



Laboratório  
Nacional de  
Computação  
Científica



DEXLAB  
EXTREME DATA LAB



Conselho Nacional de Desenvolvimento  
Científico e Tecnológico

# EAMC 2020

## Desenvolvimento de Web Apps via Shiny R, Relembrando R

Instrutor: Rafael Silva Pereira

Laboratorio Nacional de Computação Científica

17/07/2019

# Linguagem R

- ▶ R é um ambiente computacional e uma linguagem de programação que vem progressivamente se especializando em manipulação, análise e visualização gráfica de dados;
- ▶ Desenvolvida em 1993, e atualmente mantida no repositório Oficial CRAN
- ▶ Baseada na linguagem S , uma linguagem proprietária feita para análise estatística de dados
- ▶ Inicialmente foi construída utilizando-se principalmente C e Fortran, Novas funções a serem submetidas como pacotes podem ser construídas com código compilado.

# Ementa primeira parte

Este curso sera dividido em duas partes:

- ▶ Revisão da linguagem R
- ▶ O pacote Shiny

# Revisão R

Os seguintes tópicos serão discutidos no curso:

- ▶ Tipos de variáveis
- ▶ Declaração
- ▶ Estrutura de Repetição
- ▶ Vetores
- ▶ Matrizes
- ▶ DataFrames
- ▶ Listas
- ▶ Funções
- ▶ Não use loops, família apply
- ▶ Processamento paralelo: Map, Reduce.

# Tipos de variaveis

- ▶ As variaveis instanciadas na linguagem R são dinâmicas, sendo assim não é necessario declarar seu tipo.
- ▶ Novos pacotes podem introduzir novos tipos de variáveis especificos para alguns tipos de dados. Os tipos padrões de dados discutidos aqui serão os seguintes:
  - ▶ Inteiros
  - ▶ Numericos
  - ▶ Caracteres
  - ▶ Fatores
  - ▶ Logicos
- ▶ Alem disto temos estruturas de dados que armazenam estes tipos como:
  - ▶ Vetores
  - ▶ Matrizes
  - ▶ DataFrames
  - ▶ Listas

# Declaração

- ▶ As variáveis em R são dinâmicas, a declaração ocorre quando se instancia um valor a estas, e o tipo é decidido em função do valor associado.
- ▶ Como Exemplo de instanciação de variáveis, iniciemos o primeiro jupyter notebook, chamado Declaração e Operação com variáveis.

# Estrutura de Repetição

- ▶ Agora que sabemos como declarar variáveis, vemos estruturas de repetição
- ▶ Discutimos duas estruturas , For e While
- ▶ Exemplos de como utilizá-las são encontradas no notebook 'Exemplos de Loops em R'

# Vetores

- ▶ Um vetor é uma estrutura de dados que armazena elementos de mesmo tipo.
- ▶ O vetor na linguagem R não tem tamanho pre-definido, sendo dinamicamente alocado à medida que novos elementos são inseridos
- ▶ O índice começa em 1
- ▶ Uma propriedade da linguagem é que as operações são vetorizadas, sendo assim não é necessário a estrutura de loop para atuar neste.
- ▶ Operações com vetor podem ser vistas no notebook 'Vetores Exemplos'



# Matrizes

- ▶ Assim como temos vetores, a linguagem também traz matrizes como estrutura.
- ▶ Matrizes devem armazenar variáveis numéricas em todas as suas opções
- ▶ Matrizes em R têm várias operações de álgebra linear implementadas através de código Fortran otimizado com LAPACK na versão nativa. Exemplos de como operar com matrizes serão discutidas no notebook 'Matrizes Exemplos'

# DataFrames

- ▶ Dataframes são similares a tabelas
- ▶ Tem a estrutura similar a matriz
- ▶ Diferentemente da matriz, cada coluna pode armazenar um tipo de variavel diferente
- ▶ A interpretação de um dataframe e que cada coluna representa uma variavel, enquanto cada linha representa uma observação.
- ▶ Todas as operações vetorizadas podem ser aplicadas sobre esta estrutura
- ▶ Existem varias variações do dataframe dentro da linguagem R, focando em diferentes aspectos para sua otimização
- ▶ Exemplo de como utilizalo serão discutidos no notebook 'Utilizando DataFrames e algumas variações'

# Listas

- ▶ Listas em R são uma estrutura generica para organização de seus dados
- ▶ Uma propriedade das listas e que esta funciona como um vetor que pode armazenar qualquer tipo de variavel ou estrutura de dados em um de seus indices
- ▶ Isto é valido inclusive para armazenar outras listas
- ▶ Estruturas não contidas na linguagem base tambem podem ser armazenadas em listas.

# Funções

- ▶ Funções em R são definidas sem um tipo específico de retorno
- ▶ Devido a isto, dado que uma determinada função possa ser definida para mais de um tipo de estrutura de dados, esta funcionaria para todos os tipos
- ▶ Como exemplo dado uma função que pede  $a$  e  $b$  e retorna  $a+b$ , esta funcionaria se  $a$  e  $b$  fossem escalares, vetores de mesmo tamanho, matrizes de mesma dimensão, ou qualquer outra combinação em que a definição de soma matemática fosse definida.
- ▶ Se queremos retornar mais de um objeto em uma função podemos utilizar uma lista como retorno
- ▶ Exemplos de como trabalhar com funções serão discutidas no notebook 'Funções em R'

## Familia apply

- ▶ Como já demonstrado as operações em R são nativamente vetorizadas.
- ▶ Devido a isto, a utilização de estruturas de loop não escala eficientemente na linguagem.
- ▶ Para solucionar isto, se utiliza a família de funções apply, que aplicam o loop de forma escalável e eficiente.
- ▶ Domínio da família apply permite trabalhar com grandes bases de dados em R, uma vez que a família apply permite código rodar em paralelo utilizando vários núcleos em CPU, utilizando os vários processadores de uma GPU, ou até mesmo distribuindo os dados em várias máquinas.
- ▶ Uma discussão de como utilizar-se a família de funções apply está em 'Loops otimizados: apply

# Map Reduce

- ▶ A função map opera de forma semelhante a lapply, se torna mais comum utilizando-se do tidyverse, aonde se torna mais intuitiva sua sintaxe.
- ▶ Já a função Reduce tem como propriedade receber uma lista e uma função, que é aplicada recursivamente sobre a lista, de forma a retornar um valor ou estrutura.
- ▶ Estas se tornam interessantes em processamento paralelo, seja utilizando-se de varias cores com a biblioteca parallel/gpuR ou utilizando-se de computação distribuida com MPI.

# Acknowledgments



- ▶ Assim terminamos a primeira parte do curso
  - ▶ Segunda Parte: Quinta feira
  - ▶ Contato: [rpereira@lncc.br](mailto:rpereira@lncc.br)
- 
- ▶ Duvidas?