Oxford Genomics CENTRE

UNIVERSITY OF OXFORD

Developing R Programming Skills

18th-19th October 2018

Helen Lockstone and Ben Wright

Bioinformatics Core

wellcome centre human genetics

The Wellcome Trust Centre for Human Genetics

UNIVERSITY OF OXFORD

Developing R Programming Skills

- Two-day workshop style course – self-paced and tailored to level of background knowledge
- Main course tutorial https://tinyurl.com/ya9ce5ub
- Additional material at:
  http://www.well.ox.ac.uk/bioinformatics/training/Developing_R_Skills_Oct2018

- Key Course Aims
  - Introduce the R software through practical sessions
  - Help make R and associated resources accessible to novice programmers
  - Emphasise good programming practice

Housekeeping

- Part of teaching for DPhil programme in Genomic Medicine and Statistics at WHG, opened out to Medical Sciences Division

- No fire alarm test due – if alarm sounds, please leave by main building door
- Tea/coffee provided at 11am and lunch available from the canteen (12.30-1.30)
- Wrap up around 3.30 each day

Introductory Remarks

Why learn R?

- The ability to handle and analyse large-scale datasets is, and will continue to be, a key skill in modern biological research, as well as many other fields
- Generating data far outstrips our ability to interpret and make sense of it – and it is still hard to recruit good bioinformaticians
- R has become a key programming language for genomics due to the diverse set of packages available through BioConductor https://www.bioconductor.org/

The downsides to R

- R is not an intuitive language to learn, even for those who are familiar with programming concepts
- It can take many months to start to feel comfortable writing your own R code, so be prepared for some investment of time, hard work and a steep learning curve
- R's extensive functionality and versatility are its main attributes but also the reason it can be hard to know where to begin…

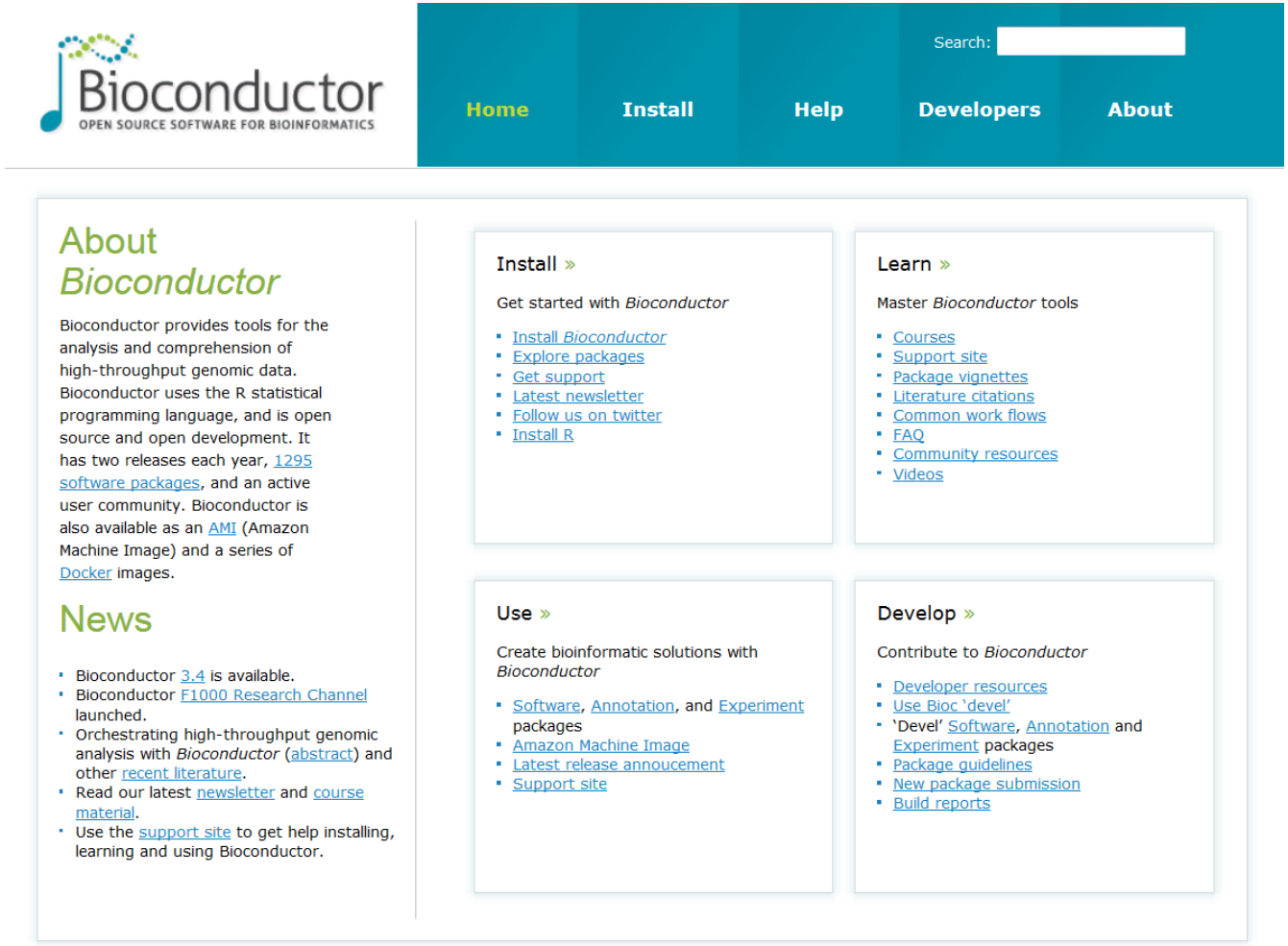
Tips for Successful Programming

- Logical thinking and problem solving skills are vital
  - Decide what steps needed to solve a task
  - Troubleshooting error messages
  - Testing code to ensure it does what it should
- Consistency, accuracy and attention to detail
  - Easy to make unintentional mistakes (which R may well execute with no warning message)
  - Check what code is doing at every step
  - Mentally predict what should happen, so you can spot potential errors
- Accept that it can be a frustrating process!

What is R?

- R is a powerful software package for statistical analysis and also a high-level programming language
- Originally written in the 1990s for teaching statistics by Ross Ihaka and Robert Gentleman at the Department of Statistics of the University of Auckland
- It's open-source, available for free for Win, Mac, Linux and regularly updated (maintained by R Core development team)
- http://www.r-project.org/index.html
- Has become a mainstream research tool
  - Users have contributed hundreds of add-on packages (CRAN)
  - Bioconductor resource is invaluable for genomic data https://www.bioconductor.org/

Bioconductor



Bioconductor packages extend functionality of R to all aspects of handling, processing and analysing genomic data

https://www.bioconductor.org/

Getting started with R

- Some aspects of R are analogous to tasks you might routinely perform in Excel:
  - sorting/filtering data
  - finding a particular occurrence of text
  - simple data summary statistics and tests
  - plotting graphs

- R provides far more complex functionality besides, particularly for statistical modelling/analysis and data visualisation. It can also perform small useful tasks with a simple line or two of code (where there may not be a quick way to do the equivalent in Excel) e.g. finding the overlap of two lists of genes

Getting started with R

- For those without a programming background, the biggest challenge might be getting used to how the R language is structured, and working with objects to store and manipulate data
- Imagine that instead of using Excel, we are giving R step-by-step instructions of what we want to do (e.g. filter a column for all values <0.05):
  - Our data could be created in Excel and read into R (like a table); it can be stored in an appropriate object (named by the user)
  - We'd need to specify the column of interest somehow (R has more than one way to do this) and set the condition 'values less than 0.05'
  - Finally we'd have to decide whether to display the output or save it in a new variable
- Although some aspects of R can be quite unintuitive and take some time to become familiar with, they provide the flexibility that makes the software so powerful.

Getting started with R

- R is extremely extensive and versatile, quite possibly no two people use it in the same way. So how best to get started?'

- Hopefully this course will help, and R itself it very well documented with a large and active user community (mailing lists, online forums like stack overflow etc) – usually googling any R problem will find relevant threads.

- Learn by doing – follow examples in the manuals or start by running scripts written by others to understand what the code does before moving on to modifying code or writing your own scripts from scratch.

- It's impossible to remember the details of all R functions (or even know about all of them!) – make use of the help pages to check syntax, arguments, examples of usage etc

Some general advice

- A computer will do exactly what you tell it to do and you need to tell it every little step, in the right order
  - Imagine writing instructions that someone can follow to produce a particular result, including every tiny detail they would need
  - Akin to instructions for an experimental protocol - important to be very precise
- Comment your code as much as possible – if you need to revisit it at a later date, it will make much more sense!

- Check and double check (and check again) that your code is doing what you ***think*** it is doing – R has some default behaviours that may not always be realised, and can lead to problems if not spotted
  - Inspect the objects created, their length, contents and so on.
  - Manually check a few elements of the output to be sure they are correct

Problem solving

- You will inevitably run into problems when you are programming - try solving them by:
  - reading the error message
  - reading the help file
  - make a list of what could possibly be wrong and check these possibilities one by one
  - break down a complex step into smaller, simpler steps (this is also a useful way to build up a more complex piece of code)
  - If you really cannot solve a problem, remember there are usually alternative ways in R to achieve the same goal (using a related function for example)
  - Google the problem – many helpful forums/lists
  - If all else fails, post a question on the relevant help list (but be sure to read the guidance first!)

R documentation and help

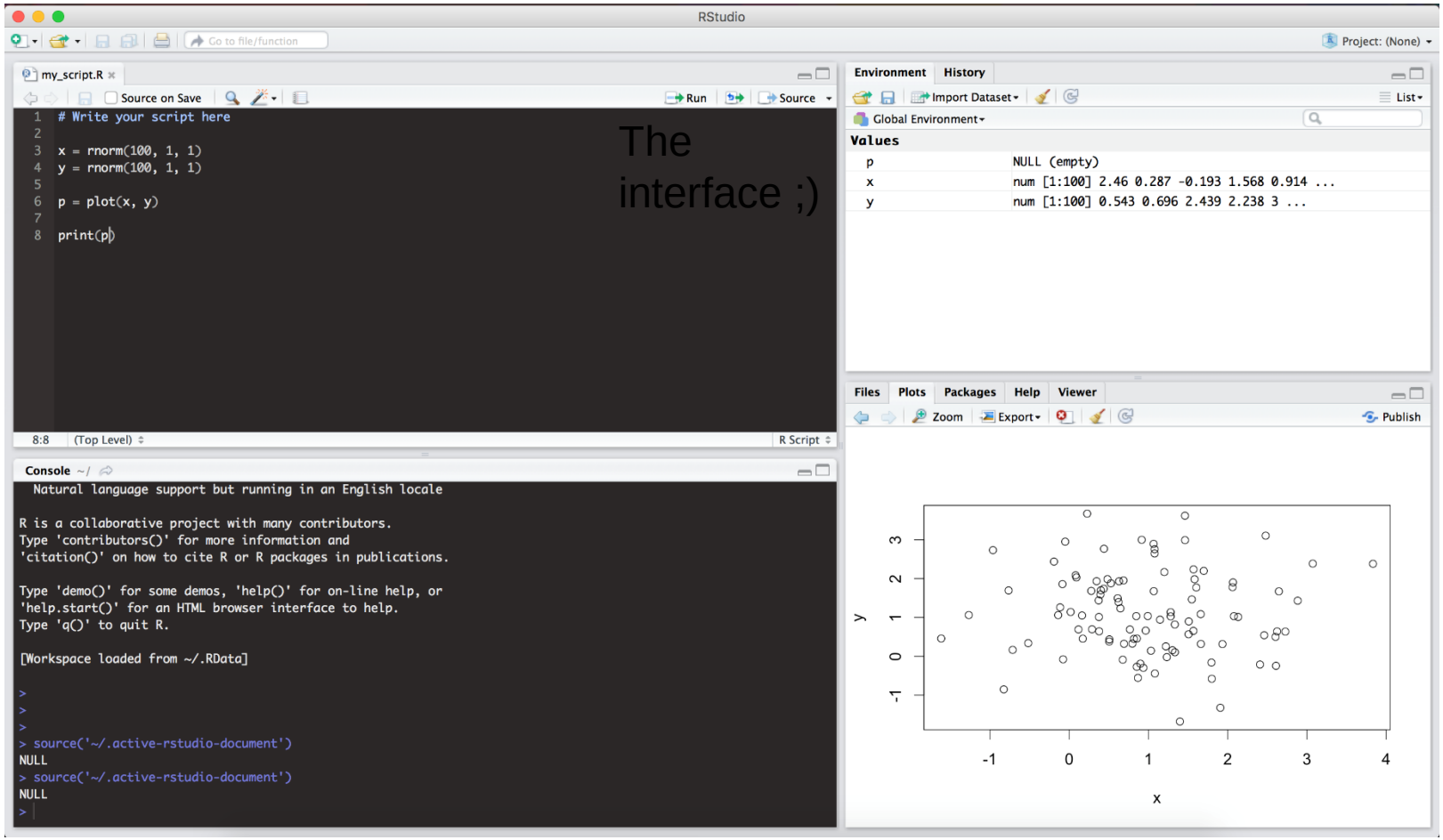
- `> help.start()`
  - Most useful if you don't know which command to use
  - Starts a Search Engine in your favourite browser
  - Can get this also from the R GUI Help Menu
- If you do know which command to use but need to check details
- `> ?plot`
- `or equivalently`
- `> help(plot) # help("plot") in RStudio`
  - Brings up the help page on the function 'plot'
- From the R GUI Help menu or the R website, you can get Manuals (in PDF)
  - An Introduction to R
  - R Reference manual

R documentation and help

- Further reading
  - Michael J Crawley (2005) Statistics: An Introduction using R. Wiley: Chichester, England
    - This is an excellent book, and well worth working through
    - Also has an associated web page with datasets, exercises etc
    - http://www.imperial.ac.uk/bio/research/crawley/statistics/
  - A selection of recommended online resources, tutorials, local courses and textbooks are collated at: https://help.it.ox.ac.uk/courses/R
  - R-help and BioConductor mailing lists
  - Stack Overflow

https://www.rstudio.com

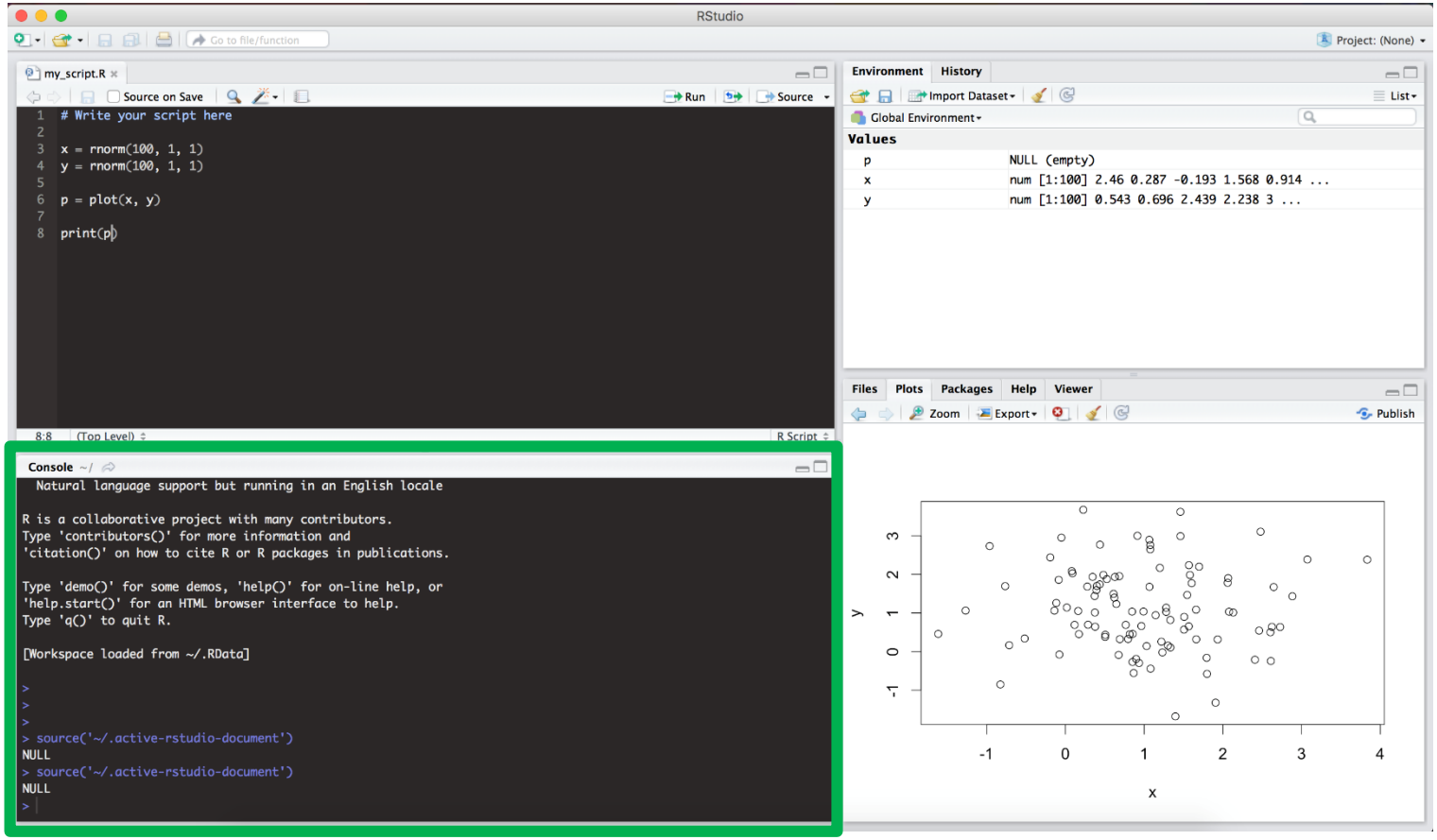
RStudio Interface



An interactive and easy-to-use interface with many features that make working with R easier

Slide courtesy of Quentin Ferry

RStudio Interface

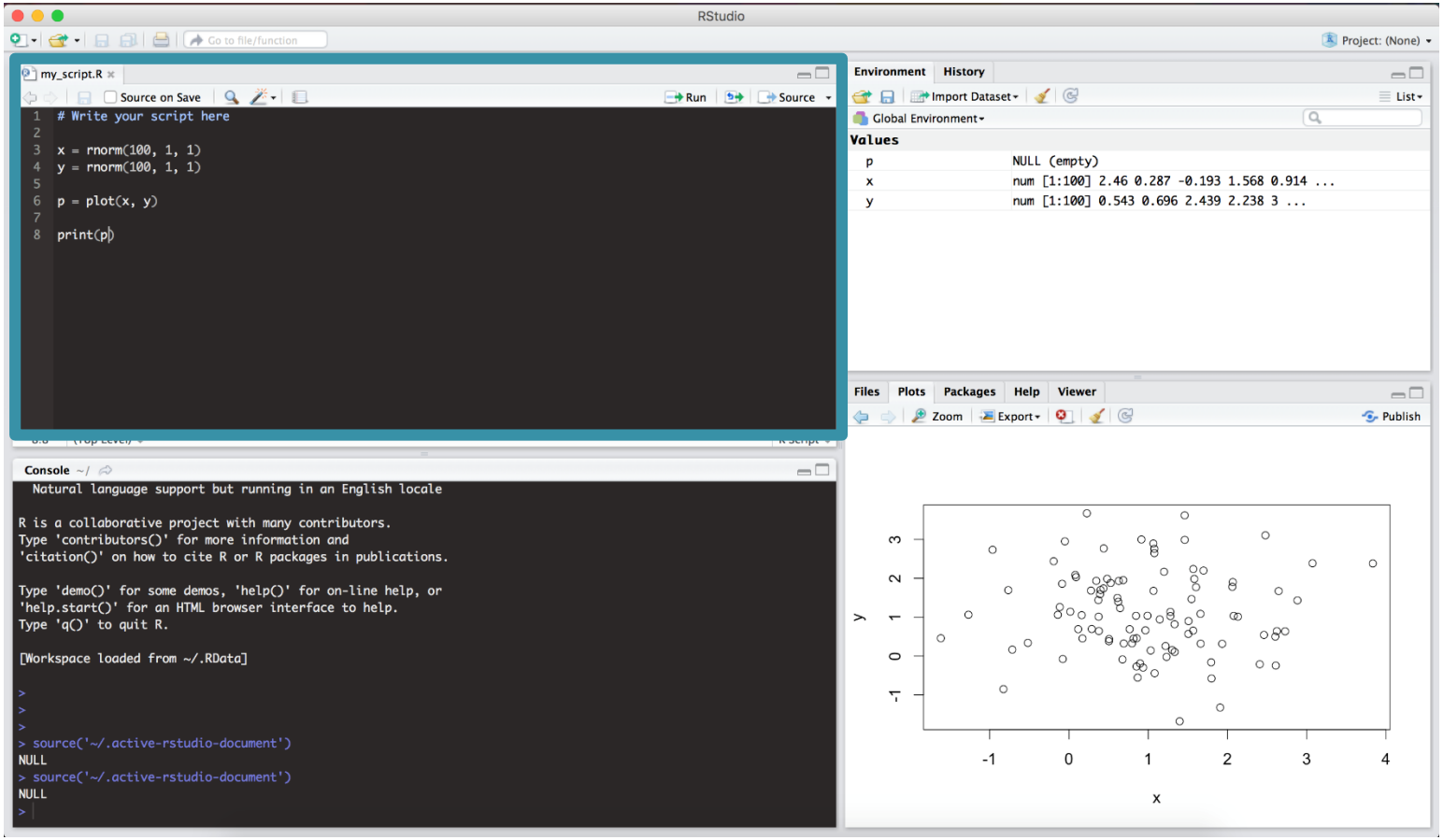


Console for entering R commands

Slide courtesy of Quentin Ferry

RStudio Interface

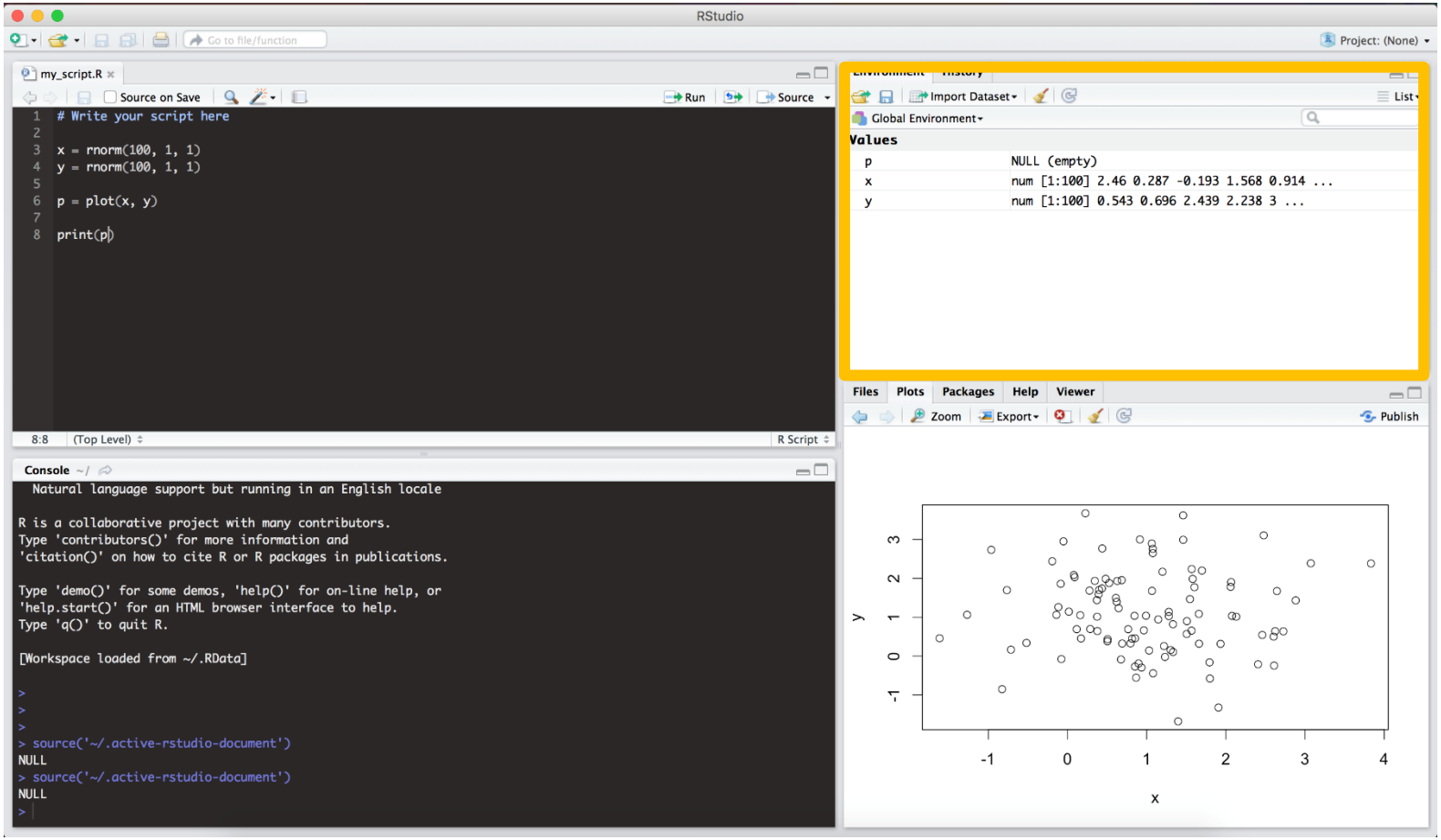
Script, data tables, etc…



Slide courtesy of Quentin Ferry

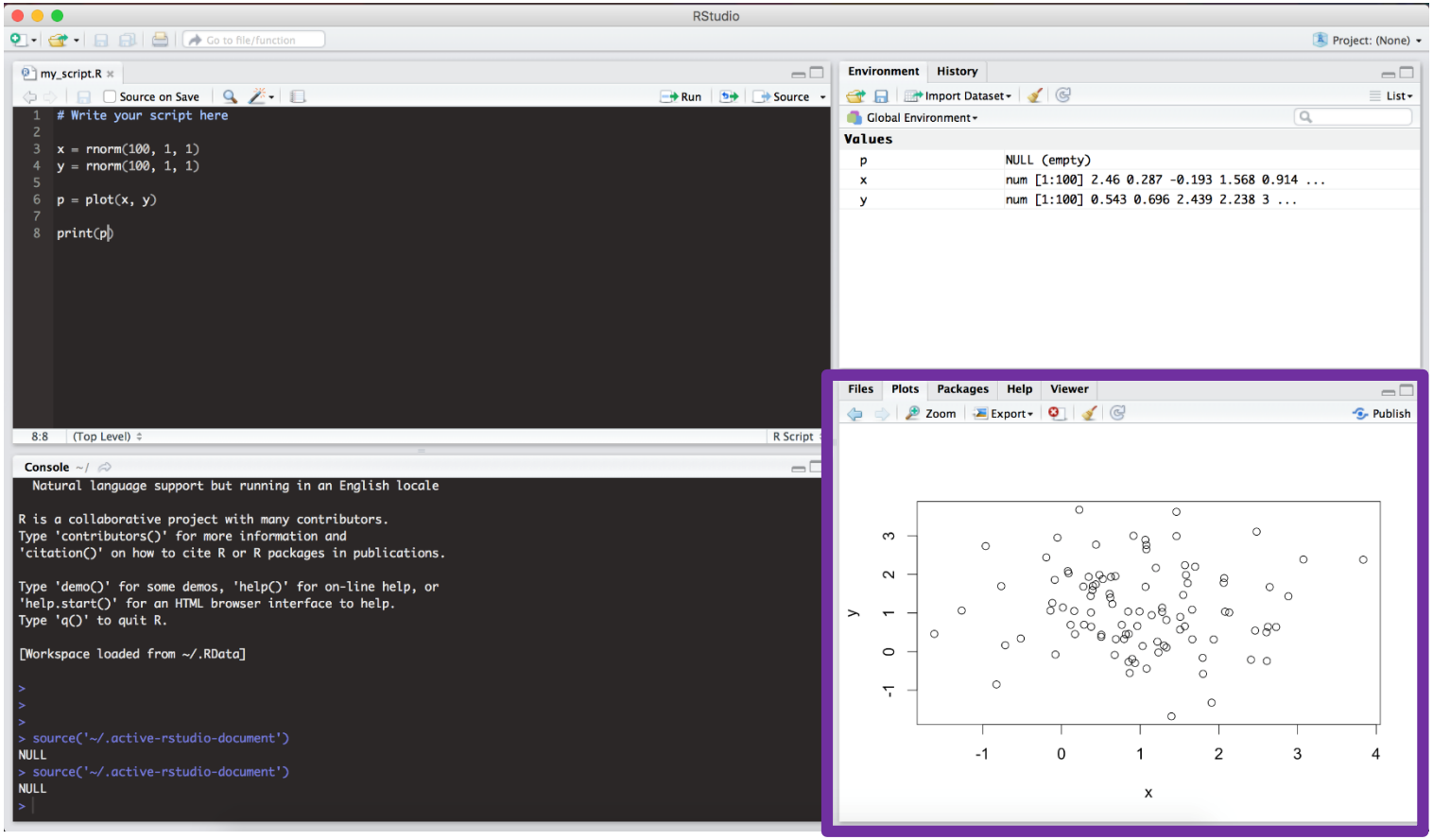
RStudio Interface

Clickable list of variables in memory



Slide courtesy of Quentin Ferry

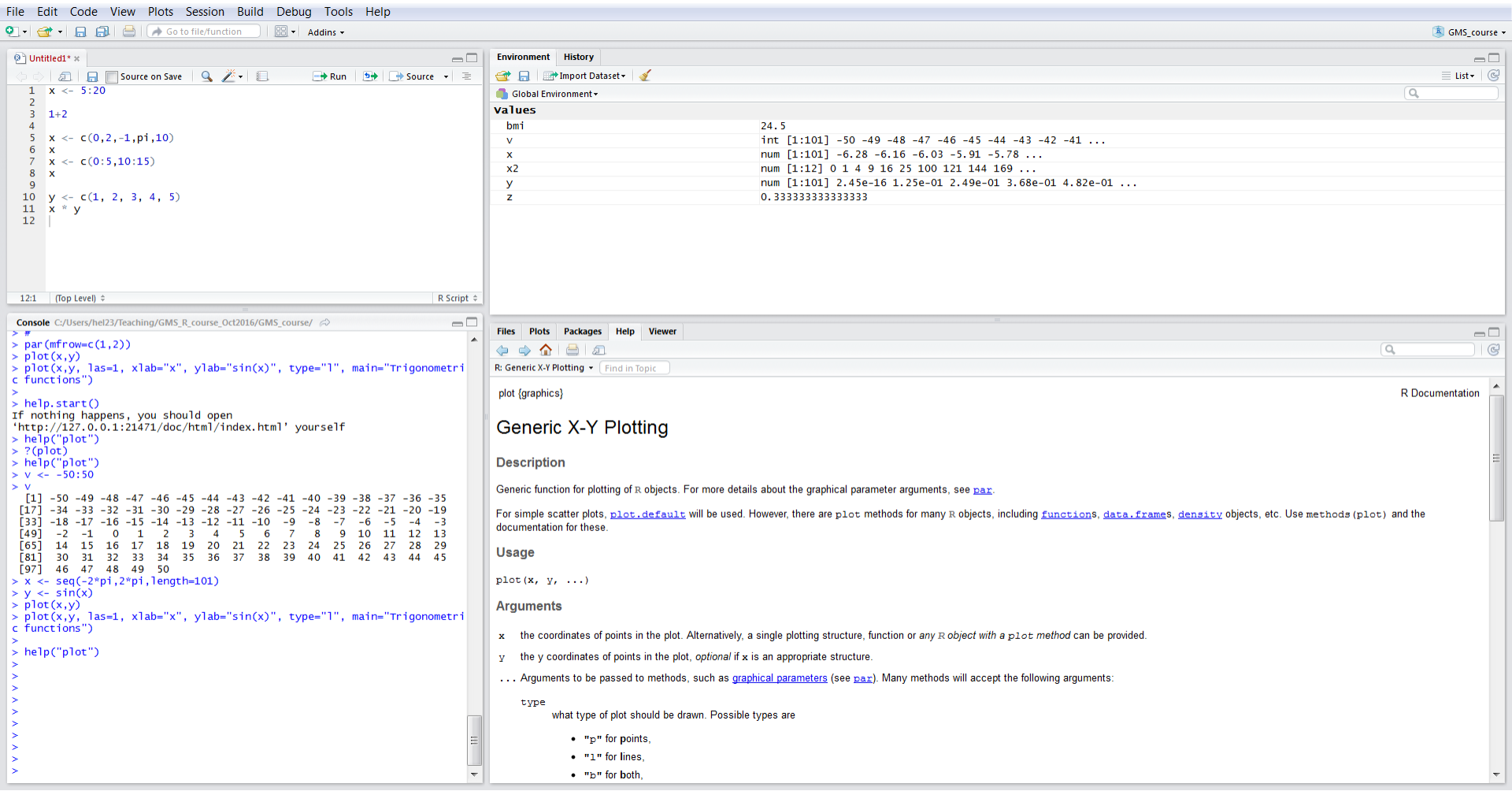
RStudio Interface



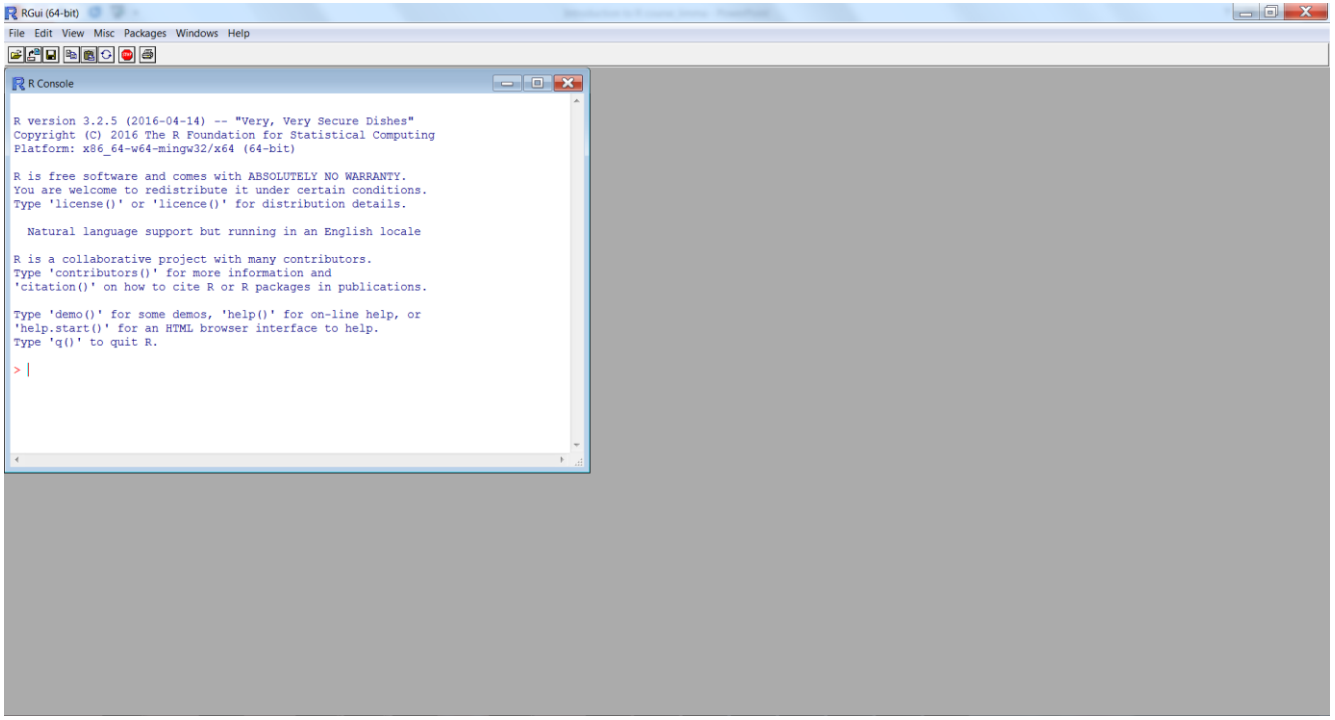
Plots, Files, Packages, help etc…

Slide courtesy of Quentin Ferry

RStudio on Windows



The R interface (Windows)



- From here we can start working in R and do thinks like:
  - Check what packages are already installed
  - Install new packages
  - Check the current working directory or change to a new one
  - Access the help options
- As always, there are several ways to do things