Additional Material for Lists and its Operations

Lists in R

A list is an R object. It has components of mixed data types like

- strings
- numbers
- vectors or
- some other list inside it

A list can also contain a matrix or a function as its elements.

Creating a list in R

For example, we will create a list containing a vector, a matrix and one built-in dataset. First, we need to declare these three components. As `iris` dataset is having 150 rows, we will consider its seven rows only.

```
myVector <- c(1:5)
myMatrix <- matrix(1:15, nrow = 5, ncol = 3, byrow = TRUE)
myDataSet <- iris[c(1:2,51:52,100:102),]
```

Now, we use `list` function to create the list and `names` function to add names to the elements of our list.

```
myList <- list(myVector, myMatrix, myDataSet)
names(myList) <- c("vector", "matrix", "dataset")
print(myList)
```

```
## $vector
## [1] 1 2 3 4 5
##
## $matrix
##      [,1] [,2] [,3]
## [1,]    1    2    3
## [2,]    4    5    6
## [3,]    7    8    9
## [4,]   10   11   12
## [5,]   13   14   15
##
## $dataset
##     Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 1            5.1         3.5          1.4         0.2     setosa
## 2            4.9         3.0          1.4         0.2     setosa
## 51           7.0         3.2          4.7         1.4 versicolor
## 52           6.4         3.2          4.5         1.5 versicolor
## 100          5.7         2.8          4.1         1.3 versicolor
## 101          6.3         3.3          6.0         2.5  virginica
## 102          5.8         2.7          5.1         1.9  virginica
```

Accessing the elements of a list

In case of named list, we can retrieve the elements by their names.

```
myList$dataset
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 1            5.1         3.5          1.4         0.2     setosa
## 2            4.9         3.0          1.4         0.2     setosa
## 51           7.0         3.2          4.7         1.4 versicolor
## 52           6.4         3.2          4.5         1.5 versicolor
## 100          5.7         2.8          4.1         1.3 versicolor
## 101          6.3         3.3          6.0         2.5  virginica
## 102          5.8         2.7          5.1         1.9  virginica
```

Also, the elements of a list can be retrieved by using the single square bracket `[]` operator.

```
myList[3]
```

```
## $dataset
##     Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 1            5.1         3.5          1.4         0.2     setosa
## 2            4.9         3.0          1.4         0.2     setosa
## 51           7.0         3.2          4.7         1.4 versicolor
## 52           6.4         3.2          4.5         1.5 versicolor
## 100          5.7         2.8          4.1         1.3 versicolor
## 101          6.3         3.3          6.0         2.5  virginica
## 102          5.8         2.7          5.1         1.9  virginica
```

In order to reference a list member directly, we have to use the double square bracket `[[]]` operator.

```
myList[[3]]
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 1            5.1         3.5          1.4         0.2     setosa
## 2            4.9         3.0          1.4         0.2     setosa
## 51           7.0         3.2          4.7         1.4 versicolor
## 52           6.4         3.2          4.5         1.5 versicolor
## 100          5.7         2.8          4.1         1.3 versicolor
## 101          6.3         3.3          6.0         2.5  virginica
## 102          5.8         2.7          5.1         1.9  virginica
```

Thus, we can observe that the difference between single square bracket and double square brackets is that with double square brackets, the name of the element (`dataset`) is not displayed. According to a discussion in the forum of **Analytics Vidhya**, we need `[[` (double square brackets) when working with lists. This is because when `[` (single square bracket) is applied to a list, it always returns a list: it never gives us the contents of the list. To get the contents, we need `[[` (double square brackets). This can also be verified by using `class` function.

```
class(myList$dataset); class(myList[3]); class(myList[[3]])
```

```
## [1] "data.frame"
```

```
## [1] "list"
```

```
## [1] "data.frame"
```