

MAS1343: Computational Probability and Statistics (with R)

Dr Lee Fawcett

School of Mathematics & Statistics

Semester 2, 2014/2015

Part I

Introduction and Housekeeping

1.1.1 Lecturer information: who am I?

- The lecturer for this module is Dr Lee Fawcett.
- If you have any questions, comments or feedback on this course I can be contacted via email.
- More complicated questions about the course are best dealt with face-to-face.

Timetable

- See your lecture notes for a full week-by-week schedule, including hand-in deadlines
- Generally, we meet twice per week:

Timetable

- See your lecture notes for a full week-by-week schedule, including hand-in deadlines
- Generally, we meet twice per week:
 - Mondays 9–11 :-(in LT3
 - Wednesdays 9–10 :-(in the Herschel PC cluster
- Occasionally, we will also meet on Thursdays at 3 in LT2.

1.1.2 Office hours

The scheduled office hour for this module is Thursday 3–5.

I also have the following office hours for MAS2317, so you can drop in then too: Friday 1–2.

In general, I am happy to see students anytime (if I'm free), except on Wednesdays and Friday after 3pm.

1.1.3 Lectures

- Your calendar at the front of your notes booklet contains a full schedule for this course.
- Check your ncl e-mail regularly for further information
- This course has:
 - Lectures - fewer than other modules
 - Problem classes: in lecture time - fewer than other modules
 - Tutorials: in lecture time - fewer than other modules
 - Drop-in sessions
 - Computing practicals - loads more than other modules
 - Feedback sessions: in lecture time

1.1.3 Lectures

Lectures will also be recorded. You can access the recordings through BB or at the following webpage:

<http://www.mas.ncl.ac.uk/~nlf8/teaching/mas1343/recap/>

The lecture slides will (eventually) be put on BB.

1.1.3 Lectures

Lectures will also be recorded. You can access the recordings through BB or at the following webpage:

<http://www.mas.ncl.ac.uk/~nlf8/teaching/mas1343/recap/>

The lecture slides will (eventually) be put on BB.

1.1.4 Assessment

- End of semester examination worth 60% of your overall mark. In order to pass this module you must get at least 35% in the exam.
- Assessment of course-work worth 40%:
 - A **single** CBA - due week 3;
 - Practicals;
 - Demonstrations of computer skills at the lab.

Practical Weightings

Not all practicals are worth the same marks:

Practical	1	2	3	4	5	6
Weight(%)	5	4	9	9	5	5

Table: Approximate coursework weightings. On average, 1% of coursework will take 1 hour (assuming you are familiar with your lecture notes). CBA 1 is worth 3% of your final mark.

1.1.5 Practicals

During each computer practical, your general computer skills will be assessed. In particular

- Are your files organised in a sensible manner? For example, do you have a directory for MAS1343 and sub-directories for each practical?
- Are your files named in a structured manner?
- Do you use the correct text editor?

To obtain these marks, you must attend at least five computer labs. These marks will form part of your final assessment.

1.1.5 Practicals

All computer practicals are held in the Herschel cluster.

1.1.5 Practicals

All computer practicals are held in the Herschel cluster. A list of other free computers in the University can be found:

<http://www.ncl.ac.uk/iss/teaching/clusters/usage.php>

1.1.5 Practicals

All computer practicals are held in the Herschel cluster. A list of other free computers in the University can be found:

`http://www.ncl.ac.uk/iss/teaching/clusters/usage.php`

and the mobile site is:

`https://m.ncl.ac.uk/iss/`

If you use a machine in a cluster and it is missing the necessary software, contact Dr Chris Graham.

1.2 What is **R**?

- **R** is a computer package that is widely used.
- For example Google, IBM, Shell and spread betting companies use **R** for statistical software development and data analysis.

1.2 What is **R**?

- **R** is a computer package that is widely used.
- For example Google, IBM, Shell and spread betting companies use **R** for statistical software development and data analysis.
- See the NY times article at <http://tinyurl.com/9m5qyh>.
- **R** uses a command line interface, though several graphical user interfaces are available.

1.2 What is **R**?

- **R** is a computer package that is widely used.
- For example Google, IBM, Shell and spread betting companies use **R** for statistical software development and data analysis.
- See the NY times article at <http://tinyurl.com/9m5qyh>.
- **R** uses a command line interface, though several graphical user interfaces are available.
- Why use a command line interface as opposed to Excel, say?

- **R** is highly extensible through the use of user-submitted libraries for specific functions or specific areas of study.
- A particular strength of **R** is its graphical facilities, which produce quality graphs that can include mathematical symbols.

- **R** is highly extensible through the use of user-submitted libraries for specific functions or specific areas of study.
- A particular strength of **R** is it's graphical facilities, which produce quality graphs that can include mathematical symbols.
- In this class and MAS1343 we use **R**. Since you do so much **R** programming in this class, the **R** in MAS1343 will seem fairly easy.
- See BB for a list of recommended books

1.2.2 Previous knowledge

- This course is intended to teach you the basics of programming.
- **No** previous programming knowledge is assumed.
- Overall, the programming related aspects probably accounts for 50% of the final mark in this course - there are programming questions in the exam!
- It is crucial that you come to all the practicals for this course.

Course Summary

- The mathematics/statistics in this course is straightforward. Much easier than in other modules
- Each individual R command is also easy to understand

Course Summary

- The mathematics/statistics in this course is straightforward. Much easier than in other modules
- Each individual R command is also easy to understand
- **But** this module is usually rated as one of the most difficult. I will ask you to apply the knowledge you learn, and problem–solve.

1.3 Movie data set

- The internet movie database (IMDB), is a website devoted to collecting movie data supplied by studios and fans.
- It claims to be the biggest movie database on the web and is run by amazon. More about information imdb.com can be found online.
- Movies were selected for inclusion if they:
 - had a known length;
 - had been rated by at least one imdb user;
 - had an mpaa rating.

IMDB data set

- Title. Title of the movie.

IMDB data set

- Title. Title of the movie.
- Year. Year of release.

IMDB data set

- Title. Title of the movie.
- Year. Year of release.
- Budget. Total budget in US dollars. If the budget isn't known, then it is stored as '-1'.

IMDB data set

- Title. Title of the movie.
- Year. Year of release.
- Budget. Total budget in US dollars. If the budget isn't known, then it is stored as '-1'.
- Length. Length in minutes.

IMDB data set

- Title. Title of the movie.
- Year. Year of release.
- Budget. Total budget in US dollars. If the budget isn't known, then it is stored as '-1'.
- Length. Length in minutes.
- Rating. Average IMDB user rating.

IMDB data set

- Title. Title of the movie.
- Year. Year of release.
- Budget. Total budget in US dollars. If the budget isn't known, then it is stored as '-1'.
- Length. Length in minutes.
- Rating. Average IMDB user rating.
- Votes. Number of IMDB users who rated this movie.

IMDB data set

- Title. Title of the movie.
- Year. Year of release.
- Budget. Total budget in US dollars. If the budget isn't known, then it is stored as '-1'.
- Length. Length in minutes.
- Rating. Average IMDB user rating.
- Votes. Number of IMDB users who rated this movie.
- r1-10. Percentage (to nearest 5%) of users who rated this movie a 1,..., 10.

A big data set

- mpaa. MPAA rating.

A big data set

- mpaa. MPAA rating.
- Action, Animation,... Binary variables representing if movie was classified as belonging to that genre. A movie can belong to more one genre. See for example the film *Ab blaze* in Table 1.1.

A big data set

- mpaa. MPAA rating.
- Action, Animation,... Binary variables representing if movie was classified as belonging to that genre. A movie can belong to more one genre. See for example the film *Ab blaze* in Table 1.1.
- There are a total of 24 variables and 4847 films.
- This is **only a subset** of the data, the actual data set contains information on over 50,000 movies.

Title	Year	Length	Budget	Voting statistics					Movie genre							
				Rating	Votes	r1	...	r10	mpaa	Action	Animation	Comedy	Drama	Documentary	Romance	Short
A.k.a. Cassius	1970	85	-1	5.7	43	4.5	...	14.5	PG	0	0	0	0	1	0	0
AKA	2002	123	-1	6.0	335	24.5	...	1.5	R	0	0	0	1	0	0	0
Alien Vs. Pred	2004	102	45000000	5.4	14651	4.5	...	4.5	PG-13	1	0	0	0	0	0	0
Abandon	2002	99	25000000	4.7	2364	4.5	...	4.5	PG-13	0	0	0	1	0	0	0
Abendland	1999	146	-1	5.0	46	14.5	...	24.5	R	0	0	0	0	0	0	0
Aberration	1997	93	-1	4.8	149	14.5	...	4.5	R	0	0	0	0	0	0	0
Abilene	1999	104	-1	4.9	42	0.0	...	24.5	PG	0	0	0	1	0	0	0
Ablaze	2001	97	-1	3.6	98	24.5	...	14.5	R	1	0	0	1	0	0	0
Abominable Dr	1971	94	-1	6.7	1547	4.5	...	14.5	PG-13	0	0	0	0	0	0	0
About Adam	2000	105	-1	6.4	1303	4.5	...	4.5	R	0	0	1	0	0	1	0

1.4 Module R package

This module has an associated R package. Installing this package is straightforward:

```
> install.packages("mas1343",  
+                 repos="http://R-Forge.R-project.org",  
+                 type="source")
```

To load the package, use

```
> library(mas1343)
```

Part II

Introduction to R

2.1 A simple R session

At its most basic R, can be used as a calculator, for example multiplication and subtraction...

```
> 5*5  
[1] 25  
> 10.2/6  
[1] 1.7
```

...and more 'advanced' operations

```
> 2^3  
[1] 8
```

...and more 'advanced' operations

```
> 2^3  
[1] 8
```

```
> exp(1.5)  
[1] 4.481689
```

```
> log(10)  
[1] 2.302585
```

...and more 'advanced' operations

```
> 2^3  
[1] 8
```

```
> exp(1.5)  
[1] 4.481689
```

```
> log(10)  
[1] 2.302585
```

```
> 4 %% 3  
[1] 1
```

2.1.1 Assignment operations

- In the practicals we will use assignment, i.e. $x = 5$.
- However, computer assignment is different from typical mathematical assignment.
- For example,

```
> x = 5  
> x = x + 1  
> x  
[1] 6
```

- Notice that when we type `x = 5`, R doesn't display or print any output to the screen.
- If we want to see what value has been assigned to the variable, we type `x`.
- Another equivalent way is to surround the expression with brackets.
- For example

```
> (x = 2*x)  
[1] 12
```

- You can also use the `<-` operator for assignment. This is, for almost all situations, identical to the `=` operator.

Question 1

What is the value of x?

```
> x = 5  
> x = 2*x + 1  
> x
```

- 1 5
- 2 11
- 3 12

Question 1

What is the value of x?

```
> x = 5  
> x = 2*x + 1  
> x  
[1] 11
```

- 1 5
- 2 11
- 3 12

Question 2

What is the value of x?

```
> x = 5  
> y= 6  
> x = x + 2*y  
> x + 5  
> x
```

- 1 5
- 2 10
- 3 17
- 4 22

Question 2

What is the value of x?

```
> x = 5  
> y= 6  
> x = x + 2*y  
> x + 5  
[1] 22  
> x  
[1] 17
```

- 1 5
- 2 10
- 3 17
- 4 22

Question 3

What is the value of x?

```
> x = 5
```

```
> y = 6
```

```
> x + 5
```

```
> x
```

- 1 5
- 2 10
- 3 17
- 4 22

Question 3

What is the value of x?

```
> x = 5  
> y = 6  
> x + 5  
[1] 10  
> x  
[1] 5
```

- 1 5
- 2 10
- 3 17
- 4 22

2.1.2 Data types

R has a variety of data types:

```
> (v = TRUE)
[1] TRUE
```

2.1.2 Data types

R has a variety of data types:

```
> (v = TRUE)
[1] TRUE
```

```
> (w = "fred")
[1] "fred"
```

2.1.2 Data types

R has a variety of data types:

```
> (v = TRUE)
[1] TRUE
```

```
> (w = "fred")
[1] "fred"
```

```
> (x = 5.0)
[1] 5
```

Other some “special” data types:

```
> (y = 5/0)
[1] Inf
> (z = y-y)
[1] NaN
```

Type	Example 1	Example 2	Example 3	Example 4
Doubles	2	3.1242	-45.6	4e-10
Logicals/Boolean	TRUE	FALSE		
Characters	"FRED"	"x"	"Male"	"TRUE"
Infinity	Inf	5/0		

Table: Summary of data types in R.

Another important data type in R is NA. This is used to represent missing values. A list of data tables is given in Table 1.1.

Question 4

What type of variable is x?

```
> x = "1"
```

- 1 Character
- 2 Double
- 3 Logical

Question 4

What type of variable is x?

```
> x = "1"
```

- 1 Character
- 2 Double
- 3 Logical

2.2 The R workspace

Once you create a variable, R stores that variable in memory for reuse. You can view available variables with the `ls()` command:

```
> ls()  
[1] "movies"
```

To delete a variable in R, we use the `rm` function. For example,

```
> x = 0  
> y = 1  
> z = 2  
> rm(x, y, z)  
[1] "movies" "x"      "y"      "z"
```

To delete a variable in R, we use the `rm` function. For example,

```
> x = 0  
> y = 1  
> z = 2  
> ls()  
[1] "movies" "x"      "y"      "z"
```

```
> rm(x)
```

To delete a variable in R, we use the `rm` function. For example,

```
> x = 0  
> y = 1  
> z = 2  
> ls()  
[1] "movies" "x"      "y"      "z"
```

```
> rm(x)
```

```
> ls()  
[1] "movies" "y"      "z"
```

The nuclear option

We can remove everything in the workspace using:

```
> rm(list=ls())
```

2.3 Vectors

- Vectors are the most basic of all data structures, but are used in almost all R code.
- A R vector contains n values of the same type, where n can be zero.
- For example

```
> c(0, 1, 2, 3, 4, 5)
[1] 0 1 2 3 4 5
> (my_first_vec = c(0, 1, 2, 3, 4, 5))
[1] 0 1 2 3 4 5
> (my_second_vec = c("Male", "Female", "Male"))
[1] "Male"    "Female"  "Male"
```

In R, when we type:

```
> x = 5  
> y = "Fred"
```

We have actually created a vector of doubles and characters (of length one).

There are special functions in R to determine type of a variable:

```
> x = 5
```

There are special functions in R to determine type of a variable:

```
> x = 5
```

```
> is.double(x)  
[1] TRUE
```

There are special functions in R to determine type of a variable:

```
> x = 5
```

```
> is.double(x)  
[1] TRUE
```

```
> is.character(x)  
[1] FALSE
```

There are special functions in R to determine type of a variable:

```
> x = 5
```

```
> is.double(x)  
[1] TRUE
```

```
> is.character(x)  
[1] FALSE
```

```
> is.vector(x)  
[1] TRUE
```

To determine the length of vector in R, we use the `length` function:

```
> length(my_first_vec)
[1] 6
> length(my_second_vec)
[1] 3
```

To create sequences of numbers we use the seq command. For example,

```
> (x1 = seq(1, 6))  
[1] 1 2 3 4 5 6
```

To create sequences of numbers we use the `seq` command. For example,

```
> (x1 = seq(1, 6))  
[1] 1 2 3 4 5 6
```

```
> (x2 = seq(-4, 4, by=2))  
[1] -4 -2  0  2  4
```

Command description	Example	Result
Length	<code>length(x)</code>	4
Reverse order	<code>rev(x)</code>	3,5,5,1
Sort	<code>sort(x)</code>	1,3,5,5
Sum	<code>sum(x)</code>	14
Extract unique elements	<code>unique(x)</code>	1,5,3
Indices of particular elements	<code>which(x==5)</code>	2,3

Table: Useful vector functions. In the above examples, `x = c(1,5,5,3)`. Check the associated R help for further information.

Question 5

What is the value of x?

```
> x = seq(2, 6)
```

- 1 2,3,4,5,6
- 2 2,6
- 3 6,4,2

Question 5

What is the value of x?

```
> x = seq(2, 6)
```

```
> x
```

```
[1] 2 3 4 5 6
```

1 2,3,4,5,6

2 2,6

3 6,4,2

Question 6

What is the value of blob?

```
> blob = seq(2, -2, by=-2)
```

- -2 -1 0 1 2
- -2 0 2
- 2 0 -2
- 2 1 0 -1 -2

Question 6

What is the value of blob?

```
> (blob = seq(2, -2, by=-2))  
[1] 2 0 -2
```

- -2 -1 0 1 2
- -2 0 2
- 2 0 -2
- 2 1 0 -1 -2

2.3.1 Vector operations

- When our data is in a vector structure, we can apply standard operations to the entire vector.
- For example,

```
> (x = seq(-4, 4))  
[1] -4 -3 -2 -1  0  1  2  3  4
```

2.3.1 Vector operations

- When our data is in a vector structure, we can apply standard operations to the entire vector.
- For example,

```
> (x = seq(-4, 4))  
[1] -4 -3 -2 -1 0 1 2 3 4
```

```
> x*x  
[1] 16 9 4 1 0 1 4 9 16
```

```
> x - 5
```

```
[1] -9 -8 -7 -6 -5 -4 -3 -2 -1
```

```
> x - 5
```

```
[1] -9 -8 -7 -6 -5 -4 -3 -2 -1
```

```
> x + x
```

```
[1] -8 -6 -4 -2  0  2  4  6  8
```

Question 7

What is the value of z?

```
> x = c(1, 10, 100, 1000)
> y = c(1, 5, 10, 100)
> z = x/y
```

- 1 1, 10, 100, 1000
- 2 1, 5, 10, 1000
- 3 1, 1, 1, 1
- 4 1, 2, 10, 10

Question 7

What is the value of z?

```
> x = c(1, 10, 100, 1000)
> y = c(1, 5, 10, 100)
> (z = x/y)
[1] 1 2 10 10
```

- 1 1, 10, 100, 1000
- 2 1, 5, 10, 1000
- 3 1, 1, 1, 1
- 4 1, 2, 10, 10

Question 8

What is the value of y?

```
> x = seq(1, 5)  
> y = length(x + 1)
```

- 1 1
- 2 5
- 3 10
- 4 12

Question 8

What is the value of y?

```
> x = seq(1, 5)
> (y = length(x + 1))
[1] 5
```

- 1 1
- 2 5
- 3 10
- 4 12

2.3.2 Extracting elements from vectors

- R has a number of useful of methods that we use to extract subsets of our data.
- For example to pick out particular elements:

```
> my_first_vec[2]
[1] 1
> my_second_vec[2:3]
[1] "Female" "Male"
> my_first_vec[4:2]
[1] 3 2 1
```

- We can also use other arguments.
- For example to remove the last entry in the vector, we use the `length` function:

```
> l = length(my_first_vec)
> my_first_vec[1:(l-1)]
[1] 0 1 2 3 4
```

- We determine the length of the vector using the `length` function and select particular elements using the `[·]` operator.

Question 9

What is the value of x?

```
> y = seq(2, 6)  
> x = y[3:length(y)]
```

- 1 2,3,4,5,6
- 2 3,4,5,6
- 3 4,5,6
- 4 3

Question 9

What is the value of x?

```
> y = seq(2, 6)
> (x = y[3:length(y)])
[1] 4 5 6
```

- 1 2,3,4,5,6
- 2 3,4,5,6
- 3 4,5,6
- 4 3

2.4 Logical vectors

Boolean	A	B	$\bar{A}$	$\bar{B}$	$A \cap B$	$A \cup B$
R	A	B	!A	!B	A & B	A B

Table: Truth table for Boolean operation

2.4 Logical vectors

Boolean	A	B	$\bar{A}$	$\bar{B}$	$A \cap B$	$A \cup B$
R	A	B	!A	!B	A & B	A B
	TRUE	TRUE	FALSE	FALSE		

Table: Truth table for Boolean operation

2.4 Logical vectors

Boolean	A	B	$\bar{A}$	$\bar{B}$	$A \cap B$	$A \cup B$
R	A	B	!A	!B	A & B	A B
	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE

Table: Truth table for Boolean operation

2.4 Logical vectors

Boolean R	A A	B B	$\bar{A}$!A	$\bar{B}$!B	$A \cap B$ A & B	$A \cup B$ A B
	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE
	TRUE	FALSE	FALSE	TRUE		

Table: Truth table for Boolean operation

2.4 Logical vectors

Boolean R	A A	B B	$\bar{A}$!A	$\bar{B}$!B	$A \cap B$ A & B	$A \cup B$ A B
	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE
	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE

Table: Truth table for Boolean operation

2.4 Logical vectors

Boolean R	A A	B B	$\bar{A}$!A	$\bar{B}$!B	$A \cap B$ A & B	$A \cup B$ A B
	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE
	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE
	FALSE	TRUE	TRUE	FALSE		

Table: Truth table for Boolean operation

2.4 Logical vectors

Boolean R	A A	B B	$\bar{A}$!A	$\bar{B}$!B	$A \cap B$ A & B	$A \cup B$ A B
	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE
	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE
	FALSE	TRUE	TRUE	FALSE	FALSE	TRUE

Table: Truth table for Boolean operation

2.4 Logical vectors

Boolean R	A A	B B	$\bar{A}$!A	$\bar{B}$!B	$A \cap B$ A & B	$A \cup B$ A B
	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE
	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE
	FALSE	TRUE	TRUE	FALSE	FALSE	TRUE
	FALSE	FALSE	TRUE	TRUE		

Table: Truth table for Boolean operation

2.4 Logical vectors

Boolean R	A A	B B	$\bar{A}$!A	$\bar{B}$!B	$A \cap B$ A & B	$A \cup B$ A B
	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE
	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE
	FALSE	TRUE	TRUE	FALSE	FALSE	TRUE
	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE

Table: Truth table for Boolean operation

Question 10

What is the value of z?

```
> x = TRUE  
> y = FALSE  
> z = x & !y
```

- ① TRUE
- ② FALSE

Question 10

What is the value of z?

```
> x = TRUE  
> y = FALSE  
> (z = x & !y)  
[1] TRUE
```

- 1 TRUE
- 2 FALSE

Question 11

What is the value of z?

```
> x = TRUE  
> y = FALSE  
> z = 5  
> x = (y | !y) & x  
> z = x
```

- 1 TRUE
- 2 FALSE

Question 11

What is the value of z?

```
> x = TRUE
> y = FALSE
> z = 5
> x = (y | !y) & x
> (z = x)
[1] TRUE
```

- 1 TRUE
- 2 FALSE

2.4.1 Using logicals for sub-setting vectors

We can construct vectors of logical operators and use them to take subsets of vectors. For example,

```
> (logic1 = c(TRUE, FALSE, TRUE, FALSE))  
[1] TRUE FALSE TRUE FALSE  
> (vec1 = seq(1, 4))  
[1] 1 2 3 4  
> vec1[logic1]  
[1] 1 3
```

2.4.2 Relational operators

When programming it is often necessary to test relations for equality and inequality. To do this in R we use the relation operators. First let's define some variables:

```
> x = 5  
> y = 7
```

To test for equality we use ==:

```
> x == 5  
[1] TRUE  
  
> x == y  
[1] FALSE
```

Similarly, to test for inequality we use !=:

```
> x != 5  
[1] FALSE  
  
> y != x  
[1] TRUE
```

There are also commands for greater/less than:

```
> y > 6  
[1] TRUE  
> x >= 5  
[1] TRUE  
> x <= y  
[1] TRUE
```

We can also apply these techniques to vectors. For example:

```
> (vec2 = seq(0, 10, by=2.5))  
[1] 0.0 2.5 5.0 7.5 10.0  
> vec2 > 3  
[1] FALSE FALSE TRUE TRUE TRUE  
> vec2 < 9  
[1] TRUE TRUE TRUE TRUE FALSE  
> (vec2 > 3) & (vec2 < 9)  
[1] FALSE FALSE TRUE TRUE FALSE
```

We can also combine logical operators:

```
> vec2 > 3  
[1] FALSE FALSE  TRUE  TRUE  TRUE  
> !(vec2 > 3)  
[1]  TRUE  TRUE FALSE FALSE FALSE
```

2.4.2 Relational operators

Operator	Tests for	Example	Result
==	Equality	x == 5	TRUE
!=	Inequality	x != 5	FALSE
<	Less than	x < 5	FALSE
<=	Less or equal	x <= 5	TRUE
>	Greater	x > 5	FALSE
>=	Greater or equal	x >= 5	TRUE

Table: Summary of R relational operators. The example is for $x = 5$.

2.4.3 Vector partitions

We can construct vectors of logical operators and use them to take subsets of vectors. For example,

```
> (logic1 = c(TRUE, FALSE, TRUE, FALSE))  
[1] TRUE FALSE TRUE FALSE  
> (vec1 = seq(1, 4))  
[1] 1 2 3 4  
> vec1[logic1]  
[1] 1 3
```

```
> vec2
[1] 0.0 2.5 5.0 7.5 10.0
> vec2 > 3 & vec2 < 9
[1] FALSE FALSE TRUE TRUE FALSE
> vec2[vec2 > 3 & vec2 < 9]
[1] 5.0 7.5
```

```
> library(mas1343)
> data(Budget)
> data(Length)
```

Using relational operators allows us to extract subsets of data very easily.
Consider the movie budgets:

```
> length(Budget)
[1] 4847
```

Using relational operators allows us to extract subsets of data very easily.
Consider the movie budgets:

```
> length(Budget)
[1] 4847
```

To select movies where the budget is known, we use the following command...

```
> non_zero_b = Budget[Budget != -1]
> length(non_zero_b)
[1] 1785
```

... and to select movies where the movie length is greater than 60 mins but shorter than 90 mins

```
> m_l = Length[Length > 60 & Length < 90]
```

and to select movies where the movie length is greater than 60 mins but shorter than 90 mins:

```
> m_l = Length[Length > 60 & Length < 90]
```

Question 12

What is the value of z?

```
> x = seq(-2, 10)  
> z = x[(x >= 7) ]
```

- TRUE
- 7,8,9,10
- -2, -1, 7,8,9,10
- 0,1,2,3,4,5,6

Question 12

What is the value of z?

```
> x = seq(-2, 10)
> (z = x[(x >= 7) ])
[1] 7 8 9 10
```

- TRUE
- 7,8,9,10
- -2, -1, 7,8,9,10
- 0,1,2,3,4,5,6

Question 13

What is the value of z?

```
> x = seq(-2, 10)
> z = x[(x >= 7) | x < 0]
```

- TRUE
- 7,8,9,10
- -2, -1, 7,8,9,10
- 0,1,2,3,4,5,6

Question 13

What is the value of z?

```
> x = seq(-2, 10)
> (z = x[(x >= 7) | x < 0])
[1] -2 -1 7 8 9 10
```

- TRUE
- 7,8,9,10
- -2, -1, 7,8,9,10
- 0,1,2,3,4,5,6

2.5 Data frames

- A data frame is a special kind of object.
- We use data frames for storing and managing data sets that have a rectangular structure.
- Typically the rows correspond to *cases* and the columns to *variables*.
- The crucial difference between a data frame and a `matrix` is that all values in a matrix must be of the same type.
- The next code segment constructs a simple data frame.

2.5 Data frames

First, we construct three vectors:

```
> age = c(24, 26, 25, 21)
> sex = c("Male", "Female", "Male", "Female")
> respond = c(TRUE, FALSE, FALSE, FALSE)
```

Then we combine them using the `data.frame` function:

2.5 Data frames

First, we construct three vectors:

```
> age = c(24, 26, 25, 21)
> sex = c("Male", "Female", "Male", "Female")
> respond = c(TRUE, FALSE, FALSE, FALSE)
```

Then we combine them using the `data.frame` function:

```
> (df1 = data.frame(age=age, gender=sex,
+   respond=respond))
  age gender respond
1  24   Male    TRUE
2  26 Female   FALSE
3  25   Male   FALSE
4  21 Female   FALSE
```

2.5 Data frames

- The data frame `df1` has three columns and four rows.
- Once we put our data into a data frame, then data manipulation is easier.
- To calculate the dimensions of a data frame we use `dim`:

```
> dim(df1)
[1] 4 3
```

2.5 Data frames

To extract the first column we use square brackets:

```
> df1[,1]  
[1] 24 26 25 21
```

Similarly, we can get the first row

2.5 Data frames

To extract the first column we use square brackets:

```
> df1[,1]
[1] 24 26 25 21
```

Similarly, we can get the first row

```
> df1[1, ]
  age gender respond
1  24   Male     TRUE
```

2.5 Data frames

The column names are also easily manipulated

```
> colnames(df1)
[1] "age"      "gender"   "respond"
```

2.5 Data frames

The column names are also easily manipulated

```
> colnames(df1)
[1] "age"      "gender"   "respond"
```

```
> (colnames(df1) = c("Age", "Sex", "Respond"))
[1] "Age"      "Sex"      "Respond"
```

When we download the movies data set, we automatically create a data frame:

```
> dim(movies)
[1] 4847  24
```

When we download the movies data set, we automatically create a data frame:

```
> dim(movies)
[1] 4847  24
```

```
> colnames(movies)[1:4]
[1] "Title"  "Year"   "Length" "Budget"
```

When we download the movies data set, we automatically create a data frame:

```
> dim(movies)
[1] 4847  24
```

```
> colnames(movies)[1:4]
[1] "Title"  "Year"   "Length" "Budget"
```

```
> movies[4061, 1:4]
```

	Title	Year	Length	Budget
4061	Star Wars: Episode I - The Phantom Menace	1999	133	1.15e+08

2.5.1 Subsets of data frames

We can also retrieve subsets from the data frame. For example, if we wanted only female responses, then...

```
> (female_only = df1$Sex=="Female")  
[1] FALSE  TRUE FALSE  TRUE  
> (df2 = df1[female_only, ])  
  Age  Sex Respond  
2  26 Female  FALSE  
4  21 Female  FALSE
```

2.5.1 Subsets of data frames

... or people 25 and over,

```
> over_25 = df1$Age>=25  
> (df3 = df1[over_25, ])  
  Age  Sex Respond  
2  26 Female  FALSE  
3  25  Male  FALSE
```

2.5.2 Example: movie data

We can select movies where the budget is greater than \$100,000

```
> m1 = movies[movies$Budget > 100000,]  
> dim(m1)  
[1] 1738 24
```

2.5.2 Example: movie data

We can select movies where the budget is greater than \$100,000

```
> m1 = movies[movies$Budget > 100000,]  
> dim(m1)  
[1] 1738 24
```

Or movies that cost more than \$100,000 but are not R rated:

```
> m2 = movies[movies$Budget > 100000 & movies$mpaa !=  
"R",]  
> dim(m2)  
[1] 727 24
```

Or movies that are either PG or PG-13:

```
> m3 = movies[  
+   movies$mpaa == "PG" | movies$mpaa == "PG-13",]  
> dim(m3)  
[1] 1515   24
```

Question 14

What type of variable is m1?

```
> m1 = movies[movies$Budget > 10000,]
```

- A vector of doubles
- A vector of String
- A vector of Logical
- A Data frame

Question 14

What type of variable is m1?

```
> m1 = movies[movies$Budget > 10000,]  
> is.data.frame(m1)  
[1] TRUE
```

- A vector of doubles
- A vector of String
- A vector of Logicals
- A Data frame

Question 15

What type of variable is m2?

```
> m2 = movies$Budget > 10000
```

- A vector of doubles
- A vector of Strings
- A vector of Logicals
- A Data frame

Question 15

What type of variable is m2?

```
> m2 = movies$Budget > 10000  
> is.vector(m2)  
[1] TRUE  
> is.logical(m2)  
[1] TRUE
```

- A vector of doubles
- A vector of String
- **A vector of Logicals**
- A Data frame

Part III

Summary Statistics

3.1.1 Sample mean

One of the most important and widely used measures of location is the mean:

$$\begin{aligned}\bar{x} &= \frac{x_1 + x_2 + \dots + x_n}{n} \\ &= \frac{1}{n} \sum_{i=1}^n x_i .\end{aligned}$$

3.1.1 Sample mean

One of the most important and widely used measures of location is the mean:

$$\begin{aligned}\bar{x} &= \frac{x_1 + x_2 + \dots + x_n}{n} \\ &= \frac{1}{n} \sum_{i=1}^n x_i .\end{aligned}$$

So if our data set was $\{0, 3, 2, 0\}$, then $n = 4$.

3.1.1 Sample mean

One of the most important and widely used measures of location is the mean:

$$\begin{aligned}\bar{x} &= \frac{x_1 + x_2 + \dots + x_n}{n} \\ &= \frac{1}{n} \sum_{i=1}^n x_i .\end{aligned}$$

So if our data set was $\{0, 3, 2, 0\}$, then $n = 4$. Hence,

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i =$$

3.1.1 Sample mean

One of the most important and widely used measures of location is the mean:

$$\begin{aligned}\bar{x} &= \frac{x_1 + x_2 + \dots + x_n}{n} \\ &= \frac{1}{n} \sum_{i=1}^n x_i .\end{aligned}$$

So if our data set was $\{0, 3, 2, 0\}$, then $n = 4$. Hence,

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i = \frac{0 + 3 + 2 + 0}{4} =$$

3.1.1 Sample mean

One of the most important and widely used measures of location is the mean:

$$\begin{aligned}\bar{x} &= \frac{x_1 + x_2 + \dots + x_n}{n} \\ &= \frac{1}{n} \sum_{i=1}^n x_i .\end{aligned}$$

So if our data set was $\{0, 3, 2, 0\}$, then $n = 4$. Hence,

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i = \frac{0 + 3 + 2 + 0}{4} = 1.25.$$

3.1.2 Sample median

- The sample median is the middle observation when the data is ranked in increasing order.
- Denote the ranked observations as $x_{(1)}, x_{(2)}, \dots, x_{(n)}$.

3.1.2 Sample median

- The sample median is the middle observation when the data is ranked in increasing order.
- Denote the ranked observations as $x_{(1)}, x_{(2)}, \dots, x_{(n)}$.
- The sample median is defined as:

$$\text{Sample median} = \begin{cases} x_{(n+1)/2}, & n \text{ odd;} \\ \frac{1}{2}x_{(n/2)} + \frac{1}{2}x_{(n/2+1)}, & n \text{ even.} \end{cases}$$

3.1.2 Sample median

- The sample median is the middle observation when the data is ranked in increasing order.
- Denote the ranked observations as $x_{(1)}, x_{(2)}, \dots, x_{(n)}$.
- The sample median is defined as:

$$\text{Sample median} = \begin{cases} x_{(n+1)/2}, & n \text{ odd;} \\ \frac{1}{2}x_{(n/2)} + \frac{1}{2}x_{(n/2+1)}, & n \text{ even.} \end{cases}$$

- The median is more robust than the sample mean, but has less useful mathematical properties.

3.1.2 Sample median

- For our simple data set $\{0, 3, 2, 0\}$, to calculate the median we re-order it to: $\{0, 0, 2, 3\}$.
- Then take the average of the middle two observations to get 1.

3.1.3 Sample mode

- The mode is the value which occurs with the greatest frequency.
- Only makes sense to calculate or use it with discrete data.

3.1.4 Examples

Calculate the mean and median of $\{1, 7, 13, 14, 20\}$.

Solution

3.1.4 Examples

Calculate the mean and median of $\{1, 7, 13, 14, 20\}$.

Solution

The mean is:

$$\bar{x} = \frac{1}{5}(1 + 7 + 13 + 14 + 20) = 11.$$

3.1.4 Examples

Calculate the mean and median of $\{1, 7, 13, 14, 20\}$.

Solution

The mean is:

$$\bar{x} = \frac{1}{5}(1 + 7 + 13 + 14 + 20) = 11.$$

The median is the middle value, so we get 13.

3.1.4 Examples

Calculate the mean and median of $\{7, 13, 14, 20\}$.

Solution

3.1.4 Examples

Calculate the mean and median of $\{7, 13, 14, 20\}$.

Solution

The mean is:

$$\bar{x} = \frac{1}{4}(7 + 13 + 14 + 20) = 13.5$$

3.1.4 Examples

Calculate the mean and median of $\{7, 13, 14, 20\}$.

Solution

The mean is:

$$\bar{x} = \frac{1}{4}(7 + 13 + 14 + 20) = 13.5$$

The median is the middle value, so we get 13.5.

3.1.4 Examples

Calculate the mean and median of $\{1, 7, 13, 14, 200\}$.

3.1.4 Examples

Calculate the mean and median of $\{1, 7, 13, 14, 200\}$.

Solution

3.1.4 Examples

Calculate the mean and median of $\{1, 7, 13, 14, 200\}$.

Solution

The mean is:

$$\bar{x} = \frac{1}{5}(1 + 7 + 13 + 14 + 200) = 47$$

3.1.4 Examples

Calculate the mean and median of $\{1, 7, 13, 14, 200\}$.

Solution

The mean is:

$$\bar{x} = \frac{1}{5}(1 + 7 + 13 + 14 + 200) = 47$$

The median is the middle value, so we get 13.

3.2 Measures of spread

As well as knowing the location statistics of a data set, we also need to know how variable or 'spread-out' our data are.

3.2.1 Range

The range is easy to calculate.

It is simply the largest minus the smallest:

3.2.1 Range

The range is easy to calculate.

It is simply the largest minus the smallest:

$$\text{Range} = x_{(n)} - x_{(1)} .$$

3.2.1 Range

The range is easy to calculate.

It is simply the largest minus the smallest:

$$\text{Range} = x_{(n)} - x_{(1)} .$$

- So for our data set of $\{0, 3, 2, 0\}$, the range is $3 - 0 = 3$.
- It is very useful for data checking purposes, but in general it's not very robust.

3.2.2 Sample variance and standard deviation

The sample variance, s^2 , is defined as

$$s^2 =$$

$$=$$

The second formula is easier for calculations.

3.2.2 Sample variance and standard deviation

The sample variance, s^2 , is defined as

$$s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$$
$$=$$

The second formula is easier for calculations.

3.2.2 Sample variance and standard deviation

The sample variance, s^2 , is defined as

$$\begin{aligned}s^2 &= \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2 \\ &= \frac{1}{(n-1)} \left\{ \left(\sum_{i=1}^n x_i^2 \right) - n\bar{x}^2 \right\} .\end{aligned}$$

The second formula is easier for calculations.

3.2.2 Sample variance and standard deviation

So for our data set $\{0, 3, 2, 0\}$, we have

$$\sum_{i=1}^4 x_i^2 = 0^2 + 3^2 + 2^2 + 0^2 = 13$$

So

3.2.2 Sample variance and standard deviation

So for our data set $\{0, 3, 2, 0\}$, we have

$$\sum_{i=1}^4 x_i^2 = 0^2 + 3^2 + 2^2 + 0^2 = 13$$

So

$$s^2 = \frac{1}{n-1} \left\{ \left(\sum_{i=1}^n x_i^2 \right) - n\bar{x}^2 \right\} =$$

3.2.2 Sample variance and standard deviation

So for our data set $\{0, 3, 2, 0\}$, we have

$$\sum_{i=1}^4 x_i^2 = 0^2 + 3^2 + 2^2 + 0^2 = 13$$

So

$$s^2 = \frac{1}{n-1} \left\{ \left(\sum_{i=1}^n x_i^2 \right) - n\bar{x}^2 \right\} = \frac{1}{3} (13 - 4 \times 1.25^2) =$$

3.2.2 Sample variance and standard deviation

So for our data set $\{0, 3, 2, 0\}$, we have

$$\sum_{i=1}^4 x_i^2 = 0^2 + 3^2 + 2^2 + 0^2 = 13$$

So

$$s^2 = \frac{1}{n-1} \left\{ \left(\sum_{i=1}^n x_i^2 \right) - n\bar{x}^2 \right\} = \frac{1}{3} (13 - 4 \times 1.25^2) = 2.25.$$

3.2.2 Sample variance and standard deviation

So for our data set $\{0, 3, 2, 0\}$, we have

$$\sum_{i=1}^4 x_i^2 = 0^2 + 3^2 + 2^2 + 0^2 = 13$$

So

$$s^2 = \frac{1}{n-1} \left\{ \left(\sum_{i=1}^n x_i^2 \right) - n\bar{x}^2 \right\} = \frac{1}{3} (13 - 4 \times 1.25^2) = 2.25.$$

The sample standard deviation, s , is the square root of the sample variance, i.e. for our toy example $s = \sqrt{2.25} = 1.5$.

3.2.2 Sample variance and standard deviation

So for our data set $\{0, 3, 2, 0\}$, we have

$$\sum_{i=1}^4 x_i^2 = 0^2 + 3^2 + 2^2 + 0^2 = 13$$

So

$$s^2 = \frac{1}{n-1} \left\{ \left(\sum_{i=1}^n x_i^2 \right) - n\bar{x}^2 \right\} = \frac{1}{3} (13 - 4 \times 1.25^2) = 2.25.$$

The sample standard deviation, s , is the square root of the sample variance, i.e. for our toy example $s = \sqrt{2.25} = 1.5$.

The standard deviation is often preferred as it is in the original units of the data.

3.2.3 Examples

Calculate the variance of $\{1, 7, 13, 14, 20\}$.

3.2.3 Examples

Calculate the variance of $\{1, 7, 13, 14, 20\}$.

Solution

We calculate

$$\sum_{i=1}^5 x_i^2 =$$

3.2.3 Examples

Calculate the variance of $\{1, 7, 13, 14, 20\}$.

Solution

We calculate

$$\sum_{i=1}^5 x_i^2 = 1^2 +$$

3.2.3 Examples

Calculate the variance of $\{1, 7, 13, 14, 20\}$.

Solution

We calculate

$$\sum_{i=1}^5 x_i^2 = 1^2 + 7^2 +$$

3.2.3 Examples

Calculate the variance of $\{1, 7, 13, 14, 20\}$.

Solution

We calculate

$$\sum_{i=1}^5 x_i^2 = 1^2 + 7^2 + 13^2 +$$

3.2.3 Examples

Calculate the variance of $\{1, 7, 13, 14, 20\}$.

Solution

We calculate

$$\sum_{i=1}^5 x_i^2 = 1^2 + 7^2 + 13^2 + 14^2 +$$

3.2.3 Examples

Calculate the variance of $\{1, 7, 13, 14, 20\}$.

Solution

We calculate

$$\sum_{i=1}^5 x_i^2 = 1^2 + 7^2 + 13^2 + 14^2 + 20^2 =$$

3.2.3 Examples

Calculate the variance of $\{1, 7, 13, 14, 20\}$.

Solution

We calculate

$$\sum_{i=1}^5 x_i^2 = 1^2 + 7^2 + 13^2 + 14^2 + 20^2 = 815$$

3.2.3 Examples

Calculate the variance of $\{1, 7, 13, 14, 20\}$.

Solution

We calculate

$$\sum_{i=1}^5 x_i^2 = 1^2 + 7^2 + 13^2 + 14^2 + 20^2 = 815$$

and $n\bar{x}^2 = 605$.

3.2.3 Examples

Calculate the variance of $\{1, 7, 13, 14, 20\}$.

Solution

We calculate

$$\sum_{i=1}^5 x_i^2 = 1^2 + 7^2 + 13^2 + 14^2 + 20^2 = 815$$

and $n\bar{x}^2 = 605$. So

$$s^2 = \frac{815 - 605}{4} = 52.5 .$$

3.2.3 Examples

Calculate the variance of $\{7, 13, 14, 20\}$.

3.2.3 Examples

Calculate the variance of $\{7, 13, 14, 20\}$.

Solution

3.2.3 Examples

Calculate the variance of $\{7, 13, 14, 20\}$.

Solution

We calculate

$$\sum_{i=1}^4 x_i^2 =$$

3.2.3 Examples

Calculate the variance of $\{7, 13, 14, 20\}$.

Solution

We calculate

$$\sum_{i=1}^4 x_i^2 = 7^2 +$$

3.2.3 Examples

Calculate the variance of $\{7, 13, 14, 20\}$.

Solution

We calculate

$$\sum_{i=1}^4 x_i^2 = 7^2 + 13^2 +$$

3.2.3 Examples

Calculate the variance of $\{7, 13, 14, 20\}$.

Solution

We calculate

$$\sum_{i=1}^4 x_i^2 = 7^2 + 13^2 + 14^2 +$$

3.2.3 Examples

Calculate the variance of $\{7, 13, 14, 20\}$.

Solution

We calculate

$$\sum_{i=1}^4 x_i^2 = 7^2 + 13^2 + 14^2 + 20^2 =$$

3.2.3 Examples

Calculate the variance of $\{7, 13, 14, 20\}$.

Solution

We calculate

$$\sum_{i=1}^4 x_i^2 = 7^2 + 13^2 + 14^2 + 20^2 = 814$$

3.2.3 Examples

Calculate the variance of $\{7, 13, 14, 20\}$.

Solution

We calculate

$$\sum_{i=1}^4 x_i^2 = 7^2 + 13^2 + 14^2 + 20^2 = 814$$

and $n\bar{x}^2 = 729$.

3.2.3 Examples

Calculate the variance of $\{7, 13, 14, 20\}$.

Solution

We calculate

$$\sum_{i=1}^4 x_i^2 = 7^2 + 13^2 + 14^2 + 20^2 = 814$$

and $n\bar{x}^2 = 729$. So

$$s^2 = \frac{814 - 729}{3} = 28.3.$$

3.2.3 Examples

Calculate the variance of $\{1, 7, 13, 14, 200\}$.

3.2.3 Examples

Calculate the variance of $\{1, 7, 13, 14, 200\}$.

Solution

3.2.3 Examples

Calculate the variance of $\{1, 7, 13, 14, 200\}$.

Solution

We calculate

$$\sum_{i=1}^5 x_i^2 =$$

3.2.3 Examples

Calculate the variance of $\{1, 7, 13, 14, 200\}$.

Solution

We calculate

$$\sum_{i=1}^5 x_i^2 = 40415$$

3.2.3 Examples

Calculate the variance of $\{1, 7, 13, 14, 200\}$.

Solution

We calculate

$$\sum_{i=1}^5 x_i^2 = 40415$$

and $n\bar{x}^2 = 11045$.

3.2.3 Examples

Calculate the variance of $\{1, 7, 13, 14, 200\}$.

Solution

We calculate

$$\sum_{i=1}^5 x_i^2 = 40415$$

and $n\bar{x}^2 = 11045$. So

$$s^2 = 7342.5 .$$

3.2.4 Quartiles and the interquartile range

The upper and lower quartiles are defined as follows:

- $Q1$: Lower quartile = $(n + 1)/4^{\text{th}}$ smallest observation.
- $Q3$: Upper quartile = $3(n + 1)/4^{\text{th}}$ smallest observation.

We can linearly interpolate between adjacent observations if necessary.

3.2.4 Quartiles and the interquartile range

The upper and lower quartiles are defined as follows:

- $Q1$: Lower quartile = $(n + 1)/4^{\text{th}}$ smallest observation.
- $Q3$: Upper quartile = $3(n + 1)/4^{\text{th}}$ smallest observation.

We can linearly interpolate between adjacent observations if necessary.

The interquartile range is the difference between the third and first quartile, i.e.

$$IQR = Q3 - Q1.$$

Examples

For the following data sets, calculate the inter-quartile range:

Examples: $\{5, 6, 7, 8\}$

The lower quartile is the 5/4 smallest observation, i.e. $x_{(1.25)}$.

Examples: $\{5, 6, 7, 8\}$

The lower quartile is the $5/4$ smallest observation, i.e. $x_{(1.25)}$.

The value of $x_{(1)} = 5$ and $x_{(2)} = 6$.

Examples: $\{5, 6, 7, 8\}$

The lower quartile is the $5/4$ smallest observation, i.e. $x_{(1.25)}$.

The value of $x_{(1)} = 5$ and $x_{(2)} = 6$. So

$$x_{(1.25)} =$$

Examples: $\{5, 6, 7, 8\}$

The lower quartile is the 5/4 smallest observation, i.e. $x_{(1.25)}$.

The value of $x_{(1)} = 5$ and $x_{(2)} = 6$. So

$$x_{(1.25)} = x_{(1)} +$$

Examples: $\{5, 6, 7, 8\}$

The lower quartile is the 5/4 smallest observation, i.e. $x_{(1.25)}$.

The value of $x_{(1)} = 5$ and $x_{(2)} = 6$. So

$$x_{(1.25)} = x_{(1)} + 0.25 \times (x_{(2)} - x_{(1)}) =$$

Examples: $\{5, 6, 7, 8\}$

The lower quartile is the 5/4 smallest observation, i.e. $x_{(1.25)}$.

The value of $x_{(1)} = 5$ and $x_{(2)} = 6$. So

$$x_{(1.25)} = x_{(1)} + 0.25 \times (x_{(2)} - x_{(1)}) = 5 + \frac{6 - 5}{4} =$$

Examples: $\{5, 6, 7, 8\}$

The lower quartile is the 5/4 smallest observation, i.e. $x_{(1.25)}$.

The value of $x_{(1)} = 5$ and $x_{(2)} = 6$. So

$$x_{(1.25)} = x_{(1)} + 0.25 \times (x_{(2)} - x_{(1)}) = 5 + \frac{6 - 5}{4} = 5.25$$

Examples: $\{5, 6, 7, 8\}$

The lower quartile is the $5/4$ smallest observation, i.e. $x_{(1.25)}$.

The value of $x_{(1)} = 5$ and $x_{(2)} = 6$. So

$$x_{(1.25)} = x_{(1)} + 0.25 \times (x_{(2)} - x_{(1)}) = 5 + \frac{6 - 5}{4} = 5.25$$

Similarly, the upper quartile is the $3 \times (4 + 1)/4 = 3.75^{\text{th}}$ smallest observation. So

$$x_{(3.75)} =$$

Examples: $\{5, 6, 7, 8\}$

The lower quartile is the $5/4$ smallest observation, i.e. $x_{(1.25)}$.

The value of $x_{(1)} = 5$ and $x_{(2)} = 6$. So

$$x_{(1.25)} = x_{(1)} + 0.25 \times (x_{(2)} - x_{(1)}) = 5 + \frac{6 - 5}{4} = 5.25$$

Similarly, the upper quartile is the $3 \times (4 + 1)/4 = 3.75^{\text{th}}$ smallest observation. So

$$x_{(3.75)} = x_{(3)} +$$

Examples: $\{5, 6, 7, 8\}$

The lower quartile is the $5/4$ smallest observation, i.e. $x_{(1.25)}$.

The value of $x_{(1)} = 5$ and $x_{(2)} = 6$. So

$$x_{(1.25)} = x_{(1)} + 0.25 \times (x_{(2)} - x_{(1)}) = 5 + \frac{6 - 5}{4} = 5.25$$

Similarly, the upper quartile is the $3 \times (4 + 1)/4 = 3.75^{\text{th}}$ smallest observation. So

$$x_{(3.75)} = x_{(3)} + 0.75 \times (x_{(4)} - x_{(3)}) =$$

Examples: $\{5, 6, 7, 8\}$

The lower quartile is the $5/4$ smallest observation, i.e. $x_{(1.25)}$.

The value of $x_{(1)} = 5$ and $x_{(2)} = 6$. So

$$x_{(1.25)} = x_{(1)} + 0.25 \times (x_{(2)} - x_{(1)}) = 5 + \frac{6 - 5}{4} = 5.25$$

Similarly, the upper quartile is the $3 \times (4 + 1)/4 = 3.75^{\text{th}}$ smallest observation. So

$$x_{(3.75)} = x_{(3)} + 0.75 \times (x_{(4)} - x_{(3)}) = 7.75$$

So

Examples: $\{5, 6, 7, 8\}$

The lower quartile is the $5/4$ smallest observation, i.e. $x_{(1.25)}$.

The value of $x_{(1)} = 5$ and $x_{(2)} = 6$. So

$$x_{(1.25)} = x_{(1)} + 0.25 \times (x_{(2)} - x_{(1)}) = 5 + \frac{6 - 5}{4} = 5.25$$

Similarly, the upper quartile is the $3 \times (4 + 1)/4 = 3.75^{\text{th}}$ smallest observation. So

$$x_{(3.75)} = x_{(3)} + 0.75 \times (x_{(4)} - x_{(3)}) = 7.75$$

So

$$IQR = 7.75 - 5.25 = 2.5.$$

Examples: $\{10, 15, 20, 25, 50\}$

The lower quartile is the $(5 + 1)/4$ smallest observation, i.e. $x_{(1.5)}$, so

Examples: $\{10, 15, 20, 25, 50\}$

The lower quartile is the $(5 + 1)/4$ smallest observation, i.e. $x_{(1.5)}$, so

$$x_{(1.5)} =$$

Examples: $\{10, 15, 20, 25, 50\}$

The lower quartile is the $(5 + 1)/4$ smallest observation, i.e. $x_{(1.5)}$, so

$$x_{(1.5)} = x_{(1)} +$$

Examples: $\{10, 15, 20, 25, 50\}$

The lower quartile is the $(5 + 1)/4$ smallest observation, i.e. $x_{(1.5)}$, so

$$x_{(1.5)} = x_{(1)} + 0.5 \times (x_{(2)} - x_{(1)}) =$$

Examples: $\{10, 15, 20, 25, 50\}$

The lower quartile is the $(5 + 1)/4$ smallest observation, i.e. $x_{(1.5)}$, so

$$x_{(1.5)} = x_{(1)} + 0.5 \times (x_{(2)} - x_{(1)}) = 12.5$$

Examples: $\{10, 15, 20, 25, 50\}$

The lower quartile is the $(5 + 1)/4$ smallest observation, i.e. $x_{(1.5)}$, so

$$x_{(1.5)} = x_{(1)} + 0.5 \times (x_{(2)} - x_{(1)}) = 12.5$$

Similarly, the upper quartile is the $3 \times (5 + 1)/4 = 4.5^{\text{th}}$ smallest observation.

Examples: $\{10, 15, 20, 25, 50\}$

The lower quartile is the $(5 + 1)/4$ smallest observation, i.e. $x_{(1.5)}$, so

$$x_{(1.5)} = x_{(1)} + 0.5 \times (x_{(2)} - x_{(1)}) = 12.5$$

Similarly, the upper quartile is the $3 \times (5 + 1)/4 = 4.5^{\text{th}}$ smallest observation. So

$$x_{(4.5)} = x_{(4)} + 0.5 \times (x_{(5)} - x_{(4)}) = 37.5.$$

So

Examples: $\{10, 15, 20, 25, 50\}$

The lower quartile is the $(5 + 1)/4$ smallest observation, i.e. $x_{(1.5)}$, so

$$x_{(1.5)} = x_{(1)} + 0.5 \times (x_{(2)} - x_{(1)}) = 12.5$$

Similarly, the upper quartile is the $3 \times (5 + 1)/4 = 4.5^{\text{th}}$ smallest observation. So

$$x_{(4.5)} = x_{(4)} + 0.5 \times (x_{(5)} - x_{(4)}) = 37.5.$$

So

$$IQR = 37.5 - 12.5 = 25.$$

Examples: $\{1, 7, 13, 14, 20\}$

Examples: $\{1, 7, 13, 14, 20\}$

The lower quartile is the $(5 + 1)/4$ smallest observation, i.e. $x_{(1.5)}$, so

$$x_{(1.5)} =$$

Examples: $\{1, 7, 13, 14, 20\}$

The lower quartile is the $(5 + 1)/4$ smallest observation, i.e. $x_{(1.5)}$, so

$$x_{(1.5)} = x_{(1)} + 0.5 \times (x_{(2)} - x_{(1)}) = 4$$

Examples: $\{1, 7, 13, 14, 20\}$

The lower quartile is the $(5 + 1)/4$ smallest observation, i.e. $x_{(1.5)}$, so

$$x_{(1.5)} = x_{(1)} + 0.5 \times (x_{(2)} - x_{(1)}) = 4$$

Similarly, the upper quartile is the $3 \times (5 + 1)/4 = 4.5^{\text{th}}$ smallest observation.

Examples: $\{1, 7, 13, 14, 20\}$

The lower quartile is the $(5 + 1)/4$ smallest observation, i.e. $x_{(1.5)}$, so

$$x_{(1.5)} = x_{(1)} + 0.5 \times (x_{(2)} - x_{(1)}) = 4$$

Similarly, the upper quartile is the $3 \times (5 + 1)/4 = 4.5^{\text{th}}$ smallest observation. So

$$x_{(4.5)} =$$

Examples: $\{1, 7, 13, 14, 20\}$

The lower quartile is the $(5 + 1)/4$ smallest observation, i.e. $x_{(1.5)}$, so

$$x_{(1.5)} = x_{(1)} + 0.5 \times (x_{(2)} - x_{(1)}) = 4$$

Similarly, the upper quartile is the $3 \times (5 + 1)/4 = 4.5^{\text{th}}$ smallest observation. So

$$x_{(4.5)} = x_{(4)} +$$

Examples: $\{1, 7, 13, 14, 20\}$

The lower quartile is the $(5 + 1)/4$ smallest observation, i.e. $x_{(1.5)}$, so

$$x_{(1.5)} = x_{(1)} + 0.5 \times (x_{(2)} - x_{(1)}) = 4$$

Similarly, the upper quartile is the $3 \times (5 + 1)/4 = 4.5^{\text{th}}$ smallest observation. So

$$x_{(4.5)} = x_{(4)} + 0.5 \times (x_{(5)} - x_{(4)}) =$$

Examples: $\{1, 7, 13, 14, 20\}$

The lower quartile is the $(5 + 1)/4$ smallest observation, i.e. $x_{(1.5)}$, so

$$x_{(1.5)} = x_{(1)} + 0.5 \times (x_{(2)} - x_{(1)}) = 4$$

Similarly, the upper quartile is the $3 \times (5 + 1)/4 = 4.5^{\text{th}}$ smallest observation. So

$$x_{(4.5)} = x_{(4)} + 0.5 \times (x_{(5)} - x_{(4)}) = 17.$$

So

Examples: $\{1, 7, 13, 14, 20\}$

The lower quartile is the $(5 + 1)/4$ smallest observation, i.e. $x_{(1.5)}$, so

$$x_{(1.5)} = x_{(1)} + 0.5 \times (x_{(2)} - x_{(1)}) = 4$$

Similarly, the upper quartile is the $3 \times (5 + 1)/4 = 4.5^{\text{th}}$ smallest observation. So

$$x_{(4.5)} = x_{(4)} + 0.5 \times (x_{(5)} - x_{(4)}) = 17.$$

So

$$IQR = 17 - 4 = 13.$$

Examples: $\{7, 13, 14, 20\}$

The lower quartile is the $5/4$ smallest observation, i.e. $x_{(1.25)}$. The value of $x_{(1)} = 7$ and $x_{(2)} = 13$.

Examples: $\{7, 13, 14, 20\}$

The lower quartile is the 5/4 smallest observation, i.e. $x_{(1.25)}$. The value of $x_{(1)} = 7$ and $x_{(2)} = 13$. So

$$x_{(1.25)} = x_{(1)} + 0.25 \times (x_{(2)} - x_{(1)}) = 7 + \frac{13 - 7}{4} = 8.5$$

Examples: $\{7, 13, 14, 20\}$

The lower quartile is the 5/4 smallest observation, i.e. $x_{(1.25)}$. The value of $x_{(1)} = 7$ and $x_{(2)} = 13$. So

$$x_{(1.25)} = x_{(1)} + 0.25 \times (x_{(2)} - x_{(1)}) = 7 + \frac{13 - 7}{4} = 8.5$$

Examples: $\{7, 13, 14, 20\}$

The lower quartile is the $5/4$ smallest observation, i.e. $x_{(1.25)}$. The value of $x_{(1)} = 7$ and $x_{(2)} = 13$. So

$$x_{(1.25)} = x_{(1)} + 0.25 \times (x_{(2)} - x_{(1)}) = 7 + \frac{13 - 7}{4} = 8.5$$

Similarly, the upper quartile is the $3 \times (4 + 1)/4 = 3.75^{\text{th}}$ smallest observation.

Examples: $\{7, 13, 14, 20\}$

The lower quartile is the $5/4$ smallest observation, i.e. $x_{(1.25)}$. The value of $x_{(1)} = 7$ and $x_{(2)} = 13$. So

$$x_{(1.25)} = x_{(1)} + 0.25 \times (x_{(2)} - x_{(1)}) = 7 + \frac{13 - 7}{4} = 8.5$$

Similarly, the upper quartile is the $3 \times (4 + 1)/4 = 3.75^{\text{th}}$ smallest observation. So

$$x_{(3.75)} = x_{(3)} + 0.75 \times (x_{(4)} - x_{(3)}) = 18.5.$$

So

Examples: $\{7, 13, 14, 20\}$

The lower quartile is the $5/4$ smallest observation, i.e. $x_{(1.25)}$. The value of $x_{(1)} = 7$ and $x_{(2)} = 13$. So

$$x_{(1.25)} = x_{(1)} + 0.25 \times (x_{(2)} - x_{(1)}) = 7 + \frac{13 - 7}{4} = 8.5$$

Similarly, the upper quartile is the $3 \times (4 + 1)/4 = 3.75^{\text{th}}$ smallest observation. So

$$x_{(3.75)} = x_{(3)} + 0.75 \times (x_{(4)} - x_{(3)}) = 18.5.$$

So

$$IQR = 18.5 - 8.5 = 10.$$

3.3 Using R

Command	Comment	Example
mean	Calculates the mean of a vector	mean(x)
sd	Calculates the standard deviation of a vector	sd(x)
var	Calculates the variance of a vector	var(x)
quantile	The vector quartiles. Make sure you use type=6	quantile(x, type=6)
range	Calculates the vector range	range(x)
summary	Calculates the quartiles However , it doesn't use type=6 quartiles.	summary(x)

Table: Summary of R commands in this chapter.

Command	Comment	Example
mean	Calculates the mean of a vector	mean(x)
sd	Calculates the standard deviation of a vector	sd(x)
var	Calculates the variance of a vector	var(x)
quantile	The vector quartiles. Make sure you use type=6	quantile(x, type=6)
range	Calculates the vector range	range(x)
summary	Calculates the quartiles However, it doesn't use type=6 quartiles.	summary(x)

Table: Summary of R commands in this chapter.