FSAN815/ELEG815: Foundations of Statistical Learning

Gonzalo R. Arce

Chapter 0: Introduction

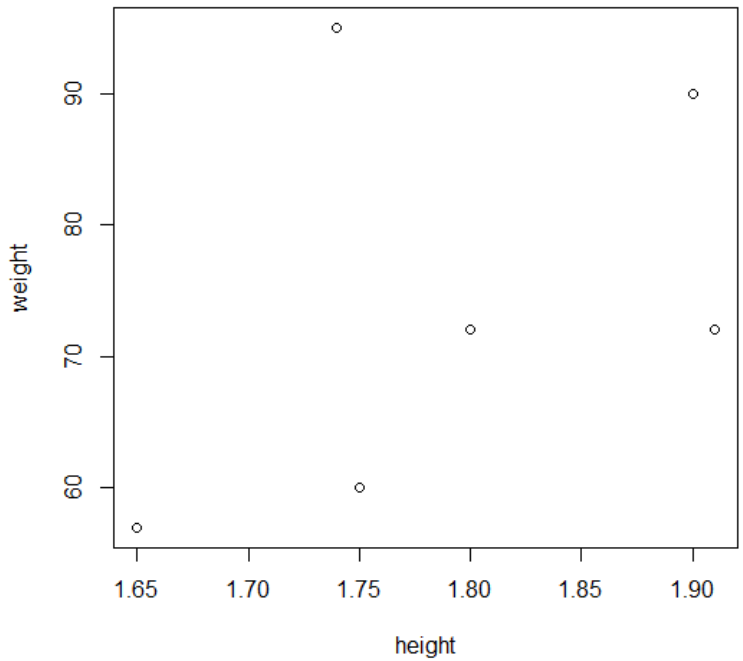
Fall 2014

First Step

- A calculator
- Assignments
  - To variables $x < -2$
  - To vectors $weight < -c(60, 72, 57, 90, 95, 72)$
- Mean: $mean(weight)$
- Standard deviation: $sd(weight)$

Graphics

A simple example:
$height < -c(1.75, 1.80, 1.65, 1.90, 1.74, 1.91)$ $plot(height, weight)$



R Language Essentials

- Expression evaluation: The user enters an expression; the system evaluates it and prints the result.
- Object: Expressions work on objects. Anything that can be assigned to a variable.
- Functions and arguments: A function name is followed by a set of parentheses containing one or more arguments.

Vectors

- Character Vector: c("Huey","Dewey","Louie")
- Logical Vector: c(T,T,F,T)
- Numeric Vector: c(1,2,3,4)

- Missing value: NA
- C function:
  x <- c(1, 2, 3)
  y <- c(10, 20)
  c(x, y, 5)

- Assign names to elements: x<-c(red="Huey", blue="Dewey", green="Louie")
- Get names: names(x)

Matrix and Factors

- Matrix
  - matrix function:
    matrix(1:12,nrow=3,byrow=T)
  - x <- 1:12
    dim(x) <- c(3,4)

- Factor: Used with categorical variables, indicating some subdivision of data, such as social class, primary diagnosis...
  pain <- c(0,3,2,2,1)
  fpain <- factor(pain,levels=0:3)
  levels(fpain) <- c("none","mild","medium","severe")

List and Data Frame

- List: To combine a collection of objects into a larger composite object:
  An example list concerning pre- and postmenstrual energy intake in a group of women:
  intake.pre <-
  c(5260,5470,5640,6180,6390, 6515,6805,7515,7515,8230,8770)
  intake.post <-
  c(3910,4220,3885,5160,5645,4680,5265,5975,6790,6900,7335)
  mylist <- list(before=intake.pre,after=intake.post)

- Data frame: A list of vectors and/or factors of the same length that are related 'across'.
  d <- data.frame(intake.pre,intake.post)
  As with lists, components can be accessed using the $ notation:
  d$intake.pre

Indexing and Conditional Selection

- Indexing:
  intake.pre[5]
  intake.pre[c(3,5,7)]
  intake.pre[-c(3,5,7)]
- Conditional selection:
  intake.post[intake.pre > 7000]
  intake.post[intake.pre > 7000 & intake.pre <= 8000]
- Comparison operators: <, >, ==, <=, >=, and !=
- logical operators: &, |, and !

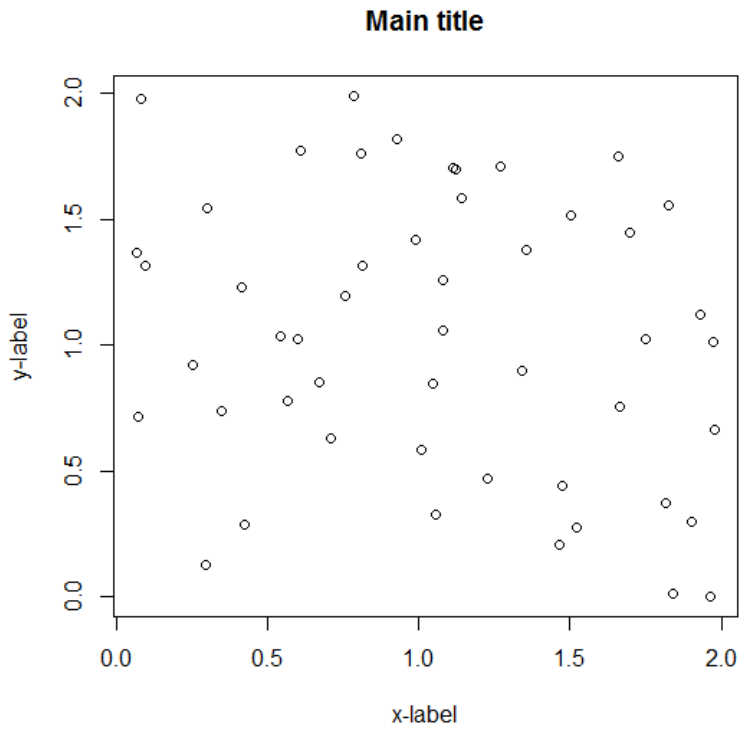
The R Environment

- List: ls()
- Remove: rm()
- Save the workplace: sink("myfile")
- Script: work with a script editor window
- Save and load history: savehistory, loadhistory
- Get help: help()
- Load package: library()

The graphics subsystem

- Plot layout:

```
x <- runif(50,0,2)
y <- runif(50,0,2)
plot(x, y, main="Main title", xlab="x-label", ylab="y-label")
```



R Programming

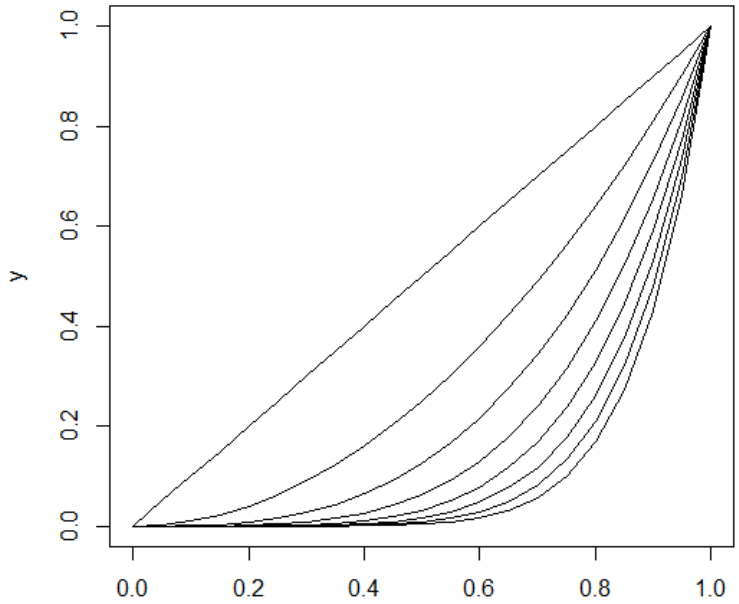
- Function:
  hist.first100 <- function(x)
  {
  x=x[1:100]
  hist(x)
  }
- Flow control:

- while:
  y <- 12345
  x <- y/2
  while (abs(x*x-y) > 1e-10) x <- (x + y/x)/2
- Repeat and Break
  x <- y/2 > repeat
  x <- (x + y/x)/2
  if (abs(x*x-y) < 1e-10) break

Flow Control

- if
  if (all(abs(x*x - y) < 1e-10)) break

- for
  x <- seq(0, 1, .05)
  plot(x, x, ylab='y', type='l')
  for (j in 2 : 8) lines(x, $x \wedge j$)



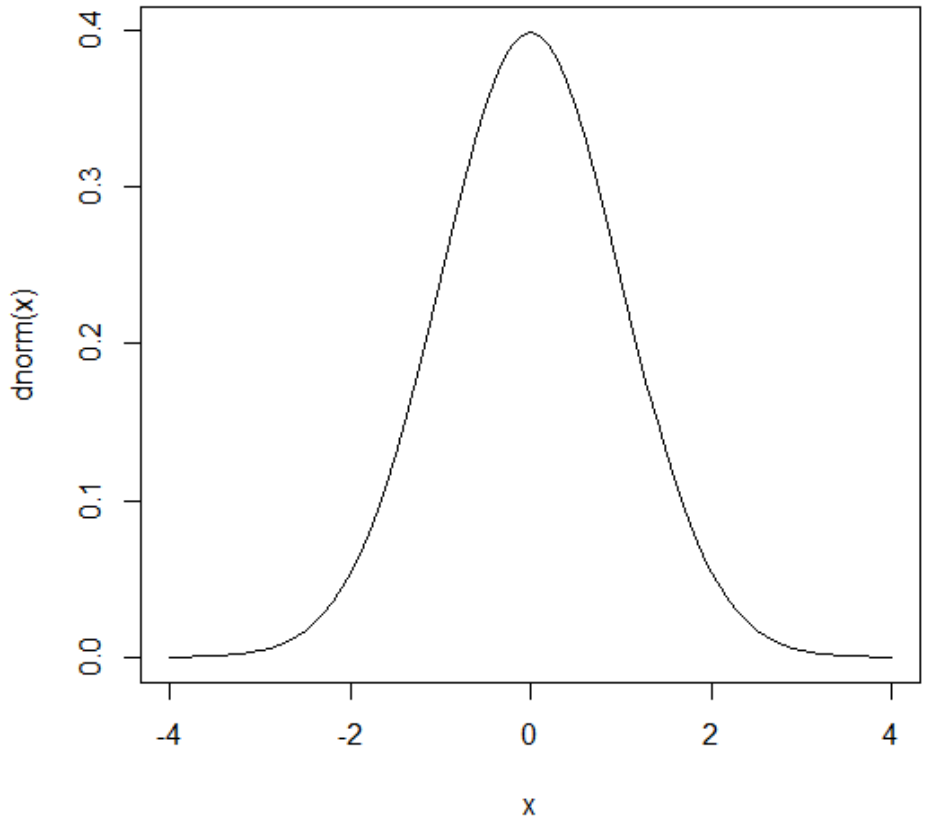
Probability and Distribution

- Random Sample:
  sample(1:40,5) : get 5 numbers randomly from 1 to 40
  sample(c("succ", "fail"), 10, replace=T, prob=c(0.9, 0.1))

- Probability calculations:
  1/prod(40:36) for a given order
- choose(): Number of ways to choose a set of numbers.
  1/choose(40,5) for choose a set of 5 numbers.

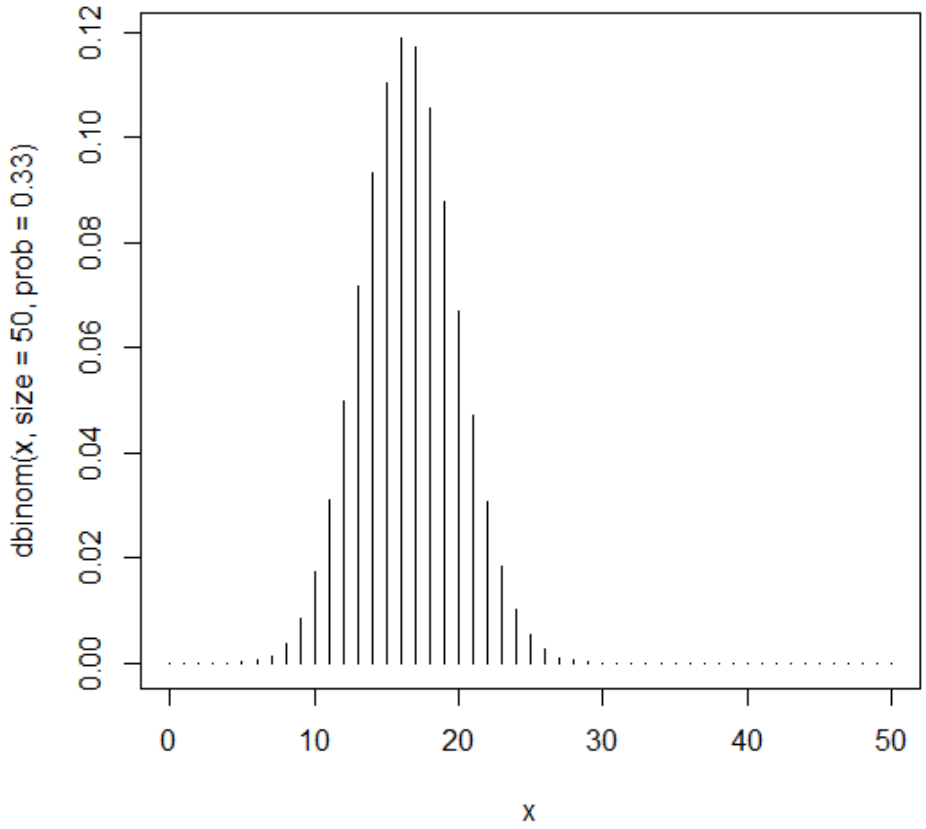
Built-in Distributions

- Density:
  x <- seq(-4,4,0.1)
  plot(x,dnorm(x),type="l")



Built-in Distributions

- Discrete Distribution:
  x <- 0:50
  plot(x,dbinom(x,size=50,prob=.33),type="h")



Cumulative Distribution Function and Quantiles

- Cumulative distribution function
  pnorm(160,mean=132,sd=13)
  pbinom(16,size=20,prob=.5)
- Quantiles
  qnorm(0.5,mean=150,sd=20)
- Random Numbers
  rnorm(10,mean=7,sd=5)
  rbinom(10,size=20,prob=.5)

Summary Statistics For A Single Group

- Mean: mean()
- Standard Deviation: sd()
- Variance: var()
- Median: median()
- Missing value:
  attach(juul)
  mean(igf1,na.rm=T)
  sum(!is.na(igf1))
- Summary:
  summary(igf1)
  summary(juul)

Summary Statistics For A Single Group

- Summary:

```
detach(juul)
juul$sex <- factor(juul$sex,labels=c("M","F"))
juul$menarche <- factor(juul$menarche,labels=c("No","Yes"))
juul$tanner <- factor(juul$tanner,labels=c("I","II","III","IV","V"))
attach(juul)
summary(juul)
```

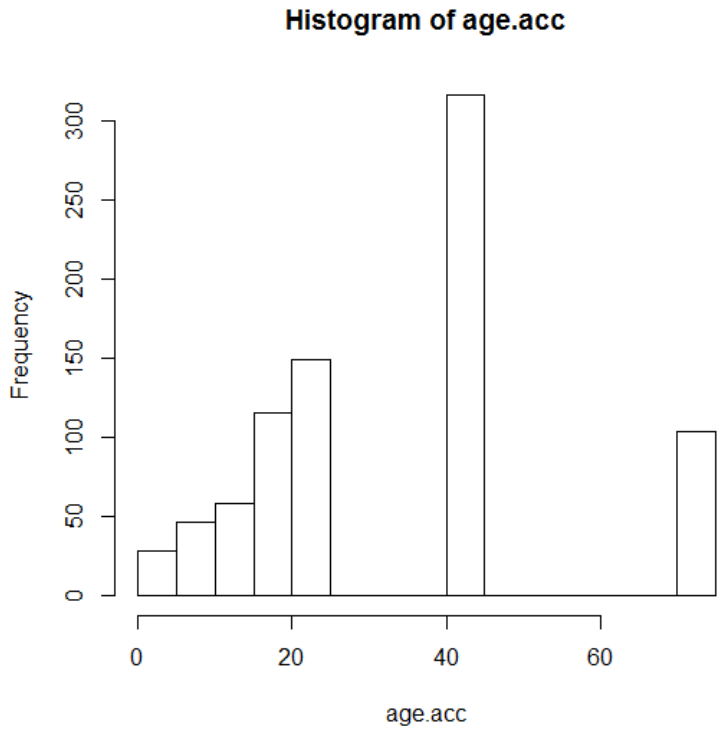
- Transform():

```
juul <- transform(juul,sex=factor(sex,labels=c("M","F")),
+ menarche=factor(menarche,labels=c("No","Yes")),
+ tanner=factor(tanner,labels=c("I","II","III","IV","V")))
```

Graphical Display of Distributions

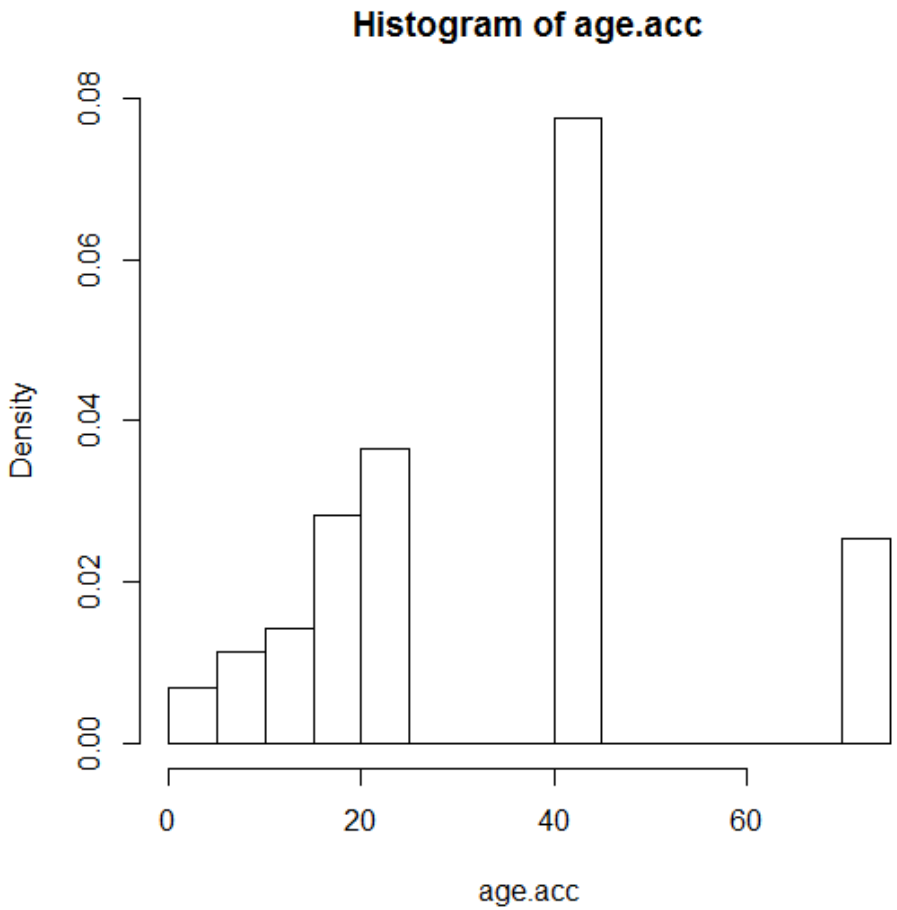
- Histogram

```
mid.age <- c(2.5,7.5,13,16.5,17.5,19,22.5,44.5,70.5)
acc.count <- c(28,46,58,20,31,64,149,316,103)
age.acc <- rep(mid.age,acc.count)
hist(age.acc)
```



Graphical Display of Distributions

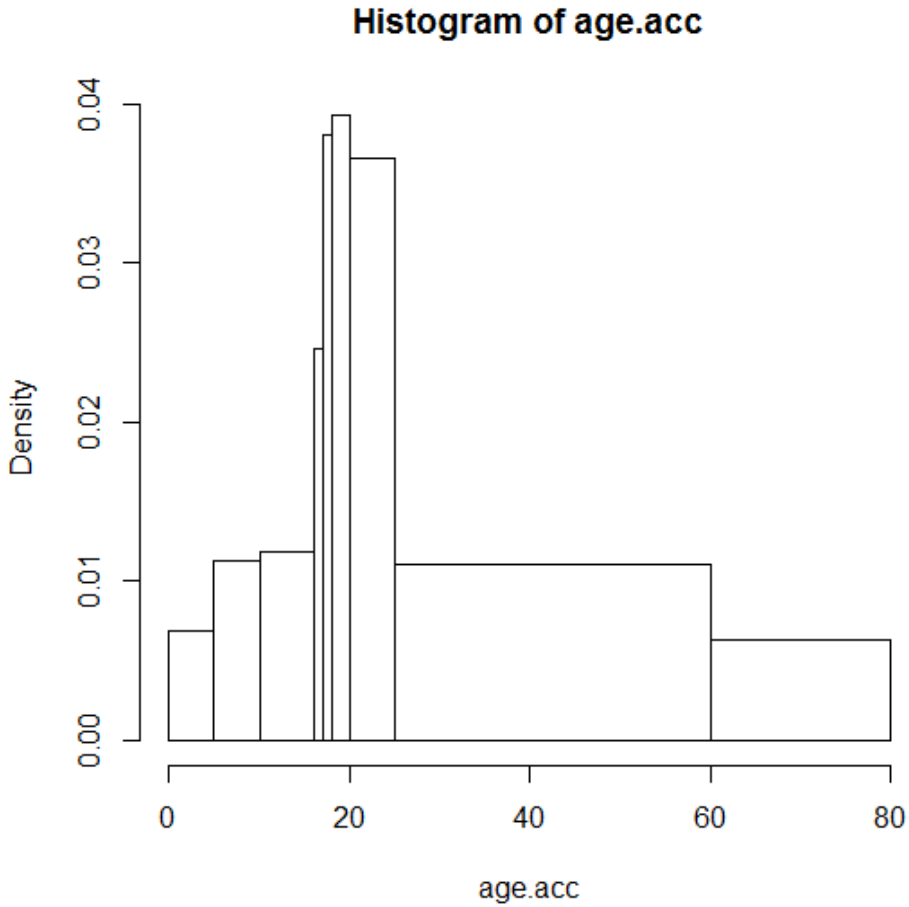
- Histogram
  hist(age.acc,freq=F)



Graphical Display of Distributions

- Histogram

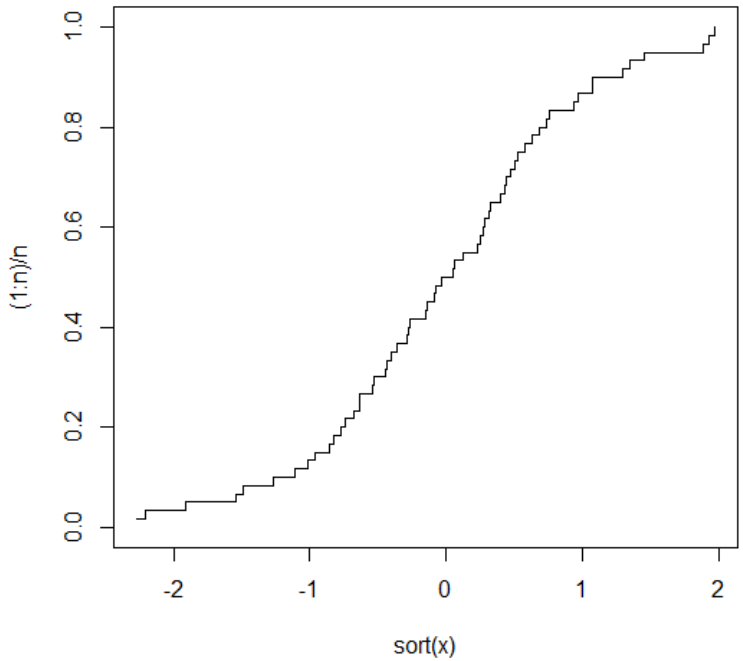
  brk <- c(0,5,10,16,17,18,20,25,60,80)
  hist(age.acc,breaks=brk)



Empirical Cumulative Distribution

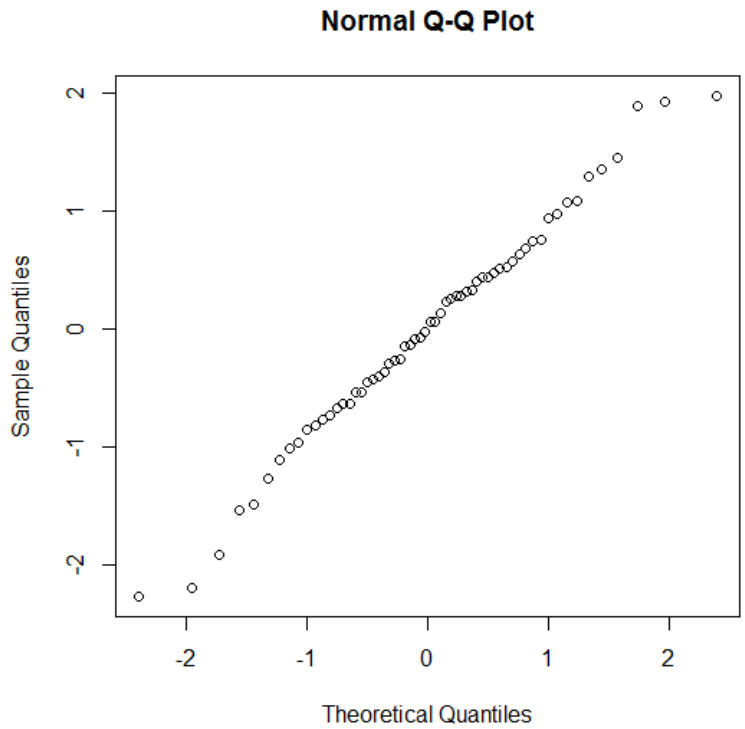
- Empirical cumulative distribution:
  The fraction of data smaller than or equal to x

```
x <- rnorm(60)
n <- length(x)
plot(sort(x),(1:n)/n,type="s",ylim=c(0,1))
```



Q-Q Plots

- Q-Q plots:
  Plot the *k*th smallest observation against the expected value of the *k*th smallest observation out of *n* in a standard normal distribution.
  qqnorm(x)

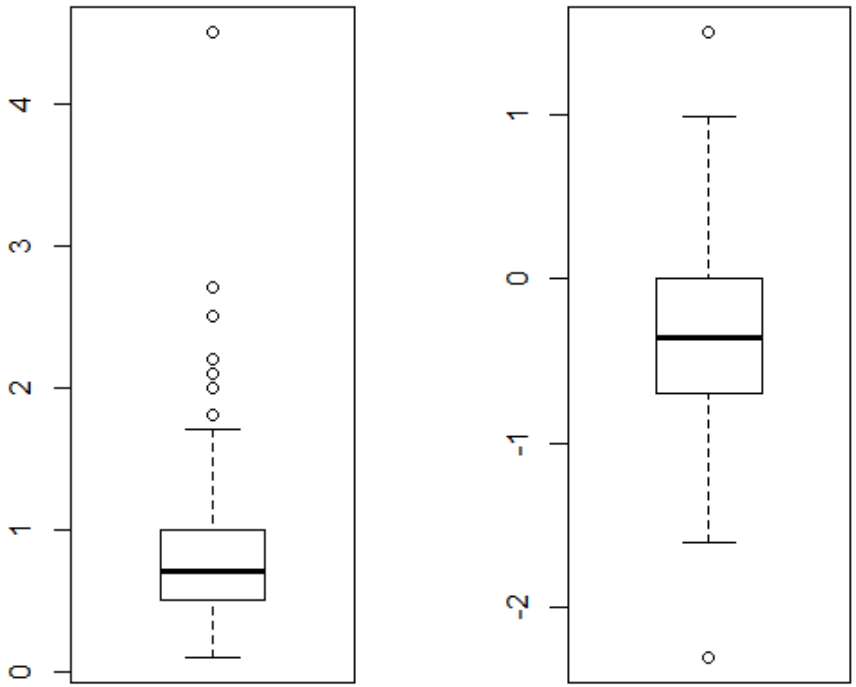


Boxplots

- Hinge:
  The lower hinge is the median of the lower half of the data up to and including the median.
  The upper hinge is the median of the upper half of the data up to and including the median.
- Whiskers:
  Largest or smallest observation that falls within a distance of 1.5 times the box size from the nearest hinge.
- Boxplot():
  par(mfrow=c(1,2))
  boxplot(IgM)
  boxplot(log(IgM))
  par(mfrow=c(1,1))

Boxplots

- Hinge
- Whiskers
- Boxplot()



Load Data in R

- Load data into R as a data frame:
  mydata=read.table("C:/Users/delll/Desktop/datahwk1.txt")
- Convert it to the form of a matrix:
  mydata <- as.matrix(mydata)
- Delete the column names:
  mydata <- matrix(mydata, ncol = ncol(mydata), dimnames = NULL)
- Take the first row:
  x=mydata[1,]
- Plot the data:
  plot(x,type='l')