

The big picture

Remember that the final goal is to have a GitHub repository where you can showcase your work. Assignment 1 was to create the repository. Assignment two required you to develop one R script to clean some data in that same repository. **For this third assignment, you will visualize the data you cleaned last week.** Your fourth assignment will require you to work with spatial data. Your final project will leverage the data and visualizations you'll produce to wrap it all together.

This assignment

Task: Develop one data visualization script that reads the clean data you exported last week, visualizes them, and exports 1-2 figures in `.png` format.

Your visualization should meet the following criteria (50% of your grade):

- ☐ (10%) You must use at least two different **geoms** to show *different* visualizations¹.
- ☐ (10%) You must modify labels as appropriate, and make sure to include units when relevant. Use the **caption** argument to attribute the source of your data.
- ☐ (10%) All text appears in **Sentence case**, as relevant.
- ☐ (10%) You must use any theme other than the default theme.
- ☐ (10%) At least one of your figures should have panels, produced either with **facet_*()** or by combining multiple plots with **cowplot::plot_grid()** or the **patchwork** package.

Additionally, your script should have the following (50% of your grade):

- ☐ (10%) Be called **data_visualization.R** and be saved inside the **scripts/03_content/** folder
- ☐ (10%) Contain code documentation using comments **#**
- ☐ (10%) Clearly indicate all packages and data loaded at the top of the script²
- ☐ (10%) Uses **relative** paths to read data and write figures. Figure(s) are exported to the **results/img** folder.
- ☐ (10%) I can reproduce your figure without needing to modify any code.

Turning in your assignment

- Please share the link to your github repo via Canvas
- The deadline for this assignment is Nov 2 by 23:59

¹*e.g.* Don't produce the same plot simply switching **geom_line()** to **geom_point()**

²I recommend you use [my snippets](#)

Resources

Class material

- [Intro to ggplot slides](#)
- [More on ggplot slides](#)
- [ggplot live coding session](#)

R4DS

- [Visualization](#) (Look at [section 1.6](#) for saving your plot)
- [Layers](#)

Example of a figure that would get 100%

```
## SET UP #####
# Load packages
library(EVR628tools)
library(tidyverse)
library(cowplot)

# Load data
data("data_geartypes") # In your case you will load them from your data/processed folder

## PROCESSING #####

# My data are already clean, but I need to make some final touches for visualization
data_vis <- data_geartypes |>
  mutate(
    geartype = str_replace_all(geartype, "_", " "), # swap the underscores for spaces
    geartype = str_to_sentence(geartype), # Make them into sentence case
    geartype = fct_lump_n(geartype, 5), # Lump them (5 categories + others)
    geartype = fct_reorder(geartype, effort_hours, .fun = mean)) # Order them based on mean

## VISUALIZE #####

# Build a figure
p1 <- ggplot(data = data_vis,
             mapping = aes(x = effort_hours, y = geartype)) +
  stat_summary(geom = "pointrange",
              fun.data = mean_se,
              pch = 21,
              color = "black",
              fill = "steelblue") +
  labs(title = "Mean fishing effort by gear",
       subtitle = "Category 'Other' contains 7 gears combined",
```

```

      x = "Fishing effort (hours) [Mean ± SE]",
      y = "Gear",
      caption = "Data come from the `gfwr` package") +
  theme_minimal(base_size = 12) +
  scale_x_continuous(expand = c(0.1, 1))

# Build my second figure
p2 <- data_vis |>
  group_by(geartype) |>
  summarize(n_vessels = n_distinct(vessel_id)) |>
  ggplot(mapping = aes(x = n_vessels, y = fct_reorder(geartype, n_vessels))) +
  geom_col(fill = "cadetblue", color = "black") +
  labs(title = "Fleet capacity by gear",
       subtitle = "Category 'Other' contains 7 gears combined",
       x = "Fleet capacity (# of vessels)",
       y = "Gear",
       caption = "Data come from the `gfwr` package") +
  theme_minimal(base_size = 12)

my_plot <- plot_grid(p1, p2,
                    ncol = 1,
                    labels = c("A", "B"))

## EXPORT #####
ggsave(plot = my_plot,
       filename = "results/img/effort_and_capacity.png", # Export my file as png
       width = 8,
       height = 8)

```

