Prova 2 - Estatística Econômica Aplicada

Bernardo Paulsen
Matheus Bragagnolo

Conteúdo

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 Schonerwald 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

2.1 Importação de Bibliotecas

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): função *accuracy*;
- *lubridate* (Grolemund e Wickham 2011): função *parse_date_time*;
- *notsTest* (Alonzo Matamoros e Nieto-Reyes 2020): função *Lm.test*;
- *quantmod* (Ryan e Ulrich 2020): função *xts*;
- *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*.

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

2.2 Definição de Funções

Abaixo definimos as funções que serão utilizadas ao longo do trabalho.

```r
all_box <- function(data){
  # Retorna os p-valores de testes de Ljung-Box até o lag 25
  boxs <- matrix(nrow=25,ncol=1)
  for (i in 1:25){
    box <- Box.test(data,lag=i)
    boxs[i] <- box$p.value
  }
```

```
  return(boxs)
}

select_adf <- function(data, typ){
  # Retorna o menor lag do teste ADF no qual os residuos se comportam como
  #ruido branco.
  results <- matrix(,nrow=25,ncol=24)
  for (i in 1:24){
    adf <- ur.df(data, type=typ, lags=i)
    results[,i] <- all_box(adf@res)
  }
  oks <- c()
  for (i in 1:24){
    if (min(results[,i]) > .05){
      oks <- append(oks, i)
    }
  }
  return(oks[1])
}

all_orders <- function(max_p, max_q, max_P, max_Q){
  # Retorna matriz com todas as ordens possiveis para o SARIMA dadas as ordens
  #maximas.
  ps = c()
  qs = c()
  Ps = c()
  Qs = c()
  for (p in 0:max_p){
    for (q in 0:max_q){
      for (P in 0:max_P){
        for (Q in 0:max_Q){
          ps <- append(ps,p)
          qs <- append(qs,q)
          Ps <- append(Ps,P)
          Qs <- append(Qs,Q)
        }
      }
    }
  }
  order_s <- matrix(c(ps,qs,Ps,Qs),ncol=4)
  return(order_s)
}

estimate <- function(data,d,D,order,f){
  # Estima um modelo SARIMA
```

```
  res    <- matrix(nrow=27)
  model  <- sarima(data,
                  order[1], d, order[2],
                  order[3], D, order[4], f)
  res[1] <- model$AIC
  res[2] <- model$BIC
  for (e in 1:25){
    box <- Box.test(model$fit$residuals,lag=e)
    res[e+2] <- box$p.value
  }
  return(res)
}

select_model <- function(data,max_p,d,max_q,max_P,D,max_Q,freq){
  # Seleciona o melhor modelo SARIMA seguindo a metodologia Box-Jenkins
  all_order <- all_orders(max_p,max_q,max_P,max_Q)
  len       <- length(all_order[,1])
  results   <- matrix(ncol=len,nrow=27)
  print('Modelos a estimar:')
  print(len)
  print('Estimando modelos, calculando AICs e fazendo testes de Ljung-Box...')
  for (i in 1:len){
    order <- all_order[i,]
    res   <- tryCatch(estimate(data,1,0,order,12),
                      error = function(e){
                        return(matrix(c(1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                      })
    results[,i] <- res
  }
  print('Selecionando os modelos que pelo teste de Ljung-Box...')
  models <- c()
  for (i in 1:len){
    if (min(results[3:27,i]) > .05){
      models <- append(models,i)
    }
  }
  print('Modelos selecionados:')
  print('p q P Q')
  for (m in models){
    print(all_order[m,])
  }
  print('Verificando menor AIC...')
  aics <- results[1,models]
  bics <- results[2,models]
  inda <- which.min(aics)
```

```r
  indb <- which.min(bics)
  moda <- models[inda]
  modb <- models[indb]
  print('Modelo selecionado por critério AIC:')
  print('p q P Q')
  print(all_order[moda,])
  print('Modelo selecionado por critério BIC:')
  print('p q P Q')
  print(all_order[modb,])
  both <- matrix(c(all_order[moda,],all_order[modb,]), nrow=4)
  return(both)
}

prediction <- function(data, p, q, P, Q){
  fs <- c()
  for (i in 10:1){
    start      <- 10 - i
    until      <- length(data)-i
    data       <- data1$Value[start:until]
    prediction <- sarima.for(data, 1,
                         p, 1, q,
                         P, 0, Q, 12)
    fs <- append(fs, prediction$pred)
  }
    return(fs)
}
```

2.3 Definição do Diretório de Trabalho

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

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

3 Dados

3.1 Importação dos Dados

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

```r
file_name    <- 'dataset.Rds'
sample_begin <- 5775
```

```
sample_end   <- 6564
dataset      <- readRDS(file_name)[sample_begin:sample_end,]
```

Agora, podemos analisar brevementos 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 Tramemento 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
```

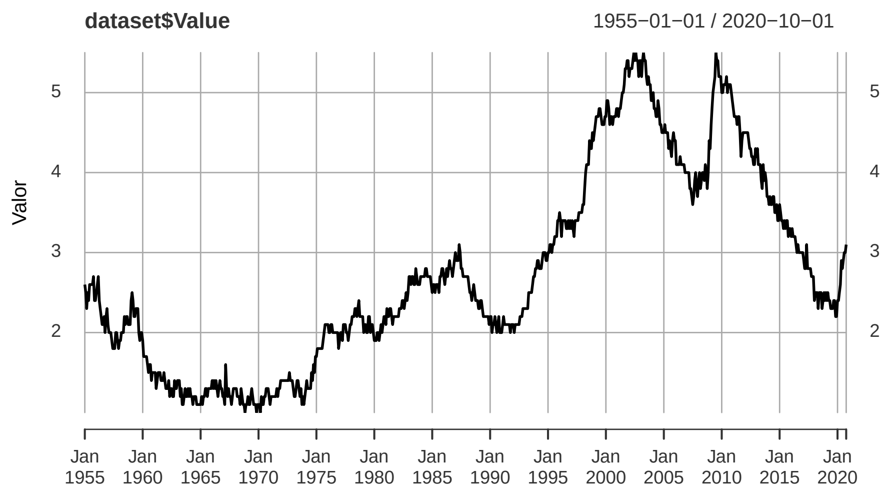
Agora, vimos que o formato das datas é *datetime*, e as datas passaram de uma coluna de dados para índice dos valores da série temporal.

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

```r
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 metodolodia de Chen e Liu 1993.

```r
ol <- tso(ts(dataset))
ol

## Series: ts(dataset)
## Regression with ARIMA(0,1,1) errors
##
## Coefficients:
##           ma1   AO147
##       -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   AO 147  147  0.3986 5.032
```

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

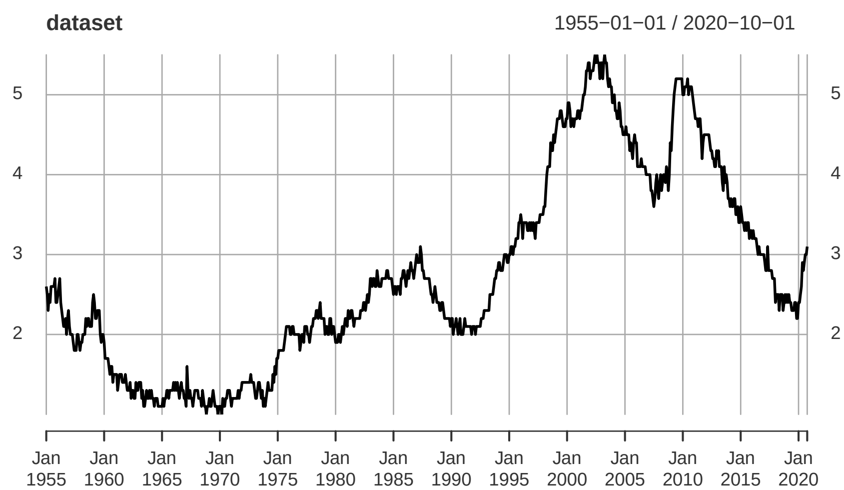
```
dataset <- tsclean(dataset)
```

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.

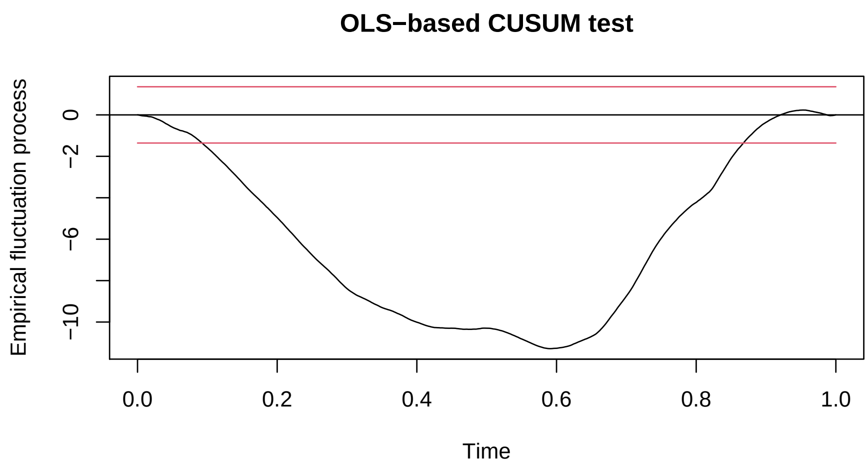
5.1 Processo de Flutuação Empírica

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 provavél 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)
point

## [1] 468
```

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 Parte da Série Temporal

6.1 Definição da Ordem de Integração

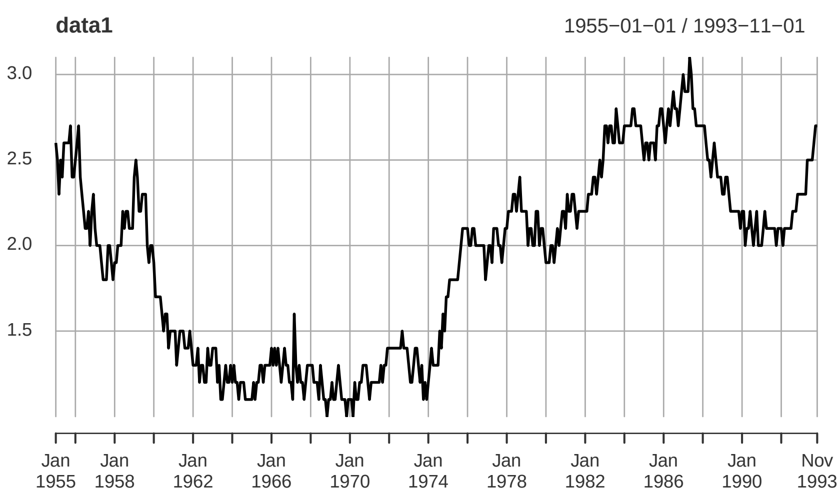
O primeiro passo na metodologia Box-Jenkins (Box et al. 2015) é a definição da ordem de integração da série temporal .

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)
```



O gráfico assemelha-se a um passeio aleatório, portanto, indicandio a presença de raíz unitária. Para coletar mais indícios analisamos abaixo a função de autocorrelação e a função de autocorrelação parcial.

Figura 5: FAC da Série Temporal

```
acf(as.matrix(data1), lag.max=60)
```

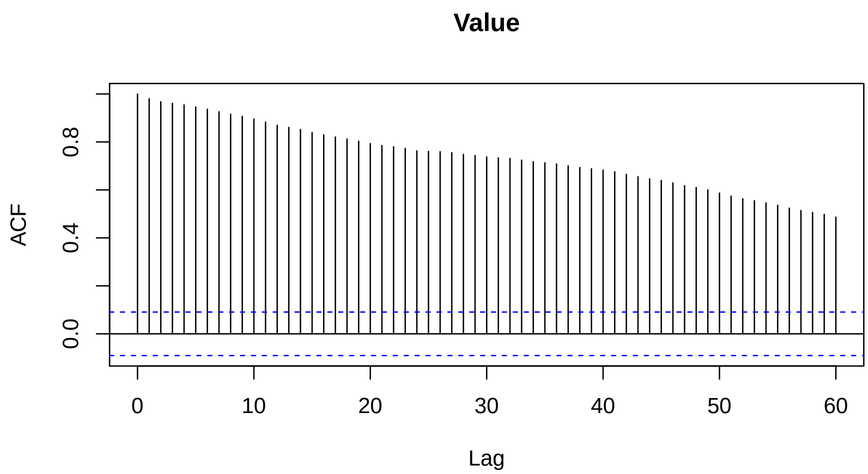
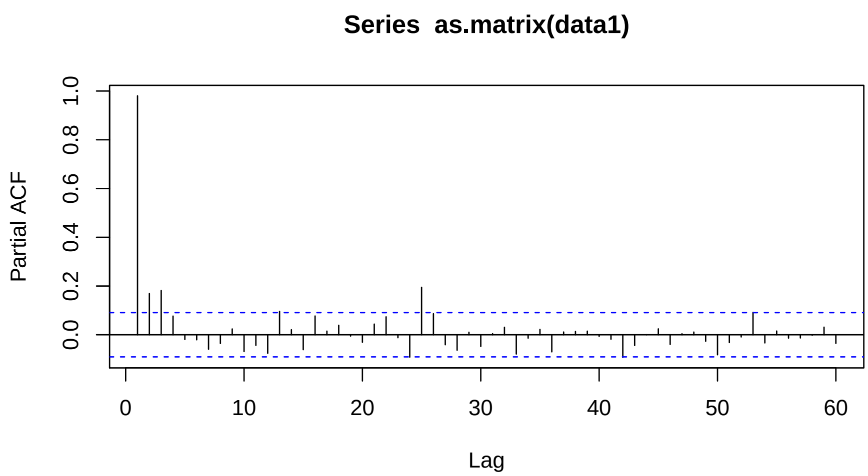


Figura 6: FACP da Série Temporal

```
pacf(as.matrix(data1), lag.max=60)
```



A função de autocorrelação aparentemente possui decaimento lento, indicando possibilidade de raiz unitária. Para testar a hipótese de presença de raiz unitária cutilizamos o teste Dickey-Fuller aumentado (Dickey e Fuller 1979). O teste será realizado sem *drift* ou tendência, pois a visualização do gráfico da série temporal indica a não presença de tais. 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 esse passos para encontrar o menor lag que retorna resíduos que sejam ruído branco.

```
lag <- select_adf(data1$Value,"none")
lag

## [1] 24
```

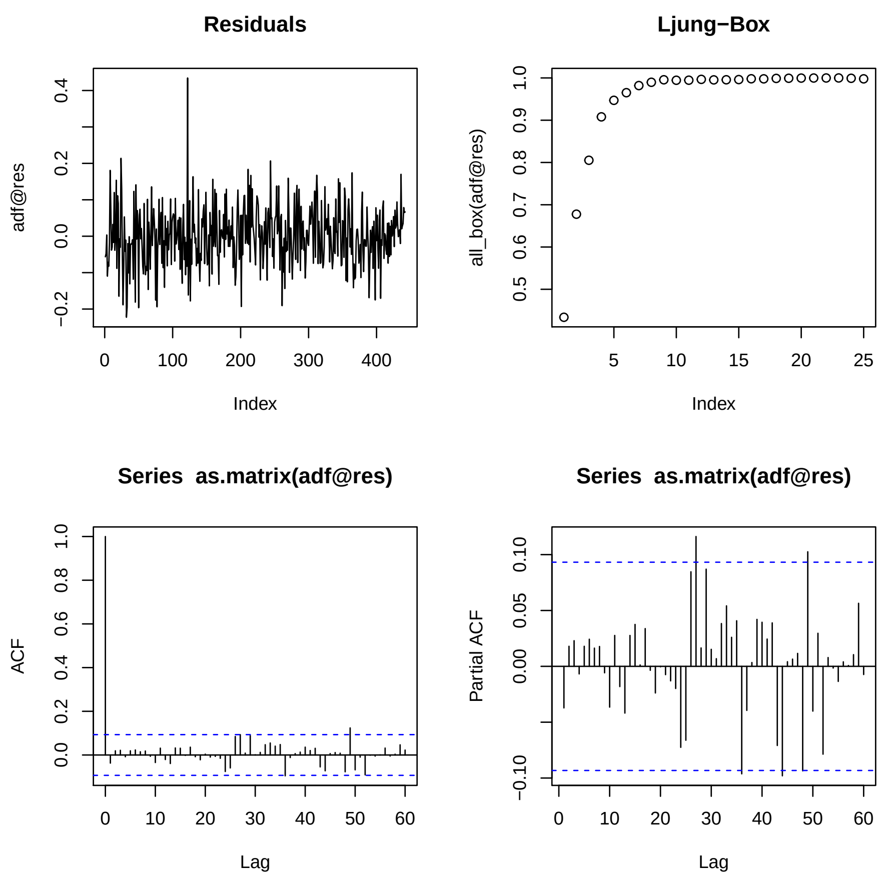
Os resultados dos passos anteriores indicam que o menor lag para o teste é o 24, então realizamos o teste com essa ordem.

```
adf <- ur.df(data1$Value, type="none", lags=lag)
```

Abaixo analisamos os resíduos do teste.

Figura 7: Resíduos

```
par(mfrow = c(2,2))
plot(adf@res, type='l', main='Residuals')
plot(all_box(adf@res), main='Ljung-Box')
acf(as.matrix(adf@res), lag.max=60)
pacf(as.matrix(adf@res), lag.max=60)
```



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 as estatísticas do teste e os valores críticos

```
adf@teststat

##                tau1
## statistic 0.1471326

adf@cval

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

O valor da estatística do teste é de 0,1471. Os valores críticos para o teste são de -2,58 (1%), -1,95 (5%) e -1,62 (10%). Sendo assim, o valor do teste não ultrapassou os valores críticos para nenhum grau de significância. O teste indica que não há evidências suficientes para rejeitarmos a hipótese nula de presença de raíz 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.

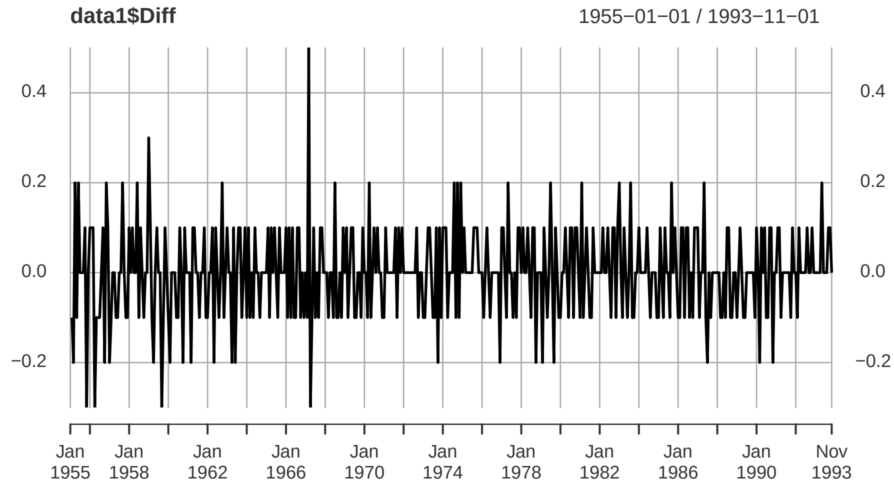
```
summary(data1)

##      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 8: Primeira Diferença

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



A analise visual do gráfico sugere variação em torno de uma média sem distanciamento grande por longos períodos, portanto, indica que a estacionariedade da série foi obtida na primeira diferenciação.

Figura 9: FAC da Primeira Diferença

```r
acf(as.matrix(na.omit(data1$Diff)), lag.max=60)
```

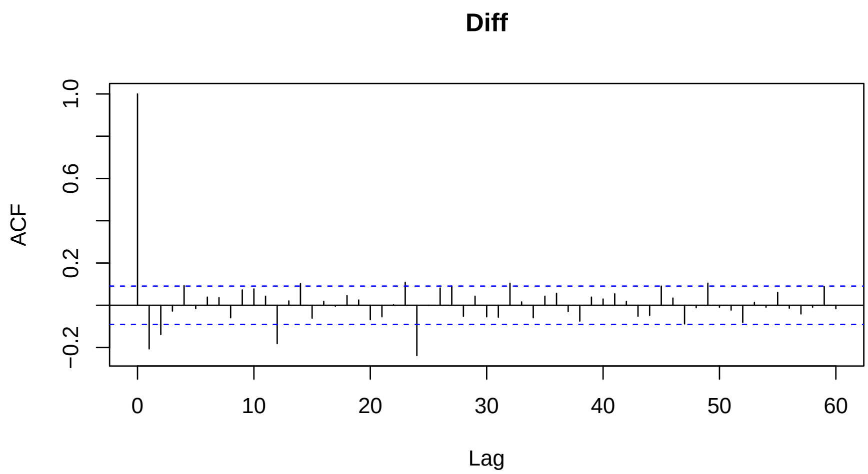
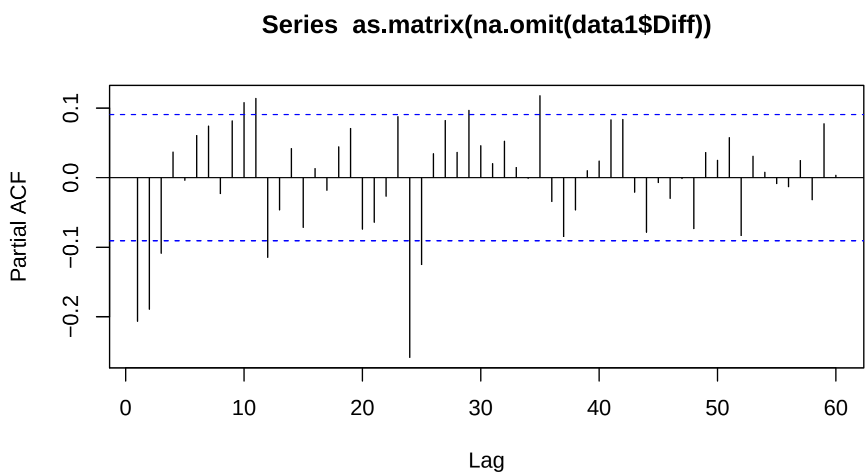


Figura 10: FACP da Primeira Diferença

```r
pacf(as.matrix(na.omit(data1$Diff)), lag.max=60)
```



A análise dos gráficos pós diferenciação indica rápido decaimento nas funções

ACF e PACF, reforçando a hipótese de que não há raíz unitária. Agora testaremos, com o teste de Dickey-Fuller aumentado, a presença de raiz unitária. Não adicionaremos drift ou tendância no teste pois não há indício pelo gráfico da série de existência de algum desses parâmetros. Seguiremos os passos descritos acima, quando aplicamos o teste na série temporal original.

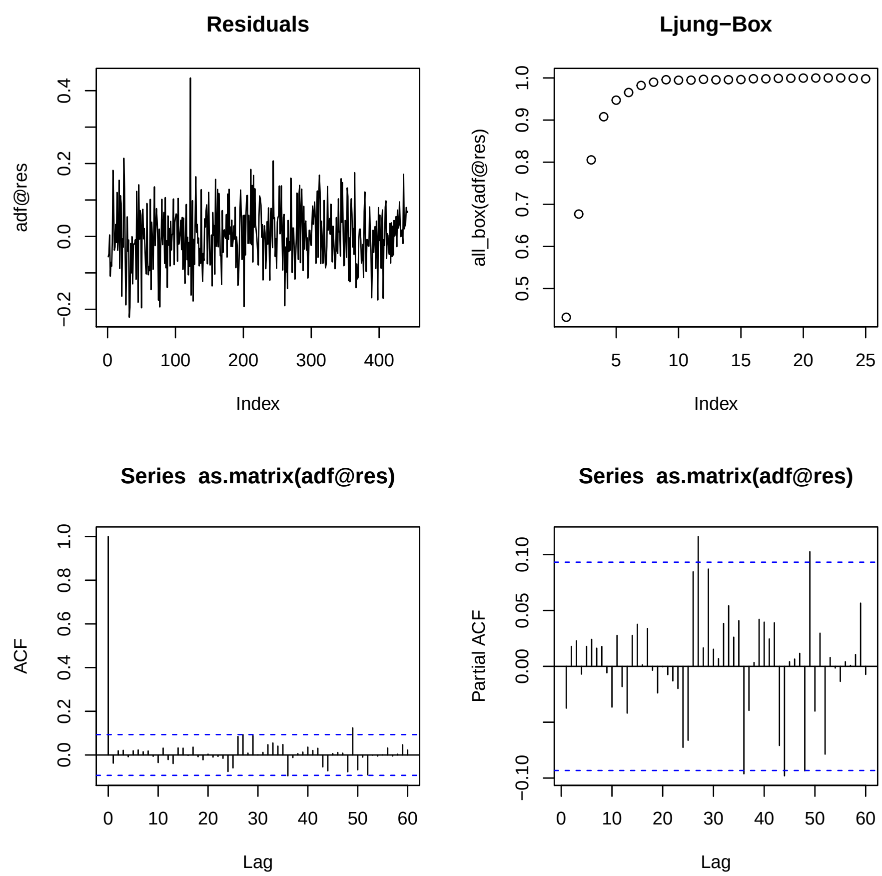
```
lag <- select_adf(na.omit(data1$Diff),"none")
lag

## [1] 23
```

Os resultados dos passos anteriores indicam que o menor lag para o teste é o 23, então realizamos o teste com essa ordem.

```
adf <- ur.df(na.omit(data1$Diff),"none",lags=lag)
```

Figura 11: Resíduos

```
par(mfrow = c(2,2))
plot(adf@res, type='l', main='Residuals')
plot(all_box(adf@res), main='Ljung-Box')
acf(as.matrix(adf@res), lag.max=60)
pacf(as.matrix(adf@res), lag.max=60)
```



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 as estatísticas do teste e os valores críticos

```r
adf@teststat
```

```
##              tau1
## statistic -5.1105
```

```r
adf@cval
```

```
##       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 aponta que não há evidências suficientes de presença de raíz unitária. Portanto, consideramos a série estacionária após a primeira diferenciação. O resultado obtido é, então, que a série temporal original é integrada de ordem 1 - I(1).

6.1.4 Diferença Sazonal

A função de autocorrelação indica que existem lags sazonais significativos, no entando, o decaimento é rápido, indicanco não existência de raiz unitária sazonal.Testamos a seguir a hipótese nula de não presença de raiz unitária sazonal com o teste de Canova e Hansen (Canova e Hansen 1995).

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

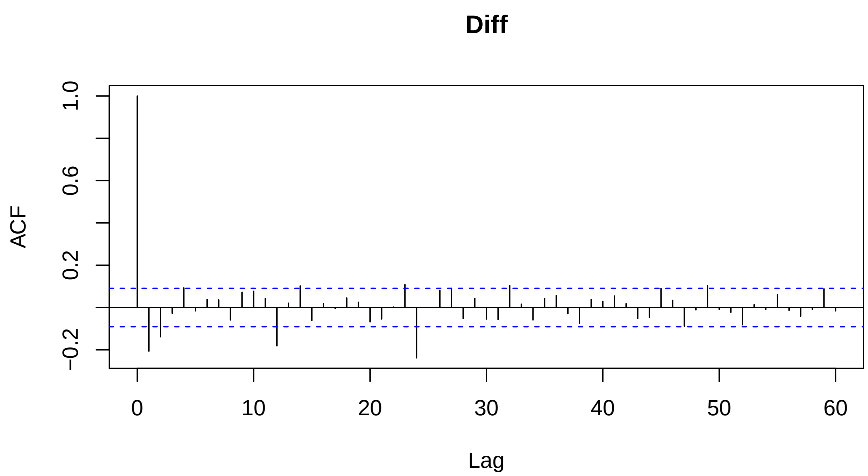
```
## [1] 0.5464967
```

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.

6.2 Identificação das Possíveis Formas Funcionais

Figura 12: FAC da Primeira Diferença

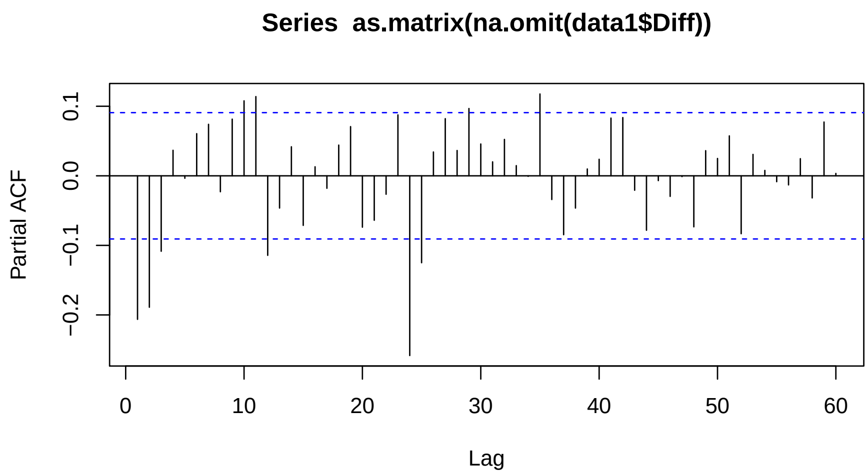
```
acf(as.matrix(na.omit(data1$Diff)), lag.max=60)
```



A função de autocorrelação trunca após o segundo lag. Como mencionado anteriormente, a FAC da série da primeira diferença mostra que há decaimento rápido do padrão de sazonalidade (múltiplo de 12) a partir do lag número 24.

Figura 13: FACP da Primeira Diferença

```
pacf(as.matrix(na.omit(data1$Diff)), lag.max=60)
```



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 (3,1,2)(3,0,2)12.

6.3 Estimação

Na subseção anterior definimos as ordems 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 os modelos que passam o teste dos resíduos que apresente melhores critério de informação (AIC e BIC). Esses passos são realizados no *chunk* abaixo.

```
order <- select_model(data1$Value,3,1,2,3,0,2,12)
```

Abaixo, as ordens dos melhores modelos pelos critérios AIC e BIC.

```
# Ordem pelo critério AIC
print("p q P Q")

## [1] "p q P Q"

order[,1]

## [1] 2 2 1 2
```

```r
# Ordem pelo critério BIC
print("p q P Q")

## [1] "p q P Q"

order[,2]

## [1] 2 2 0 2
```

O modelo com melhor critério AIC é o SARIMA (2,1,2)(1,0,2)12, e o com melhor critério BIC é o SARIMA(2,1,2)(0,0,2)12. Pelo princípio da parcimônia, optamos pelo modelo mais simples (menos parâmetros), que apresenta menor critério BIC.

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

6.4 Diagnóstico dos Resíduos

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

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 14: Resíduos

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

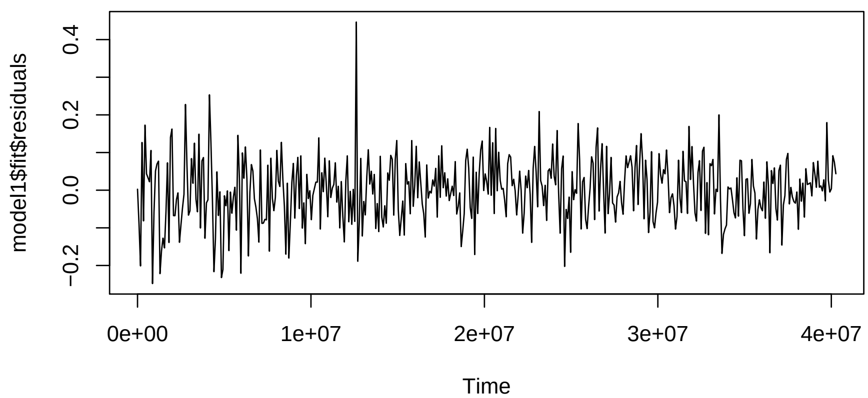


Figura 15: FAC dos Resíduos

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

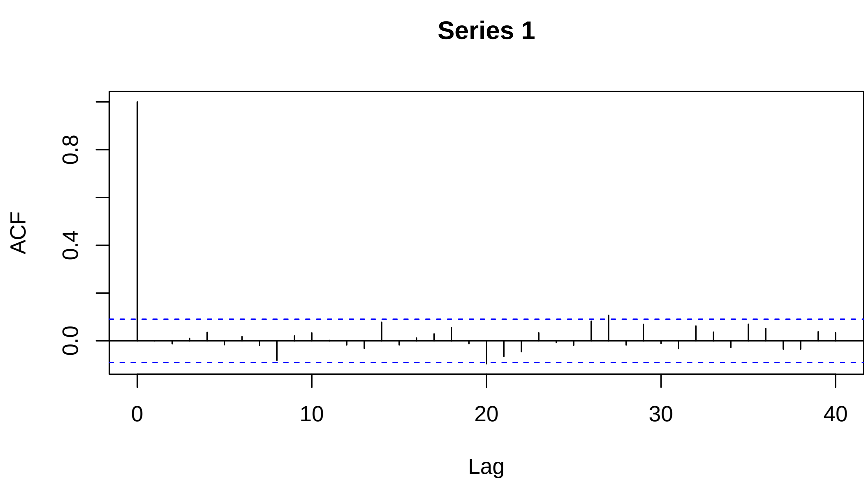
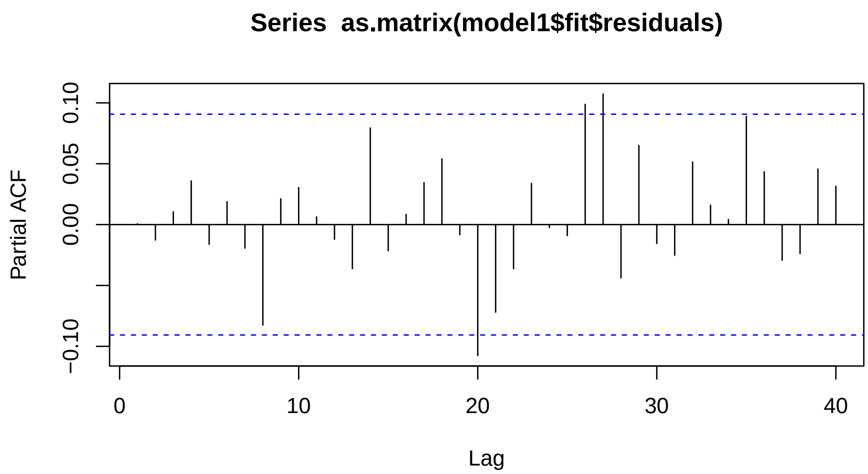


Figura 16: FACP dos Resíduos

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

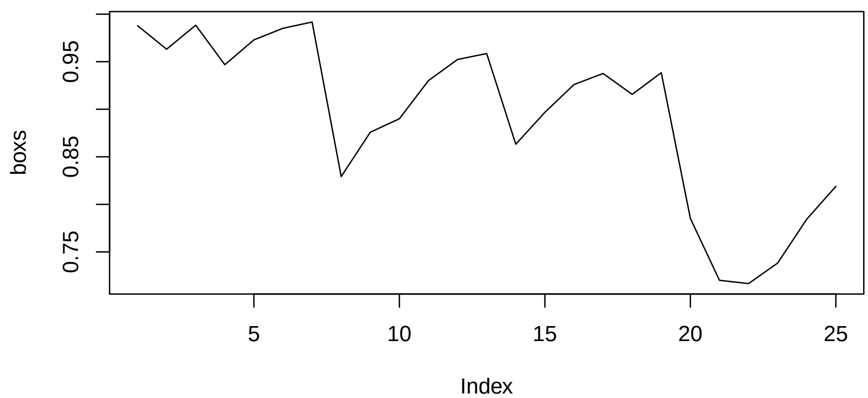


Tanto o gráfico dos resíduos quanto a sua função de autocorrelação e sua função de autocorrelação parcial indicam que os resíduos se comportam como ruído branco, já que a quantidade de lags significativos é a esperada para um nível de significância de 5%. o Para testar essa hipótese, realizamos o teste 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.

```r
boxs <- all_box(model1$fit$residuals)
```

Figura 17: P-Valores de Ljung-Box

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



Abaixo, o valor mínimo do teste de Ljung-Box entre todos os lags considerados.

```
min(boxs)

## [1] 0.7166852
```

Os p-valores não rejeitam a hipótese nula de independência da distribuição, uma vez que todos são superiores a 0,05.

6.4.2 Homoscedasticidade

Testamos a homoscedasticidade dos resíduos do modelo selecionado apicando a análise acima nos resíduos ao quadrado.

Figura 18: Resíduos ao Quadrado

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

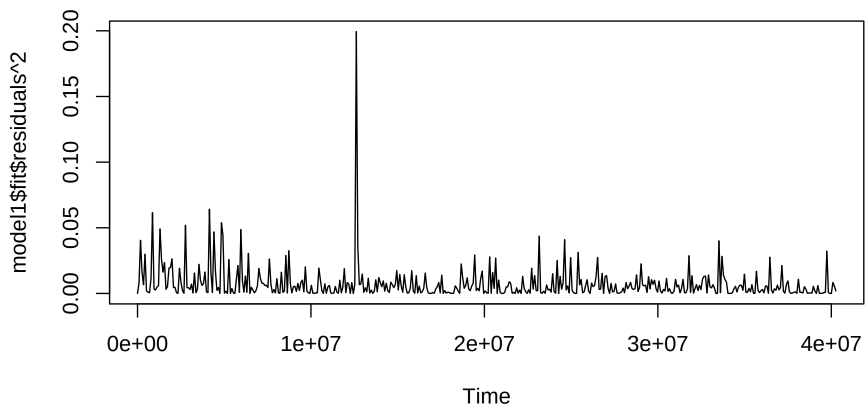


Figura 19: FAC dos Resíduos ao Quadrado

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

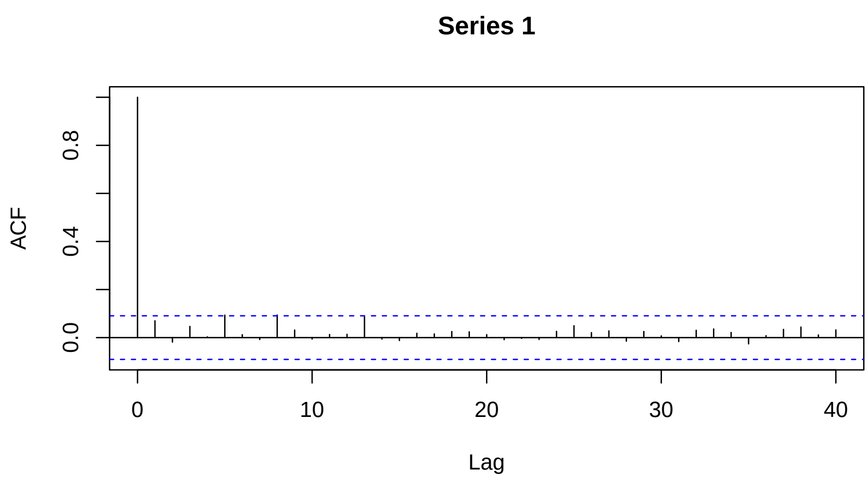
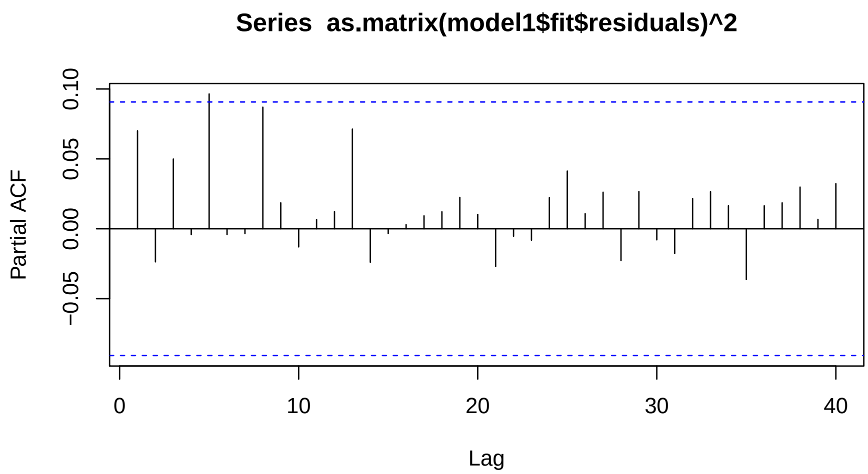


Figura 20: FACP dos Resíduos ao Quadrado

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

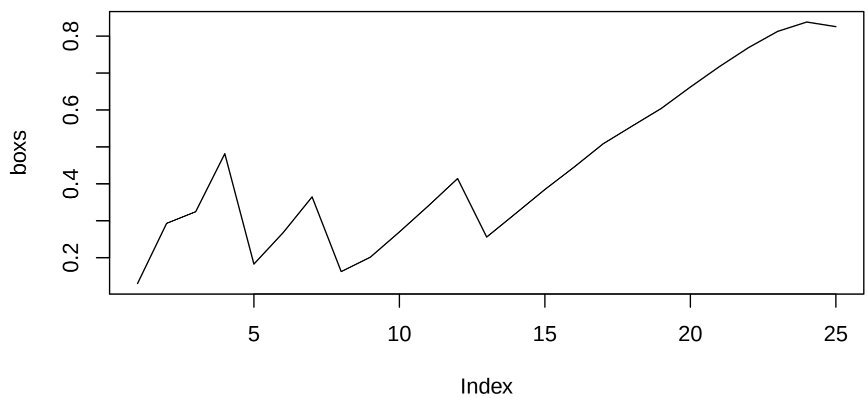


Tanto o gráfico dos resíduos ao quadrado quanto a sua função de autocorrelação e sua função de autocorrelação parcial indicam homoscedasticidade. Para testar essa hipótese, realizamos o teste de Ljung-Box nos resíduos ao quadrado para todos os lags até o vinte e cinco. Os testes são realizados no *chunk*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

```
boxs <- all_box(model1$fit$residuals^2)
```

Figura 21: P-Valores de Ljung-Box

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



Abaixo, o valor mínimo do teste de Ljung-Box entre todos os lags considerados.

```
min(boxs)

## [1] 0.1300694
```

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.

6.5 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 10 períodos da amostra. Faremos previsão *rolling window*, ou seja: prevemos sempre apenas o período imediatamente subsequente, utilizando a mesma quantidade de dados para todas as previsões.

```r
fs <- prediction(data1$Value,2,2,0,2)
```

Abaixo, as previsões realizadas.

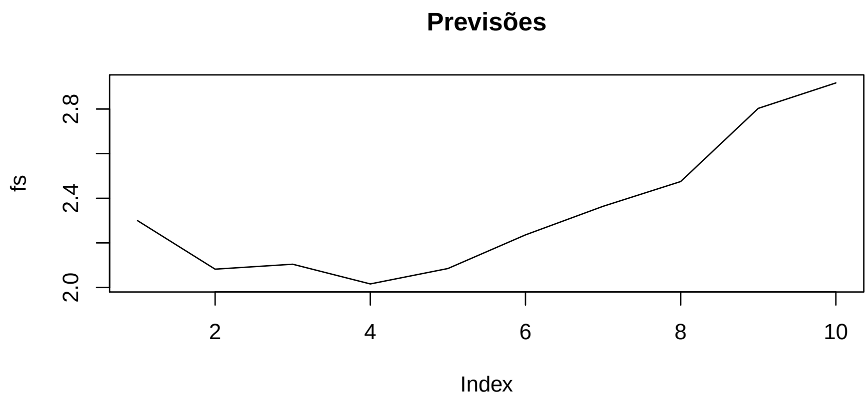
```r
fs
```

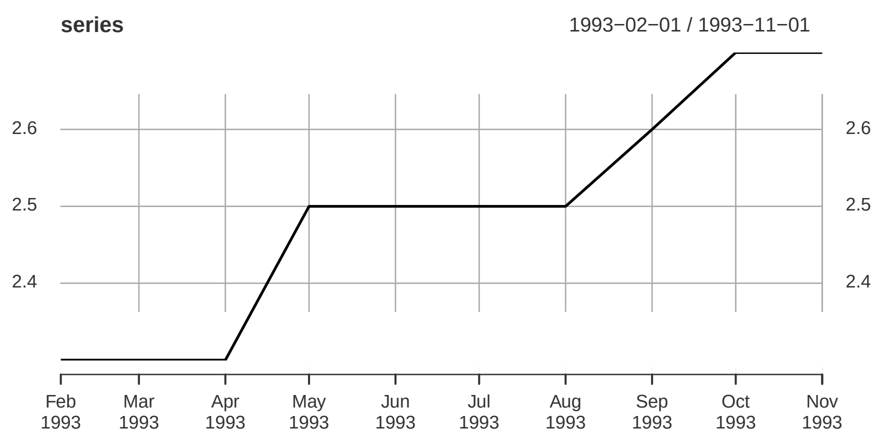
```
##  [1] 2.299853 2.082237 2.104396 2.016007 2.085035 2.235921 2.363979 2.474957
##  [9] 2.802898 2.917021
```

Abaixo, os gráficos das previsões e da série original.

Figura 22: Previsões e Série Original

```
length <- length(data1$Value)
start  <- length - 9
series <- data1$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).

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

##                 ME      RMSE       MAE      MPE     MAPE
## Test set 0.1517694 0.2556732 0.2157534 6.290162 8.659937
```

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.

```r
TheilU(series,fs)

## [1] 0.1025074
```

O índice do teste foi de 0,102, menor que 1. O modelo fez previsões melhores que uma simples adivinhação.

7 Segunda Parte da Série Temporal

7.1 Definição da Ordem de Integração

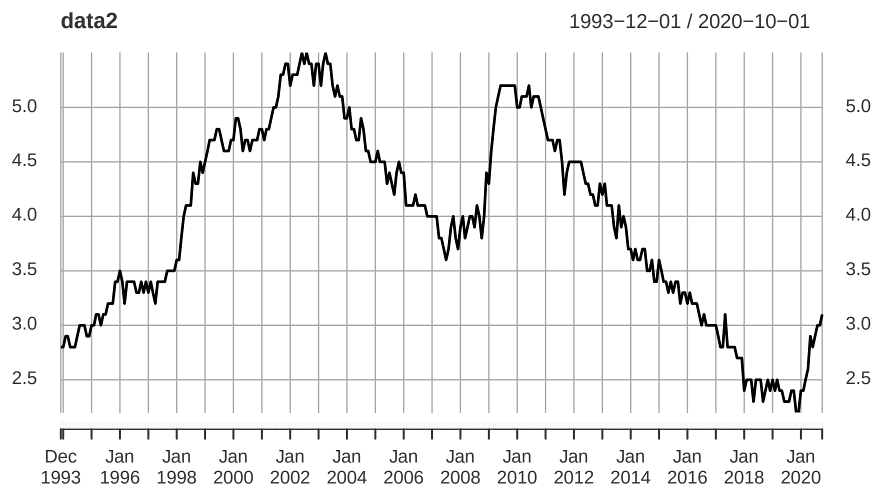
O primeiro passo na metodologia Box-Jenkins (Box et al. 2015) é a definição da ordem de integração da série temporal .

7.1.1 Análise da Série Original

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

Figura 23: Série Temporal

```r
plot(data2)
```



O gráfico assemelha-se a um passeio aleatório, portanto, indicandio a presença de raíz unitária. Para coletar mais indícios visuais analisamos abaixo a função de autocorrelação e a fução de autocorrelação parcial.

Figura 24: FAC da Série Temporal

```
acf(as.matrix(data2), lag.max=60)
```

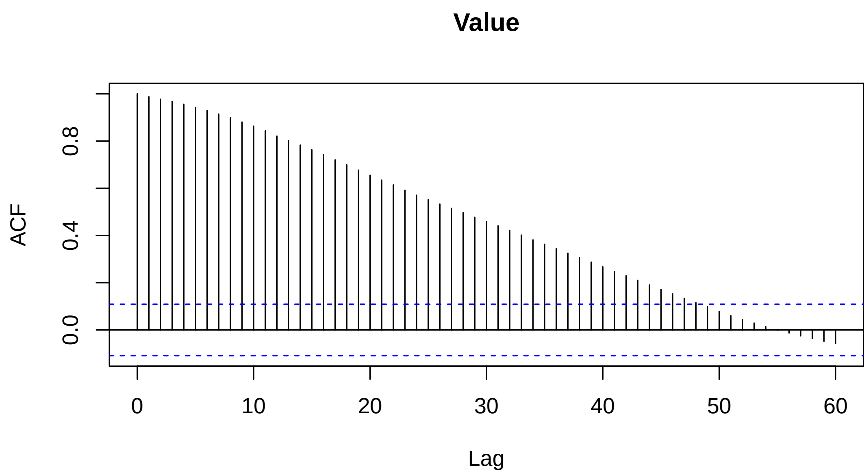
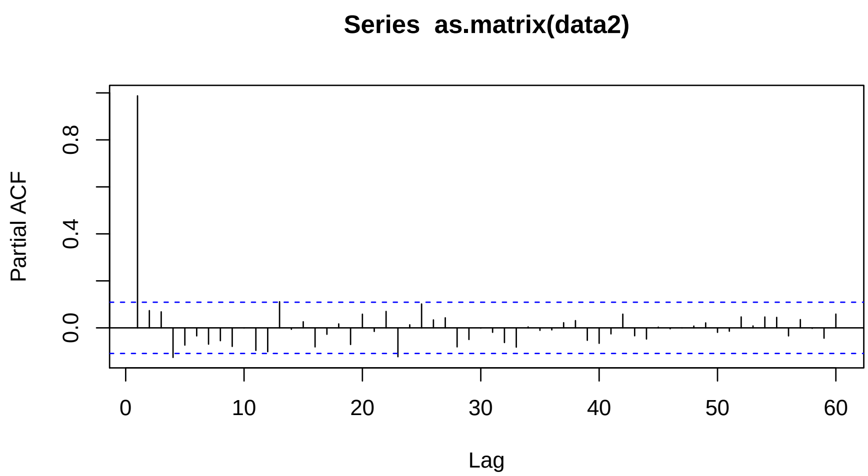


Figura 25: FACP da Série Temporal

```
pacf(as.matrix(data2), lag.max=60)
```



A função de autocorrelação aparentemente possui decaimento lento, indicando 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 sem *drift* ou tendência, pois a visualização do gráfico da série temporal não indica a presença de tais. 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.

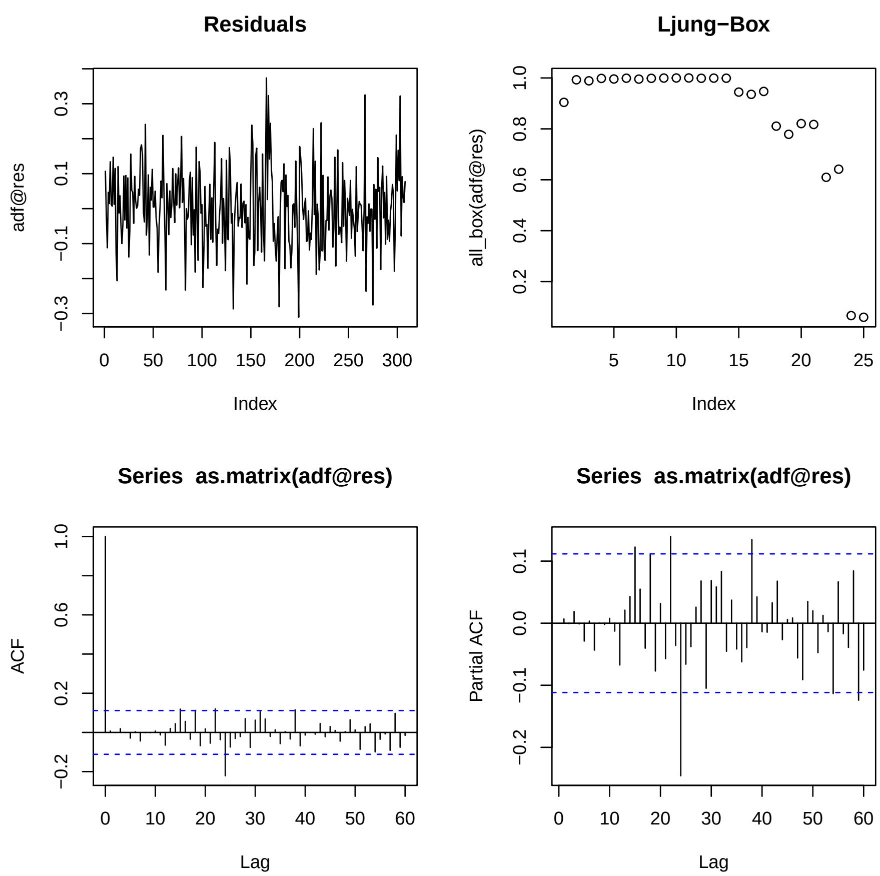
```
lag <- select_adf(data2$Value, "none")
lag

## [1] 14
```

Escolhemos a ordem de lag 14.

```
adf <- ur.df(na.omit(data2$Value), "none", lags=lag)
```

Figura 26: Resíduos

```
par(mfrow = c(2,2))
plot(adf@res, type='l', main='Residuals')
plot(all_box(adf@res), main='Ljung-Box')
acf(as.matrix(adf@res), lag.max=60)
pacf(as.matrix(adf@res), lag.max=60)
```



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 as estatísticas do teste e os valores críticos

```
adf@teststat

##                 tau1
## statistic -0.2342756

adf@cval

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

O valor da estatística do teste é de -0.2343. Os valores críticos para o teste são de -2,58 (1%), -1,95 (5%) e -1,62 (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.

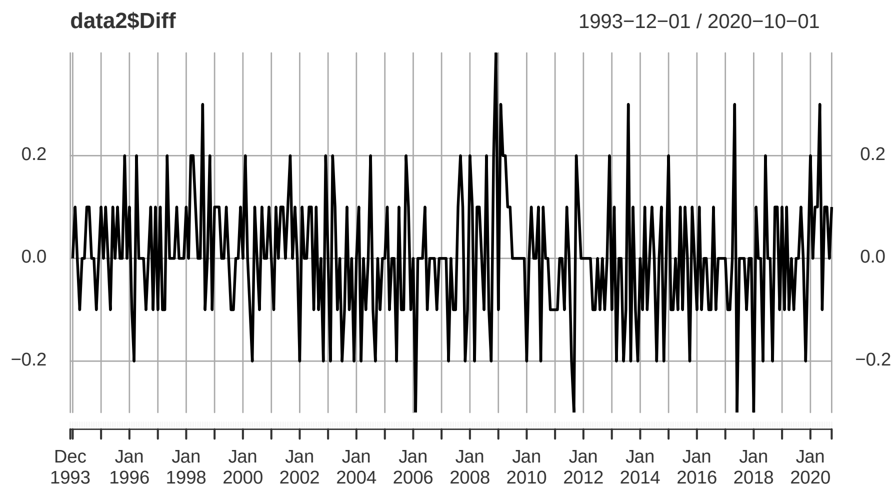
```
summary(data2)

##      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 27: Primeira Diferença

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



A analise visual do gráfico sugere variação em torno de uma média sem distanciamento grande por longos períodos, portanto, indica que a estacionariedade da série foi obtida na primeira diferenciação.

Figura 28: FAC da Primeira Diferença

```
acf(as.matrix(na.omit(data2$Diff)), lag.max=60)
```

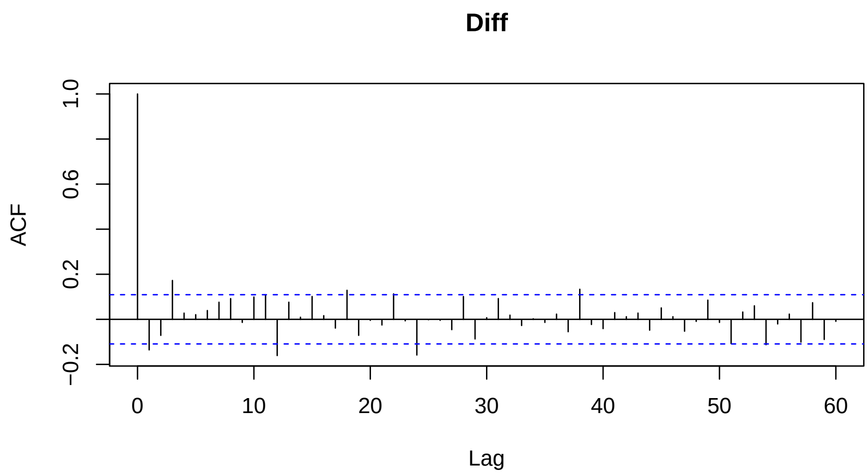
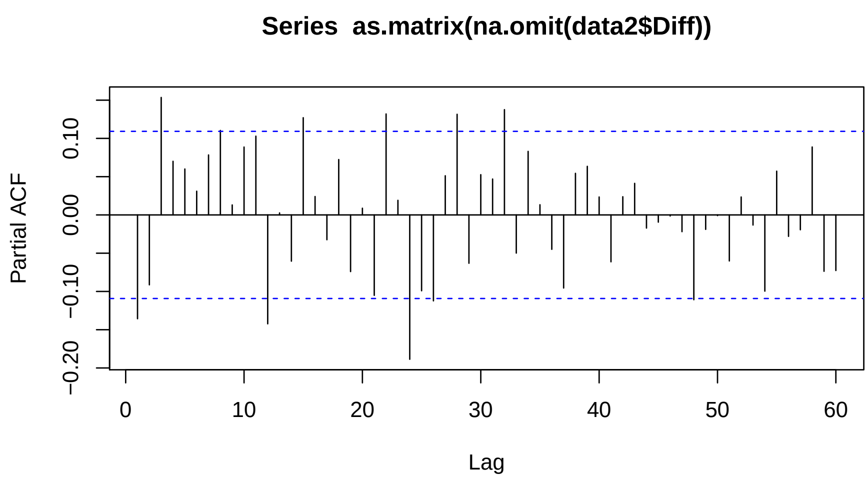


Figura 29: FACP da Primeira Diferença

```
pacf(as.matrix(na.omit(data2$Diff)), lag.max=60)
```



A análise dos gráficos pós diferenciação indica rápido decaimento nas funções

ACF e PACF, reforçando a hipótese de que não há raíz unitária. Agora testaremos, com o teste de Dickey-Fuller aumentado, a presença de raiz unitária. Não adicionaremos drift ou tendância no teste pois não há indício pelo gráfico da série de existência de algum desses parâmetros. Seguiremos os passos descritos acima, quando aplicamos o teste na série temporal original.

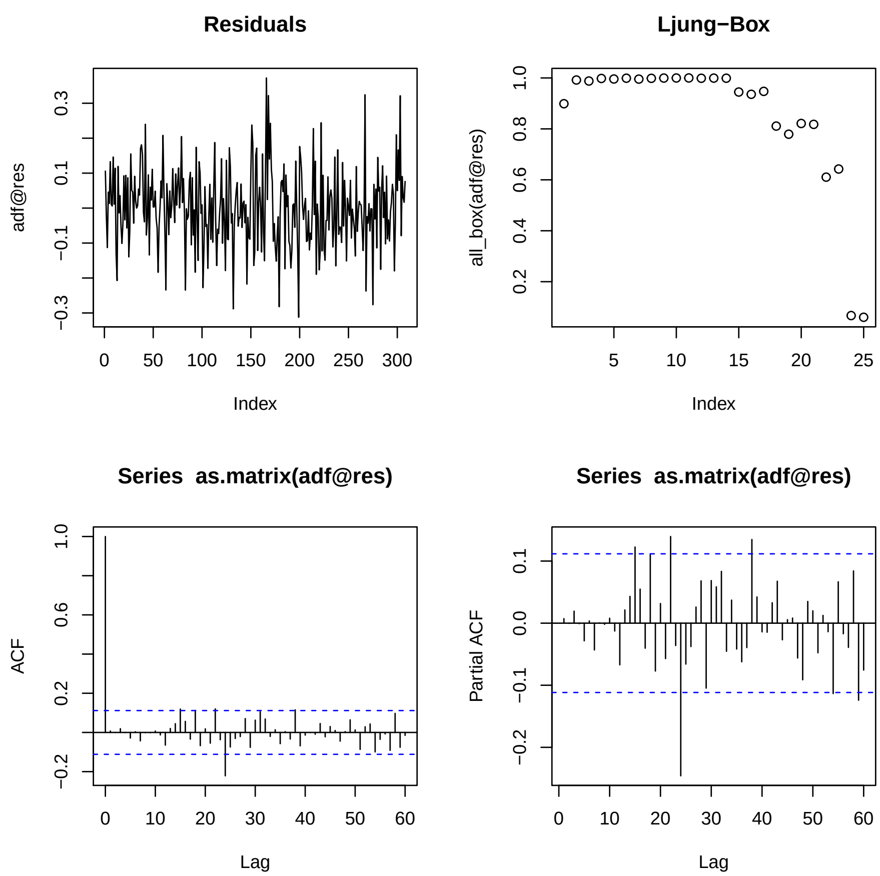
```
lag <- select_adf(na.omit(data2$Diff),"none")
lag

## [1] 13
```

Escolhemos a ordem de lag 13.

```
adf <- ur.df(na.omit(data2$Diff), lags=lag)
```

Figura 30: Resíduos

```
par(mfrow = c(2,2))
plot(adf@res, type='l', main='Residuals')
plot(all_box(adf@res), main='Ljung-Box')
acf(as.matrix(adf@res), lag.max=60)
pacf(as.matrix(adf@res), lag.max=60)
```



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 as estatísticas do teste e os valores críticos

```r
adf@teststat
```

```
##                 tau1
## statistic -3.491532
```

```r
adf@cval
```

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

O valor da estatística do teste é de 3,4915. 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).

7.1.4 Diferença Sazonal

A função de autocorrelação indica que existem lags sazonais significativos, no entando, o decaimento é rápido, indicanco não existência de raiz unitária sazonal.Testamos a seguir a hipótese nula de não presença de raiz unitária sazonal com o teste de Canova e Hansen (Canova e Hansen 1995).

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

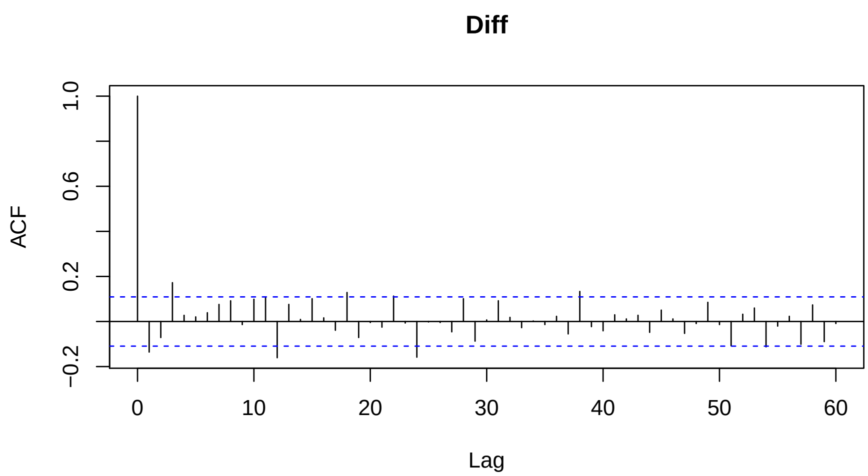
```
## [1] 0.6117258
```

O teste retornou um p-valor de 0.6117, o que significa que não podemos rejeitar a hipótese nula de não presença de raiz unitária sazonal.

7.2 Identificação das Possíveis Formas Funcionais

Figura 31: FAC da Primeira Diferença

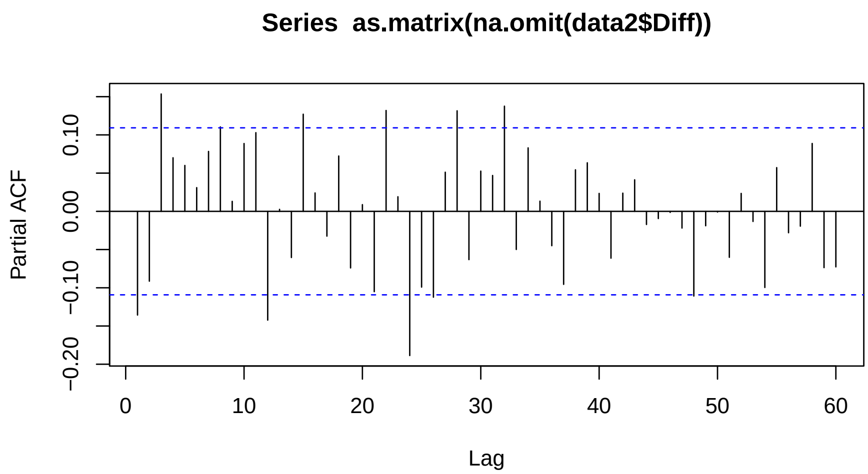
```r
acf(as.matrix(na.omit(data2$Diff)), lag.max=60)
```



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

Figura 32: FACP da Primeira Diferença

```
pacf(as.matrix(na.omit(data2$Diff)), lag.max=60)
```



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 (3,1,3)(2,0,2)12.

7.3 Estimação

Na subseção anterior definimos as ordems 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 os modelos que passam o teste dos resíduos que apresentam melhores critérios de informação (AIC e BIC). Esses passos são realizados no *chunk* abaixo.

```
order <- select_model(data2$Value,3,1,3,2,0,2,12)
```

Abaixo, as ordens dos melhores modelos pelos critérios AIC e BIC.

```
# Ordem pelo critério AIC
print("p q P Q")

## [1] "p q P Q"

order[,1]

## [1] 3 1 0 2
```

```r
# Ordem pelo critério BIC
print("p q P Q")

## [1] "p q P Q"

order[,2]

## [1] 1 2 0 2
```

O modelo com melhor critério AIC é o SARIMA (3,1,1)(0,0,2)12. O modelo com melhor critério BIC é o SARIMA (1,1,2)(0,0,2)12. Pelo critério da parcimônia selecionamos o modelo com melhor critério BIC, pois é o que apresenta menor quantidade de parâmetros.

```r
model2 <- sarima(data2$Value,
                 1, 1, 2,
                 0, 0, 2, 12)
```

7.4 Diagnóstico dos Resíduos

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

7.4.1 Independência

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

Figura 33: Resíduos

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

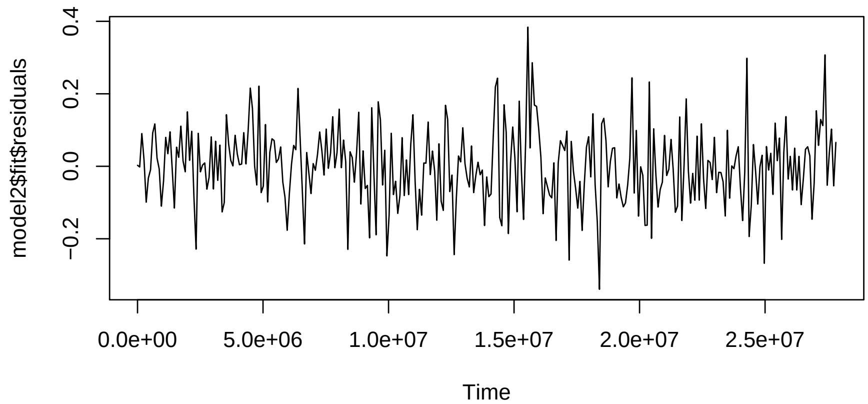


Figura 34: FAC dos Resíduos

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

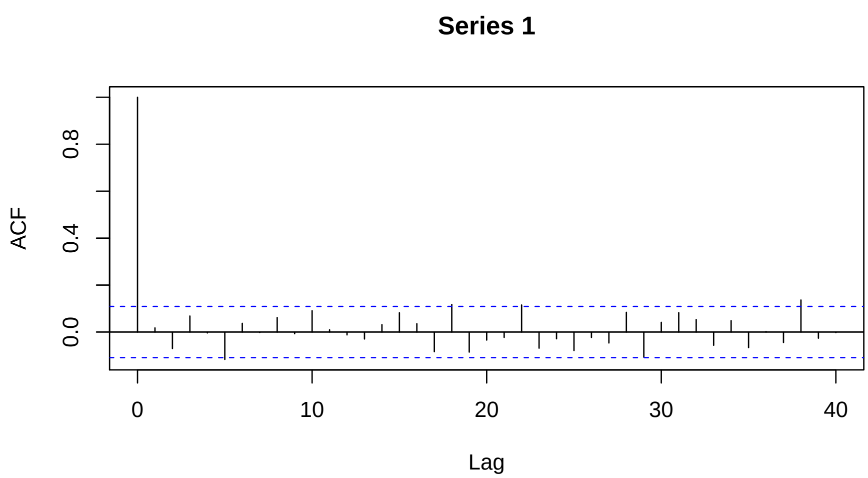
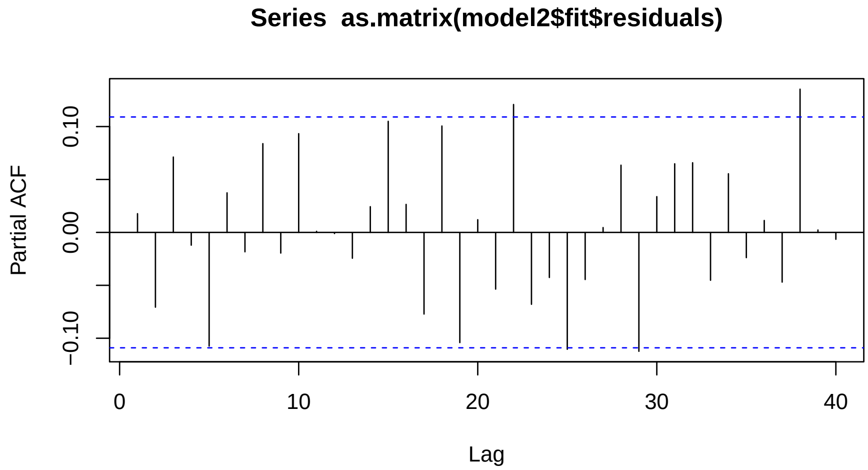


Figura 35: FACP dos Resíduos

```r
pacf(as.matrix(model2$fit$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, já que a quantidade de lags significativos é a esperada para um nível de significância de 5%. o 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.

```r
boxs <- all_box(model2$fit$residuals)
```

Figura 36: P-Valores de Ljung-Box

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

Abaixo, o valor mínimo do teste de Ljung-Box entre todos os lags considerados.

```
min(boxs)

## [1] 0.1309436
```

Os p-valores não rejeitam a hipótese nula de independência da distribuição, uma vez que todos são superiores a 0,05.

7.4.2 Homoscedasticidade

Testamos a homoscedasticidade dos resíduos do modelo selecionado apicando a análise acima nos resíduos ao quadrado.

Figura 37: Resíduos ao Quadrado

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

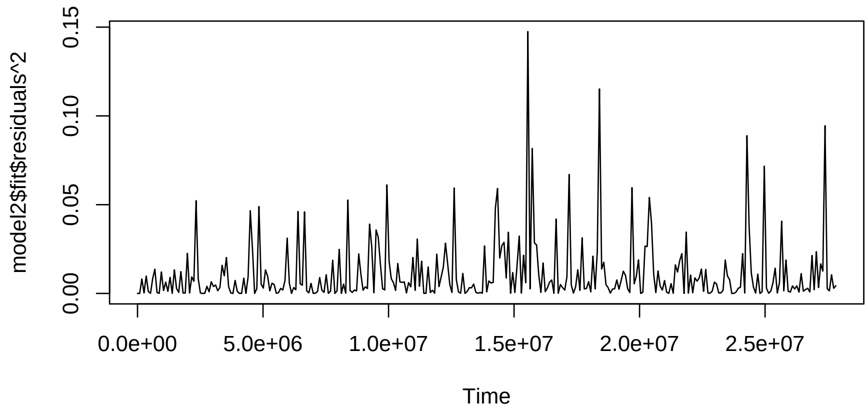


Figura 38: FAC dos Resíduos ao Quadrado

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

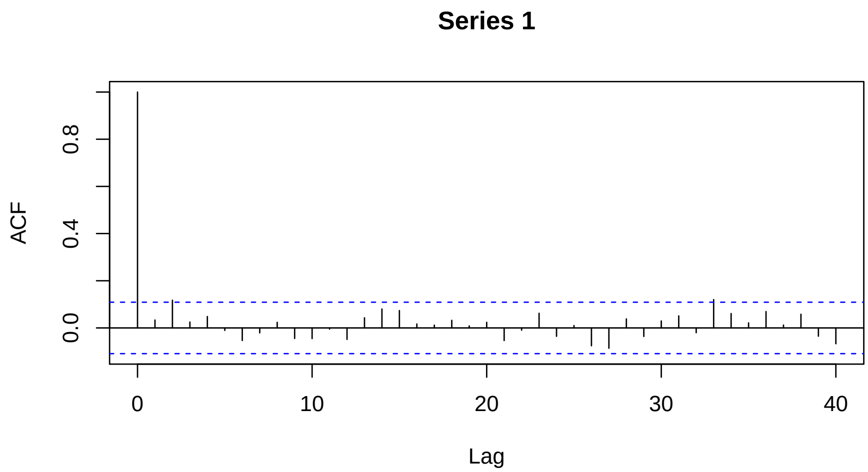
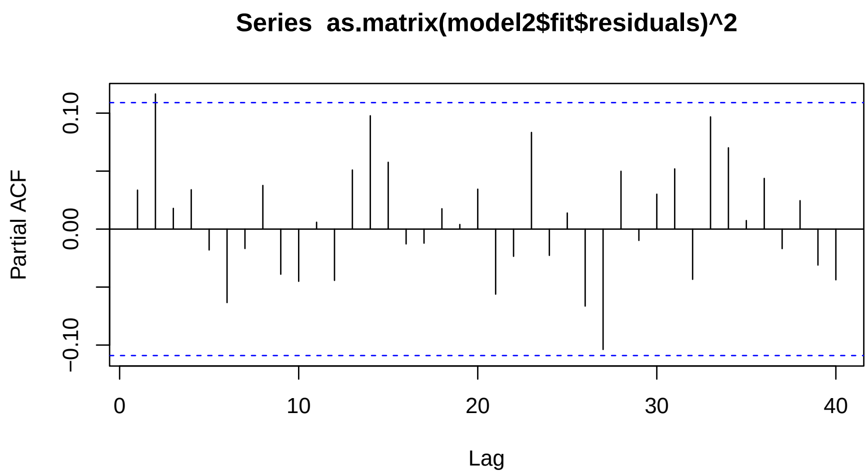


Figura 39: FACP dos Resíduos ao Quadrado

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

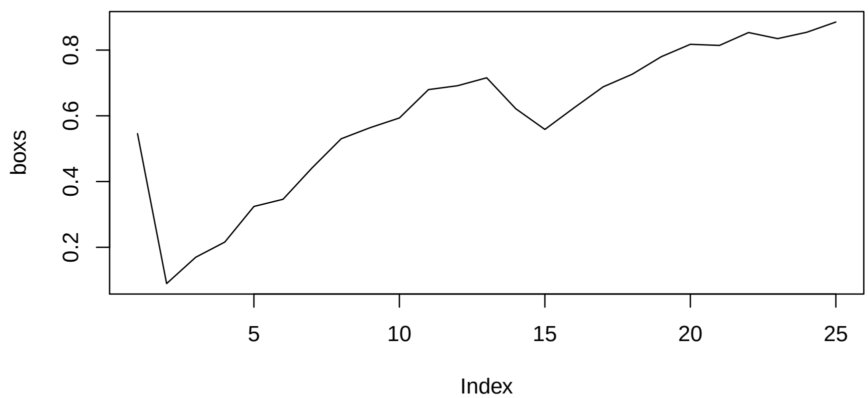


Tanto o gráfico dos resíduos ao quadrado quanto a sua função de autocorrelação e sua função de autocorrelação parcial indicam homoscedasticidade. Para testar essa hipótese, realizamos o teste de Ljung-Box nos resíduos ao quadrado para todos os lags até o vinte e cinco. Os testes são realizados no *chunk*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

```
boxs <- all_box(model2$fit$residuals^2)
```

Figura 40: P-Valores de Ljung-Box

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



Abaixo, o valor mínimo do teste de Ljung-Box entre todos os lags considerados.

```
min(boxs)

## [1] 0.08952865
```

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.

7.5 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 10 períodos da amostra. Faremos previsão recursiva, ou seja: prevemos sempre apenas o período imediatamente subsequente, utilizando todos os dados até então.

```r
fs <- prediction(data2$Value,1,2,0,2)
```

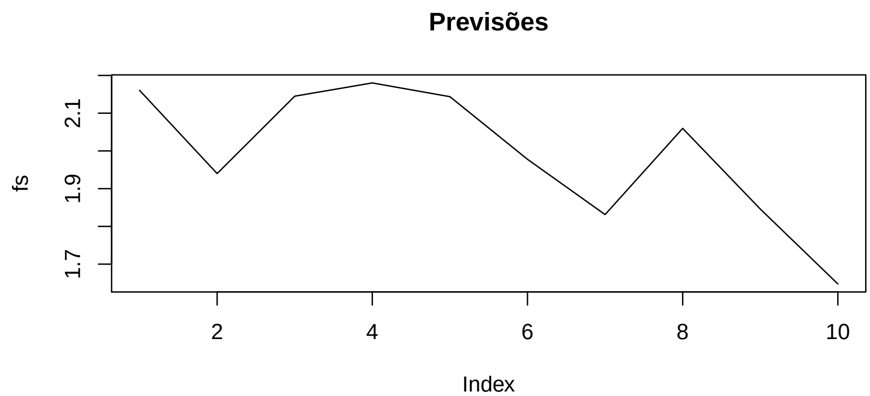
Abaixo, as previsões realizadas.

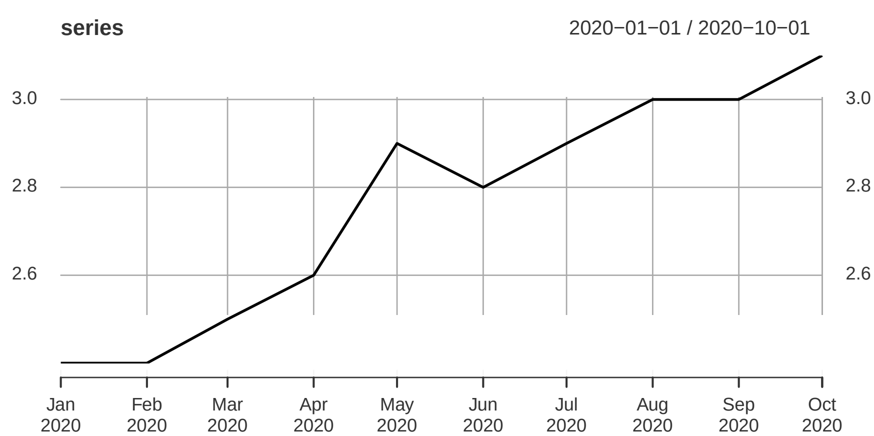
```r
fs

##  [1] 2.160667 1.939964 2.144997 2.180143 2.143683 1.977818 1.831369 2.059727
##  [9] 1.846058 1.647538
```

Abaixo, os gráficos das previsões e da série original.

Figura 41: Previsões e Série Original

```
length <- length(data2$Value)
start  <- length - 9
series <- data2$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 0.7668036 0.8536119 0.7668036 26.84425 26.84425
```

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] 0.3080207
```

O índice do teste foi de 0,308, 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 Grolemund 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.