R e RStudio

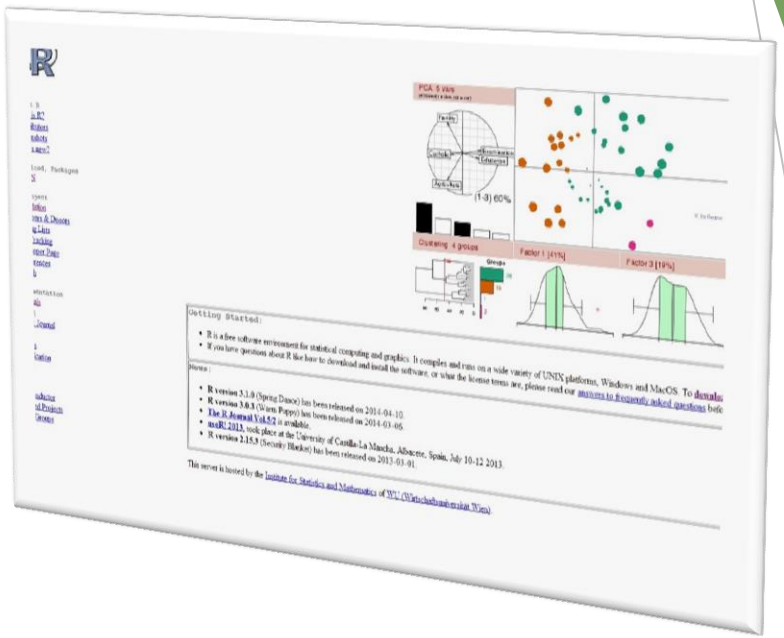
Prof. Raphael de Souza

R

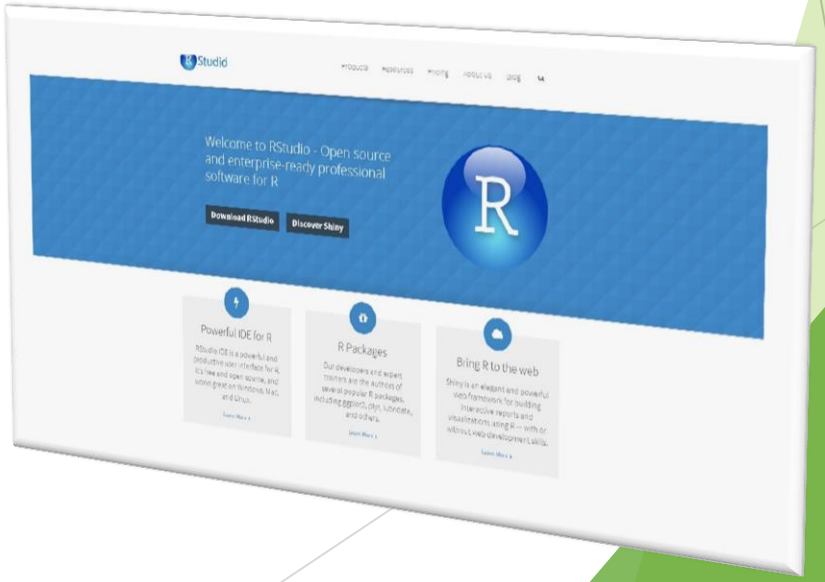
- O que é o R?
- é um conjunto integrado de funcionalidades para manipulação de dados, cálculo e exibição gráfica

Vamos Começar!

R - http://www.r-project.org/



Rstudio - http://www.rstudio.com/



RStudio

- O que é RStudio?

- é um conjunto de ferramentas integradas projetadas para ajudá-lo a ser mais produtivo com R

RStudio


RStudio
File Edit Code View Plots Session Build Debug Tools Help
Introduction to R.R
Source on Save
Run
Source
# Introduction to R
# Copyright 2013 by Ani Katchova
# Set working directory to where csv file is located
setwd("C:/users/ricardo/onedrive/cursos/r/")
#Verificar qual o working directory
getwd()
#Listando Arquivos no diretório de t
list.files()
#ajuda por comando
?getwd
#Ajuda por pesquisa
help.search("dir")
# Read the data
mydata<- read.csv("intro_auto.csv")
attach(mydata)
# List the variables
names(mydata)
# Show first lines of data
head(mydata)
mydata[1:10,]
(Top Level)
R Script
Janela de Codificação
Environment
History
Import Dataset
Clear
List
Global Environment
Histórico
Files
Plots
Packages
Help
Viewer
Zoom
Export
Clear All
Plots (gráficos)
Console ~/
R version 3.1.0 (2014-04-10) -- "Spring Dance"
Copyright (C) 2014 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)
R is free software and comes with ABSOLUTEL
You are welcome to redistribute it under ce
Type 'license()' or 'licence()' for distrib
R is a collaborative project with many cont
Type 'contributors()' for more information
'citation()' on how to cite R or R packages
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.
>
Console

R: Pacotes (Packages)

- O R permite aos usuários a importação de pacotes (packages) para expandir as funcionalidades da ferramenta.
- Antes de carregar um pacote, o mesmo deve ser baixado do servidor CRAN:
  - http://cran.r-project.org/web/packages/
- O pacotes devem ser carregados todas as vezes que o R for iniciado (com exceção da instalação).

R: Características

- # - Significa comentário
- Case-Sensitive - Ou seja, letras maiúsculas ou minúsculas FAZEM diferença

R: Variáveis

- A=4 % numeric
- Nome='jose' % caracter ou char
- T=30.5 % numeric
- Logico=TRUE % booleano ou logical

R: Estrutura de Dados

- Vetor
- Matrizes
- Listas: Vetor com elementos distintos
- Data Frames: matriz com elementos distintos

R: Estrutura de Dados

```
vetor = 1:10
matrix = matrix(c(1,2,'c'),ncol=3)
lista = list(1,'c')
dataframe = data.frame(c('a','b'),1:2)
```

R: Estrutura de Dados

- Nome de Colunas e Linhas
  - names(nome_da_variável)
    - Função para visualizar o nome das colunas de um data frame ou do list (Header ou Cabeçalho)
    - Exemplo: `names(meus_dados)`
  - rownames(nome_da_variavel)
    - Função para visualizar o nome das linhas de um data frame
    - Exemplo: `rownames(meus_dados)`

R: Importação de Dados

- Para importar arquivos do tipo *.csv, utilizamos a função read.csv()
  - Exemplo 01: Importar todos os dados de um arquivo
    - `meus_dados <- read.csv("nome_do_arquivo")`
  - Exemplo 02: Importar todos os dados de um arquivo, menos a primeira linha
    - `meus_dados <- read.csv("nome_do_arquivo",skip = 1)`
  - Exemplo 03: Mesmo que o exemplo 02, mas importando o cabeçalho
    - `meus_dados <- read.csv("nome_do_arquivo",skip = 1,header=TRUE)`
  - Exemplo 04: Mesmo que o exemplo 03 e convertendo os valores = 0 para NA
    - `meus_dados <- read.csv("nome_do_arquivo", skip=1,header=TRUE,na.strings="0")`

R: Importação de Dados

- Para acessar uma determinada coluna em um conjunto de dados utilize a sintaxe:
  - `<conjunto_de_dados>$<nome_da_coluna>`, onde:
    - `<conjunto_de_dados>` é o nome da variável do conjunto de dados
    - `<nome_da_coluna>` é o nome da coluna a ser acessada.
  - Exemplo: `meusdados$marca`

R: Operações

- Cálculos Matemáticos

Aritméticos	
+	Adição
-	Subtração
*	Multiplicação
/	Divisão
^	Exponenciação
%/%	Divisão Inteira: 5 %/% 2 = 2
%%	Resto da Divisão 5 %% 2 = 1

R: Operações

Funções	
abs()	Valor Absoluto
exp(x)	Antilog, (e^x)
log10(x)	Log base 10
log(x)	Log base e de x (ln)
log(x,n)	Log para base n de x
max()	Valor Máximo
mean()	Média
median()	Mediana
min()	Valor Mínimo
range()	Retorna min() & max()
sqrt()	Raiz Quadrada
sum()	Soma

R: Operações

Funções de Vetores	
cumsum(x)	A soma de todos os elementos até o ponto x
cumprod(x)	O produto de todos os elementos até o ponto x
cummax(x)	Números que são os máximos cumulativos dos valores x até que ponto não decrescente
cummin(x)	Números que são os mínimos cumulativos dos valores x até que ponto não crescente

R: Operações

Transformações trigonométricas	
cos() tan() sin()	Transformações em radianos
acos() atan() asin()	Transformações inversas
acosh() atanh() asinh()	Transformações hiperbólicas inversas

Exercício

- Realize a leitura dos dados do arquivo ‘dados.csv’ e faça a média de cada coluna

R: Estrutura Condicional

- As condições podem ter os símbolos:
  - >
  - <
  - >=
  - <=
  - ==
  - !=

R: Estrutura Condicional

- As condições podem ter conectivos:
  - && (E)
  - || (Ou)
  - ! (Não)

R: Estrutura Condicional

```r
a=1
if(a>=1){
   print(a)
}
```

R: Estrutura Condicional

```r
a=1
if(a>=1 && a <= 3){
   print(a)
}
```

R: Estrutura Condicional

```
a=1
if(a>4 ||a < 3){
  print(a)
}else{
  print('dentro do else')
}
```

R: Estrutura Condicional

```
a=1
if(a>4){
   print(a)
}else if(a<3){
   print('dentro do else')
}
```

R: Estrutura Condicional

```
a=3
if(a>4){
  print(a)
}else if(a<3){
  print('dentro do else1')
}else if(a==2){
  print('dentro do else2')
}else if(a==1){
  print('dentro do else3')
}else{
  print('else')
}
```

Estrutura Condicional: Exemplo

- A nota final de um estudante é calculada a partir de três notas atribuídas, respectivamente, a um trabalho de laboratório, a uma avaliação semestral e a um exame final. As média das três notas mencionadas obedece aos seguintes pesos:
  - Trabalho do Laboratório: 2
  - Avaliação Semestral: 3
  - Exame Final: 5
- Elabore um algoritmo para um programa que receba as três notas, calcule a média ponderada do aluno e classifique se o aluno está aprovado (média >= 5) ou reprovado (média < 5)

R: Vetor e Matriz

```
> a=c(1,2,3)
> a
[1] 1 2 3
> a=1:10
> a
 [1]  1  2  3  4  5  6  7  8  9 10
> a=seq(1,10,by=2)
> a
[1] 1 3 5 7 9
> a=matrix(c(1,2,3,4,5,6,7,8,9),nrow=3,ncol=3)
> a
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
```

R: Estruturas de Repetição

```
a=1
while(a<10){
   print(a)
   a=a+1
}
```

R: Estruturas de Repetições

```
for(i in 1:10){
    print(i)
}
```

R: Estruturas de Repetições

```r
for(i in seq(1,10,by=2)){
  print(i)
}
```

R: Estruturas de Repetições

```r
for(i in seq(10,1,by=-1)){
  print(i)
}
```

R: Estruturas de Repetições

```
for(i in seq(10,1,by=-0.1)){
  print(i)
}
```

R: Percorrendo Vetor

```
v=seq(10,1,by=-0.1)
for(i in 1:length(v)){
  print(v[i])
}
```

R: Percorrendo Matriz

```
v=matrix(c(1,2,3,4,5,6,7,8,9),nrow=3,ncol=3)
t=dim(v)
for(i in 1:t[1]){
  for(j in 1:t[2]){
    print(v[i,j])
  }
}
```

R: Funções

```
funcao <- function (input_args){

  return (output_arg)
}
```

Estrutura de Repetição: Exemplo

- Escreva um algoritmo que lê 15 valores reais, armazene em um vetor e depois, encontra o maior e o menor deles e mostra o resultado

Estrutura de Repetição: Exemplo

- Faça um algoritmo para imprimir a tabuada do 1 ao 10.

R: Help

- Para ver o que uma determinada função do Matlab faz basta digitar:
- ??<FUNÇÃO>
- Ex: ??readline

R: Leitura de Arquivos

```
>> dados=xlsread('excel.xls');
>> dadosNovo=xlsread('excelNovo.xlsx');
```

R: Leitura de Arquivos

- dados= read.csv ('FILENAME',header=TRUE,sep=“,”);
- Onde header significa que a primeira linha representa o cabeçalho dos dados
- E sep representa qual o caracter que esta separando as colunas
- Ambos são opcionais

Sumarização de Dados

Sumarização de Dados

- Sumarização em Linhas e Colunas
  - Use `apply()` em um data frame ou matriz para aplicar uma função em colunas ou linhas. O código genérico é `apply(<dados>,<row/col>,<function>)`, onde:
    - `<dados>` é um data frame ou matriz
    - `<row/col>` é uma opção numérica indicando onde será aplicado a função
      - 1 - em linhas
      - 2 - em colunas
    - `<function>` é uma custom function ou qualquer função do R. Exemplos: `mean, min, length, sd, which.min, which.max.`
  - Valores nulos podem ser excluídos adicionando o parâmetro `na.rm=TRUE`

Sumarização de Dados

- Sumarização em Linhas e Colunas
  - Exemplo 01: Valores mínimos de linha
    - `apply(dados, 1, min)`

dados

Obs1	Obs2
5	0
1	7
6	2

0	1	2

Sumarização de Dados

- Sumarização em Linhas e Colunas
  - Exemplo 02: Valores mínimos de linha, ignorando valores nulos
    - `apply(dados, 2, min, na.rm=TRUE)`

dados

Obs1	Obs2
5	0
1	7
6	NA

`apply(data, 2, min, na.rm=TRUE)`

Obs1	Obs2
1	0

Sumarização de Dados

- Sumarização em Linhas e Colunas
  - Exemplo 03: Para cada linha, determinal qual coluna contém o menor valor
    - `apply(data, 1, which.min)`

dados

Obs1	Obs2
5	0
1	7
6	2

Sumarização de Dados

- Sumarização em várias colunas
  - Use `sapply()` para aplicar uma função em várias colunas. O código genérico é
  - `sapply(<dados>,<function>)`, onde:
    - `<dados>` é um data frame
    - `<function>` é uma custom function ou uma função nativa. Exemplos: `mean, min, sd, length, which.min, which.max.`
  - Valores nulos podem ser excluídos adicionando o parâmetro `na.rm=TRUE`

Sumarização de Dados

- Sumarização em várias colunas
  - Exemplo 01: Médias das colunas
    - `sapply(dados,mean)`

dados

x	y
5	3
11	5
5	1

x	y
7	3

Sumarização de Dados

- Sumarização em várias colunas
  - Exemplo 02: Médias das colunas, ignorando os valores nulos
    - `sapply(dados,mean,na.rm=TRUE)`

dados

x	y
5	3
NA	5
5	1

`sapply(dados,mean,na.rm=TRUE)`

x	y
5	3

Sumarização de Dados

- Sumarização em várias colunas
  - Exemplo 03: Custom Function
    - `custom.fun <- function(x) 2+x # Adiciona 2 a cada número`
    - `sapply(dados,custom.fun)`

dados

x	y
5	3
11	5
5	1

```
custom.fun <- function(x) 2+x
sapply(dados,custom.fun)
```

x	y
7	5
13	5
7	3

Sumarização de Dados

- Sumarização em várias colunas
  - Use `lapply()` em um data frame para aplicar uma função em colunas e possuir uma lista de retorno. O código genérico é `lapply(<dados>,<function>)`, onde:
    - `<dados>` é um data frame
    - `<function>` é uma custom function ou uma função nativa. Exemplos: `mean, min, sd, length, which.min, which.max.`
  - Valores nulos podem ser excluídos adicionando o parâmetro `na.rm=TRUE`

Sumarização de Dados

- Sumarização em várias colunas
  - Exemplo 01: Valores mínimos de colunas
    - `lapply(dados,min)`

dados

Obs1	Ob2
5	0
1	7
6	2

Obs1

1

Obs2

0

Sumarização de Dados

- Sumarização em várias colunas
  - Exemplo 02: Médias das colunas, ignorando valores nulos
    - `lapply(dados,mean,na.rm=T)`

dados

Obs1	Ob2
5	0
1	NA
6	2

Obs1

4

Obs2

1

Sumarização de Dados

- Sumarização em várias colunas
  - Observação: `lapply()` é similar a `sapply()`
    - `lapply()` retorna uma lista;
    - `sapply()` retorna um vetor ou matriz

Sumarização de Dados

- Agrupamento de dados com uma coluna
  - Use `tapply()` para aplicar uma função dentro de uma coluna através de um grupo(s). O código genérico é `tapply(<coluna_com_os_valores>, <coluna_com_os_grupos>, function>)`, onde:
    - `<coluna_com_os_valores>` são os valores a serem agrupados
    - `<coluna_com_os_grupos>` são os grupos
    - `<function>` é uma custom function ou uma função nativa. Exemplos: `mean, min, sd, length, which.min, which.max.`
  - Valores nulos podem ser excluídos adicionando o parâmetro `na.rm=TRUE`

Sumarização de Dados

- Agrupamento de dados com uma coluna
  - Exemplo 01: Média da Coluna
    - `tapply(dados$X, data$Site, mean)`

dados

Site	X
A	3
A	7
A	2
B	1
B	5
B	0

`tapply(dados$X, data$Site, mean)`

A	B
4	2

Sumarização de Dados

- Agrupamento de dados com uma coluna
  - Exemplo 02: Valores mínimos da Coluna, ignorando valores nulos
    - `tapply(data$X, data$Site, min, na.rm=TRUE)`

dados

site	X
A	3
A	NA
A	2
B	1
B	5
B	0

```
tapply(data$X, data$Site,
min, na.rm=TRUE)
```

A	B
2	0

Sumarização de Dados

- Agrupamento de dados com uma coluna
  - Exemplo 03: Tabulação cruzada com duas colunas
    - `tapply(data$X, list(data$Ano, data$Site), mean)`

dados

Ano	Site	X
2001	A	3
2002	A	7
2002	A	2
2001	B	1
2002	B	5
2002	B	0

`tapply(data$X, list(data$Ano, data$Site), mean)`

	A	B
2001	3.0	1.0
2002	4.5	2.5

Sumarização de Dados

- Agrupamento de dados com uma coluna
  - Observação: `tapply()` retorna uma tabela, com as variáveis do grupo retornado como fatores. Use `data.frame()` para converter para um data frame e `as.numeric(as.character(dados$X))` para converter fatores em variáveis numéricas.

Exercício

- Realize a leitura dos dados do arquivo ‘dados.csv’ e faça a média de cada coluna

Regressão Linear

Regressão

- É uma técnica que permite explorar e inferir a relação de uma variável dependente (variável de resposta) com variáveis independentes específicas (variáveis explicatórias).

Regressão Linear

- Quando a variável dependente é linearmente proporcional a variável independente

Regressão Linear

- No R temos a função ‘lm’ que é utilizada para ‘fitar’ modelos lineares

- Ex: fit=lm(y~x)

Regressão Linear

- Pode-se utilizar também modelos polinomiais com a função ‘poly’
- Ex: fit=lm(y~poly(x,2))
- O segundo parâmetro representa o grau do polinômio

Regressão Linear

- Depois de criado o modelo é necessário utilizá-lo para obter os dados previstos com ele
- Para isso utilizamos a função ‘predict’
- Ex: y1=predict(fit, data.frame(x=x))

Regressão Linear

- Pode-se obter algumas estatísticas com a função ‘summary’

- Ex: summary(fit)

Regressão Linear

- Pode-se obter os coeficientes do modelo com ‘coef’
- Ex: coef(fit)

Regressão Linear

- Realiza-se a plotagem para verificar o modelo

- Ex: plot(x,y)
- lines(x,y1)

Exercício

- Realize a leitura dos dados do arquivo ‘dados.csv’ e faça a regressão de um das variáveis com a variável PAR

ANOVA

ANOVA

▶É a técnica estatística que permite avaliar afirmações sobre as médias de populações, ou seja, visa, fundamentalmente, verificar se existe uma diferença significativa entre as médias e se os fatores exercem influência em alguma variável dependente

ANOVA

- Para realizar a anova no R basta utilizar a função ‘aov’
- Ex: resp =aov(var_dependente~var_independente)

ANOVA

- Para extrair as estatísticas utilize a função ‘summary’
- Ex: summary(resp)

ANOVA

- Lembre-se que TODA análise estatística possui pressuposto, no caso da ANOVA é que o resíduos possuam distribuição normal
- Para isso utilize-se da função ‘shapiro.test’
- Ex: shapiro.test(resp$residuals)

ANOVA

- Para visualizar os grupos pode-se utilizar os gráficos boxplot
- Ex: boxplot(var_dependente~var_independente)

ANOVA: Exemplo

- Utilizando o arquivo exemplo1 realize a anova das colunas F (tamanho da planta) e P (adubo utilizado)
- O que podemos deduzir?

Exercício

- Realize a leitura dos dados do arquivo ‘exemplo1.csv’ e verifique se existe diferença entre os tratamentos

Plotagem

Plotagem de Gráficos

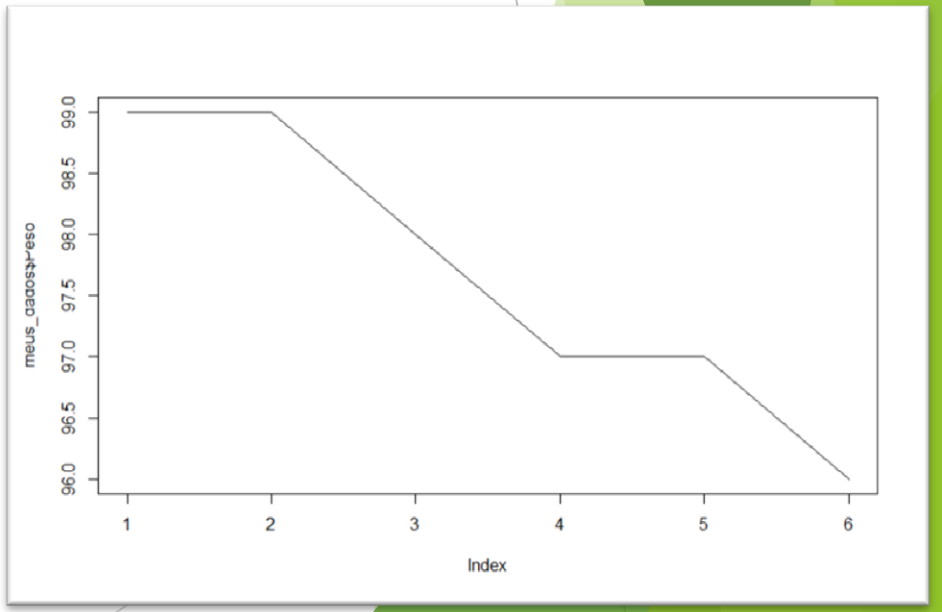
- Criando Gráficos
  - Linhas
    - Use `plot(x, y, type = "l")` para produzir um gráfico de linha.
      - `type = "p"`: pontos
      - `type = "l"`: linhas
      - `type = "b"`: linhas e pontos

meus_dados

Dia	Peso
154	99
155	99
156	98
157	97
156	97
155	96

```
plot(meus_dados$Peso, type="l")
```

Criar um gráfico de linha usando a coluna “Peso”



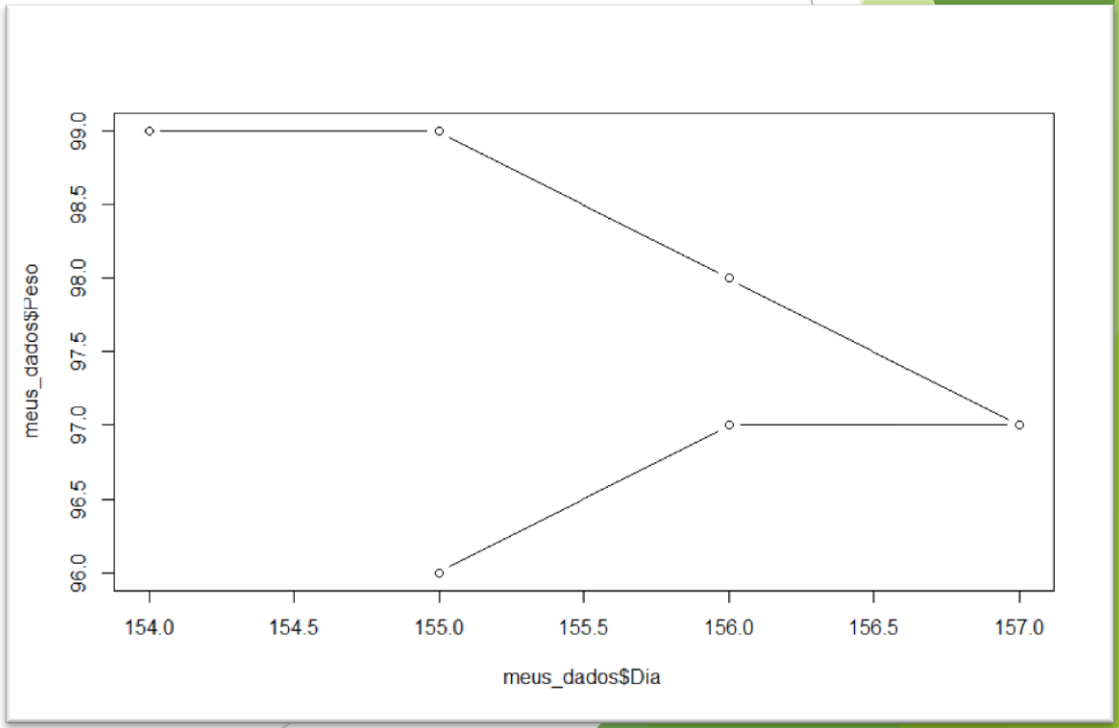
Plotagem de Gráficos

- Criando Gráficos
  - Linhas
    - Use Linhas e pontos para traçar um gráfico Dia vs peso
      - `plot(meus_dados$Dia,meus_dados$Peso, type="b")`

meus_dados

Dia	Peso
154	99
155	99
156	98
157	97
156	97
155	96

```
plot(meus_dados$Dia,meus
_dados$Peso, type="b")
```



Plotagem de Gráficos

- Criando Gráficos
  - Linhas
    - Nota: Existem dois formatos diferentes para traçar gráficos:
      - `plot(y ~ x, type="l")`
      - `plot(y,x, type="l")`

Plotagem de Gráficos

- Criando Gráficos
  - Histograma
    - Um histograma é uma representação gráfica de uma distribuição de dados. Um histograma pode exibir a frequência ou densidade (frequência relativa) de observações ao longo de um intervalo de dados especificado.
      - `hist(dados$coluna)# Histograma de Frequência`
      - `hist(dados$coluna,freq=FALSE) # Histograma de densidade`

Plotagem de Gráficos

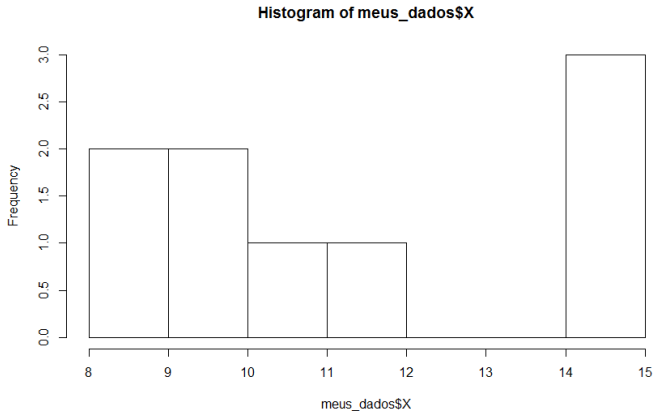
- Criando Gráficos
  - Histograma

meus_dados

X
15
10
12
11
15
10
9
8
15

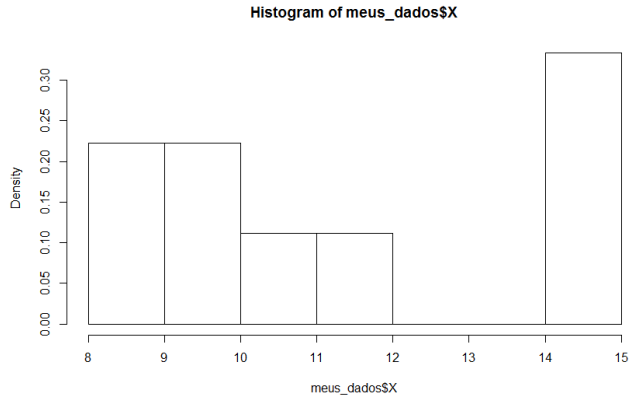
```
hist(meus_dados$X)
```

Histograma de Frequência




```
hist(meus_dados$X,freq=FALSE)
```

Histograma de Densidade



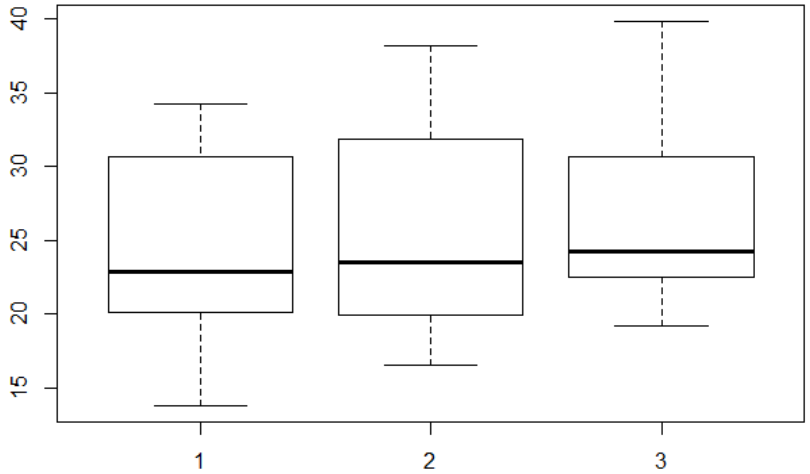
Plotagem de Gráficos

- Criando Gráficos
  - Boxplot

meus_dados

Y	X
1	15
1	10
1	12
2	11
2	15
2	10
3	9
3	8
3	15

```
boxplot(meus_dados$X~dados$Y)
```



Plotagem de Gráficos

- Argumentos de Plotagem
  - Rótulos de Eixo e Títulos

Argumento	Descrição
xlab=“legenda"	Legenda do eixo X
ylab=“legenda"	Legenda do eixo Y
main=“titulo"	Título do gráfico
sub=“subtitulo"	Sub Título do gráfico
cex.axis=numero	Tamanho de anotação dos eixos X e Y
cex.lab=numero	Tamanho de legenda dos eixos X e Y
cex.main=numero	Tamanho do título
cex.sub=numero	Tamanho do sub título
col.lab="cor"	Cores do eixo X e Y
col.main="cor"	Cor do título
col.sub="cor"	Cor do sub título
las=numero	Estilo de rótulos do eixo, paralelo (0), horizontal (1), perpendicular (2), vertical (3).
xaxt="s"	Estilo do eixo-x: "s" é o padrão; "n" suprime plotagem do eixo.
yaxt="s"	Estilo do eixo-y: "s" é o padrão; "n" suprime plotagem do eixo.

Plotagem de Gráficos

- Argumentos de Plotagem
  - Opções de Visualização de dados

Argumento	Descrição
bg="cor"	Cor do plano de fundo
bty="opção"	Estilo da borda em torno do gráfico, opções: "o", "l", "7", "c", "u", "]"
cex=numero	Tamanho dos pontos sobre o gráfico
col="cor"	Cor do objeto plotado
fg="cor"	Cor de primeiro plano do gráfico
lty=numero	Tipo de linha, opções (1,2,3,4,5,6)
lwd=numero	Largura da linha
new=logico	Desenhar outro gráfico em cima de outro existente(TRUE or FALSE)
pch=numero	Formato do ponto (entre 1 e 25)
type="opção"	Tipo de gráfico(option: "p", "l", "b", "c", "S", "n", "o", "h" "s")
xlim, ylim	Limites dos eixos do gráfico

Plotagem de Gráficos

- Argumentos de Plotagem
  - Customizações de plotagem
    - Definido usando `par()` antes de plotar. Exemplo:
      - `par(ask=T)`
      - `plot(x,y)`

Argumento	Descrição
ask=logico	Pergunte antes de prosseguir para o próximo gráfico(TRUE/FALSE)
bg="cor"	Cor de fundo da região do gráfico
mai=c(bottom, left, top, right)	Tamanho da Margem de plotagem (polegadas)
mar=c(bottom, left, top, right)	Tamanho da Margem de plotagem (linhas)
mfrow=c(row, columns)	Número e disposição dos números
mfcol=c(row, columns)	Número e disposição dos números
oma=c(bottom, left, top, right)	Tamanho da margem externa (linhas)
omi=c(bottom, left, top, right)	Tamanho da margem externa (polegadas)

Mapas

Mapas

- library(maps) #mapas simples, eixos, escala, cidades
- library(mapdata) #base de dados WorldHires e rios
- library(rworldmap) #outra base de dados de mapas do mundo
- library(maptools) #Ler ESRI shapefiles
- library(mapproj) #Projeções e grids
- library(ggmap) #Gmaps, OSM + mapas baseados em ggplot2
- library(rgdal)

Mapas

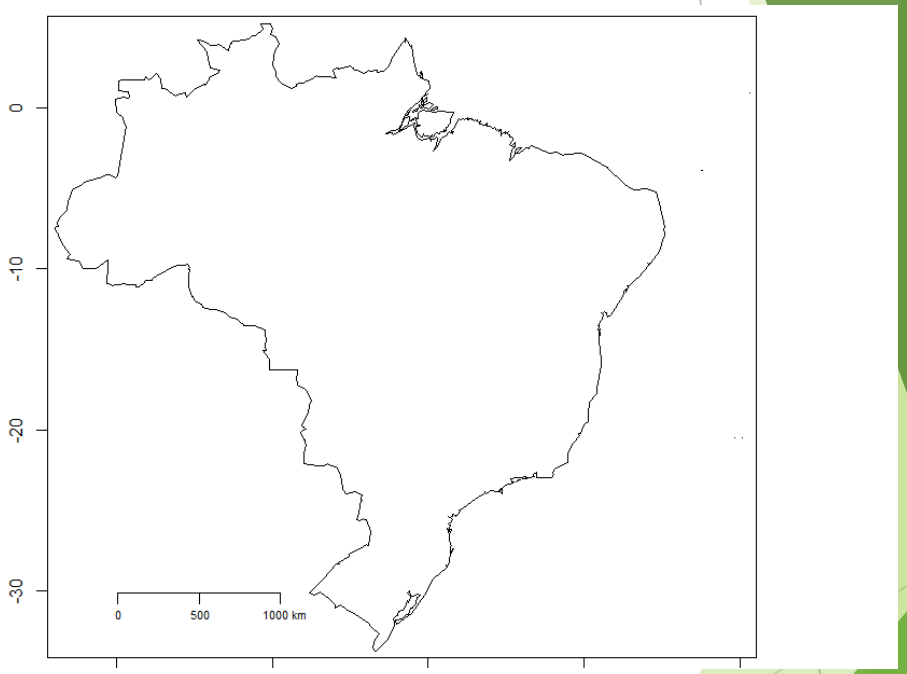
- library(maps)
- library(mapdata)
- map("worldHires","Brazil")

Mapas

- map.axes()

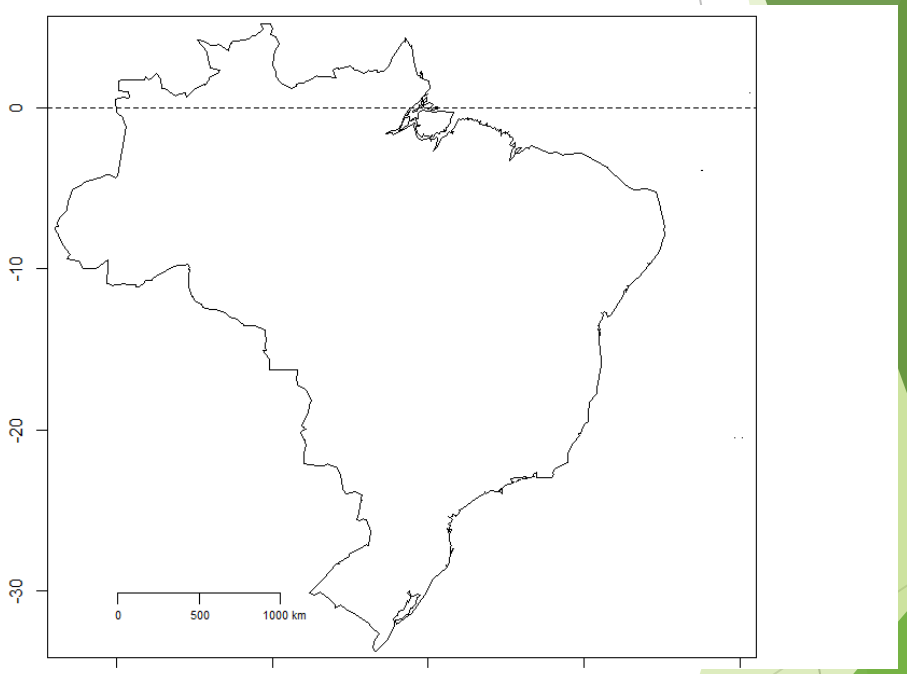
Mapas

- map.scale(ratio = F, cex = 0.7)



Mapas

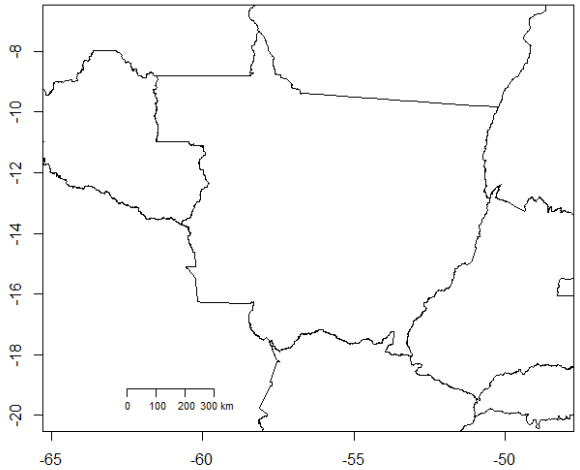
- abline(h = 0, lty = 2)



Mapas

- map("world","Brazil", fill=T, col="grey90")
- map(,,add=T)
- map.axes()
- map.scale(ratio=F, cex=0.7)
- abline(h=0, lty = 2)
- map.cities(country = "Brazil", minpop=2000000, pch=19, cex=1.2)# pacote maps

Mapas

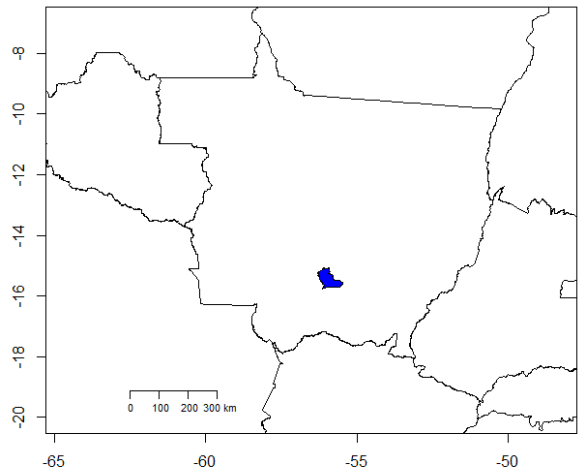


- library(maps)
- library(maptools)

- estados <- readShapePoly("estados_2010/estados_2010.shp")
- plot(estados,xlim=c(-63,-50),ylim=c(-20,-7))
- map.axes()
- map.scale(ratio = F, cex = 0.7)

Mapas

- estados <- readShapePoly("estados_2010/estados_2010.shp")
- municipios <- readShapePoly("municipios_2010/municipios_2010.shp")
- cuiaba = municipios[municipios$nome=='Cuiabá',]
- plot(estados,xlim=c(-63,-50),ylim=c(-20,-7))
- plot(cuiaba,col=c('blue'),add=TRUE)
- map.axes()
- map.scale(ratio = F, cex = 0.7)



Apoio

- Ricardo Frederico Figueiredo e Salles
- http://rstudio-pubs-static.s3.amazonaws.com/176768_ec7fb4801e3a4772886d61e65885fbdd.html