

Introdução ao R com Exemplos Aplicados

Cláudio Djissey Shikida*

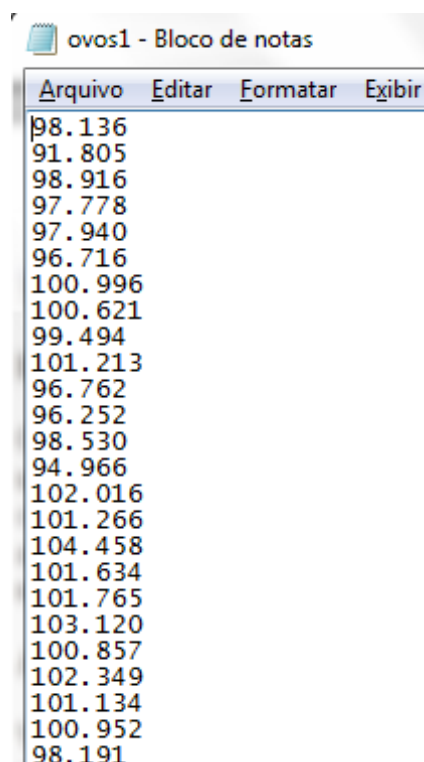
* Professor do Ibmec Minas Gerais.

Introdução

O programa R não é como os outros programas econométricos. Na verdade, a econometria é apenas uma de suas qualidades. O R funciona à base de módulos que podem ser instalados e removidos conforme as restrições do usuário. Portanto, não seria correto dizer que estas notas de aula ensinam a operar no ambiente R. Na verdade, escolhi alguns tópicos para ilustrar como funciona o R. Nesta nota, mostraremos como estimar um modelo (S)ARIMA, seguindo a metodologia Box-Jenkins¹.

ARIMA no R

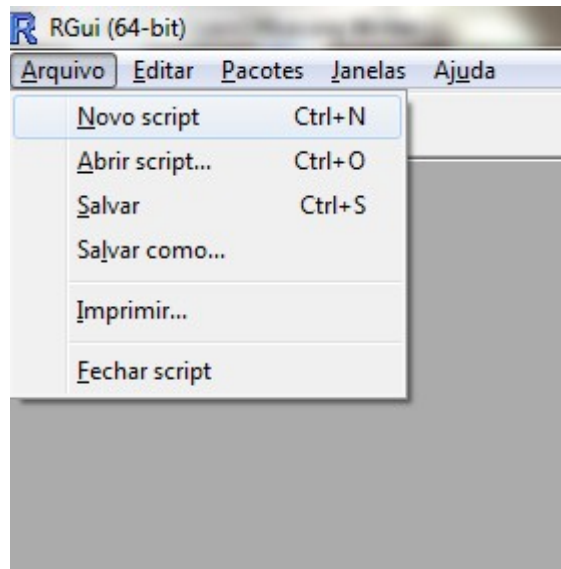
Vamos trabalhar com a série de produção de ovos (mil dúzias, abrangência nacional) do IBGE. Após consulta à base de dados SIDRA, obtivemos os dados. O primeiro passo é colocá-los em formato ASCII – erroneamente conhecido como “formato txt” - e, como esta é apenas uma série, vamos omitir o nome da variável inicialmente. Exportamos os dados do Excel para o arquivo ovos1.txt. O arquivo tem a seguinte aparência:



```
ovos1 - Bloco de notas
Arquivo  Editar  Formatar  Exibir
98.136
91.805
98.916
97.778
97.940
96.716
100.996
100.621
99.494
101.213
96.762
96.252
98.530
94.966
102.016
101.266
104.458
101.634
101.765
103.120
100.857
102.349
101.134
100.952
98.191
```

A série começa em janeiro/1987 e vai até março/2010. Uma vez que estamos com a série em formato ASCII, no R, criamos um novo *script*.

¹ Estas notas de aula se baseiam em rotina escrita por Erik Alencar Figueiredo, professor do doutorado da UFPB. A ele todos os créditos e, a mim, o *disclaimer* usual: todos os erros e omissões são de minha exclusiva responsabilidade (ainda que na terra do *mensalão* isso não signifique muita coisa...). Agradeço ao professor Cristiano Costa por me provocar em plena manhã de sábado, e aos marceneiros que, ao me desligarem da internet, telefone e TV a cabo, impedindo-me, inclusive, de desencaixotar meus livros, obrigaram-me a cumprir, obsessivamente, a tarefa de gerar este bem público. Desculpo-me com o(s) leitor(es) pela formatação imprecisa do texto, mas meus conhecimentos de OpenOffice são similares aos meus conhecimentos acerca dos resultados da Mega-Sena.



Abre-se uma tela similar à do bloco de notas e nosso primeiro comando é importar o arquivo ovos1.txt. Eis o comando, já no novo *script*:

```
Sem nome - Editor R
###Chamando os dados das dúzias de ovos
y=scan("C:/Users/cdshikida/Documents/Meus Documentos/cursos/Econometria II 2009/ovos/ovos1.txt")
```

Escreva-o e, ao final da “frase”, digite Ctrl+R (a tecla ctrl e a tecla R). Você poderá ver uma nova janela, no canto superior esquerdo, com o mesmo comando em vermelho. Se não houver mensagens de erro, podemos prosseguir.

Pois bem, há várias formas de se importar dados para o R, e talvez esta seja a maior dificuldade de quem está a iniciar sua convivência com o programa. Vale, realmente a pena, tentar usar o procedimento acima (criar o arquivo ASCII e depois chamá-lo com o comando “scan”).

Voltando à nossa série, o R não sabe que os dados referem-se a uma série de tempo. Assim, vamos declará-lo como série de tempo, informando o período inicial e a frequência do dado (mensal, anual, etc). Em nosso caso, os dados são mensais.

```
###Chamando os dados das dúzias de ovos

y=scan("C:/Users/cdshikida/Documents/Meus Documentos/curso:

###Declarando y como uma série temporal

y.ts=ts(y, start=c(1987,1), frequency=12)
```

Note que a frequência 12 se refere a dados mensais. Dados trimestrais seriam invocados com *frequency* = 4. O comando “ts” transforma a nossa única variável, que chamamos de y inicialmente (vide comando inicial) em uma série de tempo, y.ts.

Em seguida, podemos obter dados básicos da série como a média, a mediana ou o desvio-padrão.

```

R Console
> ###Declarando y como uma série temporal
>
> y.ts=ts(y, start=c(1987,1), frequency=12)
>

Sem nome - Editor R
###Chamando os dados das dúzias de ovos
y=scan("C:/Users/cdshikida/Documents/Meus Document
###Declarando y como uma série temporal
y.ts=ts(y, start=c(1987,1), frequency=12)
###Obtendo informacoes preliminares sobre a seri
mean(y.ts)
median(y.ts)
sqrt(var(y.ts))
fivenum(y.ts)

```

Você pode selecionar a área que começa em *mean(y.ts)* e termina em *fivenum(y.ts)* e digitar, novamente, Ctrl+R (a tecla ctrl e a tecla R). Os resultados aparecerão na tela ao lado, com os comandos em vermelho. Note o último comando. Ele retorna o mínimo valor, o primeiro quartil, a mediana, o terceiro quartil o valor máximo da série.

```

Arquivo  Editar  Pacotes  Janelas  Ajuda
R Console
> ###Declarando y como uma série temporal
>
> y.ts=ts(y, start=c(1987,1), frequency=12)
> mean(y.ts)
[1] 137.9372
>
> median(y.ts)
[1] 130.528
>
> sqrt(var(y.ts))
[1] 29.97073
>
> fivenum(y.ts)
[1]  91.8050 114.6405 130.5280 159.5785 207.7400
>

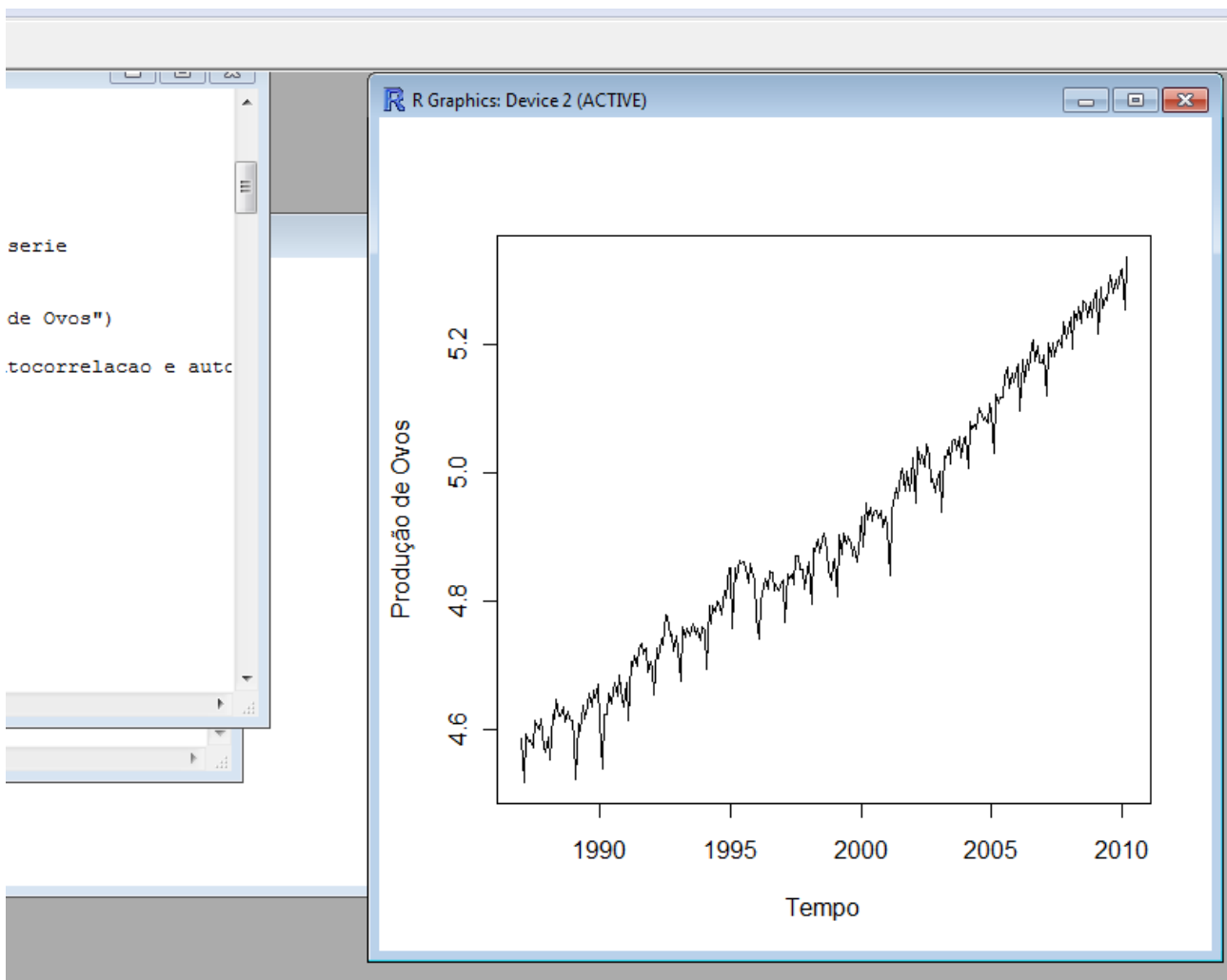
Sem nome - Editor R
###Chamando os dados das dúzias de ovos
y=scan("C:/Users/cdshikida/Documents/Meus
###Declarando y como uma série temporal
y.ts=ts(y, start=c(1987,1), frequency=12)|
###Obtendo informacoes preliminares sobre
mean(y.ts)
median(y.ts)

```

Observe que as saídas dos resultados sempre estão em cor azul. Pois bem, vamos em frente. Vamos obter o logaritmo da série e, em seguida, pedir um gráfico simples da mesma contra o tempo (você não se esquece mais do Ctrl+R, certo?).

```
#####Obtendo o Logaritmo da variável  
  
ly.ts=log(y.ts)  
  
#####Plotando o grafico do logaritmo da serie  
  
plot(ly.ts, xlab="Tempo", ylab="Produção de Ovos")
```

Observe o resultado:



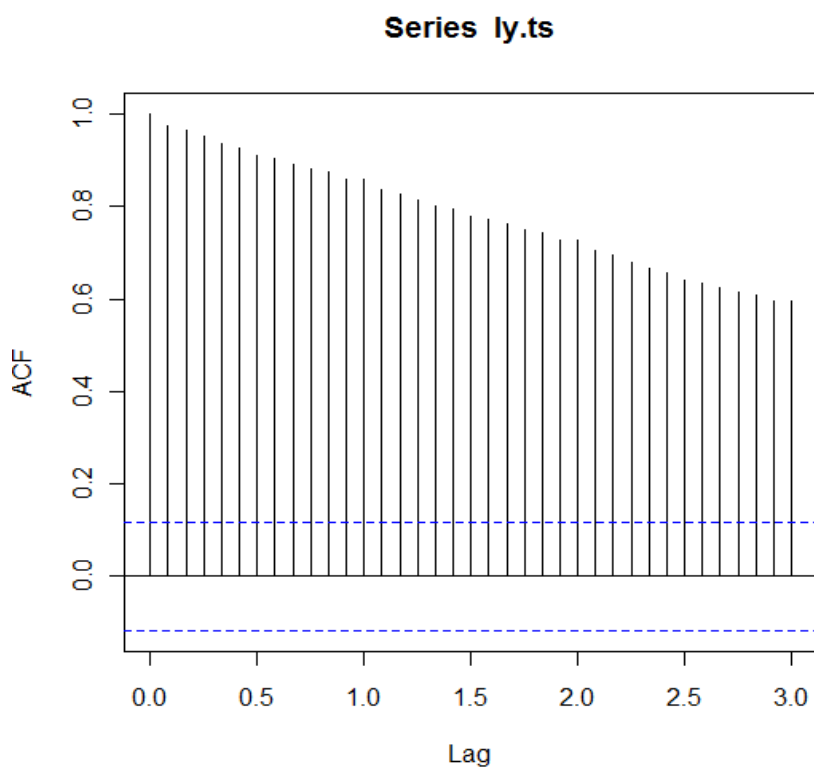
Notou como declarar o título dos eixos? Bem, é fácil ver que a série tem uma tendência de crescimento marcada por oscilações que levantam fortes suspeitas de que tenhamos aí um padrão sazonal. Assim, vamos começar a análise Box-Jenkins *per se*.

Inicialmente, identificamos o(s) modelo(s). Como fazemos isso? Por meio das funções de autocorrelação e autocorrelação parcial. Vejamos o que temos para o nível da série e, também para a sua primeira diferença. Eis os comandos.

```
####Informações sobre suas funções de autocorrelacao e autocorrelacao parcial
###Nivel
## ou : acf2(ly.ts)
acf(ly.ts, lag.max=36)
pacf(ly.ts, lag.max=36)
###Primeira diferenca
acf(diff(ly.ts), lag.max=36)
pacf(diff(ly.ts), lag.max=36)
## ou: acf2(diff(ly.ts))
```

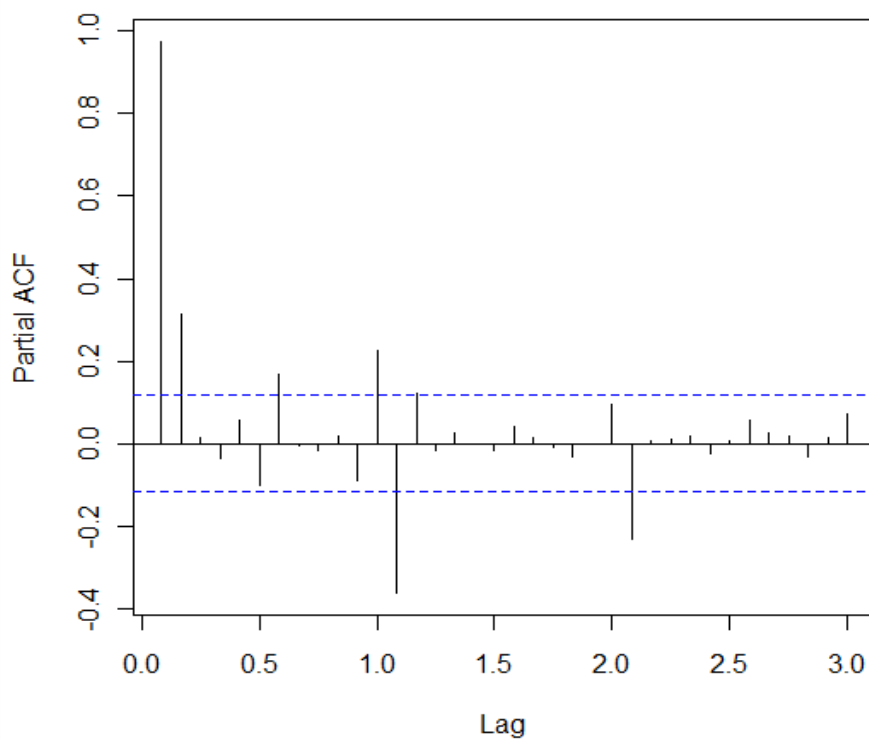
Note que podemos obter a função de autocorrelação da primeira diferença de duas formas, usando o comando `acf` ou o comando `acf2`².

R Graphics: Device 2 (ACTIVE)

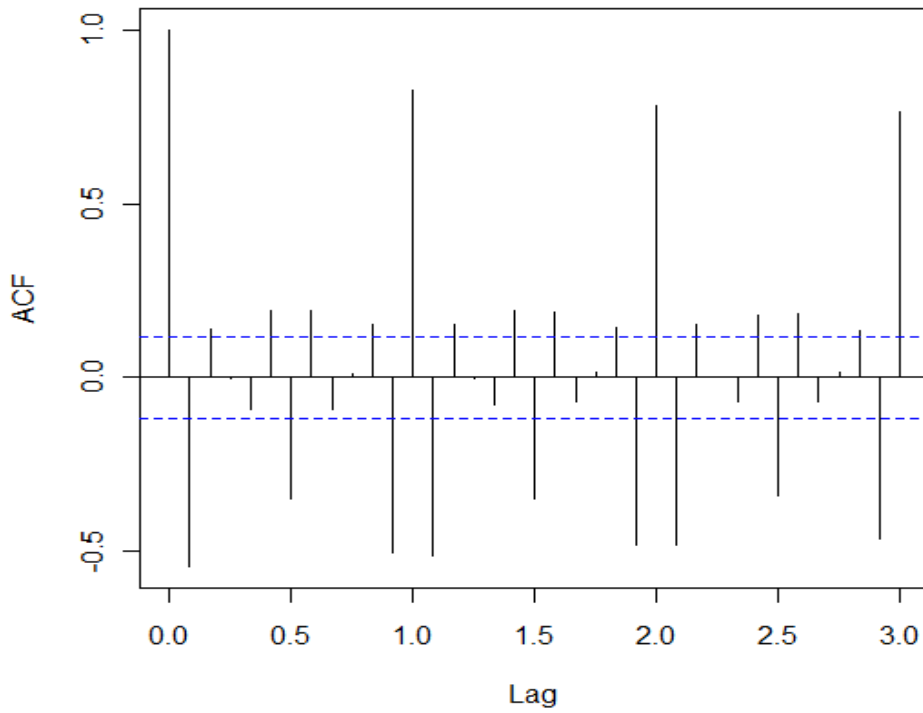


² Este, contudo, deve ser obtido por meio da opção de “instalar pacotes”. Falaremos sobre isso mais adiante.

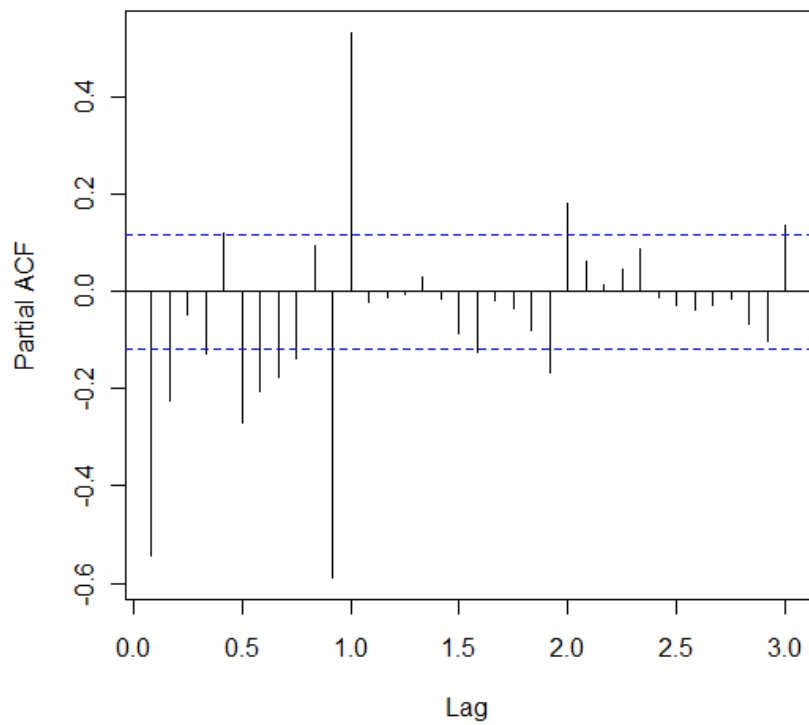
Series ly.ts



Series diff(ly.ts)



Series diff(ly.ts)



Observando a ACF da série em nível, podemos suspeitar de que a mesma seja integrada de ordem 1 ou I(1). A sazonalidade também parece ser importante (e isso já era facilmente perceptível a partir do gráfico da série contra o tempo. Para fins didáticos, vamos supor que estimemos apenas um modelo (na metodologia Box-Jenkins estima-se vários modelos iniciais, mas isso não nos ajudaria a conhecer novos comandos). Estimaremos um SARIMA (0,1,1) (0,1,1). Chamaremos o modelo de ajuste1. Vejamos o comando:

```
#####
###A modelagem padrao eh:

ajuste.1=arima(ly.ts, order=c(0,1,1), seasonal=list(order=c(0,1,1)))

ajuste.1
```

E o resultado...

```
> acf(ly.ts, lag.max=36)
> pacf(diff(ly.ts), lag.max=36)
> pacf(ly.ts, lag.max=36)
> pacf(diff(ly.ts), lag.max=36)
> ajuste.1=arima(ly.ts, order=c(0,1,1), seasonal=list(order=c(0,1,1)))
> ajuste.1

Call:
arima(x = ly.ts, order = c(0, 1, 1), seasonal = list(order = c(0, 1, 1)))

Coefficients:
          ma1          sma1
      -0.0999   -0.9010
s.e.    0.0752    0.0466

sigma^2 estimated as 0.0001953:  log likelihood = 748.54,  aic = -1491.07
> |
```

Note que os coeficientes parecem ser estatisticamente significativos. Assim, façamos uma previsão três passos à frente e, para comparar com a série, voltamos com estes valores à escala original.

```
#####Pedindo uma previsao 3 passos a frente

predict(ajuste.1, 3)$pred

####Retornando a escala original

p1=exp(predict(ajuste.1, 3)$pred)

p1

#####Graficamente

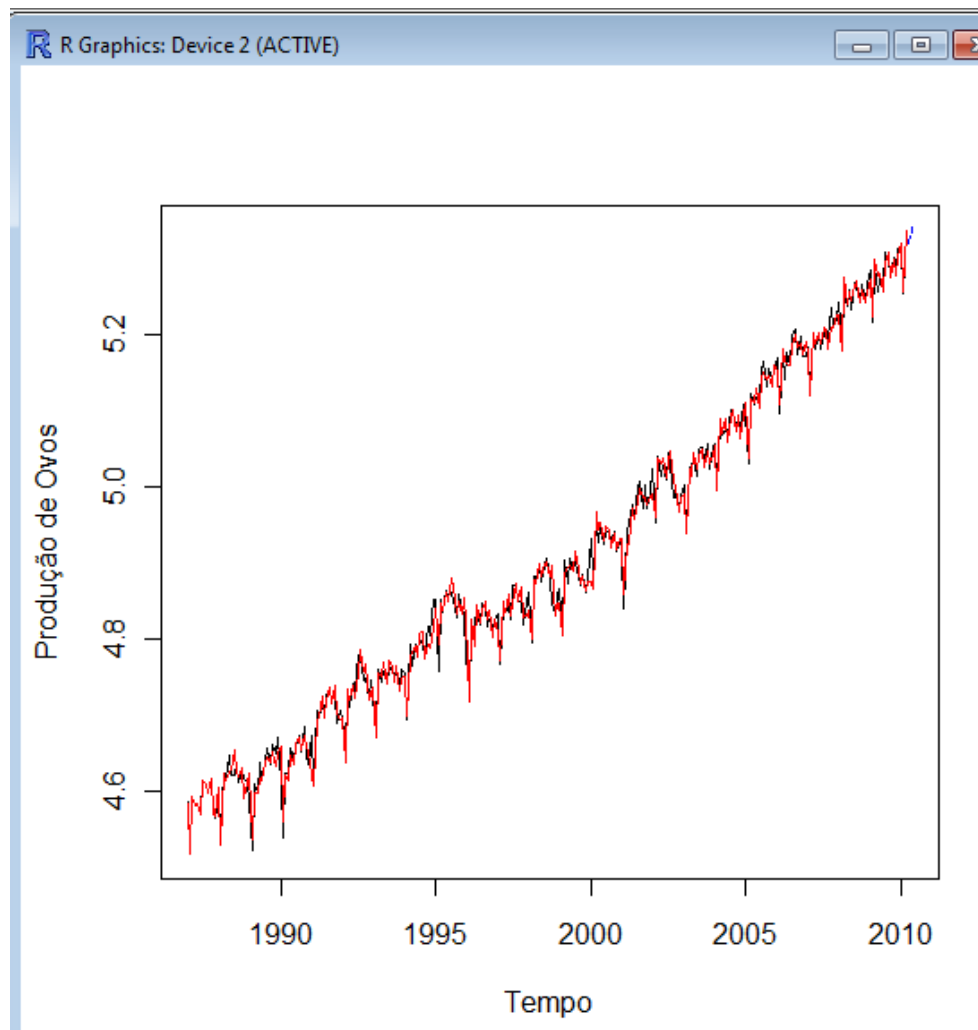
plot.ts(ly.ts, xlim=c(1987.1, 2010.3), xlab="Tempo", ylab="Produção de Ovos")
lines(ly.ts - ajuste.1$resid, col="red")
lines(log(p1), lty=2, col="blue")
```

Vejamos agora os resultados:

```
Coefficients:
      ma1      sma1
-0.0999 -0.9010
s.e.    0.0752  0.0466

sigma^2 estimated as 0.0001953: log likelihood = 748.54, aic = -1491.07
> predict(ajuste.1, 3)$pred
      Apr      May      Jun
2010 5.320369 5.340383 5.324637
>
> #####Retornando a escala original
>
> p1=exp(predict(ajuste.1, 3)$pred)
>
> p1
      Apr      May      Jun
2010 204.4594 208.5927 205.3338
>
> #####Graficamente
>
> plot.ts(ly.ts, xlim=c(1987.1, 2010.3), xlab="Tempo", ylab="Produção de Ovos")
> lines(ly.ts - ajuste.1$resid, col="red")
> lines(log(p1), lty=2, col="blue")
> |
```

O gráfico, em janela própria (geralmente à direita da tela) é o que se segue:



Se o leitor prosseguir com um pouco mais de suor próprio, notará que um ajuste melhor é um SARIMA (1,1,1) (0,1,1). Obviamente, os comandos são similares aos que fizemos anteriormente.

```
ajuste.2=arima(ly.ts, order=c(1,1,1), seasonal=list(order=c(0,1,1)))
ajuste.2
```

Os resultados estão a seguir. Novamente, temos parâmetros significativos e o modelo é bem melhor do que o anteriormente estimado, se considerarmos o critério AIC³.

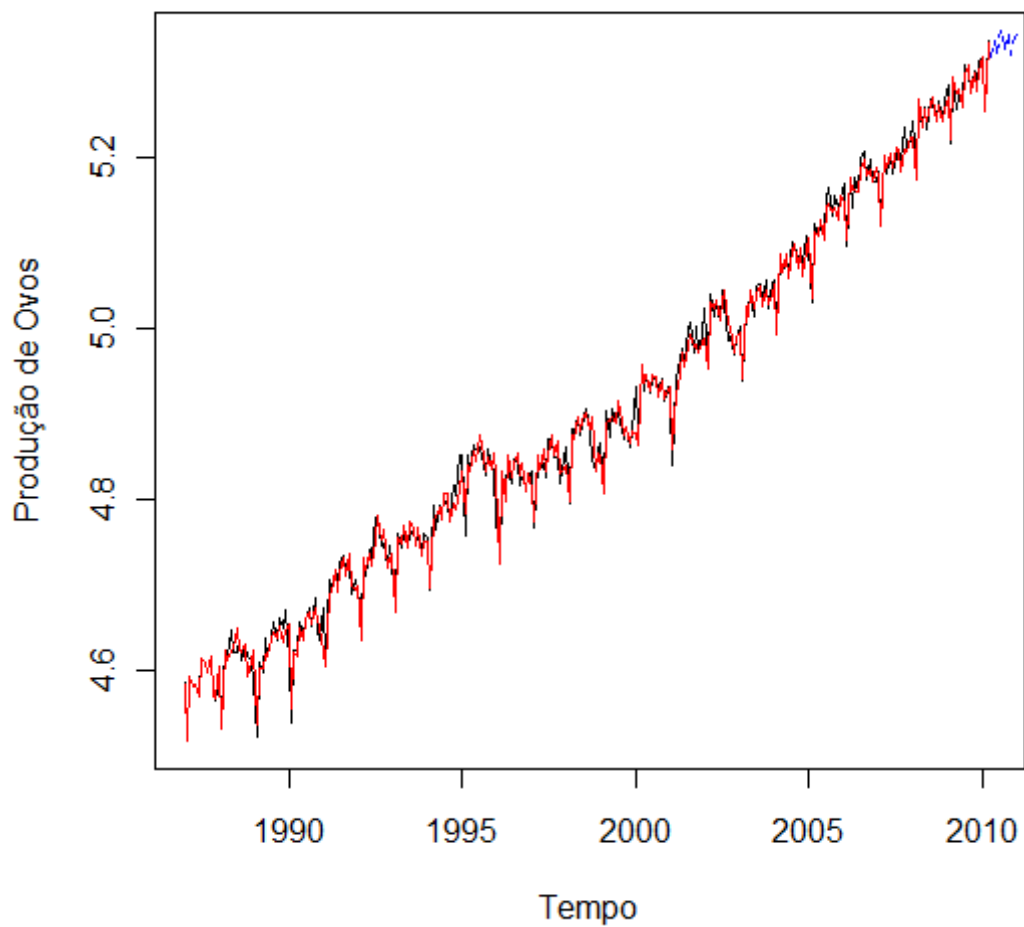
```
Call:
arima(x = ly.ts, order = c(1, 1, 1), seasonal = list(order = c(0, 1, 1)))

Coefficients:
      ar1      ma1      sma1
  0.7230 -0.9062 -0.8962
s.e. 0.0714  0.0443  0.0462

sigma^2 estimated as 0.0001841:  log likelihood = 756.22,  aic = -1504.44
> |
```

3 Embora seja prática comum utilizar critérios como o AIC ou o SBC, Bueno (2008) afirma que estes critérios seriam adequados apenas para processos AR puros. Um procedimento alternativo, implementado no *freeware* Jmulti é o de Hannan-Rissanen.

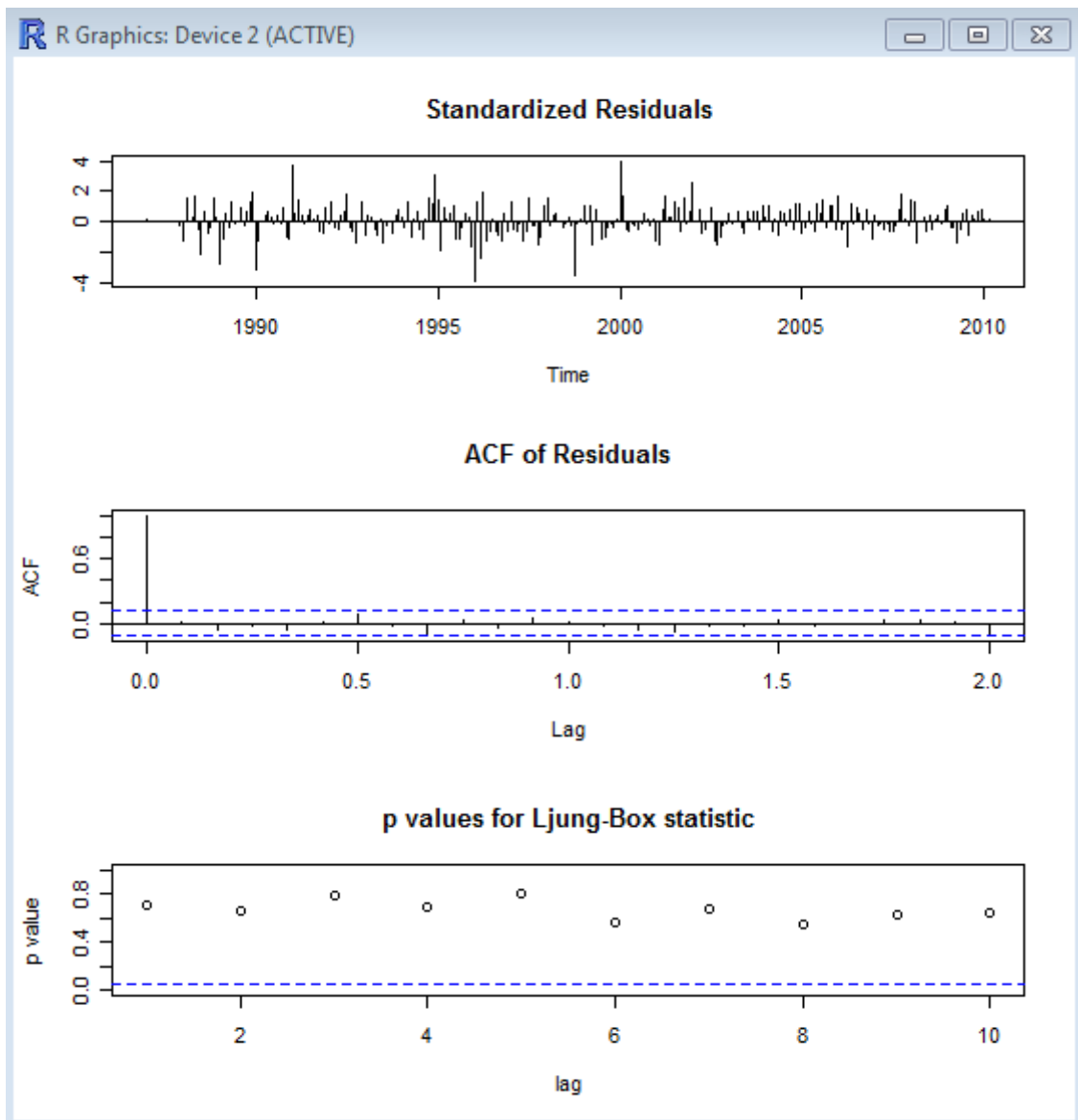
Fazendo a previsão dez passos à frente e o gráfico da série contra a previsão, tem-se:



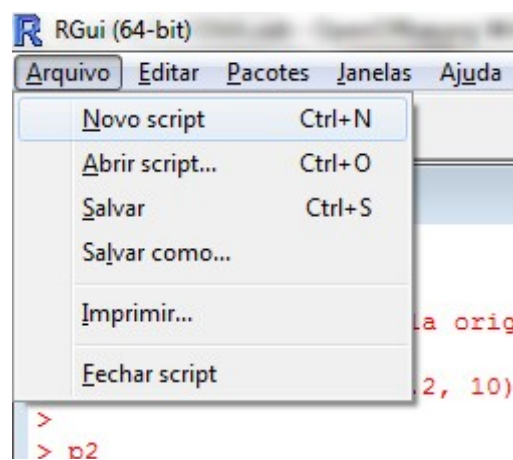
Finalmente, vejamos o comportamento dos resíduos desta regressão.

```
#####Por fim, os testes  
tsdiag(ajuste.2)
```

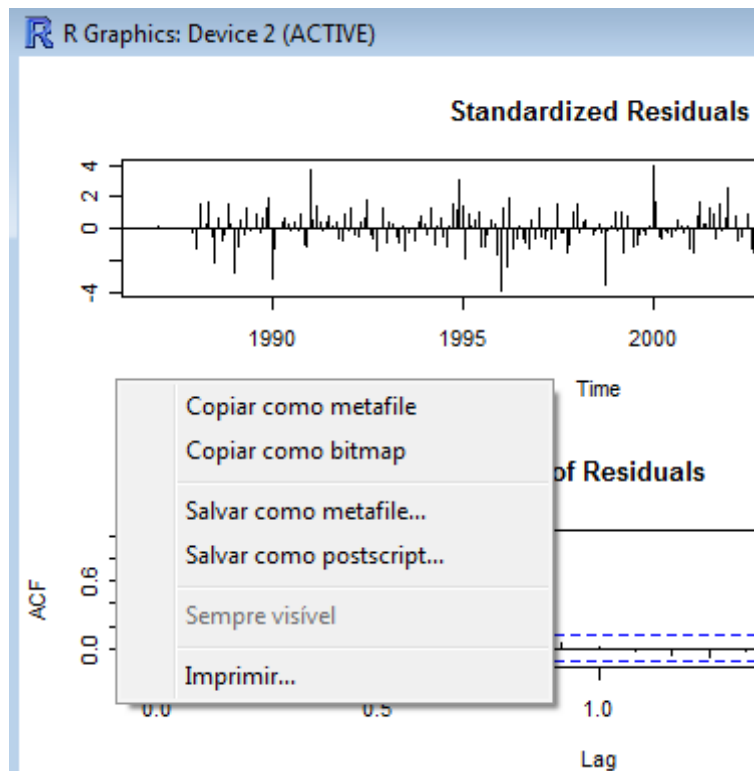
O resultado é um conjunto de três gráficos: os resíduos padronizados, a ACF dos resíduos e os p-valores da estatística Ljung-Box para as autocorrelações dos resíduos.



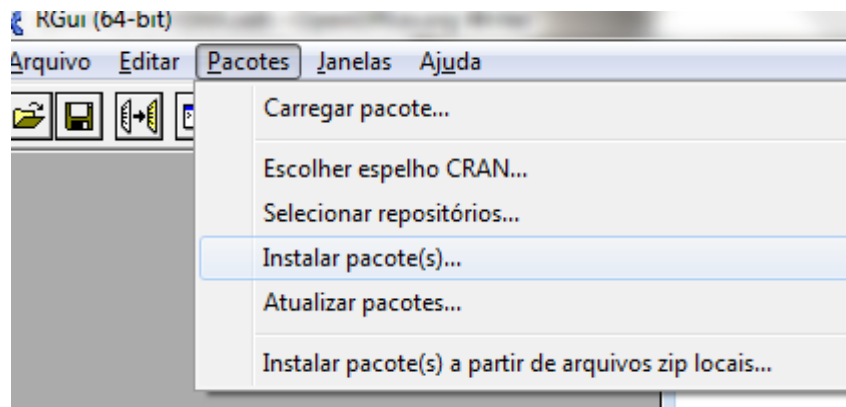
Os comandos apresentados até aqui estão todos em um arquivo de *script* e o mesmo pode ser salvo facilmente.



Os gráficos do R também podem ser facilmente salvos. Ao clicar em qualquer um deles com o botão direito do *mouse*, obtém-se várias opções de arquivos de imagem.



Finalmente, como prometido na nota de rodapé 2, eis aqui algumas dicas sobre pacotes para R. Existe uma ampla biblioteca de pacotes no *site* oficial do R (<http://www.r-project.org>) e nos *mirrors* lá indicados. Do próprio R, você pode obter uma lista de pacotes⁴ para instalação. Vejamos os comandos:



Após escolher “Instalar pacotes”, você obterá uma longa lista de *mirrors* (ou, como outro comando do menu diz: “espelho CRAN” para vários países do mundo. Basta escolher um deles e você obterá uma imensa lista de programas.

No próprio *site* oficial do R existe uma página que agrupa e fala brevemente dos principais pacotes do R para esta espécie genial de econométristas que alegram nossas pesquisas. Fica ao leitor a tarefa de encontrar esta página e, a partir daí, boa diversão!

Teste ADF no R

Como visto acima, um aspecto muito importante do estudo de séries de tempo encontra-se na identificação do modelo. Embora as ACF e PACF nos ajudem um bocado com os padrões AR e MA, também é verdade que devemos saber quantas vezes diferenciar a série para que a mesma seja

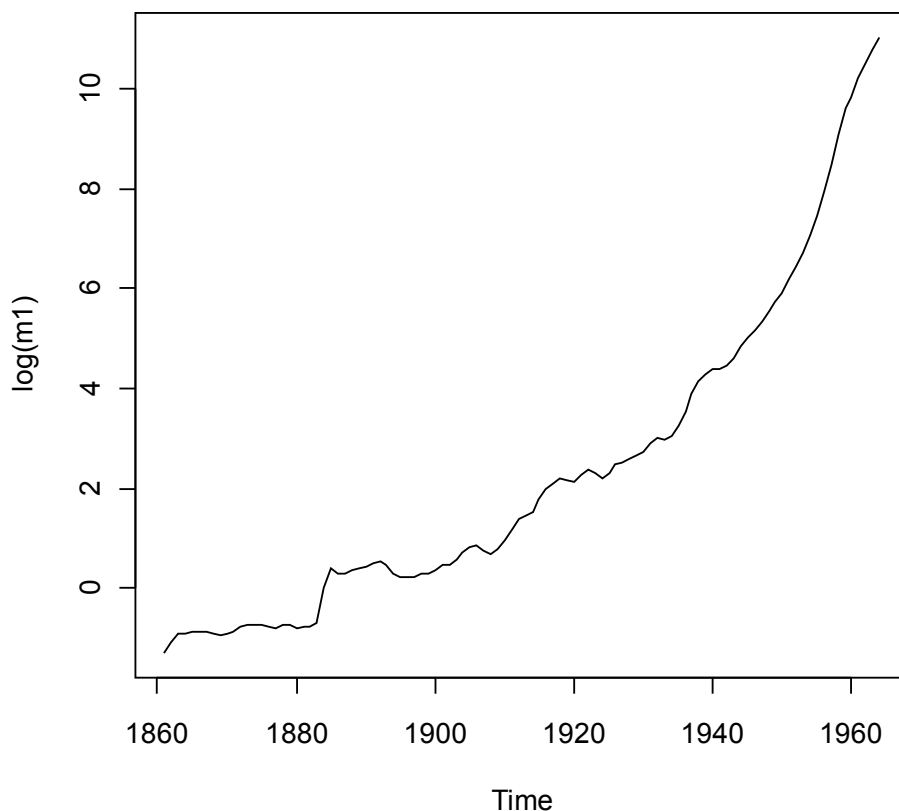
⁴ Uma lista que, aliás, cresce a cada dia...

(aparentemente) estacionária (ou estacionária de segunda ordem). Na metodologia original de Box-Jenkins, não havia um teste específico para isto e o procedimento acima é um bom exemplo de como se identificava o modelo.

Contudo, com o advento dos testes de raiz unitária, o estudo acerca da ordem de diferenciação das séries ficou bem mais interessante. Há várias formas de se fazer testes de raiz unitária no R. Nesta seção, veremos como fazer o mais popular teste de raiz unitária, o ADF (*Augmented Dickey-Fuller*). Para maiores referências, recomendo qualquer uma das referências bibliográficas. Um pacote popular – que contém vários testes de raiz unitária – é o pacote URCA. Uma vez que você o tenha instalado em seu computador, deverá “invocá-lo” para trabalhar com suas séries. O comando para isso é:

```
library(urca)
```

Vamos trabalhar com outras séries para este exemplo. Como sou um admirador dos bons trabalhos em história econômica, usarei uma série muito estudada pelo economista cubano (grande estudioso da economia brasileira) e pioneiro no uso da econometria de séries de tempo no Brasil, Carlos Manuel Peláez. Vejamos o gráfico da série escolhida (oferta de moeda), em escala logarítmica.



Vejamos como fazer um teste ADF no nível de $\log(m1)$ e na sua primeira diferença. Eis os comandos:

```

library(urca)

lm1.ct <- ur.df(lm1, type='trend', lags=10, selectlags=c("BIC"))
plot(lm1.ct)
summary(lm1.ct)

lm1.co<-ur.df(lm1, type='drift', lags=10, selectlags=c("BIC"))
plot(lm1.co)
summary(lm1.co)

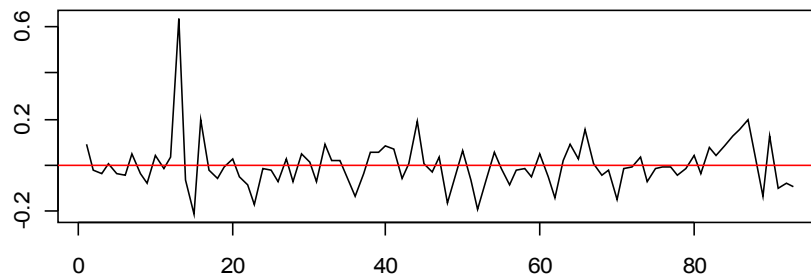
lm1diff<- diff(lm1)
lm1diff.ct<-ur.df(lm1, type='none', lags=10, selectlags=c("BIC"))
plot(lm1diff.ct)
summary(lm1diff.ct)

```

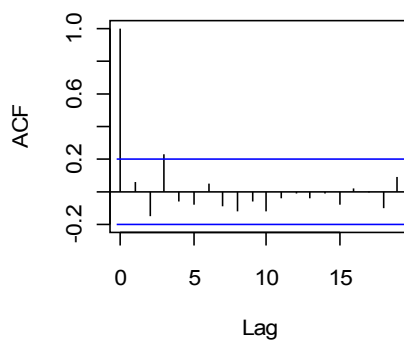
Note que, primeiramente, `lm1.ct` receberá o resultado do teste ADF, com constante e tendência (`type='trend'`) e a seleção de defasagens será feita pelo critério BIC, a partir de 10 defasagens. O segundo comando nos diz que `lm1.co` receberá o resultado do teste ADF, apenas com constante (`type='drift'`), com seleção similar de defasagens. Finalmente, criamos a série `lm1diff` como a diferença da série `lm1` e testamos a existência de raiz unitária nesta série.

Em todos os casos, pede-se o *plot* do teste e também o sumário (*summary*) dos resultados. Vejamos o primeiro teste.

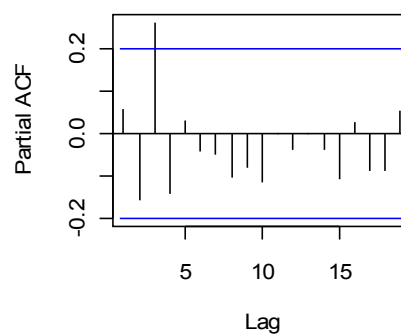
Residuals



Autocorrelations of Residuals



Partial Autocorrelations of Residuals



Além dos gráficos (*plot*), o sumário:

```
R Console
#####
# Augmented Dickey-Fuller Test Unit Root Test #
#####

Test regression trend

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

Residuals:
    Min       1Q   Median       3Q      Max
-0.21107 -0.05298 -0.01487  0.04148  0.63707

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.0335276  0.0464218   0.722  0.47206
z.lag.1      0.0218245  0.0124777   1.749  0.08376 .
tt          -0.0001233  0.0012148  -0.102  0.91938
z.diff.lag1  0.6745488  0.1036737   6.506 4.55e-09 ***
z.diff.lag2 -0.3247653  0.1089712  -2.980  0.00372 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1072 on 88 degrees of freedom
Multiple R-squared:  0.5776,    Adjusted R-squared:  0.5584
F-statistic: 30.09 on 4 and 88 DF,  p-value: 8.96e-16

Value of test-statistic is: 1.7491 8.2923 7.5837

Critical values for test statistics:
      1pct  5pct 10pct
tau3 -3.99 -3.43 -3.13
phi2  6.22  4.75  4.07
phi3  8.43  6.49  5.47
```

O leitor que já estudou o teste ADF reconhecerá os valores críticos encontrados para as três diferentes especificações de teste (τ_3 , ϕ_2 e ϕ_3). Antes de mais nada, vale a pena comentar que, em geral, pacotes como o URCA não são tão esotéricos assim: existe sempre uma documentação apropriada que pode ser obtida junto à página do projeto R. Assim, na página principal, em *search*, digite URCA.



Download, Packages

[CRAN](#)

R Project

[Foundation](#)

[Members & Donors](#)

[Mailing Lists](#)

[Bug Tracking](#)

[Developer Page](#)

[Conferences](#)

[Search](#)

Documentation

[Manuals](#)

[FAQs](#)

[The R Journal](#)

[Wiki](#)

[Books](#)

[Certification](#)

[Other](#)

Misc

[Bioconductor](#)

[Related Projects](#)

[User Groups](#)

[Links](#)

Obtém-se o seguinte resultado:

Google

urca

Search the Web Search r-project.org

About 746 results (0.19 seconds) [Advanced](#)

Everything

- Images
- Videos
- News
- Shopping
- More

▶ [CRAN - Package urca](#) ☆ 🔍
9 Sep 2010 ... **urca**: Unit root and cointegration tests for time series data. Unit root and cointegration tests encountered in applied econometric analysis ...
cran.r-project.org/web/packages/urca/index.html - [Cached](#) - [Similar](#)

[PDF] [Package 'urca'](#) ☆ 🔍
File Format: PDF/Adobe Acrobat - [Quick View](#)
9 Sep 2010 ... `ablrttest`, `alrttest`, `blrtest`, `ca.jo`, `ca.jo-class` and `urca-class`. `cajolst` Methods for Function plot in Package **urca**. Description ...
cran.r-project.org/web/packages/urca/urca.pdf - [Similar](#)

Note que o segundo *link* o leva direto ao manual de instruções do pacote URCA.

Claro, uma outra dica importante, para quem é da área de economia (geralmente o leitor desta apostila é tido como da área, com p-valor de 0.0023) é a página específica do projeto R para econometria. Eis o endereço: <http://cran.r-project.org/web/views/Econometrics.html> .

O que vem por aí?

Nas próximas versões desta apostila, veremos um aumento da seção sobre raiz unitária, bem como uma possível extensão da bibliografia.

Bibliografia

BUENO, R.de L. da S. *Econometria de Séries Temporais*. Thomson-Learning, 2008.
SHIKIDA, C.D. & FIGUEIREDO, E. A. *Notas de Aula de Econometria II – versão 0.1*, 2011.