R version 4.3.0 (2023-04-21 ucrt) -- "Already Tomorrow"

Copyright (C) 2023 The R Foundation for Statistical Computing

Platform: x86\_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.

You are welcome to redistribute it under certain conditions.

Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.

Type 'contributors()' for more information and

'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or

'help.start()' for an HTML browser interface to help.

Type 'q()' to quit R.

[Workspace loaded from C:/files/projects/programacao/python/acoes\_data/.RData]

> load("C:/files/projects/programacao/python/acoes\_data/.RData")

> library(tidyverse)

── **Attaching core tidyverse packages** ─────────────────────────────────────────────── tidyverse 2.0.0 ──

✔ dplyr 1.1.2 ✔ readr 2.1.4

✔ forcats 1.0.0 ✔ stringr 1.5.0

✔ ggplot2 3.4.2 ✔ tibble 3.2.1

✔ lubridate 1.9.2 ✔ tidyr 1.3.0

✔ purrr 1.0.1

── **Conflicts** ───────────────────────────────────────────────────────────────── tidyverse\_conflicts() ──

✖ dplyr::filter() masks stats::filter()

✖ dplyr::lag() masks stats::lag()

ℹ Use the conflicted package to force all conflicts to become errors

Warning message:

package ‘lubridate’ was built under R version 4.3.1

> library(gridExtra)

Attaching package: ‘gridExtra’

The following object is masked from ‘package:dplyr’:

combine

> library(ggplot2)

> library(readxl)

>

> setwd("C:/files/projects/programacao/python/acoes\_data/")

>

> df\_tri = read.csv("data/acoesb3.csv")

>

> df\_day = read.csv("data/acoesb3cot.csv")

>

> #função para removendo valores infinitos

>

> rmv\_inf\_values\_row <- function(df) {

+ rows\_with\_inf <- apply(df, 1, function(row) any(is.infinite(row)))

+ df <- df[!rows\_with\_inf, ]

+ return(df)

+ }

>

> # função para remover valores NA

>

> rmv\_na\_val <- function(df) {

+ incomplete\_rows <- !complete.cases(df)

+

+ if (any(incomplete\_rows)) {

+ df <- df[complete.cases(df), ]

+ }

+

+ return(df)

+ }

>

> #coletando as datas dos últimos balanços

> ult\_bal\_dates = c()

> for (i in 1:nrow(df\_tri)){

+ if (df\_tri[i,"ultBal"] %in% ult\_bal\_dates){}

+ else{

+ ult\_bal\_dates = c(ult\_bal\_dates, df\_tri[i,"ultBal"])

+ }

+ }

>

> #retirando os valores das empresas que tem os trimestres vigentes diferentes

> #de 31/03, 30/06, 30/09 e 31/12

> for (t in ult\_bal\_dates) {

+ if (endsWith(t, '03-31')){} else if(endsWith(t, '06-30')){}

+ else if(endsWith(t, '09-30')){} else if(endsWith(t, '12-31')){}else {

+ ult\_bal\_dates = ult\_bal\_dates[-match(t, ult\_bal\_dates)]

+ }

+ }

>

> #separando os df por períodos

>

> #lista para conter os períodos

> trimestres = list()

>

> #lista para conter os critérios por períodos

> crit\_tri = list()

>

> qtd\_dias = 6\*30 #qtd de dias de variação

>

> #criando os valores dos critérios

> for (per in 1:length(ult\_bal\_dates)){

+ print(paste('Criando o DF para o periodo: ', ult\_bal\_dates[per]))

+

+

+ #Criando as variáveis do db trimestral

+ ult\_bal = codigo = roic = cres\_rec5 = divb = disp = ativc = c();

+ ativ = patl = recl12 = ebit12 = Lucl12 = recl3 = ebit3 = Lucl3 = ultIns = c()

+

+ #loop através das linhas do db trimestral

+ for (i in 1:nrow(df\_tri)){

+

+ #escolhendo o trimestre a ser trabalhado

+ if (df\_tri[i,"ultBal"] == ult\_bal\_dates[per]){

+

+ ult\_bal = c(ult\_bal, df\_tri[i,"ultBal"])

+ ultIns = c(ultIns, df\_tri[i, "ultInsert"])

+ codigo = c(codigo, df\_tri[i,"codigo"])

+ roic = c(roic, df\_tri[i,"roic"])

+ cres\_rec5 = c(cres\_rec5, df\_tri[i,"cresRec5a"])

+ divb = c(divb, df\_tri[i,"divBruta"])

+ disp = c(disp, df\_tri[i,"disponib"])

+ ativc = c(ativc, df\_tri[i,"ativCirc"])

+ ativ = c(ativ, df\_tri[i,"ativos"])

+ patl = c(patl, df\_tri[i,"patLiq"])

+ recl12 = c(recl12, df\_tri[i,"recLiq12m"])

+ ebit12 = c(ebit12, df\_tri[i,"ebit12m"])

+ Lucl12 = c(Lucl12, df\_tri[i,"LucLiq12m"])

+ recl3 = c(recl3, df\_tri[i,"recLiq3m"])

+ ebit3 = c(ebit3, df\_tri[i, "ebit3m"])

+ Lucl3 = c(Lucl3, df\_tri[i, "LucLiq3m"])

+ }

+ }

+

+ #df trimestral das demonstrações

+ df\_tri\_dem = data.frame(ult\_bal, roic, cres\_rec5,

+ divb, disp, ativc, ativ, patl, recl12,

+ ebit12, Lucl12, recl3, ebit3, Lucl3, ultIns,

+ row.names = codigo)

+

+ #criando as variaveis do db diário

+

+ ult\_cot = cotAtual = cod = divY = n\_ac = cotF = n\_acF = c()

+

+ # loop nas linhas do dbday

+ for (i in 1:nrow(df\_day)){

+

+ # obtendo o número da posição do codigo atual no df\_tri\_dem

+ j\_dem = match(df\_day[i,"cod"], row.names(df\_tri\_dem))

+ if (is.na(j\_dem)){}

+ else {

+ # verificando se o codigo da ação já pertence a variavel cod

+ if (df\_day[i,"cod"] %in% cod){

+

+ #Se pertence:

+ # obtendo o número da posição desse código no cod

+ j = match(df\_day[i,"cod"], cod)

+

+ # verificando se o valor da ultima cotação da iteração atual é

+ # menor ou igual a data de inserção + dias

+ if (as.Date((df\_day[i,'ultCot'])) <= as.Date(ultIns)[j\_dem]+qtd\_dias){

+

+ # verificando se o valor da ultima cotação da iteração atual é maior que

+ #o valor pertencente à variável e menor ou igual a data de inserção

+ if (as.Date(df\_day[i,"ultCot"]) > as.Date(df\_day[j,"ultCot"]) &&

+ as.Date(df\_day[i,'ultCot']) <= as.Date(ultIns[j\_dem])){

+

+ #se for maior: retirando o valor de cada variável correspondente à

+ #posição do valor pertencente

+ ult\_cot = ult\_cot[-j]; cod = cod[-j]; cotAtual = cotAtual[-j];

+ divY = divY[-j]; n\_ac = n\_ac[-j]; cotF = cotF[-j]; n\_acF = n\_acF[-j]

+

+ #colocando os valores que teve a maior ultCot (preço atualizado)

+ ult\_cot = c(ult\_cot, df\_day[i,"ultCot"])

+ cod = c(cod, df\_day[i,"cod"])

+ cotAtual = c(cotAtual, df\_day[i,"cotAtual"])

+ divY = c(divY, df\_day[i, "divYield"])

+ n\_ac = c(n\_ac, df\_day[i, "nAcoes"])

+ cotF = c(cotF, df\_day[i, "cotAtual"])

+ n\_acF = c(n\_acF, df\_day[i, "nAcoes"])

+ } else {

+ cotF[j] = df\_day[i, "cotAtual"]

+

+ n\_acF[j] = df\_day[i, "nAcoes"]

+ }

+

+ }

+

+

+ } else if (as.Date(df\_day[i,'ultCot']) <= as.Date(ultIns[j\_dem])) {

+

+ #se não pertence e é menor que o período:

+ #apenas colocando o valor da iteração atual

+ ult\_cot = c(ult\_cot, df\_day[i,"ultCot"])

+ cod = c(cod, df\_day[i,"cod"])

+ cotAtual = c(cotAtual, df\_day[i,"cotAtual"])

+ divY = c(divY, df\_day[i, "divYield"])

+ n\_ac = c(n\_ac, df\_day[i, "nAcoes"])

+ cotF = c(cotF, df\_day[i, "cotAtual"])

+ n\_acF = c(n\_acF, df\_day[i, "nAcoes"])

+ }

+ }

+ }

+

+

+ # ordenando as variáves antes de se unirem ao trimestral

+ ult\_cot\_t = cotAtual\_t = divY\_t = n\_ac\_t = cotF\_t = n\_acF\_t = c()

+

+ #loop através das linhas do db trimestral das demonstrações

+ for (i in 1:nrow(df\_tri\_dem)){

+

+ #encontrando a posição do codigo atual na linha

+ j = match(row.names(df\_tri\_dem)[i], cod)

+

+ ult\_cot\_t = c(ult\_cot\_t, ult\_cot[j])

+ cotAtual\_t = c(cotAtual\_t, cotAtual[j])

+ divY\_t = c(divY\_t, divY[j])

+ n\_ac\_t = c(n\_ac\_t, n\_ac[j])

+ cotF\_t = c(cotF\_t, cotF[j])

+ n\_acF\_t = c(n\_acF\_t, n\_acF[j])

+ }

+

+ df\_tri\_dem$ult\_cot\_t = ult\_cot\_t; df\_tri\_dem$cotAtual\_t = cotAtual\_t

+ df\_tri\_dem$divY\_t = divY\_t; df\_tri\_dem$n\_ac\_t = n\_ac\_t

+ df\_tri\_dem$cotF\_t = cotF\_t; df\_tri\_dem$n\_acF\_t = n\_acF\_t

+

+

+ #gridExtra::grid.table(df\_tri\_3t22 %>% slice(1:20))

+

+ arr\_ind = 3 #casas decimais dos indices

+

+ # criando os índices

+ lpa = round(Lucl12/n\_ac\_t, arr\_ind)

+ lpa3 = round(as.numeric(Lucl3)/n\_ac\_t, arr\_ind)

+ vpa = round(patl/n\_ac\_t, arr\_ind)

+ cx\_a = round(disp/n\_ac\_t, arr\_ind)

+ ativc\_a = round(ativc/n\_ac\_t, arr\_ind)

+ ativ\_a = round(ativ/n\_ac\_t, arr\_ind)

+ divb\_a = round(divb/n\_ac\_t, arr\_ind)

+ ebit\_a = round(ebit12/n\_ac\_t, arr\_ind)

+ ebit\_a3 = round(as.numeric(ebit3)/n\_ac\_t, arr\_ind)

+ dividend = round((as.numeric(divY\_t)/100)\*cotAtual\_t, arr\_ind)

+ rec\_a = round(recl12/n\_ac\_t, arr\_ind)

+ rec\_a3 = round(as.numeric(recl3)/n\_ac\_t, arr\_ind)

+ cot = cotAtual\_t

+

+

+ crit = data.frame(lpa, lpa3, vpa, cx\_a, ativc\_a, ativ\_a, divb\_a, ebit\_a,

+ ebit\_a3, dividend, rec\_a, rec\_a3, cot,

+ row.names = codigo)

+

+ ############# filtrando o db ###################################

+

+

+

+ crit = rmv\_inf\_values\_row(crit)

+

+

+ crit <- rmv\_na\_val(crit)

+

+

+ #passando o df para a lista

+ trimestres[[per]] = df\_tri\_dem

+

+

+ colnames(crit) = c("LPA", 'LPA (tri)', "VPA",

+ "Caixa/Ação", "Ativos Circulantes/Ação",

+ "Ativos/Ação", "Dív Bruta/Ação", "EBIT/Ação",

+ 'EBIT/Ação (tri)',

+ "Dividendos",

+ 'Receita/ Ação',

+ 'Receita/Ação (tri)',

+ 'Preço da Ação')

+

+ #gridExtra::grid.table(crit %>% slice(1:20))

+

+ crit\_tri[[per]] = crit

+ print(paste('DF do periodo: ', ult\_bal\_dates[per], ' criado'))

+ }

[1] "Criando o DF para o periodo: 2022-03-31"

[1] "DF do periodo: 2022-03-31 criado"

[1] "Criando o DF para o periodo: 2022-06-30"

[1] "DF do periodo: 2022-06-30 criado"

[1] "Criando o DF para o periodo: 2022-09-30"

[1] "DF do periodo: 2022-09-30 criado"

[1] "Criando o DF para o periodo: 2022-12-31"

[1] "DF do periodo: 2022-12-31 criado"

[1] "Criando o DF para o periodo: 2023-03-31"

[1] "DF do periodo: 2023-03-31 criado"

[1] "Criando o DF para o periodo: 2023-06-30"

[1] "DF do periodo: 2023-06-30 criado"

[1] "Criando o DF para o periodo: 2023-09-30"

[1] "DF do periodo: 2023-09-30 criado"

[1] "Criando o DF para o periodo: 2023-12-31"

[1] "DF do periodo: 2023-12-31 criado"

Warning messages:

1: NAs introduced by coercion

2: NAs introduced by coercion

3: NAs introduced by coercion

4: NAs introduced by coercion

>

> ################################################################################

> ######### função para retornar as variações dos preços ##############

> ################################################################################

>

> #per\_i => periodo a ser analisado

> addVarPrice = function(per\_i){

+

+ var\_price = c()

+

+ for (r in 1:nrow(crit\_tri[[per\_i]])) {

+

+ # encontrando a posição da linha do codigo da iteração no df das demonstrações

+ j\_dem = match(row.names(crit\_tri[[per\_i]])[r], row.names(trimestres[[per\_i]]))

+

+ # encontrando a posição da linha do codigo da iteração no df dos criterios

+ # do período anterior

+ j\_dem\_bef = match(row.names(crit\_tri[[per\_i]])[r], row.names(crit\_tri[[per\_i-1]]))

+

+ # criando o df das variacoes dos indices

+

+ price\_i = trimestres[[per\_i]][j\_dem,'cotAtual\_t']

+ price\_f = trimestres[[per\_i]][j\_dem,'cotF\_t']\*

+ (trimestres[[per\_i]][j\_dem,'n\_acF\_t']/trimestres[[per\_i]][j\_dem, 'n\_ac\_t'])

+

+ var\_price = c(var\_price, round((price\_f - price\_i)/price\_i, 4))

+

+ }

+

+ # adicionando as variações dos precos ao df crit do periodo

+ crit\_tri[[per\_i]]$v\_price = var\_price

+

+ colnames(crit\_tri[[per\_i]]) = c("L/P", 'L/P (tri)', "VPA/P", "ROE" , 'ROE (tri)', "ROIC",

+ "(Caixa/Ação)/Preço", "(Ativos Circulantes/Ação)/Preço",

+ "(Ativos/Ação)/Preço", "Dív Bruta/Caixa", "Marg. EBIT",

+ 'Marg. EBIT (tri)', "Marg. Líquida", 'Marg. Líquida (tri)',

+ "Cresc. Rec. (5 Anos)", "Dividendyield", "Lynch",

+ 'Lynch (tri)', "Dív. Bruta/Lucro Mensal", 'PSR (invertido)',

+ 'PSR (invertido) (tri)', 'EBIT/P', 'EBIT/P (tri)',

+ 'EBIT/Ativo', 'EBIT/Ativo (tri)', 'Div Bruta/Patrimonio', "v\_price")

+

+ #removendo valores infinitos

+

+

+ rmv\_inf\_values\_row <- function(df) {

+ rows\_with\_inf <- apply(df, 1, function(row) any(is.infinite(row)))

+ df <- df[!rows\_with\_inf, ]

+ return(df)

+ }

+

+ crit\_tri[[per\_i]] = rmv\_inf\_values\_row(crit\_tri[[per\_i]])

+

+ #removando valore NA

+

+ rmv\_na\_val <- function(df) {

+ incomplete\_rows <- !complete.cases(df)

+

+ if (any(incomplete\_rows)) {

+ df <- df[complete.cases(df), ]

+ }

+

+ return(df)

+ }

+ crit\_tri[[per\_i]] <- rmv\_na\_val(crit\_tri[[per\_i]])

+

+ return(crit\_tri[[per\_i]])

+ }

>

>

>

> ############################################################################

> ############### função para retirar valores extremos #################

> ############################################################################

>

> # remover valores extremos

> rmv\_wild = function(df, fator\_multiplicativo) {

+

+ exib\_wild = function(dados, fator\_multiplicativo){

+ # Obter estatísticas do boxplot

+ stats <- boxplot.stats(dados)

+

+ # Calcular limites personalizados

+ limite\_inferior <- stats$stats[2] - fator\_multiplicativo \* IQR(dados, na.rm = T)

+ limite\_superior <- stats$stats[4] + fator\_multiplicativo \* IQR(dados, na.rm = T)

+

+ # Identificar outliers

+ outliers <- dados[dados < limite\_inferior | dados > limite\_superior]

+

+ # Exibir os valores dos outliers

+ return(outliers)

+ }

+

+ wild\_row = c('so', 'para','length', 'nao', 'ser', 'zero')

+

+ n\_iter = 0

+ while (length(wild\_row) != 0) {

+ wild\_row = c()

+ for (c in 1:ncol(df)) {

+ if (length(exib\_wild(df[,c], fator\_multiplicativo)) != 0){

+ for (i in exib\_wild(df[,c], fator\_multiplicativo)) {

+ #numero das linhas

+ j = match(i, df[,c])

+ df[j,c] = NA

+ if (row.names(df)[j] %in% wild\_row){} else {

+ wild\_row = c(wild\_row, row.names(df)[j])

+ }

+ }

+ }

+ }

+ rmv\_na\_val <- function(df) {

+ incomplete\_rows <- !complete.cases(df)

+

+ if (any(incomplete\_rows)) {

+ df <- df[complete.cases(df), ]

+ }

+

+ return(df)

+ }

+

+ print(paste('empresas removidas na iteração', n\_iter, ':'))

+ print(wild\_row)

+ n\_iter = n\_iter+1

+

+

+ df = rmv\_na\_val(df)

+ }

+

+ return(df)

+ }

>

>

>

> #escolhendo o periodo que sera analisado

> per\_choice = crit\_tri[[4]]

>

> # verificando NA values

> colSums(is.na(per\_choice))

LPA LPA (tri) VPA Caixa/Ação

0 0 0 0

Ativos Circulantes/Ação Ativos/Ação Dív Bruta/Ação EBIT/Ação

0 0 0 0

EBIT/Ação (tri) Dividendos Receita/ Ação Receita/Ação (tri)

0 0 0 0

Preço da Ação

0

>

> boxplot(scale(per\_choice), range = 3)

>

> per\_choice=rmv\_wild(per\_choice, 3)

[1] "empresas removidas na iteração 0 :"

[1] "BRPR3" "VALE3" "CGAS5" "COCE5" "SUZB3" "PETR4" "FESA4" "GSHP3" "MWET4" "RSUL4" "PLAS3"

[12] "EEEL4" "MTSA4" "PMAM3" "UNIP6" "MOAR3" "TKNO4" "MGEL4" "TEKA4" "JFEN3" "HETA4" "PDGR3"

[23] "MNDL3" "SNSY5" "ESTR4" "SLED4" "LIGT3" "BMKS3" "OSXB3" "HOOT4" "BDLL4" "CLSC4" "NEXP3"

[34] "ATMP3" "AVLL3" "GPIV33" "FRIO3" "ODER4" "OIBR4" "RPMG3" "SGPS3" "CTNM4" "AMER3" "PCAR3"

[45] "AZUL4" "MNPR3" "AHEB6" "RNEW4" "MSPA4" "CRDE3" "CEED4" "GOLL4" "HBTS5" "JOPA4" "TXRX4"

[56] "PEAB4" "FRTA3" "BRGE6" "PATI4" "MOVI3" "MRFG3" "BRKM6" "AMBP3" "CTKA4" "GFSA3" "CEPE6"

[67] "PSSA3" "GOAU4" "JBSS3" "SOND6" "AGRO3" "CSRN6" "CEEB5" "EKTR4" "VIVT3" "GGBR4" "CRPG6"

[78] "LEVE3" "CPFE3" "ENMT4" "CGRA4" "CEBR6" "DESK3" "UGPA3" "VBBR3" "CEDO4" "CEGR3" "ARZZ3"

[89] "BAUH4" "LUXM4" "NUTR3"

[1] "empresas removidas na iteração 1 :"

[1] "SLCE3" "TEND3" "BAHI3" "AURE3" "LOGN3" "CBAV3" "EPAR3" "FHER3" "LUPA3" "EQMA3B" "VSTE3"

[12] "LOGG3" "ELET6" "SBSP3" "NORD3" "SIMH3" "EMBR3" "TELB4" "MYPK3" "RDOR3" "CIEL3" "TUPY3"

[23] "AGXY3" "NEOE3" "TPIS3" "EQTL3" "AURA33" "APER3" "EGIE3" "ENBR3" "CSNA3" "GEPA4" "TAEE4"

[34] "PFRM3" "CALI3" "WLMM4" "RENT3"

[1] "empresas removidas na iteração 2 :"

[1] "TASA4" "JALL3" "RECV3" "EMAE4" "BEEF3" "RAPT4" "PRIO3" "ENAT3" "SMTO3" "CSAN3" "SCAR3" "CBEE3"

[13] "KEPL3" "CRFB3" "BRFS3" "ALLD3" "HYPE3"

[1] "empresas removidas na iteração 3 :"

[1] "AALR3" "SEER3" "BALM4" "LVTC3" "HBRE3" "ASAI3" "WEGE3" "ORVR3"

[1] "empresas removidas na iteração 4 :"

NULL

>

> boxplot(scale(per\_choice), range = 3)

>

>

> ################################################################################

> ############# Escolhendo os Índices ##########################

> ################################################################################

>

> ############## regressão ####################

>

> # Realize a regressão linear

> modelo\_regressao <- lm(per\_choice$`Preço da Ação`~per\_choice$LPA+

+ per\_choice$`LPA (tri)`+per\_choice$VPA+

+ per\_choice$`Caixa/Ação`+

+ per\_choice$`Ativos Circulantes/Ação`+

+ per\_choice$`Ativos/Ação`+per\_choice$`Dív Bruta/Ação`+

+ per\_choice$`EBIT/Ação`+per\_choice$`EBIT/Ação (tri)`+

+ per\_choice$Dividendos+per\_choice$`Receita/ Ação`+

+ per\_choice$`Receita/Ação (tri)`)

>

> # Exiba o sumário do modelo

> summary(modelo\_regressao)

Call:

lm(formula = per\_choice$`Preço da Ação` ~ per\_choice$LPA +

per\_choice$`LPA (tri)` + per\_choice$VPA + per\_choice$`Caixa/Ação` +

per\_choice$`Ativos Circulantes/Ação` + per\_choice$`Ativos/Ação` +

per\_choice$`Dív Bruta/Ação` + per\_choice$`EBIT/Ação` +

per\_choice$`EBIT/Ação (tri)` + per\_choice$Dividendos +

per\_choice$`Receita/ Ação` + per\_choice$`Receita/Ação (tri)`)

Residuals:

Min 1Q Median 3Q Max

-9.015 -2.842 -1.392 2.039 21.363

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 2.7948466 0.8004098 3.492 0.000629 \*\*\*

per\_choice$LPA 1.4583902 1.5560329 0.937 0.350127

per\_choice$`LPA (tri)` 0.4912967 2.7244489 0.180 0.857136

per\_choice$VPA 0.4343049 0.1940018 2.239 0.026640 \*

per\_choice$`Caixa/Ação` 1.3727786 0.3593677 3.820 0.000194 \*\*\*

per\_choice$`Ativos Circulantes/Ação` -0.2781248 0.1698418 -1.638 0.103597

per\_choice$`Ativos/Ação` -0.0005864 0.1429016 -0.004 0.996731

per\_choice$`Dív Bruta/Ação` -0.1973070 0.2008700 -0.982 0.327542

per\_choice$`EBIT/Ação` 1.8322835 1.3278201 1.380 0.169651

per\_choice$`EBIT/Ação (tri)` -2.6275502 3.1211433 -0.842 0.401200

per\_choice$Dividendos 1.6727175 1.8733415 0.893 0.373328

per\_choice$`Receita/ Ação` -0.7526803 0.3118679 -2.413 0.017000 \*

per\_choice$`Receita/Ação (tri)` 2.8710691 1.0887693 2.637 0.009239 \*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.176 on 151 degrees of freedom

Multiple R-squared: 0.4801, Adjusted R-squared: 0.4388

F-statistic: 11.62 on 12 and 151 DF, p-value: 2.424e-16

>

> # regressão com algumas variáveis retiradas (valor mais alto)

> model\_fit = lm(per\_choice$`Preço da Ação`~per\_choice$LPA+

+ per\_choice$`LPA (tri)`+per\_choice$VPA+

+ per\_choice$`Caixa/Ação`+

+ per\_choice$`Ativos Circulantes/Ação`+

+ per\_choice$`Dív Bruta/Ação`+

+ per\_choice$`EBIT/Ação`+per\_choice$`EBIT/Ação (tri)`+

+ per\_choice$Dividendos+per\_choice$`Receita/ Ação`+

+ per\_choice$`Receita/Ação (tri)`)

>

> summary(model\_fit)

Call:

lm(formula = per\_choice$`Preço da Ação` ~ per\_choice$LPA +

per\_choice$`LPA (tri)` + per\_choice$VPA + per\_choice$`Caixa/Ação` +

per\_choice$`Ativos Circulantes/Ação` + per\_choice$`Dív Bruta/Ação` +

per\_choice$`EBIT/Ação` + per\_choice$`EBIT/Ação (tri)` +

per\_choice$Dividendos + per\_choice$`Receita/ Ação` + per\_choice$`Receita/Ação (tri)`)

Residuals:

Min 1Q Median 3Q Max

-9.018 -2.842 -1.391 2.041 21.364

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 2.7946 0.7957 3.512 0.000585 \*\*\*

per\_choice$LPA 1.4605 1.4595 1.001 0.318564

per\_choice$`LPA (tri)` 0.4897 2.6861 0.182 0.855598

per\_choice$VPA 0.4337 0.1120 3.872 0.000160 \*\*\*

per\_choice$`Caixa/Ação` 1.3729 0.3576 3.839 0.000181 \*\*\*

per\_choice$`Ativos Circulantes/Ação` -0.2785 0.1456 -1.913 0.057648 .

per\_choice$`Dív Bruta/Ação` -0.1979 0.1295 -1.528 0.128552

per\_choice$`EBIT/Ação` 1.8307 1.2652 1.447 0.149963

per\_choice$`EBIT/Ação (tri)` -2.6251 3.0526 -0.860 0.391169

per\_choice$Dividendos 1.6717 1.8509 0.903 0.367850

per\_choice$`Receita/ Ação` -0.7525 0.3089 -2.436 0.015986 \*

per\_choice$`Receita/Ação (tri)` 2.8707 1.0804 2.657 0.008726 \*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.159 on 152 degrees of freedom

Multiple R-squared: 0.4801, Adjusted R-squared: 0.4425

F-statistic: 12.76 on 11 and 152 DF, p-value: < 2.2e-16

>

> # regressão com algumas variáveis retiradas (valor mais alto)

> model\_fit2 = lm(per\_choice$`Preço da Ação`~per\_choice$LPA+

+ per\_choice$VPA+

+ per\_choice$`Caixa/Ação`+

+ per\_choice$`Ativos Circulantes/Ação`+

+ per\_choice$`Dív Bruta/Ação`+

+ per\_choice$`EBIT/Ação`+per\_choice$`EBIT/Ação (tri)`+

+ per\_choice$Dividendos+per\_choice$`Receita/ Ação`+

+ per\_choice$`Receita/Ação (tri)`)

>

> summary(model\_fit2)

Call:

lm(formula = per\_choice$`Preço da Ação` ~ per\_choice$LPA +

per\_choice$VPA + per\_choice$`Caixa/Ação` + per\_choice$`Ativos Circulantes/Ação` +

per\_choice$`Dív Bruta/Ação` + per\_choice$`EBIT/Ação` +

per\_choice$`EBIT/Ação (tri)` + per\_choice$Dividendos +

per\_choice$`Receita/ Ação` + per\_choice$`Receita/Ação (tri)`)

Residuals:

Min 1Q Median 3Q Max

-8.907 -2.832 -1.366 2.069 21.379

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 2.7941 0.7931 3.523 0.000563 \*\*\*

per\_choice$LPA 1.6731 0.8751 1.912 0.057765 .

per\_choice$VPA 0.4322 0.1114 3.880 0.000155 \*\*\*

per\_choice$`Caixa/Ação` 1.3826 0.3525 3.922 0.000132 \*\*\*

per\_choice$`Ativos Circulantes/Ação` -0.2845 0.1413 -2.014 0.045773 \*

per\_choice$`Dív Bruta/Ação` -0.1914 0.1240 -1.543 0.124853

per\_choice$`EBIT/Ação` 1.6749 0.9299 1.801 0.073649 .

per\_choice$`EBIT/Ação (tri)` -2.2677 2.3322 -0.972 0.332421

per\_choice$Dividendos 1.6413 1.8375 0.893 0.373142

per\_choice$`Receita/ Ação` -0.7325 0.2878 -2.545 0.011902 \*

per\_choice$`Receita/Ação (tri)` 2.8091 1.0230 2.746 0.006760 \*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.143 on 153 degrees of freedom

Multiple R-squared: 0.48, Adjusted R-squared: 0.446

F-statistic: 14.12 on 10 and 153 DF, p-value: < 2.2e-16

>

> # regressão com algumas variáveis retiradas (valor mais alto)

> model\_fit3 = lm(per\_choice$`Preço da Ação`~per\_choice$LPA+

+ per\_choice$VPA+

+ per\_choice$`Caixa/Ação`+

+ per\_choice$`Ativos Circulantes/Ação`+

+ per\_choice$`Dív Bruta/Ação`+

+ per\_choice$`EBIT/Ação`+per\_choice$`EBIT/Ação (tri)`+

+ per\_choice$`Receita/ Ação`+

+ per\_choice$`Receita/Ação (tri)`)

>

> summary(model\_fit3)

Call:

lm(formula = per\_choice$`Preço da Ação` ~ per\_choice$LPA +

per\_choice$VPA + per\_choice$`Caixa/Ação` + per\_choice$`Ativos Circulantes/Ação` +

per\_choice$`Dív Bruta/Ação` + per\_choice$`EBIT/Ação` +

per\_choice$`EBIT/Ação (tri)` + per\_choice$`Receita/ Ação` +

per\_choice$`Receita/Ação (tri)`)

Residuals:

Min 1Q Median 3Q Max

-9.111 -2.817 -1.290 2.025 21.357

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 2.9783 0.7654 3.891 0.000148 \*\*\*

per\_choice$LPA 1.9646 0.8115 2.421 0.016641 \*

per\_choice$VPA 0.4304 0.1113 3.867 0.000162 \*\*\*

per\_choice$`Caixa/Ação` 1.3981 0.3519 3.974 0.000109 \*\*\*

per\_choice$`Ativos Circulantes/Ação` -0.2817 0.1412 -1.996 0.047704 \*

per\_choice$`Dív Bruta/Ação` -0.1993 0.1236 -1.612 0.108904

per\_choice$`EBIT/Ação` 1.7613 0.9242 1.906 0.058555 .

per\_choice$`EBIT/Ação (tri)` -2.2200 2.3300 -0.953 0.342189

per\_choice$`Receita/ Ação` -0.7246 0.2874 -2.521 0.012722 \*

per\_choice$`Receita/Ação (tri)` 2.7649 1.0212 2.708 0.007544 \*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.139 on 154 degrees of freedom

Multiple R-squared: 0.4773, Adjusted R-squared: 0.4467

F-statistic: 15.62 on 9 and 154 DF, p-value: < 2.2e-16

>

> # regressão com algumas variáveis retiradas (valor mais alto)

> model\_fit4 = lm(per\_choice$`Preço da Ação`~per\_choice$LPA+

+ per\_choice$VPA+

+ per\_choice$`Caixa/Ação`+

+ per\_choice$`Ativos Circulantes/Ação`+

+ per\_choice$`Dív Bruta/Ação`+

+ per\_choice$`EBIT/Ação`+

+ per\_choice$`Receita/ Ação`+

+ per\_choice$`Receita/Ação (tri)`)

>

> summary(model\_fit4)

Call:

lm(formula = per\_choice$`Preço da Ação` ~ per\_choice$LPA +

per\_choice$VPA + per\_choice$`Caixa/Ação` + per\_choice$`Ativos Circulantes/Ação` +

per\_choice$`Dív Bruta/Ação` + per\_choice$`EBIT/Ação` +

per\_choice$`Receita/ Ação` + per\_choice$`Receita/Ação (tri)`)

Residuals:

Min 1Q Median 3Q Max

-10.092 -2.889 -1.342 2.008 21.369

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 2.9711 0.7651 3.883 0.000152 \*\*\*

per\_choice$LPA 2.0362 0.8077 2.521 0.012717 \*

per\_choice$VPA 0.4363 0.1111 3.928 0.000129 \*\*\*

per\_choice$`Caixa/Ação` 1.3859 0.3515 3.943 0.000122 \*\*\*

per\_choice$`Ativos Circulantes/Ação` -0.2943 0.1405 -2.095 0.037813 \*

per\_choice$`Dív Bruta/Ação` -0.1849 0.1226 -1.508 0.133683

per\_choice$`EBIT/Ação` 1.1556 0.6706 1.723 0.086869 .

per\_choice$`Receita/ Ação` -0.6023 0.2571 -2.343 0.020419 \*

per\_choice$`Receita/Ação (tri)` 2.3004 0.8970 2.565 0.011282 \*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.138 on 155 degrees of freedom

Multiple R-squared: 0.4742, Adjusted R-squared: 0.4471

F-statistic: 17.47 on 8 and 155 DF, p-value: < 2.2e-16

>

> # regressão com algumas variáveis retiradas (valor mais alto)

> model\_fit5 = lm(per\_choice$`Preço da Ação`~per\_choice$LPA+

+ per\_choice$VPA+

+ per\_choice$`Caixa/Ação`+

+ per\_choice$`Ativos Circulantes/Ação`+

+ per\_choice$`EBIT/Ação`+

+ per\_choice$`Receita/ Ação`+

+ per\_choice$`Receita/Ação (tri)`)

>

> summary(model\_fit5)

Call:

lm(formula = per\_choice$`Preço da Ação` ~ per\_choice$LPA +

per\_choice$VPA + per\_choice$`Caixa/Ação` + per\_choice$`Ativos Circulantes/Ação` +

per\_choice$`EBIT/Ação` + per\_choice$`Receita/ Ação` +

per\_choice$`Receita/Ação (tri)`)

Residuals:

Min 1Q Median 3Q Max

-10.012 -2.880 -1.310 1.829 21.574

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 2.8222 0.7618 3.705 0.000293 \*\*\*

per\_choice$LPA 2.5045 0.7487 3.345 0.001029 \*\*

per\_choice$VPA 0.4206 0.1110 3.787 0.000217 \*\*\*

per\_choice$`Caixa/Ação` 1.3406 0.3517 3.812 0.000198 \*\*\*

per\_choice$`Ativos Circulantes/Ação` -0.3446 0.1370 -2.515 0.012916 \*

per\_choice$`EBIT/Ação` 0.7166 0.6066 1.181 0.239242

per\_choice$`Receita/ Ação` -0.5985 0.2582 -2.318 0.021724 \*

per\_choice$`Receita/Ação (tri)` 2.3177 0.9006 2.574 0.010997 \*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.159 on 156 degrees of freedom

Multiple R-squared: 0.4665, Adjusted R-squared: 0.4425

F-statistic: 19.49 on 7 and 156 DF, p-value: < 2.2e-16

>

> # regressão com algumas variáveis retiradas (valor mais alto)

> model\_fit6 = lm(per\_choice$`Preço da Ação`~per\_choice$LPA+

+ per\_choice$VPA+

+ per\_choice$`Caixa/Ação`+

+ per\_choice$`Ativos Circulantes/Ação`+

+ per\_choice$`Receita/ Ação`+

+ per\_choice$`Receita/Ação (tri)`)

>

> summary(model\_fit6)

Call:

lm(formula = per\_choice$`Preço da Ação` ~ per\_choice$LPA +

per\_choice$VPA + per\_choice$`Caixa/Ação` + per\_choice$`Ativos Circulantes/Ação` +

per\_choice$`Receita/ Ação` + per\_choice$`Receita/Ação (tri)`)

Residuals:

Min 1Q Median 3Q Max

-10.580 -3.041 -1.404 1.816 21.628

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 2.8548 0.7623 3.745 0.000253 \*\*\*

per\_choice$LPA 3.0932 0.5595 5.529 1.32e-07 \*\*\*

per\_choice$VPA 0.4378 0.1102 3.972 0.000108 \*\*\*

per\_choice$`Caixa/Ação` 1.4540 0.3387 4.292 3.08e-05 \*\*\*

per\_choice$`Ativos Circulantes/Ação` -0.3875 0.1323 -2.929 0.003908 \*\*

per\_choice$`Receita/ Ação` -0.5249 0.2508 -2.093 0.037988 \*

per\_choice$`Receita/Ação (tri)` 2.2088 0.8970 2.462 0.014880 \*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.165 on 157 degrees of freedom

Multiple R-squared: 0.4617, Adjusted R-squared: 0.4411

F-statistic: 22.44 on 6 and 157 DF, p-value: < 2.2e-16

>

> # regressão com algumas variáveis retiradas (valor mais alto)

> model\_fit7 = lm(per\_choice$`Preço da Ação`~per\_choice$LPA+

+ per\_choice$VPA+

+ per\_choice$`Caixa/Ação`+

+ per\_choice$`Ativos Circulantes/Ação`+

+ per\_choice$`Receita/Ação (tri)`)

>

> summary(model\_fit7)

Call:

lm(formula = per\_choice$`Preço da Ação` ~ per\_choice$LPA +

per\_choice$VPA + per\_choice$`Caixa/Ação` + per\_choice$`Ativos Circulantes/Ação` +

per\_choice$`Receita/Ação (tri)`)

Residuals:

Min 1Q Median 3Q Max

-9.107 -3.146 -1.353 2.253 21.670

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 2.8160 0.7701 3.657 0.000348 \*\*\*

per\_choice$LPA 3.1368 0.5650 5.552 1.17e-07 \*\*\*

per\_choice$VPA 0.3955 0.1095 3.612 0.000408 \*\*\*

per\_choice$`Caixa/Ação` 1.4210 0.3420 4.155 5.31e-05 \*\*\*

per\_choice$`Ativos Circulantes/Ação` -0.4021 0.1335 -3.011 0.003033 \*\*

per\_choice$`Receita/Ação (tri)` 0.4222 0.2781 1.518 0.130950

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.22 on 158 degrees of freedom

Multiple R-squared: 0.4467, Adjusted R-squared: 0.4292

F-statistic: 25.51 on 5 and 158 DF, p-value: < 2.2e-16

>

> # regressão com algumas variáveis retiradas (valor mais alto)

> model\_fit8 = lm(per\_choice$`Preço da Ação`~per\_choice$LPA+

+ per\_choice$VPA+

+ per\_choice$`Caixa/Ação`+

+ per\_choice$`Ativos Circulantes/Ação`)

>

> summary(model\_fit8)

Call:

lm(formula = per\_choice$`Preço da Ação` ~ per\_choice$LPA +

per\_choice$VPA + per\_choice$`Caixa/Ação` + per\_choice$`Ativos Circulantes/Ação`)

Residuals:

Min 1Q Median 3Q Max

-8.806 -3.393 -1.432 2.237 21.469

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 3.1065 0.7491 4.147 5.46e-05 \*\*\*

per\_choice$LPA 3.2643 0.5611 5.818 3.18e-08 \*\*\*

per\_choice$VPA 0.4131 0.1093 3.778 0.000223 \*\*\*

per\_choice$`Caixa/Ação` 1.3144 0.3360 3.911 0.000136 \*\*\*

per\_choice$`Ativos Circulantes/Ação` -0.2865 0.1101 -2.601 0.010179 \*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.242 on 159 degrees of freedom

Multiple R-squared: 0.4386, Adjusted R-squared: 0.4245

F-statistic: 31.06 on 4 and 159 DF, p-value: < 2.2e-16

# gráficos