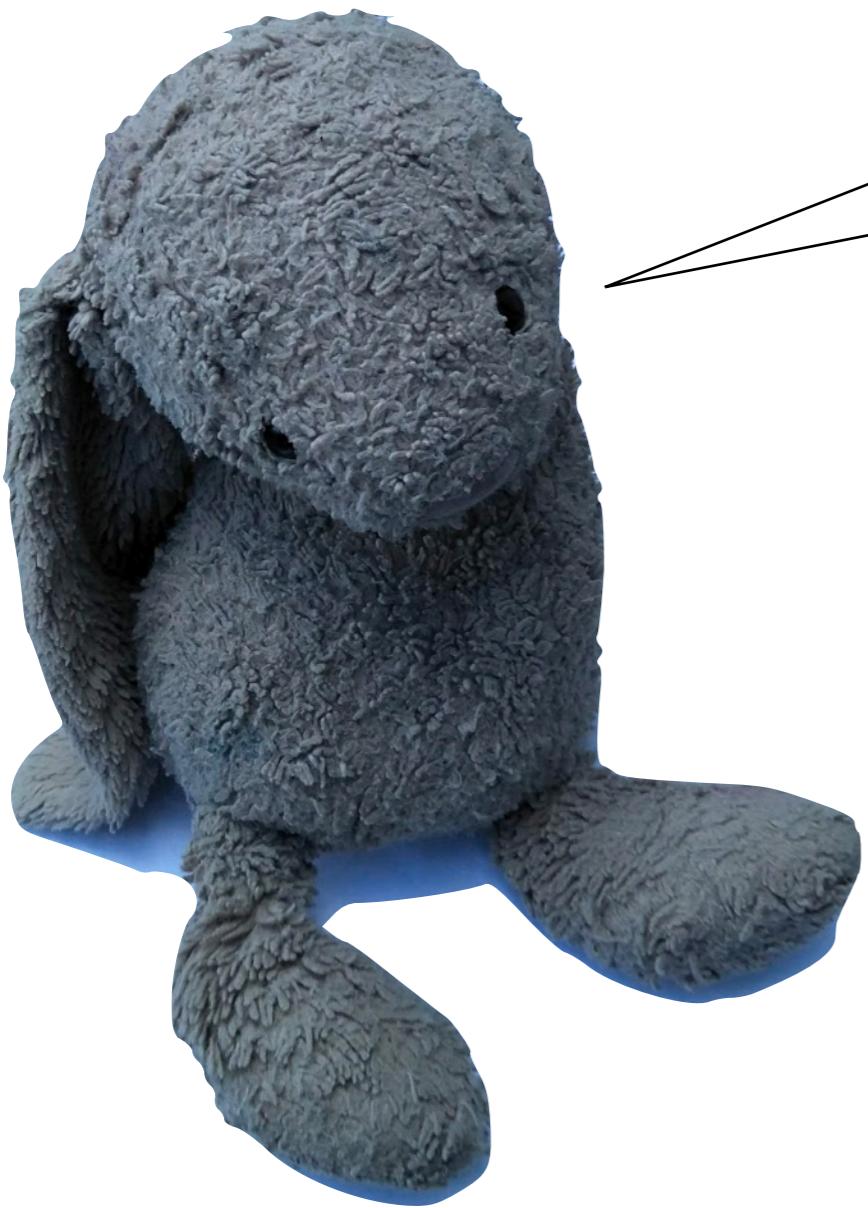


# **Basics of R: Loading and saving data**

Research Methods for Human Inquiry  
Andrew Perfors

# First, a brief discussion of folder organisation



Here's mine. Please don't laugh.

# First, a brief discussion of folder organisation



# First, a brief discussion of folder organisation

Mine are better. They're all in a folder, like this!

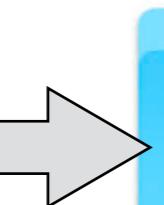


# First, a brief discussion of folder organisation

- 📄 mcmcsamples\_expt2\_subj19\_scenario2.mat
- 📄 mcmcsamples\_expt2\_subj19\_scenario3.mat
- 📄 mcmcsamples\_expt2\_subj20\_scenario1.mat
- 📄 mcmcsamples\_expt2\_subj20\_scenario2.mat
- 📄 mcmcsamples\_expt2\_subj20\_scenario3.mat
- 📄 modelfits1.mat
- 📄 modelfits1 copy.mat
- 📄 modelfits2.mat
- 📄 modelfits2 copy.mat
- 📄 parameterdescriptives.m
- ▀ participant1scenario3expt1.eps
- ▀ participant4scenario3expt1.eps
- ▀ participant10scenario3expt1.eps
- ▀ participant13scenario1expt1.eps
- ▀ participant15scenario1expt1.eps
- ▀ participant19scenario3expt1.eps
- 📄 recalculateVAF.m
- 📄 recalculateVAF.m~
- 📄 showtestcases.m
- 📄 stupidplots.m
- 📄 stupidplots.m~
- 📄 stupidplots2.m
- 📄 stupidplots2.m~

# It is worth doing this right from the start!

Desktop or  
Documents



rmhi



lectures



week1



week2



week3

...



tutorials



tuteweek2



tuteweek3

...



exercises



week1



week2



week3

...



assessments

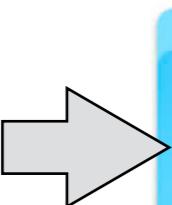


documents

Make sure you are not using OneDrive or some other cloud folder! It's easy for you or R to get confused about where things are if you do this.

# It is worth doing this right from the start!

Desktop or  
Documents



rmhi



lectures



week1



week2



week3

...



tutorials



tuteweek2



tuteweek3

...



exercises



week1



week2



week3

...



assessments



documents

I also suggest  
all file names  
avoid spaces  
(I like snake  
case):

andy\_file.pdf

# Projects

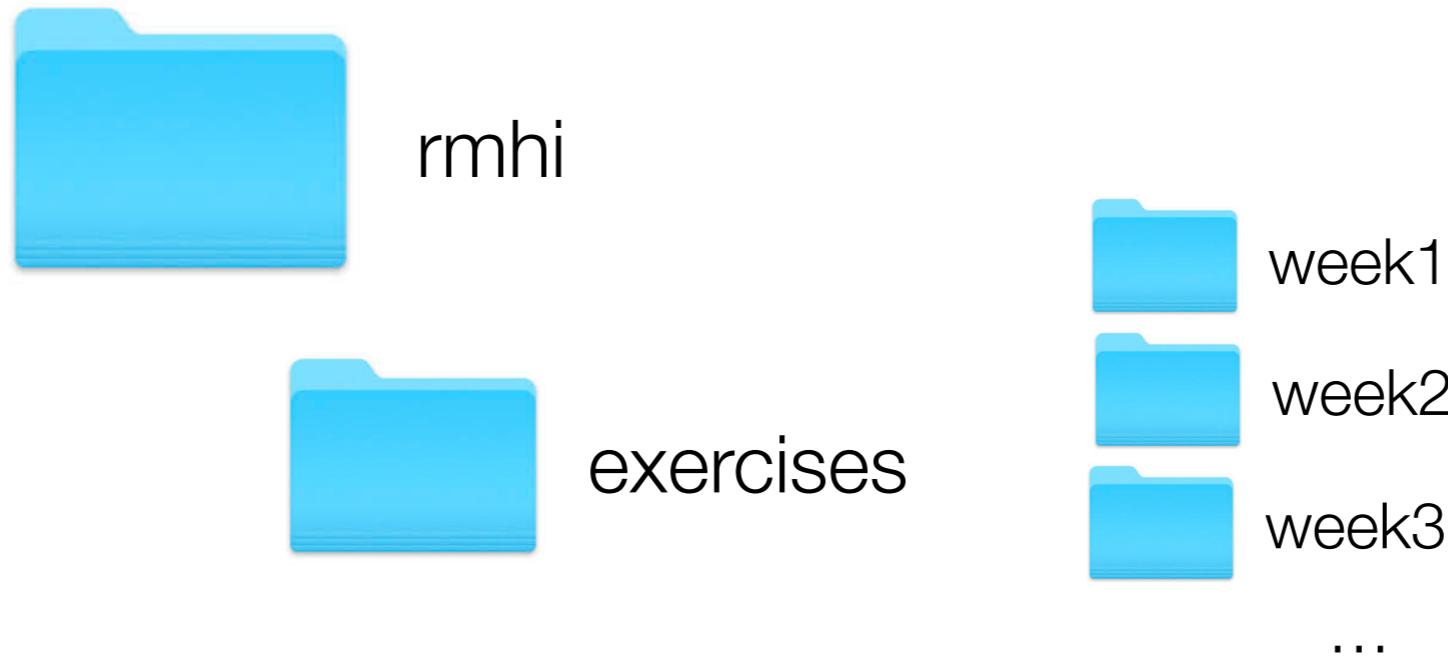
- Let you easily switch between different things you might be working on, and keeps track when there are different data sets associated with the same thing
- When used in conjunction with the “here” package, makes it really easy to find files and share with others

## Basic idea:

- Creates a .Rproj file at the “root”, which serves as a useful anchor for other packages. Lets you easily define the location of files, share things, etc.
- Don’t need to keep hunting around your directories!

# Using a project

I'm going to assume you have the folder organisation I suggested earlier:

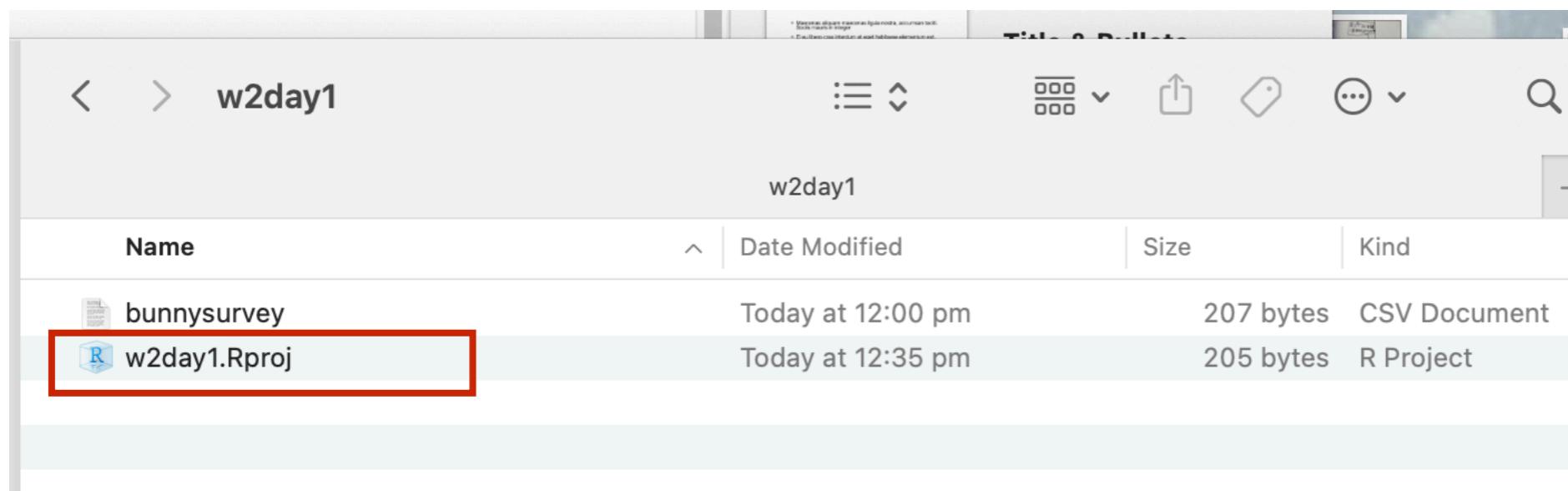


There is a zipped folder called `w2day1` which was provided with this video. Download it, unzip it, and open it in your `rmhi/exercises/week2` folder. It should create a folder called `w2day1` in the `week2` folder.

You'll notice it has an `w2day1.Rproj` icon. I created that (it's not too hard to create but beyond the scope of this subject - I'll always create them for you).

# Let's test it...

Click on the `w2day1.Rproj` icon which you can see in the `rmhi/exercises/week2/w2day1/` folder.



It should open up RStudio for you, and automatically put you in the right folder. For the tutorials, exercises, and assessments I'll be giving you files in zipped folders which each have an associated Rproj. You just download the file and **always open R by clicking on the Rproj!!**

# Loading datasets

- R can handle data in a lot of formats (Excel, csv, SPSS, etc)
  - The proprietary format for R is called Rdata. These files are saved workspaces and can be loaded using a command called `load()`
  - Rdata files contain whatever data sets, variables, functions etc that the workspace included when the file was created
  - We will *not* be using Rdata files in this subject, because most datasets used generally are csv (comma separated value). But loading all files follows similar principles and you can google for the specific commands if you have different kinds

# The “here” package

- Finds the “root” of your current project based on the RProj file, and makes it so you never have to search for your data again

```
> library(here)
> loc <- here("bunnysurvey.csv")
```

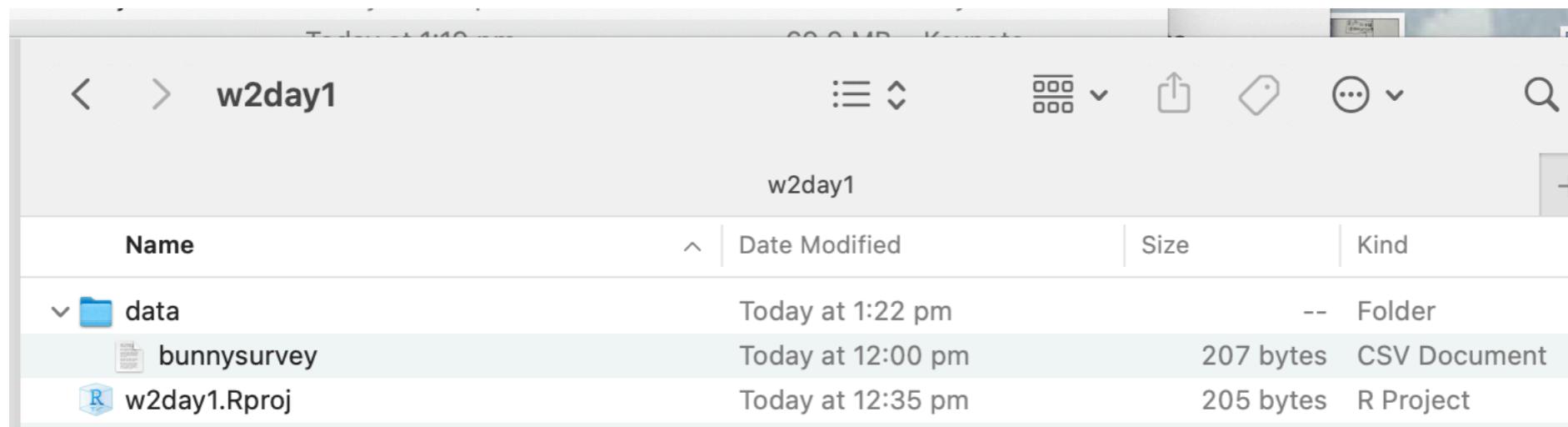
The function “here” gives the location of the file inserted as an argument. In this case, we assign that location to a variable called `loc`. We can look at the value of `loc`:

```
> loc
[1] "/Users/andyperfors/Documents/teaching/rmhi/exercises/week2/w2day1/
bunnysurvey.csv"
```

This contains the “path”, which is basically the address your computer uses to find things. The nice thing about `here()` is that it keeps track of this for you and you don’t need to worry about it!

# The “here” package

What if bunnysurvey.csv wasn’t in the same place as our Rproj? Suppose it was in a different folder called “data”?



Then you just add the folder name as part of the path

```
> loc <- here("data/bunnysurvey.csv")
```

```
> loc
[1] "/Users/andyperfors/Documents/teaching/rmhi/exercises/week2/w2day1/
data/bunnysurvey.csv"
```

# Stop being silly and read the file!

So far we've only told the computer where to look for the file. We haven't read it in yet. To do that we use the `read_csv()` function, which is in the `tidyverse` package.

```
> library(tidyverse)  
> data <- read_csv(file=loc)
```

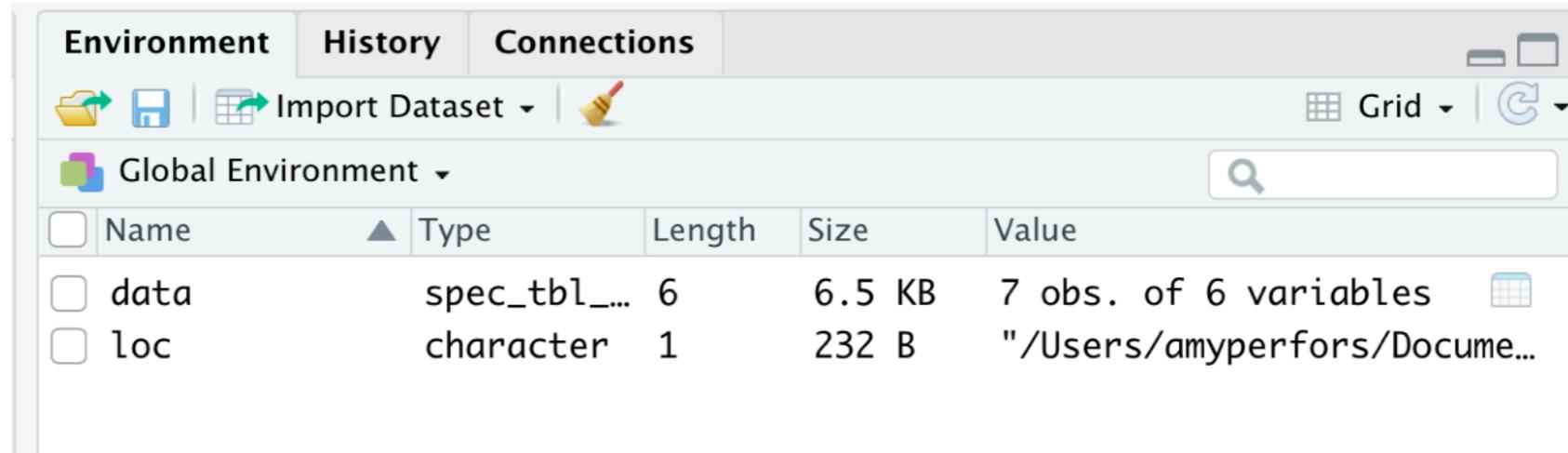
Here we're telling it to find the file that's at the location `loc`. It puts that into a variable called `data`

# We can now do all sorts of things with the data

But that will be the next video! For now, just make sure it's there

```
> data <- read_csv(file=loc)

— Column specification ——————
cols(
  name = col_character(),
  colour = col_character(),
  height = col_double(),
  bunnyrank = col_double(),
  bearrank = col_double(),
  doggyrank = col_double()
)
```



The screenshot shows the RStudio interface with the 'Environment' tab selected. The global environment contains two objects: 'data' and 'loc'. The 'data' object is a CSV file with 7 observations and 6 variables, while 'loc' is a character string representing the file path.

Name	Type	Length	Size	Value
data	spec_tbl_df	6	6.5 KB	7 obs. of 6 variables
loc	character	1	232 B	"/Users/amyperfors/Docume..."

# Writing your dataset to a file

You can use the `write_csv()` function to save your dataset

```
> write_csv(data, "bunnysurveynew.csv")
```

The variable corresponding to the dataset

The name of the file you want to write it to (if it's the same as an existing name it will write over it - be careful here!)

Automatically puts it in the current folder  
(where your Rproj is)

No exercises for this one, as long as  
you've managed to successfully load  
[`bunnysurvey.csv`](#) and write  
[`bunnysurveynew.csv`](#)