Exercícios 4

Manipulação de ficheiros

Em termos computacionais, a informação é habitualmente transmitida em blocos a que denominamos ficheiros. O **R** disponibiliza uma vasta colecção de funções para manipular ficheiros de diferentes formatos. Neste exercício, pretende-se fazer uso de algumas funções de manipulação de ficheiros, e dos objectos resultantes do seu carregamento numa sessão de **R**.

Considere o seguinte contexto. Tornou-se num reconhecido *data analyst*, e uma das maiores empresas da indústria petrolífera está a requisitar os seus serviços. Eles estão a começar a explorar um novo reservatório, tendo já disponível os dados obtidos no primeiro furo (exemplo ilustrativo na Figura 1).

Figura 1: Bomba de produção no oeste do Texas (fonte: wikimedia.org).

O engenheiro de reservatórios tem conhecimento da sua proficiência em **R**, então enumerou as seguintes tarefas a executar.

Nota: Resolva as questões recorrendo, tanto quanto possível, a uma abordagem mais funcional e menos imperativa (i.e., usando funções e evitando usar ciclos, ou, dizendo de outro modo, tirando maior proveito das funcionalidades do **R**).

1. Os dados estão disponíveis para download na seguinte localização: `http://tinyurl.com/welldata1`. Carregue os dados para uma estrutura de dados adequada.

   Solução:

   ```
   # colocar entre aspas o caminho completo para o ficheiro
   welldata1 = read.csv("/home/julio/well_data1.dat")
   ```

2. Os dados recebidos contêm 7 variáveis:

   X: Coordenada X

   Y: Coordenada Y

   Z: Coordenada Z

facies: diferentes tipos de rocha

density: densidade da rocha

porosity: porosidade da rocha

permeability: permeabilidade da rocha (medida em mD)

Para evitar problemas na análise de dados, verifique se cada coluna ficou com o tipo de dados mais adequado, e ajuste conforme necessário.

Solução:

É possível verificar o tipo de dados em cada coluna através da função `str`.

```
> str(welldata1)
'data.frame':        4438 obs. of  7 variables:
$ X           : int  73 73 73 73 73 73 73 73 73 73 ...
$ Y           : int  5 5 5 5 5 5 5 5 5 5 ...
$ Z           : int  199 198 197 196 195 194 193 192 191 190 ...
$ facies      : int  0 0 0 0 0 0 0 0 0 0 ...
$ density     : num  2.39 2.4 2.4 2.42 2.42 ...
$ porosity    : num  0.0549 0.0523 0.0522 0.0374 0.0361 0.039 0.0314 0.0464 0.0547 0.0537 ...
$ permeability: num  6.81 4.38 5.78 3.51 2.54 ...
```

Verifica-se, assim, que todas as colunas são compostas por elementos numéricos. Ora, sendo a variável *facies* categórica, o tipo de objecto mais apropriado é o `factor` e não o `numeric`. É, então, necessário alterar o tipo de elementos nesta coluna.

```
> welldata1$facies = as.factor(welldata1$facies)
> str(welldata1)
'data.frame':        4438 obs. of  7 variables:
$ X           : int  73 73 73 73 73 73 73 73 73 73 ...
$ Y           : int  5 5 5 5 5 5 5 5 5 5 ...
$ Z           : int  199 198 197 196 195 194 193 192 191 190 ...
$ facies      : Factor w/ 4 levels "0","1","2","3": 1 1 1 1 1 1 1 1 1 1 ...
$ density     : num  2.39 2.4 2.4 2.42 2.42 ...
$ porosity    : num  0.0549 0.0523 0.0522 0.0374 0.0361 0.039 0.0314 0.0464 0.0547 0.0537 ...
$ permeability: num  6.81 4.38 5.78 3.51 2.54 ...
```

3. As fácies correspondem a diferentes zonas no reservatório. Identifique quantas amostras recebeu de cada uma das zonas.

Solução:

```
> table(welldata1$facies)

   0    1    2    3
2718  456  989  275
```

4. Como primeira abordagem para identificar zonas de maior interesse (*high pay zones*), obtenha um resumo com os estatísticos base de cada variável, em cada zona. Deste modo, poderá saber os valores típicos em cada uma das fácies.

Solução:

Considerando que interessa apenas analizar os estatísticos base das variáveis relacionadas com as propriedades da rocha (últimas 3 colunas):

```
by(welldata1[, 5:7], welldata1$facies, summary)
```

5. Outra estratégia consiste em encontrar as amostras com maiores valores de porosidade ($\geq 0,3$). Identifique essas amostras. Qual é a zona com maior número dessas amostras?

Solução:

```
high.poro = subset(welldata1, welldata1$porosity >= 0.3)

> table(high.poro$facies)

0  1  2  3
0  0 59  0
```

6. Entretanto, recebeu dados adicionais, que foram recolhidos mais tarde. Foram enviados através da ligação `http://tinyurl.com/welldata2`. Este novo conjunto de dados contém as seguintes variáveis:

   X: Coordenada X
   Y: Coordenada Y
   Z: Coordenada Z
   Vp: Velocidade das ondas P (km/s)
   Vs: Velocidade das ondas S (km/s)

   Foi-lhe passada a informação de que o processo de aquisição de dados teve alguns problemas, e que é possível que tenha recebido a mesma amostra mais do que uma vez. Tome as medidas necessárias para corrigir esse possível erro.

Solução:

```
# este ficheiro está separado por tabulações e não por vírgulas
welldata2 = read.delim("/home/julio/well_data2.dat")
# encontrar e remover registos em duplicado
welldata2 = subset(welldata2, !duplicated(welldata2))
```

7. Com esse percalço ultrapassado, pode agora proceder à junção dos dois conjuntos de dados. Verifique as variáveis que têm em comum, e junte-os de acordo com as mesmas.

Solução:

```
# verificar que variáveis têm em comum
> intersect(names(welldata1), names(welldata2))
[1] "X" "Y" "Z"
# juntar
welldata = merge(welldata1, welldata2)
```

8. Embora já tenha preparado todos os dados disponíveis, pode ainda ajudar na próxima etapa da modelação do reservatório, providenciado duas variáveis úteis. A impedância acústica é dada pelo produto da densidade com a velocidade sísmica, que varia entre diferentes camadas de rocha. Adicione duas colunas ao conjunto de dados:

   Ip: impedância acústica considerando as ondas P

   Is: impedância acústica considerando as ondas S

   Solução:

```
welldata = transform(welldata, Ip = Vp * density, Is = Vs * density)
```

9. O seu trabalho está concluído. Agora, resta prepará-lo para ser entregue. Guarde os dados que preparou num formato de ficheiro apropriado.

   Solução:

```
write.csv(welldata, "/home/julio/welldata.dat", row.names=F)
```