Introduction to R

Saeed Saffari Saeed.Saffari@dal.ca

Winter 2026

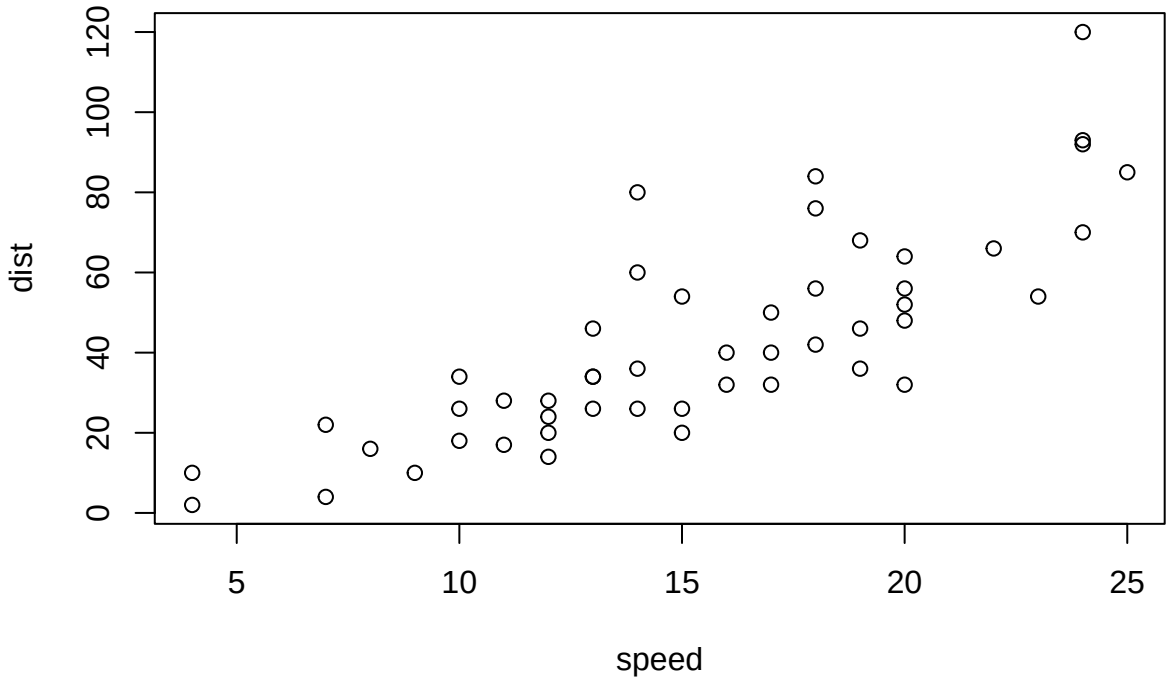
Contents

1 Introduction

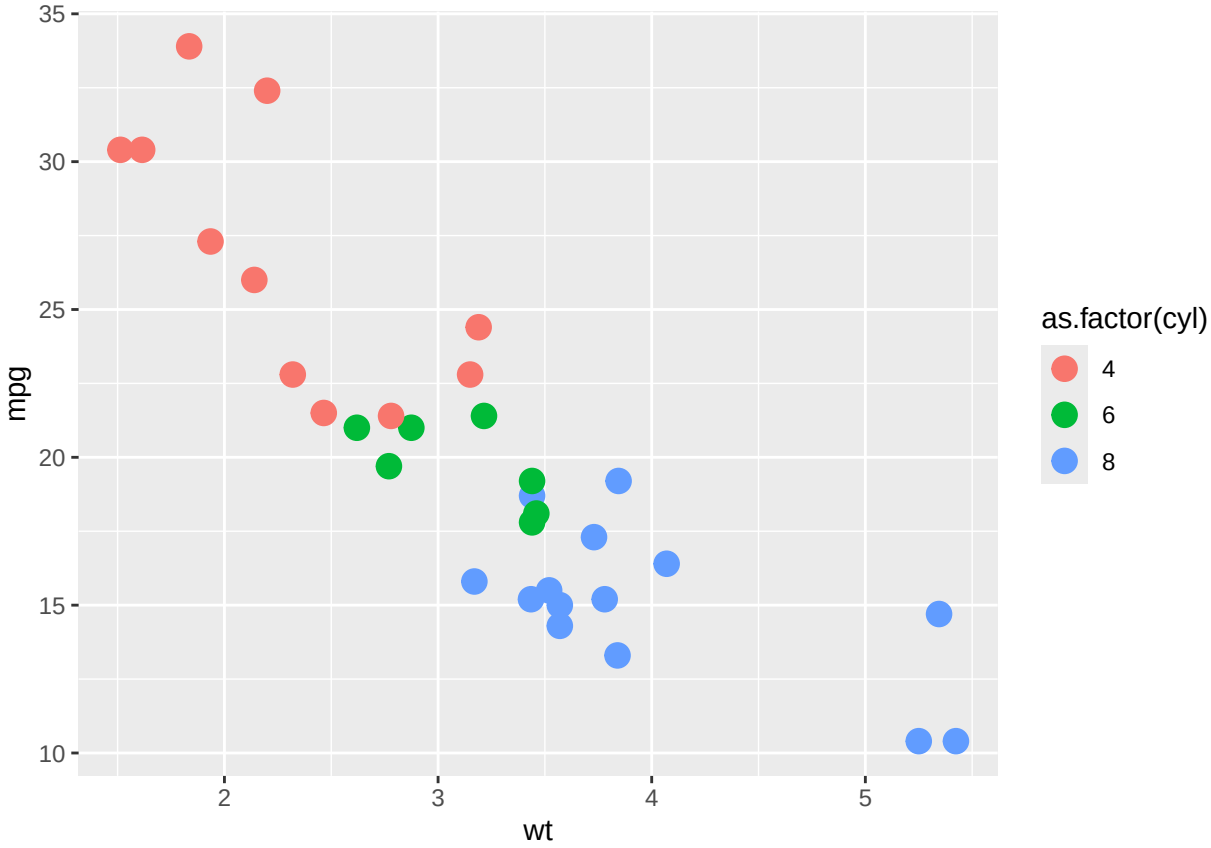
This is an R Markdown Notebook. When you execute code within the notebook, the results appear beneath the code.

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

```
plot(cars)
```




```
library(ggplot2)
ggplot(data=mtcars, aes(x=wt, y=mpg)) +
  geom_point(aes(colour = as.factor(cyl)), size=4)
```



Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Cmd+Option+I* (on Mac) or *Ctrl+Alt+I* (on Windows).

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")`.

1.1 Markdown:

$$\alpha = \beta = \frac{-\alpha \times \theta^3}{\sqrt{2 \times \theta}}$$

2 Basic of learning

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

2.1 How to Print

```
print('This is R programming course!')
```

```
## [1] "This is R programming course!"
```

2.2 Data Types (Classes)

In R, data types (or classes) define the kind of data stored in variables. Here are the most common data types in R:

Class	Description	Example	Code Example
`numeric`	Represents decimal or whole numbers	3.14, 42, -7.8	`x <- 3.14; class(x)`
`character`	Represents text strings	"Hello", "R Programming"	`z <- "Hello"; class(z)`
`logical`	Represents Boolean values (TRUE or FALSE)	TRUE, FALSE	`is_valid <- TRUE; class(is_valid)`
`complex`	Represents complex numbers	2+3i, 1-4i	`c <- 2 + 3i; class(c)`
`list`	Represents a collection of different types	A list of numbers, text	`lst <- list(1, "Hello", TRUE); class(lst)`

```
print(class(3.14))
```

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

```
print(class('R programming'))
```

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

```
print(class(TRUE))
```

```
## [1] "logical"
```

2.3 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
```

```
9 ** 0.5
```

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

```
log(2)
```

```
## [1] 0.6931472
```

```
log10(2)
```

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

```
14 %% 4
```

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

```
14 %/% 4
```

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

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

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

In R programming, code runs line by line, with only the last assignment determining the final value of a variable.

```
x <- 18
x <- 10
x <- x * 2 - 3
x
```

```
## [1] 17
```

We can save values in variables:

```
x <- 18
class(x)
```

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

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

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

2.4 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 = 6
assign('x', 2)

x < y
```

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

```
x >= y
```

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

```
x != y
```

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

```
x == y
```

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

```
b = 2
x <= b
```

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

```
x > 1 & y < 10
```

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

```
x > 1 & y > 10
```

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

```
x > 1 | y > 10
```

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

you can use below command to get special values:

```
x <- pi
x
```

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

```
e = exp(1)
e
```

```
## [1] 2.718282
```

```
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 is line is comment!

r = 0.2    # interest rate
```

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

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

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

```
## [1] 1 5 9
```

This work like *linspace* in numpy pakage in Python

```
x <- seq(1, 10, length=3)
x
```

```
## [1]  1.0  5.5 10.0
```

Replicate function:

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

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

```r
y <- rep(x, time=5)
y
```

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

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

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

2.4.1 Practice:

Simulate GDP growth from the year 2000 to 2025 with an annual growth rate of 3% starting from 1000 units.

$$GDP_t = 1000 \cdot (1 + r)^n$$

```r
years = 0:25
growth_rate = 0.03 # 3% annual rate
GDP <- 1000 * (1+growth_rate)^years
print(GDP)
```

```
##  [1] 1000.000 1030.000 1060.900 1092.727 1125.509 1159.274 1194.052 1229.874
##  [9] 1266.770 1304.773 1343.916 1384.234 1425.761 1468.534 1512.590 1557.967
## [17] 1604.706 1652.848 1702.433 1753.506 1806.111 1860.295 1916.103 1973.587
## [25] 2032.794 2093.778
```

```r
years = 2000:2025
growth_rate = 0.03 # 3% annual rate
GDP <- 1000 * (1+growth_rate)^(years-2000)
print(GDP)
```

```
##  [1] 1000.000 1030.000 1060.900 1092.727 1125.509 1159.274 1194.052 1229.874
##  [9] 1266.770 1304.773 1343.916 1384.234 1425.761 1468.534 1512.590 1557.967
## [17] 1604.706 1652.848 1702.433 1753.506 1806.111 1860.295 1916.103 1973.587
## [25] 2032.794 2093.778
```

2.5 Loops

2.5.1 if , elif

```r
age <- 14

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 can NOT vote yet!"
## [1] "You can will vote after 4 years."
```

```r
age = 16

if (age <= 4){
  price = 0
```

```
} else if (age < 16){
  price = 5
} else {
#} else if (age >= 16){
  price = 10
}

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

```
## [1] "Your cost is $ 10 ."
```

2.5.2 for loops

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

}
```

```
## [1] 1
## [1] 4
## [1] 27
## [1] 256
## [1] 3125
```

```
z = 0
for (i in 1:10){
  z = z + i
  print(z)
}
```

```
## [1] 1
## [1] 3
## [1] 6
## [1] 10
## [1] 15
## [1] 21
## [1] 28
## [1] 36
## [1] 45
## [1] 55
```

2.5.3 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
## [1] 10
```

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
```

2.5.4 Practice:

Write a conditional statement to check whether the variable is positive, negative, or zero and print the appropriate message.

```
x = 0
if (x > 0){
  print("Positive")
} else if (x < 0){
  print('negative')
} else {
  print("Zero")
}
```

```
## [1] "Zero"
```

2.6 function

```
average = function(a, b, c){
  summ = a + b + c
```

```r
  ave = summ / 3
  return(ave)
}

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

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

$$K_n = K \times (1 + r)^n$$

```r
future_value = function(k, r, n){
  k_n = k * (1+r)^n
  return(k_n)
}
```

```r
future_value(1000, 0.05, 10)
```

```
## [1] 1628.895
```

```r
future_value(500, 0.10, 5)
```

```
## [1] 805.255
```

2.6.1 Practice I:

Write a function `temp()` that converts a temperature in Celsius to Fahrenheit.

$$F = (C \times \frac{9}{5}) + 32$$

2.6.2 Practice II:

Write a function in R that takes a number as input and returns whether the number is even or odd.

3 Vetors

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

```
x <- c(10.25, 3.5, 8.75, 25, 12)
x
```

```
## [1] 10.25  3.50  8.75 25.00 12.00
```

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

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

```
x <- c(10.25, 3.5, 8.75, 25, 12, "A", 'b', "C")
x
```

```
## [1] "10.25" "3.5"   "8.75"  "25"    "12"    "A"     "b"     "C"
```

```
class(x)
```

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

Join 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(10.25, 3.5, 8.75, 25, 12)
x
```

```
## [1] 10.25  3.50  8.75 25.00 12.00
```

```
length(x)
```

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

```
length(z)
```

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

3.1 Vector Indexing

```
x
```

```
## [1] 10.25  3.50  8.75 25.00 12.00
```

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

```
## [1] 10.25
## [1] 3.5
## [1] 8.75
## [1] 25
## [1] 12
```

Use for loops for access to elemets of vectors

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

```
## [1] 10.25
## [1] 3.5
## [1] 8.75
## [1] 25
## [1] 12
```

Use index:

```
x
```

```
## [1] 10.25  3.50  8.75 25.00 12.00
```

```
x[2]
```

```
## [1] 3.5
```

```
x[-2]
```

```
## [1] 10.25  8.75 25.00 12.00
```

```
x[1:3]
```

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

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

```
## [1] 10.25  8.75 25.00
```

```
x[2] = -18
```

```
x
```

```
## [1]  10.25 -18.00   8.75  25.00  12.00
```

```
x[-3]
```

```
## [1]  10.25 -18.00  25.00  12.00
```

```
x[-3] = 0
x
```

```
## [1] 0.00 0.00 8.75 0.00 0.00
```

```
x <- c(10.25, 3.5, 8.75, 25, 12)
x
```

```
## [1] 10.25  3.50  8.75 25.00 12.00
```

```
y <- c(TRUE, FALSE, TRUE, TRUE, FALSE)
x[y]
```

```
## [1] 10.25  8.75 25.00
```

3.2 Matching Operator

```
x <- c(10,45,30,56,40,80)
x
```

```
## [1] 10 45 30 56 40 80
```

```
30 %in% x
```

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

```
37 %in% x
```

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

```
y <- c(30, 37, 40)
y %in% x
```

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

3.3 Vector Arithmetic's

```
x
```

```
## [1] 10 45 30 56 40 80
```

```
x + 2
```

```
## [1] 12 47 32 58 42 82
```

```
x = x * 2
x
```

```
## [1]  20  90  60 112  80 160
```

```
sqrt(x)
```

```
## [1]  4.472136  9.486833  7.745967 10.583005  8.944272 12.649111
```

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

x + y
```

```
## [1] 15 46 33 54
```

```
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

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

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

```
sum(x)
```

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

```
prod(x)
```

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

```r
rev(x)
```

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

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

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

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

x %*% y
```

```
##      [,1]
## [1,]  385
```

3.5 Logical Vector

```r
x
```

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

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

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

```r
x[y]
```

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

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

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

3.6 Factors

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

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

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

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

```
## [1] Male   Female Male   Male   Female
## Levels: Female Male
```

```r
table(x)
```

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

3.7 Mathematical Function in R

```
x <- c(4.235, -3.548, 5.324, 7.892)
x
```

```
## [1]  4.235 -3.548  5.324  7.892
```

```
abs(x)
```

```
## [1] 4.235 3.548 5.324 7.892
```

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

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

```
floor(x)  # smaller integer
```

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

```
round(x)
```

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

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

```
## [1]  4.24 -3.55  5.32  7.89
```

```
sqrt(abs(x))
```

```
## [1] 2.057912 1.883614 2.307379 2.809270
```

```
log10(abs(x))
```

```
## [1] 0.6268534 0.5499836 0.7262380 0.8971871
```

3.8 Random Number in R

```
x <- rnorm(1000)
#x
```

```
mean(x)
```

```
## [1] -0.04729093
```

```
sd(x)
```

```
## [1] 1.01769
```

3.9 Practice:

Given a list of students (`"Alice", "Bob", "Charlie", "David", "Eve"`) and their corresponding scores (`85, 92, 78, 55, 88`), extract the names and scores of students who passed (`score >= 60`). Also, calculate the mean score of the students who passed.

```
students <- c("Alice", "Bob", "Charlie", "David", "Eve")
scores <- c(85, 92, 78, 55, 88)

passed <- scores >= 60

scores[passed]
```

```
## [1] 85 92 78 88
```

```
students[passed]
```

```
## [1] "Alice"   "Bob"     "Charlie" "Eve"
```

```
mean(scores[passed])
```

```
## [1] 85.75
```

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)
```

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

```
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
```

```
diag(4)
```

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

```
diag(1:5)
```

```
##      [,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(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
```

```
rownames(m) = c('A', "b", 'F', 'S')
colnames(m) = c('B', "W", 'X', 'S', "L")
```

```
m
```

```
##    B  W  X  S  L
## A  1  3  5  7  9
## b 11 13 15 17 19
## F 21 23 25 27 29
## S 31 33 35 37 39
```

4.4 Matrix Indexing

Indexing in R programming is similar to Python.

```
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
```

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

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

```
m[,3] # for get single column
```

```
## [1]  5 15 25 35
```

```
m[2,3]
```

```
## [1] 15
```

```
m[2,2:4]
```

```
## [1] 13 15 17
```

```
m[1:3,2:4]
```

```
##      [,1] [,2] [,3]
## [1,]    3    5    7
## [2,]   13   15   17
## [3,]   23   25   27
```

```
m[,-2]
```

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

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

```
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
```

You can change values in matrix.

```
m[2,3] = 0
m
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    3    5    7    9
## [2,]   11   13    0   17   19
## [3,]   21   23   25   27   29
## [4,]   31   33   35   37   39
```

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

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

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

```
##      [,1] [,2] [,3]
## [1,]    1    2    3
## [2,]    4    5    6
## [3,]    8    9    1
## [4,]   10   11   12
```

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

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

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    2    3   10
## [2,]    4    5    6   11
## [3,]    8    9    1   12
```

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
```

```
A %*% B
```

```
##      [,1] [,2] [,3]
## [1,]   26    8   10
## [2,]   62   20   25
## [3,]   65   27   27
```

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)

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
```

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

```
rowSums(A)
```

```
## [1] 10 28 10
```

```
colSums((A))
```

```
## [1]  7 12 13 16
```

```
rowMeans((A))
```

```
## [1] 2.5 7.0 2.5
```

```
colMeans(A)
```

```
## [1] 2.333333 4.000000 4.333333 5.333333
```

4.6.1 Practice I

Create a 4x4 matrix of random integers between 1 and 100.

- Print the matrix.
- Calculate the row-wise sum and column-wise mean.
- Check if the matrix is symmetric by comparing it to its transpose.

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

```
id <- c(101,102,103)
name <- c("Sanaz", "Saeed", "Sarah")
scores <- c(98.45, 45.65, 78.79)

students = list(id, names, scores)
students
```

```
## [[1]]
## [1] 101 102 103
##
## [[2]]
## function (x)  .Primitive("names")
##
## [[3]]
## [1] 98.45 45.65 78.79
```

```
students[2]
```

```
## [[1]]
## function (x)  .Primitive("names")
```

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(101,102,103,104,105)
names <- c("Sanaz", 'Saeed', 'James', "Peter", 'Sarah')
scores <- c(98.5, 45.65, 78.79, 56.32, 87.3)
students <- data.frame(id, names, scores)
students
```

```
##    id names scores
## 1 101 Sanaz  98.50
## 2 102 Saeed  45.65
## 3 103 James  78.79
## 4 104 Peter  56.32
## 5 105 Sarah  87.30
```

6.2 Dataframes Indexing

```
students
```

```
##    id names scores
## 1 101 Sanaz  98.50
## 2 102 Saeed  45.65
## 3 103 James  78.79
## 4 104 Peter  56.32
## 5 105 Sarah  87.30
```

```
students[1,]
```

```
##    id names scores
## 1 101 Sanaz   98.5
```

```
students[,2]
```

```
## [1] "Sanaz" "Saeed" "James" "Peter" "Sarah"
```

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

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

```
students$scores
```

```
## [1] 98.50 45.65 78.79 56.32 87.30
```

```
students$names[3]
```

```
## [1] "James"
```

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

```
##   names scores
```

```
## 1 Sanaz  98.50
## 2 Saeed  45.65
## 3 James  78.79
## 4 Peter  56.32
## 5 Sarah  87.30
```

6.3 Dataframes `subset()` function for filtering

```
students
```

```
##    id names scores
## 1 101 Sanaz  98.50
## 2 102 Saeed  45.65
## 3 103 James  78.79
## 4 104 Peter  56.32
## 5 105 Sarah  87.30
```

```
report <- subset(students, scores < 60)
report
```

```
##    id names scores
## 2 102 Saeed  45.65
## 4 104 Peter  56.32
```

```
report <- subset(students, scores < 60 & id <=103)
report
```

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

```
report <- subset(students, scores < 60 | id <=103)
report
```

```
##    id names scores
## 1 101 Sanaz  98.50
## 2 102 Saeed  45.65
## 3 103 James  78.79
## 4 104 Peter  56.32
```

```
report <- subset(students, scores < 60, select = c(names))
report
```

```
##   names
## 2 Saeed
## 4 Peter
```

```
report <- subset(students, scores < 60, select = c(names, id))
report
```

```
##   names  id
## 2 Saeed 102
## 4 Peter 104
```

```
report <- subset(students, scores < 60, select = c(-names))
report
```

```
##    id scores
## 2 102  45.65
```

```
## 4 104  56.32
```

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

```
students
```

```
##    id names scores
## 1 101 Sanaz  98.50
## 2 102 Saeed  45.65
## 3 103 James  78.79
## 4 104 Peter  56.32
## 5 105 Sarah  87.30
```

add rows

```
students <- rbind(students, data.frame(id=106, names="Ema", scores=68.12))
students
```

```
##    id names scores
## 1 101 Sanaz  98.50
## 2 102 Saeed  45.65
## 3 103 James  78.79
## 4 104 Peter  56.32
## 5 105 Sarah  87.30
## 6 106   Ema  68.12
```

add columns

```
students <- cbind(students, age = c(18,24,19,26,34,23))
```

```
students
```

```
##    id names scores age
## 1 101 Sanaz  98.50  18
## 2 102 Saeed  45.65  24
## 3 103 James  78.79  19
## 4 104 Peter  56.32  26
## 5 105 Sarah  87.30  34
## 6 106   Ema  68.12  23
```

6.5 Saving data in csv

```
students
```

```
##    id names scores age
## 1 101 Sanaz  98.50  18
## 2 102 Saeed  45.65  24
## 3 103 James  78.79  19
## 4 104 Peter  56.32  26
## 5 105 Sarah  87.30  34
## 6 106   Ema  68.12  23
```

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

6.6 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

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

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

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

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

```
x[!y]
```

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

```
weather <- data.frame(
  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
```

```
##    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
```

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

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

```
weather[weatherNA,]
```

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

Mean

```
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
```

```
weather$temperature[is.na(weather$temperature)] <- mean(weather$temperature, na.rm = TRUE)
weather
```

```
##    id temperature wind humidity
## 1 101      25.800   78       25
## 2 102      34.200   59       45
## 3 103      26.975   63       85
## 4 104      27.400   40       NA
## 5 105      20.500   68       61
```

fix value

```
weather$humidity[is.na(weather$humidity)] <- 60
weather
```

```
##    id temperature wind humidity
## 1 101      25.800   78       25
## 2 102      34.200   59       45
## 3 103      26.975   63       85
## 4 104      27.400   40       60
## 5 105      20.500   68       61
```

interpolation

```
# Install and load imputeTS package
# install.packages("imputeTS")
library(imputeTS)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
# Fill missing values with linear interpolation
weather <- data.frame(
  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
```

```
##    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
```

```
weather$temperature <- na_interpolation(weather$temperature)
weather$humidity <- na_interpolation(weather$humidity)
weather
```

```
##    id temperature wind humidity
## 1 101        25.8   78       25
```

```
## 2 102        34.2   59       45
## 3 103        30.8   63       85
## 4 104        27.4   40       73
## 5 105        20.5   68       61
```

```
print(weather)
```

```
##    id temperature wind humidity
## 1 101        25.8   78       25
## 2 102        34.2   59       45
## 3 103        30.8   63       85
## 4 104        27.4   40       73
## 5 105        20.5   68       61
```

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
```

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 (`,`))

```
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, 10)
```

```
##     X          state abb        region population PopulationDensity murders
## 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
## 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
```

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
```

```
##     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
```

```
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
```

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

```
##             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

```
head(df)
```

```
##   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
## 6 5     Florida  FL     South   19687653            360.20     987        669
##   gunownership
## 1        0.517
## 2        0.311
## 3        0.213
## 4        0.347
## 5        0.167
## 6        0.245
```

```
names(df)
```

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

```
dim(df)
```

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

```
dfprime <- filter(df, murders > 500)
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  6      Georgia  GA         South    9920000            172.50     527
## 4 13     Michigan  MI North Central    9883640            174.80     558
## 5 16     New York  NY     Northeast   19378102            415.30     860
## 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        376        0.403
## 4        413        0.384
## 5        517        0.180
```

```
## 6        457        0.347
## 7        805        0.359
```

```
dim(dfprime)
```

```
## [1] 7 9
```

```
dfprime <- filter(df, murders > 500 & population > 10000000)
dfprime
```

```
##    X        state abb    region population PopulationDensity murders gunmurders
## 1  2   California  CA      West   37253956             244.20    1811        1257
## 2  5      Florida  FL     South   19687653             360.20     987         669
## 3 16     New York  NY Northeast   19378102             415.30     860         517
## 4 20 Pennsylvania  PA Northeast   12702379             285.30     646         457
## 5 22        Texas  TX     South   25145561              98.07    1246         805
##   gunownership
## 1        0.213
## 2        0.245
## 3        0.180
## 4        0.347
## 5        0.359
```

```
dim(dfprime)
```

```
## [1] 5 9
```

dplyr `arrange()` function

```
head(df)
```

```
##   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
## 6 5     Florida  FL     South   19687653             360.20     987         669
##   gunownership
## 1        0.517
## 2        0.311
## 3        0.213
## 4        0.347
## 5        0.167
## 6        0.245
```

```
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(population))
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   7       Illinois  IL North Central   12830632            231.90     453
## 6  20   Pennsylvania  PA     Northeast   12702379            285.30     646
## 7  18           Ohio  OH North Central   11536504            282.50     460
## 8   6        Georgia  GA         South    9920000            172.50     527
## 9  13       Michigan  MI North Central    9883640            174.80     558
## 10 17 North Carolina  NC         South    9535483            200.60     445
## 11 15     New Jersey  NJ     Northeast    8791894           1189.00     363
## 12 23       Virginia  VA         South    8001024            207.30     369
## 13 12  Massachusetts  MA     Northeast    6547629            852.10     209
## 14  8        Indiana  IN North Central    6483802            182.50     198
## 15  1        Arizona  AZ          West    6392017             57.05     352
## 16 21      Tennessee  TN         South    6346105            156.60     356
## 17 14       Missouri  MO North Central    5988927             87.26     419
## 18 11       Maryland  MD         South    5773552            606.20     424
## 19 24      Wisconsin  WI North Central    5686986            105.20     151
## 20  3       Colorado  CO          West    5029196             49.33     117
## 21  0        Alabama  AL         South    4779736             94.65     199
## 22 10      Louisiana  LA         South    4533372            105.00     437
## 23  9       Kentucky  KY         South    4339367            110.00     180
## 24 19       Oklahoma  OK         South    3751351             55.22     188
## 25  4    Connecticut  CT     Northeast    3574097            741.40     131
##    gunmurders gunownership
## 1        1257        0.213
## 2         805        0.359
## 3         669        0.245
## 4         517        0.180
## 5         364        0.202
## 6         457        0.347
## 7         310        0.324
## 8         376        0.403
## 9         413        0.384
## 10        286        0.413
## 11        246        0.123
## 12        250        0.351
## 13        118        0.126
## 14        142        0.391
## 15        232        0.311
## 16        219        0.439
## 17        321        0.417
## 18        293        0.213
## 19         97        0.444
## 20         65        0.347
## 21        135        0.517
## 22        351        0.441
## 23        116        0.477
## 24        111        0.429
## 25         97        0.167
```

7.3 dplyr rename() function

```
names(df)
```

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

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

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

```
## 18     445        286        0.413
## 19     460        310        0.324
## 20     188        111        0.429
## 21     646        457        0.347
## 22     356        219        0.439
## 23    1246        805        0.359
## 24     369        250        0.351
## 25     151         97        0.444
```

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

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

```
## 17      860        517        0.180
## 18      445        286        0.413
## 19      460        310        0.324
## 20      188        111        0.429
## 21      646        457        0.347
## 22      356        219        0.439
## 23     1246        805        0.359
## 24      369        250        0.351
## 25      151         97        0.444
```

```
names(df2)
```

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

7.4 dplyr `mutate()` function

```
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"
```

7.4.1 Practice:

Import the data and create a new column called murder_rate that shows the murder rate (murders per million people)

```
dfprime <- mutate(df, mpopulation = population / 10^6)
dfprime <- mutate(dfprime, murder_rate_m = murders / mpopulation)
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 mpopulation murder_rate_m
## 1         135        0.517    4.779736      41.63410
## 2         232        0.311    6.392017      55.06869
## 3        1257        0.213   37.253956      48.61229
## 4          65        0.347    5.029196      23.26416
## 5          97        0.167    3.574097      36.65261
## 6         669        0.245   19.687653      50.13294
## 7         376        0.403    9.920000      53.12500
## 8         364        0.202   12.830632      35.30613
## 9         142        0.391    6.483802      30.53764
## 10        116        0.477    4.339367      41.48070
## 11        351        0.441    4.533372      96.39624
## 12        293        0.213    5.773552      73.43833
## 13        118        0.126    6.547629      31.91995
## 14        413        0.384    9.883640      56.45693
## 15        321        0.417    5.988927      69.96245
## 16        246        0.123    8.791894      41.28803
## 17        517        0.180   19.378102      44.37999
## 18        286        0.413    9.535483      46.66780
## 19        310        0.324   11.536504      39.87343
## 20        111        0.429    3.751351      50.11528
## 21        457        0.347   12.702379      50.85662
## 22        219        0.439    6.346105      56.09740
## 23        805        0.359   25.145561      49.55149
## 24        250        0.351    8.001024      46.11910
## 25         97        0.444    5.686986      26.55185
```

```
dfprime <- mutate(df, murder_rate = murders / population * 10^6)
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 murder_rate
## 1         135        0.517    41.63410
## 2         232        0.311    55.06869
## 3        1257        0.213    48.61229
## 4          65        0.347    23.26416
## 5          97        0.167    36.65261
## 6         669        0.245    50.13294
## 7         376        0.403    53.12500
## 8         364        0.202    35.30613
## 9         142        0.391    30.53764
## 10        116        0.477    41.48070
## 11        351        0.441    96.39624
## 12        293        0.213    73.43833
## 13        118        0.126    31.91995
## 14        413        0.384    56.45693
## 15        321        0.417    69.96245
## 16        246        0.123    41.28803
## 17        517        0.180    44.37999
## 18        286        0.413    46.66780
## 19        310        0.324    39.87343
## 20        111        0.429    50.11528
## 21        457        0.347    50.85662
## 22        219        0.439    56.09740
## 23        805        0.359    49.55149
## 24        250        0.351    46.11910
## 25         97        0.444    26.55185
```

7.5 dplyr group_by() function

```
head(df)
```

```
##   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
## 6 5     Florida  FL     South   19687653            360.20     987        669
##   gunownership
## 1        0.517
## 2        0.311
## 3        0.213
## 4        0.347
```

```
## 5        0.167
## 6        0.245
```

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

```
## # A tibble: 25 x 9
## # Groups:   region [4]
##       X state      abb   region population PopulationDensity 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 Connectic~ 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
## # i 15 more rows
## # i 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
```

```
write.csv(summarise(dfprime, sum(murders), mean(gunownership), median(population)), file = 'report_1.csv
```

7.6 dplyr Pipe Operator %>%

```
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
```

```
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
```

Import the data and create a new column called murder_rate that shows the murder rate (murders per

million people)

```
df %>% mutate(murder_rate = murders / population * 10^6) %>% arrange(desc(murder_rate))
```

```
##     X          state abb        region population PopulationDensity murders
## 1  10      Louisiana  LA         South    4533372             105.00     437
## 2  11       Maryland  MD         South    5773552             606.20     424
## 3  14       Missouri  MO North Central    5988927              87.26     419
## 4  13       Michigan  MI North Central    9883640             174.80     558
## 5  21      Tennessee  TN         South    6346105             156.60     356
## 6   1        Arizona  AZ          West    6392017              57.05     352
## 7   6        Georgia  GA         South    9920000             172.50     527
## 8  20   Pennsylvania  PA     Northeast   12702379             285.30     646
## 9   5        Florida  FL         South   19687653             360.20     987
## 10 19       Oklahoma  OK         South    3751351              55.22     188
## 11 22          Texas  TX         South   25145561              98.07    1246
## 12  2     California  CA          West   37253956             244.20    1811
## 13 17 North Carolina  NC         South    9535483             200.60     445
## 14 23       Virginia  VA         South    8001024             207.30     369
## 15 16       New York  NY     Northeast   19378102             415.30     860
## 16  0        Alabama  AL         South    4779736              94.65     199
## 17  9       Kentucky  KY         South    4339367             110.00     180
## 18 15     New Jersey  NJ     Northeast    8791894            1189.00     363
## 19 18           Ohio  OH North Central   11536504             282.50     460
## 20  4    Connecticut  CT     Northeast    3574097             741.40     131
## 21  7       Illinois  IL North Central   12830632             231.90     453
## 22 12  Massachusetts  MA     Northeast    6547629             852.10     209
## 23  8        Indiana  IN North Central    6483802             182.50     198
## 24 24      Wisconsin  WI North Central    5686986             105.20     151
## 25  3       Colorado  CO          West    5029196              49.33     117
##    gunmurders gunownership murder_rate
## 1         351        0.441    96.39624
## 2         293        0.213    73.43833
## 3         321        0.417    69.96245
## 4         413        0.384    56.45693
## 5         219        0.439    56.09740
## 6         232        0.311    55.06869
## 7         376        0.403    53.12500
## 8         457        0.347    50.85662
## 9         669        0.245    50.13294
## 10        111        0.429    50.11528
## 11        805        0.359    49.55149
## 12       1257        0.213    48.61229
## 13        286        0.413    46.66780
## 14        250        0.351    46.11910
## 15        517        0.180    44.37999
## 16        135        0.517    41.63410
## 17        116        0.477    41.48070
## 18        246        0.123    41.28803
## 19        310        0.324    39.87343
## 20         97        0.167    36.65261
## 21        364        0.202    35.30613
## 22        118        0.126    31.91995
## 23        142        0.391    30.53764
## 24         97        0.444    26.55185
```

```
## 25         65        0.347    23.26416
```

7.6.1 Practice:

Find the states with a population greater than 10 million and murders greater than 500. Display only the state, population, and murder columns.

```
result <- df %>% filter(population > 10^7, murders > 500) %>% select(state, population, murders) %>% ar
print(result)
```

```
##          state population murders
## 1 Pennsylvania   12702379     646
## 2     New York   19378102     860
## 3      Florida   19687653     987
## 4        Texas   25145561    1246
## 5   California   37253956    1811
```

```
result
```

```
##          state population murders
## 1 Pennsylvania   12702379     646
## 2     New York   19378102     860
## 3      Florida   19687653     987
## 4        Texas   25145561    1246
## 5   California   37253956    1811
```

8 Data Visualization with `dplyr`

8.1 Bar Graphs

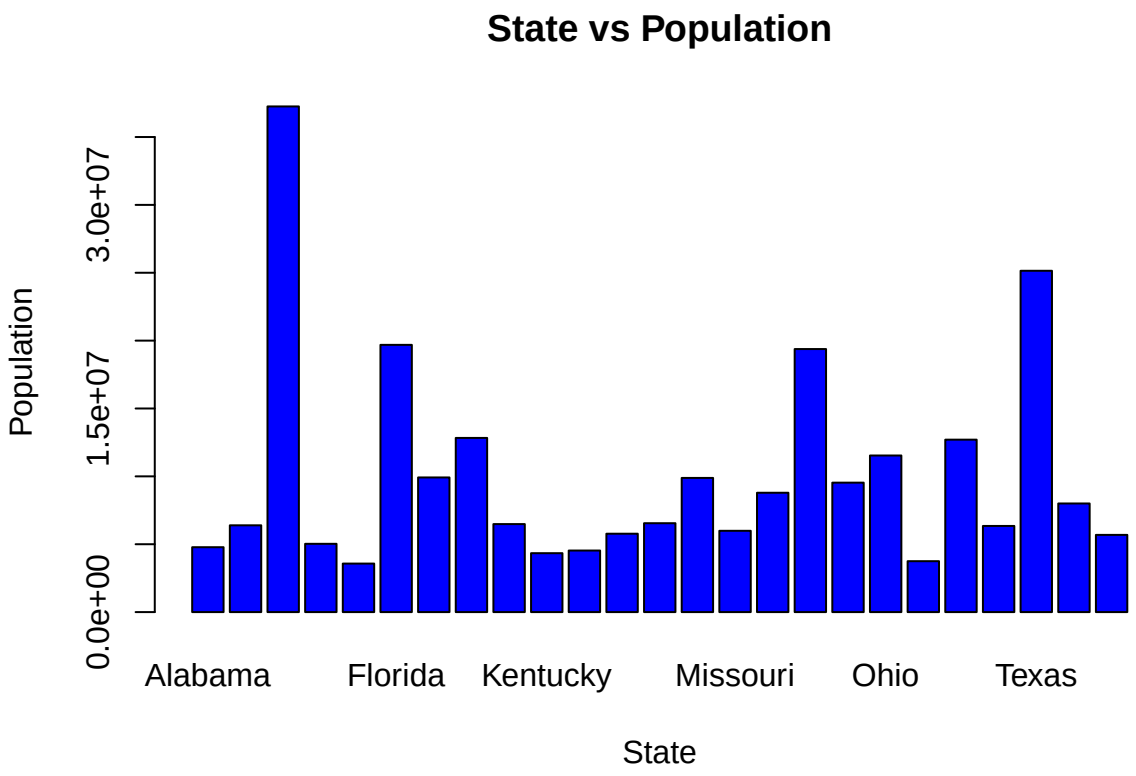
```
library(dplyr)
library(ggplot2)
names(df)
```

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

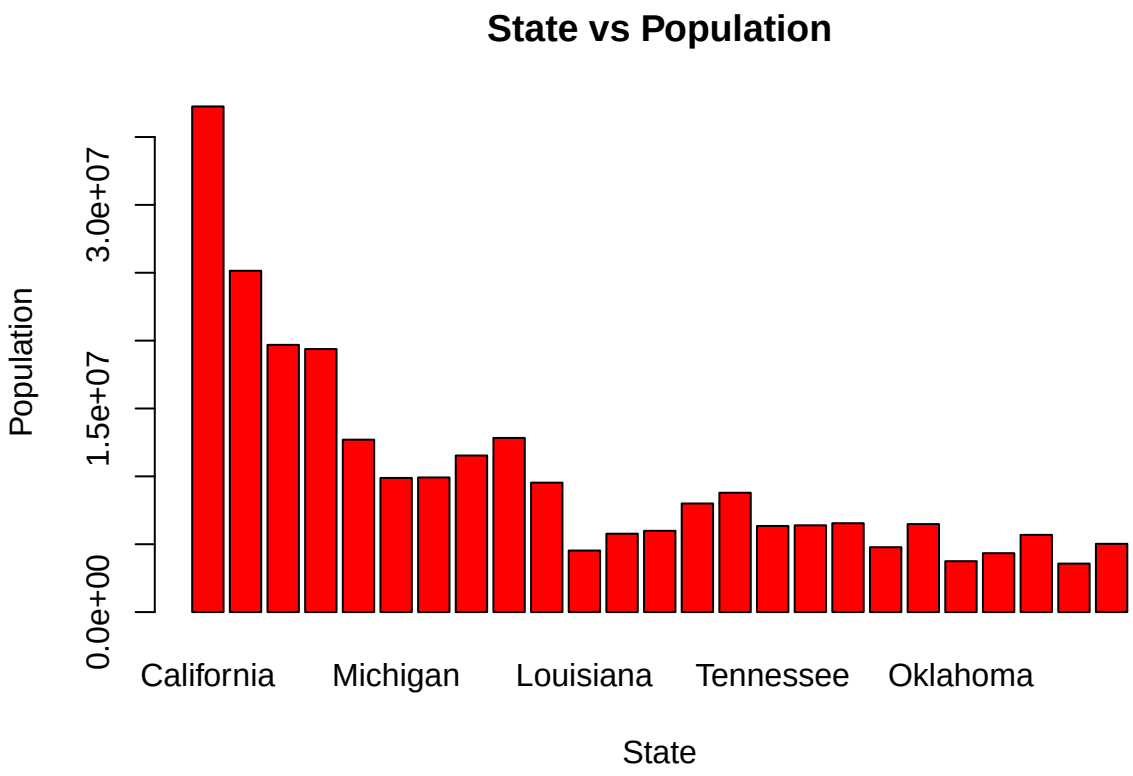
```
head(df)
```

```
##   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
## 6 5     Florida  FL     South   19687653             360.20     987        669
##   gunownership
## 1        0.517
## 2        0.311
## 3        0.213
## 4        0.347
## 5        0.167
## 6        0.245
```

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

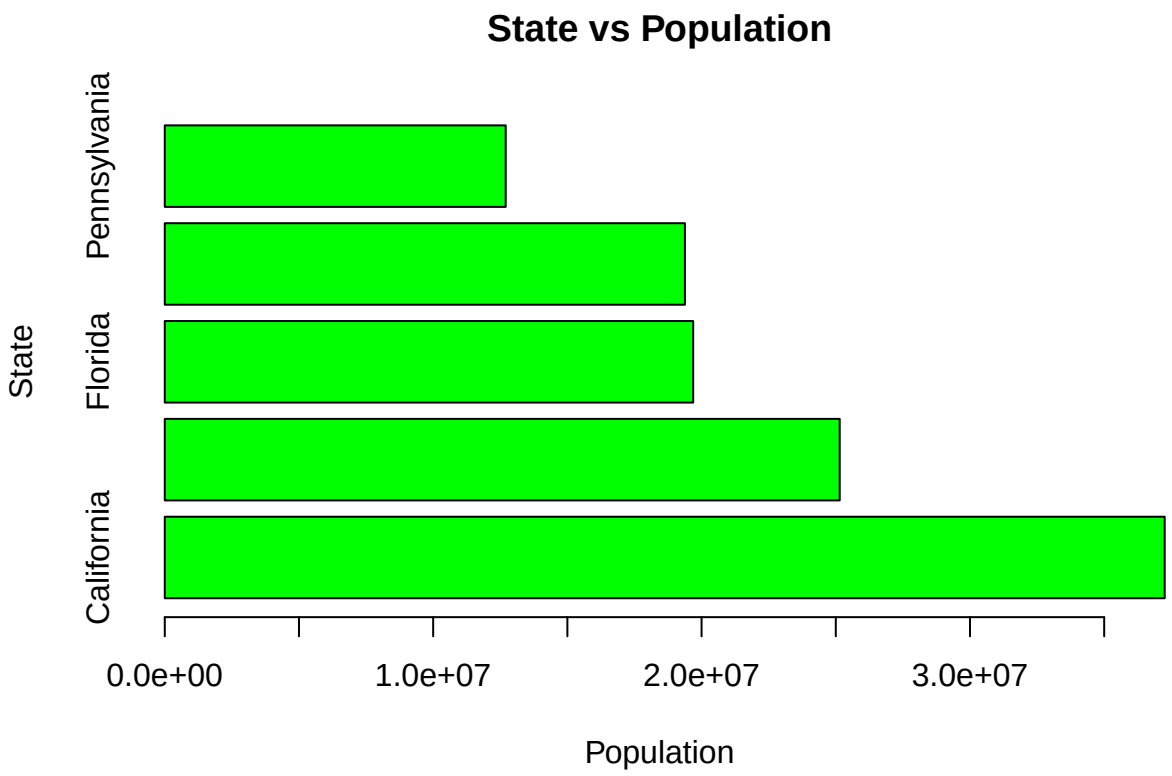



```
dfprime <- arrange(df, desc(murders)) #%>% head(5)

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



Horizontsl Bar Graphs

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

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



8.2 Histogram

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

```
##        X    Date      Price
## 1      0  1/1/70  74.253333
## 2      1  2/1/70  69.976842
## 3      2  3/1/70  72.158571
## 4      3  4/1/70  74.252727
## 5      4  5/1/70  66.665238
## 6      5  6/1/70  67.593182
## 7      6  7/1/70  72.696364
## 8      7  8/1/70  76.146667
## 9      8  9/1/70  80.462857
## 10     9 10/1/70  84.691818
## 11    10 11/1/70  86.329500
## 12    11 12/1/70  89.958182
## 13    12  1/1/71  96.691000
## 14    13  2/1/71 104.150000
## 15    14  3/1/71 109.943044
## 16    15  4/1/71 118.480476
## 17    16  5/1/71 120.598000
## 18    17  6/1/71  73.682727
## 19    18  7/1/71  57.966667
## 20    19  8/1/71  58.155455
## 21    20  9/1/71  62.643810
## 22    21 10/1/71  61.227619
## 23    22 11/1/71  57.150952
```

```
## 24   23 12/1/71  62.530000
## 25   24  1/1/72  63.198095
## 26   25  2/1/72  60.508500
## 27   26  3/1/72  63.352273
## 28   27  4/1/72  68.016500
## 29   28  5/1/72  67.691364
## 30   29  6/1/72  67.128636
## 31   30  7/1/72  64.687000
## 32   31  8/1/72  66.579565
## 33   32  9/1/72  65.593000
## 34   33 10/1/72  63.983182
## 35   34 11/1/72  66.254000
## 36   35 12/1/72  69.202105
## 37   36  1/1/73  71.974286
## 38   37  2/1/73  68.668421
## 39   38  3/1/73  66.208636
## 40   39  4/1/73  62.819500
## 41   40  5/1/73  59.308636
## 42   41  6/1/73  58.490952
## 43   42  7/1/73  59.882381
## 44   43  8/1/73  60.122174
## 45   44  9/1/73  60.303158
## 46   45 10/1/73  65.949130
## 47   46 11/1/73  64.060000
## 48   47 12/1/73  60.257500
## 49   48  1/1/74  62.394545
## 50   49  2/1/74  56.753158
## 51   50  3/1/74  54.965714
## 52   51  4/1/74  54.025238
## 53   52  5/1/74  49.764091
## 54   53  6/1/74  49.121000
## 55   54  7/1/74  47.059091
## 56   55  8/1/74  41.434545
## 57   56  9/1/74  34.645500
## 58   57 10/1/74  35.637826
## 59   58 11/1/74  36.889500
## 60   59 12/1/74  33.437619
## 61   60  1/1/75  35.321364
## 62   61  2/1/75  41.469474
## 63   62  3/1/75  46.196000
## 64   63  4/1/75  45.939545
## 65   64  5/1/75  46.538571
## 66   65  6/1/75  47.496667
## 67   66  7/1/75  50.144545
## 68   67  8/1/75  44.573810
## 69   68  9/1/75  44.275238
## 70   69 10/1/75  47.176087
## 71   70 11/1/75  48.332105
## 72   71 12/1/75  46.387727
## 73   72  1/1/76  51.776190
## 74   73  2/1/76  53.456842
## 75   74  3/1/76  52.165652
## 76   75  4/1/76  53.133333
## 77   76  5/1/76  51.614000
```

```
## 78   77  6/1/76  54.456364
## 79   78  7/1/76  56.990476
## 80   79  8/1/76  54.832273
## 81   80  9/1/76  54.710000
## 82   81 10/1/76  51.948095
## 83   82 11/1/76  51.533500
## 84   83 12/1/76  52.627727
## 85   84  1/1/77  53.602381
## 86   85  2/1/77  51.133684
## 87   86  3/1/77  50.790870
## 88   87  4/1/77  51.564500
## 89   88  5/1/77  54.615714
## 90   89  6/1/77  55.877727
## 91   90  7/1/77  55.358421
## 92   91  8/1/77  54.170435
## 93   92  9/1/77  53.026190
## 94   93 10/1/77  50.467143
## 95   94 11/1/77  51.038571
## 96   95 12/1/77  49.002381
## 97   96  1/1/78  46.687619
## 98   97  2/1/78  46.113684
## 99   98  3/1/78  46.257727
## 100  99  4/1/78  48.796500
## 101 100  5/1/78  52.423182
## 102 101  6/1/78  51.860455
## 103 102  7/1/78  52.133000
## 104 103  8/1/78  55.611304
## 105 104  9/1/78  53.384000
## 106 105 10/1/78  51.336818
## 107 106 11/1/78  48.697619
## 108 107 12/1/78  47.696000
## 109 108  1/1/79  49.173182
## 110 109  2/1/79  46.968421
## 111 110  3/1/79  47.485455
## 112 111  4/1/79  48.251500
## 113 112  5/1/79  49.417727
## 114 113  6/1/79  49.543810
## 115 114  7/1/79  50.472381
## 116 115  8/1/79  53.285652
## 117 116  9/1/79  51.476316
## 118 117 10/1/79  48.872174
## 119 118 11/1/79  47.020476
## 120 119 12/1/79  48.359500
## 121 120  1/1/80  53.474545
## 122 121  2/1/80  53.170500
## 123 122  3/1/80  47.615238
## 124 123  4/1/80  46.877143
## 125 124  5/1/80  47.526667
## 126 125  6/1/80  50.275238
## 127 126  7/1/80  54.053182
## 128 127  8/1/80  56.304762
## 129 128  9/1/80  54.062381
## 130 129 10/1/80  53.426957
## 131 130 11/1/80  58.217778
```

```
## 132 131 12/1/80  59.013182
## 133 132  1/1/81  61.805714
## 134 133  2/1/81  62.830000
## 135 134  3/1/81  66.560000
## 136 135  4/1/81  66.831429
## 137 136  5/1/81  64.642500
## 138 137  6/1/81  65.350909
## 139 138  7/1/81  61.150000
## 140 139  8/1/81  58.496190
## 141 140  9/1/81  54.233810
## 142 141 10/1/81  55.400909
## 143 142 11/1/81  56.659000
## 144 143 12/1/81  58.797727
## 145 144  1/1/82  58.102000
## 146 145  2/1/82  61.542632
## 147 146  3/1/82  61.122609
## 148 147  4/1/82  64.355714
## 149 148  5/1/82  63.350000
## 150 149  6/1/82  61.706818
## 151 150  7/1/82  66.611429
## 152 151  8/1/82  67.890455
## 153 152  9/1/82  76.098571
## 154 153 10/1/82  82.146667
## 155 154 11/1/82  89.705238
## 156 155 12/1/82  95.037727
## 157 156  1/1/83  95.754286
## 158 157  2/1/83 103.378421
## 159 158  3/1/83 104.911304
## 160 159  4/1/83 108.154500
## 161 160  5/1/83 108.135238
## 162 161  6/1/83  57.582273
## 163 162  7/1/83  52.961000
## 164 163  8/1/83  48.529565
## 165 164  9/1/83  51.168095
## 166 165 10/1/83  53.441429
## 167 166 11/1/83  54.543810
## 168 167 12/1/83  57.370476
## 169 168  1/1/84  56.764286
## 170 169  2/1/84  53.302000
## 171 170  3/1/84  52.133182
## 172 171  4/1/84  53.915500
## 173 172  5/1/84  53.985000
## 174 173  6/1/84  53.145714
## 175 174  7/1/84  50.693333
## 176 175  8/1/84  57.040435
## 177 176  9/1/84  56.646842
## 178 177 10/1/84  55.953043
## 179 178 11/1/84  57.240476
## 180 179 12/1/84  55.552500
## 181 180  1/1/85  60.047273
## 182 181  2/1/85  63.343684
## 183 182  3/1/85  62.049524
## 184 183  4/1/85  59.687143
## 185 184  5/1/85  59.798182
```

```
## 186 185  6/1/85  60.915000
## 187 186  7/1/85  62.297727
## 188 187  8/1/85  61.723636
## 189 188  9/1/85  59.489474
## 190 189 10/1/85  58.714348
## 191 190 11/1/85  63.269000
## 192 191 12/1/85  68.931429
## 193 192  1/1/86  69.951818
## 194 193  2/1/86  74.544211
## 195 194  3/1/86  76.841500
## 196 195  4/1/86  77.627727
## 197 196  5/1/86  77.694286
## 198 197  6/1/86  80.920000
## 199 198  7/1/86  76.197273
## 200 199  8/1/86  75.307143
## 201 200  9/1/86  74.324762
## 202 201 10/1/86  74.769130
## 203 202 11/1/86  78.300000
## 204 203 12/1/86  86.406364
## 205 204  1/1/87  94.157619
## 206 205  2/1/87 100.839474
## 207 206  3/1/87 106.644546
## 208 207  4/1/87 105.295238
## 209 208  5/1/87  93.260500
## 210 209  6/1/87  54.081818
## 211 210  7/1/87  56.139545
## 212 211  8/1/87  62.637143
## 213 212  9/1/87  60.895238
## 214 213 10/1/87  53.081364
## 215 214 11/1/87  44.996000
## 216 215 12/1/87  43.826364
## 217 216  1/1/88  44.877500
## 218 217  2/1/88  43.415500
## 219 218  3/1/88  43.253043
## 220 219  4/1/88  40.796500
## 221 220  5/1/88  39.919048
## 222 221  6/1/88  43.160909
## 223 222  7/1/88  42.883500
## 224 223  8/1/88  40.610870
## 225 224  9/1/88  42.294286
## 226 225 10/1/88  43.770476
## 227 226 11/1/88  44.031905
## 228 227 12/1/88  45.222857
## 229 228  1/1/89  45.406667
## 230 229  2/1/89  46.838421
## 231 230  3/1/89  45.036818
## 232 231  4/1/89  46.477500
## 233 232  5/1/89  51.030000
## 234 233  6/1/89  54.052727
## 235 234  7/1/89  54.752500
## 236 235  8/1/89  57.610870
## 237 236  9/1/89  56.646000
## 238 237 10/1/89  56.673636
## 239 238 11/1/89  57.032381
```

```
## 240 239 12/1/89  63.421000
## 241 240  1/1/90  63.915000
## 242 241  2/1/90  61.758421
## 243 242  3/1/90  63.217727
## 244 243  4/1/90  65.005500
## 245 244  5/1/90  67.288182
## 246 245  6/1/90  69.564286
## 247 246  7/1/90  72.467619
## 248 247  8/1/90  65.168261
## 249 248  9/1/90  58.233158
## 250 249 10/1/90  53.230435
## 251 250 11/1/90  53.578095
## 252 251 12/1/90  56.614000
## 253 252  1/1/91  56.804091
## 254 253  2/1/91  67.175789
## 255 254  3/1/91  68.055000
## 256 255  4/1/91  72.429091
## 257 256  5/1/91  72.508636
## 258 257  6/1/91  74.880000
## 259 258  7/1/91  73.673182
## 260 259  8/1/91  73.138636
## 261 260  9/1/91  70.304000
## 262 261 10/1/91  69.498261
## 263 262 11/1/91  67.980000
## 264 263 12/1/91  68.039524
## 265 264  1/1/92  76.941364
## 266 265  2/1/92  78.004211
## 267 266  3/1/92  77.765000
## 268 267  4/1/92  76.099524
## 269 268  5/1/92  77.246500
## 270 269  6/1/92  76.611818
## 271 270  7/1/92  76.860909
## 272 271  8/1/92  75.320000
## 273 272  9/1/92  76.610476
## 274 273 10/1/92  76.105909
## 275 274 11/1/92  79.203000
## 276 275 12/1/92  84.458182
## 277 276  1/1/93  84.985000
## 278 277  2/1/93  84.873158
## 279 278  3/1/93  87.442609
## 280 279  4/1/93  92.163810
## 281 280  5/1/93  92.511500
## 282 281  6/1/93  94.355455
## 283 282  7/1/93  97.616667
## 284 283  8/1/93  98.453636
## 285 284  9/1/93  96.861429
## 286 285 10/1/93  96.682381
## 287 286 11/1/93  96.235238
## 288 287 12/1/93 102.849546
## 289 288  1/1/94 106.242381
## 290 289  2/1/94 107.564211
## 291 290  3/1/94 104.111304
## 292 291  4/1/94  97.267895
## 293 292  5/1/94  70.630476
```

```
## 294 293  6/1/94  47.894091
## 295 294  7/1/94  48.034000
## 296 295  8/1/94  48.991739
## 297 296  9/1/94  49.662857
## 298 297 10/1/94  48.543810
## 299 298 11/1/94  48.133333
## 300 299 12/1/94  48.574762
## 301 300  1/1/95  51.061905
## 302 301  2/1/95  52.990000
## 303 302  3/1/95  54.121739
## 304 303  4/1/95  54.989474
## 305 304  5/1/95  57.263182
## 306 305  6/1/95  57.155455
## 307 306  7/1/95  58.841000
## 308 307  8/1/95  58.203043
## 309 308  9/1/95  61.283500
## 310 309 10/1/95  63.227727
## 311 310 11/1/95  65.296667
## 312 311 12/1/95  70.892000
## 313 312  1/1/96  72.839545
## 314 313  2/1/96  78.179500
## 315 314  3/1/96  77.069524
## 316 315  4/1/96  78.409048
## 317 316  5/1/96  79.975909
## 318 317  6/1/96  85.785000
## 319 318  7/1/96  83.150909
## 320 319  8/1/96  85.031818
## 321 320  9/1/96  87.210500
## 322 321 10/1/96  94.356087
## 323 322 11/1/96 101.654500
## 324 323 12/1/96  99.914762
## 325 324  1/1/97 102.480454
## 326 325  2/1/97 105.517368
## 327 326  3/1/97 103.641500
## 328 327  4/1/97 102.901818
## 329 328  5/1/97  78.680476
## 330 329  6/1/97  64.421429
## 331 330  7/1/97  70.737273
## 332 331  8/1/97  66.370476
## 333 332  9/1/97  67.629524
## 334 333 10/1/97  68.541739
## 335 334 11/1/97  69.564737
## 336 335 12/1/97  73.552727
## 337 336  1/1/98  74.528000
## 338 337  2/1/98  77.511579
## 339 338  3/1/98  79.762727
## 340 339  4/1/98  85.843810
## 341 340  5/1/98  84.058500
## 342 341  6/1/98  85.916818
## 343 342  7/1/98  92.828636
## 344 343  8/1/98  88.595238
## 345 344  9/1/98  80.260952
## 346 345 10/1/98  79.993182
## 347 346 11/1/98  90.198500
```

```
## 348 347 12/1/98  93.949545
## 349 348  1/1/99 100.580000
## 350 349  2/1/99 100.295789
## 351 350  3/1/99 106.790435
## 352 351  4/1/99 112.223810
## 353 352  5/1/99 106.521500
## 354 353  6/1/99 104.507727
## 355 354  7/1/99 115.362381
## 356 355  8/1/99 110.673636
## 357 356  9/1/99 117.984286
## 358 357 10/1/99 122.195238
## 359 358 11/1/99 135.242857
## 360 359 12/1/99 147.885000
## 361 360  1/1/00 146.463000
## 362 361  2/1/00 133.807000
## 363 362  3/1/00 141.702609
## 364 363  4/1/00 156.843684
## 365 364  5/1/00  75.753636
## 366 365  6/1/00  50.721818
## 367 366  7/1/00  52.673000
## 368 367  8/1/00  55.923043
## 369 368  9/1/00  58.201500
## 370 369 10/1/00  55.746364
## 371 370 11/1/00  52.042381
## 372 371 12/1/00  51.045000
## 373 372  1/1/01  46.370476
## 374 373  2/1/01  47.057895
## 375 374  3/1/01  42.325909
## 376 375  4/1/01  44.876000
## 377 376  5/1/01  50.261364
## 378 377  6/1/01  49.021429
## 379 378  7/1/01  46.246190
## 380 379  8/1/01  41.739130
## 381 380  9/1/01  36.296667
## 382 381 10/1/01  37.644348
## 383 382 11/1/01  39.998571
## 384 383 12/1/01  38.926500
## 385 384  1/1/02  38.751905
## 386 385  2/1/02  37.467895
## 387 386  3/1/02  39.582500
## 388 387  4/1/02  34.595000
## 389 388  5/1/02  31.809091
## 390 389  6/1/02  29.767000
## 391 390  7/1/02  27.989545
## 392 391  8/1/02  31.417273
## 393 392  9/1/02  27.533000
## 394 393 10/1/02  25.306957
## 395 394 11/1/02  25.388000
## 396 395 12/1/02  25.846667
## 397 396  1/1/03  24.647619
## 398 397  2/1/03  23.065789
## 399 398  3/1/03  25.172381
## 400 399  4/1/03  28.060000
## 401 400  5/1/03  28.437143
```

```
## 402 401  6/1/03  30.026190
## 403 402  7/1/03  28.182727
## 404 403  8/1/03  28.915714
## 405 404  9/1/03  31.065238
## 406 405 10/1/03  29.406957
## 407 406 11/1/03  28.594211
## 408 407 12/1/03  30.180455
## 409 408  1/1/04  32.674000
## 410 409  2/1/04  33.105789
## 411 410  3/1/04  31.033913
## 412 411  4/1/04  30.927143
## 413 412  5/1/04  30.382000
## 414 413  6/1/04  31.960000
## 415 414  7/1/04  32.671905
## 416 415  8/1/04  32.375000
## 417 416  9/1/04  33.522381
## 418 417 10/1/04  33.705714
## 419 418 11/1/04  35.492381
## 420 419 12/1/04  36.445455
## 421 420  1/1/05  35.799000
## 422 421  2/1/05  35.943684
## 423 422  3/1/05  35.786364
## 424 423  4/1/05  35.864762
## 425 424  5/1/05  36.384286
## 426 425  6/1/05  36.143636
## 427 426  7/1/05  34.978500
## 428 427  8/1/05  33.940870
## 429 428  9/1/05  33.790476
## 430 429 10/1/05  33.831905
## 431 430 11/1/05  34.830476
## 432 431 12/1/05  35.592381
## 433 432  1/1/06  34.445000
## 434 433  2/1/06  33.096842
## 435 434  3/1/06  33.745652
## 436 435  4/1/06  34.167368
## 437 436  5/1/06  34.452727
## 438 437  6/1/06  33.825000
## 439 438  7/1/06  32.851000
## 440 439  8/1/06  33.290000
## 441 440  9/1/06  34.566000
## 442 441 10/1/06  35.680909
## 443 442 11/1/06  35.466190
## 444 443 12/1/06  36.492500
## 445 444  1/1/07  37.206000
## 446 445  2/1/07  35.847895
## 447 446  3/1/07  34.908182
## 448 447  4/1/07  35.331500
## 449 448  5/1/07  37.162727
## 450 449  6/1/07  37.899524
## 451 450  7/1/07  39.398095
## 452 451  8/1/07  38.567391
## 453 452  9/1/07  40.217368
## 454 453 10/1/07  41.033913
## 455 454 11/1/07  38.768095
```

```
## 456 455 12/1/07  37.150500
## 457 456  1/1/08  35.183810
## 458 457  2/1/08  34.375500
## 459 458  3/1/08  34.801500
## 460 459  4/1/08  34.349545
## 461 460  5/1/08  32.074762
## 462 461  6/1/08  29.005238
## 463 462  7/1/08  27.884545
## 464 463  8/1/08  28.924762
## 465 464  9/1/08  26.360476
## 466 465 10/1/08  20.361304
## 467 466 11/1/08  17.074737
## 468 467 12/1/08  16.884545
## 469 468  1/1/09  14.472000
## 470 469  2/1/09  10.790000
## 471 470  3/1/09   9.293636
## 472 471  4/1/09  11.556667
## 473 472  5/1/09  13.473500
## 474 473  6/1/09  12.832727
## 475 474  7/1/09  11.761818
## 476 475  8/1/09  14.023333
## 477 476  9/1/09  15.591905
## 478 477 10/1/09  15.797727
## 479 478 11/1/09  15.508000
## 480 479 12/1/09  15.754545
```

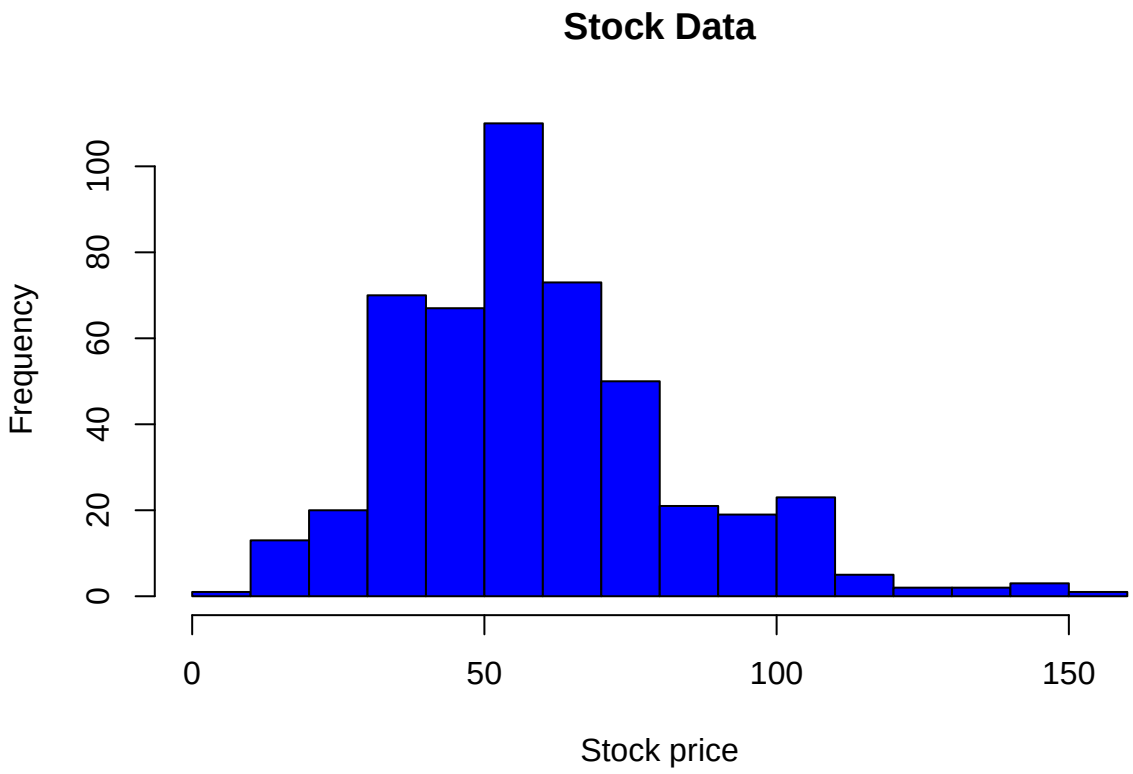
```
names(df)
```

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

```
dim(df)
```

```
## [1] 480   3
```

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



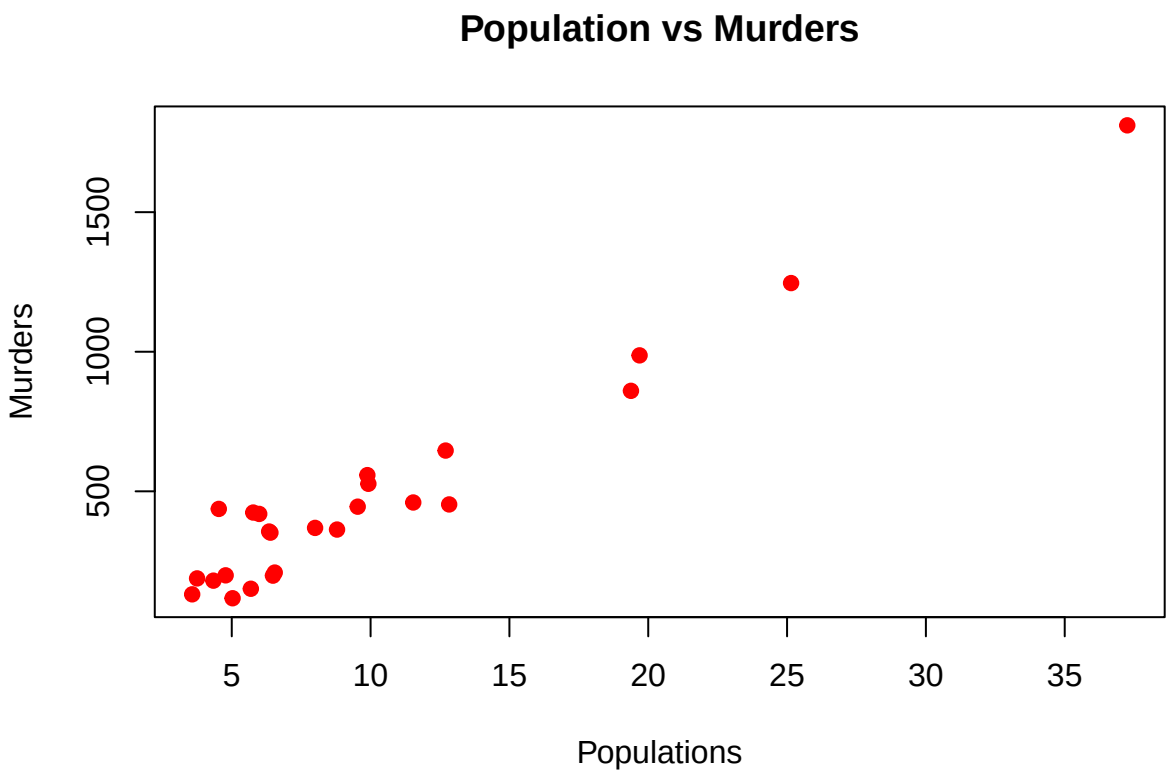
8.3 Scatter Plots

The default of `plot` functions is Scatter plot.

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

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

```
plot(df$population/10^6, df$murders,
     xlab = "Populations",
     ylab = 'Murders',
     main = 'Population vs Murders',
     col = 'red',
     pch = 19
     )
```



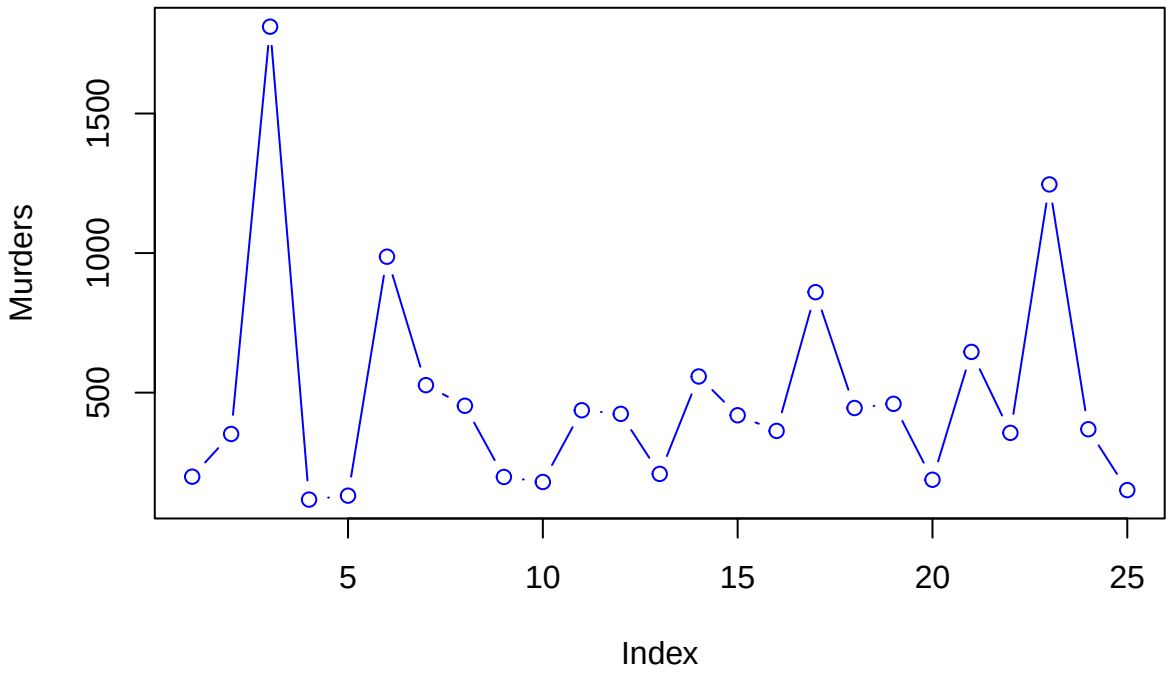
pch values

Values of pch are stored internally as integers.

Figure 1: Some values of pch

8.4 Line Graphs

```
plot(df$murders,
     type = 'b',
     ylab = 'Murders',
     col = 'blue')
```

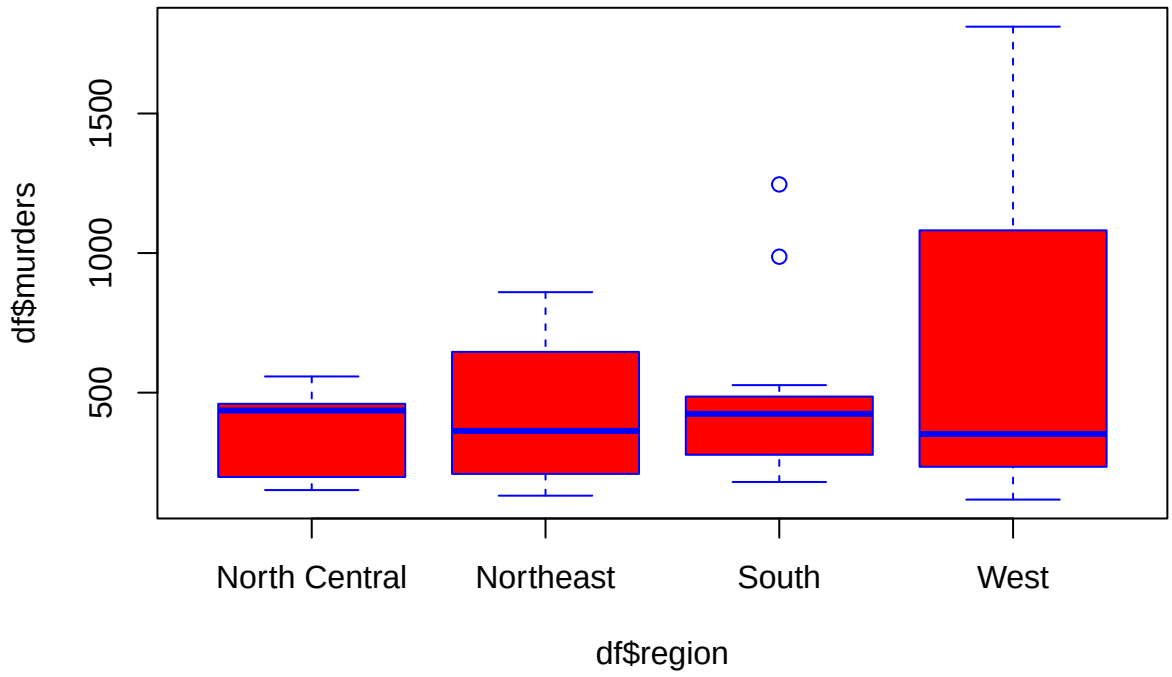


type in plot function

- **"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.

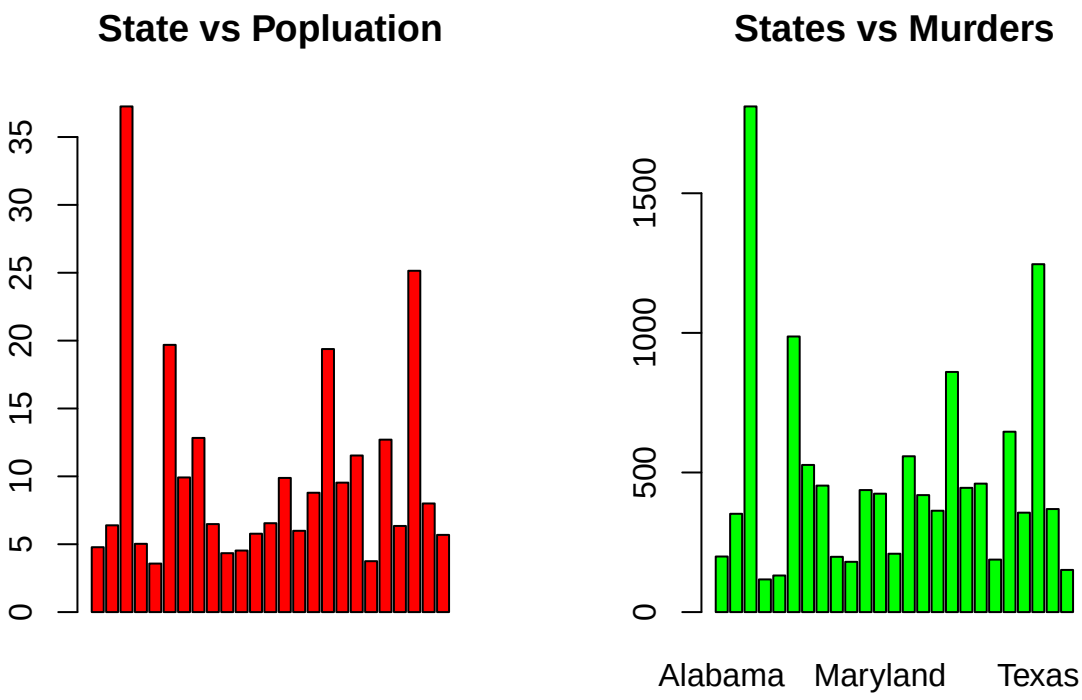
8.5 Box plots

```
boxplot(df$murders ~ df$region,
        col='red',border = 'blue'
        )
```



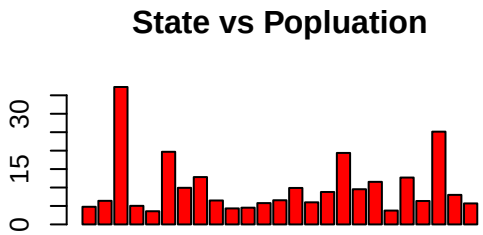
8.6 Multiple Plots in Layout

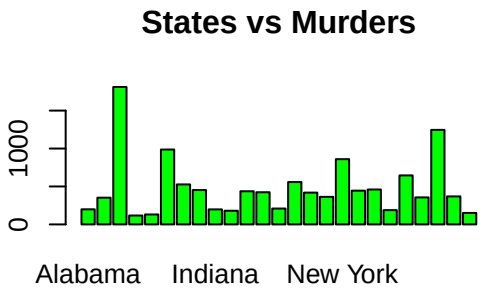
```
par(mfrow = c(1, 2))

barplot(df$population / 10^6,
        main = "State vs Popluation",
        col = 'red')

barplot(df$murders,
        main = 'States vs Murders',
        names.arg = df$state,
        col = 'green')
```

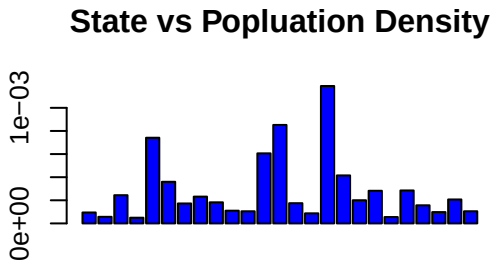


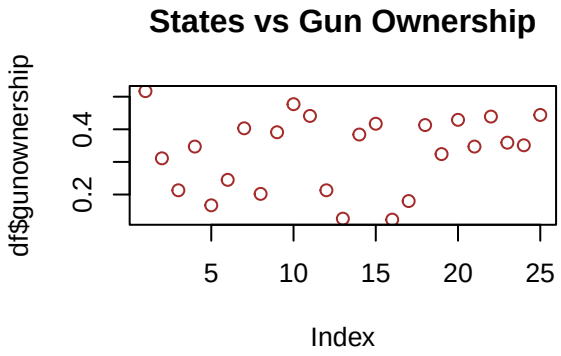

```
par(mfrow = c(2, 2))

barplot(df$population / 10^6,
        main = "State vs Popluation",
        col = 'red')

barplot(df$murders,
        main = 'States vs Murders',
        names.arg = df$state,
        col = 'green')

barplot(df$PopulationDensity / 10^6,
        main = "State vs Popluation Density",
        col = 'blue')

plot(df$gunownership,
        main = 'States vs Gun Ownership',
        names.arg = df$state,
        col = 'brown')
```

```
## Warning in plot.window(...): "names.arg" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "names.arg" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "names.arg" is not
## a graphical parameter
## Warning in axis(side = side, at = at, labels = labels, ...): "names.arg" is not
## a graphical parameter

## Warning in box(...): "names.arg" is not a graphical parameter

## Warning in title(...): "names.arg" is not a graphical parameter
```


State vs Popluation
0
15
30


States vs Murders
0
1000
Alabama
Indiana
New York


State vs Popluation Density
0e+00
1e−03


States vs Gun Ownership
df$gunownership
0.2
0.4
5
10
15
20
25
Index

9 Regressions and Models

9.1 Simple Linear Regression

Linear regression is used to model the relationship between a dependent variable and one or more independent variables by fitting a line through the data.

```
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
```

```
model <- lm(murders ~ population, data = df)
```

```
summary(model)
```

```
##
## Call:
## lm(formula = murders ~ population, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -159.382  -67.999   -8.542   49.987  224.582
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6.110e+00  3.151e+01  -0.194    0.848
## population   4.820e-05  2.475e-06  19.476 8.62e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 95.03 on 23 degrees of freedom
## Multiple R-squared:  0.9428, Adjusted R-squared:  0.9403
## F-statistic: 379.3 on 1 and 23 DF,  p-value: 8.623e-16
```

```
names(df)
```

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

9.2 Multiple linear regression

Multiple regression extends linear regression by adding more independent variables to better predict the dependent variable.

```
multi_model <- lm(murders ~ population + gunownership, data = df)
```

```
summary(multi_model)
```

```
##
## Call:
## lm(formula = murders ~ population + gunownership, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -139.017  -51.392   -2.738   37.700  204.717
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -9.828e+01  7.408e+01  -1.327    0.198
## population    4.942e-05  2.586e-06  19.110 3.44e-15 ***
```

```
## gunownership  2.416e+02  1.764e+02   1.369    0.185
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 93.27 on 22 degrees of freedom
## Multiple R-squared:  0.9473, Adjusted R-squared:  0.9425
## F-statistic: 197.8 on 2 and 22 DF,  p-value: 8.674e-15
```

9.2.0.1 Practice: Add region (as a categorical variable) to the multiple regression model.

```
# Multiple regression model

multi_model_region <- lm(murders ~ population + gunownership + as.factor(region), data = df)

summary(multi_model_region)
```

```
##
## Call:
## lm(formula = murders ~ population + gunownership + as.factor(region),
##     data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -105.79  -57.93  -15.43   29.63  176.13
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                -8.751e+01  1.065e+02  -0.821   0.4216
## population                  4.891e-05  2.727e-06  17.938 2.28e-13 ***
## gunownership                9.279e+01  2.443e+02   0.380   0.7082
## as.factor(region)Northeast  1.298e+01  6.897e+01   0.188   0.8528
## as.factor(region)South      8.573e+01  4.740e+01   1.809   0.0864 .
## as.factor(region)West       2.699e+01  6.842e+01   0.394   0.6977
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 92.12 on 19 degrees of freedom
## Multiple R-squared:  0.9556, Adjusted R-squared:  0.9439
## F-statistic: 81.83 on 5 and 19 DF,  p-value: 3.503e-12
```

9.2.1 t-test (Comparing Means)

A t-test compares the means of two groups to determine if they are statistically different.

```
# Perform a t-test comparing murders between two regions
region1 <- df[df$region == "Northeast", "murders"]
region2 <- df[df$region == "South", "murders"]
```

```
t_test_result <- t.test(region1, region2)
print(t_test_result)
```

```
##
##  Welch Two Sample t-test
##
## data:  region1 and region2
## t = -0.26603, df = 8.5857, p-value = 0.7965
```

```
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -433.2737  342.6918
## sample estimates:
## mean of x mean of y
##  441.8000  487.0909
```

9.2.2 Correlation Test

A correlation test checks the strength and direction of the relationship between two variables.

```
# Correlation test between population and murders
cor_test_result <- cor.test(df$population, df$murders)
print(cor_test_result)
```

```
##
##  Pearson's product-moment correlation
##
## data:  df$population and df$murders
## t = 19.476, df = 23, p-value = 8.623e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9343406 0.9873198
## sample estimates:
##       cor
## 0.9709935
```

```
# Correlation test gunownership vs murders
cor.test(df$gunownership, df$murders)
```

9.2.2.1 Practice

```
##
##  Pearson's product-moment correlation
##
## data:  df$gunownership and df$murders
## t = -1.3441, df = 23, p-value = 0.192
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.6009155  0.1402226
## sample estimates:
##        cor
## -0.2698602
```

```
model <- lm(murders ~ population, data = df)

summary(model)
```

```
##
## Call:
## lm(formula = murders ~ population, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -159.382  -67.999   -8.542   49.987  224.582
```

```
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6.110e+00  3.151e+01  -0.194    0.848
## population   4.820e-05  2.475e-06  19.476 8.62e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 95.03 on 23 degrees of freedom
## Multiple R-squared:  0.9428, Adjusted R-squared:  0.9403
## F-statistic: 379.3 on 1 and 23 DF,  p-value: 8.623e-16
```

9.2.3 Regression Plots

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

abline(model, col = 'red', lwd=2)
```