Noções Básicas sobre R e R Commander

Prof.Dr. Paulo Sergio Panse Silveira
Professor Associado do Departamento de Patologia Faculdade de Medicina USP

Definições

R é uma linguagem de programação com múltiplos propósitos, mas forte em procedimentos estatísticos. Para usá-la você precisa, portanto, aprender a programar. Um bom ponto de partida para quem quiser aprender a linguagem é a documentação oficial do projeto, disponível em http://cran.r-project.org/doc/manuals/R-intro.html

R Commander (abreviado para RCmdr) é uma interface gráfica (GUI) que facilita o principiante no uso das funções estatísticas disponíveis em R. Não tem, nem de longe, toda a capacidade do R, mas tem algumas vantagens:

- o básico está disponível,
- é um projeto de código aberto, que vem recebendo contribuições e incorporando capacidades,
- ao executar alguma coisa em RCmdr a sintaxe R é exibida, permitindo ao principiante começar a aprender R,
- para quem sabe R, RCmdr permite a entrada de comandos diretamente (ou modificar os comandos que o RCmdr adiantou, para executar variações)

Tutoriais e documentação

Para os interessados em aprender R / RCmdr:

- Contributed Documentation (no próprio site do R-project) :

  http://cran.r-project.org/other-docs.html

Interfaces e conexões

- RKWard, GUI alternativa para o RCmdr: sourceforge.net/apps/mediawiki/rkward
- Extensão para integrar recursos do R no OpenOffice: https://wiki.openoffice.org/wiki/R_and_Calc

O que é o R Commander?

Um resumo rápido está em http://www.rcommander.com/

Instalando em seu computador

O R Commander, comumente abreviado para ***Rcmdr***, é uma interface gráfica para [R]. Ambos precisam ser instalados para que você trabalhe. O uso é livre e o software está disponível para vários sistemas operacionais.

Consulte os seguintes *sites*:

- The R Project for Statistical Computing para encontrar as instruções de instalação do R.

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

- R Commander Installation Notes para instalar o Rcmdr.

  http://socserv.mcmaster.ca/jfox/Misc/Rcmdr/installation-notes.html

Introdução ao R Comander

O que você deve saber antes de usar o R Commander

Para instalar o R Commander, seu computador também precisa do R instalado.

Como o R Commander é disponível para diversas plataformas, presume-se que o estudante saiba utilizar seu computador. É necessário saber como baixar um arquivo da Internet (*download*), criar uma pasta (diretório) em seu computador, e encontrá-lo posteriormente.

Na área de trabalho do Windows encontre e utilize o acesso ao R, que abre a RGui, uma interface gráfica que facilita o uso do R e contém um terminal.

Neste terminal digite o comando:

```
> library(Rcmdr)
```

A janela do R Commander deve abrir.

Entrada de dados: abrindo planilha de dados

Os dados que utilizaremos aqui estão no arquivo exames1.xls, que você deve baixar para seu computador (dependendo da configuração de seu computador e navegador, experimente usar o botão da direita do mouse para fazer este *download*). Este arquivo de dados precisa ser importado pelo Rcmdr.

Acesse, no Rcmdr:

 Dados -> Importar conjunto de dados -> de arquivo de Excel...

 Data -> Import data -> from Excel...

A caixa de diálogo que aparece.

Defina o nome do conjunto de dados (**Enter name for data set**): Exames1 (apenas para que lembremos de qual arquivo os dados vieram).

Procure o arquivo Excel exames1.xls, que você gravou em seu computador e abra este arquivo.

Visualizando os dados

Após importar os dados, um conjunto de dados (*data set*) com o nome que você escolheu está ativo. Você pode conferir se a importação dos dados deu certo clicando em **Ver conjunto de dados** (**[View data set]**) para abrir nova janela.

Nesta tela aparecem os dados importados. Há também barras de rolagens horizontal e vertical, para que você possa percorrê-los. Antes de prosseguir, identifique os elementos que compõem a janela em questão, pois é fundamental entender como os dados ficam organizados:

- note que os dados de cada paciente estão organizados em linhas, identificadas na coluna à esquerda enumeradas a partir de 1 (o número de cada linha coincide com o número de identificação do paciente, mas isto é somente coincidência);
- observe que cada coluna, denominada variável, é encimada por sua identificação, o nome da variável. É através deste identificador, importado da primeira linha do arquivo texto, que você referenciará as variáveis que desejar para executar os procedimentos adiante.

  Dica: não utilize espaços ou símbolos especiais (o Rcmdr aceita "." se precisar de nomes compostos) para compor nome de identificadores; também recomenda-se iniciar os nomes das variáveis com uma letra;

Depois que verificar o sucesso da importação, sugere-se que esta janela seja fechada.

Ao lado do botão **[Ver conjunto de dados]** (**[View data set]**) aparece **[Editar conjunto de dados]** (**[Edit data set]**), o qual obviamente permite modificar os dados. A interface de edição não é muito boa. Caso precise de alterações mais extensas, é melhor arrumar os dados em uma planilha e importar novamente.

Salvando e recuperando dados

Este conjunto de dados pode ser salvo no formato do R Commander para usar mais tarde (assim não é necessário repetir a importação). Esta operação, bem como várias outras, encontra-se em:

 Dados -> conjuntos de dados ativos -> salvar conjunto de dados ativos...

 Data->Active data set->Save active data set...

Escolha a pasta e grave o arquivo que recebe o nome do *data set* e a extensão Rdata: no exemplo o arquivo gerado é **Exames1.RData**. Por esta razão, procure escolher nomes com significado para seus conjuntos de dados; não os chame de **dados**, **planilha**, **pacientes**, ou coisas similares. Quando quiser voltar a trabalhar com o mesmo conjunto de dados, busque-o com:

 Dados -> Carregar conjunto de dados

 Data->Load data set...

Elementos da tela do Rcmdr

A esta altura, a janela do R Commander deve ter vários comandos transcritos.

Identifique os elementos desta tela: menus de acesso aos comandos no topo, uma fileira de botões que indicam o *data se* ativo, permitem edição ou visualização dos dados, e uma indicação [<Sem modelo ativo>] ([<No active model>]). Abaixo aparecem três janelas: script (*Script Window*), Output (*Output Window*) e Mensagens (*Messages*). Respectivamente mostram os comandos que você teria que executar se usasse um terminal (usando o R sem a interface gráfica do Rcmdr), o que este terminal ecoaria, e mensagens do sistema.

Entrada de dados: preenchimento manual da planilha

Vamos abrir um novo conjunto de dados usando o comando

 Dados ->Novo conjunto de dados...

 Data -> New data set...

Como exercício, batize este conjunto com o nome de **Colegas**, pois armazenaremos dados de nomes de seus colegas, idade e cidade de nascimento. Portanto será uma planilha com três variáveis.

A interface é auto explicativa. Se necessário, alargue o tamanho da tela para aparecer, pelo menos, três colunas.

Renomeie as três primeiras colunas, respectivamente, como **Nome**, **Idade** e **Cidade Nasc**: clique sobre os nomes pré-definidos (**var1**, **var2** ou **var3**), escolha **nome da variável** (**Change name**) e entre com os nomes de variáveis desejados. A seguir, entre com o seguinte pequeno conjunto de dados:

```
Pedro Souza            17      São Paulo
Marines Sanchez        19      Bauru
Clóvis Concravo        21      Votuporanga
```

Clique **[X]** para s**air**. O conjunto de dados **Colegas** é criado já fica ativo. Pode conferir com **[Ver conjunto de dados]** (**[View Data Set]**).

Note que o Rcmdr não admite espaço no nome das variáveis, de forma que a terceira coluna ficou denominada como **Cidade.Nasc**. Note também que **[Ver conjunto de dados]**(**[View Data Set]**) ajusta automaticamente a largura das colunas ao conteúdo das células, facilitando a leitura.

Saindo e retornando ao R Commander

Desligando o R Commander

Saber sair corretamente do programa é importante. O mais normal é fechar a interface gráfica e o terminal, utilizando:

 Arquivo ->Sair ->Do Commander e do R

 File -> Exit -> From R Commander and R

O programa perguntará se quer salvar o *scrip* e o *output*. Por enquanto responda **[Não]** (**[No]**) para ambos; voltaremos a isto adiante.

Voltando ao R Commander

Nas seções anteriores você abriu dois *data sets* e os salvou. O primeiro, **Exames1**, foi obtido de um arquivo texto que geramos a partir de uma planilha do Excel; o segundo,**Colegas**, foi

digitado por você. Respectivamente, devem ter sido salvos no formato do Rcmdr com os nomes **Exames1.Rdata** e **Colegas.Rdata**.

Reabra, portanto, o Rcmdr e utilize:

 Dados -> Carregar conjunto de dados...

 Data -> Load data set

aponte para leitura o arquivo **Exames1.Rdata**. A seguir, repita o procedimento para ler **Colegas.Rdata**. Repare a *Script Window* onde aparece

```
load("{aqui aparece o caminho para seu arquivo}/Exames1.RData")
load("{aqui aparece o caminho para seu arquivo}/Colegas.RData")
```

Salvando e lendo *scripts*

Este é um uso mais avançado e este exemplo é muito simples (mas ilustrativo).

Pois bem... utilize:

 Arquivo -> salvar script como ...

 File -> Save script as...

e dê o nome **Leitura_Exames_e_Colegas.R** para este script.

Feche completamente o programa (**Sair -> Do Commander e do R**, **Exit -> From Commander and R**) e abra novamente. Utilize

 Arquivo -> Abrir arquivo c/ script

 File -> Open script file...

e traga de volta o conteúdo da Janela script (*Script Window*) recuperando o arquivo **Leitura_Exames_e_Colegas.R**. Repare que esta janela permite edição, de forma que você pode conferir e ajeitar o script sempre que quiser. Quanto estiver pronto, note o botão **Submeter** (**Submit**) no canto inferior direito desta janela: clique nele para executar seu script. A janela mensagens (*Messages*) mostra:

```
ERRO: Nada foi selecionado.
```

```
ERROR: Nothing is selected.
```

Selecione, então, na Janela Script (*Script Window*) as duas linhas:

```
load("{aqui aparece o caminho para seu arquivo}/Exames1.RData")
```

```
load("{aqui aparece o caminho para seu arquivo}/Colegas.RData")
```

e clique em **Submeter** (**Submit**) novamente. Desta vez a Janela output (*Output Window*) reage, mostrando

```
> load("{aqui aparece o caminho para seu arquivo}//Exames1.RData")
> load("{aqui aparece o caminho para seu arquivo}//Colegas.RData")
```

Clique o botão da fileira superior onde aparece <Não há conjunto de dados ativos> (<No active dataset>) e escolha um dos dois *data sets* lidos. A janela Mensagens (*Messages*) acusará o sucesso dizendo:

 NOTA: Os dados Exames1 tem 705 linhas e 15 colunas.

 NOTE: The dataset Exames1 has 705 rows and 15 columns.

ou

 NOTA: Os dados Colegas tem 3 linhas e 3 colunas.

 NOTE: The dataset Colegas has 3 rows and 3 columns.

de acordo com o conjunto de dados que você deixou ativo.

A partir deste exemplo simples, sofistique. Qualquer script que você salvar, reler e submeter repetirá uma sequência de ações. Com o que aprender adiante, poderá trabalhar no R Commander experimentando fazer alguma análise. Apague as linhas que não deram certo. faça coisas algo mais sofisticadas, como gráficos e análises estatísticas, teste com determinado conjunto de dados e, depois, adapte o script para repetir o trabalho em outros conjuntos de dados. Salve seus scripts "amadurecidos" e evitará um monte de (re)trabalho.

Similar, mas menos útil (pois não roda como um *script*), caso queira preservar um lembrete do que fez enquanto usava o Rcmdr, é salvar a Janela Output(*Output Window*) em um arquivo texto.

Manipulação da Planílha

Há uma quantidade de operações disponíveis nos menus do R Commander. Aqui ilustraremos algumas, que lhe permitirão:

- criar subsets
- fazer gráficos

Obtendo subconjunto dos dados

Importante: Antes de começar o exercício proposto aqui, observe que na versão xls, da qual os dados de **Exames1** vieram, a coluna de sódio sérico tem o nome de variável **NA**. Embora a importação ocorra sem problemas, o nome da variável entra em conflito com a palavra reservada do Rcmdr, **NA**, significando ***N****ot* ***A****vailable*(confira com **[View data set]**: aparece um pequeno ponto ao lado de NA). É possível consertar o arquivo texto e reimportar os dados, mas a solução mais simples e rápida, agora que já importou os dados, é entrar no modo de preenchimento manual da planilha, alterar o nome da variavel (por exemplo, para **SODIO**) e salvar seu *data se* novamente, para preservar a modificação.
Outra maneira de fazer a mesma coisa é usar **Dados -> modificação de variáveis no conjunto de dados -> Renomear variáveis...** (**Data -> Manage variables in active data set -> Rename variables...**).

Uma vez consertado e ativo seu *data set* **Exames1**, utilize:

 Dados -> Conjunto de dados ativos -> definir subconjunto de dados ativos...

 Data -> Active data set -> Subset active data set...

Observe a janela que se abre.

É possível "recortar" colunas ou linhas do *data set*.

Escolhendo colunas

Suponha que você queira apenas algumas variáveis em um novo *data set*, em um estudo que somente lhe interessa estudar colesterol (HDL e COL) e triglicérides (TRIG); para não perder a referência dos dados do paciente guardaremos também a identificação do PACIENTE, IDADE, SEXO. Basta desmarcar "Incluir todas as variáveis" ("Include all variables") e, segurando a tecla CTRL, apontar as variáveis de interesse, que aparecem em ordem alfabética: COL, HDL, IDADE, PACIENTE, SEXO e TRIG. Deixe **Expressão**(**Subset expression**) como está, <todos os casos> (<all cases>) e escolha um nome para o novo *data set* (e.g. **Gordura**).

Clique [OK]. Na **Janela de Script** (**Script Window**) aparece o comando:

```
Gordura <- subset(Exames1, select=c(COL,HDL,IDADE,PACIENTE,SEXO,TRIG))
```

Na **Output Window** ecoa-se a execução do comando R:

```
> Gordura <- subset(Exames1,
select=c(COL,HDL,IDADE,PACIENTE,SEXO,TRIG))
```

Na **janela mensagens** (**Message Window**) o sistema notifica:

NOTA: os dados Gordura tem 705 linhas e 6 colunas.

NOTE: The dataset Gordura has 705 rows and 6 columns.

o que confere, pois todos os pacientes foram transferidos (mesmo número de linhas do conjunto original), mas com apenas 6 colunas. Repare que o conjunto de dados ativo agora é **Gordura**; como sempre, você pode conferir com **[Ver conjunto de dados]** (**[View data set]**) e salvar seu conjunto de dados em um arquivo **Gordura.Rdata** para uso futuro.

Escolhendo linhas

Variáveis numéricas

Torne o *data set* **Exames1** novamente ativo e repita o processo para fazer um Subconjunto (*subset*). Podemos, por exemplo, separar os pacientes com hipernatremia (sódio sérico acima de 142 mEq/L). Desta vez vamos deixar assinalado **Incluir todas as variáveis** (**Include all variables**), portanto não há sentido em apontar nada no quadro de variáveis, e colocaremos em **Expressão** (**Subset expression**) a expressão **SODIO > 142**. Como sempre, escolheremos um nome para o novo *data set* que seja significativo (e.g. Hipernatremicos).

Ao clicar [OK] acontecerá algo imprevisto: uma mensagem de alerta (em verde) na **Janela de mensagem** (**Message Window**):

```
NOTA: Aviso em Ops.factor(SODIO, 142): > não faz sentido para fatores
```

```
WARNING: Warning in Ops.factor(SODIO, 142): > not meaningful for
factors
```

Por quê? Esta coluna de **SODIO**, que já tínhamos renomeado, ainda tem problemas. Neste caso, volte a ativar o conjunto de dados Exames1, entre no editor de dados clicando**[Editar conjunto de dados]** (**[Edit data set]**) e clique sobre o nome da variável **SODIO**. Está marcado Character (*character*)

compare com a coluna do potássio sérico. Está marcado Numérico (numeric).

SODIO, por alguma razão, está sendo tratada pelo Rcmdr como texto, enquanto **K** é numérica. Para corrigir, basta indicar que **SODIO**é *Real*, sair do editor e repetir a geração do *sub set* chamado **Hipernatremicos**. Desta vez, na Mensagens (*Message Window*), aparece:

```
NOTA: Os dados Hipernatremicos tem 25 linhas e 15 colunas.
```

```
NOTE: The dataset Hipernatremicos has 25 rows and 15 columns.
```

Podemos isolar uma "fatia" dos pacientes, por exemplo, os normonatrêmicos (com natremia igual ou superior a 120 e igual ou inferior a 142 mEq/L). Torne **Exames1** ativo e repita o procedimento, desta vez usando **???** (**Subset expression**) como **SODIO >= 120 & SODIO <= 142** (onde **&** é a sintaxe R para **AND**). A resposta deve ser:

```
NOTA: OS dados Normonatremicos tem 678 linhas e 15 colunas.
```

```
NOTE: The dataset Normonatremicos has 678 rows and 15 columns.
```

Note que $678+25=703$. Os outros dois pacientes devem ser os Hiponatrêmicos. Experimente criar um *subset* usando a **Expressão** (**Subset expression**) **SODIO < 120**. Obterá:

```
NOTA: Os dados Hiponatremicos tem 1 linhas e 15 colunas.
```

```
NOTE: The dataset Hiponatremicos has 1 rows and 15 columns.
```

Que tal verificar as demais colunas, se estão registradas como numéricas e textuais como deviam ser?

Variáveis textuais

Dados textuais (i.e., *character*) também podem ser utilizados para separar um *subset*. Suponha que eu queira isolar as mulheres em um *subset* **Exames1_mulheres**. É o mesmo procedimento (lembre-se de tornar Exames1 novamente ativo), utilizando a **Expressão** (**Subset expression**) como **SEXO == "F"** (note o uso de dois sinais de igual, sintaxe do R para comparações lógicas pois o sinal de igual simples serve para atribuições, e F entre aspas porque é uma *string*), obtendo.

```
NOTA: Os dados Exames1_mulheres tem 448 linhas e 15 colunas.
```

```
NOTE: The dataset Exames1_mulheres has 448 rows and 15 columns.
```

Variáveis textuais

Dados textuais (i.e., *character*) também podem ser utilizados para separar um *subset*. Suponha que eu queira isolar as mulheres em um *subset* **Exames1_mulheres**. É o mesmo procedimento (lembre-se de tornar Exames1 novamente ativo), utilizando a **Expressão** (**Subset expression**) como **SEXO == "F"** (note o uso de dois sinais de igual, sintaxe do R para comparações lógicas pois o sinal de igual simples serve para atribuições, e F entre aspas porque é uma *string*), obtendo.

```
NOTA: Os dados Exames1_mulheres tem 448 linhas e 15 colunas.
```

```
NOTE: The dataset Exames1_mulheres has 448 rows and 15 columns.
```

Mais exemplos?

Enquanto executava as operações acima, você prestou atenção à **Script** (**Script Window**)? Se sim, deve ter visto os comandos:

```
Gordura <- subset(Exames1, select=c(COL,HDL,IDADE,PACIENTE,SEXO,TRIG))
Hipernatremicos <- subset(Exames1, subset=SODIO > 142)
Normonatremicos <- subset(Exames1, subset=SODIO >= 120 & SODIO <= 142)
Hiponatremicos <- subset(Exames1, subset=SODIO < 120)
Exames1_mulheres <- subset(Exames1, subset=SEXO == "F")
```

O que aparece após **subset=** são expressões R válidas. Em qualquer site que dê exemplos encontrará outros operadores (como **|** que significa **or**). Poderá, com isto, descobrir como construir expressões mais complexas para gerar o *subset* que precisar.

Veja, por exemplo, http://rprogramming.net/subset-data-in-r/

Uma pequena lista dos operadores básicos, úteis para construir as expressões, é

Operadores para expressões R

==	igual
!=	diferente
&	AND
|	OR
is.na(nome_da_variável)	inclui os valores não disponíveis (NA) da variável
!is.na(nome_da_variável)	exclui os valores não disponíveis (NA) da variável
>	maior que
<	menor que
>=	maior ou igual que
<=	menor ou igual que

Parênteses também podem ser usados para a precedência de expressões. Por exemplo (absurdo, mas somente para exercitar), se quisermos um *subset* das mulheres hipernatrêmicas e dos homens normonatrêmicos, poderíamos usar a expressão

(SODIO > 142 & SEXO == "F") | ((SODIO >= 120 & SODIO <=142) & SEXO == "M")

garantindo, desta forma, que todos os **AND**e **OR** aconteçam na ordem desejada. Se o executar obterá:

```
Exemplo_complicado <- subset(Exames1, subset=(SODIO > 142 & SEXO == "F") | ((SODIO >= 120 & SODIO <=142) & SEXO == "M"))
```

e

```
NOTA: Os dados Exemplo_complicado tem 257 linhas e 15 colunas.
```

```
NOTE: The dataset Exemplo_complicado has 257 rows and 15 columns.
```

Use **Ver conjunto de dados** (**[View data set]**) e confira; procure as poucas mulheres filtradas e verá que são as hipernatrêmicas.

Recodificando variáveis

Com

 Dados -> Modificação de variáveis de conjunto de dados ativo -> Recodificar variáveis...

 Data -> Manage variables in active data set -> Recode variables...

você pode criar outras variáveis. Por exemplo, caso queiramos classificar as pessoas de acordo com suas idades, experimente:

1. selecionar a variável **IDADE**
2. escolher o nome da nova variável, e.g. **IDADE_classificacao**
3. preencher **Definições p/ recodificação** (**Enter recode directives**) com um critério de sua escolha, e.g.:

1:11 = "criança"

12:17 = "adolescente"

18:25 = "adulto jovem"

26:64 = "adulto"

65:1000 = "idoso"

Observe que nova coluna foi adicionada ao seu *data set*.

Existem outras formas de definir as diretivas --- consulte **[Ajuda]** (**[Help]**) antes de clicar **[OK]**.

Experimente "esquecer" um pedaço das idades, por exemplo, parando os adultos aos 55 anos de idade, e observe como ficam os conteúdos da variável criada.

Para saber mais

Há muitos tutoriais e dicas de uso na *Web*. Quem encontrar e testar outros, pode ampliar a lista colocando um breve comentário sobre sua experiência.

Para iniciantes

- *Getting Started with R Commander*

  Trata-se de um mini-manual de 27 páginas disponível em PDF, escrito por um dos autores do Rcmdr. É indicado para quem vai usar o R Commander pela primeira vez ou precisa de uma recordação rápida sobre seu uso. Começa do início, explicando todos os elementos da interface do R Commander, e segue fazendo algumas manipulações básicas dos dados.

Referências

1. David McCallum. *R Commander HowTo*
2. Fox J. Getting Started with R Commander