Roteiro 2

Melina Leite

Departamento de Ecologia IB-USP

Contents

Neste roteiro passaremos novamente em alguns tópicos de leitura, manipulação e transformação de dados. Logo após, trataremos de estatísticas descritivas e gráficos exploratórios. Falaremos também dos índices de diversidade mais usados em estudos de levantamento e monitoramento ambiental e como calculá-los no R. Veremos como fazer curvas de rarefação dos dados no R. Por fim, veremos algumas formas de elaborar gráficos para apresentação de resultados.

Importando e verificando os dados

Relembrando a importação de dados

Baixe os arquivos de dadosilhas e sps e faça a leitura deles no R:

```
ilhas <- read.csv2("ilhas.csv",header=T,row.names = 1)
ilhas <- as.matrix(ilhas)
sps <- read.csv2("sps.csv",header=T, dec=".")
```

A tabela `ilhas` é uma matriz da abundância de espécies (linhas) em diferentes ilhas (colunas). Quando importamos dados com as funções da família `read.table`, os objetos criados são sempre da classe data frame, por isso a necessidade de transformar o objeto em matriz.

A tabela `sps` é um data frame contendo as informações de atributos das mesmas espécies presentes em `ilhas`.

Verificação dos dados

Após a importação precisamos verifica se a tabela foi importada corretamente, e se há erros na tabela

```
str(ilhas)
str(sps)
```

volta a falar da importação de dados, transformar em data frames. mostrar as ferramentas para verificar os dados, inclusive no Rstudio (ver tabela) como consertar erros usando indexação NAs

Utilizando as funções sort, order, rank

As três funções `sort`, `order` e `rank`são relacionadas, porém fazem coisas diferentes e é preciso prestar atenção. Baixe o arquivo houses.txt e carregue ele no R. Vamos ver a diferença entre as funções na prática:

```
houses<-read.table("houses.txt",header=T)
houses

ranks <- rank(houses$Price)
sorted <- sort(houses$Price)
ordered <- order(houses$Price)

view <- data.frame(houses$Price,ranks,sorted,ordered)
view
```

A função `rank` retorna a posição do ranking que aquele preço está. Como o vetor `Price`tem 12 números, o preço mais alto (325) vai ter o maior valor (12), e o preço mais baixo (95) o menor valor, 1. Os rankings fracionados indicam empate, por exemplo existem dois preços de 188, seus rankings seriam 8 e 9, como estão empatados a função atribuiu 8.5 a ambos.

A função `sort` é a mais intuitiva, ela ordena os preços do menor para o maior, ou do maior para o menor se você usar o argumento `decreasing=TRUE`. Porém, pode ser uma função perigosa, porque se você a usa em uma coluna de um data frame, ela poderá desacoplar a coluna sendo ordenada das demais colunas. Ou seja, você só mudará a coluna em questão deixando todo o data frame inalterado, o que fará perder a conexão entre os dados das linhas e as variáveis nas colunas.

A função `order` pode ser considerada a mais importante e um pouco menos intuitiva. Veja os números na coluna `odered`, eles também estão numerados de 1 a 12 como em `ranks`, porém eles querer dizer algo bem diferente. O primeiro valor (9) é número da linha em que o menor valor (95) se encontra. O segundo valor (6) é o número da linha em que o segundo menor valor (101) se encontra, e assim por diante. Observe novamente o objeto `view` e tente entender a lógica de `order`.

A função `order` é particularmente útil na ordenação de data frames inteiros através da indexação. Veja o exemplo:

```
#ordenando houses em função do preço, perceba que a coluna Location também muda
houses[order(houses$Price), ]

# veja a diferença se eu usar a função sort
houses$Price <- sort(houses$Price)
houses # OPS! bagunçou o data frame!
```

Se você quiser ordenar o data frame por uma coluna de maneira decrescente, utilize o argumento `decreasing=TRUE` da função `order`. Essa função é relativamente boa quando queremos exportar uma tabela de dados/resultados para apresentar em relatórios/apresentações/artigos. Assim, você ordena o data frame pela coluna que você achar mais importante na hora de apresentar seus dados/resultados.

Estatística descritiva

colocar as principais funções para estatística descritiva em vetores, matrizes e data frames

Família Apply e aggregate

Gráficos exploratórios

Índices de diversidade

Curvas de Rarefação

Criação e edição de gráficos

Aprofundando-se

dar exemplos de pacotes plyr dplyr, maggritt, ggplot2

Exercícios!

1. Para gerar uma amostra de 10.000 números de uma distribuição Normal com média 30 e desvio padrão 7, utilize o comando:

```
vnormal = rnorm(10000, 30, 7)
```

- Qual o somatório das observações no vetor 'vnormal' que são maiores que 44? E maiores que 51?
- Como você excluiria a maior observação do vetor 'vnormal'?

2. Aninhamento de comunidades

O termo "aninhamento" (nesting) é usado para a situação em que comunidades mais pobres em espécies são um subconjunto das comunidades mais ricas. Uma análise exploratória rápida de aninhamento é ordenar as linhas e as colunas de uma matriz binária de ocorrência das espécies por comunidades.

1. Crie um objeto da classe `matrix` com a matriz de ocorrência de mamíferos em topos de montanhas (retire a extensão pdf). (DICA: a função read.table retorna um data frame. Use a função as.matrix para mudar a classe para matriz.)

2. Use o ordenamento por indexação para criar uma matriz com as comunidades por ordem decrescente de espécies, e as espécies por ordem decrescente de frequência de ocorrência. (OUTRA DICA: lembre-se da função apply!).

3. A matriz resultante tem sinais de aninhamento? Por que?