Introduction to tidyverse

https://www.meetup.com/Lucerne-R-User-Group/

https://github.com/Lucerne-R-User-Group/2020_06_24-inaugural-meeting

Dr Andrea De Angelis

24 June 2020

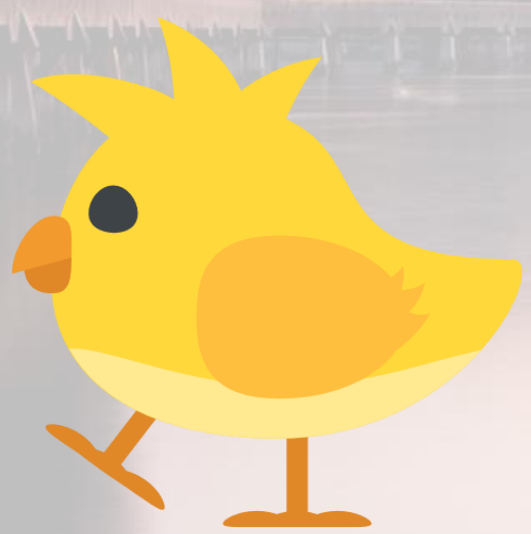
Structure of the workshop

1 **R basics**

2 **tidyverse & R4DS**

3 **dplyr & ggplot2**

Basic notions



Point-and-click

Coding (scripts)

Proprietary software

RStudio, PBC

Free software

(as in "free speech" not as "free beer")

A language

Writing software

R the language

Values

```
1, "Florida", "2010-01-25"
```

R the language

Values

```
1, "Florida", "2010-01-25"
```

Objects

```
x <- 10/3
```

R the language

Values

```
1, "Florida", "2010-01-25"
```

Objects

```
x <- 10/3
```

A name without quotes

<-
(it looks like an arrow)

object, values, or function result

R the language

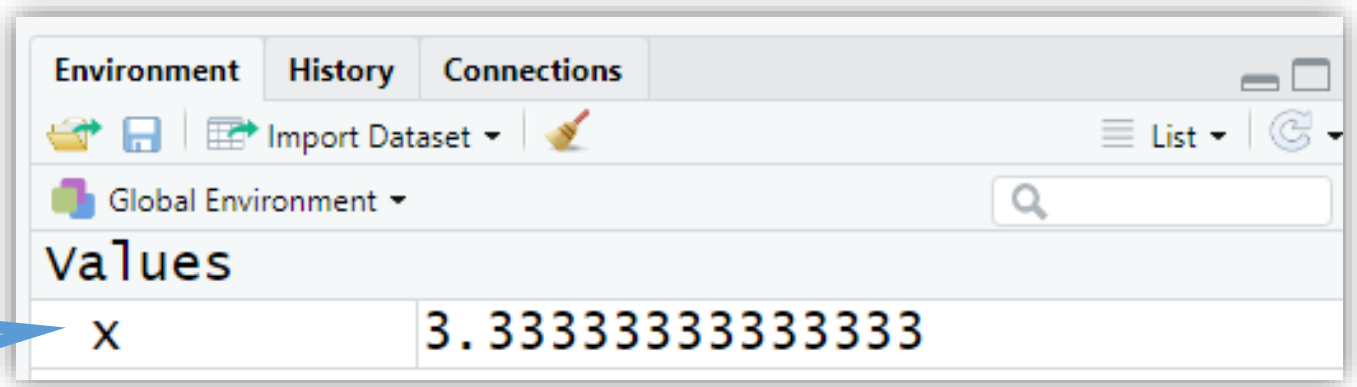
Values

```
1, "Florida", "2010-01-25"
```

Objects

```
x <- 10/3
```

Available objects appear in a box called "global environment"



R the language

Values

```
1, "Florida", "2010-01-25"
```

Objects

```
x <- c(10/3, 2, 1.42)
```

Put multiple values in an object combining with **c()**

R the language

Values

```
1, "Florida", "2010-01-25"
```

Objects

```
x <- c(10/3, 2, 1.42)
```

Functions

```
mean(x, na.rm = TRUE)
```

LuzeRn

R the language

Values

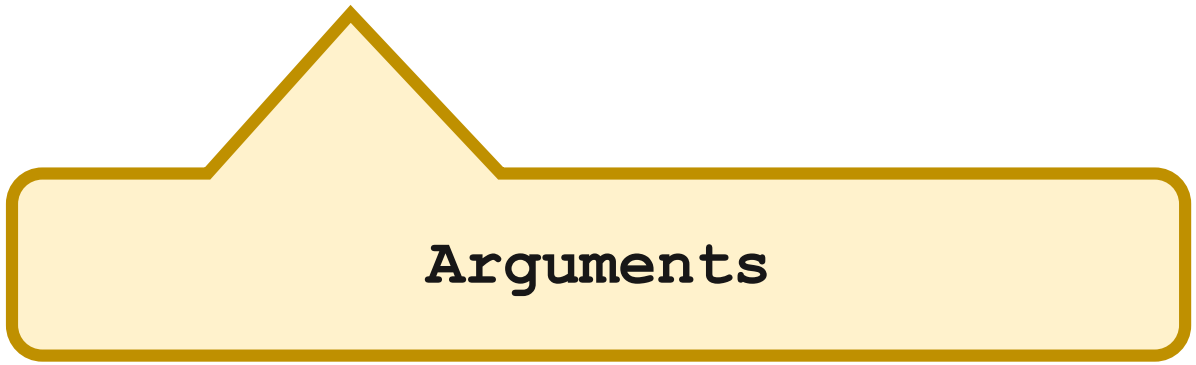
```
1, "Florida", "2010-01-25"
```

Objects

```
x <- c(10/3, 2, 1.42)
```

Functions

```
mean(x, na.rm = TRUE)
```



R the language

Values

```
1, "Florida", "2010-01-25"
```

Objects

```
x <- c(10/3, 2, 1.42)
```

Functions

```
mean(x, na.rm = TRUE)
```

Arguments: can be objects or set (equal to) values

Your Turn

Which of these are **numbers?**

A `1`

B `"1"`

C `"one"`

D `one`

00:15

Your Turn

Which of these are **numbers?**

A

```
1
# number
```

B

```
"1"
# strings
```

C

```
"one"
# strings
```

D

```
one
# object
```

Your Turn

Which of these are **numbers?**

A

```
1
# number
```

B

```
"1"
# strings
```

C

```
"one"
# strings
```

D

```
one
# object
```

Your Turn

Suppose `one <- 1`. Which of these **will work**?

A `log(1)`

B `log(one)`

C `log("1")`

D `log("one")`

00:15

Your Turn

Suppose `one <- 1`. Which of these **will work**?

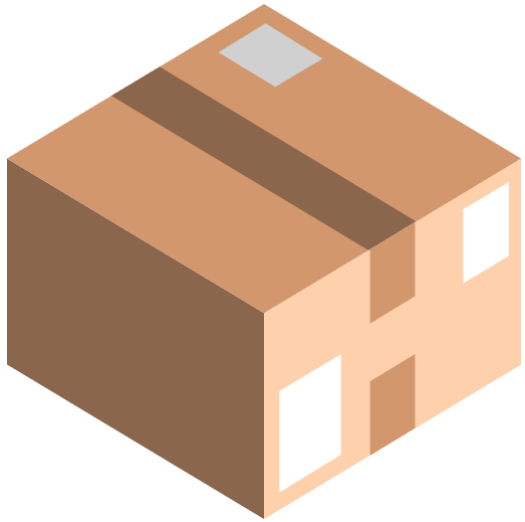
A `log(1)`

B `log(one)`

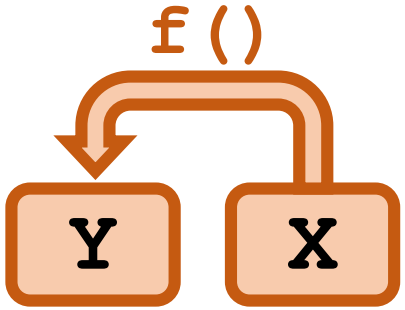
C `log("1")`

D `log("one")`

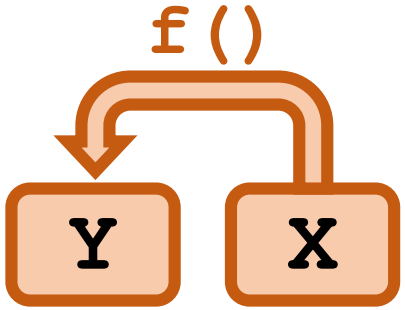
R 'packages'



Functions



Functions

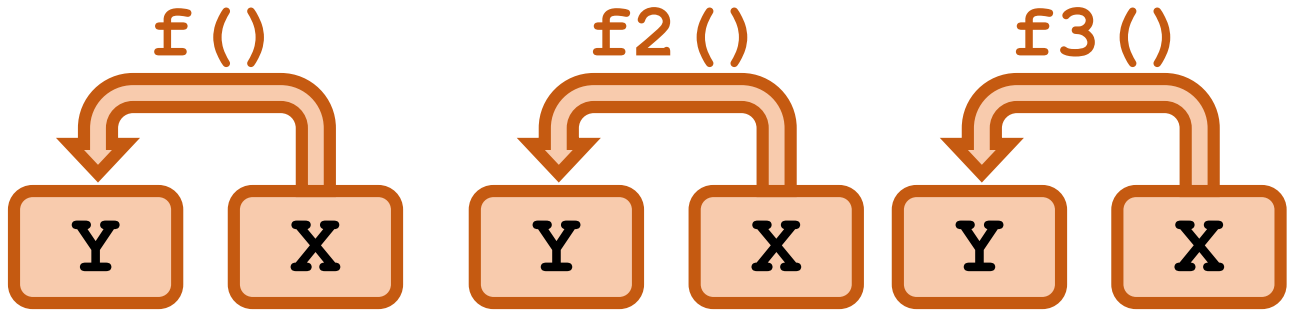


Objects

iris

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa

Many functions



Many objects

iris

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa

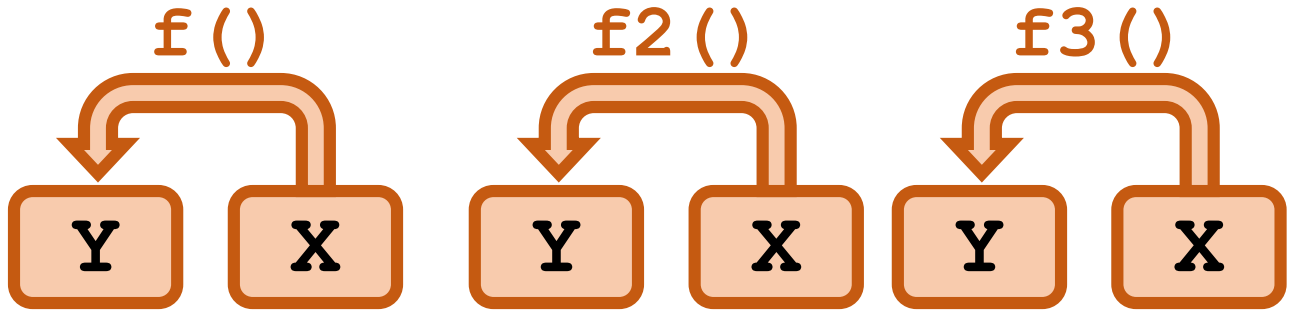
mtcars

	mpg	cyl	disp	hp	drat	wt
Mazda RX4	21.0	6	160.0	110	3.90	2.620
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875
Datsun 710	22.8	4	108.0	93	3.85	2.320
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440
Valiant	18.1	6	225.0	105	2.76	3.460
Duster 360	14.3	8	360.0	245	3.21	3.570
Merc 240D	24.4	4	146.7	62	3.69	3.190
Merc 230	22.8	4	140.8	95	3.92	3.150
Merc 280	19.2	6	167.6	123	3.92	3.440

mpg

	manufacturer	model	displ	year	cyl
1	audi	a4	1.8	1999	4
2	audi	a4	1.8	1999	4
3	audi	a4	2.0	2008	4
4	audi	a4	2.0	2008	4
5	audi	a4	2.8	1999	6
6	audi	a4	2.8	1999	6
7	audi	a4	3.1	2008	6
8	audi	a4 quattro	1.8	1999	4

Many functions



Many objects

iris

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa

mtcars

	mpg	cyl	disp	hp	drat	wt
Mazda RX4	21.0	6	160.0	110	3.90	2.620
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875
Datsun 710	22.8	4	108.0	93	3.85	2.320
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440
Valiant	18.1	6	225.0	105	2.76	3.460
Duster 360	14.3	8	360.0	245	3.21	3.570
Merc 240D	24.4	4	146.7	62	3.69	3.190
Merc 230	22.8	4	140.8	95	3.92	3.150
Merc 280	19.2	6	167.6	123	3.92	3.440

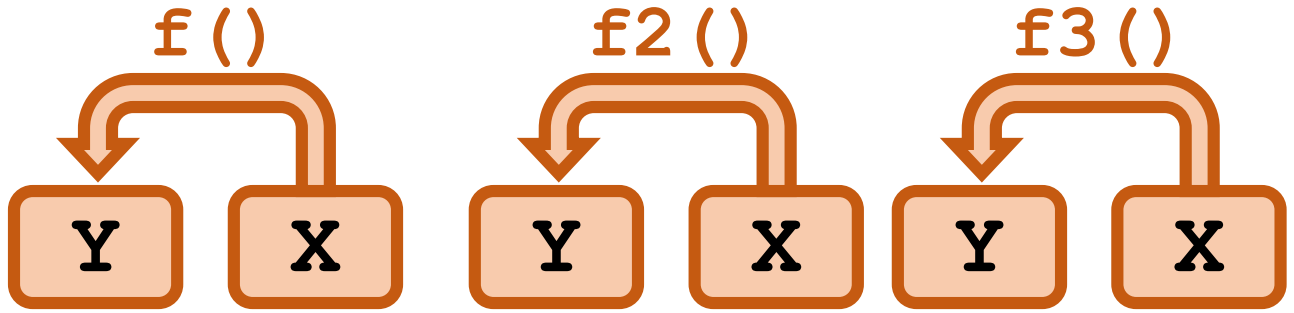
mpg

	manufacturer	model	displ	year	cyl
1	audi	a4	1.8	1999	4
2	audi	a4	1.8	1999	4
3	audi	a4	2.0	2008	4
4	audi	a4	2.0	2008	4
5	audi	a4	2.8	1999	6
6	audi	a4	2.8	1999	6
7	audi	a4	3.1	2008	6
8	audi	a4 quattro	1.8	1999	4

Package

foo

Many functions



Many objects

iris

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa

mtcars

	mpg	cyl	disp	hp	drat	wt
Mazda RX4	21.0	6	160.0	110	3.90	2.620
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875
Datsun 710	22.8	4	108.0	93	3.85	2.320
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440
Valiant	18.1	6	225.0	105	2.76	3.460
Duster 360	14.3	8	360.0	245	3.21	3.570
Merc 240D	24.4	4	146.7	62	3.69	3.190
Merc 230	22.8	4	140.8	95	3.92	3.150
Merc 280	19.2	6	167.6	123	3.92	3.440

mpg

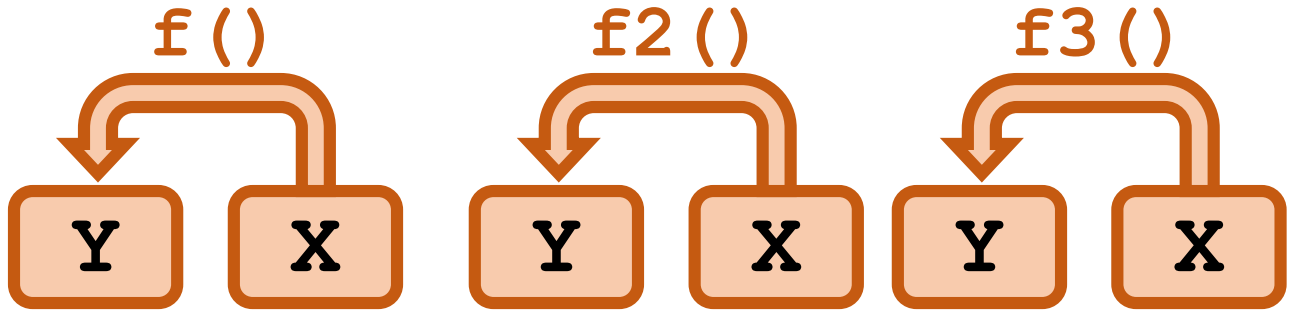
	manufacturer	model	displ	year	cyl
1	audi	a4	1.8	1999	4
2	audi	a4	1.8	1999	4
3	audi	a4	2.0	2008	4
4	audi	a4	2.0	2008	4
5	audi	a4	2.8	1999	6
6	audi	a4	2.8	1999	6
7	audi	a4	3.1	2008	6
8	audi	a4 quattro	1.8	1999	4

Package

foo

```
install.packages("foo")
library("foo")
f2(iris)
```

Many functions



Many objects

iris

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa

mtcars

	mpg	cyl	disp	hp	drat	wt
Mazda RX4	21.0	6	160.0	110	3.90	2.620
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875
Datsun 710	22.8	4	108.0	93	3.85	2.320
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440
Valiant	18.1	6	225.0	105	2.76	3.460
Duster 360	14.3	8	360.0	245	3.21	3.570
Merc 240D	24.4	4	146.7	62	3.69	3.190
Merc 230	22.8	4	140.8	95	3.92	3.150
Merc 280	19.2	6	167.6	123	3.92	3.440

mpg

	manufacturer	model	displ	year	cyl
1	audi	a4	1.8	1999	4
2	audi	a4	1.8	1999	4
3	audi	a4	2.0	2008	4
4	audi	a4	2.0	2008	4
5	audi	a4	2.8	1999	6
6	audi	a4	2.8	1999	6
7	audi	a4	3.1	2008	6
8	audi	a4 quattro	1.8	1999	4

Package

foo

```
install.packages("foo")
library("foo")
f2(iris)
```

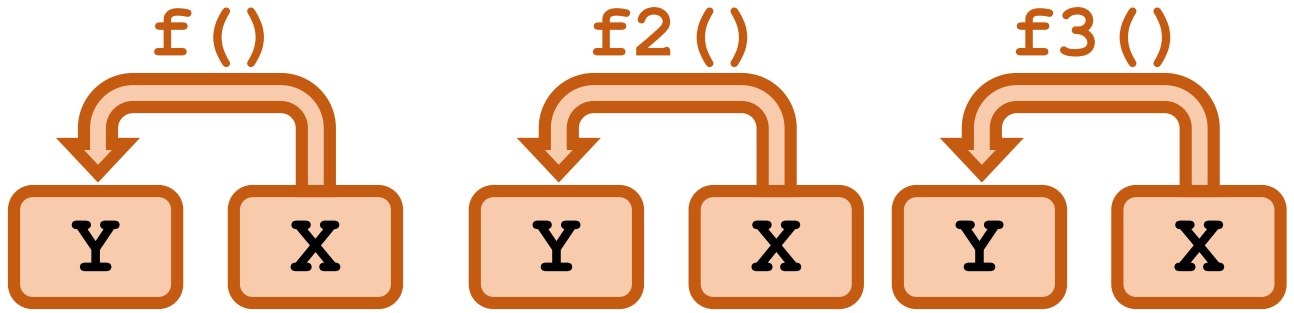
once per computer

once per session

LuzeRn

https://www.meetup.com/Lucerne-R-User-Group/

Many functions



Many objects

iris

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa

mtcars

	mpg	cyl	disp	hp	drat	wt
Mazda RX4	21.0	6	160.0	110	3.90	2.620
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875
Datsun 710	22.8	4	108.0	93	3.85	2.320
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440
Valiant	18.1	6	225.0	105	2.76	3.460
Duster 360	14.3	8	360.0	245	3.21	3.570
Merc 240D	24.4	4	146.7	62	3.69	3.190
Merc 230	22.8	4	140.8	95	3.92	3.150
Merc 280	19.2	6	167.6	123	3.92	3.440

mpg

	manufacturer	model	displ	year	cyl
1	audi	a4	1.8	1999	4
2	audi	a4	1.8	1999	4
3	audi	a4	2.0	2008	4
4	audi	a4	2.0	2008	4
5	audi	a4	2.8	1999	6
6	audi	a4	2.8	1999	6
7	audi	a4	3.1	2008	6
8	audi	a4 quattro	1.8	1999	4

Package

foo

```
install.packages("foo")
foo::f2(iris)
```

once per computer

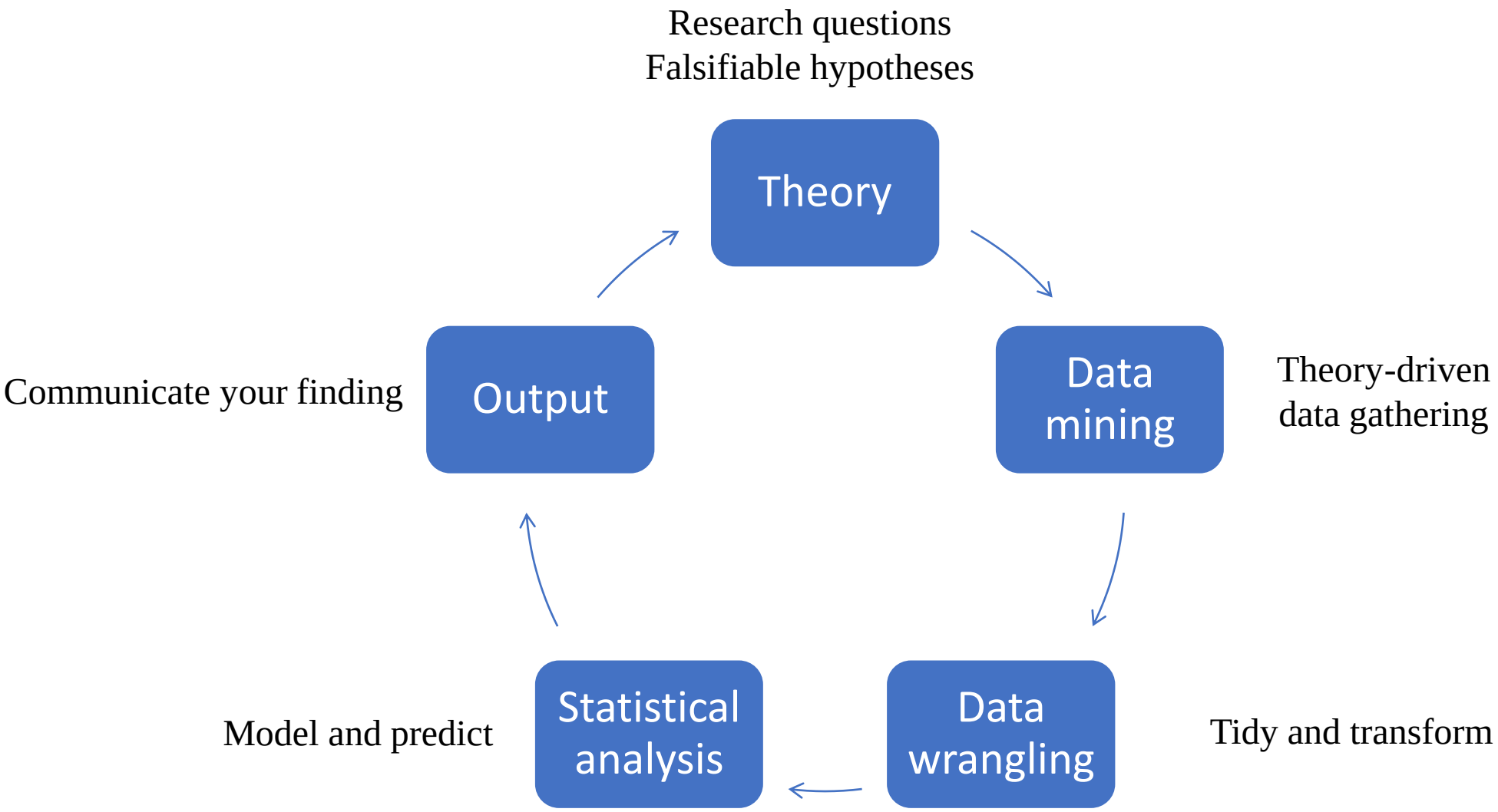
tidyverse
&
Data Science

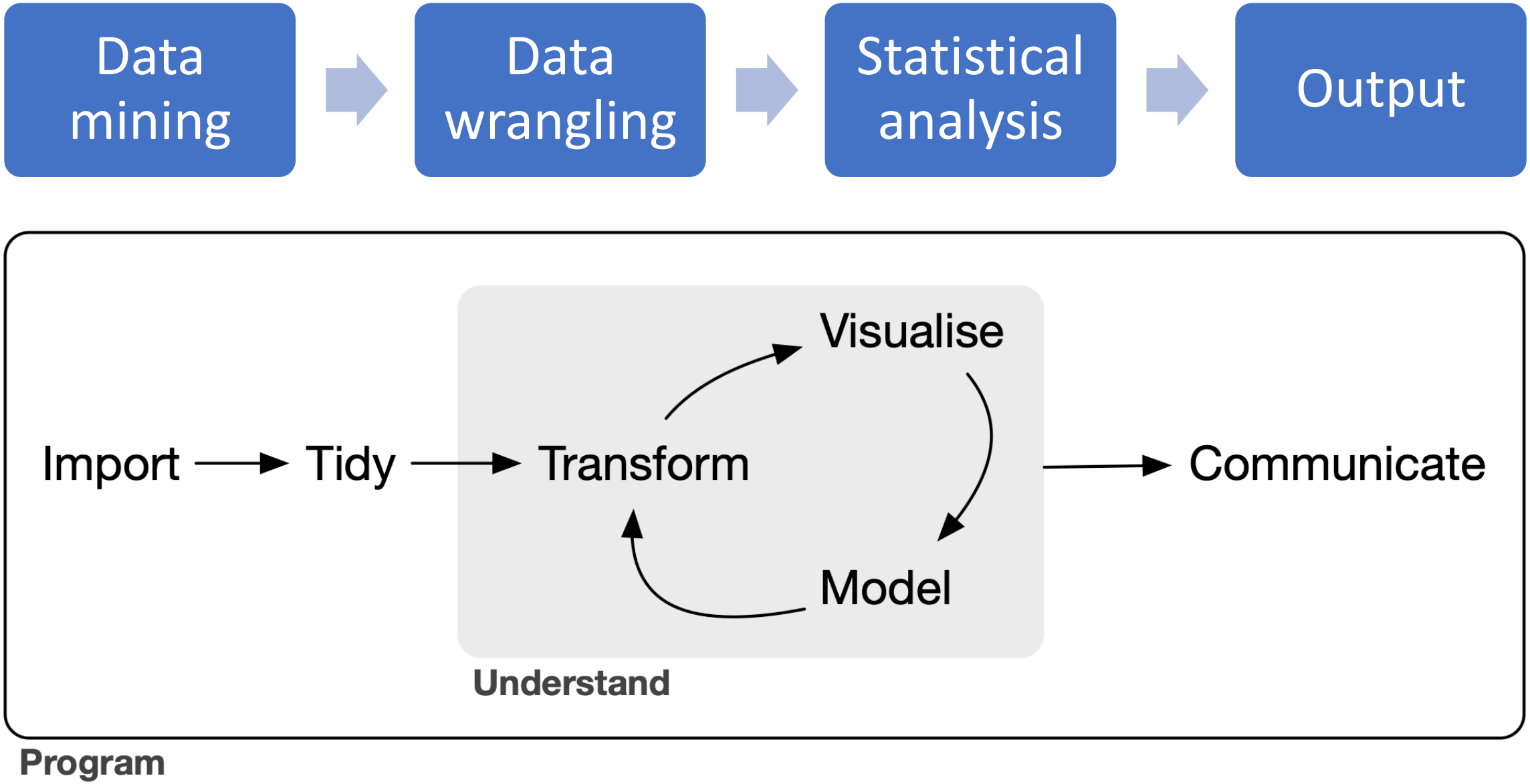
Many packages

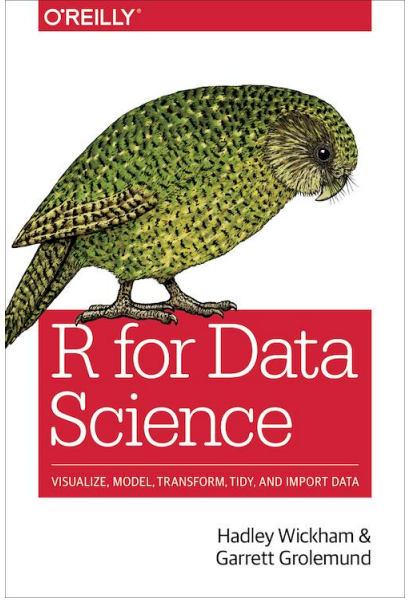

```
# install.packages("tidyverse")
library("tidyverse")
```

a shortcut

The Data Science Cycle

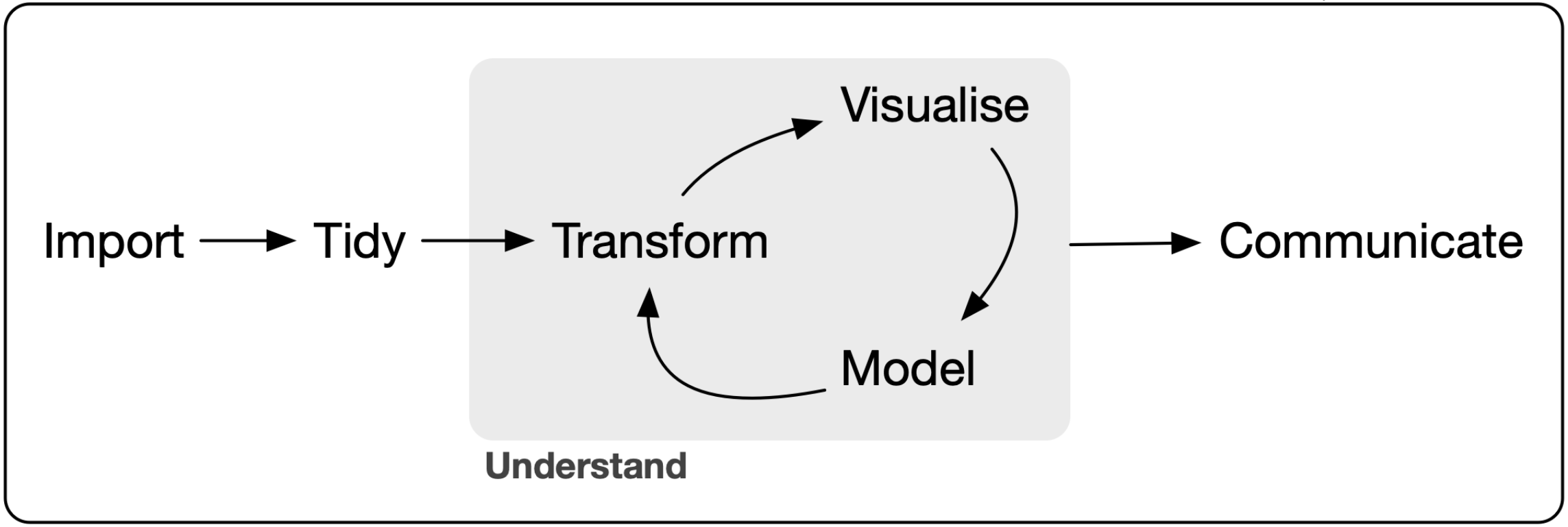


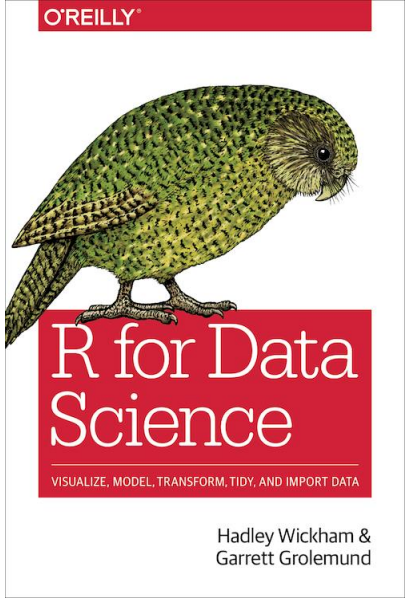

Data mining
Data wrangling
Statistical analysis
Output
Import
Tidy
Transform
Visualise
Model
Communicate
Understand
Program



R for Data Science

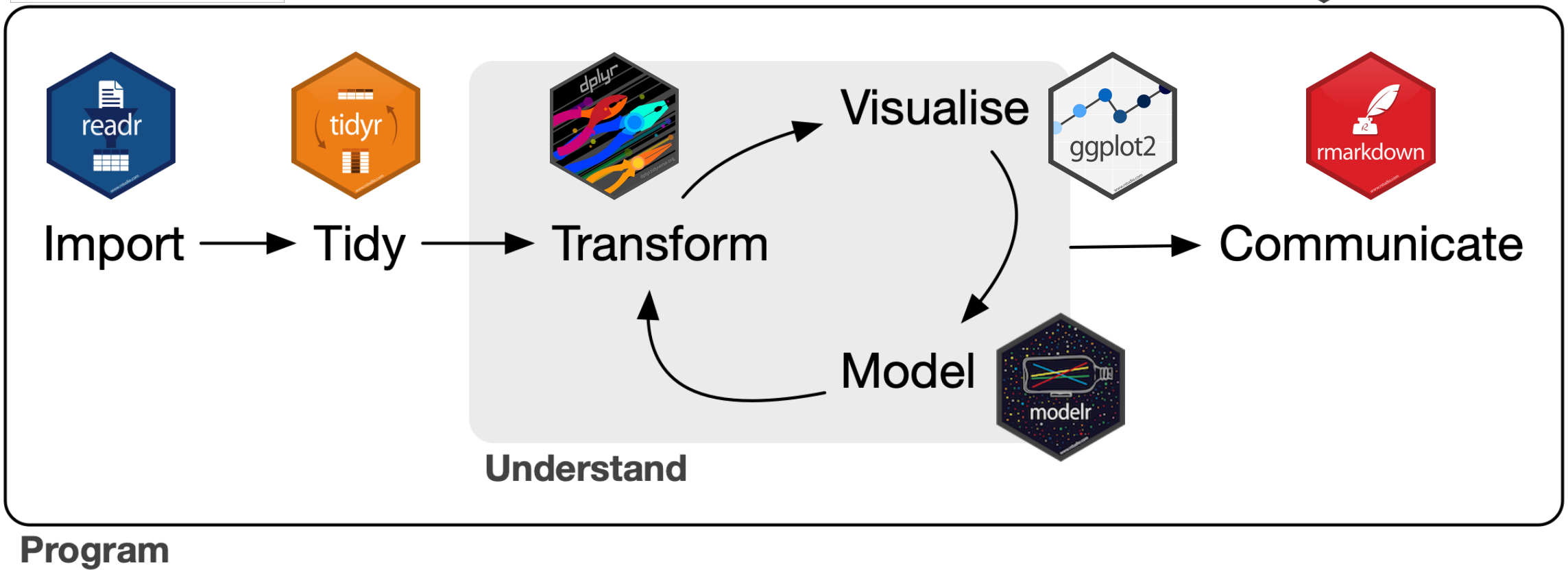
https://r4ds.had.co.nz/





R for Data Science

https://r4ds.had.co.nz/



Your Turn

If you don't have RStudio installed:

Access RStudio Cloud at this link

Register / Log in > Projects

If you do have RStudio installed:

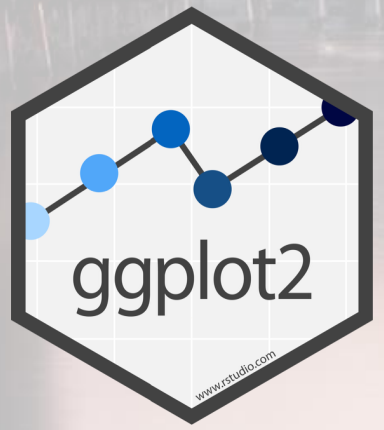
Clone this GitHub repo

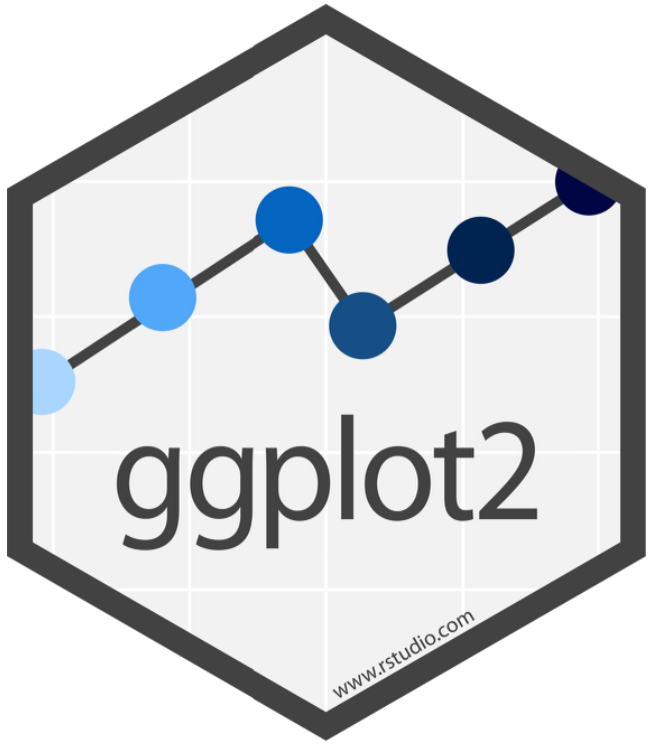
Then, open `01_introduction.R`

Load the tidyverse packages

03:00

Data visualization with ggplot2

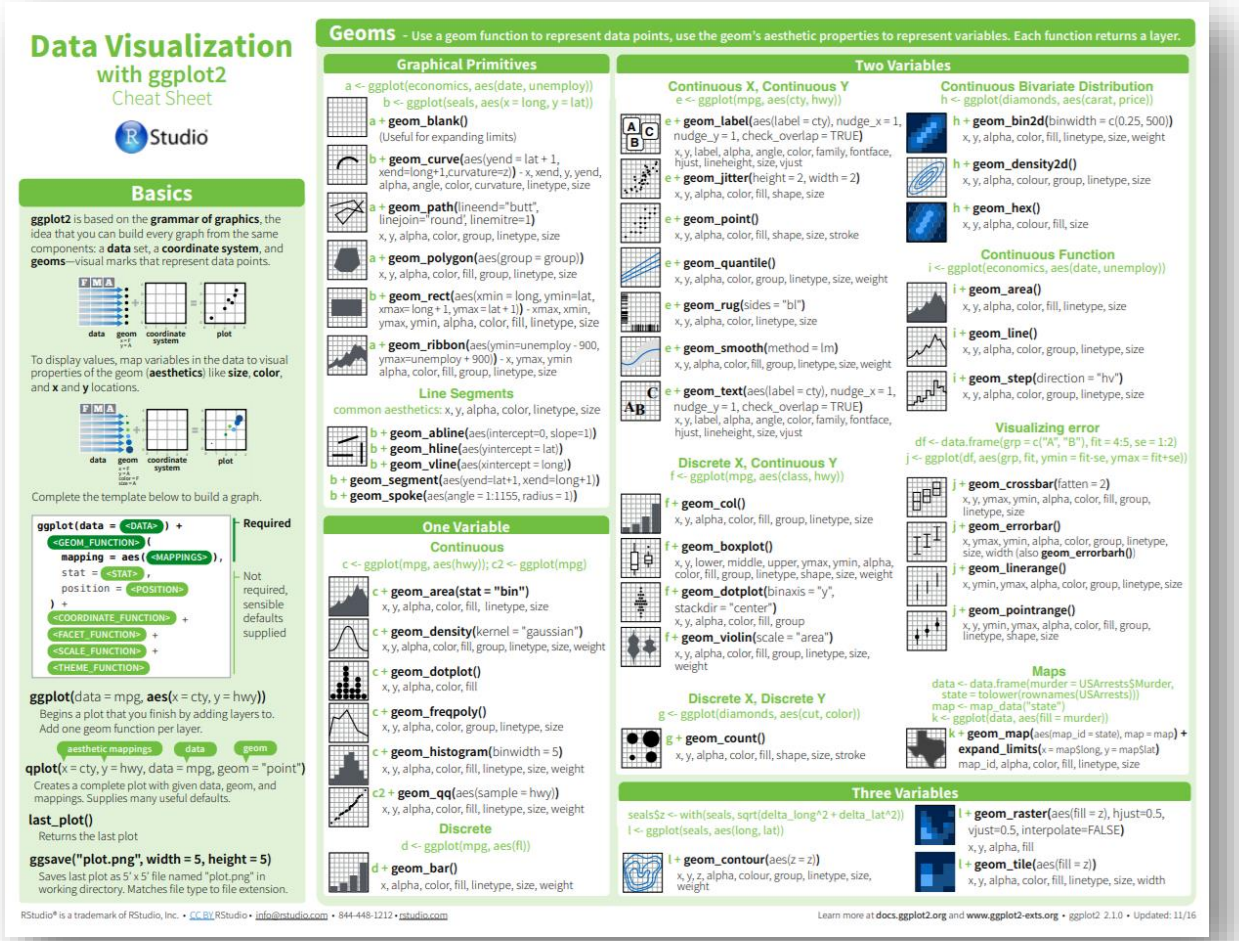




A package for creating graphics based on common principles (Grammar of Graphics).

You provide **data**, tell ggplot2 how to **map variables to aesthetics** and which **geometries** to use.
ggplot2 takes care of the rest.

ggplot2 cheatsheet



ggplot2

Luzern

https://www.meetup.com/Lucerne-R-User-Group/

Three elements of visualization

You provide **data**, tell ggplot2 how to **map** variables to **aesthetics** and which **geometries** to use.

Three elements of visualization

You provide **data**, tell ggplot2 how to **map** variables to **aesthetics** and which **geometries** to use.

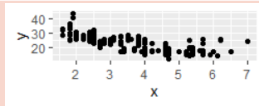
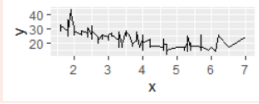
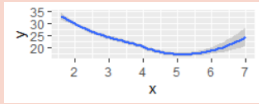
	country	continent	year	lifeExp	pop	gdpPercap
1	Afghanistan	Asia	1952	28.80100	8425333	779.4453
2	Afghanistan	Asia	1957	30.33200	9240934	820.8530
3	Afghanistan	Asia	1962	31.99700	10267083	853.1007
4	Afghanistan	Asia	1967	34.02000	11537966	836.1971
5	Afghanistan	Asia	1972	36.08800	13079460	739.9811
6	Afghanistan	Asia	1977	38.43800	14880372	786.1134
7	Afghanistan	Asia	1982	39.85400	12881816	978.0114
8	Afghanistan	Asia	1987	40.82200	13867957	852.3959
9	Afghanistan	Asia	1992	41.67400	16317921	649.3414
10	Afghanistan	Asia	1997	41.76300	22227415	635.3414
11	Afghanistan	Asia	2002	42.12900	25268405	726.7341
12	Afghanistan	Asia	2007	43.82800	31889923	974.5803
13	Albania	Europe	1952	55.23000	1282697	1601.0561

Data

Aesthetic mapping

Geometric objects

Aesthetic	Variable
x	`year`
y	`lifeExp`
size	`pop`
shape	`Continent`
...	...

Geometry	
point	
line	
smoother	
...	

mpg data

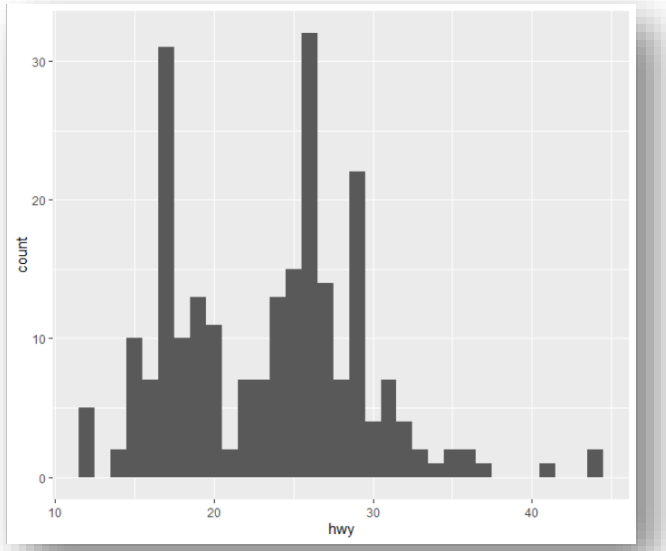
Fuel economy data for 38 models of cars

```
mpg  # print data
```

```
?mpg # access documentation
```

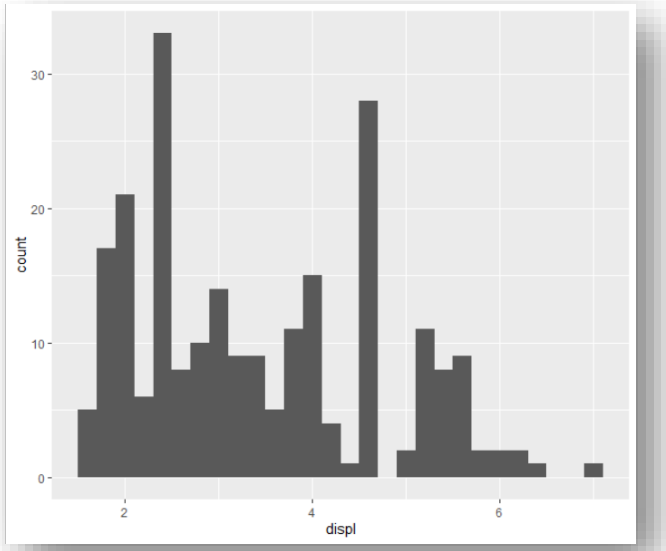
hwy

highway miles per gallon



displ

engine displacement, in litres



mpg data

Fuel economy data for 38 models of cars

```
View(mpg)
```

mpg

Filter

	manufacturer	model	displ	year	cyl	trans	drv	cty	hwy	fl	class
1	audi	a4	1.8	1999	4	auto(l5)	f	18	29	p	compact
2	audi	a4	1.8	1999	4	manual(m5)	f	21	29	p	compact
3	audi	a4	2.0	2008	4	manual(m6)	f	20	31	p	compact
4	audi	a4	2.0	2008	4	auto(av)	f	21	30	p	compact
5	audi	a4	2.8	1999	6	auto(l5)	f	16	26	p	compact
6	audi	a4	2.8	1999	6	manual(m5)	f	18	26	p	compact
7	audi	a4	3.1	2008	6	auto(av)	f	18	27	p	compact
8	audi	a4 quattro	1.8	1999	4	manual(m5)	4	18	26	p	compact
9	audi	a4 quattro	1.8	1999	4	auto(l5)	4	16	25	p	compact
10	audi	a4 quattro	2.0	2008	4	manual(m6)	4	20	28	p	compact

Your Turn

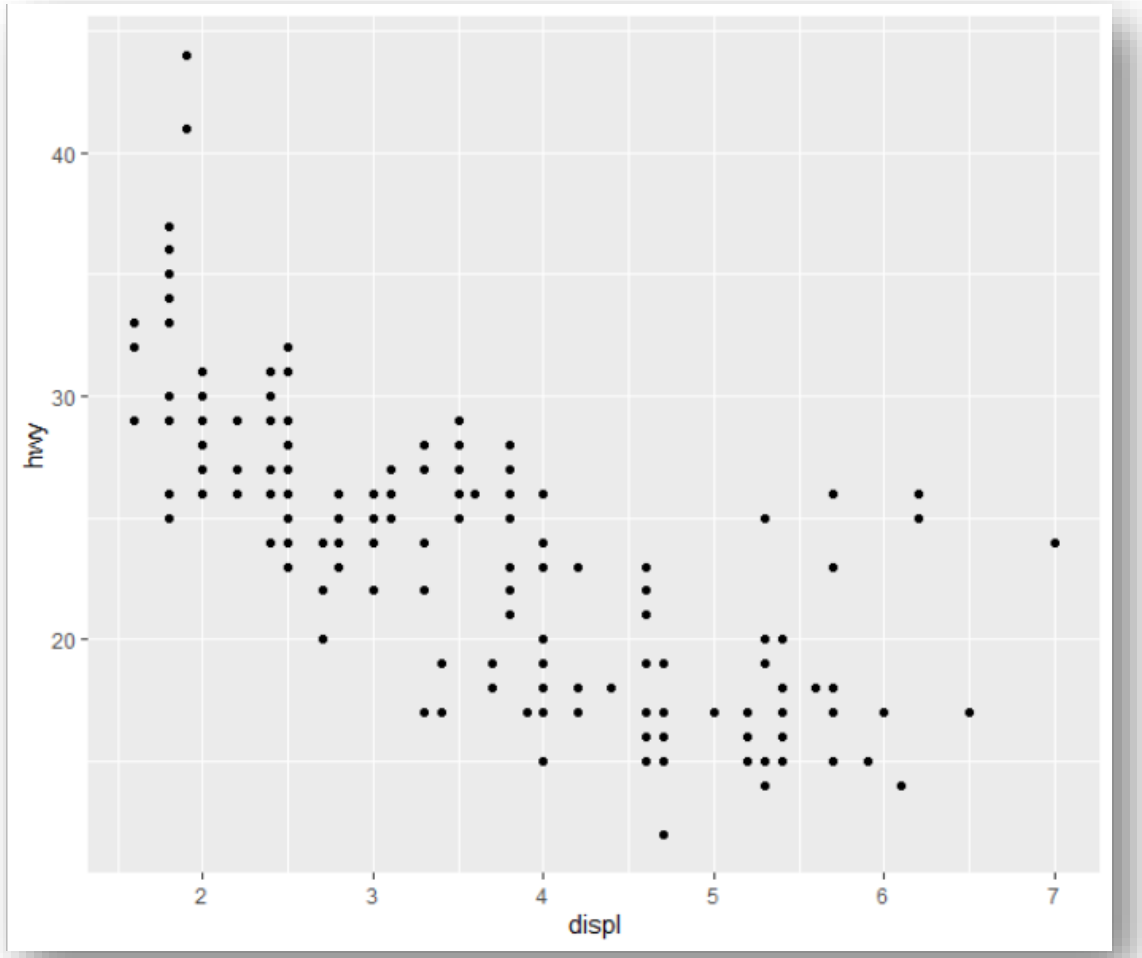
Read the following code and **mentally predict** the resulting plot.

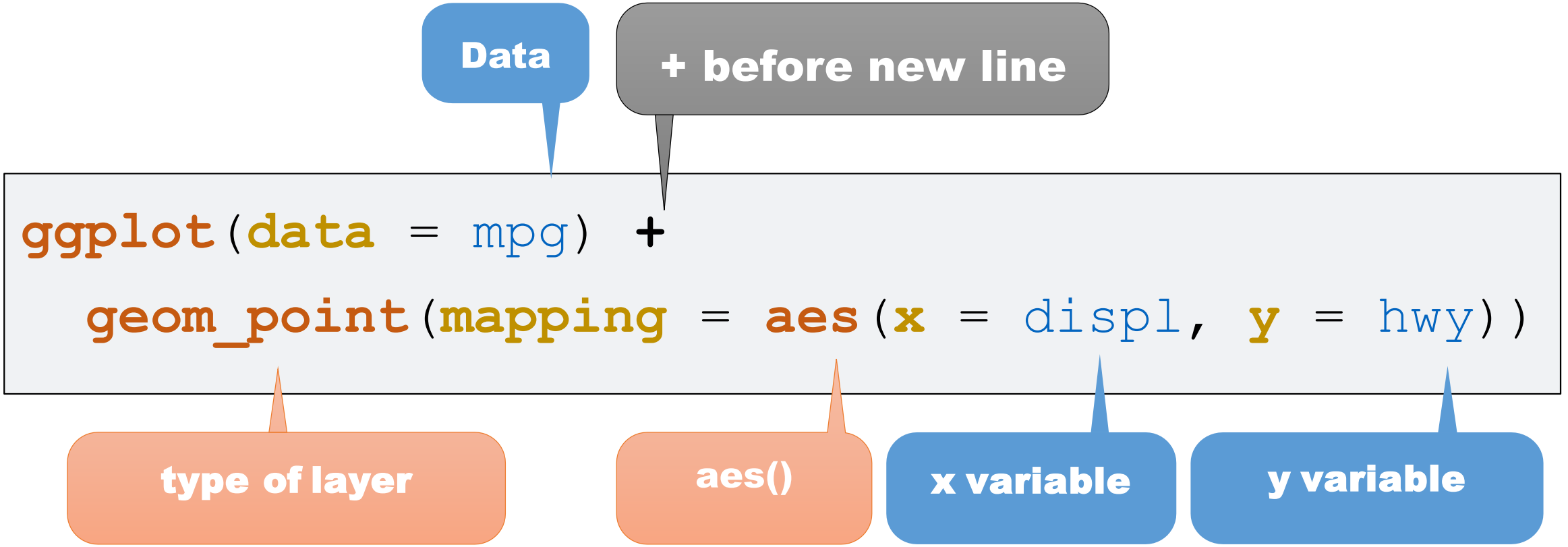
Only then, type it in the `01_introduction` script and execute.
Tip: pay attention to spelling, capitalization, and parentheses!

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy))
```

02:00

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy))
```





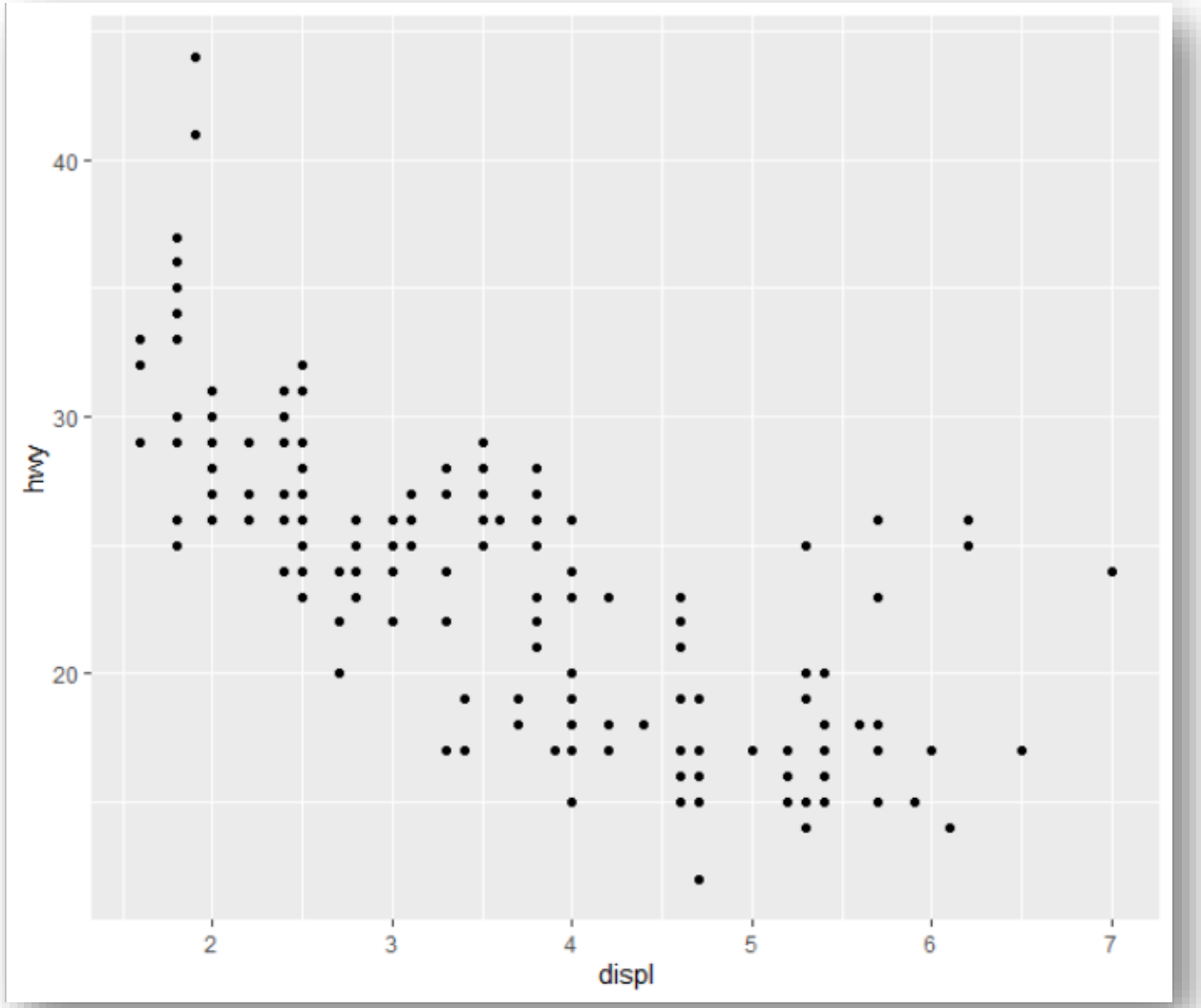
```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy))
```

1. Initialize a plot with `ggplot()`
2. Add layers with `geom_***()`
3. Map variables with `aes()`

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy))
```

1. Initialize a plot with `ggplot(data)`
2. Add layers with `geom_***(mapping)`
3. Map variables with `aes(x, y, size, ...)`

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy))
```



ggplot2

Your Turn

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy))
```

How can I **draw a "smoother"** instead of the points?

Your Turn

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy))
```

How can I **draw a "smoother"** instead of the points?

A Changing the mapping variables

B Changing the data

C Changing the geometry

D Changing one of the aestetics

00:15

Your Turn

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy))
```

How can I **draw a smoother** instead of the points?

A. Changing the mapping variables

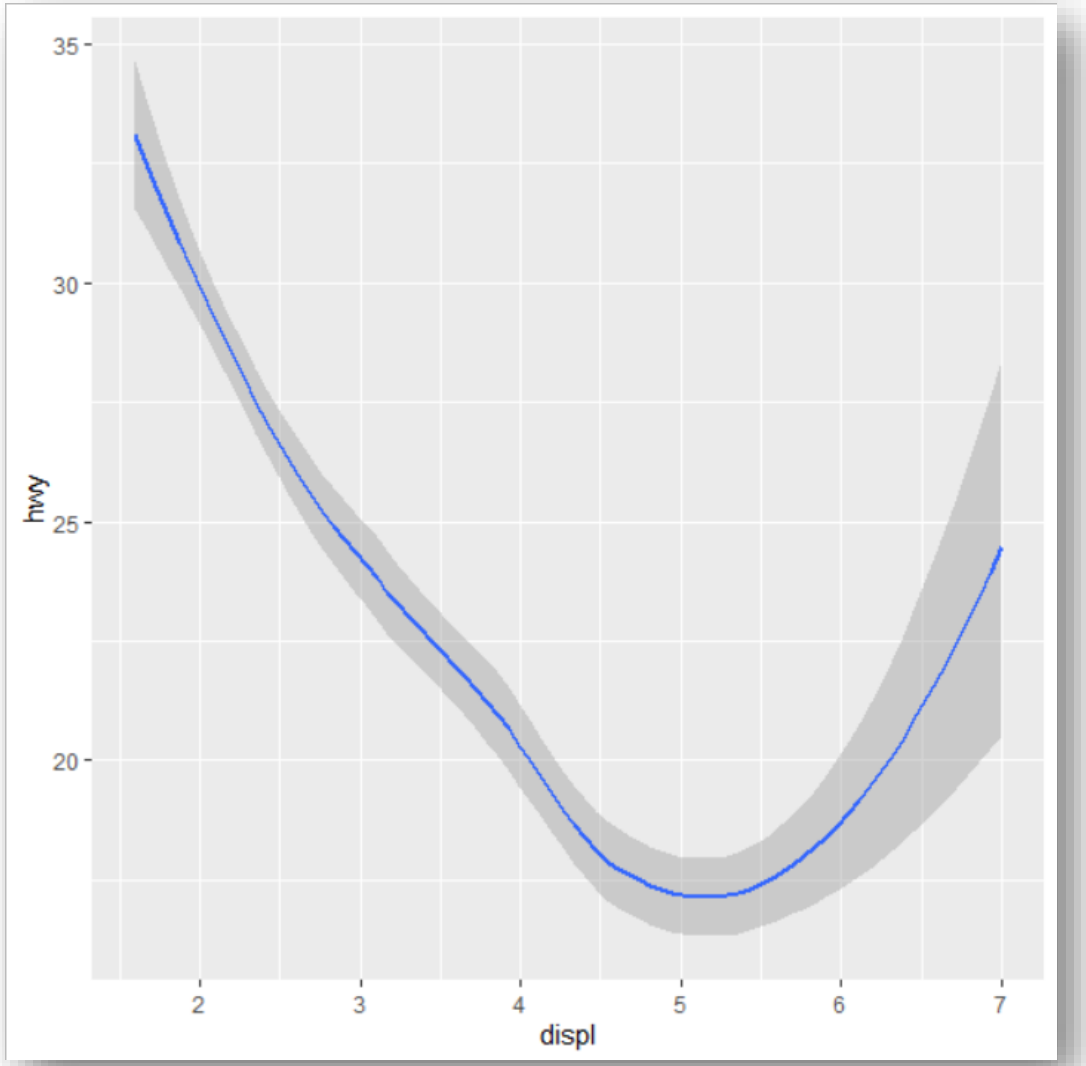
B. Changing the data

C. Changing the geometry

D. Changing one of the aestetics

00:15

```
ggplot(data = mpg) +
    geom_smooth(mapping = aes(x = displ, y = hwy))
```



Your Turn

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy))
```

How can I **add a smoother** on top of the points?

Your Turn

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy))
```

How can I **add a smoother** on top of the points?

A. Adding one mapping variable

B. Adding one more data set

C. Adding one geometry

D. Adding one aestetic

00:15

Your Turn

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy))
```

How can I **add a smoother** on top of the points?

A. Adding one mapping variable

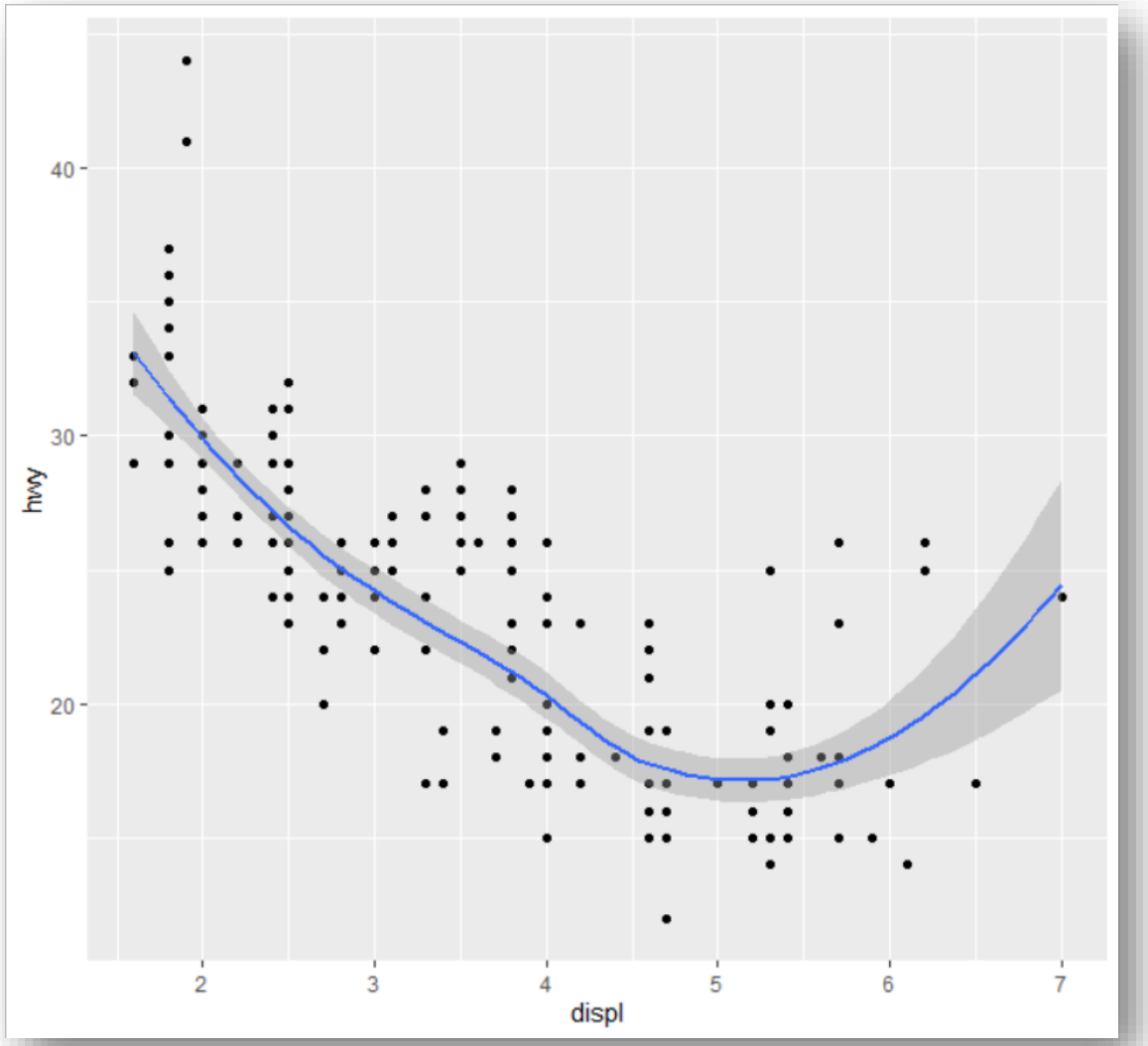
B. Adding one more data set

C. Adding one geometry

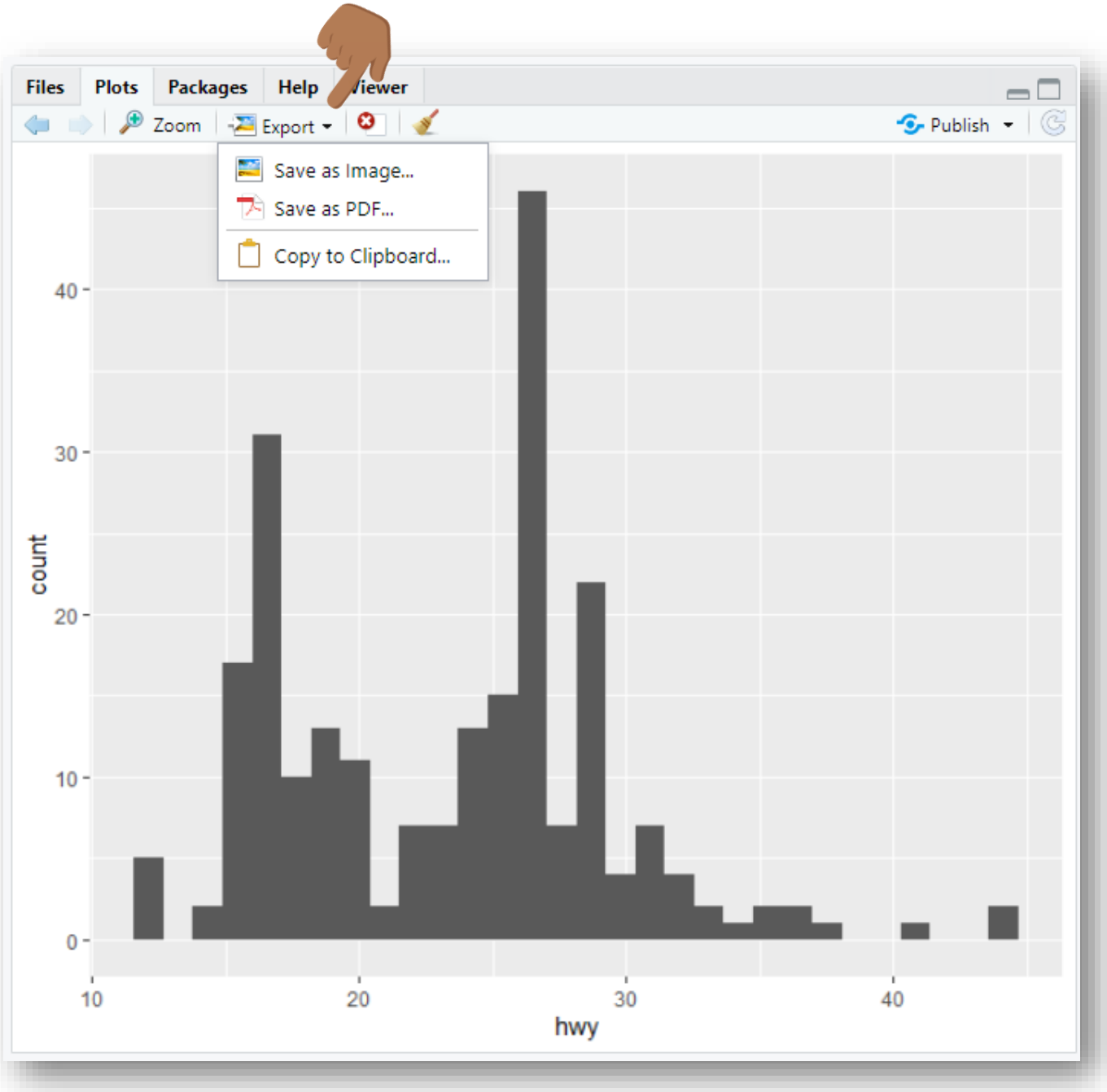
D. Adding one aestetic

00:15

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  geom_smooth(mapping = aes(x = displ, y = hwy))
```



Saving your plots



Tip: `?ggsave()`

Questions

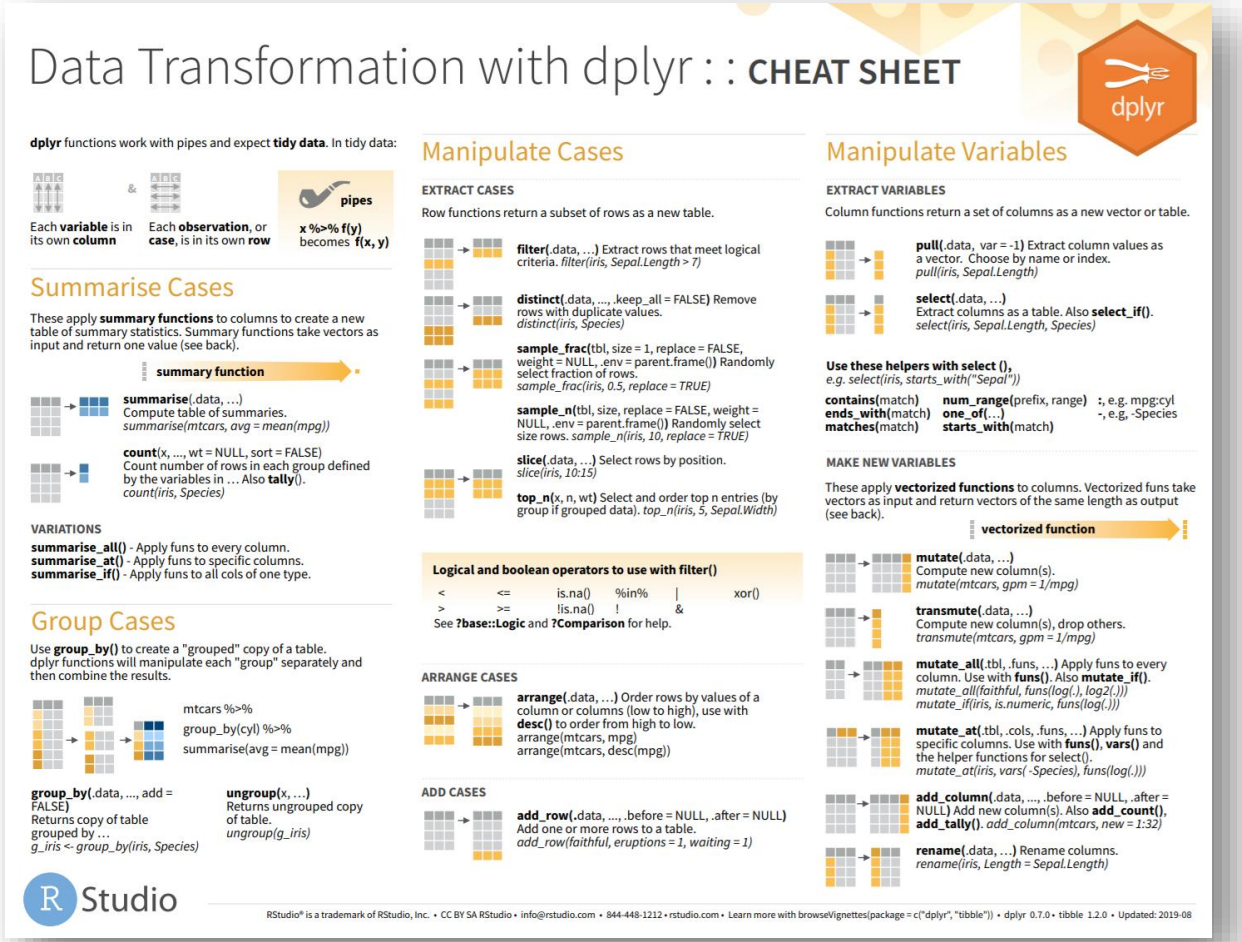
?

data manipulation with `dplyr`

A package for data transformation.

It provides six simple "verbs" corresponding to the most common data manipulation tasks.

Data transformation cheat sheet



Data Transformation with dplyr : : CHEAT SHEET

dplyr functions work with pipes and expect **tidy data**. In tidy data:

Each **variable** is in its own **column**

Each **observation**, or **case**, is in its own **row**

pipes

x %>% f(y) becomes **f(x, y)**

Summarise Cases

These apply **summary functions** to columns to create a new table of summary statistics. Summary functions take vectors as input and return one value (see back).

summary function

summarise(.data, …)
Compute table of summaries.
summarise(mtcars, avg = mean(mpg))

count(x, ..., wt = NULL, sort = FALSE)
Count number of rows in each group defined by the variables in … Also **tally**().
count(iris, Species)

VARIATIONS

summarise_all() - Apply funs to every column.
summarise_at() - Apply funs to specific columns.
summarise_if() - Apply funs to all cols of one type.

Group Cases

Use **group_by()** to create a "grouped" copy of a table. dplyr functions will manipulate each "group" separately and then combine the results.

mtcars %>%
group_by(cyl) %>%
summarise(avg = mean(mpg))

group_by(.data, ..., add = FALSE)
Returns copy of table grouped by …
g_iris <- group_by(iris, Species)

ungroup(x, …)
Returns ungrouped copy of table.
ungroup(g_iris)

Manipulate Cases

EXTRACT CASES

Row functions return a subset of rows as a new table.

filter(.data, …) Extract rows that meet logical criteria. *filter(iris, Sepal.Length > 7)*

distinct(.data, ..., .keep_all = FALSE) Remove rows with duplicate values.
distinct(iris, Species)

sample_frac(tbl, size = 1, replace = FALSE, weight = NULL, .env = parent.frame()) Randomly select fraction of rows.
sample_frac(iris, 0.5, replace = TRUE)

sample_n(tbl, size, replace = FALSE, weight = NULL, .env = parent.frame()) Randomly select size rows. *sample_n(iris, 10, replace = TRUE)*

slice(.data, …) Select rows by position.
slice(iris, 10:15)

top_n(x, n, wt) Select and order top n entries (by group if grouped data). *top_n(iris, 5, Sepal.Width)*

Logical and boolean operators to use with filter()

<	<=	is.na()	%in%	| xor()
>	>=	!is.na()	!	&

See **?base::Logic** and **?Comparison** for help.

ARRANGE CASES

arrange(.data, …) Order rows by values of a column or columns (low to high), use with **desc()** to order from high to low.
arrange(mtcars, mpg)
arrange(mtcars, desc(mpg))

ADD CASES

add_row(.data, ..., .before = NULL, .after = NULL) Add one or more rows to a table.
add_row(faithful, eruptions = 1, waiting = 1)

Manipulate Variables

EXTRACT VARIABLES

Column functions return a set of columns as a new vector or table.

pull(.data, var = -1) Extract column values as a vector. Choose by name or index.
pull(iris, Sepal.Length)

select(.data, …)
Extract columns as a table. Also **select_if()**.
select(iris, Sepal.Length, Species)

Use these helpers with select (),
e.g. select(iris, starts_with("Sepal"))

contains(match) **num_range**(prefix, range) **:**, e.g. mpg:cyl
ends_with(match) **one_of**(…) **-**, e.g. -Species
matches(match) **starts_with**(match)

MAKE NEW VARIABLES

These apply **vectorized functions** to columns. Vectorized funs take vectors as input and return vectors of the same length as output (see back).

vectorized function

mutate(.data, …)
Compute new column(s).
mutate(mtcars, gpm = 1/mpg)

transmute(.data, …)
Compute new column(s), drop others.
transmute(mtcars, gpm = 1/mpg)

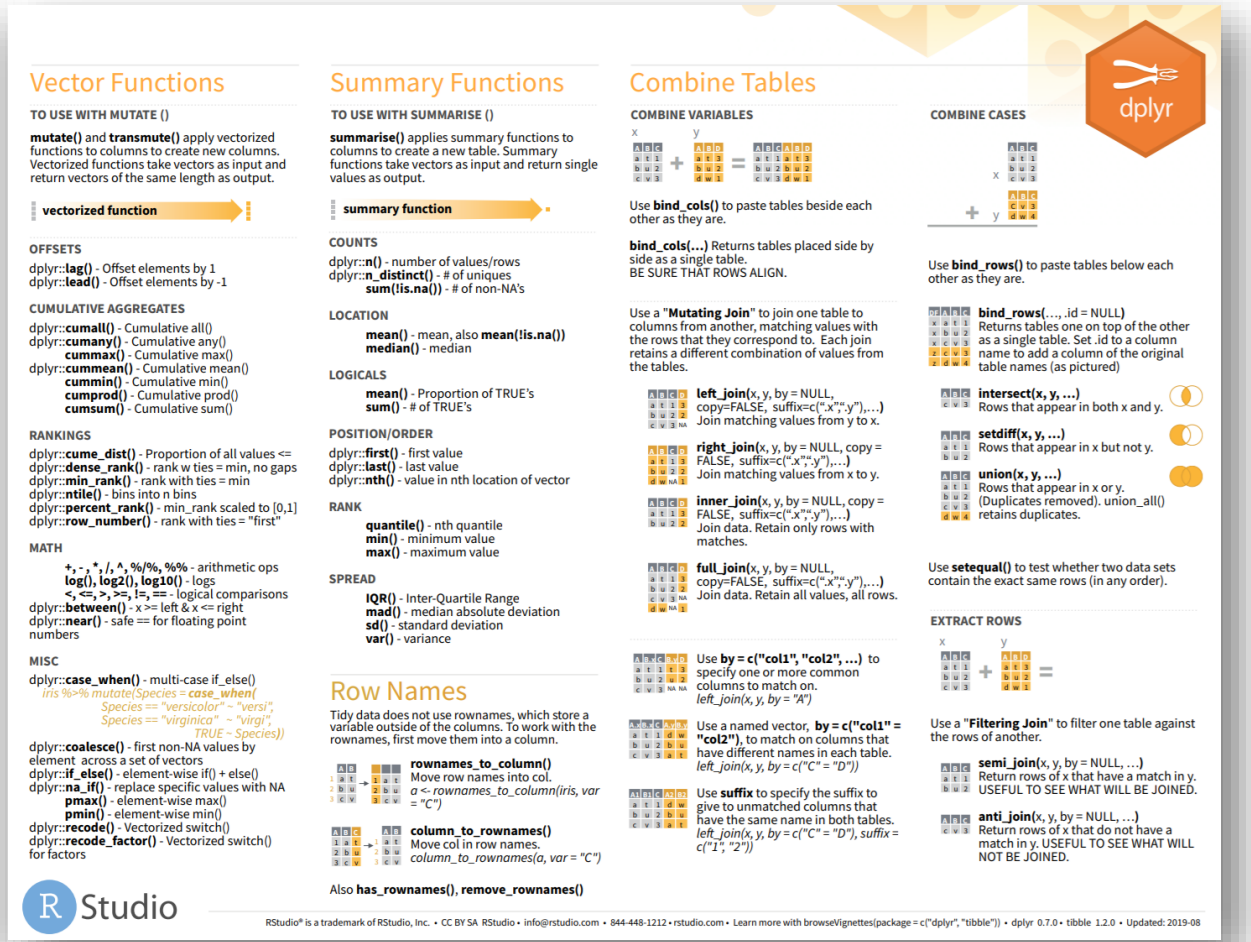
mutate_all(.tbl, .funs, …) Apply funs to every column. Use with **funs()**. Also **mutate_if()**.
mutate_all(faithful, funs(log(.), log2(.)))
mutate_if(iris, is.numeric, funs(log(.)))

mutate_at(.tbl, .cols, .funs, …) Apply funs to specific columns. Use with **funs()**, **vars()** and the helper functions for select().
mutate_at(iris, vars(-Species), funs(log(.)))

add_column(.data, ..., .before = NULL, .after = NULL) Add new column(s). Also **add_count()**, **add_tally()**. *add_column(mtcars, new = 1:32)*

rename(.data, …) Rename columns.
rename(iris, Length = Sepal.Length)

RStudio® is a trademark of RStudio, Inc. • CC BY SA RStudio • info@rstudio.com • 844-448-1212 • rstudio.com • Learn more with browseVignettes(package = c("dplyr", "tibble")) • dplyr 0.7.0 • tibble 1.2.0 • Updated: 2019-08



Vector Functions

TO USE WITH MUTATE ()

mutate() and **transmute()** apply vectorized functions to columns to create new columns. Vectorized functions take vectors as input and return vectors of the same length as output.

vectorized function

OFFSETS

dplyr::**lag()** - Offset elements by 1
dplyr::**lead()** - Offset elements by -1

CUMULATIVE AGGREGATES

dplyr::**cumall()** - Cumulative all()
dplyr::**cumany()** - Cumulative any()
cummax() - Cumulative max()
dplyr::**cummean()** - Cumulative mean()
cummin() - Cumulative min()
cumprod() - Cumulative prod()
cumsum() - Cumulative sum()

RANKINGS

dplyr::**cume_dist()** - Proportion of all values <=
dplyr::**dense_rank()** - rank w ties = min, no gaps
dplyr::**min_rank()** - rank with ties = min
dplyr::**ntile()** - bins into n bins
dplyr::**percent_rank()** - min_rank scaled to [0,1]
dplyr::**row_number()** - rank with ties = "first"

MATH

+, - , *, /, ^, %/%, %% - arithmetic ops
log(), log2(), log10() - logs
<, <=, >, >=, !=, == - logical comparisons
dplyr::**between()** - x >= left & x <= right
dplyr::**near()** - safe == for floating point numbers

MISC

dplyr::**case_when()** - multi-case if_else()
iris %>% mutate(Species = case_when(
Species == "versicolor" ~ "versi",
Species == "virginica" ~ "virgi",
TRUE ~ Species))
dplyr::**coalesce()** - first non-NA values by element across a set of vectors
dplyr::**if_else()** - element-wise if() + else()
dplyr::**na_if()** - replace specific values with NA
pmax() - element-wise max()
pmin() - element-wise min()
dplyr::**recode()** - Vectorized switch()
dplyr::**recode_factor()** - Vectorized switch() for factors

Summary Functions

TO USE WITH SUMMARISE ()

summarise() applies summary functions to columns to create a new table. Summary functions take vectors as input and return single values as output.

summary function

COUNTS

dplyr::**n()** - number of values/rows
dplyr::**n_distinct()** - # of uniques
sum(!is.na()) - # of non-NA's

LOCATION

mean() - mean, also **mean(!is.na())**
median() - median

LOGICALS

mean() - Proportion of TRUE's
sum() - # of TRUE's

POSITION/ORDER

dplyr::**first()** - first value
dplyr::**last()** - last value
dplyr::**nth()** - value in nth location of vector

RANK

quantile() - nth quantile
min() - minimum value
max() - maximum value

SPREAD

IQR() - Inter-Quartile Range
mad() - median absolute deviation
sd() - standard deviation
var() - variance

Row Names

Tidy data does not use rownames, which store a variable outside of the columns. To work with the rownames, first move them into a column.

rownames_to_column()
Move row names into col.
a <- rownames_to_column(iris, var = "C")

column_to_rownames()
Move col in row names.
column_to_rownames(a, var = "C")

Also **has_rownames()**, **remove_rownames()**

Combine Tables

COMBINE VARIABLES

Use **bind_cols()** to paste tables beside each other as they are.

bind_cols(…) Returns tables placed side by side as a single table.
BE SURE THAT ROWS ALIGN.

Use a "**Mutating Join**" to join one table to columns from another, matching values with the rows that they correspond to. Each join retains a different combination of values from the tables.

left_join(x, y, by = NULL, copy=FALSE, suffix=c(".x",".y"),…)
Join matching values from y to x.

right_join(x, y, by = NULL, copy = FALSE, suffix=c(".x",".y"),…)
Join matching values from x to y.

inner_join(x, y, by = NULL, copy = FALSE, suffix=c(".x",".y"),…)
Join data. Retain only rows with matches.

full_join(x, y, by = NULL, copy=FALSE, suffix=c(".x",".y"),…)
Join data. Retain all values, all rows.

Use **by = c("col1", "col2", …)** to specify one or more common columns to match on.
left_join(x, y, by = "A")

Use a named vector, **by = c("col1" = "col2")**, to match on columns that have different names in each table.
left_join(x, y, by = c("C" = "D"))

Use **suffix** to specify the suffix to give to unmatched columns that have the same name in both tables.
left_join(x, y, by = c("C" = "D"), suffix = c("1", "2"))

COMBINE CASES

Use **bind_rows()** to paste tables below each other as they are.

bind_rows(…, .id = NULL)
Returns tables one on top of the other as a single table. Set .id to a column name to add a column of the original table names (as pictured)

intersect(x, y, …)
Rows that appear in both x and y.

setdiff(x, y, …)
Rows that appear in x but not y.

union(x, y, …)
Rows that appear in x or y. (Duplicates removed). union_all() retains duplicates.

Use **setequal()** to test whether two data sets contain the exact same rows (in any order).

EXTRACT ROWS

Use a "**Filtering Join**" to filter one table against the rows of another.

semi_join(x, y, by = NULL, …)
Return rows of x that have a match in y. USEFUL TO SEE WHAT WILL BE JOINED.

anti_join(x, y, by = NULL, …)
Return rows of x that do not have a match in y. USEFUL TO SEE WHAT WILL NOT BE JOINED.

RStudio® is a trademark of RStudio, Inc. • CC BY SA RStudio • info@rstudio.com • 844-448-1212 • rstudio.com • Learn more with browseVignettes(package = c("dplyr", "tibble")) • dplyr 0.7.0 • tibble 1.2.0 • Updated: 2019-08

`dplyr` general syntax

1. Call any `dplyr::function()`
2. First argument is a **data frame (tibble)**
3. Arguments describing what to do using

   variable names (no quotes)
4. Result is a **data frame (tibble)**


```
function(data, ...)
```

data

function-specific argument

mpg data

Fuel economy data for 38 models of cars

```
View(mpg)
```

mpg

Filter

	manufacturer	model	displ	year	cyl	trans	drv	cty	hwy	fl	class
1	audi	a4	1.8	1999	4	auto(l5)	f	18	29	p	compact
2	audi	a4	1.8	1999	4	manual(m5)	f	21	29	p	compact
3	audi	a4	2.0	2008	4	manual(m6)	f	20	31	p	compact
4	audi	a4	2.0	2008	4	auto(av)	f	21	30	p	compact
5	audi	a4	2.8	1999	6	auto(l5)	f	16	26	p	compact
6	audi	a4	2.8	1999	6	manual(m5)	f	18	26	p	compact
7	audi	a4	3.1	2008	6	auto(av)	f	18	27	p	compact
8	audi	a4 quattro	1.8	1999	4	manual(m5)	4	18	26	p	compact
9	audi	a4 quattro	1.8	1999	4	auto(l5)	4	16	25	p	compact
10	audi	a4 quattro	2.0	2008	4	manual(m6)	4	20	28	p	compact

Your Turn

mpg

Filter

	manufacturer	model	displ	year	cyl	trans	drv	cty	hwy	fl	class
1	audi	a4	1.8	1999	4	auto(l5)	f	18	29	p	compact
2	audi	a4	1.8	1999	4	manual(m5)	f	21	29	p	compact
3	audi	a4	2.0	2008	4	manual(m6)	f	20	31	p	compact
4	audi	a4	2.0	2008	4	auto(av)	f	21	30	p	compact
5	audi	a4	2.8	1999	6	auto(l5)	f	16	26	p	compact
6	audi	a4	2.8	1999	6	manual(m5)	f	18	26	p	compact
7	audi	a4	3.1	2008	6	auto(av)	f	18	27	p	compact
8	audi	a4 quattro	1.8	1999	4	manual(m5)	4	18	26	p	compact
9	audi	a4 quattro	1.8	1999	4	auto(l5)	4	16	25	p	compact
10	audi	a4 quattro	2.0	2008	4	manual(m6)	4	20	28	p	compact

How do you say that you want to **keep only variables** **`displ`** and **`hwy`**?

Your Turn

mpg

Filter

	manufacturer	model	displ	year	cyl	trans	drv	cty	hwy	fl	class
1	audi	a4	1.8	1999	4	auto(l5)	f	18	29	p	compact
2	audi	a4	1.8	1999	4	manual(m5)	f	21	29	p	compact
3	audi	a4	2.0	2008	4	manual(m6)	f	20	31	p	compact
4	audi	a4	2.0	2008	4	auto(av)	f	21	30	p	compact
5	audi	a4	2.8	1999	6	auto(l5)	f	16	26	p	compact
6	audi	a4	2.8	1999	6	manual(m5)	f	18	26	p	compact
7	audi	a4	3.1	2008	6	auto(av)	f	18	27	p	compact
8	audi	a4 quattro	1.8	1999	4	manual(m5)	4	18	26	p	compact
9	audi	a4 quattro	1.8	1999	4	auto(l5)	4	16	25	p	compact
10	audi	a4 quattro	2.0	2008	4	manual(m6)	4	20	28	p	compact

How do you say that you want to **keep only variables** `displ` and `hwy`?

A `filter`

B `mutate`

C `arrange`

D `select`

00:15

Your Turn

mpg

Filter

	manufacturer	model	displ	year	cyl	trans	drv	cty	hwy	fl	class
1	audi	a4	1.8	1999	4	auto(l5)	f	18	29	p	compact
2	audi	a4	1.8	1999	4	manual(m5)	f	21	29	p	compact
3	audi	a4	2.0	2008	4	manual(m6)	f	20	31	p	compact
4	audi	a4	2.0	2008	4	auto(av)	f	21	30	p	compact
5	audi	a4	2.8	1999	6	auto(l5)	f	16	26	p	compact
6	audi	a4	2.8	1999	6	manual(m5)	f	18	26	p	compact
7	audi	a4	3.1	2008	6	auto(av)	f	18	27	p	compact
8	audi	a4 quattro	1.8	1999	4	manual(m5)	4	18	26	p	compact
9	audi	a4 quattro	1.8	1999	4	auto(l5)	4	16	25	p	compact
10	audi	a4 quattro	2.0	2008	4	manual(m6)	4	20	28	p	compact

How do you say that we want to **keep only variables** `displ` and `hwy`?

A `filter`

B `mutate`

C `arrange`

D `select`

mpg

Filter

	manufacturer	model	displ	year	cyl	trans	drv	cty	hwy	fl	class
1	audi	a4	1.8	1999	4	auto(l5)	f	18	29	p	compact
2	audi	a4	1.8	1999	4	manual(m5)	f	21	29	p	compact
3	audi	a4	2.0	2008	4	manual(m6)	f	20	31	p	compact
4	audi	a4	2.0	2008	4	auto(av)	f	21	30	p	compact
5	audi	a4	2.8	1999	6	auto(l5)	f	16	26	p	compact
6	audi	a4	2.8	1999	6	manual(m5)	f	18	26	p	compact
7	audi	a4	3.1	2008	6	auto(av)	f	18	27	p	compact
8	audi	a4 quattro	1.8	1999	4	manual(m5)	4	18	26	p	compact
9	audi	a4 quattro	1.8	1999	4	auto(l5)	4	16	25	p	compact
10	audi	a4 quattro	2.0	2008	4	manual(m6)	4	20	28	p	compact

```
mpg_rec <- select(mpg, displ, hwy)
```

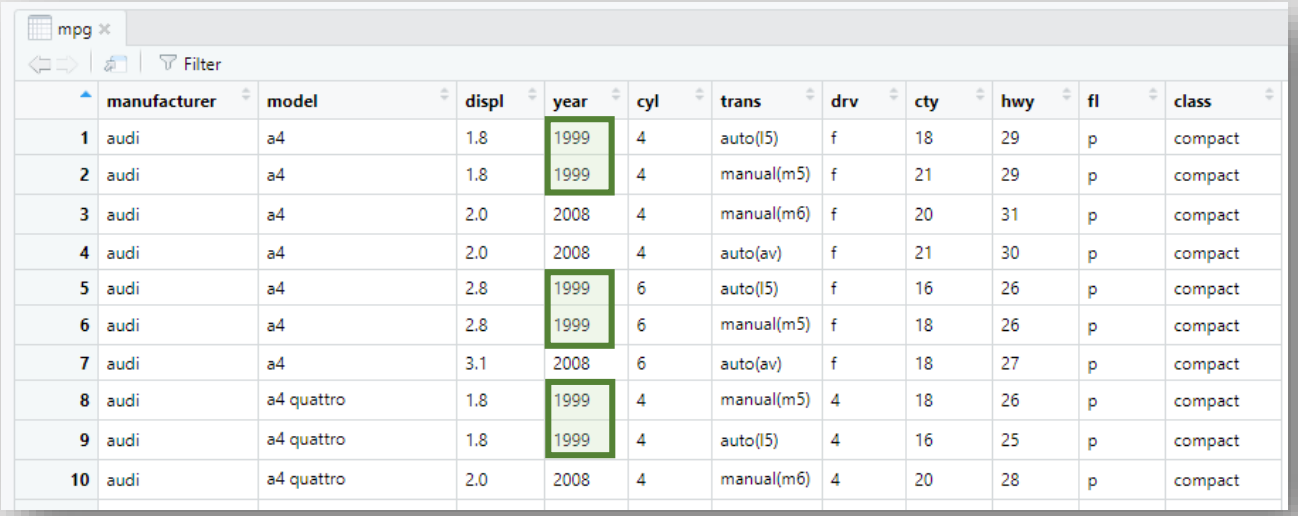
new data **data** **variables**

mpg_rec

Filter

	displ	hwy
1	1.8	29
2	1.8	29
3	2.0	31
4	2.0	30
5	2.8	26
6	2.8	26
7	3.1	27
8	1.8	26
9	1.8	25
10	2.0	28

Your Turn



mpg

Filter

	manufacturer	model	displ	year	cyl	trans	drv	cty	hwy	fl	class
1	audi	a4	1.8	1999	4	auto(l5)	f	18	29	p	compact
2	audi	a4	1.8	1999	4	manual(m5)	f	21	29	p	compact
3	audi	a4	2.0	2008	4	manual(m6)	f	20	31	p	compact
4	audi	a4	2.0	2008	4	auto(av)	f	21	30	p	compact
5	audi	a4	2.8	1999	6	auto(l5)	f	16	26	p	compact
6	audi	a4	2.8	1999	6	manual(m5)	f	18	26	p	compact
7	audi	a4	3.1	2008	6	auto(av)	f	18	27	p	compact
8	audi	a4 quattro	1.8	1999	4	manual(m5)	4	18	26	p	compact
9	audi	a4 quattro	1.8	1999	4	auto(l5)	4	16	25	p	compact
10	audi	a4 quattro	2.0	2008	4	manual(m6)	4	20	28	p	compact

How do you say that you want to **keep only certain observations**?
(e.g. cars from 1999)

Your Turn

mpg

Filter

	manufacturer	model	displ	year	cyl	trans	drv	cty	hwy	fl	class
1	audi	a4	1.8	1999	4	auto(l5)	f	18	29	p	compact
2	audi	a4	1.8	1999	4	manual(m5)	f	21	29	p	compact
3	audi	a4	2.0	2008	4	manual(m6)	f	20	31	p	compact
4	audi	a4	2.0	2008	4	auto(av)	f	21	30	p	compact
5	audi	a4	2.8	1999	6	auto(l5)	f	16	26	p	compact
6	audi	a4	2.8	1999	6	manual(m5)	f	18	26	p	compact
7	audi	a4	3.1	2008	6	auto(av)	f	18	27	p	compact
8	audi	a4 quattro	1.8	1999	4	manual(m5)	4	18	26	p	compact
9	audi	a4 quattro	1.8	1999	4	auto(l5)	4	16	25	p	compact
10	audi	a4 quattro	2.0	2008	4	manual(m6)	4	20	28	p	compact

How do you say that you want to **keep only certain observations**? (e.g. cars from 1999)

A `filter`

B `mutate`

C `arrange`

D `select`

00:15

Your Turn

mpg

Filter

	manufacturer	model	displ	year	cyl	trans	drv	cty	hwy	fl	class
1	audi	a4	1.8	1999	4	auto(l5)	f	18	29	p	compact
2	audi	a4	1.8	1999	4	manual(m5)	f	21	29	p	compact
3	audi	a4	2.0	2008	4	manual(m6)	f	20	31	p	compact
4	audi	a4	2.0	2008	4	auto(av)	f	21	30	p	compact
5	audi	a4	2.8	1999	6	auto(l5)	f	16	26	p	compact
6	audi	a4	2.8	1999	6	manual(m5)	f	18	26	p	compact
7	audi	a4	3.1	2008	6	auto(av)	f	18	27	p	compact
8	audi	a4 quattro	1.8	1999	4	manual(m5)	4	18	26	p	compact
9	audi	a4 quattro	1.8	1999	4	auto(l5)	4	16	25	p	compact
10	audi	a4 quattro	2.0	2008	4	manual(m6)	4	20	28	p	compact

How do you say that you want to **keep only certain observations**? (e.g. cars from 1999)

A `filter`

B `mutate`

C `arrange`

D `select`

00:15

mpg

Filter

	manufacturer	model	displ	year	cyl	trans	drv	cty	hwy	fl	class
1	audi	a4	1.8	1999	4	auto(l5)	f	18	29	p	compact
2	audi	a4	1.8	1999	4	manual(m5)	f	21	29	p	compact
3	audi	a4	2.0	2008	4	manual(m6)	f	20	31	p	compact
4	audi	a4	2.0	2008	4	auto(av)	f	21	30	p	compact
5	audi	a4	2.8	1999	6	auto(l5)	f	16	26	p	compact
6	audi	a4	2.8	1999	6	manual(m5)	f	18	26	p	compact
7	audi	a4	3.1	2008	6	auto(av)	f	18	27	p	compact
8	audi	a4 quattro	1.8	1999	4	manual(m5)	4	18	26	p	compact
9	audi	a4 quattro	1.8	1999	4	auto(l5)	4	16	25	p	compact
10	audi	a4 quattro	2.0	2008	4	manual(m6)	4	20	28	p	compact

```
mpg_rec <- filter(mpg, year==1999)
```

new data ↑ **data** ↑ **filtering condition** ↑

mpg_rec

Filter

	manufacturer	model	displ	year	cyl	trans	drv	cty	hwy	fl	class
1	audi	a4	1.8	1999	4	auto(l5)	f	18	29	p	compact
2	audi	a4	1.8	1999	4	manual(m5)	f	21	29	p	compact
3	audi	a4	2.8	1999	6	auto(l5)	f	16	26	p	compact
4	audi	a4	2.8	1999	6	manual(m5)	f	18	26	p	compact
5	audi	a4 quattro	1.8	1999	4	manual(m5)	4	18	26	p	compact
6	audi	a4 quattro	1.8	1999	4	auto(l5)	4	16	25	p	compact
7	audi	a4 quattro	2.8	1999	6	auto(l5)	4	15	25	p	compact
8	audi	a4 quattro	2.8	1999	6	manual(m5)	4	17	25	p	compact
9	audi	a6 quattro	2.8	1999	6	auto(l5)	4	15	24	p	midsize
10	chevrolet	c1500 suburban 2wd	5.7	1999	8	auto(l4)	r	13	17	r	suv

Questions

?

Your Turn

What about **`mutate()`** and **`arrange()`**?

Check out the code in **`` `01_introduction.R` ``** and try to find it out.

Your Turn

Open the file `02_markdown_piping.R` to learn about R Markdown and piping.

Introduction to tidyverse

https://www.meetup.com/Lucerne-R-User-Group/

https://github.com/Lucerne-R-User-Group/2020_06_24-inaugural-meeting

Dr Andrea De Angelis

24 June 2020