Atividade Final de Estatística

Matéria: Inferência Estatística

Equipe :

Spacial

Brasil

Pós-Graduação em *DataScience*

Sumário

1 Atividade Inferência e Hipótese

1.1 Informações Gerais

Informe o método utilizado e justifique. A formulação da resposta faz parte da avaliação. Exibir o código utilizado no R.

1.2 Exercícios

1. A empresa fictícia TI Systems utiliza o tempo de ponto de função para estimar o custo de um sistema.
   É considerado 2 horas como o custo de mão de obra por medida. Sabe-se que o desvio padrão é de 0,5 hora.
   No mês passado os tempos por ponto de função coletados foram de:

   `1,9 1,7 2,8 2,4 2,6 2,5 2,8 3,2 1,6 2,5`

   Usando p=0,05, verifique se o custo excede 2 horas.
   Pergunta: Qual é sua conclusão e que recomendações você consideraria fazer aos gerentes?

   - **Código**:

   ```
   amostra <- c(1.9, 1.7, 2.8, 2.4, 2.6, 2.5, 2.8, 3.2, 1.6, 2.5)
   sigma <- sd(amostra)
   N <- length(amostra)
   ```



```
med <- mean(amostra)
conf <- 0.95
gl <- N -1
Tc <- qt(0.975,lower.tail = T, df= gl)
IC <- c(med - Tc * sigma/sqrt(N),med + Tc* sigma/sqrt(N))
IC
```

- **Resposta**:
  Com o intervalo de confiança de 95% ficando entre: `2.0306 <-> 2.7694`, tem-se que o custo real está maior que o custo estimado (*2 horas*). Com isso sugere-se que o cálculo de custo de pontos por função seja alterado para mais próximo da média (e aí o apetite ao risco dos mesmo) ou que seja melhorada a mão de obra (para executar os pontos de função com menor quantidade de horas).

2. Uma indústria testa a aplicação de uma nova liga metálica na fabricação de seus produtos. Analisa-se a resistência de 7 linhas de produtos distintas. **Há aumento da resistência dos produtos ao adotar o metal 2?** Utilize 90% de confiança.

Marca	Metal 1	Metal 2
A	68	61
B	75	69
C	62	64
D	86	76
E	52	52
F	46	38
G	72	68

- **Código**:

```
# amostras
metal1 = c(68,75,62,86,52,46,72)
metal2 = c(61,69,64,76,52,38,68)
# tamanho de N   igual, confian a e graus de liberdade
N <- length(metal1)
conf <- 0.90
gl <- N-1
# amostra metal 1
med1 <- mean(metal1)
sigma1 <- sd(metal1)
Tc1 <- qt(0.95,lower.tail=T, df=gl)
IC1 <- c(med1 - Tc1*sigma1/sqrt(N),
```

```
          med1 + Tc1*sigma1/sqrt(N))
IC1
# amostra metal 2
med2 <- mean(metal2)
sigma2 <- sd(metal2)
Tc2 <- qt(0.95,lower.tail=T, df=gl)
IC2 <- c(med2 - Tc2*sigma2/sqrt(N),
          med2 + Tc2*sigma2/sqrt(N))
IC2
```

- **Resposta**:
  Conforme observa-se, os intervalos de confiança (com 90%) das amostras são: *metal 1* `55.7652 <-> 75.9491` e *metal 2* `51.8679 <-> 70.4178`. Conclui-se que não há aumento da resistência dos produtos ao adotar o **metal 2**, já que o intervalo mostrou que a resistência ficou menor.

3. Uma linha de produção tem os seguintes pesos em *kg*:

```
5.4, 4.5, 4.7, 4.0, 3.9, 5.3, 5.4, 5.1, 5.9, 7.1, 4.5, 2.7,
6.0, 4.3, 4.3, 6.0, 4.7, 3.8, 5.2, 4.9, 5.0, 5.4, 4.6, 5.6,
4.8, 5.3, 6.1, 5.4, 4.7, 6.1, 6.0, 5.5, 5.2, 4.4, 6.4, 4.4,
7.2, 6.5, 4.8, 4.0
```

A) Mediana:

- **Código**:

```
prod <- c(5.4, 4.5, 4.7, 4.0, 3.9, 5.3, 5.4,
5.1, 5.9, 7.1, 4.5, 2.7, 6.0, 4.3, 4.3, 6.0,
4.7, 3.8, 5.2, 4.9, 5.0, 5.4, 4.6, 5.6, 4.8,
5.3, 6.1, 5.4, 4.7, 6.1, 6.0, 5.5, 5.2, 4.4,
6.4, 4.4, 7.2, 6.5, 4.8, 4.0)
summary(prod)
```

- **Resposta**: 5.150

B) Média:

- **Código**:

```
summary(prod)
```

- **Resposta**: 5.128

C) Desvio Padrão:

- **Código**:

```
sd(prod)
```

- **Resposta**: 0.9229

D) Calcule o 1o quartil, 2o quartil e 3o quartil:

- **Código**:

```
summary(prod)
```

- **Resposta**: $1^oQ : 4.5$ / $2^oQ : 5.15$ / $3^oQ : 5.675$

E) Plote o *boxplot* (diagrama de caixa) da amostra:

- **Código**:

```
boxplot(prod)
```

- **Resposta**:

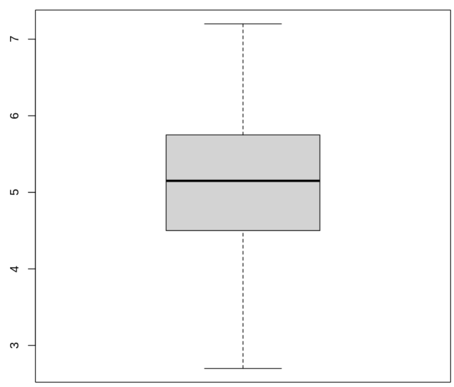

Figura 1 – BoxPlot.

F) Plote o histograma da amostra:

- **Código**:

```
hist(prod)
```

- **Resposta**:

G) qual a probabilidade um produto ter um peso entre 4.2 e 5.2?

- **Código**:

```
# p > 4.2 e p < 5.2
Z4 <- (4.2-mean(prod))/sd(prod)
```

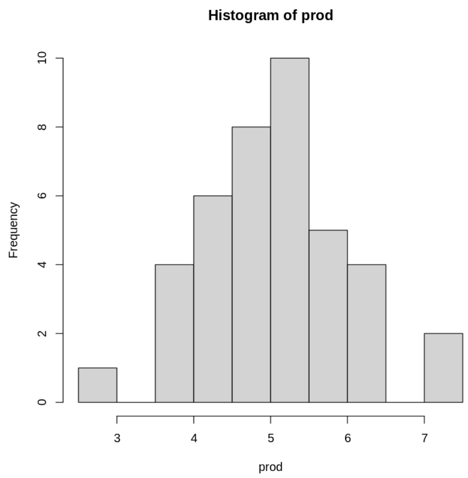

Figura 2 – Histrograma.

```
Z5 <- (5.2-mean(prod))/sd(prod)
prob <- pnorm(Z5) - pnorm(Z4)
round(prob,4)
```

– **Resposta**: A probabilidade de ficar entre 4.2 e 5.2 é de 37.38%.

2 Atividade Regressões Lineares

2.1 Informações Gerais

Informe o método utilizado e justifique. A formulação da resposta faz parte da avaliação. Exibir o código utilizado no R.

2.2 Exercícios

2.2.1 Exercício 1

1. Faça a análise conforme descrito a seguir:

   1.1 Defina a renda média *per-capita* do estado em relação a média de escolaridade do estado (y = renda, x = escolaridade) em outras palavras renda ∼ escolaridade) dos dados públicos a seguir:

```
#dados para o exercicio copie e cole no R

mec <-  data.frame(
row.names   =    c("RR", "AC", "PA", "TO", "MA", "SE", "BA",
"AL", "SP", "ES", "SC", "PR", "GO", "DF", "AP", "RO", "AM",
```

```
"PB", "RN", "PI", "PE", "CE", "RJ", "MG", "RS", "MT", "MS"),

escolaridade = c(5.7, 4.5, 4.7, 4.5, 3.6, 4.3, 4.1, 3.7, 6.8,
5.7, 6.3, 6.0, 5.5, 8.2, 6.0, 4.9, 5.5, 3.9, 4.5, 3.5, 4.6,
4.0, 7.1, 5.4, 6.4, 5.4, 5.7),

renda =  c(685, 526, 536, 520, 343, 462, 460, 454, 1076, 722,
814, 782, 689, 1499, 683, 662, 627, 423, 513, 383, 517, 448,
970, 681, 800, 775, 731)
)
```

1.2 Veja os gráficos de dispersão: Figura 3.

Código:

```
ggplot(mec,aes(x=escolaridade, y=renda,
               color=(renda/escolaridade))) +
geom_point(shape = 16, size = 5, show.legend = FALSE) +
theme_minimal()
```

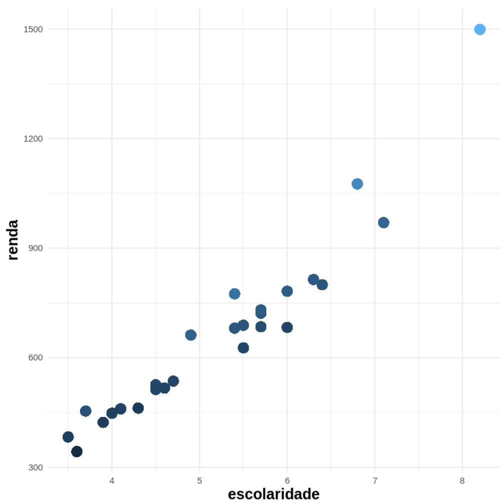

Figura 3 – Gráfico de Dispersão.

1.3 Exiba as correlações:

Código:

```
correlacao <- cor(escolaridade,renda)
```

Resposta: correlação direta forte: 0.9507

1.4 Plote os histogramas de renda e escolaridade:

Código:


```
par(
  mfrow=c(1,2),
  mar=c(4,4,1,0)
)
hist(mec$escolaridade, breaks=5 , ylim=c(0,12),
      col="#e6b9b3" , xlab="escolaridade" ,
      ylab="", main="")
hist(mec$renda, breaks=5, ylim=c(0,12),
       xlab="renda" , ylab="", main="",
        col="#b3cce6")
```

Resposta: Figura 4

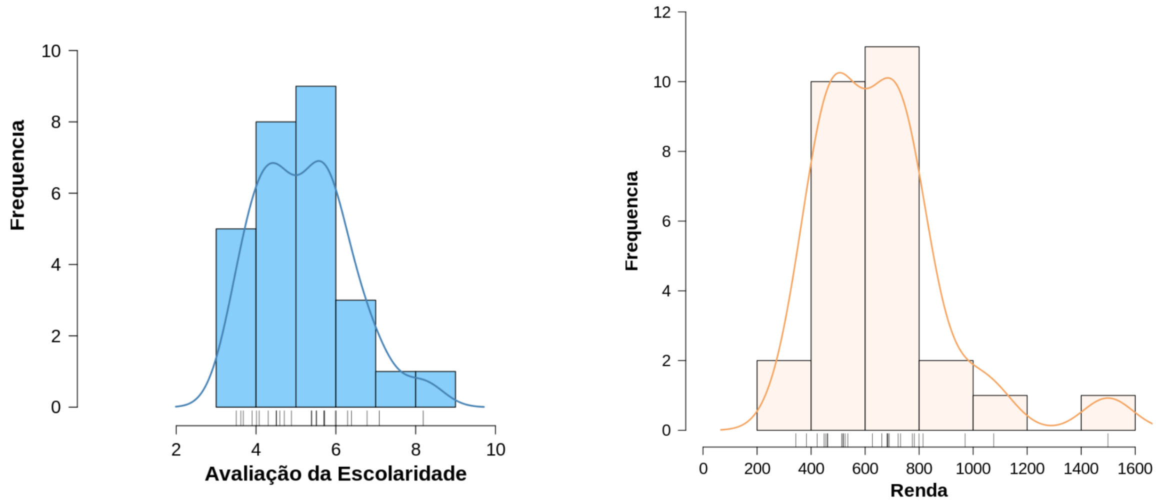

Figura 4 – Histogramas das variáveis com a linha de dispersão.

1.5 Teste de normalidade:

Código:

```
shapiro.test(renda)$p.value
shapiro.test(escolaridade)$p.value
fligner.test(renda ~ escolaridade, mec)
```

Resposta: No caso do teste de *Shapiro* em cada conjunto de dados (renda e escolaridade), o primeiro ficou menor que *0.05*, mostrando que os dados não seguem uma distribuição normal. Já no caso da escolaridade, os dados seguem.
No teste de *Fligner*, o valor foi `p-value = 0.2153`, nos indicando que há igualdade entre as variâncias.

1.6 Faça a regressão linear `lm()`:

Código:

```
modelo.linear <- lm (escolaridade ~ renda, mec)
coefficients(modelo.linear)
ggplot(mec,aes(x=renda, y=escolaridade,
          color=(renda/escolaridade)),
          show.legend = FALSE) +
       geom_point(shape = 16, size = 5,
           show.legend = FALSE) +
       theme_minimal() +
       geom_smooth(method = "lm", se = FALSE,
       show.legend = FALSE)
```

Resposta: Figura 5

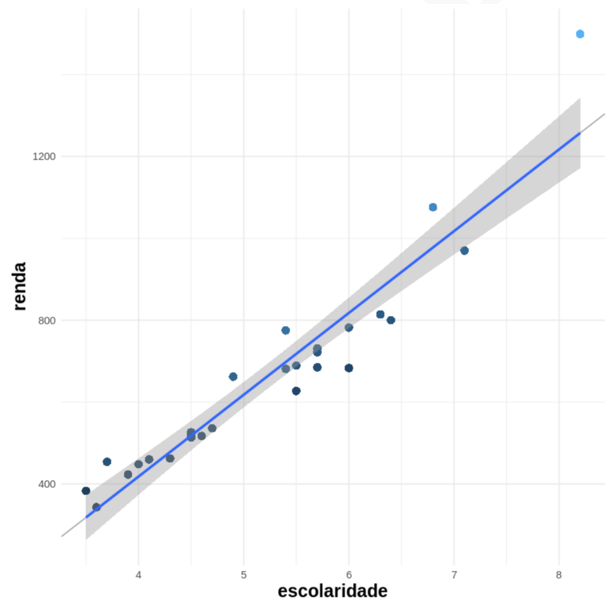

Figura 5 – Regressão Linear Simples.

1.7 Quais são os pontos com maior alavancagem?

Código:

```
sort(influence(modelo.linear)$hat, decreasing = TRUE)[1:4]
```

Resposta: `DF`, `RJ`, `PI` e `MA`.

1.8 Qual o coeficiente de determinação (R-squared)?

Código:

```
summary(modelo.linear)$r.squared
```

Resposta: Coeficiente de determinação: 0.9039, isso nos indica que o modelo (a reta) explica 90% da variação dos dados.

1.9 Verifique os resíduos com a biblioteca `library(hnp)`
Código:

```
fit <- hnp(modelo.linear, print.on=TRUE, plot=FALSE)
plot(fit)
```

Resposta: Figura 6.

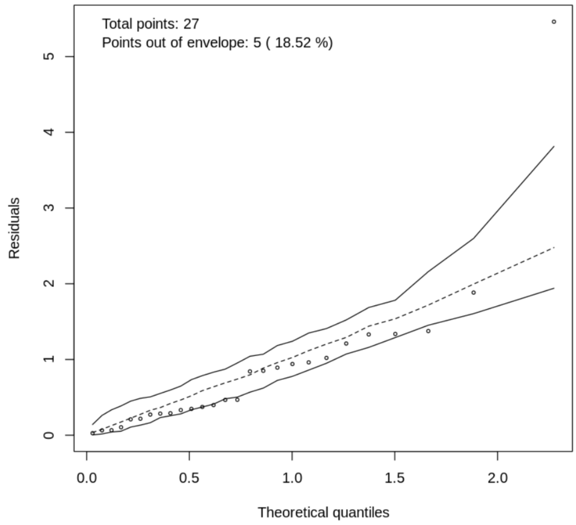

Figura 6 – Resíduos via biblioteca `hnp` aplicada ao modelo linear.

1.10 A regressão linear parece ser uma boa escolha? Por que?
Resposta: Como visto na Figura 6, o modelo linear deixa 5 pontos fora do intervalo (18.5%). Ideal seriam que todos os pontos estivessem dentro do intervalo (mesmo `DF` que é um *outlier*).

1.11 Qual das distribuições que estudamos (Binomial, Normal, Poisson, Gama, Gaussiana Inversa) tem uma semelhança com os dados mostrados pelo histograma de renda?
Resposta: *Gamma*

1.12 Faça uma regressão `glm()` com essa distribuição
Código:

```
modelo.glm <- glm(renda ~ escolaridade, data=mec,
                        family = Gamma() )
ggplot(mec,aes(x=escolaridade, y=renda, bins=4,
        color=abs(renda-modelo.linear$fitted.values)),
        breaks=1:8, ylim=c(200,1600), xlim=c(2,9),
```

```
        show.legend = FALSE) +
  geom_smooth(method = "glm", show.legend = FALSE) +
  geom_point(shape = 16, size = 4,
                show.legend = FALSE, bins=5) +
  geom_point(aes(x= escolaridade,
                   y=modelo.linear$fitted.values),
                   col = "firebrick4", size=3,
                   pch = 18, alpha=0.5)+
  geom_abline(intercept = reta[1], slope=reta[2],
                    col="goldenrod3") +
  geom_segment(aes(xend = escolaridade,
                     yend = modelo.linear$fitted.values),
                     show.legend = FALSE, bins=5) +
  theme_minimal() +
  theme(axis.title = element_text(color="black",
          size=15, face=2))+
  scale_color_viridis() +
  ggtitle("Modelo GLM")
```

Resposta: Figura 7 (nota[1]):

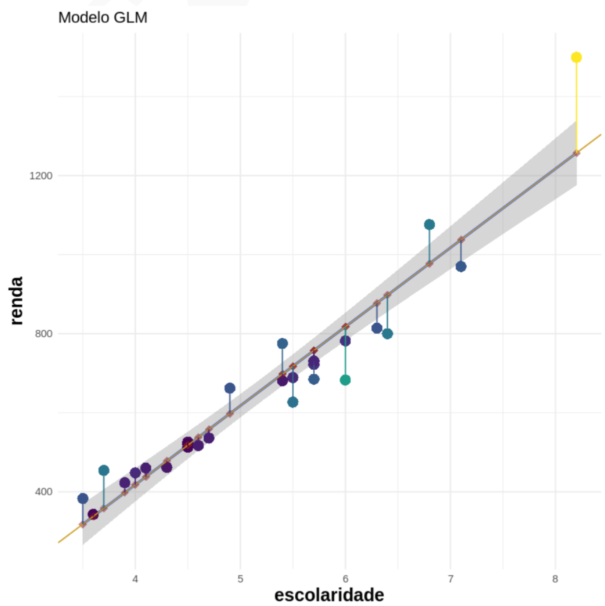

Figura 7 – Modelo `glm` com *Gamma*.

1.13 Agora estime (utilize a função `predict()`) os valores de renda para os valores de escolaridade utilizando os dois modelos (`lm()` e `glm()`) e plote os gráficos com

[1] As cores dos pontos e segmentos representam a diferença entre a reta e valor do ponto em si, sendo o mais distante em amarelo e o mais próximo no roxo. Em vermelho, os pontos projetados.

as curvas. Mostre no mesmo gráfico os valores observados em preto, preditos do modelo 1 em vermelho, e preditos no modelo 2 em verde.(utilize as funçoes `plot()`, `points()` e `points()`)

Código:

```
predicao <- data.frame(escolaridade=seq(from = 0 ,
                                        to = 10 ,
                                        by = 0.25))
predicao$rendalin <- predict(modelo.lin,predicao, type='response')
predicao$rendaglm <- predict(modelo.glm,predicao, type='response')
predicao$rendagll <- predict(modelo.gll,predicao, type='response')
predicao$rendagin <- predict(modelo.gin,predicao, type='response')
predicao$rendagil <- predict(modelo.gil,predicao, type='response')
reta <- coefficients(modelo.lin)
reta <- coefficients(modelo.lin)
ggplot(mec,
       aes(x=escolaridade, y=renda),
       color="Gray",
       ylim=c(200,1600),
       xlim=c(0,9),
       show.legend = FALSE) +
    expand_limits(x=c(0,12),
                  y=c(0, 2000))+
    geom_abline(intercept = reta[1],
                slope=reta[2],
                col="darkgrey") +
    geom_line(aes(x= escolaridade,
                  y=modelo.lin$fitted.values),
              col = "red",
              size=1,
              alpha=0.5) +
    geom_line(aes(x= escolaridade,
                  y=modelo.glm$fitted.values),
              col = "green1",
              size=1,
              alpha=0.5) +
    geom_line(aes(x= escolaridade,
                  y=modelo.gll$fitted.values),
              col = "green4",
              size=1,
```

```
                   alpha=0.5) +
          geom_line(aes(x= escolaridade,
                       y=modelo.gin$fitted.values),
                   col = "yellow2",
                   size=1,
                   alpha=0.5) +
          geom_line(aes(x= escolaridade,
                       y=modelo.gil$fitted.values),
                   col = "yellow4",
                   size=1,
                   alpha=0.5) +
          theme_minimal() +
          theme(axis.title = element_text(color="black",
                                         size=15,
                                         face=2))+
          scale_color_viridis() +
          geom_point(shape = 16,
                    size = 2,
                    show.legend = FALSE,
                    alpha=0.5) +
          ggtitle("Comparativo dos modelos")
```

Resposta: Uma visualização dos dados e modelos é apresentada na Figura 9. Nota: em vermelho a reta representando o modelo linear `lm`, em tons de verde, o `glm()`, sendo o claro do *Gamma()* e o mais escuro o *Gamma(link='log'*. Já os tons de amarelo são a visualizaçao da *gaussiana invertida* (clara) e *gaussiana invertida* com escala em log (escura), para curiosidade e ajudar a ilustrar o uso dos modelos.

1.14 Compare os modelos com a função AIC e informe qual modelo você escolhe e por que?

Código:

```
print(data.frame(
modelos = c('Linear',
            'Gamma', 'Gamma-Log',
            'GaussInv', 'GaussInv-Log'),
aic = c(AIC(modelo.lin),
        AIC(modelo.glm), AIC(modelo.gll),
        AIC(modelo.gin), AIC(modelo.gil)),
verossimilhan a = c(logLik(modelo.lin),
                    logLik(modelo.glm),logLik(modelo.gll),
```

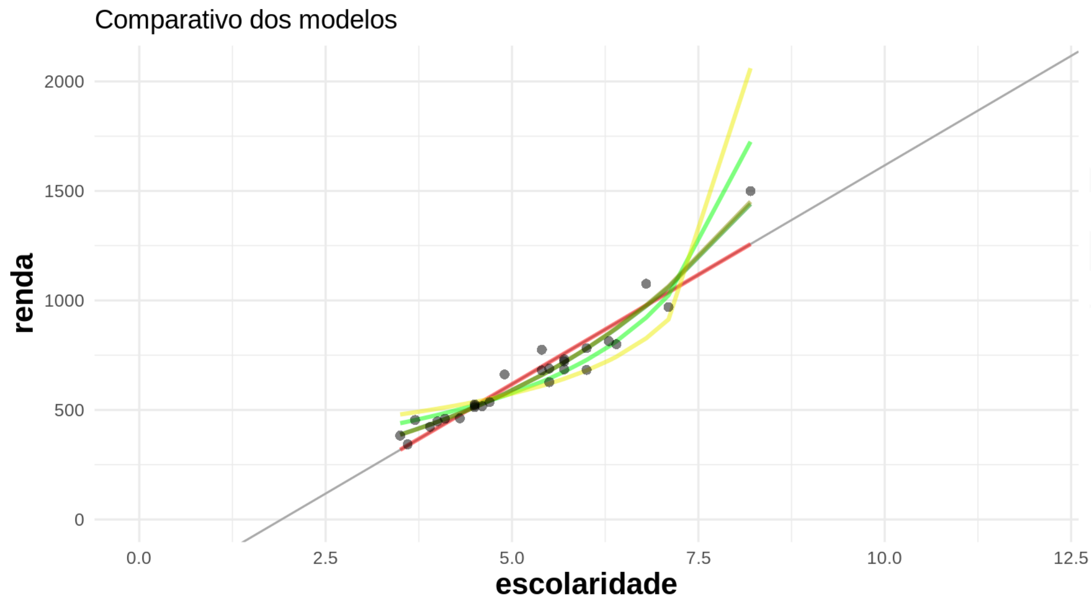

Figura 8 – Comparativo entre os modelos.

```
                    logLik(modelo.gin),logLik(modelo.gil)))
)
# saida
  modelos       aic  verossimilhan a
1        Linear 315.2632        -154.6316
2         Gamma 304.9130        -149.4565
3     Gamma-Log 288.1337        -141.0669
4      GaussInv 326.4785        -160.2392
5 GaussInv-Log 288.9597        -141.4798
```

Resposta: O modelo que escolheria seria o modelo *Gamma-Log*, que melhor se ajusta e com resíduos mais próximos, vide a maior verossimilhança e menor `AIC`. Em caso de não usar os ajustes com *log*, o próprio *Gamma* seria o escolhido (segundo os mesmos critérios). O gráfico da saída do `hnp()` mostra que a *Normal Q-Q* tem os dados mais próximos do eixo e mais concentrado no centro, como mostra de melhor ajuste conforme dados observados.

2.2.2 Exercício 2

2. Encontre uma base de dados de sua preferência, caso não possua alguma há várias disponíveis no `https://www.kaggle.com/datasets` e `http://dados.gov.br/dataset`, e faça uma análise de regressão ou *forecast* sobre alguma informação que lhe pareça importante. Atenção que todas as **análises dos resultados e gráficos devem ser exibidas, co-**

mentadas e descritas abaixo.

Resposta:
A base de dados escolhida foi a da AAVSO[2] (buscada através do portal *Simbad*[3]. O tema de busca foi o caso da estrela Betelgeuse no trimestre final de 2019 e início de 2020.

A estrela Betelgeuse sempre teve o comportamento oscilativo da sua luminância em

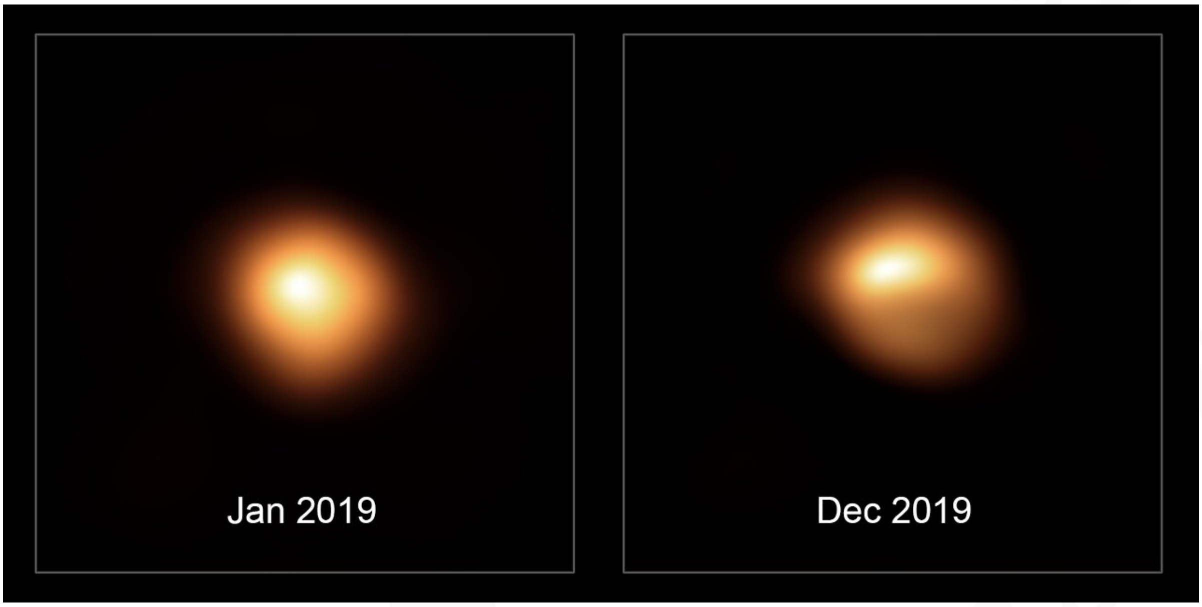

Figura 9 – Fotos do inicio e fim de 2019: Betelgeuse (α *Orinidis*).

ciclos 5.9 anos[4]. No final de 2019, astrônomos Universidade Villanova, viram que esta estava diminuindo a luminância por uma fator maior que o comum[5].
Portanto aí estava um tema interessante para avaliar os dados e testar previsões (nos momentos do ano passado e atual), além de uma previsão geral para o próximo ano.

(a) O primeiro passo foi escolher um corte no tempo (há muitas observações). No site foi selecionado desde 2015. Após descarregado, temos os seguintes passos: analisar a base e separar as informações desejadas. Comandos:
Código:

```
$ wc -l aavsodata_betelgeuse.csv
    9492 aavsodata_betelgeuse.csv
```

[2] *American Association of Variable Star Observers*, url: https://aavso.org.
[3] A busca foi feita na url: https://simbad.u-strasbg.fr/simbad/sim-id?Ident=alpha%20Orionis.
[4] Updates on the "Fainting"of Betelgeuse. url: https://www.astronomerstelegram.org/?read=13365
[5] EVOLUTIONARY TRACKS FOR BETELGEUSE. url: https://iopscience.iop.org/article/10.3847/0004-637X/819/1/7

```
$ head aavso_betelgeuse.csv
1- JD,Magnitude,Uncertainty,HQuncertainty,Band,Observer Code,
Comment Code(s),Comp Star 1,Comp Star 2,Charts,Comments,
Transfomed,Airmass,Validation Flag,Cmag,Kmag,HJD,Star Name,
Observer Affiliation,Measurement Method,Grouping Method,ADS
Reference,Digitizer,Credit
2- 2457023.54375,0.5,,,Vis.,MCPA,U,01,09,10 star,,,,Z,,,,ALF ORI,
AAVSO,STD,,,,
3- 2457023.8680,-3.084,0.050,,J,KCD,,GAMMA ORI,,13650PT,,1,,Z,
,,,ALF ORI,BAA-VSS,STD,1,,,
4- 2457023.8680,-4.076,0.050,,H,KCD,,GAMMA ORI,,13650PT,,1,,Z,
,,,ALF ORI,BAA-VSS,STD,1,,,
5- 2457024.3,0.3,,,Vis.,SPAO,,03,17,680301,,,,Z,,,,ALF ORI,UAI,
STD,,,,
```

(b) O arquivo contém `Magnitude` (observação que nos interessa), além da coluna `JD` (que após pesquisas em sites de astronomia, significa *Julian Date* - representação de datas usada na comunidade astronômica).
Cabe identificar que há muitos dados além do que gostaríamos, nos interessa a banda da luz visível, para deixar uma informação de mesma característica. Assim a coluna `Band` nos ajuda a filtar as informações que queremos. Portanto, para filtrar antes de carregar no `R`, vamos tratar de "limpar" os dados pra facilitar o uso. Assim iremos fazer 3 passos:

i. Pega as primeiras 5 colunas (até o que precisamos, `Band`):

```
# comando pra pegar o cabe alho:
$ head -n 1 aavsodata_original.csv |
                              cut -d',' -f1-2 > aavso_data.csv
# (construindo o comando) pegando as colunas 5 colunas:
$ cut -d',' -f1-5 avsodata_betelgeuse.csv
```

ii. Após isso, vamos filtrar somente as observações visuais (`Vis.`) - e com isso podemos descartar essa coluna:

```
# selecionando somente o que houver Vis:
$ cut -d',' -f1-5 avsodata_betelgeuse.csv | grep 'Vis'
```

iii. Por último teremos somente as duas primeiras colunas (data e magnitude):

```
# do resultado, pega somente a primeiras 2 colunas:
```



```
$ cut -d',' -f1-5 avsodata_betelgeuse.csv | grep 'Vis' |
                                     cut -d',' -f1-2 >> aavso_data.csv
```

Ao final deste passo, ficam-se com 7645 pontos de observação.

3. Com isso, iniciamos os trabalhos no *R*, carregando as bibliotecas e os dados em um *dataframe*:

```
# biblioteca de gr ficos
library(ggplot2)
# bibliotecas para manipula  o de datas e fun  es Astronomicas
library(astrolibR)
library(astroFns)
# manipular dataframes
library(data.table)
aavso_data <- read.csv2("aavso_data.csv", sep=",")
head(aavso_data)
```

4. Após carregar os dados, vamos limpar os dados `NA` (*Not Available*), que por algum motivo afetam a coluna *magnitude*. Vamos aproveitar (pra uso no futuro) e criar uma coluna `dia`, para termos as datas conforme relativas ao primeiro dia da base.

```
Dados_preNA <- data.frame(date=as.Date(jd2ymd(as.double(aavso_data$JD))))
Dados_preNA$magnitude <- as.double(aavso_data$Magnitude)
Dados = Dados_preNA[complete.cases(Dados_preNA),]
Dados$dia <- as.numeric(Dados$date-min(Dados$date))
```

5. Após este passo, ficam 7634 pontos de observação (9 `NA`s foram removidos). Com isso vamos ver um resumo dos dados:

```
summary(Dados)
```

6. Vamos ver como ficam os pontos observados no gráfico:

```
ggplot(Dados,
       aes(x=date,
           y=magnitude,
           color=(magnitude),
           alpha=0.5),
       ylim=c(-10,14)) +

  geom_point(shape = 16,
             size = 2,
```



```
                show.legend = FALSE,
                alpha=0.6) +
  theme_minimal() +
  theme(axis.title = element_text(color="black",
                                   size=15,
                                   face=2))
```

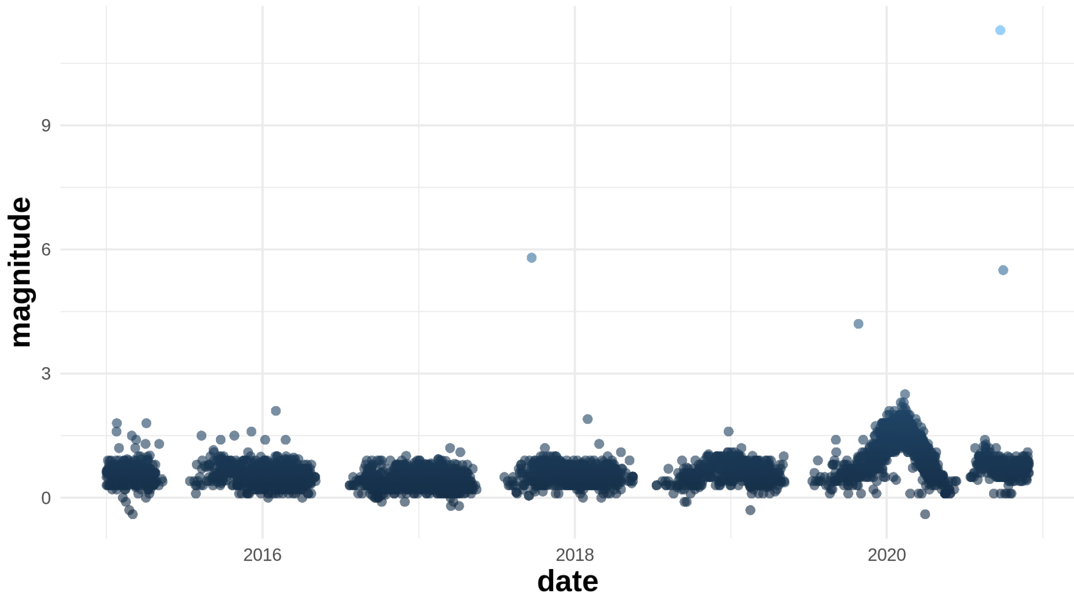

Figura 10 – Observações visuais da Betelgeuse (α *Orinidis*).

7. Na Figura 11 observa-se que há alguns valores discrepantes (que não fazem sentido). Como são 4 pontos, vamos removê-los do data.frame:

```
outlierReplace = function(dataframe, cols, rows, newValue = NA) {
    if (any(rows)) {
        set(dataframe, rows, cols, newValue)
    }
}
outlierReplace(Dados, c("magnitude"), which(Dados$magnitude > 3), NA)
```

Plotando o gráfico sem os 4 pontos discrepantes (com total agora de 7630):

8. Aplicando um modelo linear, temos:

```
# # testando um modelo linear
ggplot(Dados,
       aes(x=date,
           y=magnitude))+
  geom_line(aes(x=date,
```

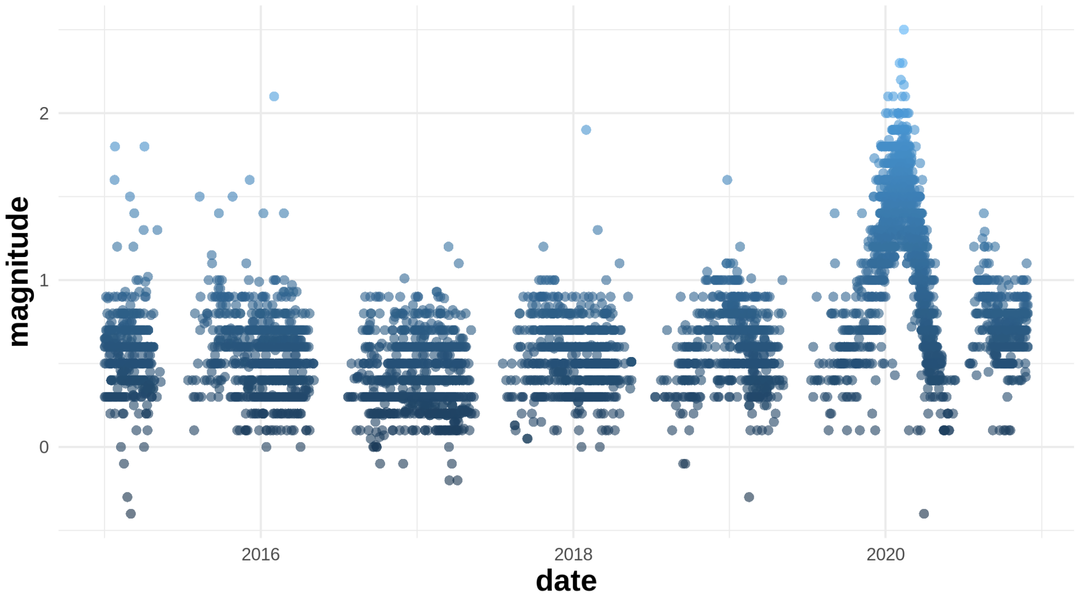

Figura 11 – Observações visuais sem valores discrepantes.

```
            y=magnitude,
            color="lightblue",
            alpha=0.2),
            show.legend = FALSE) +
  geom_smooth(method = "lm",
              se = TRUE,
              show.legend = FALSE) +
  theme_minimal() +
  theme(axis.title = element_text(color="black",
                                  size=15,
                                  face=2))

ggsave("2_2_02.png",
          plot=last_plot(),
       scale = 1:1,
       width=192,
       height=108,
       units="mm",
       dpi=300)
```

9. Como observou-se na Figura 12, o modelo linear é muito reduzido para a características dos dados, ficando muito aquém do que se esperaria. Assim, vale testar a biblioteca `forecast`.

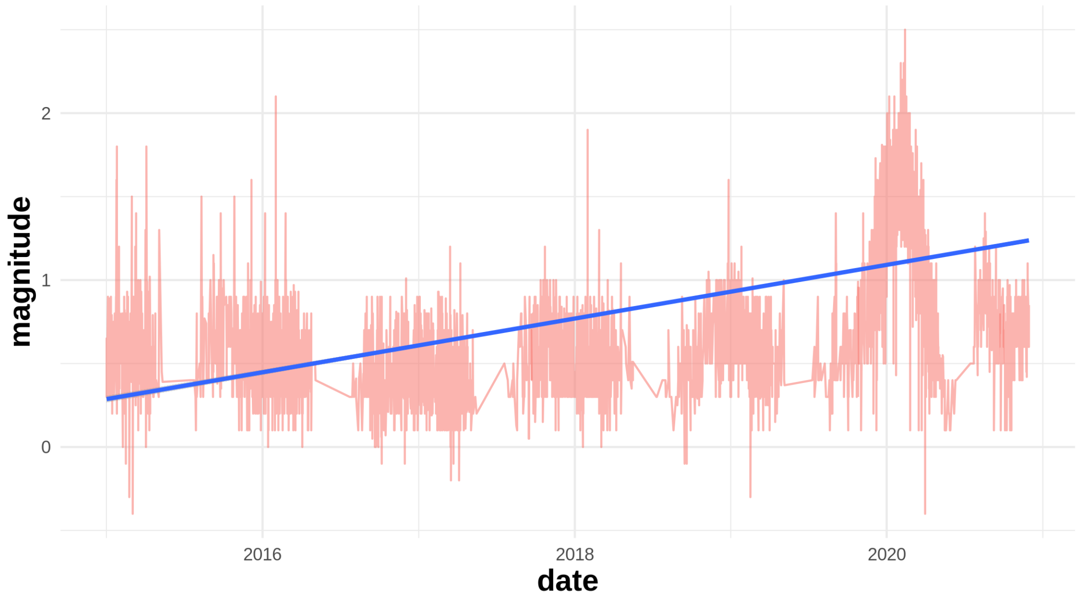

Figura 12 – Modelo Linear.

```
tsData <- ts(data=magnitude,
            frequency=365,
            start=c(2015,1),
            end=c(2020,12))
htt1 <- HoltWinters(tsData,
                    gamma=FALSE,
                    l.start = sbux[1])
plot(htt1)
prev_htt1 <- forecast(htt1, h=60)
plot(prev_htt1)
```

10. Na Figura 13, é mostrada a função de *Holt-Winters* conforme o observado e os dados ajustados. Já na Figura 14, tem-se os dados observados e a previsão para 60 dias.

```
htt2 <- HoltWinters(tsData)
prev_htt2 <- forecast(htt2, h=365)
plot(prev_htt2)
```

11. Observa-se que a previsão pra um ano (Figura 15 é bem mais ajustada ao comportamento dos dados.

```
m <- HoltWinters(tsData)
p <- predict(m, 60, prediction.interval = TRUE)
plot(m,p)
```

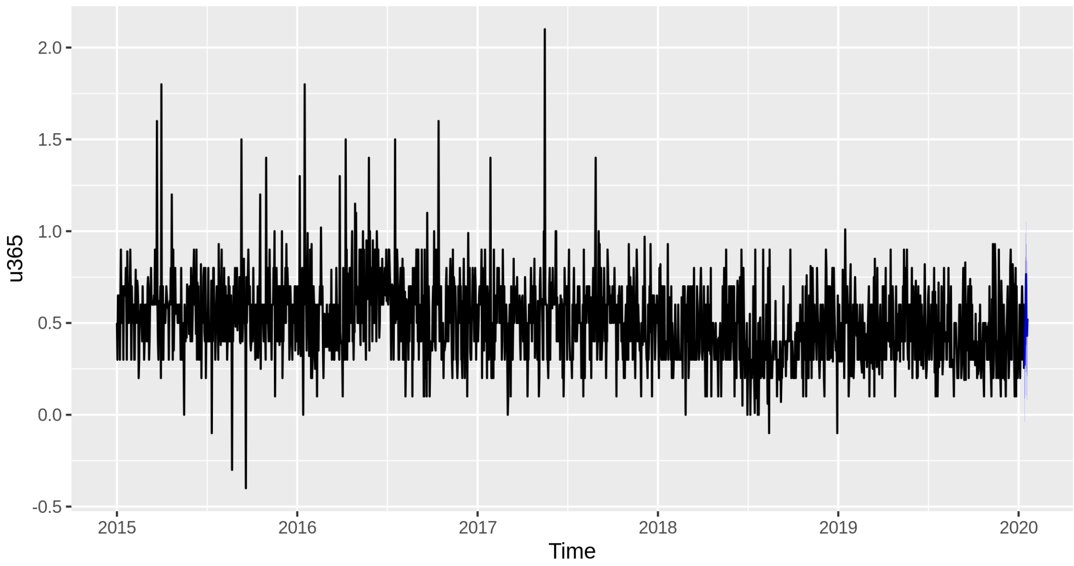

Figura 13 – *Holt-Winters* observado/ajustado.

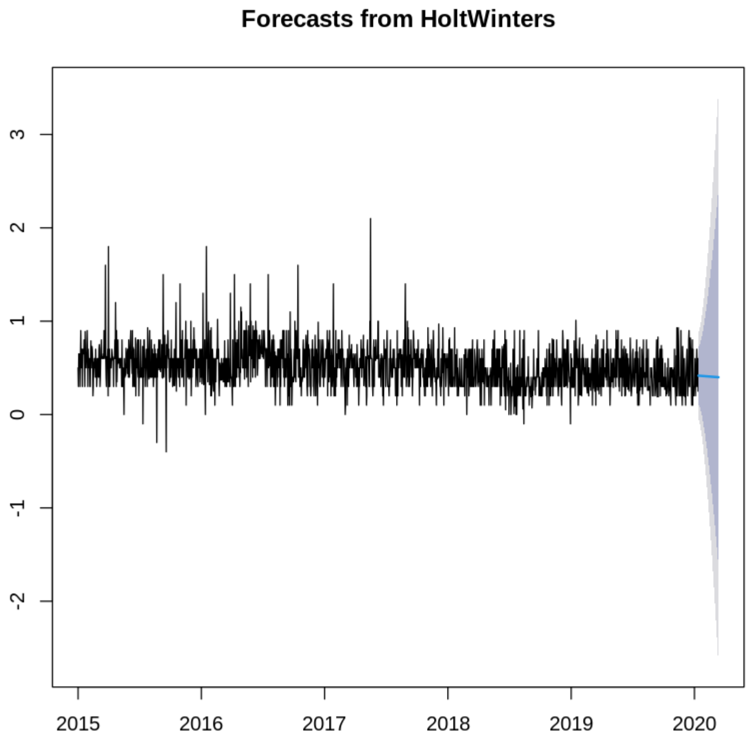

Figura 14 – Previsão com *Holt-Winters* para 60 dias.

12. Já na Figura 16, é outra opção de previsão com destaques para os intervalos superiores e inferiores.

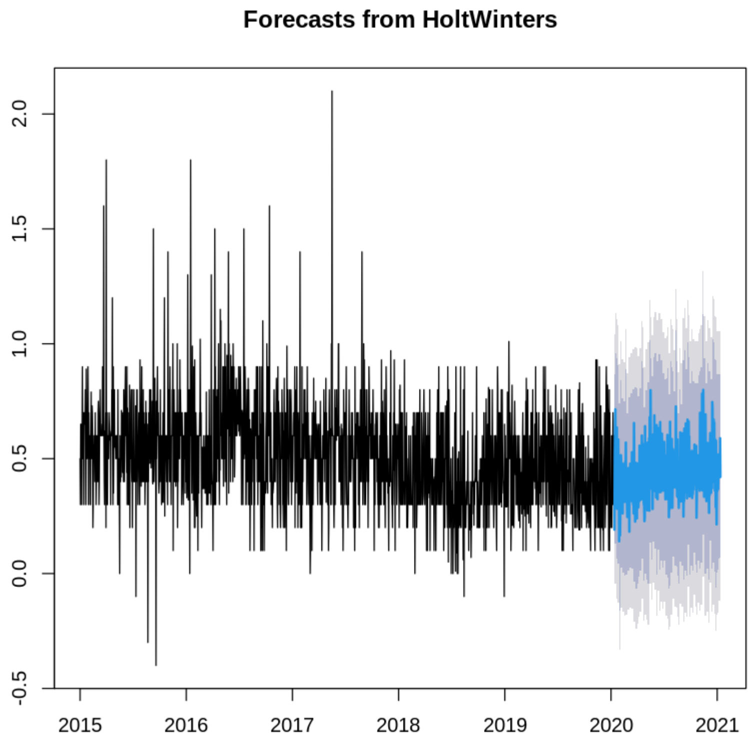

Figura 15 – Previsão com *Holt-Winters* para 1 ano.

2.2.3 Comentários

Series temporais tem inúmeros usos e fazem parte do nosso dia-a-dia. O maior desafio, sempre mencionado por quem trabalha com dados, é a obtenção dos dados, limpeza e ajustes dos mesmos. Previsão e séries temporais nos ajudam a ter parâmetros de comportamentos futuros (baseado no passado) dos dados observados. Um ponto que vale destacar é o domínio do conhecimento dos dados, pois desde a aquisição, tratamento e seleção do modelo, se faz necessário ter um conhecimento mínimo das informações envolvidas.

No caso específico do exemplo, temos que alguns modelos (como o linear) não auxiliam nas previsões. Modelos que levam em consideração sazonalidades e tendencias acabam sendo mais ajustados para o nosso exemplo, a magnitude da Betelgeuse.

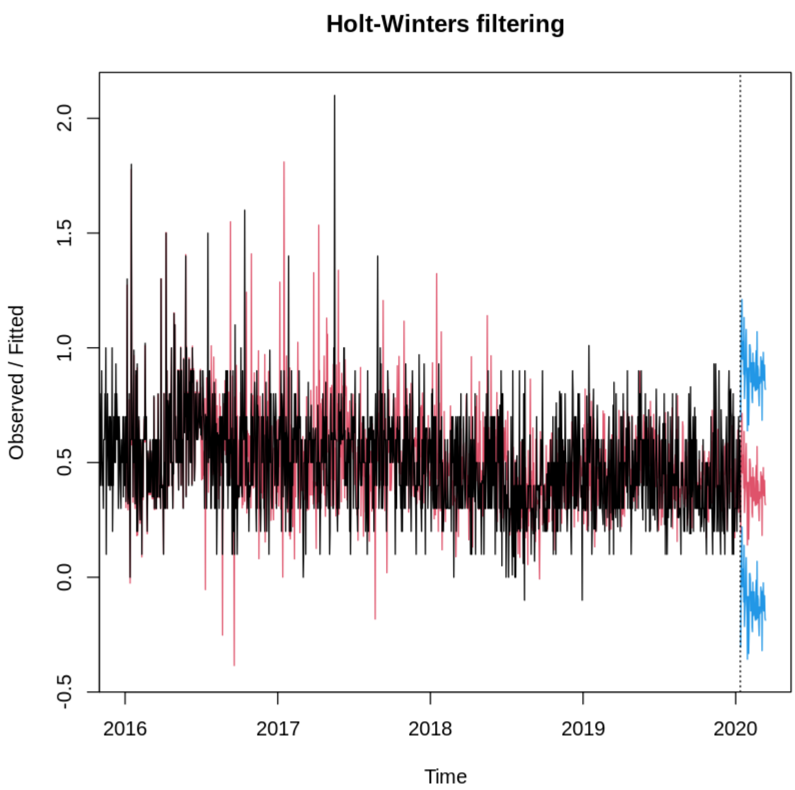

Figura 16 – Previsão para 60 dias, com intervalo destacado.