

# The structure of data

## Research Methods and Skills

25/10/2022

# Writing R Scripts

Scripts are text documents that contain a sequence of commands to be executed sequentially.

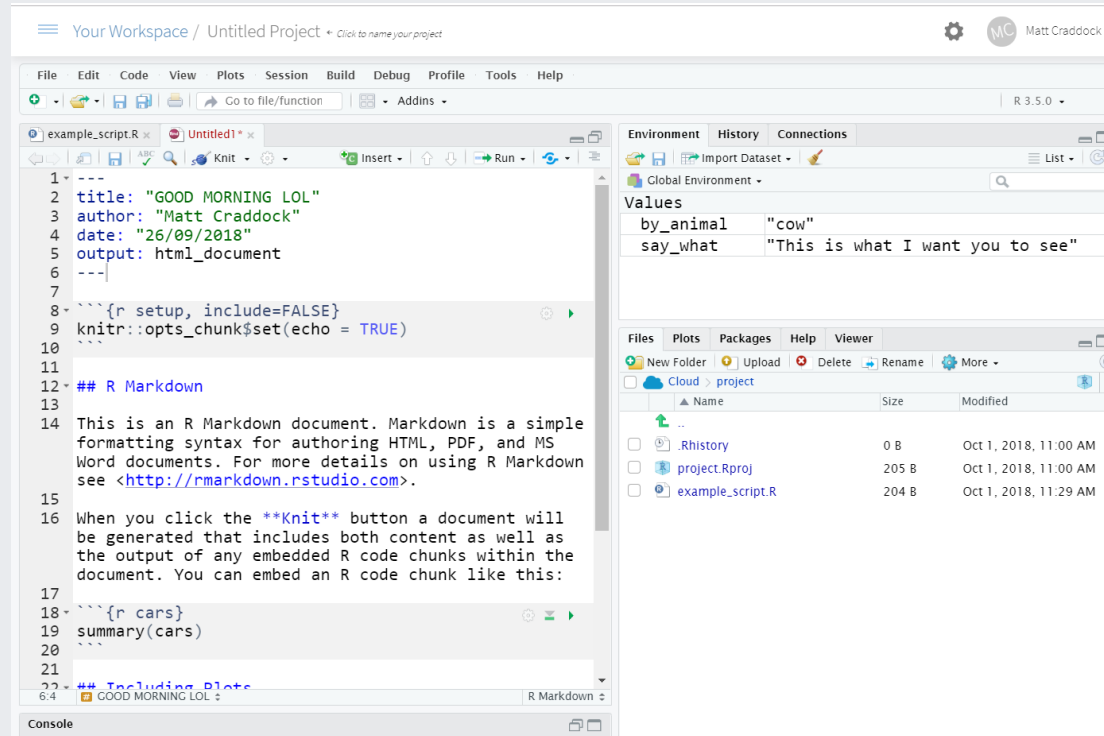
A typical script looks something like this:

```
# Load in required packages using library()  
library(tidyverse)  
  
# Define any custom functions here (we haven't covered this!)  
  
# Now load any data you want to work on. (again, we'll cover this later!)  
test_data <-  
  read_csv("data/a-random-RT-file.csv") %>% # I'll explain what %>% means later  
  rename(RT = `reaction times`)  
  
# The rest of the script then runs whatever analyses or plotting you want to do  
ggplot(test_data,  
       aes(x = RT,  
           fill = viewpoint)) +  
  geom_density()
```

# RMarkdown

RMarkdown documents contain a mixture of code and plain text.

They can be used to produce *reports* and fully formatted documents with whatever structure you choose.



# Basic data types

There are five basic data types in R:

Type	Description	Examples
integer	Whole numbers	1, 2, 3
numeric	Any real number, fractions	3.4, 2, -2.3
character	Text	"Hi there", "8.5", "ABC123"
logical	Assertion of truth/falsity	TRUE, FALSE
complex	Real and imaginary numbers	0.34+5.3i

# Containers

**Vectors** are one-dimensional collections of values of the same basic data type.

**Matrices** are two-dimensional collections of values of the same basic data type.

**Lists** are collections of objects of varying length and type.

**Data frames** are tables of data.



# Accessing elements from containers

You can use the `[]` operator after the name of an object to extract individual elements from that object.

```
one_to_four
```

```
##      Monday    Tuesday Wednesday  Thursday
##           1           2           3           4
```

```
test_matrix
```

```
##           [,1]      [,2]      [,3]
## [1,] -0.8354471  0.8782189 -0.01441798
## [2,] -1.3572108  1.2522531 -0.11939396
## [3,]  0.9219538  0.5992309  0.69609389
```

```
one_to_four["Wednesday"]
```

```
## Wednesday
##           3
```

```
test_matrix[2:3, 1:2]
```

```
##           [,1]      [,2]
## [1,] -1.3572108  1.2522531
## [2,]  0.9219538  0.5992309
```



# Tidyverse



The **tidyverse** is a collection of packages that expand R's functions in a structured, coherent way.

```
install.packages("tidyverse")
```

There are eight core **tidyverse** packages loaded using **library(tidyverse)**.

- ggplot2
- **tidyr**
- dplyr
- **tibble**
- purrr
- readr
- stringr
- forcats



# Tidyverse



You can load all these packages at once.

```
library(tidyverse) # This loads all the tidyverse packages at once
```

You can also load each one individually. We'll be using the **tibble** package next.

```
library(tibble)
```

Many of the *tidyverse* packages create or output *tibbles*, which are essentially a more user-friendly version of data frames.

# Tibbles

You can create a *tibble* similar to how you create a data frame, using **tibble()**.

```
age_tibb <-  
  tibble(Participant = 1:10,  
         cond1 = rnorm(10),  
         age_group = rep(c("Old", "Young"),  
                        each = 5))  
head(age_tibb)
```

```
## # A tibble: 6 × 3  
##   Participant  cond1 age_group  
##       <int>   <dbl>   <chr>  
## 1         1 -0.804 Old  
## 2         2  0.728 Old  
## 3         3 -0.144 Old  
## 4         4  0.905 Old  
## 5         5  1.07  Old  
## 6         6  0.134 Young
```

# Tibbles

```
age_tibb <-  
  tibble(Participant = 1:10,  
         cond1 = rnorm(10),  
         age_group = rep(c("Old", "Young"), each = 5))
```

Here I used the **rep()** function to generate a character vector with the values "Old" and "Young".

```
rep(c("Old", "Young"), each = 5)
```

```
## [1] "Old" "Old" "Old" "Old" "Old" "Young" "Young" "Young" "Young"  
## [10] "Young"
```

```
rep(c("Old", "Young"), 5)
```

```
## [1] "Old" "Young" "Old" "Young" "Old" "Young" "Old" "Young" "Old"  
## [10] "Young"
```

# Importing data into R

# Different types of file

Data comes in many different shapes, sizes, and formats.

The most common file formats you'll deal with are either Excel files or text files, but you may also find dealing with SPSS files useful.

Fortunately, R has several functions and packages for importing data!

File formats	File extension	Functions	Package
SPSS	.sav	<b>read_sav()</b>	library(haven)
Excel	.xls, .xlsx	<b>read_excel()</b>	library(readxl)
Text	.csv, .txt, .*	<b>read_csv(), read_delim()</b>	library(readr)

# Comma-separated values

[illegible]

# Excel spreadsheets

prison-population-data-tool-31-december-2017 - Saved

File Home Insert Page Layout Formulas Data Review View Help Tell me what you want to do

Calibri 11 A A

B I U - A -

Clipboard Font Alignment

Wrap Text

General

Number

Conditional Formatting

Normal Bad Good

Neutral Calculation Check Cell

Styles

A18 a Establishment\*Sex\*Age Group

	A	B	C	D	E	F
	View	Date	Establishment	Sex	Age / Custody / Nationality / Offence Group	Population
1	a Establishment*Sex*Age Group	2015-06	Altcourse	Male	Adults (21+)	
2	a Establishment*Sex*Age Group	2015-06	Altcourse	Male	Juveniles and Young Adults (15-20)	
3	a Establishment*Sex*Age Group	2015-06	Ashfield	Male	Adults (21+)	
4	a Establishment*Sex*Age Group	2015-06	Ashham Grange	Female	Adults (21+)	
5	a Establishment*Sex*Age Group	2015-06	Ashham Grange	Female	Juveniles and Young Adults (15-20)	
6	a Establishment*Sex*Age Group	2015-06	Aylesbury	Male	Adults (21+)	
7	a Establishment*Sex*Age Group	2015-06	Aylesbury	Male	Juveniles and Young Adults (15-20)	
8	a Establishment*Sex*Age Group	2015-06	Bedford	Male	Adults (21+)	
9	a Establishment*Sex*Age Group	2015-06	Bedford	Male	Juveniles and Young Adults (15-20)	
10	a Establishment*Sex*Age Group	2015-06	Belmarsh	Male	Adults (21+)	
11	a Establishment*Sex*Age Group	2015-06	Belmarsh	Male	Juveniles and Young Adults (15-20)	
12	a Establishment*Sex*Age Group	2015-06	Birmingham	Male	Adults (21+)	1
13	a Establishment*Sex*Age Group	2015-06	Brinsford	Male	Adults (21+)	
14	a Establishment*Sex*Age Group	2015-06	Brinsford	Male	Juveniles and Young Adults (15-20)	
15	a Establishment*Sex*Age Group	2015-06	Bristol	Male	Adults (21+)	
16	a Establishment*Sex*Age Group	2015-06	Bristol	Male	Juveniles and Young Adults (15-20)	
17	a Establishment*Sex*Age Group	2015-06	Brixton	Male	Adults (21+)	
18	a Establishment*Sex*Age Group	2015-06	BronzeField	Female	Adults (21+)	
19	a Establishment*Sex*Age Group	2015-06	BronzeField	Female	Juveniles and Young Adults (15-20)	
20	a Establishment*Sex*Age Group	2015-06	Buckley Hall	Male	Adults (21+)	
21	a Establishment*Sex*Age Group	2015-06	Bullington	Male	Adults (21+)	1
22	a Establishment*Sex*Age Group	2015-06	Bullington	Male	Juveniles and Young Adults (15-20)	
23	a Establishment*Sex*Age Group	2015-06	Bure	Male	Adults (21+)	
24	a Establishment*Sex*Age Group	2015-06	Cardiff	Male	Adults (21+)	
25	a Establishment*Sex*Age Group	2015-06	Cardiff	Male	Juveniles and Young Adults (15-20)	
26	a Establishment*Sex*Age Group	2015-06	Channings Wood	Male	Adults (21+)	

# Fear of Crime Dataset

Ellis & Renouf (2018) - the relationship between fear of crime and various personality measures.

Their data is openly available, stored as text in a *comma-separated-values* format (.csv).

Once again, we can use the import button or some code (with **read\_csv()**) to load this data in and automatically format it into a *tibble*.

```
library(readr)
FearofCrime <- read_csv("data/FearofCrime.CSV")
```

See also Ellis & Merdian, 2015, Frontiers in Psychology

# Fear of Crime Dataset

Ellis & Renouf (2018) collected data online using Qualtrics.

The file contains one column for each question that the participants answered, for a total of 169(!) columns.



Each row represent a single participant's responses, and their demographic information.

```
FearofCrime
```




```
## # A tibble: 301 × 169
##   ResponseID Respo...1 Name Exter...2 Status Start...3 EndDate Finis...4 Conse...5 sex
##   <chr> <chr> <chr> <lgl> <dbl> <chr> <chr> <dbl> <dbl> <dbl>
## 1 R_ai4tgG1... Defaul... Anon... NA 0 19/10/... 19/10/... 1 1 2
## 2 R_d50iATV... Defaul... Anon... NA 0 20/10/... 20/10/... 1 1 2
## 3 R_aaBVZUe... Defaul... Anon... NA 0 20/10/... 20/10/... 1 1 1
## 4 R_6nxInLK... Defaul... Anon... NA 0 20/10/... 20/10/... 1 1 1
## 5 R_6SCYbh0... Defaul... Anon... NA 0 20/10/... 20/10/... 1 1 2
## 6 R_5pCxWA6... Defaul... Anon... NA 0 20/10/... 20/10/... 1 1 1
## 7 R_d1nji6V... Defaul... Anon... NA 0 20/10/... 20/10/... 1 1 2
## 8 R_9v6ZgUh... Defaul... Anon... NA 0 20/10/... 20/10/... 1 1 1
## 9 R_5Bg7VjB... Defaul... Anon... NA 0 20/10/... 20/10/... 1 1 2
## 10 R_9Sv17lQ... Defaul... Anon... NA 0 20/10/... 20/10/... 1 1 2
## # ... with 291 more rows, 159 more variables: age <dbl>, hexaco1 <dbl>,
## # hexaco2 <dbl>, hexaco3 <dbl>, hexaco4 <dbl>, hexaco5 <dbl>, hexaco6 <dbl>,
```




## Spaces

 Your Workspace PSY9219M; Research Methods New Space







## Learn

 Guide Primers DataCamp Courses Cheat Sheets Feedback and Questions

## Info

 Terms and Conditions System Status

File · Edit · Code · View · Plots · Session · Build · Debug · Profile · Tools · Help

     Go to file/function  Addins

R 3.5.0

Console Terminal x Jobs x

/cloud/project/

&gt;

Environment History Connections

  Import Dataset

List

Global Environment







Q

Environment is empty

Files Plots Packages Help Viewer

 New Folder  Upload  Delete  Rename  More

Cloud &gt; project

	Name	Size	Modified
	..		
<input type="checkbox"/> 	.Rhistory	0 B	Oct 21, 2018, 10:47 I
<input type="checkbox"/> 	data		
<input type="checkbox"/> 	project.Rproj	205 B	Oct 22, 2018, 10:01 ,
<input type="checkbox"/> 	scripts		
<input type="checkbox"/> 	solved		

## Spaces

Your Workspace

PSY9219M; Research Method

+ New Space

## Learn

Guide

Primers

DataCamp Courses

Cheat Sheets

Feedback and Questions

## Info

Terms and Conditions

System Status

File · Edit · Code · View · Plots · Session · Build · Debug · Profile · Tools · Help

Go to file/function Addins

R 3.5.0

Console

/cloud/proj

## Import Text Data

File/Url:

Browse...

Data Preview:

Import Options:

Name: dataset

Skip: 0

☒ First Row as Names

Delimiter: Comma

Escape: None

☒ Trim Spaces

Quotes: Default

Comment: Default

☒ Open Data Viewer

Locale: Configure...

NA: Default

Code Preview:

```
library(readr)
dataset <- read_csv('data.csv')
View(dataset)
```

Reading rectangular data using readr

Import

Cancel

Spaces

Your Workspace

PSY9219M; Research Method

New Space

Learn

Guide

Primers

DataCamp Courses

Cheat Sheets

Feedback and Questions

Info

Terms and Conditions

System Status

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

R 3.5.0

Console

/cloud/proj

Import Text Data

File/Url:

Browse...

Data Preview:

Choose File

File name:

/ &gt; cloud &gt; project

..

data

scripts

solved

.Rhistory

0 B Oct 21, 2018, 10:47 PM

project.Rproj

205 B Oct 22, 2018, 10:01 AM

Import Options:

Name: dataset

Skip: 0

☒ Trim Spaces

Quotes: Default

Comment: Default

☒ Open Data Viewer

Locale: Configure...

NA: Default

Open

Cancel

Code Preview:

```
library(readr)
dataset <- read_csv('data/dataset.csv')
View(dataset)
```

Reading rectangular data using readr

Import

Cancel

Spaces

Your Workspace

PSY9219M; Research Method

New Space

Learn

Guide

Primers

DataCamp Courses

Cheat Sheets

Feedback and Questions

Info

Terms and Conditions

System Status

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function

Addins

R 3.5.0

Console

/cloud/proj

Import Text Data

File/Url:

<http://www.research.lancs.ac.uk/portal/files/104824495/FearofCrime.csv>

Update

Data Preview:

ResponseID (character)	ResponseSet (character)	Name (character)	ExternalDataReference (character)	Status (integer)	StartDate (character)	EndDate (character)
R_ai4tgG1GHNdVdqt	Default Response Set	Anonymous	NA	0	19/10/14 21:08	19/10/14 21:26
R_d5OiATV0IjBbMx	Default Response Set	Anonymous	NA	0	20/10/14 12:15	20/10/14 12:27
R_aaBVZUe9mIGiDpH	Default Response Set	Anonymous	NA	0	20/10/14 12:18	20/10/14 12:28
R_6nxlnLKQv2bucQZ	Default Response Set	Anonymous	NA	0	20/10/14 12:18	20/10/14 12:29
R_6SCYbhOP9BG5CgR	Default Response Set	Anonymous	NA	0	20/10/14 12:24	20/10/14 12:32
R_5pCxWA6qOQdnVyd	Default Response Set	Anonymous	NA	0	20/10/14 12:34	20/10/14 12:43

Previewing first 50 entries.

Import Options:

Name: FearofCrime

☒ First Row as Names

Delimiter: Comma

Escape: None

Skip: 0

☒ Trim Spaces

Quotes: Default

Comment: Default

☒ Open Data Viewer

Locale: Configure...

NA: Default

Code Preview:

```
library(readr)
read_csv(
  "http://www.research.lancs.ac.uk/portal/files/104824495/FearofCrime.csv"
)
```

Reading rectangular data using readr

Import

Cancel

Spaces

Your Workspace

PSY9219M; Research Method

New Space

Learn

Guide

Primers

DataCamp Courses

Cheat Sheets

Feedback and Questions

Info

Terms and Conditions

System Status

File Edit Code View Plots Session Build Debug Profile Tools Help



Go to file/function

Addins

R 3.5.0

Console

/cloud/proj

Import Text Data

File/Url:

<http://www.research.lancs.ac.uk/portal/files/104824495/FearofCrime.csv>

Update

Data Preview:

ResponseID (character)	ResponseSet (character)	Name (character)	ExternalDataReference (character)	Status (integer)	StartDate (character)	EndDate (character)
R_ai4tgG1GHNdVdqt	Default Response Set	Anonymous	NA	0	19/10/14 21:08	19/10/14 21:26
R_d5OiATV0IjBbMx	Default Response Set	Anonymous	NA	0	20/10/14 12:15	20/10/14 12:27
R_aaBVZUe9mIGiDpH	Default Response Set	Anonymous	NA	0	20/10/14 12:18	20/10/14 12:28
R_6nxlnLKQv2bucQZ	Default Response Set	Anonymous	NA	0	20/10/14 12:18	20/10/14 12:29
R_6SCYbhOP9BG5CgR	Default Response Set	Anonymous	NA	0	20/10/14 12:24	20/10/14 12:32
R_5pCxWA6qOQdnVyd	Default Response Set	Anonymous	NA	0	20/10/14 12:34	20/10/14 12:43

Previewing first 50 entries.

Import Options:

Name: FearofCrime

☒ First Row as Names

Delimiter: Comma

Escape: None

Skip: 0

☒ Trim Spaces

Quotes: Default

Comment: Default

☒ Open Data Viewer

Locale: Configure...

NA: Default

Code Preview:

```
library(readr)
read_csv(
  "http://www.research.lancs.ac.uk/portal/files/104824495/FearofCrime.csv")
```

Reading rectangular data using readr

Import

Cancel

## Spaces

Your Workspace

PSY9219M; Research Method

New Space

## Learn

Guide

Primers

DataCamp Courses

Cheat Sheets

Feedback and Questions

## Info

Terms and Conditions

System Status

File Edit Code View Plots Session Build Debug Profile Tools Help

FearofCrime x Filter

	ResponseID	ResponseSet	Name	ExternalDataReference	Status
1	R_ai4tgG1GHNdVdqt	Default Response Set	Anonymous	NA	0
2	R_d5OiATV0Ii8bMx	Default Response Set	Anonymous	NA	0
3	R_aaBVZUe9mIGiDpH	Default Response Set	Anonymous	NA	0
4	R_6nXlnLKQv2bucQZ	Default Response Set	Anonymous	NA	0
5	R_6SCYbhOP9BG5CgR	Default Response Set	Anonymous	NA	0
6	R_5pCxWA6qOQdnVyd	Default Response Set	Anonymous	NA	0
7	R_d1wii6V75Cnn0v	Default Response Set	Anonymous	NA	0

Showing 1 to 8 of 301 entries

Console Terminal x Jobs x

```
/cloud/project/
> library(readr)
> FearofCrime <- read_csv("http://www.research.lancs.ac.uk/portal/files/104824495/FearofCrime.csv")
Parsed with column specification:
cols(
  .default = col_integer(),
  ResponseID = col_character(),
  ResponseSet = col_character(),
  Name = col_character(),
  ExternalDataReference = col_character(),
  StartDate = col_character(),
  EndDate = col_character(),
  hexaco_First_Click = col_double(),
  hexaco_Last_Click = col_double(),
  hexaco_Page_Submit = col_double(),
```

Environment History Connections

Global Environment

## Data

FearofCrime 301 obs. of 169 variables

Files Plots Packages Help Viewer

New Folder Upload Delete Rename More

Cloud &gt; project

	Name	Size	Modified
<input type="checkbox"/>	..		
<input type="checkbox"/>	.Rhistory	0 B	Oct 21, 2018, 10:47 F
<input type="checkbox"/>	data		
<input type="checkbox"/>	project.Rproj	205 B	Oct 22, 2018, 10:01 A
<input type="checkbox"/>	scripts		
<input type="checkbox"/>	solved		

Keeping your analyses organised

Untitled 138.docx  
Untitled 241.doc  
Untitled 138 copy.docx  
Untitled 138 copy 2.docx  
Untitled 139.docx  
Untitled 40 MOM ADDRESS.jpg  
Untitled 242.doc  
Untitled 243.doc  
Untitled 243 IMPORTANT.doc  
Untitled 41.jpg



PRO TIP: NEVER LOOK IN SOMEONE  
ELSE'S DOCUMENTS FOLDER.



# RStudio Projects

On [RStudio.cloud](#), each project you create is in fact a completely separate instance of R.

By now, most of you should have [RStudio Desktop](#) installed.

Once that's up and running, you can get to grips with [RStudio projects](#)

Projects provide a nice way to organise your work into neat, individually tailored sets of directories.

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

Console Jobs

F:/GitHub/emosatdot/

```
R version 4.0.5 (2021-03-31) -- "Shake and Throw"
Copyright (C) 2021 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (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.

[Workspace loaded from F:/GitHub/emosatdot/.RData]

>
> |
```

Environment History Connections Tutorial

Import Dataset

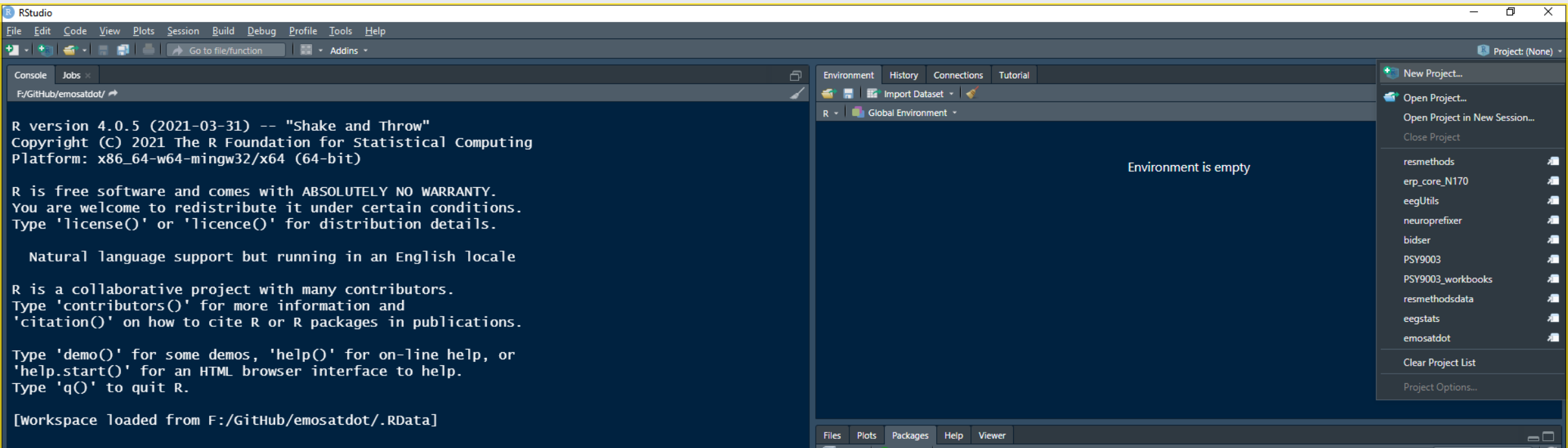
R Global Environment

Environment is empty

Files Plots Packages Help Viewer

Install Update

Name	Description	Version
<b>User Library</b>		
abind	Combine Multidimensional Arrays	1.4-5
ADG::Text	Andrew's Debugging Code Text	0.2



RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

Console Jobs

F:/GitHub/emosatdot/

R version 4.0.5 (2021-03-31) -- "Shake and Throw"  
Copyright (C) 2021 The R Foundation for Statistical Computing  
Platform: x86\_64-w64-mingw32/x64 (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.

[Workspace loaded from F:/GitHub/emosatdot/.RData]

>  
> |

Environment History Connections Tutorial




Import Dataset

R Global Environment

Environment is empty

New Project Wizard

Create Project

-  **New Directory**  
Start a project in a brand new working directory
-  **Existing Directory**  
Associate a project with an existing working directory
-  **Version Control**  
Checkout a project from a version control repository

Cancel

Description	Version
Combine Multidimensional Arrays	1.4-5
Anderson-Darling GoF test	0.3
Analysis of Factorial Experiments	0.28-1
Interpolation of Irregularly and Regularly Spaced Data	0.6-2.1
Convenience Functions for Arrays	1.1-0
Integration to 'Apache' 'Arrow'	4.0.0.1
David Scott's ASH Routines	1.0-15
'Asio' C++ Header Files	1.16.1-1

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

Console Jobs

F:/GitHub/emosatdot/

R version 4.0.5 (2021-03-31) -- "Shake and Throw"  
Copyright (C) 2021 The R Foundation for Statistical Computing  
Platform: x86\_64-w64-mingw32/x64 (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.

[Workspace loaded from F:/GitHub/emosatdot/.RData]

>  
> |

Environment History Connections Tutorial

Import Dataset

R Global Environment

Environment is empty

Project: (None)

List

New Project Wizard

Back Create New Project

Directory name:  
demo\_project

Create project as subdirectory of:  
F:/GitHub Browse...

☐ Create a git repository  
☐ Use renv with this project

☐ Open in new session

Create Project Cancel

Description Version

Combine Multidimensional Arrays	1.4-5	⊕ ●
Anderson-Darling Gof test	0.3	⊕ ●
Analysis of Factorial Experiments	0.28-1	⊕ ●
Interpolation of Irregularly and Regularly Spaced Data	0.6-2.1	⊕ ●
Convenience Functions for Arrays	1.1-0	⊕ ●
Integration to 'Apache' 'Arrow'	4.0.0.1	⊕ ●
David Scott's ASH Routines	1.0-15	⊕ ●

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

Console Jobs

F:/GitHub/emosatdot/

R version 4.0.5 (2021-03-31) -- "Shake and Throw"  
Copyright (C) 2021 The R Foundation for Statistical Computing  
Platform: x86\_64-w64-mingw32/x64 (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.

[Workspace loaded from F:/GitHub/emosatdot/.RData]

>  
> |

Environment History Connections Tutorial




Import Dataset

R Global Environment

Environment is empty

New Project Wizard

Create Project

-  **New Directory**  
Start a project in a brand new working directory
-  **Existing Directory**  
Associate a project with an existing working directory
-  **Version Control**  
Checkout a project from a version control repository

Cancel

Description	Version
Combine Multidimensional Arrays	1.4-5
Anderson-Darling GoF test	0.3
Analysis of Factorial Experiments	0.28-1
Interpolation of Irregularly and Regularly Spaced Data	0.6-2.1
Convenience Functions for Arrays	1.1-0
Integration to 'Apache' 'Arrow'	4.0.0.1
David Scott's ASH Routines	1.0-15
'Asio' C++ Header Files	1.16.1-1

# Keeping your analyses organised

Make a new RStudio project for each week's exercises!

Follow sensible structure:

Keep your data in a folder called data.

Keep your scripts or RMarkdown documents in a folder called scripts.

Give your files sensible names!

For more general workflow advice, check out What They Forgot to Teach You About R at <https://rstats.wtf/>

Relating data to structure



# Let's think about an *experiment*

The experiment is a reaction time experiment with a two-by-two repeated measures design.

Participants see pictures of objects twice. Sometimes they are seen from the *same* viewpoint twice, sometimes from *different* viewpoints each time.

There are two separate blocks of trials. The dependent variable is how long it takes them to name the objects, or *reaction time*.

You're interested in whether:

1. they get faster at naming object the second time
2. they are faster when the same view is presented both times.

# How many variables are there?

Variables	R Data Type
Participant ID	Numeric or character
Reaction times	Numeric
Block first/second	Character/factor
Viewpoint same/different	Character/factor

The final dataset needs to be able to do several things.

1. It needs to uniquely identify each participant.
2. It needs to tie each value to the right participant.
3. It needs to identify what each value represents in terms of the design.

**Tidy data**

# The three principles of tidy data

1. Each variable must have its own column.
2. Each observation must have its own row.
3. Each value must have its own cell.

country	year	cases	population
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	212258	1272015272
China	2000	216766	1280426583

variables

country	year	cases	population
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	212258	1272015272
China	2000	216766	1280426583

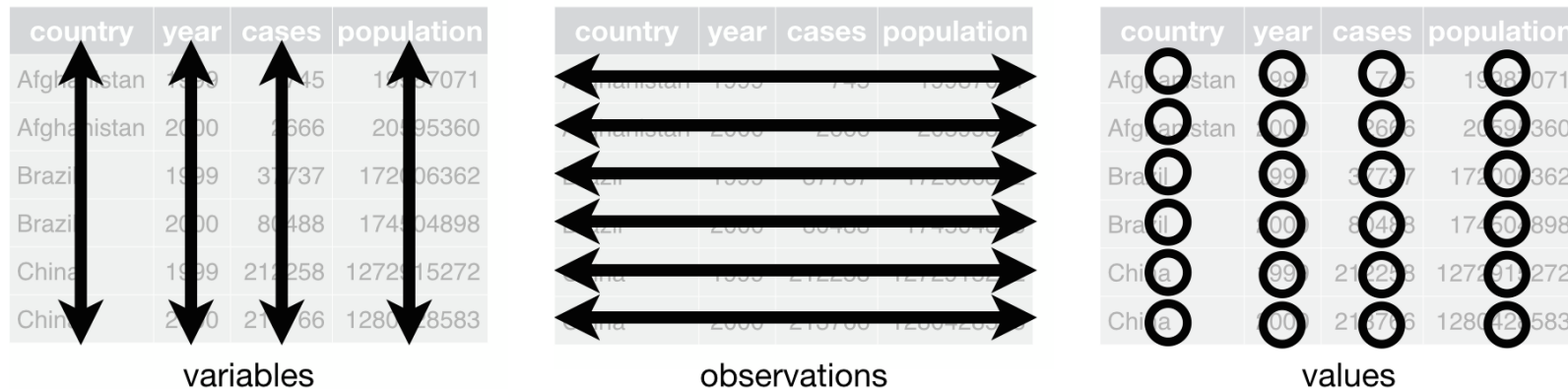
observations

country	year	cases	population
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	212258	1272015272
China	2000	216766	1280426583

values

# Why Tidy?

1. Many functions in R operate on so-called *long* format data, requiring dependent and independent variables to be in different columns of a data frame.
2. Having a consistent way to store and structure your data makes it more *generic*. This makes it easier to use it with different functions.
3. Being *generic* also makes it easier to understand a new dataset in this format.



**The many ways to structure data**

# One column for condition, one column for RT

```
## # A tibble: 40 × 3
## # Groups:   Participant [10]
##   Participant exp_condition RT
##   <int> <chr> <dbl>
## 1         1 Block1_different 407.
## 2         1 Block1_same      415.
## 3         1 Block2_different 382.
## 4         1 Block2_same      371.
## 5         2 Block1_different 420.
## 6         2 Block1_same      384.
## 7         2 Block2_different 479.
## 8         2 Block2_same      402.
## 9         3 Block1_different 368.
## 10        3 Block1_same      341.
## # ... with 30 more rows
## # i Use `print(n = ...)` to see more rows
```

This is a little awkward.

At first glance, there's no easy way to see how many variables there.

# Dependent variable split across columns

```
## # A tibble: 16 × 4
## # Groups:   Participant [8]
##   Participant Viewpoint B1RT B2RT
##   <int> <chr> <dbl> <dbl>
## 1         1 Different  536.  364.
## 2         1 Same      494.  450.
## 3         2 Different  511.  393.
## 4         2 Same      432.  371.
## 5         3 Different  536.  364.
## 6         3 Same      494.  450.
## 7         4 Different  511.  393.
## 8         4 Same      432.  371.
## 9         5 Different  536.  364.
## 10        5 Same      494.  450.
## 11        6 Different  511.  393.
## 12        6 Same      432.  371.
## 13        7 Different  536.  364.
## 14        7 Same      494.  450.
## 15        8 Different  511.  393.
## 16        8 Same      432.  371.
```

Now there's a mishmash of things:

One variable (Viewpoint) is in one column.

The Block variable is spread across two columns.

The dependent variable (reaction time) is spread across two columns.



# One column per condition

```
## # A tibble: 10 × 5
##   Participant Block1_same Block2_same Block1_different Block2_different
##   <int>         <dbl>         <dbl>         <dbl>         <dbl>
## 1           1         515.          268.          546.          413.
## 2           2         471.          249.          535.          449.
## 3           3         507.          331.          501.          386.
## 4           4         482.          312.          607.          389.
## 5           5         484.          322.          595.          431.
## 6           6         502.          301.          527.          359.
## 7           7         520.          328.          557.          398.
## 8           8         579.          272.          578.          378.
## 9           9         441.          290.          572.          401.
## 10          10         526.          285.          550.          405.
```

This is also called **wide** format.

# How many *variables* are there?

```
## # A tibble: 10 × 5
##   Participant Block1_same Block2_same Block1_different Block2_different
##   <int>         <dbl>         <dbl>         <dbl>         <dbl>
## 1           1         515.          268.          546.          413.
## 2           2         471.          249.          535.          449.
## 3           3         507.          331.          501.          386.
## 4           4         482.          312.          607.          389.
## 5           5         484.          322.          595.          431.
## 6           6         502.          301.          527.          359.
## 7           7         520.          328.          557.          398.
## 8           8         579.          272.          578.          378.
## 9           9         441.          290.          572.          401.
## 10          10         526.          285.          550.          405.
```

Four... but there are five columns.

```
ncol(example_rt_df)
```

```
## [1] 5
```

# How many *observations* are there?

```
## # A tibble: 10 × 5
##   Participant Block1_same Block2_same Block1_different Block2_different
##   <int>         <dbl>         <dbl>         <dbl>         <dbl>
## 1           1         515.         268.         546.         413.
## 2           2         471.         249.         535.         449.
## 3           3         507.         331.         501.         386.
## 4           4         482.         312.         607.         389.
## 5           5         484.         322.         595.         431.
## 6           6         502.         301.         527.         359.
## 7           7         520.         328.         557.         398.
## 8           8         579.         272.         578.         378.
## 9           9         441.         290.         572.         401.
## 10          10         526.         285.         550.         405.
```

40... but there are 10 rows.

```
nrow(example_rt_df)
```

```
## [1] 10
```

# One column per condition

```
## # A tibble: 10 × 5
##   Participant Block1_same Block2_same Block1_different Block2_different
##   <int>         <dbl>         <dbl>         <dbl>         <dbl>
## 1           1         515.          268.          546.          413.
## 2           2         471.          249.          535.          449.
## 3           3         507.          331.          501.          386.
## 4           4         482.          312.          607.          389.
## 5           5         484.          322.          595.          431.
## 6           6         502.          301.          527.          359.
## 7           7         520.          328.          557.          398.
## 8           8         579.          272.          578.          378.
## 9           9         441.          290.          572.          401.
## 10          10         526.          285.          550.          405.
```

This is also called **wide** format.

# This data is *untidy*

One variable - RT - is split across four columns.

Another variable - Block - is split across two columns.

A third variable - viewpoint - is also split across two columns.

country	1999	2000
Afghanistan	745	2666
Brazil	37737	80488
China	212258	213766

table4

country	1999	2000
Afghanistan	19987071	20595360
Brazil	172006362	174504898
China	1272915272	1280428583

table5

country	1999	2000
Afghanistan	745	2666
Brazil	37737	80488
China	212258	213766

variables

country	1999	2000
Afghanistan	745	2666
Brazil	37737	80488
China	212258	213766

observations

# Tidying your data

# Tidyr

The `tidyr` package contains functions to help tidy up your data.

We'll look now at `pivot_longer()` and `pivot_wider()`.

To start tidying our data, we need the RTs to be in a single column.

```
head(example_rt_df, n = 4)
```

```
## # A tibble: 4 × 5
##   Participant Block1_same Block2_same Block1_different Block2_different
##       <int>      <dbl>      <dbl>          <dbl>          <dbl>
## 1         1         508.        340.          522.          295.
## 2         2         523.        268.          550.          470.
## 3         3         543.        303.          667.          476.
## 4         4         556.        408.          400.          322.
```

The function `pivot_longer()` can be used to combine columns into one.

Look at the help using `?pivot_longer`

# Pivoting longer

```
pivot_longer(data,  
             cols,  
             names_to = "key",  
             values_to = "value",  
             ...)
```

The first argument, `data`, is the name of the data frame you want to modify.

`cols` are the columns you want to combine together.

`names_to` is the name of the new column that will contain the values of a single categorical variable.

`values_to` is the name of the new column containing the values for each level of that variable.



# Pivoting longer

```
long_rt <-  
  pivot_longer(example_rt_df,  
                cols = c("Block1_same",  
                          "Block1_different",  
                          "Block2_same",  
                          "Block2_different"),  
                names_to = "exp_cond",  
                values_to = "RT")  
head(long_rt)
```

```
## # A tibble: 6 × 3  
##   Participant exp_cond      RT  
##       <int> <chr>      <dbl>  
## 1         1 Block1_same    508.  
## 2         1 Block1_different 522.  
## 3         1 Block2_same    340.  
## 4         1 Block2_different 295.  
## 5         2 Block1_same    523.  
## 6         2 Block1_different 550.
```

# Pivoting longer

After we specify the "key" and "value" columns, we need to specify which columns we want to be *gathered*.

```
long_rt <-  
  pivot_longer(example_rt_df,  
                names_to = "exp_cond",  
                values_to = "RT",  
                cols = 2:5) # here I use num  
head(long_rt)
```

```
## # A tibble: 6 × 3  
##   Participant exp_cond      RT  
##       <int> <chr>      <dbl>  
## 1           1 Block1_same  508.  
## 2           1 Block2_same  340.  
## 3           1 Block1_different 522.  
## 4           1 Block2_different 295.  
## 5           2 Block1_same  523.  
## 6           2 Block2_same  268.
```

```
long_rt <-  
  pivot_longer(example_rt_df,  
                names_to = "exp_cond",  
                values_to = "RT",  
                cols = Block1_same:Block2_di  
head(long_rt)
```

```
## # A tibble: 6 × 3  
##   Participant exp_cond      RT  
##       <int> <chr>      <dbl>  
## 1           1 Block1_same  508.  
## 2           1 Block2_same  340.  
## 3           1 Block1_different 522.  
## 4           1 Block2_different 295.  
## 5           2 Block1_same  523.  
## 6           2 Block2_same  268.
```

# Splitting columns

We have the RTs in one column, but we still have another problem:

The "Block" and "Viewpoint" variables are combined into a single column.

```
head(long_rt)
```

```
## # A tibble: 6 × 3
##   Participant exp_cond      RT
##   <int> <chr>      <dbl>
## 1         1 Block1_same  508.
## 2         1 Block2_same  340.
## 3         1 Block1_different 522.
## 4         1 Block2_different 295.
## 5         2 Block1_same  523.
## 6         2 Block2_same  268.
```

# Splitting columns

Fortunately, the values in the *exp\_cond* column can be easily split:

```
unique(long_rt$exp_cond)
```

```
## [1] "Block1_same"      "Block2_same"      "Block1_different" "Block2_different"
```

The value of "Block" comes before the underscore ("\_"), while the value of "viewpoint" comes after it.

# Splitting columns

```
long_rt <-  
  pivot_longer(example_rt_df,  
                names_to = c("Block",  
                             "Viewpoint"),  
                names_sep = "_",  
                values_to = "RT",  
                cols = Block1_same:Block2_different)  
head(long_rt)
```

```
## # A tibble: 6 × 4  
##   Participant Block Viewpoint    RT  
##       <int> <chr>   <chr>   <dbl>  
## 1           1 Block1 same     508.  
## 2           1 Block2 same     340.  
## 3           1 Block1 different 522.  
## 4           1 Block2 different 295.  
## 5           2 Block1 same     523.  
## 6           2 Block2 same     268.
```

# Splitting columns

Let's look at the additional syntax.

```
pivot_longer(example_rt_df,  
              names_to = c("Block",  
                           "Viewpoint"),  
              names_sep = "_",  
              values_to = "RT",  
              cols = Block1_same:Block2_different)
```

`names_to` now has two entries, one for each new column that will be made.

`names_sep` is the character that *separates* the values you want to split.

# Your target

```
## # A tibble: 15 × 4
##   Participant Block Viewpoint    RT
##       <int> <chr>   <chr>   <dbl>
## 1           1 Block1  same    508.
## 2           1 Block2  same    340.
## 3           1 Block1  different 522.
## 4           1 Block2  different 295.
## 5           2 Block1  same    523.
## 6           2 Block2  same    268.
## 7           2 Block1  different 550.
## 8           2 Block2  different 470.
## 9           3 Block1  same    543.
## 10          3 Block2  same    303.
## 11          3 Block1  different 667.
## 12          3 Block2  different 476.
## 13          4 Block1  same    556.
## 14          4 Block2  same    408.
## 15          4 Block1  different 400.
```

You should specify name(s) for the column(s) that you'll create using the `names_to` and `values_to` arguments.

You'll need to add `names_sep` and the character that separates the two sides as well in order to match the target

Pivoting wider



# Pivoting wider

Sometimes you want to go in the *opposite* direction.

`pivot_wider()` is the *opposite* of `pivot_longer()`.

```
wide_rt <-  
  pivot_wider(long_rt,  
              names_from = c("Block",  
                             "Viewpoint"),  
              values_from = "RT")  
head(wide_rt, 10)
```

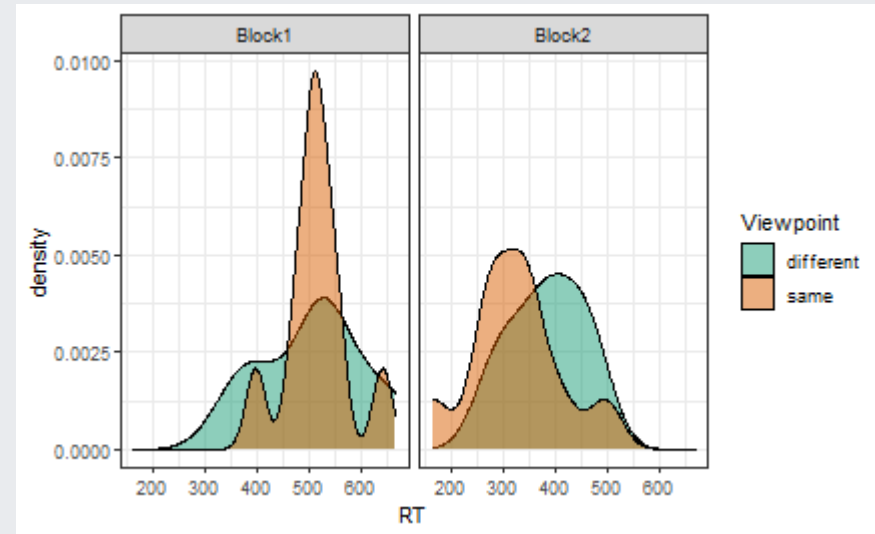
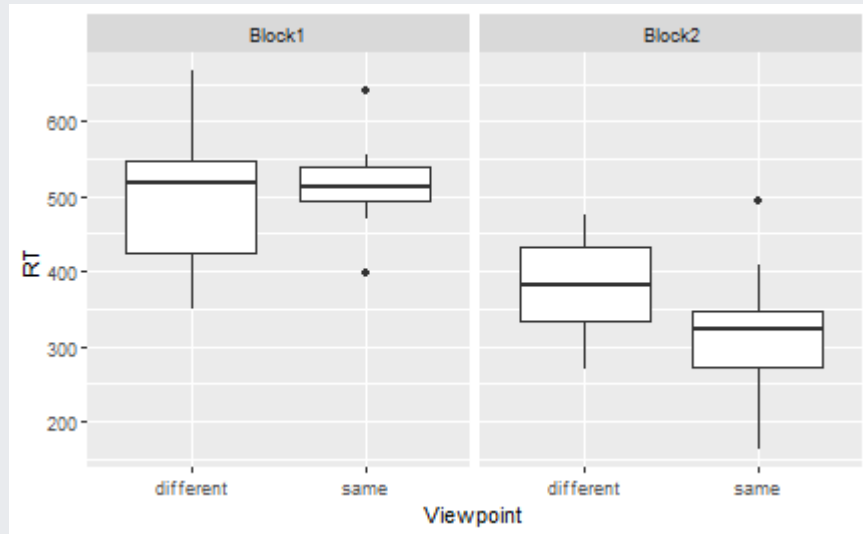
```
## # A tibble: 10 × 5  
##   Participant Block1_same Block2_same Block1_different Block2_different  
##           <int>      <dbl>      <dbl>          <dbl>          <dbl>  
## 1             1        508.        340.          522.          295.  
## 2             2        523.        268.          550.          470.  
## 3             3        543.        303.          667.          476.  
## 4             4        556.        408.          400.          322.  
## 5             5        506.        163.          539.          269.  
## 6             6        489.        287.          350.          363.  
## 7             7        398.        346.          624.          392.  
## 8             8        470.        494.          504.          374.  
## 9             9        517.        258.          396.          422.
```

**Now what?**

# Now that it's tidy...

Now that we've got the data in a tidy format, we can begin to use some of the more interesting features of R!

We can produce a boxplot using **ggplot2** (more next week!)



# Now that it's tidy...

We can produce some summary statistics using **dplyr** (more soon!)

```
## `summarise()` has grouped output by 'Block'. You can override using the  
## `.groups` argument.
```

```
## # A tibble: 4 × 4  
## # Groups:   Block [2]  
##   Block Viewpoint mean_RT sd_RT  
##   <chr>   <chr>      <dbl> <dbl>  
## 1 Block1 different    507.  100.  
## 2 Block1 same        515.   62.5  
## 3 Block2 different    382.   71.2  
## 4 Block2 same        321.   89.7
```

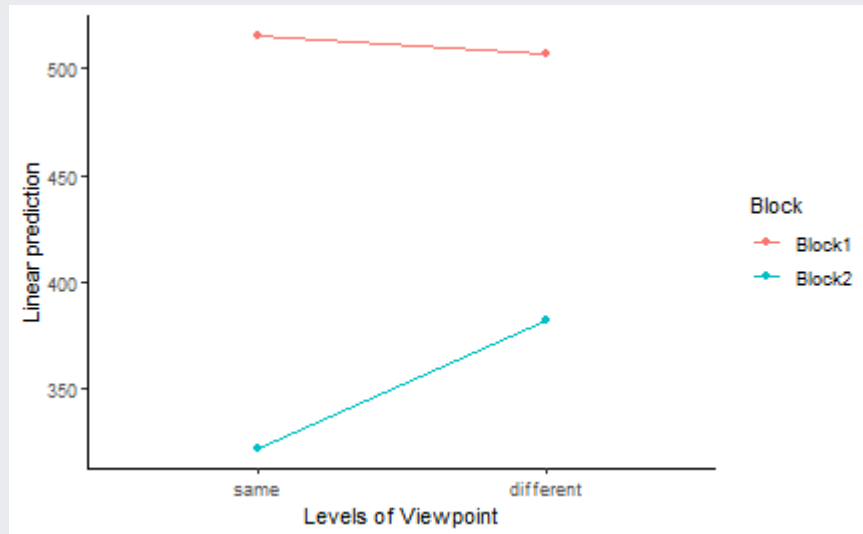
# Now that it's tidy...

We can run ANOVA with **afex**.

```
## Anova Table (Type 3 tests)
##
## Response: RT
##           Effect    df      MSE      F    ges p.value
## 1           Block 1, 9 5222.72 48.56 *** .510   <.001
## 2      Viewpoint 1, 9 7794.41   0.87 .027   .376
## 3 Block:Viewpoint 1, 9 6343.29   1.87 .046   .205
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '+' 0.1 ' ' 1
```

# Now that it's tidy...

We can create interaction plots using **emmeans**.



```
## Block = Block1:
## contrast      estimate    SE df t.ratio p.val
## same - different      8.43 39.9  9   0.212  0.83
##
## Block = Block2:
## contrast      estimate    SE df t.ratio p.val
## same - different    -60.45 35.2  9  -1.717  0.12
```

# Next week

- The following chapters of R for Data Science -
  - **Data Visualization** (Chapter 1 via the library)
  - **Graphics for communication with ggplot2** (Chapter 22 via the library)

Practice some of the skills for next week:

- **RStudio.cloud Primer**
  - Visualize Data

# A possible solution for the extra exercise!

```
set.seed(200) # if you want these exact numbers, use this line
example_rt_df <-
  tibble(Participant = seq(1, 10),
         Block1_same = rnorm(10, 500, 100),
         Block2_same = rnorm(10, 350, 100),
         Block1_different = rnorm(10, 500, 100),
         Block2_different = rnorm(10, 400, 100))
```

```
## # A tibble: 5 × 5
##   Participant Block1_same Block2_same Block1_different Block2_different
##       <int>       <dbl>       <dbl>          <dbl>          <dbl>
## 1         1         508.         340.          522.          295.
## 2         2         523.         268.          550.          470.
## 3         3         543.         303.          667.          476.
## 4         4         556.         408.          400.          322.
## 5         5         506.         163.          539.          269.
```