

Approaches to science & statistical inference

ZOL 851

Sept 6 2016

Goals for today

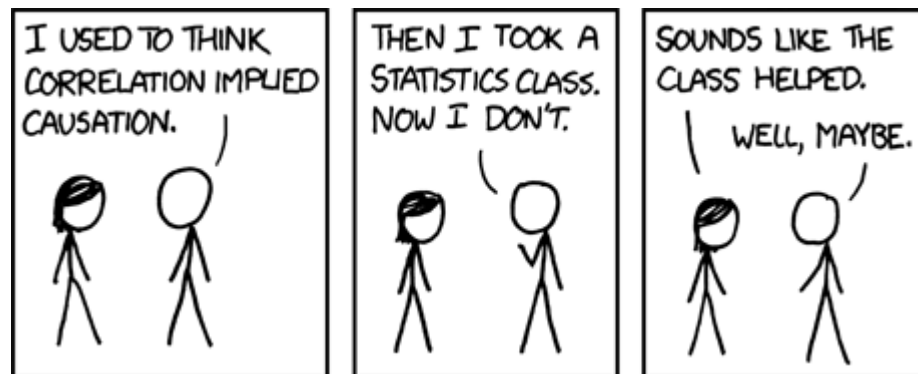
- Overview of scientific & statistical approaches
- Factors influencing experimental designs
- Different methods of categorizing data
- Introduction to R

Where does statistical inference fit into a scientific research program?

- Statistical inference is about providing a quantitative & mathematical formalism to the ideas & approaches you take to science
- Without an understanding of the approach we take to science, how can the hypotheses we generate and tests we perform be statistically useful?

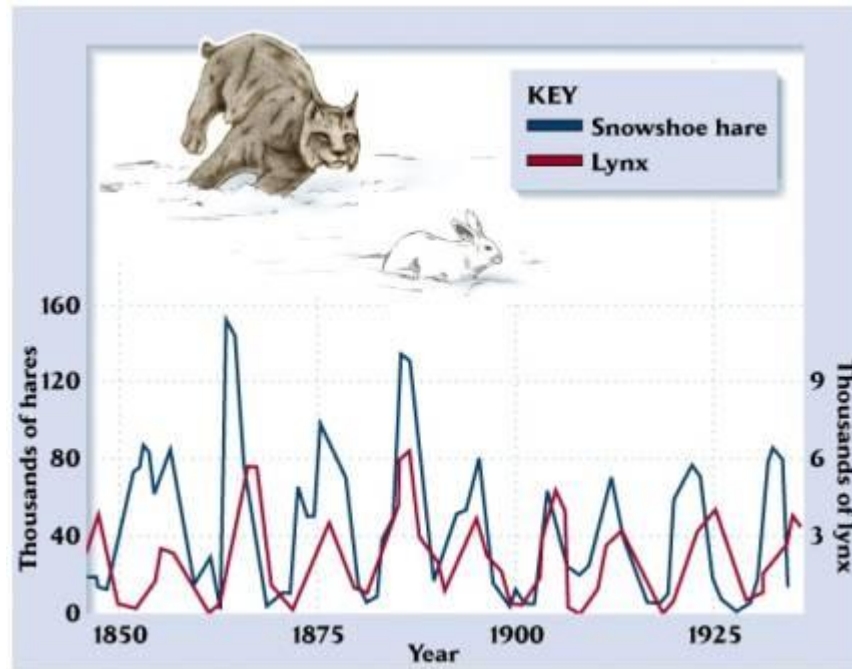
Scientific approaches

- Experimentation vs. observation
- Correlation is not causation
 - Associations can be completely unrelated
 - Due to some unmeasured 3rd variable
- Experiments as mechanism
 - Not always possible in ecology



Scientific approaches

- When experimentation is not possible, the study of natural patterns can get at mechanism



What is statistical inference?

- Attempt to evaluate a set of probabilistic hypotheses about the behavior of some data-generating mechanism
- 3 approaches
 - Bayesian
 - Likelihoodist
 - Frequentist

What is statistical inference?

- All 3 approaches use likelihood functions, where the likelihood function for a datum E on a set of hypotheses H is
 - $\Pr(E|H)$: the probability of E given H
- **Likelihoodists** use them to characterize data as evidence
 - $L = \Pr(E|H_1)/\Pr(E|H_2) > 1$
- **Bayesians** use them to update probability distributions
- **Frequentists** use them to design experiments that will generally perform well in repeated applications



Motivating example

- In 1980s, infants showing a particular pattern of respiratory problems had a ~20% survival rate until researchers developed a new therapy ECMO
 - ECMO led to 72% survival of the first 100 patients tested
- Despite early success, conventional standards (i.e. frequentist approaches) required the team to perform a randomized clinical trial using ECMO & conventional treatments side-by-side



Motivating example

- Concerned about continuing to use a seemingly inferior treatment, the team used a 'randomized play the winner' trial design
 - Result: all 11 infants given ECMO survived; 1 given conventional therapy died
- Approach still didn't meet the standard, so 2nd trial designed



Motivating example

- Still concerned about the ethics, the team designed a trial with 2 phases: it would be randomized until 4 patients died on either treatment
 - Result: 28/29 given ECMO survived; 6/10 given conventional therapy died
- This too failed to meet efficacy standards



Motivating example

- 3rd randomized trial conducted by separate team
 - Had to be terminated when early results clearly indicated ECMO superiority
 - Resulted in 54 more infant deaths under conventional therapy
- Illustrates the costs of failing to reach a consensus on an approach to statistical inference
 - Rigid application of frequentist approach in this case

Bayesian vs. frequentist inference

- A simple analogy:
- You've misplaced your phone somewhere in your house. You use a friend's phone to call it & it starts ringing somewhere—where should you search?

Frequentist

- I have a mental model that helps me identify the area from which the sound is coming. So upon hearing the ring, I infer the area of my house I should search.

Bayesian

- Apart from the mental model, I also know the locations where I've misplaced the phone in the past. So, I combine inferences using the ring & my prior info to identify an area to search.

What is a p-value?

- The p-value is used throughout frequentist statistics—from t-tests to regression analyses
- You use p-values to determine statistical significance in a hypothesis test
- But it's a slippery concept—how do you correctly interpret p-values?

What is a p-value?

1. $P(D|H_0)$: probability of observing the data given that the null hypothesis is true
2. $P(D|H_1)$: probability of observing the data given that the alternative hypothesis is true
3. $P(H_0|D)$: probability of the null hypothesis being true given the data
4. $P(H_1|D)$: probability of the alternative hypothesis being true given the data

What is a p-value?

- Need to understand **null hypotheses** to understand p-values
- In every experiment, there is an effect or difference between groups that are being tested
- There is always a possibility that there is no effect, or no difference between groups: **null hypothesis**

What is a p-value?

- Imagine an experiment for a treatment that you know is ineffective (e.g., use of tap vs. distilled water to grow a certain species of plant)
- We know the null hypothesis is true (no difference between plant growth of two groups)
- But it's possible that you will actually observe an effect just by random sampling error
- **Null hypothesis** should be interpreted as: the observed difference in the **sample**—which does **not** necessarily reflect a true difference between populations

What is a p-value?

- A low p-value suggests that your sample provides enough evidence that you can reject the null hypothesis for the entire population
- P-values address only 1 question: **how likely are your data, assuming a true null hypothesis?**
 - P-values do **not** measure support for the alternative hypothesis
 - P-values are **not** the error rate

What is a p-value?

- While a low p-value indicates that your data are unlikely assuming a true null, it cannot evaluate which of these 2 competing cases is more likely:
 - Null is true but your sample was unusual
 - Null is false

What is a p-value?

- Going back to the water/plant example, assume your experiment obtained a p-value of 0.04
- **Correct** interpretation: Assuming water treatment had no effect on plant growth, you'd obtain the observed difference or more in 4% of studies due to random sampling error
- **Incorrect**: If you reject the null hypothesis, there's a 4% chance you're making a mistake

Statistics & the scientific method

- How do we incorporate statistics into scientific reasoning?
- Fundamentally, statistics are statements of probability
- A p-value is a statement about the probability $P(\text{Data} \mid H_0)$

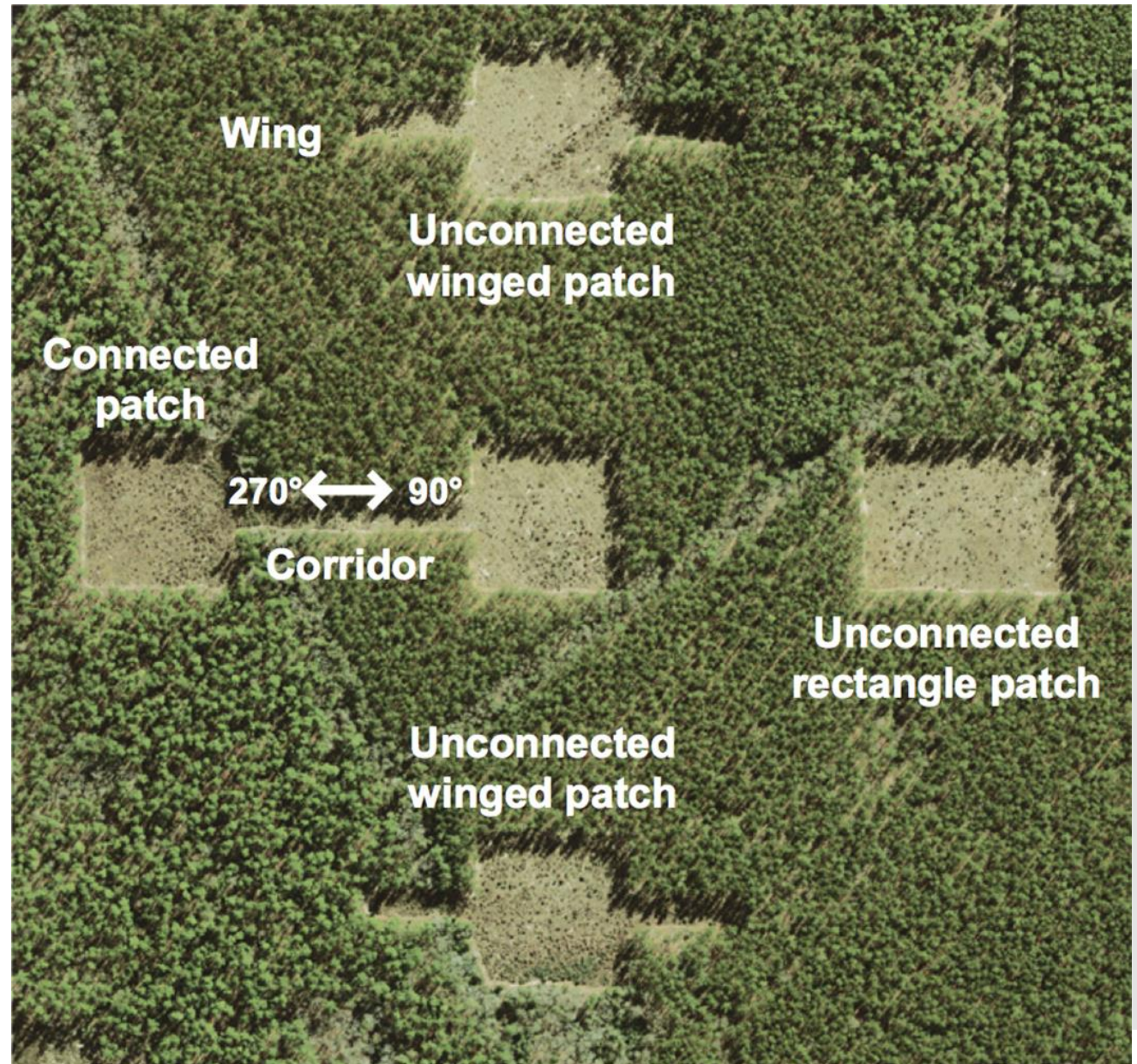
Experimental design

- Deliberately imposing a treatment on a group of objects/subjects in the interest of observing a response
- Need to design the experiment such that the right type of data is generated to answer the questions of interest
- Attempt to identify known sources of variability

Experimental design

- Randomization
 - Most reliable method to reduce bias by creating homogeneous treatment groups
 - 1. Completely randomized design
 - 2. Randomized block design
 - Subjects first divided into homogeneous blocks before being randomly assigned to group

Experimental design



Experimental design

- Replication
 - Repetition of an experiment on a large group of subjects
 - Reduces variability
 - Increases significance & confidence in results

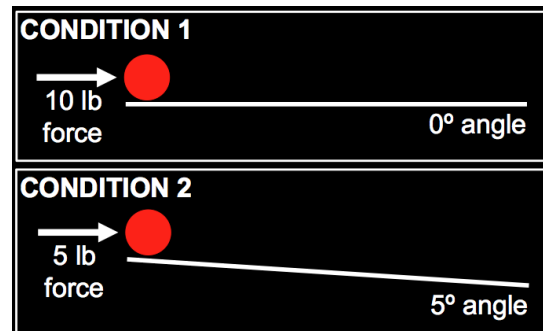


Experimental design

- Confounding factors
 - Some sources of variation are considered 'nuisance' factors that contribute to variability
 - Examples?
 - Age, sex, observer experience
 - Solution: sort subjects into blocks before randomization

Experimental design

- Multifactorial design
 - Testing one factor per experiment is insufficient and inefficient
 - Multiple factors allow for exploration of interactions
 - Some factors may be blocking factors or confounding variables
 - Extraneous variable that correlates with both the dependent and independent variable

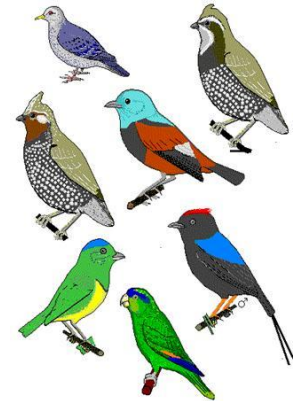


Variables

- Outcome variables
 - Or dependent variables
 - The response a treatment is meant to influence
- Explanatory variables
 - Or independent variables
 - The predictors that are either manipulated or thought to affect the outcome
 - Can have interactions between predictor variables

Variables

- Quantitative: continuous vs. discrete
 - Examples?
- Categorical: nominal vs. ordinal
 - Examples?



Intro to R



Overview

1. Why R?
2. Getting started: steep learning curve
3. The basics
4. R interface
5. How to download
6. Intro to R & R Studio programming

Why R?

- It's free!
- It runs on a variety of platforms, including Windows and MacOS
- Provides an unparalleled platform for programming new statistical methods in a straightforward way
- It has state of the art graphics capabilities

R has a steep
learning
curve

- Don't feel intimidated!
- Much of the advanced functionality of R comes from hundreds of **user-contributed packages**
- Hunting for what you want can be time consuming
- Can be difficult to get a clear overview of what procedures are available

R has a steep learning curve

- Rather than setting up a complete analysis all at once, the process is much more **interactive**
- You run a command, process the results through another command, and repeat
- Because of this, R is very **flexible and powerful** for statistical analysis

Advantages & disadvantages

Advantages

- Fast & free
- State of the art
- Active user community
- Forces you to *think* about your analysis
- Excellent for simulation, programming, computer-intensive analyses

Advantages & disadvantages

Disadvantages

- Not user-friendly at the start
- No commercial support; figuring out correct methods on your own can be frustrating
- Working with large datasets is limited by RAM

Tutorials

- All of the following are in PDF format:
 - P. Kuhnert & B. Venables, [An Introduction to R: Software for Statistical Modeling & Computing](#)
 - J.H. Maindonald, [Using R for Data Analysis and Graphics](#)
 - W.J. Owen, [The R Guide](#)
 - W.N. Venables & D. M. Smith, [An Introduction to R](#)
- Use rseek.org instead of google for R-related help/searching

The basics

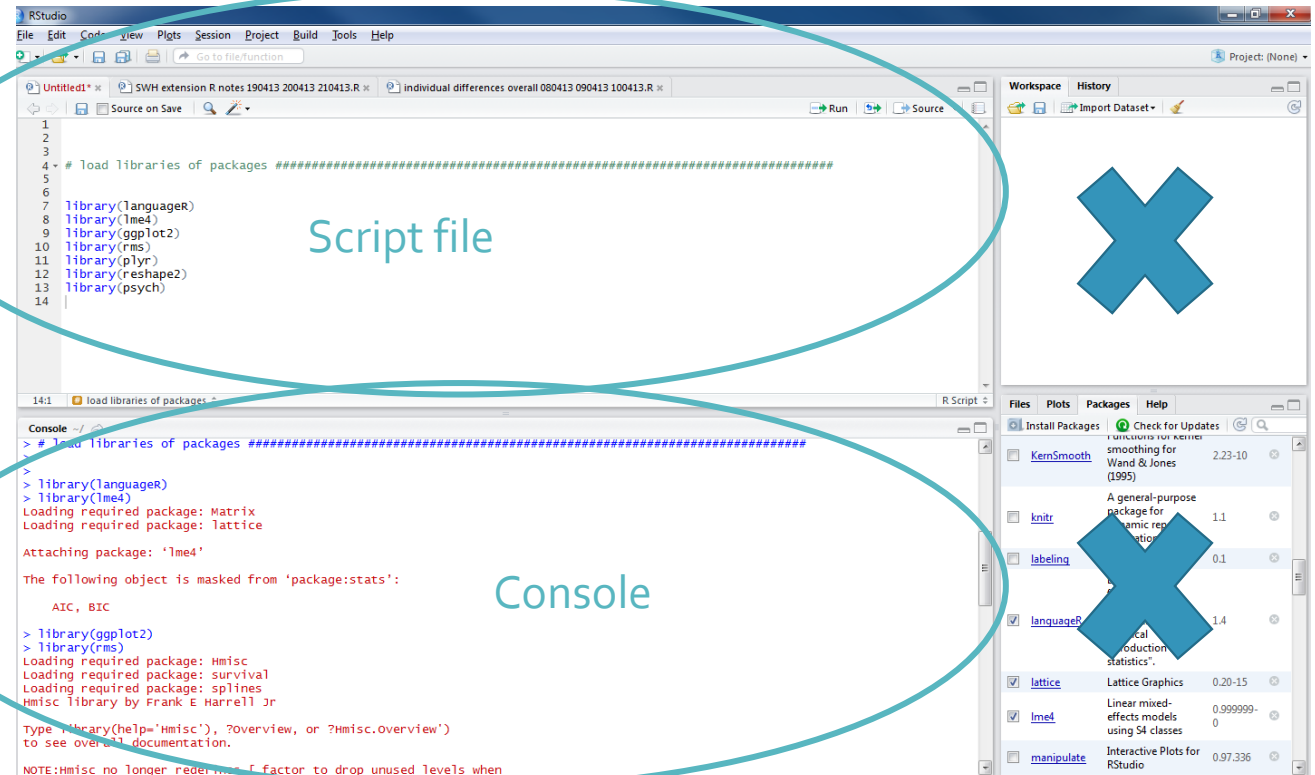
- There is a wide variety of data types, including vectors, dataframes, matrices & lists
- Most functionality is provided through built-in and user-created functions
- All data objects are kept in memory during an active session
- Basic functions available by default
- Other functions are contained in separate 'packages' to be attached to the current session as needed

The basics

- A key skill to using **R** effectively is learning how to use the built-in help system
 - Just type `help.search` or `??` followed by a command
- A fundamental design feature of **R** is that the output from most functions can be used as input to other functions

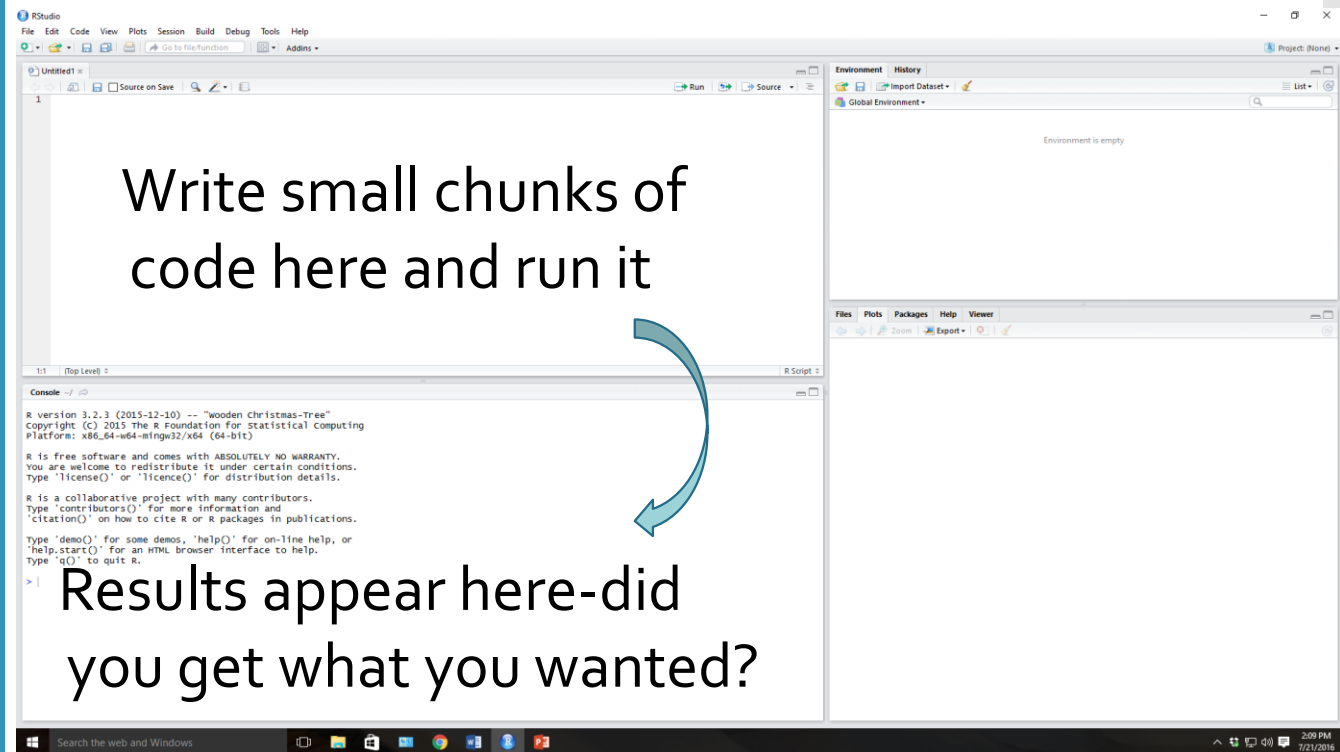
Interface: RStudio

- 2 important windows
 - Script file(s) that will be saved
 - Console that displays output and temporary input (usually unsaved)



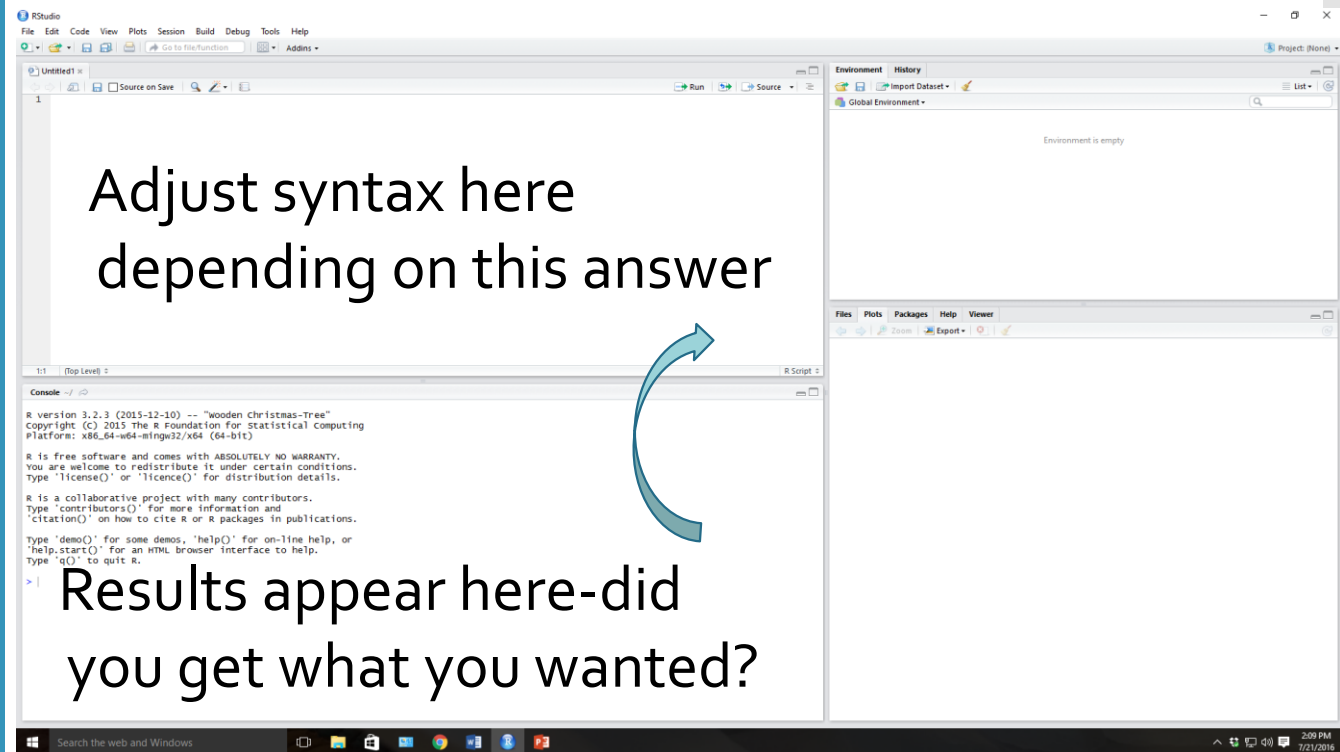
Interface: RStudio

- R sessions are interactive



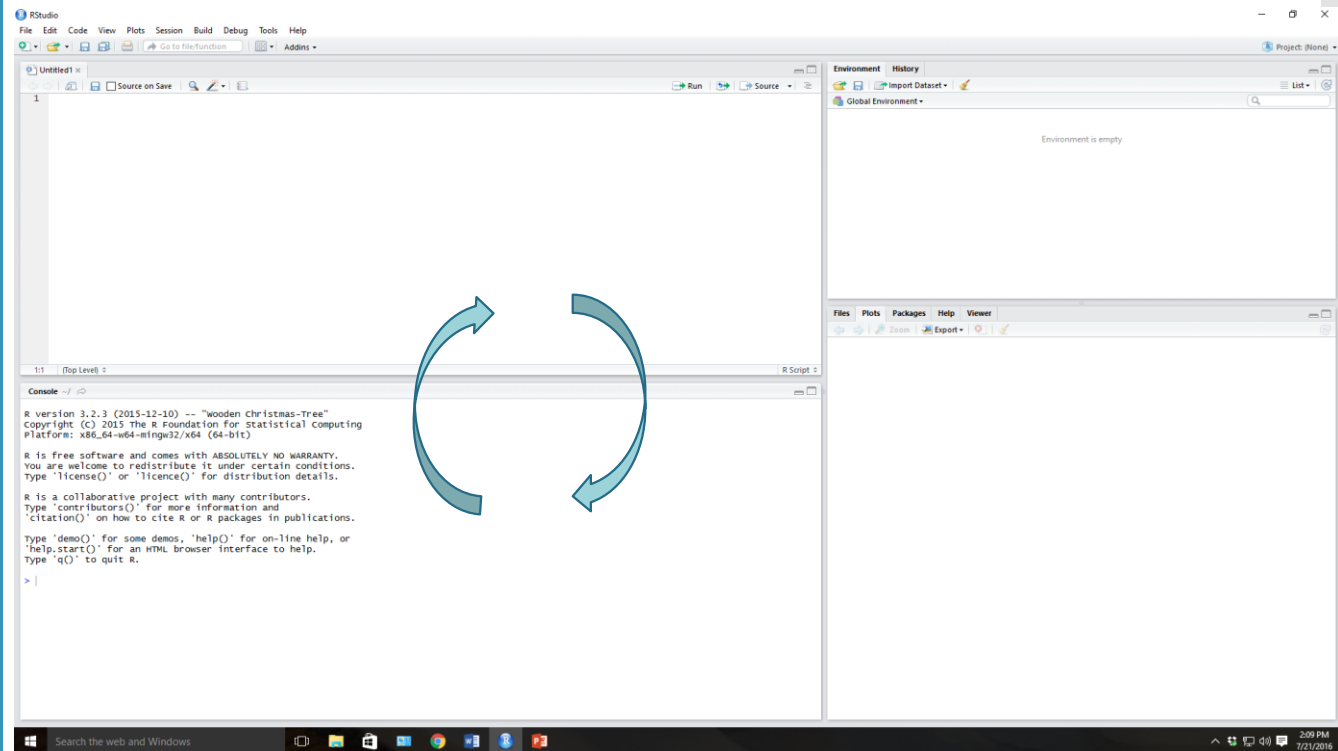
Interface: RStudio

- R sessions are interactive



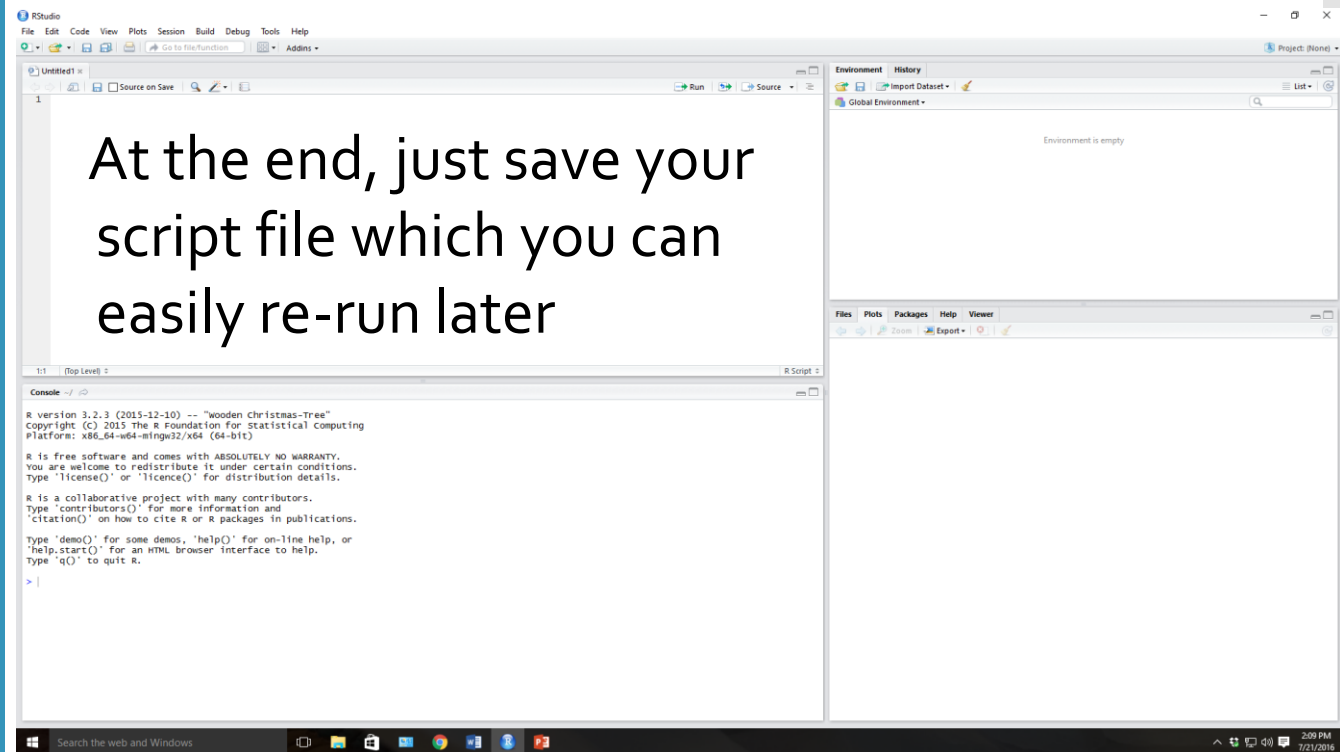
Interface: RStudio

- R sessions are interactive



Interface: RStudio

- R sessions are interactive



Interface: RStudio

- Results of calculations can be stored in **objects** using the assignment operators:
 - An arrow (<-) formed by a smaller than character and a hyphen without a space!
 - The equal character (=)
- Almost all things in R (functions, datasets, results) are **objects**

Interface: RStudio

- **Script** can be thought of as a way to make objects
- Your goal is usually to write a script that, by its end, has created the objects (e.g. statistical results) and graphics you need

Interface: RStudio

- These **objects** can then be used in other calculations
- To print the object just enter the name of the object
- There are some restrictions when giving an object a name:
 - Object names cannot contain 'strange' symbols like `!`, `+`, `-`, `#`.
 - A dot (`.`) and an underscore (`_`) are allowed
 - Object names can contain a number but cannot start with a number
 - R is **case sensitive**, `X` and `x` are two different objects

R workspace

- Objects that you create during an R session are held in memory
- The collection of objects that you currently have is called the **workspace**
- This workspace is not saved on hard drive unless you tell R to do so
- This means that your objects are lost when (1) you close R and do not save the workspace or (2) your system crashes on you during a session

R workspace

- When you close the R console window, the system will ask if you want to save the workspace image
- If you select to save the workspace image, then all the objects in your current R session are saved in a file `.RData`
- This is a binary file located in the working directory of R, which is by default the installation directory of R

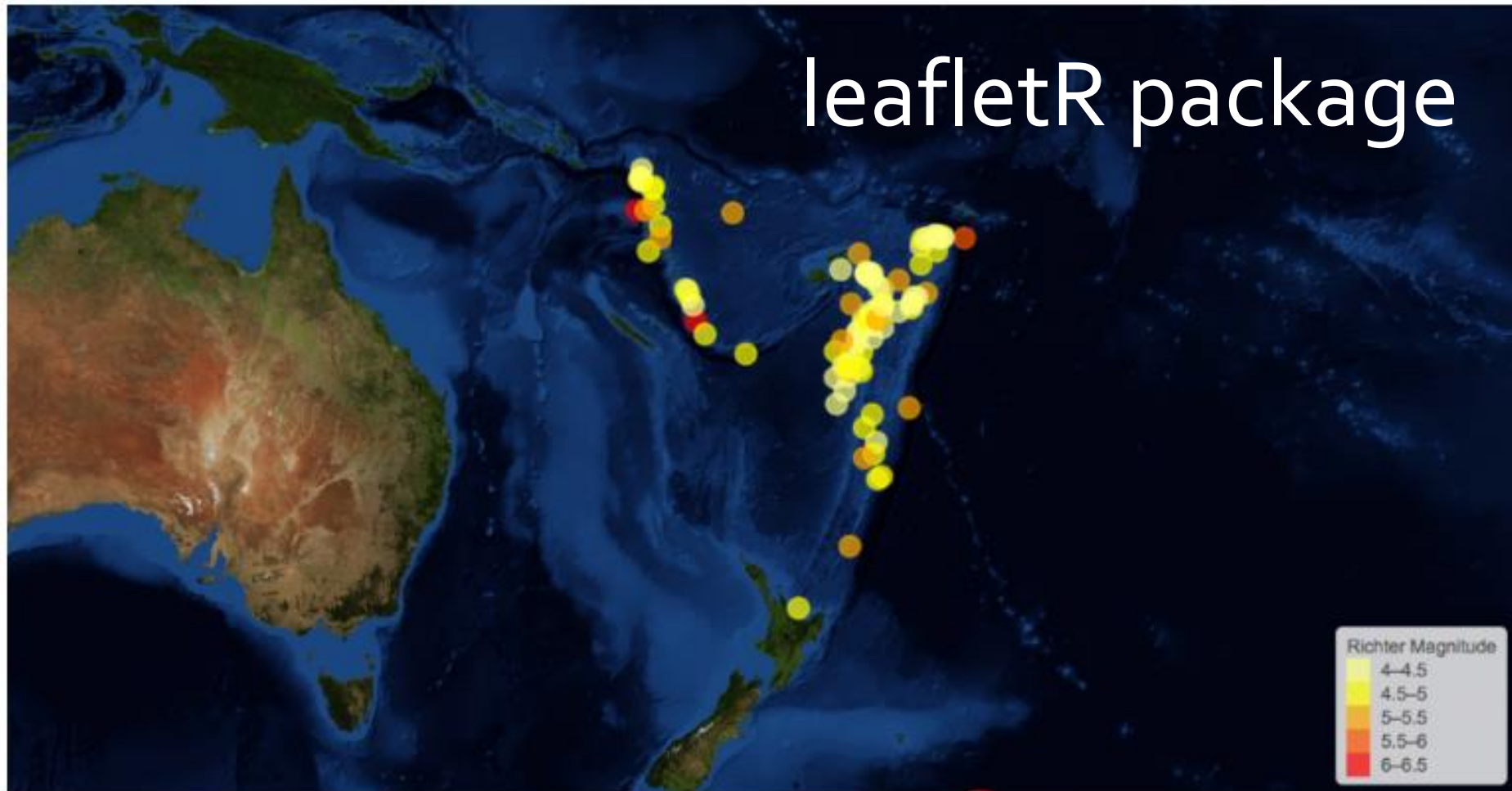
R workspace

- If you have saved a workspace image and you start R the next time, it will restore the workspace
- So all your previously saved objects are available again

R packages

- There is an active R user community and many R packages have been written and made available on CRAN for other users
- Just a few examples: there are packages for
 - Portfolio optimization, drawing maps, exporting objects to html, time series analysis, spatial statistics
 - And on and on...

leafletR package



R packages

- Some basic packages are auto downloaded when you downloaded R
- In the future, you'll find you need certain packages that aren't installed and we'll go through how to download and use them when the time comes

Built-in functions

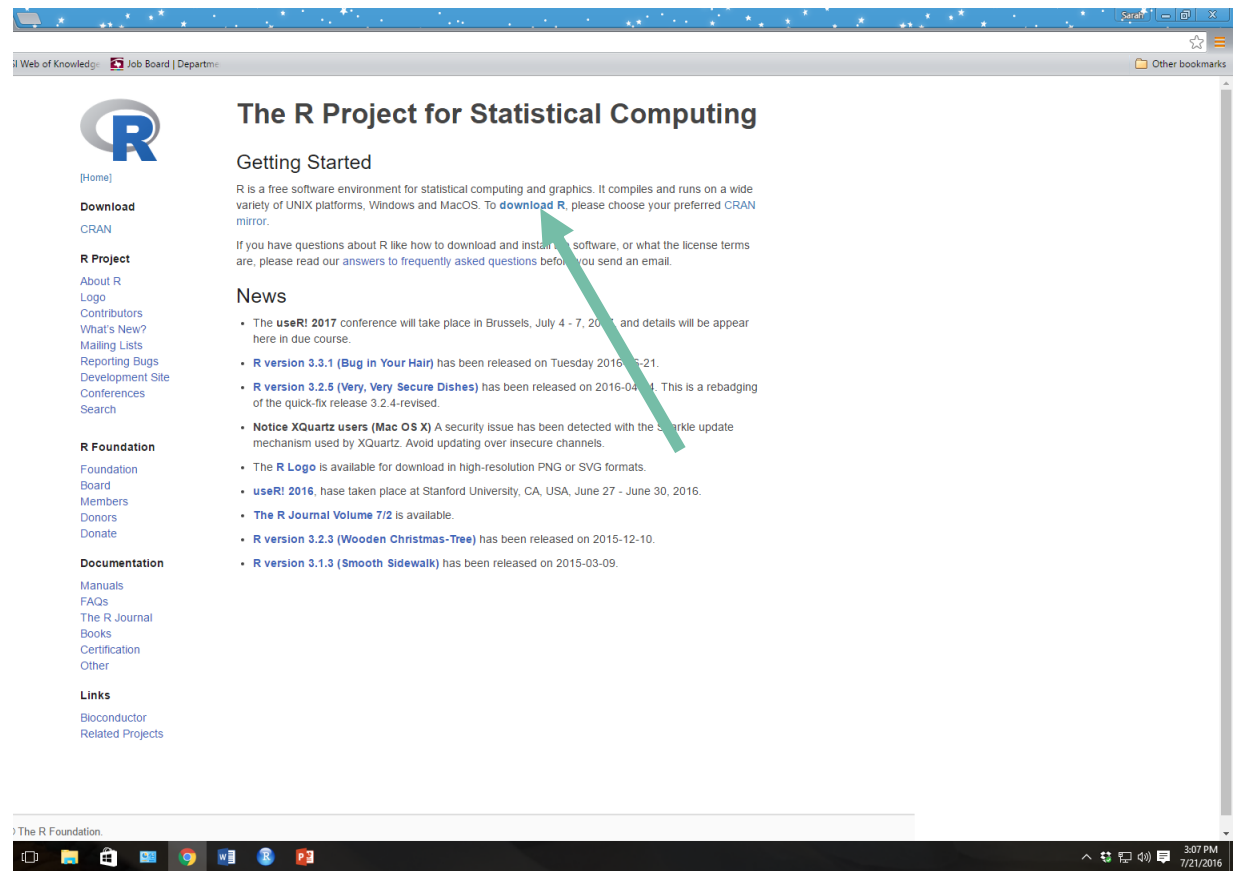
- R has many built in functions that compute different statistical procedures
- Functions in R are followed by ()
- Inside the parenthesis we write the object (vector, matrix, array, dataframe) to which we want to apply the function

Vectors, arrays, matrices

- **Vectors** are variables with one or more values of the same type
- **Arrays** are numeric objects with dimension attributes
- A **matrix** is a two dimensional array

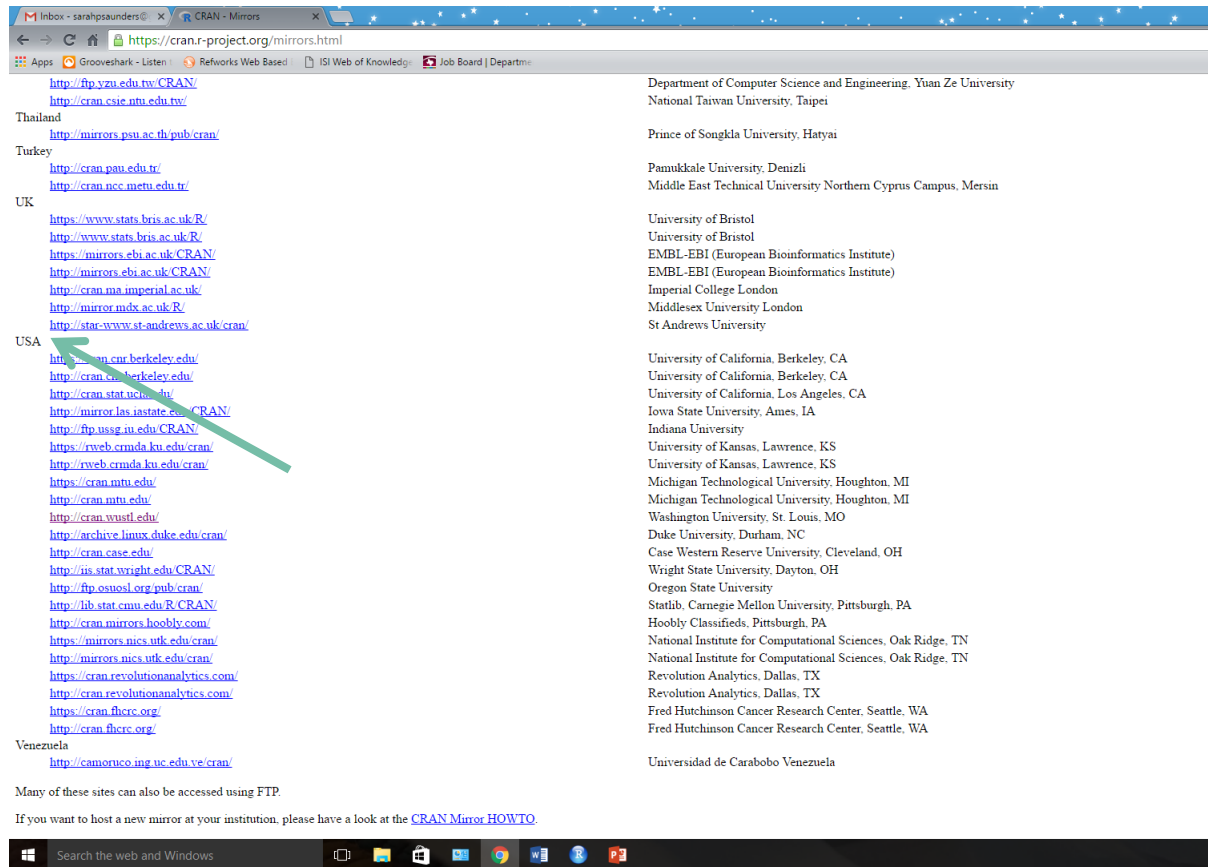
Downloading R

- To install R on your Mac or PC, go to <http://www.r-project.org/>



Downloading R

- Select CRAN mirror



The screenshot shows the CRAN Mirrors website with a list of mirrors categorized by country. A green arrow points to the USA section. The list includes mirrors for various countries and institutions, such as the University of California, Berkeley, and the University of Michigan.

USA

- <http://cran.cnr.berkeley.edu/>
- <http://cran.berkeley.edu/>
- <http://cran.stat.ucd.edu/>
- <http://mirror.las.iastate.edu/CRAN/>
- <http://ftp.usg.m.edu/CRAN/>
- <https://rweb.crmda.ku.edu/cran/>
- <http://rweb.crmda.ku.edu/cran/>
- <https://cran.mtu.edu/>
- <http://cran.mtu.edu/>
- <http://cran.wustl.edu/>
- <http://archive.linux.duke.edu/cran/>
- <http://cran.case.edu/>
- <http://iis.stat.wright.edu/CRAN/>
- <http://ftp.osnori.org/pub/cran/>
- <http://lib.stat.cmu.edu/R/CRAN/>
- <http://cran.mirrors.hoobly.com/>
- <https://mirrors.nics.utk.edu/cran/>
- <http://mirrors.nics.utk.edu/cran/>
- <https://cran.revolutionanalytics.com/>
- <https://cran.revolutionanalytics.com/>
- <https://cran.flirc.org/>
- <http://cran.flirc.org/>

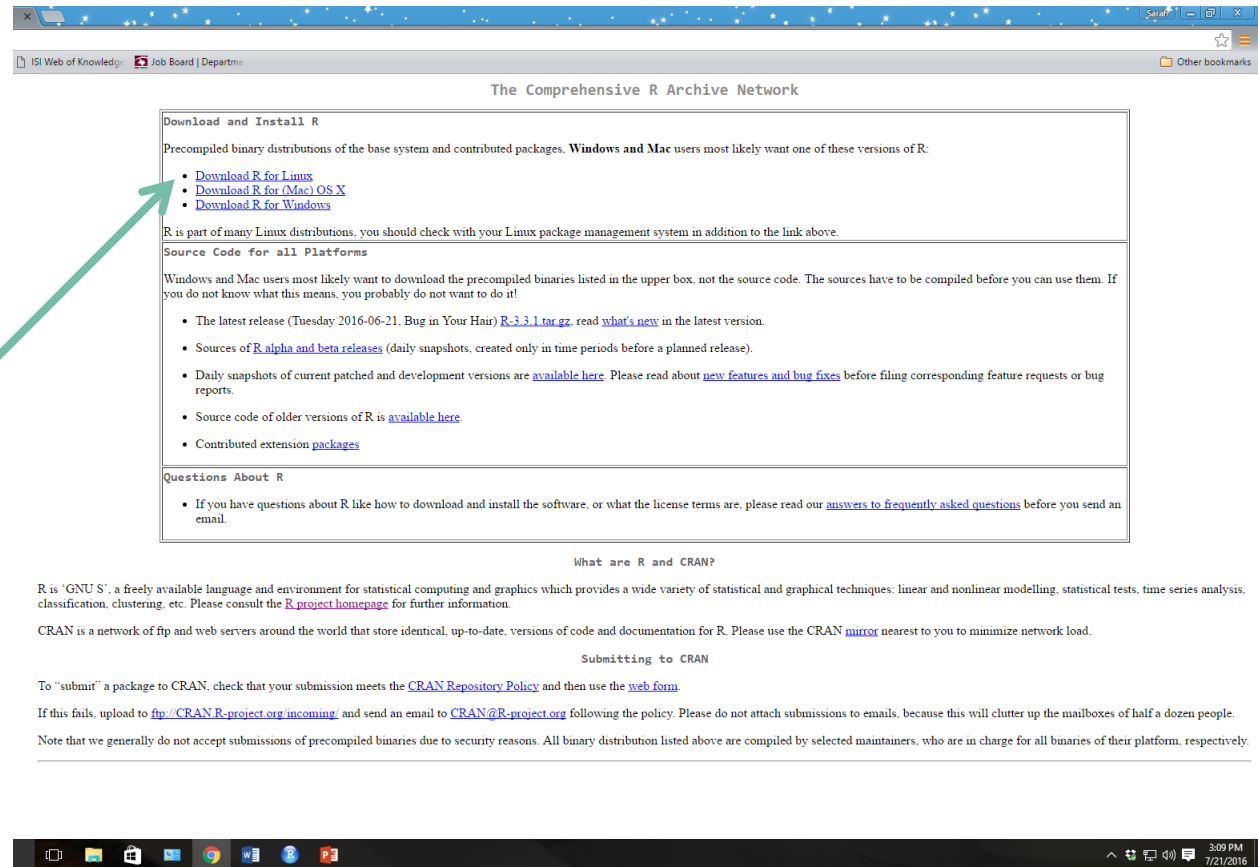
Venezuela

- <http://camomico.ing.uc.edu.ve/cran/>

Many of these sites can also be accessed using FTP.

If you want to host a new mirror at your institution, please have a look at the [CRAN Mirror HOWTO](#).

• Select your operating system



The screenshot shows a web browser window displaying the 'The Comprehensive R Archive Network' page. The page is titled 'Download and Install R' and provides instructions for downloading R. A green arrow points to the download links for Linux, Mac, and Windows. The page also includes sections for source code, frequently asked questions, and information about CRAN.

Download and Install R

Precompiled binary distributions of the base system and contributed packages. **Windows and Mac** users most likely want one of these versions of R:

- [Download R for Linux](#)
- [Download R for \(Mac\) OS X](#)
- [Download R for Windows](#)

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (Tuesday 2016-06-21, Bug in Your Hair) [R-3.3.1.tar.gz](#), read [what's new](#) in the latest version.
- Sources of [R alpha and beta releases](#) (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are [available here](#). Please read about [new features and bug fixes](#) before filing corresponding feature requests or bug reports.
- Source code of older versions of R is [available here](#).
- Contributed extension [packages](#)

Questions About R

- If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

What are R and CRAN?

R is 'GNU S', a freely available language and environment for statistical computing and graphics which provides a wide variety of statistical and graphical techniques: linear and nonlinear modelling, statistical tests, time series analysis, classification, clustering, etc. Please consult the [R project homepage](#) for further information.

CRAN is a network of ftp and web servers around the world that store identical, up-to-date, versions of code and documentation for R. Please use the CRAN [mirror](#) nearest to you to minimize network load.

Submitting to CRAN

To "submit" a package to CRAN, check that your submission meets the [CRAN Repository Policy](#) and then use the [web form](#).

If this fails, upload to [ftp://CRAN.R-project.org/incoming/](#) and send an email to CRAN@R-project.org following the policy. Please do not attach submissions to emails, because this will clutter up the mailboxes of half a dozen people.

Note that we generally do not accept submissions of precompiled binaries due to security reasons. All binary distribution listed above are compiled by selected maintainers, who are in charge for all binaries of their platform, respectively.

Downloading R

Downloading RStudio

- Once R is downloaded, install Rstudio (a nicer interface to use with R)
- <https://www.rstudio.com/products/rstudio/download/>



Products Resources Pricing About Us Blog

RStudio is a set of integrated tools designed to help you be more productive with R. It includes a console, syntax-highlighting editor that supports direct code execution, as well as tools for plotting, history, debugging and workspace management.

If you run R on a Linux server and want to enable users to remotely access RStudio using a web browser [please download RStudio Server](#).

Do you need support or a commercial license? [Check out our commercial offerings](#)

RStudio Desktop 0.99.903 — Release Notes

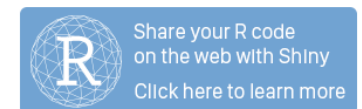
RStudio requires R 2.11.1 (or higher). If you don't already have R, you can download it [here](#).

Installers for Supported Platforms

Installers

RStudio 0.99.903 - Windows Vista/7/8/10
RStudio 0.99.903 - Mac OS X 10.6+ (64-bit)
RStudio 0.99.903 - Ubuntu 12.04+/Debian 8+ (32-bit)
RStudio 0.99.903 - Ubuntu 12.04+/Debian 8+ (64-bit)
RStudio 0.99.903 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (32-bit)
RStudio 0.99.903 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (64-bit)

Size	Date	MD5
77.1 MB	2016-07-18	716f28f2143c5e21f4acea5752e284f8
60 MB	2016-07-18	d14a1585b5a5ac0839507b9c04d460d6
81.6 MB	2016-07-18	761eae80b0ba4d4cd9051a802a2c44e2
88.3 MB	2016-07-18	98ea59d3db00e0083d3e4053514f764d
81 MB	2016-07-18	ce2ea1023d99175cb909def0fe66eba7
81.9 MB	2016-07-18	152f247255e86904cf3354afbc7b3b99



Intro to R programming

- Let's go to RStudio