

Prova 2 - Estatística Econômica Aplicada

Bernardo Paulsen
Matheus Bragagnolo

Conteúdo

1	Introdução	2
2	Preparatórios	2
3	Dados	4
3.1	Importação dos Dados	4
3.2	Tramamento dos Dados	4
3.3	Descrição dos Dados	5
4	Outliers	6
5	Quebras Estruturais	8
5.1	Processo de Flutuação Empírica	8
5.2	Teste de Existência de Quebra Estrutural	8
5.3	Estimação da Data da Quebra Estrutural	9
5.4	Divisão da Série Temporal	9
6	Primeira Série Temporal	10
6.1	Definição da Ordem de Integração	10
6.1.1	Análise da Série Original	10
6.1.2	Diferenciação	19
6.1.3	Análise da Primeira Diferença	19
6.2	Identificação das Possíveis Formas Funcionais	29
6.3	Estimação	30
6.4	Diagnóstico dos Resíduos	33
6.4.1	Independência	33
6.4.2	Homoscedasticidade	36
6.4.3	Normalidade	39
6.5	Previsão e Acurácia	39
6.5.1	Previsão	40
6.5.2	Acurácia	41

7	Segunda Série Temporal	42
7.1	Definição da Ordem de Integração	42
7.1.1	Análise da Série Original	42
7.1.2	Diferenciação	52
7.1.3	Análise da Primeira Diferença	52
7.2	Identificação das Possíveis Formas Funcionais	62
7.3	Estimação	63
7.4	Diagnóstico dos Resíduos	67
7.4.1	Independência	68
7.4.2	Homoscedasticidade	69
7.4.3	Normalidade	72
7.5	Previsão e Acurácia	72
7.5.1	Previsão	73
7.5.2	Acurácia	74
8	Conclusão	75
	Referências	75

1 Introdução

O objetivo do presente trabalho é analisar uma série temporal univariada utilizando os métodos apresentados na cadeira ECOP124 (Estatística Econômica Aplicada) ministrada pelos professores Carlos Schonewald e Fernando Sabino. O trabalho será baseado nos tópicos:

- Pontos de Mudança e Quebras Estruturais;
- Modelos ARMA Univariados;
- Mais Testes e Previsões;
- Não Estacionariedade, Testes de Raiz Unitária e Modelos ARIMA(p,d,q);
- Modelos de Volatilidade Univariada.

Como material de apoio para o desenvolvimento do trabalho encontram-se disponíveis notas de aula e vídeo-aulas. Este trabalho é referente à dupla 1, à qual coube a subsérie temporal 4 (observações 5775 à 6564). O código do trabalho pode ser encontrado em repositório do github no link: <https://github.com/bernardopaulsen/ecop124>.

2 Preparatórios

Primeriamente, precisamos importar as bibliotecas que serão necessárias para executar os códigos das próximas seções. Utilizamos as seguintes bibliotecas:

- *astsa* (Stoffer 2020): função *sarima*.

- (DescTools) (Andri et mult. al. 2020): função *TheilU*;
- *forecast* (Hyndman e Khandakar 2008): funções *auto.arima* e *forecast*;
- *lubridate* (Grolemund e Wickham 2011): função *parse_date_time*;
- *nortsTest* (Alonzo Matamoros e Nieto-Reyes 2020): função *Lm.test*;
- *quantmod* (Ryan e Ulrich 2020): função *xts*;
- *rugarch* (Ghalanos 2020): funções *ugarchfit*, *ugarchforecast* e *ugarchspec*;
- *stats* (R Core Team 2020): funções *acf* e *Box.test*;
- *strucchange* (Zeileis 2006): funções *efp* e *sctest*;
- *tseries* (Trapletti e Hornik 2020): função *pacf*;
- *tsoutliers* (Lacalle 2019): funções *tso* e *tsclean*;
- *urca* (Pfaff 2008): função *ur.df*;
- *uroot* (Lacalle 2020): função *ch.test*.

```
library('astsa')
library('DescTools')
library('forecast')
library('lubridate')
library('nortsTest')
library('quantmod')
library('rugarch')
library('stats')
library('strucchange')
library('tseries')
library('tsoutliers')
library('urca')
library('uroot')
```

Antes de importar os dados é necessário selecionar como diretório de trabalho a pasta que contém o arquivo com os dados.

```
setwd("~/Google Drive/Mestrado/Estat/Prova2/3")
```

3 Dados

3.1 Importação dos Dados

No *chunk* a seguir importamos os dados e selecionamos a amostra correspondente ao nosso grupo.

```
file_name    <- 'dataset.Rds'
sample_begin <- 5775
sample_end   <- 6564
dataset      <- readRDS(file_name)[sample_begin:sample_end,]
```

Agora, podemos analisar brevementes dados importados (sumário e primeiros valores).

```
summary(dataset)

##      TIME      Value
## Length:790    Min.   :1.000
## Class :character 1st Qu.:1.900
## Mode  :character Median :2.450
##                      Mean  :2.741
##                      3rd Qu.:3.600
##                      Max.   :5.500

head(dataset)

## # A tibble: 6 x 2
##   TIME      Value
##   <chr>    <dbl>
## 1 1955-01     2.6
## 2 1955-02     2.5
## 3 1955-03     2.3
## 4 1955-04     2.5
## 5 1955-05     2.4
## 6 1955-06     2.6
```

Como podemos verificar acima, a função *summary* nos mostra que os elementos da coluna *TIME* são do tipo *character*, e a função *head* que o formato das datas é “YYYY-MM”. Essas informações serão úteis na próxima subseção, quando formos tratar os dados.

3.2 Tramamento dos Dados

Para o uso dos dados nas próximas seções é necessário transformar os dados da coluna *TIME* do formato *character* para o formato *datetime* (para isso usamos

as informações coletadas na subseção anterior). Além disso, é necessário transformar a estrutura de dados de ‘tabela’ para ‘série temporal’. Isso é feito no *chunk* a seguir.

```
dataset$TIME <- parse_date_time(dataset$TIME,
                                orders = 'ym')
dataset      <- xts(dataset$Value,
                    order.by = dataset$TIME)
colnames(dataset) <- 'Value'
```

Mais uma vez, analisamos os dados.

```
summary(dataset)

##          Index                      Value
## Min.      :1955-01-01 00:00:00   Min.      :1.000
## 1st Qu.:1971-06-08 12:00:00   1st Qu.:1.900
## Median :1987-11-16 00:00:00   Median :2.450
## Mean    :1987-11-16 00:25:31   Mean     :2.741
## 3rd Qu.:2004-04-23 12:00:00   3rd Qu.:3.600
## Max.     :2020-10-01 00:00:00   Max.      :5.500

head(dataset)

##          Value
## 1955-01-01    2.6
## 1955-02-01    2.5
## 1955-03-01    2.3
## 1955-04-01    2.5
## 1955-05-01    2.4
## 1955-06-01    2.6
```

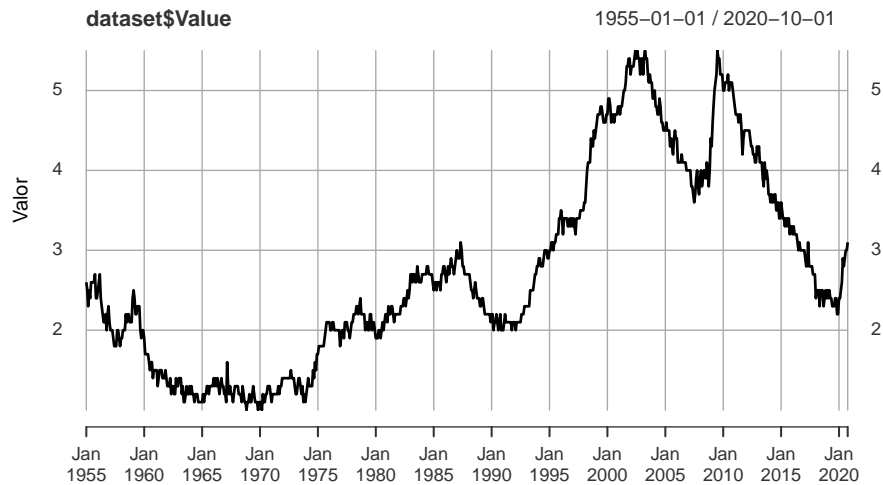
Agora, vimos que o formato das datas é *datetime*, e as datas passaram de uma coluna de dados para índice dos valores da série temporal.

3.3 Descrição dos Dados

Como último passo na importação dos dados, os descrevemos. A média e os valores mínimos e máximos estão na subseção acima, então agora mostramos apenas o gráfico da série temporal (Figura 1). Maiores descrições dos dados serão apresentadas no tempo devido.

Figura 1: Série Temporal

```
plot(dataset$Value,  
      type = 'l',  
      xlab = 'Tempo',  
      ylab = 'Valor')
```



4 Outliers

Antes de estimar os modelos, detectamos e removemos outliers. No teste abaixo, cinco tipos de outliers são considerados: outliers aditivos, mudanças de nível, mudanças temporárias, outliers inovadores e mudanças de nível sazonal. O teste segue a metodologia de Chen e Liu 1993.

```
ol <- tso(ts(dataset))  
ol  
  
## Series: ts(dataset)  
## Regression with ARIMA(0,1,1) errors  
##  
## Coefficients:  
##          ma1    A0147  
##       -0.1885  0.3986  
## s.e.    0.0365  0.0792  
##  
## sigma^2 estimated as 0.01059:  log likelihood=675.7  
## AIC=-1345.4   AICc=-1345.37   BIC=-1331.39
```

```
##
## Outliers:
##   type ind time coefhat tstat
## 1   A0 147  147  0.3986 5.032
```

Considerando o resultado do teste do *chunk* anterior, substituímos o valor outlier com a função *tsclean*.

```
dataset <- tsclean(dataset)
```

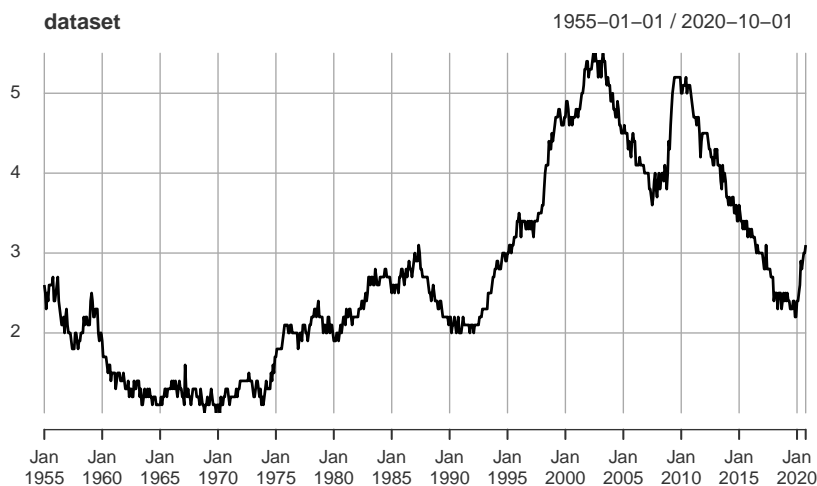
No próximo *chunk* apresentamos o sumário dos novos dados, junto ao gráfico dos novos dados.

```
summary(dataset)
```

```
##           Index                               Value
## Min.      :1955-01-01 00:00:00   Min.      :1.00
## 1st Qu.   :1971-06-08 12:00:00   1st Qu.   :1.90
## Median    :1987-11-16 00:00:00   Median    :2.45
## Mean      :1987-11-16 00:25:31   Mean      :2.74
## 3rd Qu.   :2004-04-23 12:00:00   3rd Qu.   :3.60
## Max.      :2020-10-01 00:00:00   Max.      :5.50
```

Figura 2: Série Temporal sem Outliers

```
plot(dataset)
```



5 Quebras Estruturais

Nesta seção testamos a existência de quebra estrutural na série. No caso de existência de quebra estrutural, estimamos a data de quebra.

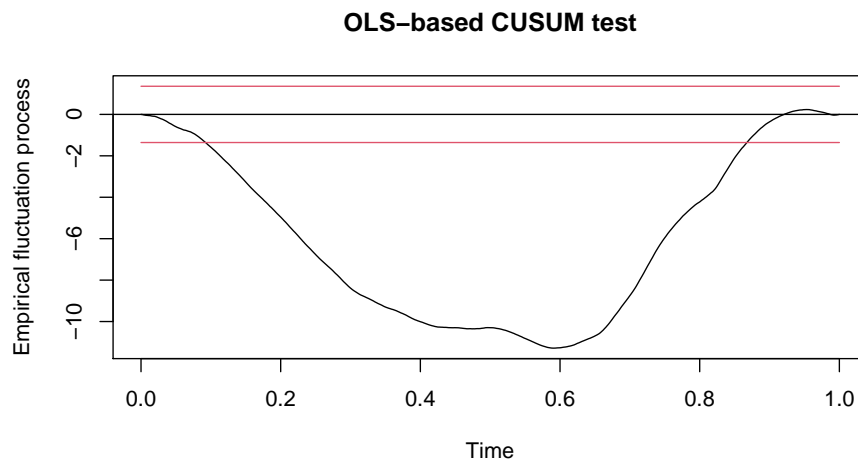
5.1 Processo de Flutuação Empírica

Priemeiramente, calculamos o processo de flutuação empírica e o apresentamos no gráfico abaixo.

```
efp1 <- efp(dataset~1, type="OLS-CUSUM")
```

Figura 3: Processo de Flutuação Empírica

```
plot(efp1)
```



O gráfico mostra que o processo ultrapassa os pontos críticos do intervalo de confiança, indicando que existe quebra estrutural na série temporal.

5.2 Teste de Existência de Quebra Estrutural

Na subseção anterior verificamos que o processo de flutuação empírica ultrapassa os limites do intervalo de confiança. Agora, testamos a hipótese de quebra estrutural.

```
sctest(dataset~1, type="OLS-CUSUM")
```



```
##
## OLS-based CUSUM test
##
## data: dataset ~ 1
## S0 = 11.284, p-value < 2.2e-16
```

O teste rejeita a hipótese nula de não existência de quebra estrutural, portanto podemos considerar que existe quebra estrutural na série temporal.

5.3 Estimação da Data da Quebra Estrutural

A estimativa de data mais provável de quebra estrutural é o ponto do processo de flutuação empírica que mais se distancia dos pontos máximos do intervalo de confiança. Então, verificamos qual é esse valor.

```
point <- which.min(efp1$process)
```

5.4 Divisão da Série Temporal

Na subseção anterior estimamos o ponto mais provável de quebra estrutural. Agora, separamos a série temporal original nesse ponto, como objetivo de obter uma série temporal para cada processo gerador.

```
data1 <- dataset[1:point-1]
data2 <- dataset[point:length(dataset)]
```

Abaixo apresentamos os sumários das duas novas séries temporais.

```
summary(data1)
```

##	Index	Value
##	Min. :1955-01-01 00:00:00	Min. :1.000
##	1st Qu.:1964-09-16 00:00:00	1st Qu.:1.300
##	Median :1974-06-01 00:00:00	Median :2.000
##	Mean :1974-06-01 08:47:16	Mean :1.905
##	3rd Qu.:1984-02-15 12:00:00	3rd Qu.:2.300
##	Max. :1993-11-01 00:00:00	Max. :3.100

```
summary(data2)
```

##	Index	Value
##	Min. :1993-12-01 00:00:00	Min. :2.200
##	1st Qu.:2000-08-16 12:00:00	1st Qu.:3.200
##	Median :2007-05-01 00:00:00	Median :4.000
##	Mean :2007-05-02 04:00:44	Mean :3.948
##	3rd Qu.:2014-01-16 12:00:00	3rd Qu.:4.700
##	Max. :2020-10-01 00:00:00	Max. :5.500

6 Primeira Série Temporal

6.1 Definição da Ordem de Integração

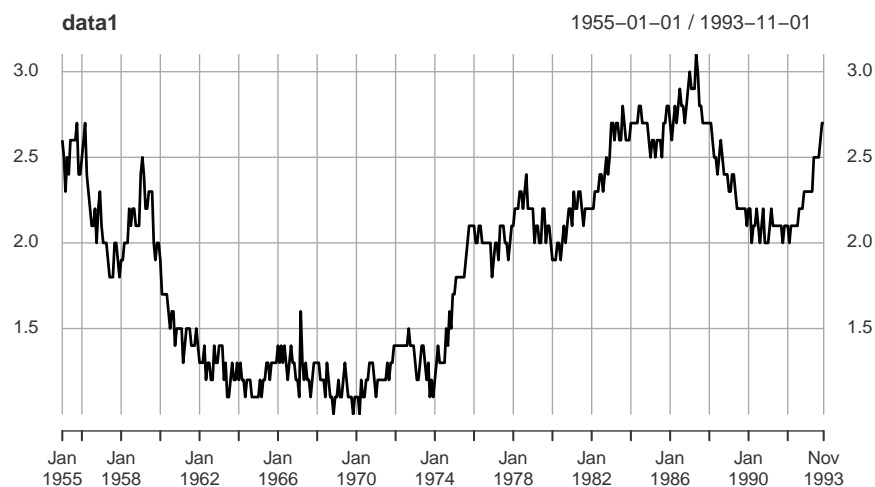
O primeiro passo na metodologia Box-Jenkins (Box et al. 2015) é a definição da ordem de integração da série temporal .

6.1.1 Análise da Série Original

Começamos o processo de definição da ordem de integração analisando o gráfico da série temporal.

Figura 4: Série Temporal

```
plot(data1)
```



A visualização da série temporal indica presença de raiz unitária. Para coletar mais indícios visuais analisamos abaixo a função de autocorrelação.

Figura 5: FAC da Série Temporal

```
acf(as.matrix(data1), lag.max=40)
```

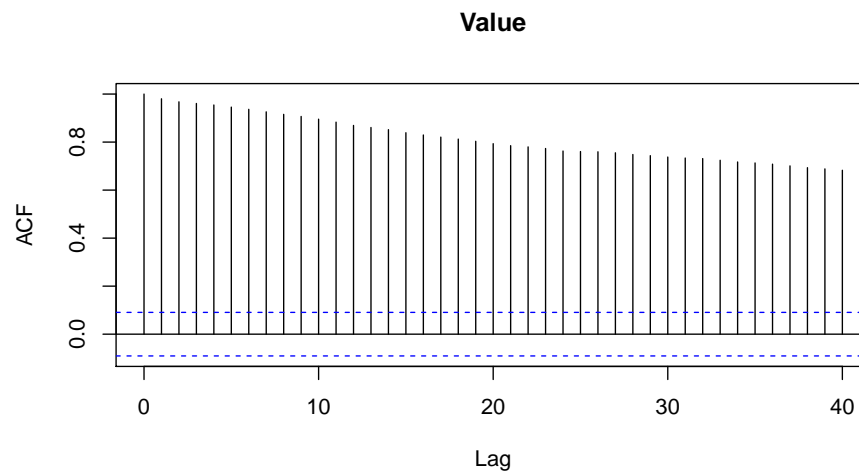
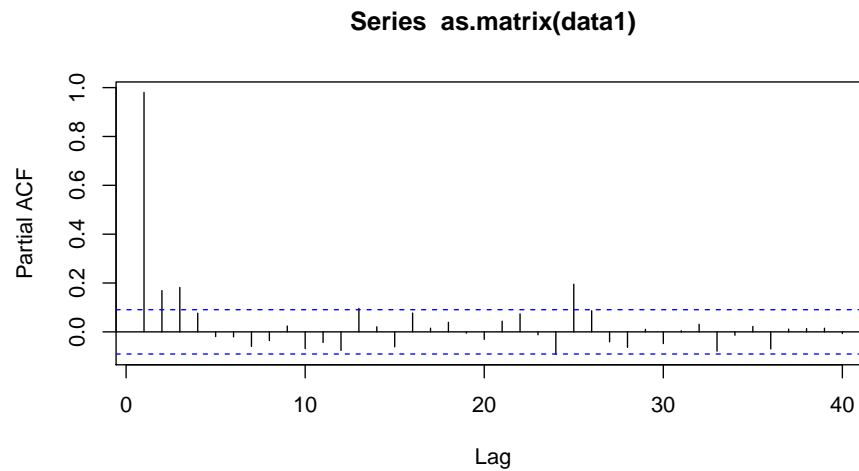


Figura 6: FACP da Série Temporal

```
pacf(as.matrix(data1), lag.max=40)
```



A função de autocorrelação aparentemente possui decaimento linear, indi-

cando possibilidade de raiz unitária. Para testar a hipótese de presença de raiz unitária com o teste Dickey-Fuller aumentado (Dickey e Fuller 1979). O teste será realizado com *drift*, pois a visualização do gráfico da série temporal indica a presença de tal. Para escolher o lag do teste, começaremos pelo lag 1 e, caso os resíduos do teste forem ruído branco, aceitamos o lag. Caso os resíduos não apresentarem comportamento de ruído branco, repetimos os passos com o lag imediatamente maior. No *chunk* abaixo realizamos testes ADF para 24 lags, e para cada teste testamos os resíduos com testes de Ljung-Box (Ljung e Box 1978) até 25 lags.

```
results <- matrix(,nrow=25,ncol=24)
for (i in 1:24){
  adf <- ur.df(data1$Value, type='drift', lags=i)
  for (e in 1:25){
    box <- Box.test(adf$res, lag=e)
    results[e,i] <- box$p.value
  }
}
```

Nas figuras abaixo encontram-se os resultados dos testes de Ljung-Box para os resíduos de cada um dos testes de Dickey-Fuller aumentados.

Figura 7: Resultados dos Testes de Ljung-Box para os Lags 1 a 6

```
par(mfrow = c(3,2))
for (i in 1:6){
  plot(results[,i], main=i)
}
```

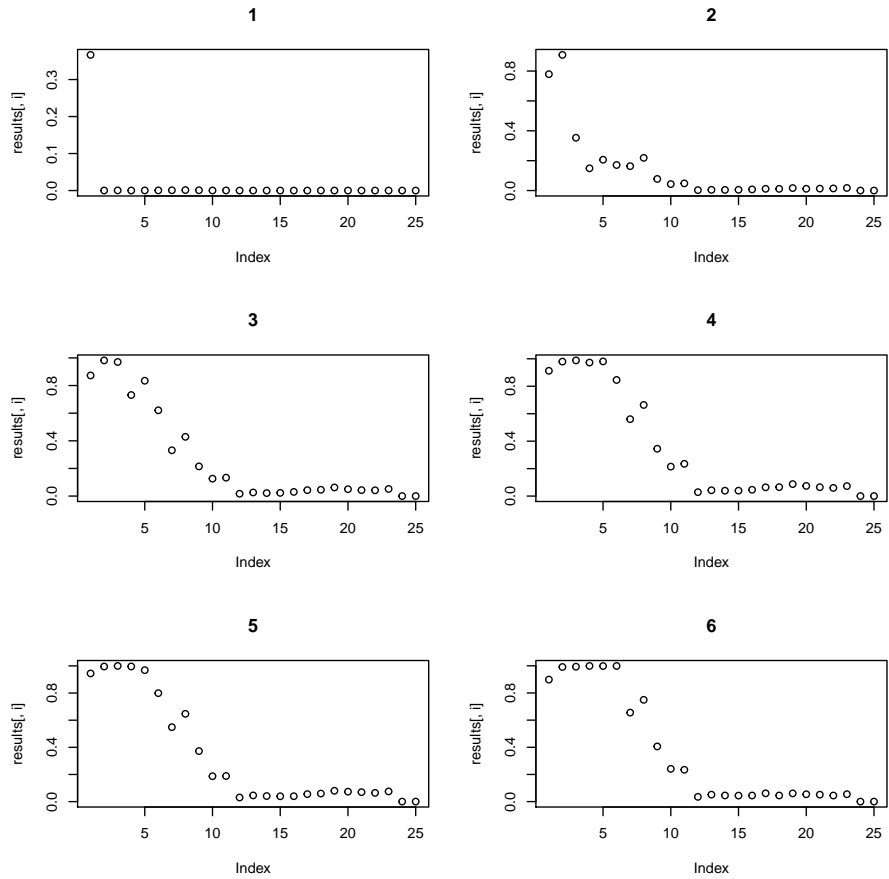


Figura 8: Resultados dos Testes de Ljung-Box para os Lags 7 a 12

```
par(mfrow = c(3,2))
for (i in 7:12){
  plot(results[,i], main=i)
}
```

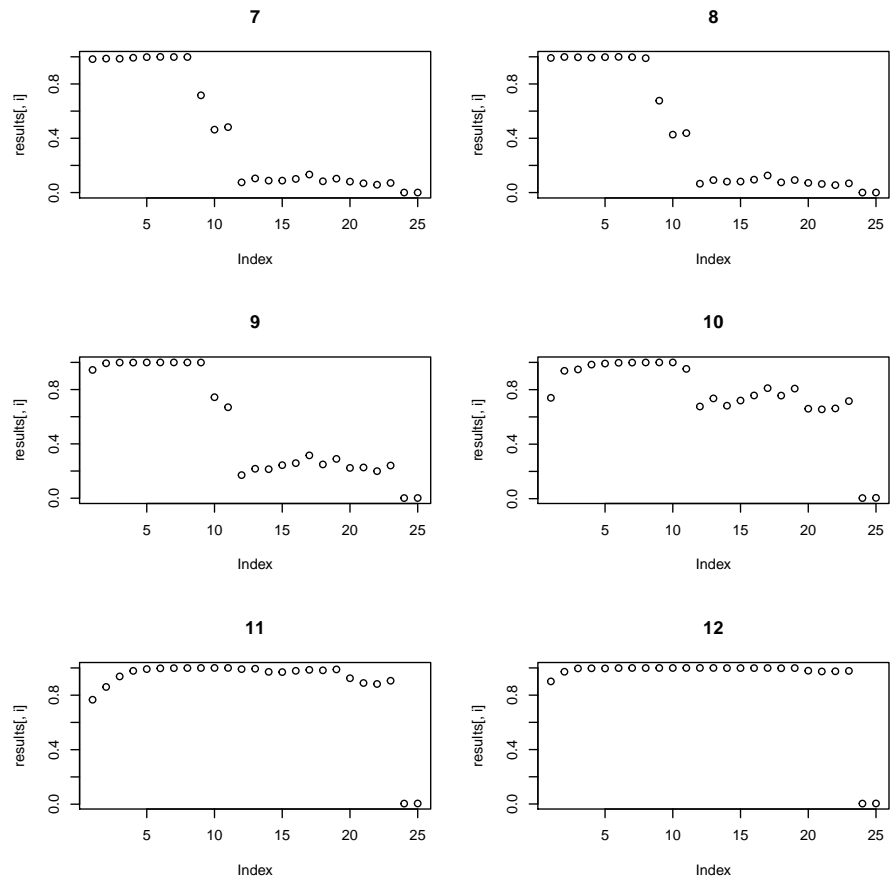


Figura 9: Resultados dos Testes de Ljung-Box para os Lags 13 a 18

```
par(mfrow = c(3,2))
for (i in 13:18){
  plot(results[,i], main=i)
}
```

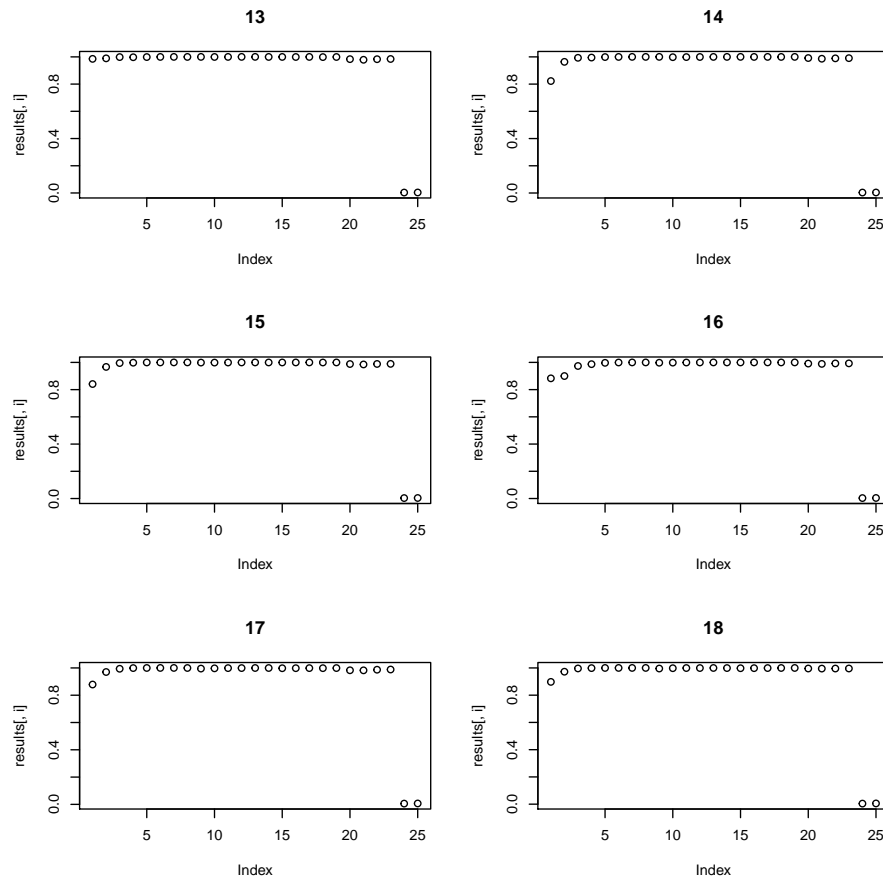
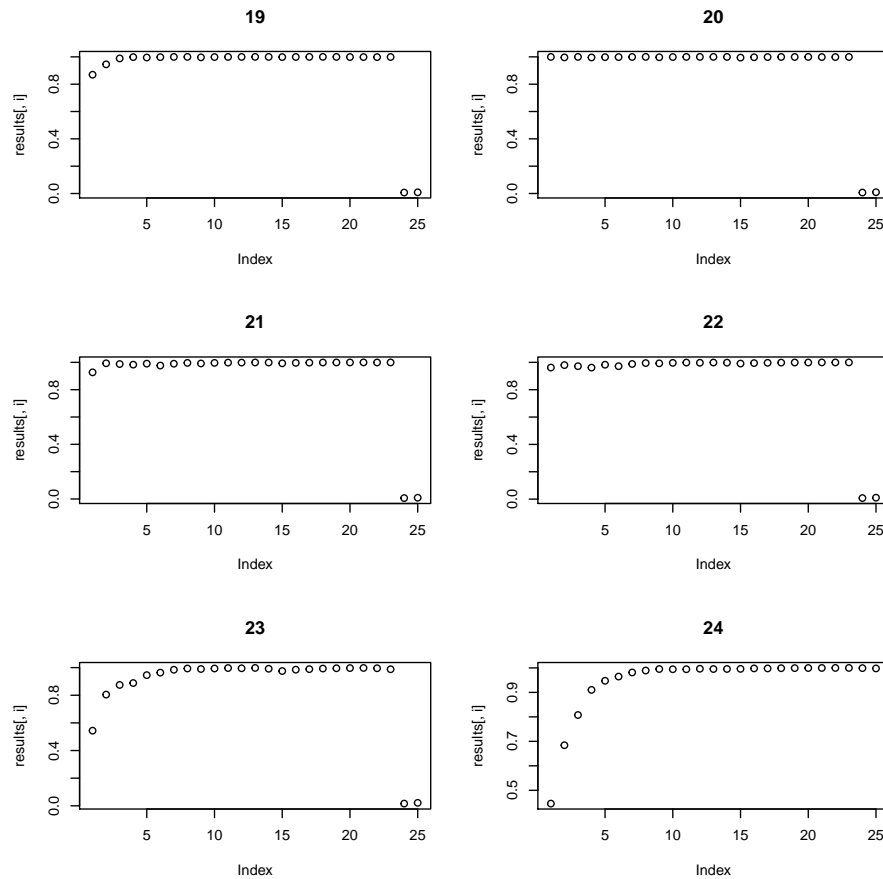


Figura 10: Resultados dos Testes de Ljung-Box para os Lags 19 a 24

```
par(mfrow = c(3,2))
for (i in 19:24){
  plot(results[,i], main=i)
}
```

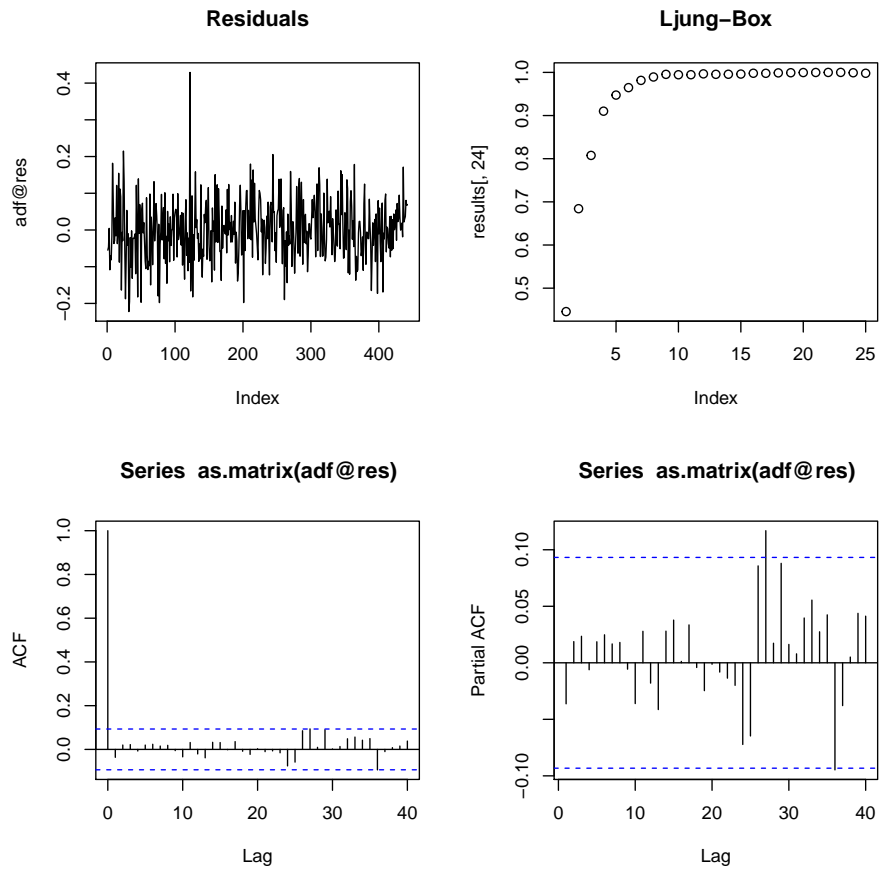


Nos gráficos acima podemos ver que a única ordem de lags que gera resíduos que são ruído branco é a ordem 24, então realizamos o teste com essa ordem.

```
adf <- ur.df(data1$Value, type='drift', lags=24)
```


Figura 11: Resíduos

```
par(mfrow = c(2,2))
plot(adf@res, type='l', main='Residuals')
plot(results[,24], main='Ljung-Box')
acf(as.matrix(adf@res), lag.max=40)
pacf(as.matrix(adf@res), lag.max=40)
```



Tanto o gráfico dos resíduos quanto os gráficos das funções de autocorrelação e autocorrelação parcial indicam que os resíduos do teste se comportam como ruído branco. Os p-valores do teste de Ljung-Box não rejeitam a hipótese de independência dos resíduos.

Agora, visualizamos o sumário do teste.

```
summary(adf)

##
## #####
## # 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
## -0.22223 -0.05501 -0.00398  0.05128  0.42905
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.012568   0.015044   0.835   0.4039
## z.lag.1       -0.005905   0.007713  -0.766   0.4444
## z.diff.lag1   -0.248088   0.047395  -5.234 2.63e-07 ***
## z.diff.lag2   -0.199103   0.048971  -4.066 5.73e-05 ***
## z.diff.lag3   -0.101610   0.049779  -2.041  0.0419 *
## z.diff.lag4    0.058018   0.049808   1.165  0.2448
## z.diff.lag5    0.051130   0.049664   1.030  0.3038
## z.diff.lag6    0.108875   0.049703   2.190  0.0290 *
## z.diff.lag7    0.113831   0.049872   2.282  0.0230 *
## z.diff.lag8    0.013664   0.049902   0.274  0.7844
## z.diff.lag9    0.072790   0.049379   1.474  0.1412
## z.diff.lag10   0.140793   0.048782   2.886  0.0041 **
## z.diff.lag11   0.089136   0.049219   1.811  0.0709 .
## z.diff.lag12  -0.119215   0.049379  -2.414  0.0162 *
## z.diff.lag13  -0.010514   0.049107  -0.214  0.8306
## z.diff.lag14   0.033982   0.048935   0.694  0.4878
## z.diff.lag15  -0.043762   0.048442  -0.903  0.3668
## z.diff.lag16   0.022091   0.048367   0.457  0.6481
## z.diff.lag17  -0.004231   0.048328  -0.088  0.9303
## z.diff.lag18   0.071577   0.048143   1.487  0.1378
## z.diff.lag19   0.033270   0.047968   0.694  0.4883
## z.diff.lag20  -0.077872   0.047713  -1.632  0.1034
## z.diff.lag21  -0.070659   0.047849  -1.477  0.1405
## z.diff.lag22  -0.053579   0.047661  -1.124  0.2616
## z.diff.lag23   0.016124   0.046540   0.346  0.7292
## z.diff.lag24  -0.253202   0.044893  -5.640 3.14e-08 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08449 on 416 degrees of freedom
## Multiple R-squared:  0.2345, Adjusted R-squared:  0.1885
## F-statistic: 5.098 on 25 and 416 DF,  p-value: 2.328e-13
##
##
## Value of test-statistic is: -0.7656 0.3598
##
## Critical values for test statistics:
##      1pct  5pct 10pct
## tau2 -3.44 -2.87 -2.57
## phi1  6.47  4.61  3.79
```

O valor da estatística do teste é de -0,7656. Os valores críticos para o teste são de -3,44 (1%), -2,87 (5%) e -2,5 (10%). Sendo assim, o valor do teste não ultrapassou os valores críticos para nenhum grau de significância. O teste então aceita a hipótese nula de presença de raiz unitária.

6.1.2 Diferenciação

Como tentativa para estacionarizar a série, aplicamos a primeira diferença.

```
data1$Diff <- diff(data1)
```

Abaixo, o sumário dos novos dados.

```
summary(data1)
```

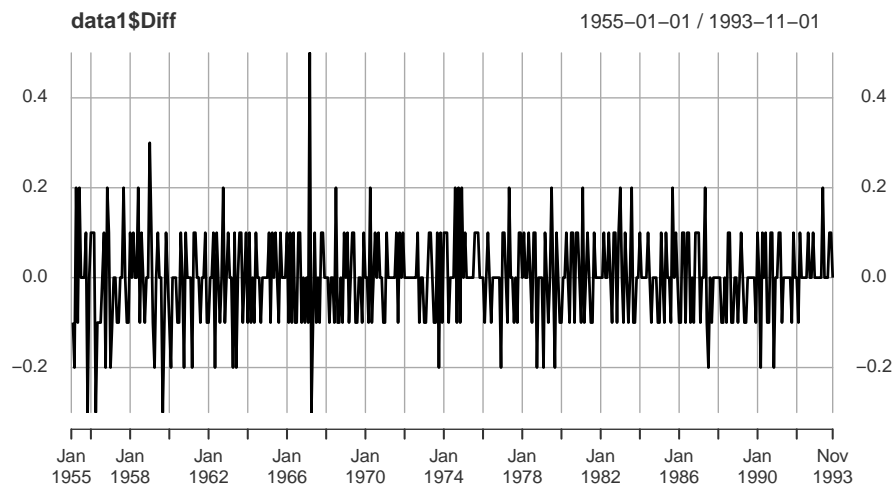
	Index	Value	Diff
##	Min. :1955-01-01 00:00:00	Min. :1.000	Min. :-0.3000000
##	1st Qu.:1964-09-16 00:00:00	1st Qu.:1.300	1st Qu.: -0.1000000
##	Median :1974-06-01 00:00:00	Median :2.000	Median : 0.0000000
##	Mean :1974-06-01 08:47:16	Mean :1.905	Mean : 0.0002146
##	3rd Qu.:1984-02-15 12:00:00	3rd Qu.:2.300	3rd Qu.: 0.1000000
##	Max. :1993-11-01 00:00:00	Max. :3.100	Max. : 0.5000000
##			NA's :1

6.1.3 Análise da Primeira Diferença

Começamos a nova análise analisando o gráfico da primeira diferença da série temporal.

Figura 12: Primeira Diferença

```
plot(data1$Diff)
```



A visualização da série temporal não indica presença de raiz unitária. Para coletar mais indícios visuais analisamos abaixo a função de autocorrelação.

Figura 13: FAC da Primeira Diferença

```
acf(as.matrix(na.omit(data1$Diff)), lag.max=40)
```

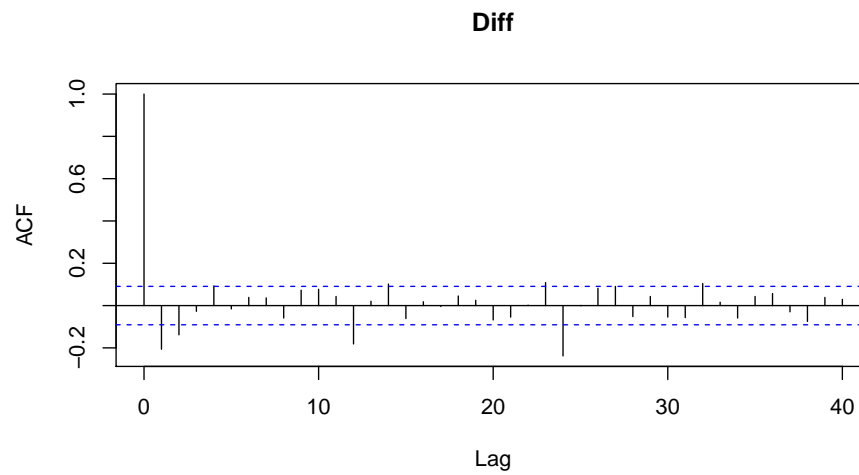
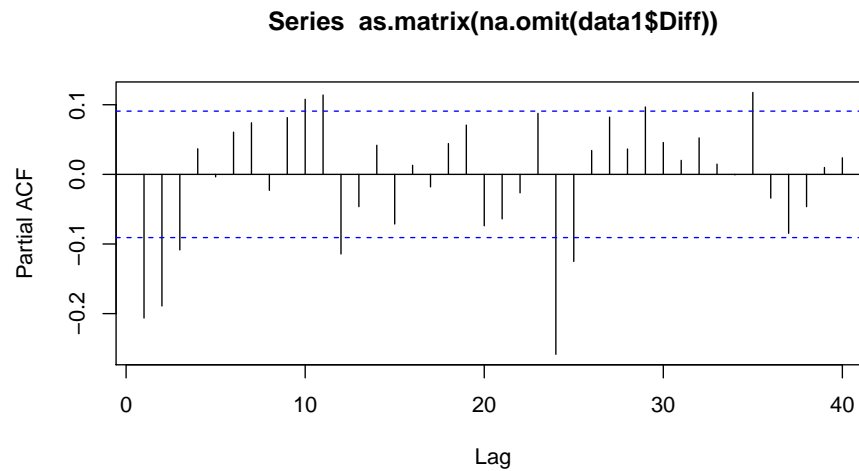


Figura 14: FACP da Primeira Diferença

```
pacf(as.matrix(na.omit(data1$Diff)), lag.max=40)
```



A função de autocorrelação indica padrão sazonal, então pode ser que seja

necessário diferenciar sazonalmente a série diferenciada para transformá-la em estacionária. Testamos a seguir a hipótese nula de não presença de raiz unitária sazonal com o teste de Canova e Hansen (Canova e Hansen 1995).

```
ch = ch.test(ts(na.omit(data1$Diff), frequency=12), type="dummy", sid=c(1:12))
print(ch)

##
## Canova and Hansen test for seasonal stability
##
## data: ts(na.omit(data1$Diff), frequency = 12)
##
##      statistic pvalue
## [1,]      1.3662 0.5465
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Test type: seasonal dummies
## NW covariance matrix lag order: 18
## First order lag: no
## Other regressors: no
## P-values: interpolation in original tables
```

O teste retornou um p-valor de 0.5465, o que significa que não podemos rejeitar a hipótese nula de não presença de raiz unitária sazonal. Agora que sabemos que não existe raiz unitária sazonal, testaremos, com o teste de Dickey-Fuller aumentado, a presença de raiz unitária não sazonal. Seguiremos os passos descritos acima, quando aplicamos o teste na série temporal original.

```
results <- matrix(,nrow=25,ncol=24)
for (i in 1:24){
  adf <- ur.df(na.omit(data1$Diff), type='drift', lags=i)
  for (e in 1:25){
    box <- Box.test(adf$res,lag=e)
    results[e,i] <- box$p.value
  }
}
```

Figura 15: Resultados dos Testes de Ljung-Box para os Lags 1 a 6

```
par(mfrow = c(3,2))
for (i in 1:6){
  plot(results[,i], main=i)
}
```

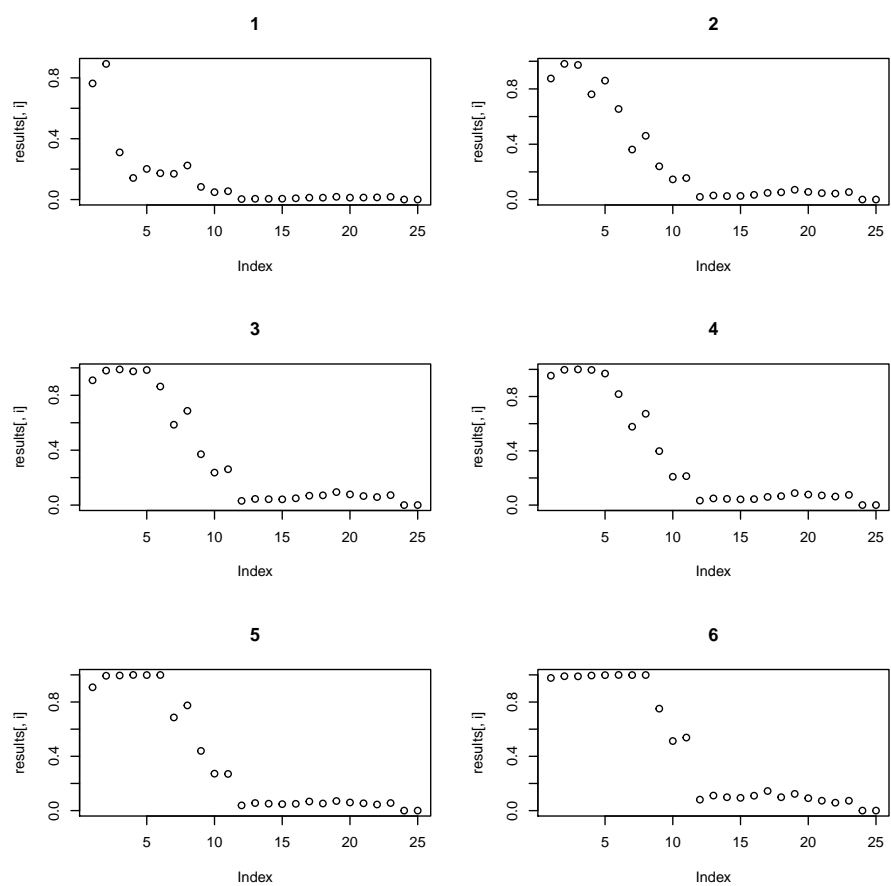


Figura 16: Resultados dos Testes de Ljung-Box para os Lags 7 a 12

```
par(mfrow = c(3,2))
for (i in 7:12){
  plot(results[,i], main=i)
}
```

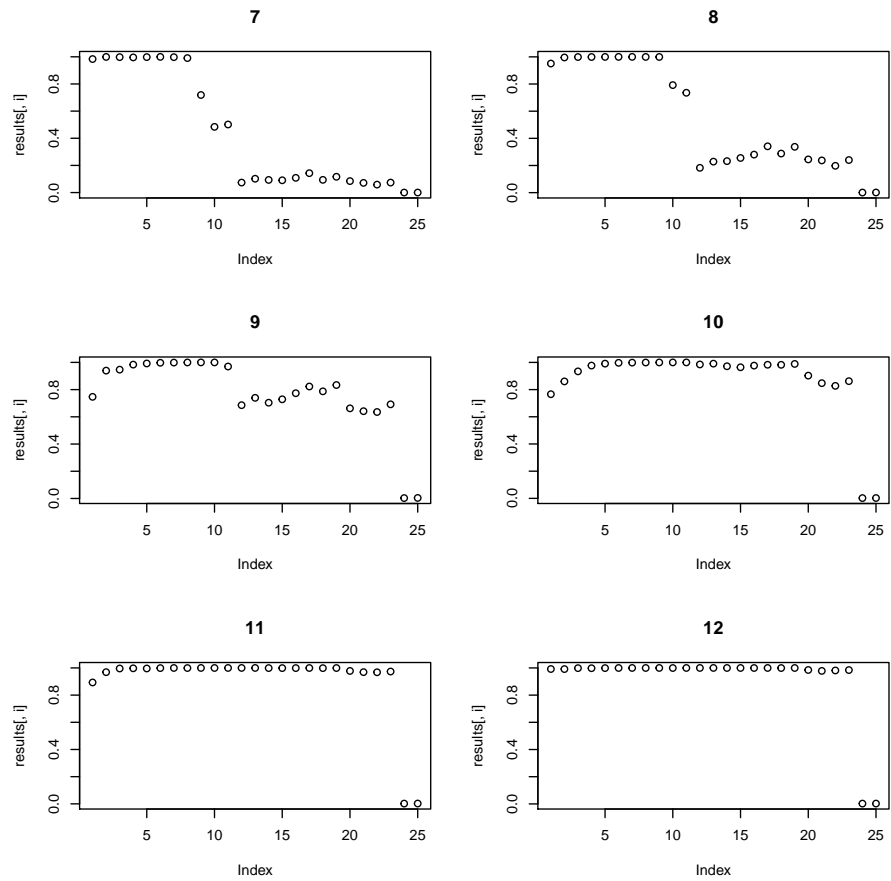


Figura 17: Resultados dos Testes de Ljung-Box para os Lags 13 a 18

```
par(mfrow = c(3,2))
for (i in 13:18){
  plot(results[,i], main=i)
}
```

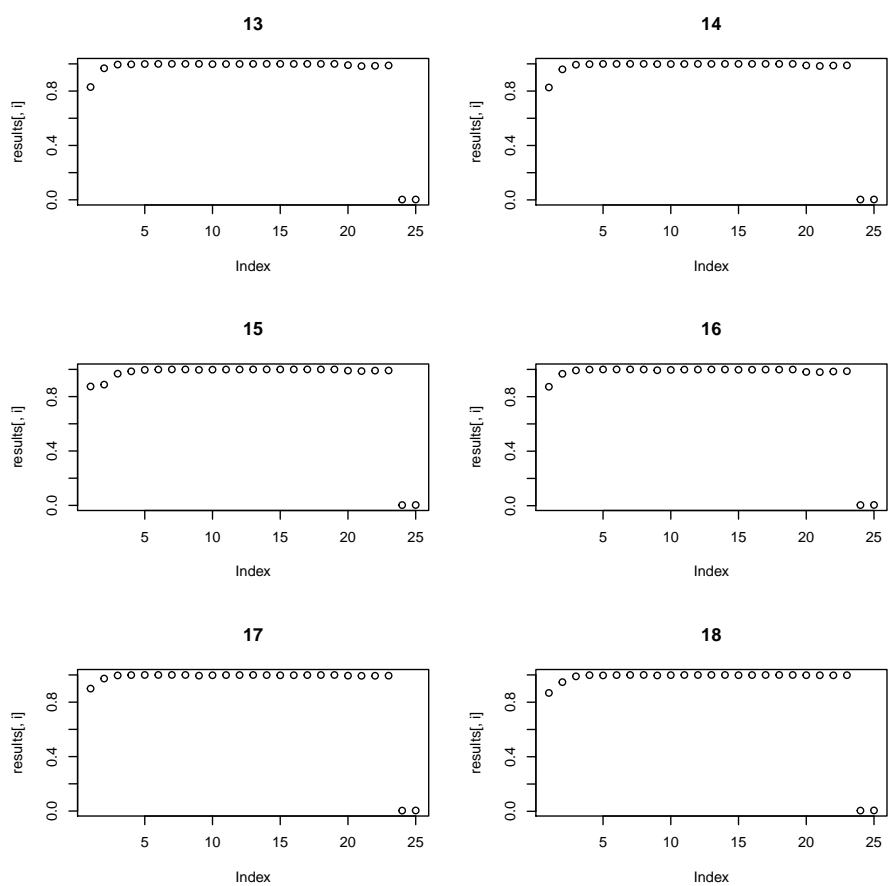
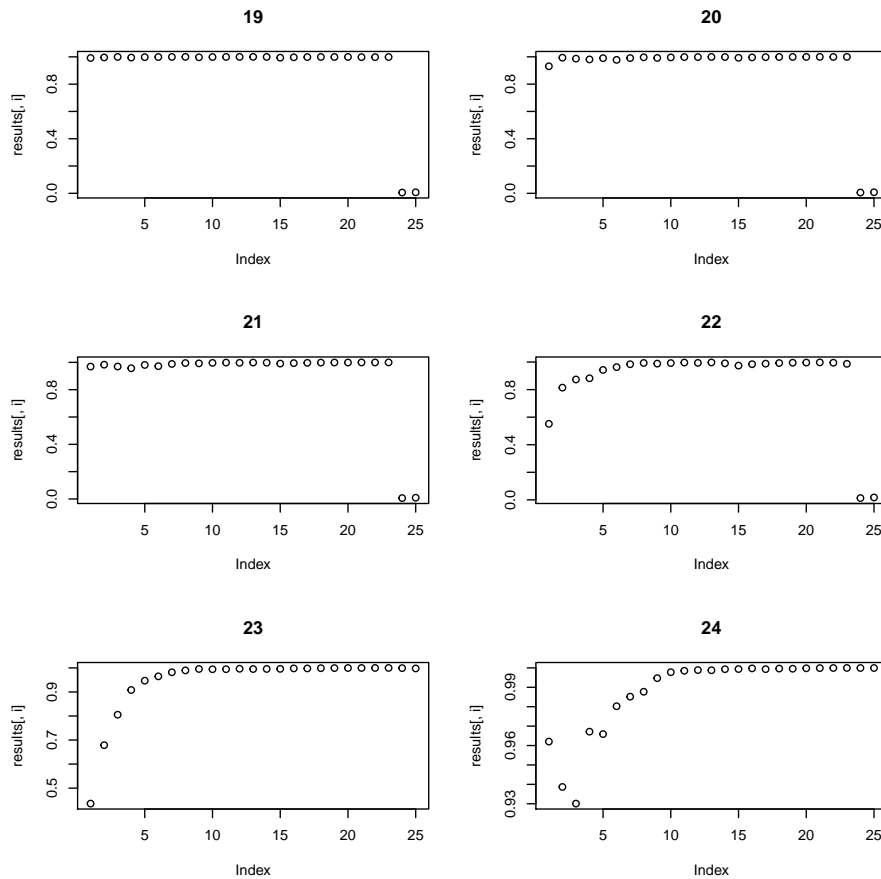


Figura 18: Resultados dos Testes de Ljung-Box para os Lags 19 a 24

```
par(mfrow = c(3,2))
for (i in 19:24){
  plot(results[,i], main=i)
}
```

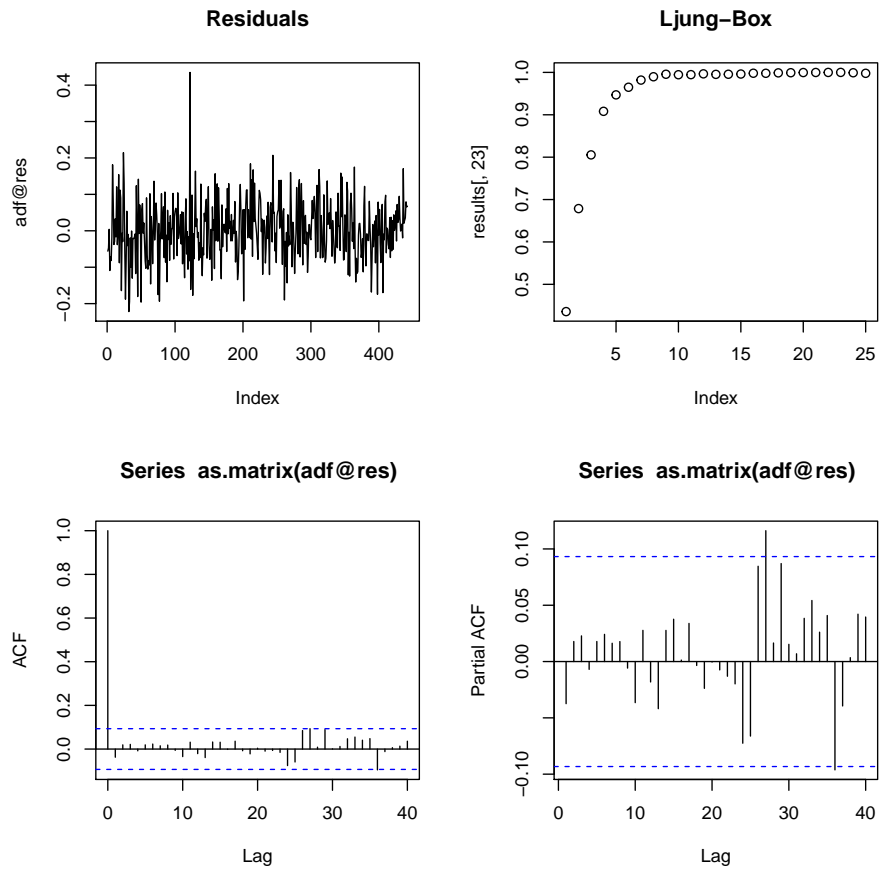


Nos gráficos acima podemos ver que a primeira ordem de lags que gera resíduos que são ruído branco é a ordem 23, então realizamos o teste com essa ordem e apresentamos os gráficos dos resíduos e dos testes nos resíduos.

```
adf <- ur.df(na.omit(data1$Diff), lags=23)
```

Figura 19: Resíduos

```
par(mfrow = c(2,2))
plot(adf@res, type='l', main='Residuals')
plot(results[,23], main='Ljung-Box')
acf(as.matrix(adf@res), lag.max=40)
pacf(as.matrix(adf@res), lag.max=40)
```



Tanto o gráfico dos resíduos quanto os gráficos das funções de autocorrelação e autocorrelação parcial indicam que os resíduos do teste se comportam como ruído branco. Os p-valores do teste de Ljung-Box não rejeitam a hipótese de independência dos resíduos.

Agora, visualizamos o sumário do teste.

```
summary(adf)

##
## #####
## # 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
## -0.22188 -0.05360 -0.00105  0.05312  0.43487
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## z.lag.1      -1.42558     0.27895  -5.111 4.90e-07 ***
## z.diff.lag1   0.17388     0.27409   0.634 0.526170
## z.diff.lag2  -0.02840     0.26795  -0.106 0.915656
## z.diff.lag3  -0.13271     0.26169  -0.507 0.612327
## z.diff.lag4  -0.07700     0.25629  -0.300 0.763987
## z.diff.lag5  -0.02814     0.25109  -0.112 0.910805
## z.diff.lag6   0.07834     0.24538   0.319 0.749681
## z.diff.lag7   0.18963     0.23904   0.793 0.428057
## z.diff.lag8   0.20060     0.23263   0.862 0.389015
## z.diff.lag9   0.27072     0.22598   1.198 0.231590
## z.diff.lag10  0.40889     0.21970   1.861 0.063433 .
## z.diff.lag11  0.49504     0.21314   2.323 0.020681 *
## z.diff.lag12  0.37274     0.20730   1.798 0.072884 .
## z.diff.lag13  0.35967     0.20260   1.775 0.076576 .
## z.diff.lag14  0.39096     0.19667   1.988 0.047473 *
## z.diff.lag15  0.34434     0.18902   1.822 0.069218 .
## z.diff.lag16  0.36352     0.17944   2.026 0.043414 *
## z.diff.lag17  0.35626     0.16889   2.109 0.035497 *
## z.diff.lag18  0.42467     0.15598   2.723 0.006748 **
## z.diff.lag19  0.45444     0.13990   3.248 0.001254 **
## z.diff.lag20  0.37292     0.12147   3.070 0.002280 **
## z.diff.lag21  0.29879     0.09860   3.030 0.002593 **
## z.diff.lag22  0.24201     0.07265   3.331 0.000941 ***
## z.diff.lag23  0.25544     0.04474   5.709 2.16e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

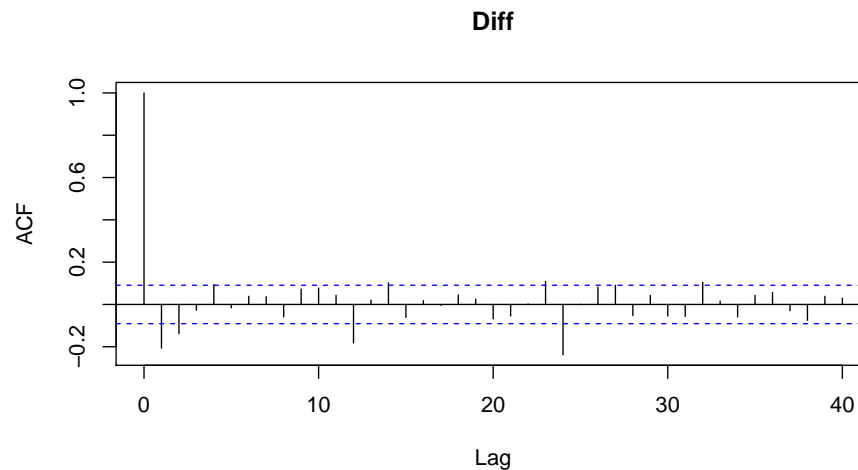
```
## Residual standard error: 0.08436 on 418 degrees of freedom
## Multiple R-squared:  0.6836, Adjusted R-squared:  0.6654
## F-statistic: 37.62 on 24 and 418 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -5.1105
##
## Critical values for test statistics:
##      1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
```

O valor da estatística do teste é de -5,1105. Os valores críticos para o teste são de -2,58 (1%), -1,95 (5%) e -1,62 (10%). Sendo assim, o valor do teste ultrapassou os valores críticos para todos os graus de significância. O teste então rejeita a hipótese nula de presença de raiz unitária. O resultado obtido é, então, que a série temporal original é integrada de ordem um - $I(1)$.

6.2 Identificação das Possíveis Formas Funcionais

Figura 20: FAC da Primeira Diferença

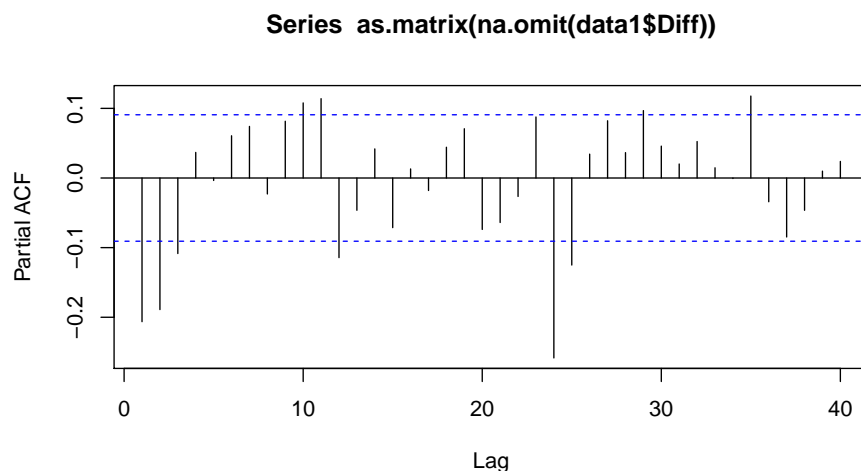
```
acf(as.matrix(na.omit(data1$Diff)), lag.max=40)
```



A função de autocorrelação tem até o segundo lag significativo, enquanto mostra um lag significativo no fator sazonal.

Figura 21: FACP da Primeira Diferença

```
pacf(as.matrix(na.omit(data1$Diff)), lag.max=40)
```



A função de autocorrelação parcial tem até o terceiro lag significativo, enquanto mostra dois lags significativos no fator sazonal.

A análise das funções acima sugere que uma ordem máxima para o modelo SARIMA é $(2,1,3)(1,0,2)_{12}$.

6.3 Estimação

Na subseção anterior definimos as ordens máximas do modelo SARIMA. Nesta seção vamos estimar todos os modelos até as ordens máximas, testar seus resíduos para checar se comportam-se como ruído branco e, finalmente, escolher o modelo que passa o teste dos resíduos que apresente melhor critério de informação.

Primeiramente, estimamos todos os modelos até as ordens máximas (72 modelos) e realizamos os testes de Ljung-Box.

[illegible]

```
results <- matrix(ncol=72,nrow=25)
for (i in 1:72){
  model <- sarima(data1$Value,
```

```

        ps[i], 1, qs[i],
        Ps[i], 0, Qs[i], 12)
  for (e in 1:25){
    box <- Box.test(model$fit$residuals, lag=e)
    results[e,i] <- box$p.value
  }
}

```

Em segundo lugar, escolhemos os modelos que passam no teste de Ljung-Box.

```

models <- c()
for (i in 1:72){
  if (min(results[,i]) > .05){
    models <- append(models, i)
  }
}
models

## [1] 2 4 14 16 26 28

```

Como podemos ver, apenas seis modelos apresentam resíduos com comportamento de ruído branco. Os estimamos novamente para avaliar o critério de informação.

```

aics <- c()
for (i in models){
  model <- sarima(data1$Value,
                  ps[i], 1, qs[i],
                  Ps[i], 0, Qs[i], 12)
  aics <- append(aics, model$AIC)
}

```

Abaixo, os critérios de informação de cada modelo.

```

models

## [1] 2 4 14 16 26 28

aics

## [1] -2.029506 -2.032146 -2.029555 -2.031749 -2.004273 -2.012301

```

Escolhemos, então, o modelo com menor critério de informação.

```

ind <- which.min(aics)
mod <- models[ind]
ps[mod]

```

```
## [1] 2

qs[mod]

## [1] 2

Ps[mod]

## [1] 1

Qs[mod]

## [1] 2
```

O modelo com melhor critério de informação é o segundo, que corresponde ao 4 na lista dos 72 modelos, que é o SARIMA (2,1,2)(1,0,2)₁₂. Estimamos então novamente esse modelo.

```
model1 <- sarima(data1$Value,
                 2, 1, 2,
                 1, 0, 2, 12)
```

Abaixo, o sumário do modelo.

```
print(model1)

## $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          sma1          sma2  constant
##          1.4362    -0.4843    -1.7137     0.7823    -0.2753    -0.0223    -0.3761     0.0005
## s.e.    0.1221     0.1340     0.0960     0.1061     0.1809     0.1706     0.0742     0.0027
##
## sigma^2 estimated as 0.007315:  log likelihood = 482.49,  aic = -946.98
##
## $degrees_of_freedom
## [1] 458
##
## $ttable
##          Estimate          SE  t.value p.value
## ar1          1.4362  0.1221   11.7620  0.0000
## ar2         -0.4843  0.1340   -3.6154  0.0003
```



```
## ma1      -1.7137 0.0960 -17.8433 0.0000
## ma2       0.7823 0.1061  7.3758 0.0000
## sar1      -0.2753 0.1809 -1.5216 0.1288
## sma1      -0.0223 0.1706 -0.1305 0.8962
## sma2      -0.3761 0.0742 -5.0678 0.0000
## constant  0.0005 0.0027  0.1882 0.8508
##
## $AIC
## [1] -2.032146
##
## $AICc
## [1] -2.03147
##
## $BIC
## [1] -1.952108
```

6.4 Diagnóstico dos Resíduos

Nesta seção iremos fazer o diagnóstico dos resíduos. Para isso vamos analisar sua independência, homoscedasticidade e distribuição.

6.4.1 Independência

Para analisar a independência dos resíduos analisamos o gráfico dos resíduos, a função de autocorrelação e a função de autocorrelação parcial.

Figura 22: Resíduos

```
plot(model1$fit$residuals)
```

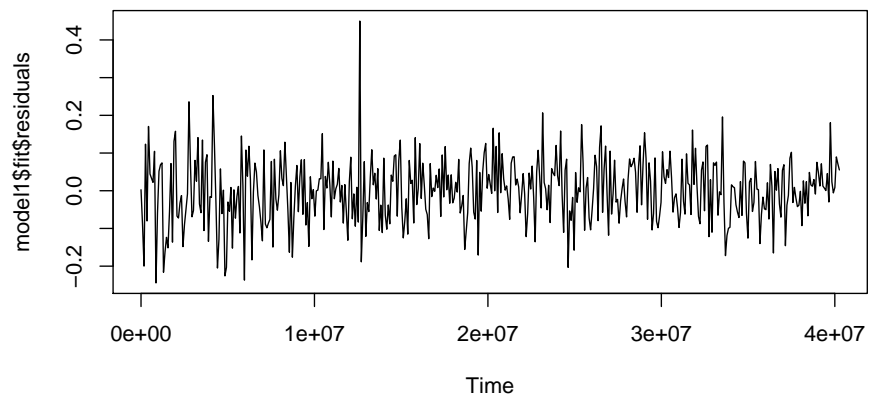


Figura 23: FAC dos Resíduos

```
acf(as.matrix(model1$fit$residuals), lag.max=40)
```

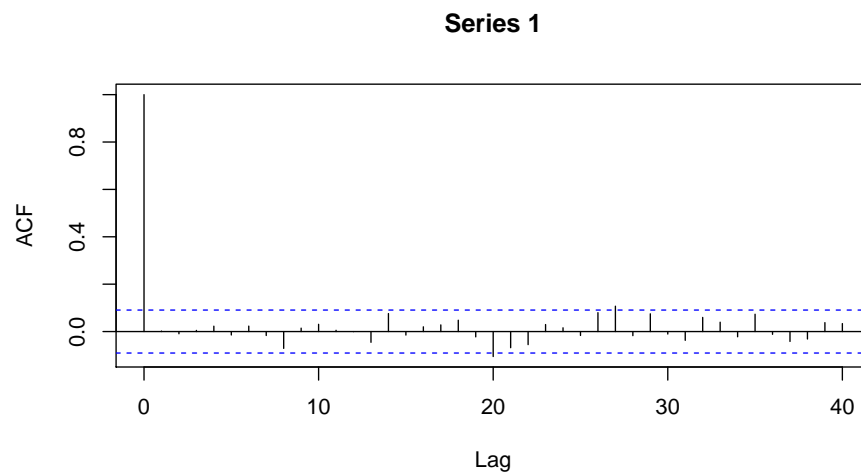
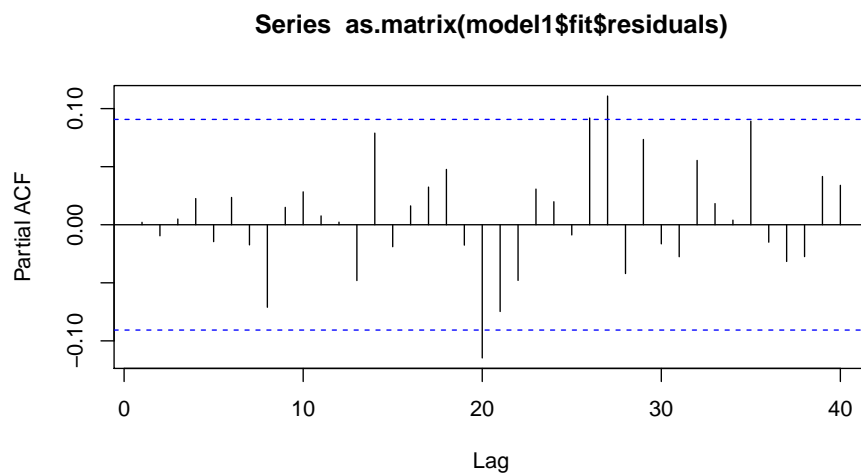


Figura 24: FACP dos Resíduos

```
pacf(as.matrix(model1$fit$residuals), lag.max=40)
```

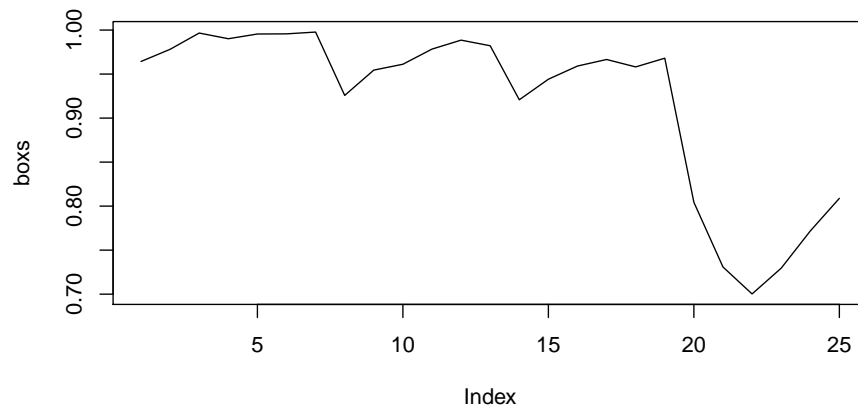


Tanto o gráfico dos resíduos quanto a sua função de autocorrelação e sua função de autocorrelação parcial indicam que os resíduos se comportam como ruído branco. Para testar essa hipótese, realizamos o teste de Ljung-Box para todos os lags até o vinte e cinco. Os testes são realizados no *chunk* abaixo, e na figura abaixo estão os p-valores dos testes para cada lag.

```
boxs <- matrix(nrow=25,ncol=1)
for (i in 1:25){
  box <- Box.test(model1$fit$residuals,lag=i)
  boxs[i] <- box$p.value
}
```

Figura 25: P-Valores de Ljung-Box

```
plot(boxs, type='l')
```



Os p-valores não rejeitam a hipótese nula de independência da distribuição.

6.4.2 Homoscedasticidade

Testamos a homoscedasticidade dos resíduos aplicando a análise acima no resíduo ao quadrado.

Figura 26: Resíduos ao Quadrado

```
plot(model1$fit$residuals^2)
```

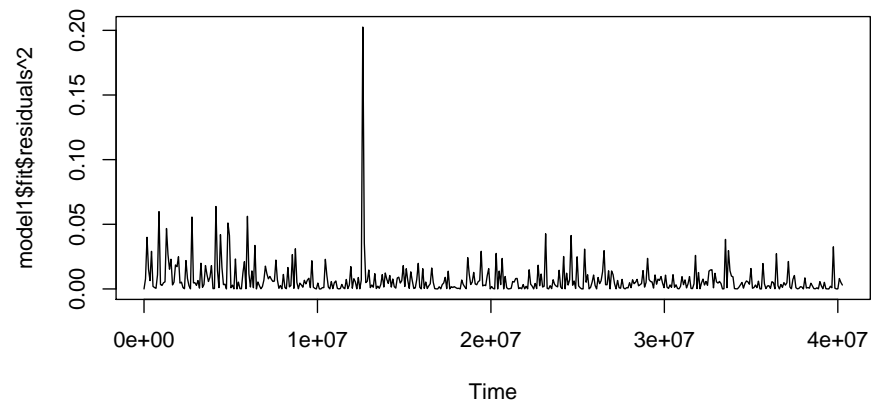


Figura 27: FAC dos Resíduos ao Quadrado

```
acf(as.matrix(model1$fit$residuals)^2, lag.max=40)
```

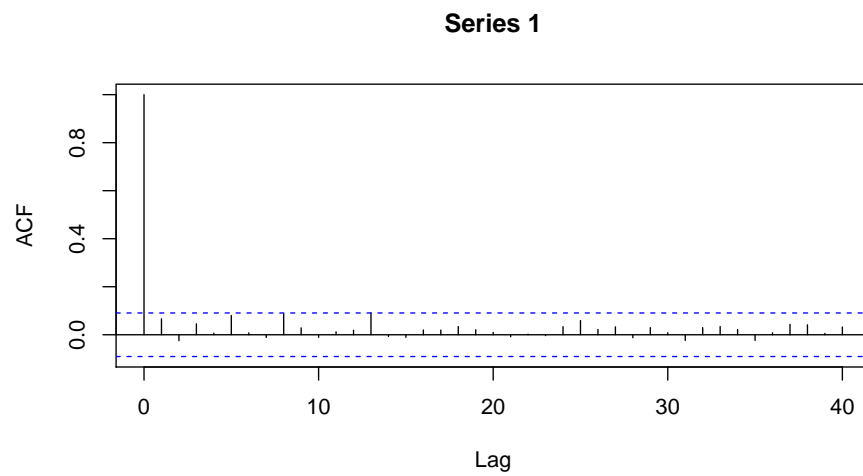
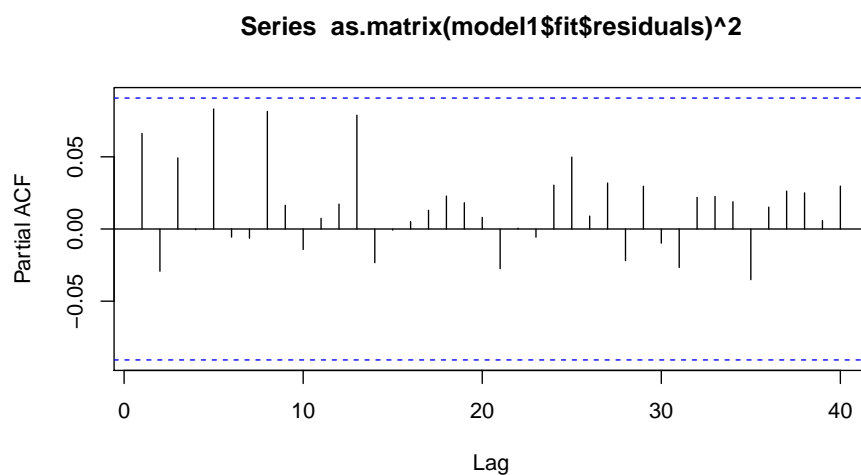


Figura 28: FACP dos Resíduos ao Quadrado

```
pacf(as.matrix(model1$fit$residuals)^2, lag.max=40)
```

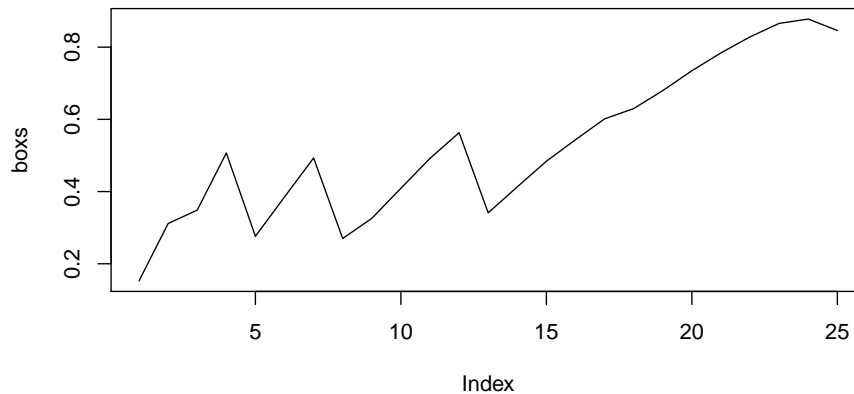


Tanto o gráfico dos resíduos ao quadrado quanto a sua função de autocorrelação e sua função de autocorrelação parcial indicam homoscedasticidade. Para testar essa hipótese, realizamos o teste de Ljung-Box nos resíduos ao quadrado para todos os lags até o vinte e cinco. Os testes são realizados no *chunk* abaixo, e na figura abaixo estão os p-valores dos testes para cada lag.

```
boxs <- matrix(nrow=25,ncol=1)
for (i in 1:25){
  box <- Box.test(model1$fit$residuals^2,lag=i)
  boxs[i] <- box$p.value
}
```

Figura 29: P-Valores de Ljung-Box

```
plot(boxs, type='l')
```



Os p-valores não rejeitam a hipótese nula de independência da distribuição. Os resíduos podem ser considerados então homoscedásticos, pois os resíduos ao quadrado são ‘bem comportados’ (ruído branco). Não é necessário então estimar um modelo de variância condicional.

6.4.3 Normalidade

Para testar a normalidade dos resíduos aplicamos o teste de Shapiro-Wilk (SHAPIRO e WILK 1965).

```
shapiro.test(model1$fit$residuals)

##
##  Shapiro-Wilk normality test
##
## data:  model1$fit$residuals
## W = 0.98772, p-value = 0.0005713
```

O teste rejeita a hipótese nula de normalidade dos resíduos.

6.5 Previsão e Acurácia

Nesta subseção faremos previsões e testaremos a acurácia das previsões feitas, como passo na avaliação do modelo estimado.

6.5.1 Previsão

Agora, realizaremos previsões para os últimos 20 períodos da amostra. Faremos previsão recursiva, ou seja: prevemos sempre apenas o período imediatamente subsequente, utilizando todos os dados até então.

```
fs <- c()
for (i in 20:1){
  until <- length(data1$Value)-i
  data <- data1$Value[1:until]
  prediction <- sarima.for(data, 1,
                           2, 1, 2,
                           1, 0, 2, 12)
  fs <- append(fs, prediction$pred)
}
```

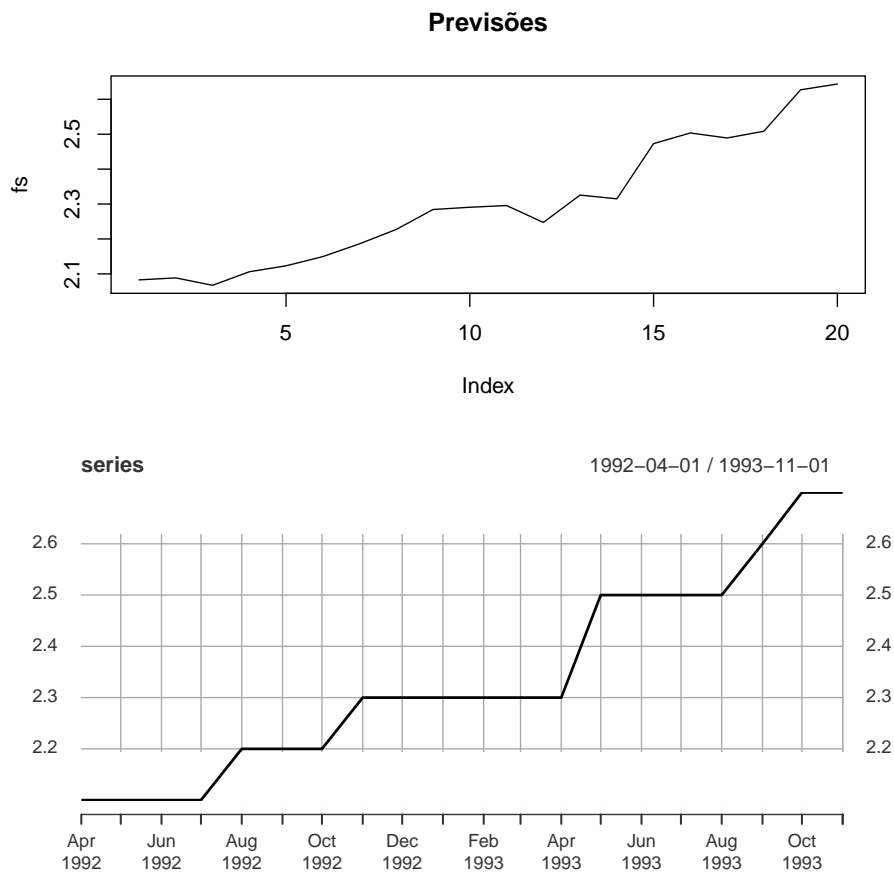
Abaixo, as previsões realizadas.

```
fs
## [1] 2.082725 2.088396 2.067254 2.105981 2.123054 2.149433 2.186097 2.227275
## [9] 2.284345 2.290782 2.295716 2.247239 2.325614 2.314883 2.473123 2.503603
## [17] 2.489219 2.508543 2.627183 2.643718
```

Abaixo, os gráficos das previsões e da série original.

Figura 30: Previsões e Série Original

```
length <- length(data1$Value)
start <- length - 19
series <- data1$Value[start:length]
par(mfrow = c(2,1))
plot(fs,type='l', main='Previsões')
plot(series)
```



6.5.2 Acurácia

Agora, vamos calcular a acurácia das previsões. Utilizaremos cinco medidas alternativas: erro médio (ME); erro quadrático médio (RMSE); erro médio absoluto (MAE); erro de porcentagem média (MPE); erro de porcentagem média absoluta (MAPE).

```
accuracy(fs,as.matrix(series))
```

```
##           ME           RMSE           MAE           MPE           MAPE
## Test set 0.03829089 0.05968404 0.04181064 1.58768 1.741936
```

As medidas acima podem ser úteis na escolha entre modelos, mas não nos dizem muito sobre o modelo se não temos outro para comparar. Para isso usamos o cálculo do Theil's U, que compara as previsões com o que seria uma 'adivinhação'. Caso o valor do cálculo for menor que 1, as previsões são melhores que adivinhação. Caso for maior, são piores. O teste é realizado abaixo.

```
TheilU(series,fs)
```

```
## [1] 0.02542164
```

O índice do teste foi de 0,02542164, menor que 1. O modelo fez previsões melhores que uma simples adivinhação.

7 Segunda Série Temporal

7.1 Definição da Ordem de Integração

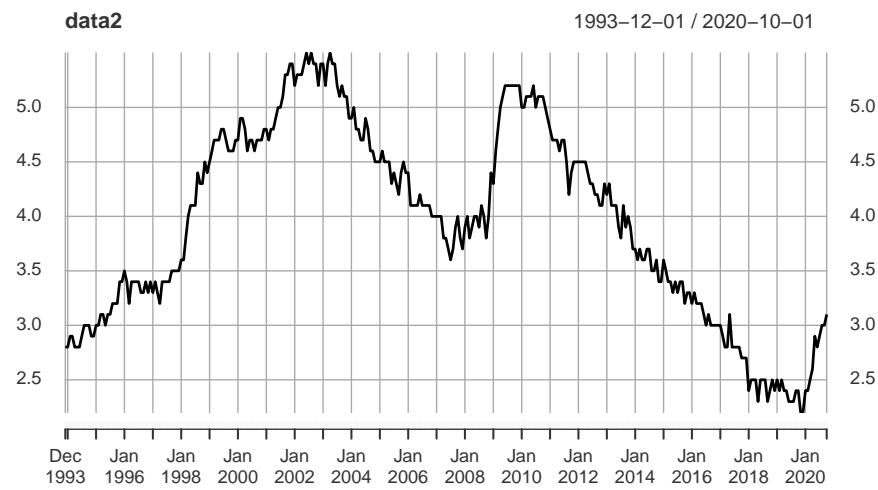
O primeiro passo na metodologia Box-Jenkins (Box et al. 2015) é a definição da ordem de integração da série temporal .

7.1.1 Análise da Série Original

Começamos o processo de definição da ordem de integração analisando o gráfico da série temporal.

Figura 31: Série Temporal

```
plot(data2)
```



A visualização da série temporal indica presença de raiz unitária. Para coletar mais indícios visuais analisamos abaixo a função de autocorrelação.

Figura 32: FAC da Série Temporal

```
acf(as.matrix(data2), lag.max=40)
```

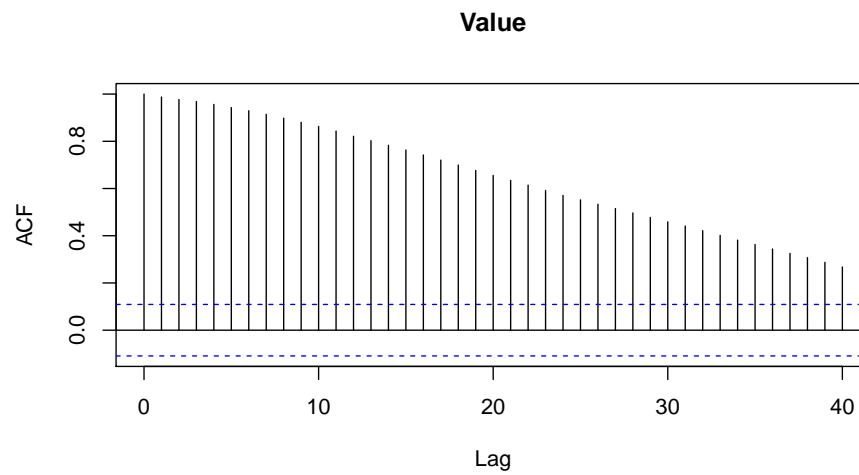
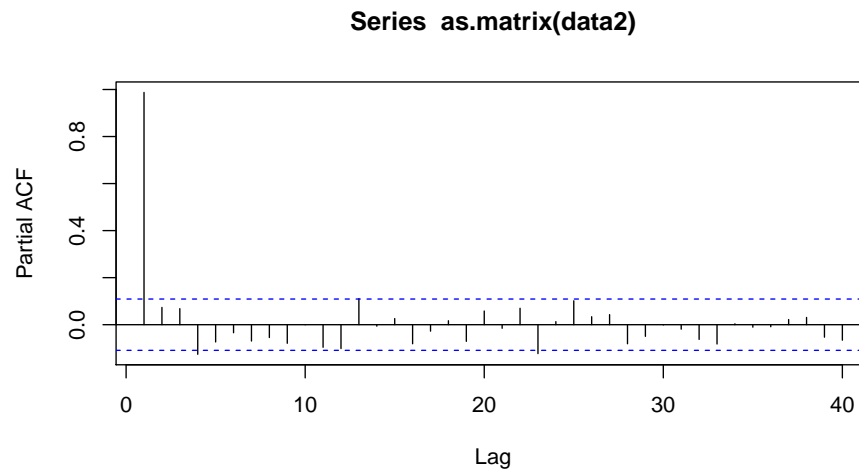


Figura 33: FACP da Série Temporal

```
pacf(as.matrix(data2), lag.max=40)
```



A função de autocorrelação aparentemente possui decaimento linear, indi-

cando possibilidade de raiz unitária. Para testar a hipótese de presença de raiz unitária com o teste Dickey-Fuller aumentado (Dickey e Fuller 1979). O teste será realizado com *drift*, pois a visualização do gráfico da série temporal indica a presença de tal. Para escolher o lag do teste, começaremos pelo lag 1 e, caso os resíduos do teste forem ruído branco, aceitamos o lag. Caso os resíduos não apresentarem comportamento de ruído branco, repetimos os passos com o lag imediatamente maior. No *chunk* abaixo realizamos testes ADF para 24 lags, e para cada teste testamos os resíduos com testes de Ljung-Box (Ljung e Box 1978) até 25 lags.

```
results <- matrix(,nrow=25,ncol=24)
for (i in 1:24){
  adf <- ur.df(data2$Value, type='drift', lags=i)
  for (e in 1:25){
    box <- Box.test(adf$res, lag=e)
    results[e,i] <- box$p.value
  }
}
```

Nas figuras abaixo encontram-se os resultados dos testes de Ljung-Box para os resíduos de cada um dos testes de Dickey-Fuller aumentados.

Figura 34: Resultados dos Testes de Ljung-Box para os Lags 1 a 6

```
par(mfrow = c(3,2))
for (i in 1:6){
  plot(results[,i], main=i)
}
```

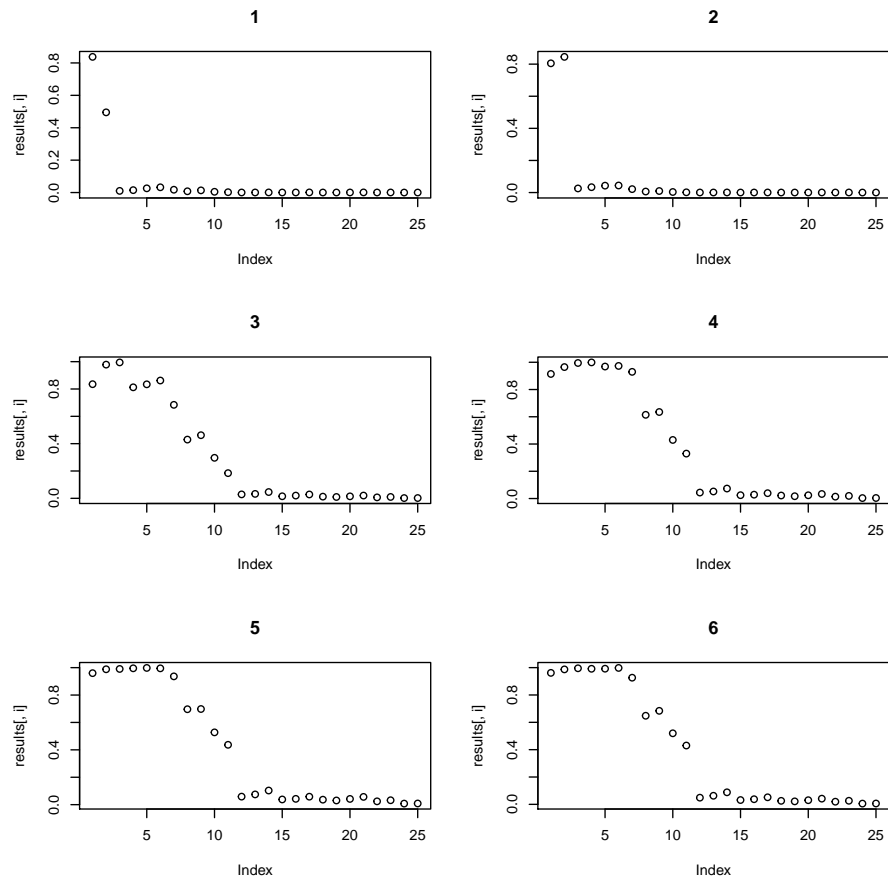


Figura 35: Resultados dos Testes de Ljung-Box para os Lags 7 a 12

```
par(mfrow = c(3,2))
for (i in 7:12){
  plot(results[,i], main=i)
}
```

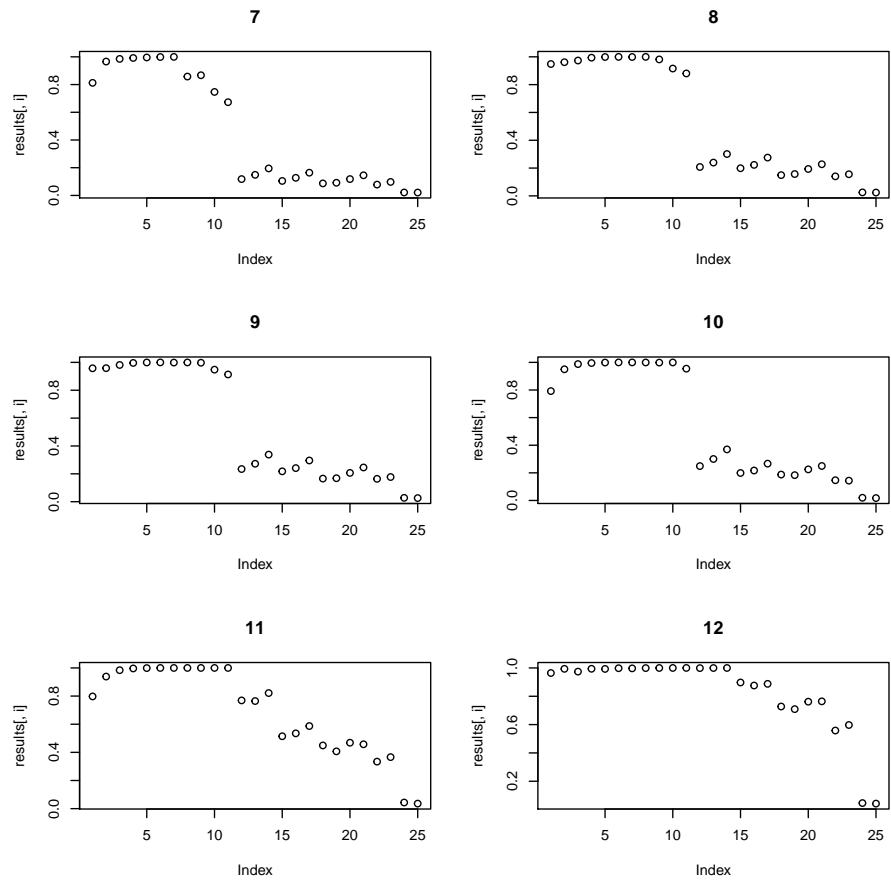


Figura 36: Resultados dos Testes de Ljung-Box para os Lags 13 a 18

```
par(mfrow = c(3,2))
for (i in 13:18){
  plot(results[,i], main=i)
}
```

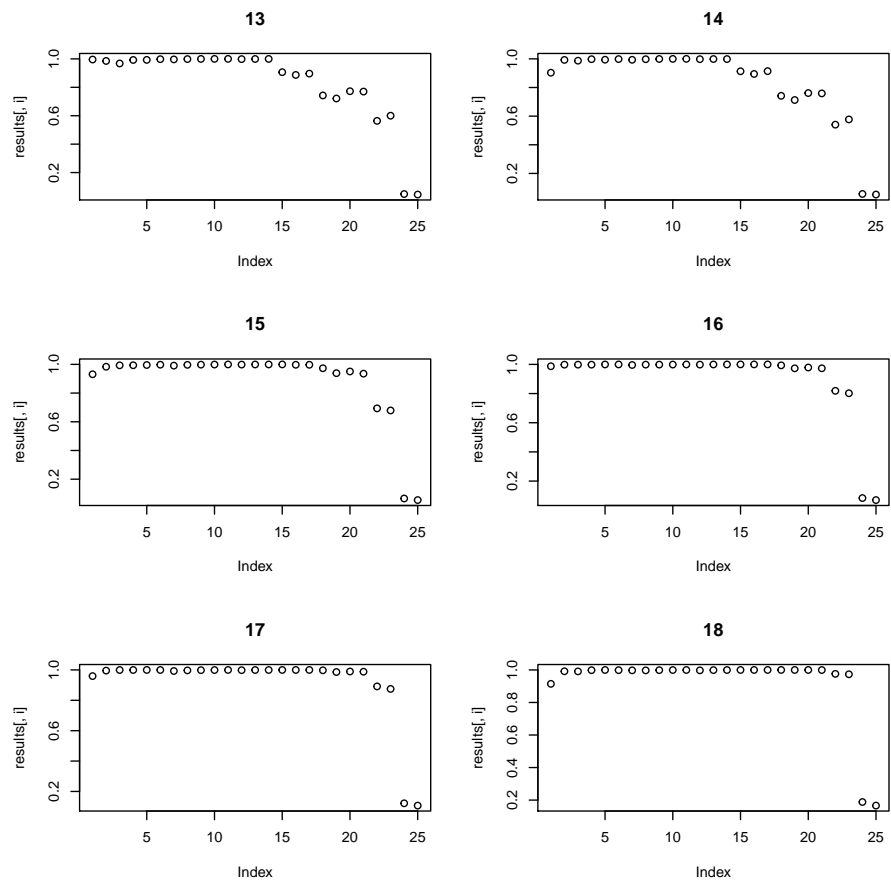
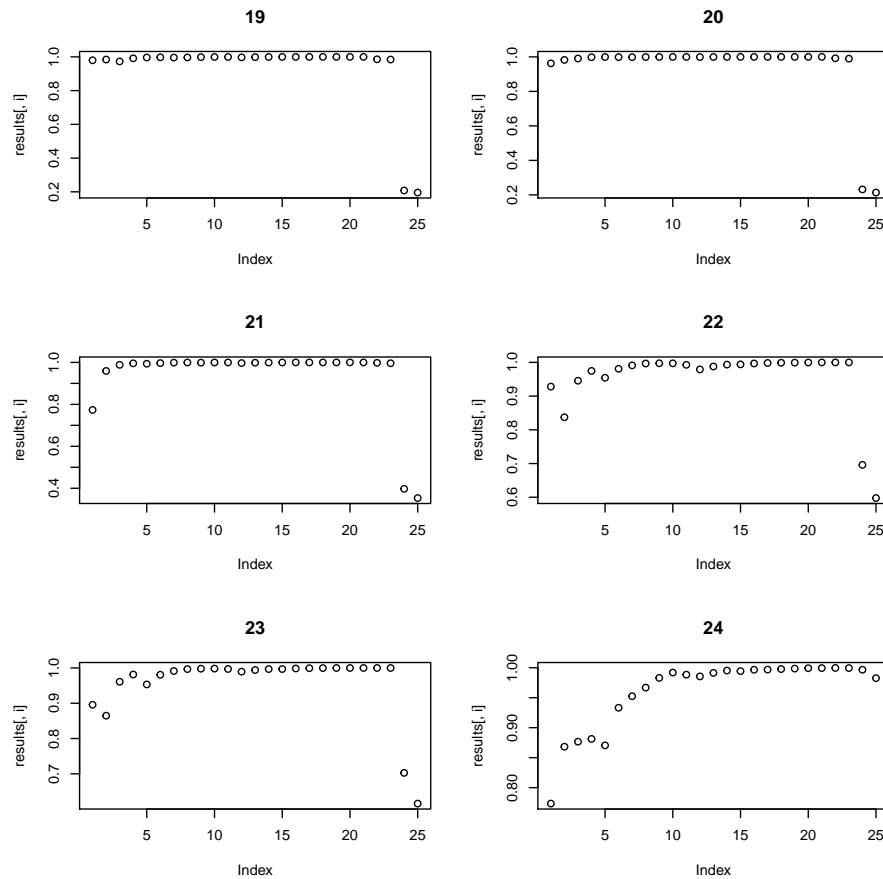


Figura 37: Resultados dos Testes de Ljung-Box para os Lags 19 a 24

```
par(mfrow = c(3,2))
for (i in 19:24){
  plot(results[,i], main=i)
}
```

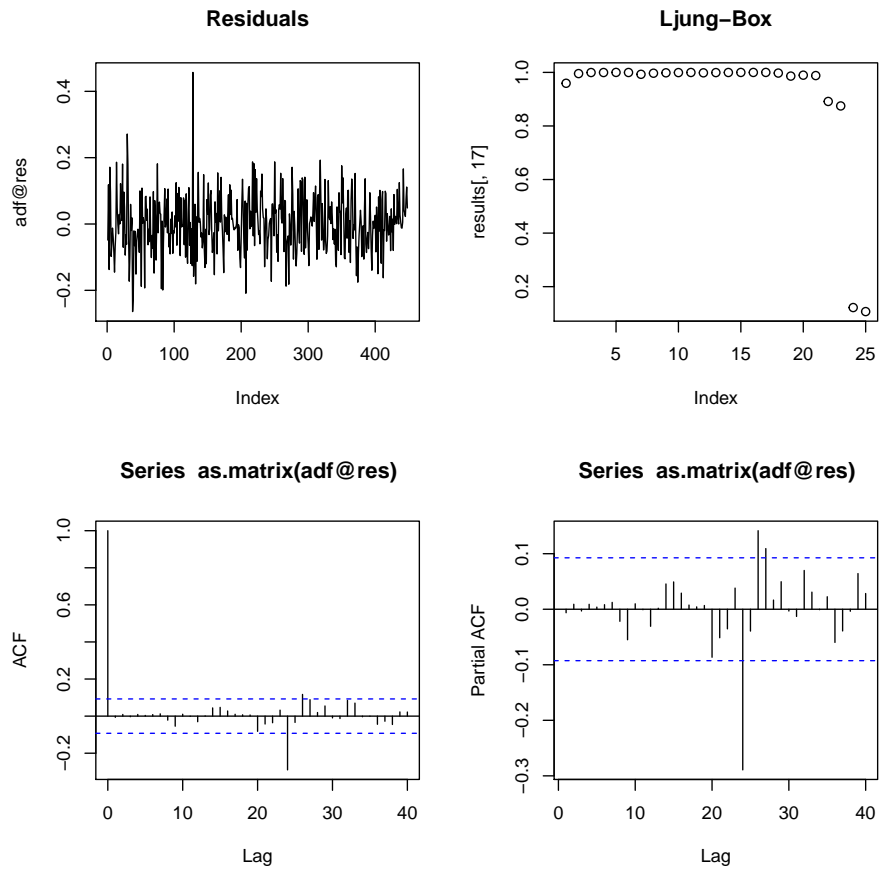


Nos gráficos acima podemos ver que a única ordem de lags que gera resíduos que são ruído branco é a ordem 17, então realizamos o teste com essa ordem e apresentamos os gráficos dos resíduos e dos testes nos resíduos.

```
adf <- ur.df(na.omit(data1$Diff), lags=17)
```

Figura 38: Resíduos

```
par(mfrow = c(2,2))
plot(adf@res, type='l', main='Residuals')
plot(results[,17], main='Ljung-Box')
acf(as.matrix(adf@res), lag.max=40)
pacf(as.matrix(adf@res), lag.max=40)
```



Tanto o gráfico dos resíduos quanto os gráficos das funções de autocorrelação e autocorrelação parcial indicam que os resíduos do teste se comportam como ruído branco. Os p-valores do teste de Ljung-Box não rejeitam a hipótese de independência dos resíduos. Agora, apresentamos o sumário do teste.

```
summary(adf)
##
```

```

## #####
## # 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
## -0.26414 -0.05624  0.00072  0.05925  0.45708
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## z.lag.1      -1.13700     0.25139  -4.523  7.9e-06 ***
## z.diff.lag1  -0.14668     0.24511  -0.598   0.5499
## z.diff.lag2  -0.37154     0.23879  -1.556   0.1205
## z.diff.lag3  -0.46688     0.23261  -2.007   0.0454 *
## z.diff.lag4  -0.39939     0.22732  -1.757   0.0796 .
## z.diff.lag5  -0.36377     0.22132  -1.644   0.1010
## z.diff.lag6  -0.26499     0.21563  -1.229   0.2198
## z.diff.lag7  -0.16782     0.21082  -0.796   0.4265
## z.diff.lag8  -0.14821     0.20483  -0.724   0.4697
## z.diff.lag9  -0.03067     0.19716  -0.156   0.8765
## z.diff.lag10  0.09369     0.18732   0.500   0.6172
## z.diff.lag11  0.18400     0.17636   1.043   0.2974
## z.diff.lag12  0.08676     0.16318   0.532   0.5952
## z.diff.lag13  0.04422     0.14647   0.302   0.7629
## z.diff.lag14  0.05164     0.12688   0.407   0.6842
## z.diff.lag15 -0.02298     0.10257  -0.224   0.8228
## z.diff.lag16 -0.02397     0.07521  -0.319   0.7501
## z.diff.lag17 -0.04025     0.04625  -0.870   0.3847
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08864 on 430 degrees of freedom
## Multiple R-squared:  0.6542, Adjusted R-squared:  0.6397
## F-statistic: 45.2 on 18 and 430 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -4.5229
##
## Critical values for test statistics:

```

```
##      1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
```

O valor da estatística do teste é de -2,2007. Os valores críticos para o teste são de -3,44 (1%), -2,87 (5%) e -2,57 (10%). Sendo assim, o valor do teste não ultrapassou os valores críticos para nenhum grau de significância. O teste então aceita a hipótese nula de presença de raiz unitária.

7.1.2 Diferenciação

Como tentativa para estacionarizar a série, aplicamos a primeira diferença.

```
data2$Diff <- diff(data2)
```

Abaixo, o sumário dos novos dados.

```
summary(data2)
```

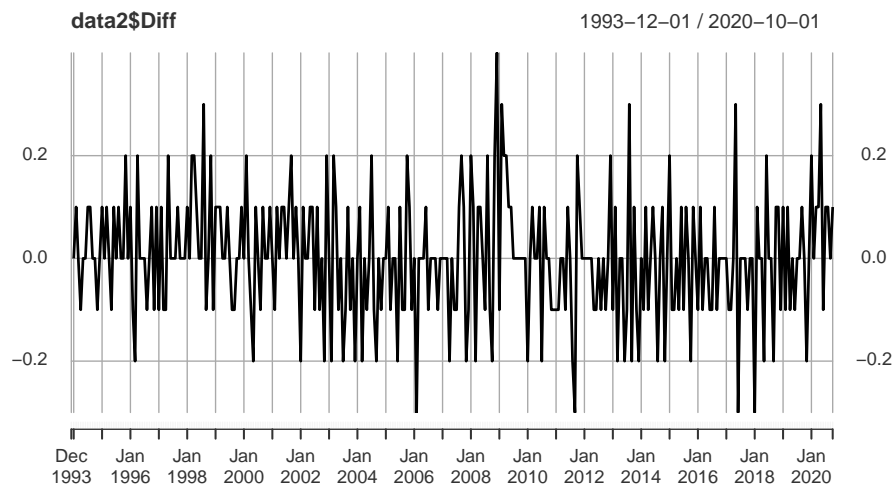
##	Index	Value	Diff
##	Min. :1993-12-01 00:00:00	Min. :2.200	Min. :-0.3000000
##	1st Qu.:2000-08-16 12:00:00	1st Qu.:3.200	1st Qu.: -0.1000000
##	Median :2007-05-01 00:00:00	Median :4.000	Median : 0.0000000
##	Mean :2007-05-02 04:00:44	Mean :3.948	Mean : 0.0009317
##	3rd Qu.:2014-01-16 12:00:00	3rd Qu.:4.700	3rd Qu.: 0.1000000
##	Max. :2020-10-01 00:00:00	Max. :5.500	Max. : 0.4000000
##			NA's :1

7.1.3 Análise da Primeira Diferença

Começamos a nova análise analisando o gráfico da primeira diferença da série temporal.

Figura 39: Primeira Diferença

```
plot(data2$Diff)
```



A visualização da série temporal não indica presença de raiz unitária. Para coletar mais indícios visuais analisamos abaixo a função de autocorrelação.

Figura 40: FAC da Primeira Diferença

```
acf(as.matrix(na.omit(data2$Diff)), lag.max=40)
```

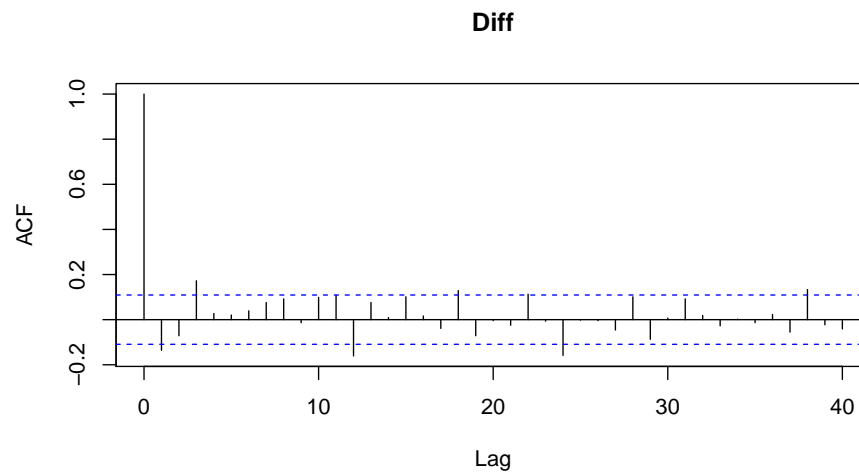
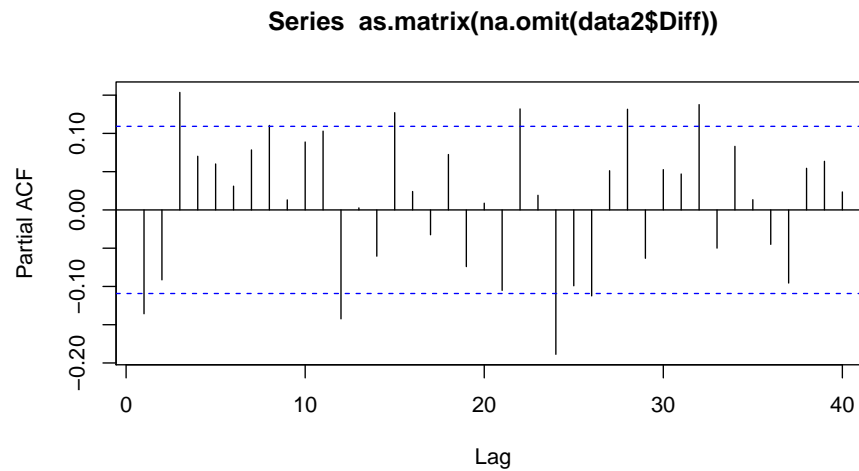


Figura 41: FACP da Primeira Diferença

```
pacf(as.matrix(na.omit(data2$Diff)), lag.max=40)
```



A função de autocorrelação indica padrão sazonal, então pode ser que seja

necessário diferenciar sazonalmente a série diferenciada para transformá-la em estacionária. Testamos a seguir a hipótese nula de não presença de raiz unitária sazonal com o teste de Canova e Hansen (Canova e Hansen 1995).

```
ch = ch.test(ts(na.omit(data2$Diff), frequency=12), type="dummy", sid=c(1:12))
print(ch)

##
## Canova and Hansen test for seasonal stability
##
## data: ts(na.omit(data2$Diff), frequency = 12)
##
##      statistic pvalue
## [1,]      1.1697 0.6117
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Test type: seasonal dummies
## NW covariance matrix lag order: 16
## First order lag: no
## Other regressors: no
## P-values: interpolation in original tables
```

O teste retornou um p-valor de 0.6317, o que significa que não podemos rejeitar a hipótese nula de não presença de raiz unitária sazonal. Agora que sabemos que não existe raiz unitária sazonal, testaremos, com o teste de Dickey-Fuller aumentado, a presença de raiz unitária não sazonal. Desta vez não adicionamos *drift* ao modelo, pois a visualização dos dados não sugere presença de *drift*. Seguiremos os passos descritos acima, quando aplicamos o teste na série temporal original.

```
results <- matrix(nrow=25,ncol=24)
for (i in 1:24){
  adf <- ur.df(na.omit(data2$Diff), type='drift', lags=i)
  for (e in 1:25){
    box <- Box.test(adf$res,lag=e)
    results[e,i] <- box$p.value
  }
}
```

Figura 42: Resultados dos Testes de Ljung-Box para os Lags 1 a 6

```
par(mfrow = c(3,2))
for (i in 1:6){
  plot(results[,i], main=i)
}
```

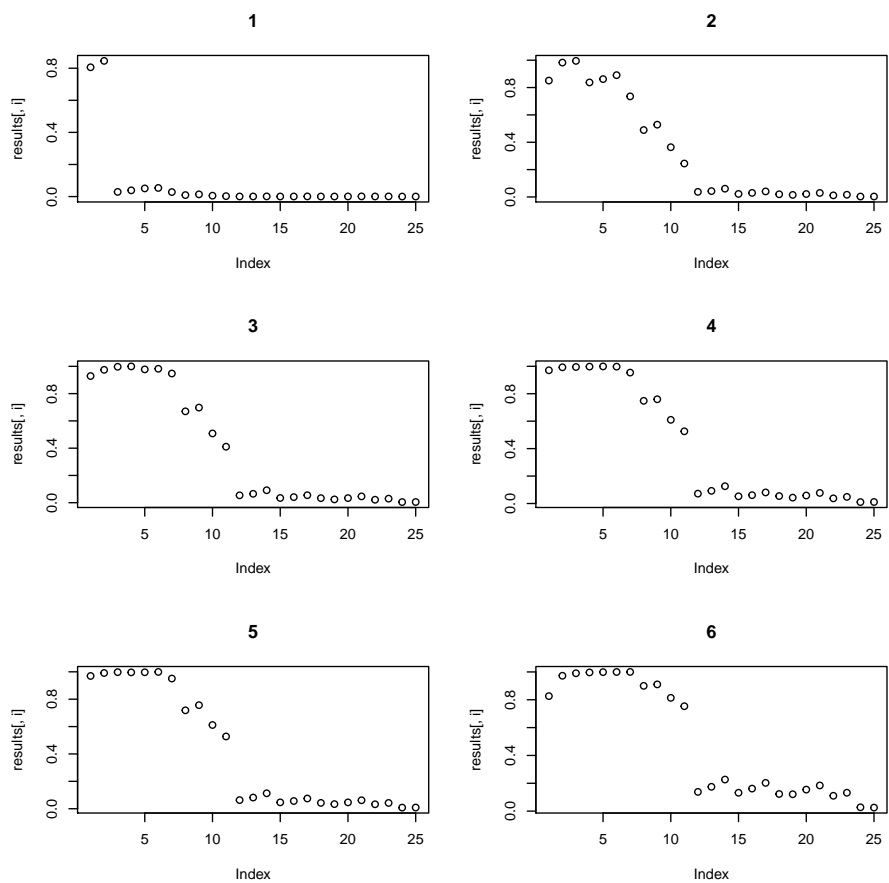


Figura 43: Resultados dos Testes de Ljung-Box para os Lags 7 a 12

```
par(mfrow = c(3,2))
for (i in 7:12){
  plot(results[,i], main=i)
}
```

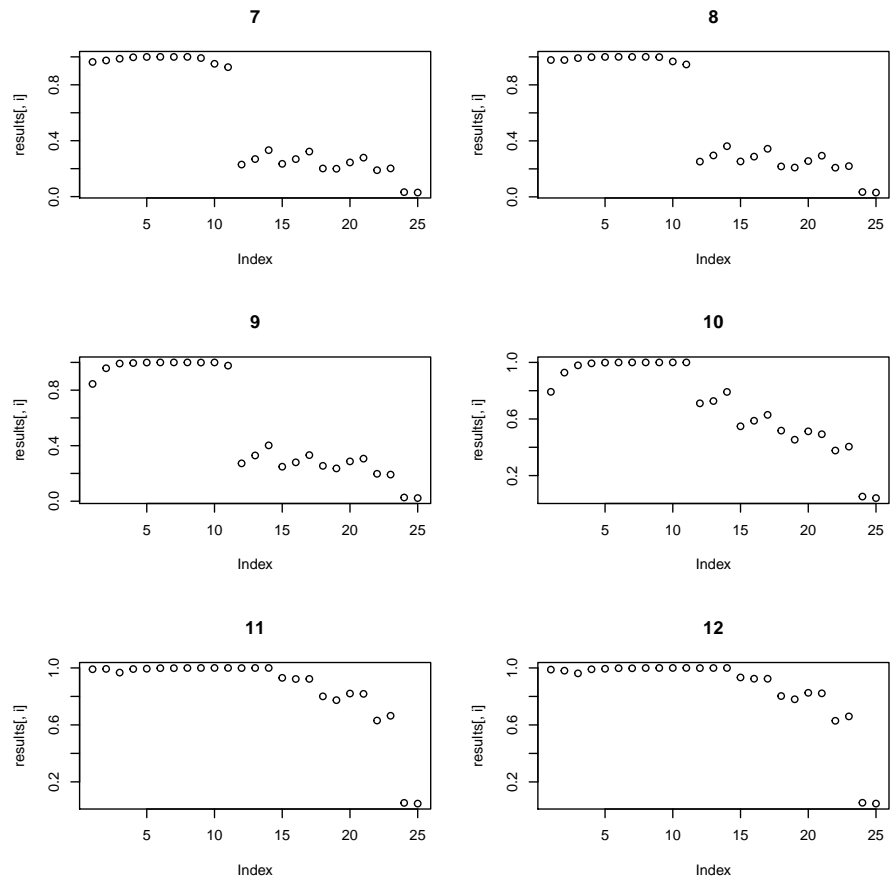


Figura 44: Resultados dos Testes de Ljung-Box para os Lags 13 a 18

```
par(mfrow = c(3,2))
for (i in 13:18){
  plot(results[,i], main=i)
}
```

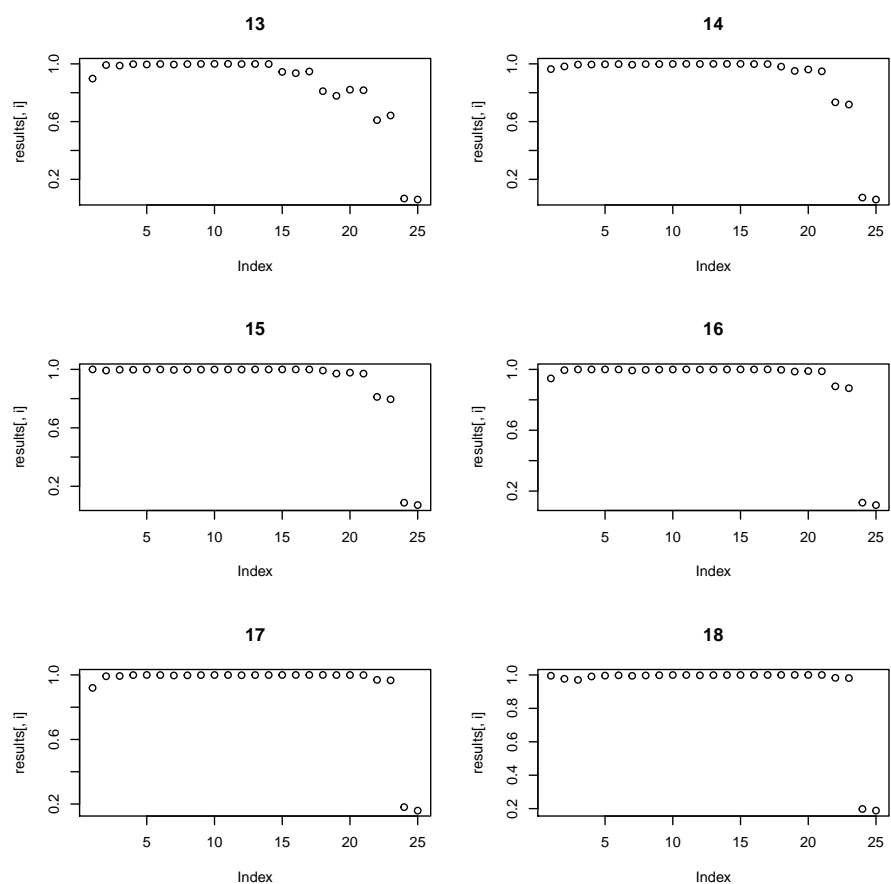
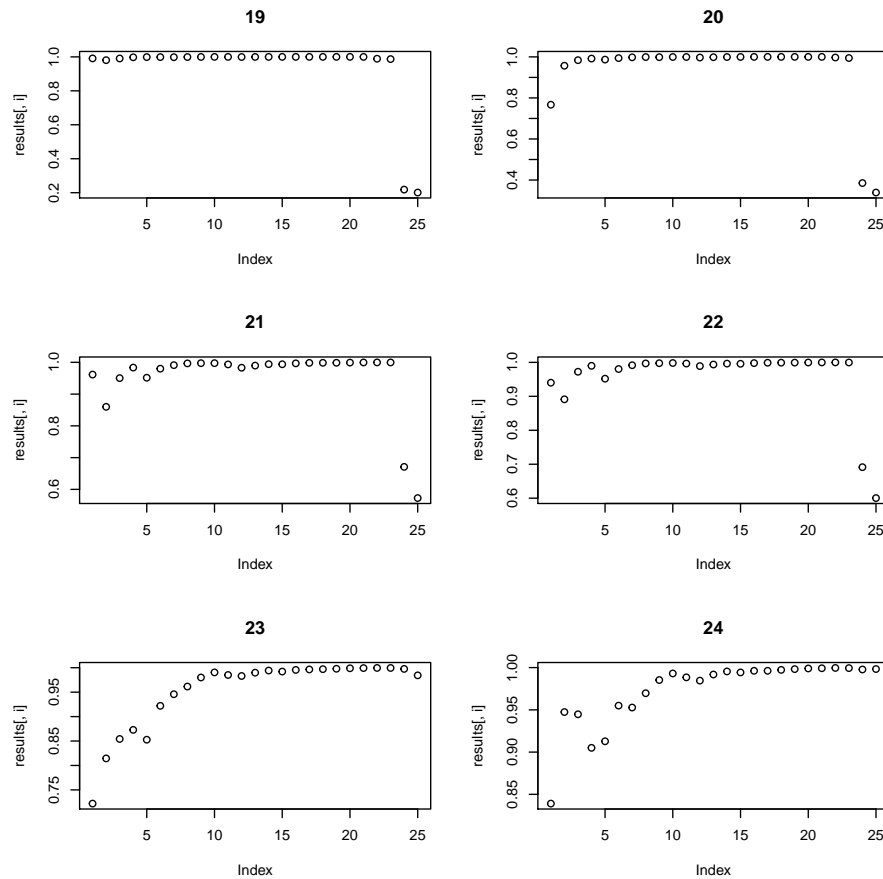


Figura 45: Resultados dos Testes de Ljung-Box para os Lags 19 a 24

```
par(mfrow = c(3,2))
for (i in 19:24){
  plot(results[,i], main=i)
}
```

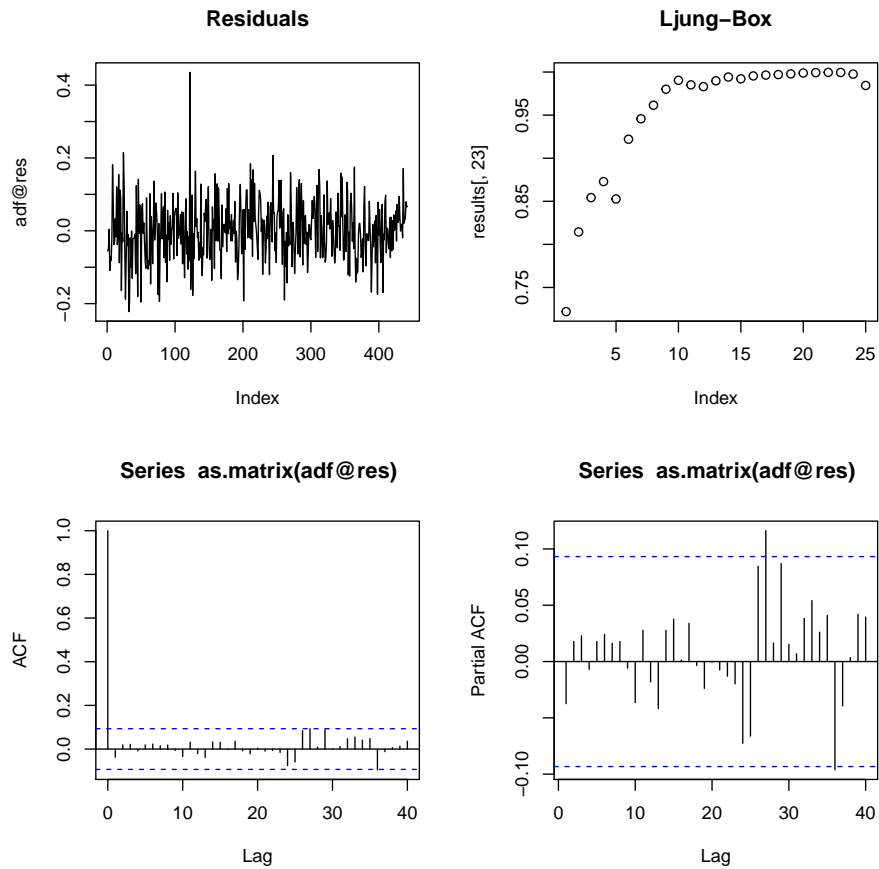


Nos gráficos acima podemos ver que a primeira ordem de lags que gera resíduos que são ruído branco é a ordem 23, então realizamos o teste com essa ordem e apresentamos os gráficos dos resíduos e dos testes nos resíduos.

```
adf <- ur.df(na.omit(data1$Diff), lags=23)
```

Figura 46: Resíduos

```
par(mfrow = c(2,2))
plot(adf@res, type='l', main='Residuals')
plot(results[,23], main='Ljung-Box')
acf(as.matrix(adf@res), lag.max=40)
pacf(as.matrix(adf@res), lag.max=40)
```



Tanto o gráfico dos resíduos quanto os gráficos das funções de autocorrelação e autocorrelação parcial indicam que os resíduos do teste se comportam como ruído branco. Os p-valores do teste de Ljung-Box não rejeitam a hipótese de independência dos resíduos. Agora, apresentamos o sumário do teste.

```
summary(adf)
##
```

```
## #####
## # 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
## -0.22188 -0.05360 -0.00105  0.05312  0.43487
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## z.lag.1      -1.42558     0.27895  -5.111 4.90e-07 ***
## z.diff.lag1    0.17388     0.27409   0.634 0.526170
## z.diff.lag2   -0.02840     0.26795  -0.106 0.915656
## z.diff.lag3   -0.13271     0.26169  -0.507 0.612327
## z.diff.lag4   -0.07700     0.25629  -0.300 0.763987
## z.diff.lag5   -0.02814     0.25109  -0.112 0.910805
## z.diff.lag6    0.07834     0.24538   0.319 0.749681
## z.diff.lag7    0.18963     0.23904   0.793 0.428057
## z.diff.lag8    0.20060     0.23263   0.862 0.389015
## z.diff.lag9    0.27072     0.22598   1.198 0.231590
## z.diff.lag10   0.40889     0.21970   1.861 0.063433 .
## z.diff.lag11   0.49504     0.21314   2.323 0.020681 *
## z.diff.lag12   0.37274     0.20730   1.798 0.072884 .
## z.diff.lag13   0.35967     0.20260   1.775 0.076576 .
## z.diff.lag14   0.39096     0.19667   1.988 0.047473 *
## z.diff.lag15   0.34434     0.18902   1.822 0.069218 .
## z.diff.lag16   0.36352     0.17944   2.026 0.043414 *
## z.diff.lag17   0.35626     0.16889   2.109 0.035497 *
## z.diff.lag18   0.42467     0.15598   2.723 0.006748 **
## z.diff.lag19   0.45444     0.13990   3.248 0.001254 **
## z.diff.lag20   0.37292     0.12147   3.070 0.002280 **
## z.diff.lag21   0.29879     0.09860   3.030 0.002593 **
## z.diff.lag22   0.24201     0.07265   3.331 0.000941 ***
## z.diff.lag23   0.25544     0.04474   5.709 2.16e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08436 on 418 degrees of freedom
## Multiple R-squared:  0.6836, Adjusted R-squared:  0.6654
```

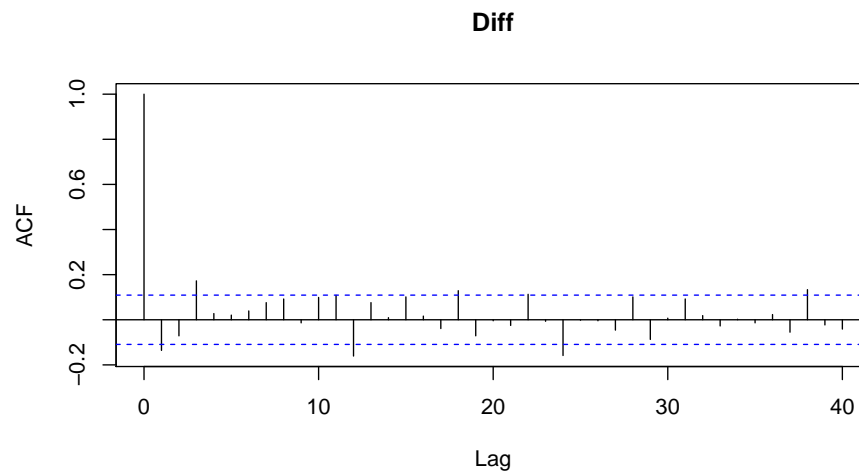
```
## F-statistic: 37.62 on 24 and 418 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -5.1105
##
## Critical values for test statistics:
##      1pct  5pct 10pct
## tau1 -2.58 -1.95 -1.62
```

O valor da estatística do teste é de -2,9787. Os valores críticos para o teste são de -2,58 (1%), -1,95 (5%) e -1,62 (10%). Sendo assim, o valor do teste ultrapassou os valores críticos para todos os graus de significância. O teste então rejeita a hipótese nula de presença de raiz unitária. O resultado obtido é, então, que a série temporal original é integrada de ordem um - I(1).

7.2 Identificação das Possíveis Formas Funcionais

Figura 47: FAC da Primeira Diferença

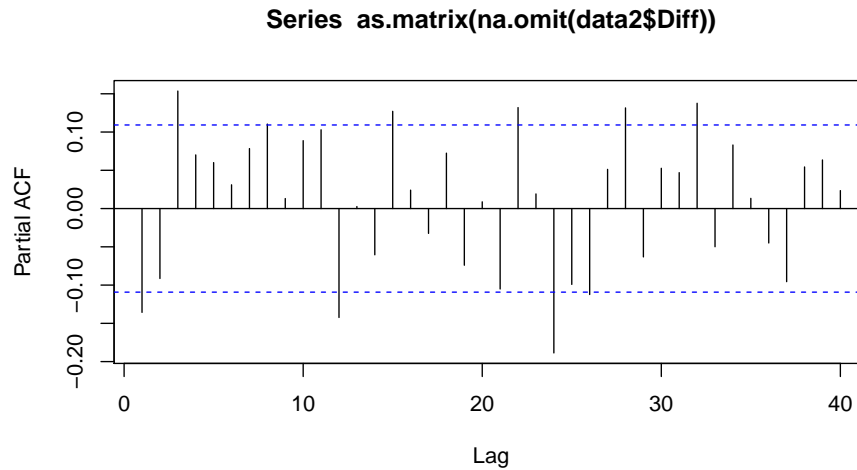
```
acf(as.matrix(na.omit(data2$Diff)), lag.max=40)
```



A função de autocorrelação tem até o terceiro lag significativo, enquanto mostra um lag significativo no fator sazonal.

Figura 48: FACP da Primeira Diferença

```
pacf(as.matrix(na.omit(data2$Diff)), lag.max=40)
```



A função de autocorrelação parcial tem até o terceiro lag significativo, enquanto mostra três lags significativos no fator sazonal.

A análise das funções acima sugere que uma ordem máxima para o modelo SARIMA é $(2,1,3)(1,0,2)_{12}$.

7.3 Estimação

Na subseção anterior definimos as ordens máximas do modelo SARIMA. Nesta seção vamos estimar todos os modelos até as ordens máximas, testar seus resíduos para checar se comportam-se como ruído branco e, finalmente, escolher o modelo que passa o teste dos resíduos que apresente melhor critério de informação.

Primeiramente, estimamos todos os modelos até as ordens máximas (72 modelos) e realizamos os testes de Ljung-Box.

[illegible]

```
results <- matrix(ncol=96,nrow=25)
for (i in 1:96){
  model <- sarima(data2$Value,
```

```

        ps[i], 1, qs[i],
        Ps[i], 0, Qs[i], 12)
for (e in 1:25){
  box <- Box.test(model$fit$residuals, lag=e)
  results[e,i] <- box$p.value
}
}

```

Em segundo lugar, escolhemos os modelos que passam no teste de Ljung-Box.

```

models <- c()
for (i in 1:96){
  if (min(results[,i]) > .05){
    models <- append(models, i)
  }
}
models

## [1] 1 2 5 13 14 17 25 26 29 37 38 41 49 50 53

```

Como podemos ver, apenas seis modelos apresentam resíduos com comportamento de ruído branco. Os estimamos novamente para avaliar o critério de informação.

```

aics <- c()
for (i in models){
  model <- sarima(data2$Value,
                  ps[i], 1, qs[i],
                  Ps[i], 0, Qs[i], 12)
  aics <- append(aics, model$AIC)
}

```

Abaixo, os critérios de informação de cada modelo.

```

models

## [1] 1 2 5 13 14 17 25 26 29 37 38 41 49 50 53

aics

## [1] -1.600023 -1.603858 -1.602997 -1.589686 -1.593681 -1.593786 -1.589401
## [8] -1.593450 -1.593655 -1.593969 -1.598147 -1.599064 -1.580736 -1.584343
## [15] -1.584862

```

Escolhemos, então, o modelo com menor critério de informação.


```

ind <- which.min(aics)
mod <- models[ind]
ps[mod]

## [1] 1

qs[mod]

## [1] 3

Ps[mod]

## [1] 1

Qs[mod]

## [1] 3

```

O modelo com melhor critério de informação é o segundo, que corresponde ao 4 na lista dos 72 modelos, que é o SARIMA (1,1,2)(0,0,2)₁₂. Estimamos então novamente esse modelo.

```

model2 <- sarima(data2$Value,
                 1, 1, 3,
                 1, 0, 3, 12)

```

Abaixo, o sumário do modelo.

```

print(model2)

## $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          ma1          ma2          ma3          sar1          sma1          sma2          sma3
##          0.9703      -1.1658      0.1775      0.0990      0.8984      -1.2372      0.0270      0.2941
## s.e.      0.0184      0.0625      0.0983      0.0643      0.0785      0.0945      0.0945      0.0653
##      constant
##          0.0040
## s.e.      0.0146
##
## sigma^2 estimated as 0.01084:  log likelihood = 268.22,  aic = -516.44
##

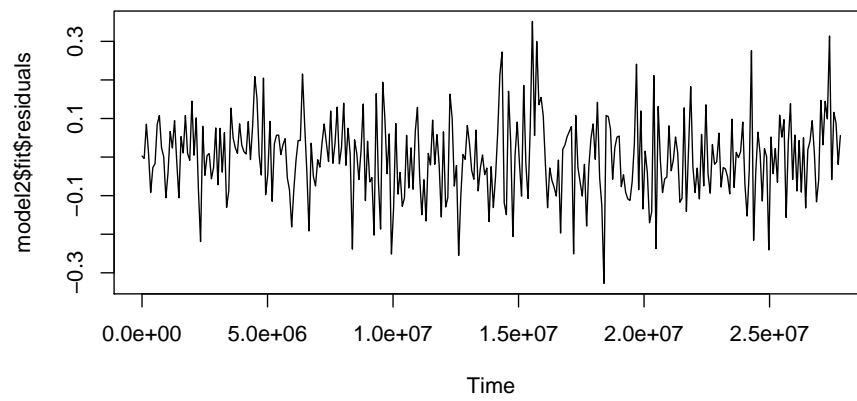
```

```
## $degrees_of_freedom
## [1] 313
##
## $ttable
##      Estimate      SE  t.value p.value
## ar1      0.9703 0.0184  52.8741  0.0000
## ma1     -1.1658 0.0625 -18.6441  0.0000
## ma2      0.1775 0.0983   1.8049  0.0721
## ma3      0.0990 0.0643   1.5397  0.1247
## sar1      0.8984 0.0785  11.4417  0.0000
## sma1     -1.2372 0.0945 -13.0876  0.0000
## sma2      0.0270 0.0945   0.2861  0.7750
## sma3      0.2941 0.0653   4.5076  0.0000
## constant  0.0040 0.0146   0.2724  0.7855
##
## $AIC
## [1] -1.603858
##
## $AICc
## [1] -1.602066
##
## $BIC
## [1] -1.486636
```

7.4 Diagnóstico dos Resíduos

Figura 49: Resíduos

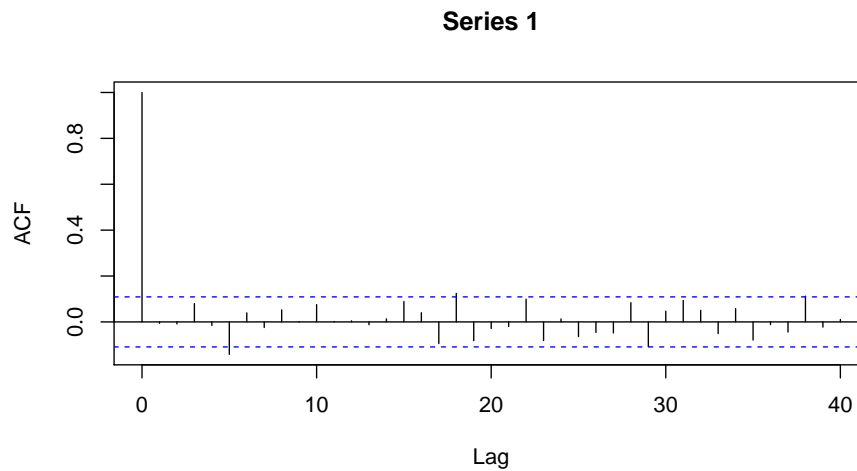
```
plot(model2$fit$residuals)
```



7.4.1 Independência

Figura 50: FAC dos Resíduos

```
acf(as.matrix(model2$fit$residuals), lag.max=40)
```

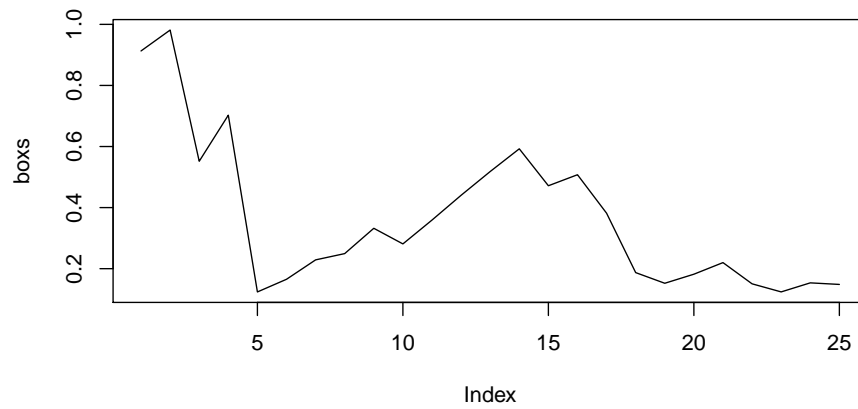


Tanto o gráfico dos resíduos quanto a sua função de autocorrelação indicam que os resíduos se comportam como ruído branco. Para testar essa hipótese, realizamos o teste de Ljung-Box para todos os lags até o vinte e cinco. Os testes são realizados no *chunk* abaixo, e na figura abaixo estão os p-valores dos testes para cada lag.

```
boxs <- matrix(nrow=25,ncol=1)
for (i in 1:25){
  box <- Box.test(model2$fit$residuals,lag=i)
  boxs[i] <- box$p.value
}
```

Figura 51: P-Valores de Ljung-Box

```
plot(boxs, type='l')
```



Os p-valores não rejeitam a hipótese nula de independência da distribuição.

7.4.2 Homoscedasticidade

Testamos a homoscedasticidade dos resíduos aplicando a análise acima no resíduo ao quadrado.

Figura 52: Resíduos ao Quadrado

```
plot(model2$fit$residuals^2)
```

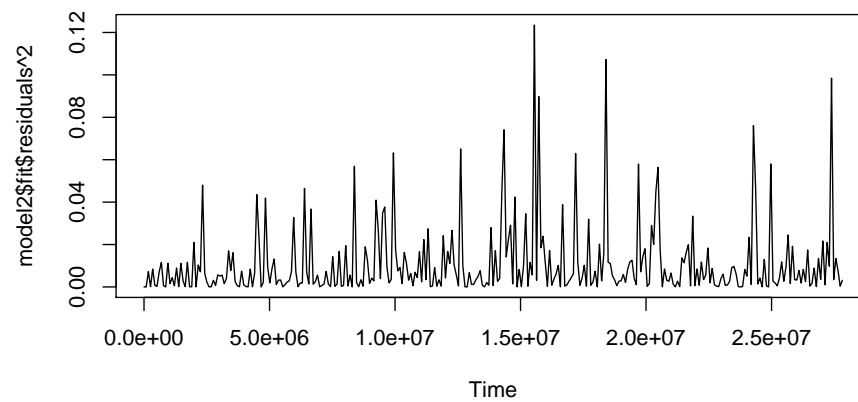


Figura 53: FAC dos Resíduos ao Quadrado

```
acf(as.matrix(model2$fit$residuals)^2, lag.max=40)
```

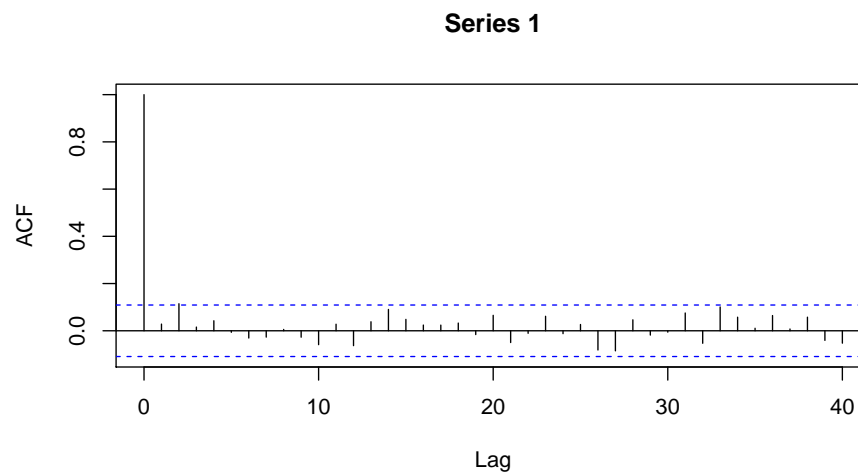
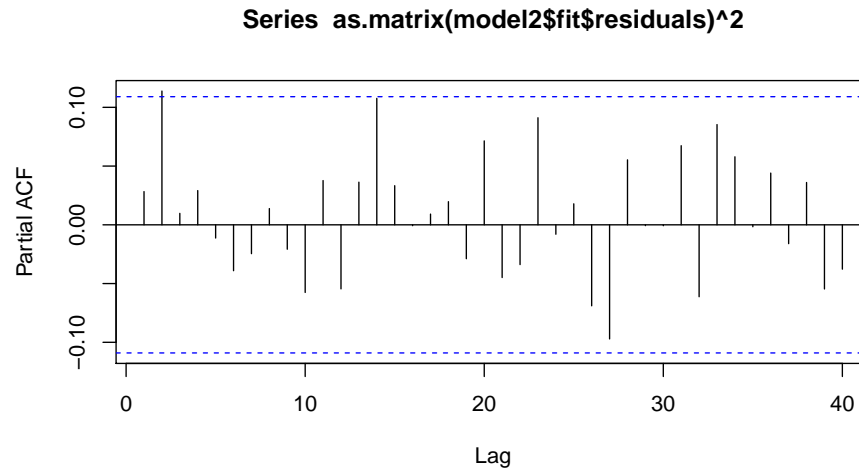


Figura 54: FACP dos Resíduos ao Quadrado

```
pacf(as.matrix(model2$fit$residuals)^2, lag.max=40)
```

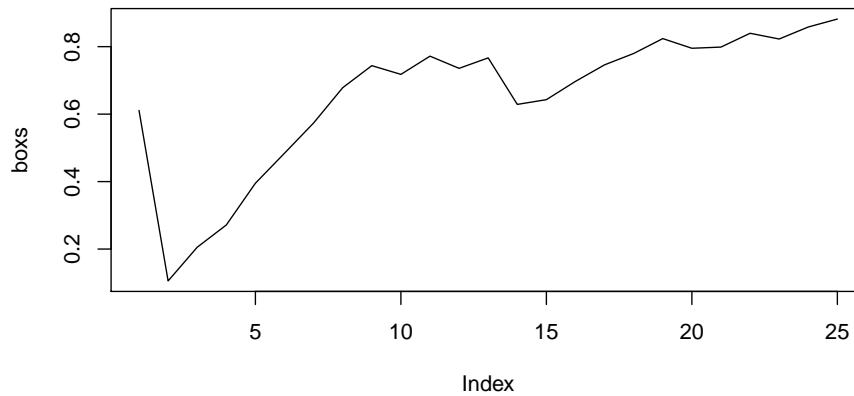


Tanto o gráfico dos resíduos ao quadrado quanto a sua função de autocorrelação e sua função de autocorrelação parcial indicam homoscedasticidade. Para testar essa hipótese, realizamos o teste de Ljung-Box nos resíduos ao quadrado para todos os lags até o vinte e cinco. Os testes são realizados no *chunk* abaixo, e na figura abaixo estão os p-valores dos testes para cada lag.

```
boxs <- matrix(nrow=25,ncol=1)
for (i in 1:25){
  box <- Box.test(model2$fit$residuals^2,lag=i)
  boxs[i] <- box$p.value
}
```

Figura 55: P-Valores de Ljung-Box

```
plot(boxs, type='l')
```



Os p-valores não rejeitam a hipótese nula de independência da distribuição. Os resíduos podem ser considerados então homoscedásticos, pois os resíduos ao quadrado são ‘bem comportados’ (ruído branco). Não é necessário então estimar um modelo de variância condicional.

7.4.3 Normalidade

Para testar a normalidade dos resíduos aplicamos o teste de Shapiro-Wilk (SHAPIRO e WILK 1965).

```
shapiro.test(model2$fit$residuals)

##
##  Shapiro-Wilk normality test
##
## data:  model2$fit$residuals
## W = 0.99433, p-value = 0.2742
```

O teste não rejeita a hipótese nula de normalidade dos resíduos.

7.5 Previsão e Acurácia

Nesta subseção faremos previsões e testaremos a acurácia das previsões feitas, como passo na avaliação do modelo estimado.

7.5.1 Previsão

Agora, realizaremos previsões para os últimos 20 períodos da amostra. Faremos previsão recursiva, ou seja: prevemos sempre apenas o período imediatamente subsequente, utilizando todos os dados até então.

```
fs <- c()
for (i in 20:1){
  until <- length(data2$Value)-i
  data <- data2$Value[1:until]
  prediction <- sarima.for(data, 1,
                           1, 1, 3,
                           1, 0, 3, 12)
  fs <- append(fs, prediction$pred)
}
```

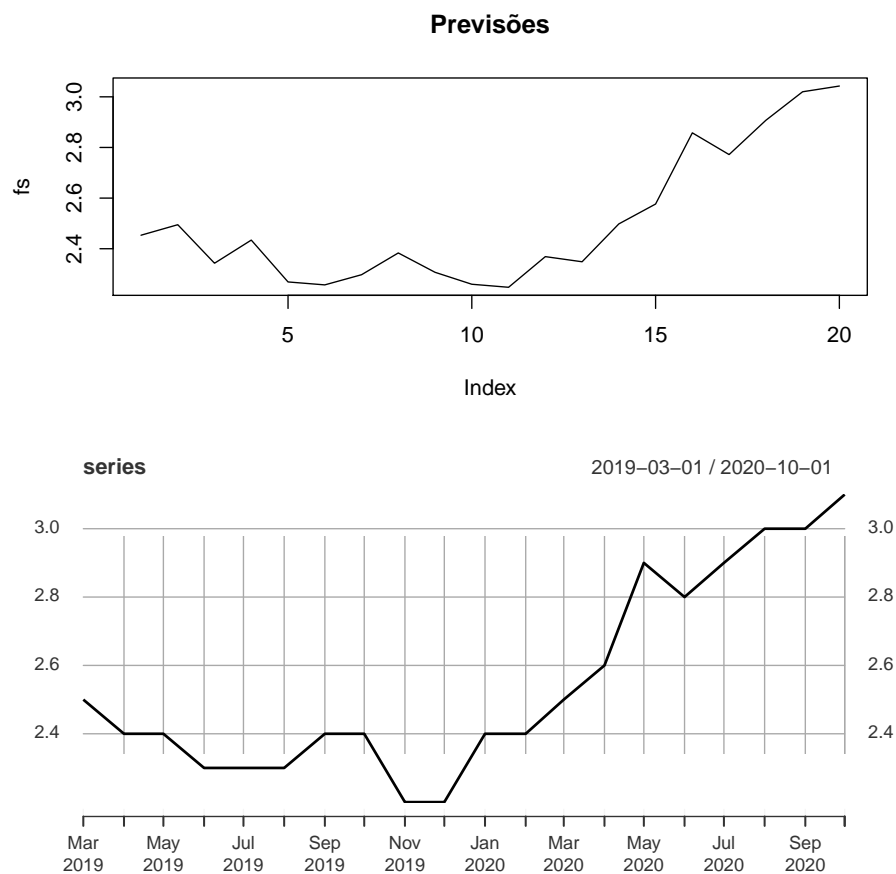
Abaixo, as previsões realizadas.

```
fs
## [1] 2.453337 2.494854 2.342738 2.434107 2.268806 2.257146 2.297302 2.383312
## [9] 2.306875 2.259689 2.247672 2.368786 2.348496 2.498029 2.576578 2.857734
## [17] 2.771846 2.907254 3.019997 3.042637
```

Abaixo, os gráficos das previsões e da série original.

Figura 56: Previsões e Série Original

```
length <- length(data2$Value)
start <- length - 19
series <- data2$Value[start:length]
par(mfrow = c(2,1))
plot(fs,type='l', main='Previsões')
plot(series)
```



7.5.2 Acurácia

Agora, vamos calcular a acurácia das previsões. Utilizaremos cinco medidas alternativas: erro médio (ME); erro quadrático médio (RMSE); erro médio absoluto (MAE); erro de porcentagem média (MPE); erro de porcentagem média absoluta (MAPE).

```
accuracy(fs,as.matrix(series))

##              ME      RMSE      MAE      MPE      MAPE
## Test set 0.04314027 0.1129687 0.09046585 1.525351 3.533608
```

As medidas acima podem ser úteis na escolha entre modelos, mas não nos dizem muito sobre o modelo se não temos outro para comparar. Para isso usamos o cálculo do Theil's U, que compara as previsões com o que seria uma 'adivinhação'. Caso o valor do cálculo for menor que 1, as previsões são melhores que adivinhação. Caso for maior, são piores. O teste é realizado abaixo.

```
TheilU(series,fs)

## [1] 0.04403311
```

O índice do teste foi de 0,04403311, menor que 1. O modelo fez previsões melhores que uma simples adivinhação.

8 Conclusão

Neste trabalho utilizamos as técnicas apresentadas na disciplina Estatística Econômica Aplicada para analisar uma série temporal. A série original possuía um outlier e uma quebra estrutural, então foram estimados modelos para duas séries temporais diferentes (duas partes da série temporal original). Ambas as séries temporais apresentaram sazonalidade, e não foram necessários modelos de variância condicional. As previsões dos modelos mostraram-se melhores que pura adivinhação, apontando para uma boa qualidade dos modelos.

Referências

- [And20] Signorell Andri et mult. al. *DescTools: Tools for Descriptive Statistics*. R package version 0.99.39. 2020. URL: <https://cran.r-project.org/package=DescTools>.
- [ANR20] Asael Alonzo Matamoros e Alicia Nieto-Reyes. *nortsTest: Assessing Normality of Stationary Process*. R package version 1.0.0. 2020. URL: <https://CRAN.R-project.org/package=nortsTest>.
- [Box+15] G.E.P. Box et al. *Time Series Analysis: Forecasting and Control*. Wiley Series in Probability and Statistics. Wiley, 2015. ISBN: 9781118674925. URL: <https://books.google.com.br/books?id=rNt5CgAAQBAJ>.

- [CH95] Fabio Canova e Bruce E. Hansen. “Are Seasonal Patterns Constant Over Time? A Test for Seasonal Stability”. Em: *Journal of Business & Economic Statistics* 13.3 (1995), pp. 237–252. DOI: 10.1080/07350015.1995.10524598. eprint: <https://www.tandfonline.com/doi/pdf/10.1080/07350015.1995.10524598>. URL: <https://www.tandfonline.com/doi/abs/10.1080/07350015.1995.10524598>.
- [CL93] Chung Chen e Lon-Mu Liu. “Joint Estimation of Model Parameters and Outlier Effects in Time Series”. Em: *JASA. Journal of the American Statistical Association* 88 (mar. de 1993). DOI: 10.2307/2290724.
- [DF79] David A. Dickey e Wayne A. Fuller. “Distribution of the Estimators for Autoregressive Time Series With a Unit Root”. Em: *Journal of the American Statistical Association* 74.366 (1979), pp. 427–431. ISSN: 01621459. URL: <http://www.jstor.org/stable/2286348>.
- [Gha20] Alexios Ghalanos. *rugarch: Univariate GARCH models*. R package version 1.4-4. 2020.
- [GW11] Garrett Grolmund e Hadley Wickham. “Dates and Times Made Easy with lubridate”. Em: *Journal of Statistical Software* 40.3 (2011), pp. 1–25. URL: <https://www.jstatsoft.org/v40/i03/>.
- [HK08] Rob J Hyndman e Yeasmin Khandakar. “Automatic time series forecasting: the forecast package for R”. Em: *Journal of Statistical Software* 26.3 (2008), pp. 1–22. URL: <https://www.jstatsoft.org/article/view/v027i03>.
- [Lac19] Javier López de Lacalle. *tsoutliers: Detection of Outliers in Time Series*. R package version 0.6-8. 2019. URL: <https://CRAN.R-project.org/package=tsoutliers>.
- [Lac20] Javier López de Lacalle. *uroot: Unit Root Tests for Seasonal Time Series*. R package version 2.1-2. 2020. URL: <https://CRAN.R-project.org/package=uroot>.
- [LB78] G. M. Ljung e G. E. P. Box. “On a Measure of Lack of Fit in Time Series Models”. Em: *Biometrika* 65.2 (1978), pp. 297–303. ISSN: 00063444. URL: <http://www.jstor.org/stable/2335207>.
- [Pfa08] B. Pfaff. *Analysis of Integrated and Cointegrated Time Series with R*. Second. ISBN 0-387-27960-1. New York: Springer, 2008. URL: <http://www.pfaffikus.de>.
- [R C20] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2020. URL: <https://www.R-project.org/>.
- [RU20] Jeffrey A. Ryan e Joshua M. Ulrich. *quantmod: Quantitative Financial Modelling Framework*. R package version 0.4.18. 2020. URL: <https://CRAN.R-project.org/package=quantmod>.

- [Sto20] David Stoffer. *astsa: Applied Statistical Time Series Analysis*. R package version 1.12. 2020. URL: <https://CRAN.R-project.org/package=astsa>.
- [SW65] S. S. SHAPIRO e M. B. WILK. “An analysis of variance test for normality (complete samples)†”. Em: *Biometrika* 52.3-4 (dez. de 1965), pp. 591–611. ISSN: 0006-3444. DOI: 10.1093/biomet/52.3-4.591. eprint: <https://academic.oup.com/biomet/article-pdf/52/3-4/591/962907/52-3-4-591.pdf>. URL: <https://doi.org/10.1093/biomet/52.3-4.591>.
- [TH20] Adrian Trapletti e Kurt Hornik. *tseries: Time Series Analysis and Computational Finance*. R package version 0.10-48. 2020. URL: <https://CRAN.R-project.org/package=tseries>.
- [Zei06] Achim Zeileis. “Implementing a Class of Structural Change Tests: An Econometric Computing Approach”. Em: *Computational Statistics & Data Analysis* 50 (2006), pp. 2987–3008.