

# Chapter 1 - Module Administration and Assessment

## 1 General Module Information

Note that this module is taken as

- MAS6002 by MSc students in the School of Mathematics and Statistics
- MAS6024 by MSc students in the School of Computer Science
- MAS468 by MMath students in the School of Mathematics and Statistics

### 1.1 Module Delivery

The module is run as a series of 1.5 or 2 hour computer classes. There are no formal lectures as lectures are not a good way for you to learn programming skills. After each session I will post a Blackboard message outlining anything that I think distance-learners will find interesting or useful. I'll also emphasize aspects that people found difficult in the session. The computer sessions will finish before the Christmas break so there are no computer classes in Semester 2. There is a reading week (usually in Week 7); there is no class that week.

### 1.2 Guidelines Relating to Blackboard

You can set Blackboard up to send you an email notification when a new message is posted on the discussion board by following these instructions:

1. Click on your name on the left hand side of the screen on the main Blackboard page.
2. Click on 'Email notifications'.
3. Make sure 'New discussion messages' is ticked and choose how often you want to get emailed.

**Exercise:** Do this now. This should now happen for all your modules.

You also have the option of posting anonymously if you are more comfortable doing that.

Please don't email staff with questions. Instead put them on Blackboard for everyone to see. Please answer questions from other students - this is what the discussion boards on Blackboard are for. Although it is polite to say thank you when someone answers your question, try to avoid doing this so that we only get Blackboard message notifications for something we want to read.

If you have a question about the assignments that you don't think should be put on Blackboard then do email the person setting the assignment and they will decide whether it is general enough to appear on the message boards. If you are really struggling with an assignment please contact the person setting the assignment and they may be able to offer some further explanation or hints. Staff will do their best to help you as much as they reasonably can but only if you ask.

### 1.3 Office Hours

You are welcome to come and see me whenever you need help. Please just invite me to a meeting (in my room, Hicks I22a) via my Google calendar. Instructions for doing this are below:

1. Go to Google calendar through MUSE
2. On the left is the *other calendars* option. Enter the email address of the person you'd like to invite. You should be able to see their calendar
3. Pick a slot in your calendar and enter an *event*
4. Click on the event in your calendar and click on *Edit event*
5. Go to *Add Guests* on the right
6. Invite the person (with their email address)
7. The person you invite gets an email saying they have been invited
8. The person accepts/rejects the invite

You should get an email from the person saying they've accepted/rejected the meeting. If you don't something has gone wrong. Distance-learners are welcome to Skype or phone me by appointment.

## 2 General Information about Assessments

MAS6002 students should refer to the documents *Rules for non-invigilated assessment* and *Submitting coursework via Blackboard* for general information about assessment rules and policies on the MSc. MAS468 should already be familiar with rules relating to coursework. MAS6024 students should be given information from Computer Science.

### 2.1 Which Assignments do I Take?

Assignment	format	MAS6002	MAS6024	MAS468	Due in (month)
------------	--------	---------	---------	--------	----------------

Assignment	format	MAS6002	MAS6024	MAS468	Due in (month)
1	take-home exam	Yes	No	No	early Oct
2	individual R project	Yes	Yes	Yes	late Oct
3	individual R project	Yes	Yes	Yes	late Jan
4	group presentation	Yes	No	No	late May

Table 1: Assignments by module code

## 2.2 More Information on the Assessments

- Assignment 1 is a take-home exam based on the Foundation Material partially covered in Intro Week. You will receive separate instructions about what is required.
- Assignments 2 and 3 are the statistical/R programming assignments.
- Assignment 4 is based on a group oral presentation that takes place after the exams in the first exam week. You will receive separate instructions about what is required in the middle of Semester 2.

## 2.3 Assessment Feedback

You won't get your original submissions back for *any* assignment so make sure you keep a copy safe. For *all* assignments you'll receive:

- a letter (*F*, *P*, *M*, *D* or *D*<sup>+</sup>) from Table 2 to indicate the overall standard of the work submitted
- generic feedback on Blackboard

For assignments 2 and 3 you'll also receive an individual feedback sheet detailing the strengths and weaknesses in your submitted work.

## 2.4 Scaling

The marks on the individual assignments might be scaled. Those assignments that were considered too easy might be scaled down whilst those that were considered too hard might be scaled up. This scaling is done once all the module marks are available in June.

Grade	Interpretation
<i>D</i> <sup>+</sup>	High distinction ( $\geq 80\%$ )
<i>D</i>	Distinction ( $\geq 70\%$ )
<i>M</i>	Merit ( $\geq 60\%$ )
<i>P</i>	Pass ( $\geq 50\%$ )
<i>F</i>	Fail ( $\leq 50\%$ )

Table 2: Grade interpretation by letter

As a benchmark, typically only around 5% -7% of students get high distinctions.

## 2.5 Late Work and Extensions

Late work will be subject to penalties in line with the University guidance (see the right hand box here (<http://www.shef.ac.uk/ssid/exams/policies>): basically, lose 5% for each day or part of a day for the first 5 days, and then score nothing. Extensions will only be granted for good reason accompanied by suitable evidence when appropriate. If you want to ask for an extension please contact our Kostas Triantafyllopoulos, our Course Director, (see *Guidelines on seeking extensions for coursework*).

## 2.6 Determination of the Overall Module Mark for MAS6002

- Each student will be awarded an overall mark for MAS6002 which will be based on the four individual scores.
- All four assignments have a weighting of 25%.
- A pass in all four assignments guarantees a pass overall.
- If you fail any assignment we will examine your grade profile across all four assignments to determine whether you should pass the module.
- Because of the nature of the assignments particular importance is given to Assignments 2 and 3 when deciding whether you pass the module.
- If your mean score over the four assignments is over 50% but you fail *either* assignment 2 or 3 you **will not necessarily be awarded a pass** overall.

## 2.7 Determination of the Overall Module Mark for MAS6024 and MAS468

- You're overall mark will be the average of the two assignment scores.
- Both assignments have a weighting of 50%.

## 2.8 Return of Module Marks

For MAS468, MAS6024 and MAS6002 the final module mark will be returned after the Semester 2 examiners' meeting in June (there isn't time to return them at the Semester 1 examiners' meeting).

# 3 Information about Assignments 2 and 3

## 3.1 Report Submission Format

For Assignments 2 and 3 you should:

- submit all parts of the assignment together as a **single pdf document**
- prepare your pdf report using R Markdown (you'll be shown how to do this during the module)
- submit it via Blackboard

Your submission will be checked by our plagiarism detection software. For MAS6002 students the report must be named as follows:

- *yourstudentnumber-6002-A2.pdf* for Assignment 2,
- *yourstudentnumber-6002-A3.pdf* for Assignment 3,

Replace 6002 with 6024 or 468 as appropriate.

Note that *yourstudentnumber* is your registration number and should be a 9 digit number starting with 19 (for e.g. 190123456).

MAS6002 students will be given separate instructions for Assignments 1 and 4.

## 3.2 Report Content and Structure

Some important points:

- you should explain what your code does in a way that would be helpful for a typical MSc student (i.e. *R* functions described in these notes don't need further explanation but new ones probably do)
- unnecessarily long reports may be penalized
- we do not want a formal report structure; just specify which part of the question you are answering and provide your answer
- there is an R Markdown template on Blackboard for you to use (highly recommended)
- presentation is important which is why we insist on *R Markdown* (or knitr if you prefer to use that instead).

## 3.3 Assessment Criteria

In this module you are being marked on your ability to:

- program efficiently in R
- communicate what the code does if you feel this is necessary for the reader to understand it
- select and implement appropriate statistical/probabilistic approaches correctly
- draw sensible statistical/probabilistic conclusions from your results
- communicate these conclusions in a concise and technically correct way
- where necessary produce high quality graphics to support your conclusions

## 3.4 Good Practice/Common Mistakes

Here are a few hints to help you avoid making common mistakes in your assignments. Once you have completed the first draft of your assignment check through this list and edit your assignment as appropriate.

- Make sure you have actually answered **all** the questions **completely**.
- If you use any notation in your assignment answer then make sure you define it carefully (e.g. define  $x_{ij}$  to be the number of ....).
- Make sure your registration number is at the top of your submission and that your name is not anywhere in the document.

Many common mistakes relate to plots. Here is a list of things to do/avoid:

- give all figures a number and provide informative captions if appropriate

- give informative axis labels (don't necessarily use *R* default names)
- make sure axis labels are big enough to read easily
- make plots big enough to read easily
- is a legend needed?
- would it help the reader if plots were grouped together or if several plots were combined into a single plot?

## 3.5 Books

There are many user-friendly books available covering all aspects of statistical computing with *R*. Many are quite technical, whilst others are aimed at the beginner. Some are also available for free on-line. I mention some below that you might find useful, either for this module or if you use *R* in your workplace or in further study.

### 1. Books related to general programming in *R*:

- Grolemund (2015) is an excellent book that assumes little knowledge but provides very clear explanations of the main factors to consider when programming in *R* and is reviewed here (<http://blog.revolutionanalytics.com/2015/03/review-of-hands-on-programming-with-r.html>). It is free to access here ([https://d1b10bmlvqabco.cloudfront.net/attach/ighbo26t3ua52t/igp9099yy4v10/igz7vp4w5su9/OReilly\\_HandsOn\\_Programming\\_with\\_R\\_2015.pdf](https://d1b10bmlvqabco.cloudfront.net/attach/ighbo26t3ua52t/igp9099yy4v10/igz7vp4w5su9/OReilly_HandsOn_Programming_with_R_2015.pdf))
- Gillespie and Lovelace (2016) is free here (<https://csgillespie.github.io/efficientR/>)
- Matloff (2011)

### 2. Books related to *R Markdown* and *knitr*:

- Xie (2016) is an excellent and comprehensive introduction and reference to *R Markdown* and *knitr* and is highly recommended. There is an outline of the book and its chapters here (<https://github.com/yihui/knitr-book>)
- Xie, Allaire, and Grolemund (2018) is *the* manual for *R Markdown* and is free here (<https://bookdown.org/yihui/rmarkdown/>)

### 3. Books showing how to produce professional and flexible graphics (including *ggplot*):

- Wickham (2016) teaches the actual 'Grammar of Graphics' that makes *ggplot* so powerful and is highly recommended for those wishing to really understand this grammar
- Chang (2013) shows how to control lots of graphical parameters in a more recipe-like way. Good as a reference book for those wanting quick answers to specific questions

### 4. For help with data manipulation look at:

- Spector (2008)
- Wickham and Grolemund (2016) is a good source of information and examples of using the *dplyr* package

### 5. Those interested in data science more generally:

- Wickham and Grolemund (2016)

### 6. There are several books for very advanced *R* users wanting to know how to combine *R* with much faster object oriented programming languages like C++ or how to use parallel programming in *R*:

- Eddebuettel (2013)
- Matloff (2016)
- McCallum and Weston (2012)

These are not recommended for this module unless you are already a highly proficient programmer. But if you ever need to parallelise your code or use C++ to speed it up then these are for you.

### 7. For advanced *R* users

Wickham (2015).

The name says it all really. A challenging read and not for the beginner!

### 8. If you want to know a little about the history (and future) of *R* see Nick Thieme's article

(<https://rss.onlinelibrary.wiley.com/doi/epdf/10.1111/j.1740-9713.2018.01169.x>) in Significance Magazine (a publication issued by the Royal Statistical Society)

## References

Chang, W. 2013. *R Graphics Cookbook*. O'Reilly Media Inc.

Eddebuettel, D. 2013. *Seamless R and C++ Integration with Rcpp*. Springer.

Gillespie, C., and R. Lovelace. 2016. *Efficient R Programming*. O'Reilly Media Inc.

Grolemund, G. 2015. *Hands-on Programming with R*. O'Reilly Media Inc.

Matloff, N. 2011. *The Art of R Programming*. No Starch Press.

— — —. 2016. *Parallel Computing for Data Science*. CRC Press.

McCallum, E., and S. Weston. 2012. *Parallel R*. O'Reilly Media Inc.

Spector, P. 2008. *Data Manipulation with R*. Springer.

Wickham, H. 2015. *Advanced R*. CRC Press.

— — —. 2016. *Ggplot2 - Elegant Graphics for Data Analysis, Second Edition*. Springer.

Wickham, H., and G. Grolemund. 2016. *R for Data Science*. O'Reilly Media Inc.

Xie, Y. 2016. *Dynamic Documents with R and Knitr*. CRC Press.

Xie, Y., J.J. Allaire, and G. Grolemund. 2018. *R Markdown: The Definitive Guide*. CRC Press.