Session 2 - Introducing R 2

Farokh Kakaei

January 2015

Contents

1 Subsetting

- Accessing values to a specific portion of a data structure (vectors, matrices, lists, data frames)

1.1 Vectors

- Subsetting vectors using an integer index (vectors are one based, i.e. indices start from 1)

```
# make randomness reproducible

# creating and showing base vector
vec<-sample(1:100, 15, replace=FALSE)
vec
```

```
##  [1] 94 75 95  8 21 33 92 12  4 15 49 60 79 70 68
```

```
# retrieving the fifth element
vec[5]
```

```
## [1] 21
```

```
# assigning 500 to the fifth element
vec[5]<-500
vec
```

```
##  [1]  94  75  95   8 500  33  92  12   4  15  49  60  79  70  68
```

- Negative index

```
# creating and showing the base vector
vec<-sample(1:100, 15, replace=FALSE)
vec
```

```
##  [1]  7  6 68 11 33 72 86 49 71 10 77 45 21 60 61
```

```
# retrieving all elements except the fifth one
vec[-5]
```

```
##  [1]  7  6 68 11 72 86 49 71 10 77 45 21 60 61
```

```
# assigning 500 to all elements except the fifth one
vec[-5]<-500
vec
```

```
##  [1] 500 500 500 500  33 500 500 500 500 500 500 500 500 500 500
```

- Subsetting vectors using an integer index vector(a vector of indices)

```r
vec<-sample(1:100, 15, replace=FALSE)
vec
```

```
##  [1] 74  1 25 26  4 35 95 89 56 40 80 23 55 67 69
```

```r
vec[6:10]
```

```
## [1] 35 95 89 56 40
```

```r
vec2<-vec
vec3<-vec
vec2[6:10]<-500
vec3[6:10]<-c(500, 600, 700, 800, 1000)
vec2
```

```
##  [1]  74   1  25  26   4 500 500 500 500 500  80  23  55  67  69
```

```r
vec3
```

```
##  [1]   74    1   25   26    4  500  600  700  800 1000   80   23   55   67
## [15]   69
```

```r
vec[seq(1,15, by=3)]
```

```
## [1] 74 26 95 40 55
```

```r
vec2<-vec
vec3<-vec
vec2[seq(1,15, by=3)]<-100
vec3[seq(1,15, by=3)]<-c(100, 200, 300, 400, 500)
vec2
```

```
##  [1] 100   1  25 100   4  35 100  89  56 100  80  23 100  67  69
```

```r
vec3
```

```
##  [1] 100   1  25 200   4  35 300  89  56 400  80  23 500  67  69
```

```r
vec[c(1,3,7, 13)]
```

```
## [1] 74 25 95 55
```

```r
vec2<-vec
vec3<-vec
vec2[c(1,3,7, 13)]<-200
vec3[c(1,3,7, 13)]<-c(200, 300, 400, 500)
vec2
```

```
##  [1] 200   1 200  26   4  35 200  89  56  40  80  23 200  67  69
```

```r
vec3
```

```
##  [1] 200   1 300  26   4  35 400  89  56  40  80  23 500  67  69
```

```r
vec[rep(c(1,2), 2)]
```

```
## [1] 74  1 74  1
```

```r
vec2<-vec
vec3<-vec
vec2[rep(c(1,2), 2)]<-300
vec3[rep(c(1,2), 2)]<-c(300, 400, 500, 600)
vec2
```

```
##  [1] 300 300  25  26   4  35  95  89  56  40  80  23  55  67  69
```

```r
vec3
```

```
##  [1] 500 600  25  26   4  35  95  89  56  40  80  23  55  67  69
```

```r
iv<-sample(1:15, 6)
iv
```

```
## [1] 14  2  8 11  6  7
```

```r
vec[iv]
```

```
## [1] 67  1 89 80 35 95
```

```r
vec2<-vec
vec3<-vec
vec2[iv]<-400
vec3[iv]<-c(400, 500, 600, 700, 800, 900)
vec2
```

```
##  [1]  74 400  25  26   4 400 400 400  56  40 400  23  55 400  69
```

```r
vec3
```

```
##  [1]  74 500  25  26   4 800 900 600  56  40 700  23  55 400  69
```

- Negative index vectors

```r
vec<-sample(1:100, 15, replace=FALSE)
vec
```

```
##  [1]   7  92  22  71  69  85  21  75  86  18 100  15  54  94  81
```

```r
vec[-10:-6]
```

```
##  [1]   7  92  22  71  69 100  15  54  94  81
```

```r
vec2<-vec
vec2[-10:-6]<-100
vec2
```

```
##  [1] 100 100 100 100 100  85  21  75  86  18 100 100 100 100 100
```

```r
vec[-(6:10)]
```

```
##  [1]   7  92  22  71  69 100  15  54  94  81
```

```r
vec2<-vec
vec2[-(6:10)]<-200
vec2
```

```
##  [1] 200 200 200 200 200  85  21  75  86  18 200 200 200 200 200
```

```r
vec[-c(1,3,7, 13)]
```

```
##  [1]  92  71  69  85  75  86  18 100  15  94  81
```

```r
vec2<-vec
vec2[-c(1,3,7, 13)]<-300
vec2
```

```
##  [1]   7 300  22 300 300 300  21 300 300 300 300 300  54 300 300
```

```r
iv<-sample(1:15, 6)
iv
```

```
## [1]  1  4 15  6  8 12
```

```r
vec[-iv]
```

```
## [1]  92  22  69  21  86  18 100  54  94
```

```r
vec2<-vec
vec2[-iv]<-400
vec2
```

```
##  [1]   7 400 400  71 400  85 400  75 400 400 400  15 400 400  81
```

- Using name indices

```r
vec1<-c(1, 2,3 ,4, 5)
names(vec1)<-c("A", "B", "C", "D", "E")
vec1
```

```
## A B C D E
## 1 2 3 4 5
```

```r
vec2<-c(a=1, b=2, c=3, d=4, e=5)
vec2
```

```
## a b c d e
## 1 2 3 4 5
```

```r
vec1["B"]
```

```
## B
## 2
```

```r
vec2["c"]
```

```
## c
## 3
```

```r
vec1[c("B", "A", "C")]
```

```
## B A C
## 2 1 3
```

```r
vec2[c("a", "e", "b")]
```

```
## a e b
## 1 5 2
```

- Using logical vectors: accessing values meeting some condition(s)

```r
# creating and showing the base vector
vec<-sample(seq(-100,100), 20)
vec
```

```
##  [1]    2  -83   64  -35  -64  -50   39  -82   48  -38  -43   17  -13  -47
## [15]   57  -25 -100  -70   61   42
```

```r
# retrieve all the values lower than zero
vec[vec<0]
```

```
##  [1]  -83  -35  -64  -50  -82  -38  -43  -13  -47  -25 -100  -70
```

```
# change all the values lower than zero to 200
vec2<-vec
vec2[vec2<0] <- 200
vec
```

```
##  [1]    2  -83   64  -35  -64  -50   39  -82   48  -38  -43   17  -13  -47
## [15]   57  -25 -100  -70   61   42
```

```
vec2
```

```
##  [1]   2 200  64 200 200 200  39 200  48 200 200  17 200 200  57 200 200
## [18] 200  61  42
```

```
# retrieve all the values greater than zero and lower than 50
vec[vec>0 & vec<50]
```

```
## [1]  2 39 48 17 42
```

```
# change all the values greater than zero and lower than 50 to 300
vec2<-vec
vec2[vec>0 & vec<50] <- 300
vec
```

```
##  [1]    2  -83   64  -35  -64  -50   39  -82   48  -38  -43   17  -13  -47
## [15]   57  -25 -100  -70   61   42
```

```
vec2
```

```
##  [1]  300  -83   64  -35  -64  -50  300  -82  300  -38  -43  300  -13  -47
## [15]   57  -25 -100  -70   61  300
```

```
# retrieve all the even values
vec[(vec %% 2)==0]
```

```
##  [1]    2   64  -64  -50  -82   48  -38 -100  -70   42
```

```
# retrieve all the values greater than mean
mean(vec)
```

```
## [1] -16
```

```
vec[vec > mean(vec)]
```

```
## [1]   2  64  39  48  17 -13  57  61  42
```

```
# retrieve all the values which their distance from
# mean is lower than one standard deviation
mean(vec)
```

```
## [1] -16
```

```
sd(vec)
```

```
## [1] 53.537
```

```
vec[abs(vec-mean(vec)) < sd(vec)]
```

```
##  [1]   2 -35 -64 -50 -38 -43  17 -13 -47 -25
```

```
# correct character values
x<-c("Bill", "Mike", "Bill", "John", "Mike", "Bill")
x
```

```
## [1] "Bill" "Mike" "Bill" "John" "Mike" "Bill"
```

```
x[x=="Bill"]<-"William"
x
```

```
## [1] "William" "Mike"    "William" "John"    "Mike"    "William"
```

1.2 Matrices and Arrays

- Retrieving entire rows or columns

```
# creating and showing base matrix
mat<-matrix(sample(1:50, 24), 4, 6)
mat
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]   38   45   29   49   32   43
## [2,]    7   19    6   39   40   50
## [3,]   12   15    4   35   27   26
## [4,]    9   20   44   22   16   17
```

```
# retrieving row 3
mat[3,]
```

```
## [1] 12 15  4 35 27 26
```

```
# retrieving all rows but 3
mat[-3,]
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]   38   45   29   49   32   43
## [2,]    7   19    6   39   40   50
## [3,]    9   20   44   22   16   17
```

```r
# retrieving rows 1 and 3
mat[c(1,3),]
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]   38   45   29   49   32   43
## [2,]   12   15    4   35   27   26
```

```r
# retrieving all rows but 1 and 3
mat[-c(1,3),]
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    7   19    6   39   40   50
## [2,]    9   20   44   22   16   17
```

```r
# retrieving column 3
mat[,3]
```

```
## [1] 29  6  4 44
```

```r
# retrieving all columns but 3
mat[,-3]
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]   38   45   49   32   43
## [2,]    7   19   39   40   50
## [3,]   12   15   35   27   26
## [4,]    9   20   22   16   17
```

```r
# retrieving columns 3 and 6
mat[,c(3,6)]
```

```
##      [,1] [,2]
## [1,]   29   43
## [2,]    6   50
## [3,]    4   26
## [4,]   44   17
```

```r
# retrieving all columns but 3 and 6
mat[, -c(3,6)]
```

```
##      [,1] [,2] [,3] [,4]
## [1,]   38   45   49   32
## [2,]    7   19   39   40
## [3,]   12   15   35   27
## [4,]    9   20   22   16
```

- Using integer indices (matrix rows and columns are one based, i.e. indices start from 1)

```r
# creating and showing base matrix
mat<-matrix(sample(1:20, 12), 3, 4)
mat
```

```
##      [,1] [,2] [,3] [,4]
## [1,]   12   20    9    2
## [2,]   17   19    3    4
## [3,]    7   15   11    8
```

```r
# retrieving the element at row 2 and column 3
mat[2,3]
```

```
## [1] 3
```

```r
# assigning 50 to the element at row 2 and column 3
mat[2,3]<-50
mat
```

```
##      [,1] [,2] [,3] [,4]
## [1,]   12   20    9    2
## [2,]   17   19   50    4
## [3,]    7   15   11    8
```

- Using integer index vectors

```r
# creating and showing base matrix
mat<-matrix(sample(1:30, 12), 3, 4)
mat
```

```
##      [,1] [,2] [,3] [,4]
## [1,]   16   17   25   19
## [2,]   22   26    8   29
## [3,]    3   18    2   24
```

```r
# retrieving the elements at rows 1 and 2  and columns 2 and 3
mat[1:2,2:3]
```

```
##      [,1] [,2]
## [1,]   17   25
## [2,]   26    8
```

```r
# assigning a submatrix to the elements at rows 1 and 2  and columns 2 and 3
submat<-matrix(c(50, 51,52,53), 2, 2)
mat[1:2,2:3]<-submat
mat
```

```
##      [,1] [,2] [,3] [,4]
## [1,]   16   50   52   19
## [2,]   22   51   53   29
## [3,]    3   18    2   24
```

- Using an index matrix

```
# creating and showing base matrix
mat<-matrix(sample(1:50, 24), 4, 6)
mat
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]   32   45   38   47   48   10
## [2,]   26    6   30   43   46    3
## [3,]   44   31   11   13   25   27
## [4,]    1   40   16    4    2   24
```

```
# retrieving the elements at rows 1 and 2  and columns 2 and 3
im<-matrix(c(1,2,2,4,4,6), 3,2)
im
```

```
##      [,1] [,2]
## [1,]    1    4
## [2,]    2    4
## [3,]    2    6
```

```
mat[im]
```

```
## [1] 47 43  3
```

- Retrieving vectors as Matrices (drop=FALSE)

```
mat<-matrix(sample(1:50, 12), 3, 4)
mat
```

```
##      [,1] [,2] [,3] [,4]
## [1,]   31   38   13   15
## [2,]   45    9    5   37
## [3,]   20    1   26    4
```

```
# the result of the following command is a vector (oflength one)
mat[2,3]
```

```
## [1] 5
```

```
# the result of the following command is a vector (of length four)
mat[3,]
```

```
## [1] 20  1 26  4
```

```
# the result of the following command is a matrix (one by one)
mat[2,3, drop=FALSE]
```

```
##      [,1]
## [1,]    5
```

```
# the result of the following command is a matrix (one by four)
mat[3, , drop=FALSE]
```

```
##      [,1] [,2] [,3] [,4]
## [1,]   20    1   26    4
```

1.3 Data Frames

- Retireving rows using row indices

```
# building the sample data frame
id<-1:10
name<-sample(LETTERS, 10, replace=TRUE)
listening<-sample(seq(6,7, by=0.5), 10, replace=TRUE)
reading<-listening + sample(seq(-1,1, 0.5),1)
writing<-listening + sample(seq(-1,1, 0.5),1)
speaking<-listening + sample(seq(-1,1, 0.5),1)
ielts<-data.frame(id, name, listening, reading, writing, speaking)
ielts
```

```
##    id name listening reading writing speaking
## 1   1    I       6.0     5.0     6.5      5.0
## 2   2    Q       6.5     5.5     7.0      5.5
## 3   3    H       6.0     5.0     6.5      5.0
## 4   4    E       6.5     5.5     7.0      5.5
## 5   5    T       6.5     5.5     7.0      5.5
## 6   6    G       7.0     6.0     7.5      6.0
## 7   7    I       6.0     5.0     6.5      5.0
## 8   8    H       6.5     5.5     7.0      5.5
## 9   9    H       7.0     6.0     7.5      6.0
## 10 10    T       7.0     6.0     7.5      6.0
```

```
# retireving row number 4
ielts[4, ]
```

```
##   id name listening reading writing speaking
## 4  4    E       6.5     5.5       7      5.5
```

```
# retireving rows 4,6,9
ielts[c(4,6,9), ]
```

```
##   id name listening reading writing speaking
## 4  4    E       6.5     5.5     7.0      5.5
## 6  6    G       7.0     6.0     7.5      6.0
## 9  9    H       7.0     6.0     7.5      6.0
```

```
# retireving rows 1 to 5
ielts[1:5, ]
```

```
##   id name listening reading writing speaking
## 1  1    I       6.0     5.0     6.5      5.0
## 2  2    Q       6.5     5.5     7.0      5.5
## 3  3    H       6.0     5.0     6.5      5.0
## 4  4    E       6.5     5.5     7.0      5.5
## 5  5    T       6.5     5.5     7.0      5.5
```

```
# retireving all rows but 4
ielts[-4, ]
```

```
##    id name listening reading writing speaking
## 1   1    I       6.0     5.0     6.5      5.0
## 2   2    Q       6.5     5.5     7.0      5.5
## 3   3    H       6.0     5.0     6.5      5.0
## 5   5    T       6.5     5.5     7.0      5.5
## 6   6    G       7.0     6.0     7.5      6.0
## 7   7    I       6.0     5.0     6.5      5.0
## 8   8    H       6.5     5.5     7.0      5.5
## 9   9    H       7.0     6.0     7.5      6.0
## 10 10    T       7.0     6.0     7.5      6.0
```

```
# retireving all rows but 4,6, and 9
ielts[-c(4,6,9), ]
```

```
##    id name listening reading writing speaking
## 1   1    I       6.0     5.0     6.5      5.0
## 2   2    Q       6.5     5.5     7.0      5.5
## 3   3    H       6.0     5.0     6.5      5.0
## 5   5    T       6.5     5.5     7.0      5.5
## 7   7    I       6.0     5.0     6.5      5.0
## 8   8    H       6.5     5.5     7.0      5.5
## 10 10    T       7.0     6.0     7.5      6.0
```

```
# retireving all rows but 1 to 5
ielts[-(1:5), ]
```

```
##    id name listening reading writing speaking
## 6   6    G       7.0     6.0     7.5      6.0
## 7   7    I       6.0     5.0     6.5      5.0
## 8   8    H       6.5     5.5     7.0      5.5
## 9   9    H       7.0     6.0     7.5      6.0
## 10 10    T       7.0     6.0     7.5      6.0
```

- Retireving columns using column indices

```
# showing the sample data frame again
ielts
```

```
##    id name listening reading writing speaking
## 1   1    I       6.0     5.0     6.5      5.0
## 2   2    Q       6.5     5.5     7.0      5.5
## 3   3    H       6.0     5.0     6.5      5.0
## 4   4    E       6.5     5.5     7.0      5.5
## 5   5    T       6.5     5.5     7.0      5.5
## 6   6    G       7.0     6.0     7.5      6.0
## 7   7    I       6.0     5.0     6.5      5.0
## 8   8    H       6.5     5.5     7.0      5.5
## 9   9    H       7.0     6.0     7.5      6.0
## 10 10    T       7.0     6.0     7.5      6.0
```

```
# retireving coulmn 2 (name)
ielts[, 2]
```

```
##  [1] I Q H E T G I H H T
## Levels: E G H I Q T
```

```
# retireving coulmns 2 (name) and 3 (listening)
ielts[, c(2,3)]
```

```
##    name listening
## 1     I       6.0
## 2     Q       6.5
## 3     H       6.0
## 4     E       6.5
## 5     T       6.5
## 6     G       7.0
## 7     I       6.0
## 8     H       6.5
## 9     H       7.0
## 10    T       7.0
```

```
# retireving coulmns 3 to 6
ielts[, 3:6]
```

```
##    listening reading writing speaking
## 1        6.0     5.0     6.5      5.0
## 2        6.5     5.5     7.0      5.5
## 3        6.0     5.0     6.5      5.0
## 4        6.5     5.5     7.0      5.5
## 5        6.5     5.5     7.0      5.5
## 6        7.0     6.0     7.5      6.0
## 7        6.0     5.0     6.5      5.0
## 8        6.5     5.5     7.0      5.5
## 9        7.0     6.0     7.5      6.0
## 10       7.0     6.0     7.5      6.0
```

```
# retireving all coulmns but 2 (name)
ielts[, -2]
```

```
##    id listening reading writing speaking
## 1   1       6.0     5.0     6.5      5.0
## 2   2       6.5     5.5     7.0      5.5
## 3   3       6.0     5.0     6.5      5.0
## 4   4       6.5     5.5     7.0      5.5
## 5   5       6.5     5.5     7.0      5.5
## 6   6       7.0     6.0     7.5      6.0
## 7   7       6.0     5.0     6.5      5.0
## 8   8       6.5     5.5     7.0      5.5
## 9   9       7.0     6.0     7.5      6.0
## 10 10       7.0     6.0     7.5      6.0
```

```
# retireving all columns but 2 (name) and 3 (listening)
ielts[, -c(2,3)]
```

```
##    id reading writing speaking
## 1   1     5.0     6.5      5.0
## 2   2     5.5     7.0      5.5
## 3   3     5.0     6.5      5.0
## 4   4     5.5     7.0      5.5
## 5   5     5.5     7.0      5.5
## 6   6     6.0     7.5      6.0
## 7   7     5.0     6.5      5.0
## 8   8     5.5     7.0      5.5
## 9   9     6.0     7.5      6.0
## 10 10     6.0     7.5      6.0
```

```
# retireving all coulmns but 3 to 6
ielts[, -(3:6)]
```

```
##    id name
## 1   1    I
## 2   2    Q
## 3   3    H
## 4   4    E
## 5   5    T
## 6   6    G
## 7   7    I
## 8   8    H
## 9   9    H
## 10 10    T
```

- Retireving columns using column names

```
# showing the sample data frame again
ielts
```

```
##    id name listening reading writing speaking
## 1   1    I       6.0     5.0     6.5      5.0
## 2   2    Q       6.5     5.5     7.0      5.5
## 3   3    H       6.0     5.0     6.5      5.0
## 4   4    E       6.5     5.5     7.0      5.5
## 5   5    T       6.5     5.5     7.0      5.5
## 6   6    G       7.0     6.0     7.5      6.0
## 7   7    I       6.0     5.0     6.5      5.0
## 8   8    H       6.5     5.5     7.0      5.5
## 9   9    H       7.0     6.0     7.5      6.0
## 10 10    T       7.0     6.0     7.5      6.0
```

```
# retireving coulmn "name"
ielts[, "name"]
```

```
##  [1] I Q H E T G I H H T
## Levels: E G H I Q T
```

```
# retireving coulmns "name" and "listening"
ielts[, c("name" , "listening")]
```

```
##    name listening
## 1     I       6.0
## 2     Q       6.5
## 3     H       6.0
## 4     E       6.5
## 5     T       6.5
## 6     G       7.0
## 7     I       6.0
## 8     H       6.5
## 9     H       7.0
## 10    T       7.0
```

- Retireving columns using `<data frame name>$<column name>` (the most convinient way of retrieving a single column)

```
# showing the sample data frame again
ielts
```

```
##    id name listening reading writing speaking
## 1   1    I       6.0     5.0     6.5      5.0
## 2   2    Q       6.5     5.5     7.0      5.5
## 3   3    H       6.0     5.0     6.5      5.0
## 4   4    E       6.5     5.5     7.0      5.5
## 5   5    T       6.5     5.5     7.0      5.5
## 6   6    G       7.0     6.0     7.5      6.0
## 7   7    I       6.0     5.0     6.5      5.0
## 8   8    H       6.5     5.5     7.0      5.5
## 9   9    H       7.0     6.0     7.5      6.0
## 10 10    T       7.0     6.0     7.5      6.0
```

```
# retireving coulmn "name"
ielts$name
```

```
##  [1] I Q H E T G I H H T
## Levels: E G H I Q T
```

```r
# retireving coulmn "listening"
ielts$listening
```

```
##  [1] 6.0 6.5 6.0 6.5 6.5 7.0 6.0 6.5 7.0 7.0
```

```r
# compute mean of writing column
mean(ielts$writing)
```

```
## [1] 7
```

- Retireving data based on conditions

```r
# showing the sample data frame again
ielts
```

```
##    id name listening reading writing speaking
## 1   1    I       6.0     5.0     6.5      5.0
## 2   2    Q       6.5     5.5     7.0      5.5
## 3   3    H       6.0     5.0     6.5      5.0
## 4   4    E       6.5     5.5     7.0      5.5
## 5   5    T       6.5     5.5     7.0      5.5
## 6   6    G       7.0     6.0     7.5      6.0
## 7   7    I       6.0     5.0     6.5      5.0
## 8   8    H       6.5     5.5     7.0      5.5
## 9   9    H       7.0     6.0     7.5      6.0
## 10 10    T       7.0     6.0     7.5      6.0
```

```r
# retireving rows of data where writing is >= 6.5
ielts[ielts$writing >= 6.5 , ]
```

```
##    id name listening reading writing speaking
## 1   1    I       6.0     5.0     6.5      5.0
## 2   2    Q       6.5     5.5     7.0      5.5
## 3   3    H       6.0     5.0     6.5      5.0
## 4   4    E       6.5     5.5     7.0      5.5
## 5   5    T       6.5     5.5     7.0      5.5
## 6   6    G       7.0     6.0     7.5      6.0
## 7   7    I       6.0     5.0     6.5      5.0
## 8   8    H       6.5     5.5     7.0      5.5
## 9   9    H       7.0     6.0     7.5      6.0
## 10 10    T       7.0     6.0     7.5      6.0
```

```r
# retireving rows of data where writing is >= 6.5 and speaking is >=6
ielts[ielts$writing >= 6.5 & ielts$speaking >= 6, ]
```

```
##    id name listening reading writing speaking
## 6   6    G         7       6     7.5        6
## 9   9    H         7       6     7.5        6
## 10 10    T         7       6     7.5        6
```

```r
# retireving rows of data where writing is >= 6.5 or speaking is >=6
ielts[ielts$writing >= 6.5 | ielts$speaking >= 6, ]
```

```
##    id name listening reading writing speaking
## 1   1    I       6.0     5.0     6.5      5.0
## 2   2    Q       6.5     5.5     7.0      5.5
## 3   3    H       6.0     5.0     6.5      5.0
## 4   4    E       6.5     5.5     7.0      5.5
## 5   5    T       6.5     5.5     7.0      5.5
## 6   6    G       7.0     6.0     7.5      6.0
## 7   7    I       6.0     5.0     6.5      5.0
## 8   8    H       6.5     5.5     7.0      5.5
## 9   9    H       7.0     6.0     7.5      6.0
## 10 10    T       7.0     6.0     7.5      6.0
```

```r
# retireving just name column where writing is >= 6.5
ielts[ielts$writing >= 6.5 , "name"]
```

```
##  [1] I Q H E T G I H H T
## Levels: E G H I Q T
```

```r
# or
ielts[ielts$writing >= 6.5 , 2]
```

```
##  [1] I Q H E T G I H H T
## Levels: E G H I Q T
```

```r
# or
ielts[ielts$writing >= 6.5 , ]$name
```

```
##  [1] I Q H E T G I H H T
## Levels: E G H I Q T
```

```r
# retireving id and name columns where speaking > 6
ielts[ielts$speaking > 6 , c("id", "name")]
```

```
## [1] id   name
## <0 rows> (or 0-length row.names)
```

```r
# retireving all data where speaking is >= average of speakings
ielts[ielts$speaking >= mean(ielts$speaking) , ]
```

```
##    id name listening reading writing speaking
## 2   2    Q       6.5     5.5     7.0      5.5
## 4   4    E       6.5     5.5     7.0      5.5
## 5   5    T       6.5     5.5     7.0      5.5
## 6   6    G       7.0     6.0     7.5      6.0
## 8   8    H       6.5     5.5     7.0      5.5
## 9   9    H       7.0     6.0     7.5      6.0
## 10 10    T       7.0     6.0     7.5      6.0
```

2 Input/Output data

- Almost all of the statistical data of the real world projects are saved in files.
- They should be read and loaded to memory to be processed.
- The results of statistical analyses have to be written to files too.

2.1 Reading tabular data

- Tabular data are ideal to be loaded as data frames.
- Columns are separated from each other by some separator characters (whitespace, tab, comma, ...)
- Each line represents a row of data.
- Reading data frames using `read.table()`
  - default separator is whitespace

```
# assuming that the file "ielts.txt" is in the working directory

# read the entire data
ielts2<-read.table("ielts.txt", header=TRUE)
ielts2
```

```
##    id name listening reading writing speaking
## 1   1    N       6.0     5.5     7.0      5.0
## 2   2    Z       7.0     6.5     8.0      6.0
## 3   3    X       6.5     6.0     7.5      5.5
## 4   4    N       6.5     6.0     7.5      5.5
## 5   5    I       6.5     6.0     7.5      5.5
## 6   6    T       6.5     6.0     7.5      5.5
## 7   7    I       7.0     6.5     8.0      6.0
## 8   8    I       6.0     5.5     7.0      5.0
## 9   9    U       7.0     6.5     8.0      6.0
## 10 10    A       7.0     6.5     8.0      6.0
```

```
# read the first 6 rows
ielts3<-read.table("ielts.txt", header=TRUE, nrows=6)
ielts3
```

```
##   id name listening reading writing speaking
## 1  1    N       6.0     5.5     7.0      5.0
## 2  2    Z       7.0     6.5     8.0      6.0
## 3  3    X       6.5     6.0     7.5      5.5
## 4  4    N       6.5     6.0     7.5      5.5
## 5  5    I       6.5     6.0     7.5      5.5
## 6  6    T       6.5     6.0     7.5      5.5
```

- Reading data frames using `read.csv()`
  - csv stands for Comma Separated Values
  - much like `read.table`
  - default separator is comma

```r
# assuming that the file "ielts.csv" is in the working directory

# read the entire data
ielts2<-read.csv("ielts.csv", header=TRUE)
ielts2
```

```
##     X id name listening reading writing speaking
## 1   1  1    K       6.5     7.0     5.5      6.5
## 2   2  2    Y       6.0     6.5     5.0      6.0
## 3   3  3    X       7.0     7.5     6.0      7.0
## 4   4  4    A       6.0     6.5     5.0      6.0
## 5   5  5    Q       6.0     6.5     5.0      6.0
## 6   6  6    X       6.5     7.0     5.5      6.5
## 7   7  7    T       6.0     6.5     5.0      6.0
## 8   8  8    A       6.0     6.5     5.0      6.0
## 9   9  9    J       7.0     7.5     6.0      7.0
## 10 10 10    T       6.5     7.0     5.5      6.5
```

```r
# read the first 6 rows
ielts3<-read.csv("ielts.csv", header=TRUE, nrows=6)
ielts3
```

```
##   X id name listening reading writing speaking
## 1 1  1    K       6.5     7.0     5.5      6.5
## 2 2  2    Y       6.0     6.5     5.0      6.0
## 3 3  3    X       7.0     7.5     6.0      7.0
## 4 4  4    A       6.0     6.5     5.0      6.0
## 5 5  5    Q       6.0     6.5     5.0      6.0
## 6 6  6    X       6.5     7.0     5.5      6.5
```

2.2 Reading text files

- Use `readLines()` to read all or some lines from a text file

```r
# assuming that the file "myfile.txt" is in the working directory

# reading all the file
lines<-readLines("myfile.txt")
lines
```

```
##  [1] "We assume no responsibility for errors or omissions."
##  [2] "Damages resulting from the use of the information contained herein."
##  [3] "Problems specific to the modern age. Problems of the nuclear age. The Victorian age."
##  [4] "It takes ages to cook. I've been waiting for ages."
##  [5] "It's been ages since we last spoke. It's been an age since we last spoke."
##  [6] "Her back went bent with age. This cheese improves with age. This wine improves with age."
##  [7] "His temper hasn't improved with age."
##  [8] "Act you age, please."
##  [9] "He was prosecuted for having sex with a girl who was under age."
## [10] "In this age, it can sometimes seem like every system is connected to every other system."
## [11] "Will you be delivering services or consuming them?"
## [12] "It is a key part of all modern, public-facing applications."
## [13] "This book is here to help you navigate your way along the road ahead."
## [14] "You will see how to devise great solutions."
## [15] "This book has you covered, from technical details to the big picture."
## [16] "PHP has always taken on the mission to solve the web problem."
## [17] "Her voice took on a troubled tone."
```

```
# reading first 4 lines
lines<-readLines("myfile.txt", 4)
lines
```

```
## [1] "We assume no responsibility for errors or omissions."
## [2] "Damages resulting from the use of the information contained herein."
## [3] "Problems specific to the modern age. Problems of the nuclear age. The Victorian age."
## [4] "It takes ages to cook. I’ve been waiting for ages."
```

2.3 Reading web pages

- `readLines()` can be used to read all or some lines from a web page

```
# reading all the web page
lines<-readLines("http://www.google.com")
```

```
## Warning in readLines("http://www.google.com"): incomplete final line found
## on 'http://www.google.com'
```

```
lines
```

```
## [1] "<!doctype html><html itemscope=\"\" itemtype=\"http://schema.org/WebPage\" lang=\"en-IR\"><head
## [2] "function _gjh(){!_gjuc()&&window.google&&google.x&&google.x({id:\"GJH\"},function(){google.nav&
## [3] "if (!iesg){document.f&&document.f.q.focus();document.gbqf&&document.gbqf.q.focus();}"
## [4] "}"
## [5] "})();</script><div id=\"mngb\">   <div id=gbar><nobr><b class=gb1>Search</b> <a class=gb1 href=
## [6] "a.i.Z,window.gbar.elr&&a.i.$(window.gbar.elr()),window.gbar.elc&&window.gbar.elc(a.i.$),a.i.G(!
## [7] "});})();</script> </div> </span><br clear=\"all\" id=\"lgpd\"><div id=\"lga\"><img alt=\"Google
## [8] "</script></div></body></html>"
```

```
# reading first 4 lines
lines<-readLines("http://www.google.com", 4)
lines
```

```
## [1] "<!doctype html><html itemscope=\"\" itemtype=\"http://schema.org/WebPage\" lang=\"en-IR\"><head
## [2] "function _gjh(){!_gjuc()&&window.google&&google.x&&google.x({id:\"GJH\"},function(){google.nav&
## [3] "if (!iesg){document.f&&document.f.q.focus();document.gbqf&&document.gbqf.q.focus();}"
## [4] "}"
```

2.4 Writing tabular data

- Writing data frames using `write.table()`
  - default separator is whitespace

```r
# write the entire data
write.table(ielts, "ielts.data.txt", row.names=FALSE)
```

- Writing data frames using `write.csv()`
  - default separator is comma

```r
# write the entire data
write.table(ielts, "ielts.data.csv", row.names=FALSE)
```

2.5 Writing to text files

- Use `writeLines()` to write text vectors to a text file

```r
lines<-c("line 1", "line 2", "line 3")

# writing to file
writeLines(lines, "myTextFile.txt")

# reading file
readLines("myTextFile.txt")
```

```
## [1] "line 1" "line 2" "line 3"
```

2.6 Output to the screen

- Object's name followed by Enter

```r
# show value of a vector
v<-sample(1:10, 5)
v
```

```
## [1] 4 5 3 2 6
```

```r
# show value of a matrix
m<-matrix(sample(1:10, 9), 3, 3)
m
```

```
##      [,1] [,2] [,3]
## [1,]    5    7    4
## [2,]   10    2    9
## [3,]    1    6    3
```

```r
# show value of a data frame
df<-data.frame(a=sample(1:10, 4), b=sample(1:10, 4),
               c=sample(c("AB", "CD", "EF"), 4, replace=TRUE))
df
```

```
##    a  b  c
## 1  5 10 AB
## 2  7  3 CD
## 3 10  9 EF
## 4  9  7 AB
```

```
# show contents of a function
mySquare<-function(x){
  return(x^2)
}
mySquare
```

```
## function(x){
##   return(x^2)
## }
```

- Using `print()` function

```
# show value of a vector
v<-sample(1:10, 5)
print(v)
```

```
## [1] 1 4 3 9 7
```

```
# show value of a matrix
m<-matrix(sample(1:10, 9), 3, 3)
print(m)
```

```
##      [,1] [,2] [,3]
## [1,]    6    5    7
## [2,]    2    3    4
## [3,]    8   10    9
```

```
# show value of a data frame
df<-data.frame(a=sample(1:10, 4), b=sample(1:10, 4),
               c=sample(c("AB", "CD", "EF"), 4, replace=TRUE))
print(df)
```

```
##    a b  c
## 1  1 4 AB
## 2  9 7 CD
## 3  4 9 EF
## 4 10 5 EF
```

```
# show contents of a function
mySquare<-function(x){
  return(x^2)
}
print(mySquare)
```

```
## function(x){
##   return(x^2)
## }
```

- Using `cat()` function to concatenate and print values
  - `\n` means newline
  - `\t` means tab
  - `\\` means backslash \
  - `\'` means ASCII apostrophe '
  - `\"` means ASCII quotation mark "
  - \\` means ASCII grave accent (backtick) `

```r
item<-50
cat("Item no :", item)
```

```
## Item no : 50
```

```r
result<-matrix(sample(1:10, 9), 3, 3)
cat("The result is \n", result)
```

```
## The result is
##  7 8 5 1 2 10 3 4 6
```

```r
v<-c(sample(1:10), 3)
cat("1:", v[1], "\t 2:", v[2], "\t 3:", v[3])
```

```
## 1: 1      2: 7    3: 10
```

```r
cat("Let me introduce \"Tom Hanks\"")
```

```
## Let me introduce "Tom Hanks"
```