

Material Suplementar – O desenvolvimento do script de Programação em Ambiente R seguiu estes passos:

Passo 1: Instalar os pacotes necessários para a análise e previsão das séries temporais

```
Install.packages("forecast", "TSA", "tseries", "seasonal", "fma", "nnfor")

#Carregamento dos pacotes
library(forecast)
library(TSA)
library(tseries)
library(fma)
library(nnfor)
```

Passo 2: Abrir o banco de dados e determinar o diretório de trabalho

```
setwd("C:/Users/Desktop/Artigo Protocolo")
load("C:/Users/Desktop/Artigo Protocolo/Amapá.Rdata")
```

Passo 3: Criar a série temporal para o horizonte de previsão de 12 meses

Para executar os modelos de previsão em 'R', é preciso converter os dados em um objeto de série temporal, procedimento feito na primeira linha de código abaixo. Os argumentos '*start*' e '*end*' especificam o tempo da primeira e o tempo da última observação, respectivamente. O argumento '*frequency*' especifica o número de observações por unidade de tempo.

```
Set.seed(123)
x.h12 <- ts(malaria$AP, start = c(1997,1), frequency = 12)
x.h12 = log(x.h12)
test_x.h12 <- window(x.h12, start=c(2016,1))
x.h12 <- window(x.h12, end=c(2015,12))
```

Passo 4: Fazer a modelagem e previsão conforme a função a seguir:

```
models.h12 <- list(mod_arima = auto.arima(x.h12, ic='aicc', stepwise=FALSE),
mod_ets = ets(x.h12, ic='aicc', restrict=FALSE),
mod_neural = nnetar(x.h12, p=12, size=25),
mod_tbats = tbats(x.h12, ic='aicc', seasonal.periods=12),
mod_bats = bats(x.h12, ic='aicc', seasonal.periods=12),
mod_stlm = stlm(x.h12, s.window=12, ic='aicc', robust=TRUE, method='ets'),
mod_structst = StructTS(x.h12),
mod_elm = elm(x.h12),
mod_mlp = mlp(x.h12)
```

Passo 5: Fazer a previsão e apresentar os melhores modelos

```
acc.h12 <- lapply(forecasts.h12, function(f){  
  accuracy(f, test_x.h12)[2,,drop=FALSE]  
})  
acc.h12 <- Reduce(rbind, acc.h12)  
row.names(acc.h12) <- names(forecasts.h12)  
acc.h12 <- acc.h12[order(acc.h12[, 'MASE']),]  
round(acc.h12, 2)
```

Passo 6: Repetir os passos 4 e 5 para o horizonte de seis e três meses

Passo 7: Fazer os gráficos para as previsões realizadas e repetir para os horizontes de previsão de seis e três meses

```
s.realMT <- log(malaria$AP)  
  
#h12 plot()  
  
forecast_h12_arima <- c(forecasts.h12$mod_arima$fitted, forecasts.h12$mod_arima$mean)  
forecast_h12_exp <- c(forecasts.h12$mod_exp$fitted, forecasts.h12$mod_exp$mean)  
forecast_h12_neural <- c(forecasts.h12$mod_neural$fitted, forecasts.h12$mod_neural$mean)  
forecast_h12_tbats <- c(forecasts.h12$mod_tbats$fitted, forecasts.h12$mod_tbats$mean)  
forecast_h12_bats <- c(forecasts.h12$mod_bats$fitted, forecasts.h12$mod_bats$mean)  
forecast_h12_stl <- c(forecasts.h12$mod_stl$fitted, forecasts.h12$mod_stl$mean)  
forecast_h12_sts <- c(forecasts.h12$mod_sts$fitted, forecasts.h12$mod_sts$mean)  
forecast_h12_elm <- c(forecasts.h12$mod_elm$fitted, forecasts.h12$mod_elm$mean)  
forecast_h12_mlp <- c(forecasts.h12$mod_mlp$fitted, forecasts.h12$mod_mlp$mean)  
forecast_h12_naive <- c(forecasts.h12$naive$fitted, forecasts.h12$naive$mean)  
  
png("Fig_AP_malaria.png", units="mm", height=170, width=170, res=600)  
par(mar=c(4,4,2.2,0.1))  
par(mfrow=c(2,2))  
  
st <- 1:12  
  
plot(st, s.realMT[229:240], type = "l", lwd = 2, ylim = c(4,10), ylab = "Número de casos (log-n)", xlab = "2017 (mês)", main="A")  
points(st, forecast_h12_arima[229:240], type = "l", lwd = 2, col = 2, lty = "dashed")  
points(st, forecast_h12_exp[229:240], type = "l", lwd = 2, col = 3, lty = 3)  
points(st, forecast_h12_neural[229:240], type = "l", lwd = 2, col = 4, lty = 4)  
points(st, forecast_h12_tbats[229:240], type = "l", lwd = 2, col = 5, lty = 5)  
points(st, forecast_h12_bats[229:240], type = "l", lwd = 2, col = 6, lty = 6)  
points(st, forecast_h12_stl[229:240], type = "l", lwd = 2, col = 7, lty = 7)  
points(st, forecast_h12_sts[229:240], type = "l", lwd = 2, col = 8, lty = 8)
```

Continua

Continuação

```
points(st,forecast_h12_elm[224:235],type = "l",lwd = 2,col = 9,lty = 9)
points(st,forecast_h12_mlp[224:235],type = "l",lwd = 2,col = 10,lty = 10)
points(st,forecast_h12_naive[229:240],type = "l",lwd = 2,col = 11,lty = 11)

#Legenda
plot(229:240,s.realMT[229:240],type='n',axes=F,ann=F)
legend(229,7,legend=c("Real","Arima","Exponencial","Neural","TBATS","BATS","STL","STS","ELM",
"MLP","Naive"),lty=c(1,2,3,4,5,6,7,8,9,10,11),col=c(1,2,3,4,5,6,7,8,9,10,11),lwd=2,bty="n")

dev.off()
```