

# The structure of data

## Research Methods and Skills

15/10/2024

# 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!)

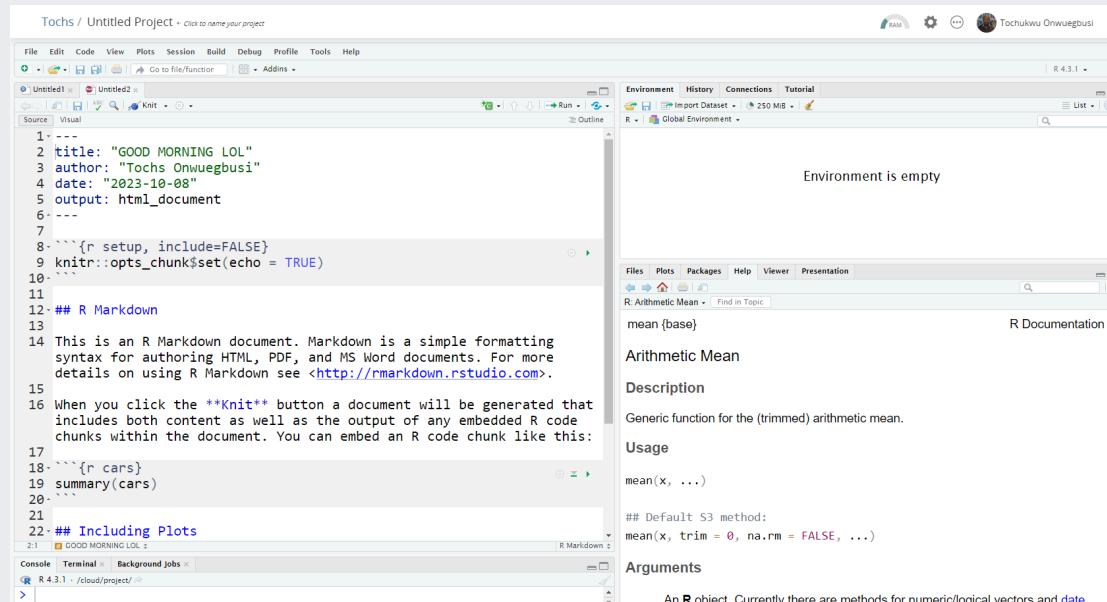
# Now load any data you want to work on. (again, we'll cover this later!)
test_data <-
  read_csv("data/a-random-RT-file.csv") %>% # I'll explain what %>% means later
  rename(RT = `reaction times`)

# 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.



The screenshot shows the RStudio interface with an R Markdown file open. The code editor contains the following R Markdown code:

```
1 ---  
2 title: "GOOD MORNING LOL"  
3 author: "Tochs Onwuegbusi"  
4 date: "2023-10-08"  
5 output: html_document  
6 ---  
7  
8 ```{r setup, include=FALSE}  
9 knitr::opts_chunk$set(echo = TRUE)  
10```  
11  
12 ## R Markdown  
13  
14 This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.  
15  
16 When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:  
17  
18 ```{r cars}  
19 summary(cars)  
20```  
21  
22 ## Including Plots
```

The RStudio interface includes the following panels:

- Environment:** Shows "Environment is empty".
- Help:** Shows the documentation for the `mean` function, including its description as "Generic function for the (trimmed) arithmetic mean" and usage information.
- Console:** Shows the message "GOOD MORNING LOL".

# 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,] -0.3575405 -1.5433706 -1.272520
## [2,]  0.3369514 -0.2820301 -0.460967
## [3,] -0.4763274 -0.2685244 -2.327759
```

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

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

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

```
##           [,1]           [,2]
## [1,]  0.3369514 -0.2820301
## [2,] -0.4763274 -0.2685244
```



tidyverse

[www.rstudio.com](http://www.rstudio.com)

# 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 -1.18   Old  
## 2          2 -0.580  Old  
## 3          3 -0.659  Old  
## 4          4  0.354  Old  
## 5          5  0.200  Old  
## 6          6  0.0819 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)

# Importing data into R

# Comma-separated values

# Excel spreadsheets

# 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 Response¹ 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>,
```

```
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.
```

```
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.
```

&gt; |

The screenshot shows the RStudio interface. The top bar includes tabs for Environment, History, Connections, and Tutorial, with the Environment tab selected. A red box highlights the 'Import Dataset' button. Below the tabs, the Global Environment pane displays the message 'Environment is empty'. In the bottom right corner, there is a file browser window titled 'Files' showing the contents of a folder named 'F:/Week 4 Teaching'. The browser includes buttons for New Folder, New Blank File, Delete, Rename, and More. The folder contains four items: '.Rhistory' (1.1 KB, modified Oct 14, 2023, 10:09 PM), 'Data' (empty folder), 'Script' (empty folder), and 'Week 4 Teaching.Rproj' (218 B, modified Oct 14, 2023, 10:09 PM).

Name	Size	Modified
..		
.Rhistory	1.1 KB	Oct 14, 2023, 10:09 PM
Data		
Script		
Week 4 Teaching.Rproj	218 B	Oct 14, 2023, 10:09 PM

Week 4 Teaching - RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

Console Terminal x Background Jobs x Environment History Connections Tutorial

R 4.0.5 Import Text Data

File/URL:  Browse...

Data Preview:

```
R is f
You ar
Type '
R is a
Type '
'citat
Type '
'help.
Type '
> |
```

0:09 PM

0:09 PM

Import Options:

Name: <input type="text" value="dataset"/>	<input checked="" type="checkbox"/> First Row as Names	Delimiter: <input type="button" value="Comma"/>	Escape: <input type="button" value="None"/>
Skip: <input type="text" value="0"/>	<input checked="" type="checkbox"/> Trim Spaces	Quotes: <input type="button" value="Default"/>	Comment: <input type="button" value="Default"/>
	<input checked="" type="checkbox"/> Open Data Viewer	Locale: <input type="button" value="Configure..."/>	NA: <input type="button" value="Default"/>

Code Preview:

```
library(readr)
dataset <- read_csv(NULL)
View(dataset)
```

② Reading rectangular data using readr

Import Cancel

This screenshot shows the 'Import Text Data' dialog in RStudio. The 'File/URL:' field is empty. The 'Data Preview:' section displays the R command history from the console. The 'Import Options:' section contains settings for reading CSV files, including 'Name' set to 'dataset', 'Skip' set to 0, and 'Delimiter' set to 'Comma'. The 'Code Preview:' section shows the R code generated for reading the dataset: `library(readr)` followed by `dataset <- read\_csv(NULL)` and `View(dataset)`. The bottom status bar indicates 'Reading rectangular data using readr'.

Week 4 Teaching - RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

Console Terminal x Background Jobs x Environment History Connections Tutorial

R 4.0.5 Import Text Data

R version 4.0.5 (2020-09-22) -- "The Zen of Simplicity"

Copyright (C) 2020 The R Foundation for Statistical Computing

Platform: x86\_64-w64-mingw32

R is free software and comes with ABSOLUTELY NO WARRANTY.

You are welcome to redistribute it under certain conditions.

Type 'help()' or ' '? for help, and 'q()' to quit R.

R is a language and environment for statistical computing and graphics.

Type 'help()' or ' ?' for help, and 'q()' to quit R.

Type 'help()' or ' ?' for help, and 'q()' to quit R.

> |

Choose File

Organise New folder

Name Date modified Type Size

Weekly Exercise

Training 2021 17/06/2021 12:02 File folder

Tutor 30/01/2019 14:55 File folder

Use 21/11/2022 20:22 File folder

use and delete 05/04/2018 11:32 File folder

Violence Reduction Partnership 22/09/2023 18:47 File folder

VIP 01/09/2022 13:07 File folder

VODAFONE - S23 Uktra 19/07/2023 17:26 File folder

Voluntary Career Speaker 17/06/2015 00:06 File folder

Week 4 Demo 25/10/2022 09:37 File folder

Week 4 Teaching 25/10/2022 11:31 File folder

File name: All Files (\*.\*) Open Cancel

Import Options:

Name: dataset 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)
dataset <- read_csv(NULL)
View(dataset)
```

Reading rectangular data using read

Import Cancel

0:09 PM 0:09 PM

Week 4 Teaching - RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins List Week 4 Teaching F/

Console Terminal × Background Jobs × Environment History Connections Tutorial

R 4.0.5 Import Text Data

R version 4.0.5 (2021-06-25) -- "Rhubarb"

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 'help()' or '?("<topic>")' for help on a specific topic.

R is a Type 'citat' platform.

Type 'help()' or '?("<topic>")' for help on a specific topic.

> | 0:09 PM 0:09 PM

Data Preview:

ResponseID (character)	ResponseSet (character)	Name (character)	ExternalDataReference (logical)	Status (double)	StartDate (character)	EndDate (character)	Finished (double)	Consent Form / This study includes a range of questionnaires collecting / demographic and individual... (double)	sex (double)	age (double)	hexacol (double)	hexaco2 (double)	hexaco3 (double)
R_aI4tgG1GHNdVdqt	Default Response Set	Anonymous	NA		0 20/10/2014 12:10	20/10/2014 12:19	1		1	2	26	4	5
R_d5OjATV0lJIBbMx	Default Response Set	Anonymous	NA		0 20/10/2014 12:15	20/10/2014 12:27	1		1	2	66	2	4
R_aaBV2UE9mlGIDpH	Default Response Set	Anonymous	NA		0 20/10/2014 12:18	20/10/2014 12:28	1		1	1	41	1	5
R_6nxlnLKQv2bucQZ	Default Response Set	Anonymous	NA		0 20/10/2014 12:18	20/10/2014 12:29	1		1	1	46	1	5
R_6SCYbhOP9BG5CgR	Default Response Set	Anonymous	NA		0 20/10/2014 12:24	20/10/2014 12:32	1		1	2	53	2	4
R_5pCxWA6qOQdnVyd	Default Response Set	Anonymous	NA		0 20/10/2014 12:34	20/10/2014 12:43	1		1	1	33	2	4
R_dInj6V7FcPpn0x	Default Response Set	Anonymous	NA		0 20/10/2014 12:02	20/10/2014 12:54	1		1	1	39	2	4
R_9v6ZgUhKfVX6yp	Default Response Set	Anonymous	NA		0 20/10/2014 12:33	20/10/2014 13:01	1		1	2	38	1	2
R_5Bg7VjBhOZja2gt	Default Response Set	Anonymous	NA		0 20/10/2014 13:17	20/10/2014 13:25	1		1	2	19	4	4
R_95v17lQGWXwo23z	Default Response Set	Anonymous	NA		0 20/10/2014 13:00	20/10/2014 13:25	1		1	2	19	2	2
R_eL1py2Fms0XNOuh	Default Response Set	Anonymous	NA		0 20/10/2014 13:22	20/10/2014 13:32	1		1	1	49	1	2
R_8i66z9VnZcuxKyV	Default Response Set	Anonymous	NA		0 20/10/2014 13:24	20/10/2014 13:54	1		1	1	37	2	4
R_9GKITTj2nQa1X6d	Default Response Set	Anonymous	NA		0 20/10/2014 13:47	20/10/2014 14:00	1		1	2	24	1	2
R_1B1yulegO16INJX	Default Response Set	Anonymous	NA		0 20/10/2014 14:30	20/10/2014 14:47	1		1	2	43	2	4
R_eaCrcTcmZBUWsqV	Default Response Set	Anonymous	NA		0 20/10/2014 14:53	20/10/2014 15:03	1		1	2	19	2	5
R_9BI9eK3TW71jgXz	Default Response Set	Anonymous	NA		0 20/10/2014 15:00	20/10/2014 15:00	-		-	-	--	-	-

Previewing first 50 entries. 7 parsing errors.

Import Options:

Name: FearofCrime	<input checked="" type="checkbox"/> First Row as Names	Delimiter: Comma	Escape: None
Skip: 0	<input checked="" type="checkbox"/> Trim Spaces	Quotes: Default	Comment: Default
<input checked="" type="checkbox"/> Open Data Viewer	Locale: Configure...	NA: Default	

Code Preview:

```
Tibble (readr)
FearofCrime <- read_csv("Data/Fearofcrime.csv")
View(FearofCrime)
```

② Reading rectangular data using readr Import Cancel

Week 4 Teaching - RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins List Week 4 Teaching F/

Console Terminal × Background Jobs × Environment History Connections Tutorial

R 4.0.5 Import Text Data

R version 4.0.5 (2021-06-25) -- "Rhubarb"

Platform: x86\_64-w64-mingw32

Data Preview:

ResponseID (character)	ResponseSet (character)	Name (character)	ExternalDataReference (logical)	Status (double)	StartDate (character)	EndDate (character)	Finished (double)	Consent Form / This study includes a range of questionnaires collecting / demographic and individual... (double)	sex (double)	age (double)	hexacol (double)	hexaco2 (double)	hexaco3 (double)
R_aI4tgG1GHNdVdqt	Default Response Set	Anonymous	NA		0 20/10/2014 12:04 21:26	0 20/10/2014 12:14 21:26	1	1	2	26	4	5	
R_d5OjATV0lJIBbMx	Default Response Set	Anonymous	NA		0 20/10/2014 12:10	20/10/2014 12:19	1	1	2	66	2	4	
R_aaBV2UE9mlGIDpH	Default Response Set	Anonymous	NA		0 20/10/2014 12:15	20/10/2014 12:27	1	1	1	41	1	5	
R_6nxlnLKQv2bucQZ	Default Response Set	Anonymous	NA		0 20/10/2014 12:18	20/10/2014 12:28	1	1	1	46	1	5	
R_6SCYbhOP9BG5CgR	Default Response Set	Anonymous	NA		0 20/10/2014 12:18	20/10/2014 12:29	1	1	2	53	2	4	
R_5pCxWA6qOQdnVyd	Default Response Set	Anonymous	NA		0 20/10/2014 12:24	20/10/2014 12:32	1	1	1	33	2	4	
R_dInjI6V7FcPpn0x	Default Response Set	Anonymous	NA		0 20/10/2014 12:34	20/10/2014 12:43	1	1	2	41	1	5	
R_9v6ZgUhKfVX6yp	Default Response Set	Anonymous	NA		0 20/10/2014 12:02	20/10/2014 12:54	1	1	1	39	2	4	
R_5Bg7VjBhOZja2gt	Default Response Set	Anonymous	NA		0 20/10/2014 12:33	20/10/2014 13:01	1	1	2	38	1	2	
R_9Sv17IQGWXwo23z	Default Response Set	Anonymous	NA		0 20/10/2014 13:17	20/10/2014 13:25	1	1	2	19	4	4	
R_eL1py2Fms0XNOuh	Default Response Set	Anonymous	NA		0 20/10/2014 13:00	20/10/2014 13:25	1	1	2	19	2	2	
R_8i66z9VnZcuxKyV	Default Response Set	Anonymous	NA		0 20/10/2014 13:22	20/10/2014 13:32	1	1	1	49	1	2	
R_9GKITTj2nQa1X6d	Default Response Set	Anonymous	NA		0 20/10/2014 13:24	20/10/2014 13:54	1	1	1	37	2	4	
R_1B1yulegO16INJX	Default Response Set	Anonymous	NA		0 20/10/2014 13:47	20/10/2014 14:00	1	1	2	24	1	2	
R_eaCrcTcmZBUWsqV	Default Response Set	Anonymous	NA		0 20/10/2014 14:30	20/10/2014 14:47	1	1	2	43	2	4	
R_9BI9eK3TW71jgXz	Default Response Set	Anonymous	NA		0 20/10/2014 14:53	20/10/2014 15:03	1	1	2	19	2	5	
...	...	...	...	...	...	...	...	...	...	...	...	...	...

Previewing first 50 entries. 7 parsing errors.

Import Options:

Name: FearofCrime	<input checked="" type="checkbox"/> First Row as Names	Delimiter: Comma	Escape: None
Skip: 0	<input checked="" type="checkbox"/> Trim Spaces	Quotes: Default	Comment: Default
<input checked="" type="checkbox"/> Open Data Viewer	Locale: Configure...	NA: Default	

Code Preview:

```
library(readr)
FearofCrime <- read_csv("Data/Fearofcrime.csv")
View(FearofCrime)
```

② Reading rectangular data using readr Import Cancel

	ResponseID	ResponseSet	Name	ExternalDataReference	Status	StartDate	EndDate	Finished	Consent Form / This study includes a range of questionnaires collecting / demographic and individual...
1	R_ai4tgG1GHNdVdq	Default Response Set	Anonymous	NA	0	19/10/2014 21:08	19/10/2014 21:26	1	
2	R_d5OjATV0JiBbMx	Default Response Set	Anonymous	NA	0	20/10/2014 12:10	20/10/2014 12:19	1	
3	R_aaBVZUe9mlGIDpH	Default Response Set	Anonymous	NA	0	20/10/2014 12:15	20/10/2014 12:27	1	
4	R_6nxlnLKQv2bucQZ	Default Response Set	Anonymous	NA	0	20/10/2014 12:18	20/10/2014 12:28	1	
5	R_6SCYbOP9BG5CgR	Default Response Set	Anonymous	NA	0	20/10/2014 12:18	20/10/2014 12:29	1	
6	R_5pCxWA6qOQdnVyd	Default Response Set	Anonymous	NA	0	20/10/2014 12:24	20/10/2014 12:32	1	
7	R_d1nji6V7FCppn0x	Default Response Set	Anonymous	NA	0	20/10/2014 12:34	20/10/2014 12:43	1	
8	R_9v6ZgUhKflVX6yp	Default Response Set	Anonymous	NA	0	20/10/2014 12:02	20/10/2014 12:54	1	
9	R_5Bg7VjBhOZja2gt	Default Response Set	Anonymous	NA	0	20/10/2014 12:33	20/10/2014 13:01	1	
10	R_9Sv17IQGWXwo23z	Default Response Set	Anonymous	NA	0	20/10/2014 13:17	20/10/2014 13:25	1	
11	R_eLipy2Fms0XNOuh	Default Response Set	Anonymous	NA	0	20/10/2014 13:00	20/10/2014 13:25	1	
12	R_8i66z9VnZcuxKyV	Default Response Set	Anonymous	NA	0	20/10/2014 13:22	20/10/2014 13:32	1	
13	R_9GKTTj2nQa1X6d	Default Response Set	Anonymous	NA	0	20/10/2014 13:24	20/10/2014 13:54	1	
14	R_1B1yulegO16INJX	Default Response Set	Anonymous	NA	0	20/10/2014 13:47	20/10/2014 14:00	1	
15	R_1B1yulegO16INJX	Default Response Set	Anonymous	NA	0	20/10/2014 14:20	20/10/2014 14:47	1	

Showing 1 to 15 of 301 entries, 169 total columns

Console Terminal × Background Jobs ×

```
R 4.0.5 - F:/Week 4 Teaching/ 
> library(readr)
> FearofCrime <- read_csv("Data/Fearofcrime.csv")
Rows: 301 Columns: 169
-- Column specification --
Delimiter: ","
chr (5): ResponseID, ResponseSet, Name, StartDate, EndDate
dbl (163): status, Finished, Consent Form / This study includes a range...
lg1 (1): ExternalDataReference

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
> View(FearofCrime)
```

Environment History Connections Tutorial

Import Dataset 114 MB

R Global Environment

Data

F FearofCrime 301 obs. of 169 variables

Files Plots Packages Help Viewer Presentation

New Folder New Blank File Delete Rename More

F Week 4 Teaching

Name Size Modified

- .Rhistory 1.1 KB Oct 14, 2023, 10:09 PM
- Data
- Script
- Week 4 Teaching.Rproj 218 B Oct 14, 2023, 10:09 PM

# 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....



PROTIP: 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

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

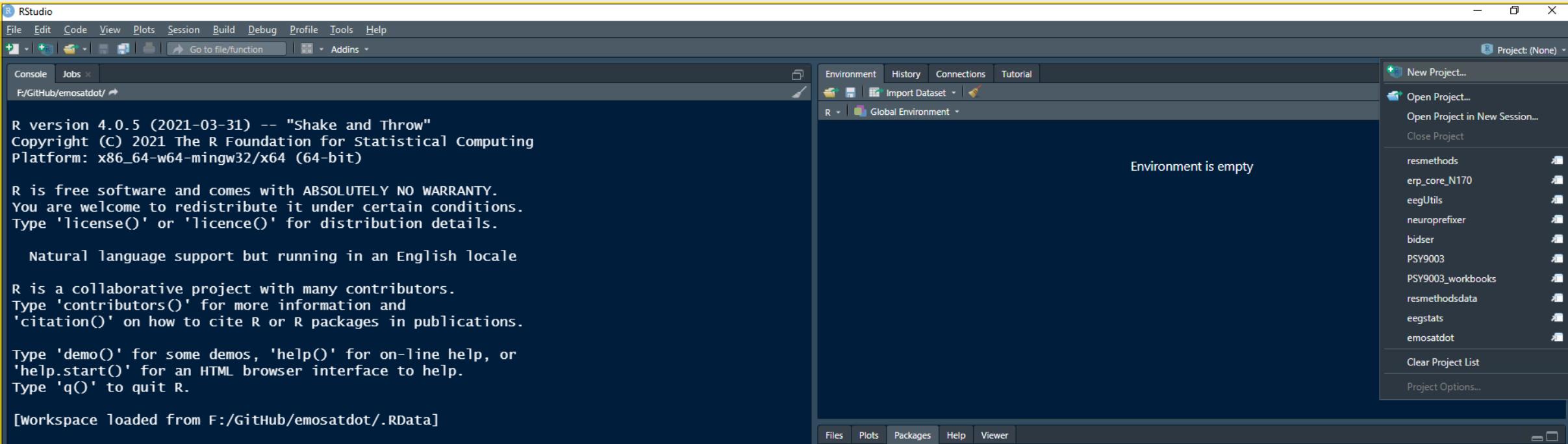
R Global Environment ↵

Environment is empty

Files Plots Packages Help Viewer

Install Update

Name	Description	Version
abind	Combine Multidimensional Arrays	1.4-5
ABC-Test	Adaptive Bayesian C-Estimator	0.3



RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Console Jobs F:/GitHub/emosatdot/ Addins Project: (None)

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]

>

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

ash

AsioHeaders

Environment History Connections Tutorial Import Dataset Global Environment

Environment is empty

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 Global Environment

Environment is empty

New Project Wizard Create New Project

Back Directory name: demo\_project

Create project as subdirectory of: F:/GitHub

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

Console Jobs F:/GitHub/emosatdot/ Addins Project: (None) ▾

Go to file/function

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

ash AsioHeaders

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

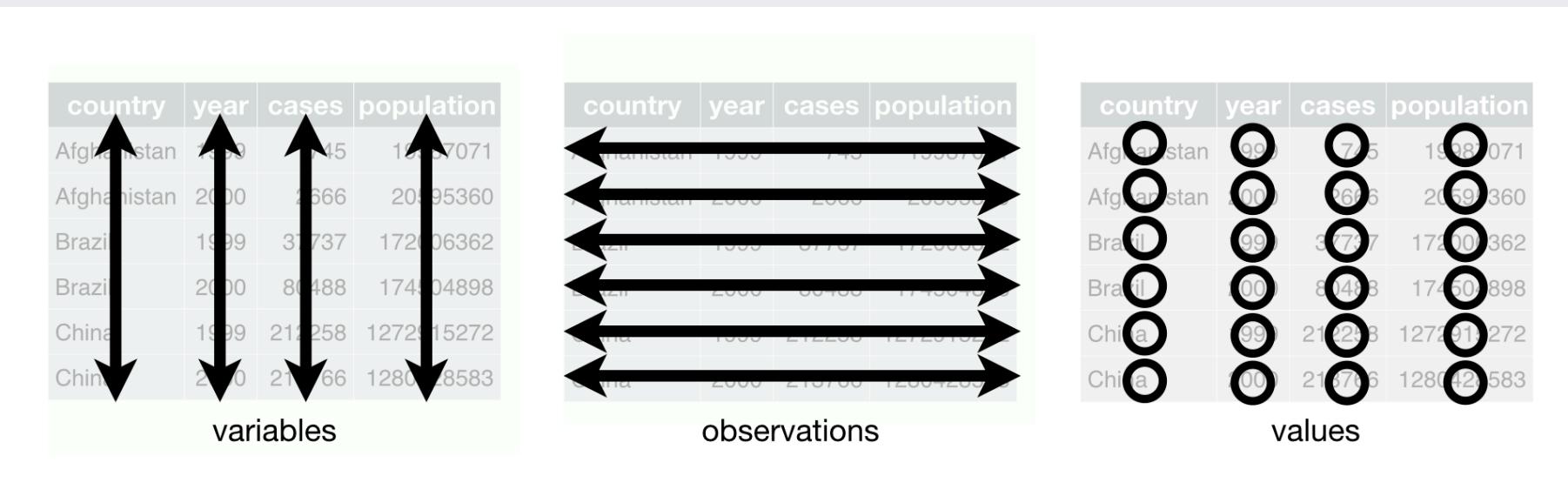
The final dataset needs to be able to do several things.

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

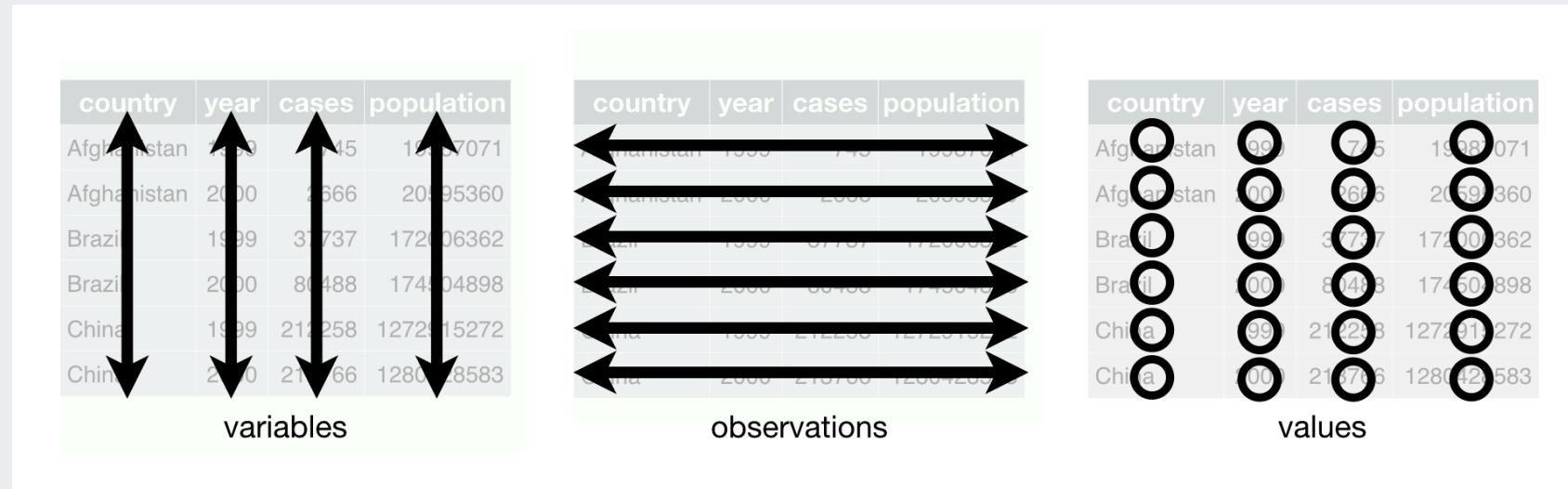
# The three principles of tidy data

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



# 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
```

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	745	2666
Brazil	37737	80488
China	212258	213766

variables

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

observations

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

table5

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

variables

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

observations

# Tidying your data

# Tidyr

The `tidyverse` 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)
```

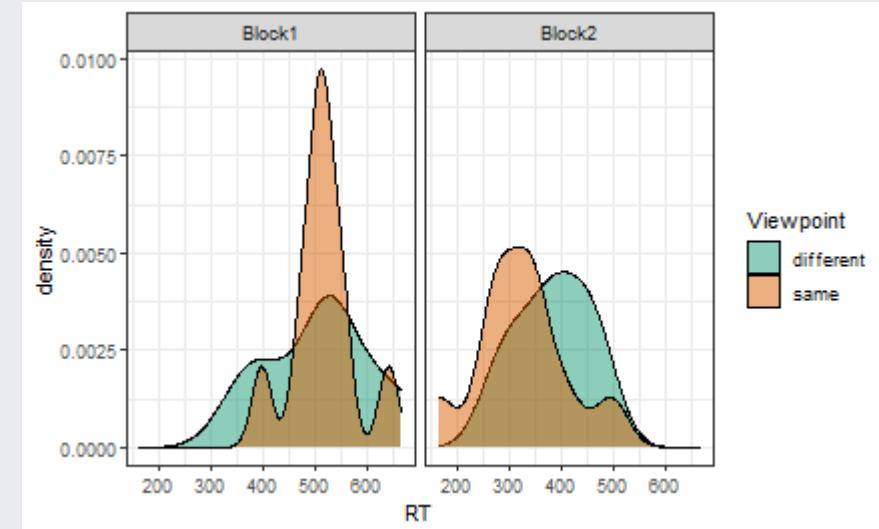
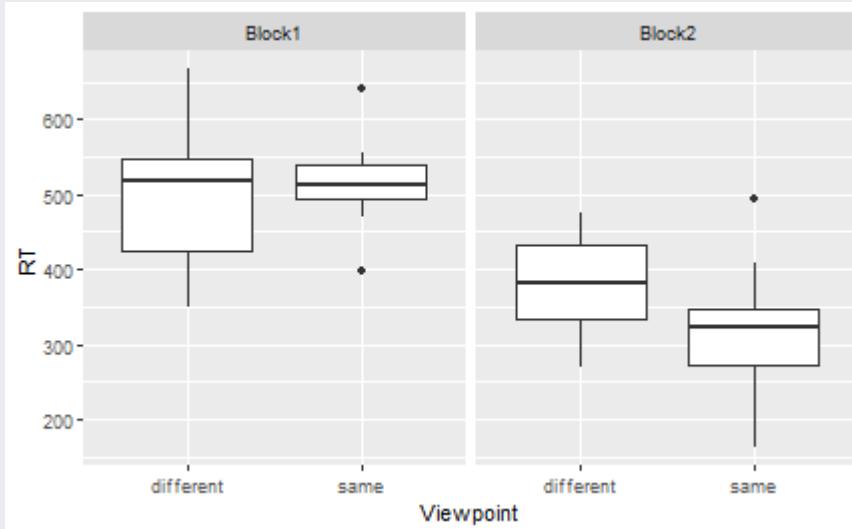
```
## # A tibble: 10 × 5  
##   Participant Block1_same Block2_same Block1_different Block2_different  
##       <int>      <dbl>      <dbl>          <dbl>          <dbl>  
## 1           1      508.     340.          522.         295.  
## 2           2      523.     268.          550.         470.  
## 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
```

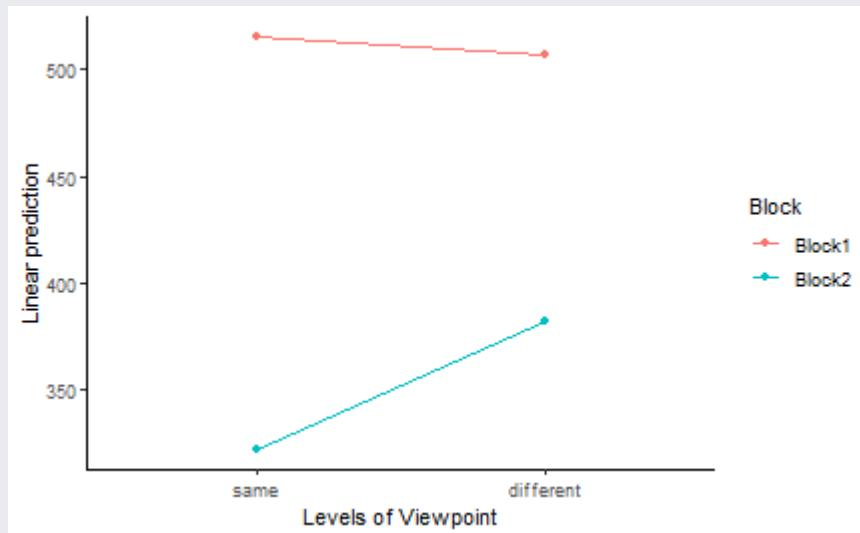
# Now that it's tidy...

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.value  
##   same - different     8.43 39.9  9  0.212  0.83  
##  
## Block = Block2:  
##   contrast      estimate    SE df t.ratio p.value  
##   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.
```