**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

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")