CHAPTER 25

Subsetting

It is quite often that we input a big data set first. Then we slice it according to our need. In this chapter, we discuss various ways to do so. Assume that we have the following vector of **x**.

```
> x<-seq(1,5,by=0.123)
> x
  [1] 1.000 1.123 1.246 1.369 1.492 1.615 1.738 1.861 1.984 2.107
[11] 2.230 2.353 2.476 2.599 2.722 2.845 2.968 3.091 3.214 3.337
[21] 3.460 3.583 3.706 3.829 3.952 4.075 4.198 4.321 4.444 4.567
[31] 4.690 4.813 4.936
```

For a subset of **x**, we use the following codes:

```
> y<-x[1:10]
> y
[1] 1.000 1.123 1.246 1.369 1.492 1.615 1.738 1.861 1.984 2.107
```

For a matrix, we can choose certain columns and/or rows. Assume x is an n-by-m matrix.

```
> a<-x[,1] # choose the first column
> b<-x[1:100,1:2] # choose the first 100 rows & columns 1 and 2
```

25.1. Introduction

In this chapter, we discuss how to retrieve part of a data set for further analysis. For example, we have an R data set called **retD50.RData** with only three columns (ticker, date, and return). The data set can be downloaded by using the link at http://canisius.edu/~yany/RData/retD50.RData. We can save it to a specific subdirectory. Then we change our working directory by clicking **File** then **Change dir...** and choose your correct directory. Below, we load the R data set from our working directory.

```
> web<-url("http://canisius.edu/~yany/RData/retD50.RData")
> load(web)
> close(web)
> head(retD50)
  ticker date ret
1883624 IBM 1962-01-02 -99.000000
1883625 IBM 1962-01-03 0.007663
1883626 IBM 1962-01-04 -0.011407
1883627 IBM 1962-01-05 -0.019231
1883628 IBM 1962-01-08 -0.019608
1883629 IBM 1962-01-09 0.012000
```

To choose a specific stock, we apply an equal condition (==) to column 1.

```
> load(retD50.RData)
> ibm<-subset(retD50,retD50[,1]== "IBM")
```

The **unique()** function can be used to show all unique value such as number of stocks available in the data set.

```
> y<-unique(retD50[,1])
> length(y)
[1] 50
```

25.2. Scalar, Vector, and Matrix

The scalar is a variable that takes many different values at different times. However, at one specific time, it takes one value only.

```
> x<-10
```

A vector is a column of data n by 1 (i.e., it has n data items).

```
> x<-1:10
```

A matrix has n rows and m columns (n by m). Thus, a matrix has n*m data items.

```
>x<-1:12
> y<-matrix(x,3,4,byrow=T)
> y
  [,1] [,2] [,3] [,4]
[1,]  1  2  3  4
[2,]  5  6  7  8
[3,]  9 10 11 12
```

There is only one data type for a matrix. Obviously the y variable from the above codes is an integer type.

```
> typeof(y)
[1] "integer"
```

However, if we assign a string to just one data item, the whole matrix becomes string instead of numeric.

```
> y[1,1]<-"good"
> y
     [,1]   [,2] [,3] [,4]
[1,] "good" "2"  "3"  "4"
[2,] "5"    "6"  "7"  "8"
[3,] "9"    "10" "11" "12"
> typeof(y)
[1] "character"
```

25.3. Getting a Subset from a Vector

The easiest way to get a subset is to specify the beginning and ending positions.

```
> x<-rnorm(50)
> y<-x[1:10] # retain the first 10 values
```

A negative-vector index indicates exclusion. Assume that a vector of x has fifty values; **x[10]** indicates the tenth value while **x[-10]** has forty-nine values except the tenth value.

```
> x<-rnorm(50)
> y<-x[-10]
> length(y)
[1] 49
```

25.4. Getting a Subset from a Matrix

Below, we show three ways to retrieve a subset from a matrix.

```
X[1:2,] # the first two rows
X[1,] # the first row
X[1: (n-1),5] # rows from 1 to n-1 and 5th column
```

The subset() function allows us to choose a subset from a given data set based on certain conditions.

```
> ibm<-subset(x,x[,1]=='IBM')
```

25.5. Getting a Specific Year's Data

If a variable related to the date is defined by the **as.Date()** function, then the codes will be simple. First, we download and save an R data set called **retDIMB.RData** from the Web page http://canisius.edu/~yany/RData/retDIBM.RData. For a specific subtime period, we can use the following program:

```
> load("retDIBM.RData")
> date1<- as.Date("2011-02-02")
> date2<-as.Date("2011-02-11")
> x<-subset(ibm,ibm[,1]>=date1 & ibm[,1]<=date2)
```

After we change our working directory that contains the above data set, we issue the following codes. The **load()** function is used to upload an R data set.

```
> load("retDIBM.RData")
> ls()
[1] "EDM1" "ibm" "sp500" "x" "y"
> head(ibm)
  date ret
1 1962-01-02 -99.000000
2 1962-01-03 0.007663
3 1962-01-04 -0.011407
4 1962-01-05 -0.019231
5 1962-01-08 -0.019608
6 1962-01-09 0.012000
> x<-subset(ibm,format(ibm[,1], "%Y ")==2000)
> dim(x)
[1] 252 2
```

With two conditions (conditions A and B) to filter out our data, we use & for an *and* condition (i.e., both A and B conditions must be true) and | for an *or* condition (i.e., at least one condition is true). For example, if we want to retrieve IBM's return for the period from **date1** to **date2**, we use the following codes:

```
> load("retDIBM.RData ")
> date1<-as.Date("2000-02-02")
> date2<-as.Date("2000-02-10")
> x<-subset(retD50,retD50[,1]== "IBM " & retD50[,2]>date1 & retD50[,2]<date2)
> x
  ticker date ret
1893215 IBM 2000-02-03 0.031870
1893216 IBM 2000-02-04 -0.012821
1893217 IBM 2000-02-07 -0.012987
1893218 IBM 2000-02-08 0.042265
1893219 IBM 2000-02-09 -0.012146
```

If the date-related variable is defined as an integer, we can use the following codes. For example, **19260113/10000= 1926.0113**. Its integer will be 1926.

```
> y<-subset(x,ticker=='A' & as.integer(date/10000)==1926)
```

Exercises

25.1. How do you get the dimensions of a matrix?

25.2. How do you get rows 25 to 50 and columns 1 to 20 from a matrix of x?

25.3. You are given a matrix of x. How do you print the first and last several lines of x?

25.4. What are the advantages and disadvantages of a data frame versus a matrix?

25.5. Can a matrix hold different types of data?

25.6. Download the **retD50.Rdata** and retrieve two stocks' data, such as IBM's and DELL's, from http://canisius.edu/~yany/RData/retD50.RData.