# Regressão - Aplicações em Dados Financeiros

Professor João Gabriel de Moraes Souza

29/07/2022

## Bibliotecas Necessárias

```
suppressMessages(library(tidyverse))
suppressMessages(library(tidyquant))
suppressMessages(library(timetk))
suppressMessages(library(scales))
suppressMessages(library(quantmod))
suppressMessages(library(reshape2))
suppressMessages(library(car))
suppressMessages(library(agricolae))
suppressMessages(library(ggpubr))
suppressMessages(library(olsrr))
suppressMessages(library(sandwich))
suppressMessages(library(lmtest))
suppressMessages(library(graphics))
suppressMessages(library(forecast))
```

## Construindo Base de Dados

```
acoes = c('BBAS3.SA', 'BRFS3.SA', 'CIEL3.SA', 'PETR3.SA', 'MGLU3.SA', '^BVSP')
acoes_date = c('date', 'BBAS3.SA', 'BRFS3.SA', 'CIEL3.SA', 'PETR3.SA',
               'MGLU3.SA', '^BVSP')

getSymbols(acoes, src='yahoo',
from='2015-01-01',
warning=FALSE)
```

```
## pausing 1 second between requests for more than 5 symbols

## Warning: ^BVSP contains missing values. Some functions will not work if objects
## contain missing values in the middle of the series. Consider using na.omit(),
## na.approx(), na.fill(), etc to remove or replace them.

## pausing 1 second between requests for more than 5 symbols

## [1] "BBAS3.SA" "BRFS3.SA" "CIEL3.SA" "PETR3.SA" "MGLU3.SA" "^BVSP"
```

```
prices = data.frame(rename_index = "date"(BBAS3.SA),BBAS3.SA$BBAS3.SA.Close,
                   BRFS3.SA$BRFS3.SA.Close,
                   CIEL3.SA$CIEL3.SA.Close,PETR3.SA$PETR3.SA.Close,
                   MGLU3.SA$MGLU3.SA.Close, BVSP$BVSP.Close) %>%
        `colnames<-` (acoes_date) %>%
```

```
        drop_na()
head(prices)
```

```
##                  date BBAS3.SA BRFS3.SA CIEL3.SA PETR3.SA MGLU3.SA ^BVSP
## 2015-01-02 2015-01-02    22.65    62.18 22.98611     9.00 0.232812 48512
## 2015-01-05 2015-01-05    22.18    61.00 22.20486     8.27 0.237187 47517
## 2015-01-06 2015-01-06    22.49    61.55 21.75926     8.06 0.234062 48001
## 2015-01-07 2015-01-07    23.48    64.30 21.96180     8.45 0.241875 49463
## 2015-01-08 2015-01-08    23.56    63.15 22.56945     9.02 0.240000 49943
## 2015-01-09 2015-01-09    22.54    61.78 23.04398     9.29 0.231875 48840
```

## Calculando Retornos

Os retornos calculados serão os valores de retornos contínuos, ou seja $\mathbb{E}(R_i) = log(P_{it}) - log(P_{it-1})$.

```
returns = prices %>%
gather(asset, prices, -date) %>%
group_by(asset) %>%
tq_transmute(mutate_fun = periodReturn,
period='daily',
type='log') %>%
spread(asset, daily.returns) %>%
select(date, acoes)
```

```
## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(acoes)` instead of `acoes` to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

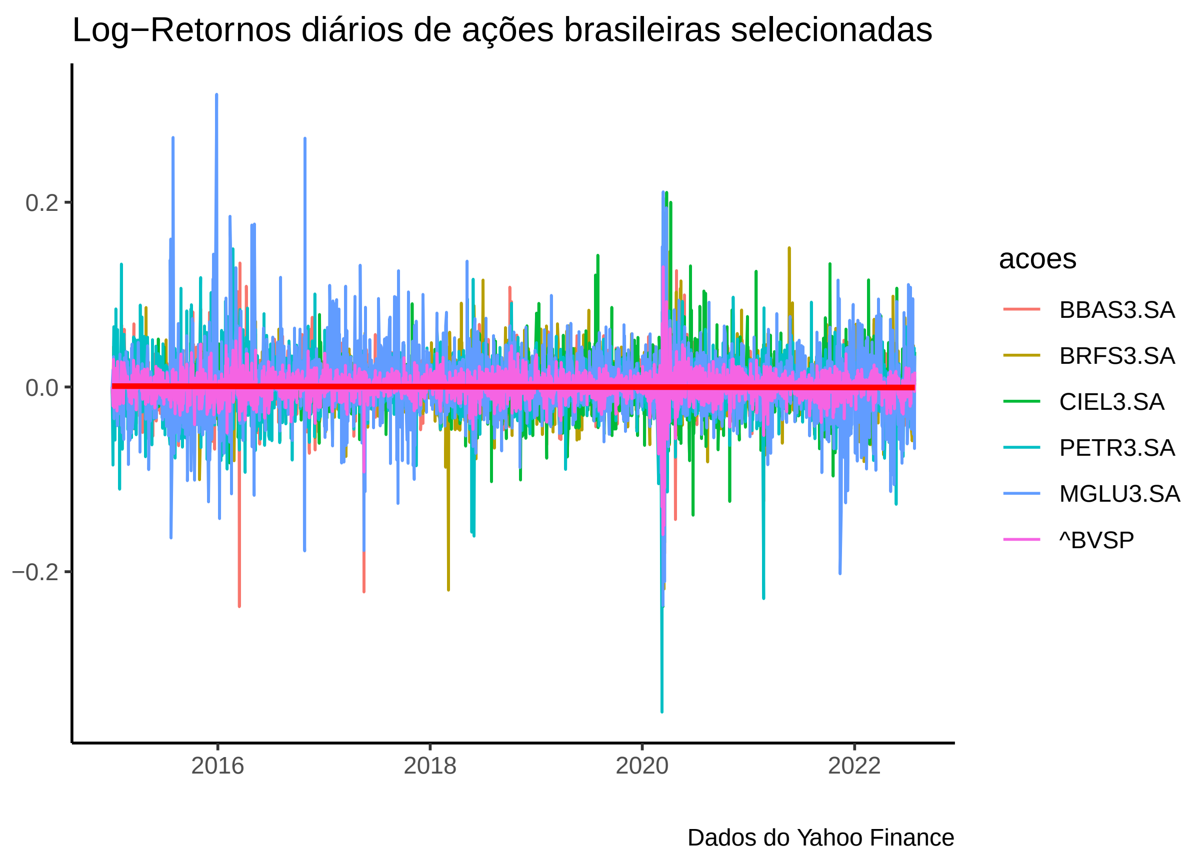
```
head(returns)
```

```
## # A tibble: 6 x 7
##   date       BBAS3.SA BRFS3.SA CIEL3.SA PETR3.SA MGLU3.SA  `^BVSP`
##   <date>        <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
## 1 2015-01-02  0        0        0          0       0        0
## 2 2015-01-05 -0.0210  -0.0192  -0.0346   -0.0846  0.0186  -0.0207
## 3 2015-01-06  0.0139   0.00898 -0.0203   -0.0257 -0.0133   0.0101
## 4 2015-01-07  0.0431   0.0437   0.00927   0.0473  0.0328   0.0300
## 5 2015-01-08  0.00340 -0.0180   0.0273    0.0653 -0.00778  0.00966
## 6 2015-01-09 -0.0443  -0.0219   0.0208    0.0295 -0.0344  -0.0223
```

Para visualização criamos um gráfico similar ao feito no Python

```
returns_melt = melt(returns, id.vars = "date")
  colnames(returns_melt) = c("date", "acoes", "retornos")

ggplot(returns_melt, aes(x=date, y= retornos, col = acoes))+
  geom_line() +
  theme_classic() +
  geom_smooth(method = "lm", col="red") +
  labs(x='', y='',
title='Log-Retornos diários de ações brasileiras selecionadas',
caption='Dados do Yahoo Finance')
```

```
## `geom_smooth()` using formula 'y ~ x'
```




## Calculando da Carteira

Para a construção da Carteira iremos aplicar a seguinte fórmula:

$$\mathbb{E}(R_{pt}) = \sum_{i=1}^{n} \sum_{t=1}^{T} w_i \mathbb{E}(R_{it})$$

Para isso vamos usar o *dataframe* chamado *returns_melt*.

```
pesos = c(0.15, 0.15, 0.15, 0.15, 0.4, 0)
pesos
```

```
## [1] 0.15 0.15 0.15 0.15 0.40 0.00
```

```
sum(pesos)
```

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

```
wts_tbl <- tibble(acoes = acoes,
                  wts = pesos)
head(wts_tbl)
```

```
## # A tibble: 6 x 2
##   acoes      wts
##   <chr>    <dbl>
## 1 BBAS3.SA  0.15
## 2 BRFS3.SA  0.15
```

```
## 3 CIEL3.SA  0.15
## 4 PETR3.SA  0.15
## 5 MGLU3.SA  0.4
## 6 ^BVSP     0
```

```
returns_port = left_join(returns_melt,wts_tbl, by = 'acoes')
head(returns_port)
```

```
##         date    acoes     retornos  wts
## 1 2015-01-02 BBAS3.SA  0.000000000 0.15
## 2 2015-01-05 BBAS3.SA -0.020968870 0.15
## 3 2015-01-06 BBAS3.SA  0.013879784 0.15
## 4 2015-01-07 BBAS3.SA  0.043078229 0.15
## 5 2015-01-08 BBAS3.SA  0.003401321 0.15
## 6 2015-01-09 BBAS3.SA -0.044258763 0.15
```

```
returns_port %>%
  group_by(acoes) %>%
  slice(c(1,2))
```

```
## # A tibble: 12 x 4
## # Groups:   acoes [6]
##    date       acoes    retornos   wts
##    <date>     <chr>       <dbl> <dbl>
##  1 2015-01-02 ^BVSP      0       0
##  2 2015-01-05 ^BVSP     -0.0207  0
##  3 2015-01-02 BBAS3.SA   0       0.15
##  4 2015-01-05 BBAS3.SA  -0.0210  0.15
##  5 2015-01-02 BRFS3.SA   0       0.15
##  6 2015-01-05 BRFS3.SA  -0.0192  0.15
##  7 2015-01-02 CIEL3.SA   0       0.15
##  8 2015-01-05 CIEL3.SA  -0.0346  0.15
##  9 2015-01-02 MGLU3.SA   0       0.4
## 10 2015-01-05 MGLU3.SA   0.0186  0.4
## 11 2015-01-02 PETR3.SA   0       0.15
## 12 2015-01-05 PETR3.SA  -0.0846  0.15
```

```
returns_port <- returns_port %>%
  mutate(wt_retornos = wts * retornos)
```

Vamos agora dar uma olhada na nova variável criada

```
returns_port %>%
  group_by(acoes) %>%
  slice(c(1,2))
```

```
## # A tibble: 12 x 5
## # Groups:   acoes [6]
##    date       acoes    retornos   wts wt_retornos
##    <date>     <chr>       <dbl> <dbl>       <dbl>
##  1 2015-01-02 ^BVSP      0       0        0
##  2 2015-01-05 ^BVSP     -0.0207  0        0
##  3 2015-01-02 BBAS3.SA   0       0.15     0
##  4 2015-01-05 BBAS3.SA  -0.0210  0.15    -0.00315
##  5 2015-01-02 BRFS3.SA   0       0.15     0
##  6 2015-01-05 BRFS3.SA  -0.0192  0.15    -0.00287
##  7 2015-01-02 CIEL3.SA   0       0.15     0
```

```
##  8 2015-01-05 CIEL3.SA  -0.0346  0.15    -0.00519
##  9 2015-01-02 MGLU3.SA   0       0.4      0
## 10 2015-01-05 MGLU3.SA   0.0186  0.4      0.00745
## 11 2015-01-02 PETR3.SA   0       0.15     0
## 12 2015-01-05 PETR3.SA  -0.0846  0.15    -0.0127
```

Agora temos os retornos ponderados pelos pesos de cada investimento das ações na carteira, basta então somar estes valores por grupo de ação e teremos os valores dos retornos diários da carteira.

```
returns_port <- returns_port %>%
  group_by(date) %>%
  summarise(port_ret = sum(wt_retornos))

head(returns_port)
```

```
## # A tibble: 6 x 2
##   date       port_ret
##   <date>        <dbl>
## 1 2015-01-02  0
## 2 2015-01-05 -0.0164
## 3 2015-01-06 -0.00878
## 4 2015-01-07  0.0346
## 5 2015-01-08  0.00858
## 6 2015-01-09 -0.0162
```

Juntamos agora com a base de retornos

```
returns = left_join(returns ,returns_port, by = 'date')
head(returns)
```

```
## # A tibble: 6 x 8
##   date       BBAS3.SA BRFS3.SA CIEL3.SA PETR3.SA MGLU3.SA  `^BVSP` port_ret
##   <date>        <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
## 1 2015-01-02  0        0        0          0       0         0        0
## 2 2015-01-05 -0.0210  -0.0192  -0.0346   -0.0846  0.0186  -0.0207  -0.0164
## 3 2015-01-06  0.0139   0.00898 -0.0203   -0.0257 -0.0133   0.0101  -0.00878
## 4 2015-01-07  0.0431   0.0437   0.00927   0.0473  0.0328   0.0300   0.0346
## 5 2015-01-08  0.00340 -0.0180   0.0273    0.0653 -0.00778  0.00966  0.00858
## 6 2015-01-09 -0.0443  -0.0219   0.0208    0.0295 -0.0344  -0.0223  -0.0162
```

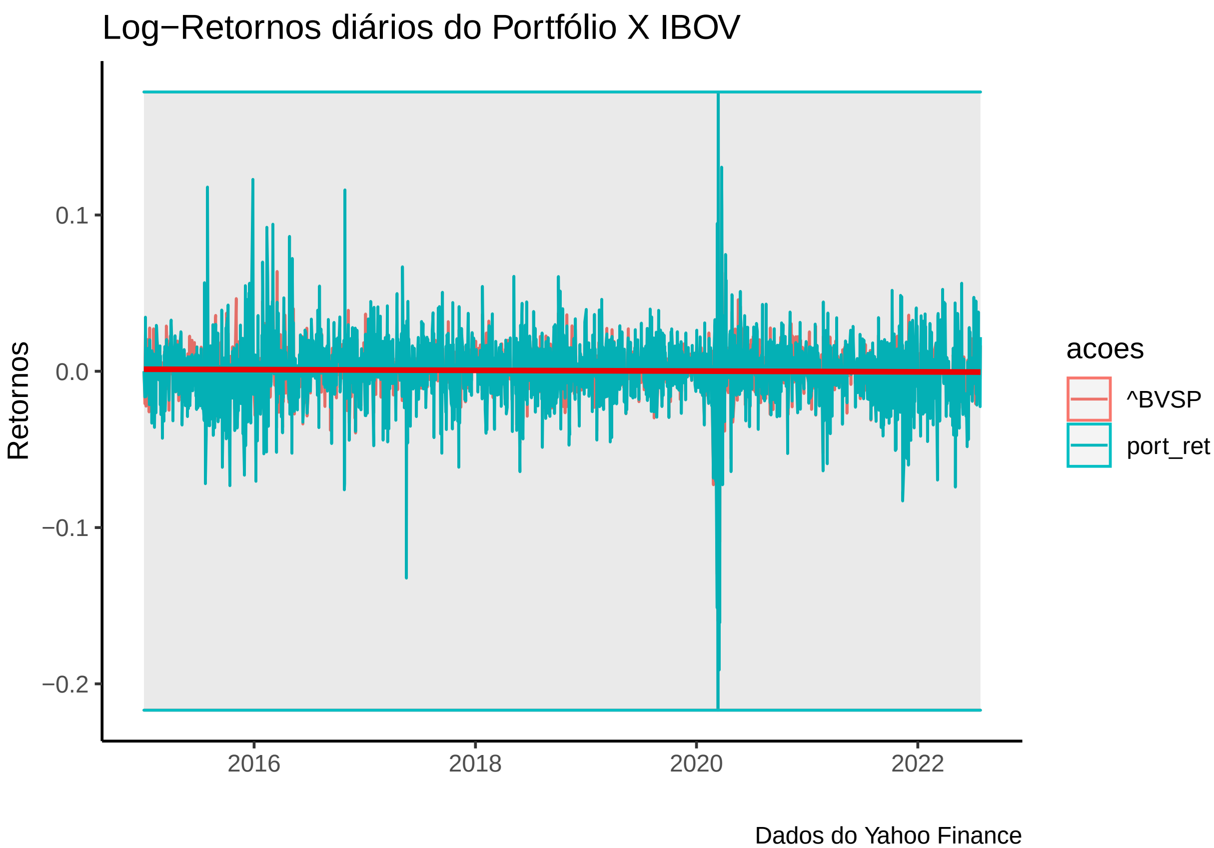
```
returns_port_melt = returns %>%
  select(c(date,`^BVSP`, port_ret))

returns_port_melt_plot = melt(returns_port_melt , id.vars = "date")
  colnames(returns_port_melt_plot) = c("date", "acoes", "retornos")

returns_port_melt_plot %>%
  group_by(acoes) %>%
  slice(c(1,2))
```

```
## # A tibble: 4 x 3
## # Groups:   acoes [2]
##   date       acoes    retornos
##   <date>     <fct>       <dbl>
## 1 2015-01-02 ^BVSP      0
## 2 2015-01-05 ^BVSP     -0.0207
## 3 2015-01-02 port_ret   0
```

```
## 4 2015-01-05 port_ret  -0.0164
```

```r
ggplot(returns_port_melt_plot, aes(x=date, y= retornos, col = acoes))+
  geom_line() +
  theme_classic() +
  geom_smooth(method = "lm", col="red") +
   geom_ribbon(aes(ymin = min(retornos), ymax = max(retornos), group = acoes),
               alpha = 0.05) +
  labs(x='', y='Retornos',
title='Log-Retornos diários do Portfólio X IBOV',
caption='Dados do Yahoo Finance')
```

```
## `geom_smooth()` using formula 'y ~ x'
```

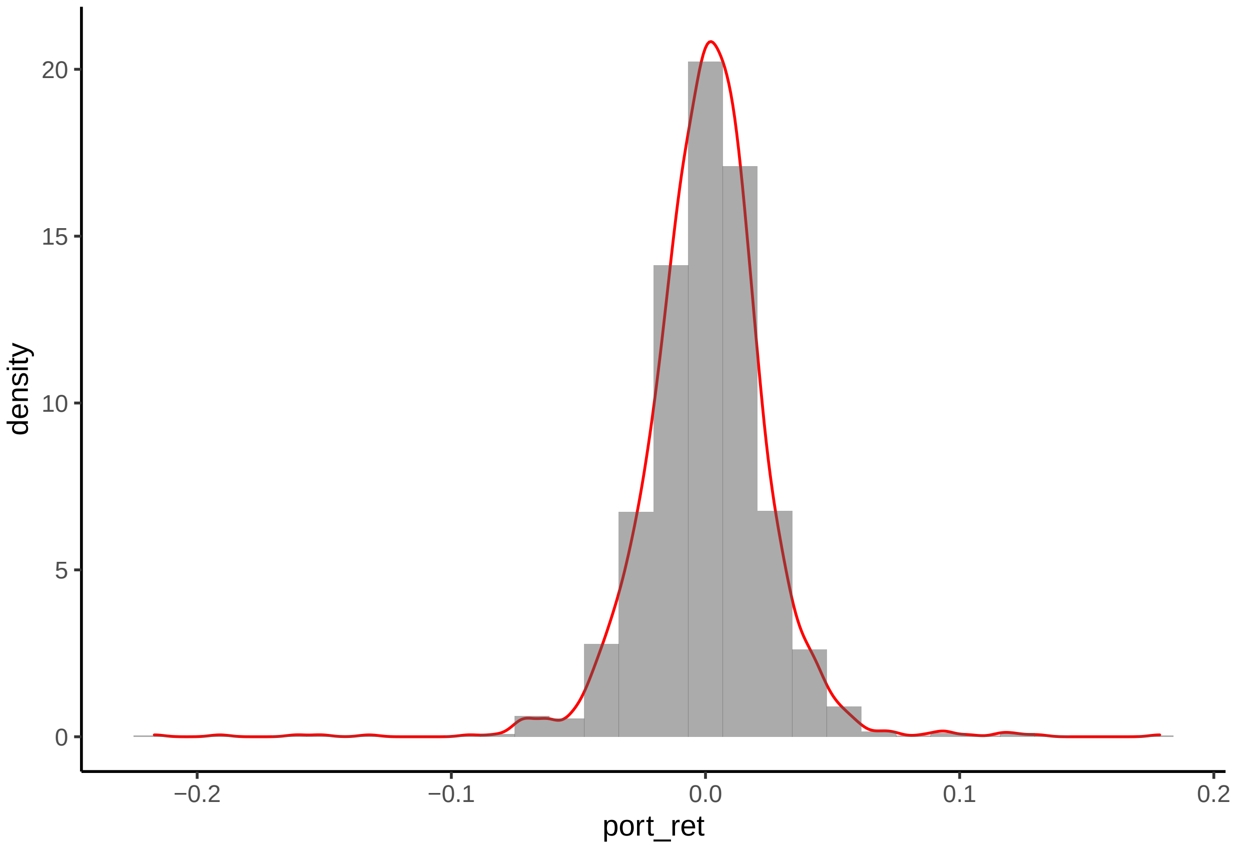



## Analisando as FDPs das variáveis de interesse

Fazemos inicialmente uma inspeção visual dos histogramas de alguns retornos

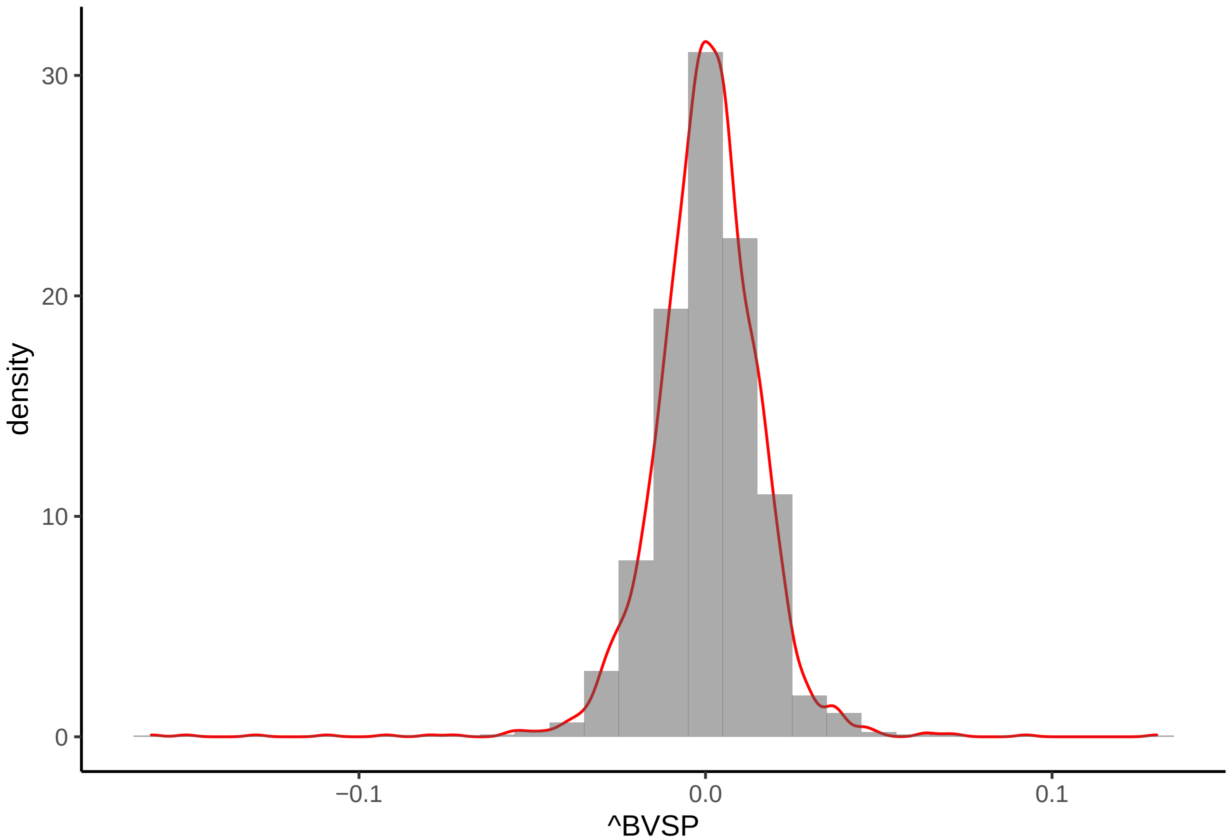
```r
ggplot(returns) +
  geom_density(aes(x=port_ret), colour = "red") +
  geom_histogram(aes(x=port_ret, y=..density..), alpha = 0.5) +
  theme_classic()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



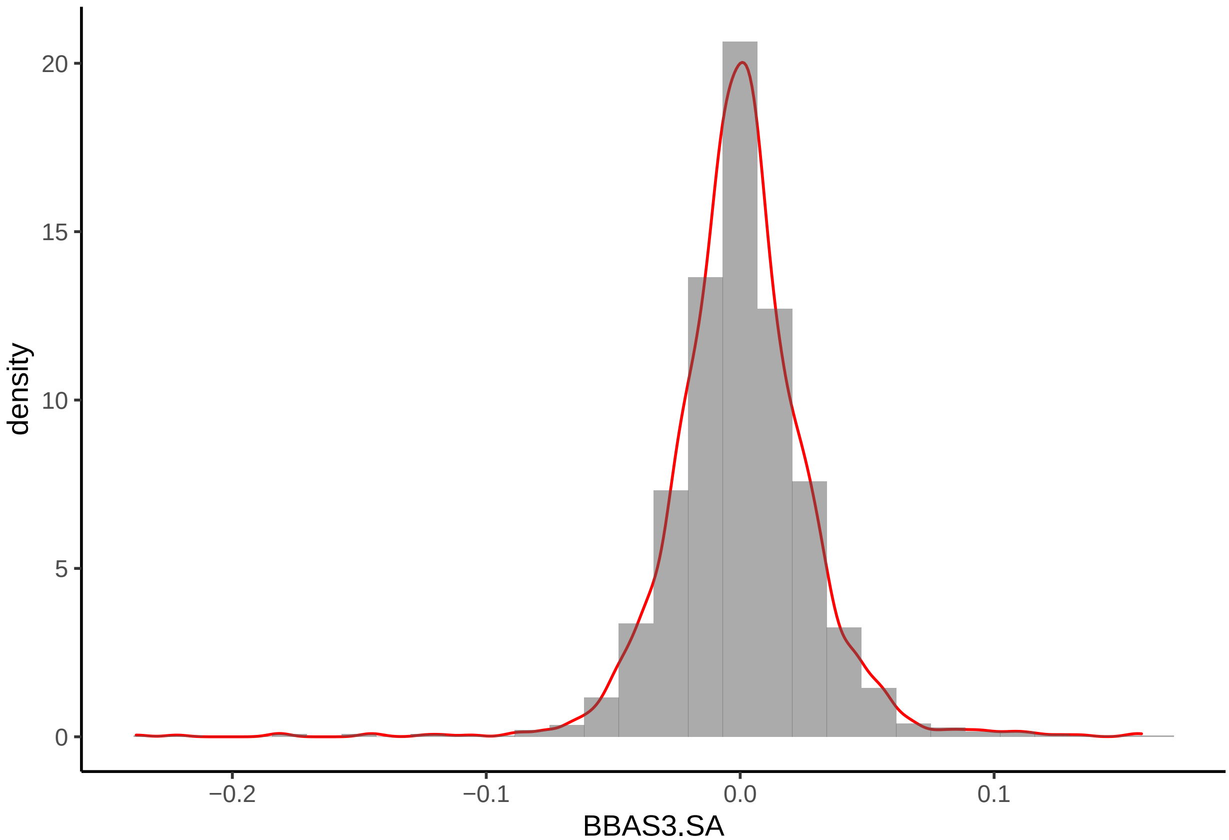

```
ggplot(returns) +
  geom_density(aes(x=`^BVSP`), colour = "red") +
  geom_histogram(aes(x=`^BVSP`, y=..density..), alpha = 0.5) +
  theme_classic()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



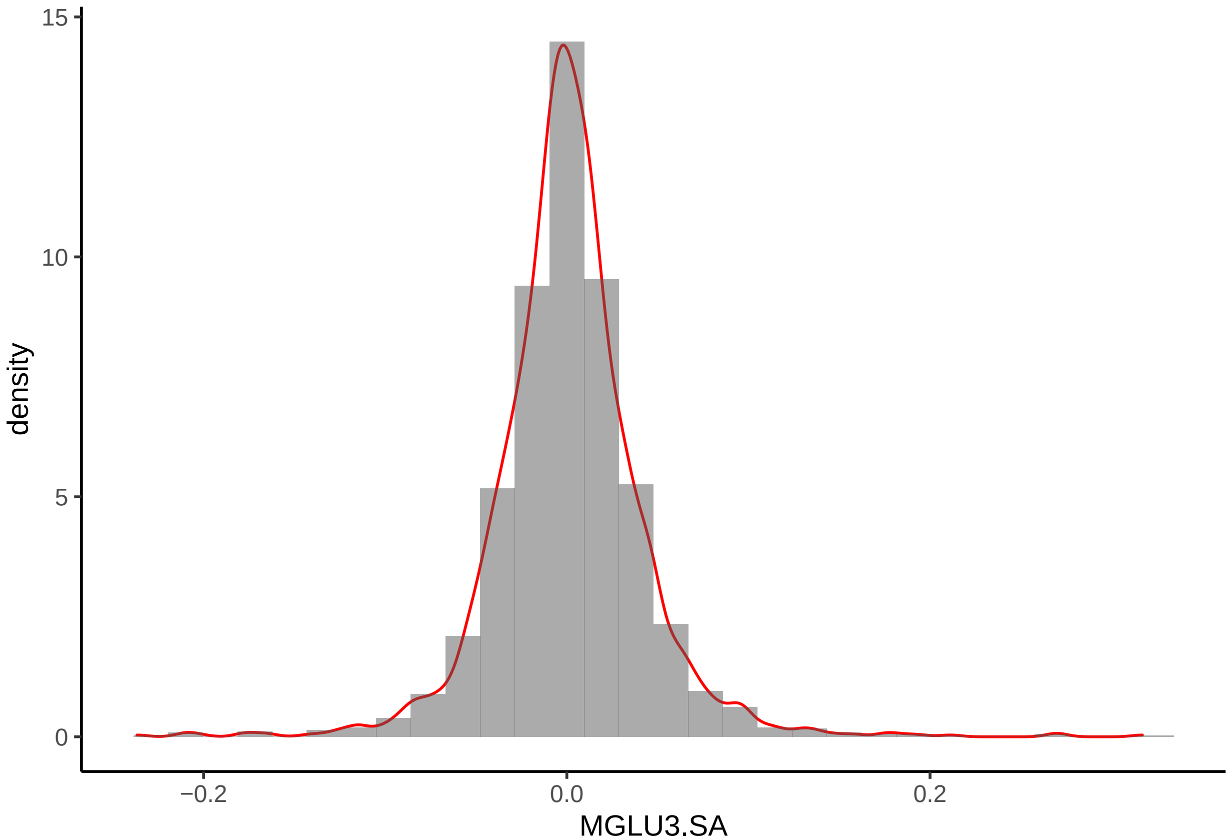

```
ggplot(returns) +
  geom_density(aes(x=BBAS3.SA), colour = "red") +
  geom_histogram(aes(x=BBAS3.SA, y=..density..), alpha = 0.5) +
  theme_classic()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
ggplot(returns) +
  geom_density(aes(x=MGLU3.SA), colour = "red") +
  geom_histogram(aes(x=MGLU3.SA, y=..density..), alpha = 0.5) +
  theme_classic()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




## Estimando o Modelo de Regressão

De modo a estimar o modelo por MQO, definindo o retorno de um ativo qualquer $R_i$ como a variável dependente e o índice *IBOVESPA* como a variável independente, fazemos uso da função lm() do R para realizar uma regressão linear simples. Este modelo é conhecido como *Market Model*.

$$\mathbb{E}(R_i) = \beta_1 + \beta_2 \cdot \mathbb{E}(R_m)$$

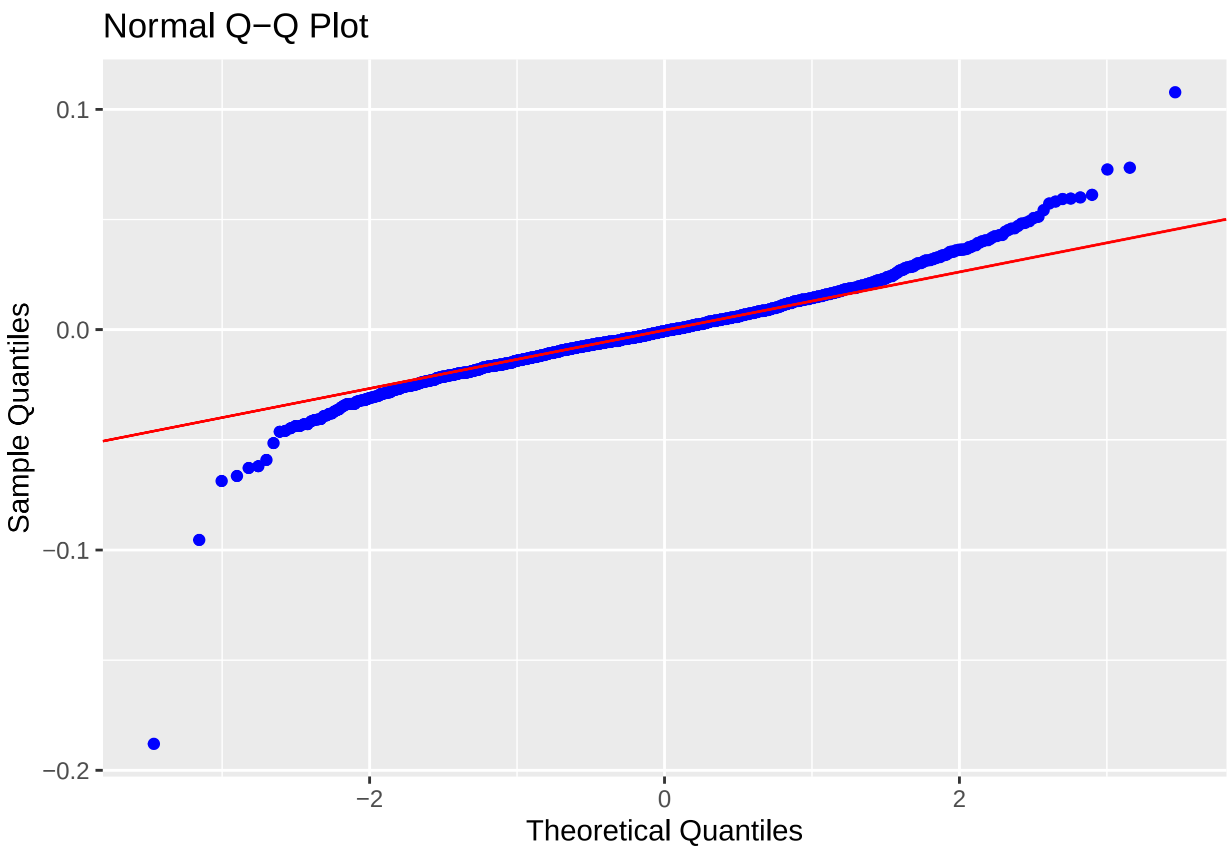
```
bb_market_model <- lm(BBAS3.SA ~ `^BVSP` , data = returns)
summary(bb_market_model)
```

```
##
## Call:
## lm(formula = BBAS3.SA ~ `^BVSP`, data = returns)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.187994 -0.009162 -0.000679  0.008670  0.107754
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.000299   0.000388  -0.771    0.441
## `^BVSP`      1.370393   0.023674  57.886   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.01679 on 1871 degrees of freedom
## Multiple R-squared:  0.6417, Adjusted R-squared:  0.6415
## F-statistic:  3351 on 1 and 1871 DF,  p-value: < 2.2e-16
```

## Diagnóstico do Modelo

```r
ols_plot_resid_qq(bb_market_model)
```



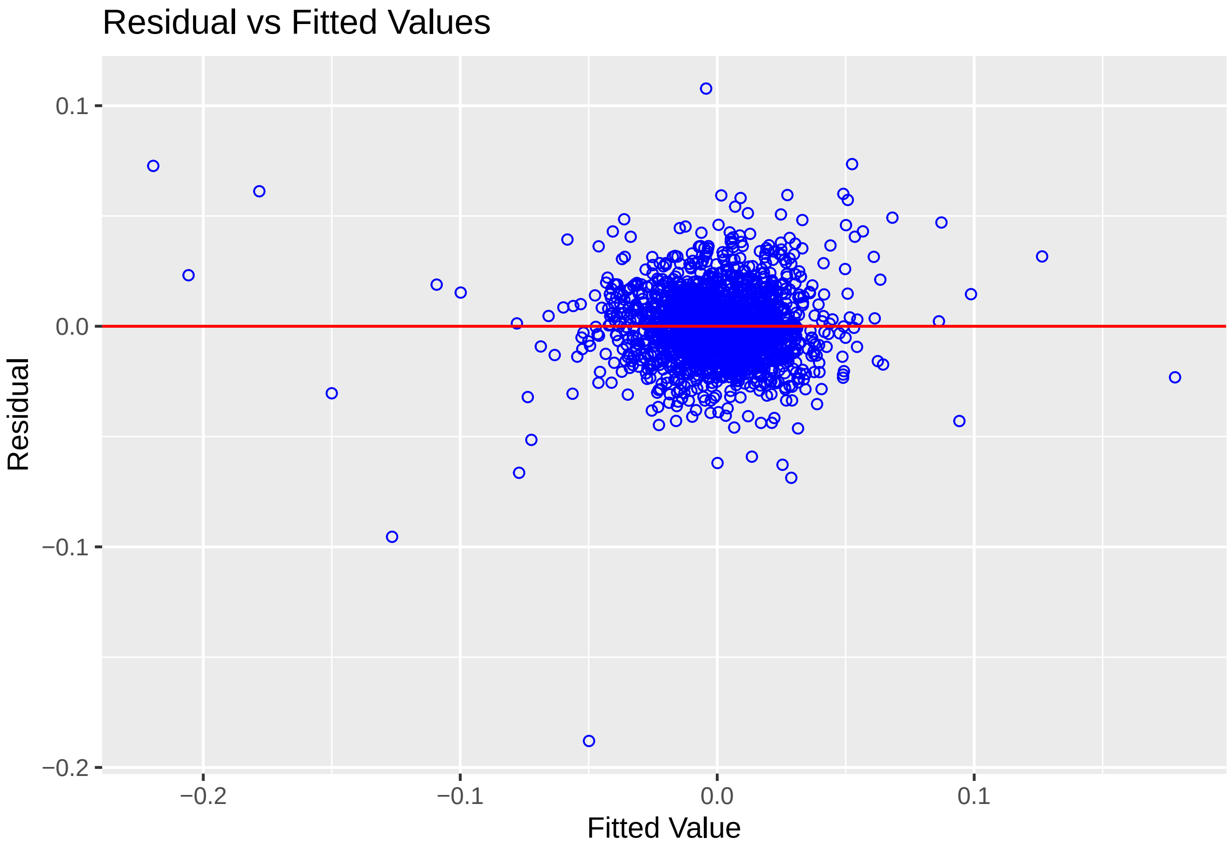

```r
ols_test_normality(bb_market_model)
```

```
## -----------------------------------------------
##        Test             Statistic       pvalue
## -----------------------------------------------
## Shapiro-Wilk              0.9394         0.0000
## Kolmogorov-Smirnov        0.0555         0.0000
## Cramer-von Mises        604.8432         0.0000
## Anderson-Darling         12.6457         0.0000
## -----------------------------------------------
```

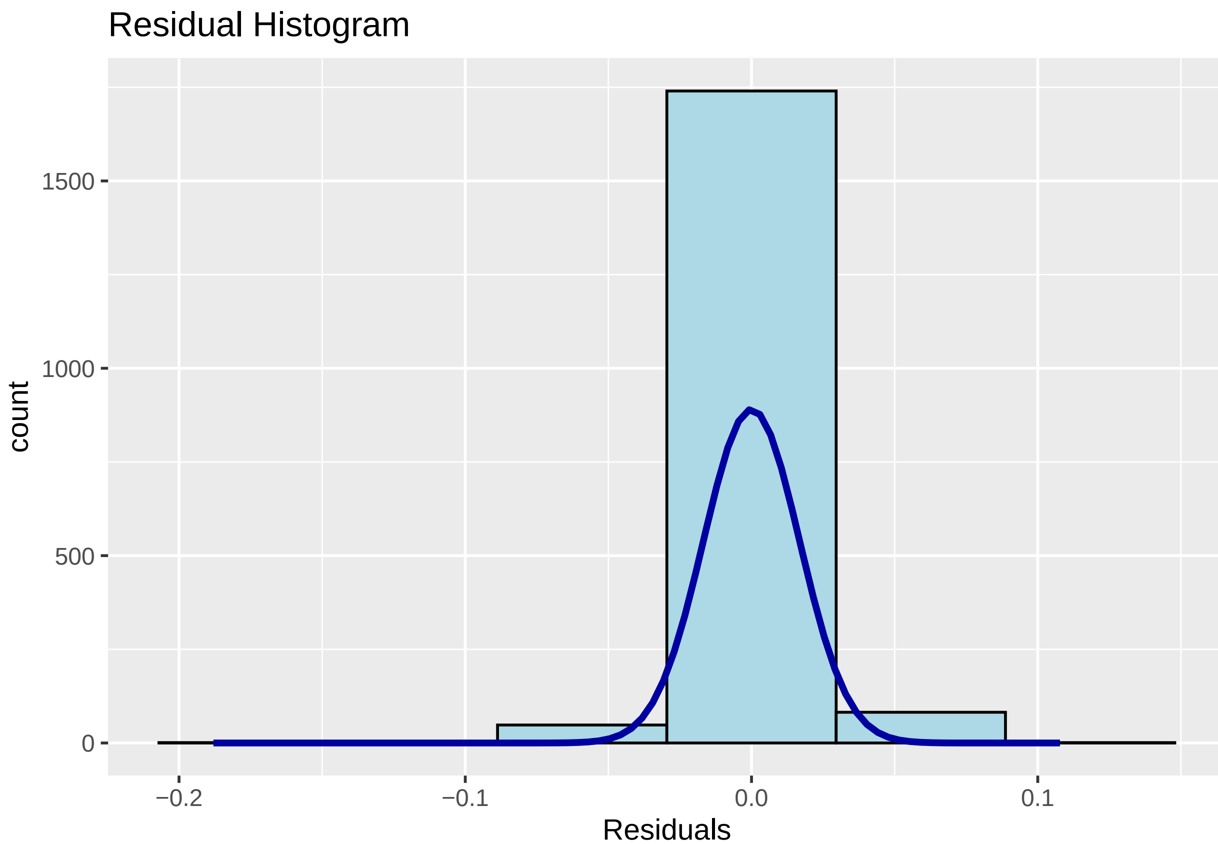
```r
ols_test_correlation(bb_market_model)
```

```
## [1] 0.9678642
```

```
ols_plot_resid_fit(bb_market_model)
```




```
ols_plot_resid_hist(bb_market_model)
```




## Testes de Homogeneidade da Variância (Homecedasticidade) e Autocorrelação

```
lmtest::bptest(bb_market_model)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  bb_market_model
## BP = 10.18, df = 1, p-value = 0.00142
```

```
car::ncvTest(bb_market_model)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 64.4396, Df = 1, p = 9.9537e-16
```

```
lmtest::dwtest(bb_market_model)
```

```
##
##  Durbin-Watson test
##
## data:  bb_market_model
## DW = 2.0151, p-value = 0.629
## alternative hypothesis: true autocorrelation is greater than 0
```

## Corrigindo as Estimações para Heterocedasticidade

```
coeftest(bb_market_model, vcov = vcovHC(bb_market_model, "HC1"))
```

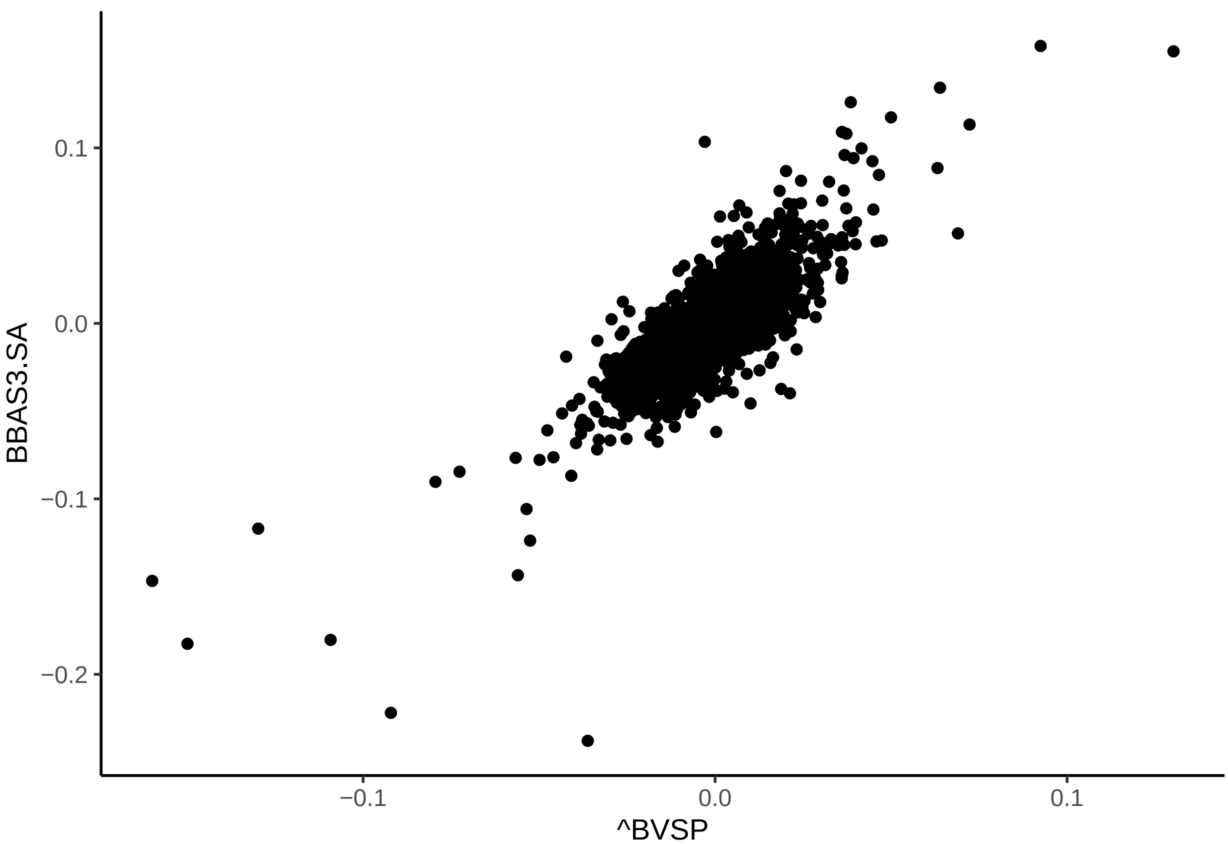
```
##
## t test of coefficients:
##
##                Estimate  Std. Error t value Pr(>|t|)
## (Intercept) -0.00029902  0.00039077 -0.7652   0.4442
## `^BVSP`      1.37039341  0.04604237 29.7637   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

## Plotando as observações em um Gráfico

Agora iremos plotar os dados e o modelo estimado em um gráfico.

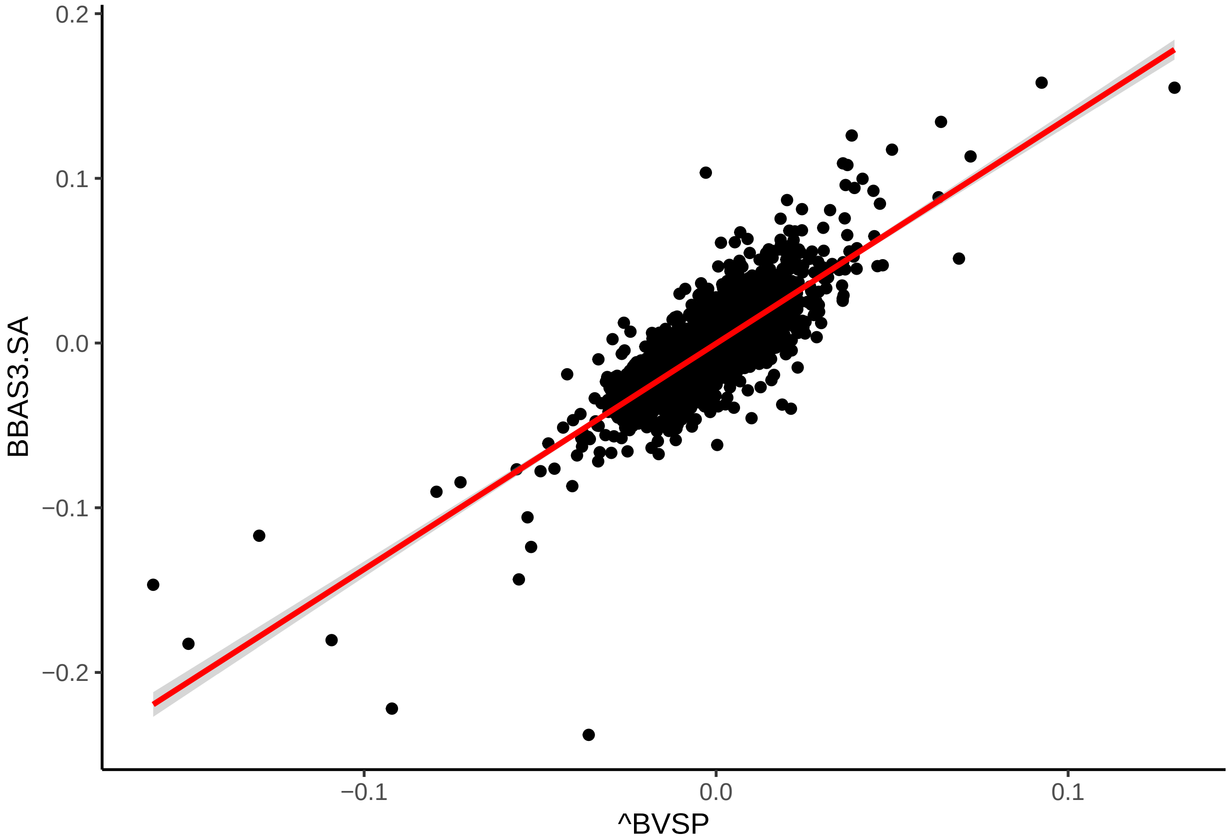
1º plotamos o gráfico só com as observações:

```
data.graph = ggplot(returns, aes(x=`^BVSP`, y=BBAS3.SA))+
                     geom_point() +
                     theme_classic()
data.graph
```




```
data.graph = data.graph +
  geom_smooth(method="lm", col="red", level=0.95)

data.graph
```

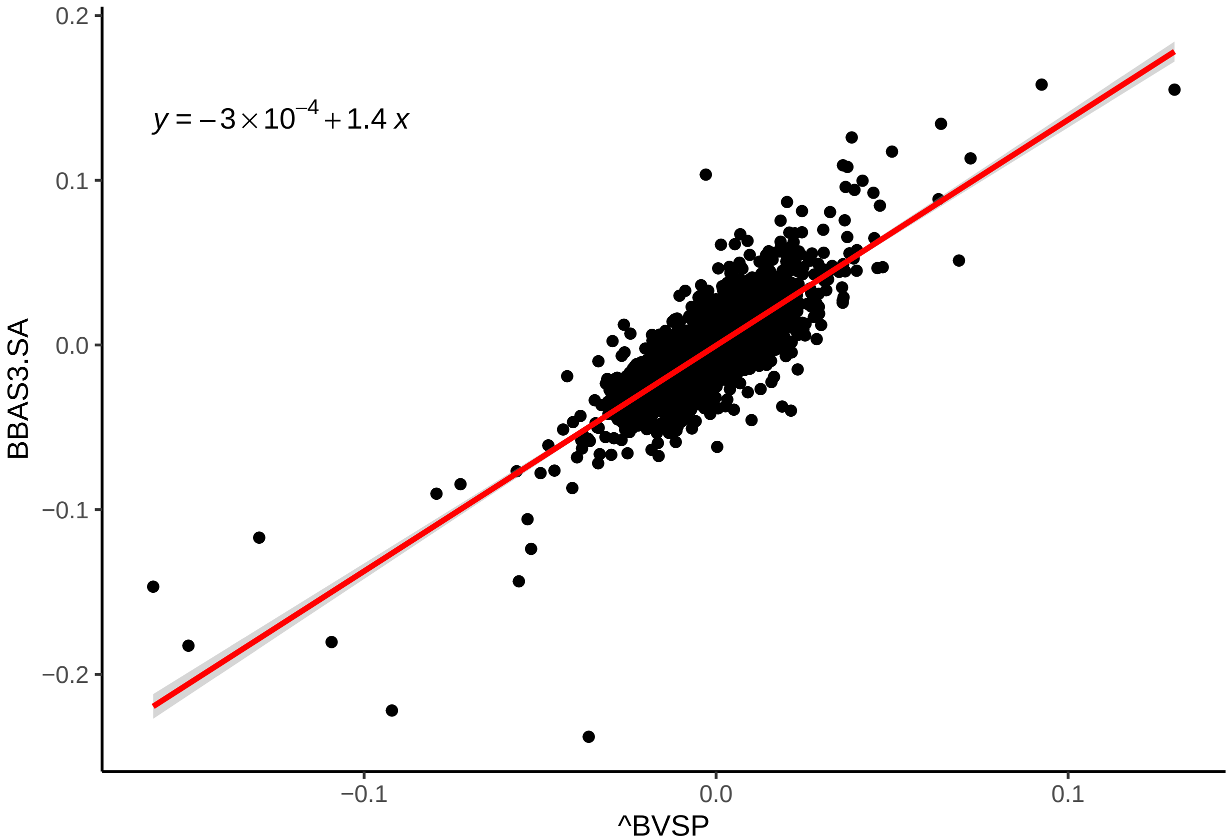
```
## `geom_smooth()` using formula 'y ~ x'
```




Por fim plotamos o gráfico com a regressão linear proposta

```
data.graph <- data.graph +
  stat_regline_equation()

data.graph
```

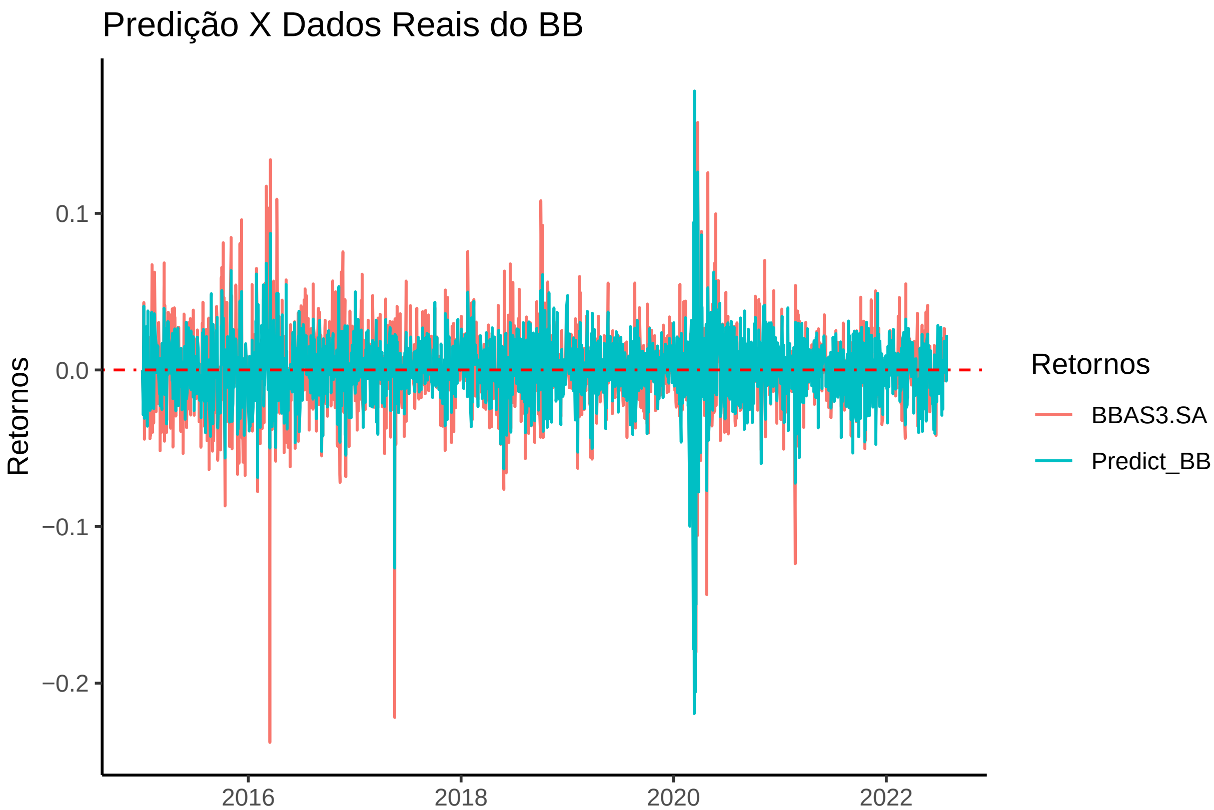
```
## `geom_smooth()` using formula 'y ~ x'
```




### Forecast - Predição do Retornos Esperados do BB

Vamos tentar agora gerar a predição com o modelo proposto.

```
real_data_bb = returns %>% select(date, BBAS3.SA)
predict_data_bb = data.frame(predict(bb_market_model))
  colnames(predict_data_bb) = c("Predict_BB")

data_bb = cbind(real_data_bb, predict_data_bb)


ggplot(data_bb, aes(x=date))+
  geom_line(aes(y = BBAS3.SA, colour = "BBAS3.SA")) +
  geom_line(aes(y = Predict_BB, colour = "Predict_BB")) +
  theme_classic() +
  geom_hline(yintercept = 0, color="red", linetype = "dotdash") +
  labs(x='', y='Retornos',
title='Predição X Dados Reais do BB',
color = "Retornos")
```



Predição X Dados Reais do BB
Retornos
0.1
0.0
−0.1
−0.2
2016
2018
2020
2022
Retornos
BBAS3.SA
Predict_BB