Visualização de Regressão Linear

Visualização dos dados de uma regressão - R

A EDA (*Exploratory Data Analysis*) é essencial para "materializar" os dados que estamos trabalhando para o leitor. O artigo de hoje é justamente para a visualização de dados em uma Regressão Linear simples.

Equação de uma regressão linear simples

```
## Bibliotecas utilizadas
library(ggside)
```

```
## Carregando pacotes exigidos: ggplot2
```

```
## Registered S3 method overwritten by 'ggside':
##   method from
##   +.gg   ggplot2
```

```
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v tibble  3.1.7     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
## v purrr   0.3.4
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(tidymodels)
```

```
## -- Attaching packages --------------------------------------- tidymodels 0.2.0 --
```

```
## v broom        0.8.0     v rsample      0.1.1
## v dials        0.1.1     v tune         0.2.0
## v infer        1.0.0     v workflows    0.2.6
## v modeldata    0.1.1     v workflowsets 0.2.1
## v parsnip      0.2.1     v yardstick    0.0.9
## v recipes      0.2.0
```

```
## -- Conflicts ---------------------------------------- tidymodels_conflicts() --
## x scales::discard() masks purrr::discard()
## x dplyr::filter()   masks stats::filter()
## x recipes::fixed()  masks stringr::fixed()
## x dplyr::lag()      masks stats::lag()
## x yardstick::spec() masks readr::spec()
## x recipes::step()   masks stats::step()
## * Learn how to get started at https://www.tidymodels.org/start/
```

```
library(tidyquant)
```

```
## Carregando pacotes exigidos: lubridate

##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union

## Carregando pacotes exigidos: PerformanceAnalytics

## Carregando pacotes exigidos: xts

## Carregando pacotes exigidos: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

##
## Attaching package: 'xts'

## The following objects are masked from 'package:dplyr':
##
##     first, last

##
## Attaching package: 'PerformanceAnalytics'

## The following object is masked from 'package:graphics':
##
##     legend

## Carregando pacotes exigidos: quantmod

## Carregando pacotes exigidos: TTR
```

```
##
## Attaching package: 'TTR'

## The following object is masked from 'package:dials':
##
##     momentum

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(readxl)
library(performance)
```

```
##
## Attaching package: 'performance'

## The following objects are masked from 'package:yardstick':
##
##     mae, rmse
```

Caso alguma biblioteca não esteja instalada, você utilizar o código abaixo:

install.packages("biblioteca")

Para a regressão utilizamos duas séries temporais, os histórico do índice Ibovespa e o índice S&P 500, ambos com histórico de 3 anos.

O preço é uma função dos retornos, neste sentido é lógico pensar em fazer a regressão entre as duas séries utilizando os retornos ao invés do valor absoluto do índice. Portanto vamos tratar os dados removendo os valores nulos e calcular os retornos.

```
## Importando dados
df <- read_excel('economatica.xlsx', skip = 3)
names(df) <- c('Data', 'Ibov', 'Sp500', 'Dolar') ## renomeando colunas

df <- df %>%
  transform(Ibov = as.numeric(Ibov),
            Sp500 = as.numeric(Sp500),
            Dolar = as.numeric(Dolar)) %>%
  na.omit(df) ## Removendo valores nulos
```

```
## Warning in eval(substitute(list(...)), `_data`, parent.frame()): NAs
## introduzidos por coerção

## Warning in eval(substitute(list(...)), `_data`, parent.frame()): NAs
## introduzidos por coerção

## Warning in eval(substitute(list(...)), `_data`, parent.frame()): NAs
## introduzidos por coerção
```

```r
## Calculando os retornos
df <- df %>%
  mutate(Ibov_ret = (Ibov/lag(Ibov)-1)*100,
         Sp500_ret = (Sp500/lag(Sp500)-1)*100,
         Dolar_ret = (Dolar/lag(Dolar)-1)*100)
```

Podemos agora plotar o gráfico de dispersão entre as duas séries.

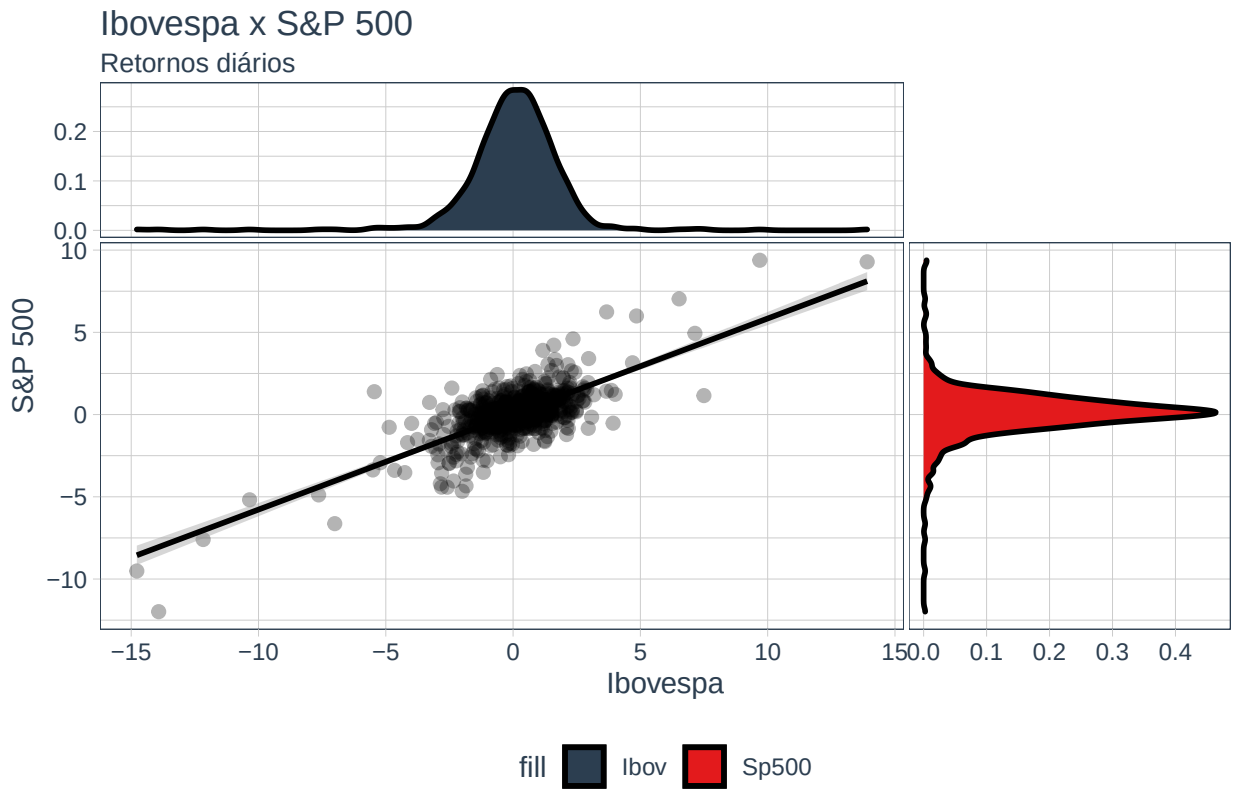
```r
df %>%
  ggplot(aes(x = Ibov_ret, y = Sp500_ret))+
  geom_point(size = 2, alpha = 0.30, position = 'jitter')+
  stat_smooth(method = 'lm', color = 'black')+
  geom_xsidedensity(aes(y = after_stat(density), fill = 'Ibov'),
                    alpha = 1.5,
                    size = 1,
                    position = 'stack')+
  geom_ysidedensity(aes(x = after_stat(density), fill = 'Sp500'),
                     alpha = 1.5,
                     size = 1,
                     position = 'stack')+
  scale_color_tq()+
  scale_fill_tq()+
  theme_tq()+
  theme(ggside.panel.scale.x = 0.4,
        ggside.panel.scale.y = 0.4)+
  labs(title = 'Ibovespa x S&P 500',
       subtitle = 'Retornos diários',
       x = 'Ibovespa',
       y = 'S&P 500',
       caption = 'Fonte: Economatica.  Elaboração própria')
```

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

## Warning: Removed 1 rows containing non-finite values (stat_smooth).

## Warning: Removed 1 rows containing non-finite values (stat_density).
## Removed 1 rows containing non-finite values (stat_density).

## Warning: Removed 1 rows containing missing values (geom_point).
```



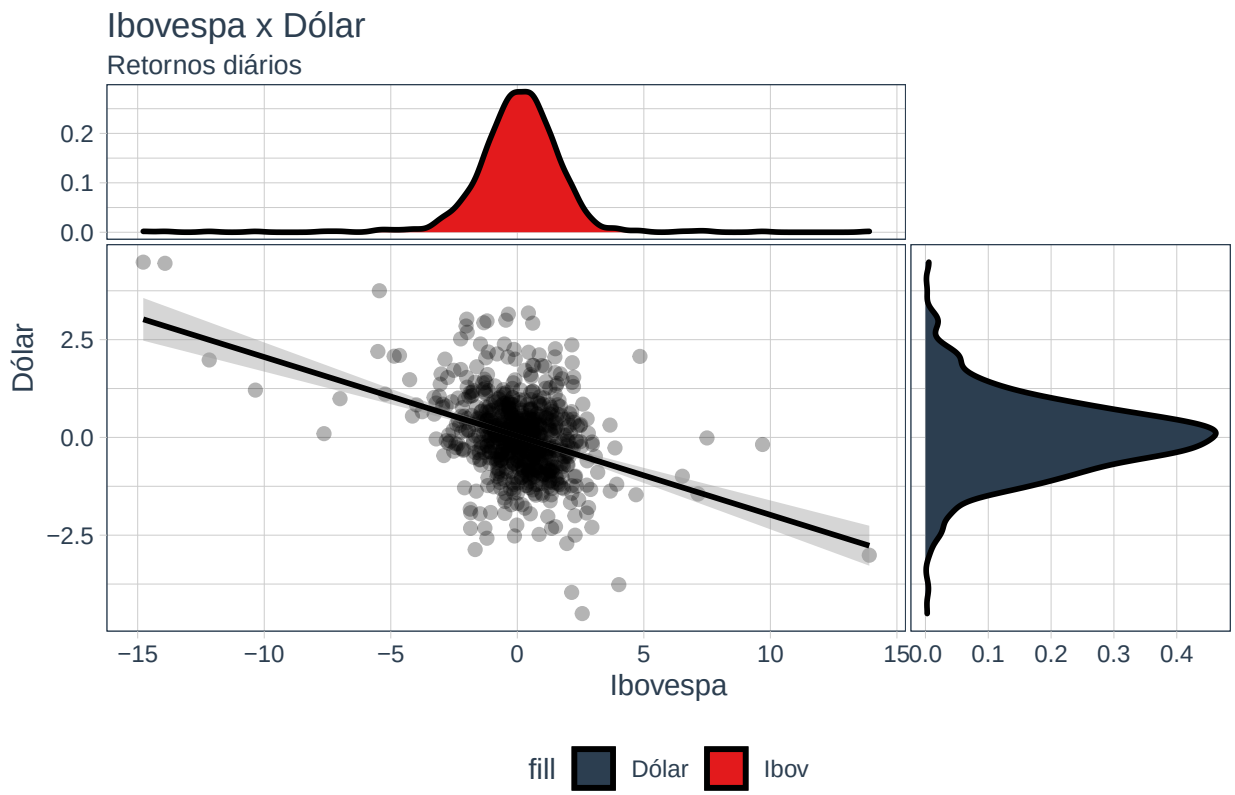

```
df %>%
  ggplot(aes(x = Ibov_ret, y = Dolar_ret))+
  geom_point(size = 2, alpha = 0.30, position = 'jitter')+
  stat_smooth(method = 'lm', color = 'black')+
  geom_xsidedensity(aes(y = after_stat(density), fill = 'Ibov'),
                    alpha = 1.5,
                    size = 1,
                    position = 'stack')+
  geom_ysidedensity(aes(x = after_stat(density), fill = 'Dólar'),
                     alpha = 1.5,
                     size = 1,
                     position = 'stack')+
  scale_color_tq()+
  scale_fill_tq()+
  theme_tq()+
  theme(ggside.panel.scale.x = 0.4,
        ggside.panel.scale.y = 0.4)+
  labs(title = 'Ibovespa x Dólar',
       subtitle = 'Retornos diários',
       x = 'Ibovespa',
       y = 'Dólar',
       caption = 'Fonte: Economatica.  Elaboração própria')
```

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

## Warning: Removed 1 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 1 rows containing non-finite values (stat_density).
## Removed 1 rows containing non-finite values (stat_density).

## Warning: Removed 1 rows containing missing values (geom_point).
```

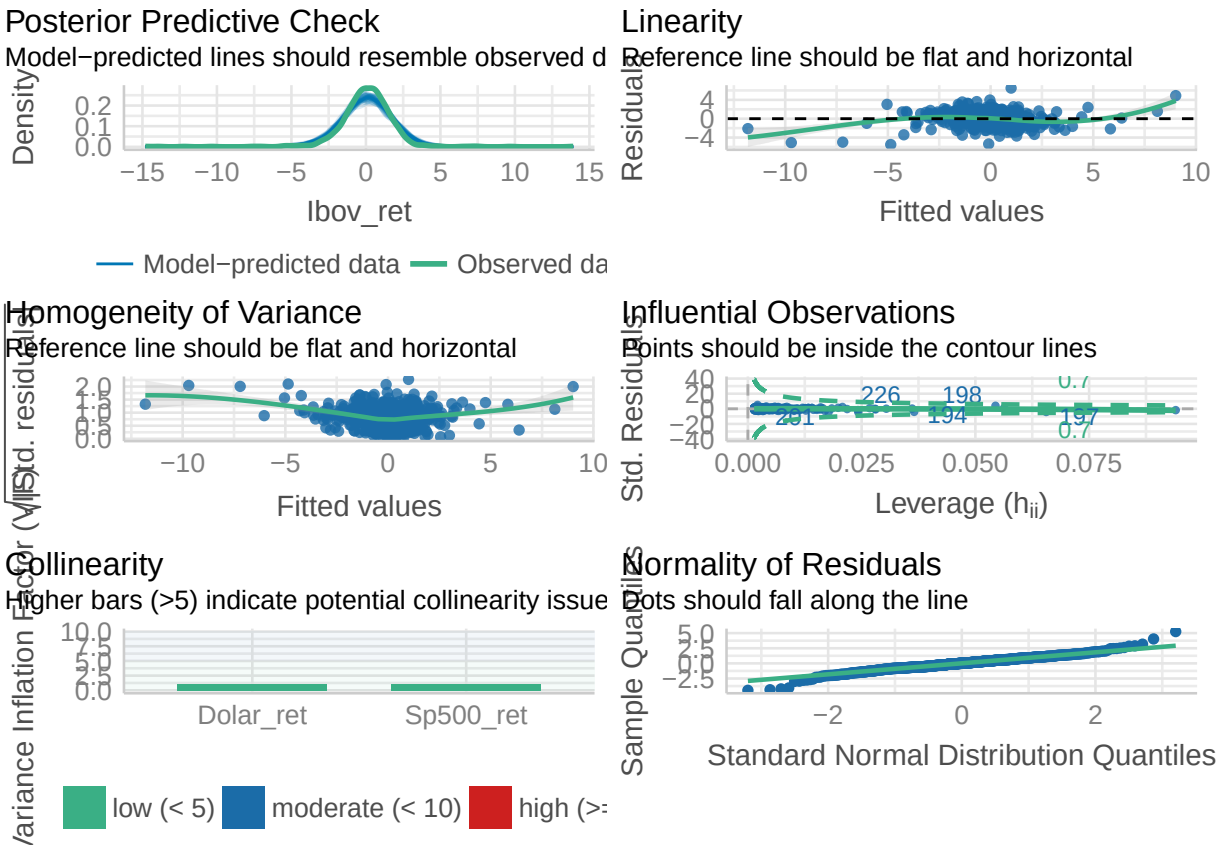


Fonte: Economatica. Elaboração própria

Agora podemos de fato rodar uma regressão entre as duas variáveis.

```
model_ln <- linear_reg() %>%
  set_engine('lm') %>%
  fit(Ibov_ret~Sp500_ret+Dolar_ret, data = df)

check_model(model_ln)
```


Posterior Predictive Check
Model−predicted lines should resemble observed d
Density
0.2
0.1
0.0
−15
−10
−5
0
5
10
15
Ibov_ret
Model−predicted data
Observed da
Linearity
Reference line should be flat and horizontal
Residuals
4
0
−4
−10
−5
0
5
10
Fitted values
Homogeneity of Variance
Reference line should be flat and horizontal
Std. residuals
2.0
1.5
1.0
0.5
0.0
−10
−5
0
5
10
Fitted values
Influential Observations
Points should be inside the contour lines
Std. Residuals
40
20
0
−20
−40
226
198
291
194
197
0.7
0.000
0.025
0.050
0.075
Leverage (hii)
Collinearity
Higher bars (>5) indicate potential collinearity issue
Variance Inflation Factor (VIF)
10.0
7.5
5.0
2.5
0.0
Dolar_ret
Sp500_ret
low (< 5)
moderate (< 10)
high (>
Normality of Residuals
Dots should fall along the line
Sample Quantiles
5.0
2.5
0.0
−2.5
−2
0
2
Standard Normal Distribution Quantiles