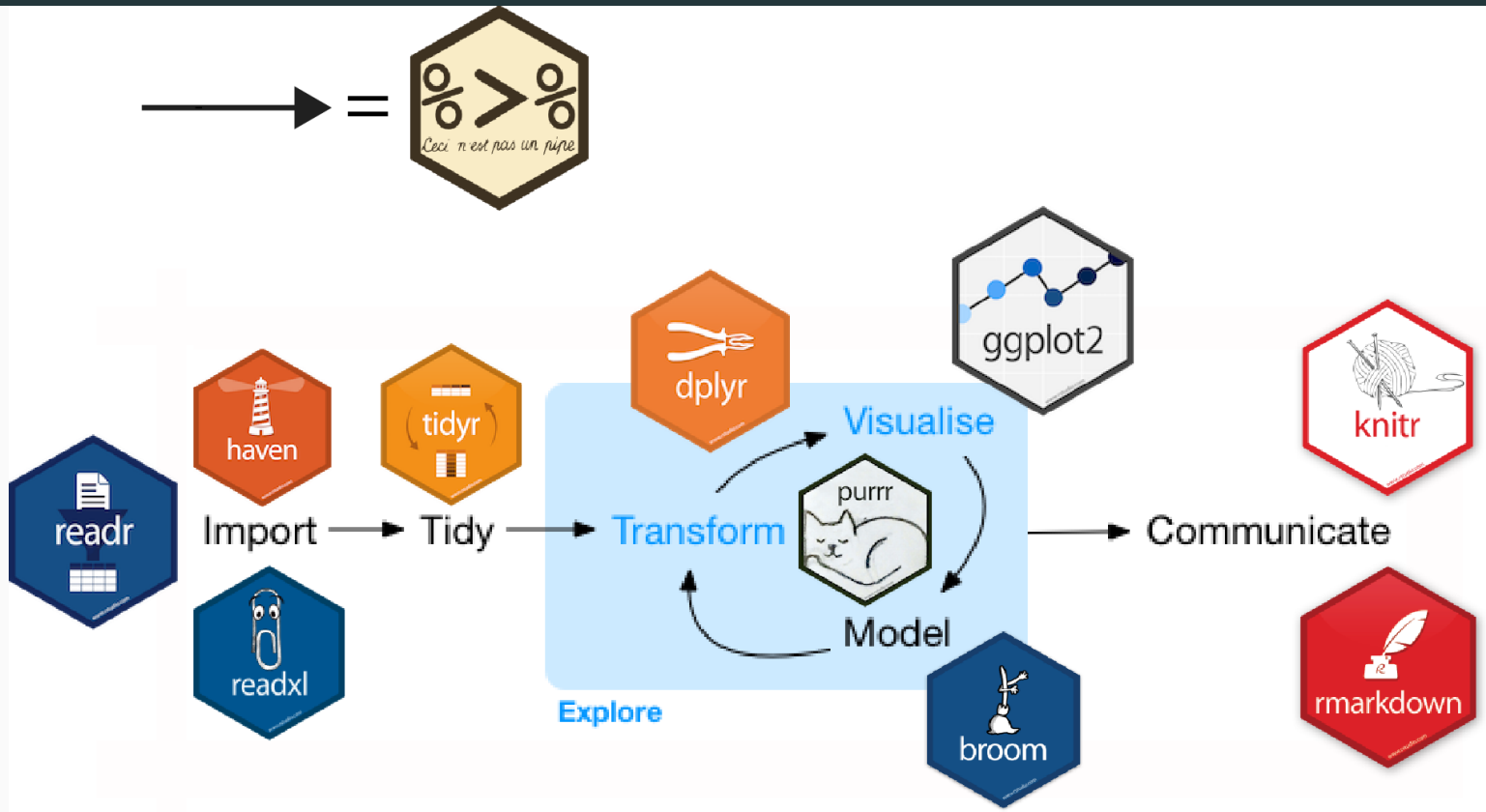
Introduction to data wrangling and tidy data

Stephanie J. Spielman
CB2R Data Science Workshop, Summer 2020

Enter the tidyverse

Enter the tidyverse



data.frames

factors

strings

We will be ~~learning~~ starting to learn...

- `dplyr` ("dee-plier") for manipulating ("wrangling") data
- `tidyr` ("tidy-are") for cleaning and *tidying* data
- `ggplot2` ("gee-gee-plot") for data visualization
- `tibble` ("tibble") for making data frames more enjoyable to work with (incidentally)
- `readr` ("read-are") and `readxl` ("read-excel") for reading and writing data
- `rmarkdown` ("are-markdown") for producing professional documents

The philosophy of tidy data

"Messy" vs tidy data

messy

	treatmenta	treatmentb
John Smith	—	2
Jane Doe	16	11
Mary Johnson	3	1

	John Smith	Jane Doe	Mary Johnson
treatmenta	—	16	3
treatmentb	2	11	1

tidy

name	trt	result
John Smith	a	—
Jane Doe	a	16
Mary Johnson	a	3
John Smith	b	2
Jane Doe	b	11
Mary Johnson	b	1

Which version is tidy?

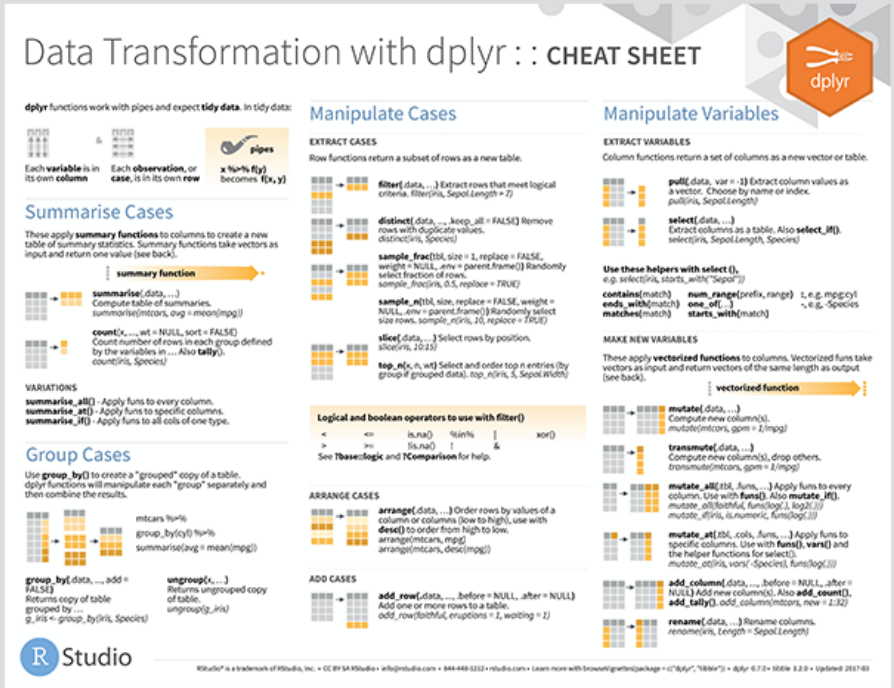
First, identify what the *variables* are in the dataset (the *data types* can help us!!)

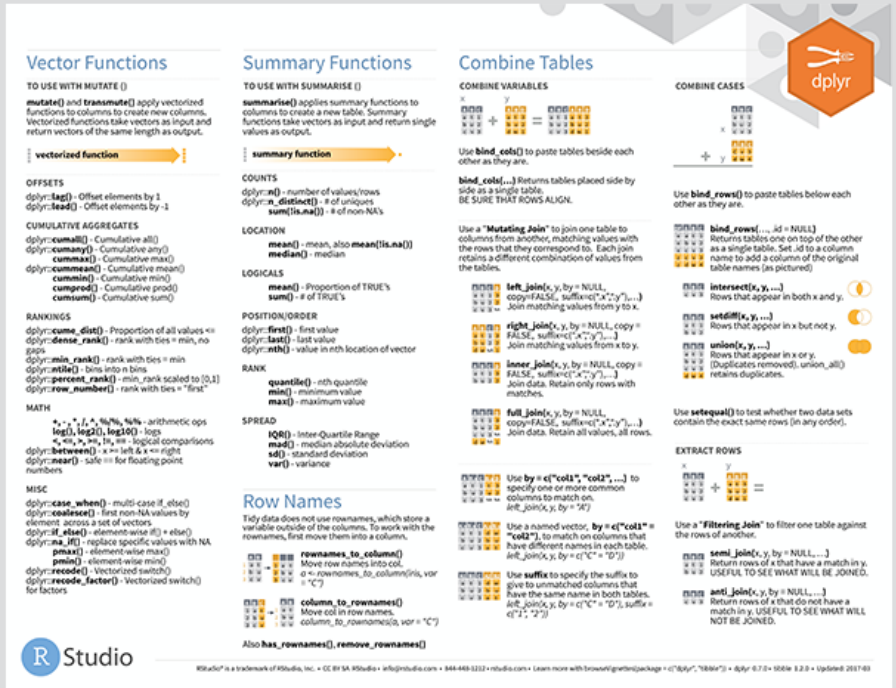
```
## # A tibble: 5 x 3
##    seal oxygen_use_nonfeeding oxygen_use_feeding
##   <dbl>                 <dbl>               <dbl>
## 1     1                  42.2                71
## 2     2                  51.7                77.3
## 3     3                  82.8                78.6
## 4     4                  66.5                96.1
## 5     5                  81.9               107.
```

```
## # A tibble: 10 x 3
##     seal experiment_type oxygen_use
##    <dbl> <chr>                <dbl>
##  1     1 nonfeeding            42.2
##  2     1 feeding               71
##  3     2 nonfeeding            51.7
##  4     2 feeding               77.3
##  5     3 nonfeeding            82.8
##  6     3 feeding               78.6
##  7     4 nonfeeding            66.5
##  8     4 feeding               96.1
##  9     5 nonfeeding            81.9
## 10     5 feeding              107.
```

Data wrangling with dplyr

I do not know all of the CHEATSHEET (let alone the whole package!). But, I *know about* the cheatsheet.





Data wrangling with dplyr

The greatest hits of `dplyr` verbs

This lesson

- `glimpse()` as an alternative to `str()` or `head()`
- `select()` subsets **columns**
- `filter()` subsets **rows**
- `mutate()` adds new columns
- `rename()` changes the names of columns
- `arrange()` changes the order of rows

Future lessons

- `group_by()` and `ungroup()`
- `summarize()`
- `tally()`
- `count()`

Time-pending lessons

- `bind_rows()`
- `bind_cols()`
- `left_join()` and friends

The great and powerful **pipe**

Examples of piping, and formatting goals

```
## tired
log(5)
```

```
## [1] 1.609438
```

```
## Wired
5 %>% log()
```

```
## [1] 1.609438
```

```
## ON FIRE
5 %>%
  log()
```

```
## [1] 1.609438
```

```
## KILN-LEVEL FIRE
5 %>%
  log(10)
```

```
## [1] 0.69897
```

Cold tolerance experiment on grass species

```
head(plant)
```

```
## # A tibble: 6 x 5
##   plant_id origin treatment  co2_conc co2_rate
##   <ord>    <fct>  <fct>         <dbl>    <dbl>
## 1 Qn1      Quebec nonchilled       95     16
## 2 Qn1      Quebec nonchilled      175     30.4
## 3 Qn1      Quebec nonchilled      250     34.8
## 4 Qn1      Quebec nonchilled      350     37.2
## 5 Qn1      Quebec nonchilled      500     35.3
## 6 Qn1      Quebec nonchilled      675     39.2
```

```
nrow(plant)
```

```
## [1] 84
```

```
ncol(plant)
```

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

Loading libraries

```
## LOAD THE DPLYR LIBRARY
library(dplyr) # must load for each R session
```

```
## your first dplyr function!
glimpse(plant)
```

```
## Rows: 84
## Columns: 5
## $ plant_id  <ord> Qn1, Qn1, Qn1, Qn1, Qn1, Qn1, Qn1, Qn2, Qn2, Qn2, Qn2, Qn2,…
## $ origin    <fct> Quebec, Quebec, Quebec, Quebec, Quebec, Quebec, Quebec, Que…
## $ treatment <fct> nonchilled, nonchilled, nonchilled, nonchilled, nonchilled,…
## $ co2_conc  <dbl> 95, 175, 250, 350, 500, 675, 1000, 95, 175, 250, 350, 500, …
## $ co2_rate  <dbl> 16.0, 30.4, 34.8, 37.2, 35.3, 39.2, 39.7, 13.6, 27.3, 37.1,…
```

First, ALWAYS LOOK AT YOUR DATA

```
levels(plant$plant_id)
```

```
##  [1] "Qn1" "Qn2" "Qn3" "Qc1" "Qc3" "Qc2" "Mn3" "Mn2" "Mn1" "Mc2" "Mc3" "Mc1"
```

```
levels(plant$origin)
```

```
## [1] "Quebec"      "Mississippi"
```

```
levels(plant$treatment)
```

```
## [1] "nonchilled" "chilled"
```

Subsetting columns with select

```
names(plant)
```

```
## [1] "plant_id"  "origin"    "treatment" "co2_conc"  "co2_rate"
```

Select `plant_id` and `origin`

```
plant %>%
  select(plant_id, origin)
```

```
## # A tibble: 84 x 2
##    plant_id origin
##    <ord>    <fct>
##  1 Qn1      Quebec
##  2 Qn1      Quebec
##  3 Qn1      Quebec
##  4 Qn1      Quebec
##  5 Qn1      Quebec
##  6 Qn1      Quebec
##  7 Qn1      Quebec
##  8 Qn2      Quebec
##  9 Qn2      Quebec
## 10 Qn2      Quebec
```

Select all columns *except* `treatment`

```
plant %>%
  select(-treatment)
```

```
## # A tibble: 84 x 4
##    plant_id origin co2_conc co2_rate
##    <ord>    <fct>     <dbl>    <dbl>
##  1 Qn1      Quebec       95     16
##  2 Qn1      Quebec      175     30.4
##  3 Qn1      Quebec      250     34.8
##  4 Qn1      Quebec      350     37.2
##  5 Qn1      Quebec      500     35.3
##  6 Qn1      Quebec      675     39.2
##  7 Qn1      Quebec     1000     39.7
##  8 Qn2      Quebec       95     13.6
##  9 Qn2      Quebec      175     27.3
## 10 Qn2      Quebec      250     37.1
```

The power of the pipe

```
plant %>%
  select(plant_id, origin)
```

```
## # A tibble: 84 x 2
##    plant_id origin
##    <ord>    <fct>
##  1 Qn1      Quebec
##  2 Qn1      Quebec
##  3 Qn1      Quebec
##  4 Qn1      Quebec
##  5 Qn1      Quebec
##  6 Qn1      Quebec
##  7 Qn1      Quebec
##  8 Qn2      Quebec
##  9 Qn2      Quebec
## 10 Qn2      Quebec
## # … with 74 more rows
```

```
plant %>%
  select(plant_id, origin) %>%
  head()
```

```
## # A tibble: 6 x 2
##   plant_id origin
##   <ord>    <fct>
## 1 Qn1      Quebec
## 2 Qn1      Quebec
## 3 Qn1      Quebec
## 4 Qn1      Quebec
## 5 Qn1      Quebec
## 6 Qn1      Quebec
```

```
plant %>%
  select(plant_id, origin) %>%
  head(2)
```

```
## # A tibble: 2 x 2
##   plant_id origin
##   <ord>    <fct>
## 1 Qn1      Quebec
```

Subsetting rows with filter

```
names(plant)
```

```
## [1] "plant_id"  "origin"    "treatment" "co2_conc"  "co2_rate"
```

Only plants from Mississippi

```
plant %>%
  filter(origin == "Mississippi")
```

```
## # A tibble: 42 x 5
##    plant_id origin      treatment  co2_conc co2_rate
##    <ord>    <fct>       <fct>         <dbl>    <dbl>
##  1 Mn1      Mississippi nonchilled       95     10.6
##  2 Mn1      Mississippi nonchilled      175     19.2
##  3 Mn1      Mississippi nonchilled      250     26.2
##  4 Mn1      Mississippi nonchilled      350     30
##  5 Mn1      Mississippi nonchilled      500     30.9
##  6 Mn1      Mississippi nonchilled      675     32.4
##  7 Mn1      Mississippi nonchilled     1000     35.5
##  8 Mn2      Mississippi nonchilled       95     12
##  9 Mn2      Mississippi nonchilled      175     22
## 10 Mn2      Mississippi nonchilled      250     30.6
## # … with 32 more rows
```

Subsetting rows with filter

```
names(plant)
```

```
## [1] "plant_id"  "origin"    "treatment" "co2_conc"  "co2_rate"
```

Only plants with rate of co2 uptake of at least 30

```
plant %>%
  filter(co2_rate ⩾ 30)
```

```
## # A tibble: 41 x 5
##    plant_id origin treatment  co2_conc co2_rate
##    <ord>    <fct>  <fct>         <dbl>    <dbl>
##  1 Qn1      Quebec nonchilled      175     30.4
##  2 Qn1      Quebec nonchilled      250     34.8
##  3 Qn1      Quebec nonchilled      350     37.2
##  4 Qn1      Quebec nonchilled      500     35.3
##  5 Qn1      Quebec nonchilled      675     39.2
##  6 Qn1      Quebec nonchilled     1000     39.7
##  7 Qn2      Quebec nonchilled      250     37.1
##  8 Qn2      Quebec nonchilled      350     41.8
##  9 Qn2      Quebec nonchilled      500     40.6
## 10 Qn2      Quebec nonchilled      675     41.4
## # … with 31 more rows
```

Subsetting rows with filter

```
names(plant)
```

```
## [1] "plant_id"  "origin"    "treatment" "co2_conc"  "co2_rate"
```

Only Mississippi plants with rate of at least 30

```
plant %>%
  # Use a , for "and" (&) in filter()
  filter(co2_rate >= 30, origin == "Mississippi")
```

```
## # A tibble: 9 x 5
##   plant_id origin      treatment  co2_conc co2_rate
##   <ord>    <fct>       <fct>         <dbl>    <dbl>
## 1 Mn1      Mississippi nonchilled      350     30
## 2 Mn1      Mississippi nonchilled      500     30.9
## 3 Mn1      Mississippi nonchilled      675     32.4
## 4 Mn1      Mississippi nonchilled     1000     35.5
## 5 Mn2      Mississippi nonchilled      250     30.6
## 6 Mn2      Mississippi nonchilled      350     31.8
## 7 Mn2      Mississippi nonchilled      500     32.4
## 8 Mn2      Mississippi nonchilled      675     31.1
## 9 Mn2      Mississippi nonchilled     1000     31.5
```

Subsetting rows with filter

Only plants exposed to a co2_conc of 500 or 675

```
plant %>%
  filter(co2_conc == 500, co2_conc == 675)
```

```
## # A tibble: 0 x 5
## # … with 5 variables: plant_id <ord>, origin <fct>, treatment <fct>,
## #   co2_conc <dbl>, co2_rate <dbl>
```

```
plant %>%
  ## Need to use the OR operator |
  filter(co2_conc == 500 | co2_conc == 675)
```

```
## # A tibble: 24 x 5
##    plant_id origin treatment  co2_conc co2_rate
##    <ord>    <fct>  <fct>         <dbl>    <dbl>
##  1 Qn1      Quebec nonchilled      500     35.3
##  2 Qn1      Quebec nonchilled      675     39.2
##  3 Qn2      Quebec nonchilled      500     40.6
##  4 Qn2      Quebec nonchilled      675     41.4
##  5 Qn3      Quebec nonchilled      500     42.9
##  6 Qn3      Quebec nonchilled      675     43.9
##  7 Qc1      Quebec chilled         500     32.5
##  8 Qc1      Quebec chilled         675     35.4
```

Introducing the %in% operator

```
"Stephanie" %in% c("Stephanie", "Spielman")
## [1] TRUE

10 %in% 15:20
## [1] FALSE
```

Only plants exposed to a co2_conc of 500 or 675

```
plant %>%
  filter(co2_conc %in% c(675, 500)) %>%
  head(5) # to fit on slide
```

```
## # A tibble: 5 x 5
##   plant_id origin treatment  co2_conc co2_rate
##   <ord>    <fct>  <fct>         <dbl>    <dbl>
## 1 Qn1      Quebec nonchilled      500     35.3
## 2 Qn1      Quebec nonchilled      675     39.2
## 3 Qn2      Quebec nonchilled      500     40.6
## 4 Qn2      Quebec nonchilled      675     41.4
## 5 Qn3      Quebec nonchilled      500     42.9
```

Combining dplyr verbs

Only Mississippi plants with rate of at least 30, and remove origin column

```
plant %>%
  filter(co2_rate >= 30, origin == "Mississippi") %>%
  select(-origin) %>%
  head() ## to fit on slides!
```

```
## # A tibble: 6 x 4
##   plant_id treatment  co2_conc co2_rate
##   <ord>    <fct>         <dbl>    <dbl>
## 1 Mn1      nonchilled      350     30
## 2 Mn1      nonchilled      500     30.9
## 3 Mn1      nonchilled      675     32.4
## 4 Mn1      nonchilled     1000     35.5
## 5 Mn2      nonchilled      250     30.6
## 6 Mn2      nonchilled      350     31.8
```

```
# we can SAVE the output!!
plant %>%
  filter(co2_rate >= 30, origin == "Mississippi") %>%
  select(-origin) -> miss_plant_over30
```

And don't forget to COMMENT!!

```
plant %>%
  # Keep only plants with update ⩾ 30 and who are from mississippi
  filter(co2_rate ⩾ 30, origin == "Mississippi") %>%
  # Remove the origin column
  select(-origin) → miss_plant_over30
```

A note about namespaces

- `filter()` and `select()` are functions in the `dplyr` package (library)
- `select()` is ALSO a function in another library called `MASS`
    - Package that accompanies book "Modern Applied Statistics with S"
- `filter()` is ALSO a function in another library called `stats`
    - Package that comes with R
- **How does R know which `filter()` or `select()` you are using?**

A note about namespaces

- All variables, functions, definitions of any kind are part of a **namespace**
- R searches the *most recently loaded* namespace first
    - If you get a weird error, you probably didn't load the library
- You can always *explicitly* use functions as `namespace::function()`
    - `dplyr::filter()`
    - `dplyr::select()`

Creating new columns with mutate()

```
names(plant)
```

```
## [1] "plant_id"  "origin"    "treatment" "co2_conc"  "co2_rate"
```

Currently, `co2_conc` is in units mL/L. What if we want it in dL/L?

```
plant %>%
  mutate(co2_conc_dl = co2_conc / 10)
```

```
## # A tibble: 84 x 6
##    plant_id origin treatment  co2_conc co2_rate co2_conc_dl
##    <ord>    <fct>  <fct>         <dbl>    <dbl>       <dbl>
##  1 Qn1      Quebec nonchilled       95     16           9.5
##  2 Qn1      Quebec nonchilled      175     30.4        17.5
##  3 Qn1      Quebec nonchilled      250     34.8        25
##  4 Qn1      Quebec nonchilled      350     37.2        35
##  5 Qn1      Quebec nonchilled      500     35.3        50
##  6 Qn1      Quebec nonchilled      675     39.2        67.5
##  7 Qn1      Quebec nonchilled     1000     39.7       100
##  8 Qn2      Quebec nonchilled       95     13.6         9.5
##  9 Qn2      Quebec nonchilled      175     27.3        17.5
## 10 Qn2      Quebec nonchilled      250     37.1        25
## # … with 74 more rows
```

```r
plant %>%
  ## This will OVERWRITE the existing column - be careful!!
  mutate(co2_conc = co2_conc / 10) %>%
  head(2)
```

```
## # A tibble: 2 x 5
##   plant_id origin treatment  co2_conc co2_rate
##   <ord>    <fct>  <fct>         <dbl>    <dbl>
## 1 Qn1      Quebec nonchilled      9.5     16
## 2 Qn1      Quebec nonchilled     17.5     30.4
```

Renaming columns with rename()

- Syntax: `rename(NEW = OLD)`. Very easy to mess up! That's ok!

```
plant %>%
  rename(concentration = co2_conc) %>%
  head(2)
```

```
## # A tibble: 2 x 5
##   plant_id origin treatment  concentration co2_rate
##   <ord>    <fct>  <fct>              <dbl>    <dbl>
## 1 Qn1      Quebec nonchilled            95     16
## 2 Qn1      Quebec nonchilled           175     30.4
```

```
plant %>%
  rename(co2_conc = concentration)
```

```
## Error: Can't rename columns that don't exist.
##  [31mx [39m Column `concentration` doesn't exist.
```

Arranging rows with arrange()

```
small_plant ← plant %>% select(co2_rate, co2_conc, origin)
```

Arrange in order of `co2_rate`

```
small_plant %>%
  arrange(co2_rate)
```

```
## # A tibble: 84 x 3
##    co2_rate co2_conc origin
##       <dbl>    <dbl> <fct>
##  1      7.7       95 Mississippi
##  2      9.3       95 Quebec
##  3     10.5       95 Mississippi
##  4     10.6       95 Mississippi
##  5     10.6       95 Mississippi
##  6     11.3       95 Mississippi
##  7     11.4      175 Mississippi
##  8     12         95 Mississippi
##  9     12.3      250 Mississippi
## 10     12.5      500 Mississippi
## # … with 74 more rows
```

Arrange in **descending** order of `co2_rate`

```
small_plant %>%
  arrange(desc(co2_rate))
```

```
## # A tibble: 84 x 3
##    co2_rate co2_conc origin
##       <dbl>    <dbl> <fct>
##  1     45.5     1000 Quebec
##  2     44.3     1000 Quebec
##  3     43.9      675 Quebec
##  4     42.9      500 Quebec
##  5     42.4     1000 Quebec
##  6     42.1      350 Quebec
##  7     41.8      350 Quebec
##  8     41.4      675 Quebec
##  9     41.4     1000 Quebec
## 10     40.6      500 Quebec
## # … with 74 more rows
```

A mini wrangle!

```
head(plant)
```

```
## # A tibble: 6 x 5
##   plant_id origin treatment  co2_conc co2_rate
##   <ord>    <fct>  <fct>         <dbl>    <dbl>
## 1 Qn1      Quebec nonchilled       95     16
## 2 Qn1      Quebec nonchilled      175     30.4
## 3 Qn1      Quebec nonchilled      250     34.8
## 4 Qn1      Quebec nonchilled      350     37.2
## 5 Qn1      Quebec nonchilled      500     35.3
## 6 Qn1      Quebec nonchilled      675     39.2
```

Find the ratio of concentration/rate for all the plants from Quebec exposed to a "chilled" environment

- First, make a plan - how would you do this **by hand**?
- Second, WRITE YOUR PLAN DOWN
- Third, go *line by line* to build up your plan

Conc/rate ratio for chilled Quebec plants

```
plant
```

```
## # A tibble: 84 x 5
##    plant_id origin treatment  co2_conc co2_rate
##    <ord>    <fct>  <fct>         <dbl>    <dbl>
##  1 Qn1      Quebec nonchilled       95     16  
##  2 Qn1      Quebec nonchilled      175     30.4
##  3 Qn1      Quebec nonchilled      250     34.8
##  4 Qn1      Quebec nonchilled      350     37.2
##  5 Qn1      Quebec nonchilled      500     35.3
##  6 Qn1      Quebec nonchilled      675     39.2
##  7 Qn1      Quebec nonchilled     1000     39.7
##  8 Qn2      Quebec nonchilled       95     13.6
##  9 Qn2      Quebec nonchilled      175     27.3
## 10 Qn2      Quebec nonchilled      250     37.1
## # … with 74 more rows
```

Conc/rate ratio for chilled Quebec plants

```
plant %>%
  filter(origin == "Quebec")
```

```
## # A tibble: 42 x 5
##    plant_id origin treatment  co2_conc co2_rate
##    <ord>    <fct>  <fct>         <dbl>    <dbl>
##  1 Qn1      Quebec nonchilled       95     16
##  2 Qn1      Quebec nonchilled      175     30.4
##  3 Qn1      Quebec nonchilled      250     34.8
##  4 Qn1      Quebec nonchilled      350     37.2
##  5 Qn1      Quebec nonchilled      500     35.3
##  6 Qn1      Quebec nonchilled      675     39.2
##  7 Qn1      Quebec nonchilled     1000     39.7
##  8 Qn2      Quebec nonchilled       95     13.6
##  9 Qn2      Quebec nonchilled      175     27.3
## 10 Qn2      Quebec nonchilled      250     37.1
## # … with 32 more rows
```

Conc/rate ratio for chilled Quebec plants

```
plant %>%
  filter(origin == "Quebec") %>%
  filter(treatment == "chilled")
```

```
## # A tibble: 21 x 5
##    plant_id origin treatment co2_conc co2_rate
##    <ord>    <fct>  <fct>        <dbl>    <dbl>
##  1 Qc1      Quebec chilled         95     14.2
##  2 Qc1      Quebec chilled        175     24.1
##  3 Qc1      Quebec chilled        250     30.3
##  4 Qc1      Quebec chilled        350     34.6
##  5 Qc1      Quebec chilled        500     32.5
##  6 Qc1      Quebec chilled        675     35.4
##  7 Qc1      Quebec chilled       1000     38.7
##  8 Qc2      Quebec chilled         95      9.3
##  9 Qc2      Quebec chilled        175     27.3
## 10 Qc2      Quebec chilled        250     35
## # … with 11 more rows
```

Conc/rate ratio for chilled Quebec plants

```
plant %>%
  filter(origin == "Quebec") %>%
  filter(treatment == "chilled") %>%
  mutate(ratio = co2_conc / co2_rate)
```

```
## # A tibble: 21 x 6
##    plant_id origin treatment co2_conc co2_rate ratio
##    <ord>    <fct>  <fct>        <dbl>    <dbl> <dbl>
##  1 Qc1      Quebec chilled         95     14.2  6.69
##  2 Qc1      Quebec chilled        175     24.1  7.26
##  3 Qc1      Quebec chilled        250     30.3  8.25
##  4 Qc1      Quebec chilled        350     34.6 10.1
##  5 Qc1      Quebec chilled        500     32.5 15.4
##  6 Qc1      Quebec chilled        675     35.4 19.1
##  7 Qc1      Quebec chilled       1000     38.7 25.8
##  8 Qc2      Quebec chilled         95      9.3 10.2
##  9 Qc2      Quebec chilled        175     27.3  6.41
## 10 Qc2      Quebec chilled        250     35    7.14
## # … with 11 more rows
```

Why you have to go step-by-step

```
plant %>%
  filter(Origin == "Quebec") %>%
  filter(treatment == "chill") %>%
  mutate(ratio = conc_co2 / rate)
```

```
## Error: Problem with `filter()` input `..1`.
## x object 'Origin' not found
## ℹ Input `..1` is `Origin == "Quebec"`.
```

```
plant %>%
  filter(origin == "Quebec") %>%
  filter(treatment == "chill") %>%
  mutate(ratio = conc_co2 / co2_rate)
```

```
## Error: Problem with `mutate()` input `ratio`.
## x object 'conc_co2' not found
## ℹ Input `ratio` is `conc_co2/co2_rate`.
```