R for data science class 1

Dr. Pacifique

2025-09-14

Important definitions in data science

Statistics

Modelling

Computation

Calculus

Today's class,we will study

Installation of R

Installation of R studio

Calculator in r

Data importation in R

Data description using numerical measures and graphs.

Data structure (Vectors, Factors, Lists, Data frame, matrix and arrays)

Vector: The foundational R data structure

Numeric

```
x <- c(0.5, 0.6)
age<-c(20,35,32,29)
summary(age)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   20.00   26.75   30.50   29.00   32.75   35.00

age<-c(35,24,18,24)
mean(age)

## [1] 25.25
```

Logical

```
x <- c(TRUE, FALSE)
x <- c(T, F)
```

Character

```
x <- c("a", "b", "c" )
class<-c("M","F","F","M")
```

Integer

```
x <- 9:29
```

Complex

```
x <- c(1+0i, 2+4i)
```

Matrix and operation within matrices

```
x<- matrix(1:6, nrow = 2, ncol = 3)
x

##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    2    4    6

(y<-matrix(1:6, nrow = 3, ncol = 2))

##      [,1] [,2]
## [1,]    1    4
## [2,]    2    5
## [3,]    3    6

x%*%y

##      [,1] [,2]
## [1,]   22   49
## [2,]   28   64
```

List

```
x <- list(1, "a", TRUE, 1 + 4i)
x

## [[1]]
## [1] 1
##
## [[2]]
## [1] "a"
##
## [[3]]
## [1] TRUE
##
## [[4]]
## [1] 1+4i
```

Factor

```
x <- factor(c("yes", "yes", "no", "yes", "no"))
```

Create a vector with NAs in it

```
x <- c(1, 2, NA, 10, 3)
```

Return a logical vector indicating which elements are NA

```
is.na(x)

## [1] FALSE FALSE  TRUE FALSE FALSE

x <- c(1, 2, 4, "NA", 5)
bad <- is.na(x)
print(bad)

## [1] FALSE FALSE FALSE FALSE FALSE

x[!bad]

## [1] "1"  "2"  "4"  "NA" "5"
```

What if there are multiple R objects and you want to take the subset with no missing values in any of those objects?

```
x <- c(1, 2, NA, 4, NA, 5)
y <- c("a", "b", NA, "d", NA, "f")
good <- complete.cases(x, y)
good

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

Coercion

If character is present, in a vector, R convert everything in the vector to character strings.

If a vector only contains logical and numbers, R will convert the logical to numbers, Every true becomes a 1, and every FALSE becomes 0

```
sum(c(TRUE,TRUE,FALSE,FALSE,FALSE))

## [1] 2
```

Create data

Create data Using data frame Data frame is more general than matrix How??? Because data frame can contain different modes of data (Numeric, character and so on) Similar to what you can see in SPSS, SAS,…

###Let's create a data frame

```
studentID<-c(1,2,3,4,5)
math_score<-c(12,17,10,9,NA)
gender<-c("M","F","M","M","F")
it_score<-c(13,18,11,10,19)
scoredata<-data.frame(studentID,gender,math_score,it_score)
scoredata
```

```
##   studentID gender math_score it_score
## 1         1      M         12       13
## 2         2      F         17       18
## 3         3      M         10       11
## 4         4      M          9       10
## 5         5      F         NA       19
```

```
#View(scoredata)
```

Create data from keyboard

Steps 1. Create data frame (or matrix) with variable names 2. Invoke the text editor in the data objected created at first step

```
data_class2<-data.frame(height=numeric(0),weight=numeric(0),bmi=numeric
(0))
data_class2<-edit(data_class2)
```

Data Importation

R has some features that can allow to import data from different sources (It can be text file, spreadsheet, or database)

1. Data from excel
2. Data statistical packages (SAS, SPSS, Stata)
3. Data from Text files (ASCII, XML, Webscraping)
4. Data from database management systems (SQL,MySQL, Oracle, Access)

HW1: Import data from Statistical package and from Database management systems

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

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

```
head(data_class)
```

```
##   HEIGHT WEIGHT
## 1    161     50
## 2    155     49
## 3    158     42
## 4    170     65
## 5    160     60
## 6    156     52
```

```
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[10:20,]
```

```
##    HEIGHT WEIGHT
## 10    167     51
## 11    160     60
## 12    155     42
## 13    154     53
## 14    155     48
## 15    157     48
## 16    157     48
## 17    160     53
## 18    158     52
## 19    160     51
## 20    160     53
```

```
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
```

In case you want to use data set built in R

```
data() # list of datasets currently available
data("airquality")
variable.names(airquality)
```

```
## [1] "Ozone"   "Solar.R" "Wind"    "Temp"    "Month"   "Day"
```

```
str(airquality)
```

```
## 'data.frame':    153 obs. of  6 variables:
##  $ Ozone  : int  41 36 12 18 NA 28 23 19 8 NA ...
##  $ Solar.R: int  190 118 149 313 NA NA 299 99 19 194 ...
##  $ Wind   : num  7.4 8 12.6 11.5 14.3 14.9 8.6 13.8 20.1 8.6 ...
##  $ Temp   : int  67 72 74 62 56 66 65 59 61 69 ...
##  $ Month  : int  5 5 5 5 5 5 5 5 5 5 ...
##  $ Day    : int  1 2 3 4 5 6 7 8 9 10 ...
```

Exploration data analysis

Examine numerical variable using common summary

```
data_class

##    HEIGHT WEIGHT
## 1     161     50
## 2     155     49
## 3     158     42
## 4     170     65
## 5     160     60
## 6     156     52
## 7     162     58
## 8     158     46
## 9     158     45
## 10    167     51
## 11    160     60
## 12    155     42
## 13    154     53
## 14    155     48
## 15    157     48
## 16    157     48
## 17    160     53
## 18    158     52
## 19    160     51
## 20    160     53
## 21    152     44
## 22    154     56
## 23    150     63
## 24    161     52
## 25    162     57
## 26    164     49
## 27    161     52
## 28    155     54
## 29    159     46
## 30    163     50
## 31    159     61
## 32    160     55
## 33    158     45
## 34    165     63
## 35    156     60
## 36    163     56
## 37    155     52
## 38    164     47
## 39    163     52
## 40    168     55
## 41    157     48
## 42    164     58

summary(data_class)
```

```
##      HEIGHT          WEIGHT
##  Min.   :150.0   Min.   :42.0
##  1st Qu.:156.2   1st Qu.:48.0
##  Median :159.5   Median :52.0
##  Mean   :159.4   Mean   :52.4
##  3rd Qu.:162.0   3rd Qu.:56.0
##  Max.   :170.0   Max.   :65.0
```

```
str(data_class)
```

```
## 'data.frame':    42 obs. of  2 variables:
##  $ HEIGHT: int  161 155 158 170 160 156 162 158 158 167 ...
##  $ WEIGHT: int  50 49 42 65 60 52 58 46 45 51 ...
```

Summary Statistics

```
summary(data_class)
```

```
##      HEIGHT          WEIGHT
##  Min.   :150.0   Min.   :42.0
##  1st Qu.:156.2   1st Qu.:48.0
##  Median :159.5   Median :52.0
##  Mean   :159.4   Mean   :52.4
##  3rd Qu.:162.0   3rd Qu.:56.0
##  Max.   :170.0   Max.   :65.0
```

```
apply(data_class,2,mean)
```

```
##    HEIGHT    WEIGHT
## 159.38095  52.40476
```

```
apply(data_class,2,sd)
```

```
##   HEIGHT   WEIGHT
## 4.276723 5.806050
```

```
c(mean(data_class$HEIGHT),sd(data_class$HEIGHT))
```

```
## [1] 159.380952   4.276723
```

```
c(mean(data_class$WEIGHT),sd(data_class$WEIGHT))
```

```
## [1] 52.40476  5.80605
```

```
c(Mean=mean(data_class$HEIGHT),SD=sd(data_class$HEIGHT))
```

```
##       Mean         SD
## 159.380952   4.276723
```

```
c(Mean=mean(data_class$WEIGHT),SD=sd(data_class$WEIGHT))
```

```
##     Mean       SD
## 52.40476  5.80605
```

Variation and covariation using both numerical and graphs

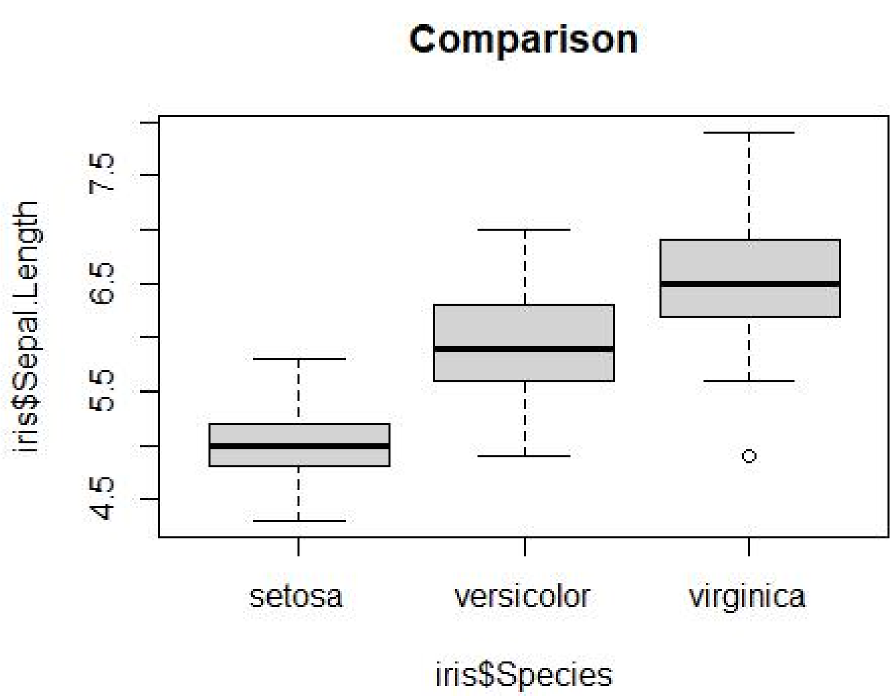
Relationship between a categorical and a continuous variable

```
data("iris")
head(iris)

##   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

mean(iris$Petal.Length)

## [1] 3.758

mean(iris$Petal.Length[iris$Species=="setosa"])

## [1] 1.462

mean(iris$Petal.Length[iris$Species=="versicolor"])

## [1] 4.26

mean(iris$Petal.Length[iris$Species=="virginica"])

## [1] 5.552

## shortcut
by(iris$Petal.Length,iris$Species,mean)

## iris$Species: setosa
## [1] 1.462
## ------------------------------------------------------------
## iris$Species: versicolor
## [1] 4.26
## ------------------------------------------------------------
## iris$Species: virginica
## [1] 5.552

by(iris$Petal.Length,iris$Species,sd)

## iris$Species: setosa
## [1] 0.173664
## ------------------------------------------------------------
## iris$Species: versicolor
## [1] 0.469911
## ------------------------------------------------------------
## iris$Species: virginica
## [1] 0.5518947
```

```
by(iris$Petal.Length,iris$Species,summary)

## iris$Species: setosa
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   1.400   1.500   1.462   1.575   1.900
## --------------------------------------------------------------
## iris$Species: versicolor
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    3.00    4.00    4.35    4.26    4.60    5.10
## --------------------------------------------------------------
## iris$Species: virginica
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   4.500   5.100   5.550   5.552   5.875   6.900
```

Visualize the differences in continuous variables between categories using Box-and-whisker plot

```
boxplot(iris$Sepal.Length~iris$Species,data=iris, main="Comparison")
```



Relationship between two categorical variables

```
ucba<-data.frame(UCBAdmissions)
head(ucba)

##      Admit Gender Dept Freq
## 1 Admitted   Male    A  512
## 2 Rejected   Male    A  313
## 3 Admitted Female    A   89
## 4 Rejected Female    A   19
```

```
## 5 Admitted   Male    B  353
## 6 Rejected   Male    B  207
```

```r
cross<-xtabs(Freq~Gender+Admit,data=ucba)
(cross<-xtabs(Freq~Gender+Admit,data=ucba))
```

```
##         Admit
## Gender   Admitted Rejected
##   Male       1198     1493
##   Female      557     1278
```

```r
## Is there gender bias in UCB graduate admission process?
prop.table(cross,2)
```

```
##         Admit
## Gender    Admitted  Rejected
##   Male   0.6826211 0.5387947
##   Female 0.3173789 0.4612053
```

Simpson's paradox

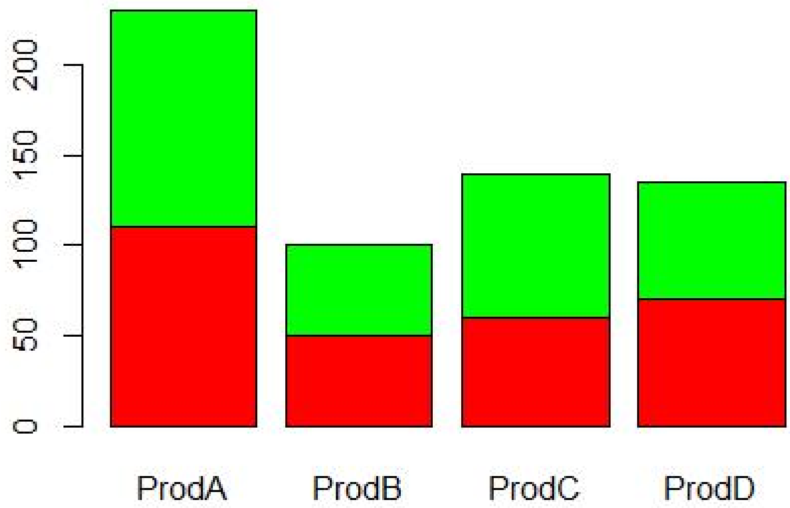
Phenomenon, where a trend that appears in combined groups of data disappears or reverses when broken down into groups.

```r
cross2<-xtabs(Freq~Gender+Admit,data=ucba[ucba$Dept=="A",])
prop.table(cross2,1)
```

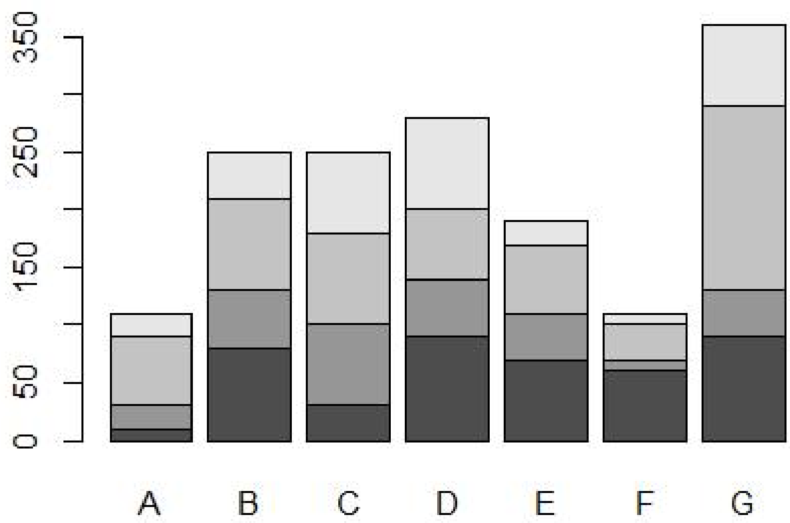
```
##         Admit
## Gender    Admitted  Rejected
##   Male   0.6206061 0.3793939
##   Female 0.8240741 0.1759259
```

Bar Chart

```r
dat <- read.table(text ="ProdA ProdB ProdC ProdD
1 110 50 60 70
2 120 50 80 65", header= TRUE)

  barplot(as.matrix(dat),beside=FALSE,col=c("Red","green"))
```

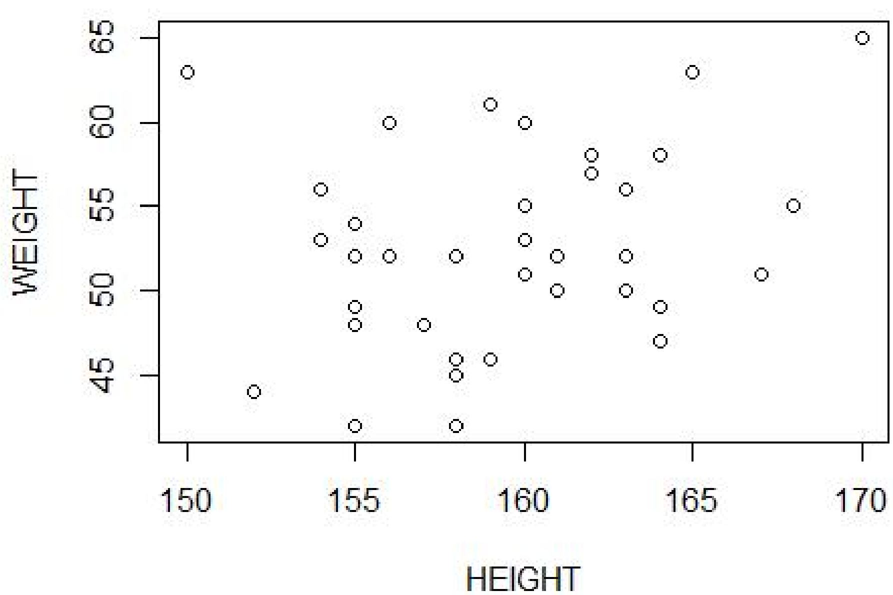



```
#barplot(as.matrix(dat),beside=TRUE,col=c("gold3","red"))

dat <- read.table(text = "A   B   C   D   E   F    G
1 10 80 30 90 70 60  90
2 20 50 70 50 40 10  40
3 60 80 80 60  60 30 160
4 20 40 70 80 20 10  70", header = TRUE)

barplot(as.matrix(dat))
```

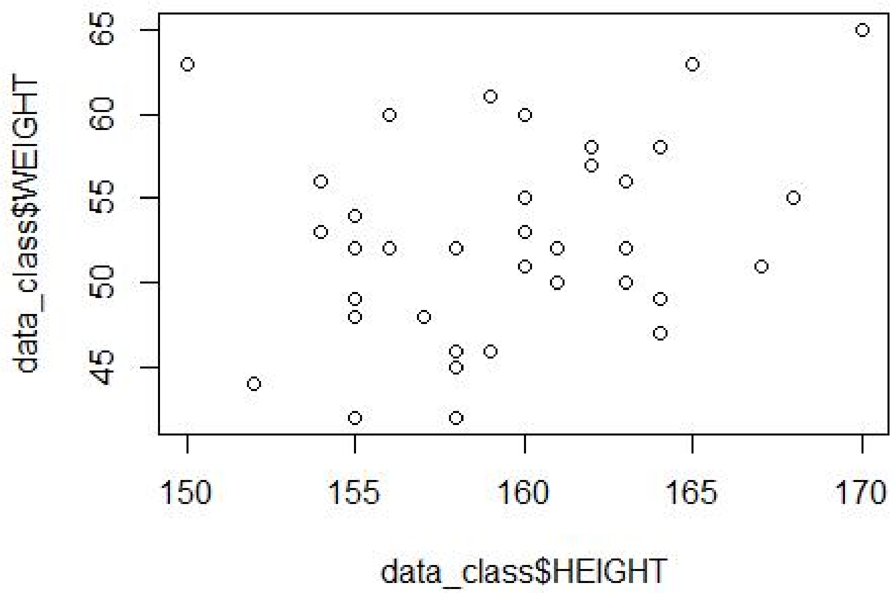


Scatter Plot

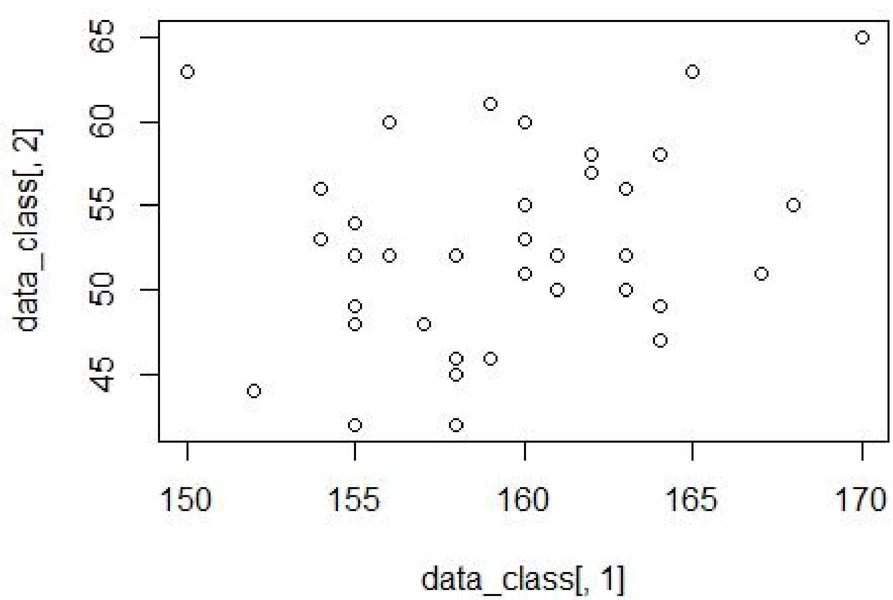
```
plot(WEIGHT~HEIGHT,data=data_class)
```



```r
plot(data_class$HEIGHT,data_class$WEIGHT)
```



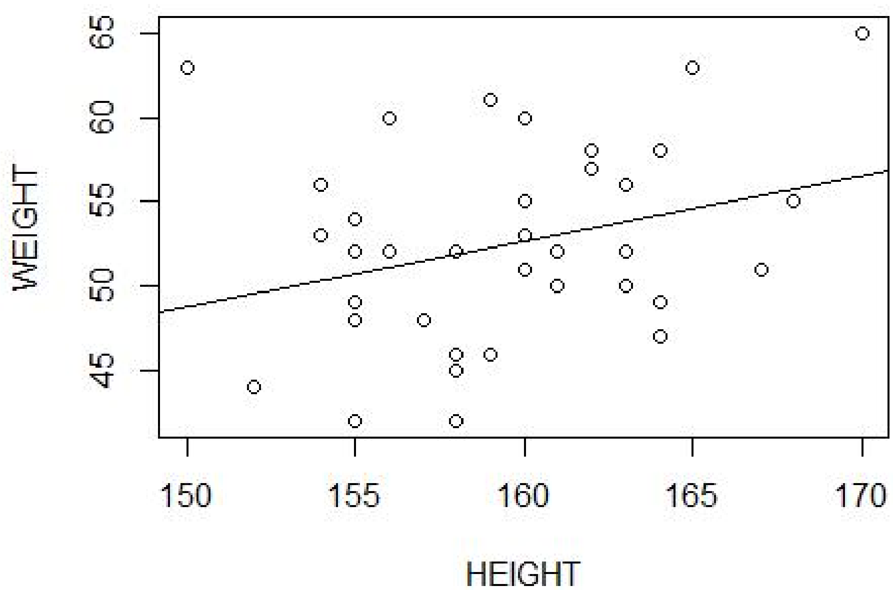

```r
plot(data_class[,1],data_class[,2])
```



```r
cor(data_class$HEIGHT,data_class$WEIGHT)

## [1] 0.2853684
```

Linear Models

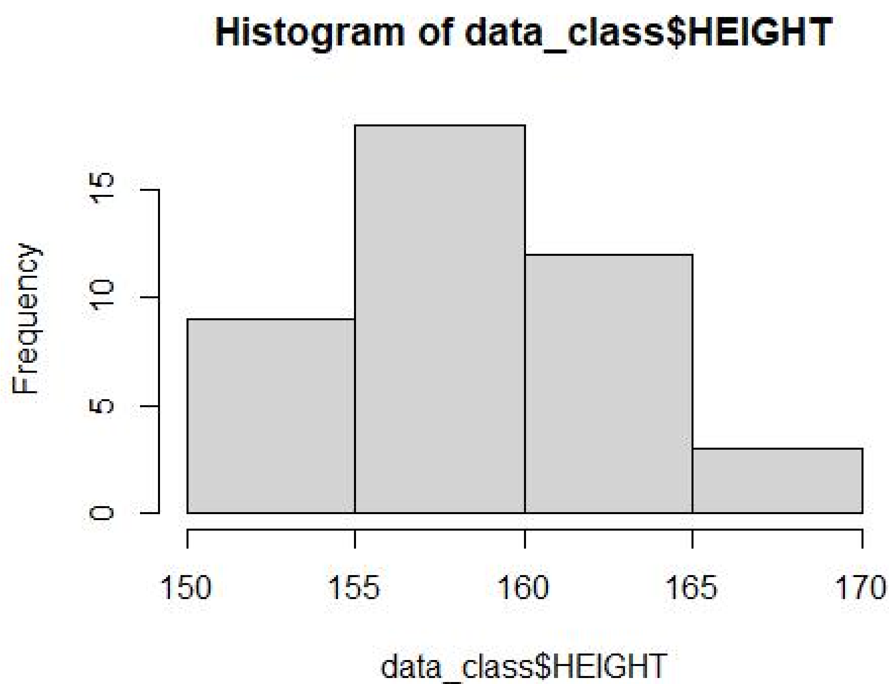
```r
plot(WEIGHT~HEIGHT,data=data_class)
abline(lm(WEIGHT~HEIGHT,data=data_class)$coefficient)
```



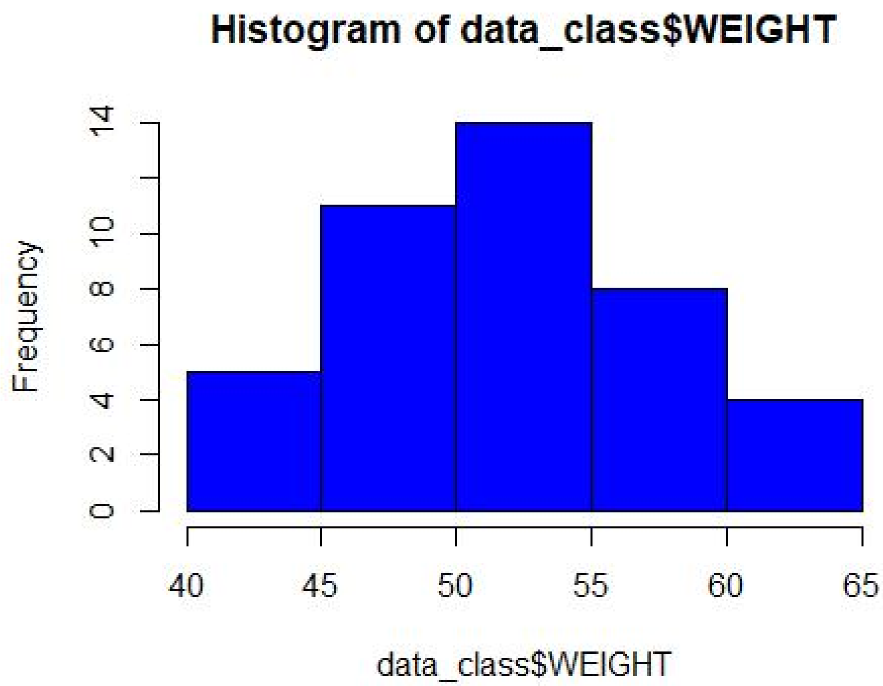

```r
data_lm<-lm(WEIGHT~HEIGHT,data=data_class)
summary(data_lm)

## 
## Call:
## lm(formula = WEIGHT ~ HEIGHT, data = data_class)
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
## -9.870 -3.726 -0.888  3.509 14.230 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)  
## (Intercept)  -9.3417    32.8007  -0.285    0.777  
## HEIGHT        0.3874     0.2057   1.883    0.067 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 5.634 on 40 degrees of freedom
```

```
## Multiple R-squared:  0.08144,    Adjusted R-squared:  0.05847
## F-statistic: 3.546 on 1 and 40 DF,  p-value: 0.06697
```

```r
hist(data_class$HEIGHT)
```

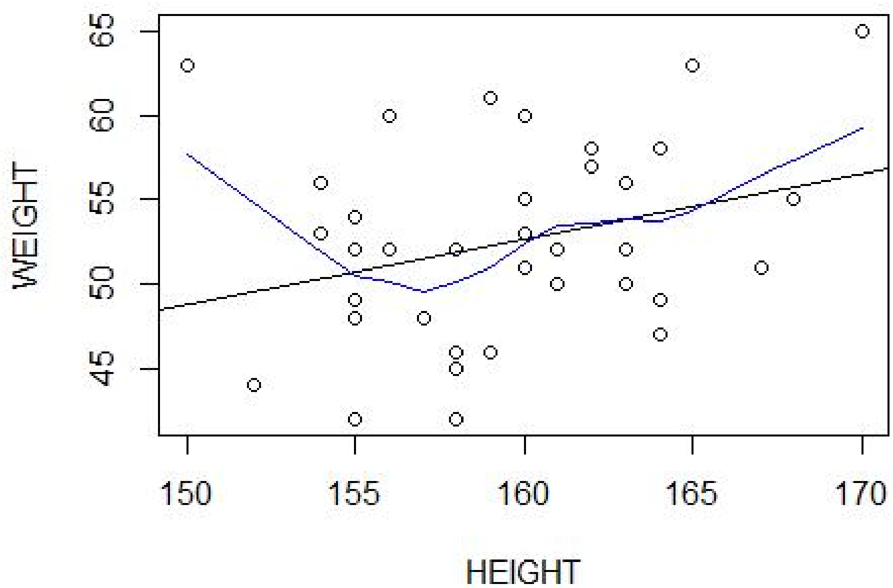



```r
hist(data_class$WEIGHT,col = "blue")
```



Linear with other line

```
plot(WEIGHT~HEIGHT,data=data_class)
abline(lm(WEIGHT~HEIGHT,data=data_class)$coefficient)
lines(lowess(data_class$HEIGHT,data_class$WEIGHT),col="blue")
```




```
cor(data_class$HEIGHT,data_class$WEIGHT)

## [1] 0.2853684

model1<-lm(WEIGHT~HEIGHT,data=data_class)
```

Data management

Create another Variable

```
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=2))

##  [1] 19.29 20.40 16.82 22.49 23.44 21.37 22.10 18.43 18.03 18.29 23.
44 17.48
```

```
## [13] 22.35 19.98 19.47 19.47 20.70 20.83 19.92 20.70 19.04 23.61 28.
00 20.06
## [25] 21.72 18.22 20.06 22.48 18.20 18.82 24.13 21.48 18.03 23.14 24.
65 21.08
## [37] 21.64 17.47 19.57 19.49 19.47 21.56
```

```r
head(cbind(data_class,BMI),n=5)
```

```
##   HEIGHT WEIGHT   BMI
## 1    161     50 19.29
## 2    155     49 20.40
## 3    158     42 16.82
## 4    170     65 22.49
## 5    160     60 23.44
```

```r
new_data_class<-cbind(data_class,BMI)
tail(cbind(data_class,BMI),n=10)
```

```
##    HEIGHT WEIGHT   BMI
## 33    158     45 18.03
## 34    165     63 23.14
## 35    156     60 24.65
## 36    163     56 21.08
## 37    155     52 21.64
## 38    164     47 17.47
## 39    163     52 19.57
## 40    168     55 19.49
## 41    157     48 19.47
## 42    164     58 21.56
```

Summary of BMI

```r
summary(BMI)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   16.82   19.10   20.23   20.64   22.00   28.00
```

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

```
## 
##   The decimal point is at the |
## 
##   16 | 8
##   17 | 55
##   18 | 0022348
##   19 | 03555569
##   20 | 0114778
##   21 | 145667
##   22 | 1455
##   23 | 1446
##   24 | 17
##   25 |
```

```
##    26 |
##    27 |
##    28 | 0
```