

Aula 01 - Parte 2 da prática com R

Reproduza os exercícios
no ambiente
RStudio
e/ou
Jupyter Notebook

Ler arquivo Excel

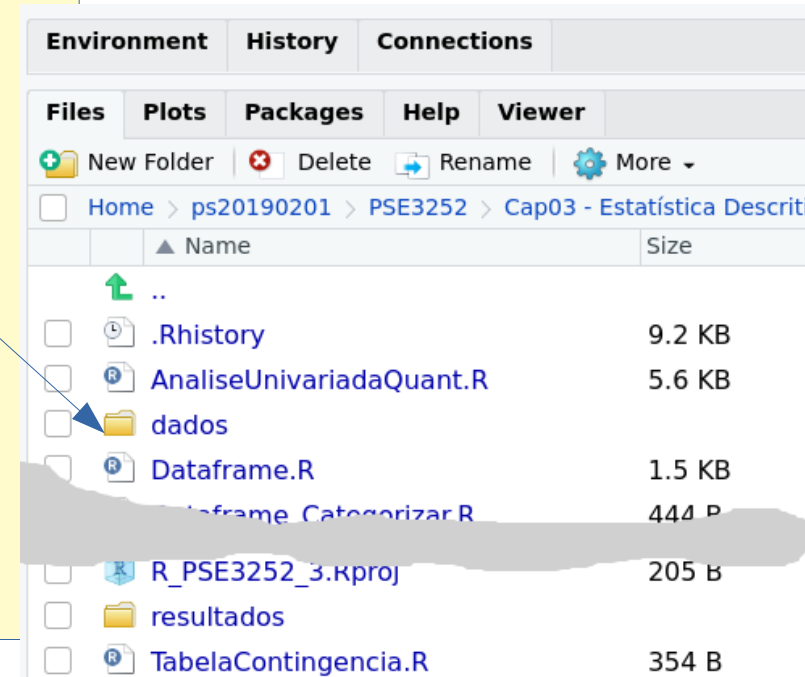
Vamos utilizar outro arquivo (Adm2008.xlsx) porque precisamos de um exemplo com maior número de indivíduos.

Dataframe_Leitura.R

```
# Dataframe_Leitura.R  
# le um arquivo em formato Excel e  
# armazena em um dataframe chamado Dados  
  
library(readxl)  
Dados <- read_excel(file.path("dados", "Adm2008.xlsx"))  
View(Dados)
```

Adm2008.xlsx

	A	B	C	D	E	F	G	H
1	Nome	Genero	Estatura	MCT	Idade	Aprovado	Conceito	Idade_Cat
2	Beatriz	Feminino	1.61	53	20	Sim	A	20 - 20
3	Camila	Feminino	1.56	50	21	Sim	B	21 - 21
4	Christiane	Feminino	1.72	60	20	Sim	C	20 - 20
5	Debora	Feminino	1.57	44	21	Sim	A	21 - 21
6	Denise	Feminino	1.68	57	19	Sim	B	< 19
7	Elaine	Feminino	1.69	60	22	Sim	C	22 - 25



Ler arquivo Excel

Vamos utilizar outro arquivo (Adm2008.xlsx) porque precisamos de um exemplo com maior número de indivíduos.

```
Files Plots Packages Help Viewer
R: Read xls and xlsx files Find in Topic
read_excel {readxl} R Documentation

Read xls and xlsx files

Description
Read xls and xlsx files

read_excel() calls excel_format() to determine if path is xls or xlsx, based on the file extension and the
file itself, in that order. Use read_xls() and read_xlsx() directly if you know better and want to prevent
such guessing.

Usage
read_excel(path, sheet = NULL, range = NULL, col_names = TRUE,
  col_types = NULL, na = "", trim_ws = TRUE, skip = 0,
  n_max = Inf, guess_max = min(1000, n_max),
  progress = readxl_progress(), .name_repair = "unique")
read_xls(path, sheet = NULL, range = NULL, col_names = TRUE,
  col_types = NULL, na = "", trim_ws = TRUE, skip = 0,
  n_max = Inf, guess_max = min(1000, n_max),
  progress = readxl_progress(), .name_repair = "unique")
read_xlsx(path, sheet = NULL, range = NULL, col_names = TRUE,
  col_types = NULL, na = "", trim_ws = TRUE, skip = 0,
  n_max = Inf, guess_max = min(1000, n_max),
  progress = readxl_progress(), .name_repair = "unique")

Arguments
path      Path to the xls/xlsx file.
sheet     Sheet to read. Either a string (the name of a sheet), or an integer (the position of the
          sheet). Ignored if the sheet is specified via range. If neither argument specifies the sheet,
          defaults to the first sheet.
range     A cell range to read from, as described in cell-specification. Includes typical Excel ranges
```

```
Console Terminal x
~/ps20200114/MSP1290_202
> library(readxl)
> ?read_excel
> |
```



Examples

```
datasets <- readxl_example("datasets.xlsx")
read_excel(datasets)

# Specify sheet either by position or by name
read_excel(datasets, 2)
read_excel(datasets, "mtcars")

# Skip rows and use default column names
read_excel(datasets, skip = 148, col_names = FALSE)

# Recycle a single column type
read_excel(datasets, col_types = "text")

# Specify some col_types and guess others
read_excel(datasets, col_types = c("text", "numeric", "numeric", "numeric", "numeric"))
```

Ler arquivo Excel

Dados x Dataframe_Leitura.R x								
Filter								
	Nome	Genero	Estatura	MCT	Idade	Aprovado	Conceito	Idade_Cat
1	Beatriz	Feminino	1.61	53	20	Sim	A	20 - 20
2	Camila	Feminino	1.56	50	21	Sim	B	21 - 21
3	Christiane	Feminino	1.72	60	20	Sim	C	20 - 20
4	Debora	Feminino	1.57	44	21	Sim	A	21 - 21
5	Denise	Feminino	1.68	57	19	Sim	B	<= 19
6	Elaine	Feminino	1.69	60	22	Sim	C	22 - 25
7	Elisa	Feminino	1.70	65	24	Sim	A	22 - 25
8	Ermínia	Feminino	1.64	58	22	Sim	B	22 - 25
9	Fabiana	Feminino	1.70	50	20	Sim	C	20 - 20
10	Flávia	Feminino	1.57	55	19	Sim	A	<= 19
11	Gisele	Feminino	1.61	60	21	Sim	B	21 - 21
12	Júlia	Feminino	1.67	64	20	Sim	C	20 - 20
13	Juliana	Feminino	1.70	59	21	Sim	A	21 - 21
14	Jully	Feminino	1.64	52	21	Sim	B	21 - 21

Showing 1 to 15 of 89 entries

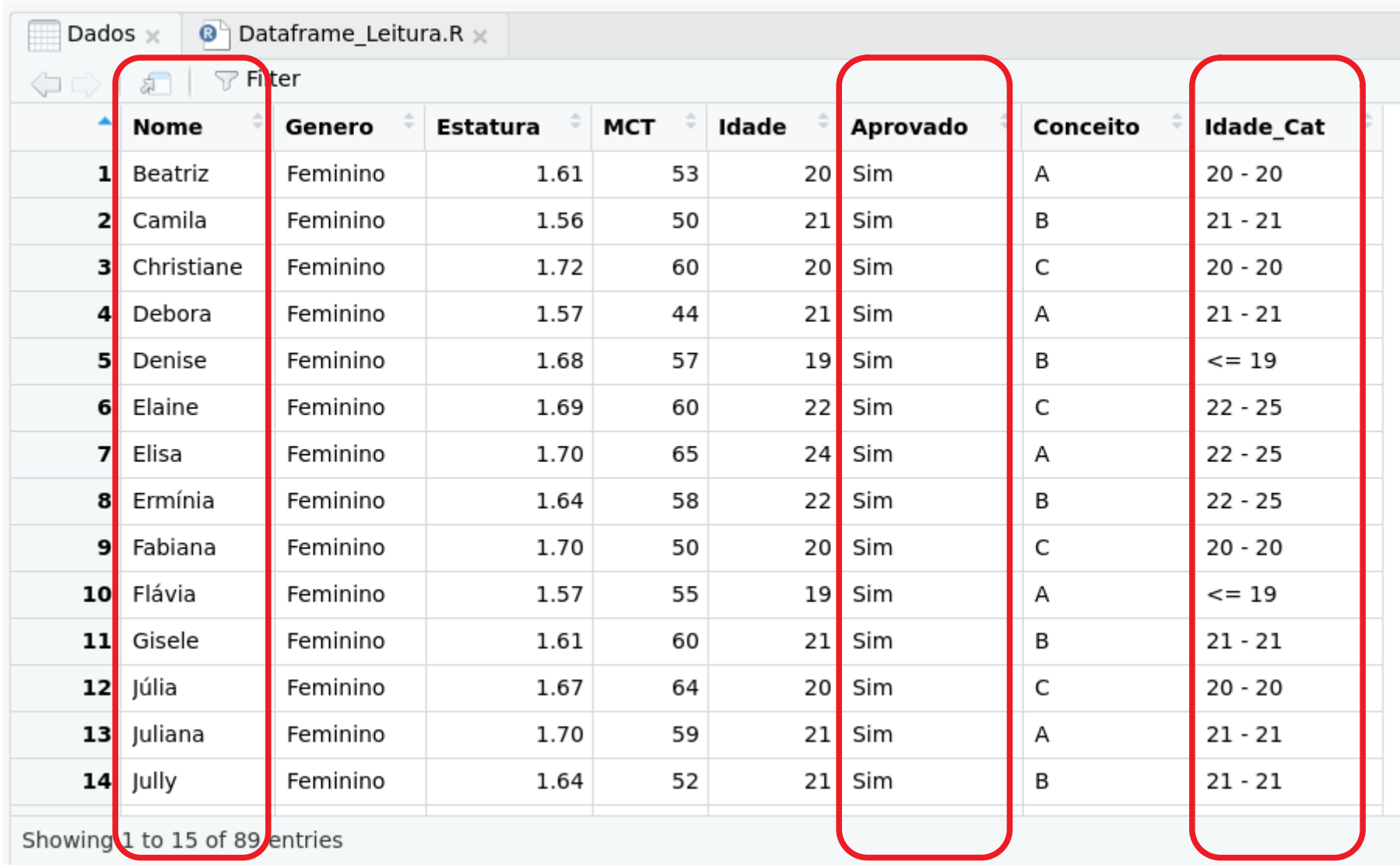
Excluir variável de dataframe

Dataframe_ExcluirVariavel.R

```
# Dataframe_ExcluirVariavel.R
# abre planilha Excel, remove colunas e
# salva planilha Excel com outro nome

library(openxlsx)
library(readxl)
Dados <- readxl::read_excel(file.path("dados", "Adm2008.xlsx"))
View(Dados)
Dados[, "Nome"] <- NULL
Dados[, "Aprovado"] <- NULL
Dados[, "Idade_Cat"] <- NULL
View(Dados)
openxlsx::write.xlsx(Dados, file.path("dados", "Adm2008_v2.xlsx"))
```

Excluir variável de dataframe



A screenshot of a data table interface. The table has 9 columns: 'Nome', 'Genero', 'Estatura', 'MCT', 'Idade', 'Aprovado', 'Conceito', and 'Idade_Cat'. The first column, 'Nome', is highlighted with a red box. The seventh column, 'Aprovado', and the eighth column, 'Idade_Cat', are also highlighted with red boxes. The table contains 14 rows of data. At the bottom, it says 'Showing 1 to 15 of 89 entries'.

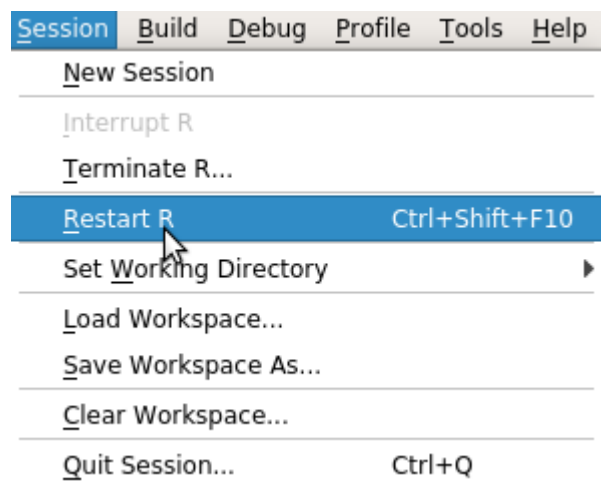
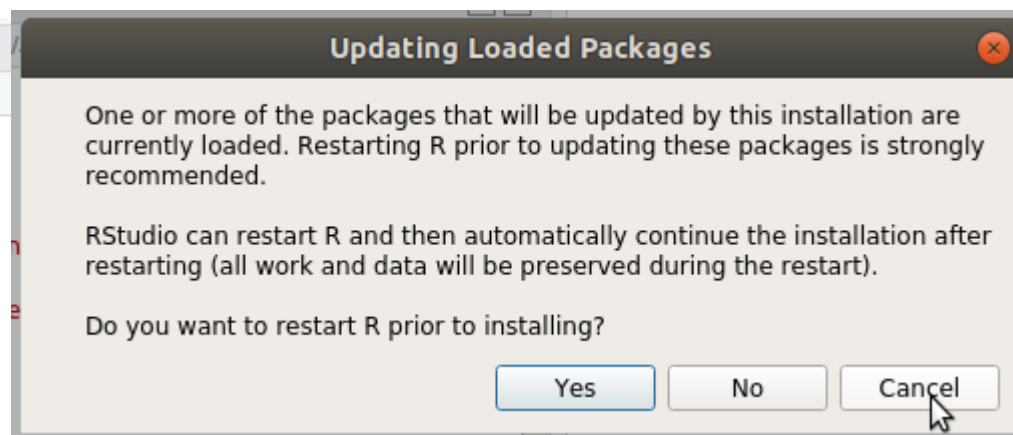
	Nome	Genero	Estatura	MCT	Idade	Aprovado	Conceito	Idade_Cat
1	Beatriz	Feminino	1.61	53	20	Sim	A	20 - 20
2	Camila	Feminino	1.56	50	21	Sim	B	21 - 21
3	Christiane	Feminino	1.72	60	20	Sim	C	20 - 20
4	Debora	Feminino	1.57	44	21	Sim	A	21 - 21
5	Denise	Feminino	1.68	57	19	Sim	B	<= 19
6	Elaine	Feminino	1.69	60	22	Sim	C	22 - 25
7	Elisa	Feminino	1.70	65	24	Sim	A	22 - 25
8	Ermínia	Feminino	1.64	58	22	Sim	B	22 - 25
9	Fabiana	Feminino	1.70	50	20	Sim	C	20 - 20
10	Flávia	Feminino	1.57	55	19	Sim	A	<= 19
11	Gisele	Feminino	1.61	60	21	Sim	B	21 - 21
12	Júlia	Feminino	1.67	64	20	Sim	C	20 - 20
13	Juliana	Feminino	1.70	59	21	Sim	A	21 - 21
14	Jully	Feminino	1.64	52	21	Sim	B	21 - 21

Showing 1 to 15 of 89 entries

Instalando *packages*



```
Console Terminal x
~/ps20200114/MSP1290_2020/Aula 1 - Estatística Descritiva/Adm_R/
> install.packages("openxlsx")
Error in install.packages : Updating loaded packages
> |
```



```
Console Terminal x
~/ps20200114/MSP1290_2020/Aula 1 - Estatística Descritiva/Adm_R/
> install.packages("openxlsx")
Error in install.packages : Updating loaded packages

Restarting R session...
```

Instalando *packages*



```
Console Terminal x
~/ps20200114/MSP1290_2020/Aula 1 - Estatística Descritiva/Adm_R/
> install.packages("openxlsx")
Error in install.packages : Updating loaded packages

Restarting R session...

> install.packages("openxlsx")
Installing package into '/home/silveira/R/x86_64-pc-linux-gnu-library/3.6'
(as 'lib' is unspecified)
trying URL 'https://cloud.r-project.org/src/contrib/openxlsx_4.1.4.tar.gz'
Content type 'application/x-gzip' length 1383224 bytes (1.3 MB)
=====
downloaded 1.3 MB

* installing *source* package 'openxlsx' ...
** package 'openxlsx' successfully unpacked and MD5 sums checked
mv: cannot move '/home/silveira/R/x86_64-pc-linux-gnu-library/3.6/openxlsx' to '/home/silveira/R/x86_64-pc-linux-gnu-library/3.6/00LOCK-openxlsx/openxlsx': Permission denied
ERROR: cannot remove earlier installation, is it in use?
* removing '/home/silveira/R/x86_64-pc-linux-gnu-library/3.6/openxlsx'
Warning in install.packages :
  installation of package 'openxlsx' had non-zero exit status

The downloaded source packages are in
  '/tmp/RtmpySiasY/downloaded_packages'
> |
```





Instalando *packages*



```
silveira@silveira: ~  
File Edit View Search Terminal Help  
silveira@silveira:~$ sudo R  
[sudo] password for silveira:  
  
R version 3.6.2 (2019-12-12) -- "Dark and Stormy Night"  
Copyright (C) 2019 The R Foundation for Statistical Computing  
Platform: x86_64-pc-linux-gnu (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.  
  
Natural language support but running in an English locale  
  
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.  
  
> install.packages("openxlsx")
```

Instalando *packages*



```
silveira@silveira: ~
File Edit View Search Terminal Help
silveira@silveira:~$ sudo R
[sudo] password for silveira:

R version 3.6.2 (2019-12-12) -- "Dark and Stormy Night"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (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.

  Natural language support but running in an English locale

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.

> install.packages("openxlsx")
Installing package into '/home/silveira/R/x86_64-pc-linux-gnu-library/3.6'
(as 'lib' is unspecified)
trying URL 'https://cloud.r-project.org/src/contrib/openxlsx_4.1.4.tar.gz'
Content type 'application/x-gzip' length 1383224 bytes (1.3 MB)
=====
downloaded 1.3 MB

* installing *source* package 'openxlsx' ...
** package 'openxlsx' successfully unpacked and MD5 sums checked
** using staged installation
** libs
```

Natural language support but running in an English locale

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.

```
> install.packages("openxlsx")
=====
downloaded 1.3 MB

* installing *source* package 'openxlsx' ...
** package 'openxlsx' successfully unpacked and MD5 sums checked
** using staged installation
** libs
g++ -std=gnu++11 -I"/usr/share/R/include" -DNDEBUG -I"/home/silveira/R/x86_64-pc-l
inux-gnu-library/3.6/Rcpp/include" -fpic -g -O2 -fdebug-prefix-map=/build/r-base
-t3diwe/r-base-3.6.2=. -fstack-protector-strong -Wformat -Werror=format-security -W
date-time -D_FORTIFY_SOURCE=2 -g -c RcppExports.cpp -o RcppExports.o
g++ -std=gnu++11 -I"/usr/share/R/include" -DNDEBUG -I"/home/silveira/R/x86_64-pc-l
inux-gnu-library/3.6/Rcpp/include" -fpic -g -O2 -fdebug-prefix-map=/build/r-base
-t3diwe/r-base-3.6.2=. -fstack-protector-strong -Wformat -Werror=format-security -W
date-time -D_FORTIFY_SOURCE=2 -g -c helper_functions.cpp -o helper_functions.o
g++ -std=gnu++11 -I"/usr/share/R/include" -DNDEBUG -I"/home/silveira/R/x86_64-pc-l
inux-gnu-library/3.6/Rcpp/include" -fpic -g -O2 -fdebug-prefix-map=/build/r-base
-t3diwe/r-base-3.6.2=. -fstack-protector-strong -Wformat -Werror=format-security -W
date-time -D_FORTIFY_SOURCE=2 -g -c load_workbook.cpp -o load_workbook.o
gcc -std=gnu99 -I"/usr/share/R/include" -DNDEBUG -I"/home/silveira/R/x86_64-pc-lin
ux-gnu-library/3.6/Rcpp/include" -fpic -g -O2 -fdebug-prefix-map=/build/r-base-t
3diwe/r-base-3.6.2=. -fstack-protector-strong -Wformat -Werror=format-security -Wda
```

■ ■ ■

```
* DONE (openxlsx)

The downloaded source packages are in
  '/tmp/RtmpbRT0Mq/downloaded_packages'
> 
```



Excluir variável de dataframe

Dataframe_ExcluirVariavel.R

```
# Dataframe_ExcluirVariavel.R  
# abre planilha Excel, remove colunas e  
# salva planilha Excel com outro nome
```

```
library(openxlsx)
```

```
library(readxl)
```

```
Dados <- readxl::read_excel(file.path("dados", "Adm2008.xlsx"))
```

```
View(Dados)
```

```
Dados[, "Nome"] <- NULL
```

```
Dados[, "Aprovado"] <- NULL
```

```
Dados[, "Idade_Cat"] <- NULL
```

```
View(Dados)
```

```
openxlsx::write.xlsx(Dados, file.path("dados", "Adm2008_v2.xlsx"))
```



retomando

Excluir variável de dataframe

Dados

Dataframe_ExcluirVariavel.R

Filter

	Genero	Estatura	MCT	Idade	Conceito
1	Feminino	1.61	53	20	A
2	Feminino	1.56	50	21	B
3	Feminino	1.72	60	20	C
4	Feminino	1.57	44	21	A
5	Feminino	1.68	57	19	B
6	Feminino	1.69	60	22	C
7	Feminino	1.70	65	24	A
8	Feminino	1.64	58	22	B
9	Feminino	1.70	50	20	C
10	Feminino	1.57	55	19	A
11	Feminino	1.61	60	21	B
12	Feminino	1.67	64	20	C
13	Feminino	1.70	59	21	A
14	Feminino	1.64	52	21	B

Showing 1 to 15 of 89 entries

Uma olhada rápida nos dados

Dataframe_Descritiva.R

```
# Dataframe_Descritiva.R  
# importa planilha Excel e  
# mostra uma visao geral dos dados
```

```
library(readxl)  
Dados <- readxl::read_excel(file.path("dados", "Adm2008_v2.xlsx"))
```

```
colunas <- names(Dados)  
cat ("\nnomes das colunas:\n")  
print (colunas)
```

```
cat ("\nestrutura do dataframe:\n")  
str(Dados)
```

```
cat ("\nvisao geral dos dados:\n")  
sumario <- summary(Dados)  
print (sumario)
```

Uma olhada rápida nos dados

Dataframe_Descritiva.R

nomes das colunas:

```
[1] "Genero"      "Estatura" "MCT"        "Idade"      "Conceito"
```

estrutura do dataframe:

Classes 'tbl_df', 'tbl' and 'data.frame': 89 obs. of 5 variables:

```
$ Genero : chr "Feminino" "Feminino" "Feminino" "Feminino" ...
$ Estatura: num  1.61 1.56 1.72 1.57 1.68 1.69 1.7 1.64 1.7 1.57 ...
$ MCT     : num   53 50 60 44 57 60 65 58 50 55 ...
$ Idade   : num   20 21 20 21 19 22 24 22 20 19 ...
$ Conceito: chr   "A" "B" "C" "A" ...
```

visao geral dos dados:

Genero	Estatura	MCT	Idade	Conceito
Length:89	Min. : 1.500	Min. : 43	Min. :18.00	Length:89
Class :character	1st Qu.: 1.640	1st Qu.: 55	1st Qu.:20.00	Class :character
Mode :character	Median : 1.710	Median : 65	Median :20.00	Mode :character
	Mean : 3.669	Mean : 66	Mean :21.16	
	3rd Qu.: 1.780	3rd Qu.: 76	3rd Qu.:22.00	
	Max. :176.000	Max. :105	Max. :33.00	

Uma olhada rápida nos dados

Dataframe_Descritiva_Grafico.R

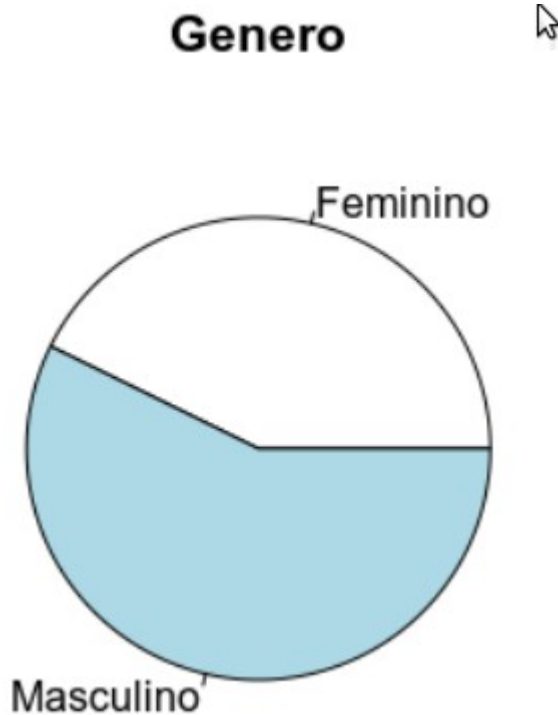
```
# Dataframe_Descritiva_Grafico.R
# importa planilha Excel
# mostra uma visao geral dos dados
# gera boxplot e density plot das variaveis numericas
# gera pie plot e barplot das variaveis em texto

library(readxl)
Dados <- read_excel(file.path("dados","Adm2008_v2.xlsx"))

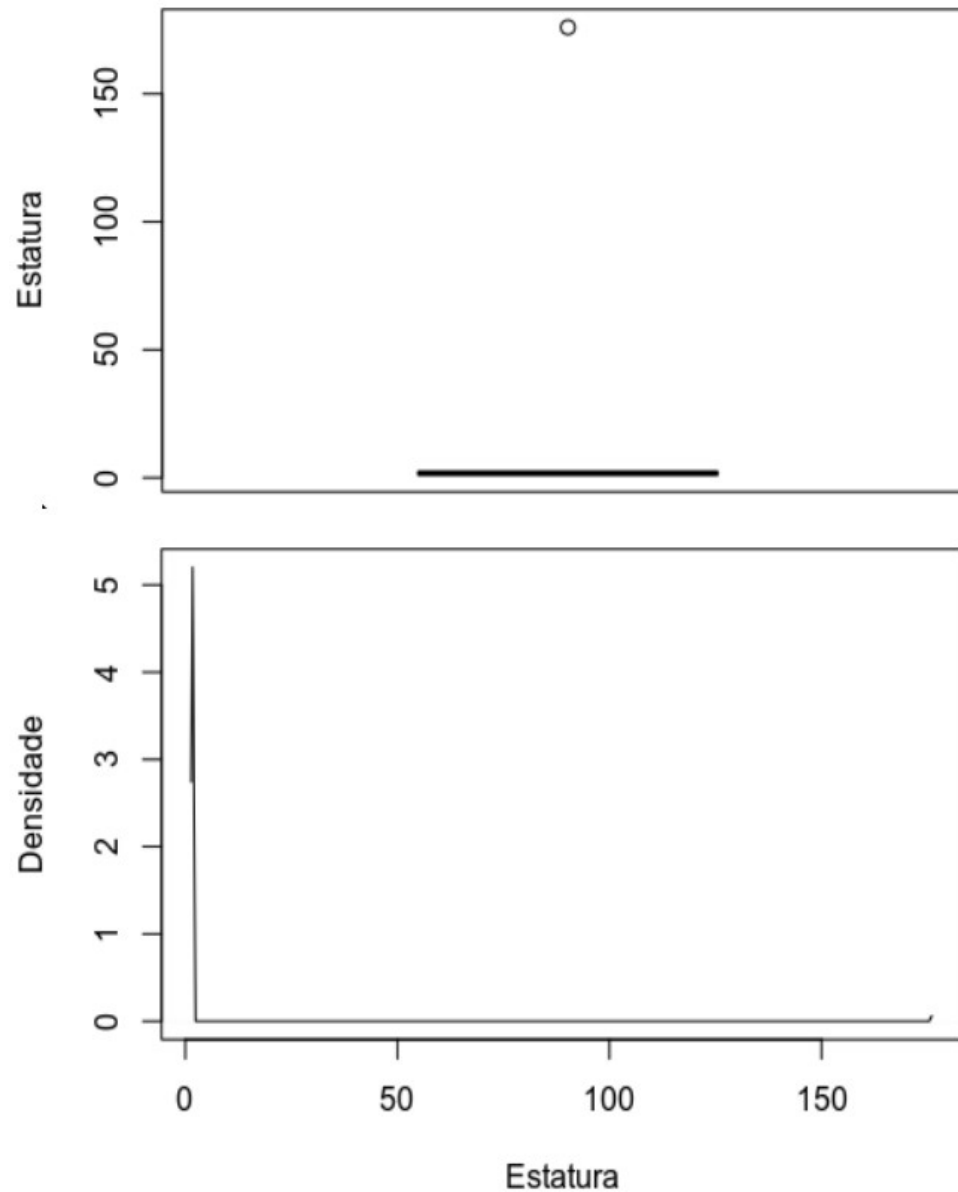
coltipo <- as.vector(sapply(Dados, typeof))
for (i in 1:length(Dados))
{
  # titulo
  grf_titulo_main <- names(Dados)[i]

  # tipos numericos
  if (coltipo[i] == "double" || coltipo[i] == "integer" )
  {
    boxplot (Dados[i], xlab="",ylab=grf_titulo_main)
    dados_densidade <- density(Dados[[i]], na.rm = TRUE)
    plot (dados_densidade, main=NA, xlab=grf_titulo_main, ylab="Densidade")
  }
  # tipos textuais
  if (coltipo[i] == "character")
  {
    dt_col <- table(Dados[i])
    nomes <- names(dt_col)
    fatias <- as.numeric(dt_col)
    pie(fatias, labels=nomes, main=grf_titulo_main)
    barplot(fatias, names.arg = nomes, main=grf_titulo_main)
  }
}
```

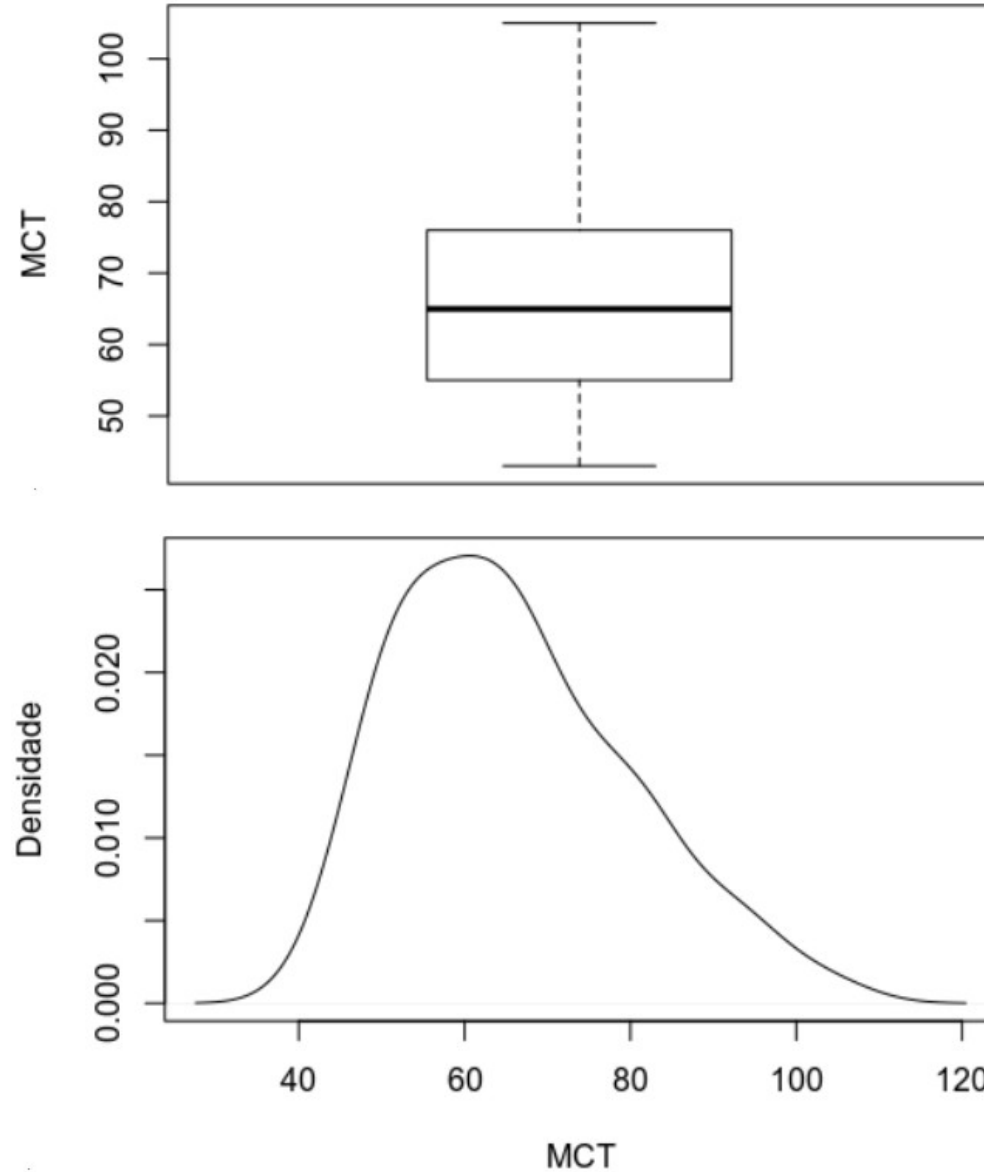

Uma olhada rápida nos dados



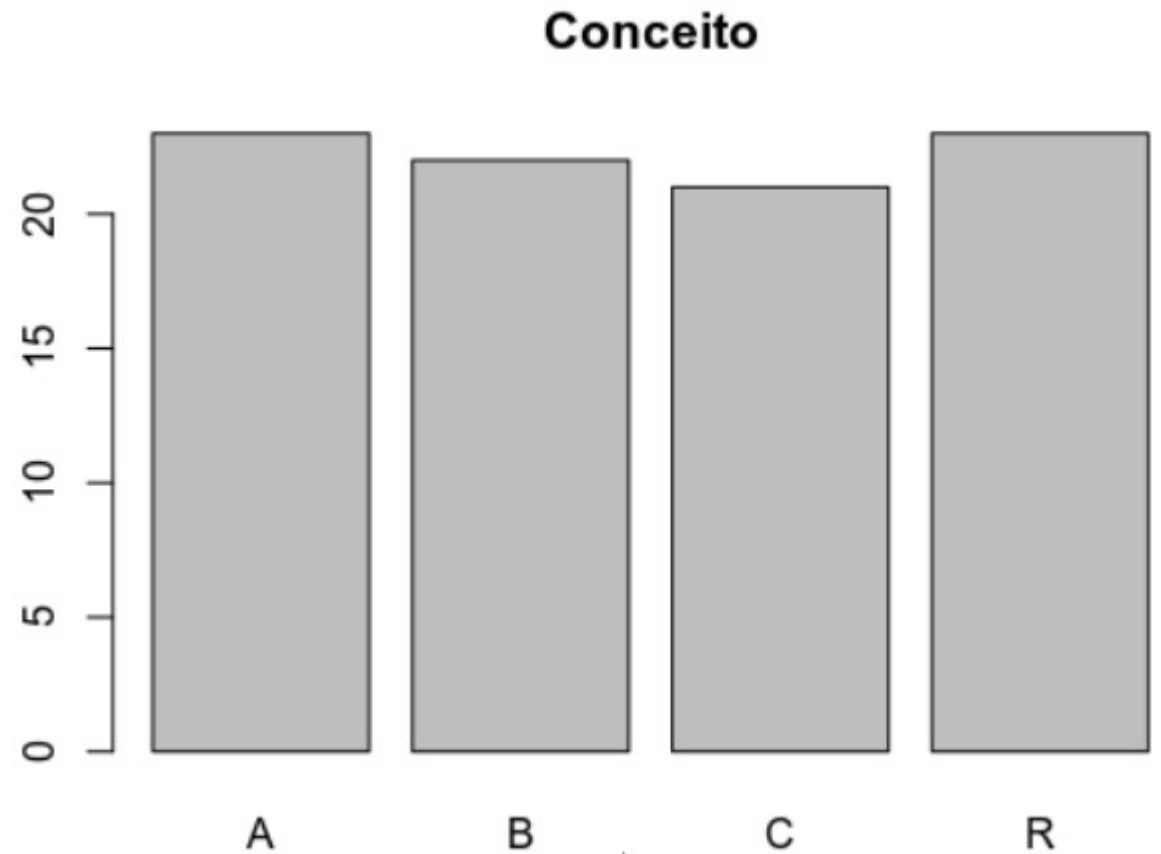
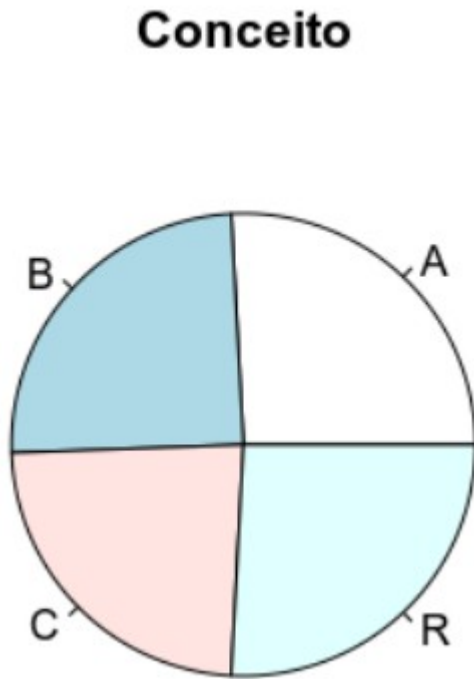
Uma olhada rápida nos dados



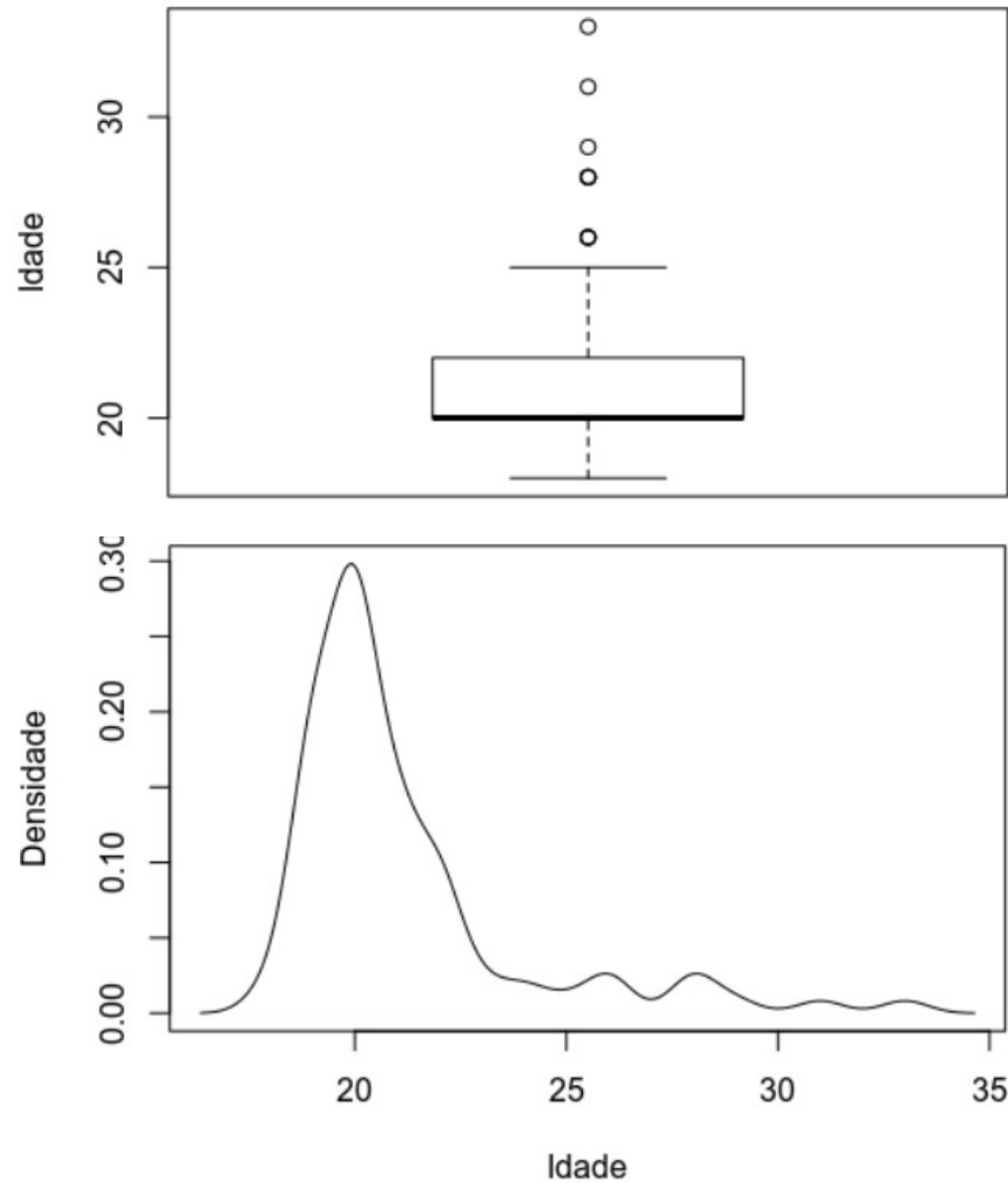
Uma olhada rápida nos dados



Uma olhada rápida nos dados



Uma olhada rápida nos dados

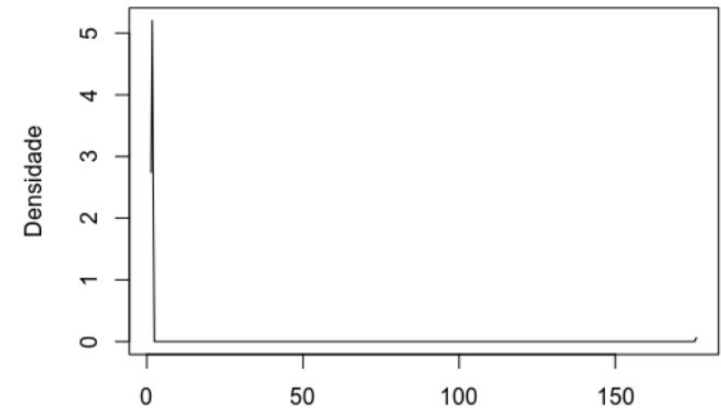


O que está errado nas estaturas?

Estatura	
Min.	: 1.500
1st Qu.:	1.640
Median	: 1.710
Mean	: 3.669
3rd Qu.:	1.780
Max.	:176.000



boxplot

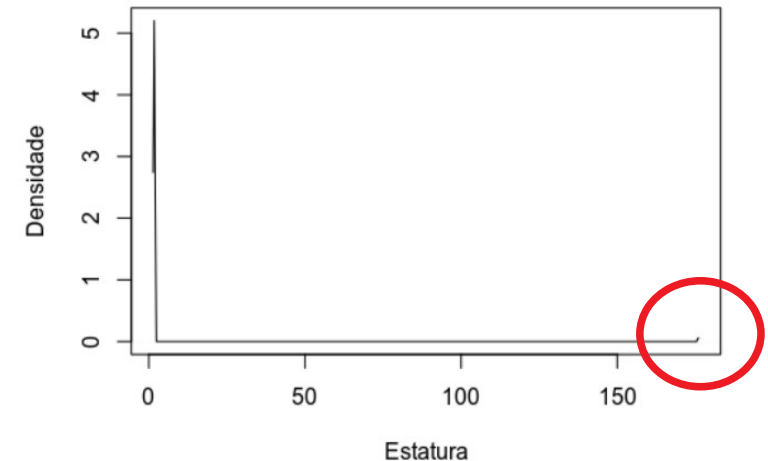
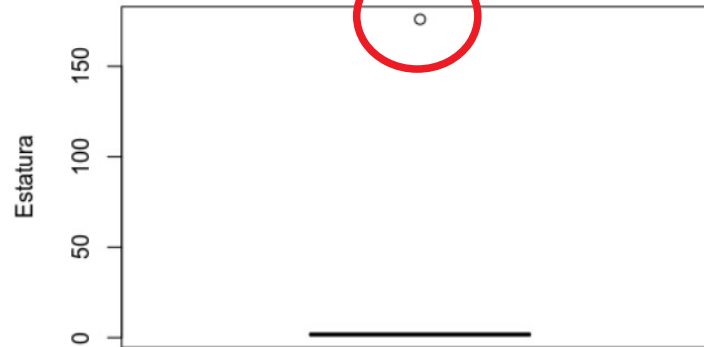


density plot

O que está errado nas estaturas?

(corrigindo usando a Console)

Estatura	
Min.	: 1.500
1st Qu.	: 1.640
Median	: 1.710
Mean	: 3.669
3rd Qu.	: 1.780
Max.	: 176.000



```
> max(Dados$Estatura)
[1] 176
> Dados[Dados$Estatura == max(Dados$Estatura),]
# A tibble: 1 x 5
  Genero    Estatura    MCT Idade Conceito
  <chr>      <dbl> <dbl> <dbl> <chr>
1 Masculino    176     90    20 B
> Dados$Estatura[Dados$Estatura == max(Dados$Estatura)] <- 1.76
> summary(Dados)
```

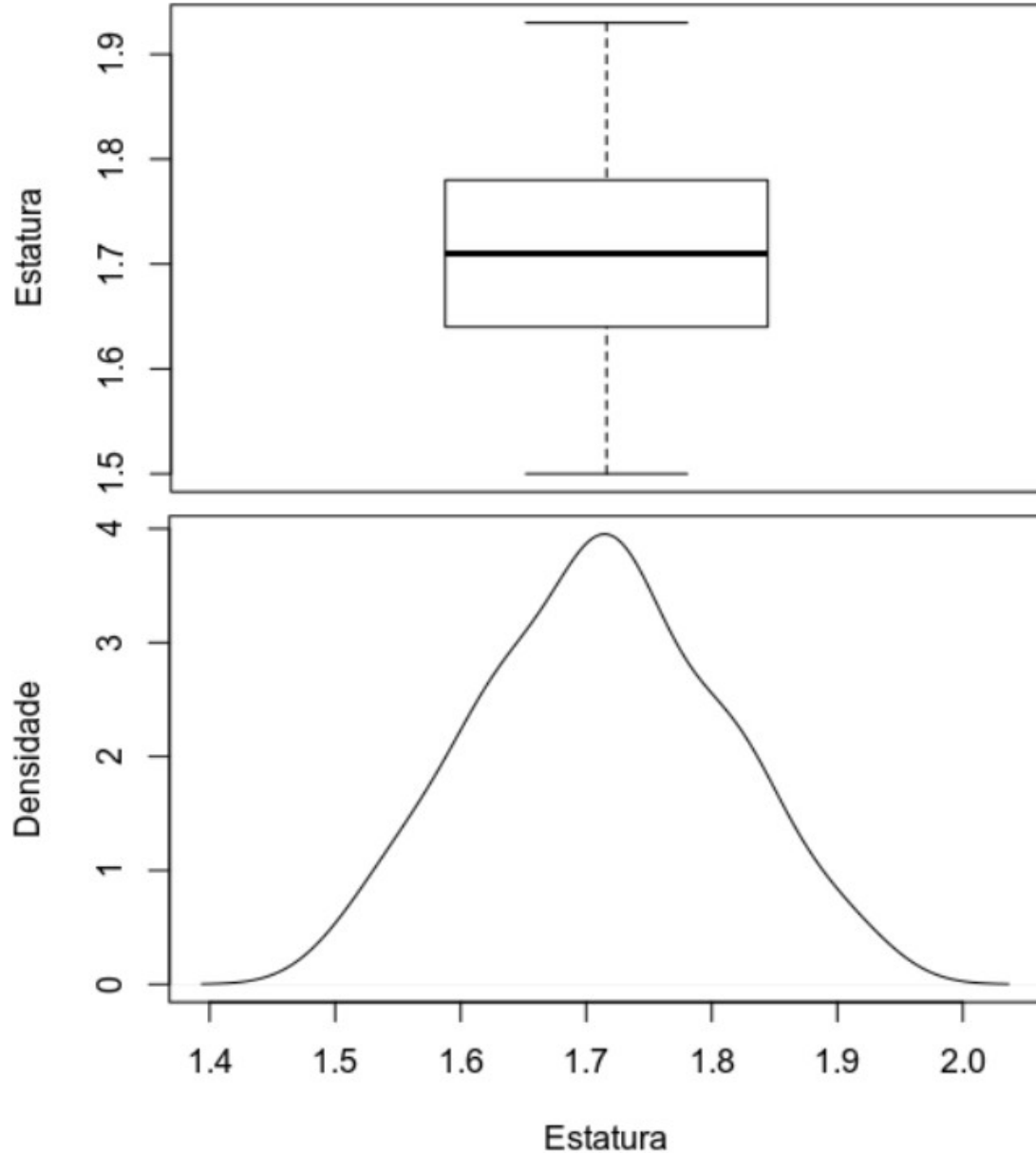
Genero	Estatura	MCT	Idade	Conceito
Length:89	Min. :1.500	Min. : 43	Min. :18.00	Length:89
Class :character	1st Qu.:1.640	1st Qu.: 55	1st Qu.:20.00	Class :character
Mode :character	Median :1.710	Median : 65	Median :20.00	Mode :character
	Mean :1.709	Mean : 66	Mean :21.16	
	3rd Qu.:1.780	3rd Qu.: 76	3rd Qu.:22.00	
	Max. :1.910	Max. :105	Max. :33.00	

```
> write.xlsx(Dados, file.path("dados", "Adm2008_v2.xlsx"))
```

O que está errado nas estaturas?

(corrigindo usando a Console)

boxplot



density plot

Criar variável quantitativa, criar variável categórica e inclui-las em dataframe

Dataframe_Categorizar.R

<http://www.abeso.org.br/>



Associação Brasileira para o Estudo da Obesidade e da Síndrome Metabólica

Entenda seu nível:



Acima de 40,00

Obesidade Grau III

Sinal vermelho! Nessas faixas de IMC o risco de doenças associadas está entre grave e muito grave. Não perca tempo! Busque ajuda profissional já!



35,0 - 39,9

Obesidade Grau II

Sinal vermelho! Nessas faixas de IMC o risco de doenças associadas está entre grave e muito grave. Não perca tempo! Busque ajuda profissional já!



30,0 - 34,9

Obesidade Grau I

Sinal de alerta! Chegou na hora de se cuidar, mesmo que seus exames sejam normais. Vamos dar início a mudanças hoje! Cuide de sua alimentação. Você precisa iniciar um acompanhamento com nutricionista e/ou endocrinologista.



25,0 - 29,9

Sobrepeso/pré- obesidade

Atenção! Você está com sobrepeso. Embora ainda não seja obeso, algumas pessoas já podem apresentar doenças associadas, como diabetes e hipertensão nessa faixa de IMC. Reveja e melhore seus hábitos!



18,6 - 24,9

Peso normal

Parabéns, você está com peso normal, mas é importante que você mantenha hábitos saudáveis de vida para que continue assim.



Abaixo de 18,5

Abaixo do peso

Isso pode ser apenas uma característica pessoal, mas pode, também, ser sinal de desnutrição.

Criar variável quantitativa, criar variável categórica e inclui-las em dataframe

Dataframe_Categorizar.R

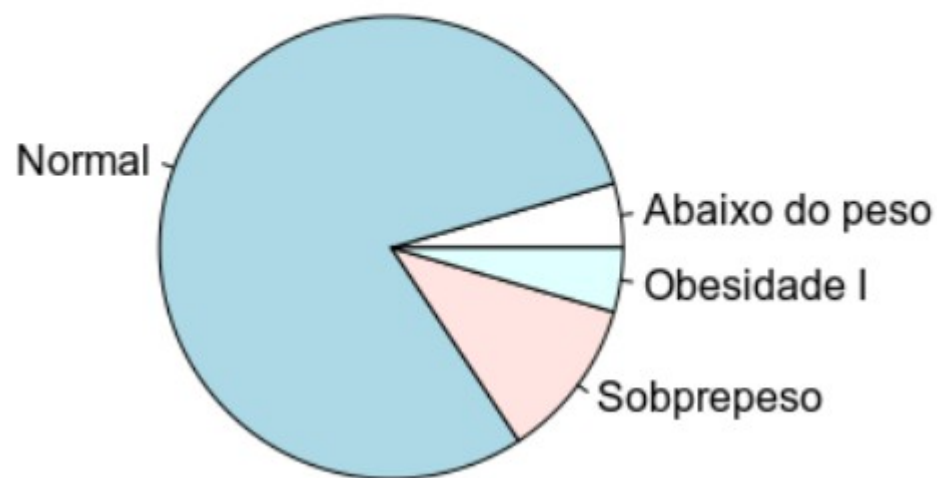
```
library(openxlsx)
library(readxl)
Dados <- read_excel(
  file.path("dados", "Adm2008_v2.xlsx")
)

View(Dados)
Dados$IMC <- Dados$MCT / (Dados$Estatura^2)
classe <- c("Abaixo do peso",
           "Normal", "Sobrepeso",
           "Obesidade I", "Obesidade II",
           "Obesidade III")
pc <- c(0, 18.5, 24.9, 29.9, 34.9, 39.9, +Inf)
Dados$IMC_classe <- cut(Dados$IMC, pc, classe)
View(Dados)
write.xlsx (Dados,
  file.path("dados", "Adm2008_v3.xlsx")
)

# grafico pie (opcional)
dt_col <- table(Dados$IMC_classe)
dt_col <- dt_col[dt_col>0] # elimina classes
com contagem == 0
nomes <- names(dt_col)
fatias <- as.numeric(dt_col)
pie(fatias, labels=nomes, main="Categorias pelo IMC")
```

Abaixo do peso	≤ 18.5
Peso normal	18.6 a 24.9
Sobrepeso	25.0 a 29.9
Obesidade Grau I	30.0 a 34.9
Obesidade Grau II	35.0 a 39.9
Obesidade Grau III	≥ 40.0

Categorias pelo IMC



Criar variável quantitativa, criar variável dicotômica e inclui-las em dataframe

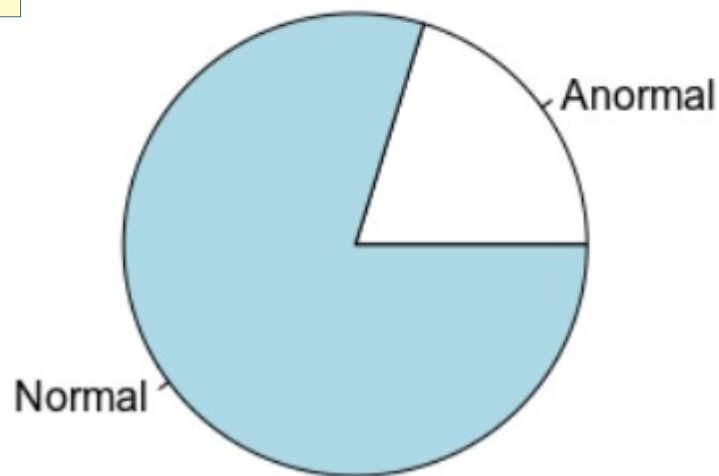
Dataframe_Dicotomizar.R

```
library(openxlsx)
library(readxl)
Dados <- read_excel(file.path("dados","Adm2008_v3.xlsx"))
View(Dados)
classe <- c("Anormal",
            "Normal",
            "Anormal")
pc <- c(0, 18.5, 24.9, +Inf)
Dados$IMC_dicot <- cut(Dados$IMC, pc, classe)
View(Dados)
write.xlsx(Dados, file.path("dados","Adm2008_v4.xlsx"))

# grafico pie (opcional)
dt_col <- table(Dados$IMC_dicot)
dt_col <- dt_col[dt_col>0] # elimina classes com contagem
== 0
nomes <- names(dt_col)
fatias <- as.numeric(dt_col)
pie(fatias, labels=nomes, main="Categorias pelo IMC")
```

Abaixo do peso	≤ 18.5
Peso normal	18.6 a 24.9
Sobrepeso	25.0 a 29.9
Obesidade Grau I	30.0 a 34.9
Obesidade Grau II	35.0 a 39.9
Obesidade Grau III	≥ 40.0

Categorias pelo IMC



Variável nominal em R

Execute na Console do RStudio



```
> genero <- c("M", "M", "M", "M", "F", "F", "f", "F")
> is.factor(genero)
[1] FALSE
> genero.nom <- factor(genero)
> is.factor(genero.nom)
[1] TRUE
> levels(genero.nom)
[1] "f" "F" "M"
> # Lista valores distintos
> genero.nom <- factor(genero, levels=unique(genero))
> levels(genero.nom)
[1] "M" "F" "f"
> # Corrige o 'f'
> genero <- toupper(genero)
> genero.nom <- factor(genero, levels=unique(genero))
> levels(genero.nom)
[1] "M" "F"
> # Altera a ordem das categorias
> genero.nom <- factor(genero, levels=c("F", "M"))
> levels(genero.nom)
[1] "F" "M"
> # Frequencias absolutas das categorias
> table(genero.nom)
genero.nom
F M
4 4
```

Variável ordinal em R

Execute na Console do Rstudio
ou Jupyter Notebook



```
> # Criando fator ordinal
> fxetaria <- c("Idoso", "Jovem", "Adulto", "Jovem",
+             "Idoso", "Adulto", "Joven", "Adulto")
> (categs <- unique(fxetaria))
[1] "Idoso" "Jovem" "Adulto" "Joven"
> (fxetaria.ord <- ordered(fxetaria, levels=categs))
[1] Idoso Jovem Adulto Jovem Idoso Adulto Joven Adulto
Levels: Idoso < Jovem < Adulto < Joven
> is.factor(fxetaria.ord)
[1] TRUE
> levels(fxetaria.ord)
[1] "Idoso" "Jovem" "Adulto" "Joven"
> # resolve o valor com erro de digitacao
> fxetaria <- gsub("Joven", "Jovem", fxetaria)
> (fxetaria.ord <- ordered(fxetaria, levels=unique(fxetaria)))
[1] Idoso Jovem Adulto Jovem Idoso Adulto Jovem Adulto
Levels: Idoso < Jovem < Adulto
> is.factor(fxetaria.ord)
[1] TRUE
> levels(fxetaria.ord)
[1] "Idoso" "Jovem" "Adulto"
> # altera a ordem das categorias (do mais novo ao mais velho)
> fxetaria.ord <- ordered(fxetaria, levels=c("Jovem", "Adulto", "Idoso"))
> levels(fxetaria.ord)
[1] "Jovem" "Adulto" "Idoso"
> fxetaria.ord
[1] Idoso Jovem Adulto Jovem Idoso Adulto Jovem Adulto
Levels: Jovem < Adulto < Idoso
```

Qual medida de tendência central você deve usar?

- A média **não é** robusta à presença de *outlier*
- A mediana **é** robusta à presença de *outlier*
- A moda **é** robusta à presença de *outlier*
 - A moda pode não existir ou ser múltipla para variável qualitativa ou quantitativa discreta
 - A moda sempre existe e é única para variável quantitativa contínua



Moda

- Variável quantitativa discreta
(e.g., inteiros, contagem, binária)
 - Valores igualmente mais frequentes obtidos por meio de tabela de frequências ou gráfico de pontos (dotplot)
- Variável quantitativa contínua
 - Estimativa baseada no gráfico de densidade (e.g., método de Parzen)

Moda

- Notas de disciplina de pós-graduação:
A, B, C, R, R
 - Mediana = C
 - Moda = R
- Sexo de estudantes: M, M, M, F, F
 - Moda = M

```
> conceito <- c("R","C","B","A","R")
> median(conceito)
[1] "C"
> table(conceito)
conceito
A B C R
1 1 1 2
> sexo <- c("M","M","M","F","F")
> table(sexo)
sexo
F M
2 3
>
```

Moda em Variável qualitativa (ordinal ou nominal)
Valores igualmente mais frequentes obtidos por meio de tabela de frequências ou gráfico de barras ou setores

Moda

Estaturas(cm): 165, 170, 170, 170, 175

Média = $(165+170+170+170+175)/5 = 170$

Mediana = 170

Moda = 170

ModaEstatura.R

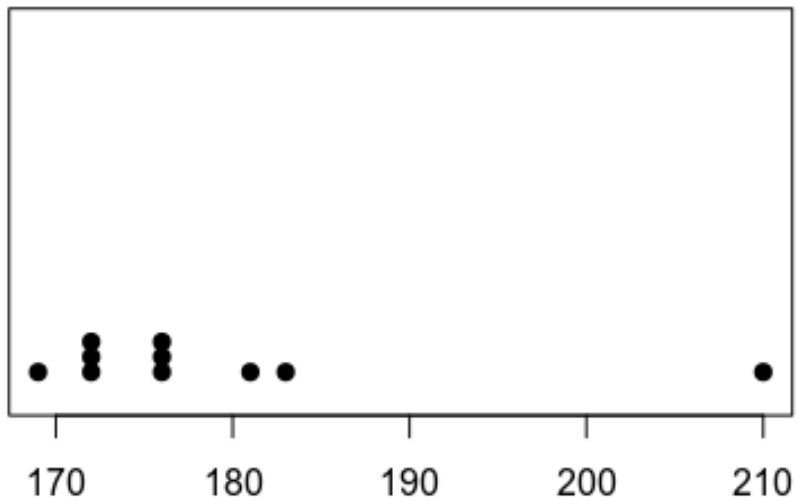
```
estatm <- c(169, 172, 176, 183, NA, 172, 181, 210, 172, 176, 176)

# moda discreta
modadiscreta <- function(x){w=table(x); w[max(w)==w]}
modad <- modadiscreta(estatm)
modas <- names(modad)
freqs <- as.vector(modad)
cat ("Moda(s) discreta(s) amostral(is): ",modas," com ",freqs[1],"
ocorrendia(s)\n")
stripchart(estatm, method="stack", offset=0.5, at=0.15, pch=19)

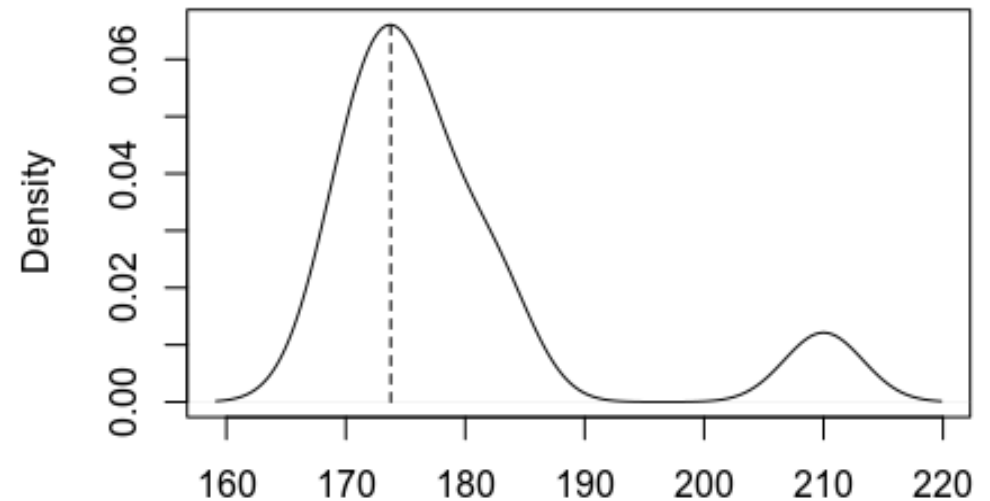
# moda por densidade
d <- density(estatm,na.rm=TRUE)
moda <- d$x[which.max(d$y)]
cat("Moda continua amostral =",moda, "\n")
# grafico
plot(d)
lines(c(moda,moda), c(min(d$y),max(d$y)), lty=2)
```

Moda

```
Moda(s) discreta(s) amostral(is): 172 176 com 3 ocorrencia(s)  
Moda continua amostral = 173.7593
```



density.default(x = estatm, na.rm = TRUE)



N = 10 Bandwidth = 3.284

Que medida de tendência central é mais apropriada para os seguintes conjuntos de dados?

a) 1 23 23 25 26 27 29 30

b) 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 3 3 4 50

c) 1 1 1 2 2 3 3 4 4 5 5 6 6 7 8

d) 1 101 104 106 108 109 111 200

Que medida de tendência central é mais apropriada para os seguintes conjuntos de dados?

a) 1 23 23 25 26 27 29 30

mediana

b) 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 3 3 4 50

moda

c) 1 1 1 2 2 3 3 4 4 5 5 6 6 7 8

média

d) 1 101 104 106 108 109 111 200

mediana

Mais exemplos de representações gráficas

- Análise de uma variável qualitativa
 - Gráfico de setores (torta ou pizza)
 - Gráfico de barras
 - Gráfico de linha
- Análise de uma variável quantitativa
 - Histograma
 - Gráfico de pontos (*dotplot*)
 - Gráfico de caixa-e-bigodes (*boxplot*)
- Análise de duas variáveis quantitativas
 - Gráfico de dispersão (*scatterplot*)

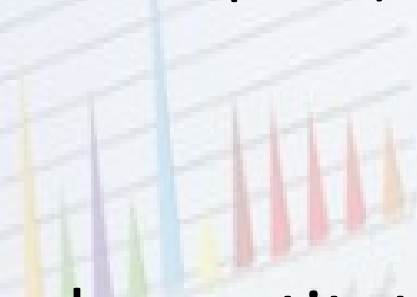


Gráfico de setores em R

GraficoSetores.R

```
faixaetaria <- sort(factor(c("Idoso", "Idoso", "Idoso", "Idoso",  
                             "Adulto", "Adulto", "Adulto", "Jovem")))  
  
rotulo <- unique(faixaetaria)  
freq <- summary(faixaetaria)  
dt_tmp <- data.frame(c(2,3,1),rotulo,freq)  
names(dt_tmp) <- c("ordem","rotulo","freq")  
dt_tmp <- dt_tmp[order(dt_tmp$ordem),]  
pie(dt_tmp$freq,  
     label = dt_tmp$rotulo,  
     main = "Grafico de setores",  
     xlab = "Faixa Etaria",  
     col = gray(seq(0.4,1.0,  
                     length = length(dt_tmp$rotulo))))
```

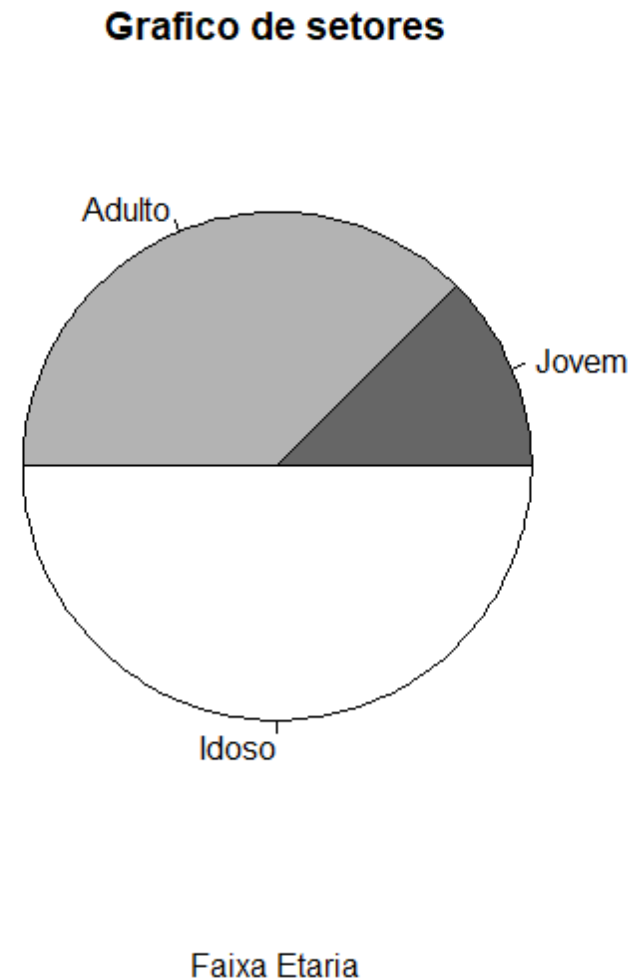


Gráfico de setores em R

Formato PDF

GraficoSetoresPDF.R

```
pdf("GraficoSetores.pdf")
faixaetaria <- sort(factor(c("Idoso", "Idoso", "Idoso", "Idoso",
                             "Adulto", "Adulto", "Adulto", "Jovem"))))
rotulo <- unique(faixaetaria)
freq <- summary(faixaetaria)
dt_tmp <- data.frame(c(2,3,1),rotulo,freq)
names(dt_tmp) <- c("ordem","rotulo","freq")
dt_tmp <- dt_tmp[order(dt_tmp$ordem),]
pie(dt_tmp$freq,
    label = dt_tmp$rotulo,
    main = "Grafico de setores",
    xlab = "Faixa Etaria",
    col = gray(seq(0.4,1.0,
    length = length(dt_tmp$rotulo))))
dev.off()
```

Gráfico de setores em R

Formato EPS

GraficoSetoresEPS.R

```
library(RcmdrMisc)
setEPS()
postscript("GraficoSetores.eps")
faixaetaria <- sort(factor(c("Idoso", "Idoso", "Idoso", "Idoso",
                             "Adulto", "Adulto", "Adulto", "Jovem"))
rotulo <- unique(faixaetaria)
freq <- summary(faixaetaria)
dt_tmp <- data.frame(c(2,3,1),rotulo,freq)
names(dt_tmp) <- c("ordem","rotulo","freq")
dt_tmp <- dt_tmp[order(dt_tmp$ordem),]
pie(dt_tmp$freq,
     label = dt_tmp$rotulo,
     main = "Grafico de setores",
     xlab = "Faixa Etaria",
     col = gray(seq(0.4,1.0,
                    length = length(dt_tmp$rotulo))))
dev.off()
```


Gráfico de barras em R

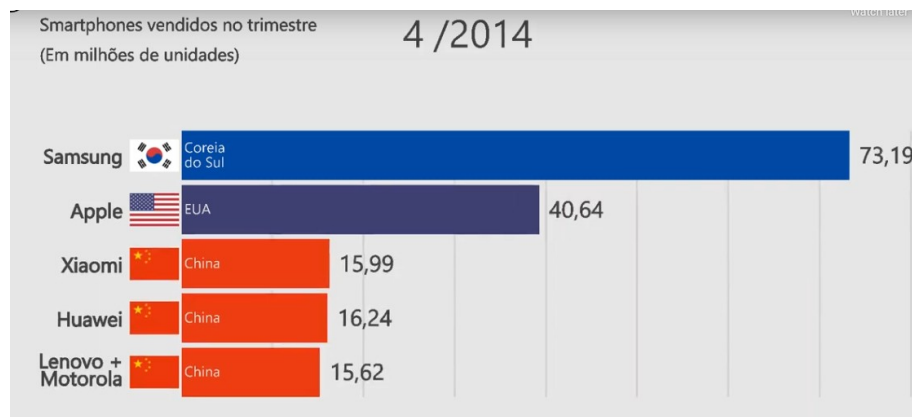
GraficoBarras.R

```
faixaetaria <- sort(factor(c("Idoso","Idoso","Idoso","Idoso",
                             "Adulto","Adulto","Adulto","Jovem"))
rotulo <- unique(faixaetaria)
freq <- summary(faixaetaria)
dt_tmp <- data.frame(c(2,3,1),rotulo,freq)
names(dt_tmp) <- c("ordem","rotulo","freq")
dt_tmp <- dt_tmp[order(dt_tmp$ordem),]
barplot(dt_tmp$freq,
        main = "Grafico de barras",
        xlab = "Faixa Etaria", ylab = "Frequencia Absoluta",
        names.arg = dt_tmp$rotulo,
        col = gray(seq(0.4,1.0,length = length(dt_tmp$rotulo))))
```



<https://www.statmethods.net/graphs/bar.html>

<https://youtu.be/IHF6A5Tri1U>



Plot Grouped Data: Box plot, Bar Plot and More

- <http://www.sthda.com/english/articles/32-r-graphics-essentials/132-plot-grouped-data-box-plot-bar-plot-and-more/#sinaplot>

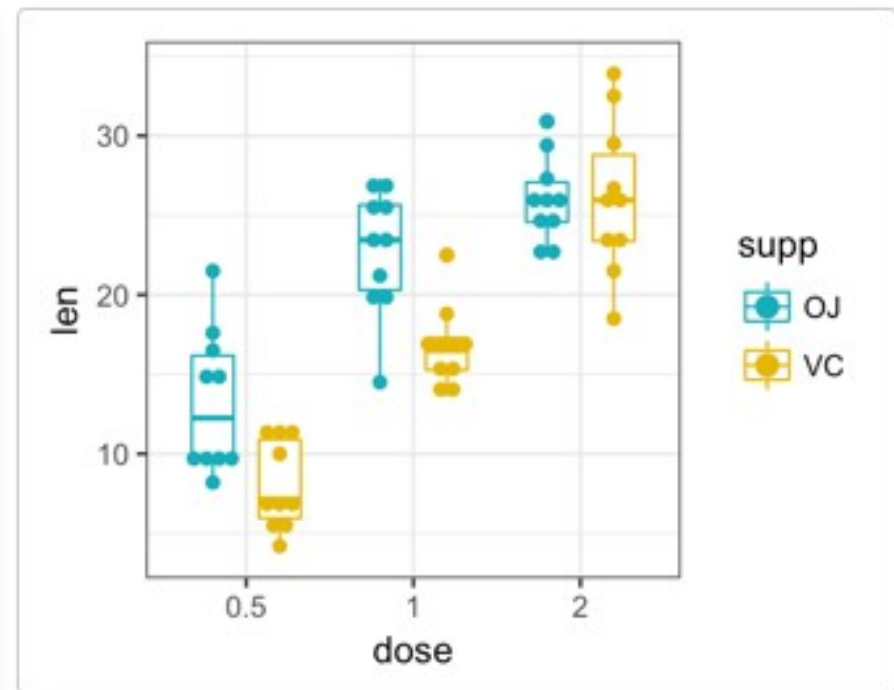
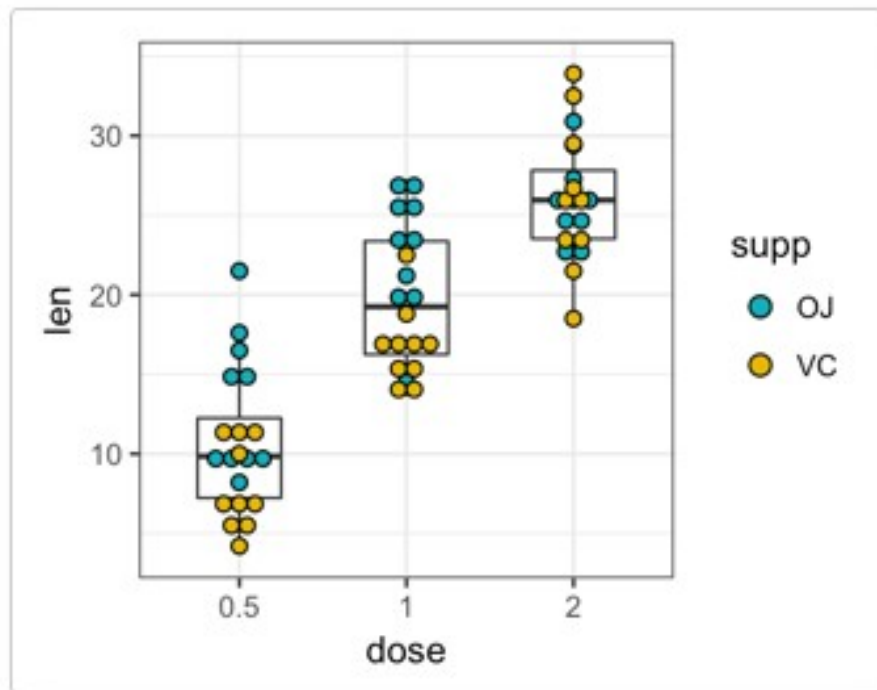
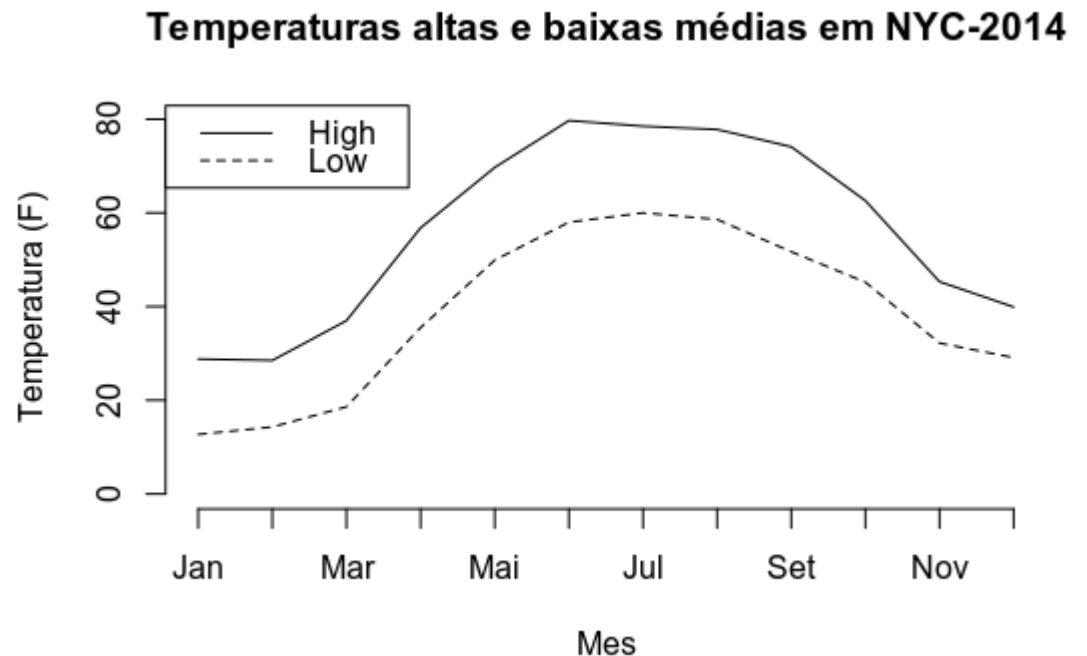


Gráfico de linhas em R

GraficoLinhas.R

```
mes <- c("Jan", "Fev", "Mar", "Abr", "Mai", "Jun",  
        "Jul", "Ago", "Set", "Out", "Nov", "Dez")  
mes <- ordered(mes, levels=mes)  
high_NYC2014 <- c(28.8, 28.5, 37.0, 56.8, 69.7, 79.7, 78.5, 77.8, 74.1, 62.6, 45.3, 39.9)  
low_NYC2014 <- c(12.7, 14.3, 18.6, 35.5, 49.9, 58.0, 60.0, 58.6, 51.7, 45.2, 32.2, 29.1)  
plot(c(1:length(mes)), high_NYC2014, type="l",  
     ylim=c(0,max(high_NYC2014)),  
     main="Temperaturas altas e baixas médias em NYC-2014",  
     xlab="Mes", ylab="Temperatura (F)", axes=FALSE)  
axis(1, at=mes, labels=mes)  
axis(2)  
lines(mes, low_NYC2014,  
      main = "Temperaturas altas e baixas médias em NYC",  
      lty=2)  
legend("topleft", c("High","Low"), lty=1:2)
```



<https://plot.ly/r/line-charts/>

Gráfico de linhas em R

GraficoLinhas2.R

```
mes <- c("Jan", "Fev", "Mar", "Abr", "Mai", "Jun",  
        "Jul", "Ago", "Set", "Out", "Nov", "Dez")  
mes_num <- 1:length(mes)  
high_NYC2014 <- c(28.8, 28.5, 37.0, 56.8, 69.7, 79.7, 78.5, 77.8, 74.1, 62.6, 45.3, 39.9)  
low_NYC2014 <- c(12.7, 14.3, 18.6, 35.5, 49.9, 58.0, 60.0, 58.6, 51.7, 45.2, 32.2, 29.1)  
dados <- data.frame(mes, high_NYC2014, low_NYC2014)  
minx <- min(mes_num, na.rm=TRUE)  
maxx <- max(mes_num, na.rm=TRUE)  
miny <- min(high_NYC2014, low_NYC2014, na.rm=TRUE)  
maxy <- max(high_NYC2014, low_NYC2014, na.rm=TRUE)  
plot(NA, NA,  
     main = "Temperaturas altas e baixas médias em NYC-2014",  
     xlim=c(minx,maxx), ylim=c(miny,maxy),  
     xaxt="n", yaxt="n",  
     xlab = "Mes",  
     ylab = "Temperatura (F)")  
# x axis  
axis(side=1, at=c(minx-1,maxx+1), labels = FALSE)  
text(x=mes_num, par("usr")[3],  
     labels = mes, srt = 30, pos = 1, xpd = TRUE)  
# y axis  
axis(side=2, las=1)  
# high temps  
high_cor <- "#333333" # cinza escuro  
high_lty <- 2  
lines(mes_num,high_NYC2014,lwd=2,lty=high_lty,col=high_cor)  
points(mes_num,high_NYC2014,pch=25,col=high_cor,bg=high_cor)  
# low temps  
low_cor <- "#666666" # cinza medio  
low_lty <- 3  
lines(mes_num,low_NYC2014,lwd=2,lty=low_lty,col=low_cor)  
points(mes_num,low_NYC2014,pch=24,col=low_cor,bg=low_cor)  
# high-low vertical lines  
hl_x <- c()  
hl_y <- c()  
for (m in 1:length(mes))  
{  
  hl_x <- c(hl_x, mes_num[m],      mes_num[m],      NA)  
  hl_y <- c(hl_y, high_NYC2014[m], low_NYC2014[m], NA)  
}  
lines(hl_x,hl_y,lty=2, lwd=1,col="#888888")  
# legenda  
legend("topleft",  
      c("Maximas", "Minimas"),  
      col=c(high_cor,low_cor),  
      pt.bg=c(high_cor,low_cor),  
      pch=c(25,24),  
      lwd=c(2,2),  
      lty=c(high_lty,low_lty),  
      box.lwd=0, bg="transparent")
```

Temperaturas altas e baixas médias em NYC-2014

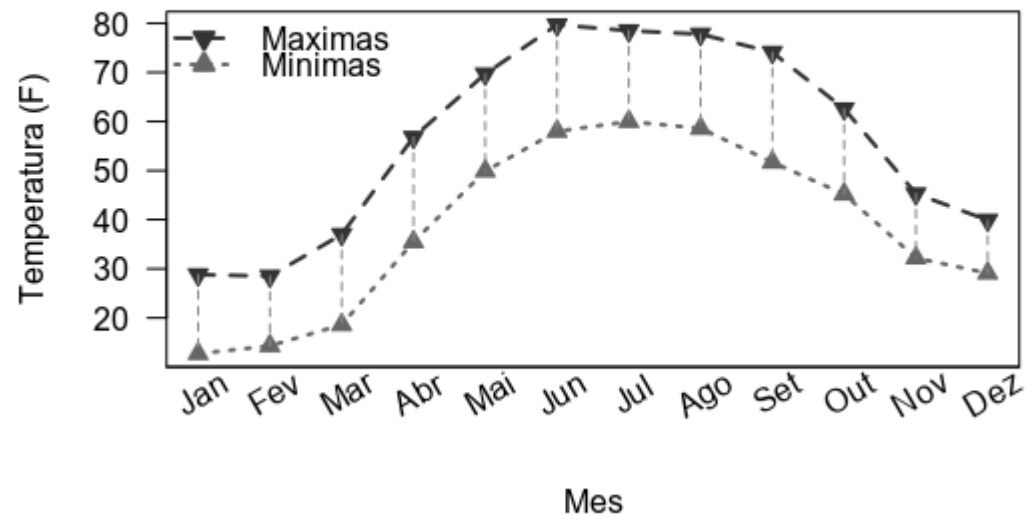


Gráfico de linhas em R

GraficoLinhas3.R

```
mes <- c("Jan", "Fev", "Mar", "Abr", "Mai", "Jun",  
        "Jul", "Ago", "Set", "Out", "Nov", "Dez")  
mes_num <- 1:length(mes)  
high_NYC2014 <- c(28.8, 28.5, 37.0, 56.8, 69.7, 79.7, 78.5, 77.8, 74.1, 62.6, 45.3, 39.9)  
low_NYC2014 <- c(12.7, 14.3, 18.6, 35.5, 49.9, 58.0, 60.0, 58.6, 51.7, 45.2, 32.2, 29.1)  
dados <- data.frame(mes, high_NYC2014, low_NYC2014)  
minx <- min(mes_num, na.rm=TRUE)  
maxx <- max(mes_num, na.rm=TRUE)  
miny <- min(high_NYC2014, low_NYC2014, na.rm=TRUE)  
maxy <- max(high_NYC2014, low_NYC2014, na.rm=TRUE)  
plot(NA, NA,  
     main = "Temperaturas altas e baixas médias em NYC",  
     xlim=c(minx,maxx), ylim=c(miny,maxy),  
     xaxt="n", yaxt="n",  
     xlab = "Mes",  
     ylab = "Temperatura (F)")  
# x axis  
axis(side=1, at=c(minx-1,maxx+1), labels = FALSE)  
text(x=mes_num, par("usr")[3],  
     labels = mes, srt = 30, pos = 1, xpd = TRUE)  
# y axis  
axis(side=2, las=1)  
# high temps  
high_cor <- "#f43328" # tijolo  
lines(mes_num,high_NYC2014,lwd=2,col=high_cor)  
points(mes_num,high_NYC2014,pch=25,col=high_cor,bg=high_cor)  
# low temps  
low_cor <- "#1965B0" # azul cobalto  
lines(mes_num,low_NYC2014,lwd=2,col=low_cor)  
points(mes_num,low_NYC2014,pch=24,col=low_cor,bg=low_cor)  
# high-low vertical lines  
hl_x <- c()  
hl_y <- c()  
for (m in 1:length(mes))  
{  
  hl_x <- c(hl_x, mes_num[m],      mes_num[m],      NA)  
  hl_y <- c(hl_y, high_NYC2014[m], low_NYC2014[m], NA)  
}  
lines(hl_x,hl_y,lty=2, lwd=1,col="#888888")  
# legenda  
legend("topleft",  
      c("Maximas", "Minimas"),  
      col=c(high_cor,low_cor),  
      pt.bg=c(high_cor,low_cor),  
      pch=c(25,24),  
      lwd=c(2,2),  
      lty=c(high_lty,low_lty),  
      box.lwd=0, bg="transparent")
```

Temperaturas altas e baixas médias em NYC-2014

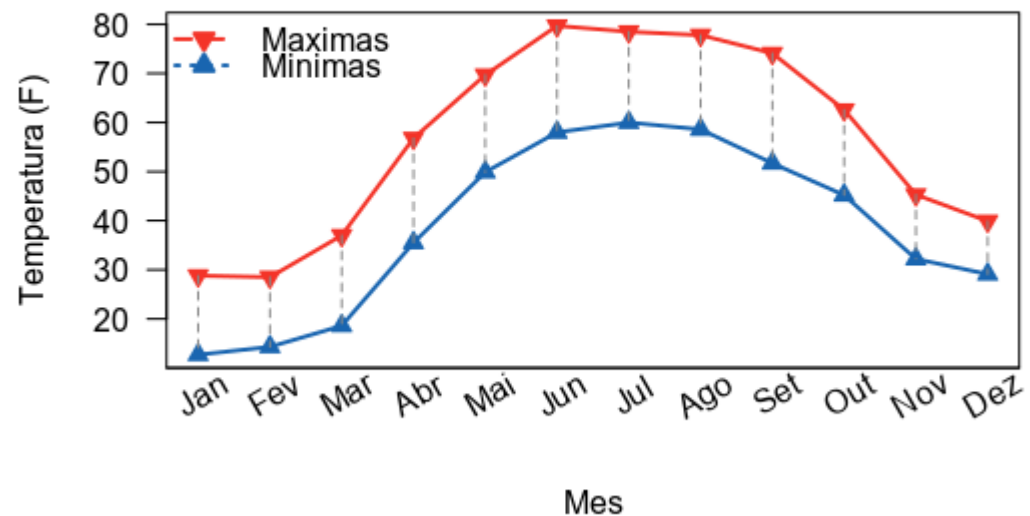
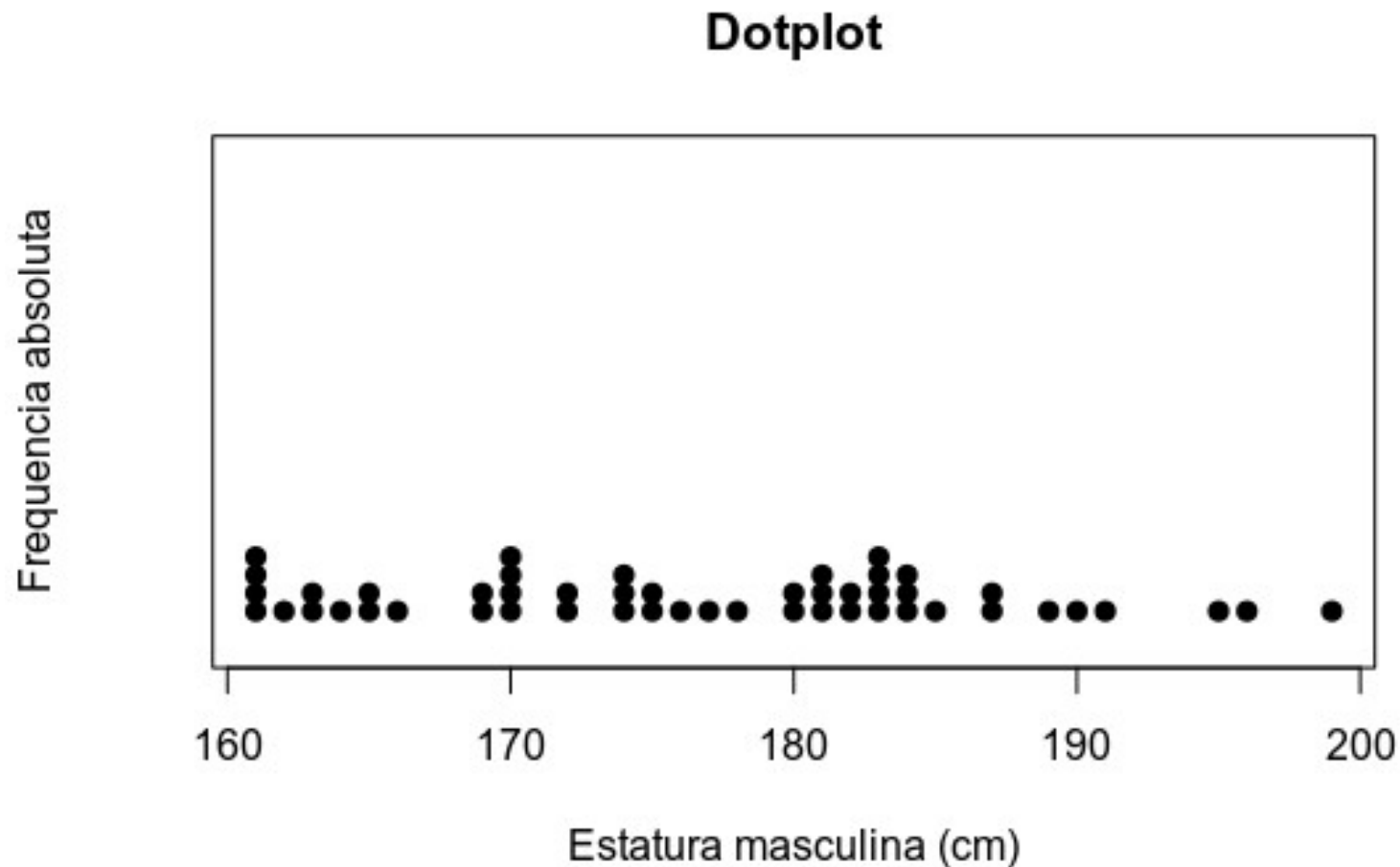


Gráfico de pontos (dotplot) em R

GraficoPontos.R

```
estatura <- as.integer(rnorm(n=50, mean=177, sd=10))  
stripchart(estatura, method="stack", offset=0.5, at=0.15, pch=19,  
           xlab = "Estatura masculina (cm)", ylab = "Frequencia absoluta",  
           main="Dotplot")
```



Histograma em R

GraficoHistograma.R

```
estatura <- rnorm(n=50, mean=177, sd=10)
hist(estatura,
      xlab = "Estatura masculina (cm)",
      ylab = "Frequencia absoluta",
      main = "Histograma")
```

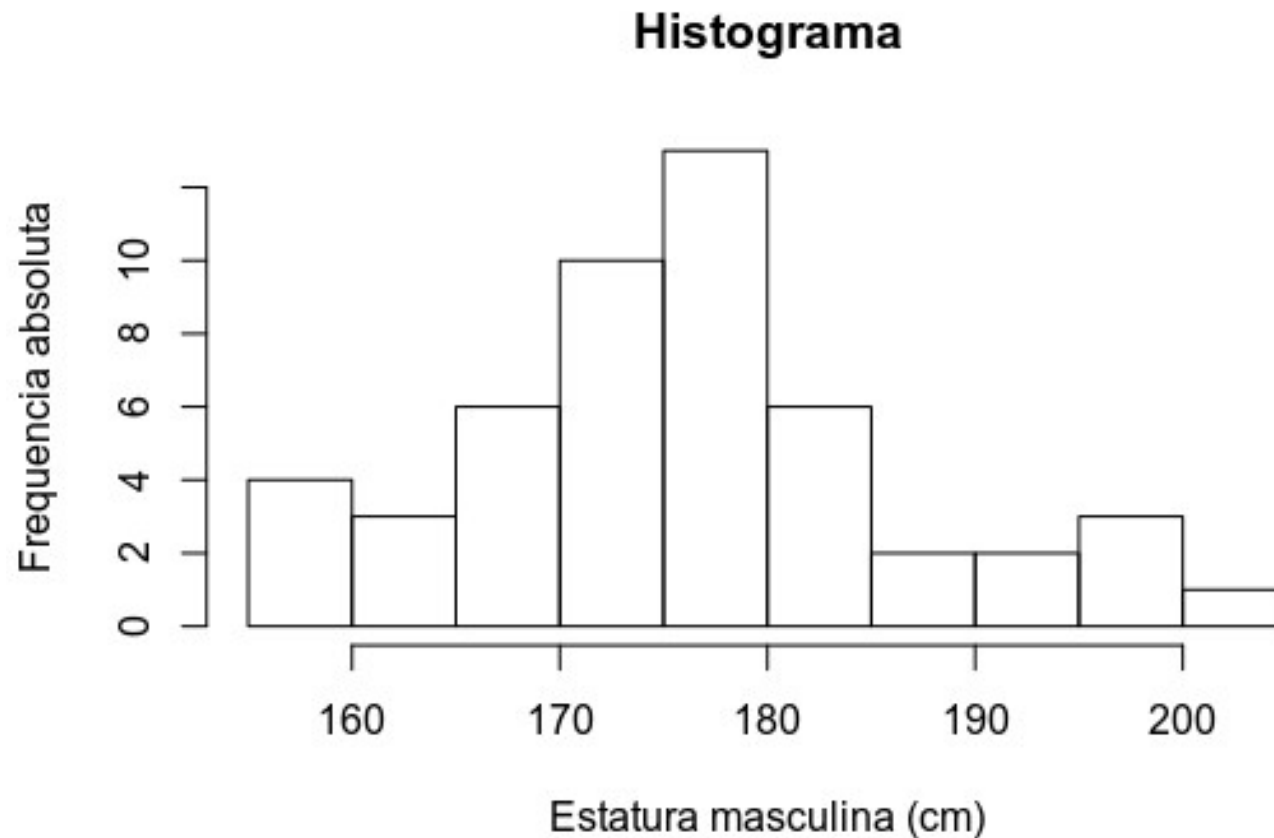


Gráfico de dispersão em R

GraficoDispersao.R

```
library(readxl)
Dados <- read_excel(file.path("dados", "Adm2008_v2.xlsx"))
N <- min(sum(!is.na(Dados$Estatura[Dados$Genero=="Feminino"])),
        Dados$MCT[Dados$Genero=="Feminino"])
plot(Dados$Estatura[Dados$Genero=="Feminino"],
     Dados$MCT[Dados$Genero=="Feminino"],
     main=paste("Estudantes femininas de Administração Noturno
FEA-USP 2008", "\nN =", N),
     xlab="Estatura (cm)",
     ylab="Massa Corporal Total (kg)")
```

Estudantes femininas de Administração Noturno
FEA-USP 2008
N = 38

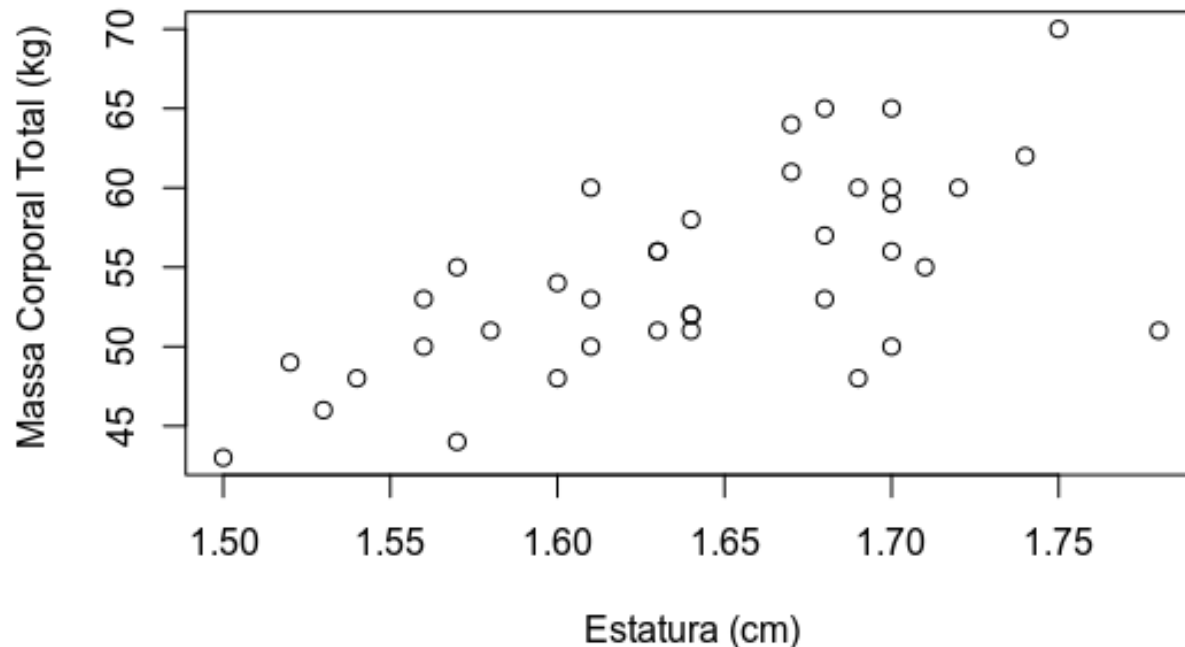


Gráfico de caixas bidimensional em R

GraficoBagplot.R

```
library(readxl)
library("aplpack")
Dados <- read_excel(file.path("dados", "Adm2008_v2.xlsx"))
N <- min(sum(!is.na(Dados$Estatura[Dados$Genero=="Feminino"])),
        Dados$MCT[Dados$Genero=="Feminino"])
bagplot(Dados$Estatura[Dados$Genero=="Feminino"],
        Dados$MCT[Dados$Genero=="Feminino"],
        main=paste("Estudantes femininas de Administração Noturno
FEA-USP 2008", "\nN =", N),
        xlab="Estatura (m)",
        ylab="Massa Corporal Total (kg)",
        na.rm = TRUE)
```

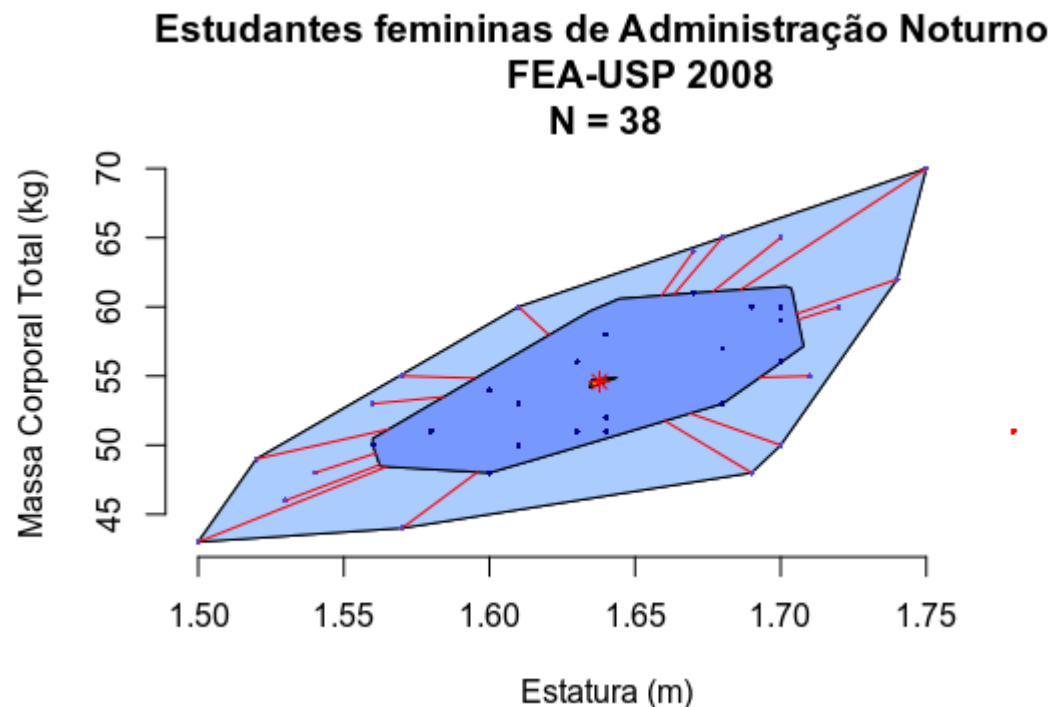


Gráfico de densidade em R

GraficoDensidade.R

```
estatura.masc <- rnorm(n=50, mean=177, sd=10)
plot(density(estatura.masc),
     main="Grafico de densidade",
     xlab="Estatura masculina (cm)", ylab="Densidade")
rug(estatura.masc)
```

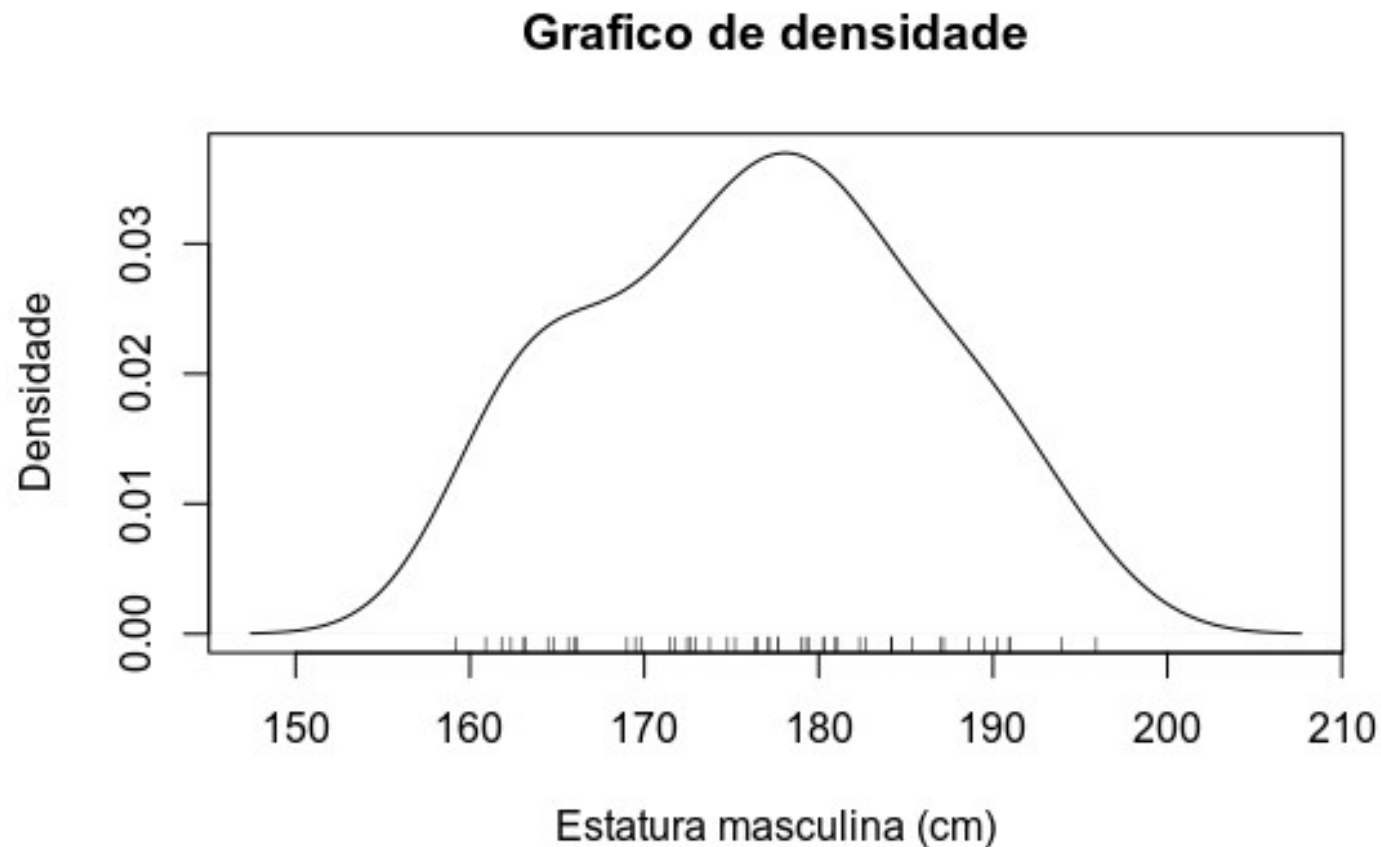


Gráfico de densidade: comparação de grupos

GraficoDensidade_grupos.R

	mean	sd	IQR	0%	25%	50%	75%	100%	Estatura:n
Feminino	1.641316	0.06798931	0.0975	1.50	1.60	1.64	1.6975	1.78	38
Masculino	1.763529	0.08076691	0.1000	1.56	1.72	1.76	1.8200	1.93	51

```
library(readxl)
library(car)
Dados <- read_excel(file.path("dados", "Adm2008_v2.xlsx"))
densityPlot(Estatura~factor(Genero), data=Dados, bw=bw.SJ,
            adjust=1, kernel=dnorm, method="adaptive")
```

	Nome	Genero	Estatura
1	Beatriz	Feminino	1.61
2	Camila	Feminino	1.56
3	Christiane	Feminino	1.72
4	Debora	Feminino	1.57
5	Denise	Feminino	1.68
6	Elaine	Feminino	1.69
7	Elisa	Feminino	1.70
8	Ermínia	Feminino	1.64
9	Fabiana	Feminino	1.70
10	Flávia	Feminino	1.57

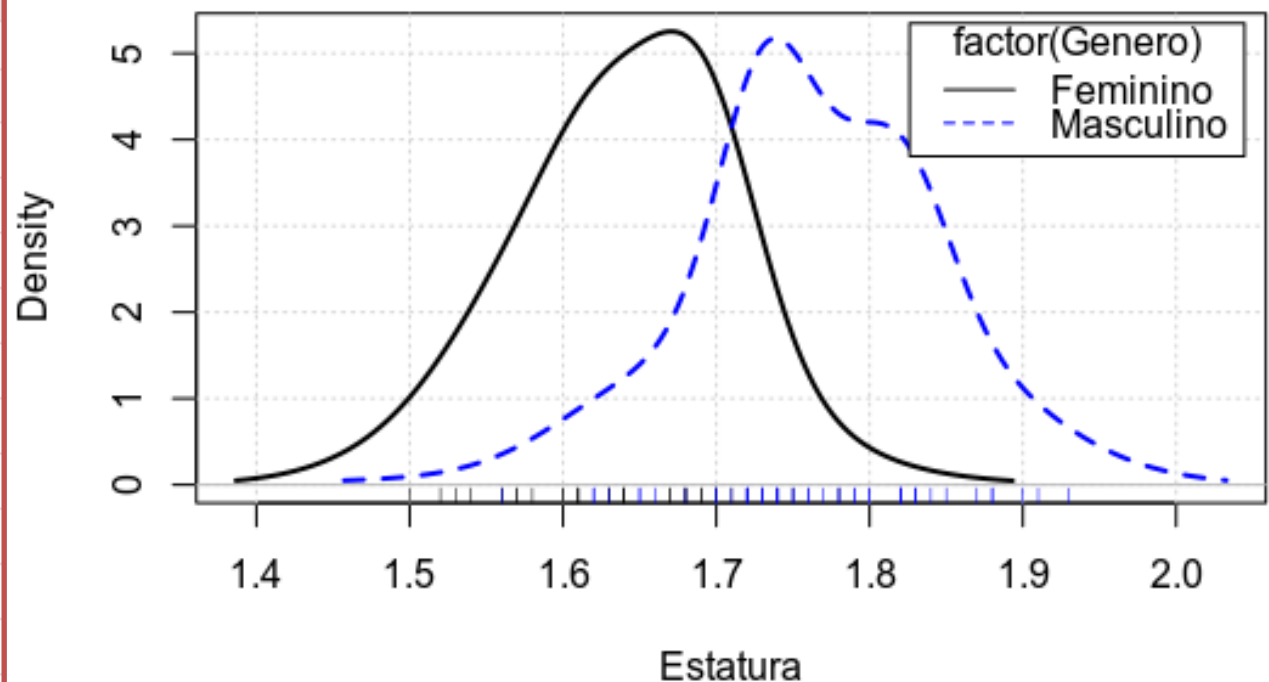


Gráfico de função em R

GraficoFuncao.R

```
D <- seq(0,100,1)
v <- 1/(1 + D)
plot(D, v, type="l", main="v = 1/(1 + D)")
```

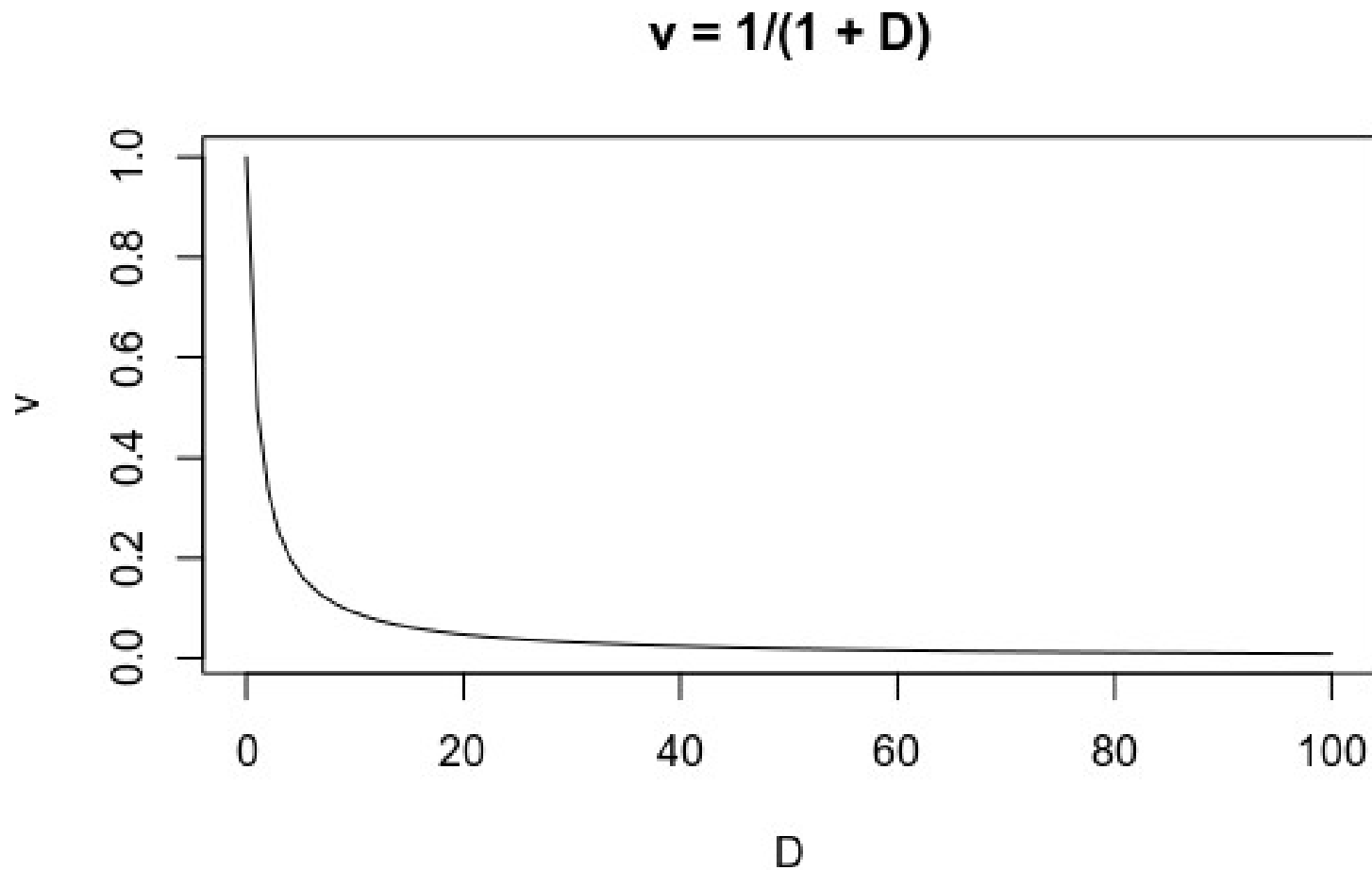


Gráfico de intervalo de confiança de 95% de média populacional em R

GraficoIC95.R

```
library(RcmdrMisc)
estatura <- c(176, 183, 173, 191, 157, 152, 174, 166)
genero <- factor(c("M", "M", "M", "M", "F", "F", "F", "F"))
tabela <- data.frame(estatura, genero)
with(tabela, plotMeans(estatura, genero, error.bars="conf.int", level=0.95,
                        xlab="Genero", ylab="Estatura", main="IC95%",
                        connect=FALSE))
```

```
> tabela
  estatura genero
1     176      M
2     183      M
3     173      M
4     191      M
5     157      F
6     152      F
7     174      F
8     166      F
```

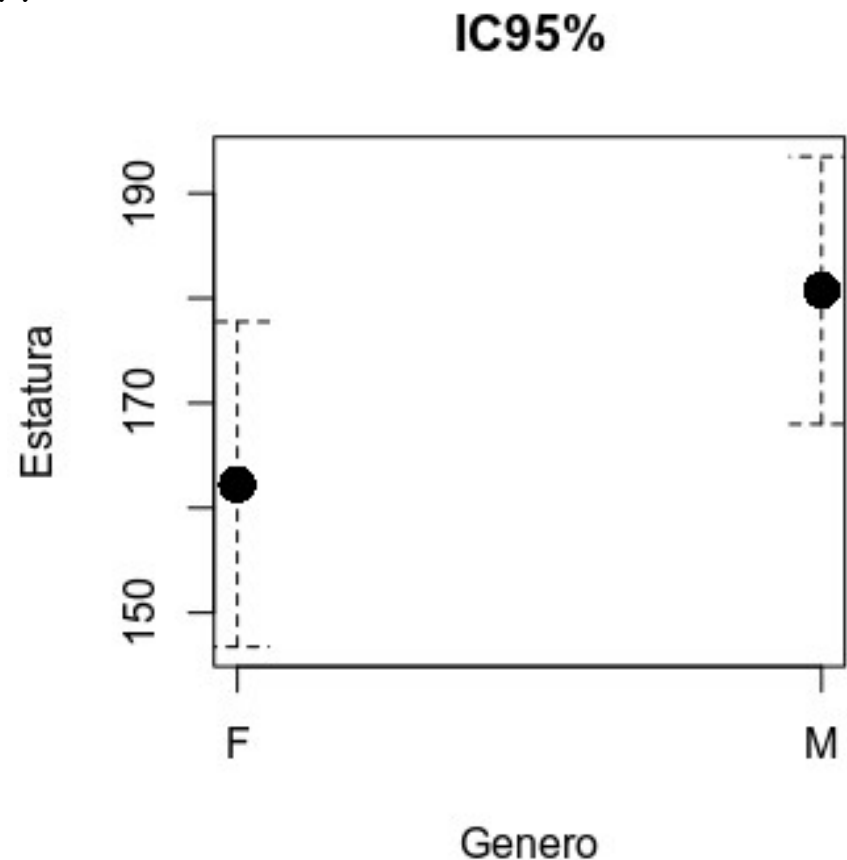


Gráfico de caixa (boxplot) em R

GraficoBoxplot.R

```
estatura <- c(176, 183, 173, 191, 157, 152, 174, 266)
genero <- factor(c("M", "M", "M", "M", "F", "F", "F", "F"))
tabela <- data.frame(estatura, genero)
boxplot(estatura,
        main=paste("Boxplot\nN =", sum(!is.na(estatura))),
        ylab="Estatura (cm)",
        data=tabela)
```

```
> tabela
  estatura genero
1     176      M
2     183      M
3     173      M
4     191      M
5     157      F
6     152      F
7     174      F
8     266      F
```

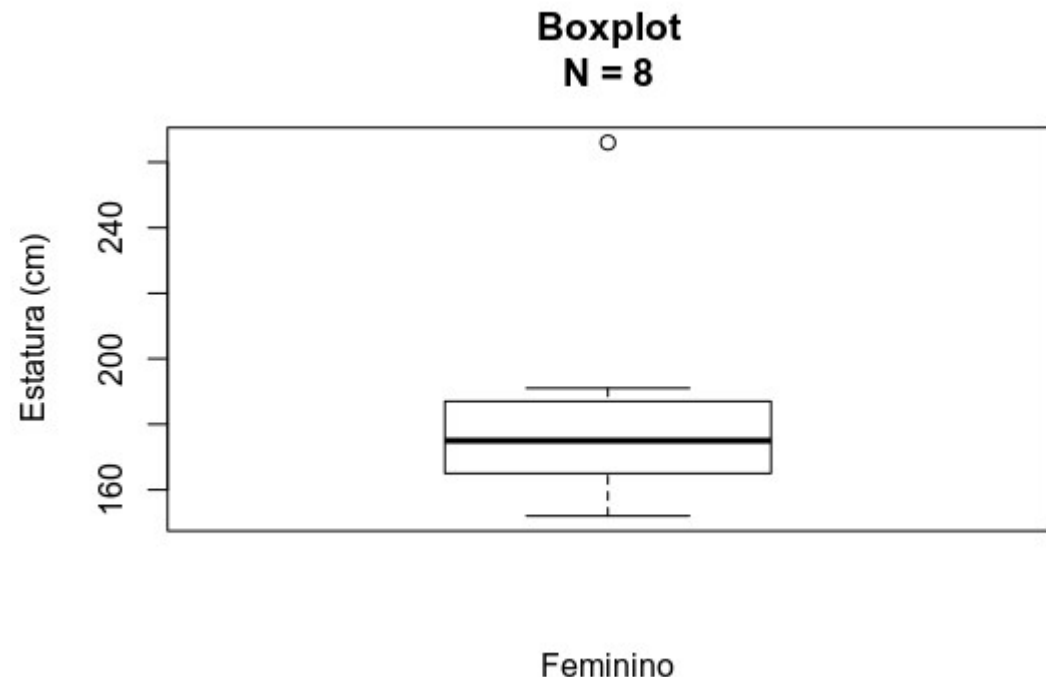


Gráfico de caixa (boxplot) em R

GraficoBoxplotF.R

```
estatura <- c(176, 183, 173, 191, 157, 152, 174, 166)
genero <- factor(c("M", "M", "M", "M", "F", "F", "F", "F"))
tabela <- data.frame(estatura, genero)
boxplot(estatura[genero=="F"],
        main=paste("Boxplot\nN =", sum(is.na(estatura[genero=="F"]))),
        ylab="Estatura (cm)",
        xlab="Feminino",
        data=tabela)
```

```
> tabela
  estatura genero
1     176      M
2     183      M
3     173      M
4     191      M
5     157      F
6     152      F
7     174      F
8     266      F
```

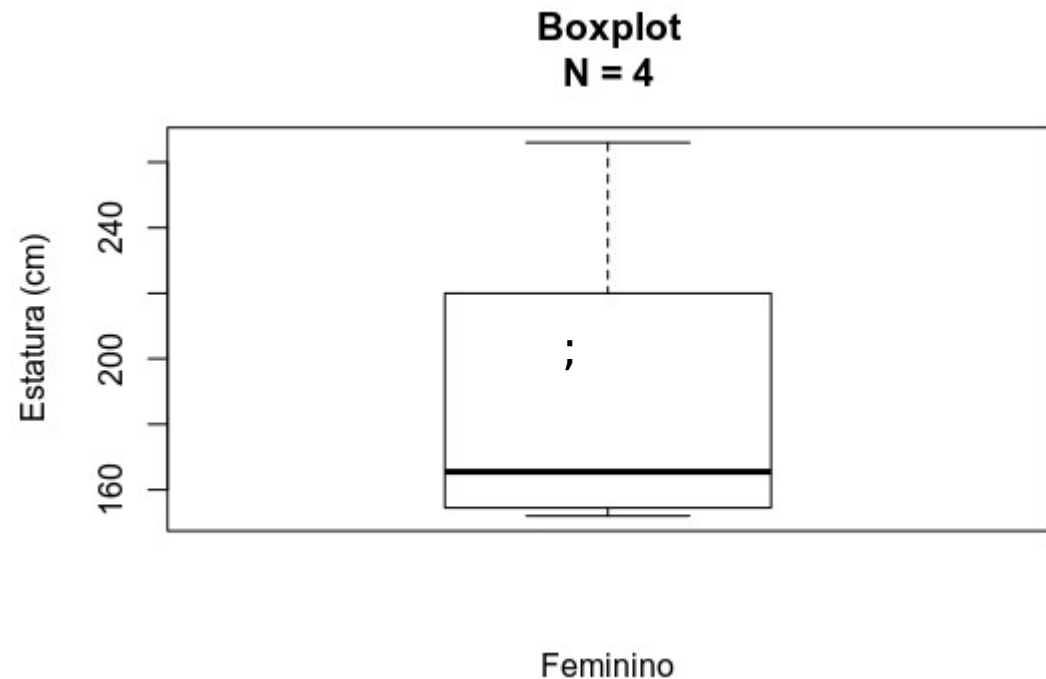
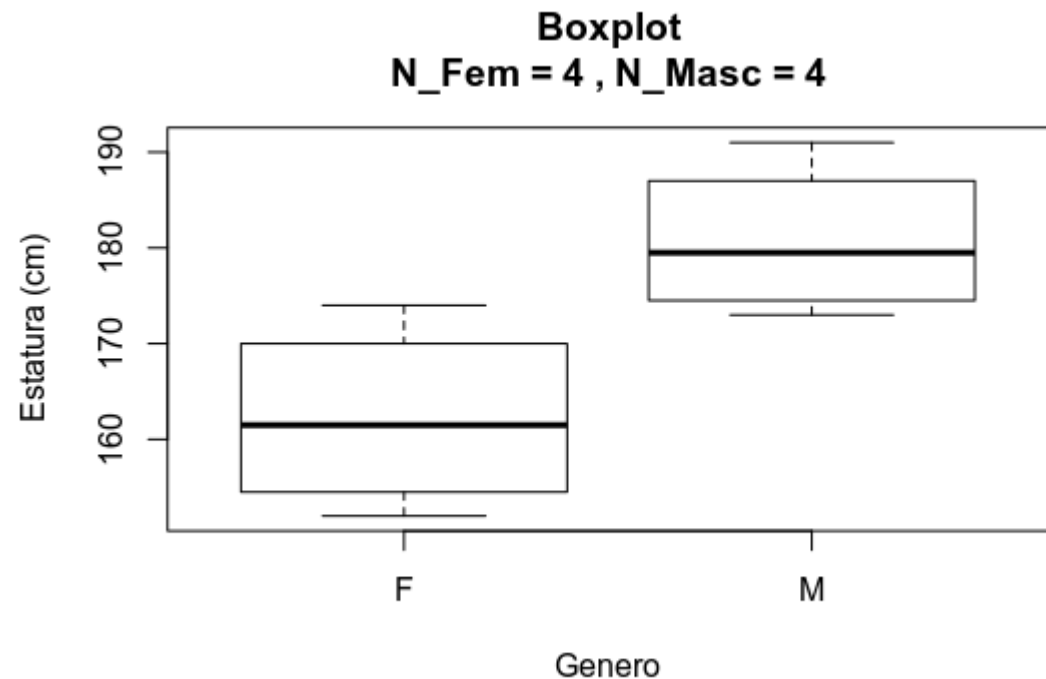


Gráfico de caixa (boxplot) em R

GraficoBoxplot_grupos.R

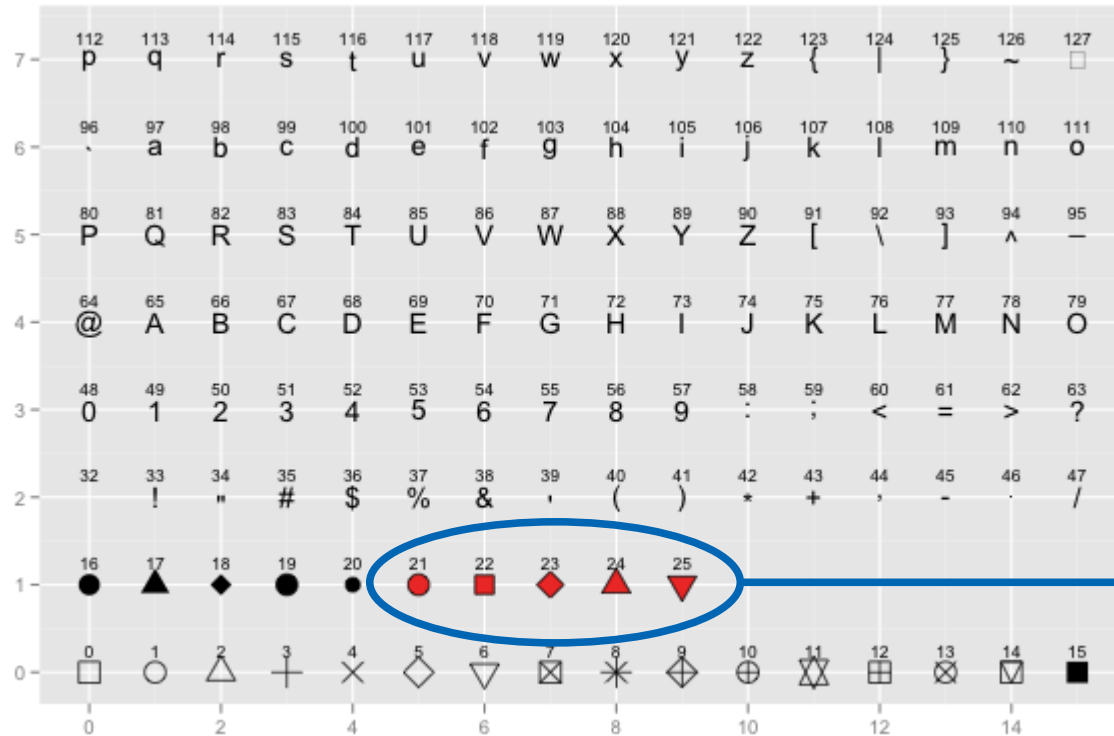
```
estatura <- c(176, 183, 173, 191, 157, 152, 174, 166)
genero <- factor(c("M", "M", "M", "M", "F", "F", "F", "F"))
tabela <- data.frame(estatura, genero)
boxplot(estatura ~ genero,
        main=paste("Boxplot\nN_Fem =", sum(!is.na(estatura[genero=="F"])),
                  ", N_Masc =", sum(!is.na(estatura[genero=="M"]))),
        ylab="Estatura (cm)",
        xlab="Genero",
        data=tabela)
```

```
> tabela
  estatura genero
1     176      M
2     183      M
3     173      M
4     191      M
5     157      F
6     152      F
7     174      F
8     166      F
```



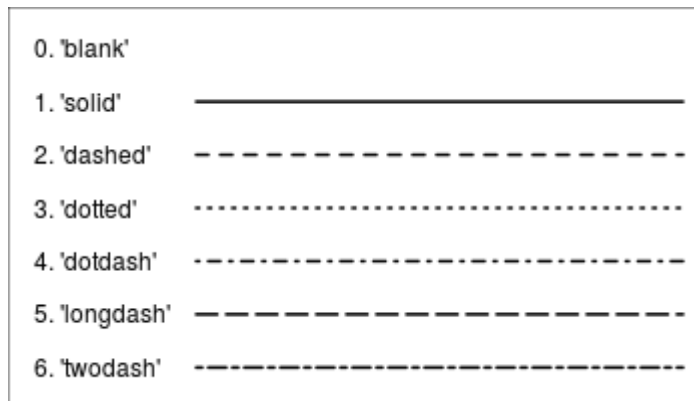
Parâmetros gráficos

pch =



col = "#RRGGBBTT"
bg = "#RRGGBBTT"
friendlycolor()

lty =



lwd =



Parâmetros gráficos

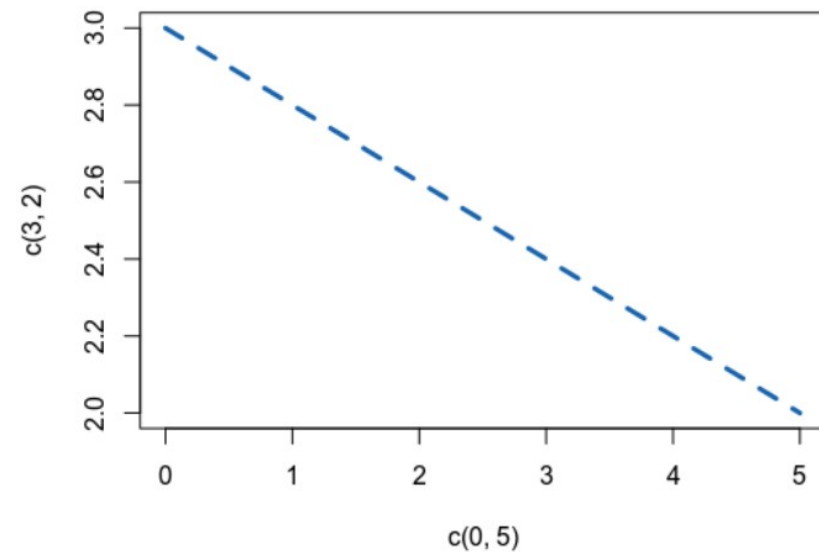
```
friendlycolor <- function(idx)
{
  fcor <- c()

  fcor <- c(fcor, "#882E72") # violeta escuro #1
  fcor <- c(fcor, "#994F88") # violeta 5 #2
  fcor <- c(fcor, "#AA6F9E") # violeta 4 #3
  fcor <- c(fcor, "#BA8DB4") # violeta 3 #4
  fcor <- c(fcor, "#CAACCB") # violeta 2 #5
  fcor <- c(fcor, "#D9CCE3") # violeta claro #6
  fcor <- c(fcor, "#0d5092") # azul naval #7
  fcor <- c(fcor, "#1965B0") # azul cobalto #8
  ...
  fcor <- c(fcor, "#507052") # verde musgo #13
  ...
  fcor <- c(fcor, "#ac4d12") # ocre #19
  ...
  fcor <- c(fcor, "#42150A") # marrom escuro #25
  ...
  fcor <- c(fcor, "#111111") # cinza 1 #31
  fcor <- c(fcor, "#222222") # cinza 2 #32
  ...
  fcor <- c(fcor, "#dddddd") # cinza 13 #44
  fcor <- c(fcor, "#eeeeee") # cinza 14
  fcor <- c(fcor, "#000000") # preto
  fcor <- c(fcor, "#ffffff") # branco
  return (fcor[idx])
}
```

col = "#RRGGBBTT"
bg = "#RRGGBBTT"

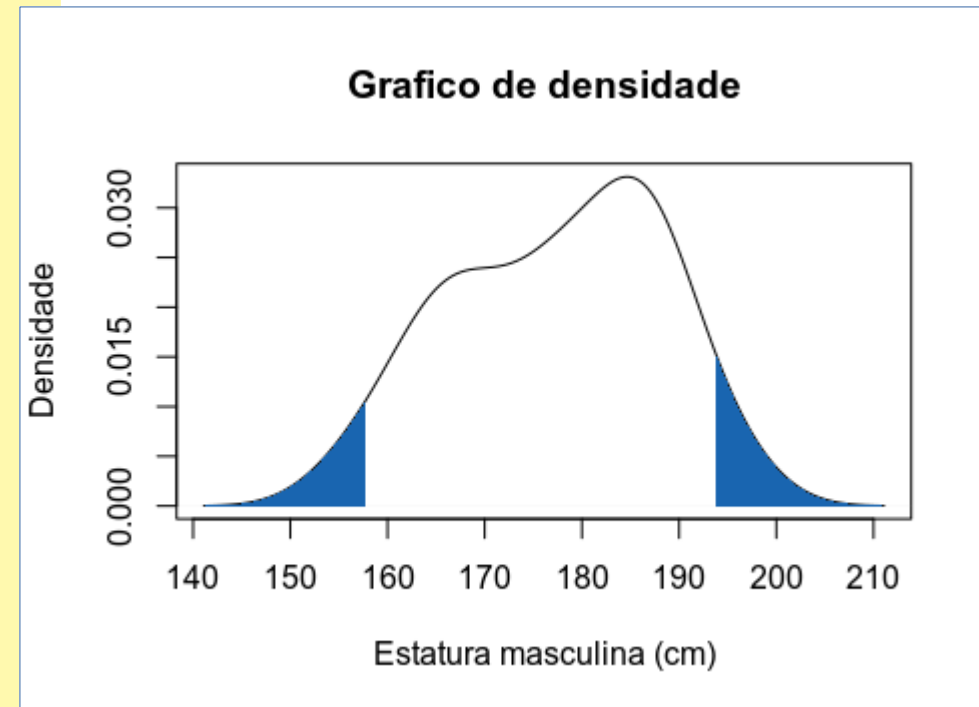
TT de 00 a FF

```
> source("friendlycolor.R")
> plot(c(0,5), c(3,2), type = "l",
      lty=2, lwd=3, col=friendlycolor(8))
```



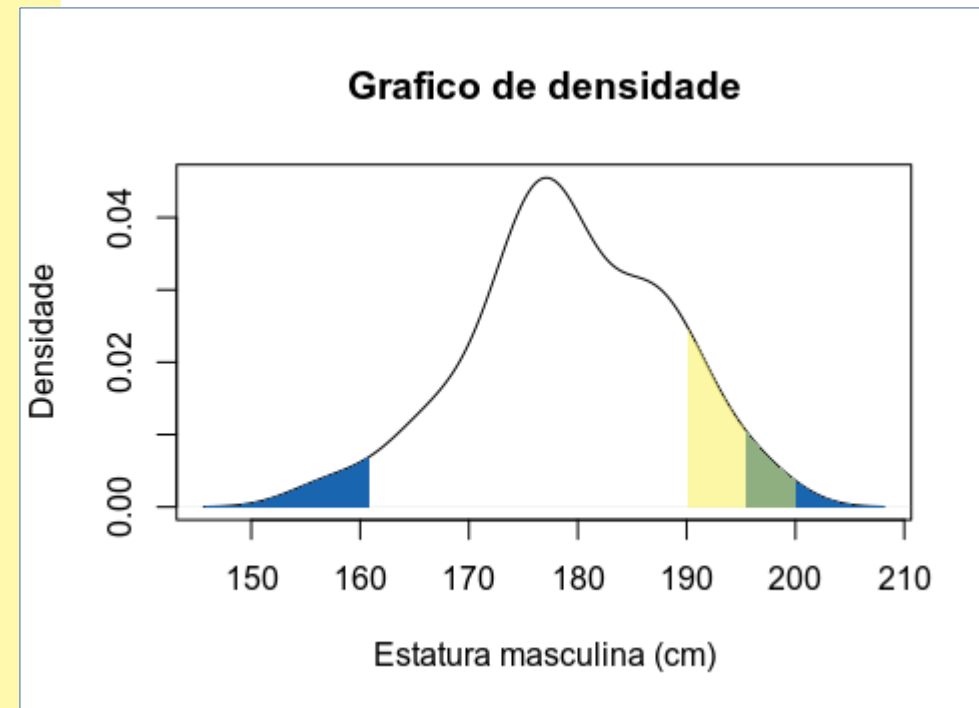
main, xlab, ylab, polygon() GraficoDensidadePolygon.R

```
source("friendlycolor.R")
estatura.masc <- rnorm(n=50, mean=177, sd=10)
estatdens <- density(estatura.masc)
plot(estatdens,
     main="Grafico de densidade",
     xlab="Estatura masculina (cm)", ylab="Densidade")
# caudas
limites <- quantile(estatura.masc, probs = c(0.025,0.975))
polx <- estatdens$x[estatdens$x<=limites[1]]
polx <- c(min(polx), polx, max(polx))
poly <- estatdens$y[estatdens$x<=limites[1]]
poly<- c(0, poly, 0)
polygon(polx,poly,border=NA,
       col=friendlycolor(8)
       )
polx <- estatdens$x[estatdens$x>=limites[2]]
polx <- c(min(polx), polx, max(polx))
poly <- estatdens$y[estatdens$x>=limites[2]]
poly<- c(0, poly, 0)
polygon(polx,poly,border=NA,
       col=friendlycolor(8)
       )
```



main, xlab, ylab, polygon() GraficoDensidadePolygon.R

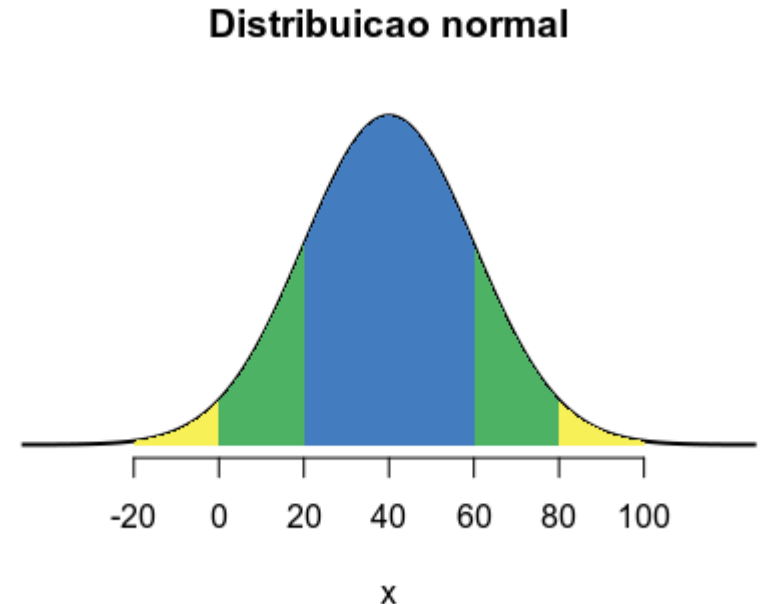
```
source("friendlycolor.R")
estatura.masc <- rnorm(n=50, mean=177, sd=10)
estatdens <- density(estatura.masc)
plot(estatdens,
     main="Grafico de densidade",
     xlab="Estatura masculina (cm)", ylab="Densidade")
# caudas
limites <- quantile(estatura.masc, probs = c(0.025,0.975))
polx <- estatdens$x[estatdens$x<=limites[1]]
polx <- c(min(polx), polx, max(polx))
poly <- estatdens$y[estatdens$x<=limites[1]]
poly<- c(0, poly, 0)
polygon(polx,poly,border=NA,
        col=friendlycolor(8)
        )
polx <- estatdens$x[estatdens$x>=limites[2]]
polx <- c(min(polx), polx, max(polx))
poly <- estatdens$y[estatdens$x>=limites[2]]
poly<- c(0, poly, 0)
polygon(polx,poly,border=NA,
        col=friendlycolor(8)
        )
# transparencia (amarelo)
polx <- estatdens$x[estatdens$x>=190 & estatdens$x<=200]
polx <- c(min(polx), polx, max(polx))
poly <- estatdens$y[estatdens$x>=190 & estatdens$x<=200]
poly<- c(0, poly, 0)
polygon(polx,poly,border=NA,
        col=paste(friendlycolor(24) ,"88", sep="")
        )
```



Distribuição normal (gaussiana)

GraficoNormal.R

```
# GraficoNormal.R
source ("friendlycolor.R")
# normal
media <- 40
desvpad <- 20
x <- seq(from=media-5*desvpad, to=media+5*desvpad, by=0.01)
y <- dnorm(x, mean=media, sd=desvpad)
xy <- data.frame(x,y)
plot(x,y,
      main="Distribuicao normal", xlab="x", ylab=NA,
      xlim=c(media-4*desvpad,media+4*desvpad),
      axes=FALSE,
      type="l",lwd=2
    )
desvios <- c(-3,-2,-1, 0, 1, 2, 3)
axis(side = 1, at = media+desvios*desvpad)
# caudas
prob.cauda <- pnorm(media+desvios*desvpad, mean=media,
sd=desvpad)
x.cauda <- qnorm(prob.cauda, mean=media, sd=desvpad)
# amarelo, verde, azul, NA, azul, verde, amarelo
cor <- c(24, 15, 9, NA, 9, 15, 24)
for (d in 1:3) # 3, 2 e 1 desvios-padrao
{
  polx <- xy$x[xy$x>=x.cauda[d] & xy$x<=x.cauda[8-d]]
  poly <- xy$y[xy$x>=x.cauda[d] & xy$x<=x.cauda[8-d]]
  polx <- c(min(polx), polx, max(polx))
  poly<- c(0, poly, 0)
  polygon(polx,poly,border=NA,col=friendlycolor(cor[d]))
}
```



média = 40
desvio-padrão = 20