



**PROGRAMA DE PÓS-GRADUAÇÃO
CURSO DE ESPECIALIZAÇÃO EM ESTATÍSTICA APLICADA**

SÉRIES TEMPORAIS

**SALVADOR
2019**

ANA PAULA DE SOUZA VANDERLEY

SÉRIES TEMPORAIS

Trabalho apresentado ao Curso de Especialização em Estatística Aplicada do Centro Universitário Jorge Amado, como requisito para a conclusão da disciplina Séries Temporais, sob a orientação do Profº Everaldo Guedes .

SALVADOR
2019

INTRODUÇÃO

Uma série temporal é uma coleção de observações feitas sequencialmente ao longo do tempo. Séries Temporais tem Movimentos característicos como tendência, sazonalidade, ciclo. Movimentos esses que podemos verificar através de uma decomposição das séries.

Nesse trabalho foram analisadas duas séries escolhidas a partir do banco de dados fornecidos em sala que foi o banco de dados gerais . as duas variáveis escolhidas foram minerais e celulose. Foi feita a decomposição dos dados para análise dos componentes principais como tendência, sazonalidade e ciclo das séries.

A tendência de uma série indica o seu comportamento “de longo prazo”, isto é, se ela cresce, decresce ou permanece estável, e qual a velocidade destas mudanças. Nos casos mais comuns trabalha-se com tendência constante, linear ou quadrática.

Os ciclos são caracterizados pelas oscilações de subida e de queda nas séries, de forma suave e repetida, ao longo da componente de tendência.

A sazonalidade em uma série corresponde às oscilações de subida e de queda que sempre ocorrem em um determinado período do ano, do mês, da semana ou do dia.

Testes foram aplicados com a finalidade de escolher o melhor modelo que explicasse as séries. Escolhido o modelo foi feita uma análise dos resíduos.

IMPORTANDO O BANCO DE DADOS

```
setwd("C:/Users/Bruno/Documents/ANA PAULA")
library(readxl)
serie2 <- read_excel("serie2.xlsx")
View(serie2)
```

CRIANDO DATAFRAME COM AS VARIAVÉIS ESCOLHIDAS

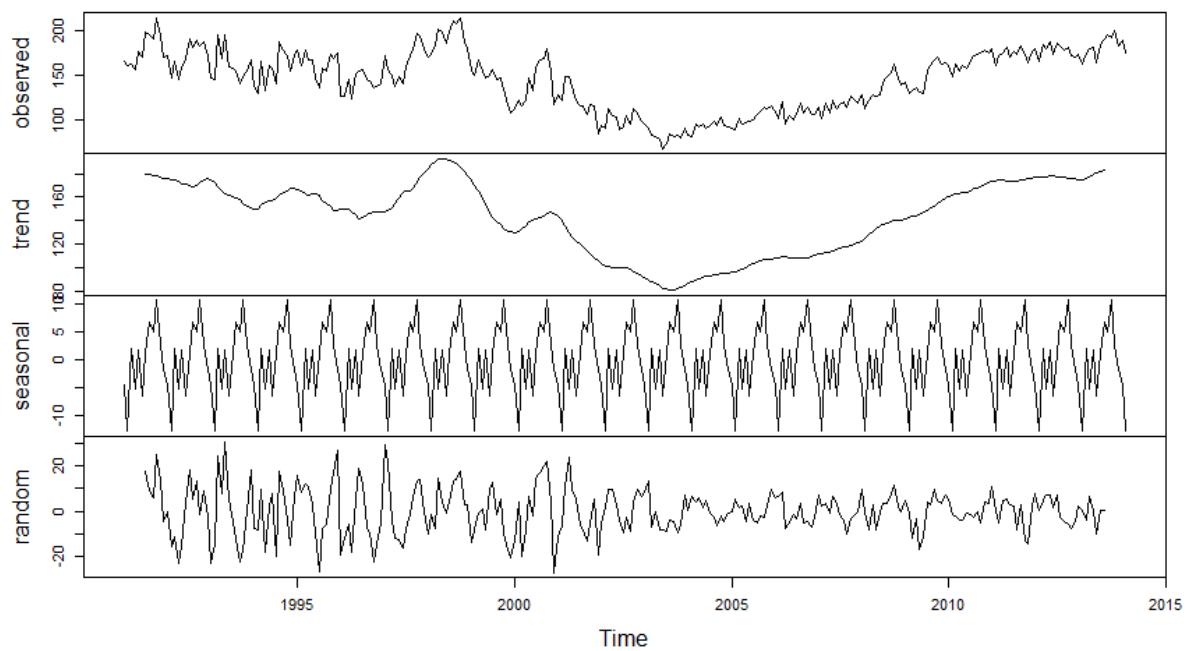
```
dataminerais <- ts(serie2$minerais,start=c(1991,01), frequency=12)
datacelulose <- ts(serie2$celulose,start=c(1991,01),frequency=12)
```

DECOMPOSIÇÃO

Através da decomposição clássica testamos os modelos aditivos e multiplicativos em cada série analisada a fim de verificar qual o melhor modelo. Verificamos através de análise dos gráficos que o melhor modelo para a variável “minerais” foi o multiplicativo pois foi o que apresentou uma menor amplitude.

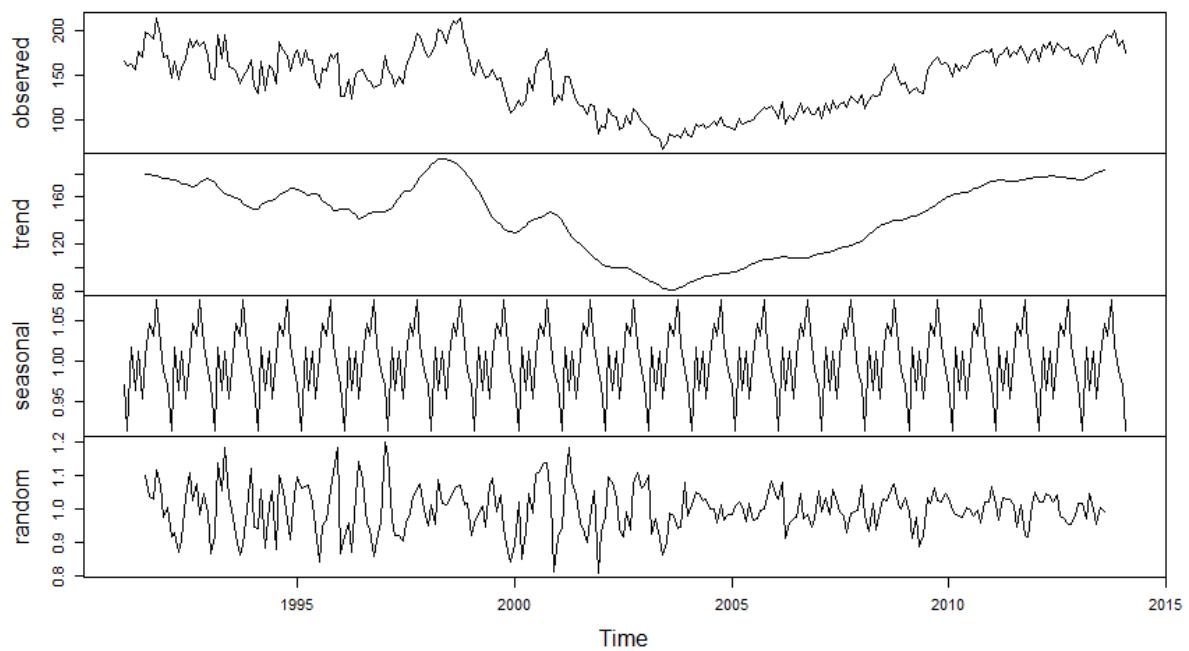
```
decomp1x <- decompose(dataminerais, type="additive")
View(decomp1x)
plot(decomp1x)
```

Decomposition of additive time series



```
decomp2x <- decompose(dataminerais, type="multiplicative")
view(decomp1x)
plot(decomp2x)
```

Decomposition of multiplicative time series



```

range(decomp1x$random, na.rm=TRUE)
[1] -27.08978 30.58054
range(decomp1x$random, na.rm=TRUE)
[1] -27.08978 30.58054
ATx1 = range(decomp1x$random, na.rm=TRUE)[2] - range(decomp1x$random,
na.rm=TRUE)[1]
print(ATx1)
[1] 57.67032
range(decomp2x$random, na.rm=TRUE)
[1] 0.8087073 1.2014739
ATx2 = range(decomp2x$random, na.rm=TRUE)[2] - range(decomp2x$random,
na.rm=TRUE)[1]
print(ATx2)
[1] 0.3927666

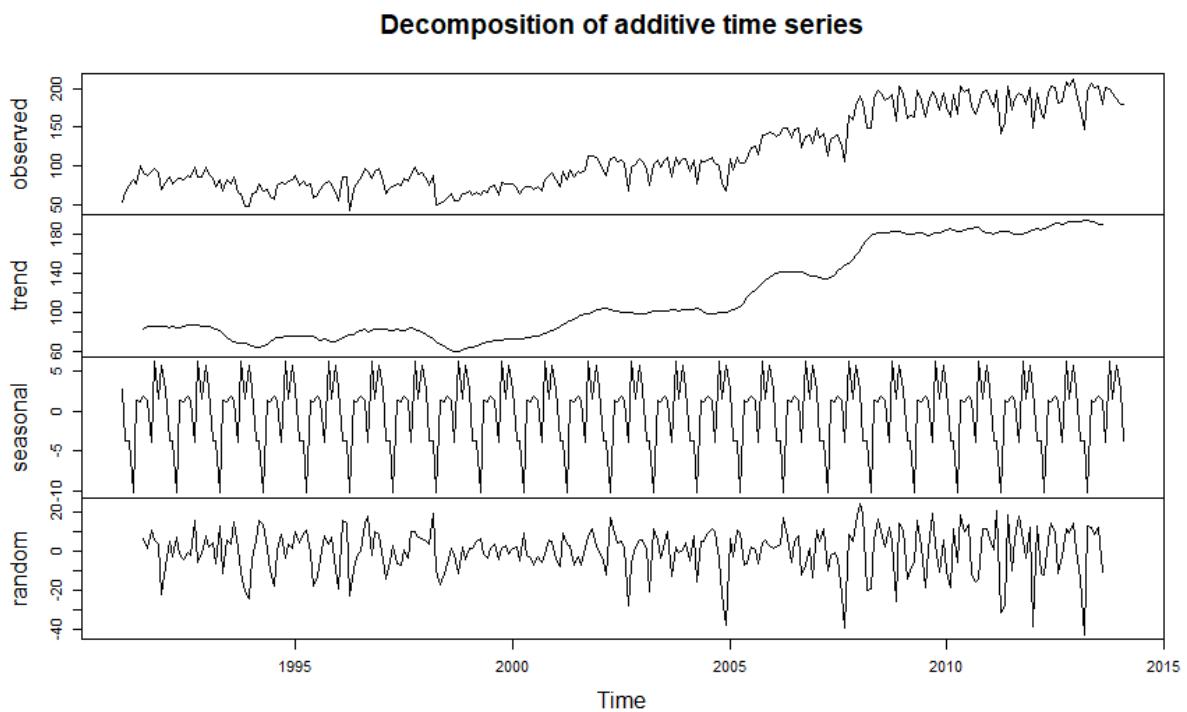
```

Já na decomposição para a variável “celulose” o melhor modelo foi o aditivo como podemos ver nos gráficos abaixo. A série celulose apresenta algumas sazonalidades e uma tendência crescente.

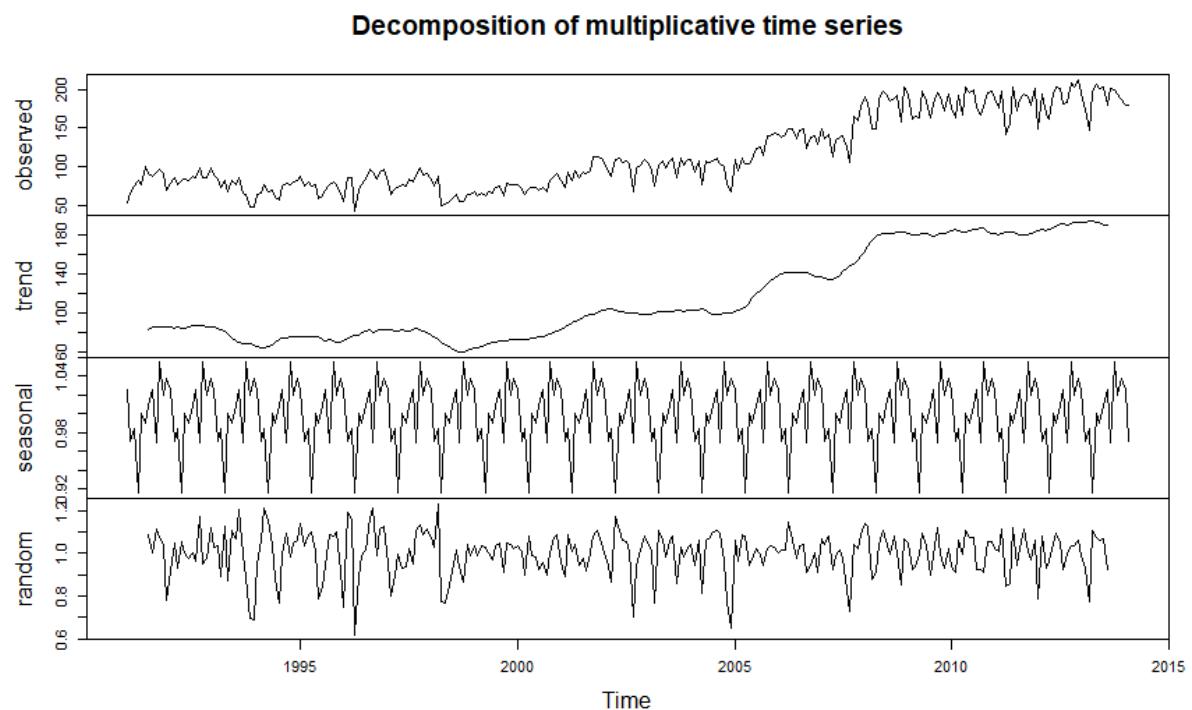
```

decomp1x <- decompose(datacelulose, type="additive")
plot(decomp1x)

```



```
decomp2x <- decompose(datacelulose, type="multiplicative")
plot(decomp2x)
```

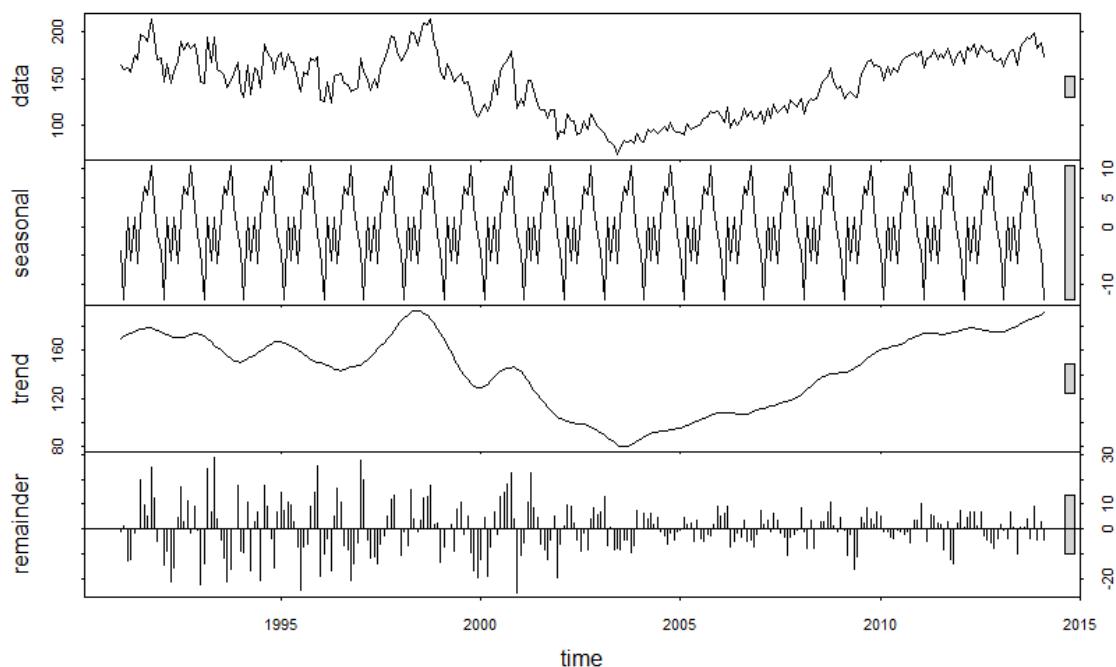


```
range(decomp1x$random, na.rm=TRUE)
[1] -42.72913 24.35183
ATx1 = range(decomp1x$random, na.rm=TRUE)[2] - range(decomp1x$random,
na.rm=TRUE)[1]
print(ATx1)
[1] 67.08097
range(decomp2x$random, na.rm=TRUE)
[1] 0.6185776 1.2314803
ATx2 = range(decomp2x$random, na.rm=TRUE)[2] - range(decomp2x$random,
na.rm=TRUE)[1]
print(ATx2)
[1] 0.6129027
```

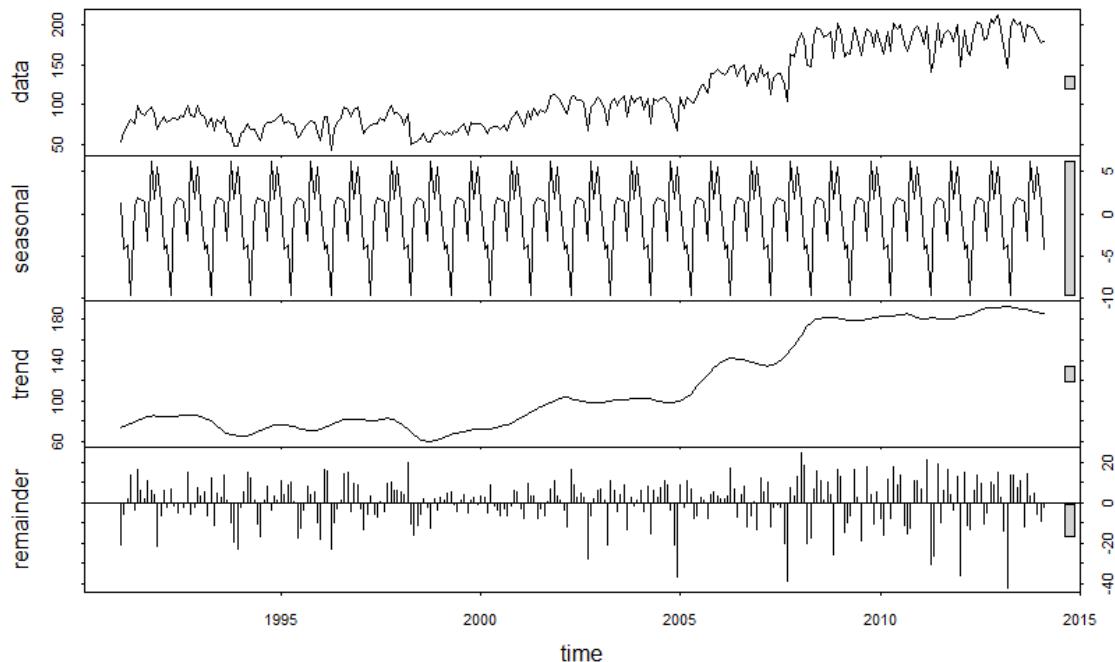
DECOMPOSIÇÃO STL

Utilizamos também a decomposição STL que decompõe a série com três componentes tendência, sazonalidade e restante. A decomposição STL corrige a tendência, regressão local a medida que há diferenças ele muda a tendência da série.

```
> plot(stl(dataminerais,s.window="periodic"))
```



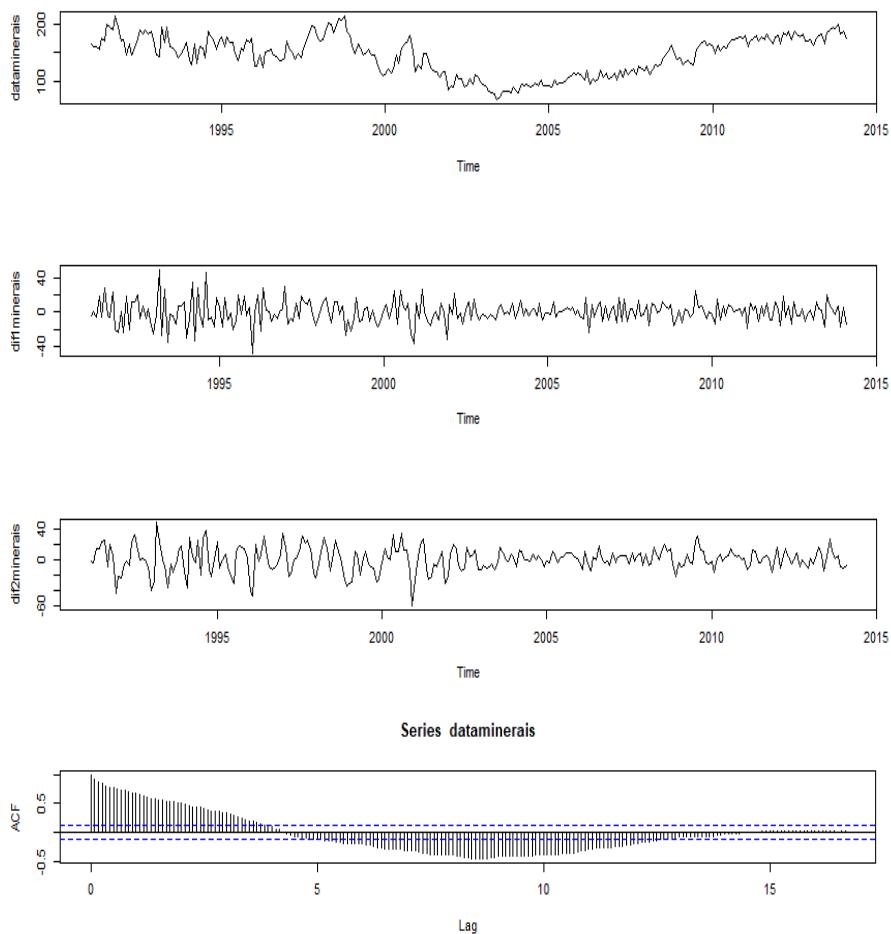
```
> plot(stl(datacelulose,s.window="periodic"))
```



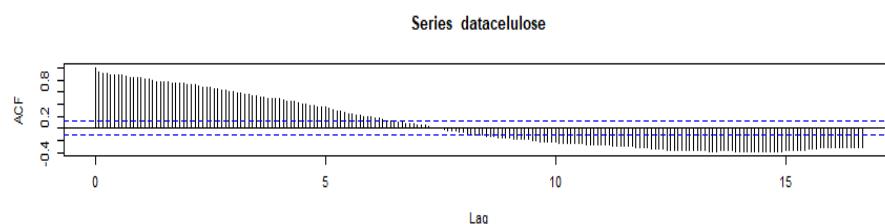
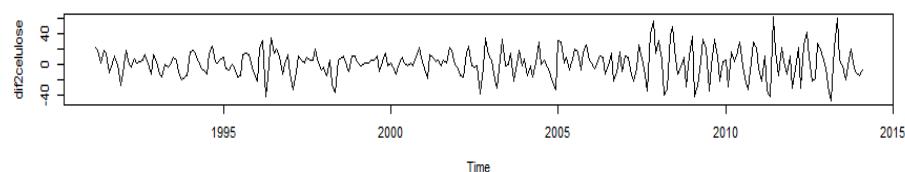
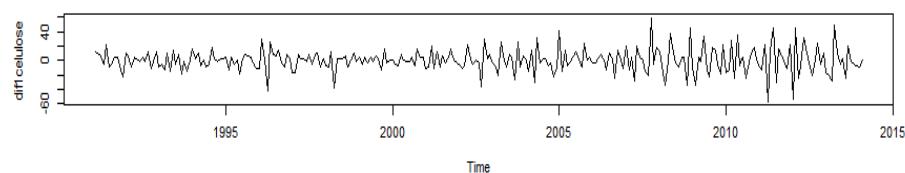
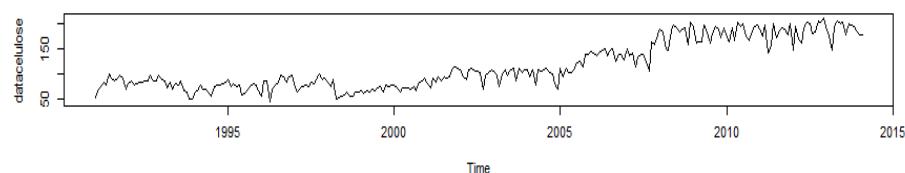
AUTOCORRELAÇÃO

Em séries temporais, é importante realizar uma análise da autocorrelação de uma série, para entender, principalmente sua aleatoriedade, já que várias técnicas partem dessa premissa. Autocorrelação é simplesmente a correlação entre uma série e ela mesma defasada. É a correlação entre os valores da série em um determinado período de tempo, e os valores da mesma série em um outro momento no tempo.

```
dif1minerais <- diff(dataminerais, lag = 1)
dif2minerais <- diff(dataminerais, lag = 2)
par(mfrow=c(3,1))
plot(dataminerais)
plot(dif1minerais)
plot(dif2minerais)
acf(dataminerais, lag.max=200)
```



```
dif1celulose <- diff(datacelulose, lag = 1)
dif2celulose <- diff(datacelulose, lag = 2)
par(mfrow=c(3,1))
plot(datacelulose)
plot(dif1celulose)
plot(dif2celulose)
acf(datacelulose, lag.max=200)
```



INSTALANDO PACOTES NECESSÁRIOS

```
library(timeDate)
library(timeSeries)
library(fBasics)
```

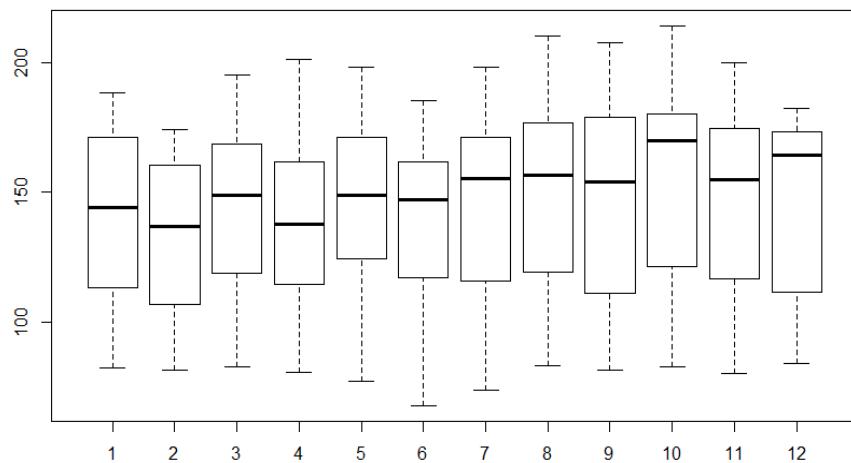
SÚMARIO

```
basicStats(dataminerais)
```

	dataminerais
nobs	278.000000
NAs	0.000000
Minimum	67.740000
Maximum	214.250000
1. Quartile	115.667500
3. Quartile	172.130000
Mean	144.585288
Median	149.120000
Sum	40194.710000
SE Mean	2.038924
LCL Mean	140.571533
UCL Mean	148.599042
Variance	1155.704756
Stdev	33.995658
Skewness	-0.244255
Kurtosis	-1.004602

O boxplot verifica o centro dos dados (a média ou mediana), a amplitude dos dados (máximo – mínimo), a simetria ou assimetria do conjunto de dados e a presença de outliers. Podemos notar nos gráficos abaixo que a variável celulose apresentou uma maior amplitude ou seja uma maior variação dos dados. Já a variável minerais apresentou maior assimetria .

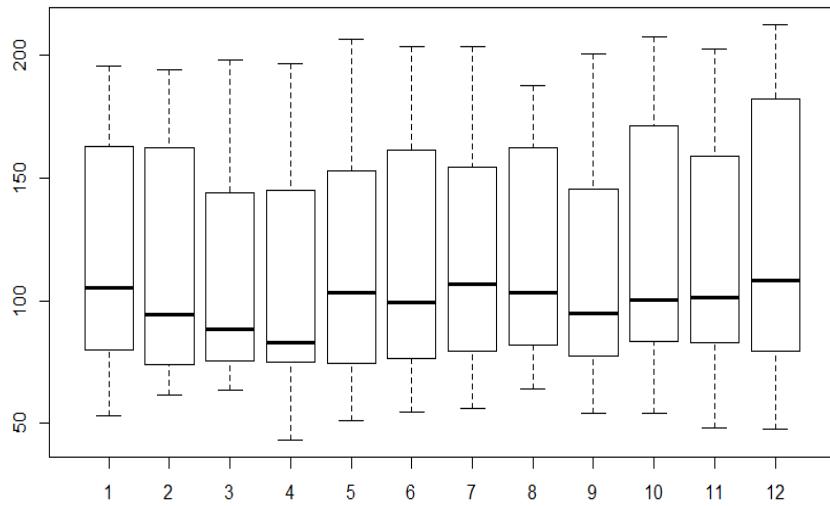
```
boxplot(dataminerais~cycle(dataminerais)
```



```
basicStats(datacelulose)
```

	datacelulose
nobs	278.000000
NAs	0.000000
Minimum	43.190000
Maximum	212.560000
1. Quartile	76.525000
3. Quartile	159.095000
Mean	115.597842
Median	98.260000
Sum	32136.200000
SE Mean	2.855822
LCL Mean	109.975971
UCL Mean	121.219712
Variance	2267.289465
Stdev	47.616063
Skewness	0.591243
Kurtosis	-1.077893

```
boxplot(datacelulose~cycle(datacelulose))
```



NORMALIDADE

Os testes de normalidade são utilizados para verificar se a distribuição de probabilidade associada a um conjunto de dados pode ser aproximada pela distribuição normal. Foram usados o teste de Jarque-Bera e Shapiro Wilk.

O teste Jarque-Bera utiliza como parâmetros os coeficientes de curtose e assimetria (que na normal são de 3 e 0, respectivamente)

O teste de Jarque-Bera tem como hipótese nula a normalidade. Assim, se o p-valor for menor do que 5% (ou 10%), $p < 0,05$ ($p < 0,10$), então o autor rejeita a normalidade. Já se $p > 0,05$, aceita-se a normalidade.

O valor de p é uma probabilidade que mede a evidência contra a hipótese nula. Um valor de p menor fornece uma evidência mais forte contra a hipótese nula.

```
jarqueberaTest(dataminerais)
```

Title:
Jarque - Bera Normalality Test

Test Results:
STATISTIC:

```

X-squared: 14.151
P VALUE:
Asymptotic p value: 0.0008456

Description:
Wed Oct 02 15:54:14 2019 by user: Bruno

```

p<0,05 (p<0,10), então rejeita a normalidade.

```

> shapiro.test(dataminerais)

Shapiro-wilk normality test

data: dataminerais
W = 0.96429, p-value = 2.234e-06

```

Podemos afirmar com nível de significância de 5% que a amostra provém de uma população normal pois o p valor deu > que 0,05.

```

> jarqueberaTest(datacelulose)

Title:
Jarque - Bera Normality Test

Test Results:
STATISTIC:
X-squared: 29.4859
P VALUE:
Asymptotic p value: 3.956e-07

Description:
Wed Oct 02 15:54:49 2019 by user: Bruno

```

p>0,05, aceita-se a normalidade.

```

> shapiro.test(datacelulose)

Shapiro-wilk normality test

data: datacelulose
W = 0.89029, p-value = 2.735e-13

```

Podemos afirmar com nível de significância de 5% que a amostra provém de uma população normal pois o p valor deu > que 0,05.

ESTACIONARIEDADE DA SÉRIE

A estacionariedade numa série temporal significa que os dados oscilam sobre uma média constante, independente do tempo, com a variância das flutuações permanecendo essencialmente a mesma.

```
> library(urca)

ur1dataminerais <- ur.df(dataminerais, lags=5, type="none",
selectlags="AIC")
> ur2dataminerais <- ur.df(dataminerais, lags=5, type="drift",
selectlags="AIC")
> ur3dataminerais <- ur.df(dataminerais, lags=5, type="trend",
selectlags="AIC")

> summary(ur1dataminerais)

#####
# 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 
-45.25   -7.55    0.84    8.27   49.36 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
z.lag.1     -0.001621  0.005194 -0.312 0.755214    
z.diff.lag1 -0.314085  0.059925 -5.241 3.24e-07 ***  
z.diff.lag2 -0.027087  0.062537 -0.433 0.665265    
z.diff.lag3 -0.083496  0.062696 -1.332 0.184076    
z.diff.lag4 -0.218484  0.059842 -3.651 0.000314 ***  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 12.63 on 267 degrees of freedom
Multiple R-squared:  0.1377, Adjusted R-squared:  0.1216 
F-statistic: 8.528 on 5 and 267 DF,  p-value: 1.656e-07

value of test-statistic is: -0.3121

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

```

summary(ur2dataminerais)

#####
# 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 
-44.704 -8.233 -0.128  8.201 49.961 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 6.361304   3.491107   1.822 0.069556 .  
z.lag.1    -0.043711   0.023671  -1.847 0.065915 .  
z.diff.lag1 -0.286949   0.061497  -4.666 4.87e-06 *** 
z.diff.lag2 -0.004665   0.063472  -0.073 0.941470  
z.diff.lag3 -0.061341   0.063598  -0.965 0.335671  
z.diff.lag4 -0.201815   0.060282  -3.348 0.000932 *** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 12.58 on 266 degrees of freedom
Multiple R-squared:  0.1483, Adjusted R-squared:  0.1323 
F-statistic: 9.266 on 5 and 266 DF,  p-value: 3.789e-08

```

value of test-statistic is: -1.8466 1.7092

Critical values for test statistics:

	1pct	5pct	10pct
tau2	-3.44	-2.87	-2.57
phi1	6.47	4.61	3.79

```
summary(ur3dataminerais)
```

```

#####
# 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 
-44.233 -8.051 -0.151  8.060 50.517 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 5.240734   3.973254   1.319 0.188307 .  
z.lag.1    -0.041670   0.023949  -1.740 0.083027 .  
tt         0.005848   0.009859   0.593 0.553593 
```

```

z.diff.lag1 -0.289959  0.061781  -4.693  4.31e-06 ***
z.diff.lag2 -0.007622  0.063745  -0.120  0.904915
z.diff.lag3 -0.064558  0.063907  -1.010  0.313325
z.diff.lag4 -0.204251  0.060495  -3.376  0.000845 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

```

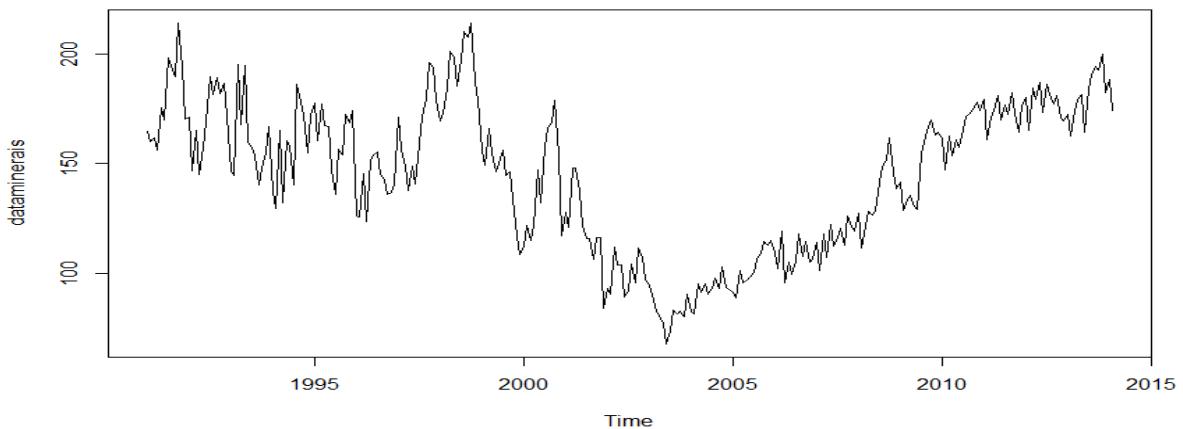
Residual standard error: 12.59 on 265 degrees of freedom
 Multiple R-squared: 0.1495, Adjusted R-squared: 0.1302
 F-statistic: 7.761 on 6 and 265 DF, p-value: 1.056e-07

value of test-statistic is: -1.74 1.254 1.8767

Critical values for test statistics:

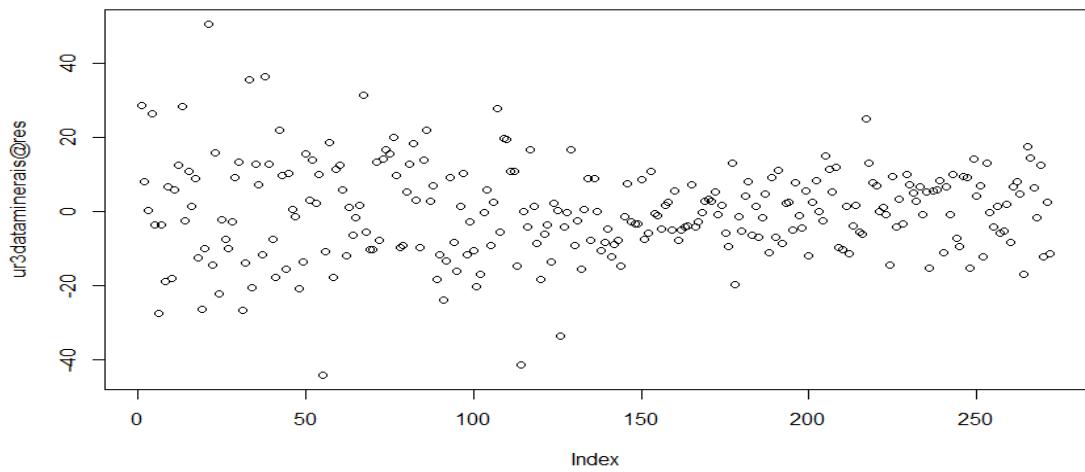
	1pct	5pct	10pct
tau3	-3.98	-3.42	-3.13
phi2	6.15	4.71	4.05
phi3	8.34	6.30	5.36

> plot(dataminerais)

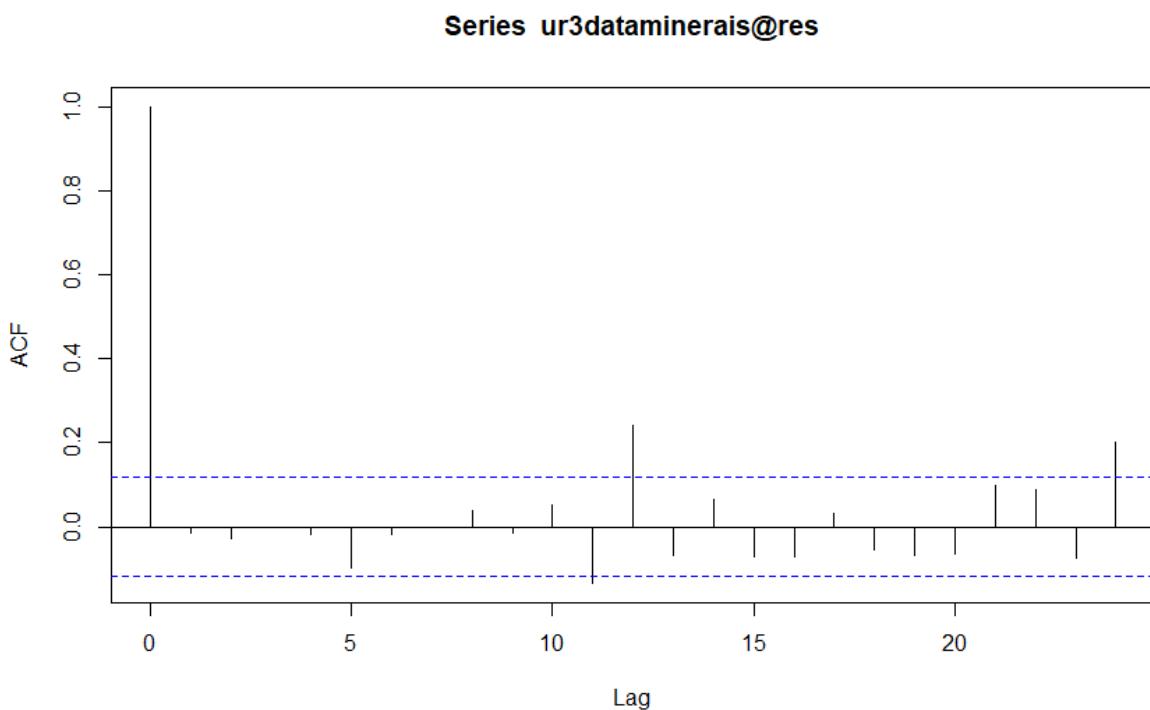


Podemos notar que na variável minerais não tem uma tendência definida .

> plot(ur3dataminerais\$res)



```
> acf(ur3dataminerais@res)
```



No gráfico ACF, o eixo vertical indica a autocorrelação e o horizontal a defasagem. A linha tracejada azul indica onde é significativamente diferente de

zero. Como é possível ver na imagem acima, praticamente todos os valores ACF estão dentro do limite da linha tracejada azul. Ou seja, autocorrelação igual a zero, indicando que a série é aleatória.

```

ur1datacelulose <- ur.df(datacelulose, lags=5, type="none",
selectlags="AIC")
> ur2datacelulose <- ur.df(datacelulose, lags=5, type="drift",
selectlags="AIC")
> ur3datacelulose <- ur.df(datacelulose, lags=5, type="trend",
selectlags="AIC")

> summary(ur1datacelulose)

#####
# 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 
-51.792  -7.084   1.916   8.762  45.636 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
z.lag.1     0.006349  0.006912  0.919   0.3592    
z.diff.lag1 -0.549019  0.061264 -8.962  < 2e-16 ***  
z.diff.lag2 -0.430420  0.067733 -6.355 9.02e-10 ***  
z.diff.lag3 -0.309954  0.070051 -4.425 1.41e-05 ***  
z.diff.lag4 -0.284975  0.067419 -4.227 3.26e-05 ***  
z.diff.lag5 -0.114722  0.060703 -1.890  0.0599 .  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 14.01 on 266 degrees of freedom
Multiple R-squared:  0.2533, Adjusted R-squared:  0.2365 
F-statistic: 15.04 on 6 and 266 DF,  p-value: 8.13e-15

value of test-statistic is: 0.9185

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

summary(ur2datacelulose)

```

```
#####
# 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 
-50.967 -7.507  1.611  8.163 45.491 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 1.95651   2.30466   0.849   0.3967  
z.lag.1    -0.00833   0.01862  -0.447   0.6550  
z.diff.lag1 -0.53763   0.06275  -8.568 8.78e-16 *** 
z.diff.lag2 -0.42118   0.06864  -6.136 3.06e-09 *** 
z.diff.lag3 -0.30272   0.07060  -4.288 2.53e-05 *** 
z.diff.lag4 -0.27980   0.06773  -4.131 4.84e-05 *** 
z.diff.lag5 -0.11190   0.06083  -1.840   0.0669 .  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 14.01 on 265 degrees of freedom
Multiple R-squared:  0.2551, Adjusted R-squared:  0.2382 
F-statistic: 15.12 on 6 and 265 DF,  p-value: 6.905e-15

value of test-statistic is: -0.4473 0.7817

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

```
summary(ur3datacelulose)

#####
# 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 
-50.198 -7.114  1.511  8.653 42.487 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 1.95651   2.30466   0.849   0.3967  
z.lag.1    -0.00833   0.01862  -0.447   0.6550  
tt          0.00000   0.00000  -0.000   1.0000  
z.diff.lag1 -0.53763   0.06275  -8.568 8.78e-16 *** 
z.diff.lag2 -0.42118   0.06864  -6.136 3.06e-09 *** 
z.diff.lag3 -0.30272   0.07060  -4.288 2.53e-05 *** 
z.diff.lag4 -0.27980   0.06773  -4.131 4.84e-05 *** 
z.diff.lag5 -0.11190   0.06083  -1.840   0.0669 .  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
(Intercept) 4.18850 2.43781 1.718 0.086943 .
z.lag.1 -0.10422 0.04137 -2.519 0.012351 *
tt 0.06233 0.02408 2.589 0.010163 *
z.diff.lag1 -0.46594 0.06798 -6.854 5.05e-11 ***
z.diff.lag2 -0.36600 0.07118 -5.142 5.30e-07 ***
z.diff.lag3 -0.26077 0.07171 -3.636 0.000332 ***
z.diff.lag4 -0.24969 0.06801 -3.671 0.000292 ***
z.diff.lag5 -0.09484 0.06054 -1.567 0.118426
---
Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

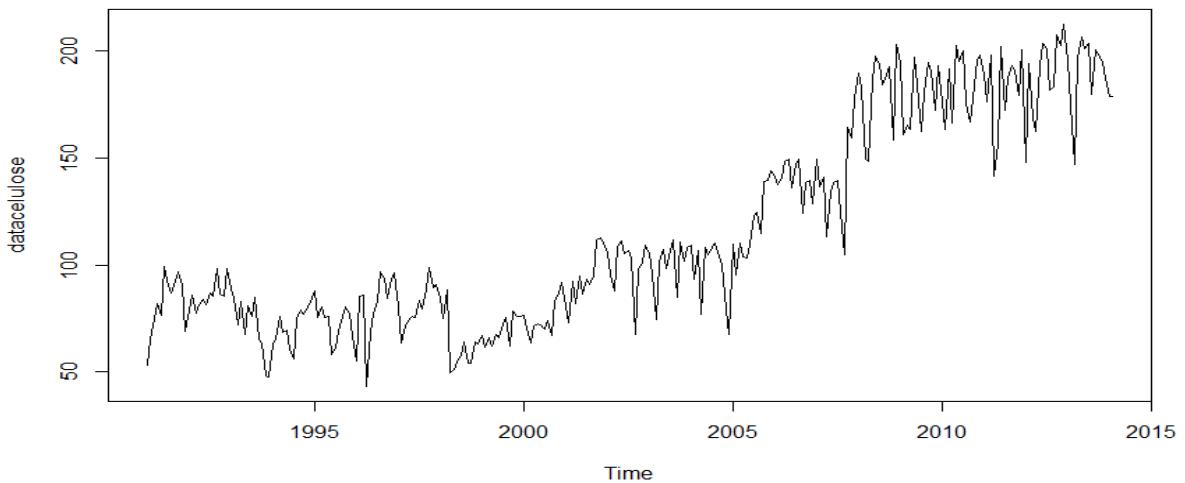
Residual standard error: 13.86 on 264 degrees of freedom
 Multiple R-squared: 0.2735, Adjusted R-squared: 0.2543
 F-statistic: 14.2 on 7 and 264 DF, p-value: 1.227e-15

Value of test-statistic is: -2.5193 2.7664 3.4533

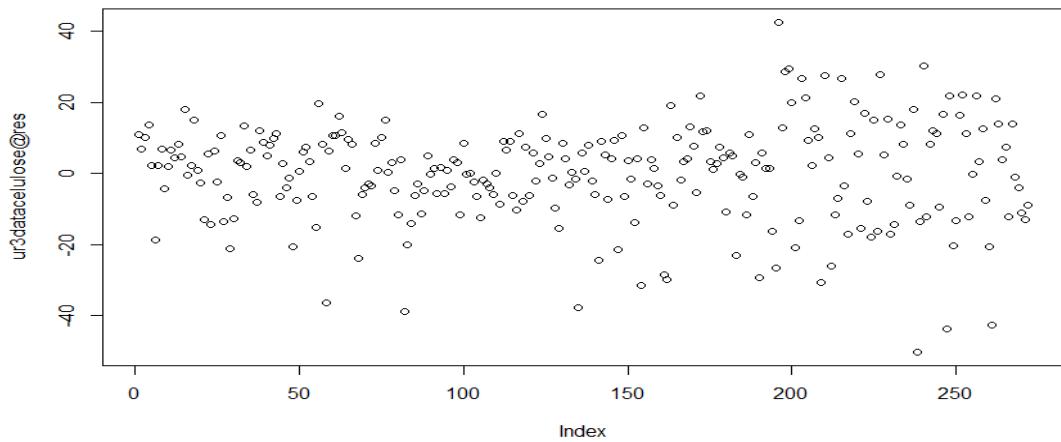
Critical values for test statistics:

	1pct	5pct	10pct
tau3	-3.98	-3.42	-3.13
phi2	6.15	4.71	4.05
phi3	8.34	6.30	5.36

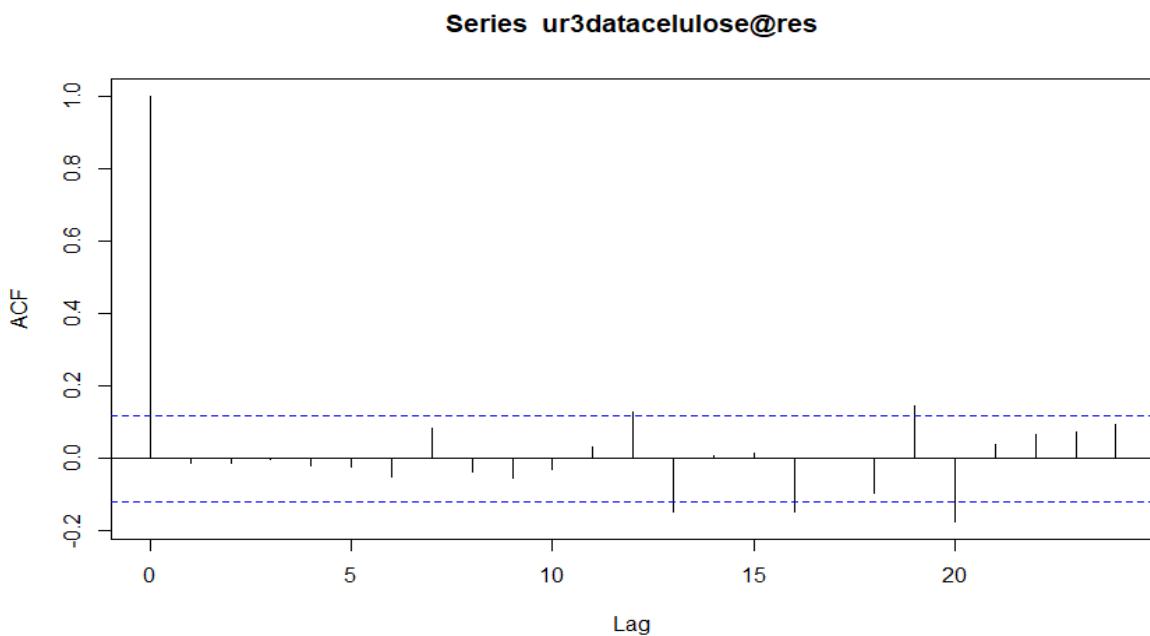
> plot(datacelulose)



> plot(ur3datacelulose@res)



```
> acf(ur3datacelulose@res)
```



Autocorrelação igual a zero, indicando que a série é aleatória.

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

```
library(tseries)
```

A estatística ADF, usada no teste, é um número negativo, e quanto mais negativo, mais indicativo o teste se torna de rejeitar a hipótese nula.

Nesse caso podemos verificar nos valores do teste abaixo que a variável celulose apresentou um valor negativo maior ou seja tem um índice maior de rejeitar-se a hipótese nula.

```
adf.test(dataminerais)
```

```
Augmented Dickey-Fuller Test
```

```
data: dataminerais
Dickey-Fuller = -1.4576, Lag order = 6, p-value = 0.8046
alternative hypothesis: stationary
```

```
> adf.test(datacelulose)
```

```
Augmented Dickey-Fuller Test
```

```
data: datacelulose
Dickey-Fuller = -2.4082, Lag order = 6, p-value = 0.4041
alternative hypothesis: stationary
```

```
> install.packages("nonlinearTseries")
```

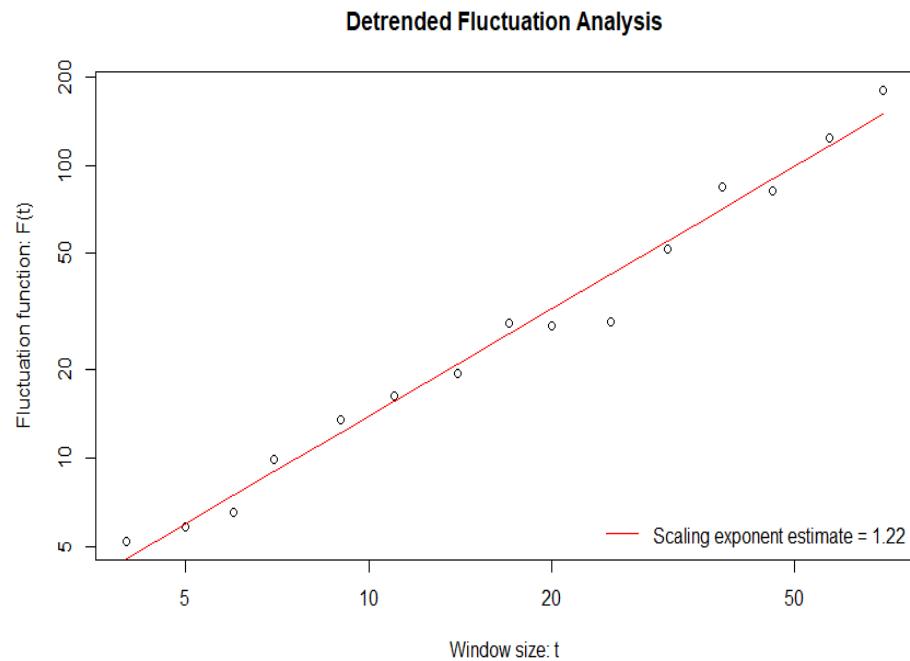
TESTE DFA

```
install.packages("nonlinearTseries")
```

```
> par(mfrow=c(2,1))
> plot(dataminerais)
> acf(dataminerais, lag.max=70)
```

```
> DFAdataminerais <- dfa(dataminerais,
window.size.range=c(4,length(dataminerais)/4), npoints=15, do.plot=FALSE)
```

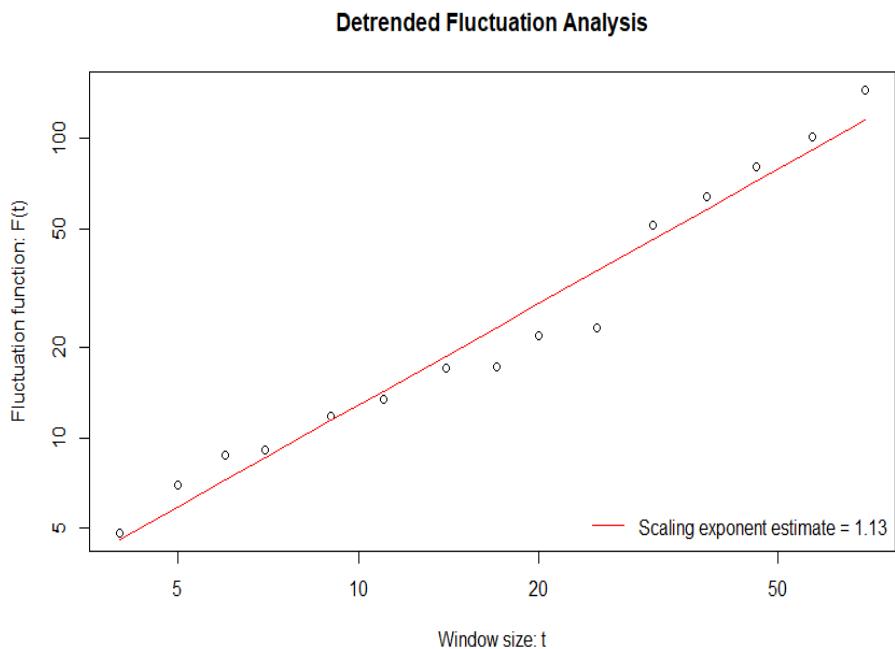
```
> expoentdataminerais <- estimate(DFAdataminerais, do.plot=T)
```



```
> print(expoentdataminerais)
```

```
[1] 1.221455
attr(,"fitted")
attr(,"fitted")$`x`
[1] 4 5 6 7 9 11 14 17 20 25 31 38 46 57 70
attr(,"fitted")$y
      1      2      3      4      5      6      7      8      9      10      11 
 4.535224 5.956210 7.441943 8.983775 12.211639 15.603571 20.948531 26.555090 32.386157 42.533456 55.314768 
      12      13      14      15 
70.932375 89.576447 116.394410 149.594031
```

```
> DFAdatacelulose <- dfa(datacelulose,
window.size.range=c(4,length(datacelulose)/4), npoints=15, do.plot=FALSE)
> expoentedatacelulose <- estimate(DFAdatacelulose, do.plot=T)
```



```
> print(expoentdatacelulose)
```

```
[1] 1.12/80s
attr(,"fitted")
attr(,"fitted")$`x`
[1] 4 5 6 7 9 11 14 17 20 25 31 38 46 57 70

attr(,"fitted")$y
      1      2      3      4      5      6      7      8      9      10     11 
 4.576814 5.886599 7.230528 8.603533 11.422911 14.324195 18.801708 23.404517 28.112887 36.158186 46.086472 
     12     13     14     15 
57.983072 71.925810 91.602638 115.488696
```

MODELOS DE PREVISÃO

Os modelos ARIMA são modelos estatísticos lineares para análise de séries temporais. Um modelo auto-regressivo integrado de médias móveis. Os termos auto-regressivos correspondem a defasagens da série transformada (isto é, série estacionária obtida por diferenciação) e as médias móveis a defasagens dos erros aleatórios.

```
library(forecast)
> modeldataminerais<-auto.arima(dataminerais)
> summary(modeldataminerais)
```

Series: dataminerais
ARIMA(5,1,1)(2,0,0)[12]

Coefficients:

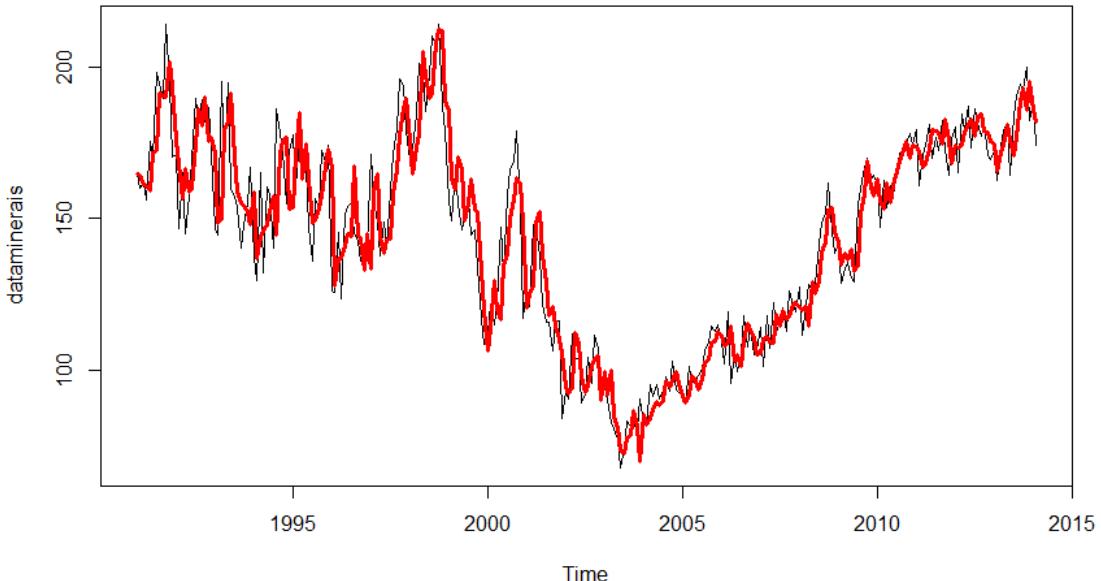
	ar1	ar2	ar3	ar4	ar5	ma1	sar1	sar2
s.e.	0.0377	-0.0014	-0.0520	-0.1943	-0.0607	-0.3076	0.2337	0.2001
	0.3512	0.1053	0.0637	0.0646	0.0986	0.3505	0.0618	0.0620

sigma^2 estimated as 141.1: log likelihood=-1075.63
AIC=2169.25 AICc=2169.93 BIC=2201.87

Training set error measures:

	ME	RMSE	MAE	MPE	MAPE	MASE
ACF1						
Training set	0.107895	11.68406	8.778594	-0.3299551	6.132001	0.4592913
	-0.0014267					
> accuracy(modellataminerais\$fitted, dataminerais)						
	ME	RMSE	MAE	MPE	MAPE	ACF1 Theil's
U						
Test set	0.107895	11.68406	8.778594	-0.3299551	6.132001	-0.0014267
	0.8615679					

```
> moddataminerais <- arima(dataminerais, order = c(5,1,1), seasonal =
  list(order = c(2,0,0)))
> preditodataminerais <- fitted(moddataminerais)
> plot(dataminerais)
> lines(preditodataminerais, col="red", lwd=3)
```



```

> modeldatacelulose<-auto.arima(datacelulose)
> summary(modeldatacelulose)
Series: datacelulose
ARIMA(1,1,1)(0,0,2)[12]

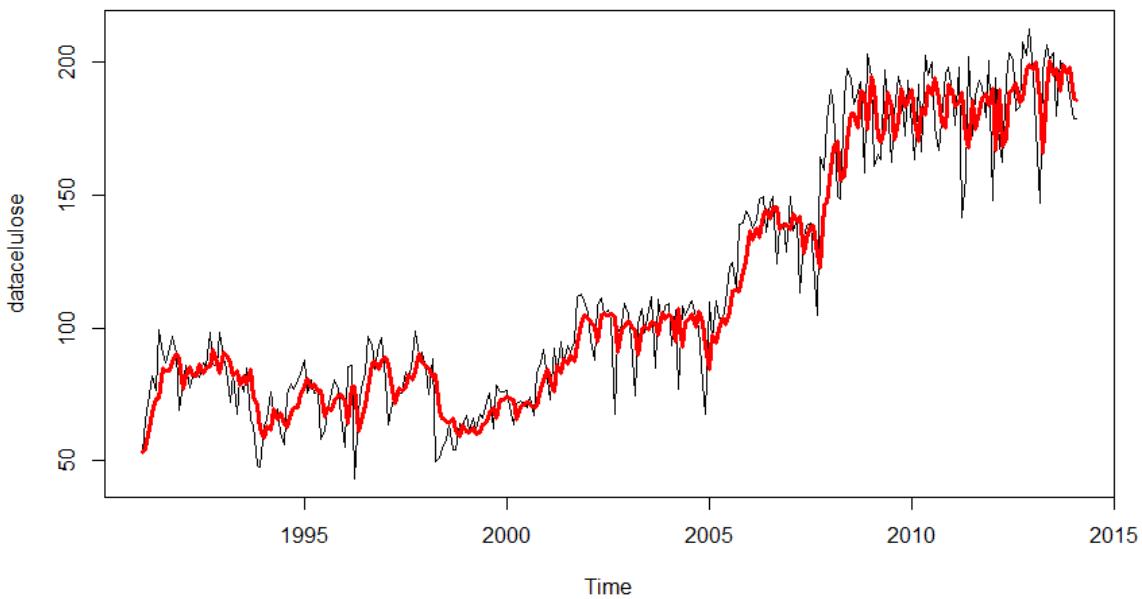
Coefficients:
            ar1      ma1     sma1     sma2
            0.2295 -0.7670  0.1295  0.1012
s.e.    0.0933  0.0636  0.0624  0.0578

sigma^2 estimated as 192:  log likelihood=-1119.65
AIC=2249.29  AICc=2249.52  BIC=2267.41

Training set error measures:
               ME      RMSE      MAE      MPE      MAPE      MASE
ACF1
Training set 1.23426 13.72995 10.41738 -0.3227058 9.998286 0.6231079 -
0.001624971
> accuracy(modeldatacelulose$fitted, datacelulose)
               ME      RMSE      MAE      MPE      MAPE      ACF1 Theil's
U
Test set 1.23426 13.72995 10.41738 -0.3227058 9.998286 -0.001624971
0.8583306

> moddatacelulose <- arima(datacelulose, order = c(1,1,1), seasonal =
list(order = c(0,0,2)))
> preditodatacelulose <- fitted(moddatacelulose)
> plot(datacelulose)
> lines(preditodatacelulose, col="red", lwd=3)

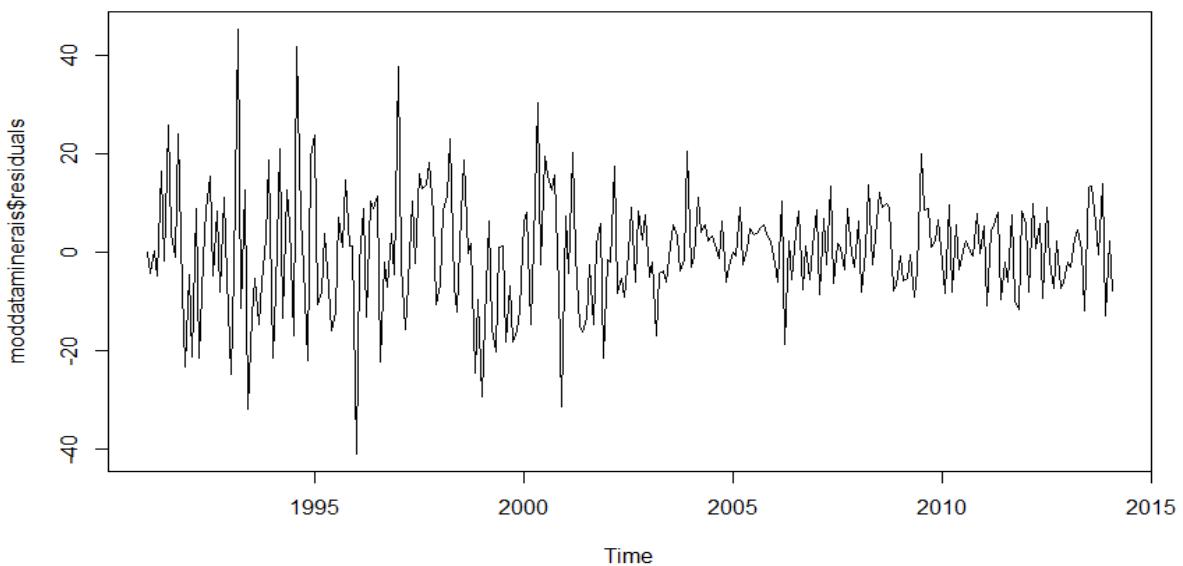
```



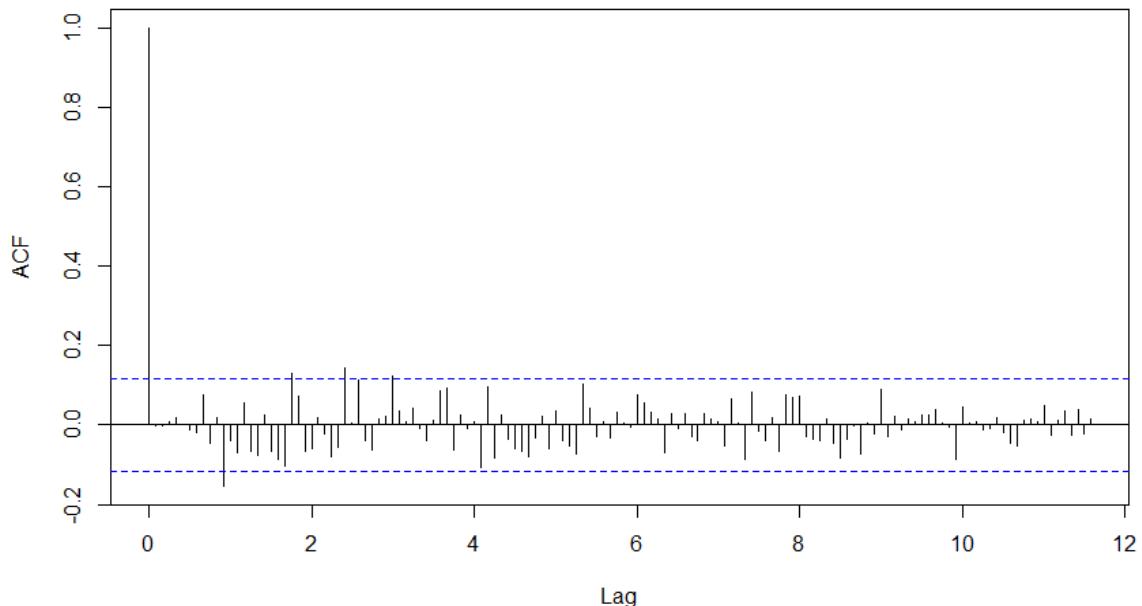
RESIDUOS

Com relação ao ACF, uma autocorrelação positiva é identificada por um agrupamento de resíduos com o mesmo sinal; a negativa é identificada por rápidas mudanças nos sinais de resíduos consecutivos. O ideal é todos os lags (exceto 0, que sempre vai ser igual a 1) tenham valores dentro do intervalo de confiança, ou seja próximos de zero, e sem tendências.

```
> plot(moddataminerais$residuals)
> acf(moddataminerais$residuals, lag.max=(length(dataminerais)/2))
```

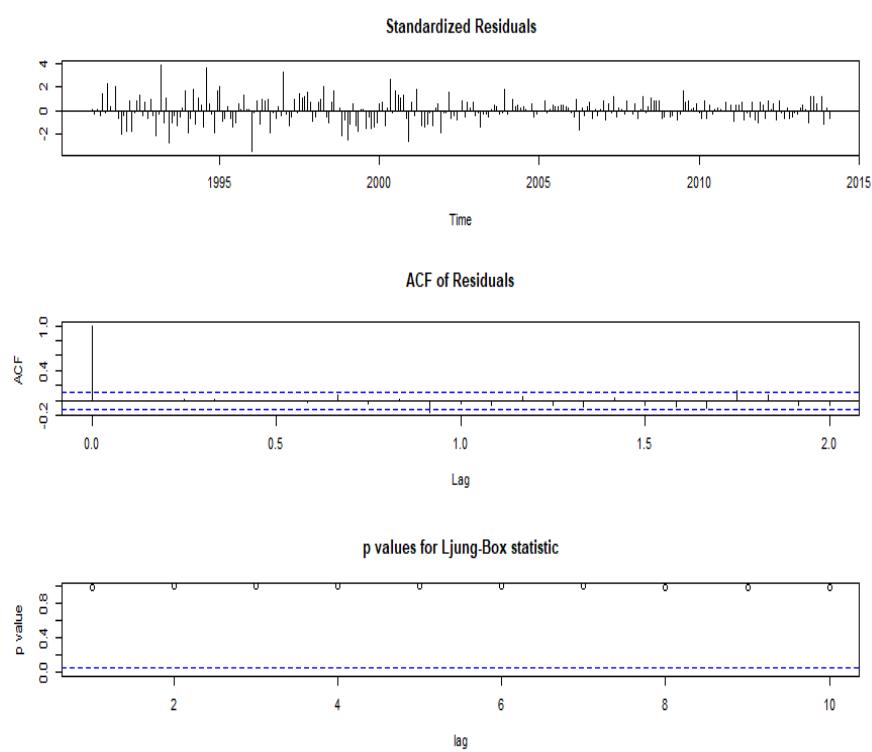


Series moddataminerais\$residuals

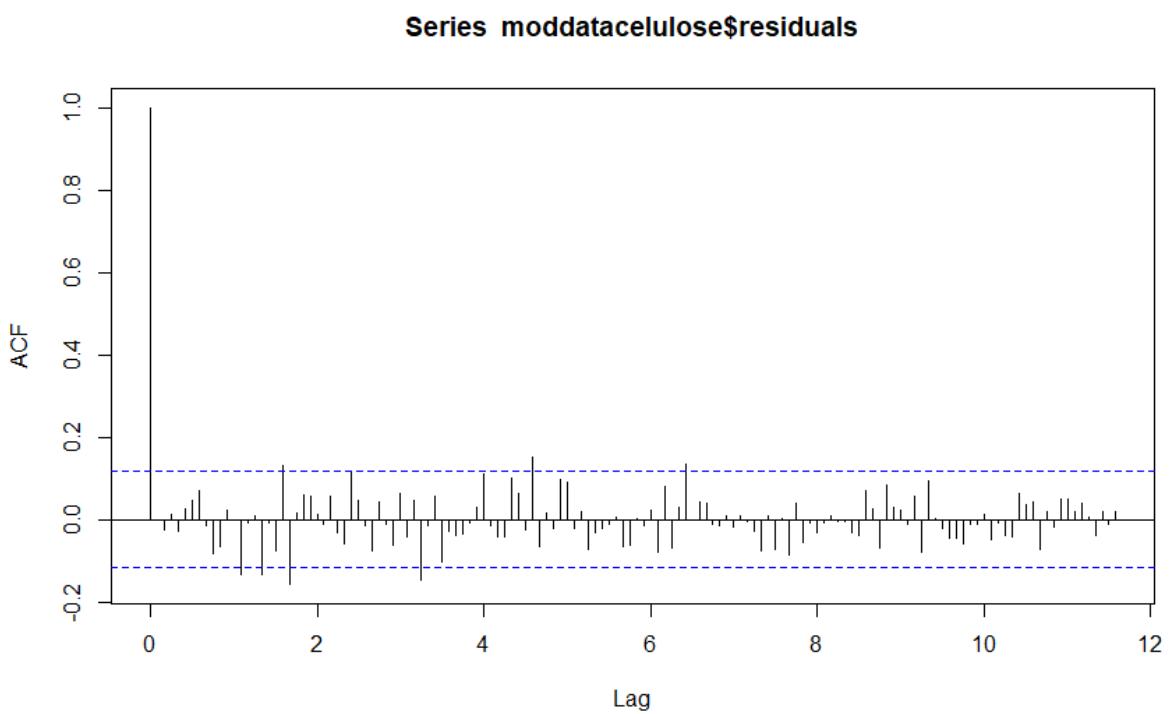
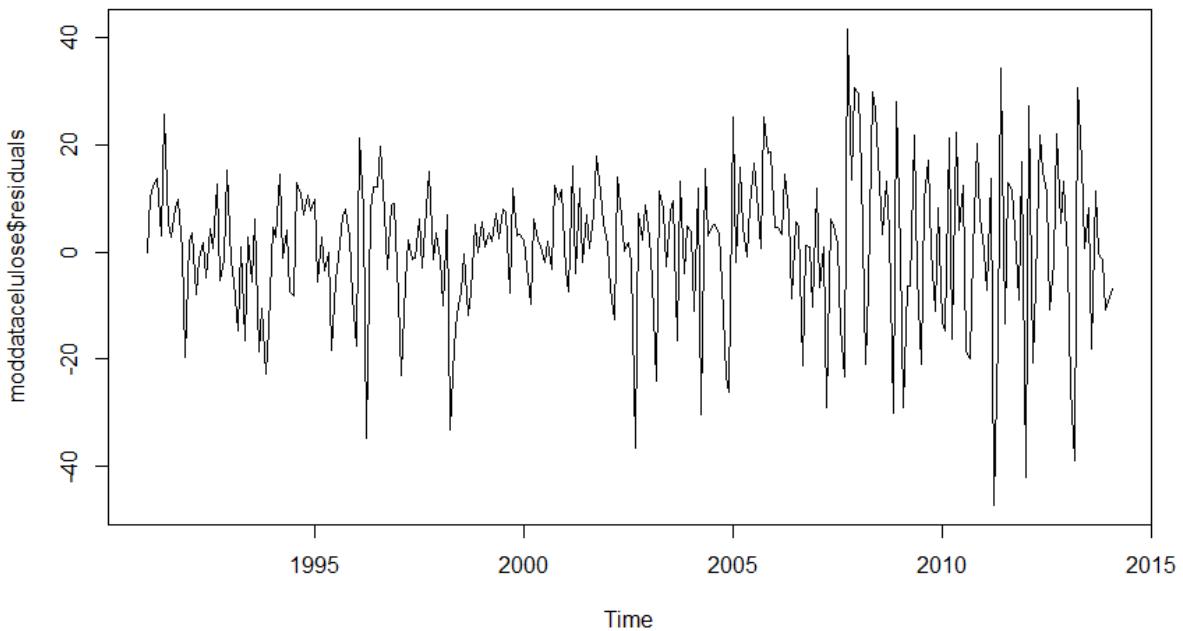


Valores dentro do intervalo de confiança.

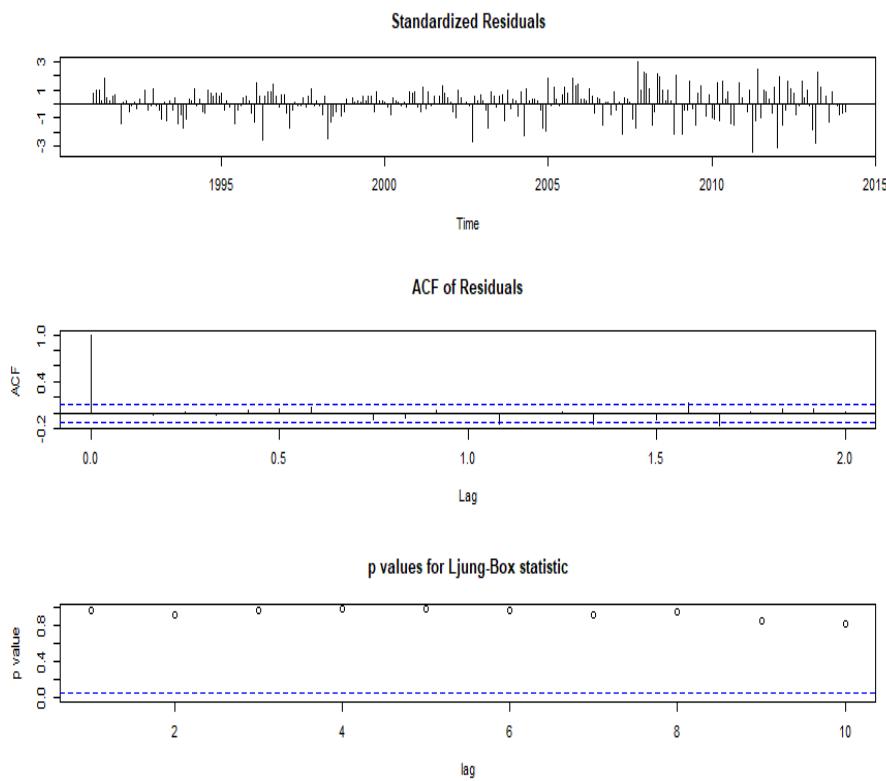
```
> tsdiag(moddataminerais)
```



```
> plot(moddatacelulose$residuals)
> acf(moddatacelulose$residuals, lag.max=(length(datacelulose)/2))
```



```
> tsdiag(moddatacelulose)
```



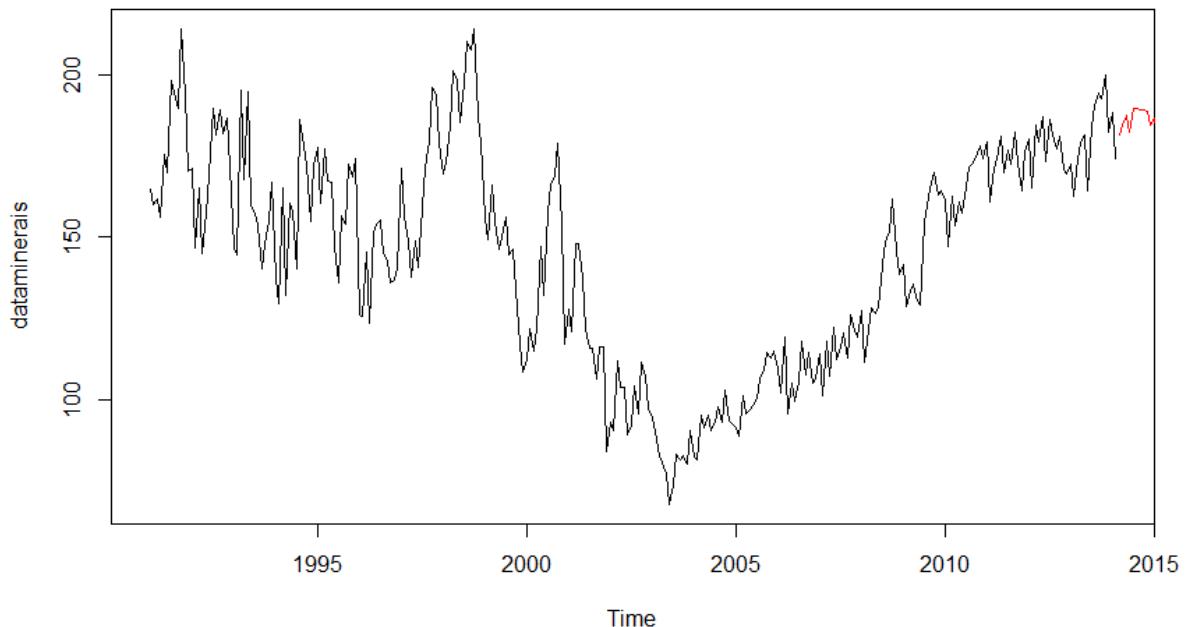
PREVER MODELOS A FRENTE

```
projecaodataminerais<-predict(moddataminerais, n.ahead=36)
> projecaodataminerais
```

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
2014		181.6118	184.5814	187.6025	182.2432	189.7182	189.7626	189.2674	189.2701	188.8424	184.5509	
2015	186.6957	181.5219	185.6498	187.2114	188.4076	183.6558	189.3573	190.7241	191.3405	190.9976	192.3692	187.8488
2016	189.5553	185.4920	187.9539	188.9129	189.7976	187.6157	190.4444	190.7728	190.8177	190.7378	190.9726	189.0574
2017	189.8854	187.9007										

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
2014		11.70513	14.49268	16.75629	18.48311	19.33254	20.09936	20.90431	21.74621	22.68588	23.63439	
2015	24.53420	25.37915	26.96531	28.21689	29.40664	30.51882	31.47469	32.39787	33.30949	34.20838	35.10668	35.98971
2016	36.84897	37.68460	39.21491	40.47745	41.69374	42.84094	43.84117	44.81104	45.77319	46.72784	47.68669	48.63471
2017	49.56234	50.46869										

```
> plot(dataminerais)
> lines(projecaodataminerais$pred, col="red")
```



```

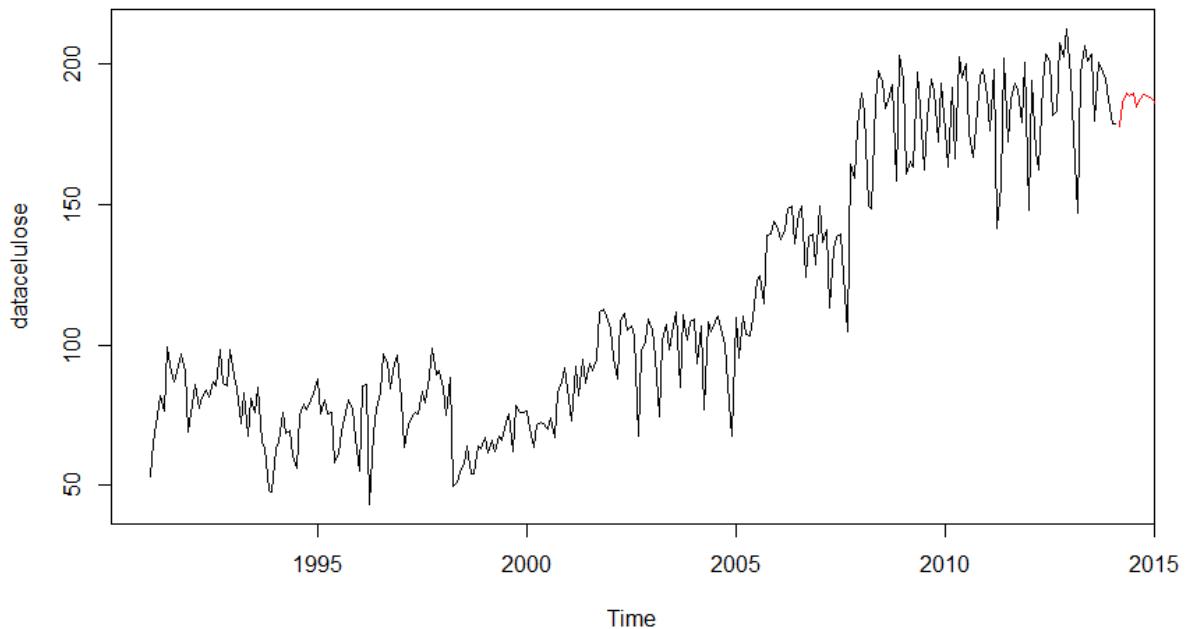
> projecaodatacelulose<-predict(moddatacelulose, n.ahead=36)
> projecaodatacelulose

$pred
      Jan      Feb      Mar      Apr      May      Jun      Jul      Aug      Sep      Oct      Nov      Dec
?2014 177.4993 186.0520 189.7944 188.9010 189.6623 184.7360 187.5133 189.3475 188.4785 187.9286
?2015 186.6380 183.8093 181.8820 187.5715 188.1822 187.0874 187.6184 185.2803 187.3029 186.8424 186.6308 185.5866
?2016 185.3469 185.2354 185.7479 185.8655 185.8925 185.8987 185.9001 185.9005 185.9006 185.9006 185.9006 185.9006
?2017 185.9006 185.9006

$se
      Jan      Feb      Mar      Apr      May      Jun      Jul      Aug      Sep      Oct      Nov      Dec
?2014 13.75471 15.15422 15.85570 16.42182 16.94678 17.45105 17.94007 18.41586 18.87960 19.33222
?2015 19.77447 20.20705 21.06187 21.64313 22.16093 22.65637 23.13886 23.61096 24.07369 24.52766 24.97337 25.41126
?2016 25.84174 26.26516 26.96179 27.48570 27.96756 28.43412 28.89155 29.34148 29.78453 30.22107 30.65139 31.07575
?2017 31.49440 31.90755

> plot(datacelulose)
> lines(projecaodatacelulose$pred, col="red")

```



RESIDUOS

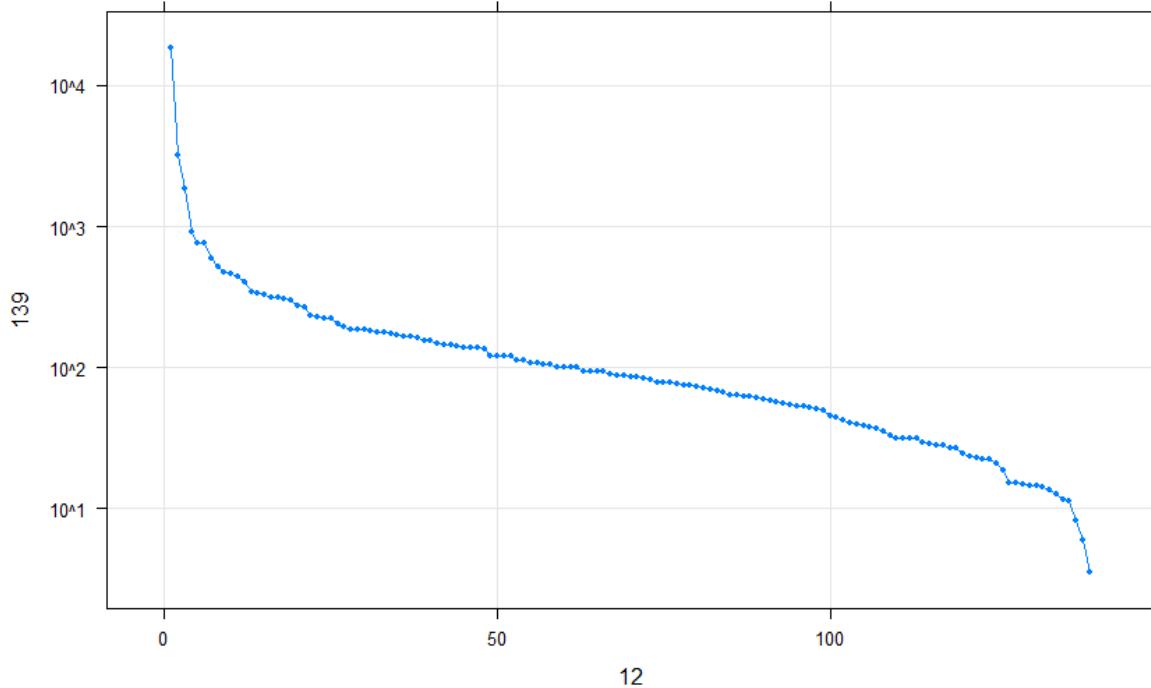
```
> a1 <- moddataminerais$residuals
> durbinWatsonTest(as.numeric(a1), alternative="two.sided")
[1] 2.001007
> a1 <- moddatacelulose$residuals
> durbinWatsonTest(as.numeric(a1), alternative="two.sided")
[1] 1.985762
```

RSSA

```
> library(Rssa)

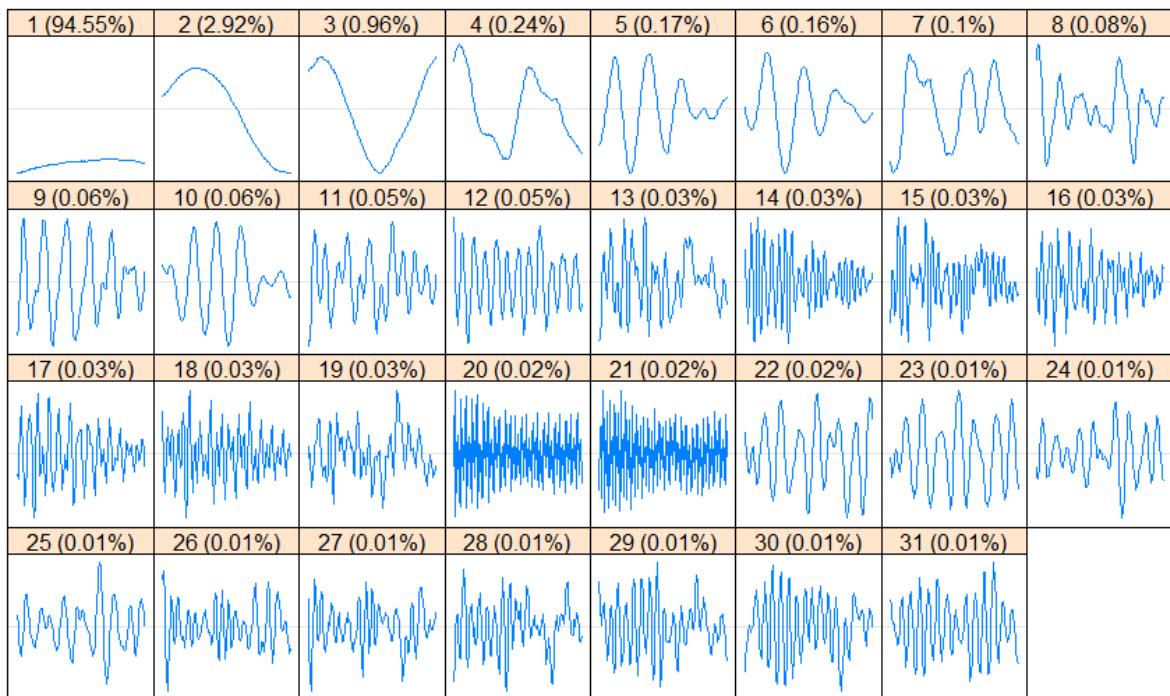
> s <- ssa(dataminerais, L=139, svd.method="svd")
> plot(s, type="values", ylab="139", xlab="12", main="")
```

Podemos ver os componentes da série por ordem de importância.



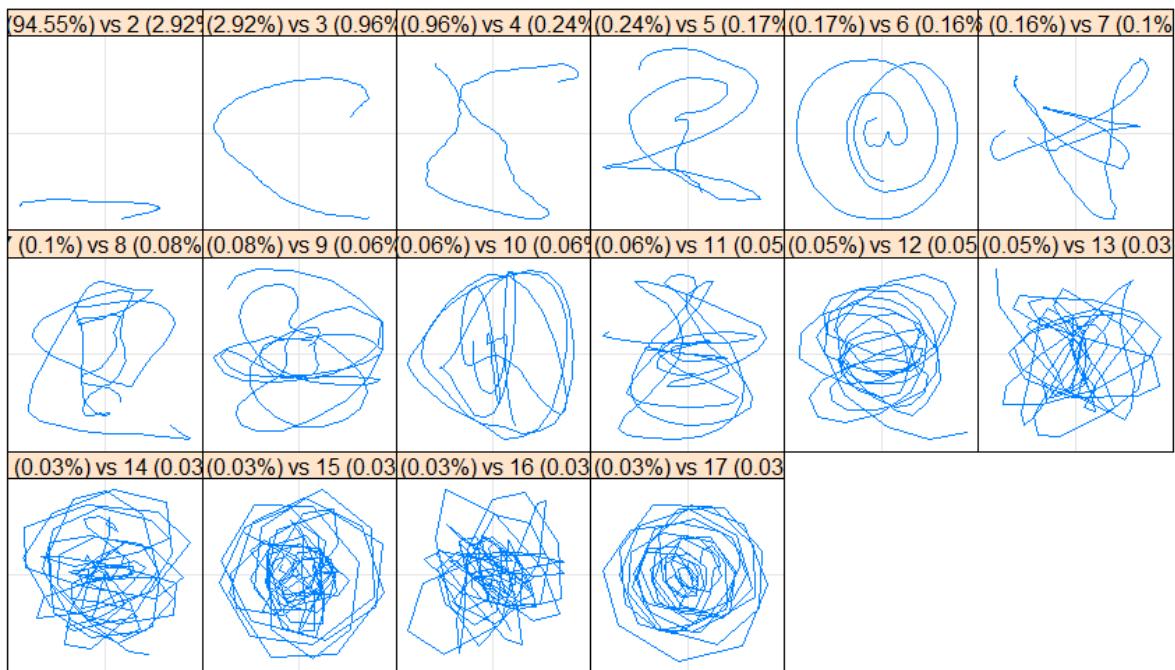
```
> plot(s, type="vectors", idx=1:31, main="")
```

Exibe o spectro dos componentes.



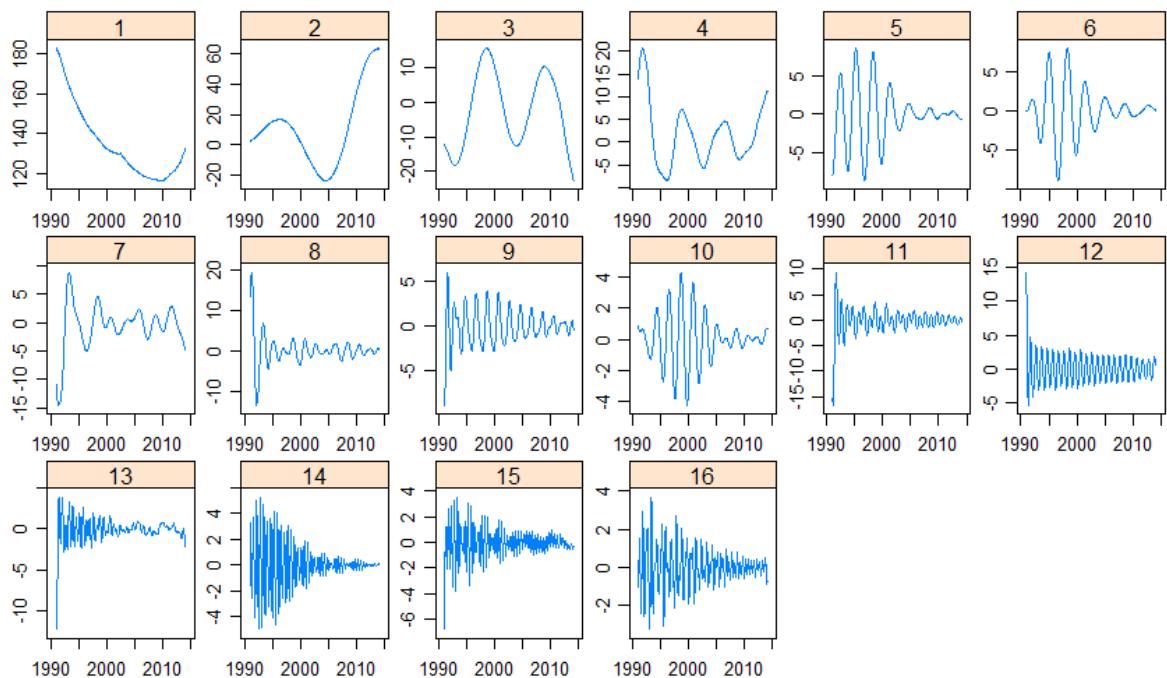
```
> plot(s, type="paired", idx=1:16)
```

Pairs of eigenvectors

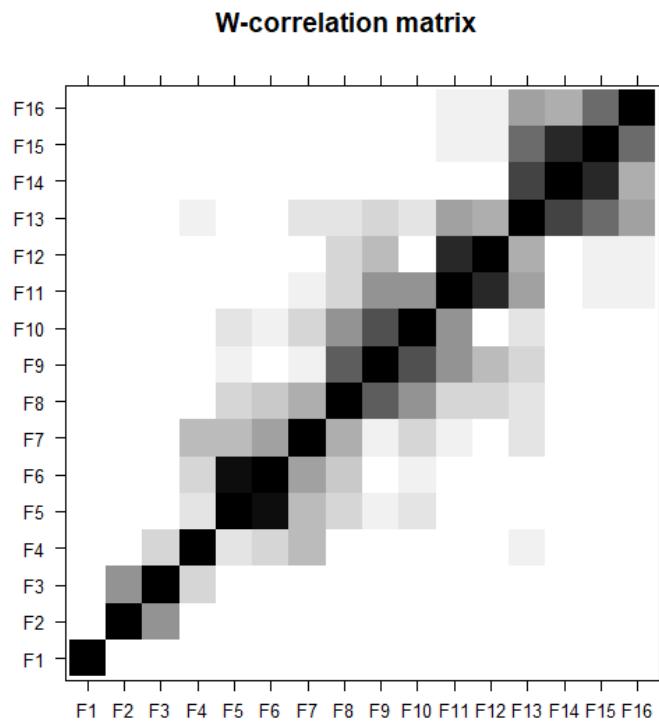


```
> plot(s, type = "series", groups = as.list(1:16))
```

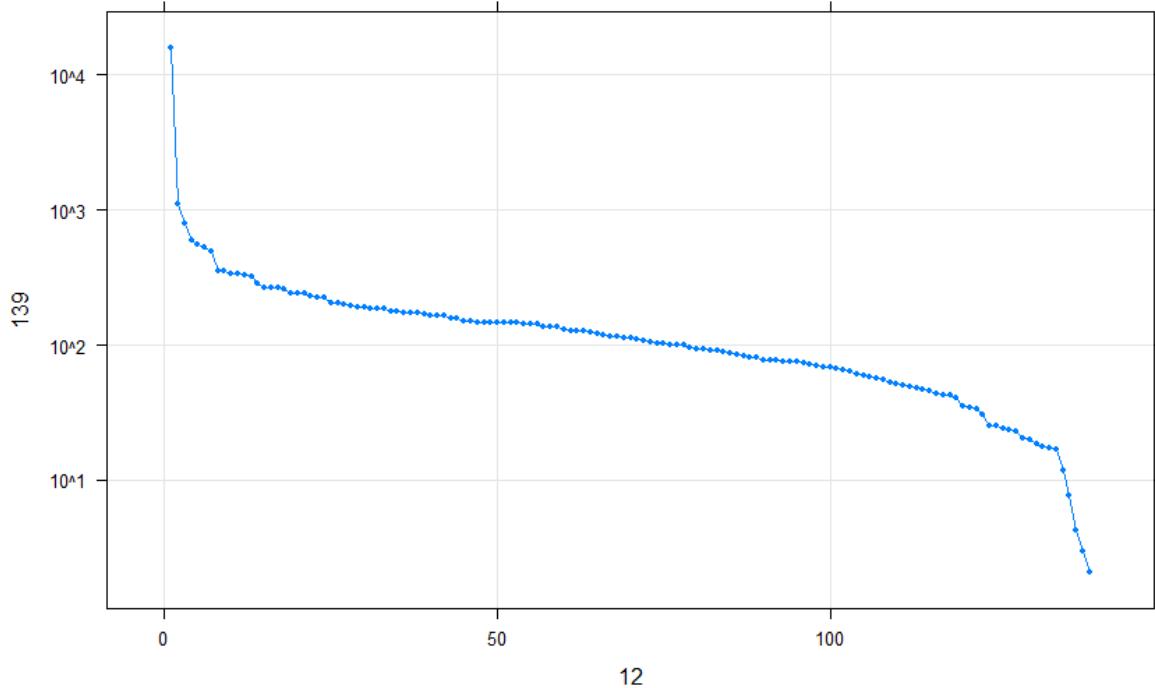
Reconstructed series



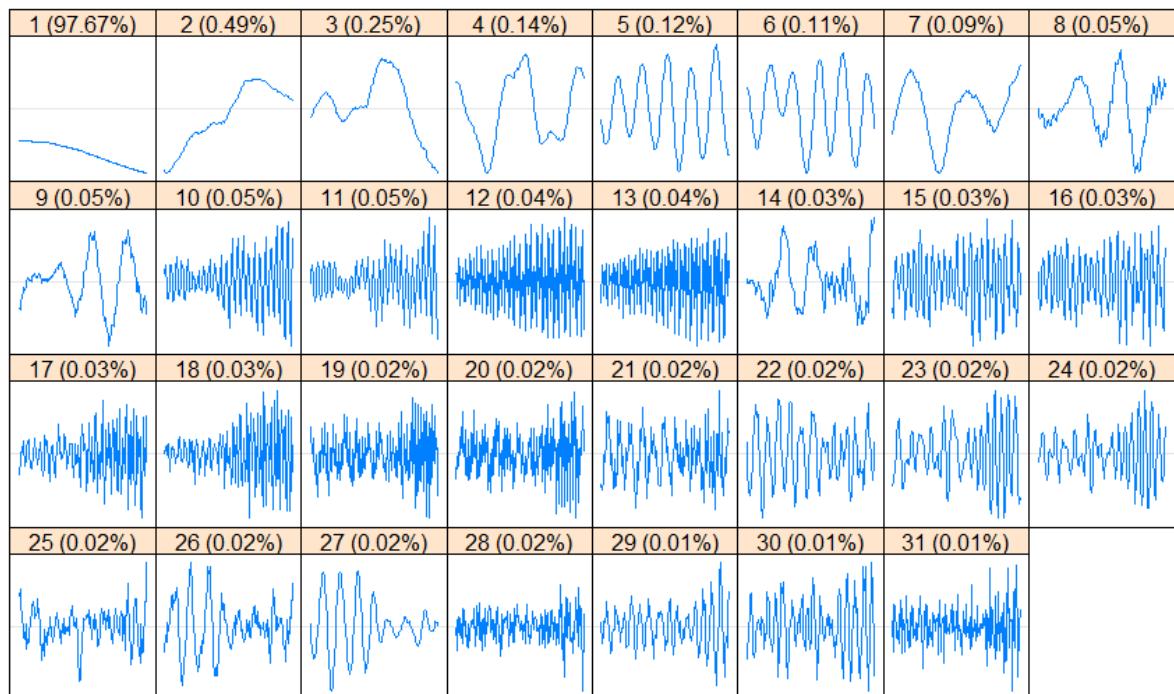
```
> plot(wcor(s, groups = 1:16))
```



```
> s <- ssa(datacelulose, L=139, svd.method="svd")
> plot(s, type="values", ylab="139", xlab="12", main="")
```

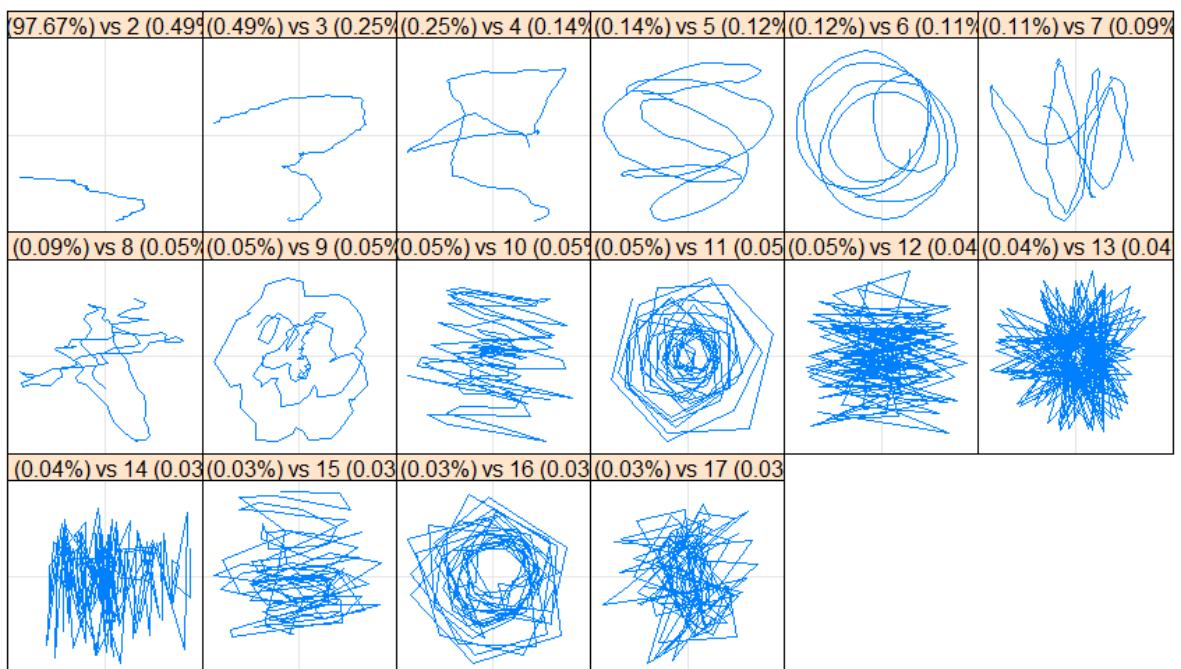


```
> plot(s, type="vectors", idx=1:31, main="")
```



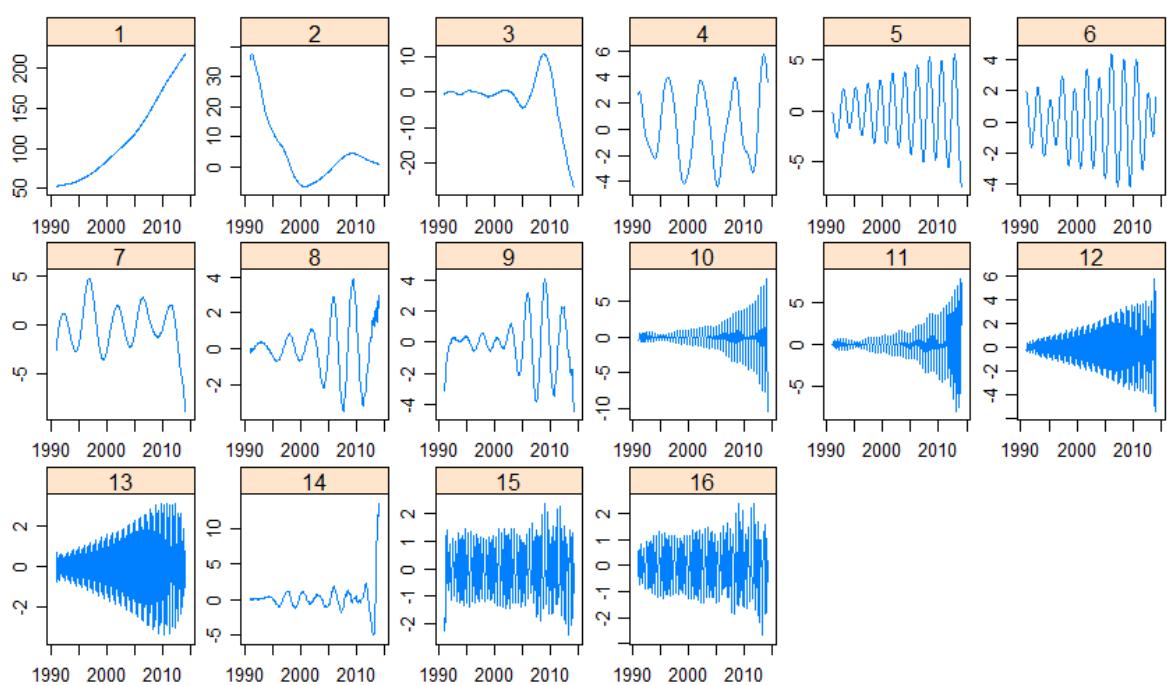
```
> plot(s, type="paired", idx=1:16)
```

Pairs of eigenvectors



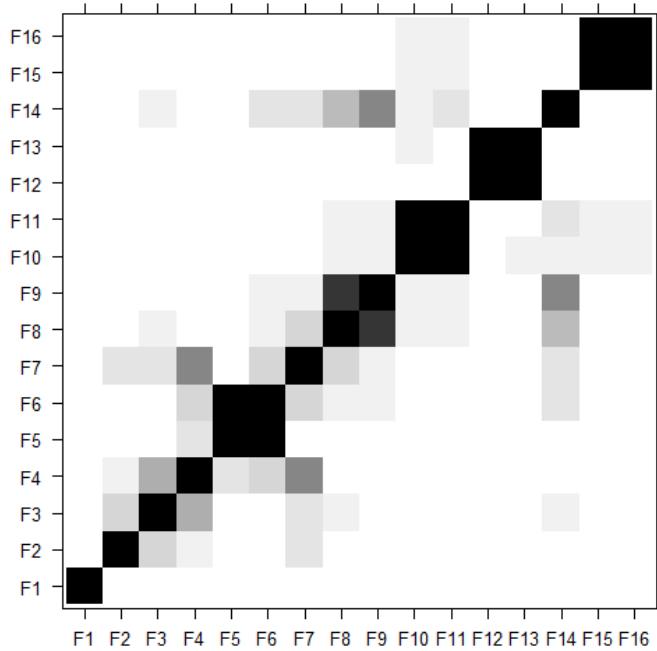
```
> plot(s, type = "series", groups = as.list(1:16))
```

Reconstructed series

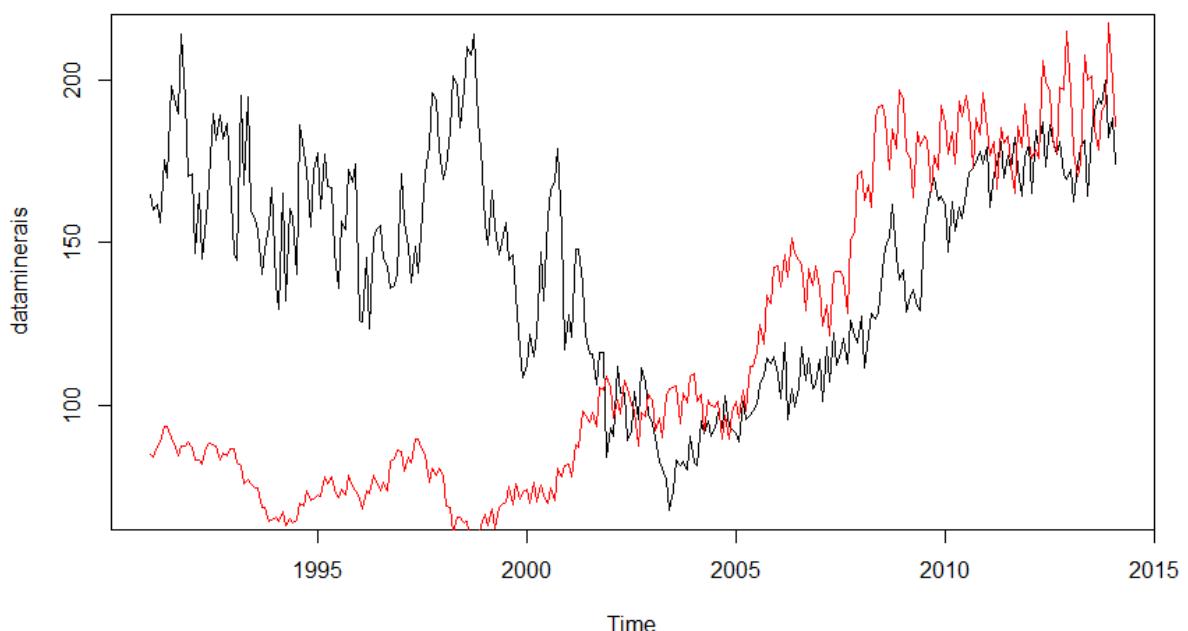


```
> plot(wcor(s, groups = 1:16))
```

W-correlation matrix



```
> res1 <- reconstruct(s, groups = list(1:16))
> plot(dataminerais)
> lines(res1$F1, col="red")
> p <- predict(s, groups = list(1:16), base = "original", method =
  "recurrent",
+               len = 5, R=100)
> lines(p, col="red")
> points(p, col="red")
```



```
> res1 <- reconstruct(s, groups = list(1:16))
> plot(datacelulose)
> lines(res1$F1, col="red")
> p <- predict(s, groups = list(1:16), base = "original", method =
"recurrent",
+               len = 5, R=100)
> lines(p, col="red")
> points(p, col="red")
```

