Trabalho Final de Séries Temporais

Carlos Alberto Torres Quintanilha Neto

Turma: MRJ02021-TBABD-8

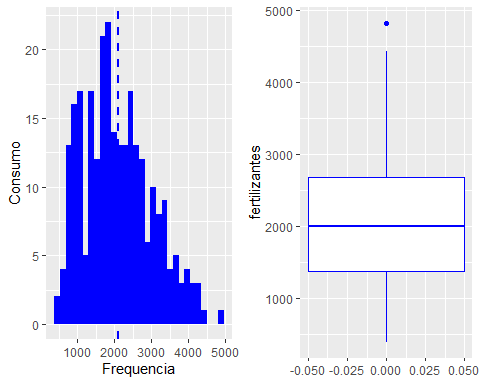
12/12/2020

##### 

# Comentário Inicial - Alálise Exploratória

## Informações sobre a Dataset

b1 <- fertcons %>% ggplot(aes(x=fertilizantes)) +   
 geom\_histogram(bins=30, fill="blue")+  
 xlab("Frequencia")+  
 ylab("Consumo")+  
 geom\_vline(aes(xintercept=mean(fertilizantes)),color="blue", linetype="dashed", size=1)  
  
b2 <- fertcons %>% ggplot(aes(y= fertilizantes)) +   
 geom\_boxplot(width = .1, color = "blue")  
   
grid.arrange(b1,b2,ncol = 2)



any(is.na(fertcons))

## [1] FALSE

pander(summary(fertcons))

|  |
| --- |
| fertilizantes |
| Min. : 396 |
| 1st Qu.:1381 |
| Median :1997 |
| Mean :2097 |
| 3rd Qu.:2688 |
| Max. :4824 |

Pelo gráfico perece que o histograma de frequência e consumo não apresenta uma distribuição normal, pois há um viés a esquerda. O dataset não apresenta “NA” e possui 268 observações. O boxplot destaca um outlier.

Realizando os testes de normalidade verificamos que há divergências entre o teste de Shapiro e o Komolgorov, no primeiro os dados não são considerados normais (P-Value < 0.05) e no segundo são considerados normais (P-Value > 0.05)

# Shapiro-Wilk  
  
shapiro.test(fertcons$fertilizantes)

##   
## Shapiro-Wilk normality test  
##   
## data: fertcons$fertilizantes  
## W = 0.97385, p-value = 7.936e-05

# Kolmogorov-Smirnov  
  
ks.test(fertcons$fertilizantes, mean(fertcons$fertilizantes), sd(fertcons$fertilizantes))

## Warning in ks.test(fertcons$fertilizantes, mean(fertcons$fertilizantes), :  
## cannot compute correct p-values with ties

##   
## Two-sample Kolmogorov-Smirnov test  
##   
## data: fertcons$fertilizantes and mean(fertcons$fertilizantes)  
## D = 0.54104, p-value = 0.9325  
## alternative hypothesis: two-sided

## Criando e analisando a série temporal

Foi criada uma série temporal chamada ts.fertcons seu gráfico foi plotado e decomposto.

É possível identificar no gráfico uma leve tendência de crescimento pois a série inicia em 1998 com valores em torno de 1.000 e o seu final está em torno de 3.000 indicando que a entrega de fertilizantes tem aumentado ao longo dos anos. Em 2008, provavelmente por conta da crise econômica, observa-se um vale no gráfico que contraria um pouco a tendência.

É possível verificar a existência de sazonalidade, cuja amplitude vai aumentando, com exceção de 2003,2004, 2008 e 2009, parecendo indicar um padrão multiplicativo.

# Criando a TS  
  
ts.fertcons <- ts(fertcons, frequency = 12, start = c(1998,1), end = c(2020,4))  
class(ts.fertcons)

## [1] "ts"

# Plot simples da série  
  
frequency(ts.fertcons)

## [1] 12

start(ts.fertcons)

## [1] 1998 1

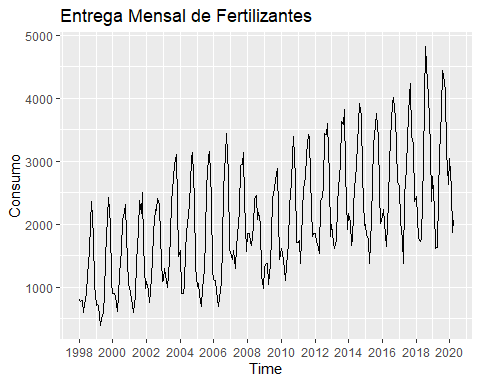
end(ts.fertcons)

## [1] 2020 4

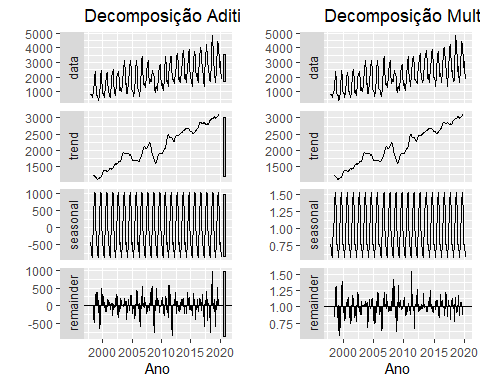
autoplot(ts.fertcons, frequency = 12, start=c(1998,1)) +   
 ylab("Consumo") +  
 scale\_x\_continuous(breaks = scales::extended\_breaks(12))+  
 ggtitle("Entrega Mensal de Fertilizantes")

## Warning: Ignoring unknown parameters: frequency, start

## Scale for 'x' is already present. Adding another scale for 'x', which will  
## replace the existing scale.



d1 <- ts.fertcons%>%  
 decompose(type= "additive")%>%  
 autoplot() + xlab("Ano")+  
 ggtitle("Decomposição Aditiva")  
  
d2 <- ts.fertcons%>%  
 decompose(type= "multiplicative")%>%  
 autoplot() + xlab("Ano")+  
 ggtitle("Decomposição Multiplicativa")  
  
grid.arrange(d1,d2,ncol = 2)

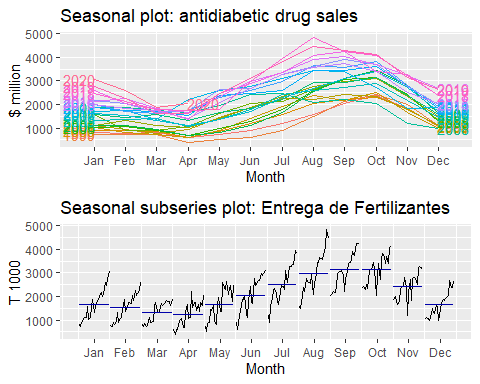


### Sazonalidade

Verifica-se que os meses de agosto, setembro e outubro, são os de maior entrega de fertilizantes e os piores meses parecem ser os de março e abril iniciando um novo ciclo de crescimento em maio, o que parece adequado aos períodos de safra e entressafra.

Parece haver um crescimento da entrega, em valores absolutos, em todos os meses ao longo dos anos embora haja vales em alguns anos indicando possíveis crises econômicas como a de 2008.

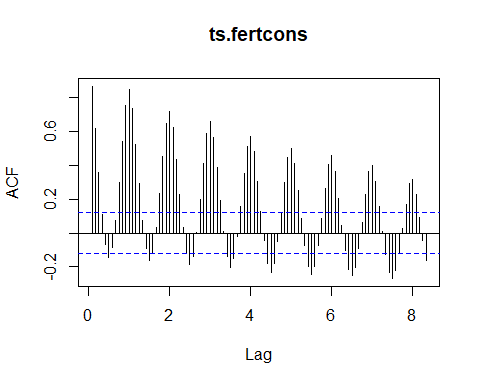
q1 <- ggseasonplot(ts.fertcons, year.labels=TRUE, year.labels.left=TRUE) +  
 ylab("$ million") +  
 ggtitle("Seasonal plot: antidiabetic drug sales")  
  
q2 <- ggsubseriesplot(ts.fertcons) +  
 ylab("T 1000") +  
 ggtitle("Seasonal subseries plot: Entrega de Fertilizantes")  
  
  
grid.arrange(q1,q2, nrow =2, heights=c(40,40))



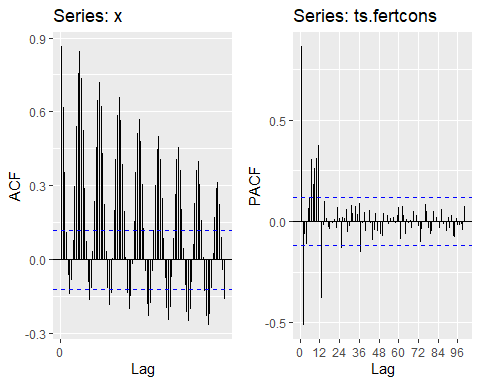
## Autocorrelação

No correlograma da série é possível notar a presença da sazonalidade nos picos múltiplos de 12, como as cavidades tentem a ficar dois quartos atrás dos picos os múltiplos de 6 são os mais negativos. O decaimento muito lento do ACF pode ser sinal de não estacionariedade. Como o decaimento do PACF é menos lento que o do ACF modelos com componentes autorregressivos tem a possibilidade de se adequar melhor.

a1 <- ggAcf(ts.fertcons, lag=100)



a2 <- ggPacf(ts.fertcons, lag=100)  
grid.arrange(a1,a2,ncol = 2)



### Estacionariedade da Série

Nos testes formais temos as seguintes conclusões:

* ADF: H0 - Série não estacionária -> Rejeitada
* KPPS: H0 - Série estacionária -> Rejeitada
* PP: H0 - Série não estacionária -> Rejeitada

A análise gráfica e o teste KPPS indicam que a série não é estacionária, porém os testes ADF e PP indicam que a série é estacionaria.

adf.test(ts.fertcons)

## Warning in adf.test(ts.fertcons): p-value smaller than printed p-value

##   
## Augmented Dickey-Fuller Test  
##   
## data: ts.fertcons  
## Dickey-Fuller = -10.174, Lag order = 6, p-value = 0.01  
## alternative hypothesis: stationary

kpss.test(ts.fertcons)

## Warning in kpss.test(ts.fertcons): p-value smaller than printed p-value

##   
## KPSS Test for Level Stationarity  
##   
## data: ts.fertcons  
## KPSS Level = 2.644, Truncation lag parameter = 5, p-value = 0.01

pp.test(ts.fertcons)

## Warning in pp.test(ts.fertcons): p-value smaller than printed p-value

##   
## Phillips-Perron Unit Root Test  
##   
## data: ts.fertcons  
## Dickey-Fuller Z(alpha) = -81.213, Truncation lag parameter = 5, p-value  
## = 0.01  
## alternative hypothesis: stationary

Ao tentar obter convergência nos resultados dos testes através da diferenciação da série observamos que com uma diferenciação é possível manter os resultados do ADF e do PP e inverter o resultado do kpss obtendo indicação de estacionariedade nos 3 testes.

#Utililzando ndiff para saber o número de diferenciações necessárias  
ndiffs(ts.fertcons)

## [1] 1

# Diferenciando a série  
ts.diff\_fertcons <- diff(ts.fertcons)  
  
# Testando  
  
adf.test(ts.diff\_fertcons)

## Warning in adf.test(ts.diff\_fertcons): p-value smaller than printed p-value

##   
## Augmented Dickey-Fuller Test  
##   
## data: ts.diff\_fertcons  
## Dickey-Fuller = -12.85, Lag order = 6, p-value = 0.01  
## alternative hypothesis: stationary

kpss.test(ts.diff\_fertcons)

## Warning in kpss.test(ts.diff\_fertcons): p-value greater than printed p-value

##   
## KPSS Test for Level Stationarity  
##   
## data: ts.diff\_fertcons  
## KPSS Level = 0.011981, Truncation lag parameter = 5, p-value = 0.1

pp.test(ts.diff\_fertcons)

## Warning in pp.test(ts.diff\_fertcons): p-value smaller than printed p-value

##   
## Phillips-Perron Unit Root Test  
##   
## data: ts.diff\_fertcons  
## Dickey-Fuller Z(alpha) = -135.18, Truncation lag parameter = 5, p-value  
## = 0.01  
## alternative hypothesis: stationary

# Modelos de Estudo

Inicialmente vamos montar a base de treino e a base de teste que servirão para todo o exercício.

train <- window(ts.fertcons, start = c(2007,1), end = c(2018,12))  
test <-window(ts.fertcons, start = c(2019,1),end = c(2020,4))

## Modelos por Exponential Smoothing

Vamos utilizar 3 modelos e tentar comparar a capacidade de forecasting dos mesmos, a saber:

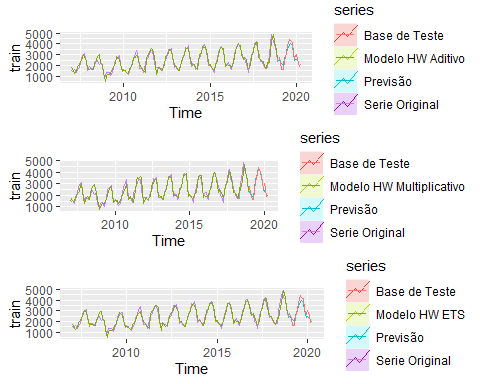
* Holt Winters Aditivo
* Holt Winters Multiplicativo
* ETS

No caso do ETS vamos verificar se ele indica outra possibilidade ou se a indicação bate com HWA ou HWM.

# Aditivo  
  
  
fit\_holt\_ad <- hw(train,h=14, seasonal = "additive" ,level=95)  
summary(fit\_holt\_ad)

##   
## Forecast method: Holt-Winters' additive method  
##   
## Model Information:  
## Holt-Winters' additive method   
##   
## Call:  
## hw(y = train, h = 14, seasonal = "additive", level = 95)   
##   
## Smoothing parameters:  
## alpha = 0.856   
## beta = 1e-04   
## gamma = 1e-04   
##   
## Initial states:  
## l = 2312.1115   
## b = 6.7753   
## s = -534.0436 139.6893 877.0535 1017.621 955.1693 484.0101  
## 97.7664 -322.4482 -874.8917 -788.5711 -571.2841 -480.0711  
##   
## sigma: 283.4815  
##   
## AIC AICc BIC   
## 2359.071 2363.928 2409.557   
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set -2.831316 267.2689 206.2041 -0.7949484 9.330779 0.6332786  
## ACF1  
## Training set 0.02211294  
##   
## Forecasts:  
## Point Forecast Lo 95 Hi 95  
## Jan 2019 2458.947 1903.3331 3014.560  
## Feb 2019 2374.347 1642.9242 3105.769  
## Mar 2019 2163.893 1291.3644 3036.421  
## Apr 2019 2084.273 1090.4490 3078.097  
## May 2019 2643.407 1541.5347 3745.278  
## Jun 2019 3070.358 1870.1029 4270.612  
## Jul 2019 3463.470 2172.2861 4754.653  
## Aug 2019 3941.253 2565.1155 5317.390  
## Sep 2019 4010.441 2554.2800 5466.603  
## Oct 2019 3876.609 2344.5802 5408.639  
## Nov 2019 3145.963 1541.6330 4750.293  
## Dec 2019 2478.961 805.4348 4152.488  
## Jan 2020 2539.748 799.7401 4279.756  
## Feb 2020 2455.148 651.1102 4259.186

# Multiplicativo  
  
  
fit\_holt\_multi <- hw(train,h=14, seasonal = "multiplicative", level=95)  
  
  
  
# Plotando os modelos  
  
h1 <- autoplot(train, serie="Serie Original") +   
 autolayer(fit\_holt\_ad$fitted, serie = "Modelo HW Aditivo") +  
 autolayer(fit\_holt\_ad, serie = "Previsão",showgap = F, PI=F) +  
 autolayer(test, series= "Base de Teste")  
  
  
h2 <- autoplot(train, serie="Serie Original") +   
 autolayer(fit\_holt\_multi$fitted, serie = "Modelo HW Multiplicativo") +  
 autolayer(fit\_holt\_multi, serie = "Previsão",showgap = F, PI=F) +  
 autolayer(test, series= "Base de Teste")  
  
  
  
# Utilizando modelo ETS ( Error, Trend, Seasonal)  
  
ets\_model <- ets(train)  
fit\_ets <- forecast(ets\_model, h=14)  
  
  
h3 <- autoplot(train, serie="Serie Original") +   
 autolayer(fit\_ets$fitted, serie = "Modelo HW ETS") +  
 autolayer(fit\_ets, serie = "Previsão",showgap = F, PI=F) +  
 autolayer(test, series= "Base de Teste")  
  
# Resultado do Modelo ETS  
  
grid.arrange(h1,h2,h3, nrow=3)



pander(summary(ets\_model))

ETS(A,N,A)

Call: ets(y = train)

Smoothing parameters: alpha = 0.8702 gamma = 1e-04

Initial states: l = 2145.9902 s = -534.9342 137.9396 890.1354 1002.916 900.3112 484.3706 96.9238 -325.0633 -865.0764 -790.5557 -568.6864 -428.281

sigma: 280.5454

AIC AICc BIC

2354.305 2358.055 2398.852

Training set error measures: ME RMSE MAE MPE MAPE MASE Training set 6.248057 266.5592 203.7757 -0.3261722 9.142053 0.6258205 ACF1 Training set 0.01554042

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | ME | RMSE | MAE | MPE | MAPE | MASE | ACF1 |
| **Training set** | 6.248 | 266.6 | 203.8 | -0.3262 | 9.142 | 0.6258 | 0.01554 |

print("Modelo HW - Aditivo")

## [1] "Modelo HW - Aditivo"

pander(accuracy(fit\_holt\_ad))

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | ME | RMSE | MAE | MPE | MAPE | MASE | ACF1 |
| **Training set** | -2.831 | 267.3 | 206.2 | -0.7949 | 9.331 | 0.6333 | 0.02211 |

print("Modelo HW - Multiplicativo")

## [1] "Modelo HW - Multiplicativo"

pander(accuracy(fit\_holt\_multi))

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | ME | RMSE | MAE | MPE | MAPE | MASE | ACF1 |
| **Training set** | 2.874 | 289.7 | 218.9 | -1.153 | 9.867 | 0.6722 | 0.1815 |

print("Modelo ETS")

## [1] "Modelo ETS"

pander(accuracy(fit\_ets))

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | ME | RMSE | MAE | MPE | MAPE | MASE | ACF1 |
| **Training set** | 6.248 | 266.6 | 203.8 | -0.3262 | 9.142 | 0.6258 | 0.01554 |

AIC(fit\_holt\_ad$model)

## [1] 2359.071

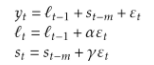
AIC(fit\_holt\_multi$model)

## [1] 2396.305

AIC(ets\_model)

## [1] 2354.305

A comparação entre os modelos HWA e HWM parece pender para o HWA tanto no que diz respeito aos resultados de acurácia quanto aos resultados do AIC. O modelo ETS, que atinge métricas melhores que os dois outros em todas as métricas, propõe uma abordagem com erro e sazonalidades aditivos e sem tendencia, cuja equação segue abaixo:



GitHub Logo

## SARIMA

Vamos passar ao teste com o modelo ARIMA com sazonalidade, SARIMA.

autoarima\_model <- auto.arima(train, seasonal = TRUE, stepwise=FALSE, approximation = FALSE)  
pander(summary(autoarima\_model))

Series: train ARIMA(1,0,0)(2,1,0)[12] with drift

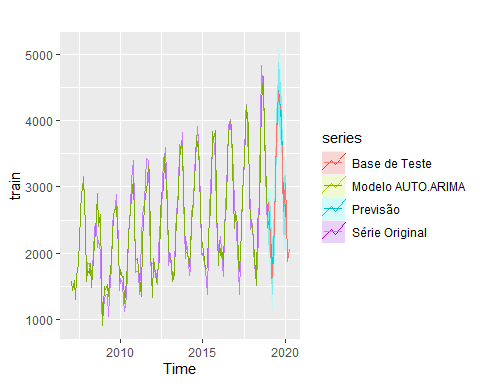
Coefficients: ar1 sar1 sar2 drift 0.6065 -0.6751 -0.4413 8.0281 s.e. 0.0689 0.0868 0.0861 2.4571

sigma^2 estimated as 72763: log likelihood=-928.45 AIC=1866.9 AICc=1867.37 BIC=1881.31

Training set error measures: ME RMSE MAE MPE MAPE MASE Training set -6.591897 254.3198 191.3922 -1.834856 8.615046 0.5877892 ACF1 Training set -0.01089265

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | ME | RMSE | MAE | MPE | MAPE | MASE | ACF1 |
| **Training set** | -6.592 | 254.3 | 191.4 | -1.835 | 8.615 | 0.5878 | -0.01089 |

fit\_autoarima\_model <- forecast(autoarima\_model, h=14)  
  
autoplot(train, series ="Série Original") +  
 autolayer(fit\_autoarima\_model$fitted, series = "Modelo AUTO.ARIMA")+  
 autolayer(fit\_autoarima\_model, series = "Previsão", showgap = FALSE) +  
 autolayer(test, series= "Base de Teste")



t\_test(autoarima\_model)

## Coeffs Std.Errors t Crit.Values Rej.H0  
## ar1 0.6065187 0.06890490 8.802258 1.977054 TRUE  
## sar1 -0.6750799 0.08683187 7.774564 1.977054 TRUE  
## sar2 -0.4412634 0.08610028 5.124994 1.977054 TRUE  
## drift 8.0281107 2.45707100 3.267350 1.977054 TRUE

print("Modelo Auto Arima")

## [1] "Modelo Auto Arima"

pander(accuracy(fit\_autoarima\_model))

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | ME | RMSE | MAE | MPE | MAPE | MASE | ACF1 |
| **Training set** | -6.592 | 254.3 | 191.4 | -1.835 | 8.615 | 0.5878 | -0.01089 |

AIC(fit\_autoarima\_model$model)

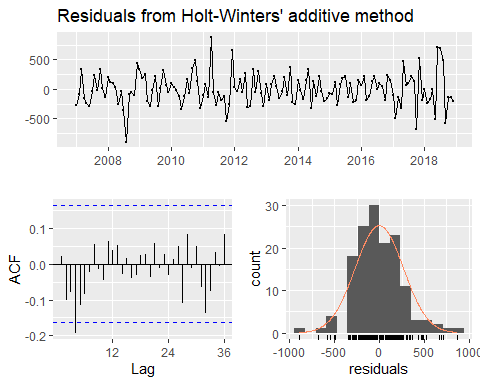
## [1] 1866.898

Na parte não sazonal, o modelo sugere ordem 1 de autorregressividade, 0 de diferenciação e de MA. Na parte sazonal, o modelo sugere ordem 2 de autoregressividade, 1 de diferenciação e 0 de MA com 12 observações anuais.

## Análise de Resíduos ( correlaçaõ)

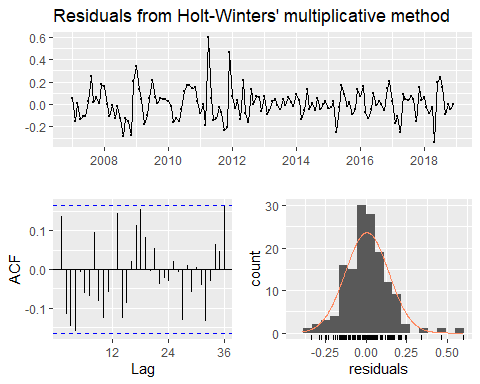
Avaliando os resíduos de cada um dos modelos é possível verificar pelos p-values do Ljung-Box test que o modelo aditivo apresenta um p-value um pouco maior que 0,05, o ETS(A,N,A) possui um p-value maior que 0,05. Ambos possuem resíduos independentes e identicamente distribuídos. No caso do HW multiplicativo o p-value é muito pequeno permitindo rejeitar a hipótese nula, isto é, os resíduos do modelo apresentam problemas de correlação entre os resíduos. O modelo sarima possui um p-value maior que 0,05 e que o modelo ETS. Do ponto de vista dos resíduos o modelo autoarima parece ser o que melhor se adequa.

checkresiduals(fit\_holt\_ad)



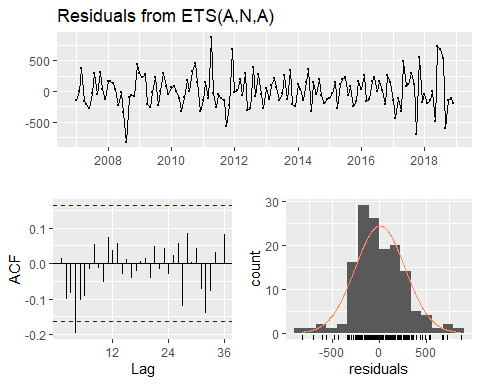
##   
## Ljung-Box test  
##   
## data: Residuals from Holt-Winters' additive method  
## Q\* = 15.44, df = 8, p-value = 0.05113  
##   
## Model df: 16. Total lags used: 24

checkresiduals(fit\_holt\_multi)



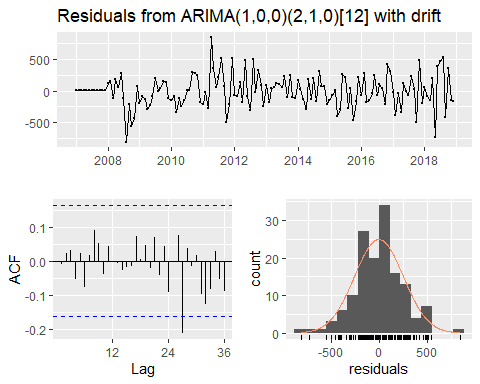
##   
## Ljung-Box test  
##   
## data: Residuals from Holt-Winters' multiplicative method  
## Q\* = 33.994, df = 8, p-value = 4.073e-05  
##   
## Model df: 16. Total lags used: 24

checkresiduals(fit\_ets)



##   
## Ljung-Box test  
##   
## data: Residuals from ETS(A,N,A)  
## Q\* = 15.626, df = 10, p-value = 0.1108  
##   
## Model df: 14. Total lags used: 24

checkresiduals(fit\_autoarima\_model)



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(1,0,0)(2,1,0)[12] with drift  
## Q\* = 8.3969, df = 20, p-value = 0.9889  
##   
## Model df: 4. Total lags used: 24

## Teste de Heterocedasticidade

Apenas o modelo Holt Winters multiplicativo se revela heterocedastico.

ArchTest(fit\_holt\_ad$residuals,lags = 12)

##   
## ARCH LM-test; Null hypothesis: no ARCH effects  
##   
## data: fit\_holt\_ad$residuals  
## Chi-squared = 15.531, df = 12, p-value = 0.2137

ArchTest(fit\_holt\_multi$residuals,lags = 12)

##   
## ARCH LM-test; Null hypothesis: no ARCH effects  
##   
## data: fit\_holt\_multi$residuals  
## Chi-squared = 22.676, df = 12, p-value = 0.0306

ArchTest(fit\_ets$residuals,lags = 12)

##   
## ARCH LM-test; Null hypothesis: no ARCH effects  
##   
## data: fit\_ets$residuals  
## Chi-squared = 20.587, df = 12, p-value = 0.05676

ArchTest(fit\_autoarima\_model$residuals,lags = 12)

##   
## ARCH LM-test; Null hypothesis: no ARCH effects  
##   
## data: fit\_autoarima\_model$residuals  
## Chi-squared = 6.7125, df = 12, p-value = 0.876

### Teste de Normalidade

O único modelo que passa nos dois testes de normalidade é o SARIMA.

jb.norm.test(fit\_holt\_ad$residuals, nrepl = 2000)

##   
## Jarque-Bera test for normality  
##   
## data: fit\_holt\_ad$residuals  
## JB = 8.9917, p-value = 0.0185

jb.norm.test(fit\_holt\_multi$residuals,nrepl = 2000)

##   
## Jarque-Bera test for normality  
##   
## data: fit\_holt\_multi$residuals  
## JB = 53.604, p-value < 2.2e-16

jb.norm.test(fit\_ets$residuals,nrepl = 2000)

##   
## Jarque-Bera test for normality  
##   
## data: fit\_ets$residuals  
## JB = 8.4071, p-value = 0.0255

jb.norm.test(fit\_autoarima\_model$residuals,nrepl = 2000)

##   
## Jarque-Bera test for normality  
##   
## data: fit\_autoarima\_model$residuals  
## JB = 5.2427, p-value = 0.06

shapiro.test(fit\_holt\_ad$residuals)

##   
## Shapiro-Wilk normality test  
##   
## data: fit\_holt\_ad$residuals  
## W = 0.98093, p-value = 0.04234

shapiro.test(fit\_holt\_multi$residuals)

##   
## Shapiro-Wilk normality test  
##   
## data: fit\_holt\_multi$residuals  
## W = 0.9606, p-value = 0.0003821

shapiro.test(fit\_ets$residuals)

##   
## Shapiro-Wilk normality test  
##   
## data: fit\_ets$residuals  
## W = 0.97925, p-value = 0.02777

shapiro.test(fit\_autoarima\_model$residuals)

##   
## Shapiro-Wilk normality test  
##   
## data: fit\_autoarima\_model$residuals  
## W = 0.9866, p-value = 0.1773

# Conclusão

Abaixo apresenta-se um resumo dos resultados dos resíduos de cada um dos modelos, é possível verificar que o Holt Winters Multiplicativo não passou em nenhum teste e que os modelos de simplificação exponencial não passaram nos testes de normalidade. O modelo SARIMA passou em todos os testes.

comparativo\_residuos <- data.frame(modelo=c("Holt\_Winters\_Aditivo","Holt\_Winters\_Multiplicativo","ETS","Auto\_Arima\_Sazonal"))  
comparativo\_residuos$LjungBox <- c("S","N","S","S")  
comparativo\_residuos$Norm\_JB <- c("N","N","N","S")  
comparativo\_residuos$Norm\_SH <- c("N","N","N","S")  
comparativo\_residuos$Homocedasticidade <- c("S","N","S","S")  
  
pander(comparativo\_residuos)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| modelo | LjungBox | Norm\_JB | Norm\_SH | Homocedasticidade |
| Holt\_Winters\_Aditivo | S | N | N | S |
| Holt\_Winters\_Multiplicativo | N | N | N | N |
| ETS | S | N | N | S |
| Auto\_Arima\_Sazonal | S | S | S | S |

Abaixo apresenta-se um comparativo dos resultados levando em consideração o AIC, que só pode ser comparado entre modelos da mesma categoria, e o RMSE. Dos modelos por simplificação exponencial o ETS apresentou o menor AIC e o menor RMSE. Comparando o RMSE do ETS com o SARIMA vemos que o SARIMA se adequa melhor

comparativo\_indices <- data.frame(modelo=c("Holt\_Winters\_Aditivo","Holt\_Winters\_Multiplicativo","ETS","Auto\_Arima\_Sazonal"))  
comparativo\_indices$AIC <- c(AIC(fit\_holt\_ad$model),AIC(fit\_holt\_multi$model),AIC(fit\_ets$model),AIC(fit\_autoarima\_model$model))  
  
comparativo\_indices$RMSE <- c(data.frame(accuracy(fit\_holt\_ad))$RMSE,data.frame(accuracy(fit\_holt\_multi))$RMSE,data.frame(accuracy(fit\_ets))$RMSE,data.frame(accuracy(fit\_autoarima\_model))$RMSE)  
  
pander(comparativo\_indices)

|  |  |  |
| --- | --- | --- |
| modelo | AIC | RMSE |
| Holt\_Winters\_Aditivo | 2359 | 267.3 |
| Holt\_Winters\_Multiplicativo | 2396 | 289.7 |
| ETS | 2354 | 266.6 |
| Auto\_Arima\_Sazonal | 1867 | 254.3 |

Face ao exposto, conclui-se que o modelo SARIMA parece se adequar melhor na comparação com os outros.