MAS1802: Problem Solving II (Statistical Computing with R)

Dr Lee Fawcett

School of Mathematics & Statistics

Semester 2, 2015/2016

1.1 Housekeeping

MAS1802 – A tale of two halves:

- *Before* Easter: Statistical computing with **R**
- *After* Easter: pencil-and-paper problem solving in groups

These slides refer to the computing part only.

1.1 Housekeeping

MAS1802 – A tale of two halves:

- *Before* Easter: Statistical computing with **R**
- *After* Easter: pencil-and-paper problem solving in groups

These slides refer to the computing part only.

1.1 Housekeeping

Things we've changed since MAS1801:

- You didn't like how MatLab was self-taught: I have timetabled a weekly lecture for the computing part of MAS1802
- You wanted more support in the computer practicals: I have scheduled a weekly drop-in, two days before the assignment deadline, in the computer cluster
- You thought the project should have been more structured: The nature of this course requires some open-ended work, however the project for the computing part of MAS1802 will have more direction
- You didn't like the fact that you had to view the MatLab notes on the screen: I will give out a full set of printed notes in the lectures
- MAS1801 - pencil-and-paper, then computing: MAS1802 - computing, then pencil-and-paper
- Some changes will be made to the pencil-and-paper part of the module too – e.g. no presentations...
- You wanted sweeties in class...

1.1 Housekeeping

Things we've changed since MAS1801:

- You didn't like how MatLab was self-taught: I have timetabled a weekly lecture for the computing part of MAS1802
- You wanted more support in the computer practicals: I have scheduled a weekly drop-in, two days before the assignment deadline, in the computer cluster
- You thought the project should have been more structured: The nature of this course requires some open-ended work, however the project for the computing part of MAS1802 will have more direction
- You didn't like the fact that you had to view the MatLab notes on the screen: I will give out a full set of printed notes in the lectures
- MAS1801 - pencil-and-paper, then computing: MAS1802 - computing, then pencil-and-paper
- Some changes will be made to the pencil-and-paper part of the module too – e.g. no presentations...
- You wanted sweeties in class...

1.1 Housekeeping

Things we've changed since MAS1801:

- You didn't like how MatLab was self-taught: I have timetabled a weekly lecture for the computing part of MAS1802
- You wanted more support in the computer practicals: I have scheduled a weekly drop-in, two days before the assignment deadline, in the computer cluster
- You thought the project should have been more structured: The nature of this course requires some open-ended work, however the project for the computing part of MAS1802 will have more direction
- You didn't like the fact that you had to view the MatLab notes on the screen: I will give out a full set of printed notes in the lectures
- MAS1801 - pencil-and-paper, then computing: MAS1802 - computing, then pencil-and-paper
- Some changes will be made to the pencil-and-paper part of the module too – e.g. no presentations...
- You wanted sweeties in class...

1.1 Housekeeping

Things we've changed since MAS1801:

- You didn't like how MatLab was self-taught: I have timetabled a weekly lecture for the computing part of MAS1802
- You wanted more support in the computer practicals: I have scheduled a weekly drop-in, two days before the assignment deadline, in the computer cluster
- You thought the project should have been more structured: The nature of this course requires some open-ended work, however the project for the computing part of MAS1802 will have more direction
- You didn't like the fact that you had to view the MatLab notes on the screen: I will give out a full set of printed notes in the lectures
- MAS1801 - pencil-and-paper, then computing: MAS1802 - computing, then pencil-and-paper
- Some changes will be made to the pencil-and-paper part of the module too – e.g. no presentations...
- You wanted sweeties in class...

1.1 Housekeeping

Things we've changed since MAS1801:

- You didn't like how MatLab was self-taught: I have timetabled a weekly lecture for the computing part of MAS1802
- You wanted more support in the computer practicals: I have scheduled a weekly drop-in, two days before the assignment deadline, in the computer cluster
- You thought the project should have been more structured: The nature of this course requires some open-ended work, however the project for the computing part of MAS1802 will have more direction
- You didn't like the fact that you had to view the MatLab notes on the screen: I will give out a full set of printed notes in the lectures
- MAS1801 - pencil-and-paper, then computing: MAS1802 - computing, then pencil-and-paper
- Some changes will be made to the pencil-and-paper part of the module too – e.g. no presentations...
- You wanted sweeties in class...

1.1 Housekeeping

Things we've changed since MAS1801:

- You didn't like how MatLab was self-taught: I have timetabled a weekly lecture for the computing part of MAS1802
- You wanted more support in the computer practicals: I have scheduled a weekly drop-in, two days before the assignment deadline, in the computer cluster
- You thought the project should have been more structured: The nature of this course requires some open-ended work, however the project for the computing part of MAS1802 will have more direction
- You didn't like the fact that you had to view the MatLab notes on the screen: I will give out a full set of printed notes in the lectures
- MAS1801 - pencil-and-paper, then computing: MAS1802 - computing, then pencil-and-paper
- Some changes will be made to the pencil-and-paper part of the module too – e.g. no presentations...
- You wanted sweeties in class...

1.1 Housekeeping

Things we've changed since MAS1801:

- You didn't like how MatLab was self-taught: I have timetabled a weekly lecture for the computing part of MAS1802
- You wanted more support in the computer practicals: I have scheduled a weekly drop-in, two days before the assignment deadline, in the computer cluster
- You thought the project should have been more structured: The nature of this course requires some open-ended work, however the project for the computing part of MAS1802 will have more direction
- You didn't like the fact that you had to view the MatLab notes on the screen: I will give out a full set of printed notes in the lectures
- MAS1801 - pencil-and-paper, then computing: MAS1802 - computing, then pencil-and-paper
- Some changes will be made to the pencil-and-paper part of the module too – e.g. no presentations...
- You wanted sweeties in class...

1.1 Housekeeping

Things we've changed since MAS1801:

- You didn't like how MatLab was self-taught: I have timetabled a weekly lecture for the computing part of MAS1802
- You wanted more support in the computer practicals: I have scheduled a weekly drop-in, two days before the assignment deadline, in the computer cluster
- You thought the project should have been more structured: The nature of this course requires some open-ended work, however the project for the computing part of MAS1802 will have more direction
- You didn't like the fact that you had to view the MatLab notes on the screen: I will give out a full set of printed notes in the lectures
- MAS1801 - pencil-and-paper, then computing: MAS1802 - computing, then pencil-and-paper
- Some changes will be made to the pencil-and-paper part of the module too – e.g. no presentations...
- You wanted sweeties in class...

1.1 Housekeeping

Things we've changed since MAS1801:

- You didn't like how MatLab was self-taught: I have timetabled a weekly lecture for the computing part of MAS1802
- You wanted more support in the computer practicals: I have scheduled a weekly drop-in, two days before the assignment deadline, in the computer cluster
- You thought the project should have been more structured: The nature of this course requires some open-ended work, however the project for the computing part of MAS1802 will have more direction
- You didn't like the fact that you had to view the MatLab notes on the screen: I will give out a full set of printed notes in the lectures
- MAS1801 - pencil-and-paper, then computing: MAS1802 - computing, then pencil-and-paper
- Some changes will be made to the pencil-and-paper part of the module too – e.g. no presentations...
- You wanted sweeties in class...

1.1 Housekeeping

Things we've changed since MAS1801:

- You didn't like how MatLab was self-taught: I have timetabled a weekly lecture for the computing part of MAS1802
- You wanted more support in the computer practicals: I have scheduled a weekly drop-in, two days before the assignment deadline, in the computer cluster
- You thought the project should have been more structured: The nature of this course requires some open-ended work, however the project for the computing part of MAS1802 will have more direction
- You didn't like the fact that you had to view the MatLab notes on the screen: I will give out a full set of printed notes in the lectures
- MAS1801 - pencil-and-paper, then computing: MAS1802 - computing, then pencil-and-paper
- Some changes will be made to the pencil-and-paper part of the module too – e.g. no presentations...
- You wanted sweeties in class...

1.1 Housekeeping

Things we've changed since MAS1801:

- You didn't like how MatLab was self-taught: I have timetabled a weekly lecture for the computing part of MAS1802
- You wanted more support in the computer practicals: I have scheduled a weekly drop-in, two days before the assignment deadline, in the computer cluster
- You thought the project should have been more structured: The nature of this course requires some open-ended work, however the project for the computing part of MAS1802 will have more direction
- You didn't like the fact that you had to view the MatLab notes on the screen: I will give out a full set of printed notes in the lectures
- MAS1801 - pencil-and-paper, then computing: MAS1802 - computing, then pencil-and-paper
- Some changes will be made to the pencil-and-paper part of the module too – e.g. no presentations...
- You wanted sweeties in class...

1.1 Housekeeping

Things we've changed since MAS1801:

- You didn't like how MatLab was self-taught: I have timetabled a weekly lecture for the computing part of MAS1802
- You wanted more support in the computer practicals: I have scheduled a weekly drop-in, two days before the assignment deadline, in the computer cluster
- You thought the project should have been more structured: The nature of this course requires some open-ended work, however the project for the computing part of MAS1802 will have more direction
- You didn't like the fact that you had to view the MatLab notes on the screen: I will give out a full set of printed notes in the lectures
- MAS1801 - pencil-and-paper, then computing: MAS1802 - computing, then pencil-and-paper
- Some changes will be made to the pencil-and-paper part of the module too – e.g. no presentations...
- You wanted sweeties in class...

1.1 Housekeeping

Computing – General course structure:

Type of session	Day	Time	Place
Lecture	Thursday	9–10	Herschel LT2
Practical	Friday	9–11	Herschel PC
Drop-in	Monday	10–11	Herschel PC (Blue zone)
Office hour	Monday	1–2	My office (2.07)
Deadline:	*Wednesday*	*4pm/11.45pm*	*General office/online*

- This week: No Monday drop-in, no Friday practical
- Next week: Monday drop-in $\longrightarrow$ Introductory practical; no Wednesday deadline

1.1 Housekeeping

Computing – General course structure:

Type of session	Day	Time	Place
Lecture	Thursday	9–10	Herschel LT2
Practical	Friday	9–11	Herschel PC
Drop-in	Monday	10–11	Herschel PC (Blue zone)
Office hour	Monday	1–2	My office (2.07)
Deadline:	*Wednesday*	*4pm/11.45pm*	*General office/online*

- This week: No Monday drop-in, no Friday practical
- Next week: Monday drop-in $\longrightarrow$ Introductory practical; no Wednesday deadline

1.1 Housekeeping

Computing – General course structure:

Type of session	Day	Time	Place
Lecture	Thursday	9–10	Herschel LT2
Practical	Friday	9–11	Herschel PC
Drop-in	Monday	10–11	Herschel PC (Blue zone)
Office hour	Monday	1–2	My office (2.07)
Deadline:	*Wednesday*	*4pm/11.45pm*	*General office/online*

- This week: No Monday drop-in, no Friday practical
- Next week: Monday drop-in $\longrightarrow$ Introductory practical; no Wednesday deadline

1.1.1 Lecturer information: who am I?

- The module leader, and lecturer for the computing part of 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

1.1.3 Lectures

- Fewer lectures for this course than for other modules (nature of the course)
- The calendar at the front of this handout contains a full schedule for the computing part of this course
- A summarised version of the notes will be used as slides - interspersed with quizes and other really fun things. Pay attention!
- Notes, slides and practical sheets will be posted to BlackBoard, and a dedicated webpage, soon after the lecture/practical session
- Lectures will be recorded via ReCap
- Check your `ncl` e-mail regularly: class cancellations etc.

1.1.3 Lectures

- Fewer lectures for this course than for other modules (nature of the course)
- The calendar at the front of this handout contains a full schedule for the computing part of this course
- A summarised version of the notes will be used as slides - interspersed with quizes and other really fun things. Pay attention!
- Notes, slides and practical sheets will be posted to BlackBoard, and a dedicated webpage, soon after the lecture/practical session
- Lectures will be recorded via ReCap
- Check your `ncl` e-mail regularly: class cancellations etc.

1.1.3 Lectures

- Fewer lectures for this course than for other modules (nature of the course)
- The calendar at the front of this handout contains a full schedule for the computing part of this course
- A summarised version of the notes will be used as slides - interspersed with quizes and other really fun things. Pay attention!
- Notes, slides and practical sheets will be posted to BlackBoard, and a dedicated webpage, soon after the lecture/practical session
- Lectures will be recorded via ReCap
- Check your `ncl` e-mail regularly: class cancellations etc.

1.1.3 Lectures

- Fewer lectures for this course than for other modules (nature of the course)
- The calendar at the front of this handout contains a full schedule for the computing part of this course
- A summarised version of the notes will be used as slides - interspersed with quizes and other really fun things. Pay attention!
- Notes, slides and practical sheets will be posted to BlackBoard, and a dedicated webpage, soon after the lecture/practical session
- Lectures will be recorded via ReCap
- Check your ncl e-mail regularly: class cancellations etc.

1.1.3 Lectures

- Fewer lectures for this course than for other modules (nature of the course)
- The calendar at the front of this handout contains a full schedule for the computing part of this course
- A summarised version of the notes will be used as slides - interspersed with quizes and other really fun things. Pay attention!
- Notes, slides and practical sheets will be posted to BlackBoard, and a dedicated webpage, soon after the lecture/practical session
- Lectures will be recorded via ReCap
- Check your `ncl` e-mail regularly: class cancellations etc.

1.1.4 Computer practicals

- Lots of support - make good use of this!






- Attendance register taken
- You are more likely to get high marks for the coursework if:
  - You have attended the lecture: In practicals, I will assume you have been to the lecture the day before
  - You don't panic! Read the notes carefully, and type code slowly
  - You ask for help
  - You try to have fun
  - You are organised!

1.1.4 Computer practicals

- Lots of support - make good use of this!






- Attendance register taken
- You are more likely to get high marks for the coursework if:
  - You have attended the lecture: In practicals, I will assume you have been to the lecture the day before
  - You don't panic! Read the notes carefully, and type code slowly
  - You ask for help
  - You try to have fun
  - You are organised!

1.1.4 Computer practicals

- Lots of support - make good use of this!






- Attendance register taken
- You are more likely to get high marks for the coursework if:
  - You have attended the lecture: In practicals, I will assume you have been to the lecture the day before
  - You don't panic! Read the notes carefully, and type code slowly
  - You ask for help
  - You try to have fun
  - You are organised!

1.1.4 Computer practicals

- Lots of support - make good use of this!






- Attendance register taken
- You are more likely to get high marks for the coursework if:
  - You have attended the lecture: In practicals, I will assume you have been to the lecture the day before
  - You don't panic! Read the notes carefully, and type code slowly
  - You ask for help
  - You try to have fun
  - You are organised!

1.1.4 Computer practicals

- Lots of support - make good use of this!






- Attendance register taken
- You are more likely to get high marks for the coursework if:
  - You have attended the lecture: In practicals, I will assume you have been to the lecture the day before
  - You don't panic! Read the notes carefully, and type code slowly
  - You ask for help
  - You try to have fun
  - You are organised!

1.1.4 Computer practicals

- Lots of support - make good use of this!






- Attendance register taken
- You are more likely to get high marks for the coursework if:
  - You have attended the lecture: In practicals, I will assume you have been to the lecture the day before
  - You don't panic! Read the notes carefully, and type code slowly
  - You ask for help
  - You try to have fun
  - You are organised!

1.1.4 Computer practicals

- Lots of support - make good use of this!






- Attendance register taken
- You are more likely to get high marks for the coursework if:
  - You have attended the lecture: In practicals, I will assume you have been to the lecture the day before
  - You don't panic! Read the notes carefully, and type code slowly
  - You ask for help
  - You try to have fun
  - You are organised!

1.1.5 Assessment

- End of semester examination worth **40%** of your overall mark. This will include material from the pencil-and-paper part of the course after Easter
- Assessment of course-work worth **60%** of your overall mark:
  - 'Weekly' computing exercises (10%: 3@ 3.33% each)
  - Group computing project (30%)
  - Problem-solving exercises (20%)

The weekly computer exercises will be due on the following dates, all of which are Wednesdays:

17th February (23.45); 24th February; 9th March (23:45).

The group computing project will be due in by no later than 4pm, 18th April.

1.1.5 Assessment

- End of semester examination worth **40%** of your overall mark. This will include material from the pencil-and-paper part of the course after Easter
- Assessment of course-work worth **60%** of your overall mark:
  - 'Weekly' computing exercises (10%: 3@ 3.33% each)
  - Group computing project (30%)
  - Problem-solving exercises (20%)

The weekly computer exercises will be due on the following dates, all of which are Wednesdays:

17th February (23.45); 24th February; 9th March (23:45).

The group computing project will be due in by no later than 4pm, 18th April.

1.1.5 Assessment

- End of semester examination worth **40%** of your overall mark. This will include material from the pencil-and-paper part of the course after Easter
- Assessment of course-work worth **60%** of your overall mark:
  - 'Weekly' computing exercises (10%: 3@ 3.33% each)
  - Group computing project (30%)
  - Problem-solving exercises (20%)

The weekly computer exercises will be due on the following dates, all of which are Wednesdays:

17th February (23.45); 24th February; 9th March (23:45).

The group computing project will be due in by no later than 4pm, 18th April.

1.1.6 Quiz questions

- Last summer, we wrote lots of quiz questions in `Numbas` to help develop your computing skills for this part of the module
- These quiz questions are written within the usual computer-based assessment (CBA) framework – if you're interested, go to:

  `www.mas.ncl.ac.uk/~nlf8/basicR`

- I will make lectures more fun by including quiz questions here, too!

1.1.6 Quiz questions

- Last summer, we wrote lots of quiz questions in `Numbas` to help develop your computing skills for this part of the module
- These quiz questions are written within the usual computer-based assessment (CBA) framework – if you're interested, go to:

  `www.mas.ncl.ac.uk/~nlf8/basicR`

- I will make lectures more fun by including quiz questions here, too!

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?

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?

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?

1.2 What is **R**?

- **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 (better than MatLab!) that can include mathematical symbols
- **R**'s syntax is similar to that used in MatLab
- **R** has loads of built-in functions for statistical analysis, but can also do lots of basic mathematical operations too
- **R** is completely free (unlike MatLab), as is the Rstudio software which we will use in this course
- Named after Ross Ihaka and Robert Gentleman (Auckland University, NZ) who developed the software
- Industry-standard statistical software, and used right throughout your degree

1.2 What is **R**?

- **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 (better than MatLab!) that can include mathematical symbols
- **R**'s syntax is similar to that used in MatLab
- **R** has loads of built-in functions for statistical analysis, but can also do lots of basic mathematical operations too
- **R** is completely free (unlike MatLab), as is the Rstudio software which we will use in this course
- Named after Ross Ihaka and Robert Gentleman (Auckland University, NZ) who developed the software
- Industry-standard statistical software, and used right throughout your degree

1.2 What is **R**?

- **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 (better than MatLab!) that can include mathematical symbols
- **R**'s syntax is similar to that used in MatLab
- **R** has loads of built-in functions for statistical analysis, but can also do lots of basic mathematical operations too
- **R** is completely free (unlike MatLab), as is the Rstudio software which we will use in this course
- Named after Ross Ihaka and Robert Gentleman (Auckland University, NZ) who developed the software
- Industry-standard statistical software, and used right throughout your degree

1.2 What is **R**?

- **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 (better than MatLab!) that can include mathematical symbols
- **R**'s syntax is similar to that used in MatLab
- **R** has loads of built-in functions for statistical analysis, but can also do lots of basic mathematical operations too
- **R** is completely free (unlike MatLab), as is the Rstudio software which we will use in this course
- Named after Ross Ihaka and Robert Gentleman (Auckland University, NZ) who developed the software
- Industry-standard statistical software, and used right throughout your degree

1.2 What is **R**?

- **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 (better than MatLab!) that can include mathematical symbols
- **R**'s syntax is similar to that used in MatLab
- **R** has loads of built-in functions for statistical analysis, but can also do lots of basic mathematical operations too
- **R** is completely free (unlike MatLab), as is the Rstudio software which we will use in this course
- Named after Ross Ihaka and Robert Gentleman (Auckland University, NZ) who developed the software
- Industry-standard statistical software, and used right throughout your degree

1.2 What is **R**?

- **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 (better than MatLab!) that can include mathematical symbols
- **R**'s syntax is similar to that used in MatLab
- **R** has loads of built-in functions for statistical analysis, but can also do lots of basic mathematical operations too
- **R** is completely free (unlike MatLab), as is the Rstudio software which we will use in this course
- Named after Ross Ihaka and Robert Gentleman (Auckland University, NZ) who developed the software
- Industry-standard statistical software, and used right throughout your degree

1.2.1 Accessing **R**

- R is installed on all University machines
- In this class we will be using Rstudio, which is an R IDE
- Both R and Rstudio are free, so you can install them on you own computer. See:

  `http://www.ncl.ac.uk/maths/students/teaching/installingr/`

1.2.1 Accessing **R**

- R is installed on all University machines
- In this class we will be using Rstudio, which is an R IDE
- Both R and Rstudio are free, so you can install them on you own computer. See:

  `http://www.ncl.ac.uk/maths/students/teaching/installingr/`

1.2.1 Accessing **R**

- R is installed on all University machines
- In this class we will be using Rstudio, which is an R IDE
- Both R and Rstudio are free, so you can install them on you own computer. See:

  `http://www.ncl.ac.uk/maths/students/teaching/installingr/`

1.2.2 Previous computing knowledge

- This course is intended to teach you the basics of programming in **R**
- **No** previous programming knowledge is assumed, although MatLab should have helped your general computing skills
- It is crucial that you come to all the practicals for this course
- It is also vital that you let me/other staff know, in the practical sessions, if you're not sure about anything... *Burying your head in the sand will not help. Neither will crying*

1.2.2 Previous computing knowledge

- This course is intended to teach you the basics of programming in **R**
- **No** previous programming knowledge is assumed, although MatLab should have helped your general computing skills
- It is crucial that you come to all the practicals for this course
- It is also vital that you let me/other staff know, in the practical sessions, if you're not sure about anything... *Burying your head in the sand will not help. Neither will crying*

1.2.2 Previous computing knowledge

- This course is intended to teach you the basics of programming in **R**
- **No** previous programming knowledge is assumed, although MatLab should have helped your general computing skills
- It is crucial that you come to all the practicals for this course
- It is also vital that you let me/other staff know, in the practical sessions, if you're not sure about anything... *Burying your head in the sand will not help. Neither will crying*

1.2.2 Previous computing knowledge

- This course is intended to teach you the basics of programming in **R**
- **No** previous programming knowledge is assumed, although MatLab should have helped your general computing skills
- It is crucial that you come to all the practicals for this course
- It is also vital that you let me/other staff know, in the practical sessions, if you're not sure about anything... *Burying your head in the sand will not help. Neither will crying*

1.2.3 Recommended R textbooks

- At `http://goo.gl/nzQTK`, I have constructed a list of suitable books (with comments) on R programming
- Since you will be using R throughout your degree, then it may be worthwhile buying a good R textbook...
- ... however, a really good, free-to-download book, written by Venables and Smith, can be found here:

  `www.cran.r-project.org/doc/manuals/R-intro.pdf`

1.2.3 Recommended R textbooks

- At `http://goo.gl/nzQTK`, I have constructed a list of suitable books (with comments) on R programming
- Since you will be using R throughout your degree, then it may be worthwhile buying a good R textbook...
- ... however, a really good, free-to-download book, written by Venables and Smith, can be found here:

`www.cran.r-project.org/doc/manuals/R-intro.pdf`

1.2.3 Recommended R textbooks

- At `http://goo.gl/nzQTK`, I have constructed a list of suitable books (with comments) on R programming
- Since you will be using R throughout your degree, then it may be worthwhile buying a good R textbook...
- ... however, a really good, free-to-download book, written by Venables and Smith, can be found here:

  `www.cran.r-project.org/doc/manuals/R-intro.pdf`

1.3 Movie dataset

- 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 information about 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.

1.3 Movie dataset

- **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

1.3 Movie dataset

- **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

1.3 Movie dataset

- **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

1.3 Movie dataset

- **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

1.3 Movie dataset

- **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

1.3 Movie dataset

- **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

1.3 Movie dataset

- **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

1.3 Movie dataset

- **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 *Ablaze* 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

1.3 Movie dataset

- **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 *Ablaze* 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

1.3 Movie dataset

- **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 *Ablaze* 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

				Voting statistics					Movie genre							
Title	Year	Length	Budget	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, for multiplication and division...

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

...and more ‘advanced’ operations:

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

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

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

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

...and more ‘advanced’ operations:

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

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

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

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

...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 computer 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`.
- An alternative 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
```

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

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

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
```

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
```

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

```
> rm(x)
```

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

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

```
> 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
```

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

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

```
> is.vector(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
```

```
> is.vector(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
```

```
> is.vector(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
```

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

To determine the length of a 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
```

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

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	`length(x)`	4
Reverse order	`rev(x)`	3,5,5,1
Sort	`sort(x)`	1,3,5,5
Sum	`sum(x)`	14
Extract unique elements	`unique(x)`	1,5,3
Indices of particular elements	`which(x==5)`	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
```

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

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

```
> 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
	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

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
	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE
	FALSE	TRUE	TRUE	FALSE	FALSE	TRUE
	FALSE	FALSE	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
	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE
	FALSE	TRUE	TRUE	FALSE	FALSE	TRUE
	FALSE	FALSE	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
	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE
	FALSE	TRUE	TRUE	FALSE	FALSE	TRUE
	FALSE	FALSE	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
	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE
	FALSE	TRUE	TRUE	FALSE	FALSE	TRUE
	FALSE	FALSE	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
	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE
	FALSE	TRUE	TRUE	FALSE	FALSE	TRUE
	FALSE	FALSE	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
	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE
	FALSE	TRUE	TRUE	FALSE	FALSE	TRUE
	FALSE	FALSE	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
	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE
	FALSE	TRUE	TRUE	FALSE	FALSE	TRUE
	FALSE	FALSE	TRUE	TRUE	FALSE	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
	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
```

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

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
```

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
```

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:

```
> (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

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

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

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"
```

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

```
> 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
```

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
```

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
```

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
```

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

Data Summaries

3.1.1 Measures of average: The sample mean

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

$$\bar{x} = \frac{x_1 + x_2 + \ldots + x_n}{n}$$
$$= \frac{1}{n} \sum_{i=1}^{n} x_i \ .$$

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

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

The mean can be distorted if our sample contains outliers.

3.1.1 Measures of average: The sample mean

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

$$\bar{x} = \frac{x_1 + x_2 + \ldots + x_n}{n} = \frac{1}{n}\sum_{i=1}^{n} x_i \ .$$

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

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

The mean can be distorted if our sample contains outliers.

3.1.1 Measures of average: The sample mean

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

$$\bar{x} = \frac{x_1 + x_2 + \ldots + x_n}{n}$$
$$= \frac{1}{n}\sum_{i=1}^{n} x_i \ .$$

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

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

The mean can be distorted if our sample contains outliers.

3.1.1 Measures of average: The sample mean

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

$$\bar{x} = \frac{x_1 + x_2 + \ldots + x_n}{n}$$
$$= \frac{1}{n}\sum_{i=1}^{n} x_i \ .$$

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

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

The mean can be distorted if our sample contains outliers.

3.1.1 Measures of average: The sample mean

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

$$\bar{x} = \frac{x_1 + x_2 + \ldots + x_n}{n}$$
$$= \frac{1}{n}\sum_{i=1}^{n} x_i \ .$$

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

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

The mean can be distorted if our sample contains outliers.

3.1.1 Measures of average: The sample median

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

3.1.1 Measures of average: The sample median

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

3.1.1 Measures of average: The sample median

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

3.1.1 Measures of average: The 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.

3.1.1 Measures of average: The sample mode

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

3.1.2 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.

3.1.2 Measures of spread: The 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.

3.1.2 Measures of spread: The 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.

3.1.2 Measures of spread: The 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.

3.1.2 Measures of spread: Variance/standard deviation

The sample variance, s^2, is defined as

$$s^2 = $$

$$= $$

The second formula is easier for calculations.

3.1.2 Measures of spread: Variance/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.

3.1.2 Measures of spread: Variance/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.

3.1.2 Measures of spread: Variance/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

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

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.

3.1.2 Measures of spread: Variance/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

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

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.

3.1.2 Measures of spread: Variance/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

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

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.

3.1.2 Measures of spread: Variance/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

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

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.

3.1.2 Measures of spread: Variance/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

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

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.

3.1.2 Measures of spread: Variance/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

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

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.

3.1.2 Measures of spread: Quartiles and the IQR

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.

$$IQR = Q3 - Q1.$$

3.1.2 Measures of spread: Quartiles and the IQR

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.

$$IQR = Q3 - Q1.$$

3.1.3 Example

The number of earth tremours recorded for five randomly chosen locations in Iceland, close to the *Mid-Atlantic ridge*, is recorded below:

Location	1	2	3	4	5
# tremours	7	1	14	13	20

- Calculate the mean and median number of earth tremours for this sample.

  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.

3.1.3 Example

The number of earth tremours recorded for five randomly chosen locations in Iceland, close to the *Mid-Atlantic ridge*, is recorded below:

Location	1	2	3	4	5
# tremours	7	1	14	13	20

- Calculate the mean and median number of earth tremours for this sample.

  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.

3.1.3 Example

- There was a recording error for location 5, which is closest to the Mid-Atlantic Ridge; this observation *should have* been 200, and not 20. Find new values for the mean and median, and comment.

Solution:

The mean is:

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

The median is the middle value, so we still get 13. However, we can clearly see distortion in the mean here.

3.1.3 Example

- There was a recording error for location 5, which is closest to the Mid-Atlantic Ridge; this observation *should have* been 200, and not 20. Find new values for the mean and median, and comment.

  Solution:

  The mean is:

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

  The median is the middle value, so we still get 13. However, we can clearly see distortion in the mean here.

3.1.3 Example

- Find the standard deviation and IQR for the earth tremours data, and comment.

Solution:

We have

$$\sum x^2 = 1^2 + 7^2 + \ldots + 200^2 = 40415.$$

Thus,

$$s = \sqrt{\frac{1}{4}\left\{40415 - 5 \times 47^2\right\}} = 85.69.$$

We also have that $Q1$ is in position 1.5 and $Q3$ position 4.5, giving:

$$Q1 = \frac{1+7}{2} = 4 \quad \text{and} \quad Q3 = \frac{14+200}{2} = 107,$$

giving

$$IQR = 107 = 4 = 103.$$

3.1.3 Example

- Find the standard deviation and IQR for the earth tremours data, and comment.

Solution:

We have

$$\sum x^2 = 1^2 + 7^2 + \ldots + 200^2 = 40415.$$

Thus,

$$s = \sqrt{\frac{1}{4}\left\{40415 - 5 \times 47^2\right\}} = 85.69.$$

We also have that $Q1$ is in position 1.5 and $Q3$ position 4.5, giving:

$$Q1 = \frac{1+7}{2} = 4 \quad \text{and} \quad Q3 = \frac{14+200}{2} = 107,$$

giving

$$IQR = 107 = 4 = 103.$$

3.1.3 Example

- Find the standard deviation and IQR for the earth tremours data, and comment.

Solution:

We have

$$\sum x^2 = 1^2 + 7^2 + \ldots + 200^2 = 40415.$$

Thus,

$$s = \sqrt{\frac{1}{4}\left\{40415 - 5 \times 47^2\right\}} = 85.69.$$

We also have that $Q1$ is in position 1.5 and $Q3$ position 4.5, giving:

$$Q1 = \frac{1+7}{2} = 4 \quad \text{and} \quad Q3 = \frac{14+200}{2} = 107,$$

giving

$$IQR = 107 = 4 = 103.$$

3.2 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.

3.3 Graphical summaries

- Graphical displays of data can be very useful in showing the main features of a dataset
- The appropriate form of graph depends on the nature of the variables being displayed and what aspects are to be shown
- However it should always be borne in mind that the object is to provide a clear and truthful representation of the data, not to distort and not to impress with unnecessary “fancy” features

3.3 Graphical summaries

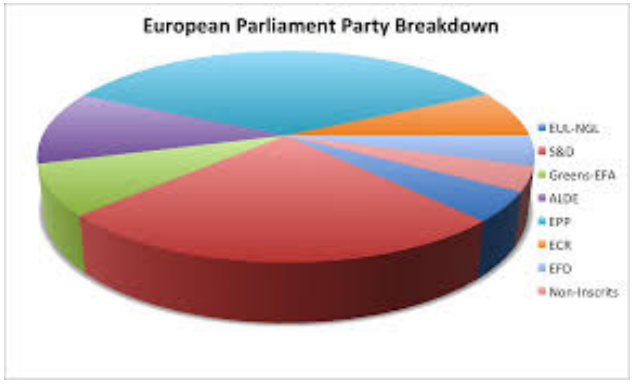
- Graphical displays of data can be very useful in showing the main features of a dataset
- The appropriate form of graph depends on the nature of the variables being displayed and what aspects are to be shown
- However it should always be borne in mind that the object is to provide a clear and truthful representation of the data, not to distort and not to impress with unnecessary “fancy” features

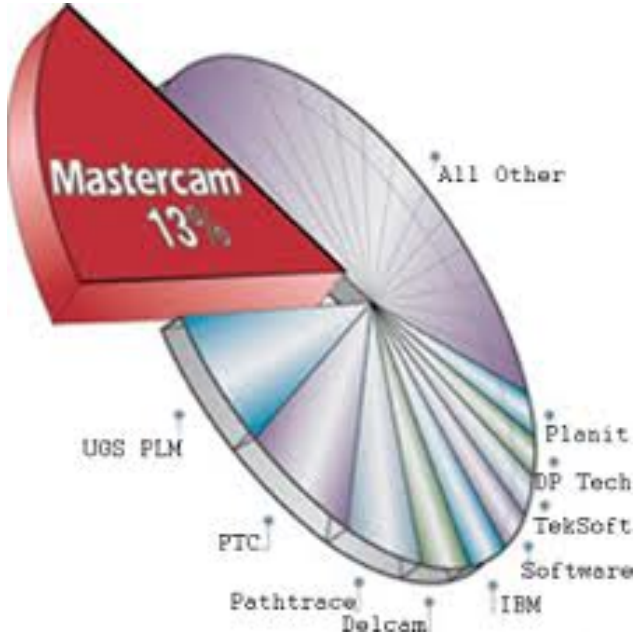
3.3 Graphical summaries

- Graphical displays of data can be very useful in showing the main features of a dataset
- The appropriate form of graph depends on the nature of the variables being displayed and what aspects are to be shown
- However it should always be borne in mind that the object is to provide a clear and truthful representation of the data, not to distort and not to impress with unnecessary “fancy” features

3.3 Graphical summaries

- Graphical displays of data can be very useful in showing the main features of a dataset
- The appropriate form of graph depends on the nature of the variables being displayed and what aspects are to be shown
- However it should always be borne in mind that the object is to provide a clear and truthful representation of the data, not to distort and not to impress with unnecessary “fancy” features

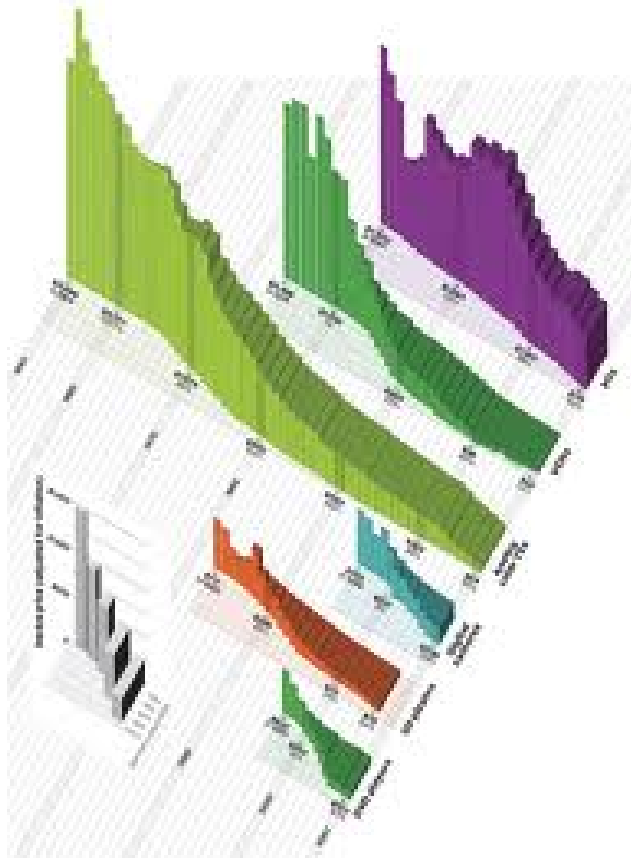




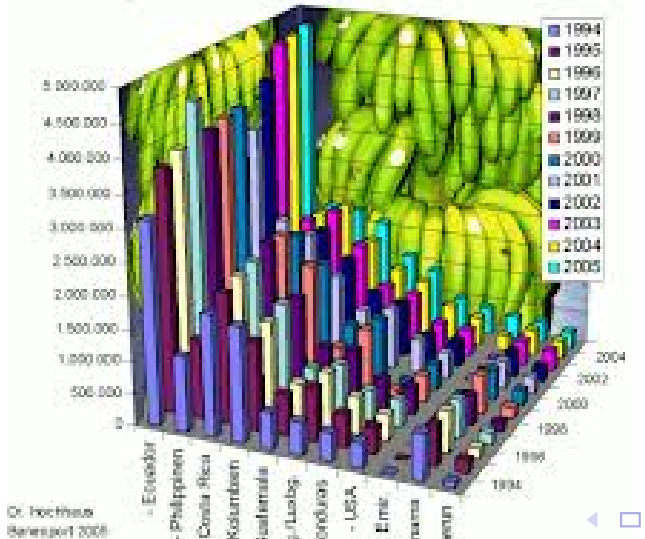


http://www.fusioncharts.com/explore/pie-doughnut-charts



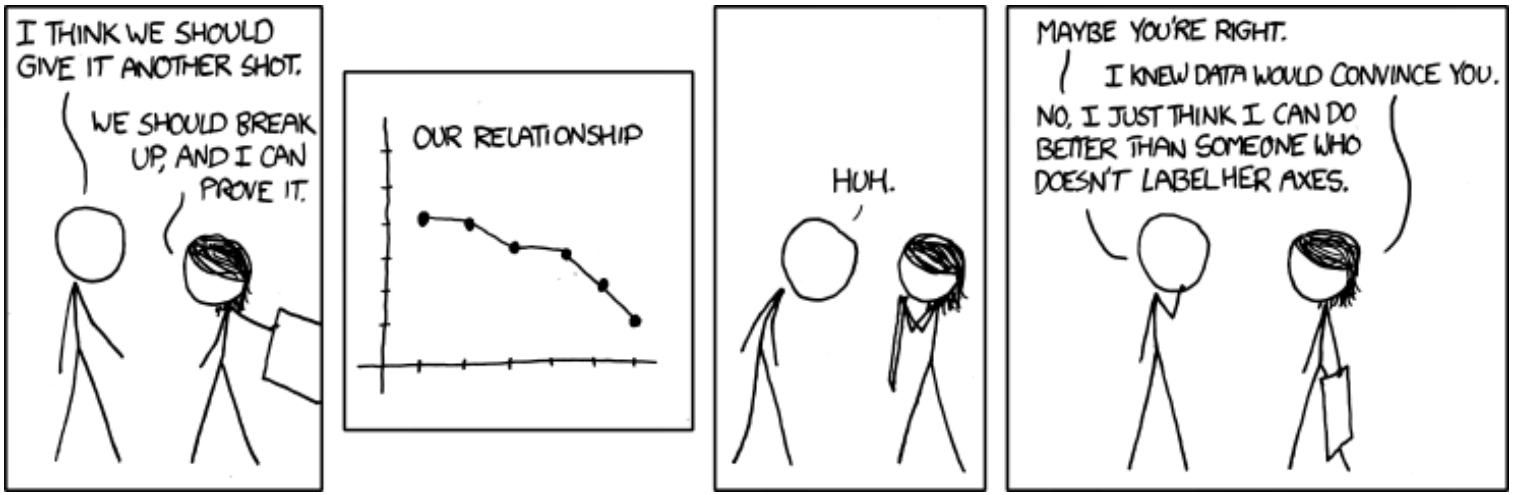


Export von Bananen in Tonnen von 1994-2005



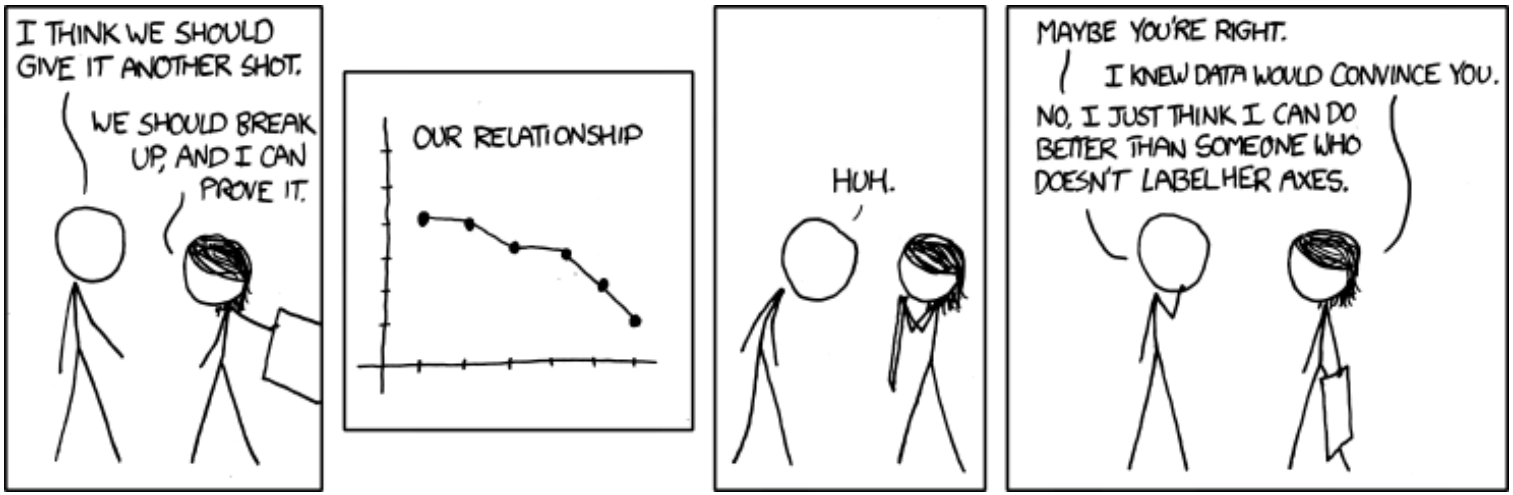
Dr. Hochhaus
Bananenexport 2005

3.3 Label your %#?* axes!



Rant over!

3.3 Label your %#?* axes!



Rant over!

3.3.2 Qualitative data: bar charts

- The most useful way to display qualitative data is usually with a bar chart
- The length of each bar is proportional to the frequency of the corresponding value of the variable in the sample of data
- Note that the widths of the bars should be equal to avoid giving a false impression

3.3.2 MPAA breakdown

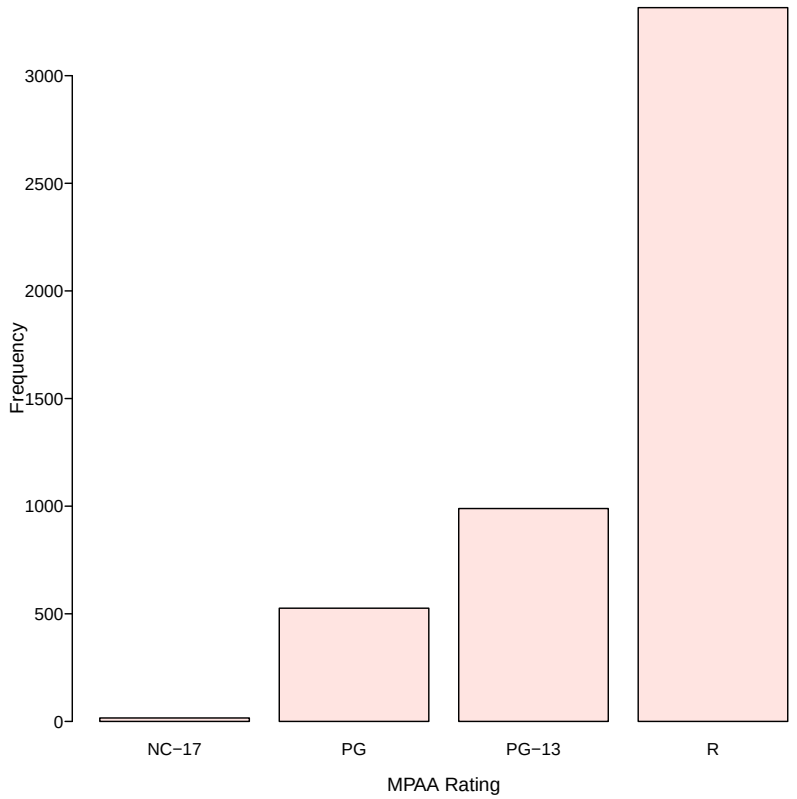


Figure: Barchart of the mpaa ratings for 4847 films.

3.3.2 MPAA breakdown

```
> table(movies$mpaa)
NC-17    PG PG-13     R
   16   526   989  3316
```

Inside the `barplot` function:

```
> barplot(table(movies$mpaa), xlab="MPAA Rating",
+         ylab="Frequency", border = "black",
+         col="mistyrose")
```

Remember to load the data first!

```
> library(mas1343)
> data(movies)
```

3.3.2 MPAA breakdown

```
> table(movies$mpaa)
NC-17     PG PG-13      R
   16    526    989   3316
```

Inside the `barplot` function:

```
> barplot(table(movies$mpaa), xlab="MPAA Rating",
+         ylab="Frequency", border = "black",
+         col="mistyrose")
```

Remember to load the data first!

```
> library(mas1343)
> data(movies)
```

3.3.2 MPAA breakdown

```
> table(movies$mpaa)
NC-17    PG PG-13     R
   16   526   989  3316
```

Inside the `barplot` function:

```
> barplot(table(movies$mpaa), xlab="MPAA Rating",
+         ylab="Frequency", border = "black",
+         col="mistyrose")
```

Remember to load the data first!

```
> library(mas1343)
> data(movies)
```

3.3.3 Histograms

- To represent the distribution of a sample of values of a continuous variable we can use a histogram
- The range of values of the variable is divided into intervals, known as *classes* or bins, and the frequencies in classes are represented by columns
- As the variable is continuous, there are no gaps between neighbouring columns — unlike a bar chart
- Note also that, strictly speaking, it is the *area* of the column which is proportional to the frequency, not the height
- The reason for this is that columns need not be of the same width
- Computer software tends to use columns of the same width, meaning that we can just use frequencies on thr y-axis (giving “frequency histograms”)
- However this default can be overridden in R if you really want to

3.3.3 Histograms

- To represent the distribution of a sample of values of a continuous variable we can use a histogram
- The range of values of the variable is divided into intervals, known as *classes* or bins, and the frequencies in classes are represented by columns
- As the variable is continuous, there are no gaps between neighbouring columns — unlike a bar chart
- Note also that, strictly speaking, it is the *area* of the column which is proportional to the frequency, not the height
- The reason for this is that columns need not be of the same width
- Computer software tends to use columns of the same width, meaning that we can just use frequencies on thr y-axis (giving "frequency histograms")
- However this default can be overridden in R if you really want to

3.3.3 Histograms

- To represent the distribution of a sample of values of a continuous variable we can use a histogram
- The range of values of the variable is divided into intervals, known as *classes* or bins, and the frequencies in classes are represented by columns
- As the variable is continuous, there are no gaps between neighbouring columns — unlike a bar chart
- Note also that, strictly speaking, it is the *area* of the column which is proportional to the frequency, not the height
- The reason for this is that columns need not be of the same width
- Computer software tends to use columns of the same width, meaning that we can just use frequencies on thr y-axis (giving “frequency histograms”)
- However this default can be overridden in R if you really want to

3.3.3 Histograms

- To represent the distribution of a sample of values of a continuous variable we can use a histogram
- The range of values of the variable is divided into intervals, known as *classes* or bins, and the frequencies in classes are represented by columns
- As the variable is continuous, there are no gaps between neighbouring columns — unlike a bar chart
- Note also that, strictly speaking, it is the *area* of the column which is proportional to the frequency, not the height
- The reason for this is that columns need not be of the same width
- Computer software tends to use columns of the same width, meaning that we can just use frequencies on thr y-axis (giving “frequency histograms”)
- However this default can be overridden in R if you really want to

3.3.3 Histograms

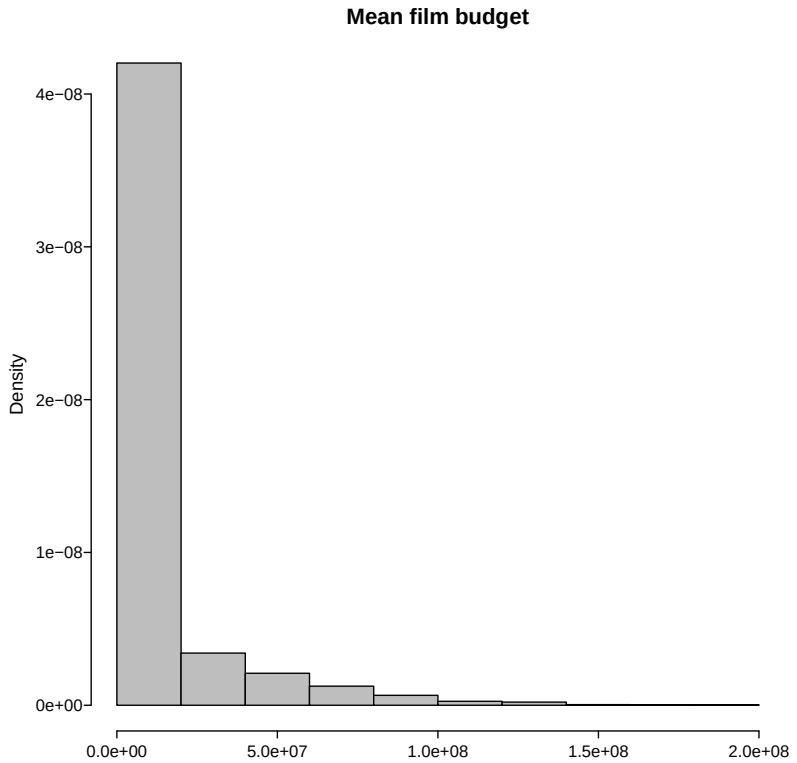


Figure: Histogram of movie budgets.

3.3.3 Histograms

- Figure 3.2 shows a histogram of the movie budgets
- When dealing with *relative* frequency (proportions, often termed "density histograms"), we can easily work out the height of each bin using this formula:

$$\text{Height} = \frac{\text{frequency}}{n \times \text{ Bin-width}}$$

- When the y-axis is labelled with density or relative frequencies, the area under the histogram is one
- Bin widths should be chosen so that you get a good idea of the distribution of the data, without being swamped by random variation

3.3.3 Histograms

- Figure 3.2 shows a histogram of the movie budgets
- When dealing with *relative* frequency (proportions, often termed "density histograms"), we can easily work out the height of each bin using this formula:

$$\text{Height} = \frac{\text{frequency}}{n \times \text{ Bin-width}}$$

- When the y-axis is labelled with density or relative frequencies, the area under the histogram is one
- Bin widths should be chosen so that you get a good idea of the distribution of the data, without being swamped by random variation

3.3.3 Histograms

To generate Figure 3.2 in R we use the following commands:

```
> hist(movies$Budget, col="grey",
+      main="Mean film budget", freq=FALSE,
+      xlab="Budget ($)")
```

3.3.3 How many bins should we have?

First we will define the notation we will use:

- n: the sample size;
- k: the number of bins in the histogram;
- h: the bin-width.

Then the number of bins we will use to construct a histogram is:

$$k = \left\lceil \frac{\max(x) - \min(x)}{h} \right\rceil \tag{3.1}$$

where $\lceil \cdot \rceil$ is the ceiling function.

Table 3.2: Sturges' rule

- $k_{ST} = \lceil \log_2 n + 1 \rceil$
- Default
- Tends not to be very good for $n > 30$.

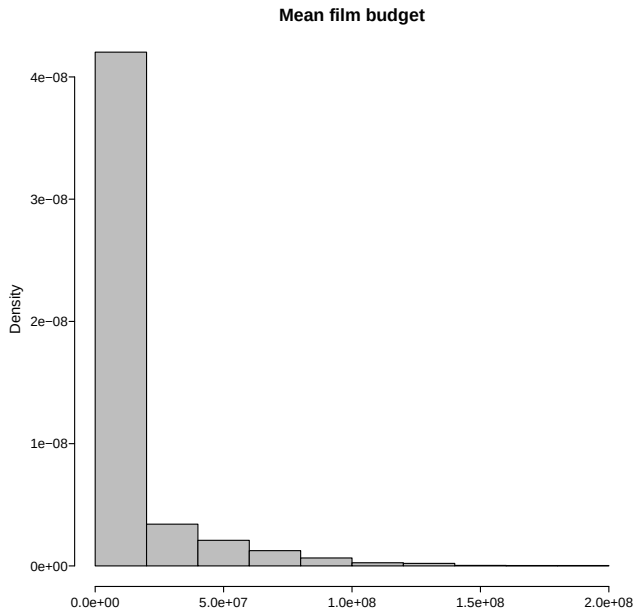


Table 3.2: Scotts' rule

- $h_{SC} = 3.49 \times s \times n^{-1/3}$
- breaks = “Scott"

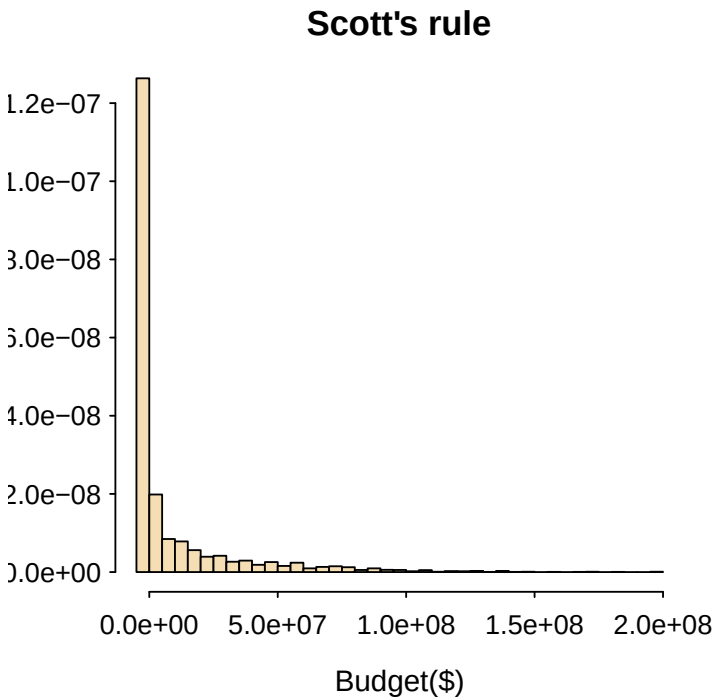
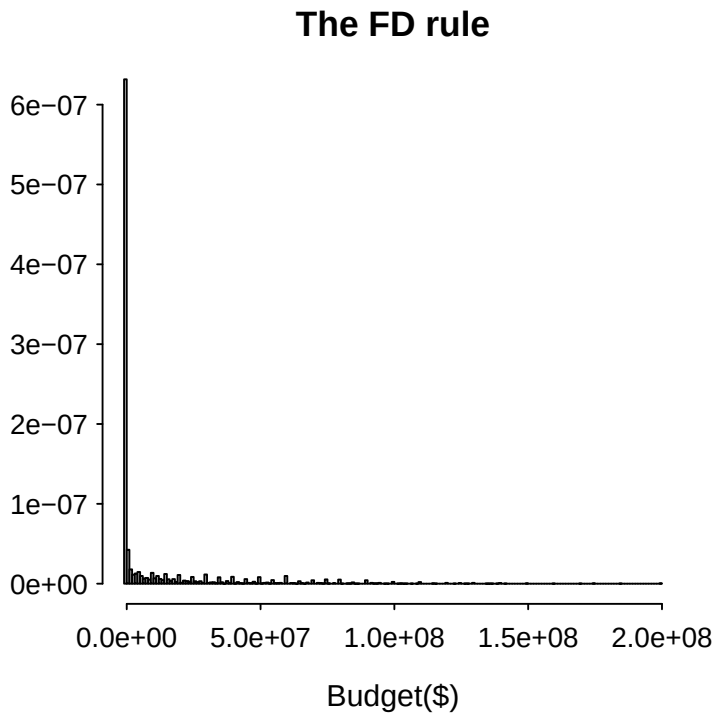


Table 3.2: Freedman-Diaconis' rule

- $h_{FR} = 2 \times IQR(x) \times n^{-1/3}$
- breaks = "FD"
- When the distribution is symmetric, this is very similar to Scott's rule.



3.3.4 Box and whisker plots

A box and whisker plot, often referred to simply as a boxplot, is another way to represent continuous data.

This kind of plot is particularly useful for comparing two or more groups, by placing the boxplots side-by-side.

Figures 3.4 and 3.5 shows boxplots of film length for different categories of film

- Central bar in the box: median
- Top/bottom of the box: Q3/Q1
- Whisker: smallest value within $1.5IQR$ of $Q1$ and largest value within $1.5IQR$ of $Q3$
- outliers: asterisks - points outside the whiskers

3.3.4 Box and whisker plots

A box and whisker plot, often referred to simply as a boxplot, is another way to represent continuous data.

This kind of plot is particularly useful for comparing two or more groups, by placing the boxplots side-by-side.

Figures 3.4 and 3.5 shows boxplots of film length for different categories of film

- Central bar in the box: median
- Top/bottom of the box: Q3/Q1
- Whisker: smallest value within $1.5IQR$ of $Q1$ and largest value within $1.5IQR$ of $Q3$
- outliers: asterisks - points outside the whiskers

3.3.4 Box and whisker plots

A box and whisker plot, often referred to simply as a boxplot, is another way to represent continuous data.

This kind of plot is particularly useful for comparing two or more groups, by placing the boxplots side-by-side.

Figures 3.4 and 3.5 shows boxplots of film length for different categories of film

- Central bar in the box: median
- Top/bottom of the box: Q3/Q1
- Whisker: smallest value within $1.5IQR$ of $Q1$ and largest value within $1.5IQR$ of $Q3$
- outliers: asterisks - points outside the whiskers

3.3.4 Box and whisker plots

A box and whisker plot, often referred to simply as a boxplot, is another way to represent continuous data.

This kind of plot is particularly useful for comparing two or more groups, by placing the boxplots side-by-side.

Figures 3.4 and 3.5 shows boxplots of film length for different categories of film

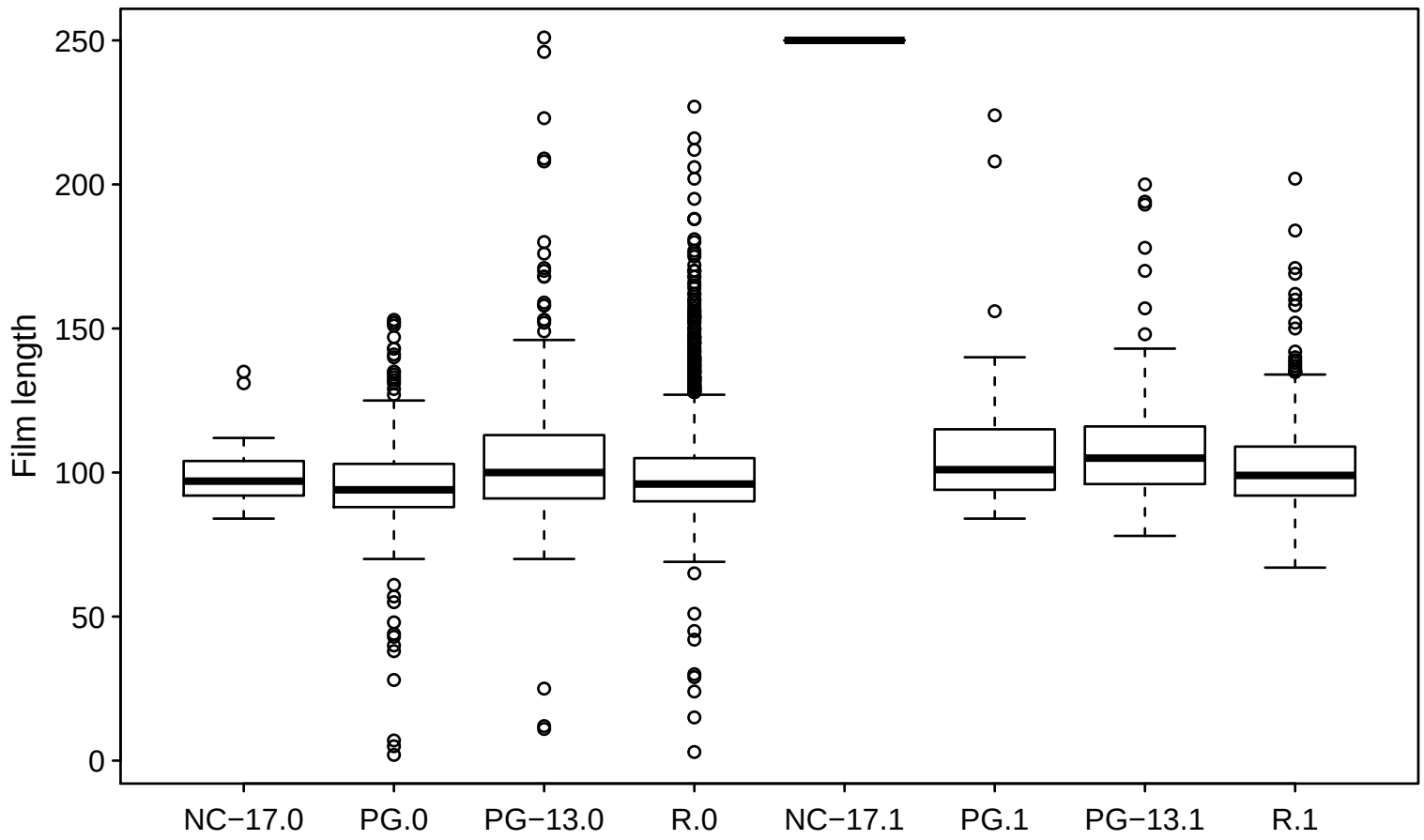
- Central bar in the box: median
- Top/bottom of the box: Q3/Q1
- Whisker: smallest value within $1.5IQR$ of $Q1$ and largest value within $1.5IQR$ of $Q3$
- outliers: asterisks - points outside the whiskers

3.3.4 Box and whisker plots

To do this in R we use the following commands:

```
> par(mfrow=c(1, 2))
> boxplot(movies$Length, ylab="Film length",
+         col="bisque")
> boxplot(movies$Length ~ movies$mpaa,
+         ylab="Film length",
+         col="bisque")
```

3.3.4 Box and whisker plots




```
> boxplot(movies$Length ~ movies$mpaa + movies$Romance)
```

Part IV

Control Statements and Functions

4.1.1 Basic functions

This function takes in a single argument x and returns x^2:

```
> Fun1 = function(x) {
+   return (x*x)
+ }
```

4.1.1 Basic functions

The key elements in the function call are:

- The word `function`;
- The brackets `()` which enclose the **argument** list. This list may be empty.
- A sequence of statements in curly braces `{ }`.
- A `return` statement.

4.1.1 Basic functions

The key elements in the function call are:

- The word `function`;
- The brackets $()$ which enclose the **argument** list. This list may be empty.
- A sequence of statements in curly braces $\{\ \}$.
- A `return` statement.

4.1.1 Basic functions

The key elements in the function call are:

- The word `function`;
- The brackets $()$ which enclose the **argument** list. This list may be empty.
- A sequence of statements in curly braces $\{\ \}$.
- A `return` statement.

4.1.1 Basic functions

The key elements in the function call are:

- The word `function`;
- The brackets $()$ which enclose the **argument** list. This list may be empty.
- A sequence of statements in curly braces $\{\ \}$.
- A `return` statement.

4.1.1 Basic functions

We ‘call’ Fun1 in the following manner:

```
> Fun1(5)
[1] 25
```

```
> y = Fun1(10)
```

```
> y
[1] 100
```

```
> z = c(1, 2, 3, 4)
> Fun1(z)
[1]  1  4  9 16
```

4.1.1 Basic functions

We ‘call’ Fun1 in the following manner:

```
> Fun1(5)
[1] 25
```

```
> y = Fun1(10)
```

```
> y
[1] 100
```

```
> z = c(1, 2, 3, 4)
> Fun1(z)
[1]  1  4  9 16
```

4.1.1 Basic functions

We ‘call’ Fun1 in the following manner:

```
> Fun1(5)
[1] 25
```

```
> y = Fun1(10)
```

```
> y
[1] 100
```

```
> z = c(1, 2, 3, 4)
> Fun1(z)
[1]  1  4  9 16
```

4.1.1 Basic functions

We ‘call’ Fun1 in the following manner:

```
> Fun1(5)
[1] 25
```

```
> y = Fun1(10)
```

```
> y
[1] 100
```

```
> z = c(1, 2, 3, 4)
> Fun1(z)
[1]  1  4  9 16
```

4.1.1 Basic functions

Of course, the old saying ‘Garbage in, garbage out’ is true:

```
> Fun1()
Error in Fun1() :  argument "x" is missing, no default
```

```
> Fun1("5")
Error in x * x :  non-numeric argument to binary
operator
```

Question 1

What is the value of y?

```
> fun1 = function(x) {
+          return(2*x)
+ }
> y = fun1(2)
```

- 5
- 10
- 2
- 4

Question 1

What is the value of y?

```
> x = 5
> fun1 = function(x) {
+          return(2*x)
+ }
> (y= fun1(2))
[1] 4
```

- 5
- 10
- 2
- 4

Question 2

What is the value of y?

```
> fun1 = function(x) {
+    z = 3
+    return(x*z)
+ }
> y = fun1(2)
```

- 2
- 3
- 4
- 6

Question 2

What is the value of y?

```
> fun1 = function(x) {
+   z = 3
+   return(x*z)
+ }
> (y = fun1(2))
[1] 6
```

- 2
- 3
- 4
- 6

Question 3

What is the value of y?

```
> fun1 = function(x) {
+    z = 3
+    return(x*z)
+ }
> k = 5
> y = fun1(k)
```

Question 3

What is the value of y?

```
> fun1 = function(x) {
+   z = 3
+   return(x*z)
+ }
> k = 5
> (y = fun1(k))
[1] 15
```

4.1.1 Basic functions

Other variations to this simple function are:

```
> Fun2 = function(x=1) {
+   return (x*x)
+ }
```

```
> Fun2()
[1] 1
```

```
> Fun2(4)
[1] 16
```

4.1.1 Basic functions

Other variations to this simple function are:

```
> Fun2 = function(x=1) {
+   return (x*x)
+ }
```

```
> Fun2()
[1] 1
```

```
> Fun2(4)
[1] 16
```

4.1.1 Basic functions

Other variations to this simple function are:

```
> Fun2 = function(x=1) {
+   return (x*x)
+ }
```

```
> Fun2()
[1] 1
```

```
> Fun2(4)
[1] 16
```

4.1.1 Basic functions

```
> Fun3 = function(x, y) {
+    return (x*y)
+ }
```

```
> Fun3(3, 4)
[1] 12
```

4.1.1 Basic functions

```
> Fun3 = function(x, y) {
+    return (x*y)
+ }
```

```
> Fun3(3, 4)
[1] 12
```

Question 4

What is the value of y?

```
> fun1 = function(x=3) {
+          return(2*x)
+ }
> y = fun1(2)
```

- 3
- 4
- 6

Question 4

What is the value of y?

```
> fun1 = function(x=3) {
+          return(2*x)
+ }
> (y = fun1(2))
[1] 4
```

- 3
- 4
- 6

Question 5

What is the value of y?

```
> fun1 = function(x=3) {
+   return(2*x)
+ }
> y = fun1()
```

- 3
- 4
- 6

Question 5

What is the value of y?

```
> fun1 = function(x=3) {
+   return(2*x)
+ }
> (y = fun1())
[1] 6
```

- 3
- 4
- 6

Question 6

What is the value of y?

```
> fun1 = function(x=3) {
+    x = 2
+    return(2*x)
+ }
> y = fun1()
```

- 3
- 4
- 6

Question 6

What is the value of y?

```
> fun1 = function(x=3) {
+   x = 2
+   return(2*x)
+ }
> (y = fun1())
[1] 4
```

- 3
- 4
- 6

4.1.2 A more useful function

Here the function below takes in a vector, plots a histogram and returns a vector containing the mean and standard deviation:

```
> Investigate = function(values) {
+   hist(values)
+   m_std = c(mean(values), sd(values))
+   return(m_std)
+ }
```

4.1.2 A more useful function

Once we have created our function, we can put it to good use:

```
> Investigate(movies$Rating)
[1] 5.522715 1.451864
```

```
> Investigate(movies$Length)
[1] 100.87807  17.34152
```

```
> Investigate(movies$Budget)
[1] 10286893 23039711
```

4.1.2 A more useful function

Once we have created our function, we can put it to good use:

```
> Investigate(movies$Rating)
[1] 5.522715 1.451864
```

```
> Investigate(movies$Length)
[1] 100.87807  17.34152
```

```
> Investigate(movies$Budget)
[1] 10286893 23039711
```

4.1.2 A more useful function

Once we have created our function, we can put it to good use:

```
> Investigate(movies$Rating)
[1] 5.522715 1.451864
```

```
> Investigate(movies$Length)
[1] 100.87807  17.34152
```

```
> Investigate(movies$Budget)
[1] 10286893 23039711
```

4.1.3 Variable scope

4.1.3 Variable scope

- When we call a function, R first looks for *local* variables, then *global* variables.
- For example, Fun4 uses a global variable:

```
> blob = 5
> Fun4 = function() {
+   return(blob)
+ }
```

```
> Fun4()
[1] 5
```

4.1.3 Variable scope

- When we call a function, R first looks for *local* variables, then *global* variables.
- For example, Fun4 uses a global variable:

```
> blob = 5
> Fun4 = function() {
+   return(blob)
+ }
```

```
> Fun4()
[1] 5
```

4.1.3 Variable scope

But Fun5 uses a local variable:

```
> blob = 5
> Fun5 = function() {
+   blob = 6
+   return(blob)
+ }
```

```
> Fun5()
[1] 6
```

```
> blob
[1] 5
```

4.1.3 Variable scope

But Fun5 uses a local variable:

```
> blob = 5
> Fun5 = function() {
+   blob = 6
+   return(blob)
+ }
```

```
> Fun5()
[1] 6
```

```
> blob
[1] 5
```

4.1.3 Variable scope

But Fun5 uses a local variable:

```
> blob = 5
> Fun5 = function() {
+   blob = 6
+   return(blob)
+ }
```

```
> Fun5()
[1] 6
```

```
> blob
[1] 5
```

Question 7

What is the value of y?

```
> x = 1
> fun1 = function(x) {
+   x = 2
+   return(x)
+ }
> y= fun1(3)
```

- 1
- 2
- 3

Question 7

What is the value of y?

```
> x = 1
> fun1 = function(x) {
+   x = 2
+   return(x)
+ }
> (y= fun1(3))
[1] 2
```

- 1
- 2
- 3

Question 8

What is the value of y?

```
> x = 1
> fun1 = function(x=3) {
+    x = 2
+    return(x)
+ }
> y= fun1()
```

- 1
- 2
- 3

Question 8

What is the value of y?

```
> x = 1
> fun1 = function(x=3) {
+   x = 2
+   return(x)
+ }
> (y= fun1())
[1] 2
```

- 1
- 2
- 3

Question 9

What is the value of y?

```
> x = 1
> fun1 = function(x=3) {
+   return(2*x)
+ }
> x = fun1(x)
> y = fun1(x)
```

Question 9

What is the value of y?

```
> x = 1
> fun1 = function(x=3) {
+   return(2*x)
+ }
> x = fun1(x)
> (y = fun1(x))
[1] 4
```

4.2 Conditionals

- Conditional statements are features of a programming language which perform different computations or actions depending on whether a condition evaluates to TRUE or FALSE.
- They are used in almost all computer programs.

4.2.1 `If` statements

The basic structure of an `if` statement is:

```
> if(expr) {
+   ##do something
+ }
```

where `expr` is evaluated to be either `TRUE` or `FALSE`.

4.2.1 `If` statements

The following example illustrates `if` statements in R:

```
> x = 5
> y = 5
> if(x<5) {
+   y = 0
+ }
```

```
> y
[1] 5
```

In this code chunk, x $<$ 5 evaluates to be FALSE so the following brackets are not evaluated.

4.2.1 `If` statements

The following example illustrates `if` statements in R:

```
> x = 5
> y = 5
> if(x<5) {
+   y = 0
+ }
```

```
> y
[1] 5
```

In this code chunk, x $<$ 5 evaluates to be FALSE so the following brackets are not evaluated.

4.2.1 If statements

We test for greater than in a similar manner:

```
> x = 5
> y = 5
> if(x > 0) {
+   y = 0
+ }
> y
[1] 0
```

Here x $>$ 0 evaluates to be TRUE so, y is set equal to 0. If we wanted to test for equality with zero, then we would use $=>$.

4.3.2 `If else` statements

We can link together a number of `if` statements:

```
> x = 0
> if(x > 0) {
+   cat("x is greater than zero")
+ } else if(x < 0) {
+   cat("x is less than zero")
+ } else {
+   cat("x must be zero!")
+   cat("\n")
+ }
x must be zero!
```

The final `else` is optional.

4.2.2 If else statements

```
> IsNegative = function(value) {
+   is_neg = FALSE
+   if(value < 0) {
+     is_neg = TRUE
+   }
+   return(is_neg)
+ }
```

```
> IsNegative(1)
[1] FALSE
```

```
> IsNegative(-4.6)
[1] TRUE
```

4.2.2 If else statements

```
> IsNegative = function(value) {
+   is_neg = FALSE
+   if(value < 0) {
+     is_neg = TRUE
+   }
+   return(is_neg)
+ }
```

```
> IsNegative(1)
[1] FALSE
```

```
> IsNegative(-4.6)
[1] TRUE
```

4.2.2 If else statements

```
> IsNegative = function(value) {
+   is_neg = FALSE
+   if(value < 0) {
+     is_neg = TRUE
+   }
+   return(is_neg)
+ }
```

```
> IsNegative(1)
[1] FALSE
```

```
> IsNegative(-4.6)
[1] TRUE
```

4.2.2 If else statements

A more sophisticated function could be:

```
> IsGreaterThan = function(value1, value2) {
+   is_greater_than = FALSE
+   if(value1 > value2) {
+     is_greater_than = TRUE
+   }
+   return(is_greater_than)
+ }
```

4.2.2 If else statements

```
> IsGreaterThan(-5, -6)
[1] TRUE
```

```
> IsGreaterThan(10, 10)
[1] FALSE
```

4.2.2 If else statements

```
> IsGreaterThan(-5, -6)
[1] TRUE
```

```
> IsGreaterThan(10, 10)
[1] FALSE
```

Question 1

What is the value of y?

```
> x = 5
> if(x==6)  {
+    y = FALSE
+ } else {
+    y = TRUE
+ }
```

- TRUE
- FALSE

Question 1

What is the value of y?

```
> x = 5
> if(x==6)  {
+    y = FALSE
+ } else {
+    y = TRUE
+ }
> y
[1] TRUE
```

- TRUE
- FALSE

Question 2

What is the value of y?

```
> x = 5
> if(x > 6){
+    y = 0
+ } else if(x >= 5) {
+    y = 1
+ } else {
+ y = 2
+ }
```

- 0
- 1
- 2

Question 2

What is the value of y?

```
> x = 5
> if(x > 6){
+    y = 0
+ }else if(x >= 5) {
+    y = 1
+ } else {
+   y = 2
+ }
> y
[1] 1
```

- 0
- 1
- 2

Question 3

What is the value of z?

```
> x = 5
> y = "male"
> if(x > 6 & y == "Female"){
+   z = 0
+ }else if(x < 5 & y == "Male") {
+   z = 1
+ } else {
+   z = 2
+ }
```

- 0
- 1
- 2

Question 3

What is the value of z?

```
> x = 5
> y = "male"
> if(x > 6 & y == "Female"){
+   z = 0
+ }else if(x < 5 & y == "Male") {
+   z = 1
+ } else {
+   z = 2
+ }
> z
[1] 2
```

- 0
- 1
- 2

4.3 Control statements

- At times we would like to perform some operation on a vector or a data frame.
- Often R has built-in functions that will do this for you, e.g. `mean`, `sd`, other times we have to write our own functions.
- For example, suppose we want to calculate $\sum_{i=1}^{10} i^2$.
- In R we can use a `for` loop:

```
> x = 0
> for(i in 1:10) {
+   x = x + i^2
+ }
```

```
> x
[1] 385
```

4.3 Control statements

- At times we would like to perform some operation on a vector or a data frame.
- Often R has built-in functions that will do this for you, e.g. `mean`, `sd`, other times we have to write our own functions.
- For example, suppose we want to calculate $\sum_{i=1}^{10} i^2$.
- In R we can use a `for` loop:

```
> x = 0
> for(i in 1:10) {
+   x = x + i^2
+ }
```

```
> x
[1] 385
```

4.3 Control statements

$\sum_{j=-5}^{-1} e^j / j^2$,

```
> total = 0
> for(j in -5:-1) {
+   total = total + exp(j)/j^2
+ }
```

```
> total
[1] 0.4086594
```

4.3 Control statements

$\sum_{j=-5}^{-1} e^j / j^2$,

```
> total = 0
> for(j in -5:-1) {
+   total = total + exp(j)/j^2
+ }
```

```
> total
[1] 0.4086594
```

4.3 Control statements

A more tricky example is: calculate $\sum e^k / k^2$, for $k = 3, 6, 9, \ldots, 21$:

```
> total = 0
> for(i in 1:7) {
+   k = i*3
+   total = total + exp(k)/k^2
+ }
> total
[1] 3208939
```

Question 4

What is the value of x?

```
> total = 0
> for(blob in 1:4) {
+   total = total + 1
+ }
> x = total
```

Question 4

What is the value of x?

```
> total = 0
> for(blob in 1:4) {
+   total = total + 1
+ }
> (x = total)
[1] 4
```

Question 5

What is the value of x?

```
> total = 0
> for(blob in 1:10) {
+   if(blob > 9) {
+     total = total + 1
+   }
+ }
> x = total
```

Question 5

What is the value of x?

```
> total = 0
> for(blob in 1:10) {
+   if(blob > 9) {
+     total = total + 1
+   }
+ }
> (x = total)
[1] 1
```

4.4 Putting it all together

Rather than have to constantly write R code to solve the summations in §4.3 we can create a function to solve the general form:

$$\sum_{i=i_s}^{i_e} \frac{e^i}{i^2} \quad \text{for } i = i_s,\ i_s + j,\ i_s + 2j,\ \ldots,\ i_e\ .$$

4.4 Putting it all together

So in R we have:

```
> Summation1 = function(i_s, i_e, j) {
+   total = 0
+   for(i in i_s:(i_e/j)) {
+     k = i*j
+     total = total + exp(k)/k^2
+   }
+   return(total)
+ }
```

```
> Summation1(-5, -1, 1)
[1] 0.4086594
```

```
> Summation1(3, 21, 3)
[1] 3208925
```

4.4 Putting it all together

So in R we have:

```
> Summation1 = function(i_s, i_e, j) {
+   total = 0
+   for(i in i_s:(i_e/j)) {
+     k = i*j
+     total = total + exp(k)/k^2
+   }
+   return(total)
+ }
```

```
> Summation1(-5, -1, 1)
[1] 0.4086594
```

```
> Summation1(3, 21, 3)
[1] 3208925
```

4.4 Putting it all together

So in R we have:

```
> Summation1 = function(i_s, i_e, j) {
+   total = 0
+   for(i in i_s:(i_e/j)) {
+     k = i*j
+     total = total + exp(k)/k^2
+   }
+   return(total)
+ }
```

```
> Summation1(-5, -1, 1)
[1] 0.4086594
```

```
> Summation1(3, 21, 3)
[1] 3208925
```

4.5.1 The apply function

- We use the apply function when we want to *apply* the same function to every row or column of a data frame.
- For example, suppose we have a data frame with three columns:

```
> (df4 = data.frame(c1 = 1:4, c2 = 4:7, c3 = 2:5))
  c1 c2 c3
1  1  4  2
2  2  5  3
3  3  6  4
4  4  7  5
```

4.5.1 The apply function

- The apply function takes (at least) three arguments.
- The first argument is the data frame, the second the number 1 or 2 indicating row or column and the third a function to apply to each row or column.
- So

```
> apply(df4, 1, mean)
[1] 2.333333 3.333333 4.333333 5.333333
```

calculates the mean value of every row.

Standard deviation of every column

```
> apply(df4, 2, sd)
       c1        c2        c3
1.290994 1.290994 1.290994
```

Suppose one of the columns was non-numeric:

```
> (df5 = data.frame(c1 = 1:3, c2 = 4:6,
+     c3 = LETTERS[1:3]))
  c1 c2 c3
1  1  4  A
2  2  5  B
3  3  6  C
```

Then taking the mean doesn't really make sense:

```
> apply(df5, 1, mean)
[1] NA NA NA
```

Instead, we remove the column, then calculate the mean:

```
> apply(df5[ ,1:2], 1, mean)
[1] 2.5 3.5 4.5
```

Suppose one of the columns was non-numeric:

```
> (df5 = data.frame(c1 = 1:3, c2 = 4:6,
+     c3 = LETTERS[1:3]))
  c1 c2 c3
1  1  4  A
2  2  5  B
3  3  6  C
```

Then taking the mean doesn't really make sense:

```
> apply(df5, 1, mean)
[1] NA NA NA
```

Instead, we remove the column, then calculate the mean:

```
> apply(df5[ ,1:2], 1, mean)
[1] 2.5 3.5 4.5
```

Question 6

What type of variable is z?

```
> apply(z, 1, mean)
```

- Data frame
- Double
- String
- logical

Question 6

What type of variable is z?

```
> apply(z, 1, mean)
```

- Data frame
- Double
- String
- logical

Question 7

Which of these statements is correct?

```
> apply(z, 2, median)
```

- We calculate the median of z
- We calculate the median for each row of z
- We calculate the median for each column of z
- We calculate the mean of z

Question 7

Which of these statements is correct?

```
> apply(z, 2, median)
```

- We calculate the median of z
- We calculate the median for each row of z
- We calculate the median for each column of z
- We calculate the mean of z

4.5.2 The `tapply` function

- The function `tapply` is very useful, but at first glance can be tricky to understand.
- It's best described using an example:

```
> tapply(movies$Length, movies$mpaa, mean)
    NC-17         PG      PG-13          R
110.18750  97.23384 104.97877 100.18818
```

4.5.2 The `tapply` function

- In the above code, we have calculated the average movie length conditional on its MPAA rating.
- So the average length of a PG movie is 97 minutes and the average NC-17 movie length is 110mins.
- With `tapply` we can do we very interesting things.

4.5.2 The `tapply` function

- For example, in the next piece of code, we plot the average movie rating conditional on its length:

```
> tapply(movies$Length, movies$Rating, mean)[1:6]
   1  1.2  1.3  1.4  1.5  1.6
85.5 93.0 87.5 85.0 67.0 86.0
```

```
> rating_by_len = tapply(movies$Length, movies$Rating,
+   mean)
> plot(names(rating_by_len), rating_by_len)
```

4.5.2 The `tapply` function

- For example, in the next piece of code, we plot the average movie rating conditional on its length:

```
> tapply(movies$Length, movies$Rating, mean)[1:6]
   1  1.2  1.3  1.4  1.5  1.6
85.5 93.0 87.5 85.0 67.0 86.0
```

```
> rating_by_len = tapply(movies$Length, movies$Rating,
+   mean)
> plot(names(rating_by_len), rating_by_len)
```

4.5.2 The `tapply` function

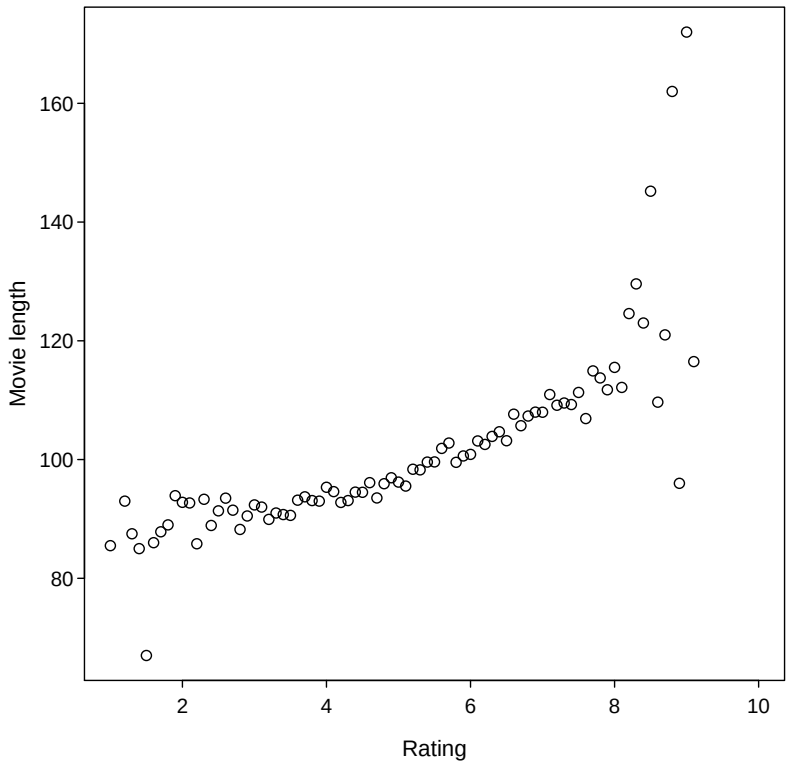


Figure: Plot of movie length conditional on it's rating.

4.6 Help

- R has a very good help system.
- If you need information about a particular function, say `plot`.
- Then typing `?plot` in a R terminal will bring up the associated help page.
- The internet is another very good source of R help.
- Unfortunately, using Google isn't particularly useful since the letter "R" appears on most web pages!
- However, you can use

  `http://www.rseek.org/`

  Using this search engine limits searches to R web-pages.

Part V

Random Number Generation

5.1 Randomness: quantifying uncertainty

- The concepts of uncertainty and randomness have intrigued humanity for a long time.
- The world around us is not deterministic and we are faced continually with chance occurrences.
- Uncertainty is inherent in nature; for example, the behaviour of fundamental physical particles, genes and chromosomes in biology, and individuals in society under stress or strain.
- The methodology for exploring uncertainty involves the use of *random numbers*.

5.2 Pseudo–random numbers

Suppose we need to obtain a list of random digits $0, 1, 2, \ldots, 9$. How might we go about this? There are several options:

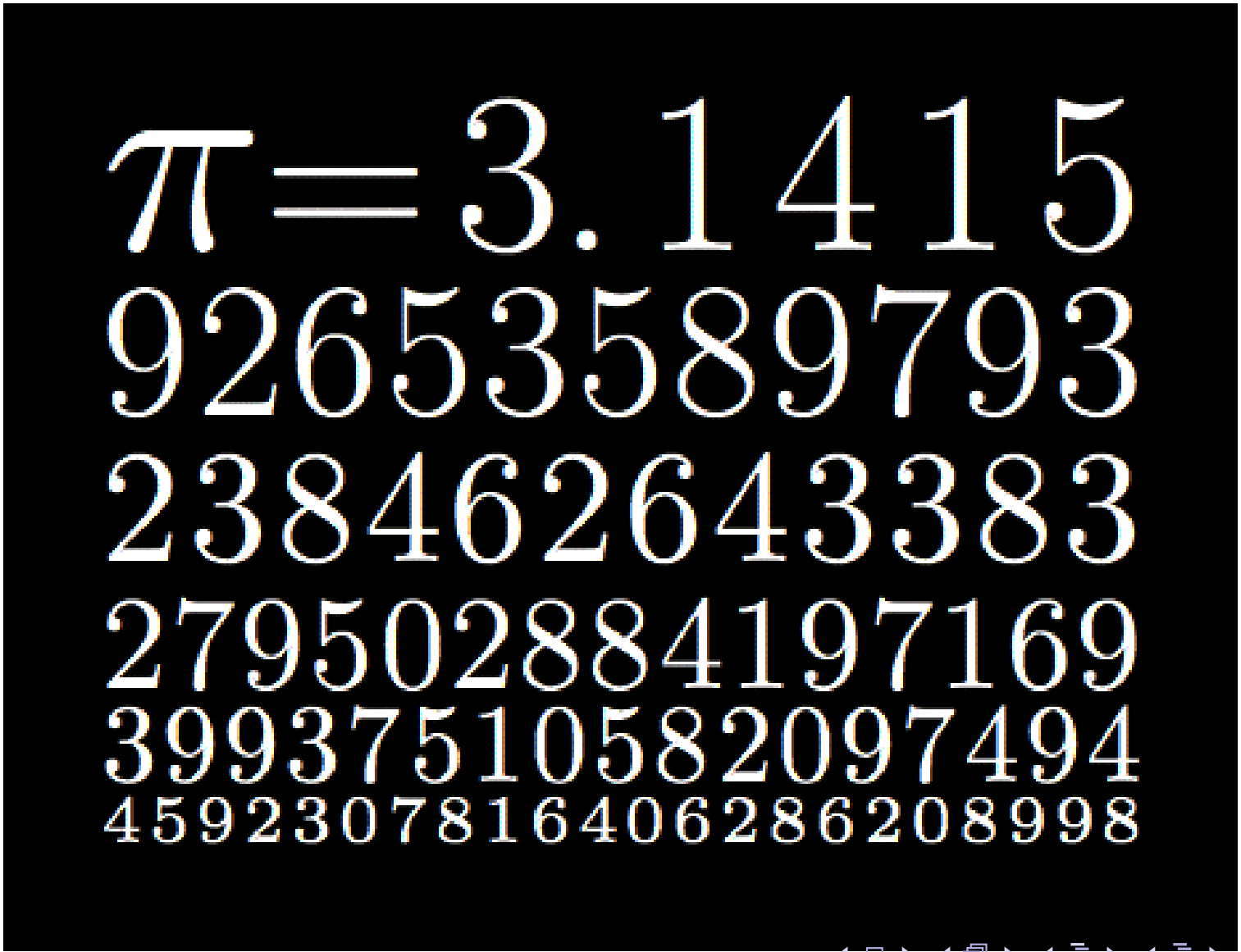
Fair ten–sided die

If the sides are labelled from 0 to 9 then tosses of this die will yield the required digits.



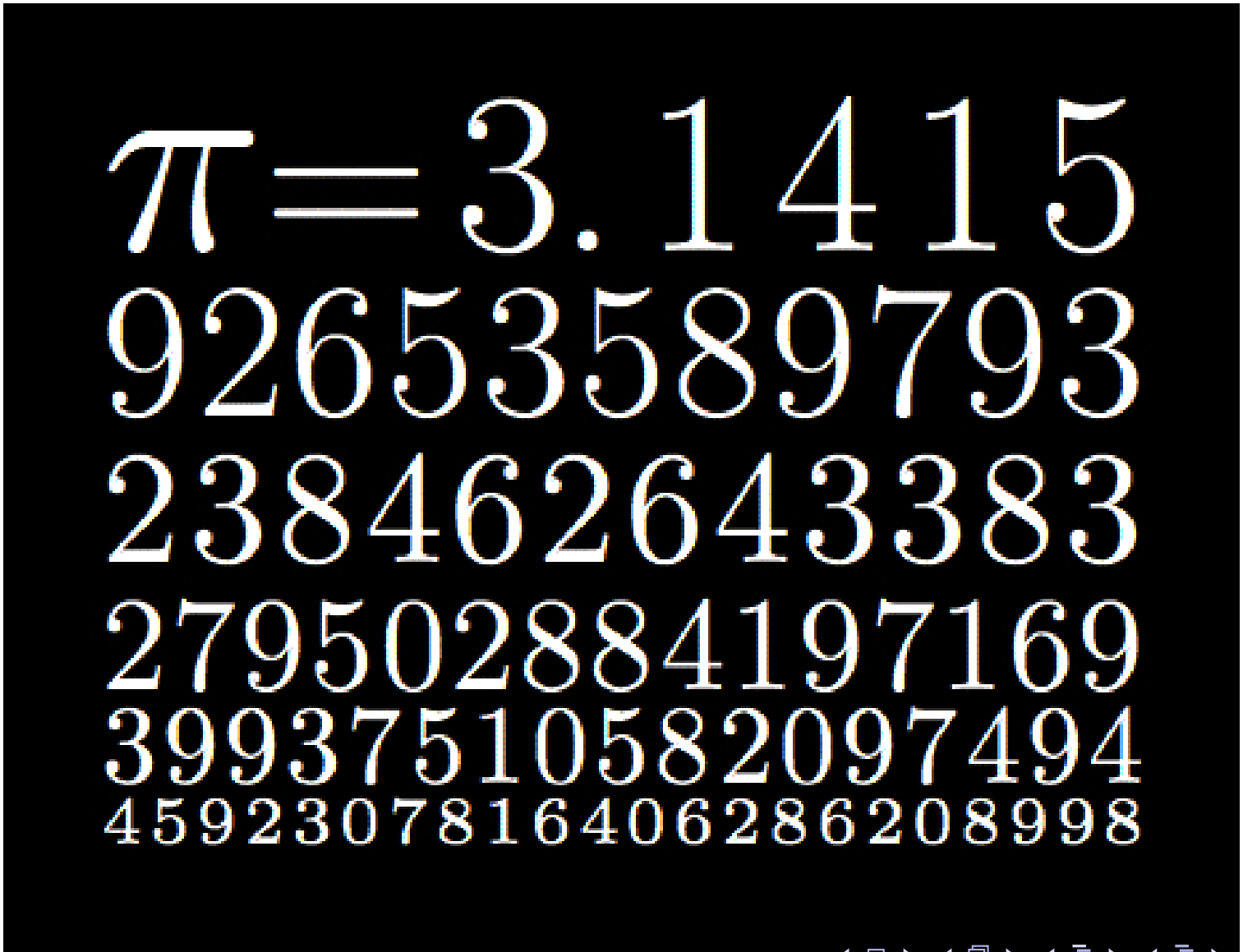
Decimal expansion of π

Irrational number – decimal expansion goes on forever, with no pattern!



Decimal expansion of π

Irrational number – decimal expansion goes on forever, with no pattern!



Tosses of a fair coin

Toss a fair coin four times. The following equally likely outcomes could correspond to the integers shown:

H H H H	0	H T H T	5
H H H T	1	H T T H	6
H H T H	2	H T T T	7
H H T T	3	T H H H	8
H T H H	4	T H H T	9

The coin has no ‘memory’, and so each block of 4 tosses is independent of any other. If any outcome other than those listed occurs, we can ignore and toss a new set of 4.

This method is rather inefficient as a lot of the time a combination of outcomes is rejected!

Tosses of a fair coin

Toss a fair coin four times. The following equally likely outcomes could correspond to the integers shown:

H H H H	0	H T H T	5
H H H T	1	H T T H	6
H H T H	2	H T T T	7
H H T T	3	T H H H	8
H T H H	4	T H H T	9

The coin has no ‘memory’, and so each block of 4 tosses is independent of any other. If any outcome other than those listed occurs, we can ignore and toss a new set of 4.

This method is rather inefficient as a lot of the time a combination of outcomes is rejected!

Tosses of a fair coin

Toss a fair coin four times. The following equally likely outcomes could correspond to the integers shown:

H H H H	0	H T H T	5
H H H T	1	H T T H	6
H H T H	2	H T T T	7
H H T T	3	T H H H	8
H T H H	4	T H H T	9

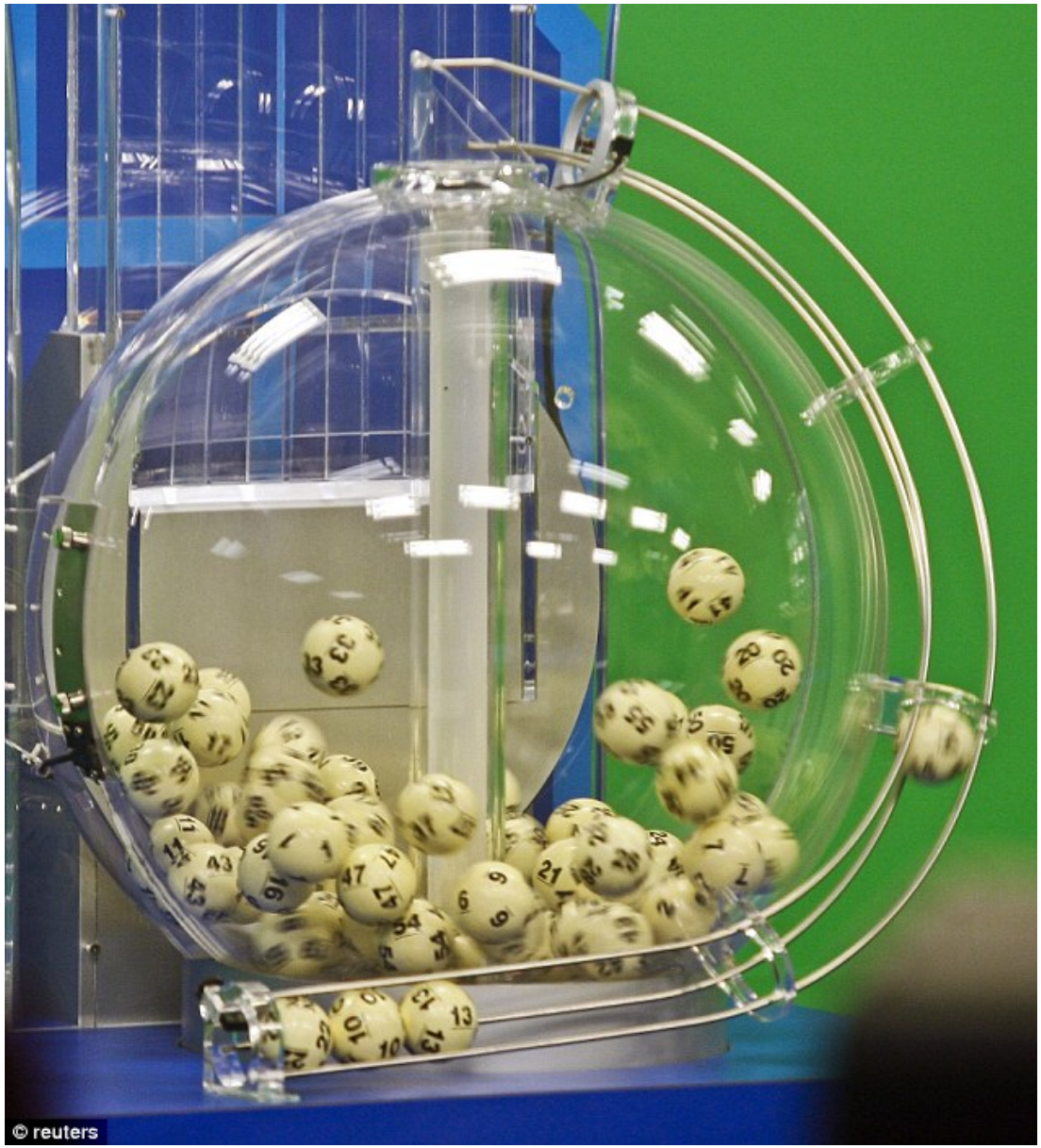
The coin has no ‘memory’, and so each block of 4 tosses is independent of any other. If any outcome other than those listed occurs, we can ignore and toss a new set of 4.

This method is rather inefficient as a lot of the time a combination of outcomes is rejected!

Other physical devices: Wheels of fortune



Other physical devices: Lottery machines



Other physical devices: Gamma ray counters

- Quantum mechanics predicts that the nuclear decay of atoms is random
- **Idea:** Use a geiger counter to generate random numbers!

Other physical devices: Gamma ray counters

- Quantum mechanics predicts that the nuclear decay of atoms is random
- **Idea:** Use a geiger counter to generate random numbers!

5.2 Pseudo–random numbers

- Mechanical and electronic devices are not reproducible...
- So we use *Pseudo–random numbers generators* (PRNG)

5.2 Pseudo–random numbers

- Mechanical and electronic devices are not reproducible...
- So we use *Pseudo–random numbers generators* (PRNG)

5.2 Pseudo–random numbers

The German Federal Office for Information Security (*Bundesamt für Sicherheit in der Informationstechnik*, or BSI) has established criteria for quality RNG:

1. A sequence of random numbers has a high probability of containing no identical consecutive elements
2. A sequence of numbers which is indistinguishable from 'true random' numbers (tested using statistical tests)
3. It should be impossible to calculate – or guess – from any given sub-sequence, any previous or future values in the sequence
4. It should be impossible, for all practical purposes, for an attacker to calculate, or guess, the values used in the random number algorithm

Points 3 and 4 are crucial for many applications.

5.2 Pseudo–random numbers

The German Federal Office for Information Security (*Bundesamt für Sicherheit in der Informationstechnik*, or BSI) has established criteria for quality RNG:

1. A sequence of random numbers has a high probability of containing no identical consecutive elements
2. A sequence of numbers which is indistinguishable from 'true random' numbers (tested using statistical tests)
3. It should be impossible to calculate – or guess – from any given sub-sequence, any previous or future values in the sequence
4. It should be impossible, for all practical purposes, for an attacker to calculate, or guess, the values used in the random number algorithm

Points 3 and 4 are crucial for many applications.

5.2 Pseudo–random numbers

The German Federal Office for Information Security (*Bundesamt für Sicherheit in der Informationstechnik*, or BSI) has established criteria for quality RNG:

1. A sequence of random numbers has a high probability of containing no identical consecutive elements
2. A sequence of numbers which is indistinguishable from ‘true random’ numbers (tested using statistical tests)
3. It should be impossible to calculate – or guess – from any given sub-sequence, any previous or future values in the sequence
4. It should be impossible, for all practical purposes, for an attacker to calculate, or guess, the values used in the random number algorithm

Points 3 and 4 are crucial for many applications.

5.2 Pseudo–random numbers

The German Federal Office for Information Security (*Bundesamt für Sicherheit in der Informationstechnik*, or BSI) has established criteria for quality RNG:

1. A sequence of random numbers has a high probability of containing no identical consecutive elements
2. A sequence of numbers which is indistinguishable from 'true random' numbers (tested using statistical tests)
3. It should be impossible to calculate – or guess – from any given sub-sequence, any previous or future values in the sequence
4. It should be impossible, for all practical purposes, for an attacker to calculate, or guess, the values used in the random number algorithm

Points 3 and 4 are crucial for many applications.

5.2 Pseudo–random numbers

The German Federal Office for Information Security (*Bundesamt für Sicherheit in der Informationstechnik*, or BSI) has established criteria for quality RNG:

1. A sequence of random numbers has a high probability of containing no identical consecutive elements
2. A sequence of numbers which is indistinguishable from ‘true random’ numbers (tested using statistical tests)
3. It should be impossible to calculate – or guess – from any given sub-sequence, any previous or future values in the sequence
4. It should be impossible, for all practical purposes, for an attacker to calculate, or guess, the values used in the random number algorithm

Points 3 and 4 are crucial for many applications.

5.2 Pseudo–random numbers

The German Federal Office for Information Security (*Bundesamt für Sicherheit in der Informationstechnik*, or BSI) has established criteria for quality RNG:

1. A sequence of random numbers has a high probability of containing no identical consecutive elements
2. A sequence of numbers which is indistinguishable from ‘true random’ numbers (tested using statistical tests)
3. It should be impossible to calculate – or guess – from any given sub-sequence, any previous or future values in the sequence
4. It should be impossible, for all practical purposes, for an attacker to calculate, or guess, the values used in the random number algorithm

Points 3 and 4 are crucial for many applications.

5.2 Pseudo–random numbers

As we will see in this course, it is important to be able to reproduce a sequence of random numbers.

The advent of digital computers gave rise to a number of techniques that use a recursive relation, that is, the number r_i in a sequence is a function of the preceding numbers $r_{i-1}, r_{i-2}, \ldots$.

Some of these PRNG's have quite cool names: for example,

- the *Super-Duper* PRNG
- the *Blum-Blum-Shub* PRNG (after Lenore Blum, Manuel Blum and Michael Shub)
- the *Mersenne-Twister* PRNG (as used by R)

5.2 Pseudo–random numbers

As we will see in this course, it is important to be able to reproduce a sequence of random numbers.

The advent of digital computers gave rise to a number of techniques that use a recursive relation, that is, the number r_i in a sequence is a function of the preceding numbers $r_{i-1}, r_{i-2}, \ldots$.

Some of these PRNG's have quite cool names: for example,

- the *Super-Duper* PRNG
- the *Blum-Blum-Shub* PRNG (after Lenore Blum, Manuel Blum and Michael Shub)
- the *Mersenne-Twister* PRNG (as used by R)

5.2 Pseudo–random numbers

As we will see in this course, it is important to be able to reproduce a sequence of random numbers.

The advent of digital computers gave rise to a number of techniques that use a recursive relation, that is, the number r_i in a sequence is a function of the preceding numbers $r_{i-1}, r_{i-2}, \ldots$.

Some of these PRNG's have quite cool names: for example,

- the *Super-Duper* PRNG
- the *Blum-Blum-Shub* PRNG (after Lenore Blum, Manuel Blum and Michael Shub)
- the *Mersenne-Twister* PRNG (as used by R)

5.2 Pseudo–random numbers

As we will see in this course, it is important to be able to reproduce a sequence of random numbers.

The advent of digital computers gave rise to a number of techniques that use a recursive relation, that is, the number r_i in a sequence is a function of the preceding numbers $r_{i-1}, r_{i-2}, \ldots$.

Some of these PRNG's have quite cool names: for example,

- the *Super-Duper* PRNG
- the *Blum-Blum-Shub* PRNG (after Lenore Blum, Manuel Blum and Michael Shub)
- the *Mersenne-Twister* PRNG (as used by R)

5.2 Pseudo–random numbers

As we will see in this course, it is important to be able to reproduce a sequence of random numbers.

The advent of digital computers gave rise to a number of techniques that use a recursive relation, that is, the number r_i in a sequence is a function of the preceding numbers $r_{i-1}, r_{i-2}, \ldots$.

Some of these PRNG's have quite cool names: for example,

- the *Super-Duper* PRNG
- the *Blum-Blum-Shub* PRNG (after Lenore Blum, Manuel Blum and Michael Shub)
- the *Mersenne-Twister* PRNG (as used by R)

5.3.1 The `runif` function

```
> runif(n, min=0, max=1)
```

- This function will generate `n` random numbers between the values of `min` and `max`.
- If the arguments `min` or `max` are omitted, then the default values are 0 and 1 respectively.

5.3.1 The runif function

For example,

```
> runif(1)
[1] 0.2729965
> runif(1)
[1] 0.9990863
> runif(5)
[1] 0.8122783 0.9069950 0.1731072 0.3454292 0.6102412
> runif(1, 6, 7)
[1] 6.427512
```

5.3.1 The runif function

```
> set.seed(12345)
> runif(1)
[1] 0.7209039
> runif(1)
[1] 0.8757732
```


```
> set.seed(12345)
> runif(1)
[1] 0.7209039
```

5.3.1 The runif function

```
> set.seed(12345)
> runif(1)
[1] 0.7209039
> runif(1)
[1] 0.8757732
```

```
> set.seed(12345)
> runif(1)
[1] 0.7209039
```

5.3.2 The sample function

Another important R function that we will use is the sample function:

```
> sample(x, size, replace = FALSE, prob = NULL)
```

5.3.2 The `sample` function

This takes the following arguments

- `x`: a list of values
- `size`: non-negative integer giving the number of items to choose.
- `replace`: Should sampling be with replacement? Default: `FALSE`.
- `prob`: A vector of probability weights. Default: All values equally likely.

5.3.2 Example usage of `sample`

Suppose we wish to sample five numbers from $\{1, 2, 3, 4, 5, 6\}$, then

```
> set.seed(1)
> x = c(1, 2, 3, 4, 5, 6)
> sample(x, 5)
[1] 2 6 3 4 1
```

We can also sample with replacement:

```
> sample(x, 5, replace=TRUE)
[1] 6 6 4 4 1
```

This means that values **may** appear more than once.

5.3.3 Simulating the Captial One Cup draw

We are in the semi-finals of the *Capital One* Cup, and need to organise the draw for the final stage. The remaining teams are:

Manchester Utd, Mancehster City, Sunderland, West Ham.

5.3.3 Simulating the Captial One Cup draw

Here's how we do this in R:

```
> set.seed(3)
> teams = c("Man Utd", "Man City", "Sunderland", "West
Ham")
> sample(teams, 4)
[1] "Man Utd"    "Sunderland" "West Ham"   "Man City"
```

So we have 'Man Utd vs Sunderland' and 'West Ham vs Man City'.

5.3.3 Simulating the Capital One Cup draw

However, if we think Sunderland are likely to get beaten by Man Utd, we can rig the voting:

```
> prob_weights = c(0.4, 0.4, 0.05, 0.2)
> sample(teams, 4, prob=prob_weights)
[1] "Man Utd"    "Man City"   "West Ham"
"Sunderland"
```

That's better!

5.3.3 Simulating the Capital One Cup draw



5.4 Simulating from discrete probability distributions

Suppose we want to simulate a discrete random variable X which has probability mass function

$$\Pr[X = x_j] = p_j \quad \text{for } j = 1, 2, \ldots,$$

and where $\sum_{\text{all}} p_j = 1$.

5.4 Simulating from discrete probability distributions

To accomplish this we first simulate a value u from a $U(0,1)$ distribution and set

$$X = \begin{cases} x_1 & \text{if } u < p_1 \\ x_2 & \text{if } p_1 \le u < p_1 + p_2 \\ x_3 & \text{if } p_1 + p_2 \le u < p_1 + p_2 + p_3 \\ \vdots & \vdots \\ x_j & \text{if } \sum_{i=1}^{j-1} p_i \le u < \sum_{i=1}^{j} p_i \\ \vdots & \vdots \end{cases}$$

5.4 Simulating from discrete probability distributions

To accomplish this we first simulate a value u from a $U(0,1)$ distribution and set

$$X = \begin{cases} x_1 & \text{if } u < p_1 \\ x_2 & \text{if } p_1 \le u < p_1 + p_2 \\ x_3 & \text{if } p_1 + p_2 \le u < p_1 + p_2 + p_3 \\ \vdots & \vdots \\ x_j & \text{if } \sum_{i=1}^{j-1} p_i \le u < \sum_{i=1}^{j} p_i \\ \vdots & \vdots \end{cases}$$

5.4 Simulating from discrete probability distributions

To accomplish this we first simulate a value u from a $U(0,1)$ distribution and set

$$X = \begin{cases} x_1 & \text{if } u < p_1 \\ x_2 & \text{if } p_1 \le u < p_1 + p_2 \\ x_3 & \text{if } p_1 + p_2 \le u < p_1 + p_2 + p_3 \\ \vdots & \vdots \\ x_j & \text{if } \sum_{i=1}^{j-1} p_i \le u < \sum_{i=1}^{j} p_i \\ \vdots & \vdots \end{cases}$$

5.4 Simulating from discrete probability distributions

To accomplish this we first simulate a value u from a $U(0,1)$ distribution and set

$$X = \begin{cases} x_1 & \text{if } u < p_1 \\ x_2 & \text{if } p_1 \le u < p_1 + p_2 \\ x_3 & \text{if } p_1 + p_2 \le u < p_1 + p_2 + p_3 \\ \vdots & \vdots \\ x_j & \text{if } \sum_{i=1}^{j-1} p_i \le u < \sum_{i=1}^{j} p_i \\ \vdots & \vdots \end{cases}$$

5.4 Simulating from discrete probability distributions

To accomplish this we first simulate a value u from a $U(0,1)$ distribution and set

$$X = \begin{cases} x_1 & \text{if } u < p_1 \\ x_2 & \text{if } p_1 \le u < p_1 + p_2 \\ x_3 & \text{if } p_1 + p_2 \le u < p_1 + p_2 + p_3 \\ \vdots & \vdots \\ x_j & \text{if } \sum_{i=1}^{j-1} p_i \le u < \sum_{i=1}^{j} p_i \\ \vdots & \vdots \end{cases}$$

5.4 Simulating from discrete probability distributions

To accomplish this we first simulate a value u from a $U(0,1)$ distribution and set

$$X = \begin{cases} x_1 & \text{if } u < p_1 \\ x_2 & \text{if } p_1 \le u < p_1 + p_2 \\ x_3 & \text{if } p_1 + p_2 \le u < p_1 + p_2 + p_3 \\ \vdots & \vdots \\ x_j & \text{if } \sum_{i=1}^{j-1} p_i \le u < \sum_{i=1}^{j} p_i \\ & \vdots \end{cases}$$

5.4 Simulating from discrete probability distributions

To accomplish this we first simulate a value u from a $U(0,1)$ distribution and set

$$X = \begin{cases} x_1 & \text{if } u < p_1 \\ x_2 & \text{if } p_1 \le u < p_1 + p_2 \\ x_3 & \text{if } p_1 + p_2 \le u < p_1 + p_2 + p_3 \\ \vdots & \vdots \\ x_j & \text{if } \sum_{i=1}^{j-1} p_i \le u < \sum_{i=1}^{j} p_i \\ \vdots & \vdots \end{cases}$$

5.4 Simulating from discrete probability distributions

Now, for a Uniform random variable u, and for $0 < a < b < 1$, it is the case that $\Pr[a \leq U < b] = b - a$. Thus,

$$\begin{aligned} \Pr[X = x_j] &= \Pr\left[\sum_{i=1}^{j-1} p_i \leq U < \sum_{i=1}^{j} p_i\right] \\ &= \sum_{i=1}^{j} p_i - \sum_{i=1}^{j-1} p_i \\ &= p_j \,. \end{aligned}$$

as required.

5.4 Simulating from discrete probability distributions

Now, for a Uniform random variable u, and for $0 < a < b < 1$, it is the case that $\Pr[a \leq U < b] = b - a$.Thus,

$$\begin{aligned} \Pr[X = x_j] &= \Pr\left[\sum_{i=1}^{j-1} p_i \leq U < \sum_{i=1}^{j} p_i\right] \\ &= \sum_{i=1}^{j} p_i - \sum_{i=1}^{j-1} p_i \\ &= p_j \,. \end{aligned}$$

as required.

5.4 Simulating from discrete probability distributions

Now, for a Uniform random variable u, and for $0 < a < b < 1$, it is the case that $\Pr[a \leq U < b] = b - a$.Thus,

$$\begin{aligned} \Pr[X = x_j] &= \Pr\left[\sum_{i=1}^{j-1} p_i \leq U < \sum_{i=1}^{j} p_i\right] \\ &= \sum_{i=1}^{j} p_i - \sum_{i=1}^{j-1} p_i \\ &= p_j \,. \end{aligned}$$

as required.

5.4 Simulating from discrete probability distributions

Now, for a Uniform random variable u, and for $0 < a < b < 1$, it is the case that $\Pr[a \leq U < b] = b - a$.Thus,

$$\begin{aligned} \Pr[X = x_j] &= \Pr\left[\sum_{i=1}^{j-1} p_i \leq U < \sum_{i=1}^{j} p_i\right] \\ &= \sum_{i=1}^{j} p_i - \sum_{i=1}^{j-1} p_i \\ &= p_j \,. \end{aligned}$$

as required.

5.4.1 Example

Simulate a random variable with the following probability mass function:

x	1	2	3	4
$\Pr[X = x]$	0.2	0.15	0.25	0.4

5.4.1 Solution

We calculate the CDF of the distribution

x	1	2	3	4
$\Pr[X = x]$	0.2	0.15	0.25	0.4
$\Pr[X \leq x]$	0.2	0.35	0.60	1.0

Then we generate an observation u from $U \sim \text{Uniform}(0, 1)$, so

- if $u < 0.2$, set $X = 1$;
- if $0.2 \leq u < (0.2 + 0.15) = 0.35$, set $X = 2$;
- if $0.35 \leq u < (0.2 + 0.15 + 0.25) = 0.6$, set $X = 3$;
- if $u \geq 0.6$, set $X = 4$.

Suppose $u = 0.821$. Then $X = 4$.

5.4.1 Solution

We calculate the CDF of the distribution

x	1	2	3	4
$\Pr[X = x]$	0.2	0.15	0.25	0.4
$\Pr[X \leq x]$	0.2	0.35	0.60	1.0

Then we generate an observation u from $U \sim \text{Uniform}(0, 1)$, so

- if $u < 0.2$, set $X = 1$;
- if $0.2 \leq u < (0.2 + 0.15) = 0.35$, set $X = 2$;
- if $0.35 \leq u < (0.2 + 0.15 + 0.25) = 0.6$, set $X = 3$;
- if $u \geq 0.6$, set $X = 4$.

Suppose $u = 0.821$. Then $X = 4$.

5.4.1 Solution

We calculate the CDF of the distribution

x	1	2	3	4
$\Pr[X = x]$	0.2	0.15	0.25	0.4
$\Pr[X \leq x]$	0.2	0.35	0.60	1.0

Then we generate an observation u from $U \sim \text{Uniform}(0, 1)$, so

- if $u < 0.2$, set $X = 1$;
- if $0.2 \leq u < (0.2 + 0.15) = 0.35$, set $X = 2$;
- if $0.35 \leq u < (0.2 + 0.15 + 0.25) = 0.6$, set $X = 3$;
- if $u \geq 0.6$, set $X = 4$.

Suppose $u = 0.821$. Then $X = 4$.

5.4.1 Solution

We calculate the CDF of the distribution

x	1	2	3	4
$\Pr[X = x]$	0.2	0.15	0.25	0.4
$\Pr[X \leq x]$	0.2	0.35	0.60	1.0

Then we generate an observation u from $U \sim \text{Uniform}(0, 1)$, so

- if $u < 0.2$, set $X = 1$;
- if $0.2 \leq u < (0.2 + 0.15) = 0.35$, set $X = 2$;
- if $0.35 \leq u < (0.2 + 0.15 + 0.25) = 0.6$, set $X = 3$;
- if $u \geq 0.6$, set $X = 4$.

Suppose $u = 0.821$. Then $X = 4$.

5.4.1 Solution

We calculate the CDF of the distribution

x	1	2	3	4
$\Pr[X = x]$	0.2	0.15	0.25	0.4
$\Pr[X \leq x]$	0.2	0.35	0.60	1.0

Then we generate an observation u from $U \sim \text{Uniform}(0, 1)$, so

- if $u < 0.2$, set $X = 1$;
- if $0.2 \leq u < (0.2 + 0.15) = 0.35$, set $X = 2$;
- if $0.35 \leq u < (0.2 + 0.15 + 0.25) = 0.6$, set $X = 3$;
- if $u \geq 0.6$, set $X = 4$.

Suppose $u = 0.821$. Then $X = 4$.

5.4.1 Solution

We calculate the CDF of the distribution

x	1	2	3	4
$\Pr[X = x]$	0.2	0.15	0.25	0.4
$\Pr[X \leq x]$	0.2	0.35	0.60	1.0

Then we generate an observation u from $U \sim \text{Uniform}(0, 1)$, so

- if $u < 0.2$, set $X = 1$;
- if $0.2 \leq u < (0.2 + 0.15) = 0.35$, set $X = 2$;
- if $0.35 \leq u < (0.2 + 0.15 + 0.25) = 0.6$, set $X = 3$;
- if $u \geq 0.6$, set $X = 4$.

Suppose $u = 0.821$. Then $X = 4$.

5.4.1 Solution

We calculate the CDF of the distribution

x	1	2	3	4
$\Pr[X = x]$	0.2	0.15	0.25	0.4
$\Pr[X \leq x]$	0.2	0.35	0.60	1.0

Then we generate an observation u from $U \sim \text{Uniform}(0, 1)$, so

- if $u < 0.2$, set $X = 1$;
- if $0.2 \leq u < (0.2 + 0.15) = 0.35$, set $X = 2$;
- if $0.35 \leq u < (0.2 + 0.15 + 0.25) = 0.6$, set $X = 3$;
- if $u \geq 0.6$, set $X = 4$.

Suppose $u = 0.821$. Then $X = 4$.

5.4.1 Solution

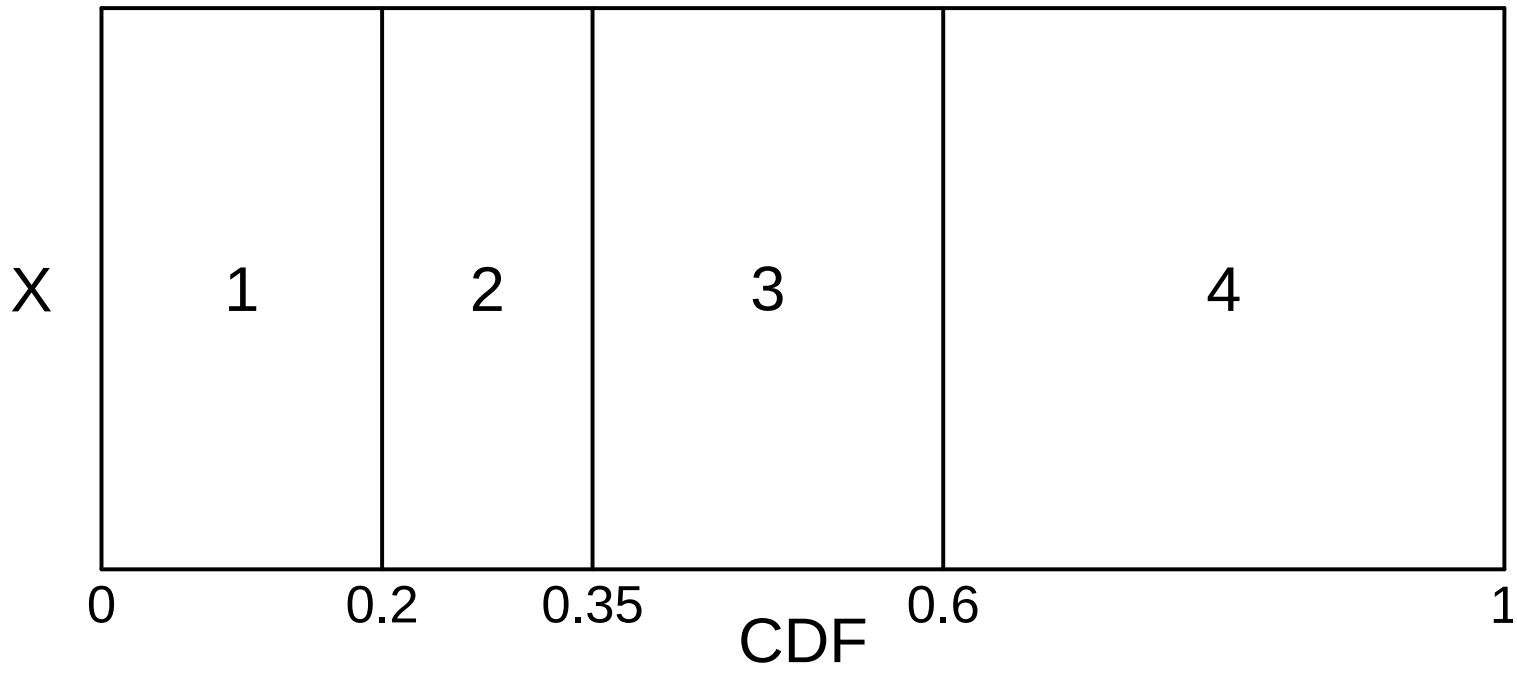


Figure: Simulating discrete random numbers.

5.4.1 Using R

To simulate from the above distribution in R, there are two (obvious) methods that we can use: the `sample` command:

```
> x = c(1, 2, 3, 4)
> prob = c(0.2, 0.15, 0.25, 0.4)
> sum(prob)
[1] 1
> sample(x, 1, prob, replace=TRUE)
[1] 4
```

5.4.1 Using R

Or use a bunch of `if` statements:

```
> u = runif(1)
> if(u <= 0.2) {
+   X = 1
+ } else if(u <= (0.2+0.15)) {
+   X = 2
+ } else if(u <= (0.2+0.15+0.25)) {
+   X = 3
+ } else {
+   X = 4
+ }
```

5.4.2 Simulating a Poisson random variable

For the uniform random numbers,

$$0.253,\ 0.588,\ 0.789,$$

simulate three random numbers from a Poisson distribution with mean $\lambda = 2$.

5.4.2 Solution

The pdf of the poisson distribution is

$$Pr[X = x] = \frac{e^{-\lambda}\lambda^{x}}{x!}$$

where $\lambda > 0$. So we have

x	0	1	2	3	4
$\Pr[X = x]$	0.135	0.271	0.271	0.180	0.090
$\Pr[X \leq x]$	0.135	0.406	0.677	0.857	0.947

5.4.2 Solution

The pdf of the poisson distribution is

$$Pr[X = x] = \frac{e^{-\lambda}\lambda^{x}}{x!}$$

where $\lambda > 0$. So we have

x	0	1	2	3	4
$\Pr[X = x]$	0.135	0.271	0.271	0.180	0.090
$\Pr[X \leq x]$	0.135	0.406	0.677	0.857	0.947

5.4.2 Solution

The pdf of the poisson distribution is

$$Pr[X = x] = \frac{e^{-\lambda}\lambda^{x}}{x!}$$

where $\lambda > 0$. So we have

x	0	1	2	3	4
$\Pr[X = x]$	0.135	0.271	0.271	0.180	0.090
$\Pr[X \leq x]$	0.135	0.406	0.677	0.857	0.947

5.4.2 Solution

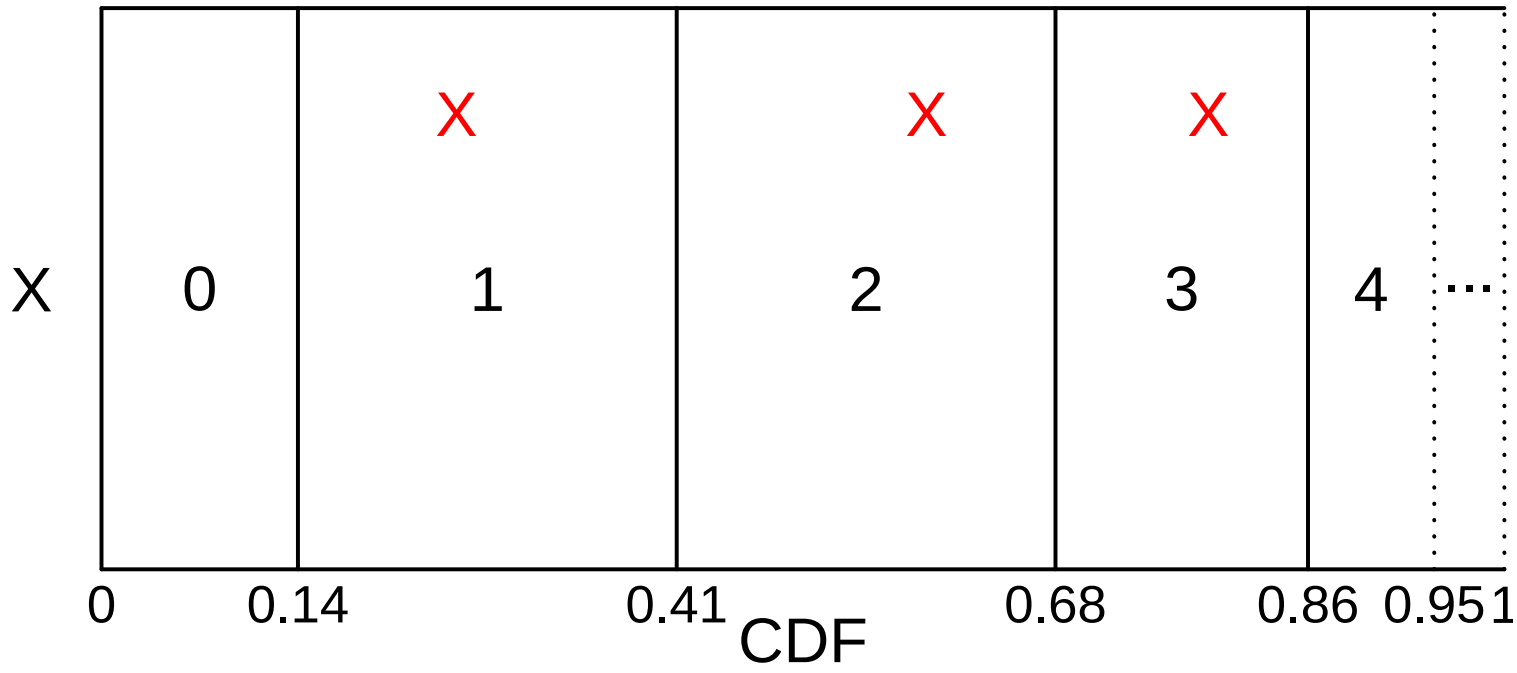


Figure: Diagram illustrating on simulating discrete random numbers from the Poisson distribution.

Hence our random numbers are 1, 2 and 3.

5.4.2 Using R

```
> rpois(1,2)
[1] 3
```

```
> rpois(10,2)
[1] 0 1 2 0 0 0 2 1 3 1
```

5.4.3 Simulating from other distributions

Simulating random values from other standard probability distributions is easy in R.

Just as we have seen the intrinsic functions `runif` and `rpois`, there are other functions such as `rgeom` and `rbinom`.

5.4.3 Simulating from other distributions

Simulating random values from other standard probability distributions is easy in R.

Just as we have seen the intrinsic functions `runif` and `rpois`, there are other functions such as `rgeom` and `rbinom`.

5.4.3 Simulating from other distributions

For example, suppose $X \sim \text{Bin}(6, 0.2)$ and $Y \sim \text{Geom}(0.75)$, and we want to simulate 5 random values from X and 3 random values from Y:

```
> (X=rbinom(5,6,0.2))
[1] 0 1 1 3 0
```

```
> (Y=rgeom(3,0.75))
[1] 2 0 1
```

5.4.3 Variations on a theme

We have seen, for example, `runif`, `rpois`, `rbinom` and `rgeom`.

Replacing the `r` prefix with:

- d, calculates the density of the distribution at a particular value.
  For example, `dpois(1,2)` would find $Pr[X = 1]$ when $X \sim Po(2)$
- p, calculates the distribution function at a particular value.
  For example, `pbinom(2,6,0.2)` would find
  $Pr[X \leq 2] = Pr[X = 0] + Pr[X = 1] + Pr[X = 2]$ when
  $X \sim \text{Bin}(6, 0.2)$
- q, finds quantiles of a distribution for a particular probability.
  For example, `qnorm(0.7,0,1)` finds x such that $Pr[X \leq x] = 0.75$,
  when $X \sim N(0, 1)$

5.4.3 Variations on a theme

We have seen, for example, `runif`, `rpois`, `rbinom` and `rgeom`.

Replacing the `r` prefix with:

- d, calculates the density of the distribution at a particular value.
  For example, `dpois(1,2)` would find $Pr[X = 1]$ when $X \sim Po(2)$
- p, calculates the distribution function at a particular value.
  For example, `pbinom(2,6,0.2)` would find
  $Pr[X \leq 2] = Pr[X = 0] + Pr[X = 1] + Pr[X = 2]$ when
  $X \sim \text{Bin}(6, 0.2)$
- q, finds quantiles of a distribution for a particular probability.
  For example, `qnorm(0.7,0,1)` finds x such that $Pr[X \leq x] = 0.75$,
  when $X \sim N(0, 1)$

5.4.3 Variations on a theme

We have seen, for example, `runif`, `rpois`, `rbinom` and `rgeom`.

Replacing the `r` prefix with:

- d, calculates the density of the distribution at a particular value.
  For example, `dpois(1,2)` would find $Pr[X = 1]$ when $X \sim Po(2)$
- p, calculates the distribution function at a particular value.
  For example, `pbinom(2,6,0.2)` would find
  $Pr[X \leq 2] = Pr[X = 0] + Pr[X = 1] + Pr[X = 2]$ when
  $X \sim \text{Bin}(6, 0.2)$
- q, finds quantiles of a distribution for a particular probability.
  For example, `qnorm(0.7,0,1)` finds x such that $Pr[X \leq x] = 0.75$,
  when $X \sim N(0, 1)$

5.4.3 Variations on a theme

We have seen, for example, `runif`, `rpois`, `rbinom` and `rgeom`.

Replacing the `r` prefix with:

- d, calculates the density of the distribution at a particular value.
  For example, `dpois(1,2)` would find $Pr[X = 1]$ when $X \sim Po(2)$
- p, calculates the distribution function at a particular value.
  For example, `pbinom(2,6,0.2)` would find $Pr[X \leq 2] = Pr[X = 0] + Pr[X = 1] + Pr[X = 2]$ when $X \sim \text{Bin}(6, 0.2)$
- q, finds quantiles of a distribution for a particular probability.
  For example, `qnorm(0.7,0,1)` finds x such that $Pr[X \leq x] = 0.75$, when $X \sim N(0, 1)$

5.4.3 Variations on a theme

We have seen, for example, `runif`, `rpois`, `rbinom` and `rgeom`.

Replacing the `r` prefix with:

- d, calculates the density of the distribution at a particular value.
  For example, `dpois(1,2)` would find $Pr[X = 1]$ when $X \sim Po(2)$
- p, calculates the distribution function at a particular value.
  For example, `pbinom(2,6,0.2)` would find
  $Pr[X \leq 2] = Pr[X = 0] + Pr[X = 1] + Pr[X = 2]$ when
  $X \sim \text{Bin}(6, 0.2)$
- q, finds quantiles of a distribution for a particular probability.
  For example, `qnorm(0.7,0,1)` finds x such that $Pr[X \leq x] = 0.75$, when $X \sim N(0, 1)$

5.4.3 Variations on a theme

We have seen, for example, `runif`, `rpois`, `rbinom` and `rgeom`.

Replacing the `r` prefix with:

- d, calculates the density of the distribution at a particular value.
  For example, `dpois(1,2)` would find $Pr[X = 1]$ when $X \sim Po(2)$
- p, calculates the distribution function at a particular value.
  For example, `pbinom(2,6,0.2)` would find
  $Pr[X \leq 2] = Pr[X = 0] + Pr[X = 1] + Pr[X = 2]$ when
  $X \sim \text{Bin}(6, 0.2)$
- q, finds quantiles of a distribution for a particular probability.
  For example, `qnorm(0.7,0,1)` finds x such that $Pr[X \leq x] = 0.75$, when $X \sim N(0, 1)$

5.4.3 Variations on a theme

We have seen, for example, `runif`, `rpois`, `rbinom` and `rgeom`.

Replacing the `r` prefix with:

- d, calculates the density of the distribution at a particular value.
  For example, `dpois(1,2)` would find $Pr[X = 1]$ when $X \sim Po(2)$
- p, calculates the distribution function at a particular value.
  For example, `pbinom(2,6,0.2)` would find
  $Pr[X \leq 2] = Pr[X = 0] + Pr[X = 1] + Pr[X = 2]$ when
  $X \sim \text{Bin}(6, 0.2)$
- q, finds quantiles of a distribution for a particular probability.
  For example, `qnorm(0.7,0,1)` finds x such that $Pr[X \leq x] = 0.75$,
  when $X \sim N(0, 1)$

Part VI

Monte Carlo Methods

6.1 Introduction

- The term “Monte Carlo” is used to describe any simulation study which involves random numbers.
- The name is a reference to the famous Monte Carlo Casino in Monaco, where repetition of random events is the order of the day!

6.2 What is a simulation study?

- For our purposes, a simulation study is any study where we study the properties of a system using random numbers.
- We have already seen a simulation study: for example, the monopoly excercise in your project.
- In general, suppose we do an experiment which has the event A as one possible outcome.
- We would like to estimate the probability of A, denoted by $\Pr[A]$.

6.2 What is a simulation study?

- For our purposes, a simulation study is any study where we study the properties of a system using random numbers.
- We have already seen a simulation study: for example, the monopoly excercise in your project.
- In general, suppose we do an experiment which has the event A as one possible outcome.
- We would like to estimate the probability of A, denoted by $\Pr[A]$.

6.2 What is a simulation study?

- Then by repeatedly simulating the experiment, it is simple to estimate $\Pr[A]$, the probability of the event A, using $P_F(A)$, where:

$$P_F(A) = \frac{\text{No. of times } A \text{ occurs}}{\text{Number of times experiment simulated}}.$$

- Here $P_F(A)$ is the *frequency estimate* of $\Pr[A]$.
- Why does this work?

6.2 What is a simulation study?

- Well suppose we denote the number of times we simulate the experiment by n, then $P_F(A)$ has the following important property:

$$\text{as } n \to \infty, \text{ then } P_F(A) \to \Pr[A].$$

- This means that the more simulations we do, the more *accurate* our estimate of $\Pr[A]$.

6.3 Monte Carlo Integration

- We would like to evaluate $\int_0^1 f(x)\,dx$, but the function may be too complicated to integrate.
- We can find an approximate answer by noting that the integral is equal to the area under the curve, and using Monte Carlo methods.
- We design an experiment which would work in general, even if the function was defined on a general range (a, b), and if $f(x) \in (0, c)$, for any positive value c.

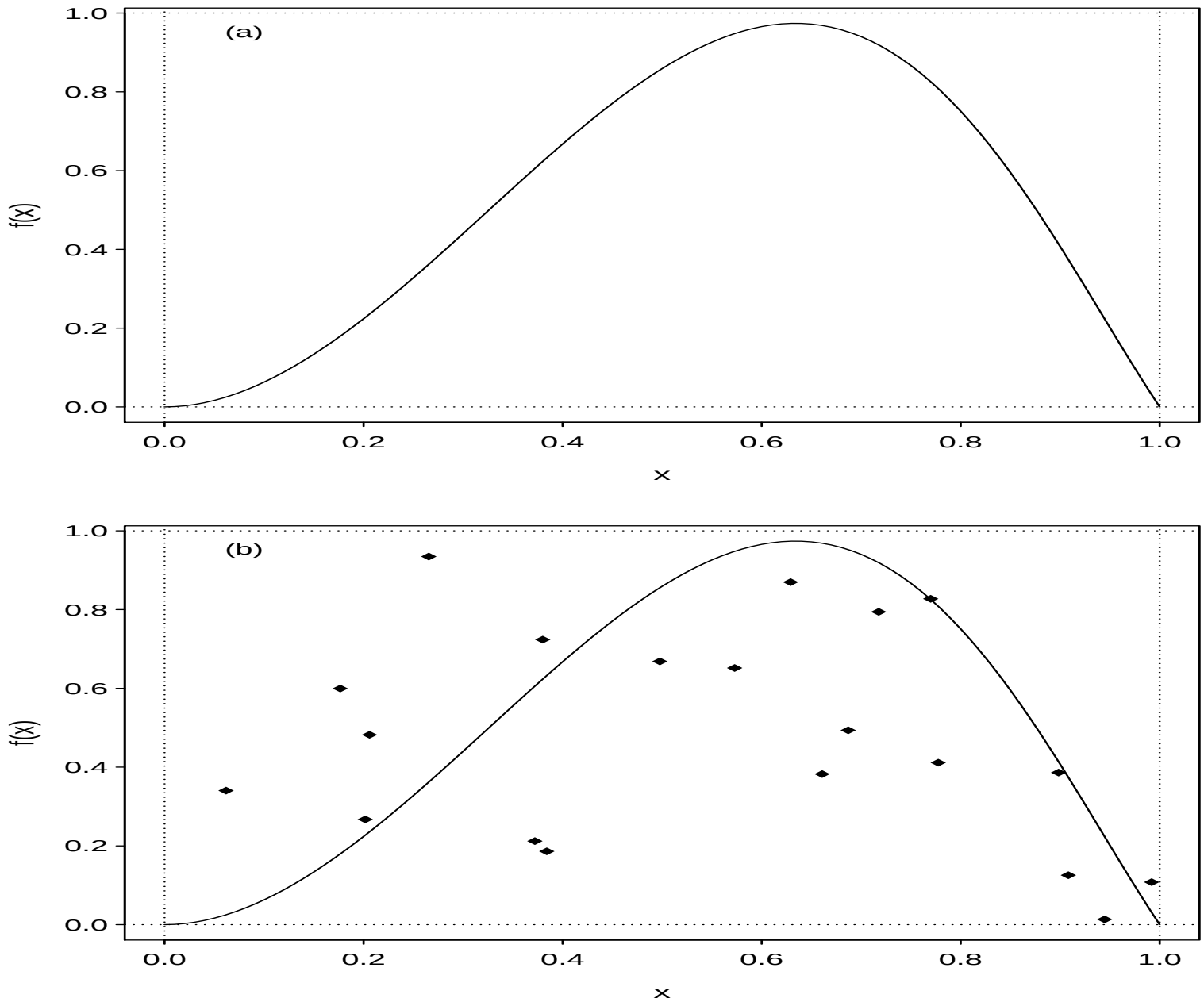


Figure: (a) An example function. (b) Twenty points randomly placed on the graph.

6.3 Monte Carlo Integration

We generate a random data point from a *simulation grid*. Let

$$A = \{\text{the data point lies below the curve}\}.$$

Then

$$\Pr[A] = \frac{\text{area under curve}}{\text{area of simulation grid}},$$

so that

$$\begin{aligned}\int_a^b f(x)\,dx &= [\text{area under curve}] \\ &= \Pr[A] \times [\text{area of simulation grid}].\end{aligned}$$

6.3 Monte Carlo Integration

We generate a random data point from a *simulation grid*. Let

$$A = \{\text{the data point lies below the curve}\}.$$

Then

$$\Pr[A] = \frac{\text{area under curve}}{\text{area of simulation grid}},$$

so that

$$\int_a^b f(x)\,dx = [\text{area under curve}]$$
$$= \Pr[A] \times [\text{area of simulation grid}].$$

6.3 Monte Carlo Integration

We generate a random data point from a *simulation grid*. Let

$$A = \{\text{the data point lies below the curve}\}.$$

Then

$$\Pr[A] = \frac{\text{area under curve}}{\text{area of simulation grid}},$$

so that

$$\begin{aligned}\int_a^b f(x)\,dx &= [\text{area under curve}] \\ &= \Pr[A] \times [\text{area of simulation grid}].\end{aligned}$$

6.3.1 Example

- Consider the function we saw earlier in Figure 6.1, defined on $(0, 1)$, and with $f(x) \in (0, 1)$.
- Estimate $\int_0^1 f(x)dx$ using Monte Carlo methods.

6.3.1 Solution: Step 1

We simulate n data points from the simulation grid; here this is the unit square $(0,1) \times (0,1)$.

6.3.1 Solution: Step 1

We simulate n data points from the simulation grid; here this is the unit square $(0, 1) \times (0, 1)$.

6.3.1 Solution: Step 2

Each of the coordinates x and y are generated using a $U(0,1)$ random variable. Note the fact that

$$A = \{\text{data point } (x, y) \text{ lies below the curve}\} = \{y < f(x)\}.$$

6.3.1 Solution: Step 2

Each of the coordinates x and y are generated using a $U(0,1)$ random variable. Note the fact that

$$A = \{\text{data point } (x, y) \text{ lies below the curve}\} = \{y < f(x)\}.$$

6.3.1 Solution: Step 2

Each of the coordinates x and y are generated using a $U(0,1)$ random variable. Note the fact that

$$A = \{\text{data point } (x, y) \text{ lies below the curve}\} = \{y < f(x)\}.$$

6.3.1 Solution: Step 2

Each of the coordinates x and y are generated using a $U(0,1)$ random variable. Note the fact that

$$A = \{\text{data point } (x, y) \text{ lies below the curve}\} = \{y < f(x)\}.$$

6.3.1 Solution: Step 3

If r points lie below the curve, then $\Pr[A] \simeq P_F(a) = r/n$, and thus

$$\begin{aligned}
\int_0^1 f(x)\,dx &= \Pr[A] \times [\text{area of simulation grid}] \\
&= \Pr[A] \times 1 \\
&= \Pr[A] \\
&\simeq P_F(A) \\
&= r/n.
\end{aligned}$$

6.3.1 Solution: Step 3

If r points lie below the curve, then $\Pr[A] \simeq P_F(a) = r/n$, and thus

$$\begin{aligned}
\int_0^1 f(x)\,dx &= \Pr[A] \times [\text{area of simulation grid}] \\
&= \Pr[A] \times 1 \\
&= \Pr[A] \\
&\simeq P_F(A) \\
&= r/n.
\end{aligned}$$

6.3.1 Solution: Step 3

If r points lie below the curve, then $\Pr[A] \simeq P_F(a) = r/n$, and thus

$$\begin{aligned}
\int_0^1 f(x)\,dx &= \Pr[A] \times [\text{area of simulation grid}] \\
&= \Pr[A] \times 1 \\
&= \Pr[A] \\
&\simeq P_F(A) \\
&= r/n.
\end{aligned}$$

6.3.1 Solution: Step 3

If r points lie below the curve, then $\Pr[A] \simeq P_F(a) = r/n$, and thus

$$\begin{aligned}
\int_0^1 f(x)\,dx &= \Pr[A] \times \text{[area of simulation grid]} \\
&= \Pr[A] \times 1 \\
&= \Pr[A] \\
&\simeq P_F(A) \\
&= r/n.
\end{aligned}$$

6.3.1 Solution: Step 3

If r points lie below the curve, then $\Pr[A] \simeq P_F(a) = r/n$, and thus

$$\begin{aligned}
\int_0^1 f(x)\,dx &= \Pr[A] \times [\text{area of simulation grid}] \\
&= \Pr[A] \times 1 \\
&= \Pr[A] \\
&\simeq P_F(A) \\
&= r/n.
\end{aligned}$$

6.3.1 Solution: Step 3

If r points lie below the curve, then $\Pr[A] \simeq P_F(a) = r/n$, and thus

$$\begin{aligned}
\int_0^1 f(x)\,dx &= \Pr[A] \times [\text{area of simulation grid}] \\
&= \Pr[A] \times 1 \\
&= \Pr[A] \\
&\simeq P_F(A) \\
&= r/n.
\end{aligned}$$

6.3.1 Solution: Step 3

If r points lie below the curve, then $\Pr[A] \simeq P_F(a) = r/n$, and thus

$$\begin{aligned}
\int_0^1 f(x)\,dx &= \Pr[A] \times [\text{area of simulation grid}] \\
&= \Pr[A] \times 1 \\
&= \Pr[A] \\
&\simeq P_F(A) \\
&= r/n.
\end{aligned}$$

6.3.1 Solution: Step 3

If r points lie below the curve, then $\Pr[A] \simeq P_F(a) = r/n$, and thus

$$\begin{aligned}
\int_0^1 f(x)\,dx &= \Pr[A] \times [\text{area of simulation grid}] \\
&= \Pr[A] \times 1 \\
&= \Pr[A] \\
&\simeq P_F(A) \\
&= r/n.
\end{aligned}$$

6.3.1 Solution: Step 3

If r points lie below the curve, then $\Pr[A] \simeq P_F(a) = r/n$, and thus

$$\begin{aligned}
\int_0^1 f(x)\,dx &= \Pr[A] \times [\text{area of simulation grid}] \\
&= \Pr[A] \times 1 \\
&= \Pr[A] \\
&\simeq P_F(A) \\
&= r/n.
\end{aligned}$$

6.3.1 Solution: Step 3

If r points lie below the curve, then $\Pr[A] \simeq P_F(a) = r/n$, and thus

$$\begin{aligned}
\int_0^1 f(x)\,dx &= \Pr[A] \times [\text{area of simulation grid}] \\
&= \Pr[A] \times 1 \\
&= \Pr[A] \\
&\simeq P_F(A) \\
&= r/n.
\end{aligned}$$

6.3.1 Solution: Step 4

In our case, from the figure above we estimate:

$$\int_0^1 f(x)\,dx \simeq \frac{r}{n} = \frac{12}{20} = 0.6\ .$$

6.3.1 Solution: Step 4

In our case, from the figure above we estimate:

$$\int_0^1 f(x)\,dx \simeq \frac{r}{n} = \frac{12}{20} = 0.6\ .$$

6.3.1 Solution: Step 4

In our case, from the figure above we estimate:

$$\int_0^1 f(x)\,dx \simeq \frac{r}{n} = \frac{12}{20} = 0.6\ .$$

6.3.1 Solution: Step 4

In our case, from the figure above we estimate:

$$\int_0^1 f(x)\,dx \simeq \frac{r}{n} = \frac{12}{20} = 0.6\ .$$

6.3.2 Example using R

We can also easily implement the above algorithm in R. As an extreme example, consider the function

$$f(x) = \left| \sin\{\sin[\sin(x^4)]\} \right| .$$

We wish to calculate

$$\int_0^2 f(x)\, dx = \int_0^2 \left| \sin\{\sin[\sin(x^4)]\} \right| dx ,$$

(which evaluates to approximately 0.71875). First plot the function to determine the necessary region:

6.3.2 Example using R

We can also easily implement the above algorithm in R. As an extreme example, consider the function

$$f(x) = \left| \sin\{\sin[\sin(x^4)]\}\right| .$$

We wish to calculate

$$\int_0^2 f(x)\,dx = \int_0^2 \left| \sin\{\sin[\sin(x^4)]\}\right| dx ,$$

(which evaluates to approximately 0.71875). First plot the function to determine the necessary region:

6.3.2 Example using R

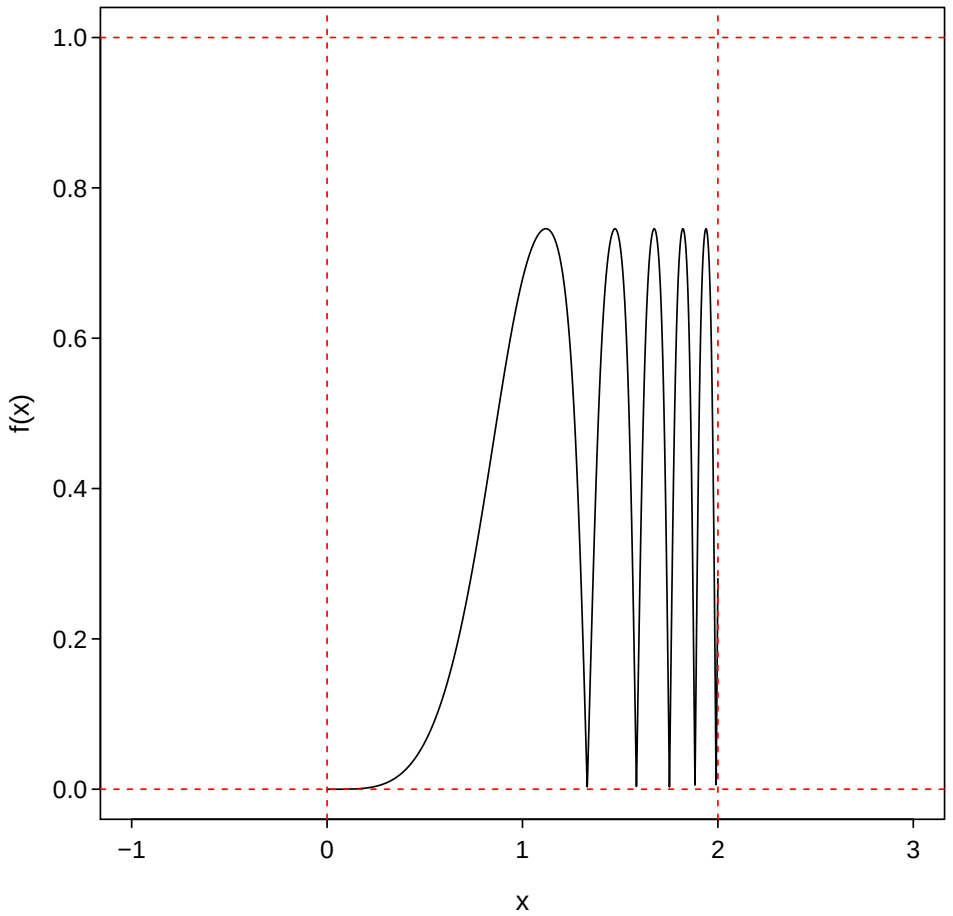


Figure: A plot of $\left|\sin\{\sin[\sin(x^4)]\}\right|$ with the sampling region.

6.3.2 Example using R

The plot generated in the code above is shown in figure 6.2. Then we use a `for` loop to simulate lots of random numbers:

```
> set.seed(1)
> N = 100000
> no_of_hits = 0
> for(i in 1:N) {
+   x = runif(1, 0, 2); y = runif(1)
+   if(abs(sin(sin(sin(x^4)))) > y) {
+     no_of_hits = no_of_hits + 1
+   }
+ }
> area_under_curve = no_of_hits/N*2
> area_under_curve
```