Additional Material for Indexing and Slicing Data Frames

Sudhakar Kumar

18 February 2019

Data Frame

A data frame is a rectangular collection of variables (in the columns) and observations (in the rows). For example, the data frame we created from the data of captains and we saved it as **captaincy**. Let us have a look at this data frame:

```
captaincy <- read.csv("CaptaincyData.csv")
print(captaincy)
```

```
##     names    Y played won lost   victory
## 1    Mahi 2012     45  22   12 0.4888889
## 2  Sourav 2004     49  21   13 0.4285714
## 3   Azhar 2000     47  14   14 0.2978723
## 4   Sunny 1980     47   9    8 0.1914894
## 5 Pataudi 1965     40   9   19 0.2250000
## 6  Dravid 2008     25   8    6 0.3200000
```

Here, we can see that this data frame contains 6 columns, named **names**, **Y**, **played**, **won**, **lost** and **victory**. Under each column, we have observations for 6 different captains.

Indexing a data frame

This refers to the task of extracting any row(s) or column(s) of the data frame. It also means selecting a subset of the elements in order to use them in further analysis. Depending upon the method of indexing used, we can categorize it as given below:

- **Numeric Indexing** – If we use the row number or column number for extracting required information, it is known as numeric indexing.
- **Logical Indexing** – If we apply certain condition to extract row(s) or column(s), it is known as logical indexing.
- **Name Indexing** – If we use the row name or column name for extracting required information, it is known as name indexing.

We have learnt how to index rows or columns in a data frame. To index a data frame, we use the bracket notation with two indices. Just like in matrix algebra, the indices for a data frame follow the `R x C` principle. It means first index (`R`) is for rows and the second index (`C`) is for columns. When we want to extract only column(s), we use the second index and leave the first index blank. Similarly, when we want to extract only row(s), we use the first index and leave the second index blank. Leaving an index blank indicates that we want to keep all the elements in that dimension.

While indexing third row of **captaincy**, we used the command `captaincy[3,]` as given below. Here, 3 refers to the row number, and a blank index after comma indicated that we want to keep all the elements in third row.

```
captaincy <- read.csv("CaptaincyData.csv")
captaincy[3,]
```

```
##   names    Y played won lost   victory
## 3 Azhar 2000     47  14   14 0.2978723
```

For extracting the third column, we used the command `captaincy[3]` as given below. Here, we have used only one index and no comma. R language will take this index as a column number by default and print all the elements available in the third column (played).

```
captaincy <- read.csv("CaptaincyData.csv")
captaincy[3]
```

```
##   played
## 1     45
## 2     49
## 3     47
## 4     47
## 5     40
## 6     25
```

For extracting third column, we can also use `captaincy[,3]`, as given below. Here, the row number is left blank and thus, R language will print all the elements of third column.

```
captaincy <- read.csv("CaptaincyData.csv")
captaincy[,3]
```

```
## [1] 45 49 47 47 40 25
```

For extracting the third entry of fourth column of captaincy data frame, we used double square bracket notation. Thus, we used `captaincy[[4]][3]` as given below. The double square brackets can be used to reference data frame columns. An additional set of square brackets can be used in conjunction with the `[[]]` to reference a specific element in that vector of elements. Here, 4 in double square brackets is used to reference fourth column of captaincy data frame. Next, 3 in square brackets indicates that we are looking for third element of fourth column.

```
captaincy <- read.csv("CaptaincyData.csv")
captaincy[[4]][3]
```

```
## [1] 14
```

This is similar to extracting the element located as third row and fourth column. Hence, this can also be achieved by using `captaincy[3,4]` as given below.

```
captaincy <- read.csv("CaptaincyData.csv")
captaincy[3,4]
```

```
## [1] 14
```

Slicing a data frame

The process of selecting specific rows and columns of data based on some criteria is commonly known as slicing a data frame. This is also known as subsetting data. Suppose we want to subset **captaincy data frame** for extracting the names of **Mahi**, **Sourav**, **Pataudi** along with the number of matches they **played** and **won**. For this, we need to know the data of first, second and fifth rows with their corresponding entries in first (names), third (played) and fourth (won) column. So, we type the command as given below.

```
captaincy <- read.csv("CaptaincyData.csv")
captaincy[c(1,2,5), c(1,3,4)]
```

```
##     names played won
## 1    Mahi     45  22
## 2  Sourav     49  21
## 5 Pataudi     40   9
```