

Trabalho Prático 2

Análise de Séries Temporais - 1/2023

Ana Tércia Freires da Silva -

Gabriel Vêras Monteiro - 19/0106794

Gabriela Carneiro de Almeida - 18/0120816

Table of contents

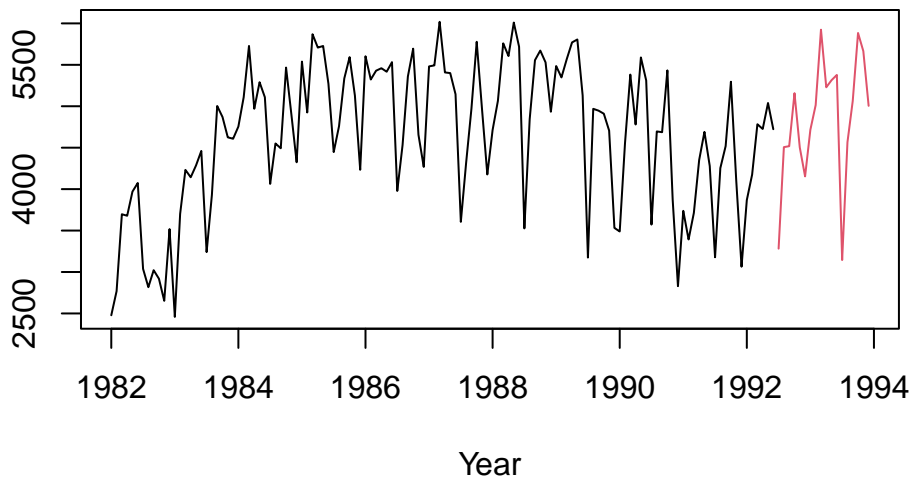
Introdução: série selecionada, características e decomposição	2
Modelos ARIMA: seleção, transformações e resíduos	4
Sem transformação de Box-Cox	4
Com transformação de Box-Cox	11
Modelos ETS: seleção, transformações e resíduos	17
Resíduos	31
Modelo com transformação	31
Seleção	31
Resíduos	33
Estudo de desempenho preditivo	33
Resultados da Janela Deslizante	33
Performance em relação aos horizontes de previsão	33
ARIMA	33
ETS	33
Resultados	33
Apêndice	33

Introdução: série selecionada, características e decomposição

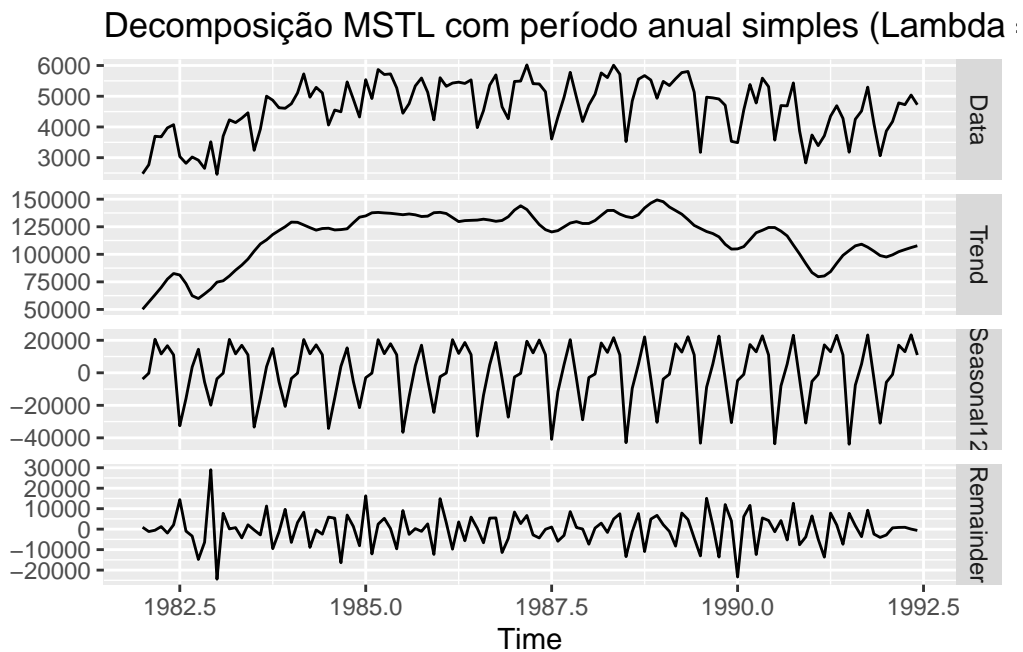
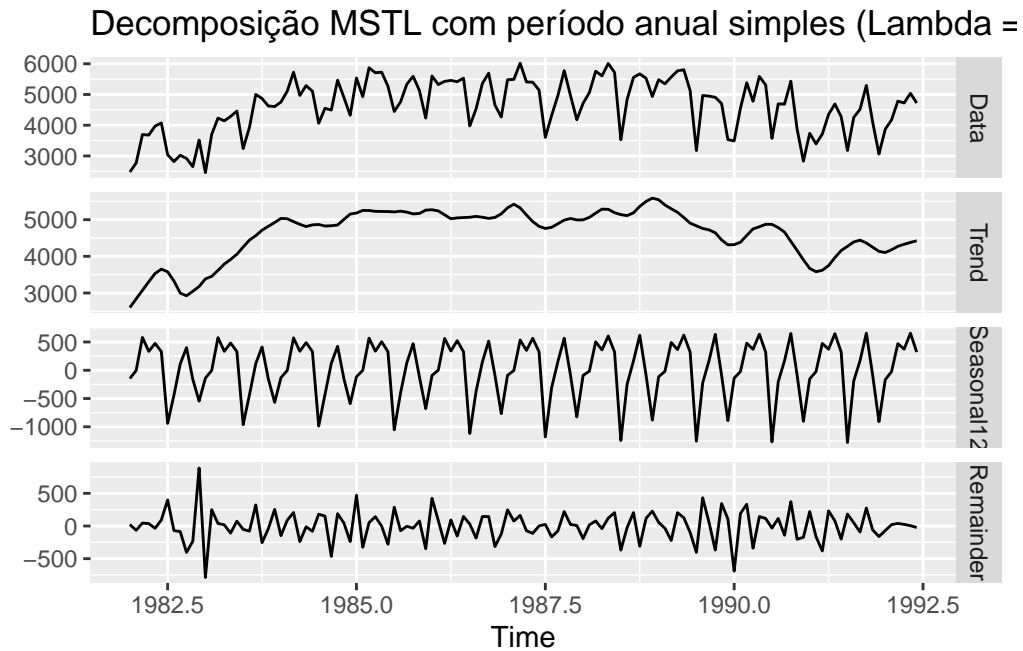
A série temporal escolhida foi a de número *id* correspondente a 1891. De acordo com a definição do próprio pacote, refere-se a *Pneumatic casings, original equipment*. Foram realizadas medidas mensais de 1982 a 1992 e o horizonte de previsão requerido é das 18 ocorrências seguintes.

O gráfico da série, com *in* e *out-sample*, é exposto a seguir.

Série Temporal M3–1891



Após visualização inicial da série, foi feita sua decomposição via MSTL. Foram feitas duas decomposições, uma ajustando $\lambda = \text{"auto"}$ e $\lambda = \text{NULL}$, resultando em duas composições muito similares.



Aplicando a decomposição MSTL com sazonalidade anual, é possível visualizar uma tendência de crescimento no início das medições, passando por uma estabilização e posterior decréscimo. Ao final, a série parece retomar um pequeno crescimento. Além disso, é possível notar que a série tem um componente sazonal mensal, com ciclos claros. Por fim, o ruído parece se

aproximar de um padrão de ruído branco.

Modelos ARIMA: seleção, transformações e resíduos

Sem transformação de Box-Cox

```
KPSS Test for Level Stationarity
```

```
data:  serie_ms
```

```
KPSS Level = 0.52764, Truncation lag parameter = 4, p-value = 0.03544
```

```
[1] 1
```

```
[1] 1
```

```
KPSS Test for Level Stationarity
```

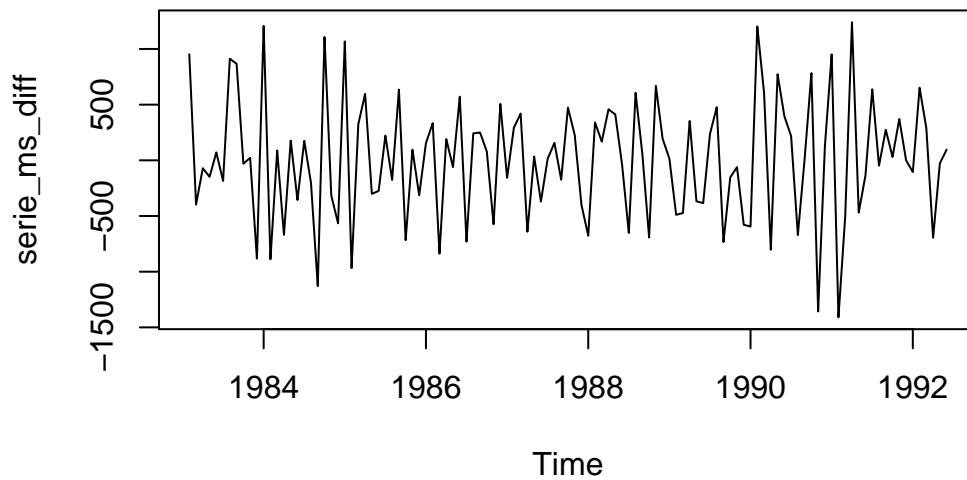
```
data:  serie_ms_diff
```

```
KPSS Level = 0.044309, Truncation lag parameter = 4, p-value = 0.1
```

Inicialmente foi feito o teste de estacionaridade KPSS na série segundo as hipóteses:

$$\begin{cases} H_0 : \text{O processo é estacionário} \\ H_1 : \text{O processo possui raiz unitária} \end{cases} \quad (1)$$

O teste indica que a série não é estacionária ($p\text{-valor} = 0,03544$), portanto, é necessário aplicar diferenciações para tornar a série estacionária antes de ajustar um modelo adequado que se ajuste a série em análise. O número de diferenciações simples necessárias é igual a um, dessa forma, a estimativa para ‘ d ’ do modelo SARIMA é de ‘ d ’ = 1. Além disso, em se tratando de uma série sazonal, é necessário verificar o número de diferenciações sazonais para retirar o efeito da sazonalidade. Analogamente, foi obtido o valor de uma diferenciação sazonal necessária, se tratando, portanto, do ‘ D ’ do modelo SARIMA, sendo ‘ D ’ = 1. Para a diferenciação sazonal, pelo ciclo sazonal ser igual a 12, é necessário a utilização do lag sazonal igual a 12 ($\text{lag} = 12$).

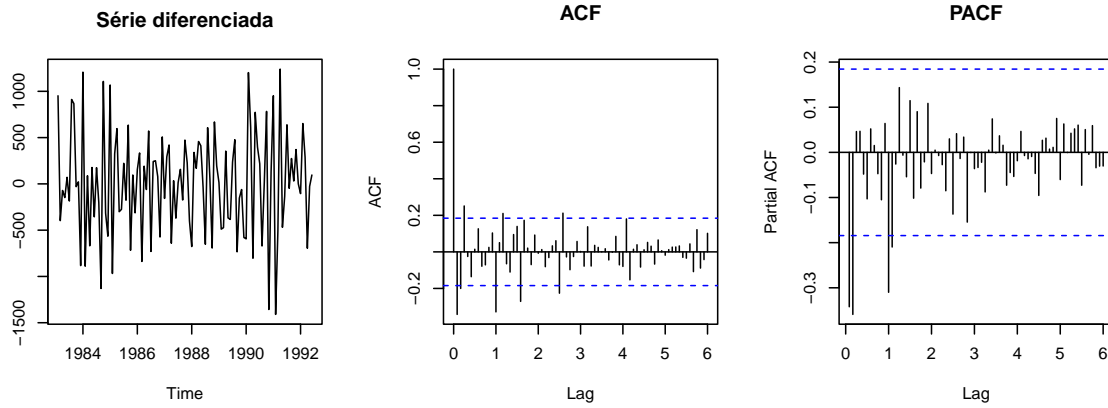


KPSS Test for Level Stationarity

```
data:  serie_ms_diff
KPSS Level = 0.044309, Truncation lag parameter = 4, p-value = 0.1
```

O gráfico acima mostra o comportamento da série após as diferenciações simples e sazonal. A série aparenta ter comportamento estacionário, o que foi confirmado a partir da aplicação do teste KPSS, cujas hipóteses já foram explicitadas anteriormente. Por meio do teste foi possível notar que a série realmente está estacionária após as diferenciações ($p\text{-valor} = 0,1$), dessa forma, é possível prosseguir com o procedimento de seleção do modelo.

Para tal, se segue a análise dos gráficos ACF e PACF.

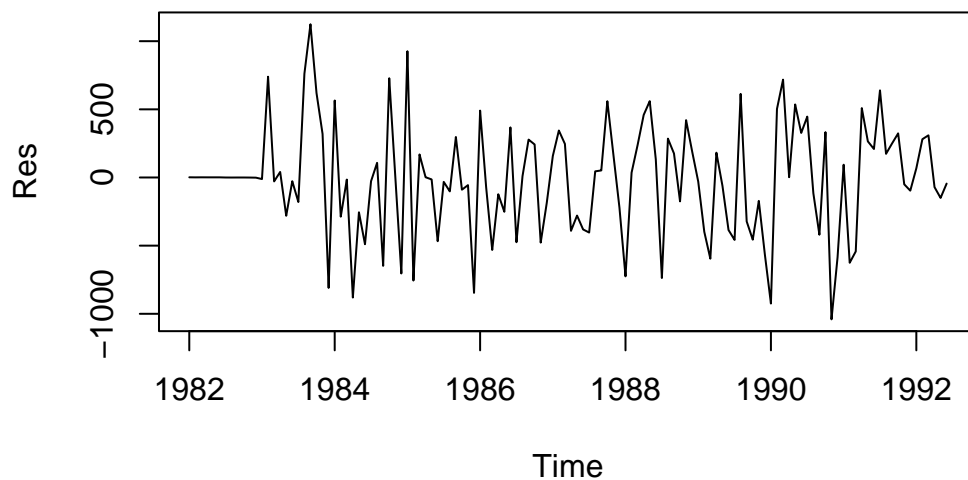


É possível perceber que no ACF há uma quebra no primeiro lag simples, ao passo que ocorre uma quebra no segundo lag do PACF. Porém, nenhum dos gráficos apresenta comportamento bem definido, dificultando a determinação do restante dos parâmetros do modelo SARIMA. Nesse contexto, foi feito a combinação dos valores de p e q , entre 0 e 2, e de P e Q , nos valores 0 ou 1, desconsiderando quando p e q fossem zero simultaneamente e, analogamente, para P e Q . O melhor modelo, criado pelas combinações de valores explicitados acima, foi selecionado segundo o menor valor de AICc.

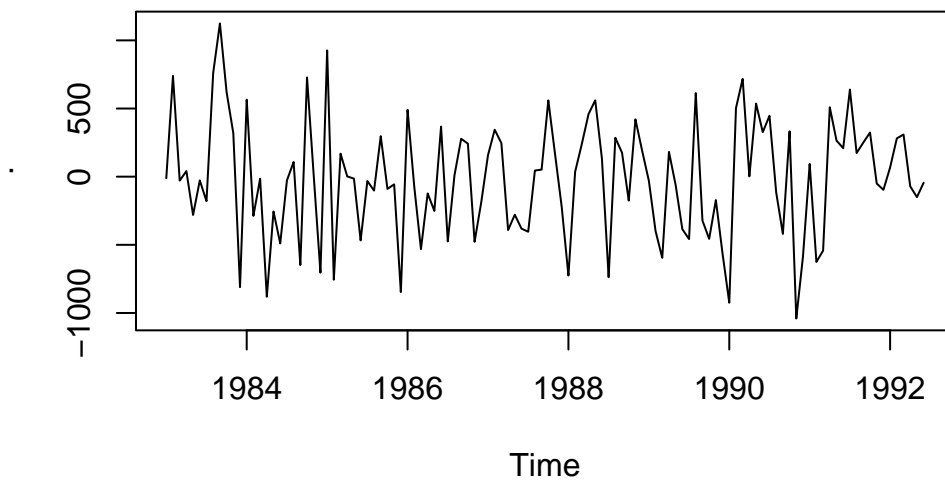
$p = 0$, $d = 1$, $q = 0$, $P = 0$, $D = 1$, $Q = 0$, AICc = 1750.899
 $p = 0$, $d = 1$, $q = 1$, $P = 0$, $D = 1$, $Q = 0$, AICc = 1729.316
 $p = 0$, $d = 1$, $q = 3$, $P = 0$, $D = 1$, $Q = 0$, AICc = 1729.225
 $p = 1$, $d = 1$, $q = 2$, $P = 0$, $D = 1$, $Q = 0$, AICc = 1729.027
 $p = 2$, $d = 1$, $q = 0$, $P = 0$, $D = 1$, $Q = 0$, AICc = 1724.938
 $p = 0$, $d = 1$, $q = 1$, $P = 0$, $D = 1$, $Q = 1$, AICc = 1709.211
 $p = 2$, $d = 1$, $q = 0$, $P = 0$, $D = 1$, $Q = 1$, AICc = 1706.182

[1] 1706.182

A partir dos AICc, o modelo selecionado foi $SARIMA(2,1,0)(0,1,1)_{12}$, assim, pode-se prosseguir com a análise dos resíduos.



É notável que há uma sequência de resíduos iguais a zero no início da série, e estes podem afetar as análises, dessa forma, os zeros foram desconsiderados, sendo feita análise de resíduos de um ano após o início da série, dados pela figura a seguir:



KPSS Test for Level Stationarity

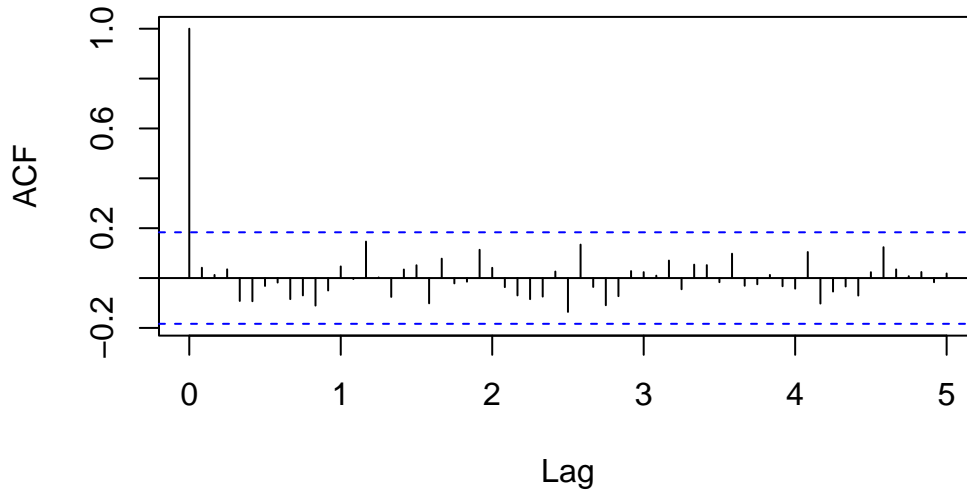
data: E

KPSS Level = 0.062374, Truncation lag parameter = 4, p-value = 0.1

Os resíduos aparentam ser estacionários, com média 0 e com variância constante. Agora, foi testada a estacionariedade com um teste KPSS, cujas as hipóteses já foram mencionadas. Como resultado, obtem-se que o processo é estacionário (p-valor $> 0, 1$).

Além disso, a partir do gráfico do ACF, mostrado abaixo, pode-se verificar a independência dos resíduos.

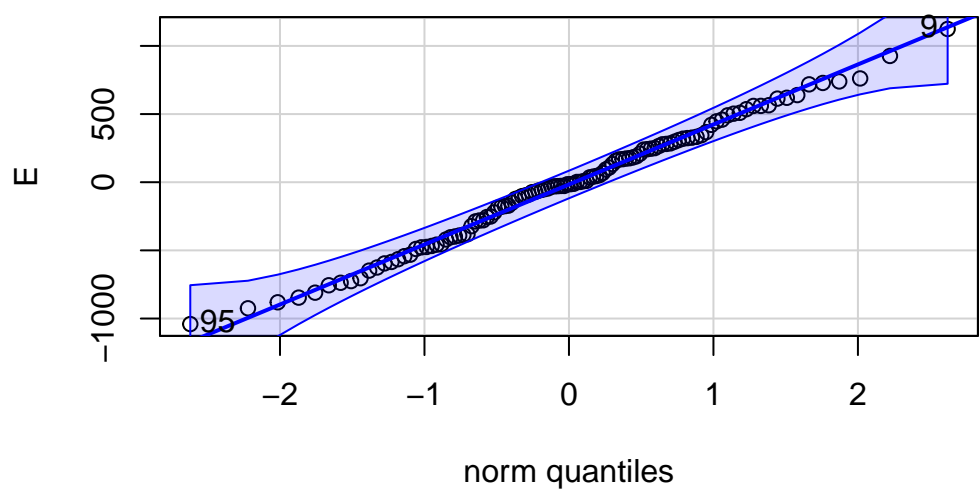
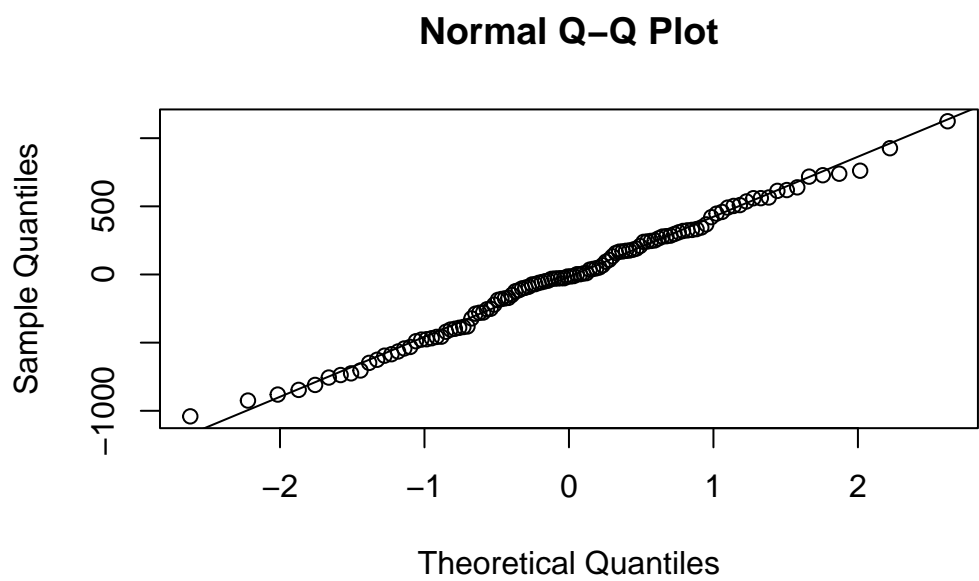
Series E



Observamos que praticamente todos os valores das autocorrelações não são significantes, indicando a independência dos resíduos, o que necessita a confirmação pelo teste de Ljung-Box, com base nas hipóteses:

$$\begin{cases} H_0 : \text{Todas as correlações são iguais a zero} \\ H_1 : \text{Ao menos uma correlação é diferente de zero} \end{cases} \quad (2)$$

Assim, utilizando o teste com 20 graus de liberdade, obteve-se p-valor de 0, 8923, e para 15, obteve-se 0, 8741, dessa maneira, utilizando o $\alpha = 0, 05$, não se rejeita H_0 , corroborando com a análise gráfica de que os resíduos são independentes. Então, pode-se testar a normalidade, primeiramente, por meio de uma representação gráfica.



[1] 9 95

Shapiro-Wilk normality test

```
data: E
W = 0.99331, p-value = 0.8598
```

Box-Ljung test

```
data: E
X-squared = 12.639, df = 20, p-value = 0.8923
```

```
      ar1      ar2      sma1
-0.5096443 -0.3354277 -0.5080884
```

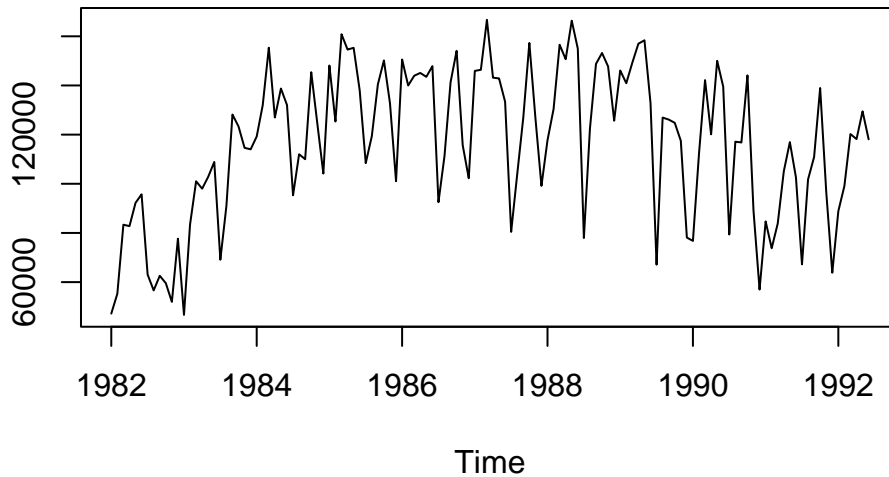
```
[1] 189154.6
```

Assim, pela imagem, com envelope de 95%, é basicamente certa a normalidade dos resíduos, para confirmar, é conduzido o teste de Shapiro-Wilk, sob hipóteses:

$$\begin{cases} H_0 : \text{Os resíduos seguem distribuição normal} \\ H_1 : \text{Os resíduos não seguem distribuição normal} \end{cases} \quad (3)$$

Assim, com nível de significância de 0, 05, o teste de Shapiro-Wilk confirma que os resíduos estão distribuídos segundo uma distribuição normal.

Com transformação de Box-Cox



Percebe-se grande similaridade com a série original, e, pode-se prosseguir da mesma maneira que anteriormente com o procedimento de seleção. Primeiramente, testando a estacionariedade da série com o teste KPSS a um nível de significância de 5%, é possível concluir que a série não é estacionária ($p\text{-valor} = 0,03658$). Dessa maneira, analogamente à análise da série sem a transformação Box-Cox, é necessário aplicar 1 diferenciação simples e de uma diferenciação sazonal para tornar a série estacionária, sendo $d = 1$ e $D = 1$. Da mesma maneira que na análise anterior, foi utilizado um sazonal de 12 para a diferenciação sazonal, dessa maneira, a série diferenciada é a que se segue:

KPSS Test for Level Stationarity

data: x2

KPSS Level = 0.5226, Truncation lag parameter = 4, p-value = 0.03658

[1] 1

[1] 1

Series: x2

ARIMA(0,1,1)(0,1,1)[12]

Coefficients:

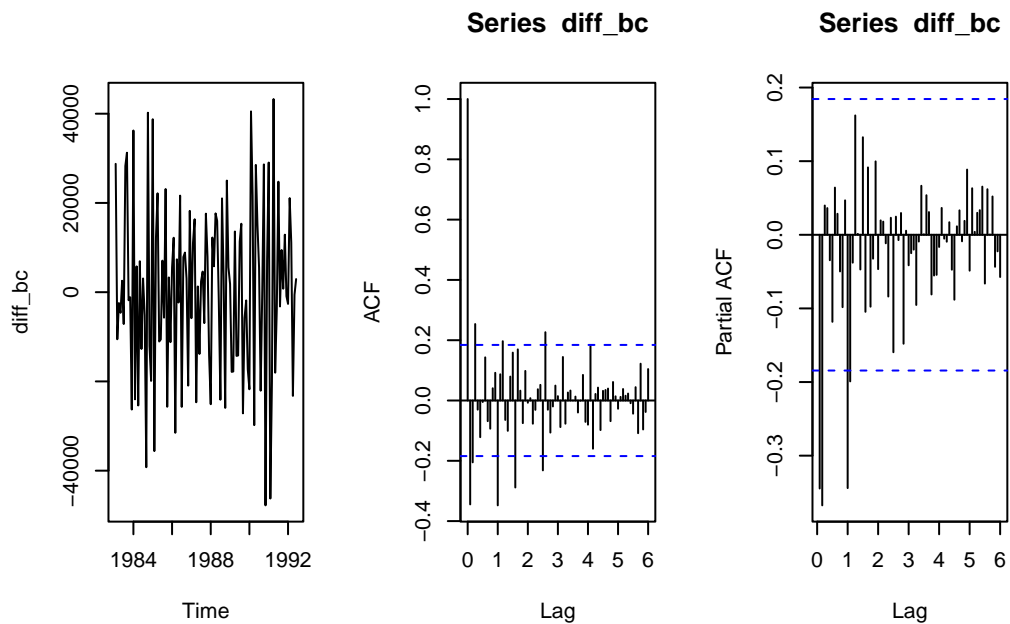
	ma1	sma1
	-0.5166	-0.5532
s.e.	0.0825	0.1060

$\sigma^2 = 238912745$: log likelihood = -1251.65
AIC=2509.31 AICc=2509.53 BIC=2517.49

KPSS Test for Level Stationarity

data: diff_bc

KPSS Level = 0.044088, Truncation lag parameter = 4, p-value = 0.1



Assim ficou a série a partir de 1 diferenciação simples e sazonal. Agora, analogamente à série sem transformação, é possível observar que a série tem aparência estacionária, o que deve ser confirmado novamente a partir do teste de KPSS de sob hipóteses já explicitadas. Assim, a partir de um p-valor maior que 0.1, pôde-se confirmar que a série diferenciada é estacionária e que é possível prosseguir com o procedimento de seleção.

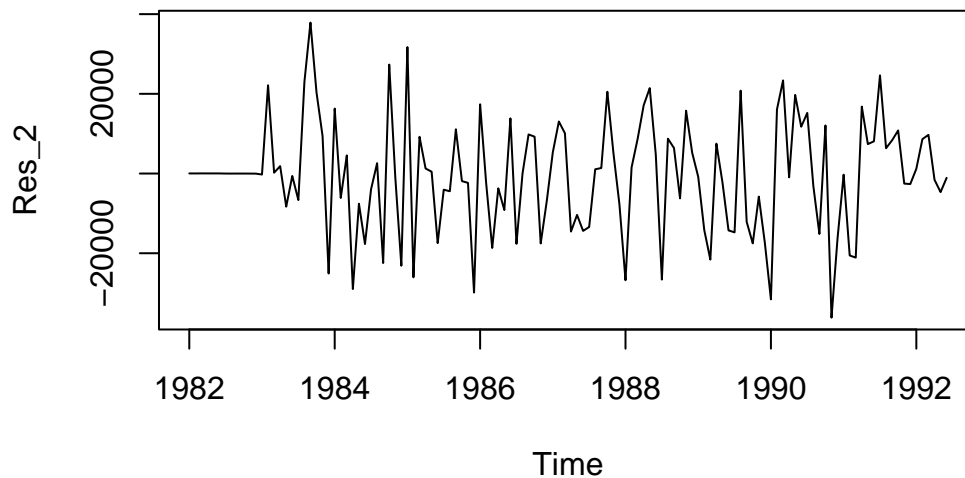
$p = 0$, $d = 1$, $q = 0$, $P = 0$, $D = 1$, $Q = 0$, AICc = 2553.741
 $p = 0$, $d = 1$, $q = 1$, $P = 0$, $D = 1$, $Q = 0$, AICc = 2531.436

$p = 0$, $d = 1$, $q = 3$, $P = 0$, $D = 1$, $Q = 0$, $AICc = 2531.137$
 $p = 1$, $d = 1$, $q = 2$, $P = 0$, $D = 1$, $Q = 0$, $AICc = 2530.789$
 $p = 2$, $d = 1$, $q = 0$, $P = 0$, $D = 1$, $Q = 0$, $AICc = 2526.922$
 $p = 0$, $d = 1$, $q = 1$, $P = 0$, $D = 1$, $Q = 1$, $AICc = 2509.529$
 $p = 2$, $d = 1$, $q = 0$, $P = 0$, $D = 1$, $Q = 1$, $AICc = 2506.5$

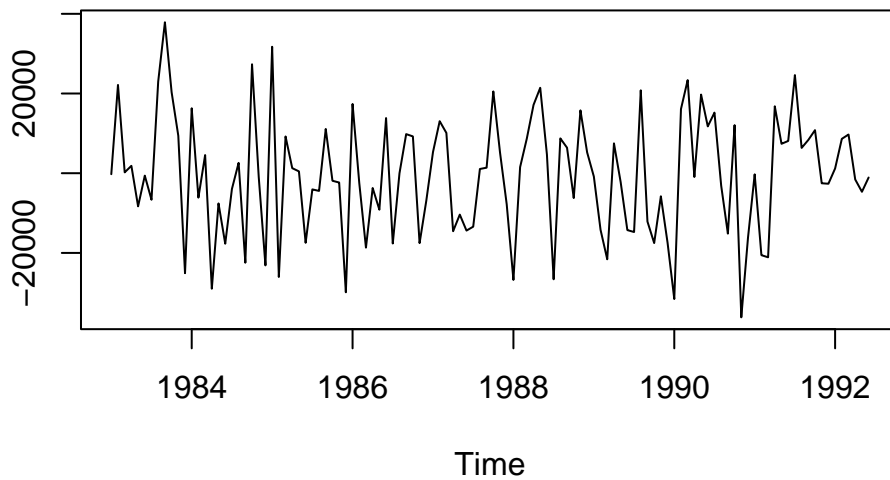
[1] 2506.5

Assim, pode-se observar comportamento similar ao da série sem a transformação de Boxcox. Nesse sentido, no gráfico ACF há uma quebra no primeiro lag simples, ao passo que há uma quebra no segundo lag no gráfico PACF. Nenhum dos gráficos apresentou comportamento bem definido. Para os lags sazonais, percebem-se valores significativos apenas no primeiro lag, sendo assim, os valores de p e q serão testados entre 0 e 2, já P e Q , nos valores 0 ou 1, desconsiderando quando p e q forem 0 simultaneamente, e analogamente para P e Q . A partir dos $AICc$, observou-se o melhor menor valor de 2506, 5 e assim, o modelo ajustado foi um $SARIMA(2, 1, 0)(0, 1, 1)_{12}$, assim com no ajuste de modelo sem utilização da transformação de Box-Cox.

Seguindo para análise dos resíduos, é possível observar que há uma sequência de resíduos iguais a zero no início da série.

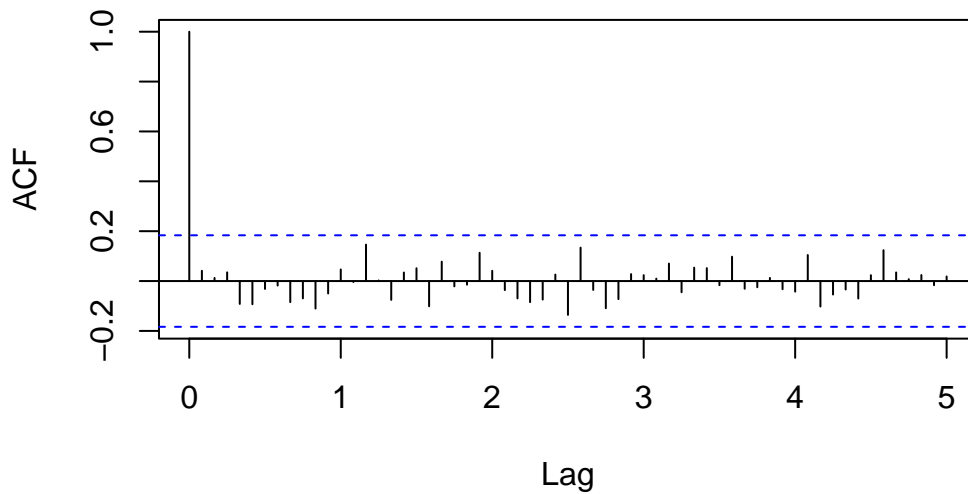


Assim, analogamente à série sem transformação, serão considerados apenas os resíduos um ano após o início das observações, dados pela figura a seguir:

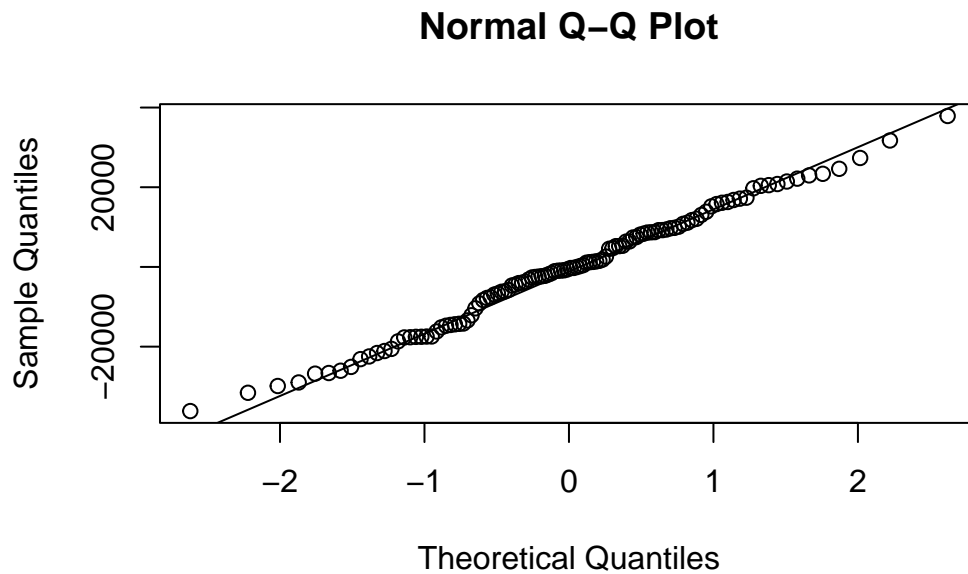


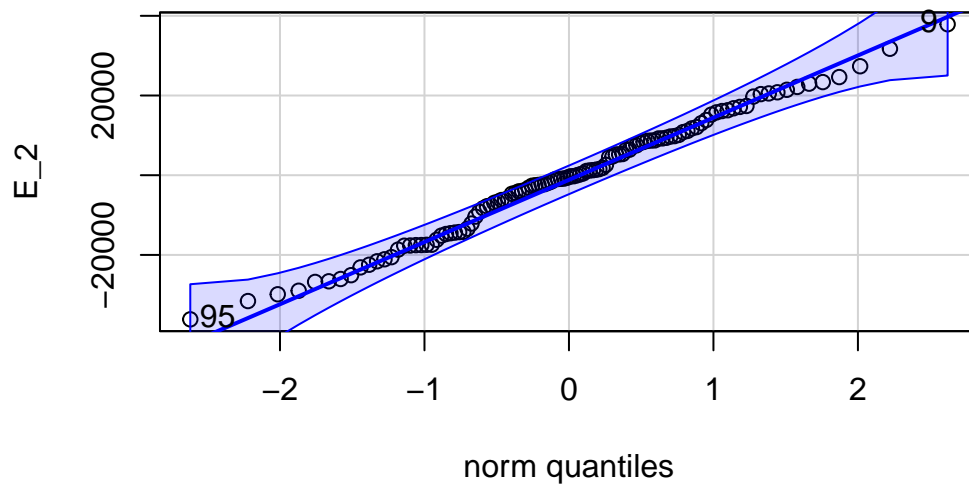
Agora, os resíduos aparentam estacionariedade graficamente, mas para testear realmente, utiliza-se o teste de KPSS. O teste mostra que os resíduos são, de fato, estacionários ($p\text{-valor} > 0, 1$). Além disso, pode-se visualizar pelo gráfico do ACF, a independência dos resíduos:

ACF dos resíduos (Box-Cox)



Assim, de maneira análoga à vista na série sem transformação de Box-Cox, basicamente não há valores significativos, e, para realmente confirmar independência, utiliza-se novamente o teste do tipo Ljung-Box. Assim, com 20 graus de liberdade, o p-valor foi de 0,86, e com 15 graus, o p-valor resultou em 0.8443, portanto, para um nível de significância de 5%, não se rejeita H_0 , indicando que a suspeita após a análise gráfica estava correta, e os resíduos realmente são independentes. Por fim, pode-se prosseguir com a análise da normalidade dos resíduos, por meio da análise gráfica primeiramente.





```
[1] 9 95
```

Shapiro-Wilk normality test

data: E_2

W = 0.99199, p-value = 0.7513

KPSS Test for Level Stationarity

data: E_2

KPSS Level = 0.062576, Truncation lag parameter = 4, p-value = 0.1

Box-Ljung test

data: E_2

X-squared = 13.39, df = 20, p-value = 0.86

ar1	ar2	sma1
-0.5059202	-0.3433980	-0.5264162

```
[1] 224602516
```


A partir da análise gráfica, é possível observar indicação de normalidade dos resíduos. Seguindo com o teste de Shapiro-Wilk, há evidências de que os resíduos seguem distribuição normal (p-valor=0,7513).

Modelos ETS: seleção, transformações e resíduos

```
[1] 1
ETS(A,Ad,A)

Call:
ets(y = dados, model = model$modelo[i], damped = model$damp[i])

Smoothing parameters:
  alpha = 0.46
  beta  = 1e-04
  gamma = 1e-04
  phi   = 0.9628

Initial states:
  l = 2873.2425
  b = 92.0764
  s = -742.5204 -167.6501 488.8474 123.6439 -288.2857 -1061.117
      308.8648 564.7236 357.8395 543.2575 24.055 -151.6583

sigma: 428.6046

      AIC      AICc      BIC
2154.365 2160.757 2205.418
[1] 2
ETS(A,A,A)

Call:
ets(y = dados, model = model$modelo[i], damped = model$damp[i])

Smoothing parameters:
  alpha = 0.4973
  beta  = 0.0149
  gamma = 1e-04

Initial states:
  l = 2881.5847
```

```

      b = 83.4981
      s = -730.8711 -172.4496 489.9627 121.5461 -293.2277 -1060.087
          310.166 563.2084 359.0362 538.7735 26.4968 -152.5545

sigma: 435.4436

      AIC      AICc      BIC
2157.505 2163.171 2205.721
[1] 3
[1] "Error in ets(dados, model = model$modelo[i], damped = model$damp[i]) : \n Forbidden model"
attr(,"class")
[1] "try-error"
attr(,"condition")
<simpleError in ets(dados, model = model$modelo[i], damped = model$damp[i]): Forbidden model
[1] 4
[1] "Error in ets(dados, model = model$modelo[i], damped = model$damp[i]) : \n Forbidden model"
attr(,"class")
[1] "try-error"
attr(,"condition")
<simpleError in ets(dados, model = model$modelo[i], damped = model$damp[i]): Forbidden model
[1] 5
ETS(A,Ad,N)

Call:
ets(y = dados, model = model$modelo[i], damped = model$damp[i])

Smoothing parameters:
  alpha = 0.2022
  beta  = 1e-04
  phi   = 0.9711

Initial states:
  l = 3262.2671
  b = 54.9193

sigma: 699.6524

      AIC      AICc      BIC
2267.017 2267.723 2284.034
[1] 6
ETS(A,A,N)

Call:

```

```
ets(y = dados, model = model$modelo[i], damped = model$damp[i])
```

```
Smoothing parameters:
```

```
alpha = 0.2561
```

```
beta = 1e-04
```

```
Initial states:
```

```
l = 3319.6533
```

```
b = 11.3537
```

```
sigma: 703.2468
```

```
      AIC      AICc      BIC
2267.345 2267.845 2281.526
```

```
[1] 7
```

```
[1] "Error in ets(dados, model = model$modelo[i], damped = model$damp[i]) : \n Forbidden model"
attr(,"class")
```

```
[1] "try-error"
```

```
attr(,"condition")
```

```
<simpleError in ets(dados, model = model$modelo[i], damped = model$damp[i]): Forbidden model
```

```
[1] 8
```

```
[1] "Error in ets(dados, model = model$modelo[i], damped = model$damp[i]) : \n Forbidden model"
attr(,"class")
```

```
[1] "try-error"
```

```
attr(,"condition")
```

```
<simpleError in ets(dados, model = model$modelo[i], damped = model$damp[i]): Forbidden model
```

```
[1] 9
```

```
[1] "Error in ets(dados, model = model$modelo[i], damped = model$damp[i]) : \n Forbidden model"
attr(,"class")
```

```
[1] "try-error"
```

```
attr(,"condition")
```

```
<simpleError in ets(dados, model = model$modelo[i], damped = model$damp[i]): Forbidden model
```

```
[1] 10
```

```
[1] "Error in ets(dados, model = model$modelo[i], damped = model$damp[i]) : \n Forbidden model"
attr(,"class")
```

```
[1] "try-error"
```

```
attr(,"condition")
```

```
<simpleError in ets(dados, model = model$modelo[i], damped = model$damp[i]): Forbidden model
```

```
[1] 11
```

```
[1] "Error in ets(dados, model = model$modelo[i], damped = model$damp[i]) : \n Forbidden model"
attr(,"class")
```

```
[1] "try-error"
```

```
attr(,"condition")
```

```

<simpleError in ets(dados, model = model$modelo[i], damped = model$damp[i]): Forbidden model
[1] 12
[1] "Error in ets(dados, model = model$modelo[i], damped = model$damp[i]) : \n  Forbidden mo
attr(,"class")
[1] "try-error"
attr(,"condition")
<simpleError in ets(dados, model = model$modelo[i], damped = model$damp[i]): Forbidden model
[1] 13
[1] "Error in ets(dados, model = model$modelo[i], damped = model$damp[i]) : \n  Forbidden mo
attr(,"class")
[1] "try-error"
attr(,"condition")
<simpleError in ets(dados, model = model$modelo[i], damped = model$damp[i]): Forbidden model
[1] 14
ETS(A,N,A)

Call:
ets(y = dados, model = model$modelo[i], damped = model$damp[i])

Smoothing parameters:
  alpha = 0.5082
  gamma = 1e-04

Initial states:
  l = 4021.6024
  s = -739.9261 -173.8532 484.3761 131.9132 -289.0238 -1079.279
      320.7234 639.2768 334.7555 547.0071 16.7813 -192.7512

sigma: 446.9512

      AIC      AICc      BIC
2162.348 2166.712 2204.892
[1] 15
[1] "Error in ets(dados, model = model$modelo[i], damped = model$damp[i]) : \n  Forbidden mo
attr(,"class")
[1] "try-error"
attr(,"condition")
<simpleError in ets(dados, model = model$modelo[i], damped = model$damp[i]): Forbidden model
[1] 16
[1] "Error in ets(dados, model = model$modelo[i], damped = model$damp[i]) : \n  Forbidden mo
attr(,"class")
[1] "try-error"
attr(,"condition")

```

```

<simpleError in ets(dados, model = model$modelo[i], damped = model$damp[i]): Forbidden model
[1] 17
[1] "Error in ets(dados, model = model$modelo[i], damped = model$damp[i]) : \n  Forbidden mo
attr(,"class")
[1] "try-error"
attr(,"condition")
<simpleError in ets(dados, model = model$modelo[i], damped = model$damp[i]): Forbidden model
[1] 18
ETS(A,N,N)

```

Call:

```
ets(y = dados, model = model$modelo[i], damped = model$damp[i])
```

Smoothing parameters:

```
alpha = 0.2673
```

Initial states:

```
l = 3135.6167
```

```
sigma: 698.2623
```

```

      AIC      AICc      BIC
2263.601 2263.798 2272.110

```

```
[1] 19
```

```
ETS(M,Ad,A)
```

Call:

```
ets(y = dados, model = model$modelo[i], damped = model$damp[i])
```

Smoothing parameters:

```
alpha = 0.2848
```

```
beta  = 1e-04
```

```
gamma = 1e-04
```

```
phi   = 0.9703
```

Initial states:

```
l = 2865.1634
```

```
b = 94.8011
```

```
s = -741.5687 -168.7565 488.969 123.7533 -288.0692 -1055.661
```

```
      308.9004 564.5967 357.8223 543.1346 24.3001 -157.4208
```

```
sigma: 0.1091
```

```

      AIC      AICc      BIC
2194.143 2200.536 2245.197
[1] 20
ETS(M,A,A)

```

Call:

```
ets(y = dados, model = model$modelo[i], damped = model$damp[i])
```

Smoothing parameters:

```

alpha = 0.3114
beta  = 0.0116
gamma = 1e-04

```

Initial states:

```

l = 2884.0832
b = 93.0202
s = -693.2597 -165.8946 487.6633 180.0481 -292.417 -1047.902
    300.6935 552.5341 275.5544 542.2602 -1.1823 -138.0979

```

sigma: 0.1098

```

      AIC      AICc      BIC
2196.119 2201.786 2244.336
[1] 21
ETS(M,Ad,M)

```

Call:

```
ets(y = dados, model = model$modelo[i], damped = model$damp[i])
```

Smoothing parameters:

```

alpha = 0.3659
beta  = 9e-04
gamma = 1e-04
phi   = 0.9768

```

Initial states:

```

l = 2902.7456
b = 90.5988
s = 0.8623 0.9587 1.0966 1.02 0.9395 0.7798
    1.0788 1.1318 1.0769 1.1159 0.9953 0.9446

```

sigma: 0.103

```

      AIC      AICc      BIC
2179.418 2185.810 2230.471
[1] 22
ETS(M,A,M)

```

Call:

```
ets(y = dados, model = model$modelo[i], damped = model$damp[i])
```

Smoothing parameters:

```

alpha = 0.3308
beta  = 0.0105
gamma = 1e-04

```

Initial states:

```

l = 2908.8774
b = 83.9825
s = 0.8561 0.9567 1.0901 1.0216 0.9383 0.7799
    1.0936 1.1445 1.0914 1.1194 0.9845 0.924

```

sigma: 0.1048

```

      AIC      AICc      BIC
2184.055 2189.722 2232.272
[1] 23
ETS(M,Ad,N)

```

Call:

```
ets(y = dados, model = model$modelo[i], damped = model$damp[i])
```

Smoothing parameters:

```

alpha = 0.2222
beta  = 1e-04
phi   = 0.9679

```

Initial states:

```

l = 3263.0794
b = 78.9552

```

sigma: 0.1515

```

      AIC      AICc      BIC
2267.284 2267.990 2284.302
[1] 24

```

ETS(M,A,N)

Call:

```
ets(y = dados, model = model$modelo[i], damped = model$damp[i])
```

Smoothing parameters:

alpha = 0.3295

beta = 0.0043

Initial states:

l = 3262.7271

b = 72.8391

sigma: 0.1514

AIC	AICc	BIC
-----	------	-----

2269.144	2269.644	2283.325
----------	----------	----------

[1] 25

```
[1] "Error in ets(dados, model = model$modelo[i], damped = model$damp[i]) : \n Forbidden model"
attr(,"class")
```

[1] "try-error"

```
attr(,"condition")
```

```
<simpleError in ets(dados, model = model$modelo[i], damped = model$damp[i]): Forbidden model
```

[1] 26

```
[1] "Error in ets(dados, model = model$modelo[i], damped = model$damp[i]) : \n Forbidden model"
attr(,"class")
```

[1] "try-error"

```
attr(,"condition")
```

```
<simpleError in ets(dados, model = model$modelo[i], damped = model$damp[i]): Forbidden model
```

[1] 27

ETS(M,Md,M)

Call:

```
ets(y = dados, model = model$modelo[i], damped = model$damp[i])
```

Smoothing parameters:

alpha = 0.3705

beta = 0.0045

gamma = 1e-04

phi = 0.9698

Initial states:

l = 2912.1767


```

b = 1.0349
s = 0.8469 0.9568 1.1026 1.0285 0.9386 0.7826
    1.0772 1.1308 1.0838 1.1176 0.9921 0.9424

sigma: 0.1036

      AIC      AICc      BIC
2181.290 2187.683 2232.343
[1] 28
ETS(M,M,M)

Call:
ets(y = dados, model = model$modelo[i], damped = model$damp[i])

Smoothing parameters:
  alpha = 0.5088
  beta  = 0.016
  gamma = 0.2117

Initial states:
  l = 2913.1548
  b = 1.018
  s = 0.9015 0.8722 1.0315 1.0343 0.9391 0.8241
    1.0926 1.0937 1.0985 1.1675 1.0174 0.9277

sigma: 0.1086

      AIC      AICc      BIC
2192.252 2197.918 2240.469
[1] 29
ETS(M,Md,N)

Call:
ets(y = dados, model = model$modelo[i], damped = model$damp[i])

Smoothing parameters:
  alpha = 0.2129
  beta  = 1e-04
  phi   = 0.958

Initial states:
  l = 3262.3304
  b = 1.0242

```

```

sigma: 0.1518

      AIC      AICc      BIC
2267.707 2268.413 2284.725
[1] 30
ETS(M,M,N)

Call:
ets(y = dados, model = model$modelo[i], damped = model$damp[i])

Smoothing parameters:
  alpha = 0.3019
  beta  = 1e-04

Initial states:
  l = 3261.8504
  b = 1.004

sigma: 0.1537

      AIC      AICc      BIC
2268.625 2269.125 2282.806
[1] 31
[1] "Error in ets(dados, model = model$modelo[i], damped = model$damp[i]) : \n  Forbidden mo
attr("class")
[1] "try-error"
attr("condition")
<simpleError in ets(dados, model = model$modelo[i], damped = model$damp[i]): Forbidden model
[1] 32
ETS(M,N,A)

Call:
ets(y = dados, model = model$modelo[i], damped = model$damp[i])

Smoothing parameters:
  alpha = 0.3794
  gamma = 1e-04

Initial states:
  l = 4022.7944
  s = -722.5848 -226.1109 480.6945 122.9523 -196.8044 -1055.387
      250.5592 656.0401 367.2533 563.3214 -43.8181 -196.1153

```

```

sigma: 0.1216

      AIC      AICc      BIC
2216.087 2220.451 2258.631
[1] 33
[1] "Error in ets(dados, model = model$modelo[i], damped = model$damp[i]) : \n Forbidden model"
attr(,"class")
[1] "try-error"
attr(,"condition")
<simpleError in ets(dados, model = model$modelo[i], damped = model$damp[i]): Forbidden model
[1] 34
ETS(M,N,M)

Call:
ets(y = dados, model = model$modelo[i], damped = model$damp[i])

Smoothing parameters:
  alpha = 0.3108
  gamma = 0.0022

Initial states:
  l = 4061.2745
  s = 0.8621 0.9501 1.1041 1.0246 0.9445 0.7811
      1.0861 1.1315 1.0851 1.1101 0.9928 0.9279

sigma: 0.1124

      AIC      AICc      BIC
2196.415 2200.778 2238.959
[1] 35
[1] "Error in ets(dados, model = model$modelo[i], damped = model$damp[i]) : \n Forbidden model"
attr(,"class")
[1] "try-error"
attr(,"condition")
<simpleError in ets(dados, model = model$modelo[i], damped = model$damp[i]): Forbidden model
[1] 36
ETS(M,N,N)

Call:
ets(y = dados, model = model$modelo[i], damped = model$damp[i])

Smoothing parameters:

```

```

alpha = 0.3085

Initial states:
  l = 3191.2984

sigma: 0.1551

      AIC      AICc      BIC
2265.474 2265.671 2273.983
[1] 37
[1] "Error in ets(dados, model = model$modelo[i], damped = model$damp[i]) : \n Invalid error t
attr(,"class")
[1] "try-error"
attr(,"condition")
<simpleError in ets(dados, model = model$modelo[i], damped = model$damp[i]): Invalid error t
[1] 38
[1] "Error in ets(dados, model = model$modelo[i], damped = model$damp[i]) : \n Invalid error
attr(,"class")
[1] "try-error"
attr(,"condition")
<simpleError in ets(dados, model = model$modelo[i], damped = model$damp[i]): Invalid error t
[1] 39
[1] "Error in ets(dados, model = model$modelo[i], damped = model$damp[i]) : \n Invalid error
attr(,"class")
[1] "try-error"
attr(,"condition")
<simpleError in ets(dados, model = model$modelo[i], damped = model$damp[i]): Invalid error t
[1] 40
[1] "Error in ets(dados, model = model$modelo[i], damped = model$damp[i]) : \n Invalid error
attr(,"class")
[1] "try-error"
attr(,"condition")
<simpleError in ets(dados, model = model$modelo[i], damped = model$damp[i]): Invalid error t
[1] 41
[1] "Error in ets(dados, model = model$modelo[i], damped = model$damp[i]) : \n Invalid error
attr(,"class")
[1] "try-error"
attr(,"condition")
<simpleError in ets(dados, model = model$modelo[i], damped = model$damp[i]): Invalid error t
[1] 42
[1] "Error in ets(dados, model = model$modelo[i], damped = model$damp[i]) : \n Invalid error
attr(,"class")
[1] "try-error"

```

```

attr("condition")
<simpleError in ets(dados, model = model$modelo[i], damped = model$damp[i]): Invalid error ty
[1] 43
[1] "Error in ets(dados, model = model$modelo[i], damped = model$damp[i]) : \n  Invalid error
attr("class")
[1] "try-error"
attr("condition")
<simpleError in ets(dados, model = model$modelo[i], damped = model$damp[i]): Invalid error ty
[1] 44
[1] "Error in ets(dados, model = model$modelo[i], damped = model$damp[i]) : \n  Invalid error
attr("class")
[1] "try-error"
attr("condition")
<simpleError in ets(dados, model = model$modelo[i], damped = model$damp[i]): Invalid error ty
[1] 45
[1] "Error in ets(dados, model = model$modelo[i], damped = model$damp[i]) : \n  Invalid error
attr("class")
[1] "try-error"
attr("condition")
<simpleError in ets(dados, model = model$modelo[i], damped = model$damp[i]): Invalid error ty
[1] 46
[1] "Error in ets(dados, model = model$modelo[i], damped = model$damp[i]) : \n  Invalid error
attr("class")
[1] "try-error"
attr("condition")
<simpleError in ets(dados, model = model$modelo[i], damped = model$damp[i]): Invalid error ty
[1] 47
[1] "Error in ets(dados, model = model$modelo[i], damped = model$damp[i]) : \n  Invalid error
attr("class")
[1] "try-error"
attr("condition")
<simpleError in ets(dados, model = model$modelo[i], damped = model$damp[i]): Invalid error ty
[1] 48
[1] "Error in ets(dados, model = model$modelo[i], damped = model$damp[i]) : \n  Invalid error
attr("class")
[1] "try-error"
attr("condition")
<simpleError in ets(dados, model = model$modelo[i], damped = model$damp[i]): Invalid error ty
[1] 49
[1] "Error in ets(dados, model = model$modelo[i], damped = model$damp[i]) : \n  Invalid error
attr("class")
[1] "try-error"
attr("condition")

```

```

<simpleError in ets(dados, model = model$modelo[i], damped = model$damp[i]): Invalid error ty
[1] 50
[1] "Error in ets(dados, model = model$modelo[i], damped = model$damp[i]) : \n  Invalid error
attr(,"class")
[1] "try-error"
attr(,"condition")
<simpleError in ets(dados, model = model$modelo[i], damped = model$damp[i]): Invalid error ty
[1] 51
[1] "Error in ets(dados, model = model$modelo[i], damped = model$damp[i]) : \n  Invalid error
attr(,"class")
[1] "try-error"
attr(,"condition")
<simpleError in ets(dados, model = model$modelo[i], damped = model$damp[i]): Invalid error ty
[1] 52
[1] "Error in ets(dados, model = model$modelo[i], damped = model$damp[i]) : \n  Invalid error
attr(,"class")
[1] "try-error"
attr(,"condition")
<simpleError in ets(dados, model = model$modelo[i], damped = model$damp[i]): Invalid error ty
[1] 53
[1] "Error in ets(dados, model = model$modelo[i], damped = model$damp[i]) : \n  Invalid error
attr(,"class")
[1] "try-error"
attr(,"condition")
<simpleError in ets(dados, model = model$modelo[i], damped = model$damp[i]): Invalid error ty
[1] 54
[1] "Error in ets(dados, model = model$modelo[i], damped = model$damp[i]) : \n  Invalid error
attr(,"class")
[1] "try-error"
attr(,"condition")
<simpleError in ets(dados, model = model$modelo[i], damped = model$damp[i]): Invalid error ty

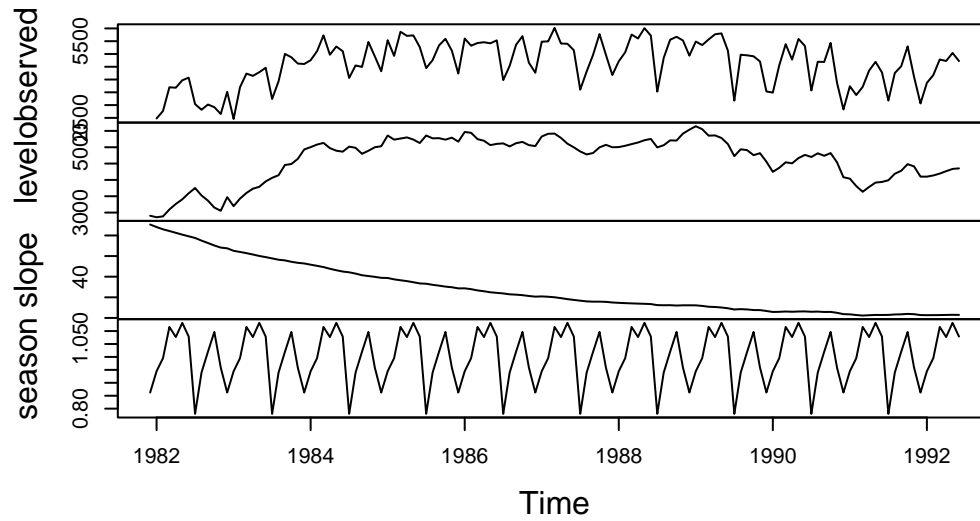
```

Modelo	AIC	AICc	BIC
ETS(Ad,A,A)	2154.36	2160.76	2205.42
ETS(A,A,A)	2157.50	2163.17	2205.72
ETS(A,N,A)	2162.35	2166.71	2204.89
ETS(M,Ad,M)	2179.42	2185.81	2230.47
ETS(M,M,M)	2181.29	2187.68	2232.34
ETS(M,A,M)	2184.06	2189.72	2232.27
ETS(M,M,M)	2192.25	2197.92	2240.47
ETS(M,Ad,A)	2194.14	2200.54	2245.20

(continued)

Modelo	AIC	AICc	BIC
ETS(M,A,A)	2196.12	2201.79	2244.34
ETS(M,N,M)	2196.41	2200.78	2238.96

Decomposition by ETS(M,Ad,M) method



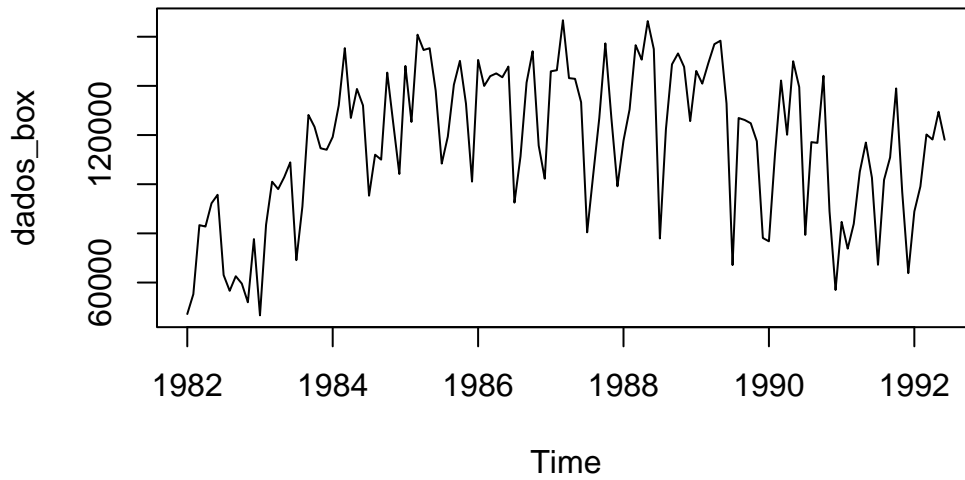
Resíduos

Modelo com transformação

Seleção

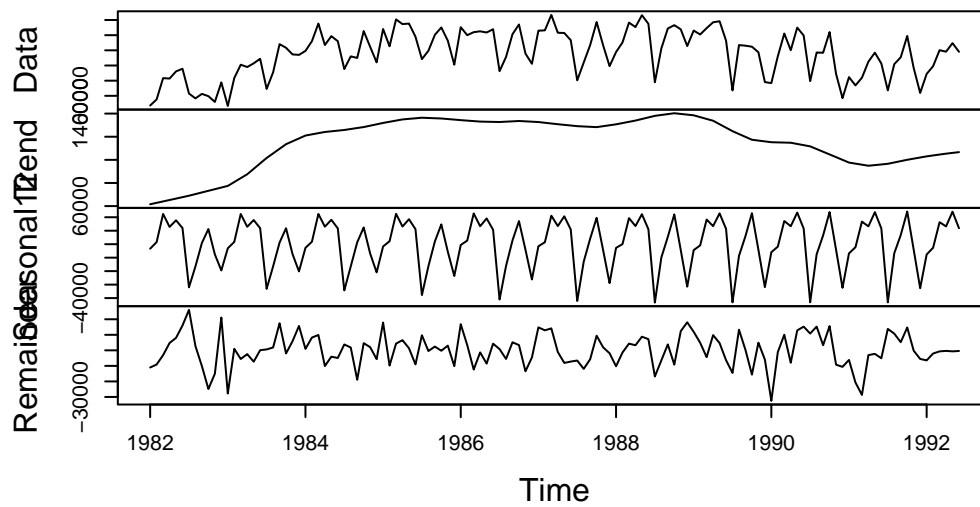
a série com transformacao

Série com transformação Box–Cox $\lambda = 1.422$



decomposicao

Decomposição da série com transformação Box–Cox



selecao do modelo com transformação

Modelo transformado	AIC	AICc	BIC
ETS(Ad,A,A)	2154.36	2160.76	2205.42
ETS(A,A,A)	2157.50	2163.17	2205.72
ETS(A,N,A)	2162.35	2166.71	2204.89
ETS(M,Ad,M)	2179.42	2185.81	2230.47
ETS(M,M,M)	2181.29	2187.68	2232.34
ETS(M,A,M)	2184.06	2189.72	2232.27
ETS(M,M,M)	2192.25	2197.92	2240.47
ETS(M,Ad,A)	2194.14	2200.54	2245.20
ETS(M,A,A)	2196.12	2201.79	2244.34
ETS(M,N,M)	2196.41	2200.78	2238.96

OS MODELOS SAO OS MESMO, PODEMO SELECIONAR O SEGUNDO MELHOR

Resíduos

Estudo de desempenho preditivo

Resultados da Janela Deslizante

Performance em relação aos horizontes de previsão

ARIMA

ETS

Resultados

apresente em tabelas e gráficos as previsões dos 4 modelos selecionados e também apresente em uma tabela os resultados de acurácia dos 4 modelos selecionados e dos modelos benchmarks. Comente os resultados de modo objetivo;

Apêndice

Todo o projeto de composição deste documento pode ser encontrado aqui: https://github.com/cesar-galvao/trabalhos_series