

Probability with *R*:

An Introduction

with

Computer Science Applications

Jane M Horgan *

*Professor Jane M Horgan, School of Computing, Dublin City University; email: jhorgan@computing.dcu.ie

1 The *R* Language

Installing *R*

- Go to the Cran website at <http://cran.r-project.org/>
- Click 'Download and Install *R* '
- Choose an operating system e.g. Windows;
- Choose the 'base' package
- Select the Download program
- Press the option 'Run' and continue

R is now installed.

R Documentation

<http://cran.r-project.org/manuals>

- An Introduction to *R*
- The *R* Language Definition
- Writing *R* Extensions
- *R* Data Import/Export
- *R* Installation and Administration.
- *R* Internals
- The *R* Reference Index

Basics

- `6+7*3/2`
`[1] 16.5`
- `x <- 1:4`
`x`
`[1] 1 2 3 4`
- `x2 <- x**2`
`x2`
`[1] 1 4 9 16`
- `X <- 10`
`prod1 <- X*x`
`prod1`
`[1] 10 20 30 40`
- **Objects:** The entities that *R* creates and manipulates, e.g. variables, arrays, strings, functions.
- **Workspace:** All objects created in *R* are stored in *workspace*

Getting Help

- click the *Help* button on the toolbar.
- `help()`
- `help.start()`
- `demo()`
- `?read.table`
- `help.search ("data.entry")`
- `apropos (boxplot}`
`"boxplot", "boxplot.default", "boxplot.stat`

Data Entry and Summary

Entering data from the screen to a vector

Example: 1.1

```
downtime <-c(0, 1, 2, 12, 12, 14, 18, 21, 21, 23,  
            24,25,28,29,30,30,30,33,36,44,45,47,51)
```

```
mean(downtime)
```

```
[1] 25.04348
```

```
median(downtime)
```

```
[1] 25
```

```
range(downtime)
```

```
[1] 0 51
```

```
sd(downtime)
```

```
[1] 14.27164
```

Data Entry

Entering data from a file to a data frame

Example 1.2: Examination results: *results.txt*

gender	arch1	prog1	arch2	prog2
m	99	98	83	94
m	NA	NA	86	77
m	97	97	92	93
m	99	97	95	96
m	89	92	86	94
m	91	97	91	97
m	100	88	96	85
f	86	82	89	87

and so on

NA indicates missing value.

No mark for *arch1* and *prog1* in second record.

```
results <-read.table ('G:/data/results.txt' , header = T)
```

```
results$arch1[5]
```

```
[1] 89
```

Alternatively

```
attach(results)
```

```
names(results)
```

allows you to access without prefix *results*.

```
arch1[5]
```

```
[1] 89
```

Missing Values

```
mean(arch1)
[1] NA
```

No result because some marks are missing.

```
na.rm = T (not available, remove)
or
na.rm = TRUE
```

```
mean(arch1, na.rm = T)
[1] 63.56897
```

```
mean(prog1, na.rm = T)
[1] 59.01709
```

```
mean(arch2, na.rm = T)
[1] 51.97391
```

```
mean(prog2, na.rm = T)
[1] 53.78378
```

```
mean(results, na.rm = T)
gender    arch1    prog1    arch2    prog2
  NA 63.56897 59.01709 51.97391 53.78378
```

Summary Statistics:

- **Measures of Central Tendency:** Typical or central points:
 - *Mean:* Sum of all values divided by the number of cases.
 - *Median:* Middle value. 50% of data below and 50% above.
 - *Mode:* Most commonly occurring value, value with the highest frequency.

- **Measures of Dispersion:** Spread or variation in the data.
 - *Standard Deviation (sd):* The square root of the average squared deviations from the mean.
measures how the data values differ from the mean. A small standard deviation implies most values are near the average. A large standard deviation indicates that values are widely spread above and below the average.
 - *Range:* Lowest and highest values.
 - *Quartiles:* Divides data into quarters. 2nd quartile is median
 - *Interquartile Range:* 1st and 3rd quartiles, middle 50% of the data.

Summary Statistics

Downtime:

```
summary(downtime)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.00	16.00	25.00	25.04	31.50	51.00

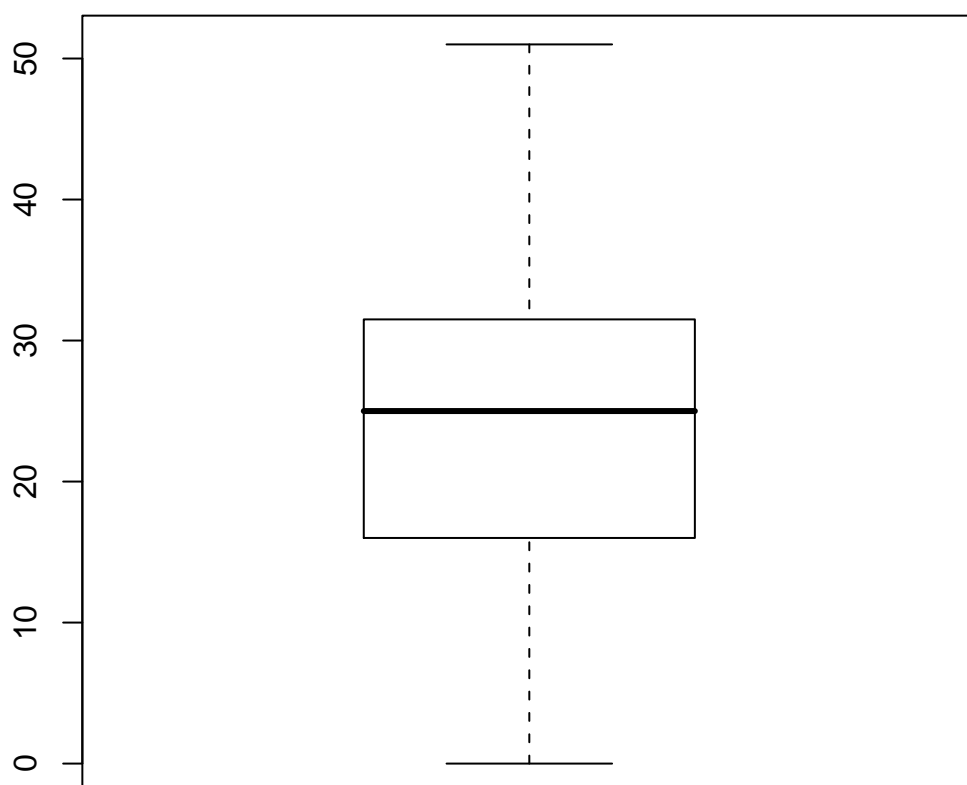
Examination Results:

```
summary(results)
```

gender	arch1	prog1	arch2	pro
f: 19	Min. : 3.00	Min. :12.00	Min. : 6.00	Min.
m:100	1st Qu.: 46.75	1st Qu.:40.00	1st Qu.:40.00	1st Qu.
	Median : 68.50	Median :64.00	Median :48.00	Median
	Mean : 63.57	Mean :59.02	Mean :51.97	Mean
	3rd Qu.: 83.25	3rd Qu.:78.00	3rd Qu.:61.00	3rd Qu.
	Max. :100.00	Max. :98.00	Max. :98.00	Max.
	NA's : 3.00	NA's : 2.00	NA's : 4.00	NA's

Graphical Displays: Boxplots

```
boxplot(downtime)
```



Graphical Displays: Boxplots

- *Boxplot (Box and Whiskers Plot)*

A graphical summary based on the median, quartiles and extreme values.

Box represents the interquartile range which contains 50% of cases.

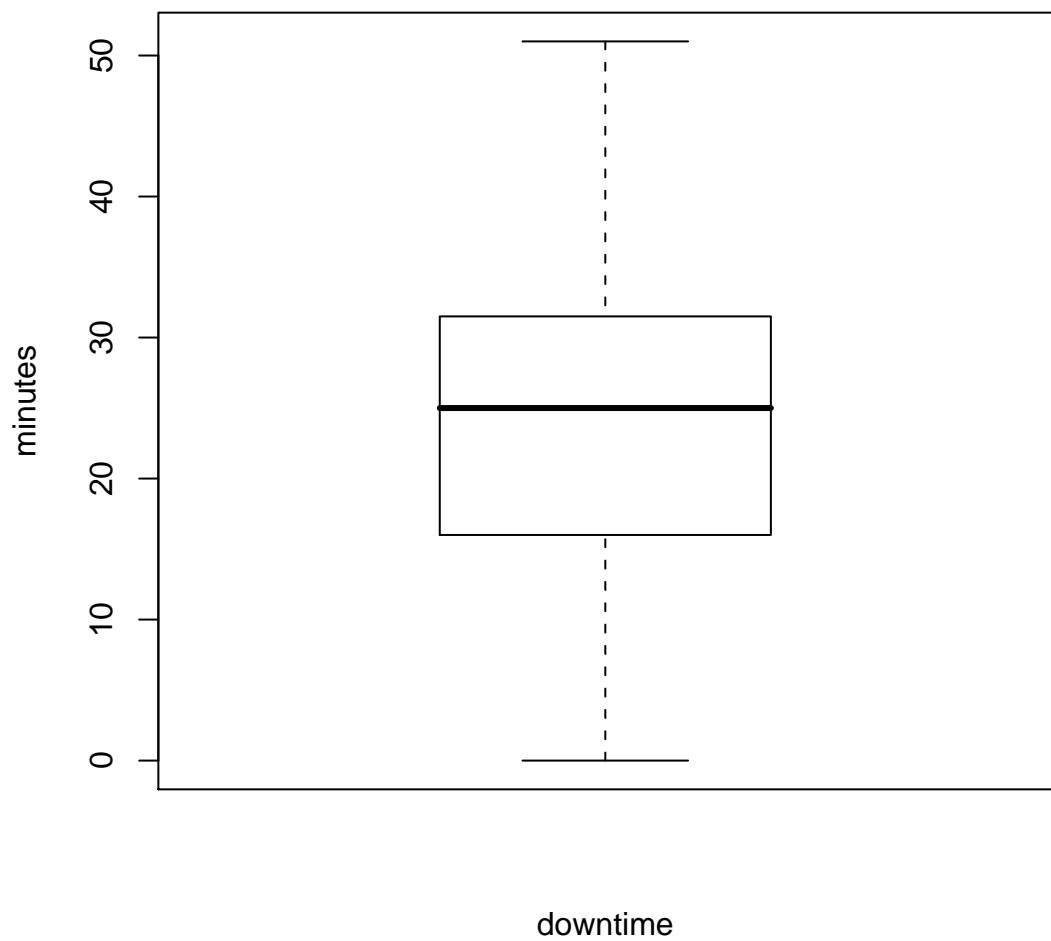
Whiskers are lines that extend from the box to the highest and lowest values.

Line across the box indicates the median.

Extreme values are cases more than 1.5 box lengths from the upper or lower end of the box.

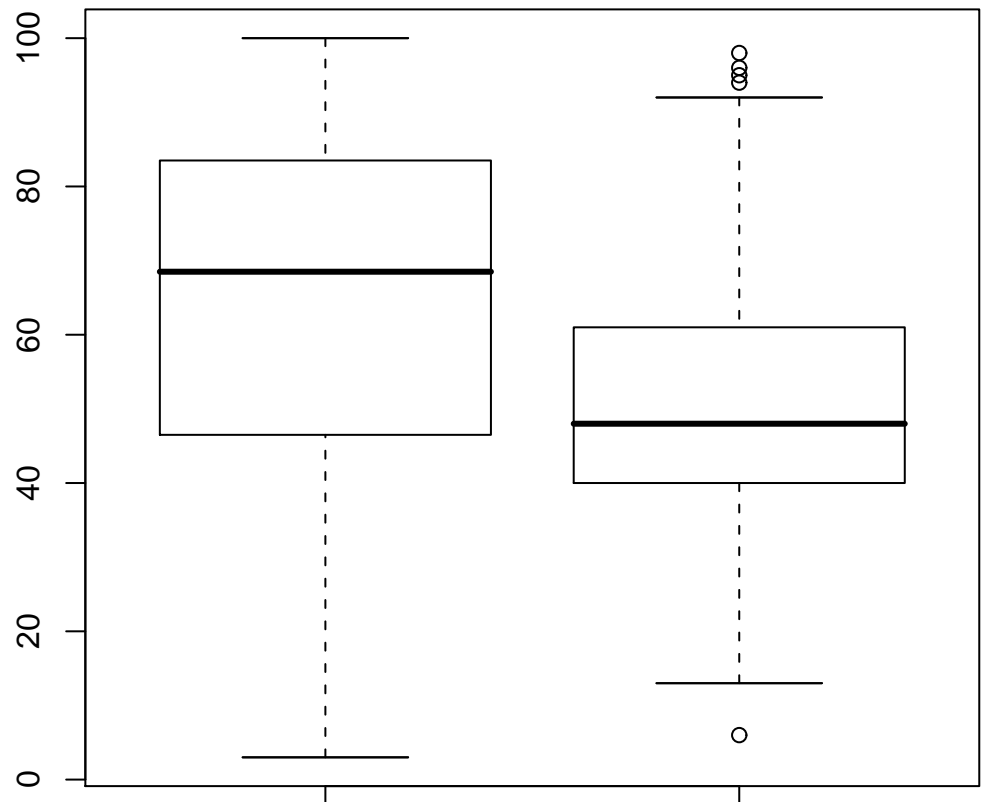
Graphical Displays: Boxplots

```
boxplot(downtime,  
        xlab = "downtime",  
        ylab = "minutes")
```



Graphical Displays: Multiple Boxplots

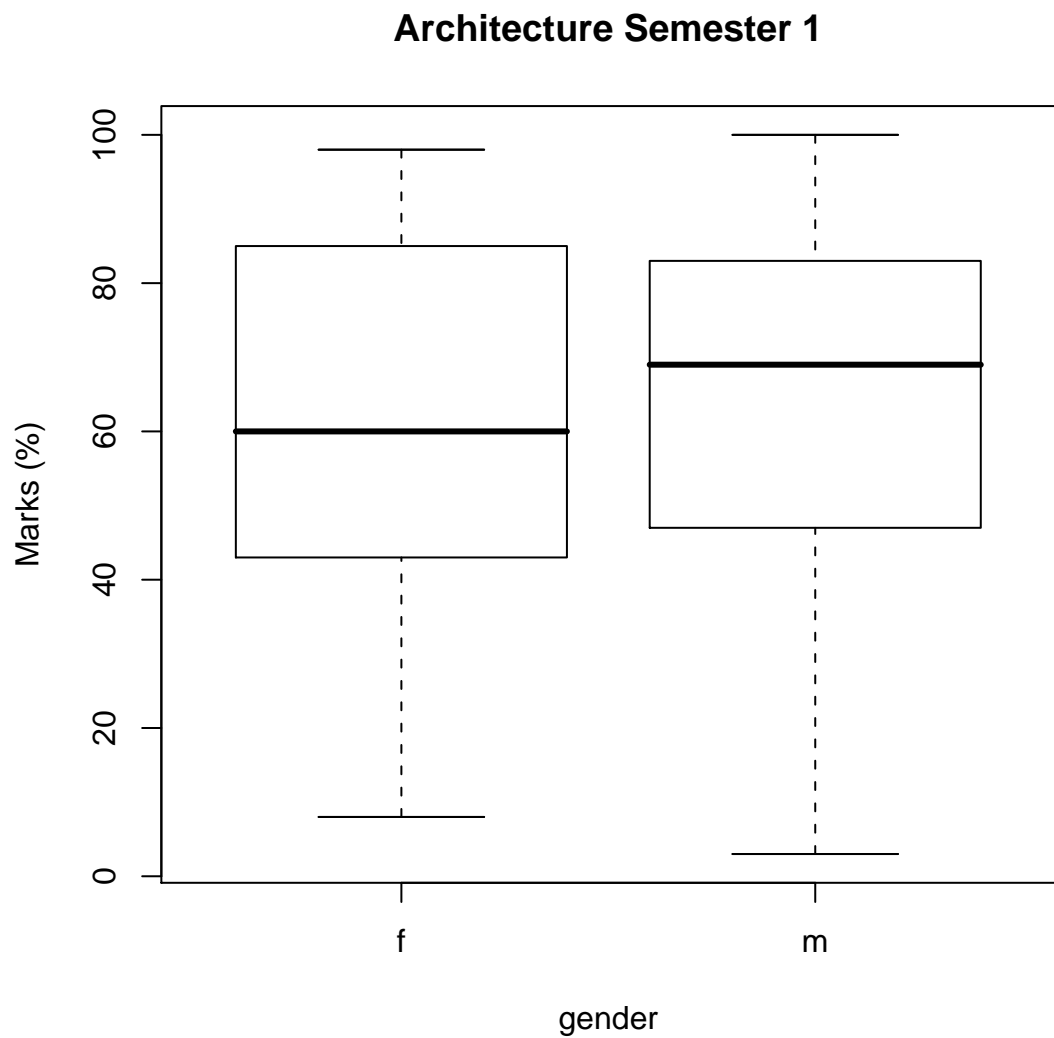
```
boxplot(arch1, arch2,  
        xlab="Architecture Semesters 1, and 2")
```



Architecture, Semesters 1, and 2

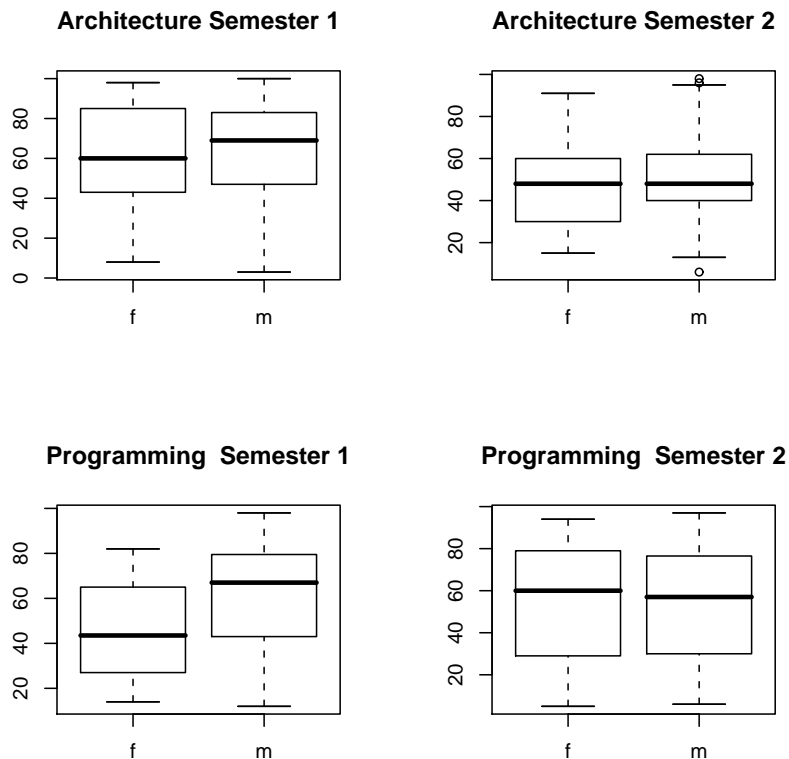
Graphical Displays: Multiple Boxplots

```
boxplot(arch1~gender,  
        xlab = "gender",  
        ylab = "Marks (%)",  
        main = "Architecture Semester 1")
```



par

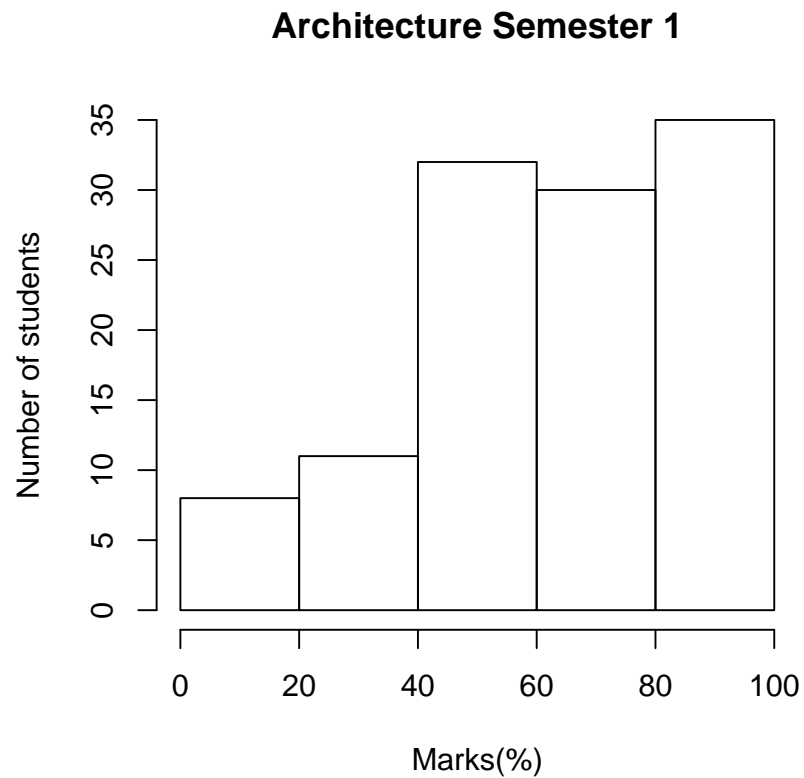
```
par (mfrow = c(2,2))
boxplot (arch1~gender,
        main = "Architecture Semester 1")
boxplot(arch2~gender,
        main = "Architecture Semester 2")
boxplot(prog1~gender,
        main = "Probability Semester 1")
boxplot(prog2~gender,
        main = "Probability Semester 2")
```



par (mfrow = c(1,1)) restores full screen.

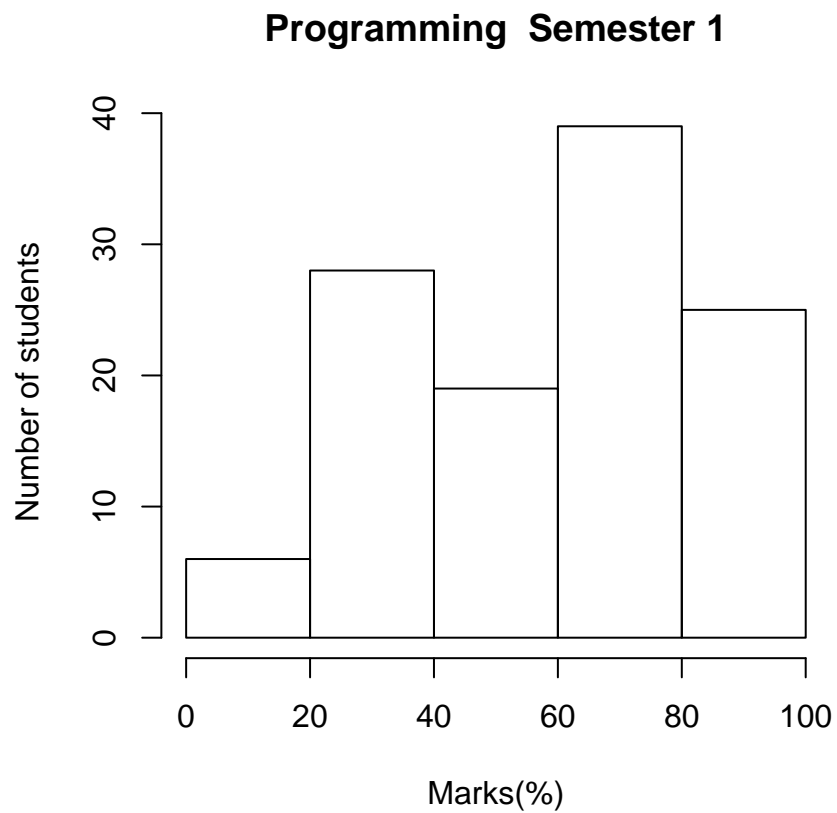
Histograms

```
hist(  
  arch1, breaks = 5,  
  xlab = "Marks(%)",  
  ylab = "Number of students",  
  main = "Architecture Semester 1"  
)
```



Histograms

```
hist(prog1, "breaks = 5",  
     xlab = "marks(%)",  
     ylab = "Number of Students",  
     main = "Architecture Semester 1"  
    )
```



Histograms

```

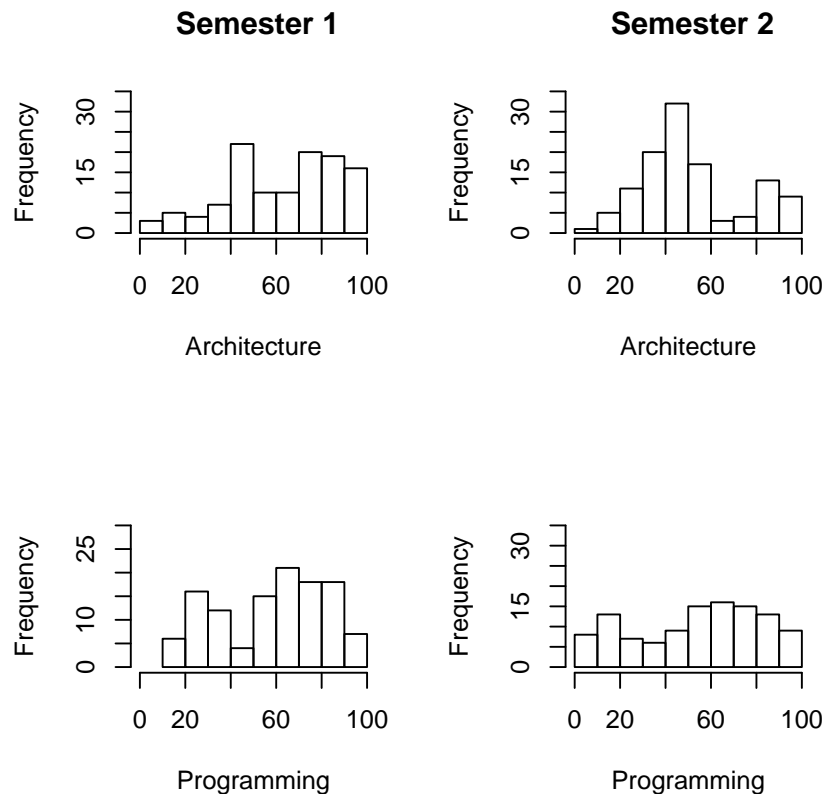
par (mfrow = c(2,2))
hist(arch1, xlab = "Architecture",
     main = " Semester 1", ylim = c(0, 35))

hist(arch2, xlab = "Architecture",
     main = " Semester 2", ylim = c(0, 35))

hist(prog1, xlab = "Programming",
     main = " ", ylim = c(0, 35))

hist(prog2, xlab = "Programming",
     main = " ", ylim = c(0, 35))

```



Stem and Leaf

```
downtime
```

```
[1]  0  1  2 12 12 14 18 21 21 23 24 25  
    28 29 30 30 30 33 36 44 45 47 51
```

```
stem(downtime, scale = 2)
```

```
0 | 012  
1 | 2248  
2 | 1134589  
3 | 00036  
4 | 457  
5 | 1
```

Notice:

Shape of the data based on the actual numbers observed.

Stem usually depict the 10s and the leaves depict the units.

Stem and Leaf

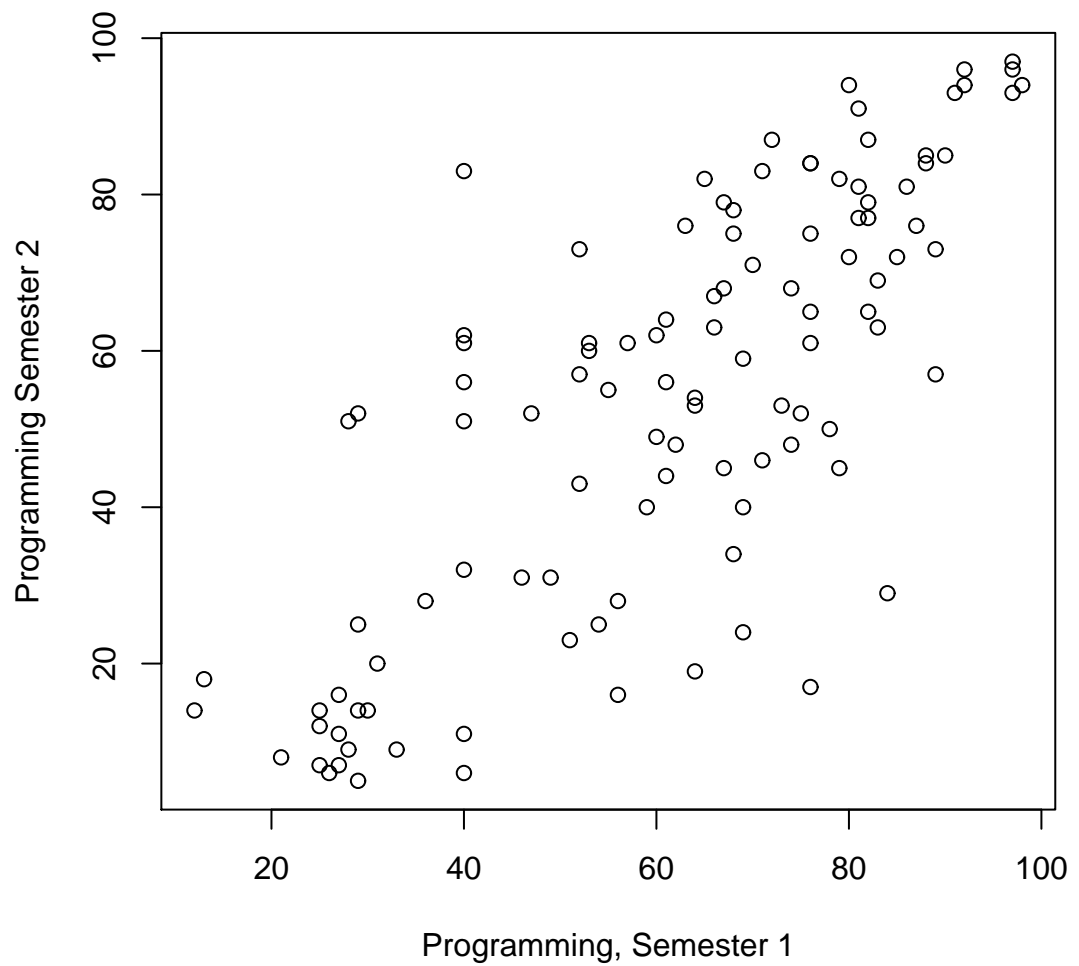
```
stem(prog1)
```

The decimal point is 1 digit(s) to the right

```
1 | 2344
1 | 59
2 | 11
2 | 5556777889999
3 | 0113
3 | 6
4 | 00000000
4 | 6779
5 | 12223344
5 | 56679
6 | 0011123444
6 | 566777888999
7 | 0112344
7 | 566666899
8 | 001112222334
8 | 5678899
9 | 0122
9 | 7778
```

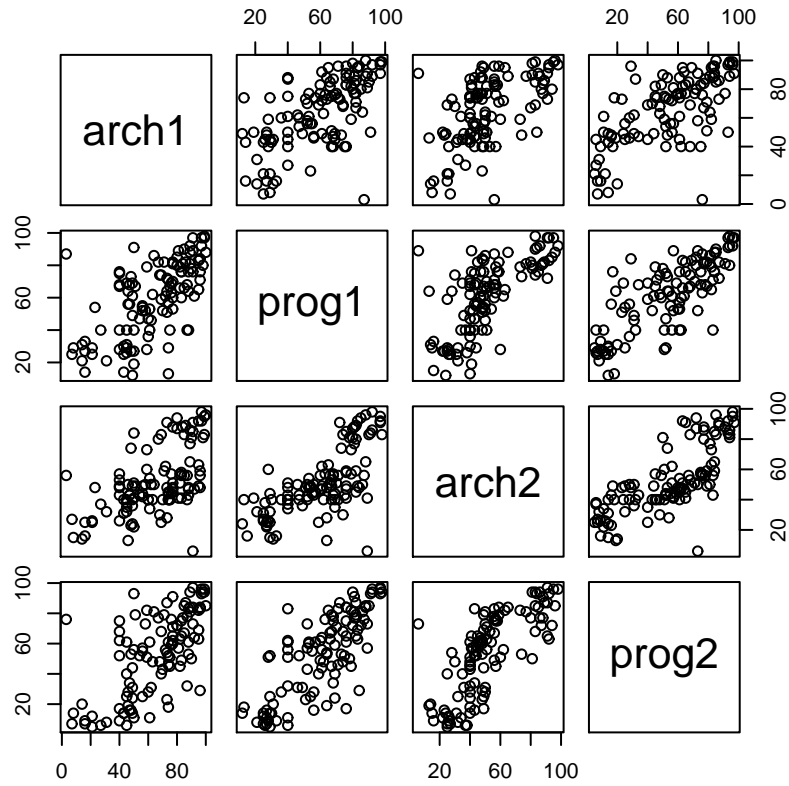
Scatter Plots

```
plot(prog1, prog2,  
      xlab = "Programming, Semester 1",  
      ylab = "Programming, Semester 2")
```



Pairs

```
pairs(results[2:5])
```



Graphical Display vs Summary Statistics

Data Set 1		Data Set 2		Data Set 3		Data Set 4	
x1	y1	x2	y2	x3	y3	x4	y4
10	8.04	10	9.14	10	7.46	8	6.58
8	6.95	8	8.14	8	6.77	8	5.76
13	7.58	13	8.74	13	12.74	8	7.71
9	8.81	9	8.77	9	7.11	8	8.84
11	8.33	11	9.26	11	7.81	8	8.47
14	9.96	14	8.10	14	8.84	8	7.04
6	7.24	6	6.13	6	6.08	8	5.25
4	4.26	4	3.10	4	5.39	19	12.50
12	10.84	12	9.13	12	8.15	8	5.56
7	4.82	7	7.26	7	6.42	8	7.91
5	5.68	5	4.74	5	5.73	8	6.89

First read the data into separate vectors:

```
x1<-c(10, 8, 13, 9, 11, 14, 6, 4, 12, 7, 5)
y1<-c(8.04, 6.95, 7.58, 8.81, 8.33, 9.96, 7.24, 4.26, 10.84, 4.82, 5.68)

x2 <- c(10, 8, 13, 9, 11, 14, 6, 4, 12, 7, 5)
y2 <-c(9.14, 8.14, 8.74, 8.77, 9.26, 8.10, 6.13, 3.10, 9.13, 7.26, 4.74)

x3<- c(10, 8, 13, 9, 11, 14, 6, 4, 12, 7, 5)
y3 <- c(7.46, 6.77, 12.74, 7.11, 7.81, 8.84, 6.08, 5.39, 8.15, 6.42, 5.73)

x4<- c(8, 8, 8, 8, 8, 8, 8, 19, 8, 8, 8)
y4 <- c(6.58, 5.76, 7.71, 8.84, 8.47, 7.04, 5.25, 12.50, 5.56, 7.91, 6.89)
```

Calculate the means

```
mean (data.frame(x1, x2, x3, x4))
x1 x2 x3 x4
 9  9  9  9

mean (data.frame(y1, y2, y3, y4))
      y1      y2      y3      y4
7.500909 7.500909 7.500000 7.500909
```

And the standard deviations:

```
sd(data.frame(x1, x2, x3, x4))
      x1      x2      x3      x4
3.316625 3.316625 3.316625 3.316625

sd(data.frame(y1, y2, y3, y4))
      y1      y2      y3      y4
2.031568 2.031657 2.030424 2.030579
```

Everything seems the same!

