PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO
Departamento de Engenharia Industrial
Rua Marquês de São Vicente, 225
22453-900 – Rio de Janeiro
Brasil

ENG1536 – Inferência Estatística

Laboratório – Guia de Estudo 4

O tema deste estudo são bases de dados em R e importação de dados externos. Você já aprendeu diversos elementos úteis do R, incluindo a construção de vetores, aritmética vetorial, distribuições de probabilidade e gráficos. No entanto, todos esses recursos para análise de dados têm pouca utilidade se não formos capazes de criar na área de trabalho do R bases de dados reais, as quais geralmente serão importadas de bases externas.

Em R, bases de dados são geralmente construídas num objeto da classe **moldura de dados** (*data frame*). Tecnicamente, as molduras de dados são uma classe específica de **lista**, então vale a pena conhecer primeiro esse tipo de objeto. Digite:

```
aluno <- list(nome="João",idade=20,CR=7.5,estrangeiro=FALSE)
```

A função `list` acima funciona de maneira muito parecida com a função `c` já conhecida por nós: ela combina os seus argumentos numa coleção ordenada de objetos. A diferença é que no caso do vetor os componentes devem ser todos do mesmo tipo, enquanto na lista os componentes podem ser de tipos diferentes. No exemplo acima, a lista criada é guardada num objeto chamado `aluno` e ela contém quatro componentes: um texto (“João”), dois números (20 e 7,5) e um valor lógico (FALSE). Além disso, os componentes numa lista geralmente são nomeados. Isso serve para facilitar o acesso a componentes específicos da lista. Por exemplo, o CR do aluno pode ser pescado da lista de três maneiras, observe as diferenças:

```
aluno[3]; aluno[[3]]; aluno$CR
```

Na última maneira de busca, não precisamos lembrar que o CR do aluno está na terceira posição da lista, basta acrescentar ao nome da lista o nome do componente, precedido por cifrão (`$CR`). Qual a diferença entre a primeira e a segunda maneira de busca?

Voltando agora às molduras de dados: é possível pensar nelas como um empilhamento de listas similares, cujos componentes têm o mesmo nome e tipo, mas conteúdo diferente. Por exemplo, uma moldura chamada `turma` poderia ter na sua primeira linha o nome de um primeiro aluno, a sua idade, o seu CR e o sinalizador lógico de estrangeiros; na segunda linha da moldura, as mesmas informações, na mesma ordem, de outro aluno, e assim em diante. Se você já trabalhou com planilhas, pode ser mais fácil pensar na moldura de dados usando essa analogia. Cada linha da planilha contém diferentes informações sobre um item na base de dados (no exemplo anterior, os dados sobre um aluno). Cada coluna da planilha contém a mesma informação para todos os itens na base de dados (por exemplo, todos os nomes dos alunos na turma). Além disso, cada coluna normalmente tem um nome descritivo.

A instalação padrão do R vem com diversos exemplos de molduras de dados em objetos ocultos na área de trabalho. Por exemplo, digite o nome deste objeto: `faithful`. Essa moldura contém dados sobre erupções do Velho Fiel, um géiser famoso que existe no Parque Nacional de Yellowstone, nos EUA. Cada linha da moldura contém duas informações sobre uma erupção: a duração em minutos da erupção (coluna com nome `eruptions`) e o tempo de espera até a próxima erupção (coluna com nome `waiting`). Se você estiver interessado apenas no vetor de durações das erupções, você pode referir-se a ele pela mesma notação de busca com cifrão que aprendemos antes. Experimente digitar: `faithful$eruptions`. Se as colunas de uma moldura de dados serão usadas com frequência numa sessão com o R, é mais conveniente incluir o nome da moldura na árvore de buscas de nomes de objetos que o R utiliza sempre que você usa qualquer nome numa expressão. Isso se faz pela função `attach`:

```
attach(faithful); hist(eruptions); mean(eruptions)
```

Como você vê, depois da função `attach` acima, não precisamos mais fazer referência ao nome da moldura para trabalhar com a sua coluna de duração de erupções.

Para criar uma moldura de dados a partir de vetores que contêm as informações das colunas (todas na mesma ordem!), podemos usar a função `data.frame`. Estude este exemplo:

```
nm <- c("João","Mary"); ids <- c(20,19); crs <- c(7.5, 7.8); nac <- c(FALSE,TRUE)
turma <- data.frame(nome=nm,idade=ids,CR=crs,estrangeiro=nac); turma
```

Normalmente as molduras de dados não serão criadas a partir de vetores como acima, ou a partir de listas, mas serão importadas de bases de dados externas. O tema da importação de dados em R é extenso e importante ao ponto de merecer um pequeno manual à parte: *R Data Import/Export*, o qual você encontra dentro do menu Ajuda do R, opção “Manuais (em PDF)”. No âmbito desta disciplina, é suficiente você aprender sobre a função `read.table` e as suas derivadas, na conclusão deste estudo no próximo parágrafo.

Existem dois padrões universalmente usados de armazenamento de dados em arquivos de texto. O padrão mais comum chama-se CSV (*comma separated values*). Nele, as colunas de dados em cada linha são delimitadas pelo caractere da vírgula, e o separador decimal é o caractere do ponto. Por exemplo, a pequena turma de dois alunos que criamos antes com a função `data.frame` seria armazenada num arquivo de texto CSV da seguinte forma:

```
João,20,7.5,FALSE
Mary,19,7.8,TRUE
```

Existe uma variação do padrão CSV em que o delimitador das colunas é o caractere ponto-e-vígula, enquanto o separador decimal é a vírgula. A mesma turma de dois alunos seria armazenada assim:

```
João;20;7,5;FALSE
Mary;19;7,8;TRUE
```

Para importar dados nesses padrões, você usa respectivamente as funções `read.csv` e `read.csv2`, tendo por argumento o nome entre aspas do arquivo de texto que contém os dados (as duas funções são variações da função `read.table`). O produto dessas funções é uma moldura de dados com a mesma informação contida no arquivo externo. Por exemplo: `read.csv("turma.csv")`.

O segundo padrão comum de armazenamento de dados em texto é o das colunas de largura fixa. Nele, cada coluna tem um número predeterminado de caracteres. Não há um caractere que

separe as colunas. No exemplo da turma, suponha que o criador da base de dados tenha especificado 15 caracteres para o nome dos alunos, 2 caracteres para as idades, 4 caracteres para o CR e 5 caracteres para o último valor lógico. O arquivo de texto nesse formato teria o seguinte conteúdo:

```
João           20 7,5FALSE
Mary           19 7,8TRUE
```

Observe que há um espaço aparente entre a idade e o CR apenas porque se definiu que o campo CR ocuparia 4 caracteres (para que se possa representar um CR 10,0).

Para importar dados nesse formato, usa-se a função `read.fwf` (*fixed width format,* outra variação da mesma função `read.table`). Na lista de exercícios, você terá oportunidade de experimentar com essas funções de importação.

Lista de exercícios 4

(1) Considere de novo o exemplo da turma de dois alunos. Crie uma lista chamada `aluna` que contenha os dados de Mary exatamente como no primeiro exemplo do guia. Depois, recupere cada um dos dados da aluna, de maneira que você não precise lembrar da ordem dos dados.

(2) Os elementos de uma lista são objetos que podem receber novo valores individualmente através do operador de designação. Mude o nome na lista `aluna` que você criou no exercício anterior de “Mary” para “Magdalene”.

(3) Crie um vetor alfanumérico chamado `dst` de três elementos com os textos: “Normal”, “Uniforme” e “Poisson”. Depois, crie um vetor numérico chamado `rn` de três elementos contendo, nesta ordem, um número aleatório normal padrã0; um número aleatório de uma distribuição uniforme entre 0 e 1; e um número aleatório de uma distribuição de Poisson com parâmetro 2. Finalmente, usando esses dois vetores, crie uma moldura de dados chamada `sorteios`, cuja primeira coluna com nome “distribuição” é o vetor `dst` e a segunda com nome “exemplo” é o vetor `rn`.

(4) Descubra se existe em R uma função que desfaça o que a função `attach` faz.

(5) Use o Bloco de Notas do Windows para criar manualmente, no diretório de trabalho, os arquivos de texto nos padrões CSV e CSV alternativo com a turma de dois alunos, como descritos no guia de estudo. Depois, usando as funções adequadas, importe esses dados para uma muldura dentro do R.

(6) Faça o mesmo exercício, mas agora com o arquivo no padrão de colunas com larguras fixas. Você precisará estudar a documentação da função `read.fwf`.

* * *