

DATA SOCIETY™

“If you can’t explain it simply, you don’t understand it well enough.”

- Albert Einstein

Setting up R: overview

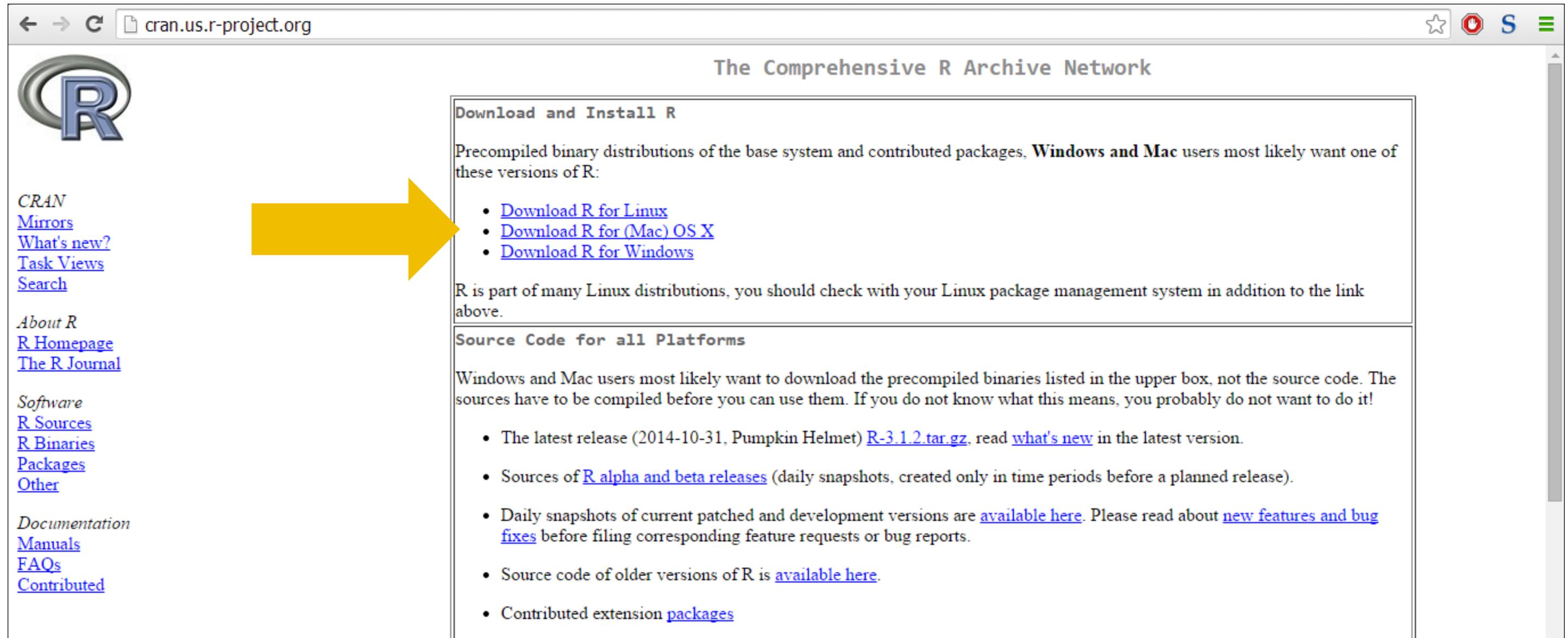
1. What is R?
2. Download R from the CRAN website (<http://cran.r-project.org/>)
 - R for Windows
 - R for Mac
3. Install R Studio (<http://www.rstudio.com/products/rstudio/download/>)
 - RStudio a brief tour
4. Running a script
 - Variables
5. Reading in a data
 - Manually
 - Through the script

What is R?

- R is a statistical programming software
 - Has many similar features to SAS, Excel and SPSS
 - Scripting language
- It's free and open source
- Has lots of helpful pre-built functions
 - You can build your models quicker
- Easy to learn



Install R



The screenshot shows the CRAN website at cran.us.r-project.org. The page title is "The Comprehensive R Archive Network". On the left, there is a navigation menu with links for "CRAN", "Mirrors", "What's new?", "Task Views", "Search", "About R", "R Homepage", "The R Journal", "Software", "R Sources", "R Binaries", "Packages", "Other", "Documentation", "Manuals", "FAQs", and "Contributed". The main content area is divided into two sections. The top section, "Download and Install R", contains the text: "Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:" followed by a list of three links: "Download R for Linux", "Download R for (Mac) OS X", and "Download R for Windows". Below this list, it says "R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above." The bottom section, "Source Code for all Platforms", contains the text: "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!" followed by a list of four items: "The latest release (2014-10-31, Pumpkin Helmet) [R-3.1.2.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.", and "Source code of older versions of R is [available here](#)." and "Contributed extension [packages](#)". A large yellow arrow points from the left side of the page towards the "Download and Install R" section.

R for Windows



The screenshot shows a web browser window with the address bar containing "cran.us.r-project.org". The page title is "R-3.1.2 for Windows (32/64 bit)". On the left, there is the R logo and the text "CRAN Mirrors". A large yellow arrow points from the R logo to a grey box containing three links: "Download R 3.1.2 for Windows (54 megabytes, 32/64 bit)", "Installation and other instructions", and "New features in this version".

← → ↻ ☆ 🔒 S ☰

 **R-3.1.2 for Windows (32/64 bit)**

[Download R 3.1.2 for Windows \(54 megabytes, 32/64 bit\)](#)
[Installation and other instructions](#)
[New features in this version](#)

CRAN
Mirrors

R for Mac

← → ↻ ☆ 🔒 S ☰

R for Mac OS X

This directory contains binaries for a base distribution and packages to run on Mac OS X (release 10.6 and above). Mac OS 8.6 to 9.2 (and Mac OS X 10.1) are no longer supported but you can find the last supported release of R for these systems (which is R 1.7.1) [here](#). Releases for old Mac OS X systems (through Mac OS X 10.5) and PowerPC Macs can be found in the [old](#) directory.

Note: CRAN does not have Mac OS X systems and cannot check these binaries for viruses. Although we take precautions when assembling binaries, please use the normal precautions with downloaded executables.

R 3.1.2 "Pumpkin Helmet" released on 2014/10/31

This binary distribution of R and the GUI supports 64-bit Intel based Macs on Mac OS X 10.6 (Snow Leopard) or higher.

Please check the MD5 checksum of the downloaded image to ensure that it has not been tampered with or corrupted during the mirroring process. For example type `md5 R-3.1.2-mavericks.pkg` in the *Terminal* application to print the MD5 checksum for the R-3.1.2-mavericks.pkg image. On Mac OS X 10.7 and later you can also validate the signature using `pkgutil --check-signature R-3.1.2-mavericks.pkg`

Files:

[R-3.1.2-snowleopard.pkg](#)
MD5-hash: 8a093200b567282932992defff5daf1d
SHA1-hash: e8aee3cc4d3d97d8e5237fb50afmede38e1fb993
(ca. 68MB)

R 3.1.2 binary for Mac OS X 10.6 (Snow Leopard) and higher, signed package. Contains R 3.1.2 framework, R.app GUI 1.65 in 64-bit for Intel Macs. The above file is an Installer package which can be installed by double-clicking. Depending on your browser, you may need to press the control key and click on this link to download the file.

This package contains the R framework, 64-bit GUI (R.app) and Tcl/Tk 8.6.0 X11 libraries. The latter component is optional and can be omitted when choosing "custom install", it is only needed if you want to use the `tcltk` R package. GNU Fortran is **NOT** included (needed if you want to compile packages from sources that contain FORTRAN code) please see [the tools directory](#).

[R-3.1.2-mavericks.pkg](#)
MD5-hash: d8fb6eaf80357dd058aal691c684e091
SHA1-hash: 61c78cbb3024bf648032006fe19d8421c52ac8ba
(ca. 55MB)

R 3.1.2 binary for Mac OS X 10.9 (Mavericks) and higher, signed package. It contains the same software versions as above, but this R build has been built with Xcode 5 to leverage new compilers and functionalities in Mavericks not available in earlier OS X versions.

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

About R
[R Homepage](#)
[The R Journal](#)

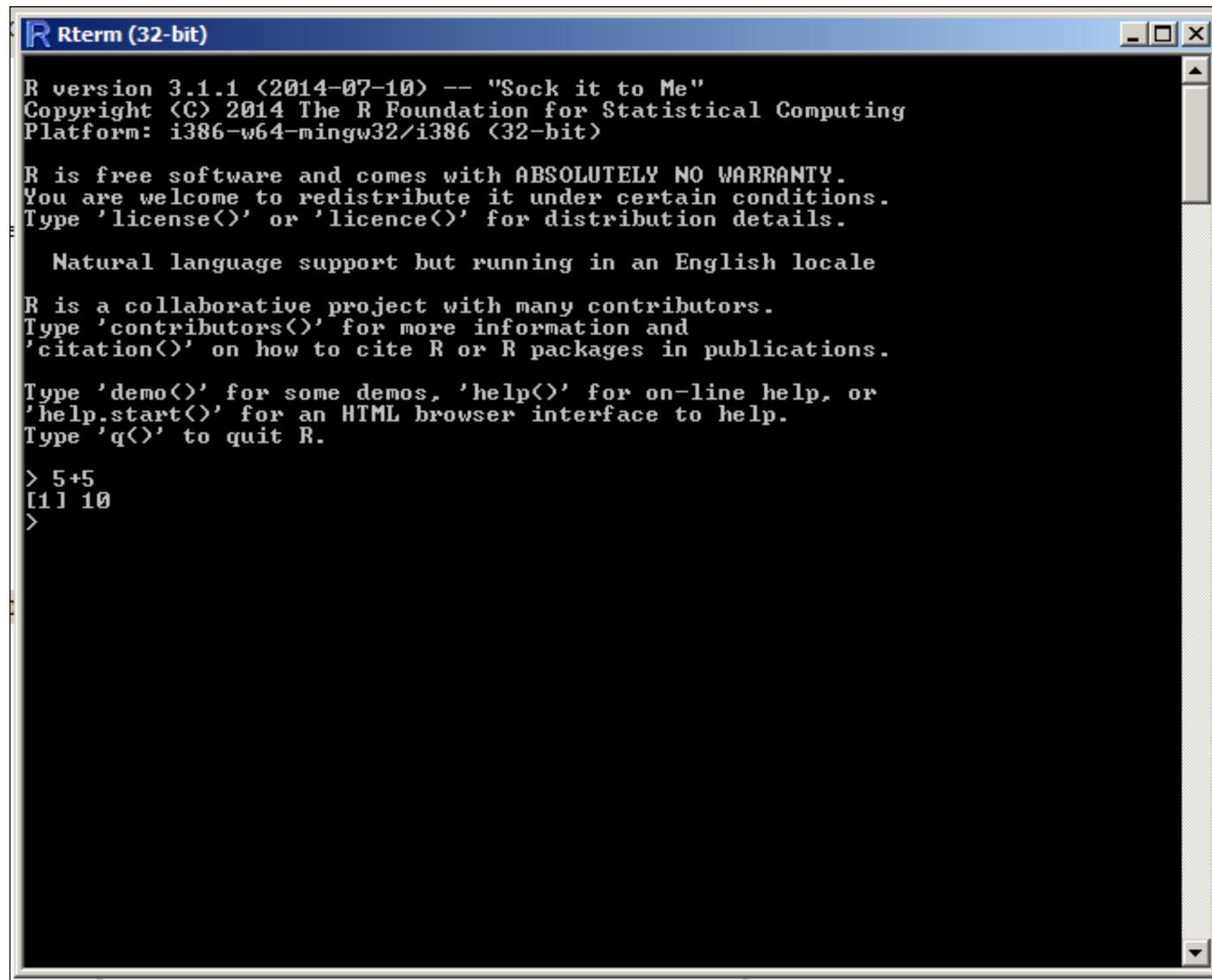
Software
[R Sources](#)
[R Binaries](#)
[Packages](#)
[Other](#)

Documentation
[Manuals](#)
[FAQs](#)
[Contributed](#)

What is RStudio?

The user interface in the R terminal is bulky and non-intuitive

RStudio provides a better interface for a more intuitive user experience



```
Rterm (32-bit)
R version 3.1.1 (2014-07-10) -- "Sock it to Me"
Copyright (C) 2014 The R Foundation for Statistical Computing
Platform: i386-w64-mingw32/i386 (32-bit)

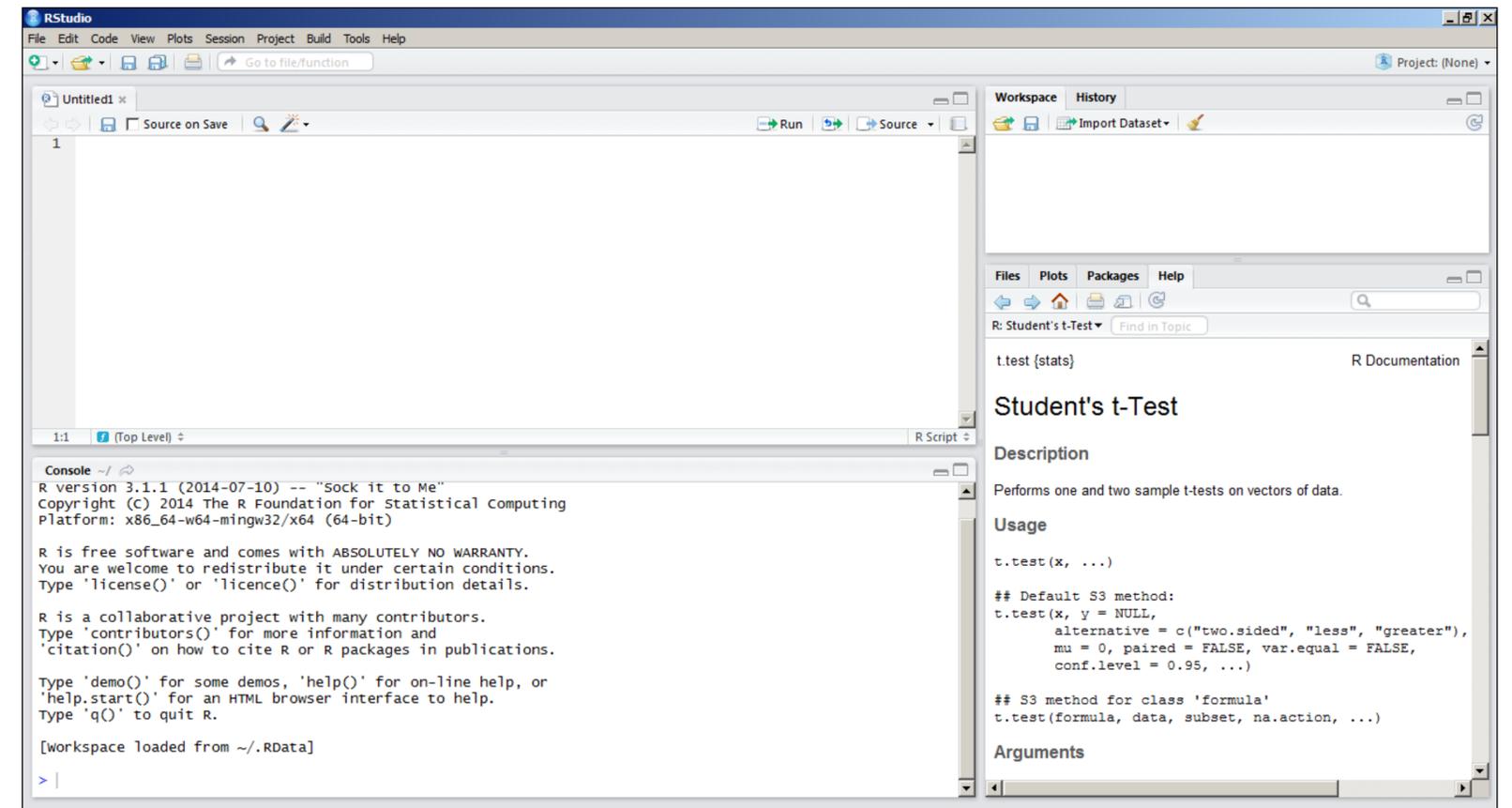
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

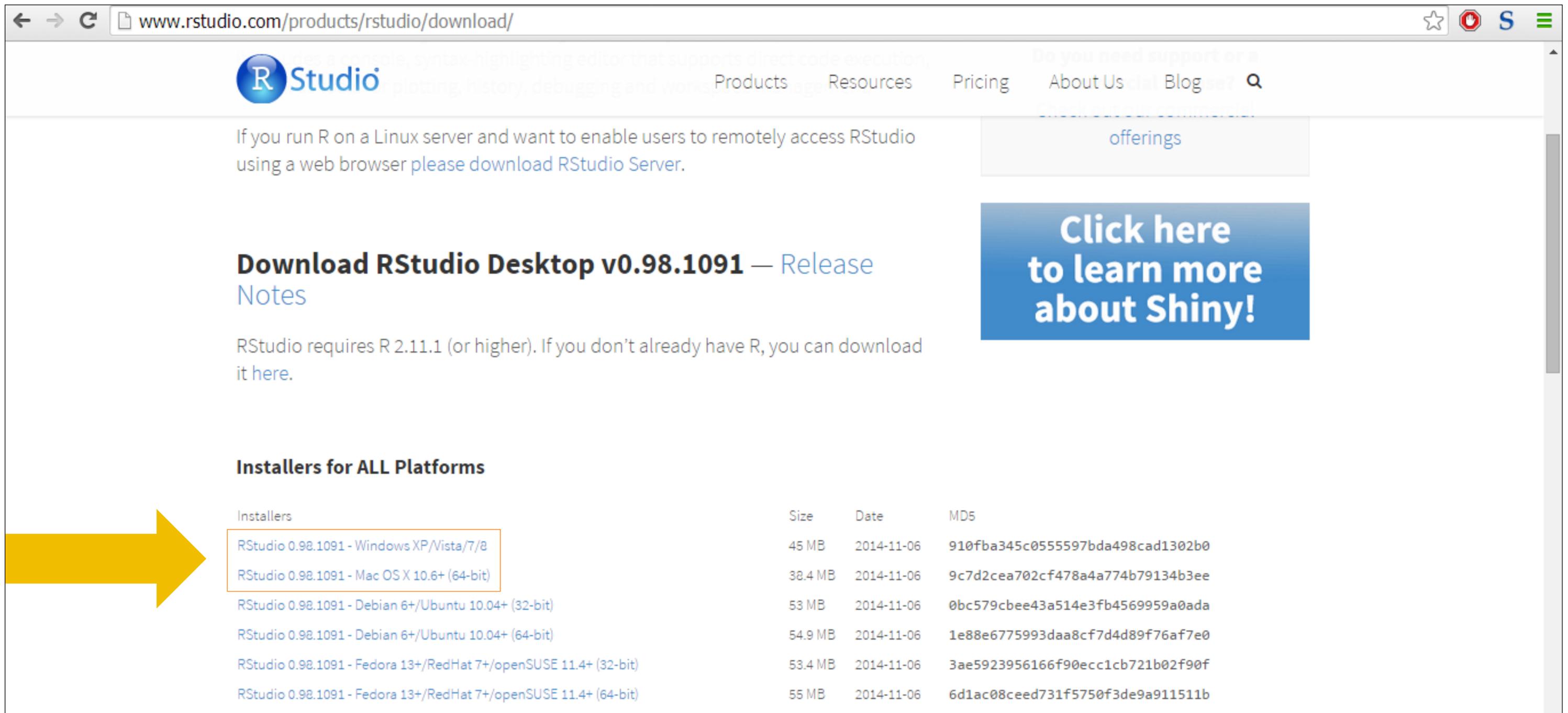
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

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.

> 5+5
[1] 10
>
```



Install RStudio



The screenshot shows the RStudio website's download page. A yellow arrow on the left points to the first link in the 'Installers for ALL Platforms' table: 'RStudio 0.98.1091 - Windows XP/Vista/7/8'. The page includes a navigation menu, a search bar, and a blue button that says 'Click here to learn more about Shiny!'.

← → ↻ www.rstudio.com/products/rstudio/download/ ☆ 🔒 S ☰

RStudio Provides a console, syntax-highlighting editor that supports direct code execution, plotting, history, debugging and workspace saving. Products Resources Pricing About Us Blog

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](#).

Download RStudio Desktop v0.98.1091 — [Release Notes](#)

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

Installers for ALL Platforms

Installers	Size	Date	MD5
RStudio 0.98.1091 - Windows XP/Vista/7/8	45 MB	2014-11-06	910fba345c0555597bda498cad1302b0
RStudio 0.98.1091 - Mac OS X 10.6+ (64-bit)	38.4 MB	2014-11-06	9c7d2cea702cf478a4a774b79134b3ee
RStudio 0.98.1091 - Debian 6+/Ubuntu 10.04+ (32-bit)	53 MB	2014-11-06	0bc579cbee43a514e3fb4569959a0ada
RStudio 0.98.1091 - Debian 6+/Ubuntu 10.04+ (64-bit)	54.9 MB	2014-11-06	1e88e6775993daa8cf7d4d89f76af7e0
RStudio 0.98.1091 - Fedora 13+/RedHat 7+/openSUSE 11.4+ (32-bit)	53.4 MB	2014-11-06	3ae5923956166f90ecc1cb721b02f90f
RStudio 0.98.1091 - Fedora 13+/RedHat 7+/openSUSE 11.4+ (64-bit)	55 MB	2014-11-06	6d1ac08ceed731f5750f3de9a911511b

[Click here to learn more about Shiny!](#)

RStudio

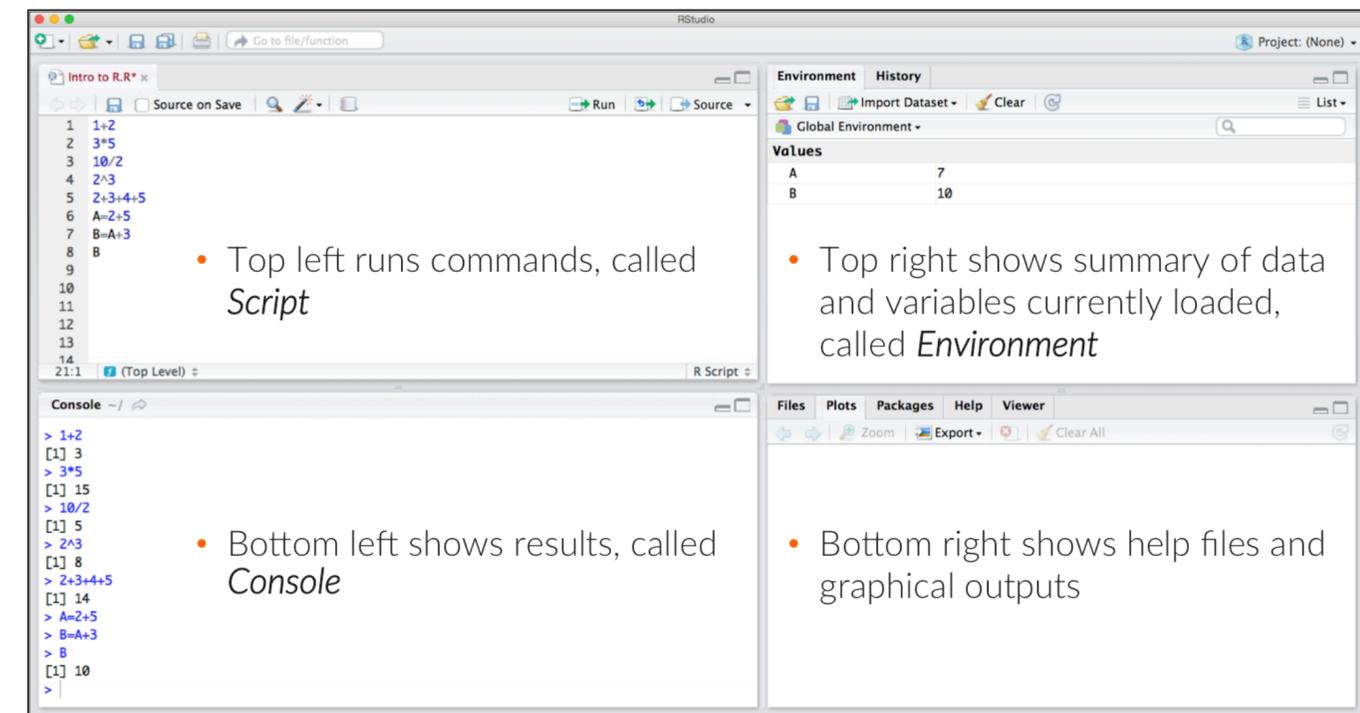
The screenshot shows the RStudio interface with the following panes and callouts:

- Top Left (Script):** Contains R code for calculations and variable assignments. Callout: "Top left runs commands, called *Script*".
- Top Right (Environment):** Shows the current environment with variables A (value 7) and B (value 10). Callout: "Top right shows summary of data and variables currently loaded, called *Environment*".
- Bottom Left (Console):** Shows the output of the commands from the script. Callout: "Bottom left shows results, called *Console*".
- Bottom Right (Files/Plots/Packages/Help/Viewer):** Contains tabs for file management, plots, packages, help, and a viewer. Callout: "Bottom right shows help files and graphical outputs".

Script and Console in RStudio

- Script
 - We use scripts to generate reusable code
 - Scripts are like macros
 - Code is not executed here unless you press “Run” (see next slide)
 - Good coding practice:
 - When saving a script use a name that describes what the script does (i.e. counter, graph)
 - Comment your code using “#”

- Console
 - This is where the code executes
 - It is the actual R program (backend)
 - Anything that you type in here will be executed



Running code from script

The screenshot displays the RStudio interface. The top-left pane shows a script editor with the following R code:

```
1 # Setting up variables
2 # Remember things following # aren't executed
3 a=3
4 b=2
5 d=4
6 |
7 #a=10
8 a+b+d
```

The text "Type code in here" is overlaid in orange on the script editor. The bottom-left pane shows the console with the following output:

```
Type 'license()' or 'licence()' for distribution details.
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.
[Workspace loaded from ~/.RData]
> |
```

The right-hand side of the interface shows the help window for "Student's t-Test". The title is "Student's t-Test" and the description is "Performs one and two sample t-tests on vectors of data." The usage section shows the function signature: `t.test(x, ...)` and the default S3 method: `t.test(x, y = NULL, alternative = c("two.sided", "less", "greater"), mu = 0, paired = FALSE, var.equal = FALSE)`.

Running code from a script

The screenshot shows the RStudio interface. The main editor window displays a script named 'Intro to R.R*' with the following code:

```
1 # Setting up variables
2 # Remember things following # aren't executed
3 a=3
4 b=2
5 d=4
6
7 #a=10
8 a+b+d
```

The code from line 3 to line 8 is highlighted in blue. An orange text box is overlaid on the highlighted code, containing the text: "Highlight the code you want to run".

The console window at the bottom left shows the R startup message:

```
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.

[Workspace loaded from ~/.RData]
```

The right-hand side of the interface shows the 'Workspace' and 'History' panes, and the 'Files', 'Plots', 'Packages', and 'Help' panes. The 'Help' pane is currently displaying the documentation for the 't.test' function, titled 'Student's t-Test'.

Running code from a script

The screenshot shows the RStudio interface. The main editor window displays a script named 'Intro to R.R*' with the following code:

```
1 # Setting up variables
2 # Remember things following # aren't executed
3 a=3
4 b=2
5 d=4
6
7 #a=10
8 a+b+d
```

A yellow arrow points to the 'Run' button in the toolbar above the script editor. A blue highlight covers the code from line 3 to line 8. Overlaid on the script editor is the following text:

To run the code in the console
Press: "Run"
or
Hit: "Ctrl" or "command" + "enter"

The console window at the bottom left shows the R startup message:

```
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.

[Workspace loaded from ~/.RData]
```

The right-hand pane shows the 'R Documentation' for 'Student's t-Test'. The title is 'Student's t-Test' and the description is 'Performs one and two sample t-tests on vectors of data.' The usage section shows the function signature: `t.test(x, ...)`.

Running code from a script

The screenshot displays the RStudio interface with the following components:

- Source Editor:** Contains R code:

```
1 # Setting up variables
2 # Remember things following # aren't executed
3 a=3
4 b=2
5 d=4
6
7 #a=10
8 a+b+d
```
- Console:** Shows the execution output:

```
> # Setting up variables
> # Remember things following # aren't executed
> a=3
> b=2
> d=4
>
> #a=10
> a+b+d
[1] 9
```
- Workspace:** A table showing the current environment:

Variable	Value
a	3
b	2
d	4
- Documentation:** The 'Student's t-Test' help page is visible in the background.

Two yellow arrows highlight the execution flow: one points from the 'Run' button in the source editor to the console, and another points from the console output back to the source editor.

Code is executed
in the console

Running code from a script

The screenshot shows the RStudio interface. The main editor window displays a script with the following code:

```
1 # Setting up variables
2 # Remember things following # aren't executed
3 a=3
4 b=2
5 d=4
6
7 #a=10
8 a+b+d
```

The console window shows the execution of this code:

```
> # Setting up variables
> # Remember things following # aren't executed
> a=3
> b=2
> d=4
>
> #a=10
> a+b+d
[1] 9
```

A yellow arrow points to the line `> #a=10` in the console, with the following text: "Notice #a=10 is a comment and not 'executed', otherwise a+b+d = 16 rather than 9".

The right-hand pane shows the 'Workspace' and 'History' tabs. The 'Values' table is visible:

Variable	Value
a	3
b	2
d	4

The bottom-right pane shows the 'R Documentation' for 'Student's t-Test'.

Student's t-Test

Description
Performs one and two sample t-tests on vectors of data.

Usage
t.test(x, ...)

Variables

The screenshot shows the RStudio interface. The source editor on the left contains the following R code:

```
1 # Setting up variables
2 # Remember things following # aren't executed
3 a=3
4 b=2
5 d=4
6
7 #a=10
8 a+b+d
```

A yellow arrow points from the code to the Workspace pane on the right, which is highlighted with an orange border. The Workspace pane shows the following table of values:

Variable	Value
a	3
b	2
d	4

The Console at the bottom shows the execution of the code:

```
> # Setting up variables
> # Remember things following # aren't executed
> a=3
> b=2
> d=4
>
> #a=10
> a+b+d
[1] 9
```

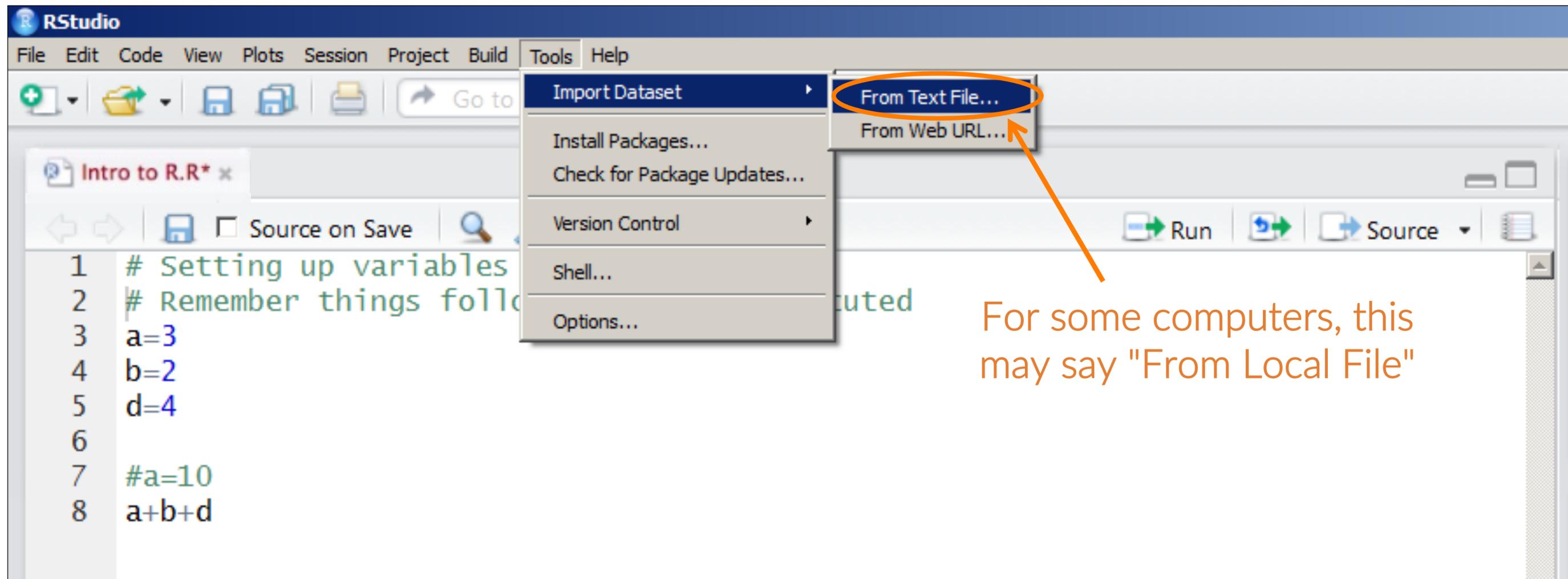
The right-hand pane shows the R Documentation for the `t.test` function, titled "Student's t-Test".

Once the code is run
the variables are
“stored” in memory.
Here we have a, b, d

Reading in data

- Data comes in many forms
 - R can read in most formats
- The most common to work with now days are
 - `csv` files – comma separated value files
 - `tsv` files – tab separated value files
- R can also read in others (excel and json) but that is not covered here

Reading in data: manually



Tools > Import Dataset > From Text File...

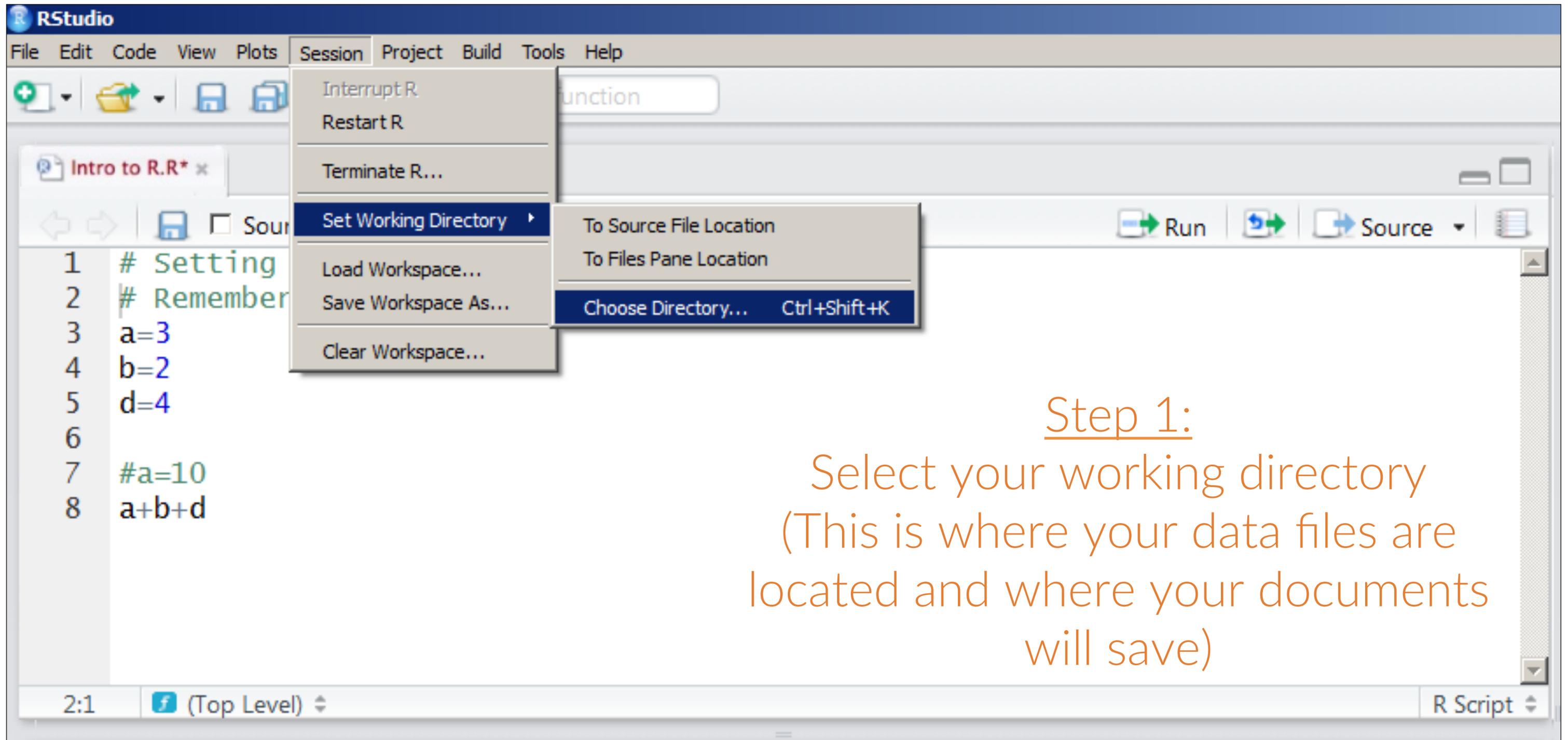
Reading in data: manually

- RStudio will automatically select options for you such as heading, sometimes it makes mistakes so be sure to double check

The screenshot shows the RStudio interface with the 'Import Dataset' dialog box open. The dialog is configured to import a CSV file named 'Credit.Data'. The 'Input File' field contains a preview of the data, which is a CSV file with columns: Customer Name, Delinquency (days), Monthly Spend (\$000's), and % Delinquent. The 'Data Frame' section shows a preview of the data as a table with columns: Customer.Name, Delinquency..days., and Monthly.Spend...00. The 'Console' window shows the execution of R code: '# Setting up variables', '# Remember things follow', and 'a=3'.

Customer Name	Delinquency (days)	Monthly Spend (\$000's)	% Delinquent
Andrews	5	48	49
Banks	4	38	41
Becker	45	6.5	49
Bennett	67	5.2	93
Berry	22	1.3	2
Blair	21	1.7	2
Blake	6	35	48
Bowen	28	12	71
Boyd	21	1.2	8
Bradley	78	6.5	94
Bryant	36	5.5	46
Bush	23	13	64
Cain	7	42	42

Reading in data: through the script

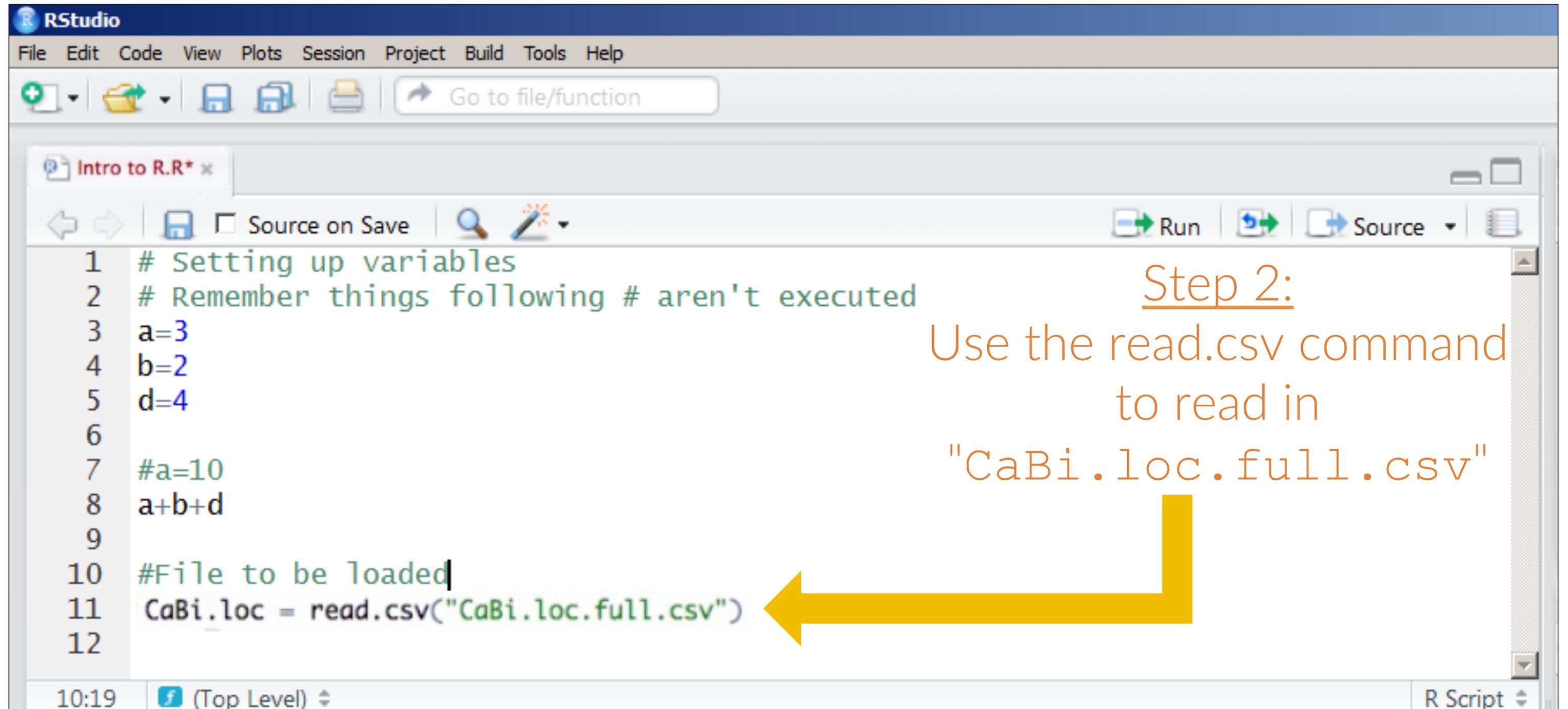


The screenshot shows the RStudio interface. The 'Session' menu is open, and 'Set Working Directory' is selected. A sub-menu is displayed with 'Choose Directory...' highlighted. The code editor shows the following R script:

```
1 # Setting
2 # Remember
3 a=3
4 b=2
5 d=4
6
7 #a=10
8 a+b+d
```

Step 1:
Select your working directory
(This is where your data files are
located and where your documents
will save)

Reading in data: through the script



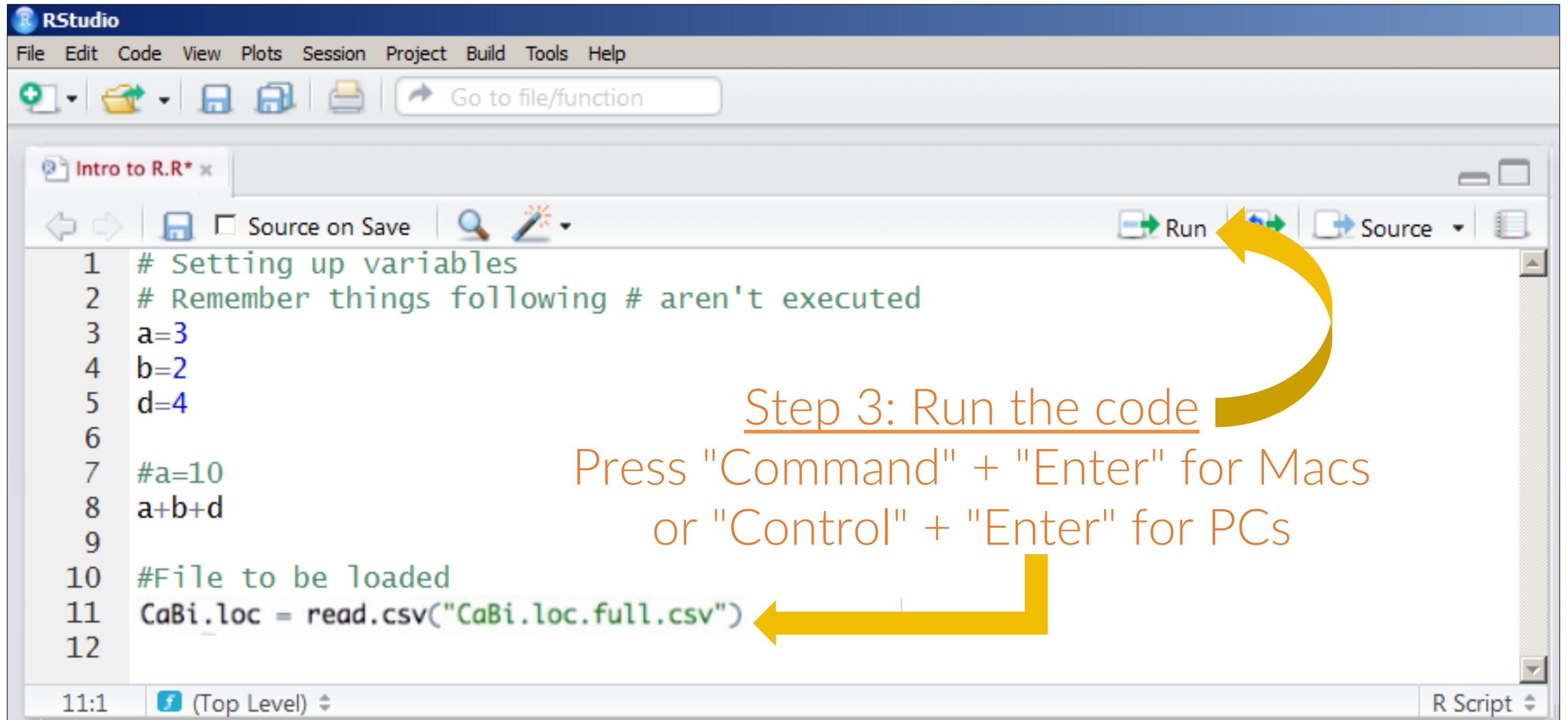
The screenshot shows the RStudio interface with a script editor containing the following R code:

```
1 # Setting up variables
2 # Remember things following # aren't executed
3 a=3
4 b=2
5 d=4
6
7 #a=10
8 a+b+d
9
10 #File to be loaded
11 CaBi.loc = read.csv("CaBi.loc.full.csv")
12
```

Step 2:
Use the read.csv command
to read in
"CaBi.loc.full.csv"

A yellow arrow points from the text "Step 2: Use the read.csv command to read in 'CaBi.loc.full.csv'" to the line of code in the script: `CaBi.loc = read.csv("CaBi.loc.full.csv")`.

Reading in data: through the script



The screenshot shows the RStudio interface with a script editor containing the following code:

```
1 # Setting up variables
2 # Remember things following # aren't executed
3 a=3
4 b=2
5 d=4
6
7 #a=10
8 a+b+d
9
10 #File to be loaded
11 CaBi.loc = read.csv("CaBi.loc.full.csv")
12
```

Annotations on the screenshot include:

- A yellow arrow pointing from the `Run` button in the toolbar to the text "Step 3: Run the code".
- Yellow arrows pointing from the text "Press 'Command' + 'Enter' for Macs or 'Control' + 'Enter' for PCs" to the `CaBi.loc = read.csv("CaBi.loc.full.csv")` line in the script.

Reading in data: through the script

The screenshot shows the RStudio interface with the following components:

- Source Editor:** Contains R code for setting variables and reading a CSV file.

```
1 # Setting up variables
2 # Remember things following # aren't executed
3 a=3
4 b=2
5 d=4
6
7 #a=10
8 a+b+d
9
10 #File to be loaded
11 CaBi.loc = read.csv("CaBi.loc.full.csv")
12
```
- Console:** Shows the execution of the script, with the current line highlighted. An orange arrow points to the path `~/Desktop/Data Society/Intro to DS exercises/` in the console header, with the text "Working directory location" next to it.

```
~/Desktop/Data Society/Intro to DS exercises/
> CaBi.loc = read.csv("CaBi.loc.full.csv")
```
- Data Viewer:** Shows the loaded data as a table with 2278 observations and 15 variables. The table content is as follows:

Variable	Value
a	3
b	2
d	4

Setting up R: overview

1. What is R?
2. Download R from the CRAN website (<http://cran.us.r-project.org/>)
 - R for Windows
 - R for Mac
3. Install R Studio (<http://www.rstudio.com/products/rstudio/download/>)
 - RStudio a brief tour
4. Running a script
 - Variables
5. Reading in a data
 - Manually
 - Through the script