

Programação com R

Elifranio Alves Cruz, Msc

Agenda R

- **Sintaxe e Configurações Gerais**

- Introdução ao R; Usando o R com Gui e linha de comando
- Histórico; Pacotes e gerenciamento; Arquivos e Diretórios
- Opções de configuração, Encerrando o R
- Tipos de Dados; Comentários; Operadores
- Formulas; Datas; Estrutura de Objetos
- Conjunto de dados Nativos do R; Vetores; Matrizes e Arrays
- Listas; Data Frame; Séries Temporais; Fatores; outras funções
- Attach; Detach e With; Criando sequências; Length; Scan
- Conversão de Tipos; Importando dados; visualizando linhas e colunas
- Funções cumulativas; Aplicando funções sobre linhas e colunas;
- Tabelas de Contingência; Números aleatórios; Estruturas de programação;
- Estruturas Condicionais; laços; Criando funções

```
plot( iris$Sepal.Length, iris$Sepal.Width,ylab= "Largura" ,xlab = "Comprimento", col="blue", main="Iris", pch=20)
```

Introdução ao R;

➤ Usando o R com Gui e linha de comando

- `y = cars[,1]; x = cars[,2]`
- `lm(y ~ x , cars)`
- `cars`
- `lm(formula=y ~ x ,data= cars)`
- `lm(y ~ x , cars, qr=FALSE)`
- `getwd()`
- `getwd(); a=3; print(a)`
- `plot(iris$Sepal.Length, iris$Sepal.Width,ylab= "Largura" ,xlab = "Comprimento", col="blue", main="Iris", pch=20)`

➤ Histórico

- `history(max.show=5)`

➤ Pacotes

- `installed.packages()`
- `install.packages("ggplot2",dependencies=TRUE)`

Instruções Gerais do R

➤ Pacotes

- `library(ggplot2)`
- `search()`
- `detach("package:ggplot2")`
- `remove.packages("ggplot2")`
- `getOption("defaultPackages")`

➤ Obtendo Ajuda

- `help.start()`
- `help("search")`
- `?search`
- `apropos("search")`
- `args("sd")`

➤ Arquivos e Diretórios

- `getwd()`
- `setwd("C:/Data")`
- `getwd()`
- `dir()`

➤ Opções de Configuração

- `2/3`
- `getOption("OutDec")`
- `options(OutDec=",")`

➤ Encerrando o R

- `save.image()`
- `save.image("ProjB.RData")`
- `load(file=" ProjB.RData")`

➤ Tipos de Dados

- `var1 <- 10`
- `var1`
- `class(var1)`
- `x <- 10`
- `x`
- `x[1]`
- `is.numeric(var1)`
- `var2 <- "Estatística"`
- `Var2`

Instruções Gerais do R

➤ Tipos de Dados

- `alfa <- "Estatística: "`
- `delta <- "R "`
- `x <- paste(alfa,delta)`
- `X`
- `varInt1 <- 10L`
- `VarInt2 <- as.integer(10)`
- `class(varInt1);class(VarInt2)`
- `Pop <- 5345L`
- `Cons <- 45667778L`
- `Med <- Cons / Pop`
- `Med`
- `6 * 2 + 10`
- `6 * (2 + 10)`
- `objects()`

➤ Tipos de Dados

- `val <- 10`
- `objects()`
- `val`
- `rm(val)`
- `val`
- `objects()`
- `a <- 10`
- `b <- 10`
- `c <- 20`
- `objects()`
- `rm(list=objects())`
- `objects()`
- `a <- NA`
- `A`
- `a + 1`
- `is.na(a)`

➤ Comentários

- `#Testar se o valor de "a" é NA`
- `is.na(a)`

Instruções Gerais do R

➤ Fórmulas

- fórmula = class ~ checking_status + duration + credit_history + purpose
- search()
- detach("package:ggplot2")
- remove.packages("ggplot2")
- getOption("defaultPackages")

➤ Datas

- data <- as.Date("2013-01-31")
- Data
- data <- as.Date("2013-01-31")
- data2 <- as.Date("2013-01-25")
- dif <- data - data2
- Dif
- data <- as.Date("2013-03-25")
- format(data, format="%d/%m/%Y")
- format(data, format="%A, %d de %B de %Y")

➤ Conjuntos de Dados Nativos do R

- data()
- USArrests
- data(package = .packages(all.available = TRUE))

➤ Conjuntos de Dados Nativos do R

- library(cluster)
- Agriculture
- data(package = .packages(all.available = TRUE))
- library(cluster)
- Agriculture
- help(agriculture)

➤ Vetores

- vetNum <- c(10,2,3,6,5)
- vetChar <- c("a","b","c")
- vetBool <- c(T,T,F,T)
- vetNum
- vetChar
- vetBool

Instruções Gerais do R

➤ Vetores (continuação)

- `rivers[3]`
- `rivers[c(1,3,50)]`
- `rivers[c(1:5)]`
- `rivers[rivers>1400]`
- `rivers[-1]`
- `max(rivers)`
- `min(rivers)`
- `length(rivers)`
- `sort(rivers)`
- `sort(rivers,decreasing=TRUE)`
- `order(rivers)`
- `vetNum = c(10,2,3,6,5)`
- `vetNum2 = c(4,5,3,9,12)`
- `vetNum + vetNum2`
- `vetNum`
- `vetNum + 5`

➤ Vetores (continuação)

- `vetNum`
- `vetNum3`
- `vetNum + vetNum3`
- `vetNum3[4:5] <- 0`
- `vetNum3`
- `vetNum + vetNum3`
- `a <- c(10,15,20,NA)`
- `sd(a)`
- `sd(a,na.rm=TRUE)`
- `0 / 0`

➤ Matrizes e Arrays

- `USPersonalExpenditure`
- `USPersonalExpenditure[1,3]`

Instruções Gerais do R

➤ Matrizes e Arrays

- `USPersonalExpenditure[1:2,c(1,3)]`
- `USPersonalExpenditure[,1]`
- `USPersonalExpenditure[1,]`
- `USPersonalExpenditure["Food and Tobacco","1940"]`
- `Mtrx = matrix(c(1,2,3,4,5,6),nrow=2,ncol=3,byrow=TRUE)`
- `Mtrx`
- `Mtrx = arix(c(1,2,3,4,5,6),nrow=2,ncol=3,byrow=FALSE)`
- `Mtrx`
- `dimnames(Mtrx) = list(c("Lin1","Lin2"),c("Col1","Col2","Col3"))`
- `Mtrx`
- `newUSP <- USPersonalExpenditure`
- `newUSP`
- `dim(USPersonalExpenditure)`

➤ Listas

- `ability.cov`
- `ability.cov$center`
- `class(ability.cov$cov)`
- `class(ability.cov$center)`
- `ability.cov$cov[1,3]`
- `ability.cov[[1]][2,1]`
- `ability.cov[[3]][1]`
- `minhaLista <- list(USPersonalExpenditure,ability.cov$center,c(2,5,6,9),"Elemento")`
- `minhaLista`

➤ Data Frame

- `animals`
- `animals[5,5]`
- `animals["ant","gro"]`
- `animals$war`

Instruções Gerais do R

➤ Data Frame

- `class(DNase$Run)`
- `class(DNase$conc)`
- `Índice = c(1, 2, 3)`
- `Nomes = c("Maria", "Pedro", "José")`
- `Brasileiro = c(TRUE, TRUE, FALSE)`
- `peessoas = data.frame(Indice, Nomes, Brasileiro)`
- `peessoas`
- `class(peessoas$Indice)`
- `class(peessoas$Nomes)`
- `class(peessoas$Brasileiro)`
- `Dados <- edit(DNase)`
- `fix(DNase)`

➤ Séries Temporais

- `class(AirPassengers)`
- `AirPassengers`
- `mst <- ts(c(1:48), start=c(2010, 1), end=c(2013, 12), frequency=12)`
- `mst`
- `minhaLista`

➤ Fatores

- `dias <- factor(c(1:7), labels=c("Dom", "Seg", "Ter", "Qua", "Qui", "Sex", "Sab"), ordered=TRUE)`
- `dias`

➤ Attach, Detach e With

- `head(cars)`
- `sum(speed)`
- `attach(cars)`
- `sum(speed)`
- `detach(cars)`

Instruções Gerais do R

➤ Attach, Detach e With

- `with(cars, sum(speed))`
- `var1 <- c(1:10)`
- `var1`
- `seq(from=10, to=100, by=10)`
- `seq (from=1, to=10, by=0.5)`

➤ Length

- `varA <- c(10,15,20,25)`
- `varB <- c(10,15,20,25,30)`
- `length(varA)`
- `length(varB)`
- `length(varA) <- length(varB)]`
- `length(varA)`
- `varA`

➤ Scan

- `meuVetor = scan()`
- `meuVetor = scan()`
- `meuVetor`
- `meuVetor = scan(what="character")`
- `meuVetor`
- `meuVetor = scan(what="boolean")`
- `meuVetor`

➤ Conversões de Tipos

- `var <- "9"`
- `class(var)`
- `var1 <- as.numeric(var)`
- `class(var1)`
- `class(cars)`
- `lista = as.list(cars)`
- `class(lista)`
- `class(ability.cov)`
- `dataframe <- as.data.frame(ability.cov)`
- `class(dataframe)`

Instruções Gerais do R

➤ Importando dados

- `tabela <- read.table("Fraud.csv",sep=";")`
- `class(tabela)`
- `arquivo <- file.choose()`
- `tabela <- read.table(arquivo,sep=";")`
- `tabela <- read.table(file.choose(),sep=";")`
- `list.files("C:\\Data")`
- `dir(".\\older")`

➤ Lendo e Salvando Dados em Disco

- `carros <- cars`
- `save(carros,file="carros.rdata")`
- `rm(carros)`
- `carros`
- `load("carros.RData")`
- `head(carros)`
- `save(cars,Titanic,file="carros.rdata")`
- `head(DNase)`
- `head(DNase,n=4L)length(varA)`

➤ Head e Tail

- `head(DNase)`
- `head(DNase,n=4L)`

➤ Sumarizando Dados

- `summary(DNase)`

➤ Visualizando Nomes de Linhas e Colunas

- `colnames(USArrests)`
- `rownames(USArrests)`

➤ Funções Cumulativas

- `AirPassengers`
- `cumsum(AirPassengers)`
- `cummax(AirPassengers)`

➤ Aplicando Funções Sobre Linhas ou Colunas

- `apply(USPersonalExpenditure,1,median)`
- `apply(USPersonalExpenditure,2,median)`

Instruções Gerais do R

➤ Tabelas de Contingência

- `head(infert)`
- `table(infert$education,infert$induced)`

➤ Números Aleatórios

- `sample(10)`
- `amostra <- sample(1:1000,size=10,replace=FALSE)`
- `amostra`
- `set.seed(2013)`
- `sample(1:1000,size=10,replace=FALSE)`
- `set.seed(2013)`
- `sample(1:1000,size=10,replace=FALSE)`
- `sample(1:1000,size=10,replace=FALSE)`
- `sample(1:1000,size=10,replace=FALSE)`
- `amostra <- sample(1:141,size=10,replace=FALSE)`
- `rios <- rivers[c(amostra)]`
- `rios`

➤ Estruturas de Programação

- `a <- 1`
- `b <- 5`
- `if (a<b) print("Menor")`
- `if (a<b) print("Menor") else print ("Maior")`
- `if (a>0 & a<=1) print("verdadeiro")`
- `if (a>0 & a<=1)`
 - `{`
 - `print("verdadeiro")`
 - `print("Mais uma função")`
 - `} else`
 - `{`
 - `print("falso")`
 - `print("Mais uma função")`
 - `}`
- `head(DNase,n=4L)`

Instruções Gerais do R

➤ Laços

- `a<-0`
- `repeat {`
- `if (a == 5) break else`
- `a<-a+1`
- `print(a) }`
- `#`
- `while (a < 5) {`
- `a <- a + 1;`
- `print(a) }`
- `#`
- `for (i in women[,1]) print(i)`

➤ Criando Funções

- `par = function(x){`
- `x%%2 == 0`
- `}`
- `par(10)`
- `par(5)`
- `par = function(x){`
- `if (! is.integer(x)) warning("Valor não inteiro")`
- `x%%2 == 0`
- `}`
- `par(2)`
- `par(2L)`
- `par(2)`

