R programming for Natural Resource Professionals

Lecture 9:

Data Visualization I:

Intro to ggplot2

What is ggplot2?

- Developed by Hadley Wickham
- 10+ years old
- Philosophy emphasizes readability
- Over 80 extensions
- gg = “grammar of graphics”
  - Standardized phrasing for graphical plotting elements

ggplot2: function structures

```
ggplot(data = <DATA>, mapping = aes(<MAPPINGS>)) +
     <GEOM_FUNCTION>()
```

Or

```
<DATA> |>
     ggplot(mapping = aes(<MAPPINGS>)) +
     <GEOM_FUNCTION>()
```

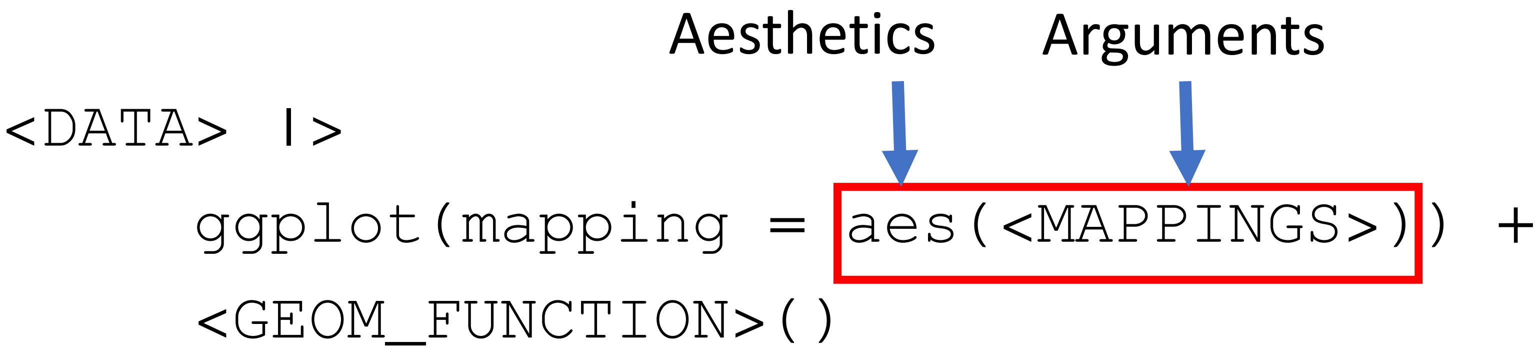
ggplot2: function structures

Aesthetics Arguments

```
<DATA> |>
        ggplot(mapping = aes(<MAPPINGS>)) +
        <GEOM_FUNCTION>()
```

x and y are specified within the `aes()` argument: `aes(x = xValues, y = yValues)`

ggplot2: function structures


Aesthetics

Arguments

```
<DATA> |>
     ggplot(mapping = aes(<MAPPINGS>)) +
     <GEOM_FUNCTION>()
```

Aesthetics arguments are responsive to data variables (e.g., color changes with value)

ggplot2: function structures

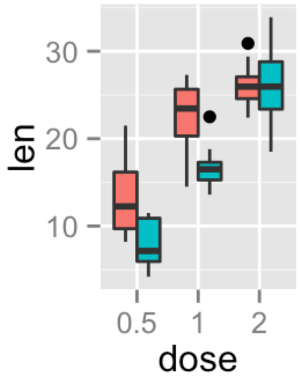
```
<DATA> |>
     ggplot(mapping = aes(<MAPPINGS>)) +
     <GEOM_FUNCTION>(non-aes() arguments))
```

`geom_...()` arguments placed outside of `aes()` are fixed values that are unresponsive to variables in the data

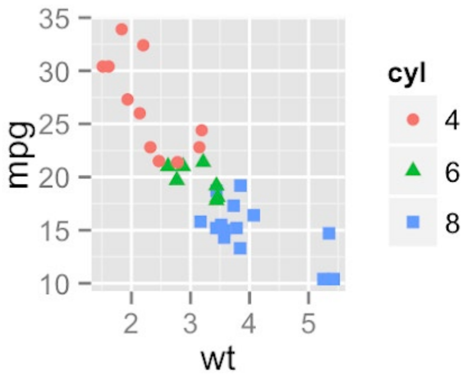
ggplot2: function structures

```
<DATA> |>
     ggplot(mapping = aes(<MAPPINGS>)) +
     <GEOM_FUNCTION>()
```

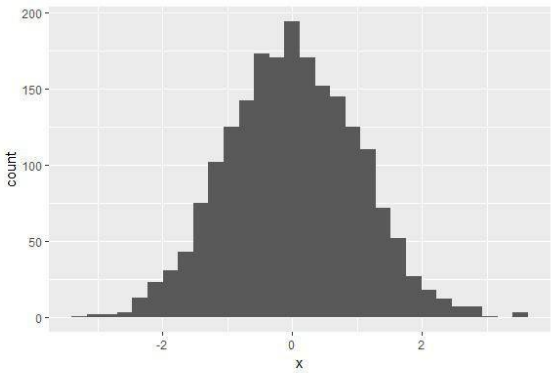
`geom_boxplot()`



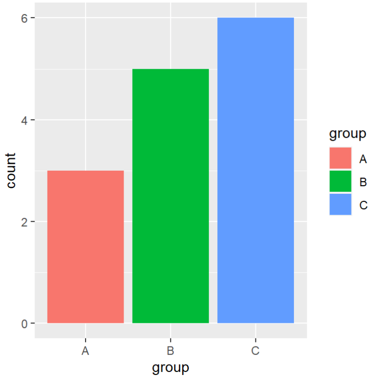
`geom_point()`



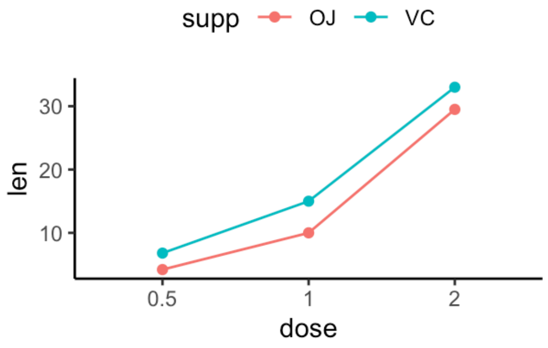
`geom_histogram()`



`geom_bar()`

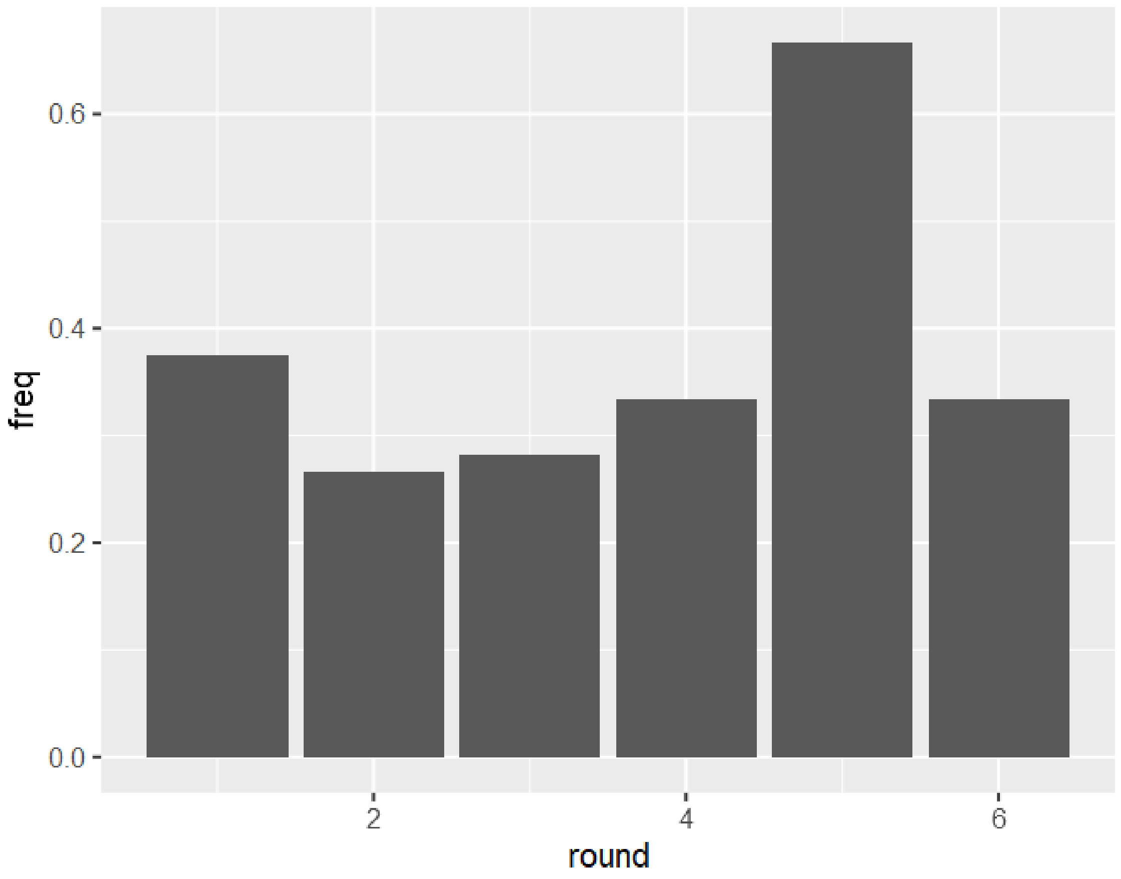


`geom_line()`

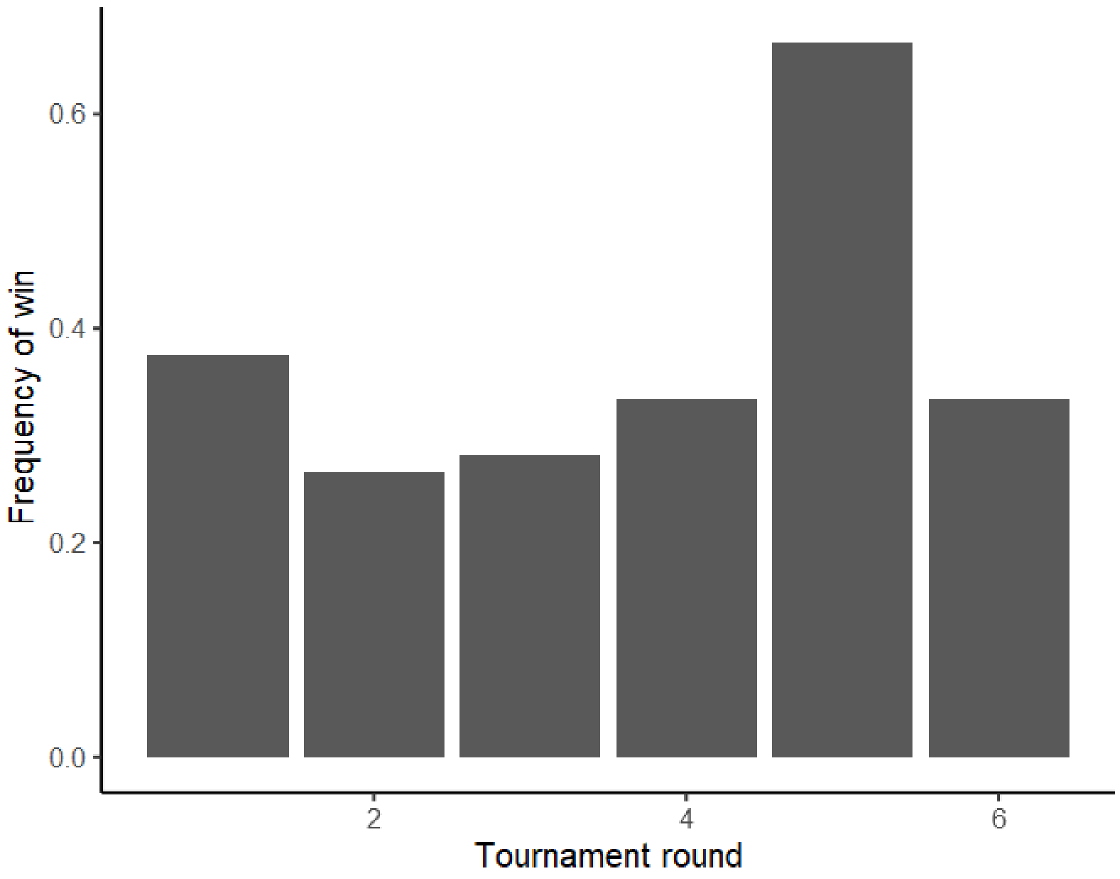


- “Publication quality” figures are the standard for grading
- What does “publication quality” mean?
    - Adequate resolution/size
    - Clearly axis labels
    - Appropriate captions that stand alone
    - Appropriate font sizes
    - Appropriate use of color/contrast

Not polished



Polished



ggplot2 plot polishing

`ggtitle()`: add a title and subtitle

```
<DATA> |>
     ggplot(mapping = aes(<MAPPINGS>)) +
     <GEOM_FUNCTION>() +
     ggtitle(title = “title”,
             subtitle = “subtitle”)
```

ggplot2 plot polishing

`xlab()`: customize x axis label

`ylab()`: customize y axis label

```
<DATA> |>
     ggplot(mapping = aes(<MAPPINGS>)) +
     <GEOM_FUNCTION>() +
     xlab(“label text”) +
     ylab(“label text”)
```

ggplot2 plot polishing

`geom_smooth()`: add trend lines; see arguments

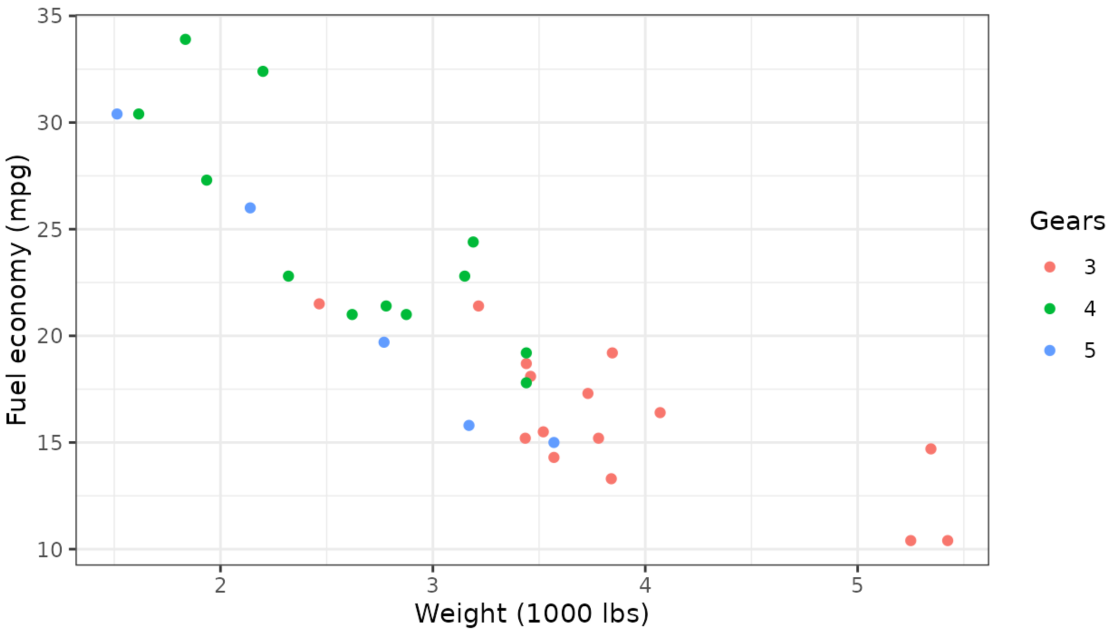
```
<DATA> |>
     ggplot(mapping = aes(<MAPPINGS>)) +
     <GEOM_FUNCTION>() +
     geom_smooth(method = “lm”)
```

ggplot2 plot polishing

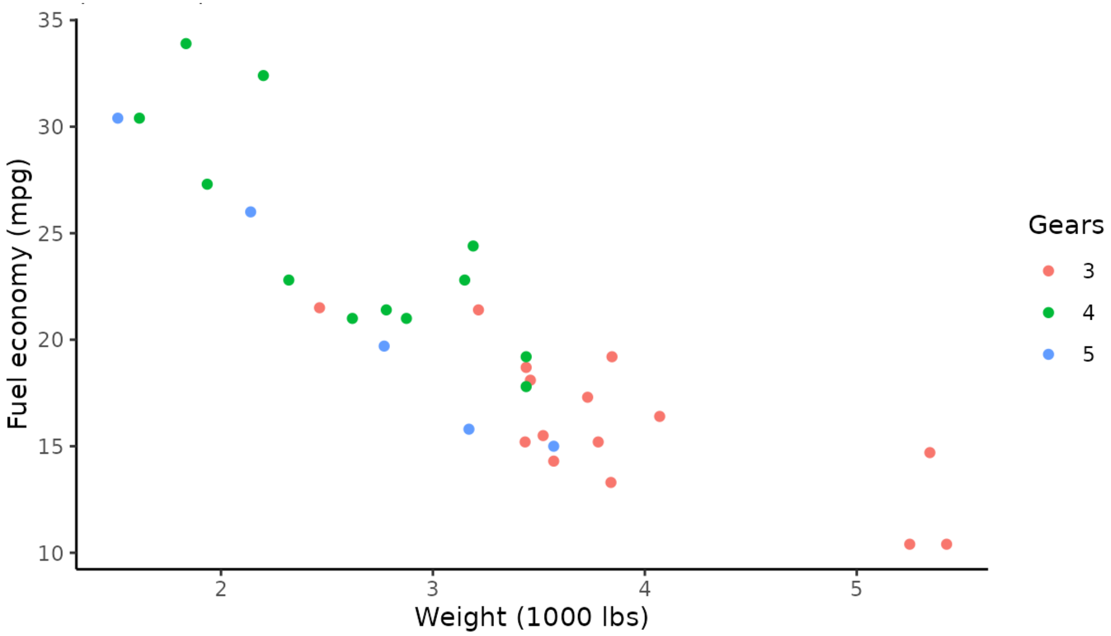
`theme_*()`: Reformat visualization based on packaged settings

See all options: https://ggplot2.tidyverse.org/reference/ggtheme.html

`theme_bw()`



`theme_classic()`



ggplot2 plot polishing

```
theme_update()

plot + theme()
```

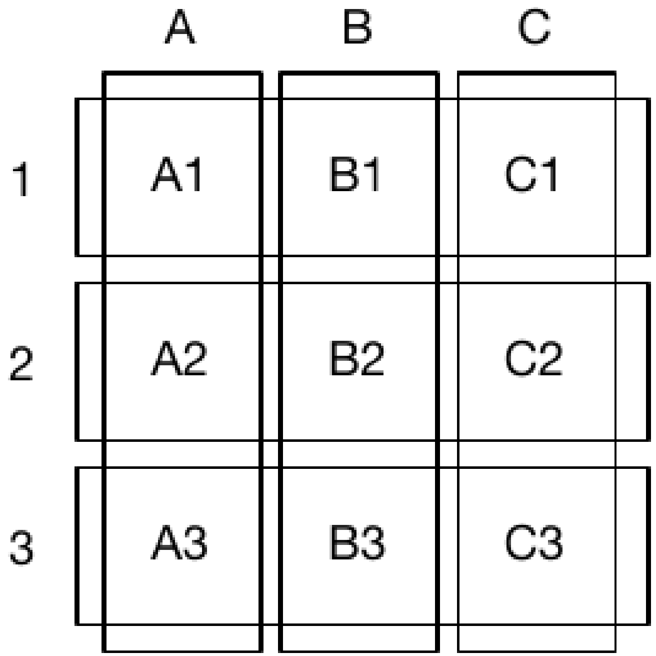
```
theme(line, rect, text, title, aspect.ratio, axis.title, axis.title.x, axis.title.x.top, axis.title.x.bottom,
axis.title.y, axis.title.y.left,axis.title.y.right, axis.text, axis.text.x, axis.text.x.top,
axis.text.x.bottom, axis.text.y, axis.text.y.left, axis.text.y.right, axis.ticks, axis.ticks.x,
axis.ticks.x.top, axis.ticks.x.bottom, axis.ticks.y, axis.ticks.y.left, axis.ticks.y.right, axis.ticks.length,
axis.line, axis.line.x, axis.line.x.top, axis.line.x.bottom, axis.line.y, axis.line.y.left, axis.line.y.right,
legend.background, legend.margin, legend.spacing, legend.spacing.x, legend.spacing.y, legend.key,
legend.key.size, legend.key.height, legend.key.width, legend.text, legend.text.align, legend.title,
legend.title.align, legend.position, legend.direction, legend.justification, legend.box, legend.box.just,
legend.box.margin, legend.box.background, legend.box.spacing, panel.background, panel.border, panel.spacing,
panel.spacing.x, panel.spacing.y, panel.grid, panel.grid.major, panel.grid.minor, panel.grid.major.x,
panel.grid.major.y, panel.grid.minor.x, panel.grid.minor.y, panel.ontop, plot.background, plot.title,
plot.subtitle, plot.caption, plot.tag, plot.tag.position, plot.margin, strip.background, strip.background.x,
strip.background.y, strip.placement, strip.text, strip.text.x, strip.text.y, strip.switch.pad.grid,
strip.switch.pad.wrap)
```

https://ggplot2.tidyverse.org/reference/theme.html

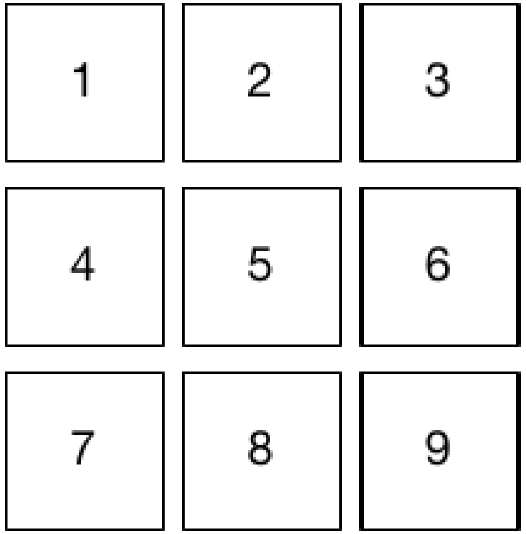
ggplot2 plot polishing

`facet_wrap()`: 1-dimension plot set arranged into a grid

`facet_grid()`: 2-dimensional plot set



facet_grid



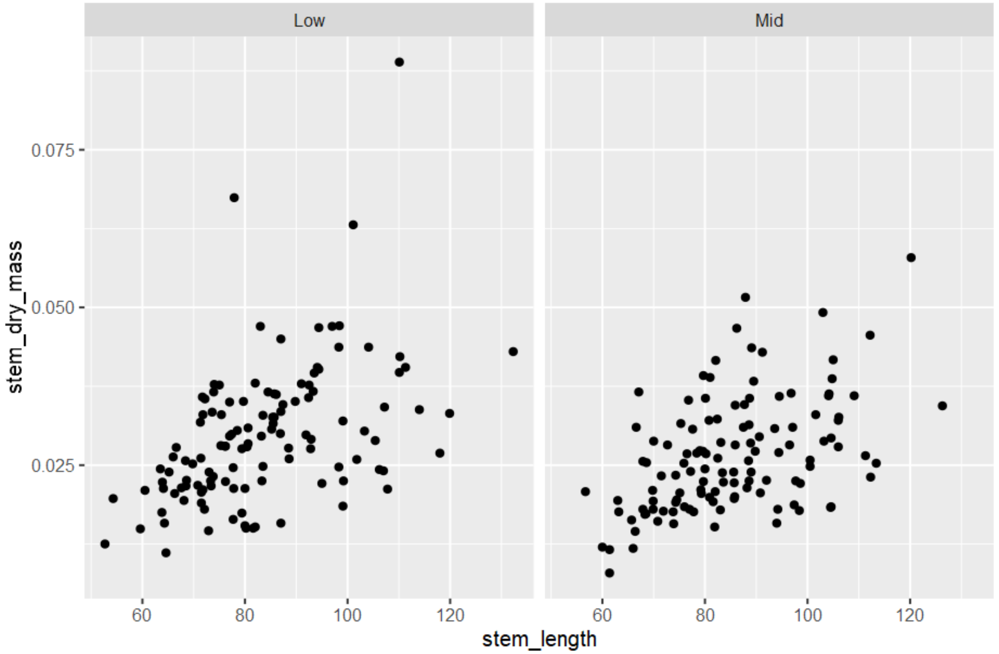
facet_wrap

ggplot2 plot polishing

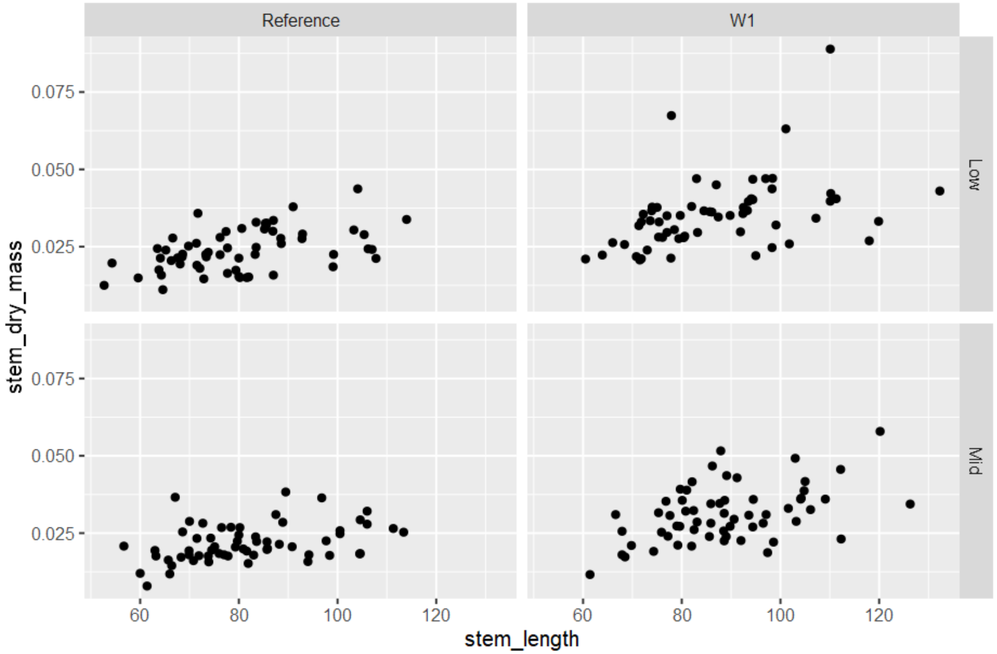
`facet_wrap()`: 1-dimension plot set arranged into a grid

`facet_grid()`: 2-dimensional plot set

`facet_wrap()`



`theme_grid()`



ggplot2 plot polishing

`facet_wrap()`: 1-dimension plot set arranged into a grid

`facet_grid()`: 2-dimensional plot set

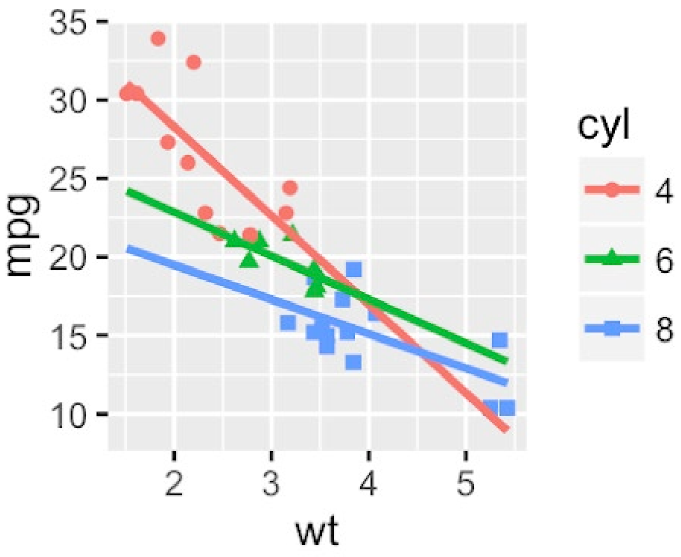
Adjusting scales: argument is `scales()` and options are “fixed” (default), “free_x,” “free_y,” or “free”

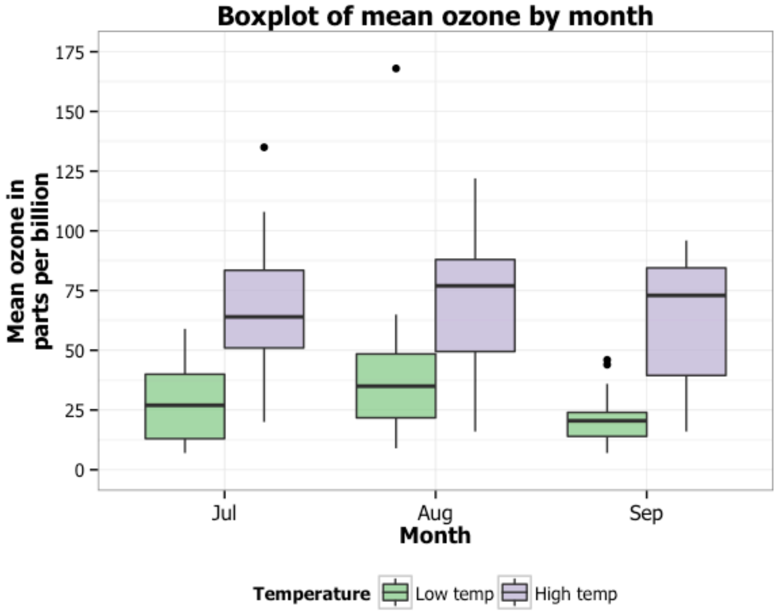
ggplot2 plot polishing

Using color:

Some plot styles require “color” arguments and some require “fill” arguments.

```
<DATA> |>
      ggplot() +
      <GEOM_FUNCTION>(mapping = aes(x = var1, y = var2, color = var3))
```





ggplot2 plot polishing

Using color:

```
scale_color_manual(values=c(”red”,”black”))
scale_fill_manual(values=c(”red”,”black”))
```

ggplot2 plotting in Rmarkdown

~~~
```{r, fig.align = ‘center’}
plot(cars)
```
~~~

Other options

```
fig.cap = ’Figure 1...’
fig.height = 4
fig.width = 4
```

And more…

ggplot2 plot saving

```
ggsave()
```

Arguments: file name including extension and path if needed, dimensions, units, dpi, …

```
ggsave(”mtcars.pdf”, width = 4, height = 4)
ggsave(”mtcars.pdf”, width = 20, height = 20, units = ”cm”)
```

Data visualization best practices

- Figures (and captions) should stand alone
  - Tables do not need to do this
- Figures convey the message of the paper
- People will grab figures for their own purposes

  (example here)

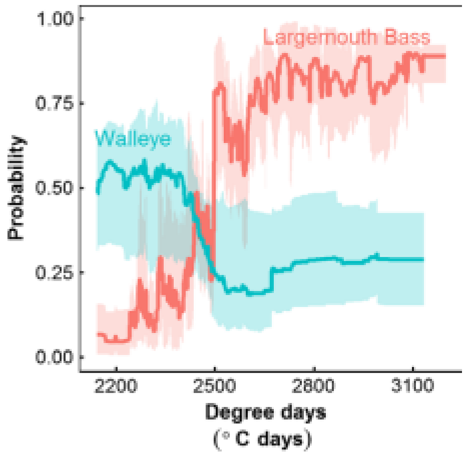
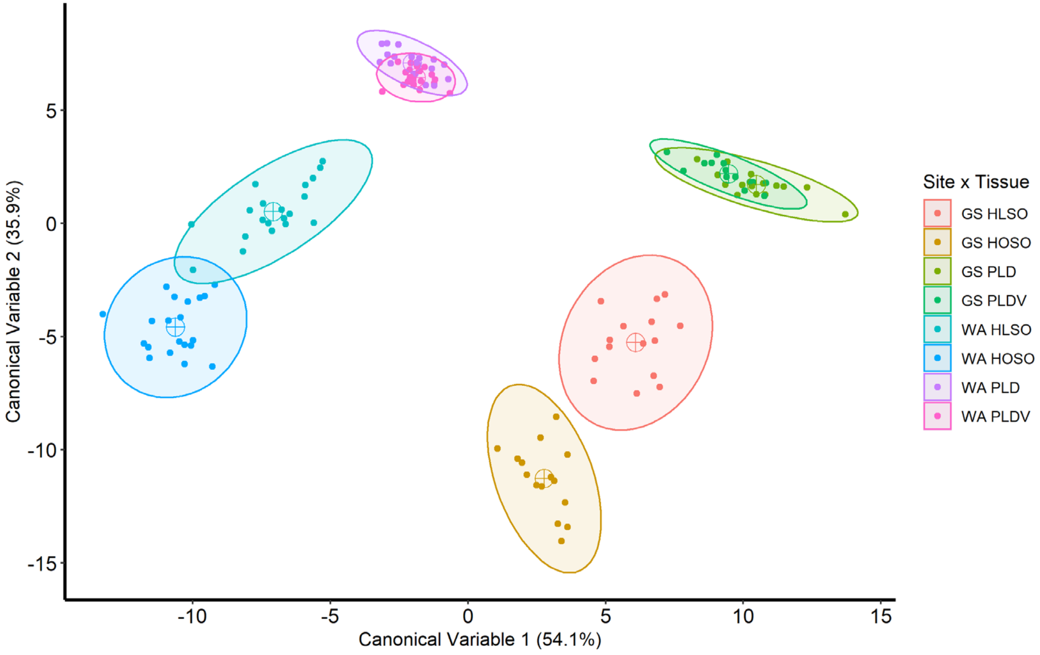


Fig. 1 Predicted probability of successful walleye recruitment (electrofishing catch rates ≥10 age-0 fish per mile; blue line) and high largemouth bass relative abundance (electrofishing catch rates ≥ season-specific median catch rates; orange line) as a function of mean water temperature degree days (base temperature 5 °C) in contemporary period (1989–2014). Predicted probabilities are based on species-specific random forest models with additional predictors: lake area, conductivity, and shoreline complexity for walleye and lake order and Secchi depth for largemouth bass. Solid lines show median probability for a given value of degree days across all possible combinations of other predictors; shaded ribbon shows 25th–75th percentile.

Data visualization best practices

Color blind friendly palettes

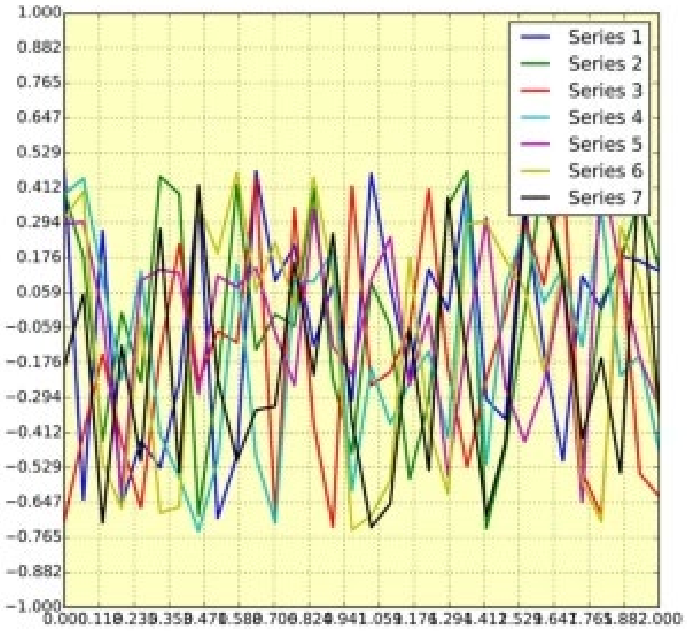
- https://www.color-blindness.com/coblis-color-blindness-simulator/

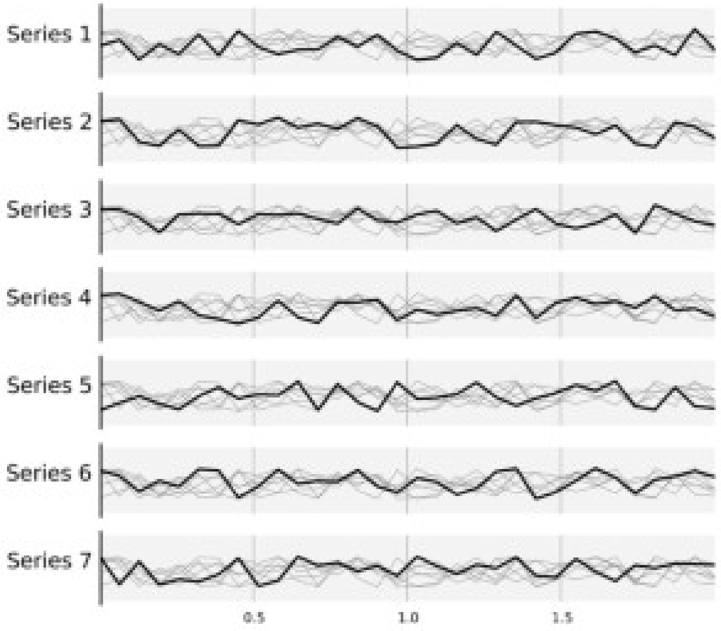


Data visualization best practices

Excessive complexity

- Complex figures that nobody can understand

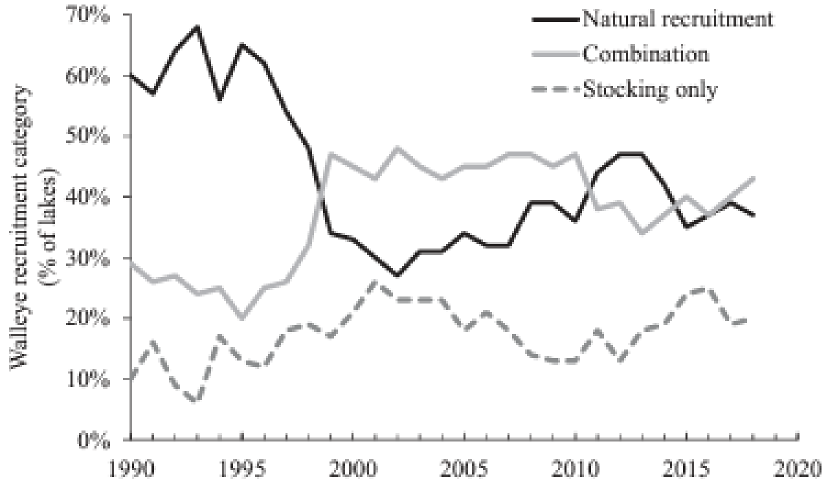




Data visualization best practices

Avoid double-axis figures

- Double axis graphs



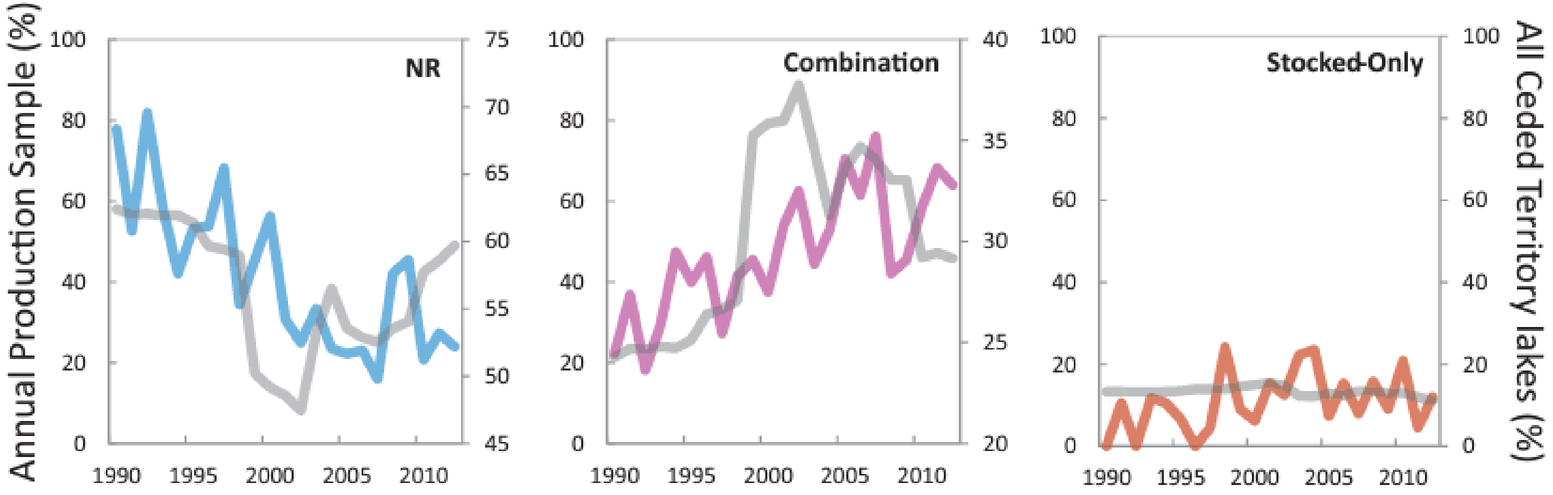
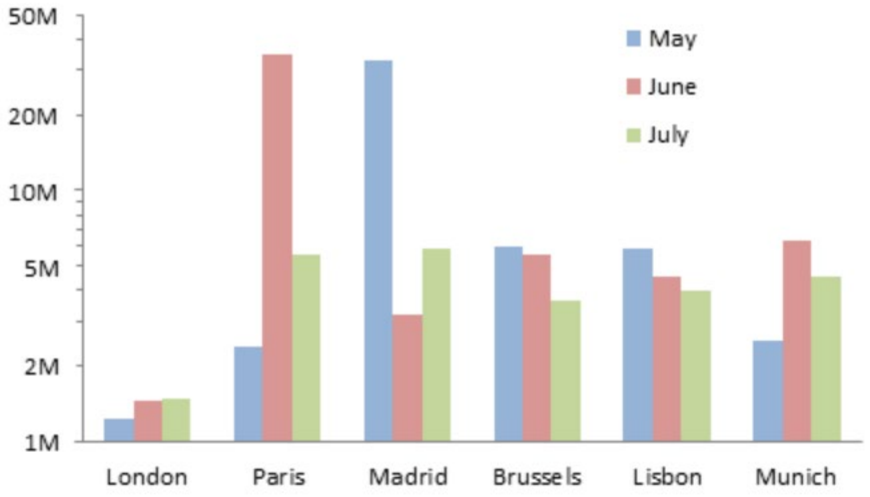
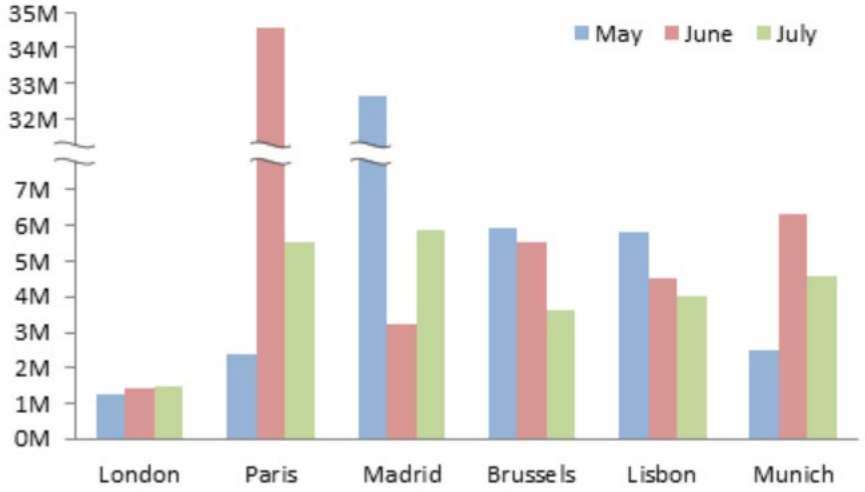


Fig. 4. Proportion of walleye (*Sander vitreus*) lakes classified into natural reproduction (NR), combination, and stocked recruitment categories. Walleye lakes across the entirety of the Wisconsin Ceded Territory are plotted on the secondary *y* axis as a dashed gray line. Production lakes (i.e., lakes where data were available for production calculations) are plotted on the primary *y* axis in color. [Colour online.]

Data visualization best practices

Avoid misleading axes


50M
20M
10M
5M
2M
1M
May
June
July
London
Paris
Madrid
Brussels
Lisbon
Munich


35M
34M
33M
32M
7M
6M
5M
4M
3M
2M
1M
0M
May
June
July
London
Paris
Madrid
Brussels
Lisbon
Munich