Data Wrangling

Using R to interrogate your data

Reading files

Before we can work on our data, we need to load it in R.

Therefore, we need to read the data files. R has many ways to do this:

```
read.table
read.delim
read.csv
read.csv2
read.delim2
```

These are built-in “functions” that R provides

- Encapsulates or “hides” functionality, in this case opening, reading, closing a file
- We do not care how it opens or closes files, just that it reads our data correctly

All those read files, but with slightly different default behavior, suggested by the name. For example, “read.csv” will read CSV files (Comma-Separated Values) by default.

Suggestion: Pick `read.table` and learn some of the more common ways to “customize” how it reads your file(s).

Using an R function

Common use:

More about this in class

```
my_data = read.table(‘/home/brian/myfile.tsv’, sep=’\t’, stringsAsFactors=F)
```

Location of file to read. Note the quotes (Can use either ‘ or “ quotes)

How each column/field is separated

Anything inside the parentheses is called an “argument” to this function, which affects how it behaves. Each argument is separated by a comma

Stores the file’s contents in the `my_data` variable. Just like in algebra if we write x=3, the variable x stores the value of 3

Sometimes in R, you will see `my_data <- …` instead of `my_data = …`
These are the same for anything you will be doing. There is a subtle difference in some advanced contexts, but don’t worry about that

Where are my files?

When you open RStudio, it will “start” in some folder (“/home/brian” here. That is a folder on my computer). You can find where you are by typing `getwd()` in your RStudio console, as shown on the right side. “wd” means “working directory” so you can read `getwd()` as “Get working directory”

To open files, you can find them in two ways:

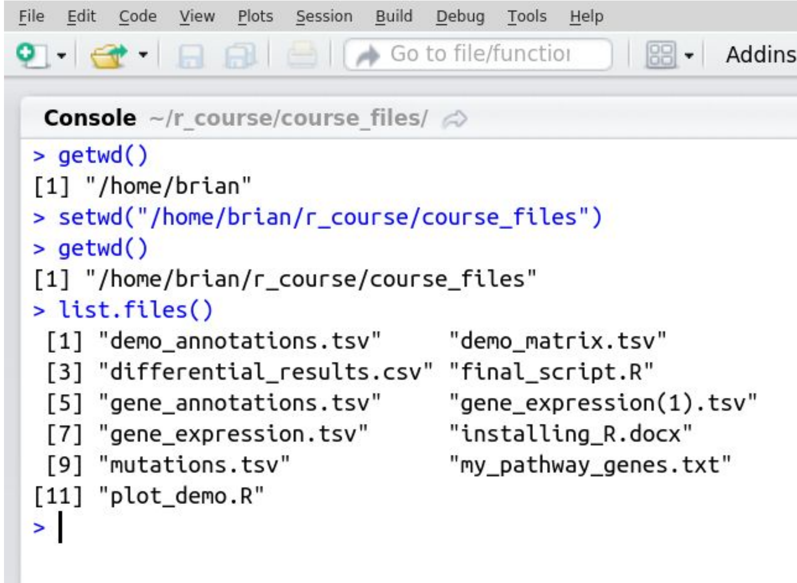
- Absolute location
- Relative location

Absolute locations can be very long to type. To read my mutations file I could write:

```
data = read.table(“/home/brian/r_course/course_files/mutations.tsv”)
```

Relative locations are *relative* to your working directory. To change that, you use the `setwd` function (“set working directory”). On the right side, I have set my working directory to be `/home/brian/r_course/course_files`. Once that is set, I can open the mutations file with

```
data = read.table(“mutations.tsv”)
```



```
> getwd()
[1] "/home/brian"
> setwd("/home/brian/r_course/course_files")
> getwd()
[1] "/home/brian/r_course/course_files"
> list.files()
 [1] "demo_annotations.tsv"     "demo_matrix.tsv"
 [3] "differential_results.csv" "final_script.R"
 [5] "gene_annotations.tsv"     "gene_expression(1).tsv"
 [7] "gene_expression.tsv"      "installing_R.docx"
 [9] "mutations.tsv"            "my_pathway_genes.txt"
[11] "plot_demo.R"
> 
```

Where are my files?

When you downloaded your files, you hopefully remembered where you saved them.

Use the `setwd` function and set it to the folder where you saved your course files. See the previous slide where we use `setwd()`. If you get errors, try to diagnose with Google (or email us for help).

For Mac, your folders will look something like “/Users/brian/some_folder/xyz”

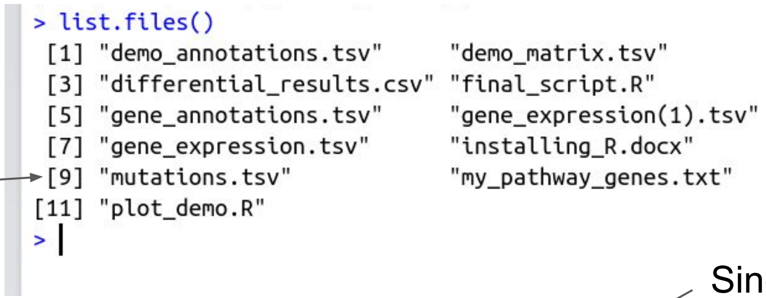
For Windows, your folders are (probably) named something like “C:\\Users\brian\some_folder”

- Note the direction of the slashes (Windows uses “\” instead of “/”)
- However, when using with RStudio, type forward slash:

  setwd(“C:/Users/brian/some_folder”)

Now back to reading files

Here, we read mutations.tsv. Assume that your mutations.tsv file is in the same folder as your working directory. If you type `list.files()`, you should see your files listed:



```
> list.files()
 [1] "demo_annotations.tsv"      "demo_matrix.tsv"
 [3] "differential_results.csv" "final_script.R"
 [5] "gene_annotations.tsv"      "gene_expression(1).tsv"
 [7] "gene_expression.tsv"       "installing_R.docx"
 [9] "mutations.tsv"             "my_pathway_genes.txt"
[11] "plot_demo.R"
>
```

Our file

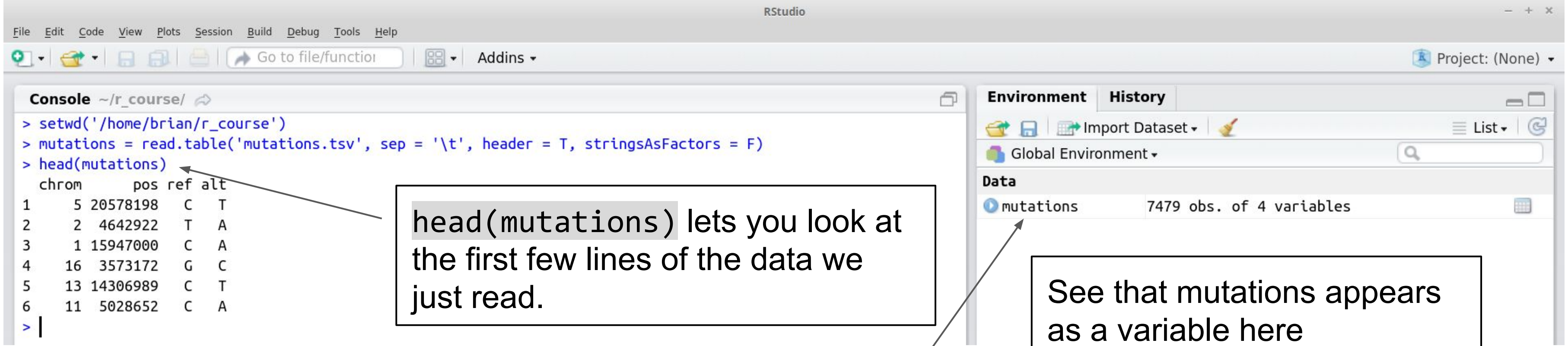
Since it’s just “mutations.tsv”, R knows to look in the current folder (the current working directory)

```
mutations = read.table(‘mutations.tsv’,

                       sep=’\t’,

                       header=T,

                       stringsAsFactors=F)
```

Each column separated by a TAB character. ‘\t’ is how we write that.

The file has names for the columns. T me True.

We’ll talk about this in class.



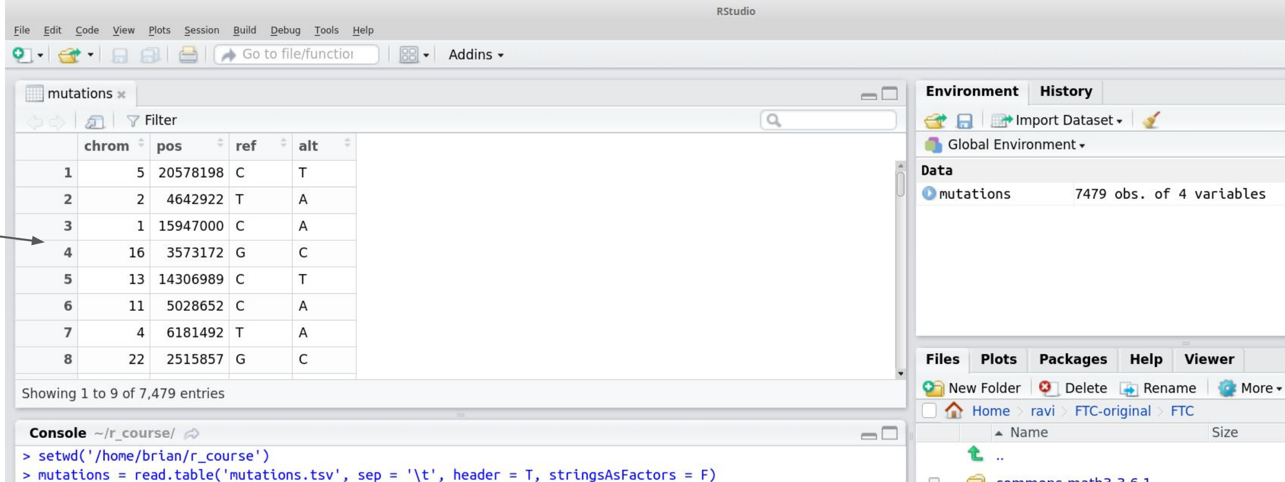
```
> setwd('/home/brian/r_course')
> mutations = read.table('mutations.tsv', sep = '\t', header = T, stringsAsFactors = F)
> head(mutations)
  chrom      pos ref alt
1     5 20578198   C   T
2     2  4642922   T   A
3     1 15947000   C   A
4    16  3573172   G   C
5    13 14306989   C   T
6    11  5028652   C   A
> |
```

`head(mutations)` lets you look at the first few lines of the data we just read.

See that mutations appears as a variable here

Next, click on mutations

The console panel slides down, and we get this nicer view of the data

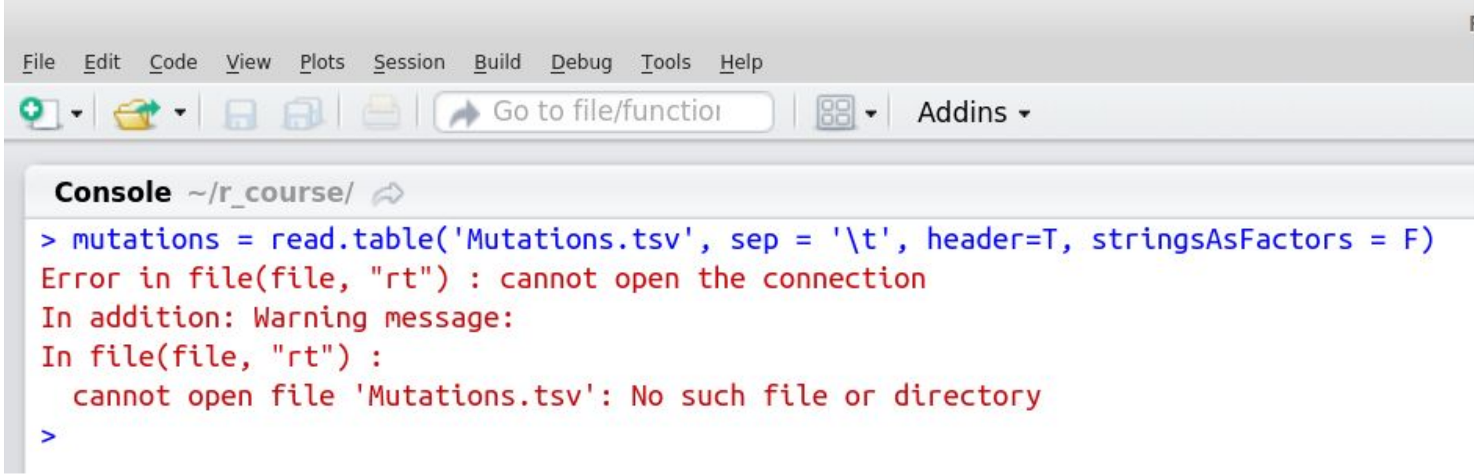

RStudio
File Edit Code View Plots Session Build Debug Tools Help
Go to file/function
Addins
mutations ×
Filter
Environment History
Import Dataset
Global Environment
Data
mutations 7479 obs. of 4 variables
Files Plots Packages Help Viewer
New Folder Delete Rename More
Home › ravi › FTC-original › FTC
▲ Name
Size
..
Console ~/r_course/

	chrom	pos	ref	alt
1	5	20578198	C	T
2	2	4642922	T	A
3	1	15947000	C	A
4	16	3573172	G	C
5	13	14306989	C	T
6	11	5028652	C	A
7	4	6181492	T	A
8	22	2515857	G	C

Showing 1 to 9 of 7,479 entries

```
> setwd('/home/brian/r_course')
> mutations = read.table('mutations.tsv', sep = '\t', header = T, stringsAsFactors = F)
```

What if we mess up?

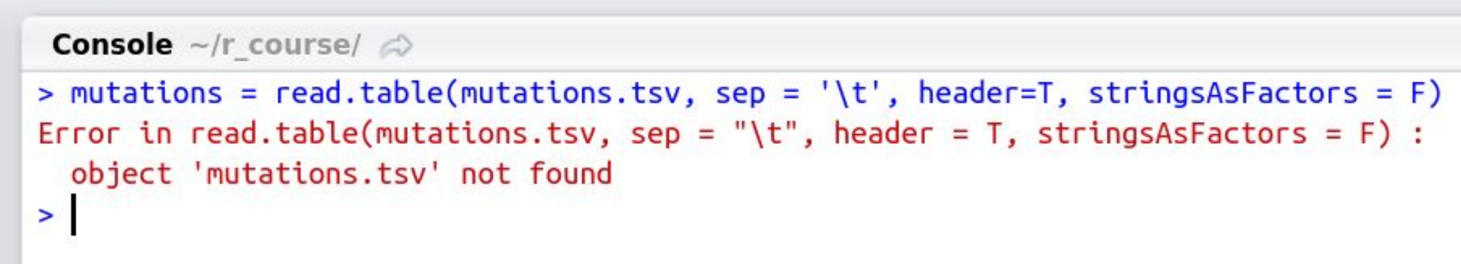


```
> mutations = read.table('Mutations.tsv', sep = '\t', header=T, stringsAsFactors = F)
Error in file(file, "rt") : cannot open the connection
In addition: Warning message:
In file(file, "rt") :
  cannot open file 'Mutations.tsv': No such file or directory
>
```

Here, we made a mistake and capitalized the first letter, typing “Mutations.tsv” instead of “mutations.tsv”. R is case-sensitive, so it says it could not find that file. Correct the file name and try again.

Pro-tip:
Hit the UP arrow on your keyboard and it will go back to your previous commands. This way you do not have to type it all again

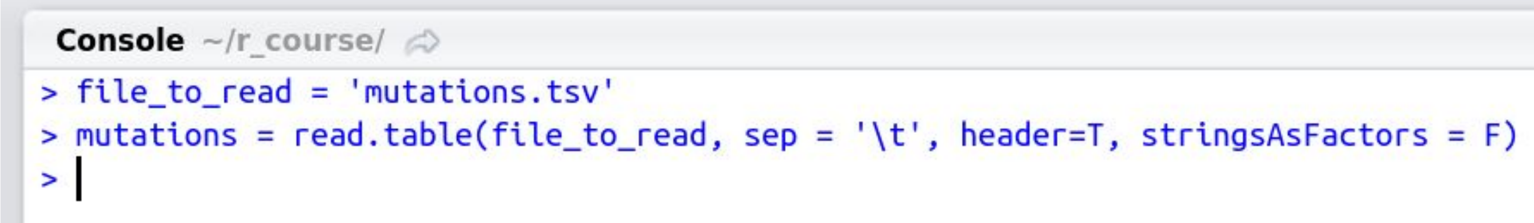
A subtle mistake



```
> mutations = read.table(mutations.tsv, sep = '\t', header=T, stringsAsFactors = F)
Error in read.table(mutations.tsv, sep = "\t", header = T, stringsAsFactors = F) :
  object 'mutations.tsv' not found
> 
```

Here, we did not put mutations.tsv inside quotation marks. Without the quotes, R looks for a variable named `mutations.tsv`, which does not exist. Therefore, it gives an error-- R does not know which file to read.

Sometimes, you might see this:



```
> file_to_read = 'mutations.tsv'
> mutations = read.table(file_to_read, sep = '\t', header=T, stringsAsFactors = F)
> 
```

This works because `file_to_read` is a variable which holds the value `‘mutations.tsv’`. This time, `read.table` can find a variable named `file_to_read`. R effectively replaces `file_to_read` with ‘mutations.tsv’.

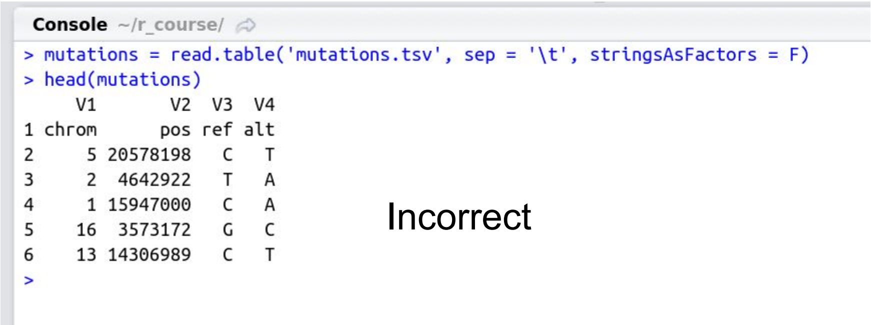
What if we mess up?

Sometimes the command is technically correct (no errors), but gives you something do not want.

For example:
Note that if we leave out the `header=T` argument (top panel), R assumes `header=F`.

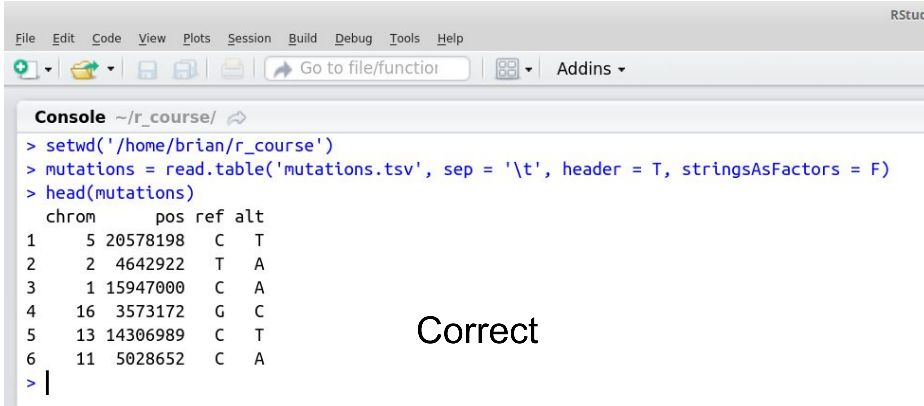
Since it assumes there is no header line, R incorrectly reads the column names as if they were real data! In row 1, you can see our column names. Instead, R gives default names of `V1,...,V4`.

Always check these sort of things- they are not true errors, but they can give unexpected results.



```
> mutations = read.table('mutations.tsv', sep = '\t', stringsAsFactors = F)
> head(mutations)
     V1       V2  V3  V4
1 chrom      pos ref alt
2     5 20578198   C   T
3     2  4642922   T   A
4     1 15947000   C   A
5    16  3573172   G   C
6    13 14306989   C   T
>
```

Incorrect



```
> setwd('/home/brian/r_course')
> mutations = read.table('mutations.tsv', sep = '\t', header = T, stringsAsFactors = F)
> head(mutations)
  chrom      pos ref alt
1     5 20578198   C   T
2     2  4642922   T   A
3     1 15947000   C   A
4    16  3573172   G   C
5    13 14306989   C   T
6    11  5028652   C   A
> |
```

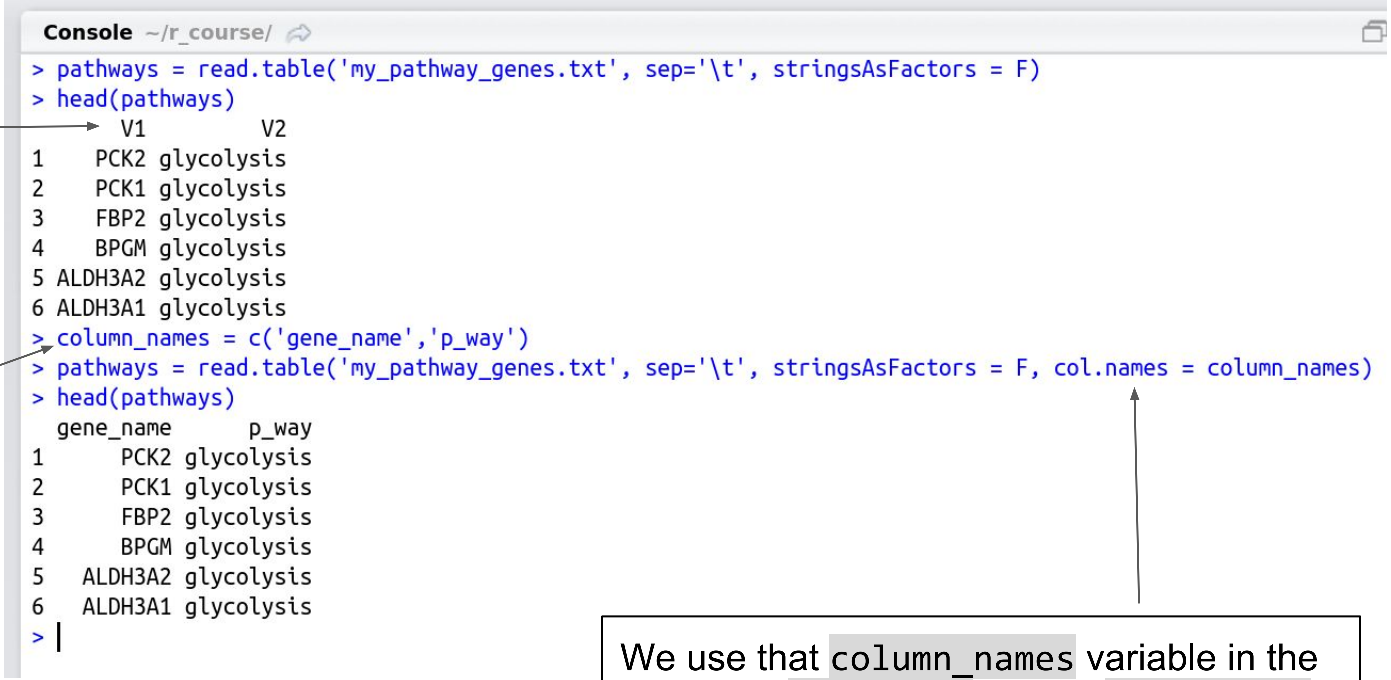
Correct

What about the pathway file, which does NOT have column names?

Default V1, V2 column names. Not very descriptive, so not ideal

`c('x', 'y', 'z')` makes a list out of everything inside the parentheses.

We define the column names we want in the `column_names` variable. We know we have two columns, so this should have two values.



```
> pathways = read.table('my_pathway_genes.txt', sep='\t', stringsAsFactors = F)
> head(pathways)
       V1         V2
1    PCK2 glycolysis
2    PCK1 glycolysis
3    FBP2 glycolysis
4    BPGM glycolysis
5 ALDH3A2 glycolysis
6 ALDH3A1 glycolysis
> column_names = c('gene_name','p_way')
> pathways = read.table('my_pathway_genes.txt', sep='\t', stringsAsFactors = F, col.names = column_names)
> head(pathways)
  gene_name      p_way
1      PCK2 glycolysis
2      PCK1 glycolysis
3      FBP2 glycolysis
4      BPGM glycolysis
5   ALDH3A2 glycolysis
6   ALDH3A1 glycolysis
> |
```

We use that `column_names` variable in the argument `col.names`, so the `read.table` function knows how to name the columns

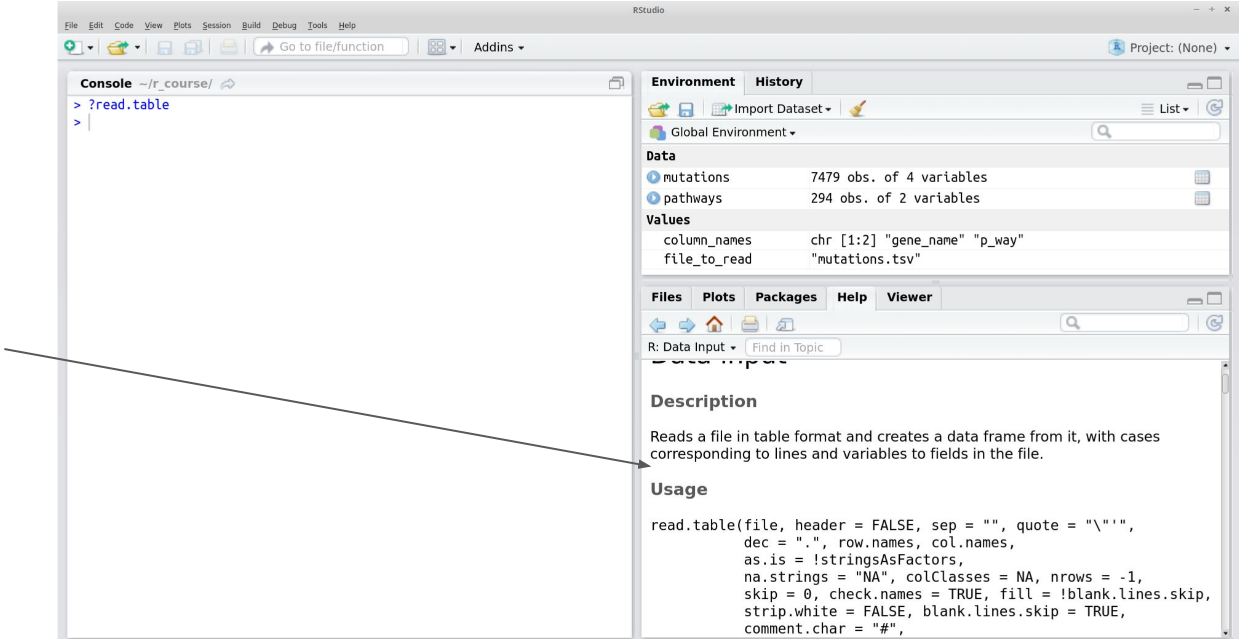
Getting help.

What if you forgot how to specify column names? It's impossible to remember all the different arguments to R's functions. To get help, you can Google, or use R's built-in help.

To get help about the `read.table` function, type:
`?read.table`

Note that the bottom right panel changes its tab to "Help" and you see a description, usage, etc.

Scroll through that-- it's a very detailed description of all the possible arguments. Often quite overwhelming, so Google is sometimes easier/quicker



```
> ?read.table
>
```

Data

mutations	7479 obs. of 4 variables
pathways	294 obs. of 2 variables

Values

column_names	chr [1:2] "gene_name" "p_way"
file_to_read	"mutations.tsv"

Data Input

Description

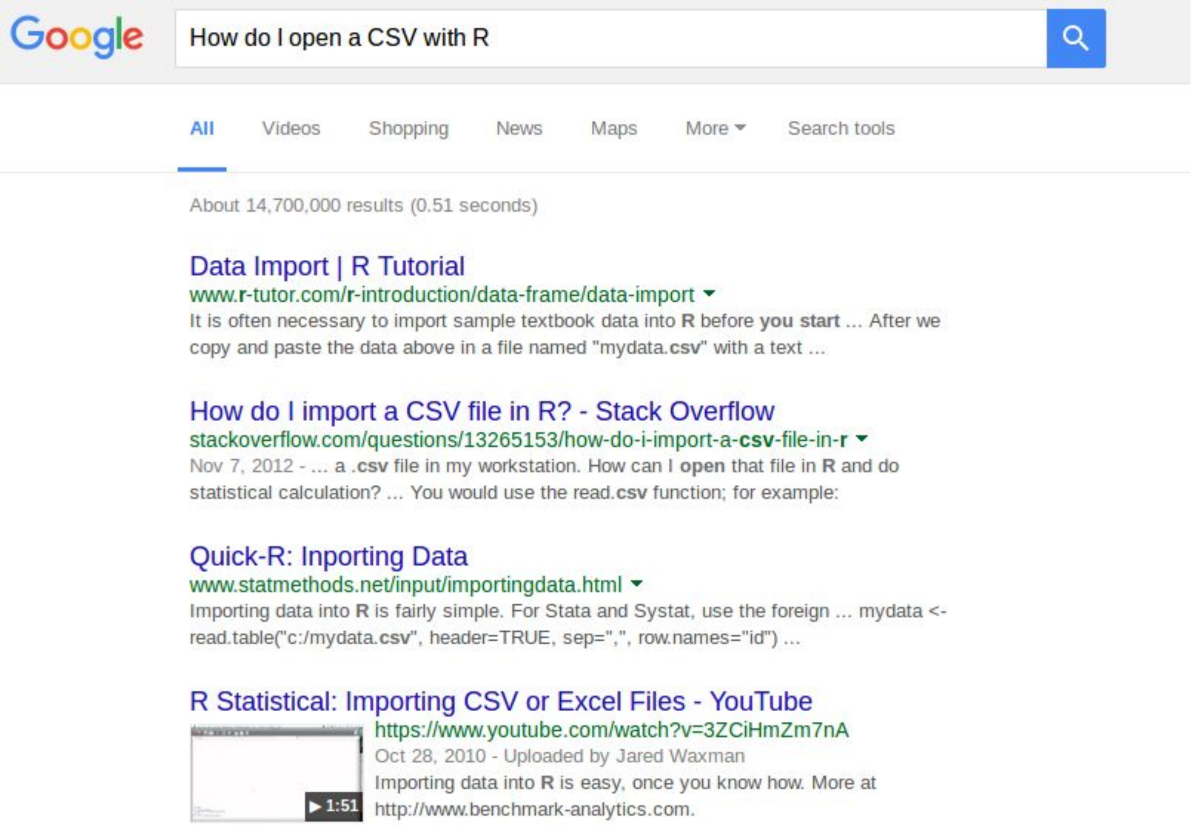
Reads a file in table format and creates a data frame from it, with cases corresponding to lines and variables to fields in the file.

Usage

```
read.table(file, header = FALSE, sep = "", quote = "\"'",
           dec = ".", row.names, col.names,
           as.is = !stringsAsFactors,
           na.strings = "NA", colClasses = NA, nrows = -1,
           skip = 0, check.names = TRUE, fill = !blank.lines.skip,
           strip.white = FALSE, blank.lines.skip = TRUE,
           comment.char = "#",
```

I’m really stuck. Now what?

Google it. Someone has probably already asked the same question



How do I open a CSV with R

About 14,700,000 results (0.51 seconds)

Data Import | R Tutorial
www.r-tutor.com/r-introduction/data-frame/data-import
It is often necessary to import sample textbook data into **R** before **you start** ... After we copy and paste the data above in a file named "mydata.**csv**" with a text ...

How do I import a CSV file in R? - Stack Overflow
stackoverflow.com/questions/13265153/how-do-i-import-a-**csv**-file-in-**r**
Nov 7, 2012 - ... a .**csv** file in my workstation. How can I **open** that file in **R** and do statistical calculation? ... You would use the read.**csv** function; for example:

Quick-R: Inporting Data
www.statmethods.net/input/importingdata.html
Importing data into **R** is fairly simple. For Stata and Systat, use the foreign ... mydata <- read.table("c:/mydata.**csv**", header=TRUE, sep=",", row.names="id") ...

R Statistical: Importing CSV or Excel Files - YouTube
https://www.youtube.com/watch?v=3ZCiHmZm7nA
Oct 28, 2010 - Uploaded by Jared Waxman
Importing data into **R** is easy, once you know how. More at http://www.benchmark-analytics.com.

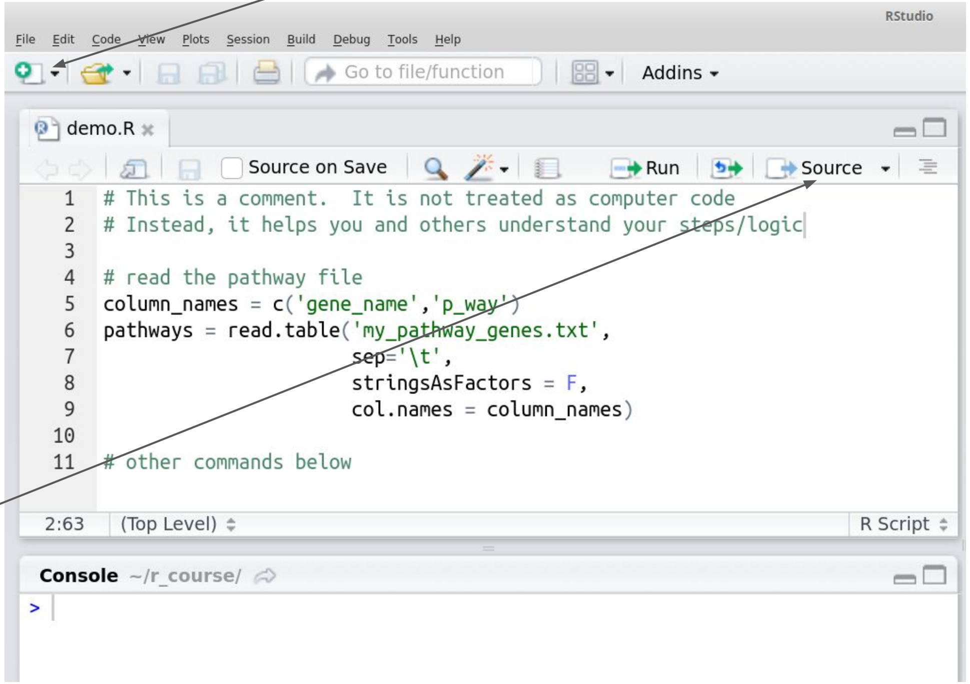
Writing your code

Up to this point, we have been typing into the “live” R console window for demonstration purposes. That is fine for small things and for testing your commands. However, you will usually want to save all your commands into a R “script”.

As shown in the RStudio video, you can create a new R script file and type the commands into there. Once you are sure your command works in the live console window, you can copy/paste into the file and then save it.

After saving, click “Source” to run all your commands.

Click to create new R script



```
# This is a comment.  It is not treated as computer code
# Instead, it helps you and others understand your steps/logic

# read the pathway file
column_names = c('gene_name','p_way')
pathways = read.table('my_pathway_genes.txt',
                      sep='\t',
                      stringsAsFactors = F,
                      col.names = column_names)

# other commands below
```

Exercise (You MUST do this before class):

Read all of the data files into R. Create a new script file (save as workshop.R) to perform all steps below. Execute by clicking “source”...there will most likely be errors-- nobody is perfect! Try to figure out what went wrong and how to fix it.

Answers are on the next slide, but please try your best!

1. differential_results.csv (call this `dge_results`)
2. gene_expression.tsv (call this `expressions`)
3. gene_annotations.tsv (call this `annotations`)
4. my_pathway_genes.txt (call this `pathways`)
    a. Name the columns as “gene_name”, “p_way”)
5. Mutations.tsv (call this `mutations`)

Save as the named variables so we are all using consistent names

The answer --->

Please type this code into your RStudio file editor and save it as “workshop.R”

Be careful if you copy/paste from this PDF. Sometimes it can copy hidden things which can cause very strange errors in R

Make sure it runs (click “Source”) correctly before class

```
dge_results <- read.table('differential_results.csv',
                        sep=',',
                        header=T,
                        stringsAsFactors=F)

expressions = read.table('gene_expression.tsv',
                        sep='\t',
                        header=T,
                        stringsAsFactors=F)

annotations = read.table('gene_annotations.tsv',
                        sep='\t',
                        header=T,
                        stringsAsFactors=F)

column_names = c('gene_name','p_way')
pathways = read.table('my_pathway_genes.txt',
                        sep='\t',
                        col.names=column_names,
                        stringsAsFactors=F)

mutations = read.table('mutations.tsv',
                        sep='\t',
                        header=T,
                        stringsAsFactors=F)
```