Introduction to ggplot2

Visualizing data in R

Download the section 4 .Rmd handout to `STAT240/lecture/sect04-ggplot-basics`.

Download the file `lake-mendota-winters-2023.csv` to `STAT240/data`

Material in this section is covered by Chapter 6 on the notes website.

Each year, scientists record when Lake Mendota freezes and thaws.

Our data contains one row per winter season.

- Starts at 1855-56, ends at 2022-23
- `year1` is the starting year of the season
- `duration` is the number of days Lake Mendota was closed (frozen)

Load the data with the `read_csv` command. You'll need to have `tidyverse loaded`.

Explore the data with `View` and `glimpse`.

Note the variable types of each column.

`ggplot2` is a package included in `tidyverse` that stands for "grammar of graphics".

- Can make many types of graph with similar code
- Rich customization tools

Graphs made with this package have a specific structure.

We provide `ggplot2` with a **dataframe** and a **mapping**.

- Which variables control the aspects of the plot?
- Start with `x = year1` and `y = duration`.

Then `ggplot2` builds a graph in layers.

- First layer: empty canvas
- Second layer: Markings according to mapping

We have the option to customize color, shape, position, transparency, etc.

Let's build a plot to answer the following question.

How has the duration of time Lake Mendota turns to ice each winter changed over the last 168 years?

What types of visualizations would be a good way to answer this question?

The R command is `ggplot` and we set `aes` to define the mapping.

When we build the basic framework of this plot, the axes are there, but the data doesn't show up.

We need to pick a **geom** (geometric object) to specify what time of plot we want. A line or dot plot is appropriate here.

There are dozens of choices!

- `geom_line`
- `geom_point`
- `geom_text`
- `geom_smooth`
- `geom_boxplot`
- `geom_histogram`
- `geom_density`
- `geom_bar`

And more...

ggplots are extremely customizable. Customization options go in the chosen `geom` function.

For example, we could change the color, shape, size, and transparency of the points in our dot plot.

Some of these aspects can be mapped to variables.

Let's add a third variable in our plot, `intervals`. This is the number of closures.

This is part of our mapping, so the `color = intervals` needs to go in our `aes()`.

R will automatically apply a legend and pick a default color scheme.

The legend is a bit misleading - R belives that `intervals` (which is numeric) refers to a continuous variable.

We want it to recognize there are only two options, 1 or 2.

Use `as.factor`, which tells R to treat `intervals` as a categorical variable.

Note that aesthetics can be either constant or variable.

- Constant: treats all data the same
- Variable: tied to a column in our df

Consider a plot with both points and lines. Which layer is on top? Change the geom aesthetics.

The aesthetics we set in `geom_point` did not affect `geom_line`, and vice versa.

- Local aesthetics only affect one layer
- Global aesthetics apply to all layers

Variable aesthetics can also be either global or local.

Mappings with `aes` can be set either for all layers, or for a specific geom. Make sure you understand:

- Global vs local aesthetics
- Constant vs variable aesthetics

Finally, let's identify some common mistakes when building gplots.