

# Introdução ao R

Cristiano de Carvalho Santos  
professor.pacotes.estatisticos@gmail.com  
Grupo Google: PacotesEstatisticos2016

Departamento de Estatística,  
Universidade Federal de Minas Gerais (UFMG)

# Aspectos gerais

- ▶ Onde baixar o R: <https://cran.r-project.org/>
- ▶ O R faz distinção entre maiúsculas e minúsculas
- ▶ Utiliza o ponto como separador de casas decimais
- ▶ Utiliza notação científica para representar números muito grandes ou pequenos. Ex:  $10^5 = 1e + 05$
- ▶ Comentários em um script são feitos utilizando o #
- ▶ A função `citation( )` indica como citar o R

# Comandos de ajuda no R

- ▶ Para saber mais informações sobre uma função use:  
`help(nome da função)` ou `?nome da função`
- ▶ Mas o que fazer quando não sabemos qual função do R faz a análise desejada? Ex: `help.search(median)` ou `??median`
- ▶ Você também pode buscar ajuda na internet, no site do R, com o comando `RsiteSearch( )`
- ▶ No menu principal, em Ajuda, são disponíveis alguns manuais e comandos de ajuda.
- ▶ Existem muitas apostilas sobre R e fóruns de discussão na internet

# Demonstrações

- ▶ `demo()`
- ▶ `demo(graphics)`
- ▶ `demo(image)`
- ▶ `demo(persp)`

# Tipos de objetos

Os tipos básicos de objetos do R são:

- ▶ vetores
- ▶ matrizes e array
- ▶ data-frames
- ▶ listas
- ▶ funções

# Funções

- ▶ Função é um tipo objeto especial que recebe algum "input" e produz um "output"
- ▶ O conteúdo das funções podem ser vistos digitando o nome da função (sem os parênteses)
- ▶ Entretanto isto não é disponível desta forma para todas as funções como por exemplo em min, max, rnorm e lines
- ▶ Podemos criar novas funções manualmente

# Vetores

- ▶ Vetores são o tipo básico e mais simples de objeto para armazenar dados no R.
- ▶ Podem ser formados por números, nomes, elementos lógicos, desde que todos os elementos sejam do mesmo tipo
- ▶ O R é uma linguagem vetorial, e portanto capaz de operar vetores e matrizes diretamente sem a necessidade de "loops".

## Algumas funções relacionadas

- ▶ `c( )` (“c” de concatenar), `scan( )`
- ▶ `rep( )`, `seq( )`, “:” são usados para indicar seleção de elementos
- ▶ Os colchetes [ ] são usados para indicar seleção de elementos
- ▶ `is.vector( )`, `is.numeric( )`, `is.character( )`, `length( )`, `mode( )`
- ▶ `as.vector( )`, `names`, etc



# Matriz

- ▶ Uma matriz é uma coleção de vetores de mesmo comprimento organizados um do lado do outro.
- ▶ Todos os elementos de um vetor e também de uma matriz devem ser do mesmo tipo, isto é, devem ser todos numéricos ou devem ser todos caracteres.

## Algumas funções relacionadas

- ▶ `matrix`, `cbind`, `rbind`, `diag`
- ▶ `dim`, `length`
- ▶ `colnames`, `rownames`, `dimnames`
- ▶ `sum`, `rowSums`, `colSums`
- ▶ Os sinais `*` e `/` fazem operações ponto a ponto, e para isso as matrizes envolvidas devem ter a mesma dimensão
- ▶ Para multiplicação matricial você deve usar `% * %`
- ▶ As funções `t(A)` e `solve(A)` retornam as matrizes transposta e inversa de uma matriz  $A$  definida anteriormente

# Array

- ▶ O conceito de array generaliza a ideia de matriz
- ▶ Enquanto em uma matriz os elementos são organizados em duas dimensões (linhas e colunas), em um array os elementos podem ser organizados em um número arbitrário de dimensões
- ▶ No R um array é definido utilizando a função `array()`

# Data.frame

- ▶ O “data-frame” é uma estrutura semelhante à uma matriz porém com cada coluna sendo tratada separadamente.
- ▶ Podemos ter colunas de valores numéricos e colunas de caracteres no mesmo objeto.
- ▶ Note entretanto que dentro de uma mesma coluna todos elementos ainda serão forçados a serem do mesmo tipo.
- ▶ O \$ pode ser usado para selecionar uma das colunas

## Algumas funções relacionadas

- ▶ data.frame, fix, edit, expand.grid

# Lista

- ▶ Listas são estruturas genéricas e flexíveis que permitem armazenar diversos formatos em um único objeto.
- ▶ Uma lista é portanto uma coleção de objetos.
- ▶ Para listas há duas opções para se selecionar elementos: colchetes [ ] ou colchetes duplos [[ ]].

## Algumas funções relacionadas

- ▶ list

# Bancos de dados já disponíveis no R

- ▶ Com o comando `data()` vemos quais os bancos dados já carregados no R
- ▶ Informações sobre estes bancos de dados podem ser obtidas usando a função `help(nome do banco)`
- ▶ Os bancos de dados podem ser acessados usando apenas o nome

# Como criar uma função

```
Nome = function(argumentos)
{
    processamento de alguma conta
    return(resultado)
}
```

**Exemplo:** podemos criar a função  $fc$  que calcule  $a^+b$  para valores  $a$  e  $b$  passados como argumentos da função

```
fc = function(a, b)
{
    c = a^b+b
    return(c)
}
```

- ▶ Apenas um objeto pode ser retornado em cada função
- ▶ Se desejarmos retornar mais de um objeto podemos criar uma lista
- ▶ Se não usarmos a função *return*, a função construída retornará o último resultado calculado dentro da função
- ▶ Podemos usar a função *print* para imprimir na tela o resultado de alguma conta realizada ou frase dentro da função



# Objetos na área de trabalho

- ▶ A funções *objects*( ) e *ls*( ) mostram todos os objetos da área de trabalho
- ▶ A função *rm*(*A*) remove o objeto *A* da área de trabalho
- ▶ *rm(list = ls(all = TRUE))* remove todos os objetos da área de trabalho

# Diretório de trabalho

- ▶ Diretório de trabalho é a pasta onde você irá guardar os arquivos com resultados de suas análises ou de onde irá ler dados de arquivos externos.
- ▶ É possível alterar no menu principal
- ▶ `getwd( )` mostra qual é o diretório de trabalho atual
- ▶ `setwd(" C : Users / CristianoSantos / Desktop")` altera o diretório de trabalho para o endereço dentro da função

# Salvando a Área de Trabalho

- ▶ Para salvar os objetos criados na durante sua sessão de trabalho
- ▶ Vá ao menu principal, clique em arquivo e depois em salvar área de trabalho.

Observe que na linha de comandos aparece a sintaxe do comando executado

```
save.image("C:\\Meus documentos aula1.RData")
```

É possível adicionar este comando no script para que a área de trabalho seja salva automaticamente.

# Salvando o Histórico de Comandos

- ▶ Para salvar os comandos utilizados durante a sua sessão de trabalho
- ▶ Vá ao menu principal, clique em arquivo e depois em salvar histórico.

Observe que na linha de comandos aparece a sintaxe do comando executado

```
save.image("C:\\Meus documentos\\aulaR\\aula1")
```

# Salvando o “Output”

- ▶ Salvar os resultados mostrados no console em um arquivo de texto
- ▶ Vá ao menu principal, clique em arquivo e depois em salvar em arquivo
- ▶ É possível salvar apenas um parte dos resultados que esteja selecionada

# Lista de Exercícios 1

- ▶ Forma de entrega: Mandar por email um arquivo “.txt” ou “.R” com os comandos utilizados na resolução da lista de exercícios.
- ▶ Salvar arquivo com nome Lista1-nomes dos autores-incompleta ou Lista1-nomes dos autores-final.

## Exercícios:

1. Mostre comandos que podem ser usados para criar os objetos e/ou executar as instruções a seguir: (sem digitar diretamente todos os dados)
  - a) o vetor com a sequência de valores (-3, -2, -1, 0, 1, 2, 3);
  - b) o vetor com a sequência de valores (2.4, 3.4, 4.4, 5.4, 6.4, 7.4, 8.4, 9.4, 10.4);
  - c) o vetor com todos os números ímpares de 1 a 80;



4. A função  $sum(x)$  retorna a soma dos elementos do vetor  $x$ . A expressão  $z = rep(x, 10)$  faz o vetor  $z$  igual a uma seqüência de 10 vetores  $x$ . Use estas e outras funções para calcular a soma dos 100 primeiros termos das séries:
- a)  $1 + 1/2 + 1/3 + 1/4 + \dots$
  - b)  $1 + 1/22 + 1/42 + 1/62 + 1/82 + \dots$
  - c)  $1 - 1/2 + 1/3 - 1/4 + 1/5 - 1/6 + \dots$
5. Utilize o comando “subset” para selecionar os pesos e tempos dos frangos que tiveram a dieta 2 no banco de dados “ChickWeight” disponível no R. Utilize novamente o “subset” para selecionar pesos no tempo igual a 2 dos frangos que tiveram a dieta 2 e calcule a média destes pesos.
- Dicas:** consulte o help “subset” da função, rode os exemplos, use o operador == para condição de igualdade e o operador & para checar duas condições ao mesmo tempo.