Trabalho Final de Séries Temporais

Carlos Alberto Torres Quintanilha Neto

Turma: MRJ02021-TBABD-8

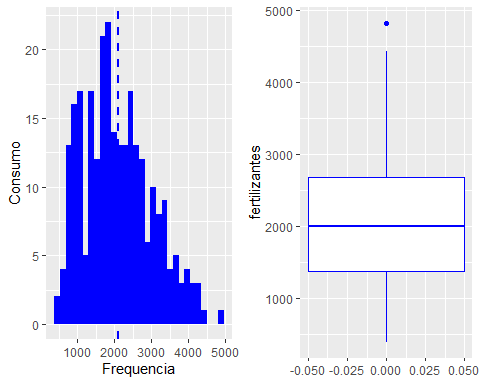
12/12/2020

##### 

# 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 e seu gráfico foi plotado e decomposto.

É possível identificar no gráfico uma 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)

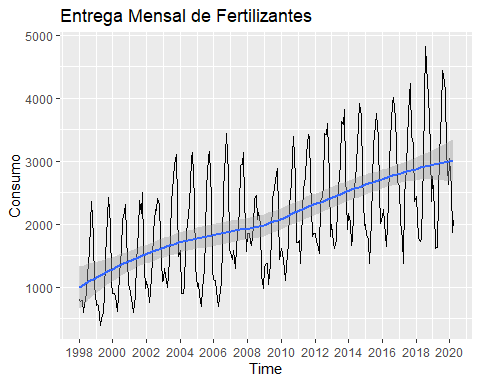
## [1] 2020 4

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

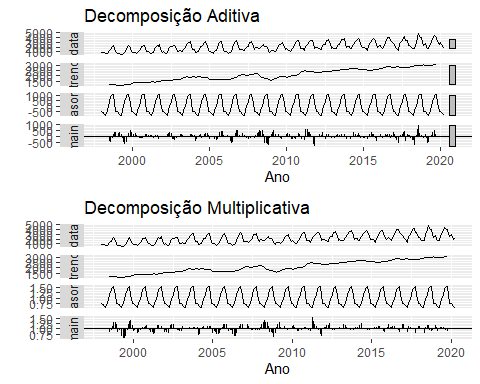
## Warning: Ignoring unknown parameters: frequency, start

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

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'



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,nrow = 2)

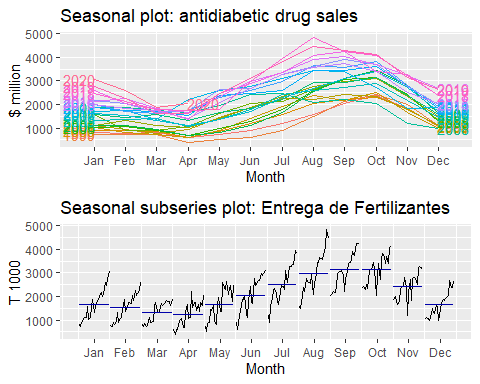


### Sazonalidade

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.

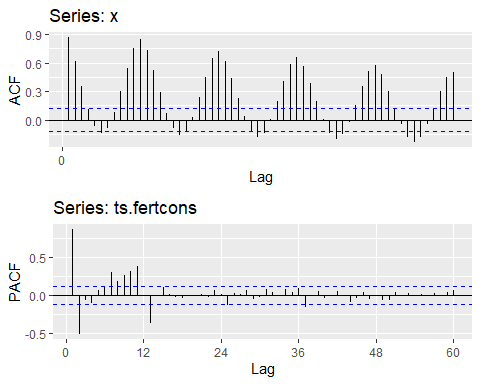
q1 <- ggseasonplot(ts.fertcons, year.labels=TRUE, year.labels.left=TRUE) +  
 ylab("$ million") +  
 ggtitle("Seasonal plot: antidiabetic drug sales")  
  
q2 <- ggsubseriesplot(ts.fertcons) +  
 ylab("T 1000") +  
 ggtitle("Seasonal subseries plot: Entrega de Fertilizantes")  
  
  
grid.arrange(q1,q2,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



## Autocorrelação

No correlograma da série é possível notar a presença da sazonalidade nos picos múltiplos de 12 indicando que a presença de uma diferenciação na parte sazonal pode melhorar a perfornce dos modelos. 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.

grid.arrange(ggAcf(ts.fertcons,lag=60),ggPacf(ts.fertcons,lag=60),nrow = 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 <- adf.test(ts.fertcons)

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

KPSS <- kpss.test(ts.fertcons)

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

PP <- pp.test(ts.fertcons)

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

avaliacao <- data.frame(Teste=c("ADF", "KPPS","PP"))  
avaliacao$H0 <- c("Nao Estacionaria","Estacionaria","Nao Estacionaria")  
avaliacao$Resultado <- c( (ADF$p.value > 0.05), (KPSS$p.value > 0.05), (PP$p.value > 0.05))  
  
pander(avaliacao, split.table=Inf)

|  |  |  |
| --- | --- | --- |
| Teste | H0 | Resultado |
| ADF | Nao Estacionaria | FALSE |
| KPPS | Estacionaria | FALSE |
| PP | Nao Estacionaria | FALSE |

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. A presença de 1 spike no primeiro lag indica a possibilidade de que a presença de um grau de média móvel nos modelos possa melhorar sua performance.

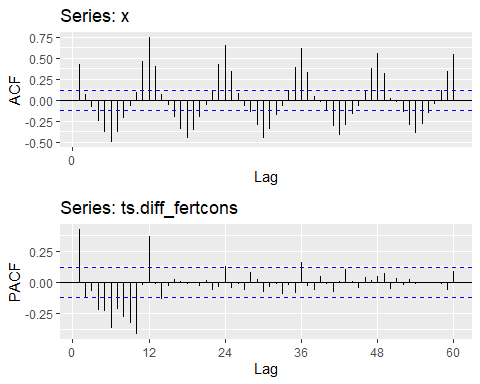
#Utililzando ndiff para saber o número de diferenciações necessárias  
ndiffs(ts.fertcons)

## [1] 1

nsdiffs(ts.fertcons)

## [1] 1

# Diferenciando a série  
ts.diff\_fertcons <- diff(ts.fertcons)  
  
grid.arrange(ggAcf(ts.diff\_fertcons,lag=60),ggPacf(ts.diff\_fertcons,lag=60),nrow = 2)



# Testando  
  
ADF\_Diff <- adf.test(ts.diff\_fertcons)

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

KPSS\_Diff <- kpss.test(ts.diff\_fertcons)

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

PP\_Diff <- pp.test(ts.diff\_fertcons)

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

avaliacao\_diff <- data.frame(Teste=c("ADF\_DIFF", "KPPS\_DIFF","PP\_DIFF"))  
avaliacao\_diff$H0 <- c("Nao Estacionaria","Estacionaria","Nao Estacionaria")  
avaliacao\_diff$Resultado <- c( (ADF\_Diff$p.value > 0.05), (KPSS\_Diff$p.value > 0.05), (PP\_Diff$p.value > 0.05))  
  
pander(avaliacao\_diff, split.table=Inf)

|  |  |  |
| --- | --- | --- |
| Teste | H0 | Resultado |
| ADF\_DIFF | Nao Estacionaria | FALSE |
| KPPS\_DIFF | Estacionaria | TRUE |
| PP\_DIFF | Nao Estacionaria | FALSE |

# 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 adequação dos mesmos para o forecasting, a saber:

* **Holt Winters Aditivo (HWA)**
* **Holt Winters Multiplicativo (HWM)**
* **ETS**

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

# HW Aditivo  
  
fit\_holt\_ad <- hw(train,h=16, 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 = 16, 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  
## Mar 2020 2244.694 378.8078 4110.580  
## Apr 2020 2165.074 239.3112 4090.838

# HW Multiplicativo  
  
  
fit\_holt\_multi <- hw(train,h=16, seasonal = "multiplicative", level=95)  
summary(fit\_holt\_multi)

##   
## Forecast method: Holt-Winters' multiplicative method  
##   
## Model Information:  
## Holt-Winters' multiplicative method   
##   
## Call:  
## hw(y = train, h = 16, seasonal = "multiplicative", level = 95)   
##   
## Smoothing parameters:  
## alpha = 0.574   
## beta = 2e-04   
## gamma = 0.2096   
##   
## Initial states:  
## l = 2239.6086   
## b = 3.7549   
## s = 0.6186 1.0998 1.432 1.4238 1.3205 1.2327  
## 1.0968 0.9359 0.7032 0.7402 0.7347 0.6617  
##   
## sigma: 0.1409  
##   
## AIC AICc BIC   
## 2396.305 2401.162 2446.792   
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE ACF1  
## Training set 2.873734 289.6995 218.8679 -1.153256 9.866531 0.6721706 0.1814767  
##   
## Forecasts:  
## Point Forecast Lo 95 Hi 95  
## Jan 2019 2350.092 1700.9004 2999.283  
## Feb 2019 2104.078 1432.4040 2775.752  
## Mar 2019 1871.226 1202.7512 2539.700  
## Apr 2019 1813.319 1102.9816 2523.656  
## May 2019 2604.476 1501.2991 3707.653  
## Jun 2019 3371.474 1843.1537 4899.795  
## Jul 2019 3906.228 2025.9978 5786.458  
## Aug 2019 4413.236 2171.5315 6654.940  
## Sep 2019 4293.150 2003.4031 6582.897  
## Oct 2019 4027.382 1781.2963 6273.468  
## Nov 2019 3314.405 1388.2507 5240.560  
## Dec 2019 2403.021 952.1158 3853.927  
## Jan 2020 2391.929 851.2651 3932.594  
## Feb 2020 2141.494 717.5890 3565.400  
## Mar 2020 1904.464 599.5682 3209.361  
## Apr 2020 1845.493 544.5077 3146.479

# Utilizando modelo ETS ( Error, Trend, Seasonal)  
  
ets\_model <- ets(train)  
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

# Efetuando o forecast  
  
fit\_ets <- forecast(ets\_model, h=16)  
summary(fit\_ets)

##   
## Forecast method: ETS(A,N,A)  
##   
## Model Information:  
## 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   
##   
## 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  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## Jan 2019 2500.610 2141.0770 2860.144 1950.7514 3050.469  
## Feb 2019 2360.208 1883.6171 2836.799 1631.3249 3089.092  
## Mar 2019 2138.379 1568.2787 2708.479 1266.4860 3010.271  
## Apr 2019 2063.806 1413.5073 2714.105 1069.2597 3058.353  
## May 2019 2603.811 1882.1713 3325.451 1500.1583 3707.464  
## Jun 2019 3025.787 2239.2513 3812.323 1822.8844 4228.690  
## Jul 2019 3413.340 2566.8692 4259.812 2118.7745 4707.906  
## Aug 2019 3829.260 2926.8253 4731.695 2449.1052 5209.415  
## Sep 2019 3931.763 2976.6381 4886.888 2471.0256 5392.500  
## Oct 2019 3818.985 2813.9287 4824.041 2281.8840 5356.086  
## Nov 2019 3066.828 2014.2061 4119.450 1456.9817 4676.674  
## Dec 2019 2393.940 1295.8009 3492.080 714.4810 4073.400  
## Jan 2020 2500.610 1358.7762 3642.445 754.3257 4246.895  
## Feb 2020 2360.208 1176.2910 3544.126 549.5630 4170.854  
## Mar 2020 2138.379 913.8237 3362.934 265.5834 4011.174  
## Apr 2020 2063.806 799.9195 3327.693 130.8583 3996.754

# Plotando os modelos  
  
h1 <- autoplot(train, serie="Serie Original", start=c(2007,1)) +   
 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") +  
 scale\_x\_continuous(breaks = scales::extended\_breaks(12))+  
 theme(axis.text.x=element\_text(angle=-45,vjust=0.5))+  
 ggtitle("Modelo HW Aditivo")

## Warning: Ignoring unknown parameters: start

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

h2 <- autoplot(train, serie="Serie Original", start=c(2007,1)) +   
 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")+  
 scale\_x\_continuous(breaks = scales::extended\_breaks(12))+  
 theme(axis.text.x=element\_text(angle=-45,vjust=0.5))+  
 ggtitle("Modelo HW Multiplicativo")

## Warning: Ignoring unknown parameters: start

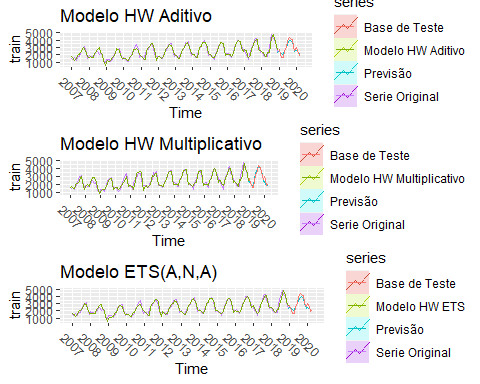
## Scale for 'x' is already present. Adding another scale for 'x', which will  
## replace the existing scale.

h3 <- autoplot(train, serie="Serie Original",start=c(2007,1)) +   
 autolayer(fit\_ets$fitted, serie = "Modelo HW ETS") +  
 autolayer(fit\_ets, serie = "Previsão",showgap = F, PI=F) +  
 autolayer(test, series= "Base de Teste")+  
 scale\_x\_continuous(breaks = scales::extended\_breaks(12))+  
 theme(axis.text.x=element\_text(angle=-45,vjust=0.5))+  
 ggtitle("Modelo ETS(A,N,A)")

## Warning: Ignoring unknown parameters: start

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

grid.arrange(h1,h2,h3, nrow=3)



# Resultados dos Modelos  
  
# Modelo HW - Aditivo  
  
pander(accuracy(fit\_holt\_ad,test), split.table = Inf)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | ME | RMSE | MAE | MPE | MAPE | MASE | ACF1 | Theil’s U |
| **Training set** | -2.831 | 267.3 | 206.2 | -0.7949 | 9.331 | 0.6333 | 0.02211 | NA |
| **Test set** | 41.96 | 313.6 | 265.5 | -1.747 | 10.67 | 0.8155 | 0.5448 | 0.587 |

# Modelo HW - Multiplicativo  
  
pander(accuracy(fit\_holt\_multi,test), split.table = Inf)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | ME | RMSE | MAE | MPE | MAPE | MASE | ACF1 | Theil’s U |
| **Training set** | 2.874 | 289.7 | 218.9 | -1.153 | 9.867 | 0.6722 | 0.1815 | NA |
| **Test set** | 64.53 | 266.1 | 208.5 | 1.891 | 8.211 | 0.6403 | 0.4556 | 0.4538 |

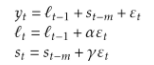
# Modelo ETS  
  
pander(accuracy(fit\_ets,test), split.table = Inf)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | ME | RMSE | MAE | MPE | MAPE | MASE | ACF1 | Theil’s U |
| **Training set** | 6.248 | 266.6 | 203.8 | -0.3262 | 9.142 | 0.6258 | 0.01554 | NA |
| **Test set** | 98.64 | 332.3 | 283.8 | 0.3387 | 10.8 | 0.8716 | 0.6083 | 0.5863 |

# Comparando os AIC´s  
  
comparativo\_SME\_aic <- data.frame(Modelo=c("HWA","HWM","TSE"))  
comparativo\_SME\_aic$AIC <- c(AIC(fit\_holt\_ad$model),AIC(fit\_holt\_multi$model),AIC(ets\_model))  
pander(comparativo\_SME\_aic, split.table = Inf)

|  |  |
| --- | --- |
| Modelo | AIC |
| HWA | 2359 |
| HWM | 2396 |
| TSE | 2354 |

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:



ETS(A,N,A)

## SARIMA

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

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

# Validando a viabilidade do modelo  
  
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

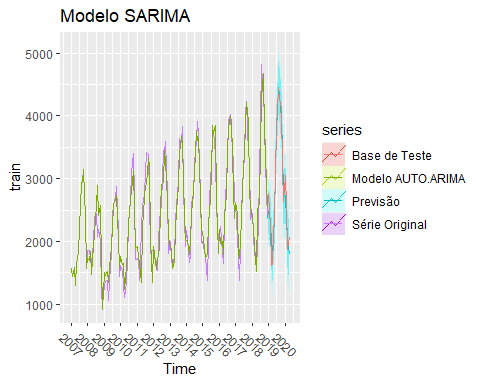
# Efetuando o Forecast  
  
fit\_autoarima\_model <- forecast(autoarima\_model, h=16)  
summary(fit\_autoarima\_model)

##   
## Forecast method: ARIMA(1,0,0)(2,1,0)[12] with drift  
##   
## Model Information:  
## 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  
##   
## 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  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## Jan 2019 2475.516 2129.822 2821.211 1946.822 3004.210  
## Feb 2019 2317.505 1913.195 2721.815 1699.166 2935.843  
## Mar 2019 1971.494 1547.656 2395.331 1323.290 2619.698  
## Apr 2019 1794.844 1364.045 2225.642 1135.994 2453.693  
## May 2019 2382.508 1949.177 2815.839 1719.785 3045.231  
## Jun 2019 3159.596 2725.337 3593.855 2495.454 3823.738  
## Jul 2019 3752.496 3317.896 4187.096 3087.833 4417.159  
## Aug 2019 4449.480 4014.755 4884.205 3784.625 5114.335  
## Sep 2019 4347.247 3912.476 4782.019 3682.322 5012.173  
## Oct 2019 3948.888 3514.100 4383.677 3283.937 4613.840  
## Nov 2019 3445.919 3011.124 3880.713 2780.958 4110.880  
## Dec 2019 2705.486 2270.689 3140.283 2040.522 3370.450  
## Jan 2020 2730.520 2281.234 3179.806 2043.397 3417.644  
## Feb 2020 2354.179 1899.679 2808.679 1659.081 3049.276  
## Mar 2020 2038.016 1581.613 2494.419 1340.008 2736.024  
## Apr 2020 1798.158 1341.057 2255.259 1099.083 2497.234

# Plotando o modelo  
  
autoplot(train, series ="Série Original",start=c(2007,1)) +  
 autolayer(fit\_autoarima\_model$fitted, series = "Modelo AUTO.ARIMA")+  
 autolayer(fit\_autoarima\_model, series = "Previsão", showgap = FALSE) +  
 autolayer(test, series= "Base de Teste")+  
 scale\_x\_continuous(breaks = scales::extended\_breaks(12))+  
 theme(axis.text.x=element\_text(angle=-45,vjust=0.5))+  
 ggtitle("Modelo SARIMA")

## Warning: Ignoring unknown parameters: start

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



# Resultado do Modelo   
  
pander(accuracy(fit\_autoarima\_model,test), split.table = Inf)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | ME | RMSE | MAE | MPE | MAPE | MASE | ACF1 | Theil’s U |
| **Training set** | -6.592 | 254.3 | 191.4 | -1.835 | 8.615 | 0.5878 | -0.01089 | NA |
| **Test set** | 7.259 | 196 | 165.7 | -0.5639 | 6.89 | 0.5089 | -0.04593 | 0.3479 |

# AIC  
  
AIC(fit\_autoarima\_model$model)

## [1] 1866.898

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

Como mostrado anteriormente, a série apresenta uma tendência que pode ser retirada com uma diferenciação. Como o modelo sugerido não está efetuando esta diferenciação na parte não sazonal, vamos testar um modelo com esta difernciação. Adicionalmente vamos incluir 1 grau de MA na parte não sazonal e verificar como fica o comportamento frente ao modelo sugerido pelo auto.arima.

sarima\_model <- Arima(train, order= c(1,1,1), seasonal = list(order=c(2,1,0), period=12))  
summary(sarima\_model)

## Series: train   
## ARIMA(1,1,1)(2,1,0)[12]   
##   
## Coefficients:  
## ar1 ma1 sar1 sar2  
## 0.6191 -1.0000 -0.6695 -0.4346  
## s.e. 0.0702 0.0305 0.0875 0.0868  
##   
## sigma^2 estimated as 73454: log likelihood=-924.13  
## AIC=1858.25 AICc=1858.73 BIC=1872.63  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 13.56028 254.5237 186.3283 -0.5070671 8.023416 0.5722375  
## ACF1  
## Training set -0.02401995

# Testando a viabilidade do modelo   
  
t\_test(sarima\_model)

## Coeffs Std.Errors t Crit.Values Rej.H0  
## ar1 0.6191367 0.07022158 8.816901 1.977304 TRUE  
## ma1 -0.9999998 0.03045683 32.833354 1.977304 TRUE  
## sar1 -0.6695316 0.08747287 7.654164 1.977304 TRUE  
## sar2 -0.4346400 0.08678131 5.008452 1.977304 TRUE

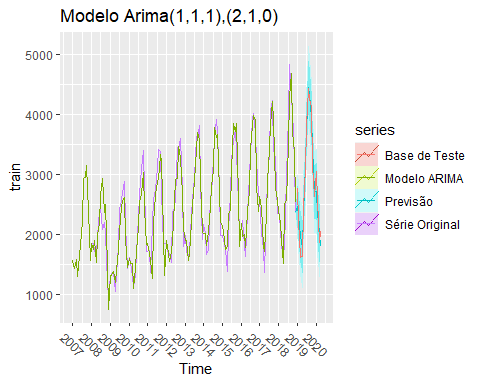
# Efetuando o Forecast  
  
fit\_sarima\_model <- forecast(sarima\_model, h=16)  
summary(fit\_sarima\_model)

##   
## Forecast method: ARIMA(1,1,1)(2,1,0)[12]  
##   
## Model Information:  
## Series: train   
## ARIMA(1,1,1)(2,1,0)[12]   
##   
## Coefficients:  
## ar1 ma1 sar1 sar2  
## 0.6191 -1.0000 -0.6695 -0.4346  
## s.e. 0.0702 0.0305 0.0875 0.0868  
##   
## sigma^2 estimated as 73454: log likelihood=-924.13  
## AIC=1858.25 AICc=1858.73 BIC=1872.63  
##   
## Error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 13.56028 254.5237 186.3283 -0.5070671 8.023416 0.5722375  
## ACF1  
## Training set -0.02401995  
##   
## Forecasts:  
## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## Jan 2019 2474.075 2125.271 2822.878 1940.626 3007.524  
## Feb 2019 2312.784 1900.994 2724.573 1683.006 2942.561  
## Mar 2019 1967.637 1533.217 2402.056 1303.249 2632.024  
## Apr 2019 1791.877 1348.540 2235.214 1113.852 2469.902  
## May 2019 2376.653 1929.608 2823.698 1692.957 3060.349  
## Jun 2019 3157.046 2708.380 3605.712 2470.870 3843.221  
## Jul 2019 3753.756 3304.342 4203.171 3066.436 4441.077  
## Aug 2019 4452.603 4002.822 4902.383 3764.723 5140.483  
## Sep 2019 4346.837 3896.867 4796.807 3658.668 5035.007  
## Oct 2019 3948.784 3498.711 4398.857 3260.456 4637.111  
## Nov 2019 3444.159 2994.027 3894.290 2755.742 4132.575  
## Dec 2019 2701.750 2251.584 3151.915 2013.281 3390.218  
## Jan 2020 2727.462 2260.577 3194.348 2013.422 3441.502  
## Feb 2020 2352.573 1878.840 2826.306 1628.061 3077.086  
## Mar 2020 2036.266 1559.568 2512.964 1307.219 2765.313  
## Apr 2020 1798.232 1320.176 2276.287 1067.108 2529.355

# Plotando o modelo  
  
autoplot(train, series ="Série Original",start=c(2007,1)) +  
 autolayer(fit\_sarima\_model$fitted, series = "Modelo ARIMA")+  
 autolayer(fit\_sarima\_model, series = "Previsão", showgap = FALSE) +  
 autolayer(test, series= "Base de Teste")+  
 scale\_x\_continuous(breaks = scales::extended\_breaks(12))+  
 theme(axis.text.x=element\_text(angle=-45,vjust=0.5))+  
 ggtitle("Modelo Arima(1,1,1),(2,1,0)")

## Warning: Ignoring unknown parameters: start

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



# Resultado do Modelo   
  
pander(accuracy(fit\_sarima\_model,test), split.table = Inf)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | ME | RMSE | MAE | MPE | MAPE | MASE | ACF1 | Theil’s U |
| **Training set** | 13.56 | 254.5 | 186.3 | -0.5071 | 8.023 | 0.5722 | -0.02402 | NA |
| **Test set** | 9.094 | 195.6 | 165.2 | -0.4788 | 6.857 | 0.5074 | -0.04702 | 0.3468 |

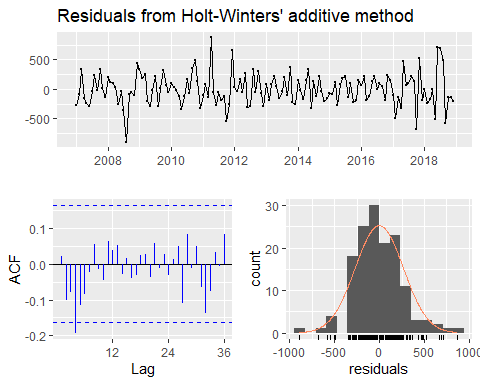
# AIC  
  
AIC(fit\_sarima\_model$model)

## [1] 1858.254

## Análise de Resíduos ( correlaçaõ)

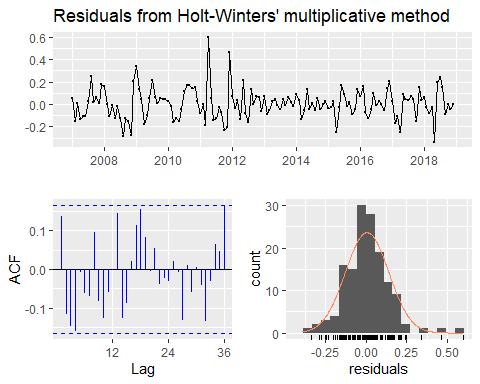
Avaliando os resíduos de cada um dos modelos é possível verificar pelo Ljung-Box test, cuja hipótese nula indica resíduos independentes, que os modelos HW aditivo, ETS, Arima((1,0,0),(2,1,0)) e ARIMA((1,1,1),(2,1,0)) apresentam um p-value maior que 0,05 , indicando que todos possuem resíduos independentes e identicamente distribuídos. No caso do HW multiplicativo o p-value é muito pequeno indicando a rejeição da hipótese nula, isto é, os resíduos do modelo apresentam problemas de correlação entre os resíduos.

FHWA\_LBOX <- checkresiduals(fit\_holt\_ad$model, col="blue")



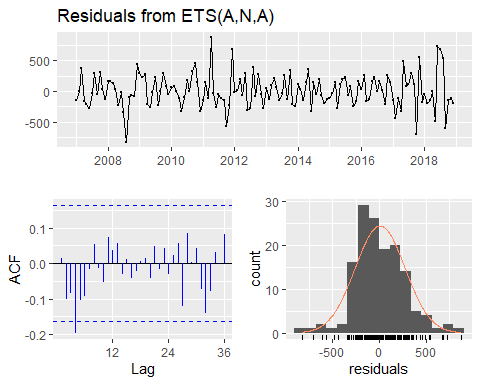
##   
## Ljung-Box test  
##   
## data: Residuals from Holt-Winters' additive method  
## Q\* = 15.44, df = 8, p-value = 0.05113  
##   
## Model df: 16. Total lags used: 24

FHWM\_LBOX <- checkresiduals(fit\_holt\_multi$model,col = "blue")



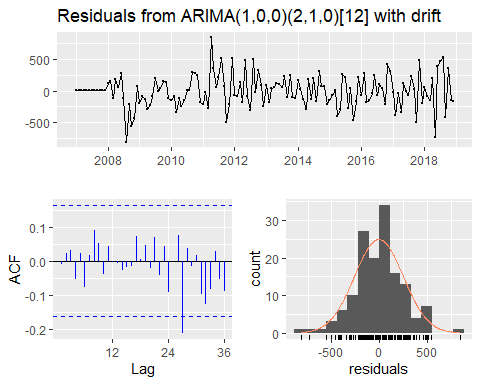
##   
## Ljung-Box test  
##   
## data: Residuals from Holt-Winters' multiplicative method  
## Q\* = 33.994, df = 8, p-value = 4.073e-05  
##   
## Model df: 16. Total lags used: 24

FETS\_LBOX <- checkresiduals(fit\_ets, col="blue")



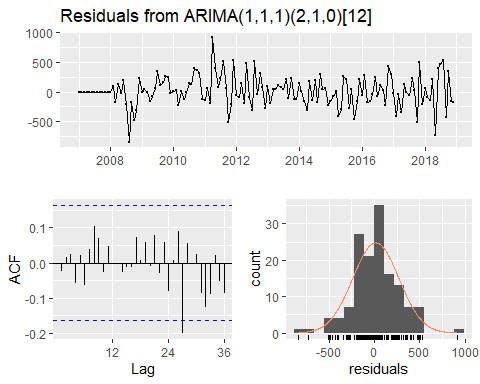
##   
## Ljung-Box test  
##   
## data: Residuals from ETS(A,N,A)  
## Q\* = 15.626, df = 10, p-value = 0.1108  
##   
## Model df: 14. Total lags used: 24

FAUTOARIMA\_LBOX <- checkresiduals(autoarima\_model,col="blue")



##   
## 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

FSARIMA\_LBOX <- checkresiduals(sarima\_model)



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(1,1,1)(2,1,0)[12]  
## Q\* = 9.2225, df = 20, p-value = 0.9802  
##   
## Model df: 4. Total lags used: 24

avaliacao\_LBOX <- data.frame(Modelo=c("HWA","HWM","ETS","ARIMA(1,0,0),(2,1,0)","ARIMA(1,1,1),(2,1,0)"))  
avaliacao\_LBOX$OK <- c((FHWA\_LBOX$p.value > 0.05), (FHWM\_LBOX$p.value > 0.05), (FETS\_LBOX$p.value > 0.05),(FAUTOARIMA\_LBOX$p.value>0.05),(FSARIMA\_LBOX$p.value > 0.05))  
  
pander(avaliacao\_LBOX, split.table=Inf)

|  |  |
| --- | --- |
| Modelo | OK |
| HWA | TRUE |
| HWM | FALSE |
| ETS | TRUE |
| ARIMA(1,0,0),(2,1,0) | TRUE |
| ARIMA(1,1,1),(2,1,0) | TRUE |

## Teste de Heterocedasticidade

Apenas o modelo Holt Winters Multiplicativo rejeita a hipótese nula indicando heterocedasticidade.

# Modelo HWA  
FHWA\_ARCH <- ArchTest(fit\_holt\_ad$model$residuals,lags = 16)  
  
#Modelo HWM  
FHWM\_ARCH <- ArchTest(fit\_holt\_multi$model$residuals,lags = 16)  
  
#Modelo ETS(A,N,A)  
FETS\_ARCH <- ArchTest(ets\_model$residuals,lags = 16)  
  
#Modelo AutoARIMA  
FAUTOARIMA\_ARCH <- ArchTest(autoarima\_model$residuals,lags = 16)  
  
#Modelo SARIMA  
  
FSARIMA\_ARCH <- ArchTest(sarima\_model$residuals,lags = 16)  
  
  
avaliacao\_ARCH <- data.frame(Modelo=c("HWA","HWM","ETS","ARIMA(1,0,0),(2,1,0)","ARIMA(1,1,1),(2,1,0)"))  
avaliacao\_ARCH$OK <- c((FHWA\_ARCH$p.value > 0.05), (FHWM\_ARCH$p.value > 0.05), (FETS\_ARCH$p.value > 0.05),(FAUTOARIMA\_ARCH$p.value>0.05),(FSARIMA\_ARCH$p.value > 0.05))  
  
pander(avaliacao\_ARCH, split.table=Inf)

|  |  |
| --- | --- |
| Modelo | OK |
| HWA | TRUE |
| HWM | FALSE |
| ETS | TRUE |
| ARIMA(1,0,0),(2,1,0) | TRUE |
| ARIMA(1,1,1),(2,1,0) | TRUE |

### Teste de Normalidade

O único modelo que passa nos dois testes de normalidade é o ARIMA(1,0,0),(2,1,0), sugerido pelo autoarima.

# Teste Jarque-Bera  
  
FHWA\_JB <- jb.norm.test(fit\_holt\_ad$model$residuals)  
FHWM\_JB <- jb.norm.test(fit\_holt\_multi$model$residuals)  
FETS\_JB <- jb.norm.test(ets\_model$residuals)  
FAUTOARIMA\_JB <- jb.norm.test(autoarima\_model$residuals)  
FSARIMA\_JB <- jb.norm.test(sarima\_model$residuals)  
  
avaliacao\_JB <- data.frame(Modelo=c("HWA","HWM","ETS","ARIMA(1,1,1),(2,1,0)","ARIMA(1,0,0),(2,1,0)"))  
avaliacao\_JB$OK <- c((FHWA\_JB$p.value > 0.05), (FHWM\_JB$p.value > 0.05), (FETS\_JB$p.value > 0.05),(FAUTOARIMA\_JB$p.value>0.05),(FSARIMA\_JB$p.value > 0.05))  
  
pander(avaliacao\_JB, split.table=Inf)

|  |  |
| --- | --- |
| Modelo | OK |
| HWA | FALSE |
| HWM | FALSE |
| ETS | FALSE |
| ARIMA(1,1,1),(2,1,0) | TRUE |
| ARIMA(1,0,0),(2,1,0) | FALSE |

# Teste Shapiro  
  
FHWA\_SH <- shapiro.test(fit\_holt\_ad$model$residuals)  
FHWM\_SH <- shapiro.test(fit\_holt\_multi$model$residuals)  
FETS\_SH <- shapiro.test(ets\_model$residuals)  
FAUTOARIMA\_SH <- shapiro.test(autoarima\_model$residuals)  
FSARIMA\_SH <- shapiro.test(sarima\_model$residuals)  
  
avaliacao\_SH <- data.frame(Modelo=c("HWA","HWM","ETS","ARIMA(1,1,1),(2,1,0)","ARIMA(1,1,1),(2,1,0)"))  
avaliacao\_SH$OK <- c((FHWA\_SH$p.value > 0.05), (FHWM\_SH$p.value > 0.05), (FETS\_SH$p.value > 0.05),(FAUTOARIMA\_SH$p.value>0.05),(FSARIMA\_SH$p.value > 0.05))  
  
FSARIMA\_SH$p.value

## [1] 0.02990345

pander(avaliacao\_SH, split.table=Inf)

|  |  |
| --- | --- |
| Modelo | OK |
| HWA | FALSE |
| HWM | FALSE |
| ETS | FALSE |
| ARIMA(1,1,1),(2,1,0) | TRUE |
| ARIMA(1,1,1),(2,1,0) | FALSE |

# 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 do auto.arima (1,0,0),(2,1,0) passou em todos os testes e o modelo SARIMA que tentamos (1,1,1),(2,1,0) não passou apenas no teste de normalidade.

comparativo\_residuos <- data.frame(Modelo=c("HWA","HWM","ETS(A,N,A)","Auto\_Arima","ARIMA(1,1,1),(2,1,0)"))  
comparativo\_residuos$LjungBox <- c("S","N","S","S","S")  
comparativo\_residuos$Norm\_JB <- c("N","N","N","S","N")  
comparativo\_residuos$Norm\_SH <- c("N","N","N","S","N")  
comparativo\_residuos$Homocedasticidade <- c("S","N","S","S","S")  
  
pander(comparativo\_residuos, split.table = Inf)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Modelo | LjungBox | Norm\_JB | Norm\_SH | Homocedasticidade |
| HWA | S | N | N | S |
| HWM | N | N | N | N |
| ETS(A,N,A) | S | N | N | S |
| Auto\_Arima | S | S | S | S |
| ARIMA(1,1,1),(2,1,0) | S | N | N | S |

Abaixo apresenta-se um comparativo dos resultados levando em consideração o AIC, que só pode ser comparado entre modelos da mesma categoria, o ETS (A,N,A) apresentou o menor AIC em comparação com o HWM e o HWA, já o SARIMA(1,1,1),(2,1,0) apresentou o melhor AIC em relação aos modelos SARIMA. Em relação ao RMSE o modelo que apresentou o melhor RMSE no teste foi o SARIMA(1,1,1),(2,1,0) no treinamento ofi o SARIMA(1,0,0),(2,1,0) oferecido pelo auto.arima.

comparativo\_indices <- data.frame(modelo=c("HWA-Treino","HWA-Teste","HWM-Treino","HWM-Test","ETS-Treino","ETS-Test","AUTOARIMA-Treino","AUTOARIMA-Teste","ARIMA(1,1,1),(2,1,0)-Treino","ARIMA(1,1,1),(2,1,0)-Teste"))  
comparativo\_indices$AIC <- c(AIC(fit\_holt\_ad$model),AIC(fit\_holt\_ad$model),AIC(fit\_holt\_multi$model),AIC(fit\_holt\_multi$model),AIC(fit\_ets$model),AIC(fit\_ets$model),AIC(fit\_autoarima\_model$model),AIC(fit\_autoarima\_model$model),AIC(fit\_sarima\_model$model),AIC(fit\_sarima\_model$model))  
  
comparativo\_indices$RMSE <- c(data.frame(accuracy(fit\_holt\_ad,test))$RMSE,data.frame(accuracy(fit\_holt\_multi,test))$RMSE,data.frame(accuracy(fit\_ets,test))$RMSE,data.frame(accuracy(fit\_autoarima\_model,test))$RMSE,data.frame(accuracy(fit\_sarima\_model,test))$RMSE)  
  
pander(comparativo\_indices, split.table = Inf)

|  |  |  |
| --- | --- | --- |
| modelo | AIC | RMSE |
| HWA-Treino | 2359 | 267.3 |
| HWA-Teste | 2359 | 313.6 |
| HWM-Treino | 2396 | 289.7 |
| HWM-Test | 2396 | 266.1 |
| ETS-Treino | 2354 | 266.6 |
| ETS-Test | 2354 | 332.3 |
| AUTOARIMA-Treino | 1867 | 254.3 |
| AUTOARIMA-Teste | 1867 | 196 |
| ARIMA(1,1,1),(2,1,0)-Treino | 1858 | 254.5 |
| ARIMA(1,1,1),(2,1,0)-Teste | 1858 | 195.6 |

Se desprezarmos os testes de normalidade, o modelo SARIMA (1,1,1),(2,1,0) seria o mais adequado porém, face ao exposto, concluimos que o modelo SARIMA (1,0,0),(2,1,0), apontado pelo comando auto.ariam, parece se adequar melhor na comparação com os outros por passar nos testes, apresentar bons parâmetros RMSE e AIC se commparado com outros modelos SARIMA.

A seguir uma comparação final do melhor modelo com previsão até dezembro de 2021

fit1\_autoarima\_model <- forecast(autoarima\_model, h=36)  
  
pander(accuracy(fit1\_autoarima\_model,test), split.table = Inf)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | ME | RMSE | MAE | MPE | MAPE | MASE | ACF1 | Theil’s U |
| **Training set** | -6.592 | 254.3 | 191.4 | -1.835 | 8.615 | 0.5878 | -0.01089 | NA |
| **Test set** | 7.259 | 196 | 165.7 | -0.5639 | 6.89 | 0.5089 | -0.04593 | 0.3479 |

autoplot(train, series ="Série Original",start=c(2007,1)) +  
 autolayer(fit1\_autoarima\_model$fitted, series = "Modelo AUTO.ARIMA")+  
 autolayer(fit1\_autoarima\_model, series = "Previsão", showgap = FALSE) +  
 autolayer(test, series= "Base de Teste")+  
 scale\_x\_continuous(breaks = scales::extended\_breaks(12))+  
 theme(axis.text.x=element\_text(angle=-45,vjust=0.5))+  
 ggtitle("Modelo SARIMA")

## Warning: Ignoring unknown parameters: start

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

