Análise de Séries Temporais

1.0 - Aula Prática

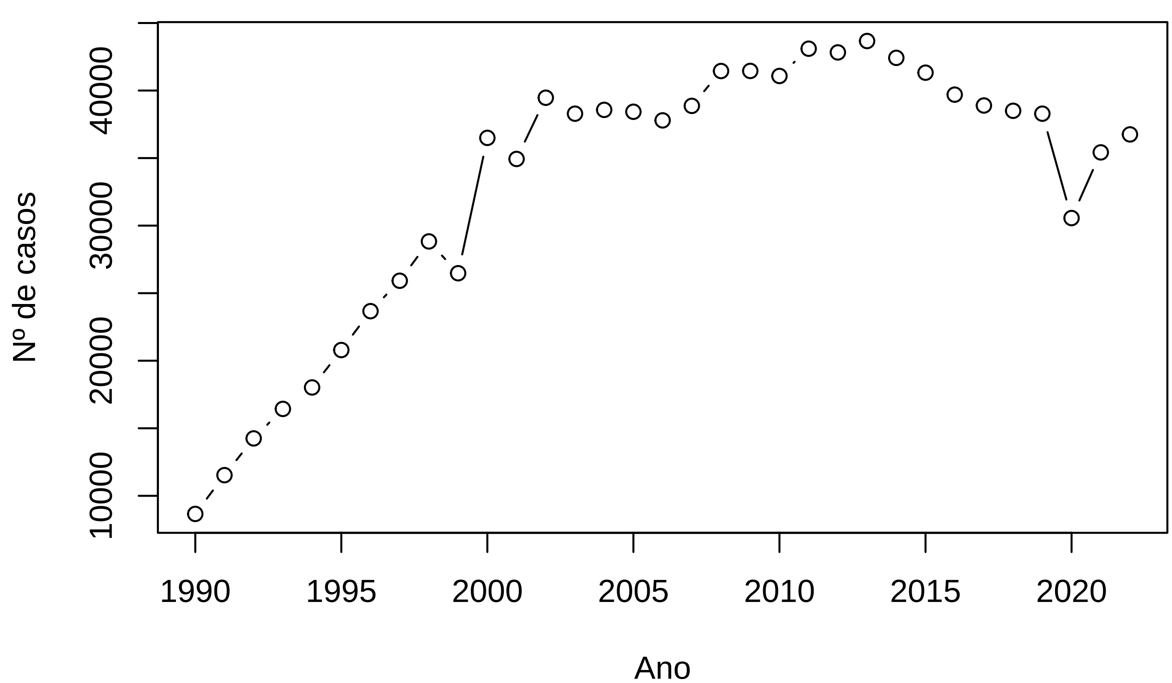
Prof. Dr. Eder Angelo Milani

20/06/2025

Exemplo de Modelo ARIMA

A série temporal em estudo é do número anual de casos de AIDS notificados no SINAN, declarados no SIM e registrados no SISCEL/SICLOM, por ano de diagnóstico no Brasil, de 1990 a 2022.

```
serie_AIDS <- c(8660, 11530, 14250, 16433, 18022, 20792, 23670, 25921, 28832, 26477,
                36497, 34936, 39467, 38287, 38569, 38430, 37792, 38865, 41445, 41451,
                41079, 43098, 42823, 43666, 42421, 41323, 39696, 38893, 38501, 38288,
                30562, 35424, 36753)

serie_AIDS_ts <- ts(serie_AIDS, start= 1990, frequency = 1)

plot(serie_AIDS_ts, type="b", ylab="Nº de casos", xlab="Ano")
```



```r
#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
```

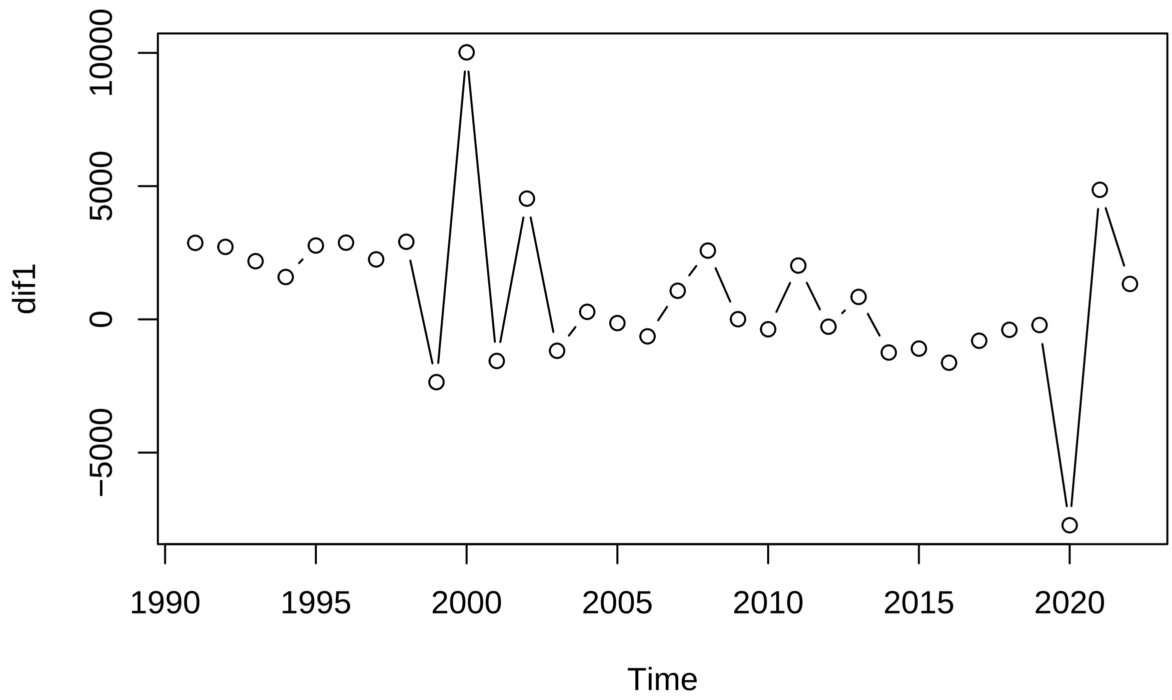
```r
# teste de raiz unitaria
#install.packages("tseries")
library(tseries)
```

```
## Warning: pacote 'tseries' foi compilado no R versão 4.4.3
```

```r
adf.test(serie_AIDS_ts)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  serie_AIDS_ts
## Dickey-Fuller = -0.7542, Lag order = 3, p-value = 0.9555
## alternative hypothesis: stationary
```

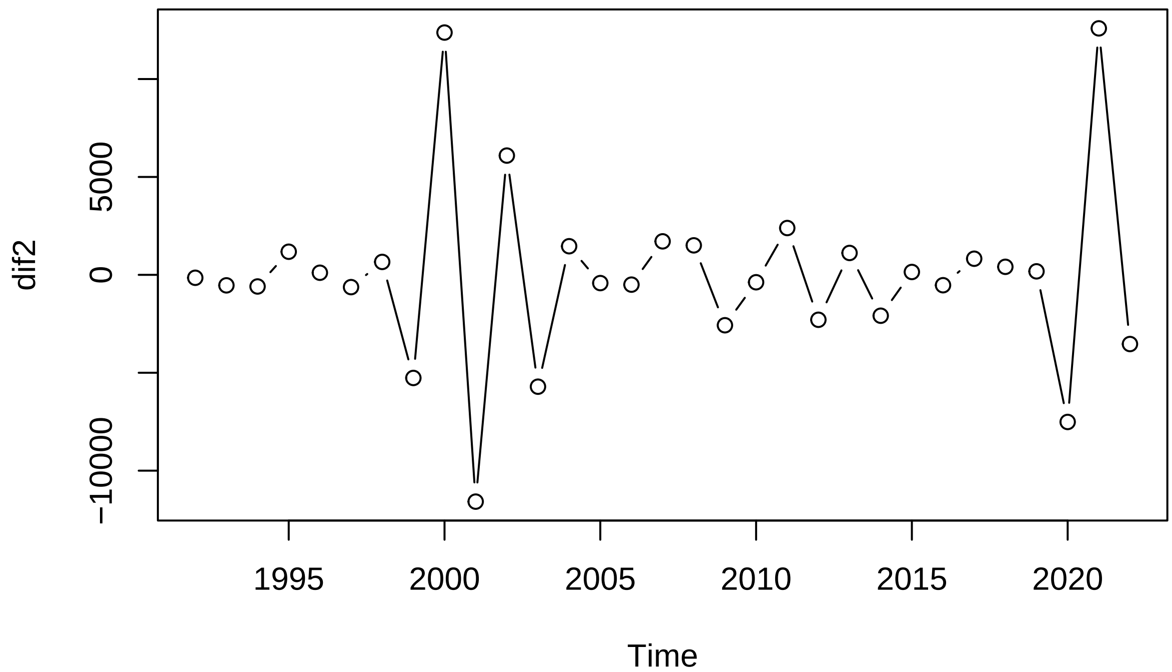
```r
# Hipoteses do teste:
# H0 (hipotese nula): a serie nao e estacionaria (possui raiz unitaria).
# H1 (hipotese alternativa): a serie e estacionaria (nao possui raiz unitaria)


# p-valor = 0.9555
# existe evidencias a favor da hipotese nula
# que a serie nao e estacionaria

# primeira diferenca da serie
dif1 <- diff(serie_AIDS_ts)
plot(dif1, type="b")
```




```
adf.test(dif1)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  dif1
## Dickey-Fuller = -3.1133, Lag order = 3, p-value = 0.1433
## alternative hypothesis: stationary
```

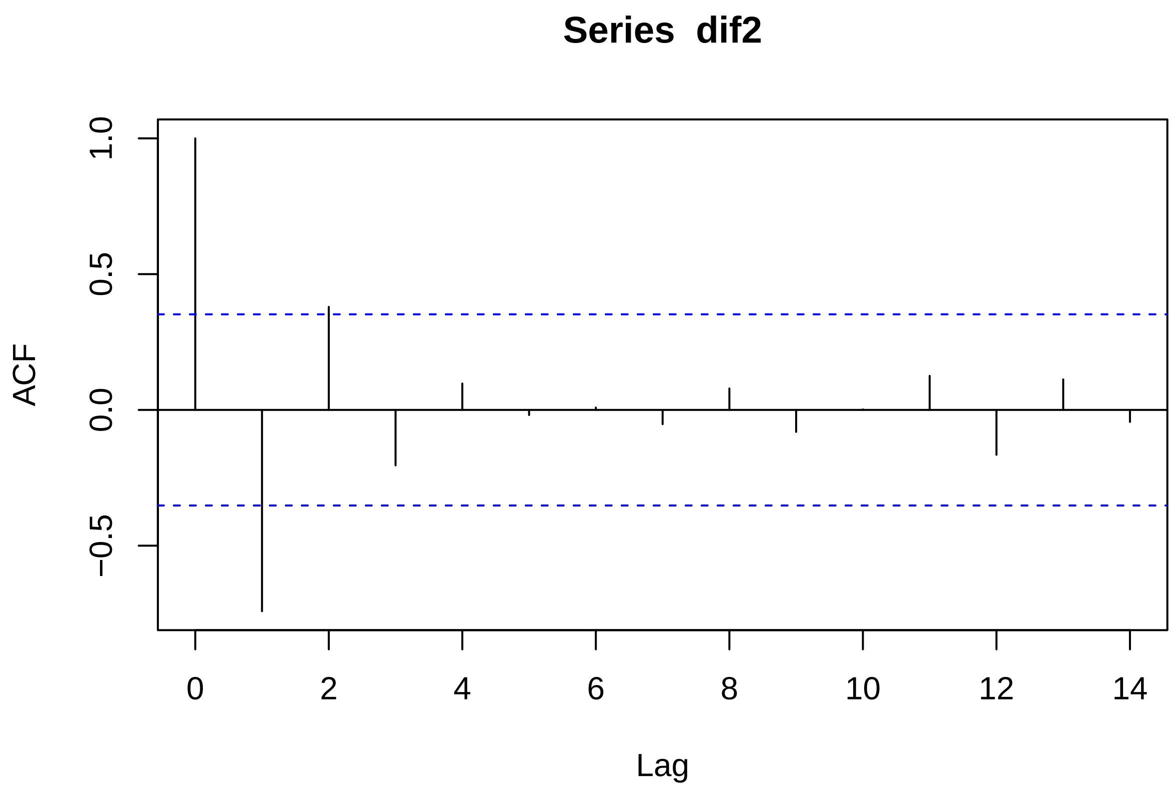
```
# p-valor = 0.1433
# existe evidencias a favor da hipotese nula
# que a serie nao e estacionaria

# segunda diferenca da serie
dif2 <- diff(dif1)
plot(dif2, type="b")
```



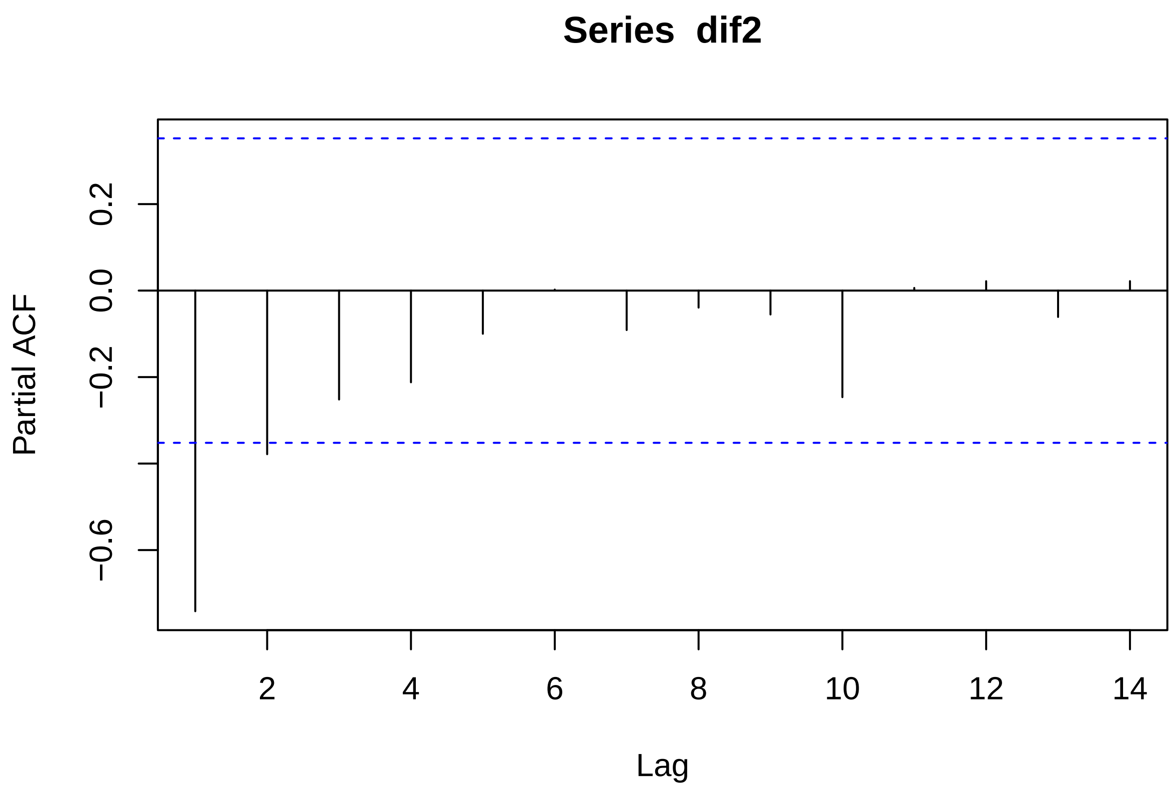

```r
adf.test(dif2)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  dif2
## Dickey-Fuller = -3.9935, Lag order = 3, p-value = 0.02234
## alternative hypothesis: stationary
```

```r
# p-valor = 0.02234
# existe evidencias a favor de rejeitar a hipotese nula
# indicando que a serie e estacionaria

# vamos identificar a ordem do modelo ARMA

acf(dif2)
```




```r
pacf(dif2)
```



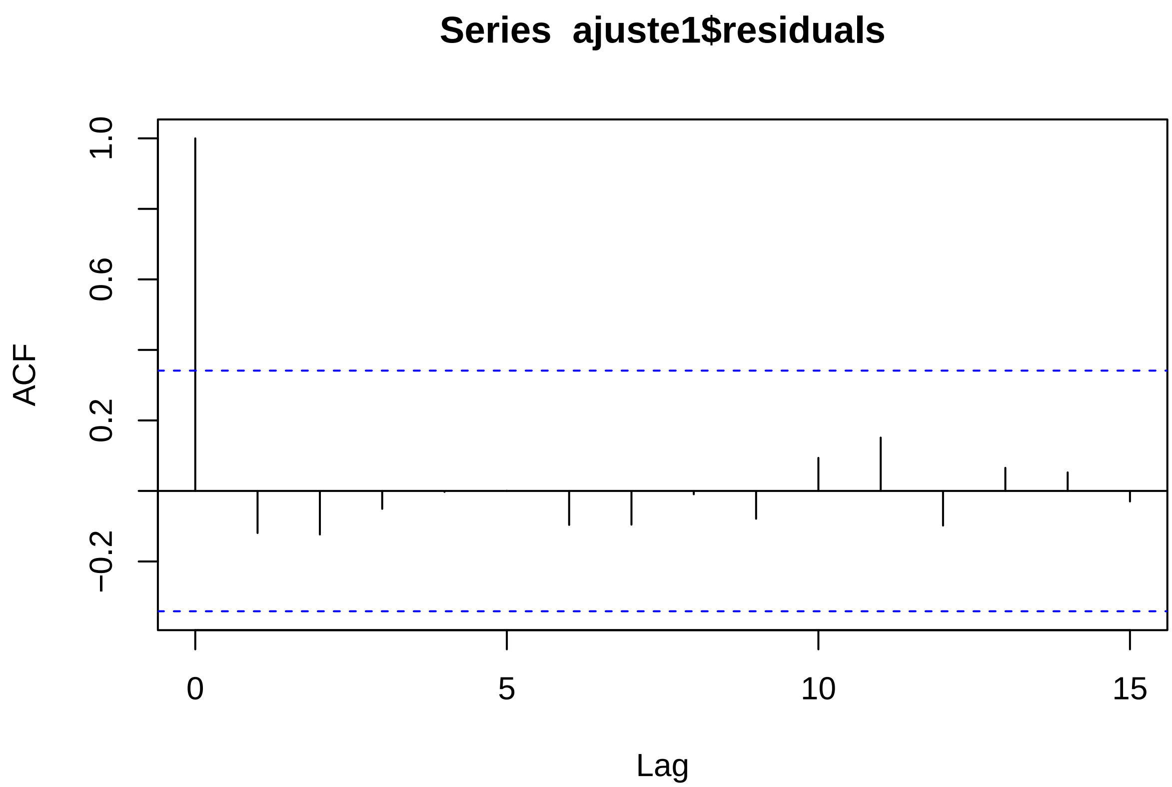

```r
# pode ser uma ARMA(1.1)

# vamos ajustar alguns modelos
ajuste1 <- Arima(serie_AIDS_ts, c(1, 2, 1))
ajuste1
```

```
## Series: serie_AIDS_ts
## ARIMA(1,2,1)
##
## Coefficients:
##           ar1      ma1
##       -0.4721  -0.7558
## s.e.   0.1648   0.1184
##
## sigma^2 = 7275979:  log likelihood = -288.71
## AIC=583.42   AICc=584.31   BIC=587.72
```

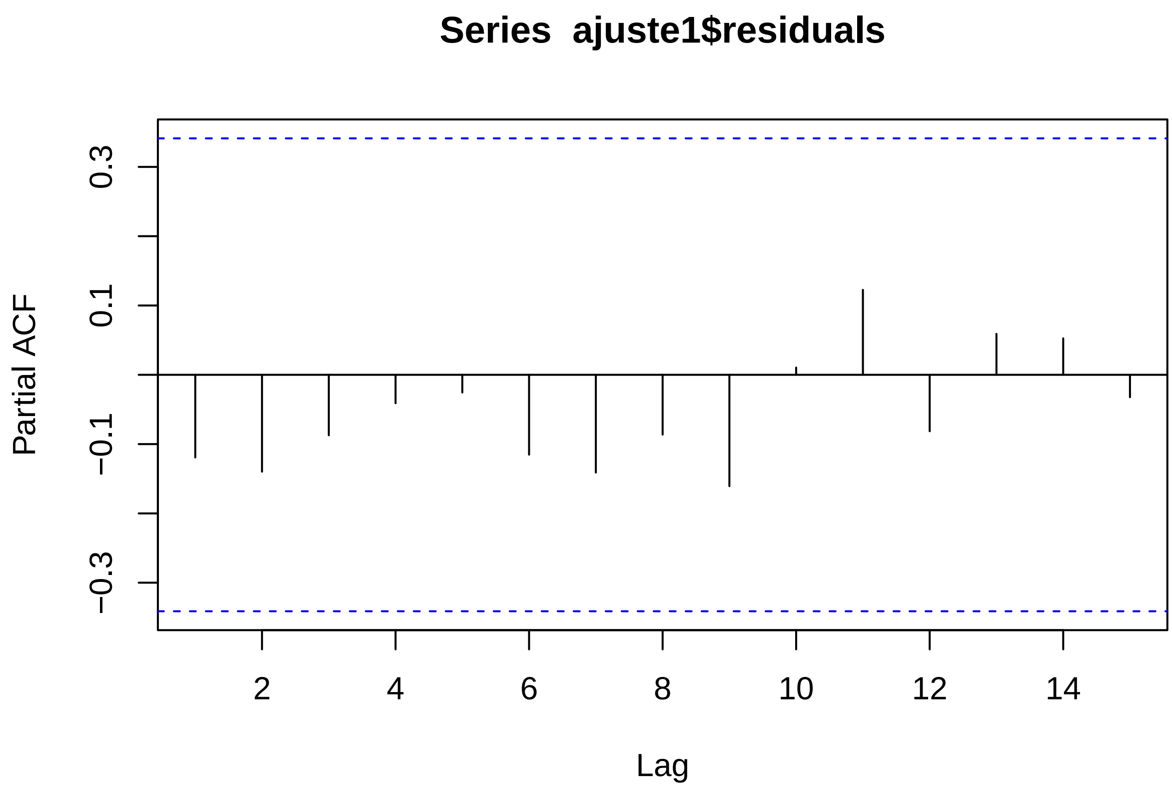
```r
confint(ajuste1)
```

```
##          2.5 %     97.5 %
## ar1 -0.7951462 -0.1490450
## ma1 -0.9878570 -0.5237943
```

```r
# diagnostico
acf(ajuste1$residuals)
```




```
pacf(ajuste1$residuals)
```



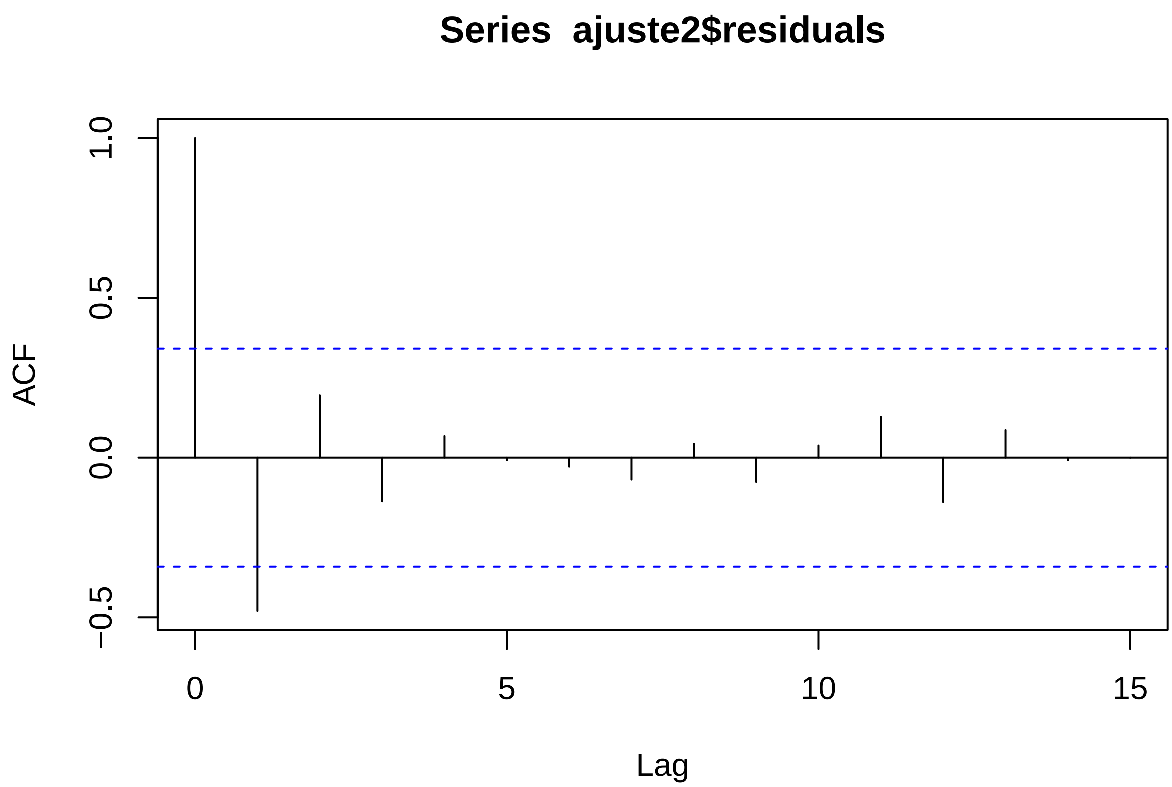

```
# vamos tentar alguns modelos mais parcimoniosos
ajuste2 <- Arima(serie_AIDS_ts, c(0, 2, 1))
ajuste2
```

```
## Series: serie_AIDS_ts
## ARIMA(0,2,1)
##
## Coefficients:
##           ma1
##       -0.8693
## s.e.   0.0793
##
## sigma^2 = 8774455:  log likelihood = -291.99
## AIC=587.97   AICc=588.4   BIC=590.84
```

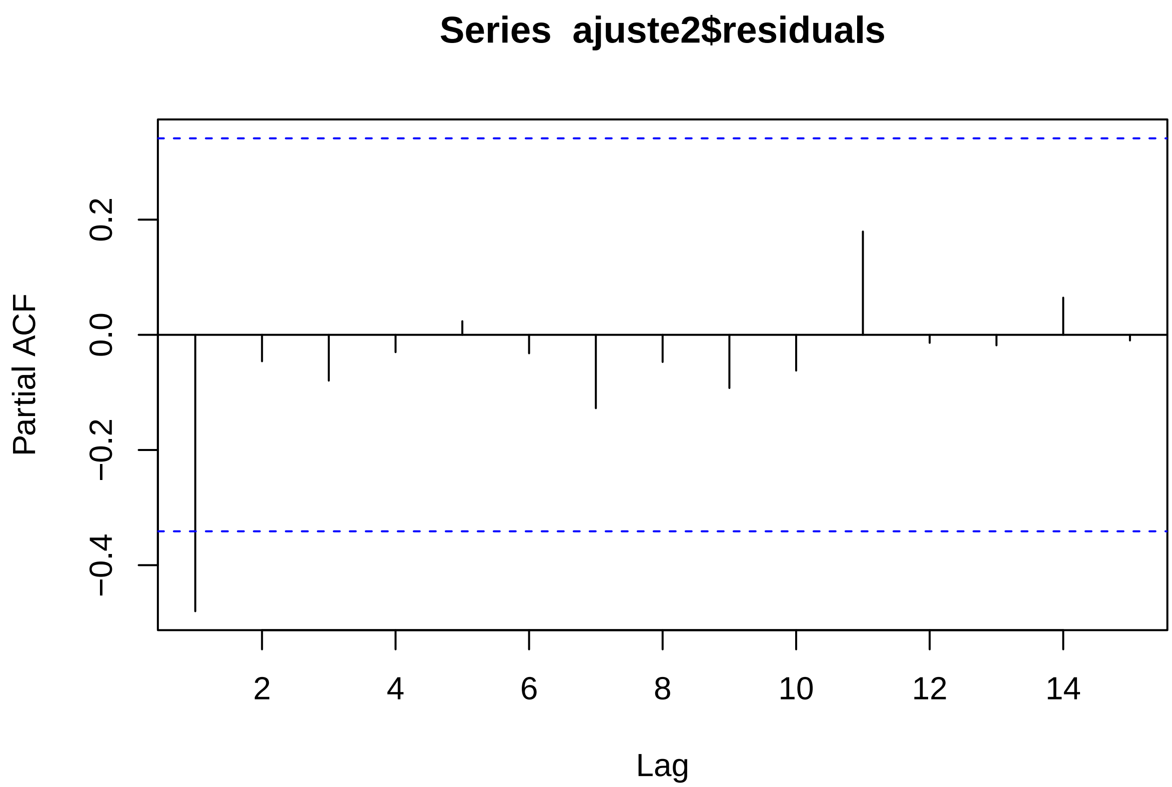
```
confint(ajuste2)
```

```
##         2.5 %     97.5 %
## ma1 -1.024801 -0.7137855
```

```
# diagnostico
acf(ajuste2$residuals)
```




```
pacf(ajuste2$residuals)
```



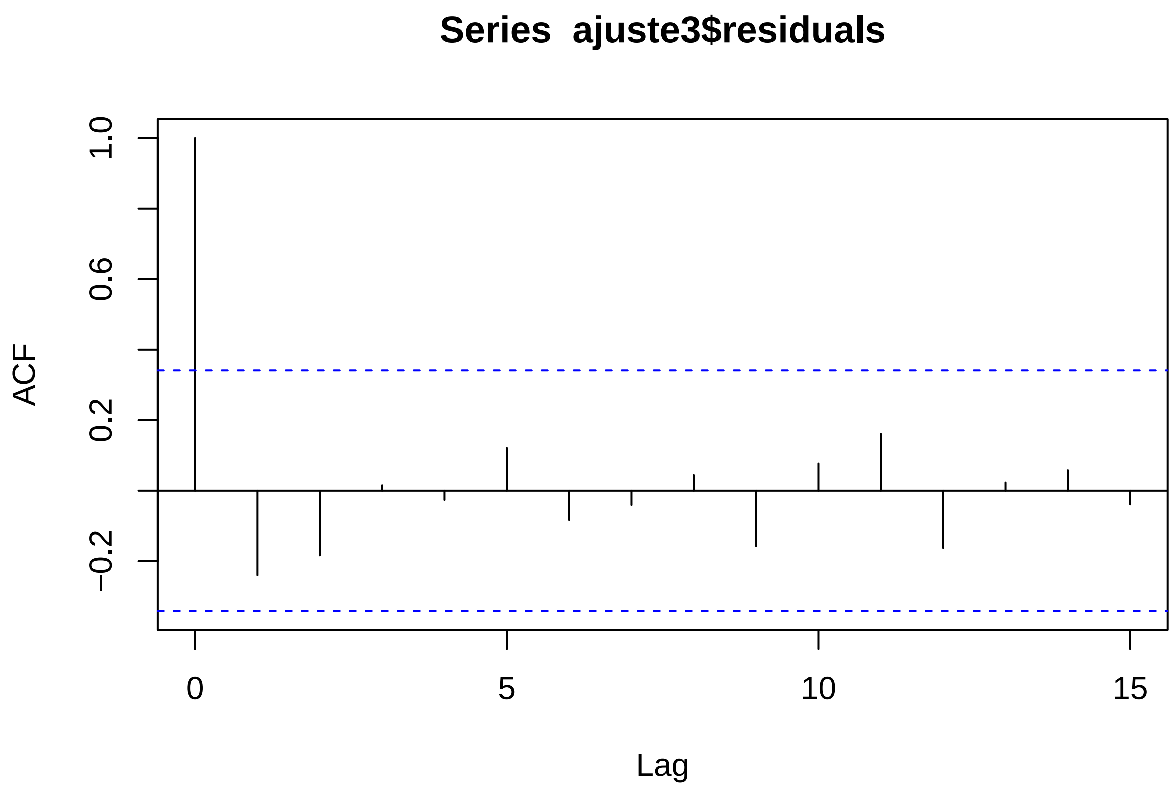

```r
# vamos ajustar mais um modelo
ajuste3 <- Arima(serie_AIDS_ts, c(1, 2, 0))
ajuste3
```

```
## Series: serie_AIDS_ts
## ARIMA(1,2,0)
##
## Coefficients:
##           ar1
##       -0.7332
## s.e.   0.1147
##
## sigma^2 = 9490567:  log likelihood = -292.88
## AIC=589.77   AICc=590.2   BIC=592.64
```

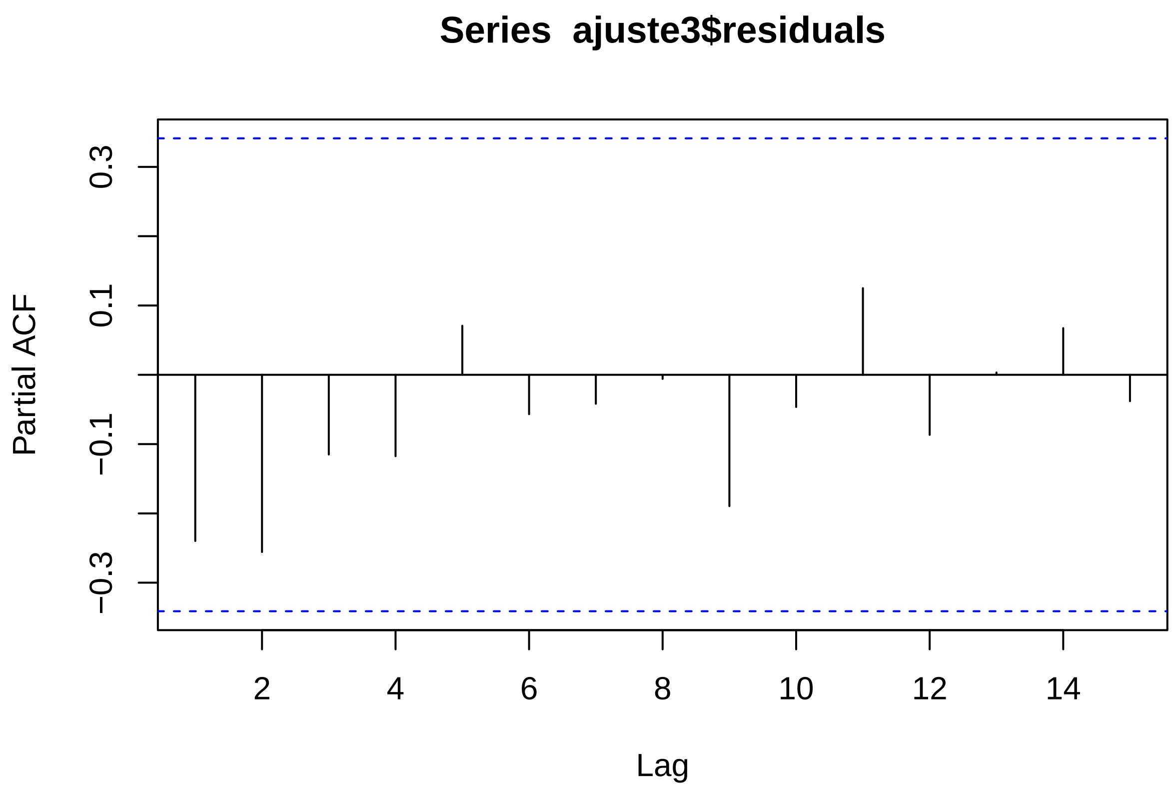
```r
confint(ajuste3)
```

```
##          2.5 %     97.5 %
## ar1 -0.957998 -0.5084292
```

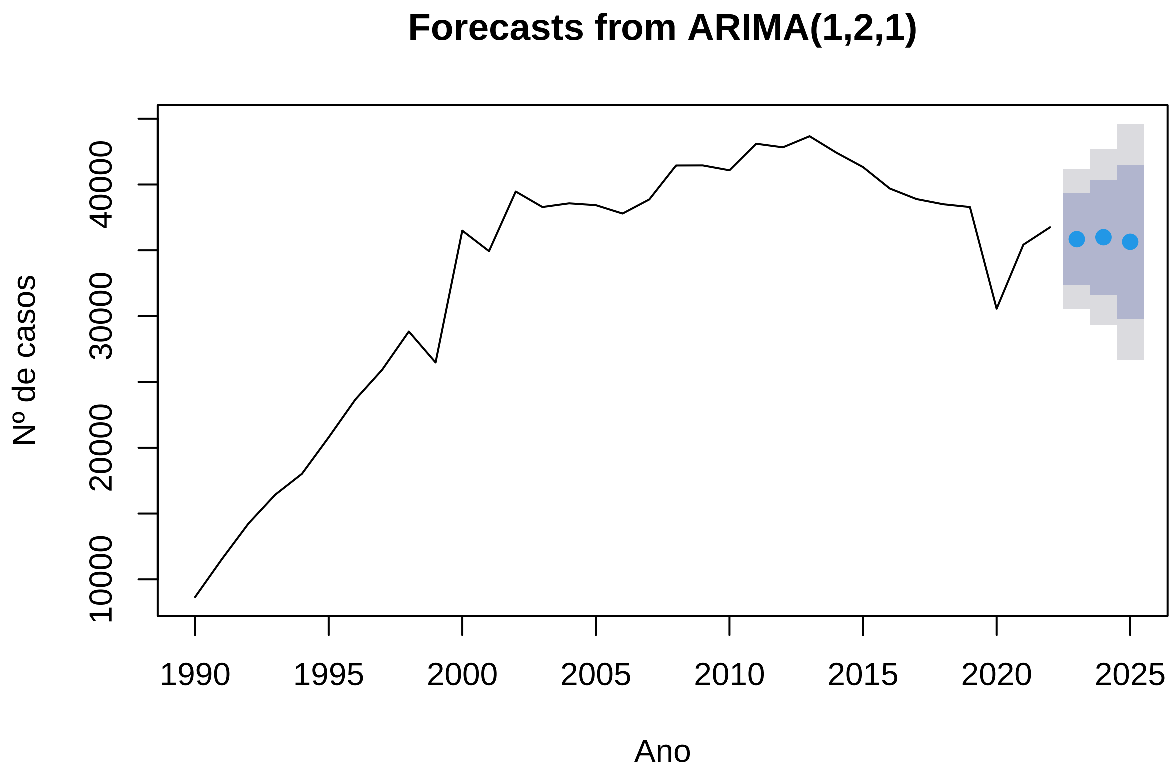
```r
# diagnostico
acf(ajuste3$residuals)
```




```
pacf(ajuste3$residuals)
```




```
# Pelo criterio AIC
# ARIMA(1,2,1) - AIC = 583.42
# ARIMA(0,2,1) - AIC = 587.97
# ARIMA(1,2,0) - AIC = 589.77
# O melhor foi o modelo ARIMA(1,2,1)

previsao <- forecast(ajuste1, h=3)
plot(previsao, ylab="Nº de casos", xlab="Ano")
```



Forecasts from ARIMA(1,2,1)
Nº de casos
10000
20000
30000
40000
1990
1995
2000
2005
2010
2015
2020
2025
Ano

Sazonalidade determinística

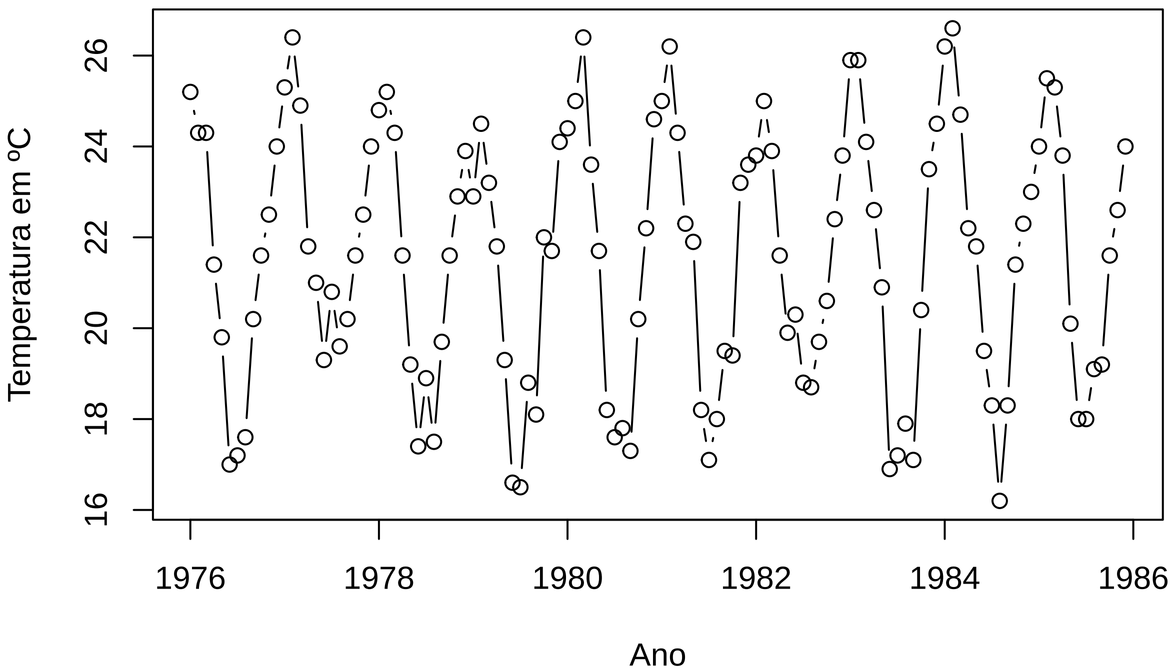
Exemplo de uma série temporal com sazonalidade

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).

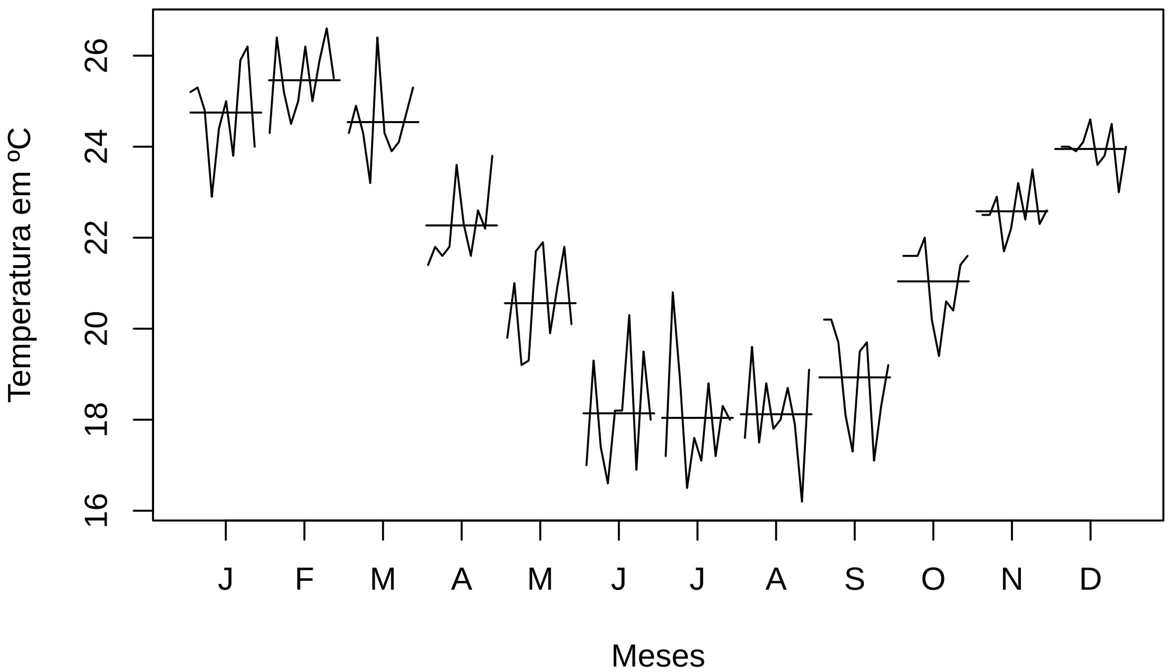
```
# fazendo a leitura do conjunto de dados
setwd("G:\\Meu Drive\\UFG\\Especializacao\\Aulas de series temporais\\Codigos")

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      20
```

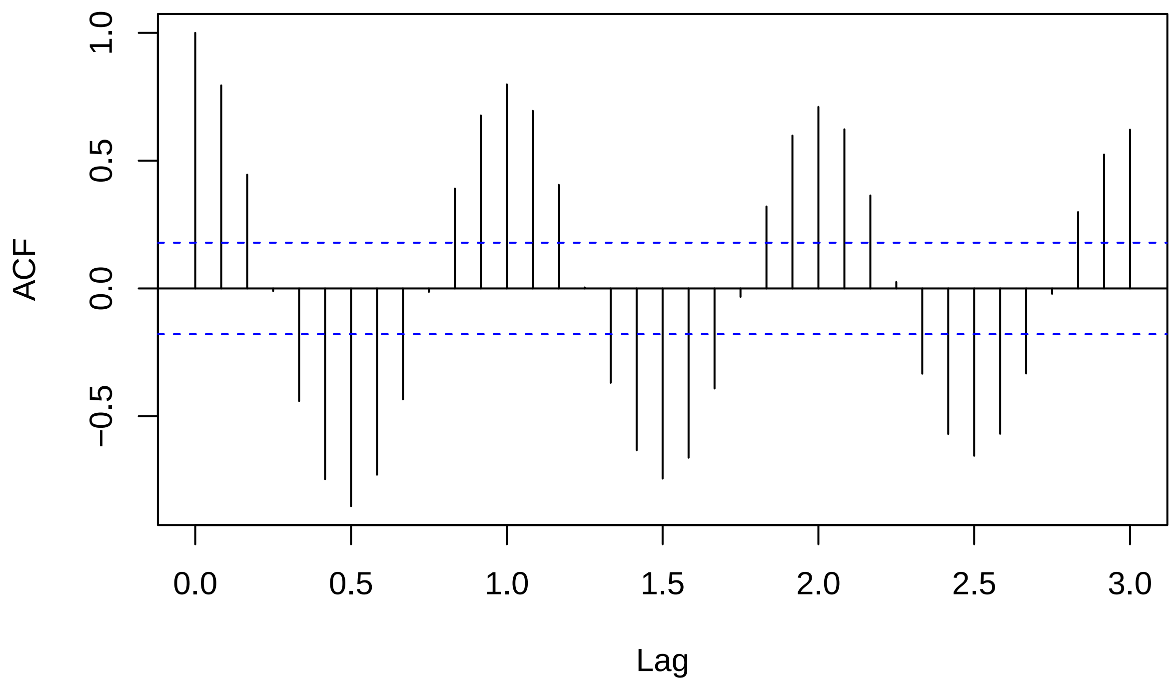
```
temp.cananeia_ts <- ts(temperatura$Cananeia, start = c(1976, 1), frequency = 12)

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



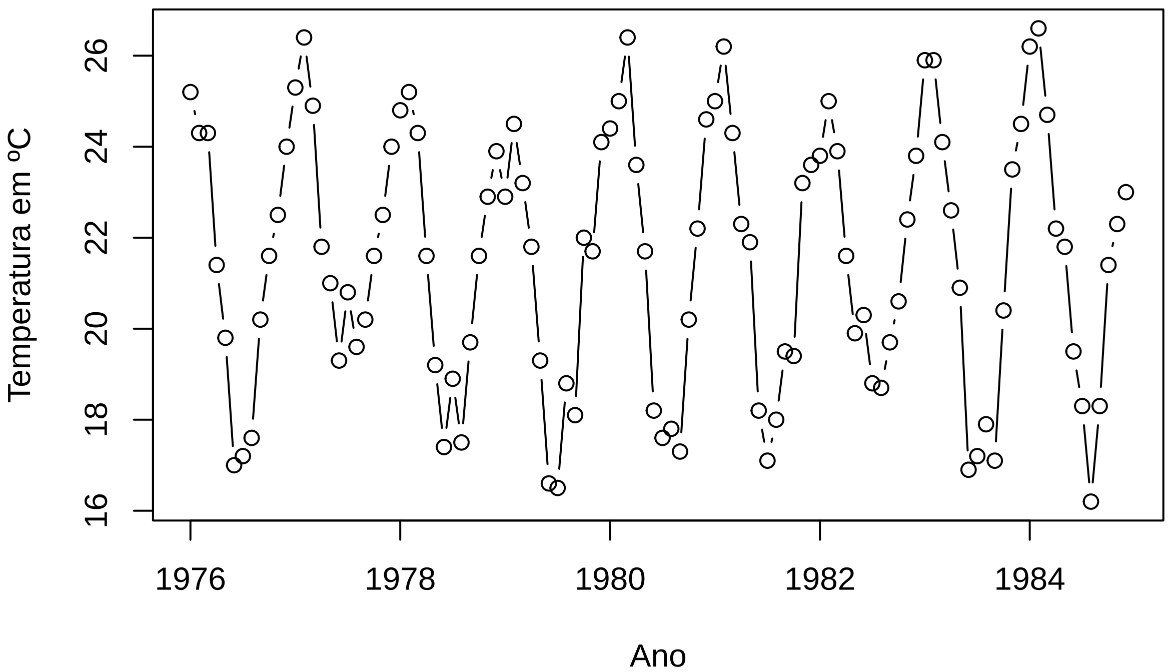

```
library(forecast)

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




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




```
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 ºC", xlab="Ano")
```




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

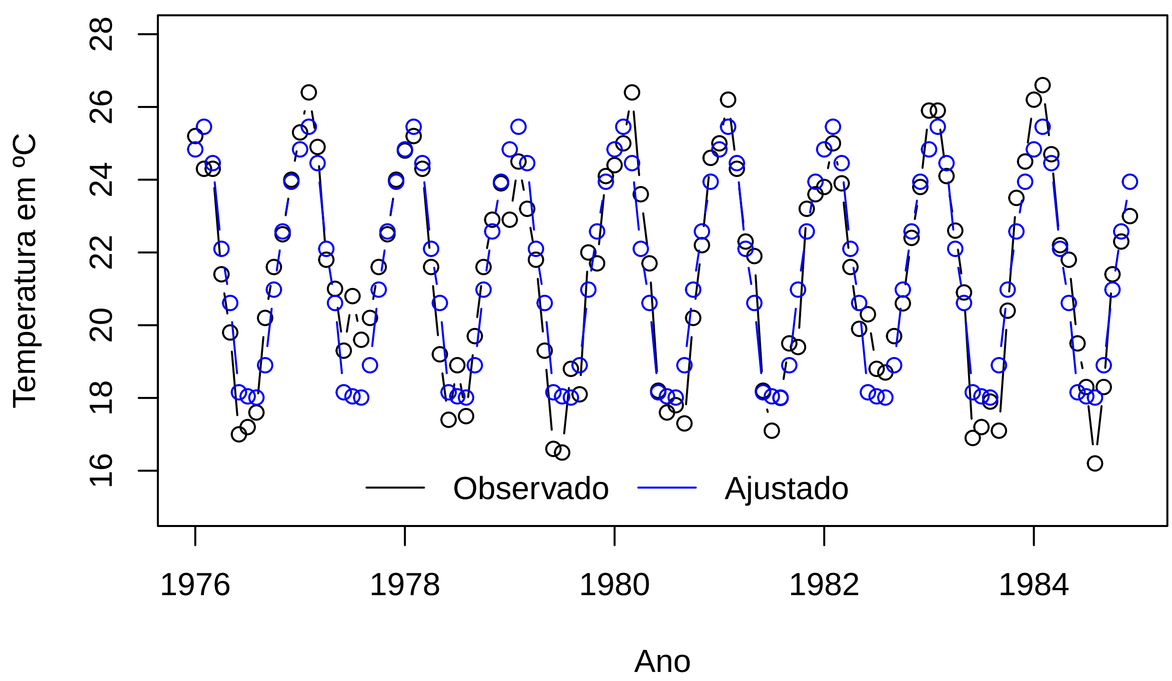
```
##
## Call:
## tslm(formula = trein_temp.cananeia_ts ~ season)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.93333 -0.58333 -0.06111  0.62222  2.75556
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  24.8333     0.3229  76.901  < 2e-16 ***
## 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
```

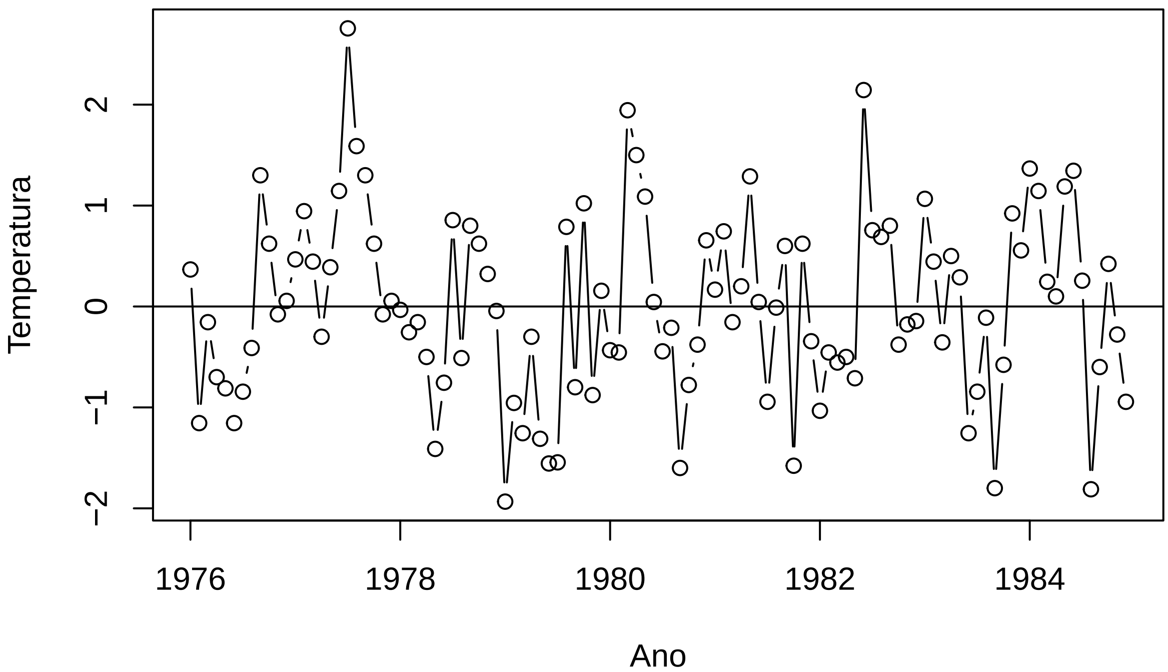
```
AIC(ajuste1)
```

```
## [1] 312.9199
```

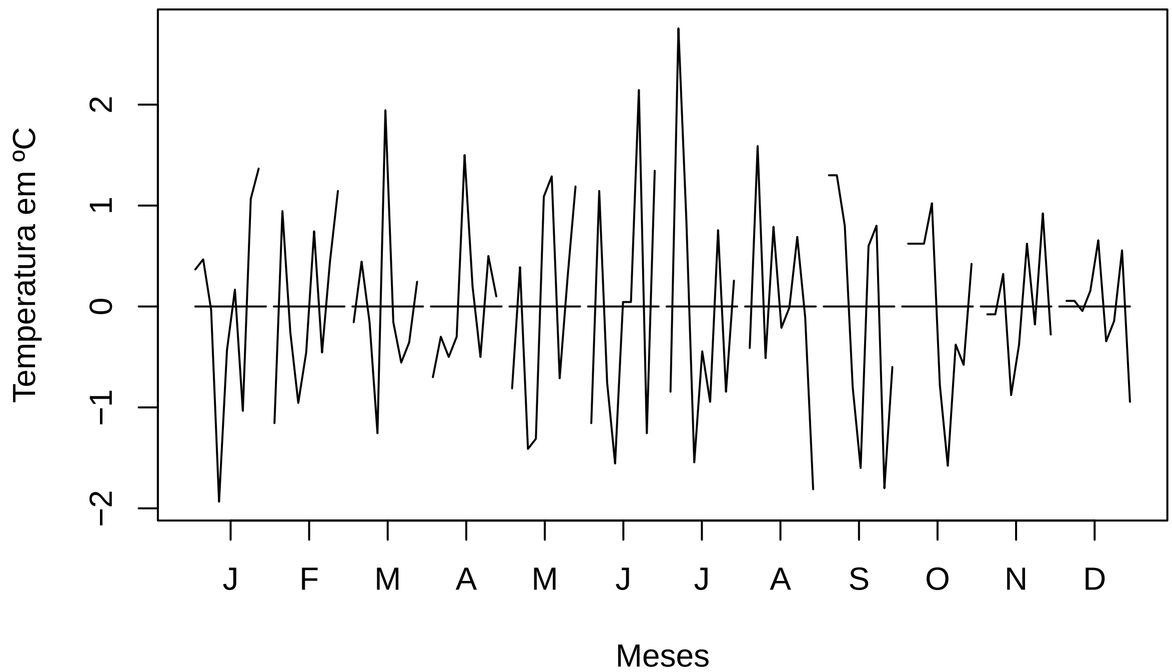
```
# grafico da serie temporal
plot(trein_temp.cananeia_ts, type="b", ylab="Temperatura em ºC", xlab="Ano", ylim=c(15, 28))
# valores ajustados
lines(ajuste1$fitted.values, type="b", col="blue")
legend("bottomright", lty=c(1,1), col=c("black", "blue"), c("Observado", "Ajustado"), bty = "n", ncol=3
```




```
# serie livre de sazonalidade
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)
```



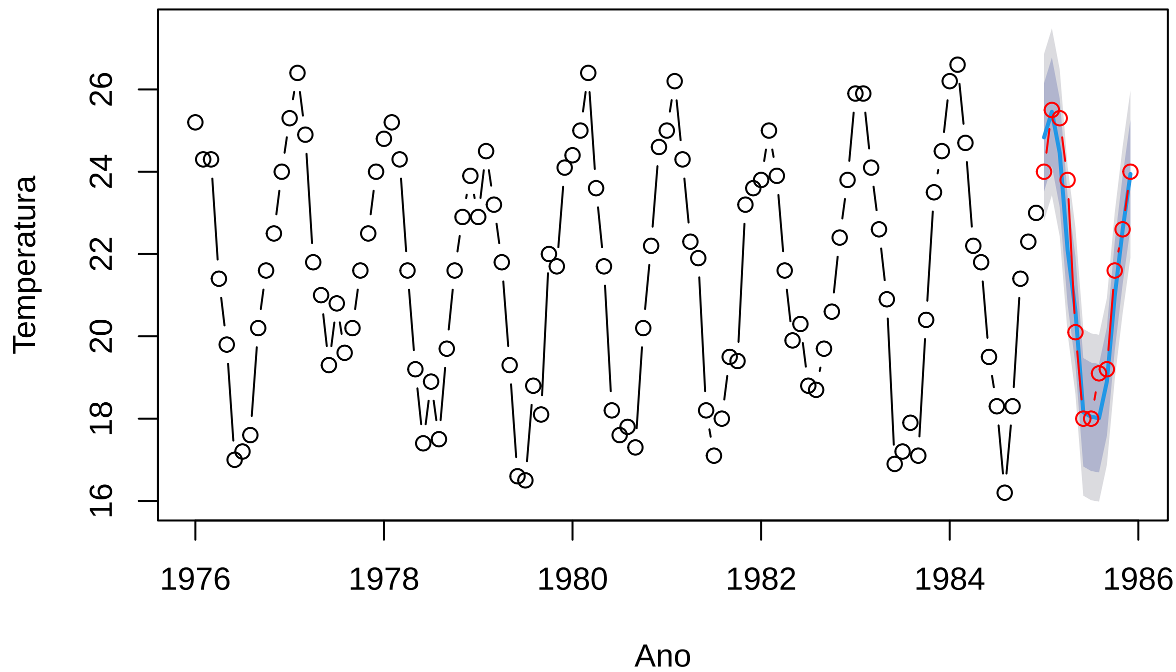

```r
# grafico dos meses separadamente
monthplot(cananeia_ss_ts,  xlab="Meses", ylab="Temperatura em ºC", main="")
```




```r
# previsao
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
```

```r
# grafico com os valores observados e preditos

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

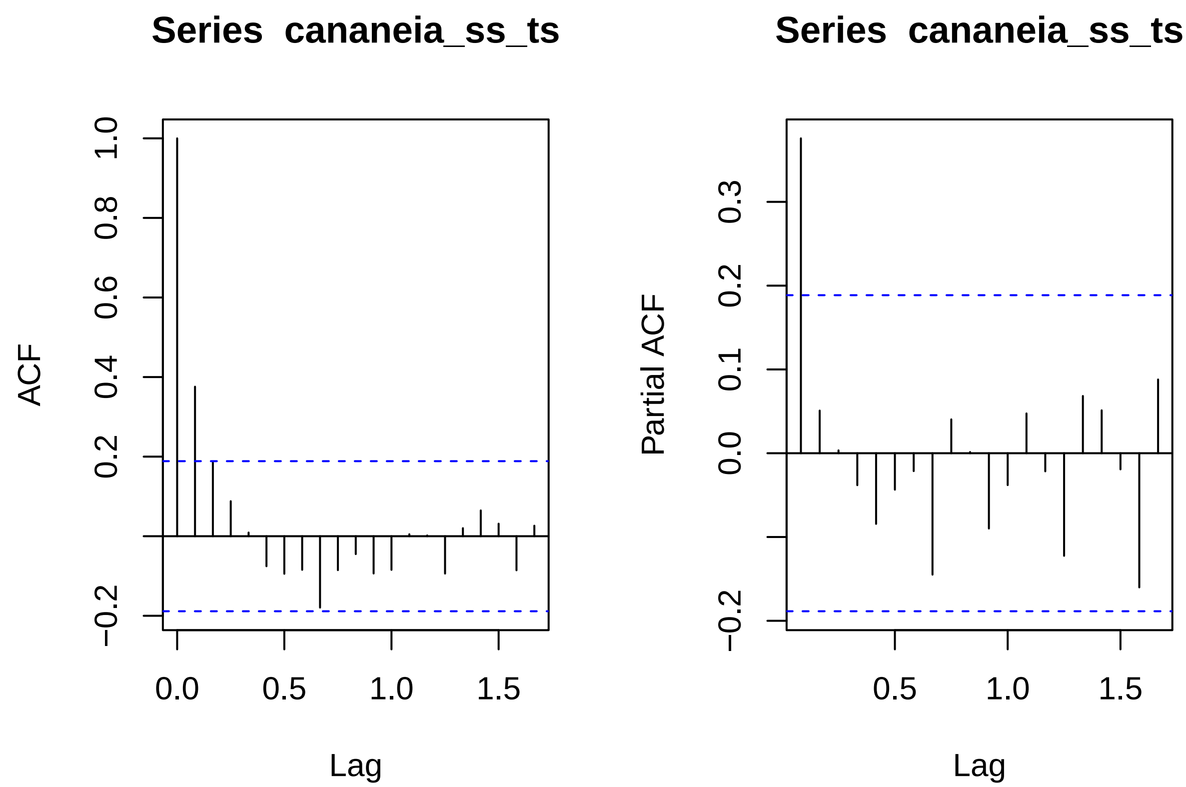
```
cat("MAE:", MAE, "\n")
```

```
## MAE: 0.5185185
```

```
cat("MAPE:", MAPE, "%\n")
```

```
## MAPE: 2.354584 %
```

```
# vamos agora realizar uma analise para verificar
# se os residuos sao ruido branco

par(mfrow=c(1,2))
acf(cananeia_ss_ts)
pacf(cananeia_ss_ts)
```

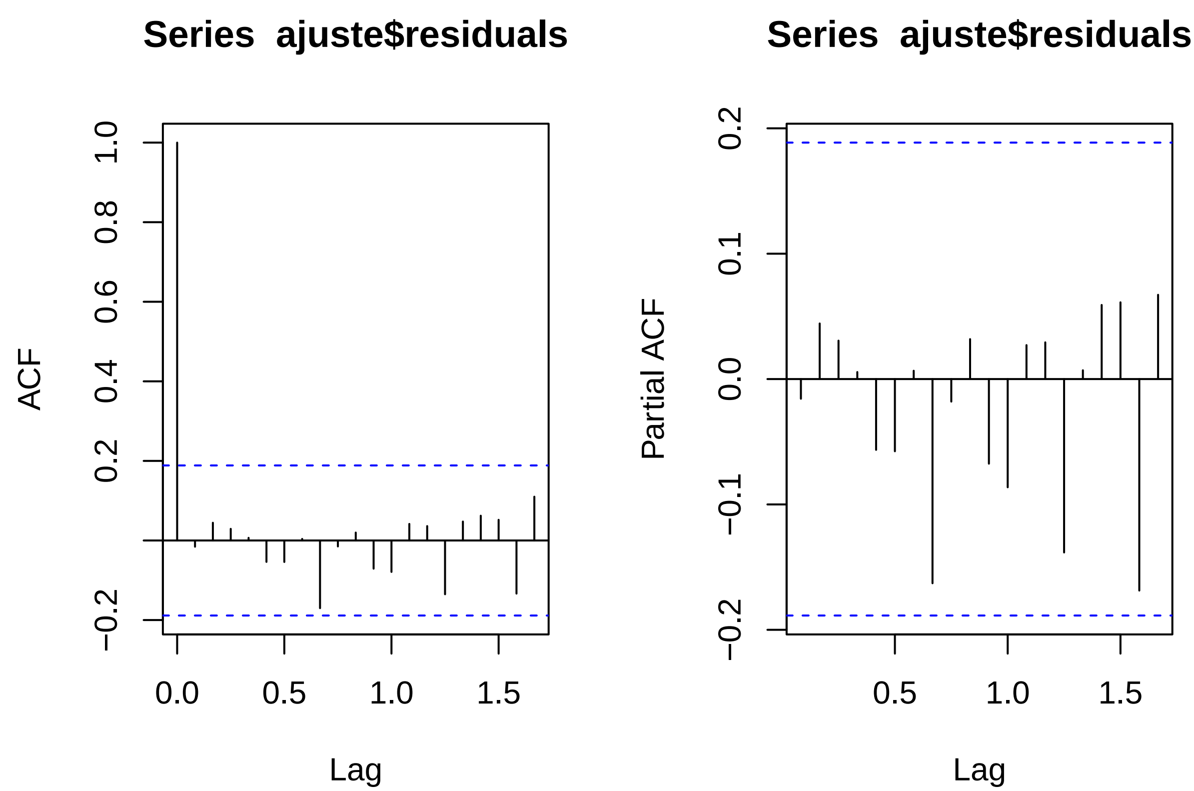



```r
# note que tanto na acf, quanto na pacf
# obtemos lags significativos
# neste caso devemos entao utilizar o modelo
# Z_t = \mu_t + N_t,
# sendo que mu_t ja foi estimado pela regressao
# para o ajuste de N_t utilizamos a serie sem sazonalidade
# pelos graficos da acf e pacf uma indicacao de modelo e o AR(1)

# ajuste automatico
ajuste <- auto.arima(cananeia_ss_ts)
summary(ajuste)
```

```
## Series: cananeia_ss_ts
## ARIMA(1,0,0) with zero mean
##
## Coefficients:
##          ar1
##       0.3766
## s.e.  0.0890
##
## sigma^2 = 0.7218:  log likelihood = -135.21
## AIC=274.43   AICc=274.54   BIC=279.79
##
## Training set error measures:
##                           ME      RMSE       MAE      MPE     MAPE      MASE
## Training set -0.003543479 0.8456269 0.6478436 37.88551 141.5954 0.5957182
```

```
##                      ACF1
## Training set -0.01569051
```

```
# vamos verificar a qualidade do ajuste

par(mfrow=c(1,2))
acf(ajuste$residuals)
pacf(ajuste$residuals)
```




```
# notamos que o modelo AR(1) se ajustou bem a serie


# agora vamos reajustar o modelo considerando a parte AR + regressao
# o erro padrao estimado na regressao pode estar bem estimado
# pois os erros do modelo nao sao independentes
# para isso precisamos utilizar a matriz do modelo de regressao

matriz_1 <- seasonaldummy(trein_temp.cananeia_ts) # matriz da parte sazonal
dim(matriz_1)
```

```
## [1] 108  11
```

```
#matriz_1[1:24,]

matriz_2 <- rbind(rep(0, 11), matriz_1[1:107,])
#matriz_2[1:24,]

names <- c("Int", "Fev", "Mar", "Abr", "Mai", "Jun", "Jul",
```

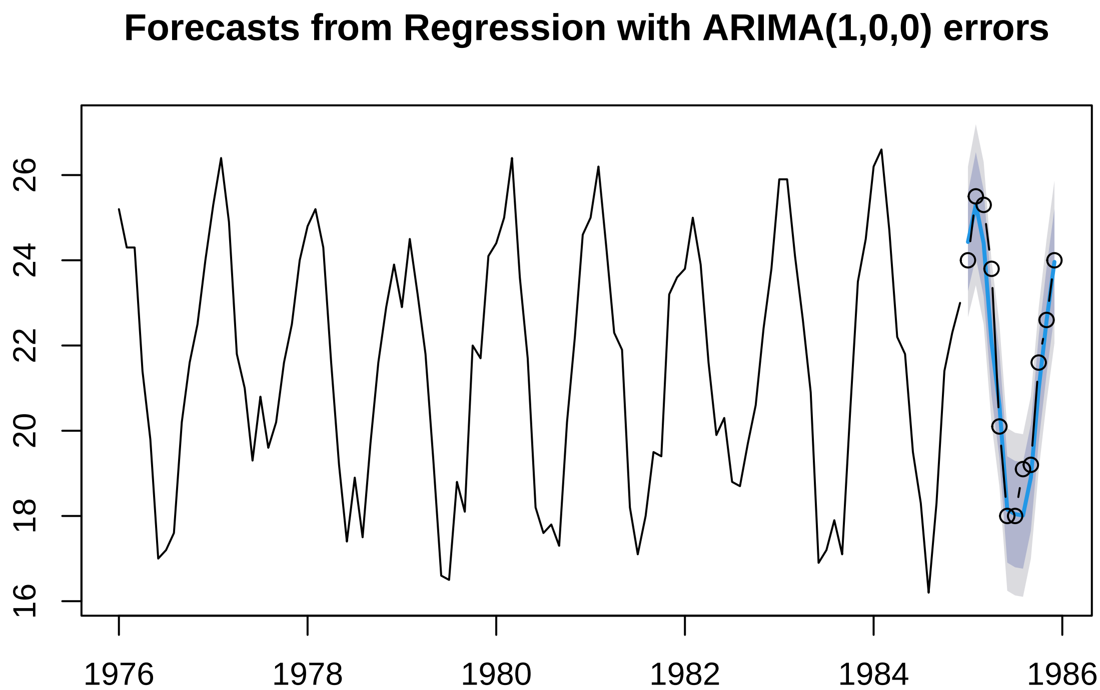
```r
           "Ago", "Set", "Out", "Nov", "Dez")

matriz_modelo <- cbind(rep(1, 108), matriz_2)
colnames(matriz_modelo) <-  names

#matriz_modelo[1:24, ]

# ajuste do AR(1) + regressao
ajuste_final <- Arima(trein_temp.cananeia_ts, c(1, 0, 0), xreg = matriz_modelo, include.mean = F)
ajuste_final
```

```
## Series: trein_temp.cananeia_ts
## Regression with ARIMA(1,0,0) errors
##
## Coefficients:
##          ar1      Int     Fev      Mar      Abr      Mai      Jun      Jul
##       0.3773  24.7930  0.6473  -0.3432  -2.6952  -4.1827  -6.6377  -6.7486
## s.e.  0.0891   0.3023  0.3389   0.3974   0.4173   0.4245   0.4271   0.4278
##          Ago      Set      Out      Nov      Dez
##       -6.7816  -5.8921  -3.8128  -2.2088  -0.8315
## s.e.   0.4273   0.4250   0.4185   0.4005   0.3473
##
## sigma^2 = 0.8127:  log likelihood = -135.2
## AIC=298.4   AICc=302.91   BIC=335.95
```

```r
# so regressao - AIC = 312.9199
# AR(1) + regressao - AIC = 298.4

# previsao
previsao2 <- forecast(ajuste_final, xreg = matriz_modelo[1:12, ], h=12)
previsao2
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## Jan 1985       24.43023 23.27490 25.58556 22.66331 26.19715
## Feb 1985       25.30346 24.06863 26.53829 23.41495 27.19197
## Mar 1985       24.39817 23.15243 25.64391 22.49298 26.30336
## Apr 1985       22.07835 20.83107 23.32563 20.17079 23.98590
## May 1985       20.60295 19.35545 21.85045 18.69506 22.51084
## Jun 1985       18.15251 16.90497 19.40004 16.24457 20.06045
## Jul 1985       18.04339 16.79585 19.29093 16.13545 19.95134
## Aug 1985       18.01100 16.76346 19.25854 16.10306 19.91895
## Sep 1985       18.90074 17.65320 20.14828 16.99279 20.80869
## Oct 1985       20.98014 19.73260 22.22768 19.07219 22.88809
## Nov 1985       22.58419 21.33665 23.83173 20.67624 24.49214
## Dec 1985       23.96150 22.71396 25.20904 22.05355 25.86945
```

```r
par(mfrow=c(1,1))
plot(previsao2)
lines(cananeia_novos_ts, type="b")
```




```r
# calculo do erro

# Raiz do Erro Quadratico Medio (REQM / RMSE)
RMSE2 <- sqrt(mean((cananeia_novos_ts - previsao2$mean)^2))

# Erro Absoluto Medio (EAM / MAE)
MAE2 <- mean(abs(cananeia_novos_ts - previsao2$mean))

# Erro Percentual Absoluto Medio (MAPE)
MAPE2 <- mean(abs(cananeia_novos_ts - previsao2$mean) / abs(cananeia_novos_ts)) * 100

# ajuste so com regressao
cat("RMSE:", RMSE, "\n")
```

```
## RMSE: 0.7218803
```

```r
cat("MAE:", MAE, "\n")
```

```
## MAE: 0.5185185
```

```r
cat("MAPE:", MAPE, "%\n")
```

```
## MAPE: 2.354584 %
```

```r
# ajuste com AR(1) + regressao
cat("RMSE:", RMSE2, "\n")
```

```
## RMSE: 0.7036003
```

```r
cat("MAE:", MAE2, "\n")
```

```
## MAE: 0.5009611
```

```r
cat("MAPE:", MAPE2, "%\n")
```

```
## MAPE: 2.276053 %
```

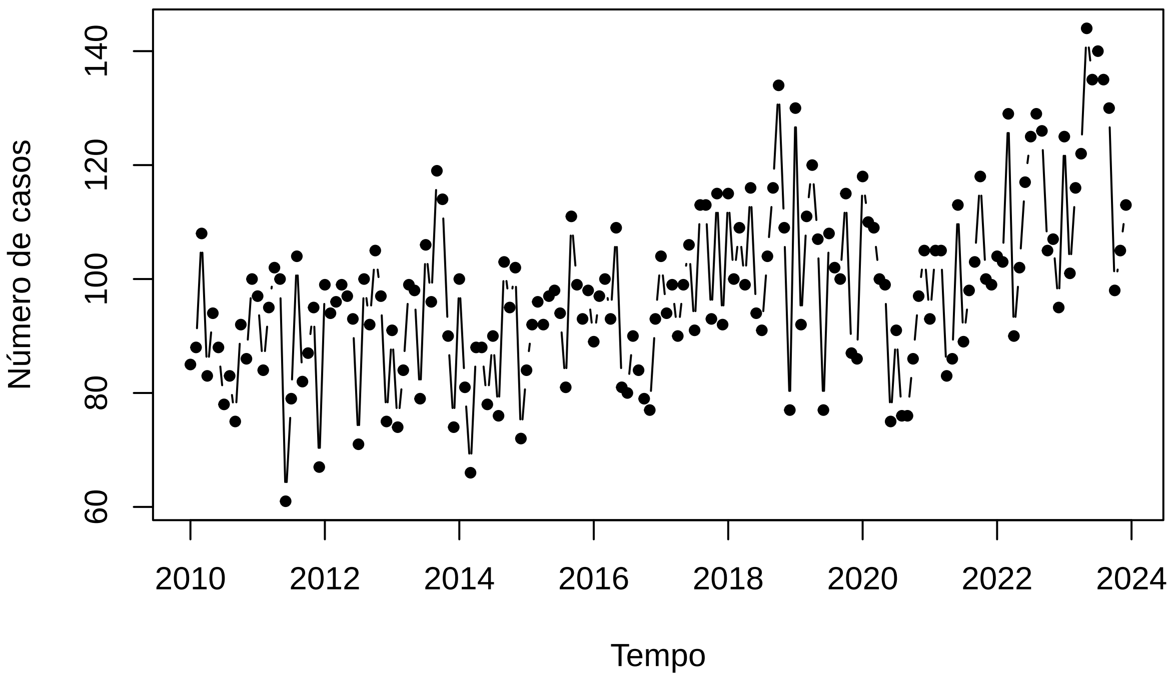
```r
# previsoes melhores!!!
```

Sazonalidade estocástica

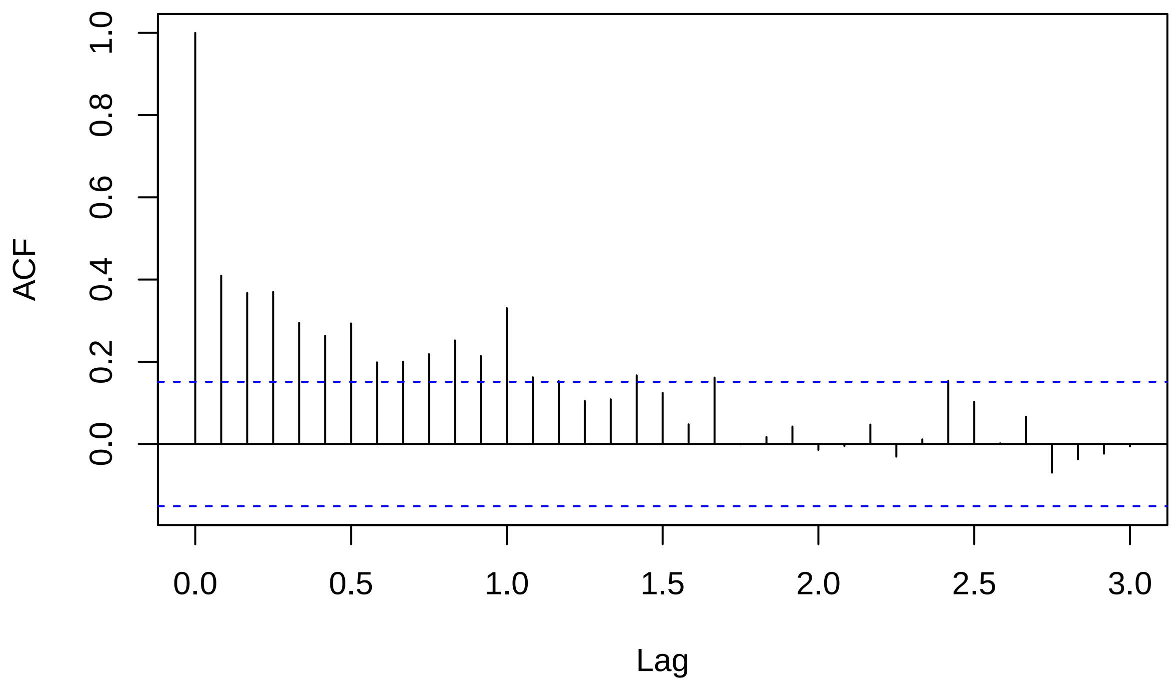
Análise da série temporal de tuberculose

Considerando a série temporal do número de casos de tuberculose no Estado de Goiás, com o número de casos mensal de janeiro de 2010 a dezembro de 2023, vamos obter o gráfico das funções FAC e FACP, na sequência ajustar alguns modelos da família ARMA.

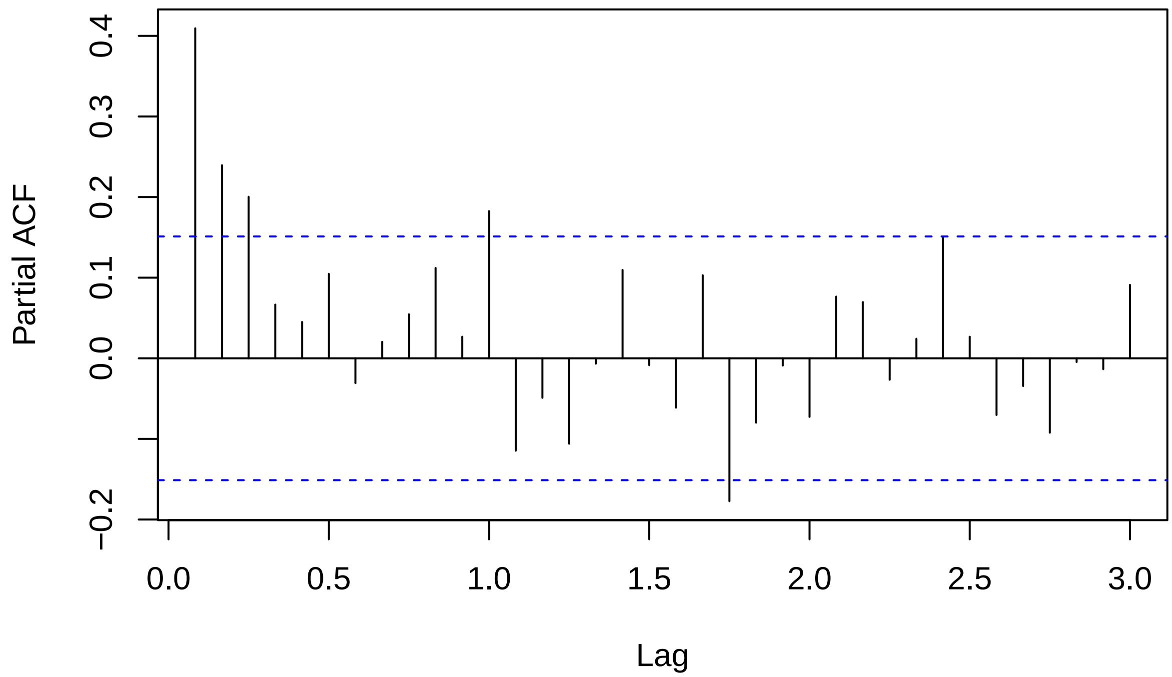
```
# serie temporal do numero de casos de tuberculose em Goias

serie_tuberculose <- c(85,88,108,83,94,88,78,83,75,92,86,100,
                       97,84,95,102,100,61,79,104,82,87,95,67,
                       99,94,96,99,97,93,71,100,92,105,97,75,
                       91,74,84,99,98,79,106,96,119,114,90,74,
                       100,81,66,88,88,78,90,76,103,95,102,72,
                       84,92,96,92,97,98,94,81,111,99,93,98,
                       89,97,100,93,109,81,80,90,84,79,77,93,
                       104,94,99,90,99,106,91,113,113,93,115,92,
                       115,100,109,99,116,94,91,104,116,134,109,77,
                       130,92,111,120,107,77,108,102,100,115,87,86,
                       118,110,109,100,99,75,91,76,76,86,97,105,
                       93,105,105,83,86,113,89,98,103,118,100,99,
                       104,103,129,90,102,117,125,129,126,105,107,95,
                       125,101,116,122,144,135,140,135,130,98,105,113)

tuberculose_ts <- ts(serie_tuberculose, start= c(2010, 1), frequency = 12)

plot(tuberculose_ts, type="b", pch=20 , ylab="Número de casos", xlab="Tempo")
```




```
#par(mfrow=c(1,2))
# acf da serie
acf(tuberculose_ts, main="", lag.max=36)
```



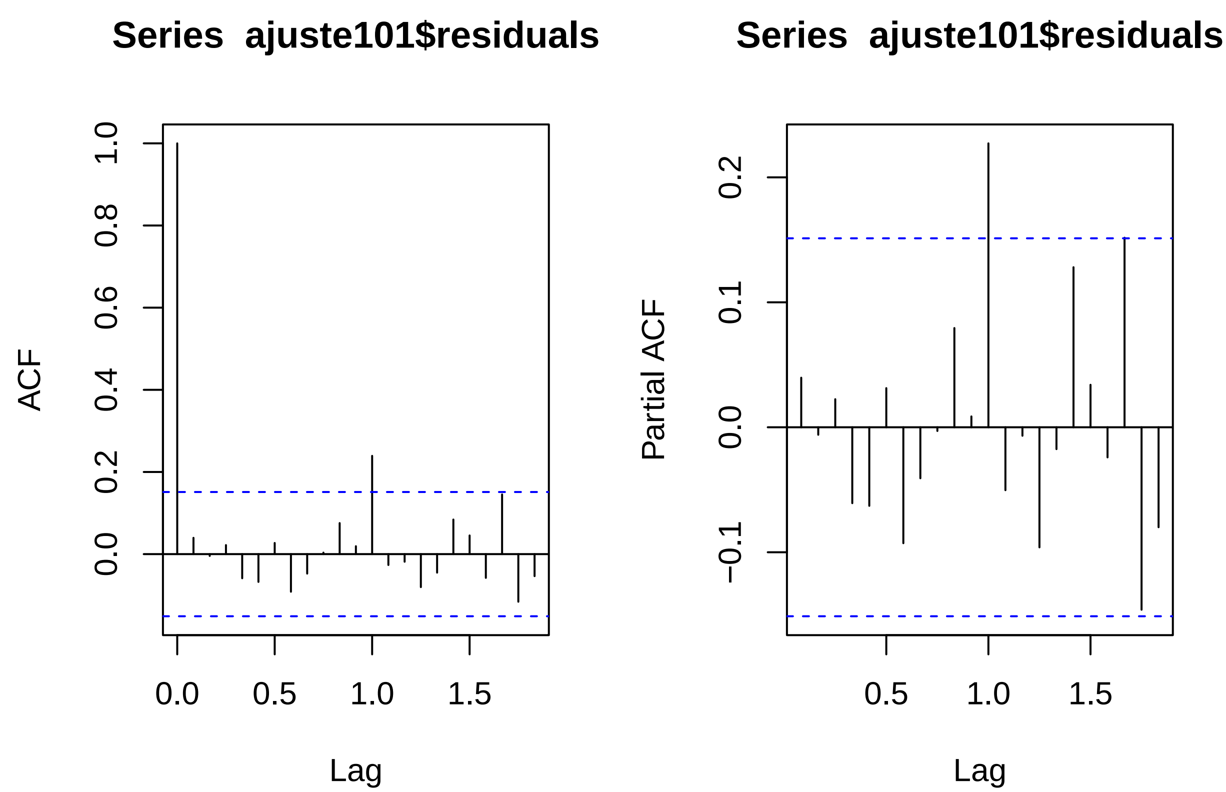

```
# pacf da serie
pacf(tuberculose_ts, main="", lag.max=36)
```




```r
# "melhor" modelo encontrado
ajuste101 <- Arima(tuberculose_ts, c(1, 0, 1))
ajuste101
```

```
## Series: tuberculose_ts
## ARIMA(1,0,1) with non-zero mean
##
## Coefficients:
##          ar1      ma1     mean
##       0.9652  -0.7888  99.1243
## s.e.  0.0362   0.0833   5.5643
##
## sigma^2 = 174.4:  log likelihood = -670.84
## AIC=1349.67   AICc=1349.92   BIC=1362.17
```

```r
# vamos analisar a qualidade do ajuste
par(mfrow=c(1,2))
# acf da serie
acf(ajuste101$residuals)
pacf(ajuste101$residuals)
```



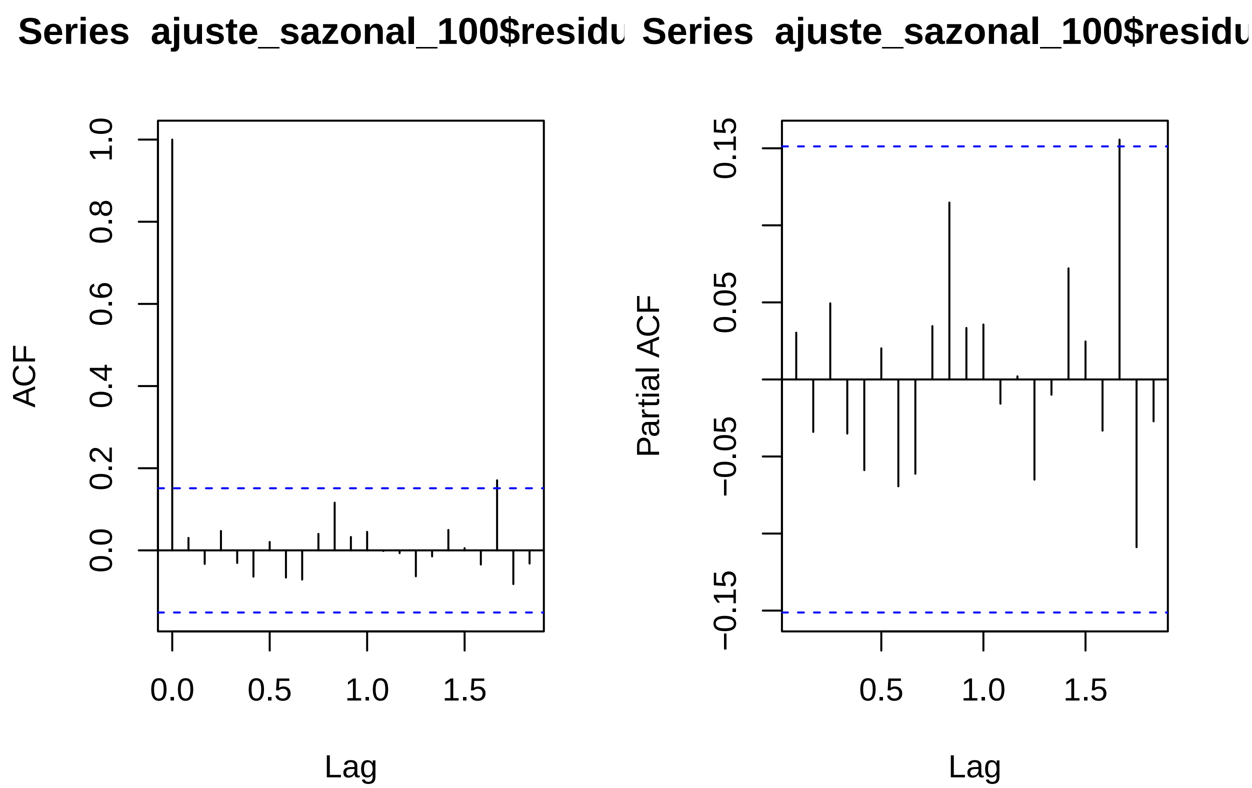

```r
# ja vimos que o modelo inconpleto tambem nao resolveu
# o problema da correlacao no lag 12
# vamos agora proceder o ajuste usando o modelo sazonal

# SARIMA(1,0,1)(1,0,0)
ajuste_sazonal_100 <- Arima(tuberculose_ts, c(1,0,1), seasonal = c(1, 0, 0))
ajuste_sazonal_100
```

```
## Series: tuberculose_ts
## ARIMA(1,0,1)(1,0,0)[12] with non-zero mean
##
## Coefficients:
##          ar1      ma1    sar1     mean
##       0.9393  -0.7513  0.2732  98.9759
## s.e.  0.0530   0.1029  0.0793   5.0696
##
## sigma^2 = 163.1:  log likelihood = -665.14
## AIC=1340.28   AICc=1340.65   BIC=1355.9
```

```r
confint(ajuste_sazonal_100)
```

```
##                2.5 %      97.5 %
## ar1        0.8354667   1.0430803
## ma1       -0.9529352  -0.5496713
## sar1       0.1177420   0.4286107
## intercept 89.0396074 108.9121069
```

```r
# diagnostico
acf(ajuste_sazonal_100$residuals)
pacf(ajuste_sazonal_100$residuals)
```



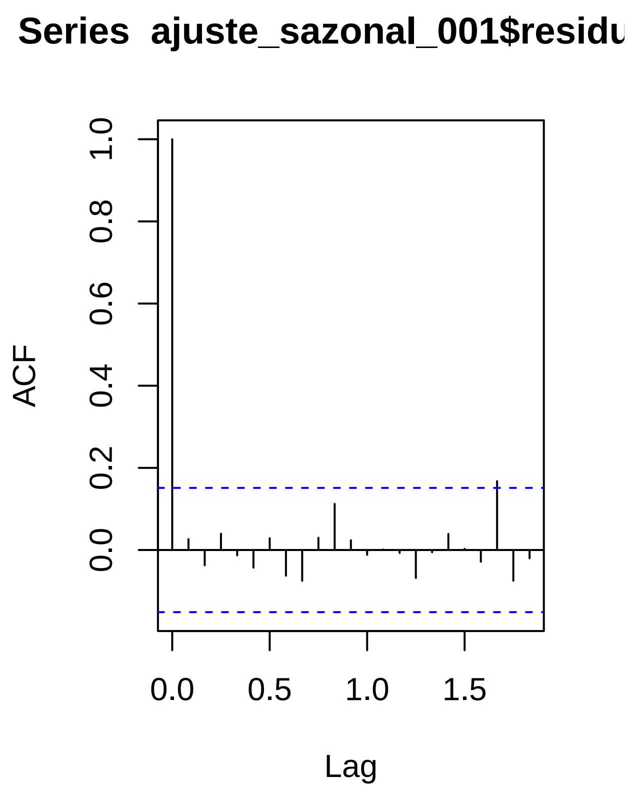

```r
# SARIMA(1,0,1)(0,0,1)
ajuste_sazonal_001 <- Arima(tuberculose_ts, c(1,0,1), seasonal = c(0, 0, 1))
ajuste_sazonal_001
```

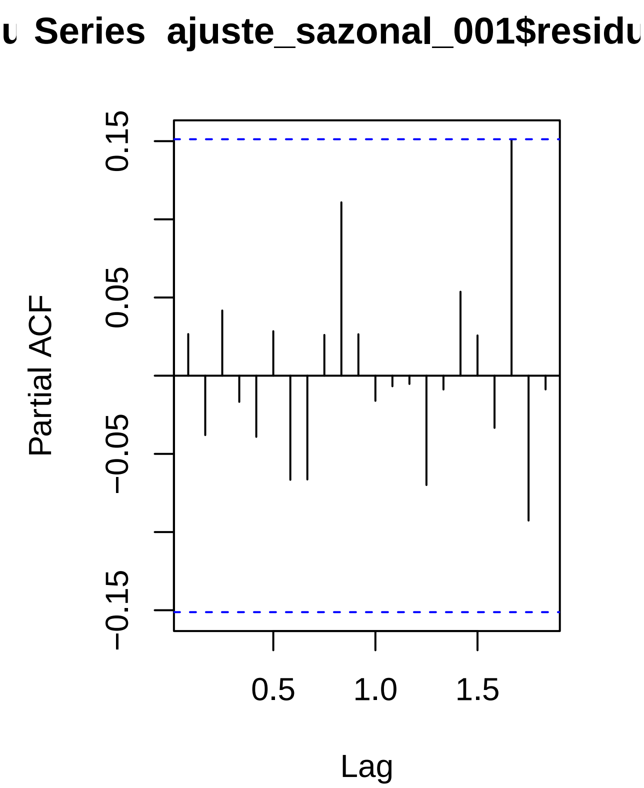
```
## Series: tuberculose_ts
## ARIMA(1,0,1)(0,0,1)[12] with non-zero mean
##
## Coefficients:
##          ar1      ma1    sma1     mean
##       0.9335  -0.7411  0.3461  98.8361
## s.e.  0.0567   0.1072  0.0837   4.7186
##
## sigma^2 = 159.3:  log likelihood = -663.47
## AIC=1336.94   AICc=1337.31   BIC=1352.56
```

```r
confint(ajuste_sazonal_001)
```

```
##                 2.5 %      97.5 %
## ar1         0.8224898   1.0445970
## ma1        -0.9512833  -0.5309338
## sma1        0.1820164   0.5100917
## intercept  89.5877889 108.0843471
```

```r
acf(ajuste_sazonal_001$residuals)
pacf(ajuste_sazonal_001$residuals)
```





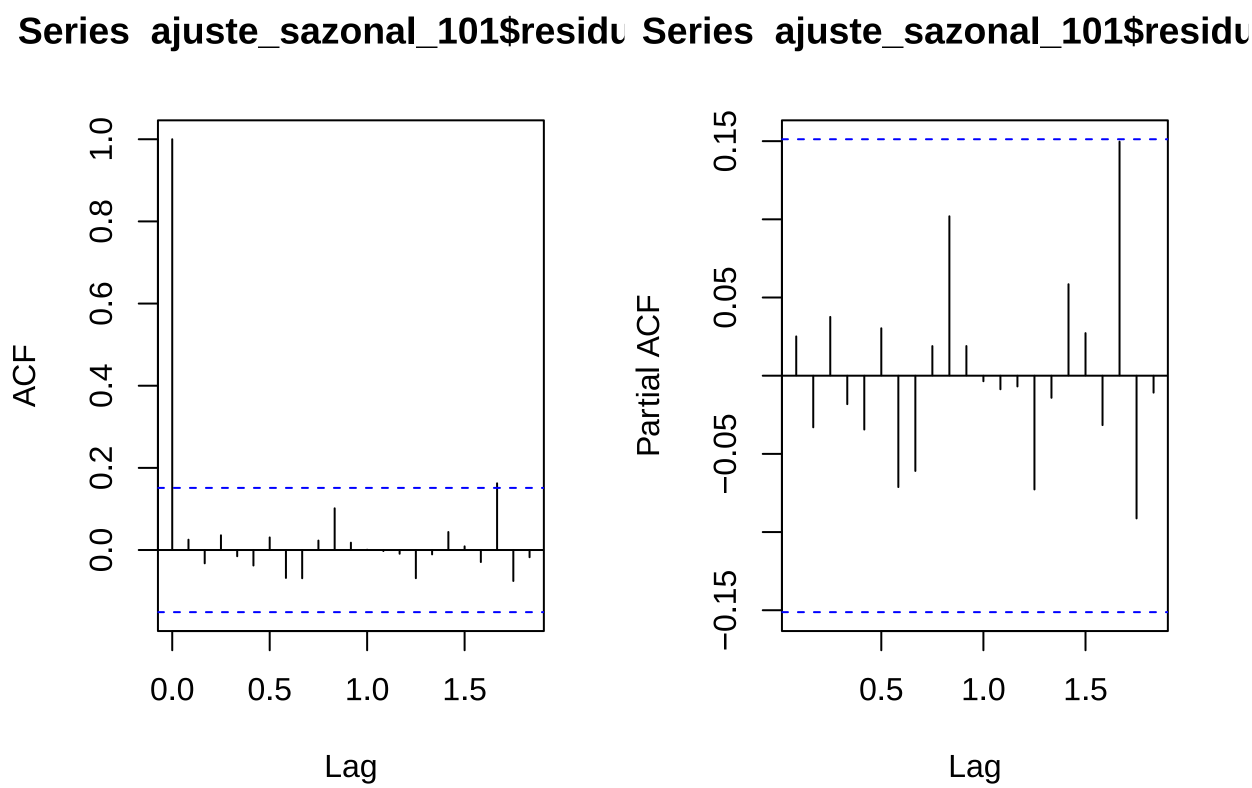

```r
# SARIMA(1,0,1)(1,0,1)
ajuste_sazonal_101 <- Arima(tuberculose_ts, c(1,0,1), seasonal = c(1, 0, 1))
ajuste_sazonal_101
```

```
## Series: tuberculose_ts
## ARIMA(1,0,1)(1,0,1)[12] with non-zero mean
##
## Coefficients:
##          ar1      ma1     sar1    sma1     mean
##       0.9325  -0.7339  -0.1355  0.4655  98.7762
## s.e.  0.0577   0.1111   0.2109  0.1915   4.5979
##
## sigma^2 = 159.9:  log likelihood = -663.27
## AIC=1338.54   AICc=1339.06   BIC=1357.28
```

```r
confint(ajuste_sazonal_101)
```

```
##                 2.5 %      97.5 %
## ar1        0.81942841   1.0456694
## ma1       -0.95168183  -0.5161752
## sar1      -0.54876849   0.2778649
## sma1       0.09013774   0.8408372
## intercept 89.76445157 107.7879537
```

```r
acf(ajuste_sazonal_101$residuals)
pacf(ajuste_sazonal_101$residuals)
```




```r
# valor dos criterios
# SARIMA(1,0,1)(1,0,0) - AIC=1340.28
# SARIMA(1,0,1)(0,0,1) - AIC=1336.94
# SARIMA(1,0,1)(1,0,1) - AIC=1338.54

# buscanco o melhor modelo utilizando
# o auto.arima

ajuste_auto_arima <- auto.arima(tuberculose_ts)
ajuste_auto_arima
```

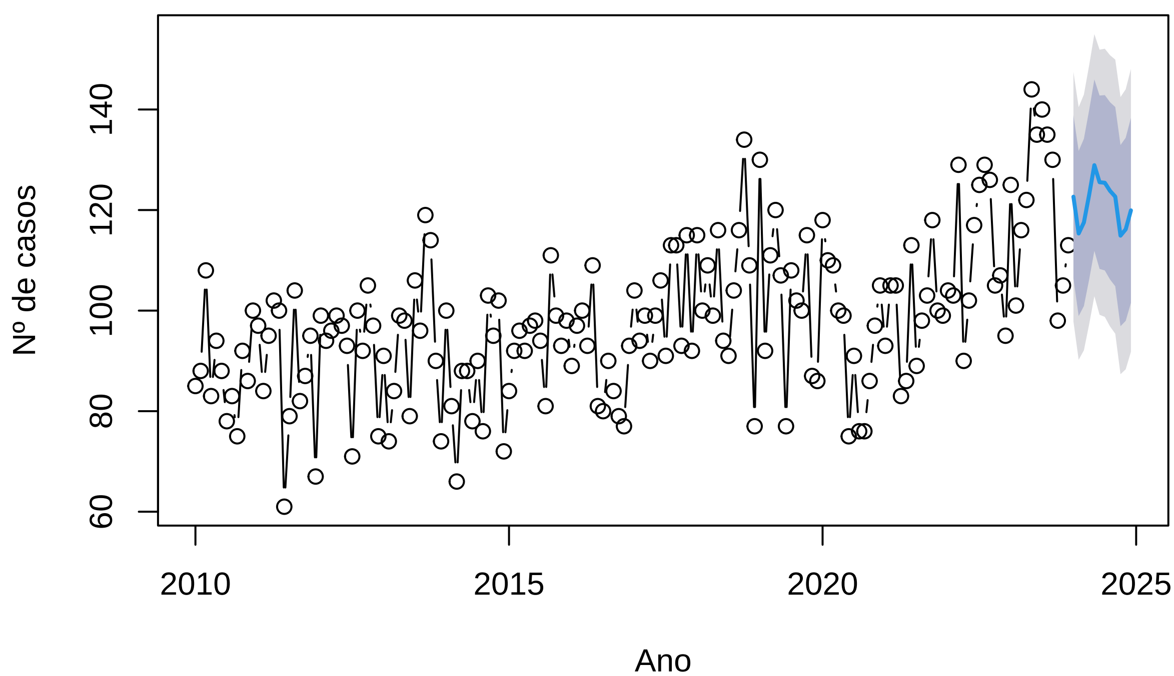
```
## Series: tuberculose_ts
## ARIMA(0,1,1)(0,0,1)[12]
##
## Coefficients:
##           ma1    sma1
##       -0.8385  0.3250
## s.e.   0.0527  0.0809
##
## sigma^2 = 160.1:  log likelihood = -661.01
## AIC=1328.03   AICc=1328.17   BIC=1337.38
```

```r
# previsao
previsao <- forecast(ajuste_auto_arima, h = 12)
```

```r
previsao
```

```
##          Point Forecast     Lo 80    Hi 80     Lo 95    Hi 95
## Jan 2024       122.6560 106.44150 138.8705  97.85808 147.4539
## Feb 2024       115.3380  98.91349 131.7625  90.21888 140.4572
## Mar 2024       117.5381 100.90622 134.1700  92.10182 142.9745
## Apr 2024       122.9946 106.15786 139.8314  97.24502 148.7442
## May 2024       128.9260 111.88686 145.9651 102.86689 154.9851
## Jun 2024       125.5345 108.29536 142.7736  99.16952 151.8994
## Jul 2024       125.4221 107.98528 142.8589  98.75478 152.0894
## Aug 2024       123.8205 106.18823 141.4529  96.85425 150.7869
## Sep 2024       122.6643 104.83869 140.4900  95.40235 149.9263
## Oct 2024       114.9113  96.89437 132.9282  87.35678 142.4658
## Nov 2024       116.1706  97.96437 134.3768  88.32660 144.0145
## Dec 2024       119.9386 101.54506 138.3321  91.80813 148.0690
```

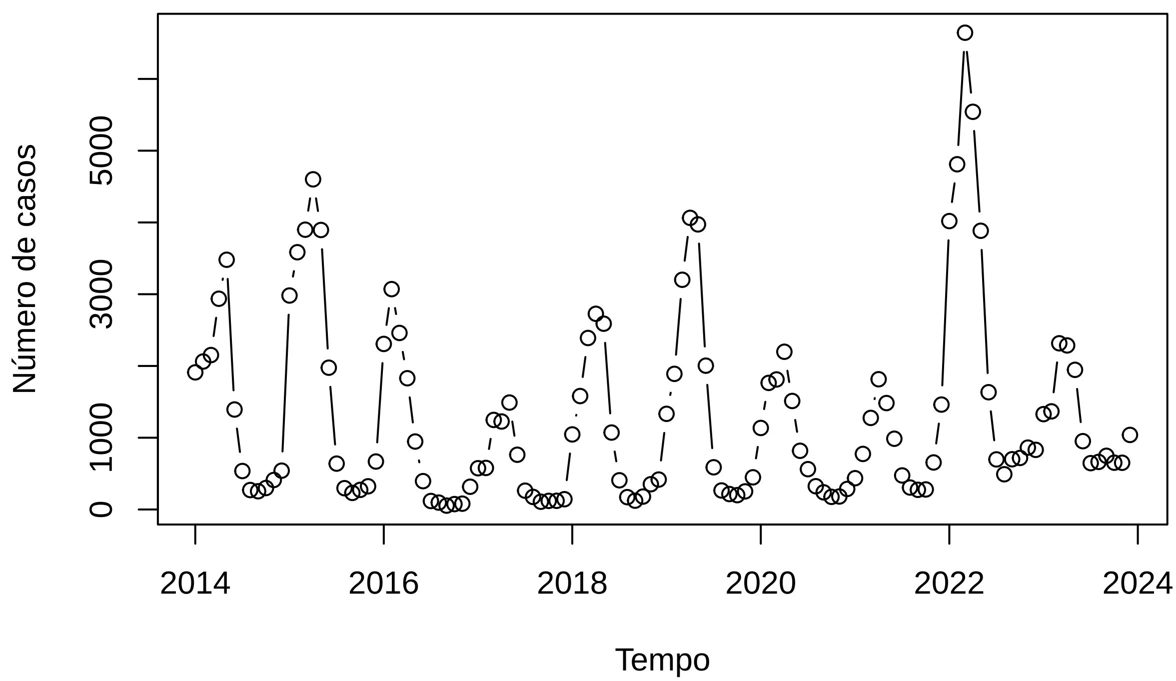
```r
# grafico com os valores observados e preditos
par(mfrow=c(1,1))
plot(previsao, type="b", ylab="Nº de casos", xlab="Ano", main="")
```



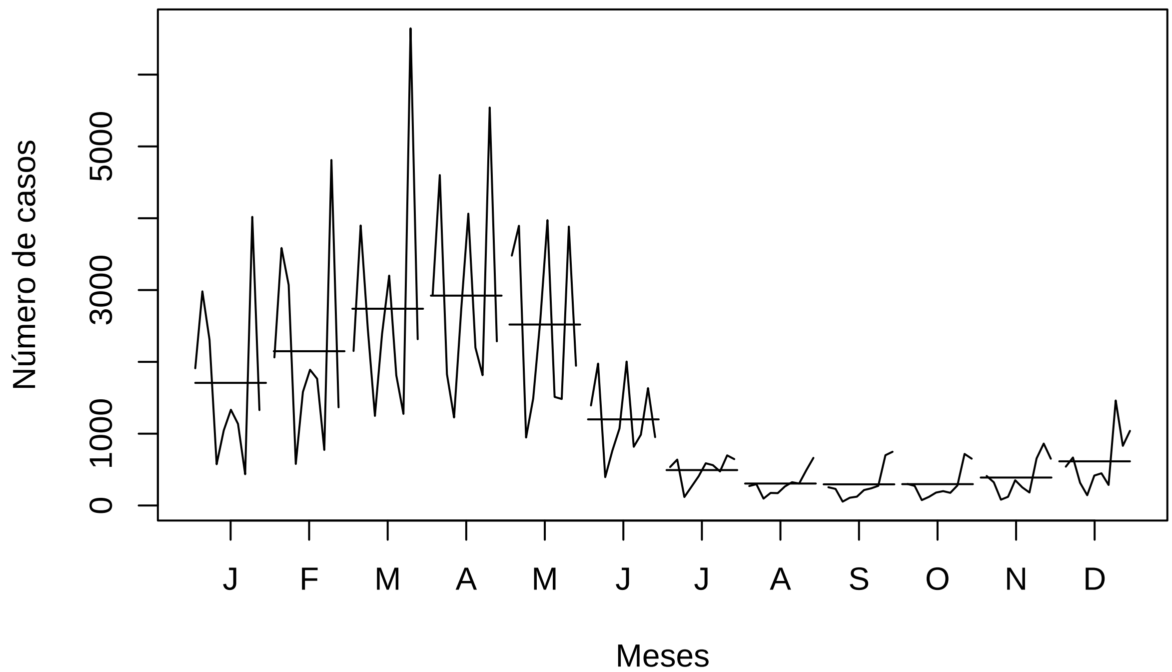
Série temporal do número de casos de dengue

A seguir é apresentada a série temporal do número de casos de dengue no Estado de Goiás (UF da residência é Goiás) com exame sorológico (IgM) positivo, compreende o número de casos mensal de janeiro de 2014 a dezembro de 2023. Os valores foram obtidos a partir do Sistema de Informação de Agravos de Notificação - Sinan Net - acessado em 20/09/2024, no link aqui.

```r
# serie do numero de casos mensal
serie_dengue <- c(1911, 2062, 2152, 2937, 3480, 1393, 536, 271, 254, 299, 411, 541,
                  2982, 3585, 3899, 4601, 3895, 1976, 639, 297, 231, 273, 323, 668,
                  2307, 3071, 2460, 1829, 947, 395, 118, 96, 54, 75, 81, 317,
                  575, 579, 1248, 1226, 1490, 763, 262, 175, 108, 120, 121, 143,
                  1046, 1581, 2390, 2727, 2589, 1072, 407, 172, 123, 180, 352, 417,
                  1333, 1890, 3201, 4064, 3973, 2004, 588, 265, 215, 200, 252, 448,
                  1136, 1764, 1812, 2198, 1512, 818, 562, 324, 239, 176, 181, 286,
                  436, 774, 1276, 1815, 1483, 986, 474, 306, 274, 279, 655, 1462,
                  4019, 4811, 6644, 5542, 3884, 1634, 698, 491, 700, 718, 862, 831,
                  1329, 1367, 2316, 2286, 1946, 952, 646, 663, 748, 652, 651, 1039)

dengue_ts <- ts(serie_dengue, start= c(2014, 1), frequency = 12)

plot(dengue_ts, type="b", ylab="Número de casos", xlab="Tempo")
```




```r
library(forecast)
# temos sazonalidade?
monthplot(dengue_ts,  xlab="Meses", ylab="Número de casos", main="")
```




```r
# usando o grafico acima, temos!
# a sazonalidade e deterministica?

# A sazonalidade nao e deterministica
# Nao podemos usar as tecnicas de regressao
# Vamos proceder com as tecnicas para sazonalidade nao deterministica

# vamos utilizar a funcao auto.arima

ajuste1 <- auto.arima(dengue_ts)
ajuste1
```

```
## Series: dengue_ts
## ARIMA(1,0,3)(0,1,1)[12]
##
## Coefficients:
##          ar1     ma1     ma2     ma3     sma1
##       0.3858  0.8146  0.7832  0.3074  -0.6846
## s.e.  0.2932  0.3020  0.3670  0.2563   0.1063
##
## sigma^2 = 222206:  log likelihood = -820.34
## AIC=1652.68   AICc=1653.51   BIC=1668.77
```

```r
confint(ajuste1)
```

```
##            2.5 %     97.5 %
## ar1  -0.18884405  0.9604451
```

```
## ma1   0.22262326  1.4066335
## ma2   0.06384351  1.5025522
## ma3  -0.19486754  0.8096784
## sma1 -0.89293864 -0.4762321
```

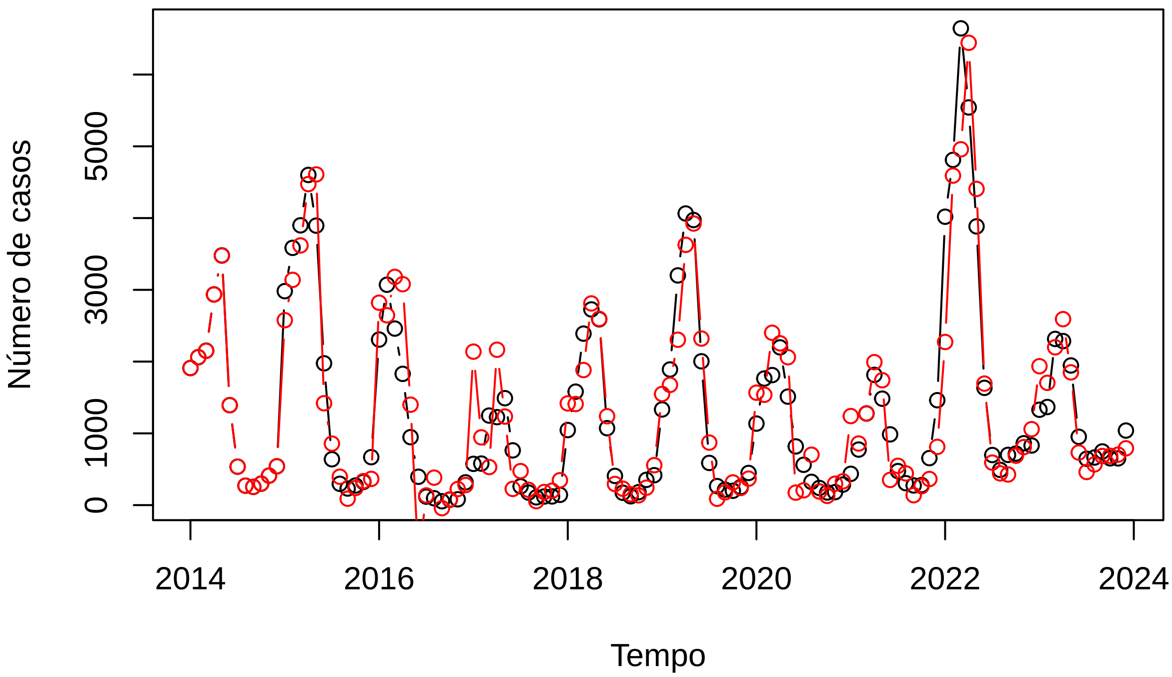
```
# note que a ordem MA(3) nao foi significativa
# vamos refazer o ajuste, mas agora usando o modelo
# SARIMA(1,0,2)X(0,1,1)12
ajuste2 <- Arima(dengue_ts, order=c(1,0,2), seasonal = c(0, 1, 1))
ajuste2
```

```
## Series: dengue_ts
## ARIMA(1,0,2)(0,1,1)[12]
##
## Coefficients:
##          ar1     ma1     ma2     sma1
##       0.6696  0.4764  0.3607  -0.7282
## s.e.  0.0863  0.0991  0.1280   0.1178
##
## sigma^2 = 222447:  log likelihood = -821.49
## AIC=1652.98   AICc=1653.57   BIC=1666.39
```

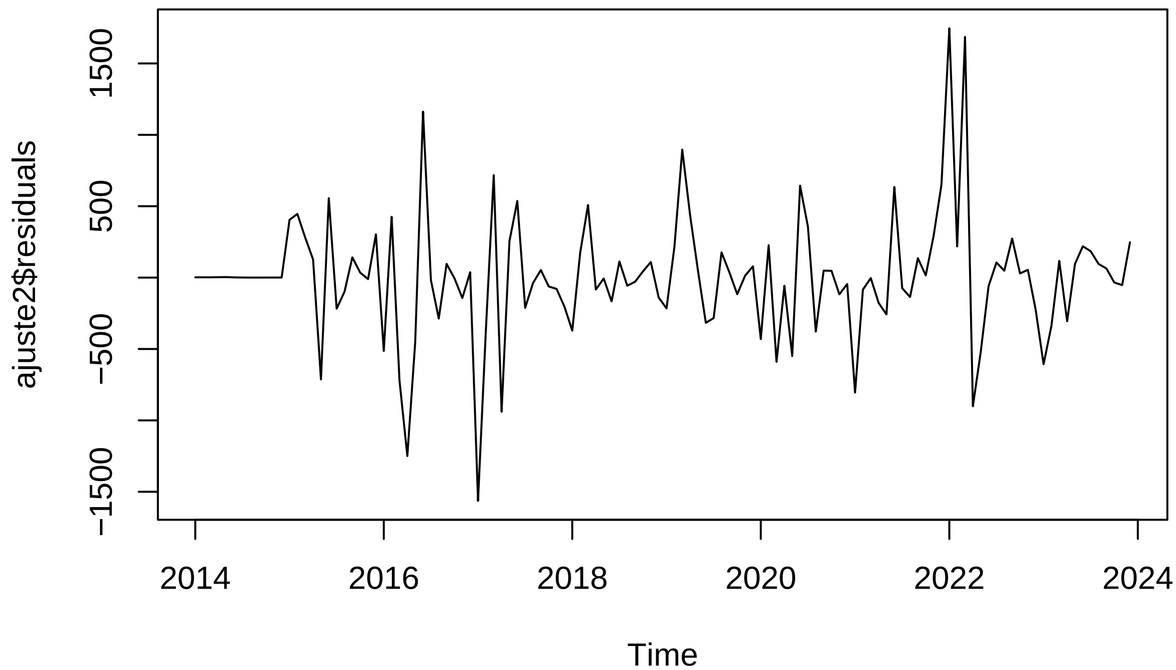
```
confint(ajuste2)
```

```
##           2.5 %     97.5 %
## ar1   0.5003839  0.8387313
## ma1   0.2822757  0.6706239
## ma2   0.1098945  0.6115528
## sma1 -0.9590338 -0.4972873
```

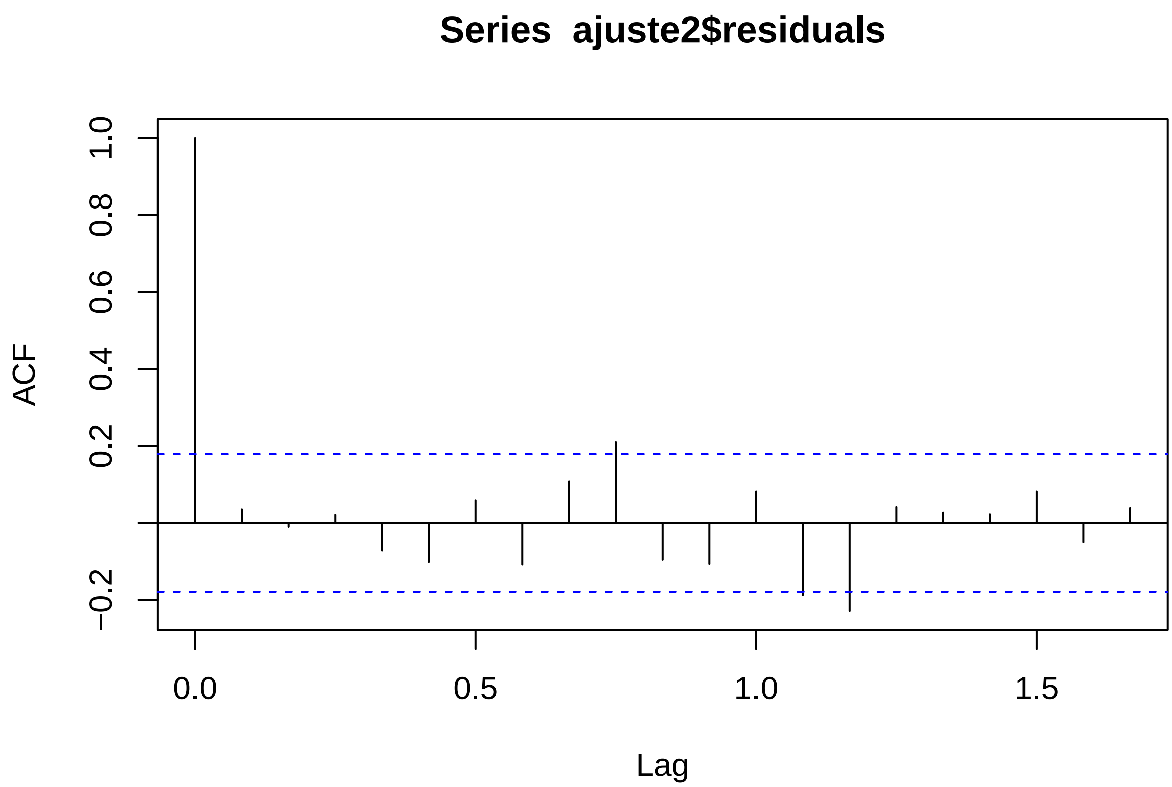
```
# Observe que agora todos os parametros foram significativos
# Vamos agora observar o modelo ajustado
plot(dengue_ts, type="b", ylab="Número de casos", xlab="Tempo")
lines(ajuste2$fitted, col="red", type="b")
```



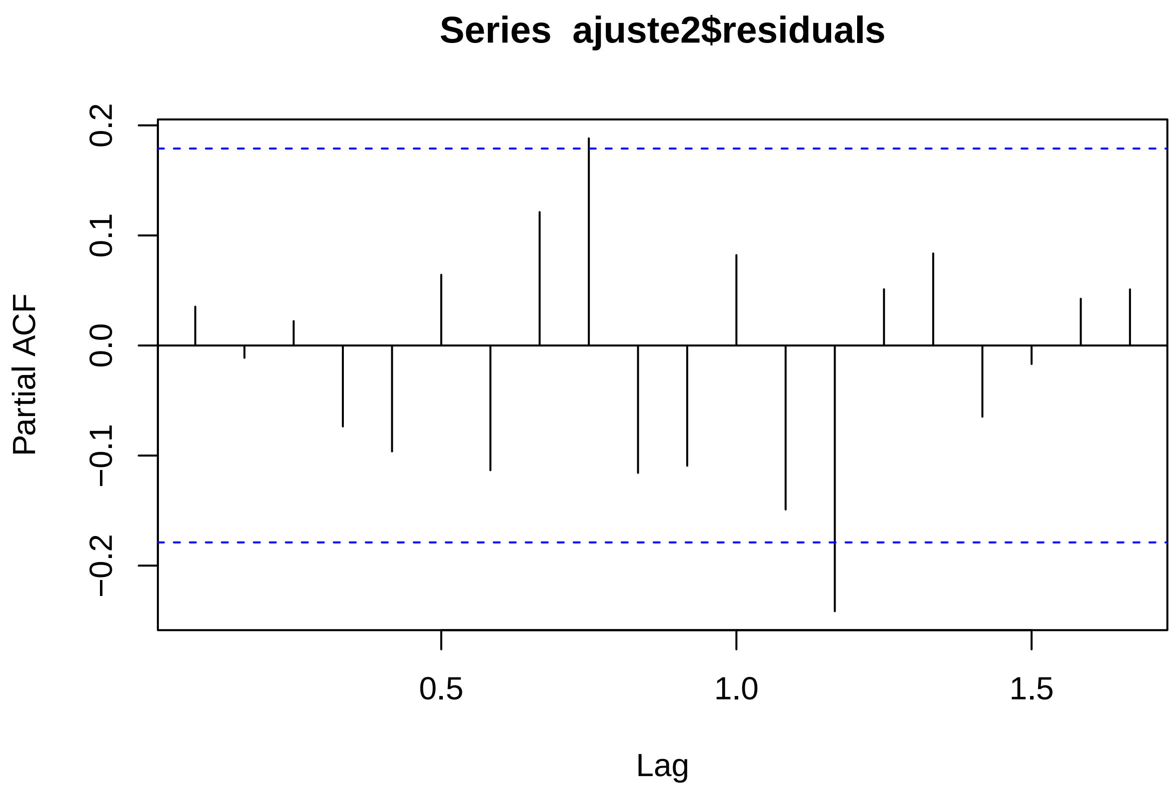

```r
# agora o plot dos residuos
plot(ajuste2$residuals)
```




```r
# acf e pacf dos residuos
acf(ajuste2$residuals)
```




```r
pacf(ajuste2$residuals)
```




```r
# discussao com especialista da area
# cuidado com correlacai espuria


# teste para verificar se os residuos sao ruido branco
Box.test(ajuste2$residuals, lag = 12)
```

```
##
##  Box-Pierce test
##
## data:  ajuste2$residuals
## X-squared = 13.837, df = 12, p-value = 0.3112
```

```r
# vamos agora realizar previsao
forecast(ajuste2, h=12)
```

```
##          Point Forecast       Lo 80    Hi 80      Lo 95    Hi 95
## Jan 2024      2073.2595 1468.42703 2678.092  1148.2480 2998.271
## Feb 2024      2469.7163 1549.93813 3389.495  1063.0370 3876.396
## Mar 2024      3245.4967 2100.42525 4390.568  1494.2610 4996.732
## Apr 2024      3178.7222 1945.94517 4411.499  1293.3524 5064.092
## May 2024      2549.0246 1278.89361 3819.156   606.5269 4491.522
## Jun 2024      1242.8537  -43.67028 2529.378  -724.7150 3210.422
## Jul 2024       605.1414 -688.66201 1898.945 -1373.5602 2583.843
## Aug 2024       444.4793 -852.56912 1741.528 -1539.1851 2428.144
## Sep 2024       473.5991 -824.89019 1772.088 -1512.2689 2459.467
## Oct 2024       445.9381 -853.17142 1745.048 -1540.8785 2432.755
```

```
## Nov 2024       546.1041 -753.22725 1845.435 -1441.0517 2533.260
## Dec 2024       815.8058 -483.58244 2115.194 -1171.4370 2803.049
```

```r
plot(forecast(ajuste2, h=12), xlab="Meses", ylab="Número de casos", main="Previsao do modelo SARIMA(1,0
```