## Day 1 – R Programming

```
> #install.packages('caret')
> num = 10
> num
[1] 10
> library('caret')
> x = 10.2
> y <- 10
> z = "Hello"
> x
[1] 10.2
> y
[1] 10
> z
[1] "Hello"
> as.integer(x)
[1] 10
> a = 1 + 10i
> a
[1] 1+10i
> sqrt(144)
[1] 12
> a = 5; b = 15
> out = a > b
> out
[1] FALSE
> age <- c(21, 25, 28, 30, 20, 26)
> age
[1] 21 25 28 30 20 26
> id = c(1:10)  #range values from 1-10
> id
 [1]  1  2  3  4  5  6  7  8  9 10
> seq(1, 20)
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
> seq(2, 20, 2)   #range values from 2 to 20 with offset 2
 [1]  2  4  6  8 10 12 14 16 18 20
> loan_default <- c(TRUE, FALSE, FALSE, TRUE, TRUE)
> loan_default
[1]  TRUE FALSE FALSE  TRUE  TRUE
> place_names <- c("China", "India", "Denmark", "UK", "Finland")
> place_names
[1] "China"   "India"   "Denmark" "UK"      "Finland"
> class(loan_default)
[1] "logical"
```

```
> class(age)
[1] "numeric"
> class(z)
[1] "character"
> num_as_str <- c("10", "30", "40", "50")
> class(num_as_str)
[1] "character"
> numbers <- as.integer(num_as_str)
> class(numbers)
[1] "integer"
> mean(numbers)
[1] 32.5
> max(age)
[1] 30
> min(numbers)
[1] 10
> median(age)
[1] 25.5
> range(numbers)
[1] 10 50
> var(age)
[1] 15.2
> sort(age)
[1] 20 21 25 26 28 30
> sort(age, decreasing = TRUE)
[1] 30 28 26 25 21 20
> random_ele <- c(15, 2.5, TRUE, "Hello")
> random_ele
[1] "15"   "2.5"  "TRUE" "Hello"
> class(random_ele)
[1] "character"
> mat <- c(1:16)
> mat <- matrix(mat, ncol=4)
> mat
     [,1] [,2] [,3] [,4]
[1,]    1    5    9   13
[2,]    2    6   10   14
[3,]    3    7   11   15
[4,]    4    8   12   16
> mat1 <- c(1:16)
> mat1 <- matrix(mat1, ncol = 4, byrow = T)
> mat1
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
```

```
[2,]   5   6   7   8
[3,]   9  10  11  12
[4,]  13  14  15  16
> matrix(c(56, 72, 25, 14, 87, 99), ncol = 3, byrow = T)
     [,1] [,2] [,3]
[1,]  56  72  25
[2,]  14  87  99
> mat1[2,]
[1] 5 6 7 8
> mat1[2,2]
[1] 6
> mat1[,4]
[1]  4  8 12 16
> matr = matrix(c(5:16), nrow = 3, byrow = TRUE)
> column.names <- c("COL1", "COL2", "COL3")
> row.names <- c("ROW1", "ROW2", "ROW3")
> column.names <- c("COL1", "COL2", "COL3", "COL4")
> result <- matrix(c(5:16), nrow = 3, byrow = TRUE, dimnames = list(row.names, column.names))
> result
     COL1 COL2 COL3 COL4
ROW1   5   6   7   8
ROW2   9  10  11  12
ROW3  13  14  15  16
> employee = list(1, c("John", "Rose"), c(12000, 15000))
> employee
[[1]]
[1] 1

[[2]]
[1] "John" "Rose"

[[3]]
[1] 12000 15000

> employee[[1]]
[1] 1
> employee[[2]]
[1] "John" "Rose"
> employee[[3]]
[1] 12000 15000
> employee = list(EmpID=1, EmpName=c("John", "Rose"), basic_pay=c(12000, 15000))
> employee
$EmpID
[1] 1
```

```
$EmpName
[1] "John" "Rose"

$basic_pay
[1] 12000 15000

> employee$EmpName
[1] "John" "Rose"
> list_of_expenses <- list(100, 150, 350, 50)
> class((list_of_expenses))
[1] "list"
> expenses <- unlist(list_of_expenses)
> class(expenses)
[1] "numeric"
> length(expenses)
[1] 4
> days_from_purchase <- c(10, 15, 20, 25)
> days_from_purchase
[1] 10 15 20 25
> ctf <- as.factor(days_from_purchase)
> typeof(ctf)
[1] "integer"
> class(ctf)
[1] "factor"
> age <- c(21, 42, 28, 31, 19)
> names <- c("John", "Sachin", "Rahul", "Ravi", "Sameer")
> salary <- c(12000, 20000, 25000, 16000, 28000)
> ownhouse <- c(TRUE, FALSE, TRUE, TRUE, FALSE)
> mydf <- data.frame(names, age, salary, ownhouse)
> mydf
   names age salary ownhouse
1   John  21  12000     TRUE
2 Sachin  42  20000    FALSE
3  Rahul  28  25000     TRUE
4   Ravi  31  16000     TRUE
5 Sameer  19  28000    FALSE
> stock_price <- c(110.55, 102.50, 145.90, 130.70, 160.45, 112.80)
> stock_mat <- matrix(stock_price, ncol = 2, byrow = T)
> stock_df = data.frame(stock_mat)
> stock_df
     X1    X2
1 110.55 102.5
2 145.90 130.7
```

3 160.45 112.8
> colnames(stock_df) <- c("Open Price", "Close Price")
> letters[1:10]
 [1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j"
> letters[1:26]
 [1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p" "q" "r" "s" "t" "u" "v" "w" "x" "y" "z"
> rownames(stock_df) <- letters[1:3]
> stock_df
  Open Price Close Price
a    110.55      102.5
b    145.90      130.7
c    160.45      112.8
> stock_df$`Close Price`
[1] 102.5 130.7 112.8

## Day 2 – R Programming
> X <- matrix(c(50, 70, 40, 90, 60, 80, 50, 90, 100, 50, 30, 70), nrow = 3)
> X
   [,1] [,2] [,3] [,4]
[1,]  50  90  50  50
[2,]  70  60  90  30
[3,]  40  80  100  70
> rowSums(X)
[1] 240 250 290
> colSums(X)
[1] 160 230 240 150
> X <- rbind(X, apply(X, 2, mean))  #Add a row and apply mean function columnswise - 2, for rowwise its 1
> X
      [,1]    [,2] [,3] [,4]
[1,] 50.00000 90.00000  50  50
[2,] 70.00000 60.00000  90  30
[3,] 40.00000 80.00000  100  70
[4,] 53.33333 76.66667  80  50
> X <- cbind(X, apply(X, 1, var))   #Add a column and apply variance function rowwise - 1
> X
      [,1]    [,2] [,3] [,4]    [,5]
[1,] 50.00000 90.00000  50  50 400.0000
[2,] 70.00000 60.00000  90  30 625.0000
[3,] 40.00000 80.00000  100  70 625.0000
[4,] 53.33333 76.66667  80  50 240.7407
> X <- matrix(c(50, 70, 40, 90, 60, 80, 50, 90, 100, 50, 30, 70), nrow = 3)
> X <- cbind(X, apply(X, 1, sd))    #Add a column and apply standard deviation function rowwise - 1
> X

```
     [,1] [,2] [,3] [,4] [,5]
[1,]  50  90  50  50  20
[2,]  70  60  90  30  25
[3,]  40  80 100  70  25
> X <- rbind(X, apply(X, 2, max))   #Add a row and apply maximum function columnswise - 2, for rowwise its 1
> X
     [,1] [,2] [,3] [,4] [,5]
[1,]  50  90  50  50  20
[2,]  70  60  90  30  25
[3,]  40  80 100  70  25
[4,]  70  90 100  70  25
> stock_df[[1]]   #1st column
[1] 110.55 145.90 160.45
> stock_df[[2]]   #2nd column
[1] 102.5 130.7 112.8
> stock_df
  Open Price Close Price BuyOrSell
a     110.55       102.5      Sell
b     145.90       130.7      Sell
c     160.45       112.8      Sell
> stock_df[1:2, 2]
[1] 102.5 130.7
> stock_df[1:3, 1:2]
  Open Price Close Price
a     110.55       102.5
b     145.90       130.7
c     160.45       112.8
> stock_df[, 1:2]
  Open Price Close Price
a     110.55       102.5
b     145.90       130.7
c     160.45       112.8
> stock_df[c(1, 3), 1:2]
  Open Price Close Price
a     110.55       102.5
c     160.45       112.8
> stock_df[-1, 1]
[1] 145.90 160.45
> stock_df[-c(1, 3), 1:2]
  Open Price Close Price
b      145.9       130.7
> v_sub <- stock_df[1:3, 2]
> v_sub
```

```
[1] 102.5 130.7 112.8
> df_subsetdata <- stock_df[1:3, 2, drop=F]
> df_subsetdata
  Close Price
a      102.5
b      130.7
c      112.8
> class(v_sub)
[1] "numeric"
> class(df_subsetdata)
[1] "data.frame"
> setwd("C:/zubeda/PGA02_Zubu/R Programming")  #Set current working directory
> housing_df <- read.csv("Housing.csv")
> housing_df
     price  area bedrooms bathrooms stories mainroad guestroom basement
1  13300000 7420     4         2        3      yes       no       no
2  12250000 8960     4         4        4      yes       no       no
3  12250000 9960     3         2        2      yes       no      yes
4  12215000 7500     4         2        2      yes       no      yes
5  11410000 7420     4         1        2      yes      yes      yes
6  10850000 7500     3         3        1      yes       no      yes
7  10150000 8580     4         3        4      yes       no       no
8  10150000 16200    5         3        2      yes       no       no
9   9870000 8100     4         1        2      yes      yes      yes
10  9800000 5750     3         2        4      yes      yes       no
11  9800000 13200    3         1        2      yes       no      yes
12  9681000 6000     4         3        2      yes      yes      yes
13  9310000 6550     4         2        2      yes       no       no
14  9240000 3500     4         2        2      yes       no       no
15  9240000 7800     3         2        2      yes       no       no
16  9100000 6000     4         1        2      yes       no      yes
17  9100000 6600     4         2        2      yes      yes      yes
18  8960000 8500     3         2        4      yes       no       no
19  8890000 4600     3         2        2      yes      yes       no
20  8855000 6420     3         2        2      yes       no       no
21  8750000 4320     3         1        2      yes       no      yes
22  8680000 7155     3         2        1      yes      yes      yes
23  8645000 8050     3         1        1      yes      yes      yes
24  8645000 4560     3         2        2      yes      yes      yes
25  8575000 8800     3         2        2      yes       no       no
26  8540000 6540     4         2        2      yes      yes      yes
27  8463000 6000     3         2        4      yes      yes      yes
28  8400000 8875     3         1        1      yes       no       no
29  8400000 7950     5         2        2      yes       no      yes
```

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 30 | 8400000 | 5500 | 4 | 2 | 2 | yes | no | yes |
| 31 | 8400000 | 7475 | 3 | 2 | 4 | yes | no | no |
| 32 | 8400000 | 7000 | 3 | 1 | 4 | yes | no | no |
| 33 | 8295000 | 4880 | 4 | 2 | 2 | yes | no | no |
| 34 | 8190000 | 5960 | 3 | 3 | 2 | yes | yes | yes |
| 35 | 8120000 | 6840 | 5 | 1 | 2 | yes | yes | yes |
| 36 | 8080940 | 7000 | 3 | 2 | 4 | yes | no | no |
| 37 | 8043000 | 7482 | 3 | 2 | 3 | yes | no | no |
| 38 | 7980000 | 9000 | 4 | 2 | 4 | yes | no | no |
| 39 | 7962500 | 6000 | 3 | 1 | 4 | yes | yes | no |
| 40 | 7910000 | 6000 | 4 | 2 | 4 | yes | no | no |
| 41 | 7875000 | 6550 | 3 | 1 | 2 | yes | no | yes |
| 42 | 7840000 | 6360 | 3 | 2 | 4 | yes | no | no |
| 43 | 7700000 | 6480 | 3 | 2 | 4 | yes | no | no |
| 44 | 7700000 | 6000 | 4 | 2 | 4 | yes | no | no |
| 45 | 7560000 | 6000 | 4 | 2 | 4 | yes | no | no |
| 46 | 7560000 | 6000 | 3 | 2 | 3 | yes | no | no |
| 47 | 7525000 | 6000 | 3 | 2 | 4 | yes | no | no |
| 48 | 7490000 | 6600 | 3 | 1 | 4 | yes | no | no |
| 49 | 7455000 | 4300 | 3 | 2 | 2 | yes | no | yes |
| 50 | 7420000 | 7440 | 3 | 2 | 1 | yes | yes | yes |
| 51 | 7420000 | 7440 | 3 | 2 | 4 | yes | no | no |
| 52 | 7420000 | 6325 | 3 | 1 | 4 | yes | no | no |
| 53 | 7350000 | 6000 | 4 | 2 | 4 | yes | yes | no |
| 54 | 7350000 | 5150 | 3 | 2 | 4 | yes | no | no |
| 55 | 7350000 | 6000 | 3 | 2 | 2 | yes | yes | no |
| 56 | 7350000 | 6000 | 3 | 1 | 2 | yes | no | no |
| 57 | 7343000 | 11440 | 4 | 1 | 2 | yes | no | yes |
| 58 | 7245000 | 9000 | 4 | 2 | 4 | yes | yes | no |
| 59 | 7210000 | 7680 | 4 | 2 | 4 | yes | yes | no |
| 60 | 7210000 | 6000 | 3 | 2 | 4 | yes | yes | no |
| 61 | 7140000 | 6000 | 3 | 2 | 2 | yes | yes | no |
| 62 | 7070000 | 8880 | 2 | 1 | 1 | yes | no | no |
| 63 | 7070000 | 6240 | 4 | 2 | 2 | yes | no | no |
| 64 | 7035000 | 6360 | 4 | 2 | 3 | yes | no | no |
| 65 | 7000000 | 11175 | 3 | 1 | 1 | yes | no | yes |
| 66 | 6930000 | 8880 | 3 | 2 | 2 | yes | no | yes |
| 67 | 6930000 | 13200 | 2 | 1 | 1 | yes | no | yes |
| 68 | 6895000 | 7700 | 3 | 2 | 1 | yes | no | no |
| 69 | 6860000 | 6000 | 3 | 1 | 1 | yes | no | no |
| 70 | 6790000 | 12090 | 4 | 2 | 2 | yes | no | no |
| 71 | 6790000 | 4000 | 3 | 2 | 2 | yes | no | yes |
| 72 | 6755000 | 6000 | 4 | 2 | 4 | yes | no | no |
| 73 | 6720000 | 5020 | 3 | 1 | 4 | yes | no | no |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 74 | 6685000 | 6600 | 2 | 2 | 4 | yes | no | yes |
| 75 | 6650000 | 4040 | 3 | 1 | 2 | yes | no | yes |
| 76 | 6650000 | 4260 | 4 | 2 | 2 | yes | no | no |

| | hotwaterheating | airconditioning | parking | prefarea | furnishingstatus |
|---|---|---|---|---|---|
| 1 | no | yes | 2 | yes | furnished |
| 2 | no | yes | 3 | no | furnished |
| 3 | no | no | 2 | yes | semi-furnished |
| 4 | no | yes | 3 | yes | furnished |
| 5 | no | yes | 2 | no | furnished |
| 6 | no | yes | 2 | yes | semi-furnished |
| 7 | no | yes | 2 | yes | semi-furnished |
| 8 | no | no | 0 | no | unfurnished |
| 9 | no | yes | 2 | yes | furnished |
| 10 | no | yes | 1 | yes | unfurnished |
| 11 | no | yes | 2 | yes | furnished |
| 12 | yes | no | 2 | no | semi-furnished |
| 13 | no | yes | 1 | yes | semi-furnished |
| 14 | yes | no | 2 | no | furnished |
| 15 | no | no | 0 | yes | semi-furnished |
| 16 | no | no | 2 | no | semi-furnished |
| 17 | no | yes | 1 | yes | unfurnished |
| 18 | no | yes | 2 | no | furnished |
| 19 | no | yes | 2 | no | furnished |
| 20 | no | yes | 1 | yes | semi-furnished |
| 21 | yes | no | 2 | no | semi-furnished |
| 22 | no | yes | 2 | no | unfurnished |
| 23 | no | yes | 1 | no | furnished |
| 24 | no | yes | 1 | no | furnished |
| 25 | no | yes | 2 | no | furnished |
| 26 | no | yes | 2 | yes | furnished |
| 27 | no | yes | 0 | yes | semi-furnished |
| 28 | no | no | 1 | no | semi-furnished |
| 29 | yes | no | 2 | no | unfurnished |
| 30 | no | yes | 1 | yes | semi-furnished |
| 31 | no | yes | 2 | no | unfurnished |
| 32 | no | yes | 2 | no | semi-furnished |
| 33 | no | yes | 1 | yes | furnished |
| 34 | no | no | 1 | no | unfurnished |
| 35 | no | yes | 1 | no | furnished |
| 36 | no | yes | 2 | no | furnished |
| 37 | yes | no | 1 | yes | furnished |
| 38 | no | yes | 2 | no | furnished |
| 39 | no | yes | 2 | no | unfurnished |
| 40 | no | yes | 1 | no | semi-furnished |

| 41 | no | yes | 0 | yes | furnished |
| 42 | no | yes | 0 | yes | furnished |
| 43 | no | yes | 2 | no | unfurnished |
| 44 | no | no | 2 | no | semi-furnished |
| 45 | no | yes | 1 | no | furnished |
| 46 | no | yes | 0 | no | semi-furnished |
| 47 | no | yes | 1 | no | furnished |
| 48 | no | yes | 3 | yes | furnished |
| 49 | no | no | 1 | no | unfurnished |
| 50 | no | yes | 0 | yes | semi-furnished |
| 51 | no | no | 1 | yes | unfurnished |
| 52 | no | yes | 1 | no | unfurnished |
| 53 | no | yes | 1 | no | furnished |
| 54 | no | yes | 2 | no | semi-furnished |
| 55 | no | yes | 1 | no | semi-furnished |
| 56 | no | yes | 1 | no | unfurnished |
| 57 | no | no | 1 | yes | semi-furnished |
| 58 | no | yes | 1 | yes | furnished |
| 59 | no | yes | 1 | no | semi-furnished |
| 60 | no | yes | 1 | no | furnished |
| 61 | no | no | 1 | no | semi-furnished |
| 62 | no | yes | 1 | no | semi-furnished |
| 63 | no | yes | 1 | no | furnished |
| 64 | no | yes | 2 | yes | furnished |
| 65 | no | yes | 1 | yes | furnished |
| 66 | no | yes | 1 | no | furnished |
| 67 | yes | no | 1 | no | furnished |
| 68 | no | no | 2 | no | unfurnished |
| 69 | no | yes | 1 | no | furnished |
| 70 | no | no | 2 | yes | furnished |
| 71 | no | yes | 0 | yes | semi-furnished |
| 72 | no | yes | 0 | no | unfurnished |
| 73 | no | yes | 0 | yes | unfurnished |
| 74 | no | no | 0 | yes | furnished |
| 75 | yes | no | 1 | no | furnished |
| 76 | yes | no | 0 | no | semi-furnished |

 [ reached 'max' / getOption("max.print") -- omitted 469 rows ]
> dim(housing_df)  #no. of rows, no. of columns
[1] 545  13
> filter_df <- housing_df[housing_df$price > 10000000, ]
> filter_df
   price  area bedrooms bathrooms stories mainroad guestroom basement
1 13300000  7420     4     2     3    yes     no     no
2 12250000  8960     4     4     4    yes     no     no

```
3 12250000 9960    3    2    2    yes    no    yes
4 12215000 7500    4    2    2    yes    no    yes
5 11410000 7420    4    1    2    yes    yes   yes
6 10850000 7500    3    3    1    yes    no    yes
7 10150000 8580    4    3    4    yes    no    no
8 10150000 16200   5    3    2    yes    no    no
  hotwaterheating airconditioning parking prefarea furnishingstatus
1        no           yes      2     yes      furnished
2        no           yes      3     no       furnished
3        no           no       2     yes   semi-furnished
4        no           yes      3     yes      furnished
5        no           yes      2     no       furnished
6        no           yes      2     yes   semi-furnished
7        no           yes      2     yes   semi-furnished
8        no           no       0     no      unfurnished
> filt_df <- housing_df[housing_df$area > 6000, ]
> filt_df
     price  area bedrooms bathrooms stories mainroad guestroom basement
1  13300000 7420    4    2    3    yes    no    no
2  12250000 8960    4    4    4    yes    no    no
3  12250000 9960    3    2    2    yes    no    yes
4  12215000 7500    4    2    2    yes    no    yes
5  11410000 7420    4    1    2    yes    yes   yes
6  10850000 7500    3    3    1    yes    no    yes
7  10150000 8580    4    3    4    yes    no    no
8  10150000 16200   5    3    2    yes    no    no
9   9870000 8100    4    1    2    yes    yes   yes
11  9800000 13200   3    1    2    yes    no    yes
13  9310000 6550    4    2    2    yes    no    no
15  9240000 7800    3    2    2    yes    no    no
17  9100000 6600    4    2    2    yes    yes   yes
18  8960000 8500    3    2    4    yes    no    no
20  8855000 6420    3    2    2    yes    no    no
22  8680000 7155    3    2    1    yes    yes   yes
23  8645000 8050    3    1    1    yes    yes   yes
25  8575000 8800    3    2    2    yes    no    no
26  8540000 6540    4    2    2    yes    yes   yes
28  8400000 8875    3    1    1    yes    no    no
29  8400000 7950    5    2    2    yes    no    yes
31  8400000 7475    3    2    4    yes    no    no
32  8400000 7000    3    1    4    yes    no    no
35  8120000 6840    5    1    2    yes    yes   yes
36  8080940 7000    3    2    4    yes    no    no
37  8043000 7482    3    2    3    yes    no    no
```

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 38 | 7980000 | 9000 | 4 | 2 | 4 | yes | no | no |
| 41 | 7875000 | 6550 | 3 | 1 | 2 | yes | no | yes |
| 42 | 7840000 | 6360 | 3 | 2 | 4 | yes | no | no |
| 43 | 7700000 | 6480 | 3 | 2 | 4 | yes | no | no |
| 48 | 7490000 | 6600 | 3 | 1 | 4 | yes | no | no |
| 50 | 7420000 | 7440 | 3 | 2 | 1 | yes | yes | yes |
| 51 | 7420000 | 7440 | 3 | 2 | 4 | yes | no | no |
| 52 | 7420000 | 6325 | 3 | 1 | 4 | yes | no | no |
| 57 | 7343000 | 11440 | 4 | 1 | 2 | yes | no | yes |
| 58 | 7245000 | 9000 | 4 | 2 | 4 | yes | yes | no |
| 59 | 7210000 | 7680 | 4 | 2 | 4 | yes | yes | no |
| 62 | 7070000 | 8880 | 2 | 1 | 1 | yes | no | no |
| 63 | 7070000 | 6240 | 4 | 2 | 2 | yes | no | no |
| 64 | 7035000 | 6360 | 4 | 2 | 3 | yes | no | no |
| 65 | 7000000 | 11175 | 3 | 1 | 1 | yes | no | yes |
| 66 | 6930000 | 8880 | 3 | 2 | 2 | yes | no | yes |
| 67 | 6930000 | 13200 | 2 | 1 | 1 | yes | no | yes |
| 68 | 6895000 | 7700 | 3 | 2 | 1 | yes | no | no |
| 70 | 6790000 | 12090 | 4 | 2 | 2 | yes | no | no |
| 74 | 6685000 | 6600 | 2 | 2 | 4 | yes | no | yes |
| 77 | 6650000 | 6420 | 3 | 2 | 3 | yes | no | no |
| 78 | 6650000 | 6500 | 3 | 2 | 3 | yes | no | no |
| 83 | 6615000 | 10500 | 3 | 2 | 1 | yes | no | yes |
| 86 | 6510000 | 8250 | 3 | 2 | 3 | yes | no | no |
| 87 | 6510000 | 6670 | 3 | 1 | 3 | yes | no | yes |
| 89 | 6475000 | 7410 | 3 | 1 | 1 | yes | yes | yes |
| 90 | 6440000 | 8580 | 5 | 3 | 2 | yes | no | no |
| 92 | 6419000 | 6750 | 2 | 1 | 1 | yes | yes | yes |
| 94 | 6300000 | 7200 | 3 | 2 | 1 | yes | no | yes |
| 97 | 6300000 | 9000 | 3 | 1 | 1 | yes | no | yes |
| 98 | 6300000 | 6400 | 3 | 1 | 1 | yes | yes | yes |
| 99 | 6293000 | 6600 | 3 | 2 | 3 | yes | no | no |
| 101 | 6230000 | 6600 | 3 | 2 | 1 | yes | no | yes |
| 104 | 6195000 | 6350 | 3 | 2 | 3 | yes | yes | no |
| 108 | 6125000 | 6420 | 3 | 1 | 3 | yes | no | yes |
| 110 | 6090000 | 6615 | 4 | 2 | 2 | yes | yes | no |
| 111 | 6090000 | 6600 | 3 | 1 | 1 | yes | yes | yes |
| 112 | 6090000 | 8372 | 3 | 1 | 3 | yes | no | no |
| 114 | 6083000 | 9620 | 3 | 1 | 1 | yes | no | yes |
| 115 | 6020000 | 6800 | 2 | 1 | 1 | yes | yes | yes |
| 116 | 6020000 | 8000 | 3 | 1 | 1 | yes | yes | yes |
| 117 | 6020000 | 6900 | 3 | 2 | 1 | yes | yes | yes |
| 119 | 5950000 | 6420 | 3 | 1 | 1 | yes | no | yes |
| 120 | 5950000 | 7020 | 3 | 1 | 1 | yes | no | yes |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 121 | 5950000 | 6540 | 3 | 1 | 1 | yes | yes | yes |
| 122 | 5950000 | 7231 | 3 | 1 | 2 | yes | yes | yes |
| 123 | 5950000 | 6254 | 4 | 2 | 1 | yes | no | yes |
| 124 | 5950000 | 7320 | 4 | 2 | 2 | yes | no | no |
| 125 | 5950000 | 6525 | 3 | 2 | 4 | yes | no | no |
| 126 | 5943000 | 15600 | 3 | 1 | 1 | yes | no | no |

| | hotwaterheating | airconditioning | parking | prefarea | furnishingstatus |
|---|---|---|---|---|---|
| 1 | no | yes | 2 | yes | furnished |
| 2 | no | yes | 3 | no | furnished |
| 3 | no | no | 2 | yes | semi-furnished |
| 4 | no | yes | 3 | yes | furnished |
| 5 | no | yes | 2 | no | furnished |
| 6 | no | yes | 2 | yes | semi-furnished |
| 7 | no | yes | 2 | yes | semi-furnished |
| 8 | no | no | 0 | no | unfurnished |
| 9 | no | yes | 2 | yes | furnished |
| 11 | no | yes | 2 | yes | furnished |
| 13 | no | yes | 1 | yes | semi-furnished |
| 15 | no | no | 0 | yes | semi-furnished |
| 17 | no | yes | 1 | yes | unfurnished |
| 18 | no | yes | 2 | no | furnished |
| 20 | no | yes | 1 | yes | semi-furnished |
| 22 | no | yes | 2 | no | unfurnished |
| 23 | no | yes | 1 | no | furnished |
| 25 | no | yes | 2 | no | furnished |
| 26 | no | yes | 2 | yes | furnished |
| 28 | no | no | 1 | no | semi-furnished |
| 29 | yes | no | 2 | no | unfurnished |
| 31 | no | yes | 2 | no | unfurnished |
| 32 | no | yes | 2 | no | semi-furnished |
| 35 | no | yes | 1 | no | furnished |
| 36 | no | yes | 2 | no | furnished |
| 37 | yes | no | 1 | yes | furnished |
| 38 | no | yes | 2 | no | furnished |
| 41 | no | yes | 0 | yes | furnished |
| 42 | no | yes | 0 | yes | furnished |
| 43 | no | yes | 2 | no | unfurnished |
| 48 | no | yes | 3 | yes | furnished |
| 50 | no | yes | 0 | yes | semi-furnished |
| 51 | no | no | 1 | yes | unfurnished |
| 52 | no | yes | 1 | no | unfurnished |
| 57 | no | no | 1 | yes | semi-furnished |
| 58 | no | yes | 1 | yes | furnished |
| 59 | no | yes | 1 | no | semi-furnished |

| 62  | no  | yes | 1 | no  | semi-furnished |
| 63  | no  | yes | 1 | no  | furnished      |
| 64  | no  | yes | 2 | yes | furnished      |
| 65  | no  | yes | 1 | yes | furnished      |
| 66  | no  | yes | 1 | no  | furnished      |
| 67  | yes | no  | 1 | no  | furnished      |
| 68  | no  | no  | 2 | no  | unfurnished    |
| 70  | no  | no  | 2 | yes | furnished      |
| 74  | no  | no  | 0 | yes | furnished      |
| 77  | no  | yes | 0 | yes | furnished      |
| 78  | no  | yes | 0 | yes | furnished      |
| 83  | no  | yes | 1 | yes | furnished      |
| 86  | no  | yes | 0 | no  | furnished      |
| 87  | no  | no  | 0 | yes | unfurnished    |
| 89  | no  | yes | 2 | yes | unfurnished    |
| 90  | no  | no  | 2 | no  | furnished      |
| 92  | no  | no  | 2 | yes | furnished      |
| 94  | no  | yes | 3 | no  | semi-furnished |
| 97  | no  | no  | 1 | yes | furnished      |
| 98  | no  | yes | 1 | yes | semi-furnished |
| 99  | no  | yes | 0 | yes | unfurnished    |
| 101 | no  | yes | 0 | yes | unfurnished    |
| 104 | no  | yes | 0 | no  | furnished      |
| 108 | no  | no  | 0 | yes | unfurnished    |
| 110 | yes | no  | 1 | no  | semi-furnished |
| 111 | no  | no  | 2 | yes | semi-furnished |
| 112 | no  | yes | 2 | no  | unfurnished    |
| 114 | no  | no  | 2 | yes | furnished      |
| 115 | no  | no  | 2 | no  | furnished      |
| 116 | no  | yes | 2 | yes | semi-furnished |
| 117 | no  | no  | 0 | yes | unfurnished    |
| 119 | no  | yes | 0 | yes | furnished      |
| 120 | no  | yes | 2 | yes | semi-furnished |
| 121 | no  | no  | 2 | yes | furnished      |
| 122 | no  | yes | 0 | yes | semi-furnished |
| 123 | no  | no  | 1 | yes | semi-furnished |
| 124 | no  | no  | 0 | no  | furnished      |
| 125 | no  | no  | 1 | no  | furnished      |
| 126 | no  | yes | 2 | no  | semi-furnished |

```
 [ reached 'max' / getOption("max.print") -- omitted 81 rows ]
> price <- 5
> if(price > 5) {
+   print("Sell the stock")
+ } else {
```

```
+   print("Buy the stock")
+ }
[1] "Buy the stock"
> source("Conditional.R")
[1] "Buy the stock"
> stock_df
  Open Price Close Price BuyOrSell
a    110.55      102.5      Sell
b    145.90      130.7      Sell
c    160.45      112.8      Sell
> stock_df$BuyOrSell <- ifelse(stock_df$`Close Price` < 80, "Buy", "Sell")
> stock_df
  Open Price Close Price BuyOrSell
a    110.55      102.5      Sell
b    145.90      130.7      Sell
c    160.45      112.8      Sell
> for (x in 1:10) { print(x ^ 2) } #i raised to 2
[1] 1
[1] 4
[1] 9
[1] 16
[1] 25
[1] 36
[1] 49
[1] 64
[1] 81
[1] 100
> mtcars   #inbuilt dataset
                   mpg cyl  disp  hp drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive    21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
Valiant           18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
Duster 360        14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
Merc 240D         24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
Merc 230          22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
Merc 280          19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
Merc 280C         17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
Merc 450SE        16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
Merc 450SL        17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
Merc 450SLC       15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
Cadillac Fleetwood 10.4  8 472.0 205 2.93 5.250 17.98  0  0    3    4
```

```
Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0 0   3   4
Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0 0   3   4
Fiat 128        32.4   4 78.7 66 4.08 2.200 19.47  1 1   4   1
Honda Civic      30.4   4 75.7 52 4.93 1.615 18.52  1 1   4   2
Toyota Corolla    33.9   4 71.1 65 4.22 1.835 19.90  1 1   4   1
Toyota Corona     21.5   4 120.1  97 3.70 2.465 20.01  1 0   3   1
Dodge Challenger   15.5   8 318.0 150 2.76 3.520 16.87  0 0   3   2
AMC Javelin      15.2   8 304.0 150 3.15 3.435 17.30  0 0   3   2
Camaro Z28      13.3   8 350.0 245 3.73 3.840 15.41  0 0   3   4
Pontiac Firebird   19.2   8 400.0 175 3.08 3.845 17.05  0 0   3   2
Fiat X1-9       27.3   4 79.0 66 4.08 1.935 18.90  1 1   4   1
Porsche 914-2     26.0   4 120.3  91 4.43 2.140 16.70  0 1   5   2
Lotus Europa     30.4   4 95.1 113 3.77 1.513 16.90  1 1   5   2
Ford Pantera L     15.8   8 351.0 264 4.22 3.170 14.50  0 1   5   4
Ferrari Dino     19.7   6 145.0 175 3.62 2.770 15.50  0 1   5   6
Maserati Bora     15.0   8 301.0 335 3.54 3.570 14.60  0 1   5   8
Volvo 142E       21.4   4 121.0 109 4.11 2.780 18.60  1 1   4   2
> iris    #inbuilt dataset
```

|    | Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species |
|----|--------------|-------------|--------------|-------------|---------|
| 1  | 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| 2  | 4.9 | 3.0 | 1.4 | 0.2 | setosa |
| 3  | 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| 4  | 4.6 | 3.1 | 1.5 | 0.2 | setosa |
| 5  | 5.0 | 3.6 | 1.4 | 0.2 | setosa |
| 6  | 5.4 | 3.9 | 1.7 | 0.4 | setosa |
| 7  | 4.6 | 3.4 | 1.4 | 0.3 | setosa |
| 8  | 5.0 | 3.4 | 1.5 | 0.2 | setosa |
| 9  | 4.4 | 2.9 | 1.4 | 0.2 | setosa |
| 10 | 4.9 | 3.1 | 1.5 | 0.1 | setosa |
| 11 | 5.4 | 3.7 | 1.5 | 0.2 | setosa |
| 12 | 4.8 | 3.4 | 1.6 | 0.2 | setosa |
| 13 | 4.8 | 3.0 | 1.4 | 0.1 | setosa |
| 14 | 4.3 | 3.0 | 1.1 | 0.1 | setosa |
| 15 | 5.8 | 4.0 | 1.2 | 0.2 | setosa |
| 16 | 5.7 | 4.4 | 1.5 | 0.4 | setosa |
| 17 | 5.4 | 3.9 | 1.3 | 0.4 | setosa |
| 18 | 5.1 | 3.5 | 1.4 | 0.3 | setosa |
| 19 | 5.7 | 3.8 | 1.7 | 0.3 | setosa |
| 20 | 5.1 | 3.8 | 1.5 | 0.3 | setosa |
| 21 | 5.4 | 3.4 | 1.7 | 0.2 | setosa |
| 22 | 5.1 | 3.7 | 1.5 | 0.4 | setosa |
| 23 | 4.6 | 3.6 | 1.0 | 0.2 | setosa |
| 24 | 5.1 | 3.3 | 1.7 | 0.5 | setosa |
| 25 | 4.8 | 3.4 | 1.9 | 0.2 | setosa |

| | | | | |
|---|---|---|---|---|
| 26 | 5.0 | 3.0 | 1.6 | 0.2 setosa |
| 27 | 5.0 | 3.4 | 1.6 | 0.4 setosa |
| 28 | 5.2 | 3.5 | 1.5 | 0.2 setosa |
| 29 | 5.2 | 3.4 | 1.4 | 0.2 setosa |
| 30 | 4.7 | 3.2 | 1.6 | 0.2 setosa |
| 31 | 4.8 | 3.1 | 1.6 | 0.2 setosa |
| 32 | 5.4 | 3.4 | 1.5 | 0.4 setosa |
| 33 | 5.2 | 4.1 | 1.5 | 0.1 setosa |
| 34 | 5.5 | 4.2 | 1.4 | 0.2 setosa |
| 35 | 4.9 | 3.1 | 1.5 | 0.2 setosa |
| 36 | 5.0 | 3.2 | 1.2 | 0.2 setosa |
| 37 | 5.5 | 3.5 | 1.3 | 0.2 setosa |
| 38 | 4.9 | 3.6 | 1.4 | 0.1 setosa |
| 39 | 4.4 | 3.0 | 1.3 | 0.2 setosa |
| 40 | 5.1 | 3.4 | 1.5 | 0.2 setosa |
| 41 | 5.0 | 3.5 | 1.3 | 0.3 setosa |
| 42 | 4.5 | 2.3 | 1.3 | 0.3 setosa |
| 43 | 4.4 | 3.2 | 1.3 | 0.2 setosa |
| 44 | 5.0 | 3.5 | 1.6 | 0.6 setosa |
| 45 | 5.1 | 3.8 | 1.9 | 0.4 setosa |
| 46 | 4.8 | 3.0 | 1.4 | 0.3 setosa |
| 47 | 5.1 | 3.8 | 1.6 | 0.2 setosa |
| 48 | 4.6 | 3.2 | 1.4 | 0.2 setosa |
| 49 | 5.3 | 3.7 | 1.5 | 0.2 setosa |
| 50 | 5.0 | 3.3 | 1.4 | 0.2 setosa |
| 51 | 7.0 | 3.2 | 4.7 | 1.4 versicolor |
| 52 | 6.4 | 3.2 | 4.5 | 1.5 versicolor |
| 53 | 6.9 | 3.1 | 4.9 | 1.5 versicolor |
| 54 | 5.5 | 2.3 | 4.0 | 1.3 versicolor |
| 55 | 6.5 | 2.8 | 4.6 | 1.5 versicolor |
| 56 | 5.7 | 2.8 | 4.5 | 1.3 versicolor |
| 57 | 6.3 | 3.3 | 4.7 | 1.6 versicolor |
| 58 | 4.9 | 2.4 | 3.3 | 1.0 versicolor |
| 59 | 6.6 | 2.9 | 4.6 | 1.3 versicolor |
| 60 | 5.2 | 2.7 | 3.9 | 1.4 versicolor |
| 61 | 5.0 | 2.0 | 3.5 | 1.0 versicolor |
| 62 | 5.9 | 3.0 | 4.2 | 1.5 versicolor |
| 63 | 6.0 | 2.2 | 4.0 | 1.0 versicolor |
| 64 | 6.1 | 2.9 | 4.7 | 1.4 versicolor |
| 65 | 5.6 | 2.9 | 3.6 | 1.3 versicolor |
| 66 | 6.7 | 3.1 | 4.4 | 1.4 versicolor |
| 67 | 5.6 | 3.0 | 4.5 | 1.5 versicolor |
| 68 | 5.8 | 2.7 | 4.1 | 1.0 versicolor |
| 69 | 6.2 | 2.2 | 4.5 | 1.5 versicolor |

| | | | | |
|---|---|---|---|---|
| 70 | 5.6 | 2.5 | 3.9 | 1.1 versicolor |
| 71 | 5.9 | 3.2 | 4.8 | 1.8 versicolor |
| 72 | 6.1 | 2.8 | 4.0 | 1.3 versicolor |
| 73 | 6.3 | 2.5 | 4.9 | 1.5 versicolor |
| 74 | 6.1 | 2.8 | 4.7 | 1.2 versicolor |
| 75 | 6.4 | 2.9 | 4.3 | 1.3 versicolor |
| 76 | 6.6 | 3.0 | 4.4 | 1.4 versicolor |
| 77 | 6.8 | 2.8 | 4.8 | 1.4 versicolor |
| 78 | 6.7 | 3.0 | 5.0 | 1.7 versicolor |
| 79 | 6.0 | 2.9 | 4.5 | 1.5 versicolor |
| 80 | 5.7 | 2.6 | 3.5 | 1.0 versicolor |
| 81 | 5.5 | 2.4 | 3.8 | 1.1 versicolor |
| 82 | 5.5 | 2.4 | 3.7 | 1.0 versicolor |
| 83 | 5.8 | 2.7 | 3.9 | 1.2 versicolor |
| 84 | 6.0 | 2.7 | 5.1 | 1.6 versicolor |
| 85 | 5.4 | 3.0 | 4.5 | 1.5 versicolor |
| 86 | 6.0 | 3.4 | 4.5 | 1.6 versicolor |
| 87 | 6.7 | 3.1 | 4.7 | 1.5 versicolor |
| 88 | 6.3 | 2.3 | 4.4 | 1.3 versicolor |
| 89 | 5.6 | 3.0 | 4.1 | 1.3 versicolor |
| 90 | 5.5 | 2.5 | 4.0 | 1.3 versicolor |
| 91 | 5.5 | 2.6 | 4.4 | 1.2 versicolor |
| 92 | 6.1 | 3.0 | 4.6 | 1.4 versicolor |
| 93 | 5.8 | 2.6 | 4.0 | 1.2 versicolor |
| 94 | 5.0 | 2.3 | 3.3 | 1.0 versicolor |
| 95 | 5.6 | 2.7 | 4.2 | 1.3 versicolor |
| 96 | 5.7 | 3.0 | 4.2 | 1.2 versicolor |
| 97 | 5.7 | 2.9 | 4.2 | 1.3 versicolor |
| 98 | 6.2 | 2.9 | 4.3 | 1.3 versicolor |
| 99 | 5.1 | 2.5 | 3.0 | 1.1 versicolor |
| 100 | 5.7 | 2.8 | 4.1 | 1.3 versicolor |
| 101 | 6.3 | 3.3 | 6.0 | 2.5 virginica |
| 102 | 5.8 | 2.7 | 5.1 | 1.9 virginica |
| 103 | 7.1 | 3.0 | 5.9 | 2.1 virginica |
| 104 | 6.3 | 2.9 | 5.6 | 1.8 virginica |
| 105 | 6.5 | 3.0 | 5.8 | 2.2 virginica |
| 106 | 7.6 | 3.0 | 6.6 | 2.1 virginica |
| 107 | 4.9 | 2.5 | 4.5 | 1.7 virginica |
| 108 | 7.3 | 2.9 | 6.3 | 1.8 virginica |
| 109 | 6.7 | 2.5 | 5.8 | 1.8 virginica |
| 110 | 7.2 | 3.6 | 6.1 | 2.5 virginica |
| 111 | 6.5 | 3.2 | 5.1 | 2.0 virginica |
| 112 | 6.4 | 2.7 | 5.3 | 1.9 virginica |
| 113 | 6.8 | 3.0 | 5.5 | 2.1 virginica |

| 114 | 5.7 | 2.5 | 5.0 | 2.0 | virginica |
| 115 | 5.8 | 2.8 | 5.1 | 2.4 | virginica |
| 116 | 6.4 | 3.2 | 5.3 | 2.3 | virginica |
| 117 | 6.5 | 3.0 | 5.5 | 1.8 | virginica |
| 118 | 7.7 | 3.8 | 6.7 | 2.2 | virginica |
| 119 | 7.7 | 2.6 | 6.9 | 2.3 | virginica |
| 120 | 6.0 | 2.2 | 5.0 | 1.5 | virginica |
| 121 | 6.9 | 3.2 | 5.7 | 2.3 | virginica |
| 122 | 5.6 | 2.8 | 4.9 | 2.0 | virginica |
| 123 | 7.7 | 2.8 | 6.7 | 2.0 | virginica |
| 124 | 6.3 | 2.7 | 4.9 | 1.8 | virginica |
| 125 | 6.7 | 3.3 | 5.7 | 2.1 | virginica |
| 126 | 7.2 | 3.2 | 6.0 | 1.8 | virginica |
| 127 | 6.2 | 2.8 | 4.8 | 1.8 | virginica |
| 128 | 6.1 | 3.0 | 4.9 | 1.8 | virginica |
| 129 | 6.4 | 2.8 | 5.6 | 2.1 | virginica |
| 130 | 7.2 | 3.0 | 5.8 | 1.6 | virginica |
| 131 | 7.4 | 2.8 | 6.1 | 1.9 | virginica |
| 132 | 7.9 | 3.8 | 6.4 | 2.0 | virginica |
| 133 | 6.4 | 2.8 | 5.6 | 2.2 | virginica |
| 134 | 6.3 | 2.8 | 5.1 | 1.5 | virginica |
| 135 | 6.1 | 2.6 | 5.6 | 1.4 | virginica |
| 136 | 7.7 | 3.0 | 6.1 | 2.3 | virginica |
| 137 | 6.3 | 3.4 | 5.6 | 2.4 | virginica |
| 138 | 6.4 | 3.1 | 5.5 | 1.8 | virginica |
| 139 | 6.0 | 3.0 | 4.8 | 1.8 | virginica |
| 140 | 6.9 | 3.1 | 5.4 | 2.1 | virginica |
| 141 | 6.7 | 3.1 | 5.6 | 2.4 | virginica |
| 142 | 6.9 | 3.1 | 5.1 | 2.3 | virginica |
| 143 | 5.8 | 2.7 | 5.1 | 1.9 | virginica |
| 144 | 6.8 | 3.2 | 5.9 | 2.3 | virginica |
| 145 | 6.7 | 3.3 | 5.7 | 2.5 | virginica |
| 146 | 6.7 | 3.0 | 5.2 | 2.3 | virginica |
| 147 | 6.3 | 2.5 | 5.0 | 1.9 | virginica |
| 148 | 6.5 | 3.0 | 5.2 | 2.0 | virginica |
| 149 | 6.2 | 3.4 | 5.4 | 2.3 | virginica |
| 150 | 5.9 | 3.0 | 5.1 | 1.8 | virginica |

```
> names(mtcars)  #variable/column names
 [1] "mpg" "cyl" "disp" "hp"  "drat" "wt"  "qsec" "vs"  "am"  "gear" "carb"
> for (c in names(mtcars)) { print(c) }
[1] "mpg"
[1] "cyl"
[1] "disp"
[1] "hp"
```

```
[1] "drat"
[1] "wt"
[1] "qsec"
[1] "vs"
[1] "am"
[1] "gear"
[1] "carb"
> price <- 12.99
> while (price < 15) {
+   price <- price + 1
+   print(price)
+ }
[1] 13.99
[1] 14.99
[1] 15.99
> check_price <- function(x) {
+   if(x > 110) {
+     print("Price beyond threshold")
+   } else {
+     print("Price within threshold")
+   }
+ }
> check_price(200)
[1] "Price beyond threshold"
> myvect <- c(10, 20, 30, NA, 60, 80)
> mean(myvect)
[1] NA
> sd(myvect)
[1] NA
> min(myvect)
[1] NA
> mean(myvect, na.rm = TRUE)
[1] 40
> stock_price <- c(10, 5, 20, 15, 12, 22)
> matrix_form <- matrix(stock_price, ncol = 2, byrow = TRUE)
> matrix_form
     [,1] [,2]
[1,]  10   5
[2,]  20  15
[3,]  12  22
> apply(matrix_form, 1, sum)
[1] 15 35 34
> apply(matrix_form, 2, sum)
[1] 42 42
```

```
> lapply(1:3, function(x) x ^ 2)  #Returns list
[[1]]
[1] 1

[[2]]
[1] 4

[[3]]
[1] 9

> sapply(1:3, function(x) x ^ 2)  #Returns vector
[1] 1 4 9
> l <- lapply(1:3, function(x) x ^ 2)
> class(l)
[1] "list"
> s <- sapply(1:3, function(x) x ^ 2)
> class(s)
[1] "numeric"
> #Initial Date: 1/1/1970
> purchase_on <- 365
> class(purchase_on) <- "Date"  #Convert to Date & Adds 365 days to the default date
> purchase_on
[1] "1971-01-01"
> purchase_on <- -10
> class(purchase_on) <- "Date"  #Convert to Date & Subtracts 10 days from the default date
> purchase_on
[1] "1969-12-22"
> purchase_date <- as.Date(365, origin=as.Date("2015-03-31"))  #365 days added to origin date
> purchase_date
[1] "2016-03-30"
> sale_date <- as.Date(-10, origin=as.Date("2015-02-10"))     #10 days subtracted from origin date
> sale_date
[1] "2015-01-31"
> format(sale_date, "%Y")
[1] "2015"
> format(sale_date, "%m")
[1] "01"
> format(sale_date, "%b")
[1] "Jan"
> format(sale_date, "%B")
[1] "January"
> Sys.Date()
[1] "2022-02-15"
> format(Sys.Date(), "%d/%m/%Y")
```

```
[1] "15/02/2022"
> as.Date("2021/02/04", format="%Y/%m/%d")   #convert a format of date to date type
[1] "2021-02-04"
> as.Date(purchase_date) > as.Date(sale_date)
[1] TRUE
> as.Date(purchase_date) < as.Date(sale_date)
[1] FALSE
> first_date <- "2020-05-16"
> second_date <- "2020-12-24"
> as.Date(first_date) > as.Date(second_date)
[1] FALSE
> as.Date(first_date) < as.Date(second_date)
[1] TRUE
> dim(housing_df)
[1] 545  13
> str(housing_df)
'data.frame':    545 obs. of  13 variables:
 $ price         : int  13300000 12250000 12250000 12215000 11410000 10850000 10150000 10150000
9870000 9800000 ...
 $ area          : int  7420 8960 9960 7500 7420 7500 8580 16200 8100 5750 ...
 $ bedrooms      : int  4 4 3 4 4 3 4 5 4 3 ...
 $ bathrooms     : int  2 4 2 2 1 3 3 3 1 2 ...
 $ stories       : int  3 4 2 2 2 1 4 2 2 4 ...
 $ mainroad      : chr  "yes" "yes" "yes" "yes" ...
 $ guestroom     : chr  "no" "no" "no" "no" ...
 $ basement      : chr  "no" "no" "yes" "yes" ...
 $ hotwaterheating : chr  "no" "no" "no" "no" ...
 $ airconditioning : chr  "yes" "yes" "no" "yes" ...
 $ parking       : int  2 3 2 3 2 2 2 0 2 1 ...
 $ prefarea      : chr  "yes" "no" "yes" "yes" ...
 $ furnishingstatus: chr  "furnished" "furnished" "semi-furnished" "furnished" ...
> summary(housing_df)
    price              area          bedrooms       bathrooms
 Min.   : 1750000   Min.   : 1650   Min.   :1.000   Min.   :1.000
 1st Qu.: 3430000   1st Qu.: 3600   1st Qu.:2.000   1st Qu.:1.000
 Median : 4340000   Median : 4600   Median :3.000   Median :1.000
 Mean   : 4766729   Mean   : 5151   Mean   :2.965   Mean   :1.286
 3rd Qu.: 5740000   3rd Qu.: 6360   3rd Qu.:3.000   3rd Qu.:2.000
 Max.   :13300000   Max.   :16200   Max.   :6.000   Max.   :4.000
    stories        mainroad          guestroom           basement
 Min.   :1.000   Length:545         Length:545          Length:545
 1st Qu.:1.000   Class :character   Class :character    Class :character
 Median :2.000   Mode  :character   Mode  :character    Mode  :character
 Mean   :1.806
```

```
   3rd Qu.:2.000
   Max.   :4.000
   hotwaterheating   airconditioning     parking        prefarea
   Length:545        Length:545       Min.   :0.0000   Length:545
   Class :character  Class :character  1st Qu.:0.0000   Class :character
   Mode :character   Mode :character   Median :0.0000   Mode :character
                                       Mean   :0.6936
                                       3rd Qu.:1.0000
                                       Max.   :3.0000
   furnishingstatus
   Length:545
   Class :character
   Mode :character
```

## Day 3 – R Programming

```
> ages <- c(34, 45, 26, 32, 21)
> location <- c("Urban", "Rural", "Urban", "Rural", "Urban")
> tapply(ages, location, mean)    #location wise age mean
Rural Urban
 38.5  27.0
> #history()        #get previous command
> setwd("C:/zubeda/PGA02_Zubu/R Programming")  #Set current working directory
> housing_df = read.csv("Housing.csv")
> housing_df
    price  area bedrooms bathrooms stories mainroad guestroom basement hotwaterheating
airconditioning parking
1  13300000  7420    4        2        3      yes      no       no        no           yes   2
2  12250000  8960    4        4        4      yes      no       no        no           yes   3
3  12250000  9960    3        2        2      yes      no       yes       no           no    2
4  12215000  7500    4        2        2      yes      no       yes       no           yes   3
5  11410000  7420    4        1        2      yes      yes      yes       no           yes   2
6  10850000  7500    3        3        1      yes      no       yes       no           yes   2
7  10150000  8580    4        3        4      yes      no       no        no           yes   2
8  10150000 16200    5        3        2      yes      no       no        no           no    0
9   9870000  8100    4        1        2      yes      yes      yes       no           yes   2
10  9800000  5750    3        2        4      yes      yes      no        no           yes   1
11  9800000 13200    3        1        2      yes      no       yes       no           yes   2
12  9681000  6000    4        3        2      yes      yes      yes       yes          no    2
13  9310000  6550    4        2        2      yes      no       no        no           yes   1
14  9240000  3500    4        2        2      yes      no       no        yes          no    2
15  9240000  7800    3        2        2      yes      no       no        no           no    0
16  9100000  6000    4        1        2      yes      no       yes       no           no    2
17  9100000  6600    4        2        2      yes      yes      yes       no           yes   1
18  8960000  8500    3        2        4      yes      no       no        no           yes   2
```

| 19 | 8890000 | 4600 | 3 | 2 | 2 | yes | yes | no | no | yes | 2 |
| 20 | 8855000 | 6420 | 3 | 2 | 2 | yes | no | no | no | yes | 1 |
| 21 | 8750000 | 4320 | 3 | 1 | 2 | yes | no | yes | yes | no | 2 |
| 22 | 8680000 | 7155 | 3 | 2 | 1 | yes | yes | yes | no | yes | 2 |
| 23 | 8645000 | 8050 | 3 | 1 | 1 | yes | yes | yes | no | yes | 1 |
| 24 | 8645000 | 4560 | 3 | 2 | 2 | yes | yes | yes | no | yes | 1 |
| 25 | 8575000 | 8800 | 3 | 2 | 2 | yes | no | no | no | yes | 2 |
| 26 | 8540000 | 6540 | 4 | 2 | 2 | yes | yes | yes | no | yes | 2 |
| 27 | 8463000 | 6000 | 3 | 2 | 4 | yes | yes | yes | no | yes | 0 |
| 28 | 8400000 | 8875 | 3 | 1 | 1 | yes | no | no | no | no | 1 |
| 29 | 8400000 | 7950 | 5 | 2 | 2 | yes | no | yes | yes | no | 2 |
| 30 | 8400000 | 5500 | 4 | 2 | 2 | yes | no | yes | no | yes | 1 |
| 31 | 8400000 | 7475 | 3 | 2 | 4 | yes | no | no | no | yes | 2 |
| 32 | 8400000 | 7000 | 3 | 1 | 4 | yes | no | no | no | yes | 2 |
| 33 | 8295000 | 4880 | 4 | 2 | 2 | yes | no | no | no | yes | 1 |
| 34 | 8190000 | 5960 | 3 | 3 | 2 | yes | yes | yes | no | no | 1 |
| 35 | 8120000 | 6840 | 5 | 1 | 2 | yes | yes | yes | no | yes | 1 |
| 36 | 8080940 | 7000 | 3 | 2 | 4 | yes | no | no | no | yes | 2 |
| 37 | 8043000 | 7482 | 3 | 2 | 3 | yes | no | no | yes | no | 1 |
| 38 | 7980000 | 9000 | 4 | 2 | 4 | yes | no | no | no | yes | 2 |
| 39 | 7962500 | 6000 | 3 | 1 | 4 | yes | yes | no | no | yes | 2 |
| 40 | 7910000 | 6000 | 4 | 2 | 4 | yes | no | no | no | yes | 1 |
| 41 | 7875000 | 6550 | 3 | 1 | 2 | yes | no | yes | no | yes | 0 |
| 42 | 7840000 | 6360 | 3 | 2 | 4 | yes | no | no | no | yes | 0 |
| 43 | 7700000 | 6480 | 3 | 2 | 4 | yes | no | no | no | yes | 2 |
| 44 | 7700000 | 6000 | 4 | 2 | 4 | yes | no | no | no | no | 2 |
| 45 | 7560000 | 6000 | 4 | 2 | 4 | yes | no | no | no | yes | 1 |
| 46 | 7560000 | 6000 | 3 | 2 | 3 | yes | no | no | no | yes | 0 |
| 47 | 7525000 | 6000 | 3 | 2 | 4 | yes | no | no | no | yes | 1 |
| 48 | 7490000 | 6600 | 3 | 1 | 4 | yes | no | no | no | yes | 3 |
| 49 | 7455000 | 4300 | 3 | 2 | 2 | yes | no | yes | no | no | 1 |
| 50 | 7420000 | 7440 | 3 | 2 | 1 | yes | yes | yes | no | yes | 0 |
| 51 | 7420000 | 7440 | 3 | 2 | 4 | yes | no | no | no | no | 1 |
| 52 | 7420000 | 6325 | 3 | 1 | 4 | yes | no | no | no | yes | 1 |
| 53 | 7350000 | 6000 | 4 | 2 | 4 | yes | yes | no | no | yes | 1 |
| 54 | 7350000 | 5150 | 3 | 2 | 4 | yes | no | no | no | yes | 2 |
| 55 | 7350000 | 6000 | 3 | 2 | 2 | yes | yes | no | no | yes | 1 |
| 56 | 7350000 | 6000 | 3 | 1 | 2 | yes | no | no | no | yes | 1 |
| 57 | 7343000 | 11440 | 4 | 1 | 2 | yes | no | yes | no | no | 1 |
| 58 | 7245000 | 9000 | 4 | 2 | 4 | yes | yes | no | no | yes | 1 |
| 59 | 7210000 | 7680 | 4 | 2 | 4 | yes | yes | no | no | yes | 1 |
| 60 | 7210000 | 6000 | 3 | 2 | 4 | yes | yes | no | no | yes | 1 |
| 61 | 7140000 | 6000 | 3 | 2 | 2 | yes | yes | no | no | no | 1 |
| 62 | 7070000 | 8880 | 2 | 1 | 1 | yes | no | no | no | yes | 1 |

| | price | area | bedrooms | bathrooms | stories | mainroad | guestroom | basement | hotwaterheating | airconditioning | parking |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 63 | 7070000 | 6240 | 4 | 2 | 2 | yes | no | no | no | yes | 1 |
| 64 | 7035000 | 6360 | 4 | 2 | 3 | yes | no | no | no | yes | 2 |
| 65 | 7000000 | 11175 | 3 | 1 | 1 | yes | no | yes | no | yes | 1 |
| 66 | 6930000 | 8880 | 3 | 2 | 2 | yes | no | yes | no | yes | 1 |
| 67 | 6930000 | 13200 | 2 | 1 | 1 | yes | no | yes | yes | no | 1 |
| 68 | 6895000 | 7700 | 3 | 2 | 1 | yes | no | no | no | no | 2 |
| 69 | 6860000 | 6000 | 3 | 1 | 1 | yes | no | no | no | yes | 1 |
| 70 | 6790000 | 12090 | 4 | 2 | 2 | yes | no | no | no | no | 2 |
| 71 | 6790000 | 4000 | 3 | 2 | 2 | yes | no | yes | no | yes | 0 |
| 72 | 6755000 | 6000 | 4 | 2 | 4 | yes | no | no | no | yes | 0 |
| 73 | 6720000 | 5020 | 3 | 1 | 4 | yes | no | no | no | yes | 0 |
| 74 | 6685000 | 6600 | 2 | 2 | 4 | yes | no | yes | no | no | 0 |
| 75 | 6650000 | 4040 | 3 | 1 | 2 | yes | no | yes | yes | no | 1 |
| 76 | 6650000 | 4260 | 4 | 2 | 2 | yes | no | no | yes | no | 0 |

| | prefarea | furnishingstatus |
|---|---|---|
| 1 | yes | furnished |
| 2 | no | furnished |
| 3 | yes | semi-furnished |
| 4 | yes | furnished |
| 5 | no | furnished |
| 6 | yes | semi-furnished |
| 7 | yes | semi-furnished |
| 8 | no | unfurnished |
| 9 | yes | furnished |
| 10 | yes | unfurnished |
| 11 | yes | furnished |
| 12 | no | semi-furnished |
| 13 | yes | semi-furnished |
| 14 | no | furnished |
| 15 | yes | semi-furnished |
| 16 | no | semi-furnished |
| 17 | yes | unfurnished |
| 18 | no | furnished |
| 19 | no | furnished |
| 20 | yes | semi-furnished |
| 21 | no | semi-furnished |
| 22 | no | unfurnished |
| 23 | no | furnished |
| 24 | no | furnished |
| 25 | no | furnished |
| 26 | yes | furnished |
| 27 | yes | semi-furnished |
| 28 | no | semi-furnished |
| 29 | no | unfurnished |

| 30 | yes | semi-furnished |
|----|-----|----------------|
| 31 | no  | unfurnished    |
| 32 | no  | semi-furnished |
| 33 | yes | furnished      |
| 34 | no  | unfurnished    |
| 35 | no  | furnished      |
| 36 | no  | furnished      |
| 37 | yes | furnished      |
| 38 | no  | furnished      |
| 39 | no  | unfurnished    |
| 40 | no  | semi-furnished |
| 41 | yes | furnished      |
| 42 | yes | furnished      |
| 43 | no  | unfurnished    |
| 44 | no  | semi-furnished |
| 45 | no  | furnished      |
| 46 | no  | semi-furnished |
| 47 | no  | furnished      |
| 48 | yes | furnished      |
| 49 | no  | unfurnished    |
| 50 | yes | semi-furnished |
| 51 | yes | unfurnished    |
| 52 | no  | unfurnished    |
| 53 | no  | furnished      |
| 54 | no  | semi-furnished |
| 55 | no  | semi-furnished |
| 56 | no  | unfurnished    |
| 57 | yes | semi-furnished |
| 58 | yes | furnished      |
| 59 | no  | semi-furnished |
| 60 | no  | furnished      |
| 61 | no  | semi-furnished |
| 62 | no  | semi-furnished |
| 63 | no  | furnished      |
| 64 | yes | furnished      |
| 65 | yes | furnished      |
| 66 | no  | furnished      |
| 67 | no  | furnished      |
| 68 | no  | unfurnished    |
| 69 | no  | furnished      |
| 70 | yes | furnished      |
| 71 | yes | semi-furnished |
| 72 | no  | unfurnished    |
| 73 | yes | unfurnished    |

```
74    yes      furnished
75    no       furnished
76    no    semi-furnished
 [ reached 'max' / getOption("max.print") -- omitted 469 rows ]
> dev.off()          #clear plot window
null device
      1
> par(mfrow=c(2,1))    #subplots/partions of 2 rows, 1 col
> #Univariate Analysis
> hist(housing_df$area, col = "orange")
> boxplot(housing_df$area, col = "light blue")
> dev.off()
null device
      1
> boxplot(housing_df$area, horizontal = T, col = "light blue")
> dev.off()
null device
      1
> summary(mtcars)
     mpg          cyl          disp         hp          drat          wt          qsec
 Min.  :10.40  Min.  :4.000  Min.  : 71.1  Min.  : 52.0  Min.  :2.760  Min.  :1.513  Min.  :14.50
 1st Qu.:15.43  1st Qu.:4.000  1st Qu.:120.8  1st Qu.: 96.5  1st Qu.:3.080  1st Qu.:2.581  1st Qu.:16.89
 Median :19.20  Median :6.000  Median :196.3  Median :123.0  Median :3.695  Median :3.325
Median :17.71
 Mean  :20.09  Mean  :6.188  Mean  :230.7  Mean  :146.7  Mean  :3.597  Mean  :3.217  Mean
:17.85
 3rd Qu.:22.80  3rd Qu.:8.000  3rd Qu.:326.0  3rd Qu.:180.0  3rd Qu.:3.920  3rd Qu.:3.610  3rd
Qu.:18.90
 Max.  :33.90  Max.  :8.000  Max.  :472.0  Max.  :335.0  Max.  :4.930  Max.  :5.424  Max.  :22.90
     vs           am           gear          carb
 Min.  :0.0000  Min.  :0.0000  Min.  :3.000  Min.  :1.000
 1st Qu.:0.0000  1st Qu.:0.0000  1st Qu.:3.000  1st Qu.:2.000
 Median :0.0000  Median :0.0000  Median :4.000  Median :2.000
 Mean  :0.4375  Mean  :0.4062  Mean  :3.688  Mean  :2.812
 3rd Qu.:1.0000  3rd Qu.:1.0000  3rd Qu.:4.000  3rd Qu.:4.000
 Max.  :1.0000  Max.  :1.0000  Max.  :5.000  Max.  :8.000
> #Bivariate Analysis
> table(mtcars$vs, mtcars$gear)   #Frequency table/Cross table


    3  4  5
 0 12  2  4
 1  3 10  1
> #row index - vs, col index - gear
> df_numeric_vars <- Filter(is.numeric, housing_df)   #Filter(condition, df)
```

```
> names(df_numeric_vars)
[1] "price"    "area"    "bedrooms" "bathrooms" "stories"  "parking"
> df_categorical_vars <- Filter(is.factor, housing_df)
> names(df_categorical_vars)
character(0)
> rownames(mtcars)
 [1] "Mazda RX4"         "Mazda RX4 Wag"    "Datsun 710"        "Hornet 4 Drive"
 [5] "Hornet Sportabout" "Valiant"          "Duster 360"        "Merc 240D"
 [9] "Merc 230"          "Merc 280"         "Merc 280C"         "Merc 450SE"
[13] "Merc 450SL"        "Merc 450SLC"      "Cadillac Fleetwood" "Lincoln Continental"
[17] "Chrysler Imperial" "Fiat 128"         "Honda Civic"       "Toyota Corolla"
[21] "Toyota Corona"     "Dodge Challenger" "AMC Javelin"       "Camaro Z28"
[25] "Pontiac Firebird"  "Fiat X1-9"        "Porsche 914-2"     "Lotus Europa"
[29] "Ford Pantera L"    "Ferrari Dino"     "Maserati Bora"     "Volvo 142E"
> #?data/fn/keyword - get help documentation internally
> #??data/fn/keyword - get help documentation online
> ?mtcars
> ?iris
> counts <- table(mtcars$vs, mtcars$gear)
> #Side by Side barplot
> barplot(counts, main="Car Distribution by Gears and VS", xlab="Number of Gears", ylab="Frequency",
col=c("darkblue", "red"), legend=rownames(counts), beside=TRUE)
> dev.off()
null device
        1
> #Stacked barplot
> barplot(counts, main="Car Distribution by Gears and VS", xlab="Number of Gears", ylab="Frequency",
col=c("darkblue", "red"), legend=rownames(counts), names.arg=c("3", "4", "5"))
> #names.arg - label appear at the bottom of each bar
> nas <- sapply(housing_df, function(X) sum(is.na(x)))   #Missing value checking
> nas
        price          area       bedrooms      bathrooms        stories       mainroad
            0             0             0             0             0             0
     guestroom       basement  hotwaterheating airconditioning       parking       prefarea
            0             0             0             0             0             0
furnishingstatus
            0
> missing_percent <- (nas * 100) / (nrow(housing_df))
> missing_percent
        price          area       bedrooms      bathrooms        stories       mainroad
            0             0             0             0             0             0
     guestroom       basement  hotwaterheating airconditioning       parking       prefarea
            0             0             0             0             0             0
furnishingstatus
```

```
          0
> colnames(mtcars)
 [1] "mpg"  "cyl"  "disp" "hp"   "drat" "wt"   "qsec" "vs"   "am"   "gear" "carb"
> names(mtcars)
 [1] "mpg"  "cyl"  "disp" "hp"   "drat" "wt"   "qsec" "vs"   "am"   "gear" "carb"
> dev.off()
null device
          1
> library(dplyr)
> library(ggplot2)
> data.frame(missing_percent, variable=colnames(housing_df))%>%  #redirection operator/pipe
operator for chaining commands with dependency, passing output of one to another
+   ggplot(aes(variable, missing_percent)) +
+   geom_bar(stat="identity") +  #height of bars to represent values in the data
+   labs(x="Features", y="Percent of Missing values") +
+   theme(axis.text.x=element_text(angle=90, hjust=1))
> #aes(reorder(variable col, - or + the variable to be sorted)) sorts output in asc or desc order
> paste("Hello", "Everybody")   #Concats elements seperated by spaces
[1] "Hello Everybody"
> paste("A", "1", sep="")      #Concats elements with no spaces
[1] "A1"
> x <- c(32, 12, 30, 45)
> labels <- c("Mumbai", "Chennai", "Pune", "Banglore")
> pct <- round(x / sum(x) * 100)
> lbls <- paste(labels, pct)
> lbls <- paste(lbls, "%", sep="")
> pct
[1] 27 10 25 38
> lbls
[1] "Mumbai 27%"   "Chennai 10%"  "Pune 25%"     "Banglore 38%"
> pie(x, labels=lbls, col=rainbow(length(lbls)), main="City Pie Chart")   #rainbow(length) will generate 4
hexdecimal values
> legend("topright", c("Mumbai", "Chennai", "Pune", "Banglore"), cex=0.5, fill=rainbow(length(x)))
#cex=Controls zoom of the font
> legend("topright", c("Mumbai", "Chennai", "Pune", "Banglore"), cex=1, fill=rainbow(length(x)))
> #install.packages("Quandl")
> library("Quandl")
```

## Day 4 – R Programming

```
> dev.off()
null device
          1
> setwd("C:/zubeda/PGA02_Zubu/R Programming")
> library("plyr")
```

```
> library("ggplot2")
> df_AP <- read.csv("ADANIPORTS.csv")
> edit(df_AP)
```

| | Date | Symbol | Series | Prev.Close | Open | High | Low | Last | Close | VWAP | Volume | Turnover |
|---|------|--------|--------|-----------|------|------|-----|------|-------|------|--------|----------|
| 1 | 2007-11-27 | MUNDRAPORT | EQ | 440.00 | 770.00 | 1050.00 | 770.00 | 959.00 | 962.90 | 984.72 | 27294366 | 2687719053785000 |
| 2 | 2007-11-28 | MUNDRAPORT | EQ | 962.90 | 984.00 | 990.00 | 874.00 | 885.00 | 893.90 | 941.38 | 4581338 | 431276530165000 |
| 3 | 2007-11-29 | MUNDRAPORT | EQ | 893.90 | 909.00 | 914.75 | 841.00 | 887.00 | 884.20 | 888.09 | 5124121 | 455065846265000 |
| 4 | 2007-11-30 | MUNDRAPORT | EQ | 884.20 | 890.00 | 958.00 | 890.00 | 929.00 | 921.55 | 929.17 | 4609762 | 428325662830000 |
| 5 | 2007-12-03 | MUNDRAPORT | EQ | 921.55 | 939.75 | 995.00 | 922.00 | 980.00 | 969.30 | 965.65 | 2977470 | 287519974300000 |
| 6 | 2007-12-04 | MUNDRAPORT | EQ | 969.30 | 985.00 | 1056.00 | 976.00 | 1049.00 | 1041.45 | 1015.39 | 4849250 | 492386736075000 |
| 7 | 2007-12-05 | MUNDRAPORT | EQ | 1041.45 | 1061.00 | 1099.50 | 1050.00 | 1084.00 | 1082.45 | 1082.79 | 2848209 | 308400973015000 |
| 8 | 2007-12-06 | MUNDRAPORT | EQ | 1082.45 | 1089.00 | 1109.70 | 1051.00 | 1090.10 | 1081.30 | 1087.03 | 1749516 | 190177114020000 |
| 9 | 2007-12-07 | MUNDRAPORT | EQ | 1081.30 | 1100.00 | 1134.00 | 1078.00 | 1100.00 | 1102.40 | 1106.57 | 2247904 | 248746530710000 |
| 10 | 2007-12-10 | MUNDRAPORT | EQ | 1102.40 | 1110.00 | 1110.00 | 1061.10 | 1073.55 | 1075.40 | 1080.38 | 1012350 | 109372679360000 |
| 11 | 2007-12-11 | MUNDRAPORT | EQ | 1075.40 | 1081.00 | 1089.00 | 1041.00 | 1046.00 | 1047.65 | 1067.80 | 810464 | 86541556460000 |
| 12 | 2007-12-12 | MUNDRAPORT | EQ | 1047.65 | 1032.00 | 1065.00 | 1016.00 | 1036.90 | 1036.80 | 1043.92 | 744799 | 77751369165000 |
| 13 | 2007-12-13 | MUNDRAPORT | EQ | 1036.80 | 1040.00 | 1150.00 | 1030.25 | 1131.15 | 1129.95 | 1109.09 | 3067687 | 340233907520000 |
| 14 | 2007-12-14 | MUNDRAPORT | EQ | 1129.95 | 1139.90 | 1140.00 | 1101.10 | 1107.00 | 1110.50 | 1119.55 | 1070737 | 119874627765000 |
| 15 | 2007-12-17 | MUNDRAPORT | EQ | 1110.50 | 1140.00 | 1168.00 | 1021.50 | 1052.00 | 1044.25 | 1102.42 | 1404955 | 154884767715000 |
| 16 | 2007-12-18 | MUNDRAPORT | EQ | 1044.25 | 1045.00 | 1109.90 | 1031.55 | 1085.00 | 1074.95 | 1077.84 | 1226984 | 132249513310000 |
| 17 | 2007-12-19 | MUNDRAPORT | EQ | 1074.95 | 1091.00 | 1116.00 | 1046.30 | 1078.00 | 1066.90 | 1082.93 | 845666 | 91579757645000 |
| 18 | 2007-12-20 | MUNDRAPORT | EQ | 1066.90 | 1083.50 | 1083.50 | 1051.00 | 1067.00 | 1060.20 | 1065.52 | 623288 | 66412706110000 |
| 19 | 2007-12-24 | MUNDRAPORT | EQ | 1060.20 | 1095.00 | 1192.00 | 1085.25 | 1160.00 | 1156.80 | 1160.77 | 2060892 | 239221361310000 |
| 20 | 2007-12-26 | MUNDRAPORT | EQ | 1156.80 | 1175.00 | 1214.00 | 1148.00 | 1212.00 | 1199.90 | 1183.30 | 1467031 | 173593856540000 |

| No | Date | Symbol | Series | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 21 | 2007-12-27 | MUNDRAPORT | EQ | 1199.90 | 1215.00 | 1240.00 | 1204.00 | 1209.00 | 1211.65 | 1222.58 | 977495 | 119506465945000 |
| 22 | 2007-12-28 | MUNDRAPORT | EQ | 1211.65 | 1189.40 | 1274.00 | 1175.00 | 1270.00 | 1249.10 | 1221.31 | 1164138 | 142177280540000 |
| 23 | 2007-12-31 | MUNDRAPORT | EQ | 1249.10 | 1263.35 | 1295.00 | 1261.00 | 1268.00 | 1268.80 | 1277.64 | 737249 | 94194213815000 |
| 24 | 2008-01-01 | MUNDRAPORT | EQ | 1268.80 | 1279.00 | 1319.00 | 1263.70 | 1308.00 | 1296.85 | 1285.72 | 491348 | 63173462100000 |
| 25 | 2008-01-02 | MUNDRAPORT | EQ | 1296.85 | 1310.25 | 1324.00 | 1270.00 | 1300.15 | 1307.45 | 1302.15 | 703815 | 91647340425000 |
| 26 | 2008-01-03 | MUNDRAPORT | EQ | 1307.45 | 1305.00 | 1314.70 | 1261.15 | 1267.15 | 1275.80 | 1289.24 | 505058 | 65114250075000 |
| 27 | 2008-01-04 | MUNDRAPORT | EQ | 1275.80 | 1278.80 | 1294.80 | 1233.00 | 1239.90 | 1240.35 | 1256.03 | 550795 | 69181674340000 |
| 28 | 2008-01-07 | MUNDRAPORT | EQ | 1240.35 | 1240.00 | 1278.90 | 1215.00 | 1233.00 | 1227.25 | 1244.76 | 630963 | 78539769975000 |
| 29 | 2008-01-08 | MUNDRAPORT | EQ | 1227.25 | 1240.00 | 1255.00 | 1185.00 | 1202.00 | 1204.80 | 1217.08 | 530499 | 64565951270000 |
| 30 | 2008-01-09 | MUNDRAPORT | EQ | 1204.80 | 1200.00 | 1210.00 | 1151.00 | 1181.00 | 1180.25 | 1176.37 | 627507 | 73818313330000 |
| 31 | 2008-01-10 | MUNDRAPORT | EQ | 1180.25 | 1185.00 | 1199.80 | 1110.00 | 1118.00 | 1121.55 | 1156.44 | 438806 | 50745246590000 |
| 32 | 2008-01-11 | MUNDRAPORT | EQ | 1121.55 | 1128.00 | 1130.00 | 1063.00 | 1096.00 | 1085.85 | 1087.78 | 616938 | 67109272025000 |
| 33 | 2008-01-14 | MUNDRAPORT | EQ | 1085.85 | 1082.40 | 1082.40 | 1031.10 | 1035.00 | 1035.15 | 1042.40 | 835916 | 87135710755000 |
| 34 | 2008-01-15 | MUNDRAPORT | EQ | 1035.15 | 1045.60 | 1078.70 | 1036.05 | 1057.00 | 1049.55 | 1050.69 | 830493 | 87259337110000 |
| 35 | 2008-01-16 | MUNDRAPORT | EQ | 1049.55 | 1046.00 | 1064.00 | 1000.00 | 1038.30 | 1030.40 | 1032.86 | 816188 | 84300609685000 |
| 36 | 2008-01-17 | MUNDRAPORT | EQ | 1030.40 | 1050.00 | 1053.50 | 1011.00 | 1014.95 | 1020.90 | 1033.73 | 336003 | 34733490900000 |
| 37 | 2008-01-18 | MUNDRAPORT | EQ | 1020.90 | 1010.00 | 1072.00 | 974.90 | 995.00 | 994.60 | 1022.57 | 676854 | 69213280915000 |
| 38 | 2008-01-21 | MUNDRAPORT | EQ | 994.60 | 995.00 | 1005.00 | 795.70 | 853.00 | 825.05 | 880.77 | 788623 | 69459899855000 |
| 39 | 2008-01-22 | MUNDRAPORT | EQ | 825.05 | 700.00 | 810.00 | 660.05 | 739.00 | 735.55 | 703.20 | 546161 | 38406113705000 |
| 40 | 2008-01-23 | MUNDRAPORT | EQ | 735.55 | 760.00 | 881.90 | 760.00 | 862.20 | 857.00 | 818.67 | 535462 | 43836526980000 |
| 41 | 2008-01-24 | MUNDRAPORT | EQ | 857.00 | 875.00 | 935.00 | 812.00 | 814.70 | 814.15 | 854.83 | 511017 | 43683319425000 |
| 42 | 2008-01-25 | MUNDRAPORT | EQ | 814.15 | 820.00 | 883.00 | 820.00 | 866.00 | 865.70 | 858.33 | 404045 | 34680333860000 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 43 | 2008-01-28 | MUNDRAPORT | EQ | 865.70 | 835.00 | 835.00 | 783.20 | 822.00 | 820.80 | 804.38 | 467052 |
| | 37568552380000 | | | | | | | | | |
| 44 | 2008-01-29 | MUNDRAPORT | EQ | 820.80 | 840.00 | 860.00 | 820.05 | 840.00 | 840.75 | 841.27 | 220070 |
| | 18513823345000 | | | | | | | | | |
| 45 | 2008-01-30 | MUNDRAPORT | EQ | 840.75 | 849.80 | 864.00 | 822.25 | 834.00 | 830.45 | 833.82 | 286190 |
| | 23863110660000 | | | | | | | | | |
| 46 | 2008-01-31 | MUNDRAPORT | EQ | 830.45 | 831.00 | 849.90 | 812.55 | 836.60 | 837.65 | 833.58 | 194300 |
| | 16196555895000 | | | | | | | | | |
| 47 | 2008-02-01 | MUNDRAPORT | EQ | 837.65 | 831.65 | 852.30 | 820.00 | 826.00 | 825.35 | 828.09 | 204391 |
| | 16925451805000 | | | | | | | | | |
| 48 | 2008-02-04 | MUNDRAPORT | EQ | 825.35 | 847.90 | 872.40 | 840.00 | 859.00 | 856.10 | 858.77 | 280230 |
| | 24065208695000 | | | | | | | | | |
| 49 | 2008-02-05 | MUNDRAPORT | EQ | 856.10 | 856.00 | 857.00 | 830.00 | 834.65 | 834.30 | 842.06 | 162093 |
| | 13649192020000 | | | | | | | | | |
| 50 | 2008-02-06 | MUNDRAPORT | EQ | 834.30 | 803.00 | 824.90 | 780.00 | 809.00 | 807.50 | 810.50 | 193260 |
| | 15663794125000 | | | | | | | | | |
| 51 | 2008-02-07 | MUNDRAPORT | EQ | 807.50 | 825.00 | 830.00 | 792.00 | 795.90 | 796.25 | 809.53 | 212932 |
| | 17237575975000 | | | | | | | | | |
| 52 | 2008-02-08 | MUNDRAPORT | EQ | 796.25 | 810.00 | 830.00 | 765.15 | 786.00 | 784.05 | 781.48 | 285025 |
| | 22274252000000 | | | | | | | | | |
| 53 | 2008-02-11 | MUNDRAPORT | EQ | 784.05 | 785.00 | 785.00 | 695.00 | 699.00 | 711.20 | 736.23 | 223955 |
| | 16488264325000 | | | | | | | | | |
| 54 | 2008-02-12 | MUNDRAPORT | EQ | 711.20 | 725.00 | 734.95 | 655.60 | 689.00 | 681.30 | 681.38 | 303409 |
| | 20673577510000 | | | | | | | | | |
| 55 | 2008-02-13 | MUNDRAPORT | EQ | 681.30 | 815.90 | 815.90 | 664.00 | 678.00 | 670.95 | 681.68 | 214900 |
| | 14649214640000 | | | | | | | | | |
| 56 | 2008-02-14 | MUNDRAPORT | EQ | 670.95 | 680.00 | 714.00 | 680.00 | 710.00 | 709.80 | 704.71 | 269032 |
| | 18959036175000 | | | | | | | | | |
| 57 | 2008-02-15 | MUNDRAPORT | EQ | 709.80 | 700.00 | 763.70 | 681.25 | 729.00 | 728.75 | 734.23 | 353049 |
| | 25921872820000 | | | | | | | | | |
| 58 | 2008-02-18 | MUNDRAPORT | EQ | 728.75 | 735.00 | 775.00 | 735.00 | 772.00 | 771.60 | 762.33 | 342580 |
| | 26115882900000 | | | | | | | | | |
| 59 | 2008-02-19 | MUNDRAPORT | EQ | 771.60 | 779.00 | 786.90 | 760.20 | 767.00 | 763.90 | 772.24 | 137412 |
| | 10611555840000 | | | | | | | | | |
| 60 | 2008-02-20 | MUNDRAPORT | EQ | 763.90 | 750.00 | 760.00 | 720.00 | 740.00 | 732.10 | 730.61 | 197489 |
| | 14428706935000 | | | | | | | | | |
| 61 | 2008-02-21 | MUNDRAPORT | EQ | 732.10 | 762.00 | 762.00 | 730.10 | 738.90 | 737.60 | 741.53 | 125558 |
| | 9310465240000 | | | | | | | | | |
| 62 | 2008-02-22 | MUNDRAPORT | EQ | 737.60 | 723.00 | 737.00 | 715.00 | 724.50 | 724.00 | 726.52 | 81070 |
| | 5889922195000 | | | | | | | | | |
| 63 | 2008-02-25 | MUNDRAPORT | EQ | 724.00 | 725.05 | 758.90 | 702.30 | 707.00 | 707.65 | 711.70 | 152803 |
| | 10875065635000 | | | | | | | | | |
| 64 | 2008-02-26 | MUNDRAPORT | EQ | 707.65 | 725.00 | 744.00 | 713.00 | 735.00 | 735.80 | 733.73 | 251269 |
| | 18436350425000 | | | | | | | | | |

65 2008-02-27 MUNDRAPORT    EQ    735.80 749.70 783.40 741.00 744.00 746.40 762.47  305320 23279802440000

66 2008-02-28 MUNDRAPORT    EQ    746.40 740.00 754.90 725.05 740.00 737.75 738.91  1124918312092510000

| | Trades | Deliverable.Volume | X.Deliverble |
|---|---|---|---|
| 1 | NA | 9859619 | 0.3612 |
| 2 | NA | 1453278 | 0.3172 |
| 3 | NA | 1069678 | 0.2088 |
| 4 | NA | 1260913 | 0.2735 |
| 5 | NA | 816123 | 0.2741 |
| 6 | NA | 1537667 | 0.3171 |
| 7 | NA | 904260 | 0.3175 |
| 8 | NA | 825691 | 0.4720 |
| 9 | NA | 697763 | 0.3104 |
| 10 | NA | 417514 | 0.4124 |
| 11 | NA | 415191 | 0.5123 |
| 12 | NA | 363848 | 0.4885 |
| 13 | NA | 1040076 | 0.3390 |
| 14 | NA | 525239 | 0.4905 |
| 15 | NA | 670298 | 0.4771 |
| 16 | NA | 449420 | 0.3663 |
| 17 | NA | 344171 | 0.4070 |
| 18 | NA | 276356 | 0.4434 |
| 19 | NA | 807879 | 0.3920 |
| 20 | NA | 469389 | 0.3200 |
| 21 | NA | 355431 | 0.3636 |
| 22 | NA | 503564 | 0.4326 |
| 23 | NA | 316377 | 0.4291 |
| 24 | NA | 172911 | 0.3519 |
| 25 | NA | 221397 | 0.3146 |
| 26 | NA | 217437 | 0.4305 |
| 27 | NA | 230237 | 0.4180 |
| 28 | NA | 239404 | 0.3794 |
| 29 | NA | 228866 | 0.4314 |
| 30 | NA | 259280 | 0.4132 |
| 31 | NA | 200150 | 0.4561 |
| 32 | NA | 312121 | 0.5059 |
| 33 | NA | 570824 | 0.6829 |
| 34 | NA | 504259 | 0.6072 |
| 35 | NA | 478517 | 0.5863 |
| 36 | NA | 145194 | 0.4321 |
| 37 | NA | 278615 | 0.4116 |
| 38 | NA | 474223 | 0.6013 |
| 39 | NA | 376194 | 0.6888 |

```
40    NA        283881    0.5302
41    NA        258346    0.5056
42    NA        178177    0.4410
43    NA        241365    0.5168
44    NA         74141    0.3369
45    NA        165926    0.5798
46    NA        103890    0.5347
47    NA        115715    0.5661
48    NA        128195    0.4575
49    NA         96153    0.5932
50    NA        110565    0.5721
51    NA        106275    0.4991
52    NA        154857    0.5433
53    NA        118002    0.5269
54    NA        187180    0.6169
55    NA        108761    0.5061
56    NA        148611    0.5524
57    NA        110621    0.3133
58    NA        154099    0.4498
59    NA         47543    0.3460
60    NA         89397    0.4527
61    NA         37956    0.3023
62    NA         31808    0.3924
63    NA         71403    0.4673
64    NA         53136    0.2115
65    NA         84490    0.2767
66    NA         36730    0.3265
 [ reached 'max' / getOption("max.print") -- omitted 3256 rows ]
> names(df_AP)
 [1] "Date"          "Symbol"         "Series"         "Prev.Close"       "Open"
 [6] "High"          "Low"            "Last"           "Close"            "VWAP"
[11] "Volume"        "Turnover"       "Trades"         "Deliverable.Volume" "X.Deliverble"
> head(df_AP)    #get first 6 rows
      Date    Symbol Series Prev.Close  Open   High Low Last  Close   VWAP  Volume        Turnover
Trades
1 2007-11-27 MUNDRAPORT    EQ    440.00 770.00 1050.00 770  959  962.90  984.72 27294366
2687719053785000    NA
2 2007-11-28 MUNDRAPORT    EQ    962.90 984.00  990.00 874  885  893.90  941.38 4581338
431276530165000    NA
3 2007-11-29 MUNDRAPORT    EQ    893.90 909.00  914.75 841  887  884.20  888.09 5124121
455065846265000    NA
4 2007-11-30 MUNDRAPORT    EQ    884.20 890.00  958.00 890  929  921.55  929.17 4609762
428325662830000    NA
```

```
5 2007-12-03 MUNDRAPORT    EQ    921.55 939.75  995.00 922  980  969.30  965.65  2977470
287519974300000    NA
6 2007-12-04 MUNDRAPORT    EQ    969.30 985.00 1056.00 976 1049 1041.45 1015.39  4849250
492386736075000    NA
  Deliverable.Volume X.Deliverble
1       9859619    0.3612
2       1453278    0.3172
3       1069678    0.2088
4       1260913    0.2735
5        816123    0.2741
6       1537667    0.3171
> v <- c(8, 14, 26, 5, 43)
> plot(v, type="o")  #Line plot with points
> plot(v, type="p")  #Points plot
> plot(v, type="l")  #Line plot without points
> plot(v, type="o", col="red", xlab="Month", ylab="Rainfall", main="Rainfall Chart")
> v <- c(12, 14, 28, 5, 44)
> t <- c(15, 8, 8, 10, 13)
> plot(v, type="o", col="blue", xlab="Month", ylab="Rainfall", main="Rainfall Chart")
> lines(t, type="o", col="red")
> df_aapl <- read.csv("AAPL.csv")
> head(df_aapl)
     Date  Open  High   Low Close Adj.Close   Volume
1 2021-02-17 131.25 132.22 129.47 130.84  130.0669  97918500
2 2021-02-18 129.20 130.00 127.41 129.71  128.9436  96856700
3 2021-02-19 130.24 130.71 128.80 129.87  129.1027  87668800
4 2021-02-22 128.01 129.72 125.60 126.00  125.2555 103916400
5 2021-02-23 123.76 126.71 118.39 125.86  125.1164 158273000
6 2021-02-24 124.94 125.56 122.23 125.35  124.6094 111039900
> df_waltdisney <- read.csv("DIS.csv")
> head(df_waltdisney)
     Date  Open  High   Low Close Adj.Close   Volume
1 2021-02-17 185.36 187.63 182.16 186.44    186.44 11391800
2 2021-02-18 184.79 186.40 182.84 183.00    183.00 12380900
3 2021-02-19 184.27 184.78 182.79 183.65    183.65  8834500
4 2021-02-22 181.74 194.02 181.53 191.76    191.76 18799600
5 2021-02-23 193.59 198.94 188.66 197.09    197.09 23191400
6 2021-02-24 197.58 200.60 195.33 197.51    197.51 16205900
> df_nike <- read.csv("NKE.csv")
> head(df_nike)
     Date  Open  High   Low Close Adj.Close   Volume
1 2021-02-17 141.30 144.56 140.21 143.99  142.9153  6437100
2 2021-02-18 142.98 145.39 141.21 145.09  144.0071  4486800
3 2021-02-19 145.43 145.50 141.50 142.02  140.9601  7486000
```

```
4 2021-02-22 141.54 142.46 136.26 136.67  135.6500  8985900
5 2021-02-23 136.03 136.83 131.58 136.13  135.1140 10364100
6 2021-02-24 135.06 135.96 133.95 135.65  134.6376  6360900
> df_aapl <- cbind(df_aapl, Stock="")
> df_waltdisney <- cbind(df_waltdisney, Stock="")
> df_nike <- cbind(df_nike, Stock="")
> head(df_aapl)
     Date  Open   High   Low  Close Adj.Close   Volume Stock
1 2021-02-17 131.25 132.22 129.47 130.84  130.0669  97918500
2 2021-02-18 129.20 130.00 127.41 129.71  128.9436  96856700
3 2021-02-19 130.24 130.71 128.80 129.87  129.1027  87668800
4 2021-02-22 128.01 129.72 125.60 126.00  125.2555 103916400
5 2021-02-23 123.76 126.71 118.39 125.86  125.1164 158273000
6 2021-02-24 124.94 125.56 122.23 125.35  124.6094 111039900
> head(df_waltdisney)
     Date  Open   High   Low  Close Adj.Close   Volume Stock
1 2021-02-17 185.36 187.63 182.16 186.44    186.44 11391800
2 2021-02-18 184.79 186.40 182.84 183.00    183.00 12380900
3 2021-02-19 184.27 184.78 182.79 183.65    183.65  8834500
4 2021-02-22 181.74 194.02 181.53 191.76    191.76 18799600
5 2021-02-23 193.59 198.94 188.66 197.09    197.09 23191400
6 2021-02-24 197.58 200.60 195.33 197.51    197.51 16205900
> head(df_nike)
     Date  Open   High   Low  Close Adj.Close   Volume Stock
1 2021-02-17 141.30 144.56 140.21 143.99  142.9153  6437100
2 2021-02-18 142.98 145.39 141.21 145.09  144.0071  4486800
3 2021-02-19 145.43 145.50 141.50 142.02  140.9601  7486000
4 2021-02-22 141.54 142.46 136.26 136.67  135.6500  8985900
5 2021-02-23 136.03 136.83 131.58 136.13  135.1140 10364100
6 2021-02-24 135.06 135.96 133.95 135.65  134.6376  6360900
> df_aapl$Stock <- paste(df_aapl$Stock, "Bertrandt", sep="")
> df_waltdisney$Stock <- paste(df_waltdisney$Stock, "Deutsche Bank", sep="")
> df_nike$Stock <- paste(df_nike$Stock, "Siemens", sep="")
> head(df_aapl)
     Date  Open   High   Low  Close Adj.Close   Volume    Stock
1 2021-02-17 131.25 132.22 129.47 130.84  130.0669  97918500 Bertrandt
2 2021-02-18 129.20 130.00 127.41 129.71  128.9436  96856700 Bertrandt
3 2021-02-19 130.24 130.71 128.80 129.87  129.1027  87668800 Bertrandt
4 2021-02-22 128.01 129.72 125.60 126.00  125.2555 103916400 Bertrandt
5 2021-02-23 123.76 126.71 118.39 125.86  125.1164 158273000 Bertrandt
6 2021-02-24 124.94 125.56 122.23 125.35  124.6094 111039900 Bertrandt
> head(df_waltdisney)
     Date  Open   High   Low  Close Adj.Close   Volume      Stock
1 2021-02-17 185.36 187.63 182.16 186.44    186.44 11391800 Deutsche Bank
```

```
2 2021-02-18 184.79 186.40 182.84 183.00    183.00 12380900 Deutsche Bank
3 2021-02-19 184.27 184.78 182.79 183.65    183.65  8834500 Deutsche Bank
4 2021-02-22 181.74 194.02 181.53 191.76    191.76 18799600 Deutsche Bank
5 2021-02-23 193.59 198.94 188.66 197.09    197.09 23191400 Deutsche Bank
6 2021-02-24 197.58 200.60 195.33 197.51    197.51 16205900 Deutsche Bank
> head(df_nike)
     Date  Open  High   Low Close Adj.Close   Volume   Stock
1 2021-02-17 141.30 144.56 140.21 143.99  142.9153  6437100 Siemens
2 2021-02-18 142.98 145.39 141.21 145.09  144.0071  4486800 Siemens
3 2021-02-19 145.43 145.50 141.50 142.02  140.9601  7486000 Siemens
4 2021-02-22 141.54 142.46 136.26 136.67  135.6500  8985900 Siemens
5 2021-02-23 136.03 136.83 131.58 136.13  135.1140 10364100 Siemens
6 2021-02-24 135.06 135.96 133.95 135.65  134.6376  6360900 Siemens
> df_allStocks <- rbind(df_aapl, df_waltdisney, df_nike)
> df_allStocks
     Date  Open  High   Low Close Adj.Close    Volume     Stock
1  2021-02-17 131.25 132.22 129.47 130.84  130.0669  97918500 Bertrandt
2  2021-02-18 129.20 130.00 127.41 129.71  128.9436  96856700 Bertrandt
3  2021-02-19 130.24 130.71 128.80 129.87  129.1027  87668800 Bertrandt
4  2021-02-22 128.01 129.72 125.60 126.00  125.2555 103916400 Bertrandt
5  2021-02-23 123.76 126.71 118.39 125.86  125.1164 158273000 Bertrandt
6  2021-02-24 124.94 125.56 122.23 125.35  124.6094 111039900 Bertrandt
7  2021-02-25 124.68 126.46 120.54 120.99  120.2751 148199500 Bertrandt
8  2021-02-26 122.59 124.85 121.20 121.26  120.5436 164560400 Bertrandt
9  2021-03-01 123.75 127.93 122.79 127.79  127.0350 116307900 Bertrandt
10 2021-03-02 128.41 128.72 125.01 125.12  124.3807 102260900 Bertrandt
11 2021-03-03 124.81 125.71 121.84 122.06  121.3388 112966300 Bertrandt
12 2021-03-04 121.75 123.60 118.62 120.13  119.4202 178155000 Bertrandt
13 2021-03-05 120.98 121.94 117.57 121.42  120.7026 153766600 Bertrandt
14 2021-03-08 120.93 121.00 116.21 116.36  115.6725 154376600 Bertrandt
15 2021-03-09 119.03 122.06 118.79 121.09  120.3745 129525800 Bertrandt
16 2021-03-10 121.69 122.17 119.45 119.98  119.2711 111943300 Bertrandt
17 2021-03-11 122.54 123.21 121.26 121.96  121.2394 103026500 Bertrandt
18 2021-03-12 120.40 121.17 119.16 121.03  120.3149  88105100 Bertrandt
19 2021-03-15 121.41 124.00 120.42 123.99  123.2574  92403800 Bertrandt
20 2021-03-16 125.70 127.22 124.72 125.57  124.8281 115227900 Bertrandt
21 2021-03-17 124.05 125.86 122.34 124.76  124.0229 111932600 Bertrandt
22 2021-03-18 122.88 123.18 120.32 120.53  119.8179 121229700 Bertrandt
23 2021-03-19 119.90 121.43 119.68 119.99  119.2811 185549500 Bertrandt
24 2021-03-22 120.33 123.87 120.26 123.39  122.6610 111912300 Bertrandt
25 2021-03-23 123.33 124.24 122.14 122.54  121.8160  95467100 Bertrandt
26 2021-03-24 122.82 122.90 120.07 120.09  119.3805  88530500 Bertrandt
27 2021-03-25 119.54 121.66 119.00 120.59  119.8775  98844700 Bertrandt
28 2021-03-26 120.35 121.48 118.92 121.21  120.4938  94071200 Bertrandt
```

| 29 | 2021-03-29 | 121.65 | 122.58 | 120.73 | 121.39 | 120.6728 | 80819200 | Bertrandt |
| 30 | 2021-03-30 | 120.11 | 120.40 | 118.86 | 119.90 | 119.1916 | 85671900 | Bertrandt |
| 31 | 2021-03-31 | 121.65 | 123.52 | 121.15 | 122.15 | 121.4283 | 118323800 | Bertrandt |
| 32 | 2021-04-01 | 123.66 | 124.18 | 122.49 | 123.00 | 122.2733 | 75089100 | Bertrandt |
| 33 | 2021-04-05 | 123.87 | 126.16 | 123.07 | 125.90 | 125.1561 | 88651200 | Bertrandt |
| 34 | 2021-04-06 | 126.50 | 127.13 | 125.65 | 126.21 | 125.4643 | 80171300 | Bertrandt |
| 35 | 2021-04-07 | 125.83 | 127.92 | 125.14 | 127.90 | 127.1443 | 83466700 | Bertrandt |
| 36 | 2021-04-08 | 128.95 | 130.39 | 128.52 | 130.36 | 129.5898 | 88844600 | Bertrandt |
| 37 | 2021-04-09 | 129.80 | 133.04 | 129.47 | 133.00 | 132.2142 | 106686700 | Bertrandt |
| 38 | 2021-04-12 | 132.52 | 132.85 | 130.63 | 131.24 | 130.4646 | 91420000 | Bertrandt |
| 39 | 2021-04-13 | 132.44 | 134.66 | 131.93 | 134.43 | 133.6357 | 91266500 | Bertrandt |
| 40 | 2021-04-14 | 134.94 | 135.00 | 131.66 | 132.03 | 131.2499 | 87222800 | Bertrandt |
| 41 | 2021-04-15 | 133.82 | 135.00 | 133.64 | 134.50 | 133.7053 | 89347100 | Bertrandt |
| 42 | 2021-04-16 | 134.30 | 134.67 | 133.28 | 134.16 | 133.3673 | 84922400 | Bertrandt |
| 43 | 2021-04-19 | 133.51 | 135.47 | 133.34 | 134.84 | 134.0433 | 94264200 | Bertrandt |
| 44 | 2021-04-20 | 135.02 | 135.53 | 131.81 | 133.11 | 132.3235 | 94812300 | Bertrandt |
| 45 | 2021-04-21 | 132.36 | 133.75 | 131.30 | 133.50 | 132.7112 | 68847100 | Bertrandt |
| 46 | 2021-04-22 | 133.04 | 134.15 | 131.41 | 131.94 | 131.1605 | 84566500 | Bertrandt |
| 47 | 2021-04-23 | 132.16 | 135.12 | 132.16 | 134.32 | 133.5264 | 78657500 | Bertrandt |
| 48 | 2021-04-26 | 134.83 | 135.06 | 133.56 | 134.72 | 133.9240 | 66905100 | Bertrandt |
| 49 | 2021-04-27 | 135.01 | 135.41 | 134.11 | 134.39 | 133.5960 | 66015800 | Bertrandt |
| 50 | 2021-04-28 | 134.31 | 135.02 | 133.08 | 133.58 | 132.7907 | 107760100 | Bertrandt |
| 51 | 2021-04-29 | 136.47 | 137.07 | 132.45 | 133.48 | 132.6913 | 151101000 | Bertrandt |
| 52 | 2021-04-30 | 131.78 | 133.56 | 131.07 | 131.46 | 130.6833 | 109839500 | Bertrandt |
| 53 | 2021-05-03 | 132.04 | 134.07 | 131.83 | 132.54 | 131.7569 | 75135100 | Bertrandt |
| 54 | 2021-05-04 | 131.19 | 131.49 | 126.70 | 127.85 | 127.0946 | 137564700 | Bertrandt |
| 55 | 2021-05-05 | 129.20 | 130.45 | 127.97 | 128.10 | 127.3431 | 84000900 | Bertrandt |
| 56 | 2021-05-06 | 127.89 | 129.75 | 127.13 | 129.74 | 128.9735 | 78128300 | Bertrandt |
| 57 | 2021-05-07 | 130.85 | 131.26 | 129.48 | 130.21 | 129.6606 | 78973300 | Bertrandt |
| 58 | 2021-05-10 | 129.41 | 129.54 | 126.81 | 126.85 | 126.3147 | 88071200 | Bertrandt |
| 59 | 2021-05-11 | 123.50 | 126.27 | 122.77 | 125.91 | 125.3787 | 126142800 | Bertrandt |
| 60 | 2021-05-12 | 123.40 | 124.64 | 122.25 | 122.77 | 122.2519 | 112172300 | Bertrandt |
| 61 | 2021-05-13 | 124.58 | 126.15 | 124.26 | 124.97 | 124.4426 | 105861300 | Bertrandt |
| 62 | 2021-05-14 | 126.25 | 127.89 | 125.85 | 127.45 | 126.9122 | 81918000 | Bertrandt |
| 63 | 2021-05-17 | 126.82 | 126.93 | 125.17 | 126.27 | 125.7372 | 74244600 | Bertrandt |
| 64 | 2021-05-18 | 126.56 | 126.99 | 124.78 | 124.85 | 124.3232 | 63342900 | Bertrandt |
| 65 | 2021-05-19 | 123.16 | 124.92 | 122.86 | 124.69 | 124.1638 | 92612000 | Bertrandt |
| 66 | 2021-05-20 | 125.23 | 127.72 | 125.10 | 127.31 | 126.7728 | 76857100 | Bertrandt |
| 67 | 2021-05-21 | 127.82 | 128.00 | 125.21 | 125.43 | 124.9007 | 79295400 | Bertrandt |
| 68 | 2021-05-24 | 126.01 | 127.94 | 125.94 | 127.10 | 126.5637 | 63092900 | Bertrandt |
| 69 | 2021-05-25 | 127.82 | 128.32 | 126.32 | 126.90 | 126.3645 | 72009500 | Bertrandt |
| 70 | 2021-05-26 | 126.96 | 127.39 | 126.42 | 126.85 | 126.3147 | 56575900 | Bertrandt |
| 71 | 2021-05-27 | 126.44 | 127.64 | 125.08 | 125.28 | 124.7513 | 94625600 | Bertrandt |
| 72 | 2021-05-28 | 125.57 | 125.80 | 124.55 | 124.61 | 124.0842 | 71311100 | Bertrandt |

```
73  2021-06-01 125.08 125.35 123.94 124.28  123.7556  67637100 Bertrandt
74  2021-06-02 124.28 125.24 124.05 125.06  124.5323  59278900 Bertrandt
75  2021-06-03 124.68 124.85 123.13 123.54  123.0187  76229200 Bertrandt
76  2021-06-04 124.07 126.16 123.85 125.89  125.3588  75169300 Bertrandt
77  2021-06-07 126.17 126.32 124.83 125.90  125.3687  71057600 Bertrandt
78  2021-06-08 126.60 128.46 126.21 126.74  126.2052  74403800 Bertrandt
79  2021-06-09 127.21 127.75 126.52 127.13  126.5935  56877900 Bertrandt
80  2021-06-10 127.02 128.19 125.94 126.11  125.5778  71186400 Bertrandt
81  2021-06-11 126.53 127.44 126.10 127.35  126.8126  53522400 Bertrandt
82  2021-06-14 127.82 130.54 127.07 130.48  129.9294  96906500 Bertrandt
83  2021-06-15 129.94 130.60 129.39 129.64  129.0929  62746300 Bertrandt
84  2021-06-16 130.37 130.89 128.46 130.15  129.6008  91815000 Bertrandt
85  2021-06-17 129.80 132.55 129.65 131.79  131.2339  96721700 Bertrandt
86  2021-06-18 130.71 131.51 130.24 130.46  129.9095 108953300 Bertrandt
87  2021-06-21 130.30 132.41 129.21 132.30  131.7417  79663300 Bertrandt
88  2021-06-22 132.13 134.08 131.62 133.98  133.4146  74783600 Bertrandt
89  2021-06-23 133.77 134.32 133.23 133.70  133.1358  60214200 Bertrandt
90  2021-06-24 134.45 134.64 132.93 133.41  132.8470  68711000 Bertrandt
91  2021-06-25 133.46 133.89 132.81 133.11  132.5483  70783700 Bertrandt
92  2021-06-28 133.41 135.25 133.35 134.78  134.2113  62111300 Bertrandt
93  2021-06-29 134.80 136.49 134.35 136.33  135.7547  64556100 Bertrandt
94  2021-06-30 136.17 137.41 135.87 136.96  136.3821  63261400 Bertrandt
95  2021-07-01 136.60 137.33 135.76 137.27  136.6908  52485800 Bertrandt
96  2021-07-02 137.90 140.00 137.75 139.96  139.3694  78852600 Bertrandt
97  2021-07-06 140.07 143.15 140.07 142.02  141.4207 108181800 Bertrandt
98  2021-07-07 143.54 144.89 142.66 144.57  143.9599 104911600 Bertrandt
99  2021-07-08 141.58 144.06 140.67 143.24  142.6355 105575500 Bertrandt
100 2021-07-09 142.75 145.65 142.65 145.11  144.4977  99890800 Bertrandt
101 2021-07-12 146.21 146.32 144.00 144.50  143.8902  76299700 Bertrandt
102 2021-07-13 144.03 147.46 143.63 145.64  145.0254 100827100 Bertrandt
103 2021-07-14 148.10 149.57 147.68 149.15  148.5206 127050800 Bertrandt
104 2021-07-15 149.24 150.00 147.09 148.48  147.8534 106820300 Bertrandt
105 2021-07-16 148.46 149.76 145.88 146.39  145.7722  93251400 Bertrandt
106 2021-07-19 143.75 144.07 141.67 142.45  141.8489 121434600 Bertrandt
107 2021-07-20 143.46 147.10 142.96 146.15  145.5332  96350000 Bertrandt
108 2021-07-21 145.53 146.13 144.63 145.40  144.7864  74993500 Bertrandt
109 2021-07-22 145.94 148.20 145.81 146.80  146.1805  77338200 Bertrandt
110 2021-07-23 147.55 148.72 146.92 148.56  147.9331  71447400 Bertrandt
111 2021-07-26 148.27 149.83 147.70 148.99  148.3613  72434100 Bertrandt
112 2021-07-27 149.12 149.21 145.55 146.77  146.1507 104818600 Bertrandt
113 2021-07-28 144.81 146.97 142.54 144.98  144.3682 118931200 Bertrandt
114 2021-07-29 144.69 146.55 144.58 145.64  145.0254  56699500 Bertrandt
115 2021-07-30 144.38 146.33 144.11 145.86  145.2445  70382000 Bertrandt
116 2021-08-02 146.36 146.95 145.25 145.52  144.9059  62880000 Bertrandt
```

```
117 2021-08-03 145.81 148.04 145.18 147.36  146.7382  64786600 Bertrandt
118 2021-08-04 147.27 147.79 146.28 146.95  146.3299  56368300 Bertrandt
119 2021-08-05 146.98 147.84 146.17 147.06  146.4394  46397700 Bertrandt
120 2021-08-06 146.35 147.11 145.63 146.14  145.7413  54067400 Bertrandt
121 2021-08-09 146.20 146.70 145.52 146.09  145.6915  48908700 Bertrandt
122 2021-08-10 146.44 147.71 145.30 145.60  145.2028  69023100 Bertrandt
123 2021-08-11 146.05 146.72 145.53 145.86  145.4621  48493500 Bertrandt
124 2021-08-12 146.19 149.05 145.84 148.89  148.4838  72282600 Bertrandt
125 2021-08-13 148.97 149.44 148.27 149.10  148.6933  59318800 Bertrandt
 [ reached 'max' / getOption("max.print") -- omitted 637 rows ]
> df_allStocks$Date <- as.character(df_allStocks$Date)
> datesplit_list <- strsplit(df_allStocks$Date, "-")
> df_dates <- ldply(datesplit_list)
> colnames(df_dates) <- c("Year", "Month", "Day")
> df_allStocks <- cbind(df_allStocks, df_dates)
> names(df_allStocks)
 [1] "Date"    "Open"     "High"     "Low"      "Close"    "Adj.Close" "Volume"    "Stock"    "Year"
[10] "Month"    "Day"
> head(df_allStocks)
      Date Open  High   Low Close Adj.Close   Volume     Stock Year Month Day
1 2021-02-17 131.25 132.22 129.47 130.84  130.0669  97918500 Bertrandt 2021   02  17
2 2021-02-18 129.20 130.00 127.41 129.71  128.9436  96856700 Bertrandt 2021   02  18
3 2021-02-19 130.24 130.71 128.80 129.87  129.1027  87668800 Bertrandt 2021   02  19
4 2021-02-22 128.01 129.72 125.60 126.00  125.2555 103916400 Bertrandt 2021   02  22
5 2021-02-23 123.76 126.71 118.39 125.86  125.1164 158273000 Bertrandt 2021   02  23
6 2021-02-24 124.94 125.56 122.23 125.35  124.6094 111039900 Bertrandt 2021   02  24
> g <- ggplot(data=df_aapl, aes(x=Date, y=Open, group=1))  # group 1st param
> g <- g + geom_line(linetype="dashed")
> g
> g <- ggplot(data=df_aapl, aes(x=Date, y=Open, group=1))  # group 1st param
> g <- g + geom_line(linetype="dashed", col="red")
> g
> g <- ggplot(data=df_aapl, aes(x=Date, y=Open, group=1))  # group 1st param
> g <- g + geom_line(linetype="solid", col="red", size=1.5)
> g <- g + labs(title="Apple Inc", subtitle="Open Prices", y="Open", x="Year", caption="Yearwise Apple
Stock")
> g
> options(scipen = 999)
> ggplot(data=df_allStocks, aes(x=Stock, y=Volume)) +
+   geom_bar(stat="identity")   #if we want heights of the bars to represent values in the data, map a
value to y aes
> #scipen - avoid scientific notations by giving largest limit eg. 999
> ggplot(data=df_allStocks, aes(x=Stock, y=Volume)) +
+   geom_bar(stat="identity") + coord_flip()   #coord_flip to create horizontal plot
```

```
> ggplot(data=df_allStocks, aes(x=Stock, y=Volume)) +
+   geom_bar(stat="identity", width=0.5)   #change width of bars
> ggplot(data=df_allStocks, aes(x=Stock, y=Volume)) +
+   geom_bar(stat="identity", width=0.5, col="blue")
> ggplot(data=df_allStocks, aes(x=Stock, y=Volume, fill=Stock)) +
+   geom_bar(stat="identity", width=0.5)
> #fill=Stock - fill colors automatically as per the levels of the bar
> ggplot(df_nike, aes(x=Open)) + geom_histogram()
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
> ggplot(df_waltdisney, aes(x=Open)) + geom_histogram()
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
> ggplot(df_nike, aes(x=Volume)) + geom_histogram(fill="lightblue", color="darkblue")
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
> ggplot(df_nike, aes(x=Close)) + geom_histogram(fill="lightblue", color="darkblue")
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
> ggplot(df_nike, aes(x=Close)) + geom_histogram(fill="lightblue", color="darkblue", binwidth=3)
> ggplot(df_nike, aes(x=Open)) +
+   geom_histogram(aes(y=..density..),fill="white", colour="black") +
+   geom_density(alpha=.2, fill="Turquoise")   #alpha controls the transparency
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
> ggplot(df_nike, aes(x=Open, col=Stock)) + geom_histogram(fill="light blue", binwidth=3)
> ggplot(df_allStocks, aes(x=Open, col=Stock)) + geom_histogram(fill="light blue", binwidth=3)
#Different outline color for different stock category
> ggplot(df_waltdisney, aes(x=Open, y=Close)) + geom_point()
> ggplot(df_nike, aes(x=Open, y=Close)) + geom_point(size=2, shape=23) + geom_smooth(method="lm")
`geom_smooth()` using formula 'y ~ x'
> #size - size of point, shape - shape of point (0-25), method="lm" - draw linear model (linear regression)
line
> ggplot(df_nike, aes(x=Open, y=Close)) +
+   geom_point(shape=18, color="dark grey") +
+   geom_smooth(method="lm", linetype="dashed", color="red")
> df_midwest = read.csv("http://goo.gl/G1K41K")
> dim(df_midwest)
[1] 437  28
> summary(df_midwest)
     PID         county             state               area          poptotal         popdensity
 Min.   : 561   Length:437        Length:437        Min.   :0.00500   Min.   :   1701   Min.   :   85.05
 1st Qu.: 670   Class :character  Class :character  1st Qu.:0.02400   1st Qu.:  18840   1st Qu.:  622.41
 Median :1221   Mode :character   Mode :character   Median :0.03000   Median :  35324   Median :
1156.21
 Mean   :1437                                       Mean   :0.03317   Mean   :  96130   Mean   : 3097.74
 3rd Qu.:2059                                       3rd Qu.:0.03800   3rd Qu.:  75651   3rd Qu.: 2330.00
 Max.   :3052                                       Max.   :0.11000   Max.   :5105067   Max.   :88018.40
    popwhite         popblack        popamerindian       popasian         popother          percwhite
```

```
 Min.   :    416   Min.   :      0   Min.   :    4.0   Min.   :     0   Min.   :     0   Min.   :10.69
 1st Qu.: 18630   1st Qu.:     29   1st Qu.:   44.0   1st Qu.:    35   1st Qu.:    20   1st Qu.:94.89
 Median : 34471   Median :    201   Median :   94.0   Median :   102   Median :    66   Median :98.03
 Mean   : 81840   Mean   :  11024   Mean   :  343.1   Mean   :  1310   Mean   :  1613   Mean   :95.56
 3rd Qu.: 72968   3rd Qu.:   1291   3rd Qu.:  288.0   3rd Qu.:   401   3rd Qu.:   345   3rd Qu.:99.07
 Max.   :3204947  Max.   :1317147  Max.   :10289.0  Max.   :188565  Max.   :384119  Max.   :99.82
   percblack       percamerindan      percasian        percother        popadults         perchsd
 Min.   : 0.0000  Min.   : 0.05623  Min.   :0.0000   Min.   :0.00000  Min.   :   1287  Min.   :46.91
 1st Qu.: 0.1157  1st Qu.: 0.15793  1st Qu.:0.1737   1st Qu.:0.09102  1st Qu.:  12271  1st Qu.:71.33
 Median : 0.5390  Median : 0.21502  Median :0.2972   Median :0.17844  Median :  22188  Median :74.25
 Mean   : 2.6763  Mean   : 0.79894  Mean   :0.4872   Mean   :0.47906  Mean   :  60973  Mean   :73.97
 3rd Qu.: 2.6014  3rd Qu.: 0.38362  3rd Qu.:0.5212   3rd Qu.:0.48050  3rd Qu.:  47541  3rd Qu.:77.20
 Max.   :40.2100  Max.   :89.17738  Max.   :5.0705   Max.   :7.52427  Max.   :3291995  Max.   :88.90
   percollege       percprof      poppovertyknown   percpovertyknown percbelowpoverty percchildbelowpovert
 Min.   : 7.336  Min.   : 0.5203  Min.   :   1696  Min.   :80.90   Min.   : 2.180  Min.   : 1.919
 1st Qu.:14.114  1st Qu.: 2.9980  1st Qu.:  18364  1st Qu.:96.89   1st Qu.: 9.199  1st Qu.:11.624
 Median :16.798  Median : 3.8142  Median :  33788  Median :98.17   Median :11.822  Median :15.270
 Mean   :18.273  Mean   : 4.4473  Mean   :  93642  Mean   :97.11   Mean   :12.511  Mean   :16.447
 3rd Qu.:20.550  3rd Qu.: 4.9493  3rd Qu.:  72840  3rd Qu.:98.60   3rd Qu.:15.133  3rd Qu.:20.352
 Max.   :48.079  Max.   :20.7913  Max.   :5023523  Max.   :99.86   Max.   :48.691  Max.   :64.308
 percadultpoverty percelderlypoverty   inmetro       category
 Min.   : 1.938  Min.   : 3.547    Min.   :0.0000  Length:437
 1st Qu.: 7.668  1st Qu.: 8.912    1st Qu.:0.0000  Class :character
 Median :10.008  Median :10.869    Median :0.0000  Mode :character
 Mean   :10.919  Mean   :11.389    Mean   :0.3432
 3rd Qu.:13.182  3rd Qu.:13.412    3rd Qu.:1.0000
 Max.   :43.312  Max.   :31.162    Max.   :1.0000
> ggplot(df_midwest, aes(x=area, y=poptotal)) +
+   geom_point(shape=18, color="dark grey") +
+   geom_smooth(method="lm", linetype="dashed", color="red")
`geom_smooth()` using formula 'y ~ x'
> ggplot(df_midwest, aes(x=area, y=poptotal)) + geom_point(shape=18, color="dark
grey")+geom_smooth(method="lm", linetype="dashed", color="red") + coord_cartesian(xlim=c(0,0.1),
ylim=c(0,600000))
`geom_smooth()` using formula 'y ~ x'
> seq(1, 20, 3)
[1]  1  4  7 10 13 16 19
> g <- ggplot(df_midwest, aes(x=area, y=poptotal)) +
+   geom_point(size=2) +
+   geom_smooth(method="lm",col="black") +
+   coord_cartesian(xlim=c(0,0.1), ylim=c(0,1000000)) +
```

```
+   labs(title="Area Vs Population", subtitle = "Using midwest dataset", y="Population", x="area", caption = "Midwest Demographics")
> g + scale_x_continuous(breaks=seq(0, 0.10, 0.01))
`geom_smooth()` using formula 'y ~ x'
> g + scale_y_continuous(breaks=seq(0, 1000000, 50000))
`geom_smooth()` using formula 'y ~ x'
> g <- ggplot(df_midwest, aes(x=area, y=poptotal)) +
+   geom_point(aes(color=state), size=2) +
+   geom_smooth(method="lm",col="black") +
+   coord_cartesian(xlim=c(0,0.1), ylim=c(0,1000000)) +
+   labs(title="Area Vs Population", subtitle = "Using midwest dataset", y="Population", x="area", caption = "Midwest Demographics")
> g + scale_x_continuous(breaks=seq(0, 0.10, 0.01))
`geom_smooth()` using formula 'y ~ x'
> g + scale_y_continuous(breaks=seq(0, 1000000, 50000))
> ggplot(df_allStocks, aes(x=Month, y=Close)) + geom_boxplot()
> ggplot(df_allStocks, aes(x=Month, y=Close)) + geom_boxplot() + coord_flip()
> ggplot(df_allStocks, aes(x=Month, y=Close, color=Month)) + geom_boxplot() + coord_flip()
> ggplot(df_midwest, aes(x=state, y=poptotal)) + geom_boxplot(outlier.color = "red", outlier.shape = 1, outlier.size = 2)
> ggplot(df_allStocks, aes(x=Year, y=Close)) + geom_boxplot() + facet_grid(~ Stock)
> ggplot(df_allStocks, aes(x=Month, y=Close)) + geom_boxplot() + facet_grid(Stock ~ Year)
> ggplot(df_allStocks, aes(x=Open)) +
+   geom_histogram(color="black", fill="white") +
+   facet_grid(Stock ~ .)
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
> ggplot(df_allStocks, aes(x=Open, color=Stock)) +
+   geom_histogram(fill="white") +
+   facet_grid(Stock ~ .)
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
> ggplot(df_allStocks, aes(x=Close, color=Stock)) +
+   geom_histogram(fill="white") +
+   facet_grid(Stock ~ ., scales="free_y")
```

## Day 5 – R Programming
```
> dev.off()
null device
          1
> setwd("C:/zubeda/PGA02_Zubu/R Programming")
> library("plyr")
> library("ggplot2")
> g <- ggplot(df_midwest, aes(x=area, y=poptotal)) +
+   geom_point(shape=18, color="dark grey") +
+   geom_smooth(method="lm", linetype="dashed", col="red") +
```

```
+   coord_cartesian(xlim=c(0, 0.1), ylim=c(0, 600000))
> g <- g + theme_light()
> g
`geom_smooth()` using formula 'y ~ x'
> ggplot(df_waltdisney, aes(x=Open, y=Close)) +
+   geom_point() + theme(panel.grid.major = element_line(size=0.5, linetype="dashed", colour="red"),
panel.background=element_rect(fill="lightblue"))
> ggplot(df_allStocks, aes(x=Stock, y=Volume)) +
+   geom_bar(stat="identity") + theme(panel.grid.major = element_line(size=0.5, linetype="solid",
colour="blue"), panel.background=element_rect(fill="lightblue"))
> library(RColorBrewer)
> head(brewer.pal.info, 12)
```

|         | maxcolors | category |
|---------|-----------|----------|
| BrBG    | 11        | div      |
| PiYG    | 11        | div      |
| PRGn    | 11        | div      |
| PuOr    | 11        | div      |
| RdBu    | 11        | div      |
| RdGy    | 11        | div      |
| RdYlBu  | 11        | div      |
| RdYlGn  | 11        | div      |
| Spectral| 11        | div      |
| Accent  | 8         | qual     |
| Dark2   | 8         | qual     |
| Paired  | 12        | qual     |

|         | colorblind |
|---------|------------|
| BrBG    | TRUE       |
| PiYG    | TRUE       |
| PRGn    | TRUE       |
| PuOr    | TRUE       |
| RdBu    | TRUE       |
| RdGy    | FALSE      |
| RdYlBu  | TRUE       |
| RdYlGn  | FALSE      |
| Spectral| FALSE      |
| Accent  | FALSE      |
| Dark2   | TRUE       |
| Paired  | TRUE       |

```
> display.brewer.all()
> g <- ggplot(df_midwest, aes(x=area, y=poptotal)) +
+   geom_point(aes(color=state), size=2) +
+   geom_smooth(method="lm",col="black") +
+   coord_cartesian(xlim=c(0, 0.1), ylim=c(0, 1000000)) +
```

```
+  labs(title="Area Vs Population", subtitle = "Using midwest dataset", y="Population", x="area", caption
= "Midwest Demographics")
> g <- g + scale_colour_brewer(palette="Dark2")
> g
`geom_smooth()` using formula 'y ~ x'
> g <- ggplot(df_midwest, aes(x=area, y=poptotal)) +
+  geom_point(aes(color=state), size=2) +
+  geom_smooth(method="lm",col="black") +
+  coord_cartesian(xlim=c(0, 0.1), ylim=c(0, 1000000)) +
+  labs(title="Area Vs Population", subtitle = "Using midwest dataset", y="Population", x="area", caption
= "Midwest Demographics")
> library(grid)
> annotate_text <- "Showing population by area with best fit regression line"
> g
`geom_smooth()` using formula 'y ~ x'
> annotatechart <- grid.text(annotate_text, x=0.5, y=0.9, gp=gpar(col="darkred", fontsize=9,
fontface="plain"))
```

**Class Assessment**
```
> setwd("C:/zubeda/PGA02_Zubu/R Programming")
> #Q1)
> #II. Create a vector of length 4 using seq() function and showcase how to access the elements using
numeric indexes, logical indexes and character indexes.
> v <- seq(11, 15, length.out=4)   #returns 4 numbers, including 1st, last and middle numbers averaged if
numbers are more then limit
> v
[1] 11.00000 12.33333 13.66667 15.00000
> v[1]
[1] 11
> v[3]
[1] 13.66667
> v[c(2, 4)]
[1] 12.33333 15.00000
> v[c(TRUE, FALSE, TRUE, FALSE)]
[1] 11.00000 13.66667
> names(v) <- c("el1", "el2", "el3", "el4")
> v
    el1     el2     el3     el4
11.00000 12.33333 13.66667 15.00000
> v["el1"]
el1
 11
> y <- c("Mumbai"=400, "Delhi"=100, "Chennai"=300, "Kolkata"=200)
> y
```

```
 Mumbai   Delhi Chennai Kolkata
   400    100   300   200
> y["Chennai"]
Chennai
   300
> y["Mumbai"]
Mumbai
   400
>
> #I.     Load the in-built dataset called trees, that consists of measurements of the girth, height, and
volume of 31 black cherry trees and display rows where height is greater than 82
> ?trees
> trees
  Girth Height Volume
1   8.3    70  10.3
2   8.6    65  10.3
3   8.8    63  10.2
4  10.5    72  16.4
5  10.7    81  18.8
6  10.8    83  19.7
7  11.0    66  15.6
8  11.0    75  18.2
9  11.1    80  22.6
10 11.2    75  19.9
11 11.3    79  24.2
12 11.4    76  21.0
13 11.4    76  21.4
14 11.7    69  21.3
15 12.0    75  19.1
16 12.9    74  22.2
17 12.9    85  33.8
18 13.3    86  27.4
19 13.7    71  25.7
20 13.8    64  24.9
21 14.0    78  34.5
22 14.2    80  31.7
23 14.5    74  36.3
24 16.0    72  38.3
25 16.3    77  42.6
26 17.3    81  55.4
27 17.5    82  55.7
28 17.9    80  58.3
29 18.0    80  51.5
30 18.0    80  51.0
```

```
31 20.6   87 77.0
> dim(trees)
[1] 31 3
> nrow(trees)
[1] 31
> ncol(trees)
[1] 3
> summary(trees)
    Girth         Height      Volume
 Min.   : 8.30  Min.   :63  Min.   :10.20
 1st Qu.:11.05  1st Qu.:72  1st Qu.:19.40
 Median :12.90  Median :76  Median :24.20
 Mean   :13.25  Mean   :76  Mean   :30.17
 3rd Qu.:15.25  3rd Qu.:80  3rd Qu.:37.30
 Max.   :20.60  Max.   :87  Max.   :77.00
> names(trees)
[1] "Girth"  "Height" "Volume"
> str(trees)
'data.frame':    31 obs. of  3 variables:
 $ Girth : num  8.3 8.6 8.8 10.5 10.7 10.8 11 11 11.1 11.2 ...
 $ Height: num  70 65 63 72 81 83 66 75 80 75 ...
 $ Volume: num  10.3 10.3 10.2 16.4 18.8 19.7 15.6 18.2 22.6 19.9 ...
> trees[trees$Height > 82,]
   Girth Height Volume
6   10.8     83   19.7
17  12.9     85   33.8
18  13.3     86   27.4
31  20.6     87   77.0
>
> #Q2) For the 'StudentsPerformance' dataset, perform the following tasks:
> #I.    Analyze the student's performance in exams and write your own observations about the
students and plot the results
> #II.   Create a function to remove outliers using the IQR method
>
> #Function definition such that outliers of passed columns are removed
> students <- read.csv("StudentsPerformance.csv")
> #Get Dimensions
> nrow(students)
[1] 1000
> ncol(students)
[1] 8
> #Get data types
> str(students)
'data.frame':    1000 obs. of  8 variables:
```

```
 $ gender                 : chr  "female" "female" "female" "male" ...
 $ race.ethnicity         : chr  "group B" "group C" "group B" "group A" ...
 $ parental.level.of.education: chr  "bachelor's degree" "some college" "master's degree" "associate's
degree" ...
 $ lunch                  : chr  "standard" "standard" "standard" "free/reduced" ...
 $ test.preparation.course   : chr  "none" "completed" "none" "none" ...
 $ math.score             : int  72 69 90 47 76 71 88 40 64 38 ...
 $ reading.score          : int  72 90 95 57 78 83 95 43 64 60 ...
 $ writing.score          : int  74 88 93 44 75 78 92 39 67 50 ...
> #rename column names with new column names
> namesOfColumns <- c("Gender", "Race", "Parent_Education", "Lunch", "Test_Prep", "Math_Score",
"Reading_Score", "Writing_Score")
> colnames(students) <- namesOfColumns
> colnames(students)
[1] "Gender"        "Race"         "Parent_Education" "Lunch"        "Test_Prep"      "Math_Score"
"Reading_Score"
[8] "Writing_Score"
> summary(students)  #Summary statistics of numeric variable
   Gender        Race       Parent_Education   Lunch        Test_Prep       Math_Score
Reading_Score
 Length:1000     Length:1000     Length:1000      Length:1000     Length:1000      Min.  : 0.00  Min.
: 17.00
 Class :character  Class :character  Class :character  Class :character  Class :character  1st Qu.: 57.00
1st Qu.: 59.00
 Mode :character  Mode :character  Mode :character  Mode :character  Mode :character  Median :
66.00  Median : 70.00
                                                    Mean  : 66.09  Mean  : 69.17
                                                    3rd Qu.: 77.00  3rd Qu.: 79.00
                                                    Max.  :100.00  Max.  :100.00

 Writing_Score
 Min.  : 10.00
 1st Qu.: 57.75
 Median : 69.00
 Mean  : 68.05
 3rd Qu.: 79.00
 Max.  :100.00
>
> #Obervations
> #1. There are more females than males
> #2. Group C has the largest number of members
> #3. some college and associates degree are the most frequently occuring #parental levels of education
> #4. most students have a standard lunch
> #5. most students have not completed the test prep course
> #6. the scores for math, reading and writing are on the same scale 0-100
```

```
>
> remove_outliers <- function(x, na.rm=TRUE, ...) {
+   qnt <- quantile(x, probs=c(.25, .75), na.rm=na.rm, ...)
+   H <- 1.5 * IQR(x, na.rm = na.rm)
+   y <- x
+   y[x < (qnt[1] - H)] <- NA
+   y[x > (qnt[2] + H)] <- NA
+   y
+ }
> #Combine columns categorical cols as it is, and last 3 cols with outliers removed
> performance_data <- cbind(students[1:5], apply(students[6], 2, remove_outliers), apply(students[7], 2,
remove_outliers), apply(students[8], 2, remove_outliers))
> performance_data
```

| | Gender | Race | Parent_Education | Lunch | Test_Prep | Math_Score | Reading_Score | Writing_Score |
|---|---|---|---|---|---|---|---|---|
| 1 | female | group B | bachelor's degree | standard | none | 72 | 72 | 74 |
| 2 | female | group C | some college | standard | completed | 69 | 90 | 88 |
| 3 | female | group B | master's degree | standard | none | 90 | 95 | 93 |
| 4 | male | group A | associate's degree | free/reduced | none | 47 | 57 | 44 |
| 5 | male | group C | some college | standard | none | 76 | 78 | 75 |
| 6 | female | group B | associate's degree | standard | none | 71 | 83 | 78 |
| 7 | female | group B | some college | standard | completed | 88 | 95 | 92 |
| 8 | male | group B | some college | free/reduced | none | 40 | 43 | 39 |
| 9 | male | group D | high school | free/reduced | completed | 64 | 64 | 67 |
| 10 | female | group B | high school | free/reduced | none | 38 | 60 | 50 |
| 11 | male | group C | associate's degree | standard | none | 58 | 54 | 52 |
| 12 | male | group D | associate's degree | standard | none | 40 | 52 | 43 |
| 13 | female | group B | high school | standard | none | 65 | 81 | 73 |
| 14 | male | group A | some college | standard | completed | 78 | 72 | 70 |
| 15 | female | group A | master's degree | standard | none | 50 | 53 | 58 |
| 16 | female | group C | some high school | standard | none | 69 | 75 | 78 |
| 17 | male | group C | high school | standard | none | 88 | 89 | 86 |
| 18 | female | group B | some high school | free/reduced | none | NA | 32 | 28 |
| 19 | male | group C | master's degree | free/reduced | completed | 46 | 42 | 46 |
| 20 | female | group C | associate's degree | free/reduced | none | 54 | 58 | 61 |
| 21 | male | group D | high school | standard | none | 66 | 69 | 63 |
| 22 | female | group B | some college | free/reduced | completed | 65 | 75 | 70 |
| 23 | male | group D | some college | standard | none | 44 | 54 | 53 |
| 24 | female | group C | some high school | standard | none | 69 | 73 | 73 |
| 25 | male | group D | bachelor's degree | free/reduced | completed | 74 | 71 | 80 |
| 26 | male | group A | master's degree | free/reduced | none | 73 | 74 | 72 |
| 27 | male | group B | some college | standard | none | 69 | 54 | 55 |
| 28 | female | group C | bachelor's degree | standard | none | 67 | 69 | 75 |
| 29 | male | group C | high school | standard | none | 70 | 70 | 65 |
| 30 | female | group D | master's degree | standard | none | 62 | 70 | 75 |

| # | gender | group | parental education | lunch | test prep | math | reading | writing |
|---|--------|-------|--------------------|-------|-----------|------|---------|---------|
| 31 | female | group D | some college | standard | none | 69 | 74 | 74 |
| 32 | female | group B | some college | standard | none | 63 | 65 | 61 |
| 33 | female | group E | master's degree | free/reduced | none | 56 | 72 | 65 |
| 34 | male | group D | some college | standard | none | 40 | 42 | 38 |
| 35 | male | group E | some college | standard | none | 97 | 87 | 82 |
| 36 | male | group E | associate's degree | standard | completed | 81 | 81 | 79 |
| 37 | female | group D | associate's degree | standard | none | 74 | 81 | 83 |
| 38 | female | group D | some high school | free/reduced | none | 50 | 64 | 59 |
| 39 | female | group D | associate's degree | free/reduced | completed | 75 | 90 | 88 |
| 40 | male | group B | associate's degree | free/reduced | none | 57 | 56 | 57 |
| 41 | male | group C | associate's degree | free/reduced | none | 55 | 61 | 54 |
| 42 | female | group C | associate's degree | standard | none | 58 | 73 | 68 |
| 43 | female | group B | associate's degree | standard | none | 53 | 58 | 65 |
| 44 | male | group B | some college | free/reduced | completed | 59 | 65 | 66 |
| 45 | female | group E | associate's degree | free/reduced | none | 50 | 56 | 54 |
| 46 | male | group B | associate's degree | standard | none | 65 | 54 | 57 |
| 47 | female | group A | associate's degree | standard | completed | 55 | 65 | 62 |
| 48 | female | group C | high school | standard | none | 66 | 71 | 76 |
| 49 | female | group D | associate's degree | free/reduced | completed | 57 | 74 | 76 |
| 50 | male | group C | high school | standard | completed | 82 | 84 | 82 |
| 51 | male | group E | some college | standard | none | 53 | 55 | 48 |
| 52 | male | group E | associate's degree | free/reduced | completed | 77 | 69 | 68 |
| 53 | male | group C | some college | standard | none | 53 | 44 | 42 |
| 54 | male | group D | high school | standard | none | 88 | 78 | 75 |
| 55 | female | group C | some high school | free/reduced | completed | 71 | 84 | 87 |
| 56 | female | group C | high school | free/reduced | none | 33 | 41 | 43 |
| 57 | female | group E | associate's degree | standard | completed | 82 | 85 | 86 |
| 58 | male | group D | associate's degree | standard | none | 52 | 55 | 49 |
| 59 | male | group D | some college | standard | completed | 58 | 59 | 58 |
| 60 | female | group C | some high school | free/reduced | none | NA | NA | NA |
| 61 | male | group E | bachelor's degree | free/reduced | completed | 79 | 74 | 72 |
| 62 | male | group A | some high school | free/reduced | none | 39 | 39 | 34 |
| 63 | male | group A | associate's degree | free/reduced | none | 62 | 61 | 55 |
| 64 | female | group C | associate's degree | standard | none | 69 | 80 | 71 |
| 65 | female | group D | some high school | standard | none | 59 | 58 | 59 |
| 66 | male | group B | some high school | standard | none | 67 | 64 | 61 |
| 67 | male | group D | some high school | free/reduced | none | 45 | 37 | 37 |
| 68 | female | group C | some college | standard | none | 60 | 72 | 74 |
| 69 | male | group B | associate's degree | free/reduced | none | 61 | 58 | 56 |
| 70 | female | group C | associate's degree | standard | none | 39 | 64 | 57 |
| 71 | female | group D | some college | free/reduced | completed | 58 | 63 | 73 |
| 72 | male | group D | some college | standard | completed | 63 | 55 | 63 |
| 73 | female | group A | associate's degree | free/reduced | none | 41 | 51 | 48 |
| 74 | male | group C | some high school | free/reduced | none | 61 | 57 | 56 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 75 | male | group C | some high school | standard | none | 49 | 49 | 41 | |
| 76 | male | group B | associate's degree | free/reduced | none | 44 | 41 | 38 | |
| 77 | male | group E | some high school | standard | none | 30 | NA | NA | |
| 78 | male | group A | bachelor's degree | standard | completed | 80 | 78 | 81 | |
| 79 | female | group D | some high school | standard | completed | 61 | 74 | 72 | |
| 80 | female | group E | master's degree | standard | none | 62 | 68 | 68 | |
| 81 | female | group B | associate's degree | standard | none | 47 | 49 | 50 | |
| 82 | male | group B | high school | free/reduced | none | 49 | 45 | 45 | |
| 83 | male | group A | some college | free/reduced | completed | 50 | 47 | 54 | |
| 84 | male | group E | associate's degree | standard | none | 72 | 64 | 63 | |
| 85 | male | group D | high school | free/reduced | none | 42 | 39 | 34 | |
| 86 | female | group C | some college | standard | none | 73 | 80 | 82 | |
| 87 | female | group C | some college | free/reduced | none | 76 | 83 | 88 | |
| 88 | female | group D | associate's degree | standard | none | 71 | 71 | 74 | |
| 89 | female | group A | some college | standard | none | 58 | 70 | 67 | |
| 90 | female | group D | some high school | standard | none | 73 | 86 | 82 | |
| 91 | female | group C | bachelor's degree | standard | none | 65 | 72 | 74 | |
| 92 | male | group C | high school | free/reduced | none | 27 | 34 | 36 | |
| 93 | male | group C | high school | standard | none | 71 | 79 | 71 | |
| 94 | male | group C | associate's degree | free/reduced | completed | 43 | 45 | 50 | |
| 95 | female | group B | some college | standard | none | 79 | 86 | 92 | |
| 96 | male | group C | associate's degree | free/reduced | completed | 78 | 81 | 82 | |
| 97 | male | group B | some high school | standard | completed | 65 | 66 | 62 | |
| 98 | female | group E | some college | standard | completed | 63 | 72 | 70 | |
| 99 | female | group D | some college | free/reduced | none | 58 | 67 | 62 | |
| 100 | female | group D | bachelor's degree | standard | none | 65 | 67 | 62 | |
| 101 | male | group B | some college | standard | none | 79 | 67 | 67 | |
| 102 | male | group D | bachelor's degree | standard | completed | 68 | 74 | 74 | |
| 103 | female | group D | associate's degree | standard | none | 85 | 91 | 89 | |
| 104 | male | group B | high school | standard | completed | 60 | 44 | 47 | |
| 105 | male | group C | some college | standard | completed | 98 | 86 | 90 | |
| 106 | female | group C | some college | standard | none | 58 | 67 | 72 | |
| 107 | female | group D | master's degree | standard | none | 87 | 100 | 100 | |
| 108 | male | group E | associate's degree | standard | completed | 66 | 63 | 64 | |
| 109 | female | group B | associate's degree | free/reduced | none | 52 | 76 | 70 | |
| 110 | female | group B | some high school | standard | none | 70 | 64 | 72 | |
| 111 | female | group D | associate's degree | free/reduced | completed | 77 | 89 | 98 | |
| 112 | male | group C | high school | standard | none | 62 | 55 | 49 | |
| 113 | male | group A | associate's degree | standard | none | 54 | 53 | 47 | |
| 114 | female | group D | some college | standard | none | 51 | 58 | 54 | |
| 115 | female | group E | bachelor's degree | standard | completed | 99 | 100 | 100 | |
| 116 | male | group C | high school | standard | none | 84 | 77 | 74 | |
| 117 | female | group B | bachelor's degree | free/reduced | none | 75 | 85 | 82 | |
| 118 | female | group D | bachelor's degree | standard | none | 78 | 82 | 79 | |

```
119 female group D   some high school    standard    none      51        63        61
120 female group C    some college    standard    none    55        69        65
121 female group C  bachelor's degree    standard completed      79        92        89
122   male group B associate's degree    standard completed      91        89        92
123 female group C    some college    standard completed      88        93        93
124   male group D     high school free/reduced    none      63        57        56
125   male group E    some college    standard    none    83        80        73
 [ reached 'max' / getOption("max.print") -- omitted 875 rows ]
> dim(performance_data)
[1] 1000    8
> sum(is.na(performance_data))   # Sum of null values
[1] 19
> performance_1 <- na.omit(performance_data)
> performance_1
   Gender    Race    Parent_Education       Lunch Test_Prep Math_Score Reading_Score Writing_Score
1  female group B  bachelor's degree    standard    none      72        72        74
2  female group C    some college    standard completed      69        90        88
3  female group B   master's degree    standard    none      90        95        93
4    male group A associate's degree free/reduced    none      47        57        44
5    male group C    some college    standard    none    76        78        75
6  female group B associate's degree    standard    none      71        83        78
7  female group B    some college    standard completed      88        95        92
8    male group B    some college free/reduced    none      40        43        39
9    male group D     high school free/reduced completed      64        64        67
10 female group B     high school free/reduced    none      38        60        50
11   male group C associate's degree    standard    none      58        54        52
12   male group D associate's degree    standard    none      40        52        43
13 female group B     high school    standard    none    65        81        73
14   male group A    some college    standard completed      78        72        70
15 female group A   master's degree    standard    none      50        53        58
16 female group C  some high school    standard    none      69        75        78
17   male group C     high school    standard    none    88        89        86
19   male group C   master's degree free/reduced completed      46        42        46
20 female group C associate's degree free/reduced    none      54        58        61
21   male group D     high school    standard    none    66        69        63
22 female group B    some college free/reduced completed      65        75        70
23   male group D    some college    standard    none    44        54        53
24 female group C  some high school    standard    none    69        73        73
25   male group D  bachelor's degree free/reduced completed      74        71        80
26   male group A   master's degree free/reduced    none      73        74        72
27   male group B    some college    standard    none    69        54        55
28 female group C  bachelor's degree    standard    none      67        69        75
29   male group C     high school    standard    none    70        70        65
30 female group D   master's degree    standard    none      62        70        75
```

| # | gender | group | parent education | lunch | test prep | | | |
|---|---|---|---|---|---|---|---|---|
| 31 | female | group D | some college | standard | none | 69 | 74 | 74 |
| 32 | female | group B | some college | standard | none | 63 | 65 | 61 |
| 33 | female | group E | master's degree | free/reduced | none | 56 | 72 | 65 |
| 34 | male | group D | some college | standard | none | 40 | 42 | 38 |
| 35 | male | group E | some college | standard | none | 97 | 87 | 82 |
| 36 | male | group E | associate's degree | standard | completed | 81 | 81 | 79 |
| 37 | female | group D | associate's degree | standard | none | 74 | 81 | 83 |
| 38 | female | group D | some high school | free/reduced | none | 50 | 64 | 59 |
| 39 | female | group D | associate's degree | free/reduced | completed | 75 | 90 | 88 |
| 40 | male | group B | associate's degree | free/reduced | none | 57 | 56 | 57 |
| 41 | male | group C | associate's degree | free/reduced | none | 55 | 61 | 54 |
| 42 | female | group C | associate's degree | standard | none | 58 | 73 | 68 |
| 43 | female | group B | associate's degree | standard | none | 53 | 58 | 65 |
| 44 | male | group B | some college | free/reduced | completed | 59 | 65 | 66 |
| 45 | female | group E | associate's degree | free/reduced | none | 50 | 56 | 54 |
| 46 | male | group B | associate's degree | standard | none | 65 | 54 | 57 |
| 47 | female | group A | associate's degree | standard | completed | 55 | 65 | 62 |
| 48 | female | group C | high school | standard | none | 66 | 71 | 76 |
| 49 | female | group D | associate's degree | free/reduced | completed | 57 | 74 | 76 |
| 50 | male | group C | high school | standard | completed | 82 | 84 | 82 |
| 51 | male | group E | some college | standard | none | 53 | 55 | 48 |
| 52 | male | group E | associate's degree | free/reduced | completed | 77 | 69 | 68 |
| 53 | male | group C | some college | standard | none | 53 | 44 | 42 |
| 54 | male | group D | high school | standard | none | 88 | 78 | 75 |
| 55 | female | group C | some high school | free/reduced | completed | 71 | 84 | 87 |
| 56 | female | group C | high school | free/reduced | none | 33 | 41 | 43 |
| 57 | female | group E | associate's degree | standard | completed | 82 | 85 | 86 |
| 58 | male | group D | associate's degree | standard | none | 52 | 55 | 49 |
| 59 | male | group D | some college | standard | completed | 58 | 59 | 58 |
| 61 | male | group E | bachelor's degree | free/reduced | completed | 79 | 74 | 72 |
| 62 | male | group A | some high school | free/reduced | none | 39 | 39 | 34 |
| 63 | male | group A | associate's degree | free/reduced | none | 62 | 61 | 55 |
| 64 | female | group C | associate's degree | standard | none | 69 | 80 | 71 |
| 65 | female | group D | some high school | standard | none | 59 | 58 | 59 |
| 66 | male | group B | some high school | standard | none | 67 | 64 | 61 |
| 67 | male | group D | some high school | free/reduced | none | 45 | 37 | 37 |
| 68 | female | group C | some college | standard | none | 60 | 72 | 74 |
| 69 | male | group B | associate's degree | free/reduced | none | 61 | 58 | 56 |
| 70 | female | group C | associate's degree | standard | none | 39 | 64 | 57 |
| 71 | female | group D | some college | free/reduced | completed | 58 | 63 | 73 |
| 72 | male | group D | some college | standard | completed | 63 | 55 | 63 |
| 73 | female | group A | associate's degree | free/reduced | none | 41 | 51 | 48 |
| 74 | male | group C | some high school | free/reduced | none | 61 | 57 | 56 |
| 75 | male | group C | some high school | standard | none | 49 | 49 | 41 |

| | gender | group | parental level of education | lunch | test prep | | | |
|---|---|---|---|---|---|---|---|---|
| 76 | male | group B | associate's degree | free/reduced | none | 44 | 41 | 38 |
| 78 | male | group A | bachelor's degree | standard | completed | 80 | 78 | 81 |
| 79 | female | group D | some high school | standard | completed | 61 | 74 | 72 |
| 80 | female | group E | master's degree | standard | none | 62 | 68 | 68 |
| 81 | female | group B | associate's degree | standard | none | 47 | 49 | 50 |
| 82 | male | group B | high school | free/reduced | none | 49 | 45 | 45 |
| 83 | male | group A | some college | free/reduced | completed | 50 | 47 | 54 |
| 84 | male | group E | associate's degree | standard | none | 72 | 64 | 63 |
| 85 | male | group D | high school | free/reduced | none | 42 | 39 | 34 |
| 86 | female | group C | some college | standard | none | 73 | 80 | 82 |
| 87 | female | group C | some college | free/reduced | none | 76 | 83 | 88 |
| 88 | female | group D | associate's degree | standard | none | 71 | 71 | 74 |
| 89 | female | group A | some college | standard | none | 58 | 70 | 67 |
| 90 | female | group D | some high school | standard | none | 73 | 86 | 82 |
| 91 | female | group C | bachelor's degree | standard | none | 65 | 72 | 74 |
| 92 | male | group C | high school | free/reduced | none | 27 | 34 | 36 |
| 93 | male | group C | high school | standard | none | 71 | 79 | 71 |
| 94 | male | group C | associate's degree | free/reduced | completed | 43 | 45 | 50 |
| 95 | female | group B | some college | standard | none | 79 | 86 | 92 |
| 96 | male | group C | associate's degree | free/reduced | completed | 78 | 81 | 82 |
| 97 | male | group B | some high school | standard | completed | 65 | 66 | 62 |
| 98 | female | group E | some college | standard | completed | 63 | 72 | 70 |
| 99 | female | group D | some college | free/reduced | none | 58 | 67 | 62 |
| 100 | female | group D | bachelor's degree | standard | none | 65 | 67 | 62 |
| 101 | male | group B | some college | standard | none | 79 | 67 | 67 |
| 102 | male | group D | bachelor's degree | standard | completed | 68 | 74 | 74 |
| 103 | female | group D | associate's degree | standard | none | 85 | 91 | 89 |
| 104 | male | group B | high school | standard | completed | 60 | 44 | 47 |
| 105 | male | group C | some college | standard | completed | 98 | 86 | 90 |
| 106 | female | group C | some college | standard | none | 58 | 67 | 72 |
| 107 | female | group D | master's degree | standard | none | 87 | 100 | 100 |
| 108 | male | group E | associate's degree | standard | completed | 66 | 63 | 64 |
| 109 | female | group B | associate's degree | free/reduced | none | 52 | 76 | 70 |
| 110 | female | group B | some high school | standard | none | 70 | 64 | 72 |
| 111 | female | group D | associate's degree | free/reduced | completed | 77 | 89 | 98 |
| 112 | male | group C | high school | standard | none | 62 | 55 | 49 |
| 113 | male | group A | associate's degree | standard | none | 54 | 53 | 47 |
| 114 | female | group D | some college | standard | none | 51 | 58 | 54 |
| 115 | female | group E | bachelor's degree | standard | completed | 99 | 100 | 100 |
| 116 | male | group C | high school | standard | none | 84 | 77 | 74 |
| 117 | female | group B | bachelor's degree | free/reduced | none | 75 | 85 | 82 |
| 118 | female | group D | bachelor's degree | standard | none | 78 | 82 | 79 |
| 119 | female | group D | some high school | standard | none | 51 | 63 | 61 |
| 120 | female | group C | some college | standard | none | 55 | 69 | 65 |

```
121 female group C  bachelor's degree    standard completed      79        92        89
122  male group B associate's degree     standard completed      91        89        92
123 female group C     some college    standard completed       88        93        93
124  male group D     high school free/reduced    none     63        57        56
125  male group E     some college    standard    none     83        80        73
126 female group B     high school    standard    none     87        95        86
127  male group B  some high school    standard    none     72        68        67
128  male group D     some college    standard completed      65        77        74
 [ reached 'max' / getOption("max.print") -- omitted 863 rows ]
> nrow(performance_1)
[1] 988
> library(ggplot2)
> Data <- performance_1
>
> plot1 <-
+   ggplot() +
+   geom_bar(data = Data, aes(x = Gender), width = 0.2, fill = "green") +
+   geom_text(stat='count', data = Data, aes(x = Gender, label=..count..), vjust=-0.2) +
+   theme_bw() +
+   xlab("Gender") +
+   ylab("Number of Students") +
+   theme_classic() +
+   ggtitle("Number of Students by Gender") +
+   scale_fill_brewer(type = "qual", palette = 1, direction = 1,
+                aesthetics = "fill") +
+   ylim(0, 600)
>
> plot1
>
> #There are more 510 female students and 478 male students.
>
> #Students By race:
> plot2 <- ggplot() +
+   geom_bar(data = Data, aes(x = Race), width = 0.6, fill = "green") +
+   geom_text(data = Data, aes(x = Race, label = ..count..), stat = "count", vjust = -0.2) +
+   theme_bw() +
+   xlab("Race/Ethnicity") +
+   ylab("Number of Students") +
+   theme(
+     text = element_text(family = "Tahoma")
+   ) +
+   theme_classic()+
+   scale_fill_brewer(type = "qual", palette = 1, direction = 1,
+                aesthetics = "fill") +
```

```
+   ggtitle("Number of Students by Race/Ethnicity")
> plot2
> #There are 316 students in group C, 261 students in group D while there are only 88 students in group
A.
>
> #Plot scores by Gender to determine if there is a different score tendency for each gender
> # Math scores by Gender plot
> p <- ggplot(students, aes(Math_Score)) + geom_histogram(binwidth=5, color="gray", aes(fill=Gender))
> p <- p + xlab("Math Scores") + ylab("Gender") + ggtitle("Math Scores by Gender")
> p
>
> # Boxplot of scores and Test Prep by Gender
> b <- ggplot(students, aes(Gender, Writing_Score, color = Test_Prep))
> b <- b + geom_boxplot()
> b <- b + ggtitle("Writing scores by Gender Boxplot")
> b <- b + xlab("Gender") + ylab("Writing Scores")
> b
>
> # Reading scores by Gender plot
> p1 <- ggplot(students, aes(Reading_Score)) + geom_histogram(binwidth=5, color="gray",
aes(fill=Gender))
> p1 <- p1 + xlab("Reading Scores") + ylab("Gender") + ggtitle("Reading Scores by Gender")
> p1
>
> b1 <- ggplot(students, aes(Gender, Math_Score, color = Test_Prep))
> b1 <- b1 + geom_boxplot()
> b1 <- b1 + ggtitle("Math scores by Gender Boxplot")
> b1 <- b1 + xlab("Gender") + ylab("Math Scores")
> b1
>
> # Writing scores by Gender plot
> p2 <- ggplot(students, aes(Writing_Score)) +  geom_histogram(binwidth=5, color="gray",
aes(fill=Gender))
> p2 <- p2 + xlab("Writing Scores") + ylab("Gender") + ggtitle("Writing Scores by Gender")
> p2
>
> b2 <- ggplot(students, aes(Gender, Reading_Score, color = Test_Prep))
> b2 <- b2 + geom_boxplot()
> b2 <- b2 + ggtitle("Reading scores by Gender Boxplot")
> b2 <- b2 + xlab("Gender") + ylab("Reading Scores")
> b2
>
> #Conclusion :
>
```

```
> #1. students who completed the prep class had better scores in all three tests.
> #2. male students have received better scores in Math while female students in reading and writing.
>
> #Which gender does better in tests
> # To find out the result, we need to create a columns that stores average of score
> performance_2 <- performance_1
> performance_2$Total_score = performance_2$Math_Score + performance_2$Reading_Score
+performance_2$Writing_Score
> performance_2$Avg_score = round((performance_2$Total_score)/3,0)
> performance_2
```

| | Gender | Race | Parent_Education | Lunch | Test_Prep | Math_Score | Reading_Score | Writing_Score | Total_score | Avg_score |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | female | group B | bachelor's degree | standard | none | 72 | 72 | 74 | 218 | 73 |
| 2 | female | group C | some college | standard | completed | 69 | 90 | 88 | 247 | 82 |
| 3 | female | group B | master's degree | standard | none | 90 | 95 | 93 | 278 | 93 |
| 4 | male | group A | associate's degree | free/reduced | none | 47 | 57 | 44 | 148 | 49 |
| 5 | male | group C | some college | standard | none | 76 | 78 | 75 | 229 | 76 |
| 6 | female | group B | associate's degree | standard | none | 71 | 83 | 78 | 232 | 77 |
| 7 | female | group B | some college | standard | completed | 88 | 95 | 92 | 275 | 92 |
| 8 | male | group B | some college | free/reduced | none | 40 | 43 | 39 | 122 | 41 |
| 9 | male | group D | high school | free/reduced | completed | 64 | 64 | 67 | 195 | 65 |
| 10 | female | group B | high school | free/reduced | none | 38 | 60 | 50 | 148 | 49 |
| 11 | male | group C | associate's degree | standard | none | 58 | 54 | 52 | 164 | 55 |
| 12 | male | group D | associate's degree | standard | none | 40 | 52 | 43 | 135 | 45 |
| 13 | female | group B | high school | standard | none | 65 | 81 | 73 | 219 | 73 |
| 14 | male | group A | some college | standard | completed | 78 | 72 | 70 | 220 | 73 |
| 15 | female | group A | master's degree | standard | none | 50 | 53 | 58 | 161 | 54 |
| 16 | female | group C | some high school | standard | none | 69 | 75 | 78 | 222 | 74 |
| 17 | male | group C | high school | standard | none | 88 | 89 | 86 | 263 | 88 |
| 19 | male | group C | master's degree | free/reduced | completed | 46 | 42 | 46 | 134 | 45 |
| 20 | female | group C | associate's degree | free/reduced | none | 54 | 58 | 61 | 173 | 58 |
| 21 | male | group D | high school | standard | none | 66 | 69 | 63 | 198 | 66 |
| 22 | female | group B | some college | free/reduced | completed | 65 | 75 | 70 | 210 | 70 |
| 23 | male | group D | some college | standard | none | 44 | 54 | 53 | 151 | 50 |
| 24 | female | group C | some high school | standard | none | 69 | 73 | 73 | 215 | 72 |
| 25 | male | group D | bachelor's degree | free/reduced | completed | 74 | 71 | 80 | 225 | 75 |
| 26 | male | group A | master's degree | free/reduced | none | 73 | 74 | 72 | 219 | 73 |
| 27 | male | group B | some college | standard | none | 69 | 54 | 55 | 178 | 59 |
| 28 | female | group C | bachelor's degree | standard | none | 67 | 69 | 75 | 211 | 70 |
| 29 | male | group C | high school | standard | none | 70 | 70 | 65 | 205 | 68 |
| 30 | female | group D | master's degree | standard | none | 62 | 70 | 75 | 207 | 69 |
| 31 | female | group D | some college | standard | none | 69 | 74 | 74 | 217 | 72 |
| 32 | female | group B | some college | standard | none | 63 | 65 | 61 | 189 | 63 |
| 33 | female | group E | master's degree | free/reduced | none | 56 | 72 | 65 | 193 | 64 |

| | gender | group | parental level of education | lunch | test preparation | math | reading | writing | total | average |
|---|---|---|---|---|---|---|---|---|---|---|
| 34 | male | group D | some college | standard | none | 40 | 42 | 38 | 120 | 40 |
| 35 | male | group E | some college | standard | none | 97 | 87 | 82 | 266 | 89 |
| 36 | male | group E | associate's degree | standard | completed | 81 | 81 | 79 | 241 | 80 |
| 37 | female | group D | associate's degree | standard | none | 74 | 81 | 83 | 238 | 79 |
| 38 | female | group D | some high school | free/reduced | none | 50 | 64 | 59 | 173 | 58 |
| 39 | female | group D | associate's degree | free/reduced | completed | 75 | 90 | 88 | 253 | 84 |
| 40 | male | group B | associate's degree | free/reduced | none | 57 | 56 | 57 | 170 | 57 |
| 41 | male | group C | associate's degree | free/reduced | none | 55 | 61 | 54 | 170 | 57 |
| 42 | female | group C | associate's degree | standard | none | 58 | 73 | 68 | 199 | 66 |
| 43 | female | group B | associate's degree | standard | none | 53 | 58 | 65 | 176 | 59 |
| 44 | male | group B | some college | free/reduced | completed | 59 | 65 | 66 | 190 | 63 |
| 45 | female | group E | associate's degree | free/reduced | none | 50 | 56 | 54 | 160 | 53 |
| 46 | male | group B | associate's degree | standard | none | 65 | 54 | 57 | 176 | 59 |
| 47 | female | group A | associate's degree | standard | completed | 55 | 65 | 62 | 182 | 61 |
| 48 | female | group C | high school | standard | none | 66 | 71 | 76 | 213 | 71 |
| 49 | female | group D | associate's degree | free/reduced | completed | 57 | 74 | 76 | 207 | 69 |
| 50 | male | group C | high school | standard | completed | 82 | 84 | 82 | 248 | 83 |
| 51 | male | group E | some college | standard | none | 53 | 55 | 48 | 156 | 52 |
| 52 | male | group E | associate's degree | free/reduced | completed | 77 | 69 | 68 | 214 | 71 |
| 53 | male | group C | some college | standard | none | 53 | 44 | 42 | 139 | 46 |
| 54 | male | group D | high school | standard | none | 88 | 78 | 75 | 241 | 80 |
| 55 | female | group C | some high school | free/reduced | completed | 71 | 84 | 87 | 242 | 81 |
| 56 | female | group C | high school | free/reduced | none | 33 | 41 | 43 | 117 | 39 |
| 57 | female | group E | associate's degree | standard | completed | 82 | 85 | 86 | 253 | 84 |
| 58 | male | group D | associate's degree | standard | none | 52 | 55 | 49 | 156 | 52 |
| 59 | male | group D | some college | standard | completed | 58 | 59 | 58 | 175 | 58 |
| 61 | male | group E | bachelor's degree | free/reduced | completed | 79 | 74 | 72 | 225 | 75 |
| 62 | male | group A | some high school | free/reduced | none | 39 | 39 | 34 | 112 | 37 |
| 63 | male | group A | associate's degree | free/reduced | none | 62 | 61 | 55 | 178 | 59 |
| 64 | female | group C | associate's degree | standard | none | 69 | 80 | 71 | 220 | 73 |
| 65 | female | group D | some high school | standard | none | 59 | 58 | 59 | 176 | 59 |
| 66 | male | group B | some high school | standard | none | 67 | 64 | 61 | 192 | 64 |
| 67 | male | group D | some high school | free/reduced | none | 45 | 37 | 37 | 119 | 40 |
| 68 | female | group C | some college | standard | none | 60 | 72 | 74 | 206 | 69 |
| 69 | male | group B | associate's degree | free/reduced | none | 61 | 58 | 56 | 175 | 58 |
| 70 | female | group C | associate's degree | standard | none | 39 | 64 | 57 | 160 | 53 |
| 71 | female | group D | some college | free/reduced | completed | 58 | 63 | 73 | 194 | 65 |
| 72 | male | group D | some college | standard | completed | 63 | 55 | 63 | 181 | 60 |
| 73 | female | group A | associate's degree | free/reduced | none | 41 | 51 | 48 | 140 | 47 |
| 74 | male | group C | some high school | free/reduced | none | 61 | 57 | 56 | 174 | 58 |
| 75 | male | group C | some high school | standard | none | 49 | 49 | 41 | 139 | 46 |
| 76 | male | group B | associate's degree | free/reduced | none | 44 | 41 | 38 | 123 | 41 |
| 78 | male | group A | bachelor's degree | standard | completed | 80 | 78 | 81 | 239 | 80 |
| 79 | female | group D | some high school | standard | completed | 61 | 74 | 72 | 207 | 69 |

```
80  female group E   master's degree    standard    none    62      68      68      198     66
81  female group B associate's degree   standard    none    47      49      50      146     49
82   male group B     high school free/reduced    none    49      45      45      139     46
83   male group A    some college free/reduced completed    50      47      54      151     50
84   male group E associate's degree   standard    none    72      64      63      199     66
85   male group D    high school free/reduced    none    42      39      34      115     38
86  female group C    some college   standard    none    73      80      82      235     78
87  female group C    some college free/reduced   none    76      83      88      247     82
88  female group D associate's degree   standard    none    71      71      74      216     72
89  female group A    some college   standard    none    58      70      67      195     65
90  female group D  some high school   standard    none    73      86      82      241     80
91  female group C bachelor's degree   standard    none    65      72      74      211     70
92   male group C     high school free/reduced    none    27      34      36      97      32
93   male group C     high school   standard    none    71      79      71      221     74
94   male group C associate's degree free/reduced completed    43      45      50      138     46
95  female group B    some college   standard    none    79      86      92      257     86
96   male group C associate's degree free/reduced completed    78      81      82      241     80
97   male group B  some high school   standard completed    65      66      62      193     64
98  female group E    some college   standard completed    63      72      70      205     68
99  female group D    some college free/reduced   none    58      67      62      187     62
100 female group D  bachelor's degree   standard    none    65      67      62      194     65
101  male group B    some college   standard    none    79      67      67      213     71
102  male group D bachelor's degree   standard completed    68      74      74      216     72
103 female group D associate's degree   standard    none    85      91      89      265     88
 [ reached 'max' / getOption("max.print") -- omitted 888 rows ]
>
> #comparison of avg scores - male vs female
> ggplot(performance_2, aes( x= Avg_score, color = Gender))+
+  geom_density() +
+  geom_vline( color = "red",linetype = "dashed", lwd =0.5  ,xintercept =
mean(performance_2[performance_2$Gender == "female",]$Avg_score))+
+  geom_vline( color = "cyan",linetype = "dashed", lwd=0.5 ,  xintercept =
mean(performance_2[performance_2$Gender == "male",]$Avg_score)) +
+  labs(title ="Distribution of scores by Gender", x ="Score", y = " Density")
>
> #From the above density plot, we see that scores of female students have a higher mean than male
students.
>
> #Q3) For the given 'chinook' database, perform the following tasks:
> #install.packages("DBI")
> library(DBI)
> #install.packages("readr")
> library(readr)
> #install.packages("RSQLite")
```

```
> library(RSQLite)
>
> #I.     Connect to the above database and convert all the tables into data frame
> con <- dbConnect(RSQLite::SQLite(),"chinook.db")
> db <- dbConnect(dbDriver("SQLite"), dbname="chinook.db")
> dbListTables(db)
 [1] "albums"      "artists"     "customers"    "employees"     "genres"      "invoice_items"
"invoices"
 [8] "media_types"   "playlist_track" "playlists"     "sqlite_sequence" "sqlite_stat1"   "tracks"
>
> albums <- dbReadTable(db, "albums")
> head(albums)
  AlbumId                Title ArtistId
1     1 For Those About To Rock We Salute You      1
2     2              Balls to the Wall       2
3     3              Restless and Wild       2
4     4              Let There Be Rock       1
5     5                   Big Ones       3
6     6              Jagged Little Pill       4
> artists <- dbReadTable(db, "artists")
> head(artists)
  ArtistId          Name
1     1          AC/DC
2     2          Accept
3     3          Aerosmith
4     4    Alanis Morissette
5     5     Alice In Chains
6     6 Antônio Carlos Jobim
> customers <- dbReadTable(db, "customers")
> head(customers)
  CustomerId FirstName   LastName                    Company              Address      City
1     1      Luís  Gonçalves Embraer - Empresa Brasileira de Aeronáutica S.A. Av. Brigadeiro Faria Lima,
2170 São José dos Campos
2     2 Leonie     Köhler                    <NA>     Theodor-Heuss-Straße 34     Stuttgart
3     3 François   Tremblay                    <NA>          1498 rue Bélanger     Montréal
4     4  Bjørn    Hansen                    <NA>        Ullevålsveien 14      Oslo
5     5 František Wichterlová             JetBrains s.r.o.      Klanova 9/506        Prague
6     6  Helena     Holý                    <NA>        Rilská 3174/6        Prague
  State    Country PostalCode      Phone      Fax        Email SupportRepId
1  SP     Brazil 12227-000 +55 (12) 3923-5555 +55 (12) 3923-5566   luisg@embraer.com.br       3
2 <NA>    Germany    70174  +49 0711 2842222        <NA> leonekohler@surfeu.de       5
3  QC     Canada   H2G 1A7  +1 (514) 721-4711        <NA> ftremblay@gmail.com       3
4 <NA>    Norway     0171   +47 22 44 22 22        <NA> bjorn.hansen@yahoo.no       4
```

```
5  <NA> Czech Republic     14700   +420 2 4172 5555   +420 2 4172 5555 frantisekw@jetbrains.com
4
6  <NA> Czech Republic     14300   +420 2 4177 0449            <NA>       hholy@gmail.com        5
> employees <- dbReadTable(db, "employees")
> head(employees)
  EmployeeId LastName FirstName           Title ReportsTo         BirthDate          HireDate
Address     City
1     1   Adams    Andrew   General Manager     NA 1962-02-18 00:00:00 2002-08-14 00:00:00
11120 Jasper Ave NW Edmonton
2     2 Edwards     Nancy     Sales Manager     1 1958-12-08 00:00:00 2002-05-01 00:00:00     825
8 Ave SW  Calgary
3     3 Peacock     Jane Sales Support Agent     2 1973-08-29 00:00:00 2002-04-01 00:00:00     1111
6 Ave SW  Calgary
4     4   Park  Margaret Sales Support Agent     2 1947-09-19 00:00:00 2003-05-03 00:00:00     683
10 Street SW  Calgary
5     5 Johnson    Steve Sales Support Agent     2 1965-03-03 00:00:00 2003-10-17 00:00:00
7727B 41 Ave  Calgary
6     6 Mitchell  Michael       IT Manager     1 1973-07-01 00:00:00 2003-10-17 00:00:00 5827
Bowness Road NW  Calgary
  State Country PostalCode         Phone         Fax           Email
1  AB  Canada   T5K 2N1 +1 (780) 428-9482 +1 (780) 428-3457  andrew@chinookcorp.com
2  AB  Canada   T2P 2T3 +1 (403) 262-3443 +1 (403) 262-3322   nancy@chinookcorp.com
3  AB  Canada   T2P 5M5 +1 (403) 262-3443 +1 (403) 262-6712    jane@chinookcorp.com
4  AB  Canada   T2P 5G3 +1 (403) 263-4423 +1 (403) 263-4289 margaret@chinookcorp.com
5  AB  Canada   T3B 1Y7  1 (780) 836-9987  1 (780) 836-9543   steve@chinookcorp.com
6  AB  Canada   T3B 0C5 +1 (403) 246-9887 +1 (403) 246-9899 michael@chinookcorp.com
> genres <- dbReadTable(db, "genres")
> head(genres)
  GenreId        Name
1  1        Rock
2  2        Jazz
3  3        Metal
4  4 Alternative & Punk
5  5    Rock And Roll
6  6        Blues
> invoice_items <- dbReadTable(db, "invoice_items")
> head(invoice_items)
  InvoiceLineId InvoiceId TrackId UnitPrice Quantity
1     1     1     2  0.99     1
2     2     1     4  0.99     1
3     3     2     6  0.99     1
4     4     2     8  0.99     1
5     5     2    10   0.99     1
6     6     2    12   0.99     1
```

```
> invoices <- dbReadTable(db, "invoices")
> head(invoices)
  InvoiceId CustomerId      InvoiceDate      BillingAddress BillingCity BillingState BillingCountry
BillingPostalCode Total
1      1        2 2009-01-01 00:00:00 Theodor-Heuss-Straße 34  Stuttgart      <NA>      Germany
70174  1.98
2      2        4 2009-01-02 00:00:00      Ullevålsveien 14      Oslo      <NA>      Norway      0171
3.96
3      3        8 2009-01-03 00:00:00      Grétrystraat 63    Brussels      <NA>      Belgium      1000
5.94
4      4       14 2009-01-06 00:00:00      8210 111 ST NW    Edmonton      AB      Canada      T6G
2C7  8.91
5      5       23 2009-01-11 00:00:00      69 Salem Street      Boston      MA      USA      2113
13.86
6      6       37 2009-01-19 00:00:00      Berger Straße 10  Frankfurt      <NA>      Germany
60316  0.99
> media_types <- dbReadTable(db, "media_types")
> head(media_types)
  MediaTypeId               Name
1      1         MPEG audio file
2      2   Protected AAC audio file
3      3 Protected MPEG-4 video file
4      4   Purchased AAC audio file
5      5         AAC audio file
> playlist_track <- dbReadTable(db, "playlist_track")
> head(playlist_track)
  PlaylistId TrackId
1      1   3402
2      1   3389
3      1   3390
4      1   3391
5      1   3392
6      1   3393
> playlists <- dbReadTable(db, "playlists")
> head(playlists)
  PlaylistId     Name
1      1    Music
2      2    Movies
3      3   TV Shows
4      4  Audiobooks
5      5 90's Music
6      6  Audiobooks
> tracks <- dbReadTable(db, "tracks")
> head(tracks)
```

```
  TrackId                              Name AlbumId MediaTypeId GenreId
1       1 For Those About To Rock (We Salute You)       1           1       1
2       2                       Balls to the Wall       2           2       1
3       3                         Fast As a Shark       3           2       1
4       4                       Restless and Wild       3           2       1
5       5                     Princess of the Dawn       3           2       1
6       6                     Put The Finger On You       1           1       1
                                                  Composer Milliseconds    Bytes UnitPrice
1                   Angus Young, Malcolm Young, Brian Johnson       343719 11170334     0.99
2                                                       <NA>       342562  5510424     0.99
3                 F. Baltes, S. Kaufman, U. Dirkscneider & W. Hoffman       230619  3990994     0.99
4 F. Baltes, R.A. Smith-Diesel, S. Kaufman, U. Dirkscneider & W. Hoffman       252051  4331779     0.99
5                                 Deaffy & R.A. Smith-Diesel       375418  6290521     0.99
6                   Angus Young, Malcolm Young, Brian Johnson       205662  6713451     0.99
>
> #II.    Print the different types of music available
> genres$Name
 [1] "Rock"              "Jazz"              "Metal"             "Alternative & Punk" "Rock And Roll"     "Blues"
 [7] "Latin"             "Reggae"            "Pop"               "Soundtrack"        "Bossa Nova"        "Easy Listening"
[13] "Heavy Metal"       "R&B/Soul"          "Electronica/Dance" "World"             "Hip Hop/Rap"
"Science Fiction"
[19] "TV Shows"          "Sci Fi & Fantasy"  "Drama"             "Comedy"            "Alternative"       "Classical"
[25] "Opera"
>
> #III.    List out all the artists from the entire database
> artists$Name
 [1] "AC/DC"
 [2] "Accept"
 [3] "Aerosmith"
 [4] "Alanis Morissette"
 [5] "Alice In Chains"
 [6] "Antônio Carlos Jobim"
 [7] "Apocalyptica"
 [8] "Audioslave"
 [9] "BackBeat"
[10] "Billy Cobham"
[11] "Black Label Society"
[12] "Black Sabbath"
[13] "Body Count"
[14] "Bruce Dickinson"
[15] "Buddy Guy"
[16] "Caetano Veloso"
[17] "Chico Buarque"
[18] "Chico Science & Nação Zumbi"
```

[19] "Cidade Negra"
[20] "Cláudio Zoli"
[21] "Various Artists"
[22] "Led Zeppelin"
[23] "Frank Zappa & Captain Beefheart"
[24] "Marcos Valle"
[25] "Milton Nascimento & Bebeto"
[26] "Azymuth"
[27] "Gilberto Gil"
[28] "João Gilberto"
[29] "Bebel Gilberto"
[30] "Jorge Vercilo"
[31] "Baby Consuelo"
[32] "Ney Matogrosso"
[33] "Luiz Melodia"
[34] "Nando Reis"
[35] "Pedro Luís & A Parede"
[36] "O Rappa"
[37] "Ed Motta"
[38] "Banda Black Rio"
[39] "Fernanda Porto"
[40] "Os Cariocas"
[41] "Elis Regina"
[42] "Milton Nascimento"
[43] "A Cor Do Som"
[44] "Kid Abelha"
[45] "Sandra De Sá"
[46] "Jorge Ben"
[47] "Hermeto Pascoal"
[48] "Barão Vermelho"
[49] "Edson, DJ Marky & DJ Patife Featuring Fernanda Porto"
[50] "Metallica"
[51] "Queen"
[52] "Kiss"
[53] "Spyro Gyra"
[54] "Green Day"
[55] "David Coverdale"
[56] "Gonzaguinha"
[57] "Os Mutantes"
[58] "Deep Purple"
[59] "Santana"
[60] "Santana Feat. Dave Matthews"
[61] "Santana Feat. Everlast"
[62] "Santana Feat. Rob Thomas"

[63] "Santana Feat. Lauryn Hill & Cee-Lo"
[64] "Santana Feat. The Project G&B"
[65] "Santana Feat. Maná"
[66] "Santana Feat. Eagle-Eye Cherry"
[67] "Santana Feat. Eric Clapton"
[68] "Miles Davis"
[69] "Gene Krupa"
[70] "Toquinho & Vinícius"
[71] "Vinícius De Moraes & Baden Powell"
[72] "Vinícius De Moraes"
[73] "Vinícius E Qurteto Em Cy"
[74] "Vinícius E Odette Lara"
[75] "Vinicius, Toquinho & Quarteto Em Cy"
[76] "Creedence Clearwater Revival"
[77] "Cássia Eller"
[78] "Def Leppard"
[79] "Dennis Chambers"
[80] "Djavan"
[81] "Eric Clapton"
[82] "Faith No More"
[83] "Falamansa"
[84] "Foo Fighters"
[85] "Frank Sinatra"
[86] "Funk Como Le Gusta"
[87] "Godsmack"
[88] "Guns N' Roses"
[89] "Incognito"
[90] "Iron Maiden"
[91] "James Brown"
[92] "Jamiroquai"
[93] "JET"
[94] "Jimi Hendrix"
[95] "Joe Satriani"
[96] "Jota Quest"
[97] "João Suplicy"
[98] "Judas Priest"
[99] "Legião Urbana"
[100] "Lenny Kravitz"
[101] "Lulu Santos"
[102] "Marillion"
[103] "Marisa Monte"
[104] "Marvin Gaye"
[105] "Men At Work"
[106] "Motörhead"

[151] "UB40"
[152] "Van Halen"
[153] "Velvet Revolver"
[154] "Whitesnake"
[155] "Zeca Pagodinho"
[156] "The Office"
[157] "Dread Zeppelin"
[158] "Battlestar Galactica (Classic)"
[159] "Aquaman"
[160] "Christina Aguilera featuring BigElf"
[161] "Aerosmith & Sierra Leone's Refugee Allstars"
[162] "Los Lonely Boys"
[163] "Corinne Bailey Rae"
[164] "Dhani Harrison & Jakob Dylan"
[165] "Jackson Browne"
[166] "Avril Lavigne"
[167] "Big & Rich"
[168] "Youssou N'Dour"
[169] "Black Eyed Peas"
[170] "Jack Johnson"
[171] "Ben Harper"
[172] "Snow Patrol"
[173] "Matisyahu"
[174] "The Postal Service"
[175] "Jaguares"
[176] "The Flaming Lips"
[177] "Jack's Mannequin & Mick Fleetwood"
[178] "Regina Spektor"
[179] "Scorpions"
[180] "House Of Pain"
[181] "Xis"
[182] "Nega Gizza"
[183] "Gustavo & Andres Veiga & Salazar"
[184] "Rodox"
[185] "Charlie Brown Jr."
[186] "Pedro Luís E A Parede"
[187] "Los Hermanos"
[188] "Mundo Livre S/A"
[189] "Otto"
[190] "Instituto"
[191] "Nação Zumbi"
[192] "DJ Dolores & Orchestra Santa Massa"
[193] "Seu Jorge"
[194] "Sabotage E Instituto"

[195] "Stereo Maracana"
[196] "Cake"
[197] "Aisha Duo"
[198] "Habib Koité and Bamada"
[199] "Karsh Kale"
[200] "The Posies"
[201] "Luciana Souza/Romero Lubambo"
[202] "Aaron Goldberg"
[203] "Nicolaus Esterhazy Sinfonia"
[204] "Temple of the Dog"
[205] "Chris Cornell"
[206] "Alberto Turco & Nova Schola Gregoriana"
[207] "Richard Marlow & The Choir of Trinity College, Cambridge"
[208] "English Concert & Trevor Pinnock"
[209] "Anne-Sophie Mutter, Herbert Von Karajan & Wiener Philharmoniker"
[210] "Hilary Hahn, Jeffrey Kahane, Los Angeles Chamber Orchestra & Margaret Batjer"
[211] "Wilhelm Kempff"
[212] "Yo-Yo Ma"
[213] "Scholars Baroque Ensemble"
[214] "Academy of St. Martin in the Fields & Sir Neville Marriner"
[215] "Academy of St. Martin in the Fields Chamber Ensemble & Sir Neville Marriner"
[216] "Berliner Philharmoniker, Claudio Abbado & Sabine Meyer"
[217] "Royal Philharmonic Orchestra & Sir Thomas Beecham"
[218] "Orchestre Révolutionnaire et Romantique & John Eliot Gardiner"
[219] "Britten Sinfonia, Ivor Bolton & Lesley Garrett"
[220] "Chicago Symphony Chorus, Chicago Symphony Orchestra & Sir Georg Solti"
[221] "Sir Georg Solti & Wiener Philharmoniker"
[222] "Academy of St. Martin in the Fields, John Birch, Sir Neville Marriner & Sylvia McNair"
[223] "London Symphony Orchestra & Sir Charles Mackerras"
[224] "Barry Wordsworth & BBC Concert Orchestra"
[225] "Herbert Von Karajan, Mirella Freni & Wiener Philharmoniker"
[226] "Eugene Ormandy"
[227] "Luciano Pavarotti"
[228] "Leonard Bernstein & New York Philharmonic"
[229] "Boston Symphony Orchestra & Seiji Ozawa"
[230] "Aaron Copland & London Symphony Orchestra"
[231] "Ton Koopman"
[232] "Sergei Prokofiev & Yuri Temirkanov"
[233] "Chicago Symphony Orchestra & Fritz Reiner"
[234] "Orchestra of The Age of Enlightenment"
[235] "Emanuel Ax, Eugene Ormandy & Philadelphia Orchestra"
[236] "James Levine"
[237] "Berliner Philharmoniker & Hans Rosbaud"
[238] "Maurizio Pollini"

[239] "Academy of St. Martin in the Fields, Sir Neville Marriner & William Bennett"
[240] "Gustav Mahler"
[241] "Felix Schmidt, London Symphony Orchestra & Rafael Frühbeck de Burgos"
[242] "Edo de Waart & San Francisco Symphony"
[243] "Antal Doráti & London Symphony Orchestra"
[244] "Choir Of Westminster Abbey & Simon Preston"
[245] "Michael Tilson Thomas & San Francisco Symphony"
[246] "Chor der Wiener Staatsoper, Herbert Von Karajan & Wiener Philharmoniker"
[247] "The King's Singers"
[248] "Berliner Philharmoniker & Herbert Von Karajan"
[249] "Sir Georg Solti, Sumi Jo & Wiener Philharmoniker"
[250] "Christopher O'Riley"
[251] "Fretwork"
[252] "Amy Winehouse"
[253] "Calexico"
[254] "Otto Klemperer & Philharmonia Orchestra"
[255] "Yehudi Menuhin"
[256] "Philharmonia Orchestra & Sir Neville Marriner"
[257] "Academy of St. Martin in the Fields, Sir Neville Marriner & Thurston Dart"
[258] "Les Arts Florissants & William Christie"
[259] "The 12 Cellists of The Berlin Philharmonic"
[260] "Adrian Leaper & Doreen de Feis"
[261] "Roger Norrington, London Classical Players"
[262] "Charles Dutoit & L'Orchestre Symphonique de Montréal"
[263] "Equale Brass Ensemble, John Eliot Gardiner & Munich Monteverdi Orchestra and Choir"
[264] "Kent Nagano and Orchestre de l'Opéra de Lyon"
[265] "Julian Bream"
[266] "Martin Roscoe"
[267] "Göteborgs Symfoniker & Neeme Järvi"
[268] "Itzhak Perlman"
[269] "Michele Campanella"
[270] "Gerald Moore"
[271] "Mela Tenenbaum, Pro Musica Prague & Richard Kapp"
[272] "Emerson String Quartet"
[273] "C. Monteverdi, Nigel Rogers - Chiaroscuro; London Baroque; London Cornett & Sackbu"
[274] "Nash Ensemble"
[275] "Philip Glass Ensemble"
>
> #IV.   List out all the countries where the customer resides and plot a bar graph showing the number of customers from the respective country
> unique(customers$Country)
 [1] "Brazil"      "Germany"     "Canada"      "Norway"      "Czech Republic" "Austria"      "Belgium"
 [8] "Denmark"     "USA"         "Portugal"    "France"      "Finland"     "Hungary"      "Ireland"
[15] "Italy"       "Netherlands" "Poland"      "Spain"       "Sweden"      "United Kingdom" "Australia"

```
[22] "Argentina"    "Chile"        "India"
> plot2 <-
+   ggplot() +
+   geom_bar(data = customers, aes(x = Country), width = 0.3, fill = "turquoise") +
+   geom_text(stat='count', data = customers, aes(x = Country, label=..count..), vjust=-0.2) +
+   theme_bw() +
+   xlab("Country") +
+   ylab("Number of Customers") +
+   theme_classic() +
+   theme(axis.text.x=element_text(angle=90, hjust=1)) +
+   ggtitle("Number of Customers by Country") +
+   scale_fill_brewer(type = "qual", palette = 1, direction = 1,
+               aesthetics = "fill")
> plot2
```

## Day 6 – R Programming

**Happiness ~ Income (Simple Linear Regression)**

```
> #install.packages("broom")
> #install.packages("ggpubr")
> library(ggplot2)
> library(dplyr)
> library(broom)
> library(ggpubr)
> setwd("C:/zubeda/PGA02_Zubu/R Programming/Models")
> dev.off()
null device
      1
>
> #Importing Data
> income.data <- read.csv("income.data.csv")
> income.data
   X   income happiness
1    1 3.862647 2.3144890
2    2 4.979381 3.4334898
3    3 4.923957 4.5993734
4    4 3.214372 2.7911138
5    5 7.196409 5.5963983
6    6 3.729643 2.4585559
7    7 4.674517 3.1929918
8    8 4.498104 1.9071368
9    9 3.121631 2.9424499
10   10 4.639914 3.7379416
11   11 4.632840 3.1754061
12   12 2.773179 2.0090465
```

```
13   13 7.119479 5.9518141
14   14 7.466653 5.9605473
15   15 2.117742 1.4457989
16   16 2.559166 2.8985831
17   17 2.354793 1.2311675
18   18 2.388157 2.3129881
19   19 4.755680 2.6661160
20   20 1.994275 2.5847290
21   21 7.310916 5.7474441
22   22 3.528319 2.5465246
23   23 2.428752 1.2007855
24   24 3.542748 3.0782934
25   25 5.227201 4.3177609
26   26 6.691993 5.3814787
27   27 3.900410 3.5652243
28   28 2.291055 0.9534130
29   29 2.380513 2.1691613
30   30 2.549609 2.0607943
31   31 6.933296 6.2991013
32   32 1.855645 1.5903559
33   33 3.589023 2.2509294
34   34 6.826478 5.9142477
35   35 2.070602 2.1918337
36   36 5.224205 5.7678144
37   37 2.243114 0.9728829
38   38 7.076166 5.0105774
39   39 4.190672 2.2396650
40   40 1.956486 1.9275788
41   41 5.061758 3.3580716
42   42 3.982190 2.4000873
43   43 3.065059 3.4079800
44   44 3.682877 2.5761763
45   45 3.789429 2.4730794
46   46 5.358716 3.7526595
47   47 5.196120 4.0876312
48   48 5.241190 3.5432037
49   49 7.101620 5.3483529
50   50 3.424021 3.0563767
51   51 2.253399 1.5584226
52   52 5.370337 3.2251328
53   53 6.225606 5.0342310
54   54 5.482862 3.8574243
55   55 4.034172 3.6190555
56   56 6.510219 4.0045377
```

```
 57   57 6.029214 4.8020918
 58   58 6.949113 4.6588904
 59   59 7.195037 5.2317030
 60   60 2.757338 2.4806065
 61   61 6.956079 5.4981472
 62   62 4.670193 4.5506370
 63   63 6.368293 3.5700136
 64   64 6.166681 4.7196653
 65   65 6.074158 4.5031082
 66   66 5.484719 5.0460818
 67   67 1.589575 0.6697159
 68   68 1.680474 1.6060724
 69   69 5.499948 4.8266027
 70   70 4.043891 2.2082405
 71   71 5.005093 4.0564931
 72   72 4.863582 3.5679052
 73   73 1.506275 1.3084873
 74   74 2.864664 4.1596093
 75   75 5.877906 4.6339151
 76   76 6.483984 5.0687479
 77   77 4.938037 3.0407973
 78   78 5.625434 3.8042989
 79   79 7.228265 5.0340038
 80   80 5.337460 3.7034379
 81   81 2.825827 2.1889381
 82   82 5.931367 5.5380475
 83   83 3.520255 3.5838752
 84   84 3.239941 3.0968856
 85   85 3.498386 2.2009822
 86   86 7.186112 5.1515983
 87   87 4.719166 5.9509863
 88   88 3.594802 2.9681871
 89   89 3.233942 2.3995613
 90   90 1.514153 0.8594991
 91   91 4.002537 1.7759326
 92   92 6.198104 4.6612612
 93   93 2.280651 0.7272212
 94   94 2.189866 0.7712866
 95   95 3.434151 3.3487882
 96   96 5.932270 3.9662154
 97   97 5.307839 2.8904474
 98   98 5.664345 3.7732607
 99   99 7.439248 6.3596000
100  100 2.134702 0.2687221
```

```
101 101 6.501275 4.3748323
102 102 3.651183 2.1558433
103 103 2.286495 1.8935569
104 104 4.748859 4.9029916
105 105 5.459161 4.8335064
106 106 3.433065 3.1722995
107 107 7.176400 5.0299517
108 108 5.506395 4.2610130
109 109 3.097616 1.6723906
110 110 4.647556 1.4970241
111 111 1.828306 1.2654889
112 112 3.534566 2.6674654
113 113 4.606176 1.9993255
114 114 5.361503 5.2318633
115 115 6.879333 5.2114013
116 116 4.317032 3.6616565
117 117 3.383164 1.4150347
118 118 4.932207 4.9330441
119 119 4.935597 4.1307783
120 120 2.601553 2.2822669
121 121 5.711264 3.9011703
122 122 6.117531 4.6919989
123 123 3.771415 3.5778007
124 124 7.117220 5.5625455
125 125 2.194882 2.3932281
126 126 5.952002 3.5647237
127 127 3.922303 2.2537215
128 128 7.081589 4.1216477
129 129 6.950745 4.1691008
130 130 3.660877 3.8238987
131 131 1.789092 0.4583776
132 132 3.540341 2.5769400
133 133 4.533395 2.9475315
134 134 4.867339 3.7399958
135 135 4.056005 3.5714465
136 136 5.634643 4.8081504
137 137 5.461636 4.0176112
138 138 3.186176 1.8398020
139 139 4.417666 3.4685738
140 140 5.760289 4.7587855
141 141 3.716700 2.3916775
142 142 2.182562 0.9929174
143 143 4.291984 3.1693802
144 144 3.410030 2.0890424
```

```
145 145 3.581097 1.8436758
146 146 3.509663 1.6166074
147 147 6.660216 5.9493475
148 148 6.271786 4.9402278
149 149 3.735018 2.8412387
150 150 4.393208 2.9443913
151 151 3.512217 3.0269182
152 152 6.239740 5.0978025
153 153 3.681486 3.0368199
154 154 7.241313 4.6828170
155 155 6.345370 4.0008192
156 156 5.939742 4.5708011
157 157 2.459321 2.0427393
158 158 2.539089 1.7486511
159 159 6.708604 6.0640130
160 160 6.831322 5.1324015
161 161 5.082658 3.1133421
162 162 6.030607 4.8218103
163 163 6.574595 4.1795210
164 164 3.574297 1.6312611
165 165 5.529908 3.8221796
166 166 2.409382 1.8575419
167 167 4.264790 3.7510893
168 168 3.530345 3.1586186
169 169 6.143150 4.9271326
170 170 5.157697 4.6001148
171 171 4.710847 2.4083799
172 172 6.847515 4.4867037
173 173 5.464640 2.7277157
174 174 4.176532 3.0207244
175 175 3.748093 3.7491807
176 176 2.274523 2.3115542
177 177 1.576366 0.9876032
178 178 1.924134 1.4611444
179 179 5.904246 4.5768565
180 180 5.189031 4.7675956
181 181 1.879868 0.4381716
182 182 2.544348 2.3512025
183 183 3.221394 3.4531027
184 184 7.260374 5.3813985
185 185 6.481617 4.8200912
186 186 5.688488 4.6587427
187 187 6.633619 5.3800702
188 188 5.972741 3.3159895
```

```
189 189 3.897738 2.7997475
190 190 6.461243 4.2067549
191 191 6.628036 4.4026632
192 192 3.118959 2.7691181
193 193 4.695964 2.7842445
194 194 1.573694 0.6880906
195 195 3.670377 3.4764991
196 196 7.194407 5.8361967
197 197 1.780479 2.0039261
198 198 2.142360 0.9713325
199 199 3.656486 2.8576144
200 200 2.090354 1.8521177
201 201 3.363097 3.5151792
202 202 2.423144 2.1005525
203 203 7.111584 6.0864783
204 204 3.039942 4.0838212
205 205 2.373232 1.5247823
206 206 1.984564 2.5795165
207 207 2.628483 1.6192261
208 208 7.136760 5.5066215
209 209 3.104918 1.0959993
210 210 1.558657 0.9685273
211 211 7.478447 4.8777255
212 212 2.813900 1.7596987
213 213 5.744540 5.0322110
214 214 6.540988 5.7138023
215 215 6.562794 4.7958429
216 216 5.470125 4.6602900
217 217 2.085131 2.7403954
218 218 4.589572 4.2505681
219 219 5.074502 3.9410193
220 220 7.463510 4.5034454
221 221 5.853906 4.6332287
222 222 3.764540 4.0308649
223 223 7.062792 6.8633880
224 224 6.377376 5.0794759
225 225 1.920863 1.6252684
226 226 7.364213 6.6182798
227 227 6.535799 4.6569986
228 228 7.300903 6.0049875
229 229 3.037232 2.4106356
230 230 6.703267 4.2612200
231 231 1.927997 1.0656238
232 232 4.223554 2.2957001
```

```
233 233 2.922706 1.9195111
234 234 4.427109 3.5813810
235 235 2.070562 0.6289421
236 236 5.224070 3.3441084
237 237 7.161873 6.1604341
238 238 2.210696 2.8612744
239 239 7.207060 4.2094706
240 240 2.184085 1.2262981
241 241 4.414998 4.7933252
242 242 5.014810 3.3137965
243 243 2.602037 2.1526984
244 244 2.917049 2.9236839
245 245 6.244342 3.4520672
246 246 6.859654 4.2583115
247 247 2.371230 3.9610328
248 248 5.964058 3.4277232
249 249 7.153674 4.7209873
250 250 2.289897 1.5719498
251 251 6.376228 6.1869456
252 252 3.540504 3.5527370
253 253 3.139826 1.2457859
254 254 6.460742 3.4269117
255 255 2.641348 1.8771470
256 256 2.002214 1.8178891
257 257 6.391428 5.2399095
258 258 2.763720 1.3840441
259 259 6.831258 3.8379778
260 260 3.827255 2.3216977
261 261 3.770671 3.3115782
262 262 3.159855 3.0573878
263 263 5.099417 4.7370460
264 264 5.610391 4.5909705
265 265 1.856372 0.6542421
266 266 5.363730 3.9811277
267 267 2.336134 2.3362760
268 268 4.975851 4.5406361
269 269 2.629547 2.1927449
270 270 2.646458 1.7881763
271 271 3.859892 2.7194322
272 272 4.121531 4.2126973
273 273 6.385941 4.4772714
274 274 3.842710 2.4688600
275 275 4.990549 3.4517725
276 276 3.400597 0.6869208
```

277 277 3.820115 2.2467457
278 278 1.909499 1.6276951
279 279 2.858464 2.4532091
280 280 7.451501 4.1099447
281 281 3.354252 2.5725982
282 282 6.707825 4.8104241
283 283 6.325906 4.8274163
284 284 1.931181 2.4657611
285 285 5.128545 3.7573790
286 286 4.278210 3.6913482
287 287 1.579470 1.9818099
288 288 2.907682 1.4102296
289 289 5.644714 3.7543014
290 290 3.571175 4.2452156
291 291 2.345108 1.4378767
292 292 5.845197 4.1598871
293 293 5.298480 3.5148918
294 294 3.434700 2.7144980
295 295 1.865995 1.3573731
296 296 5.095615 5.6384604
297 297 1.530808 2.4214647
298 298 1.618311 1.5112051
299 299 6.687677 5.5228917
300 300 7.347246 4.3966223
301 301 5.983846 3.9884528
302 302 5.196082 3.3176068
303 303 4.193007 2.7796826
304 304 2.347688 0.8465055
305 305 4.709489 2.4579217
306 306 2.307198 3.5096047
307 307 2.730769 2.6469045
308 308 3.882514 2.9779167
309 309 3.574394 1.4244505
310 310 4.159576 1.7903423
311 311 1.544634 1.6446366
312 312 3.383897 0.7316363
313 313 3.614745 2.9017886
314 314 6.503882 4.3171332
315 315 1.848413 2.0973691
316 316 4.420312 4.3915930
317 317 6.477153 5.3139569
318 318 6.561420 6.2813702
319 319 7.180907 5.6079666
320 320 2.809091 2.9468704

321 321 5.686205 3.8727888
322 322 4.800344 2.8428684
323 323 2.412912 1.4606163
324 324 2.925704 3.7528248
325 325 3.174176 3.1266032
326 326 2.685530 2.8211926
327 327 2.124429 2.7349601
328 328 2.694022 2.1975921
329 329 4.230889 4.1555409
330 330 5.350516 4.0782088
331 331 5.091580 4.4569636
332 332 6.250302 4.9392590
333 333 5.324633 3.7305700
 [ reached 'max' / getOption("max.print") -- omitted 165 rows ]
> dim(income.data)
[1] 498   3
> summary(income.data)
      X          income       happiness
 Min.   : 1.0   Min.   :1.506   Min.   :0.266
 1st Qu.:125.2   1st Qu.:3.006   1st Qu.:2.266
 Median :249.5   Median :4.424   Median :3.473
 Mean   :249.5   Mean   :4.467   Mean   :3.393
 3rd Qu.:373.8   3rd Qu.:5.992   3rd Qu.:4.503
 Max.   :498.0   Max.   :7.482   Max.   :6.863
>
> #Assumptions
> hist(income.data$happiness)   #normally distributed
> plot(happiness ~ income, data=income.data)   #linearity x ~ y
> #Homoscedasticity or homogeneity of variance will be checked after model building
>
> #Linear Regression Analysis
> income.happiness.lm <- lm(happiness ~ income, data=income.data)
> summary(income.happiness.lm)

Call:
lm(formula = happiness ~ income, data = income.data)

Residuals:
    Min      1Q   Median      3Q     Max
-2.02479 -0.48526  0.04078  0.45898  2.37805

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.20427    0.08884   2.299   0.0219 *

income     0.71383   0.01854  38.505  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7181 on 496 degrees of freedom
Multiple R-squared:  0.7493,      Adjusted R-squared:  0.7488
F-statistic:  1483 on 1 and 496 DF,  p-value: < 2.2e-16

```
> par(mfrow=c(2, 2))
> plot(income.happiness.lm)   #Homoscedasticity, Residuals normally distributed
> par(mfrow=c(1, 1))
>
> #Visualize results
> income.graph <- ggplot(income.data, aes(x=income, y=happiness)) + geom_point()
> income.graph
> income.graph <- income.graph + geom_smooth(method = "lm", col="black")
> income.graph
`geom_smooth()` using formula 'y ~ x'
> income.graph <- income.graph + stat_regline_equation(label.x=3, label.y=7)  #regression line eq. y =
mx + c
> income.graph
`geom_smooth()` using formula 'y ~ x'
> income.graph + theme_bw() +
+  labs(title="Reported Happiness as a function of Income", x="Income(x$10,000)", y="Happiness(1 to
10)")
`geom_smooth()` using formula 'y ~ x'
```

**Cars Distance ~ Speed (Simple Linear Regression)**
```
> cars
  speed dist
1    4   2
2    4  10
3    7   4
4    7  22
5    8  16
6    9  10
7   10  18
8   10  26
9   10  34
10  11  17
11  11  28
12  12  14
13  12  20
14  12  24
```

```
15  12  28
16  13  26
17  13  34
18  13  34
19  13  46
20  14  26
21  14  36
22  14  60
23  14  80
24  15  20
25  15  26
26  15  54
27  16  32
28  16  40
29  17  32
30  17  40
31  17  50
32  18  42
33  18  56
34  18  76
35  18  84
36  19  36
37  19  46
38  19  68
39  20  32
40  20  48
41  20  52
42  20  56
43  20  64
44  22  66
45  23  54
46  24  70
47  24  92
48  24  93
49  24 120
50  25  85
> ?cars
> summary(cars)
    speed         dist
 Min.   : 4.0  Min.   : 2.00
 1st Qu.:12.0  1st Qu.: 26.00
 Median :15.0  Median : 36.00
 Mean   :15.4  Mean   : 42.98
 3rd Qu.:19.0  3rd Qu.: 56.00
```

Max.   :25.0   Max.   :120.00
> plot(cars, col="blue", pch=20, cex=2, main="Relationship between Speed and Stopping Distance for 10 Cars", xlab="Speed in mph", ylab="Stopping Distance in feet")
> set.seed(1)   #generates random numbers, gives same set of numbers (Set seed every time if we need same number)
> sample(3)
[1] 1 2 3
>
> mt <- matrix(1:10, ncol = 5)
> mt
    [,1] [,2] [,3] [,4] [,5]
[1,]   1   3   5   7   9
[2,]   2   4   6   8  10
> scale(mt, center=TRUE, scale=FALSE)
    [,1] [,2] [,3] [,4] [,5]
[1,] -0.5 -0.5 -0.5 -0.5 -0.5
[2,] 0.5  0.5  0.5  0.5  0.5
attr(,"scaled:center")
[1] 1.5 3.5 5.5 7.5 9.5
>
> set.seed(2)   #Works like random_state from python
> speed.c <- scale(cars$speed, center=TRUE, scale=FALSE)
> mod1 <- lm(formula=dist ~ speed.c, data=cars)
> mod1

Call:
lm(formula = dist ~ speed.c, data = cars)

Coefficients:
(Intercept)     speed.c
   42.980       3.932

> summary(mod1)

Call:
lm(formula = dist ~ speed.c, data = cars)

Residuals:
   Min    1Q  Median    3Q    Max
-29.069 -9.525 -2.272  9.215  43.201

Coefficients:
        Estimate Std. Error t value Pr(>|t|)
(Intercept) 42.9800    2.1750 19.761 < 2e-16 ***

```
speed.c     3.9324    0.4155   9.464 1.49e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 15.38 on 48 degrees of freedom
Multiple R-squared:  0.6511,     Adjusted R-squared:  0.6438
F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.49e-12
```

**Heart Disease Prediction (Simple Linear Regression)**

```
> library(ggplot2)
> library(dplyr)
> library(broom)
> library(ggpubr)
> setwd("C:/zubeda/PGA02_Zubu/R Programming/Models")
> dev.off()
null device
      1
>
> #Importing Data
> heart.data <- read.csv("heart.data.csv")
> heart.data
    X   biking   smoking heart.disease
1    1 30.801246 10.8966080   11.7694228
2    2 65.129215  2.2195632    2.8540815
3    3  1.959665 17.5883305   17.1778035
4    4 44.800196  2.8025589    6.8166469
5    5 69.428454 15.9745046    4.0622235
6    6 54.403626 29.3331755    9.5500460
7    7 49.056162  9.0608458    7.6245070
8    8  4.784604 12.8350208   15.8546544
9    9 65.730788 11.9912973    3.0674617
10  10 35.257449 23.2776834   12.0984844
11  11 51.825567 14.4351184    6.4302482
12  12 52.936197 25.0748686    8.6082721
13  13 48.767478 11.0232710    6.7225238
14  14 26.166801  6.6457495   10.5978071
15  15 10.553075  5.9905063   14.0794783
16  16 47.163716 14.0978372    8.7448453
17  17 61.685256 16.8408167    5.4433420
18  18 33.944394  5.7585952    9.1623064
19  19 39.697624 12.6628694    9.7471858
20  20 63.124698 22.9174800    5.8582779
21  21 28.510129 14.8551064   11.7247416
22  22 18.525973 26.4049774   16.0281877
```

```
23  23 24.479470 26.9249607   15.0007154
24  24 18.358646 23.4319568   16.4882059
25  25 30.388184 16.9860864   12.3566075
26  26 52.985220 27.6890270    9.0884449
27  27 60.509448  3.9819621    3.2172143
28  28 45.247110  2.1374753    6.5937191
29  29 48.597044 10.3884264    6.6594202
30  30 25.139771  5.8363728   11.4829371
31  31 44.173095  3.9676057    7.9822275
32  32 61.146946 27.7834060    7.7224625
33  33 27.267898 16.8532932   12.1648166
34  34 49.527100 15.2497308    8.0274043
35  35 20.197206 10.3314895   13.4165474
36  36 18.811228 16.7534420   15.0873602
37  37 67.350765 23.8737268    5.2798634
38  38 29.904475 24.5845499   13.5076192
39  39 14.011760 21.0121418   15.9929864
40  40 45.815488  4.7161269    7.0406211
41  41 31.477251 22.1658255   13.2385262
42  42 17.108204  1.3528783   11.5357354
43  43  9.665082  3.5042963   12.8868168
44  44 23.933005  4.1858692   10.8944571
45  45 22.636301 13.6789828   13.4660327
46  46 27.247477 13.3797768   12.3288989
47  47 20.789602 19.4554286   14.4157831
48  48 46.613715  9.2493326    6.8093546
49  49 28.622632 12.4827339   11.3683650
50  50 21.127498 18.9413483   14.8253623
51  51 68.574349  1.8047036    0.6839264
52  52 41.684367 13.0672050    9.0888493
53  53 69.879593 17.3516599    4.2564834
54  54  9.817277 23.8189949   17.8341328
55  55  4.379280 20.6629714   17.4118109
56  56 28.610378  1.1479621   10.0558055
57  57 21.460016 22.9760018   15.0086341
58  58 27.601656  4.3883804   10.8936852
59  59 57.230504 12.8050000    5.3068180
60  60 26.397282  7.7639305   11.0243252
61  61 39.010480  0.7676801    8.7639297
62  62 11.527487  6.7396220   14.6003016
63  63 17.684287  8.2780091   13.8021255
64  64 19.935253  6.0764383   11.6320723
65  65 42.310040  8.4123966    7.4630017
66  66  1.119154 19.5503583   17.7101910
```

```
67   67 23.276821 14.3066349   13.2663850
68   68 14.965816 12.8691532   15.3003824
69   69 30.663350 16.6632038   11.7531562
70   70 22.925183 24.5987873   16.1991180
71   71 59.770308  8.7007692    5.6856819
72   72 70.456061 12.7407109    2.9059872
73   73 21.750385 18.8575107   14.1798442
74   74 49.360686 23.2094802    9.4660260
75   75  4.487242 23.4190000   18.7669334
76   76 14.693269 17.8743936   16.3479313
77   77 40.611628 25.9937106   12.0466432
78   78  8.764197  3.8990404   13.3730055
79   79 56.725412 16.1809774    6.7357709
80   80 60.551149 18.0652019    6.5167036
81   81 64.384893 10.5295309    4.5668945
82   82 20.262798 11.1787830   12.3790734
83   83 30.520086 12.4334774   11.1407355
84   84 30.461542 29.8608106   14.3299759
85   85  3.705894 21.4941886   17.8776920
86   86 15.082469 16.6152001   14.9359167
87   87 30.997842 29.0164264   13.7579867
88   88 14.625411  6.7983462   14.4789601
89   89 28.103061 14.7313505   12.7945548
90   90 34.680241  8.6381050    8.3317181
91   91  6.947463 26.1056583   18.6897979
92   92 26.860662 16.7194132   13.4876889
93   93 41.019323 12.9991873   10.2648899
94   94 31.932738 28.1828730   13.1737410
95   95 69.877147 26.3311967    6.0724855
96   96 63.029854 22.2471319    6.3567192
97   97 21.349999 12.8558535   13.7996830
98   98  3.811338 17.7237882   18.4623513
99   99 67.514106 26.9553999    6.5826851
100 100 12.580893 16.9897825   14.8556848
101 101 61.232197  4.6135505    4.3616062
102 102 64.332304 29.4237378    7.7073976
103 103 31.872439 16.2418978   10.5260293
104 104 11.936562  3.4865672   14.3914484
105 105 52.360268 19.8410769    7.8163886
106 106 22.516638  7.6951729   11.6647648
107 107 49.764822  3.5729441    6.2645744
108 108 22.792636  9.8168286   10.9117655
109 109 49.748748 20.4918307    8.7001979
110 110 68.204122  2.1929445    1.9664482
```

```
111 111 15.185101 14.5203621   14.5878253
112 112 58.046901 15.7135850    6.5593358
113 113 69.499688 23.1678748    5.7536536
114 114  2.616135  4.3190804   14.8517766
115 115  2.136343 25.8401303   19.2426878
116 116 25.771571 28.5403473   14.4293484
117 117 11.615646  6.3455289   14.4114830
118 118 17.197456 20.7334559   15.1981843
119 119 27.681662 18.6370085   11.5075951
120 120  9.648096  8.3700383   15.5763683
121 121 65.621956 14.1066235    4.0442469
122 122 46.556228 26.8937498   10.5427902
123 123 48.300472 12.0117870    7.2992774
124 124 15.141292  7.8937497   13.6191300
125 125 71.579351 18.9378855    4.0907648
126 126  4.681250 10.9882819   15.8685828
127 127 22.476723 25.9723233   14.8303440
128 128 49.296201  5.0887881    6.0737923
129 129 17.648135 27.1925636   15.1389680
130 130  9.413778  8.6085378   13.5250712
131 131 64.915668 10.8893906    3.7743880
132 132  3.950367  6.3595796   14.6357693
133 133 67.342457 16.6226193    3.7444984
134 134 71.238955 25.1941518    6.0015227
135 135 70.323878 26.2334743    5.2357312
136 136 28.424901 20.2084486   13.0038423
137 137 73.713732 14.1016522    2.9890677
138 138 56.058032  8.2061159    5.5239310
139 139 21.588199  2.5734949   11.3742898
140 140 16.276161  3.4634491   13.5987548
141 141 26.988690 15.8833260   12.9216726
142 142 11.326814  9.2306549   14.5243489
143 143 55.580584  7.6713074    5.7268270
144 144 50.603802 28.9184712   10.6355905
145 145 60.401739 24.8321410    7.5726142
146 146 71.486751 21.6373684    3.6684273
147 147 37.978507 13.9453443   10.1067938
148 148 48.692115 23.5640881    9.5990941
149 149 40.016400  7.8116349    7.6581870
150 150 32.148553  0.9690843    8.8739847
151 151 12.318283 26.8908059   17.7585948
152 152 31.659667 21.3995087   12.0931244
153 153 55.841893  2.2487162    4.5002273
154 154 28.826953 11.4098666   12.8126371
```

```
155 155 55.472287  5.2674378    4.9163834
156 156 54.354034 16.3415202    7.5917973
157 157 70.331699 15.0588637    3.0374318
158 158 54.062609 20.3888136    8.0511951
159 159 59.575645 24.7926381    8.5199279
160 160  2.818204 23.2461405   18.4829598
161 161 30.460335  2.5256544    8.9365829
162 162 22.343450 23.2046425   15.9085493
163 163 14.696886  9.8316822   13.2582035
164 164 70.902815 29.9140032    6.3350215
165 165 35.335113  9.1475020    9.7518942
166 166 72.173766 15.2736628    2.8283567
167 167 44.698217 10.0025889    7.4494126
168 168 70.361366 20.3399150    3.8971099
169 169  7.619084 26.6615229   18.6881304
170 170 29.673634  2.3967956    9.3879078
171 171 28.485683 12.6628036   11.2860570
172 172 67.423291 28.6574311    5.8094945
173 173 10.145069 11.3175197   14.8048065
174 174 59.989904 25.4558391    7.4743720
175 175 38.155015 20.1042221   11.5622808
176 176 15.466010 11.5711484   13.3774300
177 177 73.767713 16.1513316    2.3548085
178 178 31.179629  5.5684413    9.8617288
179 179  5.201611  4.3599032   15.6431142
180 180 50.249614  4.1290591    5.5419066
181 181 60.940141 21.8644959    5.4159174
182 182 20.068674 11.9294173   12.6884954
183 183 41.211215  4.1514402    7.5901660
184 184 72.394856  7.5198372    1.8701100
185 185 10.610969 19.3015155   16.7460356
186 186 45.579836 20.6168515    9.7984834
187 187 29.658506 12.1518990   12.8178071
188 188 40.056854 16.5064944    9.3926393
189 189  5.510300 17.8842193   16.0139208
190 190 32.056529 12.4794809   11.5360652
191 191 46.842870 27.3216486   10.2505847
192 192 42.425007 10.9547393    8.8282361
193 193 31.212374  7.7973828    9.7753859
194 194 13.176628  9.9874669   14.5477545
195 195 33.779739  0.9653903    6.9442975
196 196 70.690083 22.7107707    2.7084606
197 197 60.284951 15.1081402    4.9479908
198 198 16.003605 19.8941489   15.3662877
```

```
199 199 39.677219 10.2721672   9.5436557
200 200 12.885185 25.2101825   16.2725863
201 201 35.023450 22.6640373   12.5158362
202 202 10.343753 27.6468493   17.4485160
203 203 20.640893 15.3841384   14.5572879
204 204 63.238037 20.5047412    5.5609216
205 205 23.984565  7.6121169   11.5562573
206 206 44.014897  6.5796621    8.5037463
207 207 67.127924  5.8203635    2.5511506
208 208 36.538050  2.0528831    8.4950339
209 209  7.831481 26.8269709   17.5604514
210 210 40.395401  7.0274602    6.7390807
211 211 16.249914 28.3617369   17.3545771
212 212 47.584661 29.4683524   11.5999032
213 213 15.481362 18.8152012   14.9407560
214 214 70.085196 10.3826455    2.2392169
215 215  1.330485 28.7937440   20.4534962
216 216 61.542692 12.8374854    4.9734613
217 217 23.097771  0.9073611   10.6947889
218 218 65.069917  1.4876797    2.1856508
219 219 71.014542 15.2819055    4.0768235
220 220 64.065982 28.4724464    7.2862524
221 221 61.152760  2.1900536    3.5229763
222 222 22.672368 28.7353995   16.7786346
223 223 49.728740  3.4945424    5.0730746
224 224 35.480794 15.8501562   11.3729972
225 225 59.461885 18.8075341    5.6906489
226 226 31.697593 23.1906637   12.5378281
227 227 62.772108 15.2398175    4.9345783
228 228 58.668542 19.5857407    6.6620753
229 229 25.254140  7.1689519   12.0465147
230 230 22.722701  8.2882507   11.2963162
231 231  1.616922  1.4584368   16.3351186
232 232 10.353648 16.0131718   14.0886540
233 233 44.721586  7.2730788    7.6450248
234 234 29.224098 27.3570273   13.6443726
235 235 66.111593 18.6293941    4.5900481
236 236 46.728488  8.4995461    7.8255359
237 237 39.172427 17.5563764   10.7150869
238 238  9.937617 22.3426031   16.3846780
239 239 61.393845 22.9195492    6.8381847
240 240 21.608228  9.3488357   13.9959895
241 241  6.283610 20.1753293   17.5981661
242 242 61.436136 24.4659138    7.6206629
```

```
243 243  1.257841 15.1309519   16.7368633
244 244  4.699863  9.8599366   15.5694794
245 245 54.938223 13.0462165    6.2944988
246 246  7.749030  8.3870192   13.2815287
247 247 49.563424 16.2590213    8.1434851
248 248 29.731403 23.3339016   12.9964758
249 249 45.716267 23.8302604    9.2300743
250 250 47.537321 20.0742953    8.9378932
 [ reached 'max' / getOption("max.print") -- omitted 248 rows ]
> dim(heart.data)
[1] 498   4
> summary(heart.data)
      X          biking        smoking       heart.disease
 Min.  :  1.0   Min.  : 1.119   Min.  : 0.5259   Min.  : 0.5519
 1st Qu.:125.2   1st Qu.:20.205   1st Qu.: 8.2798   1st Qu.: 6.5137
 Median :249.5   Median :35.824   Median :15.8146   Median :10.3853
 Mean   :249.5   Mean   :37.788   Mean   :15.4350   Mean   :10.1745
 3rd Qu.:373.8   3rd Qu.:57.853   3rd Qu.:22.5689   3rd Qu.:13.7240
 Max.   :498.0   Max.   :74.907   Max.   :29.9467   Max.   :20.4535
>
> #Assumptions
> cor(heart.data$biking, heart.data$smoking)   #independent predictors
[1] 0.01513618
> hist(heart.data$heart.disease)    #normally distributed
> plot(heart.disease ~ biking, data=heart.data)     #linearity x ~ y
> plot(heart.disease ~ smoking, data=heart.data)    #linearity x ~ y
> #Homoscedasticity or homogeneity of variance will be checked after model building
>
> #Linear Regression Analysis
> heart.disease.lm <- lm(heart.disease ~ biking+smoking, data=heart.data)
> summary(heart.disease.lm)

Call:
lm(formula = heart.disease ~ biking + smoking, data = heart.data)

Residuals:
   Min     1Q  Median     3Q    Max
-2.1789 -0.4463  0.0362  0.4422  1.9331

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 14.984658   0.080137  186.99  <2e-16 ***
biking      -0.200133   0.001366 -146.53  <2e-16 ***
smoking      0.178334   0.003539   50.39  <2e-16 ***
```

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.654 on 495 degrees of freedom
Multiple R-squared:  0.9796,     Adjusted R-squared:  0.9795
F-statistic: 1.19e+04 on 2 and 495 DF,  p-value: < 2.2e-16

```
> par(mfrow=c(2, 2))
> plot(heart.disease.lm)   #Homoscedasticity, Residuals normally distributed
> par(mfrow=c(1, 1))
>
> #Visualize results
> #1.    Create a new dataframe with the information needed to plot the model
> plotting.data <- expand.grid(biking=seq(min(heart.data$biking), max(heart.data$biking),
length.out=30),
+                   smoking=c(min(heart.data$smoking), mean(heart.data$smoking),
max(heart.data$smoking)))
> #2.    Predict the values of heart disease based on your linear model
> plotting.data$predicted.y <- predict.lm(heart.disease.lm, newdata = plotting.data)
> #3.    Round the smoking numbers to two decimals
> plotting.data$smoking <- round(plotting.data$smoking, digits = 2)
> #4.    Change the 'smoking' variable into a factor
> plotting.data$smoking <- as.factor(plotting.data$smoking)
> #5.    Plot the original data
> heart.plot <- ggplot(heart.data, aes(x=biking, y=heart.disease)) + geom_point()
> heart.plot
> #6.    Add the regression lines
> heart.plot <- heart.plot +
+   geom_line(data=plotting.data, aes(x=biking, y=predicted.y, color=smoking), size=1.25)
> heart.plot
> #7.    Make the graph ready for publication
> heart.plot <-
+   heart.plot +
+   theme_bw() +
+   labs(title = "Rates of heart disease (% of population) \n as a function of biking to work and smoking",
+       x = "Biking to work (% of population)",
+       y = "Heart disease (% of population)",
+       color = "Smoking \n (% of population)")
> heart.plot
>
> heart.plot + annotate(geom="text", x=30, y=1.75, label=" = 15 + (-0.2*biking) + (0.178*smoking)")
```

## Day 7 – R Programming

**Class Assessment – Property Price Prediction**

```
> # In this case study we build a linear regression model
> # We use the model to predict our test data
> # We check the model performance using
> # RMSE metric
> # We demonstrate tests for autocorrelation & heteroskedasticity
> # We demonstrate VIF to detect multicollinearity
>
> library(ggplot2)
> library(dplyr)
> library(broom)
> library(ggpubr)
> # Set your working directory.
> setwd("C:/zubeda/PGA02_Zubu/R Programming/Models")
>
> #Importing Data
> propertytrainData <- read.csv("PropertyTrainData.csv")
> head(propertytrainData)
  Price Sea   Area Elevation Sewer Days Flood Distance
1  4.5  1  138.4      10 3000 -103   0    0.3
2 10.6  1   52.0       4    0 -103   0    2.5
3  1.7  0   16.1       0 2640  -98   1   10.3
4  5.0  0 1695.2       1 3500  -93   0   14.0
5  5.0  0  845.0       1 1000  -92   1   14.0
6  3.3  1    6.9       2 10000 -86   0    0.0
> dim(propertytrainData)
[1] 31  8
> summary(propertytrainData)
    Price          Sea           Area          Elevation        Sewer           Days          Flood          Distance
 Min.   : 1.70  Min.   :0.0000  Min.   :   6.90  Min.   : 0.000  Min.   :    0  Min.   :-103.00  Min.   :0.0000  Min.   : 0.000
 1st Qu.: 5.35  1st Qu.:0.0000  1st Qu.:  20.35  1st Qu.: 2.000  1st Qu.:    0  1st Qu.: -63.50  1st Qu.:0.0000  1st Qu.: 0.850
 Median :11.70  Median :1.0000  Median :  51.40  Median : 4.000  Median :  900  Median : -59.00  Median :0.0000  Median : 4.900
 Mean   :11.95  Mean   :0.6129  Mean   : 139.97  Mean   : 4.645  Mean   : 1981  Mean   : -58.65  Mean   :0.1613  Mean   : 5.132
 3rd Qu.:16.05  3rd Qu.:1.0000  3rd Qu.: 104.10  3rd Qu.: 7.000  3rd Qu.: 3450  3rd Qu.: -51.00  3rd Qu.:0.0000  3rd Qu.: 5.500
 Max.   :37.20  Max.   :1.0000  Max.   :1695.20  Max.   :20.000  Max.   :10000  Max.   :  -4.00  Max.   :1.0000  Max.   :16.500
> propertytestData <- read.csv("PropertyTestData.csv")
> head(propertytestData)
```

```
  Price Sea Area Elevation Sewer Days Flood Distance
1  12  1 1472     20 4811 -36  1     8
2   5  0 1301      1 4070 -79  0     1
3  12  1  39      17 1200 -40  1     4
4  36  0  7       18 3240 -46  0    12
5   2  0 357       7 5619 -88  1     1
6   9  1 686      12 5056 -39  1    12
> dim(propertytestData)
[1] 31  8
> summary(propertytestData)
    Price         Sea           Area         Elevation       Sewer          Days         Flood         Distance
 Min.   : 2.00  Min.   :0.0000  Min.   :  7.0  Min.   : 1.00  Min.   :   0  Min.   :-96.00  Min.   :0.0000  Min.
 : 0.000
 1st Qu.: 6.50  1st Qu.:0.0000  1st Qu.: 325.5  1st Qu.: 7.00  1st Qu.:1142  1st Qu.:-77.00  1st
 Qu.:0.0000  1st Qu.: 2.500
 Median : 9.00  Median :0.0000  Median : 657.0  Median :12.00  Median :2814  Median :-49.00
 Median :1.0000  Median : 6.000
 Mean   :11.76  Mean   :0.4839  Mean   : 733.7  Mean   :11.32  Mean   :2819  Mean   :-55.39  Mean
 :0.6129  Mean   : 6.145
 3rd Qu.:16.90  3rd Qu.:1.0000  3rd Qu.:1166.5  3rd Qu.:16.50  3rd Qu.:4273  3rd Qu.:-39.50  3rd
 Qu.:1.0000  3rd Qu.: 9.500
 Max.   :36.00  Max.   :1.0000  Max.   :1556.0  Max.   :20.00  Max.   :5775  Max.   : -7.00  Max.   :1.0000
 Max.   :12.000
>
> #EDA ~ Assumptions
> #Check for normality of dependent variable
> hist(propertytrainData$Price)
> shapiro.test(propertytrainData$Price)   #Price is not normality distributed


        Shapiro-Wilk normality test

data:  propertytrainData$Price
W = 0.90607, p-value = 0.01025

> logPrice <- log(propertytrainData$Price)
> hist(logPrice)
> shapiro.test(logPrice)   #Price is now normality distributed (p-value > 0.05)


        Shapiro-Wilk normality test

data:  logPrice
W = 0.95854, p-value = 0.2668

> propertytrainData$Logprice <- logPrice
```

```
> dim(propertytrainData)
[1] 31  9
> names(propertytrainData)
[1] "Price"    "Sea"    "Area"    "Elevation" "Sewer"    "Days"    "Flood"    "Distance" "Logprice"
> summary(propertytrainData)
   Price       Sea        Area        Elevation      Sewer        Days         Flood        Distance
Logprice
 Min.  : 1.70  Min.  :0.0000  Min.  :  6.90  Min.  : 0.000  Min.  :   0  Min.  :-103.00  Min.  :0.0000
Min.  : 0.000  Min.  :0.5306
 1st Qu.: 5.35  1st Qu.:0.0000  1st Qu.: 20.35  1st Qu.: 2.000  1st Qu.:   0  1st Qu.: -63.50  1st
Qu.:0.0000  1st Qu.: 0.850  1st Qu.:1.6750
 Median :11.70  Median :1.0000  Median : 51.40  Median : 4.000  Median :  900  Median : -59.00
Median :0.0000  Median : 4.900  Median :2.4596
 Mean  :11.95  Mean  :0.6129  Mean  : 139.97  Mean  : 4.645  Mean  : 1981  Mean  : -58.65  Mean
:0.1613  Mean  : 5.132  Mean  :2.2594
 3rd Qu.:16.05  3rd Qu.:1.0000  3rd Qu.: 104.10  3rd Qu.: 7.000  3rd Qu.: 3450  3rd Qu.: -51.00  3rd
Qu.:0.0000  3rd Qu.: 5.500  3rd Qu.:2.7746
 Max.  :37.20  Max.  :1.0000  Max.  :1695.20  Max.  :20.000  Max.  :10000  Max.  :  -4.00  Max.
:1.0000  Max.  :16.500  Max.  :3.6163
> #Check Correlation
> cor(propertytrainData[, -1])  #Read all rows, skip 1st column. Use a negative index to skip the column
from the left
          Sea       Area Elevation    Sewer      Days      Flood   Distance   Logprice
Sea      1.00000000 -0.33944108  0.47517280 -0.05004423 -0.36983885 -0.55180357 -0.74220440 -
0.04416109
Area     -0.33944108  1.00000000 -0.20945610  0.05338087 -0.34946290  0.10890203  0.55694587 -
0.22024015
Elevation  0.47517280 -0.20945610  1.00000000 -0.35940756 -0.05650853 -0.37308077 -0.36246039
0.43335591
Sewer    -0.05004423  0.05338087 -0.35940756  1.00000000 -0.15149473 -0.11305464 -0.15865389 -
0.46759131
Days     -0.36983885 -0.34946290 -0.05650853 -0.15149473  1.00000000  0.01536084  0.04438251
0.62016026
Flood    -0.55180357  0.10890203 -0.37308077 -0.11305464  0.01536084  1.00000000  0.42330840 -
0.40729809
Distance -0.74220440  0.55694587 -0.36246039 -0.15865389  0.04438251  0.42330840  1.00000000
0.06587072
Logprice -0.04416109 -0.22024015  0.43335591 -0.46759131  0.62016026 -0.40729809  0.06587072
1.00000000
> cormat <- round(cor(propertytrainData[, -1]), 3)
> cormat
          Sea   Area Elevation  Sewer   Days  Flood Distance Logprice
Sea      1.000 -0.339    0.475 -0.050 -0.370 -0.552   -0.742   -0.044
Area     -0.339  1.000   -0.209  0.053 -0.349  0.109    0.557   -0.220
```

```
Elevation  0.475 -0.209    1.000 -0.359 -0.057 -0.373  -0.362   0.433
Sewer   -0.050 0.053   -0.359 1.000 -0.151 -0.113  -0.159  -0.468
Days    -0.370 -0.349   -0.057 -0.151 1.000 0.015   0.044   0.620
Flood   -0.552 0.109   -0.373 -0.113 0.015 1.000   0.423  -0.407
Distance -0.742 0.557   -0.362 -0.159 0.044 0.423   1.000   0.066
Logprice -0.044 -0.220   0.433 -0.468 0.620 -0.407   0.066   1.000
> write.csv(cormat, "corrmatrix.csv")
>
> #install.packages("GGally")
> library(GGally)
> GGally::ggpairs(propertytrainData[, -1])   #Pairplot - histogram & lineplot
> # Note we see variables which are faily correlated with log(price)
> # are Elevation, Sewer, Date & Flood
> # Note we also see some of the IVs are also correlated with each other
> # like Elevation & Sea, Distance & Sea, Area & Distance and so on
>
> # Another way to better appreciate the relationship
> # between variables is to look at scatter plot
> # Lets plot log(price) and Date
> plot(propertytrainData$Logprice, propertytrainData$Days)
> plot(propertytrainData$Logprice, propertytrainData$Area)
> #Linear Regression Analysis
> reg_model <- lm(Logprice ~ ., data = propertytrainData[, -1])   # . refers to rest all variables
> summary(reg_model)

Call:
lm(formula = Logprice ~ ., data = propertytrainData[, -1])

Residuals:
    Min     1Q  Median    3Q    Max
-0.41605 -0.22833  0.01037  0.22662  0.63418

Coefficients:
          Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.099e+00  2.815e-01  11.006 1.22e-10 ***
Sea       -1.596e-01  2.685e-01  -0.594 0.558013
Area       -2.578e-04  2.574e-04  -1.002 0.327001
Elevation   5.053e-02  1.754e-02   2.880 0.008448 **
Sewer      -8.338e-05  3.066e-05  -2.720 0.012214 *
Days        1.479e-02  3.577e-03   4.135 0.000403 ***
Flood      -9.819e-01  2.198e-01  -4.468 0.000175 ***
Distance    4.889e-02  2.496e-02   1.958 0.062407 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Residual standard error: 0.3258 on 23 degrees of freedom
Multiple R-squared: 0.8416,     Adjusted R-squared: 0.7934
F-statistic: 17.46 on 7 and 23 DF,  p-value: 8.112e-08
> #White Spaces - Not significant, . - Poorly Significant(0.05-0.1), * - Average Significance(0.01-0.05), ** -
Significant(0.001-0.01), *** - Highly Significant(0-0.001) (in Coefficients)
>
> #New models without Sea, Area
> reg_model1 <- lm(Logprice ~ Area+Days+Distance+Flood+Elevation+Sewer, data = propertytrainData[,
-1])
> summary(reg_model1)

Call:
lm(formula = Logprice ~ Area + Days + Distance + Flood + Elevation +
   Sewer, data = propertytrainData[, -1])

Residuals:
   Min    1Q  Median    3Q    Max
-0.37796 -0.22920 -0.01371  0.20334  0.68359

Coefficients:
          Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.006e+00  2.310e-01  13.009 2.31e-12 ***
Area       -2.256e-04  2.482e-04  -0.909  0.37240
Days        1.614e-02  2.729e-03   5.914 4.22e-06 ***
Distance    5.826e-02  1.908e-02   3.053 0.00547 **
Flood      -9.154e-01  1.866e-01  -4.906 5.27e-05 ***
Elevation   4.992e-02  1.727e-02   2.890 0.00806 **
Sewer      -7.653e-05  2.802e-05  -2.731 0.01164 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3214 on 24 degrees of freedom
Multiple R-squared: 0.8392,     Adjusted R-squared: 0.799
F-statistic: 20.87 on 6 and 24 DF,  p-value: 1.978e-08

> reg_model2 <- lm(Logprice ~ Days+Distance+Flood+Elevation+Sewer, data = propertytrainData[, -1])
> summary(reg_model2)

Call:
lm(formula = Logprice ~ Days + Distance + Flood + Elevation +
   Sewer, data = propertytrainData[, -1])

Residuals:

```
    Min    1Q  Median    3Q    Max
-0.38511 -0.25256 -0.01794  0.20994  0.72640


Coefficients:
          Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.089e+00  2.115e-01  14.603 9.60e-14 ***
Days        1.724e-02  2.435e-03   7.080 2.02e-07 ***
Distance    4.784e-02  1.521e-02   3.147 0.00424 **
Flood      -8.835e-01  1.826e-01  -4.838 5.66e-05 ***
Elevation   5.048e-02  1.720e-02   2.934 0.00707 **
Sewer      -7.859e-05  2.783e-05  -2.824 0.00919 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


Residual standard error: 0.3203 on 25 degrees of freedom
Multiple R-squared: 0.8336,    Adjusted R-squared: 0.8003
F-statistic: 25.05 on 5 and 25 DF,  p-value: 5.474e-09


>
> library(car)
> car::vif(reg_model)   #Lower the value, relevant is the variable
    Sea    Area Elevation   Sewer    Days   Flood Distance
 4.995597 2.003925 1.649759 1.635122 2.174889 1.907942 3.623612
> car::vif(reg_model1)
    Area    Days Distance   Flood Elevation   Sewer
 1.915504 1.300847 2.176858 1.412978 1.644029 1.403977
> car::vif(reg_model2)
    Days Distance   Flood Elevation   Sewer
 1.043122 1.391449 1.363245 1.641901 1.394802
> #property.train1 <- subset(property.train, select=-c(Sea, Distance))
> #head(property.train1)
>
> par(mfrow=c(2, 2))
> plot(reg_model2)   #Homoscedasticity, Residuals normally distributed
> par(mfrow=c(1, 1))
> # Check residual vs fitted plot to check Heteroscedasticity
> # If there is absolutely no heteroscedasticity, you should
> # see a completely random, equal distribution of points
> # throughout the range of X axis and a flat red line.
> # In our case,
> # the red line is slightly curved and the residuals seem to
> # increase as the fitted Y values increase.
> # So, the inference here is, heteroscedasticity exists.
> # Check the Residuals Vs Fitted Curve
```

```
>
> # Alternate Check for Breusch-Pagan Test for Heteroscedasticity;
> # Ho: Homoscedasticity (Variance of residuals is constant)
> # Ha: Heteroscedasticity
> #install.packages("lmtest")
> library(lmtest)
> lmtest::bptest(reg_model)

        studentized Breusch-Pagan test

data:  reg_model
BP = 15.953, df = 7, p-value = 0.02555

> lmtest::bptest(reg_model2)

        studentized Breusch-Pagan test

data:  reg_model2
BP = 7.7561, df = 5, p-value = 0.1702
> library(e1071)
> library(caret)
Loading required package: lattice
> # How to rectify?
> # Re-build the model with new predictors.
> # Variable transformation such as Box-Cox transformation can also be tried instead of log price
> # (Normal Distribution).
> boxcoxprice <- caret::BoxCoxTrans(propertytrainData$Price)   #Normalizing Price using BoxCox
Transformation
> print(boxcoxprice)
Box-Cox Transformation

31 data points used to estimate Lambda

Input data summary:
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.70    5.35   11.70   11.95   16.05   37.20

Largest/Smallest: 21.9
Sample Skewness: 1.03

Estimated Lambda: 0.3
> propertytrainData <- cbind(propertytrainData, Newprice=predict(boxcoxprice,
propertytrainData$Price))   #add predicted normalized price column
> head(propertytrainData)
```

```
   Price Sea   Area Elevation Sewer Days Flood Distance  Logprice  Newprice
1  4.5  1 138.4      10 3000 -103   0    0.3 1.5040774 1.9007725
2 10.6  1  52.0       4    0 -103   0    2.5 2.3608540 3.4348249
3  1.7  0  16.1       0 2640  -98   1   10.3 0.5306283 0.5751964
4  5.0  0 1695.2      1 3500  -93   0   14.0 1.6094379 2.0688553
5  5.0  0  845.0      1 1000  -92   1   14.0 1.6094379 2.0688553
6  3.3  1   6.9       2 10000 -86   0    0.0 1.1939225 1.4357282
> reg_model22 <- lm(Newprice ~ Days+Distance+Flood+Elevation+Sewer, data = propertytrainData)
> summary(reg_model22)

Call:
lm(formula = Newprice ~ Days + Distance + Flood + Elevation +
   Sewer, data = propertytrainData)

Residuals:
   Min    1Q Median    3Q    Max
-0.7648 -0.4730 -0.0573  0.4420  1.7239

Coefficients:
          Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.944e+00  4.399e-01  11.238 2.89e-11 ***
Days        3.365e-02  5.064e-03   6.644 5.81e-07 ***
Distance    9.468e-02  3.162e-02   2.994 0.006128 **
Flood      -1.619e+00  3.798e-01  -4.262 0.000252 ***
Elevation   9.871e-02  3.578e-02   2.759 0.010694 *
Sewer      -1.401e-04  5.789e-05  -2.421 0.023095 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6661 on 25 degrees of freedom
Multiple R-squared:  0.8072,    Adjusted R-squared:  0.7686
F-statistic: 20.93 on 5 and 25 DF,  p-value: 3.31e-08

> lmtest::bptest(reg_model22)

        studentized Breusch-Pagan test

data:  reg_model22
BP = 4.3918, df = 5, p-value = 0.4945

> plot(reg_model22)
Hit <Return> to see next plot:
Hit <Return> to see next plot:
Hit <Return> to see next plot:
```

Hit <Return> to see next plot:
>
> #Autocorrelation: Durbin watson Test
> #H0: No Autocorrelation, Ha: Autocorrelation present
> lmtest::dwtest(reg_model)     #p-vale: 0.74

        Durbin-Watson test

data:  reg_model
DW = 2.4103, p-value = 0.7465
alternative hypothesis: true autocorrelation is greater than 0

> lmtest::dwtest(reg_model1)    #p-vale: 0.73

        Durbin-Watson test

data:  reg_model1
DW = 2.3327, p-value = 0.7328
alternative hypothesis: true autocorrelation is greater than 0

> lmtest::dwtest(reg_model2)    #p-vale: 0.67

        Durbin-Watson test

data:  reg_model2
DW = 2.265, p-value = 0.6728
alternative hypothesis: true autocorrelation is greater than 0

>
> #Fitting the model
> predicted_salesPrice <- predict(reg_model2, newdata = propertytrainData)
> predicted_salesPrice
       1         2         3         4         5         6         7         8         9        10        11        12        13
1.5963200 1.6344573 0.8009036 1.9305666 1.2607307 0.9211005 2.1182612 2.1872227 2.9673940
1.5482586 1.5654990 1.8520608 1.9548896
      14        15        16        17        18        19        20        21        22        23        24        25        26
2.6858571 2.7772486 2.7411216 2.7002102 2.6088186 2.8468662 2.7145633 2.6327763 2.6588934
2.5390465 1.8259669 1.7226961 3.0131863
      27        28        29        30        31
2.5844959 2.8667676 3.0611836 2.4123110 3.3130961
> propertytestData$PredictedPrice <- exp(predicted_salesPrice)
> write.csv(propertytestData, "predictedresult.csv")
>
> cor(propertytestData$Price, propertytestData$PredictedPrice)

[1] 0.2907895
> plot(propertytestData$Price, propertytestData$PredictedPrice)
>
> #install.packages("Metrics")
> library(Metrics)
> Metrics::rmse(propertytestData$Price, propertytestData$PredictedPrice)
[1] 8.158264

## Day 8 – R Programming

### Admission Classification – Logistic Regression
> df <- read.csv("https://stats.idre.ucla.edu/stat/data/binary.csv")
> head(df)
  admit gre  gpa rank
1     0 380 3.61   3
2     1 660 3.67   3
3     1 800 4.00   1
4     1 640 3.19   4
5     0 520 2.93   4
6     1 760 3.00   2
> str(df)
'data.frame':     400 obs. of  4 variables:
 $ admit: int  0 1 1 1 0 1 1 0 1 0 ...
 $ gre  : int  380 660 800 640 520 760 560 400 540 700 ...
 $ gpa  : num  3.61 3.67 4 3.19 2.93 3 2.98 3.08 3.39 3.92 ...
 $ rank : int  3 3 1 4 4 2 1 2 3 2 ...
> dim(df)
[1] 400   4
> edit(df)
   admit gre  gpa rank
1      0 380 3.61   3
2      1 660 3.67   3
3      1 800 4.00   1
4      1 640 3.19   4
5      0 520 2.93   4
6      1 760 3.00   2
7      1 560 2.98   1
8      0 400 3.08   2
9      1 540 3.39   3
10     0 700 3.92   2
11     0 800 4.00   4
12     0 440 3.22   1
13     1 760 4.00   1
14     0 700 3.08   2
15     1 700 4.00   1

| 16 | 0 | 480 | 3.44 | 3 |
|----|---|-----|------|---|
| 17 | 0 | 780 | 3.87 | 4 |
| 18 | 0 | 360 | 2.56 | 3 |
| 19 | 0 | 800 | 3.75 | 2 |
| 20 | 1 | 540 | 3.81 | 1 |
| 21 | 0 | 500 | 3.17 | 3 |
| 22 | 1 | 660 | 3.63 | 2 |
| 23 | 0 | 600 | 2.82 | 4 |
| 24 | 0 | 680 | 3.19 | 4 |
| 25 | 1 | 760 | 3.35 | 2 |
| 26 | 1 | 800 | 3.66 | 1 |
| 27 | 1 | 620 | 3.61 | 1 |
| 28 | 1 | 520 | 3.74 | 4 |
| 29 | 1 | 780 | 3.22 | 2 |
| 30 | 0 | 520 | 3.29 | 1 |
| 31 | 0 | 540 | 3.78 | 4 |
| 32 | 0 | 760 | 3.35 | 3 |
| 33 | 0 | 600 | 3.40 | 3 |
| 34 | 1 | 800 | 4.00 | 3 |
| 35 | 0 | 360 | 3.14 | 1 |
| 36 | 0 | 400 | 3.05 | 2 |
| 37 | 0 | 580 | 3.25 | 1 |
| 38 | 0 | 520 | 2.90 | 3 |
| 39 | 1 | 500 | 3.13 | 2 |
| 40 | 1 | 520 | 2.68 | 3 |
| 41 | 0 | 560 | 2.42 | 2 |
| 42 | 1 | 580 | 3.32 | 2 |
| 43 | 1 | 600 | 3.15 | 2 |
| 44 | 0 | 500 | 3.31 | 3 |
| 45 | 0 | 700 | 2.94 | 2 |
| 46 | 1 | 460 | 3.45 | 3 |
| 47 | 1 | 580 | 3.46 | 2 |
| 48 | 0 | 500 | 2.97 | 4 |
| 49 | 0 | 440 | 2.48 | 4 |
| 50 | 0 | 400 | 3.35 | 3 |
| 51 | 0 | 640 | 3.86 | 3 |
| 52 | 0 | 440 | 3.13 | 4 |
| 53 | 0 | 740 | 3.37 | 4 |
| 54 | 1 | 680 | 3.27 | 2 |
| 55 | 0 | 660 | 3.34 | 3 |
| 56 | 1 | 740 | 4.00 | 3 |
| 57 | 0 | 560 | 3.19 | 3 |
| 58 | 0 | 380 | 2.94 | 3 |
| 59 | 0 | 400 | 3.65 | 2 |

| 60  | 0 600 2.82 | 4 |
| 61  | 1 620 3.18 | 2 |
| 62  | 0 560 3.32 | 4 |
| 63  | 0 640 3.67 | 3 |
| 64  | 1 680 3.85 | 3 |
| 65  | 0 580 4.00 | 3 |
| 66  | 0 600 3.59 | 2 |
| 67  | 0 740 3.62 | 4 |
| 68  | 0 620 3.30 | 1 |
| 69  | 0 580 3.69 | 1 |
| 70  | 0 800 3.73 | 1 |
| 71  | 0 640 4.00 | 3 |
| 72  | 0 300 2.92 | 4 |
| 73  | 0 480 3.39 | 4 |
| 74  | 0 580 4.00 | 2 |
| 75  | 0 720 3.45 | 4 |
| 76  | 0 720 4.00 | 3 |
| 77  | 0 560 3.36 | 3 |
| 78  | 1 800 4.00 | 3 |
| 79  | 0 540 3.12 | 1 |
| 80  | 1 620 4.00 | 1 |
| 81  | 0 700 2.90 | 4 |
| 82  | 0 620 3.07 | 2 |
| 83  | 0 500 2.71 | 2 |
| 84  | 0 380 2.91 | 4 |
| 85  | 1 500 3.60 | 3 |
| 86  | 0 520 2.98 | 2 |
| 87  | 0 600 3.32 | 2 |
| 88  | 0 600 3.48 | 2 |
| 89  | 0 700 3.28 | 1 |
| 90  | 1 660 4.00 | 2 |
| 91  | 0 700 3.83 | 2 |
| 92  | 1 720 3.64 | 1 |
| 93  | 0 800 3.90 | 2 |
| 94  | 0 580 2.93 | 2 |
| 95  | 1 660 3.44 | 2 |
| 96  | 0 660 3.33 | 2 |
| 97  | 0 640 3.52 | 4 |
| 98  | 0 480 3.57 | 2 |
| 99  | 0 700 2.88 | 2 |
| 100 | 0 400 3.31 | 3 |
| 101 | 0 340 3.15 | 3 |
| 102 | 0 580 3.57 | 3 |
| 103 | 0 380 3.33 | 4 |

| | | | | |
|---|---|---|---|---|
| 104 | 0 | 540 | 3.94 | 3 |
| 105 | 1 | 660 | 3.95 | 2 |
| 106 | 1 | 740 | 2.97 | 2 |
| 107 | 1 | 700 | 3.56 | 1 |
| 108 | 0 | 480 | 3.13 | 2 |
| 109 | 0 | 400 | 2.93 | 3 |
| 110 | 0 | 480 | 3.45 | 2 |
| 111 | 0 | 680 | 3.08 | 4 |
| 112 | 0 | 420 | 3.41 | 4 |
| 113 | 0 | 360 | 3.00 | 3 |
| 114 | 0 | 600 | 3.22 | 1 |
| 115 | 0 | 720 | 3.84 | 3 |
| 116 | 0 | 620 | 3.99 | 3 |
| 117 | 1 | 440 | 3.45 | 2 |
| 118 | 0 | 700 | 3.72 | 2 |
| 119 | 1 | 800 | 3.70 | 1 |
| 120 | 0 | 340 | 2.92 | 3 |
| 121 | 1 | 520 | 3.74 | 2 |
| 122 | 1 | 480 | 2.67 | 2 |
| 123 | 0 | 520 | 2.85 | 3 |
| 124 | 0 | 500 | 2.98 | 3 |
| 125 | 0 | 720 | 3.88 | 3 |
| 126 | 0 | 540 | 3.38 | 4 |
| 127 | 1 | 600 | 3.54 | 1 |
| 128 | 0 | 740 | 3.74 | 4 |
| 129 | 0 | 540 | 3.19 | 2 |
| 130 | 0 | 460 | 3.15 | 4 |
| 131 | 1 | 620 | 3.17 | 2 |
| 132 | 0 | 640 | 2.79 | 2 |
| 133 | 0 | 580 | 3.40 | 2 |
| 134 | 0 | 500 | 3.08 | 3 |
| 135 | 0 | 560 | 2.95 | 2 |
| 136 | 0 | 500 | 3.57 | 3 |
| 137 | 0 | 560 | 3.33 | 4 |
| 138 | 0 | 700 | 4.00 | 3 |
| 139 | 0 | 620 | 3.40 | 2 |
| 140 | 1 | 600 | 3.58 | 1 |
| 141 | 0 | 640 | 3.93 | 2 |
| 142 | 1 | 700 | 3.52 | 4 |
| 143 | 0 | 620 | 3.94 | 4 |
| 144 | 0 | 580 | 3.40 | 3 |
| 145 | 0 | 580 | 3.40 | 4 |
| 146 | 0 | 380 | 3.43 | 3 |
| 147 | 0 | 480 | 3.40 | 2 |

| 148 | 0 | 560 | 2.71 | 3 |
|---|---|---|---|---|
| 149 | 1 | 480 | 2.91 | 1 |
| 150 | 0 | 740 | 3.31 | 1 |
| 151 | 1 | 800 | 3.74 | 1 |
| 152 | 0 | 400 | 3.38 | 2 |
| 153 | 1 | 640 | 3.94 | 2 |
| 154 | 0 | 580 | 3.46 | 3 |
| 155 | 0 | 620 | 3.69 | 3 |
| 156 | 1 | 580 | 2.86 | 4 |
| 157 | 0 | 560 | 2.52 | 2 |
| 158 | 1 | 480 | 3.58 | 1 |
| 159 | 0 | 660 | 3.49 | 2 |
| 160 | 0 | 700 | 3.82 | 3 |
| 161 | 0 | 600 | 3.13 | 2 |
| 162 | 0 | 640 | 3.50 | 2 |
| 163 | 1 | 700 | 3.56 | 2 |
| 164 | 0 | 520 | 2.73 | 2 |
| 165 | 0 | 580 | 3.30 | 2 |
| 166 | 0 | 700 | 4.00 | 1 |
| 167 | 0 | 440 | 3.24 | 4 |
| 168 | 0 | 720 | 3.77 | 3 |
| 169 | 0 | 500 | 4.00 | 3 |
| 170 | 0 | 600 | 3.62 | 3 |
| 171 | 0 | 400 | 3.51 | 3 |
| 172 | 0 | 540 | 2.81 | 3 |
| 173 | 0 | 680 | 3.48 | 3 |
| 174 | 1 | 800 | 3.43 | 2 |
| 175 | 0 | 500 | 3.53 | 4 |
| 176 | 1 | 620 | 3.37 | 2 |
| 177 | 0 | 520 | 2.62 | 2 |
| 178 | 1 | 620 | 3.23 | 3 |
| 179 | 0 | 620 | 3.33 | 3 |
| 180 | 0 | 300 | 3.01 | 3 |
| 181 | 0 | 620 | 3.78 | 3 |
| 182 | 0 | 500 | 3.88 | 4 |
| 183 | 0 | 700 | 4.00 | 2 |
| 184 | 1 | 540 | 3.84 | 2 |
| 185 | 0 | 500 | 2.79 | 4 |
| 186 | 0 | 800 | 3.60 | 2 |
| 187 | 0 | 560 | 3.61 | 3 |
| 188 | 0 | 580 | 2.88 | 2 |
| 189 | 0 | 560 | 3.07 | 2 |
| 190 | 0 | 500 | 3.35 | 2 |
| 191 | 1 | 640 | 2.94 | 2 |

| | | | |
|---|---|---|---|
| 192 | 0 800 3.54 | 3 |
| 193 | 0 640 3.76 | 3 |
| 194 | 0 380 3.59 | 4 |
| 195 | 1 600 3.47 | 2 |
| 196 | 0 560 3.59 | 2 |
| 197 | 0 660 3.07 | 3 |
| 198 | 1 400 3.23 | 4 |
| 199 | 0 600 3.63 | 3 |
| 200 | 0 580 3.77 | 4 |
| 201 | 0 800 3.31 | 3 |
| 202 | 1 580 3.20 | 2 |
| 203 | 1 700 4.00 | 1 |
| 204 | 0 420 3.92 | 4 |
| 205 | 1 600 3.89 | 1 |
| 206 | 1 780 3.80 | 3 |
| 207 | 0 740 3.54 | 1 |
| 208 | 1 640 3.63 | 1 |
| 209 | 0 540 3.16 | 3 |
| 210 | 0 580 3.50 | 2 |
| 211 | 0 740 3.34 | 4 |
| 212 | 0 580 3.02 | 2 |
| 213 | 0 460 2.87 | 2 |
| 214 | 0 640 3.38 | 3 |
| 215 | 1 600 3.56 | 2 |
| 216 | 1 660 2.91 | 3 |
| 217 | 0 340 2.90 | 1 |
| 218 | 1 460 3.64 | 1 |
| 219 | 0 460 2.98 | 1 |
| 220 | 1 560 3.59 | 2 |
| 221 | 0 540 3.28 | 3 |
| 222 | 0 680 3.99 | 3 |
| 223 | 1 480 3.02 | 1 |
| 224 | 0 800 3.47 | 3 |
| 225 | 0 800 2.90 | 2 |
| 226 | 1 720 3.50 | 3 |
| 227 | 0 620 3.58 | 2 |
| 228 | 0 540 3.02 | 4 |
| 229 | 0 480 3.43 | 2 |
| 230 | 1 720 3.42 | 2 |
| 231 | 0 580 3.29 | 4 |
| 232 | 0 600 3.28 | 3 |
| 233 | 0 380 3.38 | 2 |
| 234 | 0 420 2.67 | 3 |
| 235 | 1 800 3.53 | 1 |

```
236    0 620 3.05    2
237    1 660 3.49    2
238    0 480 4.00    2
239    0 500 2.86    4
240    0 700 3.45    3
241    0 440 2.76    2
242    1 520 3.81    1
243    1 680 2.96    3
244    0 620 3.22    2
245    0 540 3.04    1
246    0 800 3.91    3
247    0 680 3.34    2
248    0 440 3.17    2
249    0 680 3.64    3
250    0 640 3.73    3
 [ reached 'max' / getOption("max.print") -- omitted 150 rows ]
> sum(is.na(df))
[1] 0
> summary(df)
    admit           gre           gpa           rank
 Min.   :0.0000   Min.   :220.0   Min.   :2.260   Min.   :1.000
 1st Qu.:0.0000   1st Qu.:520.0   1st Qu.:3.130   1st Qu.:2.000
 Median :0.0000   Median :580.0   Median :3.395   Median :2.000
 Mean   :0.3175   Mean   :587.7   Mean   :3.390   Mean   :2.485
 3rd Qu.:1.0000   3rd Qu.:660.0   3rd Qu.:3.670   3rd Qu.:3.000
 Max.   :1.0000   Max.   :800.0   Max.   :4.000   Max.   :4.000
> #Mean < 0.5 means more rejection of students for admission then acceptance
> xtabs(~ admit + rank, data = df)   #Frequency table
     rank
admit  1  2  3  4
    0 28 97 93 55
    1 33 54 28 12
> df$rank <- as.factor(df$rank)
> logit <- glm(admit ~ gre+gpa+rank, data=df, family="binomial")
> summary(logit)

Call:
glm(formula = admit ~ gre + gpa + rank, family = "binomial",
    data = df)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-1.6268  -0.8662  -0.6388  1.1490  2.0790
```

Coefficients:

|  | Estimate | Std. Error | z value | Pr(>|z|) |  |
|---|---|---|---|---|---|
| (Intercept) | -3.989979 | 1.139951 | -3.500 | 0.000465 | *** |
| gre | 0.002264 | 0.001094 | 2.070 | 0.038465 | * |
| gpa | 0.804038 | 0.331819 | 2.423 | 0.015388 | * |
| rank2 | -0.675443 | 0.316490 | -2.134 | 0.032829 | * |
| rank3 | -1.340204 | 0.345306 | -3.881 | 0.000104 | *** |
| rank4 | -1.551464 | 0.417832 | -3.713 | 0.000205 | *** |

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 499.98  on 399  degrees of freedom
Residual deviance: 458.52  on 394  degrees of freedom
AIC: 470.52

Number of Fisher Scoring iterations: 4

```
> #Predicting raw data
> x <- data.frame(gre=790, gpa=3.8, rank=as.factor(1))
> p <- predict(logit, x)
> p
    1
0.85426
> x <- data.frame(gre=600, gpa=3.0, rank=as.factor(3))
> p <- predict(logit, x)
> p
     1
-1.559415
```

## Automatic/Manual Car Classification – Logistic Regression

> #The in-built data set "mtcars" describes different models of a car with their various engine specifications. In "mtcars" data set, the transmission mode (automatic or manual) is described by the column am which is a binary value (0 or 1). We can create a logistic regression model between the columns "am" and 3 other columns - hp, wt and cyl.

> mtcars

|  | mpg | cyl | disp | hp | drat | wt | qsec | vs | am | gear | carb |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Mazda RX4 | 21.0 | 6 | 160.0 | 110 | 3.90 | 2.620 | 16.46 | 0 | 1 | 4 | 4 |
| Mazda RX4 Wag | 21.0 | 6 | 160.0 | 110 | 3.90 | 2.875 | 17.02 | 0 | 1 | 4 | 4 |
| Datsun 710 | 22.8 | 4 | 108.0 | 93 | 3.85 | 2.320 | 18.61 | 1 | 1 | 4 | 1 |
| Hornet 4 Drive | 21.4 | 6 | 258.0 | 110 | 3.08 | 3.215 | 19.44 | 1 | 0 | 3 | 1 |
| Hornet Sportabout | 18.7 | 8 | 360.0 | 175 | 3.15 | 3.440 | 17.02 | 0 | 0 | 3 | 2 |
| Valiant | 18.1 | 6 | 225.0 | 105 | 2.76 | 3.460 | 20.22 | 1 | 0 | 3 | 1 |
| Duster 360 | 14.3 | 8 | 360.0 | 245 | 3.21 | 3.570 | 15.84 | 0 | 0 | 3 | 4 |

```
Merc 240D          24.4  4 146.7  62 3.69 3.190 20.00  1 0    4    2
Merc 230           22.8  4 140.8  95 3.92 3.150 22.90  1 0    4    2
Merc 280           19.2  6 167.6 123 3.92 3.440 18.30  1 0    4    4
Merc 280C          17.8  6 167.6 123 3.92 3.440 18.90  1 0    4    4
Merc 450SE         16.4  8 275.8 180 3.07 4.070 17.40  0 0    3    3
Merc 450SL         17.3  8 275.8 180 3.07 3.730 17.60  0 0    3    3
Merc 450SLC        15.2  8 275.8 180 3.07 3.780 18.00  0 0    3    3
Cadillac Fleetwood  10.4  8 472.0 205 2.93 5.250 17.98  0 0    3    4
Lincoln Continental 10.4  8 460.0 215 3.00 5.424 17.82  0 0    3    4
Chrysler Imperial  14.7  8 440.0 230 3.23 5.345 17.42  0 0    3    4
Fiat 128           32.4  4 78.7  66 4.08 2.200 19.47  1 1    4    1
Honda Civic        30.4  4 75.7  52 4.93 1.615 18.52  1 1    4    2
Toyota Corolla     33.9  4 71.1  65 4.22 1.835 19.90  1 1    4    1
Toyota Corona      21.5  4 120.1  97 3.70 2.465 20.01  1 0    3    1
Dodge Challenger   15.5  8 318.0 150 2.76 3.520 16.87  0 0    3    2
AMC Javelin        15.2  8 304.0 150 3.15 3.435 17.30  0 0    3    2
Camaro Z28         13.3  8 350.0 245 3.73 3.840 15.41  0 0    3    4
Pontiac Firebird   19.2  8 400.0 175 3.08 3.845 17.05  0 0    3    2
Fiat X1-9          27.3  4 79.0  66 4.08 1.935 18.90  1 1    4    1
Porsche 914-2      26.0  4 120.3  91 4.43 2.140 16.70  0 1    5    2
Lotus Europa       30.4  4 95.1 113 3.77 1.513 16.90  1 1    5    2
Ford Pantera L     15.8  8 351.0 264 4.22 3.170 14.50  0 1    5    4
Ferrari Dino       19.7  6 145.0 175 3.62 2.770 15.50  0 1    5    6
Maserati Bora      15.0  8 301.0 335 3.54 3.570 14.60  0 1    5    8
Volvo 142E         21.4  4 121.0 109 4.11 2.780 18.60  1 1    4    2
> str(mtcars)
'data.frame':    32 obs. of  11 variables:
 $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
 $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
 $ disp: num  160 160 108 258 360 ...
 $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
 $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
 $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
 $ qsec: num  16.5 17 18.6 19.4 17 ...
 $ vs  : num  0 0 1 1 0 1 0 1 1 1 ...
 $ am  : num  1 1 1 0 0 0 0 0 0 0 ...
 $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
 $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
> dim(mtcars)
[1] 32 11
> sum(is.na(mtcars))
[1] 0
> summary(mtcars)
    mpg          cyl          disp          hp          drat          wt
```

```
 Min.   :10.40   Min.   :4.000   Min.   : 71.1   Min.   : 52.0   Min.   :2.760   Min.   :1.513
 1st Qu.:15.43   1st Qu.:4.000   1st Qu.:120.8   1st Qu.: 96.5   1st Qu.:3.080   1st Qu.:2.581
 Median :19.20   Median :6.000   Median :196.3   Median :123.0   Median :3.695   Median :3.325
 Mean   :20.09   Mean   :6.188   Mean   :230.7   Mean   :146.7   Mean   :3.597   Mean   :3.217
 3rd Qu.:22.80   3rd Qu.:8.000   3rd Qu.:326.0   3rd Qu.:180.0   3rd Qu.:3.920   3rd Qu.:3.610
 Max.   :33.90   Max.   :8.000   Max.   :472.0   Max.   :335.0   Max.   :4.930   Max.   :5.424
      qsec            vs              am             gear            carb
 Min.   :14.50   Min.   :0.0000   Min.   :0.0000   Min.   :3.000   Min.   :1.000
 1st Qu.:16.89   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:2.000
 Median :17.71   Median :0.0000   Median :0.0000   Median :4.000   Median :2.000
 Mean   :17.85   Mean   :0.4375   Mean   :0.4062   Mean   :3.688   Mean   :2.812
 3rd Qu.:18.90   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:4.000   3rd Qu.:4.000
 Max.   :22.90   Max.   :1.0000   Max.   :1.0000   Max.   :5.000   Max.   :8.000
> xtabs(~ am + cyl, data=mtcars)
   cyl
am   4  6  8
  0  3  4 12
  1  8  3  2
> table(mtcars$am, mtcars$cyl)

    4  6  8
  0  3  4 12
  1  8  3  2
>
> cars1 <- mtcars[, c("cyl", "hp", "wt", "am")]
> head(cars1)
                  cyl  hp   wt am
Mazda RX4          6 110 2.620  1
Mazda RX4 Wag      6 110 2.875  1
Datsun 710         4  93 2.320  1
Hornet 4 Drive     6 110 3.215  0
Hornet Sportabout  8 175 3.440  0
Valiant            6 105 3.460  0
>
> logit <- glm(formula=am ~ cyl+hp+wt, data=cars1, family="binomial")
> summary(logit)

Call:
glm(formula = am ~ cyl + hp + wt, family = "binomial", data = cars1)

Deviance Residuals:
    Min      1Q   Median      3Q     Max
-2.17272 -0.14907 -0.01464  0.14116  1.27641
```

Coefficients:

| | Estimate | Std. Error | z value | Pr(>|z|) | |
|---|---|---|---|---|---|
| (Intercept) | 19.70288 | 8.11637 | 2.428 | 0.0152 | * |
| cyl | 0.48760 | 1.07162 | 0.455 | 0.6491 | |
| hp | 0.03259 | 0.01886 | 1.728 | 0.0840 | . |
| wt | -9.14947 | 4.15332 | -2.203 | 0.0276 | * |

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

   Null deviance: 43.2297  on 31  degrees of freedom
Residual deviance:  9.8415  on 28  degrees of freedom
AIC: 17.841

Number of Fisher Scoring iterations: 8

```
> x <- data.frame(cyl=6, hp=110, wt=3.200)
> p <- predict(logit, x)
> p
        1
-3.064753
```