Questão 2

Heron, Alexandre, Shai, Pedrosa, Roberto

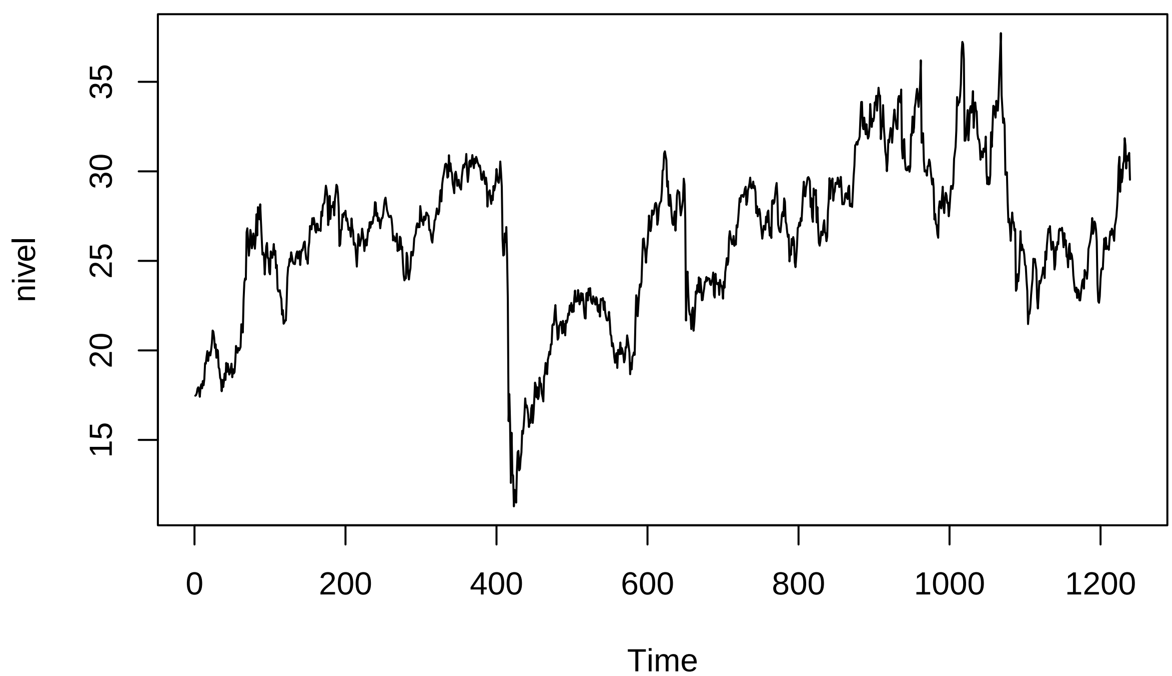
2023-08-31

Questão 2

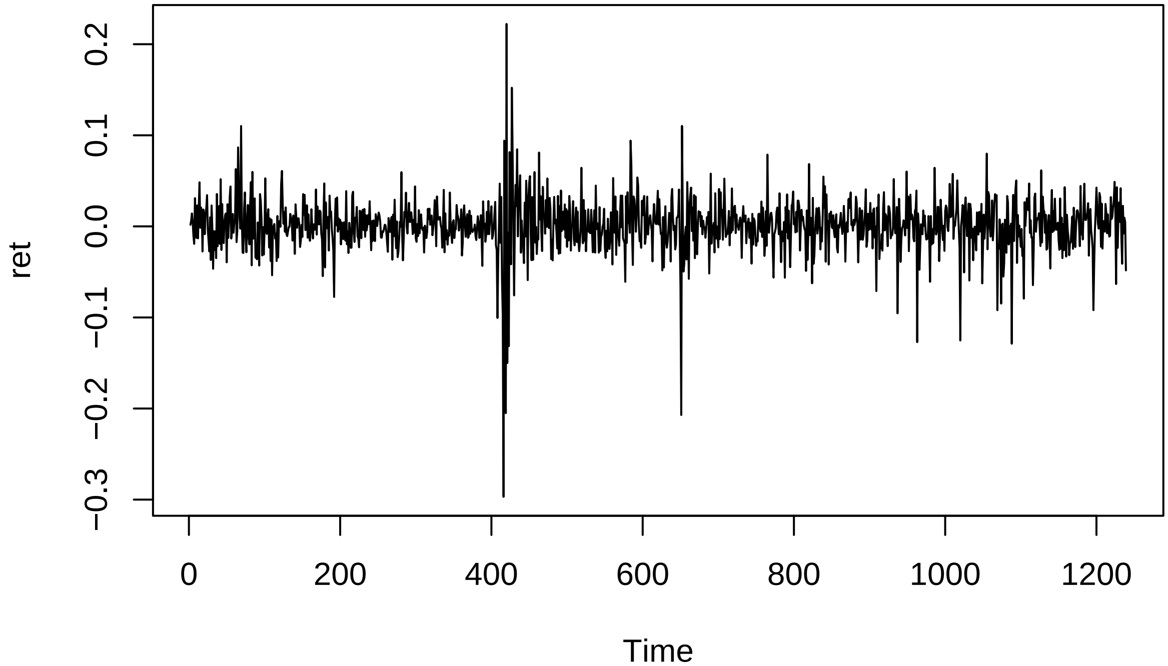
Importando a base de dados

```
petr4 <- read_xlsx("./PETR4.xlsx")
ret <- ts(petr4$`Var%`)
ret <- na.omit(ret)
nivel <- ts(petr4$Coluna1)
```

```
plot(nivel)
```

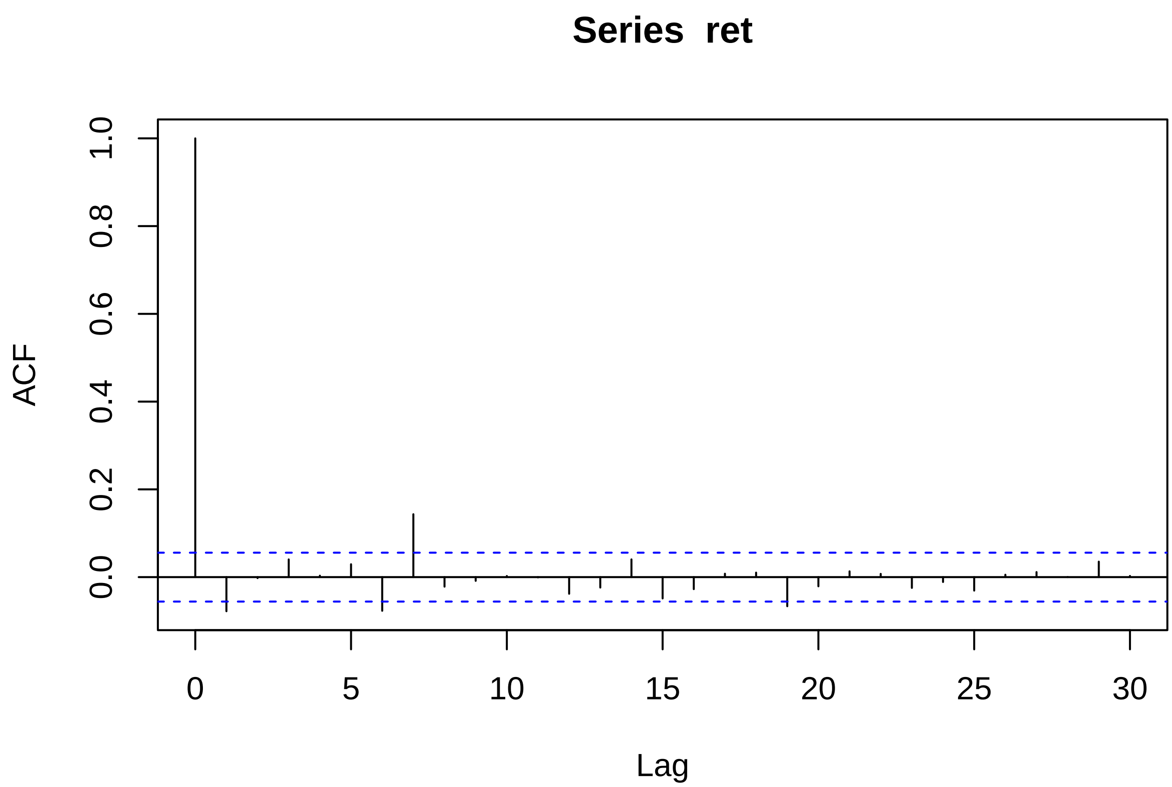


```r
plot(ret)
```

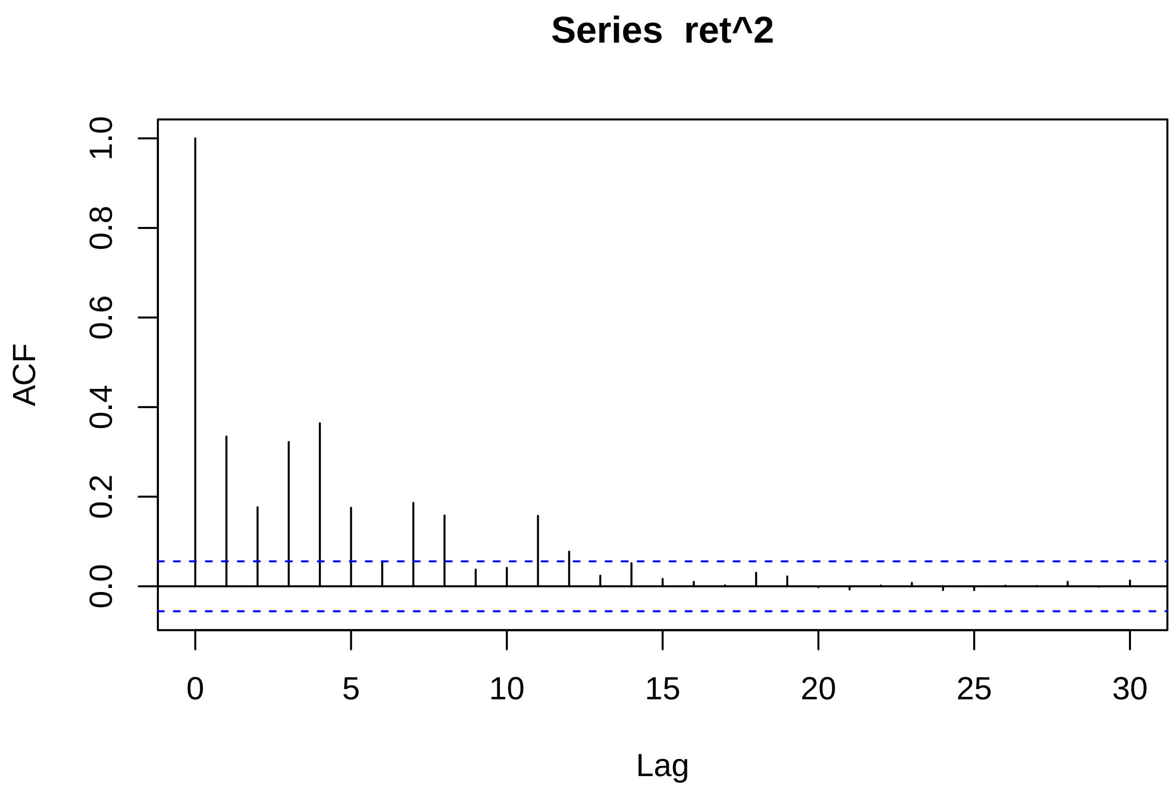


Por que utilizar o modelo GARCH(p,q)?

```r
acf(ret)
```



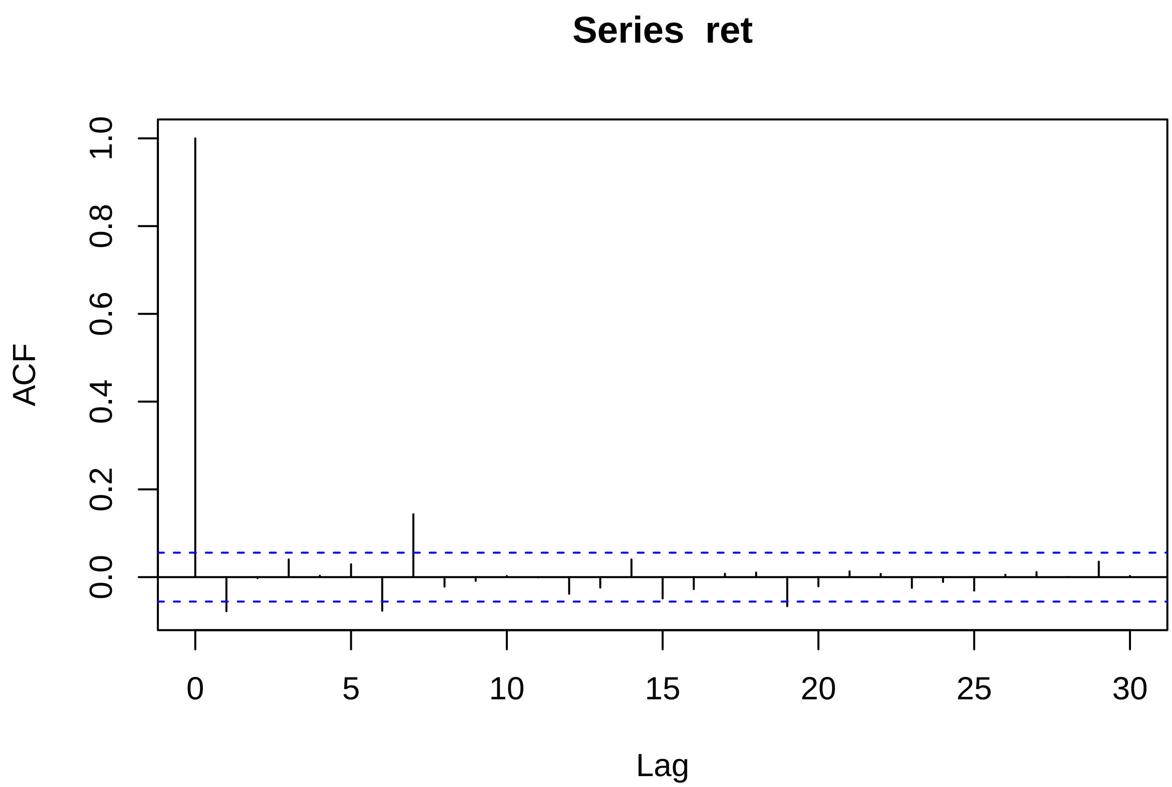

```r
acf(ret^2)
```



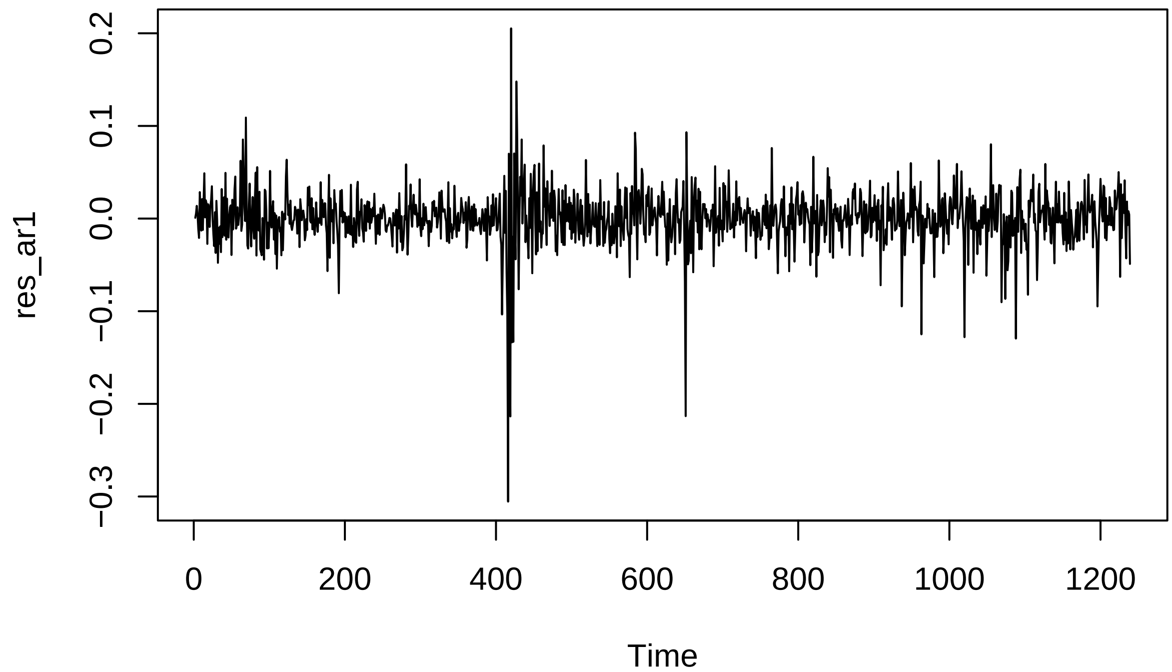
AR(1)

```
# Modelando os retornos com um AR(1)
ret_ar1 <- arima(ret, order = c(1,0,0), include.mean = TRUE, method = "ML")
```

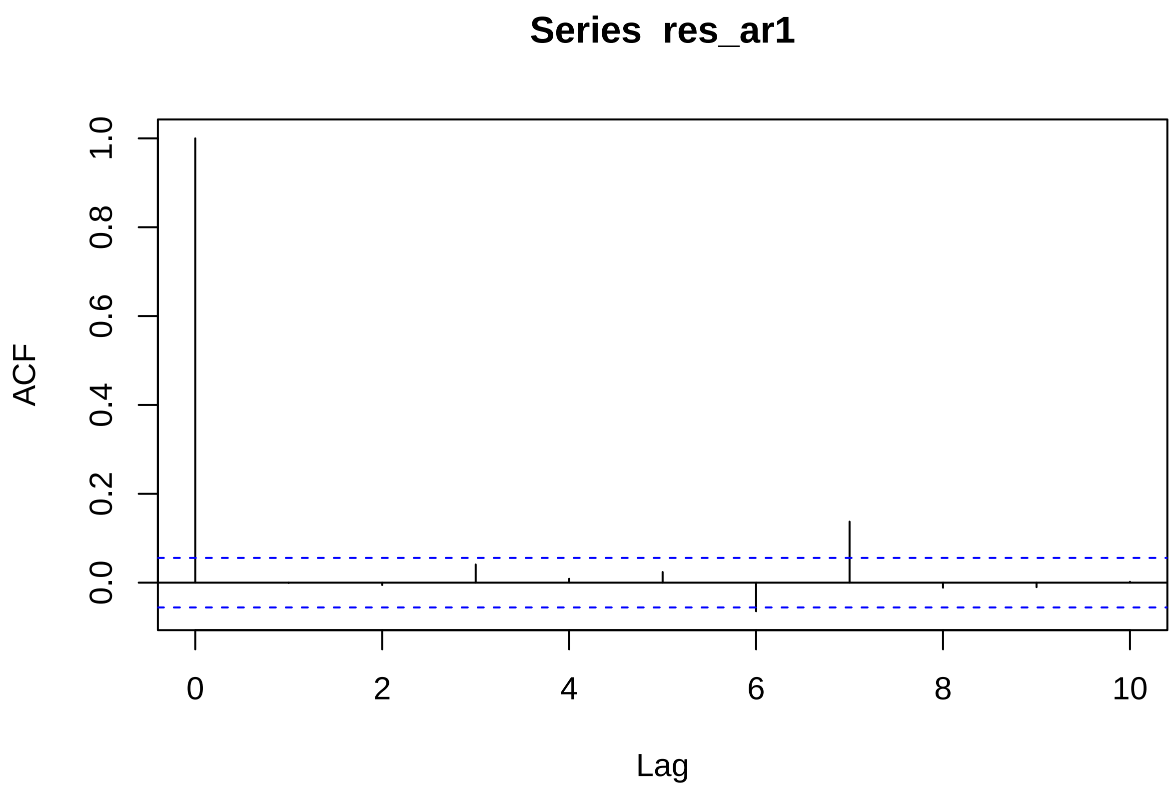
```
acf(ret)
```



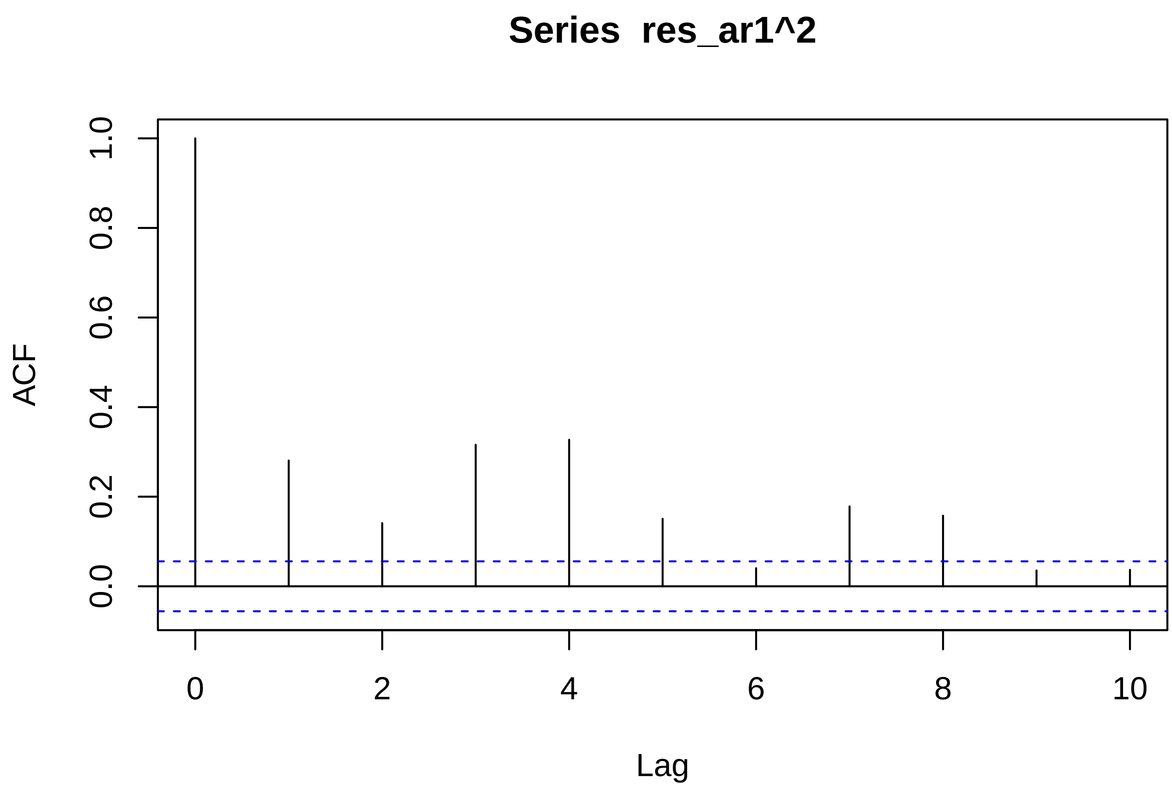

```r
# Coletando os resíduos
res_ar1 <- ret_ar1[["residuals"]]
plot(res_ar1)
```




```r
# Autocorrelação serial dos resíduos
res_ar1 <- na.omit(res_ar1)
acf(res_ar1, lag.max = 10)
```




```
acf(res_ar1^2, lag.max = 10)
```



Parte ARMA

```
# Valores possíveis para "p" e "q"
p_values <- 1:5
q_values <- 1:5
```

```
# Inicializar vetores para armazenar valores de AIC e BIC
aic_values <- matrix(NA,
                     nrow = length(p_values),
                     ncol = length(q_values))
bic_values <- matrix(NA,
                     nrow = length(p_values),
                     ncol = length(q_values))
aicc_values <- matrix(NA,
                     nrow = length(p_values),
                     ncol = length(q_values))
```

```
# Loop para calcular AIC e BIC para diferentes combinações de p e q

for (i in p_values){
  for (j in q_values){

    modelo <- Arima(
```

```r
      ret,
      order = c(i, 0, j),
      method = "ML"
      )

    aic_values[i,j] <- AIC(modelo)
    bic_values[i,j] <- BIC(modelo)
    aicc_values[i,j] <- modelo$aicc

    }
  }
```

```r
# Encontrar as posições mínimas de AIC e BIC
min_aic_pos <- which(aic_values == min(aic_values), arr.ind = TRUE)
min_bic_pos <- which(bic_values == min(bic_values), arr.ind = TRUE)
min_aicc_pos <- which(aicc_values == min(aicc_values), arr.ind = TRUE)

print(min_aic_pos)
```

```
##      row col
## [1,]   5   5
```

```r
print(min_bic_pos)
```

```
##      row col
## [1,]   1   1
```

```r
print(min_aicc_pos)
```

```
##      row col
## [1,]   5   5
```

Realizaremos, agora os dois modelos encontrados.

```r
summary(
  Arima(
      ret,
      order = c(min_aic_pos[1,1],
                0,
                min_aic_pos[1,2]),
      method = "ML"
      )
)
```

AIC Minimizador:

```
## Series: ret
## ARIMA(5,0,5) with non-zero mean
```

```
##
## Coefficients:
##           ar1     ar2      ar3     ar4     ar5     ma1      ma2     ma3
##       -0.2372  0.2474  -0.2634  0.4446  0.7223  0.1605  -0.2703  0.3219
## s.e.   0.1041  0.0757   0.0589  0.0568  0.0779  0.1117   0.0800  0.0551
##           ma4      ma5   mean
##       -0.5106  -0.7015  7e-04
## s.e.   0.0578   0.0910  1e-04
##
## sigma^2 = 0.0008799:  log likelihood = 2602.6
## AIC=-5181.2   AICc=-5180.95   BIC=-5119.75
##
## Training set error measures:
##                        ME       RMSE        MAE MPE MAPE      MASE        ACF1
## Training set 0.0004989178 0.02953167 0.01983502 NaN  Inf 0.6707098 0.001037128
```

```
summary(
  Arima(
      ret,
      order = c(min_bic_pos[1,1],
                0,
                min_bic_pos[1,2]),
      method = "ML"
      )
)
```

BIC Minimizador:

```
## Series: ret
## ARIMA(1,0,1) with non-zero mean
##
## Coefficients:
##           ar1      ma1   mean
##       -0.0393  -0.0391  9e-04
## s.e.   0.2537   0.2534  8e-04
##
## sigma^2 = 0.0008996:  log likelihood = 2586.25
## AIC=-5164.49   AICc=-5164.46   BIC=-5144.01
##
## Training set error measures:
##                        ME       RMSE        MAE MPE MAPE      MASE
## Training set 2.359824e-06 0.02995696 0.01991963 NaN  Inf 0.6735711
##                       ACF1
## Training set -8.546386e-05
```

```
summary(
  Arima(
      ret,
```

```r
      order = c(min_aicc_pos[1,1],
                0,
                min_aicc_pos[1,2]),
      method = "ML"
      )
)
```

AICc Minimizador

```
## Series: ret
## ARIMA(5,0,5) with non-zero mean
##
## Coefficients:
##           ar1     ar2      ar3     ar4     ar5     ma1      ma2     ma3
##       -0.2372  0.2474  -0.2634  0.4446  0.7223  0.1605  -0.2703  0.3219
## s.e.   0.1041  0.0757   0.0589  0.0568  0.0779  0.1117   0.0800  0.0551
##           ma4      ma5    mean
##       -0.5106  -0.7015   7e-04
## s.e.   0.0578   0.0910   1e-04
##
## sigma^2 = 0.0008799:  log likelihood = 2602.6
## AIC=-5181.2   AICc=-5180.95   BIC=-5119.75
##
## Training set error measures:
##                        ME       RMSE        MAE MPE MAPE      MASE        ACF1
## Training set 0.0004989178 0.02953167 0.01983502 NaN  Inf 0.6707098 0.001037128
```

Parcimônioso

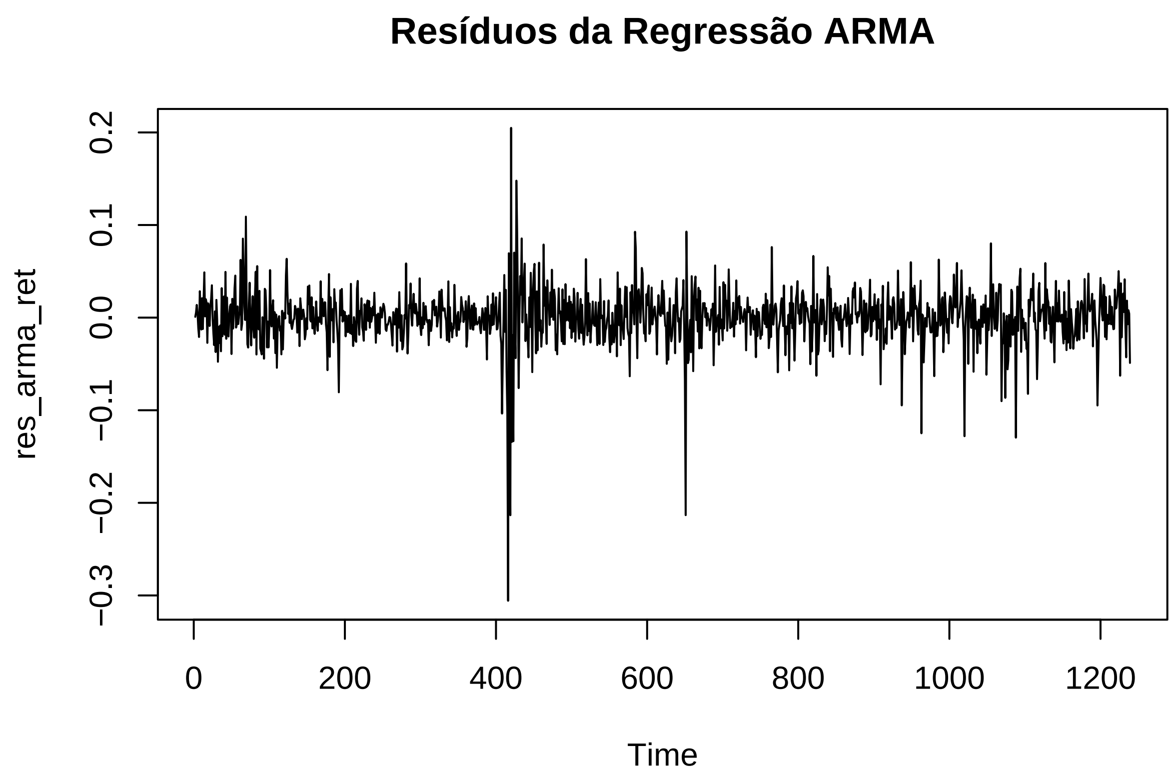
```r
arma_ret <- Arima(
      ret,
      order = c(1,
                0,
                1),
      method = "ML"
      )

arma_ret
```

```
## Series: ret
## ARIMA(1,0,1) with non-zero mean
##
## Coefficients:
##           ar1      ma1    mean
##       -0.0393  -0.0391   9e-04
## s.e.   0.2537   0.2534   8e-04
##
## sigma^2 = 0.0008996:  log likelihood = 2586.25
## AIC=-5164.49   AICc=-5164.46   BIC=-5144.01
```

Resíduos

```
res_arma_ret <- residuals(arma_ret)
```

```
plot(res_arma_ret, main = "Resíduos da Regressão ARMA")
```



Tendência Central

```
summary(res_arma_ret)
```

```
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -3.057e-01 -1.303e-02  2.853e-04  2.360e-06  1.484e-02  2.049e-01
```

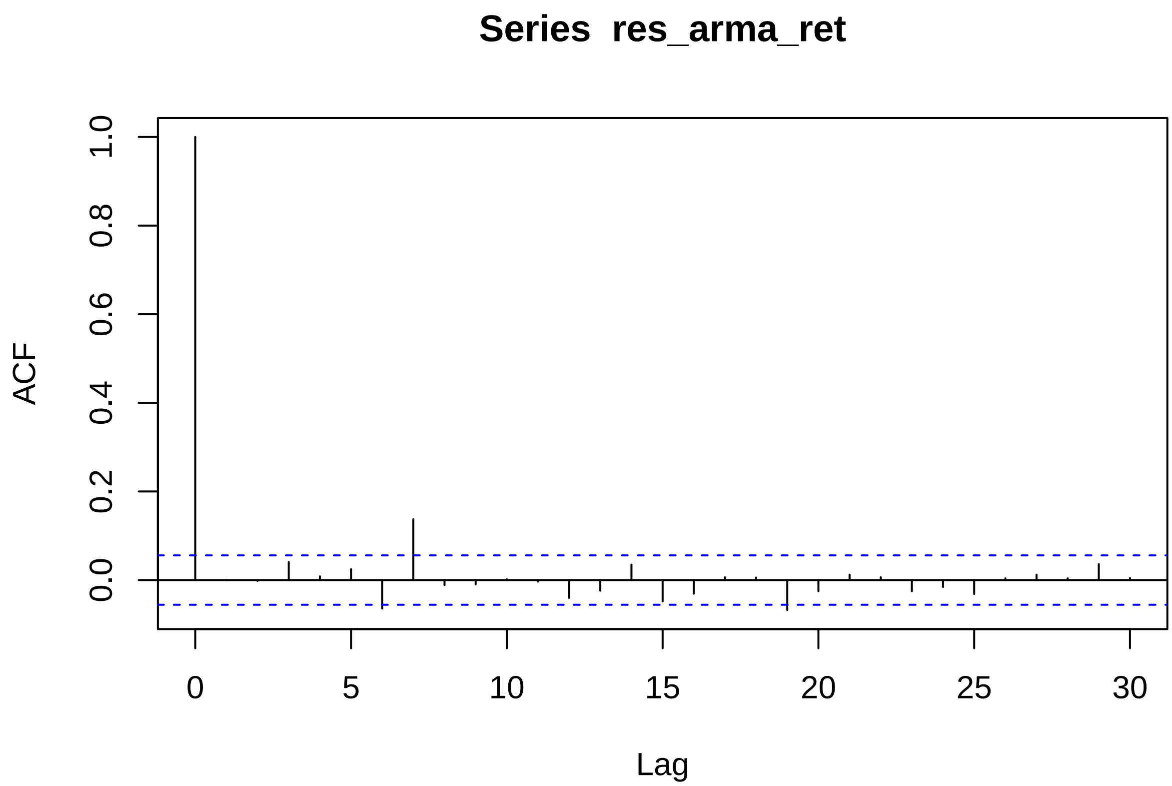
Resíduos ao Quadrado

```
summary(res_arma_ret^2)
```

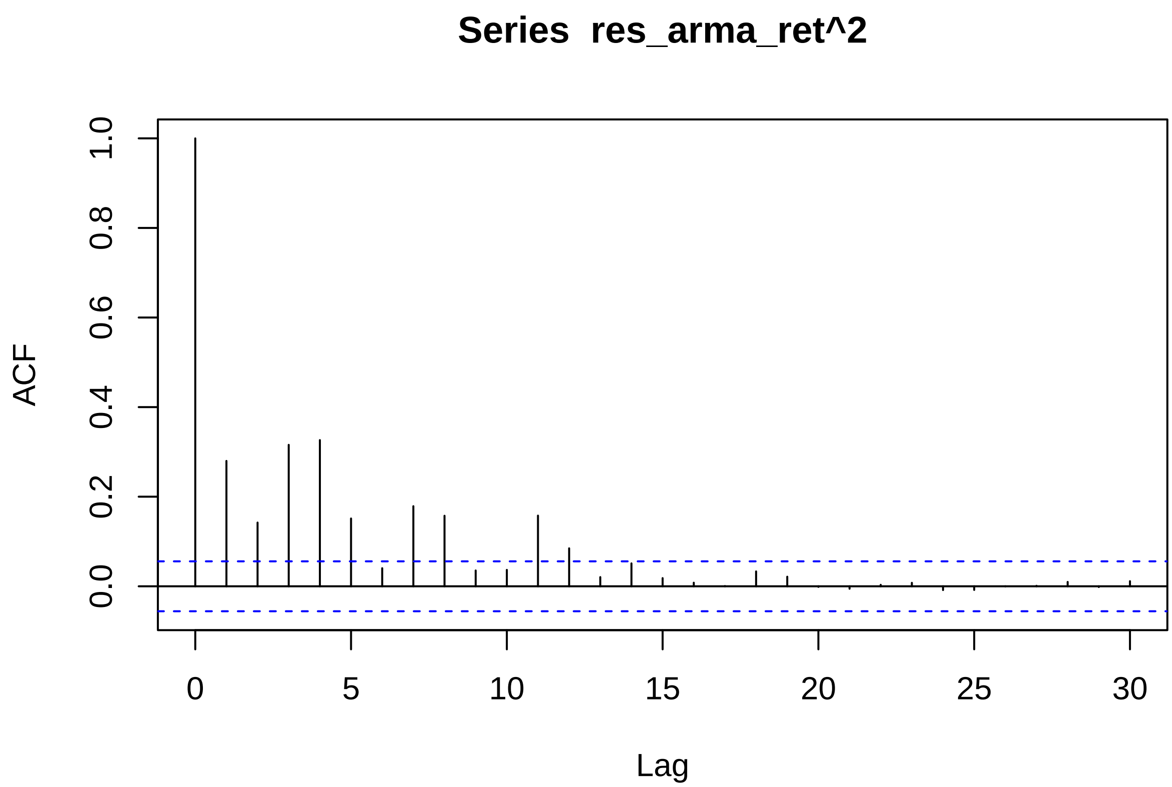
```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 0.000e+00 4.077e-05 1.972e-04 8.974e-04 7.148e-04 9.348e-02
```

ACF

```r
acf(res_arma_ret)
```




```r
acf(res_arma_ret^2)
```



Modelo Garch(1,1)

```
# Definindo a especificação do modelo garch(p,q)
spec <- ugarchspec(variance.model = list(model = "sGARCH",
                                         garchOrder = c(1, 1)),
                   mean.model = list(armaOrder = c(1, 1)))
```

```
# Ajustando o modelo garch(p,q) ao dados
fit_garch <- ugarchfit(spec, data = ret)
```

```
# Coeficientes
fit_garch@fit$matcoef
```

```
##              Estimate   Std. Error       t value      Pr(>|t|)
## mu      0.0010986245 1.684860e-04      6.520569 7.004108e-11
## ar1     0.9610907881 7.433724e-03    129.287925 0.000000e+00
## ma1    -0.9875793872 2.869302e-04 -3441.880066 0.000000e+00
## omega   0.0000780396 2.933347e-05      2.660429 7.804119e-03
## alpha1  0.1544211592 3.144427e-02      4.910948 9.063729e-07
## beta1   0.7550279287 5.995516e-02     12.593210 0.000000e+00
```

```r
# Critérios de Informação
infocriteria(fit_garch)
```

```
##
## Akaike       -4.429220
## Bayes        -4.404400
## Shibata      -4.429267
## Hannan-Quinn -4.419885
```

```r
fit_garch
```

```
##
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
##
## Conditional Variance Dynamics
## -----------------------------------
## GARCH Model  : sGARCH(1,1)
## Mean Model   : ARFIMA(1,0,1)
## Distribution : norm
##
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error    t value Pr(>|t|)
## mu      0.001099    0.000168     6.5206 0.000000
## ar1     0.961091    0.007434   129.2879 0.000000
## ma1    -0.987579    0.000287 -3441.8801 0.000000
## omega   0.000078    0.000029     2.6604 0.007804
## alpha1  0.154421    0.031444     4.9109 0.000001
## beta1   0.755028    0.059955    12.5932 0.000000
##
## Robust Standard Errors:
##         Estimate  Std. Error    t value Pr(>|t|)
## mu      0.001099    0.000149     7.3942 0.000000
## ar1     0.961091    0.008625   111.4247 0.000000
## ma1    -0.987579    0.000367 -2689.9937 0.000000
## omega   0.000078    0.000078     1.0019 0.316369
## alpha1  0.154421    0.105047     1.4700 0.141556
## beta1   0.755028    0.179725     4.2010 0.000027
##
## LogLikelihood : 2747.687
##
## Information Criteria
## ------------------------------------
##
## Akaike       -4.4292
## Bayes        -4.4044
## Shibata      -4.4293
## Hannan-Quinn -4.4199
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
```

```
##                          statistic p-value
## Lag[1]                      0.1871  0.6653
## Lag[2*(p+q)+(p+q)-1][5]     0.5596  1.0000
## Lag[4*(p+q)+(p+q)-1][9]     2.9559  0.8972
## d.o.f=2
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                          statistic p-value
## Lag[1]                      0.6948  0.4045
## Lag[2*(p+q)+(p+q)-1][5]     1.6247  0.7092
## Lag[4*(p+q)+(p+q)-1][9]     3.3007  0.7085
## d.o.f=2
##
## Weighted ARCH LM Tests
## ------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[3]    0.1035 0.500 2.000  0.7477
## ARCH Lag[5]    0.3118 1.440 1.667  0.9371
## ARCH Lag[7]    0.9903 2.315 1.543  0.9151
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  2.1536
## Individual Statistics:
## mu     0.39646
## ar1    0.10790
## ma1    0.14262
## omega  0.77741
## alpha1 0.05508
## beta1  0.42593
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:          1.49 1.68 2.12
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
## ------------------------------------
##                    t-value    prob sig
## Sign Bias          0.05986 0.95228
## Negative Sign Bias 2.04138 0.04143  **
## Positive Sign Bias 0.66126 0.50857
## Joint Effect       7.56279 0.05597   *
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     54.67    2.615e-05
## 2    30     75.31    5.501e-06
## 3    40     93.89    2.006e-06
## 4    50     93.74    1.248e-04
##
```

```
##
## Elapsed time : 0.4407332
```

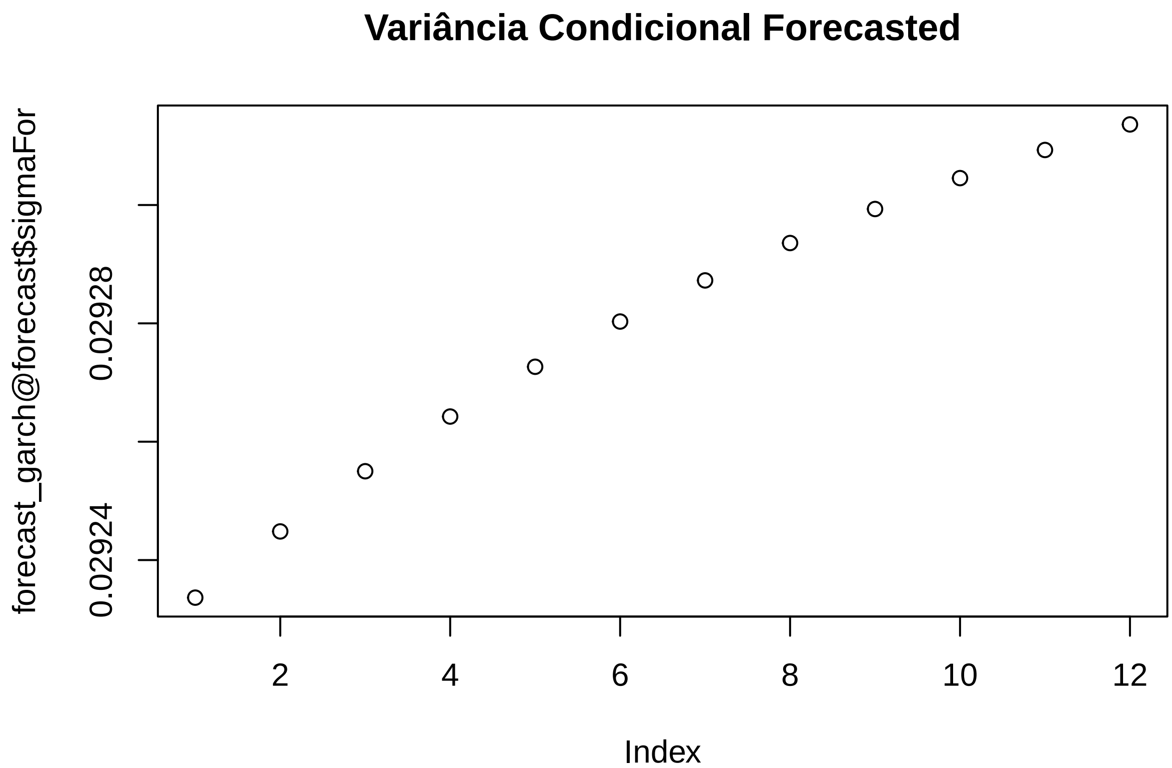
Previsão para os próximos 12 meses

```
forecast_garch <- ugarchforecast(fit_garch,
               n.ahead = 12,
               out.sample = 1239)
forecast_garch
```

```
##
## *------------------------------------*
## *       GARCH Model Forecast         *
## *------------------------------------*
## Model: sGARCH
## Horizon: 12
## Roll Steps: 0
## Out of Sample: 0
##
## 0-roll forecast [T0=1239-01-01]:
##         Series   Sigma
## T+1  0.0001809 0.02923
## T+2  0.0002167 0.02924
## T+3  0.0002510 0.02926
## T+4  0.0002840 0.02926
## T+5  0.0003157 0.02927
## T+6  0.0003461 0.02928
## T+7  0.0003754 0.02929
## T+8  0.0004035 0.02929
## T+9  0.0004306 0.02930
## T+10 0.0004566 0.02930
## T+11 0.0004816 0.02931
## T+12 0.0005056 0.02931
```

Forecast da Variância

```
plot(forecast_garch@forecast$sigmaFor, main = "Variância Condicional Forecasted")
```



Modelo Garch(1,2)

```r
# Definindo a especificação do modelo garch(p,q)
spec <- ugarchspec(variance.model = list(model = "sGARCH",
                                          garchOrder = c(1, 2)),
                   mean.model = list(armaOrder = c(1, 1)))
```

```r
# Ajustando o modelo garch(p,q) ao dados
fit_garch <- ugarchfit(spec, data = ret)
```

```r
# Coeficientes
fit_garch@fit$matcoef
```

```
##             Estimate   Std. Error      t value     Pr(>|t|)
## mu      0.0010961955 1.679120e-04     6.528392 6.647949e-11
## ar1     0.9600419664 7.550461e-03   127.150110 0.000000e+00
## ma1    -0.9871108164 2.845139e-04 -3469.464670 0.000000e+00
## omega   0.0000902979 3.042618e-05     2.967770 2.999685e-03
## alpha1  0.1889239103 3.324248e-02     5.683207 1.321920e-08
## beta1   0.2936415069 1.139354e-01     2.577262 9.958633e-03
## beta2   0.4110004929 1.082501e-01     3.796766 1.465960e-04
```

```r
# Critérios de Informação
infocriteria(fit_garch)
```

```
##
## Akaike       -4.434428
## Bayes        -4.405471
## Shibata      -4.434491
## Hannan-Quinn -4.423537
```

```r
fit_garch
```

```
##
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
##
## Conditional Variance Dynamics
## -----------------------------------
## GARCH Model  : sGARCH(1,2)
## Mean Model   : ARFIMA(1,0,1)
## Distribution : norm
##
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error    t value Pr(>|t|)
## mu      0.001096    0.000168     6.5284 0.000000
## ar1     0.960042    0.007550   127.1501 0.000000
## ma1    -0.987111    0.000285 -3469.4647 0.000000
## omega   0.000090    0.000030     2.9678 0.003000
## alpha1  0.188924    0.033242     5.6832 0.000000
## beta1   0.293642    0.113935     2.5773 0.009959
## beta2   0.411000    0.108250     3.7968 0.000147
##
## Robust Standard Errors:
##         Estimate  Std. Error    t value Pr(>|t|)
## mu      0.001096    0.000156     7.0286 0.000000
## ar1     0.960042    0.008490   113.0799 0.000000
## ma1    -0.987111    0.000335 -2943.7310 0.000000
## omega   0.000090    0.000069     1.3099 0.190218
## alpha1  0.188924    0.098516     1.9177 0.055149
## beta1   0.293642    0.122959     2.3881 0.016934
## beta2   0.411000    0.125190     3.2830 0.001027
##
## LogLikelihood : 2751.911
##
## Information Criteria
## ------------------------------------
##
## Akaike       -4.4344
## Bayes        -4.4055
## Shibata      -4.4345
## Hannan-Quinn -4.4235
##
```

```
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                     0.1909  0.6621
## Lag[2*(p+q)+(p+q)-1][5]    0.5897  1.0000
## Lag[4*(p+q)+(p+q)-1][9]    3.0807  0.8780
## d.o.f=2
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                          statistic p-value
## Lag[1]                      0.3003  0.5837
## Lag[2*(p+q)+(p+q)-1][8]     1.8468  0.8818
## Lag[4*(p+q)+(p+q)-1][14]    5.2020  0.7441
## d.o.f=3
##
## Weighted ARCH LM Tests
## ------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[4]    0.2090 0.500 2.000  0.6475
## ARCH Lag[6]    0.9121 1.461 1.711  0.7731
## ARCH Lag[8]    3.0321 2.368 1.583  0.5375
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  2.0863
## Individual Statistics:
## mu     0.37590
## ar1    0.09844
## ma1    0.14834
## omega  0.72740
## alpha1 0.05758
## beta1  0.40443
## beta2  0.42896
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:          1.69 1.9 2.35
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
## ------------------------------------
##                    t-value    prob sig
## Sign Bias          0.03161 0.97479
## Negative Sign Bias 1.72799 0.08424   *
## Positive Sign Bias 0.88158 0.37818
## Joint Effect       6.54848 0.08777   *
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     52.18    6.223e-05
## 2    30     78.27    2.071e-06
```

```
## 3    40     85.88    2.238e-05
## 4    50     91.00    2.515e-04
##
##
## Elapsed time : 0.4295468
```

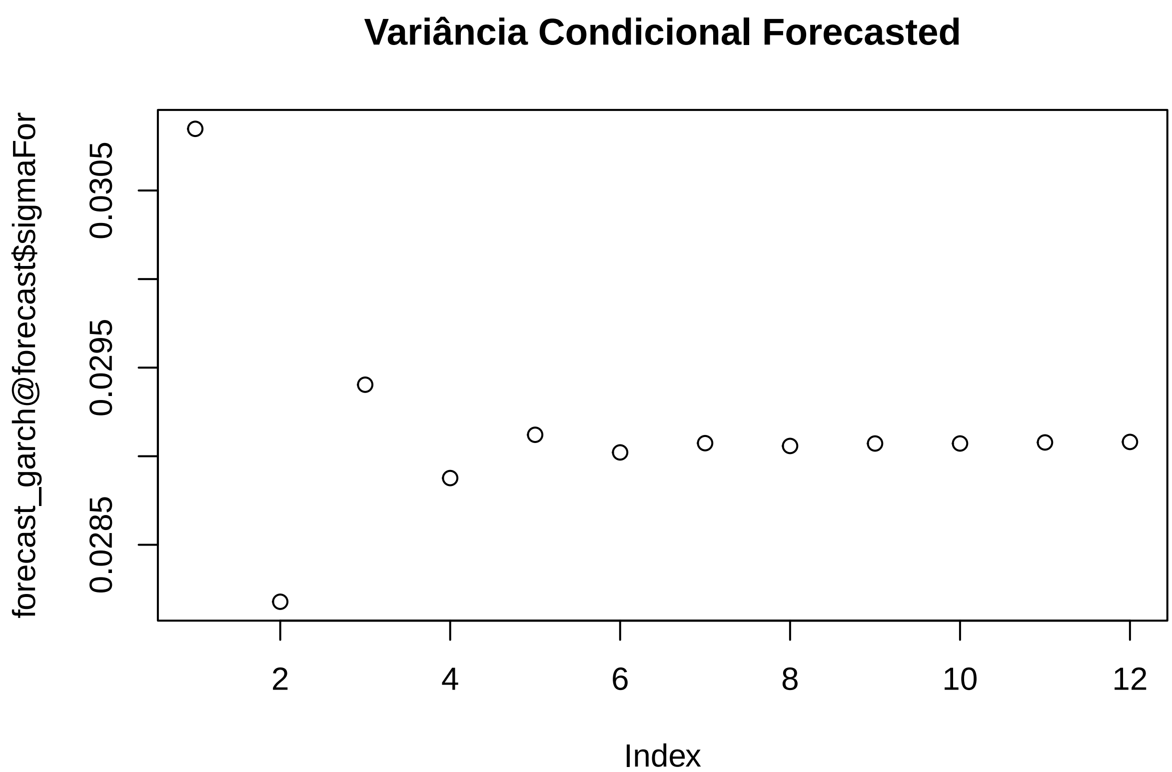
Previsão para os próximos 12 meses

```
forecast_garch <- ugarchforecast(fit_garch,
               n.ahead = 12,
               out.sample = 1239)
forecast_garch
```

```
##
## *------------------------------------*
## *       GARCH Model Forecast         *
## *------------------------------------*
## Model: sGARCH
## Horizon: 12
## Roll Steps: 0
## Out of Sample: 0
##
## 0-roll forecast [T0=1239-01-01]:
##         Series   Sigma
## T+1  8.096e-05 0.03085
## T+2  1.215e-04 0.02818
## T+3  1.605e-04 0.02940
## T+4  1.979e-04 0.02888
## T+5  2.338e-04 0.02912
## T+6  2.682e-04 0.02902
## T+7  3.013e-04 0.02907
## T+8  3.331e-04 0.02906
## T+9  3.636e-04 0.02907
## T+10 3.928e-04 0.02907
## T+11 4.209e-04 0.02908
## T+12 4.479e-04 0.02908
```

Forecast da Variância

```
plot(forecast_garch@forecast$sigmaFor, main = "Variância Condicional Forecasted")
```



Modelo Garch(1,3)

```
# Definindo a especificação do modelo garch(p,q)
spec <- ugarchspec(variance.model = list(model = "sGARCH",
                                         garchOrder = c(1, 3)),
                   mean.model = list(armaOrder = c(1, 1)))
```

```
# Ajustando o modelo garch(p,q) ao dados
fit_garch <- ugarchfit(spec, data = ret)
```

```
# Coeficientes
fit_garch@fit$matcoef
```

```
##              Estimate   Std. Error       t value     Pr(>|t|)
## mu       0.0011250749 1.619303e-04     6.9478971 3.707701e-12
## ar1      0.9594714926 7.438925e-03   128.9798535 0.000000e+00
## ma1     -0.9864580709 2.804293e-04 -3517.6710154 0.000000e+00
## omega    0.0001083531 3.369873e-05     3.2153478 1.302866e-03
## alpha1   0.2474768415 4.021061e-02     6.1545166 7.530685e-10
## beta1    0.1268292742 7.800080e-02     1.6259996 1.039497e-01
## beta2    0.0896446368 9.025361e-02     0.9932527 3.205868e-01
## beta3    0.4113852521 7.976203e-02     5.1576579 2.500580e-07
```

```r
# Critérios de Informação
infocriteria(fit_garch)
```

```
##
## Akaike       -4.443306
## Bayes        -4.410212
## Shibata      -4.443389
## Hannan-Quinn -4.430859
```

```r
fit_garch
```

```
##
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
##
## Conditional Variance Dynamics
## -----------------------------------
## GARCH Model  : sGARCH(1,3)
## Mean Model   : ARFIMA(1,0,1)
## Distribution : norm
##
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error     t value Pr(>|t|)
## mu      0.001125    0.000162     6.94790 0.000000
## ar1     0.959471    0.007439   128.97985 0.000000
## ma1    -0.986458    0.000280 -3517.67101 0.000000
## omega   0.000108    0.000034     3.21535 0.001303
## alpha1  0.247477    0.040211     6.15452 0.000000
## beta1   0.126829    0.078001     1.62600 0.103950
## beta2   0.089645    0.090254     0.99325 0.320587
## beta3   0.411385    0.079762     5.15766 0.000000
##
## Robust Standard Errors:
##         Estimate  Std. Error     t value Pr(>|t|)
## mu      0.001125    0.000152     7.38342 0.000000
## ar1     0.959471    0.007888   121.63929 0.000000
## ma1    -0.986458    0.000297 -3316.67626 0.000000
## omega   0.000108    0.000071     1.53267 0.125358
## alpha1  0.247477    0.109532     2.25941 0.023858
## beta1   0.126829    0.106546     1.19037 0.233900
## beta2   0.089645    0.130413     0.68739 0.491838
## beta3   0.411385    0.152975     2.68923 0.007162
##
## LogLikelihood : 2758.407
##
## Information Criteria
## ------------------------------------
##
## Akaike       -4.4433
## Bayes        -4.4102
## Shibata      -4.4434
```

```
## Hannan-Quinn -4.4309
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                     0.1888  0.6639
## Lag[2*(p+q)+(p+q)-1][5]    0.5931  1.0000
## Lag[4*(p+q)+(p+q)-1][9]    3.0725  0.8793
## d.o.f=2
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                          statistic p-value
## Lag[1]                   0.0003028  0.9861
## Lag[2*(p+q)+(p+q)-1][11] 2.6660526  0.9054
## Lag[4*(p+q)+(p+q)-1][19] 6.7260459  0.8101
## d.o.f=4
##
## Weighted ARCH LM Tests
## ------------------------------------
##              Statistic Shape Scale P-Value
## ARCH Lag[5]    0.01971 0.500 2.000  0.8884
## ARCH Lag[7]    0.34136 1.473 1.746  0.9382
## ARCH Lag[9]    3.38412 2.402 1.619  0.4966
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  2.023
## Individual Statistics:
## mu     0.38839
## ar1    0.10868
## ma1    0.12863
## omega  0.58350
## alpha1 0.07664
## beta1  0.35843
## beta2  0.29028
## beta3  0.38530
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:          1.89 2.11 2.59
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
## ------------------------------------
##                    t-value   prob sig
## Sign Bias           0.1086 0.9136
## Negative Sign Bias  1.2182 0.2234
## Positive Sign Bias  1.0897 0.2760
## Joint Effect        5.3672 0.1468
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
```

```
##   group statistic p-value(g-1)
## 1    20     45.88    0.0005159
## 2    30     61.11    0.0004483
## 3    40     79.67    0.0001311
## 4    50     93.83    0.0001222
##
##
## Elapsed time : 1.022884
```

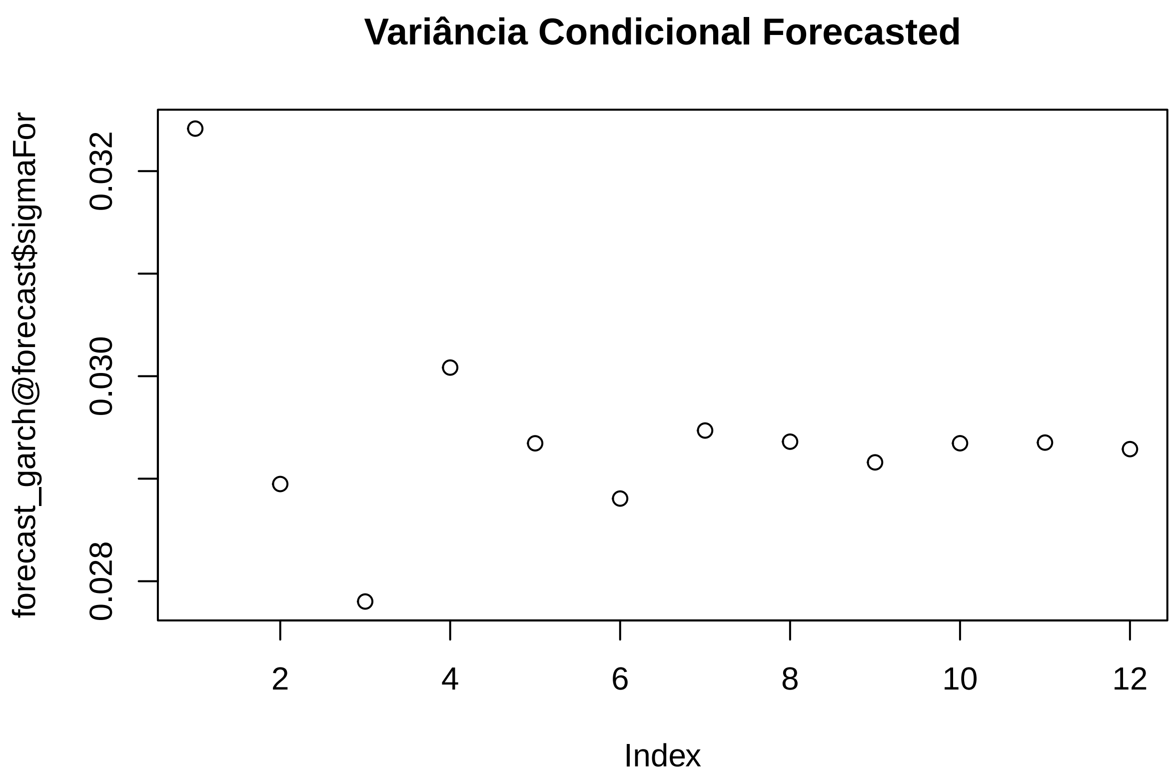
Previsão para os próximos 12 meses

```r
forecast_garch <- ugarchforecast(fit_garch,
               n.ahead = 12,
               out.sample = 1239)
forecast_garch
```

```
##
## *------------------------------------*
## *       GARCH Model Forecast         *
## *------------------------------------*
## Model: sGARCH
## Horizon: 12
## Roll Steps: 0
## Out of Sample: 0
##
## 0-roll forecast [T0=1239-01-01]:
##         Series   Sigma
## T+1  8.148e-05 0.03241
## T+2  1.238e-04 0.02895
## T+3  1.644e-04 0.02780
## T+4  2.033e-04 0.03008
## T+5  2.407e-04 0.02935
## T+6  2.765e-04 0.02881
## T+7  3.109e-04 0.02947
## T+8  3.439e-04 0.02936
## T+9  3.755e-04 0.02916
## T+10 4.059e-04 0.02935
## T+11 4.351e-04 0.02935
## T+12 4.630e-04 0.02929
```

Forecast da Variância

```r
plot(forecast_garch@forecast$sigmaFor, main = "Variância Condicional Forecasted")
```



eGARCH

```r
# Modelo eGARCH(1,1)
spec_egarch <- ugarchspec(variance.model = list(model = "eGARCH",garchOrder = c(1, 1)),
                   mean.model = list(armaOrder = c(0, 0)))
```

```r
# Ajustando o modelo eGARCH(1,1) ao dados
ret_egarch11 <- ugarchfit(spec_egarch, data = ret )
```

```r
# Coeficientes
ret_egarch11@fit$matcoef
```

```
##              Estimate   Std. Error   t value      Pr(>|t|)
## mu      0.0007327783 0.0007160823  1.023316 3.061586e-01
## omega  -0.4356214307 0.1440736870 -3.023602 2.497851e-03
## alpha1 -0.0508750578 0.0202840586 -2.508130 1.213720e-02
## beta1   0.9375916008 0.0199209020 47.065720 0.000000e+00
## gamma1  0.2456501084 0.0351146719  6.995654 2.640332e-12
```

```r
# Information Criteria
infocriteria(ret_egarch11)
```

```
##
## Akaike       -4.426421
## Bayes        -4.405738
## Shibata      -4.426454
## Hannan-Quinn -4.418642
```

```r
ret_egarch11
```

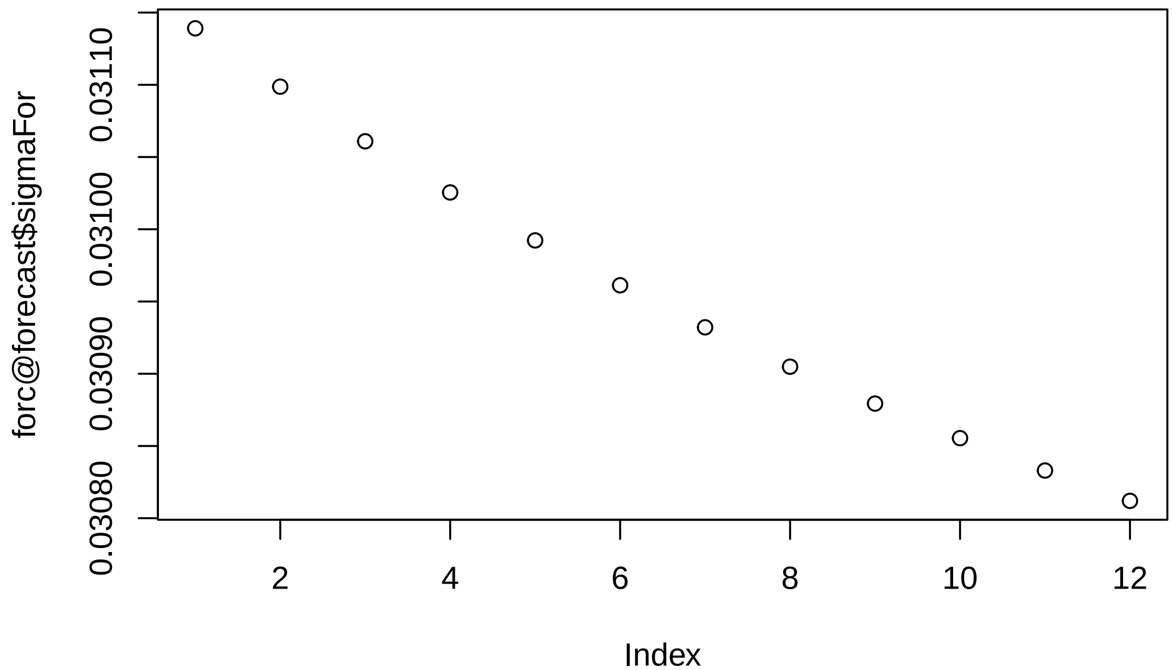
```
##
## *---------------------------------*
## *          GARCH Model Fit        *
## *---------------------------------*
##
## Conditional Variance Dynamics
## -----------------------------------
## GARCH Model  : eGARCH(1,1)
## Mean Model   : ARFIMA(0,0,0)
## Distribution : norm
##
## Optimal Parameters
## ------------------------------------
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.000733    0.000716   1.0233 0.306159
## omega  -0.435621    0.144074  -3.0236 0.002498
## alpha1 -0.050875    0.020284  -2.5081 0.012137
## beta1   0.937592    0.019921  47.0657 0.000000
## gamma1  0.245650    0.035115   6.9957 0.000000
##
## Robust Standard Errors:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu      0.000733    0.000727   1.0073 0.313783
## omega  -0.435621    0.251554  -1.7317 0.083323
## alpha1 -0.050875    0.043745  -1.1630 0.244829
## beta1   0.937592    0.035624  26.3188 0.000000
## gamma1  0.245650    0.088519   2.7751 0.005518
##
## LogLikelihood : 2744.955
##
## Information Criteria
## ------------------------------------
##
## Akaike       -4.4264
## Bayes        -4.4057
## Shibata      -4.4265
## Hannan-Quinn -4.4186
##
## Weighted Ljung-Box Test on Standardized Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                    0.02687  0.8698
```

```
## Lag[2*(p+q)+(p+q)-1][2]   0.19935  0.8546
## Lag[4*(p+q)+(p+q)-1][5]   0.60122  0.9410
## d.o.f=0
## H0 : No serial correlation
##
## Weighted Ljung-Box Test on Standardized Squared Residuals
## ------------------------------------
##                         statistic p-value
## Lag[1]                      1.376  0.2408
## Lag[2*(p+q)+(p+q)-1][5]     2.232  0.5646
## Lag[4*(p+q)+(p+q)-1][9]     3.820  0.6195
## d.o.f=2
##
## Weighted ARCH LM Tests
## ------------------------------------
##             Statistic Shape Scale P-Value
## ARCH Lag[3]  0.000279 0.500 2.000  0.9867
## ARCH Lag[5]  0.610092 1.440 1.667  0.8507
## ARCH Lag[7]  1.263246 2.315 1.543  0.8674
##
## Nyblom stability test
## ------------------------------------
## Joint Statistic:  1.682
## Individual Statistics:
## mu     0.0349
## omega  0.3767
## alpha1 0.5815
## beta1  0.4202
## gamma1 0.7258
##
## Asymptotic Critical Values (10% 5% 1%)
## Joint Statistic:          1.28 1.47 1.88
## Individual Statistic:     0.35 0.47 0.75
##
## Sign Bias Test
## ------------------------------------
##                    t-value    prob sig
## Sign Bias          0.02444 0.98050
## Negative Sign Bias 2.41017 0.01609  **
## Positive Sign Bias 0.24694 0.80499
## Joint Effect       8.17625 0.04251  **
##
##
## Adjusted Pearson Goodness-of-Fit Test:
## ------------------------------------
##   group statistic p-value(g-1)
## 1    20     57.44    9.746e-06
## 2    30     73.71    9.260e-06
## 3    40     77.93    2.118e-04
## 4    50    105.70    4.828e-06
##
##
## Elapsed time : 0.379895
```

```r
forc <- ugarchforecast(ret_egarch11, n.ahead=12)
forc
```

```
##
## *------------------------------------*
## *       GARCH Model Forecast         *
## *------------------------------------*
## Model: eGARCH
## Horizon: 12
## Roll Steps: 0
## Out of Sample: 0
##
## 0-roll forecast [T0=1239-01-01]:
##          Series   Sigma
## T+1  0.0007328 0.03114
## T+2  0.0007328 0.03110
## T+3  0.0007328 0.03106
## T+4  0.0007328 0.03103
## T+5  0.0007328 0.03099
## T+6  0.0007328 0.03096
## T+7  0.0007328 0.03093
## T+8  0.0007328 0.03090
## T+9  0.0007328 0.03088
## T+10 0.0007328 0.03086
## T+11 0.0007328 0.03083
## T+12 0.0007328 0.03081
```

```r
plot(forc@forecast$sigmaFor)
```


forc@forecast$sigmaFor
0.03080
0.03090
0.03100
0.03110
2
4
6
8
10
12
Index