

R Tutorial: Introduction

R is a programming language and environment for **statistical computing and graphics**. R has all the tools of a commercial statistical package, like SPSS, Systat or SAS, but is COMPLETELY FREE! Personally, I wish I had discovered R much earlier than I did. I use it for correlations, making box plots, ANOVAs and dozens of other statistical tests.

R also comes with many other packages (libraries) that you can easily add on for fancier statistics, phylogenetic analysis and many other things.

This tutorial teaches how to use R for basic statistical analyses, starting with how to download and read in data sets.

Explanation: What is R? <http://www.r-project.org/>

R Tutorial: Installation

To install R, go to the R website: <http://www.r-project.org/>



The R Project for Statistical Computing

[\[Home\]](#)

Download

[CRAN](#)

R Project

[About R](#)

[Logo](#)

[Contributors](#)

[What's New?](#)

[Reporting](#)

[Bugs](#)

[Development](#)

[Site](#)

[Conferences](#)

[Search](#)

R Foundation

[Foundation](#)

[Board](#)

[Members](#)

[Donors](#)

[Donate](#)

Getting Started

R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. To **download R**, please choose your preferred [CRAN mirror](#).

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.

News

- [The R Journal Volume 9/1](#) is available.
- [R version 3.4.1 \(Single Candle\)](#) has been released on Friday 2017-06-30.
- [R version 3.3.3 \(Another Canoe\)](#) has been released on Monday 2017-03-06.
- [The R Journal Volume 8/2](#) is available.
- [useR! 2017](#) (July 4 - 7 in Brussels) has opened registration and more at <http://user2017.brussels/>
- Tomas Kalibera has joined the R core team.

Click the download link.

R Tutorial: Installation

Choose the closest CRAN Mirror site for download.



USA

<http://cran.st-andrews.ac.uk/cran/>

St Andrews University

<http://cran.cnr.Berkeley.edu>

University of California,
Berkeley, CA

<http://cran.stat.ucla.edu/>

University of California, Los
Angeles, CA

I choose USA: UCLA because I am in southern California.



CRAN
[Mirrors](#)
[What's new?](#)
[Task Views](#)
[Search](#)

The Comprehensive R Archive Network

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:

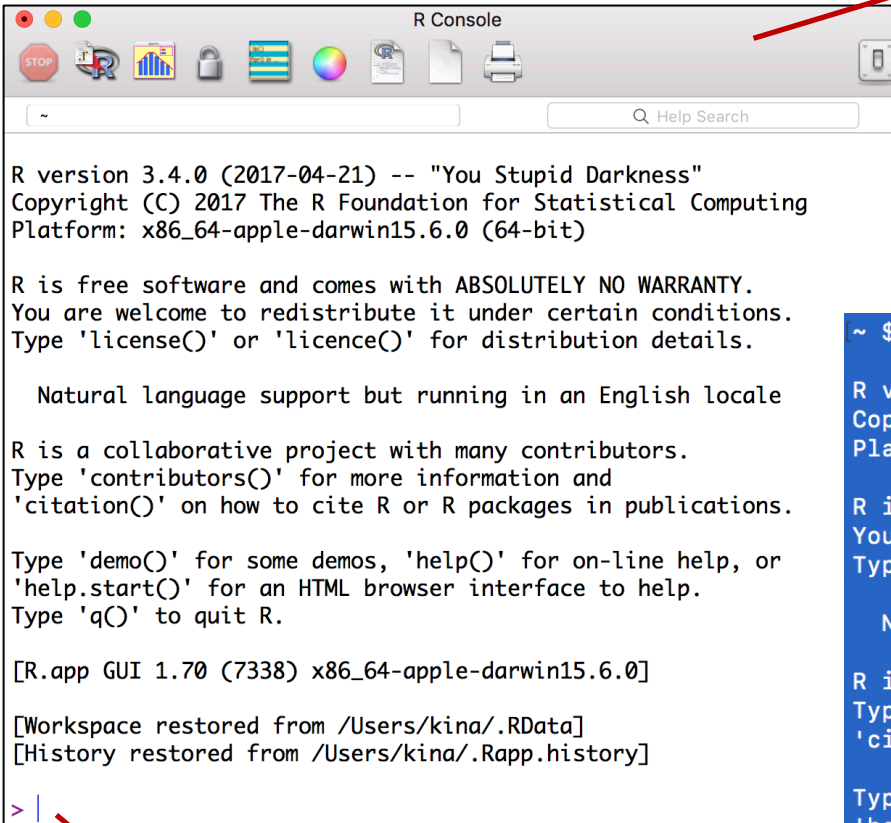
- [Download R for Linux](#)
- [Download R for \(Mac\) OS X](#)
- [Download R for Windows](#)

And I have a Mac so this is my link

R Tutorial: Opening the R Console

Double Click the R Icon and you should get a window that looks like this:

This is the R Console where you run commands.



```
R version 3.4.0 (2017-04-21) -- "You Stupid Darkness"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin15.6.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

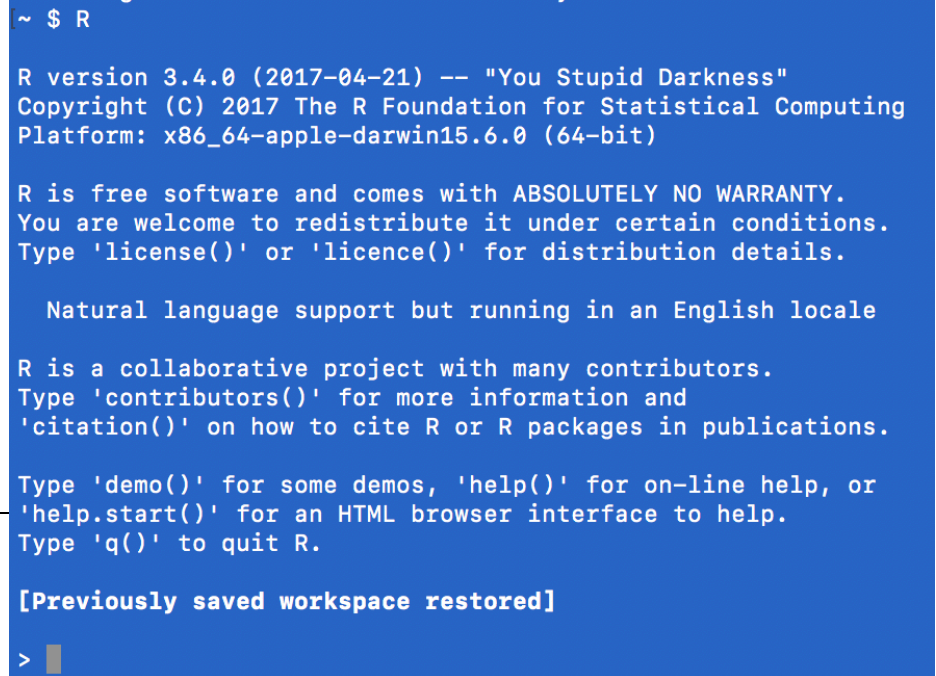
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[R.app GUI 1.70 (7338) x86_64-apple-darwin15.6.0]
[Workspace restored from /Users/kina/.RData]
[History restored from /Users/kina/.Rapp.history]

> |
```

If you are using a linux shell, just type R at the shell prompt like this:



```
~ $ R

R version 3.4.0 (2017-04-21) -- "You Stupid Darkness"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin15.6.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

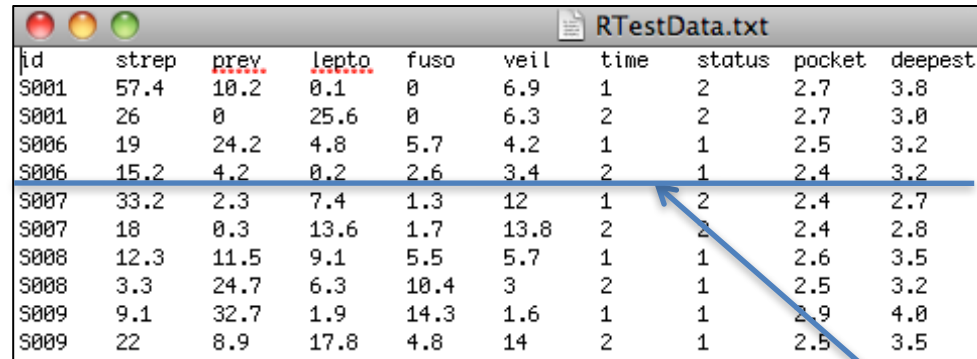
[Previously saved workspace restored]

> █
```

This is where you type commands.

R Tutorial: Reading in a dataset

Step 1: You need a text-only file that R can read. I created a tab-delimited text file called **RTestData.txt**. This data is from a gum disease study and has abundances of bacteria found in the human mouth.



id	strep	prev	lepto	fuso	veil	time	status	pocket	deepest
S001	57.4	10.2	0.1	0	6.9	1	2	2.7	3.8
S001	26	0	25.6	0	6.3	2	2	2.7	3.0
S006	19	24.2	4.8	5.7	4.2	1	1	2.5	3.2
S006	15.2	4.2	0.2	2.6	3.4	2	1	2.4	3.2
S007	33.2	2.3	7.4	1.3	12	1	2	2.4	2.7
S007	18	0.3	13.6	1.7	13.8	2	2	2.4	2.8
S008	12.3	11.5	9.1	5.5	5.7	1	1	2.6	3.5
S008	3.3	24.7	6.3	10.4	3	2	1	2.5	3.2
S009	9.1	32.7	1.9	14.3	1.6	1	1	2.9	4.0
S009	22	8.9	17.8	4.8	14	2	1	2.5	3.5

Description of RTestData: Below is the meaning of each column header.

id=Code for each patient. Are two rows for each subject: one before and one after gum cleaning.

strep=Percentage of Streptococcus bacteria

lepto=Percentage of Leptotrichia bacteria

prev=Percentage of Prevotella bacteria

fuso=Percentage of Fusobacteria bacteria

veil=Percentage of Veillonella bacteria

time=Time that sample was taking: 1 – before gum cleaning; 2 after gum cleaning

Status=Disease status: 1 is healthy, 2 is diseased gums

pocket=The average gum pocket depth across all the teeth in the mouth (in millimeters)

deepest=Depth of the deepest gum pocket in the mouth (in millimeters)

Example: 15.2% of Subject 6 (S006) are Streptococcus (strep) bacteria before treatment (time 1).

RTestData.txt can be downloaded by saving it as a text file or by copy/pasting data into a text file:

<http://kelleybioinfo.org/algorithms/basics/programming/RTestData.txt>

R Tutorial: Reading in a dataset (Mac and Linux)

Step 2: To read in your dataset, you need to know where the dataset is on your computer. (I made it easy and put it on my desktop.) Then type the path to the folder/directory in the console and hit return:

```
> d=read.table("Desktop/RTestData.txt", header=TRUE)
```

All the data are read into the variable "d" (you can use other names if you like).

"read.table" reads in a tab-delimited text file.

This is the file path – it is how R finds your text file.

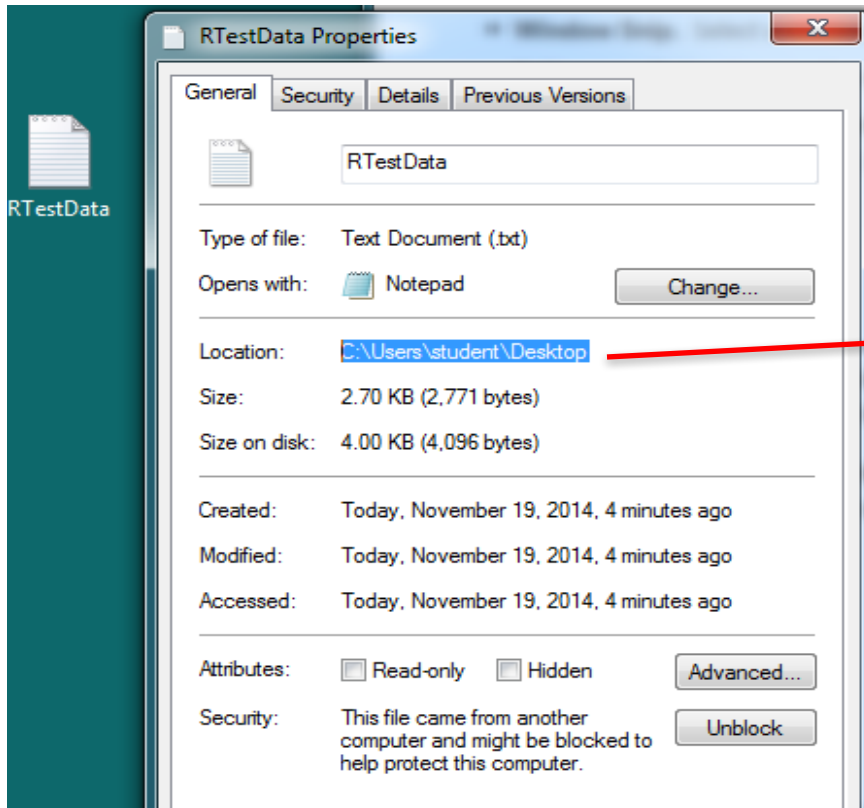
header=TRUE tells R that the variable names are on the first line of the file.

Then, if you just type "d" and hit enter you will see your lovely data!

```
> d=read.table("Desktop/RTestData.txt", header=TRUE)
> d
  id strep prev lepto fuso veil time status pocket deepest
1 S001  57.4 10.2  0.1  0.0  6.9   1     2     2.7     3.8
2 S001  26.0  0.0 25.6  0.0  6.3   2     2     2.7     3.0
3 S006  19.0 24.2  4.8  5.7  4.2   1     1     2.5     3.2
4 S006  15.2  4.2  0.2  2.6  3.4   2     1     2.4     3.2
5 S007  22.2  2.2  7.4  4.2 12.0   1     2     2.4     3.7
```

R Tutorial: Reading in a dataset (Windows)

To read in your dataset in Windows, you have to find the path to the file. To find the path, right-click the data file and choose “Properties” at the bottom of the menu. You will get a window that looks like this:



This is the file path.

To read in the table, you need to use the file path with the `read.table` function. NOTE you have to **use two backslashes between directories** to read in the file path.

```
> d=read.table("C:\\Users\\student\\Desktop\\RTestData.txt",header=TRUE)
> |
```

R Tutorial: Simple analyses

```
> d$strep
```

Use the \$ to get data for specific variables: this command returns strep values for each subject

```
[1] 57.4 26.0 22.0 11.0 1.8 4.8 6.6 9.1 9.7 18.1 8.8  
[21] 8.8 24.1 3.6 9.0 9.0 3.1 5.1 23.3 12.2 3.3 11.4 5.3 10.7 8.5 10.1 6.2 9.2 5.0  
[41] 18.6 5.0 61.7 4.3 16.1 2.9 0.9 2.7 6.6 16.2 2.9 6.4 10.7 5.6 6.1 8.1 19.1 26.2  
[61] 4.0 40.4 36.5 23.0 39.8 24.0 19.9 24.8 7.1 11.8 15.9 16.8
```

```
> mean(d$strep)
```

Get the mean (average) for strep.

```
[1] 13.95833
```

```
> strep=d$strep
```

Save data to new variable names (so you don't have to use the d\$ each time).

```
> fuso=d$fuso
```

```
> cor.test(strep,fuso)
```

The cor.test function tests if two variables are positively or negatively correlated.

Pearson's product-moment correlation

These are the correlation results.

```
data: strep and fuso  
t = -4.8811, df = 70, p-value = 6.404e-06  
alternative hypothesis: true correlation is not equal to 0  
95 percent confidence interval:  
-0.6586898 -0.3082332  
sample estimates:  
cor  
-0.503918
```

```
> plot(strep,fuso)
```

This plots the two variable and produces a neat plot that you can save and show your Mom.

