# data_wrangling() && ("manipulation" %in% R)

🐨🐟🌿 %>% X %>% R %>% 🤓📊🥰

```
> day[2]
```

Ruan van Mazijk

Notes & slides will go up here:

# tinyurl.com/r-with-ruan

(But I encourage you to make your own notes!)

# `> workshop$outline[1:3]`

## DAY 1

Tidy data principles
& `tidyr`

## DAY 2

Manipulating data
& an intro to `dplyr`

## DAY 3

Extending your data
with `mutate()`,
`summarise()`
& friends

# tidyr::

```
# Verbs to tidy your data

# Untidy observations?
gather()     # if > 1 observation per row
spread()     # if observations live in > 1 row

# Untidy variables?
separate()   # if > 1 variable per column
unite()      # if variables live in > 1 column
```

# > workshop$outline[2:3]

## DAY 2

Manipulating data
& an intro to `dplyr`

## DAY 3

Extending your data
with `mutate()`,
`summarise()`
& friends

```
# base R

data[       , columns ]


data[ rows  ,         ]
```

```r
# base R

data[, 4]
data[, "plantheight"]

data[1:10, ]
data[data$soil == "a", ]
```

```r
# base R

data[, "plantheight"]


data[data$soil == "a", ]
```

```r
# tidyverse R

data %>%
 select(plantheight)

data %>%
  filter(soil == "a")
```

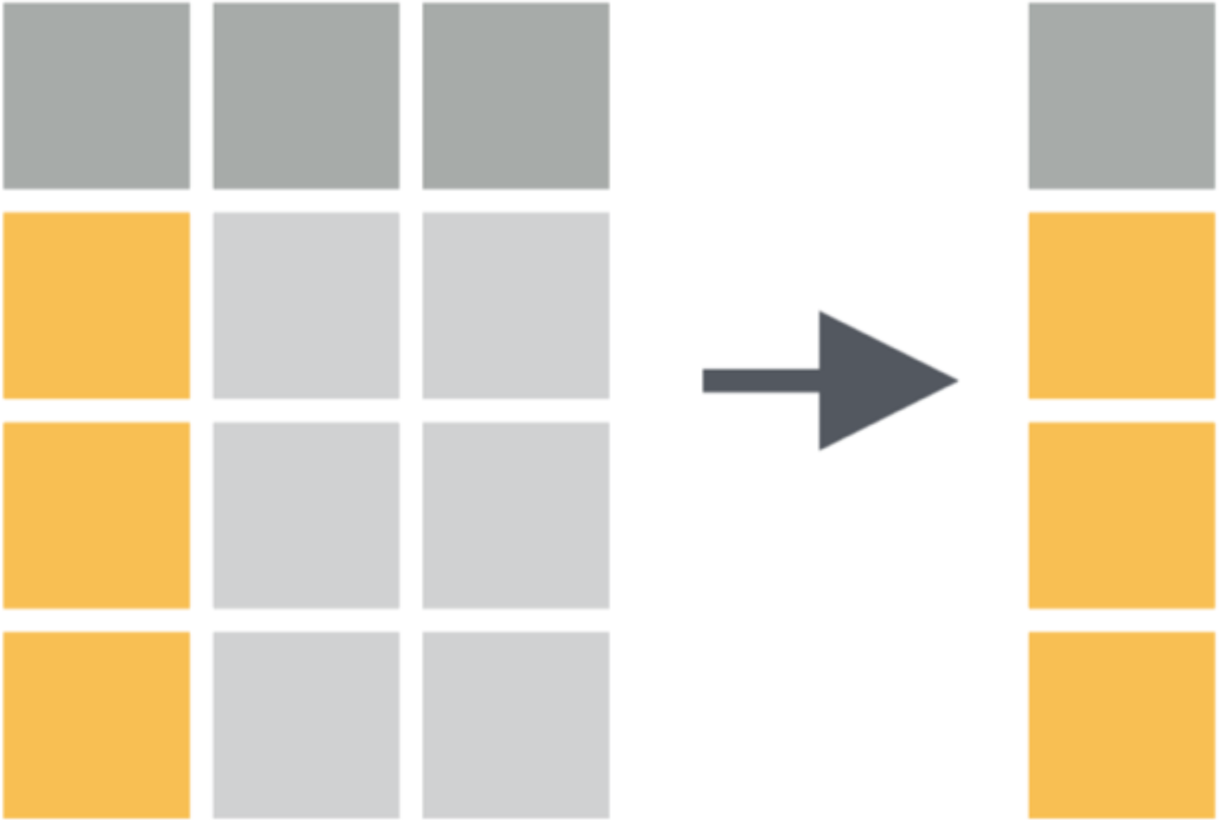
# dplyr::

```
# Verbs to manipulate your data
```

# dplyr::

```
# Verbs to manipulate your data

select()      # operates on columns
filter()      # operates on rows
```

```
data %>%
  select(...)
```



CC BY SA RStudio https://www.rstudio.com/resources/cheatsheets/

```
data %>%
  select(plant_height, soil, lon, lat, veg_type)
```

```
data %>%
  select(plant_height, soil, lon, lat, veg_type)

data %>%
  select(plant_height:veg_type)
# Think 1:10 but with words!
```

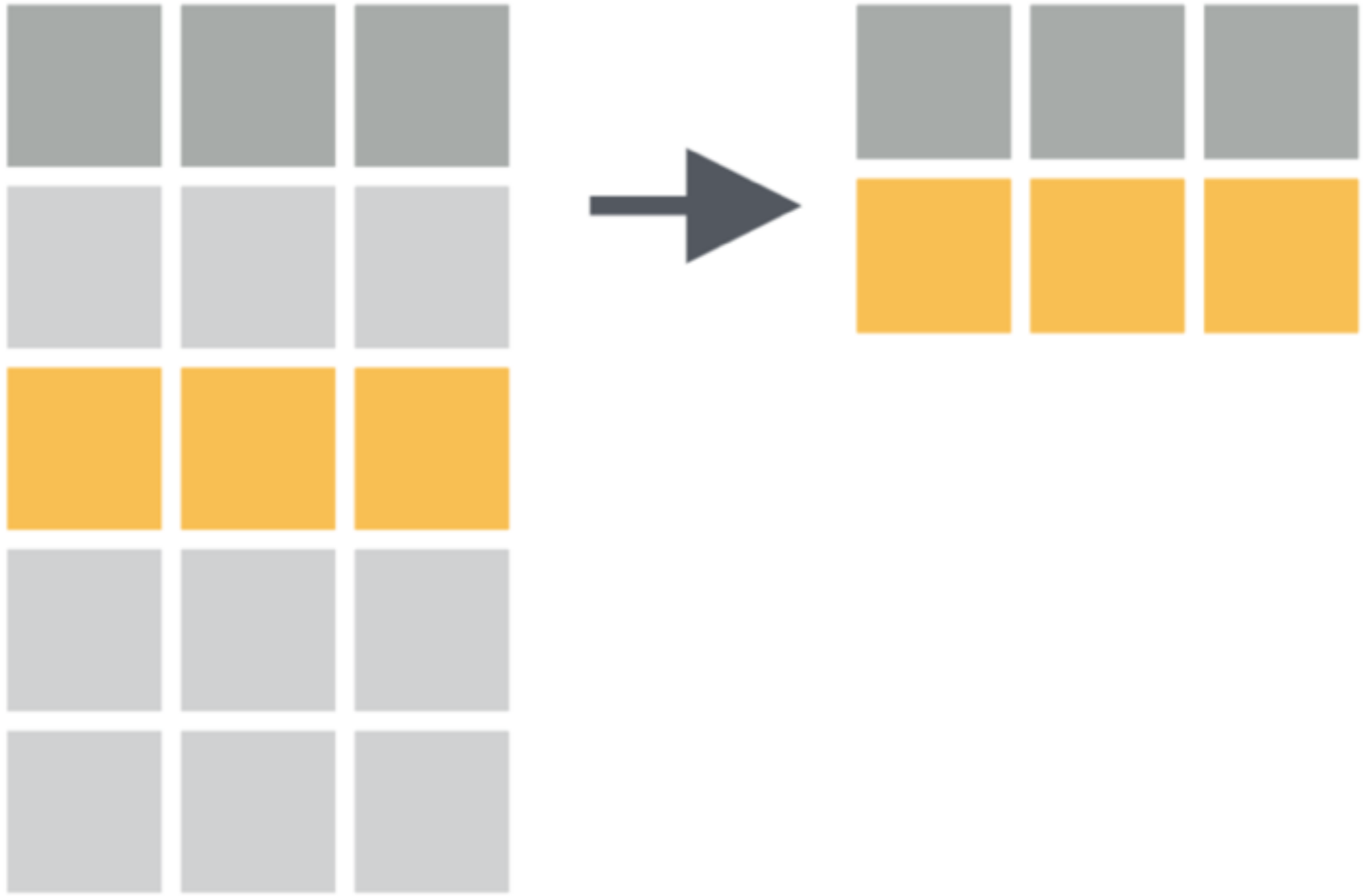
```r
data %>%
  select(plant_height, soil, lon, lat, veg_type)

data %>%
  select(plant_height:veg_type)
# Think 1:10 but with words!

data %>%
  select(-mean_annual_temp)
# Think data[, -10],
# Or like gather(key, value, -foo)
```

```r
data %>%
  select(plant_height, plant_weight, plant_LAI)
```

```
data %>%
  select(plant_height, plant_weight, plant_LAI)

data %>%
  select(starts_with("plant"))
# Also:
#   contains()   ends_with()  matches()
#   num_range()  one_of()     starts_with()
```

```r
data %>%
  select(plant_height, plant_weight, plant_LAI)

data %>%
  select(starts_with("plant"))
# Also:
#   contains()   ends_with()  matches()
#   num_range()  one_of()     starts_with()

data %>%
  select_if(is.numeric)
# Accepts base R functions (sans "()"):
#   is.logical  is.character  is.numeric
#   is.factor   is.datetime
```

```
data %>%
  filter(...)
```



CC BY SA RStudio https://www.rstudio.com/resources/cheatsheets/

```
data %>%
  filter(plant_height <= 10)
```

```r
data %>%
  filter(plant_height <= 10)

data %>%
  filter(plant_height <= 10, vegtype == "fynbos")
```

```r
data %>%
  filter(plant_height <= 10)

data %>%
  filter(plant_height <= 10, vegtype == "fynbos")
# Multiple conditions must all be satisfied
# So it "&&"s them, so it would be the same as:
data %>%
  filter(plant_height <= 10 & vegtype == "fynbos")
```

```r
data %>%
  filter(plant_height <= 10)

data %>%
  filter(plant_height <= 10, vegtype == "fynbos")
# Multiple conditions must all be satisfied
# So it "&"s them, so it would be the same as:
data %>%
  filter(plant_height <= 10 & vegtype == "fynbos")

data %>%
  filter(plant_height <= 10 | plant_weight >= 60)
# We can use "or": |
# Intervals?

data %>%
  filter(plant_height <= 10 & plant_height >= 0.5)

# There is also a tidy way!
data %>%
  filter(plant_height %>% between(0.5, 10))
```

```
> demo()
```