Time Series Analysis and Forecasting

October 15 – 16, 2018

John Carlo P. Daquis
Assistant Professor 4
UP Diliman School of Statistics

Learning Objectives

Learning Objectives

This training course on advanced forecasting methods aims to provide participants with an understanding of the principles and steps in the Box-Jenkins (ARIMA) approach in modeling and forecasting.

Learning Objectives

Specifically, this course will train the participants

- to identify, estimate and test an ARIMA model;
- to perform diagnostic checks and remedial measures for the estimated ARIMA models;
- to use RStudio for ARIMA model building;
- to forecast values using the estimated ARIMA model; and
- to evaluate forecasts from candidate ARIMA models.

Chapter Outline

Chapter Outline

I. ARIMA Modeling

1. The Univariate Time Series Model
2. Stationarity and Unit Root Tests
2. Autocorrelation and Correlograms
3. The Autoregressive Process
4. The Moving Average Process
5. Non-stationarity and Integrated Processes
6. Estimating ARIMA Models in RStudio

The Univariate Time Series Model

The Univariate Time Series Model

What is ARIMA?

- ARIMA stands for **Autoregressive Integrated Moving Average** (models).
- It is a univariate modeling approach and analysis where only past observations of a time series is used to describe and forecast the behavior of the variable through the estimated model.

The Univariate Time Series Model

What is ARIMA?

- The estimation procedure used in ARIMA is the Box-Jenkins methodology developed by statisticians George Box and Gwilym Jenkins.
- The Box-Jenkins methodology can be divided into three main steps: (i) model specification, (ii) model estimation and selection, and (iii) diagnostic checking.

The Univariate Time Series Model

Building ARIMA models

- This approach in building time series models is provides the foundation on estimating **Autoregressive Integrated Moving Average** (ARIMA) models.
- ARIMA methodology centers on the theme that the time series model uses **past values** of a time series.

The Univariate Time Series Model

Building ARIMA models

- The Box-Jenkins approach involves three stages:

    1. Model specification
    2. Parameter estimation and selection
    3. Diagnostic checking

The Univariate Time Series Model

Stage 0: descriptive procedures

- This stage is not part of the Box-Jenkins methodology but plays a vital part in time series analysis.
- Descriptive procedures include the following:
  - Line graph of the time series
  - Time series decomposition
  - Review of literature and events that may have caused the time series' behavior

The Univariate Time Series Model

Stage 1: model specification

- This stage involves doing exploratory analysis on the time series data to know what possible ARIMA models can be fitted to it.

The Univariate Time Series Model

Stage 1: model specification

- Model specification stage includes the following:
    - Using the correlogram to know the possible ARIMA order.
    - Unit root testing for stationarity
    - Potential differencing and transformations for stationarity

The Univariate Time Series Model

Stage 2: Parameter estimation and selection

This stage includes the following:

- Parameter estimation is often done using the maximum likelihood estimation method. The software will be the one doing this.

The Univariate Time Series Model

Stage 2: Parameter estimation and selection

This stage includes the following:

- Choosing the “best” model. This part is uses the following:
  - **Information criteria** (IC). These are the Akaike, Bayes and Hannan-Quinn.
  - **Forecast accuracy** measures: RMSE and MAPE
  - The **parsimony principle**

The Univariate Time Series Model

Stage 3: diagnostic checking

- Model specification stage includes the following:
  - Check if the residual series is **white noise**.
  - Check for **structural breaks** or **outliers**.
  - Check for possible **variance specification** (**ARCH**) modeling.
  - Proceed to **forecasting**.

The Univariate Time Series Model

Preliminary remarks in ARIMA modeling

- Like exponential smoothing models, ARIMA models give more weight on more recent observations.
- Thus, ARIMA models are especially suited for short-term forecasting.
- Long-term forecasts are still plausible, but with less reliability.

The Univariate Time Series Model

Preliminary remarks in ARIMA modeling

- ARIMA models require an adequate sample size. Box and Jenkins suggest a **minimum of 50 observations** or time points.
- This translates to at least 50 years of annual data, at least 12 years of quarterly data and at least four years of monthly data.

The Univariate Time Series Model

Preliminary remarks in ARIMA modeling

- A large sample size is desirable especially when working with seasonal data. For **non-seasonal data, at least 30 time points** may be adequate.
- Interpret results when using shorter time series with caution.
- Other forecasting procedures can be used for shorter time series.

Stationarity and Unit Root Tests

Stationarity and Unit Root Tests

Stationarity

- Stationarity is an assumption that must be satisfied by a time series before ARIMA models can be used.
- Making the series stationary may involve a few the following procedures:
    - Differencing
    - Transformation

Stationarity and Unit Root Tests

Stationarity

- A stationary series $\mathbf{y_t}$ has a **constant mean**/at a **constant level** and a **constant variance** over time.
- The dependence structure of y_t (called autocorrelation) moreover is not a function of time.

Stationarity and Unit Root Tests

Stationarity

- There are several tests for stationarity, one of which is the **Augmented Dickey-Fuller (ADF)** test.
- It is really a unit root test, but interpretations can be translated to a test of stationarity of the time series.

Stationarity and Unit Root Tests

Stationarity

- The ADF is a test between

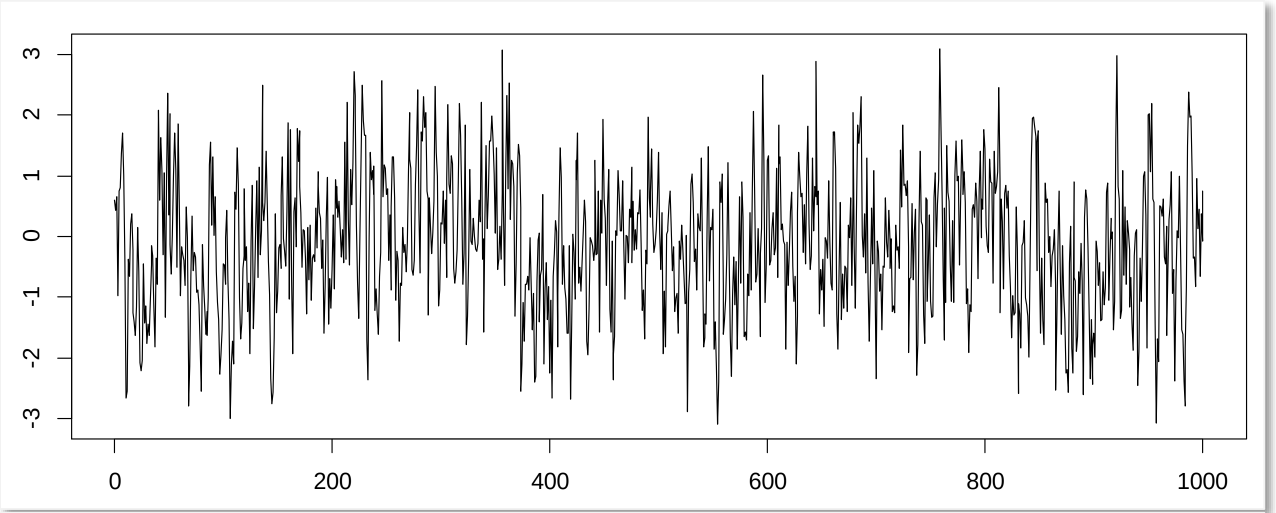
  H_o: The series is **non-stationary** vs.
  H_a: The series is **stationary**

- As with other tests, we reject the null hypothesis (H_o) if the p-value is less than or equal to the level of significance (usually 0.05)

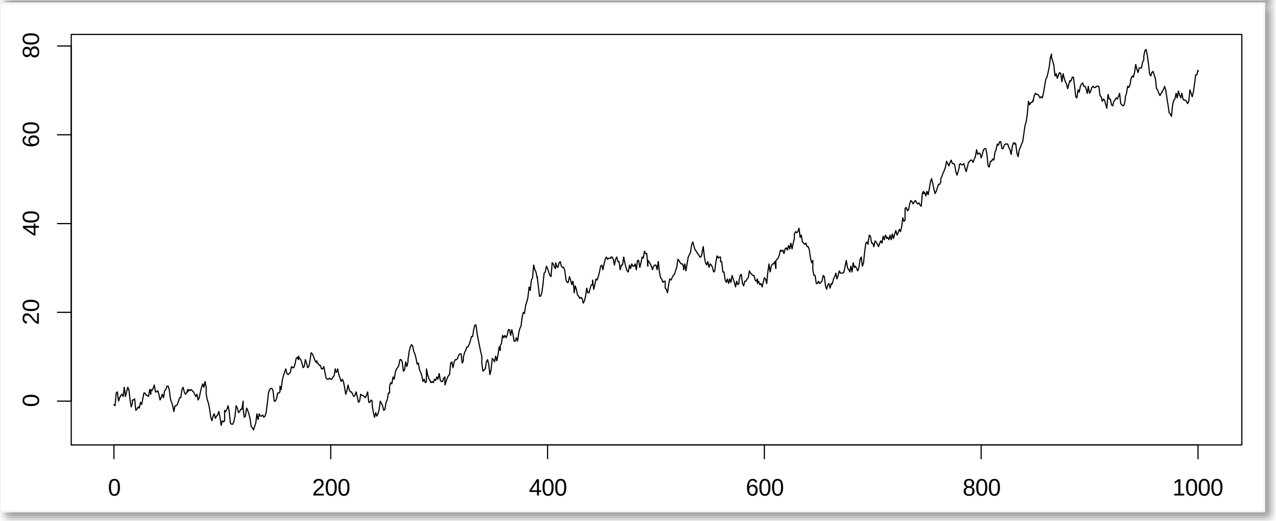
Stationarity and Unit Root Tests

Stationarity

Stationary series



Non-stationary series



Stationarity and Unit Root Tests

Stationarity and unit root tests

- A stationary series $\mathbf{y_t}$ has a **constant mean** at a **constant level** and a **constant variance** over time.
- The dependence structure of y_t (called autocorrelation) moreover is not a function of time.

Stationarity and Unit Root Tests

Stationarity tests

- Here's an RStudio output that indicates non-stationarity (a unit root is present):

```
        Augmented Dickey-Fuller Test

data:  pder.ts
Dickey-Fuller = -1.6761, Lag order = 7, p-value = 0.7149
alternative hypothesis: stationary
```

- Since p-value, 0.7149 is not less than 0.05, we do not reject the null hypothesis of unit root presence (non-stationarity).

Stationarity and Unit Root Tests

Stationarity tests

- Here's an RStudio output that indicates stationarity (a unit root is not present):

```
        Augmented Dickey-Fuller Test

data:  diff(pder.ts)
Dickey-Fuller = -6.0099, Lag order = 7, p-value = 0.01
alternative hypothesis: stationary
```

- Since p-value is less than 0.05, we reject the null hypothesis of unit root presence (non-stationarity.

Stationarity and Unit Root Tests

Stationarity tests

```
library("tseries")  # Unit Root Tests
library("forecast") # For auto.arima

#Read table and convert it to a time series onbject
pder <- read.table("pder.txt", header=T)
head(pder)

pder.ts <- ts(pder$pder,frequency=12,start=c(1980,1))
head(pder.ts)

#Performing ADF tests on the original and differenced Peso-Dollar exchange rates
adf.test(pder.ts, alternative="stationary")
adf.test(diff(pder.ts), alternative="stationary")
```

Autocorrelations and Correlograms

Autocorrelations and Correlograms

Autocorrelation and partial autocorrelation

- The **autocorrelation** and **partial autocorrelation** are vital tools in the Box-Jenkins method in model specification stage.
- The autocorrelation function is commonly known as the **ACF** while the partial autocorrelation function is also known as the **PACF**.

Autocorrelations and Correlograms

Autocorrelation and partial autocorrelation

- Both the ACF and the PACF are **correlations**, hence measure some form of linear relationship.
- Both the ACF and the PACF have the prefix "auto", which implies that these are correlations of a time series with (some past values of) itself.

Autocorrelations and Correlograms

Autocorrelation and partial autocorrelation

- The main idea in autocorrelation analysis is to calculate a correlation coefficient for each set of ordered pairs of within variable values separated by k periods or lags.
- They measures the **direction and strength of linear relationship between pairs of variables**.

Autocorrelations and Correlograms

Autocorrelation and partial autocorrelation

- The ACF and PACF are most useful when visualized as **correlograms**.
- These correlograms give us a glance on the dependence structure of the time series. In particular, the graphs will help us decide
  - if the time series is already stationary; and
  - which proper ARIMA model should be specified for the time series.

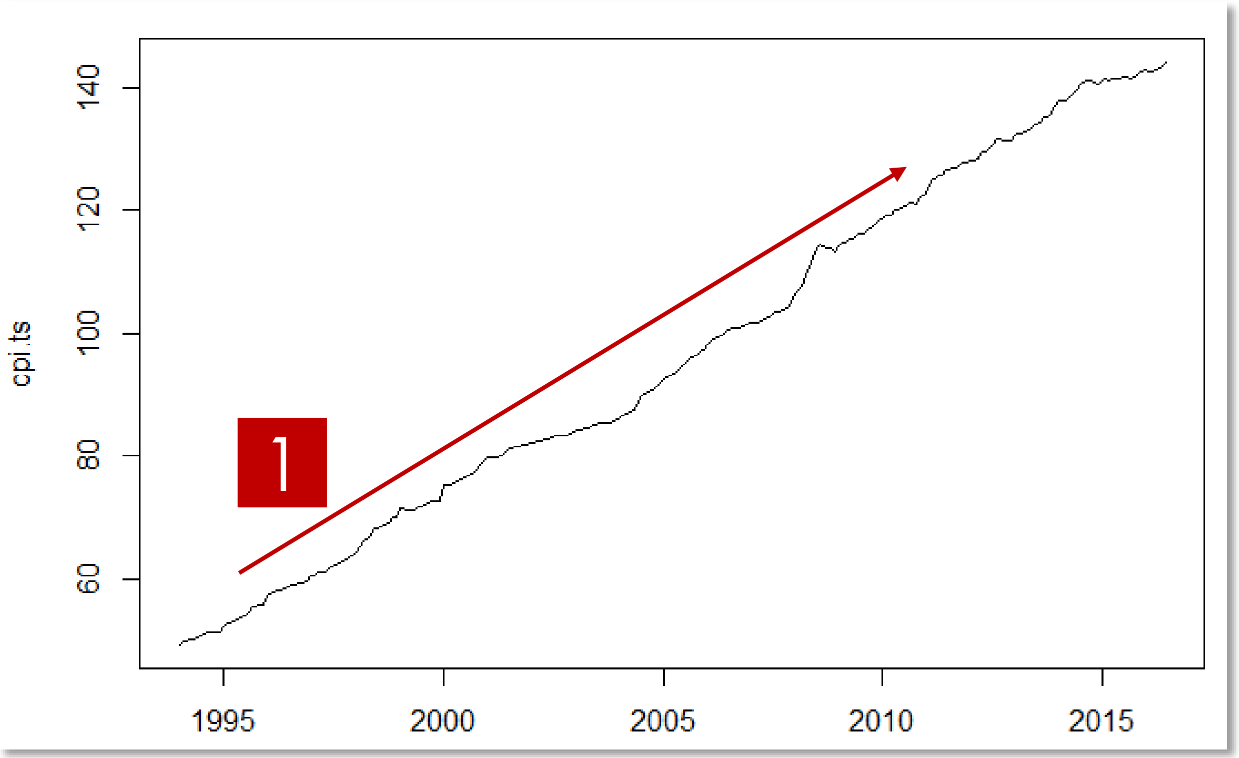
Autocorrelations and Correlograms

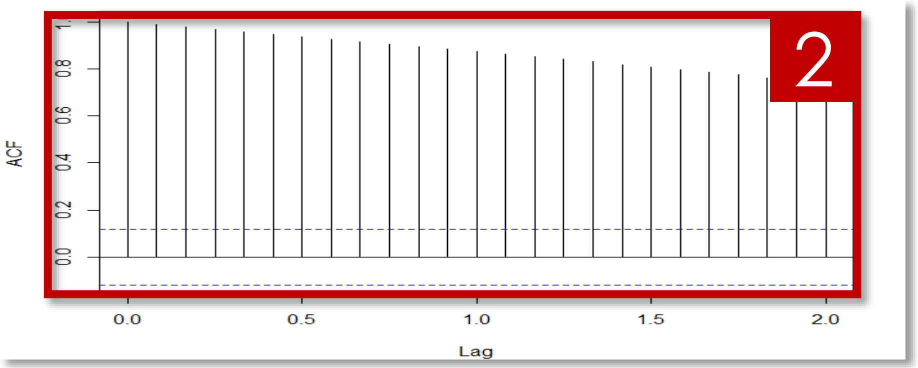
Autocorrelation and partial autocorrelation

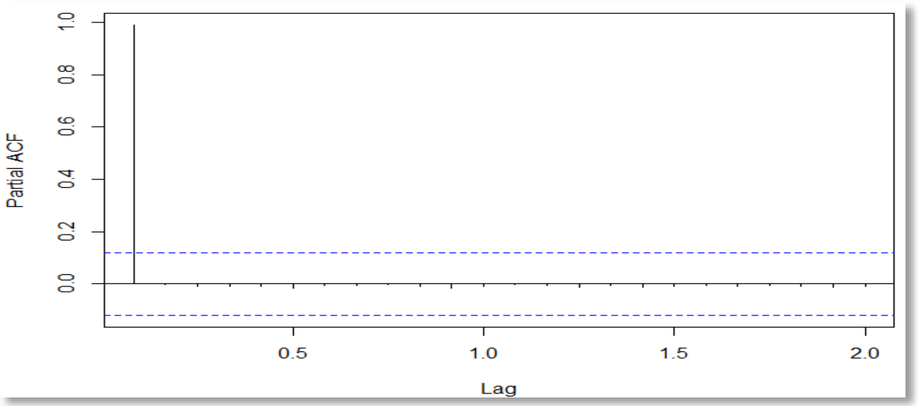
- Box and Jenkins suggest that the maximum number of useful estimated autocorrelations (called the lags) is roughly **T/4**, where T is the number of observations (e.g. if the data has T=100 time points, look at at least 25 lags).
- The software calculates the ACF and PACF values.

Autocorrelations and Correlograms

Autocorrelation and partial autocorrelation







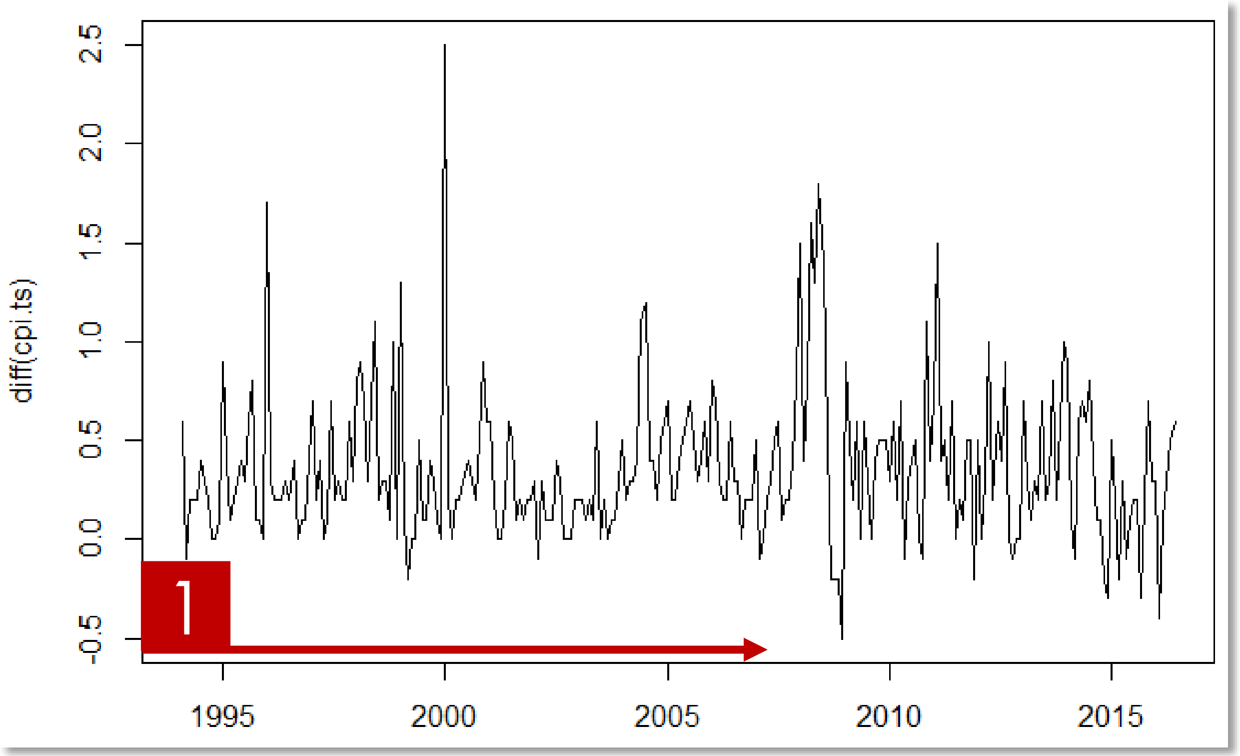
Autocorrelations and Correlograms

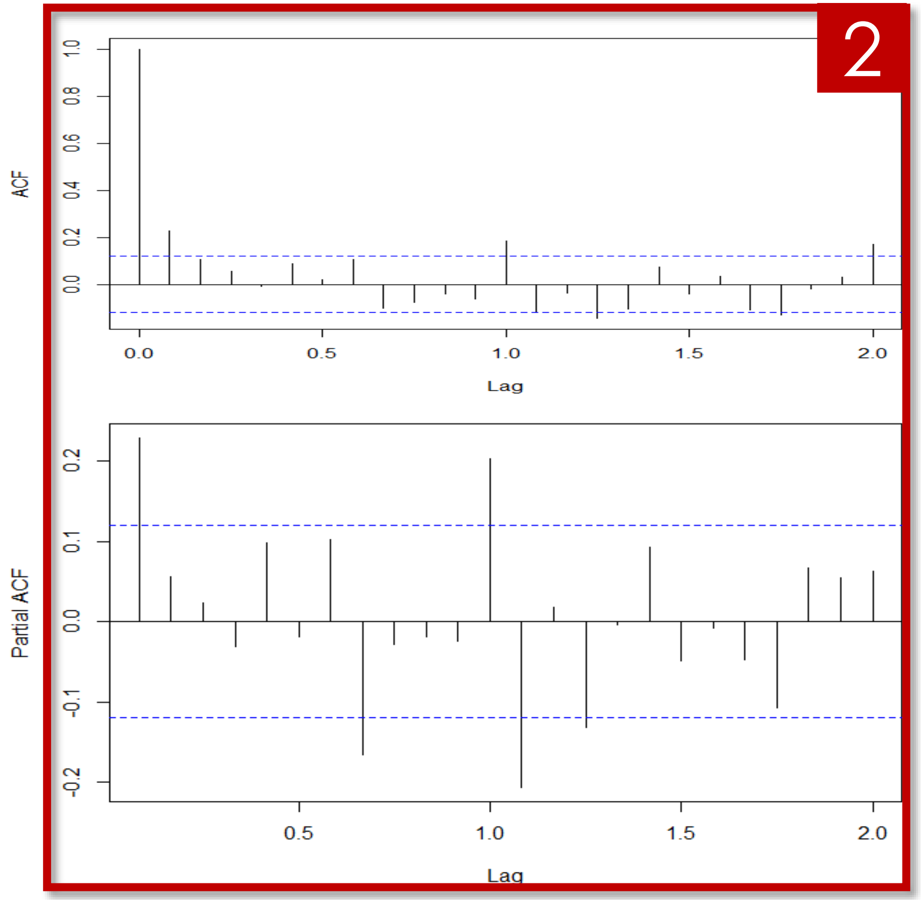
Autocorrelation and partial autocorrelation

- The graph of consumer price index (CPI) indicate an increasing trend (1).
- The ACF bars taper off very slowly. This is indicative of a non-stationary series.

Autocorrelations and Correlograms

Autocorrelation and partial autocorrelation





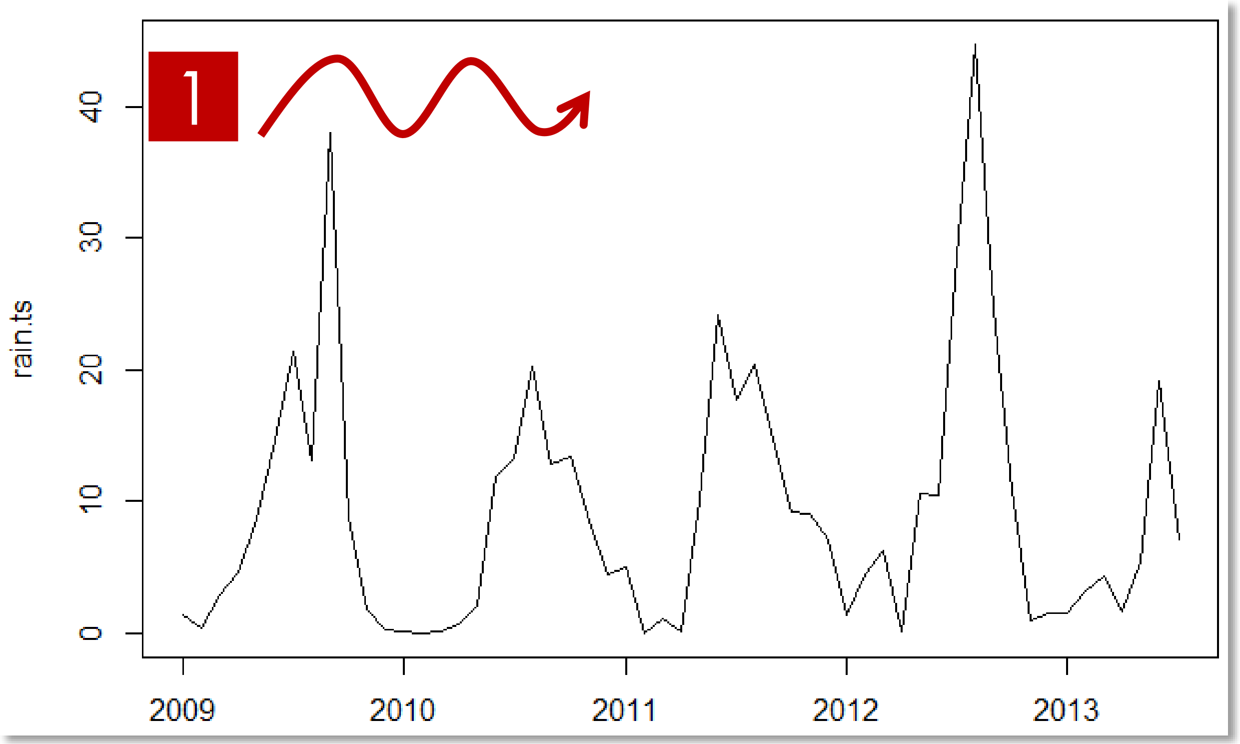
Autocorrelations and Correlograms

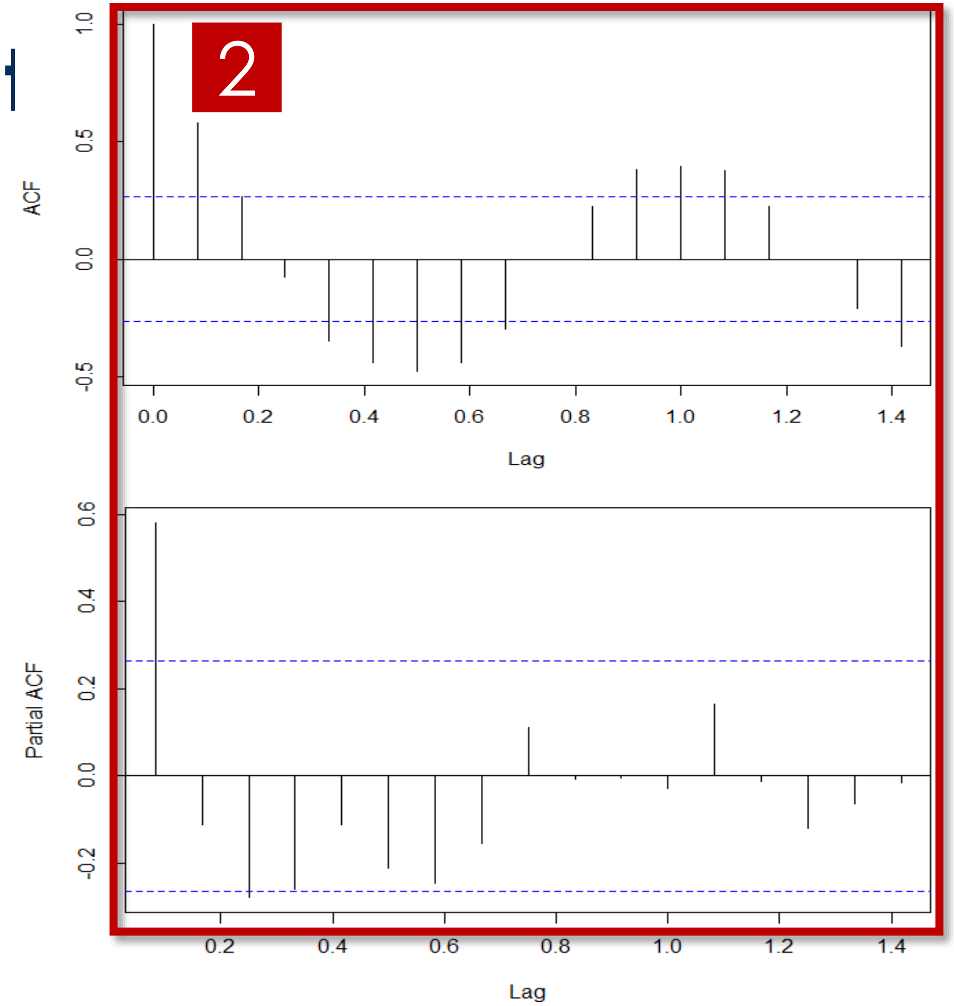
Autocorrelation and partial autocorrelation

- The graph of the first difference of CPI, or D(CPI) indicate no trend (1).
- Aside from some spikes, the ACF and the PACF functions does not seem to exhibit non-stationary behavior (2).
- Unit root tests (stationary tests) can conclusively tell if D(CPI) is already stationary or if further differencing is still needed.

Autocorrelations and Correlograms

Autocorrelation and partial aut





Autocorrelations and Correlograms

Autocorrelation and partial autocorrelation

- The graph of average rainfall (ave_rain) clearly shows seasonality behavior (1).
- This results to a sinusoidal or wave-like ACF and PACF pattern (2).
- Steps must also be done to address seasonality in time series. It can be (i) deseasonalization, (ii) adding seasonal indices, or (iii)include in the model.

Autocorrelations and Correlograms

Autocorrelation and partial autocorrelation

```
#Visualizing its dependence structure
plot(cpi.ts)
acf(cpi.ts)
pacf(cpi.ts)

plot(diff(cpi.ts))
acf(diff(cpi.ts))
pacf(diff(cpi.ts))

plot(rain.ts)
acf(rain.ts)
pacf(rain.ts)
```

The Autoregressive Process

The Autoregressive Process

The autoregressive (AR) process

- The autoregressive process of order 1, or the AR(1) is defined below:

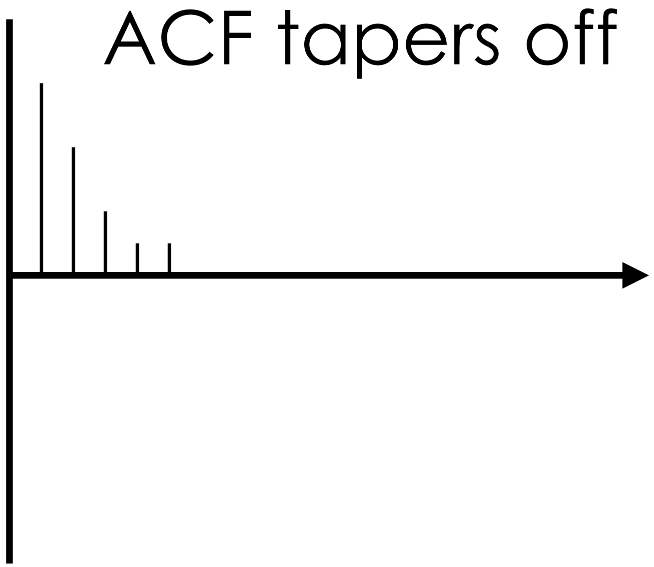
$$y_t = c + \phi_1 y_{t-1} + \varepsilon_t$$

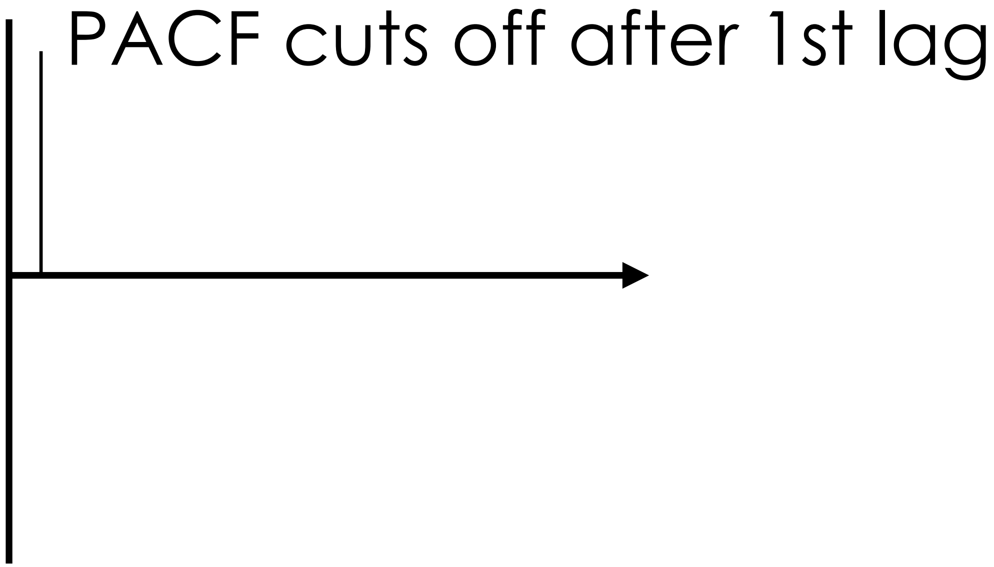
here, y_t is a stationary series and ε_t is a **white noise process** with mean zero and constant variance.

The Autoregressive Process

The autoregressive (AR) process

- The ACF and the PACF of an AR(1) process look like these:

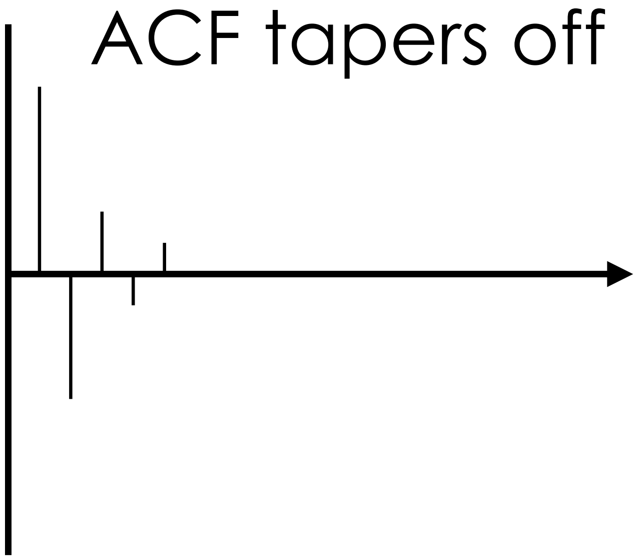


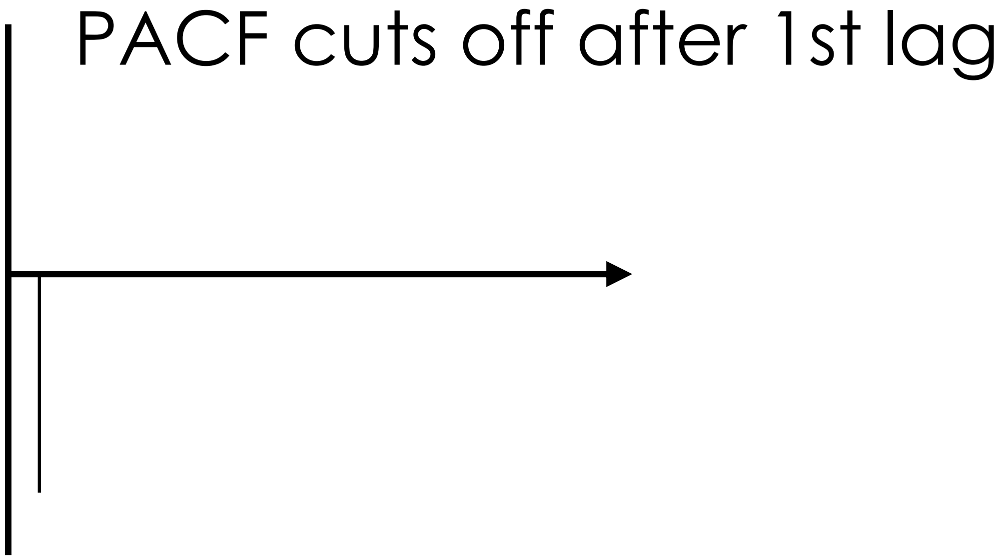


The Autoregressive Process

The autoregressive (AR) process

- The ACF and the PACF of an AR(1) process look like these:

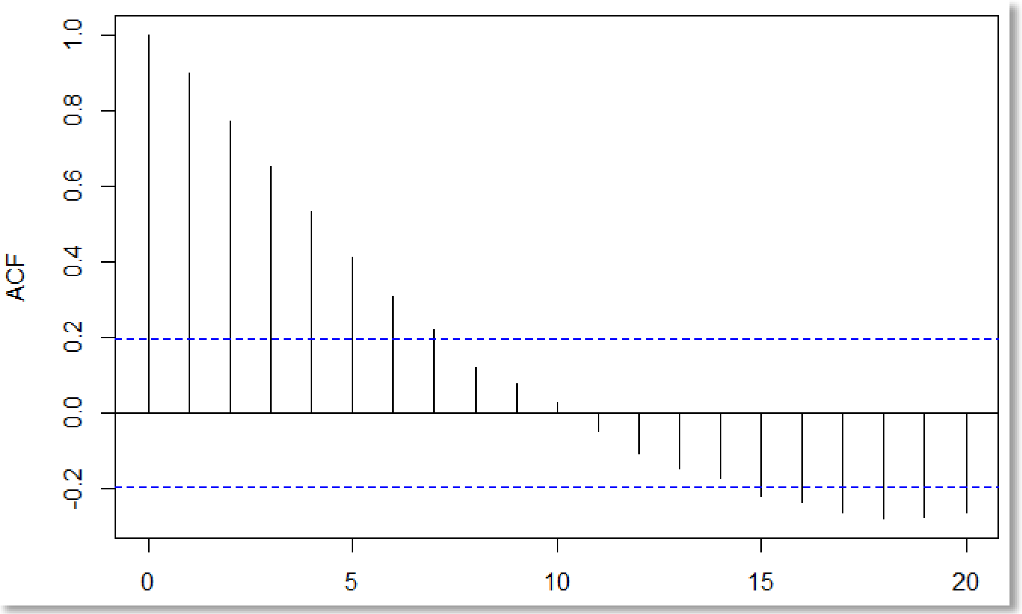


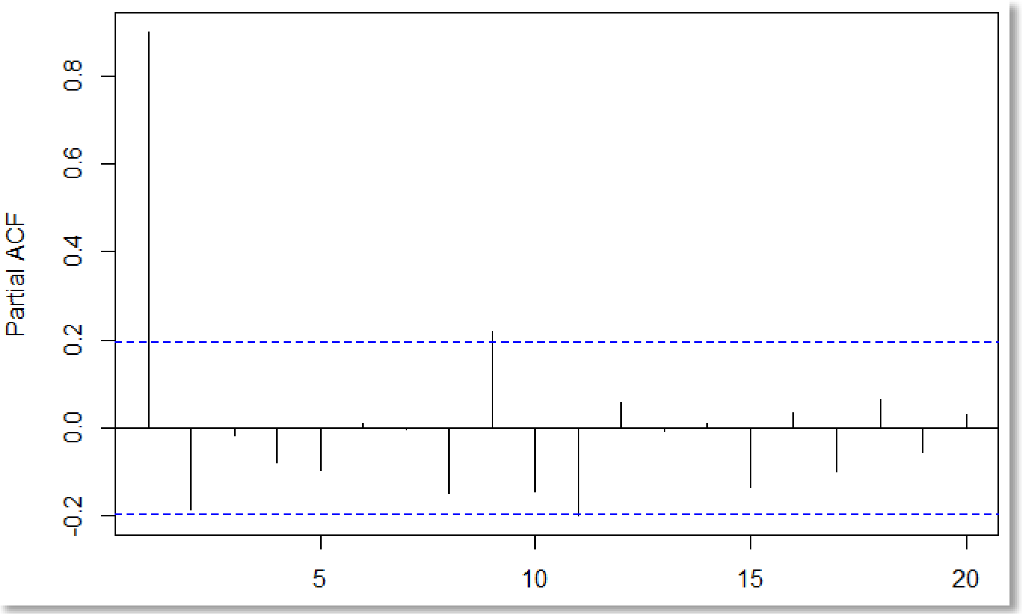


The Autoregressive Process

The AR process

- The ACF and the PACF of an AR(1) process look like these (in RStudio):

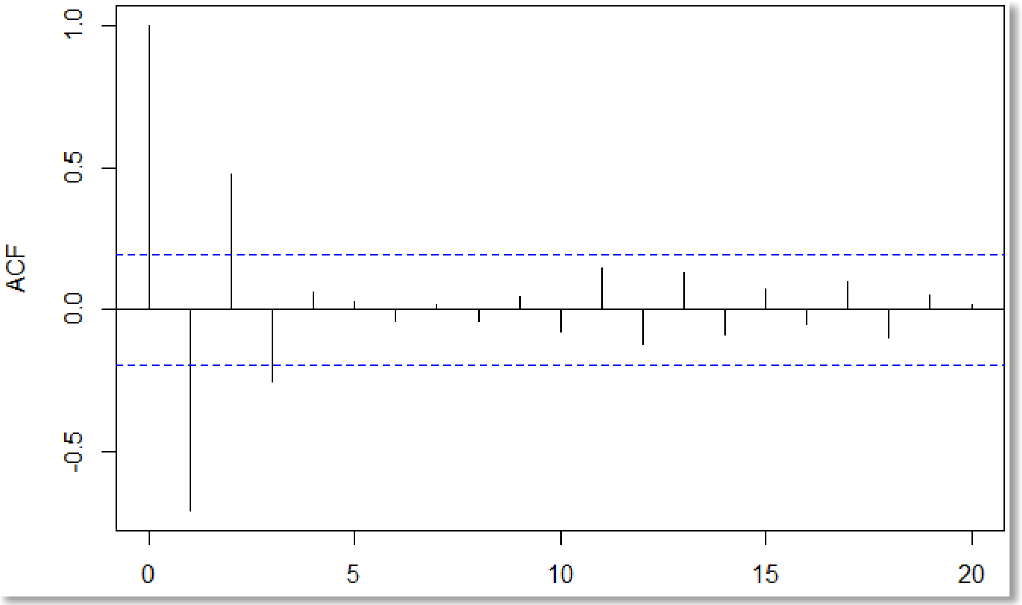


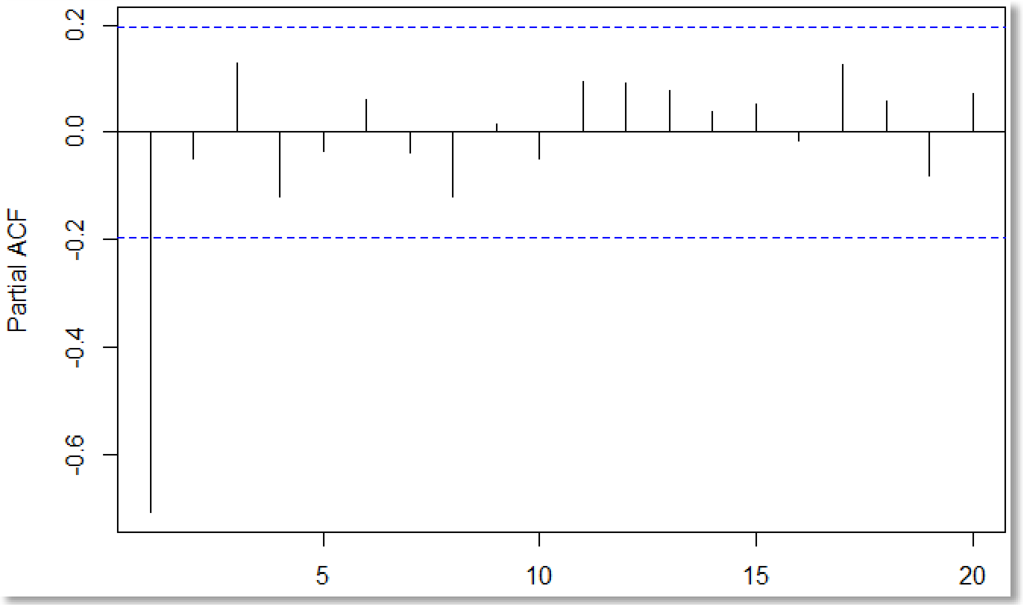


The Autoregressive Process

The AR process

- The ACF and the PACF of an AR(1) process look like these (in RStudio):





The Autoregressive Process

The autoregressive (AR) process

- The autoregressive process of order 2, or the AR(2) is defined below:

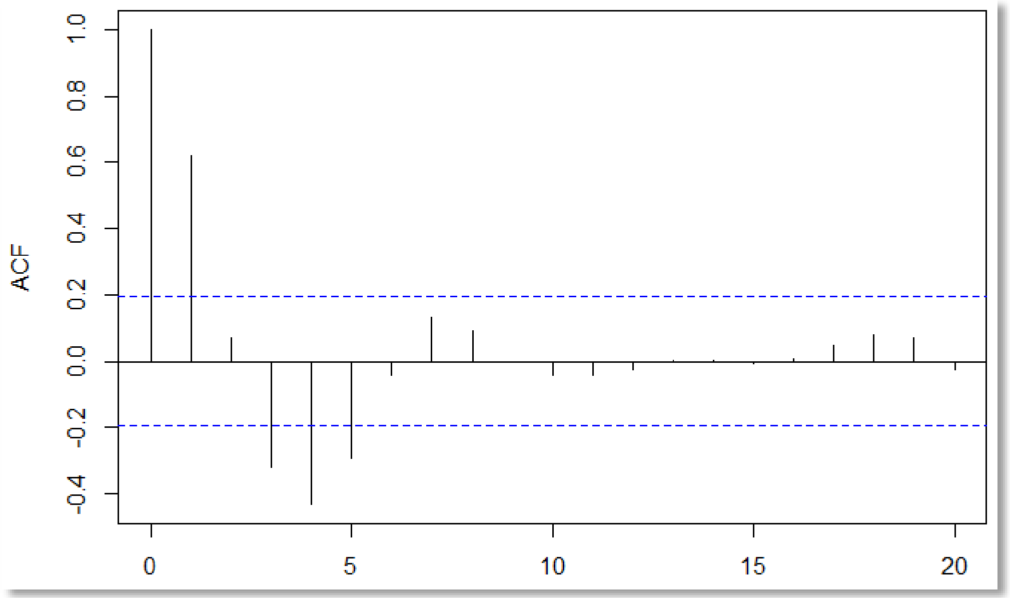
$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \varepsilon_t$$

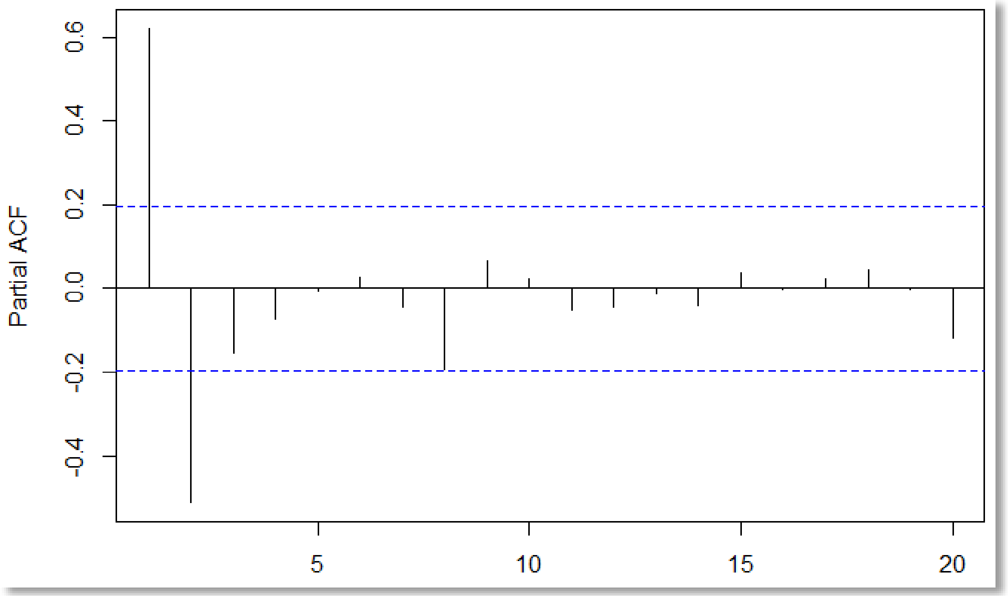
here, y_t is a stationary series and ε_t is a **white noise process** with mean zero and constant variance.

The Autoregressive Process

The AR process

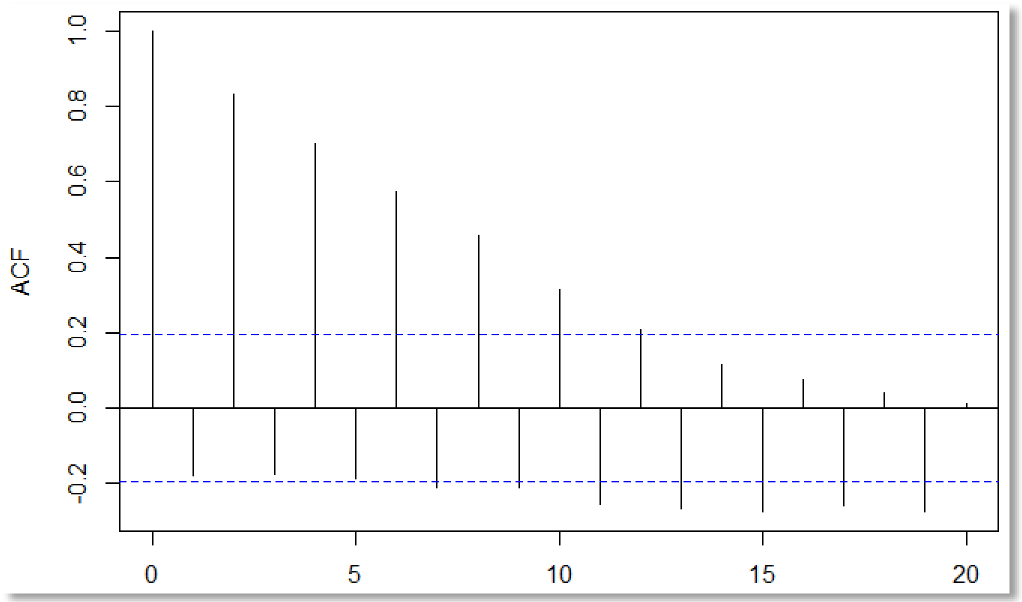
- The ACF and the PACF of an AR(2) process look like these (in RStudio):

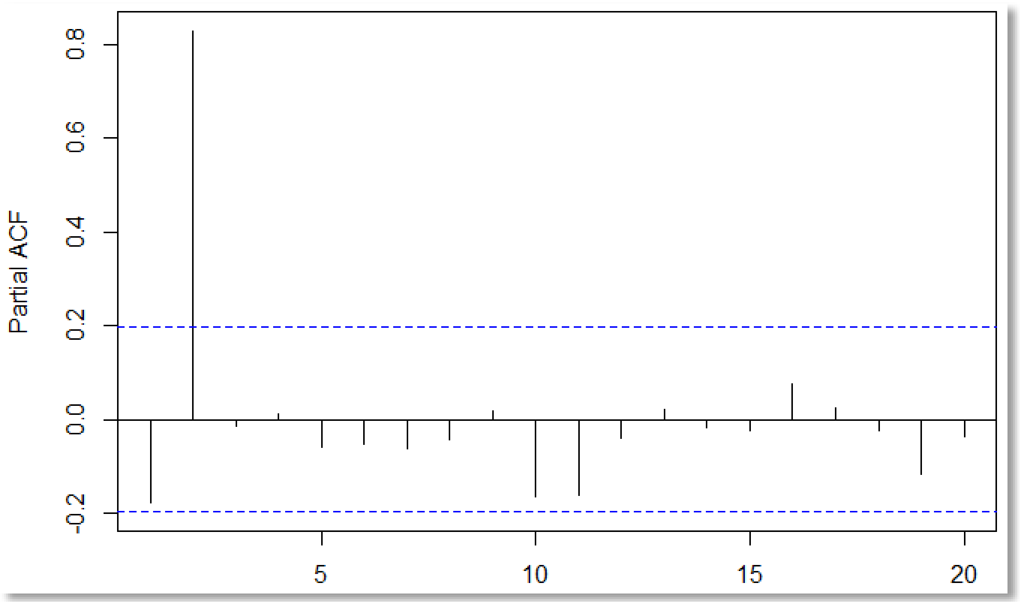




The AR process

- The ACF and the PACF of an AR(2) process look like these (in RStudio):





The Autoregressive Process

The autoregressive (AR) process

- In general, the autoregressive process of order p, or the AR(p) is defined below:

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + \varepsilon_t$$

here, the **ACF** tapers to zero while the **PACF** cuts off after lag p.

The Moving Average Process

The Moving Average Process

The moving average (MA) process

- The moving average process of order 1, or the MA(1) is defined below:

$$y_t = c + \theta_1 \varepsilon_{t-1} + \varepsilon_t$$

here, y_t is a stationary series and ε_t is a **white noise process** with mean zero and constant variance.

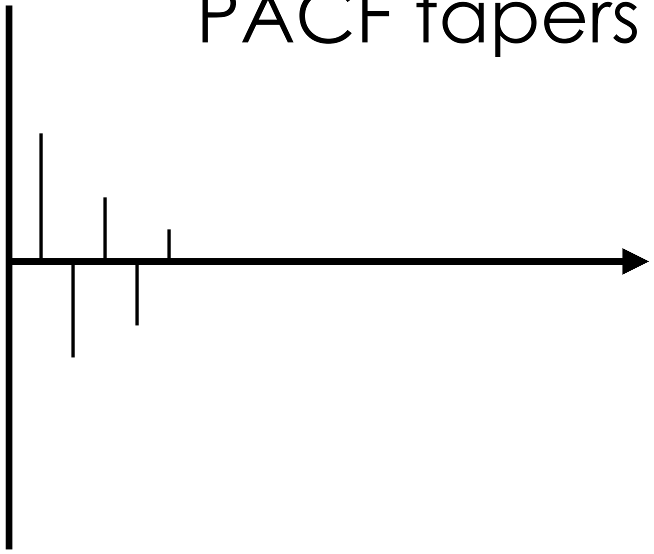
The Moving Average Process

The moving average (MA) process

- The ACF and the PACF of an MA(1) process look like these:

ACF cuts off after lag1

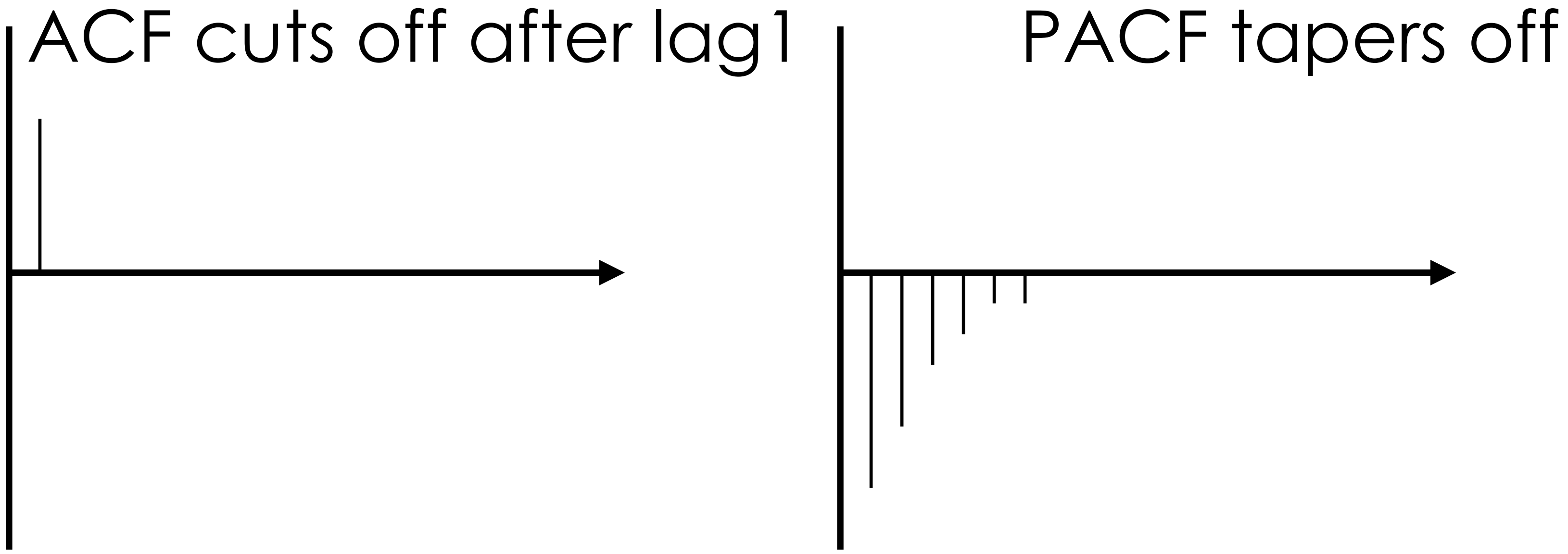
PACF tapers off



The Moving Average Process

The moving average (MA) process

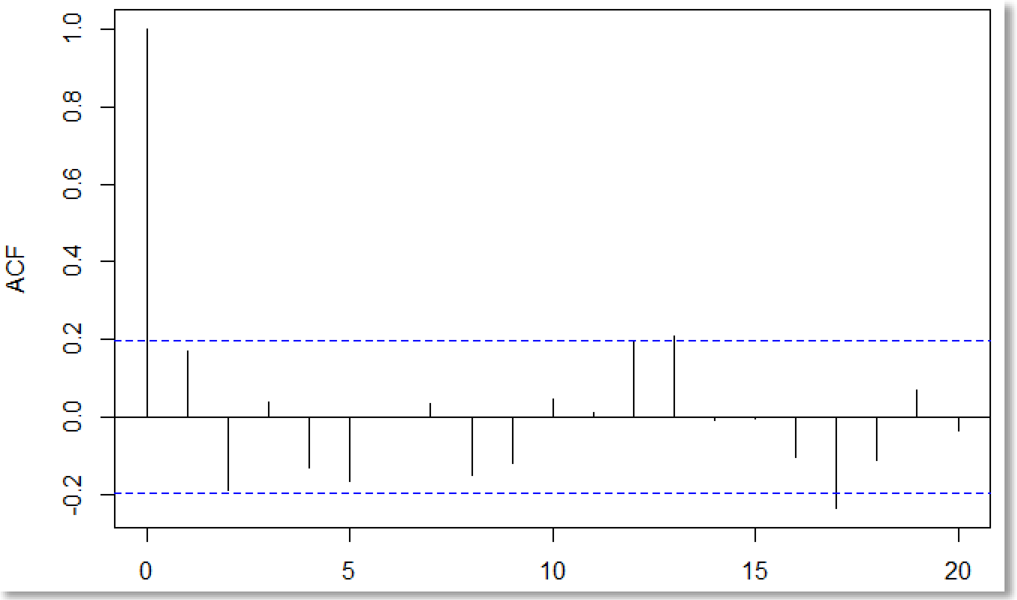
- The ACF and the PACF of an MA(1) process look like these:

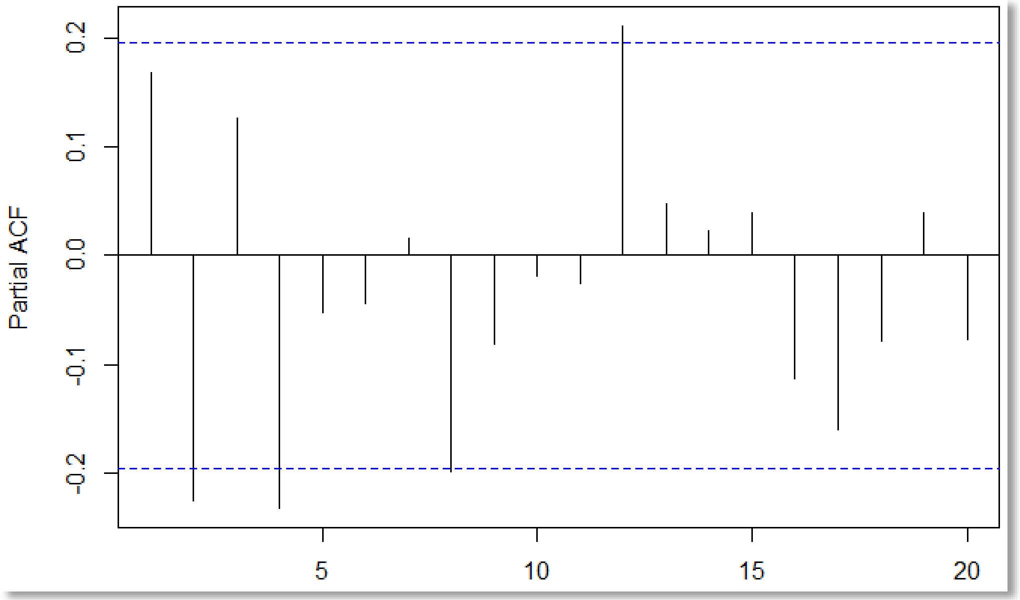


The Moving Average Process

The MA process

- The ACF and the PACF of an MA(1) process look like these (in RStudio):

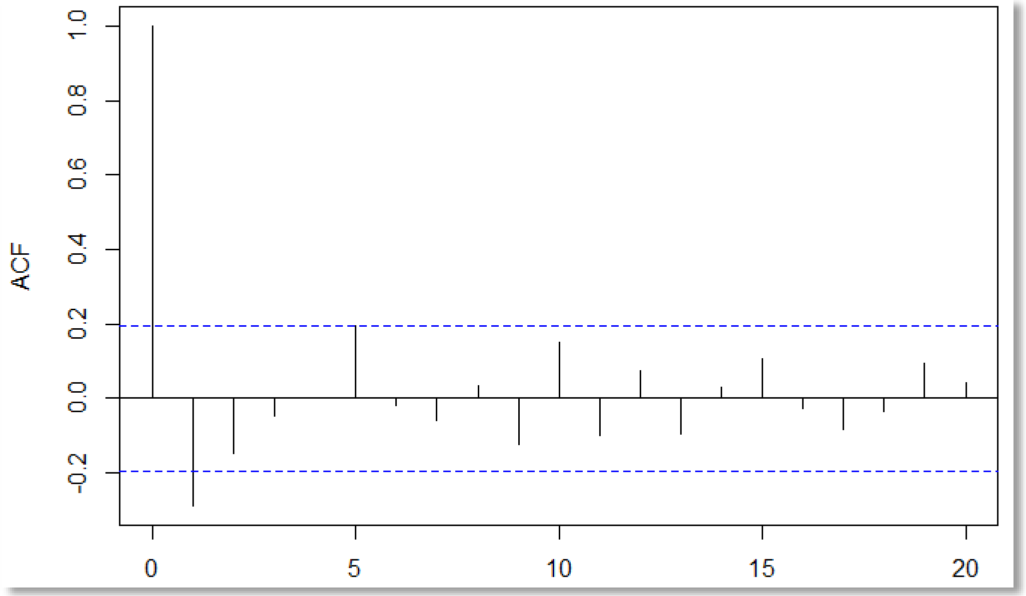


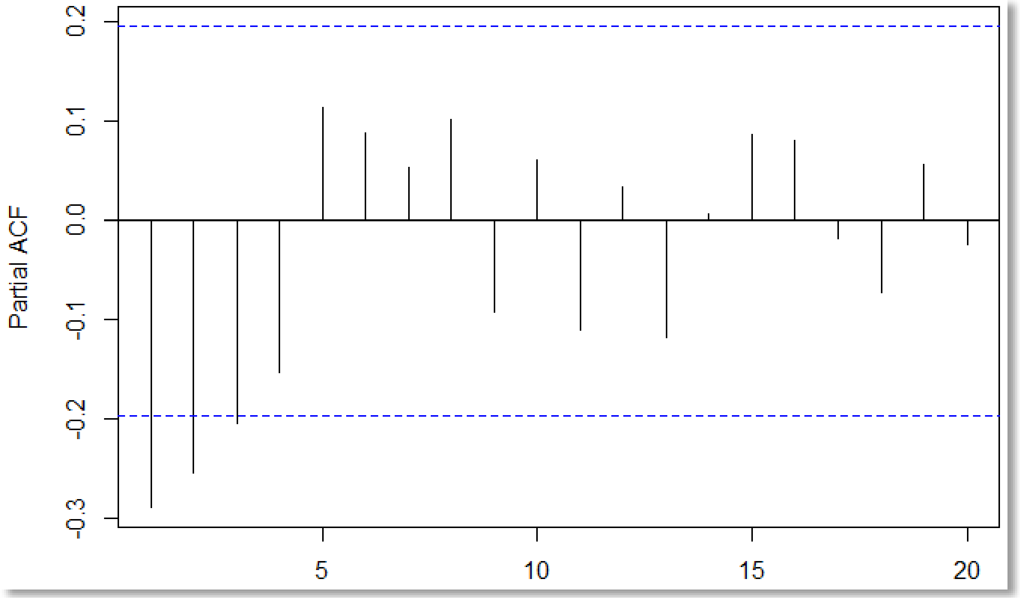


The Moving Average Process

The MA process

- The ACF and the PACF of an MA(1) process look like these (in RStudio):

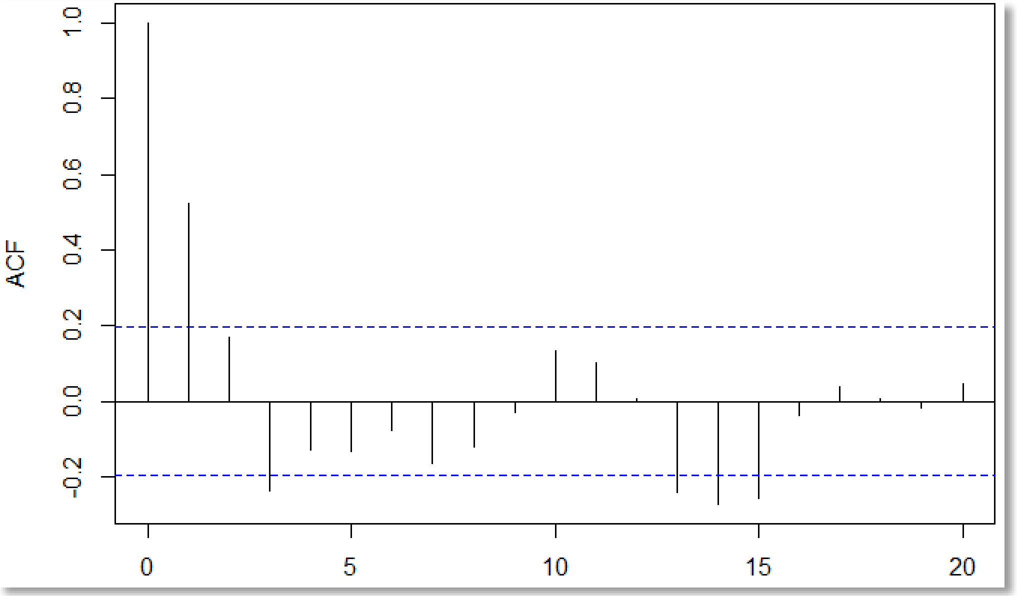


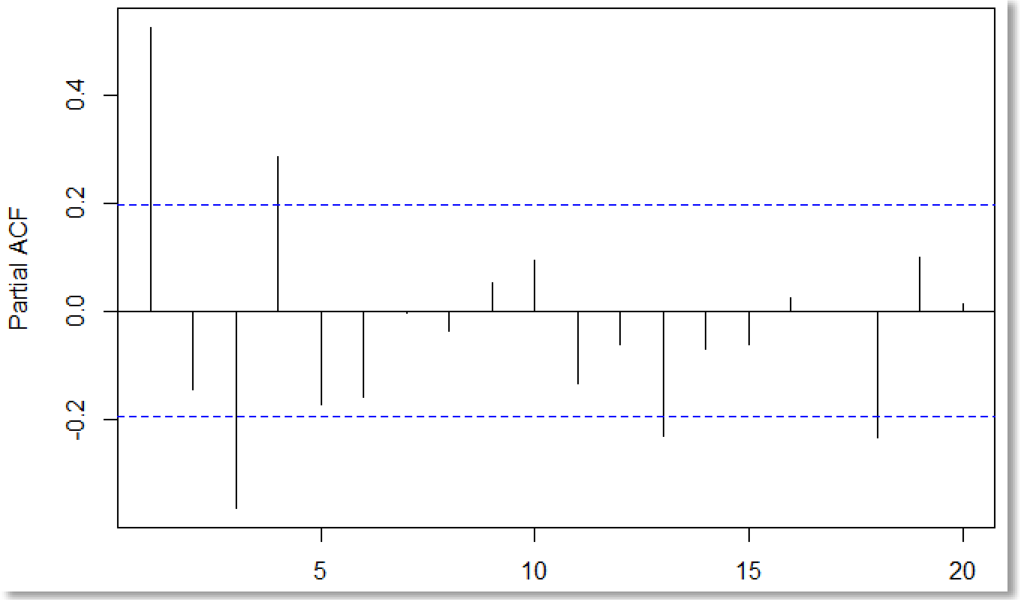


The Moving Average Process

The MA process

- The ACF and the PACF of an MA(2) process look like these (in RStudio):





The Moving Average Process

The moving average (MA) process

- The moving average process of order q, or the MA(q) is defined below:

$$y_t = c + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \mathrm{K} + \theta_q \varepsilon_{t-q} + \varepsilon_t$$

here, the **PACF** tapers to zero while the **ACF** cuts off after lag q.

The ARMA Process

The ARMA Process

The ARMA process

- The ARMA process of order p and q, or the ARMA(p,q) is defined below:

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p}$$

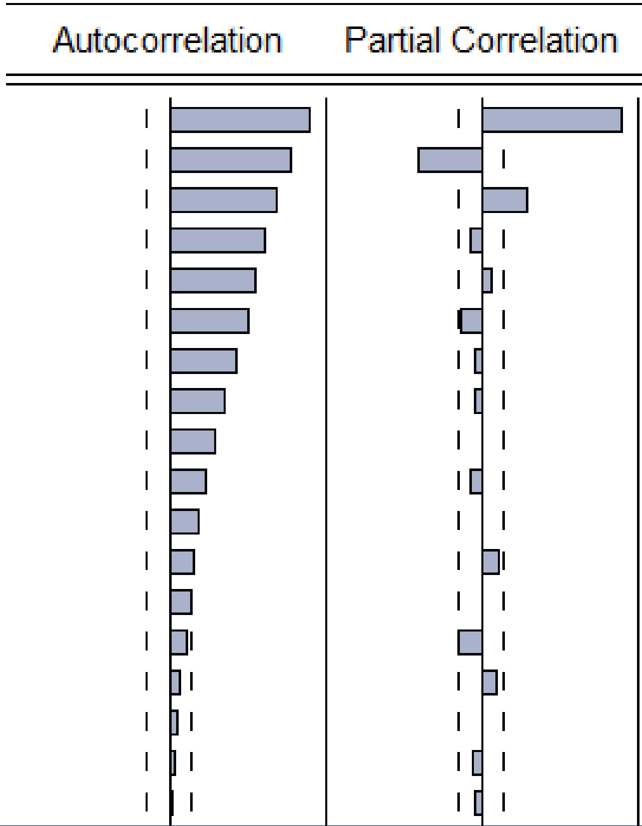
$$+ \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \dots + \theta_q \varepsilon_{t-q} + \varepsilon_t$$

The ARMA Process

The ARMA process

- The ACF and the PACF of an ARMA process look like these (in RStudio):

$$y_t = 0.9y_{t-1} + \varepsilon_t + 0.75\varepsilon_{t-1}$$



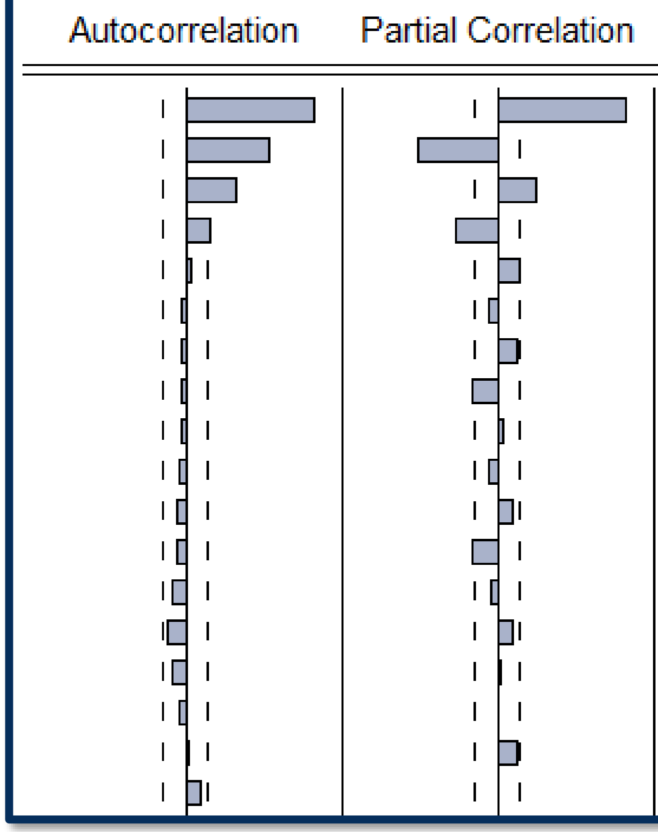
	AC	PAC
1	0.929	0.929
2	0.806	-0.421
3	0.707	0.300
4	0.635	-0.073
5	0.580	0.075
6	0.521	-0.127
7	0.450	-0.035
8	0.375	-0.042
9	0.308	0.005
10	0.248	-0.074
11	0.193	0.013
12	0.157	0.112
13	0.141	0.011
14	0.115	-0.149
15	0.081	0.096
16	0.058	0.013
17	0.039	-0.058
18	0.020	-0.045

The ARMA Process

The ARMA process

- The ACF and the PACF of an ARMA process look like these (in RStudio):

$$y_t = 0.75y_{t-1} + \varepsilon_t + 0.9\varepsilon_{t-1}$$



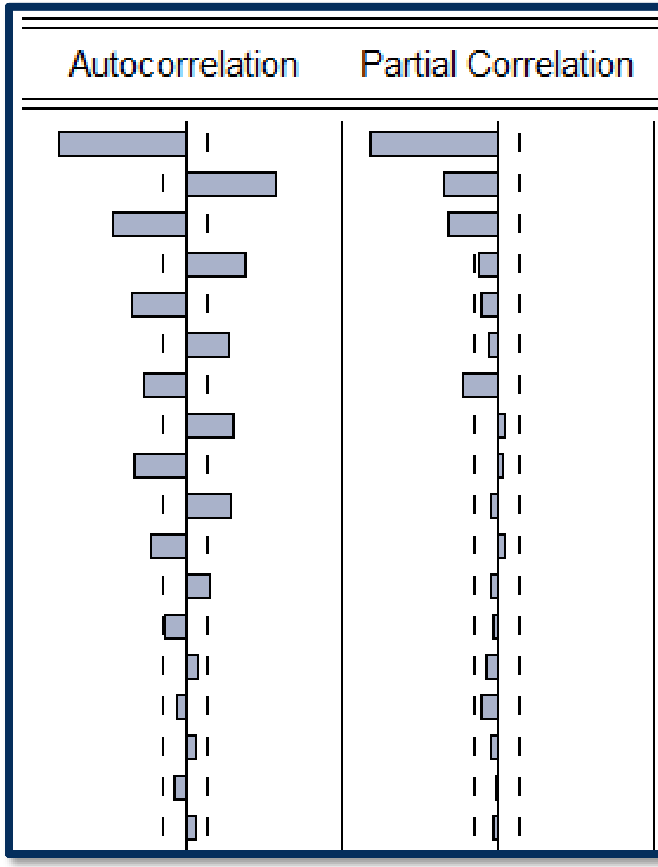
	AC	PAC
1	0.845	0.845
2	0.563	-0.531
3	0.335	0.270
4	0.163	-0.272
5	0.038	0.141
6	-0.028	-0.054
7	-0.031	0.132
8	-0.018	-0.166
9	-0.028	0.038
10	-0.048	-0.060
11	-0.051	0.095
12	-0.054	-0.156
13	-0.091	-0.037
14	-0.118	0.091
15	-0.092	0.031
16	-0.041	0.011
17	0.021	0.127
18	0.094	0.002

The ARMA Process

The ARMA process

$$y_t = -0.6 y_{t-1} + \varepsilon_t - 0.75 \varepsilon_{t-1}$$

- The ACF and the PACF of an ARMA process look like these (in RStudio):



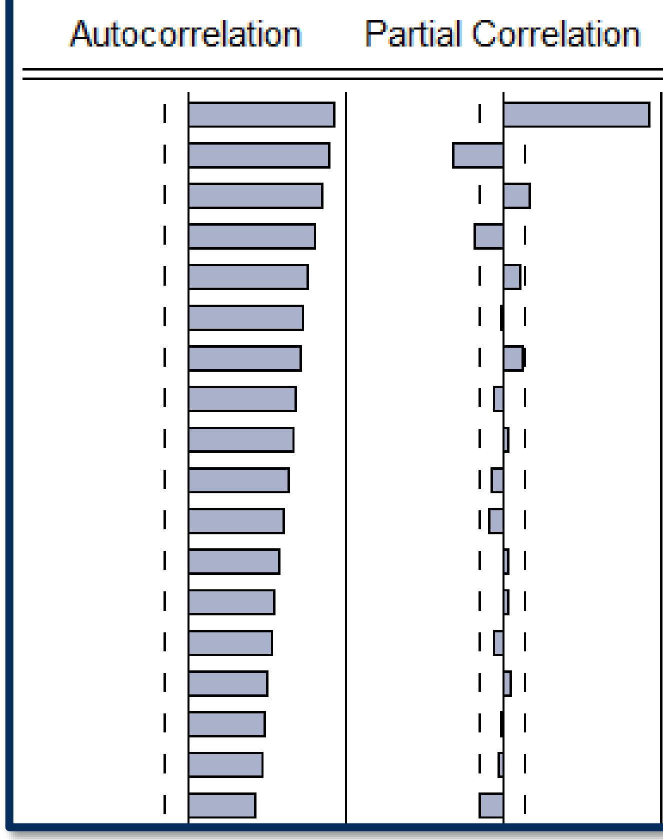
	AC	PAC
1	-0.844	-0.844
2	0.612	-0.353
3	-0.481	-0.317
4	0.406	-0.118
5	-0.348	-0.099
6	0.292	-0.058
7	-0.283	-0.227
8	0.326	0.055
9	-0.341	0.040
10	0.299	-0.041
11	-0.231	0.057
12	0.166	-0.041
13	-0.126	-0.026
14	0.085	-0.080
15	-0.060	-0.100
16	0.067	-0.041
17	-0.078	-0.003
18	0.076	-0.024

The ARMA Process

The ARMA process

- The ACF and the PACF of an ARMA process look like these (in RStudio):

$$y_t = 0.6y_{t-1} + 0.3y_{t-2} + \varepsilon_t + 0.75\varepsilon_{t-1}$$



	AC	PAC
1	0.972	0.972
2	0.926	-0.316
3	0.886	0.176
4	0.844	-0.182
5	0.803	0.121
6	0.767	-0.013
7	0.739	0.131
8	0.714	-0.059
9	0.690	0.045
10	0.667	-0.066
11	0.637	-0.097
12	0.606	0.047
13	0.579	0.048
14	0.553	-0.058
15	0.527	0.053
16	0.506	-0.008
17	0.485	-0.018
18	0.459	-0.144

The ARMA Process

The ARMA process

- The ACF and the PACF of an ARMA process look like these (in RStudio):

$$y_t = .6y_{t-1} + .3y_{t-2} + \varepsilon_t + .75\varepsilon_{t-1} + +.5\varepsilon_{t-2}$$

Autocorrelation	Partial Correlation		AC	PAC
		1	0.956	0.956
		2	0.886	-0.329
		3	0.800	-0.155
		4	0.730	0.259
		5	0.661	-0.166
		6	0.591	-0.109
		7	0.514	-0.039
		8	0.431	-0.115
		9	0.353	0.073
		10	0.285	0.027
		11	0.234	0.031
		12	0.189	-0.009
		13	0.151	-0.010
		14	0.112	-0.025
		15	0.079	0.042
		16	0.057	0.055
		17	0.039	-0.121
		18	0.021	-0.024

The ARMA Process

The ARMA process

The ACF and PACF structures of an ARMA process are the result of both the AR and MA terms:

- ACF: Initial values depend on the order q of the MA process and then later a decay as dictated by the AR process.
- PACF: Certain initial values of the PACF depend on the AR order followed by a decay from the MA part.

The ARMA Process

The ARMA process

Process	ACF	PACF
AR(p)	Decaying	First p lags non-null
MA(q)	First q lags non-null	Decaying
ARMA(p,q)	Many lags non-null	Many lags non-null

Workshop

Workshop

Workshop 4: ADF Test and Correlograms

Write an R code that tests whether or not the Peso-Dollar exchange rate (pder) is stationary or not. Note that this part was done for you.

Now, after proper differencing, investigate its dependence structure comment about the process (AR(p) or MA(q)) the time series might follow

Non-stationarity and Integrated Processes

Non-stationarity and Integrated Processes

Handling Seasonality

- Handling seasonality is an important part in ARIMA for this component brings some challenges in the model building process.
- There are two main ways to deal with seasonality: (i) remove seasonality altogether and work with the deseasonalized data, or (ii) include this component in the model.

Non-stationarity and Integrated Processes

Differencing

- Many time series are not stationary with respect to its mean (for instance, most series have an increasing or decreasing trend).
- The transformation called **differencing** is frequently applied to time-series data to induce a stationary mean.

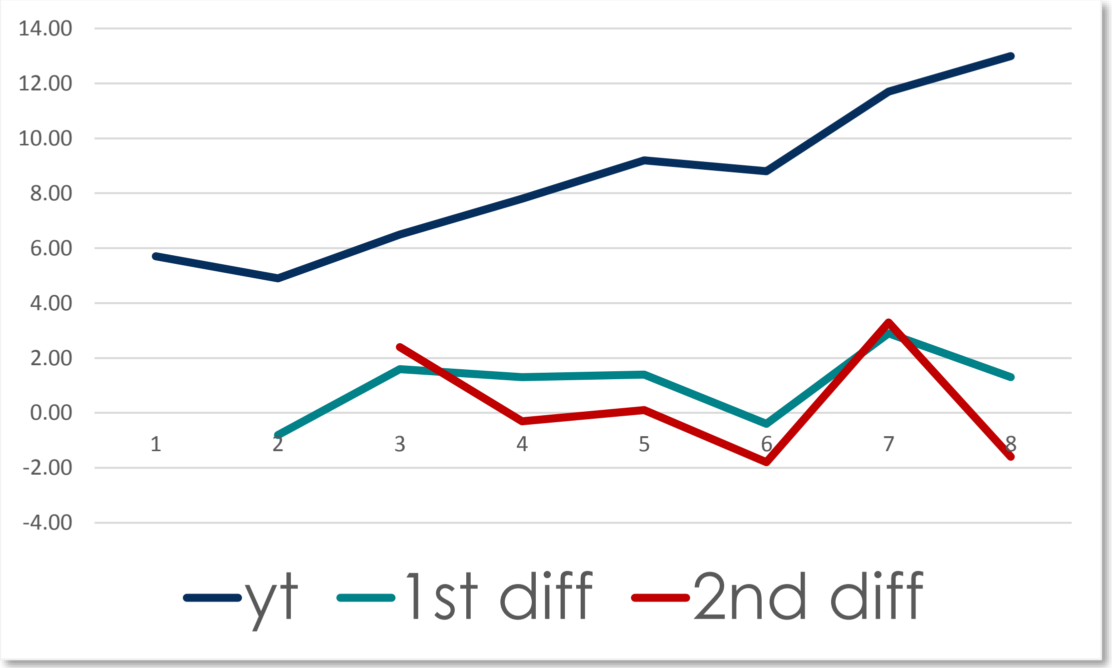
Non-stationarity and Integrated Processes

Differencing and transformation

- Differencing is a relatively simple operation that involves calculating successive pairwise deviations in the values of a data series.

Non-stationarity and Integrated Processes

y_t	1st diff	2nd diff
5.7		
4.9	-0.8	
6.5	1.6	2.4
7.8	1.3	-0.3
9.2	1.4	0.1
8.8	-0.4	-1.8
11.7	2.9	3.3
13.0	1.3	-1.6



Non-stationarity and Integrated Processes

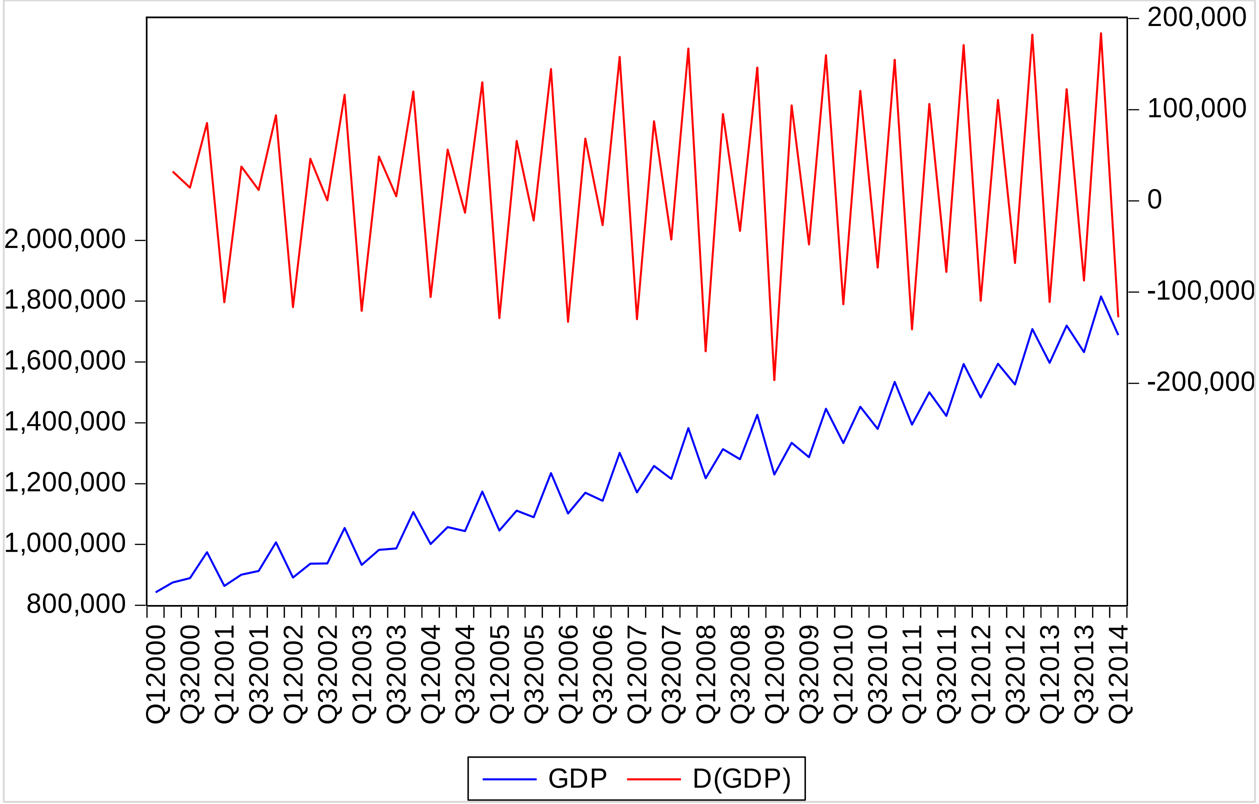
Differencing and transformation

- Time series transformation are done to achieve stationarity in fluctuations.
- The usual function used in transformation is the log function.

Non-stationarity and Integrated Processes

Differencing and transformation

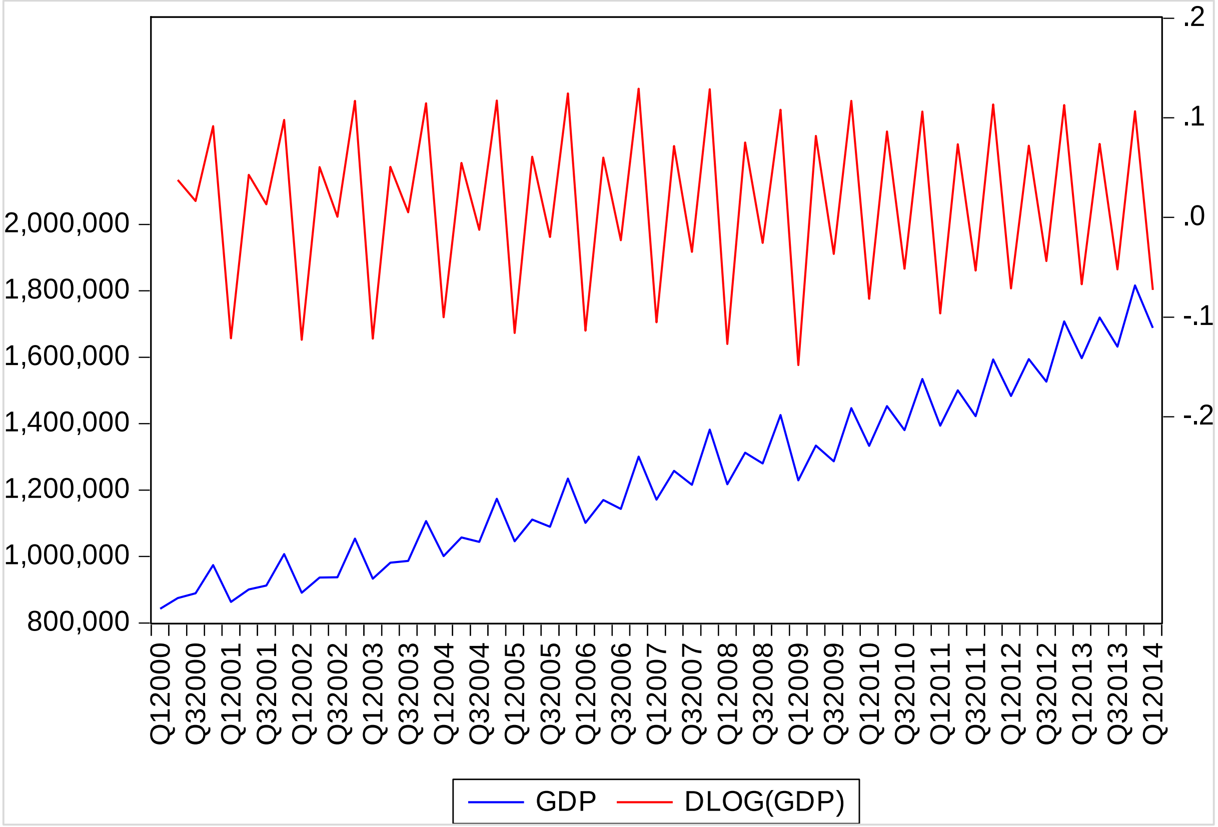
Here's a line chart of gross domestic product GDP and its differenced series D(GDP).



Non-stationarity and Integrated Processes

Differencing and transformation

Here's a line chart of gross domestic product GDP and its transformed and differenced series D(GDP).



Non-stationarity and Integrated Processes

Order of integration

- A time series is said to be **integrated of order d** if and only if it has to be differenced d times before becoming stationary.
- If $\mathbf{y_t}$ has to be differenced **once** before becoming stationary with respect to its mean, then $\mathbf{y_t}$ is **integrated of order 1**, denoted by $\mathbf{y_t \sim I(1)}$. The notation I stands for the "I", or **Integrated** in ARIMA.

Non-stationarity and Integrated Processes

Order of integration

- I(1) processes need only to be differenced once to induce stationarity.
- There are some series which need to be differenced more than once to induce stationarity. These are I(d) processes where d is greater than 1.
- I(0) time series need not be differenced for they are already stationary.

Estimating the ARIMA Model in R studio

Estimating the ARIMA Model in R studio

Takeaway Points

- In estimating an ARIMA(p,d,q) model, it is important to know what values to supply for the order parameters:
  - p : AR order
  - d : Order of integration
  - q : MA order

Estimating the ARIMA Model in R studio

Takeaway Points

- While p and q may be determined using **correlograms**, the value of d is the number of differencing used to achieve stationarity of the series.
- Thus, if a series is differenced once before it achieves stationarity, d=1.

Estimating the ARIMA Model in R studio

Takeaway Points

- When the time series follows either a pure AR or pure MA process, it is fairly easy to see using the ACF and PACF the AR/MA order.
- On the other hand, for ARIMA models where both the ACF and PACF taper off, knowing which p or q to specify becomes less clear. In such a case, some form of **automation** might come in handy.

Autocorrelations and Correlograms

ARIMA Model in R studio

```
pder <- read.table("pder.txt", header=T)
head(pder)
pder.ts <- ts(pder$pder,frequency=12,start=c(1980,1))
head(pder.ts)

adf.test(pder.ts, alternative="stationary")
adf.test(diff(pder.ts), alternative="stationary")

plot(pder.ts)
plot(diff(pder.ts))

acf(diff(pder.ts))
pacf(diff(pder.ts))

pder.arima <- arima(pder.ts,order=c(0,1,2))
pder.arima
pder.pred <- forecast(pder.arima, h = 48, level=c(97.5))
plot(pder.pred)
```

Estimating the ARIMA Model in R studio

ARIMA Model in R studio

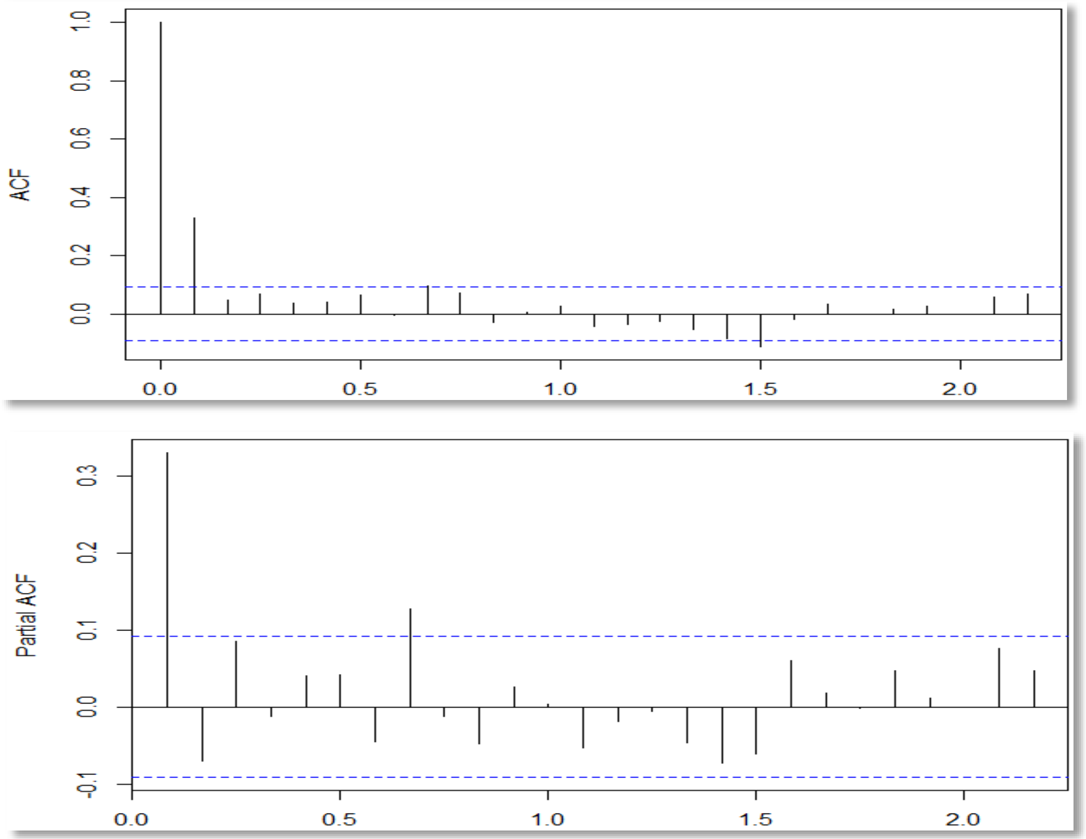
Output suggests 1^{st} differencing to achieve stationarity (i.e., d=1)

```
> adf.test(pder.ts, alternative="stationary")

        Augmented Dickey-Fuller Test

data:  pder.ts
Dickey-Fuller = -1.6761, Lag order = 7, p-value = 0.7149
alternative hypothesis: stationary

> adf.test(diff(pder.ts), alternative="stationary")

        Augmented Dickey-Fuller Test

data:  diff(pder.ts)
Dickey-Fuller = -6.0099, Lag order = 7, p-value = 0.01
alternative hypothesis: stationary
```

Estimating the ARIMA Model in R studio

ARIMA Model in R studio

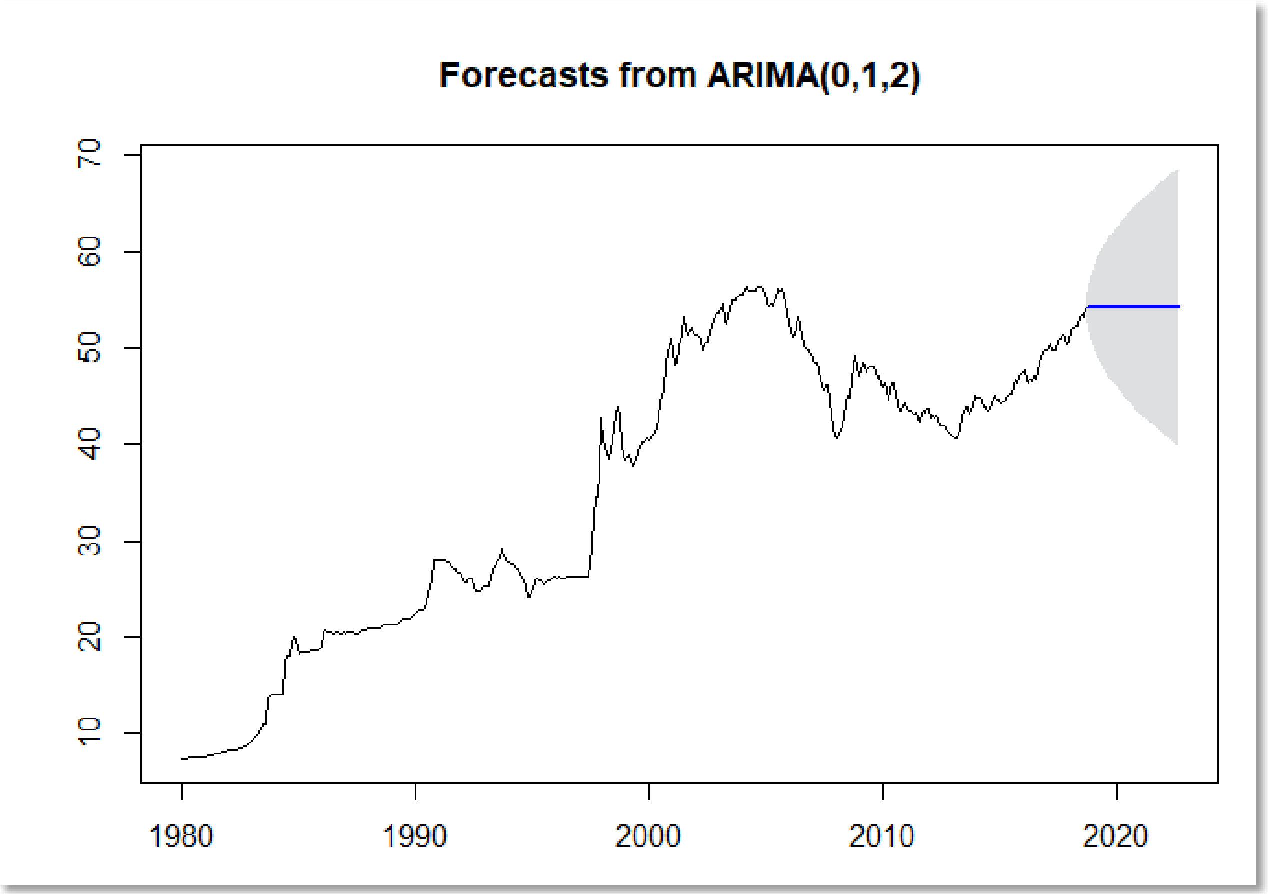
ACF of the differenced series cuts off after lag=2 while PACF tapers off, suggesting an MA(q=2) process.



Estimating the ARIMA Model in R studio

ARIMA Model in R studio

The blue line are the forecasts for the series. Note how the gray confidence bands open wide as we go farther in the horizon.



Autocorrelations and Correlograms

ARIMA Model in R studio

- Alternatively, the auto.arima function lets the program decide for the best ARIMA model to fit the data. It can even accommodate seasonal components.

```
pder.aarima<-auto.arima(pder.ts,max.order = 12,trace=TRUE,seasonal=TRUE)
pder.aarima
pder.pred <- forecast(pder.aarima, h = 48, level=c(97.5))
plot(pder.pred)
```

Workshop

Workshop

ARIMA Model in R studio

Perform the unit root test on the variable *export*. Do the necessary transformations.

For this case, one might not just use the diff() function, but as well as the diff(log(export)) function to address *export*'s nonstationarity.

Perform an auto.arima on the original series, forecasting h=48 periods. Output forecast graphs.

Workshop

ARIMA Model in R studio

Do the same procedure for the variable *import*.

Thank You