

R and ggplot2 Workshop

Khushi Desai

Regenstein Library
Center for Digital Scholarship

November 7, 2025

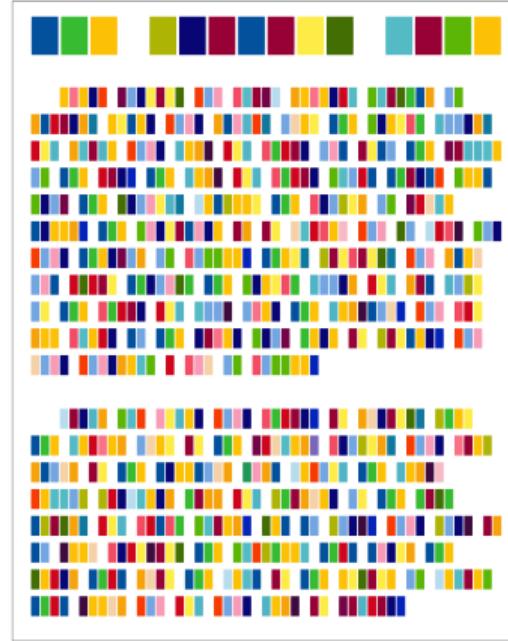
Today's Journey

- 1 ggplot2: The Grammar of Graphics
- 2 Inspiration: Celeste Ng
- 3 Anatomy of a ggplot call
- 4 Data Art: A Process
- 5 Making It Print-Ready
- 6 Wrap-up

ggplot2: The Grammar of Graphics

- ggplot2 is an R package for producing visualizations of data
- By 2008, Hadley Wickham had written the reshape and ggplot2 packages as part of his dissertation at Iowa State.
- Since then, ggplot2 has become a de facto standard among statisticians seeking to produce high-quality and reproducible data visualizations (although not all are converts).

Inspiration: Celeste Ng



Annie Dillard's *The Writing Life* transformed into color-coded blocks using ggplot2.

Anatomy of a ggplot call

Initialize:

- ➊ `ggplot()` - this creates a blank plot

```
ggplot()
```

Anatomy of a ggplot call

Data:

- ① `ggplot()` - this creates a blank plot
- ② Specifying data doesn't update the plot yet, but it does tell ggplot where the data is coming from

```
ggplot(data = cars)
```

Anatomy of a ggplot call

Aesthetic Mapping:

- ➊ `ggplot()` - this creates a blank plot
- ➋ Specifying data doesn't update the plot yet, but it does tell `ggplot` where the data is coming from.
- ➌ '`aes`' specifies the aesthetics that will be plotted. It generally produces default scales and grid lines.

```
ggplot(data = cars,  
       mapping = aes(x = speed, y = dist))
```

Anatomy of a ggplot call

Geometry:

- ① `ggplot()` - this creates a blank plot
- ② Specifying data doesn't update the plot yet, but it does tell ggplot where the data is coming from.
- ③ 'aes' specifies the aesthetics that will be plotted. It generally produces default scales and grid lines.
- ④ The 'geom' is the kind of plot layer we would like to add

```
1 ggplot(data = cars,
2       mapping = aes(x = speed, y = dist)) +
3     geom_point()
```

Anatomy of a ggplot call

Labels:

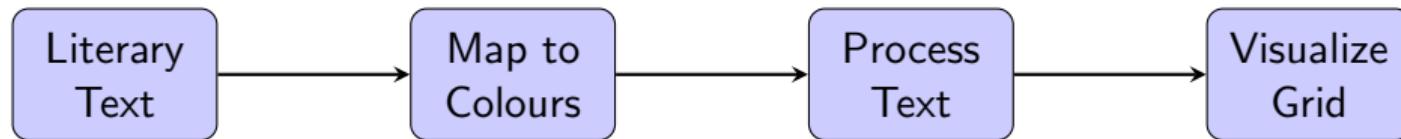
- ① `ggplot()` - this creates a blank plot
- ② Specifying data doesn't update the plot yet, but it does tell ggplot where the data is coming from.
- ③ 'aes' specifies the aesthetics that will be plotted. It generally produces default scales and grid lines.
- ④ The 'geom' is the kind of plot layer we would like to add
- ⑤ We can edit the text in the `labs` function.

```
1 ggplot(data = cars,
2     mapping = aes(x = speed, y = dist)) +
3     geom_point() +
4     labs(x = "Speed",
5         y = "Dist",
6         title = "Title here",
7         caption = "Caption here")
```

The ggplot2 cheatsheet

We have much more to cover with ggplot2! Here's the “official” cheatsheet:
<https://rstudio.github.io/cheatsheets/data-visualization.pdf>

The Process: Overview



Key Steps:

- ① Add in title and body text.
- ② Map individual alphabets to their respective colours.
- ③ Process title and body text.
- ④ Use ggplot to create a beautiful artwork!

Step 1: Reading the Text

```
my_title <- "I Must Betray You" #CHANGE THIS  
1  
2  
3 my_text <- "An old woman is fast asleep when she hears a knock at  
the door.  
Who is it? she whispers, terrified.  
It is death, the voice answers.  
Oh, good. I thought it might be the Securitate." #CHANGE THIS  
4  
5  
6
```

Tip: Experiment with different kinds of literature, poetry, prose, or quotes.

Source: Ruta Sepetys' *I Must Betray You*

Step 2: Setting up a frame

```
chars_per_row <- 15
```

Play around with this number to adjust your visual output.

Step 3: Color Mapping

```
letter_colors <- c(  
  a = "red", b = "#4ECDC4", c = "#45B7D1", d = "#FFA07E", e = "#98  
  D8C8",  
  f = "#6C5CE7", g = "#FDCB6E", h = "#E17055", i = "cyan", j = "#  
  A29BFE",  
  k = "#00B894", l = "#FFEAA7", m = "#DFE6E9", n = "#FF7675", o = "#  
  FD79A8",  
  p = "blue", q = "#6C5CE7", r = "#00B894", s = "#0984E3", t = "#  
  E17055",  
  u = "#A29BFE", v = "#55EFC4", w = "green", x = "pink", y = "orange  
 ",  
  z = "#636E72"  
)
```

Yes, you can select your own colours!

The Visualization Code

```
1 text_art <- ggplot() +
2   geom_point(data = title_data, aes(x = col, y = -row, color =
3     character),
4             size = 11, shape = 15) + #change shape and size to see
5               how the art changes
6   geom_point(data = text_data, aes(x = col, y = -row, color =
7     character),
8             size = 7.5, shape = 15) +
9   scale_color_manual(values = letter_colors) +
10  coord_equal() +
11  theme_void() +
12  theme(
13    legend.position = "none",
14    plot.margin = margin(20, 10, 20, 20)
15  )
16
17 print(text_art)
```

Example Outputs



Each color represents a letter. Patterns emerge from word frequency and structure!

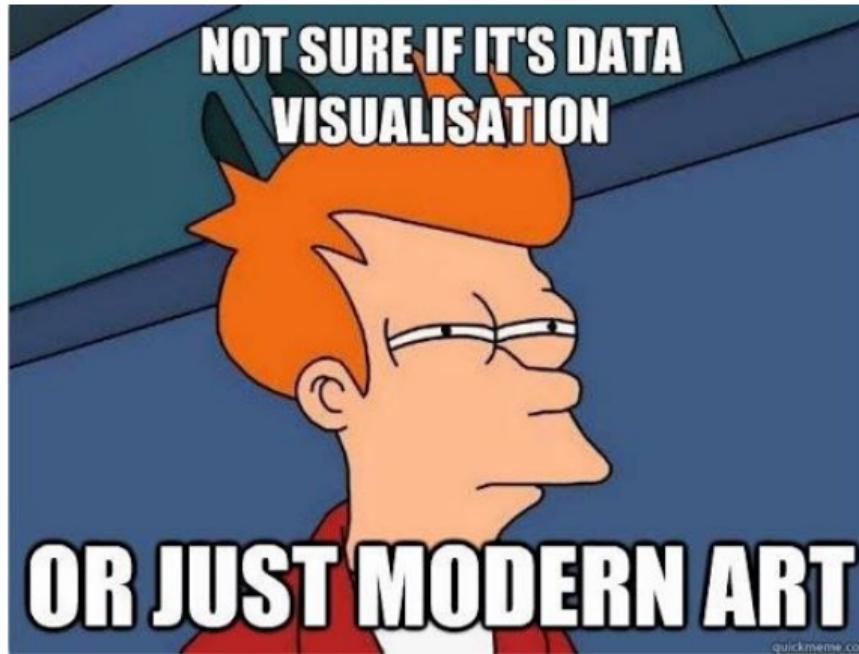
Saving Your Masterpiece

```
ggsave("i_must_betray_you.png", text_art, width = 12, height = 14,  
       dpi = 300, bg = "white")
```

Print Specifications:

- **300 DPI** for professional quality
- **Large dimensions** for wall art ($12'' \times 14''$ or larger)
- **White background** prints better than transparent

Let's dig in!



Thank You!

Questions?

Contact: khushi@uchicago.edu