Jonatan Kallus, kallus@chalmers.se

Sebastian Jobjörnsson, jobjorns@chalmers.se

Olle Nerman, nerman@chalmers.se
Course overview

Course homepage, mailing list

3 blocks

Home assignments after each block, 5 hp for whole course (or 3 hp for blocks 1 and 2)

Dalgaard: Introductory statistics with R

9:00-16:00, Rooms this week: MVH12, except Tuesday 9-13 in MVF26
Course outline

- **Block 1:** R programming, refreshing introductory statistics
  - Monday Feb 27 - Wednesday March 1
- **Block 2:**
  - Wednesday March 8 - Friday March 10
- **Block 3:**
  - Wednesday April 5 - Friday April 7
• Monday: R programming
  • Intro lecture, computer labs 1 and 2
  • Dalgaard chapter 1-2
• Tuesday: Data visualization, probability distributions
  • Intro lecture, computer labs
  • Dalgaard chapter 3-4
• Wednesday: Hypothesis tests, regression, correlation
  • Intro lecture, computer labs
  • Dalgaard chapters 5-6
• Wednesday afternoon (15:15): Intro to homework
  • ANOVA, statistical power, multiple testing, outliers
  • Assignments: computer labs
What is R?

Both a programming language and an environment for statistical computing and visualization

S and S-plus: General statistical software

1995 R is released as open-source software

Ideas:

- Universal availability
- Flexibility
- Extendability
Why is R relevant for scientists?

Rapidly increasing use of advanced mathematical and statistical modeling in science

Other scientists use it for publishing, running and sharing computational methods and algorithms (especially in biology and medicine)

Statisticians use it: immediate access to newly published methods

Reproducibility: R scripts can be stored and run again, harder to reproduce analyses in graphical applications like Excel and SPSS
Main ”competitors” to R: Python and Matlab

Matlab is commercial, R and Python are free (pros and cons: cost, open access, responsibility)

Matlab focused toward engineering, R toward data analysis and statistics, Python is general purpose (web servers, computer games...)

Very similar as languages: easy and convenient to use, slow but use C behind the scenes, (annoyingly) similar syntax
Why so much technical stuff to get started?

Programs feel easy to use quickly when they follow predetermined steps and limit possibilities.

Especially commercial programs must be liked immediately.

The same features that make a program easy to use in the beginning can become cumbersome and limiting.

But: the flexibility of R is only valuable if you learn enough to be able to use it.
Brief introduction to labs 1 and 2

Live demo on projector

Rest of the day: work on labs 1 and 2 (preferably in pairs)

Me and Sebastian are here to help
Overview of 1st lab

• Calculator
• Objects
• Functions
• Vectors and matrices
• Getting help in R
Overview of 2nd lab

- The workspace
- History
- Source files and scripting
- Data frames
- Importing and exporting data
- Packages
- R as a programming language
If you have time left

Read in Dalgaard and do exercises

Get acquainted with the built-in R help, see its ”Introduction to R”

Be creative: experiment with what you have learnt so far