Advanced R

PSYC 2020-A01 / PSYC 6022-A01 | 2025-10-24 | Lab 10

Jessica Helmer

Outline

- Assignment 9 Review
- Communication with Visualizations
- Communication with Quarto

Learning objectives:
R: Plot annotations

Mid-Course Survey

- Scratch code
- Mid-class breaks

Assignment 9 Review

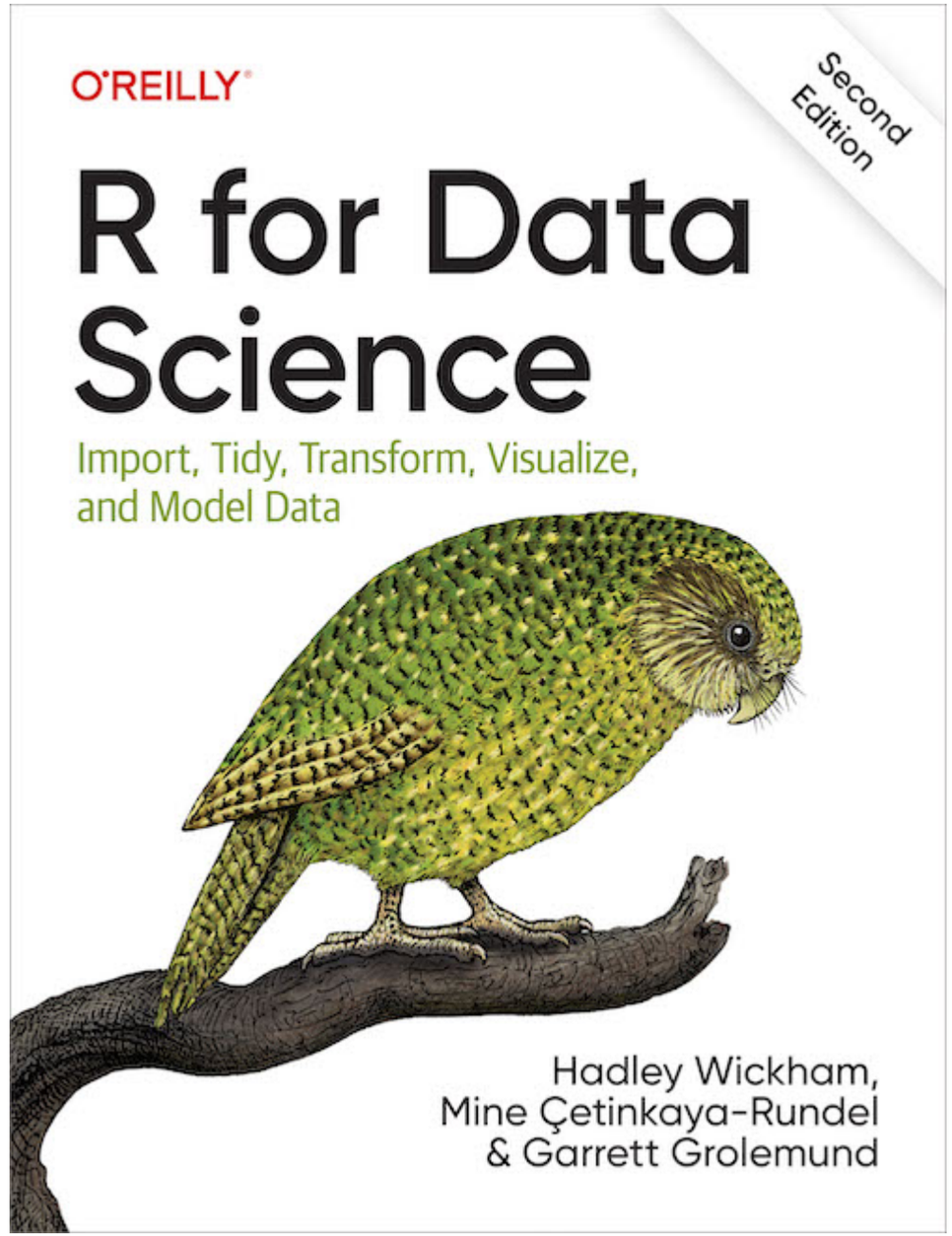
Great work!

Data Communication

Communication

Going to once again heavily lean on this book!

Feel free to reference for more R content



https://r4ds.hadley.nz/

Communication

Exploratory Data Analysis

- Audience: self
- Goal: understanding / exploring data
- You already know what's in the plot

Data Communication

- Audience: others
- Goal: communicating and showing data. Make as self-explanatory as possible
- Won't already know what's in the plot

Exploratory to "expository" (intended to explain or describe something)

Today's Data

Mostly using the fuel economy `mpg` dataset that comes with the `tidyverse` library today

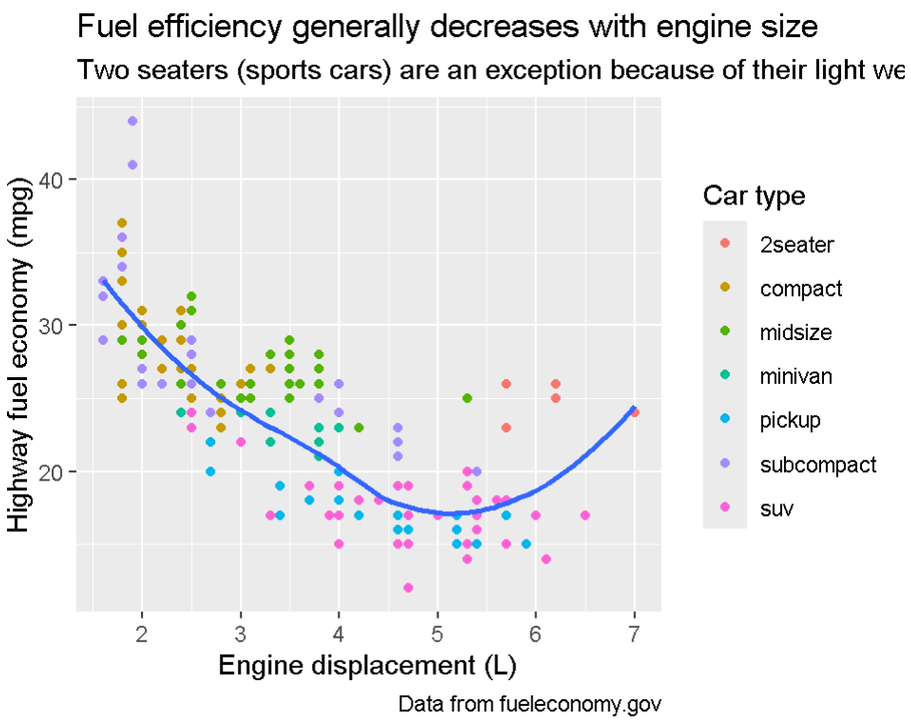
```
1 head(mpg, 10)
```

```
# A tibble: 10 × 11
   manufacturer model      displ  year   cyl trans drv     cty   hwy fl    class
   <chr>        <chr>      <dbl> <int> <int> <chr> <chr> <int> <int> <chr> <chr>
 1 audi         a4           1.8  1999     4 auto… f        18    29 p     comp…
 2 audi         a4           1.8  1999     4 manu… f        21    29 p     comp…
 3 audi         a4           2    2008     4 manu… f        20    31 p     comp…
 4 audi         a4           2    2008     4 auto… f        21    30 p     comp…
 5 audi         a4           2.8  1999     6 auto… f        16    26 p     comp…
 6 audi         a4           2.8  1999     6 manu… f        18    26 p     comp…
 7 audi         a4           3.1  2008     6 auto… f        18    27 p     comp…
 8 audi         a4 quattro   1.8  1999     4 manu… 4        18    26 p     comp…
 9 audi         a4 quattro   1.8  1999     4 auto… 4        16    25 p     comp…
10 audi         a4 quattro   2    2008     4 manu… 4        20    28 p     comp…
```

Labels

Easiest place to start with making visualizations more self-explanatory

Plot | Code



Goal of title is to explain the main finding (not just “a scatterplot of engine displacement vs. fuel economy”)

Good to include units

Annotations

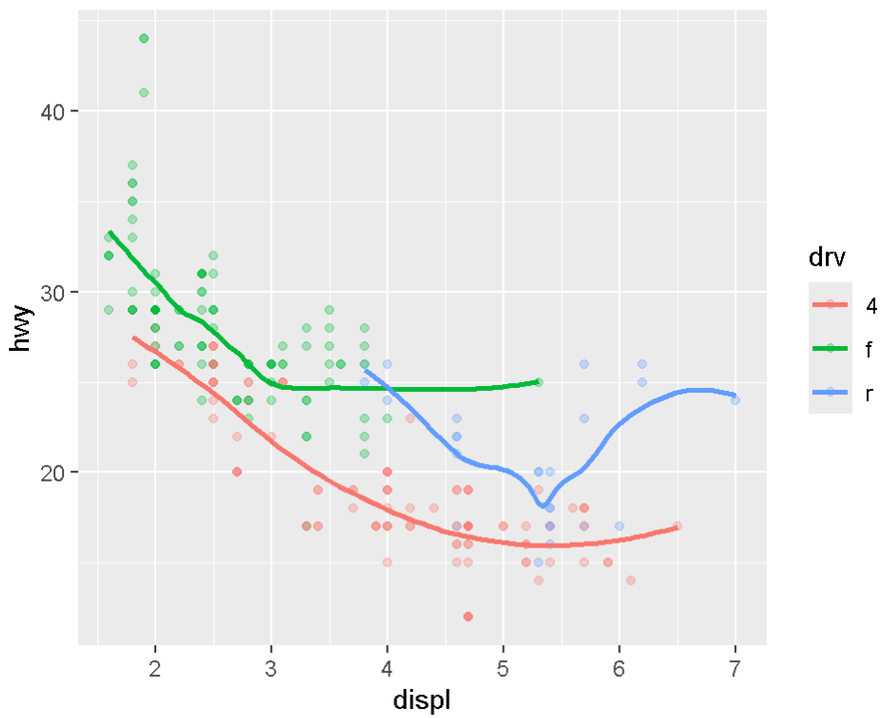
Label individual points or groups of points

`geom_text()` is like `geom_point()`, except it displays text from an `label` argument within the `aes()` function (still also needs `x` and `y` mappings)

Starting with this…

And with the goal of labeling these lines directly

Plot | Code



Annotations

Can start by making a dataframe of labels. Example goal: label by type of drive (front-wheel, rear-wheel, four-wheel)

```
mpg_labels <- mpg |>
  summarize(.by = drv,
            displ = mean(displ),
            hwy = mean(hwy)) |>
  mutate(drive_type = case_when(drv == "4" ~ "four-wheel drive",
                                drv == "f" ~ "front-wheel drive",
                                drv == "r" ~ "rear-wheel drive"))
mpg_labels
```

```
# A tibble: 3 × 4
  drv   displ   hwy drive_type
  <chr> <dbl> <dbl> <chr>
1 f      2.56  28.2 front-wheel drive
2 4      4.00  19.2 four-wheel drive
3 r      5.18  21   rear-wheel drive
```

`case_when()` Reference

Use the `case_when()` function to change the contents of a column based on some condition

`case_when(logical ~ new_value)`

```
grades <- rnorm(10, 75, 10)
grades
```

```
 [1] 57.18712 84.68568 80.48602 74.16250
62.41695 77.29268 72.82202 71.54923
 [9] 67.99412 66.01027
```

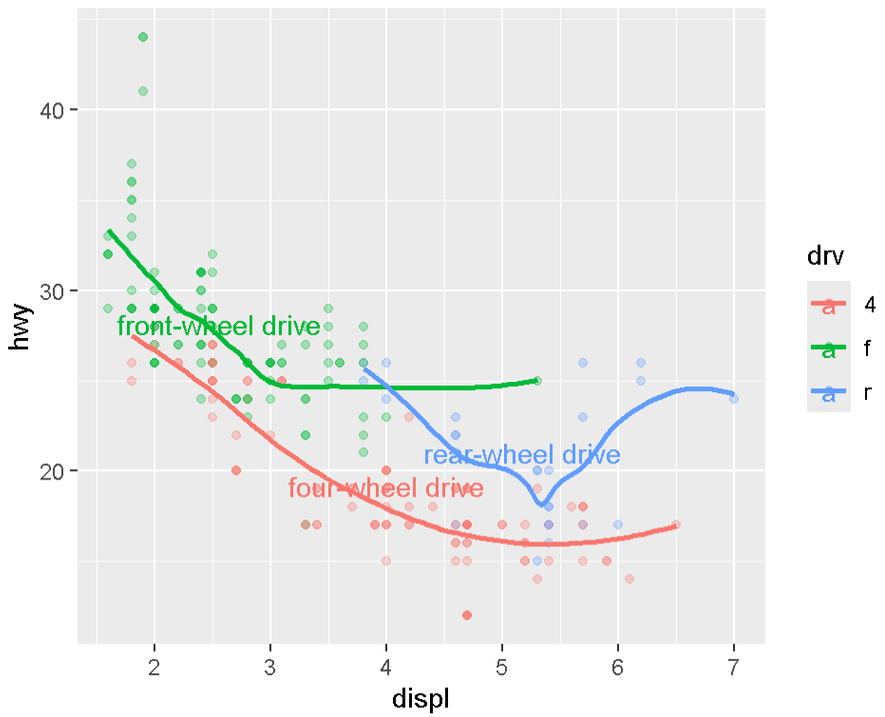
```
case_when(grades < 60 ~ "F",
          grades < 70 ~ "D",
          grades < 80 ~ "C",
          grades < 90 ~ "B",
          grades <= 100 ~ "A",
          .default = NA)
```

```
[1] "F" "B" "B" "C" "D" "C" "C" "C" "D" "D"
```

Annotations

Then, we can use those labels in the `data` argument for a `geom_text()`

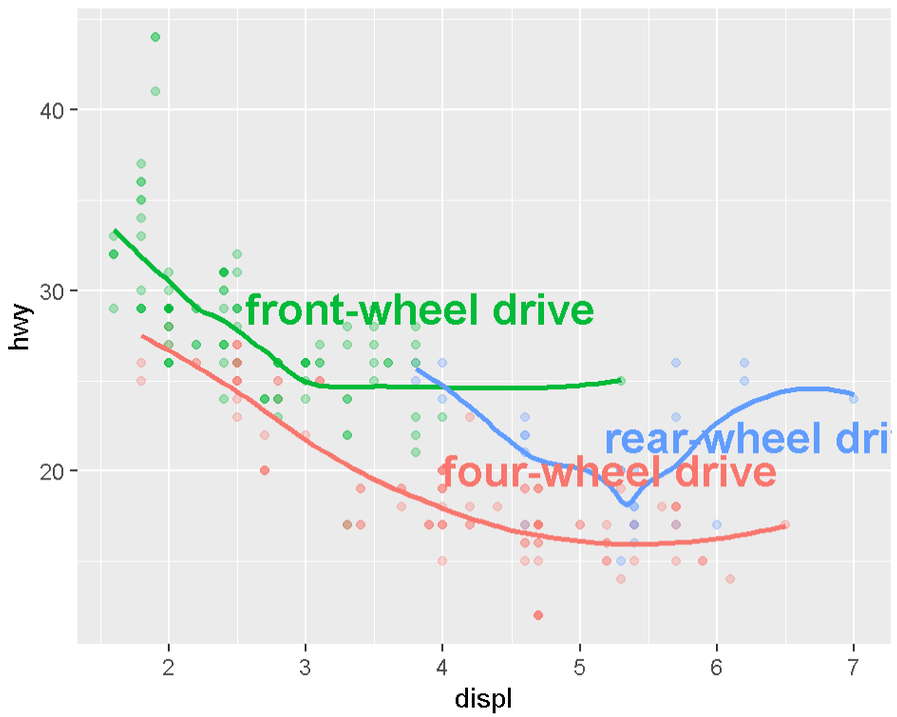
Plot | Code



Annotations

Some ways to make this look nicer…but still overlap

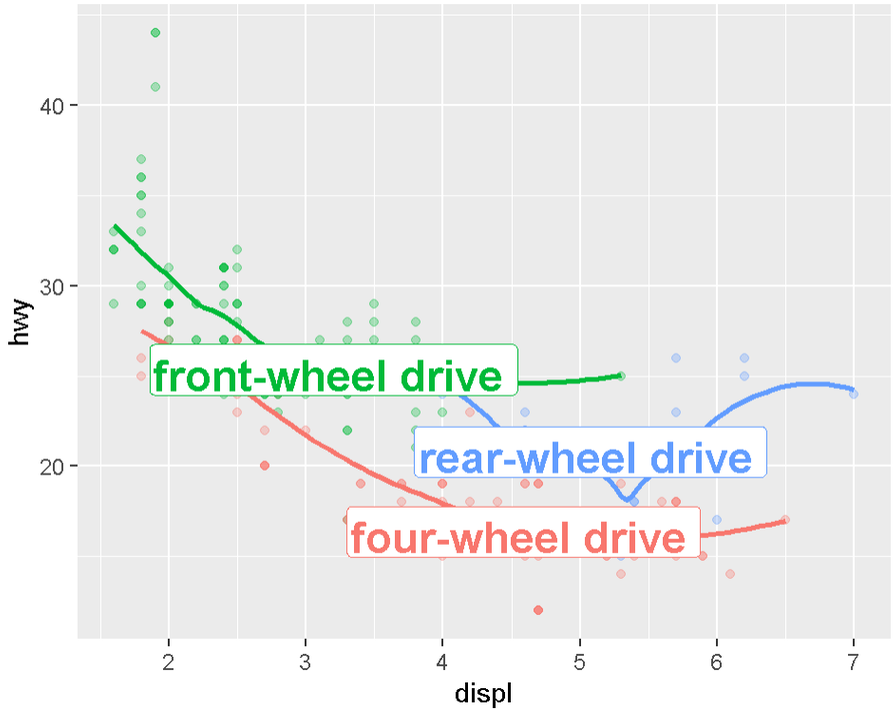
Plot | Code



Annotations: `ggrepel`

Can use `geom_label_repel()` function from the `ggrepel` package to help with this

Plot | Code


40
30
20
hwy
front-wheel drive
rear-wheel drive
four-wheel drive
2
3
4
5
6
7
displ

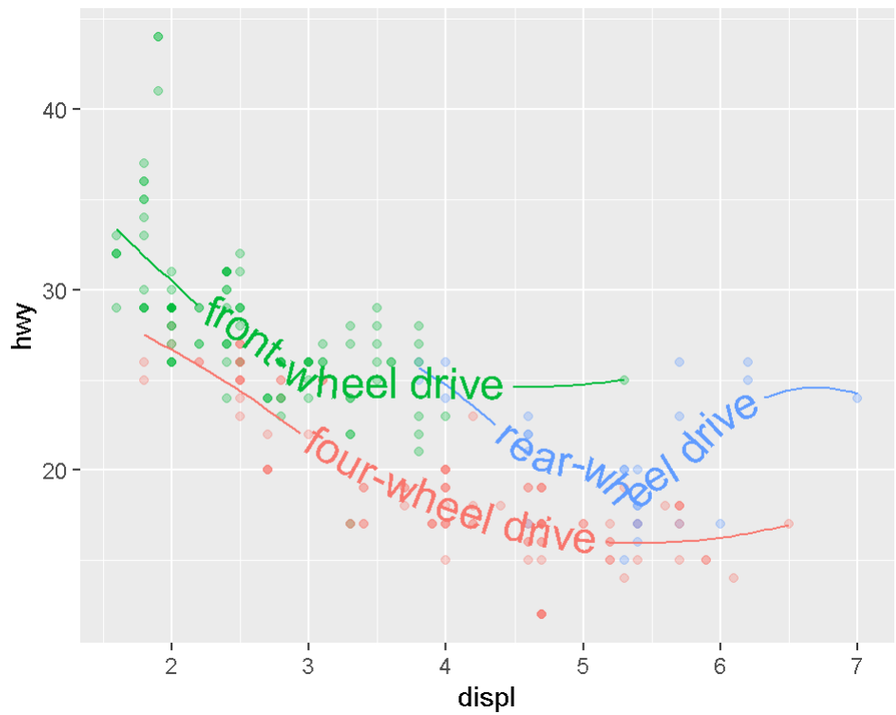
Annotations: `geomtextpath`

Or, why make your audience have to look back and forth between labels and lines?

Enter the `geomtextpath` package with its `geom_textsmooth()`.

Adds label directly onto the line (doesn't need its own `x` and `y` mappings).

Plot | Code



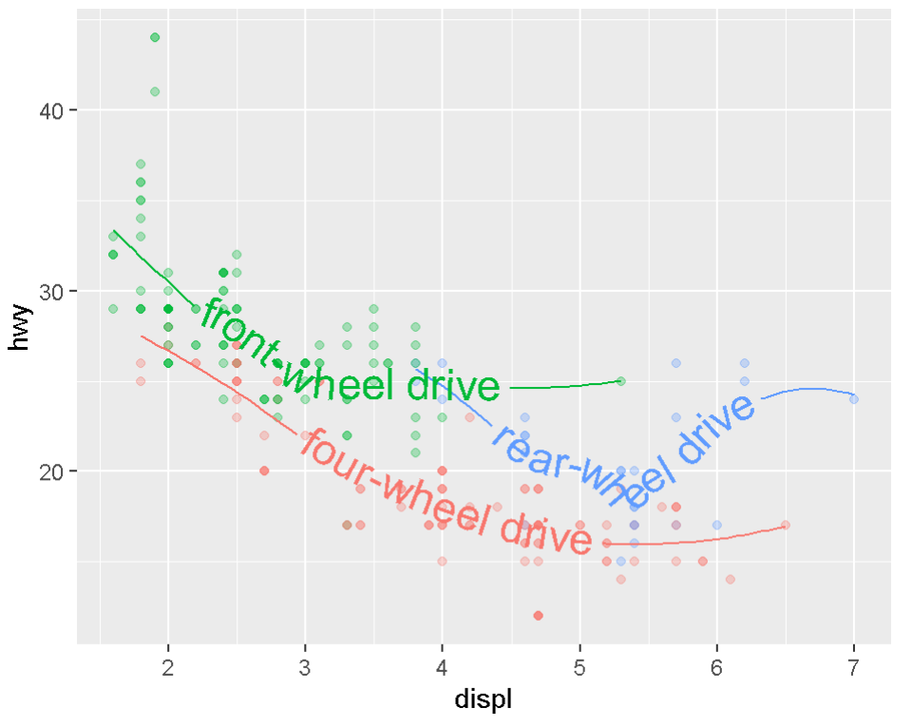
Annotations: `geomtextpath`

The `geomtextpath` package has geom functions for most aesthetics, with three changes:

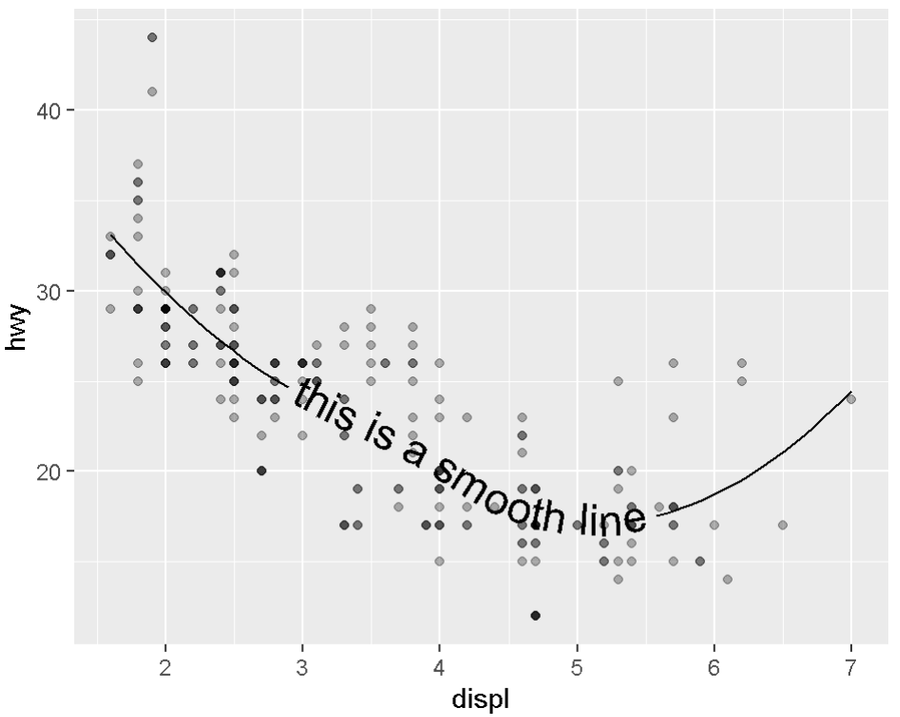
1. Either call `library(geomtextpath)` or specifically call the function with `geomtextpath::` (my preference)
2. Add `text` or `label` before the geom name (e.g., `geom_textline()`, `geom_labelsmooth()`)
3. Give it a `label` mapping (if label from the data) or `label` argument (if just some text)

Annotations: `geom_textpath`

Plot | Code



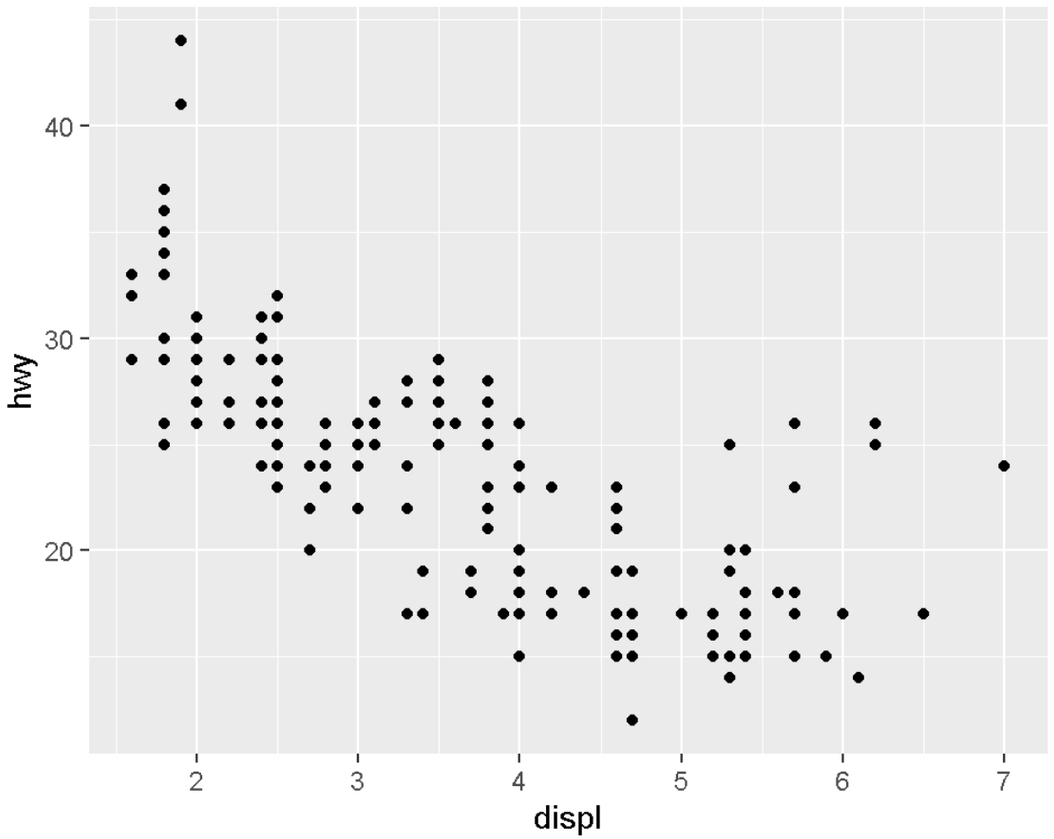
Plot | Code



Annotations: Identifying Outliers

Let's say we want to display the models of the cars that are more outliers here.

```
1  ggplot(mpg, aes(x = displ, y = hwy)) +
2    geom_point()
```



Annotations: Identifying Outliers

First, let's make a dataframe of just the outliers

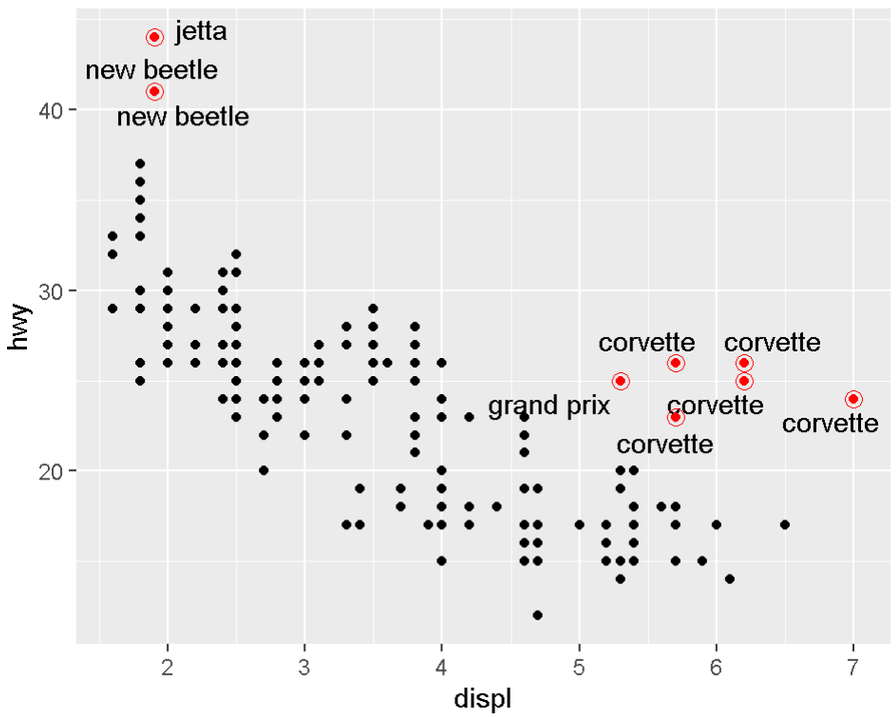
```
potential_outliers <- mpg |>
  filter(hwy > 40 | (hwy > 20 & displ > 5))
potential_outliers
```

```
# A tibble: 9 × 11
  manufacturer model      displ  year   cyl trans  drv     cty   hwy fl    class
  <chr>        <chr>      <dbl> <int> <int> <chr>  <chr> <int> <int> <chr> <chr>
1 chevrolet    corvette     5.7  1999     8 manua… r        16    26 p     2sea…
2 chevrolet    corvette     5.7  1999     8 auto(… r        15    23 p     2sea…
3 chevrolet    corvette     6.2  2008     8 manua… r        16    26 p     2sea…
4 chevrolet    corvette     6.2  2008     8 auto(… r        15    25 p     2sea…
5 chevrolet    corvette     7    2008     8 manua… r        15    24 p     2sea…
6 pontiac      grand prix   5.3  2008     8 auto(… f        16    25 p     mids…
7 volkswagen   jetta        1.9  1999     4 manua… f        33    44 d     comp…
8 volkswagen   new beetle   1.9  1999     4 manua… f        35    44 d     subc…
9 volkswagen   new beetle   1.9  1999     4 auto(… f        29    41 d     subc…
```

Annotations: Identifying Outliers

Now, we can layer them onto the plot

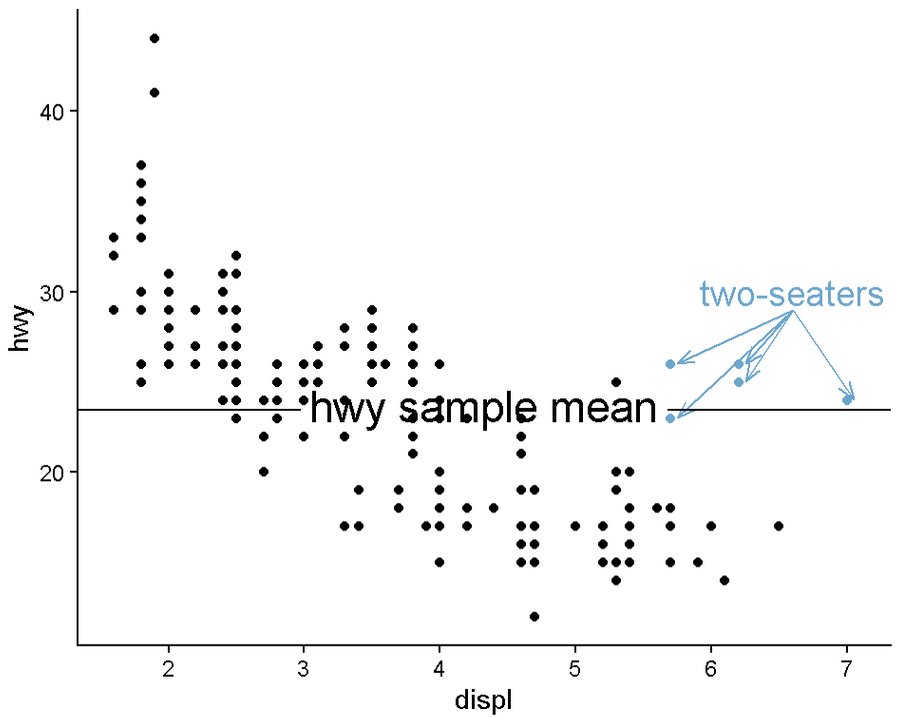
Plot | Code



Note the additional `geom_point()` layer with the `"circle open"` shape to add some extra emphasis.

Annotations: Some other useful geoms

Plot | Code



Annotate: `annotate()`

Helpful for adding bits of information or explanatory text to a plot (not mapped from or a subset of data)

- Make the main point of the visualization more immediately obvious

Can start with some text:

```
1 trend_text <- "Larger engine sizes tend to have lower fuel economy." |>
2   str_wrap(width = 30)
3 trend_text
```

```
[1] "Larger engine sizes tend to\nhave lower fuel economy."
```

`str_wrap(string)` function breaks text into multiple lines

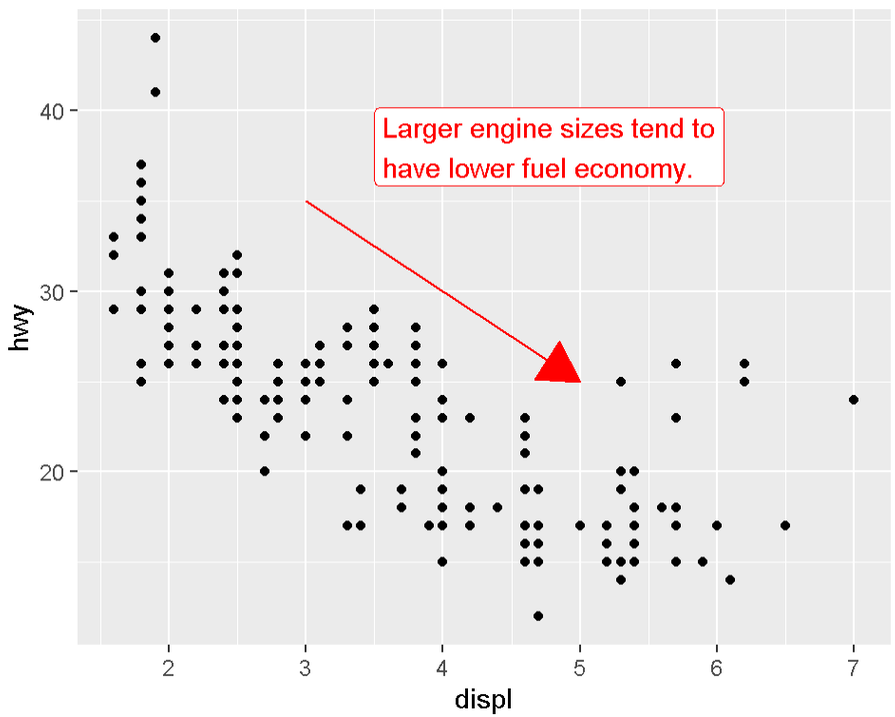
- `string` = string you want to wrap
- `width` = target line width (in number of characters)

`\n` represents a "new line" character (like a line break)

Annotate: `annotate()`

`annotate()` function looks for all the required aesthetics for the selected geom.

Plot | Code



Quarto

Quarto

Framework for combining code, results, and writing.

> Quarto files are designed to be used in three ways:

> For communicating to decision-makers, who want to focus on the conclusions, not the code behind the analysis.

> For collaborating with other data scientists (including future you!), who are interested in both your conclusions, and how you reached them (i.e. the code).

> As an environment in which to do data science, as a modern-day lab notebook where you can capture not only what you did, but also what you were thinking.

Quarto vs. RMarkdown

Like RMarkdown, but…better!

RMarkdown had many extensions to make books, presentations, etc. This unifies all of them.

> In a way, Quarto reflects everything that was learned from expanding and supporting the R Markdown ecosystem over a decade.

Quarto uses the Quarto Command Line Interface (CLI), but RStudio automatically installs and loads it when needed.

NOT an R package. If we want help, we refer to the Quarto Documentation.

Quarto Gallery

Some Quarto possibilities…

Quarto Gallery

Also…my slides and the course website.

Quarto Files

````
---
title: "Diamond sizes"
date: 2022-09-12
format: html
---

```{r}
#| label: setup
#| include: false

library(tidyverse)

smaller <- diamonds |>
  filter(carat <= 2.5)
```

We have data about `r nrow(diamonds)` diamonds.
Only `r nrow(diamonds) - nrow(smaller)` are larger than 2.5 carats.
The distribution of the remainder is shown below:

```{r}
#| label: plot-smaller-diamonds
#| echo: false

smaller |>
  ggplot(aes(x = carat)) +
  geom_freqpoly(binwidth = 0.01)
```
````

Just a text file with extension `.qmd`, contains…

1. An (optional) **YAML header** surrounded by `---`s.
2. **Chunks** of R code surrounded by ```` ``` ````.
3. Text mixed with simple text formatting like `# heading` and `_italics_`.

Quarto Documents, Presentations, Websites, oh my!

Many different formats of outputs and purposes.

All have code, figures, etc. all coming from the same place.

No need for copying and pasting, figure output updates when the code does.

Great way of documenting your work and sharing with others.

Quarto Documents, Presentations, Websites, oh my!

Check out the R4DS book and Quarto Documentation for more learning.

Can pair with GitHub to create easily shareable, version-controlled reports.

- See *Happy Git and GitHub for the useR* for more (how I learned!)

Assignment 10