

# Avaliando Ativo com distribuição GEV

Melquisadec

2024-07-02

```
if (!require("pacman")) install.packages("pacman")
```

```
## Carregando pacotes exigidos: pacman
```

```
p_load(tidyverse, xtable, extRemes, ggplot2, quantmod, Quandl, ecostats, latex2exp)
```

## Introdução Nosso interesse neste trabalho é a confiabilidade (ou probabilidade) stress-strength (abreviada por SSR) que, em termos gerais, consiste no estudo da probabilidade de falha de um sistema ou componente a partir da comparação de um stress (tensão) aplicado com um strength (resistência) do sistema. Sejam o stress  $Y$  e o strength  $X$  VAs contínuas e independentes, com função de densidade de probabilidade (PDF)  $f_Y$  e CDF  $F_X$ , respectivamente. A probabilidade SSR (também chamada confiabilidade SSR) é definida como

$$R = P(X < Y) = \int_{-\infty}^{\infty} F_X(x) f_Y(x) dx.$$

Ao aplicar essa metodologia SSR a dados financeiros do mundo real, poderíamos orientar um procedimento de seleção de ações calculando  $P(X < Y)$  quando  $X$  e  $Y$  representam retornos de duas diferentes ações. Em resumo, quando  $X$  e  $Y$  representam Va's de retorno e  $R < 1/2$ , o investidor deve escolher a ação da variável  $X$ . Se  $R > 1/2$ , ocorre o oposto. O caso  $R = 1/2$  é inconclusivo. O objetivo central do nosso trabalho foi propor a estimação da confiabilidade através de uma função  $\mathbb{H}$ , esta abordagem permite a estimação dos parâmetros sem a necessidade de impor restrições no espaço paramétrico da distribuição de  $X$  e  $Y$ . Para saber mais sobre a função  $\mathbb{H}$  recomendo a leitura do nosso paper o link estará disponível na publicação.

Para este Artigo já publicado carregamos 4 bases de dados de ações negociadas na bolsa de valores brasileiras os dados foram Ações do Banco do Brasil (BBAS3), Itaú unibanco (ITUB4), Mineradora Vale (VALE3) e varejista que atualmente foi incorporado pelo grupo Casas Bahia na época da análise a sigla do ativo era VIIA3. Os dados foram obtidos diretamente do site do yahoo.finance via comando quantmod descrito abaixo.

```
# BASE 1 BANCO DO BRASIL
```

```
BBAS3 <- quantmod::getSymbols("BBAS3.SA", src = "yahoo", auto.assign = FALSE, from = '2022-01-01',  
                             to = '2023-04-30', return.class = 'xts')
```

```
#BASE 2 ITAU UNIBANCO
```

```
ITAU <- quantmod::getSymbols("ITUB4.SA", src = "yahoo", auto.assign = FALSE, from = '2022-01-01',  
                             to = '2023-04-30', return.class = 'xts')
```

```
# BASE 3 MINERADORA VALE
```

```
VALE <- quantmod::getSymbols("VALE3.SA", src = "yahoo", auto.assign = FALSE, from = '2022-01-01',  
                             to = '2023-04-30', return.class = 'xts')
```

```
#BASE 4 GRUPO CASAS BAHIA
```

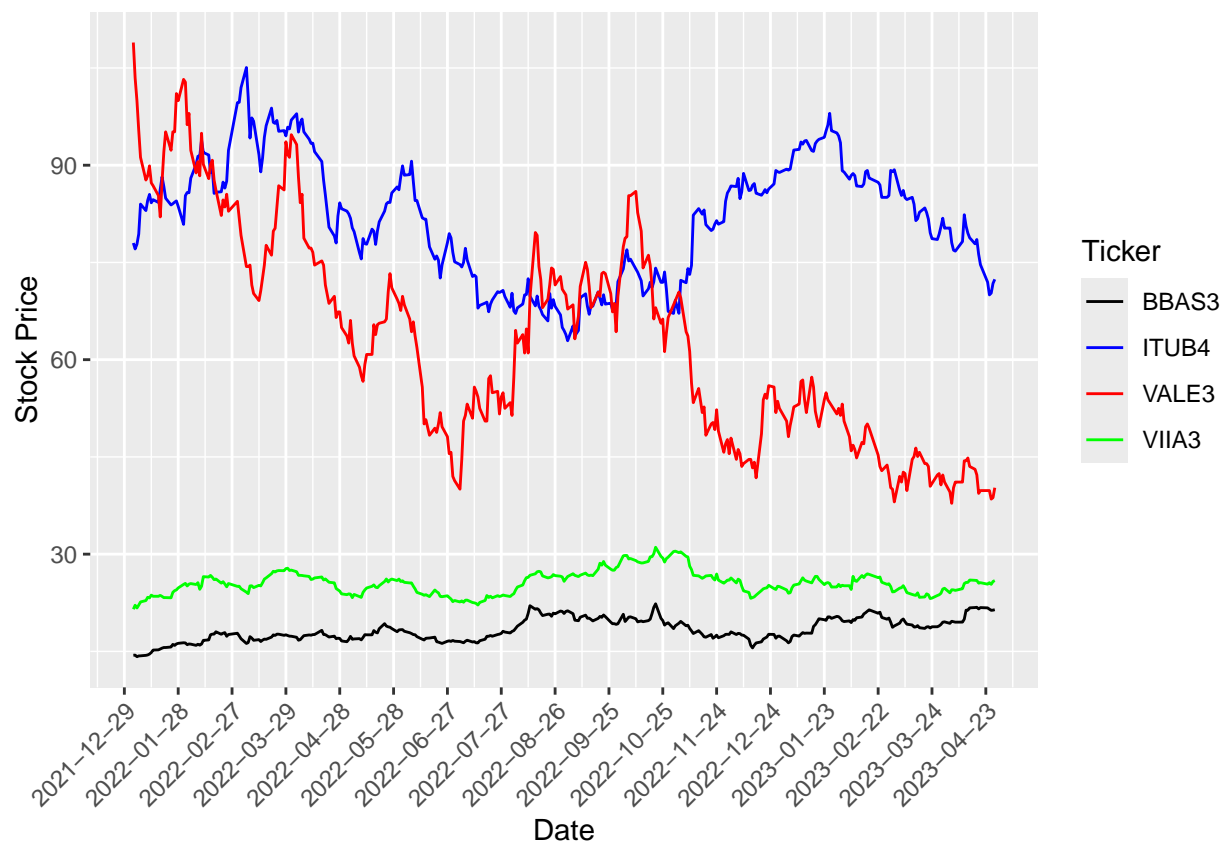
```
VIA3 <- quantmod::getSymbols("BHIA3.SA", src = "yahoo", auto.assign = FALSE,  
                             from = '2022-01-01', to = '2023-04-30',  
                             return.class = 'xts')
```

Foram realizados procedimentos de análise descritiva para, visualizar o comportamento dos dados, o gráfico abaixo demonstra a série real dos preços.

```
##### ANALISES DESCRITIVAS #####
##### SUMMARY DAS BASES#####
#base2=data.frame(LOGBB,LOGIT,LOGVL,LOGVIA)
```

```
preco = BBAS3$BBAS3.SA.Close
DATAS = index(BBAS3)
dados1=ITAU$ITUB4.SA.Close
dados2 =VALE$VALE3.SA.Close
dados3= VIA3$BHIA3.SA.Close
dados_combinados <- data.frame(DATAS,preco,dados1, dados2, dados3)
names(dados_combinados)<-c('DATAS','BBAS3','VIAA3', 'ITUB4', 'VALE3')
```

```
#Reorganizando os dados para o formato longo usando a função pivot_longer do dplyr
dados_combinados_long <- pivot_longer(dados_combinados, -DATAS, names_to = "Ticker", values_to = "preco")
cores <- c("black", "blue", "red", "green")
ggplot(dados_combinados_long, aes(x = DATAS, y = preco, color = Ticker)) +
  geom_line() +
  labs(x = "Date", y = "Stock Price") +
  scale_x_date(date_labels = "%Y-%m-%d", date_breaks = "30 day") +
  scale_color_manual(values = cores) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```
# Plotagem usando ggplot2
# ggplot(dados_combinados_long, aes(x = DATAS, y = preco, color = Ticker)) +
#   geom_line() +
#   labs(x = "Date", y = "Stock Price") +
#   scale_x_date(date_labels = "%Y-%m-%d", date_breaks = "30 day") +
```

```
# theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

Note que no intervalo avaliado as ações do setor bancários obtiveram maior valor em todo o período se comparado com a varejista e a estatal.

Neste trabalho montamos nossa contribuição considerando que os retornos financeiros são independentes, para retirar a dependência haja visto que é uma série temporal nos trabalhamos com os incrementos, ou seja, filtramos o fechamento dos preços diários e tomamos a diferença dos preços do dia posterior e anterior. Em processos similares como movimento browniano temos evidências axiomáticas que este procedimento faz com que garantimos a independência dos dados, para além disso aplicamos o logaritmo nos retornos e partir de agora vamos nos referir a série como log retornos de fechamento de preços. Este procedimento de trabalhar com os incrementos e logaritmar esta descrito nas funções abaixo.

```
#DADOS<-read.csv(file.choose())
#BB=DADOS$bbas3 %>% as.vector()
BB=BBAS3$BBAS3.SA.Close %>% as.vector()
x_div=BB[-1]/BB[-length(BB)] #INCREMENTO
LOGBB=log(x_div)
```

```
IT=ITAU$ITUB4.SA.Close %>% as.vector()
IT_div=IT[-1]/IT[-length(IT)]
LOGIT=log(IT_div)
```

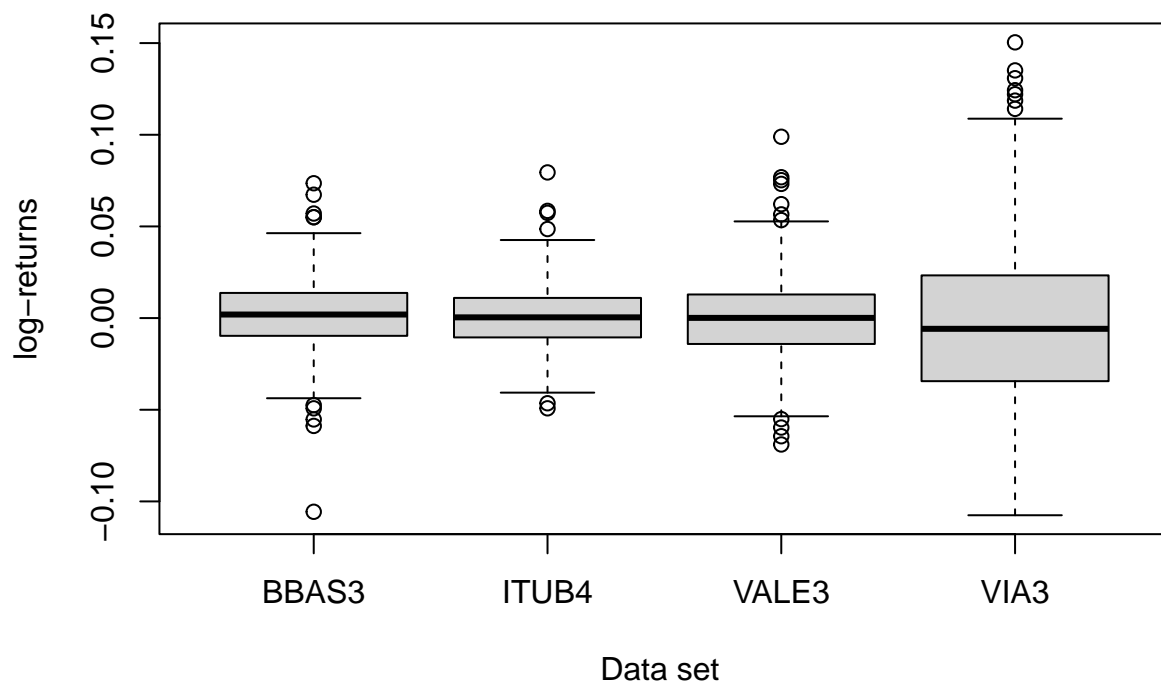
```
VL=VALE$VALE3.SA.Close %>% as.vector()
VL_div=VL[-1]/VL[-length(VL)]
LOGVL=log(VL_div)
```

```
VIA=VIA3$BHIA3.SA.Close %>% as.vector()
VIA_div=VIA[-1]/VIA[-length(VIA)]
LOGVIA=log(VIA_div)
```

Logo após realizamos a análise boxplot para verificar se há outliers ou não nos dados, o código abaixo nos fornece os boxplots.

```
lista_de_dados <- list(LOGBB, LOGIT, LOGVL, LOGVIA)
nomes_dos_datasets <- c("BBAS3", "ITUB4", "VALE3", "VIA3")

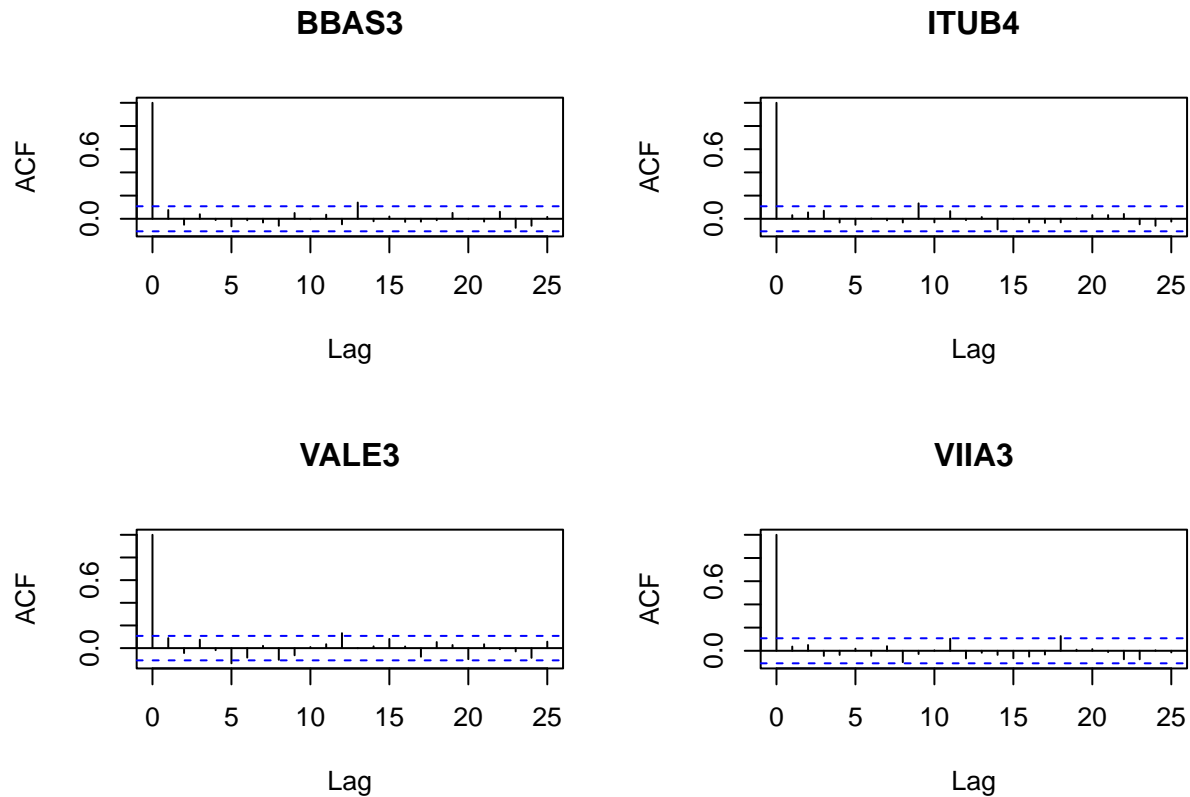
boxplot(lista_de_dados,
        names = nomes_dos_datasets,
        xlab = "Data set ",
        ylab = "log-returns",
        main = " ")
```



Nota-se que o boxplot nos indicou a presença de valores discrepante, porém como trabalharemos com distribuição de calda pesada é bom que tenha valores outliers.

Para verificar se há presença de autocorrelação plotamos um gráfico ACF

```
BBAS3=LOGBB; ITUB4=LOGIT; VALE3=LOGVL ; VIIA3=LOGVIA
par(mfrow=c(2,2))
acf(BBAS3, main="BBAS3")
acf(ITUB4, main="ITUB4")
acf(VALE3, main="VALE3")
acf(VIIA3, main="VIIA3")
```



### GRÁFICO DE AUTOCORRELAÇÃO

Nota-se que não há evidência de autocorrelação nas quatro bases. Logo abaixo montamos a tabela com as principais medidas descritivas.

```
bases=data.frame(LOGBB,LOGIT,LOGVL,LOGVIA)
descritiva= summary(bases) %>% print()
```

```
##      LOGBB      LOGIT      LOGVL
##  Min.   :-0.105659  Min.   :-0.0492130  Min.   :-0.0689439
## 1st Qu.: -0.009684  1st Qu.: -0.0104991  1st Qu.: -0.0140316
## Median :  0.001940  Median :  0.0003995  Median :  0.0001094
## Mean   :  0.001203  Mean   :  0.0005651  Mean   : -0.0002258
## 3rd Qu.:  0.013631  3rd Qu.:  0.0109299  3rd Qu.:  0.0128365
## Max.   :  0.073551  Max.   :  0.0794472  Max.   :  0.0989011
##      LOGVIA
##  Min.   :-0.107542
## 1st Qu.: -0.034388
## Median : -0.005851
## Mean   : -0.003017
## 3rd Qu.:  0.023058
## Max.   :  0.150404
```

Como neste trabalho consideramos que  $X, Y \sim GEV(\mu, \sigma, \gamma)$ , ou seja seguem distribuição GEV, logo apresentaremos como se comporta esta distribuição através dos gráficos da distribuição.

```
mu=0
sigma=0.1
```

```

k=-0.3
x=LOGBB
devd(x, loc = 0, scale = 0.1, shape =-0.3 )

```

```

## [1] 3.690113 3.453259 3.757881 3.690198 3.732294 3.722077 3.770861 3.815244
## [9] 3.856493 3.699959 3.764171 3.763530 3.740496 3.678794 3.728216 3.857058
## [17] 3.558127 3.800926 3.741103 3.732229 3.661533 3.454510 3.747078 3.657748
## [25] 3.531912 3.778524 3.571179 3.780749 3.868618 3.722291 3.842709 3.736519
## [33] 3.678794 3.837381 3.374788 3.768181 3.730558 3.179943 3.815565 3.781588
## [41] 3.340556 3.309869 2.951018 3.754445 3.794494 3.509153 3.398984 3.741918
## [49] 3.438669 3.797593 3.767417 3.655263 3.837754 3.835896 3.618120 3.784405
## [57] 3.672495 3.558233 3.723252 3.607246 3.610462 3.672381 3.563860 3.282992
## [65] 3.730816 3.828075 3.824807 3.742180 3.656145 3.728686 3.688095 3.867679
## [73] 3.109435 3.662808 3.385041 3.755626 3.355315 3.724523 3.658874 3.342085
## [81] 3.590999 3.838446 3.864263 3.292291 3.713105 3.707028 3.531754 3.844453
## [89] 3.855755 3.662785 3.678794 3.867087 3.490243 3.608959 3.868045 3.859350
## [97] 3.402888 3.605912 3.645147 3.539633 3.283614 3.810079 3.699318 3.687715
## [105] 3.496118 3.518032 3.594086 3.566840 3.662799 3.450823 3.348506 3.592530
## [113] 3.813494 3.703859 3.753702 2.989769 3.636580 3.601946 3.609326 3.838544
## [121] 3.716707 3.618316 3.765871 3.595413 3.640460 3.614070 3.601946 3.806667
## [129] 3.770625 3.456374 3.713937 3.540198 3.665039 3.838124 3.802948 3.859921
## [137] 3.681965 3.747631 3.595616 3.810319 3.714592 3.743973 3.828706 3.598853
## [145] 3.579792 3.798510 3.747553 3.830639 3.828874 3.867341 3.818531 3.761841
## [153] 3.853604 3.794719 3.307165 3.728949 3.566616 3.343909 3.424475 3.775131
## [161] 3.691873 3.453567 3.850693 3.619435 3.838707 3.646353 3.477196 3.810714
## [169] 3.727425 3.378194 2.832636 3.628594 3.861410 3.811840 3.340596 3.689615
## [177] 3.550476 3.534631 3.834712 3.809670 3.592521 3.827201 3.518931 2.973705
## [185] 3.516154 3.681647 3.614076 3.727140 3.615648 2.698806 3.814964 3.824791
## [193] 3.702506 3.486612 3.329074 3.747464 3.612303 3.760571 3.785829 3.858392
## [201] 3.844978 3.853969 1.554689 3.444809 3.107551 3.792613 3.738973 2.870745
## [209] 3.850741 3.839433 3.806863 3.118612 3.727672 3.282142 3.403086 3.358518
## [217] 3.853981 3.185069 3.376213 3.638526 3.851011 3.157966 3.790802 3.816086
## [225] 3.343835 3.826431 3.788803 3.688073 3.666082 3.845856 3.345488 3.775099
## [233] 3.653177 3.514242 3.662724 3.136477 2.619292 3.313966 3.862838 3.837223
## [241] 3.779016 3.856947 3.769095 3.766007 3.853713 3.662840 3.152828 3.825307
## [249] 3.697425 2.960829 3.420702 3.794513 3.849106 3.801389 3.557272 3.820350
## [257] 3.803220 3.601506 3.702823 3.666215 3.780051 3.804252 3.847114 3.853272
## [265] 3.587513 3.619371 3.860684 3.673327 3.537757 3.812704 3.647915 3.523750
## [273] 3.416400 3.602643 3.735770 3.499008 3.850371 3.616247 3.834266 3.736557
## [281] 3.849457 3.766496 3.782582 3.772580 3.314712 3.758174 3.043861 3.500697
## [289] 3.767473 3.171887 3.005859 3.782669 3.847004 3.848000 3.774114 3.368027
## [297] 3.416224 3.603753 3.451048 3.690386 3.778378 3.445396 3.611750 3.787391
## [305] 3.704358 3.540301 3.774982 3.684622 3.747547 3.785611 3.827202 3.706055
## [313] 3.527813 3.817866 3.637495 3.640432 3.684410 3.848224 3.687150 3.673593
## [321] 3.820896 3.703374 3.698548 3.528791 3.769458 3.655299 3.564066 3.578799
## [329] 3.691512 3.713275

```

```

#fit_LOGBB
x_points = seq(-2,6, by=0.1)

##### variando locação #####
par(mfrow=c(2,2))

curve(devd(x,0,0.5,-0.3),from=-2, to=6, main=TeX("$\sigma=0.1, \gamma=-0.3$"),

```

```

ylab = "g(x)" ,type="b", pch=15)

lines(x_points, devd(x_points, 1,0.5,-0.3), type="b",
      pch=16,
      lty=2,
      #lwd=2.0,
      col="red")
lines(x_points, devd(x_points, -1,0.5,-0.3), type="b",
      pch=17,
      lty=2,
      #lwd=2.0,
      col="blue")
legend("topright", #inset=.05,
      y=0.5,
      title=TeX("$\mu$"), c("0","1", "-1"),
      lty=c(2), pch=15:17, col=c("black", "red", "blue" ))

#####varianda scala#####

curve(devd(x,0,0.1,-0.3),from=-2, to=6,main=TeX("$\mu=0$, gamma=-0.3$"),
      ylab = "g(x)" ,type="b", pch=15)

lines(x_points, devd(x_points, 0,0.5,-0.3), type="b",
      pch=16,
      lty=2,
      #lwd=2.0,
      col="red")
lines(x_points, devd(x_points, 0,1,-0.3), type="b",
      pch=17,
      lty=2,
      #lwd=2.0,
      col="blue")
legend("topright", #inset=.05,
      y=0.5,
      title=TeX("$\sigma$"), c("0.1","0.5", "1"),
      lty=c(2), pch=15:17, col=c("black", "red", "blue" ))

#####Variando shape#####

curve(devd(x,0,0.5,-1),from=-2, to=6,main=TeX("$\mu=0$, sigma=0.5$"),
      ylab = "g(x)" ,type="b", pch=15)

lines(x_points, devd(x_points, 0,0.5,-0.3), type="b",
      pch=17,
      lty=2,
      #lwd=2.0,
      col="blue")

lines(x_points, devd(x_points, 0,0.5,-.5), type="b",
      pch=16,

```

```

lty=2,
#lwd=2.0,
col="red")

legend("topright", #inset=.05,
      y=0.5,
      title=TeX("$\gamma$"), c("-1", "-0.5", "-0.3"),
      lty=c(2), pch=15:17, col=c("black", "red", "blue" ))

##### VARIANDO SHAPE COM SINAIS TROCADOS #####

curve(devd(x,0,1,-1),from=-2, to=6,main=TeX("$\mu=0$, sigma=1$"),
      ylab = "g(x)",type="b", pch=15)

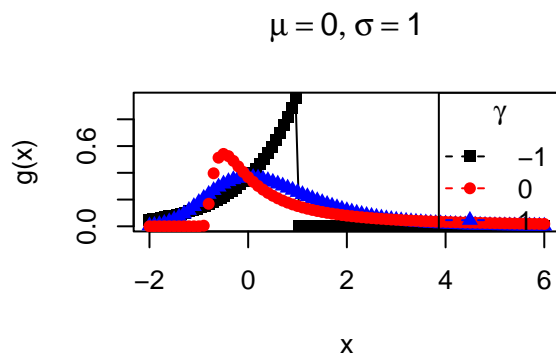
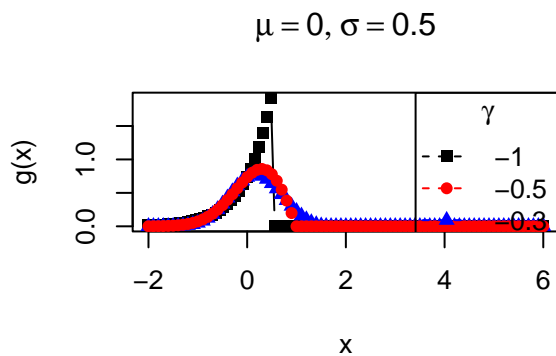
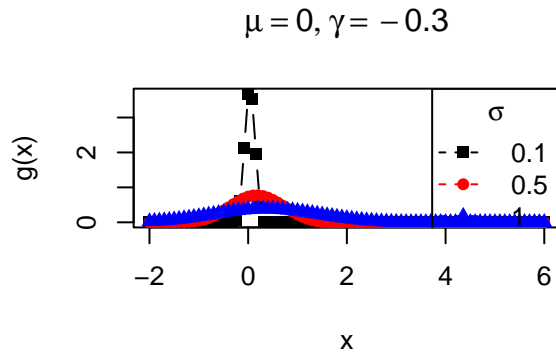
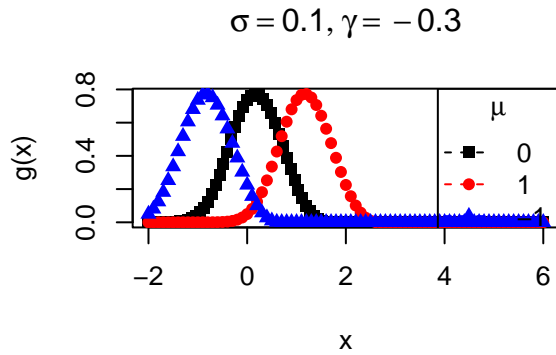
lines(x_points, devd(x_points, 0,1,0), type="b",
      pch=17,
      lty=2,
      #lwd=2.0,
      col="blue")

lines(x_points, devd(x_points, 0,1,1), type="b",
      pch=16,
      lty=2,
      #lwd=2.0,
      col="red")

legend("topright", #inset=.05,
      y=0.5,
      title=TeX("$\gamma$"), c("-1", "0", "1"),
      lty=c(2), pch=15:17, col=c("black", "red", "blue" ))

```





Note o que cada parâmetro faz nesta distribuição, no primeiro gráfico, o parâmetro de escala e o de forma estão fixos variamos o parâmetro de localização perceba que ele transmuta ou para direita ou para esquerda a forma da distribuição, já no segundo gráfico a variamos a escala e os outros ficaram fixos, perceba que esse parâmetro trabalha com a dispersão dos dados, e os dois últimos gráficos colocamos o shape, ou seja, o parâmetro de forma livre e os outros fixos, este parâmetro é muito importante pois ele determina para onde a calda da distribuição está acentuada e a velocidade que ela decai, ele controla se é calda leve ou pesada. Para além disso, a Gev particulariza 3 distribuições, que são elas Fréchet, Weibull e distribuição Gumbel e o parâmetro de forma que indica em qual caso particular os dados se adequam, porém o objetivo deste trabalho não é focar em casos particulares e sim na distribuição que generaliza, ou seja, nosso interesse é na distribuição GEV.

Para isto vamos estimar os parâmetros da distribuição gev. As funções abaixo estimam marginalmente via pacote extRemes, o método de estimação é por -logverossimilhança.

```
fit_LOGBB = fevd(LOGBB,type="GEV")
parBB1=c(fit_LOGBB$results$par%>% as.list())

fit_LOGIT = fevd(LOGIT,type="GEV")
parIT=c(fit_LOGIT$results$par%>% as.list())

(fit_LOGVL = fevd(LOGVL,type="GEV"))

##
## fevd(x = LOGVL, type = "GEV")
##
## [1] "Estimation Method used: MLE"
##
```

```
##
## Negative Log-Likelihood Value: -774.0895
##
##
## Estimated parameters:
## location scale shape
## -0.009497271 0.022220793 -0.163093276
##
## Standard Error Estimates:
## location scale shape
## 0.0013139580 0.0008637121 0.0205215189
##
## Estimated parameter covariance matrix.
## location scale shape
## location 1.726486e-06 1.203197e-07 -7.625157e-06
## scale 1.203197e-07 7.459986e-07 -7.217747e-06
## shape -7.625157e-06 -7.217747e-06 4.211327e-04
##
## AIC = -1542.179
##
## BIC = -1530.782
```

```
parVL=c(fit_LOGVL$results$par%>% as.list())
```

```
(fit_LOGVIA = fevd(LOGVIA,type="GEV"))
```

```
##
## fevd(x = LOGVIA, type = "GEV")
##
## [1] "Estimation Method used: MLE"
##
##
## Negative Log-Likelihood Value: -567.698
##
##
## Estimated parameters:
## location scale shape
## -0.02165616 0.03961411 -0.11696265
##
## Standard Error Estimates:
## location scale shape
## 0.002410209 0.001673711 0.034016012
##
## Estimated parameter covariance matrix.
## location scale shape
## location 5.809106e-06 8.927438e-07 -2.828261e-05
## scale 8.927438e-07 2.801309e-06 -2.159493e-05
## shape -2.828261e-05 -2.159493e-05 1.157089e-03
##
## AIC = -1129.396
##
## BIC = -1117.999
```

```
parVIA=c(fit_LOGVIA$results$par%>% as.list())
parametros = data.frame(
```

```

Bases = c("", "BBAS3", "ITUB4", "VALE3", "VIA3"),
Location = c("location", parBB1$location, parIT$location, parVL$location, parVIA$location),
Scale = c("scale", parBB1$scale, parIT$scale, parVL$scale, parVIA$scale),
Shape = c("shape", parBB1$shape, parIT$shape, parVL$shape, parVIA$shape)
)

# Visualização do data frame
print(parametros)

```

```

##      Bases      Location      Scale      Shape
## 1      location      scale      shape
## 2 BBAS3 -0.00631083023245223 0.0218504153182118 -0.25353884918813
## 3 ITUB4 -0.00644815643019125 0.0165100146256487 -0.154486777789925
## 4 VALE3 -0.00949727092637728 0.0222207927921888 -0.163093275576926
## 5 VIA3  -0.0216561591952117 0.0396141074349253 -0.116962645348691

```

Com os parametros estimados vamos demonstrar algumas funções para continuar a análise, utilizaremos, as funções tais como distribuição acumulada GEV, Densidade GEV, função H e função para calcular o parâmetro de confiabilidade a Fit\_R.

```

H<-function(a1, a2, a3, a4, a5){
  integrand <- function(y){exp(-a1*y-(a2*y^a3+a4)^(a5))}
  integralv<-integrate(integrand, 0, Inf)$value
  return (integralv) #essa seria a integral original calculada numericamente, para comparações
}

# GEV CDF function
gev_cdf <- function(x, mu, sigma, k) {
  if (k == 0) {
    p <- exp(-exp(-(x - mu) / sigma))
  } else {
    p <- exp(-(1 + k * (x - mu) / sigma)^(-1 / k))
  }
  return(p)
}

## Função para Estimar o R
fit_R=function(x_fit, y_fit){
  m1=y_fit[1]
  s1=y_fit[2]
  g1=y_fit[3]

  m2=x_fit[1]
  s2=x_fit[2]
  g2=x_fit[3]

  ## condicao ta atendida?

  if(g1 > 0 && g2 > 0){
    R_est=ifelse(m1-(s1/g1)>=m2-(s2/g2),
      H(1, (g2*s1)/(s2*g1), -g1, 1+(g2/s2)*(m1-m2-(s1/g1)), -1/g2 ),
      1- H(1, (g1*s2)/(s1*g2), -g2, 1+(g1/s1)*(m2-m1-(s2/g2)), -1/g1 ))
  }
}

```

```

}
if(g1 < 0 && g2 < 0){
  R_est=ifelse(m1-(s1/g1)<=m2-(s2/g2),
              H(1, (g2*s1)/(s2*g1), -g1, 1+(g2/s2)*(m1-m2-(s1/g1)), -1/g2 ),
              1- H(1, (g1*s2)/(s1*g2), -g2, 1+(g1/s1)*(m2-m1-(s2/g2)), -1/g1 ))
  #H(1, (g1*s2)/(s1*g2), -g2, 1+(g1/s1)*(m2-m1-(s2/g2)), -1/g1 ))
}
return(R_est)
}

R_hat <- function(x, y) {
  sum(x < y) / length(x)
}

```

Realizamos também um teste de Kolmogorov-smirnov para verificar se os dados se adequam a distribuição GEV. os códigos para o teste ks está definido abaixo.

```

### Colocar os parametros estimados#####
ks.test(LOGBB, gev_cdf,mu=parBB1$location,sigma=parBB1$scale, k=parBB1$shape)

```

```

##
## Asymptotic one-sample Kolmogorov-Smirnov test
##
## data: LOGBB
## D = 0.086294, p-value = 0.01467
## alternative hypothesis: two-sided

```

```

ks.test(LOGIT, gev_cdf,mu=parIT$location,sigma=parIT$scale, k=parIT$shape)

```

```

##
## Asymptotic one-sample Kolmogorov-Smirnov test
##
## data: LOGIT
## D = 0.0481, p-value = 0.4299
## alternative hypothesis: two-sided

```

```

ks.test(LOGVL, gev_cdf,mu=parVL$location ,sigma=parVL$scale, k=parVL$shape)

```

```

##
## Asymptotic one-sample Kolmogorov-Smirnov test
##
## data: LOGVL
## D = 0.057048, p-value = 0.2331
## alternative hypothesis: two-sided

```

```

ks.test(LOGVIA, gev_cdf,mu=parVIA$location,sigma=parVIA$scale , k=parVIA$shape)

```

```

##
## Asymptotic one-sample Kolmogorov-Smirnov test
##
## data: LOGVIA
## D = 0.038916, p-value = 0.6997
## alternative hypothesis: two-sided

```

A 5% de confiança apenas a BBAS3 não foi significativa porém plotamos alguns gráficos de como ficou o ajuste da distribuição aos dados e podemos perceber que o ajustou bem graficamente por este motivo

decidimos manter a BBAS3 no trabalho. abaixo o gráfico dos ajuste.

```
par(mfrow=c(2,2))      # set the plotting area into a 1*2 array

hist(LOGBB, freq = FALSE,
     probability = TRUE,
     main = "",
     breaks = 8,
     #col = "blue", border = "white",
     xlab = "BBAS3", ylab = "Frequency",
     ylim =c(0,25)
)

x_points=seq(min(LOGBB),max(LOGBB),abs(min(LOGBB)-max(LOGBB))/100)
x_gev_points=devd(x_points, loc = parBB1$location, scale =parBB1$scale, shape = parBB1$shape)

lines(x_points, x_gev_points, type="l",
     # pch=15,
     lty=2,
     lwd=2.0,
     col="blue")

legend("topright", #inset=.05,
     y=0.5,
     #title="",
     c("fit"),
     lty=c(2),
     #pch=15,
     col=c("blue"))
#####
hist(LOGIT, freq = FALSE,
     probability = TRUE,
     main = "",
     breaks = 10,
     #col = "blue", border = "white",
     xlab = "ITUB4", ylab = "Frequency",
     ylim =c(0,25)
)

x_points=seq(min(LOGIT),max(LOGIT),abs(min(LOGIT)-max(LOGIT))/100)
x_gev_points=devd(x_points, loc = parIT$location, scale =parIT$scale , shape = parIT$shape)

lines(x_points, x_gev_points, type="l",
     # pch=15,
     lty=2,
     lwd=2.0,
     col="blue")

legend("topright", #inset=.05,
     y=0.5,
     #title="",
     c("fit"),
     lty=c(2),
     #pch=15,
```

```

col=c("blue"))

#####
hist(LOGVL, freq = FALSE,
     probability = TRUE,
     main = "",
     breaks = 8,
     #col = "blue", border = "white",
     xlab = "VALE3", ylab = "Frequency",
     ylim =c(0,25)
)

x_points=seq(min(LOGVL),max(LOGVL),abs(min(LOGVL)-max(LOGVL))/100)
x_gev_points=devd(x_points, loc = parVL$location, scale =parVL$scale, shape = parVL$shape)

lines(x_points, x_gev_points, type="l",
     # pch=15,
     lty=2,
     lwd=2.0,
     col="blue")

legend("topright", #inset=.05,
     y=0.5,
     #title="",
     c("fit"),
     lty=c(2),
     #pch=15,
     col=c("blue"))

#####
hist(LOGVIA, freq = FALSE,
     probability = TRUE,
     main = "",
     breaks = 15,
     #col = "blue", border = "white",
     xlab = "VIIA3", ylab = "Frequency",
     ylim =c(0,10)
)

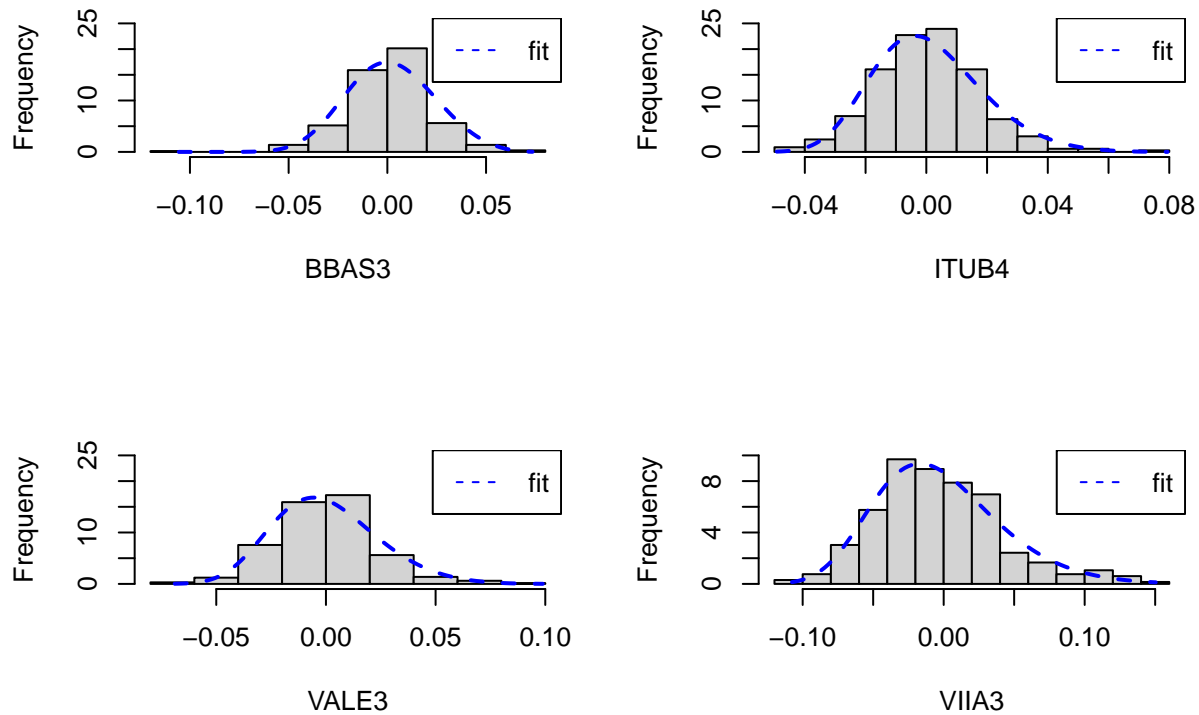
x_points=seq(min(LOGVIA),max(LOGVIA),abs(min(LOGVIA)-max(LOGVIA))/100)
x_gev_points=devd(x_points, loc = parVIA$location, scale =parVIA$scale , shape = parVIA$shape)

lines(x_points, x_gev_points, type="l",
     # pch=15,
     lty=2,
     lwd=2.0,
     col="blue")

legend("topright", #inset=.05,
     y=0.5,
     #title="",
     c("fit"),

```

```
lty=c(2),
#pch=15,
col=c("blue"))
```



Note no grafico acima o modelo GEV ajustou bem aos dados por isso decidimos avançar com as 4 bases.

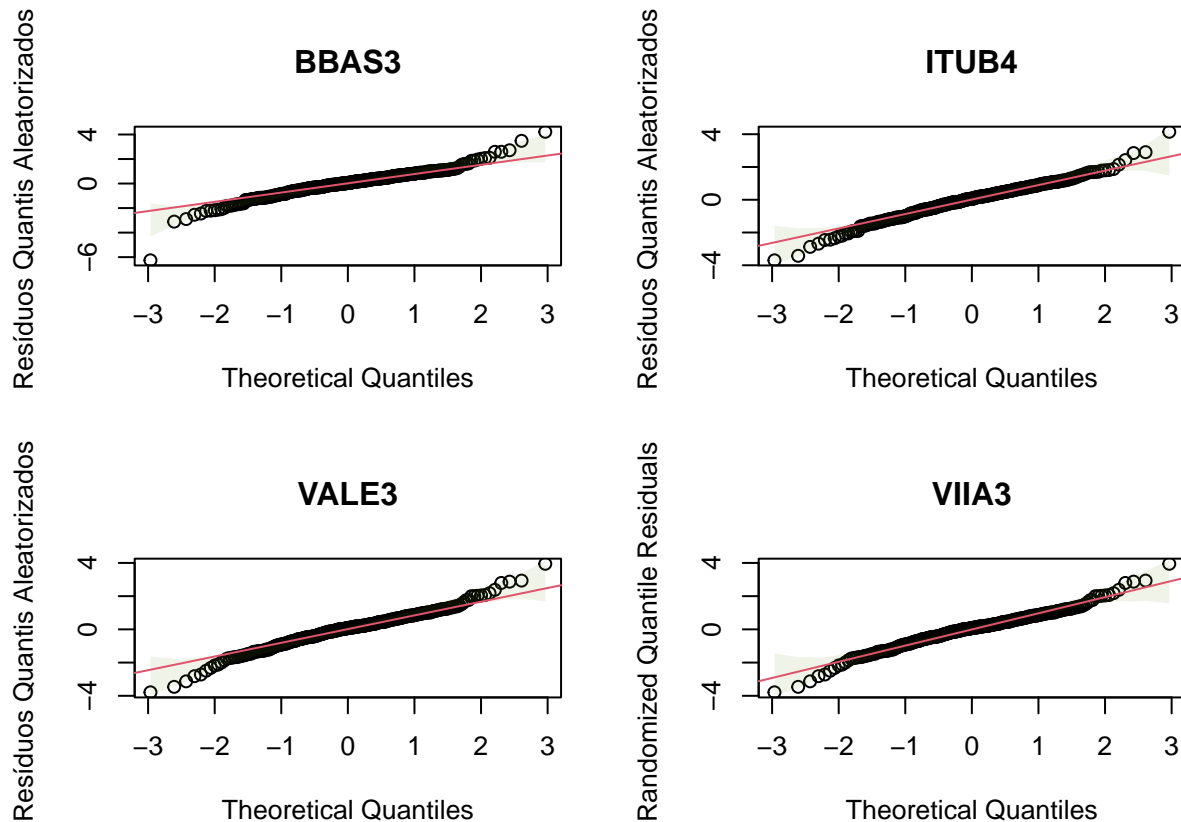
```
##### GRAFICO QQPLOT #####
```

```
par(mfrow = c(2,2))
mu1 = parBB1$location; sigma1=parBB1$scale;k1=parBB1$shape
residuo1=qnorm(gev_cdf(LOGBB, mu = mu1, sigma=sigma1,k=k1))
qqenvelope(residuo1, ylab = "Resíduos Quantis Aleatorizados ", main = "BBAS3"); qqline(residuo1, col = 1)
#qqenvelope(residuo1);qqline(residuo1)

mu2 = parIT$location; sigma2=parIT$scale;k2=parIT$shape
residuo2=qnorm(gev_cdf(LOGIT, mu = mu2, sigma=sigma2,k=k2))
qqenvelope(residuo2, ylab = "Resíduos Quantis Aleatorizados ", main = "ITUB4"); qqline(residuo2, col = 2)
#qqenvelope(residuo2);qqline(residuo2)

mu3 = parVL$location; sigma3=parVL$scale;k3=parVL$shape
residuo3=qnorm(gev_cdf(LOGVL, mu = mu3, sigma=sigma3,k=k3))
qqenvelope(residuo3, ylab = "Resíduos Quantis Aleatorizados ", main = "VALE3"); qqline(residuo3, col = 3)
#qqenvelope(residuo3);qqline(residuo3)

mu4 = parVIA$location; sigma4=parVIA$scale;k4=parVIA$shape
residuo4=qnorm(gev_cdf(LOGVIA, m = mu4, sigma=sigma4,k=k4))
qqenvelope(residuo4, ylab = "Randomized Quantile Residuals ", main = "VIIA3"); qqline(residuo4, col = 4)
#qqenvelope(residuo4);qqline(residuo4)
```



O gráfico acima é o gráfico de Resíduos Quantis Aleatorizados, basicamente contrastamos o nosso resíduos para verificar se ha normalidade, este modelo não possui este pressuposto, porém é interessante olhar os comportamento dos resíduos do modelo para ver se não houve anomalias muito grande no modelo.

Para a estimativa da confiabilidade que é o foco central do nosso trabalho utilizamos as expressões analíticas para obter a estimativa de stress-strength descrita na estimação abaixo, também utilizamos um abordagem não paramétrica que consiste em avaliar a quantidade de vezes em que uma ação foi menor que a outra e dividir pelo tamanho do intervalo assim obtemos uma estimativa de stress-strength via contagem e comprimimos para o intervalo 0 a 1 para comparar com a probabilidade via estimativa. Para fins didáticos consideraremos  $\$X1 = \$ BBAS3$ ,  $\$X2 = \$ ITUB4$ ,  $\$X3 = \$ VALE3$  e  $\$X4 = \$ VIIA3$

```
##### R não parametrico #####
R_hat <- function(x, y) {
  sum(x < y) / length(x)
}
x1=LOGBB; x2=LOGIT; x3=LOGVL; x4=LOGVIA
par_1=R_hat(x=x3,y=x1)
par_2=R_hat(x=x3,y=x2)
par_3=R_hat(x=x3,y=x4)

##### R parametrico #####
fit_R=function(x_fit, y_fit){
  m1=y_fit$location
  s1=y_fit$scale
  g1=y_fit$shape
  m2=x_fit$location
```



```

s2=x_fit$scale
g2=x_fit$shape

## condicao ta atendida?

if(g1 > 0 && g2 > 0){
  R_est=ifelse(m1-(s1/g1)>=m2-(s2/g2),
    H(1, (g2*s1)/(s2*g1), -g1, 1+(g2/s2)*(m1-m2-(s1/g1)), -1/g2 ),
    1- H(1, (g1*s2)/(s1*g2), -g2, 1+(g1/s1)*(m2-m1-(s2/g2)), -1/g1 ))
}
if(g1 < 0 && g2 < 0){
  R_est=ifelse(m1-(s1/g1)<=m2-(s2/g2),
    H(1, (g2*s1)/(s2*g1), -g1, 1+(g2/s2)*(m1-m2-(s1/g1)), -1/g2 ),
    1- H(1, (g1*s2)/(s1*g2), -g2, 1+(g1/s1)*(m2-m1-(s2/g2)), -1/g1 ))
  #H(1, (g1*s2)/(s1*g2), -g2, 1+(g1/s1)*(m2-m1-(s2/g2)), -1/g1 ))
}
return(R_est)
}

fitR1<-R_hat(LOGVL, LOGBB) %>% round(digits = 4)
fitR1P<-fit_R(parVL,parBB1) %>% round(digits = 4)

fitR2<-R_hat(LOGVL, LOGIT) %>% round(digits = 4)
fitR2P<-fit_R(parVL,parIT) %>% round(digits = 4)

fitR3<-R_hat(LOGVL, LOGVIA) %>% round(digits = 4)
fitR3P<-fit_R(parVL,parVIA) %>% round(digits = 4)

# Criar o data frame de confiabilidade
confiabilidade = data.frame(
  Bases = c("", "X3<X1", "X3<X2", "X3<X4"),
  RNP = c("RNP", fitR1, fitR2, fitR3),
  RP = c("RP", fitR1P, fitR2P, fitR3P)
)

confiabilidade

##   Bases    RNP    RP
## 1          RNP    RP
## 2 X3<X1 0.5242 0.5283
## 3 X3<X2 0.5242 0.5174
## 4 X3<X4 0.4364 0.4506

```

Em resumo, quando  $X$  e  $Y$  representam Vas de retorno e  $R < 1/2$ , é aconselhável que o investidor escolha a variável  $X$ . Se  $R > 1/2$ , ocorre o oposto. O caso  $R = 1/2$  é inconclusivo. neste caso fixamos o  $X3$  então se  $R < 1/2$  a melhor opção é escolher  $X3$ , se maior escolha o par contrário, olhando para o contexto de estimativas pontuais faz total sentido os preços de  $X1$  e  $X2$  (Ações bancárias) são maiores que  $X3$  e  $X4$ , logo comparar o modelo está apontando corretamente, por segurança fizemos uma estimativa intervalar via bootstrap omiti o código porque ainda estou trabalhando em uma solução mais rápida e otimizada assim que estiver atualizada eu disponibilizo a estimativa intervalar porém os resultados obtidos tanto para pontual quanto para intervalar está definida na tabela abaixo.