Data Basics

01-Getting-started.Rmd | 02-Data-basics.Rmd

Knit | Insert | Run

```
---
title: "Data Basics"
output: html_document
---

<!-- This file by Charlotte Wickham is licensed under a Creative Commons Attribution 4.0 International License. -->

```{r setup}
library(tidyverse)
library(haven)
library(readxl)
```

```

01-Getting-started.Rmd | 02-Data-basics.Rmd

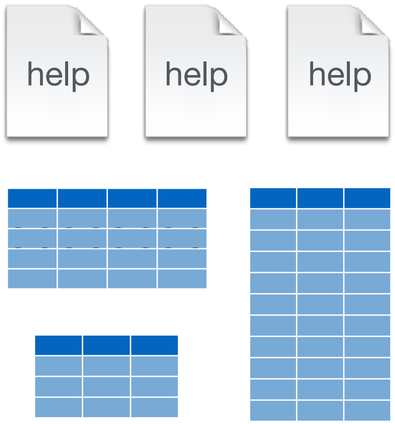
Knit | Insert | Run

```
1 ---
2 title: "Data Basics"
3 output: html_document
4 ---
5
6 <!-- This file by Charlotte Wickham is licensed under a Creative
   Commons Attribution 4.0 International License. -->
7
8
9   rse)
10
11   )
12
13
14
15
16   see much difference between these two in this
                                   one is a tibble and one is a data frame:
17
18
```

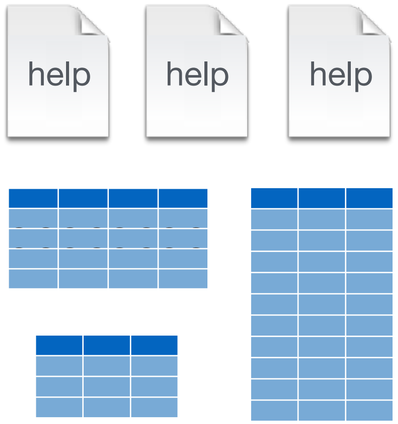
- **Data Basics**
- Chunk 1: setup
- **Tabular Data**
  - Chunk 2
  - Chunk 3
  - **Your turn 1**
  - **Your turn 2**
    - Chunk 4
  - **Your turn 3**
    - Chunk 5
    - Chunk 6
  - **Your turn 4**

23:1 | Chunk 3 | R Markdown

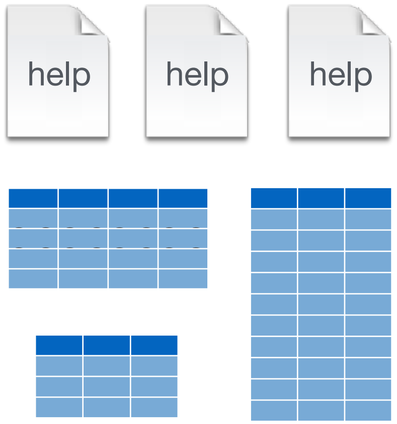
R Packages



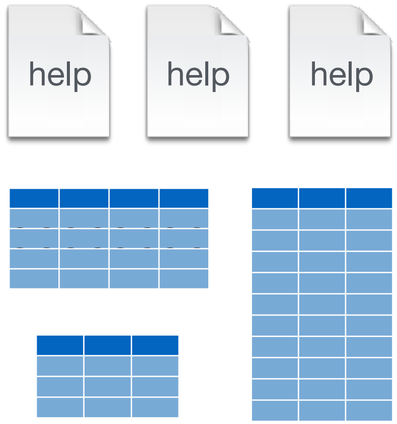
function1()
function2()
function3()
function4()



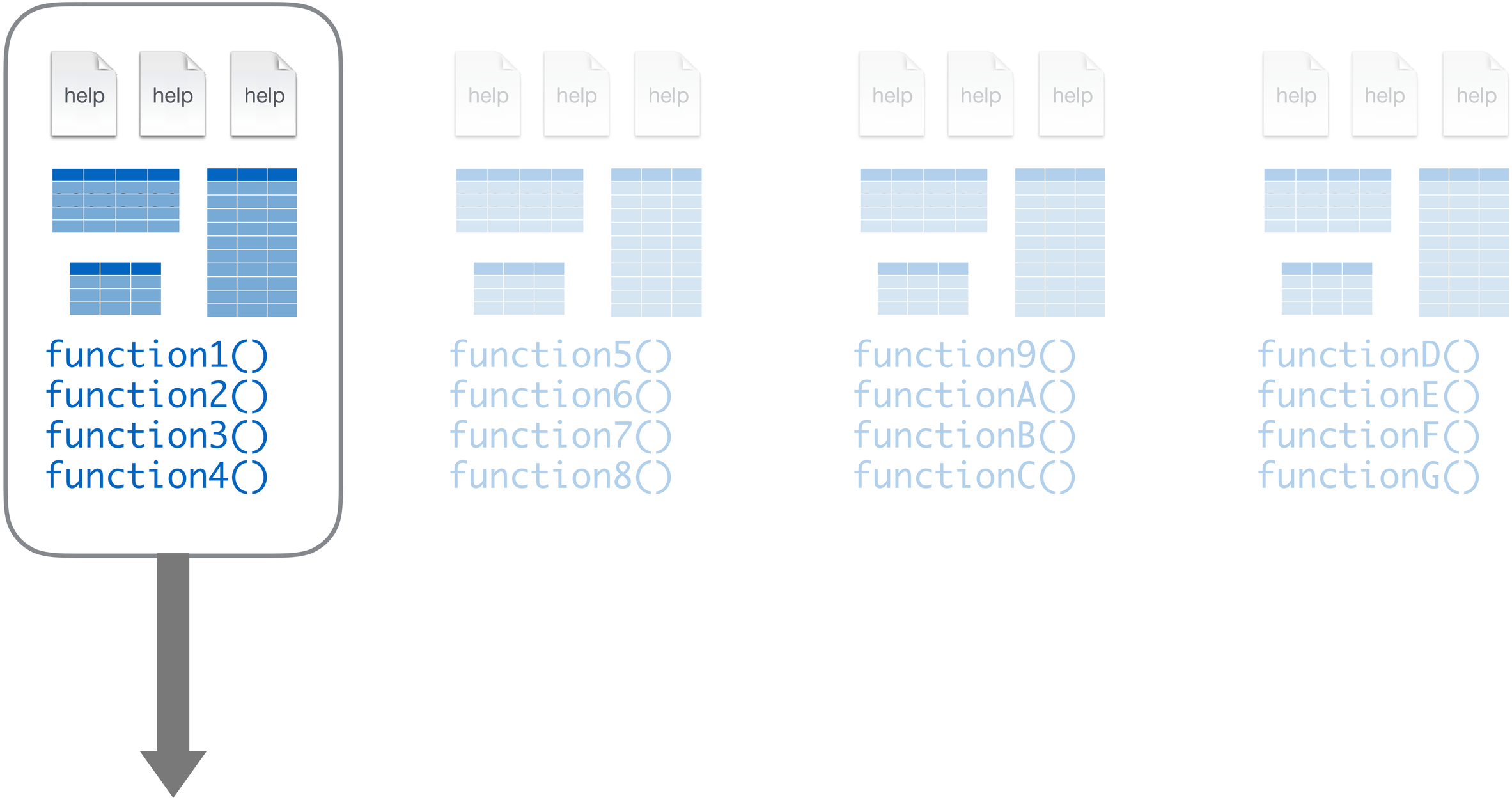
function5()
function6()
function7()
function8()



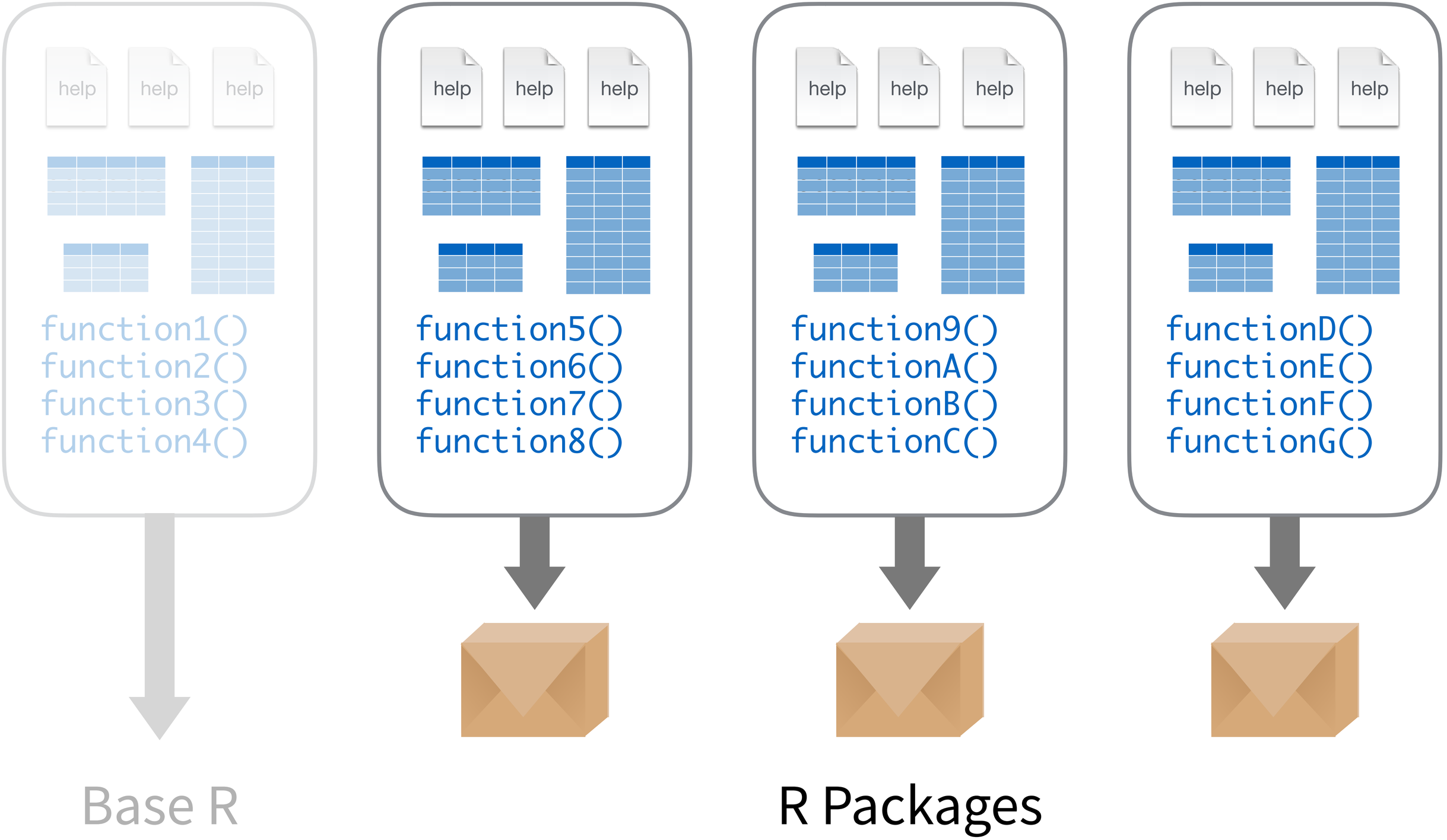
function9()
functionA()
functionB()
functionC()



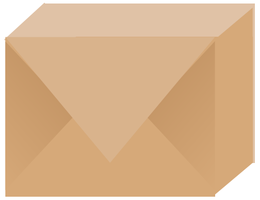
functionD()
functionE()
functionF()
functionG()


help
help
help
function1()
function2()
function3()
function4()
help
help
help
function5()
function6()
function7()
function8()
help
help
help
function9()
functionA()
functionB()
functionC()
help
help
help
functionD()
functionE()
functionF()
functionG()

Base R


help
help
help
function1()
function2()
function3()
function4()
Base R
help
help
help
function5()
function6()
function7()
function8()
help
help
help
function9()
functionA()
functionB()
functionC()
help
help
help
functionD()
functionE()
functionF()
functionG()
R Packages

tidyverse



An R package that serves as a short cut for installing and loading the components of the tidyverse.

```
library("tidyverse")
```

Using packages

1

```
install.packages("foo")
```

Downloads files to computer

1 x per computer

2

```
library("foo")
```

Loads package

1 x per R Session

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

does the equivalent of

```r
install.packages("ggplot2")
install.packages("dplyr")
install.packages("tidyr")
install.packages("readr")
install.packages("purrr")
install.packages("tibble")
install.packages("stringr")
install.packages("forcats")
install.packages("lubridate")
install.packages("hms")
install.packages("DBI")
install.packages("haven")
install.packages("httr")
install.packages("jsonlite")
install.packages("readxl")
install.packages("rvest")
install.packages("xml2")
install.packages("modelr")
install.packages("broom")
```

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

does the equivalent of

```r
install.packages("ggplot2")
install.packages("dplyr")
install.packages("tidyr")
install.packages("readr")
install.packages("purrr")
install.packages("tibble")
install.packages("stringr")
install.packages("forcats")
install.packages("lubridate")
install.packages("hms")
install.packages("DBI")
install.packages("haven")
install.packages("httr")
install.packages("jsonlite")
install.packages("readxl")
install.packages("rvest")
install.packages("xml2")
install.packages("modelr")
install.packages("broom")
```

```r
library("tidyverse")
```

does the equivalent of

```r
library("ggplot2")
library("dplyr")
library("tidyr")
library("readr")
library("purrr")
library("tibble")
library("stringr")
library("forcats")
```

Tabular Data

Data frames and tibbles

The most common kind of data objects, for rectangular data

Data frames - a type of object native to R

Tibbles - a.k.a `tbl` - a type of data frame common in the tidyverse

Tibbles have slightly different default behaviour than data frames, but in Rmarkdowm you mostly won't notice a difference.

Your Turn 1

Take a look at these two datasets, by typing their names into the **console**:

```
quakes
```

```
mpg
```

What do you notice about the difference in the way they are displayed?

quakes is a data frame

```
191  -20.02 184.09   234 5.3       71
192  -18.56 169.31   223 4.7       35
193  -17.87 182.00   569 4.6       12
194  -24.08 179.50   605 4.1       21
195  -32.20 179.61   422 4.6       41
196  -20.36 181.19   637 4.2       23
197  -23.85 182.53   204 4.6       27
198  -24.00 182.75   175 4.5       14
199  -20.41 181.74   538 4.3       31
200  -17.72 180.30   595 5.2       74
 [ reached getOption("max.print") -- omitted 800 rows ]
>
```

mpg is a tibble

```
> mpg
# A tibble: 234 x 11
   manufacturer      model displ  year   cyl      trans
          <chr>      <chr> <dbl> <int> <int>      <chr>
 1         audi         a4   1.8  1999     4   auto(l5)
 2         audi         a4   1.8  1999     4 manual(m5)
 3         audi         a4   2.0  2008     4 manual(m6)
 4         audi         a4   2.0  2008     4   auto(av)
 5         audi         a4   2.8  1999     6   auto(l5)
 6         audi         a4   2.8  1999     6 manual(m5)
 7         audi         a4   3.1  2008     6   auto(av)
 8         audi a4 quattro   1.8  1999     4 manual(m5)
 9         audi a4 quattro   1.8  1999     4   auto(l5)
10         audi a4 quattro   2.0  2008     4 manual(m6)
# ... with 224 more rows, and 5 more variables: drv <chr>,
#   cty <int>, hwy <int>, fl <chr>, class <chr>
```

Your Turn 2

Run the code in the chunk line by line with shortcut Crtl/Cmd + Enter

```
dim(x = mpg)
names(x = mpg)
glimpse(x = mpg)
View(x = mpg)
```

What do each of these functions do?

Getting an overview of data frames/tibbles

```
dim(x = mpg)      # Dimensions of data
names(x = mpg)    # Variable names
glimpse(x = mpg) # Nice overview
View(x = mpg)     # Open Viewer pane
```

Your Turn 3

Write code in the empty chunks to find:

- The number of rows in quakes.
- The names of the variables in quakes.

```
dim(x = quakes)
```

```
names(x = quakes)
```

?

for help on data

quakes and mpg are **built-in** datasets, they come with a package.

You can also use:

```
?data_name
```

to get more info on built in data

Your Turn 4

Try

```
?mpg
```

What is this data?

Vector Data

Vectors

In R vectors hold data all of the same type.

They can be constructed with `c()`

```
c(1, 3, 2, 1, 1)
```

But, you'll usually want to assign them to something

```
my_numbers <- c(1, 3, 2, 1, 1)
```

Basic data types

Integer	Whole numbers	`c(1L, 2L, 3L, 4L)`
Double	Numbers	`c(1, 2, 3, 4)`
Character	Text	`c("1", "2", "3", "4")`
Logical	True or False	`c(TRUE, FALSE, FALSE, TRUE)`

Your Turn 5

Take another look at mpg.

What kind of data is in each column?

```{r}
mpg
```

manufacturer	model	displ	year	cyl	trans	drv	cty	hwy	fl
<chr>	<chr>	<dbl>	<int>	<int>	<chr>	<chr>	<int>	<int>	<chr>

Importing Data

readr

Simple, consistent functions for working with (mostly) plain text data.

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

haven

Simple, consistent functions for working with SAS, SPSS and Stata data

```
# install.packages("tidyverse")
library(haven)
```

readxl

Simple, consistent functions for working Excel data

```
# install.packages("tidyverse")
library(readxl)
```

Other types of data

package	accesses
jsonlite	json
xml2	xml
httr	web API's
rvest	web pages (web scraping)
DBI	databases
sparklyr	data loaded into spark

readr

```
df <- read_csv("path/to/file.csv", ...)
```

object to save output into

path from working directory to file

readr

```
df <- read_csv("path/to/file.csv", ...)
```

haven

```
df <- read_spss("path/to/file.sav", ...)
```

readxl

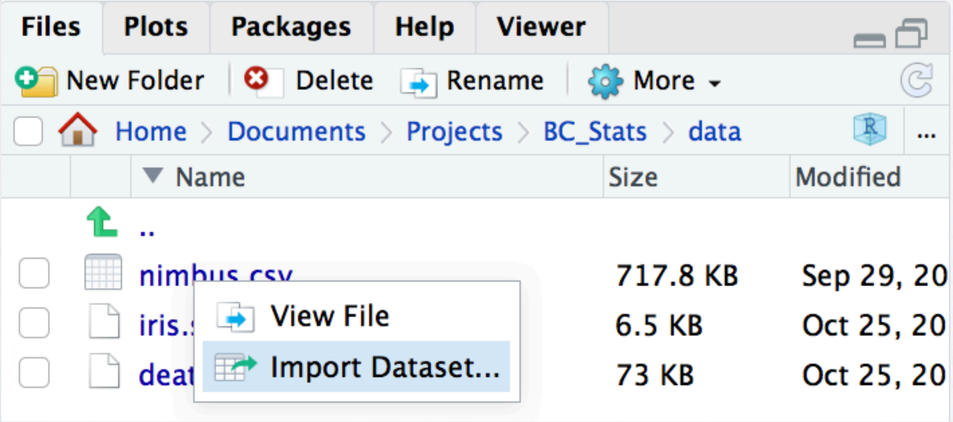
```
df <- read_excel("path/to/file.xls", ...)
```

Your Turn 6

Take a look in the `data/` directory.

Try reading in each file with the appropriate function.

(Alternatively, try the Import Data tool)



When import goes wrong...

And it will.

Check:

- data type of each column is correct
- basic summaries seem reasonable (e.g. number of rows, min and max of columns)

Try to identify the problem, then read the help for the import function for a solution.

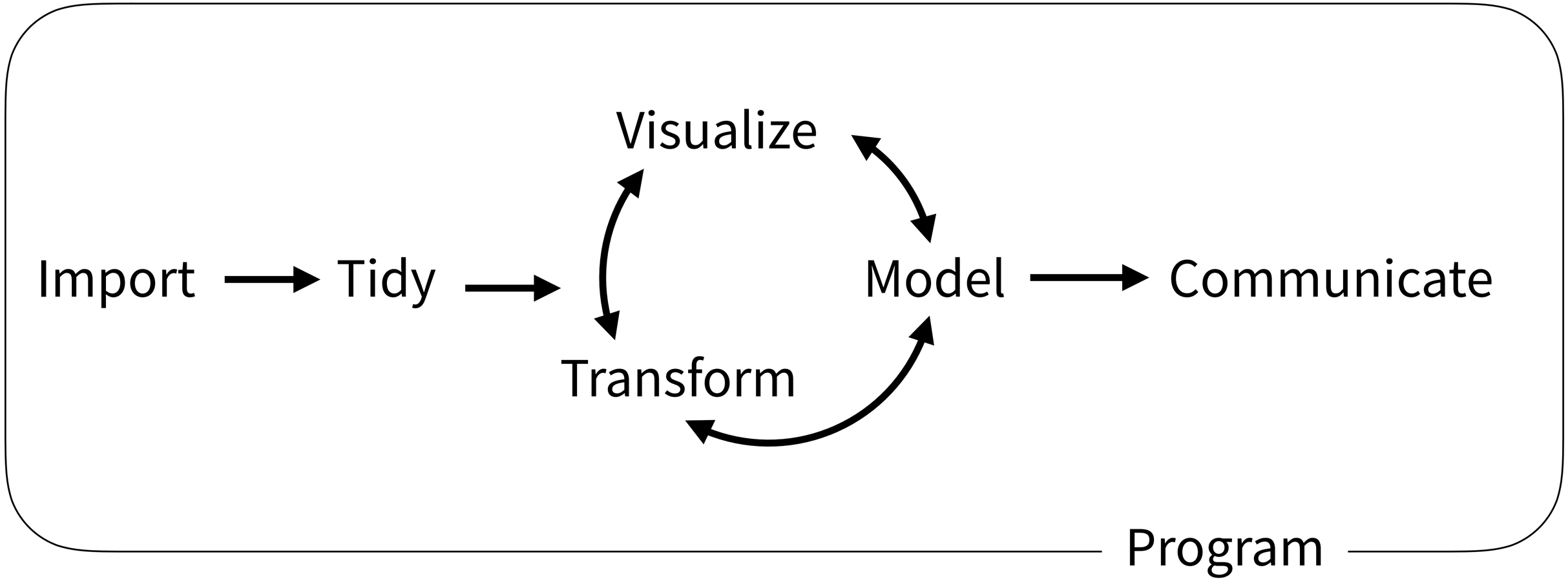
Your Turn 7

Can you see what is wrong with the Excel file when it is imported?

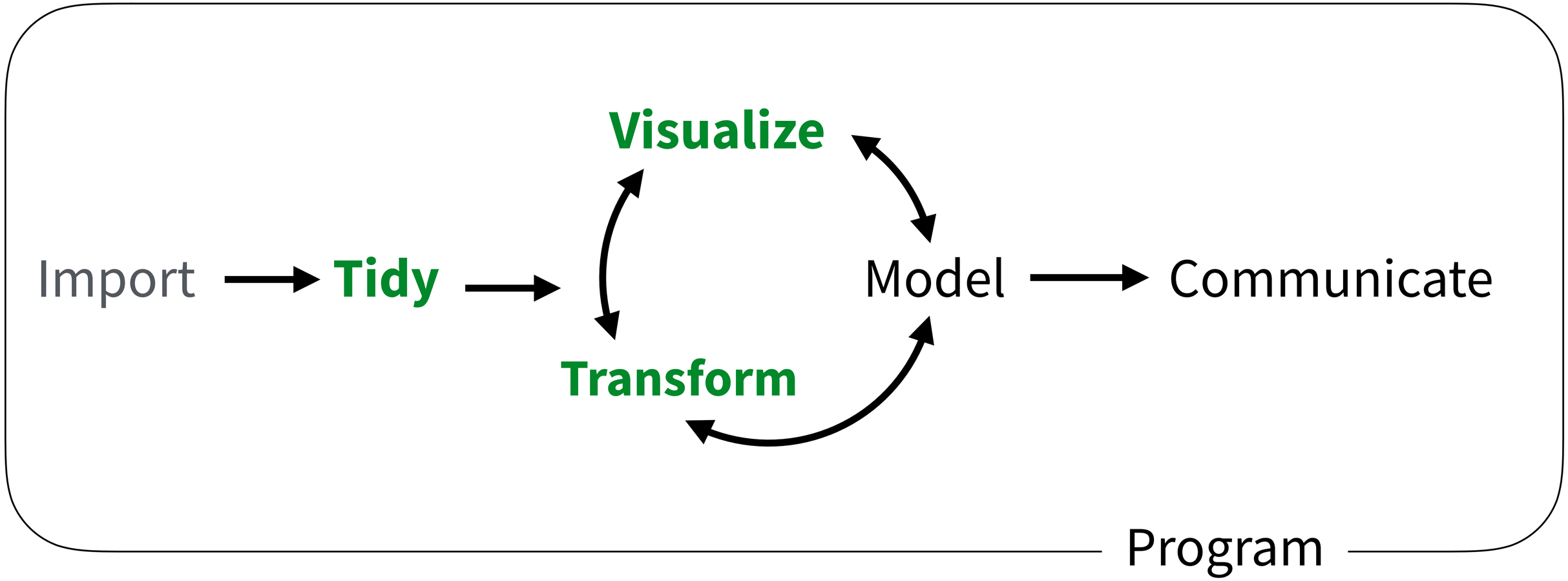
Scan the *Arguments* section of `?read_excel`, can you find an argument that might help? Try it!

Wrapping Up

(Applied) **Data Science**



(Applied) **Data Science**



Tomorrow

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = cty, y = hwy), alpha = 0.2)
```