comillas.edu

Machine Learning

Chapter 4: Forecasting II

March 2021

Contents

1

Basic Linear Processes

Basic Linear Processes
Definition

- A linear process can be represented as a linear combination of random variables (Box-Jenkins):

$$y[t] = \mu + \sum_{i=0}^{\infty} \psi_i \, \varepsilon[t-i]$$

where μ is the mean of $y[t]$, $\psi_0 = 1$ and $\{\varepsilon[t]\}$ is a sequence of iid random variables with zero mean and well defined distribution.

- We will focus on 3 types of linear processes:
    - White noise processes
    - Autoregressive processes
    - Moving average processes

Fundamental concepts
White noise process

- **Definition**: sequence of uncorrelated random variables, identically distributed with zero mean and constant variance.

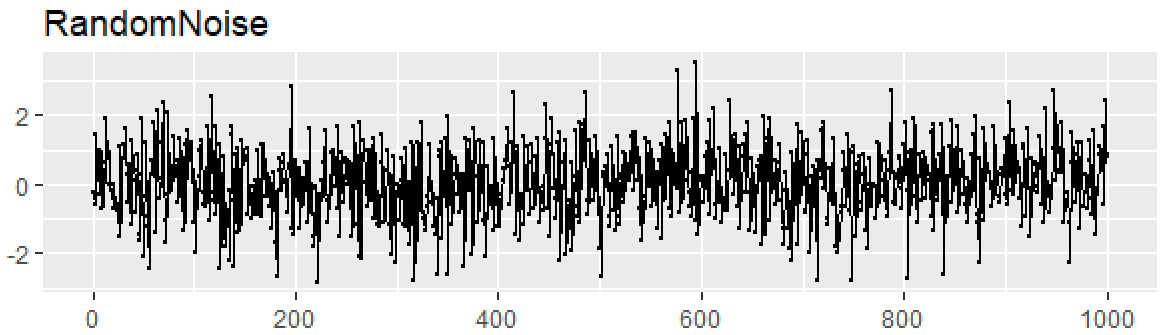
- **General expression:** $y[t] = \varepsilon[t]$

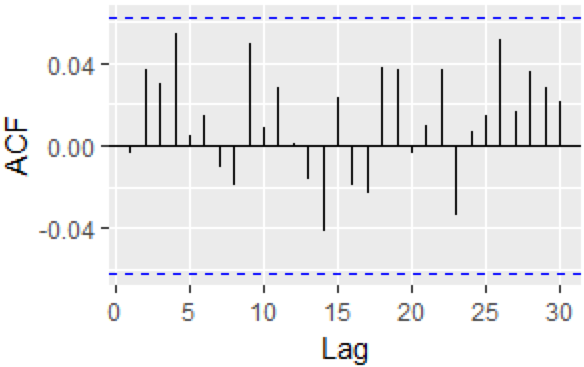
- **Properties:**

$$E(\varepsilon[t]) = 0 \qquad \forall t$$
$$E(\varepsilon[t]^2) = \sigma^2 \qquad \forall t$$
$$E(\varepsilon[t]\varepsilon[t']) = 0 \qquad \forall t \neq t'$$

- **Others:**
  - Independent or strict white noise
  - Gaussian white noise

White noise process





No significant autocorrelation coefficients

Autoregressive processes

- Process *AR(p)* (Yule, 1927):

$$y[t] = \phi_1 y[t-1] + \phi_2 y[t-2] + \ldots + \phi_p y[t-p] + \varepsilon[t]$$

or:

$$\phi(B) y[t] = \varepsilon[t]$$

where:

$$\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \ldots - \phi_p B^p$$

and B is the backshift operator: $By[t] = y[t-1]$

Basic Linear Processes
Autoregressive processes

- For an *AR(p)* to be stationary, the roots of its characteristic polynomial:

$$1 - \phi_1 B - \phi_2 B^2 - \ldots - \phi_p B^p = 0$$

have to lie outside the unit circle.

Autoregressive processes

- If we include a constant term:

$$y[t] = \phi_1 y[t-1] + \phi_2 y[t-2] + \ldots + \phi_p y[t-p] + \delta + \varepsilon[t]$$

then, under the assumption of stationarity:

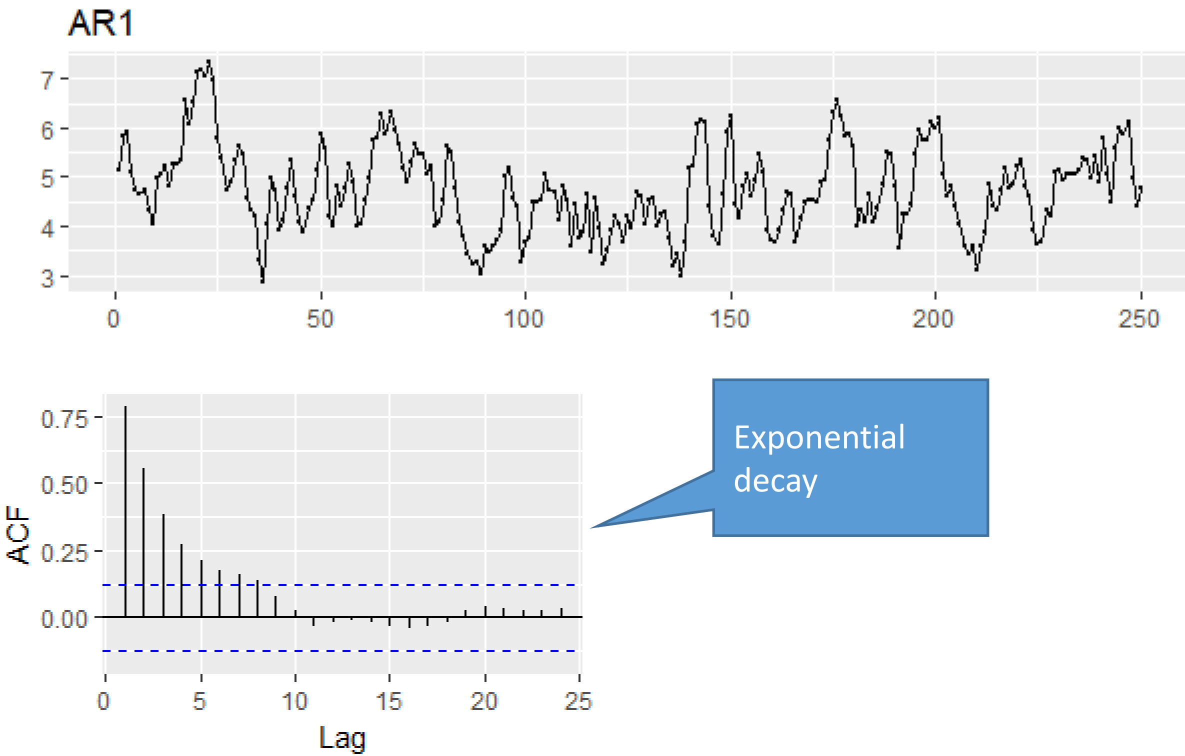
$$\mu = E(y[t]) = \frac{\delta}{1 - \phi_1 - \ldots - \phi_p} \quad \forall t$$

- On the other hand, with $\delta = 0$:

$$\gamma_0 = \phi_1 \gamma_1 + \ldots + \phi_p \gamma_p + \sigma_\varepsilon^2$$

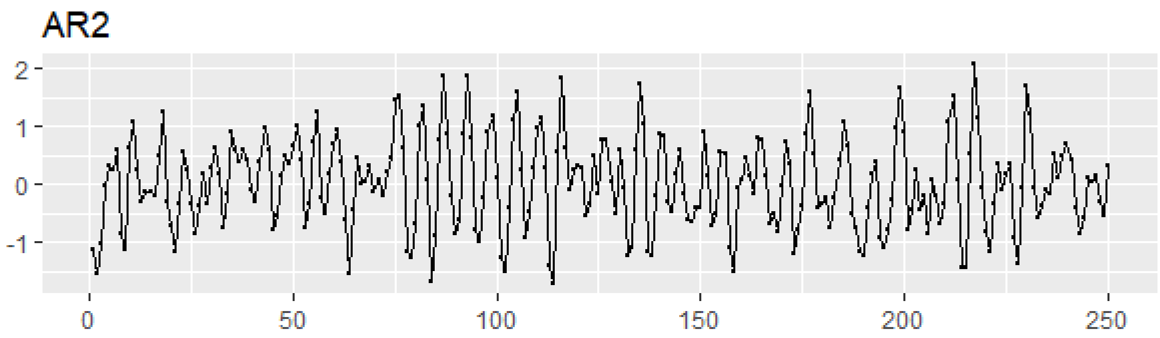
$$\gamma_\tau = \phi_1 \gamma_{\tau-1} + \ldots + \phi_p \gamma_{\tau-p} \qquad \text{for} \quad \tau > 0$$

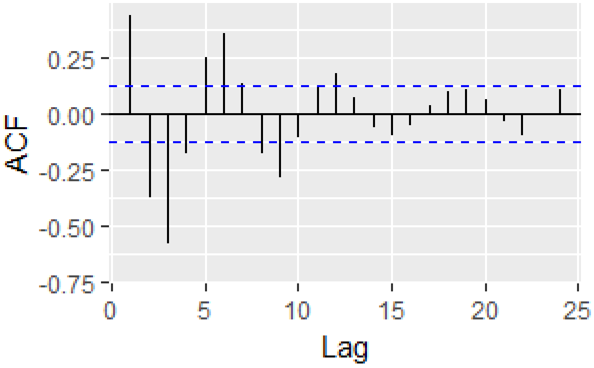
Basic Linear Processes
Autoregressive processes



Exponential decay

Autoregressive processes





Moving Average processes

- *MA(q)* process (Yule, 1921):

$$y[t] = \varepsilon[t] - \theta_1 \varepsilon[t-1] - \theta_2 \varepsilon[t-2] - \dots - \theta_q \varepsilon[t-q]$$

or:

$$y[t] = \theta(B)\varepsilon[t]$$

where:

$$\theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q$$

Moving Average processes

- It is possible to write any stationary *AR(p)* model as an *MA(∞)* :

$$
\begin{aligned}
y[t] &= \varphi y[t-1] + \varepsilon[t] \\
&= \varphi(\varphi y[t-2] + \varepsilon[t-1]) + \varepsilon[t] \\
&= \varphi^2 y[t-2] + \varphi\varepsilon[t-1] + \varepsilon[t] \\
&= \varphi^3 y[t-3] + \varphi^2\varepsilon[t-2] + \varphi\varepsilon[t-1] + \varepsilon[t]
\end{aligned}
$$

- For a *MA(q)* process to be invertible, the roots of the polynomial:

$$1 - \theta_1 B - \theta_2 B^2 - \ldots - \theta_q B^q = 0$$

have to lie outside the unit circle.

If the *MA(q)* process is invertible, then it can be written as an *AR(∞)*.

Moving Average processes

- If we include a constant term:

$$y[t] = \delta + \varepsilon[t] - \theta_1 \varepsilon[t-1] - \theta_2 \varepsilon[t-2] - \ldots - \theta_q \varepsilon[t-q]$$

then:

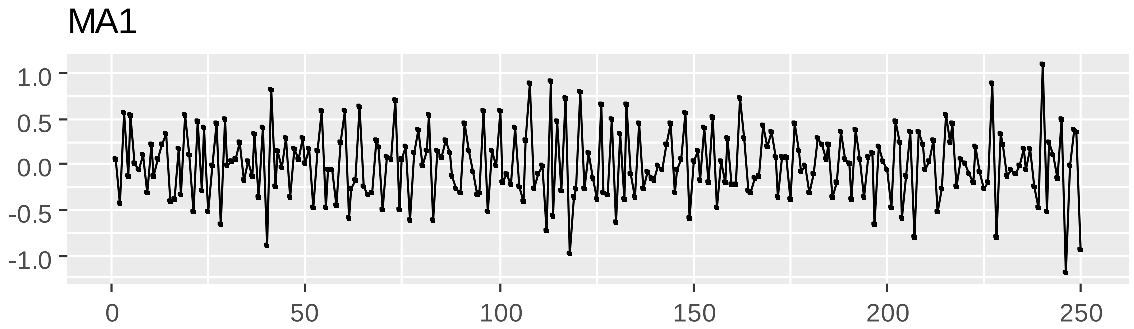
$$\mu = E(y[t]) = \delta \quad \forall t$$

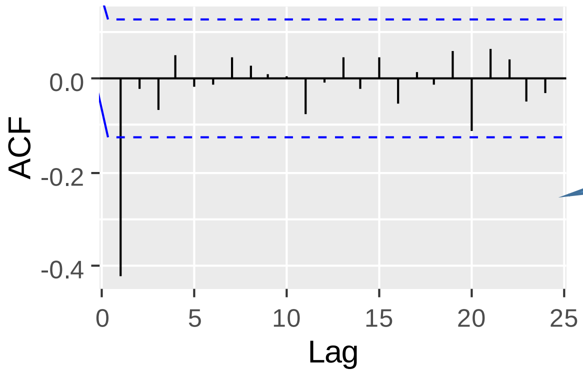
- On the other hand, with $\delta = 0$:

$$\gamma_0 = (1 + \theta_1^2 + .. + \theta_q^2)\sigma_\varepsilon^2$$

$$\gamma_\tau = \begin{cases} (-\theta_\tau + \theta_1 \theta_{\tau+1} + \ldots + \theta_{q-\tau}\theta_q) & \text{for } \tau = 1, 2, \ldots, q \\ \boxed{0 \qquad \text{for } \tau > q} \end{cases}$$

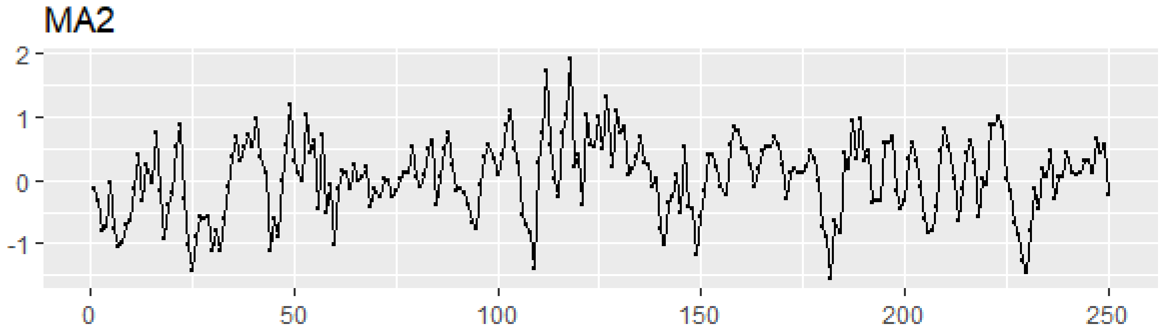
Moving Average processes

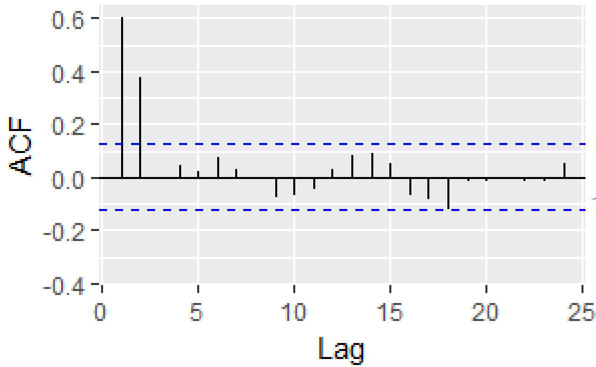




Only one significant autocorrelation coefficient

Moving Average processes





Only two significant autocorrelation coefficients

ARMA processes

- *ARMA(p,q)* process (Wold, 1938):

$$y[t]-\phi_1 y[t-1]-\ldots-\phi_p y[t-p]=\varepsilon[t]-\theta_1\varepsilon[t-1]-\ldots-\theta_q\varepsilon[t-q]$$

or: $$\phi(B)y[t]=\theta(B)\varepsilon[t]$$

ARMA processes

- For an *ARMA(p,q)* process to be stationary, the roots of the polynomial:

$$1-\phi_1 B-\phi_2 B^2-\ldots-\phi_p B^p=0$$

have to lie outside the unit circle.

- For an *ARMA(p,q)* process to be invertible, the roots of the polynomial:

$$1-\theta_1 B-\theta_2 B^2-\ldots-\theta_q B^q=0$$

have to lie outside the unit circle.

ARMA processes

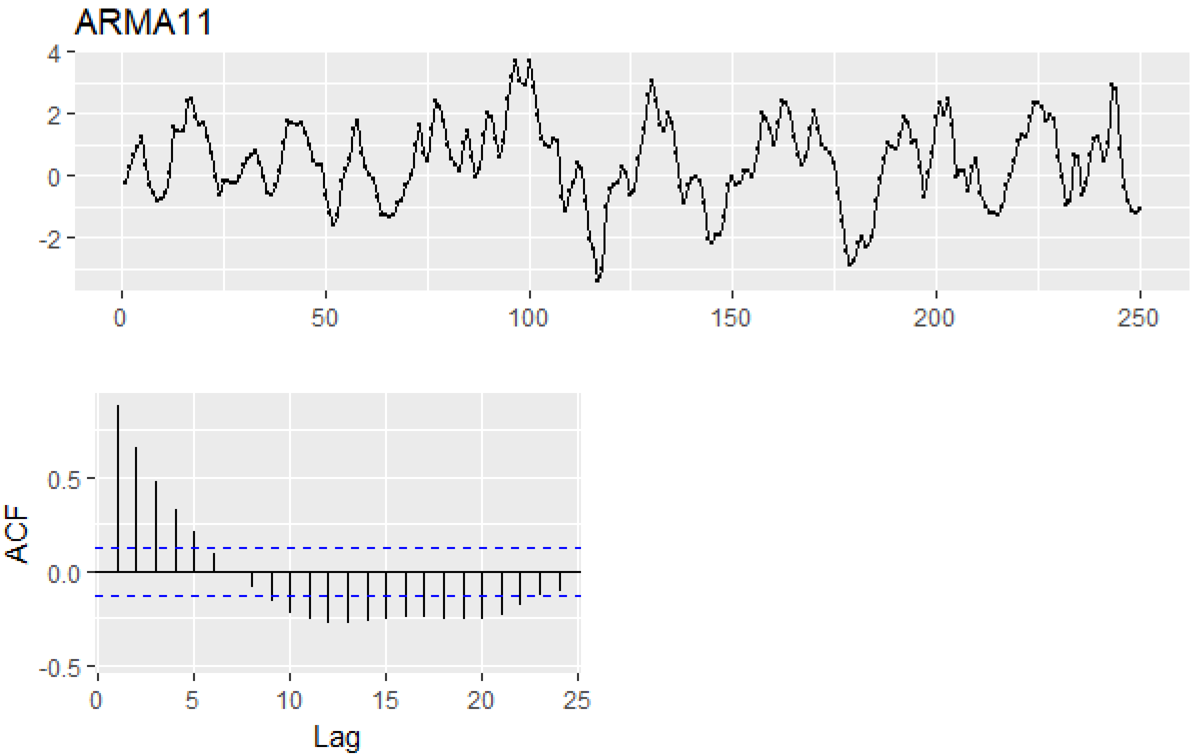
- If we include a constant term, under the assumption of stationarity:

$$\mu = E(y[t]) = \frac{\delta}{1-\phi_1-\ldots-\phi_p} \forall t$$

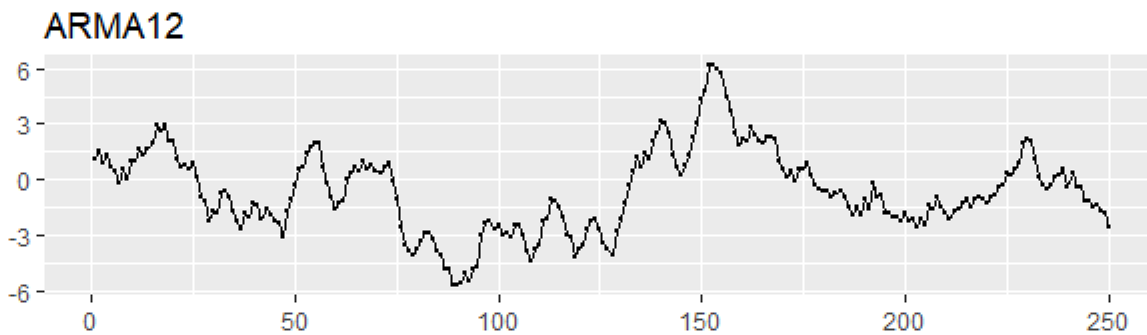
- On the other hand:

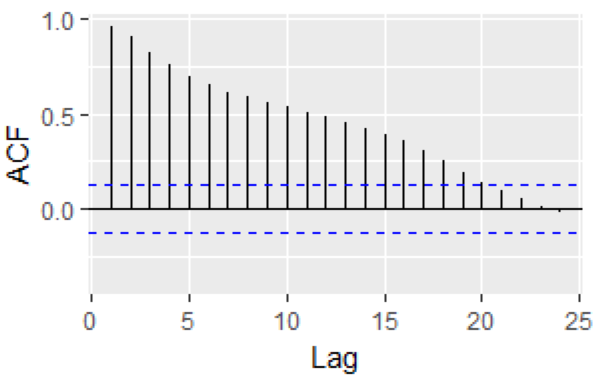
$$\gamma_\tau = \varphi_1\gamma_{\tau-1} + \ldots + \varphi_p\gamma_{\tau-p} \qquad \text{for } \tau > q$$

Basic Linear Processes
ARMA processes



ARMA processes





comillas.edu

2

ARMA Model Identification

ARMA Model Identification
Sample Autocorrelation Function (ACF)

- Autocorrelation:

$$\rho_k = \frac{\gamma_k}{\gamma_0} \quad \rightarrow \quad \hat{\rho}_k = \frac{\hat{\gamma}_k}{\hat{\gamma}_0} = \frac{\sum_{t=1}^{N-k} (y[t+k]-\bar{y})(y[t]-\bar{y})}{\sum_{t=1}^{N} (y[t]-\bar{y})^2}$$

- Under the assumption $\rho_k=0$,

$$\hat{\rho}_k \sim N\left(0, \sigma^2_{\hat{\rho}_k}\right)$$

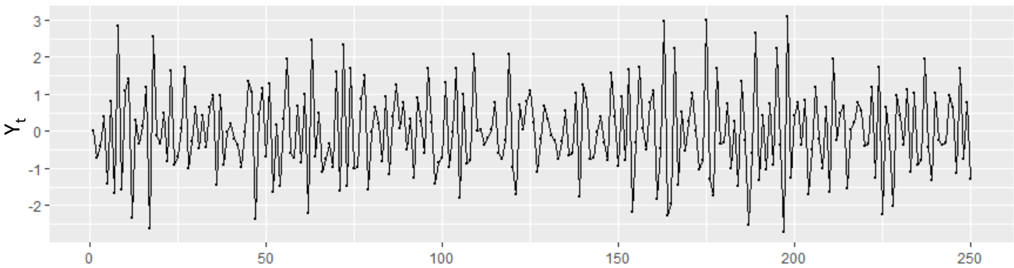
if, in addition, $\{y[t]\}$ is a $MA(q)$ process, then: $\sigma^2_{\hat{\rho}_k} \approx \frac{1}{N}(1+2\sum_{i=1}^{k-1} \hat{\rho}_i^2)$

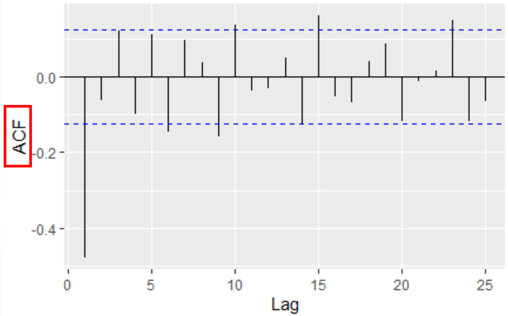
and we can stablish the 95% confidence interval:

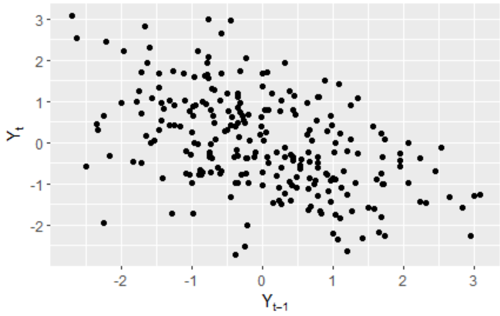
$$\boxed{\hat{\rho}_k \in \left[-1.96\sqrt{\sigma^2_{\hat{\rho}_k}} \; ; +1.96\sqrt{\sigma^2_{\hat{\rho}_k}}\right]}$$

Fundamental concepts
Stationary Processes

- Correlogram $=\{\hat{\rho}_k\}$ for $k=1,\ldots$ (not recommended for $k>N/4$)

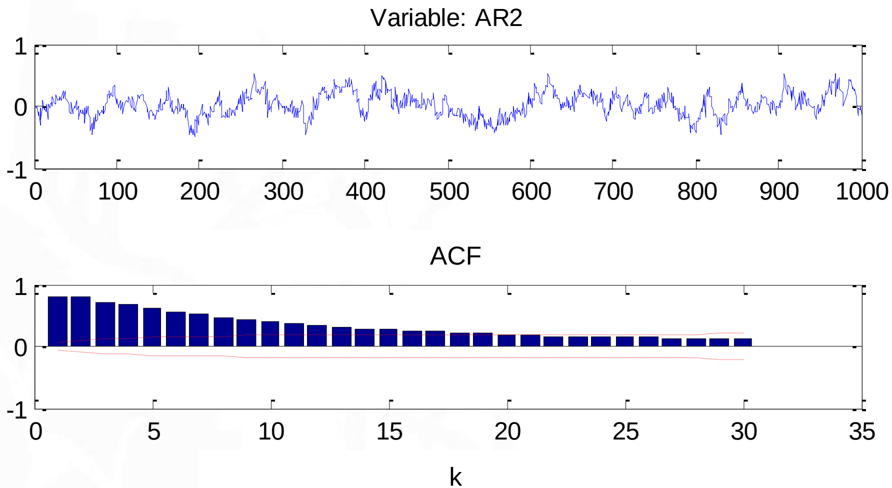






ARMA Model Identification
Partial Autocorrelation Function (PACF)

- For an *AR(p) process*, the ACF decays after *k=p*, but it never reaches 0 $\Rightarrow$ it is not easy to identify an AR process from its ACF



ARMA Model Identification
Partial Autocorrelation Function (PACF)

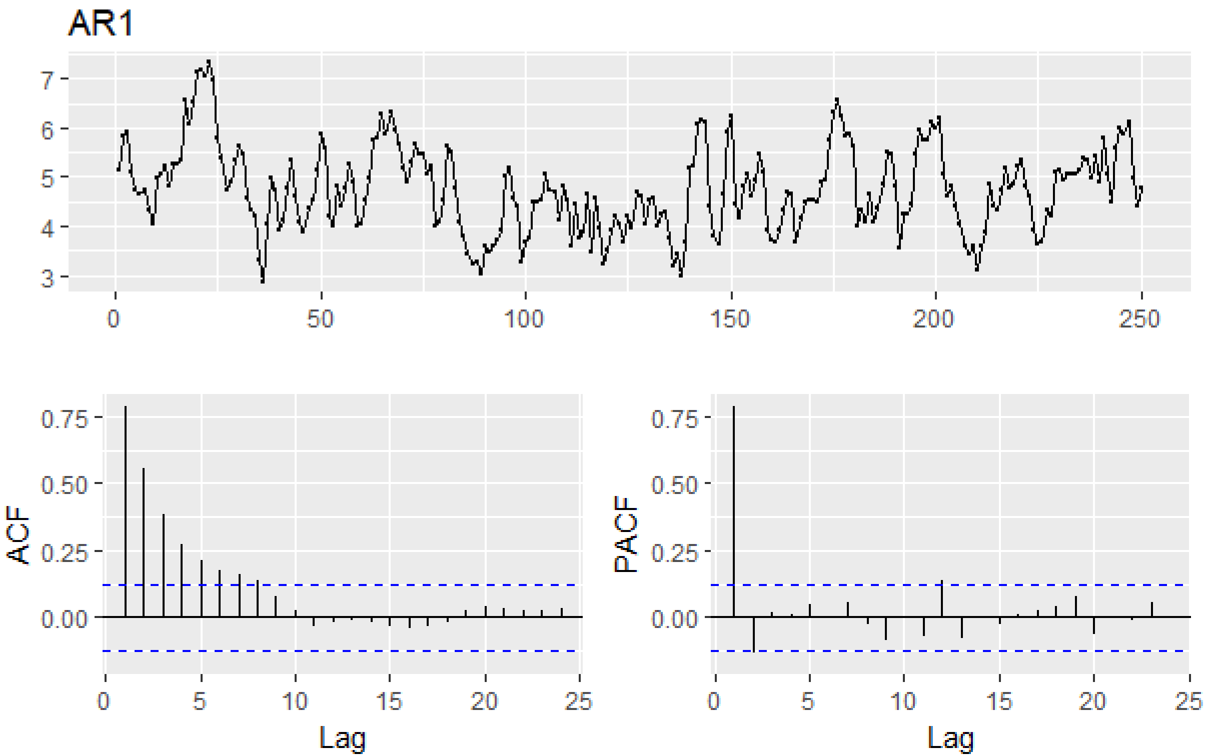
- The PACF can be obtained by linear regression, interpreting each coefficient ϕ_{kk} as the partial correlation between *y[t]* and *y[t-k]* after having eliminated in both variables the effects of *y[t-1],..,y[t-k+1]:*

$$\hat{y}[t] = \hat{\phi}_{1,1} y[t-1]$$
$$\hat{y}[t] = \hat{\phi}_{2,1} y[t-1] + \hat{\phi}_{2,2} y[t-2]$$
$$\ldots$$

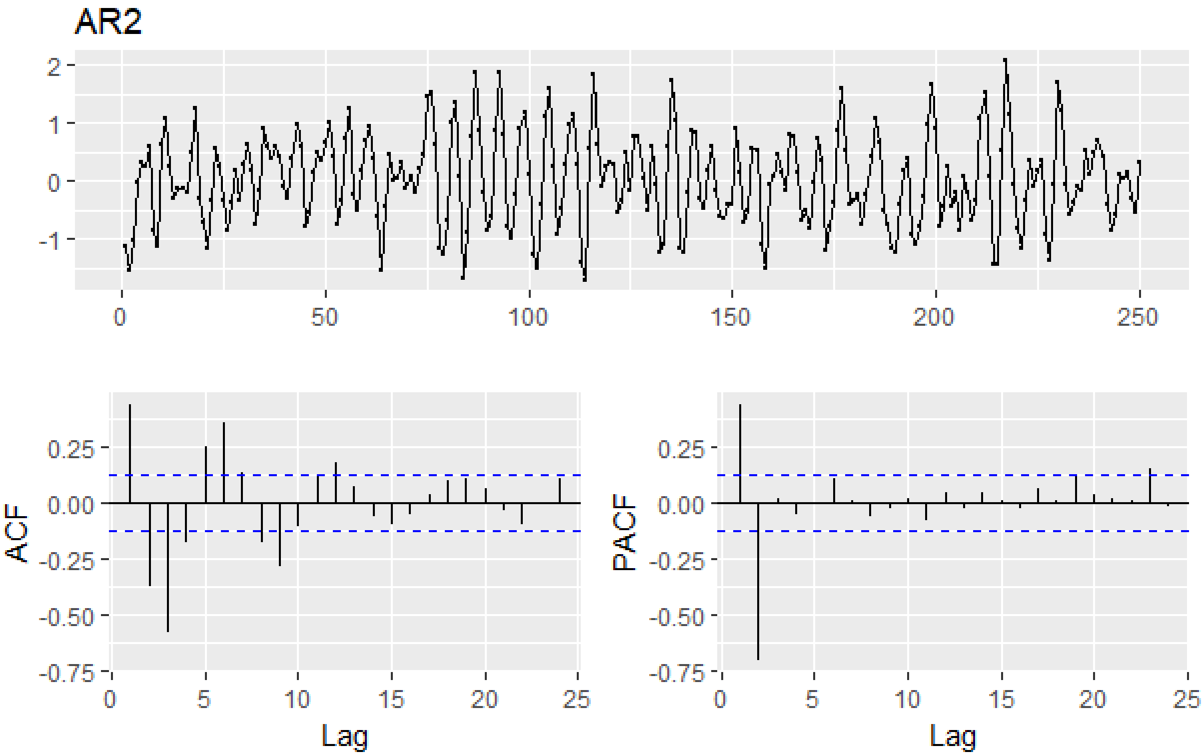
- For an *AR(p)* process and *k>p:*

$$\boxed{\hat{\phi}_{kk} \sim N\left(0\,,\ \frac{1}{N}\right) \quad \Rightarrow \quad \hat{\phi}_{kk} \in \left[-\frac{1.96}{\sqrt{N}}, \frac{1.96}{\sqrt{N}}\right]}$$

ARMA Model Identification
ACF/PACF Examples

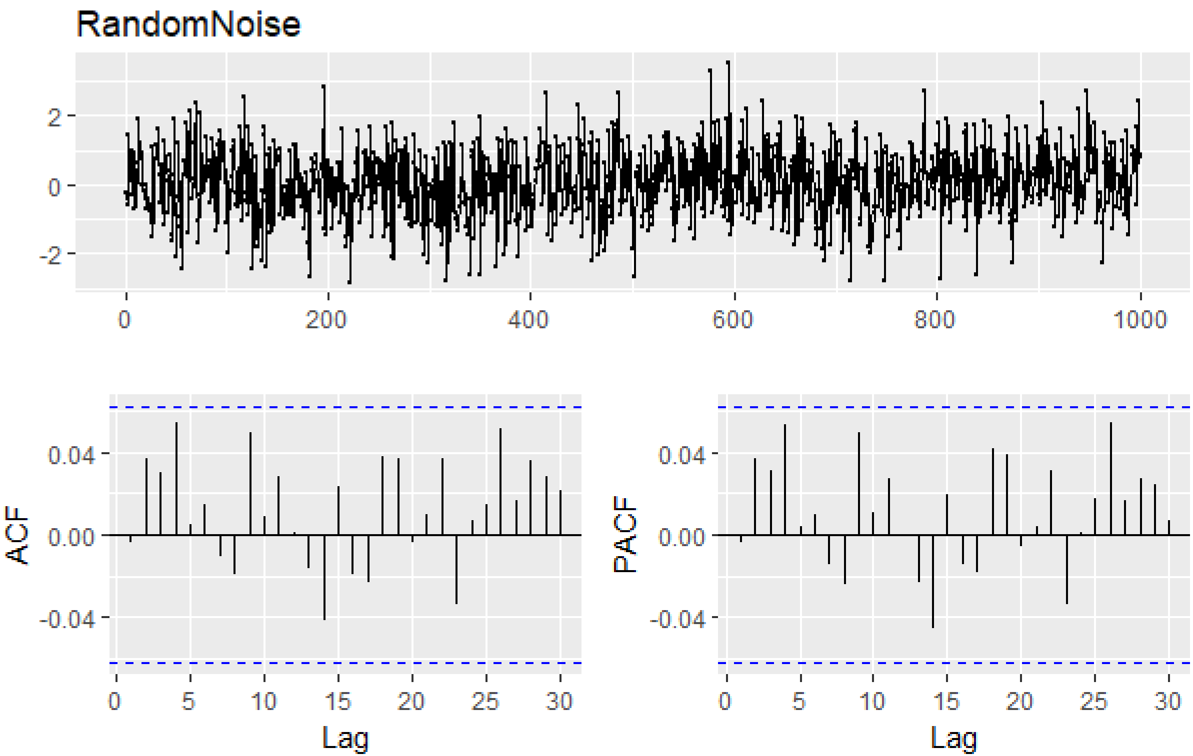


ARMA Model Identification
ACF/PACF Examples

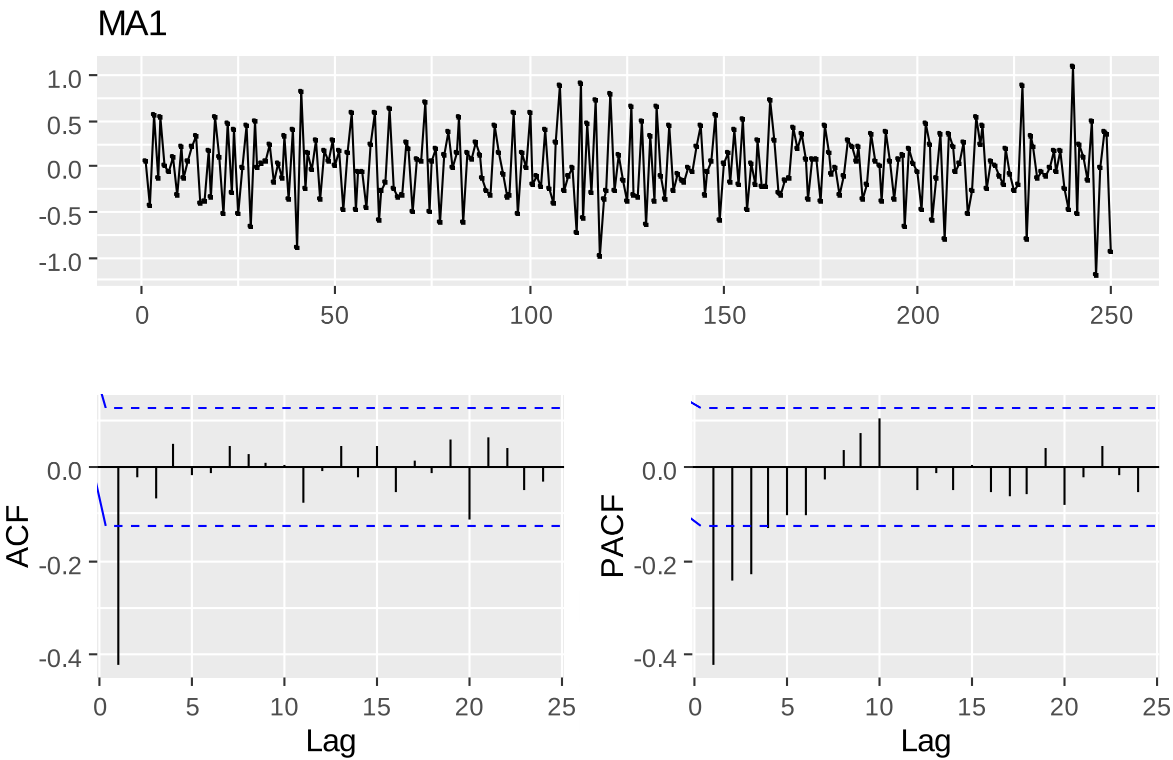


comillas.edu

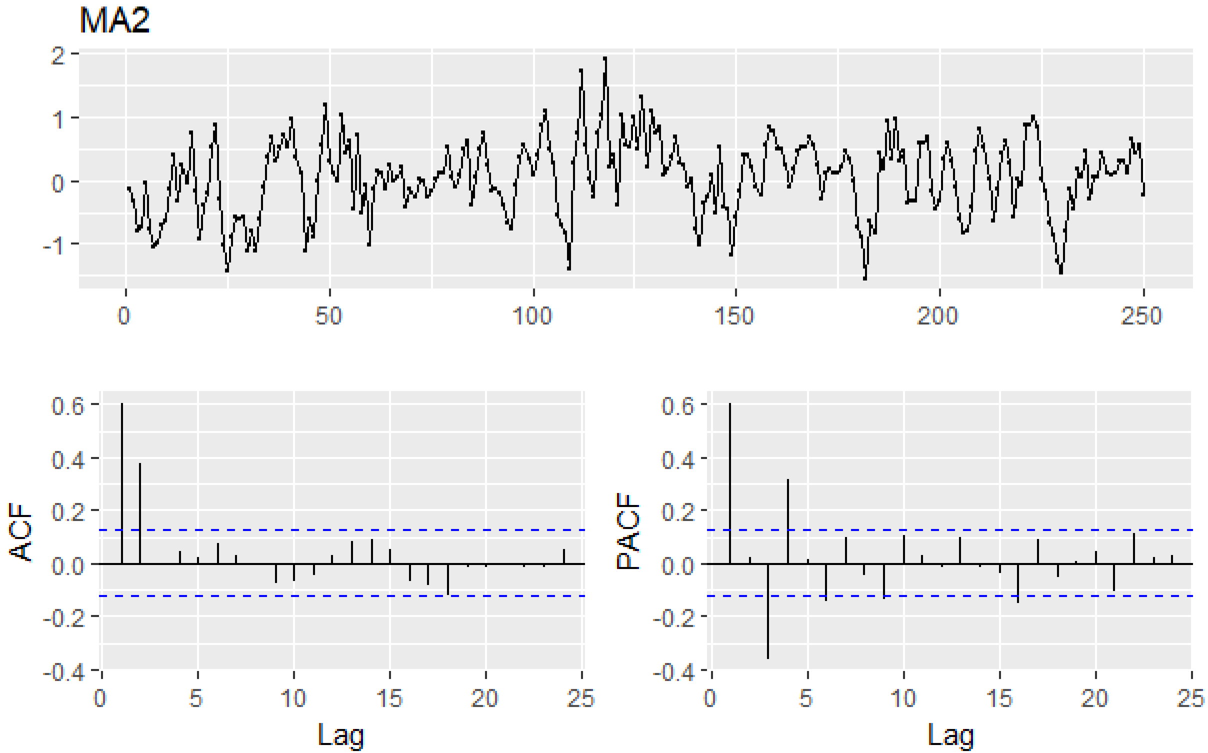
ARMA Model Identification
ACF/PACF Examples



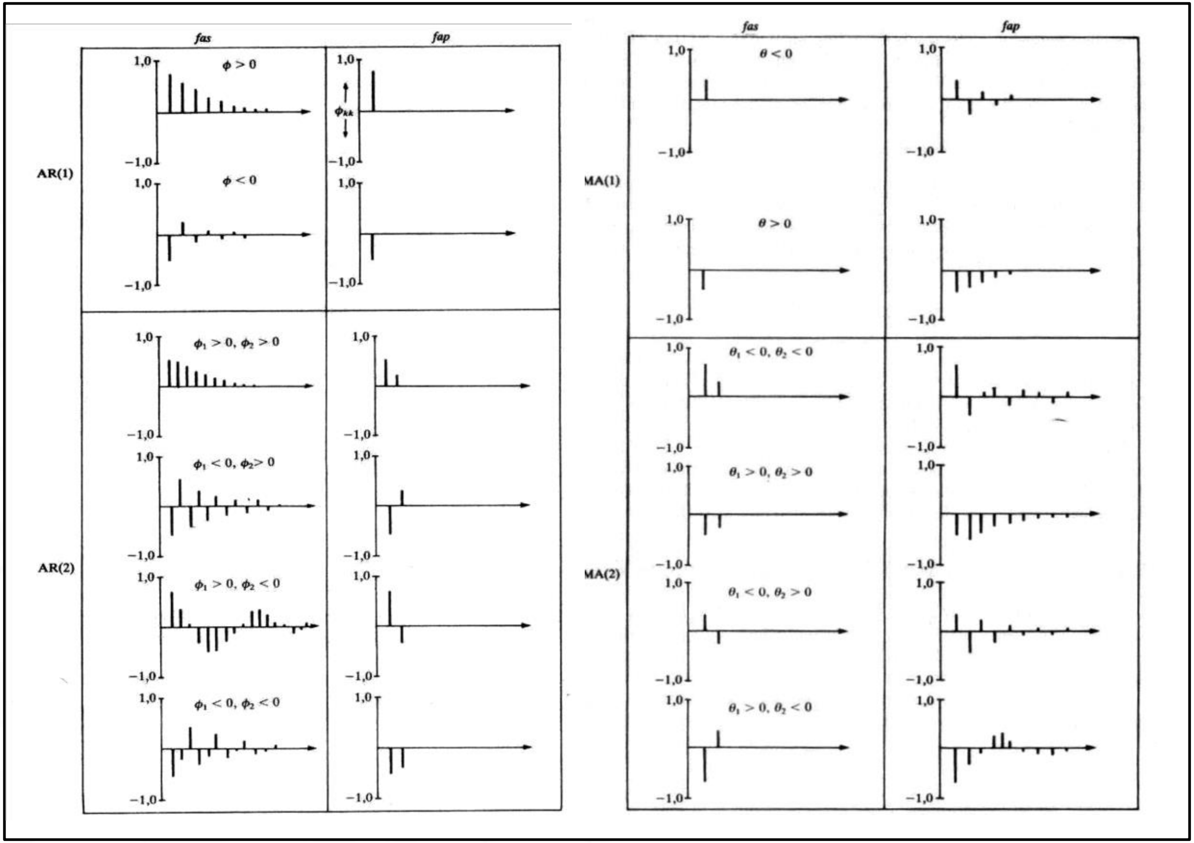
ARMA Model Identification
ACF/PACF Examples



ARMA Model Identification
ACF/PACF Examples

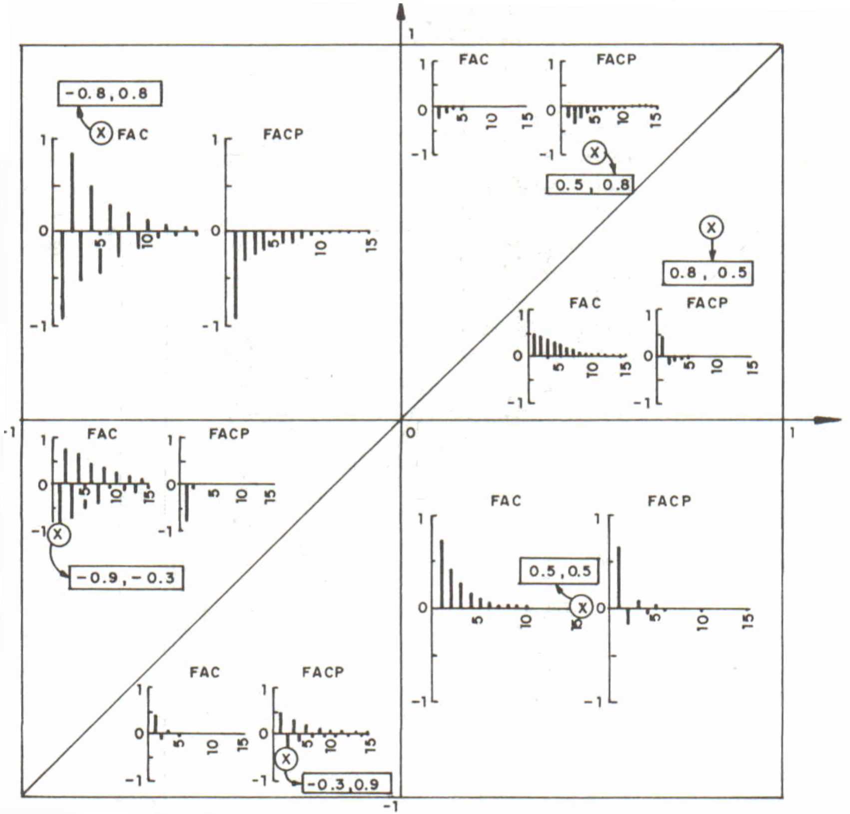


ARMA Model Identification
Identification of AR and MA processes



ARMA Model Identification

Identification of ARMA(1,1) processes



3

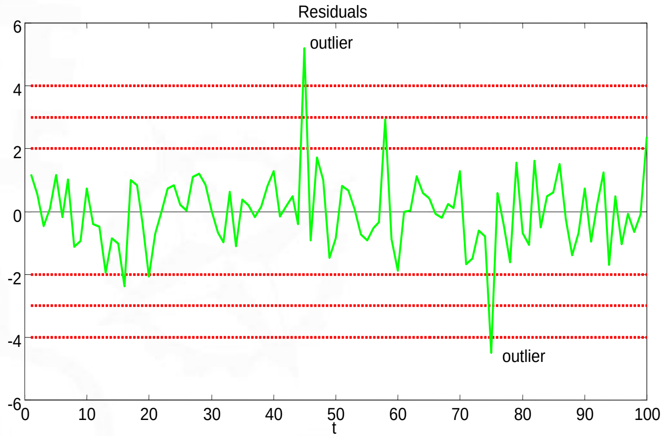
ARMA Model Diagnosis

ARMA Model Diagnosis
The ideal model

- Residuals = White Noise (Gausian)
- Stationary and invertible
- Coefficients are statistically significant and uncorrelated
- Model coefficients are sufficient to represent the series
- High degree of fit compared with other models

ARMA Model Diagnosis
Residual analysis

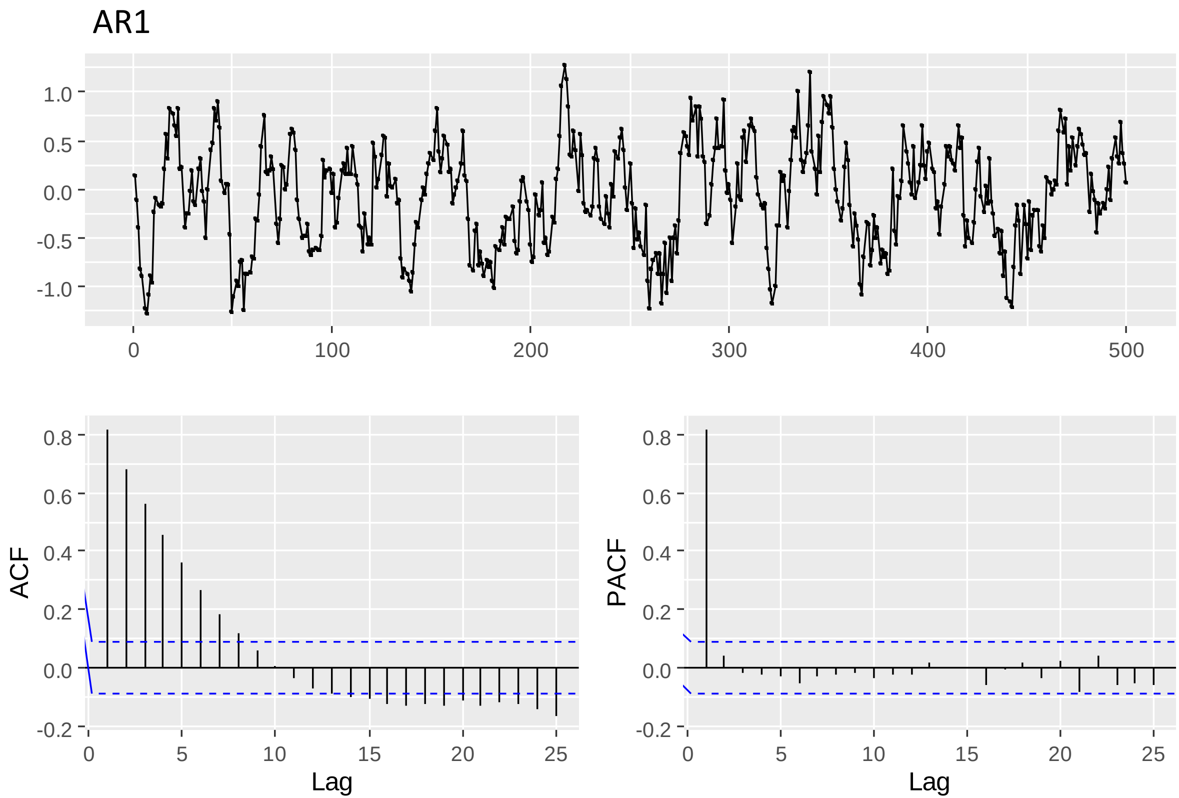
- **Plot of the standarized residuals** with different confidence limits ($\pm 2\sigma_\varepsilon$, $\pm 3\sigma_\varepsilon$, $\pm 4\sigma_\varepsilon$)



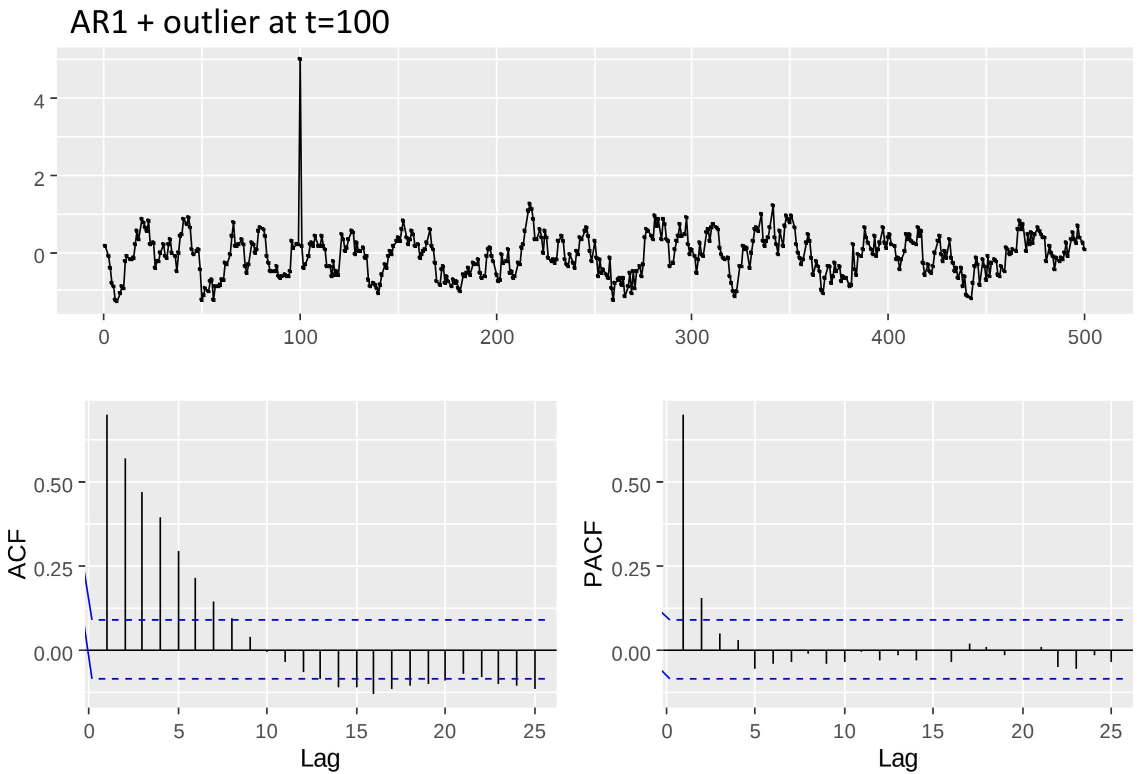
💣 Check for heteroskedasticity (constant variance)

☠ Detection of outliers

ARMA Model Diagnosis
Residual analysis: effect of an outlier



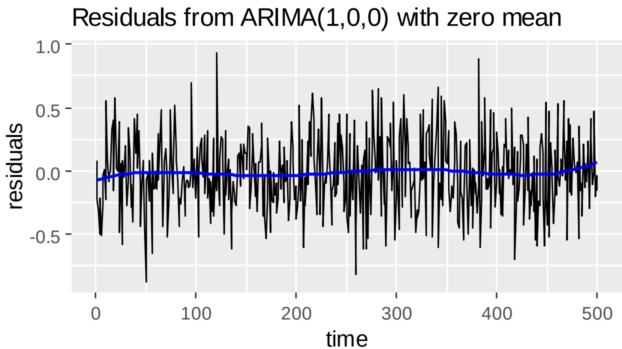
ARMA Model Diagnosis
Residual analysis: effect of an outlier

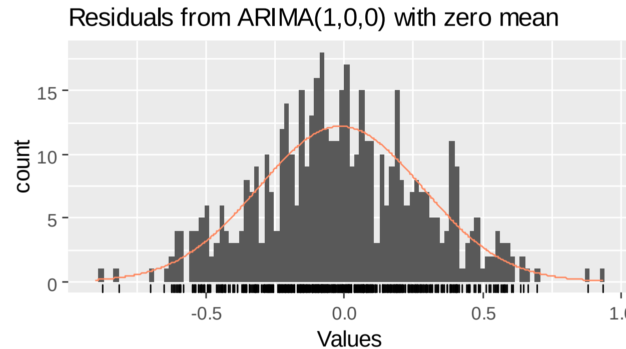


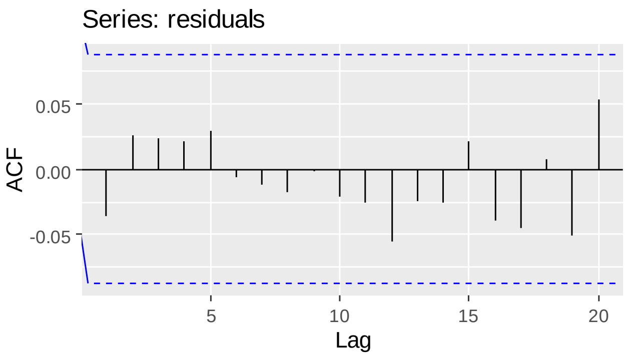
ARMA Model Diagnosis
Residual analysis

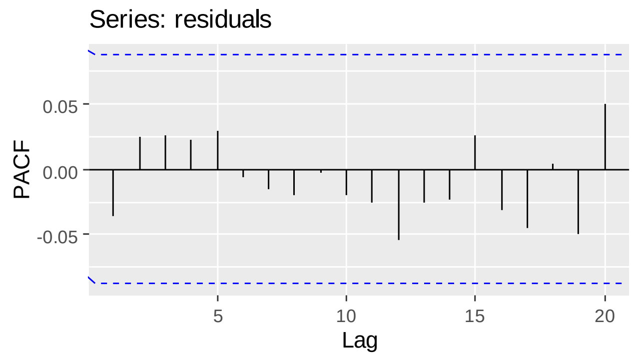
```
     Estimate Std.Error z-value Pr(>|z|)
ar1 0.799547 0.026679 29.969 < 2.2e-16 ***
```

AR1 with phi=0.8





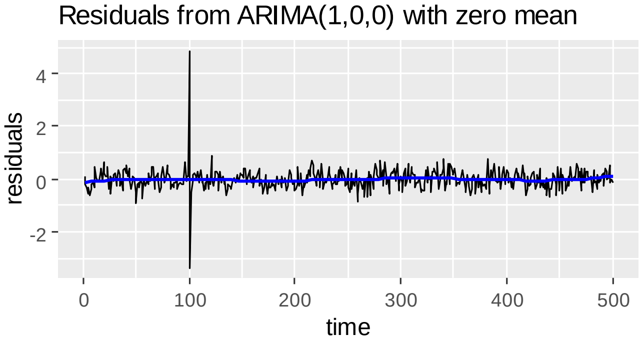


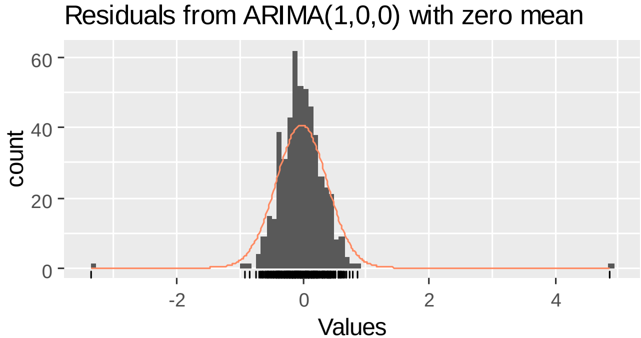


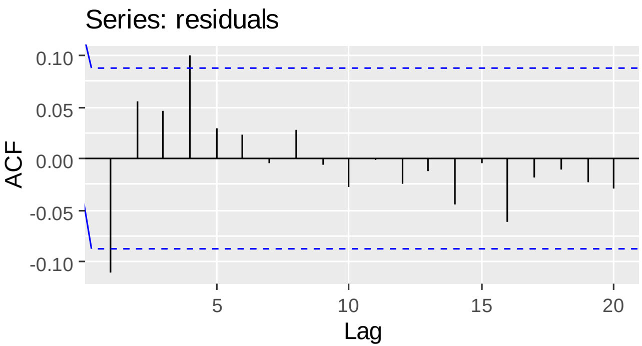
ARMA Model Diagnosis
Residual analysis

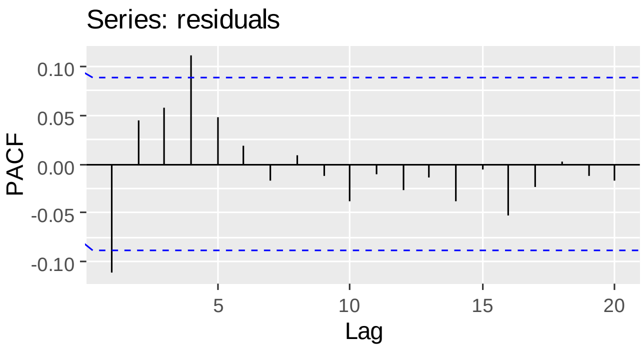
```
     Estimate Std.Error z-value Pr(>|z|)
ar1 0.664459 0.033299 19.954 < 2.2e-16 ***
```

AR1 with phi=0.8 + outlier at t=100









ARMA Model Diagnosis
Residual analysis

- Check the degree of significance of each autocorrelation coefficient. For a white noise process $\{\varepsilon[t]\}$:

$$r_\tau \ \& \ \phi_{\tau\tau} \sim N(0\,,\,1/N) \qquad \text{for all } \tau$$

([*Anderson, 1942*], [*Barlett, 1946*], [*Quenouille, 1949*]).

Therefore we can then stablish the 95% confidence interval:

$$|r_\tau| < \frac{1.96}{\sqrt{N}}$$

$$|\phi_{\tau\tau}| < \frac{1.96}{\sqrt{N}}$$

ARMA Model Diagnosis
Residual analysis

- Check the degree of significance of each autocorrelation coefficient. For a white noise process *{ε[t]}*:

$$r_\tau \ \& \ \phi_{\tau\tau} \sim N(0\,,\,1/N) \qquad \text{for all } \tau$$

([*Anderson, 1942], [Barlett, 1946], [Quenouille, 1949]*).

Therefore we can then stablish the 95% confidence interval:

$$|r_\tau| < \frac{1.96}{\sqrt{N}}$$

$$|\phi_{\tau\tau}| < \frac{1.96}{\sqrt{N}}$$

ARMA Model Diagnosis
Residual analysis

- *Portmanteau* test of the residuals:

Ljung & Box statistic (1978)

$$Q = N(N+2)\sum_{\tau=1}^{M} \frac{r_\tau^2}{N-\tau}$$

Under the null hypothesis that the residuals are independent, Q is distributed according to a χ^2 with *M-p-q* degrees of freedom.

If $Q < \chi^2_{M-p-q}(\alpha)$ accept H_0

If $Q > \chi^2_{M-p-q}(\alpha)$ reject H_0 (tipical values of α :*5%* or *1%*)

ARMA Model Diagnosis
Residual analysis

E: Critical values for χ^2 statistic

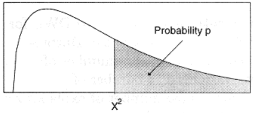
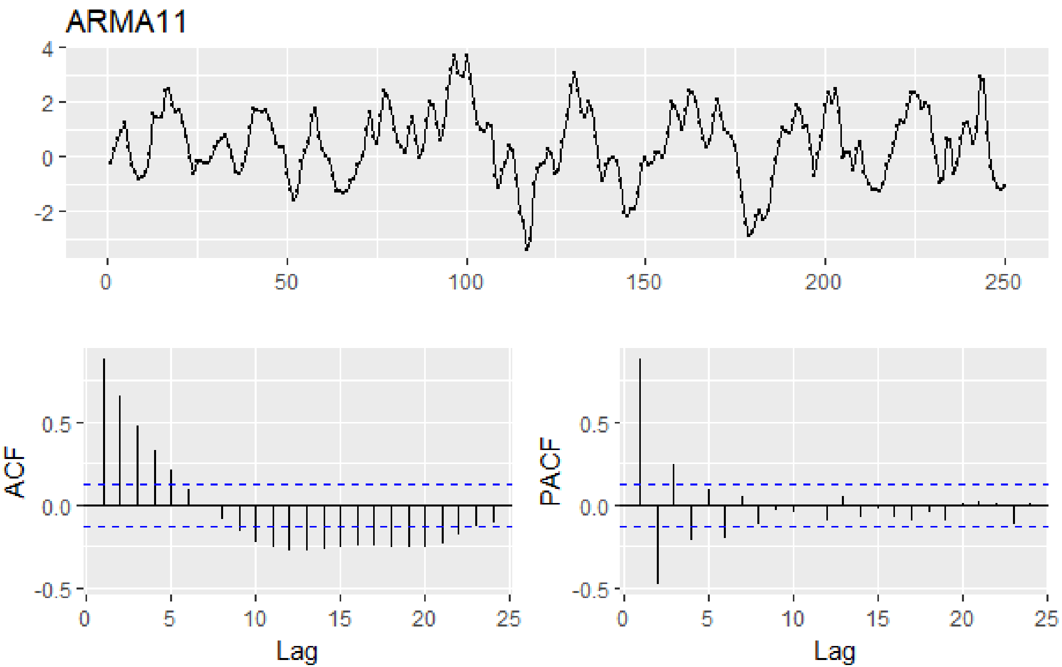


Table entry is the point X^2 with the probability p lying above it. The first column gives the degrees of freedom.

	Probability p					
df	**0.1**	**0.05**	**0.025**	**0.01**	**0.005**	**0.001**
1	2.70	3.84	5.02	6.63	7.87	10.83
2	4.60	5.99	7.37	9.21	10.59	13.82
3	6.25	7.81	9.34	11.34	12.83	16.27
4	7.77	9.48	11.14	13.27	14.86	18.47
5	9.23	11.07	12.83	15.08	16.75	20.52
6	10.64	12.59	14.44	16.81	18.54	22.46
7	12.01	14.06	16.01	18.47	20.27	24.32
8	13.36	15.50	17.53	20.09	21.95	26.12
9	14.68	16.91	19.02	21.66	23.58	27.88
10	15.98	18.30	20.48	23.20	25.18	29.59
11	17.27	19.67	21.92	24.72	26.75	31.26
12	18.54	21.02	23.33	26.21	28.30	32.91
13	19.81	22.36	24.73	27.68	29.82	34.53
14	21.06	23.68	26.11	29.14	31.31	36.12
15	22.30	24.99	27.48	30.57	32.80	37.70
16	23.54	26.29	28.84	32.00	34.26	39.25
17	24.76	27.58	30.19	33.40	35.71	40.79
18	25.98	28.86	31.52	34.80	37.15	42.31
19	27.20	30.14	32.85	36.19	38.58	43.82
20	28.41	31.41	34.17	37.56	39.99	45.31
21	29.61	32.67	35.47	38.93	41.40	46.80
22	30.81	33.92	36.78	40.28	42.79	48.27
23	32.00	35.17	38.07	41.63	44.18	49.73
24	33.19	36.41	39.36	42.98	45.55	51.18
25	34.38	37.65	40.64	44.31	46.92	52.62
26	35.56	38.88	41.92	45.64	48.29	54.05
27	36.74	40.11	43.19	46.96	49.64	55.48
28	37.91	41.33	44.46	48.27	50.99	56.89
29	39.08	42.55	45.72	49.58	52.33	58.30
30	40.26	43.77	46.98	50.89	53.67	59.70
40	51.81	55.76	59.34	63.69	66.77	73.40
50	63.17	67.50	71.42	76.15	79.49	86.66
60	74.40	79.08	83.30	88.38	91.95	99.61
70	85.53	90.53	95.02	100.43	104.21	112.32
80	96.58	101.88	106.63	112.33	116.32	124.84
90	107.56	113.15	118.14	124.12	128.30	137.21
100	118.50	124.34	129.56	135.81	140.17	149.45

ARMA Model Diagnosis
Residual analysis

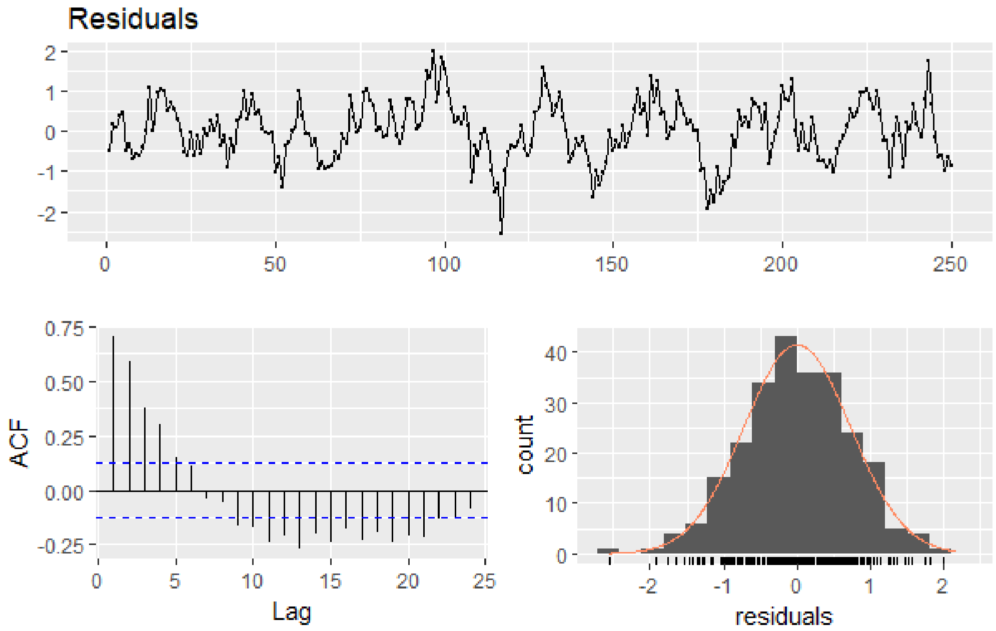
- Analysis of the residuals:
  - Example: $y[t] = 0.8\, y[t-1] - 0.8\, \varepsilon[t-1] + \varepsilon[t]$ with $\varepsilon[t] \sim N(0,0.25)$



ARMA Model Diagnosis
Residual analysis

- Model 1: $\hat{y}[t] = \varphi y[t-1] \quad \Rightarrow \quad \varphi = 0.8899$

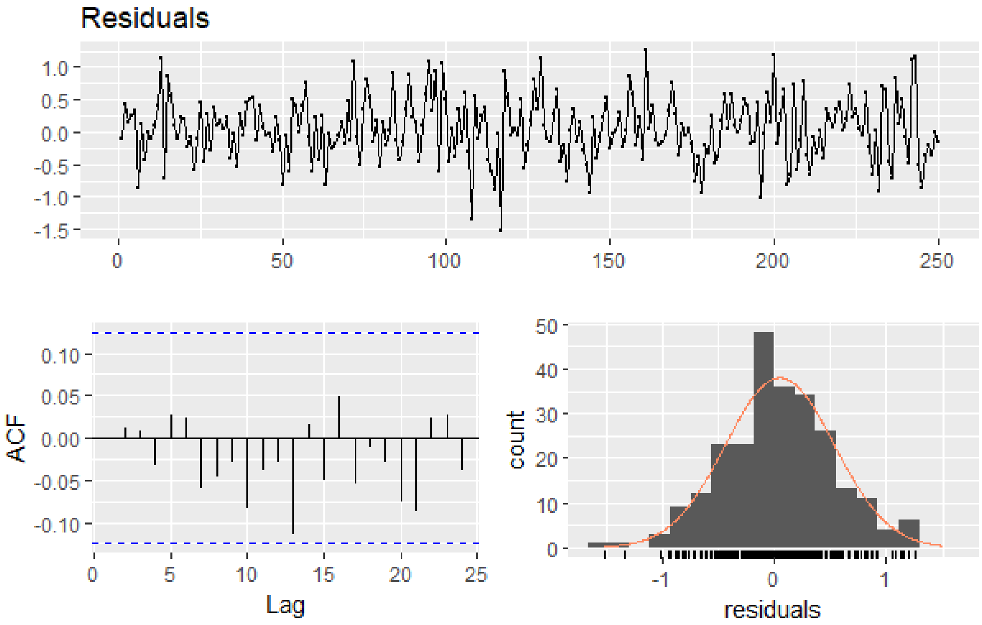
  Box-Ljung test: X-squared = 70.13, df = 20, p-value = 1.734e-07



ARMA Model Diagnosis
Residual analysis

- Model 2: $\hat{y}[t] = \varphi y[t-1] - \theta e[t-1] \Rightarrow \varphi = 0.7854 \;\; \theta = 0.8414$

  Box-Ljung test. X-squared = 12.262, df = 20, p-value = 0.9067



ARMA Model Diagnosis
Level of significance of the coefficients

- t^* statistic: $H_0 : \phi_1 = 0$

$$t^*_{N-p-q-\delta} = \frac{\hat{\phi}_1 - (\phi_1 / H_0)}{\hat{\sigma}_{\hat{\phi}_1}} = \frac{\hat{\phi}_1}{\hat{\sigma}_{\hat{\phi}_1}}$$

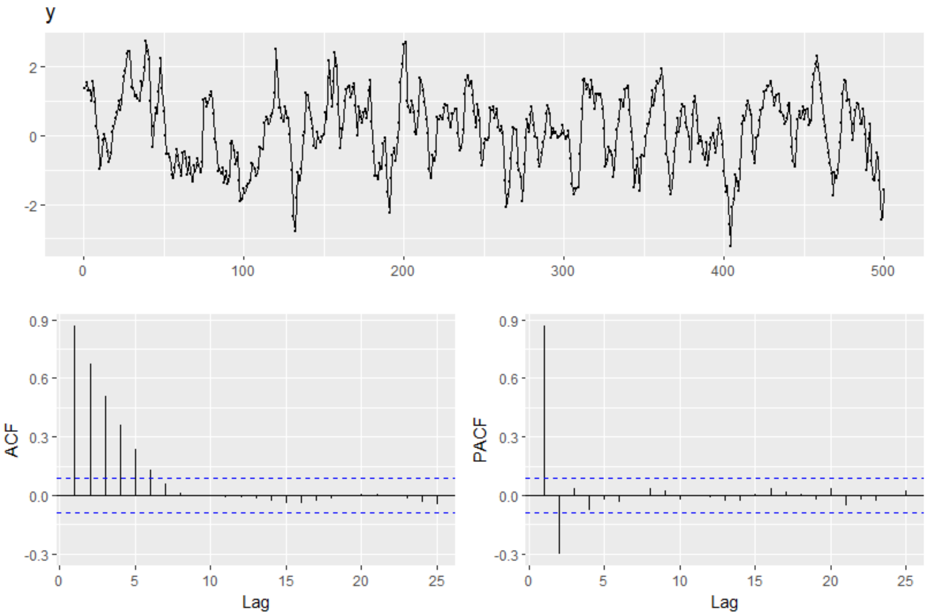
with $\delta = 1$ if a constant term is included

```
z test of coefficients:

           Estimate Std. Error z value Pr(>|z|)
ar1        0.742446   0.041857 17.7375   <2e-16 ***
ma1        0.019457   0.064988  0.2994   0.7646
intercept -0.027753   0.056188 -0.4939   0.6214
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

ARMA Model Diagnosis
Example

ARMA(1,1): AR1=0.8, MA1=0.3, s_e=0.25, N=500

```
> y <- arima.sim(n = 500, list(ar = c(0.8), ma = c(0.3)), sd = sqrt(0.25))
> ggtsdisplay(y,lag.max = 25)
```

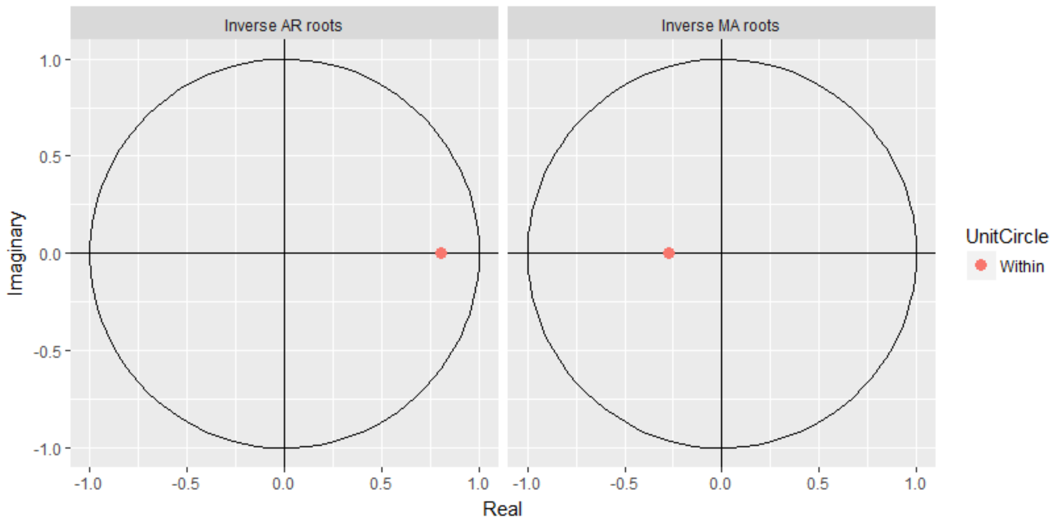


ARMA Model Diagnosis Example

```
> #Fit model
> arima.fit <- Arima(y, order=c(1,0,1), include.constant = TRUE)
> summary(arima.fit)

Series: y
ARIMA(1,0,1) with non-zero mean

Coefficients:
         ar1     ma1    mean
      0.8078  0.2710  0.2557
s.e.  0.0295  0.0482  0.1397

sigma^2 estimated as 0.2286:  log likelihood=-339.74
AIC=687.49   AICc=687.57   BIC=704.34

Training set error measures:
                       ME      RMSE      MAE      MPE     MAPE      MASE        ACF1
Training set 0.0005615644 0.4766461 0.381101 273.1589 406.4277 0.9421676 0.004156234

> coeftest(arima.fit)

z test of coefficients:

          Estimate Std. Error z value  Pr(>|z|)    
ar1       0.807751   0.029457 27.4214 < 2.2e-16 ***
ma1       0.271024   0.048205  5.6223 1.885e-08 ***
intercept 0.255653   0.139703  1.8300   0.06725 .  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

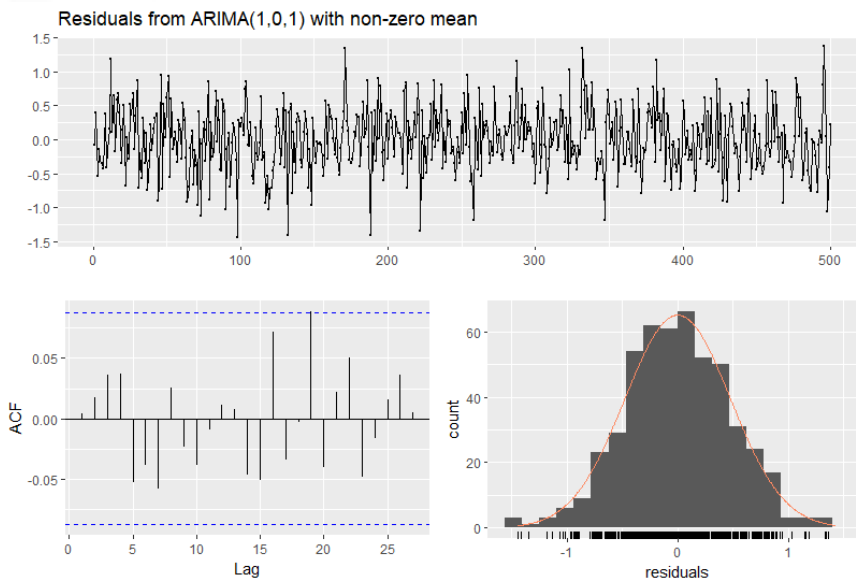
ARMA Model Diagnosis
Example

```
> autoplot(arima.fit)
```



ARMA Model Diagnosis Example

```
> checkresiduals(arima.fit)

        Ljung-Box test

data:  Residuals from ARIMA(1,0,1) with non-zero mean
Q* = 6.7621, df = 7, p-value = 0.4541

Model df: 3.   Total lags used: 10
```



ARMA Model Diagnosis
Example

```
> autoplot(y, series="Real")+ forecast::autolayer(arima.fit$fitted, series="Fitted")
```

4

ARIMA models

ARIMA models
The Box-Jenkins methodology

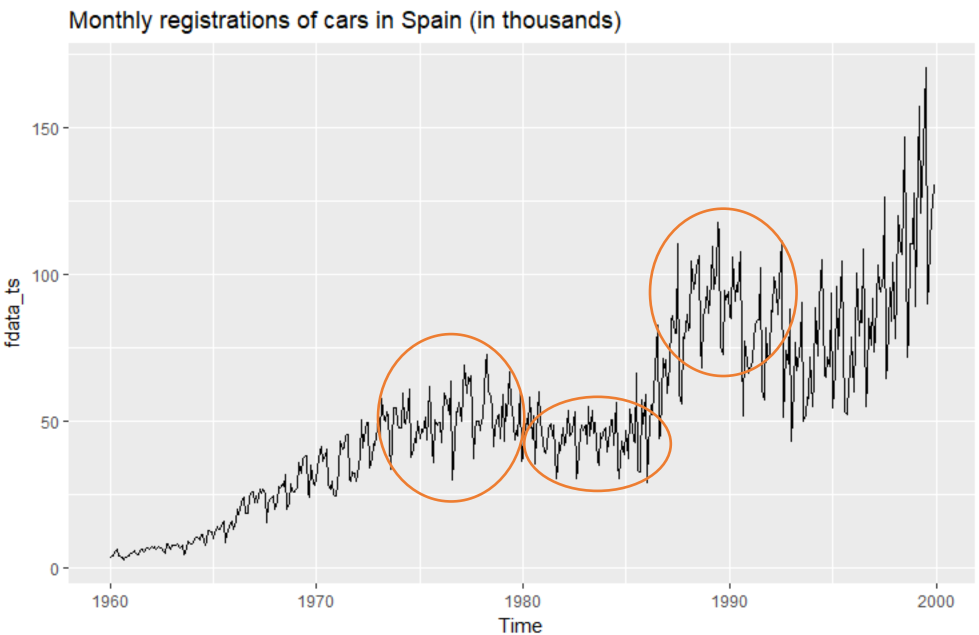
1. **Transform** the original time series in order to stabilize:
   a) The variance
   b) The mean

2. Propose and **estimate the parameters** of a tentative ARMA(p,q) model for the stationary time series.

3. **Diagnose** the resulting model

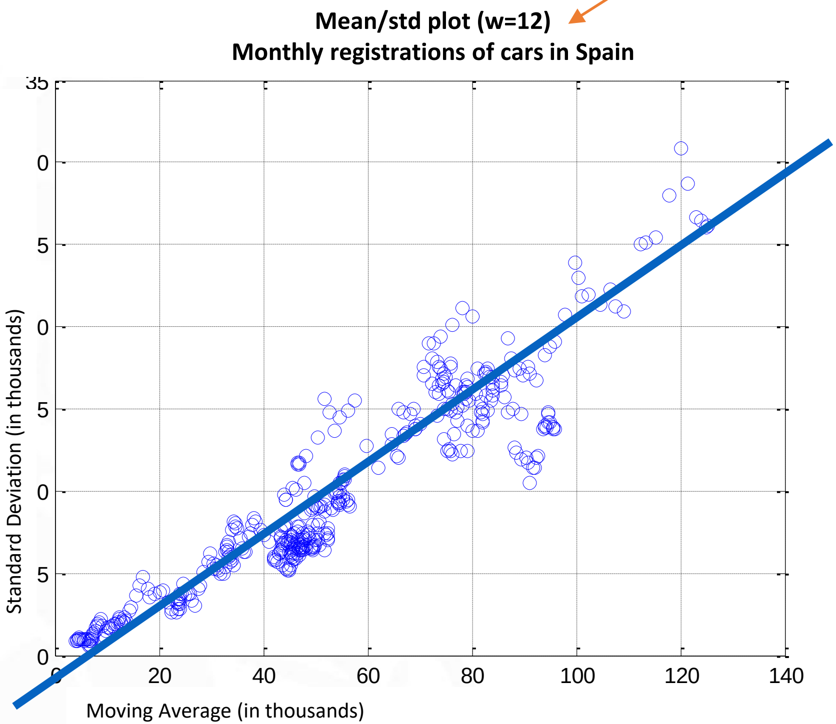
ARIMA models
Stabilizing the variance

- In many time series the volatility increases with the level:



ARIMA models
Stabilizing the variance

- The increase is often linear:



ARIMA models
Stabilizing the variance

- If the series is affected by multiplicative noise:

$$z[t] = \mu[t] \cdot u[t] \text{ with } \begin{cases} \mu[t] = f(z[t-1],...) & \text{systematic component} \\ E[u[t]] = 1 \quad \forall t & \text{innovation} \end{cases}$$

- Then:

$$E[z[t]] = \mu[t]$$

$$\sigma_{z[t]} = \left(E\left[(z[t]-\mu[t])^2\right]\right)^{1/2} = \left(E\left[(\mu[t]\cdot u[t]-\mu[t])^2\right]\right)^{1/2}$$

$$= |\mu[t]| \cdot \left(E\left[(u[t]-1)^2\right]\right)^{1/2} = \boxed{|\mu[t]| \cdot \sigma_u}$$

ARIMA models
Stabilizing the variance

- With de log transformation:

$$\begin{aligned} y[t] &= ln(z[t]) \\ &= ln(\mu[t]) + ln(u[t]) \\ &= ln(\mu[t]) + \varepsilon[t] \end{aligned}$$

- Other transformations (Box-Cox):

$$y[t] = \frac{z[t]^{\lambda} - 1}{\lambda} \quad \xrightarrow[\lambda \to 0]{} \quad ln(z[t])$$

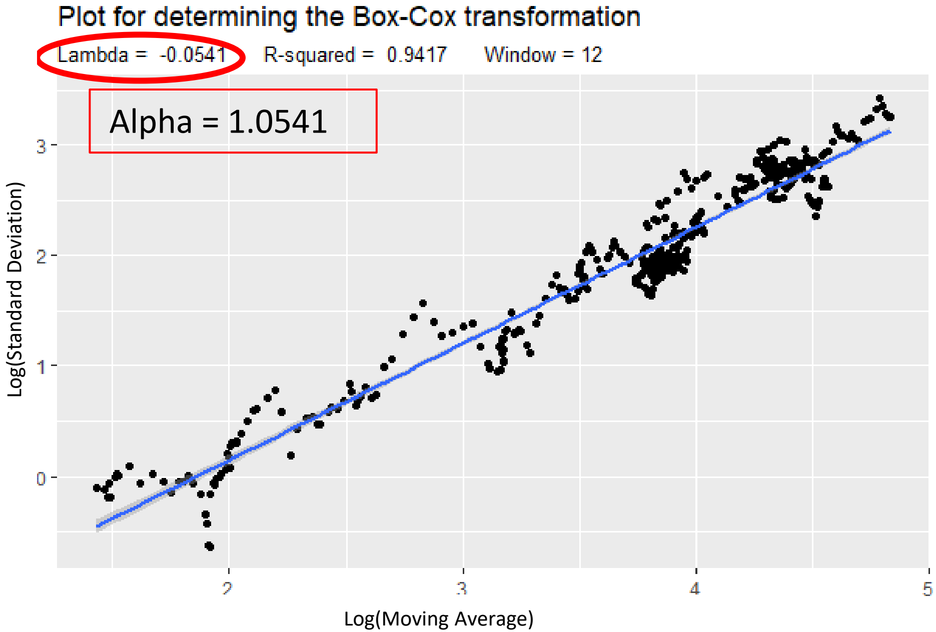
$$\boxed{\sigma_{z[t]} = |\mu[t]|^{\alpha} \cdot k \quad \Rightarrow \quad ln\left(\hat{\sigma}_{z_i}\right) = c + \alpha \cdot ln\left(\bar{z}_i\right)}$$

where $\alpha = 1 - \lambda$

ARIMA models
Stabilizing the variance

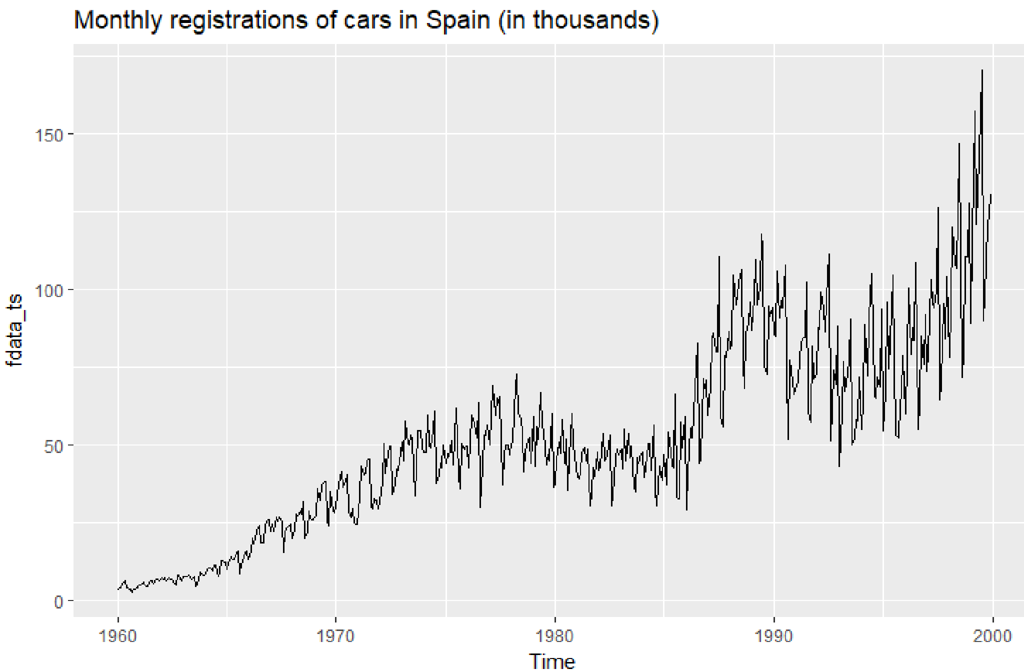
- If we take logarithms and fit a regression line:

```
> source("ArimaTF.R")
> BoxCox.lambda.plot(y, window.width = 12)
```



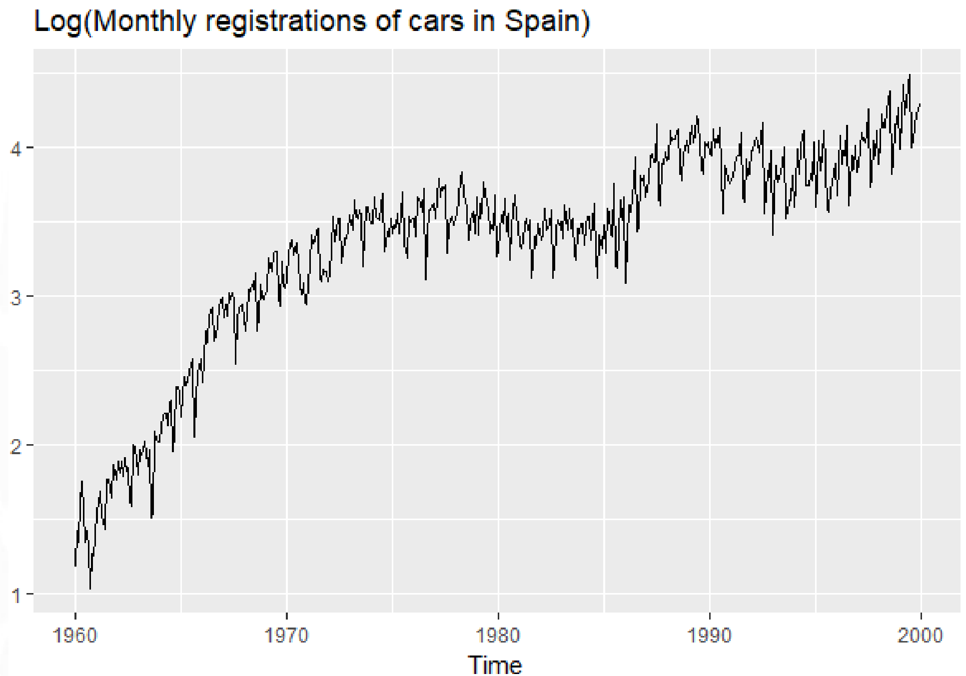
ARIMA models
Stabilizing the variance

- Original time series:



ARIMA models
Stabilizing the variance

- Logarithmic transformation:



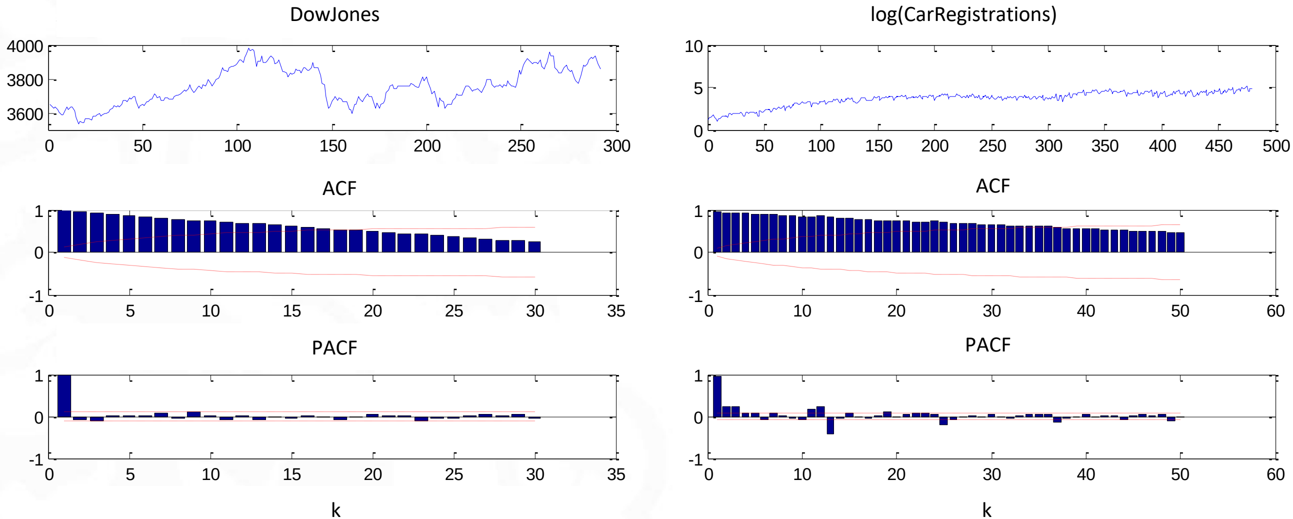
comillas.edu

ARIMA models
Integrated processes

- A process may be non-stationary in the mean, the variance, the autocorrelations or other characteristics of the distribution of the variables.
- When the level of the series is not stable over time and can for example have a tendency, the series is non-stationary on the mean.
- Integrated processes are non-stationary processes which become stationary when differenced.
- The ACF of an integrated process shows a slow linear decreasing pattern (the decrease is exponential for an ARMA process).
- Most real time series are not stationary and their average level changes with time.

ARIMA models
Stabilizing the mean

- The stationarity of the mean requires that the series keeps oscillating around a constant level. When this does not happen, the ACF has a **very slow, linear decrease**:



ARIMA models
Regular differencing

- To make the series stationary, we use differencing:

$$\nabla y[t] = (1 - B)y[t] = y'[t] = y[t] - y[t-1]$$

$$\begin{aligned}\nabla^2 y[t] = (1 - B)^2 y[t] = y''[t] &= y'[t] - y'[t-1] \\ &= (y[t] - y[t-1]) - (y[t-1] - y[t-2]) \\ &= y[t] - 2y[t-1] + y[t-2]\end{aligned}$$

ARIMA models
Regular differencing

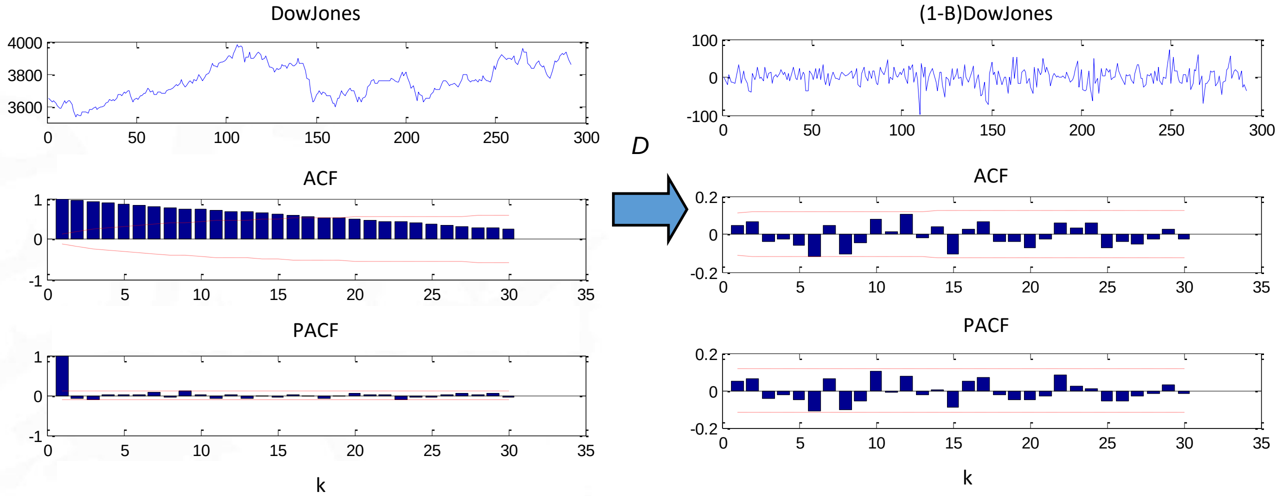
- First order differencing removes linear trends:

$$y(t) = a + bt + z(t) \qquad \text{with } z(t) \text{ stationary}$$

$$\nabla y(t) = y(t) - y(t-1) = a + bt + z(t) - (a + b(t-1) + z(t-1))$$
$$= b + z(t) - z(t-1) = \boxed{b + \alpha(t)}$$

- Second order differencing removes quadratic trends

ARIMA models
Regular differencing



$$\nabla y[t] = y[t] - y[t-1] = \varepsilon[t] \Leftrightarrow y[t] = y[t-1] + \varepsilon[t]$$

$$\equiv \text{random walk}$$

ARIMA models
Returns

- The differencing of a previously log-transformed series gives:

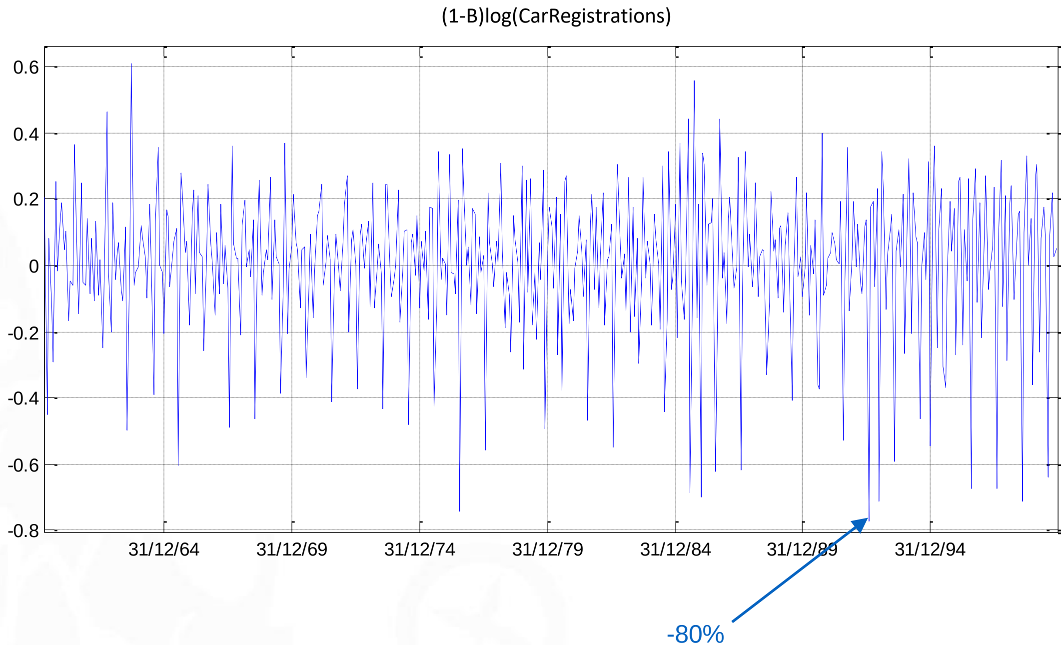
$$(1-B)\ln(z[t]) = \ln(z[t]) - \ln(z[t-1]) = \ln\left(\frac{z[t]}{z[t-1]}\right)$$

$$= \ln\left(1 + \frac{z[t]-z[t-1]}{z[t-1]}\right)$$

$$\approx \boxed{\frac{z[t]-z[t-1]}{z[t-1]}}$$

which is known as **return** (measure of the relative growth)

ARIMA models
Returns



ARIMA models
ARIMA(p,d,q)

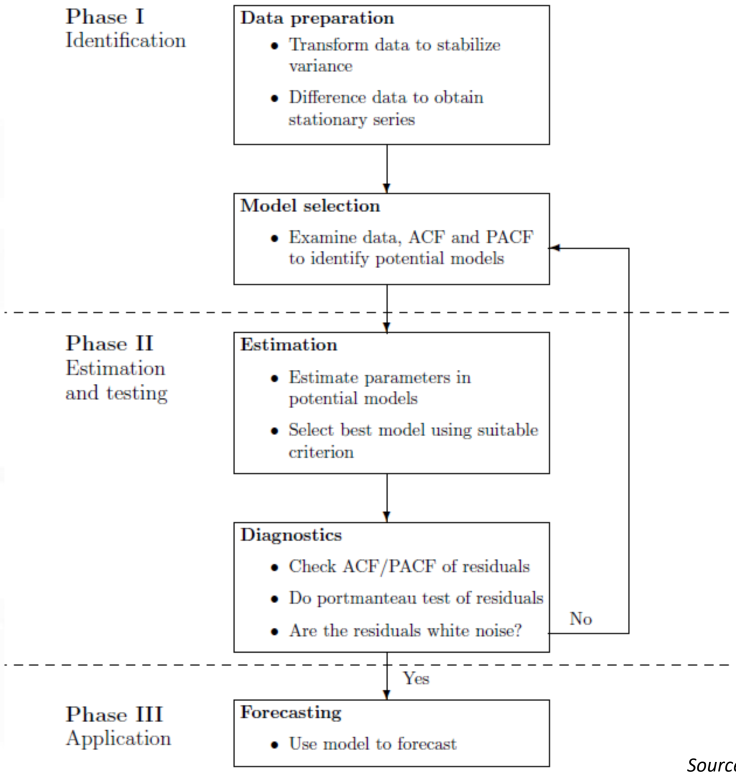
- The ARIMA(*p,d,q*) model (AutoRegressive, Integrated, Moving Average) results from the application of an ARMA model to a d^{th} differenced time series:

  - AR: *p* = autoregressive order
  - I: *d* = regular differencing order
  - MA: *q* = moving average order

- Example: ARIMA(*3,2,1*)

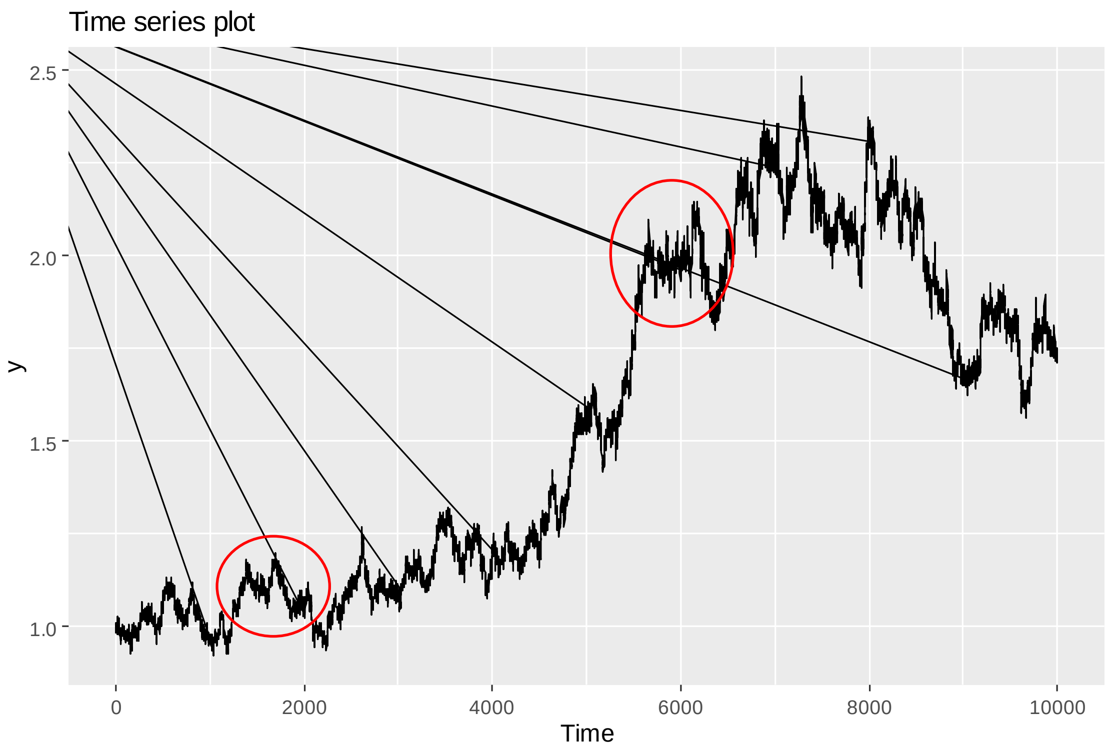
$$\underbrace{(1-\phi_1 B-\phi_2 B^2-\phi_3 B^3)}_{\text{AR(3)}}\underbrace{(1-B)^2}_{\text{Regular differencing(2)}} y[t]=\underbrace{(1-\theta_1 B)}_{\text{MA(1)}}\varepsilon[t]$$

ARIMA models
The Box-Jenkins methodology

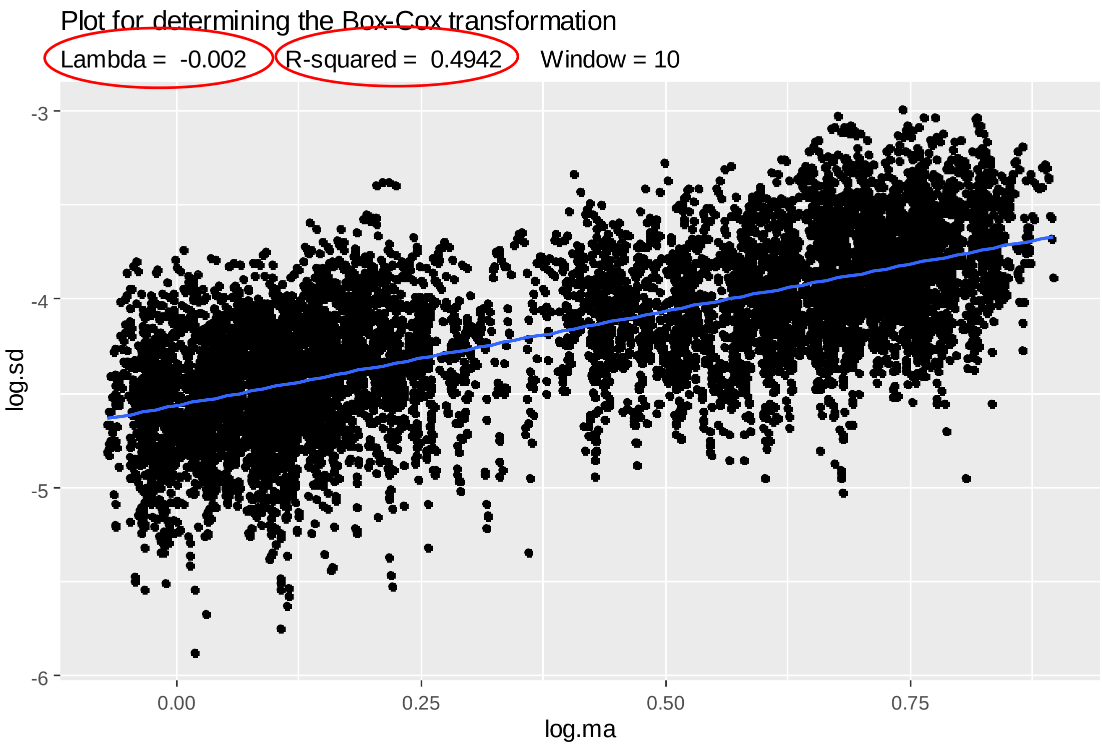


Source: Makridakis et al. (1998)

ARIMA models
Example:

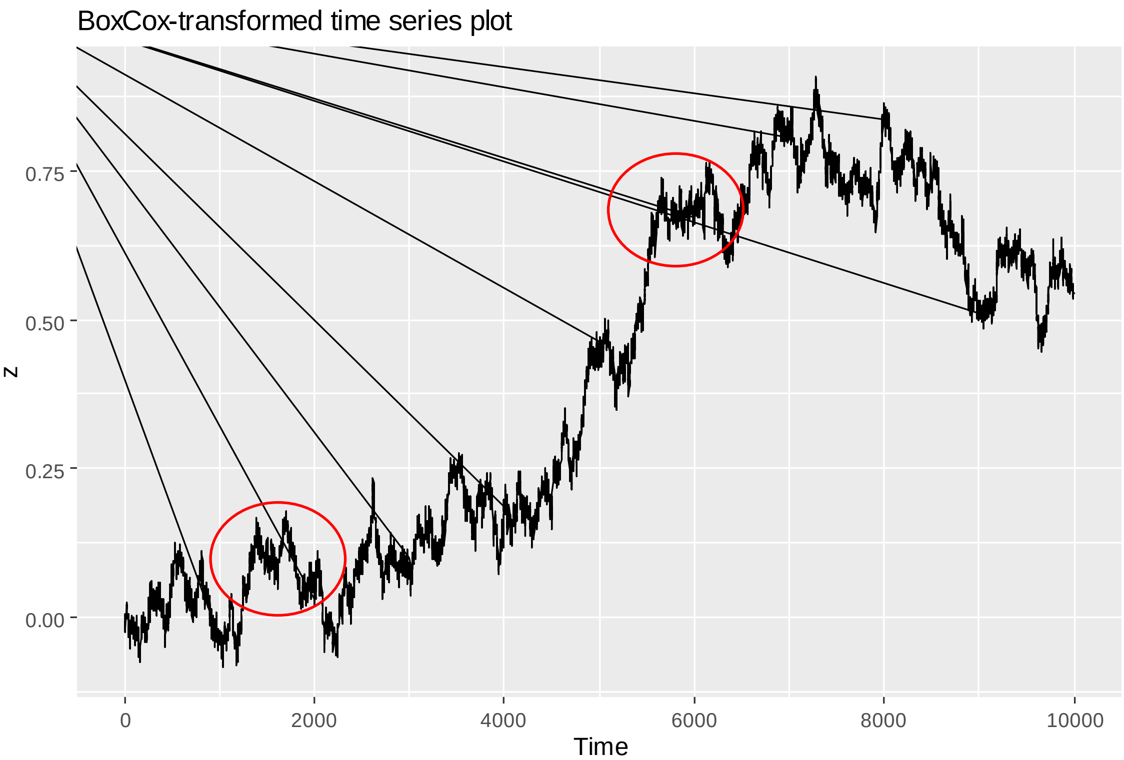


ARIMA models
Data preparation: variance stabilization



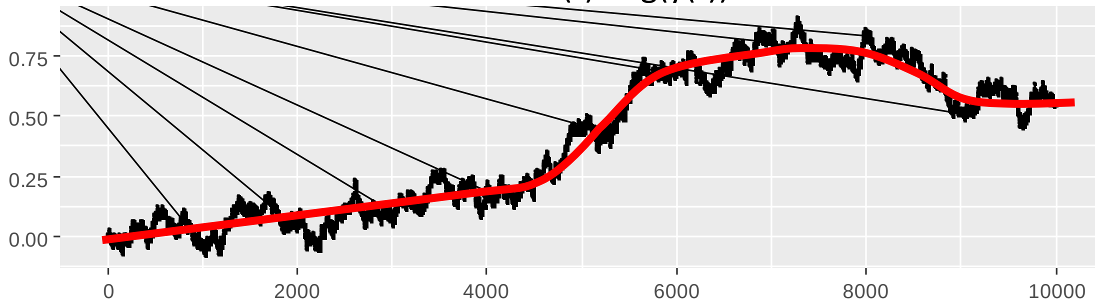
ARIMA models

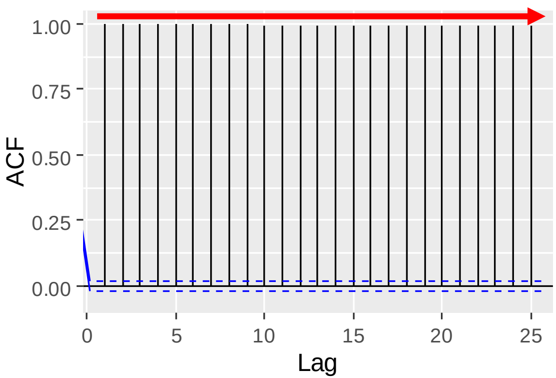
Data preparation: variance stabilization

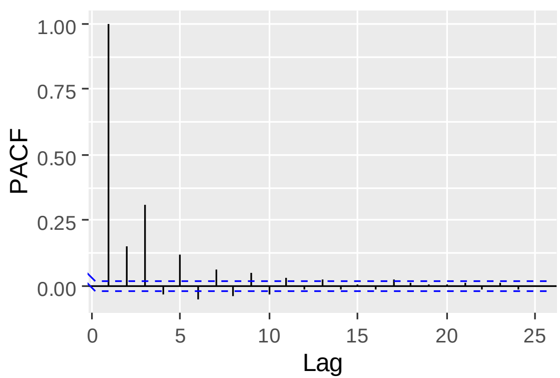


ARIMA models
Data preparation: mean stabilization

BoxCox-transformed time series: z(t)=log(y(t))

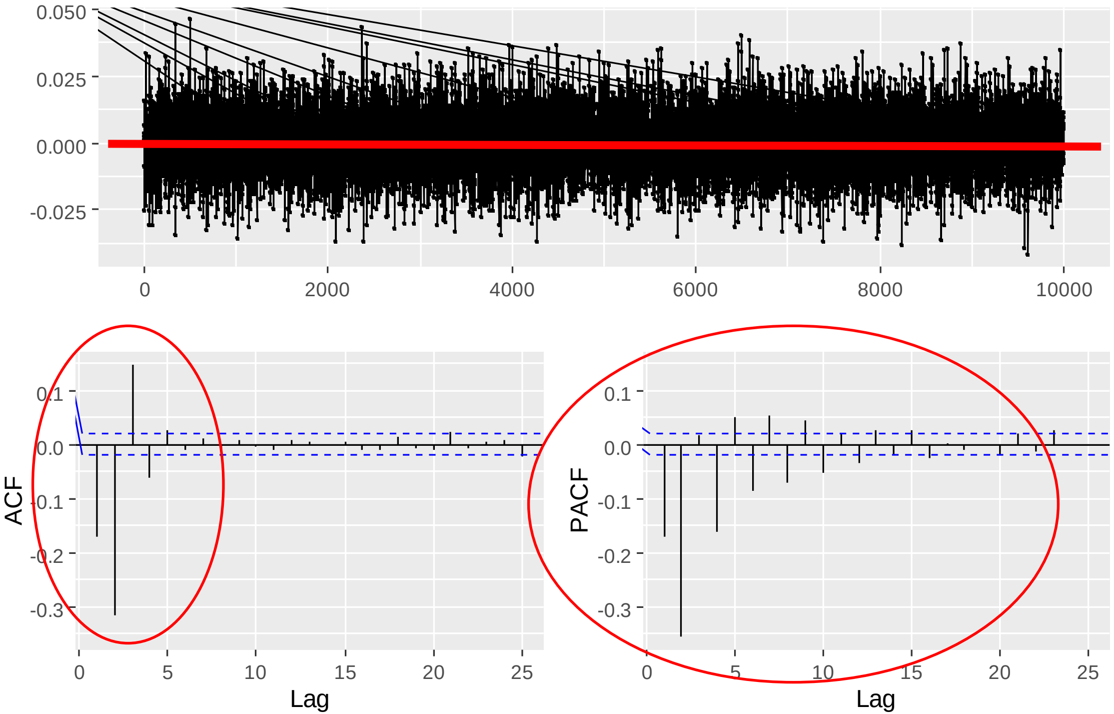






ARIMA models
Data preparation & Model selection

Differenced BoxCox-transformed time series: (1-B)z(t)=log(y(t))-log(y(t-1))



ARIMA models
Model selection & Estimation

ARIMA(2,1,2) with constant term:

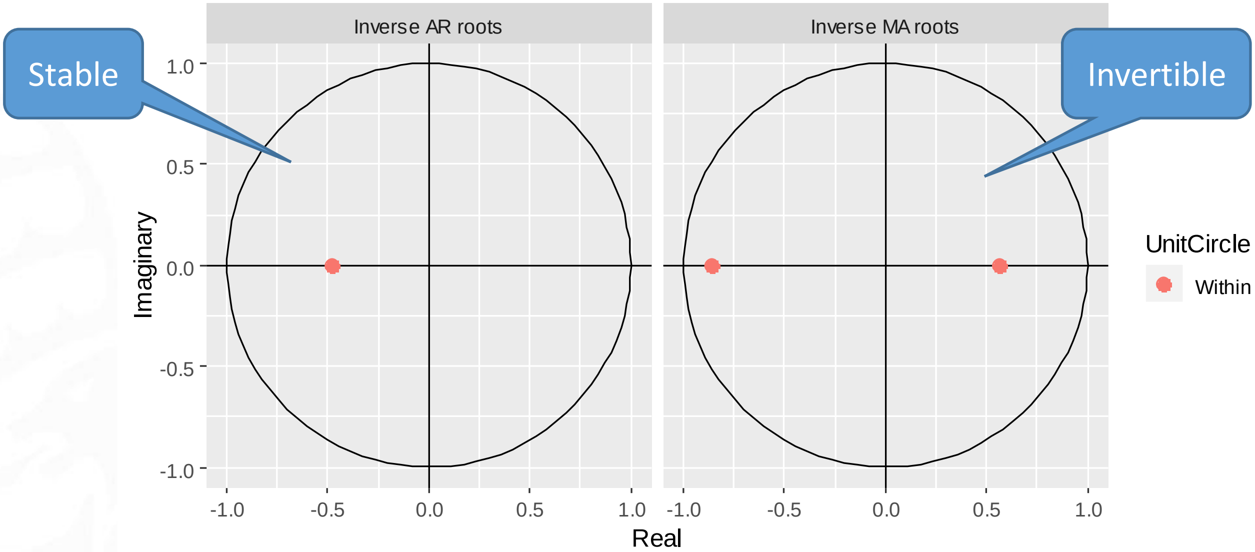
```
arima.fit <- Arima(y, order=c(2,1,2), lambda = Lambda,
                   include.constant = TRUE)

z test of coefficients:

         Estimate  Std. Error  z value Pr(>|z|)
ar1   -4.7876e-01  2.0631e-02 -23.2054   <2e-16 ***
ar2    6.3269e-03  2.0426e-02   0.3098   0.7567
ma1    2.9000e-01  1.7985e-02  16.1247   <2e-16 ***
ma2   -4.9532e-01  1.7962e-02 -27.5752   <2e-16 ***
drift  5.5754e-05  5.7606e-05   0.9678   0.3331
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

ARIMA models
Model selection & Estimation

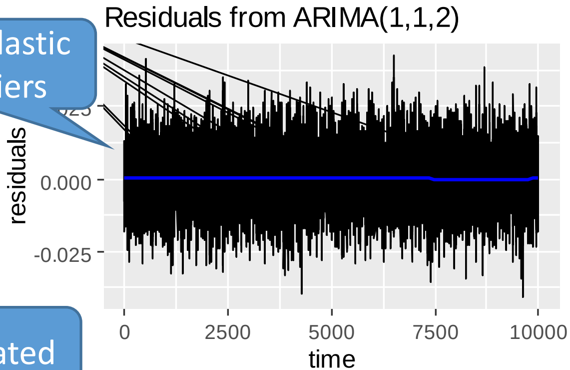
ARIMA(1,1,2)

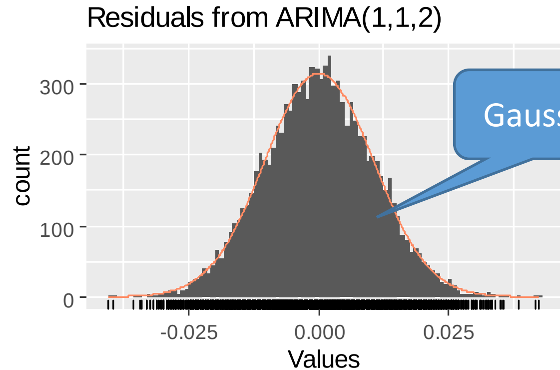
```
arima.fit <- Arima(y, order=c(1,1,2), lambda = Lambda,
                   include.constant = FALSE)

z test of coefficients:

      Estimate Std. Error z value  Pr(>|z|)
ar1 -0.4821988  0.0164979 -29.228 < 2.2e-16 ***
ma1  0.2928914  0.0150180  19.503 < 2.2e-16 ***
ma2 -0.4903797  0.0088769 -55.242 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

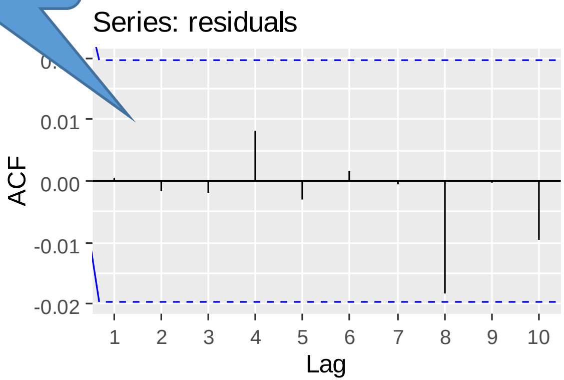
ARIMA models
Diagnostics: Root analysis



ARIMA models
Diagnostics: Analysis of the residuals

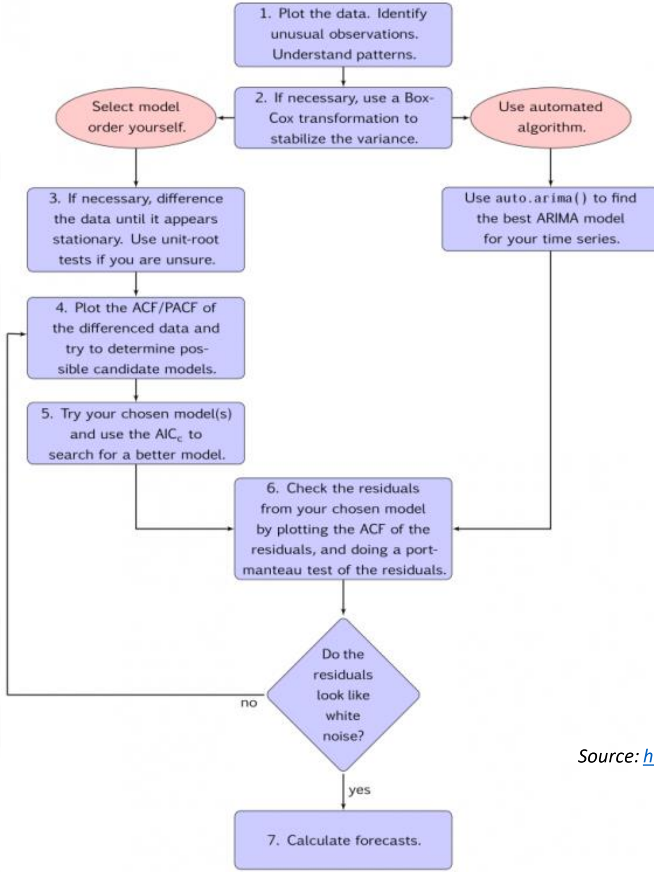







```
Ljung-Box testdata:

Residuals from ARIMA(1,1,2)

Q* = 5.1027, df = 7, p-value = 0.6474


Model df: 3.   Total lags used: 10
```

ARIMA models
The Box-Jenkins methodology



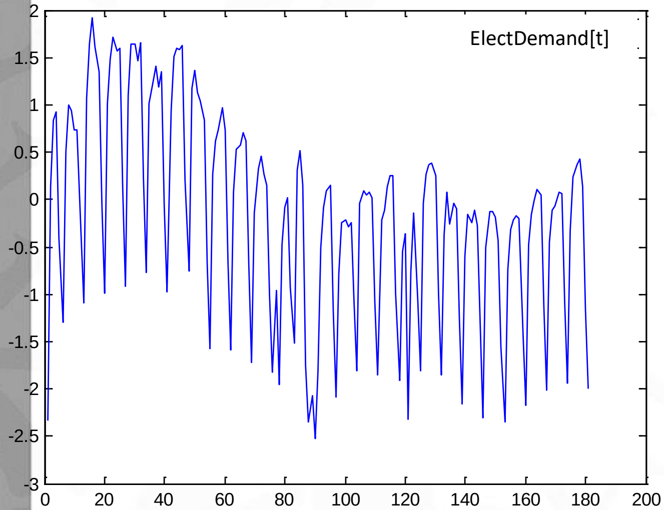
Source: https://otexts.com/fpp2/arima-r.html

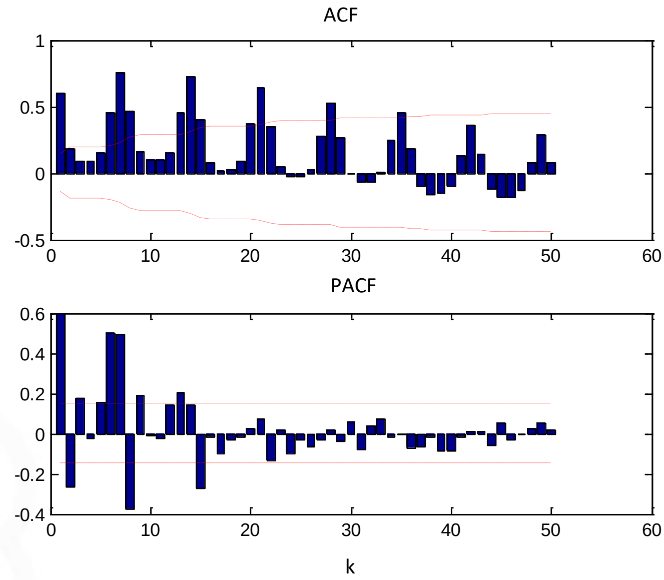
5

Seasonal ARIMA models

Seasonal ARIMA models
Seasonality

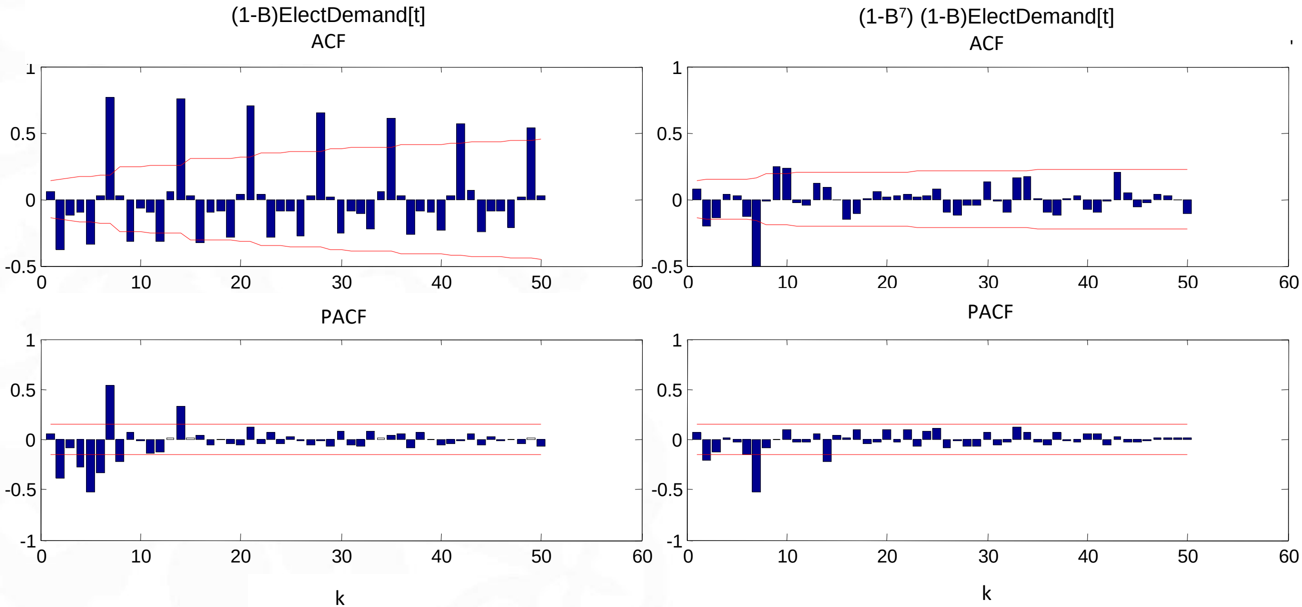
- Seasonality of period *s* is evidenced in the ACF and PACF when significative coefficients appear in the multiples of the period *s*





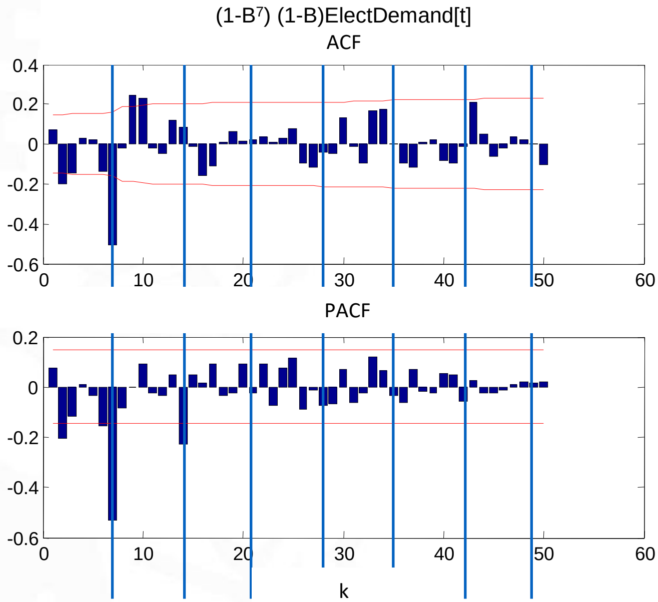
Seasonal ARIMA models
Seasonality

- Seasonal non-stationay time series may require seasonal diferencing: $$(1-B^s)y[t] = y[t] - y[t-s]$$



Seasonal ARIMA models
Seasonality

- The seasonal (multiples of *s*) coefficients of the ACF and PACF are used for the identification of the seasonal ARIMA model:



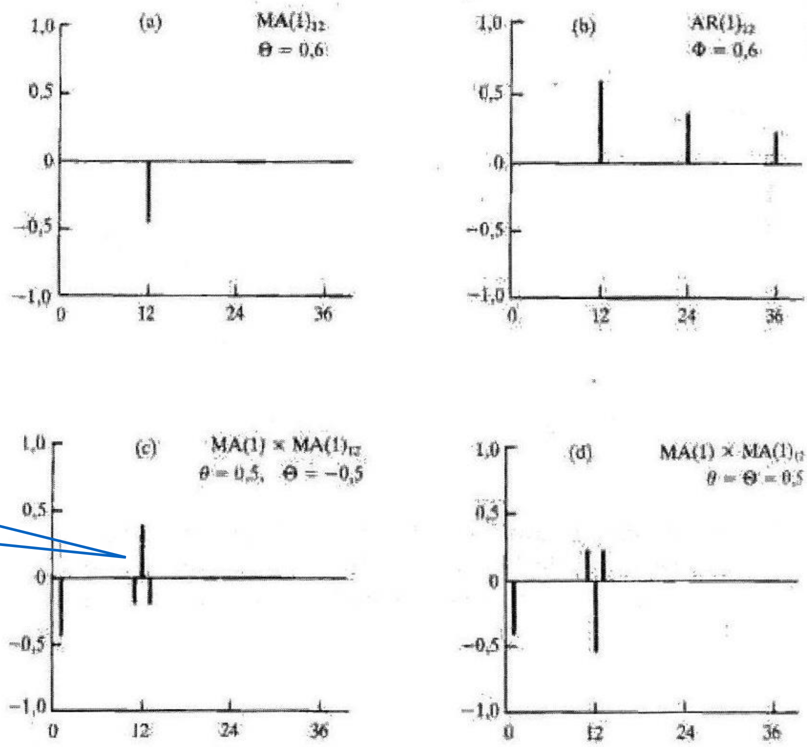
Seasonal processes ACF

a) The first (1 to 6) coefficients of the ACF are only affected by the regular component .

b) The seasonal coefficients are basically affected by the seasonal component.

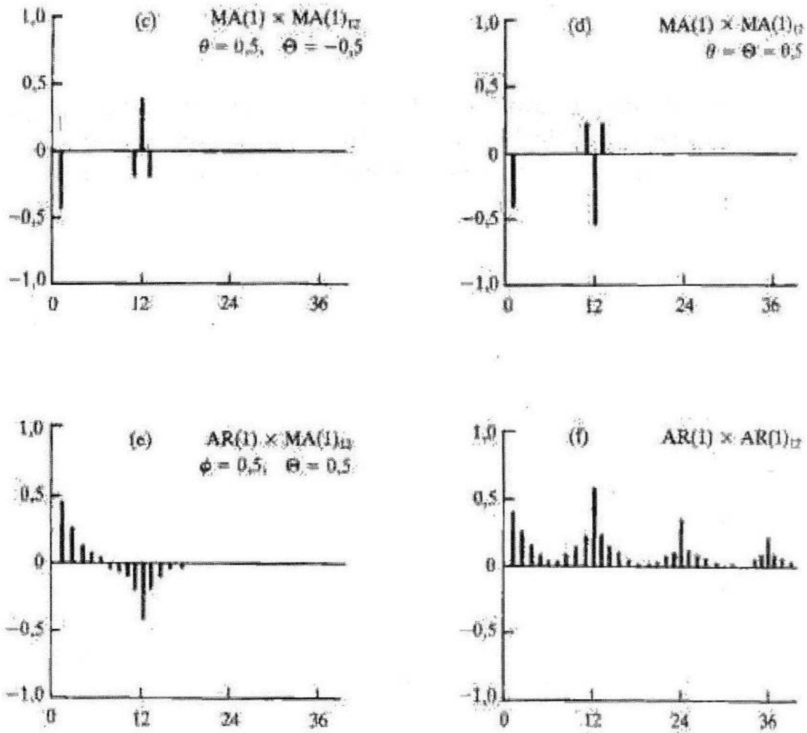
c) The ACF of the regular component is replicated at both sides of the seasonal lags.

Seasonal ARIMA models
Seasonal processes ACF



The regular pattern is replicated at both sides of the seasonal lags

Seasonal ARIMA models
Seasonal processes ACF



Seasonal processes PACF

a) The first (1 to 6) coefficients of the PACF are only affected by the regular component .

b) The seasonal coefficients are basically affected by the seasonal component.

c) The PACF of the regular component is replicated at the right side of the seasonal lags.

d) The ACF of the regular component is replicated at the left side of the seasonal lags of the PACF.

Seasonal ARIMA models
ARIMA(p,d,q)(P,D,Q)$_s$

- The ARIMA(p,d,q)(P,D,Q)$_s$ model (seasonal autoregressive, integrated, moving average) is the combination of a seasonal ARIMA model and a regular ARIMA model:
  - AR: p = autoregressive order
  - I: d = regular differencing order
  - MA: q = moving average order

  - AR$_s$: P = seasonal autoregressive order
  - I$_s$: D = seasonal differencing order
  - MA$_s$: Q = seasonal moving average order

Seasonal ARIMA models
ARIMA(p,d,q)(P,D,Q)$_s$

- Example: ARIMA($1,1,1$)($1,1,1$)$_4$

$$(1-\phi_1 B)(1-\Phi_1 B^4)(1-B)(1-B^4)y[t] \\ = (1-\theta_1 B)(1-\Theta_1 B^4)\varepsilon[t]$$

which leads to:

$$\begin{aligned} y(t) = {} & (1+\phi_1)y(t-1) - \phi_1 y(t-2) + (1+\Phi_1)y(t-4) \\ & -(1+\phi_1+\Phi_1+\phi_1\Phi_1)y(t-5) + (\phi_1+\Phi_1\phi_1)y(t-6) \\ & -\Phi_1 y(t-8) + (\Phi_1+\phi_1\Phi_1)y(t-9) - \phi_1\Phi_1 y(t-10) \\ & +\varepsilon(t) - \theta_1\varepsilon(t-1) - \Theta_1\varepsilon(t-4) + \theta_1\Theta_1\varepsilon(t-5) \end{aligned}$$

this equation is used for forecasting.

Seasonal ARIMA models
ARIMA$(p,d,q)(P,D,Q)_s$

- Identification:
  1) Plot the series and search for possible outliers.

  2) Stabilize the variance by transforming the data. Use the mean/std plot.

  3) Analyse the stationarity of the transformed series. If the data has a constant level and its ACF and PACF cancel rapidly, then it can be considered as stationary.

  4) If the series is not stationary, then we use differencing. For non-seasonal time series, apply regular differencing. For seasonal time series, we first apply seasonal differencing and once the seasonal autocorrelations have been stabilized, apply regular differencing ($d,D \leq 2$).

Seasonal ARIMA models
ARIMA$(p,d,q)(P,D,Q)_s$

5) Identify the seasonal model by analyzing the seasonal coefficients of the ACF and PACF.

6) Once the seasonal model has been identified, identify the regular component by exploring the ACF and PACF of the residuals of the seasonal model.

7) Check the significance of the coefficients

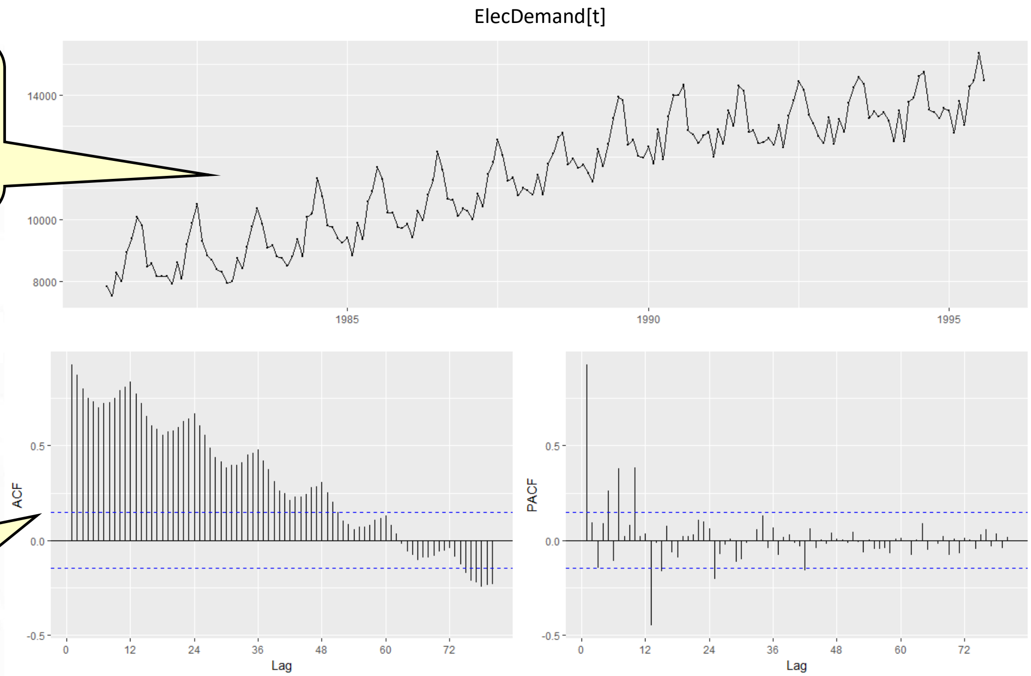
8) Analyze the residuals:
- *Outlier* detection
- Test for serial correlation (Ljung y Box test)
- Plot the histogram of the residuals (Normality test)

9) Compare different models using AIC or SBC (*M=p+q+P+Q):*

$$AIC \approx N(1+\log(2\pi)) + N\log(\hat{\sigma}_\varepsilon^2) + 2M$$

Seasonal ARIMA models
Example

```
> #Plots a time series along with its acf and either its pacf
> ggtsdisplay(y, lag.max = 80)
```

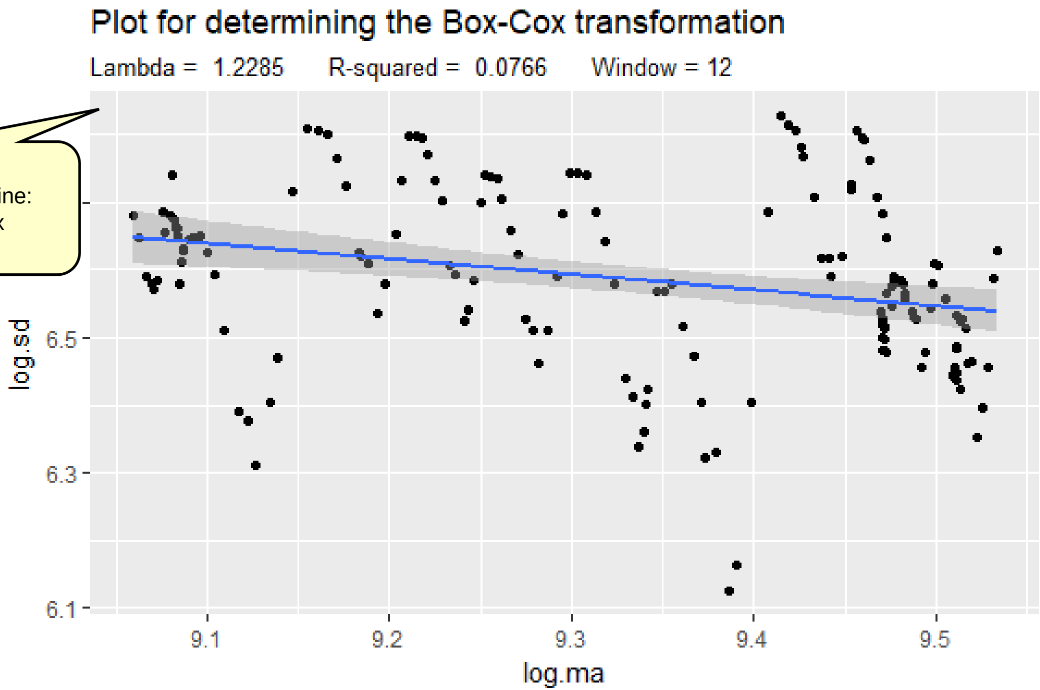


STEP 1-2:
-The series is homoscedastic (no transformation is required)
- No outliers are detected

STED 3:
-Seasonal differencing is required

Seasonal ARIMA models
Example

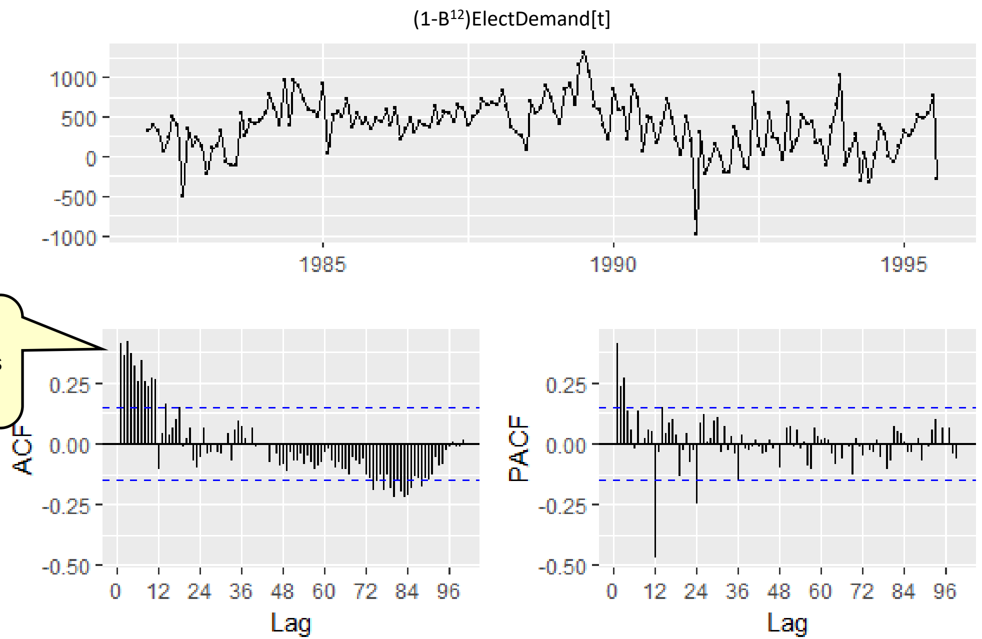
```
#Plot mean/sd scatterplot
> source("ArimaTF.R")
> BoxCox.lambda.plot(y, window.width = 12)
```



STEP 2:
-Low R-squared and flat line: no need to apply Box-Cox transformation.

Seasonal ARIMA models
Example

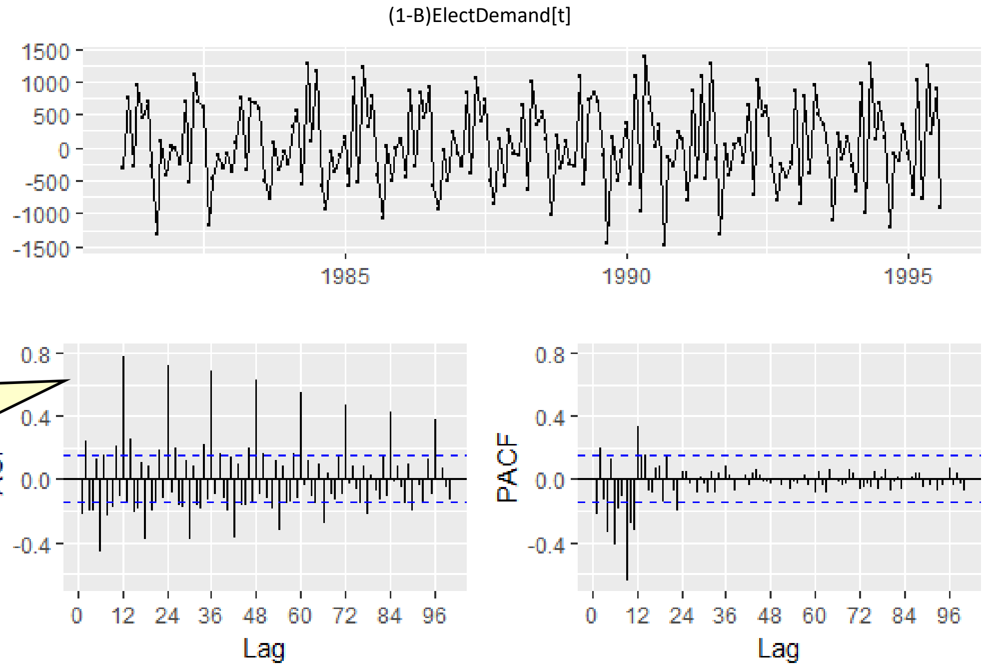
```
> #Seasonal differencing and plot ACF and PACF
> y.sdiff <- diff(y, lag = 12, differences = 1)
> #differences contains the order of differentiation
> ggtsdisplay(y.sdiff, lag.max = 100)
```



STEP 4:
-The series also requires regular differencing

Seasonal ARIMA models
Example

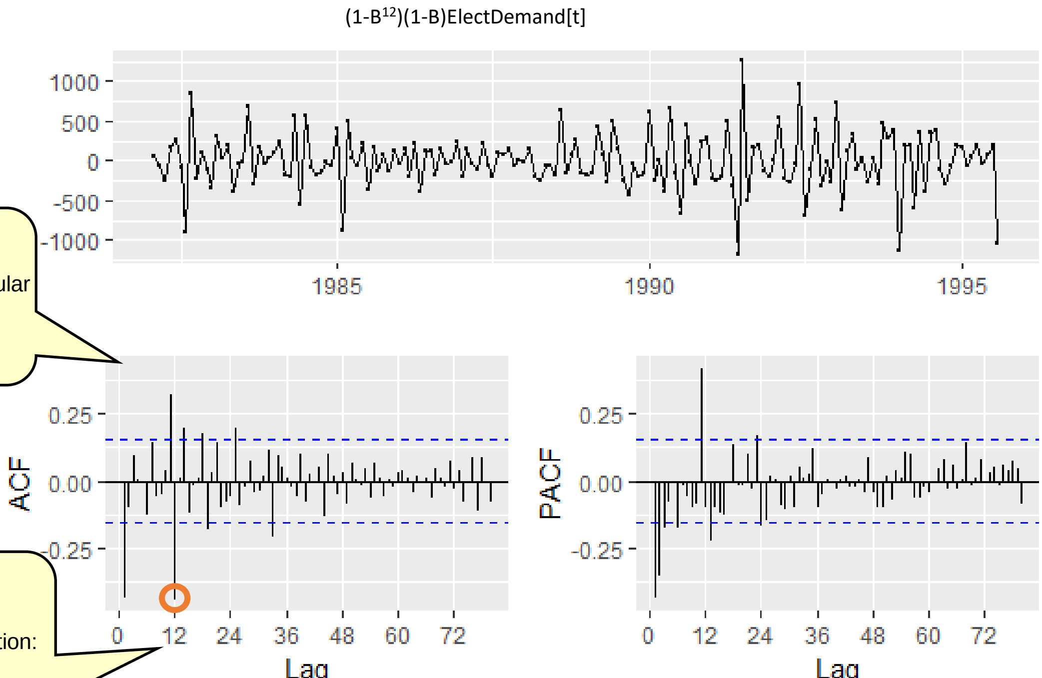
```
> #Regular differencing and plot ACF and PACF
> y.rdiff <- diff(y, lag = 1, differences = 1)
> ggtsdisplay(y.rdiff, lag.max = 100)
```



STEP 4':
- Regular differentiation reveals the need for seasonal differencing

Seasonal ARIMA models
Example

```
> #Regular and seasonal differencing and plot ACF and PACF
> y.rdiff.sdiff <- diff(y.rdiff, lag = 12, differences = 1)
> ggtsdisplay(y.rdiff.sdiff, lag.max = 80)
```



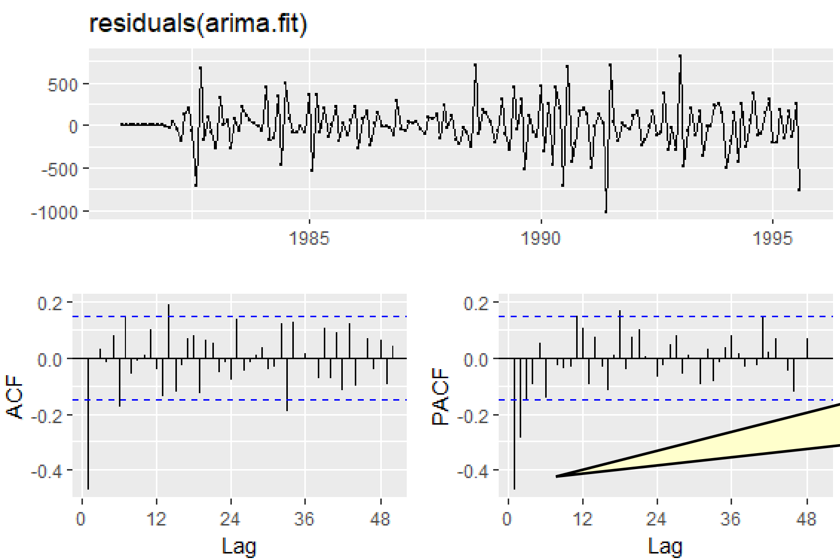
STEP 4'':
-After seasonal and regular differencing the series is stationary

STEP 5:
Seasonal model identification: ARIMA(0,1,0)(0,1,1)$_{12}$

Seasonal ARIMA models
Example

- Seasonal model: ARIMA(0,1,0)(0,1,1)$_{12}$:

```
> #Fit model with estimated order
> arima.fit <- Arima(y, order=c(0,1,0),
                     seasonal = list(order=c(0,1,1),period=12),
                     lambda = NULL,
                     include.constant = TRUE)
> ggtsdisplay(residuals(arima.fit), lag.max = 50)
```



STEP 6:
Regular component identification:
ARIMA(0,1,1)(0,1,1)$_{12}$

ARMA Model Diagnosis Example

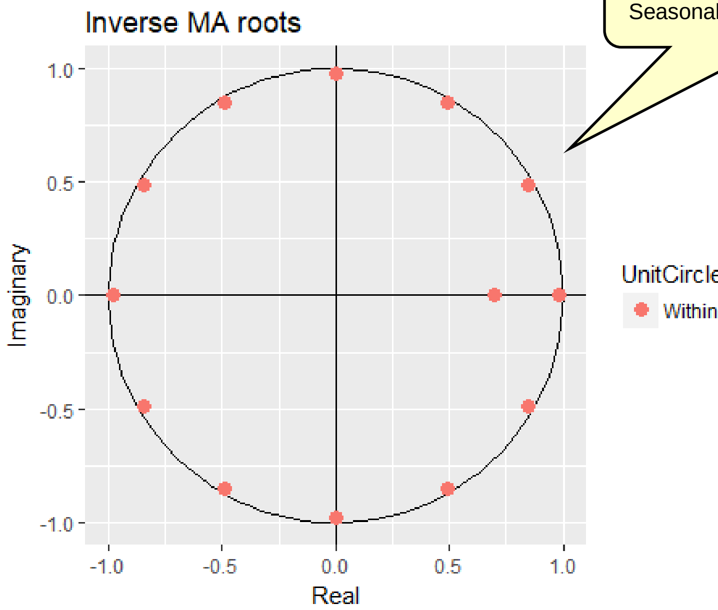
- ARIMA(0,1,1)(0,1,1)$_{12}$

```
> arima.fit <- Arima(y, order=c(0,1,1),
                     seasonal = list(order=c(0,1,1),period=12),
                     lambda = NULL, include.constant = TRUE)
> summary(arima.fit)

RIMA(0,1,1)(0,1,1)[12]

Coefficients:
          ma1     sma1
      -0.6946  -0.7676
s.e.   0.0526   0.0634

sigma^2 estimated as 50720:  log likelihood=-1118.92
AIC=2243.85   AICc=2244   BIC=2253.13

Training set error measures:
                    ME     RMSE      MAE          MPE     MAPE      MASE        ACF1
Training set -2.815167 215.4013 157.2278 -0.007999796 1.349636 0.3648184 -0.04137056
> coeftest(arima.fit)

z test of coefficients:

      Estimate Std. Error z value  Pr(>|z|)
ma1  -0.694557   0.052616 -13.201 < 2.2e-16 ***
sma1 -0.767618   0.063388 -12.110 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

STEP 7:
Level of significance of the coefficients

ARMA Model Diagnosis
Example

```
> autoplot(arima.fit)
```

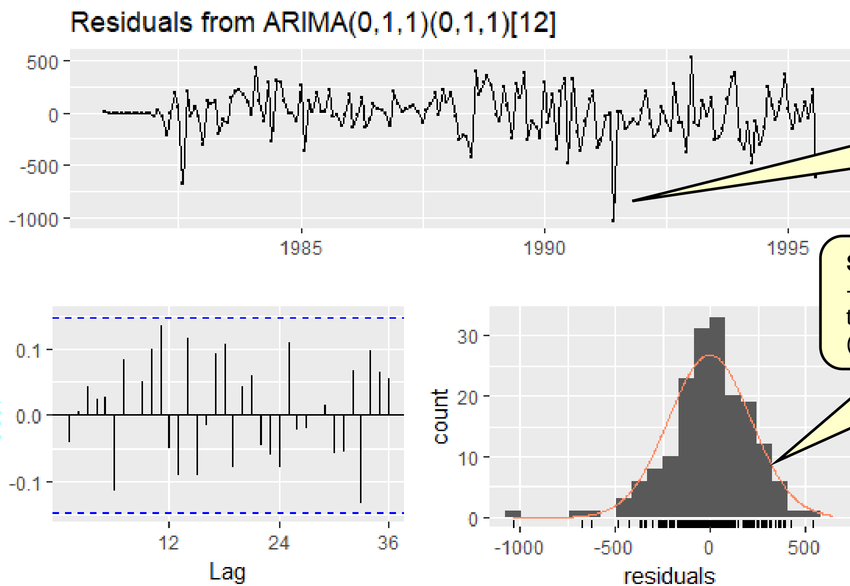


STEP 8':
Seasonal terms have complex roots

ARMA Model Diagnosis Example

```
> checkresiduals(arima.fit)

        Ljung-Box test

data:  Residuals from ARIMA(0,1,1)(0,1,1)[12]
Q* = 25.49, df = 22, p-value = 0.2742

Model df: 2.   Total lags used: 24
```

STEP 8':
Analysis of the residuals: they are uncorrelated



STEP 8:
- Possible *outlier*

STEP 8''':
-Plot the histogram of the residuals: the distribution is asymmetric (non-Normal)

Seasonal ARIMA models
Example

- Forecasting with the ARIMA(0,1,1)(0,1,1)$_{12}$:

$$(1-B)(1-B^{12})y[t] = (1-\theta_1 B)(1-\Theta_1 B^{12})\varepsilon[t]$$

which leads to:

$$y(t) = y(t-1) + y(t-12) - y(t-13) + \varepsilon(t) - \theta_1\varepsilon(t-1) - \Theta_1\varepsilon(t-12) + \theta_1\Theta_1\varepsilon(t-13)$$

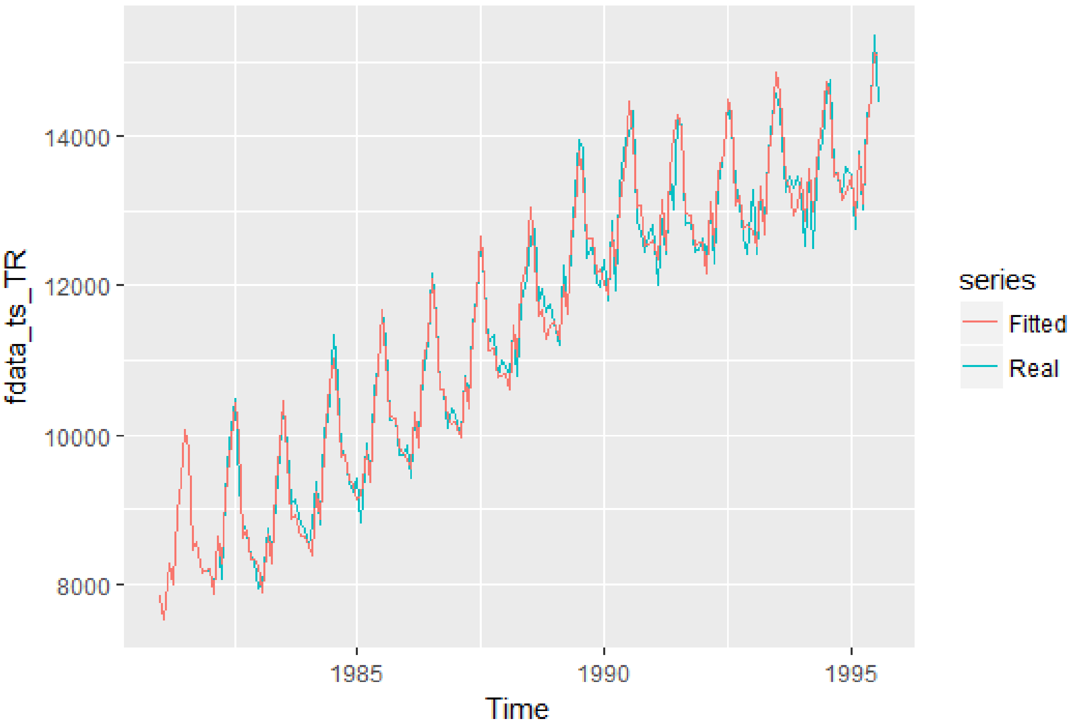
- To use this model as a predictor, it will be necessary to estimate the past values of the noise from the errors of the model:

$$\hat{y}(t+1) = y(t) + y(t-11) - y(t-12) - \theta_1 e(t) - \Theta_1 e(t-11) + \theta_1\Theta_1 e(t-12)$$

- As you toggle the prediction horizon, several lags of the error may become unavailable (they are set to 0), as well as some delays of the output (they are substituted by predictions).

ARMA Model Diagnosis Example

```
> autoplot(y, series="Real")+ forecast::autolayer(fitted(arima.fit), series="Fitted")
```



6

Dynamic Regression models

Multiple Regression

- Multiple regression model:

$$y[t] = \alpha_1 x_1[t] + \alpha_2 x_2[t] + \ldots + \alpha_n x_n[t] + \varepsilon[t]$$

where:

  - *y[t]:* output or dependent variable
  - $x_i[t]$: input, explanatory or independent variables
  - $\varepsilon[t]$: noise

- Basic hypothesis
  - Linearity
  - Independent residuals
  - Homocedasticity
  - Gaussian residuals

White noise residuals

Dynamic Regression Models
Formulation

- Formulation (Pankratz):

$$y[t] = c + \frac{\omega(B)}{\delta(B)} x[t-b] + \nu[t]$$

where:

- *y[t]:* dependent output variable
- *x[t]:* independent or explanatory input variable (one for simplicity)
- $\nu[t]$: autocorrelated ARIMA noise
- $\omega(L) = (\omega_0 - \omega_1 B - \omega_2 B^2 - \ldots - \omega_s B^s)$
- $\delta(L) = (1 - \delta_1 B - \delta_2 B^2 - \ldots - \delta_r B^r)$
- *r, s, b* constant integers (*b* represents the delayed effect of *x* on *y*)

Dynamic Regression Models
Formulation

- The dynamic regression model requires determining the orders *r*, *s* and *b*, and the values of *p*, *d* and *q* of the ARIMA noise model.

- Two methods:
  - The traditional Box and Jenkins (1970) method based on cross-correlations
  - The LTF method (Linear Transfer Function) proposed by Liu & Hanssens (1982) and Pankratz (1991)

Dynamic Regression Models
Model identification using the LTF method

1) Transform the series for stabilizing the variance

2) Fit a multiple regression model of the form:

$$y[t] = c + \alpha_0 x[t] + \alpha_1 x[t-1] + \alpha_2 x[t-2] + \ldots + \alpha_k x[t-k] + \nu[t]$$

with a large (8-10) *k* and a low order AR model for the noise.

3) If the regression errors are not stationary, then differentiate *y* and *x*. Fit the model with the differentiated series (or include a unit root in the noise model).

4) If the regression errors are stationary, identify the transfer function $\alpha(L)$ by selecting appropriate values for *b, r* and *s:*

- The value of *b* is selected as the number of samples it takes for the output to respond to the input.
- The value of *r* (order of $\delta(L)$) determines the pattern of decay in the impluse response weights.
- The value of *s* (order of $\omega(L)$) determines where the pattern of decay in the impulse response weights begins.

Dynamic Regression Models
Model identification using the LTF method

General Rules:

- For the determination of b, we analyse the number of initial non-significant coefficients ($\alpha_0, \alpha_1, \ldots, \alpha_{b-1}$)

- The value of r determines the pattern of decay of the coefficients α_i:
  - If there is no pattern of decay, but a set of non-zero coefficients followed by a cut to zero, we take $r=0$
  - If the pattern of decay is exponential, we take $r=1$
  - If the pattern of decay is damped exponential or damped sine wave, we take $r=2$

- The value of s determines the number of non-null α_i coefficients before the decay.

Model identification using the LTF method

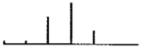
5) Identify an ARMA model for the regression errors $\nu[t]$.

6) Fit the complete model with the identified TF and ARMA model.

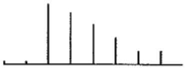
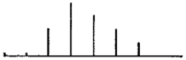
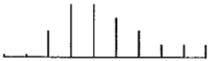
7) Analyze the residual $\varepsilon[t]$ using the general procedure.

Model identification using the LTF method

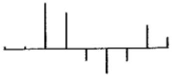
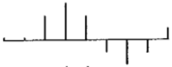
Transfer function for $r = 0$.

(b,r,s)	Transfer function	Typical impulse weights
(2, 0, 0)	$v(B)x_t = \omega_0 x_{t-2}$	
(2, 0, 1)	$v(B)x_t = (\omega_0 - \omega_1 B)x_{t-2}$	
(2, 0, 2)	$v(B)x_t = (\omega_0 - \omega_1 B - \omega_2 B^2)x_{t-2}$	

Transfer function for $r = 1$.

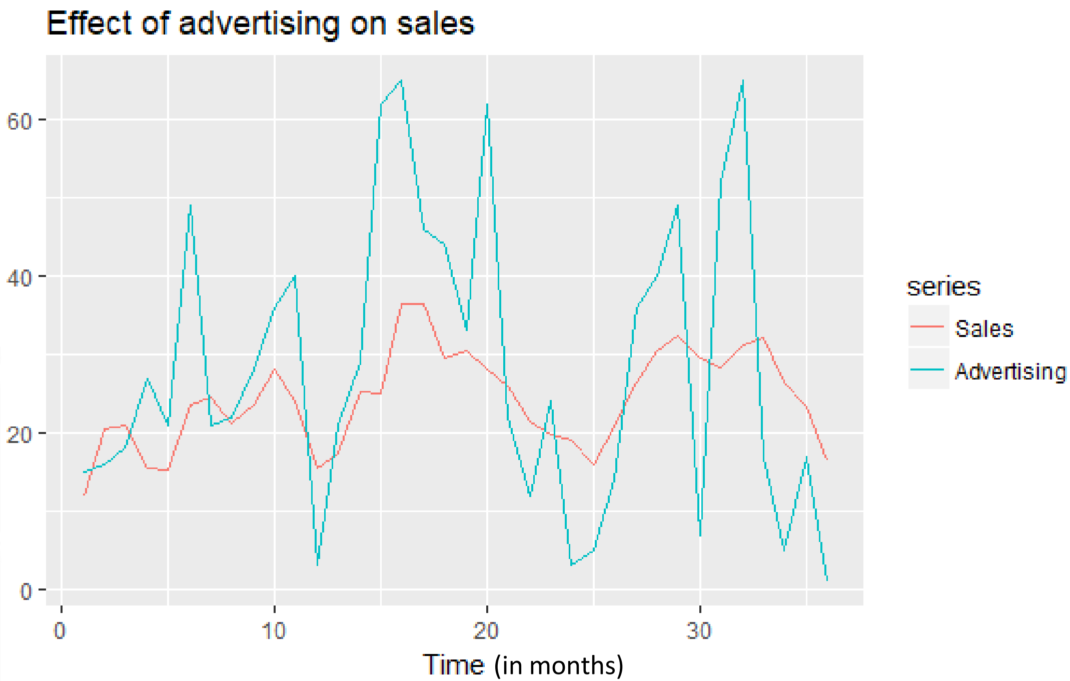
(b,r,s)	Transfer function	Typical impulse weights
(2, 1, 0)	$v(B)x_t = \dfrac{\omega_0}{(1 - \delta_1 B)}x_{t-2}$	
(2, 1, 1)	$v(B)x_t = \dfrac{(\omega_0 - \omega_1 B)}{(1 - \delta_1 B)}x_{t-2}$	
(2, 1, 2)	$v(B)x_t = \dfrac{(\omega_0 - \omega_1 B - \omega_2 B^2)}{(1 - \delta_1 B)}x_{t-2}$	

Transfer function for $r = 2$.

(b,r,s)	Transfer function	Typical impulse weights
(2, 2, 0)	$v(B)x_t = \dfrac{\omega_0}{(1 - \delta_1 B - \delta_2 B^2)}x_{t-2}$	
(2, 2, 1)	$v(B)x_t = \dfrac{(\omega_0 - \omega_1 B)}{(1 - \delta_1 B - \delta_2 B^2)}x_{t-2}$	
(2, 2, 2)	$v(B)x_t = \dfrac{(\omega_0 - \omega_1 B - \omega_2 B^2)}{(1 - \delta_1 B - \delta_2 B^2)}x_{t-2}$	

[Wei, 2006]

Dynamic Regression Models
Example



Dynamic Regression Models
Example

1) Fit the multiple regression model with AR(1) noise:

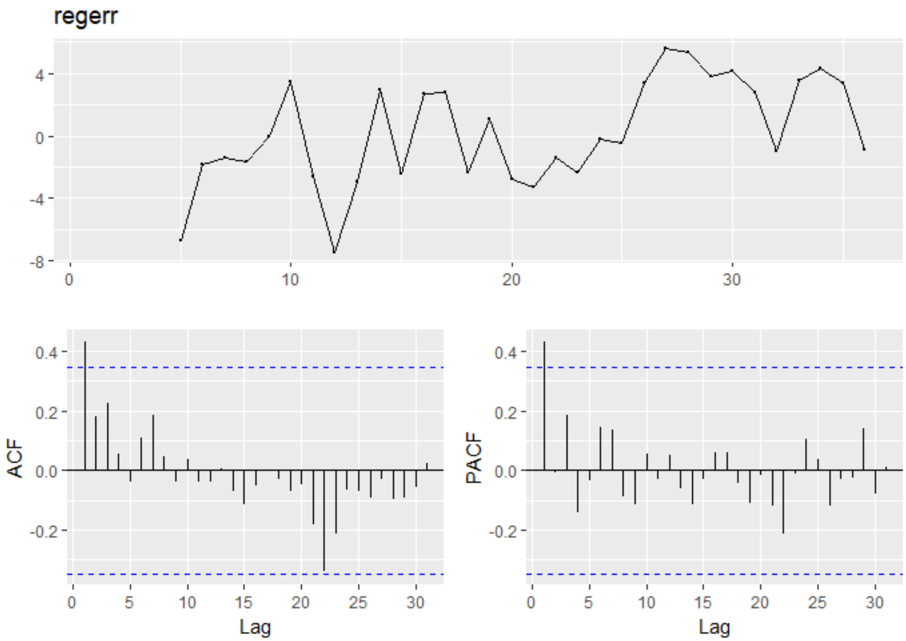
```
> library(TSA)
> arima.fit <- arima(y,
           order=c(1,0,0),              # ARIMA order for the noise
           xtransf = lag(x,0),          # Exogenous vars and lag order b
           transfer = list(c(0,4)),     # List with r and s orders
           include.mean = TRUE,         # Include intercept
           method = "ML")               # Optimization method
> summary(arima.fit)

Call:
arima(x = fdata_ts[, 1], order = c(1, 0, 0), include.mean = TRUE, method = "ML",
    xtransf = lag(fdata_ts[, 2], 0), transfer = list(c(0, 4)))

Coefficients:
         ar1  intercept  T1-MA0  T1-MA1  T1-MA2  T1-MA3  T1-MA4
      0.4873    13.7153  0.1311  0.1508  0.0497  0.0370  0.0007
s.e.  0.1786     2.7422  0.0288  0.0297  0.0300  0.0309  0.0331

sigma^2 estimated as 8.62:  log likelihood = -80.01,  aic = 174.02

Training set error measures:
                    ME     RMSE      MAE       MPE     MAPE      MASE        ACF1
Training set 0.1293064 2.936058 2.482481 -0.959295 10.56889 0.6673335 -0.04261899
```

$$y[t] = 13.7 + 0.13x[t] + 0.15x[t-1] + 0.05x[t-2] + 0.04x[t-3] - 0.0007x[t-4] + \nu[t]$$

Dynamic Regression Models
Example

2) Check regression errors are stationary ⇒ differencing is not necessary

```
> #Plot regression errors
> TF.RegressionError.plot(y, x, arima.fit, lag.max = 50)
```



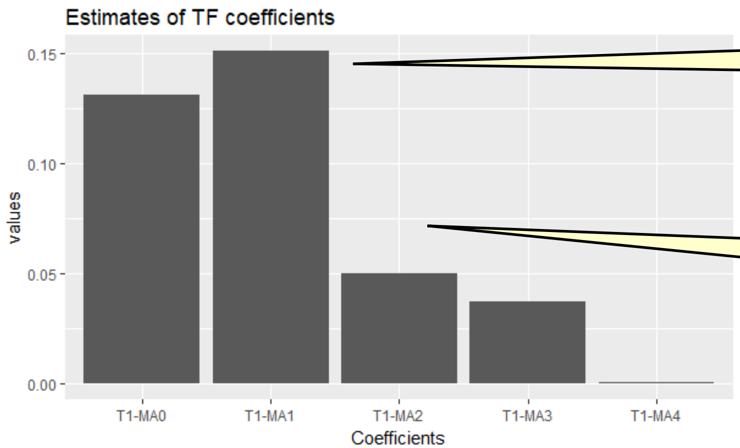
Dynamic Regression Models
Example

3) Identification of the TF: $$y[t] = c + \frac{\omega(B)}{\delta(B)} x[t-b] + \nu[t]$$

```
> source("ArimaTF.R")
> TF.Identification.plot(arima.fit) #Plot values of coefficients
```

	Estimate	Std. Error	z value	Pr(>|z|)
T1-MA0	0.1311112131	0.02884334	4.54563290	5.477042e-06
T1-MA1	0.1507797580	0.02968224	5.07979701	3.778384e-07
T1-MA2	0.0497254457	0.02997718	1.65877662	9.716081e-02
T1-MA3	0.0370476005	0.03088886	1.19938398	2.303787e-01
T1-MA4	0.0006600168	0.03310469	0.01993726	9.840934e-01

The first coef. Is significative: *b=0*



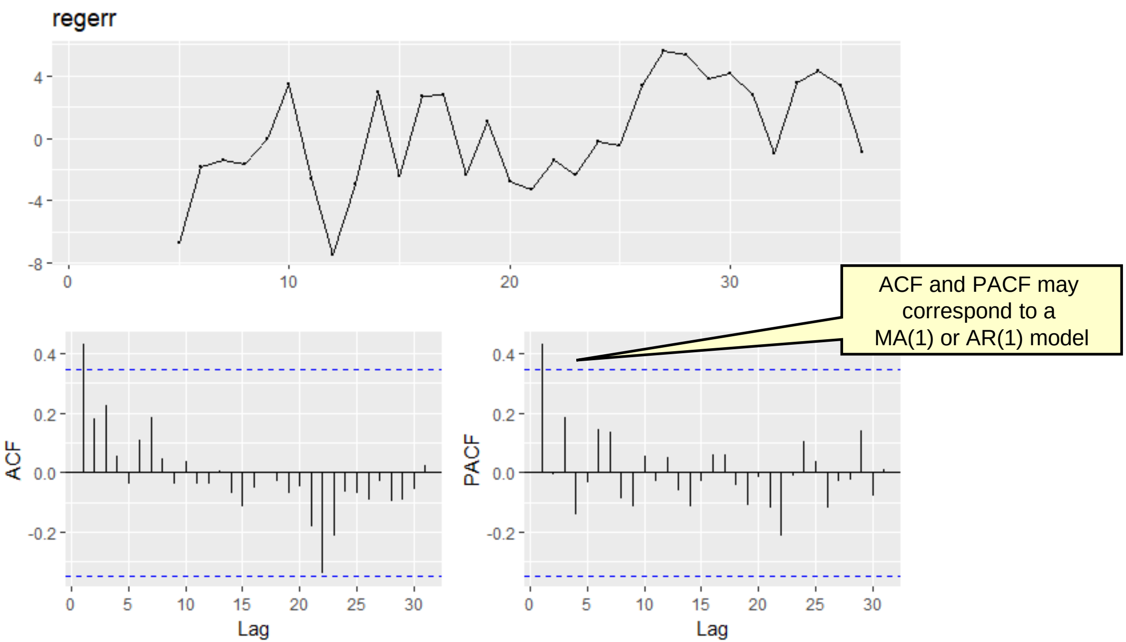
One significant coef. before the decay: *s=1*

Exponential decay: *r=1*

Dynamic Regression Models
Example

4) Identification of the ARMA model for the noise:

```
> #Plot regression errors
> TF.RegressionError.plot(y, x, arima.fit, lag.max = 50)
```



Dynamic Regression Models
Example

5) Fit the final model

$$y[t] = c + \frac{\omega_0 - \omega_1 B}{1 - \delta_1 B} x[t] + v[t]$$

with

$$v[t] = \varepsilon[t] - \theta\varepsilon[t-1]$$

```
> arima.fit <- arima(y,
            order=c(0,0,1),              # ARIMA order for the noise: MA(1)
            xtransf = lag(x,0),          # Exogenous vars and lag order b=0
            transfer = list(c(1,1)),     # List with r=1 and s=1 orders
            include.mean = TRUE,         # Include intercept
            method = "ML")               # Optimization method
```

Dynamic Regression Models
Example

5) Results in:

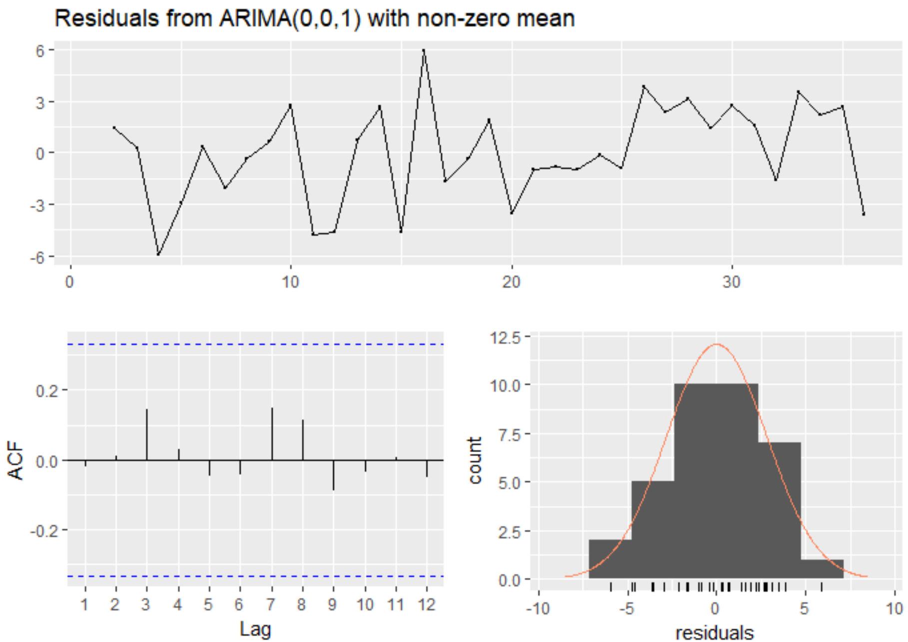
```
> summary(arima.fit)
Call:
arima(x = fdata_ts[, 1], order = c(0, 0, 1), include.mean = TRUE, method = "ML",
    xtransf = lag(fdata_ts[, 2], 0), transfer = list(c(1, 1)))

Coefficients:
         ma1  intercept  T1-AR1  T1-MA0  T1-MA1
      0.7234    14.4484  0.2888  0.1174  0.1294
s.e.  0.1370     1.9957  0.1475  0.0270  0.0327

sigma^2 estimated as 7.749:  log likelihood = -85.87,  aic = 181.73

Training set error measures:
                       ME     RMSE     MAE       MPE     MAPE      MASE        ACF1
Training set -0.004569443 2.783784 2.28845 -1.563531 9.935962 0.6151747 -0.01882031
```

$$y[t] = 14.45 + \frac{0.12 + 0.13B}{1 - 0.29B} x[t] + (1 + 0.72B)\varepsilon[t]$$

Dynamic Regression Models
Example

6) Analysis of the residuals: the white noise assumption for the residuals is accepted

```
> checkresiduals(arima.fit)
```



Model Diagnosis

- Tests on the parameters:
  - Check whether the model can be simplified by eliminating operators with values close in numerator and denominator:

$$y[t]=\frac{5-2.8B}{1-0.6B}x[t]+\frac{1-0.3B}{1-1.1B+0.2B^2}\varepsilon[t]\approx 5x[t]+\frac{1}{1-0.8B}\varepsilon[t]$$

  - The roots of the AR polinomials should fulfill the stability conditions.
  - Check that all the coefficients are significant and have a reasonable physical meaning (in particular the sign of the coefficients of the TF)

Dynamic Regression Models
Model Diagnosis

- Tests on the residuals:
  - Basically we need to check:
    - Gaussianity
    - With zero mean
    - Uncorrelated residuals
  - In order to analyze specification errors, suppose that the true model is:
    $$y[t] = \alpha(B)x[t] + \beta(B)\varepsilon[t]$$

    but we have estimated:
    $$y[t] = \hat{\alpha}_1(B)x[t] + \hat{\beta}_1(B)\hat{\varepsilon}[t]$$

    equating both expressions:
    $$\hat{\alpha}_1(B)x[t] + \hat{\beta}_1(B)\hat{\varepsilon}[t] = \alpha(B)x[t] + \beta(B)\varepsilon[t]$$

    and hence:
    $$\boxed{\hat{\varepsilon}[t] = \frac{\alpha(B) - \hat{\alpha}_1(B)}{\hat{\beta}_1(B)}x[t] + \frac{\beta(B)}{\hat{\beta}_1(B)}\varepsilon[t]}$$

Dynamic Regression Models
Model Diagnosis

$$\hat{\varepsilon}[t] = \frac{\alpha(B) - \hat{\alpha}_1(B)}{\hat{\beta}_1(B)} x[t] + \frac{\beta(B)}{\hat{\beta}_1(B)} \varepsilon[t]$$

In this expression, the following 4 cases may arise:

1. We have specified wrong both the transfer function and the noise model. In this case the estimated residuals are autocorrelated and will be correlated with *x[t]*.

2. If the transfer function is incorrect, although the noise model is correct, we will observe correlation between the residuals and *x[t]*, but also autocorrelation in the residuals by the filtered effect of *x[t]*:

$$\hat{\varepsilon}[t] = \frac{\alpha(B) - \hat{\alpha}_1(B)}{\hat{\beta}_1(B)} x[t] + \varepsilon[t]$$

3. If the transfer function is correct and the disturbance model is not, there will be residual autocorrelation but no correlation will be observed between the estimated residuals and *x[t]*.

4. If both are correct, no cross-correlation or autocorrelation are observed.

Model identification using the CCF

1) For a dynamic regression model of the form: $y[t] = \nu(L)x[t] + n[t]$

2) If we assume that *x[t]* follows an ARMA process: $\Phi_x(L)x[t] = \Theta_x(L)\alpha[t]$

where $\alpha[t]$ is a white noise process, called the pre-whitened input: $\alpha[t] = \dfrac{\Phi_x(L)}{\Theta_x(L)} x[t]$

3) If we apply the same filter to the input: $\beta[t] = \dfrac{\Phi_x(L)}{\Theta_x(L)} y[t]$

we obtain: $\boxed{\beta[t] = \nu(L)\alpha[t] + \xi[t]}$ where $\xi[t] = \dfrac{\Phi_x(L)}{\Theta_x(L)} n[t]$

and the impulse response weights for the transfer function can therefore be found as:

$$\boxed{\nu_k = \frac{\sigma_\beta}{\sigma_\alpha} \rho_{\alpha\beta}[k]}$$

as $\alpha[t]$ is a white noise process.

Model identification using the CCF

Based on the above discussion, the transfer function is obtained from the following simple steps:

1) Prewhiten the input series: $\alpha[t] = \dfrac{\Phi_x(L)}{\Theta_x(L)} x[t]$

where $\alpha[t]$ is a white noise process with zero mean and σ^2_α variance.

2) Calculate the filtered output series: $\beta[t] = \dfrac{\Phi_x(L)}{\Theta_x(L)} y[t]$

3) Calculate the sample CCF between $\alpha[t]$ and $\beta[t]$ to estimate v_k: $\hat{v}_k = \dfrac{\sigma_\beta}{\sigma_\alpha} \hat{\rho}_{\alpha\beta}[k]$

4) Identify b, r and s by matching the pattern of v_k with the known theoretical patterns.

5) Identify the noise model by analysing the regression error (check for stationarity):

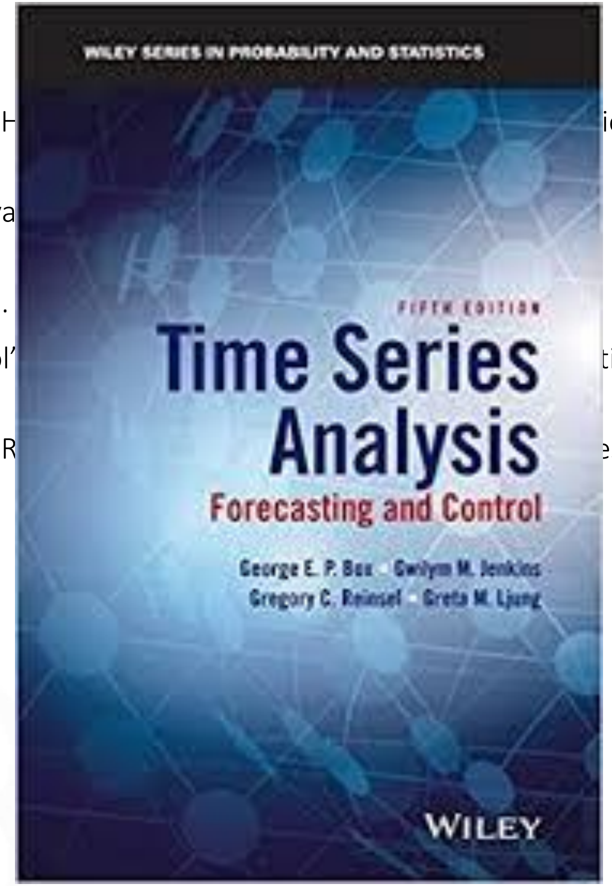
$$\hat{n}[t] = y[t] - \hat{v}(L)x[t] = y[t] - \frac{\hat{\omega}(L)}{\hat{\delta}(L)} L^b x[t]$$

7

Bibliography

Bibliography

- "Forecasting Principles and Practice". Rob J H ... ion. Otexts. 2017 (http://otexts.org/fpp3/)
- "Time series analysis. Univariate and Multiva ... Pearson Addison Wesley. 2006.
- "Análisis de series temporales". Daniel Peña.
- "Time Series Analysis: Forecasting & Control" ... tice Hall. 1994
- "Princples of Business Forecasting". K. Ord, R ... ess. 2017



Alberto Aguilera 23, E-28015 Madrid - Tel: +34 91 542 2800 - http://www.iit.comillas.edu