Trabalho 2 de Análise de Série Temporal

Carlos Lopes, Cristiane Fractal, Cristovão Moreira Freitas Junior, Fábio Karpusca Marin, Renato Bastos

20/04/2021

Apresentação do caso:

Despesas de telefonia móvel – Banco X Os dados apresentados são referentes a despesas somadas de todos os contratos de prestação de serviço de telefonia móvel, de janeiro de 2015 até março de 2021. Ao longo desse período foram assinados 12 contratos, com vigência de 5 anos cada, para uma média de 18.000 linhas ativas entre 2015 e 2019, e 25.000 linhas ativas a partir de 2020. Até o final de 2020 os contratos faturavam os seguintes serviços: • Assinatura mensal da linha e cessão de comodato do dispositivo; • Ligações telefônicas, por minuto e por destinação; • SMS; • Pacote de dados; • Roaming nacional e internacional; • Outros serviços de valor agregado. A partir de 2021 os contratos passarão a faturar os seguintes serviços: • Assinatura mensal da linha incluindo pacote de dados; • Ligações nacionais, SMS e roaming nacional são ilimitados, sem custo adicional; • Roaming internacional e outros serviços de valor agregado são cobrados à parte.

Importar a base de dados

```
Dados_Telefonia_Movel <- read_excel("Dados Telefonia Movel.xlsx")
db <- Dados_Telefonia_Movel
```

Converter a base de dados em série temporal:

```
db_ts <- ts(db$`Telefonia Móvel`, start=c(2015, 1), end=c(2021, 3), frequency = 12)
```

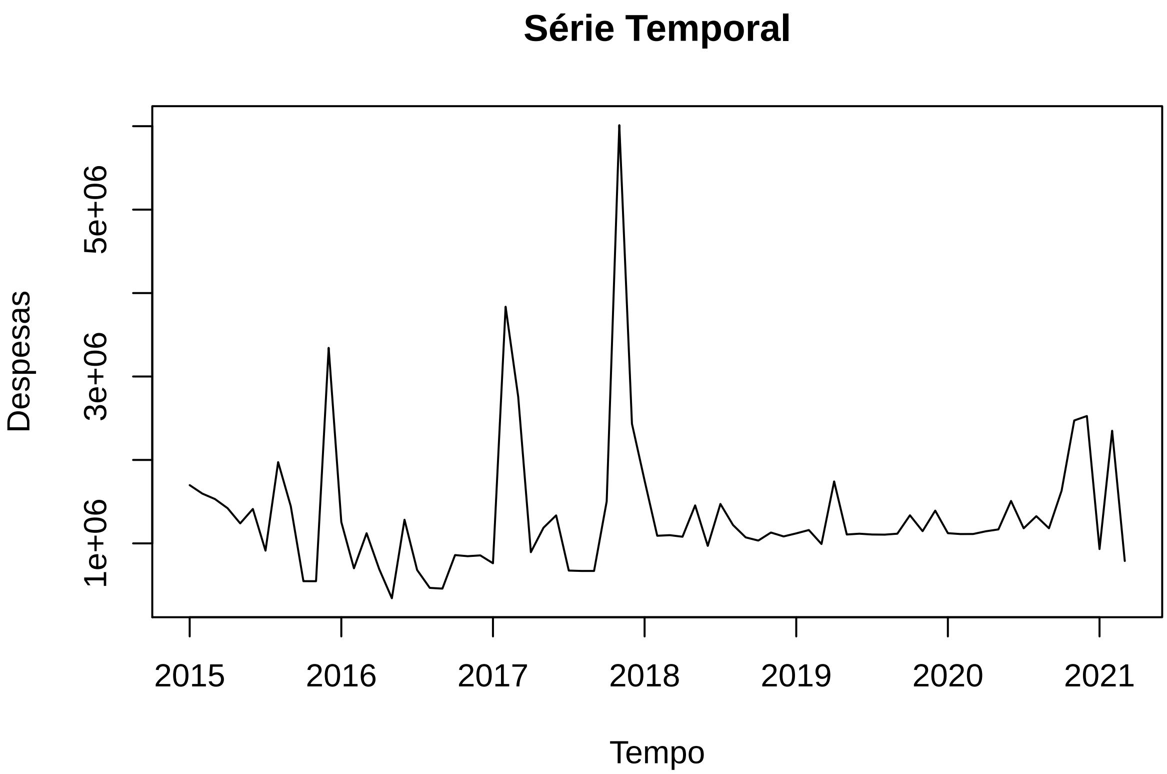
Análise estatística da série:

```
summary(db_ts)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  342538  950886 1130180 1339072 1450561 6012284
```

Plotar o gráfico da série temporal:

```
plot(db_ts, xlab="Tempo", ylab="Despesas", ylim=c(342000, 6013000), type="l", main="Série Temporal")
```



Definir o tamanho da amostra de treinamento e da amostra de teste:

```
amostra_validacao <- 10
amostra_treino <- length(db_ts) - amostra_validacao
```

Programar a amostra de treinamento:

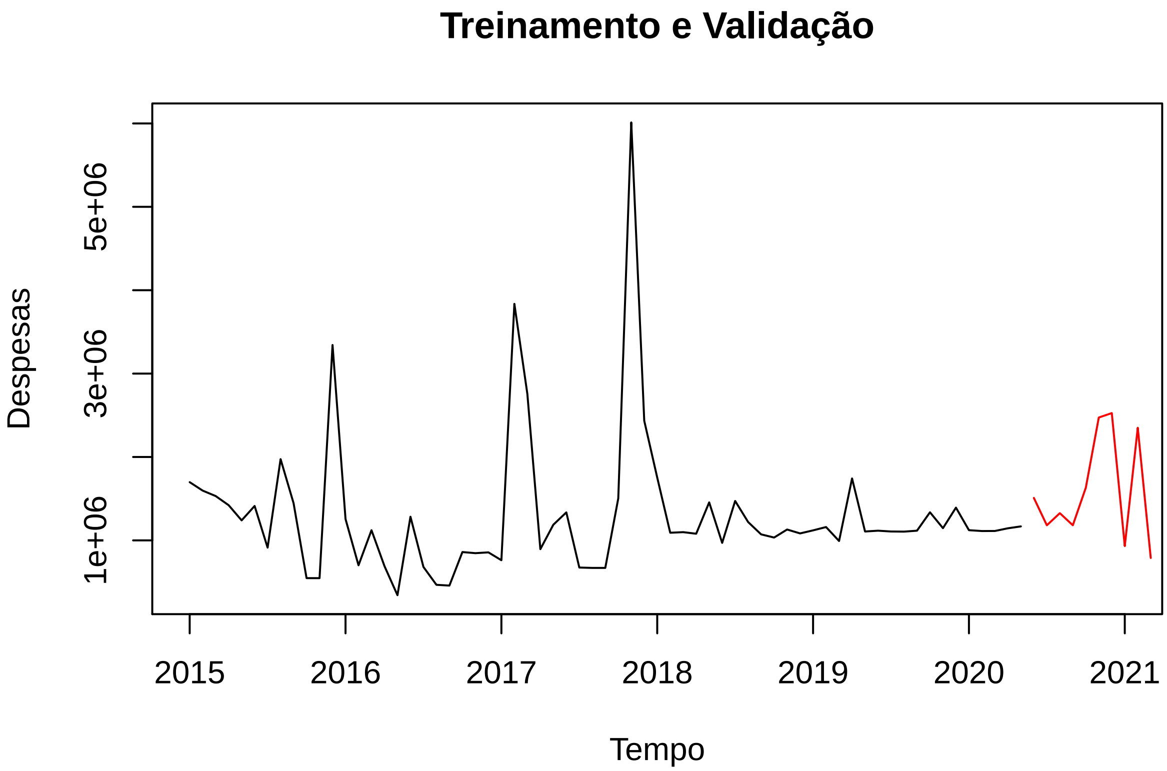
```
treinamento_ts <-window(db_ts, start=c(2015, 1), end=c(2015, amostra_treino))
```

Programar a amostra de validação:

```
validacao_ts <- window(db_ts, start=c(2015, amostra_treino + 1), end=c(2015, amostra_treino + amostra_va
```

PLotagem do gráfico do treinamento com validação:

```
plot(treinamento_ts, xlab="Tempo", ylab="Despesas", xaxt="n" , ylim=c(342000, 6013000), xlim=c(2015, 202
axis(1, at=seq(2015, 2021,1), labels=format(seq(2015, 2021,1)))
lines(validacao_ts, bty="l", col="red")
```



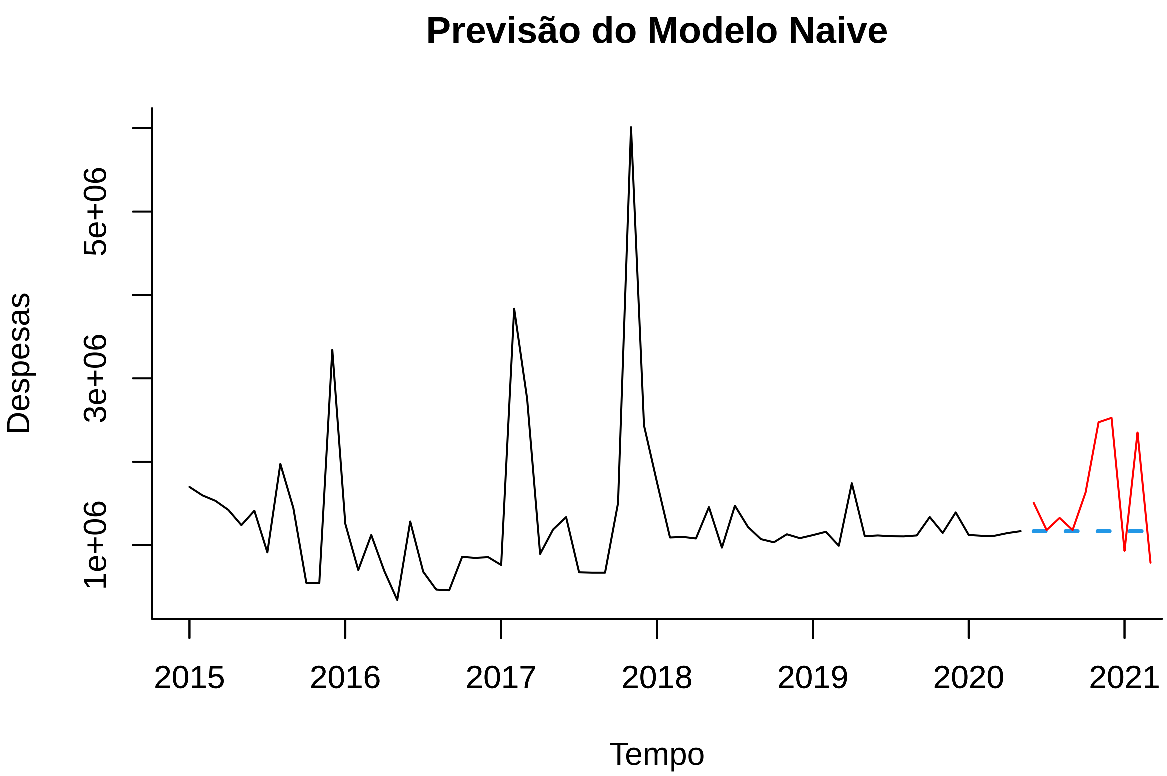
Confecção do modelo Naive:

```
modelo_naive <- naive(treinamento_ts, level=0, h=amostra_validacao)
accuracy(modelo_naive, validacao_ts)
```

```
##                      ME      RMSE      MAE       MPE     MAPE      MASE
## Training set  -8289.223 1032944.7 539975.9 -14.64966 36.97608 0.6795305
## Test set     422413.979  742063.4 545238.7  14.89507 29.54620 0.6861534
##                    ACF1 Theil's U
## Training set -0.2840960        NA
## Test set     -0.1692134 0.8705138
```

Grafico da série temporal de treinamento, validação e modelo naive

```
plot(modelo_naive, xlab="Tempo", ylab="Despesas", xaxt="s" , ylim=c(342000, 6013000), xlim=c(2015, 2021)
axis(1, at=seq(2015, 2021,1), labels=format(seq(2015, 2021,1)))
lines(validacao_ts, bty="l", col="red")
```

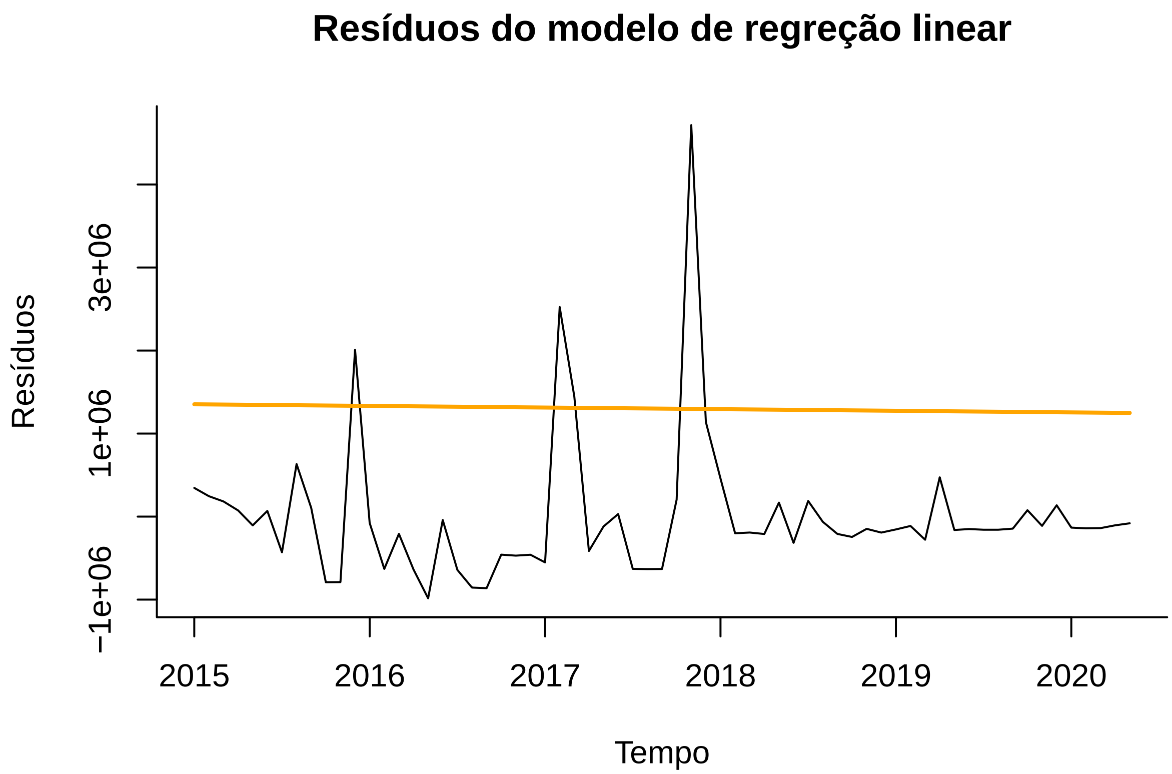


#Modelo de Tendência linear:

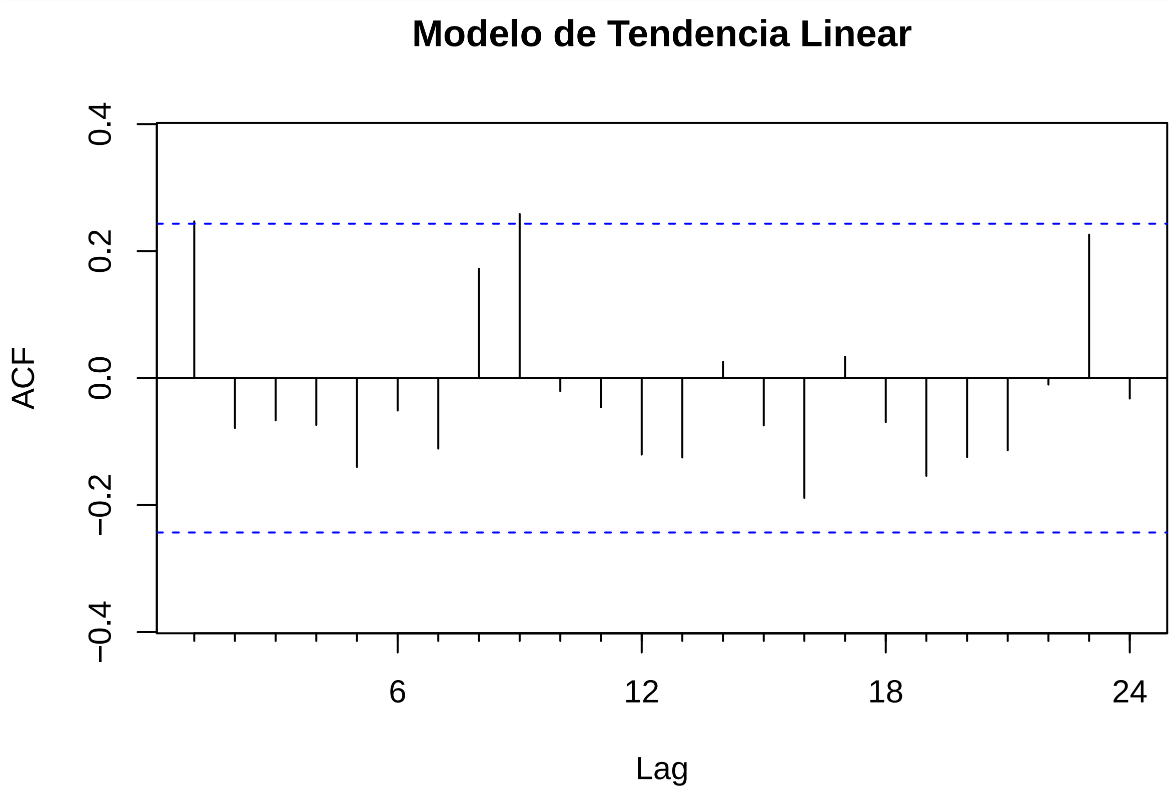
```
modelo_tendencia_linear <- tslm(treinamento_ts ~ trend)
summary(modelo_tendencia_linear)
```

```
##
## Call:
## tslm(formula = treinamento_ts ~ trend)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -984002 -414252 -145148   76907 4715064
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1354232     213063   6.356 2.61e-08 ***
## trend          -1629       5613  -0.290    0.773
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 849000 on 63 degrees of freedom
## Multiple R-squared:  0.001335,	Adjusted R-squared:  -0.01452
## F-statistic: 0.08422 on 1 and 63 DF,  p-value: 0.7726
```

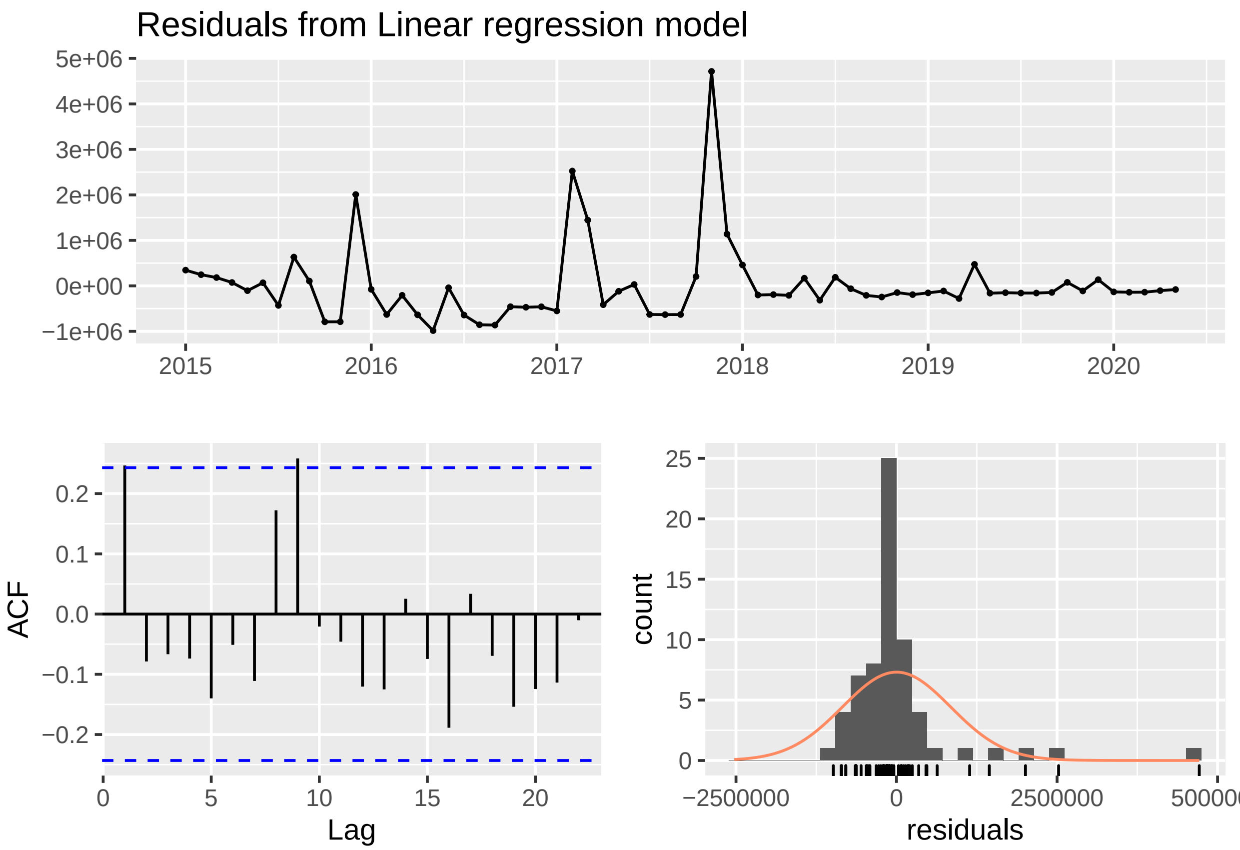
```
plot(modelo_tendencia_linear$residuals, xlab="Tempo", ylab="Resíduos", bty="l", main="Resíduos do model
lines(modelo_tendencia_linear$fitted.values, lwd="2", col="orange")
```




```
Acf(modelo_tendencia_linear$residuals, main="Modelo de Tendencia Linear")
```

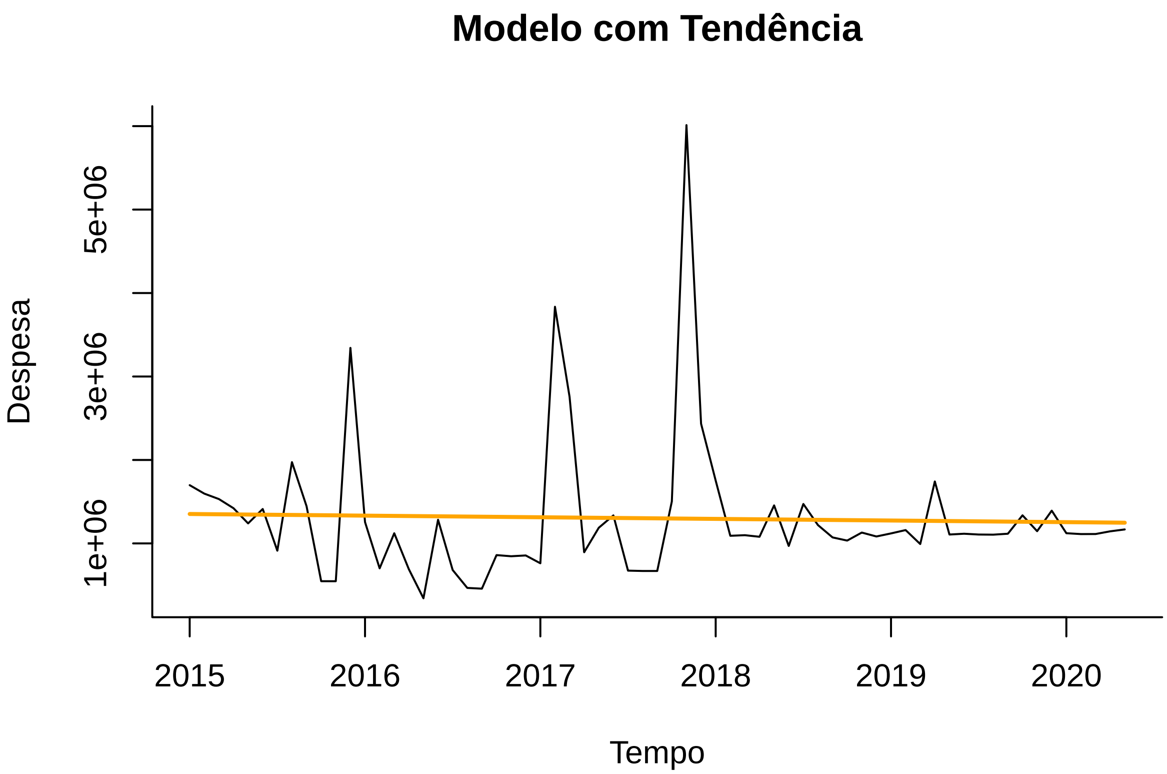



```
checkresiduals(modelo_tendencia_linear, test="LB")
```

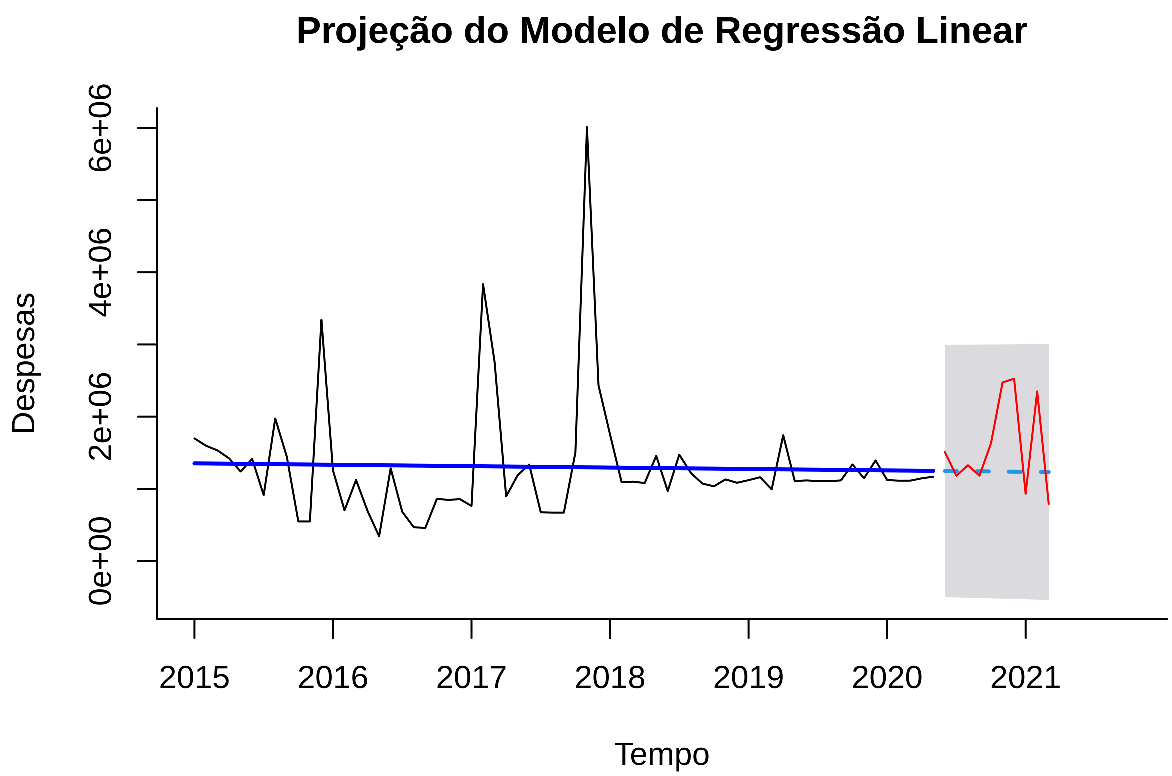



```
##
##  Ljung-Box test
##
## data:  Residuals from Linear regression model
## Q* = 17.981, df = 11, p-value = 0.08202
##
## Model df: 2.   Total lags used: 13
```

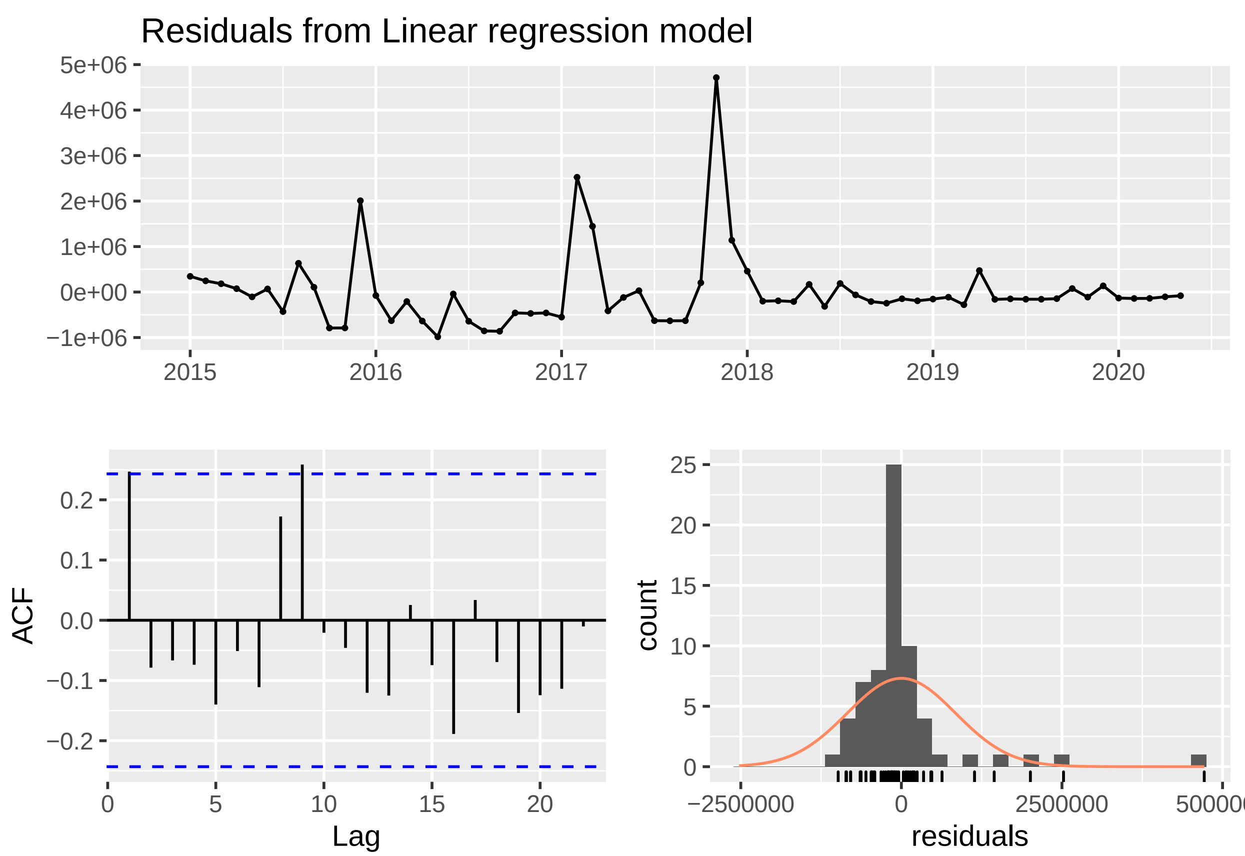
```
plot(treinamento_ts, xlab="Tempo", ylab="Despesa", bty="l", main="Modelo com Tendência")
lines(modelo_tendencia_linear$fitted.values, lwd="2", col="orange")
```




```
modelo_tendencia_linear_proj <- forecast(modelo_tendencia_linear, h=amostra_validacao, level=0.95)

plot(modelo_tendencia_linear_proj, xlab="Tempo", ylab="Despesas", xaxt="n" , xlim=c(2015, 2021.75), bty=
axis(1, at=seq(2015, 2021, 1), labels=format(seq(2015,2021,1)))
lines(validacao_ts, col="red")
lines(modelo_tendencia_linear_proj$fitted, lwd="2", col="blue")
```




```
checkresiduals(modelo_tendencia_linear, test="LB")
```




```
##
```

```
##  Ljung-Box test
##
## data:  Residuals from Linear regression model
## Q* = 17.981, df = 11, p-value = 0.08202
##
## Model df: 2.   Total lags used: 13
```

```
accuracy(modelo_tendencia_linear_proj, validacao_ts)
```

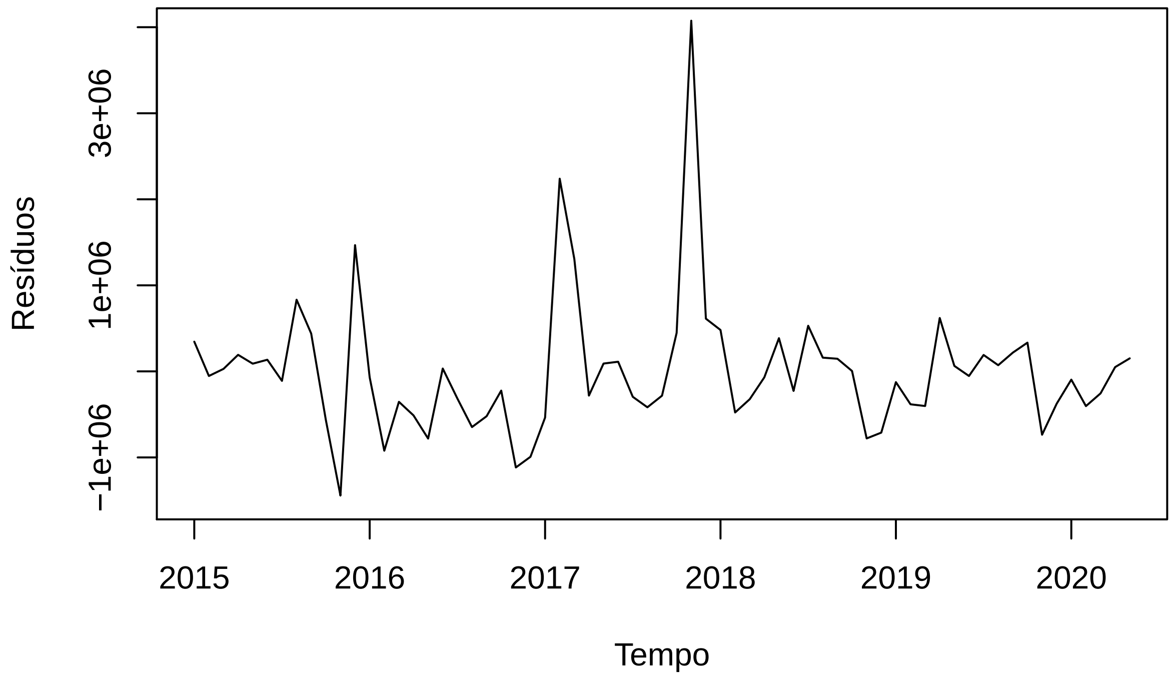
```
##                        ME     RMSE      MAE        MPE     MAPE      MASE
## Training set 1.969865e-11 835830.9 468390.5 -26.490410 42.17270 0.5894442
## Test set     3.505402e+05 704184.8 524848.6   9.676502 29.53702 0.6604936
##                    ACF1 Theil's U
## Training set  0.2467421        NA
## Test set     -0.1650778 0.8198874
```

Pelo baixo valor de R Square e AR-Squared, o Modelo de Tendência Linear não foi satisfatório. Partimos para o próximo modelo. #Modelo de Tendencia linear com Sazonalidade

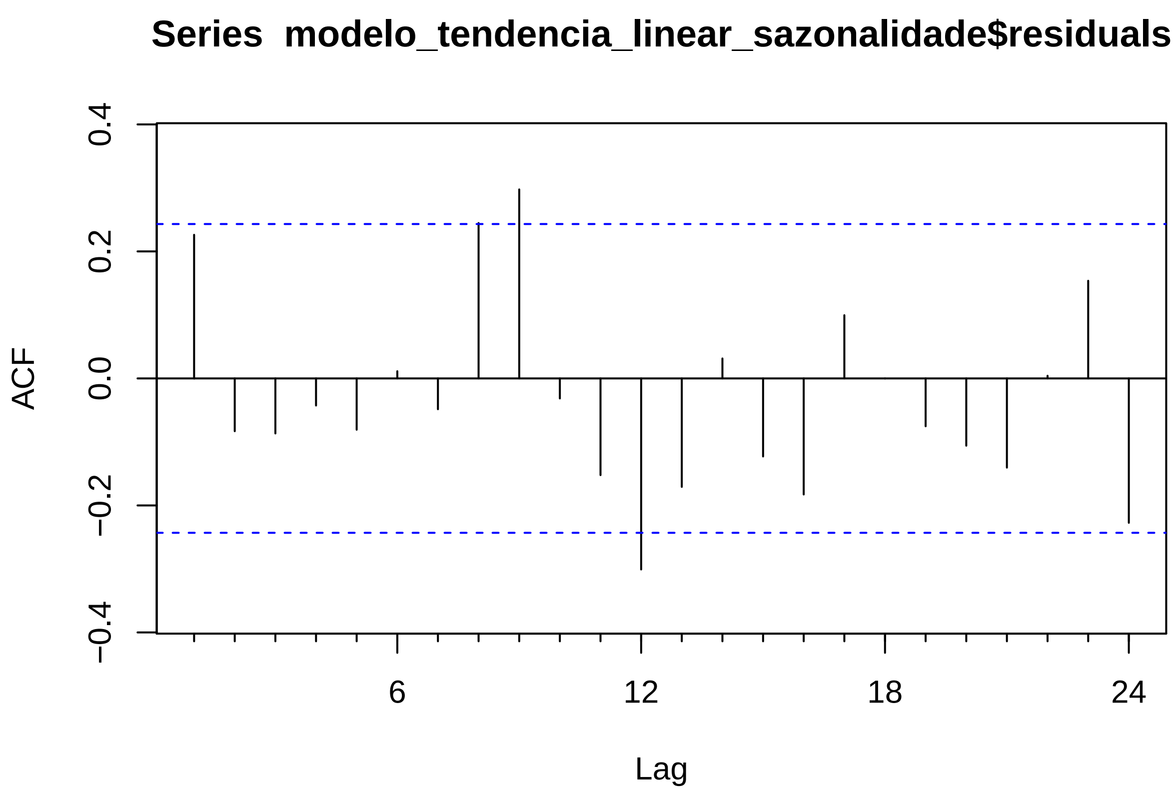
```
modelo_tendencia_linear_sazonalidade <- tslm(treinamento_ts ~ season+trend)
summary(modelo_tendencia_linear_sazonalidade)
```

```
##
## Call:
## tslm(formula = treinamento_ts ~ season + trend)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1443221  -402635   -69390   190751  4075813
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1354539     397366   3.409  0.00127 **
## season2       300114     501818   0.598  0.55240
## season3       154826     501917   0.308  0.75896
## season4      -115940     502083  -0.231  0.81829
## season5      -192768     502315  -0.384  0.70272
## season6       -63694     526308  -0.121  0.90414
## season7      -315609     526276  -0.600  0.55131
## season8      -195773     526308  -0.372  0.71142
## season9      -328337     526402  -0.624  0.53553
## season10     -222032     526561  -0.422  0.67501
## season11      660350     526782   1.254  0.21561
## season12      548013     527066   1.040  0.30327
## trend          -2240       5771  -0.388  0.69943
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 869100 on 52 degrees of freedom
## Multiple R-squared:  0.1362, Adjusted R-squared:  -0.06318
## F-statistic: 0.6831 on 12 and 52 DF,  p-value: 0.7597
```

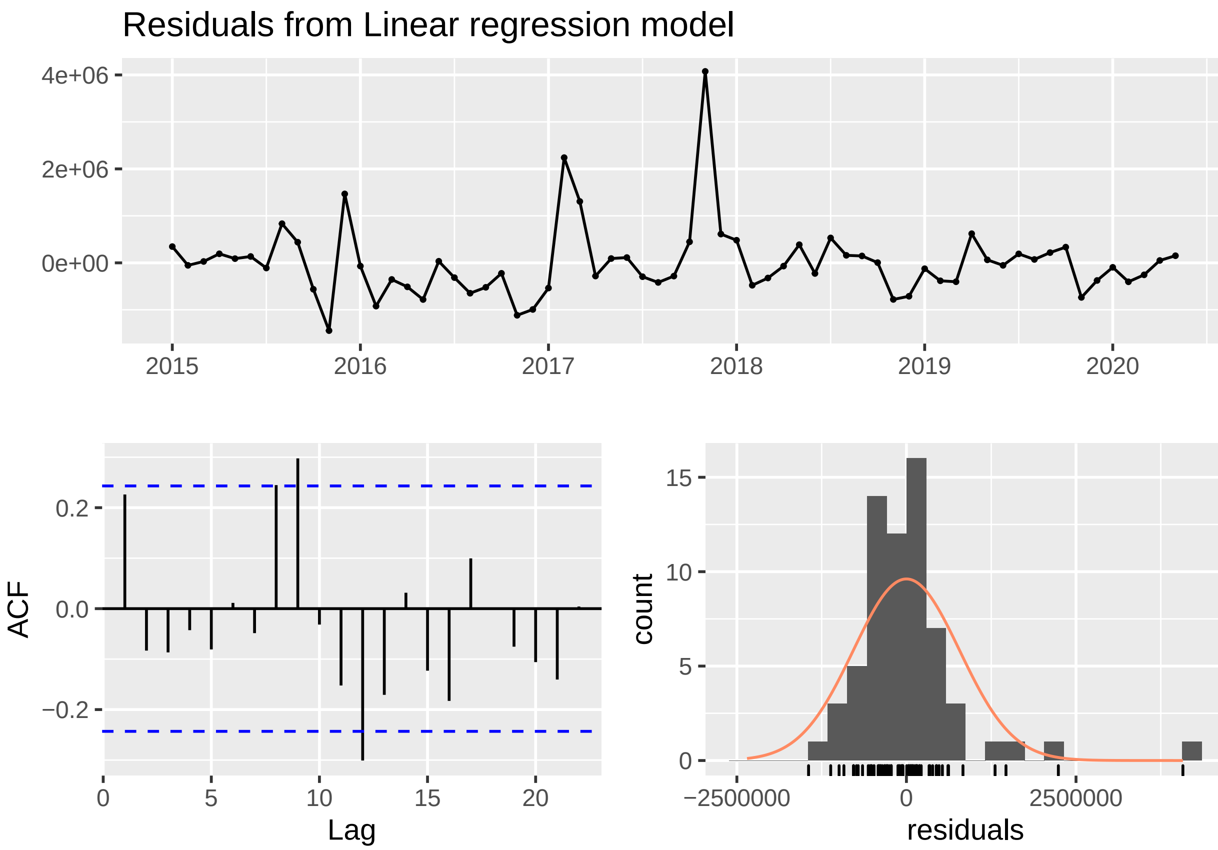
```
plot(modelo_tendencia_linear_sazonalidade$residuals, xlab="Tempo", ylab="Resíduos",ylim=c(-1500000, 400
```




```
Acf(modelo_tendencia_linear_sazonalidade$residuals)
```

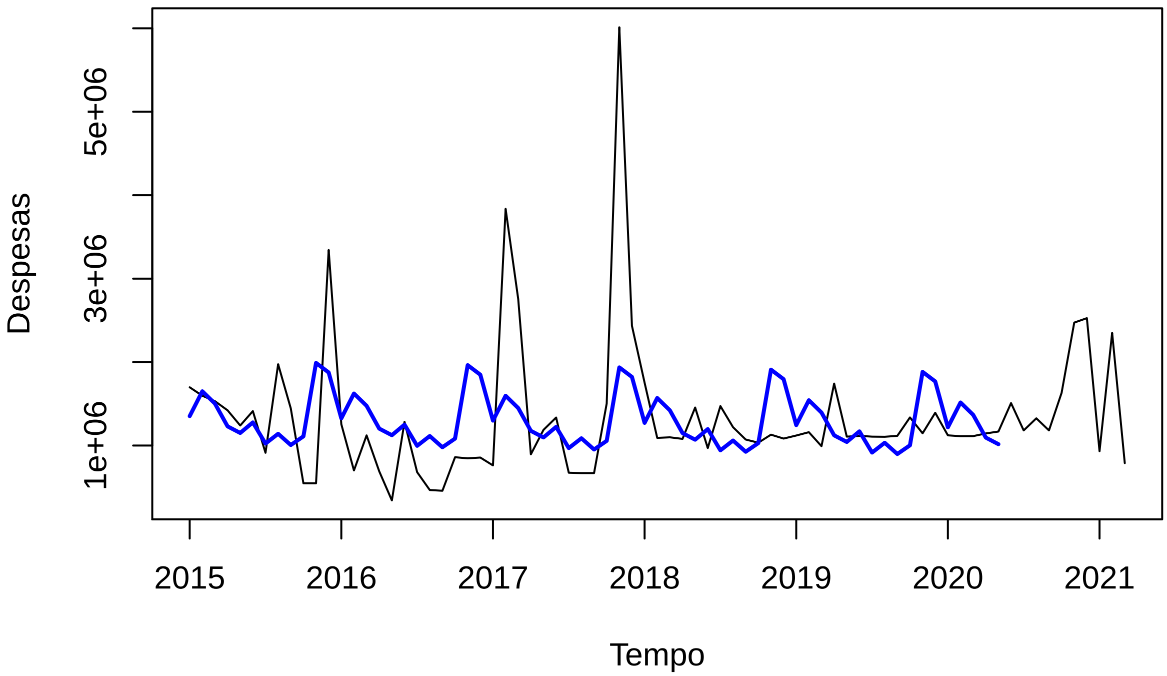



```
checkresiduals(modelo_tendencia_linear_sazonalidade, test="LB", main="Teste de Ljung-Box")
```

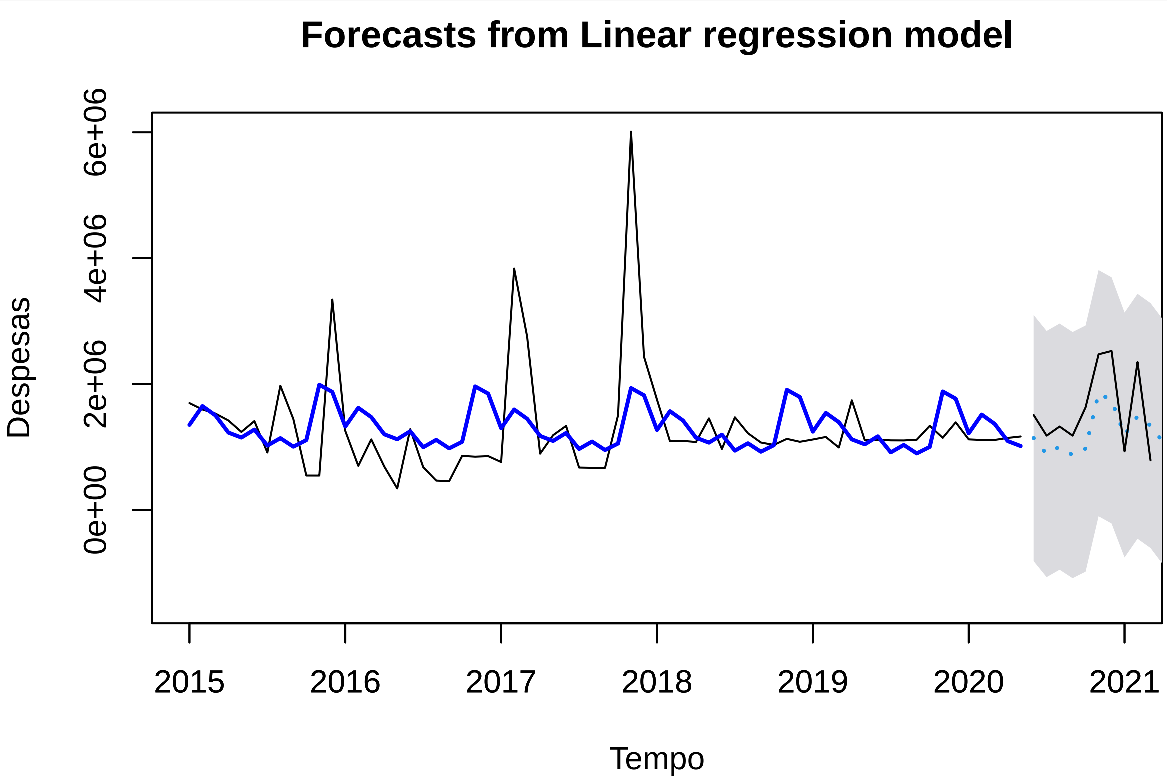



```
##
##  Ljung-Box test
##
## data:  Residuals from Linear regression model
## Q* = 32.937, df = 3, p-value = 3.32e-07
##
## Model df: 13.   Total lags used: 16
```

```
plot(db_ts, xlab="Tempo", ylab="Despesas", ylim=c(342000, 6013000), type="l")
lines(modelo_tendencia_linear_sazonalidade$fitted.values, lwd=2, col="Blue")
```




```
modelo_tendencia_linear_sazonalidade_proj <- forecast(modelo_tendencia_linear_sazonalidade, h = 55, leve

plot(modelo_tendencia_linear_sazonalidade_proj, xlab="Tempo", ylab="Despesas", xaxt="s" , ylim=c(-150000
axis(1, at=seq(2015, 2021), labels=format(seq(2015, 2021)))
lines(validacao_ts)
lines(modelo_tendencia_linear_sazonalidade_proj$fitted, lwd=2, col="blue")
```



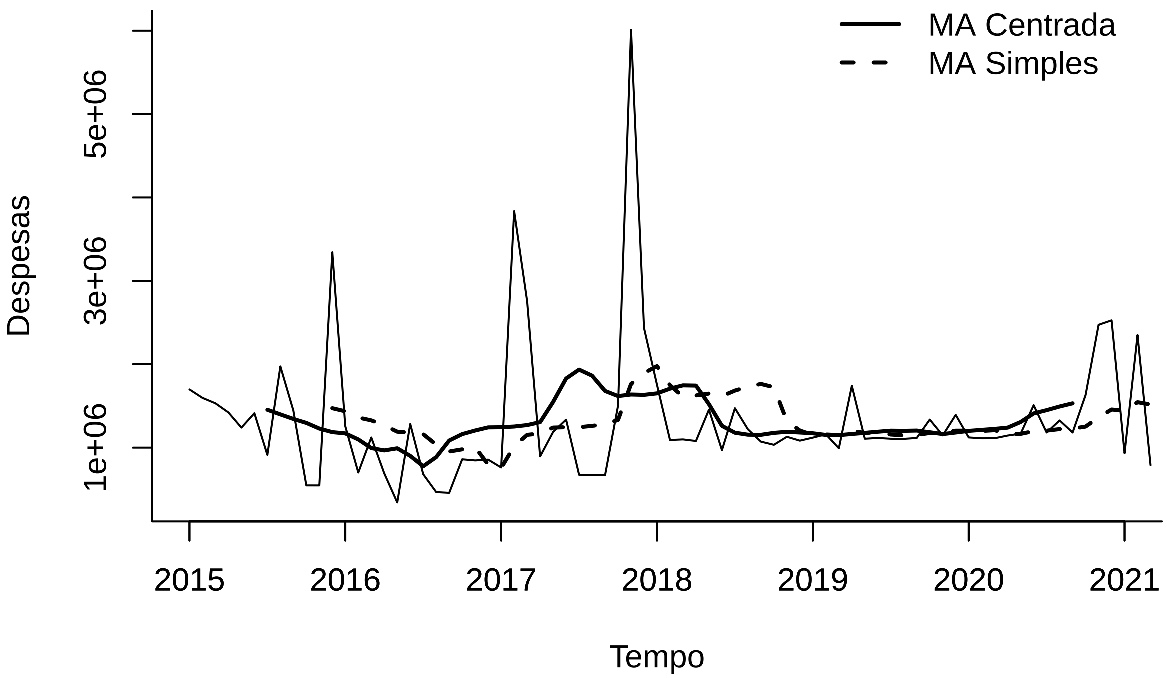
```
accuracy(modelo_tendencia_linear_sazonalidade_proj, validacao_ts)
```

```
##                         ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -2.954459e-11 777361.1 487897.5 -23.09932 43.41371 0.6139926
## Test set      3.395676e+05 543994.6 501845.1  13.43880 32.99902 0.6315449
##                    ACF1 Theil's U
## Training set  0.2261496        NA
## Test set     -0.4663291 0.6705322
```

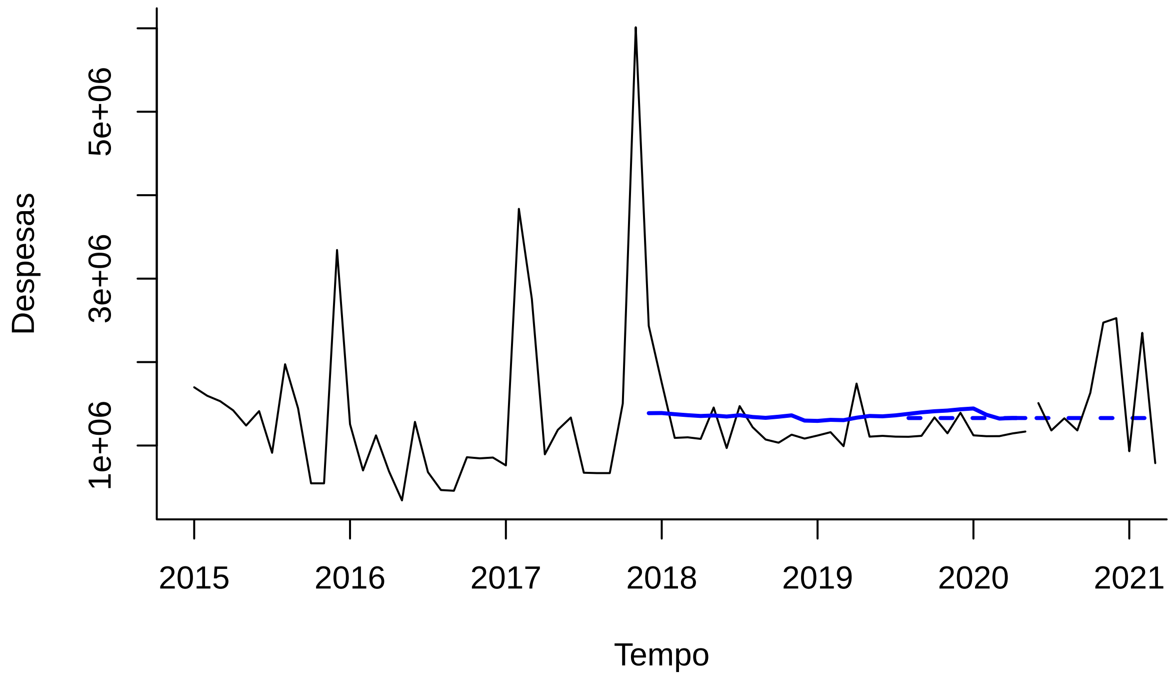
O Erro Médio Absoluto Percentual (MAPE) deste modelo foi superior ao do modelo Naive. Baseado neste dado, passaremos para um Modelo de Média móvel.

Modelo de Média Móvel

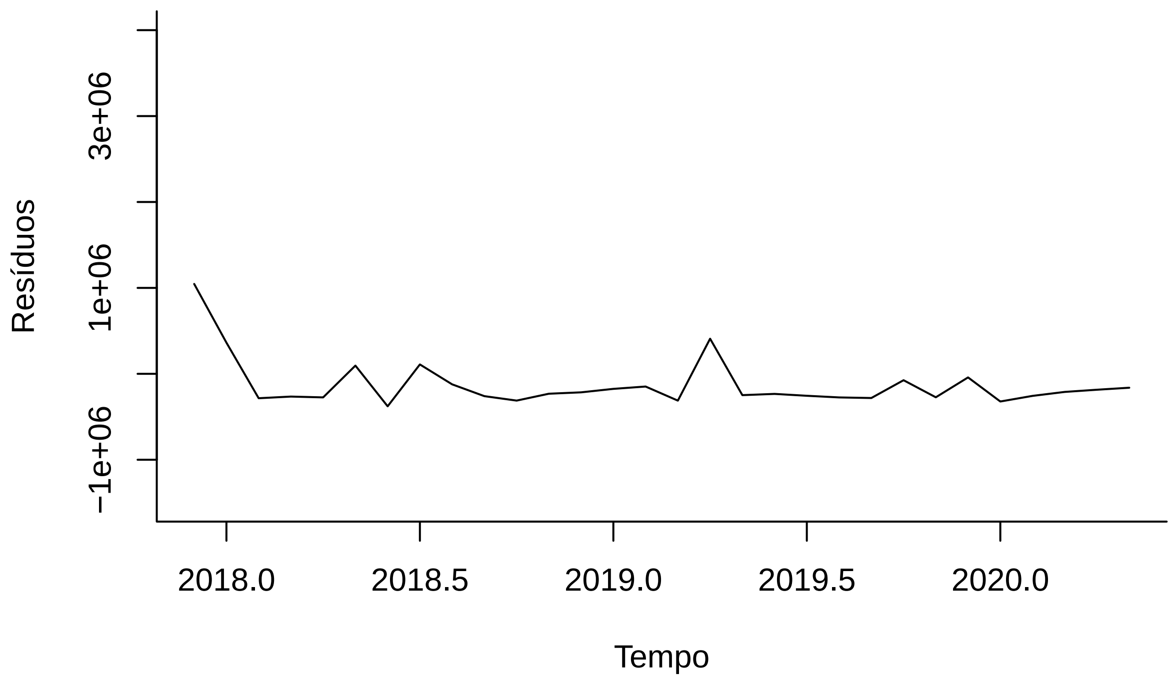
```
ma_simples <- rollmean(db_ts, k=12, align="right")
ma_centrada <- ma(db_ts, order=12)

plot(db_ts, ylim=c(342000, 6013000), ylab="Despesas", xlab="Tempo", bty="l", xaxt="s", xlim=c(2015,2021)
axis(1, at=seq(2015, 2021, 1), labels=format(seq(2015, 2021, 1)))
lines(ma_centrada, lwd=2)
lines(ma_simples, lwd=2, lty=2)
legend(2019,7000000, c("Despesas", "MA Centrada", "MA Simples"), lty=c(1,1,2), lwd=c(1,2,2), bty="n")
```



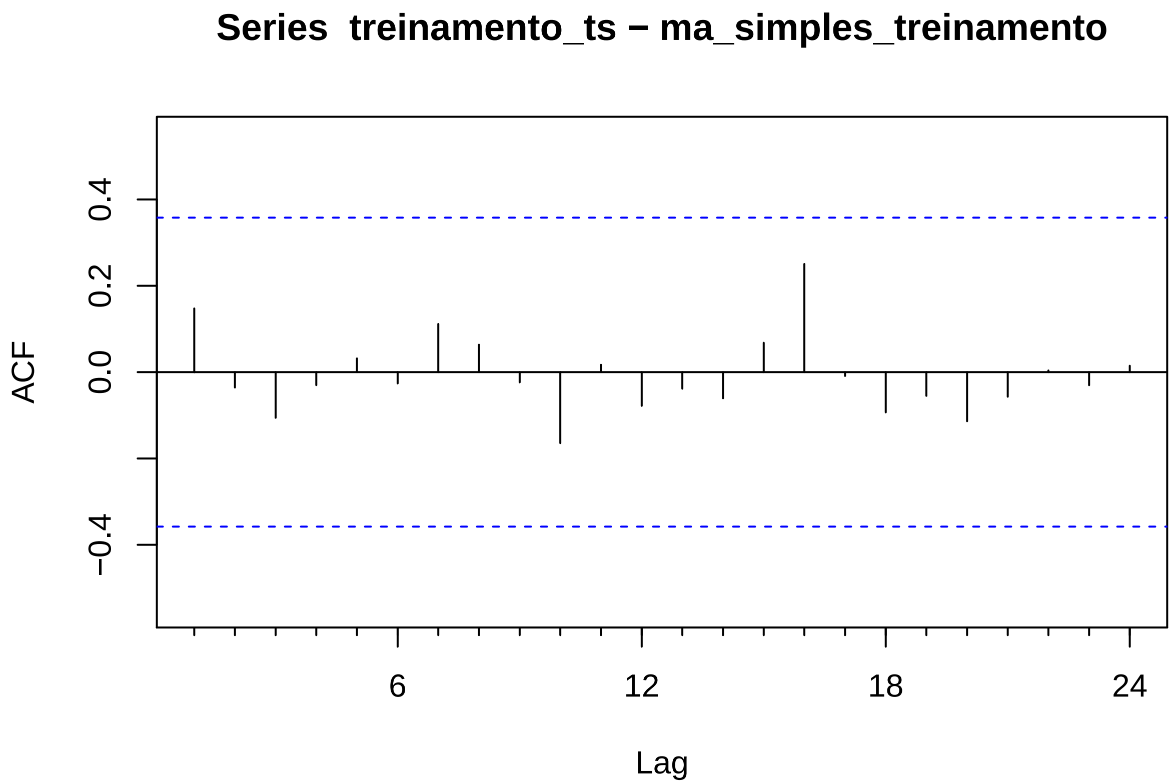

```
ma_simples_treinamento <- rollmean(treinamento_ts, k=36, align="right")
ultima_ma <- tail(ma_simples_treinamento, 1)

ma_simples_proj <- ts(rep(ultima_ma, 55), start=c(2015, 56), end = c(2015, 75), freq=12)

plot(treinamento_ts, ylim=c(342000, 6013000), ylab="Despesas", xlab="Tempo", bty="l", xaxt="n", xlim=c(
axis(1, at=seq(2015, 2021), labels=format(seq(2015, 2021)))
lines(ma_simples_treinamento, lwd=2, col="blue")
lines(ma_simples_proj, lwd=2, lty=2, col="blue")
lines(validacao_ts)
```




```
plot(treinamento_ts-ma_simples_treinamento, xlab="Tempo", ylab="Resíduos", ylim=c(-1500000, 4000000), b
```

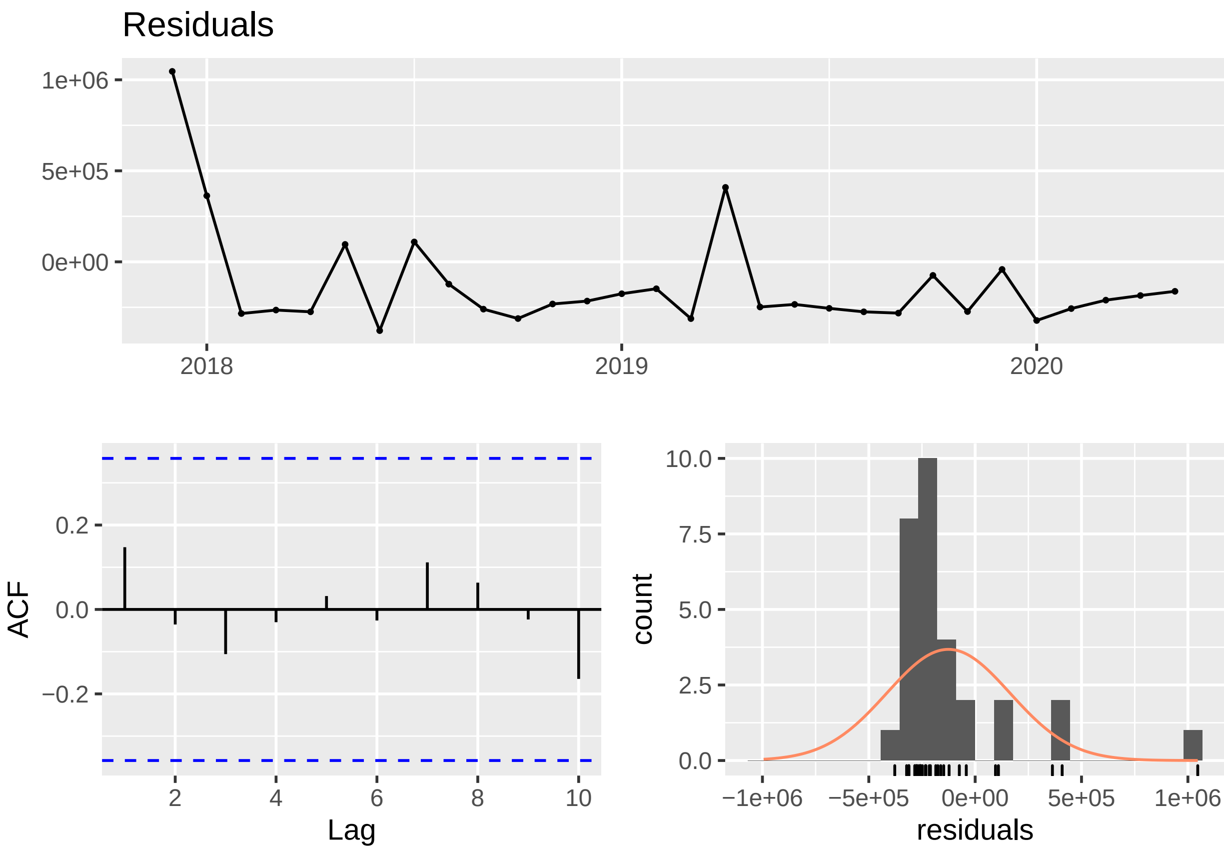



```
Acf(treinamento_ts-ma_simples_treinamento)
```




```
checkresiduals(treinamento_ts-ma_simples_treinamento, test="LB")
```

```
## Warning in modeldf.default(object): Could not find appropriate degrees of
## freedom for this model.
```




```
accuracy(ma_simples_treinamento, treinamento_ts)
```

```
##                   ME     RMSE      MAE       MPE    MAPE      ACF1 Theil's U
## Test set -125765.8 311234.6 260663.7 -14.25461 20.9992 0.1475033 0.8693382
```

```
accuracy(ma_simples_proj, validacao_ts)
```

```
##                ME     RMSE      MAE      MPE     MAPE       ACF1 Theil's U
## Test set 260499 663388.4 508141.3 3.092483 30.40187 -0.1692134 0.7576408
```

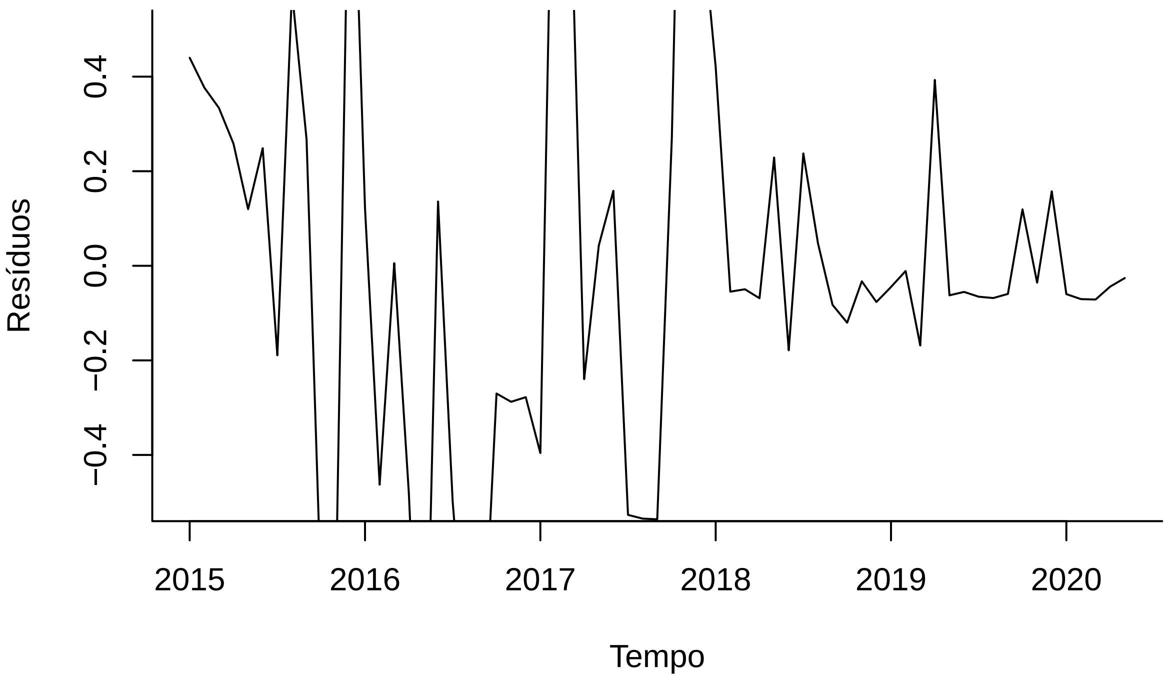
O MAPE do modelo de média móvel também foi superior ao MAPE do Modelo Naive. Modelo de tendência exponencial:

```
#Estima o modelo de tendência exp
modelo_tendencia_exp <- tslm(treinamento_ts ~ trend, lambda=0)

#resumo do modelo
summary(modelo_tendencia_exp)
```

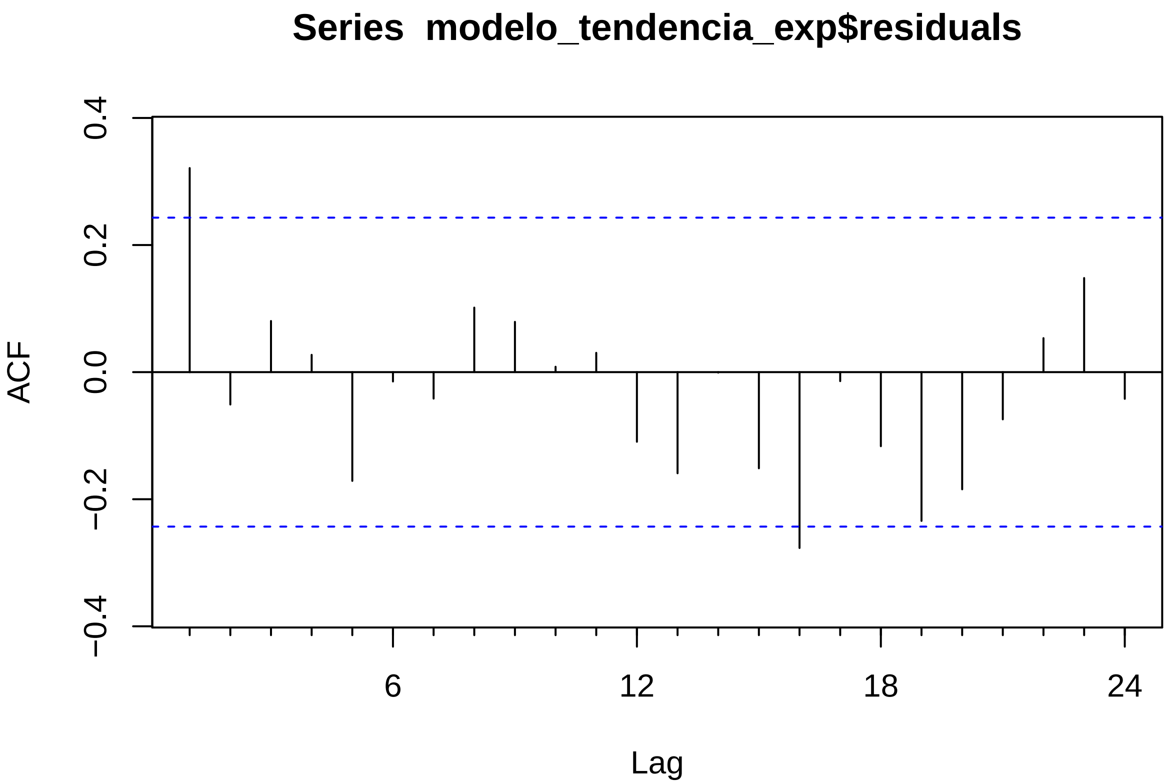
```
##
## Call:
## tslm(formula = treinamento_ts ~ trend, lambda = 0)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.18374 -0.18923 -0.04965  0.22895  1.65577
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) 13.903644   0.120598 115.289   <2e-16 ***
## trend        0.001426   0.003177   0.449    0.655
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4805 on 63 degrees of freedom
## Multiple R-squared:  0.003186,	Adjusted R-squared:  -0.01264
## F-statistic: 0.2014 on 1 and 63 DF,  p-value: 0.6552
```

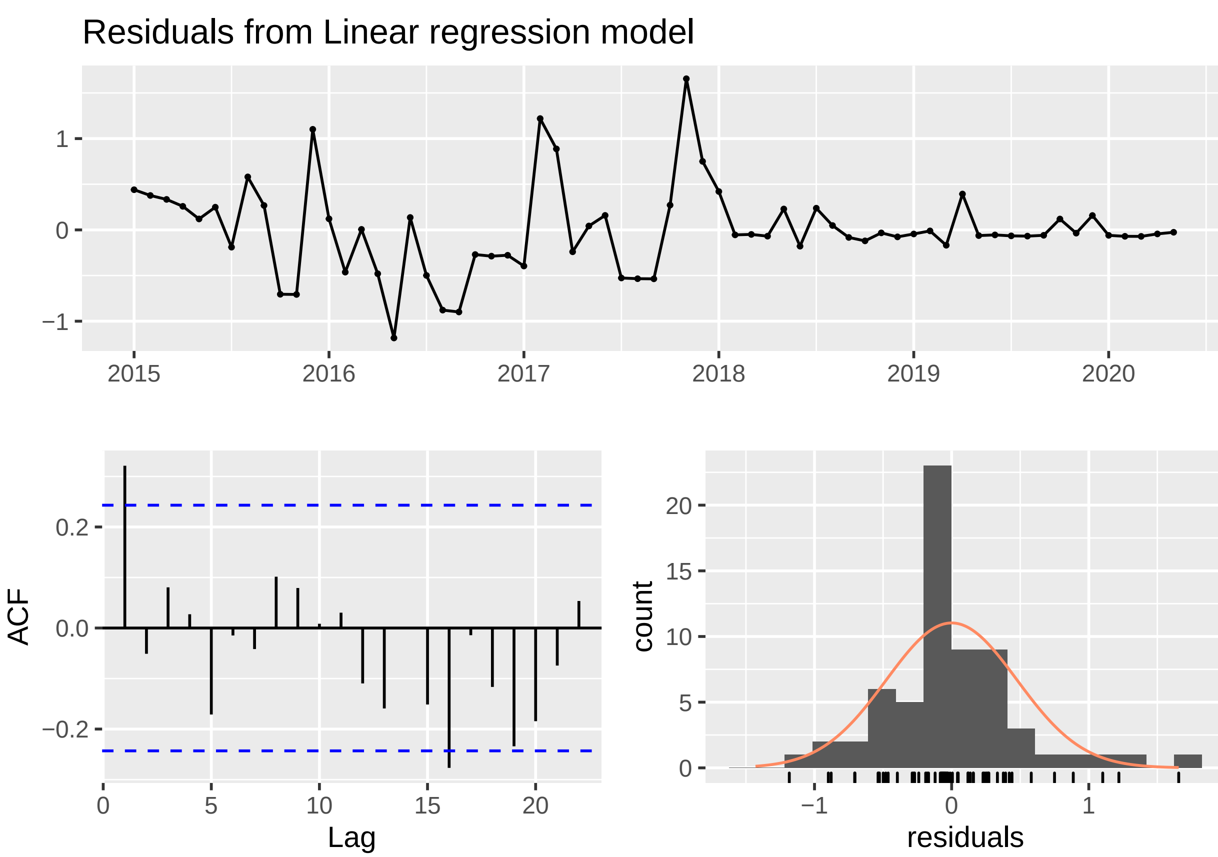
```
#Verificando resíduos

#Plotando os resíduos
plot(modelo_tendencia_exp$residuals, xlab="Tempo", ylab="Resíduos", ylim=c(-0.5, 0.5), bty="l")
```




```
#calcula a autocorrelação dos resíduos
Acf(modelo_tendencia_exp$residuals)
```




```
#verifica os resíduos com teste de Ljung-Box
checkresiduals(modelo_tendencia_exp, test="LB")
```



```
##
##  Ljung-Box test
##
## data:  Residuals from Linear regression model
## Q* = 14.446, df = 11, p-value = 0.2093
##
## Model df: 2.   Total lags used: 13
```

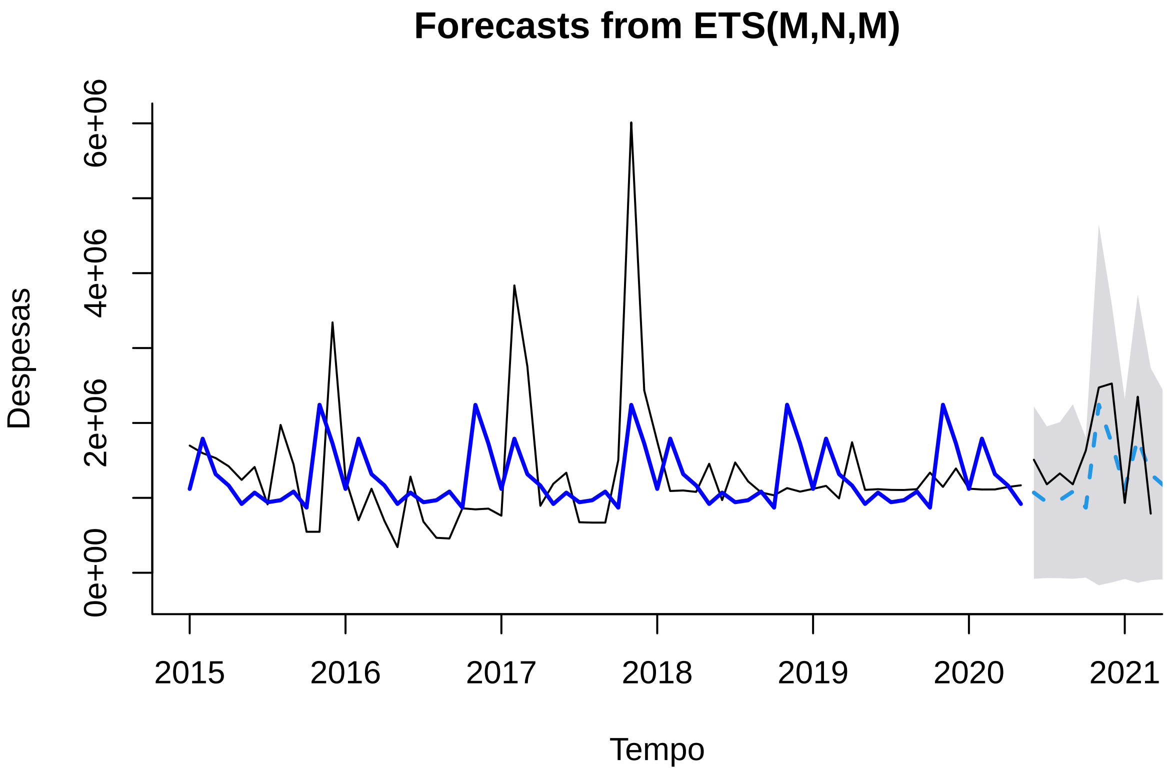
Novamente o MAPE foi pior do que o modelo Naïve.

Modelo de suavização exponencial (ZZZ)

```
modelo_ses1 <- ets(treinamento_ts, model = "ZZZ")
summary(modelo_ses1)
```

```
## ETS(M,N,M)
##
## Call:
##  ets(y = treinamento_ts, model = "ZZZ")
##
##   Smoothing parameters:
##     alpha = 1e-04
##     gamma = 1e-04
##
##   Initial states:
##     l = 1267423.279
##     s = 1.3613 1.769 0.686 0.8551 0.7642 0.7429
##            0.8444 0.7245 0.9191 1.0384 1.4123 0.8829
##
##   sigma:  0.5483
##
##      AIC     AICc      BIC
## 2028.626 2038.422 2061.242
##
## Training set error measures:
##                    ME     RMSE      MAE      MPE     MAPE     MASE      ACF1
## Training set 33208.8 794625.1 525621.3 -19.6763 45.41884 0.661466 0.1984845
```

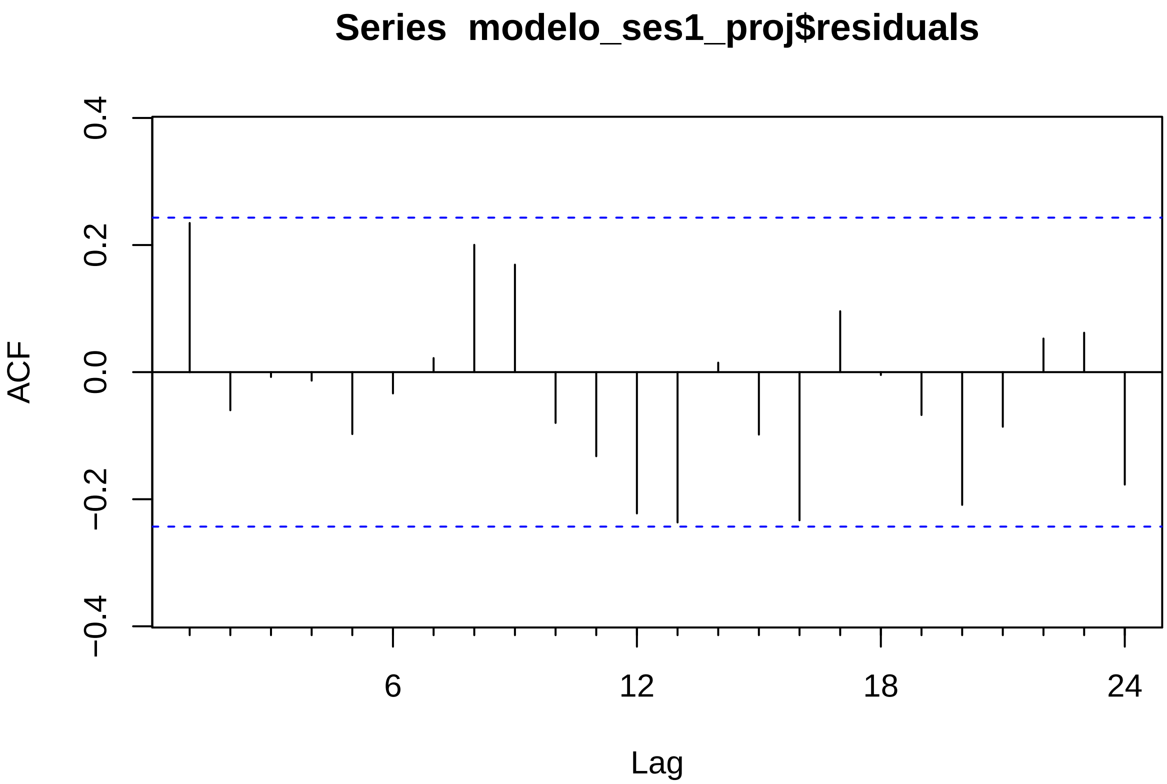
```
modelo_ses1_proj <- forecast(modelo_ses1, h=20, level=0.95)

plot(modelo_ses1_proj, ylim=c(-300000, 6013000), ylab="Despesas", xlab="Tempo", bty="l", xaxt="n", xlim=
axis(1, at=seq(2015, 2021), labels=format(seq(2015, 2021)))
lines(modelo_ses1$fitted, lwd=2, col="blue")
lines(validacao_ts)
```



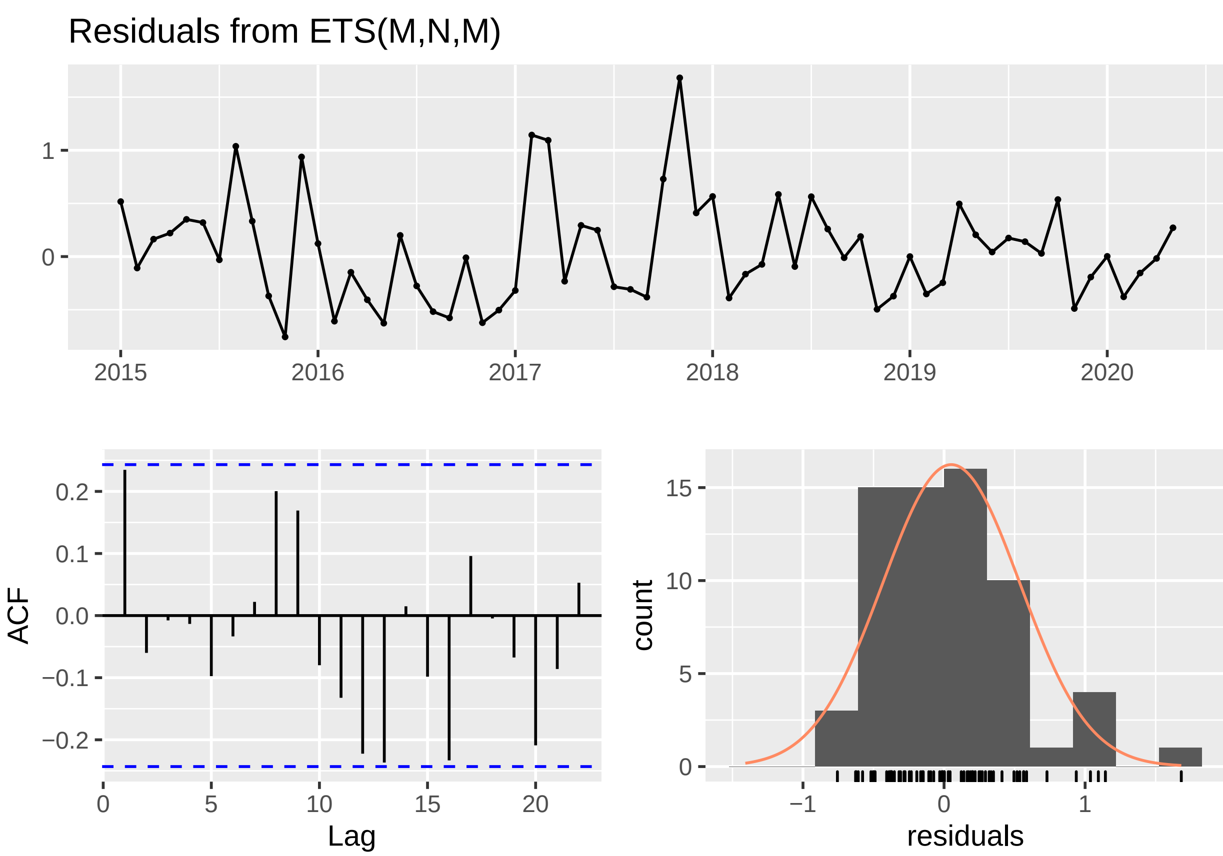

```
accuracy(modelo_ses1_proj, validacao_ts)
```

```
##                      ME     RMSE      MAE      MPE     MAPE      MASE       ACF1
## Training set  33208.8 794625.1 525621.3 -19.6763 45.41884 0.6614660  0.1984845
## Test set     276853.9 478243.0 419893.6  10.8923 28.29212 0.5284134 -0.4997656
##              Theil's U
## Training set        NA
## Test set     0.5443448
```

```
Acf(modelo_ses1_proj$residuals)
```




```r
checkresiduals(modelo_ses1_proj, test="LB")
```

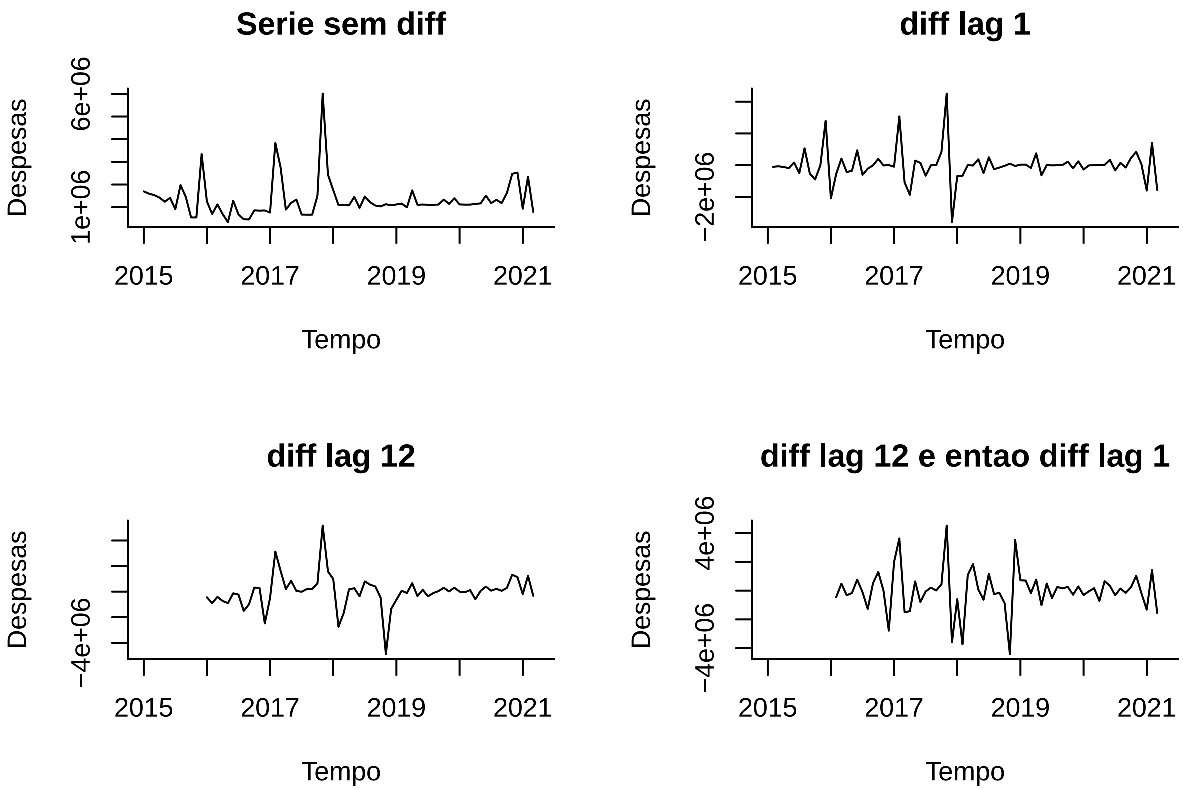


```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(M,N,M)
## Q* = 27.315, df = 3, p-value = 5.057e-06
##
## Model df: 14.   Total lags used: 17
```

Mais uma vez o MAPE foi superior ao modelo Naive, passaremos para um modelo ARIMA

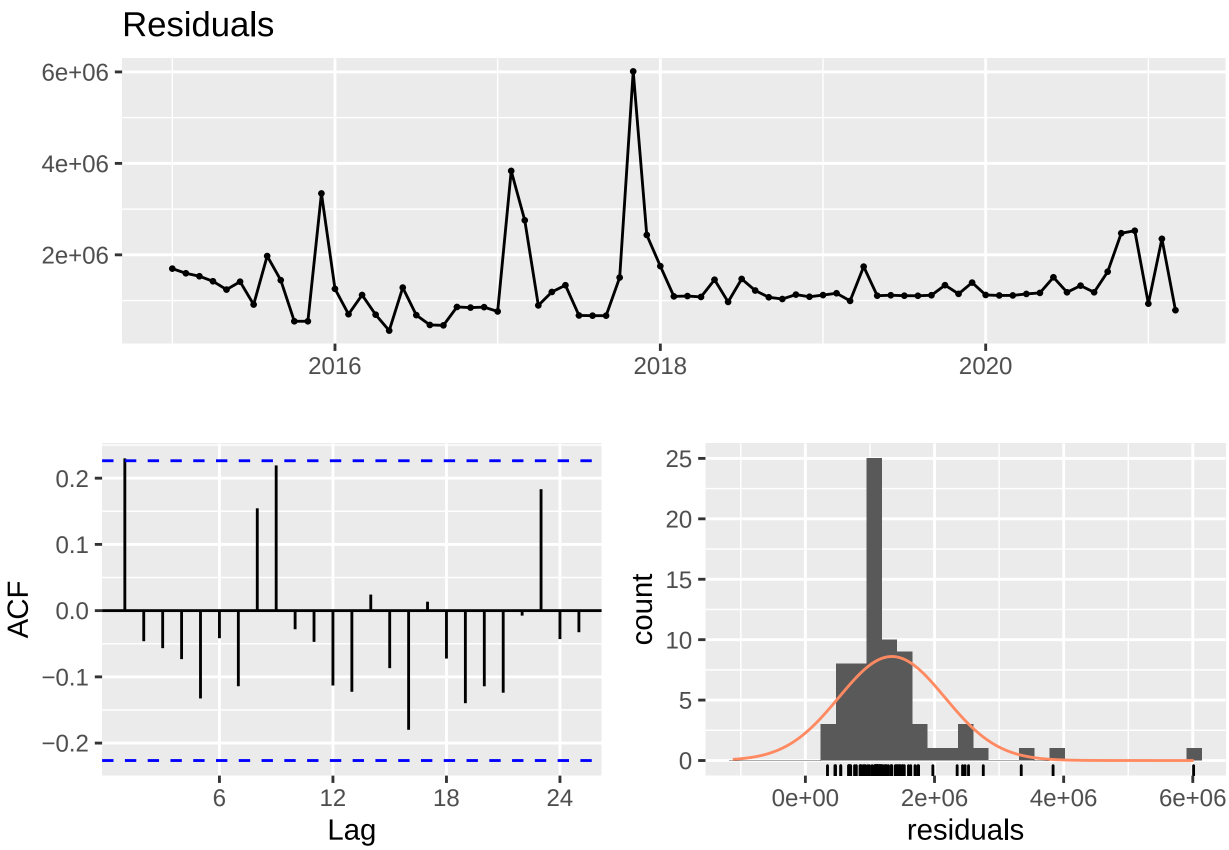
Modelo Arima:

```
#plota o grafica da projecao
par(mfrow=c(2,2))
plot(db_ts, ylab="Despesas", xlab="Tempo", bty="l", xlim=c(2015,2021.25), main=("Serie sem diff"))
plot(diff(db_ts, lag=1), ylab="Despesas", xlab="Tempo", bty="l", xlim=c(2015,2021.25), main=("diff lag 
plot(diff(db_ts, lag=12), ylab="Despesas", xlab="Tempo", bty="l", xlim=c(2015,2021.25), main=("diff lag
plot(diff(diff(db_ts, lag=12), lag=1), ylab="Despesas", xlab="Tempo", bty="l", xlim=c(2015,2021.25), ma
```



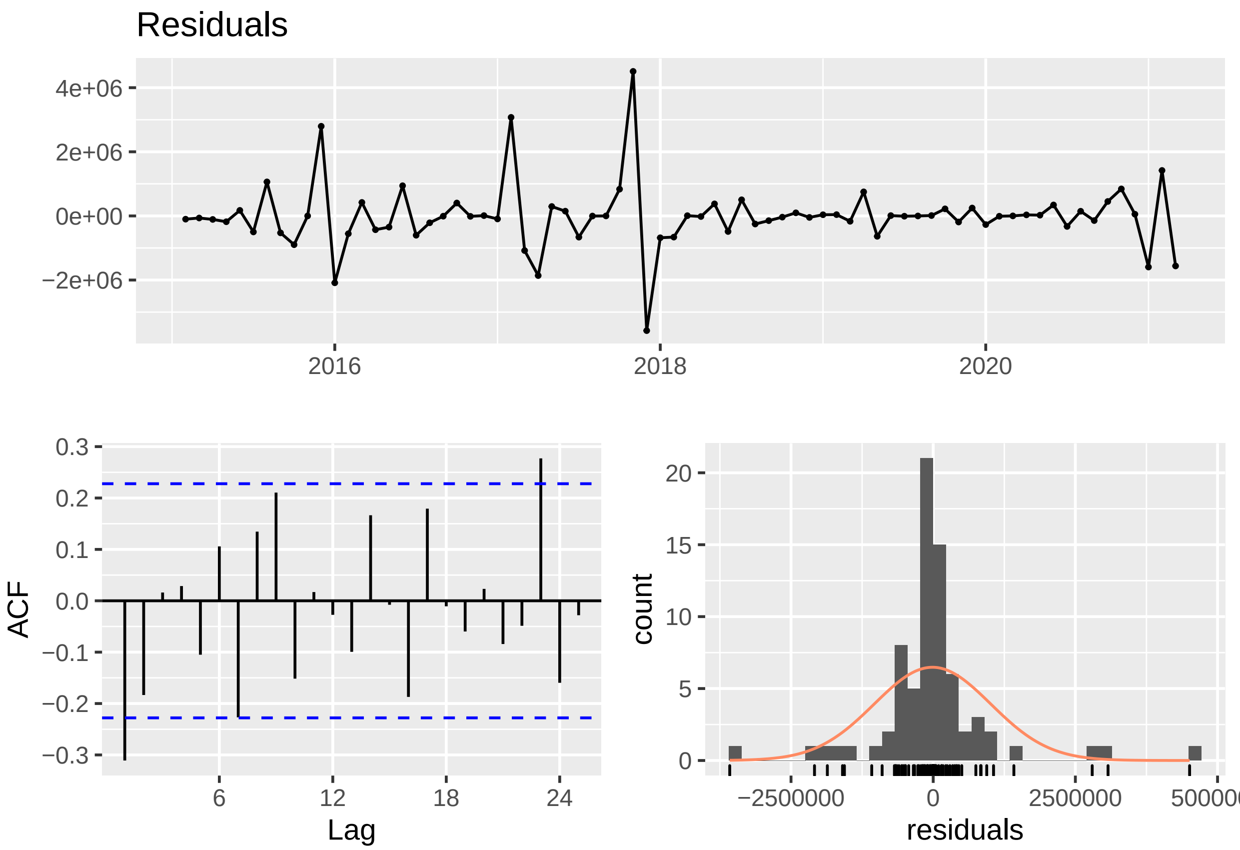

```
#checar estacionariedade
checkresiduals(db_ts)
```

```
## Warning in modeldf.default(object): Could not find appropriate degrees of
## freedom for this model.
```



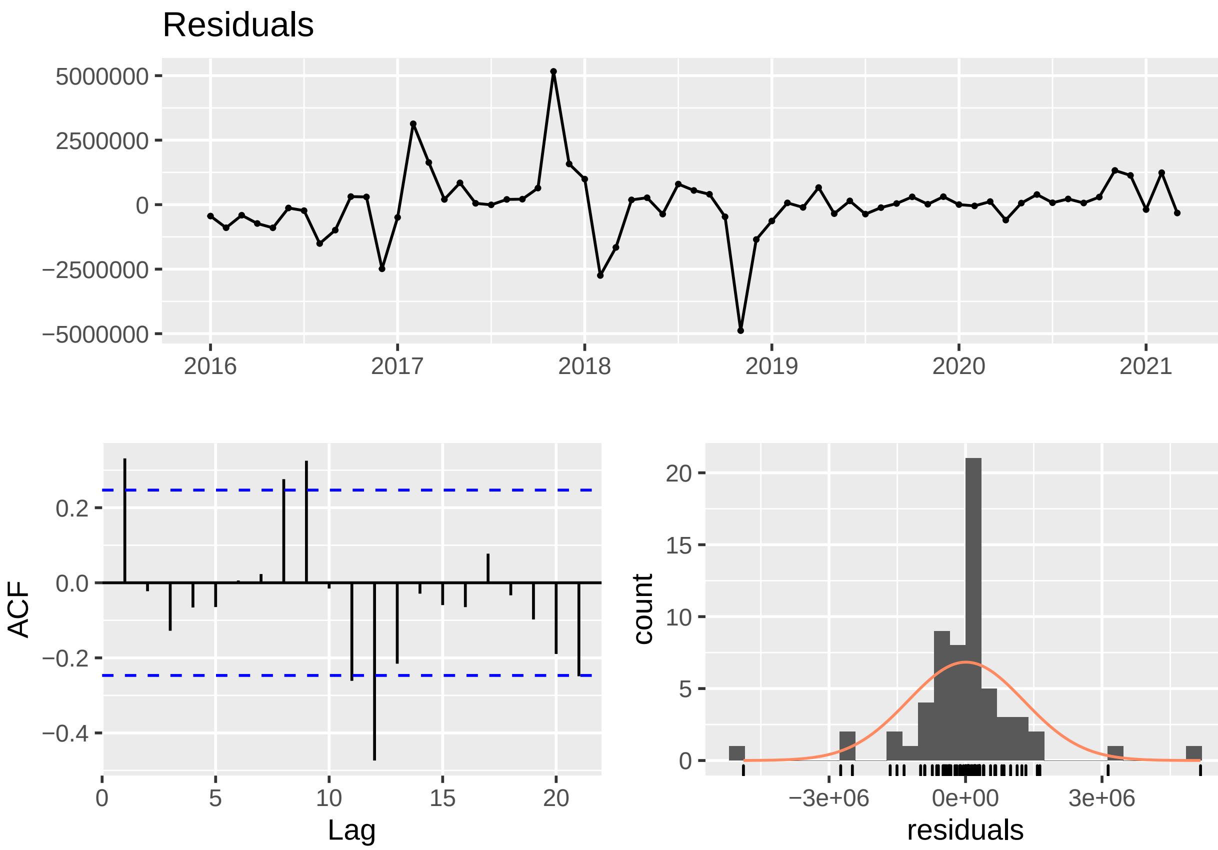

```
checkresiduals(diff(db_ts, lag=1))
```

```
## Warning in modeldf.default(object): Could not find appropriate degrees of
## freedom for this model.
```



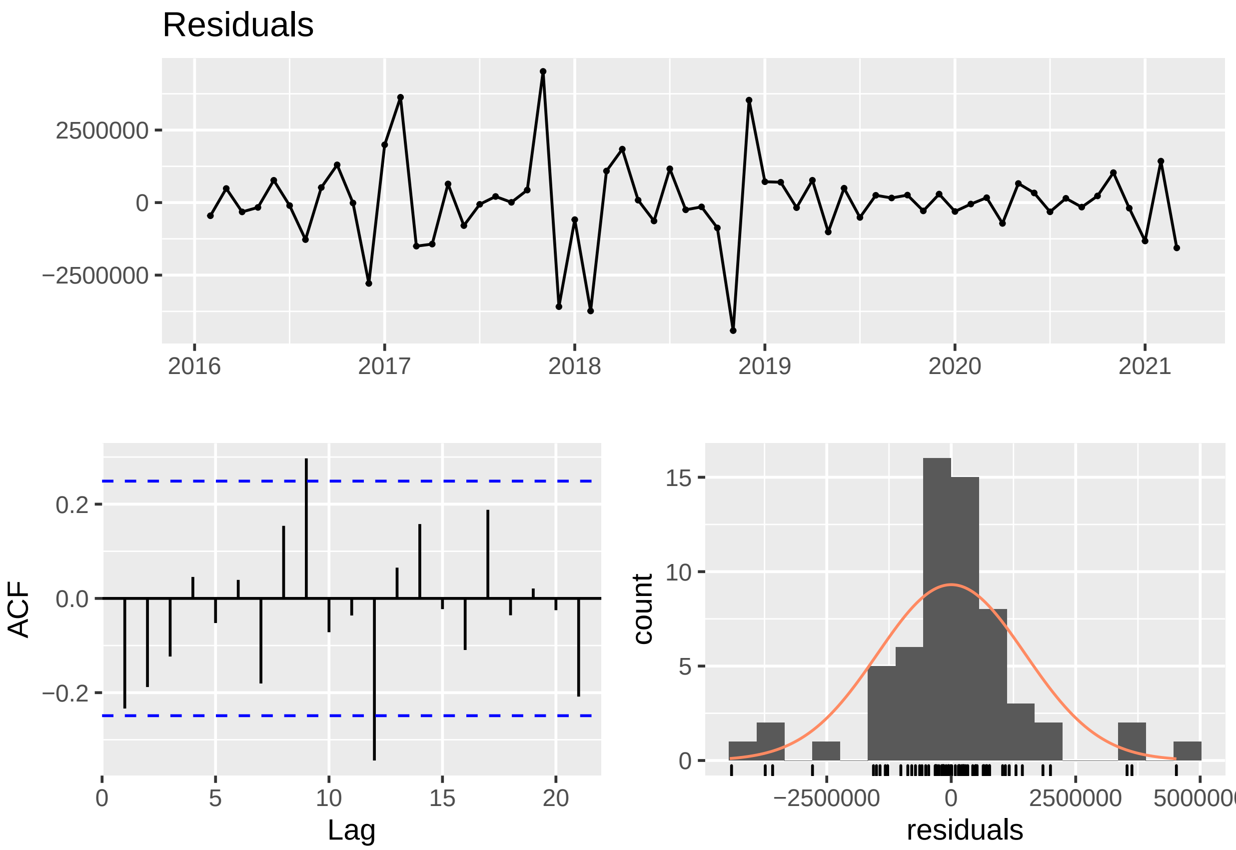

```
checkresiduals(diff(db_ts, lag=12))
```

```
## Warning in modeldf.default(object): Could not find appropriate degrees of
## freedom for this model.
```




```
checkresiduals(diff(diff(db_ts, lag=12), lag=1))
```

```
## Warning in modeldf.default(object): Could not find appropriate degrees of
## freedom for this model.
```




```
#diferencia 1 vez
db_ts_diff <- diff(db_ts, lag=1)

#executa o teste de KPSS
summary(ur.kpss(db_ts))
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 3 lags.
##
## Value of test-statistic is: 0.0621
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```
#executa o teste de KPSS
summary(ur.kpss(db_ts_diff))
```

```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 3 lags.
```

```
##
## Value of test-statistic is: 0.0283
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

```
#executa o teste de ADF
summary(ur.df(db_ts))
```

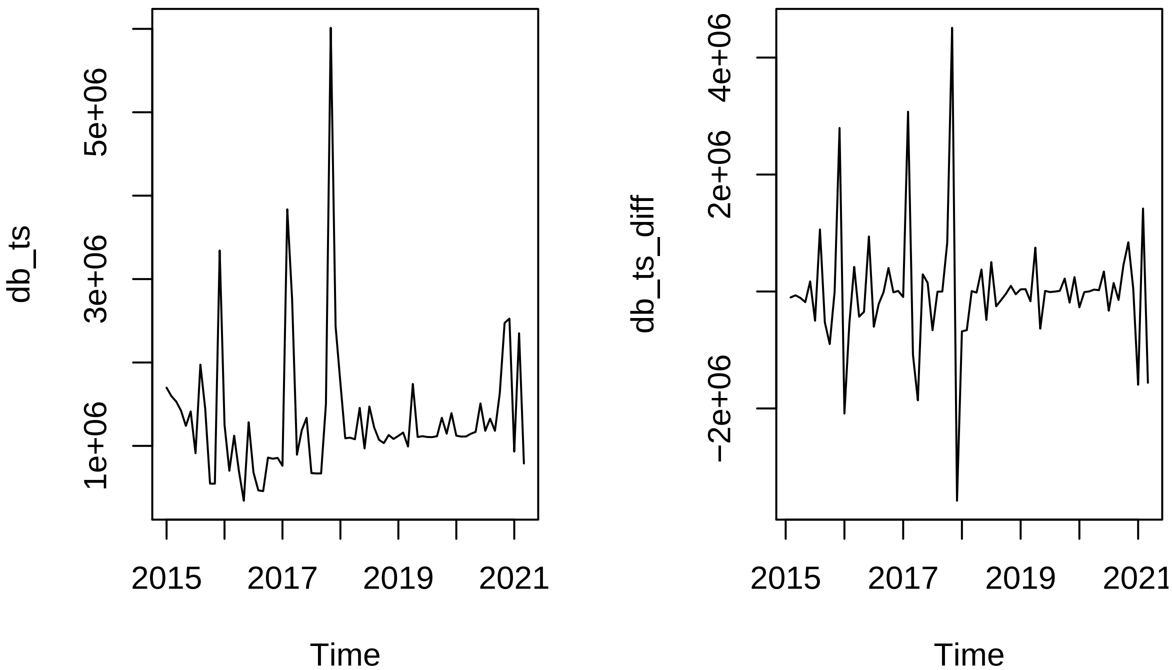
```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1654824   -47104   130667   392927  4955334
##
## Coefficients:
##            Estimate Std. Error t value Pr(>|t|)
## z.lag.1    -0.16690    0.07501  -2.225   0.0293 *
## z.diff.lag -0.23470    0.11726  -2.002   0.0491 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 951900 on 71 degrees of freedom
## Multiple R-squared:  0.1585, Adjusted R-squared:  0.1348
## F-statistic: 6.688 on 2 and 71 DF,  p-value: 0.002182
##
##
## Value of test-statistic is: -2.2251
##
## Critical values for test statistics:
##      1pct  5pct 10pct
## tau1 -2.6 -1.95 -1.61
```

```
#executa o teste de ADF
summary(ur.df(db_ts_diff))
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
```

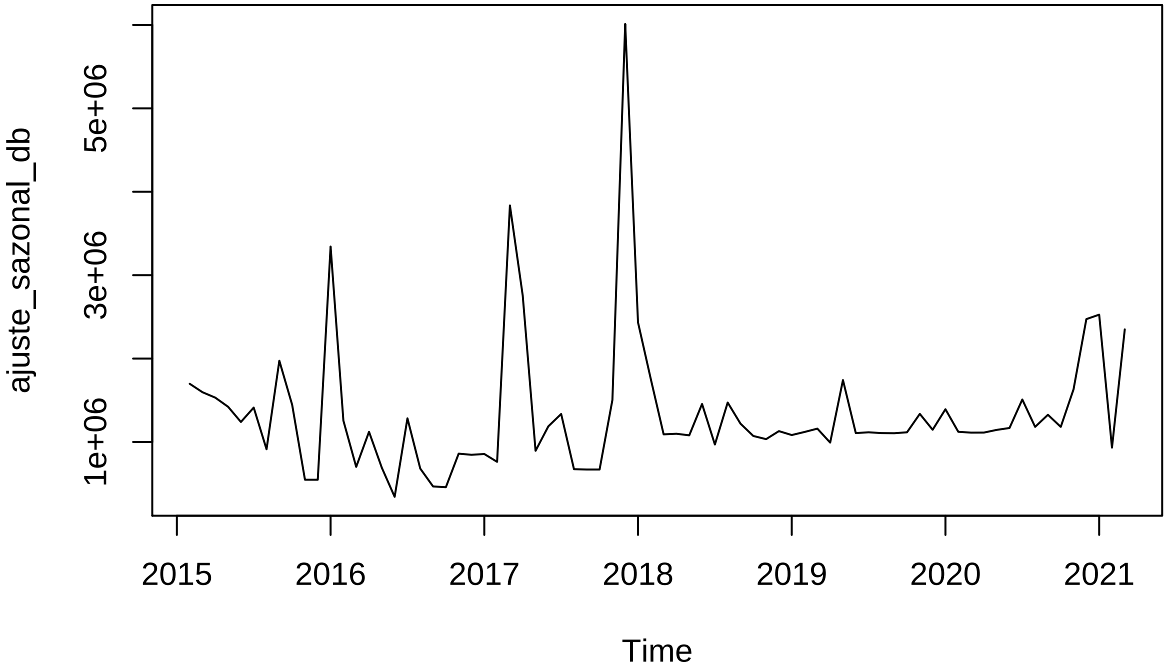
```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2083031  -346857  -117341   153056  4853907
##
## Coefficients:
##            Estimate Std. Error t value Pr(>|t|)
## z.lag.1     -1.7326     0.1858  -9.326 6.68e-14 ***
## z.diff.lag   0.3185     0.1162   2.741  0.00777 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 942200 on 70 degrees of freedom
## Multiple R-squared:  0.6862, Adjusted R-squared:  0.6772
## F-statistic: 76.53 on 2 and 70 DF,  p-value: < 2.2e-16
##
##
## Value of test-statistic is: -9.3264
##
## Critical values for test statistics:
##      1pct  5pct 10pct
## tau1 -2.6 -1.95 -1.61
```

```
##############################################################
# MODELO ARIMA
##############################################################

par(mfrow=c(1,2))
plot(db_ts)
plot(db_ts_diff)
```




```
par(mfrow=c(1,1))
```

```r
ajuste_sazonal_db <- db_ts-db_ts_diff

plot(ajuste_sazonal_db)
```




```r
#separa as amostras em treinamento e teste

#define o tamanho da amostra de teste
amostra_teste <- 20

#define o tamanho da amostra de treinamento
amostra_treino <- length(db_ts_diff) - amostra_teste



#cria a serie temporal de treinamento
treinamento_ts_diff <- window(db_ts_diff, start=c(2015, 1), end=c(2015, amostra_treino))
```

```
## Warning in window.default(x, ...): 'start' value not changed
```

```r
#cria a serie temporal de teste
validacao_ts_diff <- window(db_ts_diff, start=c(2015, amostra_treino + 1), end=c(2015,amostra_treino + a


#executa o teste de KPSS
summary(ur.kpss(treinamento_ts_diff))
```

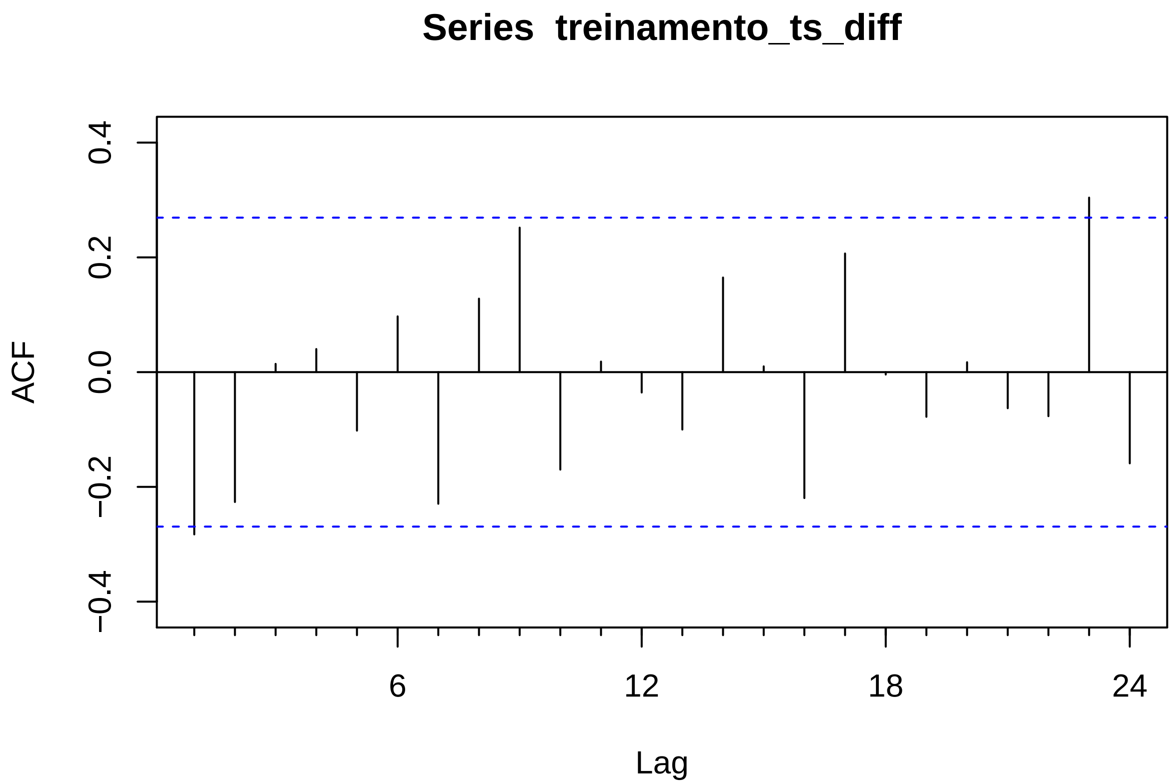
```
##
## #######################
## # KPSS Unit Root Test #
## #######################
##
## Test is of type: mu with 3 lags.
##
## Value of test-statistic is: 0.0368
```

```
##
## Critical value for a significance level of:
##                 10pct  5pct 2.5pct  1pct
## critical values 0.347 0.463  0.574 0.739
```

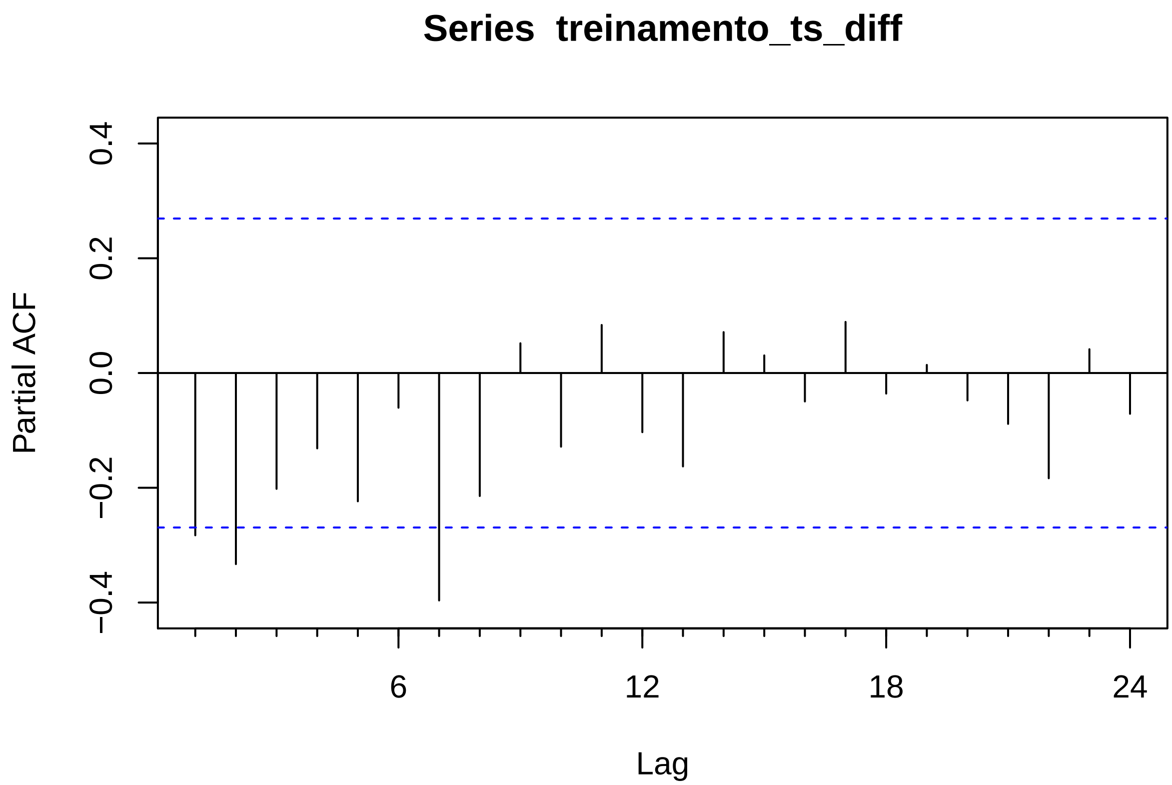
```
#executa o teste de ADF
summary(ur.df(treinamento_ts_diff))
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression none
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 - 1 + z.diff.lag)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2117586  -433296  -204615    59191  4823541
##
## Coefficients:
##            Estimate Std. Error t value Pr(>|t|)
## z.lag.1     -1.7127     0.2161  -7.926 2.48e-10 ***
## z.diff.lag   0.3351     0.1351   2.481   0.0166 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1065000 on 49 degrees of freedom
## Multiple R-squared:  0.6815, Adjusted R-squared:  0.6685
## F-statistic: 52.42 on 2 and 49 DF,  p-value: 6.713e-13
##
##
## Value of test-statistic is: -7.9263
##
## Critical values for test statistics:
##      1pct  5pct 10pct
## tau1 -2.6 -1.95 -1.61
```

```
#calcula a ACF
Acf(treinamento_ts_diff)
```




```
#calcula a PCF
Pacf(treinamento_ts_diff)
```

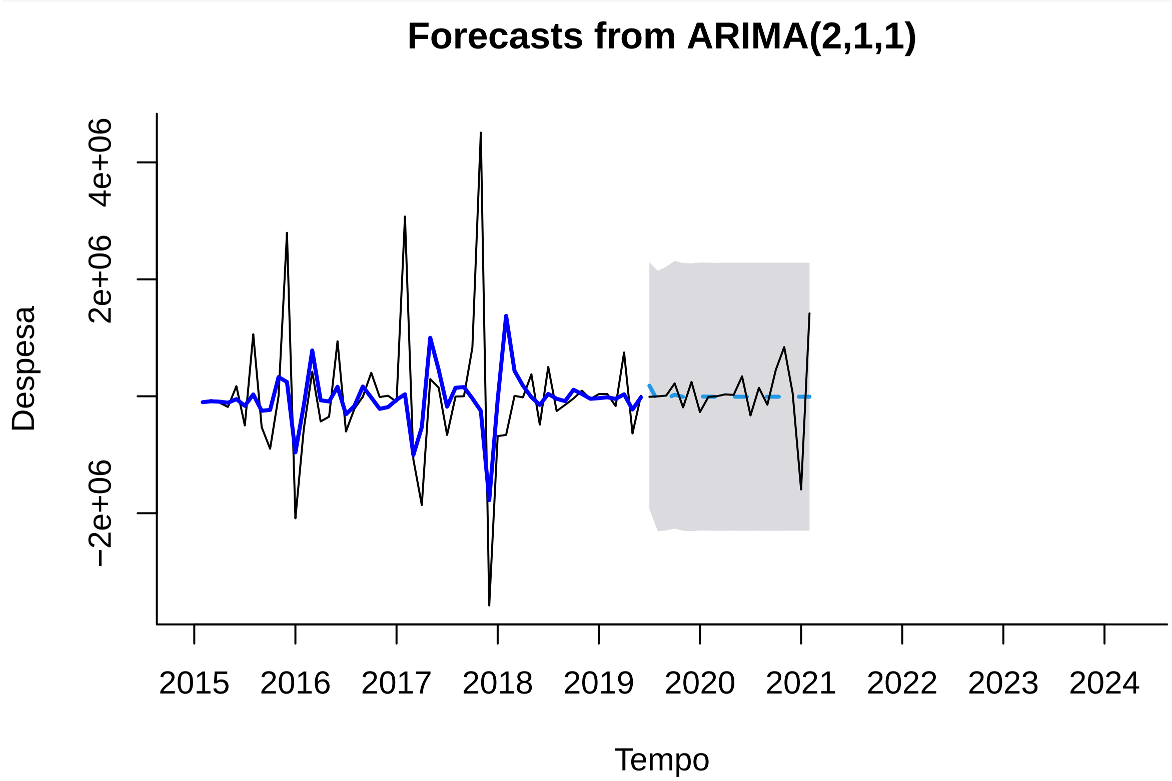



```
#Modelo Arima
Modelo_ARIMA <- Arima(treinamento_ts_diff, order = c(2,1,1))
```

```r
#resumo modelo
summary(Modelo_ARIMA)
```

```
## Series: treinamento_ts_diff
## ARIMA(2,1,1)
##
## Coefficients:
##           ar1      ar2     ma1
##       -0.3578  -0.3104  -1.000
## s.e.   0.1307   0.1290   0.051
##
## sigma^2 estimated as 1.136e+12:  log likelihood=-796.59
## AIC=1601.18   AICc=1602.04   BIC=1608.99
##
## Training set error measures:
##                   ME    RMSE      MAE      MPE     MAPE      MASE        ACF1
## Training set 28465.6 1024914 570511.2 6371.105 6675.536 0.4752428 -0.08119995
```

```r
#projeta os proximos 12 meses
modelo_ARIMA_proj <- forecast(Modelo_ARIMA, h=amostra_teste, level=0.95)
#plota o grafica da projecao
plot(modelo_ARIMA_proj, ylab="Despesa", xlab="Tempo", bty="l", xaxt="n", xlim=c(2015,2024.25), flty=2)

axis(1, at=seq(2015, 2024, 1), labels=format(seq(2015, 2024, 1)))

lines(Modelo_ARIMA$fitted, lwd=2, col="blue")

lines(validacao_ts_diff)
```



```r
#verifica precisao
accuracy(modelo_ARIMA_proj, validacao_ts_diff)
```

```
##                    ME      RMSE      MAE        MPE      MAPE      MASE
## Training set 28465.60 1024914.1 570511.2 6371.10531 6675.5360 0.4752428
## Test set     64217.44  548342.3 332683.2   17.39917  547.5184 0.2771291
##                     ACF1 Theil's U
## Training set -0.08119995        NA
## Test set     -0.38972754  1.276779
```

```r
#função auto.arima
auto.arima(treinamento_ts_diff, seasonal = FALSE, stepwise=FALSE, approximation = FALSE)
```

```
## Series: treinamento_ts_diff
## ARIMA(2,0,0) with zero mean
##
## Coefficients:
##           ar1      ar2
##       -0.3706  -0.3230
## s.e.   0.1288   0.1271
##
## sigma^2 estimated as 1.091e+12:  log likelihood=-808.87
## AIC=1623.73   AICc=1624.22   BIC=1629.64
```

O modelo ARIMA teve um desempenho pior entre os modelos apresentados. Conclusão: