INTRODUÇÃO À ECONOMETRIA

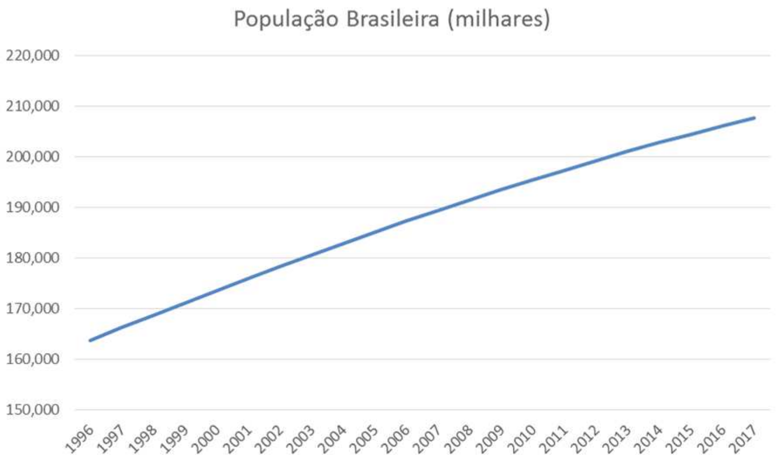
Séries de tempo

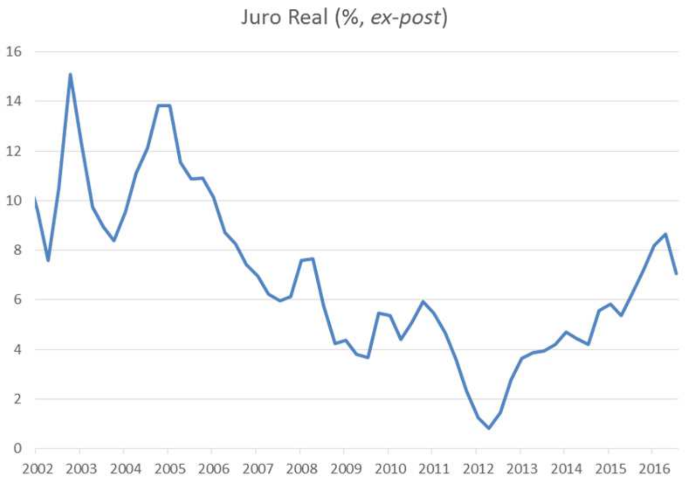
Aula 11

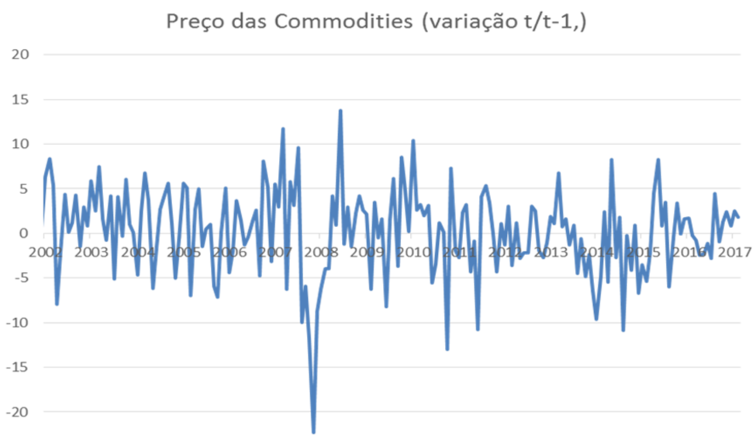
Escola Nacional de Administração Pública

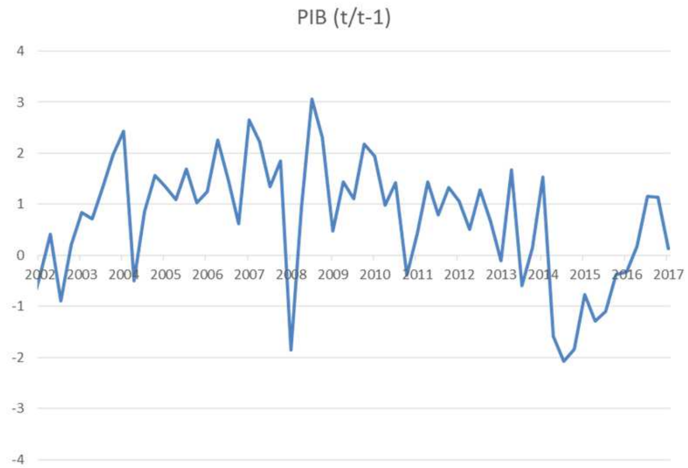
REVISÃO
INTRODUÇÃO – SÉRIES DE TEMPO

COMO CLASSIFICARIA AS SÉRIES ABAIXO?



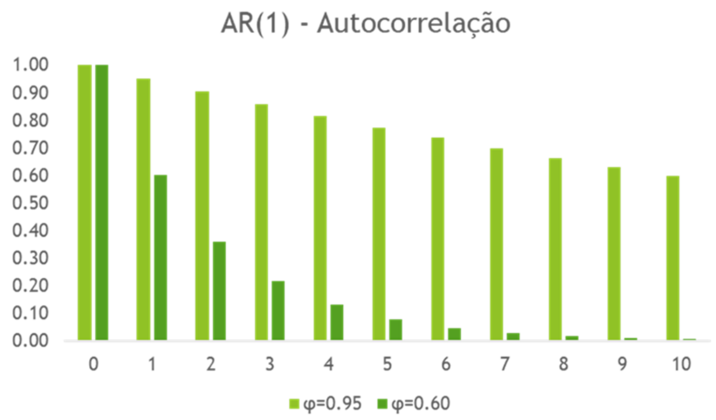


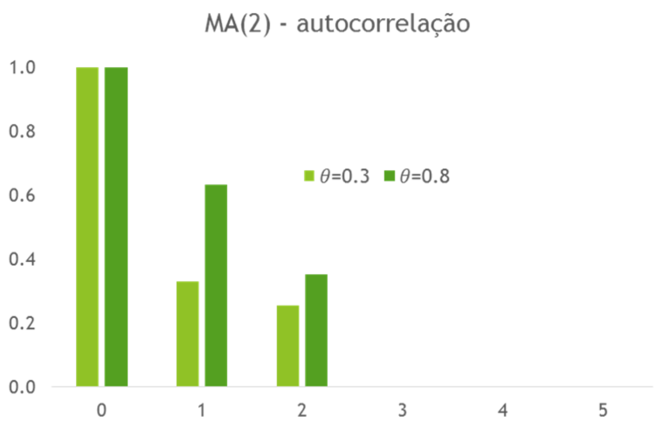


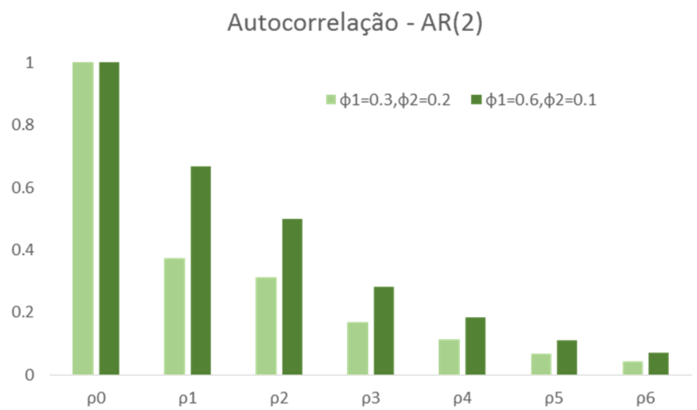


Autocorrelação – Não se esqueça!

- MA – ela interrompe na última defasagem do MA;
- AR – declinante com as defasagens;
- Maior o coeficiente, mais lento o declínio da autocorrelação;
- Variância aumento com o coeficiente.







Função de Autocorrelação Parcial - FACP

- A FACP elimina as correlações implícitas entre a variável e suas defasagens, tornando possível estimar o coeficiente da defasagem.
- Como se faz? Elimina-se as correlações implícitas entre duas variáveis.
- Equação: $y_t = \phi_{j,1} y_{t-1} + \phi_{j,2} y_{t-2} + \cdots + \phi_{j,j} y_{t-j} + e_t, \quad j = 1, 2, \ldots,$
- Procedimento consiste em regredir y_t contra y_{t-1} e obter $\widehat{\Phi}_{1,1}$, depois estima-se y_t contra y_{t-1} e y_{t-2}, obtendo $\widehat{\Phi}_{2,1}$ e $\widehat{\Phi}_{2,2}$. A FACP apresenta $\widehat{\Phi}_{1,1}$ e $\widehat{\Phi}_{2,2}$, descartando $\widehat{\Phi}_{2,1}$.

	y_{t-1}	y_{t-2}	y_{t-3}	…	y_{t-p}
1	$\widehat{\Phi}_{1,1}$	0	0	…	0
2	$\widehat{\Phi}_{2,1}$	$\widehat{\Phi}_{2,2}$	0	…	0
3	$\widehat{\Phi}_{3,1}$	$\widehat{\Phi}_{3,2}$	$\widehat{\Phi}_{3,3}$	…	0
…					
P	$\widehat{\Phi}_{p,1}$	$\widehat{\Phi}_{p,2}$	$\widehat{\Phi}_{p,3}$	…	$\widehat{\Phi}_{p,p}$

- A FACP é formada pela diagonal principal!!!

Função de Autocorrelação Parcial - FACP

- A FACP determina a defasagem “p” do AR(p);
- Ou seja, espera-se que os coeficientes $\widehat{\Phi}_{j,j}$ para j<=p são diferentes de zero, já para os coeficientes j>p, são nulos.
- Qual é o total de parâmetros j para ser estimado?
  - Enders (2009) sugere calcular a FACP até j=T/4, em que T é o tamanho da amostra.

Função de Autocorrelação Parcial - FACP

- A FAC define a defasagem do MA(q). A FACP define defasagem do AR(p).
- No primeiro caso, a FAC decai com o aumento de defasagens, e a função de autocorrelação parcial é truncada a partir da defasagem p.
- No segundo caso, a função de autocorrelação é truncada na defasagem q, e a função de autocorrelação parcial decai.
- No caso de uma ARMA (p, q), ambas as funções decaem a partir da defasagem de truncagem.

Modelo	FAC	FACP
$AR\ (p)$	Decai	Truncada na defasagem p
$MA\ (q)$	Truncada na defasagem q	Decai
$ARMA\ (p, q)$	Decai se $j > q$	Decai se $j > p$

FAC e FACP – solução única?

- As defasagens obtidas da FAC e FACP muitas vezes não são claras;
  - Difícil de identificar visualmente.
- É possível estimar mais de um modelo “correto”:
  - Resíduo do modelo é um ruído branco

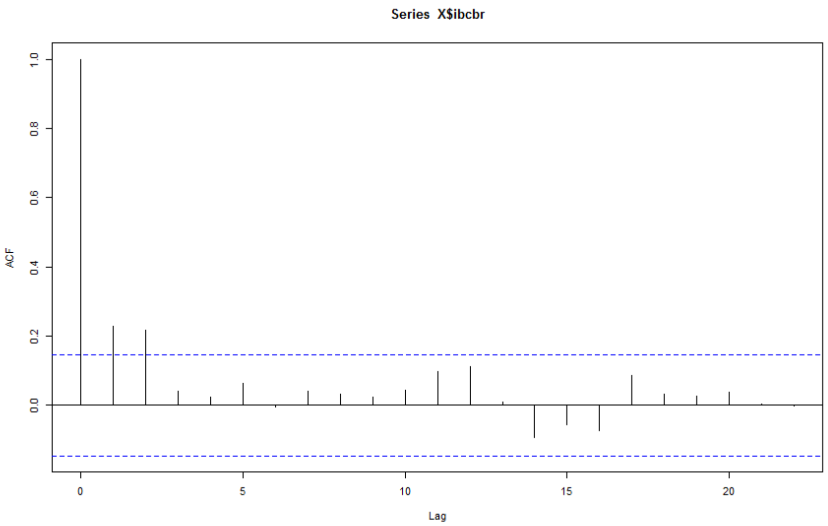
Exemplo real – IBC-br

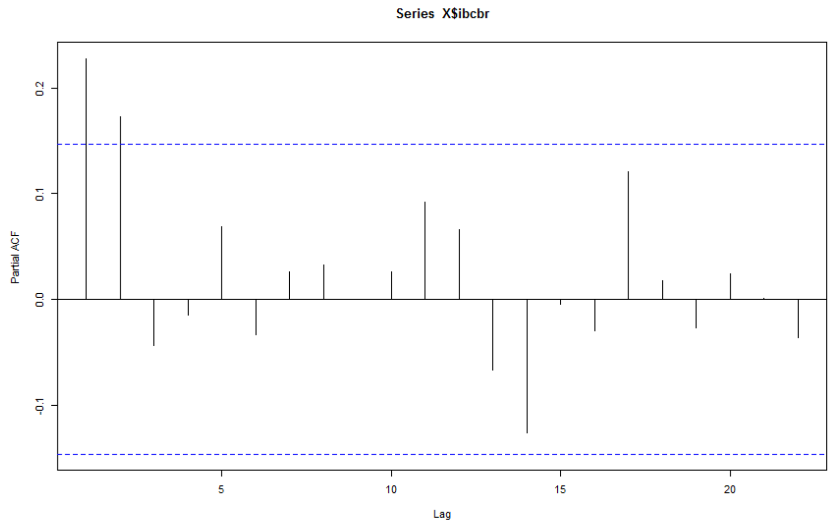
- Baixando os dados.
- setwd("C:/diretorio/diretorio")
- list.files()
- X<-read.csv("aula2_dados.csv",sep=";", dec=".", head=TRUE)
- X$data<-as.Date(X$data,'%d/%b/%y')
- head(X)

Exemplo real – IBC-br

acf(X$ibcbr)

pacf(X$ibcbr)





- Que modelo sugere?

Exemplo real – IBC-br

- Possíveis modelos:
  - ARMA(2,2)
  - AR(2)
  - MA(2)

Exemplo real – IBC-br – ARMA(2,2)

- Instalando o pacote para estimação
  - install.packages("FitARMA")
  - require(FitARMA)
- Estimando o modelo
  - ibcbr_arma22<-FitARMA(X$ibcbr,order = c(2,0,2),MeanMLEQ = TRUE)
  - coef(ibcbr_arma22)
  - erro_arma22<-resid(ibcbr_arma22)

  acf(erro_arma22)

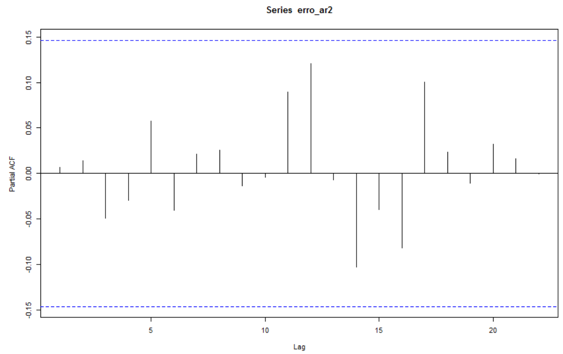
  pacf(erro_arma22)

```
                 MLE         sd     Z-ratio
phi(1)    0.02791488 0.40722151  0.06854961
phi(2)    0.02681891 0.33437672  0.08020567
theta(1) -0.17575284 0.40166412 -0.43756171
theta(2) -0.20180264 0.28398058 -0.71062126
mu        0.19182657 0.07083413  2.70810931
```

Exemplo real – IBC-br – AR(2)

- Estimando o modelo
  - ibcbr_ar2<-FitARMA(X$ibcbr,order = c(2,0,0),MeanMLEQ = TRUE)
  - coef(ibcbr_ar2)

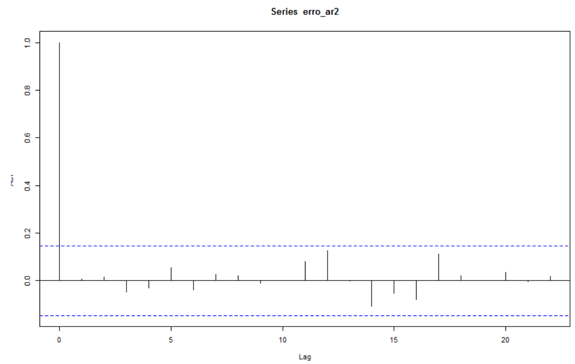
```
              MLE         sd   Z-ratio
phi(1) 0.1916543 0.07358821 2.6044153
phi(2) 0.1751424 0.07358821 2.3800335
mu     0.1926979 0.35278910 0.5462127
```



  - erro_ar2<-resid(ibcbr_ar2)

acf(erro_ar2)

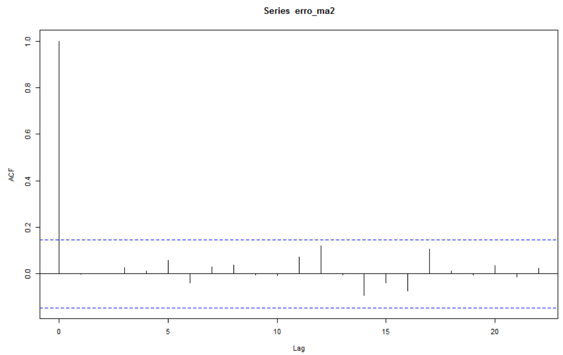
pacf(erro_ar2)



Exemplo real – IBC-br – MA(2)

- Estimando o modelo
  - ibcbr_ma2<-FitARMA(X$ibcbr,order = c(0,0,2),MeanMLEQ = TRUE)
  - coef(ibcbr_ma2)

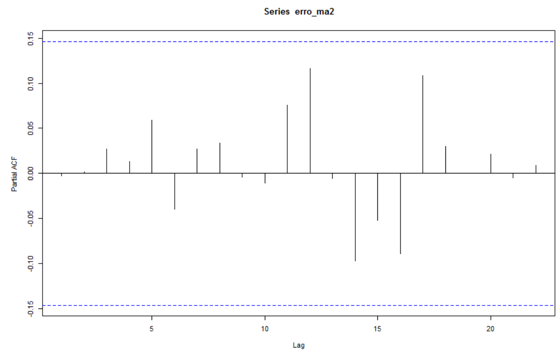
```
                 MLE         sd   Z-ratio
theta(1) -0.2026664 0.07270271 -2.787605
theta(2) -0.2320831 0.07270271 -3.192221
mu        0.1915622 0.05655121  3.387411
```



  - erro_ma2<-resid(ibcbr_ma2)

  acf(erro_ma2)

  pacf(erro_ma2)



ADICIONANDO VARIÁVEIS EXPLICATIVAS

Modelos com defasagens degeneradas

- Incluir no modelo:
  - Defasagens – garantindo que o resíduo seja um RB;
  - *Dummies* sazonais, caso a série não seja ajustada sazonalmente;
  - Variáveis explicativas que tenham poder preditivo da variável dependente.
- Modelo geral:

$$y_t = \alpha + \delta_1 y_{t-j} + \delta_2 y_{t-k} + \beta X_t + \gamma D_t + e_t \text{ para j e k >0}$$

- Importante: resíduo é um RB!

Sazonalidade – Variável binária

- Baixando os dados.
- setwd("diretorio\diretorio")
- list.files()
- W<-read.csv("aula3_dados.csv",sep=";", dec=".", head=TRUE)
- W$data<-as.Date(W$data,'%d/%b/%y')
- head(W)

Modelos com defasagens degeneradas

- Instalando o pacote

```
install.packages("stats") #pacote para na estimação e projeção

require(stats)

install.packages("dyn") #pacote para na estimação e projeção

require(dyn)
```

- Transformando os dataframes em séries temporais

```
arrec=ts(W$arrec_yoy[97:276],start = c(2003,1),frequency = 12)

pim=ts(W$pim_yoy[97:276],start = c(2003,1),frequency = 12)

varejo=ts(W$varejo_yoy[97:276],start = c(2003,1),frequency = 12)
```

- Criando grupo de variáveis

```
dados=cbind(pim,varejo)
```

Modelos com defasagens degeneradas

- Primeira estimação:

```
x<-arima(arrec,order=c(1,0,0),xreg=dados,include.mean = FALSE)
```

- Resultado da estimação

```
x<enter>
```

- Verificando os resíduos

```
erro=x$residuals

acf(erro); pacf(erro)
```

Modelos com defasagens degeneradas

- Segunda estimação:

```
x1=arima(caged,order=c(2,0,2),seasonal=c(1,0,1), xreg=dados,include.mean = FALSE)
```

- Resultado da estimação

```
summary(x1)
```

- Verificando os resíduos

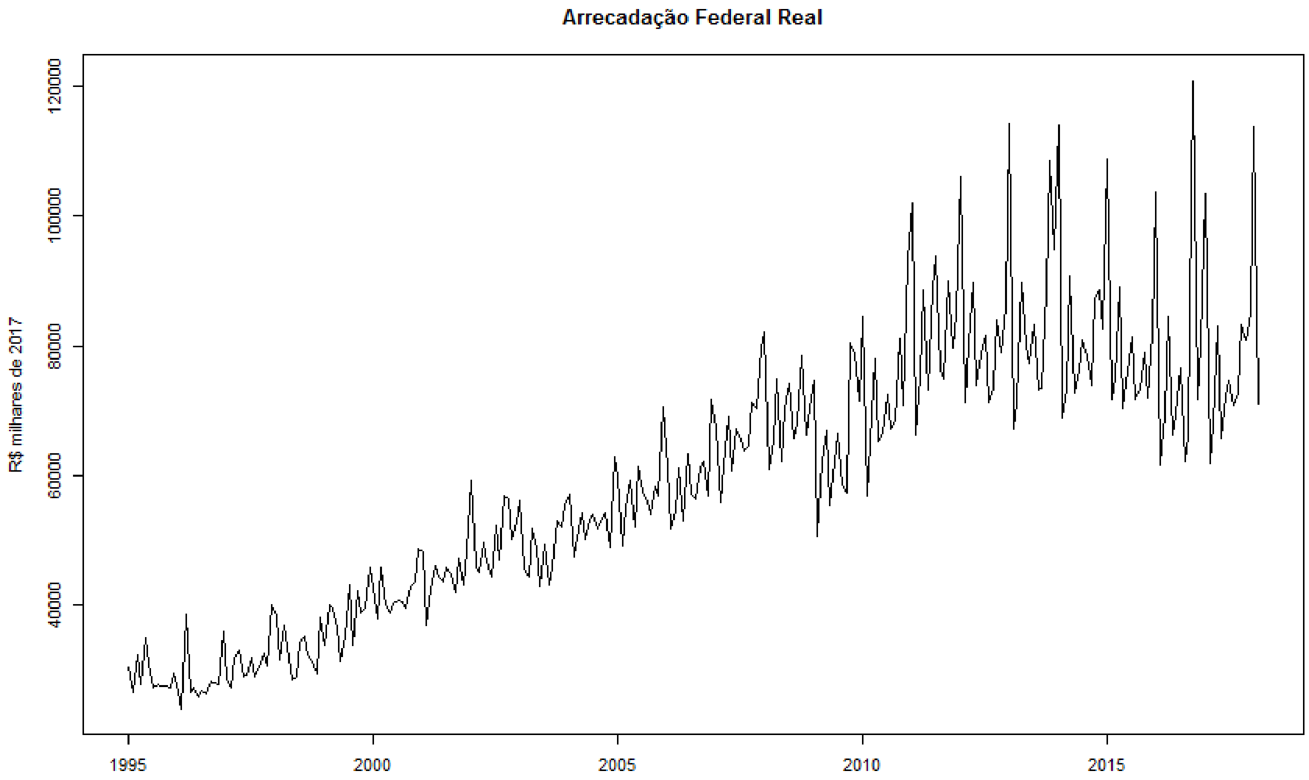
```
erro1=resid(x1)

acf(erro1); pacf(erro1)
```

SAZONALIDADE

Sazonalidade e suavização: visão tradicional

- As séries são ajustadas por algoritmos determinísticos.
- Ignora-se a modelagem de componentes estocásticos porventura existentes.
- Alisamento e dessazonalização procuram expurgar fatores que geram perturbações sistemáticas na série, para ter uma ideia mais precisa da tendência que ela segue.
- A figura a seguir mostra um padrão sazonal pela existência de picos e vales igualmente espaçados ao longo do tempo.

Sazonalidade e suavização: visão tradicional



Sazonalidade e suavização

- O processo estocástico é produto de quatro fatores:

$$y_t = C_t \times S_t \times T_t \times U_t$$

C_t é um componente de ciclo de longo prazo;
S_t é um componente sazonal;
T_t é um componente de tendência;
U_t é um componente irregular.

- Objetivo: estimar e, em seguida, expurgar esse termo da série , para fins de análise e previsão.

Sazonalidade – Variável binária

- Sazonalidade determinística S_t pode ser escrita como uma função de variáveis binárias sazonais;
- Seja a frequência sazonal:

  s=4 para trimestral;

  s=12 para mensal;

- Variável D_{jt} assume valores igual a 1 se j é igual ao mês/trimestre referente do ano:

  Exemplo: D_{1t}=1 se o mês é janeiro, D_{1t}=0 para os outros meses.

- Estimação é por dada MQO e a regressão pode ser desenhada da seguinte forma:

$$y_t = \sum_{i=1}^{s} \gamma_i D_{it} + e_t \quad \text{ou} \quad y_t = \alpha + \sum_{i=1}^{s-1} \beta_i D_{it} + e_t$$

- A diferença é se há constante ou não (multicolinearidade).

Sazonalidade – Variável binária

- Instalando o pacote para estimação sazonal com variável binária

```
install.packages("uroot")
require(uroot)
```

- Tornando o IPCA em uma série temporal

```
ipca_ts<-ts(W$ipca,start = c(1995,1),frequency=12)
ipca_ts1<-ts(W$ipca[121:278],start = c(2005,1),frequency=12)
```

- Criando as variáveis binárias:

```
sd<-seasonal.dummies(ipca_ts)
sd1<-seasonal.dummies(ipca_ts1)
```

Sazonalidade – Variável binária

- Regressão com as variáveis binárias:

```
reg1<-lm(ipca_ts~sd)
summary(reg1)
```

- Regressão com as variáveis binárias sem constante:

```
reg2<-lm(ipca_ts~sd-1)
summary(reg2)
```

- Regressão com as variáveis binárias sem constante:

```
reg3<-lm(ipca_ts1~sd1)
summary(reg3)
```

Sazonalidade – Variável binária

- Forma de estimar a série ajustada sazonalmente:
- Retirando a média:

```
ipca_ts3=ipca_ts-mean(ipca_ts)
```

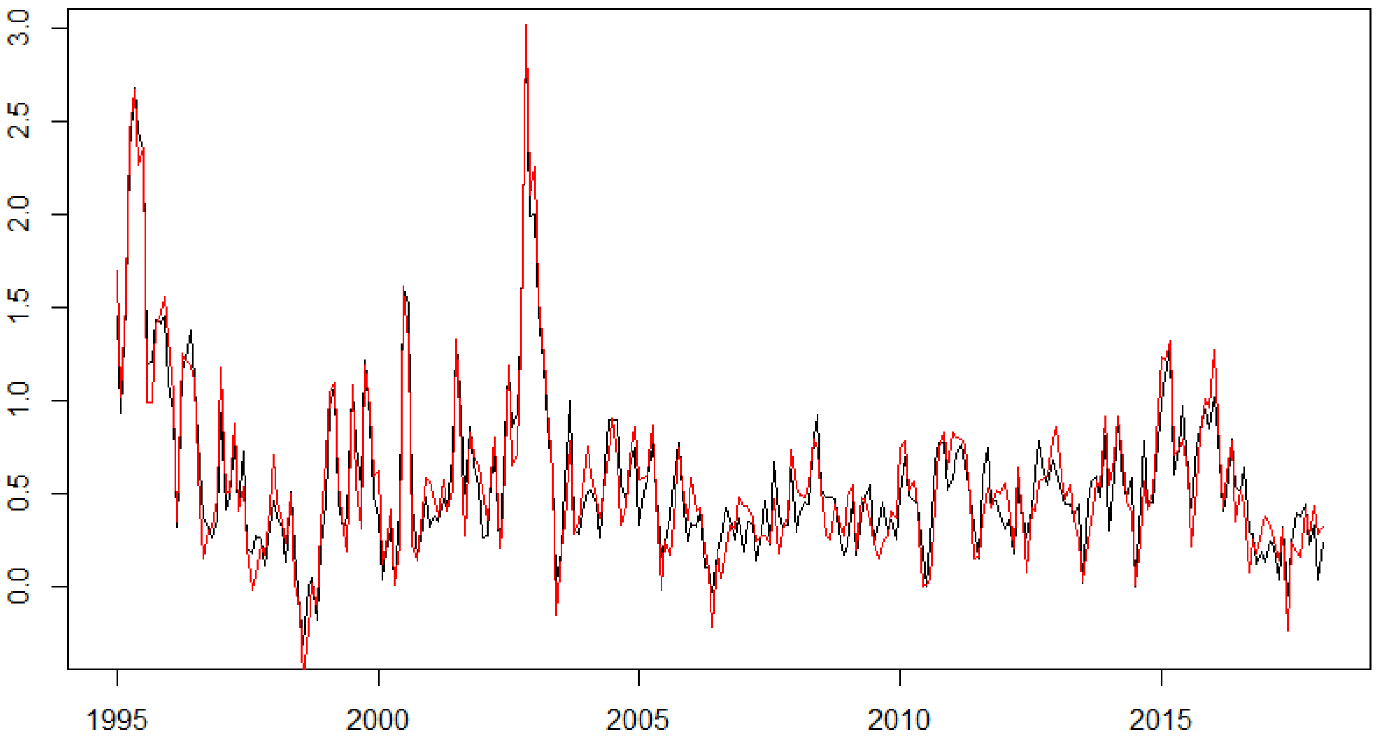
- Estimando a equação:

```
reg3<-lm(ipca_ts3~sd-1)

summary(reg3)
```

- Resultado

```
plot(predict(reg3),type="l") #fatores sazonais

ipca_sa=ts(resid(reg3)+mean(ipca_ts),start=c(1995,1),frequency=12)

plot(ipca_sa)

points(ipca_ts,type="l",col="red")
```

Sazonalidade – Variável binária



RAIZ UNITÁRIA

Motivação

- A série temporal não estacionária não pode ser estimada trivialmente.
- Problema: é impossível estimar todos os momentos da série e fazer inferências estatísticas.

A variância não condicional de um $AR(1)$ é:

$$var(y_t) = \frac{1}{1 - \phi^2}.$$

Se $\phi = 1$, o que caracteriza uma série não estacionária de raiz unitária, então a variância explode.

- Solução: diferenciar a série tantas vezes quantas sejam necessárias para estacionarizá-la.

Regressão Espúria

Considere a seguinte experiência. Gere duas séries I (1) independentemente uma da outra e regrida uma contra a outra. Qual resultado você obtém? Em 75% das vezes, parecer-lhe-á que elas são correlacionadas.

Importante lembrar que espera-se que 95% das vezes, espera-se que as séries não sejam correlacionadas.

Regressão Espúria

Criar um script no R

Digitar as seguintes linhas:

```
y<- e <- rnorm(100,0,1)
y[0]=0
y[1]=0
for (t in 2:100) y[t] <- e[t]+1*y[t-1]
plot(y,type= "l")
z<-y
```

- Rodar novamente, substituir a última linha por **x<-y**

Regressão Espúria

eq=lm(z~x)

summary(eq)

```
Call:
lm(formula = z ~ x)

Residuals:
   Min     1Q Median     3Q    Max 
-8.204 -1.659  0.070  1.787  9.049 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   3.0475     0.7403   4.117 8.02e-05 ***
x             1.5184     0.1065  14.261  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.594 on 98 degrees of freedom
Multiple R-squared:  0.6748,	Adjusted R-squared:  0.6715 
F-statistic: 203.4 on 1 and 98 DF,  p-value: < 2.2e-16
```

CONCLUSÃO: CUIDADO COM A REGRESSÃO QUE SE ESTIMA!!!!

Dickey-Fuller

Considere o seguinte modelo:

$$y_t = \phi y_{t-1} + \varepsilon_t.$$

Tendência: estimar esse modelo e usar um teste convencional de t sobre ϕ, tendo como hipótese nula $H_0 : \phi = 1$.
Alternativamente, poder-se-ia alterar o teste subtraindo y_{t-1} de ambos os lados:

$$\triangle y_t = (\phi - 1)\, y_{t-1} + \varepsilon_t = \alpha y_{t-1} + \varepsilon_t, \tag{2}$$

em que se define $\alpha \equiv \phi - 1$.
Assim, $H_0 : \phi = 1$ é equivalente a $H_0 : \alpha = 0$.
Problema: sob a nula, a distribuição do teste não é convencional, ou seja, não é igual à distribuição t estatística, pois y_t não é estacionário.

Dickey-Fuller aumentado

- Problema do teste anterior: o erro é um ruído branco. Será?

Suponha que y_t seja um processo auto-regressivo de ordem p, com raiz unitária:

$$y_t = \mu + \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} + \phi_{p+1} y_{t-p-1} + \varepsilon_t .$$

- Como testar esse modelo para raiz unitária?
- Idéia: estimar o modelo com as variáveis auto-regressivas. Forma de corrigir o desvio do valor correto da estatística, ou seja, trata-se de encontrar os desvios de y_t em relação à sua "média", para deslocar a distribuição de α em direção a zero, caso a hipótese nula seja verdadeira.

Demais testes de Dickey e Fuller

As hipóteses a testar são:

$$
\begin{aligned}
H_0 &: \alpha = \mu = 0 \rightarrow \Phi_1; \\
H_0 &: \alpha = \delta = \mu = 0 \rightarrow \Phi_2; \\
H_0 &: \alpha = \delta = 0 \rightarrow \Phi_3.
\end{aligned}
$$

Dickey-Fuller aumentado - aplicação

Atualizando o arquivo

```
X<-read.csv("aula4_exercicio.csv",sep=";", dec=".", head=TRUE)
```

Transformando as variáveis em séries temporais

```
ipca<-ts(X$ipca, start=c(2000,1), frequency=12)

populacao<-ts(X$populacao, start=c(2000,1), frequency=12)

varejo<-ts(X$varejo, start=c(2000,1), frequency=12)

selic<-ts(X$selic, start=c(2000,1), frequency=12)

selic_real<-ts(X$selic_real, start=c(2000,1), frequency=12)

brl<-ts(X$brl, start=c(2000,1), frequency=12)
```

Dickey-Fuller aumentado - aplicação

Instalando o pacote tseries – escolha automática

```
install.packages("tseries")
require(tseries)
```

Analisando o resultado para quatro séries

```
adf.test(ipca)
adf.test(populacao)
adf.test(selic)
adf.test(diff(selic))
adf.test(selic_real)
```

Escolha automática da defasagem e se há constante ou tendência

EXERCÍCIO

Séries de tempo
Exercício

1. Calcule o FAC e FACP para alguma série da economia brasileira (Juros, Varejo e etc...)
2. Estime o ARMA e observe se o resíduo é ruído branco.
3. Observe de o índice da produção industrial tem raiz unitária, se sim, como torna-la estacionária.