Parametric Statistics

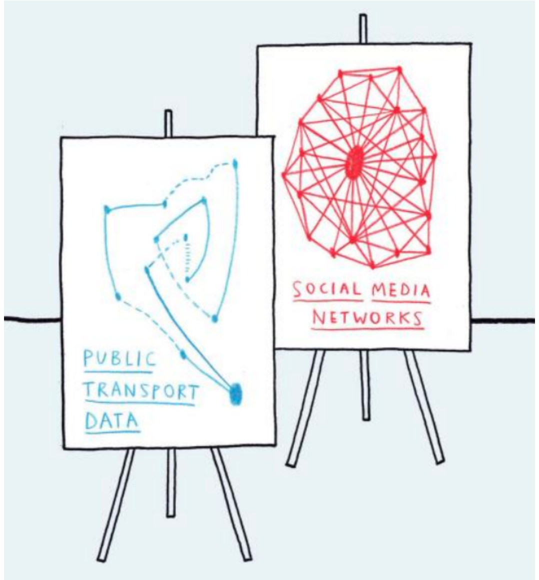
Week 12 - Time Series

Juan Carlos Medina Serrano

Technische Universität München
Hochschule für Politik
Political Data Science

Munich, 22. January 2020

political data science
https://politicaldatascience.blogspot.de



Time Series

Time series: A variable measured sequentially in time.

An important feature is that the observations close together in time are correlated!

This means if we model a series as a linear model, the errors would be autocorrelated.

Important: Time series change depending on the **sampling interval**. Think year, month, day. The plots will look very different.

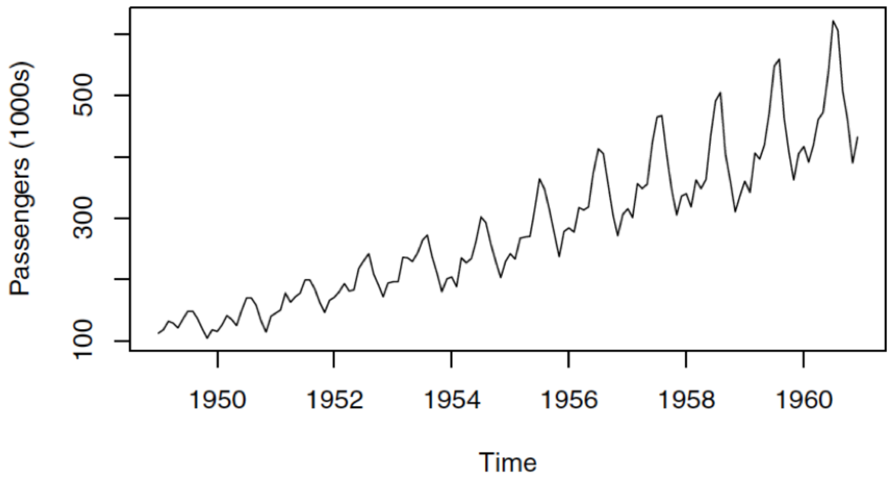
Time series are very important in economics, finance and somehow also for politics.

Trend

Does your time series have a trend?

Trend: A systematic change that is not periodic. May be linear or non-linear.

Example Number of passengers flying in the US



Trend

Normally trends are obvious by plotting the time series. However, it is better to use a statistical test:

Cox and Stuart trend test:

$$H0 : \textit{There is no Trend}$$

$$H1 : \textit{There is a Trend}$$

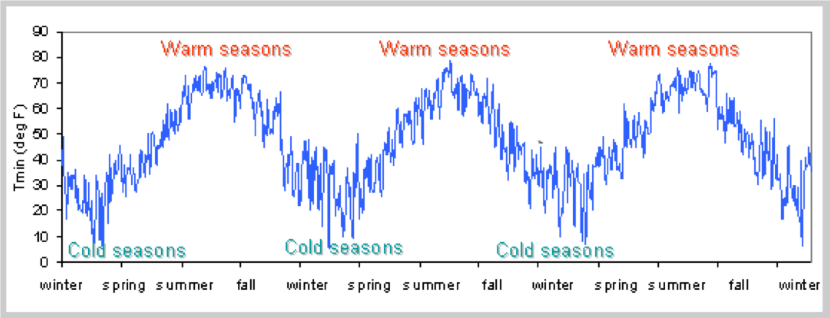
Seasonality

Does your time series have seasonality?

Seasonality: There is a repeating pattern in your time series.

Often happens with monthly data. For example, toy sales per month for several years. December will have bigger values than January each year.

Example The weather for three years:



Seasonality

Normally seasonality can be seen by plotting the time series. However, it is better to use a **ACF plot**.

We will learn about it later...

Decomposition of Time Series

To compare different time series, it is important to remove the trend and seasonality.

There are different models to represent a time series, the most common are:

Additive model: We assume that seasonality and trend are not related

$$x_t = m_t + s_t + e_t$$

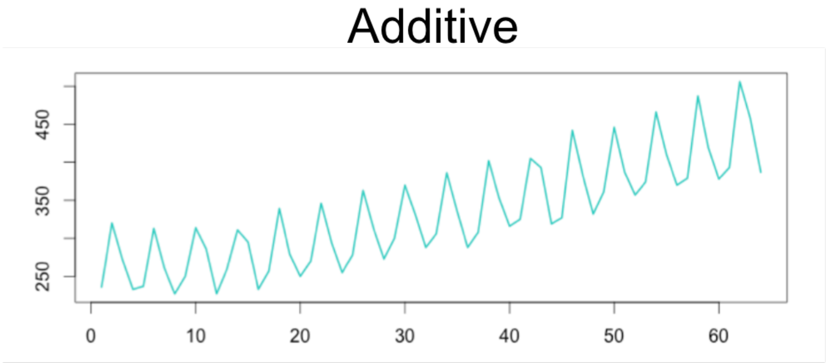
where m is the trend, s is the seasonality and e is the remaining random error term. The subscript t refers to time.

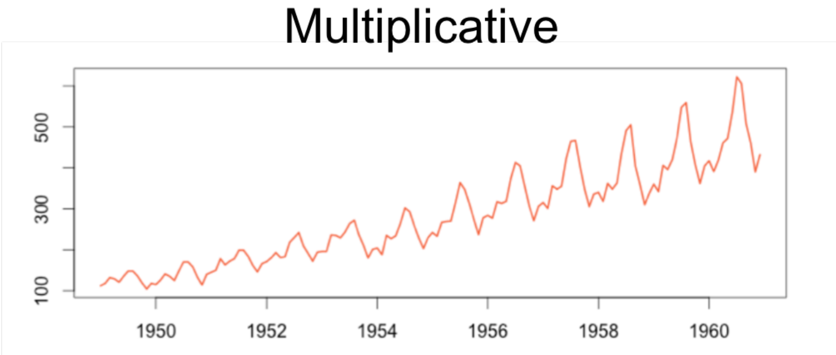
Multiplicative model: We assume that seasonality tends to increase as the trend increases:

$$x_t = m_t * s_t + e_t$$

Decomposition of Time Series

Additive or multiplicative? Plot the time series:





Removing the Trend

To remove the trend, we first need to find it!

For this we use **moving averages**: Take the mean of points close to each other.

Simple moving averages: Take points from one side.

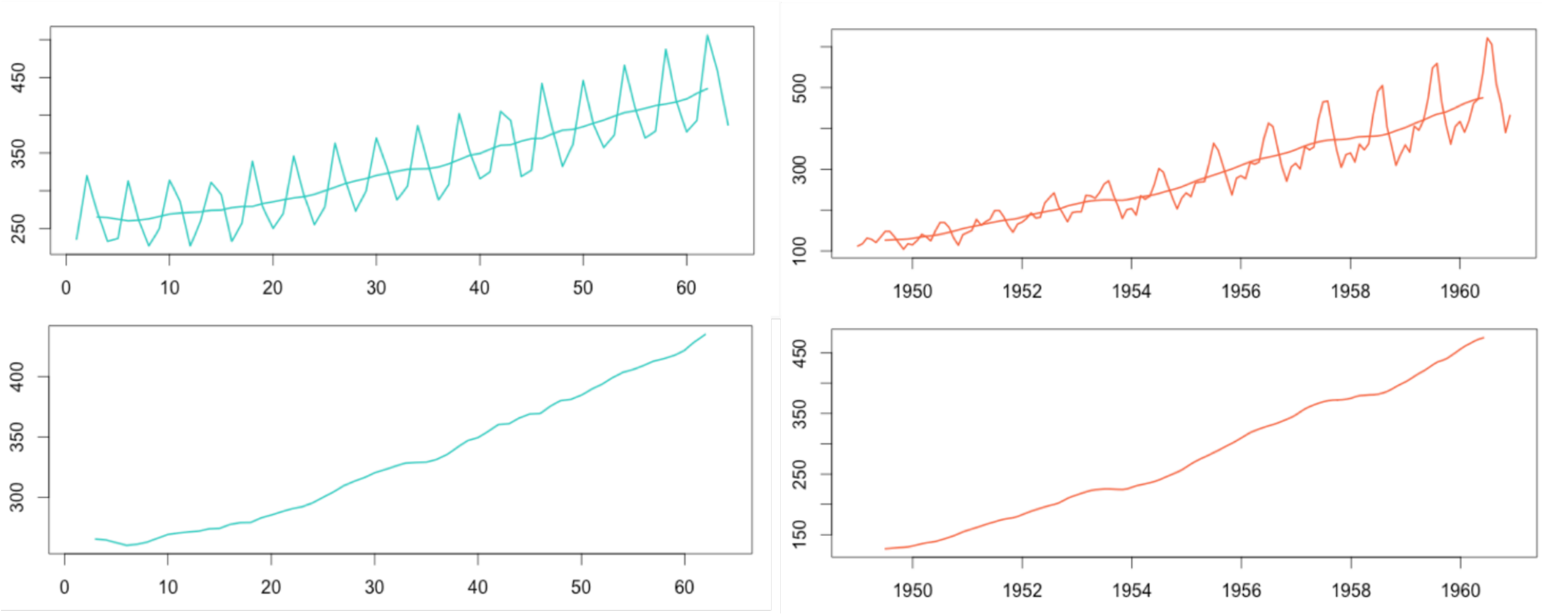
Time	1	2	3	4	5	6	7	8	9	10
Raw data	5	6	3	7	4	6	8	5	2	6
n = 2		5.5	4.5	5	5.5	5	7	6.5	3.5	4
n = 4				5.25	5	5	6.25	5.75	5.25	5.25

Central moving averages: Take points from two sides

Time	1	2	3	4	5	6	7	8	9
Raw data	5	6	3	7	4	6	8	5	2
CMA ($n = 3$)		4.67	5.33	4.67	5.67	6.00	6.33	5.00	

Removing the Trend

To remove the trend, we first need to find it!

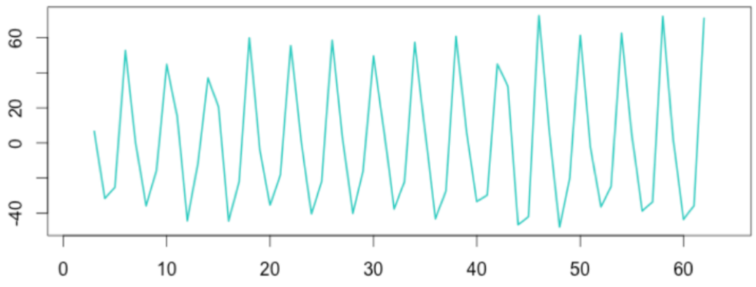


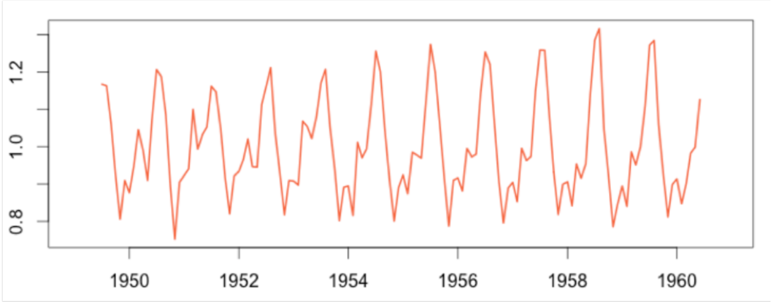
Removing the Trend

Now remove it from the time series:
If additive subtract it or if multiplicative divide by it.

$$x_t - m_t \qquad OR \qquad \frac{x_t}{m_t}$$

Remaining time series:



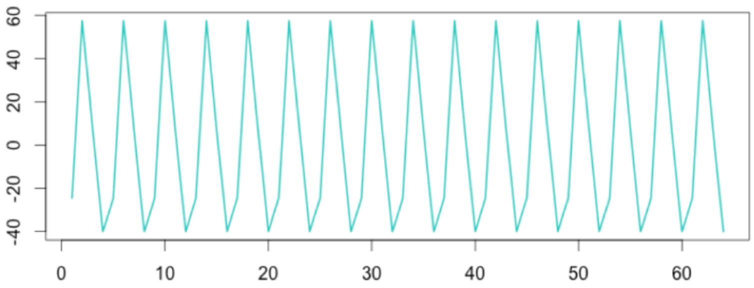


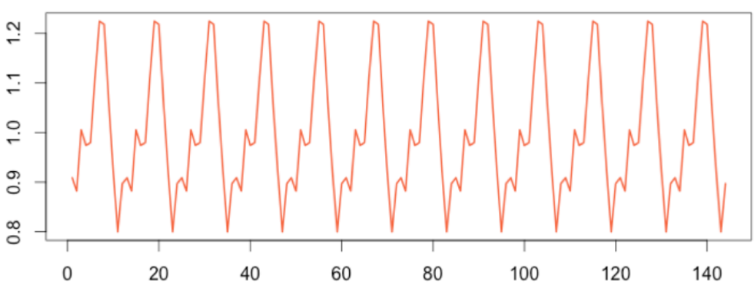
Removing the Seasonality

Now time to remove the seasonality. First find it:

For this we need to know what is the seasonality beforehand (weekly,monthly, yearly).

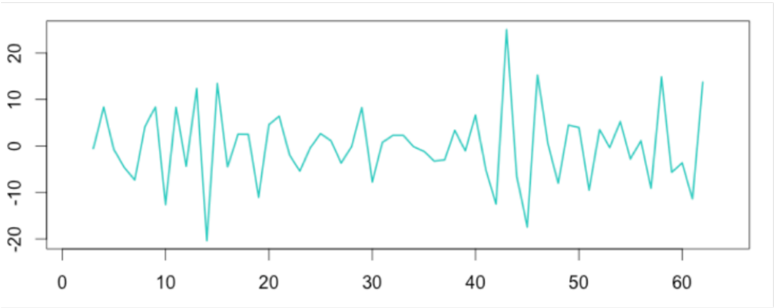
Idea: Average the seasonality to extract it.
Example for a monthly seasonality: put all the January data together and find the average, then all the February data and find the average, so on and so forth...

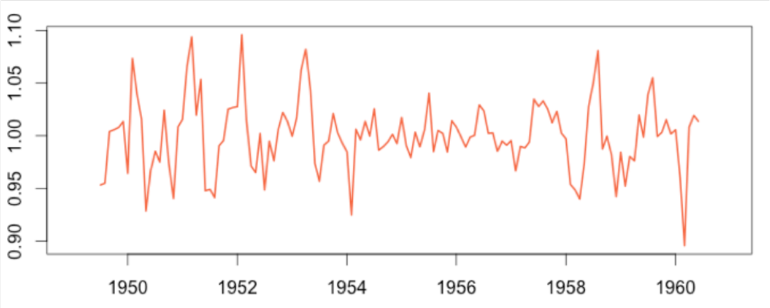




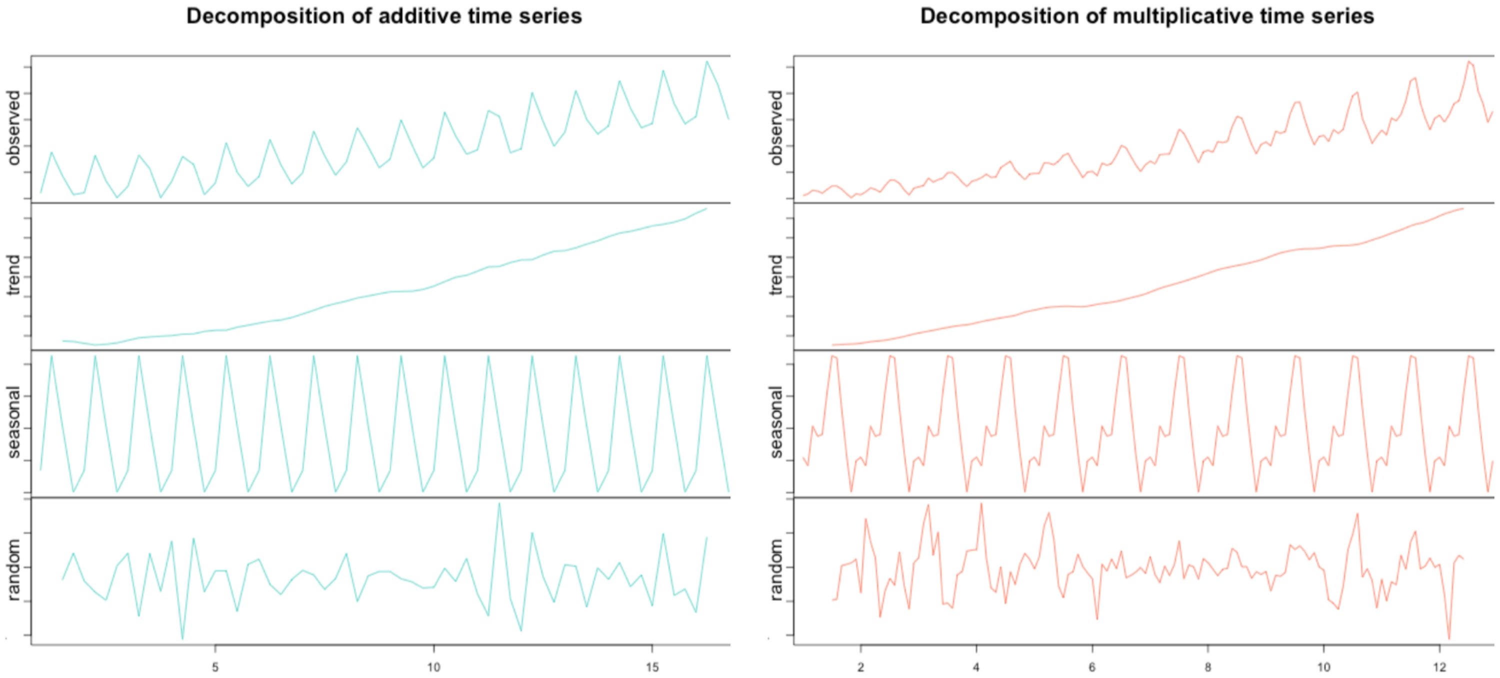
Removing the Seasonality

Removing the seasonality leaves only the random component. This is most of the times the interesting part for predictions.





Decomposition of Time Series



Stationary Time Series

A stationary time series has a constant mean and the variance as time goes on. In other words no trends, nor seasonality.

Taking the random error element from the decomposed the time series is one way to make the series stationary.

We can test for stationarity: **Dickey-Fuller test**:

$$H_0 : \textit{the time series is not stationary}$$

$$H_1 : \textit{the time series is stationary}$$

Differentiating Time Series

When we are not sure how the trend and season should be, there is a second way to remove trends and seasonality: **differentiate** the series:

Take the differences between consecutive values in the time series.

If the series is still not stationarity, we can differentiate a second time

Autocorrelation

Time series have observations that correlate with themselves, how can we measure it?

Idea:

1. Start with two copies of the time series, as they are the same they have a correlation of 1.
2. Now shift one of the copies one time step (this is called one **lag**). Find the correlation.
3. Continue for all lags

Each correlation at each lag can be seen directly in an ACF plot. (AutoCorrelation Function)

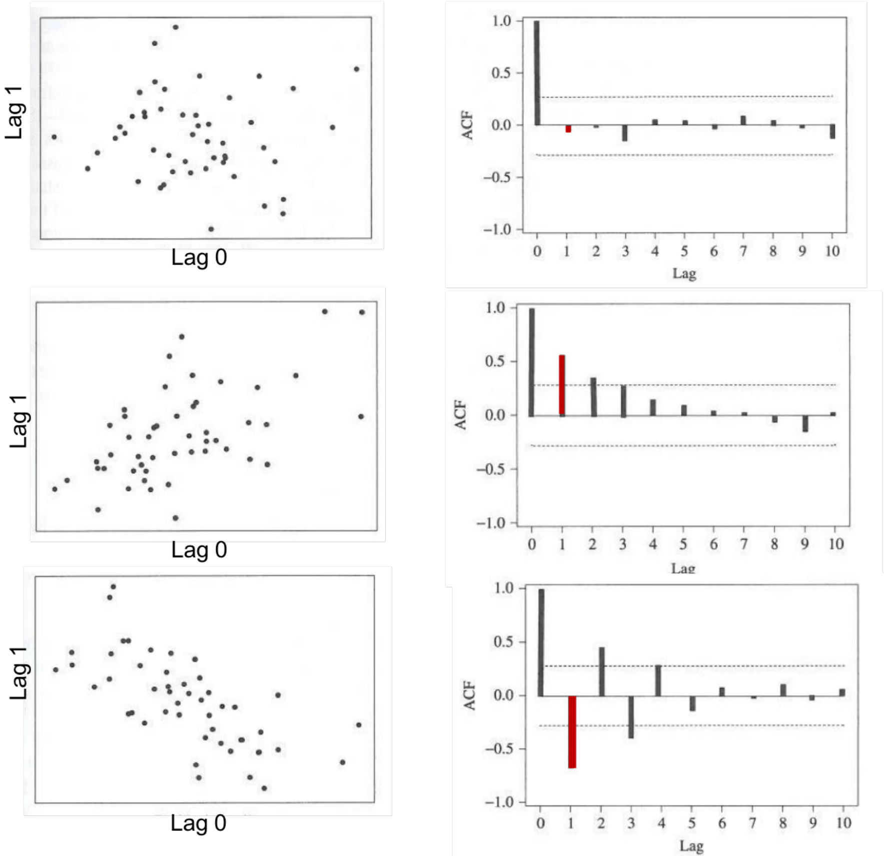
The plot includes dotted lines. Between the lines the correlation is not significant. They are the upper and lower confidence intervals.

ACF plot

None

Positive

Negative



Forecasting

We used linear regression models to predict new values.
With time series models, we can predict future values. This is called **forecasting**.

Three types of techniques:

- Find a lead variable, that is related with the variable to predict.
- Find similar data (for example sales of other products) and use this knowledge to predict the variable.
- Extrapolate: use the same time series to try to predict the future. Use only when we are sure that the trend will continue with the time

There are MANY models to predict time series, we could have a whole semester and would still need more time...

For this course, I only introduce the most famous one called **ARIMA** (extrapolation technique)

ARIMA

ARIMA model is a multiple linear regression model that takes into consideration autocorrelation!

Assumption is that the time series is **stationary** So, first make your time series stationary in case it is not!

How do we include the autocorrelation in the model?

The lags and the errors will be used as predictor variables!

$$x_t = \beta_1 x_{t-1} + \beta_2 x_{t-2} + \ldots + \beta_p x_{t-p} + error_t + \alpha_1 error_{t-1} + \alpha_2 error_{t-2} + \ldots + \alpha_q error_{t-q}$$

where $error_i$ is the difference between the moving average at ith instant and the actual value.

Remember moving average takes the mean of consecutive values.

ARIMA

We need to specify two parameters:

- Number of Auto-Regressive terms (**p**): The number of lags that we include in the model.
- Number of Moving Average terms (**q**): The number of error lags that we include in the model

Example

If p is 5, the predictors for x_t will be $x_{t-1}....x_{t-5}$.
If q is 5, the predictors for x_t will be $error_t....error_{t-5}$

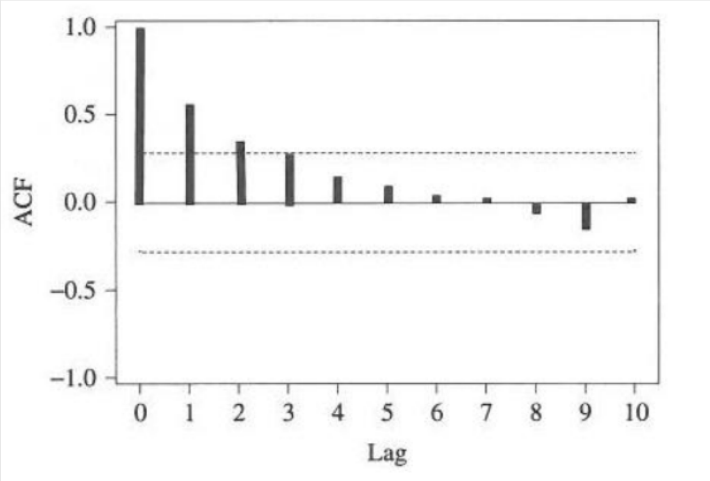
How to decide for the best values of p and q?

ARIMA

We can use the ACF plot to find the best **q** (error lags):

Choose q as the lag which crosses the upper confidence interval for the first time.

Example:



In this case, the best q is 3.

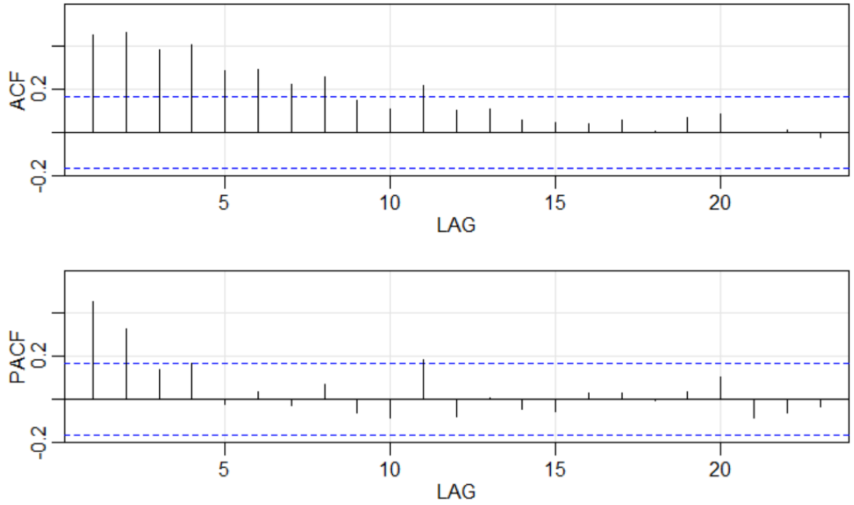
ARIMA

Finding the best **p** (lags) is similar, but we use the **PACF** plot.

The PACF (P for partial) also measures the correlation between the time series and the lagged versions of itself BUT it eliminates the variations already explained by the previous lags.

Again: Choose p as the lag which crosses the upper confidence interval for the first time.

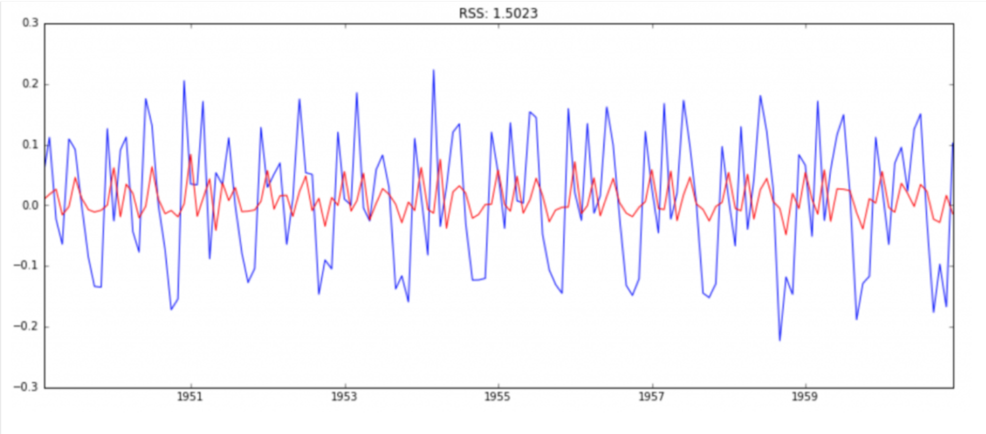
Example: In this case, the best q is 9 and the best p is 3.



ARIMA

Now we can fit the model in the same way we fitted multiple linear regression!

Example:

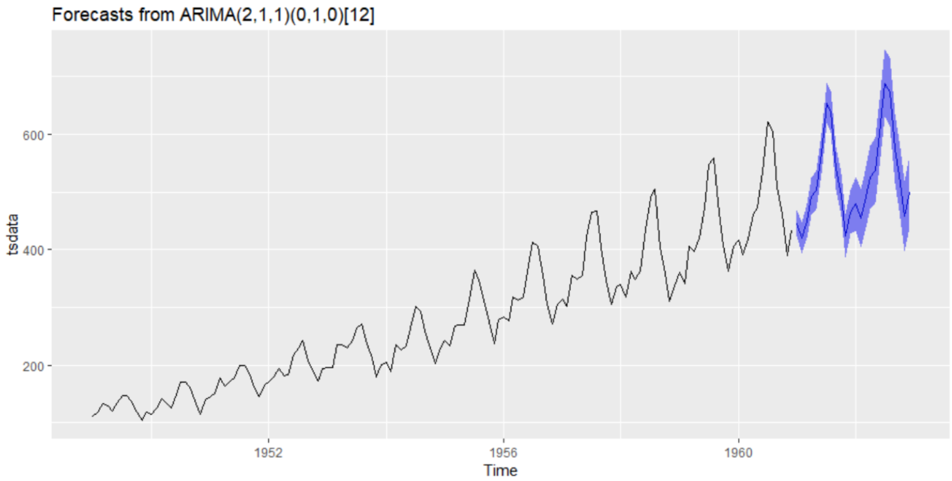


Not the best fit, but ok.
Remember that this series has the trend and seasonality taken away, we need to add it back if we want our original time series and our fitted model

ARIMA

Time to make the forecast!

Use the ARIMA model and extrapolate: obtain the value after the last time point. Use this to continue and obtain another point after that.
We can add confidence intervals to our predictions as well:



Caveat: ARIMA is only good to make short term predictions!

Literature

New book and two blog tutorials:

- Introductory time series with R: Chapters 1,2,3
- https://anomaly.io/seasonal-trend-decomposition-in-r/index.html
- https://www.analyticsvidhya.com/blog/2016/02/time-series-forecasting-codes-python/