --- | --- |
| Lags | Root MSE |
| 0 | 202.3223417 |
| 1 | 217.7128449 |
| 2 | 270.5854379 |
| 3 | 337.3084331 |
| 4 | 953.7468609 |
| 5 | 1600.083668 |
| 6 | 404.5364981 |
| 7 | 264.5839061 |

Clearly we have shown that an AR(1) is a good model for this time series.

Let’s look at our fit for our AR(1) model compared to our AR(0) model.







It actually looks like we’ve smoothed out our MA model at this point in time with differencing our values, so our lack of MA in our ARIMA model makes sense.

MA models are generally done when the following criteria are met:

* Negatively Autocorrelated at Lag  1
* When the ACF drops sharply after a few lags
* When the PACF decreases more gradually

None of these fit the criteria so an ARIMA would just be (1, 1, 0).

Let’s take a look at soy futures and do a similar thing.

|  |  |
| --- | --- |
| Results of Dickey Test |  |
| Critical Values: |  |
|  | 1%: -3.445 |
|  | 5%: -2.868 |
|  | 10%: -2.570 |
| Test Statistic | -1.851707 |
| p-value | 0.355109 |
| Lags Used | 0 |

It looks like we are going to not reject the null hypothesis that there is a root. This means we are probably looking at a non-stationary series according to the dickey test.

|  |  |
| --- | --- |
| Results of KPSS Test: |  |
| Test Statistic | 20.287009 |
| p-value | 0.010000 |
| Lags Used | 0.000000 |
| Critical Value (10%) | 0.347000 |
| Critical Value (5%) | 0.463000 |
| Critical Value (2.5%) | 0.574000 |
| Critical Value (1%) | 0.739000 |

It looks like we are going to reject the null hypothesis that the series is stationary.

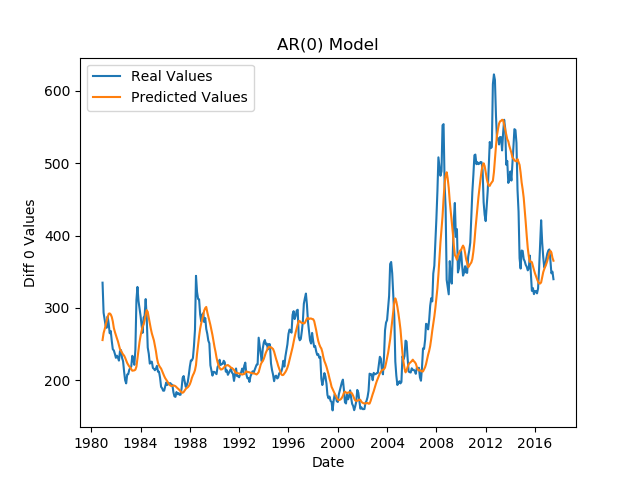
Since both tests show non-stationary results, it looks like we will need to remove a trend as well as have a difference.

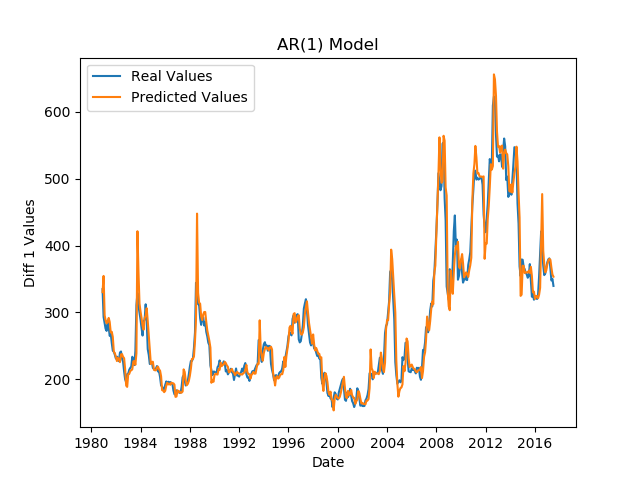
If we take a look at the AR(1)

|  |  |
| --- | --- |
| Results of Dickey Test |  |
| Critical Values: |  |
|  | 1%: -3.445 |
|  | 5%: -2.868 |
|  | 10%: -2.570 |
| Test Statistic | -1.529554e+01 |
| p-value | 4.383626e-28 |
| Lags Used | 0 |

|  |  |
| --- | --- |
| Results of KPSS Test: |  |
| Test Statistic | 0.053106 |
| p-value | 0.100000 |
| Lags Used | 0.000000 |
| Critical Value (10%) | 0.347000 |
| Critical Value (5%) | 0.463000 |
| Critical Value (2.5%) | 0.574000 |
| Critical Value (1%) | 0.739000 |

It looks like our AR(1) series is stable.





It’s possible that we’re not using the correct fitting techniques here.

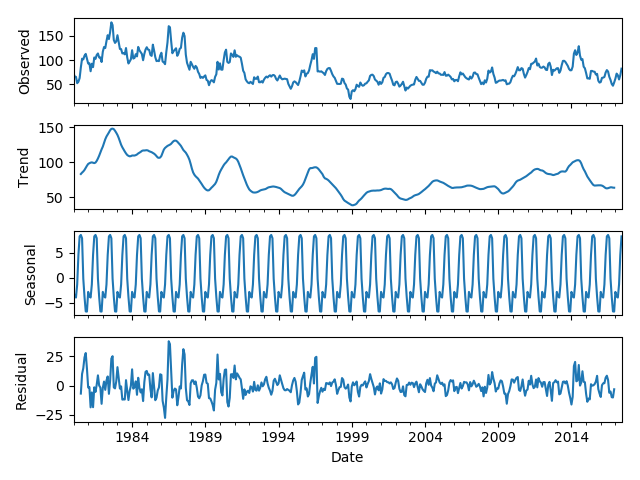
We can attempt to brute force some parameters with ARIMA fitting

|  |  |
| --- | --- |
| ARIMA | AIC |
| ARIMA(2, 1, 1) | 3231.797806159476 |
| ARIMA(1, 0, 1) | 3240.117825932588 |
| ARIMA(2, 0, 0) | 3240.6409119875557 |
| ARIMA(1, 0, 2) | 3242.1166999471693 |
| ARIMA(2, 0, 1) | 3242.117287994107 |
| … | … |

The problem with ARIMA brute forcing is that you’re generally looking at a lot of values in a series. Lower sample sizes have issues with invertibility and convergence.

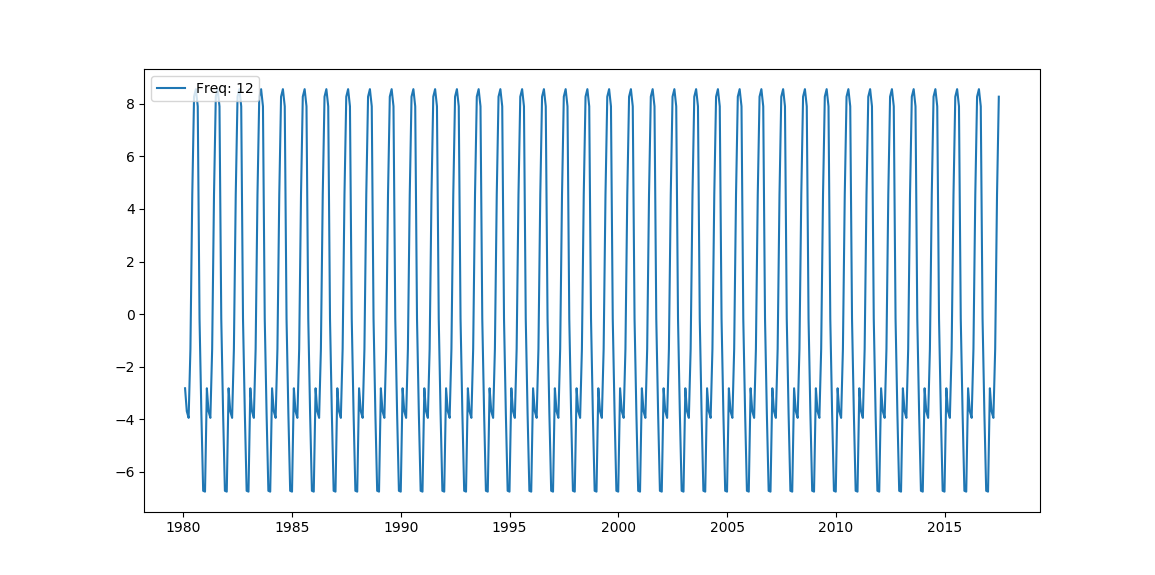
Let’s take a look at some of the tools available to us to better do a fit. We can attempt to brute force an SARIMA like this.

SARIMA takes into account seasonal patterns. Let’s take a look at the seasonal decompose plot first



The seasonal plot for our hog series shows our data has seasonality

Check residuals



Note: Seasonal decompose automatically breaks down monthly values to 12 lags. Is it possible that our ARIMA model requires SARIMA as well?

We try to brute force a SARIMA for parameters from 0 to 2 lags for both the trend and the seasonal trend.

|  |  |
| --- | --- |
| SARIMA | AIC |
| ARIMA(1, 0, 2)x(2, 2, 2, 12) | 2875.193804769013 |
| ARIMA(2, 1, 2)x(1, 2, 2, 12) | 2879.658124463105 |
| ARIMA(0, 1, 2)x(2, 2, 2, 12) | 2880.2491591904736 |
| ARIMA(1, 1, 2)x(2, 2, 2, 12) | 2881.0527786827993 |
| ARIMA(2, 1, 2)x(0, 2, 2, 12) | 2881.6540130873746 |
| ARIMA(1, 0, 2)x(1, 2, 2, 12) | 2882.275100996529 |
| … | … |

We see the SARIMAX with (1, 0, 2) x (2, 2, 2, 12) seems to fit best with our AIC score.

Sources

1. <https://www.itl.nist.gov/div898/handbook/pmc/section4/pmc41.htm>
2. <http://www.abs.gov.au/websitedbs/d3310114.nsf/home/time+series+analysis:+the+basics>
3. <https://newonlinecourses.science.psu.edu/stat510/node/60/>

<https://www.analyticsvidhya.com/blog/2018/09/non-stationary-time-series-python/>

<https://machinelearningmastery.com/gentle-introduction-autocorrelation-partial-autocorrelation/>

<https://machinelearningmastery.com/autoregression-models-time-series-forecasting-python/>

<https://coolstatsblog.com/2013/08/07/how-to-use-the-autocorreation-function-acf/>

<https://support.minitab.com/en-us/minitab/18/help-and-how-to/modeling-statistics/time-series/how-to/partial-autocorrelation/interpret-the-results/partial-autocorrelation-function-pacf/>

<https://www.quora.com/What-is-the-difference-among-auto-correlation-partial-auto-correlation-and-inverse-auto-correlation-while-modelling-an-ARIMA-series>

<https://otexts.com/fpp2/moving-averages.html>

<https://machinelearningmastery.com/sarima-for-time-series-forecasting-in-python/>

<https://quant.stackexchange.com/questions/8875/why-non-stationary-data-cannot-be-analyzed>

**ADF (Augmented Dickey Fuller)**

**https://freakonometrics.hypotheses.org/12729**

**https://www.investopedia.com/articles/trading/07/stationary.asp**

**https://www.xycoon.com/ma1\_process.htm**

**https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/**

**https://freakonometrics.hypotheses.org/12729**

**https://www.analyticsvidhya.com/blog/2016/02/time-series-forecasting-codes-python/**

**Plotting rolling mean and std dev could help**

**https://machinelearningmastery.com/moving-average-smoothing-for-time-series-forecasting-python/**