Sociology Quant Camp

Introduction to R

Module 2: piping and tidyverse

Monica Alexander, Statistical Sciences and Sociology

Using the tidyverse to manipulate real data

- In the previous module, we saw some functions and loaded in the `tidyverse` package

- Tidyverse has a range of functions that make it easier to manipulate real data

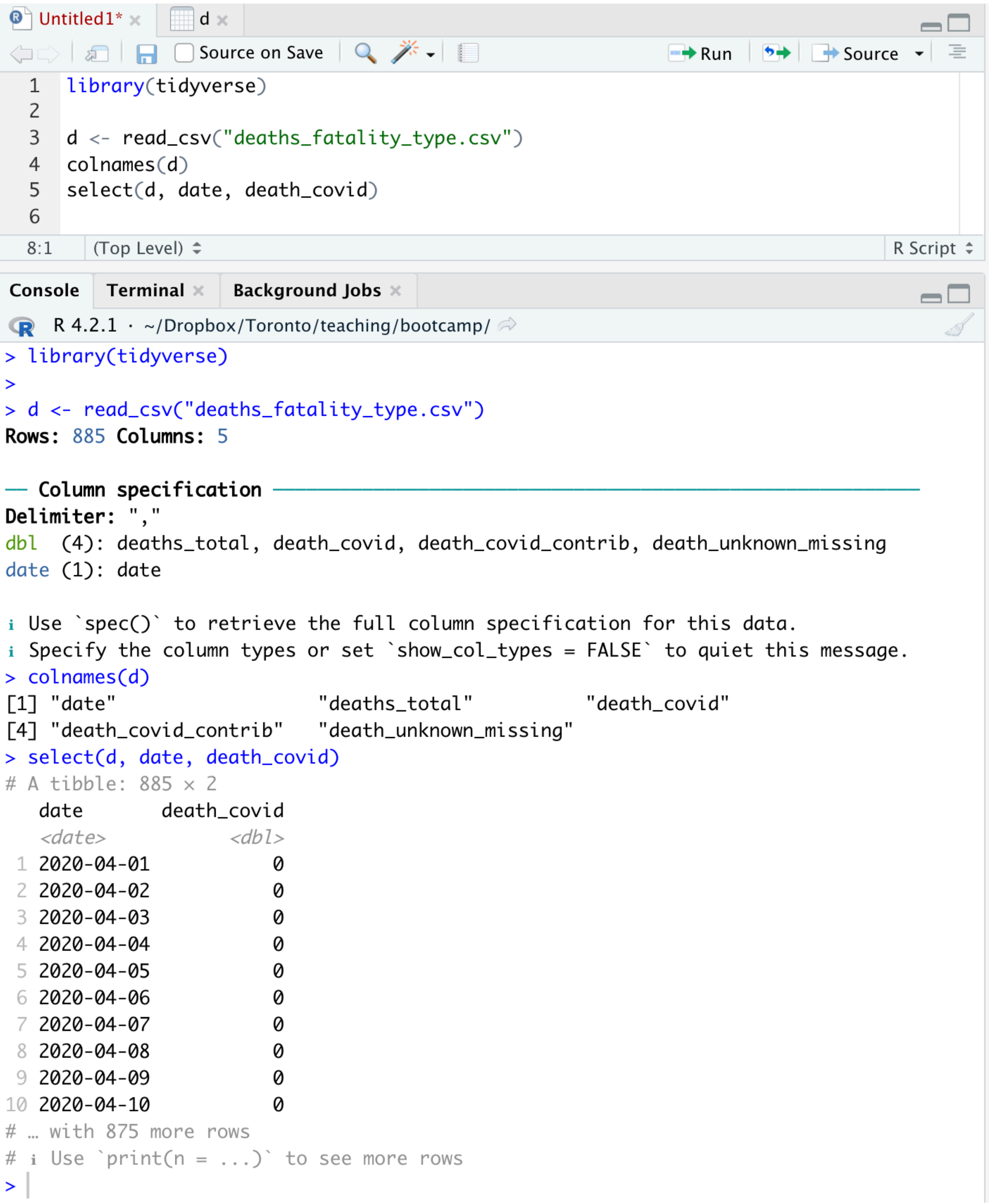
- Things like: adding columns, selecting columns, filtering out rows based on certain values…

- These functions have been specifically designed to work with datasets with lots of variables of different types

A first example

- Let’s read in the Covid-19 deaths data and select some columns

- Note that `colnames()` is a useful function to see what the columns are called



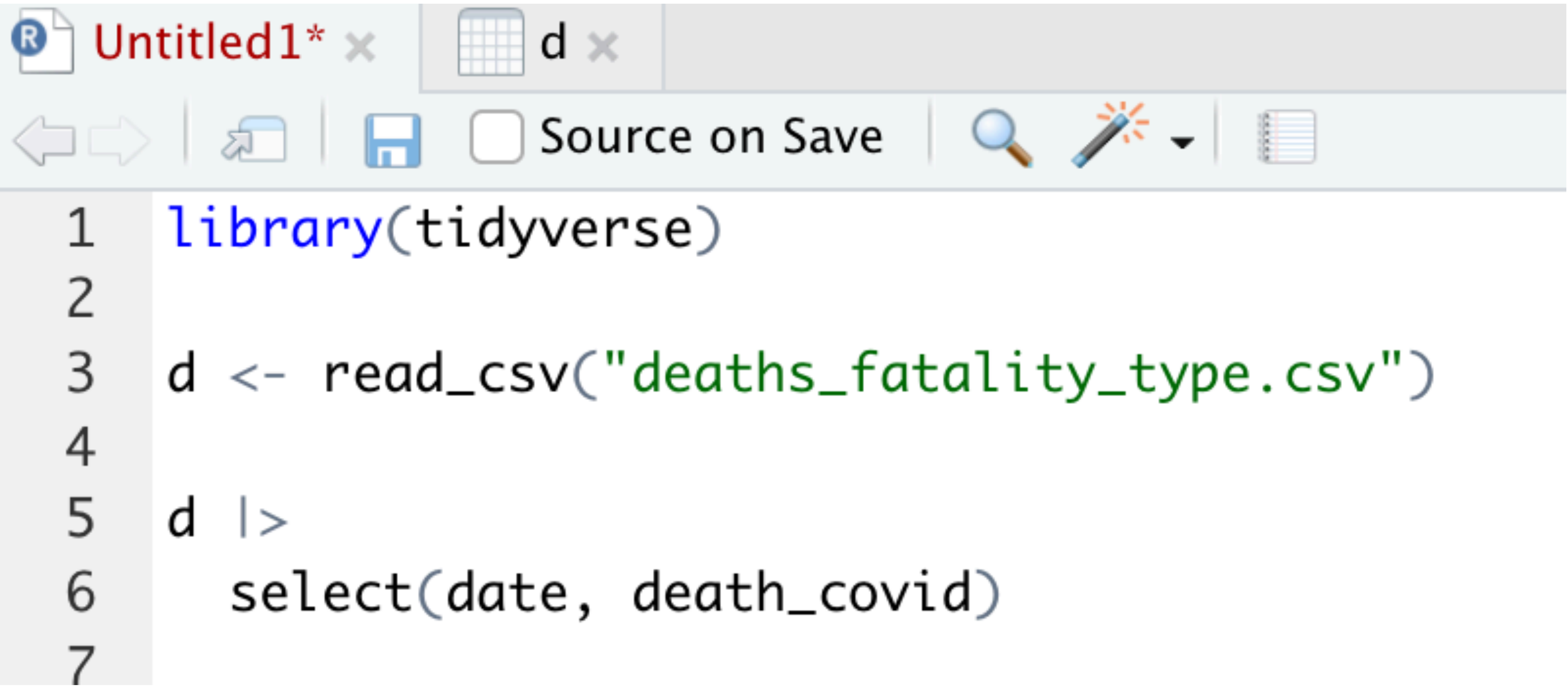
```r
library(tidyverse)

d <- read_csv("deaths_fatality_type.csv")
colnames(d)
select(d, date, death_covid)

```

```
> library(tidyverse)
>
> d <- read_csv("deaths_fatality_type.csv")
Rows: 885 Columns: 5

── Column specification ────────────────────────────────────────────────────
Delimiter: ","
dbl  (4): deaths_total, death_covid, death_covid_contrib, death_unknown_missing
date (1): date

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.
> colnames(d)
[1] "date"                "deaths_total"          "death_covid"
[4] "death_covid_contrib" "death_unknown_missing"
> select(d, date, death_covid)
# A tibble: 885 × 2
   date       death_covid
   <date>           <dbl>
 1 2020-04-01           0
 2 2020-04-02           0
 3 2020-04-03           0
 4 2020-04-04           0
 5 2020-04-05           0
 6 2020-04-06           0
 7 2020-04-07           0
 8 2020-04-08           0
 9 2020-04-09           0
10 2020-04-10           0
# … with 875 more rows
# ℹ Use `print(n = ...)` to see more rows
>
```

Demo: selecting columns

The pipe |>

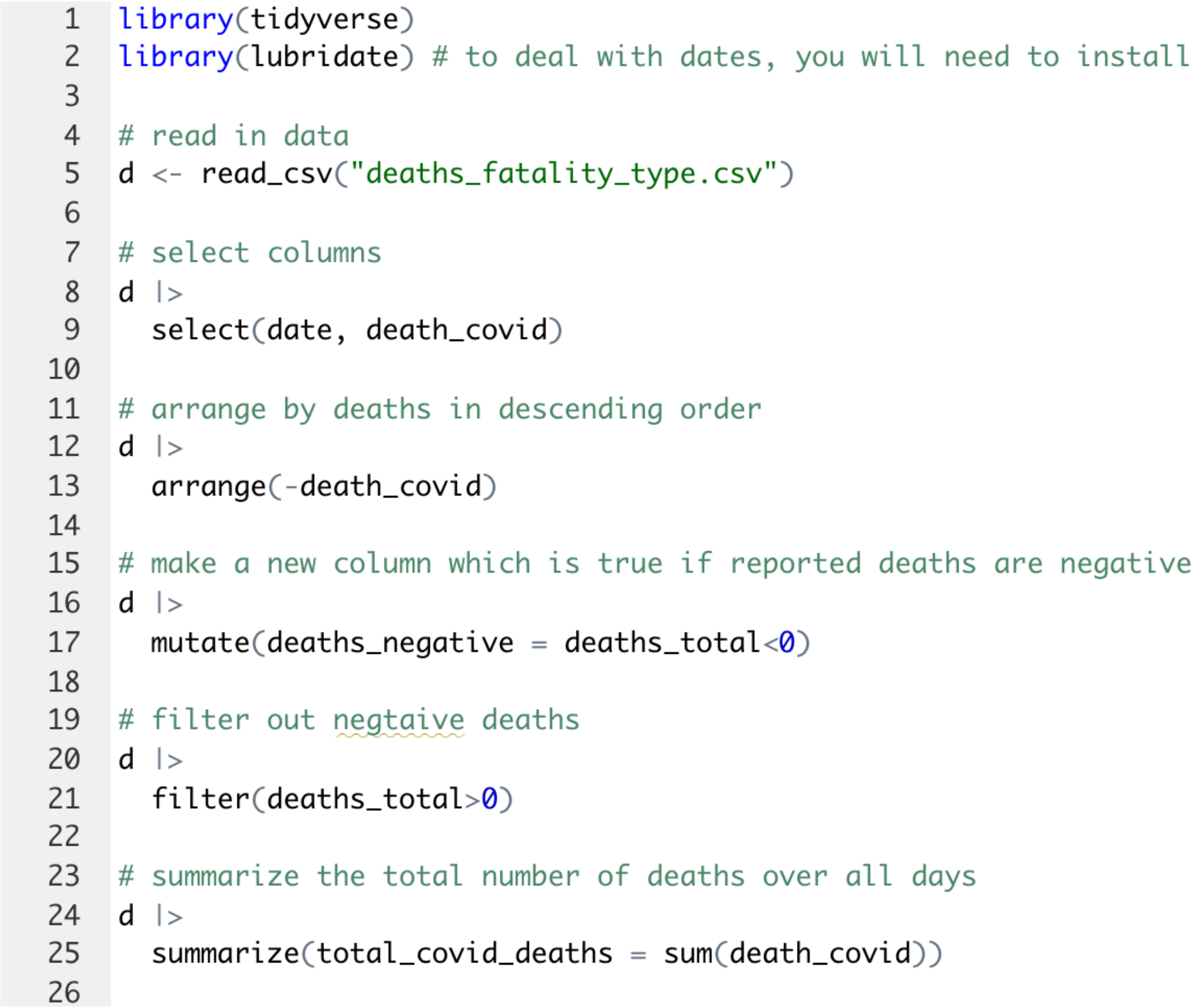
- An alternative way of writing code
- Makes the code read more like a sentence
- Read the pipe as “and then”
- So here we are taking the data AND THEN selecting columns



```r
library(tidyverse)

d <- read_csv("deaths_fatality_type.csv")

d |>
  select(date, death_covid)

```

Core tidyverse functions

- `select`: select columns
- `arrange`: sort/arrange by value
- `mutate`: make a new column
- `filter`: filter out certain rows
- `summarize`: produce summaries of data
- `group_by`: group the data by certain variable(s)



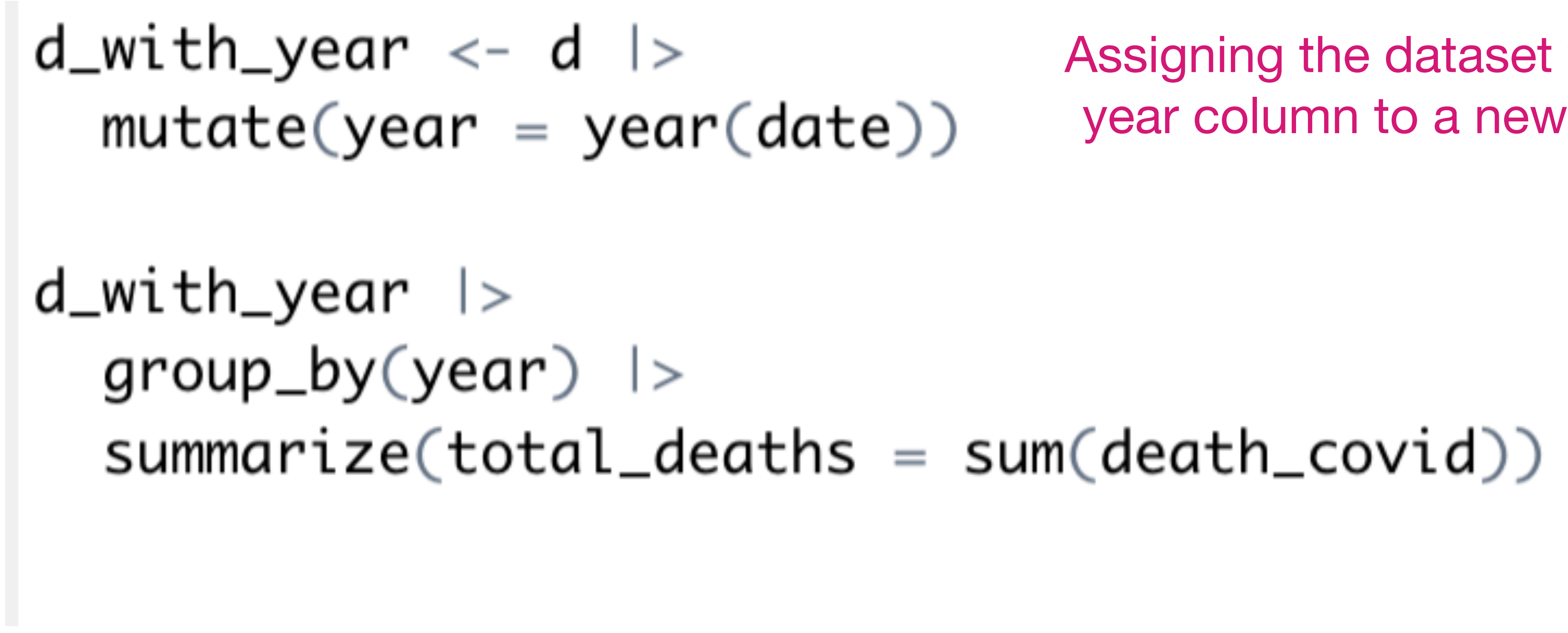
```r
library(tidyverse)
library(lubridate) # to deal with dates, you will need to install

# read in data
d <- read_csv("deaths_fatality_type.csv")

# select columns
d |>
  select(date, death_covid)

# arrange by deaths in descending order
d |>
  arrange(-death_covid)

# make a new column which is true if reported deaths are negative
d |>
  mutate(deaths_negative = deaths_total<0)

# filter out negtaive deaths
d |>
  filter(deaths_total>0)

# summarize the total number of deaths over all days
d |>
  summarize(total_covid_deaths = sum(death_covid))

```

Demo: tidyverse functions

group_by

- The group_by function is extremely powerful when used in conjunction with summarize to get summaries by groups

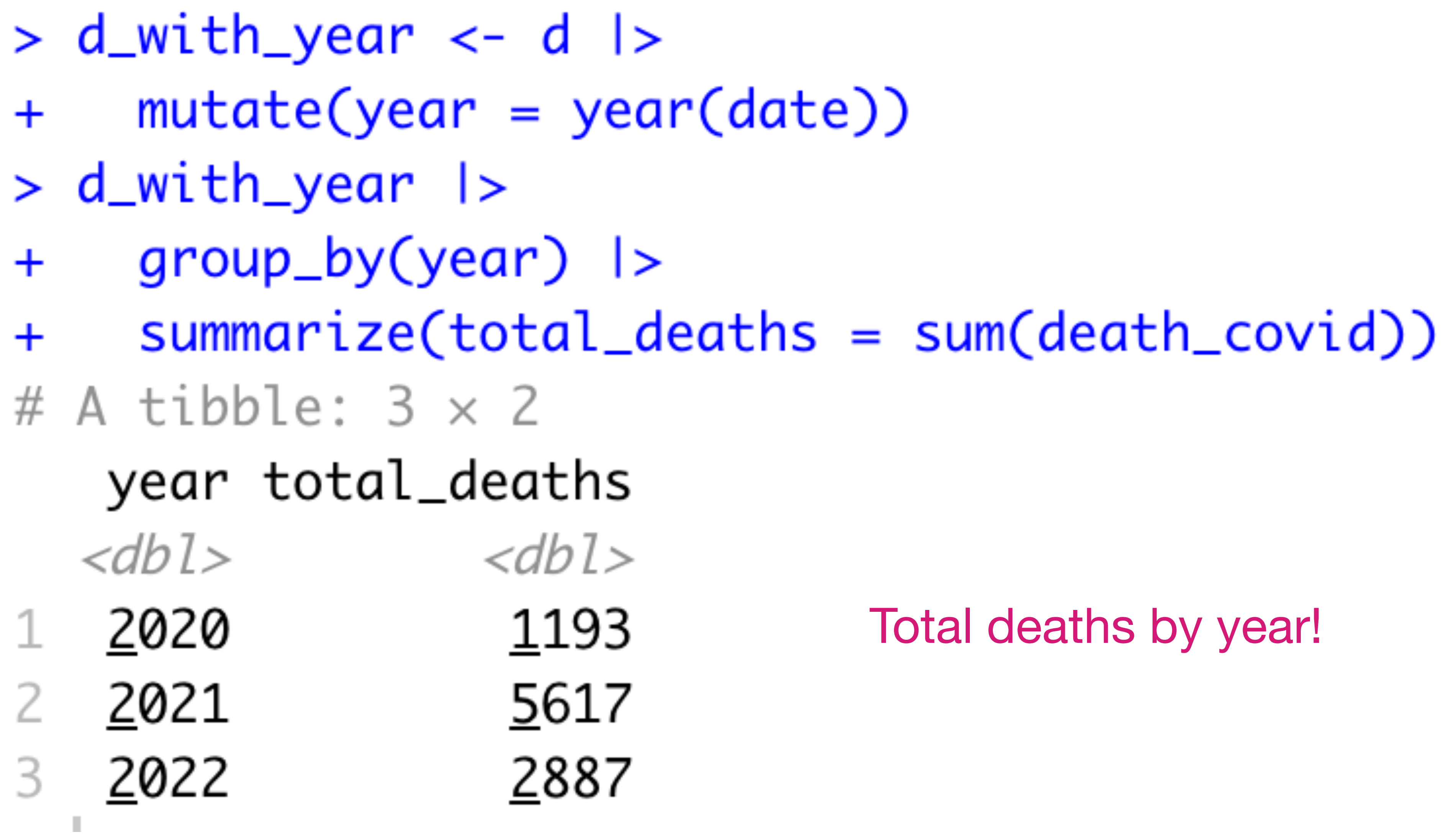
- Note that we can thread together multiple pipes!



```
d_with_year <- d |>
  mutate(year = year(date))

d_with_year |>
  group_by(year) |>
  summarize(total_deaths = sum(death_covid))
```

Assigning the dataset with new year column to a new dataset

Here is the output:



```
> d_with_year <- d |>
+   mutate(year = year(date))
> d_with_year |>
+   group_by(year) |>
+   summarize(total_deaths = sum(death_covid))
# A tibble: 3 × 2
   year total_deaths
  <dbl>        <dbl>
1  2020         1193
2  2021         5617
3  2022         2887
```

Total deaths by year!

Demo: more complicated tidyverse functions

Where to get help

- Lots of good, free online sources
  - R for Data Science: https://www.tidyverse.org/learn/
  - Telling stories with data: https://tellingstorieswithdata.com/
  - Tidyverse skills for data science: https://jhudatascience.org/tidyversecourse/intro.html
- Google/Stack Overflow
- Email
- Practice, practice, practice; don't be afraid of mistakes