

# Code quality

You can download this .qmd file from [here](#). Just hit the Download Raw File button.

## Code style

We are going to take a timeout at this point to focus a little on code quality. [Chapter 4](#) in R4DS provides a nice introduction to code style and why it's important. As they do in that chapter, we will follow the [tidyverse style guide](#) in this class.

Based on those resources, can you improve this poorly styled code chunk using the data from our DS1 Review activity?

```
library(tidyverse)
```

```
-- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
v dplyr      1.1.4      v readr      2.1.5
v forcats    1.0.0      v stringr    1.5.2
v ggplot2    3.5.2      v tibble     3.3.0
v lubridate  1.9.4      v tidyr      1.3.1
v purrr      1.1.0
```

```
-- Conflicts ----- tidyverse_conflicts() --
```

```
x dplyr::filter() masks stats::filter()
```

```
x dplyr::lag()     masks stats::lag()
```

```
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become
```

```
vaccine_data <- read_csv("https://proback.github.io/264_fall_2025/Data/vaccinations_2021.csv")
```

```
Rows: 3053 Columns: 14
```

```
-- Column specification -----
```

```
Delimiter: ","
```

```
chr  (4): state, county, region, metro_status
```

dbl (10): rural\_urban\_code, perc\_complete\_vac, tot\_pop, votes\_Trump, votes\_B...

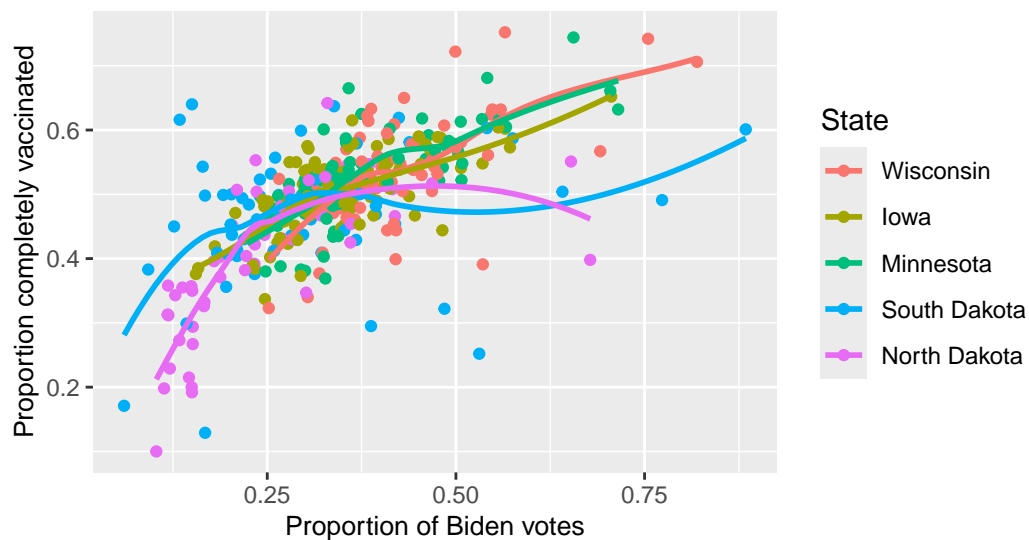
i Use ``spec()`` to retrieve the full column specification for this data.

i Specify the column types or set ``show_col_types = FALSE`` to quiet this message.

```
vaccine_data |>
  filter(state %in% c("Minnesota", "Iowa", "Wisconsin", "North Dakota", "South Dakota")) |>
  mutate(
    state_ordered = fct_reorder2(state, perc_Biden, perc_complete_vac),
    prop_Biden = perc_Biden / 100,
    prop_complete_vac = perc_complete_vac / 100
  ) |>
  ggplot(aes(x = prop_Biden, y = prop_complete_vac, color = state_ordered)) +
  geom_point() +
  geom_smooth(se = FALSE) +
  labs(
    color = "State",
    x = "Proportion of Biden votes",
    y = "Proportion completely vaccinated",
    title = "The positive relationship between Biden votes
and \n vaccination rates by county differs by state"
  ) +
  theme(axis.title = element_text(size = 10), plot.title = element_text(size = 12))
```

``geom_smooth()`` using `method = 'loess'` and `formula = 'y ~ x'`

The positive relationship between Biden votes and vaccination rates by county differs by state



## Code comments

Please read [Fostering Better Coding Practices for Data Scientists](#), which lays out a nice case for the importance of teaching good coding practices. In particular, their Top 10 List can help achieve the four Cs (correctness, clarity, containment, and consistency) that typify high-quality code:

1. Choose good names.
2. Follow a style guide consistently.
3. Create documents using tools that support reproducible workflows.
4. Select a coherent, minimal, yet powerful tool kit.
5. Don't Repeat Yourself (DRY).
6. Take advantage of a functional programming style.
7. Employ consistency checks.
8. Learn how to debug and to ask for help.
9. Get (version) control of the situation.
10. Be multilingual.

Please also read the Stack Overflow blog on [Best Practices for Writing Code Comments](#) with their set of 9 rules:

- Rule 1: Comments should not duplicate the code.
- Rule 2: Good comments do not excuse unclear code.

- Rule 3: If you can't write a clear comment, there may be a problem with the code.
- Rule 4: Comments should dispel confusion, not cause it.
- Rule 5: Explain unidiomatic code in comments.
- Rule 6: Provide links to the original source of copied code.
- Rule 7: Include links to external references where they will be most helpful.
- Rule 8: Add comments when fixing bugs.
- Rule 9: Use comments to mark incomplete implementations.

In your projects and homework for this course, we will look for good style and good commenting to optimize your abilities as a collaborating data scientist!