Non Stationary Processes, ARIMA and SARIMA processes

Dhafer Malouche

Let's recall that

- ARMA(p,q) Process $\Phi(B)X_t = \Theta(B)Z_t$ where $Z_t \sim \text{WN}(0, \sigma^2)$, and
  - $\Phi(z) = 1 + \phi_1 z + \ldots + \phi_p z^p$,
  - $\Theta(z) = 1 + \theta_1 z + \ldots + \theta_q p z^q$
- Then
  - If $\Phi(z) \neq 0$, $\forall$ z, $|z| = 1$, (X_t) is stationary
  - If $\Phi(z) \neq 0$, $\forall$ z, $|z| \leq 1$, (X_t) is causal
  - If $\Theta(z) \neq 0$, $\forall$ z, $|z| \leq 1$, (X_t) is invertible
- If $(X_t) \sim \text{MA}(q)$ then $\rho_X(h) = 0$ for all $|h| > q$
- If $(X_t) \sim \text{AR}(p)$ then $r_X(h) = 0$ for all $|h| > p$

Outline

Introducing SARIMA Models

ARIMA models

- (X_t) is called an **autoregressive integrated moving average** (ARIMA) model with order (p, d, q), if

$$\Delta^d X_t = (1 - B)^d X_t$$

is an ARMA(p, q).

- We write the model as

$$\Phi(B)(1 - B)^d X_t = \Theta(B) Z_t$$

where $(Z_t) \sim$WN$(0, \sigma^2)$, Φ and Θ are polynomials such that Φ doesn't have unit roots.

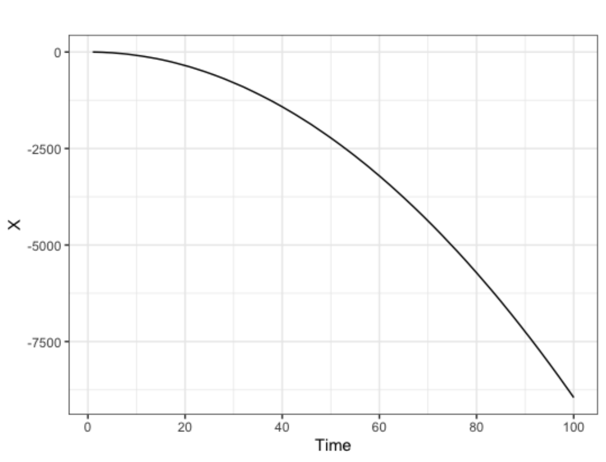
Example: (1)

$$X_t = \underbrace{.2 + .5t - .9t^2}_{\text{quadratic trend}} + Z_t$$

where $(Z_t) \sim \mathrm{WN}(0, \sigma^2)$

```
> Z=arima.sim(n = 100,model = list())
> X=.2+.5*(1:100)-.9*(1:100)^2+Z
> X=ts(X)
> library(ggplot2)
> forecast::autoplot(X)+theme_bw()
```

Example: (1)



where

```
> Z=a
> X=.
> X=t
> lib
> for
```

Example: (2)

$$X_t = a + bt + ct^2 + Z_t.$$

$$-2X_{t-1} = -2a - 2b(t-1) - 2c(t-1)^2 - 2Z_{t-1}$$

$$X_{t-2} = a + b(t-2) + c(t-2)^2 + Z_{t-2}.$$

$$(1-B)^2 X_t = \cancel{bt} - \cancel{2bt} + \cancel{2b} + \cancel{bt} - \cancel{2b}$$

$$+ \cancel{ct^2} - \cancel{2ct^2} + \cancel{4ct} - 2c$$

$$+ \cancel{ct^2} - \cancel{4ct} + 4c + (1-B)^2 Z_t$$

$$- 2c + (1-B)^2 Z_t$$

Example: (2)

$$(1-B)^2X_t = \Delta^2X_t = 2c + (1-B)^2Z_t$$

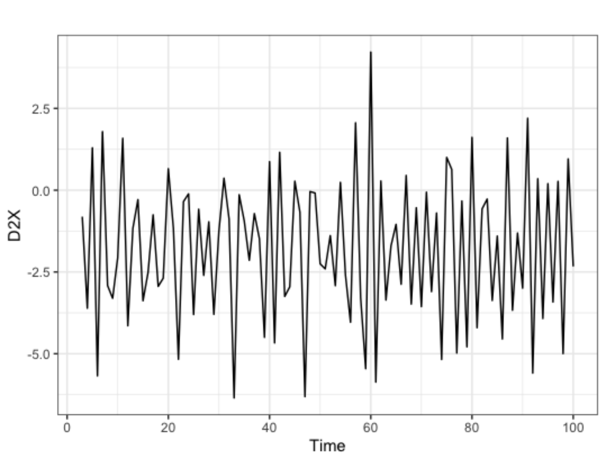
- $(1-B)^2X_t = \Delta^2X_t$ is a stationary process
- $\mathbb{E}(\Delta^2X_t) = 2c$ and $(1-B)^2Z_t \sim \mathrm{MA}(2)$

Example: (3)

```
> D2X=diff(X,differences = 2)
> D2Xa=diff(diff(X))
> sum(D2Xa-D2X==0)
[1] 98
> length(D2X)
[1] 98
> forecast::autoplot(D2X)+theme_bw()
```

Example: (3)

```
> D2X
> D2X
> sum
[1] 9
> len
[1] 9
> for
```

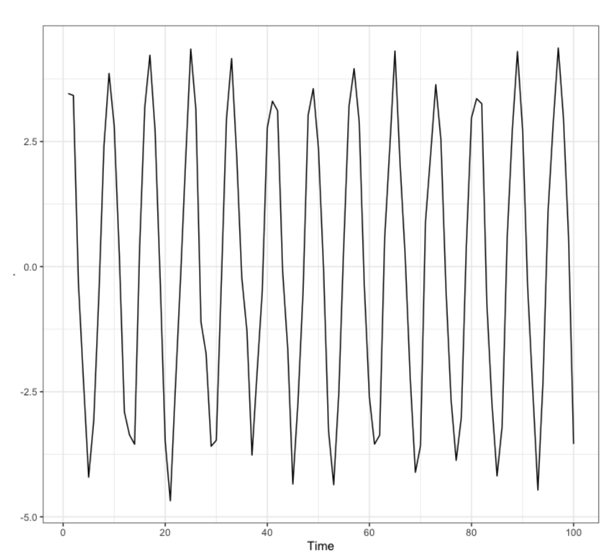


Example with seasonality

$$Y_t = \underbrace{4\cos\left(\frac{\pi\, t}{4}\right)}_{\text{Seasonal}} + X_t, \quad (X_t) \sim \text{ARMA}(1,1)$$

```
> x=arima.sim(n =100, list(ar =  -0.4, ma =  0.25),sd = .5)
> t=0:99
> y=4*cos(pi/4*t)+x
> y=as.ts(y)
```

Exam

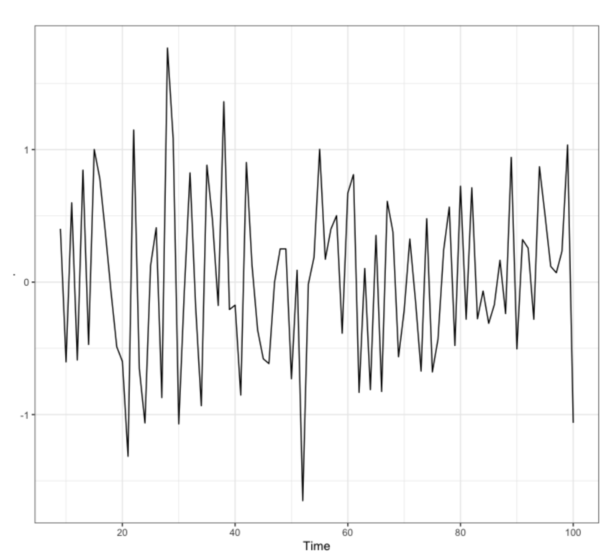



```
> x=a                                                    d = .5)
> t=0
> y=4
> y=a
```

Example with seasonality

$$\Delta_8 Y_t = (1 - B^8) Y_t$$

Exam



Example with seasonality

- $\Delta_8 Y_t = (1 - B^8) Y_t$ is a stationary process
- $$\begin{aligned} Y_{t-8} &= \cos\left(\frac{\pi\,(t-8)}{4}\right) + X_{t-8} \\ &= \cos\left(\frac{\pi\, t}{4}\right) + X_{t-8} \\ &= Y_t - X_t + X_{t-8} \end{aligned}$$
- Then $(1 - B^8) Y_t = (1 - B^8) X_t$
- $\gamma_{(1-B^8)X}(h) = \gamma(h) - \gamma(h-8) - \gamma(h+8) + \gamma(4)$, $(1 - B^8) X_t$ is stationary

Operators

- Trend Operator: $\Delta^d = (1 - B)^d$
  `>diff(,differences =d)`

$$\Delta^d X_t = (1 - B)^d X_t = \sum_{k=0}^{d} \binom{d}{k} (-1)^k X_{t-k}$$

- Seasonal Operator: $\Delta_d = (1 - B^d)$
  `>diff(,lag =d)`

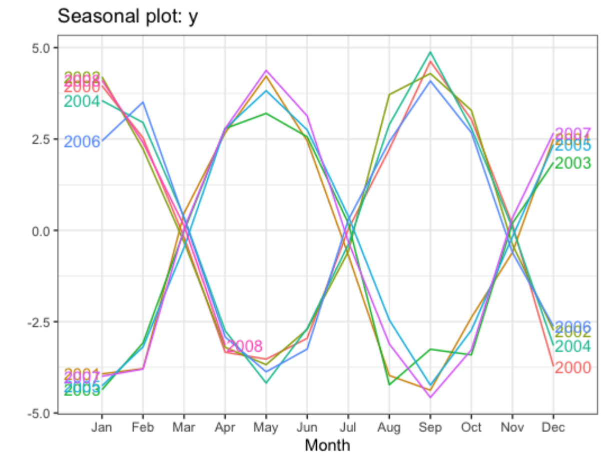
$$\Delta_d X_t = (1 - B^d)^d X_t = X_t - X_{t-d}$$

Example with seasonality

```
> forecast::ggseasonplot(y,year.labels = T,
+   year.labels.left = T)+
+     theme_bw()
```

Example with seasonality

```
> for
+    y
+
```



Seasonal ARIMA

The process (X_t) is a SARIMA$(p, d, q)_S$ process, *Seasonal ARIMA*, if the process

$$\Delta_S \times \Delta^d X_t = \Delta^d \times \Delta_S X_t = (1 - B^S)(1 - B)^d X_t = (1 - B)^d (1 - B^S) X_t$$

is an ARMA(p, q),
$\iff$

$$\Delta_S \times \Delta^d X_t = c + \frac{\Theta(B)}{\Phi(B)} Z_t, \ \ \forall\, t \in \mathbb{Z}$$

where $(Z_t) \sim$WN$(0, \sigma^2)$ and Φ and Θ are polynominals.

ARIMA, Example with R

$$(1 - B)^3 X_t, = Z_t$$

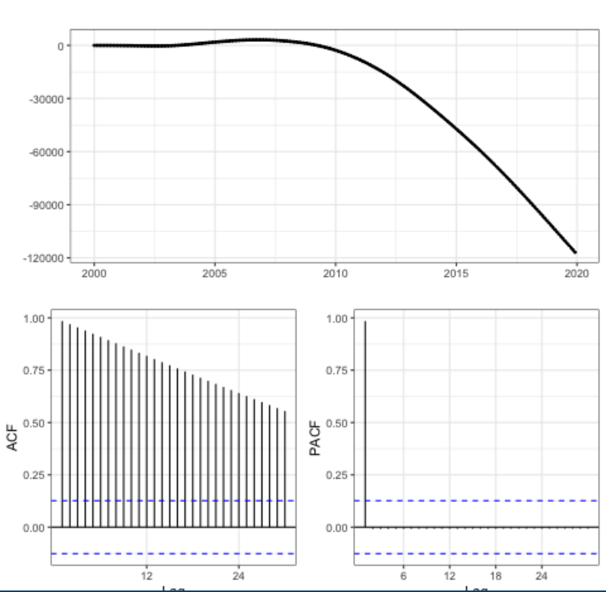
where $(Z_t) \sim$WN$(0, \sigma^2)$, then $(X_t) \sim$ SARIMA$(0, 3, 0)_0$

```
> library(sarima)
> library(forecast)
> library(ggplot2)
> set.seed("34567")
> x=sim_sarima(n=240,model=list(iorder=3))
> x=ts(x,start=c(2000,1),frequency=12)
> ggtsdisplay(x,lag.max=30,theme = theme_bw())
```

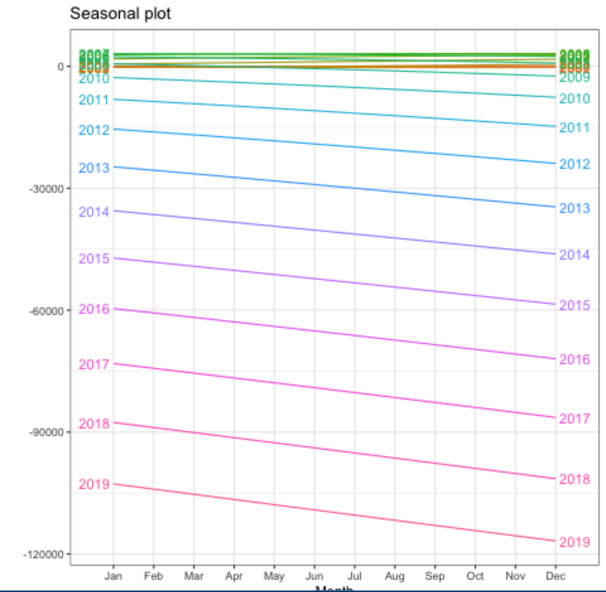
ARIM

where

```
> libra
> libra
> libra
> set.s
> x=sir
> x=ts
> ggtsc
```



ARIM



where

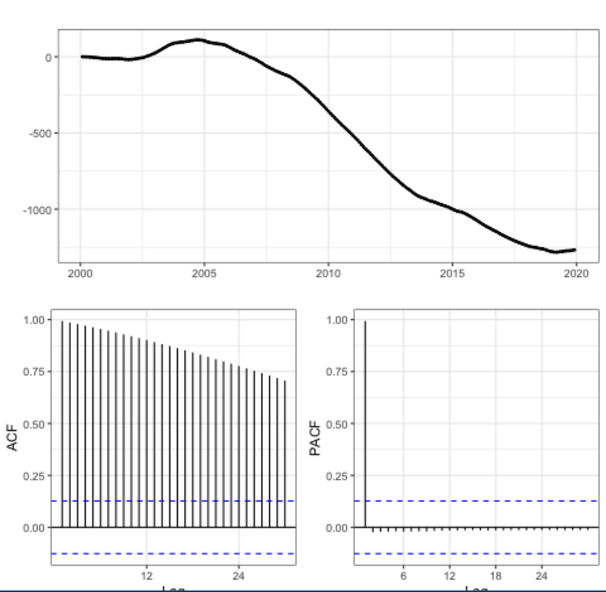
```
> libra
> libra
> libra
> set.s
> x=sin
> x=ts
> ggtsc
```

ARIMA

$$(1 - B)X_t$$

```
> ggtsdisplay(x%>%diff(difference=1),lag.max=30,theme = theme_bw())
```

ARIM



0
-500
-1000
2000
2005
2010
2015
2020
ACF
PACF
1.00
0.75
0.50
0.25
0.00
6
12
18
24

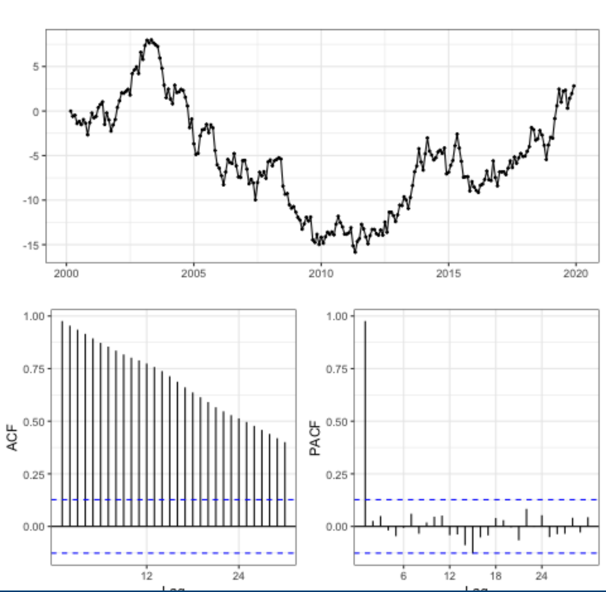

```
> ggts                                                              w())
```

ARIMA

$$(1 - B)^2 X_t$$

```
> ggtsdisplay(x%>%diff(difference=2),lag.max=30,theme = theme_bw())
```

ARIM



5
0
-5
-10
-15
2000
2005
2010
2015
2020
ACF
1.00
0.75
0.50
0.25
0.00
12
24
PACF
1.00
0.75
0.50
0.25
0.00
6
12
18
24

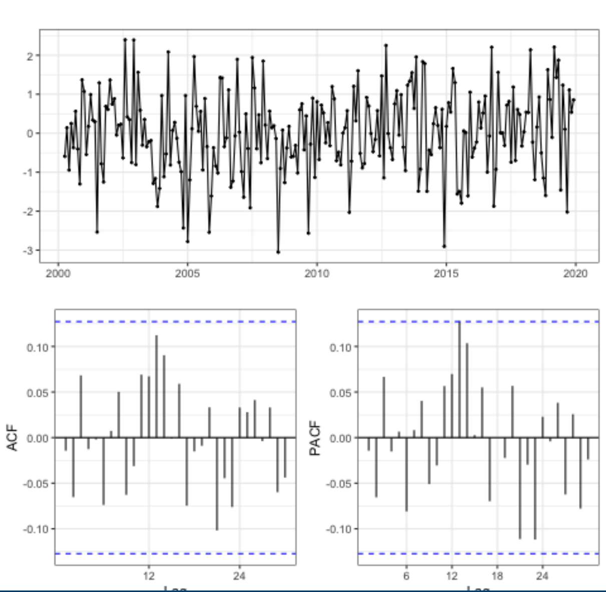

```
> ggts                                              w())
```

ARIMA

$$(1-B)^3 X_t$$

```
> ggtsdisplay(x%>%diff(difference=3),lag.max=30,theme = theme_bw())
```

ARIM



2
1
0
-1
-2
-3
2000
2005
2010
2015
2020
ACF
0.10
0.05
0.00
-0.05
-0.10
12
24
PACF
0.10
0.05
0.00
-0.05
-0.10
6
12
18
24


```
> ggts                                                   w())
```

SARIMA Models, General expression (1/3)

The stochastic process $(X_t)_{t\in\mathbb{Z}}$ is a called SARIMA$(p,d,q)(P,D,Q)_S$ if it satisfies the following

$$(1-B)^d(1-B^S)^D X_t = (1-B^S)^D(1-B)^d X_t =$$

$$c + \frac{(1+\theta_1 B + \ldots + \theta_q B^q)(1+\Theta_1 B^S + \ldots + \Theta_Q B^{SQ})}{(1-\phi_1 B - \ldots - \phi_p B^p)(1-\Phi_1 B^S - \ldots - \Phi_P B^{SP})} Z_t$$

where $(Z_t)_{t\in\mathbb{Z}}$ is a WN$(0,\sigma^2)$

$$\text{SARIMA} \quad \underbrace{(p,d,q)}_{\text{Non-seasonal}} \quad \underbrace{(P,D,Q)_S}_{\text{Seasonal}}$$

SARIMA Models, General expression (2/3)

- $S \geq 2$ is the number of seasons, number of observations per period (year), `nseasons`;
- d is the order of differencing, `iorder`;
- D order of seasonal differencing, `siorder`;
- $\Phi = (\phi_1, \ldots, \phi_p)$ AR parameters (non-seasonal), `ar`;
- $\Theta = (\theta_1, \ldots, \theta_q)$ MA parameters (non-seasonal), `ma`;
- $\Phi_s = (\Phi_1, \ldots, \Phi_P)$ seasonal SAR parameters, `sar`;
- $\Theta_s = (\Theta_1, \ldots, \Theta_Q)$ seasonal SMA parameters, `ma`;

SARIMA Models, General expression (3/3)

$$(1-B)^d(1-B^S)^D X_t = c + \frac{\Theta(B)\Theta_s(B^S)}{\Phi(B)\Phi_s(B^S)} Z_t$$

where

- $\Phi(z) = 1 - \phi_1 z - \ldots - \phi_p z^p$
- $\Phi_s(z) = 1 - \Phi_1 z - \ldots - \Phi_P z^P$
- $\Theta(z) = 1 + \theta_1 z + \ldots + \theta_q z^q$
- $\Theta_s(z) = 1 + \Theta_1 z + \ldots + \Theta_Q z^Q$

Example of SARIMA Models

- AR(1)=SARIMA$(1,0,0)(0,0,0)_0$,

$$X_t = \frac{1}{1-0.8B} Z_t, \quad (Z_t) \sim WN(0, \sigma^2)$$

  - $d = D = T = 0$
  - $\Phi(z) = 1 - 0.8z$, $\Phi_s = \Theta = \Theta_s = 1$
- SMA$(1)_2$=SARIMA$(0,0,0)(0,0,1)_2$: $T = 2$, $d = D = 0$, $\Phi = \Phi_s = \Theta = 0$, and $\Theta_s(z) = 1 + .2z$

$$X_t = (1 + .2B^2) Z_t$$

X_t can be also considered as MA(2).

Example of SARIMA Models

- SAR$(1)_4$=SARIMA$(0,0,0)(1,0,0)_4$,

$$X_t = \frac{1}{1-0.8B^4} Z_t, \quad (Z_t) \sim WN(0, \sigma^2)$$

$d = D = 0$, $T = 4$, $\Phi = \Theta = \Theta_s = 1$, and $\Phi_s(z) = 1 - 0.8z$

Example: European quarterly retail trade

`euretail`: Quarterly retail trade index in the Euro area (17 countries), 1996-2011, covering wholesale and retail trade, and repair of motor vehicles and motorcycles. (Index: 2005 = 100).

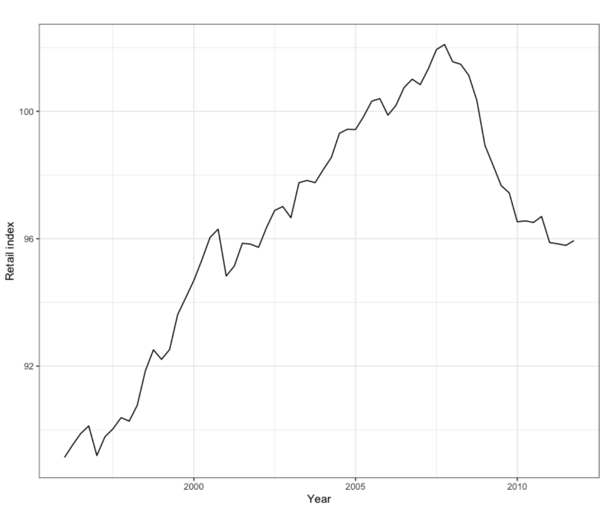
```
> library(fpp2)
> data("euretail")
> euretail%>%autoplot()+theme_bw() + ylab("Retail index") + xlab("Year")
> euretail
      Qtr1  Qtr2  Qtr3  Qtr4
1996 89.13 89.52 89.88 90.12
1997 89.19 89.78 90.03 90.38
1998 90.27 90.77 91.85 92.51
```

Example: European quarterly retail trade

eureta… …tries),
1996-2… …tor
vehicle…

```
> libra…
> data …
> euret… …b("Year")
> euret…
```

```
1996  8…
1997  8…
1998  9…
```

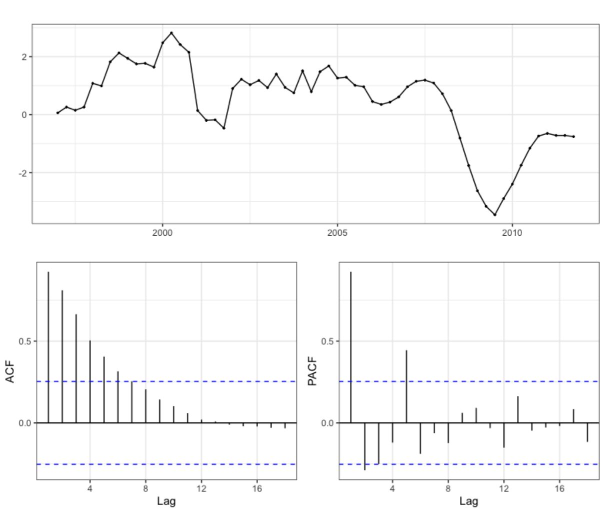


Example: European quarterly retail trade, $\Delta_4 X_t$.

```
> euretail %>% diff(lag=4) %>% ggtsdisplay(theme = theme_bw())
```

Example: European quarterly retail trade, $\Delta_4 X_t$

> eure



Example: European quarterly retail trade, $\Delta_4 X_t$.

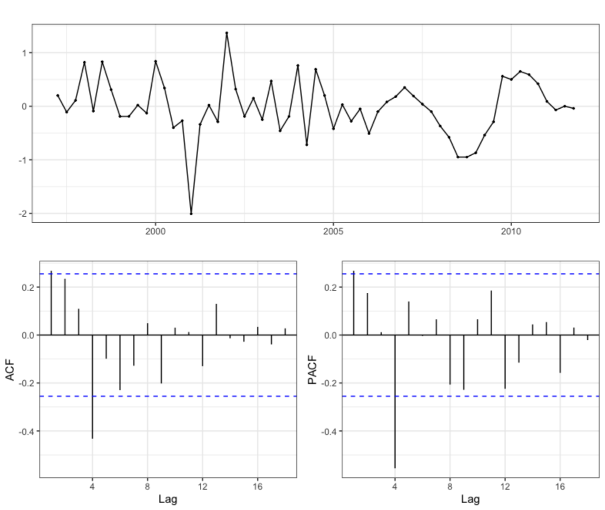
```
> euretail %>% diff(lag=4) %>% ggtsdisplay(theme = theme_bw())
```

Non-stationary of $\Delta_4 X_t$, we take an additional first difference,

Example: European quarterly retail trade, $\Delta_4 X_t$.

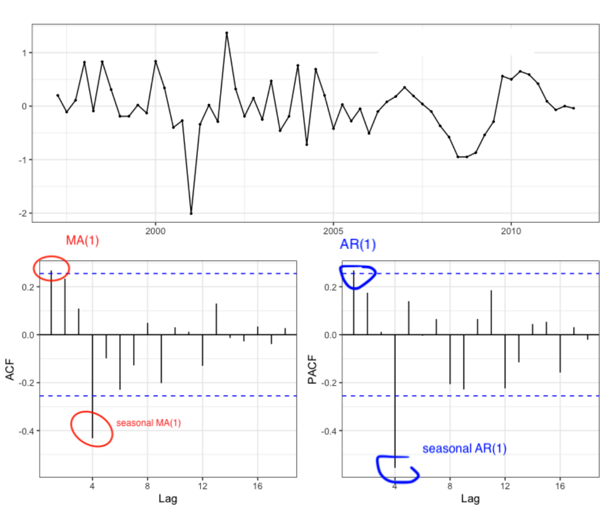
```
euretail %>% diff(lag=4) %>% diff() %>% ggtsdisplay(theme = theme_bw())
```

Example: European quarterly retail trade, $\Delta_4 X_t$



`euretai ... me_bw())`

Example: European quarterly retail trade, $\Delta_4 X_t$




```
euretai                                              me_bw())
```

Example: European quarterly retail trade, $\Delta_4 X_t$.

$$\text{SARIMA}(0,1,1)(1,1,1)_4$$

$$(1-B)(1-B^4)X_t = \frac{(1-\theta_1 B)(1-\Theta_1 B^4)}{(1-\phi_1 B)(1-\Phi_1 B^4)} Z_t$$

Example: Corticosteroid drug sales in Australia

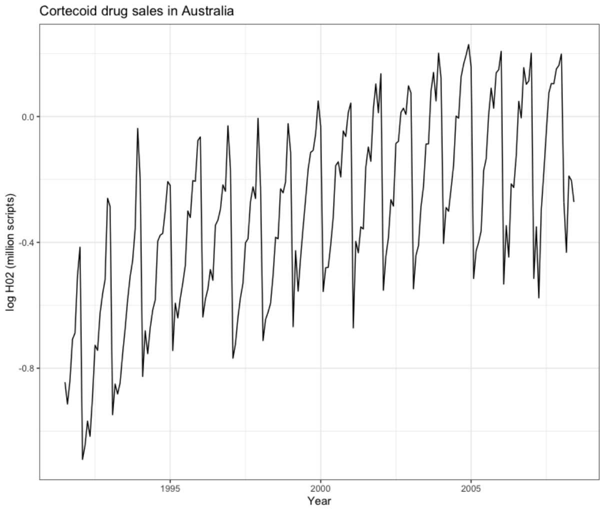
- Monthly corticosteroid drug sales in Australia from July 1991 to June 2008.
- Total monthly scripts for pharmaceutical products falling under ATC code H02, as recorded by the Australian Health Insurance Commission. Measured in millions of scripts.

```
> library(fpp2)
> data(h02)
> lh02=log(h02)
> lh02%>%autoplot()+theme_bw()+
+   ylab("log H02 (million scripts")+
+   xlab("Year")+ggtitle("Cortecoid drug sales in Australia")
```

Example: Corticosteroid drug sales in Australia

- M ... 1 to Ju
- T ... der A ... nce C

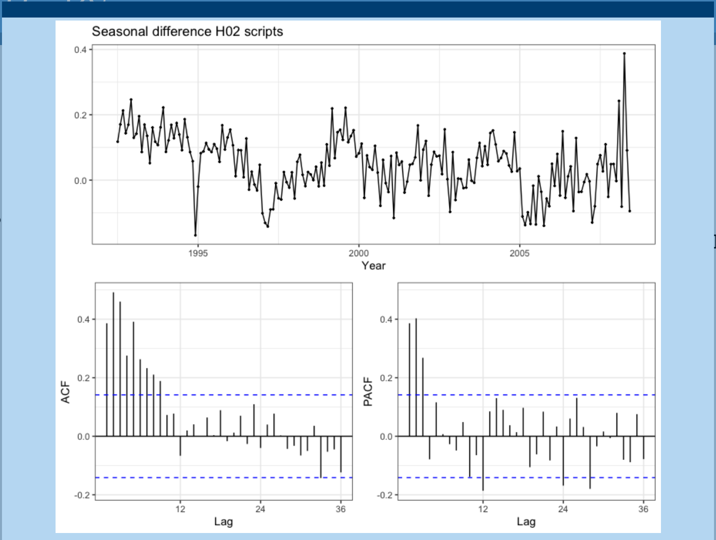
```
> libra
> data
> lh02=
> lh02%
+   yla
+   xla
```



Example: Corticosteroid drug sales in Australia, $(1 - B^{12})X_t$

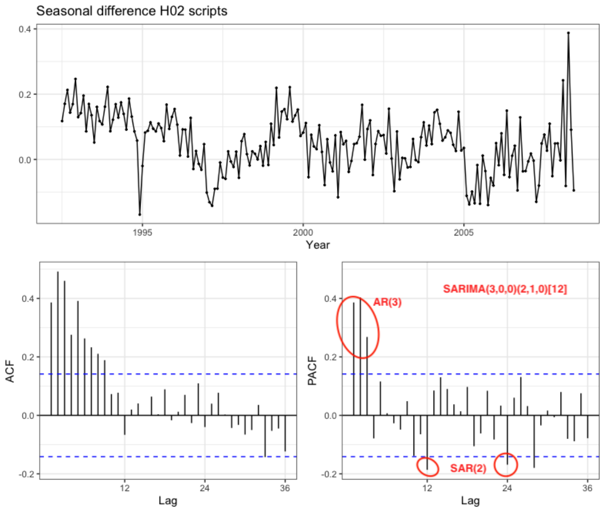
```
> lh02%>%diff(lag=12)%>%ggtsdisplay(theme = theme_bw(),
+                                   main="Seasonal difference H02 scripts",
+                                   xlab="Year")
```

Example: Corticosteroid drug sales in Australia, $(1 - B^{12})X_t$




```
> lh02%
+                                                               pts",
+
```

Example: Corticosteroid drug sales in Australia, $(1 - B^{12})X_t$




```
> lh02%
+                                                              pts",
+
```

Moving average smoothing

- When analyzing time series, we can assume an additive decomposition of a given time series:

$$x_t = S_t + T_t + R_t$$

  where
  - x_t is a time-series data,
  - S_t is the seasonal component,
  - T_t is the trend-cycle component
  - R_t is the remainder

- This decomposition can be also multiplicative: $x_t = S_t \times T_t \times R_t$

- Question: How can we estimate this decomposition?

Example with R

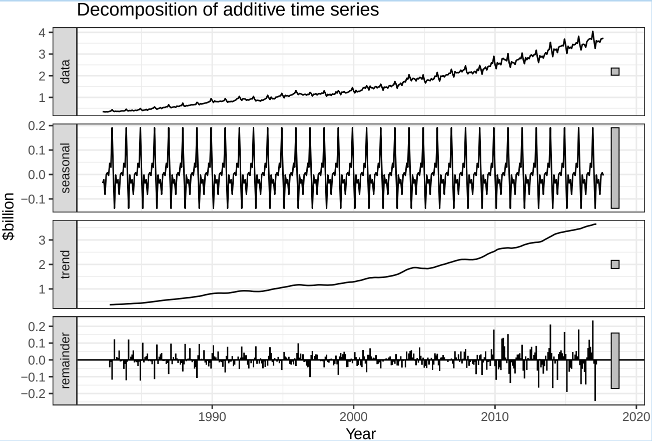
- `auscafe` data from `fpp2` package.
- Decomposition with R

```
> library(fpp2)
> data("auscafe")
> auscafe%>%decompose(type = "additive")%>%autoplot
    + theme_bw()
    + xlab("Year")
    + ylab("$billion")
```

Example with R

- aus
- Dec

```
> librar
> data(
> auscar
    + 
    + 
    + 
```



How does it work?

- T_t trend-cycle component is obtained using Moving-average smoothing.
- Since we have a monthly data, the function `decompose` computes uses a *centred moving average smoothing* of order 12.

```
> x=decompose(auscafe,type="additive")
> names(x)
```

```
## [1] "x"        "seasonal" "trend"    "random"   "figure"   "type"
```

How does it work?

```
> x$trend[1:20]
```

```
##  [1]        NA        NA        NA        NA        NA        NA 0.3565125
##  [8] 0.3576417 0.3591750 0.3610250 0.3639625 0.3673042 0.3694458 0.3707625
## [15] 0.3725042 0.3744250 0.3764750 0.3790167 0.3816917 0.3852625
```

```
> ma(auscafe,order = 12,centre = T)[1:20]
```

```
##  [1]        NA        NA        NA        NA        NA        NA 0.3565125
##  [8] 0.3576417 0.3591750 0.3610250 0.3639625 0.3673042 0.3694458 0.3707625
## [15] 0.3725042 0.3744250 0.3764750 0.3790167 0.3816917 0.3852625
```

How does it work?

- How the seasonal component is computed?

```
> z=auscafe-x$trend
> library(zoo)
> xm=months(as.yearmon(time(z)))
> xm=factor(xm,levels=unique(xm))
> dt=cbind.data.frame(z=z,xm=xm)
>
> library(dplyr)
> dd=dt%>%group_by(xm)%>%summarise(ave=mean(z,na.rm=T),n=n())
> xs=dd$ave
> xs=scale(xs,scale = F,center = T)
> xs
```

```
##                [,1]
##  [1,] -0.0348459007
##  [2,] -0.0211307047
##  [3,] -0.0813204105
##  [4,] -0.0006682047
```

How does it work?

- How the seasonal component is computed?

```r
> x$seasonal[1:12]
```

```
##  [1] -0.0348459007 -0.0211307047 -0.0813204105 -0.0006682047  0.0074998100
##  [6] -0.0040532537  0.0457839242  0.0297426147  0.1913607099  0.0078932099
## [11] -0.1384440520 -0.0018177425
```

Fitting SARIMA models with R

Using `sarima` from `astsa` package

- Input:
  - data: a univariate time series
  - p, d, q (must be specified) and P, Q, D and S (is the seasonal period)
- Output:
  - fitted parameters and their t-test
  - Error degrees of freedom
  - AIC, BIC, AICcc

Example

We will fit the `euretail` data with a SARIMA$(0,1,1)(0,1,1)_4$

```
> fit=sarima(xdata = euretail,p = 0,d = 1,q = 1,
+            P = 0,D = 1,Q = 1,S = 4,details = F)
> names(fit)
[1] "fit"                "degrees_of_freedom" "ttable"
[4] "AIC"                "AICc"               "BIC"
> fit$fit$coef
      ma1       sma1
0.2902981 -0.6912543
```

$$(1-B)(1-B^4)X_t = (1+0.290\times B)(1-0.691\times B^4)\,Z_t$$

Example

- Degree of freedom (T is the length of the TS)

$$\text{dof} = (T - DS - d) - \#\text{parameters} = (64 - 4 - 1) - 1 - 1 = 57$$

```
> fit$degrees_of_freedom
[1] 57
> length(euretail)
[1] 64
```

- aic (Akaiki Information Criteria, version 1)

$$\text{aic} = -2 \log \widehat{L}_T + 2(k+1)$$

where $\widehat{L}_k$ is the ML and k is the number of parameters (length of the vector `fit$fit$coef`).

AIC, AICc, BIC

- AIC: (Akaiki Information Criteria, version 2)

$$\mathrm{AIC}(k, T) = \frac{\mathrm{aic}}{T - d - D}$$

- AICc: (Akaiki Information Criteria, version 3)

$$\mathrm{AICc}(k, T) = \mathrm{AIC} + \frac{2k^2 + 2k}{(T - d - D - k - 1)(T - d - D)}$$

- BIC: (Bayesian Information Criteria)

$$\mathrm{BIC}(k, T) = \frac{-2 \log \widehat{L}_T + (k + 1) \times \log(T - d - D - k - 1)}{n - d - D}$$

Computing AIC, AICc and BIC with R

```
> n=length(euretail)
> d=1
> D=1
> n1=n-d-D
> ### Verifying AIC
> fit$fit$aic/n1
[1] 1.214208
> fit$AIC
[1] 1.214208
> ### Verifying AICc
> (n1 * fit$AIC + ((2 * k^2 + 2 * k)/(n1 - k - 1)))/n1
[1] 1.217488
> fit$AICc
[1] 1.217488
> ### Verifying BIC
> (-2*fit1$loglik+(k+1)*log(n1-k-1))/n1
[1] 1.314734
> fit$BIC
[1] 1.314734
```

Why AIC, AICc and BIC?

- Simulate a time series: MA(5);

$$X_t = Z_t + 0.8 Z_{t-5}$$

- **Exercise:** Compute $\rho(k)$ the autocorrelations and $r(k)$ the partial autocorrelation for $h \in \mathbb{N}$.
- Estimate 40 models: MA(j), $j = 1, \ldots, 40$
- Observe $j \mapsto \log \widehat{L}(j)$ where $\widehat{L}(j)$ is the likelihood of the estimated model MA(j), $j = 1, \ldots, 40$.

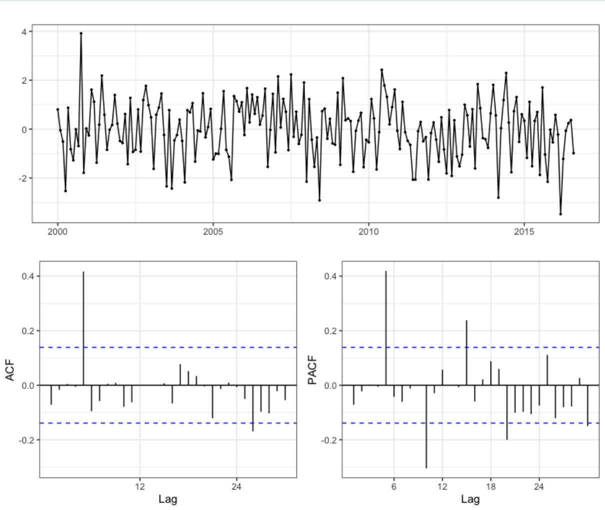
Why AIC, AICc and BIC?, with R

1. Simulate a TS from SARIMA$(0,0,5)(0,0,0)_0$
2. Estimate the models $\{\mathcal{M}_q, \;\text{ where } \mathcal{M}_q = \text{MA}(q),\; q = 1\dots 40\}$
3. Compute and compare $\text{LogLik}_q = \text{LogLik}(\mathcal{M}_q)$, $\text{AIC}_q = \text{AIC}(\mathcal{M}_q)$,..

```
> set.seed(345789)
> x <- sim_sarima(n=200, model = list(ma=c(rep(0,4),0.8)))
> x=ts(x,start=c(2000,1),frequency=12)
> ggtsdisplay(x,lag.max=30,theme = theme_bw())
```

Why AIC, AICc and BIC? with R

1. Si
2. Es ... 40}
3. C
   A




```
> set.
> x <-                                              )
> x=ts
> ggts
```

Why AIC, AICc and BIC?, with R

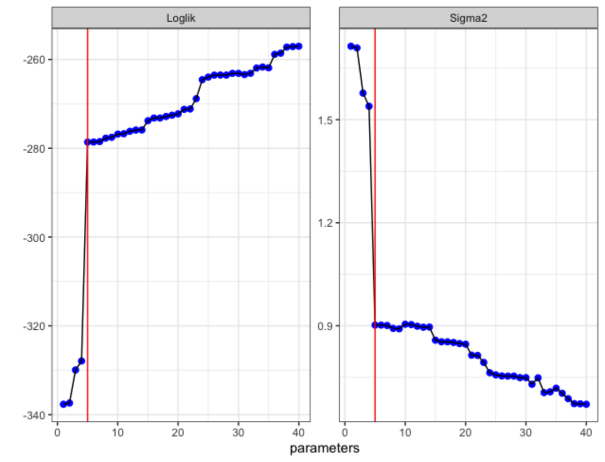
```
> mx_l=vector('list',40)
> for(j in 1:40)
+   mx_l[[j]]<-sarima(x,p = 0,d = 0,q = j,
+   P = 0,D = 0,Q = 0,S = 0,
+   details = F,no.constant = T)
```

Loglik/Variance

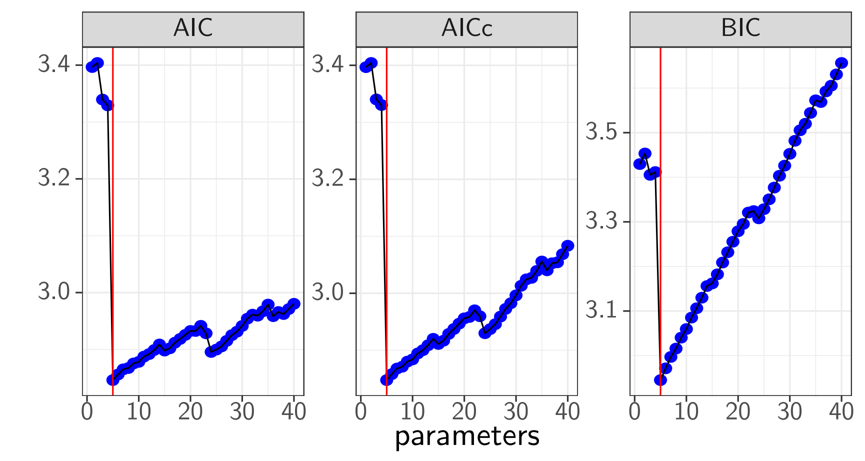
```
> df=plyr::ldply(lapply(mx_l, function(x) c(x$fit$sigma2,x$fit$loglik)))
>
> df=cbind.data.frame(p=1:40,df)
> colnames(df)[2:3]=c("Sigma2","Loglik")
>
> library(tidyr)
>
> df=df%>%pivot_longer(cols = 2:3)
>
> ggplot(df,aes(x=p,value))+geom_point(size=2,col="blue")+geom_line()+
+    theme_bw()+geom_vline(xintercept = 5,col="red")+
+   facet_wrap(~name,ncol = 2,scales = "free_y")+
+   ylab("")+xlab("parameters")
```

Loglik/Variance

```
> df=pl                                                    loglik)))
>
> df=cl
> colna
>
> libra
>
> df=di
>
> ggplo                                                    line()+
+     tl
+    fac
+    yla
```



AIC/AICc/BIC



Testing the parameters

- Hypothesis testing

$$H_0 \ \theta = 0 \text{ vs } H_0 \ \theta = 0$$

- Statistics of the test: If $\widehat{\theta}$ is the estimator of θ, then under H_0

$$T = \frac{\widehat{\theta}}{S_{\widehat{\theta}}} \sim \mathcal{T}(\text{dof})$$

and

$$T^2 = \left(\frac{\widehat{\theta}}{S_{\widehat{\theta}}}\right)^2 \sim F(1, \text{dof}) \quad \text{F-statistics}$$

where $S_{\widehat{\theta}}$ is the sample variance of $\widehat{\theta}$ and dof is the degree of freedom of the model.

Testing the parameters

```
> fit$fit$coef
      ma1       sma1
 0.2902981 -0.6912543
> fit$fit$var.coef
              ma1         sma1
ma1   0.012500671 -0.002505474
sma1 -0.002505474  0.014242590
> sqrt(diag(fit$fit$var.coef))
      ma1      sma1
0.1118064 0.1193423
> fit$fit$coef/sqrt(diag(fit$fit$var.coef))
      ma1      sma1
 2.596436 -5.792197
> fit$ttable
     Estimate     SE t.value p.value
ma1    0.2903 0.1118  2.5964   0.012
sma1  -0.6913 0.1193 -5.7922   0.000
```

Residual Analysis

- Standardized Residuals should be a Gaussiam WN.

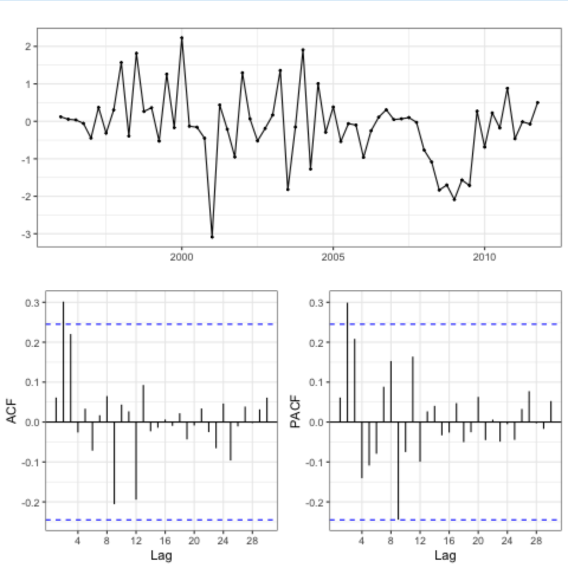
$$\widehat{r_S} = \frac{\widehat{r}}{\sqrt{S_r}}$$

- TS, ACF and PACF Charts.
- Box-Ljung Test, all p-values greater than a fixed threshold 5%
- QQnorm Chart to check gaussianity

Residual Analysis, TS, ACF and PACF

```
> stdres <- residuals/sqrt(fit$fit$sigma2)
> ggtsdisplay(stdres,lag.max=30,theme = theme_bw())
```

Resid



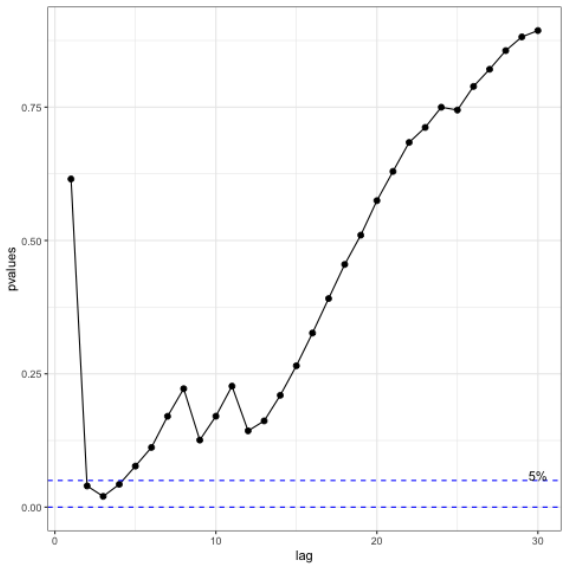

```
> stdre
> ggtsd
```

Residual Analysis, Box-Ljung Test

```
> dt=data.frame(lag=1:30,
+               pvalues=sapply(1:30,
+                              function(i) Box.test(stdres,lag = i,
+                              type = "Ljung-Box")$p.value))
> library(ggplot2)
> p<-ggplot(dt,aes(x=lag,y=pvalues))+
+   geom_line()+geom_point(size=2)+
+   geom_hline(yintercept = 0.05,linetype = "dashed",col="blue")+
+   geom_hline(yintercept = 0.0,linetype = "dashed",col="blue")+
+   theme_bw()
> p+annotate(geom = "text",x = 30,y = 0.06,label="5%")
```

Resid

```
> dt=da
+                                                                                    i,
+
+
> libra
> p<-gg
+   geo
+   geo                                                                              )+
+   geo                                                                              +
+   the
> p+ann
```

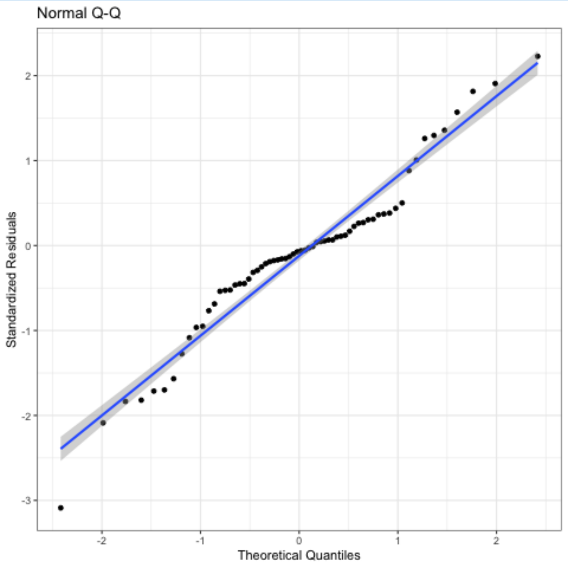


Residual Analysis, QQnorm

```
> dt2=data.frame(qqn=qqnorm(stdres,plot.it = F)[[1]],
+                stdres=stdres)
> p<-ggplot(dt2, aes(qqn, stdres))+
+   geom_point(na.rm = TRUE)
> p<-p+geom_smooth(method="lm")+
+   xlab("Theoretical Quantiles")+
+   ylab("Standardized Residuals")
> p<-p+ggtitle("Normal Q-Q")+theme_bw()
> p
```

Resid

```
> dt2=
+
> p<-g
+   ge
> p<-p
+   xl
+   yl
> p<-p
> p
```



Example: Fatalities in car accidents in France

```
> library(astsa)
> fit=sarima(xdata =m30,p = 2,d = 0,q = 3,P = 3,D = 1,Q = 1,S = 12,details =F,tol=1e-3)
> fit$fit

Call:
stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
    Q), period = S), xreg = constant, transform.pars = trans, fixed = fixed,
    optim.control = list(trace = trc, REPORT = 1, reltol = tol))

Coefficients:
         ar1     ar2     ma1     ma2     ma3     sar1     sar2     sar3     sma1  constant
      0.1804  0.1621  0.1963  0.0905  0.1760  -0.0982  -0.0476  -0.2380  -0.5380   -2.1257
s.e.  0.4734  0.1748  0.4658  0.2751  0.1551   0.1218   0.0866   0.0574   0.1427    0.2269

sigma^2 estimated as 4816:  log likelihood = -2285.03,  aic = 4592.06
```

Example: Fatalities in car accidents in France

The estimated model is then

$$(1 - B^{12})X_t = -2.1257 + \frac{(1 + 0.196B + 0.09B^2)(1 - 0.538B^{12})}{(1 - 0.18B - 0.16B^2)(1 + 0.098B^{12} + 0.048B^{24} + 0.238B^{36})} Z_t$$

Example: Fatalities in car accidents in France, t-test

```
> fit$ttable
         Estimate     SE t.value p.value
ar1        0.1804 0.4734  0.3811  0.7034
ar2        0.1621 0.1748  0.9273  0.3543
ma1        0.1963 0.4658  0.4214  0.6737
ma2        0.0905 0.2751  0.3290  0.7423
ma3        0.1760 0.1551  1.1350  0.2571
sar1      -0.0982 0.1218 -0.8060  0.4207
sar2      -0.0476 0.0866 -0.5499  0.5827
sar3      -0.2380 0.0574 -4.1492  0.0000
sma1      -0.5380 0.1427 -3.7713  0.0002
constant  -2.1257 0.2269 -9.3690  0.0000
```

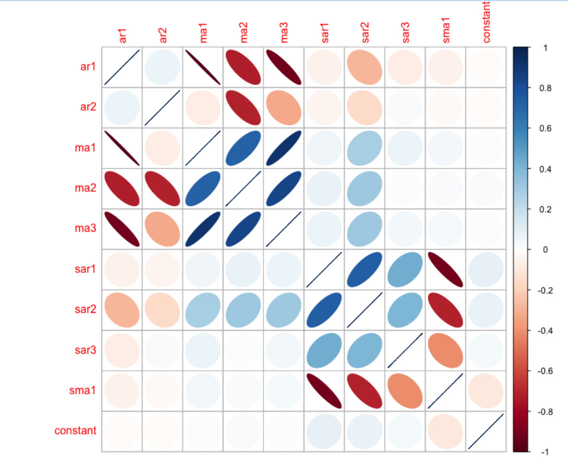
We check the correlation between the estimators of the parameters.

Example: Fatalities in car accidents in France, Correlations

```
> M=cov2cor(fit$fit$var.coef)
> library(corrplot)
> corrplot(corr = M,method = "ellipse")
```

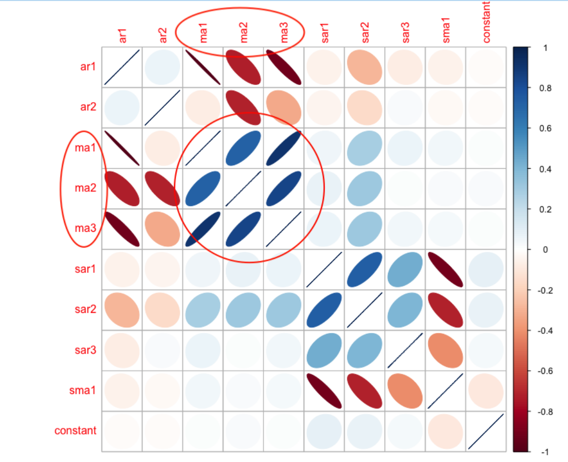
Example: Fatalities in car accidents in France, Correlations

```
> M=cov
> libra
> corrp
```



Example: Fatalities in car accidents in France, Correlations

```
> M=cov
> libra
> corrp
```



Example: Fatalities in car accidents in France, Dropping `ma3`

We try the model SARIMA$(2,0,2)(3,1,1)_{12}$, $q=2$. The `tol` argument is changed to obtain convergence

```
> fit1=sarima(xdata =m30,p = 2,d = 0,q = 2,P = 3,D = 1,Q = 1,S = 12,details =F,tol=1e-4)
> fit1$fit
Call:
stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
    Q), period = S), xreg = constant, transform.pars = trans, fixed = fixed,
    optim.control = list(trace = trc, REPORT = 1, reltol = tol))

Coefficients:
         ar1      ar2      ma1     ma2     sar1     sar2     sar3     sma1  constant
      0.7150  -0.0643  -0.4317  0.2484  -0.0809  -0.0734  -0.2222  -0.6235   -2.0411
s.e.  0.0064   0.0021   0.0076  0.0021   0.0022   0.0038   0.0035   0.0064    0.1967

sigma^2 estimated as 4786:  log likelihood = -2284.87,  aic = 4589.73
```

Example: Fatalities in car accidents in France,t-test

```
> fit1$ttable
         Estimate     SE  t.value p.value
ar1        0.7150 0.0064 112.3995       0
ar2       -0.0643 0.0021 -30.4748       0
ma1       -0.4317 0.0076 -56.9534       0
ma2        0.2484 0.0021 118.4558       0
sar1      -0.0809 0.0022 -37.2114       0
sar2      -0.0734 0.0038 -19.4437       0
sar3      -0.2222 0.0035 -62.8259       0
sma1      -0.6235 0.0064 -97.4522       0
constant  -2.0411 0.1967 -10.3787       0
> fit1$AIC
[1] 11.08631
> fit$AIC
[1] 11.09194
```

Unit root test

Augmented Dickey-Fuller Test

- This is a test where the null hypothesis H_0 claims that the TS is non-stationary vs the alternative hypothesis H_1 the TS is stationary.
- Assume that (X_t) can be written as an AR(p) (non necessarily stationary) with a linear trend:

$$X_t = \beta_1 + \beta_2 t + \varphi_1 X_{t-1} + \ldots + \varphi_p X_{t-p} + Z_t$$

where β_1 is the drift, β_2 represents the trend and (Z_t) is a stationary error.

Augmented Dickey-Fuller Test

$$\begin{aligned} X_t &= \beta_1 + \beta_2 t + (\varphi_1 + \ldots + \varphi_p)X_{t-1} \\ &\quad + \varphi_2(X_{t-2} - X_{t-1}) + \ldots + \varphi_p(X_{t-p} - X_{t-1}) + Z_t \\ &= \beta_1 + \beta_2 t + (\varphi_1 + \ldots + \varphi_p)X_{t-1} \\ &\quad \varphi_2(B^2 - B)X_t + \ldots + \varphi_p(B^p - B)X_t + Z_t \end{aligned}$$

Where

$$\varphi_2(B^2 - B)X_t + \ldots + \varphi_p(B^p - B)X_t = B(1 - B)P(B)X_t$$

where P is a polynomial with degree $p - 2$:

$$P(z) = \kappa_1 + \kappa_2 z + \ldots + \kappa_{p-1} z^{p-2}$$

Augmented Dickey-Fuller Test

Then

$$\begin{aligned} B(1-B)P(B)X_t &= \Delta\left(\kappa_1 X_{t-1} + \kappa_2 B X_{t-1} + \ldots + \kappa_{p-1} B^{p-2} X_{t-1}\right) \\ &= \kappa_1 \Delta X_{t-1} + \kappa_2 \Delta B X_{t-1} + \ldots + \kappa_{p-1} \Delta B^{p-2} X_{t-1} \\ &= \kappa_1 \Delta X_{t-1} + \kappa_2 \Delta X_{t-2} + \ldots + \kappa_{p-1} \Delta X_{t-p+1} \end{aligned}$$

$\Rightarrow$

$$\begin{aligned} \Delta X_t &= \beta_1 + \beta_2 t + \pi X_{t-1} \\ &\quad + \kappa_1 \Delta X_{t-1} + \kappa_2 \Delta X_{t-2} + \ldots + \kappa_{p-1} \Delta X_{t-p+1} + Z_t \end{aligned}$$

where $\pi = (\varphi_1 + \ldots + \varphi_p) - 1 = \Phi(1)$.

Augmented Dickey-Fuller Test, In summary

$$
\begin{aligned}
\Delta X_t &= \underbrace{\beta_1}_{\texttt{drift}} + \underbrace{\beta_2}_{\texttt{trend}} t + \pi \underbrace{X_{t-1}}_{\texttt{lag.1}} \\
&\quad + \kappa_1 \Delta X_{t-1} + \kappa_2 \Delta X_{t-2} + \ldots + \kappa_{p-1} \Delta X_{t-p+1} + Z_t
\end{aligned}
$$

$$\iff$$

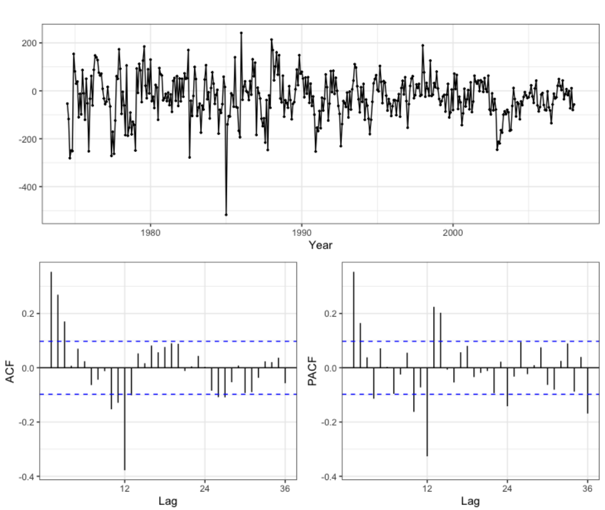
$$
\begin{aligned}
(1-(1+\pi)B)\, X_t &= \beta_1 + \beta_2 t + \\
&\quad \underbrace{\kappa_1 \Delta X_{t-1} + \kappa_2 \Delta X_{t-2} + \ldots + \kappa_{p-1} \Delta X_{t-p+1}}_{\texttt{diff.lags}} + Z_t
\end{aligned}
$$

$$\Phi(z) = 1-(1+\pi)z,\ \Phi(z) = 0 \iff z = \frac{1}{1+\pi}$$

Augmented Dickey-Fuller Test,

- `ur.df` from `urca` package
- $H_0 : \pi = 0$ vs $H_1 : \pi < 0$, `tau1` statistics, `type="none"`
- $H_0 : (\pi, \beta_1) = (0, 0)$ vs $H_1 : \pi < 0$, `tau2` and `phi1` statistics, `type="drift"`
- $H_0 : (\pi, \beta_1, \beta_2) = (0, 0, 0)$ vs $H_1 : \pi < 0$, `tau3`, `phi1` and `phi2` statistics, `type="trend"`

Example: Fatalities in car accidents in France, $\Delta_{12}X_t = X_t - X_{t-12}$



Example: Fatalities in car accidents in France, type="none"

```
> t1_m30<-m30%>%diff(lag=12)%>%ur.df(type = "none",lags = 3)
> summary(t1_m30)

###############################################
# Augmented Dickey-Fuller Test Unit Root Test #
###############################################

Test regression none


Call:
lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)

Residuals:
    Min      1Q  Median      3Q     Max
-536.19  -58.50  -12.56   36.79  331.33
```

./..

Example: Fatalities in car accidents in France, type="none"

```
Residuals:
    Min      1Q  Median      3Q     Max
-536.19  -58.50  -12.56   36.79  331.33
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
z.lag.1     -0.508782   0.062369  -8.158  4.6e-15 ***
z.diff.lag1 -0.192623   0.063966  -3.011  0.00277 **
z.diff.lag2 -0.004585   0.059900  -0.077  0.93903
z.diff.lag3  0.088259   0.049473   1.784  0.07519 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 84.22 on 395 degrees of freedom
Multiple R-squared:  0.347,        Adjusted R-squared:  0.3403
F-statistic: 52.46 on 4 and 395 DF,  p-value: < 2.2e-16

Value of test-statistic is: -8.1576

Critical values for test statistics:
      1pct  5pct 10pct
tau1 -2.58 -1.95 -1.62
```

Example: Fatalities in car accidents in France, type="none"

```
Residuals:
    Min      1Q  Median      3Q     Max
-536.19  -58.50  -12.56   36.79  331.33
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
z.lag.1     -0.508782   0.062369  -8.158  4.6e-15 ***
z.diff.lag1 -0.192623   0.063966  -3.011  0.00277 **
z.diff.lag2 -0.004585   0.059900  -0.077  0.93903
z.diff.lag3  0.088259   0.049473   1.784  0.07519 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 84.22 on 395 degrees of freedom
Multiple R-squared:  0.347,        Adjusted R-squared:  0.3403
F-statistic: 52.46 on 4 and 395 DF,  p-value: < 2.2e-16

Value of test-statistic is: -8.1576

Critical values for test statistics:
      1pct  5pct 10pct
tau1 -2.58 -1.95 -1.62
```

Example: Fatalities in car accidents in France, `type="none"`

- H_0 is then rejected at a level 5%
- $W_t = \Delta_{12} X_t$
- The model suggested is then

$$\Delta W_t = -0.508782 \times W_{t-1} - 0.192623 \times \Delta W_{t-1} + Z_t$$

  where $W_t = \Delta_{12} X_t$
- Conclusion: Stationarity of $\Delta_{12} X_t$

Example: Fatalities in car accidents in France, type="none", Selecting lags with AIC

```
> t1a_m30<-m30%>%diff(lag=12)%>%ur.df(type = "none",selectlags = 'AIC')
> summary(t1a_m30)
Call:
lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
..
Coefficients:
           Estimate Std. Error t value Pr(>|t|)
z.lag.1    -0.48127    0.05358  -8.982  < 2e-16 ***
z.diff.lag -0.19459    0.04906  -3.967 8.64e-05 ***

Residual standard error: 85.4 on 399 degrees of freedom
Multiple R-squared:  0.326,        Adjusted R-squared:  0.3226
F-statistic: 96.51 on 2 and 399 DF,  p-value: < 2.2e-16

Value of test-statistic is: -8.9819

Critical values for test statistics:
      1pct  5pct 10pct
tau1 -2.58 -1.95 -1.62
```

Example: Fatalities in car accidents in France, type="drift"

```
> t2_m30<-m30%>%diff(lag=12)%>%ur.df(type = "drift",selectlags = 'AIC')
> summary(t2_m30)

###############################################
# Augmented Dickey-Fuller Test Unit Root Test #
###############################################

Test regression drift


Call:
lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)

Residuals:
    Min      1Q  Median      3Q     Max
-517.74  -44.61   -1.21   49.48  340.30
```

./..

Example: Fatalities in car accidents in France, `type="drift"`

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -14.10522    4.46984  -3.156 0.001723 **
z.lag.1      -0.53994    0.05616  -9.615  < 2e-16 ***
z.diff.lag   -0.16532    0.04940  -3.347 0.000895 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

./..

Example: Fatalities in car accidents in France, type="drift"

```
Residual standard error: 84.46 on 398 degrees of freedom
Multiple R-squared:  0.3425,       Adjusted R-squared:  0.3392
F-statistic: 103.7 on 2 and 398 DF,  p-value: < 2.2e-16


Value of test-statistic is: -9.6147 46.2221

Critical values for test statistics:
      1pct  5pct 10pct
tau2 -3.44 -2.87 -2.57
phi1  6.47  4.61  3.79
```

Example: Fatalities in car accidents in France, `type="none"`

- H_0 is then rejected at a level 5%
- The model suggested is then

$$\Delta W_t = -14.10522 - 0.53994 \times W_{t-1} - 0.16532 \times \Delta W_{t-1} + Z_t$$

  where $W_t = \Delta_{12} X_t$

- Conclusion: Stationarity of $\Delta_{12} X_t$ with a drift

Example with `adf.test` from `tseries` package

- Computes the Augmented Dickey-Fuller test for the null that (X_t) has a unit root.
- By Default, the number of lags used in the regression is $k = \text{floor}((T-1)^{1/3})$, where T is the length of the TS.

```
> library(tseries)
> t1a=adf.test(W,alternative = "stationary")
> t1a
        Augmented Dickey-Fuller Test
data:  W
Dickey-Fuller = -6.8304, Lag order = 7, p-value = 0.01
alternative hypothesis: stationary
```

Test on white noise

```
> TT <- 100
> wn <- rnorm(TT)
> adf.test(wn,alternative = "stationary")

        Augmented Dickey-Fuller Test

data:  wn
Dickey-Fuller = -4.8612, Lag order = 4, p-value = 0.01
alternative hypothesis: stationary
```

Test on white noise with trend, with adf.test

$$X_t = 2 - 0.4t + Z_t, \quad (Z_t) \text{ is a WN}$$

```
> TT <- 100
> intercept<-2
> wn <- rnorm(TT)
> wnt<- intercept-.4* 1:TT+wn
> adf.test(wnt,alternative = "stationary")

        Augmented Dickey-Fuller Test

data:  wnt
Dickey-Fuller = -4.2069, Lag order = 4, p-value = 0.01
alternative hypothesis: stationary
```

Test on white noise with trend, with `ur.df`

```
> m1<-ur.df(wnt,type = "trend")
> m1@testreg
Call:
lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)
Residuals:
    Min      1Q  Median      3Q     Max
-2.4538 -0.6446  0.0107  0.6361  2.6039

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.83343    0.39993   4.584 1.40e-05 ***
z.lag.1     -1.03592    0.14345  -7.222 1.32e-10 ***
tt          -0.41829    0.05810  -7.199 1.47e-10 ***
z.diff.lag   0.05615    0.10260   0.547    0.586
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9968 on 94 degrees of freedom
Multiple R-squared:  0.4963,	Adjusted R-squared:  0.4803
F-statistic: 30.88 on 3 and 94 DF,  p-value: 5.552e-14
```

Test on white noise with trend, with `ur.df`

```
> m1@teststat
                tau3     phi2     phi3
statistic -6.542911 27.41225 21.42189
> m1@cval
      1pct  5pct 10pct
tau3 -4.04 -3.45 -3.15
phi2  6.50  4.88  4.16
phi3  8.73  6.49  5.47
```

Test on random walks with adf.test

```
> rw <- cumsum(rnorm(TT))
> adf.test(rw)

        Augmented Dickey-Fuller Test

data:  rw
Dickey-Fuller = -2.1181, Lag order = 4, p-value = 0.5277
alternative hypothesis: stationary
```

Stationarity Test, KPSS Test

- H_0: (X_t) is stationary with a trend and a non-zero mean vs H_1: (X_t) is not stationary.
- We assume in this test that

$$X_t = R_t + \beta_1 + \beta_2 t + U_t$$

where

  - (R_t) is a random walk; $R_t = R_{t-1} + Z_t$, $(Z_t) \sim$ Gaussian WN$(0, \sigma_z^2)$
  - (U_t) is a stationary error.
- `ur.kpss` from `urca` package

Example, Random walks and others

$(X_t) \sim$ i.i.d. $\mathcal{N}(0, \sigma^2)$ White Noise

(Y_t) s.t. $Y_t = \sum_{k=0}^{t} X_k$ Random Walk

(Z_t) s.t. $Z_t = 2 - 0.33t + X_t$ WN with a deterministic trend

(W_t) s.t. $W_t = Y_t + U_t,\ (U_t) \sim \text{ARMA}(1,1)$ RW with a stationary error

`ur.kpss` function

- `type="mu"` :

$$X_t = R_t + \beta_1 + U_t$$

  H_0: (X_t) is stationary with a non-zero mean vs H_1: (X_t) non stationary

- `type="tau"` :

$$X_t = R_t + \beta_1 + \beta_2 t + U_t$$

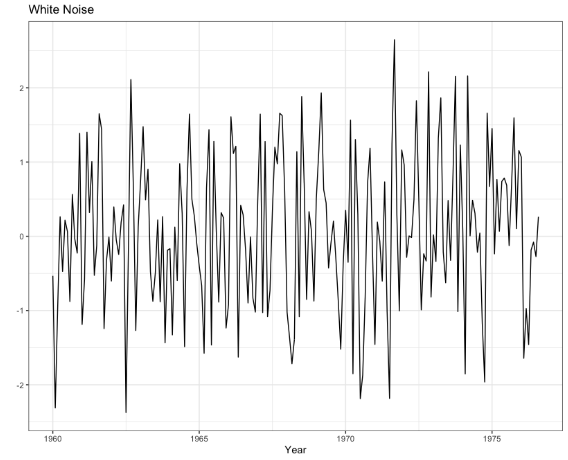
  H_0: (X_t) is stationary with a deterministic trend vs H_1: (X_t) non stationary

Example: simulating processes

```
> library(urca)
> library(sarima)
> set.seed(231)
> x=rnorm(200)
> y=cumsum(x) ## is the random walk
> z=2-.33*(0:199)+x
> w=y+sim_sarima(n=200, model = list(ma=0.8))
> x=ts(x,start=c(1960,1),frequency = 12)
> y=ts(y,start=c(1960,1),frequency = 12)
> z=ts(z,start=c(1960,1),frequency = 12)
> w=ts(w,start=c(1960,1),frequency = 12)
```

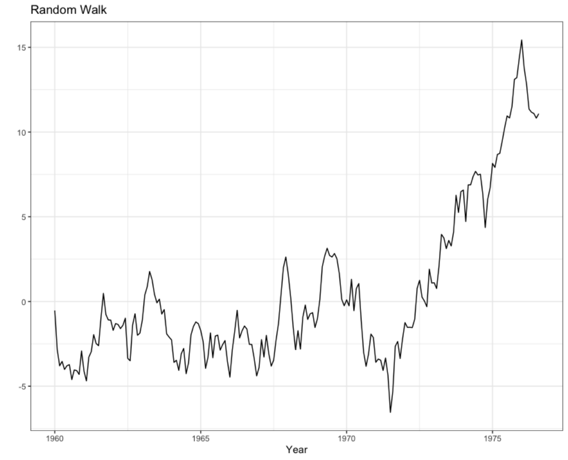
Example: simulating processes

```
> libra
> libra
> set.s
> x=rnc
> y=cur
> z=2-
> w=y+s
> x=ts
> y=ts
> z=ts
> w=ts
```



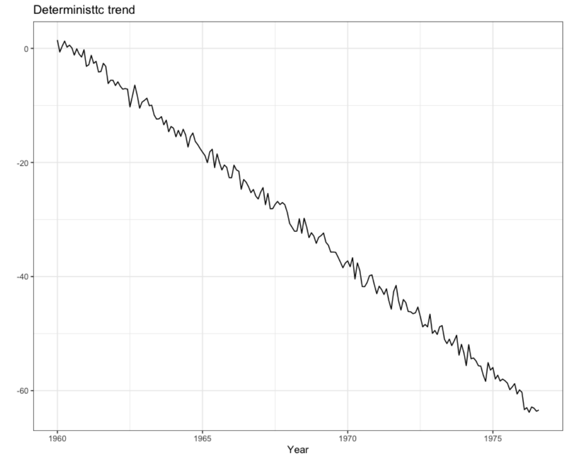
Example: simulating processes

```
> libra
> libra
> set.s
> x=rnc
> y=cur
> z=2-
> w=y+s
> x=ts
> y=ts
> z=ts
> w=ts
```



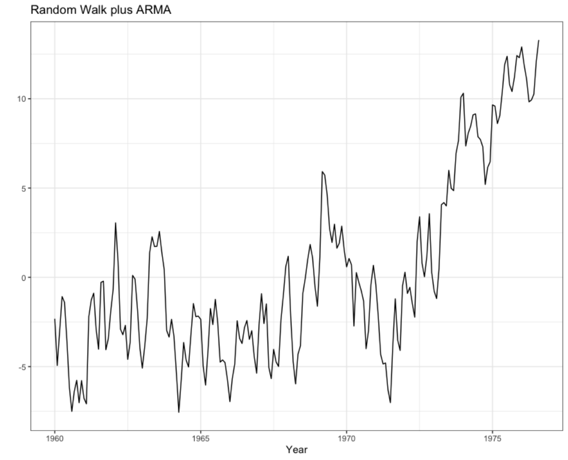
Example: simulating processes

```
> libra
> libra
> set.s
> x=rnc
> y=cur
> z=2-
> w=y+s
> x=ts
> y=ts
> z=ts
> w=ts
```



Example: simulating processes

```
> libra
> libra
> set.s
> x=rnc
> y=cur
> z=2-
> w=y+s
> x=ts
> y=ts
> z=ts
> w=ts
```



Example: testing WN, (1/2)

```
> t0=ur.kpss(x,type = "mu",use.lag = 3)
> summary(t0)

#######################
# KPSS Unit Root Test #
#######################

Test is of type: mu with 3 lags.

Value of test-statistic is: 0.1816

Critical value for a significance level of:
                10pct  5pct 2.5pct  1pct
critical values 0.347 0.463  0.574 0.739
```

Conclusion: H_0 is accepted

Example: testing WN, (2/2)

```
> t0_tau=ur.kpss(x,type = "tau",use.lag = 3)
> summary(t0_tau)

#######################
# KPSS Unit Root Test #
#######################

Test is of type: tau with 3 lags.

Value of test-statistic is: 0.0291

Critical value for a significance level of:
                10pct  5pct 2.5pct  1pct
critical values 0.119 0.146  0.176 0.216
```

Conclusion: H_0 is accepted

Example: testing RW, (1/2)

```
> t1=ur.kpss(y,type = "mu",use.lag = 3)
> summary(t1)

#######################
# KPSS Unit Root Test #
#######################

Test is of type: mu with 3 lags.

Value of test-statistic is: 2.7793

Critical value for a significance level of:
                10pct  5pct 2.5pct  1pct
critical values 0.347 0.463  0.574 0.739
```

Conclusion: H_0 is rejected

Example: testing RW, (2/2)

```
> t1_tau=ur.kpss(y,type = "tau",use.lag = 3)
> summary(t1_tau)

#######################
# KPSS Unit Root Test #
#######################

Test is of type: tau with 3 lags.

Value of test-statistic is: 0.7196

Critical value for a significance level of:
                10pct  5pct 2.5pct  1pct
critical values 0.119 0.146  0.176 0.216
```

Conclusion: H_0 is rejected

Example: testing D Trend, (1/2)

```
> t2=ur.kpss(z,type = "mu",use.lag = 3)
> summary(t2)

#######################
# KPSS Unit Root Test #
#######################

Test is of type: mu with 3 lags.

Value of test-statistic is: 5.0899

Critical value for a significance level of:
                10pct  5pct 2.5pct  1pct
critical values 0.347 0.463  0.574 0.739
```

Conclusion: H_0 is rejected

Example: testing D Trend, (2/2)

```
> t2_tau=ur.kpss(z,type = "tau",use.lag = 3)
> summary(t2_tau)

#######################
# KPSS Unit Root Test #
#######################

Test is of type: tau with 3 lags.

Value of test-statistic is: 0.0291

Critical value for a significance level of:
                10pct  5pct 2.5pct  1pct
critical values 0.119 0.146  0.176 0.216
```

Conclusion: H_0 is accepted

Example: RW + ARMA, (1/2)

```
> t3=ur.kpss(w,type = "mu",use.lag = 3)
> summary(t3)

#######################
# KPSS Unit Root Test #
#######################

Test is of type: mu with 3 lags.

Value of test-statistic is: 2.676

Critical value for a significance level of:
                10pct  5pct 2.5pct  1pct
critical values 0.347 0.463  0.574 0.739
```

Conclusion: H_0 is rejected

Example: RW + ARMA, (2/2)

```
> t3_tau=ur.kpss(w,type = "tau",use.lag = 3)
> summary(t3_tau)

#######################
# KPSS Unit Root Test #
#######################

Test is of type: tau with 3 lags.

Value of test-statistic is: 0.7088

Critical value for a significance level of:
                10pct  5pct 2.5pct  1pct
critical values 0.119 0.146  0.176 0.216
```

Conclusion: H_0 is rejected

Model Selection

Box-Cox transformation

- It aims to stabilize the variance
- Box-Cox transformation:

$$X_t^\lambda = \begin{cases} \dfrac{X_t^\lambda - 1}{\lambda} & \text{If } \lambda \neq 0 \\ \log(X_t) & \text{If } \lambda = 0 \end{cases}$$

- Guerrero Method: Estimating λ by minimising the coefficient of variation (Sample Variance/Sample Mean)
- LogLik Method: Estimating λ maximising the profile log likelihood

Example with R, Guerrero method

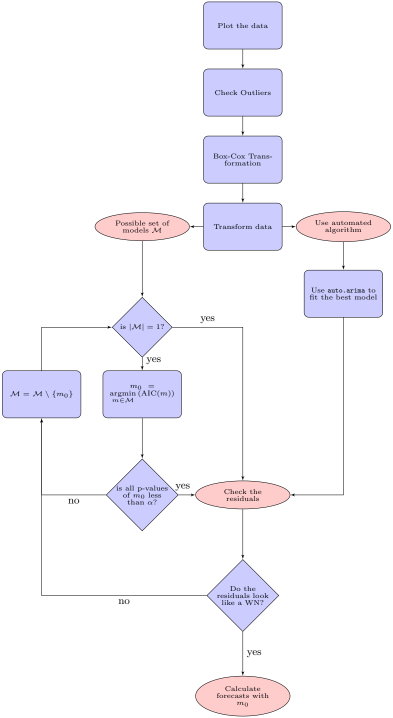
```
> library(forecast)
> library(caschrono)
> BoxCox.lambda(m30,method = "guerrero",lower = 0,upper = 10)
[1] 5.575865e-05
```

Then $\widehat{\lambda} \approx 0$, We will consider the $\log$ transformation.

Example with R, Loglik method

```
> library(forecast)
> library(caschrono)
> > BoxCox.lambda(m30,method = "loglik",lower = 0,upper = 10)
[1] 0.85
```

Then $\widehat{\lambda} \approx 0.85$, We will consider the $\dfrac{X_t^{0.85} - 1}{0.85}$ transformation.



Plot the data
Check Outliers
Box-Cox Transformation
Transform data
Possible set of models $\mathcal{M}$
Use automated algorithm
Use auto.arima to fit the best model
is $|\mathcal{M}| = 1$?
yes
yes
$\mathcal{M} = \mathcal{M} \setminus \{m_0\}$
$m_0 = \underset{m \in \mathcal{M}}{\operatorname{argmin}}(\mathrm{AIC}(m))$
is all p-values of m_0 less than α?
yes
no
Check the residuals
Do the residuals look like a WN?
no
yes
Calculate forecasts with m_0

Example m30, Step 1: checking outliers (1/2)

We use tsoutliers from forecast package

```
> library(forecast)
> library(caschrono)
> data("m30")
> x=tsoutliers(m30,lambda = NULL)
> x$index
[1]   3   5  37  62 109 139 158 163
> m30[x$index]
[1] 1616.328 1490.186 1631.294 1014.481 ....
> m30b=m30
> m30b[x$index]=x$replacements
```

Example m30, Step 1: checking outliers (2/2)

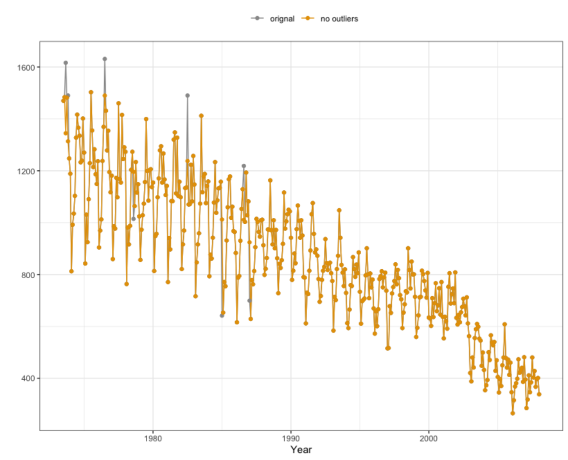
We use tsoutliers from forecast package

```
> library(zoo)
> dt=data.frame(date=as.yearmon(time(m30)),m30,m30b)
> library(reshape2)
> dt2=melt(dt,id.vars = "date")
> library(ggplot2)
> p<-ggplot(dt2,aes(x=date,y=value,col=variable))+geom_line()+geom_point(size=1.5)+theme_bw()
> p<-p+scale_color_manual(name="",values=c("#999999", "#E69F00"),
+                         breaks=c("m30","m30b"),
+                         labels=c("orignal","no outliers"))
> p<-p+theme(legend.position = "top")+xlab("Year")+ylab("")
> p
```

Example m30, Step 1: checking outliers (2/2)

We use

```
> librar
> dt=dat
> librar
> dt2=me
> librar
> p<-ggp                                                    5)+theme_bw()
> p<-p+s
+
+
> p<-p+t
> p
```



1st Method: `auto.arima`

Example m30, LogLik M.

```
> lam1=BoxCox.lambda(m30b,method = "loglik",lower = 0,upper = 10)
> lam1
[1] 0.85
> lm30A=(m30b^lam1-1)/lam1
> lmA=auto.arima(lm30A)
> lmA
Series: lm30A
ARIMA(1,0,1)(2,1,1)[12] with drift

Coefficients:
         ar1      ma1     sar1    sar2     sma1    drift
      0.8409  -0.4973  -0.0129  0.0289  -0.7615  -1.4883
s.e.  0.0521   0.0837   0.0717  0.0635   0.0473   0.1510

sigma^2 estimated as 1903:  log likelihood=-2095.61
AIC=4205.22   AICc=4205.5   BIC=4233.21
```

Example m30, Box-Cox Transformation, Guerrero M.

```
> lam2=BoxCox.lambda(m30b,method = "guerrero",lower = 0,upper = 10)
> lam2
[1] 5.575865e-05
> lm30B=log(m30b)
> lmB=auto.arima(lm30B)
> lmB
Series: lm30B
ARIMA(1,1,1)(1,1,2)[12]

Coefficients:
         ar1      ma1     sar1     sma1     sma2
      0.2151  -0.8232  -0.6055  -0.2291  -0.5061
s.e.  0.0732   0.0472      NaN      NaN      NaN

sigma^2 estimated as 0.005183:  log likelihood=482.26
AIC=-952.52   AICc=-952.31   BIC=-928.55
```

Example `m30`, Conclusions

- Remove the outliers
- Adopt the Box-Cox transformation with LogLik method

  Then

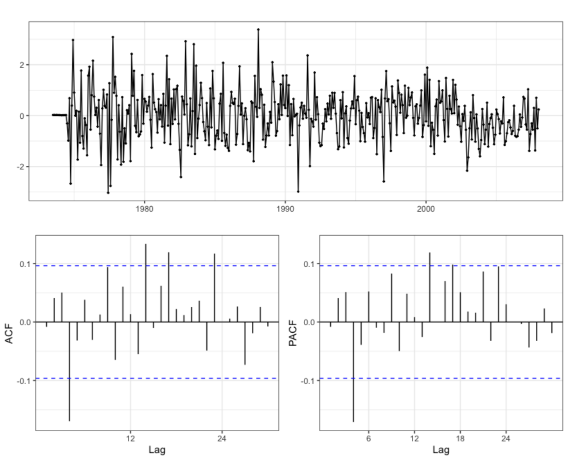
- Residual Analysis of the model `lm30A`
- Estimate d and D, search the possible set of models and estimate the best model.

Example m30, Residual analysis of lmA

```
> residuals=lmA$residuals
> stdres <- residuals/sqrt(lmA$sigma2)
> ggtsdisplay(stdres,lag.max=30,theme = theme_bw())
```

Example m30, Residual analysis of lmA

> resid
> stdre
> ggtsd



Example m30, Stationarity of the Residuals

```
> library(urca)
> t1<-ur.df(residuals,type = "none")
> summary(t1)

###############################################
# Augmented Dickey-Fuller Test Unit Root Test #
###############################################

Test regression none


Call:
lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)

Residuals:
    Min      1Q  Median      3Q     Max
-132.54  -29.52    0.94   25.57  146.93

Coefficients:
           Estimate Std. Error t value Pr(>|t|)
z.lag.1    -0.96706    0.06999 -13.817   <2e-16 ***
z.diff.lag -0.04084    0.04930  -0.828    0.408
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 42.84 on 411 degrees of freedom
Multiple R-squared:  0.5049,       Adjusted R-squared:  0.5025
F-statistic: 209.6 on 2 and 411 DF,  p-value: < 2.2e-16


Value of test-statistic is: -13.8169

Critical values for test statistics:
      1pct  5pct 10pct
tau1 -2.58 -1.95 -1.62
```

Example `m30`, Residuals are WN, method 1

`armaselect` returns the best ARMA models, with respect to the Schwarz's Bayesian Criterion (sbc).

```
> armaselect(stdres,nbmod = 10)
      p q        sbc
 [1,] 0 0 -17.423145
 [2,] 0 1 -13.905188
 [3,] 1 0 -10.419911
 [4,] 1 1  -7.933959
 [5,] 0 2  -7.644189
 [6,] 0 4  -7.277844
 [7,] 4 1  -4.694414
 [8,] 2 0  -4.076065
 [9,] 4 0  -3.511963
[10,] 2 1  -3.131837
```

Example m30, Residuals are WN, method 1

armaselect returns the best ARMA models, with respect to the Schwarz's Bayesian Criterion (sbc).

```
> armaselect(residuals,nbmod = 10)
      p q      sbc
 [1,] 0 0 3116.309
 [2,] 0 1 3119.827
 [3,] 1 0 3123.313
 [4,] 1 1 3125.799
 [5,] 0 2 3126.088
 [6,] 0 4 3126.455
 [7,] 4 1 3129.038
 [8,] 2 0 3129.657
 [9,] 4 0 3130.221
[10,] 2 1 3130.601
```

Example m30, Residuals are WN, method 2

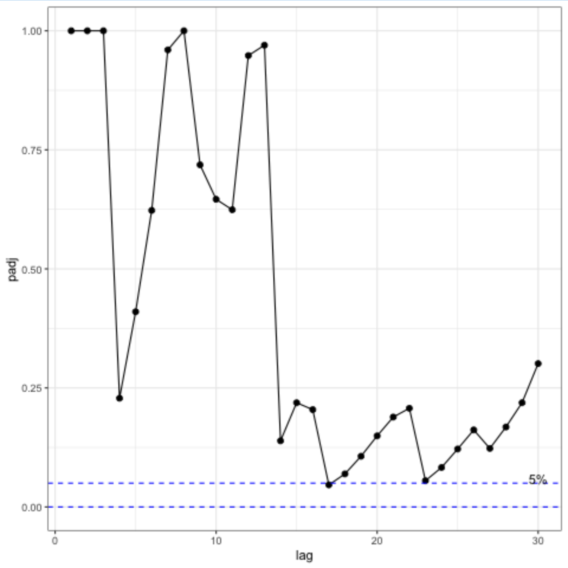
Box-Pierce and Ljung-Box Tests + A Bonferoni adjustment of the p-values

```
> dt=data.frame(lag=1:30,
+               pvalues=sapply(1:30,
+                              function(i)
+                                Box.test(stdres,lag = i,type = "Ljung-Box")$p.value))
> dt$padj=p.adjust(dt$pvalues,
+                  method = "bonferroni")
> library(ggplot2)
> p<-ggplot(dt,aes(x=lag,y=padj))+
+   geom_line()+geom_point(size=2)+
+   geom_hline(yintercept = 0.05,linetype = "dashed",col="blue")+
+   geom_hline(yintercept = 0.0,linetype = "dashed",col="blue")+
+   theme_bw()
> p+annotate(geom = "text",x = 30,y = 0.06,label="5%")
```

Exam

Box-Pi e
p-value

```
> dt=dat
+
+
+                                      alue))
> dt$pad
+
> librar
> p<-ggp
+   geom
+   geom
+   geom
+   them
> p+anno
```



Selected Model, 1st Method

```
> lmA
Series: lm30A
ARIMA(1,0,1)(2,1,1)[12] with drift

Coefficients:
         ar1      ma1     sar1    sar2     sma1    drift
      0.8409  -0.4973  -0.0129  0.0289  -0.7615  -1.4883
s.e.  0.0521   0.0837   0.0717  0.0635   0.0473   0.1510

sigma^2 estimated as 1903:  log likelihood=-2095.61
AIC=4205.22   AICc=4205.5   BIC=4233.21
> t_stat(lmA)
            ar1       ma1      sar1     sar2    sma1     drift
t.stat 16.13357 -5.939793 -0.179747 0.455423 -16.107 -9.853625
p.val   0.00000  0.000000  0.857351 0.648805   0.000  0.000000
```

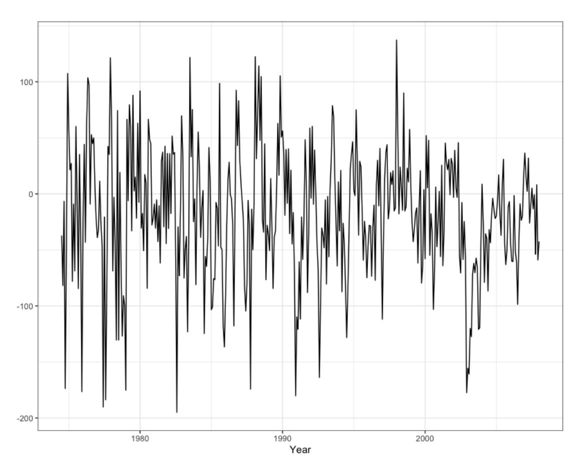
2nd Method: Searching in a set of models

Example `m30`, Differentiation of (X_t)

```
> lm30A%>%diff(lag=12)%>%autoplot()+theme_bw()+xlab("Year")+ylab("")
```

Example `m30`, Differentiation of (X_t)

```
> lm30A
```



Example m30, Dickey-Fuller test, no drift, no trend

```
> t1=lm30A%>%diff(lag=12)%>%ur.df(type="none")
> summary(t1)

###############################################
# Augmented Dickey-Fuller Test Unit Root Test #
###############################################

Test regression none


Call:
lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)

Residuals:
     Min       1Q   Median       3Q      Max
-182.127  -39.399   -9.771   28.044  159.382

Coefficients:
           Estimate Std. Error t value Pr(>|t|)
z.lag.1    -0.47473    0.05300  -8.957  < 2e-16 ***
z.diff.lag -0.18496    0.04915  -3.763 0.000193 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 54.98 on 399 degrees of freedom
Multiple R-squared:  0.3163,	Adjusted R-squared:  0.3128
F-statistic: 92.28 on 2 and 399 DF,  p-value: < 2.2e-16


Value of test-statistic is: -8.9575

Critical values for test statistics:
      1pct  5pct 10pct
tau1 -2.58 -1.95 -1.62
```

Example m30, Dickey-Fuller test, with drift, no trend

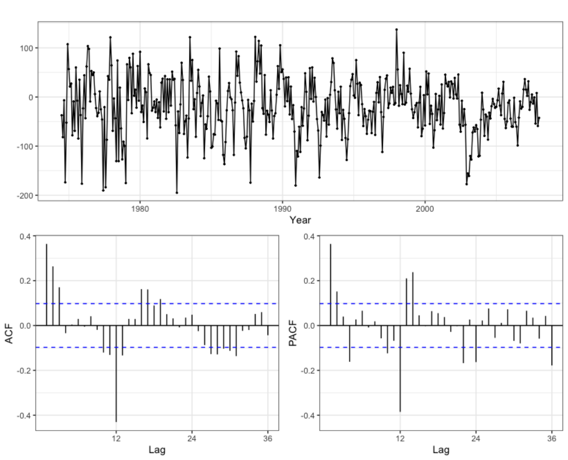
```
> t2=lm30A%>%diff(lag=12)%>%ur.df(type="drift")
> summary(t2)

###############################################
# Augmented Dickey-Fuller Test Unit Root Test #
###############################################

Test regression drift


Call:
lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)

Residuals:
    Min      1Q  Median      3Q     Max
-173.12  -31.69   -1.00   35.66  163.63

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -9.69243    2.89306  -3.350 0.000885 ***
z.lag.1     -0.54006    0.05585  -9.670  < 2e-16 ***
z.diff.lag  -0.15238    0.04950  -3.079 0.002224 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 54.29 on 398 degrees of freedom
Multiple R-squared:  0.335,	Adjusted R-squared:  0.3317
F-statistic: 100.3 on 2 and 398 DF,  p-value: < 2.2e-16


Value of test-statistic is: -9.6703 46.7581

Critical values for test statistics:
      1pct  5pct 10pct
tau2 -3.44 -2.87 -2.57
phi1  6.47  4.61  3.79
```

Example m30, Determining the orders

```
> lm30A%>%diff(lag=12)%>%ggtsdisplay(theme = theme_bw(),
+                                    main="",
+                                    xlab="Year")
```

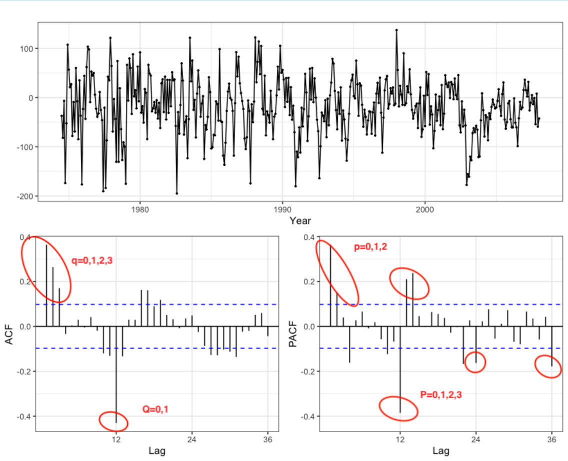
Example m30, Determining the orders

```
> lm30A
+
+
```



Example m30, Determining the orders

```
> lm30A
+
+
```



Example `m30`, Determining the orders

- $p \in \{0, 1, 2, \}$
- $q \in \{0, 1, 2, 3\}$
- $P \in \{0, 1, 2, 3\}$
- $Q \in \{0, 1\}$
- $T = 12,\ D = 1,\ d = 0$
- with a drift

Example m30, Determining the orders

```
> orders=list(p=c(0,1,2),d=0,q=c(0,1,2,3),
+             P=c(0,1,2,3),D=1,Q=c(0,1),
+             T=12)
> all_orders=expand.grid(orders$p,orders$d,orders$q,
+                        orders$P,orders$D,orders$Q,
+                        orders$T)
> colnames(all_orders)=c("p","d","q","P","D","Q","T")
> dim(all_orders)
[1] 96  7
> head(all_orders,n = 3)
  p d q P D Q  T
1 0 0 0 0 1 0 12
2 1 0 0 0 1 0 12
3 2 0 0 0 1 0 12
```

Example m30, Estimating the models

```
> models=vector('list',nrow(all_orders))
> all_orders=as.matrix(all_orders)
> for(i in 1:nrow(all_orders)){
+   print(i)
+  od=all_orders[i,1:3]
+  od_s=all_orders[i,4:6]
+  TT=all_orders[i,7]
+ models[[i]]=try(Arima(lm30A,order=od,
+        seasonal=list(order=od_s,period=TT),
+        lambda=NULL,
+        include.drift = T))
+ }
```

Example m30, Extracting the AIC

```
> aic=rep(NA,nrow(all_orders))
>  for(i in 1:nrow(all_orders)){
+      aic[i]=try(as.numeric(models[[i]]$aic))
+  }
> aic=as.numeric(aic)
Warning message:
NAs introduced by coercion
> mod_names=unlist(lapply(models,function(x)as.character(x)))
> i=grep(pattern = "ARIMA",x = mod_names)
> mod_names[-i]=NA
> dt=data.frame(model=mod_names,aic=aic)
> i=order(dt$aic,decreasing = F)[1:5]
> dt[i,]
                                model      aic
89 ARIMA(1,0,1)(3,1,1)[12] with drift 4199.128
94 ARIMA(0,0,3)(3,1,1)[12] with drift 4199.557
92 ARIMA(1,0,2)(3,1,1)[12] with drift 4200.498
90 ARIMA(2,0,1)(3,1,1)[12] with drift 4200.767
95 ARIMA(1,0,3)(3,1,1)[12] with drift 4200.814
```

Example m30, Extracting models with significant coefficients

```
> x=rep(NA,nrow(all_orders))
> for(i in 1:nrow(all_orders)){
+   x[i]=try(prod(t_stat(models[[i]])[2,]<=0.05))
+ }
> x=as.numeric(x)
> xtabs(~x)
x
 0  1
52 38
> dt=dt[which(x==1),]
> i=order(dt$aic,decreasing = F)[1:5]
> dt[i,]
                                model      aic
53 ARIMA(1,0,1)(0,1,1)[12] with drift 4201.598
58 ARIMA(0,0,3)(0,1,1)[12] with drift 4203.445
51 ARIMA(2,0,0)(0,1,1)[12] with drift 4204.721
46 ARIMA(0,0,3)(3,1,0)[12] with drift 4223.161
41 ARIMA(1,0,1)(3,1,0)[12] with drift 4226.969
```

Example m30, Residual Analysis

```
> j=as.numeric(rownames(dt))
> x=rep(NA,nrow(dt))
> for(i in 1:nrow(dt)){
+   m=models[[j[i]]]
+   tt=armaselect(m$residuals,nbmod = 4)
+   x[i]=sum(rowSums(tt[,1:2])==0)
+ }
> dt=dt[which(x==1),]
> i=order(dt$aic,decreasing = F)[1:5]
> dt[i,]
                                 model      aic
53 ARIMA(1,0,1)(0,1,1)[12] with drift 4201.598
58 ARIMA(0,0,3)(0,1,1)[12] with drift 4203.445
51 ARIMA(2,0,0)(0,1,1)[12] with drift 4204.721
46 ARIMA(0,0,3)(3,1,0)[12] with drift 4223.161
41 ARIMA(1,0,1)(3,1,0)[12] with drift 4226.969
```

Example m30,Selected Model, 2nd Method

```
> models[[53]]
Series: lm30A
ARIMA(1,0,1)(0,1,1)[12] with drift

Coefficients:
         ar1      ma1     sma1    drift
      0.8386  -0.4932  -0.7589  -1.4884
s.e.  0.0502   0.0819   0.0321   0.1493

sigma^2 estimated as 1895:  log likelihood=-2095.8
AIC=4201.6   AICc=4201.75   BIC=4221.59
> t_stat(models[[53]])
            ar1      ma1      sma1     drift
t.stat 16.70135 -6.02317 -23.66249 -9.966294
p.val   0.00000  0.00000   0.00000  0.000000
```

Conclusion

mod1.png

mod2.png

Point forecasts

- We had estimated an SARIMA$(1,0,1)(0,1,1)_{12}$:

$$(1-B^{12})X_t = \widehat{\delta} + \frac{(1+\widehat{\theta}_1 B)(1+\widehat{\Theta}_1 B^{12})}{(1-\widehat{\phi}_1 B)} Z_t$$

where $\widehat{\delta} = -1.4884$, $\widehat{\phi}_1 = 0.8386$, $\widehat{\theta}_1 = -0.4932$ and $\widehat{\Theta}_1 = -0.7589$

$\Longleftrightarrow$

$$\begin{aligned} X_t &= \widehat{\delta}(1-\widehat{\phi}_1) + \widehat{\phi}_1 X_{t-1} + X_{t-12} - \widehat{\phi}_1 X_{t-13} + \\ &\quad Z_t + \widehat{\theta}_1 Z_{t-1} + \widehat{\Theta}_1 Z_{t-12} + \widehat{\theta}_1 \widehat{\Theta}_1 Z_{t-13} \end{aligned}$$

Point forecasts

- Assume that we have observations up to time T, $t = T + 1$

$$X_{T+1} = \widehat{\delta}(1 - \widehat{\phi}_1) + \widehat{\phi}_1 X_T + X_{T-11} - \widehat{\phi}_1 X_{T-12} + Z_{T+1} + \widehat{\theta}_1 Z_T + \widehat{\Theta}_1 Z_{T-11} + \widehat{\theta}_1 \widehat{\Theta}_1 Z_{T-12}$$

- Z_{T+1} is replaced by zero and Z_T, Z_{T-11} and Z_{T-12} are replaced resp. by the residuals e_T, e_{T-11} and e_{T-12}. Then

$$\widehat{X}_{T+1|T} = \widehat{\delta}(1 - \widehat{\phi}_1) + \widehat{\phi}_1 X_T + X_{T-11} - \widehat{\phi}_1 X_{T-12} + \widehat{\theta}_1 e_T + \widehat{\Theta}_1 e_{T-11} + \widehat{\theta}_1 \widehat{\Theta}_1 e_{T-12}$$

Point forecasts

- A forecast of X_{T+2} is obtained by replaced t by $T+2$, Z_{T+2} and Z_{T+1} are both zero and

$$\widehat{X}_{T+2|T} = \widehat{\delta}(1-\widehat{\phi}_1) + \widehat{\phi}_1 \widehat{X}_{T+1|T} + X_{T-10} - \widehat{\phi}_1 X_{T-11} + \widehat{\Theta}_1 e_{T-10} + \widehat{\theta}_1 \widehat{\Theta}_1 e_{T-11}$$

- and the process continues for all h, $\widehat{X}_{T+2|T}$.

Interval forecasts

- Denote by $\widehat{\theta}_i$ the estimators of the paramameters θ_i and Θ_i in the MA and SMA components of a SARIMA, model, Then the estimation of the variance the point forecast is for all $h \geq 1$ is

$$\widehat{\sigma}_h^2 = \begin{cases} \widehat{\sigma}^2 & \text{if } h = 0 \\ \widehat{\sigma}^2 \left[1 + \sum_{i=1}^{h-1} \widehat{\theta}_i^2 \right] & \text{if } h \geq 2 \end{cases}$$

where $\widehat{\sigma}^2$ is the sample variance of the residuals

- A 95% prediction interval is then $\widehat{X}_{T+h|T} \pm 1.96\sqrt{\widehat{\sigma}_h^2}$.

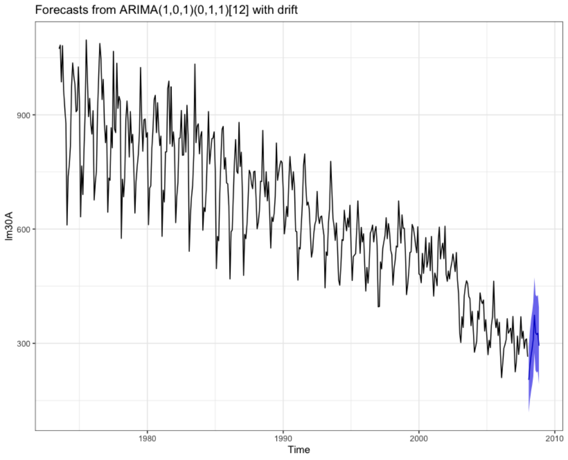
Practice with R

```
> forecast(models[[53]],h=10,level=95,lambda=NULL)
         Point Forecast    Lo 95    Hi 95
Feb 2008       204.2387 118.9182 289.5592
Mar 2008       242.0367 151.7695 332.3039
Apr 2008       267.3297 173.7401 360.9193
May 2008       286.7152 190.8580 382.5723
Jun 2008       310.7436 213.3234 408.1638
Jul 2008       374.1028 275.5982 472.6074
Aug 2008       328.3339 229.0738 427.5940
Sep 2008       323.2001 223.4121 422.9881
Oct 2008       326.6692 226.5117 426.8268
Nov 2008       293.8455 193.4288 394.2621
```

Practice with R

```
> models[[53]]%>%forecast(h=10,level=95,lambda=NULL)%>%
+  autoplot()+theme_bw()
```

Practice with R




```
> model
+  autc
```

Prediction Errors

- Mean Square Error

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(X_i - \widehat{X}_i)^2$$

- Root Mean square error

$$\text{RMSE} = \sqrt{\text{MSE}}$$

- Mean Absolute Error

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|X_i - \widehat{X}_i|$$

Practice with R

```
> library(DMwR)
> lm30A_tr<-window(lm30A, end = c(2006))
> length(lm30A_tr)
[1] 391
> mod<-Arima(lm30A_tr,order = c(1,0,1),seasonal = list(order=c(0,1,1),
+         period=12),lambda = NULL)
> mod
Series: lm30A_tr
ARIMA(1,0,1)(0,1,1)[12]

Coefficients:
         ar1      ma1     sma1
      0.9825  -0.6646  -0.7591
s.e.  0.0143   0.0701   0.0324

sigma^2 estimated as 2058:  log likelihood=-1987.22
AIC=3982.44   AICc=3982.55   BIC=3998.19
> t_stat(mod)
            ar1      ma1      sma1
t.stat 68.80624 -9.47616 -23.43307
p.val   0.00000  0.00000   0.00000
```

Practice with R

```
> f1<-forecast(mod,h=length(lm30A)-length(lm30A_tr),level=95,lambda=NULL)
> f1
         Point Forecast     Lo 95    Hi 95
Feb 2006       230.7649 141.84064 319.6892
Mar 2006       270.8738 177.56402 364.1835
Apr 2006       270.8750 173.51942 368.2305
May 2006       320.0154 218.90817 421.1227
Jun 2006       344.8071 240.20610 449.4080
Jul 2006       406.9439 299.07792 514.8098
Aug 2006       363.2112 252.28490 474.1375
.
.
.
Nov 2007       309.2161 153.57913 464.8531
Dec 2007       340.5698 182.88924 498.2503
Jan 2008       263.8639 104.23568 423.4921
> error_p=regr.eval(window(lm30A, start = c(2006,2)), f1$mean)
> error_p
         mae          mse         rmse         mape
2.733394e+01 1.062221e+03 3.259174e+01 8.998604e-02
```

Practice with R, Displaying the prediction

Create a dataset containing: Observed values, fitted values, Point forecasts, Limits of Prediction interval (95%).

```
> x_date=as.Date(time(lm30A))  #date
> x_fitted=c(mod$fitted,f1$mean) # fitted and point forecast
> x_observed=lm30A # data
> x_95lower=c(rep(NA,length(lm30A_tr)),f1$lower) # lower bound
> x_95upper=c(rep(NA,length(lm30A_tr)),f1$upper)   # upper bound
> ## data
> d_pred=data.frame(date=x_date,observed=x_observed,
+                   fitted=x_fitted,lower95=x_95lower,
+                   upper95=x_95upper)
> head(d_pred)
        date  observed    fitted lower95 upper95
1 1973-07-01 1073.4136 1072.3402      NA      NA
2 1973-08-01 1083.0620 1081.9789      NA      NA
3 1973-09-01  986.3155  985.3292      NA      NA
```

Practice with R, Displaying the result

```
> ### Reshaping the data
> library(reshape2)
> d_pred_w=melt(d_pred,measure.vars = 2:3,id.vars = c(1,4:5))
Warning message:
attributes are not identical across measure variables; they will be dropped
> ### The code for the figure
> p<- ggplot(data = d_pred_w, aes(x = date,y = value,col=variable)) +
+   geom_line() +  geom_ribbon(aes(ymin = lower95, ymax = upper95),
+                              col="black", alpha = .25) +
+   ggtitle("ARIMA(1,0,1)(0,1,1)[12]")+
+   ylab("Number of fatalities")+xlab("")
> p+theme_bw()+scale_color_manual(labels = c("Data", "Fitted"),
+                                 values = c("blue", "red"))+
+   theme(legend.title = element_blank(),
+         legend.position = "bottom")
```

Practice with R, Displaying the result

```
> ### Reshaping the data
> library(reshape2)
> d_pre
Warning
attribu                                                            opped
> ### T
> p<- g
+   geo
+
+   ggt
+   yla
> p+the
+                                  values = c("blue", "red"))+
+   theme(legend.title = element_blank(),
+         legend.position = "bottom")
```

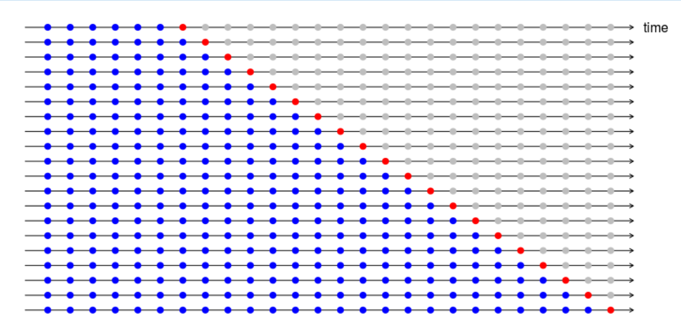
Time series cross-validation

- `tsCV` from `forecast` package
- Article: Bergmeir, Hyndman and Koo (2015)
- Let $(X_t)_{t=1,\dots,T}$ be the time series. Let $h \geq 1$. We consider a forecast function that can be applied successively to the time series $(X_t)_{t=1,\dots,T-h}$ and predict $\widehat{X}_{t+h}$. The errors are given by

$$e_{t+h} = X_{t+h} - \widehat{X}_{t+h}.$$

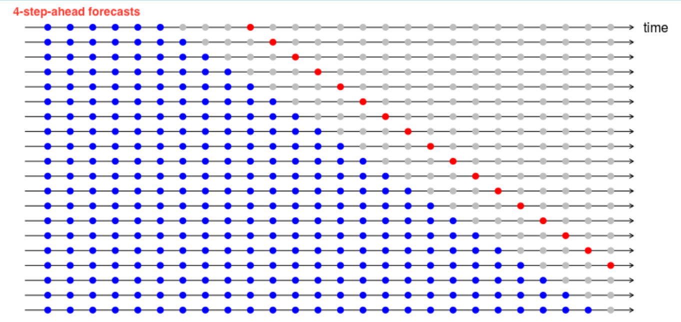
Time series cross-validation

- ts
- A
- L… a fo… e se… by



Time series cross-validation

- ts
- A
- L a
  fo e
  se by



Time series cross-validation, Example, h=1

```
> h=1
> TT=length(lm30A)
> tt=20
> x=lm30A[1:(TT-h)]
> mod<-Arima(x[1:tt],order = c(1,0,1),
+            seasonal = list(order=c(0,1,1),period=12),
+            lambda = NULL)
> xhat=forecast(mod,h=h)
> x[tt+h]-xhat$mean[1]
[1] 39.54791
> forefunction=function(x,h){
+   modx<-Arima(x,order = c(1,0,1),
+               seasonal = list(order=c(0,1,1),
+                               period=12),
+               lambda = NULL)
+   forecast(modx,h=h)
+ }
> er<- tsCV(lm30A,forefunction,h=h)
> er[tt]
[1] 39.54791
```

Time series cross-validation, Example, h=1

```
> tt=120
> x=lm30A[1:(TT-h)]
> mod<-Arima(x[1:tt],order = c(1,0,1),
+            seasonal = list(order=c(0,1,1),period=12),
+            lambda = NULL)
>
> xhat=forecast(mod,h=h)
>
> x[tt+h]-xhat$mean[1]
[1] 86.45593
> er[tt]
[1] 86.45593
```

Time series cross-validation, Example, h=2

```
> h=2
> tt=120
> x=lm30A[1:(TT-h)]
> mod<-Arima(x[1:tt],order = c(1,0,1),
+            seasonal = list(order=c(0,1,1),period=12),
+            lambda = NULL)
>
> xhat=forecast(mod,h=h)
> x[tt+h-1]-xhat$mean[1]
[1] 86.45593
> x[tt+h]-xhat$mean[2]
[1] -54.76314
>
> er<- tsCV(lm30A,forefunction,h=h)
> er[tt,]
      h=1       h=2
 86.45593 -54.76314
```

Selection by Time Series cross-validation

1. Consider the 5 five best models: minimum AIC, Significant coefficients, WN residuals
2. Consider TS cross-validation at the horizon h=3 (or more)
3. Comparing the distribution of the errors by horizon h
4. Consider the model that minimizes RMSE, MSE, and/or MAE

Selection by Time Series cross-validation, Example

```
> bestmodels=vector('list',5)
> j=as.numeric(rownames(dt[i,]))
> for(k in 1:5) bestmodels[[k]]=models[[j[k]]]
> err_cv=vector('list',5)
> for(k in 1:5){
+   print(k)
+   zz=bestmodels[[i]]$arma
+   od=c(zz[1],zz[6],zz[2])
+   od_s=c(zz[3],zz[7],zz[4])
+   forefunction=function(x,h){
+     modx<-Arima(x,order = od,
+                 seasonal = list(order=od_s,
+                                 period=zz[5]),
+                 lambda = NULL)
+     forecast(modx,h=h)
+   }
+   err_cv[[k]]=tsCV(lm30A,forefunction,h=3)
+ }
```

Selection by Time Series cross-validation, Example

```
> for(k in 1:5) names(err_cv)[k]=as.character(dt[i[k],1])
> library(plyr)
> err_cv_d=ldply(err_cv)
> err_cv_dw=melt(err_cv_d,measure.vars = 2:4)
> err_cv_dw=na.omit(err_cv_dw)
> colnames(err_cv_dw)=c("Model","h","Error")
> p<-ggplot(err_cv_dw,aes(y=Error,x=Model))+geom_boxplot(aes(fill=h),alpha=.4)+the
> p+coord_flip()
```

Selection by Time Series cross-validation, Example

```
> for(k
> libra
> err_c
> err_c
> err_c
> colna
> p<-gg                                              lpha=.4)+the
> p+coo
```

compare_boxplot.png

Selection by Time Series cross-validation, Example

```
> library(dplyr)
>
> rmse=err_cv_dw%>%group_by(Model,h)%>%
+   summarise(rmse=sqrt(sum(Error^2)))
>
> p<-ggplot(rmse,aes(y=rmse,x=h,group=Model))+
+   geom_line(aes(color=Model))+
+   geom_point(aes(color=Model),size=3)+theme_bw()
> p+ylab("RMSE")
```

Selection by Time Series cross-validation, Example

```
> libra
>
> rmse=
+   sum
>
> p<-gg
+   geo
+   geo
> p+yla
```

rmse.png