Time Series Analysis

ECE 6563, Electrical and Computer Engineering
University of New Brunswick

Recap

In the last section, we briefly introduced times series

- What is a time series
- Some examples of time series
- The objectives of time series analysis
- The basic components of a time series

This section:

- How do we remove the various components of a time series so that we can focus on the residuals…

Estimate or Eliminate

Time Series Decomposition

Step 1: Estimate the trend.

Step 2: De-trend the series.

- For an additive model, subtract the estimated trend from the series.
- For a multiplicative model, divide the series by the estimated trend.

Step 3: Estimate any seasonal factors using the de-trended series.

Step 4: Find the random (irregular), unexplained component.

- For the additive model, $Y_t = X_t - m_t - s_t$
- For the multiplicative model, $Y_t = \dfrac{X_t}{m_t s_t}$

Then: Analyze the random component! (to be continued…)

Time-Series Components

Time Series

- Trend Component
- Seasonal Component
- Cyclical Component
- Residual Component

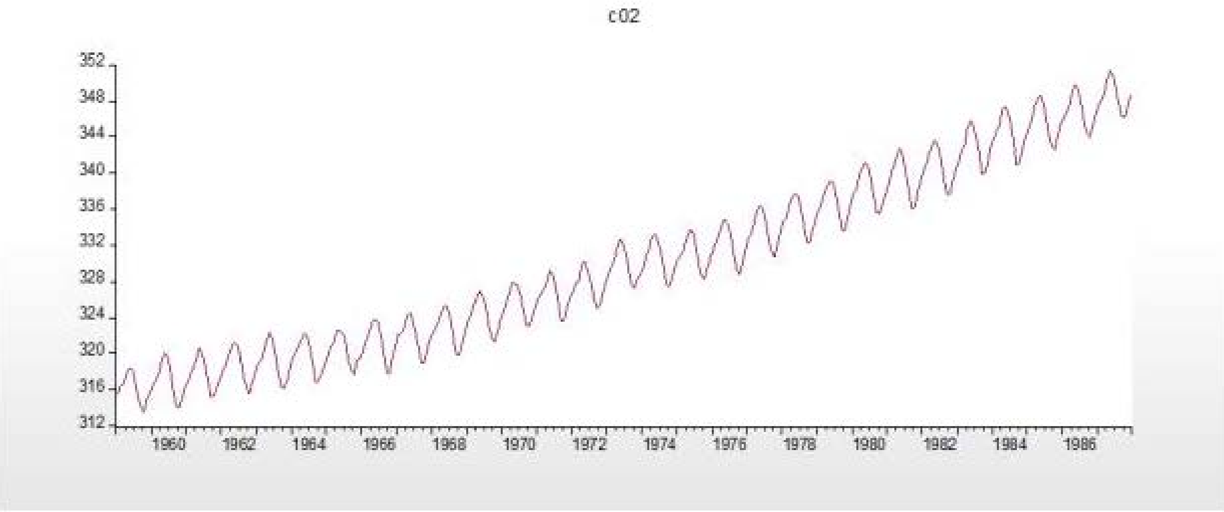
Trend Component	Seasonal Component	Cyclical Component	Residual Component
Overall, persistent, long-term movement	Regular periodic fluctuations	Irregular oscillations or movements	Irregular, Erratic, Noise, or residual fluctuations

Time-Series Components

A time series model that incorporates trend and seasonality is as follows:

$$X_t = m_t + s_t + Y_t$$

We ultimately want to be able to remove these trends so that all we have left is the residual component...



Modeling Components

So, the challenge becomes - how do we remove these various components?

$$\boxed{X_t = m_t + s_t + Y_t}$$

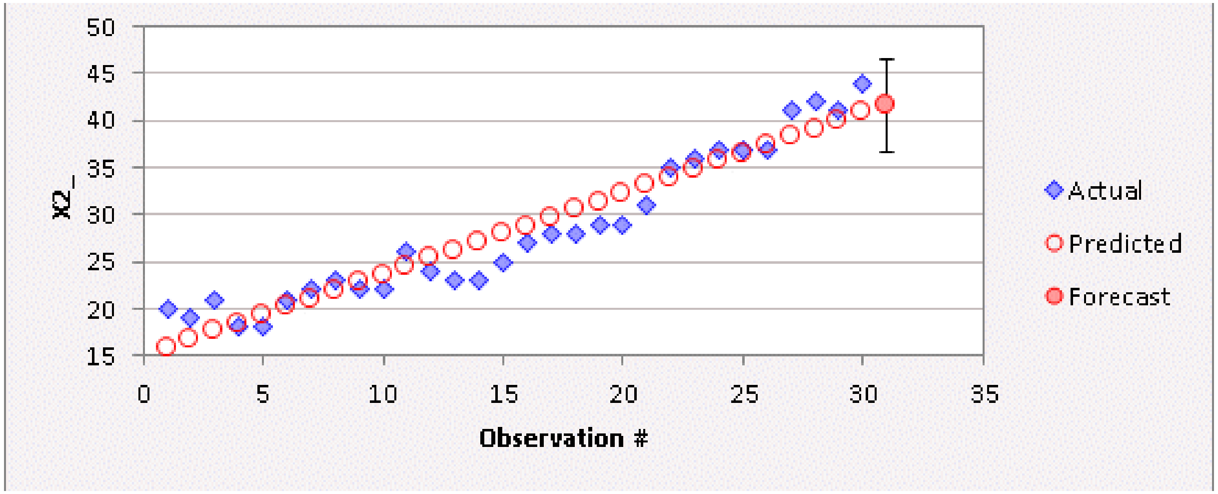
There are two basic approaches:

- Trend estimation (estimate/model the trend so that we can remove it from the signal)
- Trend elimination (remove the trend using differencing)

Parametric Trend Estimation

One approach to estimating a trend line is to assume that it can be modeled using a finite set of parameters

- Fitting a linear or quadratic curve using regression



Note: For least squares modeling, time is numbered starting with 0 and increasing by 1 for each period/sample/observation

Parametric Trend Estimation

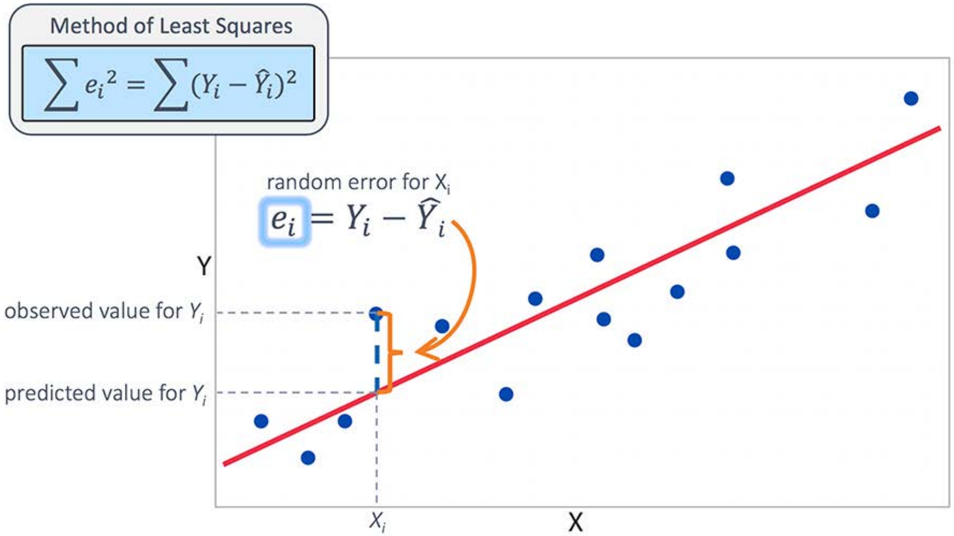
One approach to estimating a trend line is to assume that it can be modeled using a finite set of parameters

- Fitting a linear or quadratic curve using regression
- Use least squares regression (think "find the line of best fit")
- Use time (t) as the independent variable.
- Linear Example:

$$X_t = \beta_0 + \beta_1 t$$

Intercept

Slope



https://www.jmp.com/en_hk/statistics-knowledge-portal/what-is-multiple-regression/fitting-multiple-regression-model.html

Parametric Trend Estimation

Method of Least Squares

- Fitting a curve using linear regression

$$\beta_1 = \frac{\sum(x-\bar{x})(y-\bar{y})}{\sum(x-\bar{x})^2}$$

(here, x is time)

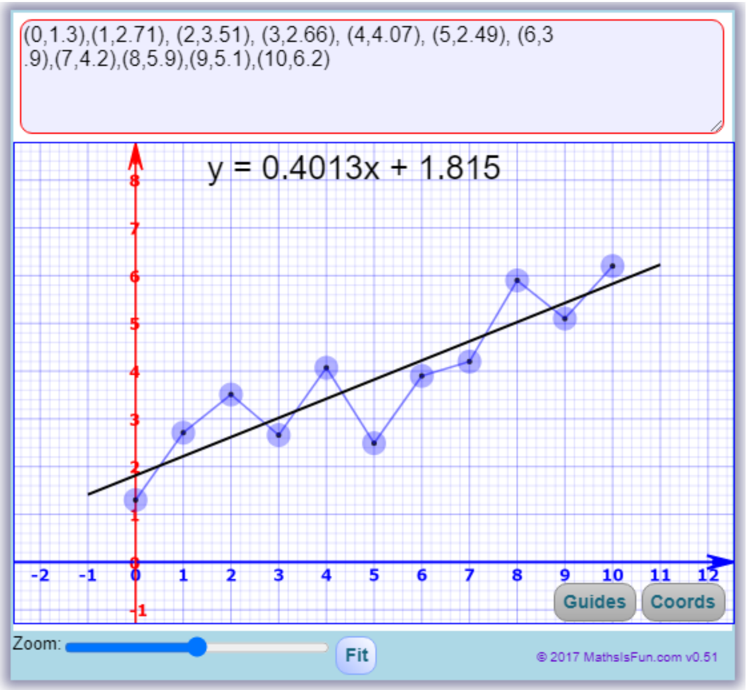
$\bar{x}, \bar{y}$ are means of x, y

$$\beta_0 = \bar{x} - \beta_1 \bar{y}$$

$$X_t = \beta_0 + \beta_1 t + Y_t$$

Intercept

Slope



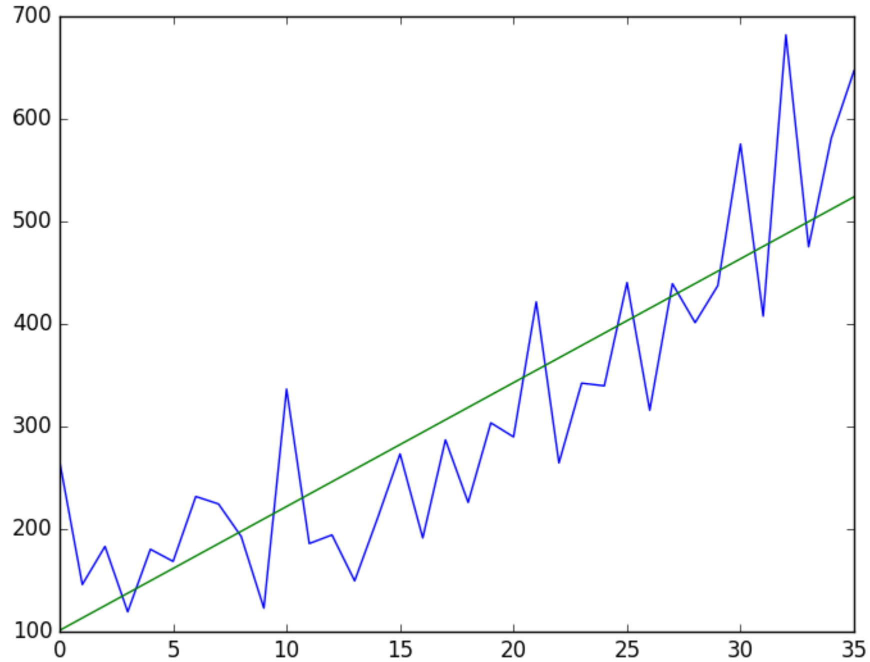
https://www.mathsisfun.com/data/least-squares-calculator.html

Parametric Trend Estimation

Least squares regression

- May not be optimal yet without knowing the structure of the remaining residuals, Y_t, but often not bad:

$$X_t = \beta_0 + \beta_1 t + Y_t \qquad\qquad X_t = \beta_0 + \beta_1 t + \beta_2\, t^2 + Y_t$$



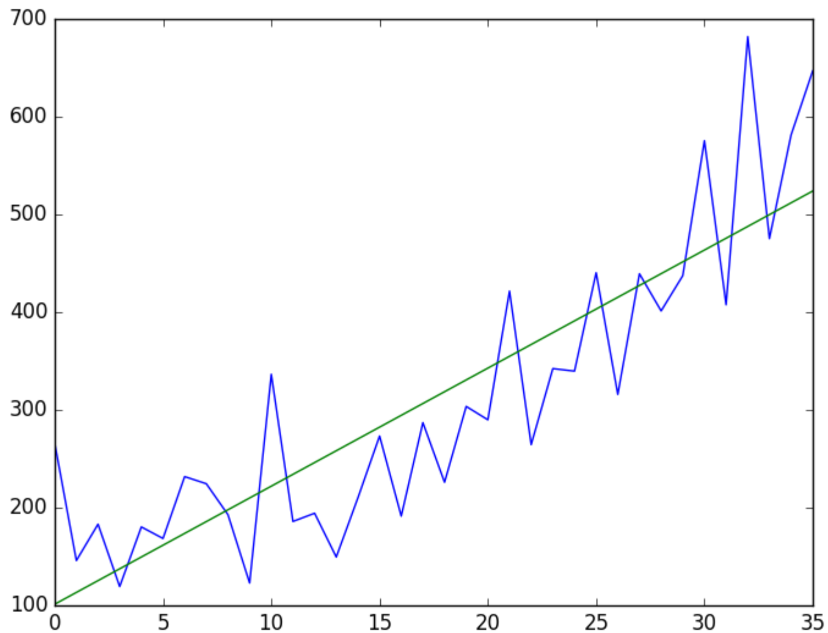
Parametric Trend Estimation

Trend Removal

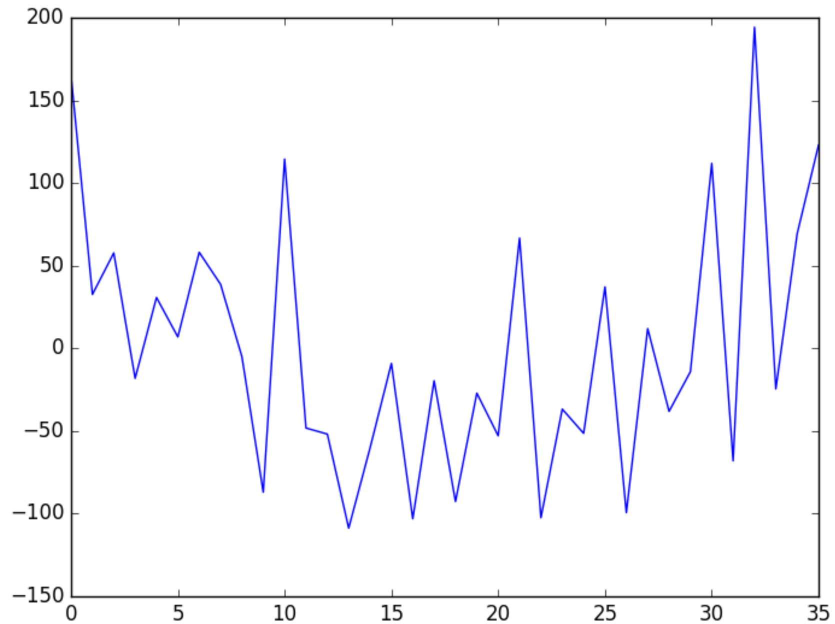
- Once the trend is known, it can be removed from the signal

$$X_t = \beta_0 + \beta_1 t + Y_t$$

$$\hat{Y}_t = X_t - \beta_0 + \beta_1 t$$



Original time series with trend line



De-trended estimate of time series

https://machinelearningmastery.com/time-series-trends-in-python/

Non-Parametric Trend Estimation

Moving Average Filter

- Parametric approaches make assumptions about the underlying data
- A trend is essentially just a slow-moving portion of the signal
- So, why not filter it out?
- We can use a moving average filter to estimate this slow trend

$$W_t = \frac{1}{(2N+1)} \sum_{j=-N}^{N} X_{t-j} \qquad \text{but } X_t = m_t + Y_t$$

$$W_t = \frac{1}{(2N+1)} \left\{ \sum_{j=-N}^{N} m_{t-j} + \sum_{j=-N}^{N} Y_{t-j} \right\} \approx m_t$$

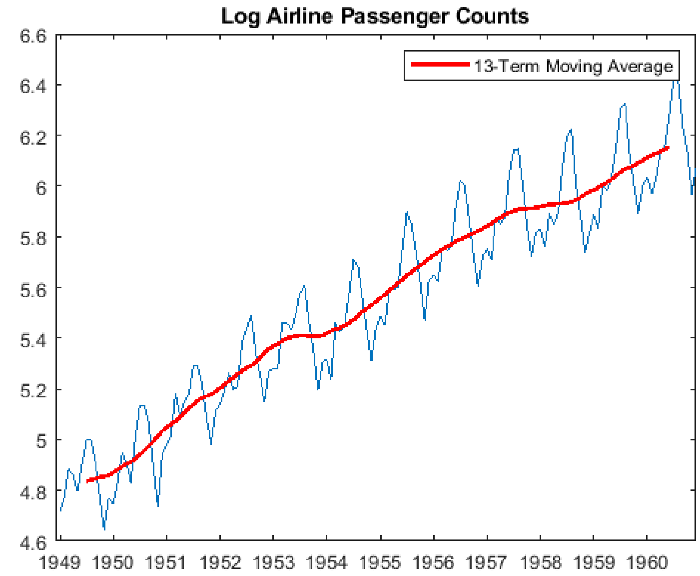
Non-Parametric Trend Estimation

Moving Average Filter

- Parametric approaches make assumptions about the underlying data
- A trend is essentially just a slow-moving portion of the signal
- So, why not filter it out?
- We can use a moving average filter to estimate this slow trend

MAF is effectively a simple low-pass FIR filter

Note: can also be done using a convolution function



https://www.mathworks.com/help/econ/moving-average-trend-estimation.html

Non-Parametric Trend Estimation

Moving Average Filter

- A benefit of MAF is that it makes no assumptions about the trend
- But, just like a real low-pass filter, it requires proper selection of the windowing function and parameters

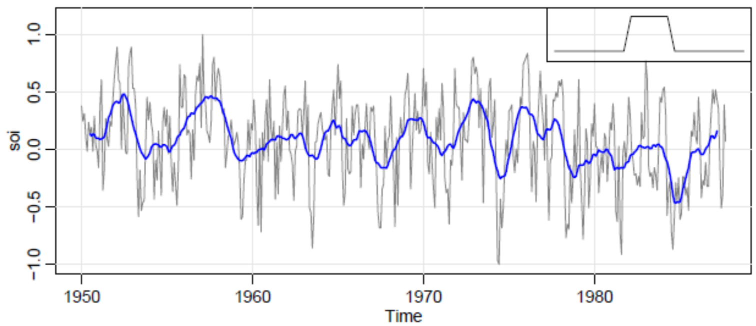
  There are many different options (e.g. exponential smoothing)

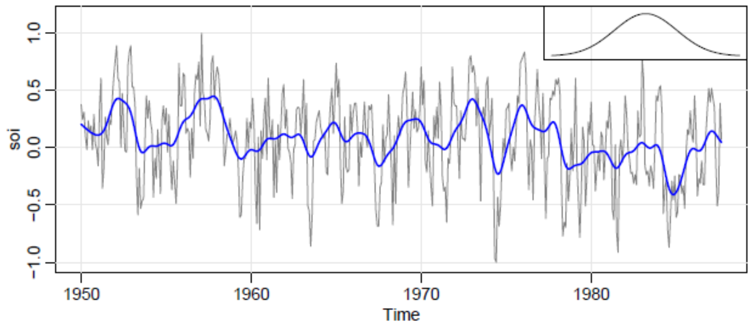
Non-Parametric Trend Estimation

Moving Average Filter

- A benefit of MAF is that it makes no assumptions about the trend
- But, just like a real low-pass filter, it requires proper selection of the windowing function and parameters

- The MAF estimate smooths the original signal using an average of the previous (or surrounding) points
- But we don’t have to use just the average
- We can use a smoothing kernel (such as in Parzen estimation)



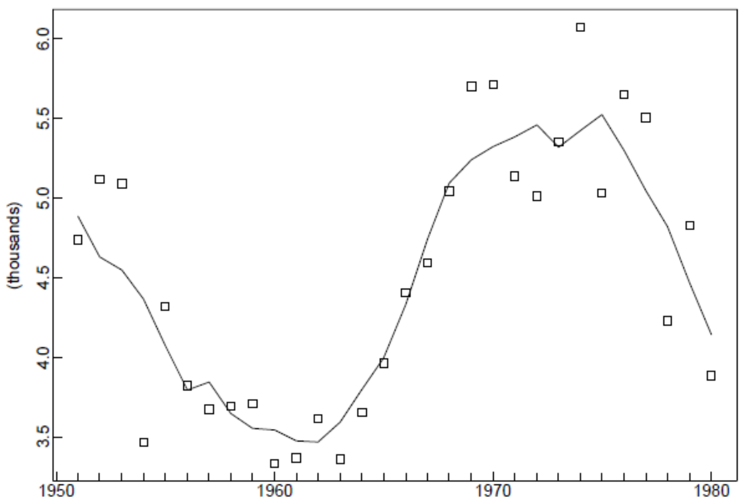


Non-Parametric Trend Estimation

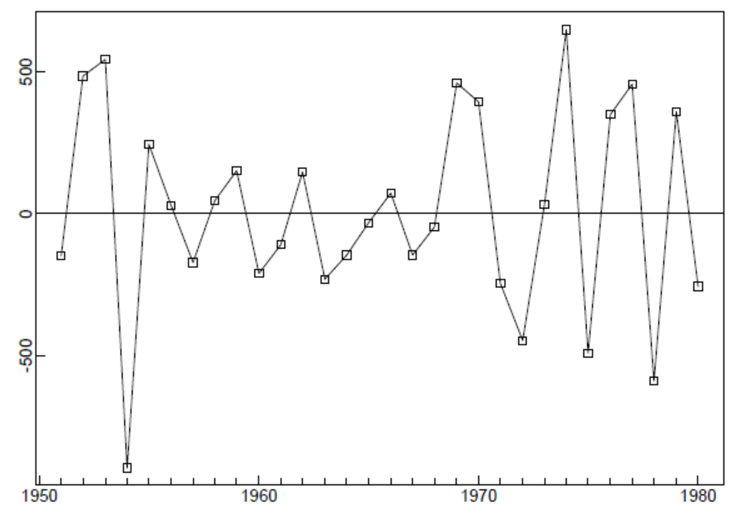
Moving Average Filter

- A benefit of MAF is that it makes no assumptions about the trend
- But, just like a real low-pass filter, it requires proper selection of the windowing function and parameters

Original time series



Detrended using MAF



Bonus! This trend was periodic…

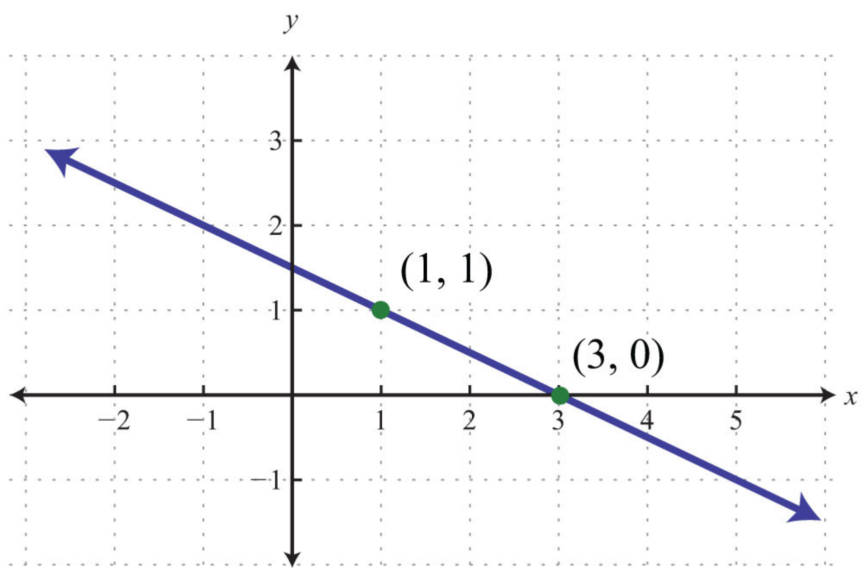
Trend Elimination using Differencing

Trend Elimination

- Instead of trying to model or estimate the trends, we can just try to remove them using differencing

$$\nabla X_t = X_t - X_{t-1}$$

What would be the result of differencing a perfect line with slope m_t?



Trend Elimination using Differencing

Trend Elimination

- Instead of trying to model or estimate the trends, we can just try to remove them using differencing

If $X_t = \beta_0 + \beta_1 t + Y_t$

$$Z_t = X_{t+1} - X_t$$

$$Z_t = \beta_1 + Y_{t+1} - Y_t$$

Trend Elimination using Differencing

Trend Elimination

- Instead of trying to model or estimate the trends, we can just try to remove them using differencing

$$\text{If} \quad X_t = \beta_0 + \beta_1 t + Y_t$$

$$Z_t = X_{t+1} - X_t$$

$$Z_t = \beta_1 + Y_{t+1} - Y_t$$

But we can also keep going to remove higher order trends…

$$\nabla^2 X_t = \nabla(\nabla X_t) \qquad or \quad \nabla^j(X_t) = \nabla(\nabla^{j-1} X_t)$$

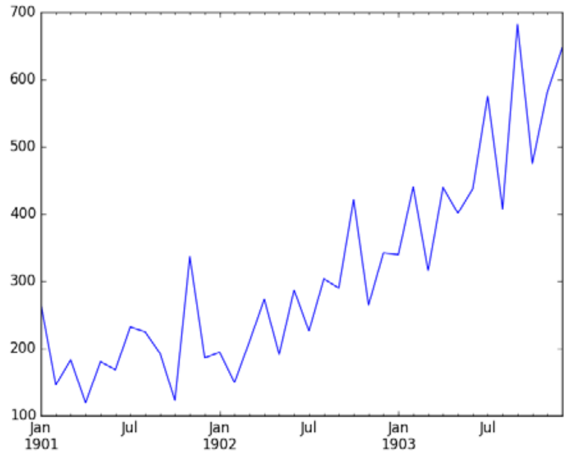
These operate the same way as polynomial functions, so

$$\nabla^2 X_t = X_t - 2X_{t-1} + X_{t-2}$$

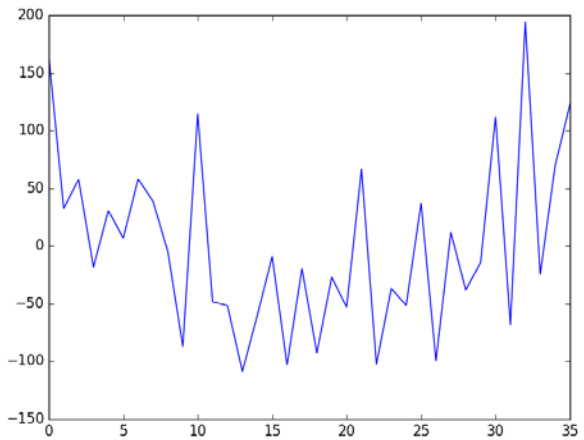
Trend Elimination using Differencing

Trend Elimination

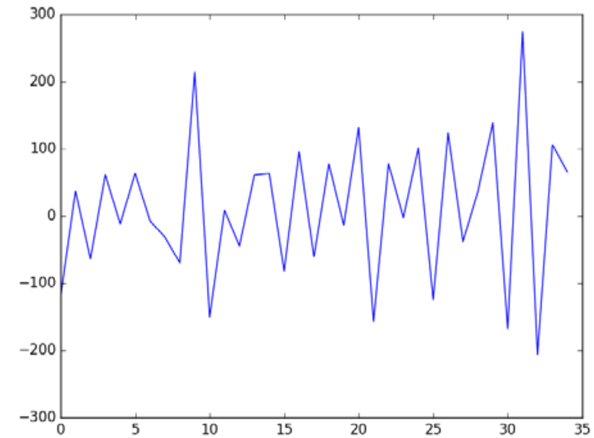
- Similar, but not the same results as trend estimation and removal



Original



De-trended using Estimation



De-trended using Differencing

Seasonality Estimation

Seasonal Averaging

- In many cases, seasonality is actual ‘seasonal’ information
- In simple cases, it is intuitive to calculate the seasonal averages (e.g. average value of sales in December vs June)

Compute the average value for each “season” (could be days, weeks, month, quarters, etc.) over time

- Assumes that other trends are removed first (otherwise, you can find the averages using a regression approach)

Subtract/divide it from the corresponding seasons

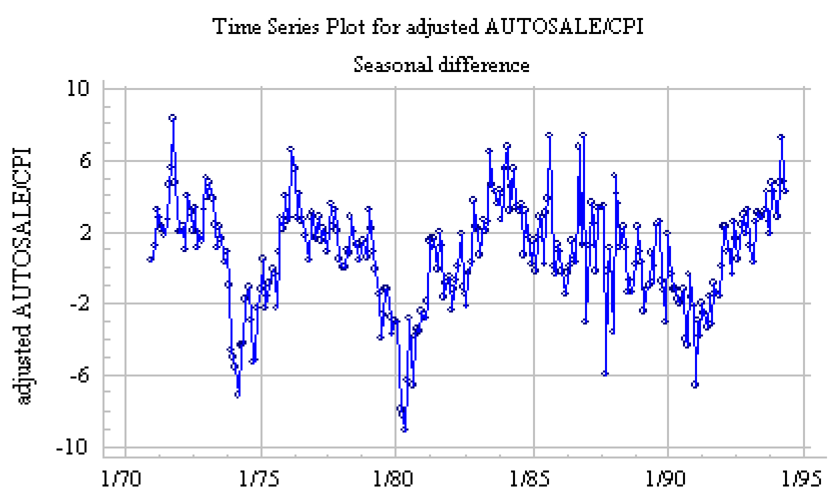
e.g. for each month, subtract the average sales for that month

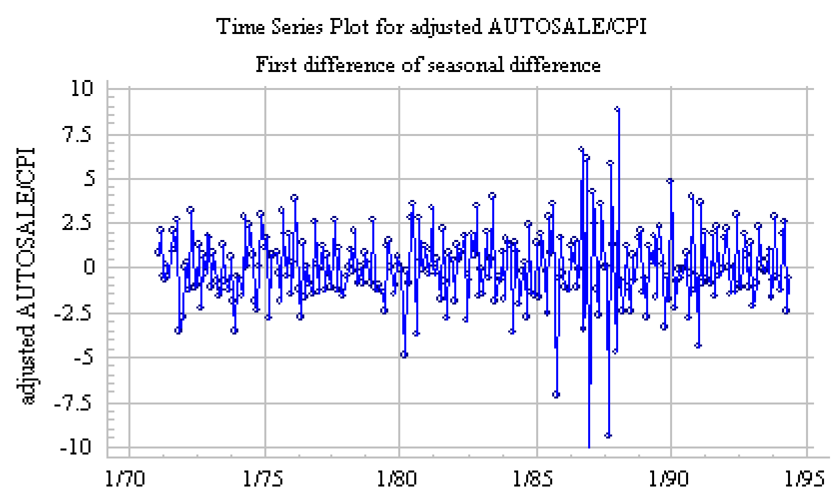
Seasonality Estimation

Seasonal Differencing

- Another crude method of handling seasonality is to use differencing
- Here, instead of the regular difference (a single step), we difference using a *lag* operator based on the seasonal period.

  e.g. for monthly values, we might use a lag-d operator of d=12, to take the difference from 12 samples prior





https://faculty.fuqua.duke.edu/~rnau/Decision411_2007/Class10notes.htm

Seasonality Estimation

Seasonal Cosine-Sine Modeling

- Because seasonality is periodic, we can model the signal using Fourier approaches

  Model the seasonal component as a sum of sines and cosines

Assume

- $s_t = \beta\cos(2\pi f t + \theta)$ where β is the amplitude, f is the frequency, and θ is the phase.

- Then, $s_t = \beta\cos(2\pi f t + \theta) = \beta_1\cos(2\pi f t) + \beta_2\sin(2\pi f t)$, where $\beta_1 = \beta\cos(\theta)$ and $\beta_2 = \beta\sin(\theta)$

- Fit a multiple linear regression model (using least squares approach) with β_1 and β_2 as the regression coefficients

Seasonality Estimation

Seasonal Cosine-Sine Modeling

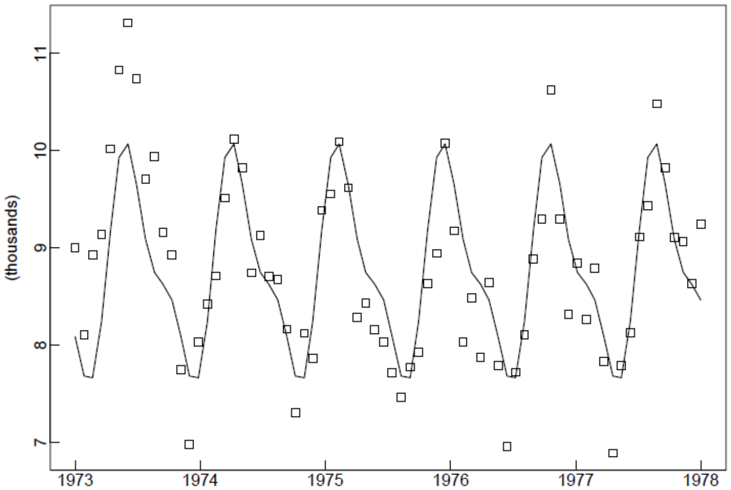
- Because seasonality is period, we can model the signal using Fourier approaches

  Model the seasonal component as a sum of sines and cosines

If seasonality has multiple frequencies, use multiple values of f (and two predicting variables for each f)

$$s_t = B_0 + \sum_{j=1}^{k} \alpha_j \cos(2\pi f_j t) + \gamma_j \sin(2\pi f_j t)$$

Here, k=2, with periods of 6 and 12 months

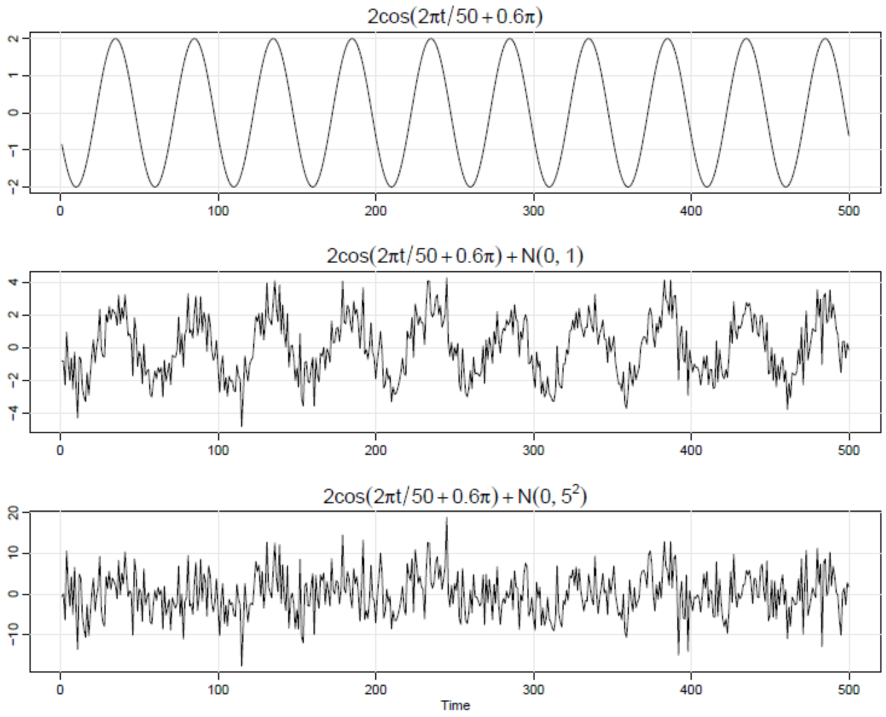


Seasonality Estimation

Seasonal Cosine-Sine Modeling

- Because seasonality is periodic, we can model the signal using Fourier approaches
  Model the seasonal component as a sum of sines and cosines

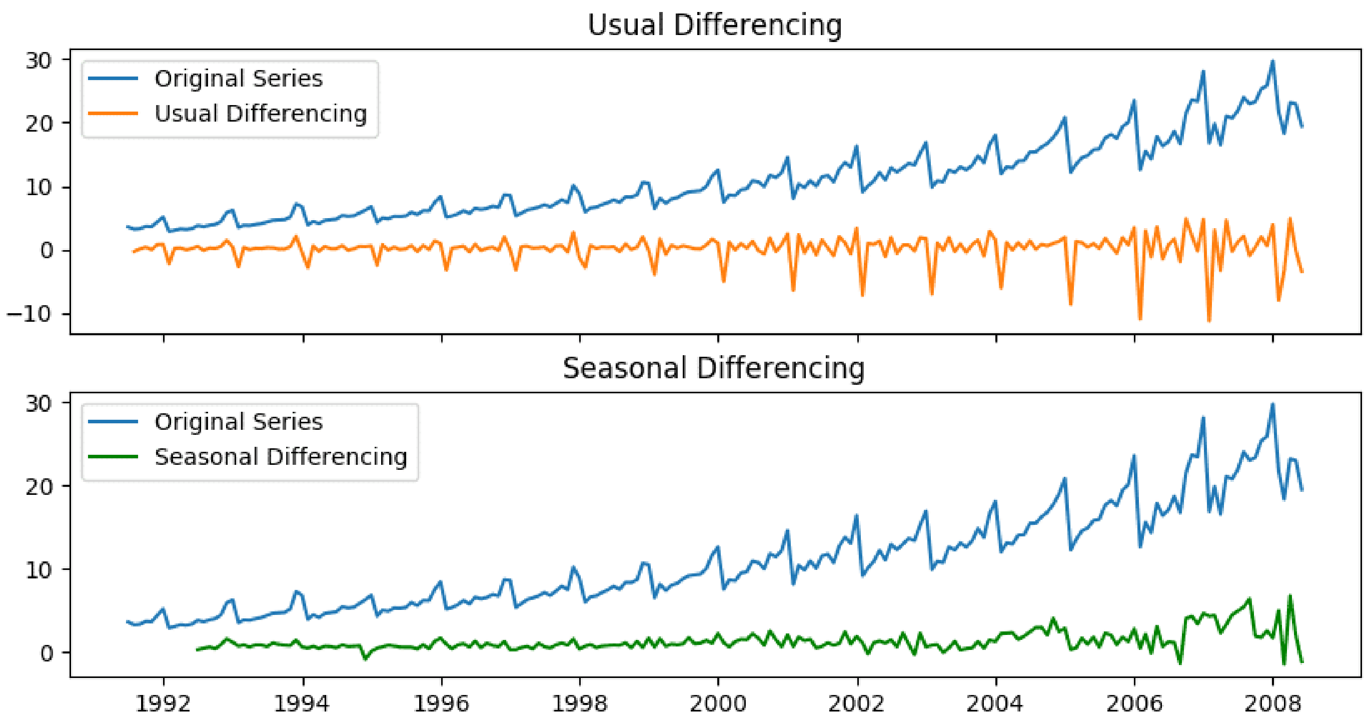
- Remember that the periodicity is not always so clear, especially in the engineering discipline



Seasonality Removal

Just as with trend removal, we can sometimes apply differencing to remove seasonality.

- We just need to difference by an amount consistent with the seasonality



Diff(1)

Diff(12)

https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/

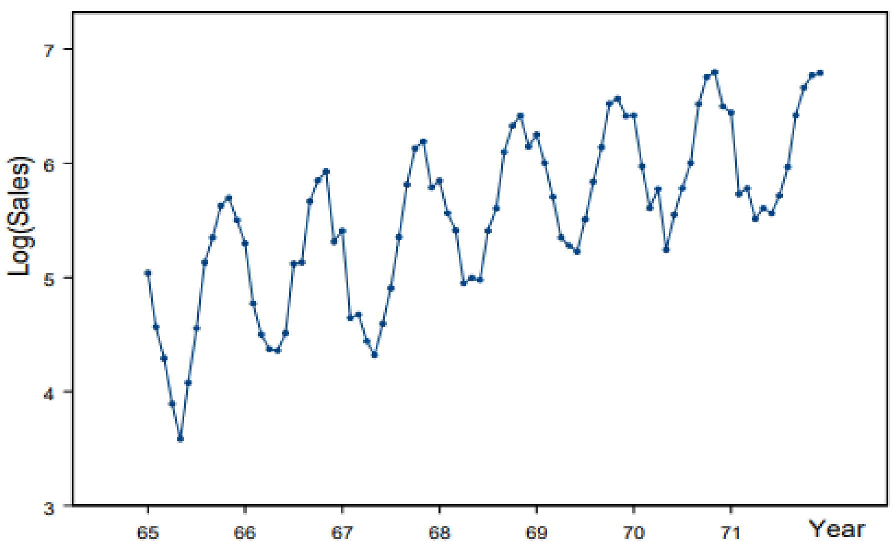
Joint Estimation

Above, we first estimated the trend, and then the seasonality

- Sometimes, we may need to re-estimate the trend once the seasonality has been removed, to further tune our model

Alternatively, we can directly fit a combined polynomial linear regression and harmonic functions to solve for the full model:

$$X_t = (B_0 + B_1 t + B_2 t^2) + \left[\sum_{j=1}^{k} \alpha_j \cos\left(2\pi f_j t\right) + \gamma_j \sin\left(2\pi f_j t\right)\right] + Y_t$$

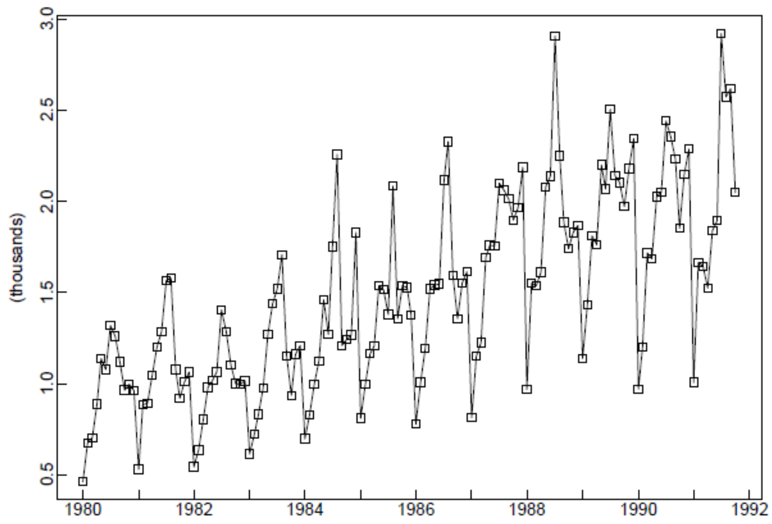


Transformation

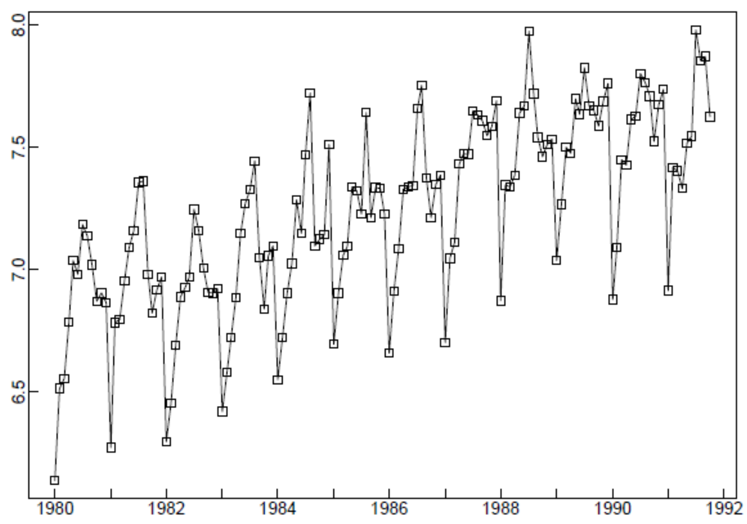
There are many other “tricks” we can use to clean up a time series. Transformation is one common approach.

- Used to accommodate for non-linear growth
- Square-root, log, Box-Cox, etc.

Sample seasonal time series



Corresponding natural log



Improves our ability to then model this as

$$X_t = m_t + s_t + Y_t$$

Now What?

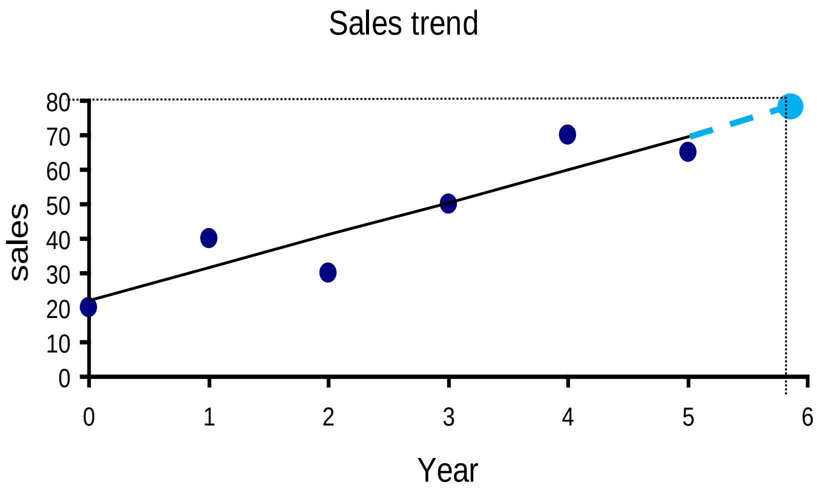
We've just spent some time talking about how to remove the trends and seasonality from a time series

- These tools can also be used to remove other cyclic aperiodic trends
- The goal was to leave only the random portion
- So now what?

- Naively, once the trend is known, we can predict future values:

Year	Time (t)	Sales (x)
2004	0	20
2005	1	40
2006	2	30
2007	3	50
2008	4	70
2009	5	65
2010	6	??

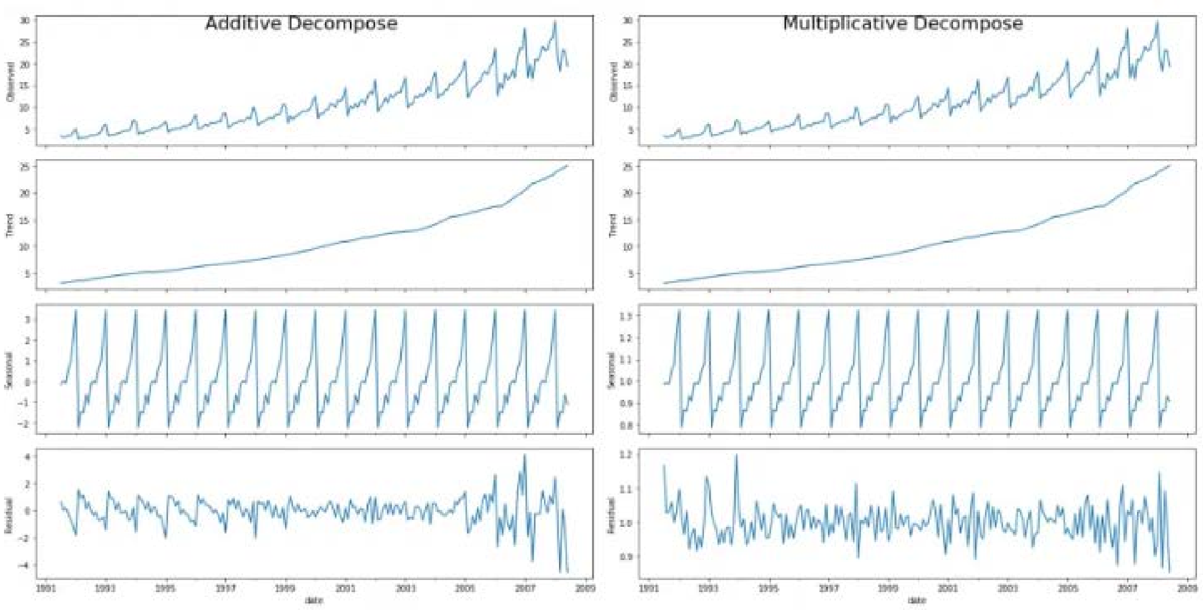
$$\hat{X}_t = 21.905 + 9.571 * \boxed{6} = 79.33$$



Now What?

The prediction performance may not be great yet though

- There may be more information hidden in the residuals!



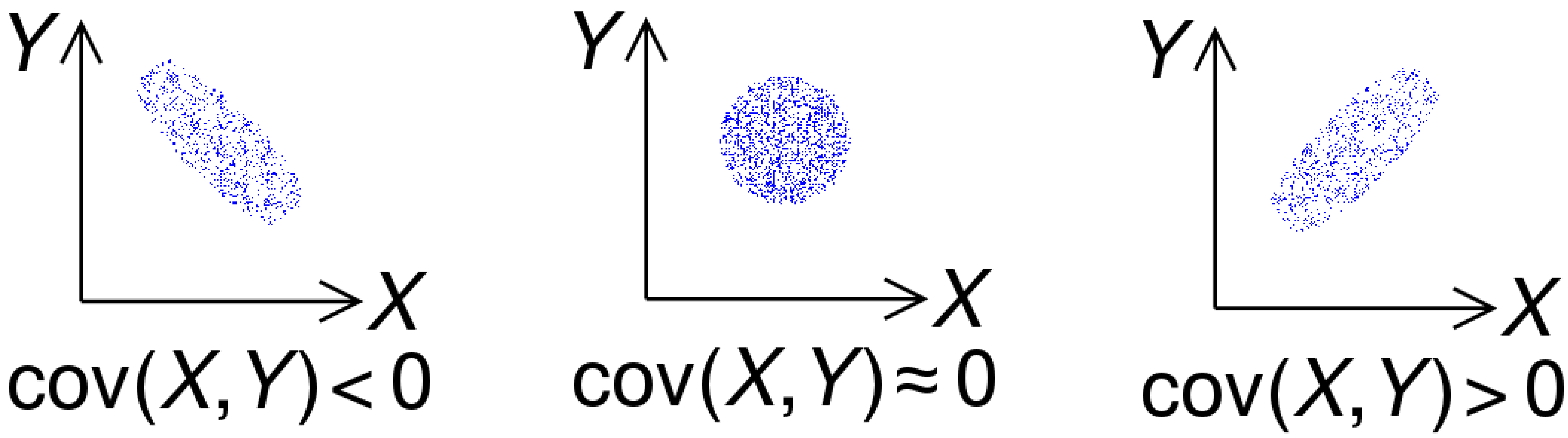
The goal of estimating/removing trends was to achieve stationarity

- System does not change its properties in time
- Mean and covariance are independent of *t*

Autocovariance

Before we introduce stationarity, it is advantageous to introduce some new measures: *autocovariance* and *autocorrelation*…

For a general multi-dimensional vector ***X*** (think features, in ML), we can calculate a covariance matrix



Autocovariance

Before we introduce stationarity, it is advantageous to introduce some new measures: *autocovariance* and *autocorrelation*…

For a general multi-dimensional vector ***X*** (think features, in ML), we can calculate a covariance matrix

- In time series, this usually involves very large numbers of random variables (every value of the time series, with every other value)

- Instead, we introduce autocovariance as an extension

$$\gamma_X(r,s) = E[(X_r - E[X_r]) \cdot (X_s - E[X_s])]$$

$$\gamma_X(r,s) = E[(X_r X_s)] - E[X_r]\, E[X_s]$$

where r and s are two moments in time, some amount of time, τ, apart

Autocovariance

- We instead introduce autocovariance as an extension

$$\gamma_X(r,s) = E[(X_r - E[X_r]) \cdot (X_s - E[X_s])]$$

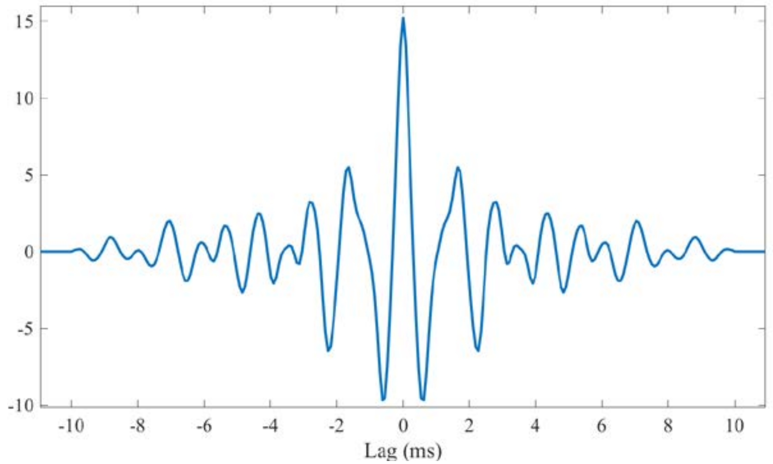
$$\gamma_X(r,s) = E[(X_r X_s)] - E[X_r]\, E[X_s]$$

where r and s are two moments in time, some amount τ apart

- That is

$$\gamma_X(\tau) = cov(X_{t+\tau}, X_t)$$

A signal's relationship with a previous/future version of itself



Autocorrelation

The autocorrelation of a time series is simply a normalized version of autocovariance

$$\gamma_X(\tau) = cov(X_{t+\tau}, X_t)$$

$$\rho_X(\tau) = \frac{\gamma_X(\tau)}{\gamma_X(0)}$$

- Shares all the properties of the autocovariance function, except that $\rho_X(0) = 1$.
- Measures the relationship between a variable's current value and its past values.
- e.g.

  If temperatures for days closer together in time are more similar than the temperatures for days that are farther apart, then the data are autocorrelated.

Stationarity

A time series is stationary if its statistical properties are similar to those of a time-shifted version of itself.

- In other words, the overall behavior of the series remains the same over time.
- This is why we have to remove trends

A series, $\{X_t\}$, is considered to be (weakly) stationary if:

$$E[X_t] = \mu \quad \text{for all } t$$

$$E[X_t{}^2] < \infty \quad \text{for all } t$$

$$\gamma(X_{t+\tau}, X_t) = \gamma_\tau$$

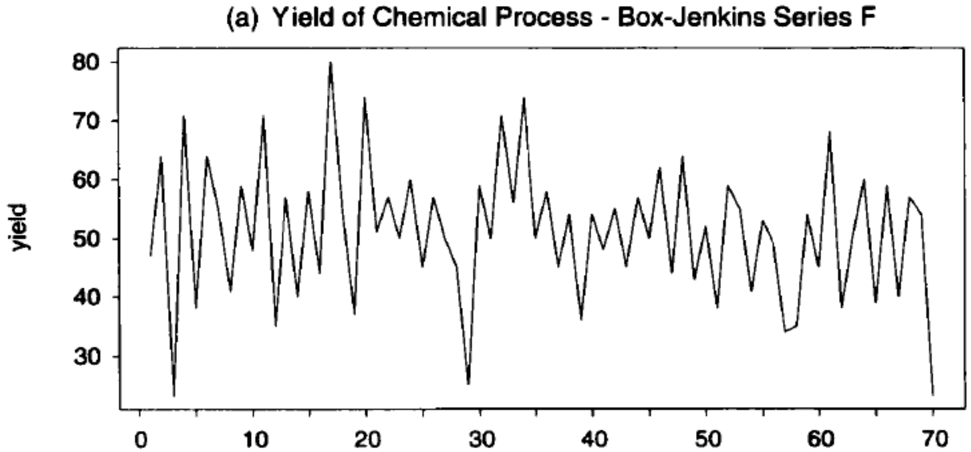
That is, for a given τ, the autocovariance of $(X_{t+\tau}, X_t)$ does not depend on t (it is constant for a given lag τ).

Stationarity

A time series is stationary if its statistical properties are similar to those of a time-shifted version of itself.

- In other words, the overall behavior of the series remains the same over time.
- This is why we have to remove trends

The observations fluctuate about a fixed mean level with constant variance over the observational period.



White Noise

A sequence of random variables $\{X_t\}$ is called white noise if

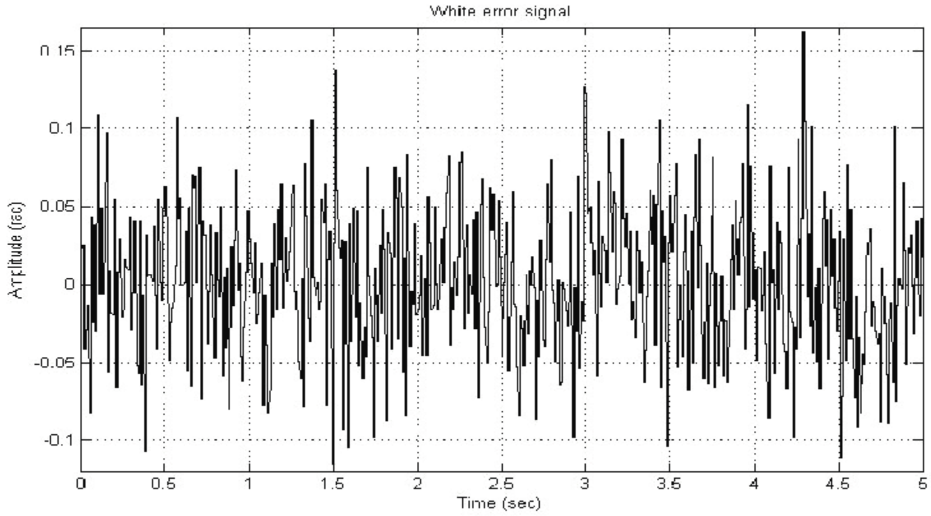
- they are uncorrelated

$$\gamma(\tau) = \begin{cases} \sigma^2, & \tau = 0 \\ 0, & otherwise \end{cases}$$

- with $\mu = 0$, and $\sigma^2 < \infty$
- denoted by:

$$\{X_t\} = WN(0, \sigma^2)$$

Note: White noise is stationary, but also completely random



IID

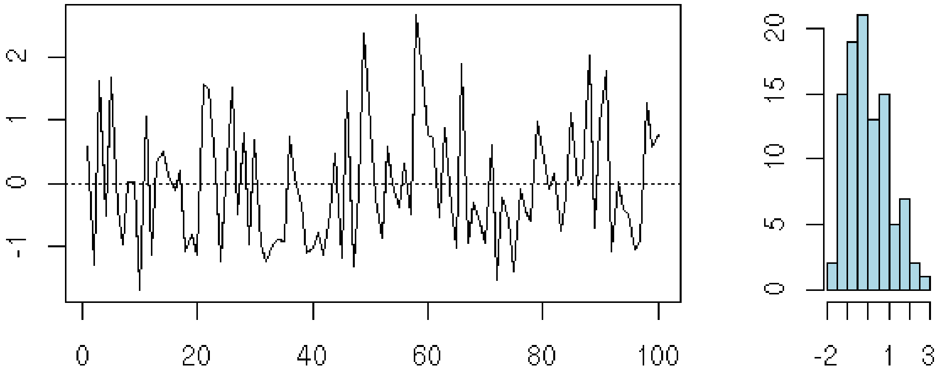
If the observations in the time series are independent, and come from the same distributions, the series is termed *iid*

- iid – *independent* and *identically distributed* random variables with zero mean

$$\{X_t\} = IID(0, \sigma^2)$$

- Past values have no bearing/impact on future values
- The best guess of the next value is to predict the mean of the random variable (zero)

Note: An IID time series is a special case of white noise



IID

If the observations in the time series are independent, and come from the same distributions, the series is termed *iid*

- iid – *independent* and *identically distributed* random variables with zero mean

$$\{X_t\} = IID(0, \sigma^2)$$

- Past values have no bearing/impact on future values
- The best guess of the next value is to predict the mean of the random variable (zero)

Remember

- Uncorrelated is the same as zero covariance
- Independent means joint probability is the product of marginal probabilities
  $p_{X,Y}(x, y) = p_X(x)p_Y(y)$

- Independent is more strict…

An IID time series is a special case of white noise

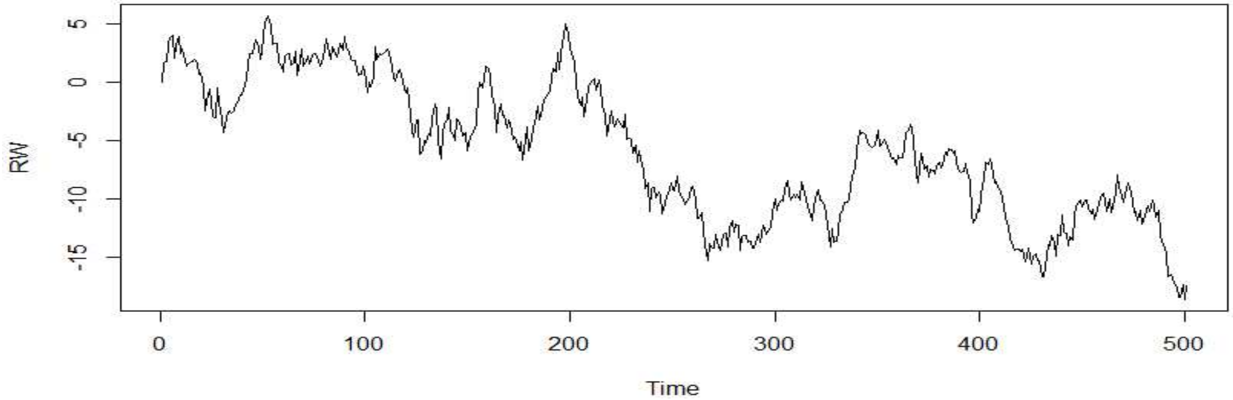
Random Walk

Flip a coin, with p(heads) = ½ , p(tails) = ½, repeatedly

- A series of 1s and -1s (heads = 1, tails = -1), where each subsequent value is independent of previous value
- Mean value over time is 0, finite variance
- An IID, stationary series

Instead of keeping the results of the tosses, take their cumulative sum

- $S_t = \sum_{j=1}^{t} X_j$
- $\{S_t, t = 1, 2, ...\}$ is called a Random Walk, and is non-stationary



Differencing?

Autocorrelation Function

If $\{X_t\}$ is considered to be a stationary time series

- We can then view the autocovariance as a function of one variable: τ
- We call this the autocovariance function (ACVF): $\gamma_X(\tau)$ or $\gamma(\tau)$

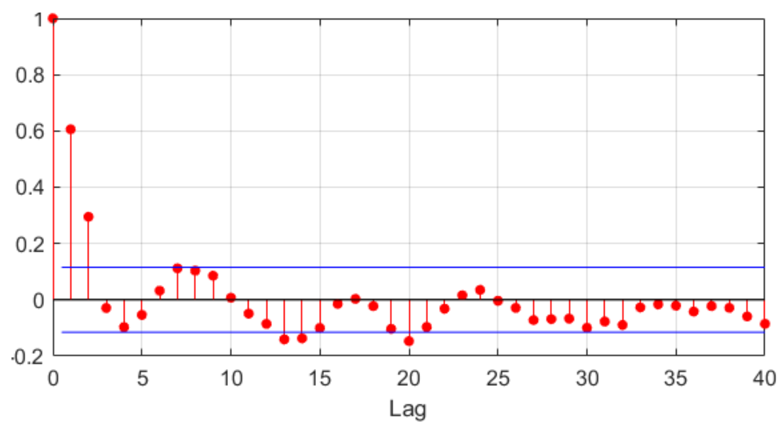
Autocorrelation Function

If $\{X_t\}$ is considered to be a stationary time series

- We can then view the autocovariance as a function of one variable: τ
- We call this the autocovariance function (ACVF): $\gamma_X(\tau)$ or $\gamma(\tau)$

Similarly, the we can define an autocorrelation function (ACF):

$$\rho_X(\tau) = \frac{\gamma_X(\tau)}{\gamma_X(0)} = corr(X_{t+\tau}, X_t) \quad \text{for all } t, \tau$$



Autocorrelation Function

We don't usually know the true generating function

- We get *observed* data

So, for a given set of observations $\{x_1, x_2, \dots x_n\}$ of a time series,

- The sample autocovariance function is

$$\hat{\gamma}_X(\tau) = \frac{1}{n}\sum_{j=1}^{n-\tau}(x_{j+\tau} - \bar{x})(x_j - \bar{x}),\ 0 \le \tau < n, \quad \bar{x} = \frac{1}{\mathrm{n}}\sum_{j=1}^{n} x_j$$

Also: $\hat{\gamma}_X(\tau) = \hat{\gamma}_X(-\tau), \ -n < \tau \le 0,$

- The sample autocorrelation function is

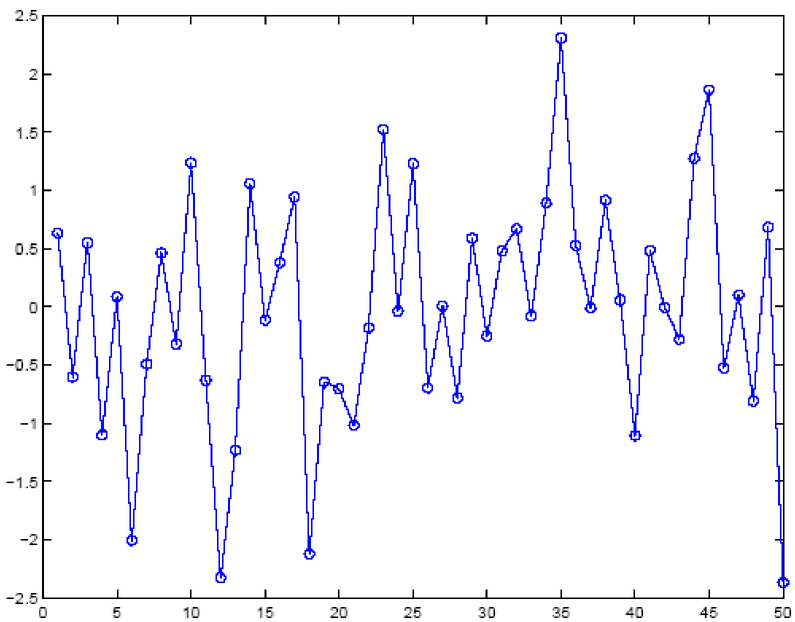
$$\hat{\rho}_X(\tau) = \frac{\hat{\gamma}_X(\tau)}{\hat{\gamma}_X(0)}, \qquad |\tau| < n.$$

Example Autocorrelation

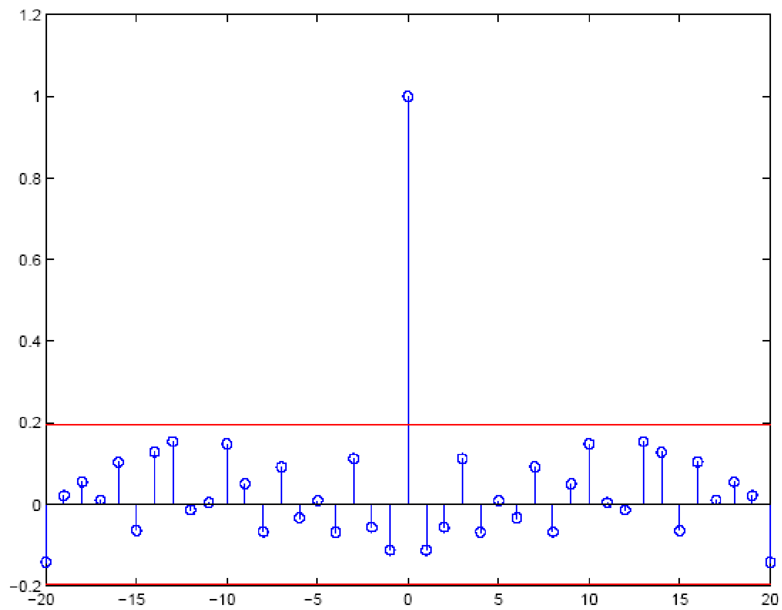
- White Gaussian Noise

  (normally distributed iid)

$$P[X_t \leq x_t] = \Phi(x_t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x_t} e^{-x^2/2} dx.$$

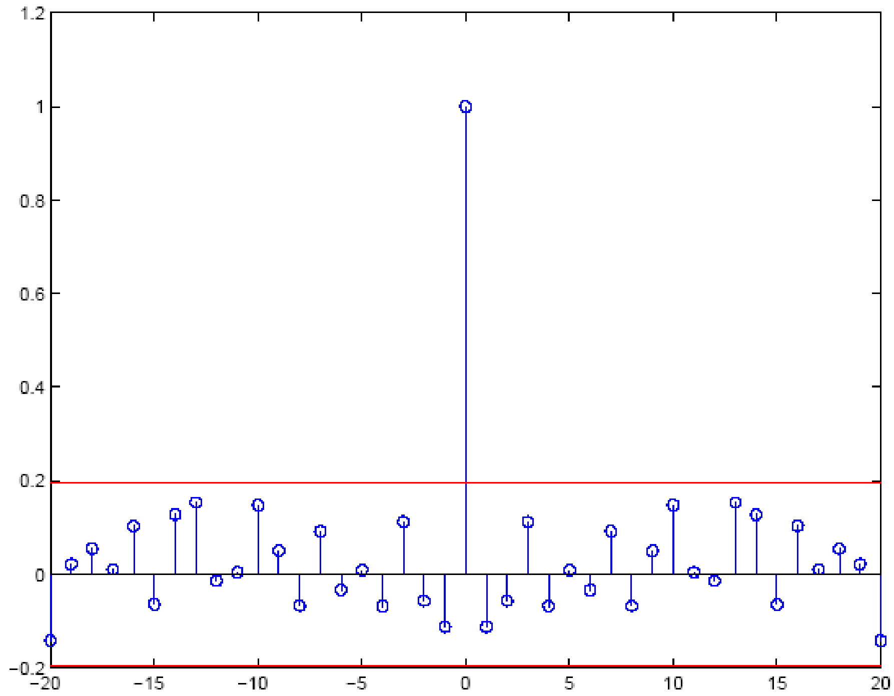


- Corresponding autocorrelation function
  - Note that all but 0 lag are low correlation



Why is it not 0 everywhere else?

Example Autocorrelation

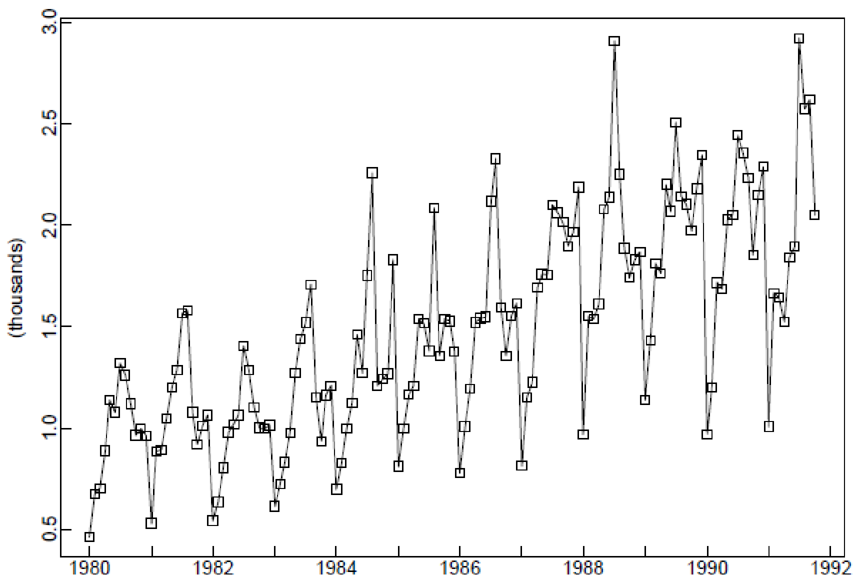


For WGN, we expect the autocorrelation to be close to zero for $\tau \neq 0$

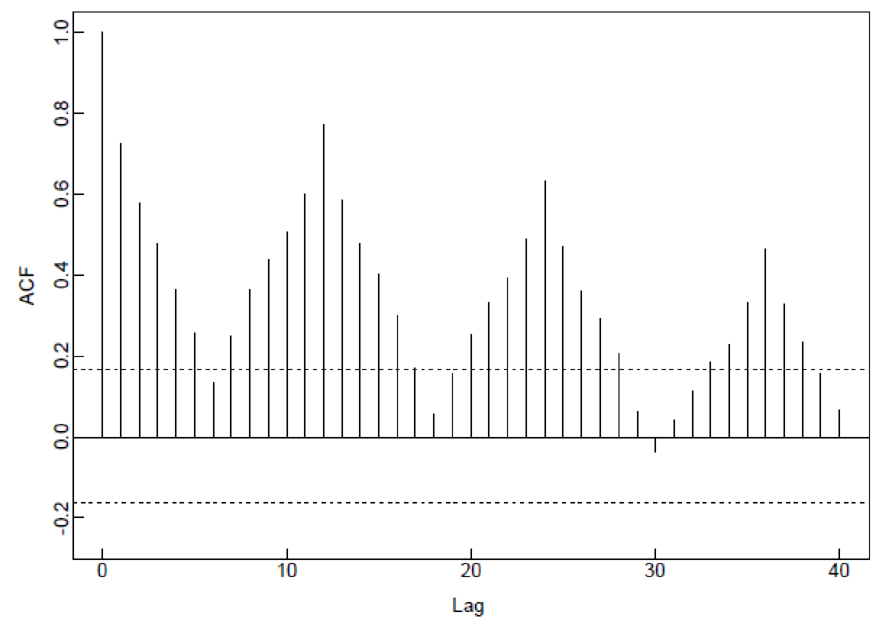
- Due to the Gaussian random variations, ~95% of the spikes should lie within $\pm {}^{2}/_{\sqrt{T}}$, where *T* is the length of the time series.
- If one or more large spikes exceed this, or many more than 5% do, then the series is probably not only white noise.

Example Autocorrelation

- Sample seasonal time series
  - Australian Wine sales



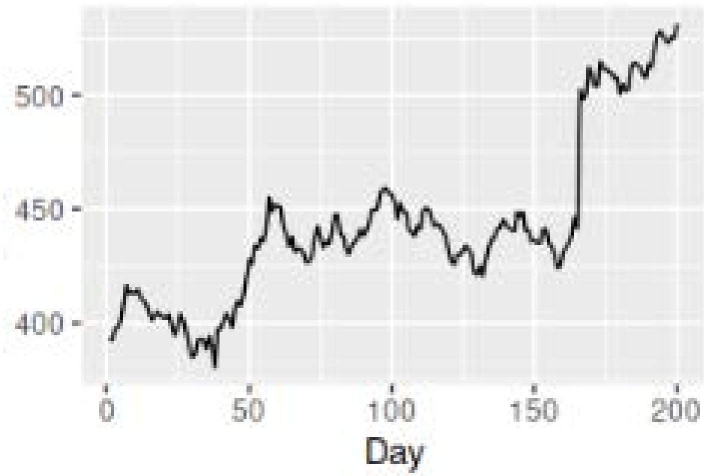
- Corresponding autocorrelation function

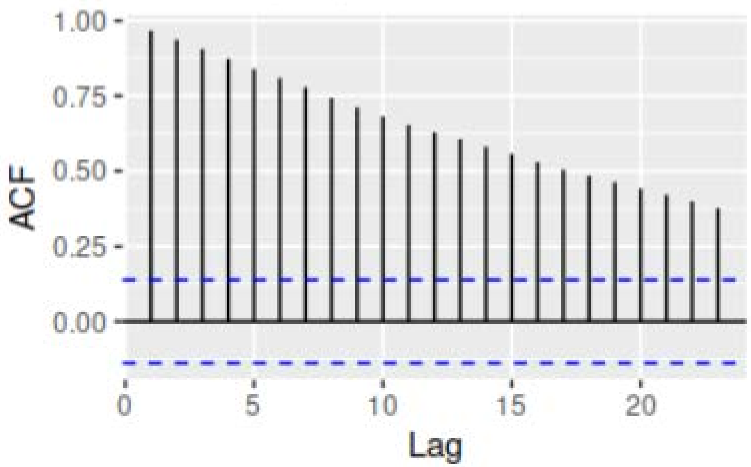


- Note that both seasonality and trend can be detected here (decay with increasing lag)

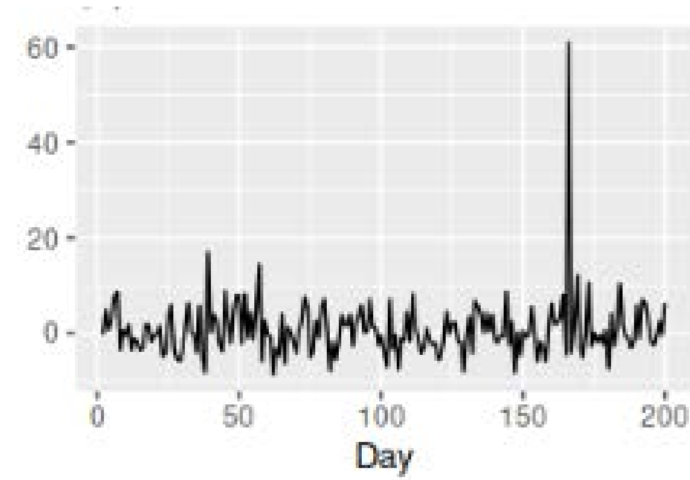
Example Autocorrelation

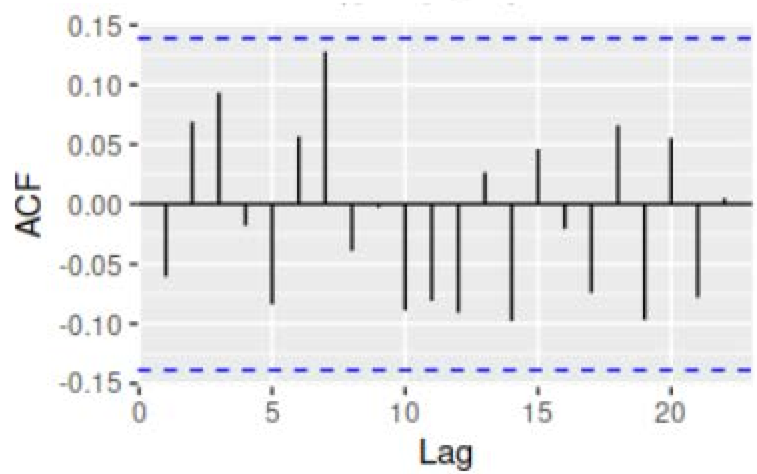
- Original Signal



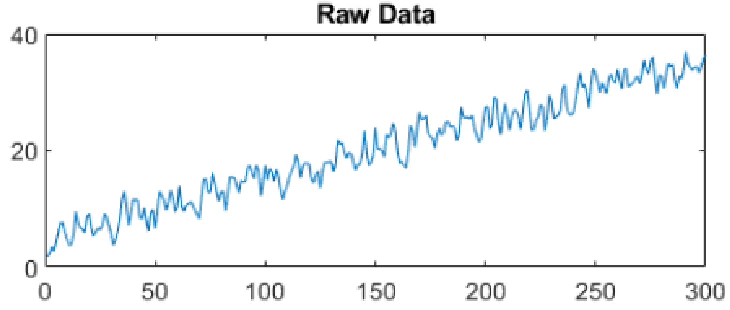


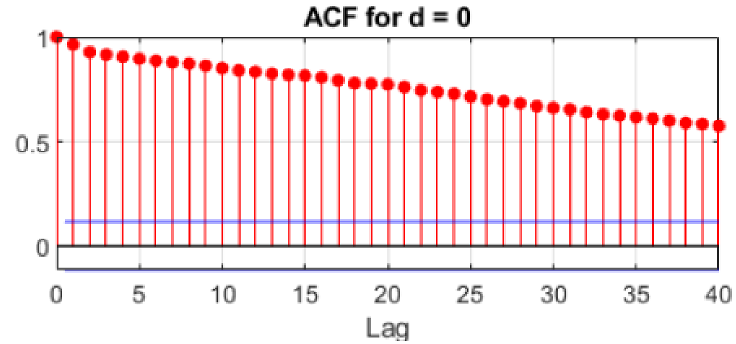
- Differenced Signal





Example Autocorrelation





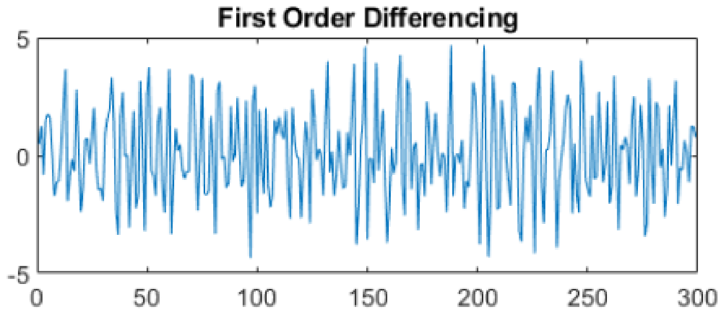
Under Differenced: A series with a trend (that requires estimation or differencing)

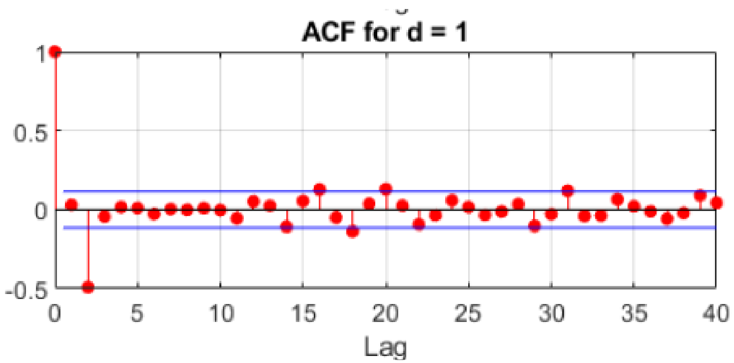
If the series has positive autocorrelations for a high number of lags, it may need to be differenced

Example Autocorrelation

Properly Differenced: You've differenced the right amount of times

If the autocorrelation at lag 1 is 0 or small negative, and the remaining autocorrelations are small and without pattern, you have the right level of differencing

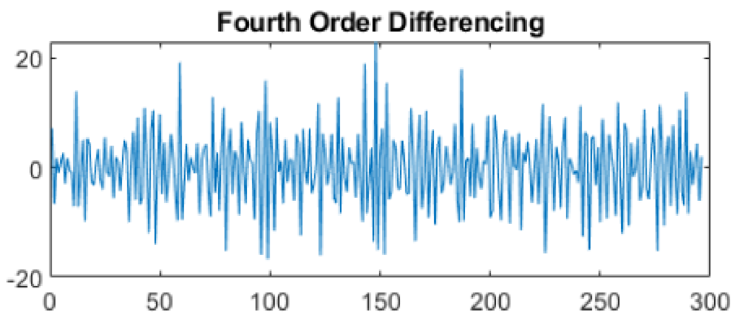


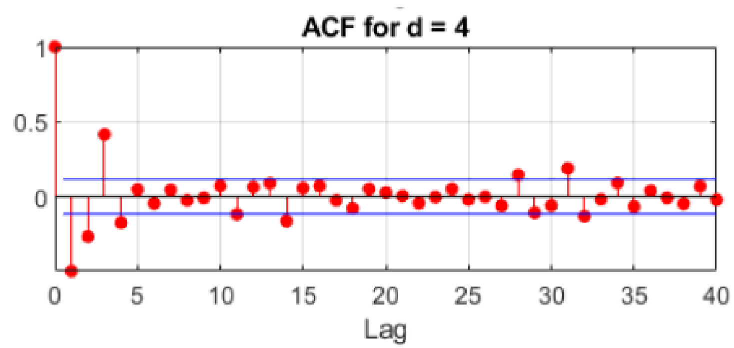


Example Autocorrelation

Over Differenced: You've differenced the signal too many times

If the autocorrelation at lag 1 is < -0.5, the order of differencing may be too high





Lag Plots

In machine learning, we often create cluster plots by plotting one feature vs another.

In time series, it can be equally informative to plot a time series vs a lagged version of itself; a lag plot

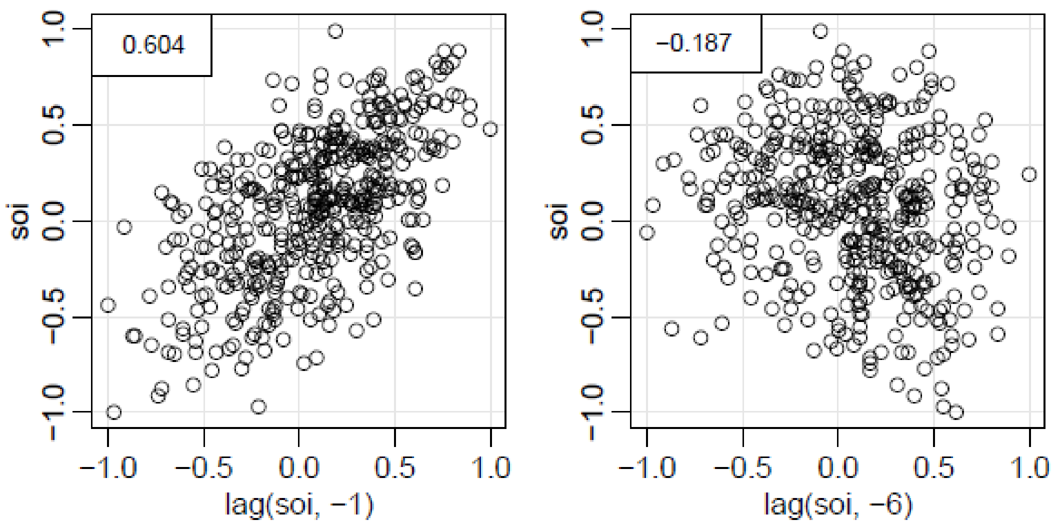
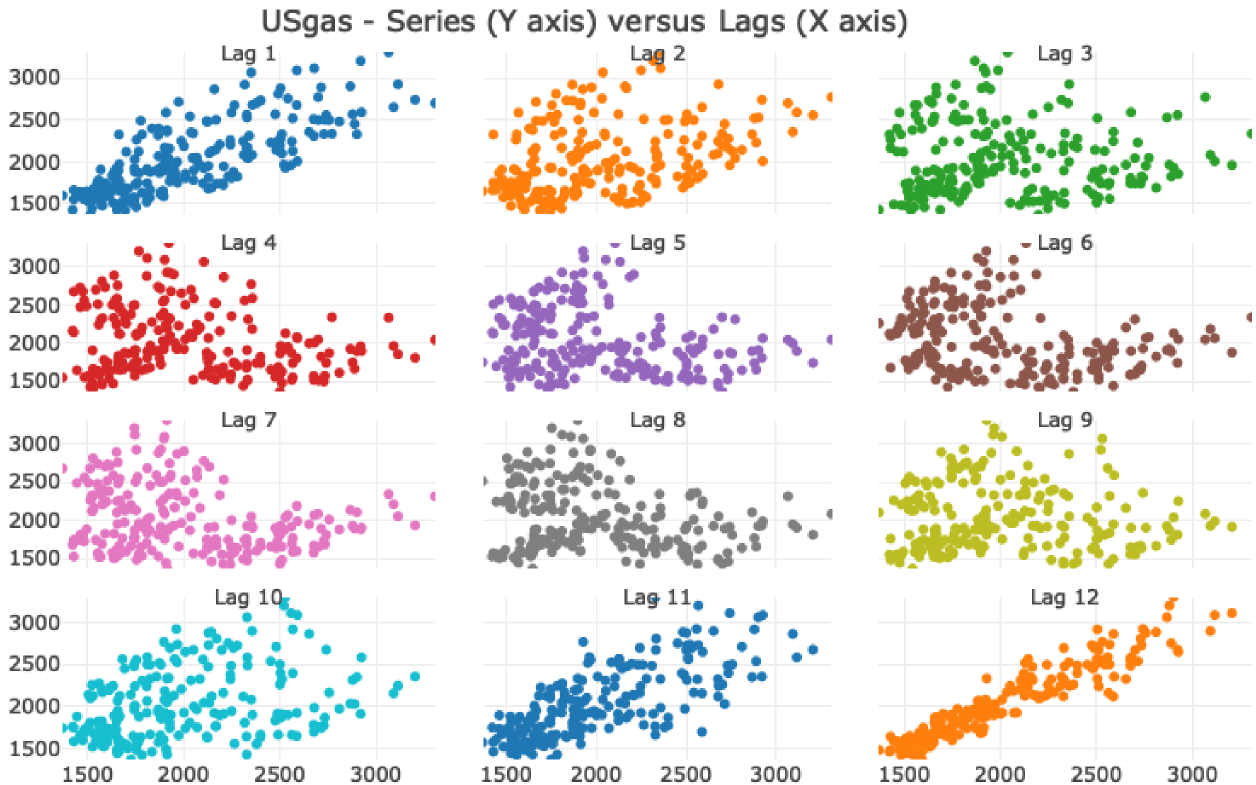


Fig. 1.14. Display for Example 1.25. For the SOI series, the scatterplots show pairs of values one month apart (left) and six months apart (right). The estimated correlation is displayed in the box.

Lag Plots

In machine learning, we often create cluster plots by plotting one feature vs another.

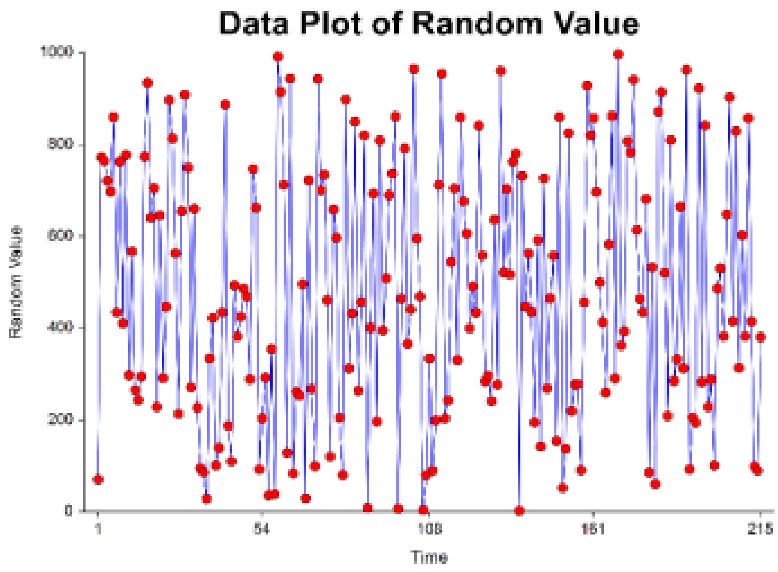
In time series, it can be equally informative to plot a time series vs a lagged version of itself; a lag plot



Lag Plots

In machine learning, we often create cluster plots by plotting one feature vs another.

In time series, it can be equally informative to plot a time series vs a lagged version of itself; a lag plot

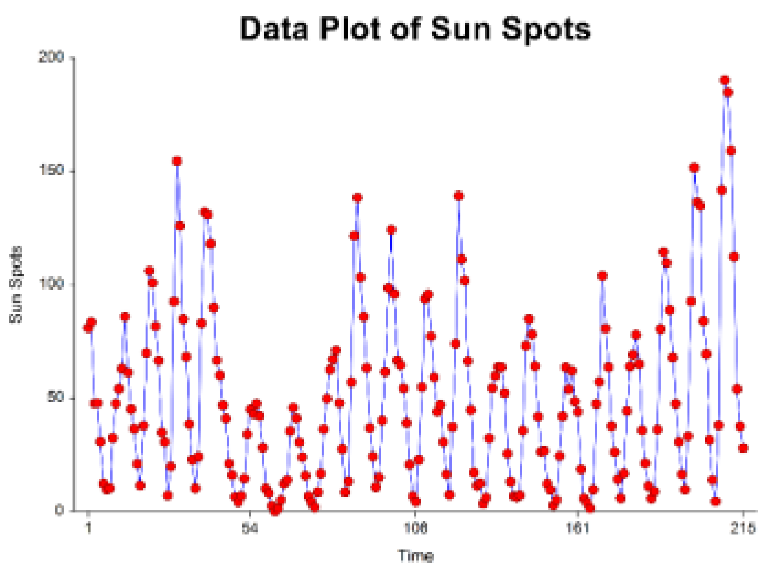


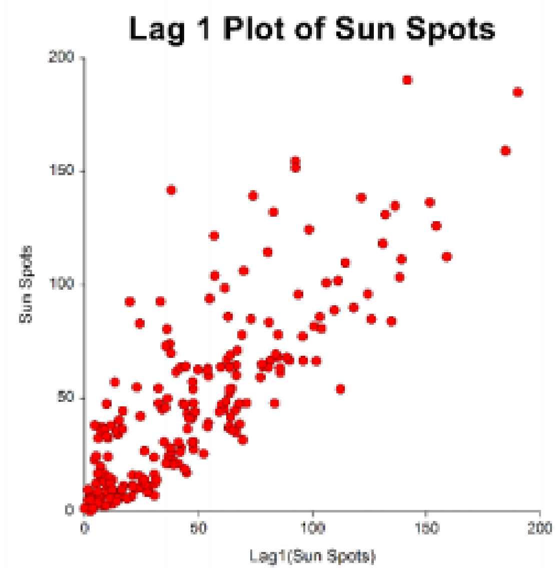
https://ncss-wpengine.netdna-ssl.com/wp-content/themes/ncss/pdf/Procedures/NCSS/Lag_Plots.pdf

Lag Plots

In machine learning, we often create cluster plots by plotting one feature vs another.

In time series, it can be equally informative to plot a time series vs a lagged version of itself; a lag plot



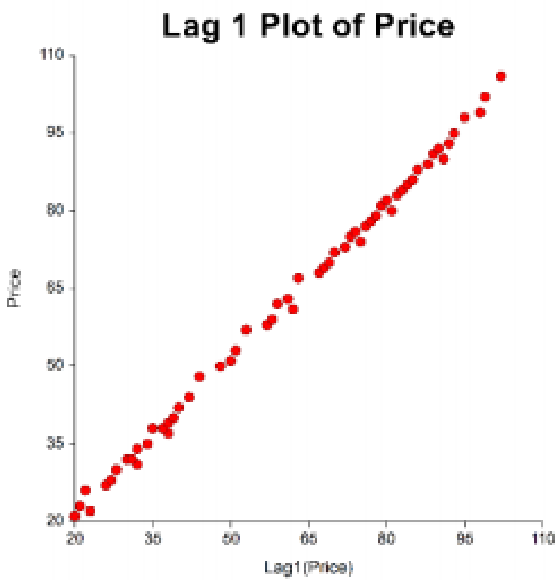


https://ncss-wpengine.netdna-ssl.com/wp-content/themes/ncss/pdf/Procedures/NCSS/Lag_Plots.pdf

Lag Plots

In machine learning, we often create cluster plots by plotting one feature vs another.

In time series, it can be equally informative to plot a time series vs a lagged version of itself; a lag plot

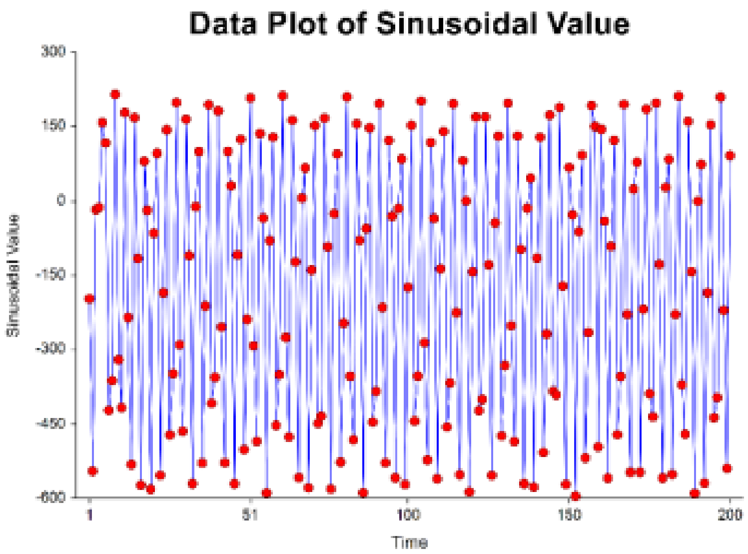


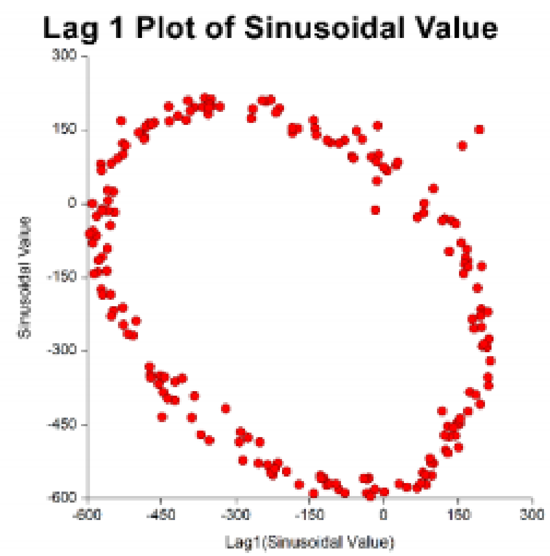
https://ncss-wpengine.netdna-ssl.com/wp-content/themes/ncss/pdf/Procedures/NCSS/Lag_Plots.pdf

Lag Plots

In machine learning, we often create cluster plots by plotting one feature vs another.

In time series, it can be equally informative to plot a time series vs a lagged version of itself; a lag plot



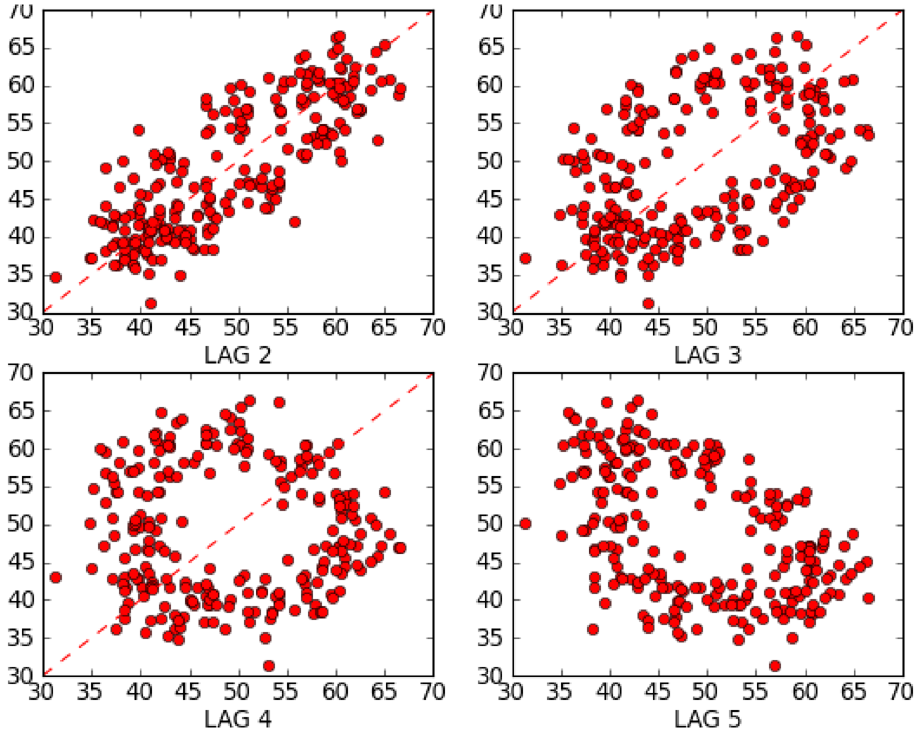


https://ncss-wpengine.netdna-ssl.com/wp-content/themes/ncss/pdf/Procedures/NCSS/Lag_Plots.pdf

Lag Plots

In machine learning, we often create cluster plots by plotting one feature vs another.

In time series, it can be equally informative to plot a time series vs a lagged version of itself; a lag plot



Tests for Stationarity

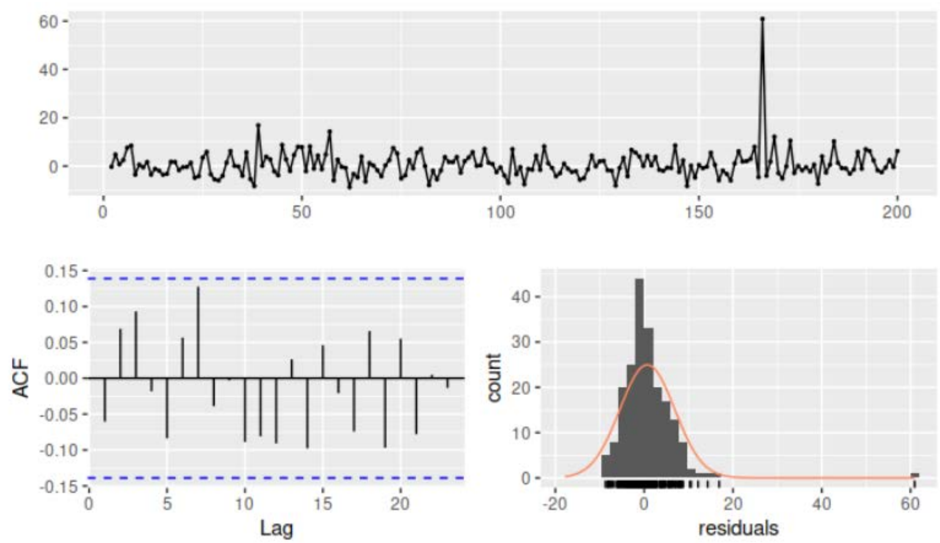
We can conduct more formal tests for stationarity.

Portmanteau tests like the Box-Pierce test (not great) or it's extension, the Ljung-Box:

$$Q = T(T+2)\sum_{\tau=1}^{h}\frac{\rho_X(\tau)^2}{T-\tau}$$

h is max lag checked
T is the number of samples

- Effectively, you sum the autocorrelation values for different lags, and check if they are "small".



Tests for Stationarity

We can also use more statistical (and better) tests like Unit Root Tests

- Augmented Dickey Fuller (ADF) Test
- KPSS test (used to test for trend stationarity)
- Philips Perron Test

A unit root is a characteristics of time series that makes them non-stationary.

- Requires knowledge of autoregression, which we'll see in a bit

For now, the idea of these tests is to test the null hypothesis that the data are stationary.

- We look for evidence that this is false, so small p-values indicated that the signal is NOT yet stationary

ARIMA Modeling

ARIMA modeling is the most classical and widely known approach in time-series analysis.

- AR stands for autoregression models
- I stands for integration (or really, differencing)
- MA stand for moving average

We've already touched on some of these aspects, but we'll consider them here as part of a more formal model.

- We'll use AR models to explain the signal once we've made it stationary

Autoregressive Models

In our previous regression approach, the output depended only on the current value of our independent variable, t.

Autoregressive (AR) models are based on the idea that the output of a series can be explained as a function of its past values.

- The 'order' of the model is defined by how many points back it looks
- AR(p)

$$x_t = c + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots \phi_p x_{t-p} + \varepsilon_t$$

ε_t is assumed to be ~iid $N(0, \sigma^2)$

Autoregressive Models

In our previous regression approach, the output depended only on the current value of our independent variable, t.

Autoregressive (AR) models are based on the idea that the output of a series can be explained as a function of its past values.

- The 'order' of the model is defined by how many points back it looks
- AR(p)

$$x_t = c + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots \phi_p x_{t-p} + \varepsilon_t$$

ε_t are the residuals, assumed to be ~iid $N(0, \sigma^2)$

The simplest AR model is one in which use a linear model to predict the value at a given time based on the value at the previous time

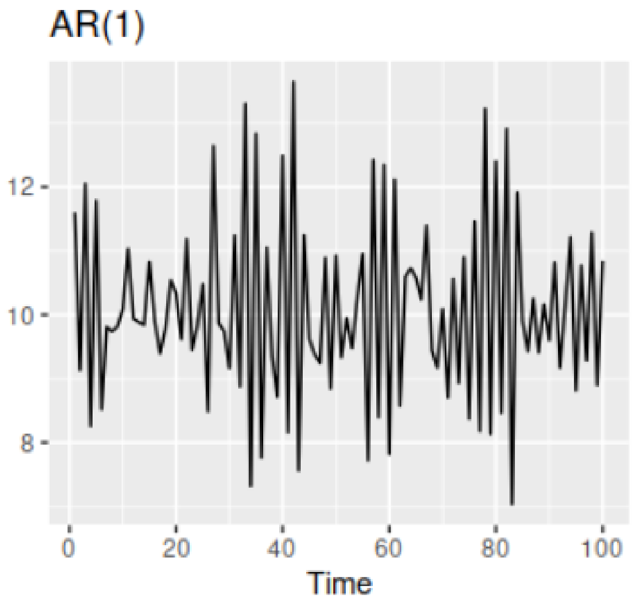
$$x_t = \phi x_{t-1} + \varepsilon_t$$

Autoregressive Models

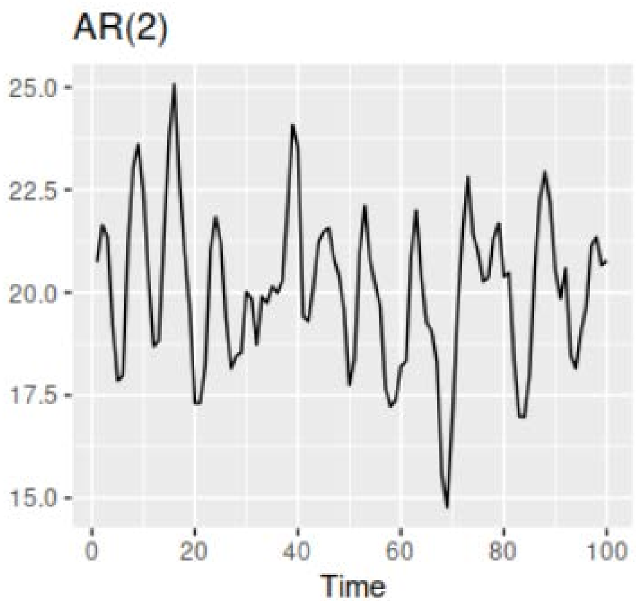
Autoregressive (AR) models are based on the idea that the output of a series can be explained as a function of its past values.

- The ‘order’ of the model is defined by how many points back it looks
- AR(p)

$$x_t = c + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots \phi_p x_{t-p} + \varepsilon_t$$



$x_t = 18 - 0.8x_{t-1} + \varepsilon_t$



$x_t = 8 + 1.3x_{t-1} - 0.7x_{t-2} + \varepsilon_t$

Autoregressive Models

For an AR(1) model: $x_t = c + \phi_1 x_{t-1} + \varepsilon_t$

- when $\phi_1 = 0$, then $x_t = \varepsilon_t$
- when $\phi_1 = 1$, and $c = 0$, then x_t is a random walk
- when $\phi_1 = 1$, and $c = 1$, then x_t is a random walk with drift
- when $\phi_1 < 0$, x_t will tend to oscillate around the mean

For stationary AR models, we need some constraints on the ϕ:

- For AR(1), $x_t = c + \phi_1 x_{t-1} + \varepsilon_t$

$$-1 < \phi_1 < 1$$

- For AR(2), $x_t = c + \phi_1 x_{t-1} + \phi_2 x_{t-2} + \varepsilon_t$

$$-1 < \phi_2 < 1 \;\; and \;\; \phi_1 + \phi_2 < 1 \;\; and \;\; \phi_2 - \phi_1 < 1$$

Autoregressive Models

Examples of 1st order AR models with different ϕ_1

Side note:

- The autocorrelation function of an AR(1) process works out to be :

$$\rho(h) = \frac{\gamma(h)}{\gamma(0)} = {\phi_1}^h$$

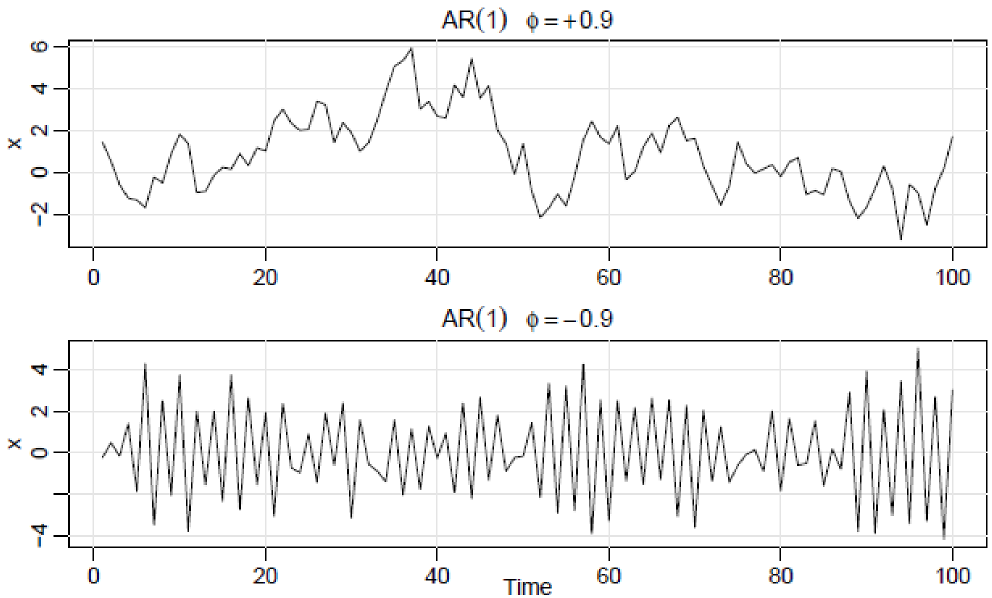


Fig. 3.1. Simulated AR(1) models: ϕ = .9 (top); ϕ = −.9 (bottom).

So, ϕ_1 is the *"slope"* in the AR(1) model and also the lag 1 autocorrelation.

Autoregressive Models

Examples of 1st order AR models with different ϕ_1

When $\phi_1 = 0.9$, points next to each other in time are positively correlated

- Successive points will tend to be close in value; smooth curve

When $\phi_1 = -0.9$, points next to each other in time are negatively correlated

- But points two points apart are positively correlated.
- Tends to jump between +ve to -ve

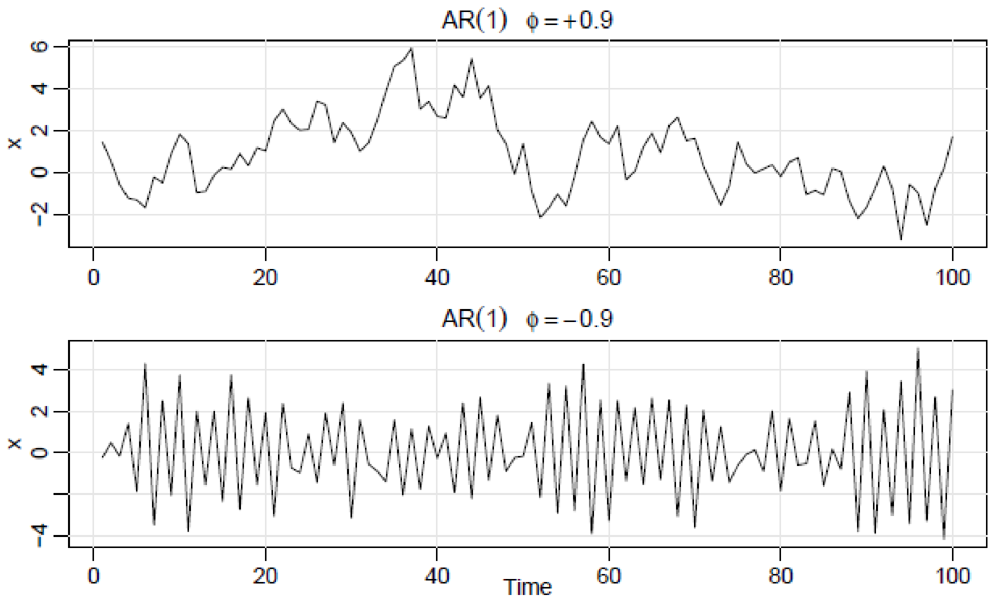
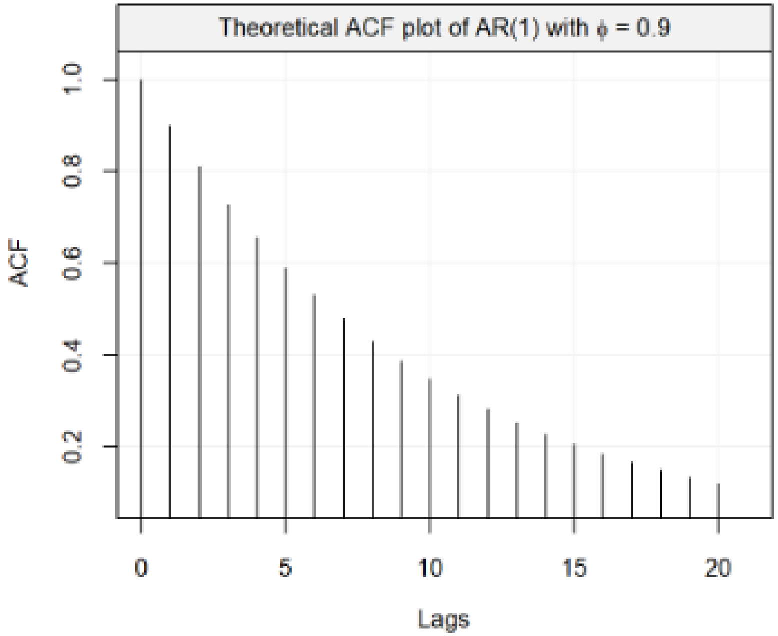


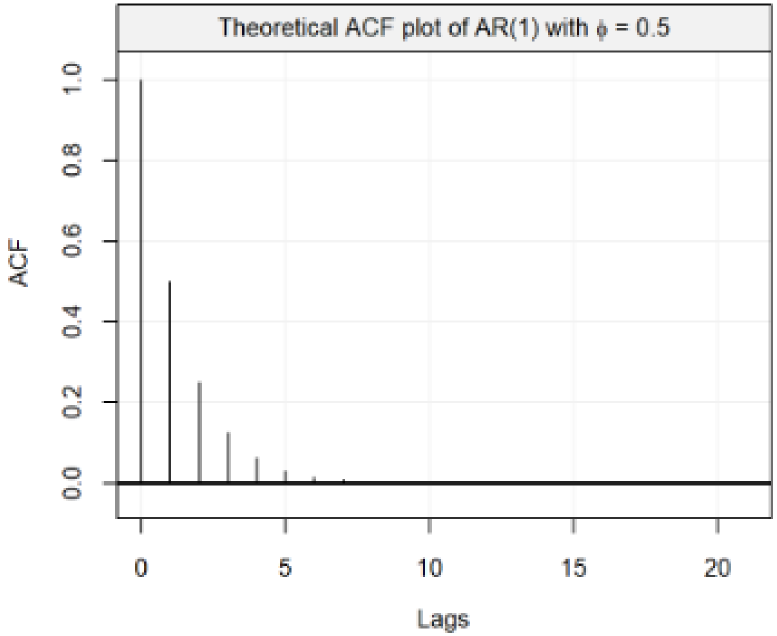
Fig. 3.1. Simulated AR(1) models: ϕ = .9 (top); ϕ = −.9 (bottom).

Autoregressive Models

The ACF of an AR(1) model with a positive value for ϕ_1 will exponentially decrease to 0, as the lag increases

- Value at lag 1 is ϕ_1, value at lag 2 is ${\phi_1}^2$, etc…



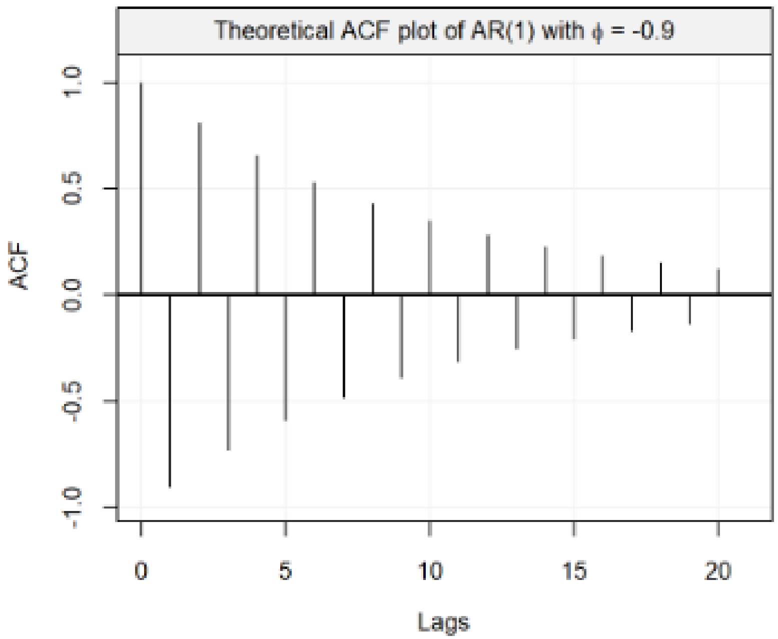


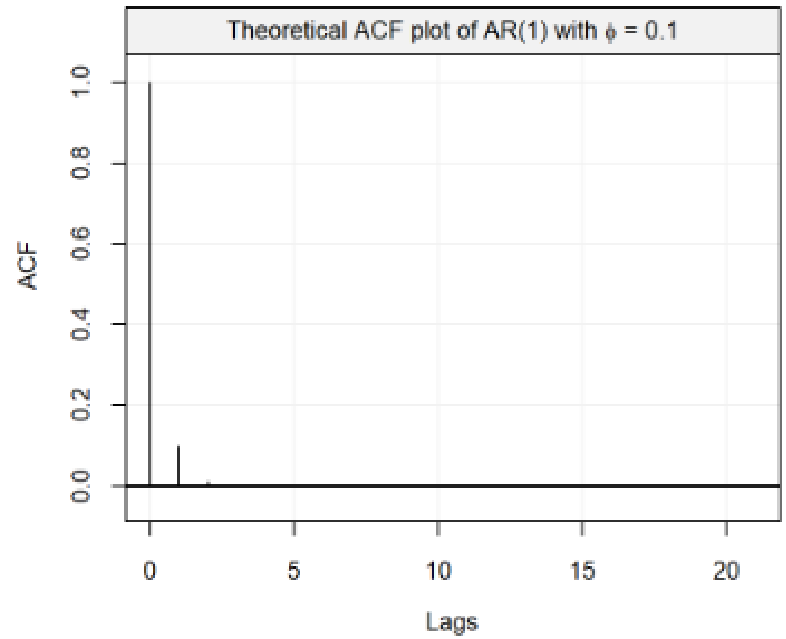
https://online.stat.psu.edu/stat510/lesson/1/1.2

Autoregressive Models

The ACF of an AR(1) model with a negative value for ϕ_1 will exponentially decrease to 0

- Smaller $|\phi_1|$, faster decay
- Value at lag 1 is ϕ_1, value at lag 2 is ${\phi_1}^2$, etc… so it oscillates around 0





https://online.stat.psu.edu/stat510/lesson/1/1.2

Partial Autocorrelation Function

The Partial Autocorrelation Function (PACF) is similar to the ACF but instead measures the incremental benefit of adding an additional lag.

- Remember that ACF measures the relationship between lagged values x_t and x_{t-k}
- But, if x_t and x_{t-1} are correlated, then x_{t-1} and x_{t-2} must also be correlated
- Also, x_t and x_{t-2} might be correlated, but we don't know if it's only because of their connection to x_{t-1} or not.
- The PACF overcomes this by measuring the relationship between x_t and x_{t-k} AFTER removing the effects of all lags in between (1, 2… k-1)

Partial Autocorrelation Function

The Partial Autocorrelation Function (PACF) is similar to the ACF but instead measures the incremental benefit of adding an additional lag.

- Remember that ACF measures the relationship between lagged values x_t and x_{t-k}
- But, if x_t and x_{t-1} are correlated, then x_{t-1} and x_{t-2} must also be correlated
- Also, x_t and x_{t-2} might be correlated, but we don't know if it's only because of their connection to x_{t-1} or not.
- The PACF overcomes this by measuring the relationship between x_t and x_{t-k} AFTER removing the effects of all lags in between (1, 2… k-1)

$$\emptyset_{11} = corr(x_{t+1}, x_t) = \rho(1)$$

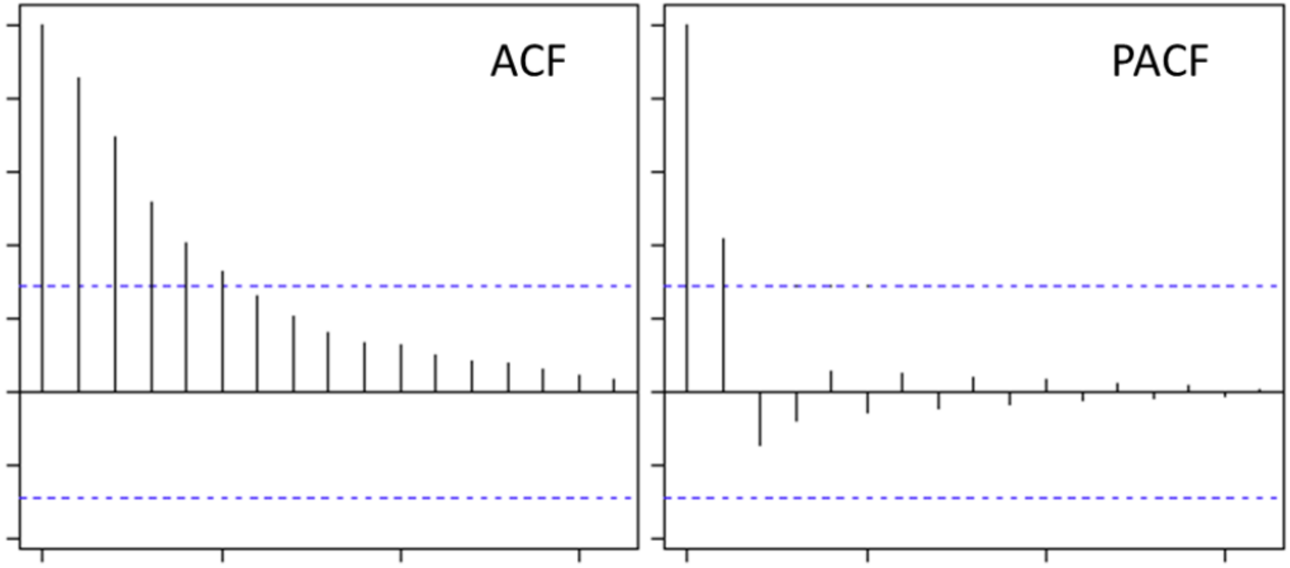
$$\emptyset_{hh} = corr(x_{t+h} - \hat{x}_{t+h}, x_t - \hat{x}_t), h \geq 1$$

$\hat{x}_{t+h}$ is the regression of x_{t+h} onto $\{x_{t+h-1}, x_{t+h-2}, \dots, x_{t+1}\}$

Partial Autocorrelation Function

The Partial Autocorrelation Function (PACF) is similar to the ACF but instead measures the incremental benefit of adding an additional lag.

- The 1st ACF and 1st PACF are the same because (there is no intermediate effect to remove)
- The cutoff point of the PACF (it drops off, or below the significance level) determines the order of an AR model (number of terms, p)
- e.g. An AR(2) process

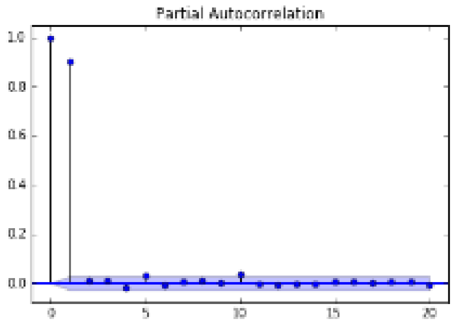


Partial Autocorrelation Function

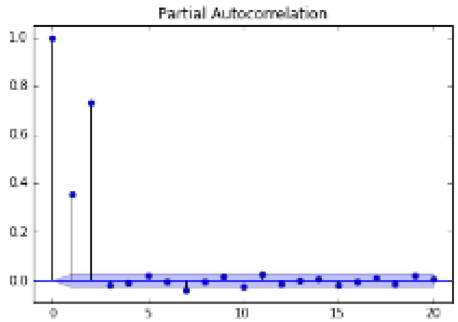
The Partial Autocorrelation Function (PACF) is similar to the ACF but instead measures the incremental benefit of adding an additional lag.

- E.g. What is the benefit of adding a 4th lag element, when you already have 3

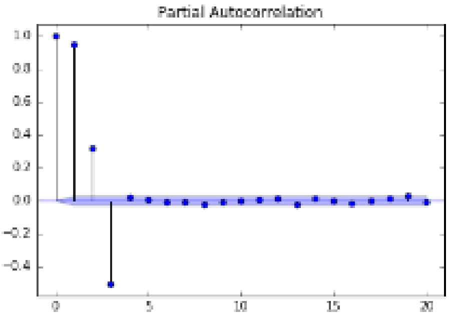
- AR(1)



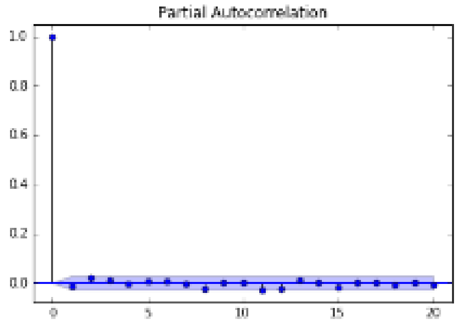
- AR(2)



- AR(3)



- White Noise



Autoregressive Features

We've seen how the parameters of an AR model can be changed to describe very different looking time series.

- We can also use those parameters (the coefficients) to describe the behaviour/characteristics of a signal (the order is also informative)
- Can be used as features to classify time series data

Published: 13 June 2016

Classification of EEG Signals Based on Autoregressive Model and Wavelet Packet Decomposition

Yong Zhang, Bo Liu, Xiaomin Ji & Dan Huang

Neural Processing Letters **45**, 365–378(2017) | Cite this article

ELSEVIER

Mathematical and Computer Modelling
Volume 10, Issue 10, 1988, Pages 793-794

Mathematical modelling reports

Autoregressive modeling of surface EMG and its spectrum with application to fatigue

Omry Paiss, Gideon F. Inbar

Show more

+ Add to Mendeley   Share   Cite

Article

Feature Extraction and Selection for Myoelectric Control Based on Wearable EMG Sensors

Angkoon Phinyomark [1,*], Rami N. Khushaba [2] and Erik Scheme [1]

1 Institute of Biomedical Engineering, University of New Brunswick, Fredericton, NB E3B 5A3, Canada; escheme@unb.ca
2 Faculty of Engineering and Information Technology, University of Technology, Sydney 2007, NSW, Australia; rami.khushaba@uts.edu.au
* Correspondence: aphinyom@unb.ca; Tel.: +1-506-453-4966

Received: 14 March 2018; Accepted: 16 May 2018; Published: 18 May 2018

check for updates

Moving Average Models

Autoregressive models assume that the data could be explained by a linear combination of the previous p data points

$$x_t = c + \ \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots \phi_p x_{t-p} + \varepsilon_t$$

Moving average models assume that the data can be explained by a linear combination of the past q residuals, ε_t

- MA model of order q, MA(q)

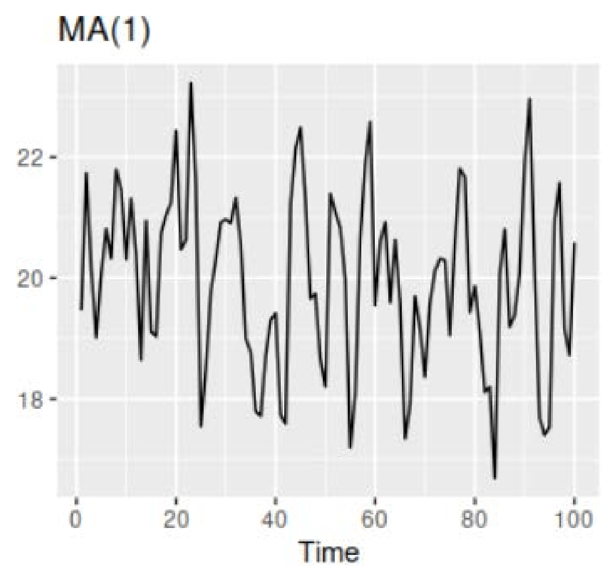
$$x_t = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots \theta_p \varepsilon_{t-q}$$

Moving Average Models

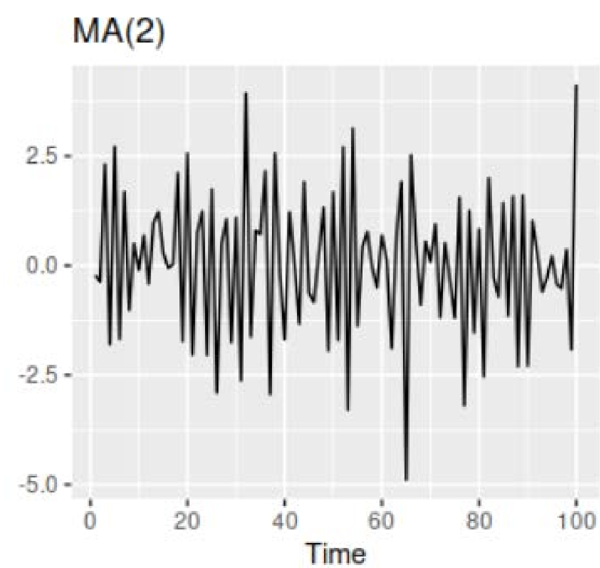
Moving average models assume that the data can be explained by a linear combination of the past q residuals, ε_t

- MA model of order q, MA(q)

$$x_t = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots \theta_p \varepsilon_{t-q}$$



$x_t = 20 + \varepsilon_t + 0.8\varepsilon_{t-1}$



$x_t = \varepsilon_t - \varepsilon_{t-1} + 0.8\varepsilon_{t-2}$

ε_t is WGN(0,1)

Moving Average Models

Examples of 1st order MA models with different θ

- The autocorrelation function of an MA(1) process works out to be :

$$\rho(h) = \begin{cases} \dfrac{\theta}{(1+\theta^2)} & h = 1 \\ 0 & h > 1 \end{cases}$$

For AR(1), $\rho(h) = {\phi_1}^h$

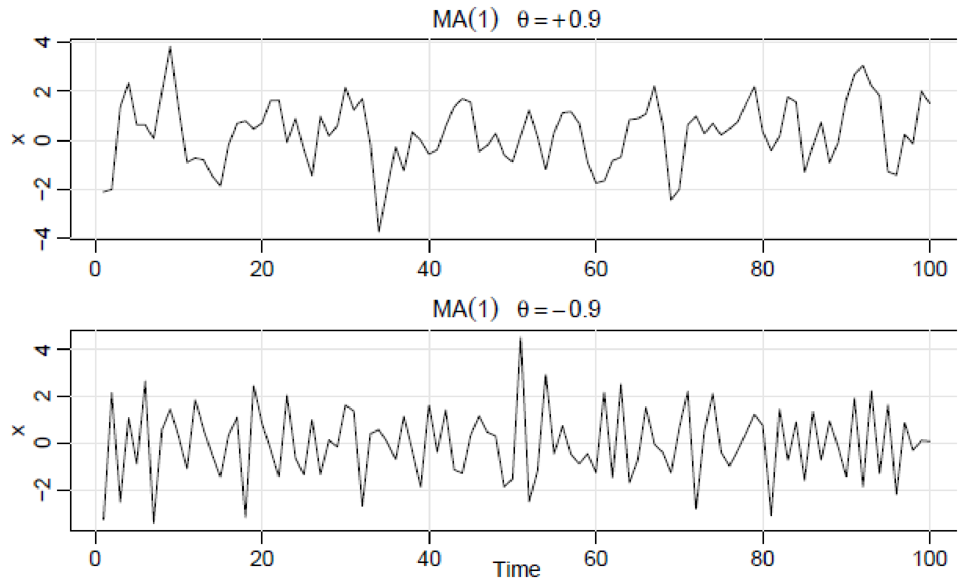


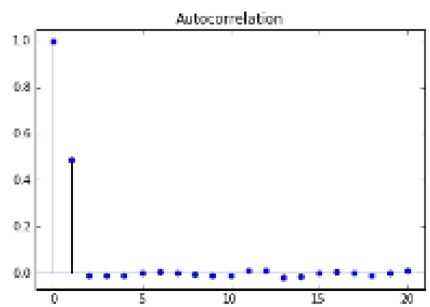
Fig. 3.2. Simulated MA(1) models: $\theta = .9$ (top); $\theta = -.9$ (bottom).

So, x_t is correlated with x_{t-1} but not with $x_{t-2}, x_{t-3}, \ldots$

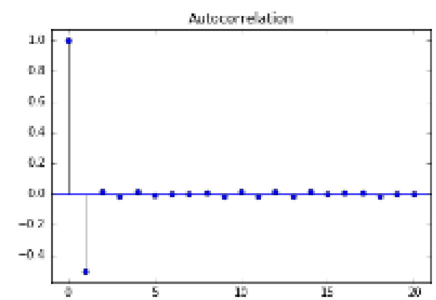
Moving Average Models

Looking at the Autocorrelation Function (ACF) of a MA(1) model, we see the effect of different θ

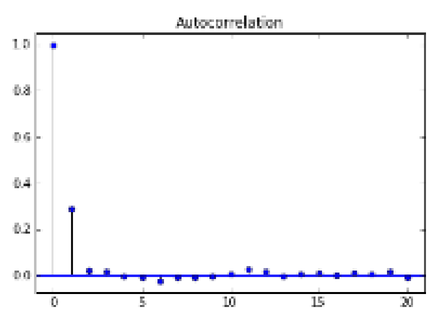
- $\theta = 0.9$



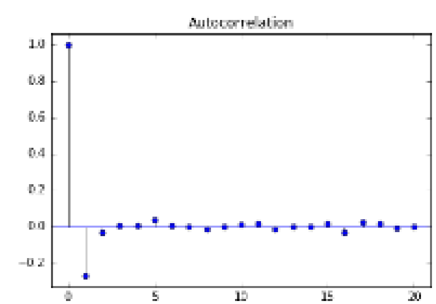
- $\theta = -0.9$



- $\theta = 0.5$



- $\theta = -0.5$



All lags > 1 are ~0 because of the 1st order MA(1) model

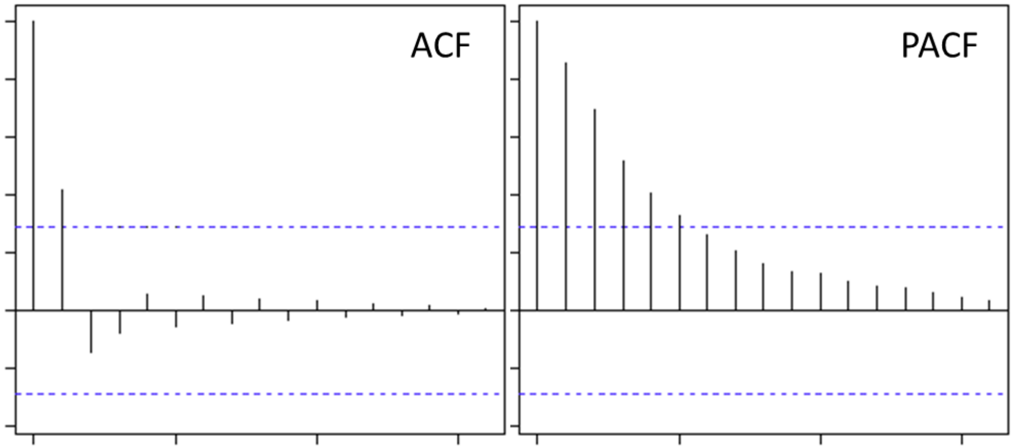
Moving Average Models

For a 2nd order MA(2) model,

- The autocorrelation is

$$\rho(1) = \frac{\theta_1 + \theta_1\theta_2}{1 + {\theta_1}^2 + {\theta_2}^2} \qquad \rho(2) = \frac{\theta_2}{1 + {\theta_1}^2 + {\theta_2}^2} \qquad \rho(h \geq 3) = 0$$

The model order of an MA(q) model is determined by the cutoff point of the ACF



Remember that the cutoff of the AR(p) model was determined using the *PACF*

Moving Average Models

Any stationary AR(p) model can be written as an MA(∞):

- A 1st order AR model, AR(1) is: $x_t = \phi x_{t-1} + \varepsilon_t$

- But x_{t-1} itself depends only on x_{t-2}…

$$x_t = \phi\{\phi x_{t-2} + \varepsilon_{t-1}\} + \varepsilon_t \qquad = \phi^2 x_{t-2} + \phi\varepsilon_{t-1} + \varepsilon_t$$

…

$$x_t = \phi^k x_{t-k} + \sum_{j=0}^{k-1} \phi^j\, \varepsilon_{t-j}$$

- If $|\phi|<1$, and k gets large, then AR(1) is effectively a linear process

$$x_t = \sum_{j=0}^{\infty} \phi^j\, \varepsilon_{t-j}$$

A MA(∞) model!

The reverse is not always true, but can be given some restrictions…

ARIMA Models

We can also combine AR(*p*), I(*d*) and MA(*q*) models to form one big Autoregressive Integrated Moving Average (ARIMA) Model

ARIMA models are described as ARIMA(*p*,*d*,*q*)

- *p* is the order of the AR components
- *d* is the order of the differencing
- *q* is the order of the MA components

A time series is ARIMA(*p*,*0*,*q*), or ARMA(*p*,*q*) if it can be modeled as:

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + \cdots \phi_p x_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots \theta_q \varepsilon_{t-q}$$

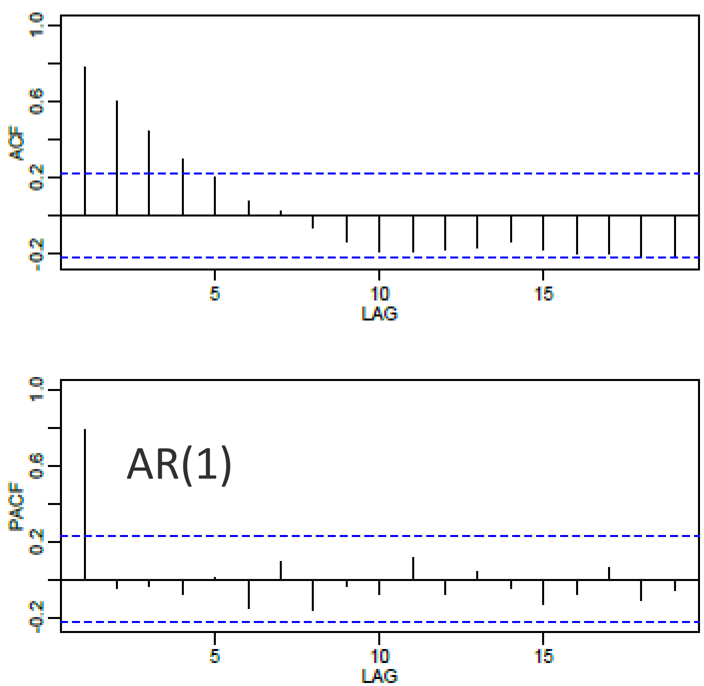
ARIMA model:

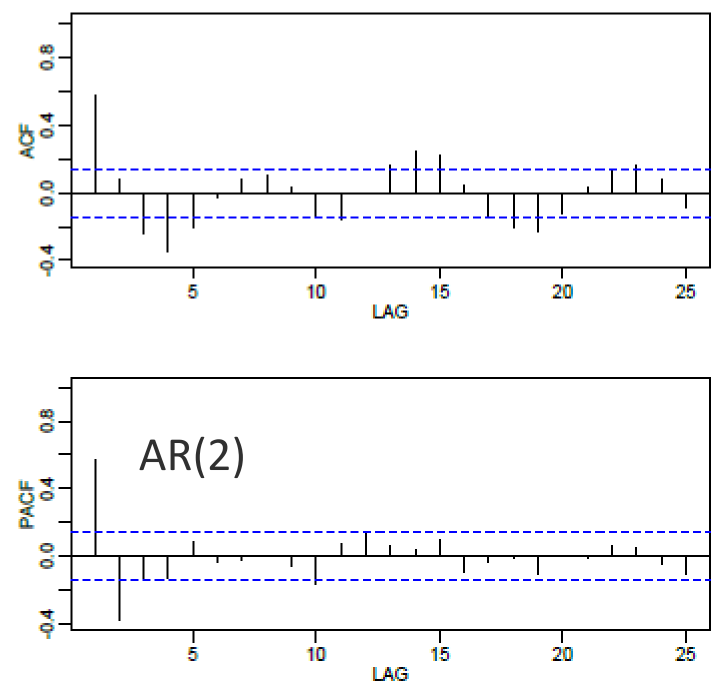
- Predicted x_t = Constant + a linear combination of lags of x (up to *p*) + a linear combination of lagged forecast errors (up to *q*)

ARIMA Model Order Selection

The data may be described by a p^{th} order AR model if in the ACF and PACF plots of the stationary data:

- the ACF decays exponentially or sinusoidally
- there is a significant spike at lag p in the PACF, but not beyond

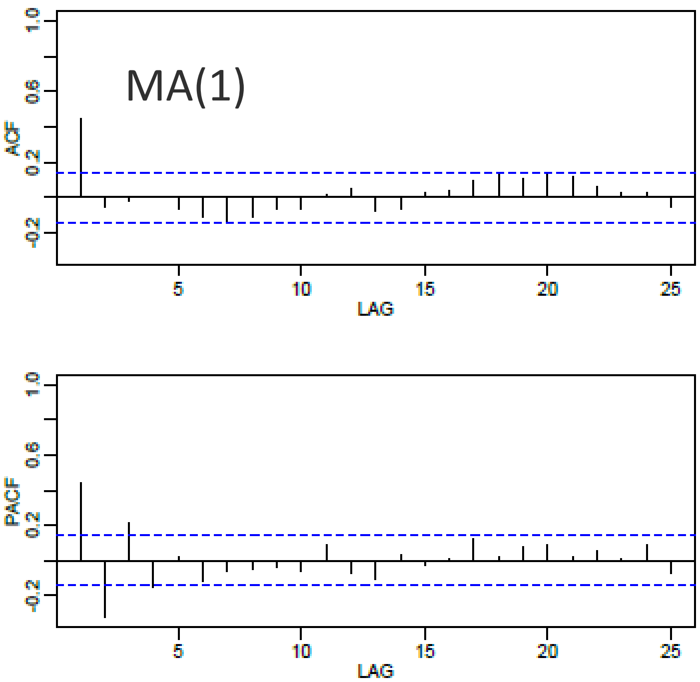




ARIMA Model Order Selection

Conversely, the data may be described by a q^{th} order MA model if in the ACF and PACF plots of the stationary data:

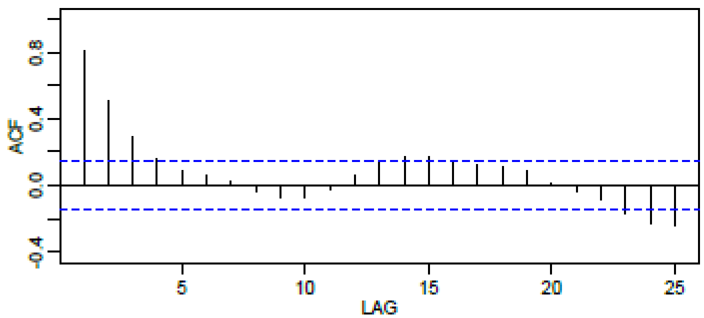
- the ACF decays exponentially or sinusoidally
- there is a significant spike at lag q in the ACF, but not beyond

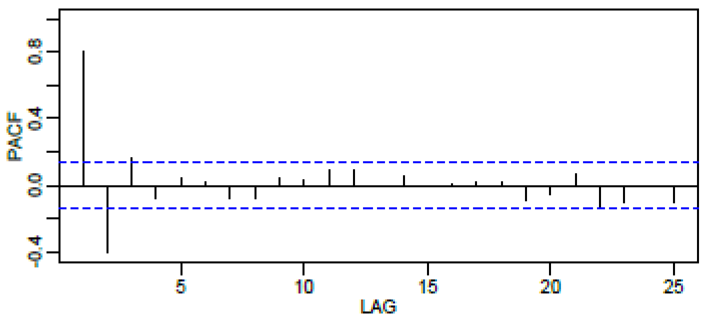


ARIMA Model Order Selection

The data may be described by a p^{th} order AR model AND a q^{th} order MA if in the ACF and PACF plots of the stationary data:

- both the ACF and PACF decay exponentially or sinusoidally
- Determination of p and q is tricky – no clear cutoff (some trial and error)





Model	ACF	PACF
AR(p) or ARMA(p,0)	Decays towards zero	Cuts off at p
MA(q) or ARMA(0,q)	Cuts off at q	Decays towards zero
ARMA(p,q)	Decays towards zero	Decays towards zero

ARIMA Model Order Selection

Model	ACF	PACF
AR(*p*) or ARMA(*p*,0)	Decays towards zero	Cuts off at *p*
MA(*q*) or ARMA(0,*q*)	Cuts off at *q*	Decays towards zero
ARMA(*p*,*q*)	Decays towards zero	Decays towards zero

Model	ACF	PACF
Non-Stationary	Doesn't Decay	Doesn't Decay
White noise (done)	All lags are ~0	All lags are ~0

Seasonal ARIMA

An assumption of applying the AR and MA components of ARIMA is that we have first stationarized the model using differencing

- as we saw earlier, this may not always remove seasonal effects
- we've seen we can remove it prior to modeling with ARIMA, however, it is sometimes more convenient to do it all at once

Seasonal Autoregressive Integrated Moving Average (SARIMA or Seasonal ARIMA) models explicitly include a seasonal component.

- Three new hyperparameters: P, D, and Q for the seasonal component
- A new parameter, m, for the period of the seasonality
- Written as

  SARIMA(p,d,q)(P,D,Q)m

Seasonal ARIMA

An assumption of applying the AR and MA components of ARIMA is that we have first stationarized the model using differencing

- as we saw earlier, this may not always remove seasonal effects
- we've seen we can remove it prior to modeling with ARIMA, however, it is sometimes more convenient to do it all at once

Seasonal Autoregressive Integrated Moving Average (SARIMA or Seasonal ARIMA) models explicitly include a seasonal component.

- Three new hyperparameters: *P*, *D*, and *Q* for the seasonal component
- A new parameter, *m*, for the period of the seasonality
- Written as

  SARIMA(*p*,*d*,*q*)(*P*,*D*,*Q*)*m*

P, *D*, and *Q* are the same as their former versions, but applied on the order of *m*, the seasonal period

- e.g. a *P*=1 model would refer back to $(t - 1m)$ and $(t - 2m)$

Q&A