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-2024.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 2023-24
- `year1` is the starting year
- `duration` is the freeze duration in days

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` stands for "grammar of graphics".

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

Code will have have a specific structure.

First, give a **dataframe** and a **mapping**.

- Which variables control the aspects of the plot?

`ggplot` builds a graph in layers.

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

Rich customization options are available.

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.

We then pick a **geom** (geometric object) to specify what time of plot we want.

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.

The variable `intervals` is the number of closures.

We want this to be part of our mapping in `aes()`.

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

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.

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.

Let's explore some other plot types.

- Smooth line
- Histograms, density plots, box plots
- Line and text annotations
- Bar plots

`geom_smooth` shows the overall tend in a time series scatterplot.

- Can optionally show *confidence intervals*
- Several different methods for calculation

So far, all of the plots are motivated by the relationship between year and duration.

Now, let's study the duration variable on its own.

Histograms, density plots, and boxplots are useful tools for a single numeric variable.

`geom_histogram` divides the data into bins and draws bars based on the number of observations.

- `binwidth` is how wide the bins should be
- `bins` is the number of bins
- `center` is the center of a bin
- `boundary` is a specific breakpoint

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

`geom_density` builds a density plot. It is similar to a histogram, but has a smooth curve.

- Good to emphasize "general trend"
- Related to integration

Consider layering both a density and histogram plot.

`geom_boxplot` creates a "box-and-whisker" plot. This visualizes the **quartiles**.

- Shows minimum, 25th, 50th, 75th percentiles, and maximum
- The box shows the middle 50% of the data
- Outliers are drawn as dots

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

- The "threshold" for outliers is 1.5$\times$ IQR
- Anything 1.5 "box lengths" away is a dot

Note: the lines only go out to data that exists.

Consider making a categorical variable for century.

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

Lines are a useful way to annotate different types of numeric plots.

- Use `geom_vline` or `geom_hline`
- Can add multiple lines

`geom_text` can do variable mapping but is also useful for text annotations.

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

A bar graph visualizes a single categorical variable.

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

We see that the x-axis is organized alphabetically.

The bar plot counts instances for us. We use `geom_col` to manually give the height.

This geom takes a categorical and numeric variable.

- Requires more manual calculation
- More flexible, not just counts

We can edit the axes of our plots to be more useful and informative.

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

The most fun part of graphing is choosing a color scheme.

- Useful (colorblind friendly) built-in scales in `viridis`
- Can make your own custom scale with `manual`
- Specify `d` or `c` for discrete and continuous color schemes

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

Use the `labs` addition to customize labels.

- Title, subtitle, and caption
- Can change labels for any mapping present in your graph
- Can make labels blank as well

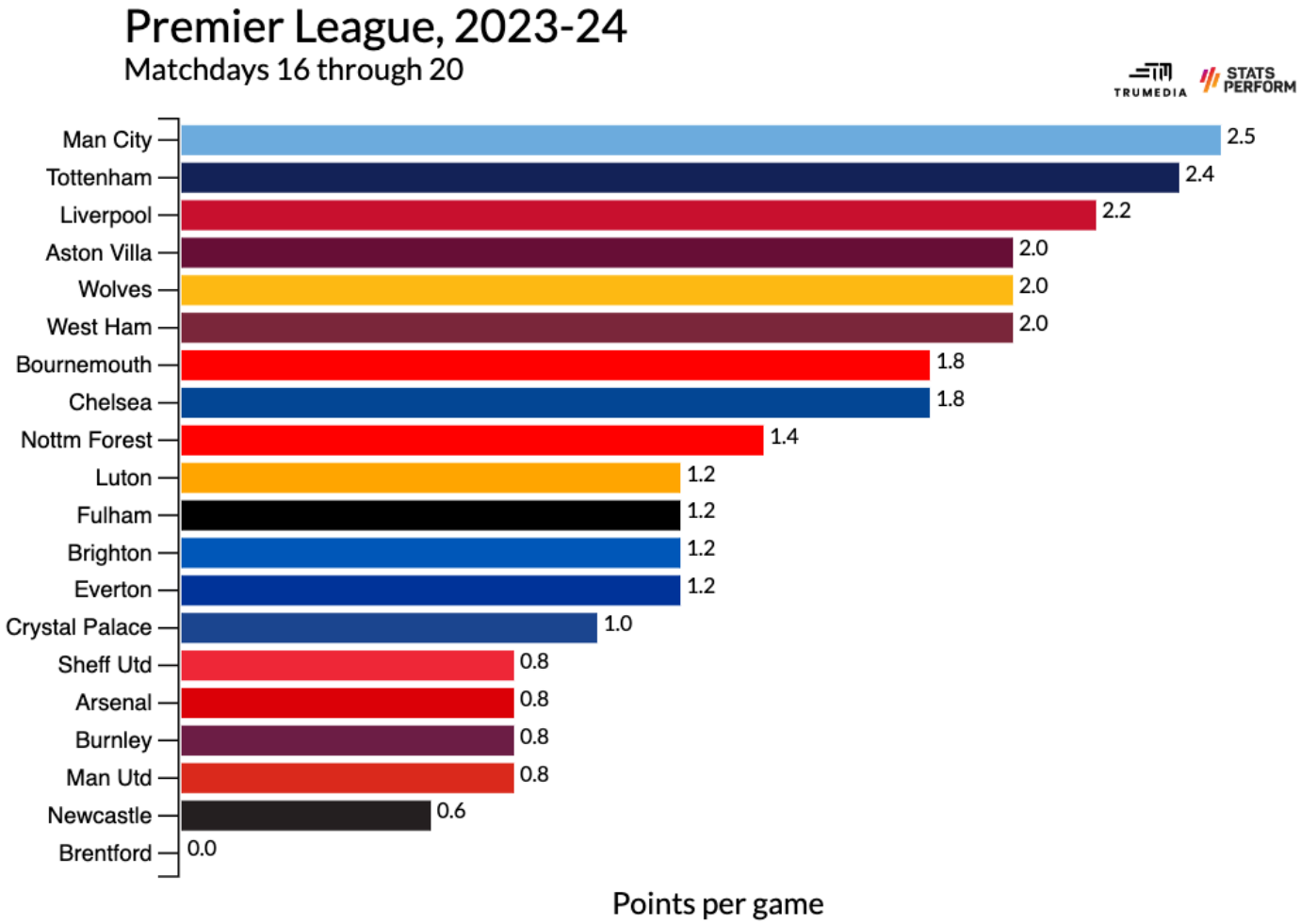
Remove labels with `NULL`.

Themes change the overall appearance of the background of your plot.

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

We can also specify the font size and family.

Here is the list of ggplot themes.



Recreate this graphic using the partial dataframe in the .Rmd.

It can be difficult to view many overlapping plots. **Faceting** splits each plot onto its own panel.

Facet based on one variable with `facet_wrap`, or two variables with `facet_grid`.

- Need to specify `vars()`.

Let's explore different ways to facet the duration data. Note that R will always try to fill in every spot of `facet_grid`.

Let's create a new column for whether a year is a leap year.

How can we use faceting to explore trends in duration across both century and leap year?

Consider the `facet_grid` graph we just made.

- The bottom right panel shows the durations among (leap years/non-leap years) in the (19th/20th/21st) century.
- We don't expect there to be a difference in duration between non-leap years and leap years. So, each (row/column) has roughly the same center across its panels.
- We expect there to be a difference in duration across centuries. So, each (row/column) has different centers across its panels.