Data management

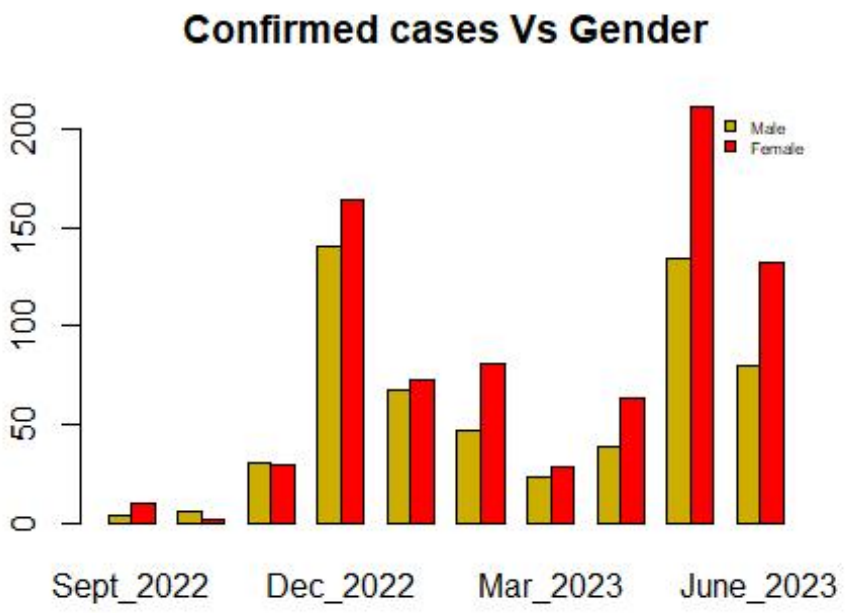
Dr. Pacifique

2025-10-01

Data Visualization and Data management

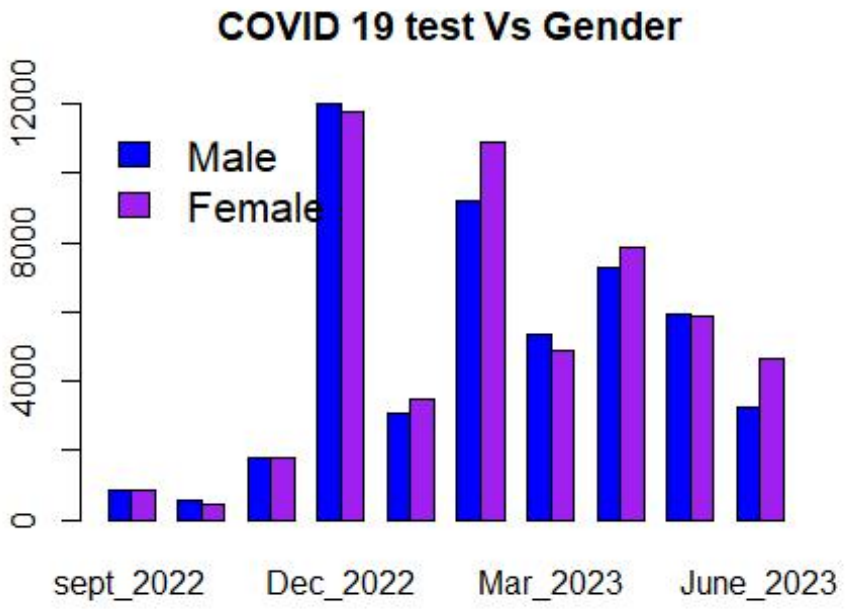
Test to treat data

```
data1<- read.table(text="Sept_2022 Oct_2022 Nov_2022 Dec_2022 Jan_2023
Feb_2023 Mar_2023 Apr_2023 May_2023 June_2023
1 4 6 30 140 67 47 23 39 134 80
2 10 2 29 164 72 81 28 63 211 132",header=TRUE)
barplot(as.matrix(data1),main="Confirmed cases Vs Gender",beside=TRUE,c
ol=c("gold3","red"))
legend("topright", c("Male","Female"),cex = 0.5,bty="n",fill = c("gold3
","red"))
```

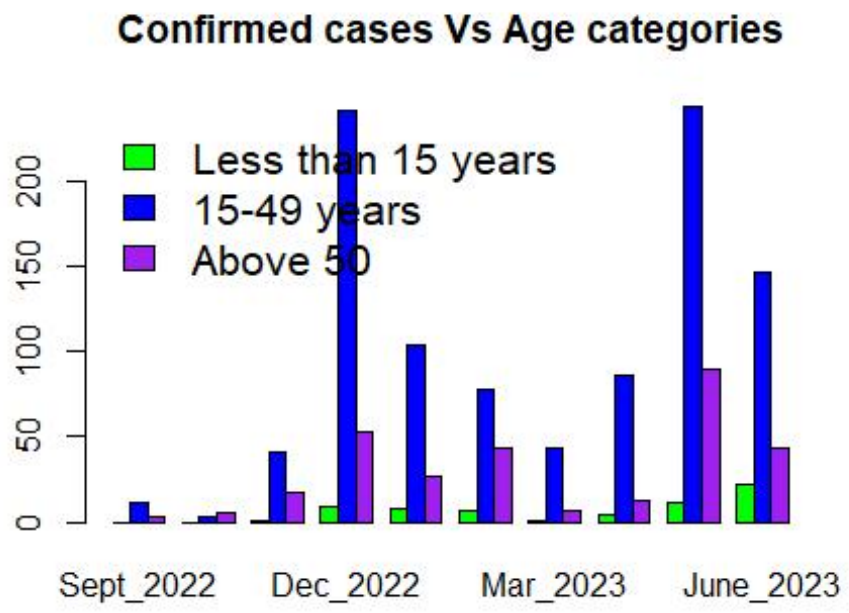



```
data3<- read.table(text="sept_2022 Oct_2022 Nov_2022 Dec_2022 Jan_2023
Feb_2023 Mar_2023 Apr_2023 May_2023 June_2023
1  867  582 1802    12003   3059    9205    5346    7269    5933    324
5
2  842  436 1805    11785   3508    10908   4902    7883    5901    464
6 ",header=TRUE)
```

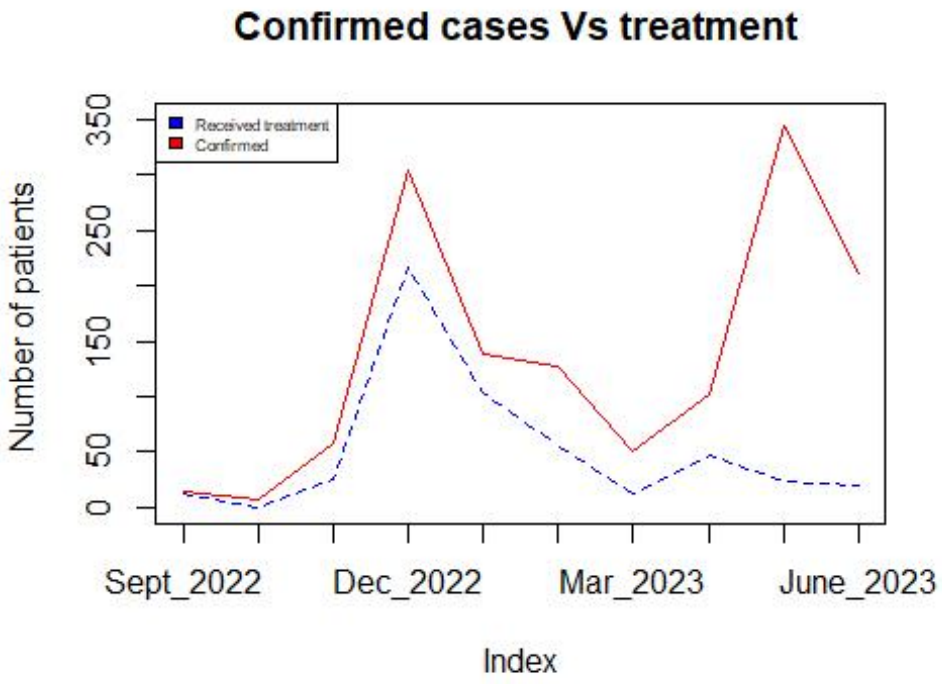
```
barplot(as.matrix(data3),main=" COVID 19 test Vs Gender",beside=TRUE,co
l=c("blue","purple"))
legend("topleft", c("Male","Female"),cex = 1.3,bty="n",fill = c("blue",
"purple"))
```



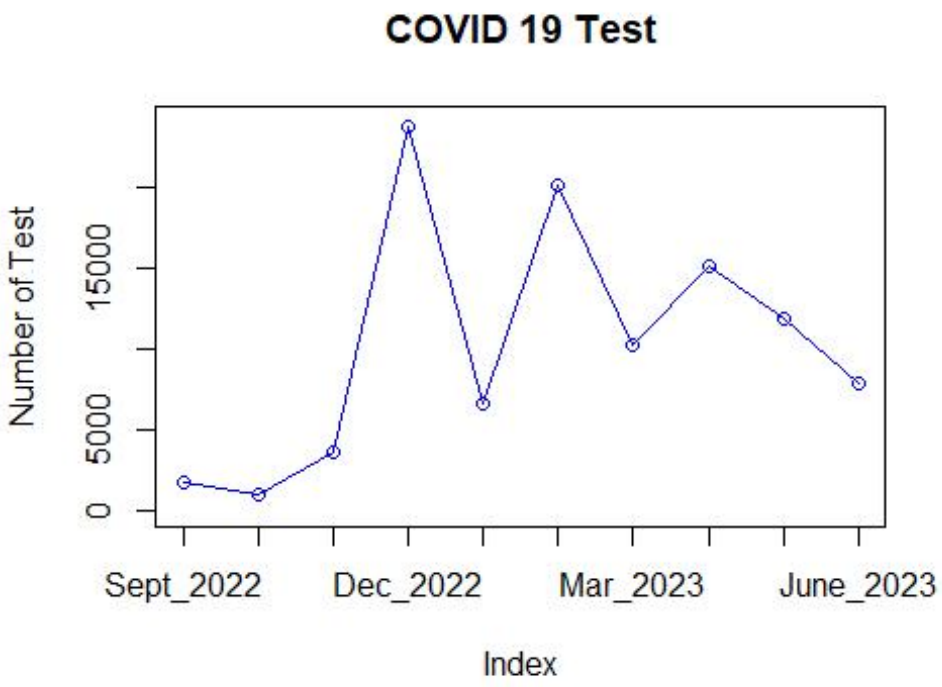

```
data2<- read.table(text="Sept_2022 Oct_2022 Nov_2022 Dec_2022 Jan_2023
Feb_2023 Mar_2023 Apr_2023 May_2023 June_2023
1 0 0 1 9 8 7 1 4 11 22
2 11 3 41 242 104 78 44 86 244 147
3 3 5 17 53 27 43 6 12 90 43 ",header=TRUE)
barplot(as.matrix(data2),main="Confirmed cases Vs Age categories",besid
e=TRUE,col=c("green","blue","purple"))
barplot(as.matrix(data2),main="Confirmed cases Vs Age categories",besid
e=TRUE,col=c("green","blue","purple"))
legend("topleft", c("Less than 15 years","15-49 years","Above 50"),cex
= 1.3,bty="n",fill = c("green","blue","purple"))
```




```
months<-c("Sept_2022","Oct_2022","Nov_2022","Dec_2022", "Jan_2023","Feb
_2023","Mar_2023","Apr_2023","May_2023","June_2023")
Confirmed_covid<-c(14,8,59,304,139,128,51,102,345,212)
plot(Confirmed_covid, type = "l",pch=21, col = "red",ylim=c(0,350),
     xaxt="n", ylab = "Number of patients ",
     main = "Confirmed cases Vs treatment ")
treatment<- c(12,0,26,217,104,57,12,47,23,20)

lines(treatment, col="blue",lty=2)

legend("topleft", legend=c("Received treatment", "Confirmed"),cex=0.5,
fill = c("blue","red"))
axis(1,at=1:10,lab=c("Sept_2022","Oct_2022","Nov_2022","Dec_2022", "Jan
_2023","Feb_2023","Mar_2023","Apr_2023","May_2023","June_2023"))
```




```
months<-c("sept_2022","Oct_2022","Nov_2022","Dec_2022", "Jan_2023","feb
_2023","Mar_2023","Apr_2023","May_2023", "June_2023")
number_of_test<-c(1709, 1018,   3607,23788,6567,20113,10248,15152,11834,
7891)
plot(number_of_test, type = "o", col = "blue",ylim=c(0,24000),
     xaxt="n", ylab = "Number of Test",
     main = "COVID 19 Test")
axis(1,at=1:10,lab=c("Sept_2022","Oct_2022","Nov_2022","Dec_2022", "Jan
_2023","Feb_2023","Mar_2023","Apr_2023","May_2023","June_2023"))
```



ggplot2

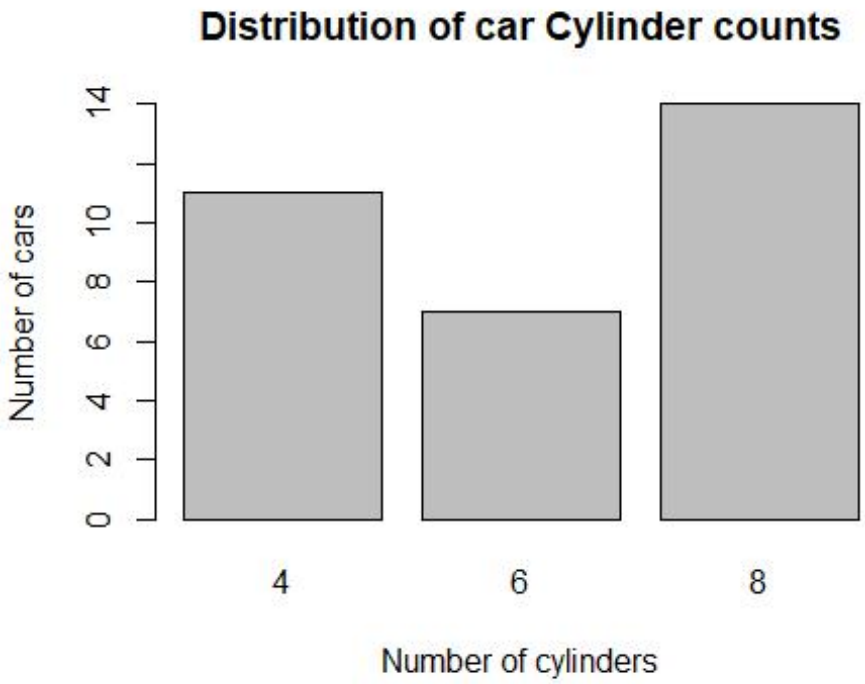
ggplot2 provides a set of tools that allows you to visualize complex data sets in a new creative way -some work need some packages to get done. -some of the graph are created using R's base graphics system

Library ggplot2

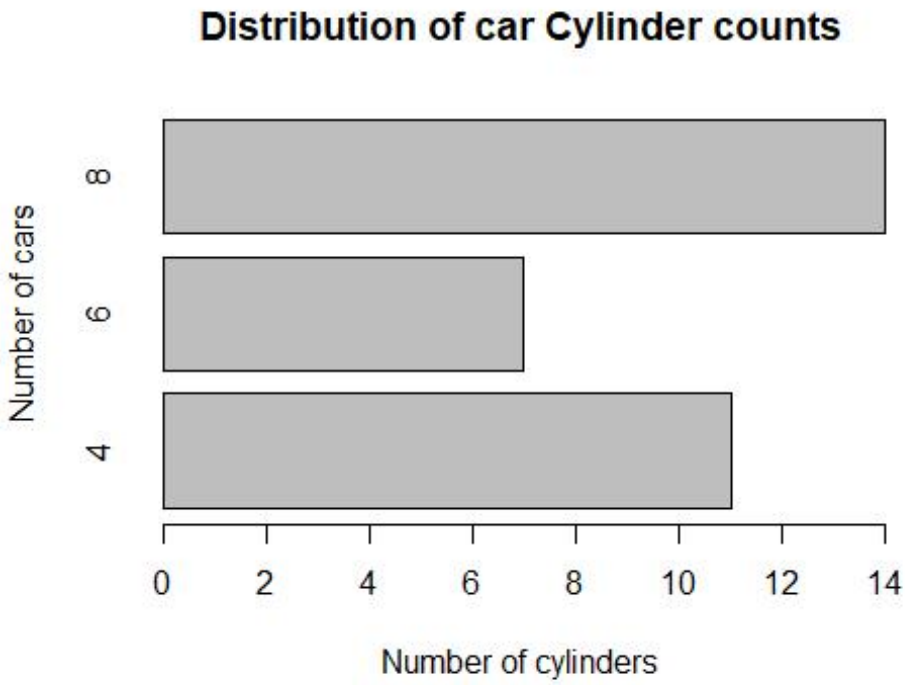
```
###install.packages("ggplot2")
library(ggplot2)
```

let us explore the data mcars

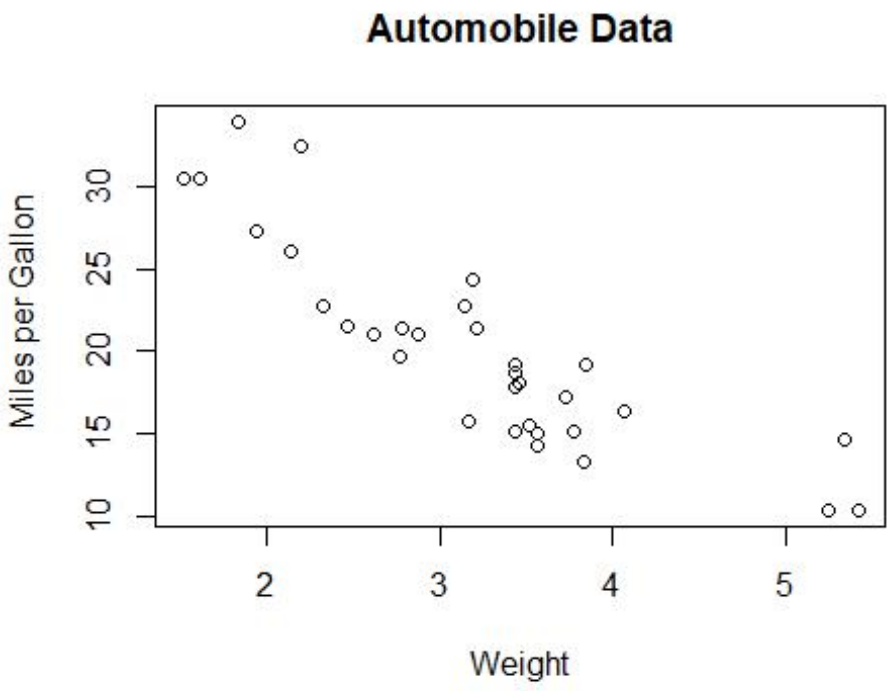
```
data(mtcars)
#dotchart(mtcars$mpg,labels=row.names(mtcars),ce#x = 0.,main="miles per
 Gallon of car #model",xlab = "MPG")

table(mtcars$cyl)

##
##  4  6  8
## 11  7 14

#barplot(mtcars$cyl)
barplot(table(mtcars$cyl),main="Distribution of car Cylinder counts",xl
ab="Number of cylinders",ylab = "Number of cars")
```



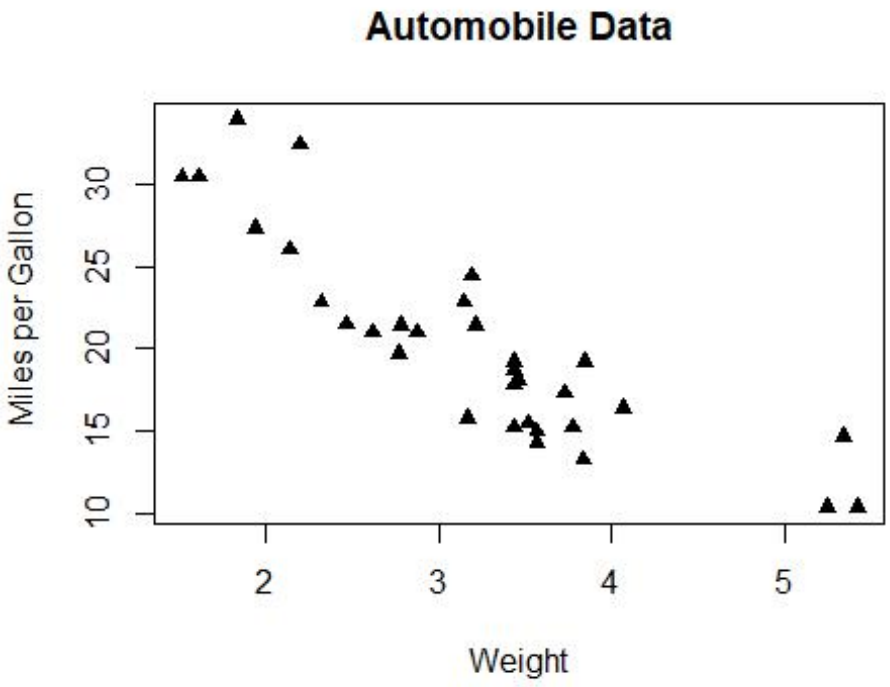

```r
barplot(table(mtcars$cyl),main="Distribution of car Cylinder counts",xlab="Number of cylinders",ylab = "Number of cars",horiz=TRUE)
```



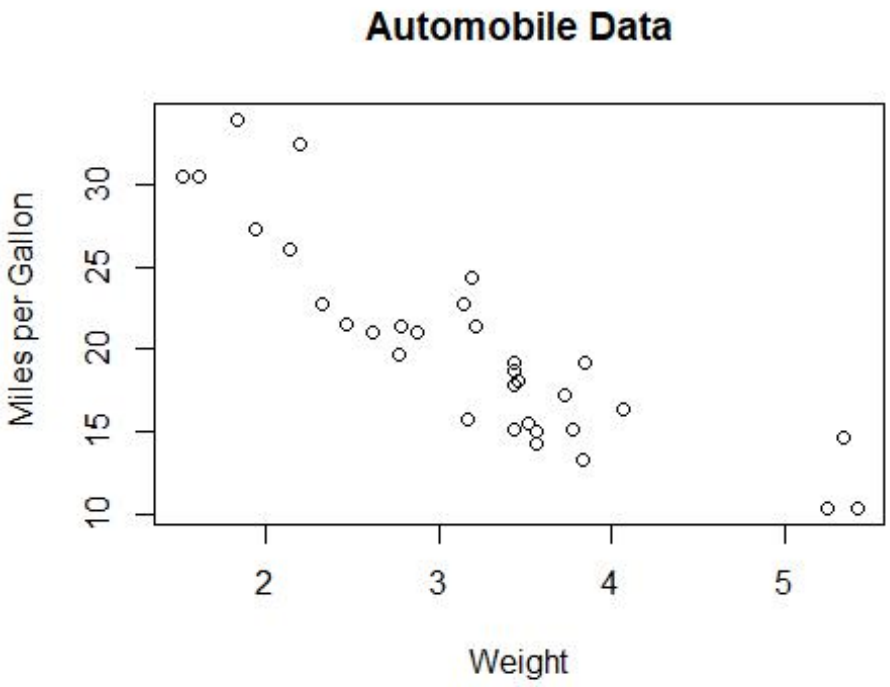
```
plot(mtcars$mpg~mtcars$wt,main="Automobile Data",xlab="Weight",ylab=" M
iles per Gallon")
```



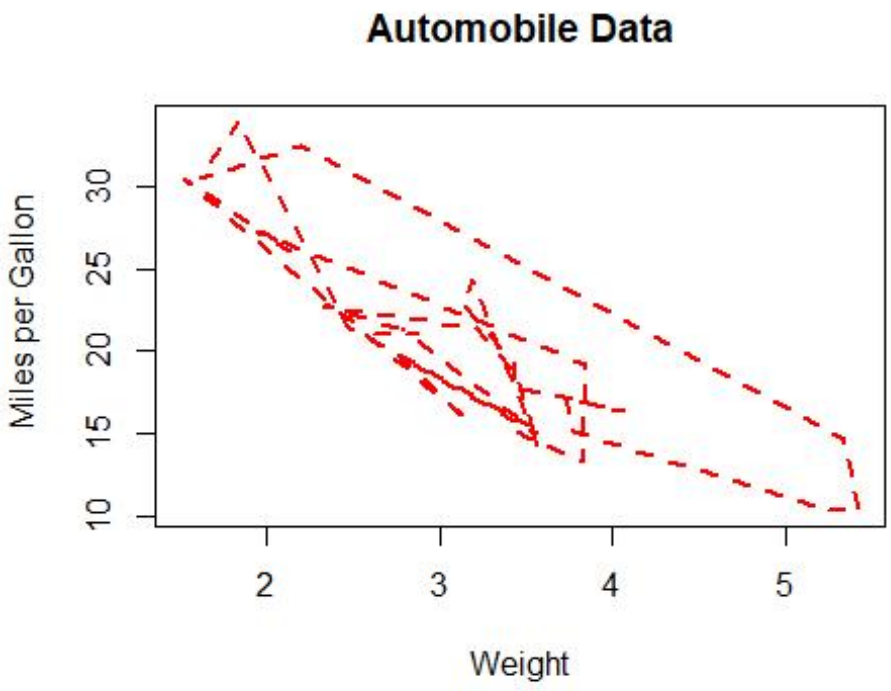

```
plot(mtcars$mpg~mtcars$wt,main="Automobile Data",xlab="Weight",ylab=" M
iles per Gallon")
plot(mtcars$mpg~mtcars$wt,pch=17,main="Automobile Data",xlab="Weight",y
lab=" Miles per Gallon")
```




```
plot(mtcars$mpg~mtcars$wt,pch=21,main="Automobile Data",xlab="Weight",y
lab=" Miles per Gallon")
```



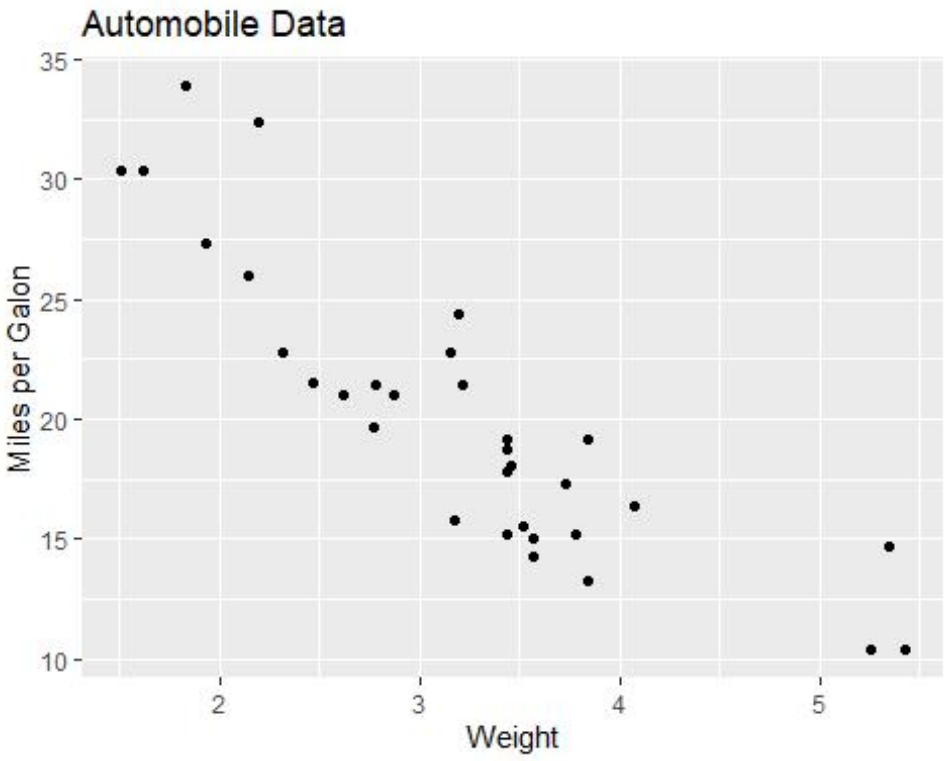
```
plot(mtcars$mpg~mtcars$wt,type="l", lty=2, lwd=2,col="red",main="Automo
bile Data",xlab="Weight",ylab=" Miles per Gallon")
```



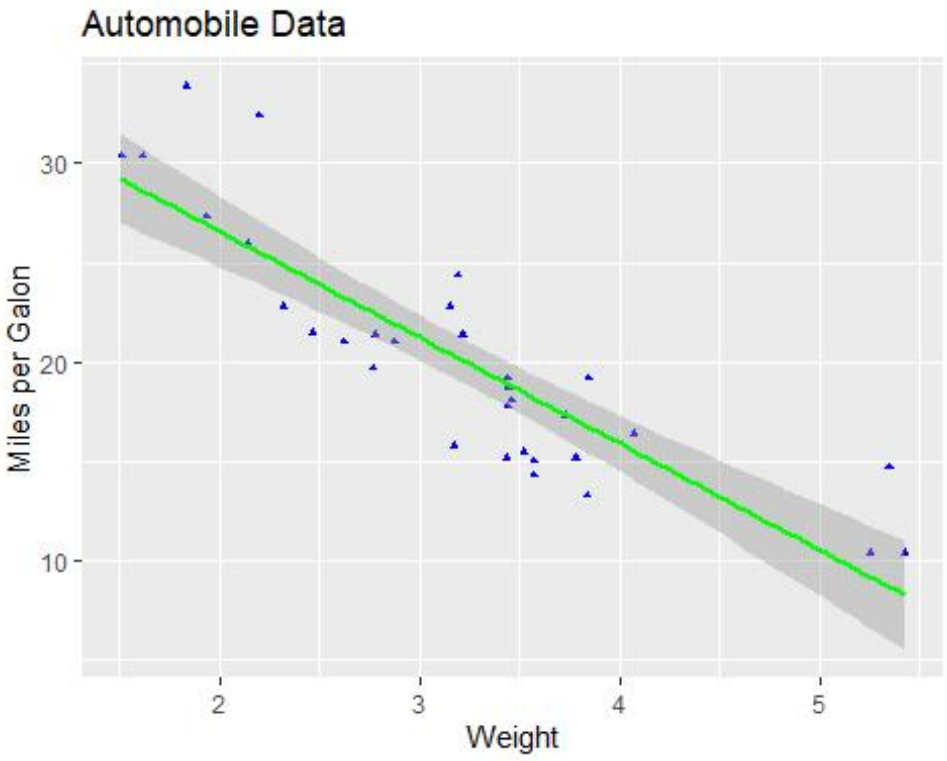
in ggplot2, plots are created by chaining together function using (+)sign. Each function modify the

##plot created up to that point

```
library(ggplot2)
ggplot(data=mtcars, aes(x=wt,y=mpg))+geom_point()+
  labs(title = "Automobile Data",x="Weight",y="Miles per Galon")
```

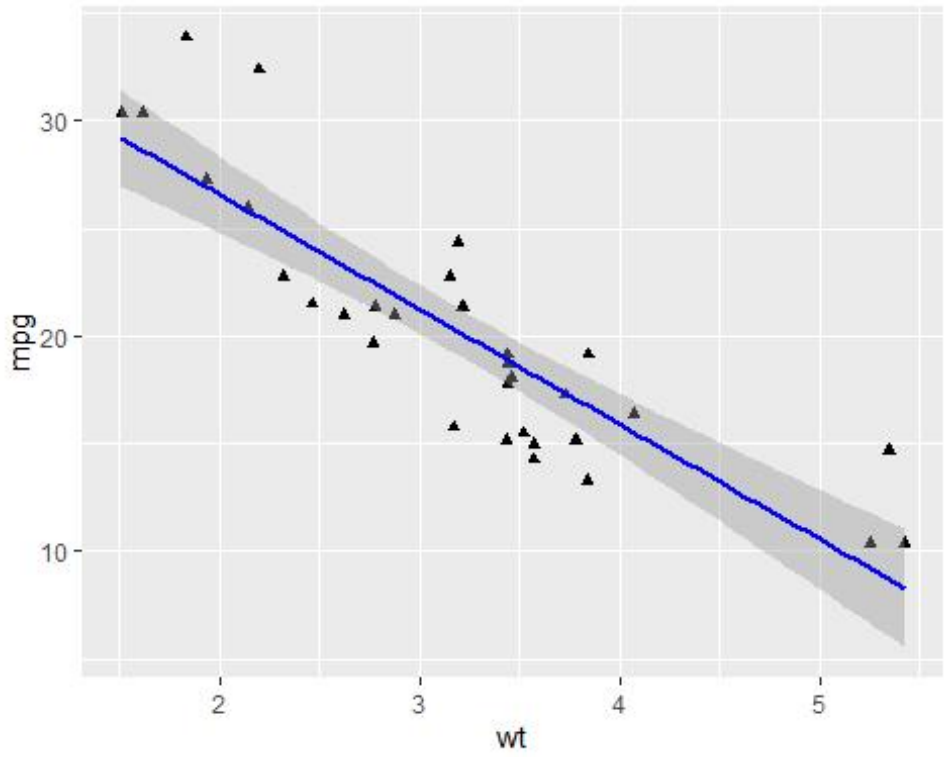



```
ggplot(data=mtcars, aes(x=wt,y=mpg))+geom_point(pch=17,color="blue",siz
e=1)+
  geom_smooth(method="lm",color="green")+
  labs(title = "Automobile Data",x="Weight",y="Miles per Galon")

## `geom_smooth()` using formula = 'y ~ x'
```

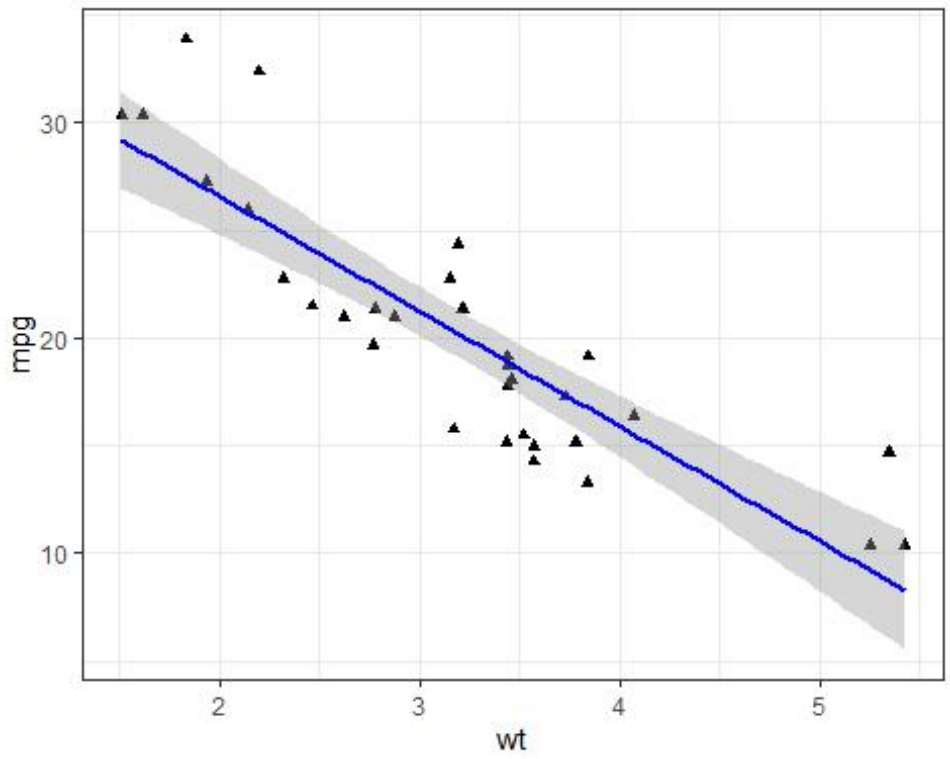



```
f0<-ggplot(data=mtcars, aes(x=wt,y=mpg))+geom_point(shape=24, fill=1,si
ze=1)+
  geom_smooth(method="lm",color="blue")

f0
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

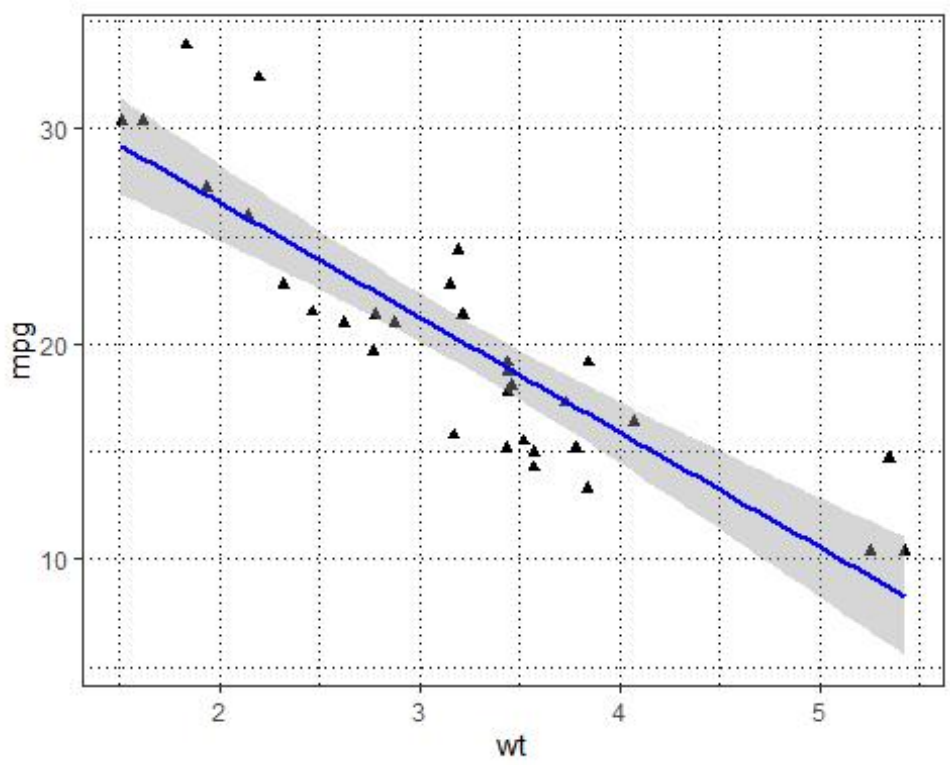



```
f1<-f0+theme_bw()
f1

## `geom_smooth()` using formula = 'y ~ x'
```



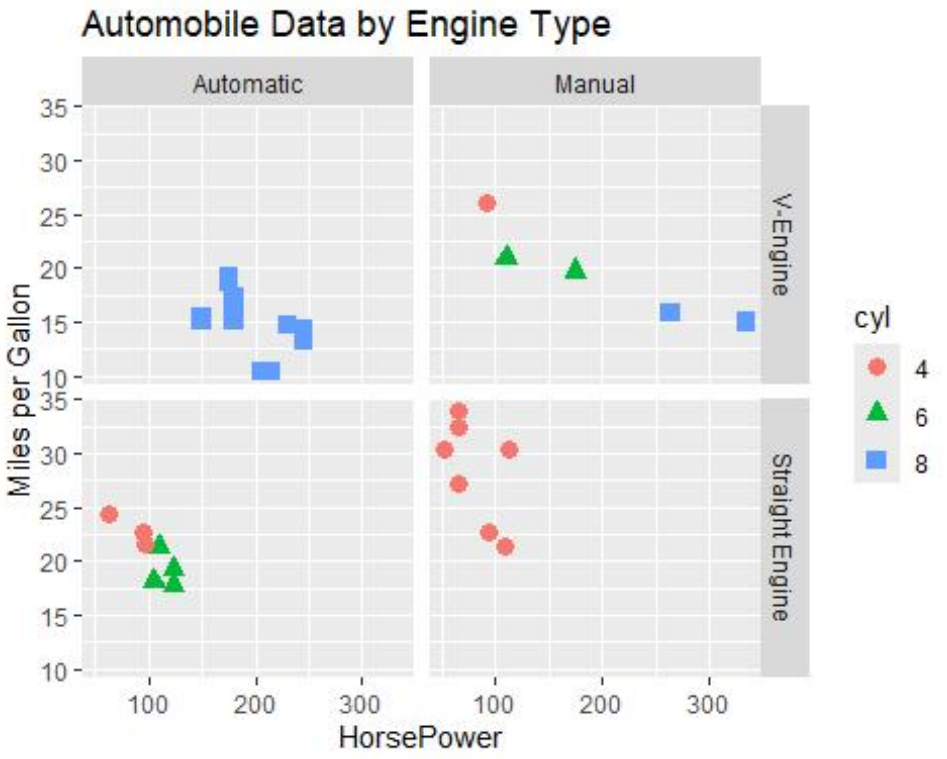

```
f2<-f1+
  theme(panel.grid = element_line(linetype = "dotted",color = "Black"))
f2
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

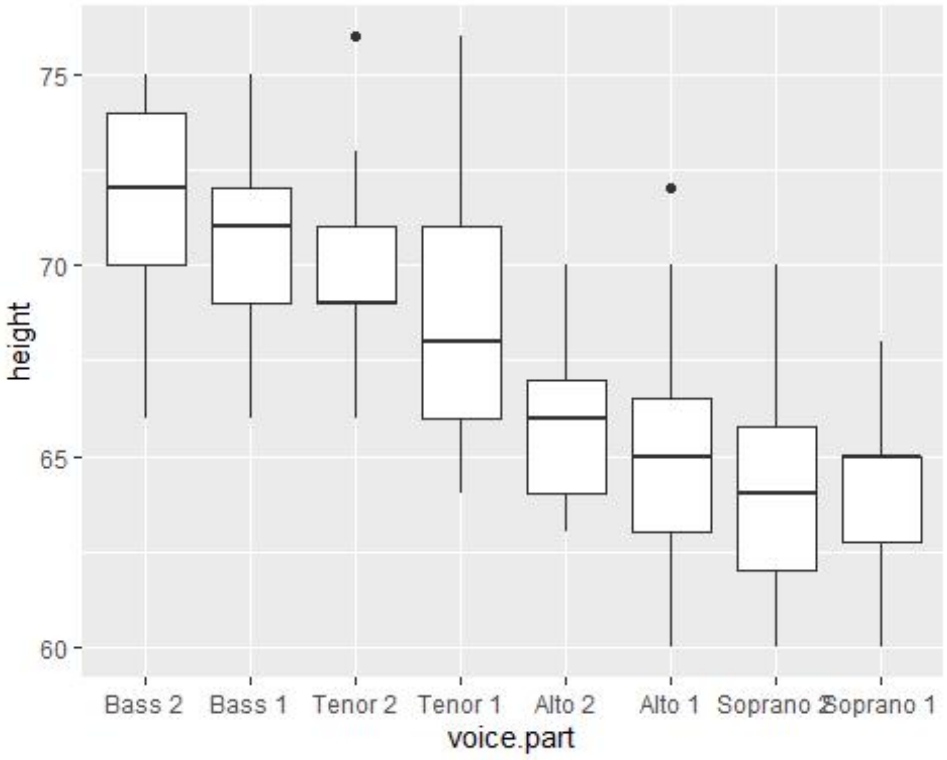


Grouping displays two or more groups of observations in a single plot

```
data(mtcars)

mtcars$am<-factor(mtcars$am, levels =c(0,1),
                  labels= c("Automatic","Manual"))
mtcars$vs<-factor(mtcars$vs, levels =c(0,1),
                  labels=c("V-Engine","Straight Engine"))
mtcars$cyl<-factor(mtcars$cyl)
ggplot(data=mtcars,aes(x=hp,y=mpg,shape=cyl,color=cyl))+
  geom_point(size=3)+
  facet_grid(vs~am)+
  labs(title = "Automobile Data by Engine Type",x="HorsePower",y="Miles
 per Gallon")
```




```
data(singer,package="lattice")
ggplot(singer,aes(x=voice.part,y=height))+geom_boxplot()
```



How work

Use the following function:

geom_bar() geom_boxplot() geom_density() geom_histogram() geom_hline() geom_jitter() geom_line() #geom_point() geom_rug() #geom_smooth() geom_text() geom_violin() geom_vline() ## Basic data management

cbind and rbind

While combining column wise, the number of rows must match but row names are ignored. when combining row-wise, both the number and the names of columns must match.

```
data2<- data.frame(x=1:3, y=c("a","b","c"))
str(data2)
```

```
## 'data.frame':    3 obs. of  2 variables:
##  $ x: int  1 2 3
##  $ y: chr  "a" "b" "c"
```

```
(cbind(data2,data.frame(z=3:1)))
```

```
##   x y z
## 1 1 a 3
## 2 2 b 2
## 3 3 c 1
```

```
(rbind(data2,data.frame(x=10,y="z")))
```

```
##    x y
## 1  1 a
## 2  2 b
## 3  3 c
## 4 10 z
```

Create another Variable

```
data_class<-read.table("C:\\Users\\Pacy\\OneDrive\\Desktop\\Big data course\\class_data.txt")
variable.names(data_class)
```

```
## [1] "HEIGHT" "WEIGHT"
```

```
head(data_class,n=5)
```

```
##   HEIGHT WEIGHT
## 1    161     50
## 2    155     49
## 3    158     42
## 4    170     65
## 5    160     60
```

```
tail(data_class)
```

```
##    HEIGHT WEIGHT
## 37    155     52
## 38    164     47
## 39    163     52
## 40    168     55
## 41    157     48
## 42    164     58
```

```
data_class[,1]
```

```
##  [1] 161 155 158 170 160 156 162 158 158 167 160 155 154 155 157 157
 160 158 160
## [20] 160 152 154 150 161 162 164 161 155 159 163 159 160 158 165 156
 163 155 164
## [39] 163 168 157 164
```

```
summary(data_class$WEIGHT)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    42.0    48.0    52.0    52.4    56.0    65.0
```

```
length(data_class$WEIGHT)
```

```
## [1] 42
```

```
data_class[,-1]
```

```
##  [1] 50 49 42 65 60 52 58 46 45 51 60 42 53 48 48 48 53 52 51 53 44
56 63 52 57
## [26] 49 52 54 46 50 61 55 45 63 60 56 52 47 52 55 48 58
```

```
attach(data_class)
(BMI<-WEIGHT/(HEIGHT/100)^2)
```

```
##  [1] 19.28938 20.39542 16.82423 22.49135 23.43750 21.36752 22.10029
18.42653
##  [9] 18.02596 18.28678 23.43750 17.48179 22.34778 19.97919 19.47341
19.47341
## [17] 20.70312 20.83000 19.92187 20.70312 19.04432 23.61275 28.00000
20.06095
## [25] 21.71925 18.21832 20.06095 22.47659 18.19548 18.81892 24.12879
21.48437
## [33] 18.02596 23.14050 24.65483 21.07720 21.64412 17.47472 19.57168
19.48696
## [41] 19.47341 21.56454
```

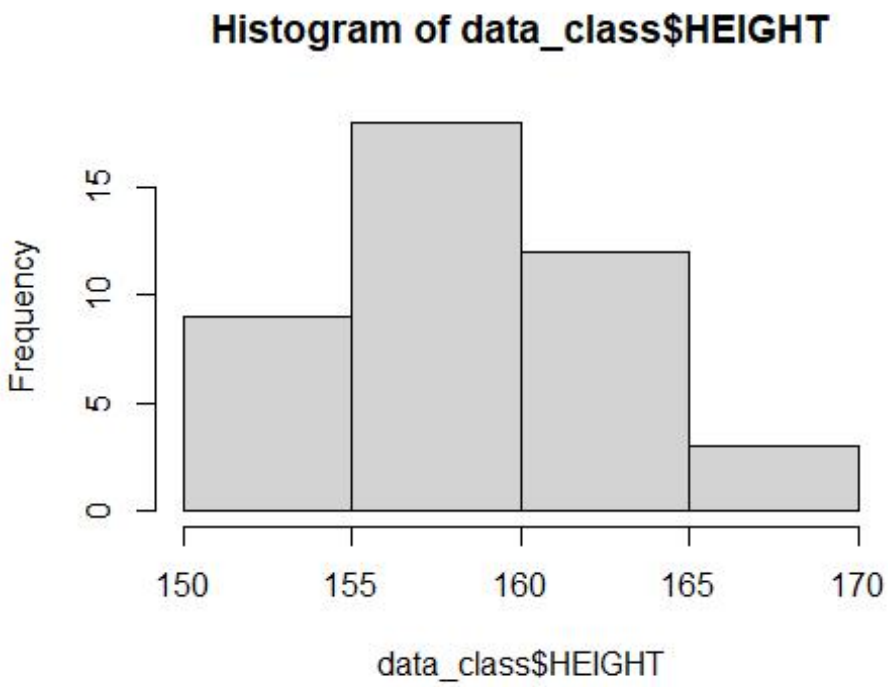
```
(BMI<-round(WEIGHT/(HEIGHT/100)^2,digit=1))
```

```
##  [1] 19.3 20.4 16.8 22.5 23.4 21.4 22.1 18.4 18.0 18.3 23.4 17.5 22.
3 20.0 19.5
## [16] 19.5 20.7 20.8 19.9 20.7 19.0 23.6 28.0 20.1 21.7 18.2 20.1 22.
```

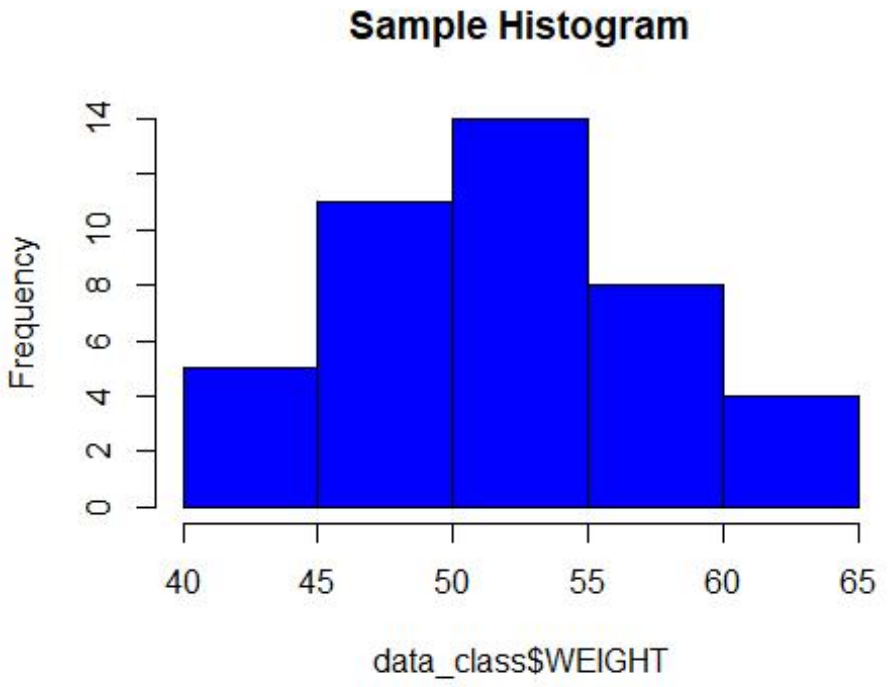
```
5 18.2 18.8
## [31] 24.1 21.5 18.0 23.1 24.7 21.1 21.6 17.5 19.6 19.5 19.5 21.6

head(cbind(data_class,BMI))

##   HEIGHT WEIGHT  BMI
## 1    161     50 19.3
## 2    155     49 20.4
## 3    158     42 16.8
## 4    170     65 22.5
## 5    160     60 23.4
## 6    156     52 21.4

tail(cbind(data_class,BMI),n=10)

##    HEIGHT WEIGHT  BMI
## 33    158     45 18.0
## 34    165     63 23.1
## 35    156     60 24.7
## 36    163     56 21.1
## 37    155     52 21.6
## 38    164     47 17.5
## 39    163     52 19.6
## 40    168     55 19.5
## 41    157     48 19.5
## 42    164     58 21.6

detach(data_class)
```

Summary of BMI

```
summary(BMI)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   16.80   19.07   20.25   20.64   22.00   28.00

hist(data_class$HEIGHT)
```




```r
hist(data_class$WEIGHT,col = "blue",main = "Sample Histogram")
```




```r
stem(BMI,scale=2)
```

```
##
##   The decimal point is at the |
##
##   16 | 8
##   17 | 55
##   18 | 0022348
##   19 | 03555569
##   20 | 0114778
##   21 | 145667
##   22 | 1355
##   23 | 1446
##   24 | 17
##   25 |
##   26 |
##   27 |
##   28 | 0
```

Subsetting

Use of $, []. or [[]]

Creation of leadership data Interest: How men and women differ in the way they lead their organizations. 5 questions were asked in this study. Example of the question: Do men and women in management position differ in the degree to which they defer to superiors? 1: strongly disagree, 2: disagree, 3:neither agree nor disagree,4: agree, 5: strongly agree

```
manager <- c(1,2,3,4,5)
country<- c("US","US","UK","UK","UK")
gender<-c("M","F","F","M","F")
age<-c(32,45,25,39,99)
q1<-c(5,3,3,3,2)
q2<-c(4,5,5,3,2)
q3<-c(5,2,5,4,1)
q4<-c(5,5,5,NA,2)
q5<-c(5,5,2,NA,1)
leadership<-data.frame(manager,country, gender, age, q1,q2,q3,q4,q5, st
ringsAsFactors = FALSE)
leadership<-data.frame(manager,country, gender, age, q1,q2,q3,q4,q5)
str(leadership)
```

```
## 'data.frame':    5 obs. of  9 variables:
##  $ manager: num  1 2 3 4 5
##  $ country: chr  "US" "US" "UK" "UK" ...
##  $ gender : chr  "M" "F" "F" "M" ...
##  $ age    : num  32 45 25 39 99
##  $ q1     : num  5 3 3 3 2
##  $ q2     : num  4 5 5 3 2
##  $ q3     : num  5 2 5 4 1
```

```
##  $ q4      : num  5 5 5 NA 2
##  $ q5      : num  5 5 2 NA 1
```

```
names(leadership)
```

```
## [1] "manager" "country" "gender"  "age"     "q1"      "q2"      "q3"

## [8] "q4"      "q5"
```

What you can do - combine the score of the five questions -handle the missing values - create a dataset of what you want -create age group or age categories - 99 indicate the value is missing

Missing value

```
leadership$age[leadership$age==99]<-NA

leadership$agecat[leadership$age >75]<-"Elder"
leadership$agecat[leadership$age>=35 & leadership$age<=75]<-"Middle Age
d"
leadership$agecat[leadership$age <35]<- "Young"
```

One way to handle missing value

Deleting all observations with missing data (Listwise deletion) is one of the several methods of handling incomplete datasets. Note: You can also replacing the missing value by the average of the remaining data.

```
is.na(leadership[,5:9])
```

```
##         q1    q2    q3    q4    q5
## [1,] FALSE FALSE FALSE FALSE FALSE
## [2,] FALSE FALSE FALSE FALSE FALSE
## [3,] FALSE FALSE FALSE FALSE FALSE
## [4,] FALSE FALSE FALSE  TRUE  TRUE
## [5,] FALSE FALSE FALSE FALSE FALSE
```

```
newdata<-na.omit(leadership)
newdata
```

```
##   manager country gender age q1 q2 q3 q4 q5      agecat
## 1       1      US      M  32  5  4  5  5  5       Young
## 2       2      US      F  45  3  5  2  5  5 Middle Aged
## 3       3      UK      F  25  3  5  5  5  2       Young
```

Sorting data

```
order(leadership$age)
```

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

```
newdata2<-leadership[order(leadership$age),]
```

Merging data sets

To merge two data frame horizontally, you use merge() function. In most cases, two data frames are joined by one or more common key variables. -example 1: merge(dataframeA, dataframeB,by="ID") -example 2: merge(dataframeA,dataframeB, by=c("ID","country"))

the second merge the two dataframes by ID and country

To join two data frame(datasets) vertically, use rbind() function: Note that the two data set must have the same variables

Subset (selecting variables,dropping variables,selecting observation)

```
newdata3<-leadership[,c(5:9)]
myvars<-c("q1","q2","q3","q4","q5")
newdata3<-leadership[myvars]
newdata4<-leadership[,c(-1,-2)]
newdata4<-leadership[,-(1:5)]
newdata5<-leadership[,c(-1,-7)]
leadership[[4]]

## [1] 32 45 25 39 NA

leadership$age

## [1] 32 45 25 39 NA

newdata6<-leadership[c(-1,-3),]
newdata6<-leadership[c(2,4,5),]
newdata7<-subset(leadership,age>=35|age<24,select=c(q1,q2,q3,q4,q5))

attach(leadership)

## The following objects are masked _by_ .GlobalEnv:
##
##     age, country, gender, manager, q1, q2, q3, q4, q5

#newdata7<-leadership[gender="M" & age>30,]
newdata7<-leadership[gender=="M" & age>30,]
detach(leadership)
```

Data management with dyplr

Data source from the package nycflights13 and ggplot2 This data contains all 336,776 flights that departed from New York city in 2013.

Data management with dyplr

Data source from the package nycflights13 and ggplot2 This data contains all 336,776 flights that departed from New york city in 2013.

```r
#install.packages("nycflights13")
#install.packages("tidyverse")
library(nycflights13)
```

```
## Warning: package 'nycflights13' was built under R version 4.2.3
```

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

```
## Warning: package 'dplyr' was built under R version 4.2.3
```

```
##
## 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
```

```r
#library(tidyverse)
```

The function filter()

#rm(list = ls())

```r
str(flights)
```

```
## tibble [336,776 × 19] (S3: tbl_df/tbl/data.frame)
##  $ year          : int [1:336776] 2013 2013 2013 2013 2013 2013 2013
 2013 2013 2013 ...
##  $ month         : int [1:336776] 1 1 1 1 1 1 1 1 1 1 ...
##  $ day           : int [1:336776] 1 1 1 1 1 1 1 1 1 1 ...
##  $ dep_time      : int [1:336776] 517 533 542 544 554 554 555 557 55
7 558 ...
##  $ sched_dep_time: int [1:336776] 515 529 540 545 600 558 600 600 60
0 600 ...
##  $ dep_delay     : num [1:336776] 2 4 2 -1 -6 -4 -5 -3 -3 -2 ...
##  $ arr_time      : int [1:336776] 830 850 923 1004 812 740 913 709 8
38 753 ...
##  $ sched_arr_time: int [1:336776] 819 830 850 1022 837 728 854 723 8
46 745 ...
##  $ arr_delay     : num [1:336776] 11 20 33 -18 -25 12 19 -14 -8 8 ...
##  $ carrier       : chr [1:336776] "UA" "UA" "AA" "B6" ...
##  $ flight        : int [1:336776] 1545 1714 1141 725 461 1696 507 57
08 79 301 ...
##  $ tailnum       : chr [1:336776] "N14228" "N24211" "N619AA" "N804JB
"  ...
##  $ origin        : chr [1:336776] "EWR" "LGA" "JFK" "JFK" ...
##  $ dest          : chr [1:336776] "IAH" "IAH" "MIA" "BQN" ...
```

```
##  $ air_time      : num [1:336776] 227 227 160 183 116 150 158 53 140
 138 ...
##  $ distance      : num [1:336776] 1400 1416 1089 1576 762 ...
##  $ hour          : num [1:336776] 5 5 5 5 6 5 6 6 6 6 ...
##  $ minute        : num [1:336776] 15 29 40 45 0 58 0 0 0 0 ...
##  $ time_hour     : POSIXct[1:336776], format: "2013-01-01 05:00:00"
"2013-01-01 05:00:00" ...
```

```
#attach(flights)
jan<-filter(flights,month==1)
jan
```

```
## # A tibble: 27,004 × 19
##     year month   day dep_time sched_dep_time dep_delay arr_time sche
d_arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
     <int>
##  1  2013     1     1      517            515         2      830
       819
##  2  2013     1     1      533            529         4      850
       830
##  3  2013     1     1      542            540         2      923
       850
##  4  2013     1     1      544            545        -1     1004
      1022
##  5  2013     1     1      554            600        -6      812
       837
##  6  2013     1     1      554            558        -4      740
       728
##  7  2013     1     1      555            600        -5      913
       854
##  8  2013     1     1      557            600        -3      709
       723
##  9  2013     1     1      557            600        -3      838
       846
## 10  2013     1     1      558            600        -2      753
       745
## # ℹ 26,994 more rows
## # ℹ 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>,
## #   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distanc
e <dbl>,
## #   hour <dbl>, minute <dbl>, time_hour <dttm>
```

```
newyear<-filter(flights,month==1, day==1)
newyear
```

```
## # A tibble: 842 × 19
##     year month   day dep_time sched_dep_time dep_delay arr_time sche
d_arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
     <int>
```

```
##  1  2013     1     1      517            515         2      830
       819
##  2  2013     1     1      533            529         4      850
       830
##  3  2013     1     1      542            540         2      923
       850
##  4  2013     1     1      544            545        -1     1004
      1022
##  5  2013     1     1      554            600        -6      812
       837
##  6  2013     1     1      554            558        -4      740
       728
##  7  2013     1     1      555            600        -5      913
       854
##  8  2013     1     1      557            600        -3      709
       723
##  9  2013     1     1      557            600        -3      838
       846
## 10  2013     1     1      558            600        -2      753
       745
## # ℹ 832 more rows
## # ℹ 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>,
## #   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distanc
e <dbl>,
## #   hour <dbl>, minute <dbl>, time_hour <dttm>
```

```
dec25<-filter(flights,month==12, day==25)
dec25
```

```
## # A tibble: 719 × 19
##     year month   day dep_time sched_dep_time dep_delay arr_time sche
d_arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
     <int>
##  1  2013    12    25      456            500        -4      649
       651
##  2  2013    12    25      524            515         9      805
       814
##  3  2013    12    25      542            540         2      832
       850
##  4  2013    12    25      546            550        -4     1022
      1027
##  5  2013    12    25      556            600        -4      730
       745
##  6  2013    12    25      557            600        -3      743
       752
##  7  2013    12    25      557            600        -3      818
       831
##  8  2013    12    25      559            600        -1      855
       856
```

```
##  9  2013    12    25      559            600        -1      849
       855
## 10  2013    12    25      600            600         0      850
       846
## # ℹ 709 more rows
## # ℹ 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>,
## #   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distanc
e <dbl>,
## #   hour <dbl>, minute <dbl>, time_hour <dttm>
```

```
jan_dec<-filter(flights,month==1|month==12)
jan_dec
```

```
## # A tibble: 55,139 × 19
##     year month   day dep_time sched_dep_time dep_delay arr_time sche
d_arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
     <int>
##  1  2013     1     1      517            515         2      830
       819
##  2  2013     1     1      533            529         4      850
       830
##  3  2013     1     1      542            540         2      923
       850
##  4  2013     1     1      544            545        -1     1004
      1022
##  5  2013     1     1      554            600        -6      812
       837
##  6  2013     1     1      554            558        -4      740
       728
##  7  2013     1     1      555            600        -5      913
       854
##  8  2013     1     1      557            600        -3      709
       723
##  9  2013     1     1      557            600        -3      838
       846
## 10  2013     1     1      558            600        -2      753
       745
## # ℹ 55,129 more rows
## # ℹ 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>,
## #   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distanc
e <dbl>,
## #   hour <dbl>, minute <dbl>, time_hour <dttm>
```

```
nov_dec<-filter(flights,month %in% c(11,12))
#detach(flights)
```

The function arrange()

This change the order

```
data_10<-arrange(flights,year,month,day)
arrange(flights,desc(arr_delay))
```

```
## # A tibble: 336,776 × 19
##     year month   day dep_time sched_dep_time dep_delay arr_time sche
d_arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
     <int>
##  1  2013     1     9      641            900      1301     1242
      1530
##  2  2013     6    15     1432           1935      1137     1607
      2120
##  3  2013     1    10     1121           1635      1126     1239
      1810
##  4  2013     9    20     1139           1845      1014     1457
      2210
##  5  2013     7    22      845           1600      1005     1044
      1815
##  6  2013     4    10     1100           1900       960     1342
      2211
##  7  2013     3    17     2321            810       911      135
      1020
##  8  2013     7    22     2257            759       898      121
      1026
##  9  2013    12     5      756           1700       896     1058
      2020
## 10  2013     5     3     1133           2055       878     1250
      2215
## # ℹ 336,766 more rows
## # ℹ 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>,
## #   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distanc
e <dbl>,
## #   hour <dbl>, minute <dbl>, time_hour <dttm>
```

The function select

This helps to selects only the variables you are interested in

```
time_var<-select(flights, year,month, day)
select(flights, year:day)
```

```
## # A tibble: 336,776 × 3
##     year month   day
##    <int> <int> <int>
##  1  2013     1     1
##  2  2013     1     1
##  3  2013     1     1
##  4  2013     1     1
##  5  2013     1     1
##  6  2013     1     1
##  7  2013     1     1
```

```
##  8  2013     1     1
##  9  2013     1     1
## 10  2013     1     1
## # ℹ 336,766 more rows
```

```
select(flights,-(year:day))
```

```
## # A tibble: 336,776 × 16
##    dep_time sched_dep_time dep_delay arr_time sched_arr_time arr_del
ay carrier
##       <int>          <int>     <dbl>    <int>          <int>     <db
l> <chr>
##  1      517            515         2      830            819
11 UA
##  2      533            529         4      850            830
20 UA
##  3      542            540         2      923            850
33 AA
##  4      544            545        -1     1004           1022       -
18 B6
##  5      554            600        -6      812            837       -
25 DL
##  6      554            558        -4      740            728
12 UA
##  7      555            600        -5      913            854
19 B6
##  8      557            600        -3      709            723       -
14 EV
##  9      557            600        -3      838            846
-8 B6
## 10      558            600        -2      753            745
 8 AA
## # ℹ 336,766 more rows
## # ℹ 9 more variables: flight <int>, tailnum <chr>, origin <chr>, de
st <chr>,
## #   air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>, time_h
our <dttm>
```

The function mutate

This is useful in adding a new Variable

```
## create a small data set
flights2<-select(flights,year:day,ends_with("delay"),distance,air_time)
(mutate(flights2,gain=arr_delay-dep_delay,
       speed=distance/air_time*60))
```

```
## # A tibble: 336,776 × 9
##     year month   day dep_delay arr_delay distance air_time  gain spe
ed
##    <int> <int> <int>     <dbl>     <dbl>    <dbl>    <dbl> <dbl> <db
```

```
l>
##  1  2013     1     1          2         11     1400      227      9  37
0.
##  2  2013     1     1          4         20     1416      227     16  37
4.
##  3  2013     1     1          2         33     1089      160     31  40
8.
##  4  2013     1     1         -1        -18     1576      183    -17  51
7.
##  5  2013     1     1         -6        -25      762      116    -19  39
4.
##  6  2013     1     1         -4         12      719      150     16  28
8.
##  7  2013     1     1         -5         19     1065      158     24  40
4.
##  8  2013     1     1         -3        -14      229       53    -11  25
9.
##  9  2013     1     1         -3         -8      944      140     -5  40
5.
## 10  2013     1     1         -2          8      733      138     10  31
9.
## # ℹ 336,766 more rows
```