Trabalho final: Milena de Godoy-Veiga

Proposta entregue: Proposta A, com justificativas e alterações destacadas

tree.gRowth (`data`, `year`, `hm`, `histogram`, `retornar` = TRUE, `file` = TRUE)

Descrição:

Uma função para análise exploratória de medidas de anéis de crescimento. A partir de data frames com as amostras distribuídas por colunas, a função entrega análises básicas da estrutura etária da população (histograma de idades e boxplot), taxa de crescimento (gráfico de crescimento acumulado) e padrão de crescimento (scatterplot de crescimento por idade).

Entrada:

Justificativa: foi sugerido adicionar um argumento para a análise de todas as planilhas dentro de um diretório. Tal sugestão não foi cumprida, não houve tempo hábil para aprender essa parte. Preferi entregar a função com todas as outras partes funcionando corretamente. Para facilitar a análise de vários arquivos, foi adicionado o argumento `file`, para o usuário poder controlar o nome dos arquivos gerados de acordo com suas preferências. O argumento lyr foi removido pois julguei que não havia necessidade, visto que os dados não viriam nesse outro formato. Os outros arugumentos foram mantidos, apenas `return` passou para `retornar`, visando evitar confusão com outros com outros comandos já existentes no sistema do R.

`data` = nome do documento .txt ou .csv em que os dados de crescimento estão. Aceita data frames que contenham uma amostra por coluna. Primeira linha contendo o nome das amostras. A segunda linha deve conter a medida do último anel formado (lyr = TRUE). Se os valores estiverem invertidos, lyr = FALSE.
`year` = um vetor com dois valores, de mês e ano em que a coleta foi feita (ex. c(6, 2018))
`hm` = em qual hemisfério a coleta foi feita (“south” ou “north”)
`histogram` = um vetor com a idade inicial, final e o intervalo desejado no histograma (ex. c(0, 300, 25)
`retornar` = TRUE retorna os data.frames gerados
`file` = o radical para nome dos arquivos retornados

Testando entradas:

`data` **Existe?**~~é um arquivo .txt ou .csv igual ao especificado (ver se primeira linha tem nomes e segunda números)~~? Se não, retorna *“Check your data **name**~~. Should be something like this: escrever um exemplo de data frame~~”*
`year` é um vetor com duas posições, e onde a primeira é um número entre 1 e 12? Se não, retorna *“Check your year. Should be something like this: my.year ← c (4,2018)”*
`hm` é igual a “sul” ou “norte”? Se não, retorna *“Check parameter hm. Should be south or north“*
`histogram` é um vetor com **mais de**três valores numéricos ~~e a divisão do segundo pelo terceiro é um número inteiro~~? Se não, retorna *“Check parameter histogram. Should be something like this: my.histogram ← c(0, 300, 25)”*

Pseudo-código:

#Arrumando as planilhas

1. A partir de `data`, criar um data.frame *"growth"*

~~-If `lty` = FALSE inverter os valores das colunas para que a segunda linha contenha os valores dos anéis formados no último ano.~~
~~-Onde não existem valores (final da coluna, já que as árvores mais velhas terão mais valores do que as mais novas), existem NA ou 0? Preencher tudo com NA\\~~

1. if (`hm` == "sul" e `year` [1] > 7) #se estivermos no hemisfério sul e a coleta for realizada depois do mês de julho, temos que subtrair um ano já que a estação de crescimento nesse hemisfério começa no meio do ano.
2. **Corrindo a planilha com while**
    1. Adicionar uma coluna *"Years"* com valores decrescentes, começando com `year` [2] – 1, **e depois arrumar os nomes dos indivíduos**
3. else
4. Adicionar uma coluna *"Years"* com valores decrescentes, começando com `year` [2]

#Análises

1. ~~for.~~**while** Para cada coluna de *"growth"* será calculado:
    1. Comprimento (length). Guardar esse valor em um vetor numérico *"ages"* #Esse comprimento é igual a idade
    2. Crescimento acumulado, que será o valor da última posição de cada coluna (ano 1 da árvore) somado a seguinte (ano 2), até o primeiro valor. Guardar estes valores em um data frame *"accumulated.growth"*, onde a primeira coluna será igual a primeira coluna de *"growth"*. **Ao inves de fazer pelo for, a planilha foi invertida, os NA foram removidos e foi feito o cálculo com a função cumsum. Após os calculos a planilha foi restaurada ao formato normal **
2. Criar um segundo data frame *"lifetime.growth.pattern"* ← *"growth"*
    1. A primeira coluna será *"cambial age"*, começando do 0 e irá até a idade da árvore mais velha.
    2. **com while**, inverter todos os valores, retira os Nas # todos os anos 0 das árvores deverão estar na segunda linha agora.
    3. Preencher com NA o final das colunas

#Resultados

1. Construir um histograma da frequência das idades, calculado a partir de *"ages"*, pelo intervalo indicado em `histogram`
2. Construir um boxplot com *"ages"*
3. Criar um gráfico de crescimento acumulado de *"accumulated growth"* mostrando todas as amostras como linhas. Adicionar uma linha de crescimento médio e o valor da inclinação.
4. Construir um scatterplot com os valores de *"lifetime.growth.pattern"* (crescimento por *"cambial age"*) Adicionar uma linha de média suavizada (LOESS).

Saída:

Salvar em pdf, ~~mostrar~~ em um painel (2,2)

1. Histograma com distribuição de idades
2. Boxplot das idades
3. Gráfico de crescimento acumulado de todas as amostras

4. Scatterplot do padrão de crescimento de todas as amostras
5. Se return = TRUE, retorna os data.frames ***"growth"***, *"accumulated growth"* e *"lifetime.growth.pattern"* **salvos em .txt no diretório do usuário**

Código da função

```
tReegrowth = function (data,  year, hm, histogram, retornar = TRUE, file) #
Cria a funcao "tReegrowth" com os argumentos "dim", "data",  "year", "hm",
"histogram", "lyr", "retornar = TRUE", "file".
{
  install.packages ("reshape2")
  library ("reshape2") #chamando o pacote necessario para melt
  ############################VERIFICANDO PARAMETROS################
  if (file.exists(data) == FALSE) # Verifica se nome entrado existe na
pasta.
    {
    stop ("Check your data name.") # Se nao, para e exibe mensagem.
    }
  if (class(year) != "numeric") #verifica se year eh um numero
    {stop ("Check your year, must be numeric") # Se nao, para e exibe
mensagem.
    }
  if (length(year) != 2) #verifica se year tem duas posicoes
    {stop ("Check your year, must have two numbers") # Se nao, para e exibe
mensagem.
    }
  if (year[1]>12 | year[1] < 1 ) #verifica mes year
    {stop ("Check month inserted in year") # Se nao, para e exibe mensagem.
    }
  if (hm != "south" && hm != "north") #verifica se hm eh south ou north
    {stop ("check your hm, must be north or south") # Se nao, para e exibe
mensagem.
    }
  if (class(histogram) != "numeric") #verifica se histogram eh um numero
    {stop ("Check your histogram, must be numeric") # Se nao, para e exibe
mensagem.
    }
  if (length(histogram) < 3) #verifica se histogram tem 3 posicoes
    {stop ("Check histogram, must have more than 3 numbers") # Se nao, para
e exibe mensagem.
    }

  if (retornar == TRUE) # Verifica se o usuario quer os arquivos e esta
analisando apenas um data frame.
  {
    if (file == FALSE) # Se sim, verifica se foi entrado valor para file.
```

```
    {
      stop ("If retornar = TRUE you need this name for the dataframes") # Se
nao, para e exibe mensagem.
    }
  }
  ##############1. Arrumando as planilhas############################
  growth <- read.table (data, header = TRUE, sep = "\t", dec = ".") #lendo a
planilha data com cabecalho, separada por tabulacao e ponto como decimal
  # Adicionando uma coluna years na primeira posicao com valores
decrescentes, comecando com year [2] menos 1
  if (hm == "south" && year [1] > 7) # verifica se a coleta foi feita no
hemisferio sul e a coleta for realizada depois do mes 7.
  {
    growth$last <- NaN #criando uma coluna com Nan
    Z <- ncol (growth) #criando objeto z com valor do numero de colunas de
growth
    while (Z>1) { #enquanto z que e igual ao numero de colunas for maior 1,
repetir
      growth [ ,Z] <- growth [ ,(Z-1)] #atribuindo o valor da penultima
coluna para a ultima coluna
      Z <- Z-1 #no proximo ciclo sera subtraido 1 de z
      }
    growth [,1] <- seq (from = year [2] - 1, by = -1, length.out = nrow
(growth)) #adicionando na primeira coluna de gorwth essa sequencia, continua
a sequencia ate o numero de linhas de growth
    x <- names (growth) #criando objeto x com os nomes da primeira colunas
de growth
    N <- length(x) #criando objeto n com o valor do comprimento de x
    while (N>1){  #enquanto N que igual ao comprimento de x for maior 1,
repetir
      x [N] <- x[N-1] #atribuindo o nome da penultima coluna para a ultima
coluna
      N = N-1 #no proximo ciclo sera subtraido 1 de
      }
  }
  ###########se não precisar subtrair 1 no ano
  else {
    growth$last <- NaN #criando uma coluna com Nan
    Z <- ncol (growth) #criando objeto z com valor do numero de colunas de
growth
    while (Z>1) { #enquanto z que e igual ao numero de colunas for maior 1,
repetir
      growth [ ,Z] <- growth [ ,(Z-1)] #atribuindo o valor da penultima
coluna para a ultima coluna
      Z <- Z-1 #no proximo ciclo sera subtraido 1 de z
      }
    growth [,1] <- seq (from = year [2], by = -1, length.out = nrow
(growth)) #adicionando na primeira coluna de gorwth essa sequencia, continua
a sequencia ate o numero de linhas de growth
    x <- names (growth) #criando objeto x com os nomes da primeira colunas
```

```
de growth
    N <- length(x) #criando objeto n com o valor do comprimento de x
    while (N>1){  #enquanto N que igual ao comprimento de x for maior 1,
repetir
      x [N] <- x[N-1] #atribuindo o nome da penultima coluna para a ultima
coluna
      N = N-1 #no proximo ciclo sera subtraido 1 de n
      }
    }
  ####arrumando os nomes
  x [1] <- "year" #trocando o nome da primeira posicao da variavel
  names (growth) <- x #atribuindo o nomes corretos a planilha
  #######PREPARANDO OBJETOS PARA OS CICLOS E ANALISES
  a = ncol (growth) #criando vetor a com numero de colunas de growth
  ages <- rep(0, (ncol(growth))-1) # cria um vetor ages com 0 = numero de
colunas menos 1
  while (a>1) { #enquanto a foir maior que 1
    a.vetor <- growth [,a] #atribuindo o valor da coluna para o vetor
    h <- length(a.vetor) #criando objeto h com o comprimento do vetor
    while (is.na (a.vetor[h])){ #enquanto encontrar NA
      a.vetor <- a.vetor [-h] #remover essa posicao
      h= h-1 #no proximo ciclo sera subtraido 1 de h
      }
    a.tamanho <- length(a.vetor) #calculando o tamanho do a.vetor
    ages [a] <- a.tamanho #colocando os dados na posicao a do vetor ages
    a = a-1 #no proximo ciclo sera subtraido 1 de a
    }
  ages <- ages[-1] #removendo 0 que fica na primeira posicao
  ############## 2. ANALISES  ########################
  ##########crescimento acumulado, que sera o valor da ultima posicao de
cada coluna (ano 1 da arvore) somado a seguinte (ano 2), ate o primeiro
valor.
  accumulated.growth <- rev (growth[nrow(growth):1,]) #invertendo a tabela
  accumulated.growth [is.na(accumulated.growth)] <- 0 #trocando na por 0
  accumulated.growth <- cumsum (accumulated.growth) #calculo do crescimento
acumulado
  accumulated.growth [accumulated.growth == 0] <- NA #colocando NA
  accumulated.growth <- rev
(accumulated.growth[nrow(accumulated.growth):1,]) #invertendo
  if (hm == "south" && year [1] > 7){
    accumulated.growth$year <- seq (from = year [2]-1, by = -1, length.out =
nrow (accumulated.growth)) #colocando os anos na primeira coluna se year
estiver nas condicoes esecificidas
    } else { accumulated.growth$year <- seq (from = year [2], by = -1,
length.out = nrow (accumulated.growth)) #colocando os anos na primeira
coluna caso nao precise subtrair 1 de year [2]
    }

  ##########padrao de crescimento
  growth2 <- read.table (data, header = TRUE, sep = "\t", dec = ".")  #lendo
a planilha data com cabecalho, separada por tabulacao e ponto como decimal
```

```
  lifetime.growth.pattern <- rev (growth2[nrow(growth2):1,]) # Inverter
todos os valores
  lifetime.growth.pattern$last <- NaN #criando uma coluna com Nan
  Y <- ncol (lifetime.growth.pattern) #criando objeto Y com valor do numero
de colunas de lifetime.growth.pattern
  while (Y>1) { #enquanto Y - numero de colunas - for maior 1, repetir
    lifetime.growth.pattern [ ,Y] <- lifetime.growth.pattern [ ,(Y-1)]
#atribuindo o valor da penultima coluna para a ultima coluna
    Y <- Y-1 #no proximo ciclo sera subtraido 1 de z
    }
  lifetime.growth.pattern [,1] <- seq (from = 0, by = 1, length.out = nrow
(lifetime.growth.pattern)) #colocando as idades das amostras na primeira
coluna
  nam <- names (lifetime.growth.pattern) #criando objeto nan com os nomes da
primeira colunas de lifetime.growth.pattern
  n <- length(nam) #criando objeto n com o valor do comprimento de nam
  while (n>1){  #enquanto n - comprimento de nam - for maior 1, repetir
    nam [n] <- nam[n-1] #atribuindo o nome da penultima coluna para a ultima
coluna
    n = n-1 #no proximo ciclo sera subtraido 1 de
    }
  nam [1] <- "CambialAge" #trocando o nome da primeira posicao da variavel
  names (lifetime.growth.pattern) <- nam #atribuindo o nomes corretos a
planilha
  # excluindo NAs
  z = ncol (lifetime.growth.pattern) #criando vetor com numero de colunas de
lifetime
  while (z>1) { #enquanto z for maior que 1
    vetor <- lifetime.growth.pattern [,z] #atribui o valor dessa posicao de
lifetime para o vetor
    i=0 #i gual a 0
    while (is.na (vetor[1])){ #enquanto essa posicao do vetor conter na
      vetor <- vetor [-1] #retirar essa posicao
      i = i+1 #no proximo ciclo sera somado 1 ao valor de 1
      }
    vetor <- c(vetor, rep(NA,i)) #cria um vetor com na repetindo i vezes
    lifetime.growth.pattern [,z] <- vetor #inserindo esse vetor na posicao
    z = z-1 #no proximo ciclo sera subtraido 1 de z
    }
  ###############
  #############3. SAIDA/GRAFICOS################
  pdf(paste (file, "exploratory" , data, ".pdf"), width = 10, height = 8)
#cria um pdf com o titulo inserido em file,"exploratory",data
  par (mfrow = c (2, 2)) # Dispositivo tera duas linhas e duas colunas.
  hist (ages, breaks = histogram,  #histograma da frequencia das idades,
calculado a partir de ages, pelo intervalo indicado em histogram
        xlab = "Age (years)", ylab = "Frequency", main = "Age distribution",
#dando nome para eixo x,eixo y e titulo
        tcl = -0.3, cex = 0.8, col = "gray")
  boxplot (ages, main = "Ages (years)", # boxplot com ages e titulo
```

```
           cex = 0.8, col = "gray")
  accu.grow.plot = melt(accumulated.growth, id.vars="year")
  mod <- lm(accu.grow.plot$value~accu.grow.plot$year)  # calcula o modelo
linear
  plot (accu.grow.plot$year, accu.grow.plot$value, col="gray", type="l",main
= "Population Growth", cex = 0.8, xlab = "Year (years)", ylab = "Accumulated
growth", bty = "l")
  abline (coef(mod), lwd = 1.75)
  grow.patt.plot = melt(lifetime.growth.pattern, id.vars="CambialAge")
  scatter.smooth (grow.patt.plot$CambialAge, grow.patt.plot$value, span =
2/3, degree = 1.5,
                  xlab = "Cambial age (years)",
                  ylab = "Annual Increment",
                  bty = "l", lty = 1, cex = 0.8, main = "Lifetime Growth
Pattern") # dando nome para eixo x, eixo y e colocando linhas apenas nas
margens 1 e 2.
  dev.off() #fecha o dispositivo de tela
  write.table(accumulated.growth, (paste (file, "Accumulated.growth.txt")),
col.names = TRUE, sep = "\t") #salvando os dataframes gerados com o nome
inserido em file seguido do nome do objeto
  write.table(lifetime.growth.pattern, (paste (file,
"Lifetime.growth.pattern.txt")), col.names = TRUE, sep = "\t")
  write.table(growth, (paste (file, "Growth.txt")), col.names = TRUE, sep =
"\t")
  }
```

Página de help

```
tReegrwoth              package:unknown                R Documentation

Explorando medidas de anéis de crescimento


Description:

     Uma função para análise exploratória de medidas de anéis de
crescimento. A partir frames com as amostras distribuídas por colunas, a
função entrega análises básicas da estrutura etária da população (histograma
de idades e boxplot), taxa de crescimento (gráfico de crescimento acumulado)
e padrão de crescimento (scatterplot de crescimento por idade).


Usage:

     tReegrwoth (data, year, hm, histogram, return = FALSE, file)


Arguments:
```

```
data:  nome do documento .txt ou .csv em que os dados de crescimento estão.
Aceita data frames que contenham uma amostra por coluna.
Primeira linha contendo o nome das amostras.

year: um vetor com dois valores, de mês e ano em que a coleta foi feita.

hm: em qual hemisfério a coleta foi feita ( “south” ou “north” ).

histogram: um vetor com pelo menos a idade inicial, média e final para
definir o intervalo desejado no histograma.

return: TRUE retorna os data.frames e um .pdf  dos gráficos gerados.

file: o radical para nome dos arquivos retornados


Details:

data é o seu arquivo com cada amostra em uma coluna, onde cada linha é um
valor de incremento anual.

year exemplo de entrada: c(6, 2018). Se o argumento hm for igual a “south” e
o mês colocado for maior do que 7, será subtraído um no ano. Isso é
necessário já que a estação de crescimento no hemisféio sul começa no meio
do ano.

histogram exemplo de entrada: seq(0, 300, 25). Esse argumento define o
argumento “breaks” na função hist que constrói o histograma com a frequência
de idades.

file é indicado para facilitar a manipulação de diversas análises
independentes, adicionando um prefixo ao nome do arquivo colocado em data.
Quando apenas a pasta é indicada, a função criará uma pasta para cada
dataframe, copiando o nome do arquivo.

Os arquivos .txt são retornados para facilitar uma futura análise caso a
análise exploratória aponte resultados promissores.

Value:

Histograma com distribuição de idades;

Boxplot da idade das amostras;

Gráfico de crescimento acumulado de todas as amostras, com reta de
regressão;

Scatterplot mostrando padrão de crescimento ao longo da vida das amostras e
linha LOESS para facilitar a interpretação do comportamento da população;
```

```
Se return = TRUE, retorna os data.frames “file, “Growht.txt”” “file,
“Accumulated.growth.txt””, “file, “Lifetime.growth.pattern”” e os gráficos
acima em um painel (2,2) salvos em um .pdf “file, "exploratory" , data,
".pdf"”.


Warning:

     Se a planilha de dados ou alguns dos argumentos não estiverem no
formato correto a função não é executada.


Author(s):

Milena de Godoy-Veiga
Milena.gveiga@gmail.com



Example:

    exdata <- data.frame(ind1 = c(rnorm(13,3,1.2), rep(NA,54)), ind2 =
c(rnorm(45.3,4,1.2), rep(NA,22)), ind3 = c(rnorm(67,3.9,1.2)),ind4 =
c(rnorm(23.3,1.1),rep(NA,44)),ind5 = c(rnorm(27,3.3,1.2), rep(NA,40)),
stringsAsFactors = FALSE )

    write.table(exdata, (paste ("exdata.txt")), col.names = TRUE, sep =
"\t")

    tReegrowth("exdata.txt", c(10, 2000), "south", seq(0, 100, 10), retornar
= TRUE, "TESTE")
```

Arquivo da função: tReegrowth

Página de HELP: Help