

# Análise de Séries Temporais

## 0.4 - Aula Prática

Prof. Dr. Eder Angelo Milani

31/05/2025

### Tendência e Sazonalidade

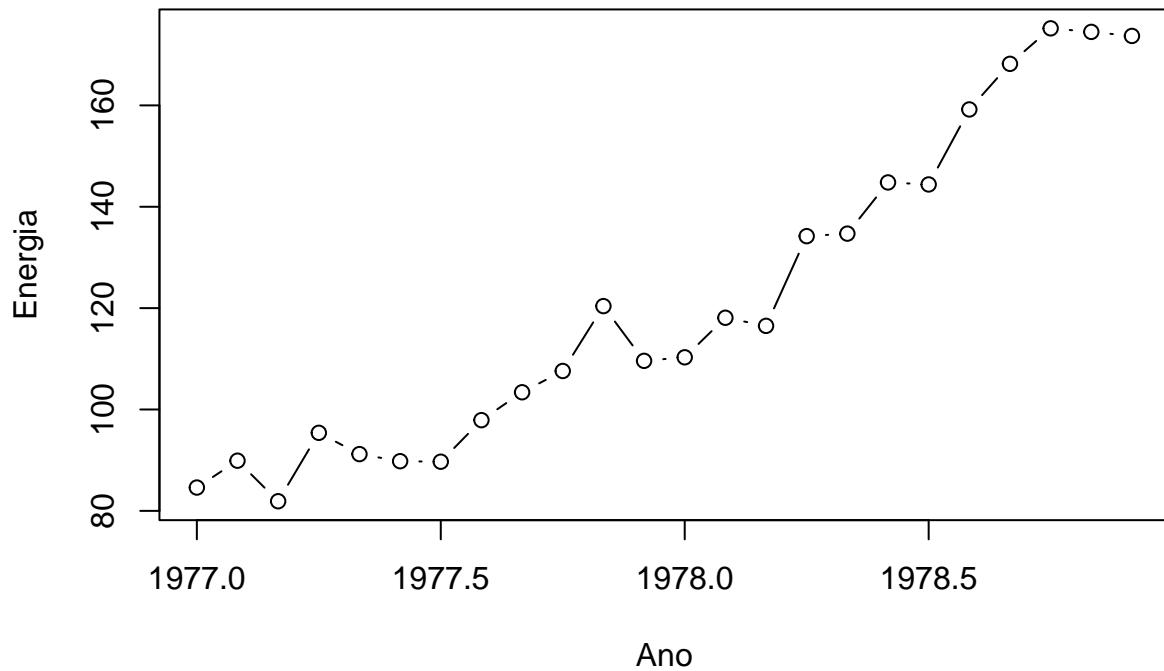
#### Exemplo de uma série temporal com tendência

A seguir apresentamos os valores mensais do consumo de energia elétrica no Estado do Espírito Santo, referentes aos anos 1977 e 1978, portanto são 24 observações. Série temporal retirada de Morettin e Toloi (2006).

```
energia <- c(84.6, 89.9, 81.9, 95.4, 91.2, 89.8, 89.7, 97.9, 103.4,
           107.6, 120.4, 109.6, 110.3, 118.1, 116.5, 134.2, 134.7,
           144.8, 144.4, 159.2, 168.2, 175.2, 174.5, 173.7)

energia_ts <- ts(energia, start=c(1977, 1), frequency = 12)

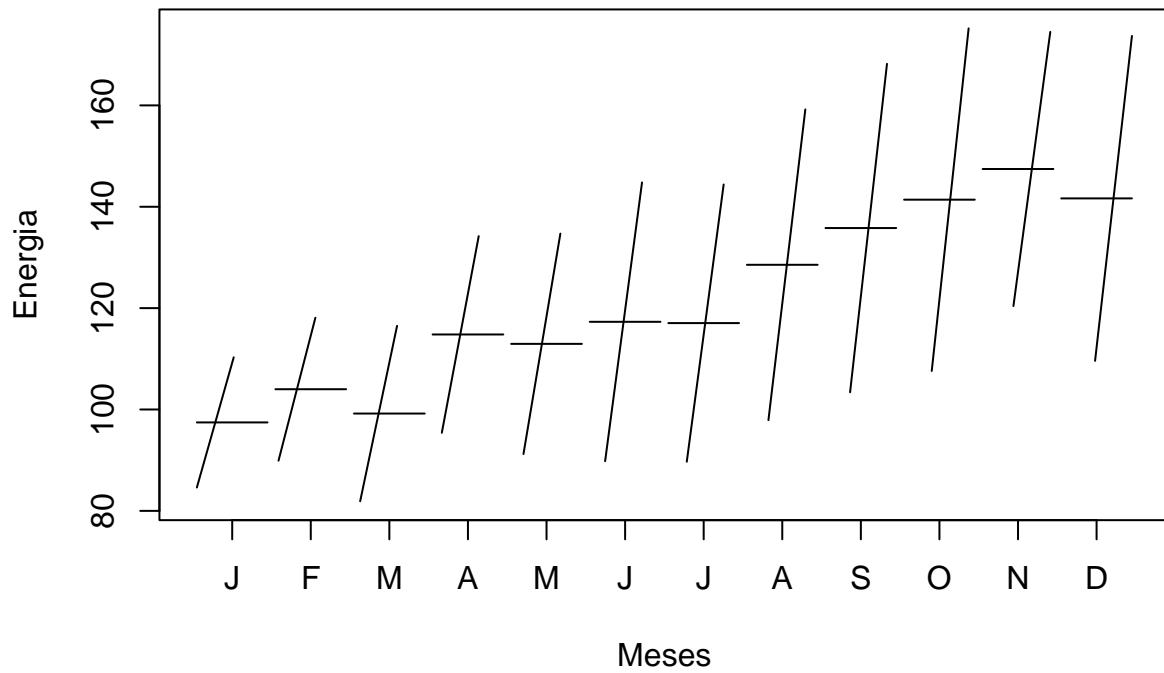
# grafico da serie temporal
plot(energia_ts, type='b', ylab = "Energia", xlab="Ano")
```



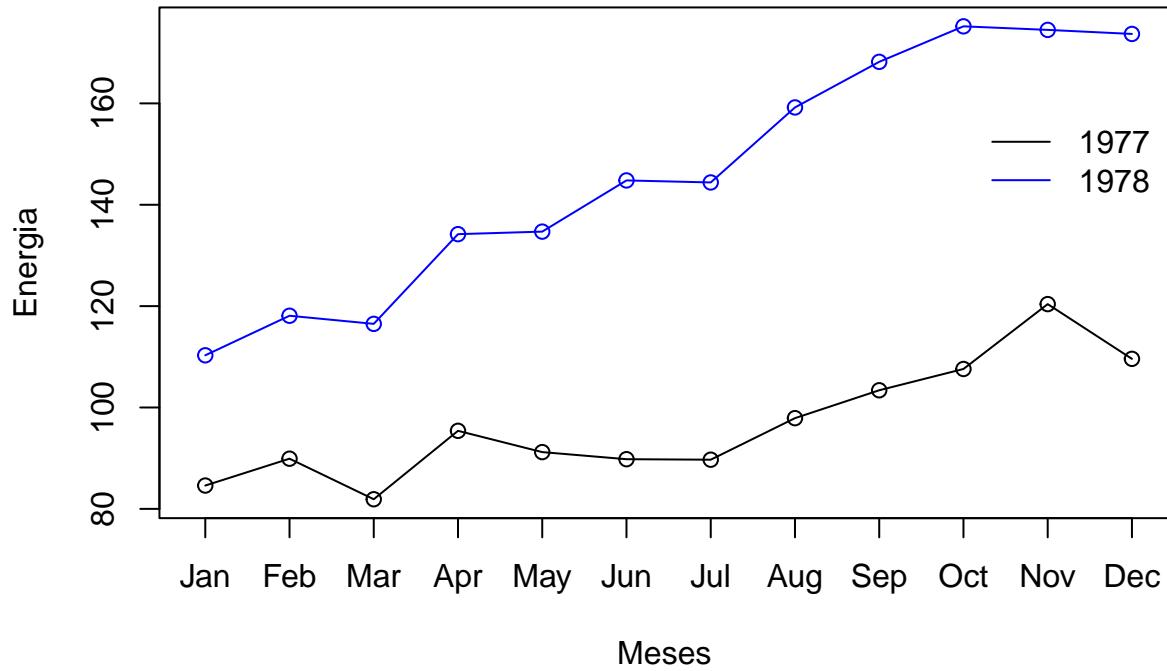
```
#install.packages("forecast")
library(forecast)

## Warning: pacote 'forecast' foi compilado no R versão 4.4.3
## Registered S3 method overwritten by 'quantmod':
##   method           from
##   as.zoo.data.frame zoo

# grafico dos meses separadamente
monthplot(energia_ts, xlab="Meses", ylab="Energia", main="")
```



```
# grafico dos anos separadamente
seasonplot(energia_ts, xlab="Meses", ylab="Energia", main="", col=c("black", "blue"))
legend(10, 160, lty=c(1,1), col=c("black", "blue"), c("1977", "1978"), bty = "n")
```



### Ajustando uma tendência polinomial

Sabemos que a função *lm* no R ajusta modelos lineares, no entanto a função *tSLM* do pacote *forecast* apresenta algumas vantagens para o ajuste de modelos envolvendo séries temporais. Vamos utilizá-la para obter as estimativas de mínimos quadrados usuais para  $(\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2, \dots, \hat{\beta}_m)$ .

```
# tendencia linear - beta_0 + beta_1 * t - para t=1, 2, ...
# a variavel y tem que ser um objeto ts
```

```
ajuste1 <- tSLM(energia_ts ~ trend)
summary(ajuste1)
```

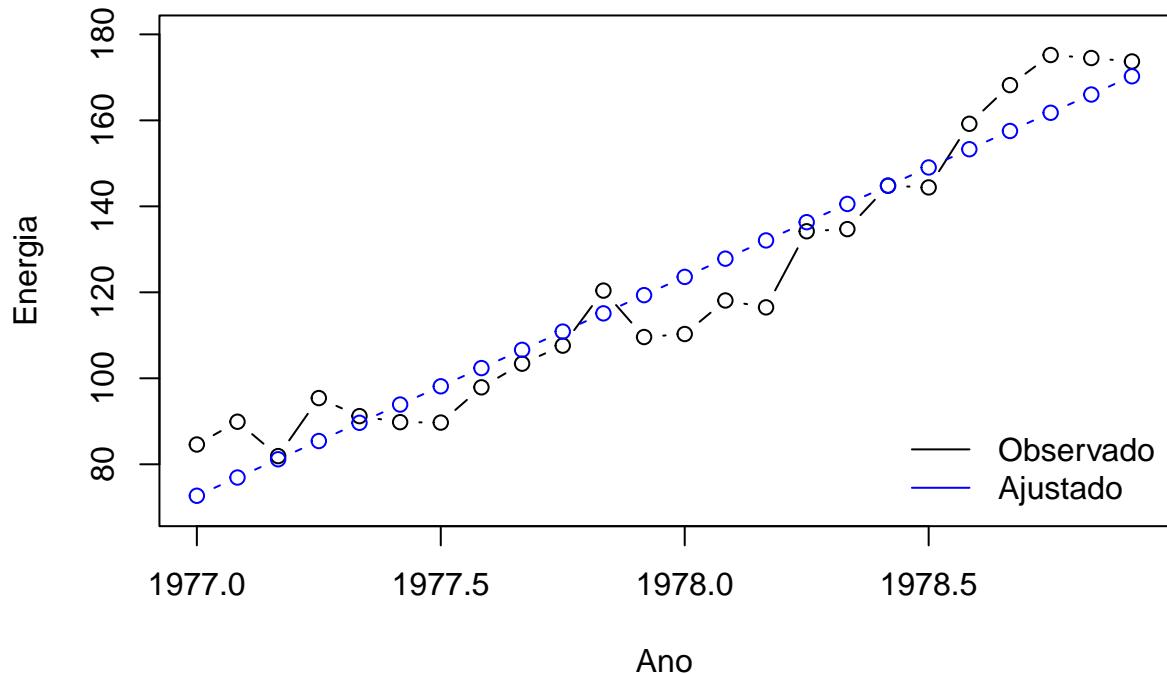
```
##
## Call:
## tSLM(formula = energia_ts ~ trend)
##
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -15.572  -4.945  -1.057   6.560  13.432 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 68.4384    3.6519  18.74 5.16e-15 ***
## trend       4.2423    0.2556  16.60 6.30e-14 ***
## ---        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```

## Residual standard error: 8.667 on 22 degrees of freedom
## Multiple R-squared:  0.9261, Adjusted R-squared:  0.9227
## F-statistic: 275.5 on 1 and 22 DF,  p-value: 6.295e-14

# grafico da serie temporal
plot(energia_ts, type='b', ylab = "Energia", xlab="Ano", ylim=c(70, 180))
# valores ajustados
lines(ajuste1$fitted.values, type="b", col="blue")
legend("bottomright", lty=c(1,1), col=c("black", "blue"), c("Observado", "Ajustado"),
      bty = "n")

```



```

# observe que a mesma logica com a funcao lm nao funciona
# modelo1 <- lm(energia ~ trend)

# tendência quadrática - beta_0 + beta_1 * t + beta_2 * t^2 - para t=1, 2, ...
ajuste2 <- tslm(energia_ts ~ trend + I(trend^2))
summary(ajuste2)

##
## Call:
## tslm(formula = energia_ts ~ trend + I(trend^2))
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -9.341 -3.382 -0.516  3.328 12.127

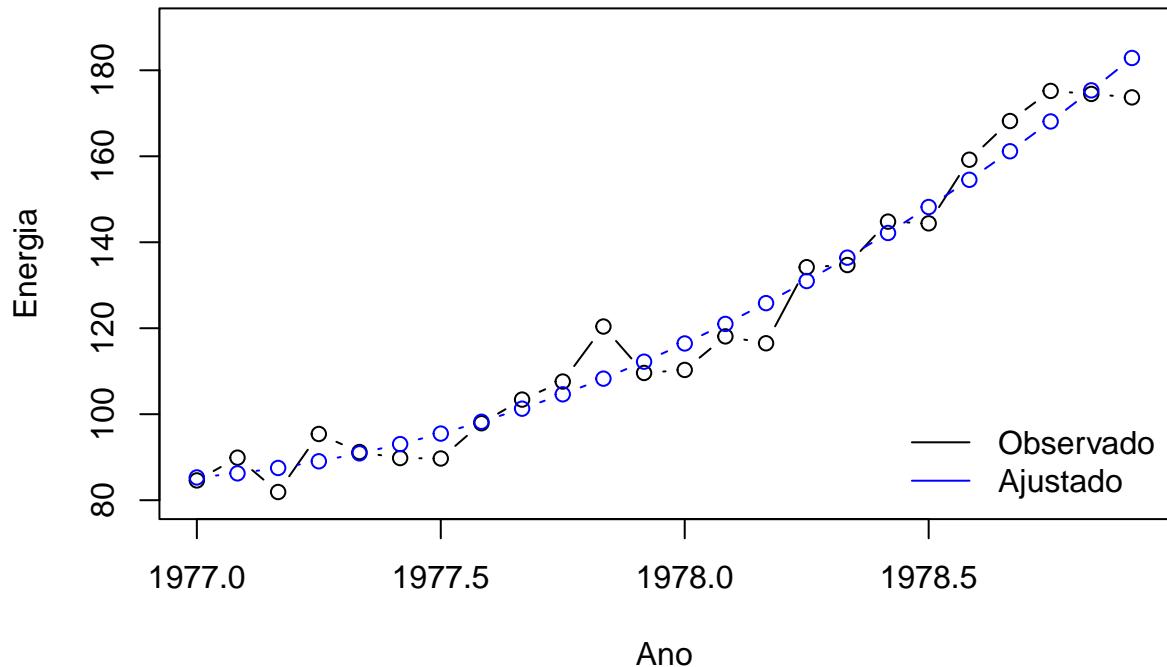
```

```

## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 84.64032   3.77345 22.430 3.74e-16 ***
## trend       0.50336   0.69547  0.724   0.477    
## I(trend^2)  0.14956   0.02701  5.538 1.70e-05 ***
## ---      
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 5.656 on 21 degrees of freedom
## Multiple R-squared:  0.9699, Adjusted R-squared:  0.9671 
## F-statistic: 338.9 on 2 and 21 DF,  p-value: < 2.2e-16

# grafico da serie temporal
plot(energia_ts, type='b', ylab = "Energia", xlab="Ano", ylim=c(80, 190))
# valores ajustados
lines(ajuste2$fitted.values, type="b", col="blue")
legend("bottomright", lty=c(1,1), col=c("black", "blue"), c("Observado", "Ajustado"),
      bty = "n")

```



```

# R2 do ajuste 1 e 0.9261
# R2 do ajuste 2 e 0.9699

AIC(ajuste1)

## [1] 175.678

```

```
AIC(ajuste2)
```

```
## [1] 156.0716
# AIC(ajuste1) = 175.678
# AIC(ajuste2) = 156.0716

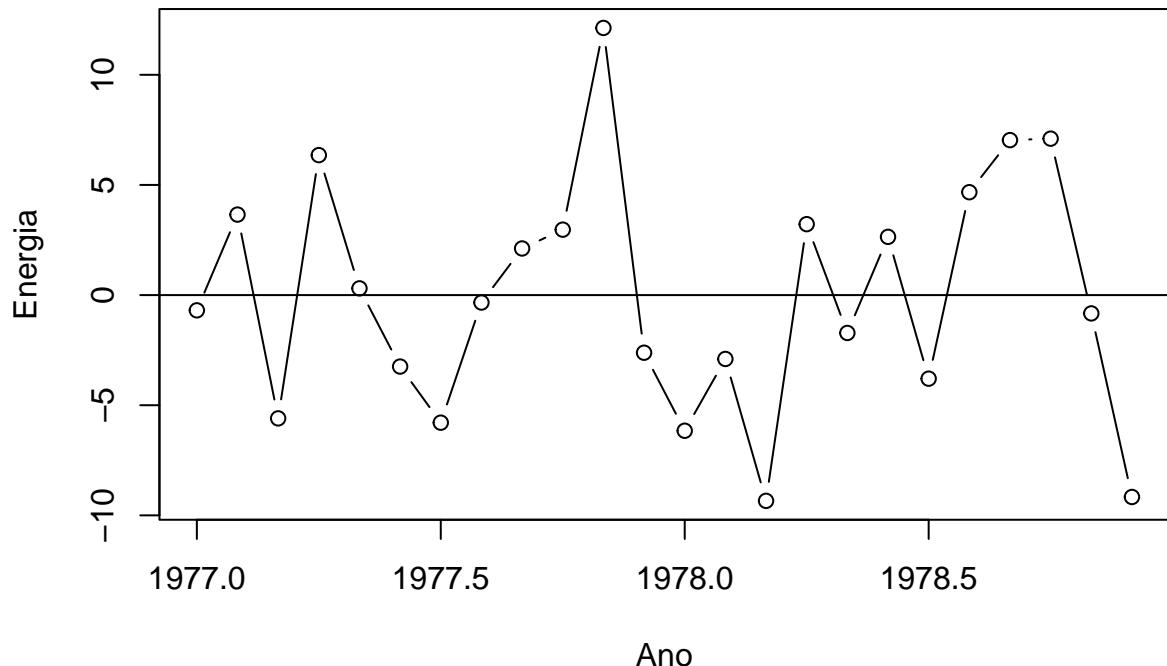
# Indicando assim que o ajuste 2 é o melhor entre os dois
```

Agora vamos obter a série livre de tendência, ou seja,  $Y_t = Z_t - \hat{T}_t$ .

```
energia_ajustada <- ajuste2$fitted.values

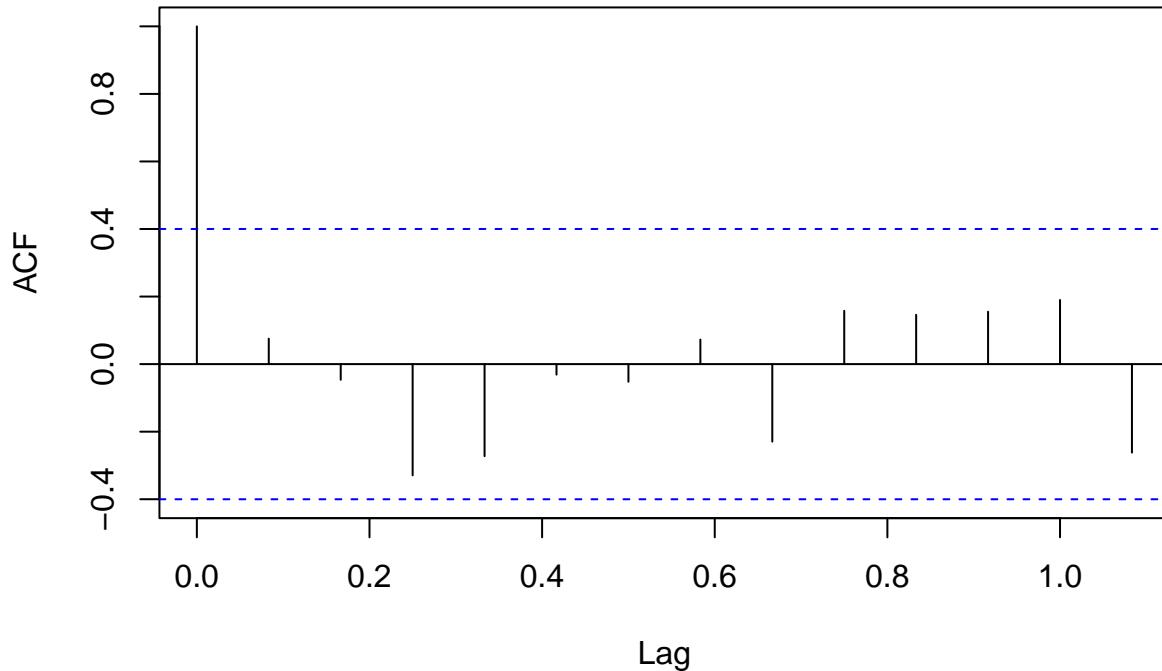
energia_st_ts <- energia_ts - energia_ajustada

# grafico da serie livre de tendencia
plot(energia_st_ts, type='b', ylab="Energia", xlab="Ano")
abline(h=0)
```



```
# grafico da fac
acf(energia_st_ts)
```

## Series energia\_st\_ts



Utilizando o modelo estimado para a tendência,  $T_t$ , podemos prever valores futuros da série. Na sequência calculamos as previsões para  $h = 1, 2, 3$  e  $4$ .

```
previsao <- forecast(ajuste2, h = 4)
previsao

##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jan 1979     190.6968 181.7006 199.6931 176.5577 204.8359
## Feb 1979     198.8276 189.3273 208.3279 183.8962 213.7589
## Mar 1979     207.2574 197.1466 217.3681 191.3666 223.1481
## Apr 1979     215.9863 205.1569 226.8157 198.9661 233.0066

# como é feita a estimacao pontual
# h=1 implica em t=25
t=25
ajuste2$coefficients[1] + ajuste2$coefficients[2] * t + ajuste2$coefficients[3] * t^2

## (Intercept)
##      190.6968

# h=2 implica em t=26
t=26
ajuste2$coefficients[1] + ajuste2$coefficients[2] * t + ajuste2$coefficients[3] * t^2

## (Intercept)
##      198.8276

# grafico com os valores observados e preditos
```

```

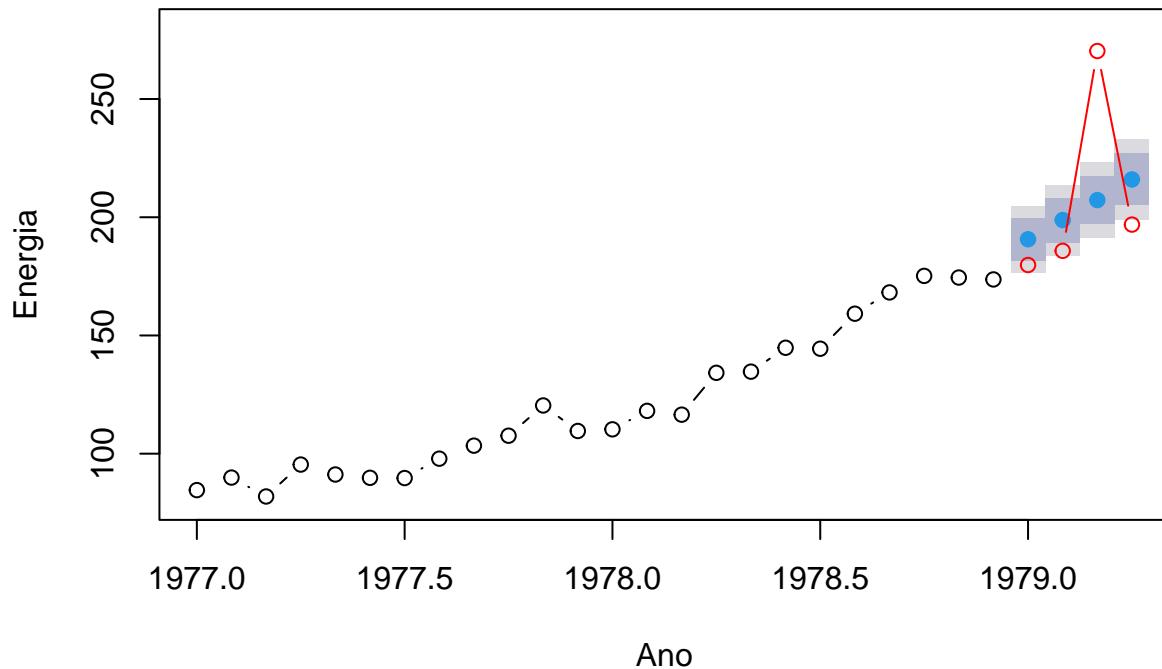
energia_novos_ts <- ts(c(179.8, 185.8, 270.3, 196.9), start=c(1979, 1), frequency = 12)

round(cbind(energia_novos_ts, previsao$mean), 1)

##          energia_novos_ts previsao$mean
## Jan 1979           179.8           190.7
## Feb 1979           185.8           198.8
## Mar 1979           270.3           207.3
## Apr 1979           196.9           216.0

plot(previsao, type="b", ylab="Energia", xlab="Ano", main="", ylim=c(80, 280))
lines(energia_novos_ts, col="red", type="b")

```



```

# calculo do erro

# Raiz do Erro Quadratico Medio (REQM / RMSE)
RMSE <- sqrt(mean((energia_novos_ts - previsao$mean)^2))

# Erro Absoluto Medio (EAM / MAE)
MAE <- mean(abs(energia_novos_ts - previsao$mean))

# Erro Percentual Absoluto Medio (MAPE)
MAPE <- mean(abs(energia_novos_ts - previsao$mean) / abs(energia_novos_ts)) * 100

cat("RMSE:", RMSE, "\n")

## RMSE: 34.01146

```

```
cat("MAE:", MAE, "\n")
## MAE: 26.51333
cat("MAPE:", MAPE, "%\n")
## MAPE: 11.52219 %
```

## Exemplo de uma série temporal com sazonalidade

A seguir apresentamos as temperaturas médias mensais, em graus centígrados, da cidade de Cananéia (município brasileiro do litoral de São Paulo), de janeiro de 1976 a dezembro de 1985. Série temporal retirada de Morettin e Toloi (2006).

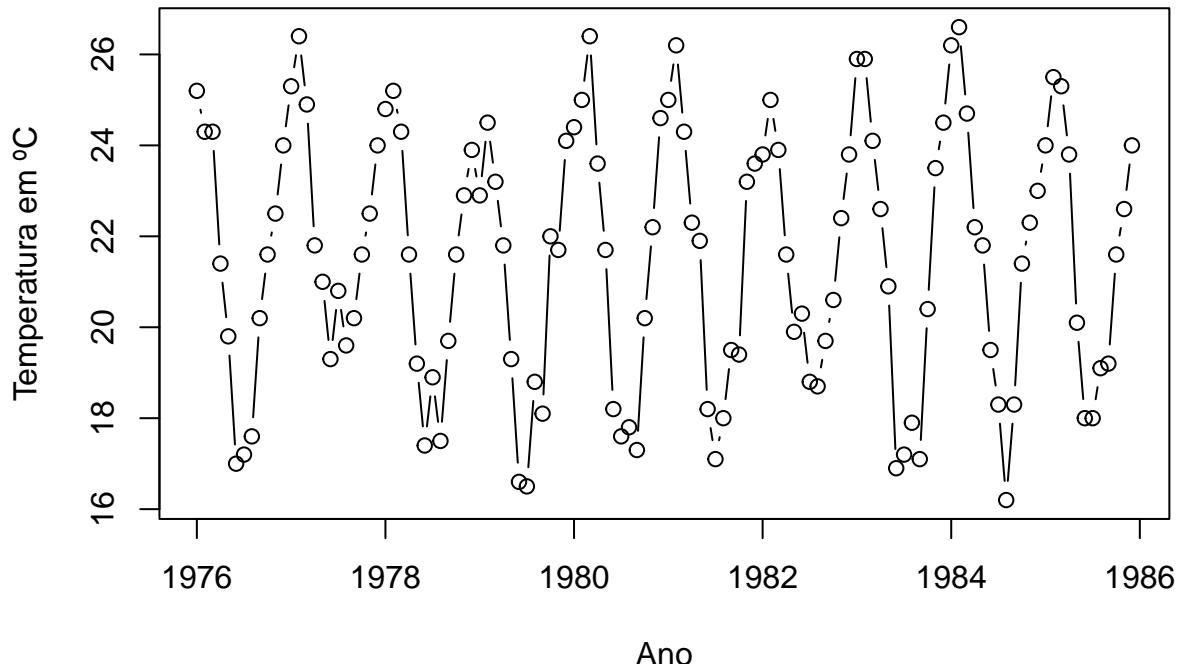
```
# fazendo a leitura do conjunto de dados
setwd("G:\\Meu Drive\\UFG\\Especializacao\\Aulas de series temporais\\Códigos")

library(readxl)
temperatura <- read_excel("temperatura.xls")
head(temperatura)

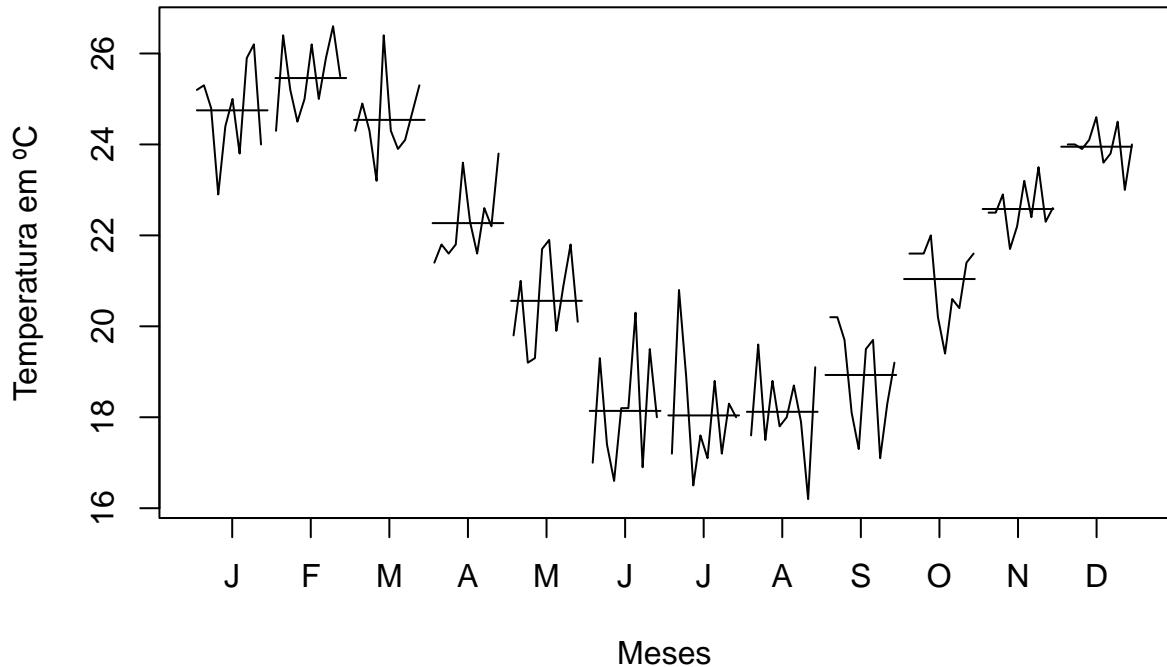
## # A tibble: 6 x 3
##   Ano Cananeia Ubatuba
##   <dbl>    <dbl>    <dbl>
## 1 1976     25.2     27.1
## 2 NA       24.3     25.3
## 3 NA       24.3     25.8
## 4 NA       21.4     23.7
## 5 NA       19.8     21.6
## 6 NA       17.0     20.0

temp.cananeia_ts <- ts(temperatura$Cananeia, start = c(1976, 1), frequency = 12)

# grafico da serie temporal
plot.ts(temp.cananeia_ts, type="b", ylab="Temperatura em °C", xlab="Ano")
```

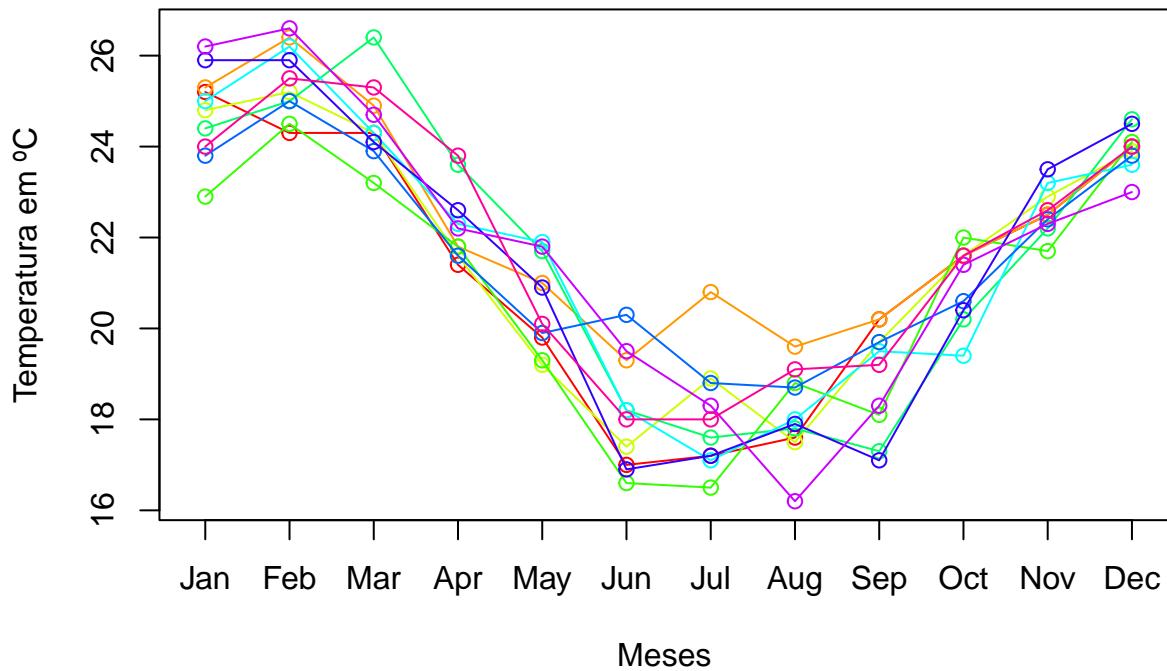


```
# grafico dos meses separadamente
monthplot(temp.cananeia_ts, xlab="Meses", ylab="Temperatura em °C", main="")
```



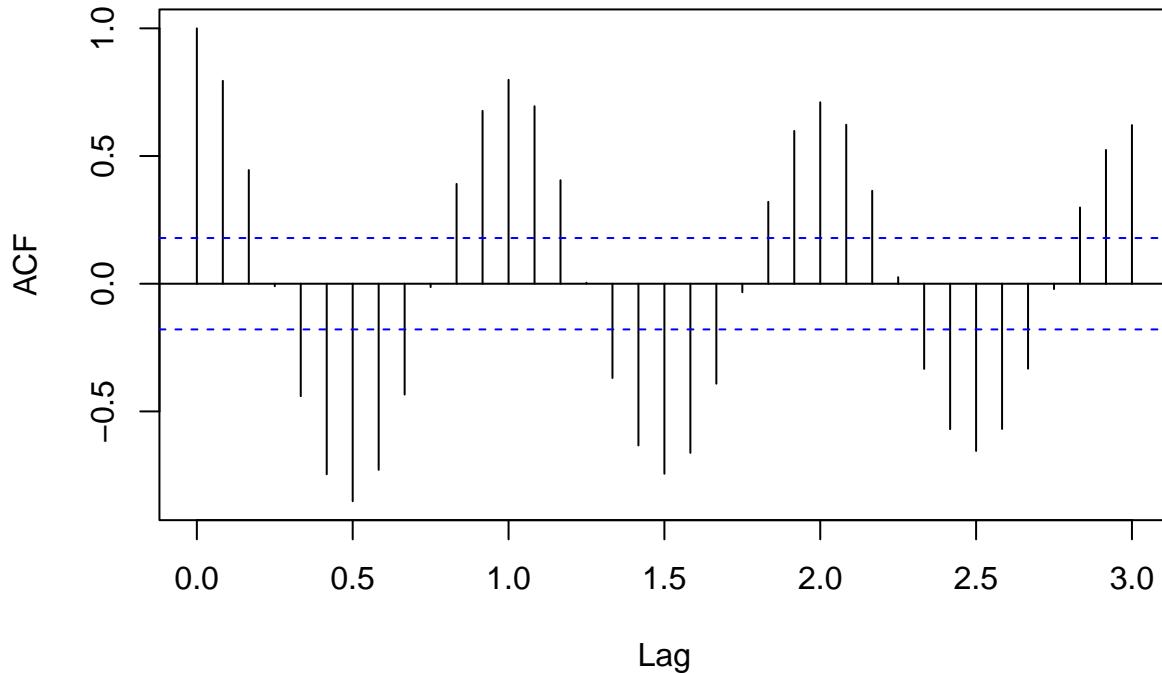
```
# grafico dos anos separadamente
cores <- rainbow(10)
anos_legenda <- as.character(1976:1985)

seasonplot(temp.cananeia_ts, xlab="Meses", ylab="Temperatura em °C", main="", col=cores)
```



```
#ylim=c(0, 30)
#legend(7, 17, lty=1, col=cores[1:5], anos_legenda[1:5], bty = "n")
#legend(10, 17, lty=1, col=cores[6:10], anos_legenda[6:10], bty = "n")

# grafico da ACF
acf(temp.cananeia_ts, main="", lag.max=36)
```



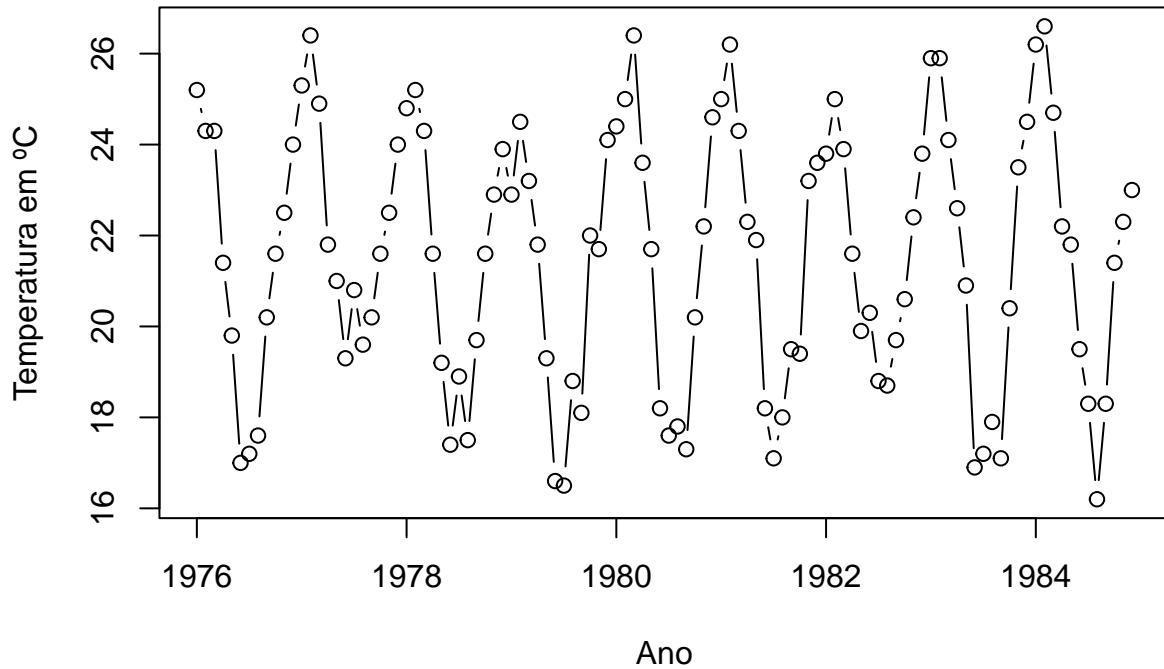
```
# tabela do slide

temp.cananeia_mt <- matrix(temperatura$Cananeia, ncol=12, nrow=10, byrow=T)
temp.cananeia_mt <- rbind(temp.cananeia_mt, colMeans(temp.cananeia_mt) )
temp.cananeia_mt <- cbind(temp.cananeia_mt, c(rowMeans(temp.cananeia_mt)))
#library(xtable)
#xtable(round(temp.cananeia_mt))
```

Vamos agora proceder um ajuste de um modelo de regressão com variáveis *dummies* para capturar a sazonalidade presente na série temporal. Para o ajuste vamos adotar 9 anos, deixando o último ano para realizar comparações com as previsões do modelo ajustado.

```
trein_temp.cananeia_ts <- ts(temperatura$Cananeia[1:108], start = c(1976, 1), frequency = 12)

# grafico da serie temporal
plot(trein_temp.cananeia_ts, type="b", ylab="Temperatura em °C", xlab="Ano")
```



```

# ajuste
ajuste1 <- tslm(trein_temp.cananeia_ts ~ season)
summary(ajuste1)

##
## Call:
## tslm(formula = trein_temp.cananeia_ts ~ season)
##
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -1.93333 -0.58333 -0.06111  0.62222  2.75556 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 24.8333    0.3229  76.901 < 2e-16 ***
## season2     0.6222    0.4567   1.362   0.1762    
## season3    -0.3778    0.4567  -0.827   0.4102    
## season4    -2.7333    0.4567  -5.985 3.72e-08 ***
## season5    -4.2222    0.4567  -9.245 6.22e-15 ***
## season6    -6.6778    0.4567 -14.622 < 2e-16 ***
## season7    -6.7889    0.4567 -14.865 < 2e-16 ***
## season8    -6.8222    0.4567 -14.938 < 2e-16 ***
## season9    -5.9333    0.4567 -12.992 < 2e-16 ***
## season10   -3.8556    0.4567  -8.442 3.26e-13 ***
## season11   -2.2556    0.4567  -4.939 3.32e-06 ***
## season12   -0.8889    0.4567  -1.946   0.0545 .  

```

```

## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9688 on 96 degrees of freedom
## Multiple R-squared:  0.8958, Adjusted R-squared:  0.8839
## F-statistic: 75.07 on 11 and 96 DF,  p-value: < 2.2e-16
AIC(ajuste1)

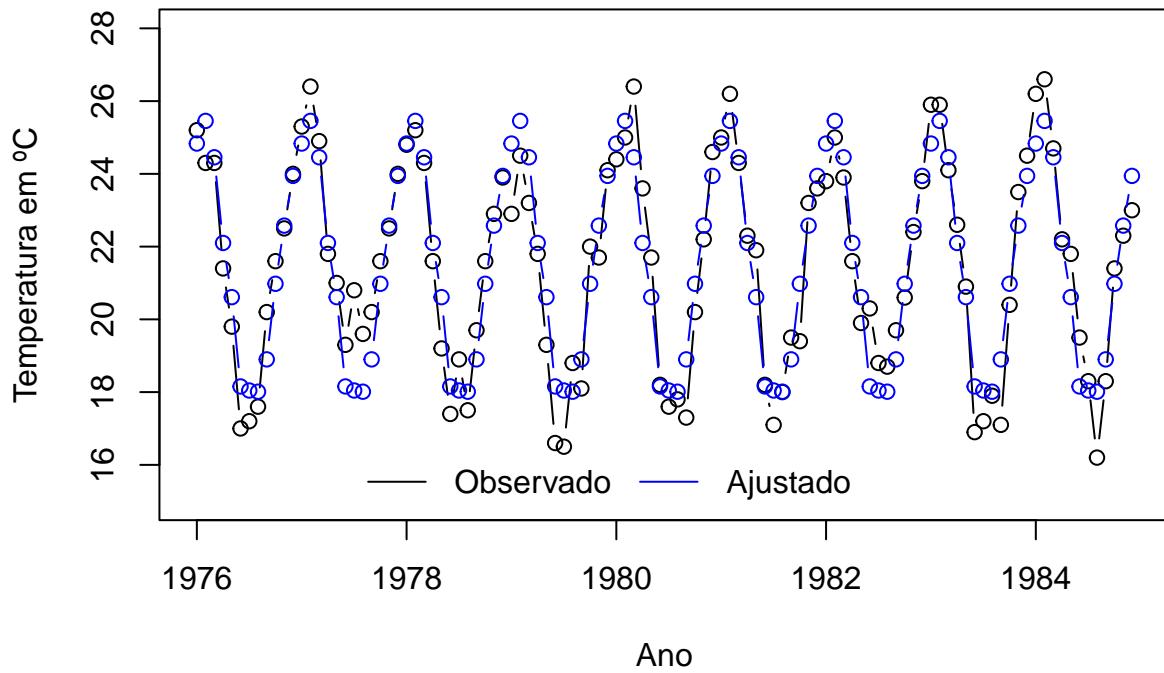
## [1] 312.9199

# qual a matriz que estou considerando?
# essa e apenas uma ideia, nao e exatamente essa
# a diferenca esta no mes de referencia
head(seasonaldummy(trein_temp.cananeia_ts)) # matriz da parte sazonal

##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov
## [1,]   1   0   0   0   0   0   0   0   0   0   0
## [2,]   0   1   0   0   0   0   0   0   0   0   0
## [3,]   0   0   1   0   0   0   0   0   0   0   0
## [4,]   0   0   0   1   0   0   0   0   0   0   0
## [5,]   0   0   0   0   1   0   0   0   0   0   0
## [6,]   0   0   0   0   0   1   0   0   0   0   0

# grafico da serie temporal
plot(trein_temp.cananeia_ts, type="b", ylab="Temperatura em °C", xlab="Ano", ylim=c(15, 28))
# valores ajustados
lines(ajuste1$fitted.values, type="b", col="blue")
legend("bottomright", lty=c(1,1), col=c("black", "blue"), c("Observado", "Ajustado"),
       bty = "n", ncol=3)

```

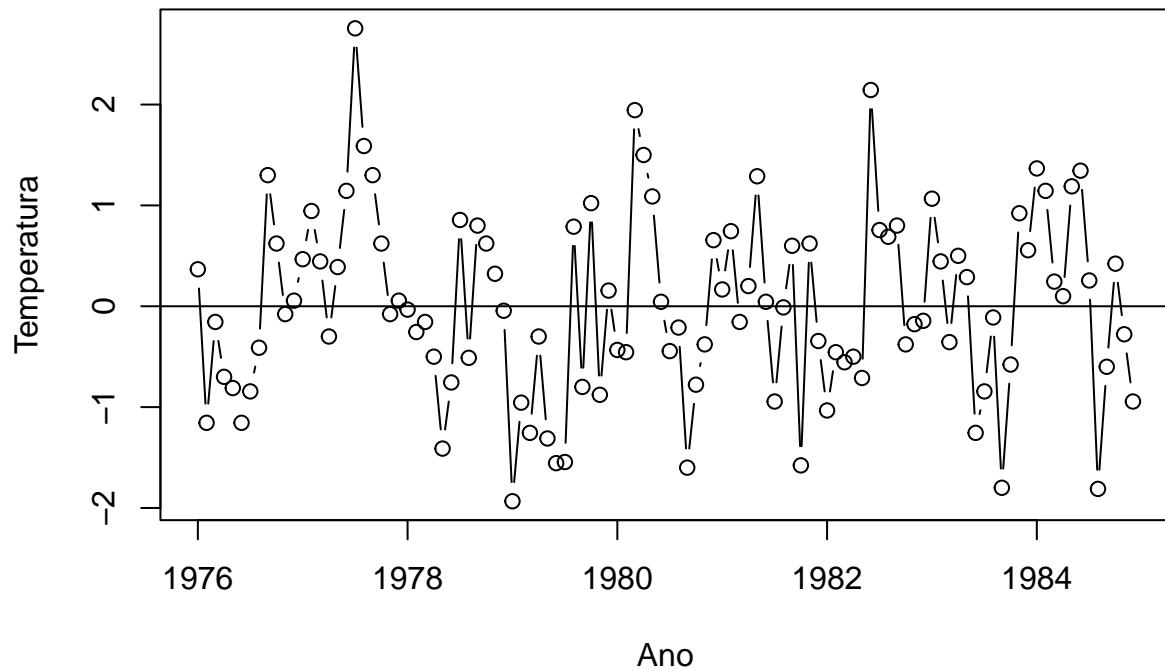


Agora vamos obter a série livre de sazonalidade, ou seja,  $Y_t = Z_t - \hat{S}_t$ .

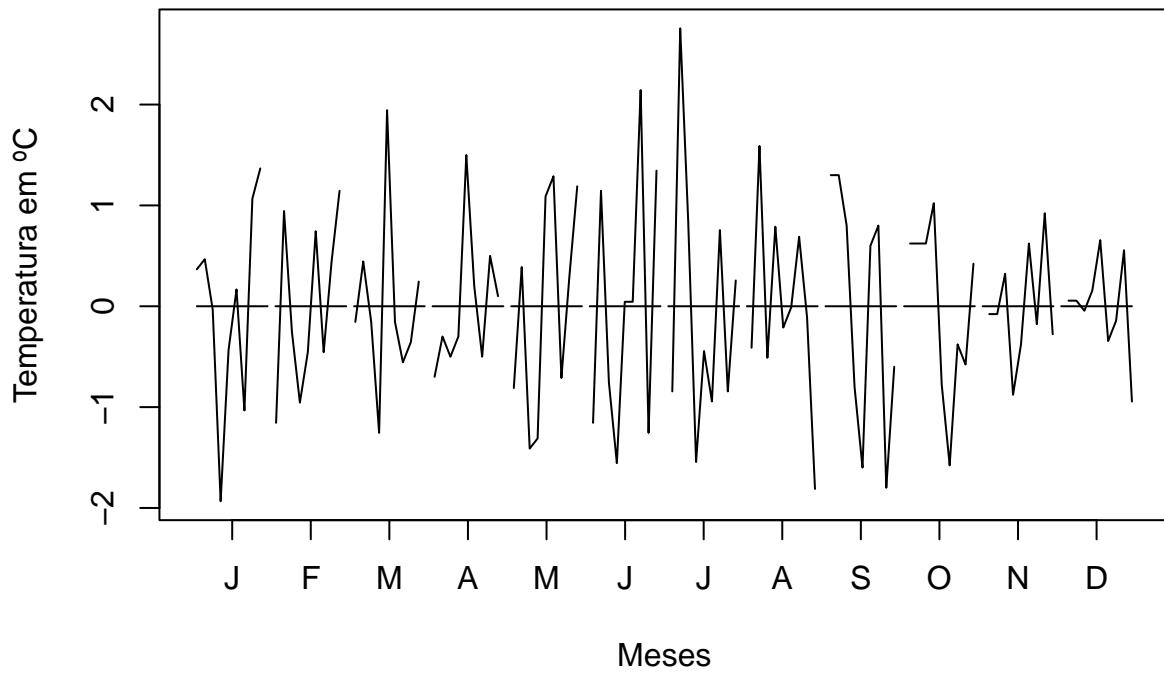
```
cananeia_ajustada <- ajuste1$fitted.values

cananeia_ss_ts <- trein_temp.cananeia_ts - cananeia_ajustada

# grafico da serie livre de sazonalidade
plot(cananeia_ss_ts, type='b', ylab="Temperatura", xlab="Ano")
abline(h=0)
```

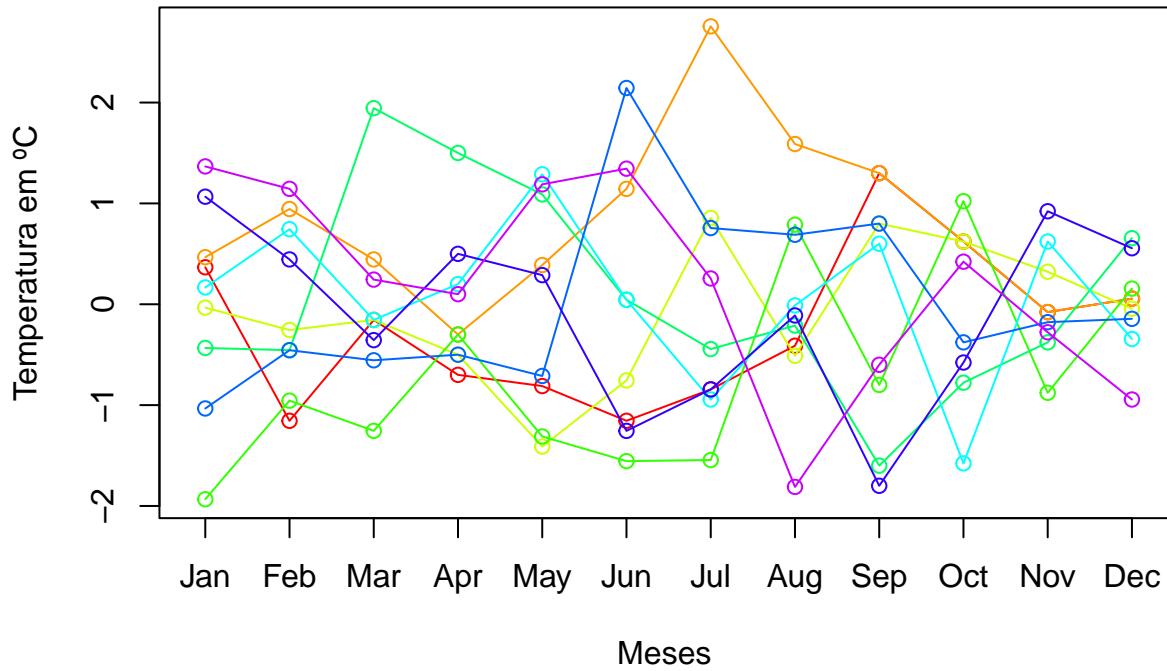


```
# grafico dos meses separadamente
monthplot(cananeia_ss_ts, xlab="Meses", ylab="Temperatura em °C", main="")
```



```
# grafico dos anos separadamente
cores <- rainbow(10)
anos_legenda <- as.character(1976:1985)

seasonplot(cananeia_ss_ts, xlab="Meses", ylab="Temperatura em °C", main="", col=cores)
```



Utilizando o modelo estimado para a sazonalidade,  $S_t$ , podemos prever valores futuros da série. Na sequência calculamos as previsões para  $h = 1, 2, \dots, 12$ .

```
previsao <- forecast(ajuste1, h = 12)
previsao
```

```
##           Point Forecast    Lo 80     Hi 80    Lo 95     Hi 95
## Jan 1985   24.83333 23.51556 26.15111 22.80629 26.86037
## Feb 1985   25.45556 24.13778 26.77333 23.42851 27.48260
## Mar 1985   24.45556 23.13778 25.77333 22.42851 26.48260
## Apr 1985   22.10000 20.78223 23.41777 20.07296 24.12704
## May 1985   20.61111 19.29334 21.92888 18.58407 22.63815
## Jun 1985   18.15556 16.83778 19.47333 16.12851 20.18260
## Jul 1985   18.04444 16.72667 19.36222 16.01740 20.07149
## Aug 1985   18.01111 16.69334 19.32888 15.98407 20.03815
## Sep 1985   18.90000 17.58223 20.21777 16.87296 20.92704
## Oct 1985   20.97778 19.66000 22.29555 18.95074 23.00482
## Nov 1985   22.57778 21.26000 23.89555 20.55074 24.60482
## Dec 1985   23.94444 22.62667 25.26222 21.91740 25.97149
```

```
# como e feita a estimacao pontual
# h=1 - mes de janeiro
x <- c(1, rep(0, 11))
# - 1 para o intercepto
# - 11 vezes 0, pois e o mes de janeiro (mes de referencia)

ajuste1$coefficients %*% x
```

```

##          [,1]
## [1,] 24.83333
# h=2 - mes de fevereiro
x <- c(1, 1, rep(0, 10))
# - 1 para o intercepto
# - 1 para o mes de fevereiro
# - 10 vezes 0, pois apenas o mes de fevereiro e diferente de 1

ajuste1$coefficients %*% x

##          [,1]
## [1,] 25.45556
# grafico com os valores observados e preditos

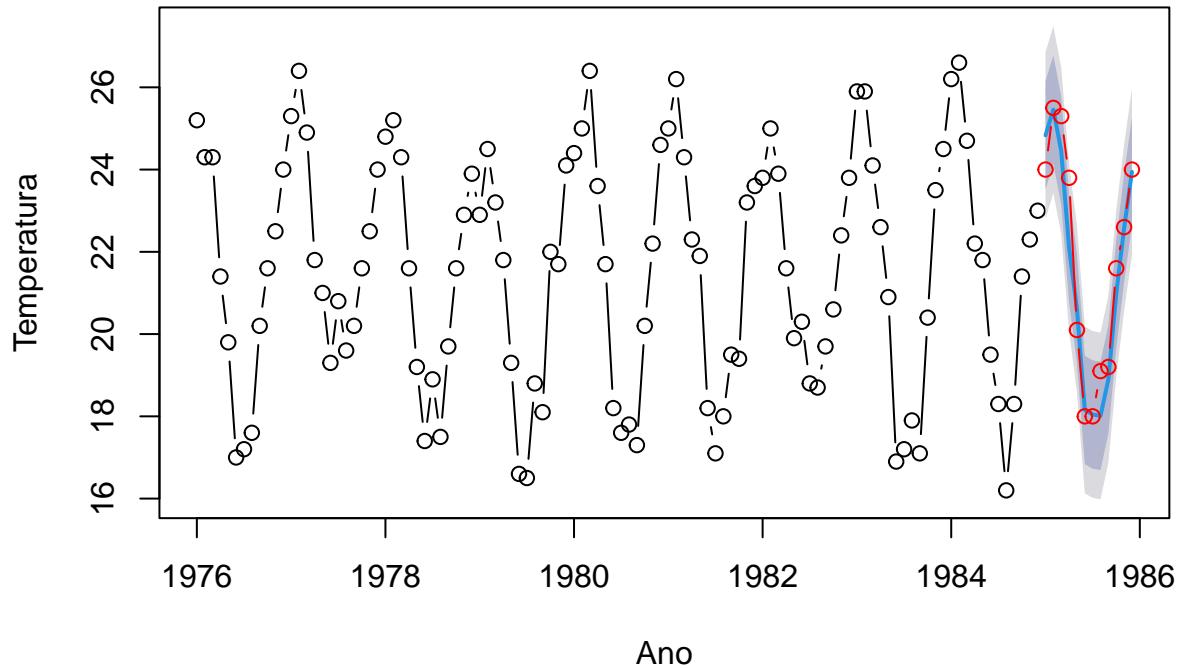
cananeia_novos_ts <- ts(temperatura$Cananeia[109:120], start = c(1985, 1), frequency = 12)

round(cbind(cananeia_novos_ts, previsao$mean), 1)

##      cananeia_novos_ts previsao$mean
## Jan 1985           24.0           24.8
## Feb 1985           25.5           25.5
## Mar 1985           25.3           24.5
## Apr 1985           23.8           22.1
## May 1985           20.1           20.6
## Jun 1985           18.0           18.2
## Jul 1985           18.0           18.0
## Aug 1985           19.1           18.0
## Sep 1985           19.2           18.9
## Oct 1985           21.6           21.0
## Nov 1985           22.6           22.6
## Dec 1985           24.0           23.9

plot(previsao, type="b", ylab="Temperatura", xlab="Ano", main="")
lines(cananeia_novos_ts, col="red", type="b")

```



```

# calculo do erro

# Raiz do Erro Quadratico Medio (REQM / RMSE)
RMSE <- sqrt(mean((cananeia_novos_ts - previsao$mean)^2))

# Erro Absoluto Medio (EAM / MAE)
MAE <- mean(abs(cananeia_novos_ts - previsao$mean))

# Erro Percentual Absoluto Medio (MAPE)
MAPE <- mean(abs(cananeia_novos_ts - previsao$mean) / abs(cananeia_novos_ts)) * 100

cat("RMSE:", RMSE, "\n")

## RMSE: 0.7218803
cat("MAE:", MAE, "\n")

## MAE: 0.5185185
cat("MAPE:", MAPE, "%\n")

## MAPE: 2.354584 %

```

## Exemplo de uma série temporal com tendência e sazonalidade

A seguir apresentamos as temperaturas médias mensais, em graus centígrados, da cidade de Ubatuba (município brasileiro do litoral de São Paulo), de janeiro de 1976 a dezembro de 1985. Série temporal retirada de Morettin e Toloi (2006).

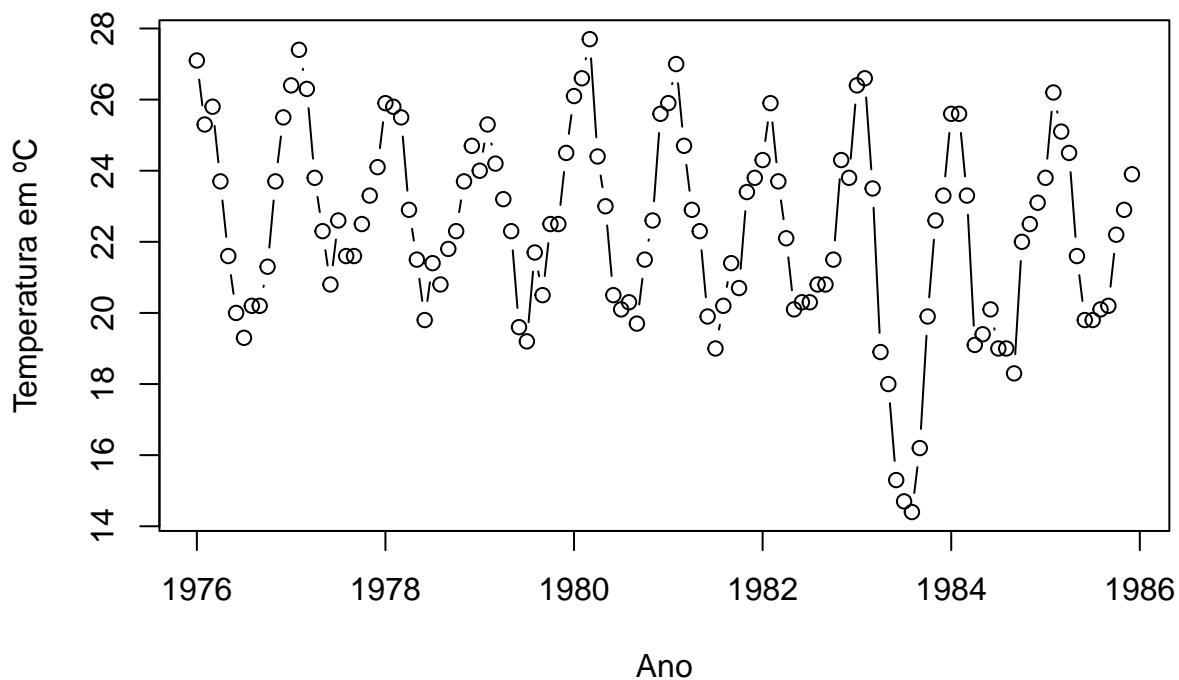
```
# fazendo a leitura do conjunto de dados
setwd("G:\\Meu Drive\\UFG\\Especializacao\\Aulas de series temporais\\Códigos")

library(readxl)
temperatura <- read_excel("temperatura.xls")
head(temperatura)

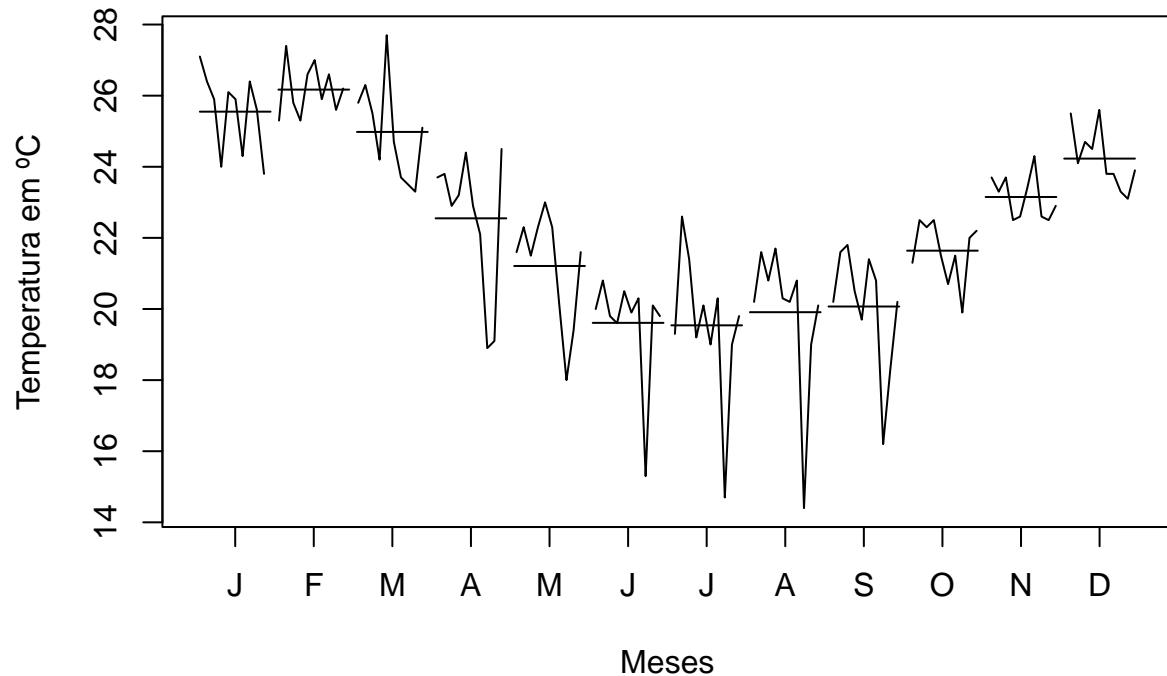
## # A tibble: 6 x 3
##   Ano Cananeia Ubatuba
##   <dbl>    <dbl>    <dbl>
## 1 1976     25.2     27.1
## 2 NA       24.3     25.3
## 3 NA       24.3     25.8
## 4 NA       21.4     23.7
## 5 NA       19.8     21.6
## 6 NA       17.0      20.0

temp.ubatuba_ts <- ts(temperatura$Ubatuba, start = c(1976, 1), frequency = 12)

# grafico da serie temporal
plot.ts(temp.ubatuba_ts, type="b", ylab="Temperatura em °C", xlab="Ano")
```

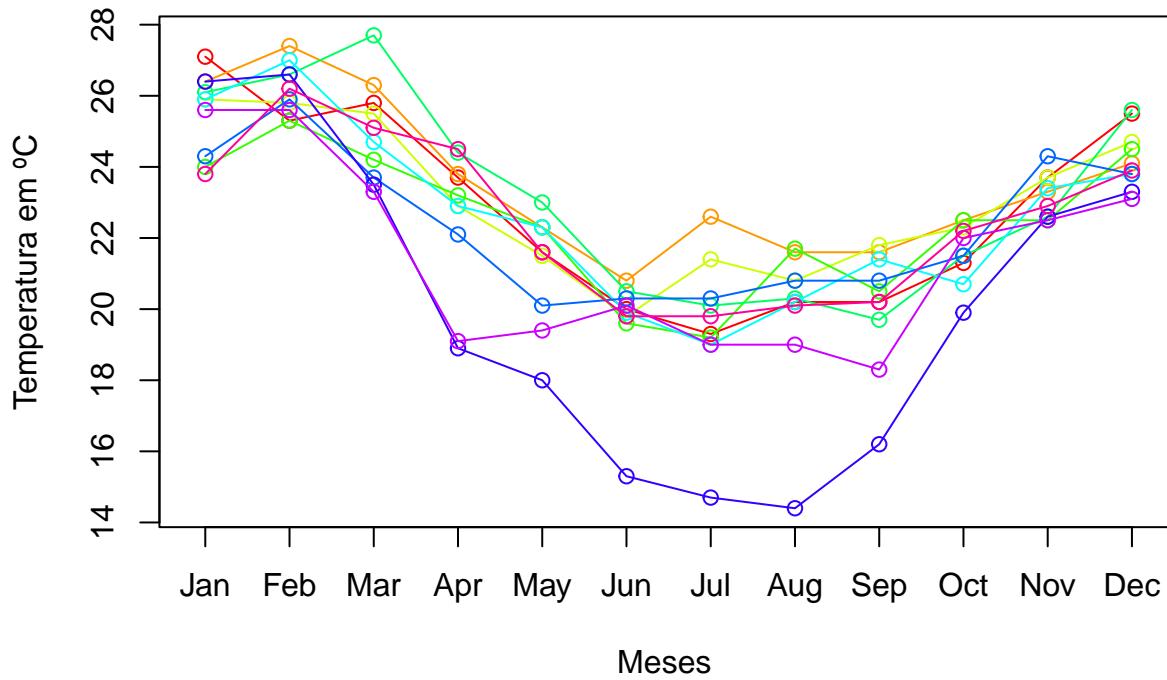


```
# grafico dos meses separadamente
monthplot(temp.ubatuba_ts, xlab="Meses", ylab="Temperatura em °C", main="")
```



```
# grafico dos anos separadamente
cores <- rainbow(10)
anos_legenda <- as.character(1976:1985)

seasonplot(temp.ubatuba_ts, xlab="Meses", ylab="Temperatura em °C", main="", col=cores)
```

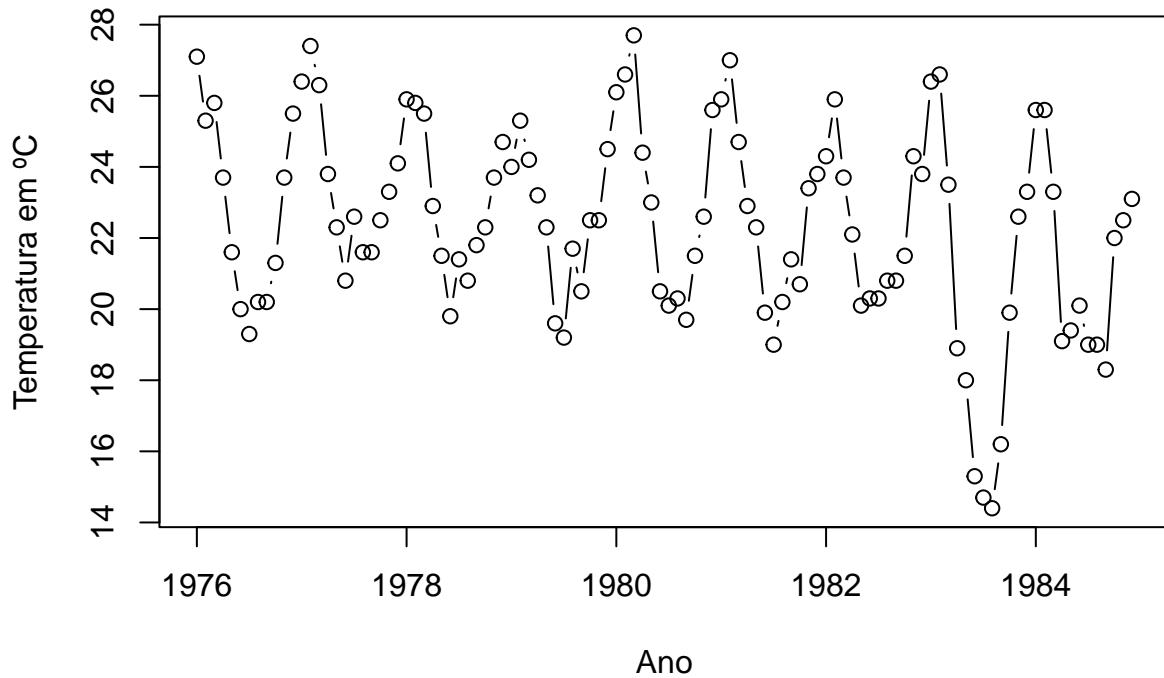


```
#ylim=c(0, 30)
#legend(7, 17, lty=1, col=cores[1:5], anos_legenda[1:5], bty = "n")
#legend(10, 17, lty=1, col=cores[6:10], anos_legenda[6:10], bty = "n")

# ajuste do modelo de regressão com variáveis *dummies* para capturar a sazonalidade
# para a tendência um polinomio de grau 1
# vamos utilizar 9 anos
# deixando o ultimo ano para realizar comparações

trein_temp.ubatuba_ts <- ts(temperatura$Ubatuba[1:108], start = c(1976, 1), frequency = 12)

# gráfico da série temporal
plot(trein_temp.ubatuba_ts, type="b", ylab="Temperatura em °C", xlab="Ano")
```



```
# ajuste
ajuste1 <- tslm(trein_temp.ubatuba_ts ~ trend + season)
summary(ajuste1)
```

```
##
## Call:
## tslm(formula = trein_temp.ubatuba_ts ~ trend + season)
##
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -4.6131 -0.5928  0.1278  0.6369  2.7333 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 26.93655  0.48026  56.087 < 2e-16 ***
## trend      -0.02433  0.00407 -5.977 3.96e-08 ***
## season2    0.44655  0.61787  0.723  0.471622    
## season3   -0.72912  0.61791 -1.180  0.240954    
## season4   -3.33812  0.61798 -5.402 4.87e-07 ***
## season5   -4.48046  0.61807 -7.249 1.11e-10 ***
## season6   -6.03391  0.61819 -9.761 5.37e-16 ***
## season7   -6.08736  0.61834 -9.845 3.55e-16 ***
## season8   -5.68525  0.61851 -9.192 8.82e-15 ***
## season9   -5.49426  0.61871 -8.880 4.08e-14 ***
## season10  -3.94771  0.61894 -6.378 6.47e-09 ***
## season11  -2.32338  0.61919 -3.752 0.000302 ***
```

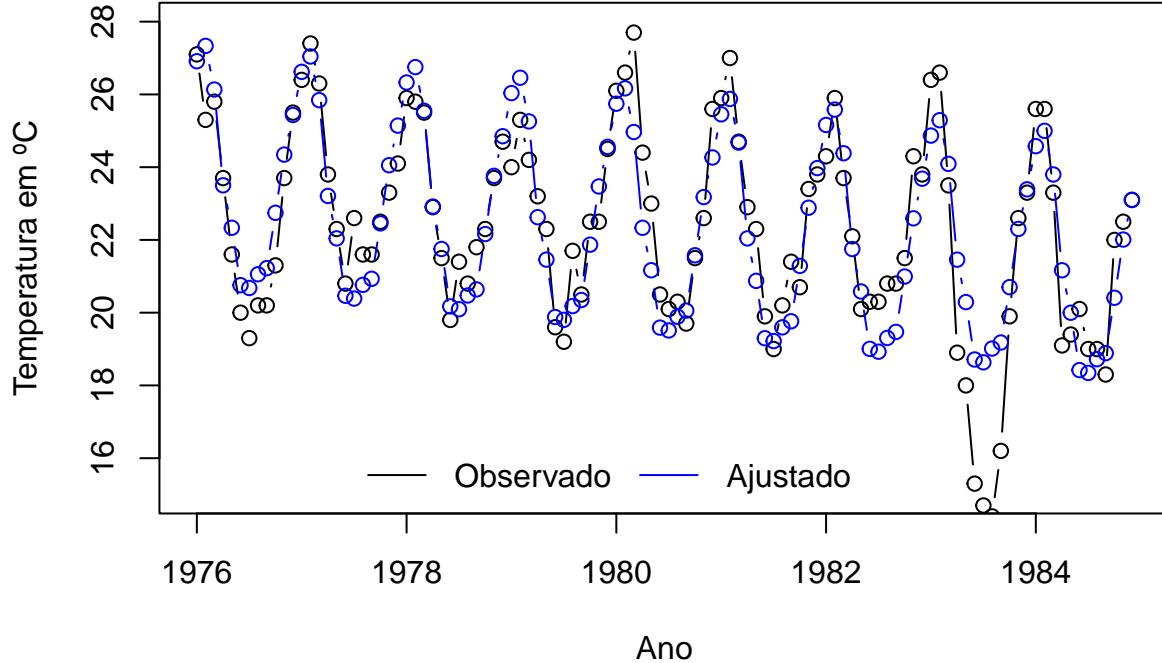
```

## season12      -1.21016      0.61947   -1.954  0.053699 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.311 on 95 degrees of freedom
## Multiple R-squared:  0.8013, Adjusted R-squared:  0.7762
## F-statistic: 31.92 on 12 and 95 DF,  p-value: < 2.2e-16
# qual a matriz que estou considerando?
X <- model.matrix(ajuste1)
head(X)

##          Jan Feb Mar Apr May Jun
## 1976 27.1 25.3 25.8 23.7 21.6 20.0
# esse comando nao funciona :(

# grafico da serie temporal
plot(trein_temp.ubatuba_ts, type="b", ylab="Temperatura em °C", xlab="Ano", ylim=c(15, 28))
# valores ajustados
lines(ajuste1$fitted.values, type="b", col="blue")
legend("bottomright", lty=c(1,1), col=c("black", "blue"), c("Observado", "Ajustado"),
       bty = "n", ncol=3)

```



```

# serie livre de sazonalidade e tendencia
ubatuba_ajustada <- ajuste1$fitted.values

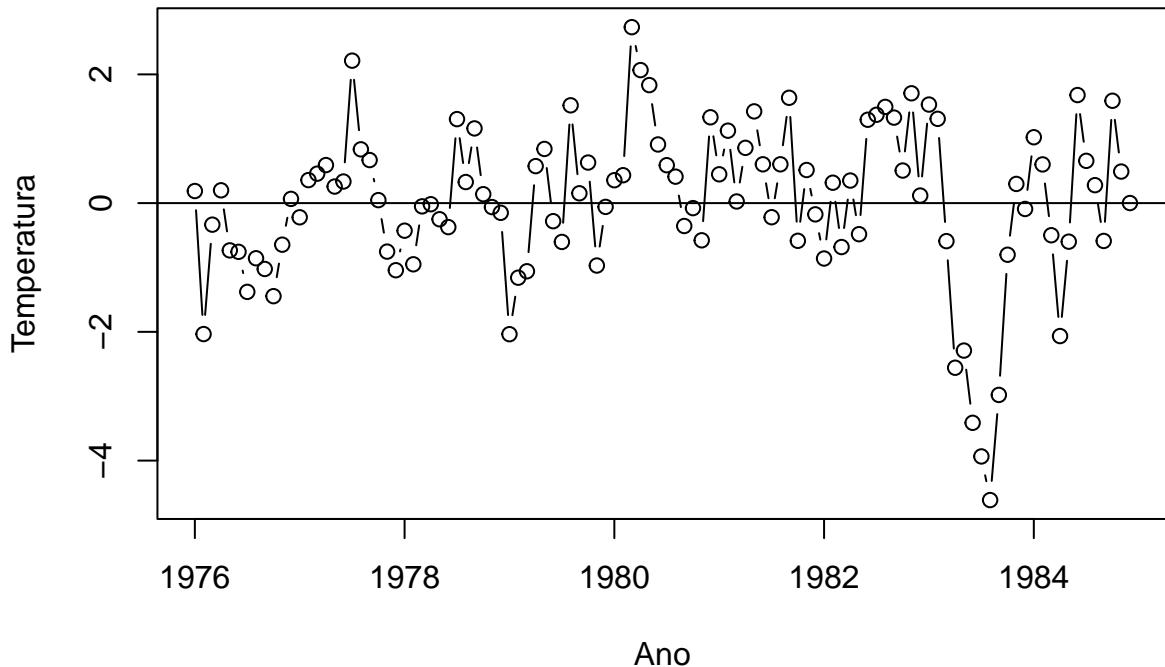
```

```

ubatuba_ss_ts <- trein_temp.ubatuba_ts - ubatuba_ajustada

# grafico da serie livre de sazonalidade
plot(ubatuba_ss_ts, type='b', ylab="Temperatura", xlab="Ano")
abline(h=0)

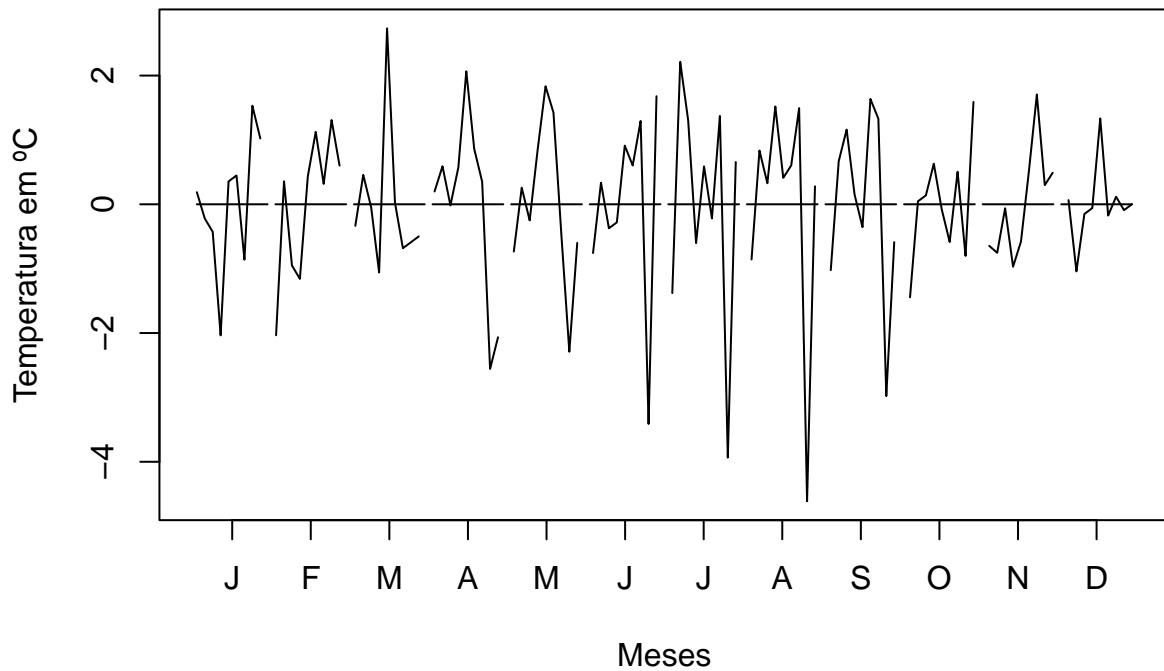
```



```

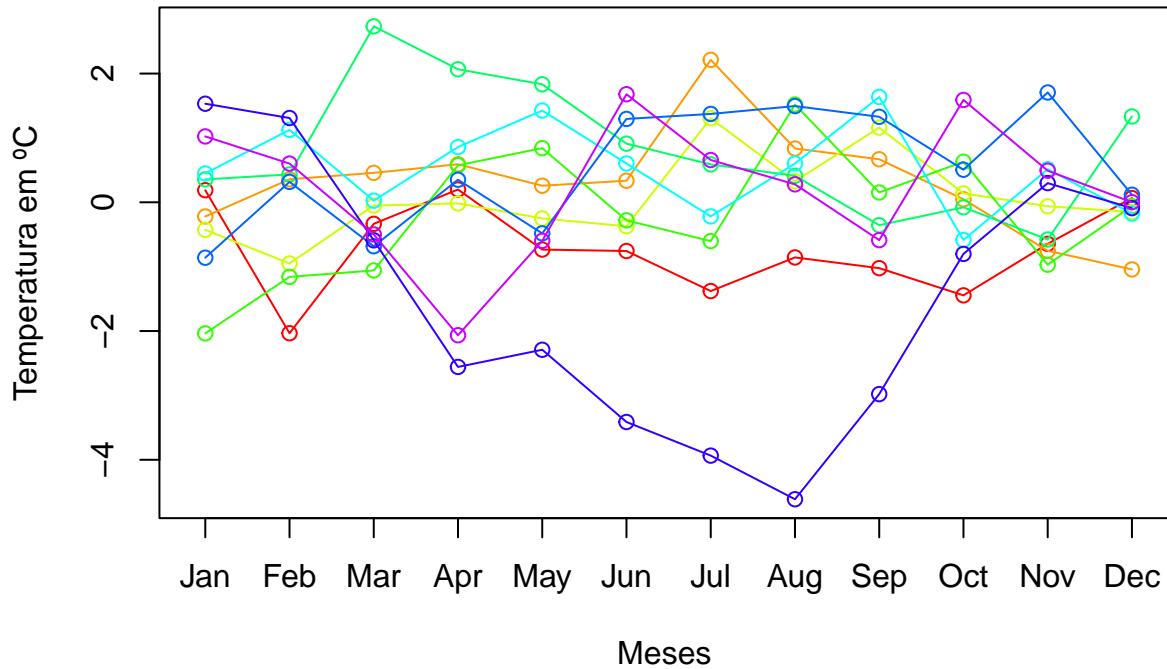
# grafico dos meses separadamente
monthplot(ubatuba_ss_ts, xlab="Meses", ylab="Temperatura em °C", main="")

```



```
# grafico dos anos separadamente
cores <- rainbow(10)
anos_legenda <- as.character(1976:1985)

seasonplot(ubatuba_ss_ts, xlab="Meses", ylab="Temperatura em °C", main="", col=cores)
```



```
# previsoes para $h=1, 2, \ldots, 12$.
```

```
previsao <- forecast(ajuste1, h = 12)
previsao
```

	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
## Jan 1985	24.28472	22.47413	26.09531	21.49944	27.07000
## Feb 1985	24.70694	22.89636	26.51753	21.92167	27.49222
## Mar 1985	23.50694	21.69636	25.31753	20.72167	26.29222
## Apr 1985	20.87361	19.06302	22.68420	18.08833	23.65889
## May 1985	19.70694	17.89636	21.51753	16.92167	22.49222
## Jun 1985	18.12917	16.31858	19.93976	15.34389	20.91444
## Jul 1985	18.05139	16.24080	19.86198	15.26611	20.83667
## Aug 1985	18.42917	16.61858	20.23976	15.64389	21.21444
## Sep 1985	18.59583	16.78524	20.40642	15.81056	21.38111
## Oct 1985	20.11806	18.30747	21.92864	17.33278	22.90333
## Nov 1985	21.71806	19.90747	23.52864	18.93278	24.50333
## Dec 1985	22.80694	20.99636	24.61753	20.02167	25.59222

```
# como e feita a estimacao pontual
# h=1 - mes de janeiro
x <- c(1, 109, rep(0, 11))
# - 1 para o intercepto
# - 109 para a tendencia
# - 11 vezes 0, pois e o mes de janeiro (mes de referencia)
```

```
ajuste1$coefficients %*% x
```

```

##           [,1]
## [1,] 24.28472

# h=2 - mes de fevereiro
x <- c(1, 110, 1, rep(0, 10))
# - 1 para o intercepto
# - 110 para a tendencia
# - 1 para o mes de fevereiro
# - 10 vezes 0, pois apenas o mes de fevereiro e diferente de 1

ajuste1$coefficients%*%x

##           [,1]
## [1,] 24.70694

# grafico com os valores observados e preditos

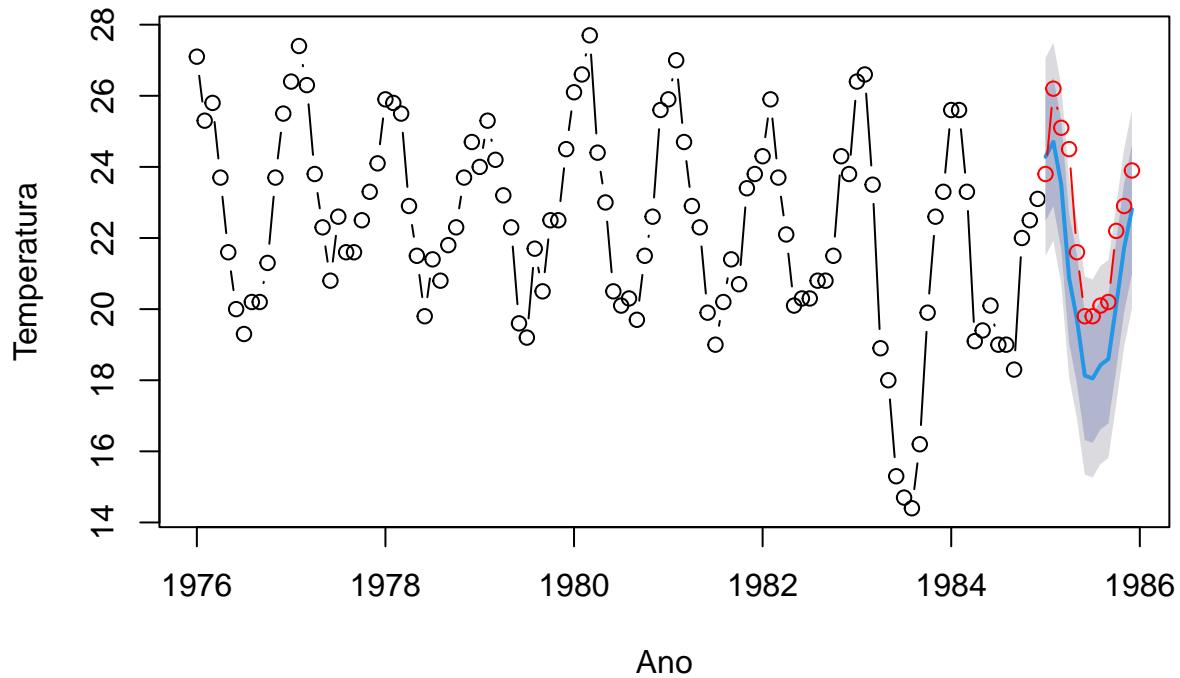
ubatuba_novos_ts <- ts(temperatura$Ubatuba[109:120], start = c(1985, 1), frequency = 12)

round(cbind(ubatuba_novos_ts, previsao$mean), 1)

##      ubatuba_novos_ts previsao$mean
## Jan 1985          23.8          24.3
## Feb 1985          26.2          24.7
## Mar 1985          25.1          23.5
## Apr 1985          24.5          20.9
## May 1985          21.6          19.7
## Jun 1985          19.8          18.1
## Jul 1985          19.8          18.1
## Aug 1985          20.1          18.4
## Sep 1985          20.2          18.6
## Oct 1985          22.2          20.1
## Nov 1985          22.9          21.7
## Dec 1985          23.9          22.8

plot(previsao, type="b", ylab="Temperatura", xlab="Ano", main="")
lines(ubatuba_novos_ts, col="red", type="b")

```



```

# calculo do erro

# Raiz do Erro Quadratico Medio (REQM / RMSE)
RMSE <- sqrt(mean((ubatuba_novos_ts - previsao$mean)^2))

# Erro Absoluto Medio (EAM / MAE)
MAE <- mean(abs(ubatuba_novos_ts - previsao$mean))

# Erro Percentual Absoluto Medio (MAPE)
MAPE <- mean(abs(ubatuba_novos_ts - previsao$mean) / abs(ubatuba_novos_ts)) * 100

cat("RMSE:", RMSE, "\n")

## RMSE: 1.822932
cat("MAE:", MAE, "\n")

## MAE: 1.678472
cat("MAPE:", MAPE, "%\n")

## MAPE: 7.52373 %

```

## Utilizando funções seno e cosseno para a sazonalidade

```

# exemplo da construcao da matriz do modelo
n <- length(temperatura$Cananeia[1:108])
x_beta0 <- rep(1, n, 1) # intercepto
x_beta1 <- seq(1, n, 1) # tendencia linear
x_beta2 <- x_beta1^2 # tendencia quadratica
x_sen12 <- sin(2*pi*x_beta1/12) # seno com periodo 12
x_cos12 <- cos(2*pi*x_beta1/12) # cosseno com periodo 12
x_sen6 <- sin(2*pi*x_beta1/6) # seno com periodo 6
x_cos6 <- cos(2*pi*x_beta1/6) # cosseno com periodo 6

x <- cbind(x_beta0, x_beta1, x_beta2, x_sen12, x_cos12, x_sen6, x_cos6)
head(x)

##      x_beta0 x_beta1 x_beta2      x_sen12      x_cos12      x_sen6 x_cos6
## [1,]      1       1      1 5.000000e-01 8.660254e-01 8.660254e-01   0.5
## [2,]      1       2      4 8.660254e-01 5.000000e-01 8.660254e-01  -0.5
## [3,]      1       3      9 1.000000e+00 6.123032e-17 1.224606e-16  -1.0
## [4,]      1       4     16 8.660254e-01 -5.000000e-01 -8.660254e-01  -0.5
## [5,]      1       5     25 5.000000e-01 -8.660254e-01 -8.660254e-01   0.5
## [6,]      1       6     36 1.224606e-16 -1.000000e+00 -2.449213e-16   1.0
#x[1:15,]

head(fourier(temp.cananeia_ts, K = 1))

##          S1-12      C1-12
## [1,] 0.5000000 0.8660254
## [2,] 0.8660254 0.5000000
## [3,] 1.0000000 0.0000000
## [4,] 0.8660254 -0.5000000
## [5,] 0.5000000 -0.8660254
## [6,] 0.0000000 -1.0000000

head(fourier(temp.cananeia_ts, K = 2))

##          S1-12      C1-12      S2-12 C2-12
## [1,] 0.5000000 0.8660254 0.8660254   0.5
## [2,] 0.8660254 0.5000000 0.8660254  -0.5
## [3,] 1.0000000 0.0000000 0.0000000  -1.0
## [4,] 0.8660254 -0.5000000 -0.8660254  -0.5
## [5,] 0.5000000 -0.8660254 -0.8660254   0.5
## [6,] 0.0000000 -1.0000000  0.0000000   1.0

# ajuste

trein_temp.cananeia_ts <- ts(temperatura$Cananeia[1:108], start = c(1976, 1), frequency = 12)

ajuste1 <- tslm(trein_temp.cananeia_ts ~ trend + I(trend^2) + fourier(trein_temp.cananeia_ts, K = 2))
summary(ajuste1)

## 
## Call:
## tslm(formula = trein_temp.cananeia_ts ~ trend + I(trend^2) +
##       fourier(trein_temp.cananeia_ts, K = 2))
## 
```

```

## Residuals:
##      Min     1Q   Median     3Q    Max
## -2.12923 -0.56552 -0.07132  0.67875 3.07772
##
## Coefficients:
##                               Estimate Std. Error t value
## (Intercept)                2.156e+01  2.880e-01 74.859
## trend                  -3.694e-03  1.218e-02 -0.303
## I(trend^2)                3.675e-05  1.083e-04  0.340
## fourier(trein_temp.cananeia_ts, K = 2)S1-12 2.565e+00  1.335e-01 19.210
## fourier(trein_temp.cananeia_ts, K = 2)C1-12 2.755e+00  1.331e-01 20.696
## fourier(trein_temp.cananeia_ts, K = 2)S2-12 1.170e-02  1.332e-01  0.088
## fourier(trein_temp.cananeia_ts, K = 2)C2-12 -2.495e-01  1.331e-01 -1.875
##                                     Pr(>|t|)
## (Intercept) <2e-16 ***
## trend        0.7623
## I(trend^2)  0.7349
## fourier(trein_temp.cananeia_ts, K = 2)S1-12 <2e-16 ***
## fourier(trein_temp.cananeia_ts, K = 2)C1-12 <2e-16 ***
## fourier(trein_temp.cananeia_ts, K = 2)S2-12  0.9301
## fourier(trein_temp.cananeia_ts, K = 2)C2-12  0.0637 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9778 on 101 degrees of freedom
## Multiple R-squared:  0.8884, Adjusted R-squared:  0.8818
## F-statistic: 134 on 6 and 101 DF, p-value: < 2.2e-16
AIC(ajuste1)

## [1] 310.3946
ajuste2 <- tslm(trein_temp.cananeia_ts ~ trend + fourier(trein_temp.cananeia_ts, K = 2))
summary(ajuste2)

##
## Call:
## tslm(formula = trein_temp.cananeia_ts ~ trend + fourier(trein_temp.cananeia_ts,
## K = 2))
##
## Residuals:
##      Min     1Q   Median     3Q    Max
## -2.15431 -0.58929 -0.07231  0.66840 3.08872
##
## Coefficients:
##                               Estimate Std. Error t value
## (Intercept)                21.4885749  0.1893576 113.481
## trend                     0.0003116  0.0030195  0.103
## fourier(trein_temp.cananeia_ts, K = 2)S1-12 2.5654324  0.1329559 19.295
## fourier(trein_temp.cananeia_ts, K = 2)C1-12 2.7551107  0.1325118 20.791
## fourier(trein_temp.cananeia_ts, K = 2)S2-12  0.0117659  0.1325806  0.089
## fourier(trein_temp.cananeia_ts, K = 2)C2-12 -0.2493856  0.1325118 -1.882
##                                     Pr(>|t|)
## (Intercept) <2e-16 ***
## trend        0.9180

```

```

## fourier(trein_temp.cananeia_ts, K = 2)S1-12 <2e-16 ***
## fourier(trein_temp.cananeia_ts, K = 2)C1-12 <2e-16 ***
## fourier(trein_temp.cananeia_ts, K = 2)S2-12 0.9295
## fourier(trein_temp.cananeia_ts, K = 2)C2-12 0.0627 .
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9735 on 102 degrees of freedom
## Multiple R-squared: 0.8883, Adjusted R-squared: 0.8828
## F-statistic: 162.2 on 5 and 102 DF, p-value: < 2.2e-16
AIC(ajuste2)

## [1] 308.5178

ajuste3 <- tslm(trein_temp.cananeia_ts ~ fourier(trein_temp.cananeia_ts, K = 2))
summary(ajuste3)

##
## Call:
## tslm(formula = trein_temp.cananeia_ts ~ fourier(trein_temp.cananeia_ts,
##       K = 2))
##
## Residuals:
##      Min      1Q      Median      3Q      Max
## -2.15914 -0.58860 -0.06728  0.67746  3.07766
##
## Coefficients:
##                               Estimate Std. Error t value
## (Intercept)                21.50556   0.09322 230.685
## fourier(trein_temp.cananeia_ts, K = 2)S1-12  2.56427   0.13184 19.450
## fourier(trein_temp.cananeia_ts, K = 2)C1-12  2.75542   0.13184 20.900
## fourier(trein_temp.cananeia_ts, K = 2)S2-12  0.01123   0.13184  0.085
## fourier(trein_temp.cananeia_ts, K = 2)C2-12 -0.24907   0.13184 -1.889
##                                         Pr(>|t|)
## (Intercept)                <2e-16 ***
## fourier(trein_temp.cananeia_ts, K = 2)S1-12 <2e-16 ***
## fourier(trein_temp.cananeia_ts, K = 2)C1-12 <2e-16 ***
## fourier(trein_temp.cananeia_ts, K = 2)S2-12  0.9323
## fourier(trein_temp.cananeia_ts, K = 2)C2-12  0.0617 .
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9688 on 103 degrees of freedom
## Multiple R-squared: 0.8882, Adjusted R-squared: 0.8839
## F-statistic: 204.7 on 4 and 103 DF, p-value: < 2.2e-16
AIC(ajuste3)

## [1] 306.5291

ajuste4 <- tslm(trein_temp.cananeia_ts ~ fourier(trein_temp.cananeia_ts, K = 1))
summary(ajuste4)

##
## Call:
## tslm(formula = trein_temp.cananeia_ts ~ fourier(trein_temp.cananeia_ts,

```

```

##      K = 1))
##
## Residuals:
##      Min     1Q Median     3Q    Max
## -2.2740 -0.6123 -0.1056  0.7521  2.9628
##
## Coefficients:
##                               Estimate Std. Error t value
## (Intercept)                21.50556   0.09392 228.97
## fourier(trein_temp.cananeia_ts, K = 1)S1-12  2.56427   0.13283 19.31
## fourier(trein_temp.cananeia_ts, K = 1)C1-12  2.75542   0.13283 20.75
##                                     Pr(>|t|)
## (Intercept)                <2e-16 ***
## fourier(trein_temp.cananeia_ts, K = 1)S1-12  <2e-16 ***
## fourier(trein_temp.cananeia_ts, K = 1)C1-12  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9761 on 105 degrees of freedom
## Multiple R-squared:  0.8844, Adjusted R-squared:  0.8822
## F-statistic: 401.5 on 2 and 105 DF,  p-value: < 2.2e-16
AIC(ajuste4)

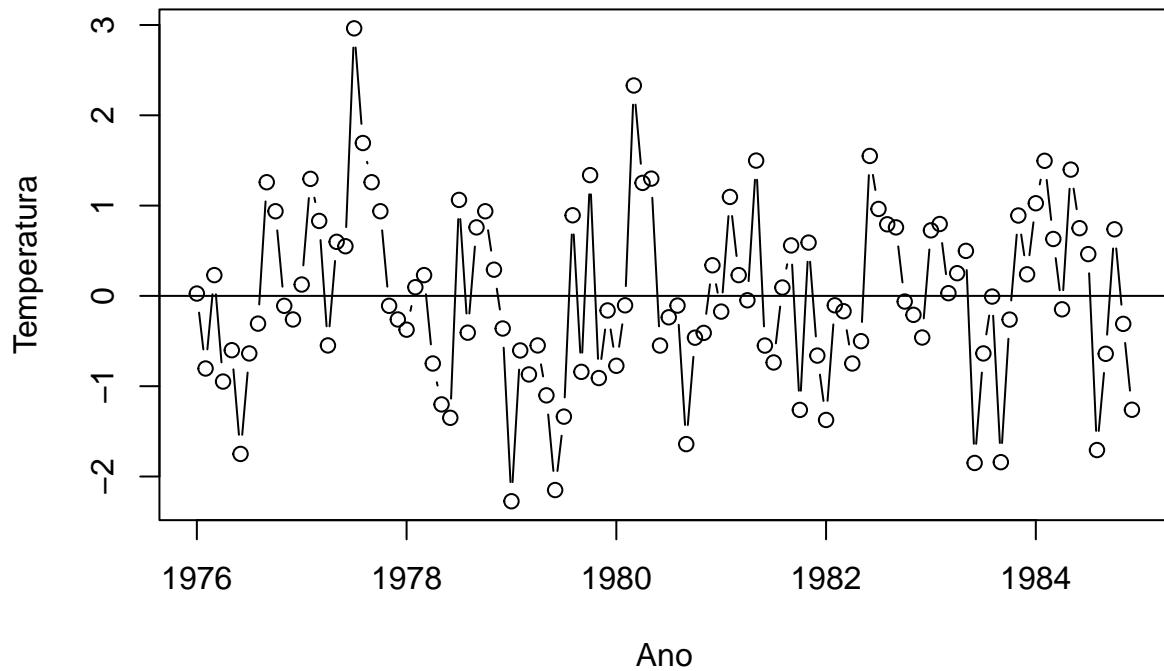
## [1] 306.2155
# utilizando o AIC
# o melhor modelo é o do ajuste4

cananeia_ajus_f <- ajuste4$fitted.values

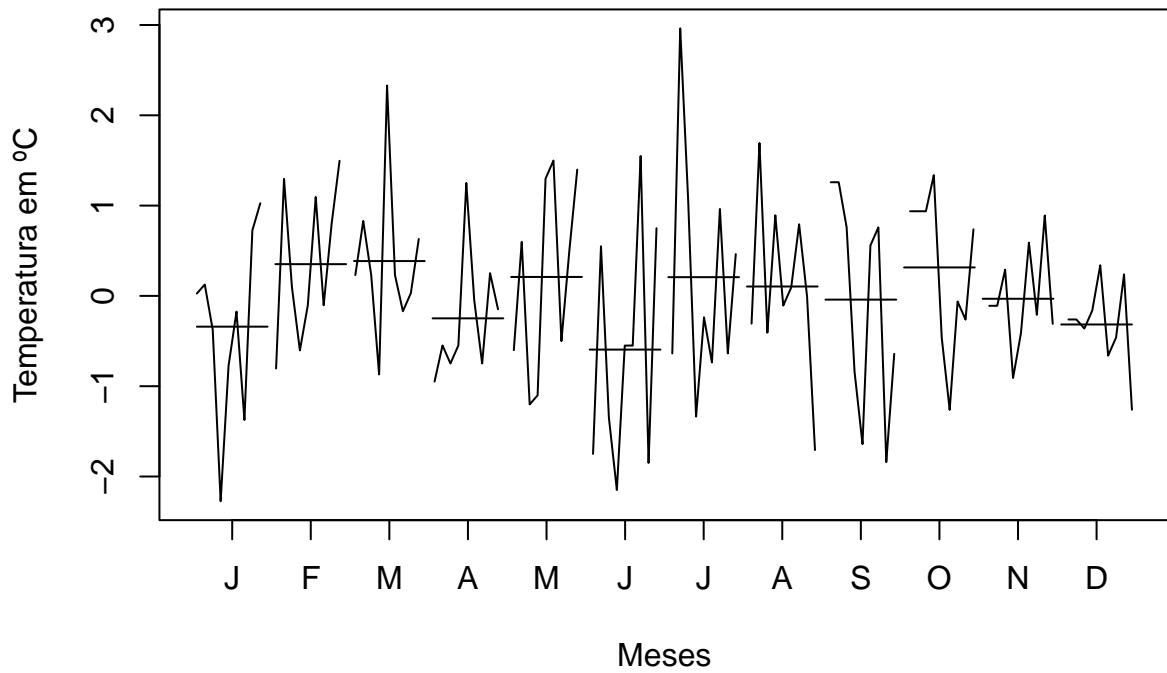
cananeia_ss_ts <- trein_temp.cananeia_ts - cananeia_ajus_f

# grafico da serie livre de sazonalidade
plot(cananeia_ss_ts, type='b', ylab="Temperatura", xlab="Ano")
abline(h=0)

```

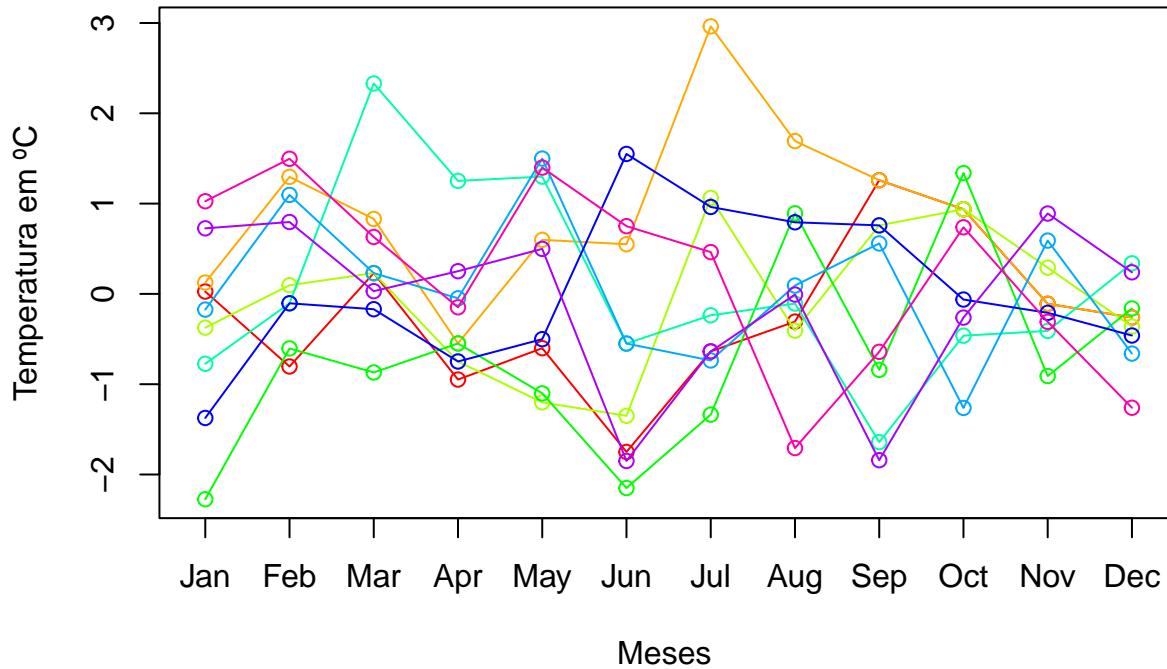


```
# grafico dos meses separadamente
monthplot(cananeia_ss_ts, xlab="Meses", ylab="Temperatura em °C", main="")
```



```
# grafico dos anos separadamente
cores <- rainbow(9)
anos_legenda <- as.character(1976:1984)

seasonplot(cananeia_ss_ts, xlab="Meses", ylab="Temperatura em °C", main="", col=cores)
```



```

# previsao

temp.cananeia_previsao <- ts(1:12, start = c(1985, 1), frequency = 12)
fc_future <- fourier(temp.cananeia_previsao, K = 1)

previsao <- forecast(ajuste4, h = 12, ts = T, newdata = fc_future)

## Warning in forecast.lm(ajuste4, h = 12, ts = T, newdata = fc_future): newdata
## column names not specified, defaulting to first variable required.

previsao

##           Point Forecast     Lo 80      Hi 80      Lo 95      Hi 95
## Jan 1985    25.17396  23.89779  26.45012  23.21190  27.13601
## Feb 1985    25.10399  23.82783  26.38015  23.14194  27.06604
## Mar 1985    24.06983  22.79366  25.34599  22.10777  26.03188
## Apr 1985    22.34857  21.07241  23.62473  20.38651  24.31062
## May 1985    20.40142  19.12526  21.67759  18.43937  22.36348
## Jun 1985    18.75013  17.47397  20.02630  16.78808  20.71219
## Jul 1985    17.83716  16.56099  19.11332  15.87510  19.79921
## Aug 1985    17.90712  16.63096  19.18328  15.94507  19.86917
## Sep 1985    18.94129  17.66512  20.21745  16.97923  20.90334
## Oct 1985    20.66254  19.38638  21.93871  18.70049  22.62460
## Nov 1985    22.60969  21.33352  23.88585  20.64763  24.57174
## Dec 1985    24.26098  22.98482  25.53714  22.29893  26.22303

# como e feita a estimacao pontual
# h=1 - mes de janeiro

```

```

x_sen12 <- sin(2*pi*109/12) # seno com periodo 12
x_cos12 <- cos(2*pi*109/12) # cosseno com periodo 12

ajuste4$coefficients%*%c(1, x_sen12, x_cos12)

##           [,1]
## [1,] 25.17396

# h=2 - mes de fevereiro
x_sen12 <- sin(2*pi*110/12) # seno com periodo 12
x_cos12 <- cos(2*pi*110/12) # cosseno com periodo 12

ajuste4$coefficients%*%c(1, x_sen12, x_cos12)

##           [,1]
## [1,] 25.10399

# grafico com os valores observados e preditos

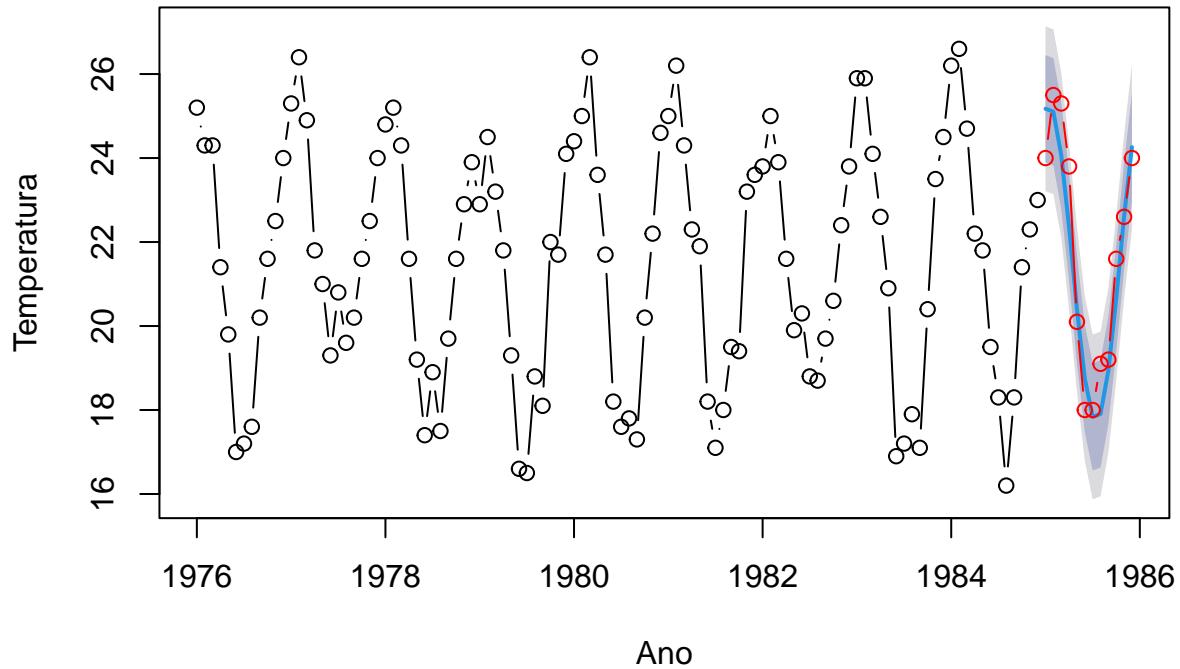
cananeia_novos_ts <- ts(temperatura$Cananeia[109:120], start = c(1985, 1), frequency = 12)

round(cbind(cananeia_novos_ts, previsao$mean), 1)

##      cananeia_novos_ts previsao$mean
## Jan 1985          24.0          25.2
## Feb 1985          25.5          25.1
## Mar 1985          25.3          24.1
## Apr 1985          23.8          22.3
## May 1985          20.1          20.4
## Jun 1985          18.0          18.8
## Jul 1985          18.0          17.8
## Aug 1985          19.1          17.9
## Sep 1985          19.2          18.9
## Oct 1985          21.6          20.7
## Nov 1985          22.6          22.6
## Dec 1985          24.0          24.3

plot(previsao, type="b", ylab="Temperatura", xlab="Ano", main="")
lines(cananeia_novos_ts, col="red", type="b")

```



```

# calculo do erro

# Raiz do Erro Quadratico Medio (REQM / RMSE)
RMSE <- sqrt(mean((cananeia_novos_ts - previsao$mean)^2))

# Erro Absoluto Medio (EAM / MAE)
MAE <- mean(abs(cananeia_novos_ts - previsao$mean))

# Erro Percentual Absoluto Medio (MAPE)
MAPE <- mean(abs(cananeia_novos_ts - previsao$mean) / abs(cananeia_novos_ts)) * 100

cat("RMSE:", RMSE, "\n")

## RMSE: 0.8302599
cat("MAE:", MAE, "\n")

## MAE: 0.6771408
cat("MAPE:", MAPE, "%\n")

## MAPE: 3.086687 %

```

Neste momento temos dois modelos ajustados para a série temporal de temperaturas médias de Cananéia. O primeiro usando variáveis *dummies* para capturar a sazonalidade, utilizando 11 parâmetros para isso, enquanto que o segundo modelo utilizou de uma função seno e uma cosseno, requerendo apenas dois parâmetros. A pergunta natural nesse momento é: entre os dois, qual o melhor modelo? A resposta é: depende!

Se o nosso interesse é buscar um modelo mais parcimonioso, que explica bem as flutuações dos meses, o nosso

critério de comparação poderia ser o AIC, sendo que no primeiro modelo é 312,9199, e do segundo, 306,2155, indicando assim que o modelo com menor quantidade de parâmetros a serem estimados é o melhor.

No entanto, se nosso enfoque for o preditivo, o modelo 1 apresenta menores valores para os critérios RMSE, MAE e MAPE, com valores de 0,7219, 0,5185 e 2,35%, respectivamente; enquanto o modelo 2, apresenta os valores 0,8303, 0,6771 e 3,09%, respectivamente.