

Introducing RStudio



As with R, so RStudio is also a free and open source project. It was founded by JJ Allaire. Work started on this in late 2010. See [Wikipedia entry](#) for further background or go to [official RStudio website](#).

RStudio is an **Integrated Development Environment** (IDE) for R. It integrates the R environment, with a highly advanced text editor, R's help system, version control, and much more into a single application. It should be stressed that RStudio does not perform any statistical operations; it simply makes it easier for you to perform such operations with R. The table below provides a summary of many of the key features of RStudio

Feature	Description
Integration of the R Console	Type commands directly in the R console within RStudio
Code execution	Directly execute code from script files
Syntax highlighting	Colour keywords and functions for easy reading
Bracket support	Matching brackets are highlighted on selection. When typing a square bracket "[", bracket "(", curly brace "{" or single or double quote, RStudio autocompletes it for you
Command completion	Press TAB before completing a command and RStudio will show a menu of matching R functions. When a function is chosen its arguments and 'help' can also be shown
Keyboard shortcuts	Common tasks can be accessed quickly by pressing a key or key combination
Help integration	RStudio allows for browsing and searching R's native help files, and offers context related help as well
Object browser	You can inspect every object defined in the running R session
History browser	RStudio makes it easy to see what commands you used and to re-execute them
Code navigation	Jump from the use of a function to its definition. Jump from code in a report to the code in the source
Data viewer	A spreadsheet-like view of data frames

Data import menus	For some of the most common data types RStudio has a menu that generates the R Read command for you
Graphics integration	Zoom, manipulate, and export graphics interactively
Project management	Easily switch between several projects
Version control	RStudio integrates the popular version control systems git and svn
Document generation	Generate pdf, html, or other report formats using RMarkdown, Sweave, or knitr with the push of a button
Publishing	Publish your reports and scripts online at Rpubs.com so that others may learn from your examples

RStudio – basic layout

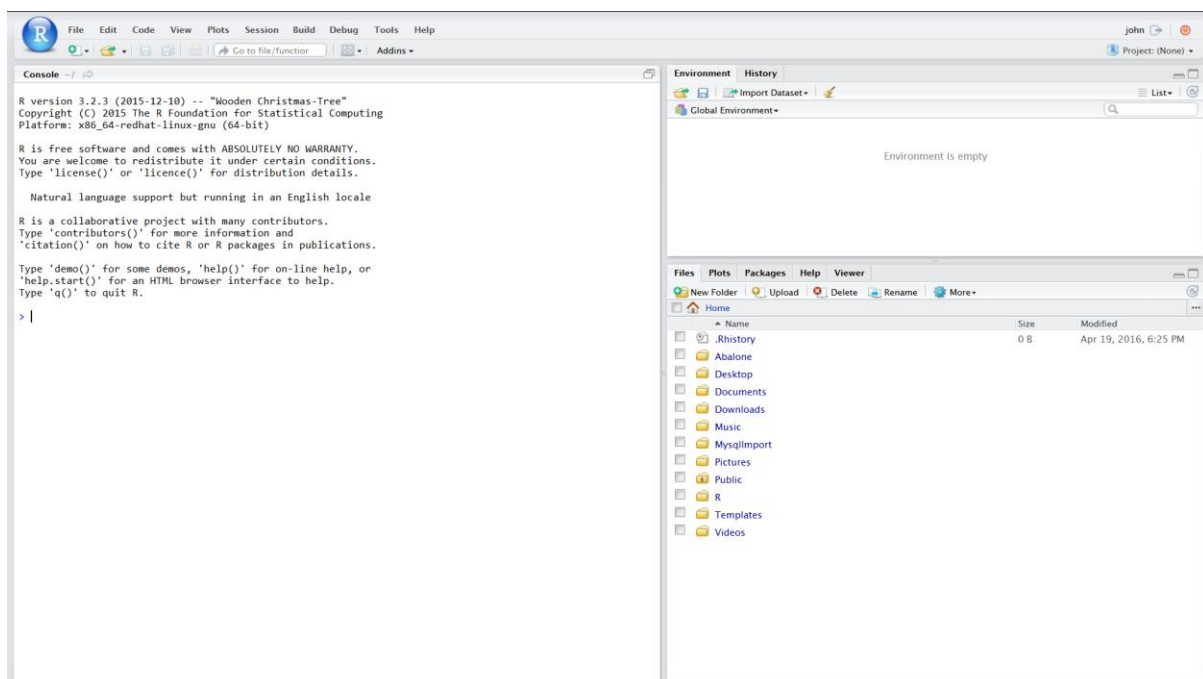
When you first start up RStudio after installation it will open with three panels with the following headings visible at the top of each panel:

Console

Environment History

Files Plots Packages Help Viewer

Overall it will look something like this:



Orientating yourself to RStudio – the Console panel

The easiest way to find out how RStudio works and what the various panels do is simply to jump in and start using it... We will start with the Console panel which for a fresh installation of RStudio is normally at the top left of your screen. Anything that is typed into this panel will be passed to R and interpreted by it.

We can start by trying some simple **arithmetic operators**. Open up your installation of RStudio now. Try typing in, or copy and paste, to your Console panel each of the following lines one by one – then press <ENTER>

```
2 + 2
```

```
5 - 3
```

```
12 * 5
```

```
100 / 25
```

```
5 ^ 3
```

```
13 %% 5           #this is 13 mod 5
```

```
13 %/% 5          #this is integer division. Note there should be no spaces in “%/%”
```

Note that in each of the above cases R interprets the expression entered and comes back with an answer

For example in the case of: 5^3 it interprets this as ‘5 to the power of 3’ and returns with

```
[1] 125
```

Try making a few entries of your own.

So far we have demonstrated R working as a powerful calculator.

Note:

1. R treats anything to the RIGHT of a ‘#’ character as a comment. Later we will encourage you to use comments liberally in your coding
2. At any time you can clear the console of its entries by using <CTRL> L. Try this now
3. You can scroll through all of the commands that you have executed by using UP and DOWN arrows – even after you have cleared the console. This can be useful if you want to modify a lengthy command that you have previously used

Orientating yourself to RStudio – the Environment panel

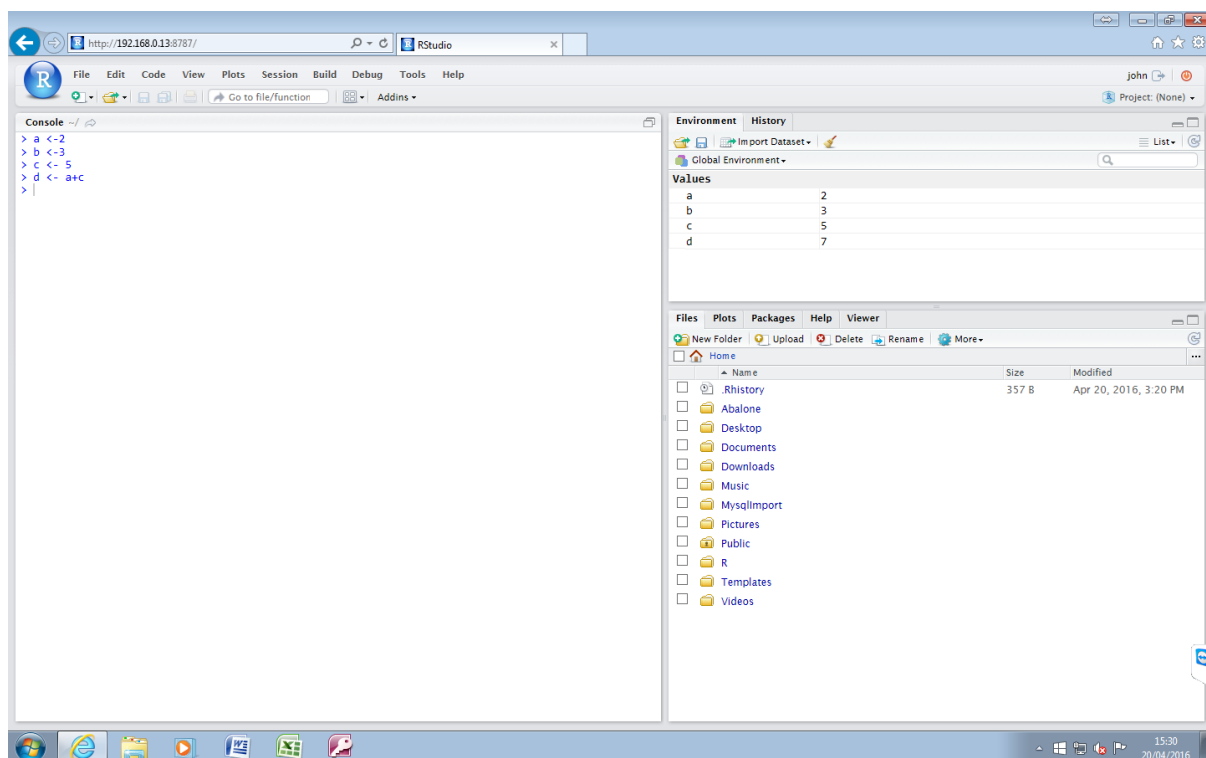
Assigning values to variables

The use of the “<-” convention is considered to be best practice for assigning values to variables. The reasons for this will become clearer when we look at logical operators later on. Try typing or copy and paste the following:

```
a <- 2  
b <- 3  
c <- 5  
d <- a+c
```

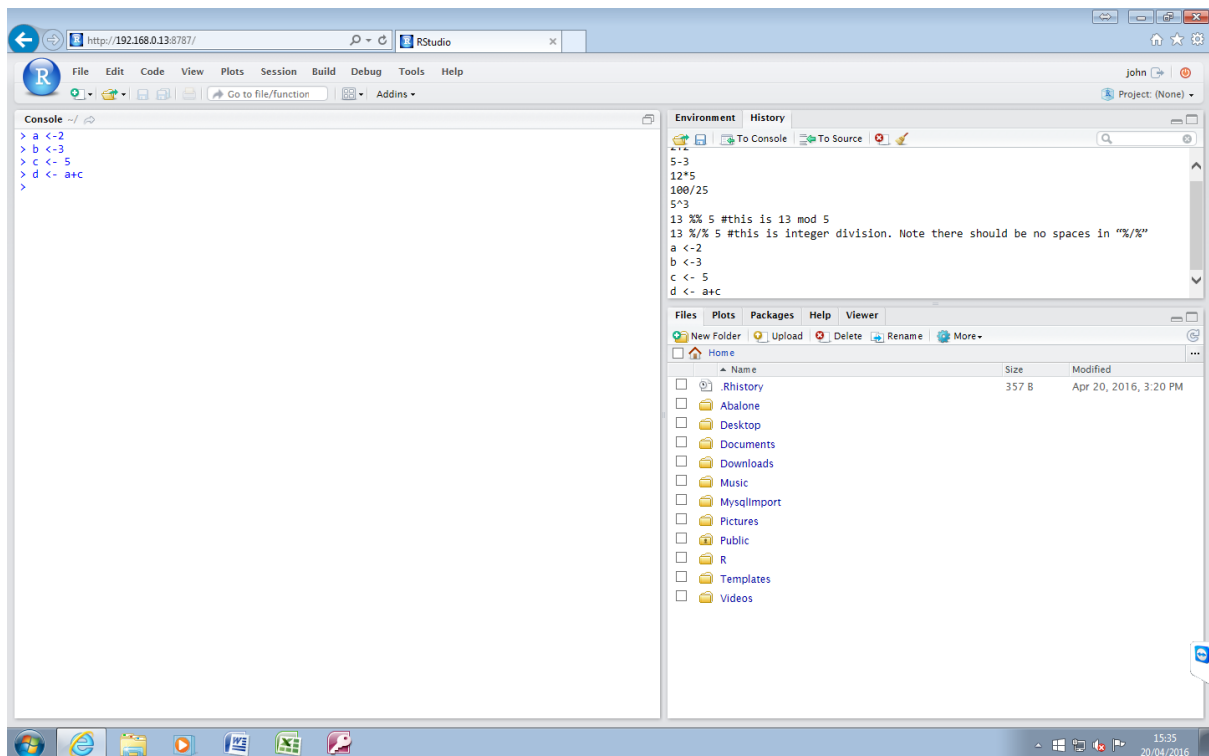
Note that this time R does not come back with any response to these in the console panel.

However, values have been assigned the variables a, b, c, d – as you will see if you now look in the top right **Environment panel**. A number of Values have appeared in this panel in response to the assignments just made. In this panel you will always see all of the variables created in this session of R. It will currently appear similar to this:



The History Panel

Furthermore, if you toggle the view to show the History panel (by clicking on the History tab) you will be able to see all of the commands that you have executed thus far during this session of R. And these will persist even if all of the entries in the Console panel have been cleared. You can scroll up and down this list using the slider



Note:

1. You can highlight one or more lines of this history and then click on “To console” to send these commands back to the Console where you can execute them again
2. Both the History panel and the Environment panel can be cleared by clicking on the broom icon
3. In each case there is also an option to save either the Environment or the History

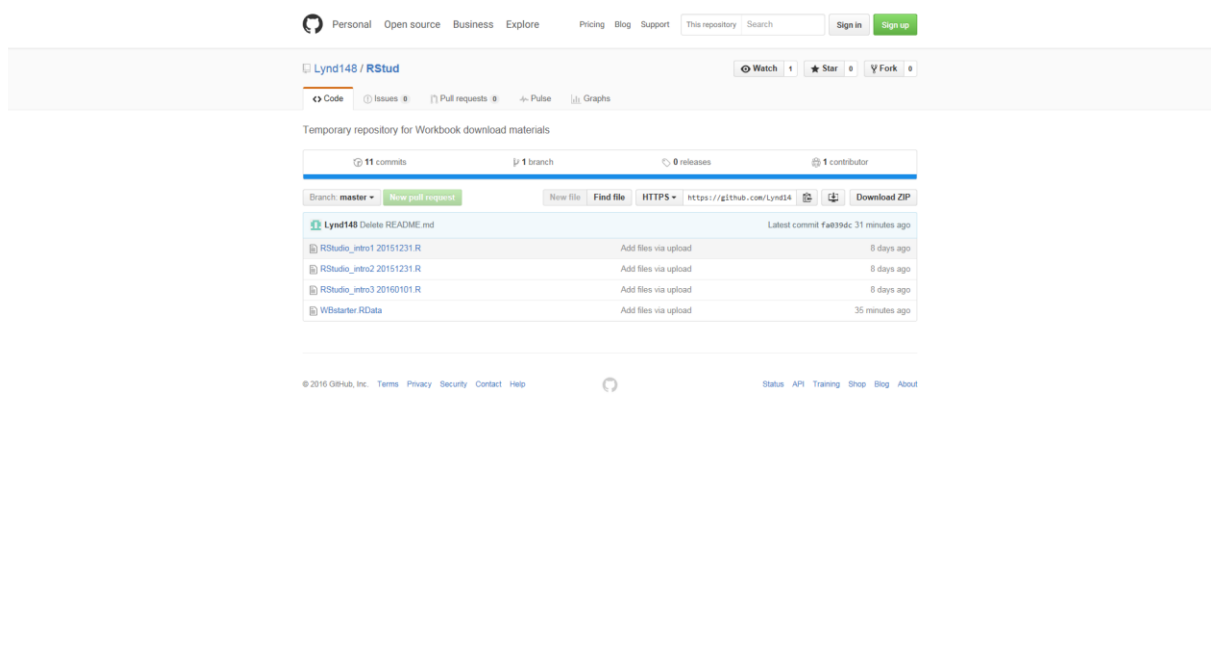
Files Plots Packages Viewer

This is the bottom right panel. The next step will be to set up a folder to hold scripts and other files that will assist us in continuing to explore the functionality of R and RStudio. First we will download some files from a web

based repository and the procedure for this will depend on whether you have RStudio installed on a desktop / notebook or whether it is server based.

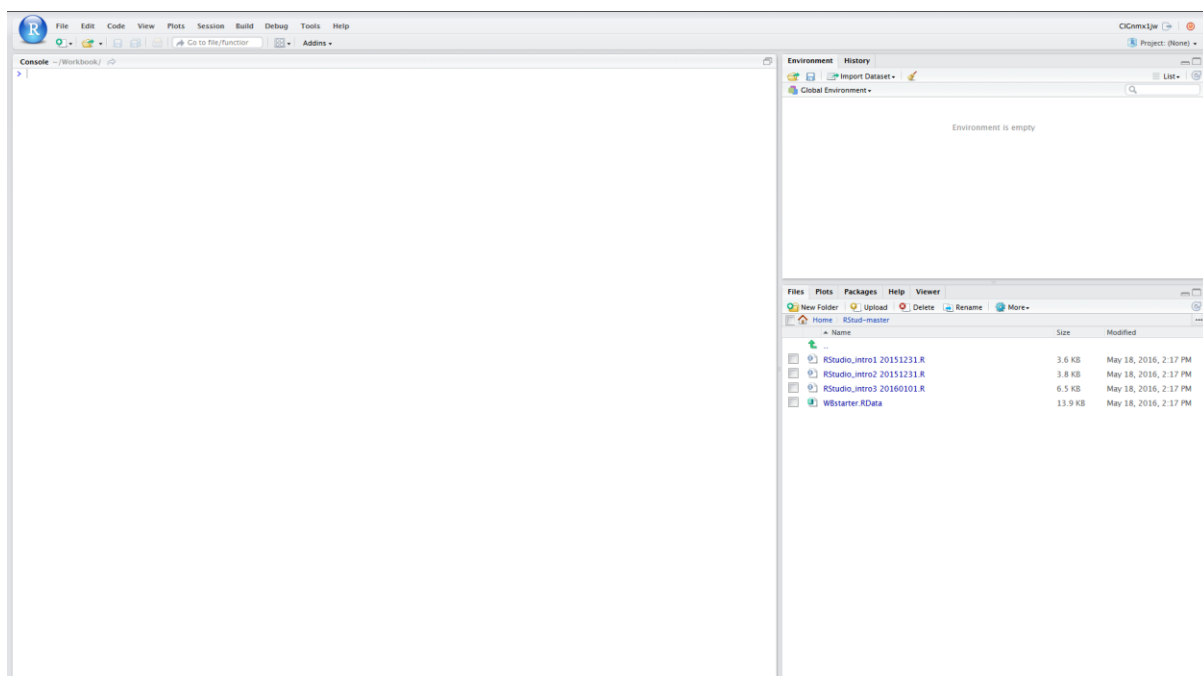
Downloading files from web repository where RStudio is desktop / notebook based

1. Open your browser and either copy and paste the following URL to the address bar or type it in: <https://github.com/Lynd148/RStud.git>
2. This should open up a github repository with the title “Temporary repository for Workbook download materials” which will look something like this and contain a number of files. Note the button marked Download ZIP below the blue line



3. Click on the Download ZIP button. Depending on your browser and settings this will either open a dialogue box asking whether you want to open or save a file called RStud-master.zip or else will result in this file being immediately downloaded to your download folder. In the former case select 'open'. In the latter case navigate to your download folder and right click on the file and select 'open'. You will then see a folder called RStud-master
4. Copy the folder **RStud-master** to your C-drive or other easily found destination

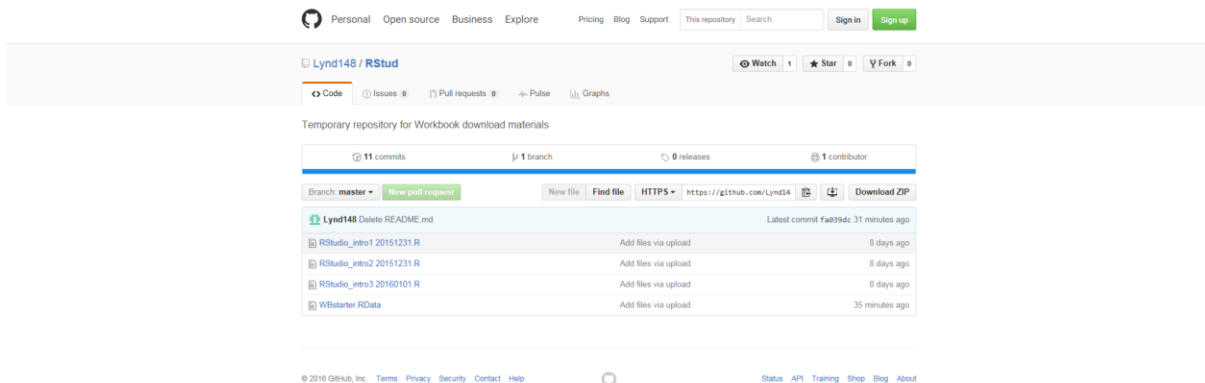
5. Now return to RStudio. Make sure that the **Files** tab of the bottom right **Files Plots Packages Help Viewer** panel is active and click on the small box at the right edge of the screen containing three dots ...
6. This should open a “Browse for folder” box. Browse to the folder **RStud-master** on your C-drive or wherever else you may have placed it
7. The folder will now open under the **Files** tab which should look similar to the screen shot below
8. Click on More tab (next to blue cogwheel icon just below the **Files Plots Packages Help Viewer** bar) and then Click on “Set as Working Directory”
9. Congratulations – you have now downloaded the scripts and other files that we need in order to proceed and set the working directory



Downloading files from web repository where RStudio is server based

1. Open your browser and either copy and paste the following URL to the address bar or type it in: <https://github.com/Lynd148/RStud.git>
2. This should open up a github repository with the title “Temporary repository for Workbook download materials” which will look something

like this and contain a number of files. Note the button marked Download ZIP below the blue line



3. Click on the Download ZIP button. Depending on your browser and settings this will either open a dialogue box asking whether you want to open or save a file called RStud-master.zip or else will result in this file being immediately downloaded to your downloads folder. In the former case select 'save' because in this scenario we want a ZIP file to be placed in your download folder
4. Having ascertained the location of your downloads folder return to RStudio. Make sure that the **Files** tab of the bottom right **Files Plots Packages Help Viewer** panel is active.
5. First need to set Working Directory
 - a. Click on Home
 - b. Click on More (next to blue cogwheel icon) and then Click on "Set as Working Directory"
6. Now look for the Upload button just below the **Files Plots Packages Help Viewer** bar. Click on this
7. This should open a "Upload Files" box. Within that box browse to the destination folder for your downloads and find the file **RStud-master.zip** and select / open this
8. This should result in the folder **RStud-master**
9. Open this

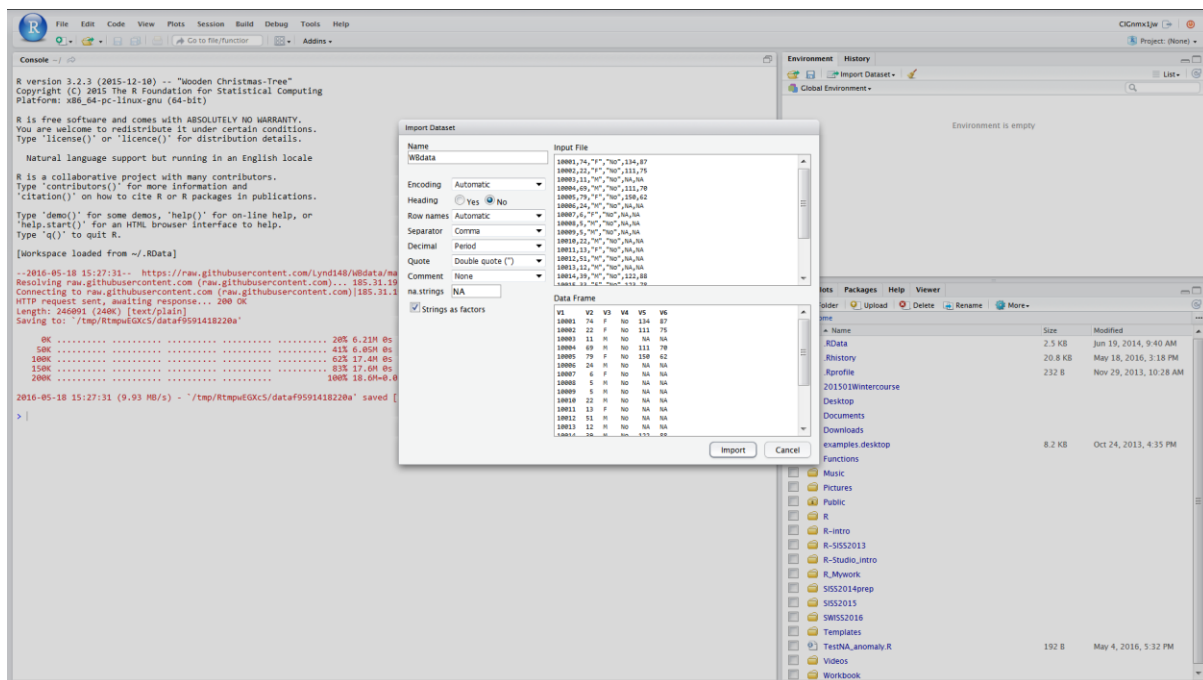
10. Once again click on More tab (next to blue cogwheel icon) and then Click on “Set as Working Directory”
11. Congratulations – you have now downloaded the scripts and other files that we need in order to proceed and set the working directory

Importing dataset from Web URL

This is where things start to get interesting. In this section we will download a heavily pseudonymised CSV file that contains one line per patient data and then set to work demonstrating what R can do.

To do this go to the top **Environment History** panel and ensure that the Environment tab is active. Just below and to the Right of this tab is the “Import Dataset” tab.

1. Click on the Import Dataset tab and select “From Web URL”
2. This will open the “Import from Web URL” box
3. Either copy and paste or type in the following URL
<https://raw.githubusercontent.com/Lynd148/WBdata/master/WBdata.csv>
4. Click on ‘OK’
5. The Import dataset wizard box will open and should look similar to the screen shot below. We will go into the details of this later but for now will simply concentrate on getting this data imported. It should be possible to accept all of the default settings



- Click on Import and note that the CSV file has now been imported. A new object WBdata has appeared in the Environment screen (top right) and this has been opened in the Source panel (top left). It clearly has rows and columns and the appearance of a table. Congratulations. You have just created your first Dataframe

The Environment (top R panel)

This now has one object – a dataframe called WBdata. It is possible to save the environment at any time and then re-open it later. Before saving on this occasion we will add some further components.

Look under the **File** tab (bottom R panel) and you will see two kinds of files there. Firstly there are .R files; these are R-scripts and we will be using these shortly. Secondly there is one .Rdata file: these store all of the objects that were in the Environment at the time that the file was saved. They represent a kind of snapshot at a point in time of the environment. **WBstarter.Rdata** was saved elsewhere and holds some useful functions that we will want to add to the environment. To do that, click on the file **WBstarter.Rdata** and click 'Yes'. You should now see that three objects have been added to the environment under the sub heading of **Functions**. We will be using these shortly when we deploy the scripts.

For now, we want to save the current state of the environment. To do that, click on the blue floppy disk icon just below the Environment tab. Note that

this defaults to the folder that is currently set as the Working Directory. Choose a suitable name such as WBenv and click Save. There is no need to append .Rdata as R will automatically do that. Note that a new file called **WBenv.Rdata** has appeared under the **Files** tab.

We are now ready to start work using the .R script files

The Source Panel

Next we will look at the Source panel within which we can run scripts and which until now may not have been visible. In preparation for this we will download some files from a web based repository and so you will need an Internet connection. to do this and so will also take a look at the facilities provided within the final bottom right panel:

Files Plots Packages Help Viewer