

Estudo de caso: Consumo de Energia Total – Comercial Brasil

1. Objetivo

O presente trabalho apresenta um estudo de caso utilizando os dados de consumo de energia total comercial do Brasil para a previsão destes a partir dos modelos de Naive, Holt, Holt-Winters Aditivo e Holt-Winters Multiplicativo. A série representa o consumo de energia elétrica na rede de distribuição comercial do Brasil, de janeiro de 2004 a fevereiro de 2020 em MegaWatts hora (MWh). A frequência de aquisição dos dados é mensal, ou seja, cada observação corresponde ao consumo acumulado durante o mês referente. Os dados podem ser vistos na Tabela 1.

Tabela 1 – Dados de consumo de energia elétrica na rede (MWh) – Total Comercial Brasil

Ano	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
2004	4.244.837	4.147.275	4.203.863	4.447.075	4.105.720	3.840.001	3.887.234	3.867.268	4.121.321	4.187.990	4.233.169	4.400.137
2005	4.492.544	4.365.606	4.552.520	4.687.080	4.377.604	4.297.519	4.147.257	4.141.719	4.361.233	4.412.144	4.536.867	4.662.469
2006	4.642.995	4.825.670	4.875.071	4.833.694	4.418.609	4.319.585	4.230.515	4.408.226	4.512.014	4.581.791	4.768.229	4.952.341
2007	4.941.580	4.923.943	5.099.470	5.287.341	4.864.118	4.587.140	4.486.888	4.520.225	4.787.183	4.881.812	5.105.641	5.161.662
2008	5.227.522	5.190.004	5.242.256	5.206.102	5.090.517	4.829.191	4.790.579	4.984.063	5.104.420	5.260.713	5.501.496	5.386.038
2009	5.367.747	5.442.517	5.698.327	5.685.560	5.247.517	5.031.436	4.996.843	5.129.084	5.304.022	5.493.177	5.763.756	6.094.576
2010	5.850.652	5.885.129	6.162.429	6.009.647	5.657.847	5.420.061	5.223.668	5.451.232	5.657.844	5.712.706	5.833.309	6.305.473
2011	6.250.067	6.330.954	6.373.328	6.297.738	6.002.047	5.718.674	5.654.276	5.879.222	6.113.102	6.156.478	6.207.693	6.497.944
2012	6.466.277	6.608.141	7.036.828	6.856.040	6.416.533	6.281.700	6.033.383	6.303.157	6.501.886	6.591.504	7.051.341	7.079.565
2013	7.202.468	7.035.408	7.146.581	7.050.788	6.832.911	6.594.674	6.453.638	6.601.480	6.760.321	7.071.235	7.382.159	7.572.217
2014	7.745.052	8.204.334	7.744.219	7.587.765	7.303.571	6.862.125	6.772.573	6.989.447	7.186.043	7.615.292	7.974.122	7.855.916
2015	8.076.143	8.089.865	7.907.780	7.906.721	7.363.736	7.072.660	6.838.562	7.077.425	7.161.169	7.630.771	7.790.700	7.852.009
2016	7.760.773	7.714.336	7.798.026	8.046.466	7.379.039	6.847.934	6.686.063	6.749.563	6.998.392	7.031.209	7.335.151	7.525.886
2017	7.778.453	7.683.653	7.964.592	7.697.452	7.248.489	7.010.216	6.655.413	6.718.745	7.114.941	7.309.398	7.491.351	7.619.787
2018	7.611.452	7.572.069	7.865.509	7.847.428	7.417.647	6.885.495	6.747.084	6.894.849	7.030.951	7.430.572	7.573.780	7.753.769
2019	8.147.232	8.208.436	8.064.555	7.803.529	7.771.927	7.116.947	6.914.766	6.951.917	7.253.908	7.756.789	8.117.505	8.065.079
2020	8.042.107	8.018.771										

Para a realização do estudo foram utilizados como base o capítulo 9 do livro “Análise de Séries Temporais: Os Modelos de Amortecimento Exponencial”, Reinaldo Souza de Castro e Fernando Luiz Cyrino de Oliveira.

Para a construção dos modelos de séries temporais foi utilizada a linguagem R.

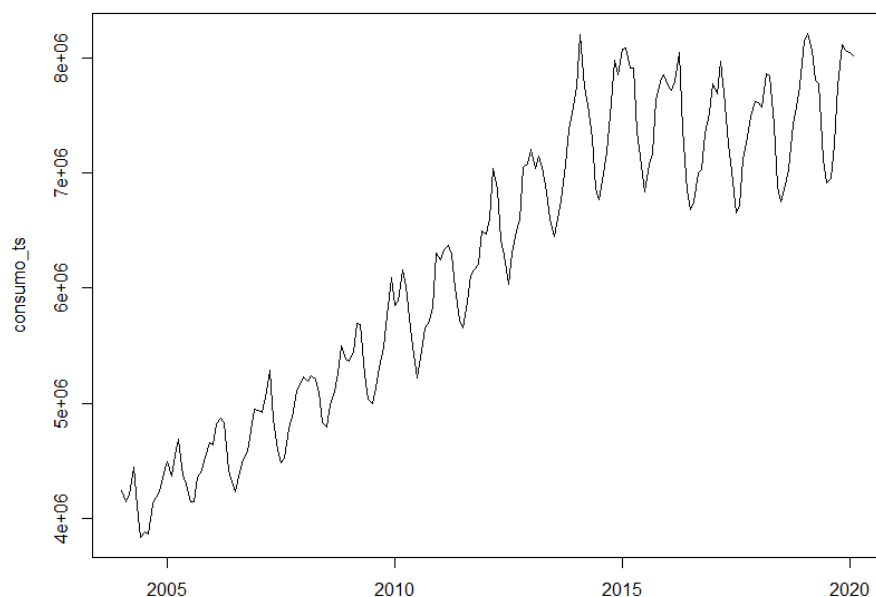


Figura 1 – Gráfico de linhas da variável consumo de energia elétrica da rede em MWh.

Estudo de caso: Consumo de Energia Total – Comercial Brasil

2. Decomposição da série temporal

Podemos inferir sobre três componentes dos dados.

- 1) Uma componente linear de crescimento dos dados;
- 2) Uma componente sazonal nos dados;
- 3) Uma componente aleatória nos dados.

Para confirmar essa hipótese, podemos decompor a série usando o seguinte comando.

```
consumo_ts_c<- decompose(consumo_ts)
```

A nova variável chamada *consumo_ts_c* armazena a decomposição da série temporal, com isso, podemos observar o gráfico das componentes separadas.

```
plot(consumo_ts_c)
```

A função *decompose* tem como argumento uma série temporal e a partir dela separa a série em três componentes: tendência, sazonalidade e ruído, utilizando um filtro de médias móveis.

No modelo aditivo, a equação será: $Y(t)=T(t)+S(t)+R(t)$

Onde,

$Y(t)$: Os dados observados, $T(t)$: Componente de tendência, $S(t)$: Componente de sazonalidade e $R(t)$: Componente aleatória.

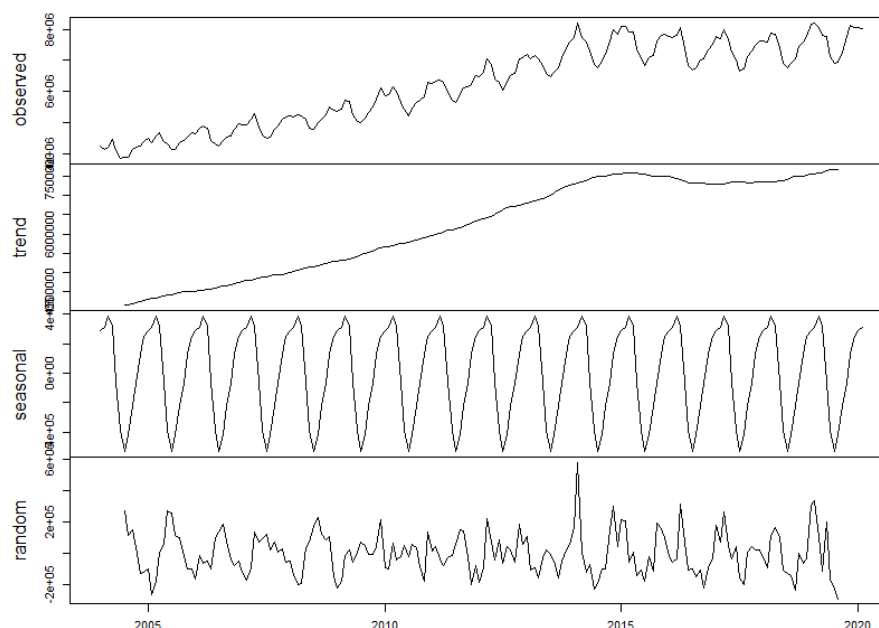


Figura 2 – Gráficos da decomposição da série.

Estudo de caso: Consumo de Energia Total – Comercial Brasil

Podemos observar acima quatro gráficos da decomposição da série temporal. O primeiro contém os valores observados da série. No segundo, observamos a componente da tendência, onde há um crescimento linear ao longo do tempo. No terceiro, a sazonalidade que é característica marcante das séries temporais de consumo de energia, e pôr fim a componente aleatória.

Agora com a série temporal explorada, podemos aplicar modelos de previsão e checar se a série temporal possui ruído branco.

3. Sazonalidade

Sazonalidade é a repetição de um padrão durante um período conhecido, atuando de forma cíclica, como por exemplo, a sazonalidade anual de uma série da média mensal de temperatura atmosférica de um determinado lugar. Para a verificação da sazonalidade foi utilizado o pacote *isSeasonal* disponível na linguagem R, realizando alguns testes disponíveis e todos deram positivo para sazonalidade, como segue:

```
> #Checagem sazonalidade
> isSeasonal(consumo_ts, test = "wo", freq=12) #WO-test
[1] TRUE
> isSeasonal(consumo_ts, test = "qs", freq=12) #QS test
[1] TRUE
> isSeasonal(consumo_ts, test = "fried", freq=12) #Friedman test
[1] TRUE
> isSeasonal(consumo_ts, test = "kw", freq=12) #Kruskall-wallis test
[1] TRUE
> isSeasonal(consumo_ts, test = "seasum", freq=12) #F-test
[1] TRUE
> isSeasonal(consumo_ts, test = "welch", freq=12) #welch test
[1] TRUE
```

4. Ruído Branco

Ruído Branco é amostra aleatória independentemente distribuída (as observações são totalmente descorrelacionadas) com média zero e variância constante. Um bom modelo de séries temporais é capaz de ter como resíduo um ruído branco, que sinaliza a captura da estrutura de autocorrelação dos dados.

A Figura 3 indica um gráfico de tempo dos resíduos, o ACF correspondente e um histograma, indicando a ausência de ruído branco.

Estudo de caso: Consumo de Energia Total – Comercial Brasil

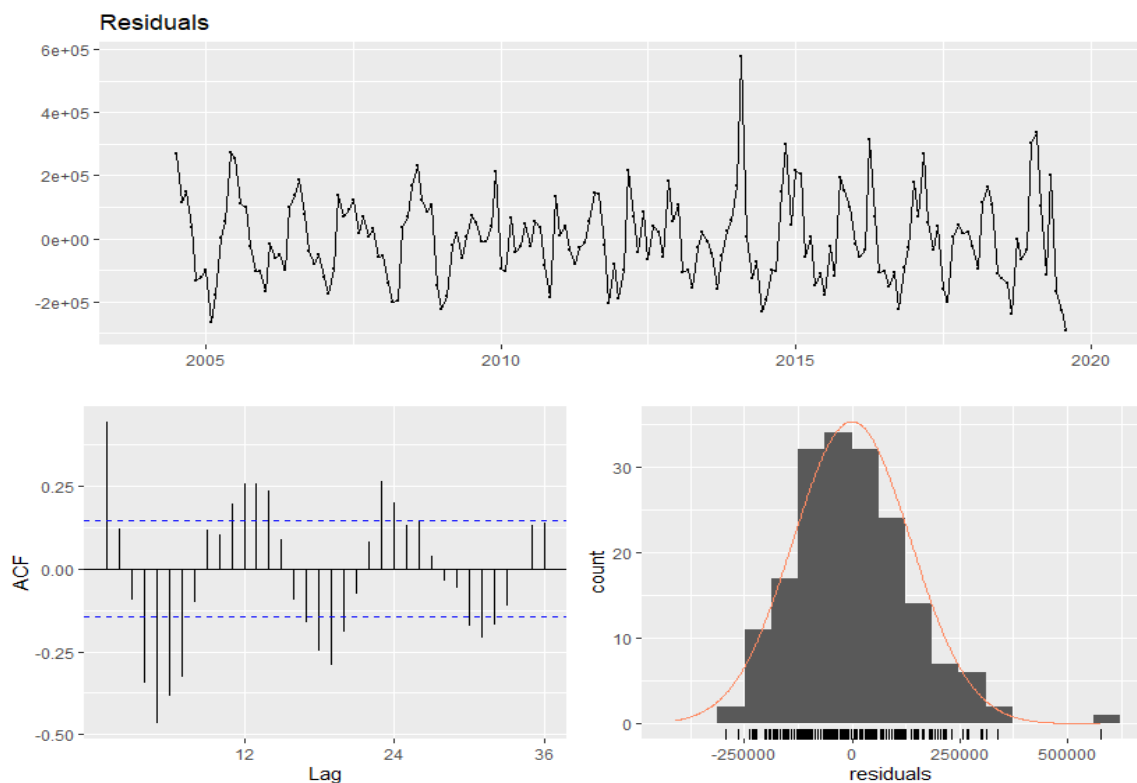


Figura 3 – Resíduos e autocorrelograma e histograma dos resíduos da série temporal em estudo.

Muitos testes estatísticos são usados para tentar rejeitar alguma hipótese nula. O teste de Ljung-Box, função utilizada para verificar a presença de ruído branco, tenta rejeitar a independência de alguns valores. O que isso significa?

Se valor $p < 0,05$: você pode rejeitar a hipótese nula assumindo 5% de chance de cometer um erro. Portanto, você pode assumir que seus valores estão mostrando dependência um do outro.

Se valor $p > 0,05$: você não tem evidência estatística suficiente para rejeitar a hipótese nula. Portanto, você não pode assumir que seus valores são dependentes. Isso pode significar que seus valores são dependentes de qualquer maneira ou pode significar que seus valores são independentes. Mas você não está provando nenhuma possibilidade específica, o que seu teste realmente disse é que você não pode afirmar a dependência dos valores, nem pode afirmar a independência dos valores.

```
Box.test(consumo_ts_c$random, lag = 1, type = "Ljung-Box", fitdf = 0)
> Box.test(consumo_ts_c$random, lag = 1, type = "Ljung-Box", fitdf = 0)

Box-Ljung test

data: consumo_ts_c$random
X-squared = 36.633, df = 1, p-value = 1.426e-09
```

Estudo de caso: Consumo de Energia Total – Comercial Brasil

Como o *p-value* é inferior à 0,05, podemos assumir que os valores estão mostrando dependência um do outro.

5. Aplicação dos Modelos de Previsão

O primeiro passo para isso é criar mais duas variáveis a partir da primeira, uma com a série de dados em que os modelos serão ajustados, (janeiro de 2004 a fevereiro de 2019) e a outra com os dados para comparação (março de 2019 a fevereiro de 2020). Essa operação pode ser facilmente feita manipulando a variável *consumo_ts* através dos índices de seus valores numéricos.

```
consumo_in_sample<-ts(Consumo, frequency=12, start=c(2004,1),end=c(2019,2))
consumo_out_of_sample<-consumo_ts[183:194]
```

Agora pode ser iniciado o processo de previsão com a função *HoltWinters*.

```
consumo1_n <-HoltWinters(consumo_in_sample, gamma = FALSE, beta = FALSE) #naive
> consumo1_n #impressão dos resultados
Holt-winters exponential smoothing without trend and without seasonal component.

Call:
Holtwinters(x = consumo_in_sample, beta = FALSE, gamma = FALSE)

Smoothing parameters:
alpha: 0.9999339
beta : FALSE
gamma: FALSE

Coefficients:
      [,1]
a 8208432
```

Acima temos a saída no console da variável *consumo1_n*. Percebe-se que o único hiperparâmetro de amortecimento é o de nível (alfa) com o coeficiente muito próximo do último valor lido (fevereiro de 2019).

```
consumo1_h <-HoltWinters(consumo_in_sample, gamma = FALSE) #holt
> consumo1_h #impressao dos resultados
Holt-winters exponential smoothing with trend and without seasonal component.

Call:
Holtwinters(x = consumo_in_sample, gamma = FALSE)

Smoothing parameters:
alpha: 1
beta : 0.03031947
gamma: FALSE

Coefficients:
      [,1]
a 8208436.00
b  29443.39
```

Estudo de caso: Consumo de Energia Total – Comercial Brasil

Acima temos a saída no console da variável `consumo1_h`. Percebe-se que os hiperparâmetros de amortecimento são o nível (alfa) e a tendência (beta).

```
consumo1_hwa <-HoltWinters(consumo_in_sample, seasonal=c("additive")) #holt-
winters aditivo
> consumo1_hwa #impressão dos resultados
Holt-winters exponential smoothing with trend and additive seasonal component.

Call:
Holtwinters(x = consumo_in_sample, seasonal = c("additive"))

Smoothing parameters:
  alpha: 0.3642203
  beta : 0
  gamma: 0.6391154

Coefficients:
      [,1]
a    7730100.58
b      23977.33
s1    500763.61
s2    381351.51
s3   -136746.60
s4   -592117.52
s5   -790339.36
s6   -651486.39
s7   -408758.70
s8    -92275.06
s9    105493.60
s10   268203.29
s11   466934.79
s12   423878.61
```

Acima temos a saída no console da variável `consumo1_hwa`. Agora com um modelo de amortecimento exponencial com tendência e sazonalidade aditiva, temos 3 hiperparâmetros (alfa, beta e gamma) e 14 coeficientes: 1 de nível, 1 de tendência e 12 de sazonalidade (pois existem 12 observações em 1 período sazonal).

```
consumo1_hwm <-HoltWinters(consumo_in_sample, seasonal=c("multiplicative"))
#holt-winters mutiplicativo
```

Estudo de caso: Consumo de Energia Total – Comercial Brasil

```
> consumo1_hwm #impressão dos resultados
Holt-winters exponential smoothing with trend and multiplicative seasonal component.

Call:
Holtwinters(x = consumo_in_sample, seasonal = c("multiplicative"))

Smoothing parameters:
  alpha: 0.4095534
  beta : 0
  gamma: 0.453129

Coefficients:
      [,1]
a  7.718042e+06
b  2.397733e+04
s1  1.060760e+00
s2  1.046595e+00
s3  9.801210e-01
s4  9.261270e-01
s5  9.016978e-01
s6  9.234805e-01
s7  9.567909e-01
s8  9.948750e-01
s9  1.022483e+00
s10 1.041083e+00
s11 1.059672e+00
s12 1.053148e+00
```

E por fim a saída do modelo de amortecimento exponencial com tendência e sazonalidade multiplicativa, com 3 hiperparâmetros (alfa, beta e gamma) e 14 coeficientes: 1 de nível, 1 de tendência e 12 de sazonalidade

A seguir, os três modelos construídos são usados para gerar previsões 12 passos à frente com a função *predict*.

6. Previsão 12 passos à frente

Cada variável recebe as previsões 12 passos à frente. Os argumentos da função *predict* são os modelos construídos e o número de previsões a serem feitas. Segue:

```
consumo1_n_f <- predict(consumo1_n, n.ahead=12)
consumo1_h_f <- predict(consumo1_h, n.ahead=12)
consumo1_hwa_f <- predict(consumo1_hwa, n.ahead=12)
consumo1_hwm_f <- predict(consumo1_hwm, n.ahead=12)
```

O gráfico da previsão da série é realizada a partir dos seguintes comandos:

```
ts.plot(consumo_out_of_sample , consumo1_n_f, consumo1_h_f , consumo1_hwa_f ,
consumo1_hwm_f, main="Previsão da série", xlab="Tempo", ylab="Consumo em
Mwh",col=rainbow(6))
legend("bottomright", inset=-.00,c("Dados","Naive","Holt","Holt-Winters
Aditivo","Holt-Winters Multiplicativo"), fill=rainbow(6),cex=0.65)
```

A inferência dos dados a partir dos modelos citados anteriormente são plotados na Figura 4 e dispostos na Tabela 2 a seguir:

Estudo de caso: Consumo de Energia Total – Comercial Brasil

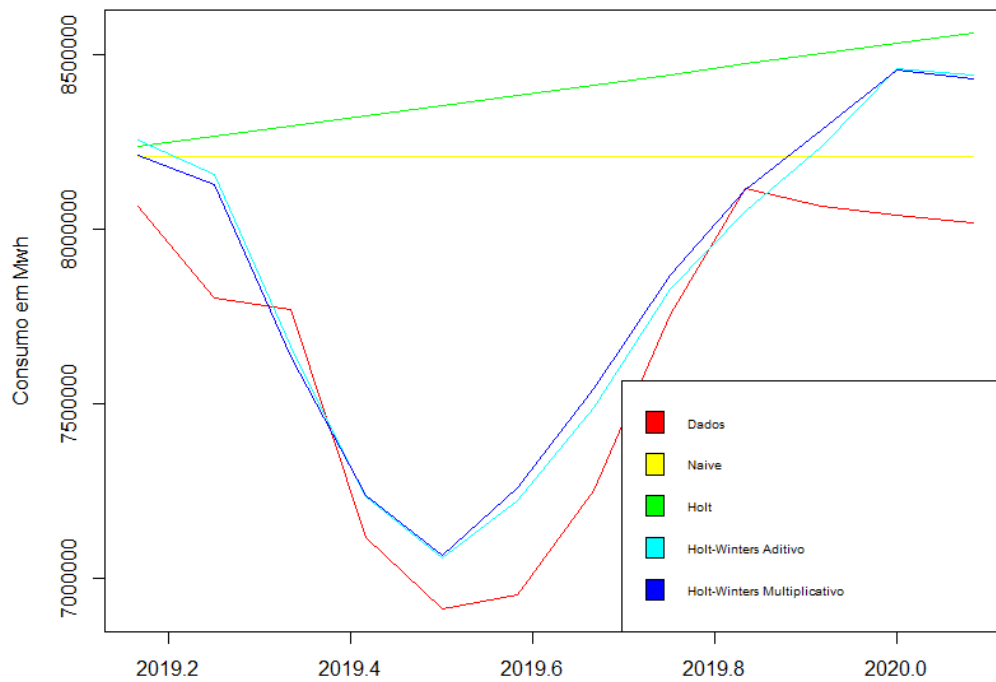


Figura 4 – Gráficos da previsão dos 4 modelos e dados de origem.

Tabela 2 – Comparação numérica dos modelos de previsão.

Ano	Meses	Observado	Naive	Holt	Holt_Winters Aditivo	Holt_Winters_ Multiplicativo
2019	março	8.064.555	8.208.432	8.237.879	8.254.842	8.212.426
2019	abril	7.803.529	8.208.432	8.267.323	8.159.407	8.127.850
2019	maio	7.771.927	8.208.432	8.296.766	7.665.286	7.635.117
2019	junho	7.116.947	8.208.432	8.326.210	7.233.892	7.236.711
2019	julho	6.914.766	8.208.432	8.355.653	7.059.648	7.067.443
2019	agosto	6.951.917	8.208.432	8.385.096	7.222.478	7.260.317
2019	setembro	7.253.908	8.208.432	8.414.540	7.489.183	7.545.141
2019	outubro	7.756.789	8.208.432	8.443.983	7.829.644	7.869.323
2019	novembro	8.117.505	8.208.432	8.473.427	8.051.390	8.112.214
2019	dezembro	8.065.079	8.208.432	8.502.870	8.238.077	8.284.745
2020	janeiro	8.042.107	8.208.432	8.532.313	8.460.786	8.458.085
2020	fevereiro	8.018.771	8.208.432	8.561.757	8.441.707	8.431.263

7. Cálculo do MAPE

O MAPE é calculado pela diferença entre os valores estimados (previsões 1 passo à frente) e os valores reais. É usado tanto para a avaliação in-sample como out-of-sample, e é uma das métricas mais utilizadas na literatura de métodos de previsão, pois fornece um resultado em percentual. O cálculo do MAPE é dado por:

$$MAPE = \frac{1}{N} \sum_{t=1}^N \left| \frac{Z_t - \hat{Z}_t}{Z_t} \right| \times 100$$

Estudo de caso: Consumo de Energia Total – Comercial Brasil

Z_t é o valor da série temporal no período t ;

\hat{Z}_t é o valor ajustado da série temporal para o período t . No caso do MAPE in-sample, é o resíduo;

N é o total de observações ajustadas.

A partir das métricas citadas, obtemos os dados dispostos na Tabela 3:

Tabela 3 – Comparação dos modelos de previsão segundo a métrica de avaliação MAPE.

Métricas	Naive	Holt	Holt_Winters_ Aditivo	Holt_Winters_ Multiplicativo
MAPE in-sample (%)	3,07	3,20	1,94	1,83
MAPE out-of-sample (%)	7,59	10,07	2,80	2,89

Como esperado, é possível observar que o método de Holt-Winters obteve taxas de erro mais baixas em todas as métricas. Isso ocorre porque o método de Holt-Winters é mais robusto e lida melhor com dados que possuem sazonalidade e tendência, que é o caso dos dados utilizados. Observamos que o modelo aditivo obteve resultados ligeiramente melhores, com diferença no MAPE de 0,09%.

8. Código na Linguagem R

#Pacotes utilizados

library(tseries)

library(forecast)

library(seastests)

library(stats)

#Lendo o Banco de Dados

Consumo <- read.delim2("C:/Users/leona/Desktop/Mestrado/1º semestre/Series temporais/Trabalho Computacional/Consumo Total.txt", header=FALSE)

#Construindo as séries

consumo_ts <- ts(Consumo, frequency=12, start=c(2004,1))

#Checagem sazonalidade

isSeasonal(consumo_ts, test = "wo", freq=12) #WO-test

isSeasonal(consumo_ts, test = "qs", freq=12) #QS test

isSeasonal(consumo_ts, test = "fried", freq=12) #Friedman test

isSeasonal(consumo_ts, test = "kw", freq=12) #Kruskall-Wallis test

isSeasonal(consumo_ts, test = "seasum", freq=12) #F-test

isSeasonal(consumo_ts, test = "welch", freq=12) #Welch test

#Plotando o gráfico da Série

ts.plot(consumo_ts)

getOption("scipen")

Estudo de caso: Consumo de Energia Total – Comercial Brasil

```

opt <- options("scipen" = 20)
ts.plot(consumo_ts, main="Consumo de energia elétrica na rede", xlab="Tempo",
ylab="Consumo em MWh")
options(opt)

#Decomposição
consumo_ts_c<- decompose(consumo_ts)
plot(consumo_ts_c)

#Checagem de ruído branco
fit <- consumo_ts_c$random
checkresiduals(fit)
Box.test(consumo_ts_c$random, lag = 1, type = "Ljung-Box", fitdf = 0)

#Criando Variáveis
consumo_in_sample<-ts(Consumo, frequency=12, start=c(2004,1),end=c(2019,2))
consumo_out_of_sample<-consumo_ts[183:194]

#Métodos de Previsão
consumo1_n <-HoltWinters(consumo_in_sample, gamma = FALSE, beta = FALSE)
#naive
consumo1_h <-HoltWinters(consumo_in_sample, gamma = FALSE) #holt
consumo1_hwa <-HoltWinters(consumo_in_sample, seasonal=c("additive")) #holt-
winters aditivo
consumo1_hwm <-HoltWinters(consumo_in_sample, seasonal=c("multiplicative"))
#holt-winters mutiplicativo

#Previsão 12 Passos a Frente
consumo1_n_f <- predict(consumo1_n, n.ahead =12)
consumo1_h_f <- predict(consumo1_h, n.ahead =12)
consumo1_hwa_f <- predict(consumo1_hwa, n.ahead =12)
consumo1_hwm_f <- predict(consumo1_hwm, n.ahead =12)

#Gráfico previsão da série
ts.plot(consumo_out_of_sample , consumo1_n_f, consumo1_h_f, consumo1_hwa_f,
consumo1_hwm_f, main="Previsão da série", xlab="Tempo", ylab="Consumo em
Mwh",col=rainbow(6))
legend("bottomright", inset=-.00,c("Observado", "Naive", "Holt", "Holt-Winters
Aditivo", "Holt-Winters Multiplicativo"), fill=rainbow(6),cex=0.65)

#Organização dos dados e exportação
meses = c("março", "abril", "maio", "junho", "julho", "agosto", "setembro", "outubro",
"novembro", "dezembro", "janeiro", "fevereiro")
conversao=data.frame(Dados = consumo_out_of_sample, Naive = consumo1_n_f, Holt
= consumo1_h_f, Holt_Winters_Aditivo = consumo1_hwa_f,
Holt_Winters_Multiplicativo = consumo1_hwm_f)

```

Estudo de caso: Consumo de Energia Total – Comercial Brasil

```
dados = data.frame(Ano = c(rep(2019,10),rep(2020,2)), Meses=meses, Observado =
conversao[,1], Naive = conversao[,2], Holt = conversao[,3], Holt_Winters_Aditivo =
conversao[,4], Holt_Winters_Multiplicativo = conversao[,5])
write.csv2(dados, "resultados.csv")
```

#Criando função MAPE e cálculo do MAPE p/ todos os modelos - OUT OF SAMPLE

```
mape <- function(actual,pred){
  mape <- mean(abs((actual - pred)/actual))*100
  return (mape)
}
```

#Cálculo do MAPE OUT OF SAMPLE

```
naive_mape=mape(dados[,3],dados[,4])
holt_mape=mape(dados[,3],dados[,5])
holt_winters_aditivo_mape=mape(dados[,3],dados[,6])
holt_winters_multiplicativo_mape=mape(dados[,3],dados[,7])
```

#Criando função MAPE e cálculo do MAPE p/ todos os modelos - IN SAMPLE

```
MAPE2 <- function(observado,modelo){
  n <- length(observado)
  mape0 <- (sum(abs((residuals(modelo))/observado)))*100
  mape1 <- (1/n)*mape0
  return(mape1)
}
```

#Cálculo do MAPE In-sample

```
naive_mape2=MAPE2(consumo_in_sample[1:(length(consumo_in_sample)-
1)],consumo1_n)
holt_mape2=MAPE2(consumo_in_sample[1:(length(consumo_in_sample)-
2)],consumo1_h)
holt_winters_aditivo_mape2=MAPE2(consumo_in_sample[1:(length(consumo_in_sam
ple)-12)],consumo1_hwa)
holt_winters_multiplicativo_mape2=MAPE2(consumo_in_sample[1:(length(consumo_i
n_sample)-12)],consumo1_hwm)
```

#Organização dos resultados das métricas e exportação

```
res_metricas=data.frame(metricas=c("MAPE IN SAMPLE","MAPE OUT OF SAMPLE"),
Naive=c(naive_mape2, naive_mape), Holt=c(holt_mape2, holt_mape),
Holt_Winters_Aditivo=c(holt_winters_aditivo_mape2,holt_winters_aditivo_mape),Holt
_Winters_Multiplicativo=c(holt_winters_multiplicativo_mape2,holt_winters_multiplica
tivo_mape))
res_metricas
write.csv2(res_metricas, "metricas.csv")
```