

Analise Estatística de Bancos

Carlos E. Carvalho

1/31/2021

Neste documento serão analisados os preços de fechamento das ações de 5 bancos durante o ano de 2020. São eles: Banco do Brasil, Santander, Bradesco, Itaú e Banrisul. O objetivo é mostrar a variação do preço das ações durante o ano, apresentar alguns valores estatísticos do preço das ações e verificar se existe alguma relação entre os bancos.

LinkedIn: <https://www.linkedin.com/in/carlos-carvalho-93204b13/> Github: <https://github.com/CarlosCarvalho1981/AnaliseEstatisticaBancos>

Definição do diretório de trabalho e carregamento das bibliotecas necessárias.

```
setwd("D:/CIENTISTA_DADOS/BANCOS")
getwd()
```

```
## [1] "D:/CIENTISTA_DADOS/BANCOS"
```

```
#http://www.quantmod.com/
```

```
#Carrega todos os pacotes
```

```
library(quantmod)
```

```
## Loading required package: xts
```

```
## Loading required package: zoo
```

```
##
```

```
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
```

```
##
```

```
## as.Date, as.Date.numeric
```

```
## Loading required package: TTR
```

```
## Registered S3 method overwritten by 'quantmod':
```

```
## method from
```

```
## as.zoo.data.frame zoo
```

```
library(xts)
library(moments)
library(readr)
library(rvest)
```

```
## Loading required package: xml2
```

```
##
```

```
## Attaching package: 'rvest'
```

```
## The following object is masked from 'package:readr':
```

```
##
```

```
##      guess_encoding
```

```
library(stringr)
library(dplyr)
```

```
##
```

```
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:xts':
```

```
##
```

```
##      first, last
```

```
## The following objects are masked from 'package:stats':
```

```
##
```

```
##      filter, lag
```

```
## The following objects are masked from 'package:base':
```

```
##
```

```
##      intersect, setdiff, setequal, union
```

```
library(lubridate)
```

```
##
```

```
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
```

```
##
```

```
##      date, intersect, setdiff, union
```

```
library(ggplot2)
library(psych)
```

```
##
```

```
## Attaching package: 'psych'
```

```
## The following objects are masked from 'package:ggplot2':
```

```
##
```

```
##      %+%, alpha
```

Aqui são definidas as datas para a coleta do valor das ações. Neste caso estão sendo selecionados todos os dias de 2020, desconsiderando o dia primeiro de Janeiro e 31 de Dezembro.

```
startDate <- as.Date("2020-01-02")
endDate <- as.Date("2022-12-30")
```

Esse vetor indica quais bancos serão pesquisados. É importante verificar no yahoo finanças (<https://br.financas.yahoo.com/>) o ticker correto para cada ação que será pesquisada.

```
bancos <- c("BBAS3.SA", "BCSA34.SA", "BBDC4.SA", "ITUB4.SA", "BRSR6.SA")
```

Para criar a estrutura do data frame com a quantidade correta de linhas, utiliza-se a estrutura coletada da primeira ação, neste caso, Banco do Brasil.

```
ticker <- bancos[1]
dfBancos <- getSymbols(ticker, src = "yahoo", from = startDate, to = endDate, auto.assign = F)
```

```
## 'getSymbols' currently uses auto.assign=TRUE by default, but will
## use auto.assign=FALSE in 0.5-0. You will still be able to use
## 'loadSymbols' to automatically load data. getOption("getSymbols.env")
## and getOption("getSymbols.auto.assign") will still be checked for
## alternate defaults.
```

```
##
## This message is shown once per session and may be disabled by setting
## options("getSymbols.warning4.0"=FALSE). See ?getSymbols for details.
```

```
## Warning: BBAS3.SA 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.
```

Em seguida é criado o gráfico de velas (candlechart) para mostrar a variação do preço dos papéis da empresa. Esse arquivo será salvo em memória e para gerar o nome correto do arquivo de forma automática é necessário executar as seguintes instruções:

1 - Cria-se uma string com a extensão do arquivo:

```
extensao <- ".png"
```

2 - Os nomes das ações (ticker) tem os caracteres “.SA” no final. Retira-se esses caracteres para que não criem problemas no nome do arquivo:

```
Completo <- str_replace(ticker, "\\.", "") #Retira o ponto (.)
Completo
```

```
## [1] "BBAS3SA"
```

```
Completo <- str_replace(Completo, "SA", "") #Retira o SA
Completo
```

```
## [1] "BBAS3"
```

3 - Junta-se o nome da empresa (sem o .SA) com a extensão:

```
Completo <- paste(Completo, extensao)
Completo
```

```
## [1] "BBAS3 .png"
```

4 - Retira-se o espaço que ficou entre as duas strings:

```
Completo <- str_replace(Completo, "\\s", "")
Completo
```

```
## [1] "BBAS3.png"
```

5 - Agora cria-se o gráfico, salvando-o na memória com o nome correto.

```
#png(filename = Completo, width = 1200, height = 600, res = 80)
candleChart(dfBancos, name = ticker)
```



```
#dev.off()
```

Para que o arquivo seja salvo na memória, é necessário descomentar as linhas 79 e 81.

Agora extrai-se apenas a coluna com o preço de fechamento em cada dia.

```
dfBancos <- dfBancos[,4]
```

E então capta-se os valores das ações de todos os outros bancos (ou empresas) que estão no vetor bancos. Esse processo é feito em um loop, captando todos os valores, extraindo apenas a coluna com o preço de fechamento de cada dia, organizando o nome do gráfico para salvar na memória e criando o gráfico.

```
for (i in 2:length(bancos)){
  ticker <- bancos[i]
  obj <- getSymbols(ticker, src = "yahoo", from = startDate, to = endDate, auto.assign = F)

  #Retira o ponto (.) do ticker
  Completo <- str_replace(ticker, "\\.", "")
  Completo
  #Retira o SA do nome
  Completo <- str_replace(Completo, "SA", "")
  Completo
  #Junta o nome com a extensão .png
  Completo <- paste(Completo, extensao)
  Completo
  #Retira o espaço do nome
  Completo <- str_replace(Completo, "\\s", "")
  Completo

  #Gera a figura
  #png(filename = Completo, width = 1200, height = 600, res = 80) - Retire o comentário para salvar o a
  candleChart(obj, name = ticker)
  #dev.off() - Retire o comentário para salvar o arquivo em memória

  obj1 <- as.vector(obj[,4])
  dfBancos <- cbind(dfBancos, obj1)
  Sys.sleep(0.3) #É importante esperar um tempo para cada requisição, senão o site pode bloquear
}
```

```
## Warning: BCSA34.SA 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.
```

```
## Warning: BBDC4.SA 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.
```



```
## Warning: ITUB4.SA 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.
```



```
## Warning: BRSR6.SA 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.
```

ITUB4.SA

[2020-01-02/2021-01-29]





Observa-se como ficou o dataset

```
head(dfBancos)
```

```
##          BBAS3.SA.Close  obj1  obj1.1 obj1.2 obj1.3
## 2020-01-02          53.80 16.77 34.10000 38.03 22.24
## 2020-01-03          53.71 17.00 34.10000 37.63 22.09
## 2020-01-06          53.00 17.12 33.49091 37.07 22.05
## 2020-01-07          52.60 16.90 32.90909 36.21 22.15
## 2020-01-08          52.12 16.88 32.40000 35.62 21.67
## 2020-01-09          51.06 16.93 31.88182 34.91 21.39
```

Altera-se o nome das colunas para que seja mais fácil de identificar.

```
colnames(dfBancos) <- c("BancoBrasil", "Santander", "Bradesco", "Itau", "Banrisul")
```

Retira-se os valores faltantes (NA) das linhas.

```
dfBancos <- dfBancos[complete.cases(dfBancos), ]
```

A partir desse ponto, observam-se algumas medidas estatísticas para cada empresa. A instrução a seguir mostra os valores mínimos e máximos, todos os quartis e a média para o preço de fechamento dos papéis de cada banco. Essas medidas são conhecidas como medidas de tendência central.

```
summary(dfBancos)
```

```
##      Index      BancoBrasil      Santander      Bradesco
## Min.   :2020-01-02 Min.   :22.13 Min.   :10.00 Min.   :16.06
## 1st Qu.:2020-04-08 1st Qu.:30.82 1st Qu.:11.77 1st Qu.:20.48
## Median :2020-07-15 Median :33.62 Median :12.73 Median :21.97
## Mean   :2020-07-15 Mean   :35.41 Mean   :13.74 Mean   :23.33
## 3rd Qu.:2020-10-19 3rd Qu.:37.40 3rd Qu.:16.22 3rd Qu.:26.36
## Max.   :2021-01-29 Max.   :53.80 Max.   :18.97 Max.   :34.10
##      Itau      Banrisul
## Min.   :20.52 Min.   :10.82
## 1st Qu.:23.69 1st Qu.:12.57
## Median :26.42 Median :13.62
## Mean   :27.21 Mean   :14.42
## 3rd Qu.:30.66 3rd Qu.:14.63
## Max.   :38.03 Max.   :22.24
```

Também é importante observar o desvio padrão e a variância dos valores.

```
#Desvio padrão
apply(dfBancos,2, sd)
```

```
## BancoBrasil  Santander  Bradesco      Itau  Banrisul
##    7.087705    2.431678    4.090909    4.152875    2.778837
```

```
#Variância
apply(dfBancos,2, var)
```

```
## BancoBrasil  Santander  Bradesco      Itau  Banrisul
##   50.235560    5.913056   16.735539   17.246368   7.721933
```

Pelo valor do desvio padrão é possível ver que o Banco do Brasil tem o maior valor, indicando que foram os papéis com a maior variação entre os bancos pesquisados, durante o ano. O coeficiente de variação, calculado abaixo, mostra essa variação em termos percentuais em relação à média. Com isso é possível fazer uma avaliação mais apropriada.

```
#Coeficiente de Variação
CV <- function(x){
  return ((sd(x)/mean(x))*100)
}

apply(dfBancos,2,CV)
```

```
## BancoBrasil  Santander  Bradesco      Itau  Banrisul
##   20.01873    17.69257   17.53410   15.26218   19.27604
```

Com isso, é possível perceber que, em relação à média, os papéis do Banco do Brasil foram os que mais variaram durante o ano, seguidos de perto pelo Banrisul.

As medidas de dispersão são outra ferramenta estatística que se pode utilizar para avaliar o comportamento das amostras. O coeficiente de assimetria representa o quanto a distribuição está próxima de uma distribuição normal (em relação a horizontal).

```
apply(dfBancos, 2, skewness)
```

```
## BancoBrasil Santander Bradesco Itau Banrisul
## 1.0937061 0.4849168 0.7071677 0.5221489 1.4547515
```

Um coeficiente de assimetria menor do que zero indica que a distribuição está deslocada para a esquerda. Se o coeficiente for maior do que zero, a distribuição está deslocada para a direita. A distribuição normal tem o coeficiente de assimetria igual a zero.

Já a curtose indica a variação da distribuição no eixo vertical.

```
apply(dfBancos, 2, kurtosis)
```

```
## BancoBrasil Santander Bradesco Itau Banrisul
## 3.343560 1.824855 2.576787 2.160203 4.049307
```

Para continuar a análise estatística observando a relação entre as variáveis, é necessário transformar o dataset em um objeto do tipo dataframe e inserir uma coluna de índice numérica.

```
dfBancos <- as.data.frame(dfBancos)
dfBancos <- dfBancos %>% mutate(id = row_number())
rownames(dfBancos) <- dfBancos$id
dfBancos$id <- NULL
head(dfBancos)
```

```
## BancoBrasil Santander Bradesco Itau Banrisul
## 1 53.80 16.77 34.10000 38.03 22.24
## 2 53.71 17.00 34.10000 37.63 22.09
## 3 53.00 17.12 33.49091 37.07 22.05
## 4 52.60 16.90 32.90909 36.21 22.15
## 5 52.12 16.88 32.40000 35.62 21.67
## 6 51.06 16.93 31.88182 34.91 21.39
```

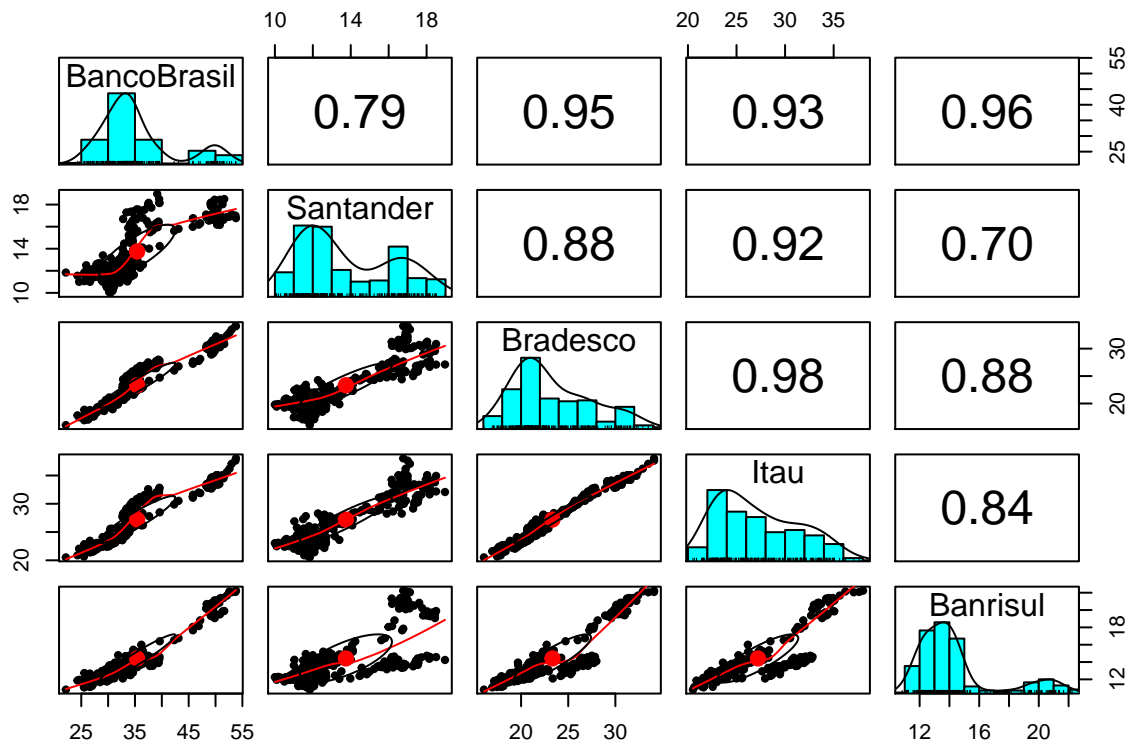
A tabela a seguir mostra a correlação entre o preço de fechamento dos papéis dos bancos analisados, em 2020. É possível ver que, aparentemente, existem algumas correlações entre as empresas. Essas correlações podem ser apenas coincidência. Lembrando também que correlação não significa causalidade. Ou seja, a correlação não indica que um papel influenciou no outro ou que essa correlação vai acontecer sempre.

```
cor(as.data.frame(dfBancos[c("BancoBrasil", "Santander", "Bradesco", "Itau", "Banrisul"])))
```

```
## BancoBrasil Santander Bradesco Itau Banrisul
## BancoBrasil 1.0000000 0.7944910 0.9533271 0.9257719 0.9584999
## Santander 0.7944910 1.0000000 0.8843192 0.9166207 0.6958777
## Bradesco 0.9533271 0.8843192 1.0000000 0.9842941 0.8759966
## Itau 0.9257719 0.9166207 0.9842941 1.0000000 0.8428838
## Banrisul 0.9584999 0.6958777 0.8759966 0.8428838 1.0000000
```

A figura a seguir ajuda a visualizar essas correlações:

```
pairs.panels(dfBancos[c("BancoBrasil", "Santander", "Bradesco", "Itau", "Banrisul")])
```

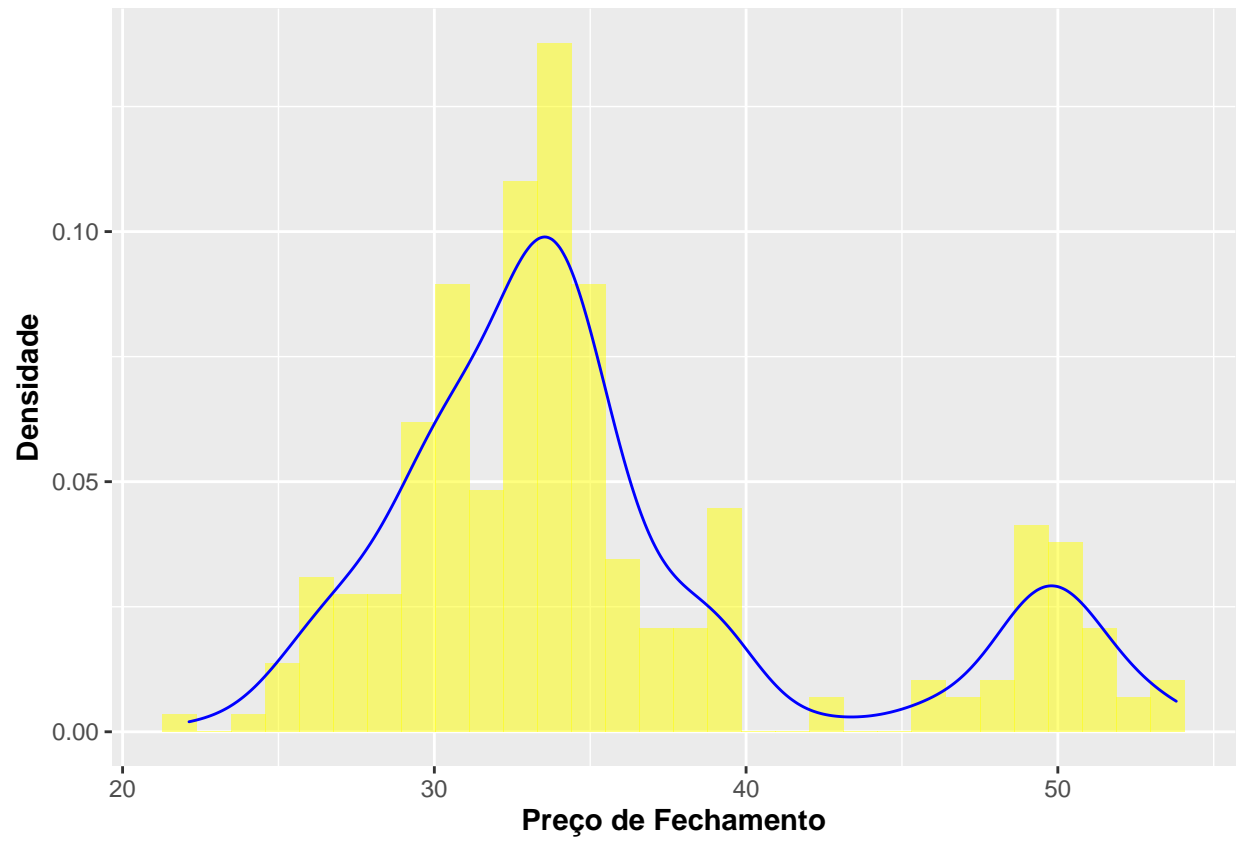


Para finalizar, observa-se um histograma e um boxplot para cada um dos bancos analisados:

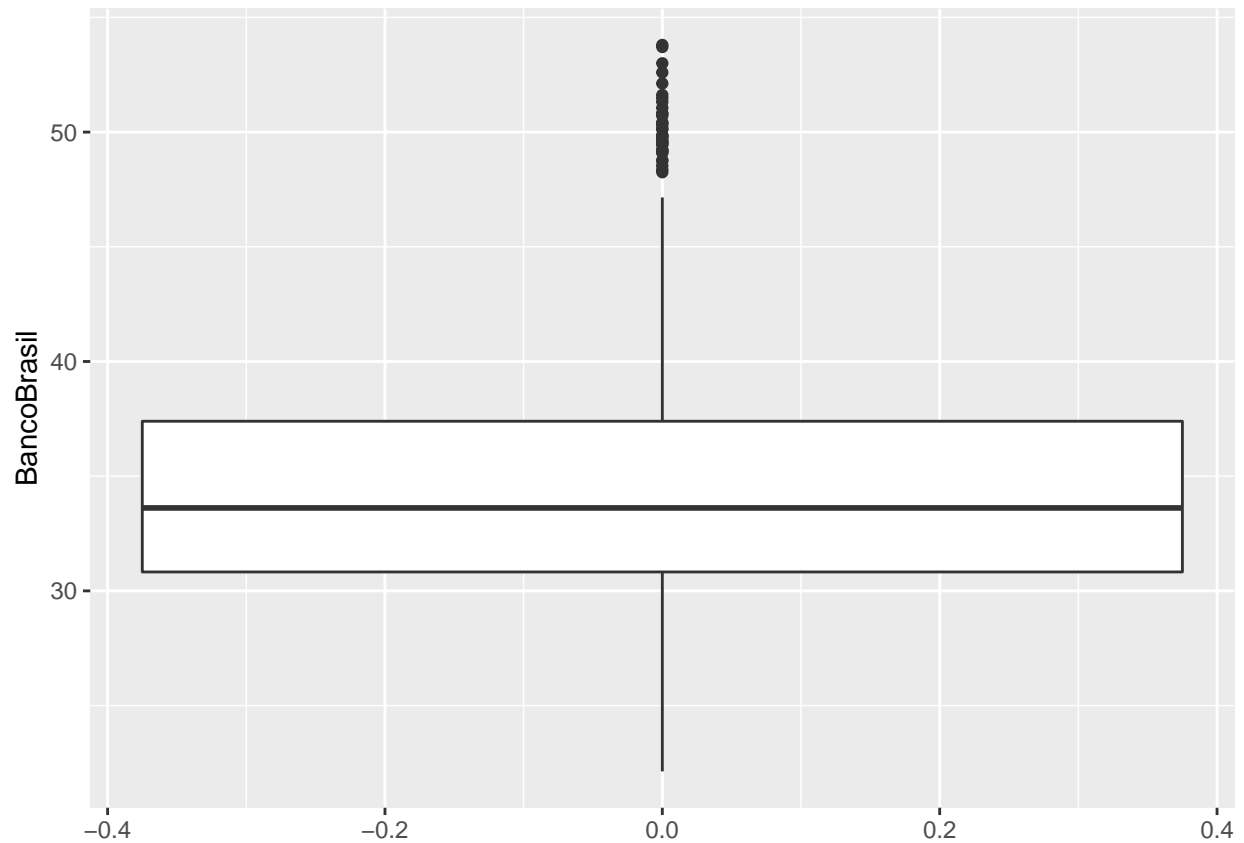
1 - Banco do Brasil:

```
ggplot(dfBancos, aes(x = BancoBrasil), binwidth = 30) +
  geom_histogram(aes(y = ..density..), fill = "yellow1", alpha = 0.5) +
  geom_density(colour = "blue") + xlab(expression(bold("Preço de Fechamento"))) +
  ylab(expression(bold("Densidade")))
```

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



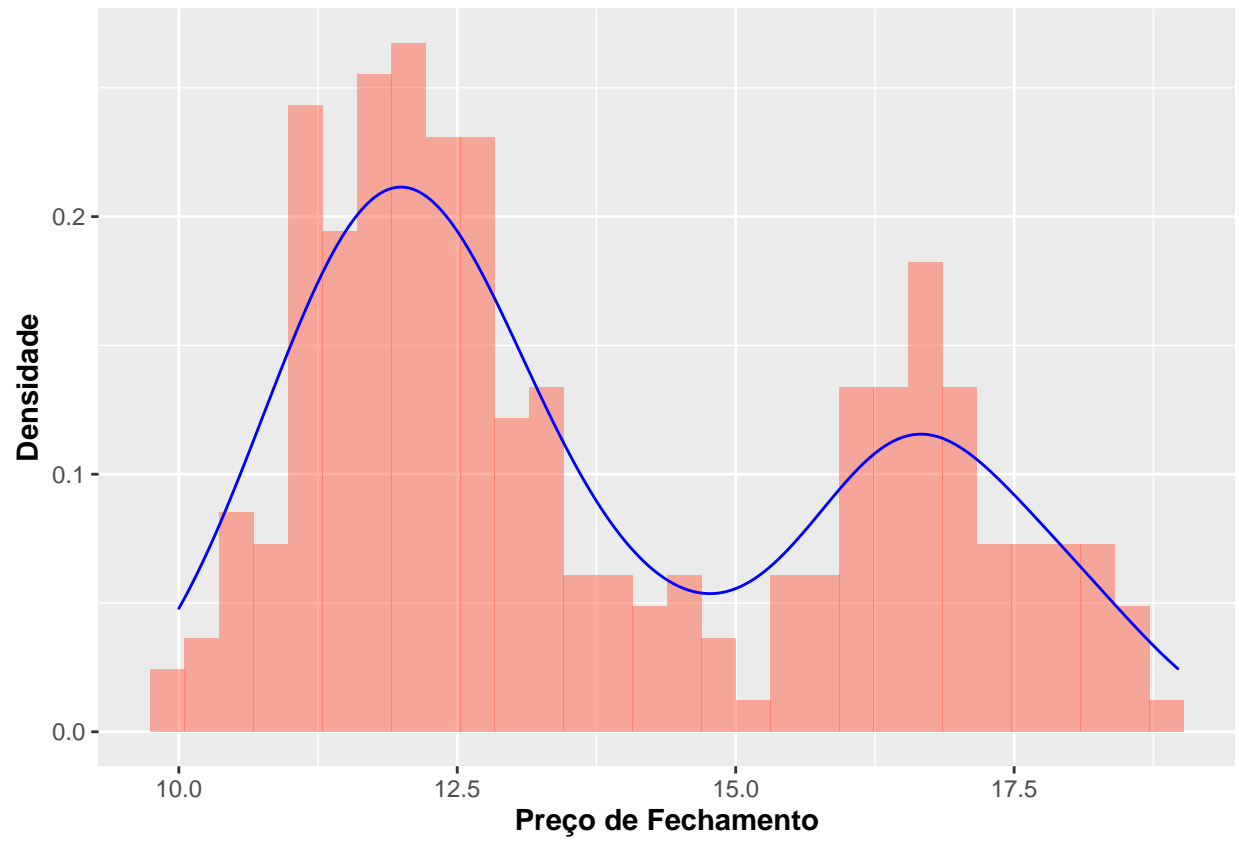
```
ggplot(dfBancos, aes(y = BancoBrasil)) + geom_boxplot()
```



2 - Santander:

```
ggplot(dfBancos, aes(x = Santander), binwidth = 30) +
  geom_histogram(aes(y = ..density..), fill = "tomato", alpha = 0.5) +
  geom_density(colour = "blue") + xlab(expression(bold("Preço de Fechamento"))) +
  ylab(expression(bold("Densidade")))
```

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



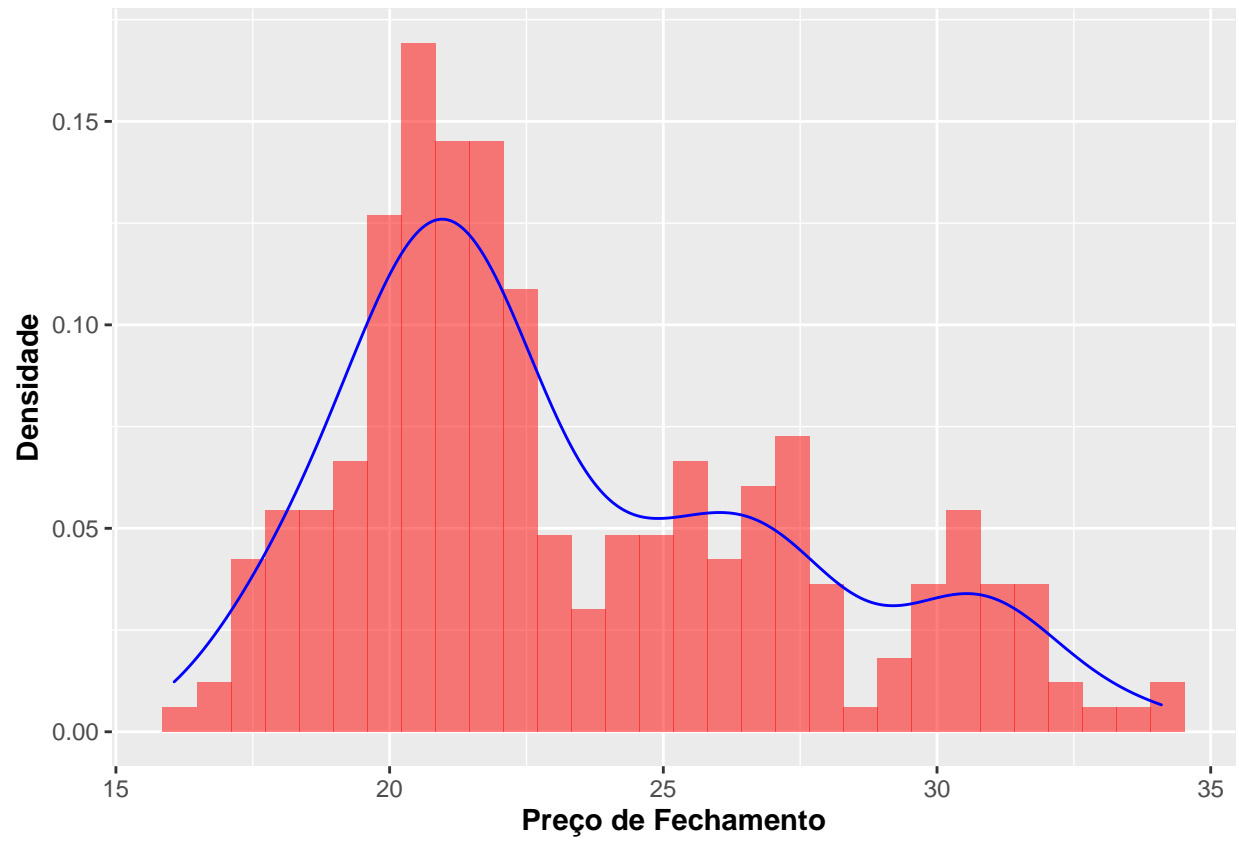
```
ggplot(dfBancos, aes(y = Santander)) + geom_boxplot()
```



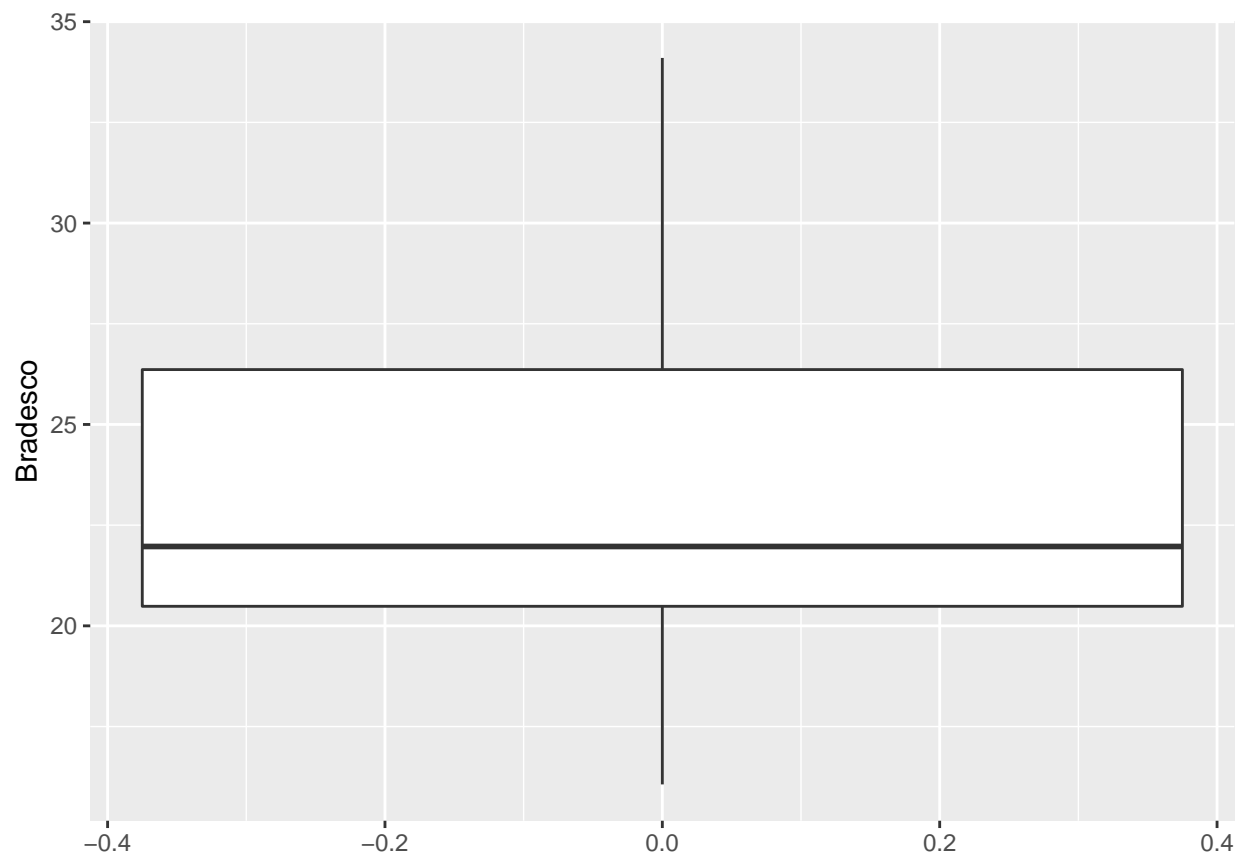
3 - Bradesco:

```
ggplot(dfBancos, aes(x = Bradesco), binwidth = 30) +  
  geom_histogram(aes(y = ..density..), fill = "red", alpha = 0.5)+  
  geom_density(colour = "blue") + xlab(expression(bold("Preço de Fechamento")))+  
  ylab(expression(bold("Densidade")))
```

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



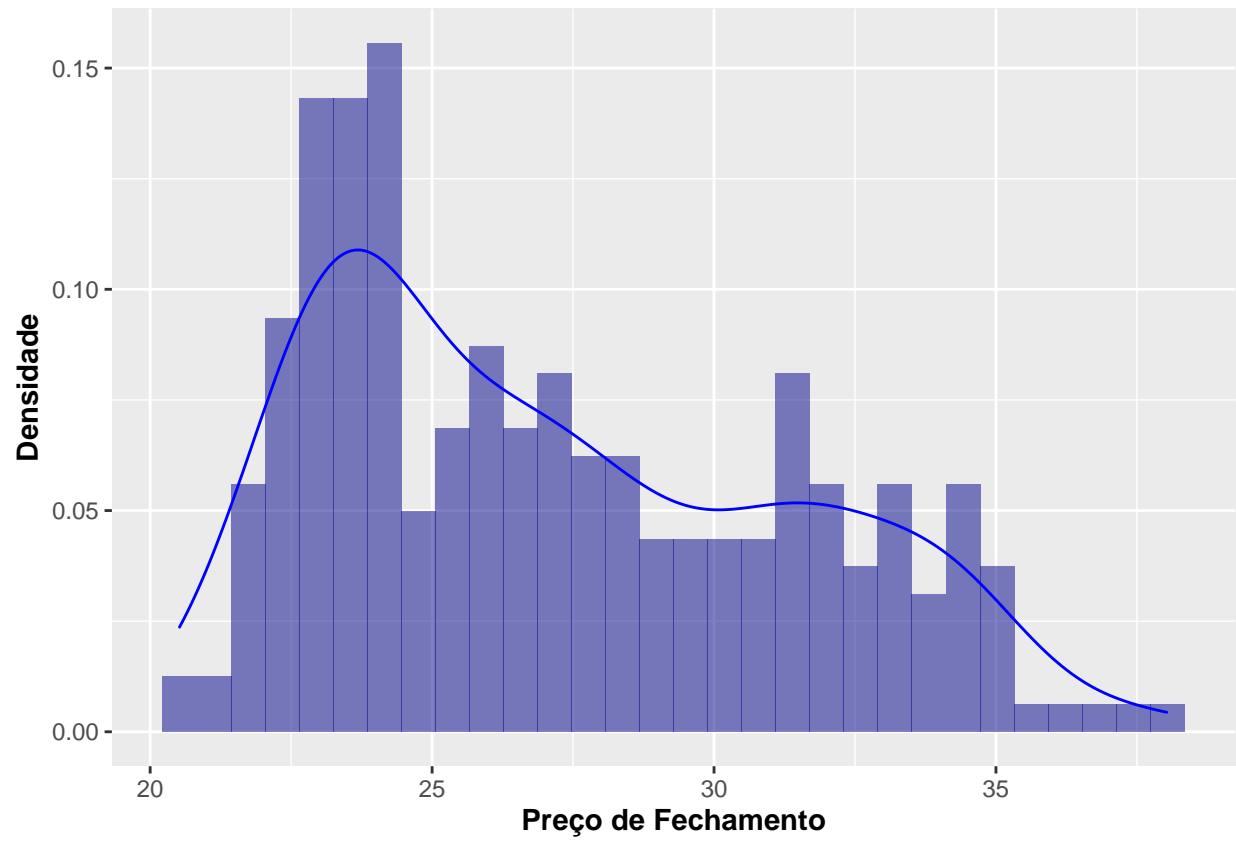
```
ggplot(dfBancos, aes(y = Bradesco)) + geom_boxplot()
```



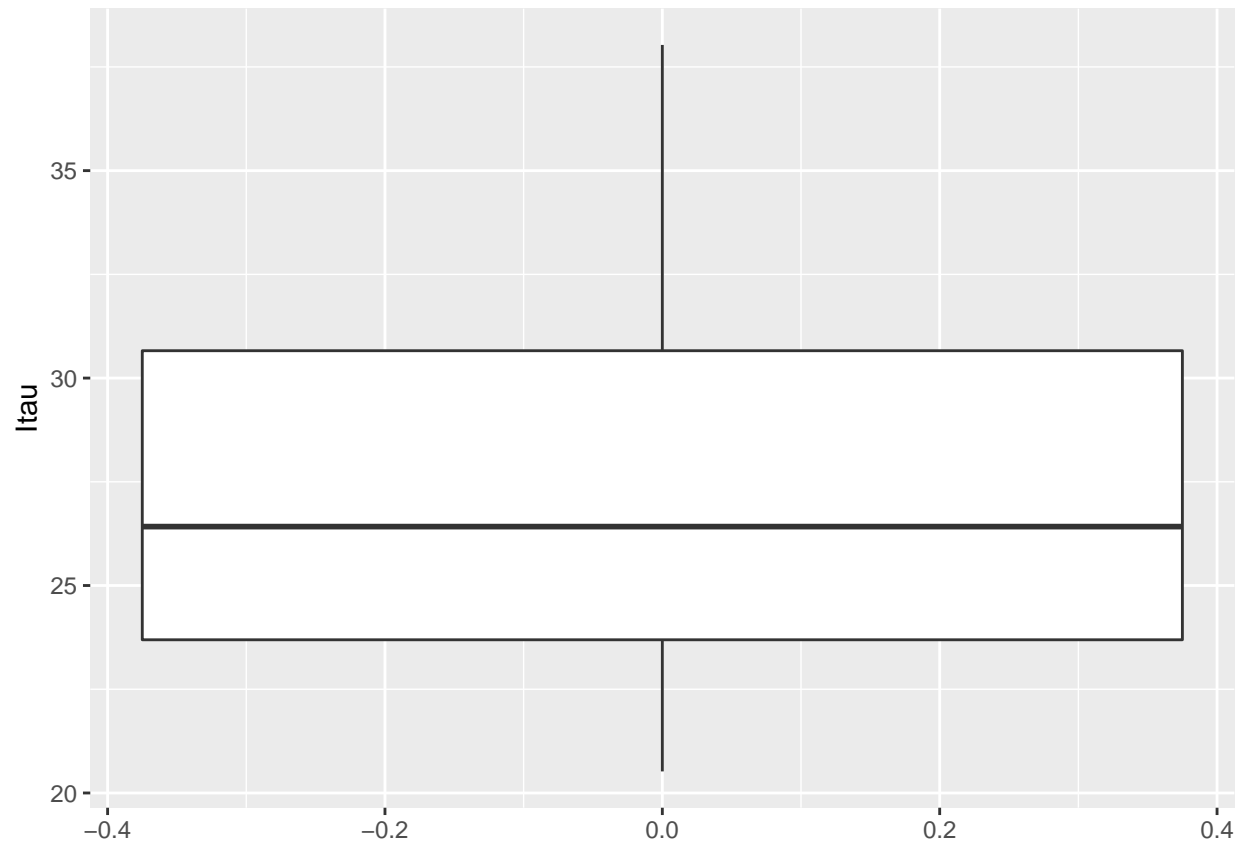
4 - Itaú:

```
ggplot(dfBancos, aes(x = Itau), binwidth = 30) +  
  geom_histogram(aes(y = ..density..), fill = "blue4", alpha = 0.5) +  
  geom_density(colour = "blue") + xlab(expression(bold("Preço de Fechamento"))) +  
  ylab(expression(bold("Densidade")))
```

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



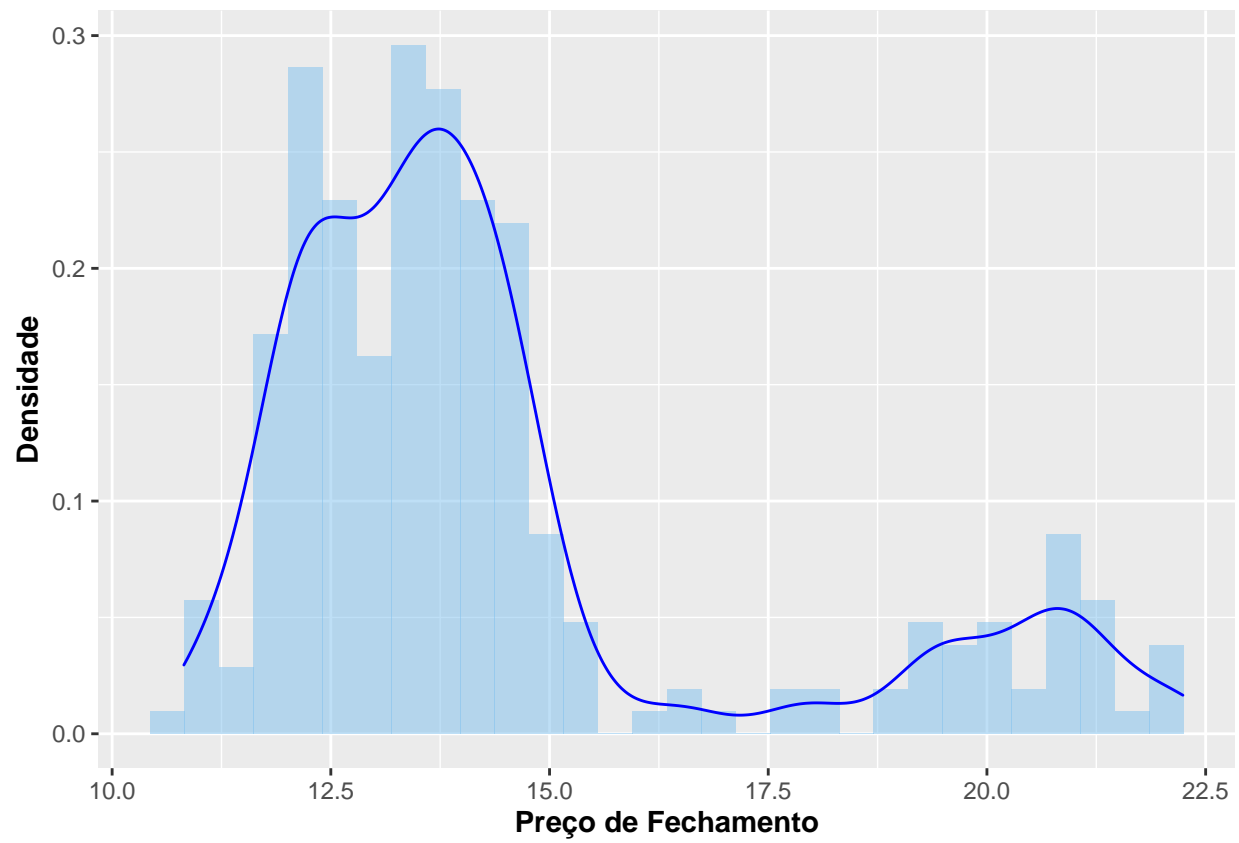
```
ggplot(dfBancos, aes(y = Itaú)) + geom_boxplot()
```



5 - Banrisul:

```
ggplot(dfBancos, aes(x = Banrisul), binwidth = 30) +  
  geom_histogram(aes(y = ..density..), fill = "skyblue2", alpha = 0.5)+  
  geom_density(colour = "blue") + xlab(expression(bold("Preço de Fechamento")))+  
  ylab(expression(bold("Densidade")))
```

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



```
ggplot(dfBancos, aes(y = Banrisul)) + geom_boxplot()
```

