Introduction to ggplot2

Visualizing data in R

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

Download the file `lake-mendota-winters-2025.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.

We have one row per winter season.

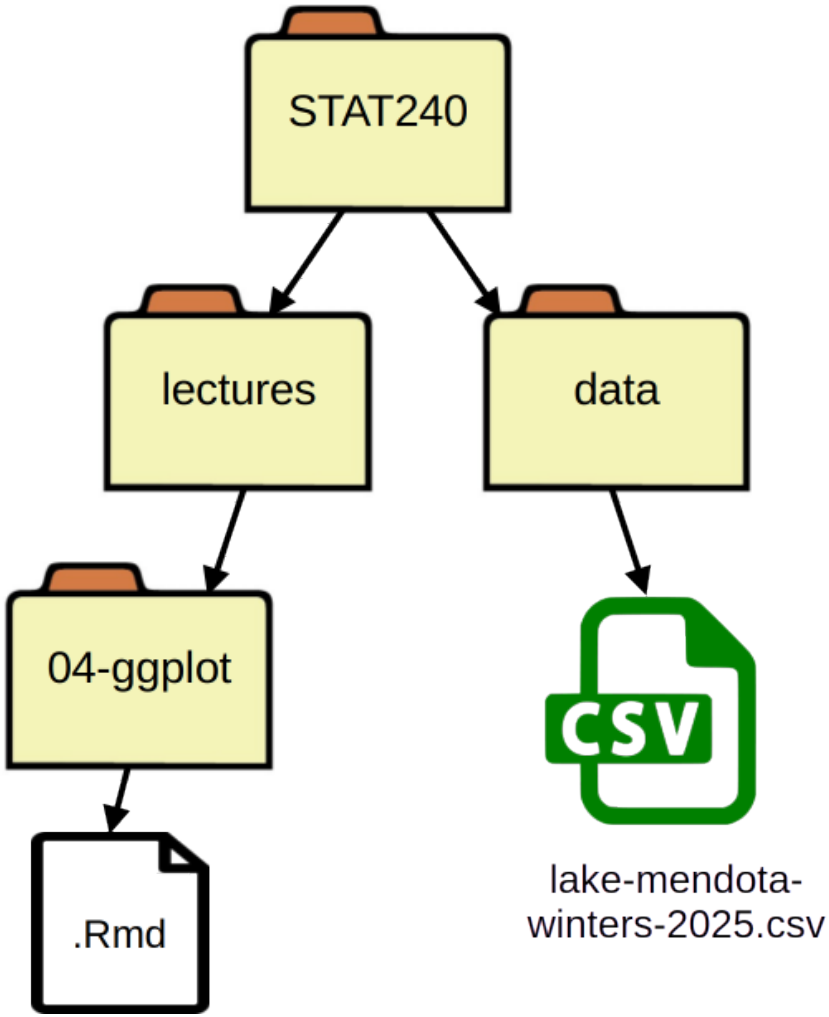
- Starts at 1855-56, ends at 2024-25
- `year1` is the starting year
- `duration` is the freeze duration in days

Load the data with the `read_csv` command.

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

Note the variable types of each column.

File structure:



`ggplot2` stands for "grammar of graphics".

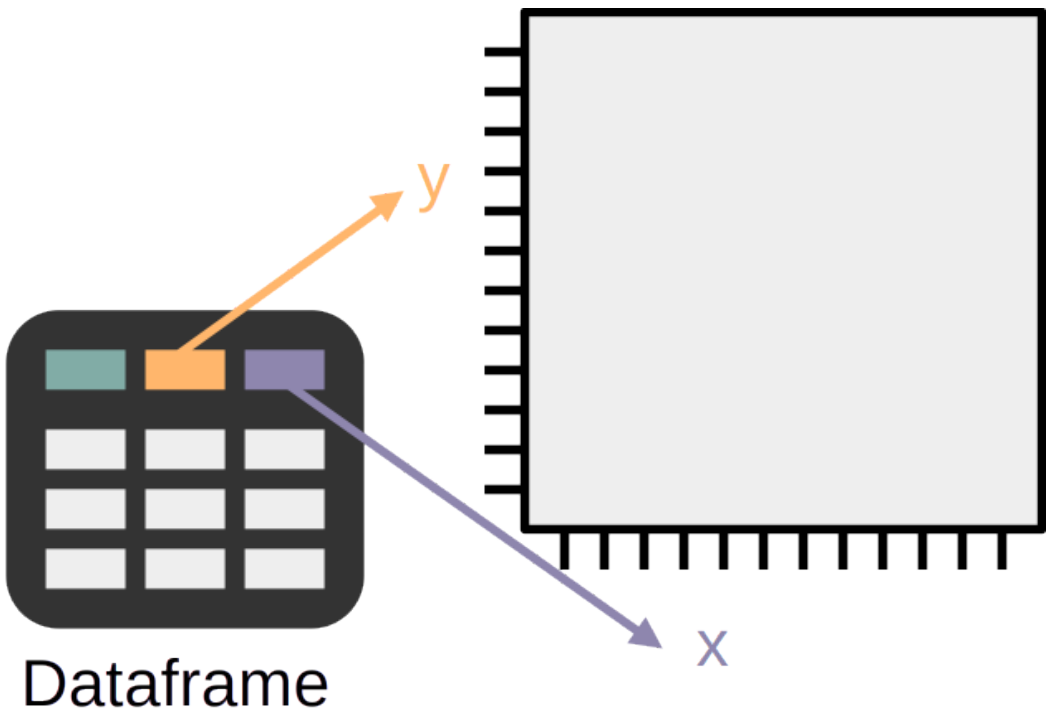
- Create different graph types with similar code
- Rich customization tools

Code will have have a specific structure.

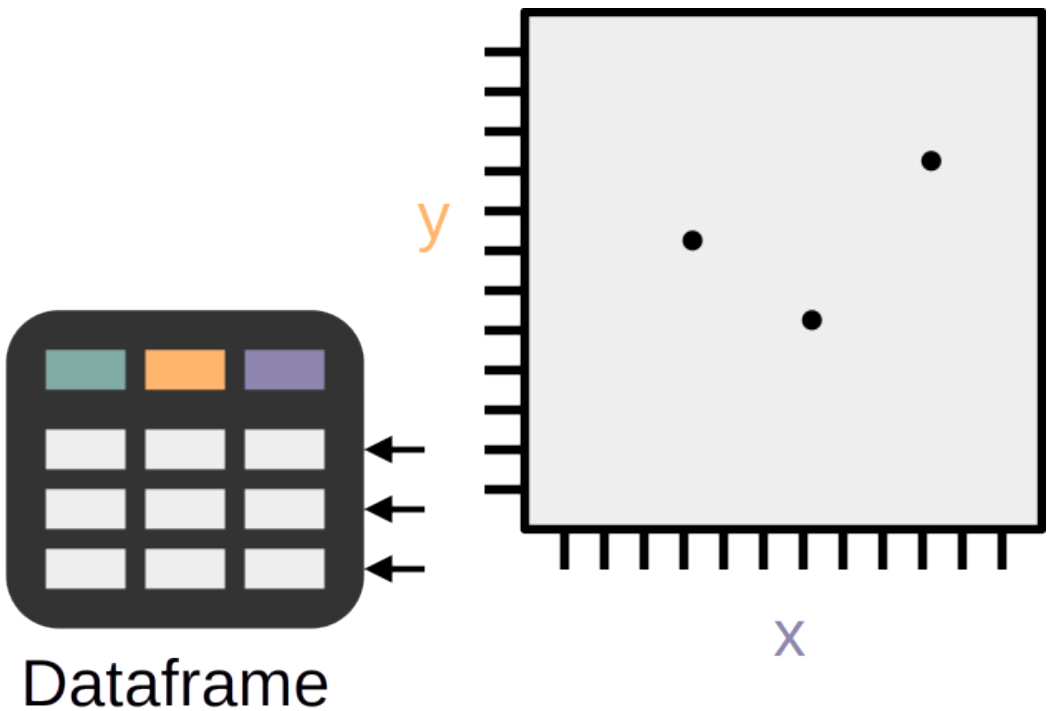
Let's build a plot to answer the following question.

How has the duration of time Lake Mendota turns to ice each winter changed over the last 170 years?

`ggplot()` builds a canvas based on a **mapping**:



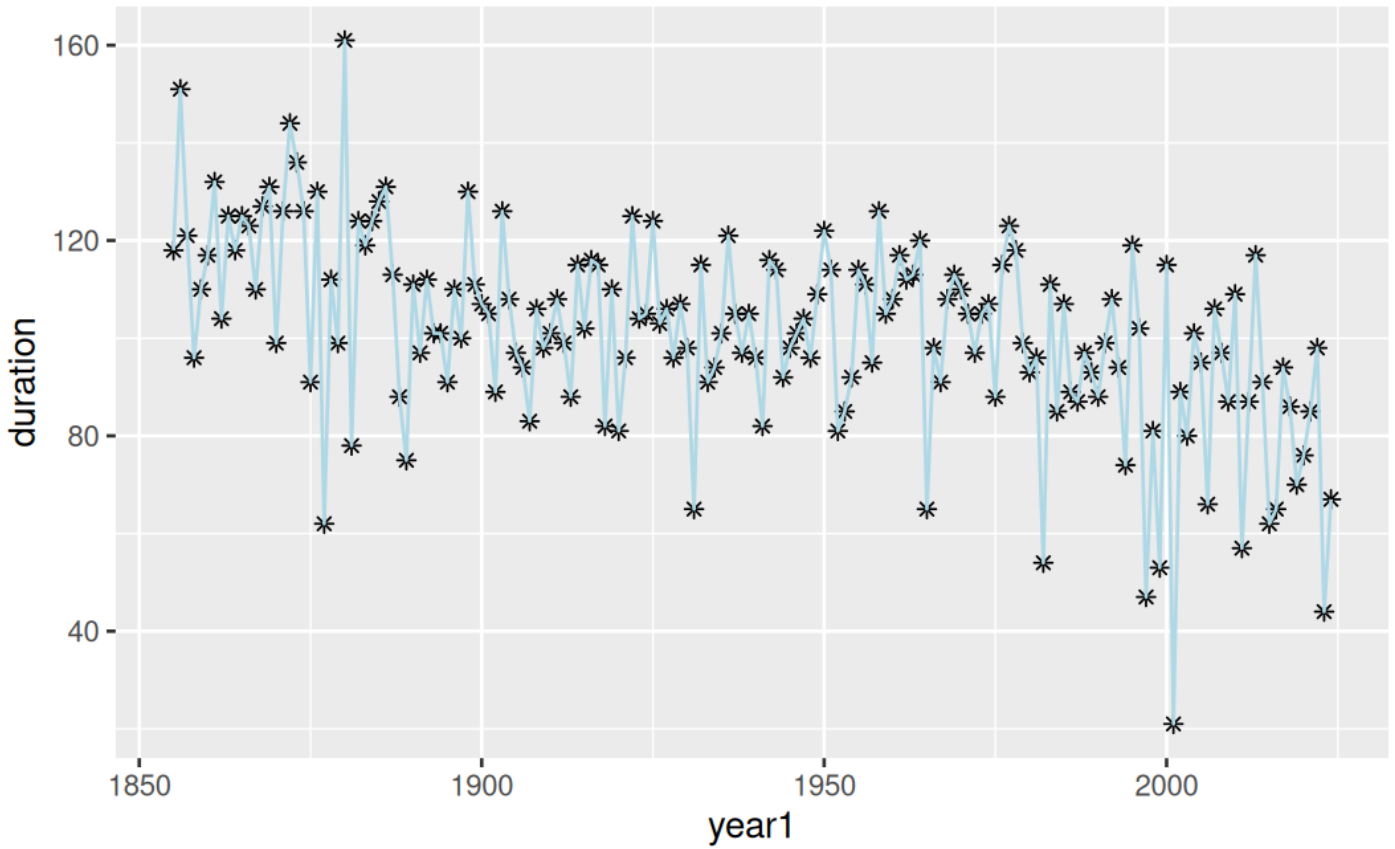
Use a `geom` to add markings:



Customization options go in the chosen `geom` function. For example:

- Color
- Shape
- Size
- Transparency

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



x and y are **variable**, and color and shape are **constant**.

There are dozens of geometries!

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

And more...

`geom_smooth()` shows the overall tend in a scatterplot.

- Different calculation methods
- Can optionally show *confidence intervals*

Let's study the duration variable on its own.

Histograms, density plots, and boxplots visualize a single numeric variable.

`geom_histogram()` divides the data into bins.

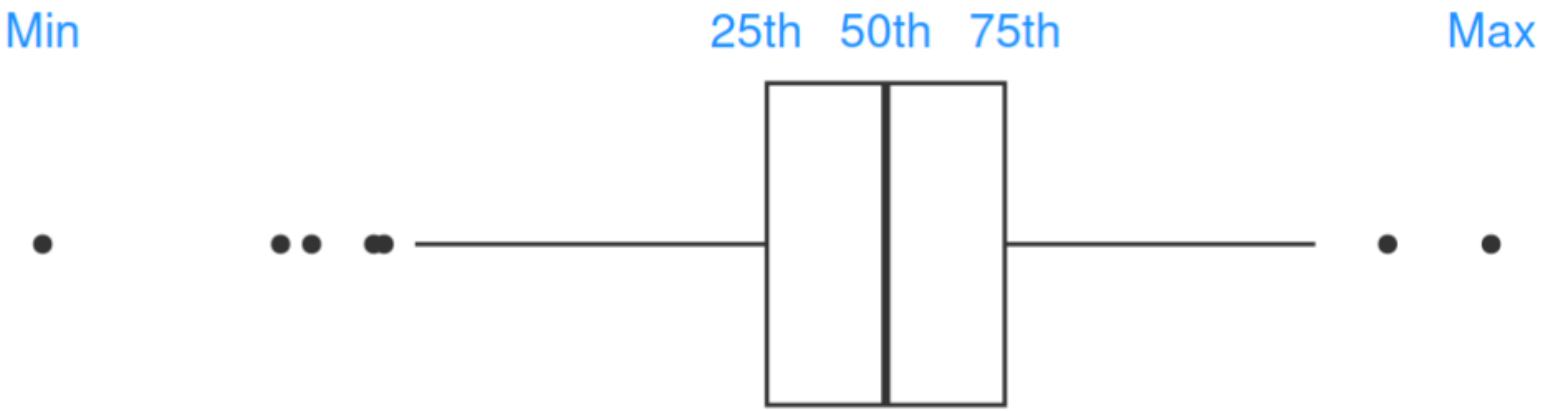
- `binwidth`: how wide the bins are
- `bins`: the number of bins

- `center`: midpoint of a bin
- `boundary`: a specific breakpoint

Use only one of (`binwidth`, `bins`) and only one of (`center`, `boundary`).

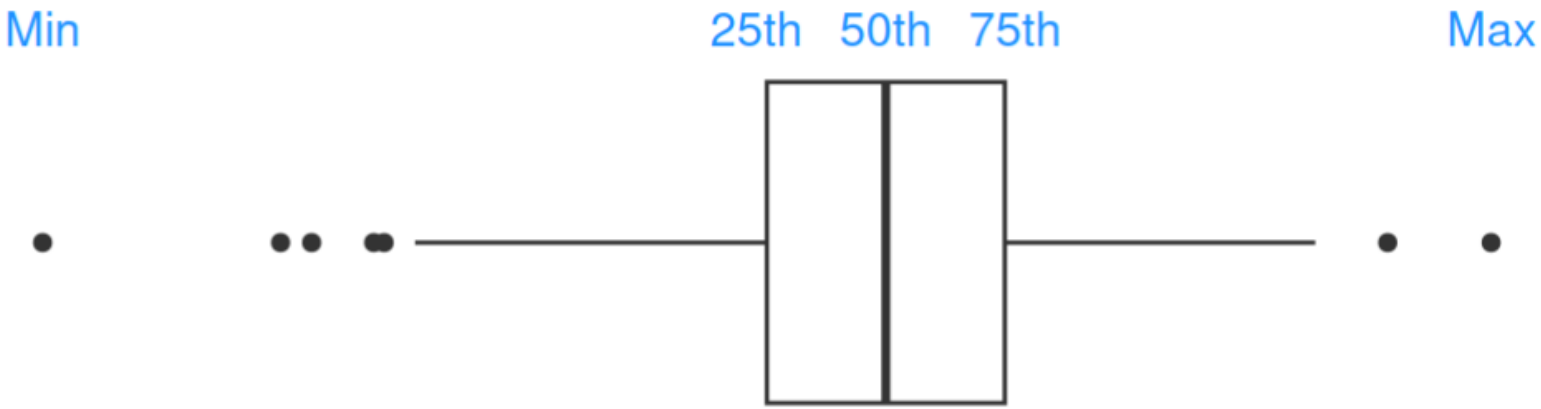
A `geom_density()` plot is similar to a histogram, but with a smooth curve.

- Shows "general trend"
- Related to integration



`geom_boxplot()` shows the **quartiles**.

- Outliers are drawn as dots



The box is the **interquartile range** (IQR).

- The "threshold" for outliers is 1.5× IQR
- Anything 1.5 "box lengths" away is a dot

(The lines only go out to data that exists.)

Consider making a categorical variable for century.

- Add `fill = century` to color-code the plots.
- What if we use `col = century` instead?
- Make a change to the density plot to make the overlapping plots more readable.

Good options for plot annotations:

- `geom_vline()`
- `geom_hline()`
- `geom_text()`

These can be mapped to variables or not.

A bar graph visualizes a single categorical variable.

Draw bars (similar to a histogram) based on the number of items in each category.

Two options: `geom_bar()` and `geom_col()`.

`geom_bar()`	`geom_col()`
Only x or y	Both x and y
Always gives counts	More flexible
Less manual calculation	Provide the bar height

Return to the scatterplot of `year` and `duration`.

How many closures occurred each year? We want to add `intervals` to the mapping.

R belives that `intervals` (which is numeric) refers to a continuous variable.

But there are only two options, 1 or 2.

Use `as.factor` to treat `intervals` as categorical.

Our point + line plot is messed up.

We want to color just the points by intervals, not the lines.

Define a mapping *inside* `geom_point()`.

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

Variable aesthetics are either:

- Global: apply to all layers
- Local: affect one layer

Make sure you understand:

- Constant vs variable aesthetics
- Global vs local aesthetics

Variable aesthetics are global or local. Constant aesthetics are always local.

We can change the axes to be more informative.

- Use `scale_x` and `scale_y` to specify the axis
- Can be `continuous` or `discrete`
- Helpful arguments: `breaks`, `labels`, `limits`, `trans`

The most fun part is choosing a color scheme.

- Colorblind friendly built-in scales in `viridis`
- Can make your own custom scale with `manual`
- Specify `d` or `c` for discrete and continuous

Here are the viridis options.
Here is a list of predefined R colors.

Use the `labs` addition to customize labels.

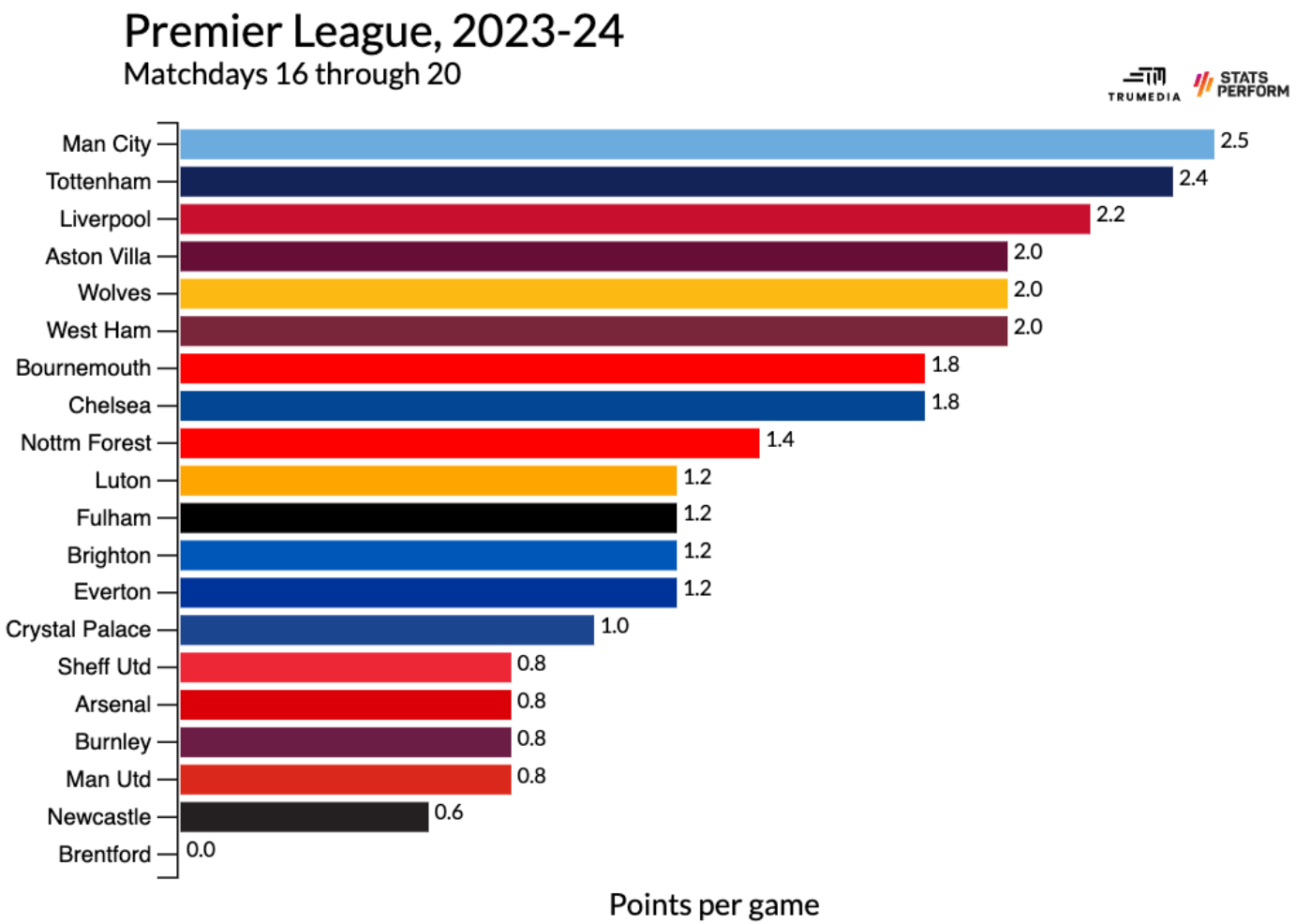
- Title, subtitle, and caption
- Edit labels for any mapping in the graph

Remove labels with `NULL`.

Themes change the overall appearance of the plot.

- Default is `theme_gray`
- Some nice ones are `theme_minimal` and `theme_classic`

Here is the list of ggplot themes.



Recreate this graphic using the partial dataframe.

Bonus topics:

- Faceting
  - `facet_grid()` and `facet_wrap()`
- Mathematical functions
  - `geom_function()`