

Análises Multivariada de dados e de Séries Temporais

Tema 03c: Introdução ao software R

Prof. Dr. Juan A. Alcántara Núñez

Introdução ao mundo de R:

1) Introdução

Vantagens e inconvenientes
A importância de ser ordenado
O diretório de Trabalho

2) Infraestrutura

Ajuda em R
script e área de trabalho
Librarias e pacotes de R

3) Importar e Exportar

Arquivos CSV

4) Tipos e Estrutura de dados

Numérico , caráter, ...
vetores, matrizes e dataframe

5) Análise de dados

Análise de Regressão simples

6) Aplicações

Análise de Regressão múltipla

Introdução ao mundo de R:

- ✓ R é uma linguagem de programação voltada para a análise de dados e elaboração de gráficos.
- ✓ A interação com R se produz através de uma interface de linha de comandos.
- ✓ É software livre e pode ser baixado de <http://www.r-project.org>

Vantagens:

- ✓ Software livre (gratuito) e pode ser modificado a vontade
- ✓ Existe uma comunidade grande de voluntários trabalhando para melhorá-lo
- ✓ Atualizado com frequência, última versão R-3.3.0 (3/05/2016)
- ✓ Funciona em Windows, Mac OS, Unix, Linux,
- ✓ Potente e versátil
- ✓ Permite elaborar gráficos de muita qualidade
- ✓ Trabalha bem com Latex, o que permite obter output de grande qualidade estética
- ✓ Está-se tornando o padrão na comunidade científica: desenvolvimento em análise são convertidos em pacotes de R

Introdução ao mundo de R:

Inconvenientes:

- ✓ O step inicial tem uma elevada altura e a sua sintaxe é relativamente exigente
- ✓ A documentação, mesmo que diversa e ampla, pode resultar complicada
- ✓ Nos erros, R não dá muitas pistas onde pode estar errando
- ✓ Alguns pacotes já foram bem contrastados, no entanto outros não
- ✓ Utiliza a linha de comandos e não uma interface gráfica

A importância de ser ordenado:

R gera uma grande quantidade de output e é sumamente recomendável usar um diretório para cada projeto de análise. Sendo a estrutura proposta dentro do diretório:

1. Dados: matéria prima
2. Scripts: códigos de R a utilizar
3. Imagens: fotografias do que está ativo em R
4. Historias: conjunto de comandos executados
5. Gráficos:
6. Anotações: para saber que é feito ou será feito no projeto

Introdução ao mundo de R:

Acesso a R

- ✓ Em Windows: dobre clique no ícone
- ✓ Ir ao sistema de menu

Diretório de Trabalho

Sempre que se entra em R é recomendado ver qual o diretório de trabalho, qual o diretório onde estão os dados e onde serão guardado os resultados. Existem duas formas de fazer isso:

Usando linha de comandos:

> `getwd()` => visualiza o diretório de trabalho

> `setwd("C:/User/SW/Documents/Esritorio")`
=> estabelece o novo diretório de trabalho

Usando o menu: clicar diretamente na aba File e procurar mudança de diretório (lembrar que esta em inglês)

Infraestrutura de R:

Ajuda em R

- > `help()` => abre o explorador (mozilla) e apresenta a documentação de ajuda. Pode ser especificado como (mean), (princomp) procura funções.
 >`help(mean)` equiv. >`?mean`
- > `help.search("conceito")` => Busca ajuda sobre um conceito, i.e. ("regression") mostra a ajuda buscando a palavra regression
 >`help("regression")` equiv. >`??regression`
- > `help.start()` => busca ajuda diretamente do site, mostra a documentação on-line separada por tópicos

Scripts

- ✓ É a forma mais confortável de interatuar com R, assim podemos guardas registro de todo o que fazemos e pode ser utilizado depois
- ✓ Facilitam a colaboração entre investigadores
- ✓ Para editar um script pode ser utilizado qualquer editor de texto

Infraestrutura de R:

Como trabalhar com Scripts

- ✓ Mediante copiar-colar
- ✓ Mediante o comando `source()`
- ✓ É importante ter comentários, os comandos iniciam com `#`

Área de Trabalho (Workspace)

Todos os objetos criados em R são armazenados na área de trabalho ou Workspace e podemos ter acesso a eles

- > `save.image("arquivo")` => Podemos guardar o workspace com o nome arquivo arquivo
- > `load("arquivo")` => ler (recuperar) um conjunto de objetos guardados no arquivo
- > `ls()` => listar todas os objetos criadas numa sessão (workspace)
- > `rm(nomeobjeto)` => apagar objetos do workspace

Infraestrutura de R:

Arquivo histórico

Todos os comandos que são executados em R (também os erros) são guardados no arquivo histórico. O Arquivo histórico é muito útil para criar os script

Podemos deslizar entre os comandos anteriores usando as seta para acima e para abaixo

> `savehistory(file="arquivo")` => pode guardar um arquivo histórico

> `loadhistory(file="arquivo")` => lê (recupera) um arquivo histórico

Paquetes em R

- ✓ Os que estão acessíveis de forma imediata. Estão instalado no modulo base. Ex. `mean()`, `var()`, `cor()`
- ✓ Os que estão instalados mais não imediatamente acessíveis. Temos que chamar ao pacote que as contem. Ex. `mvrnorm` (pacote MASS)
- ✓ Os que não estão instalados, mas estão disponíveis nos repositórios. Precisamos instalar e depois chamar. Ex. PCA (pacote FactoMineR)

Infraestrutura de R:

Paquetes em R

- ✓ Os que estão acessíveis de forma imediata. Estão instalado no modulo base. Ex. `mean()`, `var()`, `cor()`
- ✓ Os que estão instalados mais não imediatamente acessíveis. Temos que chamar ao pacote que as contem. Ex. `mvrnorm` (pacote MASS)
- ✓ Os que não estão instalados, mas estão disponíveis nos repositórios. Precisamos instalar e depois chamar. Ex. `PCA` (pacote FactoMineR)

Importar e Exportar dados em R:

Arquivos CSV (comma-separated values)

- ✓ São arquivos planos (filas por coluna) sem estrutura adicional
- ✓ As colunas estão separadas por vírgulas (ou outros separadores) e suas filas por saltos de linha
- ✓ Podem incluir nomes para as filas (observações) e para as colunas (variáveis)

Vantagens

- ✓ A diferença de outros arquivos (XLS,XLSX) , são standards
- ✓ Podem ser lidos com qualquer editor de texto

Antes de importar comprovar que

- ✓ As filas e as colunas sejam consistentes (parte de filas ou colunas em vazias)
- ✓ Não existam símbolos esquisitos, %, \$, €, __, -
- ✓ Dados perdidos.

> `read.csv(file="arquivo")` => assume que a primeira fila são os nomes

Importar e Exportar dados em R:

Exemplo CSV (carateristicas.csv)

```
altura peso idade
175 87 32
187 82 36
176 56 23
165 68 39
```

Comando a utilizar

```
> dados<-read.csv(file="carateristica.csv") => verifique o diretório
```

Exemplo CSV (carateristicas1.csv)

```
altura peso idade
"Pedro" 175 87 32
"Maria" 187 82 36
"Luis" 176 56 23
"Jose" 165 68 39
```

Comando a utilizar

```
> dados<-read.csv(file="carateristica1.csv") => verifique o diretório
```

Importar e Exportar dados em R:

Exemplo CSV (carateristicas2.csv)

```
175 87 32
187 82 36
176 56 23
165 68 39
```

Comando a utilizar

```
> dados<-read.csv(file="carateristica2.csv", header=FALSE)
=> verifique o diretório
```

Para exportar CSV (carateristicas2.csv)

Comando a utilizar

```
> write.csv(file="nome") => verifique o diretório
```

Tipo e estrutura de dados em R:

Todo em R são objetos e todos os objetos tem uma classe ou pertencem a uma classe (programação orientada a objetos)

- Classe numeric e classe character

A função básica para trabalhar com classes é a função `class()`, Comando a utilizar

```
> x<-2           => dou um valor para x
> x
[1] 2
> class(x)        => pergunta o tipo de classe de x
[1] "numeric"      => me indica que x pertence a classe numeric
```

Classes de dados em R

- ✓ Números (numeric)
- ✓ Cadeias (character)
- ✓ Fatores (factor)
- ✓ Valores lógicos (logical)

Tipo e estrutura de dados em R:

Funções para classes

- > `class()` => pergunta que tipo de classe tem?
- > `is.className()` => pergunta se o objeto pertence a uma classe dada?
- > `as.className()` => transforma a variável para uma classe dada.

Classe numeric

Os números em R podem ser inteiros (integer) o reais (real). Podem complexos (complex). Por tanto podemos usar as funções vistas anteriormente

- ✓ Soma: +
- ✓ Subtração: -
- ✓ Produto: *
- ✓ Divisão: /
- ✓ Potencia: ^
- ✓ Comparação: > , <= , >= , == , != (lembrar como se usa em programação)

Tipo e estrutura de dados em R:

Classe character

- > `as.character()` => transforma a variável para uma classe dada
- > `paste(a,b,sep=" ")` => junta cadeia de caracteres
- > `substr(string, começo, comprimento)` => extrai cadeias

Classe factor

São parecidos a classe caráter, mas guarda registro dos seus valores (são pucos). Útil para variáveis qualitativas (estado civil, sexo, etc)

- > `factor()` => transforma a variável para uma classe factor
- > `is.factor()` => pergunta se a variável é do tipo factor
- > `levels()` => mostra o conjunto de todos os possíveis valores do factor
- > `table()` => mostra a frequência absolutas de cada um dos levels

Classe logical

- > `is.logical()` => pergunta se a variável é do tipo logical
- > `& | !` => conectivas lógicas

Tipo e estrutura de dados em R:

- ✓ **Vetores (vector):** conjunto unidimensional de objetos atômicos (são porque não podem ser descompostos) em outros. Podem conter números , caracteres, fatores o valores lógicos. Todos os elementos devem ser da mesma classe. Se usam [] para identificar cada elemento
- ✓ **Matrizes (matrix):** São vetores com atributos adicionais chamados de dimensoes. Sao vetores com mais de uma dimensão. Se usa [i,j] , filas [i,] ou colunas [,j]
- ✓ **Marcos (dataframe):** Sao as estruturas de dados mais adequadas para entender a tabela de dados. Pode ter diferentes classes de dados. podemos assignar nomes as filas ou colunas de um dataframe