R for Biologists-Day_5

Maruf Ahmed Bhuiyan

7/10/2020

Set working directory according to your choice.

```
setwd("~/Desktop/cBALST_R/Day 5- 10 July 2020")
```

1.Create a variable "x" that has value 5. And another variable "y" that has value 3. Add them and store in z. Print z in the console.

```
x <- 5
y <- 3
z <- x + y
z
```

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

2.Create a vector called "tamim" with runs in five matches. The runs are 40, 50, 100, 20, 10, 35, 40. Get the average run of tamin in the tournament

There are values for seven matches.

```
tamim <- c(40, 50, 100, 20, 10, 35, 40)
tamim_average <- sum(tamim) / 7
tamim_average
```

```
## [1] 42.14286
```

```
# Alternative approach
tamim_average <- sum(tamim) / length(tamim)
tamim_average
```

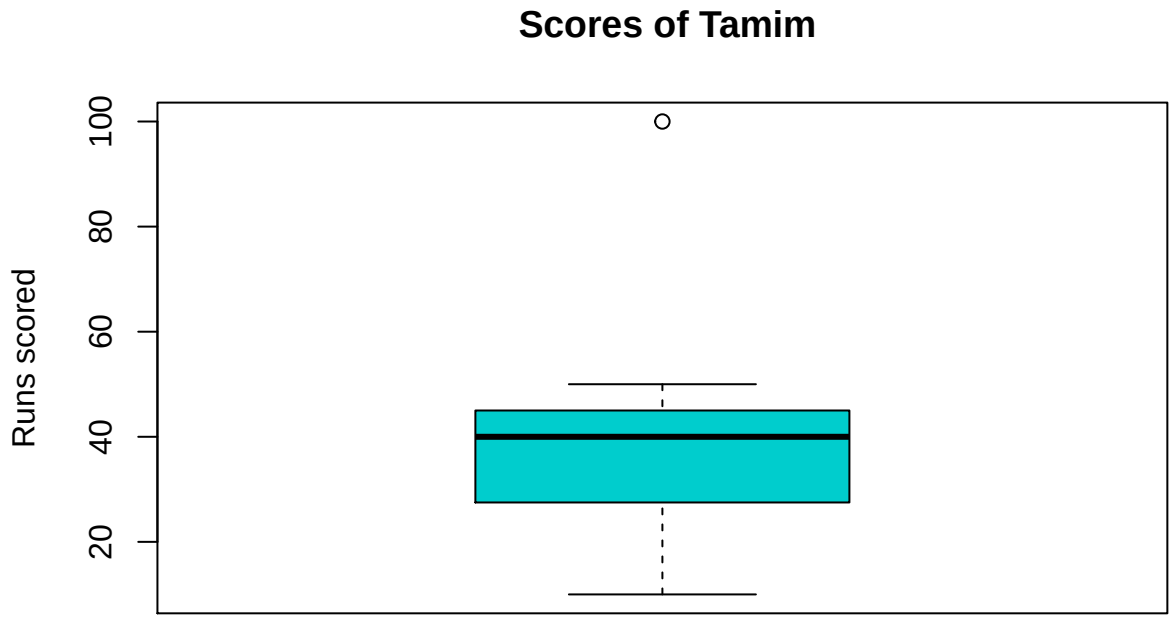
```
## [1] 42.14286
```

```r
# We can also calculate the mean which is the arithmetic average.
mean(tamim)
```

```
## [1] 42.14286
```

3.Create a boxplot of variable tamim. It should like like the following. Which one is the outlier here?

```r
boxplot(tamim, main = " Scores of Tamim", ylab = "Runs scored", col = "cyan3")
```




```r
# Outlier is any value that is 1.5 times the IQR value above or
# below the Q3 or Q1 correspondingly. The score 100 is the outlier here.

print(paste("The outlier is:", 100))
```

```
## [1] "The outlier is: 100"
```

4.Create matrix with number 1 to 100. The matrix should contain 4 columns and 25 rows.

```
mat <- matrix(1:100, nrow = 25, ncol = 4)
mat
```

```
##       [,1] [,2] [,3] [,4]
##  [1,]    1   26   51   76
##  [2,]    2   27   52   77
##  [3,]    3   28   53   78
##  [4,]    4   29   54   79
##  [5,]    5   30   55   80
##  [6,]    6   31   56   81
##  [7,]    7   32   57   82
##  [8,]    8   33   58   83
##  [9,]    9   34   59   84
## [10,]   10   35   60   85
## [11,]   11   36   61   86
## [12,]   12   37   62   87
## [13,]   13   38   63   88
## [14,]   14   39   64   89
## [15,]   15   40   65   90
## [16,]   16   41   66   91
## [17,]   17   42   67   92
## [18,]   18   43   68   93
## [19,]   19   44   69   94
## [20,]   20   45   70   95
## [21,]   21   46   71   96
## [22,]   22   47   72   97
## [23,]   23   48   73   98
## [24,]   24   49   74   99
## [25,]   25   50   75  100
```

5.Give the 4 column names. “one, two, three, four”. It should look like following

```
colnames(mat) <- c("one", "two", "three", "four")
mat
```

```
##       one two three four
##  [1,]   1  26    51   76
##  [2,]   2  27    52   77
##  [3,]   3  28    53   78
##  [4,]   4  29    54   79
##  [5,]   5  30    55   80
##  [6,]   6  31    56   81
##  [7,]   7  32    57   82
##  [8,]   8  33    58   83
```

```
##  [9,]   9  34    59   84
## [10,]  10  35    60   85
## [11,]  11  36    61   86
## [12,]  12  37    62   87
## [13,]  13  38    63   88
## [14,]  14  39    64   89
## [15,]  15  40    65   90
## [16,]  16  41    66   91
## [17,]  17  42    67   92
## [18,]  18  43    68   93
## [19,]  19  44    69   94
## [20,]  20  45    70   95
## [21,]  21  46    71   96
## [22,]  22  47    72   97
## [23,]  23  48    73   98
## [24,]  24  49    74   99
## [25,]  25  50    75  100
```

```
# Alternate approach

colname <- c("one", "two", "three", "four")
colnames(mat) <- colname
mat
```

```
##       one two three four
##  [1,]   1  26    51   76
##  [2,]   2  27    52   77
##  [3,]   3  28    53   78
##  [4,]   4  29    54   79
##  [5,]   5  30    55   80
##  [6,]   6  31    56   81
##  [7,]   7  32    57   82
##  [8,]   8  33    58   83
##  [9,]   9  34    59   84
## [10,]  10  35    60   85
## [11,]  11  36    61   86
## [12,]  12  37    62   87
## [13,]  13  38    63   88
## [14,]  14  39    64   89
## [15,]  15  40    65   90
## [16,]  16  41    66   91
## [17,]  17  42    67   92
## [18,]  18  43    68   93
## [19,]  19  44    69   94
## [20,]  20  45    70   95
## [21,]  21  46    71   96
## [22,]  22  47    72   97
## [23,]  23  48    73   98
## [24,]  24  49    74   99
## [25,]  25  50    75  100
```

6.Create a data frame that looks like following.

```
Name <- c("William", "Emma", "Sofia", "Markus", "Edward", "Thomas")
Region <- c("East", "North", "East", "South", "West", "West")
Sales <- c(50000, 52000, 90000, 34000, 42000, 72000)
Expenses <- c(42000, 43000, 50000, 44000, 38000, 39000)

df <- data.frame(Name, Region, Sales, Expenses)
df
```

```
##      Name Region Sales Expenses
## 1 William   East 50000    42000
## 2    Emma  North 52000    43000
## 3   Sofia   East 90000    50000
## 4  Markus  South 34000    44000
## 5  Edward   West 42000    38000
## 6  Thomas   West 72000    39000
```

7.Create the following list:

```
mother <- "Veronique"
father <- "Michel"
sisters <- c("Alicia", "Monica")
sisters_age <- c(12, 22)

lst <- list(mother, father, sisters, sisters_age)
lst
```

```
## [[1]]
## [1] "Veronique"
##
## [[2]]
## [1] "Michel"
##
## [[3]]
## [1] "Alicia" "Monica"
##
## [[4]]
## [1] 12 22
```

```
names(lst) <- c("mother", "father", "sisters", "sisters_age")
lst
```

```
## $mother
## [1] "Veronique"
##
## $father
## [1] "Michel"
##
```

```
## $sisters
## [1] "Alicia" "Monica"
##
## $sisters_age
## [1] 12 22
```

8.Write an if else condition where it says, if x is greater than 0 then print positive, if less than 0 print negative and if x is 0 than print zero and if anything else print please type a new number. Check what the value shows if x <- 5 and x <- 0

```
x <- 5

if (x > 0){
  print("Positive")
} else if (x < 0){
  print("Negative")
} else if (x == 0){
  print("Zero")
} else
  print("Please, type a new number")
```

```
## [1] "Positive"
```

```
x <- 0

if (x > 0){
  print("Positive")
} else if (x < 0){
  print("Negative")
} else if (x == 0){
  print("Zero")
} else
  print("Please, type a new number")
```

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

9.Create the following data frame:

```
DF1 <- data.frame(c1 = c(1,5,14,23,54),
                  c2 = c(9,15,85,3,42),
                  c3 = c(9,7,42,87,16))
DF1
```

```
##   c1 c2 c3
## 1  1  9  9
```

```
## 2  5 15  7
## 3 14 85 42
## 4 23  3 87
## 5 54 42 16
```

```
mat1 <- as.matrix(DF1)
mat1
```

```
##      c1 c2 c3
## [1,]  1  9  9
## [2,]  5 15  7
## [3,] 14 85 42
## [4,] 23  3 87
## [5,] 54 42 16
```

10.Use for loop to get the following output

```
digits <- c(1:10)

for (i in digits){
  print(paste("the year is,", i))
}
```

```
## [1] "the year is, 1"
## [1] "the year is, 2"
## [1] "the year is, 3"
## [1] "the year is, 4"
## [1] "the year is, 5"
## [1] "the year is, 6"
## [1] "the year is, 7"
## [1] "the year is, 8"
## [1] "the year is, 9"
## [1] "the year is, 10"
```

11.Install Bioconductor in R. install Deseq2 package in R. Check the following

https://www.bioconductor.org/packages/release/bioc/html/DESeq2.html

```
# if (!requireNamespace("BiocManager", quietly = TRUE))
#    install.packages("BiocManager")
# BiocManager::install(version = "3.10")

# install Deseq2
# if (!requireNamespace("BiocManager", quietly = TRUE))
#    install.packages("BiocManager")
# BiocManager::install("DESeq2")
```

12.Load the inbuild "mtcars" data frame and store it as "cars". If you View the cars you can see that the first column is "mpg" and fourth column is "hp". Swap this column such that 1st column is "hp" and the fourth column is "mpg".

```
data("mtcars")
cars <- mtcars
str(cars)
```

```
## '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 ...
```

```
# View(cars)

library(dplyr, quietly = TRUE)
```

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

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

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

```
cars %>% select("hp", "cyl", "disp",
                "mpg", "drat", "wt",
                "qsec", "vs", "am",
                "gear", "carb")
```

```
##                    hp cyl  disp  mpg drat    wt  qsec vs am gear carb
## Mazda RX4         110   6 160.0 21.0 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     110   6 160.0 21.0 3.90 2.875 17.02  0  1    4    4
## Datsun 710         93   4 108.0 22.8 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    110   6 258.0 21.4 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 175   8 360.0 18.7 3.15 3.440 17.02  0  0    3    2
```

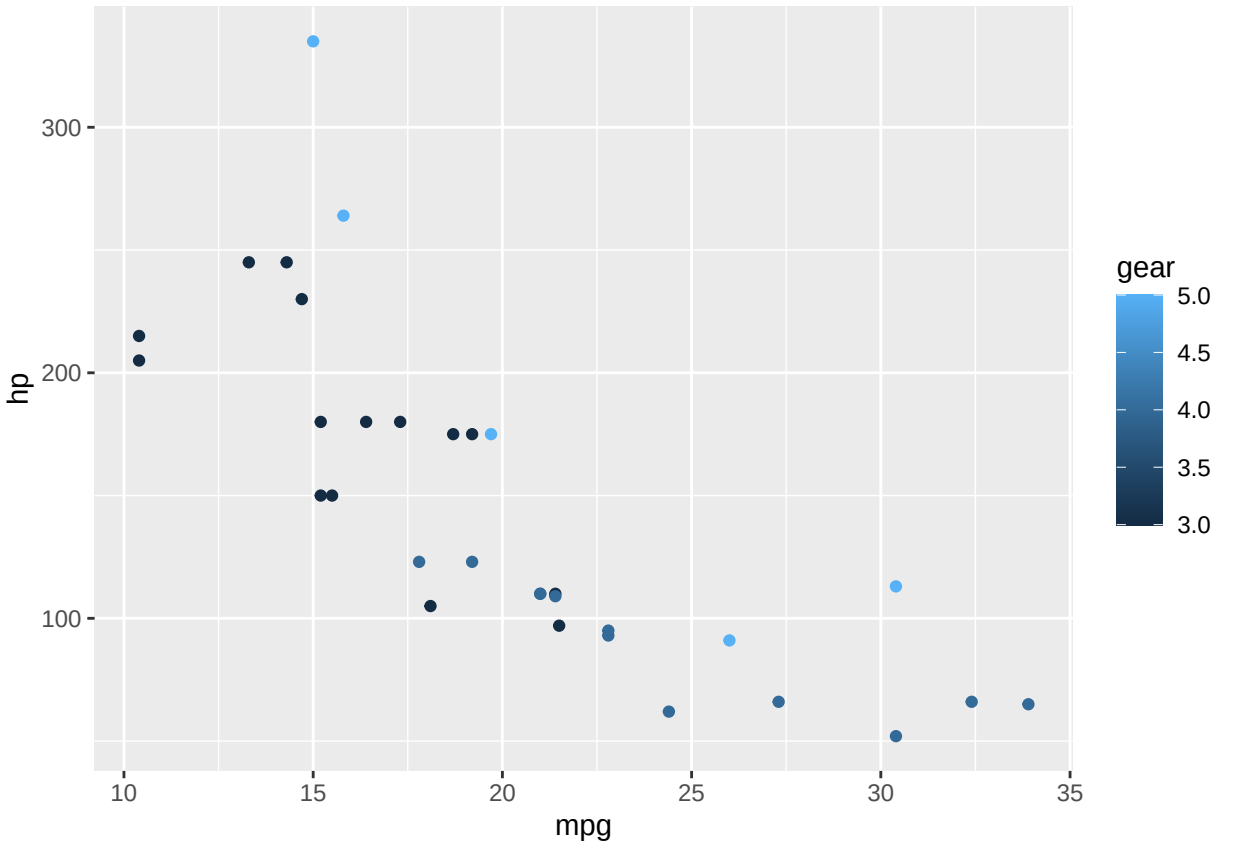
```
## Valiant             105   6 225.0 18.1 2.76 3.460 20.22  1  0    3    1
## Duster 360          245   8 360.0 14.3 3.21 3.570 15.84  0  0    3    4
## Merc 240D            62   4 146.7 24.4 3.69 3.190 20.00  1  0    4    2
## Merc 230             95   4 140.8 22.8 3.92 3.150 22.90  1  0    4    2
## Merc 280            123   6 167.6 19.2 3.92 3.440 18.30  1  0    4    4
## Merc 280C           123   6 167.6 17.8 3.92 3.440 18.90  1  0    4    4
## Merc 450SE          180   8 275.8 16.4 3.07 4.070 17.40  0  0    3    3
## Merc 450SL          180   8 275.8 17.3 3.07 3.730 17.60  0  0    3    3
## Merc 450SLC         180   8 275.8 15.2 3.07 3.780 18.00  0  0    3    3
## Cadillac Fleetwood  205   8 472.0 10.4 2.93 5.250 17.98  0  0    3    4
## Lincoln Continental 215   8 460.0 10.4 3.00 5.424 17.82  0  0    3    4
## Chrysler Imperial   230   8 440.0 14.7 3.23 5.345 17.42  0  0    3    4
## Fiat 128             66   4  78.7 32.4 4.08 2.200 19.47  1  1    4    1
## Honda Civic          52   4  75.7 30.4 4.93 1.615 18.52  1  1    4    2
## Toyota Corolla       65   4  71.1 33.9 4.22 1.835 19.90  1  1    4    1
## Toyota Corona        97   4 120.1 21.5 3.70 2.465 20.01  1  0    3    1
## Dodge Challenger    150   8 318.0 15.5 2.76 3.520 16.87  0  0    3    2
## AMC Javelin         150   8 304.0 15.2 3.15 3.435 17.30  0  0    3    2
## Camaro Z28          245   8 350.0 13.3 3.73 3.840 15.41  0  0    3    4
## Pontiac Firebird    175   8 400.0 19.2 3.08 3.845 17.05  0  0    3    2
## Fiat X1-9            66   4  79.0 27.3 4.08 1.935 18.90  1  1    4    1
## Porsche 914-2        91   4 120.3 26.0 4.43 2.140 16.70  0  1    5    2
## Lotus Europa        113   4  95.1 30.4 3.77 1.513 16.90  1  1    5    2
## Ford Pantera L      264   8 351.0 15.8 4.22 3.170 14.50  0  1    5    4
## Ferrari Dino        175   6 145.0 19.7 3.62 2.770 15.50  0  1    5    6
## Maserati Bora       335   8 301.0 15.0 3.54 3.570 14.60  0  1    5    8
## Volvo 142E          109   4 121.0 21.4 4.11 2.780 18.60  1  1    4    2
```

13.Make a scatter plot "mpg" vs "hp" and color it based on "gear". What is the difference when you use factor and when you don't

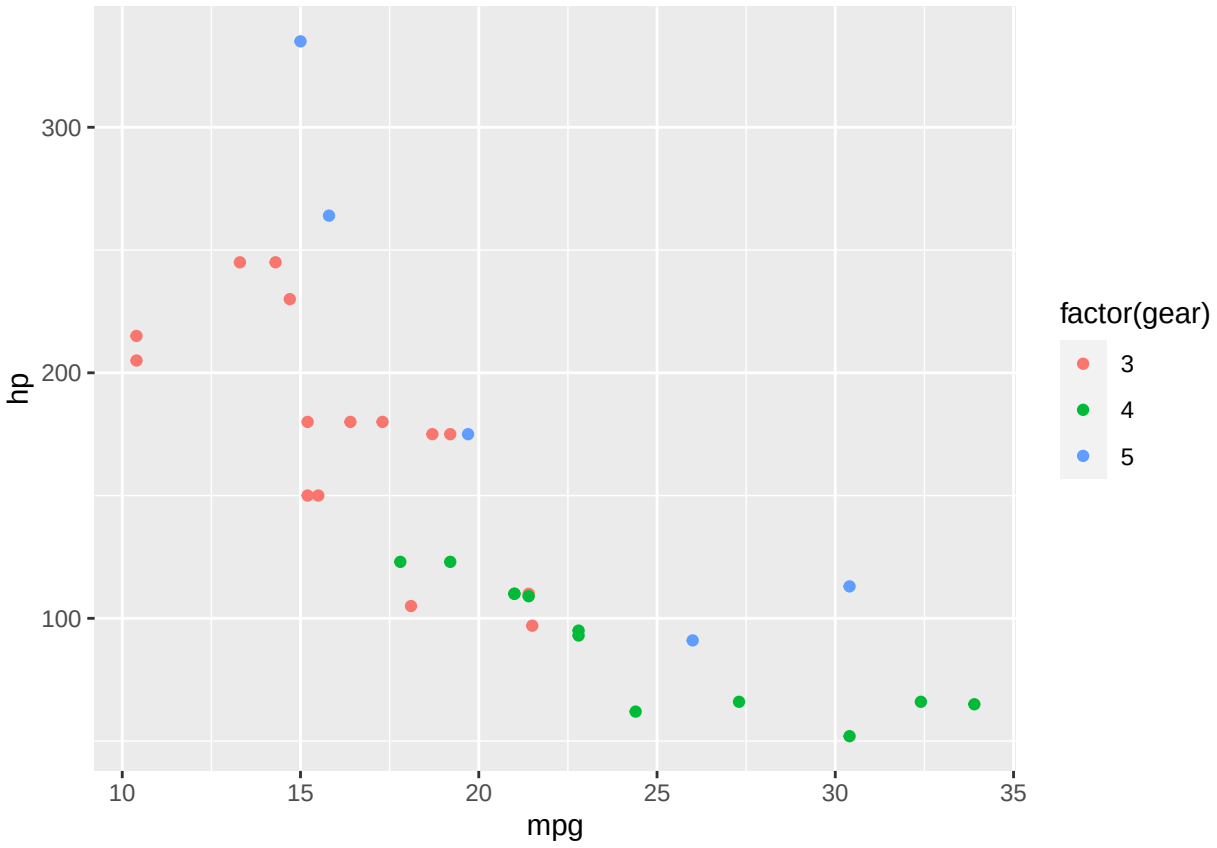
```
library(ggplot2)

head(cars)
```

```
##                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

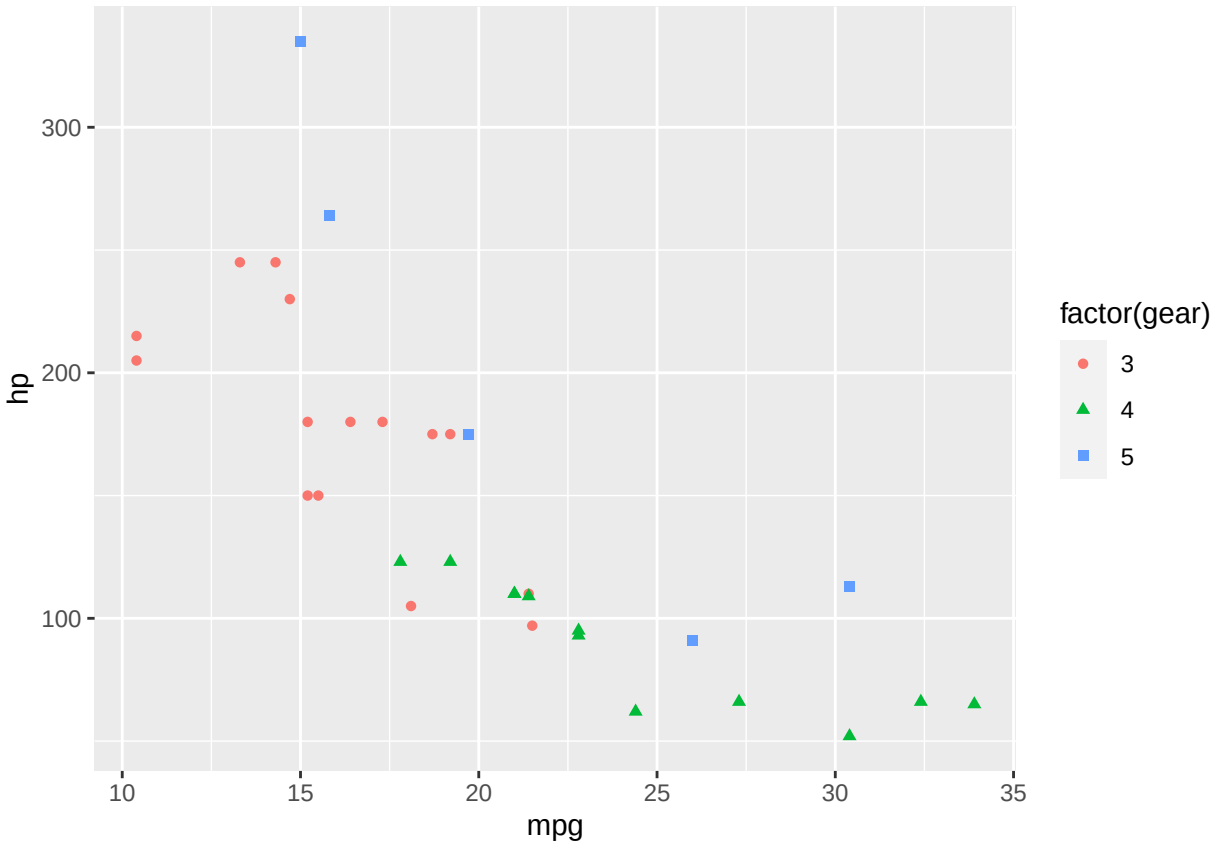
```
# Without factorization
ggplot(data = cars, aes(x = mpg , y = hp, colour= gear)) +
          geom_point()
```




```
# With factorization
ggplot(data = cars, aes(x = mpg , y = hp, colour= factor(gear))) +
          geom_point()
```




```
# We can make it more easy to understand using the shape argument
ggplot(data = cars, aes(x = mpg , y = hp,
                        colour= factor(gear),
                        shape = factor(gear))) +
          geom_point(size = 1.5)
```



14.Load the file "house.csv" in R. you can see that the "aircond" column has 0 or 1. Where 0 means no facility of air-condition and 1 means containing facility of air-condition.

```
# Location of houses.csv file

data <- read.csv("houses.csv")
str(data)
```

```
## 'data.frame':    1728 obs. of  16 variables:
##  $ X.1         : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ X           : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ price       : int  132500 181115 109000 155000 86060 120000 153000 170000 90000 122900 ...
##  $ lot_size    : num  0.09 0.92 0.19 0.41 0.11 0.68 0.4 1.21 0.83 1.94 ...
##  $ waterfront  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ age         : int  42 0 133 13 0 31 33 23 36 4 ...
##  $ land_value  : int  50000 22300 7300 18700 15000 14000 23300 14600 22200 21200 ...
##  $ construction: int  0 0 0 0 1 0 0 0 0 0 ...
##  $ air_cond    : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ fuel        : int  3 2 2 2 2 2 4 4 3 2 ...
##  $ heat        : int  4 3 3 2 2 2 3 2 4 2 ...
##  $ sewer       : int  2 2 3 2 3 2 2 2 2 1 ...
```

```
##  $ living_area : int  906 1953 1944 1944 840 1152 2752 1662 1632 1416 ...
##  $ fireplaces  : int  1 0 1 1 0 1 1 1 0 0 ...
##  $ bathrooms   : num  1 2.5 1 1.5 1 1 1.5 1.5 1.5 1.5 ...
##  $ rooms       : int  5 6 8 5 3 8 8 9 8 6 ...
```

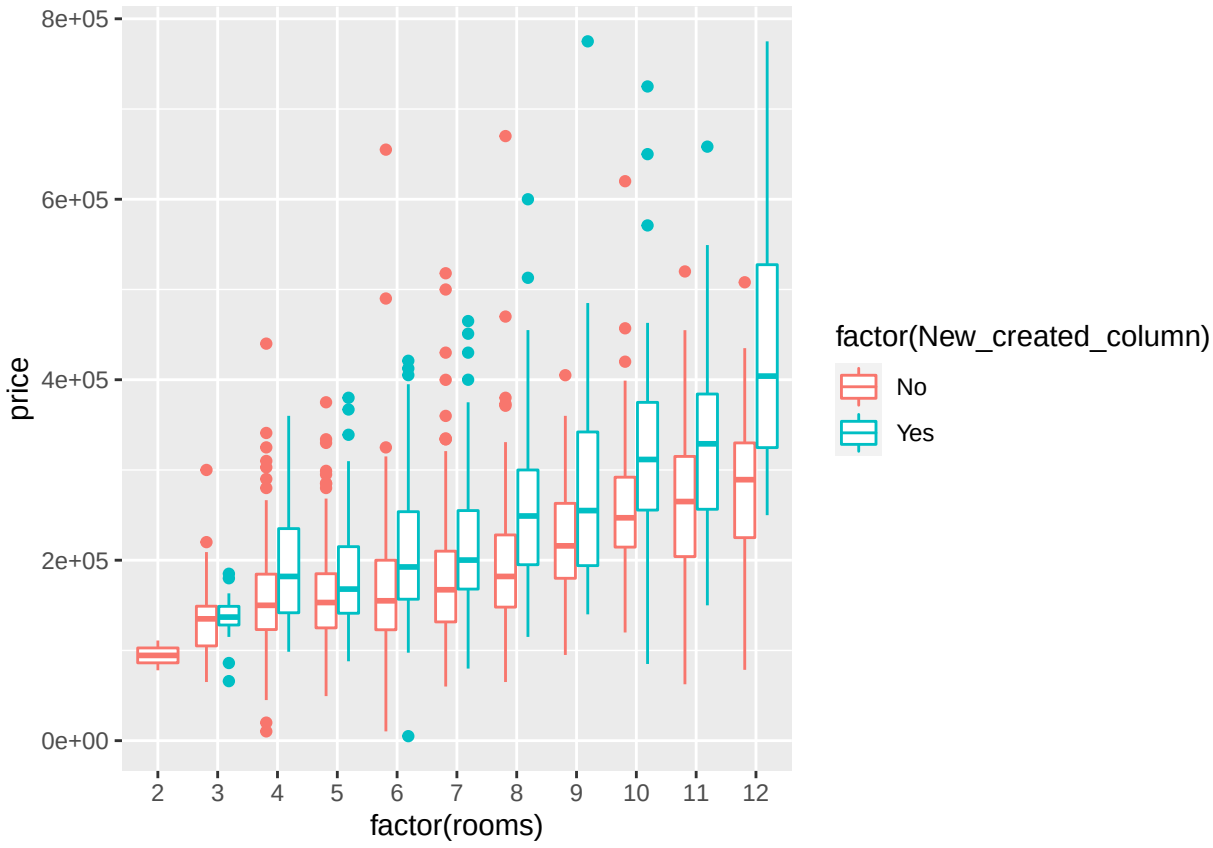
15.Make a new column titled "New_created_column" which contains "yes" or "no" value which is consistent with 0 and 1 in "air-cond" column.

```
data <- data %>% mutate( "New_created_column" =
                           ifelse(air_cond ==0, "No",
                                  ifelse(air_cond == 1, "Yes", NA)))

head(data %>% select(air_cond, New_created_column))
```

```
##   air_cond New_created_column
## 1        0                 No
## 2        0                 No
## 3        0                 No
## 4        0                 No
## 5        1                Yes
## 6        0                 No
```

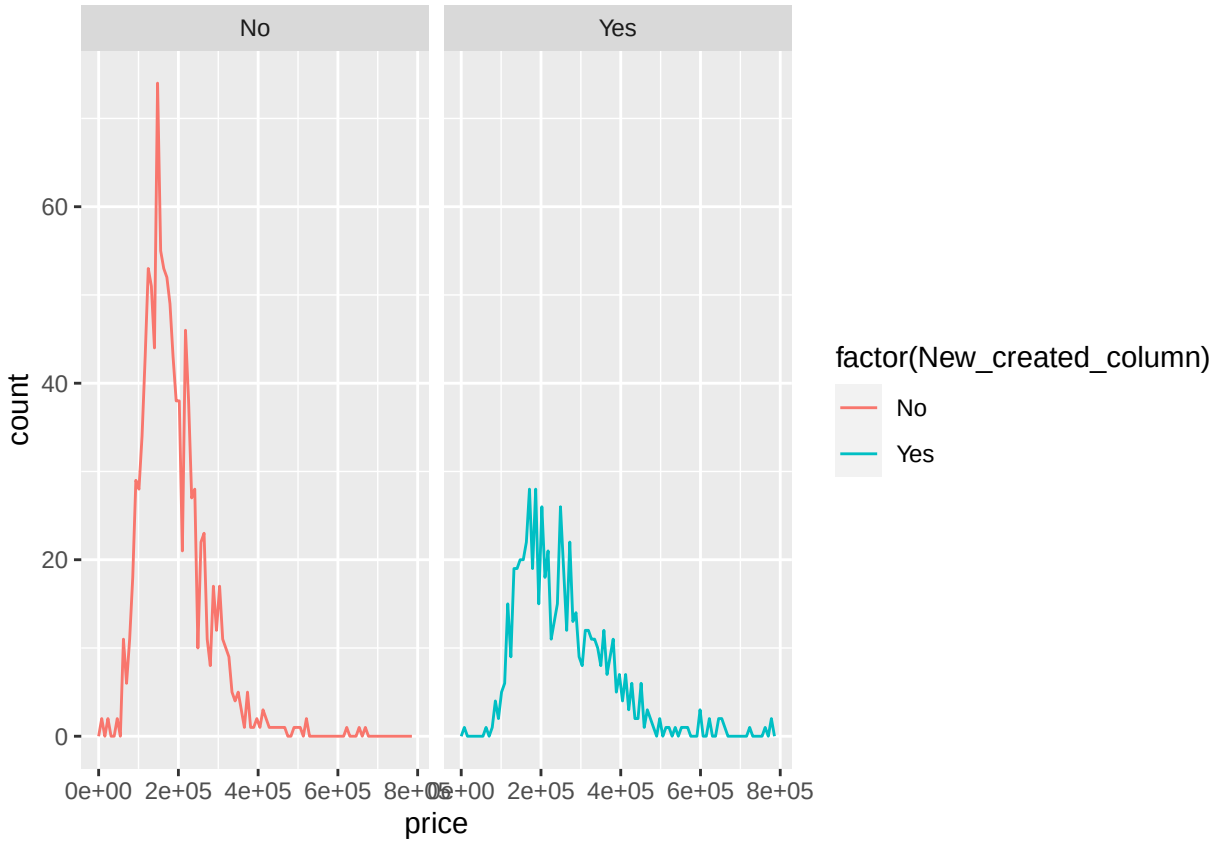
16.Make a boxplot with x axis as room number and y axis as price and compare how difference in price it makes if you have air condition facility or not. Use new_created_column.

```
ggplot(data = data, aes(x = factor(rooms),
                        y = price,
                        colour = factor(New_created_column))) +
  geom_boxplot()
```



17.Create a frequency polygon with facet_grid based on "New_created_colun

```
ggplot(data = data, aes(x = price,
                        col = factor(New_created_column)))+
  geom_freqpoly(bins = 100) +
  facet_grid( ~ New_created_column)
```



18.Say you have three genes EGFR, FOX1, TF1. You have calculated that in three cell type "cell1, cell2, cell3". You want to create a matrix where each gene will have a value for each cell type

```
cell_1 <- c(10, 100, 12)
cell_2 <- c(12, 12, 2)
cell_3 <- c(15, 20, 3)

data2 <- data.frame(cell_1, cell_2, cell_3)
data2 <- as.matrix(data2)
rownames(data2) <- c("EGFR", "TF_1", "FOX_1")
data2
```

```
##       cell_1 cell_2 cell_3
## EGFR      10     12     15
## TF_1     100     12     20
## FOX_1     12      2      3
```

19.Calculate the mean of each row and add it as the fourth column and calculate the row sum and add it as the fifth column

```
Mean <- round(rowMeans(data2),2)

Total <- rowSums(data2)

cbind(data2, Mean, Total)
```

```
##       cell_1 cell_2 cell_3  Mean Total
## EGFR      10     12     15 12.33    37
## TF_1     100     12     20 44.00   132
## FOX_1     12      2      3  5.67    17
```

20.Say you have collected some samples from 5 person. You have asked them if they believe in aliens. The response was as follows :

```
responses <- factor(c("Agree", "Agree", "Strongly Agree", "Disagree", "Agree"))

levels(responses) <- c("Strongly Agree",  "Agree", "Disagree")
responses
```

```
## [1] Strongly Agree Strongly Agree Disagree       Agree          Strongly Agree
## Levels: Strongly Agree Agree Disagree
```

21.Create the following data frame,

```
Age <- as.numeric(c(25, 31, 23, 52, 76, 49, 26))
Height <- as.numeric(c(177, 163, 190, 179, 163, 183, 164))
Weight <- as.numeric(c(57, 69, 83, 75, 70, 83, 53))
Sex <- factor(c("F", "F", "M", "M", "F", "M", "F"))

levels(Sex) <- c("F", "M")

df <- data.frame(Age, Height, Weight, Sex)
df
```

```
##   Age Height Weight Sex
## 1  25    177     57   F
## 2  31    163     69   F
## 3  23    190     83   M
## 4  52    179     75   M
## 5  76    163     70   F
## 6  49    183     83   M
## 7  26    164     53   F
```

```r
rownames(df) <- c("Alex", "Lilly", "Mark", "Oliver", "Martha", "Lucas", "Caroline")
df
```

```
##          Age Height Weight Sex
## Alex      25    177     57   F
## Lilly     31    163     69   F
## Mark      23    190     83   M
## Oliver    52    179     75   M
## Martha    76    163     70   F
## Lucas     49    183     83   M
## Caroline  26    164     53   F
```

```r
df_recoded <- df %>% mutate(Sex = recode(Sex, "F" = "M", "M" = "F"))
rownames(df_recoded) <- c("Alex", "Lilly", "Mark",
                          "Oliver", "Martha", "Lucas", "Caroline")
df_recoded
```

```
##          Age Height Weight Sex
## Alex      25    177     57   M
## Lilly     31    163     69   M
## Mark      23    190     83   F
## Oliver    52    179     75   F
## Martha    76    163     70   M
## Lucas     49    183     83   F
## Caroline  26    164     53   M
```

22.Create this data frame (make sure you import the variable Working as character and not factor).

```r
Working <- c("Yes", "No", "No", "Yes", "Yes", "No", "Yes")
class(Working)
```

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

```r
df_working <- data.frame(Working, stringsAsFactors = F)
rownames(df_working) <- c("Alex", "Lilly", "Mark", "Oliver", "Martha", "Lucas", "Caroline")


df_new <- cbind(df, df_working)
df_new
```

```
##          Age Height Weight Sex Working
## Alex      25    177     57   F     Yes
## Lilly     31    163     69   F      No
## Mark      23    190     83   M      No
## Oliver    52    179     75   M     Yes
## Martha    76    163     70   F     Yes
## Lucas     49    183     83   M      No
## Caroline  26    164     53   F     Yes
```

```r
nr <- nrow(df_new)
nc <- ncol(df_new)

print(paste("Rows:", nr, "Columns:", nc))
```

```
## [1] "Rows: 7 Columns: 5"
```

```r
lapply(df_new, class)
```

```
## $Age
## [1] "numeric"
##
## $Height
## [1] "numeric"
##
## $Weight
## [1] "numeric"
##
## $Sex
## [1] "factor"
##
## $Working
## [1] "character"
```

23.Write two string "hello" and "why am I doing this". Add this two string together and separate by ","

```r
string1 <- "Hello"
string2 <- "why am I doing this"
str3 <- "because Sohan vai told me to do so !!"

print(paste(string1, string2, str3, sep = ", "))
```

```
## [1] "Hello, why am I doing this, because Sohan vai told me to do so !!"
```

24.If name_list <- list(a = 1:200, b = "this is a string", c = "hello"). You will write a code that will add 1 to each element of the first vector of the new list. Also, add a new item z = "newItem" to the list name_list

```r
name_list <- list(a = 1:200, b = "this is a string", c = "hello")
name_list
```

```
## $a
##   [1]   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##  [19]  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
##  [37]  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
##  [55]  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
##  [73]  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
##  [91]  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108
## [109] 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
## [127] 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
## [145] 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162
## [163] 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180
## [181] 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198
## [199] 199 200
##
## $b
## [1] "this is a string"
##
## $c
## [1] "hello"
```

```
name_list[["a"]]
```

```
##   [1]   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##  [19]  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
##  [37]  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
##  [55]  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
##  [73]  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
##  [91]  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108
## [109] 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
## [127] 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
## [145] 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162
## [163] 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180
## [181] 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198
## [199] 199 200
```

```
add_vector <- rep.int(x = 1, 200)
add_vector
```

```
##   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [38] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [75] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [112] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [149] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [186] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

```
a <- name_list[["a"]] + add_vector
a
```

```
##   [1]   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19
##  [19]  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37
##  [37]  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55
##  [55]  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73
```

```
##  [73]  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91
##  [91]  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108 109
## [109] 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127
## [127] 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145
## [145] 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163
## [163] 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181
## [181] 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199
## [199] 200 201
```

```
name_list <- list(a = a, b = "this is a string",
                  c = "hello", z = "newItem")
name_list
```

```
## $a
##   [1]   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19
##  [19]  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37
##  [37]  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55
##  [55]  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73
##  [73]  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91
##  [91]  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108 109
## [109] 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127
## [127] 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145
## [145] 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163
## [163] 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181
## [181] 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199
## [199] 200 201
##
## $b
## [1] "this is a string"
##
## $c
## [1] "hello"
##
## $z
## [1] "newItem"
```

25.Download the small_counts.txt from the following location

```
# file <- "https://figshare.com/s/1d788fd384d33e913a2a"
# dest <- paste(getwd(), "small_counts.txt", sep = "/")
# dest

# if (file.exists(dest) == !T) {
#  download.file(file, dest, method = "wget", mode = "w")
# } else {
#  print("File is already downloaded !")
# }

### This is a link to the folder. We can download it manually
## by going to the browser and opening the folder and downloading
## the file manually.
```

```
### Or we can download the file from R by the direct link to the file.
"https://ndownloader.figshare.com/files/6005547?private_link=1d788fd384d33e913a2a"
```

```
## [1] "https://ndownloader.figshare.com/files/6005547?private_link=1d788fd384d33e913a2a"
```

```
file <- "https://ndownloader.figshare.com/files/6005547?private_link=1d788fd384d33e913a2a"
dest <- paste(getwd(), "small_counts.txt", sep = "/")
dest
```

```
## [1] "/Users/marufahmedbhuiyan/Desktop/cBALST_R/Day 5- 10 July 2020/small_counts.txt"
```

```
if (file.exists(dest) == !T) {
  download.file(file, dest, method = "auto")
} else {
  print("File is already downloaded !")
}
```

```
## [1] "File is already downloaded !"
```

26.Read the file in R and save it as small_counts. View the file.

```
small_counts <- read.table("small_counts.txt", header = TRUE)
small_counts
```

```
##         Sample_1 Sample_2 Sample_3 Sample_4
## Xkr4         438      300       65      237
## Sox17        106      182       82      105
## Mrpl15       309      234      337      300
## Lypla1       652      515      948      935
## Tcea1       1604     1495     1721     1317
## Rgs20          4        2       14        4
## Atp6v1h      769      752     1062      987
## Rb1cc1      1494     1412     1157      967
## Pcmtd1      1344     1242     1374     1593
## Rrs1        1691     1808     2127     1653
```

```
#View(small_counts)
```

27.Get the following output from the file

```
small_counts[,1:2]
```

```
##         Sample_1 Sample_2
## Xkr4         438      300
```

```
## Sox17        106      182
## Mrpl15       309      234
## Lypla1       652      515
## Tcea1       1604     1495
## Rgs20          4        2
## Atp6v1h      769      752
## Rb1cc1      1494     1412
## Pcmtd1      1344     1242
## Rrs1        1691     1808
```

28.Get log of the small_count so that it looks like the following

```
log(small_counts)
```

```
##         Sample_1  Sample_2 Sample_3 Sample_4
## Xkr4    6.082219 5.7037825 4.174387 5.468060
## Sox17   4.663439 5.2040067 4.406719 4.653960
## Mrpl15  5.733341 5.4553211 5.820083 5.703782
## Lypla1  6.480045 6.2441669 6.854355 6.840547
## Tcea1   7.380256 7.3098815 7.450661 7.183112
## Rgs20   1.386294 0.6931472 2.639057 1.386294
## Atp6v1h 6.645091 6.6227363 6.967909 6.894670
## Rb1cc1  7.309212 7.2527624 7.053586 6.874198
## Pcmtd1  7.203406 7.1244783 7.225481 7.373374
## Rrs1    7.433075 7.4999765 7.662468 7.410347
```

29.Download the "ResultsTable_small.txt" from the following location "https://figshare.com/s/1d788fd384d33e913a2a"

```
### This is a link to the folder. We can download it manually
### by going to the browser and opening the folder and
### downloading the file manually.

### Or we can download the file from R by the direct link to the file.
# "https://ndownloader.figshare.com/files/6005550?private_link=1d788fd384d33e913a2a"

file <-
  "https://ndownloader.figshare.com/files/6005550?private_link=1d788fd384d33e913a2a"
dest <- paste(getwd(), "ResultsTable_small.txt", sep = "/")
dest
```

```
## [1] "/Users/marufahmedbhuiyan/Desktop/cBALST_R/Day 5- 10 July 2020/ResultsTable_small.txt"
```

```
if (file.exists(dest) == !T) {
  download.file(file, dest, method = "auto")
} else {
  print("File is already downloaded !")
}
```

```
## [1] "File is already downloaded !"
```

30.This is a file which contains the gene expression data. The Entrez id is the gene name. You can search Entrez id in google to get more information.Symbol is the gene name. And "logFC" value which means how much more a gene is expressed in treatment condition compared to control condition. Read the file in R and store it as "results".

```
results <- read.table(dest, header = T)
head(results)
```

```
##   ENTREZID        SYMBOL     logFC  AveExpr         t      P.Value   adj.P.Val
## 1    24117          Wif1  1.819943 2.975545  20.10780 1.063770e-10 1.01624e-06
## 2   381290        Atp2b4 -2.143885 3.944066 -19.07495 1.982934e-10 1.01624e-06
## 3    78896 1500015O10Rik  2.807548 3.036519  18.54773 2.758828e-10 1.01624e-06
## 4   226101          Myof -2.329744 6.223525 -18.26861 3.297667e-10 1.01624e-06
## 5    16012        Igfbp6 -2.896115 1.978449 -18.21525 3.413066e-10 1.01624e-06
## 6   231830       Micall2  2.253400 4.760597  18.02627 3.858161e-10 1.01624e-06
```

```
str(results)
```

```
## 'data.frame':    40 obs. of  7 variables:
##  $ ENTREZID : int  24117 381290 78896 226101 16012 231830 16669 55987 231991 14620 ...
##  $ SYMBOL   : Factor w/ 40 levels "1500015O10Rik",..: 40 3 1 26 20 23 21 8 9 16 ...
##  $ logFC    : num  1.82 -2.14 2.81 -2.33 -2.9 ...
##  $ AveExpr  : num  2.98 3.94 3.04 6.22 1.98 ...
##  $ t        : num  20.1 -19.1 18.5 -18.3 -18.2 ...
##  $ P.Value  : num  1.06e-10 1.98e-10 2.76e-10 3.30e-10 3.41e-10 ...
##  $ adj.P.Val: num  1.02e-06 1.02e-06 1.02e-06 1.02e-06 1.02e-06 ...
```

31.Sort the file such thath the genes are orderd in highest to lowest value of "logFC".

```
head(results[,1:3], 10)
```

```
##    ENTREZID        SYMBOL     logFC
## 1     24117          Wif1  1.819943
## 2    381290        Atp2b4 -2.143885
## 3     78896 1500015O10Rik  2.807548
## 4    226101          Myof -2.329744
## 5     16012        Igfbp6 -2.896115
## 6    231830       Micall2  2.253400
## 7     16669          Krt19 -2.312721
```

```
## 8     55987         Cpxm2 -1.515469
## 9    231991         Creb5 -2.598105
## 10    14620          Gjb3  3.600094
```

```
head(results[order(-results$logFC),],10)
```

```
##    ENTREZID        SYMBOL    logFC  AveExpr        t      P.Value    adj.P.Val
## 22    16878           Lif 3.738933 6.682034 13.73344 9.105708e-09 6.541210e-06
## 10    14620          Gjb3 3.600094 3.525281 16.46627 1.113755e-09 1.718703e-06
## 25    12977          Csf1 2.835624 7.477591 13.41902 1.187300e-08 7.505634e-06
## 3     78896 1500015O10Rik 2.807548 3.036519 18.54773 2.758828e-10 1.016240e-06
## 15    11636           Ak1 2.766745 4.303475 15.27694 2.664640e-09 2.807465e-06
## 26    12654         Chil1 2.342914 5.576457 13.21976 1.408760e-08 8.306595e-06
## 29   217166         Nr1d1 2.278879 6.260878 13.12885 1.524242e-08 8.306595e-06
## 6    231830       Micall2 2.253400 4.760597 18.02627 3.858161e-10 1.016240e-06
## 13    74747         Ddit4 2.180370 6.864791 15.70145 1.938279e-09 2.356351e-06
## 20    17131         Smad7 1.972771 6.717519 14.14348 6.493642e-09 5.131276e-06
```

32.See the following figure

Type the above code in your console and check counts_matrix to see what does it create. What do you think "rpois" comment did here? And try to find what is the difference between paste and paste0. Always remember "google" is your friend.

```
counts_matrix <- data.frame(cell_1 = rpois(10,10),
                            cell_2 = rpois(10,10),
                            cell_3 = rpois(10, 30))
rownames(counts_matrix) <- paste0("gene_", 1:10)
counts_matrix <- as.matrix(counts_matrix)
counts_matrix
```

```
##         cell_1 cell_2 cell_3
## gene_1       9     12     28
## gene_2      12     10     36
## gene_3       9     11     38
## gene_4      10      8     35
## gene_5      17     12     32
## gene_6       3     15     28
## gene_7      10      8     38
## gene_8      11      7     23
## gene_9       9     10     28
## gene_10     10     13     32
```

```
## What do you think "rpois" comment did here?
## And try to find what is the difference between paste and paste0.

## rpois generates a Poisson distribution with random deviates.
## Other similar terms are dpois (density), qpois(quantile),
## ppois (log distribution function)

## The Poisson distribution is the discrete probability
```

```r
## distribution of the number of events occurring in a
## given time period, given the average number of times
## the event occurs over that time period.

## The difference between paste() and paste0() is that
## the argument sep by default is " " (paste) and ""
## (paste0). paste0() is faster than paste() if our
## objective is concatenate strings without spaces
## because we don't have to  specify the argument sep.
## For example...see the difference between these..

paste0("gene_", 1:10)
```

```
##  [1] "gene_1"  "gene_2"  "gene_3"  "gene_4"  "gene_5"  "gene_6"  "gene_7"
##  [8] "gene_8"  "gene_9"  "gene_10"
```

```r
paste("gene_", 1:10)
```

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

33.Create a heatmap from the using the following file: Explain what was done in each line. Use "##" to comment on your code file in R

Load the file "basketball.csv". Make sure you change the read.csv location from the following code.

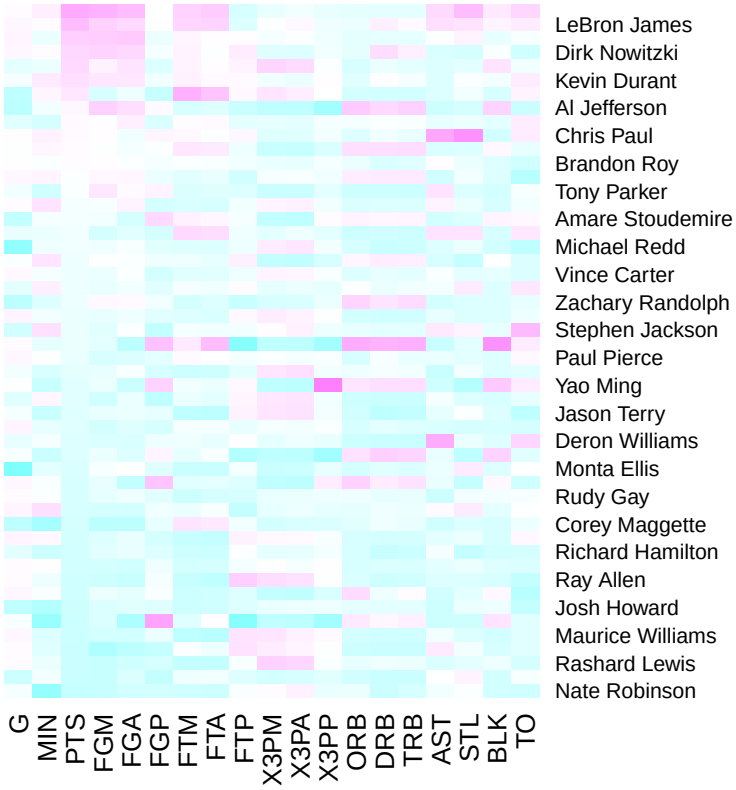
The code will/might show you error!.copy the error and put it in google and see what is the suggestion from the internet. Try to understand and solve the problem. The end of the code will show something like the following:

```r
file.exists("basketball.csv")
```

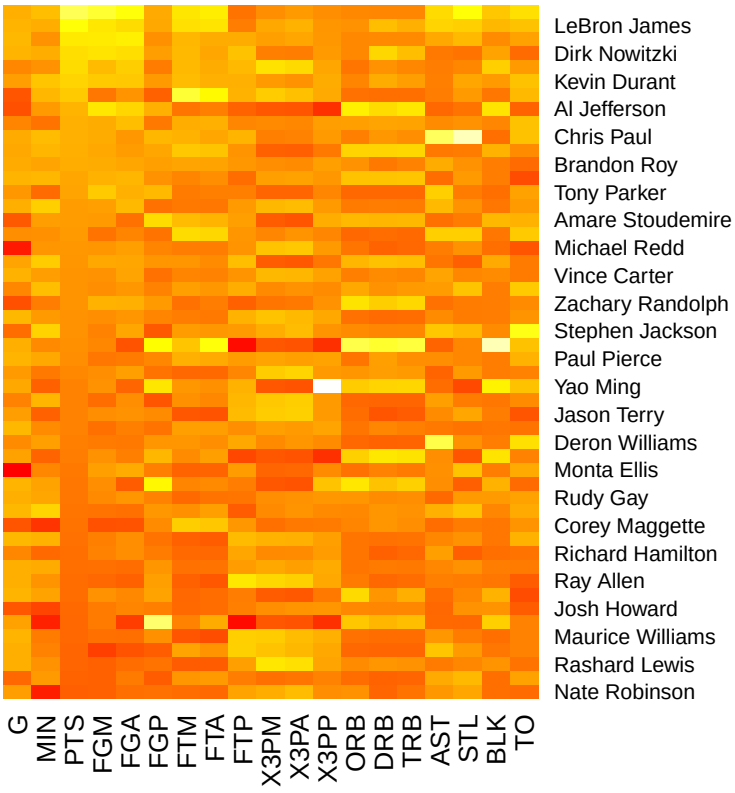
```
## [1] TRUE
```

```r
# importing the file
nba <- read.csv("basketball.csv")

# sorting the file accoring to PTS in increasing order
nba <- nba[order(nba$PTS),]

# Naming the rownames according to the Name column
row.names(nba) <- nba$Name

# Subsetting the data frame with all rows and
# 20 columns except the first one
nba <- nba[,2:20]

# Creating a matrix
```

```
nba_matrix <- data.matrix(nba)

# Creating a heatmap
nba_heatmap <- heatmap(nba_matrix, Rowv = NA,
                       Colv = NA,
                       col = cm.colors(256),
                       scale = "column",
                       margins = c(5,10))
```



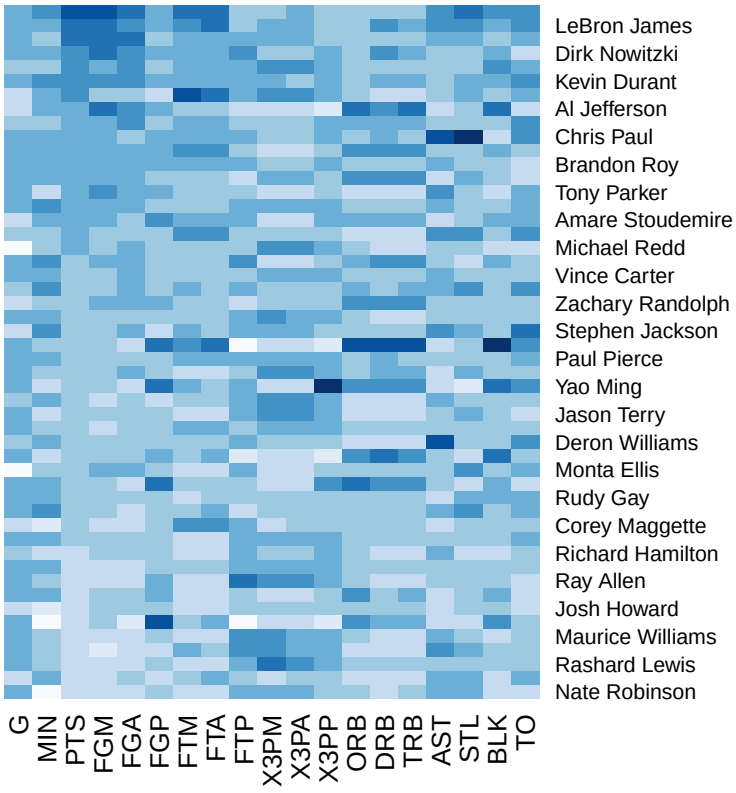

```
# The code didn't show any error to me ! So, let's continue

# I prefer the following color scheme better
nba_heatmap <- heatmap(nba_matrix, Rowv = NA,
                       Colv = NA,
                       col = heat.colors(256),
                       scale = "column",
                       margins = c(5,10))
```




```
# Blue is my favorite color. So, let's color it blue !!
if (!require("RColorBrewer")) {
install.packages("RColorBrewer")
library(RColorBrewer)
}
```

```
## Loading required package: RColorBrewer
```

```
nba_heatmap <- heatmap(nba_matrix, Rowv = NA,
                       Colv = NA,
                       col = brewer.pal(9, "Blues"),
                       scale = "column",
                       margins = c(5,10))
```


LeBron James
Dirk Nowitzki
Kevin Durant
Al Jefferson
Chris Paul
Brandon Roy
Tony Parker
Amare Stoudemire
Michael Redd
Vince Carter
Zachary Randolph
Stephen Jackson
Paul Pierce
Yao Ming
Jason Terry
Deron Williams
Monta Ellis
Rudy Gay
Corey Maggette
Richard Hamilton
Ray Allen
Josh Howard
Maurice Williams
Rashard Lewis
Nate Robinson
G
MIN
PTS
FGM
FGA
FGP
FTM
FTA
FTP
X3PM
X3PA
X3PP
ORB
DRB
TRB
AST
STL
BLK
TO