

Trabalho Prático 2

Análise de Séries Temporais - 1/2023

Ana Carolina Vianna - 18/0097261

César Augusto Galvão - 19/0011572

Yan Flávio Vianna - 14/0166149

Table of contents

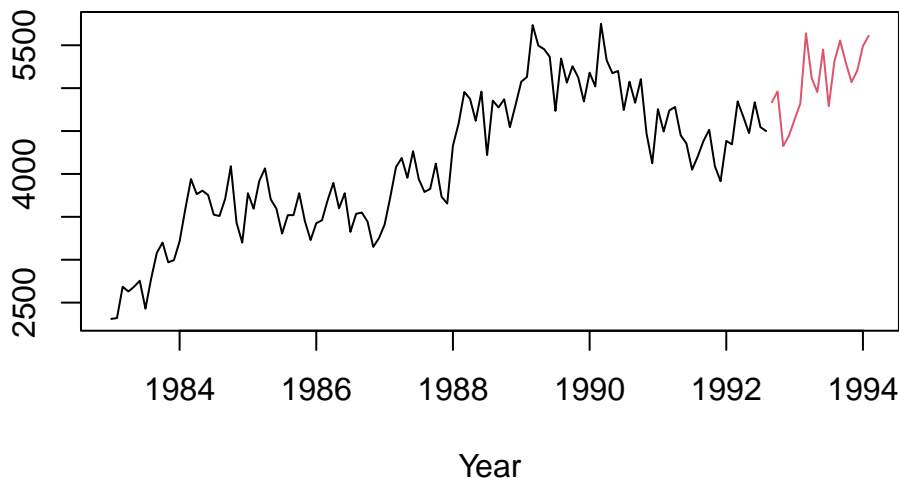
Introdução: série selecionada, características e decomposição	2
Modelos ARIMA: seleção, transformações e resíduos	3
Modelo sem transformação	3
Modelo com transformação	5
Modelos ETS: seleção, transformações e resíduos	7
Modelo sem transformação	7
Modelo com transformação	10
Estudo de desempenho preditivo	11
Resultados da Janela Deslizante	13
Performance em relação aos horizontes de previsão	13
Gráficos da previsão pontual e da previsão intervalar dos 4 modelos selecionados	14
ARIMA	15
ARIMA com transformação	16
ETS	17
ETS com transformação	18
Resultados	19
Conclusão	19
Apêndice	19

Introdução: série selecionada, características e decomposição

A série temporal escolhida foi a de número *id* correspondente a 2183. De acordo com a definição do próprio pacote, refere-se a *Fluid power shipments - hydraulic index*. Foram realizadas medidas mensais de 1983 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.

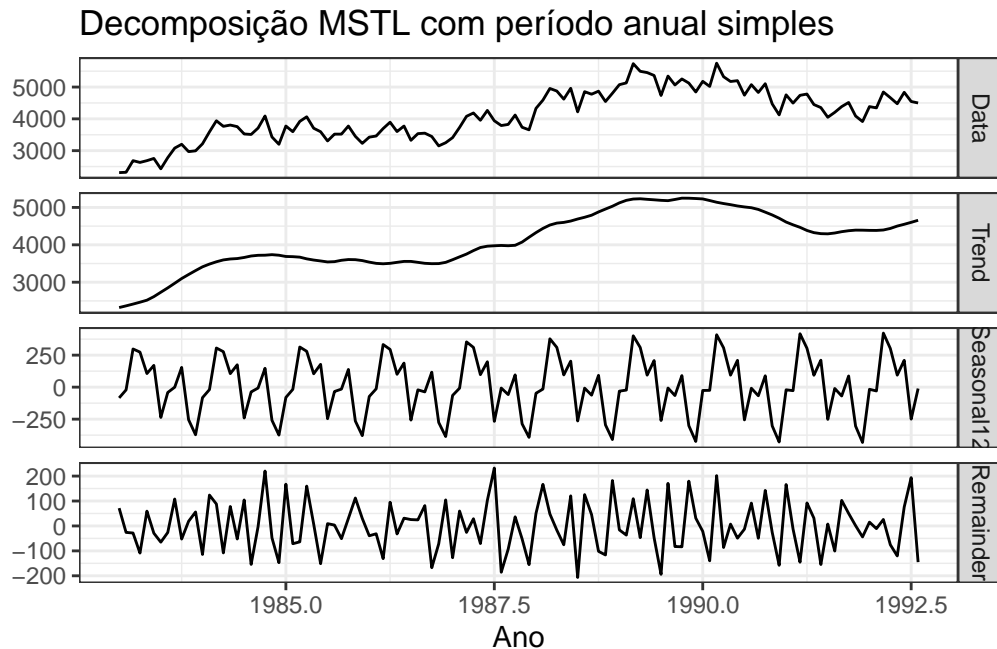
Série Temporal M3-2183



A série aparenta ter dois períodos, pelo menos: um ciclo anual e outro que compreende um período maior. No entanto, ao se tentar decompor a série com múltiplas sazonalidades, obté-se o seguinte:

- **Adicionando uma componente sazonal com ciclo menor que 1 ano** – uma das componentes sazonais apresenta heteroscedasticidade;
- **Adicionando uma componente sazonal com ciclo maior que 1 ano** – resíduos apresentam periodicidade ou heteroscedasticidade.

Optou-se portanto pela decomposição STL (apesar de os dados terem inicialmente formado um objeto `msts`) apenas com a sazonalidade anual, mas fica evidente que esta decomposição não é adequada quando se avalia a componente de tendência, que aparenta ainda carregar algum componente periódico. Os resíduos aparentam um comportamento aleatório e têm média - 0.104, o que é próximo de zero o suficiente considerando a magnitude dos dados da série. A decomposição é exposta a seguir.



Modelos ARIMA: seleção, transformações e resíduos

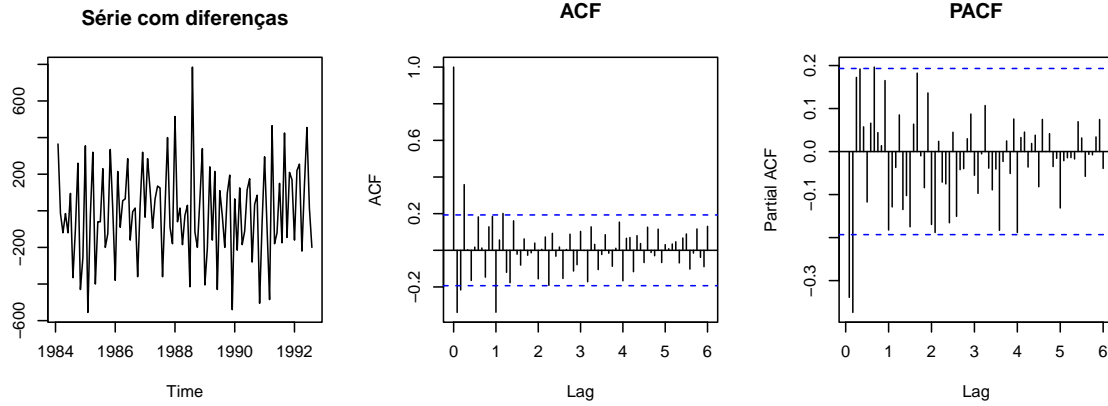
Modelo sem transformação

Seleção

Primeiramente, utilizou-se as funções *ndiffs()* e *nsdiffs()* do pacote *forecast* para identificar quantas diferenças simples e sazonais seriam necessárias para que a série se tornasse estacionária. Concluiu-se pelo resultado dessas funções que são necessárias uma diferenciação simples e uma sazonal. O teste KPSS confirma isso ao não rejeitar a hipótese nula de estacionariedade da série (com diferenças já aplicadas) ao nível de 5% de significância.

	Estatística	p-valor
KPSS Test for Level Stationarity	0.11	0.1

Prosseguimos com a seleção do melhor modelo ARIMA avaliando os gráficos de ACF e PACF. O primeiro parece apresentar quebra no primeiro lag sazonal, enquanto o segundo tem quebra no segundo lag simples, configurando um $ARIMA(2, 1, 0) \times (0, 1, 1)_{12}$ (porém os resíduos para este modelo não ficam muito bons). Entretanto, como não fica nítido um comportamento de queda amortizada, preferiu-se utilizar outro critério para a seleção do modelo.

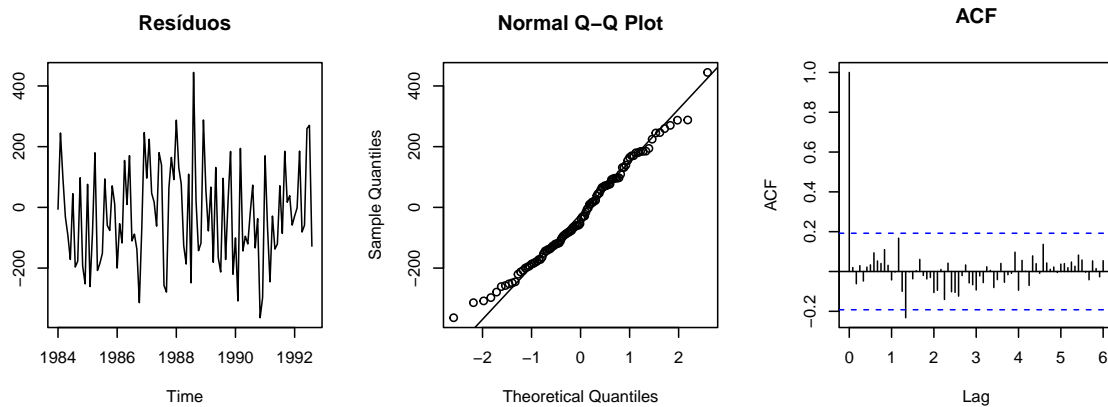


Optou-se pela varredura de combinações de p , q , P e Q , com d e D fixados em 1, como resultado das diferenciações já avaliadas. Utilizando o critério de Akaike corrigido, seleciona-se o modelo $\text{ARIMA}(2, 1, 2) \times (0, 1, 2)_{12}$ para a série, que possui o menor escore entre os modelos testados.

Ao se utilizar a função `auto.arima()`, recebe-se um modelo sugerido $\text{ARIMA}(2, 1, 2) \times (2, 1, 0)_{12}$, porém com AICc superior àquele identificado na varredura. Opta-se pelo modelo selecionado manualmente.

Resíduos

Foram retirados os zeros da inicialização para possibilitar a análise dos resíduos. Observa-se pelo gráfico que os resíduos são aleatórios e aparentemente centrados em zero, com variação constante. Além disso, verifica-se uma distribuição aproximadamente normal, mas com caudas mais pesadas. Finalmente, o gráfico ACF apresenta que a autocorrelação dos resíduos está, em sua grande maioria, dentro da banda de confiança, com exceção de um ponto, que extrapola ligeiramente a margem.



Por fim, realiza-se testes de hipótese para independência e normalidade (o teste KPSS para estacionariedade já foi apresentado) e seus resultados são apresentados na tabela a seguir. De fato, o teste de Shapiro-Wilk não rejeita a normalidade da distribuição dos resíduos apesar de o gráfico QQ apresentar caudas pesadas. Além disso, o teste Ljung-Box com *lag* igual a 15 também não rejeita a independência entre os resíduos e, conseqüentemente, os dados da série.

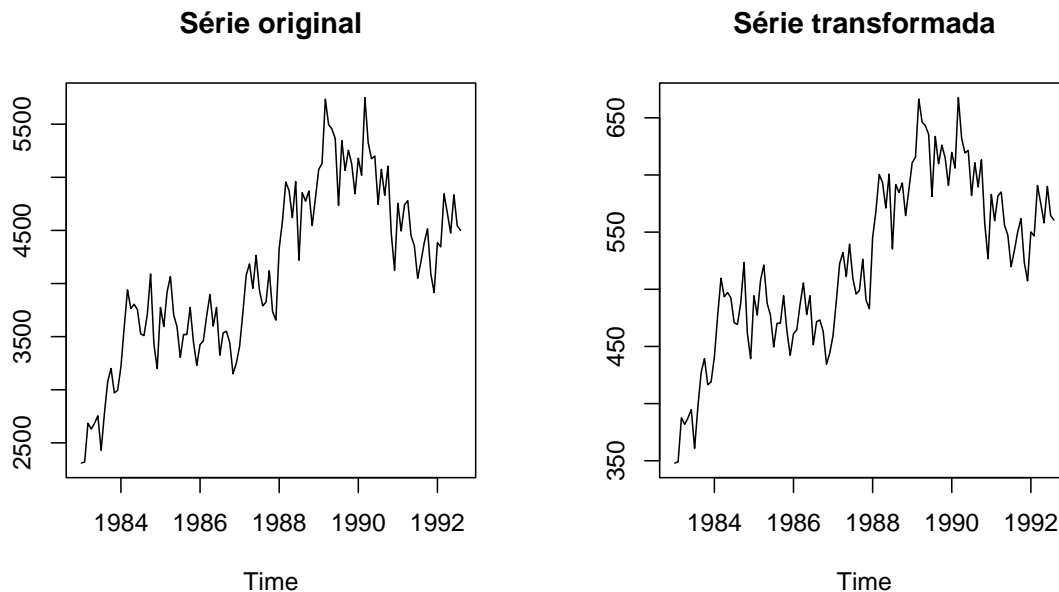
	Estatística	p-valor	Lag
Box-Ljung test	8.90	0.88	15
Shapiro-Wilk normality test	0.99	0.35	

Modelo com transformação

Seleção

Foi utilizada a função *BoxCox.lambda()* do pacote *forecast* para decidir de forma automatizada o melhor valor de lambda para a transformação de Box-Cox. A função sugere um valor de $\lambda = 0.71$.

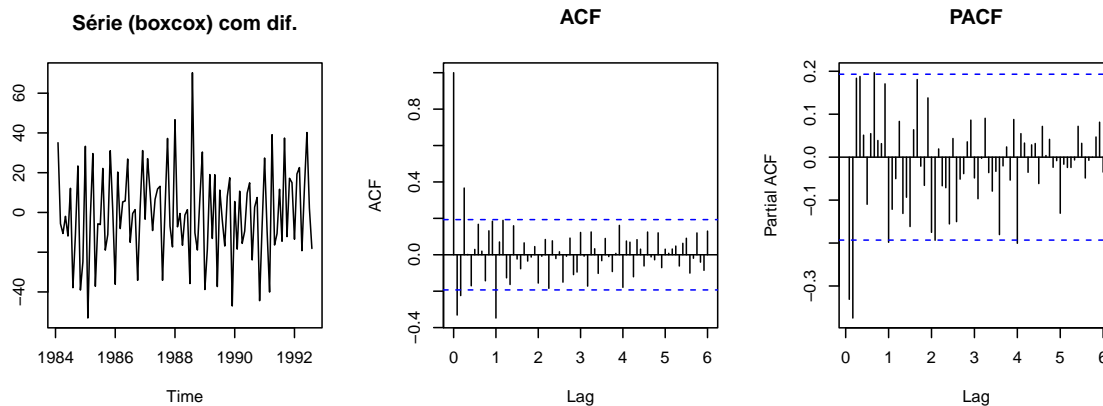
Apesar de haver uma sugestão de transformação, não é possível avaliar graficamente se houve uma diferença significativa no comportamento da série temporal excetuando-se a escala, como se pode ver nos eixos dos gráficos a seguir.



Após aplicar a transformação de Box-Cox na série, utilizou-se as funções `ndiffs()` e `nsdiffs()` para identificar quantas diferenciações simples e sazonais seriam necessárias para que a série se torne estacionária. Concluiu-se que são necessárias uma diferenciações simples e uma diferenciações sazonal, o que é confirmado pelo resultado do teste KPSS nos resíduos da série com as diferenças já aplicadas.

	Estatística	p-valor
KPSS Test for Level Stationarity	0.12	0.1

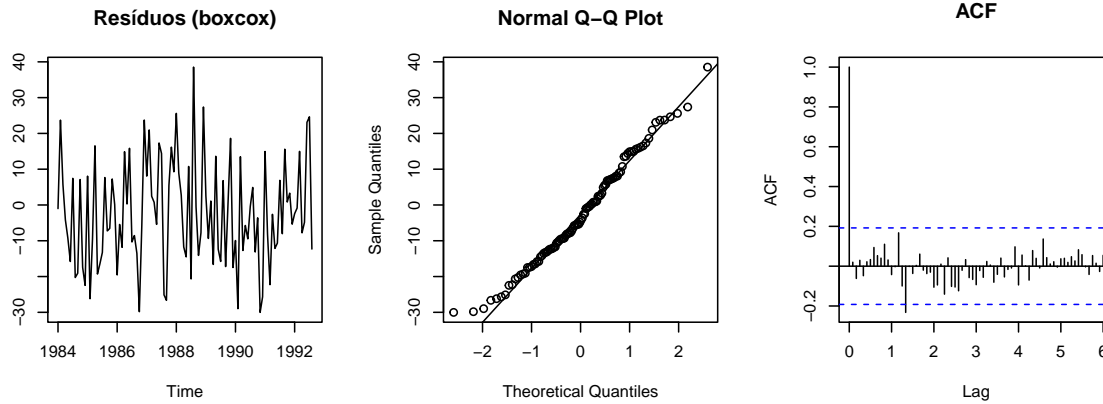
O gráfico da ACF parece apresentar quebra no primeiro lag sazonal, enquanto o PACF tem quebra no segundo lag simples, o que configura um $ARIMA(2, 1, 0) \times (0, 1, 1)_{12}$ (porém, mais uma vez, os resíduos para este modelo não ficam muito bons). Entretanto, os gráficos não evidenciam comportamentos claros para a série. Novamente, os resíduos parecem ter média igual a zero.



Foram testadas combinações de p , q , P e Q , com d e D fixados em 1 e, em seguida, selecionou-se o modelo ARIMA que apresentava menor valor do AICc. Temos, então, que o modelo escolhido para a série transformada é um $ARIMA(2, 1, 2) \times (0, 1, 2)_{12}$, assim como no caso da série sem transformação. Utilizando-se a função `auto.arima()` recebe-se uma sugestão de um modelo $ARIMA(3, 1, 1) \times (2, 1, 0)_{12}$ mas, assim como ocorre no modelo sem transformação, opta-se pelo modelo selecionado manualmente por apresentar um AICc menor.

Resíduos

Foram retirados os zeros da inicialização para seguir com a análise dos resíduos. O gráfico da série dos resíduos sugere aleatoriedade e o QQ plot distribuição aproximadamente normal. Por último, o gráfico ACF mostra que a autocorrelação dos resíduos está dentro da banda de confiança, com exceção de um ponto que excede um pouco este limite.



Assim como ocorre para a série não transformada, os testes de Shapiro-Wilk e Ljung-box com lag igual a 15 não apresentam indicação para rejeição de suas hipóteses nulas. Isto é, pode-se dizer que a série transformada tem distribuição normal e seus resíduos são independentes.

	Estatística	p-valor	Lag
Box-Ljung test	8.41	0.91	15
Shapiro-Wilk normality test	0.98	0.23	

Modelos ETS: seleção, transformações e resíduos

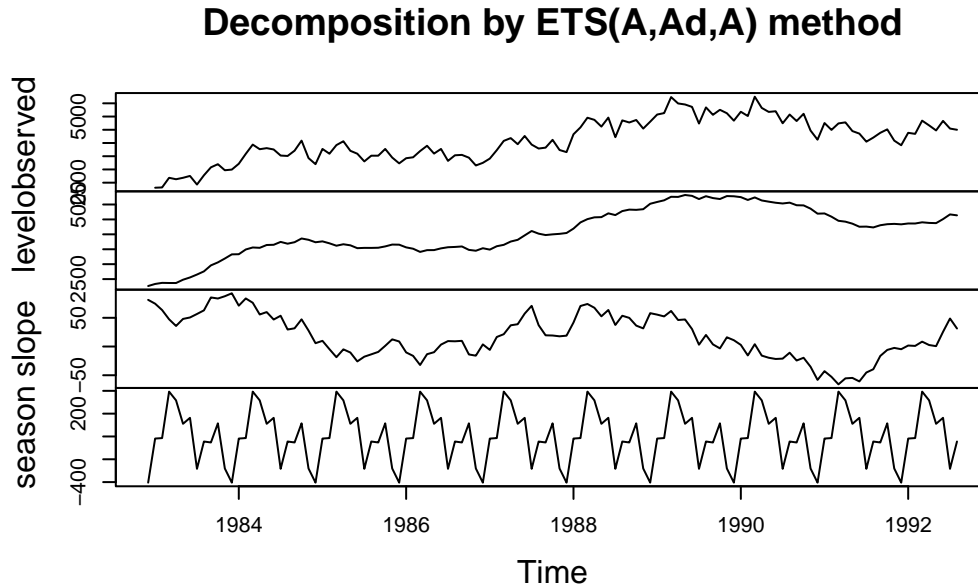
Modelo sem transformação

Seleção

Para a seleção do modelo ETS, foi realizada uma varredura com todas as combinações possíveis de erro, tendência e sazonalidade, assim como a aplicação ou não de *damp* na tendência. Os seis modelos com melhores indicadores são exibidos na tabela a seguir para comparação.

Modelo	AIC	AICc	BIC
ETS(M,Ad,M)	1761.30	1768.36	1810.87
ETS(M,M,M)	1761.94	1769.00	1811.51
ETS(A,Ad,A)	1764.25	1771.30	1813.81
ETS(M,Ad,A)	1767.73	1774.78	1817.29
ETS(M,A,M)	1769.04	1775.29	1815.86
ETS(A,A,A)	1771.20	1777.44	1818.01

De fato, modelos que misturam termos aditivos e multiplicativos apresentam os melhores indicadores AICc, mas o desempenho da função `ets()` é instável nessas condições. Por isso, opta-se pelo uso do terceiro melhor modelo, $ETS(A, Ad, A)$. A decomposição da série temporal analisada utilizando este modelo é exposta no gráfico a seguir.

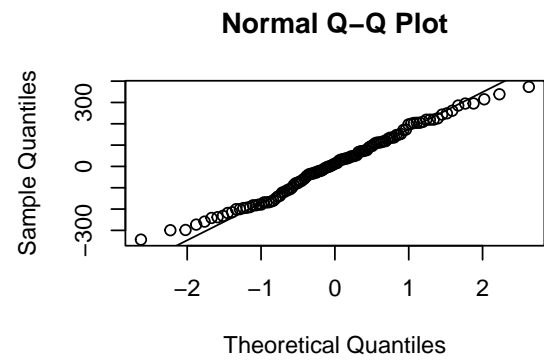
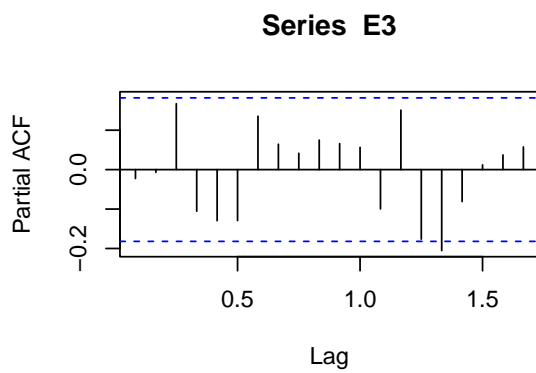
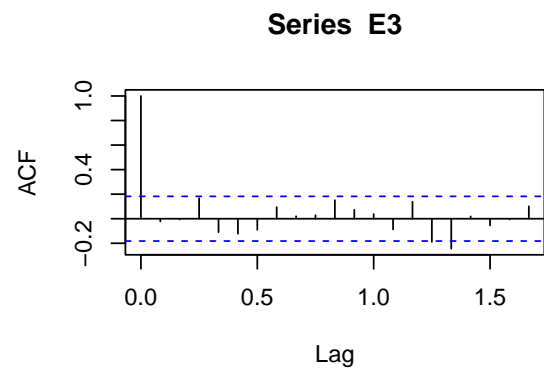
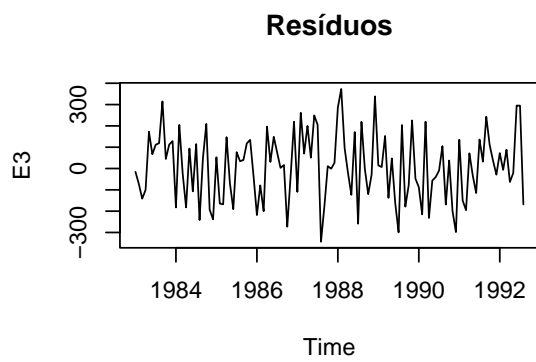


Resíduos

Uma análise visual dos resíduos indica comportamento aleatório em torno de zero e autocorrelações próximas a zero. Quanto à distribuição, a amostra parece ter uma distribuição próxima à normal, mas com caudas mais pesadas.

Observa-se ainda pelo resultado dos testes de hipótese a seguir temos uma série estacionária, com resíduos normalmente distribuídos e mutuamente independentes.

	Estatística	p-valor	Lag
KPSS Test for Level Stationarity	0.07	0.10	4
Box-Ljung test	21.35	0.13	15
Shapiro-Wilk normality test	0.99	0.38	

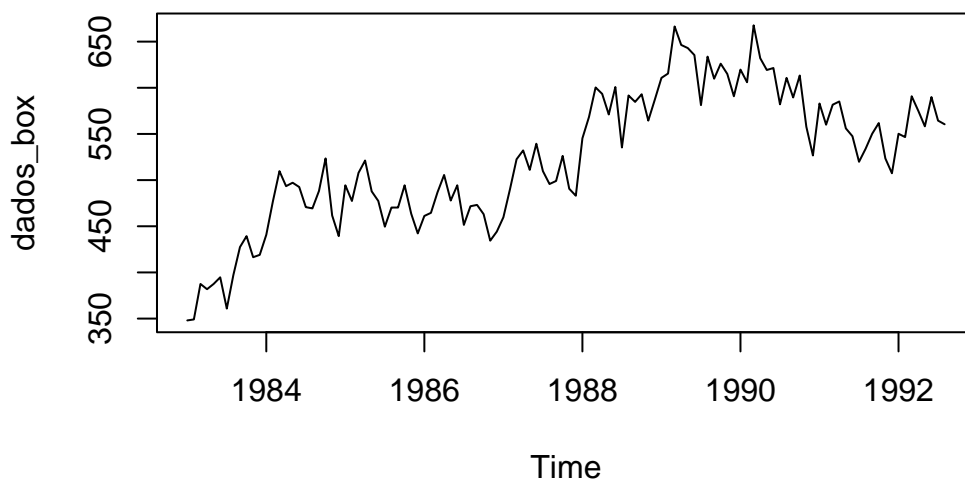


Modelo com transformação

Seleção

Utilizando a função `BoxCox.lambda()` obtém-se uma sugestão de transformação com $\lambda = 0.712$. A série transformada é exibida no gráfico a seguir.

Série com transformação Box–Cox $\lambda = 0.712$

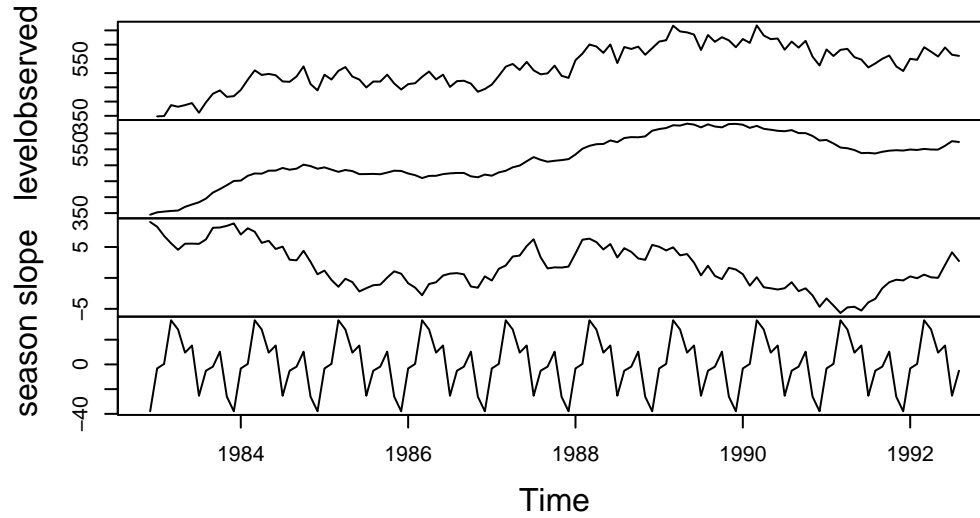


No caso do modelo com transformação BoxCox a recomendação de modelo, obtida pelo mesmo método de varredura, é primariamente a seleção já feita para o modelo sem transformação, que é o $ETS(A, Ad, A)$.

Modelo transformado	AIC	AICc	BIC
ETS(A,Ad,A)	1205.77	1212.82	1255.33
ETS(M,M,M)	1207.69	1214.75	1257.26
ETS(M,Ad,M)	1207.92	1214.98	1257.49
ETS(M,Ad,A)	1208.24	1215.29	1257.81
ETS(A,A,A)	1218.37	1224.62	1265.18
ETS(M,A,A)	1221.45	1227.69	1268.26

Os componentes do ajuste do modelo $ETS(A, Ad, A)$ são expostas a seguir e, como no caso anterior, a transformação não apresenta nenhuma mudança aparente no comportamento da série.

Decomposition by ETS(A,Ad,A) method



Resíduos

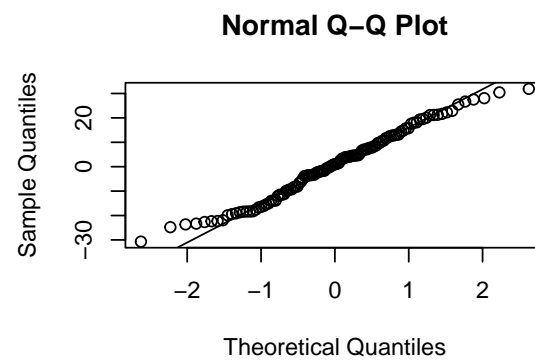
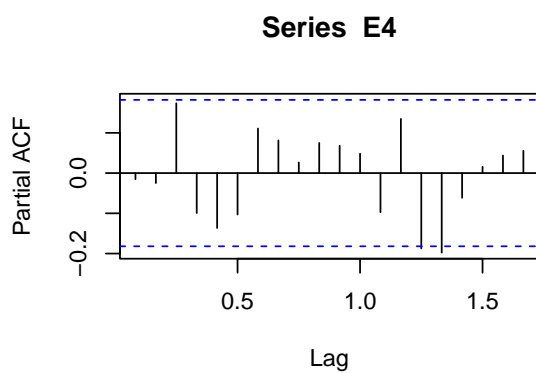
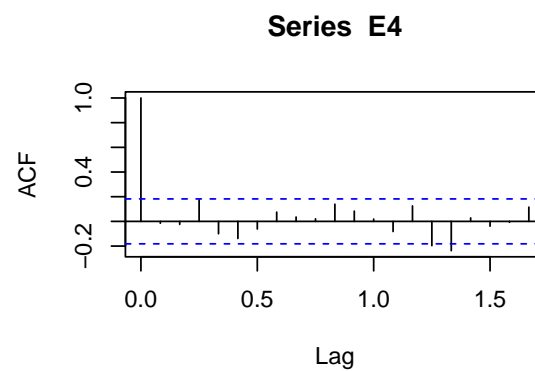
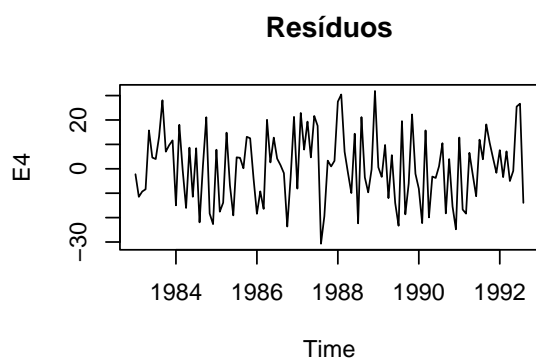
Assim como ocorre para a série não transformada, os resíduos parecem exibir comportamento aleatório em torno de zero, autocorrelações próximas a zero e distribuição próxima à normal, mas com caudas mais pesadas.

Novamente, os testes de hipótese expostos na tabela a seguir corroboram a hipótese de se tratar de uma série estacionária, com resíduos normalmente distribuídos e mutuamente independentes.

	Estatística	p-valor	Lag
KPSS Test for Level Stationarity	0.07	0.10	4
Box-Ljung test	20.48	0.15	15
Shapiro-Wilk normality test	0.98	0.16	

Estudo de desempenho preditivo

Para realizar a análise do desempenho preditivo usando uma abordagem de janela deslizante, o estudo considera uma janela de tamanho $n - 14$ e calcula os erros de previsão para horizontes de até 5 períodos. Utilizando os modelos previamente mencionados para criar as funções de

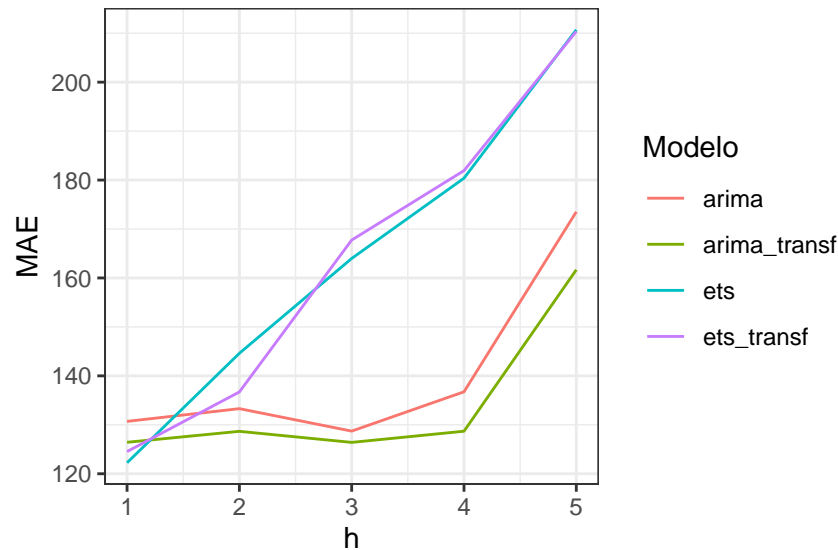


previsão, os resultados são apresentados em um gráfico e uma tabela, mostrando os erros absolutos para cada horizonte de previsão.

Resultados da Janela Deslizante

	ARIMA	ETS	ARIMA Transformada	ETS Transformada
h=1	130.701	126.428	122.244	124.525
h=2	133.301	128.668	144.609	136.707
h=3	128.709	126.408	163.981	167.741
h=4	136.742	128.690	180.380	181.928
h=5	173.517	161.675	210.707	210.361

Performance em relação aos horizontes de previsão



Ao analisar o gráfico obtido, observa-se que o modelo ARIMA, tanto para a série original quanto para a série transformada, apresentou erros médios menores na maioria dos horizontes de previsão, com exceção do horizonte 1. Portanto, o modelo $ARIMA(2, 1, 2) \times (2, 1, 0)_{12}$ obteve melhor comportamento para o caso original e o caso transformado.

Gráficos da previsão pontual e da previsão intervalar dos 4 modelos selecionados

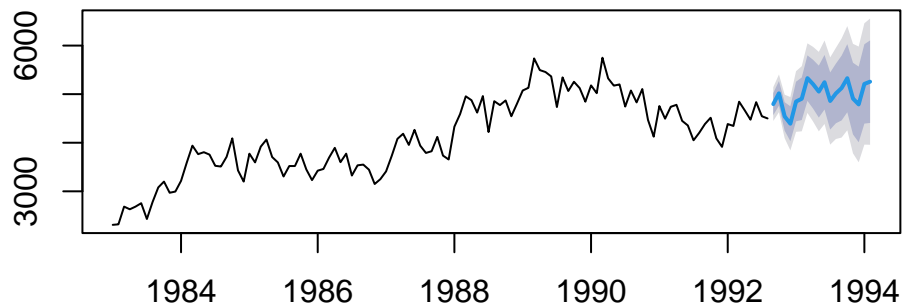
Utilizou-se o modelo $ARIMA(2, 1, 2) \times (2, 1, 0)_{12}$ para realizar previsões pontuais e intervalares dos modelos selecionados. O horizonte de previsão fornecido pelo banco de dados foi de 18 pontos.

Os gráficos fornecidos permitem visualizar as previsões pontuais e intervalares da série temporal da competição de previsão M3, com uma probabilidade de cobertura de 95%. Já as tabelas, apresentam as previsões para um horizonte de 18 pontos e fornecem os intervalos das probabilidades de cobertura 80% e 95%.

Ao analisar os gráficos e as tabelas nas seções a seguir, observa-se que em todos eles a previsão está aparentemente ajustada à série original e os intervalos de probabilidades de cobertura analisados possuem aproximadamente o mesmo espectro.

ARIMA

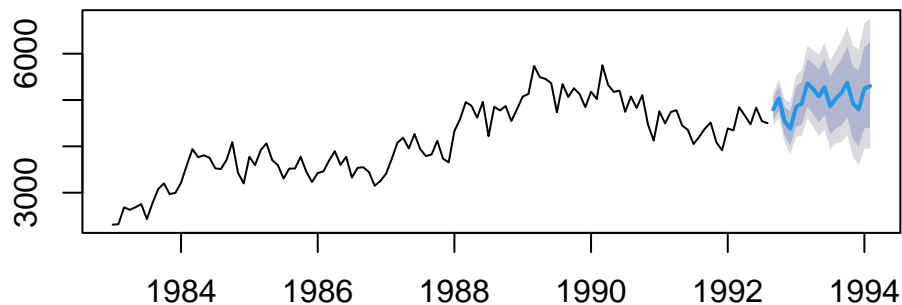
Previsão 95% de probabilidade de cobertura



Período (mês/ano)	Prev. Pontual	IC para 80%		IC para 95%	
		LI	LS	LI	LS
Sep 1992	4795.36	4568.19	5022.53	4447.93	5142.79
Oct 1992	5016.38	4763.01	5269.75	4628.88	5403.88
Nov 1992	4541.28	4248.64	4833.91	4093.73	4988.82
Dec 1992	4391.44	4032.32	4750.56	3842.21	4940.67
Jan 1993	4848.16	4440.26	5256.06	4224.33	5471.99
Feb 1993	4905.30	4467.19	5343.41	4235.26	5575.33
Mar 1993	5329.73	4861.35	5798.10	4613.41	6046.05
Apr 1993	5203.39	4700.48	5706.30	4434.25	5972.52
May 1993	5052.67	4518.13	5587.20	4235.17	5870.16
Jun 1993	5245.42	4683.93	5806.91	4386.70	6104.14
Jul 1993	4859.08	4271.73	5446.43	3960.81	5757.35
Aug 1993	5014.81	4401.33	5628.30	4076.57	5953.06
Sep 1993	5125.55	4459.12	5791.98	4106.34	6144.77
Oct 1993	5328.57	4627.99	6029.14	4257.13	6400.00
Nov 1993	4908.79	4172.04	5645.55	3782.02	6035.57
Dec 1993	4788.71	4010.22	5567.20	3598.12	5979.30
Jan 1994	5210.32	4393.63	6027.01	3961.30	6459.34
Feb 1994	5256.72	4407.77	6105.66	3958.37	6555.07

ARIMA com transformação

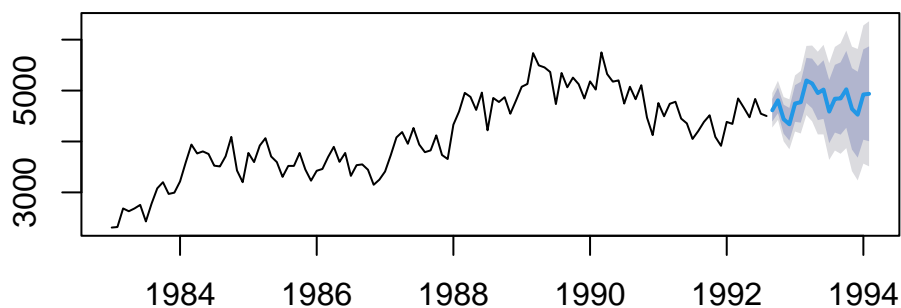
Previsão 95% de probabilidade de cobertura



Período (mês/ano)	Prev. Pontual	IC para 80%		IC para 95%	
		LI	LS	LI	LS
Sep 1992	4799.75	4565.17	5037.69	4442.37	5164.97
Oct 1992	5035.24	4767.61	5307.02	4627.65	5452.55
Nov 1992	4537.15	4237.28	4842.83	4080.96	5006.95
Dec 1992	4379.43	4016.24	4751.52	3827.68	4951.99
Jan 1993	4854.19	4428.23	5291.21	4207.36	5526.88
Feb 1993	4922.09	4462.59	5394.30	4224.66	5649.25
Mar 1993	5364.43	4861.28	5881.57	4600.79	6160.80
Apr 1993	5235.77	4700.00	5787.82	4423.23	6086.42
May 1993	5077.22	4512.97	5660.16	4222.16	5976.02
Jun 1993	5275.45	4676.15	5895.06	4367.47	6230.95
Jul 1993	4867.74	4256.48	5501.96	3942.66	5846.60
Aug 1993	5035.62	4391.14	5704.81	4060.48	6068.61
Sep 1993	5155.40	4452.82	5886.74	4093.16	6284.98
Oct 1993	5372.07	4624.76	6150.63	4242.50	6574.82
Nov 1993	4923.69	4159.71	5723.50	3770.67	6160.63
Dec 1993	4797.72	3998.70	5637.11	3593.16	6096.87
Jan 1994	5246.60	4386.32	6149.64	3949.35	6644.03
Feb 1994	5302.61	4406.47	6244.74	3951.96	6761.01

ETS

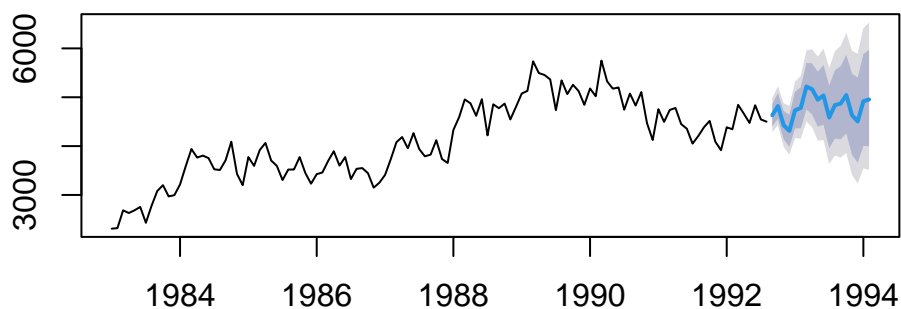
Previsão 95% de probabilidade de cobertura



Período (mês/ano)	Prev. Pontual	IC para 80%		IC para 95%	
		LI	LS	LI	LS
Sep 1992	4613.02	4391.65	4834.40	4274.46	4951.59
Oct 1992	4808.08	4557.39	5058.77	4424.68	5191.48
Nov 1992	4437.83	4153.58	4722.07	4003.11	4872.54
Dec 1992	4337.83	4016.80	4658.86	3846.85	4828.80
Jan 1993	4746.47	4386.22	5106.72	4195.51	5297.43
Feb 1993	4771.43	4370.15	5172.71	4157.73	5385.14
Mar 1993	5198.16	4754.52	5641.80	4519.66	5876.65
Apr 1993	5139.63	4652.66	5626.60	4394.88	5884.39
May 1993	4950.91	4419.94	5481.88	4138.87	5762.96
Jun 1993	5019.38	4443.96	5594.80	4139.35	5899.41
Jul 1993	4587.10	3966.96	5207.23	3638.68	5535.51
Aug 1993	4840.20	4175.23	5505.18	3823.21	5857.20
Sep 1993	4844.61	4134.77	5554.46	3759.00	5930.23
Oct 1993	5024.39	4269.79	5778.99	3870.32	6178.45
Nov 1993	4639.86	3840.68	5439.05	3417.61	5862.11
Dec 1993	4526.53	3682.99	5370.07	3236.45	5816.61
Jan 1994	4922.72	4035.12	5810.33	3565.25	6280.20
Feb 1994	4936.06	4004.71	5867.40	3511.69	6360.43

ETS com transformação

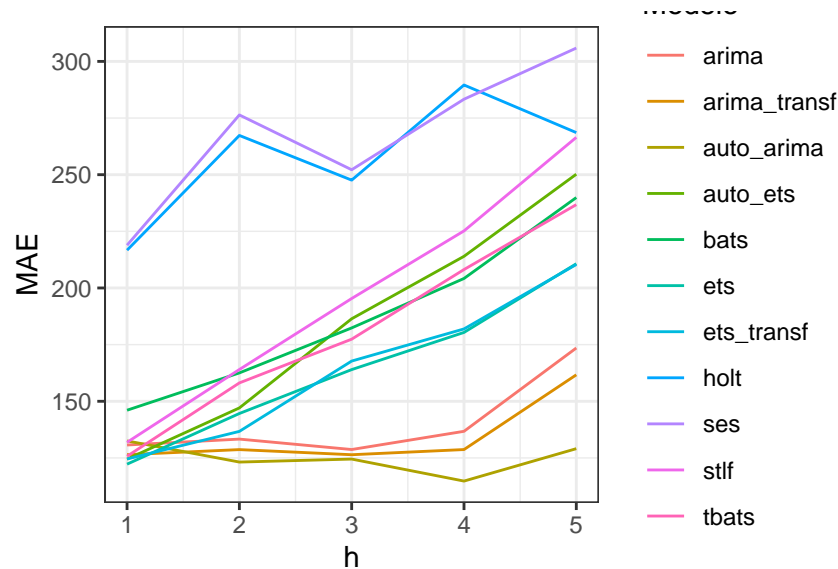
Previsão 95% de probabilidade de cobertura



Período (mês/ano)	Prev. Pontual	IC para 80%		IC para 95%	
		LI	LS	LI	LS
Sep 1992	4631.45	4406.50	4859.59	4288.73	4981.63
Oct 1992	4819.80	4560.49	5083.19	4424.91	5224.24
Nov 1992	4426.13	4138.59	4719.16	3988.65	4876.45
Dec 1992	4311.70	3989.21	4641.30	3821.45	4818.58
Jan 1993	4732.40	4360.53	5112.88	4167.27	5317.69
Feb 1993	4780.20	4365.14	5205.91	4149.88	5435.45
Mar 1993	5219.17	4748.72	5702.17	4504.93	5962.77
Apr 1993	5164.25	4650.16	5693.52	4384.40	5979.64
May 1993	4949.95	4397.25	5521.04	4112.43	5830.51
Jun 1993	5035.98	4434.72	5658.67	4125.51	5996.63
Jul 1993	4584.53	3955.55	5239.45	3633.64	5596.14
Aug 1993	4839.70	4154.97	5553.57	3804.96	5942.70
Sep 1993	4868.35	4137.02	5632.83	3764.09	6050.25
Oct 1993	5044.94	4259.92	5866.89	3860.23	6316.15
Nov 1993	4632.66	3823.65	5484.66	3414.00	5952.05
Dec 1993	4504.35	3658.79	5398.42	3232.32	5890.09
Jan 1994	4918.29	4005.02	5883.36	3544.10	6413.88
Feb 1994	4955.39	3995.84	5971.87	3512.73	6531.48

Resultados

Foi utilizada a métrica do erro absoluto médio (MAE) para comparar o desempenho preditivo dos modelos selecionados com alguns *benchmarks* da biblioteca *forecast*. Estes valores estão apresentados no gráfico abaixo e evidenciam uma melhor performance do modelo gerado pela função automática *auto.arima()*, seguido pelos modelos ARIMA selecionados manualmente.



Conclusão

Neste estudo de caso, a performance dos modelos ARIMA é superior aos demais modelos selecionados manualmente em todos os horizontes de previsão. Ainda, como é possível observar no gráfico de comparação com os *benchmarks*, as funções do grupo arima parecem se comportar como um grupo separado com desempenho melhor que as demais, pelo menos até um horizonte de previsão $h = 5$. Para esta série, portanto, indica-se um modelo ARIMA obtido pela função *auto.arima()*.

Apêndice

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

```

if(!("pacman" %in% installed.packages())){install.packages("pacman")}

pacman::p_load(Mcomp, tidyverse, forecast, fpp2, xts, tseries, tidymodels, kableExtra)


data(M3) #carrega os dados
id <- 2183 #série temporal escolhida

serie <- M3[[id]]

dados <- serie$x

plot(serie, main = "Série Temporal M3-2183")


serie_ms <- forecast::msts(dados, seasonal.periods = c(12))

# media dos resíduos é em torno de 5. Considerando a magnitude dos dados que temos usando
#tentei períodos secundários, porém o melhor ajuste é com apenas um ciclo anual. Só é poss

decomp_mstl <- mstl(serie_ms, lambda = NULL, t.window = 9)

#ajustar com lambda = "auto" até agora não parece fazer qualquer diferença


decomp_mstl %>% autoplot(main = "Decomposição MSTL com período anual simples") + labs(x =

# diferenciacoess comuns
# ndiffs(serie_ms)

#diferenciacoess sazonais
# serie_ms %>% diff() %>% nsdiffs()

serie_ms_diff <- serie_ms %>% diff() %>% diff(lag = 12)

# Inicializacao dos resíduos

```

```

fit <- Arima(dados, order=c(2,1,2), seasonal=c(0,1,2)) #modelo selecionado em outro estágio
E <- window(fit$residuals, start=c(1984,1))

kpss.test(E) %>%
  tidy()%>%
  select(method, statistic, `p.value`) %>%
  knitr::kable(
    format = "latex",
    align = c("lccc"),
    booktabs = TRUE,
    longtable = TRUE,
    linesep = "",
    escape = FALSE,
    digits = 2,
    col.names = c("", "Estatística", "p-valor")
  ) %>%
  kableExtra::kable_styling(
    position = "center",
    latex_options = c("striped", "repeat_header"),
    stripe_color = "gray!15")

# Graficos de Autocorrelacao
par(mfrow=c(1,3))
plot(serie_ms_diff, main="Série com diferenças",ylab="")
acf(serie_ms_diff, lag=12*6, main="ACF")
pacf(serie_ms_diff, lag=12*6, main="PACF")

melhor_AICc <- Inf
for(p in 0:2){
  for(q in 0:2){
    for(P in 0:2){
      for(Q in 0:2){

        #cat("p =",p," q =",q," P =",P," Q =",Q,"\n")

        tryCatch({fit <- Arima(serie_ms, order=c(p,1,q), seasonal=c(P,1,Q))}, error=function(e){})

        if(fit$aicc <= melhor_AICc){
          melhor_AICc <- fit$aicc
          #cat("p =",p," q =",q," P =",P," Q =",Q," AICc =",fit$aicc,"\n")
        }
      }
    }
  }
}

```

```

    }

  }
}

}

#melhor_AICc

# teste com auto.arima para verificar
#auto.arima(serie_ms)

# Analise de residuos
par(mfrow=c(1,3))
plot(E, main="Resíduos",ylab="")
qqnorm(E)
qqline(E)
acf(E,lag.max=12*6,main="ACF")

#Testes
box <- Box.test(E,lag=15,type="Ljung-Box") %>% tidy() %>% select(method, everything()) %>%
shap <- shapiro.test(E) %>% tidy() %>% mutate(parameter = "") %>% select(method, everything())

bind_rows(box, shap) %>%
  knitr::kable(
    format = "latex",
    align = c("lccc"),
    booktabs = TRUE,
    longtable = TRUE,
    linesep = "",
    escape = FALSE,
    digits = 2,
    col.names = c("", "Estatística", "p-valor", "Lag")
  ) %>%
  kableExtra::kable_styling(
    position = "center",
    latex_options = c("striped", "repeat_header"),
    stripe_color = "gray!15")

```

```

rm(box, shap)

# Transformacao de Box-Cox
autolambda <- BoxCox.lambda(dados)
dadosbc <- BoxCox(dados, lambda=autolambda)

par(mfrow=c(1,2))
plot(dados, main='Série original',ylab='')
plot(dadosbc, main='Série transformada',ylab='')

# Diferencas (BC)
# ndiffs(dadosbc)
# dadosbc %>% diff() %>% nsdiffs()

dadosbcdiff <- dadosbc %>% diff() %>% diff(lag = 12)

fit2 <- Arima(dadosbc, order=c(2,1,2), seasonal=c(0,1,2))# modelo ajustado em etapa poster

E2 <- window(fit2$residuals, start=c(1984,1))

kpss.test(E2) %>%
  tidy()%>%
  select(method, statistic, `p.value`) %>%
  knitr::kable(
    format = "latex",
    align = c("lccc"),
    booktabs = TRUE,
    longtable = TRUE,
    linesep = "",
    escape = FALSE,
    digits = 2,
    col.names = c("", "Estatística", "p-valor")
  ) %>%
  kableExtra::kable_styling(
    position = "center",
    latex_options = c("striped", "repeat_header"),
    stripe_color = "gray!15")

```

```

# Graficos de Autocorrelacao (BC)
par(mfrow=c(1,3))
plot(dadosbcdiff, main="Série (boxcox) com dif.",ylab="")
acf(dadosbcdiff, lag=12*6, main="ACF")
pacf(dadosbcdiff, lag=12*6, main="PACF")

# Criterio de Akaike (BC)
melhor_AICc <- Inf
for(p in 0:2){
  for(q in 0:2){
    for(P in 0:2){
      for(Q in 0:2){
        #cat("p =",p," q =",q," P =",P," Q =",Q,"\n")
        tryCatch({fit <- Arima(dadosbc, order=c(p,1,q), seasonal=c(P,1,Q))}, error=function(e){})
        if(fit$aicc <= melhor_AICc){
          melhor_AICc <- fit$aicc
          #cat("p =",p," q =",q," P =",P," Q =",Q," AICc =",fit$aicc,"\n")}
      }
    }
  }
}
# melhor_AICc

# auto.arima(serie_ms)

# Inicializacao
# Analise de residuos
par(mfrow=c(1,3))
plot(E2, main="Resíduos (boxcox)",ylab="")
qqnorm(E2)
qqline(E2)
acf(E,lag.max=12*6,main="ACF")

#Testes
box <- Box.test(E2,lag=15,type="Ljung-Box") %>% tidy() %>% select(method, everything()) %>%
shap <- shapiro.test(E2) %>% tidy() %>% mutate(parameter = "") %>% select(method, everything())

```



```

bind_rows(box, shap) %>%
  knitr::kable(
    format = "latex",
    align = c("lccc"),
    booktabs = TRUE,
    longtable = TRUE,
    linesep = "",
    escape = FALSE,
    digits = 2,
    col.names = c("", "Estatística", "p-valor", "Lag")
  ) %>%
  kableExtra::kable_styling(
    position = "center",
    latex_options = c("striped", "repeat_header"),
    stripe_color = "gray!15")

rm(box, shap)

# monta as combinações possíveis de modelo ets
model <- expand_grid(v1 = c("A", "M", "N"), v2 = c("A", "M", "N"), v3 = c("A", "M", "N"))
mutate(modelo = str_c(v1,v2,v3)) %>%
  select(modelo) %>%
  unique() %>%
  expand_grid(., damp = c(TRUE, FALSE))
# N, A, M, + damped

#funcao pra montar indicadores do modelo
criterios <- function(modelo, damp, dados) {
  ETS <- ets(dados, model = modelo, damped = damp)
  #usamos o objeto dados como um padrao

  tabela <- tibble(
    nome = modelo,
    sigla = str_c("ETS(", str_c(substr(modelo,1,1), substr(modelo,2,2), substr(modelo,3,3)),
    damped = damp,
    AIC = ETS$aic,
    AICc = ETS$aicc,
    BIC = ETS$bic)

```

```

    return(tabela)
  }

#selecionando modelos permitidos pela funcao ets
# for(i in 1:length(model$modelo)){
#   print(i)
#   print(try({ets(dados, model = model$modelo[i], damped = model$damp[i])}, silent = TRUE))
# }

selecionados <- c(1, 2, 5, 6, 14, 18:24, 27:30, 32, 34, 36)

model_select <- model[selecionados,]

tabela_modelos_ETS <- map2_df(model_select$modelo, model_select$damp, criterios, dados) %>%
  arrange(AIC) %>%
  mutate(modelo = case_when(
    damped == TRUE ~ str_replace(sigla, ",A", ",Ad"),
    .default = sigla
  ))

tabela_modelos_ETS %>%
  select("Modelo"= modelo, AIC:BIC)%>%
  head(6) %>%
  knitr::kable(
    format = "latex",
    align = c("lccc"),
    booktabs = TRUE,
    longtable = TRUE,
    linesep = "",
    escape = FALSE,
    digits = 2
  ) %>%
  kableExtra::kable_styling(
    position = "center",
    latex_options = c("striped", "repeat_header"),
    stripe_color = "gray!15")

# decomposicao ETS sem transformacao ----

```

```

fit_ets <- ets(dados, model = "AAA", damped = TRUE)

plot(fit_ets)

# Análise de resíduos ETS sem transformação
E3 <- fit_ets$residuals
par(mfrow=c(2,2))
plot(E3, main = "Resíduos")
acf(E3)
pacf(E3)
qqnorm(E3)
qqline(E3)

# # Testes para ETS sem transformação
box <- Box.test(E3,lag=15,type="Ljung-Box") %>% tidy() %>% select(method, everything()) %>%
shap <- shapiro.test(E3) %>% tidy() %>% mutate(parameter = "") %>% select(method, everything()) %>%
kpss <- kpss.test(E3) %>% tidy() %>% select(method, everything()) %>% mutate(parameter = a

bind_rows(kpss,box, shap) %>%
  knitr::kable(
    format = "latex",
    align = c("lccc"),
    booktabs = TRUE,
    longtable = TRUE,
    linesep = "",
    escape = FALSE,
    digits = 2,
    col.names = c("", "Estatística", "p-valor", "Lag")
  ) %>%
  kableExtra::kable_styling(
    position = "center",
    latex_options = c("striped", "repeat_header"),
    stripe_color = "gray!15")

rm(box, shap, kpss)

# avalia valor de lambda da transformacao boxcox
lambda <- dados %>% BoxCox.lambda()

```

```

dados_box <- dados %>% BoxCox(lambda)

# visualização e decomp da ETS com transformação
plot(dados_box,main= parse(text = paste0('"Série com transformação Box-Cox  "', '~ lambda

# for(i in 1:length(model$modelo)){
#   print(i)
#   print(try({ets(dados_box, model = model$modelo[i], damped = model$damp[i])}, silent =
# })

# selecionados_transf <- c(1, 2, 5, 6, 14, 18, 19:24, 27:30, 32, 34, 36)

model_select_transf <- model[selecionados,]

tabela_modelos_ETS_transf <- map2_df(model_select_transf$modelo, model_select_transf$damp,
  arrange(AIC) %>%
  mutate(modelo = case_when(
    damped == TRUE ~ str_replace(sigla, ",A", ",Ad"),
    .default = sigla
  ))

tabela_modelos_ETS_transf %>%
  select("Modelo transformado"= modelo, AIC:BIC)%>%
  head(6) %>%
  knitr::kable(
    format = "latex",
    align = c("lccc"),
    booktabs = TRUE,
    longtable = TRUE,
    linesep = "",
    escape = FALSE,
    digits = 2
  ) %>%
  kableExtra::kable_styling(
    position = "center",
    latex_options = c("striped", "repeat_header"),
    stripe_color = "gray!15")

```

```

fit_ets_box <- ets(dados_box, model = "AAA", damped = TRUE)

plot(fit_ets_box)

E4 <- fit_ets_box$residuals
par(mfrow=c(2,2))
plot(E4, main = "Resíduos")
acf(E4)
pacf(E4)
qqnorm(E4)
qqline(E4)

# # Testes para ETS com transformação
box <- Box.test(E4,lag=15,type="Ljung-Box") %>% tidy() %>% select(method, everything()) %>%
shap <- shapiro.test(E4) %>% tidy() %>% mutate(parameter = "") %>% select(method, everything()) %>%
kpss <- kpss.test(E4) %>% tidy() %>% select(method, everything()) %>% mutate(parameter = a

bind_rows(kpss,box, shap) %>%
  knitr::kable(
    format = "latex",
    align = c("lccc"),
    booktabs = TRUE,
    longtable = TRUE,
    linesep = "",
    escape = FALSE,
    digits = 2,
    col.names = c("", "Estatística", "p-valor", "Lag")
  ) %>%
  kableExtra::kable_styling(
    position = "center",
    latex_options = c("striped", "repeat_header"),
    stripe_color = "gray!15")

rm(box, shap, kpss)

# Definindo as funções de previsão

# Arima

```

```

f_arima <- function(y, h){
fit = Arima(y, order=c(2,1,2), seasonal=c(0,1,2))
return(forecast(fit, h))
}

# Arima com transformação
f_arima_transf <- function(y, h){
fit = Arima(y, order=c(2,1,2), seasonal=c(0,1,2), lambda = 0.712)
forecast(fit, h)
}

# ETS
f_ets <- function(y, h){
fit = ets(y, model="AAA", damped = TRUE)
forecast(fit, h)
}

# ETS com transformação
f_ets_transf <- function(y, h){
fit = ets(y, model="AAA", lambda = 0.712, damped = TRUE)
forecast(fit, h)
}

# Tamanho da série
n <- length(dados)

# Cálculo dos erros de previsão para cada método

CV_arima <- tsCV(y = dados, forecastfunction = f_arima, h = 5, initial = n-14)

CV_arima_transf <- tsCV(y = dados, forecastfunction = f_arima_transf,
                        h = 5, initial = n-14)

CV_ets <- tsCV(y = dados, forecastfunction = f_ets, h = 5, initial = n-14)

CV_ets_transf <- tsCV(y = dados, forecastfunction = f_ets_transf,
                      h = 5, initial = n-14)

# Cálculo do erro absoluto médio (MAE) para cada horizonte de previsão ----

```

```

#----- otimizado abaixo -----#
# MAE_arima <- CV_arima %>% abs() %>% colMeans(na.rm=T)
# MAE_arima_transf <- CV_arima_transf %>% abs() %>% colMeans(na.rm=T)
# MAE_ets <- CV_ets %>% abs() %>% colMeans(na.rm=T)
# MAE_ets_transf <- CV_ets_transf %>% abs() %>% colMeans(na.rm=T)
#
# tab <- cbind(as.numeric(MAE_arima), as.numeric(MAE_ets))
# tab_transf <- cbind(MAE_arima_transf, MAE_ets_transf)
#
# tab_erros <- tibble(MAE_arima, MAE_ets, MAE_arima_transf, MAE_ets_transf)
#-----#

tab_erros <- tibble(
  h = paste0("h=", c(1:5)),
  MAE_arima = CV_arima %>% abs() %>% colMeans(na.rm=T),
  MAE_arima_transf = CV_arima_transf %>% abs() %>% colMeans(na.rm=T),
  MAE_ets = CV_ets %>% abs() %>% colMeans(na.rm=T),
  MAE_ets_transf = CV_ets_transf %>% abs() %>% colMeans(na.rm=T),
)

tab_erros %>%
  knitr::kable(
    format = "latex",
    align = c("c"),
    booktabs = TRUE,
    longtable = TRUE,
    linesep = "",
    escape = FALSE,
    col.names = c('', 'ARIMA', 'ETS', 'ARIMA Transformada', 'ETS Transformada'),
    digits = 3
  ) %>%
  kableExtra::kable_styling(
    position = "center",
    latex_options = c("striped", "repeat_header"),
    stripe_color = "gray!15")

tab_erros %>%
  pivot_longer(cols = -h, names_to = "Modelo", values_to = "values") %>%
  mutate(

```

```

    Modelo = str_remove_all(Modelo, "MAE_"),
    h = str_remove_all(h, "h="),
    h = as.integer(h)) %>%
ggplot(aes(h, values, color = Modelo))+
geom_line()+
labs(y = "MAE")+
theme_bw()

# Verificando o h
h <- M3[[id]]$h

# Gráficos de previsão pontual e intervalar

# ARIMA
arima_prev95 <- f_arima(dados, h)

plot(arima_prev95, main = "Previsão 95% de probabilidade de cobertura")

linhas <- c("Sep 1992", "Oct 1992", "Nov 1992", "Dec 1992", "Jan 1993", "Feb 1993", "Mar 1993",

arima_prev95 %>%
  as_tibble() %>%
  mutate(t = linhas)%>%
  select(t, everything()) %>%
  knitr::kable(
    format = "latex",
    align = c("c"),
    booktabs = TRUE,
    longtable = TRUE,
    linesep = "",
    escape = FALSE,
    digits = 2,
    col.names = c("Período (mês/ano)", "Prev. Pontual", "LI", "LS", "LI", "LS")) %>%
  add_header_above(c(" " = 2, "IC para 80%" = 2, "IC para 95%" = 2))

# ARIMA com transformação
arima_transf_prev95 <- f_arima_transf(dados, h)

```



```

plot(arima_transf_prev95, main = "Previsão 95% de probabilidade de cobertura")

arima_transf_prev95 %>%
  as_tibble() %>%
  mutate(t = linhas)%>%
  select(t, everything()) %>%
  knitr::kable(
    format = "latex",
    align = c("c"),
    booktabs = TRUE,
    longtable = TRUE,
    linesep = "",
    escape = FALSE,
    digits = 2,
    col.names = c("Período (mês/ano)", "Prev. Pontual", "LI", "LS", "LI", "LS")) %>%
  add_header_above(c(" " = 2, "IC para 80%" = 2, "IC para 95%" = 2))

# ETS
ets_prev95 <- f_ets(dados, h)

plot(ets_prev95, main = "Previsão 95% de probabilidade de cobertura")

ets_prev95 %>%
  as_tibble() %>%
  mutate(t = linhas)%>%
  select(t, everything()) %>%
  knitr::kable(
    format = "latex",
    align = c("c"),
    booktabs = TRUE,
    longtable = TRUE,
    linesep = "",
    escape = FALSE,
    digits = 2,
    col.names = c("Período (mês/ano)", "Prev. Pontual", "LI", "LS", "LI", "LS")) %>%
  add_header_above(c(" " = 2, "IC para 80%" = 2, "IC para 95%" = 2))

# ETS com transformação
ets_transf_prev95 <- f_ets_transf(dados, h)

```

```

plot(ets_transf_prev95, main = "Previsão 95% de probabilidade de cobertura")

ets_transf_prev95 %>%
  as_tibble() %>%
  mutate(t = linhas)%>%
  select(t, everything()) %>%
  knitr::kable(
    format = "latex",
    align = c("c"),
    booktabs = TRUE,
    longtable = TRUE,
    linesep = "",
    escape = FALSE,
    digits = 2,
    col.names = c("Período (mês/ano)", "Prev. Pontual", "LI", "LS", "LI", "LS")) %>%
    add_header_above(c(" " = 2, "IC para 80%" = 2, "IC para 95%" = 2))

## MAE (Benchmarks)

n <- length(dados)

# auto.arima
f_auto_arima <- function(y, h){
  fit = auto.arima(y)
  forecast(fit, h)
}

CV_auto_arima <- tsCV(y=dados,forecastfunction=f_auto_arima,h=5,initial=n-14)

# ses
f_ses <- function(y, h){
  ses(y,h)
}
CV_ses <- tsCV(y=dados,forecastfunction=f_ses,h=5,initial=n-14)

# holt
f_holt <- function(y, h){
  holt(y,h)
}
CV_holt <- tsCV(y=dados,forecastfunction=f_holt,h=5,initial=n-14)

```

```

# ets
f_auto_ets <- function(y, h){
  fit = ets(y)
  forecast(fit, h)
}
CV_auto_ets <- tsCV(y=dados,forecastfunction=f_auto_ets,h=5,initial=n-14)

# stlf
f_stlf <- function(y, h){
  stlf(y,h)
}
CV_stlf <- tsCV(y=dados,forecastfunction=f_stlf,h=5,initial=n-14)

# bats
f_bats <- function(y, h){
  fit = bats(y)
  forecast(fit, h)
}
CV_bats <- tsCV(y=dados,forecastfunction=f_bats,h=5,initial=n-14)

# tbats
f_tbats <- function(y, h){
  fit = tbats(y)
  forecast(fit, h)
}
CV_tbats <- tsCV(y=dados,forecastfunction=f_tbats,h=5,initial=n-14)

# tabela
tabela <- tibble(
  h = paste0("h=", c(1:5)),
  MAE_arima = CV_arima %>% abs() %>% colMeans(na.rm=T),
  MAE_arima_transf = CV_arima_transf %>% abs() %>% colMeans(na.rm=T),
  MAE_ets = CV_ets %>% abs() %>% colMeans(na.rm=T),
  MAE_ets_transf = CV_ets_transf %>% abs() %>% colMeans(na.rm=T),
  MAE_auto_arima = CV_auto_arima %>% abs() %>% colMeans(na.rm=T),
  MAE_ses = CV_ses %>% abs() %>% colMeans(na.rm=T),
  MAE_holt = CV_holt %>% abs() %>% colMeans(na.rm=T),
  MAE_auto_ets = CV_auto_ets %>% abs() %>% colMeans(na.rm=T),
  MAE_stlf = CV_stlf %>% abs() %>% colMeans(na.rm=T),
  MAE_bats = CV_bats %>% abs() %>% colMeans(na.rm=T),
  MAE_tbats = CV_tbats %>% abs() %>% colMeans(na.rm=T),

```

```

)

# MAE (grafico)
tabela %>%
  pivot_longer(cols = -h, names_to = "Modelo", values_to = "values")%>%
  mutate(
    Modelo = str_remove_all(Modelo, "MAE_"),
    h = str_remove_all(h, "h="),
    h = as.integer(h)) %>%
  ggplot(aes(h, values, color = Modelo))+
  geom_line()+
  labs(y = "MAE")+
  theme_bw()

```