Introduction to R

Saeed Saffari* Saeedeh Malekan†

Fall 2021

Contents

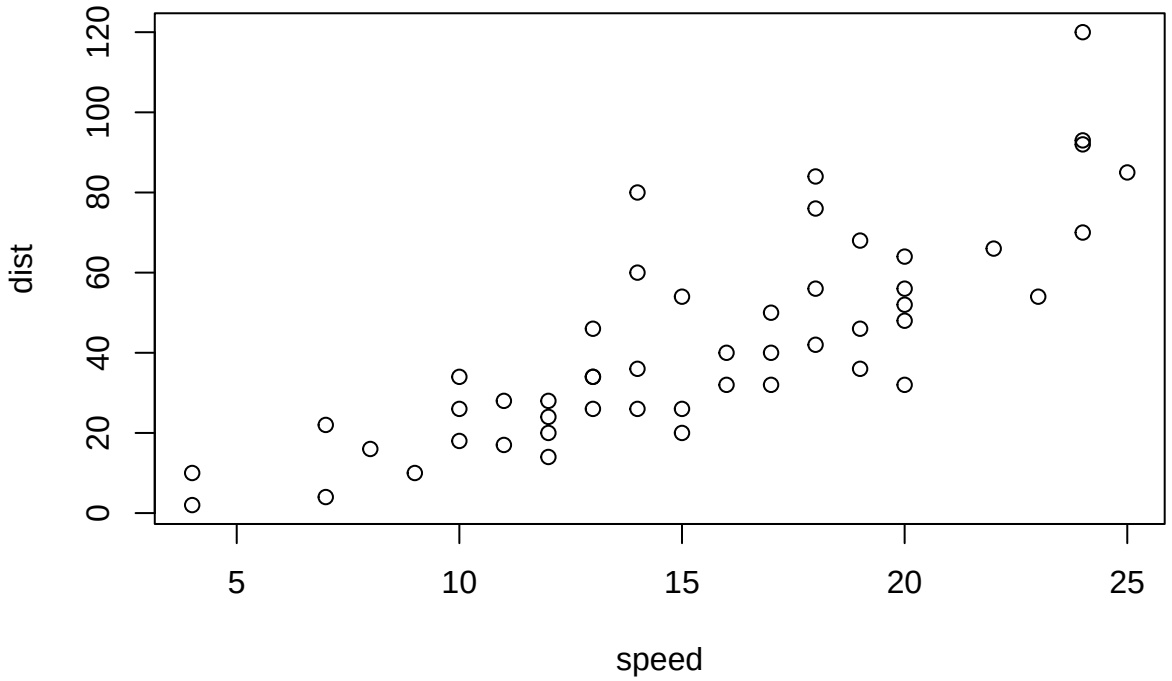
*m.saeed.saffari@ut.ac.ir
†saeedeh.malekan@ut.ac.ir

1 Introduction

This is an R Markdown Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Cmd+Shift+Enter*.

```
plot(cars)
```



Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Cmd+Option+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Cmd+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.

You can download and install packages with `install.packages("The name of package")`.

2 Basic of learning

In this part we talk about basic elements of R programming.

• Note that this is only a brief overview of **R programming**, and before that you should be familiar with the **Python programming language**.

2.1 How to Print

```
print("This is R programming Course")
```

```
## [1] "This is R programming Course"
```

2.2 Arithmetic Operators

Symbol	Task Performed
+	Addition
-	Subtraction
/	division
*	multiplication
**	to the power of
^	to the power of
%%	modulus
%/%	floor division

```
18 + 4
```

```
## [1] 22
```

```
18 - 4
```

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

```
18 * 4
```

```
## [1] 72
```

```
10 / 2
```

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

```
2 ** 3
```

```
## [1] 8
```

```
2 ^ 3
```

```
## [1] 8
```

```
9 ** 0.5
```

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

```
log10(2)
```

```
## [1] 0.30103
```

```
5 + ( 4 - 3 * 2)**3 + 1
```

```
## [1] -2
```

```
14 %% 4
```

```
## [1] 2
```

```
8 %/% 3
```

```
## [1] 2
```

we can save values in variables:

```
x <- 18
y <- 4
z <- x + y
print(z)
```

```
## [1] 22
```

```
z
```

```
## [1] 22
```

```
class(z)
```

```
## [1] "numeric"
```

```
a = 'R Programming'
class(a)
```

```
## [1] "character"
```

2.3 Relational Operators

Symbol	Task Performed
<-	Assignment
=	Assignment
assign()	Assignment
==	True, if it is equal
!=	True, if not equal to
<	less than
>	greater than
<=	less than or equal to
>=	greater than or equal to

```
z <- 10
y = 18
assign('x', 30)
z
```

```
## [1] 10
```

```
y
```

```
## [1] 18
```

```
x
```

```
## [1] 30
```

```
x < y
```

```
## [1] FALSE
```

```
x >= y
```

```
## [1] TRUE
```

```
x != y
```

```
## [1] TRUE
```

```
x == y
```

```
## [1] FALSE
```

you can use below command to get special values:

```
x <- pi
x
```

```
## [1] 3.141593
```

```
x <- letters
x
```

```
##  [1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p" "q" "r" "s"
## [20] "t" "u" "v" "w" "x" "y" "z"
```

```
x <- LETTERS
x
```

```
##  [1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" "L" "M" "N" "O" "P" "Q" "R" "S"
## [20] "T" "U" "V" "W" "X" "Y" "Z"
```

```
x <- month.name
x
```

```
##  [1] "January"   "February"  "March"     "April"     "May"       "June"
##  [7] "July"      "August"    "September" "October"   "November"  "December"
```

```
x <- month.abb
x
```

```
##  [1] "Jan" "Feb" "Mar" "Apr" "May" "Jun" "Jul" "Aug" "Sep" "Oct" "Nov" "Dec"
```

you can write comment with # :

```
# This line is comment.
```

you can creat sequence numbers with below command:

This work like *arange* in numpy pakage in Python

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

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

```
x <- 1:10 * 2
x
```

```
##  [1]  2  4  6  8 10 12 14 16 18 20
```

```
x <- seq(5)
x
```

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

```
x <- seq(from = 1, to = 9)
x
```

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

```
x <- seq(from = 1, to = 9, by = 3)
x
```

```
## [1] 1 4 7
```

```
x <- seq(1, 10, 2)
x
```

```
## [1] 1 3 5 7 9
```

This work like *linsapce* in numpy pakage in Python

```
x <- seq(1, 10, length = 4)
x
```

```
## [1]  1  4  7 10
```

Replicate function:

```
x <- 1:3
x
```

```
## [1] 1 2 3
```

```
y <- rep(x, times = 5)
y
```

```
##  [1] 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3
```

```
y <- rep(x, each = 5)
y
```

```
##  [1] 1 1 1 1 1 2 2 2 2 2 3 3 3 3 3
```

2.4 Loops

2.4.1 if , elif

```
age <- 18#as.integer(readline(prompt('Enter your age: ')))

if (age >= 18){
  print("You are old enough to vote!")
} else {
  print('You can NOT vote yet!')
  print(paste("you can will vote after ", 18 - age, " years."))
}
```

```
## [1] "You are old enough to vote!"
```

```
age <- 10 # as.integer(readline(prompt('Enter your age: ')))

if (age <= 4){
  price = 0
} else if (age <= 16){
  price = 5
```

```
} else {
  price = 10
}

print(paste("your cost is $",price,"."))
```

```
## [1] "your cost is $ 5 ."
```

2.4.2 for loops

```
for (i in 1:5){
  print(i)
}
```

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

2.4.3 while loops

```
i <- 1
while (i < 10){
  print(i)
  i <- i + 1
}
```

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

Use break and next in loops

```
for (i in 1:10){
  if (i == 5){
    break
  }
  print(i)
}
```

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

```
for (i in 1:10){
  if (i == 5){
    next
  }
```

```
  print(i)
}
```

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

2.5 function

```
ave = function(a, b, c){
  summ = a + b + c
  ave = summ / 3
  return( ave)
}

ave(34, 13, -21)
```

```
## [1] 8.666667
```

```
ave = function(a, b , c){
  #a <- as.integer(readline(prompt('Enter your first number: ')))
  #b <- as.integer(readline(prompt('Enter your second number: ')))
  #c <- as.integer(readline(prompt('Enter your third number: ')))
  summ = a + b + c
  ave = summ / 3
  return( ave)
}

ave(12, 13, 14)
```

```
## [1] 13
```

3 Vetors

The most common way to creat vectors is to use function `c()`.

```
x <- c(10.25, 3.5, 8.75, 23.15)
x
```

```
## [1] 10.25  3.50  8.75 23.15
```

```
x <- c('a', "b", 'C')
x
```

```
## [1] "a" "b" "C"
```

```
# all elemets have same type
x <- c(10.25, 3.5, 8.75, 23.15, 'a', "b", 'C')
x
```

```
## [1] "10.25" "3.5"   "8.75"  "23.15" "a"     "b"     "C"
```

```
x <- c(1,2,4,5,6,7)
x
```

```
## [1] 1 2 4 5 6 7
```

```
y <- 1:7
y
```

```
## [1] 1 2 3 4 5 6 7
```

Concat vectors

```
x <- c(10,20,30,40)

y <- c(3.5, 4.75)

z <- c(x, y)
z
```

```
## [1] 10.00 20.00 30.00 40.00  3.50  4.75
```

You can find *lenght* of vectors with *lenght()* function:

```
x <- c(1.5,3.25,8.75,13.15)
x
```

```
## [1]  1.50  3.25  8.75 13.15
```

```
length(x)
```

```
## [1] 4
```

3.1 Vector Indexing

You can indexing vectors exactly like python, but you should know that we start from ***one*** in *R programming*.

```
x <- c(10,45,30,50,35,40,80)
x
```

```
## [1] 10 45 30 50 35 40 80
```

```
x[1]
```

```
## [1] 10
```

```
x[-2]
```

```
## [1] 10 30 50 35 40 80
```

```
x[3:7]
```

```
## [1] 30 50 35 40 80
```

```
x[c(1,3,4)]
```

```
## [1] 10 30 50
```

```
length(x)
```

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

```
x[10]
```

```
## [1] NA
```

```
x[2] <- -8
x
```

```
## [1] 10 -8 30 50 35 40 80
```

```
x[10] <- 9
x
```

```
##  [1] 10 -8 30 50 35 40 80 NA NA  9
```

```
x[-3] <- 6
x
```

```
##  [1]  6  6 30  6  6  6  6  6  6  6
```

```
x <- c(10,45,30,50,35,40,80)
y <- c(TRUE,FALSE,TRUE,FALSE,FALSE,FALSE,TRUE)
x[y]
```

```
## [1] 10 30 80
```

Use for loops for access to elemets of vectors

```
for (i in x){
 print(i)
}
```

```
## [1] 10
## [1] 45
## [1] 30
## [1] 50
## [1] 35
## [1] 40
## [1] 80
```

Use index:

```
for (i in 1:length(x)){
 print(x[i])
}
```

```
## [1] 10
## [1] 45
```

```
## [1] 30
## [1] 50
## [1] 35
## [1] 40
## [1] 80
```

```
for (i in 1:10){
 print(x[i])
}
```

```
## [1] 10
## [1] 45
## [1] 30
## [1] 50
## [1] 35
## [1] 40
## [1] 80
## [1] NA
## [1] NA
## [1] NA
```

3.2 Matching Operator

```
x <- c(10,45,30,50,35,40,80)


35 %in% x
```

```
## [1] TRUE
```

```
37 %in% x
```

```
## [1] FALSE
```

```
y <- c(30, 37, 45)

y %in% x
```

```
## [1]  TRUE FALSE  TRUE
```

3.3 Vector Arthmetic's

```
x <- c(10,45,30,50)
x
```

```
## [1] 10 45 30 50
```

```
x + 2
```

```
## [1] 12 47 32 52
```

```
x <- x * 2
x
```

```
## [1]  20  90  60 100
```

```
sqrt(x)
```

```
## [1]  4.472136  9.486833  7.745967 10.000000
```

```r
x <- c(10,45,30,50)
y <- c(5,1,2,4)

x + y
```

```
## [1] 15 46 32 54
```

```r
z <- c(10,20,30)
x + z
```

```
## Warning in x + z: longer object length is not a multiple of shorter object
## length
```

```
## [1] 20 65 60 60
```

3.4 Vector Methods

```r
x <- c(10,45,30,50)
length(x)
```

```
## [1] 4
```

```r
sum(x)
```

```
## [1] 135
```

```r
prod(x)
```

```
## [1] 675000
```

```r
rev(x)
```

```
## [1] 50 30 45 10
```

```r
sort(x)
```

```
## [1] 10 30 45 50
```

```r
sort(x, decreasing =  TRUE)
```

```
## [1] 50 45 30 10
```

```r
x <- c(10,45,30,50)
y <- c(5,1,4,3)

x %*% y
```

```
##      [,1]
## [1,]  365
```

```r
crossprod(x,y)
```

```
##      [,1]
## [1,]  365
```

```r
x %o% y
```

```
##      [,1] [,2] [,3] [,4]
## [1,]   50   10   40   30
## [2,]  225   45  180  135
## [3,]  150   30  120   90
```

```
## [4,]  250   50  200  150
```

```
tcrossprod(x,y)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]   50   10   40   30
## [2,]  225   45  180  135
## [3,]  150   30  120   90
## [4,]  250   50  200  150
```

3.5 Logical Vector

```
x <- c(10,45,30,50,35)
x
```

```
## [1] 10 45 30 50 35
```

```
y <- x > 30 & x < 50
y
```

```
## [1] FALSE  TRUE FALSE FALSE  TRUE
```

```
x[y]
```

```
## [1] 45 35
```

```
x <- c(10,45,30,50,35)
y <- c(20,15,25,65,5)

x < y
```

```
## [1]  TRUE FALSE FALSE  TRUE FALSE
```

```
x[x <y]
```

```
## [1] 10 50
```

```
x <- c(10,45,30,50,35)
which(x>30)
```

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

```
x[which(x>30)]
```

```
## [1] 45 50 35
```

3.6 Factors

- Used to represent categorical data
- Treated as integer vector, having a label
- Factors are self describing

```
x <- c('Male', "Female", "Male", 'Male', "Female")
x
```

```
## [1] "Male"   "Female" "Male"   "Male"   "Female"
```

```
x <- factor(c('Male', "Female", "Male", 'Male', "Female"))
x
```

```
## [1] Male   Female Male   Male   Female
```

```
## Levels: Female Male
```

```
table(x)
```

```
## x
## Female   Male
##      2      3
```

3.7 Mathematical Function in R

```
x <- c(4.258, -3.853, 5.457, 7.504)
abs(x)
```

```
## [1] 4.258 3.853 5.457 7.504
```

```
ceiling(x)  #next integer
```

```
## [1]  5 -3  6  8
```

```
floor(x)   #samller integer
```

```
## [1]  4 -4  5  7
```

```
round(x)
```

```
## [1]  4 -4  5  8
```

```
round(x, digits = 2)
```

```
## [1]  4.26 -3.85  5.46  7.50
```

```
x <- c(16,36,30,81,25)
sqrt(x)
```

```
## [1] 4.000000 6.000000 5.477226 9.000000 5.000000
```

```
log(x)
```

```
## [1] 2.772589 3.583519 3.401197 4.394449 3.218876
```

```
log(x, base = 2)
```

```
## [1] 4.000000 5.169925 4.906891 6.339850 4.643856
```

```
log(x, base = 10)
```

```
## [1] 1.204120 1.556303 1.477121 1.908485 1.397940
```

```
log10(x)
```

```
## [1] 1.204120 1.556303 1.477121 1.908485 1.397940
```

```
x <- c(4,3,6)
factorial(x)
```

```
## [1]  24   6 720
```

3.8 Random Number in R

```
x <- rnorm(10)  # mean = 0 and std = 1
x
```

```
##  [1] -0.20226107  1.39909813 -1.22282640 -0.44736551 -0.87910725  1.48607959
##  [7]  0.48903069 -1.04173275 -0.03726971  1.12969814
```

```
x <- rnorm(10, mean = 0, sd = 1)

mean(x)
```

```
## [1] -0.3378218
```

```
sd(x)
```

```
## [1] 1.175754
```

4 Matrix

4.1 Creat Matrix

Matrix are 2-dimentsional vectors and Dimensianal attribute is of lenght 2 (rows and columns). We should to know that Matrix contain elemts of same type.

```
m <- matrix(nrow = 2, ncol = 3)
m
```

```
##      [,1] [,2] [,3]
## [1,]   NA   NA   NA
## [2,]   NA   NA   NA
```

```
dim(m)    # dimension
```

```
## [1] 2 3
```

```
dim(m)[1]
```

```
## [1] 2
```

```
m <- matrix(c(1,2,3,4,5,6))
m
```

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

```
m <- matrix(c(1,2,3,4,5,6), nrow = 2,ncol = 3)
m
```

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

```
m <- matrix(c(1,2,3,4,5,6), nrow = 2,ncol = 3, byrow = TRUE)
m
```

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

```
m <- matrix(seq(from = 1, to = 40, by = 2), nrow = 4, ncol = 5, byrow = TRUE)
m
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    3    5    7    9
## [2,]   11   13   15   17   19
## [3,]   21   23   25   27   29
## [4,]   31   33   35   37   39
```

```
dim(m)
```

```
## [1] 4 5
```

```
nrow(m)
```

```
## [1] 4
```

```
ncol(m)
```

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

```
length(m)
```

```
## [1] 20
```

4.2 Matrix diag

like `numpy.full` in python

```
m <- matrix(4,3,3)
m
```

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

like `numpy.diag` in python

```
m <- diag(1,3,3)
m
```

```
##      [,1] [,2] [,3]
## [1,]    1    0    0
## [2,]    0    1    0
## [3,]    0    0    1
```

```
m <- diag(4)
m
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    0    1    0    0
## [3,]    0    0    1    0
## [4,]    0    0    0    1
```

```
m <- diag(1:5)
m
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    0    0    0    0
## [2,]    0    2    0    0    0
## [3,]    0    0    3    0    0
## [4,]    0    0    0    4    0
## [5,]    0    0    0    0    5
```

for find the elements of diagonal of matrix:

```
m <- matrix(seq(from = 1, to = 40, by = 2), nrow = 4, ncol = 5, byrow = TRUE)
m
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    3    5    7    9
## [2,]   11   13   15   17   19
## [3,]   21   23   25   27   29
## [4,]   31   33   35   37   39
```

```
diag(m)
```

```
## [1]  1 13 25 37
```

4.3 Matrix: Naming Rows & Columns

```
m <- matrix(c(2,3,4,0,1,2,-1,-3,5), nrow = 3, ncol = 3, byrow = TRUE)
m
```

```
##      [,1] [,2] [,3]
## [1,]    2    3    4
## [2,]    0    1    2
## [3,]   -1   -3    5
```

```
rownames(m) <- c(1,2,3)
colnames(m) <- c("A", "B", 'C')
m
```

```
##    A  B C
## 1  2  3 4
## 2  0  1 2
## 3 -1 -3 5
```

4.4 Matrix Indexing

Indexing in R programming is similar to Python.

```
m <- matrix(c(2,3,4,0,1,2,-1,-2,5,8,6,-3), nrow = 4, ncol = 3, byrow = TRUE)
m
```

```
##      [,1] [,2] [,3]
## [1,]    2    3    4
## [2,]    0    1    2
## [3,]   -1   -2    5
## [4,]    8    6   -3
```

```
m[1,2]
```

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

```
m[1,]  # for get single row
```

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

```
m[,2]  # for get single column
```

```
## [1]  3  1 -2  6
```

```
m[,1:2]
```

```
##      [,1] [,2]
## [1,]    2    3
## [2,]    0    1
## [3,]   -1   -2
## [4,]    8    6
```

```
m[1:3,1:2]
```

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

```
## [2,]    0    1
## [3,]   -1   -2
```

```
m[,c(1,3)]
```

```
##      [,1] [,2]
## [1,]    2    4
## [2,]    0    2
## [3,]   -1    5
## [4,]    8   -3
```

```
m[, -2]  # every columns except 2
```

```
##      [,1] [,2]
## [1,]    2    4
## [2,]    0    2
## [3,]   -1    5
## [4,]    8   -3
```

You can change values in matrix as like as Python.

```
m <- matrix(c(2,3,4,0,1,2,-1,-2,5,8,6,-3), nrow = 4, ncol = 3, byrow = TRUE)
m
```

```
##      [,1] [,2] [,3]
## [1,]    2    3    4
## [2,]    0    1    2
## [3,]   -1   -2    5
## [4,]    8    6   -3
```

```
m[2,3] = 9
m
```

```
##      [,1] [,2] [,3]
## [1,]    2    3    4
## [2,]    0    1    9
## [3,]   -1   -2    5
## [4,]    8    6   -3
```

4.5 Matrix: `rbine()` and `cbind()` functions

You can combine matrices with `rbine()` and `cbind()` functions.

At first, we want to combine the matrices from the row.

```
A <- matrix(c(2,3,4,0,1,2,-1,-2,5), nrow = 3, ncol = 3, byrow = TRUE)
A
```

```
##      [,1] [,2] [,3]
## [1,]    2    3    4
## [2,]    0    1    2
## [3,]   -1   -2    5
```

```
B <- rbind(A, c(10,11,12))
B
```

```
##      [,1] [,2] [,3]
## [1,]    2    3    4
## [2,]    0    1    2
## [3,]   -1   -2    5
```

```
## [4,]   10   11   12
```

```
B <- matrix(1:6, nrow = 2, ncol = 3, byrow = TRUE)
B
```

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

```
C <- rbind(A,B)
C
```

```
##      [,1] [,2] [,3]
## [1,]    2    3    4
## [2,]    0    1    2
## [3,]   -1   -2    5
## [4,]    1    2    3
## [5,]    4    5    6
```

After that, we want to combine the matrices from the columns.

```
A <- matrix(c(2,3,4,0,1,2,-1,-2,5), nrow = 3, ncol = 3, byrow = TRUE)
A
```

```
##      [,1] [,2] [,3]
## [1,]    2    3    4
## [2,]    0    1    2
## [3,]   -1   -2    5
```

```
B <- cbind(A, c(10,11,12))
B
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    2    3    4   10
## [2,]    0    1    2   11
## [3,]   -1   -2    5   12
```

```
B <- matrix(1:6, nrow = 3, ncol = 2)
B
```

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

```
C <- cbind(A,B)
C
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    2    3    4    1    4
## [2,]    0    1    2    2    5
## [3,]   -1   -2    5    3    6
```

Relational Operators in Matrics:

```
A <- matrix(c(1,2,3,4,5,6,8,9,1) , nrow=3, ncol=3, byrow=TRUE)
B <- matrix(c(3,1,2,4,2,1,5,1,2), nrow=3, ncol=3, byrow=TRUE)

A
```

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

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

```
B
```

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

```
A + B
```

```
##      [,1] [,2] [,3]
## [1,]    4    3    5
## [2,]    8    7    7
## [3,]   13   10    3
```

```
A - B
```

```
##      [,1] [,2] [,3]
## [1,]   -2    1    1
## [2,]    0    3    5
## [3,]    3    8   -1
```

```
A * B
```

```
##      [,1] [,2] [,3]
## [1,]    3    2    6
## [2,]   16   10    6
## [3,]   40    9    2
```

```
A / B
```

```
##           [,1] [,2] [,3]
## [1,] 0.3333333  2.0  1.5
## [2,] 1.0000000  2.5  6.0
## [3,] 1.6000000  9.0  0.5
```

Like `numpy.transpose()` or `.T` in python

```
A <- matrix(c(1,2,3,4,5,6,8,9,1,4,2,3) , nrow=3, ncol=4, byrow=TRUE)
A
```

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

```
t(A)
```

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

4.6 Matrix Specific Functions

```
A <- matrix(1:9,3,3)
A
```

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

```
rowSums(A)
```

```
## [1] 12 15 18
```

```
colSums(A)
```

```
## [1]  6 15 24
```

```
rowMeans(A)
```

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

```
colMeans(A)
```

```
## [1] 2 5 8
```

5 Lists

5.1 Creat list

`Lists` are also collecting of data and another kind of data storage. Lists can contain elemnts of any type of R object and these elements of list don't need be same type. You can creat list by using `list()` function.

```
x <- list(10, 'Saeed', TRUE)
x
```

```
## [[1]]
## [1] 10
##
## [[2]]
## [1] "Saeed"
##
## [[3]]
## [1] TRUE
```

Creat list with vectors

```
classno <- c(101,102,103)
name <- c("Mahshid", "Saeed", "Sara")
scores <- c(98.45, 45.65, 78.79)
students <- list(classno, name, scores)
students
```

```
## [[1]]
## [1] 101 102 103
##
## [[2]]
## [1] "Mahshid" "Saeed"   "Sara"
##
## [[3]]
## [1] 98.45 45.65 78.79
```

```
students[1]
```

```
## [[1]]
## [1] 101 102 103
```

```
students[[1]]
```

```
## [1] 101 102 103
```

```
students[[1]][2]
```

```
## [1] 102
```

```
students[[1]][2] = 109
students
```

```
## [[1]]
## [1] 101 109 103
##
## [[2]]
## [1] "Mahshid" "Saeed"   "Sara"
##
## [[3]]
## [1] 98.45 45.65 78.79
```

5.2 List subset Operator

```
classno <- c(101,102,103)
name <- c("Mahshid", "Saeed", "Sara")
scores <- c(98.45, 45.65, 78.79)
students <- list("id" = classno, "name" = name, "marks" = scores)
students
```

```
## $id
## [1] 101 102 103
##
## $name
## [1] "Mahshid" "Saeed"   "Sara"
##
## $marks
## [1] 98.45 45.65 78.79
```

```
students$id
```

```
## [1] 101 102 103
```

```
students$name
```

```
## [1] "Mahshid" "Saeed"   "Sara"
```

```
students$marks
```

```
## [1] 98.45 45.65 78.79
```

```
students[c(1,3)]
```

```
## $id
## [1] 101 102 103
##
## $marks
## [1] 98.45 45.65 78.79
```

```
students[c("id", "marks")]
```

```
## $id
## [1] 101 102 103
##
## $marks
## [1] 98.45 45.65 78.79
```

```
students$name[1]
```

```
## [1] "Mahshid"
```

5.3 Lists Concatenation

```
classno <- c(101,102,103)
name <- c("Mahshid", "Saeed", "Sara")
scores <- c(98.45, 45.65, 78.79)
students <- list("id" = classno, "name" = name, "marks" = scores)
students
```

```
## $id
## [1] 101 102 103
```

```
##
## $name
## [1] "Mahshid" "Saeed"   "Sara"
##
## $marks
## [1] 98.45 45.65 78.79
```

```
age <- list(c(19,20,18))
students <- c(students,age)
students
```

```
## $id
## [1] 101 102 103
##
## $name
## [1] "Mahshid" "Saeed"   "Sara"
##
## $marks
## [1] 98.45 45.65 78.79
##
## [[4]]
## [1] 19 20 18
```

6 Dataframe

Dataframes are objects in R and used to store tabular data. Unlike a matrix in data frame each column can contain different modes of data. The first column can be numeric while the second column can be character and third column can be logical. It is a list of vectors of equal length. Dataframe can be created using `data.frame()` function or imported from various file types.

- 'read.table()"
- 'read.csv()"

6.1 Creating Dataframes

```
id <- c(101,102,103)
names <- c("Mahshid", "Saeed", "Sara")
scores <- c(98.45, 45.65, 78.79)
students <- data.frame(id, names, scores)
students
```

```
##    id   names scores
## 1 101 Mahshid  98.45
## 2 102   Saeed  45.65
## 3 103    Sara  78.79
```

```
slist <- list(id,names,scores)
slist
```

```
## [[1]]
## [1] 101 102 103
##
## [[2]]
## [1] "Mahshid" "Saeed"   "Sara"
##
## [[3]]
## [1] 98.45 45.65 78.79
```

6.2 Dataframes Indexing

```
id <- c(101,102,103)
names <- c("Mahshid", "Saeed", "Sara")
scores <- c(98.45, 45.65, 78.79)
students <- data.frame(id, names, scores)
students
```

```
##    id   names scores
## 1 101 Mahshid  98.45
## 2 102   Saeed  45.65
## 3 103    Sara  78.79
```

```
students[1,]
```

```
##    id   names scores
## 1 101 Mahshid  98.45
```

```
students[2:3,]
```

```
##    id names scores
## 2 102 Saeed  45.65
```

```
## 3 103  Sara  78.79
```

```
students[,1]
```

```
## [1] 101 102 103
```

```
students[,2:3]
```

```
##     names scores
## 1 Mahshid  98.45
## 2   Saeed  45.65
## 3    Sara  78.79
```

```
students[c(1,3), c(2,3)]
```

```
##     names scores
## 1 Mahshid  98.45
## 3    Sara  78.79
```

```
students[2,3]
```

```
## [1] 45.65
```

```
students[[3]]
```

```
## [1] 98.45 45.65 78.79
```

```
students[[3]][2]
```

```
## [1] 45.65
```

```
students[[3]][2] = 87.23
students
```

```
##    id   names scores
## 1 101 Mahshid  98.45
## 2 102   Saeed  87.23
## 3 103    Sara  78.79
```

```
students$names
```

```
## [1] "Mahshid" "Saeed"   "Sara"
```

```
students$names[2]
```

```
## [1] "Saeed"
```

6.3 Dataframes `subset()` function

```
id <- c(101,102,103)
names <- c("Mahshid", "Saeed", "Sara")
scores <- c(98.45, 45.65, 78.79)
students <- data.frame(id, names, scores)
students
```

```
##    id   names scores
## 1 101 Mahshid  98.45
## 2 102   Saeed  45.65
## 3 103    Sara  78.79
```

```
report <- subset(students, scores < 80)
report
```

```
##    id names scores
## 2 102 Saeed  45.65
## 3 103  Sara  78.79
```

```
report <- subset(students, scores < 80 & id < 103)
report
```

```
##    id names scores
## 2 102 Saeed  45.65
```

```
report <- subset(students, scores < 80, select = c(names))
report
```

```
##   names
## 2 Saeed
## 3  Sara
```

```
report <- subset(students, scores < 80, select = c(names, scores))
report
```

```
##   names scores
## 2 Saeed  45.65
## 3  Sara  78.79
```

```
report <- subset(students, scores < 80, select = -names)
report
```

```
##    id scores
## 2 102  45.65
## 3 103  78.79
```

6.4 Dataframes `rbine()` and `cbind()`

```
id <- c(101,102,103)
names <- c("Mahshid", "Saeed", "Sara")
scores <- c(98.45, 45.65, 78.79)
students <- data.frame(id, names, scores)
students
```

```
##    id   names scores
## 1 101 Mahshid  98.45
## 2 102   Saeed  45.65
## 3 103    Sara  78.79
```

add rows

```
students <- rbind(students, data.frame(id = 104, names = 'Mohammad', scores = 68.57))
students
```

```
##    id    names scores
## 1 101  Mahshid  98.45
## 2 102    Saeed  45.65
## 3 103     Sara  78.79
## 4 104 Mohammad  68.57
```

add columns

```
students <- cbind(students, age = c(18,24,19,26))
students
```

```
##    id    names scores age
## 1 101  Mahshid  98.45  18
## 2 102    Saeed  45.65  24
## 3 103     Sara  78.79  19
## 4 104 Mohammad  68.57  26
```

6.5 Dataframes `edit()` functuion

```
id <- c(101,102,103)
names <- c("Mahshid", "Saeed", "Sara")
scores <- c(98.45, 45.65, 78.79)
students <- data.frame(id, names, scores)
students
```

```
##    id   names scores
## 1 101 Mahshid  98.45
## 2 102   Saeed  45.65
## 3 103    Sara  78.79
```

```
#studentstable <- edit(students)
#studentstable
```

6.6 Saving data in csv

```
id <- c(101,102,103)
names <- c("Mahshid", "Saeed", "Sara")
scores <- c(98.45, 45.65, 78.79)
students <- data.frame(id, names, scores)
students
```

```
##    id   names scores
## 1 101 Mahshid  98.45
## 2 102   Saeed  45.65
## 3 103    Sara  78.79
```

```
write.csv(students, file = 'Scoring.csv')
```

6.7 Missing Data

In this part we find out how handle a missing data like `NA`.

This function is like `.isnull()` in python programming.

```
x <- c(10,4,NA,7,15,NaN)
x
```

```
## [1]  10   4  NA   7  15 NaN
```

```
is.na(x)
```

```
## [1] FALSE FALSE  TRUE FALSE FALSE  TRUE
```

```
is.nan(x)
```

```
## [1] FALSE FALSE FALSE FALSE FALSE  TRUE
```

Remove missing values

```r
x <- c(10,4,NA,7,15,NaN)
x
```

```
## [1]  10   4  NA   7  15 NaN
```

```r
is.na(x)
```

```
## [1] FALSE FALSE  TRUE FALSE FALSE  TRUE
```

```r
y <- is.na(x)
y
```

```
## [1] FALSE FALSE  TRUE FALSE FALSE  TRUE
```

```r
x[!y]
```

```
## [1] 10  4  7 15
```

```r
x[!is.na(x)]
```

```
## [1] 10  4  7 15
```

```r
x_2 = x[!is.na(x)]
x_2
```

```
## [1] 10  4  7 15
```

```r
id <- c(101,102,103,104,105)
temperature <- c(25.8,34.2,NA,27.4,20.5)
wind <- c(78,59,63,40,68)
humidity <- c(25,45,85,NA,61)

weather <- data.frame(id,temperature, wind, humidity)
weather
```

```
##    id temperature wind humidity
## 1 101        25.8   78       25
## 2 102        34.2   59       45
## 3 103          NA   63       85
## 4 104        27.4   40       NA
## 5 105        20.5   68       61
```

```r
weatherNA <- complete.cases(weather)
weatherNA
```

```
## [1]  TRUE  TRUE FALSE FALSE  TRUE
```

```r
weather[weatherNA,]
```

```
##    id temperature wind humidity
## 1 101        25.8   78       25
## 2 102        34.2   59       45
## 5 105        20.5   68       61
```

6.8 Dataframes Importing Data from CSV Files

Data is imported in to dataframes using: `read.csv()`

`read.csv()` arguments:

- **file:** name of the file (mandatory argument)
- **header:** logical value (default is false)

- **sep:** separator (default is comma (,))

```
df <- read.csv("sample.csv")
df
```

```
##     X                state abb        region population total
## 1   0              Alabama  AL         South    4779736   135
## 2   1               Alaska  AK          West     710231    19
## 3   2              Arizona  AZ          West    6392017   232
## 4   3             Arkansas  AR         South    2915918    93
## 5   4           California  CA          West   37253956  1257
## 6   5             Colorado  CO          West    5029196    65
## 7   6          Connecticut  CT     Northeast    3574097    97
## 8   7             Delaware  DE         South     897934    38
## 9   8 District of Columbia  DC         South     601723    99
## 10  9              Florida  FL         South   19687653   669
## 11 10              Georgia  GA         South    9920000   376
## 12 11               Hawaii  HI          West    1360301     7
## 13 12                Idaho  ID          West    1567582    12
## 14 13             Illinois  IL North Central   12830632   364
## 15 14              Indiana  IN North Central    6483802   142
## 16 15                 Iowa  IA North Central    3046355    21
## 17 16               Kansas  KS North Central    2853118    63
## 18 17             Kentucky  KY         South    4339367   116
## 19 18            Louisiana  LA         South    4533372   351
## 20 19                Maine  ME     Northeast    1328361    11
## 21 20             Maryland  MD         South    5773552   293
## 22 21        Massachusetts  MA     Northeast    6547629   118
## 23 22             Michigan  MI North Central    9883640   413
## 24 23            Minnesota  MN North Central    5303925    53
## 25 24          Mississippi  MS         South    2967297   120
## 26 25             Missouri  MO North Central    5988927   321
## 27 26              Montana  MT          West     989415    12
## 28 27             Nebraska  NE North Central    1826341    32
## 29 28               Nevada  NV          West    2700551    84
## 30 29        New Hampshire  NH     Northeast    1316470     5
## 31 30           New Jersey  NJ     Northeast    8791894   246
## 32 31           New Mexico  NM          West    2059179    67
## 33 32             New York  NY     Northeast   19378102   517
## 34 33       North Carolina  NC         South    9535483   286
## 35 34         North Dakota  ND North Central     672591     4
## 36 35                 Ohio  OH North Central   11536504   310
## 37 36             Oklahoma  OK         South    3751351   111
## 38 37               Oregon  OR          West    3831074    36
## 39 38         Pennsylvania  PA     Northeast   12702379   457
## 40 39         Rhode Island  RI     Northeast    1052567    16
## 41 40       South Carolina  SC         South    4625364   207
## 42 41         South Dakota  SD North Central     814180     8
## 43 42            Tennessee  TN         South    6346105   219
## 44 43                Texas  TX         South   25145561   805
## 45 44                 Utah  UT          West    2763885    22
## 46 45              Vermont  VT     Northeast     625741     2
## 47 46             Virginia  VA         South    8001024   250
## 48 47           Washington  WA          West    6724540    93
## 49 48        West Virginia  WV         South    1852994    27
```

```
## 50 49           Wisconsin  WI North Central    5686986    97
## 51 50             Wyoming  WY          West     563626     5
```

7 Dplyr Package

You can download and install packages with install.packages("The name of package"). In this case, run `install.packages('dplyr')` to download and install `dplyr` package.

Also, you can import packages in R with `library()` function.

```
# install.packages("dplyr")

library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(dplyr)
df <- read.csv("murders.csv")
df
```

```
##     X          state abb        region population PopulationDensity murders
## 1   0        Alabama  AL         South    4779736              94.65     199
## 2   1        Arizona  AZ          West    6392017              57.05     352
## 3   2     California  CA          West   37253956             244.20    1811
## 4   3       Colorado  CO          West    5029196              49.33     117
## 5   4    Connecticut  CT     Northeast    3574097             741.40     131
## 6   5        Florida  FL         South   19687653             360.20     987
## 7   6        Georgia  GA         South    9920000             172.50     527
## 8   7       Illinois  IL North Central   12830632             231.90     453
## 9   8        Indiana  IN North Central    6483802             182.50     198
## 10  9       Kentucky  KY         South    4339367             110.00     180
## 11 10      Louisiana  LA         South    4533372             105.00     437
## 12 11       Maryland  MD         South    5773552             606.20     424
## 13 12  Massachusetts  MA     Northeast    6547629             852.10     209
## 14 13       Michigan  MI North Central    9883640             174.80     558
## 15 14       Missouri  MO North Central    5988927              87.26     419
## 16 15     New Jersey  NJ     Northeast    8791894            1189.00     363
## 17 16       New York  NY     Northeast   19378102             415.30     860
## 18 17 North Carolina  NC         South    9535483             200.60     445
## 19 18           Ohio  OH North Central   11536504             282.50     460
## 20 19       Oklahoma  OK         South    3751351              55.22     188
## 21 20   Pennsylvania  PA     Northeast   12702379             285.30     646
## 22 21      Tennessee  TN         South    6346105             156.60     356
## 23 22          Texas  TX         South   25145561              98.07    1246
## 24 23       Virginia  VA         South    8001024             207.30     369
## 25 24      Wisconsin  WI North Central    5686986             105.20     151
##    gunmurders gunownership
## 1         135        0.517
## 2         232        0.311
## 3        1257        0.213
## 4          65        0.347
```

```
## 5          97        0.167
## 6         669        0.245
## 7         376        0.403
## 8         364        0.202
## 9         142        0.391
## 10        116        0.477
## 11        351        0.441
## 12        293        0.213
## 13        118        0.126
## 14        413        0.384
## 15        321        0.417
## 16        246        0.123
## 17        517        0.180
## 18        286        0.413
## 19        310        0.324
## 20        111        0.429
## 21        457        0.347
## 22        219        0.439
## 23        805        0.359
## 24        250        0.351
## 25         97        0.444
```

you can see head or tail of data with below function. It's work like `.head()` and `.tail()` in Pandas package in python.

```
head(df,5)
```

```
##   X       state abb    region population PopulationDensity murders gunmurders
## 1 0     Alabama  AL     South    4779736             94.65     199        135
## 2 1     Arizona  AZ      West    6392017             57.05     352        232
## 3 2  California  CA      West   37253956            244.20    1811       1257
## 4 3    Colorado  CO      West    5029196             49.33     117         65
## 5 4 Connecticut  CT Northeast    3574097            741.40     131         97
##   gunownership
## 1        0.517
## 2        0.311
## 3        0.213
## 4        0.347
## 5        0.167
```

```
tail(df,5)
```

```
##     X        state abb        region population PopulationDensity murders
## 21 20 Pennsylvania  PA     Northeast   12702379            285.30     646
## 22 21    Tennessee  TN         South    6346105            156.60     356
## 23 22        Texas  TX         South   25145561             98.07    1246
## 24 23     Virginia  VA         South    8001024            207.30     369
## 25 24    Wisconsin  WI North Central    5686986            105.20     151
##    gunmurders gunownership
## 21        457        0.347
## 22        219        0.439
## 23        805        0.359
## 24        250        0.351
## 25         97        0.444
```

like `.shape` in pandas package in python

```
dim(df)
```

```
## [1] 25  9
```

like `.describe()` in pandas package in python for understand structure of data:

```
str(df)
```

```
## 'data.frame':    25 obs. of  9 variables:
##  $ X                : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ state            : chr  "Alabama" "Arizona" "California" "Colorado" ...
##  $ abb              : chr  "AL" "AZ" "CA" "CO" ...
##  $ region           : chr  "South" "West" "West" "West" ...
##  $ population       : int  4779736 6392017 37253956 5029196 3574097 19687653 9920000 12830632 648380
##  $ PopulationDensity: num  94.7 57 244.2 49.3 741.4 ...
##  $ murders          : int  199 352 1811 117 131 987 527 453 198 180 ...
##  $ gunmurders       : int  135 232 1257 65 97 669 376 364 142 116 ...
##  $ gunownership     : num  0.517 0.311 0.213 0.347 0.167 0.245 0.403 0.202 0.391 0.477 ...
```

7.1 dplyr `select()` function

Select special columns

Select with number of columns:

```
df[c(2,4,5)]
```

```
##             state        region population
## 1         Alabama         South    4779736
## 2         Arizona          West    6392017
## 3      California          West   37253956
## 4        Colorado          West    5029196
## 5     Connecticut     Northeast    3574097
## 6         Florida         South   19687653
## 7         Georgia         South    9920000
## 8        Illinois North Central   12830632
## 9         Indiana North Central    6483802
## 10       Kentucky         South    4339367
## 11      Louisiana         South    4533372
## 12       Maryland         South    5773552
## 13  Massachusetts     Northeast    6547629
## 14       Michigan North Central    9883640
## 15       Missouri North Central    5988927
## 16     New Jersey     Northeast    8791894
## 17       New York     Northeast   19378102
## 18 North Carolina         South    9535483
## 19           Ohio North Central   11536504
## 20       Oklahoma         South    3751351
## 21   Pennsylvania     Northeast   12702379
## 22      Tennessee         South    6346105
## 23          Texas         South   25145561
## 24       Virginia         South    8001024
## 25      Wisconsin North Central    5686986
```

Select with name of columns:

```
df[c('state', "population", "murders")]
```

```
##             state population murders
## 1         Alabama    4779736     199
## 2         Arizona    6392017     352
## 3      California   37253956    1811
## 4        Colorado    5029196     117
## 5     Connecticut    3574097     131
## 6         Florida   19687653     987
## 7         Georgia    9920000     527
## 8        Illinois   12830632     453
## 9         Indiana    6483802     198
## 10       Kentucky    4339367     180
## 11      Louisiana    4533372     437
## 12       Maryland    5773552     424
## 13  Massachusetts    6547629     209
## 14       Michigan    9883640     558
## 15       Missouri    5988927     419
## 16     New Jersey    8791894     363
## 17       New York   19378102     860
## 18 North Carolina    9535483     445
## 19           Ohio   11536504     460
## 20       Oklahoma    3751351     188
## 21   Pennsylvania   12702379     646
## 22      Tennessee    6346105     356
## 23          Texas   25145561    1246
## 24       Virginia    8001024     369
## 25      Wisconsin    5686986     151
```

```
dfprime <- select(df, 'state', "region", "murders", "population")
dfprime
```

```
##             state        region murders population
## 1         Alabama         South     199    4779736
## 2         Arizona          West     352    6392017
## 3      California          West    1811   37253956
## 4        Colorado          West     117    5029196
## 5     Connecticut     Northeast     131    3574097
## 6         Florida         South     987   19687653
## 7         Georgia         South     527    9920000
## 8        Illinois North Central     453   12830632
## 9         Indiana North Central     198    6483802
## 10       Kentucky         South     180    4339367
## 11      Louisiana         South     437    4533372
## 12       Maryland         South     424    5773552
## 13  Massachusetts     Northeast     209    6547629
## 14       Michigan North Central     558    9883640
## 15       Missouri North Central     419    5988927
## 16     New Jersey     Northeast     363    8791894
## 17       New York     Northeast     860   19378102
## 18 North Carolina         South     445    9535483
## 19           Ohio North Central     460   11536504
## 20       Oklahoma         South     188    3751351
## 21   Pennsylvania     Northeast     646   12702379
## 22      Tennessee         South     356    6346105
## 23          Texas         South    1246   25145561
## 24       Virginia         South     369    8001024
```

```
## 25      Wisconsin North Central     151    5686986
```

for get names of columns, you can use below function. This work like `.columns` in pandas package in Python.

```
names(df)
```

```
## [1] "X"                 "state"             "abb"
## [4] "region"            "population"        "PopulationDensity"
## [7] "murders"           "gunmurders"        "gunownership"
```

Also you can select range of columns:

```
dfprime <- select(df, state:population)
dfprime
```

```
##             state abb        region population
## 1         Alabama  AL         South    4779736
## 2         Arizona  AZ          West    6392017
## 3      California  CA          West   37253956
## 4        Colorado  CO          West    5029196
## 5     Connecticut  CT     Northeast    3574097
## 6         Florida  FL         South   19687653
## 7         Georgia  GA         South    9920000
## 8        Illinois  IL North Central   12830632
## 9         Indiana  IN North Central    6483802
## 10       Kentucky  KY         South    4339367
## 11      Louisiana  LA         South    4533372
## 12       Maryland  MD         South    5773552
## 13  Massachusetts  MA     Northeast    6547629
## 14       Michigan  MI North Central    9883640
## 15       Missouri  MO North Central    5988927
## 16     New Jersey  NJ     Northeast    8791894
## 17       New York  NY     Northeast   19378102
## 18 North Carolina  NC         South    9535483
## 19           Ohio  OH North Central   11536504
## 20       Oklahoma  OK         South    3751351
## 21   Pennsylvania  PA     Northeast   12702379
## 22      Tennessee  TN         South    6346105
## 23          Texas  TX         South   25145561
## 24       Virginia  VA         South    8001024
## 25      Wisconsin  WI North Central    5686986
```

You can drop columns with use minus sign (-) in `select` function.

```
dfprime <- select(df, -abb)
dfprime
```

```
##     X          state        region population PopulationDensity murders
## 1   0        Alabama         South    4779736             94.65     199
## 2   1        Arizona          West    6392017             57.05     352
## 3   2     California          West   37253956            244.20    1811
## 4   3       Colorado          West    5029196             49.33     117
## 5   4    Connecticut     Northeast    3574097            741.40     131
## 6   5        Florida         South   19687653            360.20     987
## 7   6        Georgia         South    9920000            172.50     527
## 8   7       Illinois North Central   12830632            231.90     453
## 9   8        Indiana North Central    6483802            182.50     198
## 10  9       Kentucky         South    4339367            110.00     180
```

```
## 11 10      Louisiana         South    4533372            105.00     437
## 12 11       Maryland         South    5773552            606.20     424
## 13 12  Massachusetts     Northeast    6547629            852.10     209
## 14 13       Michigan North Central    9883640            174.80     558
## 15 14       Missouri North Central    5988927             87.26     419
## 16 15     New Jersey     Northeast    8791894           1189.00     363
## 17 16       New York     Northeast   19378102            415.30     860
## 18 17 North Carolina         South    9535483            200.60     445
## 19 18           Ohio North Central   11536504            282.50     460
## 20 19       Oklahoma         South    3751351             55.22     188
## 21 20   Pennsylvania     Northeast   12702379            285.30     646
## 22 21      Tennessee         South    6346105            156.60     356
## 23 22          Texas         South   25145561             98.07    1246
## 24 23       Virginia         South    8001024            207.30     369
## 25 24      Wisconsin North Central    5686986            105.20     151
##    gunmurders gunownership
## 1         135        0.517
## 2         232        0.311
## 3        1257        0.213
## 4          65        0.347
## 5          97        0.167
## 6         669        0.245
## 7         376        0.403
## 8         364        0.202
## 9         142        0.391
## 10        116        0.477
## 11        351        0.441
## 12        293        0.213
## 13        118        0.126
## 14        413        0.384
## 15        321        0.417
## 16        246        0.123
## 17        517        0.180
## 18        286        0.413
## 19        310        0.324
## 20        111        0.429
## 21        457        0.347
## 22        219        0.439
## 23        805        0.359
## 24        250        0.351
## 25         97        0.444
```

```
dfprime <- select(df, - c(abb, murders, gunmurders))
dfprime
```

```
##     X          state        region population PopulationDensity gunownership
## 1   0        Alabama         South    4779736             94.65        0.517
## 2   1        Arizona          West    6392017             57.05        0.311
## 3   2     California          West   37253956            244.20        0.213
## 4   3       Colorado          West    5029196             49.33        0.347
## 5   4    Connecticut     Northeast    3574097            741.40        0.167
## 6   5        Florida         South   19687653            360.20        0.245
## 7   6        Georgia         South    9920000            172.50        0.403
## 8   7       Illinois North Central   12830632            231.90        0.202
## 9   8        Indiana North Central    6483802            182.50        0.391
```

```
## 10  9       Kentucky         South    4339367           110.00        0.477
## 11 10      Louisiana         South    4533372           105.00        0.441
## 12 11       Maryland         South    5773552           606.20        0.213
## 13 12  Massachusetts     Northeast    6547629           852.10        0.126
## 14 13       Michigan North Central    9883640           174.80        0.384
## 15 14       Missouri North Central    5988927            87.26        0.417
## 16 15     New Jersey     Northeast    8791894          1189.00        0.123
## 17 16       New York     Northeast   19378102           415.30        0.180
## 18 17 North Carolina         South    9535483           200.60        0.413
## 19 18           Ohio North Central   11536504           282.50        0.324
## 20 19       Oklahoma         South    3751351            55.22        0.429
## 21 20   Pennsylvania     Northeast   12702379           285.30        0.347
## 22 21      Tennessee         South    6346105           156.60        0.439
## 23 22          Texas         South   25145561            98.07        0.359
## 24 23       Virginia         South    8001024           207.30        0.351
## 25 24      Wisconsin North Central    5686986           105.20        0.444
```

```
dfprime <- select(df, -(abb:murders))
dfprime
```

```
##     X          state gunmurders gunownership
## 1   0        Alabama        135        0.517
## 2   1        Arizona        232        0.311
## 3   2     California       1257        0.213
## 4   3       Colorado         65        0.347
## 5   4    Connecticut         97        0.167
## 6   5        Florida        669        0.245
## 7   6        Georgia        376        0.403
## 8   7       Illinois        364        0.202
## 9   8        Indiana        142        0.391
## 10  9       Kentucky        116        0.477
## 11 10      Louisiana        351        0.441
## 12 11       Maryland        293        0.213
## 13 12  Massachusetts        118        0.126
## 14 13       Michigan        413        0.384
## 15 14       Missouri        321        0.417
## 16 15     New Jersey        246        0.123
## 17 16       New York        517        0.180
## 18 17 North Carolina        286        0.413
## 19 18           Ohio        310        0.324
## 20 19       Oklahoma        111        0.429
## 21 20   Pennsylvania        457        0.347
## 22 21      Tennessee        219        0.439
## 23 22          Texas        805        0.359
## 24 23       Virginia        250        0.351
## 25 24      Wisconsin         97        0.444
```

7.2 dplyr `filter()` function

```
library(dplyr)

df <- read.csv("murders.csv")
names(df)
```

```
## [1] "X"                "state"            "abb"
```

```
## [4] "region"            "population"        "PopulationDensity"
## [7] "murders"           "gunmurders"        "gunownership"
```

```
dfprime <- filter(df, murders > 100)
dfprime
```

```
##     X          state abb        region population PopulationDensity murders
## 1   0        Alabama  AL         South    4779736              94.65     199
## 2   1        Arizona  AZ          West    6392017              57.05     352
## 3   2     California  CA          West   37253956             244.20    1811
## 4   3       Colorado  CO          West    5029196              49.33     117
## 5   4    Connecticut  CT     Northeast    3574097             741.40     131
## 6   5        Florida  FL         South   19687653             360.20     987
## 7   6        Georgia  GA         South    9920000             172.50     527
## 8   7       Illinois  IL North Central   12830632             231.90     453
## 9   8        Indiana  IN North Central    6483802             182.50     198
## 10  9       Kentucky  KY         South    4339367             110.00     180
## 11 10      Louisiana  LA         South    4533372             105.00     437
## 12 11       Maryland  MD         South    5773552             606.20     424
## 13 12  Massachusetts  MA     Northeast    6547629             852.10     209
## 14 13       Michigan  MI North Central    9883640             174.80     558
## 15 14       Missouri  MO North Central    5988927              87.26     419
## 16 15     New Jersey  NJ     Northeast    8791894            1189.00     363
## 17 16       New York  NY     Northeast   19378102             415.30     860
## 18 17 North Carolina  NC         South    9535483             200.60     445
## 19 18           Ohio  OH North Central   11536504             282.50     460
## 20 19       Oklahoma  OK         South    3751351              55.22     188
## 21 20   Pennsylvania  PA     Northeast   12702379             285.30     646
## 22 21      Tennessee  TN         South    6346105             156.60     356
## 23 22          Texas  TX         South   25145561              98.07    1246
## 24 23       Virginia  VA         South    8001024             207.30     369
## 25 24      Wisconsin  WI North Central    5686986             105.20     151
##    gunmurders gunownership
## 1         135        0.517
## 2         232        0.311
## 3        1257        0.213
## 4          65        0.347
## 5          97        0.167
## 6         669        0.245
## 7         376        0.403
## 8         364        0.202
## 9         142        0.391
## 10        116        0.477
## 11        351        0.441
## 12        293        0.213
## 13        118        0.126
## 14        413        0.384
## 15        321        0.417
## 16        246        0.123
## 17        517        0.180
## 18        286        0.413
## 19        310        0.324
## 20        111        0.429
## 21        457        0.347
## 22        219        0.439
```

```
## 23        805        0.359
## 24        250        0.351
## 25         97        0.444
```

```
dfprime <- filter(df, murders > 100 & population > 10000000 )
dfprime
```

```
##    X        state abb        region population PopulationDensity murders
## 1  2   California  CA          West   37253956             244.20    1811
## 2  5      Florida  FL         South   19687653             360.20     987
## 3  7     Illinois  IL North Central   12830632             231.90     453
## 4 16     New York  NY     Northeast   19378102             415.30     860
## 5 18         Ohio  OH North Central   11536504             282.50     460
## 6 20 Pennsylvania  PA     Northeast   12702379             285.30     646
## 7 22        Texas  TX         South   25145561              98.07    1246
##   gunmurders gunownership
## 1       1257        0.213
## 2        669        0.245
## 3        364        0.202
## 4        517        0.180
## 5        310        0.324
## 6        457        0.347
## 7        805        0.359
```

dplyr `arrange()` function

```
library(dplyr)

df <- read.csv("murders.csv")
names(df)
```

```
## [1] "X"                "state"            "abb"
## [4] "region"           "population"       "PopulationDensity"
## [7] "murders"          "gunmurders"       "gunownership"
```

```
dfprime <- arrange(df, murders)
dfprime
```

```
##     X          state abb        region population PopulationDensity murders
## 1   3       Colorado  CO          West    5029196              49.33     117
## 2   4    Connecticut  CT     Northeast    3574097             741.40     131
## 3  24      Wisconsin  WI North Central    5686986             105.20     151
## 4   9       Kentucky  KY         South    4339367             110.00     180
## 5  19       Oklahoma  OK         South    3751351              55.22     188
## 6   8        Indiana  IN North Central    6483802             182.50     198
## 7   0        Alabama  AL         South    4779736              94.65     199
## 8  12  Massachusetts  MA     Northeast    6547629             852.10     209
## 9   1        Arizona  AZ          West    6392017              57.05     352
## 10 21      Tennessee  TN         South    6346105             156.60     356
## 11 15     New Jersey  NJ     Northeast    8791894            1189.00     363
## 12 23       Virginia  VA         South    8001024             207.30     369
## 13 14       Missouri  MO North Central    5988927              87.26     419
## 14 11       Maryland  MD         South    5773552             606.20     424
## 15 10      Louisiana  LA         South    4533372             105.00     437
## 16 17 North Carolina  NC         South    9535483             200.60     445
## 17  7       Illinois  IL North Central   12830632             231.90     453
## 18 18           Ohio  OH North Central   11536504             282.50     460
```

```
## 19  6        Georgia  GA         South    9920000            172.50     527
## 20 13       Michigan  MI North Central    9883640            174.80     558
## 21 20   Pennsylvania  PA     Northeast   12702379            285.30     646
## 22 16       New York  NY     Northeast   19378102            415.30     860
## 23  5        Florida  FL         South   19687653            360.20     987
## 24 22          Texas  TX         South   25145561             98.07    1246
## 25  2     California  CA          West   37253956            244.20    1811
##    gunmurders gunownership
## 1          65        0.347
## 2          97        0.167
## 3          97        0.444
## 4         116        0.477
## 5         111        0.429
## 6         142        0.391
## 7         135        0.517
## 8         118        0.126
## 9         232        0.311
## 10        219        0.439
## 11        246        0.123
## 12        250        0.351
## 13        321        0.417
## 14        293        0.213
## 15        351        0.441
## 16        286        0.413
## 17        364        0.202
## 18        310        0.324
## 19        376        0.403
## 20        413        0.384
## 21        457        0.347
## 22        517        0.180
## 23        669        0.245
## 24        805        0.359
## 25       1257        0.213
```

```
# for descending order

dfprime <- arrange(df, desc(murders))
dfprime
```

```
##     X          state abb        region population PopulationDensity murders
## 1   2     California  CA          West   37253956            244.20    1811
## 2  22          Texas  TX         South   25145561             98.07    1246
## 3   5        Florida  FL         South   19687653            360.20     987
## 4  16       New York  NY     Northeast   19378102            415.30     860
## 5  20   Pennsylvania  PA     Northeast   12702379            285.30     646
## 6  13       Michigan  MI North Central    9883640            174.80     558
## 7   6        Georgia  GA         South    9920000            172.50     527
## 8  18           Ohio  OH North Central   11536504            282.50     460
## 9   7       Illinois  IL North Central   12830632            231.90     453
## 10 17 North Carolina  NC         South    9535483            200.60     445
## 11 10      Louisiana  LA         South    4533372            105.00     437
## 12 11       Maryland  MD         South    5773552            606.20     424
## 13 14       Missouri  MO North Central    5988927             87.26     419
## 14 23       Virginia  VA         South    8001024            207.30     369
## 15 15     New Jersey  NJ     Northeast    8791894           1189.00     363
```

```
## 16 21      Tennessee  TN         South    6346105           156.60     356
## 17  1        Arizona  AZ          West    6392017            57.05     352
## 18 12  Massachusetts  MA     Northeast    6547629           852.10     209
## 19  0        Alabama  AL         South    4779736            94.65     199
## 20  8        Indiana  IN North Central    6483802           182.50     198
## 21 19       Oklahoma  OK         South    3751351            55.22     188
## 22  9       Kentucky  KY         South    4339367           110.00     180
## 23 24      Wisconsin  WI North Central    5686986           105.20     151
## 24  4    Connecticut  CT     Northeast    3574097           741.40     131
## 25  3       Colorado  CO          West    5029196            49.33     117
##    gunmurders gunownership
## 1        1257        0.213
## 2         805        0.359
## 3         669        0.245
## 4         517        0.180
## 5         457        0.347
## 6         413        0.384
## 7         376        0.403
## 8         310        0.324
## 9         364        0.202
## 10        286        0.413
## 11        351        0.441
## 12        293        0.213
## 13        321        0.417
## 14        250        0.351
## 15        246        0.123
## 16        219        0.439
## 17        232        0.311
## 18        118        0.126
## 19        135        0.517
## 20        142        0.391
## 21        111        0.429
## 22        116        0.477
## 23         97        0.444
## 24         97        0.167
## 25         65        0.347
```

7.3 dplyr `rename()` function

```
library(dplyr)

df <- read.csv('murders.csv')
names(df)
```

```
## [1] "X"                "state"            "abb"
## [4] "region"           "population"       "PopulationDensity"
## [7] "murders"          "gunmurders"       "gunownership"
```

```
df2 <- rename(df, abbreviation = abb)
names(df2)
```

```
## [1] "X"                "state"            "abbreviation"
## [4] "region"           "population"       "PopulationDensity"
## [7] "murders"          "gunmurders"       "gunownership"
```

```
df2 <- rename(df, abbreviation = abb, homicide = murders)
names(df2)
```

```
## [1] "X"                 "state"             "abbreviation"
## [4] "region"            "population"        "PopulationDensity"
## [7] "homicide"          "gunmurders"        "gunownership"
```

7.4 dplyr mutate() function

```
library(dplyr)

df <- read.csv('murders.csv')
names(df)
```

```
## [1] "X"                 "state"             "abb"
## [4] "region"            "population"        "PopulationDensity"
## [7] "murders"           "gunmurders"        "gunownership"
```

```
dfprime <- mutate(df, ratio = murders / population)
dfprime
```

```
##     X          state abb        region population PopulationDensity murders
## 1   0        Alabama  AL         South    4779736              94.65     199
## 2   1        Arizona  AZ          West    6392017              57.05     352
## 3   2     California  CA          West   37253956             244.20    1811
## 4   3       Colorado  CO          West    5029196              49.33     117
## 5   4    Connecticut  CT     Northeast    3574097             741.40     131
## 6   5        Florida  FL         South   19687653             360.20     987
## 7   6        Georgia  GA         South    9920000             172.50     527
## 8   7       Illinois  IL North Central   12830632             231.90     453
## 9   8        Indiana  IN North Central    6483802             182.50     198
## 10  9       Kentucky  KY         South    4339367             110.00     180
## 11 10      Louisiana  LA         South    4533372             105.00     437
## 12 11       Maryland  MD         South    5773552             606.20     424
## 13 12  Massachusetts  MA     Northeast    6547629             852.10     209
## 14 13       Michigan  MI North Central    9883640             174.80     558
## 15 14       Missouri  MO North Central    5988927              87.26     419
## 16 15     New Jersey  NJ     Northeast    8791894            1189.00     363
## 17 16       New York  NY     Northeast   19378102             415.30     860
## 18 17 North Carolina  NC         South    9535483             200.60     445
## 19 18           Ohio  OH North Central   11536504             282.50     460
## 20 19       Oklahoma  OK         South    3751351              55.22     188
## 21 20   Pennsylvania  PA     Northeast   12702379             285.30     646
## 22 21      Tennessee  TN         South    6346105             156.60     356
## 23 22          Texas  TX         South   25145561              98.07    1246
## 24 23       Virginia  VA         South    8001024             207.30     369
## 25 24      Wisconsin  WI North Central    5686986             105.20     151
##    gunmurders gunownership        ratio
## 1         135        0.517 4.163410e-05
## 2         232        0.311 5.506869e-05
## 3        1257        0.213 4.861229e-05
## 4          65        0.347 2.326416e-05
## 5          97        0.167 3.665261e-05
## 6         669        0.245 5.013294e-05
## 7         376        0.403 5.312500e-05
```

```
## 8         364        0.202 3.530613e-05
## 9         142        0.391 3.053764e-05
## 10        116        0.477 4.148070e-05
## 11        351        0.441 9.639624e-05
## 12        293        0.213 7.343833e-05
## 13        118        0.126 3.191995e-05
## 14        413        0.384 5.645693e-05
## 15        321        0.417 6.996245e-05
## 16        246        0.123 4.128803e-05
## 17        517        0.180 4.437999e-05
## 18        286        0.413 4.666780e-05
## 19        310        0.324 3.987343e-05
## 20        111        0.429 5.011528e-05
## 21        457        0.347 5.085662e-05
## 22        219        0.439 5.609740e-05
## 23        805        0.359 4.955149e-05
## 24        250        0.351 4.611910e-05
## 25         97        0.444 2.655185e-05
```

```
names(dfprime)
```

```
##  [1] "X"                "state"            "abb"
##  [4] "region"           "population"       "PopulationDensity"
##  [7] "murders"          "gunmurders"       "gunownership"
## [10] "ratio"
```

```
select(dfprime, state, population, murders, ratio)
```

```
##             state population murders        ratio
## 1         Alabama    4779736     199 4.163410e-05
## 2         Arizona    6392017     352 5.506869e-05
## 3      California   37253956    1811 4.861229e-05
## 4        Colorado    5029196     117 2.326416e-05
## 5     Connecticut    3574097     131 3.665261e-05
## 6         Florida   19687653     987 5.013294e-05
## 7         Georgia    9920000     527 5.312500e-05
## 8        Illinois   12830632     453 3.530613e-05
## 9         Indiana    6483802     198 3.053764e-05
## 10       Kentucky    4339367     180 4.148070e-05
## 11      Louisiana    4533372     437 9.639624e-05
## 12       Maryland    5773552     424 7.343833e-05
## 13  Massachusetts    6547629     209 3.191995e-05
## 14       Michigan    9883640     558 5.645693e-05
## 15       Missouri    5988927     419 6.996245e-05
## 16     New Jersey    8791894     363 4.128803e-05
## 17       New York   19378102     860 4.437999e-05
## 18 North Carolina    9535483     445 4.666780e-05
## 19           Ohio   11536504     460 3.987343e-05
## 20       Oklahoma    3751351     188 5.011528e-05
## 21   Pennsylvania   12702379     646 5.085662e-05
## 22      Tennessee    6346105     356 5.609740e-05
## 23          Texas   25145561    1246 4.955149e-05
## 24       Virginia    8001024     369 4.611910e-05
## 25      Wisconsin    5686986     151 2.655185e-05
```

```r
dfprime <- transmute(df, ratio = murders / population)
dfprime
```

```
##           ratio
## 1  4.163410e-05
## 2  5.506869e-05
## 3  4.861229e-05
## 4  2.326416e-05
## 5  3.665261e-05
## 6  5.013294e-05
## 7  5.312500e-05
## 8  3.530613e-05
## 9  3.053764e-05
## 10 4.148070e-05
## 11 9.639624e-05
## 12 7.343833e-05
## 13 3.191995e-05
## 14 5.645693e-05
## 15 6.996245e-05
## 16 4.128803e-05
## 17 4.437999e-05
## 18 4.666780e-05
## 19 3.987343e-05
## 20 5.011528e-05
## 21 5.085662e-05
## 22 5.609740e-05
## 23 4.955149e-05
## 24 4.611910e-05
## 25 2.655185e-05
```

```r
dfprime <- transmute(df, state = state, ratio = murders / population)
dfprime
```

```
##             state        ratio
## 1         Alabama 4.163410e-05
## 2         Arizona 5.506869e-05
## 3      California 4.861229e-05
## 4        Colorado 2.326416e-05
## 5     Connecticut 3.665261e-05
## 6         Florida 5.013294e-05
## 7         Georgia 5.312500e-05
## 8        Illinois 3.530613e-05
## 9         Indiana 3.053764e-05
## 10       Kentucky 4.148070e-05
## 11      Louisiana 9.639624e-05
## 12       Maryland 7.343833e-05
## 13  Massachusetts 3.191995e-05
## 14       Michigan 5.645693e-05
## 15       Missouri 6.996245e-05
## 16     New Jersey 4.128803e-05
## 17       New York 4.437999e-05
## 18 North Carolina 4.666780e-05
## 19           Ohio 3.987343e-05
## 20       Oklahoma 5.011528e-05
```

```
## 21   Pennsylvania 5.085662e-05
## 22      Tennessee 5.609740e-05
## 23          Texas 4.955149e-05
## 24       Virginia 4.611910e-05
## 25      Wisconsin 2.655185e-05
```

7.5 dplyr group_by() function

```
library(dplyr)

df <- read.csv('murders.csv')
names(df)
```

```
## [1] "X"                "state"            "abb"
## [4] "region"           "population"       "PopulationDensity"
## [7] "murders"          "gunmurders"       "gunownership"
```

```
dfprime <- group_by(df, region)
dfprime
```

```
## # A tibble: 25 x 9
## # Groups:   region [4]
##        X state       abb   region population PopulationDensi~ murders gunmurders
##    <int> <chr>       <chr> <chr>       <int>            <dbl>   <int>      <int>
##  1     0 Alabama     AL    South     4779736             94.6     199        135
##  2     1 Arizona     AZ    West      6392017             57.0     352        232
##  3     2 California  CA    West     37253956            244.     1811       1257
##  4     3 Colorado    CO    West      5029196             49.3     117         65
##  5     4 Connecticut CT    North~    3574097            741.      131         97
##  6     5 Florida     FL    South    19687653            360.      987        669
##  7     6 Georgia     GA    South     9920000            172.      527        376
##  8     7 Illinois    IL    North~   12830632            232.      453        364
##  9     8 Indiana     IN    North~    6483802            182.      198        142
## 10     9 Kentucky    KY    South     4339367            110       180        116
## # ... with 15 more rows, and 1 more variable: gunownership <dbl>
```

```
summarise(dfprime, sum(murders))
```

```
## # A tibble: 4 x 2
##   region        `sum(murders)`
##   <chr>                  <int>
## 1 North Central           2239
## 2 Northeast               2209
## 3 South                   5358
## 4 West                    2280
```

```
summarise(dfprime, sum(murders), mean(gunownership), median(population))
```

```
## # A tibble: 4 x 4
##   region        `sum(murders)` `mean(gunownership)` `median(population)`
##   <chr>                  <int>                <dbl>                  <dbl>
## 1 North Central           2239                0.360                8183721
## 2 Northeast               2209                0.189                8791894
## 3 South                   5358                0.390                6346105
## 4 West                    2280                0.290                6392017
```

7.6 dplyr Pipe Operator %>%

```
library(dplyr)

df <- read.csv('murders.csv')
names(df)
```

```
## [1] "X"                 "state"             "abb"
## [4] "region"            "population"        "PopulationDensity"
## [7] "murders"           "gunmurders"        "gunownership"
```

```
dfprime <- arrange(df, murders)
dfprime
```

```
##     X          state abb        region population PopulationDensity murders
## 1   3       Colorado  CO          West    5029196              49.33     117
## 2   4    Connecticut  CT     Northeast    3574097             741.40     131
## 3  24      Wisconsin  WI North Central    5686986             105.20     151
## 4   9       Kentucky  KY         South    4339367             110.00     180
## 5  19       Oklahoma  OK         South    3751351              55.22     188
## 6   8        Indiana  IN North Central    6483802             182.50     198
## 7   0        Alabama  AL         South    4779736              94.65     199
## 8  12  Massachusetts  MA     Northeast    6547629             852.10     209
## 9   1        Arizona  AZ          West    6392017              57.05     352
## 10 21      Tennessee  TN         South    6346105             156.60     356
## 11 15     New Jersey  NJ     Northeast    8791894            1189.00     363
## 12 23       Virginia  VA         South    8001024             207.30     369
## 13 14       Missouri  MO North Central    5988927              87.26     419
## 14 11       Maryland  MD         South    5773552             606.20     424
## 15 10      Louisiana  LA         South    4533372             105.00     437
## 16 17 North Carolina  NC         South    9535483             200.60     445
## 17  7       Illinois  IL North Central   12830632             231.90     453
## 18 18           Ohio  OH North Central   11536504             282.50     460
## 19  6        Georgia  GA         South    9920000             172.50     527
## 20 13       Michigan  MI North Central    9883640             174.80     558
## 21 20   Pennsylvania  PA     Northeast   12702379             285.30     646
## 22 16       New York  NY     Northeast   19378102             415.30     860
## 23  5        Florida  FL         South   19687653             360.20     987
## 24 22          Texas  TX         South   25145561              98.07    1246
## 25  2     California  CA          West   37253956             244.20    1811
##    gunmurders gunownership
## 1          65        0.347
## 2          97        0.167
## 3          97        0.444
## 4         116        0.477
## 5         111        0.429
## 6         142        0.391
## 7         135        0.517
## 8         118        0.126
## 9         232        0.311
## 10        219        0.439
## 11        246        0.123
## 12        250        0.351
## 13        321        0.417
## 14        293        0.213
```

```
## 15        351        0.441
## 16        286        0.413
## 17        364        0.202
## 18        310        0.324
## 19        376        0.403
## 20        413        0.384
## 21        457        0.347
## 22        517        0.180
## 23        669        0.245
## 24        805        0.359
## 25       1257        0.213
```

```
dfprime2 <- select(dfprime, state, murders)
dfprime2
```

```
##             state murders
## 1        Colorado     117
## 2     Connecticut     131
## 3       Wisconsin     151
## 4        Kentucky     180
## 5        Oklahoma     188
## 6         Indiana     198
## 7         Alabama     199
## 8   Massachusetts     209
## 9         Arizona     352
## 10      Tennessee     356
## 11     New Jersey     363
## 12       Virginia     369
## 13       Missouri     419
## 14       Maryland     424
## 15      Louisiana     437
## 16 North Carolina     445
## 17       Illinois     453
## 18           Ohio     460
## 19        Georgia     527
## 20       Michigan     558
## 21   Pennsylvania     646
## 22       New York     860
## 23        Florida     987
## 24          Texas    1246
## 25     California    1811
```

```
head(dfprime2, 5)
```

```
##         state murders
## 1    Colorado     117
## 2 Connecticut     131
## 3   Wisconsin     151
## 4    Kentucky     180
## 5    Oklahoma     188
```

```
arrange(df, murders) %>% select(state, murders) %>% head(5)
```

```
##         state murders
## 1    Colorado     117
## 2 Connecticut     131
```

```
## 3   Wisconsin     151
## 4    Kentucky     180
## 5    Oklahoma     188
```

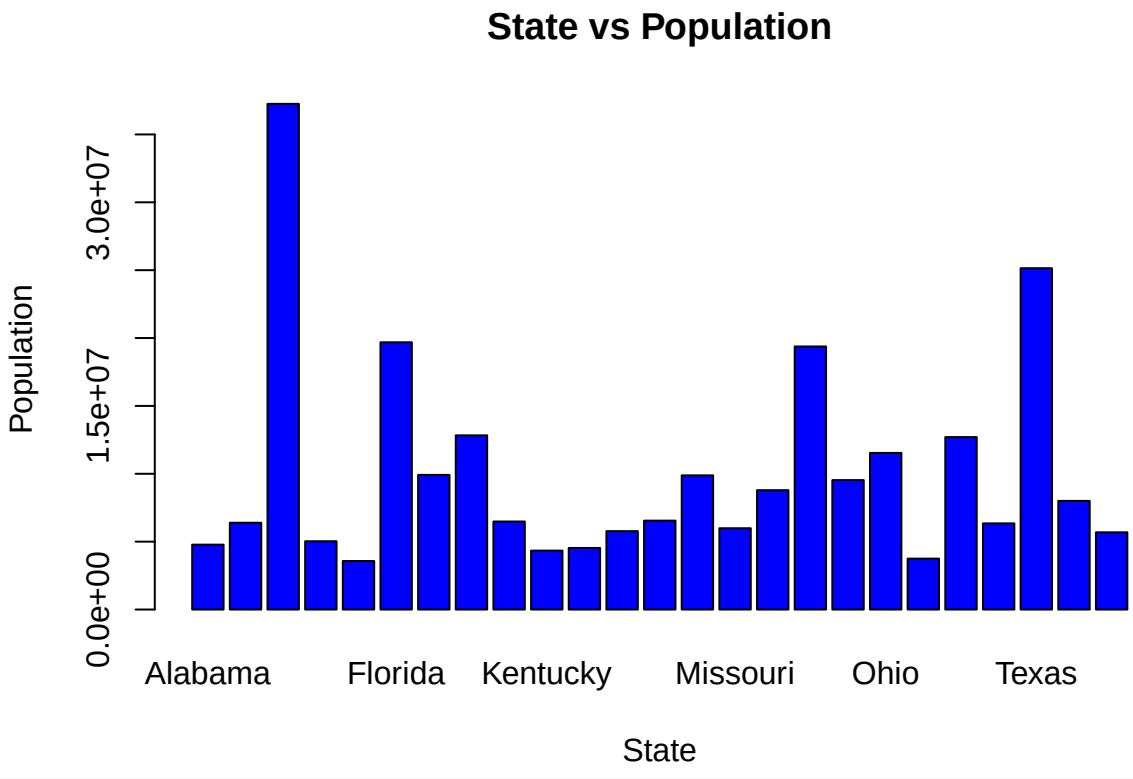
8 Data Visualization with `dplyr`

8.1 Bar Graphs

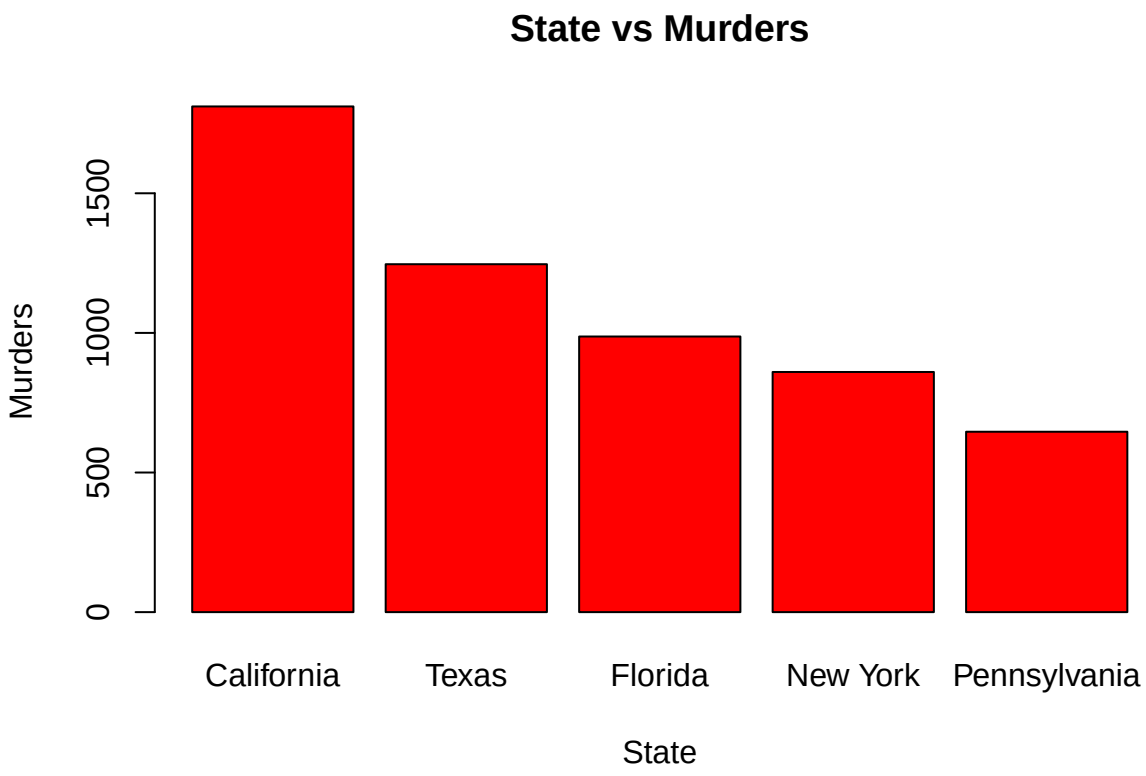
```
library(dplyr)

df <- read.csv('murders.csv')
names(df)
```

```
## [1] "X"                 "state"             "abb"
## [4] "region"            "population"        "PopulationDensity"
## [7] "murders"           "gunmurders"        "gunownership"
```

```
barplot(df$population,
        xlab = 'State',
        ylab = "Population",
        main = "State vs Population",
        names.arg = df$state,
        col = 'blue'
        )
```

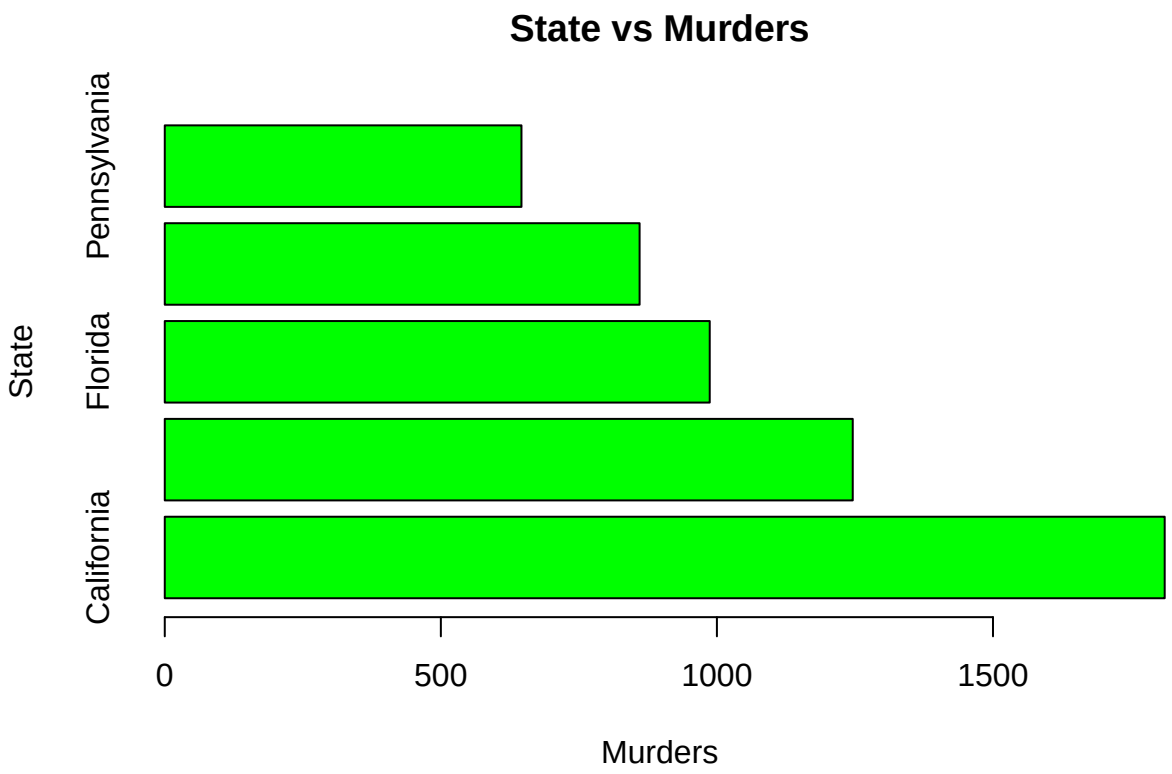



```
dfprime <- arrange(df, desc(murders)) %>% head(5)
#dfprime

barplot(dfprime$murders,
        xlab = 'State',
        ylab = "Murders",
        main = "State vs Murders",
        names.arg = dfprime$state,
        col = 'red'
        )
```



Horizontsl Bar Graphs

```
dfprime <- arrange(df, desc(murders)) %>% head(5)
#dfprime

barplot(dfprime$murders,
        xlab = 'Murders',
        ylab = "State",
        main = "State vs Murders",
        names.arg = dfprime$state,
        col = 'green',
        horiz = TRUE
        )
```



stacked Bar Plots

```
library(dplyr)

df <- read.csv('murders.csv')
names(df)
```

```
## [1] "X"                 "state"             "abb"
## [4] "region"            "population"        "PopulationDensity"
## [7] "murders"           "gunmurders"        "gunownership"
```

```
dfprime <- mutate(df, popu = population / 10000)
names(dfprime)
```

```
##  [1] "X"                 "state"             "abb"
##  [4] "region"            "population"        "PopulationDensity"
##  [7] "murders"           "gunmurders"        "gunownership"
## [10] "popu"
```

```
dfmatrix <- data.matrix(dfprime[c(2,7,10)])
dfmatrix
```

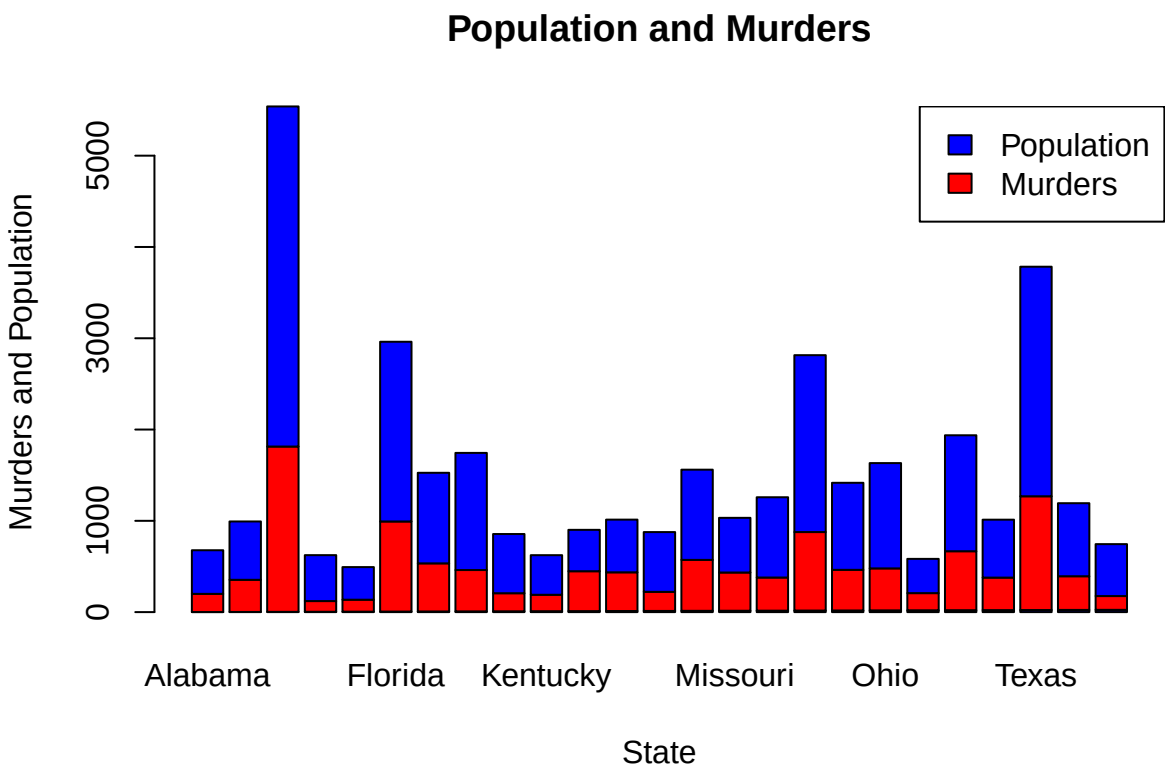
```
##       state murders      popu
##  [1,]     1     199  477.9736
##  [2,]     2     352  639.2017
##  [3,]     3    1811 3725.3956
##  [4,]     4     117  502.9196
##  [5,]     5     131  357.4097
##  [6,]     6     987 1968.7653
##  [7,]     7     527  992.0000
##  [8,]     8     453 1283.0632
##  [9,]     9     198  648.3802
```

```
## [10,]    10     180  433.9367
## [11,]    11     437  453.3372
## [12,]    12     424  577.3552
## [13,]    13     209  654.7629
## [14,]    14     558  988.3640
## [15,]    15     419  598.8927
## [16,]    16     363  879.1894
## [17,]    17     860 1937.8102
## [18,]    18     445  953.5483
## [19,]    19     460 1153.6504
## [20,]    20     188  375.1351
## [21,]    21     646 1270.2379
## [22,]    22     356  634.6105
## [23,]    23    1246 2514.5561
## [24,]    24     369  800.1024
## [25,]    25     151  568.6986
```

```
dfmatrix <- t(dfmatrix)
dfmatrix
```

```
##             [,1]     [,2]     [,3]     [,4]     [,5]     [,6] [,7]     [,8]
## state     1.0000   2.0000    3.000   4.0000   5.0000    6.000    7    8.000
## murders 199.0000 352.0000 1811.000 117.0000 131.0000  987.000  527  453.000
## popu    477.9736 639.2017 3725.396 502.9196 357.4097 1968.765  992 1283.063
##             [,9]    [,10]    [,11]    [,12]    [,13]   [,14]    [,15]    [,16]
## state     9.0000  10.0000  11.0000  12.0000  13.0000  14.000  15.0000  16.0000
## murders 198.0000 180.0000 437.0000 424.0000 209.0000 558.000 419.0000 363.0000
## popu    648.3802 433.9367 453.3372 577.3552 654.7629 988.364 598.8927 879.1894
##           [,17]    [,18]   [,19]    [,20]    [,21]    [,22]    [,23]    [,24]
## state     17.00  18.0000   19.00  20.0000   21.000  22.0000   23.000  24.0000
## murders  860.00 445.0000  460.00 188.0000  646.000 356.0000 1246.000 369.0000
## popu    1937.81 953.5483 1153.65 375.1351 1270.238 634.6105 2514.556 800.1024
##             [,25]
## state     25.0000
## murders 151.0000
## popu    568.6986
```

```
barplot(dfmatrix,
        xlab = 'State',
        ylab = 'Murders and Population',
        main = "Population and Murders",
        names.arg = df$state,
        col = c("blue", 'red')
        )
legend("topright", c("Population", "Murders"), fill = c('blue', 'red'))
```

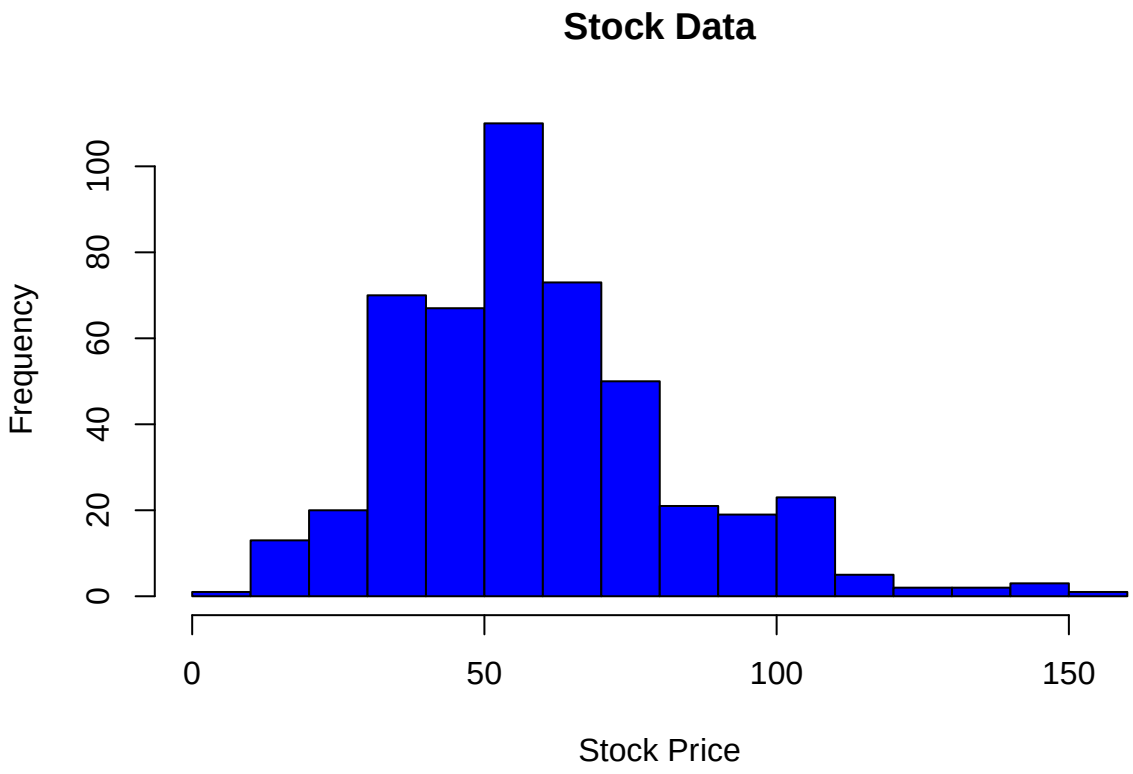


8.2 Histogram

```
library(dplyr)

df <- read.csv('GEStock.csv')
#df
names(df)
```

```
## [1] "X"     "Date"  "Price"
```

```
hist(df$Price,
     xlab = 'Stock Price',
     main = "Stock Data",
     col = 'blue',
     border = 'black',
     breaks = 20
     )
```



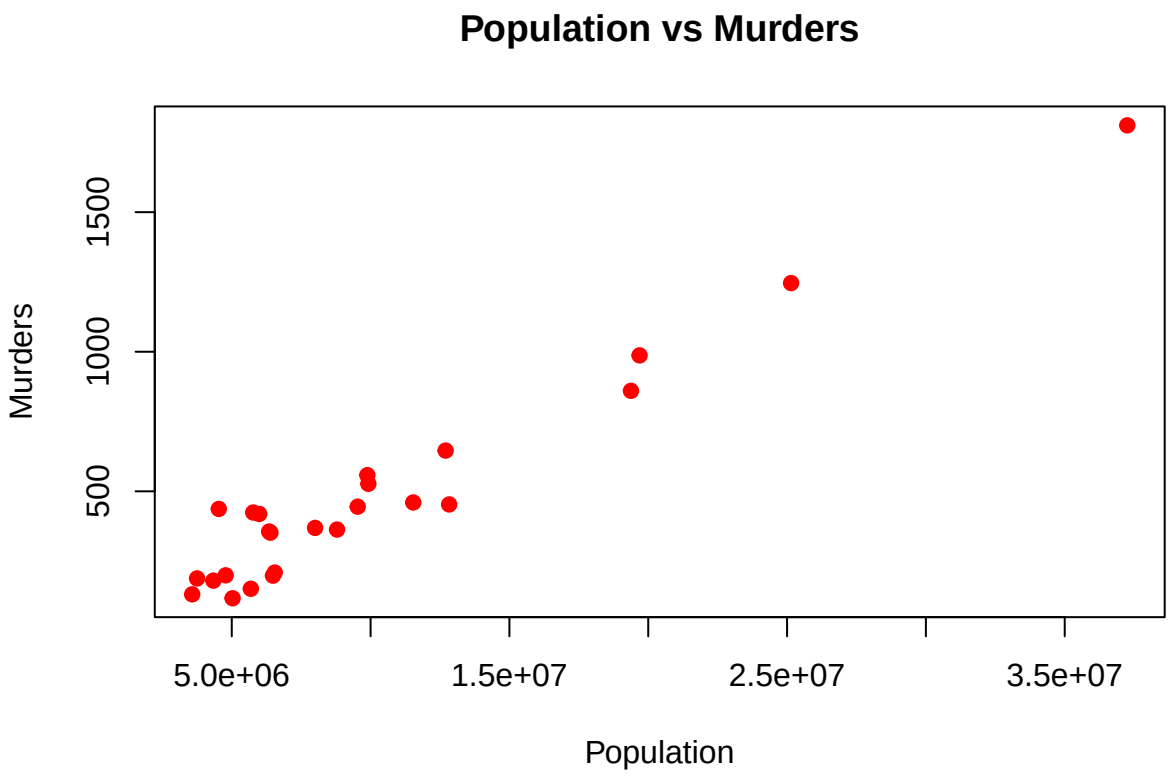
8.3 Scatter Plots

The default of `plot` functions is Scatter plot.

```
library(dplyr)

df <- read.csv('murders.csv')
names(df)
```

```
## [1] "X"                 "state"             "abb"
## [4] "region"            "population"        "PopulationDensity"
## [7] "murders"           "gunmurders"        "gunownership"
```

```
plot(df$population, df$murders,
     xlab = "Population",
     ylab = "Murders",
     main = "Population vs Murders",
     col = 'red',
     pch = 19
     )
```



pch values

Values of pch are stored internally as integers.

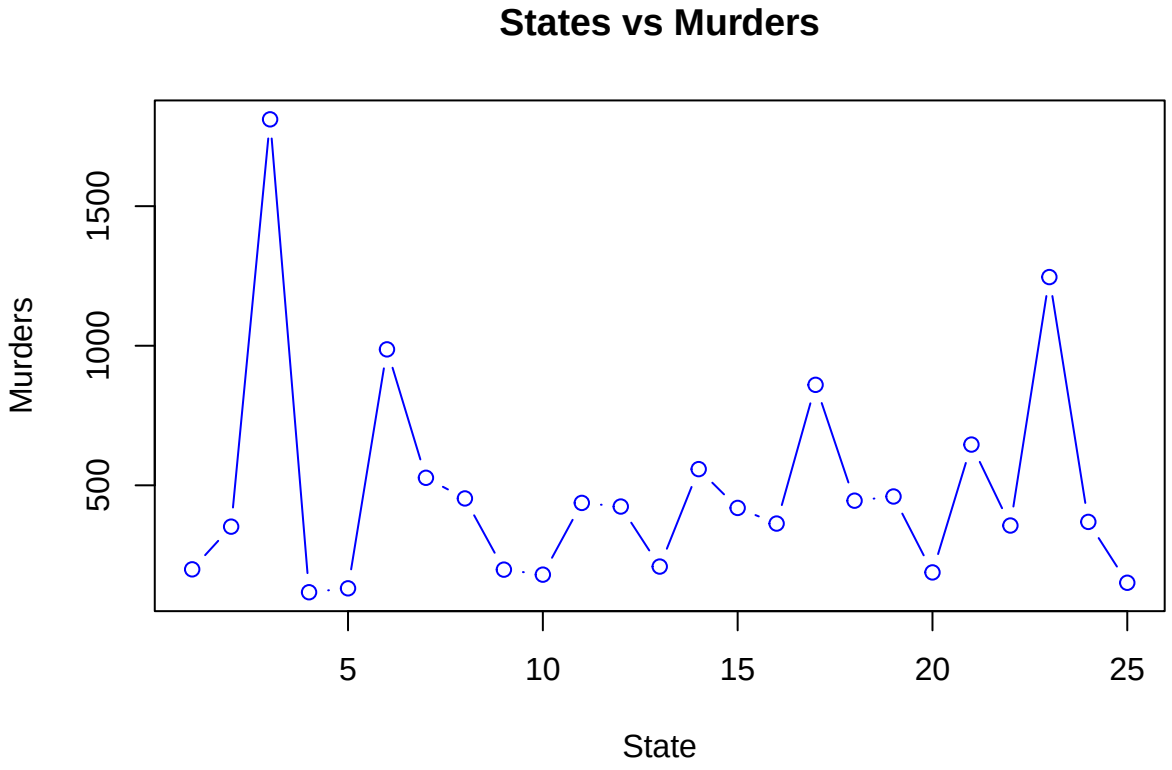
Figure 1: Some values of pch

8.4 Line Graphs

```
library(dplyr)

df <- read.csv('murders.csv')
names(df)
```

```
## [1] "X"                "state"            "abb"
## [4] "region"           "population"       "PopulationDensity"
## [7] "murders"          "gunmurders"       "gunownership"
```

```
plot(df$murders,
     type = 'b',
     xlab = 'State',
     ylab = "Murders",
     main = "States vs Murders",
     col = 'blue'
     )
```



type in plot function

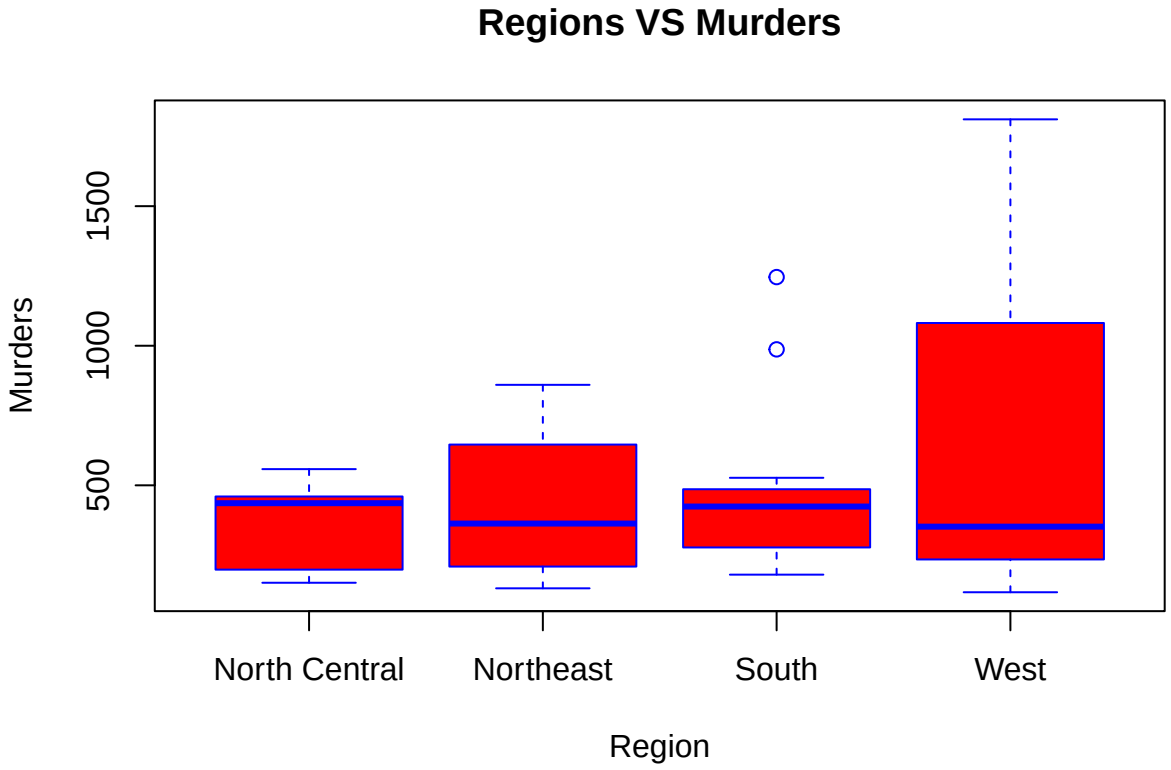
- **"p"** for points,
- **"l"** for lines,
- **"b"** for both points and lines,
- **"c"** for empty points joined by lines,
- **"o"** for overplotted points and lines,
- **"s"** and **"S"** for stair steps,
- **"h"** for histogram-like vertical lines,
- **"n"** does not produce any points or lines.

8.5 Box plots

```
library(dplyr)
df <- read.csv('murders.csv')
names(df)
```

```
## [1] "X"                 "state"             "abb"
## [4] "region"            "population"        "PopulationDensity"
## [7] "murders"           "gunmurders"        "gunownership"
```

```
boxplot(df$murders ~ df$region,
        xlab = "Region",
        ylab = "Murders",
        main = "Regions VS Murders",
        col = 'red',
        border = 'blue'
        )
```



8.6 Multiple Plots in Layout

```
par(mfrow = c(1,2))
library(dplyr)
df <- read.csv('murders.csv')
names(df)
```

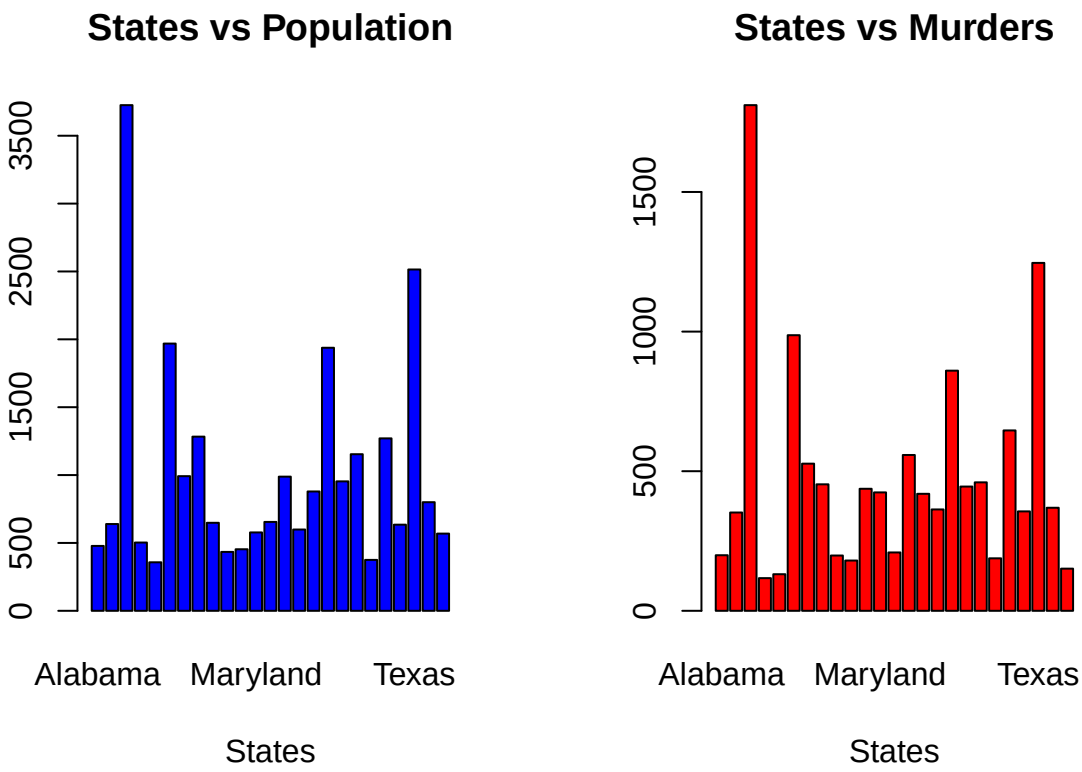
```
## [1] "X"                "state"             "abb"
## [4] "region"           "population"        "PopulationDensity"
## [7] "murders"          "gunmurders"        "gunownership"
```

```
dfprime <- mutate(df, popu = population / 10000)
names(dfprime)
```

```
##  [1] "X"                 "state"             "abb"
##  [4] "region"            "population"        "PopulationDensity"
##  [7] "murders"           "gunmurders"        "gunownership"
## [10] "popu"
```

```
barplot(dfprime$popu,
        xlab = 'States',
        main = "States vs Population",
        col = 'blue',
        names.arg = dfprime$state
        )

barplot(dfprime$murders,
        xlab = 'States',
        main = "States vs Murders",
        col = 'red',
```

```
        names.arg = dfprime$state
        )
```




```
par(mfrow = c(2,2))
library(dplyr)
df <- read.csv('murders.csv')
names(df)
```

```
## [1] "X"                "state"             "abb"
## [4] "region"           "population"        "PopulationDensity"
## [7] "murders"          "gunmurders"        "gunownership"
```

```
dfprime <- mutate(df, popu = population / 10000)
names(dfprime)
```

```
##  [1] "X"                 "state"             "abb"
##  [4] "region"            "population"        "PopulationDensity"
##  [7] "murders"           "gunmurders"        "gunownership"
## [10] "popu"
```

```
barplot(dfprime$popu,
        xlab = 'States',
        main = "States vs Population",
        col = 'blue',
        names.arg = dfprime$state
        )

barplot(dfprime$murders,
        xlab = 'States',
        main = "States vs Murders",
        col = 'red',
```

```
        names.arg = dfprime$state
        )

plot(dfprime$popu, df$murders,
     xlab = "Population",
     ylab = "Murders",
     main = 'Population VS Murders',
     col = 'red',
     pch = 19
     )

plot(df$murders,
     type = 'l',
     xlab = 'State',
     ylab = "Murders",
     main = "States vs Murders",
     col = 'blue'
     )
```