

Boxplot para os dados de Vazão

Versão 2021_04_19:

- alteração de 12 para 15pt na fonte dos gráficos.
- um condicional de precaução para o arquivo de saída "infos_.txt". O condicional (if) confere se o arquivo existe. Caso o arquivo exista, ele é removido antes que seja acrescentada as informações. Então, caso haja a execução do script mais de uma vez, não há duplicação de informações no arquivo.

Versão 2021_04_30:

- Ajustes nas saídas dos gráficos e tabelas (ajustes estéticos, como tamanho da imagem, dimensão, e umas correções gráficas)
- Foi acrescentado na saída do stats os valores de média e dispersão.

Versão 2021_05_03

- Inserida uma saída com os valores dos quantis 0.15, 0.35, 0.65 e 0.85 para cada mês;
- Gráfico da média com as linhas dos quantis;
- Saída tipo excel [colunas: ANO, MÊS, VALOR, CLASSIFICAÇÃO];
 - Um saída para cada uma das classificações;
 - Uma saída do total da estação;
- Saída com a quantidade de cada uma das classificações por mês e total de cada uma estação;
- Melhoramento da estética dos gráficos: mudanças em cores e fonte;
- Foi retidada a exportação dos valores presentes no bp\$conf;
- Foi retirada algumas redundâncias;

Versão 2021_05_08

- Correção de erro no cálculo dos quantis;
- Melhoramento dos Gráficos;
- Retira do gráfico com apenas a média;
- Novos Gráficos: Quantidade de eventos por ano (um gráfico para cada classe)

Versão 2021_05_26

- Gráfico da classificação por mês e ano (bitmap/raster) -> tipo mapa de calor

Versão 2021_06_11

- Troca do arquivo de entrada: Agora é usado os dados da ONS disponibilizado pelo relatório de 2019 e os dados da ANA apenas para o ano de 2020.

Versão 2021_06_23

- Planilha com os valores das médias;
- mapa de classificação:
 - Troca das cores;
 - Negrito nos eixo;
 - Escrita do eixo x inclinada;
 - Escrita do eixo x com intervalo de 2 anos;
 - Melhorias na formatação.

```
In [1]: #install.packages("readxl")
#install.packages("writexl")
#install.packages("gplots")
#install.packages("lattice")
#install.packages("stringr")
#install.packages("ggplot2")
#install.packages("openxlsx")
#install.packages("extrafont")
```

```
In [1]: #install.packages("extrafont")
library("readxl") # ler xls
library("writexl") # escrever
library("gplots") # gráficos
library("lattice") # visualização de dados
library("stringr") # trabalhar com string
```

```
library("ggplot2")
library("openxlsx") #para escrever em xlsx linha por linha
library("extrafont") # pacote com fontes para utilizar nos gráficos
##### ATENÇÃO AQUI #####
#extrafont::font_import() # pedirá confirmação para baixar [pode demorar]
# é necessário importar apenas uma vez
#####
extrafont::loadfonts()
# OBS: para importar os pacotes, pode-se ou não usar as aspas
```

Warning message:

"package 'readxl' was built under R version 3.6.3"Warning message:

"package 'writexl' was built under R version 3.6.3"Warning message:

"package 'gplots' was built under R version 3.6.3"

Attaching package: 'gplots'

The following object is masked from 'package:stats':

lowess

Warning message:

"package 'lattice' was built under R version 3.6.3"Warning message:

"package 'stringr' was built under R version 3.6.3"Warning message:

"package 'openxlsx' was built under R version 3.6.3"Warning message:

"package 'extrafont' was built under R version 3.6.2"Registering fonts with R

Agency FB already registered with pdfFonts().

Algerian already registered with pdfFonts().

Arial Black already registered with pdfFonts().

Arial already registered with pdfFonts().

Arial Narrow already registered with pdfFonts().

Arial Rounded MT Bold already registered with pdfFonts().

Bahnschrift already registered with pdfFonts().

Baskerville Old Face already registered with pdfFonts().

Bauhaus 93 already registered with pdfFonts().

Bell MT already registered with pdfFonts().

Berlin Sans FB already registered with pdfFonts().

No regular (non-bold, non-italic) version of Berlin Sans FB Demi. Skipping setup for this font.

Bernard MT Condensed already registered with pdfFonts().

Blackadder ITC already registered with pdfFonts().

Bodoni MT already registered with pdfFonts().

Bodoni MT Black already registered with pdfFonts().

Bodoni MT Condensed already registered with pdfFonts().

Bodoni MT Poster Compressed already registered with pdfFonts().

Book Antiqua already registered with pdfFonts().

Bookman Old Style already registered with pdfFonts().

Bookshelf Symbol 7 already registered with pdfFonts().

Bradley Hand ITC already registered with pdfFonts().

Britannic Bold already registered with pdfFonts().

Broadway already registered with pdfFonts().

No regular (non-bold, non-italic) version of Brush Script MT. Skipping setup for this font.

Calibri already registered with pdfFonts().

Calibri Light already registered with pdfFonts().

Californian FB already registered with pdfFonts().

Calisto MT already registered with pdfFonts().

No regular (non-bold, non-italic) version of Cambria. Skipping setup for this font.

Candara already registered with pdfFonts().

Candara Light already registered with pdfFonts().

Castellar already registered with pdfFonts().

Centaur already registered with pdfFonts().

Century already registered with pdfFonts().

Century Gothic already registered with pdfFonts().

Century Schoolbook already registered with pdfFonts().

Chiller already registered with pdfFonts().

Colonna MT already registered with pdfFonts().

Comic Sans MS already registered with pdfFonts().

Consolas already registered with pdfFonts().

Constantia already registered with pdfFonts().

Cooper Black already registered with pdfFonts().

Copperplate Gothic Bold already registered with pdfFonts().

Copperplate Gothic Light already registered with pdfFonts().

Corbel already registered with pdfFonts().

Corbel Light already registered with pdfFonts().

Courier New already registered with pdfFonts().

Curlz MT already registered with pdfFonts().

Dubai already registered with pdfFonts().

Dubai Light already registered with pdfFonts().

Dubai Medium already registered with pdfFonts().

Ebrima already registered with pdfFonts().

Edwardian Script ITC already registered with pdfFonts().

Elephant already registered with pdfFonts().

Engravers MT already registered with pdfFonts().

Eras Bold ITC already registered with pdfFonts().

Eras Demi ITC already registered with pdfFonts().

Eras Light ITC already registered with pdfFonts().

Eras Medium ITC already registered with pdfFonts().

Felix Titling already registered with pdfFonts().

Footlight MT Light already registered with pdfFonts().
 Forte already registered with pdfFonts().
 Franklin Gothic Book already registered with pdfFonts().
 Franklin Gothic Demi already registered with pdfFonts().
 Franklin Gothic Demi Cond already registered with pdfFonts().
 Franklin Gothic Heavy already registered with pdfFonts().
 Franklin Gothic Medium already registered with pdfFonts().
 Franklin Gothic Medium Cond already registered with pdfFonts().
 Freestyle Script already registered with pdfFonts().
 French Script MT already registered with pdfFonts().
 Gabriola already registered with pdfFonts().
 Gadugi already registered with pdfFonts().
 Garamond already registered with pdfFonts().
 Georgia already registered with pdfFonts().
 Gigi already registered with pdfFonts().
 Gill Sans Ultra Bold already registered with pdfFonts().
 Gill Sans Ultra Bold Condensed already registered with pdfFonts().
 Gill Sans MT already registered with pdfFonts().
 Gill Sans MT Condensed already registered with pdfFonts().
 Gill Sans MT Ext Condensed Bold already registered with pdfFonts().
 Gloucester MT Extra Condensed already registered with pdfFonts().
 Goudy Old Style already registered with pdfFonts().
 Goudy Stout already registered with pdfFonts().
 Haettenschweiler already registered with pdfFonts().
 No regular (non-bold, non-italic) version of Harlow Solid Italic. Skipping setup for this font.
 Harrington already registered with pdfFonts().
 High Tower Text already registered with pdfFonts().
 HoLoLens MDL2 Assets already registered with pdfFonts().
 Impact already registered with pdfFonts().
 Imprint MT Shadow already registered with pdfFonts().
 Informal Roman already registered with pdfFonts().
 Ink Free already registered with pdfFonts().
 Javanese Text already registered with pdfFonts().
 Jokerman already registered with pdfFonts().
 Juice ITC already registered with pdfFonts().
 Kievit Offc Pro already registered with pdfFonts().
 Kristen ITC already registered with pdfFonts().
 Kunstler Script already registered with pdfFonts().
 Wide Latin already registered with pdfFonts().
 Leelawadee UI already registered with pdfFonts().
 Leelawadee UI Semilight already registered with pdfFonts().
 More than one version of regular/bold/italic found for Lucida Bright. Skipping setup for this font.
 No regular (non-bold, non-italic) version of Lucida Calligraphy. Skipping setup for this font.
 Lucida Console already registered with pdfFonts().
 More than one version of regular/bold/italic found for Lucida Fax. Skipping setup for this font.
 No regular (non-bold, non-italic) version of Lucida Handwriting. Skipping setup for this font.
 More than one version of regular/bold/italic found for Lucida Sans. Skipping setup for this font.
 Lucida Sans Typewriter already registered with pdfFonts().
 Lucida Sans Unicode already registered with pdfFonts().
 No regular (non-bold, non-italic) version of Magneto. Skipping setup for this font.
 Maiandra GD already registered with pdfFonts().
 Malgun Gothic already registered with pdfFonts().
 Malgun Gothic Semilight already registered with pdfFonts().
 Marlett already registered with pdfFonts().
 Matura MT Script Capitals already registered with pdfFonts().
 Microsoft Himalaya already registered with pdfFonts().
 Microsoft Yi Baiti already registered with pdfFonts().
 Microsoft New Tai Lue already registered with pdfFonts().
 Microsoft PhagsPa already registered with pdfFonts().
 Microsoft Sans Serif already registered with pdfFonts().
 Microsoft Tai Le already registered with pdfFonts().
 Mistral already registered with pdfFonts().
 Modern No. 20 already registered with pdfFonts().
 Mongolian Baiti already registered with pdfFonts().
 Monotype Corsiva already registered with pdfFonts().
 MS Outlook already registered with pdfFonts().
 MS Reference Sans Serif already registered with pdfFonts().
 MS Reference Specialty already registered with pdfFonts().
 MT Extra already registered with pdfFonts().
 MV Boli already registered with pdfFonts().
 Myanmar Text already registered with pdfFonts().
 Niagara Engraved already registered with pdfFonts().
 Niagara Solid already registered with pdfFonts().
 Nirmala UI already registered with pdfFonts().
 Nirmala UI Semilight already registered with pdfFonts().
 OCR A Extended already registered with pdfFonts().
 Old English Text MT already registered with pdfFonts().
 Onyx already registered with pdfFonts().
 Palace Script MT already registered with pdfFonts().
 Palatino Linotype already registered with pdfFonts().
 Papyrus already registered with pdfFonts().
 Parchment already registered with pdfFonts().
 Perpetua already registered with pdfFonts().
 Perpetua Titling MT already registered with pdfFonts().
 Playbill already registered with pdfFonts().
 Poor Richard already registered with pdfFonts().
 Pristina already registered with pdfFonts().

No regular (non-bold, non-italic) version of Rage Italic. Skipping setup for this font.
 Ravie already registered with pdfFonts().
 Rockwell already registered with pdfFonts().
 Rockwell Condensed already registered with pdfFonts().
 Rockwell Extra Bold already registered with pdfFonts().
 Script MT Bold already registered with pdfFonts().
 Segoe MDL2 Assets already registered with pdfFonts().
 Segoe Print already registered with pdfFonts().
 Segoe Script already registered with pdfFonts().
 Segoe UI already registered with pdfFonts().
 Segoe UI Light already registered with pdfFonts().
 Segoe UI Semibold already registered with pdfFonts().
 Segoe UI Semilight already registered with pdfFonts().
 Segoe UI Black already registered with pdfFonts().
 Segoe UI Emoji already registered with pdfFonts().
 Segoe UI Historic already registered with pdfFonts().
 Segoe UI Symbol already registered with pdfFonts().
 Showcard Gothic already registered with pdfFonts().
 SimSun-ExtB already registered with pdfFonts().
 Snap ITC already registered with pdfFonts().
 Stencil already registered with pdfFonts().
 Sylfaen already registered with pdfFonts().
 Symbol already registered with pdfFonts().
 Tahoma already registered with pdfFonts().
 Tempus Sans ITC already registered with pdfFonts().
 Times New Roman already registered with pdfFonts().
 Trebuchet MS already registered with pdfFonts().
 Tw Cen MT already registered with pdfFonts().
 Tw Cen MT Condensed already registered with pdfFonts().
 Tw Cen MT Condensed Extra Bold already registered with pdfFonts().
 Verdana already registered with pdfFonts().
 Viner Hand ITC already registered with pdfFonts().
 No regular (non-bold, non-italic) version of Vivaldi. Skipping setup for this font.
 Vladimir Script already registered with pdfFonts().
 More than one version of regular/bold/italic found for dbldwrs. Skipping setup for this font.
 More than one version of regular/bold/italic found for Warsaw. Skipping setup for this font.
 Webdings already registered with pdfFonts().
 Wingdings already registered with pdfFonts().
 Wingdings 2 already registered with pdfFonts().
 Wingdings 3 already registered with pdfFonts().

ÁRVORE:

Boxplot_Vazao:

```
|_ _ Dados (dentro desta pasta tem que estar o .xlsx com os dados a serem analisados)

|_ _ Scripts (pasta com o script)

|_ _ Resultados_BoxPlot_SeriesVazao (esta pasta será criada com a execussão deste _script_)
```

```
In [12]: ##### M U D A R   A Q U I #####

pastaScripts = 'C:/Users/amand/OneDrive/Área de Trabalho/Boxplot_Vazao/Scripts'
setwd(pastaScripts)
#
#arquivoEntrada = "../Dados/Series_Mensais_ONS_SAR_1931_2020.xlsx"

arquivoEntrada = "../Dados/SeriesMensais_ONS1970_2019_ANA_2020.xlsx"
#
nameDirBP = "../Resultados_BoxPlot_SeriesVazao_ONS2019_ANA2020"
#caminho relativo para a criação da pasta

anoDeInicio = 1980
anoDeFim = 2020

windowsFonts(A = windowsFont("Times New Roman")) # Fonte dos gráficos
# para escolher outra fonte, use o comando font() e veja a Lista, e troque no windowsFont()

cinza = rgb(red = 0.45,blue = 0.45,green = 0.45) # "setei" um tom de cinza diferente para preencher os gráfico
# para mudar a cor das barras e linhas do gráfico, altere em 'col'

# Cores: Segue a mesma ordem do vetor 'classificacao'
# MB - B - N - A - MA
#coresClasses = c("darkblue","cornflowerblue","azure","tan1","orangered1")
#coresClasses = c("wheat","burlywood","peru","saddlebrown","black") #tons de marrom
#coresClasses = c("DodgerBlue","LightSkyBlue","White","Gold","DarkOrange")
#coresClasses = c("DodgerBlue","LightSkyBlue","PaleGreen1","Gold","DarkOrange")
coresClasses = c("DarkOrange","Gold","PaleGreen1","LightSkyBlue","DodgerBlue")

##
# tabela de cores: https://expanssiva.com.br/pg/tabela-de-cores-html-hexadecimal-e-rgb
```

```
#####

seriesVazao = read_excel(arquivoEntrada)
seriesVazao = subset(seriesVazao, seriesVazao[1]>=anoDeInicio)
seriesVazao = subset(seriesVazao, seriesVazao[1]<=anoDeFim)

# criando o diretório para os resultados
if(!dir.exists(nameDirBP)) # se a pasta não existe
  dir.create(nameDirBP)

for(ptObs in 3:length(seriesVazao)) # rodando as colunas do seriesVazao, da 3a até a última
  # obs: a primeira e a segunda são ano e mês
{
  estacao = NULL # zerando a variável
  estacao = na.omit(data.frame(seriesVazao[1:2],seriesVazao[ptObs]))
  # excluindo as linhas com NA

  nomePt = colnames(seriesVazao[ptObs])
  # PRECAUÇÃO: as vezes os nomes das colunas vem com ponto no lugar do espaço.
  # A função str_replace_all vai substituir [.] por [ ]

  nameDir = str_replace_all(colnames(seriesVazao[ptObs]),"[:blank:],[.]", "")
  # tirando pontos e espaços para usar como nome de arquivo
  path = paste(nameDirBP,"/",nameDir,sep='')
  # a função paste vai agrupar os argumentos, utilizando o separador indicado
  # o default de separador é um espaço simples

  #criando uma pasta para cada estação
  if(!dir.exists(path)) # se a pasta não existe
    dir.create(path) # a pasta é criada

  nameFileBP = paste(path, "/boxplot_",nameDir,".jpg",sep='')
  #criando o nome do gráfico boxplot

  nameFileBar = paste(path, "/barra_media_",nameDir,".jpg",sep='')
  #criando o nome do gráfico barra média

  nameFileQuantil = paste(path, "/quantil_media_",nameDir,".jpg",sep='')
  #criando o nome do gráfico barra média com quantil

  nameFileQtdEventoMB = paste(path, "/barra_quantidadeEvento_MB_",nameDir,".jpg",sep='')
  nameFileQtdEventoB = paste(path, "/barra_quantidadeEvento_B_",nameDir,".jpg",sep='')
  nameFileQtdEventoN = paste(path, "/barra_quantidadeEvento_N_",nameDir,".jpg",sep='')
  nameFileQtdEventoA = paste(path, "/barra_quantidadeEvento_A_",nameDir,".jpg",sep='')
  nameFileQtdEventoMA = paste(path, "/barra_quantidadeEvento_MA_",nameDir,".jpg",sep='')

  bp = NULL # zerando a variável

  #####CONSTRUINDO O GRÁFICO BOXPLOT #####
  graphics.off()
  #family = fonte das letras
  jpeg(filename = nameFileBP, width = 960, height = 660,family = "A",
        units = "px", pointsize = 20, quality = 100, bg = "white", res = NA)

  bp <- boxplot(estacao[3][estacao$"MÊS" ==1,],
                estacao[3][estacao$"MÊS" ==2,],
                estacao[3][estacao$"MÊS" ==3,],
                estacao[3][estacao$"MÊS" ==4,],
                estacao[3][estacao$"MÊS" ==5,],
                estacao[3][estacao$"MÊS" ==6,],
                estacao[3][estacao$"MÊS" ==7,],
                estacao[3][estacao$"MÊS" ==8,],
                estacao[3][estacao$"MÊS" ==9,],
                estacao[3][estacao$"MÊS" ==10,],
                estacao[3][estacao$"MÊS" ==11,],
                estacao[3][estacao$"MÊS" ==12,],
                main= nomePt,
                col = cinza,
                pch = 16, #plotting character, para outras opções digite '?pch'
                names= c(1:12),
                xlab = "Mês", ylab = "Vazão (m³/s)")
  dev.off() #terminou o gráfico Boxplot

  #####CONSTRUINDO O GRÁFICO DE BARRA - média #####
  media = NULL
  media = c(mean(estacao[3][estacao$"MÊS" == 1,]),mean(estacao[3][estacao$"MÊS" == 2,]),
```

```

mean(estacao[3][estacao$"MÊS" == 3,]),mean(estacao[3][estacao$"MÊS" == 4,]),
mean(estacao[3][estacao$"MÊS" == 5,]),mean(estacao[3][estacao$"MÊS" == 6,]),
mean(estacao[3][estacao$"MÊS" == 7,]),mean(estacao[3][estacao$"MÊS" == 8,]),
mean(estacao[3][estacao$"MÊS" == 9,]),mean(estacao[3][estacao$"MÊS" ==10,]),
mean(estacao[3][estacao$"MÊS" ==11,]),mean(estacao[3][estacao$"MÊS" ==12,]))

wb <- createWorkbook()
addWorksheet(wb, "Sheet1")
#primeira linha: um título
writeData(wb, "Sheet1", "Mês", startCol = 1, startRow = 1)
writeData(wb, "Sheet1", "Vazão Média (m³/s)", startCol = 1, startRow = 2)

for(i in 1:12) #inserindo os valores
{
  writeData(wb, "Sheet1", i, startCol = i+1, startRow = 1, rowNames = TRUE)
  writeData(wb, "Sheet1", round(media[i]), startCol = i+1, startRow = 2, rowNames = TRUE)
}
#salvando no arquivo
saveWorkbook(wb, paste(path,"/media_",nameDir,".xlsx",sep=''), overwrite = TRUE)

#graphics.off()
#jpeg(filename = nameFileBar, width = 960, height = 480,
#      units = "px", pointsize = 20, family = "A",
#      quality = 100, bg = "white", res = NA)
#barplot(media~c(1:12), main = nomePt,
#      ylab = expression(paste("Vazão (", m^3/s,")", sep=" ")), xlab = "Mês",
#      xlim=range(0:14), ylim=range(0:max(ceiling(1.2*media))))
#grid(nx = NA, ny = NULL, lty = "solid") #caso não queira o grid, comente essa linha.
#dev.off() #terminou o gráfico barplot

##### CONSTRUINDO O SAÍDA DOS QUANTIS #####
q15= NULL
q35= NULL
q65= NULL
q85= NULL

wb <- createWorkbook()
addWorksheet(wb, "Sheet1")
mesesDoAno = c("Janeiro", "Fevereiro", "Março", "Abril", "Maio", "Junho",
               "Julho", "Agosto", "Setembro", "Outubro", "Novembro", "Dezembro")
writeData(wb, "Sheet1", "15%", startCol = 1, startRow = 2, rowNames = TRUE)
writeData(wb, "Sheet1", "35%", startCol = 1, startRow = 3, rowNames = TRUE)
writeData(wb, "Sheet1", "65%", startCol = 1, startRow = 4, rowNames = TRUE)
writeData(wb, "Sheet1", "85%", startCol = 1, startRow = 5, rowNames = TRUE)

for(i in 1:12) # meses do ano
{
  writeData(wb, "Sheet1", mesesDoAno[i], startCol = 1+i, startRow = 1, rowNames = TRUE)
  # calculando os quantis
  q15[i] = quantile(estacao[3][estacao$"MÊS" == i,], probs = 0.15)
  q35[i] = quantile(estacao[3][estacao$"MÊS" == i,], probs = 0.35)
  q65[i] = quantile(estacao[3][estacao$"MÊS" == i,], probs = 0.65)
  q85[i] = quantile(estacao[3][estacao$"MÊS" == i,], probs = 0.85)
  #quantil = quantile(estacao[3][estacao$"MÊS" == i,], probs = c(0.15, 0.35,0.65,0.85))

  writeData(wb, "Sheet1", q15[i], startCol = 1+i, startRow = 2, rowNames = TRUE)
  writeData(wb, "Sheet1", q35[i], startCol = 1+i, startRow = 3, rowNames = TRUE)
  writeData(wb, "Sheet1", q65[i], startCol = 1+i, startRow = 4, rowNames = TRUE)
  writeData(wb, "Sheet1", q85[i], startCol = 1+i, startRow = 5, rowNames = TRUE)
}
#salvando no arquivo
saveWorkbook(wb, paste(path,"/ValoresQuantil_",nameDir,".xlsx",sep=''), overwrite = TRUE)

#####CONSTRUINDO O GRÁFICO MÉDIA COM QUANTIS #####

charMeses = c("JAN", "FEV", "MAR", "ABR", "MAI", "JUN", "JUL", "AGO", "SET", "OUT", "NOV", "DEZ")

graphics.off()
jpeg(filename = nameFileQuantil, width = 1200, height = 680,
      units = "px", pointsize = 20,
      quality = 100, bg = "white", res = NA)

plot(media~c(1:12),
      main = nomePt, type = "h",lwd = 25, col = cinza ,pch = 1,omi=c(5,60,4,2),
      xaxt = "n", family = "A", cex.axis = 1.25, cex.lab = 1.5,cex.main = 2,
      ylab = expression(paste("Vazão (", m^3/s,")", sep=" ")), xlab = "Mês",par(mar=c(5,4,4,2)+0.6),
      ylim=range(min(q15):(max(q85)*1.05)))

#O proporcional no limite máximo é para a legenda não sobrepor o gráficos
axis(side=1, at=seq(1,12,1), labels=charMeses, cex.axis = 1.25, family="A")
lines(q15, col = "blue",lwd = 3) #lwd = espessura da linha

```

```

lines(q35, col = "green", lwd = 3)
lines(q65, col = "orange", lwd = 3)
lines(q85, col = "red", lwd = 3)
legend("top", c("Média", "Quantil de 15%", "Quantil de 35%", "Quantil de 65%", "Quantil de 85%"),
      fill=c(cinza, "blue", "green", "orange", "red"), text.font = 6, cex = 1.15,
      xpd=NA, horiz=TRUE, bty='n')
### text.font= 6 -> times new roman

dev.off() #terminou o gráfico de barra com quantis em linhas

##### CLASSIFICAÇÃO - QUANTIS #####
classificacao = c("Muito Baixa", "Baixa", "Normal", "Alta", "Muito Alta")

MuitoBaixa = NULL
Baixa = NULL
Normal = NULL
Alta = NULL
MuitoAlta = NULL

#estacao[LINHA,COLUNA]
#COLUNA 1 = ANO
#COLUNA 2 = MÊS
#COLUNA 3 = VAZAO
#COLUNA 4 = CLASSIFICAÇÃO
# ou seja:
# estacao[i,3] -> vazão na linha i
# estacao[i,2] -> o mês da linha i

for(i in 1:nrow(estacao))
{ # Loop que percorre as linhas do dataframe estação
  if(estacao[i,3] <= q15[estacao[i,2]]) # se o valor da linha i é menor que o q15 do mês da linha i
  { # A classificação da vazão vai ser "Muito Baixa"
    estacao$"Classificação"[i] = classificacao[1]
  } else # caso não for, vai entrar no primeiro else
  {
    # se o valor é maior que o q15 do mês E menor que o q35 do mês
    if(estacao[i,3] > q15[estacao[i,2]] && estacao[i,3] <= q35[estacao[i,2]])
    { # Classificação = Baixa
      estacao$"Classificação"[i] = classificacao[2]
    } else # caso não for, também vai entrar no segundo else
    {
      # se o valor for maior que o q35 E menor que o q65 do mês
      if(estacao[i,3] > q35[estacao[i,2]] && estacao[i,3] <= q65[estacao[i,2]])
      { # classificacao=Normal
        estacao$"Classificação"[i] = classificacao[3]
      } else # caso não for, também vai entrar no terceiro else
      {
        # se o valor for maior que q65 e menor que o q85
        if(estacao[i,3] > q65[estacao[i,2]] && estacao[i,3] <= q85[estacao[i,2]])
        { # classificação = Alta
          estacao$"Classificação"[i] = classificacao[4]
        } else # caso não for, também vai entrar no quarto else
        { # se o valor for maior que o q85 do mês
          if(estacao[i,3] > q85[estacao[i,2]])
          { # classificação = Muito Alta
            estacao$"Classificação"[i] = classificacao[5]
          } else # testando se está funcionando
          { # se estiver tudo certo, não aparecerá nenhum print
            print(i)
          } # fim do else de teste
        } # fim do quarto else
      } # fim do terceiro else
    } # fim do segundo else
  } # fim do primeiro else
} # fim do for que percorre as linhas da estacao
# agora o dataframe estação tem uma coluna a mais, chamada "Classificação"

# filtrando os dados pela classificação
# foi criado um dataframe para cada uma das classificações
# o dataframe tem as mesmas colunas que o df da estacao
MuitoBaixa = subset(estacao, estacao[4]==classificacao[1])
Baixa = subset(estacao, estacao[4]==classificacao[2])
Normal = subset(estacao, estacao[4]==classificacao[3])
Alta = subset(estacao, estacao[4]==classificacao[4])
MuitoAlta = subset(estacao, estacao[4]==classificacao[5])

# salvando os dataframes em planilhas
write_xlsx(MuitoBaixa, paste(path, "/MuitoBaixa_", nameDir, ".xlsx", sep=''), col_names=TRUE)
write_xlsx(Baixa, paste(path, "/Baixa_", nameDir, ".xlsx", sep=''), col_names=TRUE)
write_xlsx(Normal, paste(path, "/Normal_", nameDir, ".xlsx", sep=''), col_names=TRUE)
write_xlsx(Alta, paste(path, "/Alta_", nameDir, ".xlsx", sep=''), col_names=TRUE)

```



```

write_xlsx(MuitoAlta, paste(path, "/MuitoAlta_", nameDir, ".xlsx", sep=""), col_names=TRUE)
write_xlsx(estacao, paste(path, "/Total_", nameDir, ".xlsx", sep=""), col_names=TRUE)

# EXPORTANDO OS DADOS
# planilha com as quantidades de cada uma das classificações por mês e o total
wb <- createWorkbook()
addWorksheet(wb, "Sheet1")
for(i in 1:length(classificacao))
  writeData(wb, "Sheet1", classificacao[i], startCol = 1+i, startRow = 1, rowNames = TRUE)
for(i in 1:12) # valores por mês
{
  writeData(wb, "Sheet1", i, startCol = 1, startRow = 1+i, rowNames = TRUE)
  writeData(wb, "Sheet1", nrow(subset(MuitoBaixa, MuitoBaixa[2]==i)),
    startCol = 2, startRow = 1+i, rowNames = TRUE)
  writeData(wb, "Sheet1", nrow(subset(Baixa, Baixa[2]==i)),
    startCol = 3, startRow = 1+i, rowNames = TRUE)
  writeData(wb, "Sheet1", nrow(subset(Normal, Normal[2]==i)),
    startCol = 4, startRow = 1+i, rowNames = TRUE)
  writeData(wb, "Sheet1", nrow(subset(Alta, Alta[2]==i)),
    startCol = 5, startRow = 1+i, rowNames = TRUE)
  writeData(wb, "Sheet1", nrow(subset(MuitoAlta, MuitoAlta[2]==i)),
    startCol = 6, startRow = 1+i, rowNames = TRUE)
}
# total
writeData(wb, "Sheet1", "Total", startCol = 1, startRow = 14, rowNames = TRUE)
writeData(wb, "Sheet1", nrow(MuitoBaixa), startCol = 2, startRow = 14, rowNames = TRUE)
writeData(wb, "Sheet1", nrow(Baixa), startCol = 3, startRow = 14, rowNames = TRUE)
writeData(wb, "Sheet1", nrow(Normal), startCol = 4, startRow = 14, rowNames = TRUE)
writeData(wb, "Sheet1", nrow(Alta), startCol = 5, startRow = 14, rowNames = TRUE)
writeData(wb, "Sheet1", nrow(MuitoAlta), startCol = 6, startRow = 14, rowNames = TRUE)

# salvando no arquivo
saveWorkbook(wb, paste(path, "/QuantidadeQuantil_", nameDir, ".xlsx", sep=""), overwrite = TRUE)

##### GRÁFICOS DA QUANTIDADE DE EVENTO POR ANO #####
contEventosMB = NULL
contEventosB = NULL
contEventosN = NULL
contEventosA = NULL
contEventosMA = NULL

periodo=c(anoDeInicio:anoDeFim)
for(i in c(1:length(periodo)))
{
  contEventosMB[i] = nrow(subset(MuitoBaixa, MuitoBaixa[1]==periodo[i]))
  contEventosB[i] = nrow(subset(Baixa, Baixa[1]==periodo[i]))
  contEventosN[i] = nrow(subset(Normal, Normal[1]==periodo[i]))
  contEventosA[i] = nrow(subset(Alta, Alta[1]==periodo[i]))
  contEventosMA[i] = nrow(subset(MuitoAlta, MuitoAlta[1]==periodo[i]))
}
quantEventos= data.frame(MB = contEventosMB, B = contEventosB, N = contEventosN, A = contEventosA, MA = co

graphics.off() #EVENTO DE VAZÃO MUITO BAIXA
jpeg(filename = nameFileQtdEventoMB, width = 1200, height = 680,
  units = "px", pointsize = 20,
  quality = 100, bg = "white", res = NA)
barplot(quantEventos[,1]~periodo, main=paste("Eventos de Vazão Muito Baixa:", nomePt),
  ylim=range(0:12), omi=c(5,60,4,2), family = "A", las=2, #xaxt = "n",
  cex.axis = 1.25, cex.lab = 1.5, cex.main = 2, par(mar=c(5,4,4,2)+0.6),
  ylab="Quantidade de Eventos", xlab="Ano")
grid(nx=NA, ny=NULL)
dev.off() #terminou o gráfico barplot

graphics.off() #EVENTOS DE VAZÃO BAIXA
jpeg(filename = nameFileQtdEventoB, width = 1200, height = 680,
  units = "px", pointsize = 20,
  quality = 100, bg = "white", res = NA)
barplot(quantEventos[,2]~periodo, main=paste("Eventos de Vazão Baixa:", nomePt),
  ylim=range(0:12), omi=c(5,60,4,2), family = "A", las=2, #xaxt = "n",
  cex.axis = 1.25, cex.lab = 1.5, cex.main = 2, par(mar=c(5,4,4,2)+0.6),
  ylab="Quantidade de Eventos", xlab="Ano")
grid(nx=NA, ny=NULL)
dev.off() #terminou o gráfico barplot

graphics.off() #EVENTOS DE VAZÃO NORMAL
jpeg(filename = nameFileQtdEventoN, width = 1200, height = 680,
  units = "px", pointsize = 20,
  quality = 100, bg = "white", res = NA)
barplot(quantEventos[,3]~periodo, main=paste("Eventos de Vazão Normal:", nomePt),
  ylim=range(0:12), omi=c(5,60,4,2), family = "A", las=2, #xaxt = "n",
  cex.axis = 1.25, cex.lab = 1.5, cex.main = 2, par(mar=c(5,4,4,2)+0.6),

```



```

      ylab="Quantidade de Eventos",xlab="Ano")
    grid(nx=NA, ny=NULL)
    dev.off() #terminou o gráfico barplot

graphics.off() #EVENTOS DE VAZÃO ALTA
  jpeg(filename = nameFileQtdEventoA, width = 1200, height = 680,
        units = "px", pointsize = 20,
        quality = 100, bg = "white", res = NA)
  barplot(quantEventos[,4]~periodo, main=paste("Eventos de Vazão Alta:",nomePt),
        ylim=range(0:12),omi=c(5,60,4,2),family = "A",las=2,#xaxt = "n",
        cex.axis = 1.25, cex.lab = 1.5,cex.main = 2, par(mar=c(5,4,4,2)+0.6),
        ylab="Quantidade de Eventos",xlab="Ano")
  grid(nx=NA, ny=NULL)
  dev.off() #terminou o gráfico barplot

graphics.off() #EVENTOS DE VAZÃO MUITO ALTA
  jpeg(filename = nameFileQtdEventoMA, width = 1200, height = 680,
        units = "px", pointsize = 20,
        quality = 100, bg = "white", res = NA)
  barplot(quantEventos[,5]~periodo, main=paste("Eventos de Vazão Muito Alta:",nomePt),
        ylim=range(0:12),omi=c(5,60,4,2),family = "A",las=2,#xaxt = "n",
        cex.axis = 1.25, cex.lab = 1.5,cex.main = 2, par(mar=c(5,4,4,2)+0.6),
        ylab="Quantidade de Eventos",xlab="Ano")
  grid(nx=NA, ny=NULL)
  dev.off() #terminou o gráfico barplot

##### MAPA DE CLASSES #####

nomeTitulo = paste("Classificação da Vazão:", nomePt)
nameFileBitMap = paste(path, "/bitmap_",nameDir,".jpg",sep='')
graphics.off()

myplot1 <- ggplot(estacao,aes(x=ANO, y=MÊS,fill = Classificação)) + #iniciando o plot

coord_fixed(ratio=1.5)+ # proporção do "pixels"
# maior que 1 => retângulo com altura maiores que a largura
# igual a 1 => quadrado
# menor que 1 => retângulo com largura maior que a altura
theme(# formatação
      axis.text.x = element_text( angle = 320, face = "bold"), #escrita do eixo x na vertical
      axis.text.y = element_text( face = "bold"),
      plot.title = element_text(hjust = 0.5), # centralizar título
      text = element_text(size=20),# tamanho da letra e negrito

      legend.key = element_rect(fill = "white", colour = cinza), #caixa de legenda
      legend.position="bottom", # posição da caixa de legenda
      panel.background = element_blank()+ # tirando o fundo do gráfico

      geom_raster()+
      geom_tile(color="grey")+ #linha que separa dos "pixels"
      labs(x = "Ano", y = "Mês",fill = "",title =nomeTitulo) +

      scale_fill_manual(values=coresClasses,breaks=classificacao) +

      scale_x_continuous(breaks = seq(anoDeInicio,anoDeFim,2)) +

      scale_y_continuous(breaks = c(1:12))

  jpeg(filename = nameFileBitMap, width = 630, height = 400,family = "A",
        units = "px", pointsize = 12, quality = 100, bg = "white", res = NA)
  print(myplot1)
  dev.off() #terminou o gráfico

##### FIM DO MAPA DE CLASSES #####

##### Arquivo Infos_ #####
nomeFileInfo = paste(path,"/infos_",nameDir,".txt",sep='')
#criando nome do arquivo que salvará algumas das informações geradas pelo boxplot

#esse if é uma precaução para caso rodem o script mais de uma vez.
# se o arquivo existe, ele vai ser removido antes que seja acrescentada as infos
if(file.exists(nomeFileInfo))
  file.remove(nomeFileInfo)

write(paste("Estação: ",nomePt) , file = nomeFileInfo, append = TRUE)
#escrevendo no arquivo o nome da estação
# append = TRUE significa que será acrescentada ao arquivo se existente
write(paste("Total de Outliers: ",length(bp$out)) , file = nomeFileInfo, append = TRUE)
#escrevendo no arquivo o total de outliers da estação

```

```

write("", file = nomeFileInfo, append = TRUE)# pulando linha

write("Quantidade de Outliers por Mês: ", file = nomeFileInfo, append = TRUE)

write(paste("Janeiro: ",length(bp$group[bp$group==1])) , file = nomeFileInfo, append = TRUE)
write(paste("Fevereiro: ",length(bp$group[bp$group==2])) , file = nomeFileInfo, append = TRUE)
write(paste("Março: ",length(bp$group[bp$group==3])) , file = nomeFileInfo, append = TRUE)
write(paste("Abril: ",length(bp$group[bp$group==4])) , file = nomeFileInfo, append = TRUE)
write(paste("Maio: ",length(bp$group[bp$group==5])) , file = nomeFileInfo, append = TRUE)
write(paste("Junho: ",length(bp$group[bp$group==6])) , file = nomeFileInfo, append = TRUE)
write(paste("Julho: ",length(bp$group[bp$group==7])) , file = nomeFileInfo, append = TRUE)
write(paste("Agosto: ",length(bp$group[bp$group==8])) , file = nomeFileInfo, append = TRUE)
write(paste("Setembro: ",length(bp$group[bp$group==9])) , file = nomeFileInfo, append = TRUE)
write(paste("Outubro: ",length(bp$group[bp$group==10])) , file = nomeFileInfo, append = TRUE)
write(paste("Novembro: ",length(bp$group[bp$group==11])) , file = nomeFileInfo, append = TRUE)
write(paste("Dezembro: ",length(bp$group[bp$group==12])) , file = nomeFileInfo, append = TRUE)

write("", file = nomeFileInfo, append = TRUE)# pulando linha
write("Total de Dados por Mês: ", file = nomeFileInfo, append = TRUE)
write(paste(c(1:12), bp$n) , file = nomeFileInfo, append = TRUE, sep='\t')

#####

# setando em variáveis alguns dos resultados gerados pela função boxplot
limInferior = bp$stats[1,]
quartil1 = bp$stats[2,]
mediana = bp$stats[3,]
quartil3 = bp$stats[4,]
limSuperior = bp$stats[5,]

bpStats = data.frame(mes=c(1:12),limInferior,quartil1,mediana, quartil3,limSuperior, media)
#criando um dataframe os as variáveis setadas
write_xlsx(bpStats, paste(path,"/stats_",nameDir,".xlsx",sep=''), col_names=TRUE)
#escrevendo em um arquivo .xlsx

# setando em variáveis alguns dos resultados gerados pela função boxplot

# ExtremoSup = bp$conf[1,]
# ExtremoInf = bp$conf[2,]
# bpConf = data.frame(mes=c(1:12),ExtremoSup,ExtremoInf)
# write_xlsx(bpConf, paste(path,"/conf_",nameDir,".xlsx",sep=''), col_names=TRUE)
#escrevendo em um arquivo .xlsx

ano = NULL
for(i in 1:length(bp$out))
  ano[i] = estacao[1][estacao[2]==bp$group[i] & estacao[3] == bp$out[i]]
dadosOut = data.frame(ano, Mes= bp$group, outliers=bp$out)
write_xlsx(dadosOut, paste(path,"/outliers_",nameDir,".xlsx",sep=''), col_names=TRUE)

# escrevendo numa planilha os valores dos outliers separados por mês
# separando os dados por mês
outPorMes = list(valor1 = dadosOut[3][dadosOut[2]== 1], ano1 = dadosOut[1][dadosOut[2]== 1],
  valor2 = dadosOut[3][dadosOut[2]== 2], ano2 = dadosOut[1][dadosOut[2]== 2],
  valor3 = dadosOut[3][dadosOut[2]== 3], ano3 = dadosOut[1][dadosOut[2]== 3],
  valor4 = dadosOut[3][dadosOut[2]== 4], ano4 = dadosOut[1][dadosOut[2]== 4],
  valor5 = dadosOut[3][dadosOut[2]== 5], ano5 = dadosOut[1][dadosOut[2]== 5],
  valor6 = dadosOut[3][dadosOut[2]== 6], ano6 = dadosOut[1][dadosOut[2]== 6],
  valor7 = dadosOut[3][dadosOut[2]== 7], ano7 = dadosOut[1][dadosOut[2]== 7],
  valor8 = dadosOut[3][dadosOut[2]== 8], ano8 = dadosOut[1][dadosOut[2]== 8],
  valor9 = dadosOut[3][dadosOut[2]== 9], ano9 = dadosOut[1][dadosOut[2]== 9],
  valor10 = dadosOut[3][dadosOut[2]==10], ano10 = dadosOut[1][dadosOut[2]==10],
  valor11 = dadosOut[3][dadosOut[2]==11], ano11 = dadosOut[1][dadosOut[2]==11],
  valor12 = dadosOut[3][dadosOut[2]==12], ano12 = dadosOut[1][dadosOut[2]==12])

wb <- createWorkbook()
addWorksheet(wb, "Sheet1")
# vetor com os nomes dos meses
mesesDoAno = c("Janeiro", "Fevereiro", "Março", "Abril", "Maio", "Junho",
  "Julho", "Agosto", "Setembro", "Outubro", "Novembro", "Dezembro")
#primeira linha: um título
writeData(wb, "Sheet1", paste("Valores e Anos dos 'outliers' da estação",nomePt), startCol = 1, startRow

for(i in seq(1,length(outPorMes),2)) #nome nas células
{
  writeData(wb, "Sheet1", mesesDoAno[(i/%2)+1], startCol = i, startRow = 2, rowNames = TRUE)
  writeData(wb, "Sheet1", "Valor", startCol = i, startRow = 3, rowNames = TRUE)
  writeData(wb, "Sheet1", "Ano", startCol = i+1, startRow = 3, rowNames = TRUE)
}

```

```
for(i in 1:length(outPorMes)) #inserindo os valores
{
  writeData(wb, "Sheet1", outPorMes[i], startCol = i, startRow = 4, rowNames = TRUE)
}
#salvando no arquivo
saveWorkbook(wb, paste(path,"/outliersPorMes_",nameDir,".xlsx",sep=''), overwrite = TRUE)

wb <- createWorkbook()
addWorksheet(wb, "Sheet1")
#primeira linha: um título
writeData(wb, "Sheet1", paste("Valores e Anos dos 'outliers' da estação",nomePt), startCol = 1, startRow

for(i in seq(1,length(outPorMes),2)) #nome nas células
{
  writeData(wb, "Sheet1", mesesDoAno[(i%/2)+1], startCol = i, startRow = 2, rowNames = TRUE)
}

for(i in 1:length(outPorMes)) #inserindo os valores
{
  writeData(wb, "Sheet1", round(outPorMes[[i]]), startCol = i, startRow = 3, rowNames = TRUE)
}
#salvando no arquivo
saveWorkbook(wb, paste(path,"/outliersPorMesV2_",nameDir,".xlsx",sep=''), overwrite = TRUE)

} # fim do loop que roda todas as estações
```