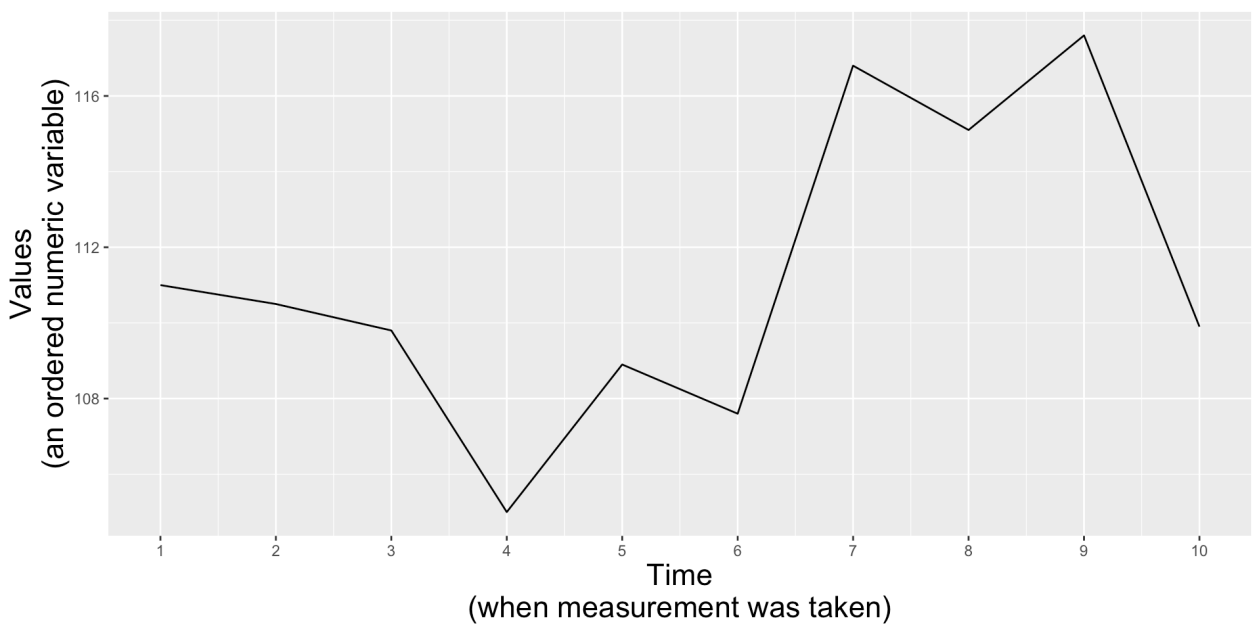
EDS 240: Lecture 2.3

Visualizing evolution

Week 2 | January 13th, 2024

Visualizing data *evolution*?

Visualizing the change in a **numeric variable** over some unit of time.



Roadmap

In this lesson, we'll explore two primary chart types:

1. line graphs
2. area charts

Roadmap

In this lesson, we'll explore two primary chart types:

1. line graphs

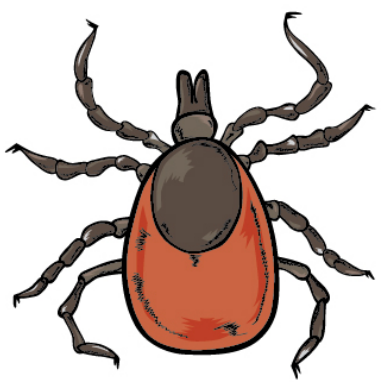
- avoiding spaghetti plots
- cutting the y-axis
- aspect ratio

2. area charts

- 1 vs. multiple groups
- how to interpret them
- stacked vs. proportional stacked area chart
- considerations

The data: Lyme disease

Lyme disease has been a nationally notifiable condition in the United States since 1991. Reports of Lyme disease are collected and verified by local and state health departments, anonymized by the National Notifiable Diseases Surveillance System (NNDSS), then shared with Centers for Disease Control and Prevention (CDC). The CDC has developed public use data sets for download to facilitate the public health and research community's access to NNDSS data on Lyme disease.

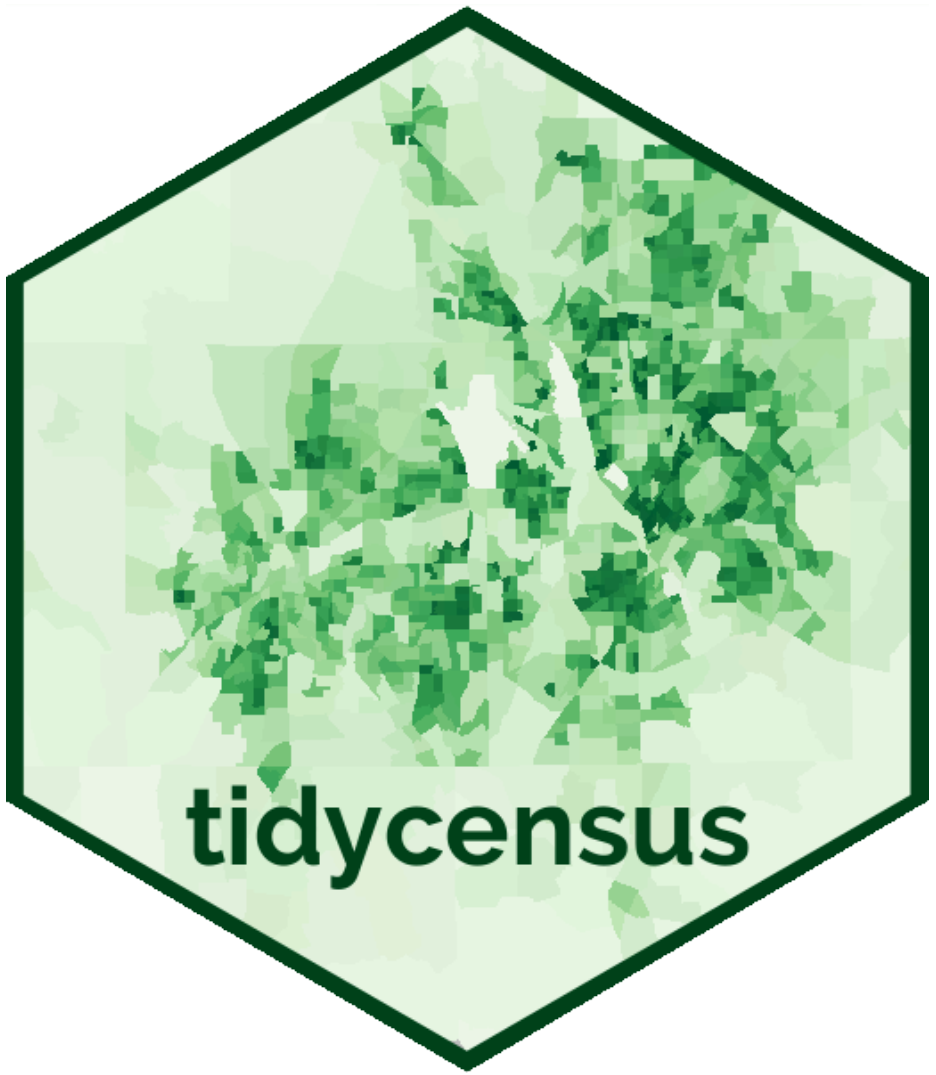


Using CDC data on Lyme disease and population estimates from the US Census Bureau (via the `{tidycensus}` package), we'll explore **changes in Lyme disease incidence (cases/100k people) over time**, by state.

`{tidycensus}` for accessing US population data

The `{tidycensus}` package allows users to interface with a select number of the US Census Bureau's data APIs and return tidyverse-ready data frames.

Using `{tidycensus}` requires that you first have an API key. **Be sure to follow the Pre-Course setup instructions for requesting and activating your key before proceeding.**



Data wrangling

See the online documentation for more information on downloading and parsing population data using the `{tidycensus}` package.

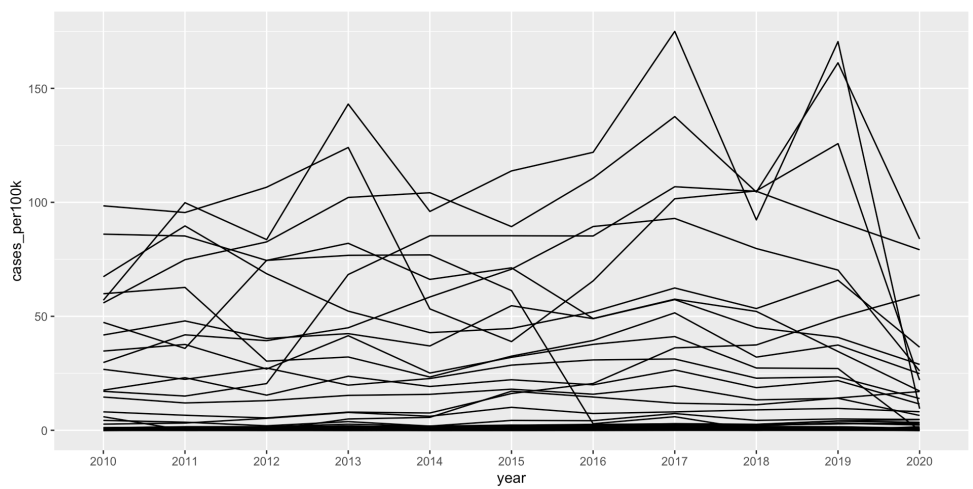
```
##~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
##                                    setup                                  ----
##~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

#.......................load libraries.......................
library(tidycensus)
library(tidyverse)
library(gghighlight)

#.......................import data.........................
lyme <- read_csv(here::here("week2", "data", "Lyme_Disease_Cases_by_State_or_Locali

##~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
##                           wrangle lyme disease data                       ----
##~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

#...........wide to long (plus some other wrangling)...........
lyme_clean <- lyme |>

  rename(state = State) |>
```

Line plots show the evolution of 1+ numeric variables

Line graphs display the **evolution of one or several *numeric* variables**. They are similar to scatter plots, but the measurement points are **ordered** (typically by their x-axis value) and joined with straight line segments. They are often used to visualize a trend in data over intervals of time. For example, changes in Lyme disease incidence (# cases / 100k people) from 2010 - 2020, by state:

A basic line graph using `geom_line()`

```
1  ggplot(lyme_pop, aes(x = year, y = cas
2    geom_line()
```

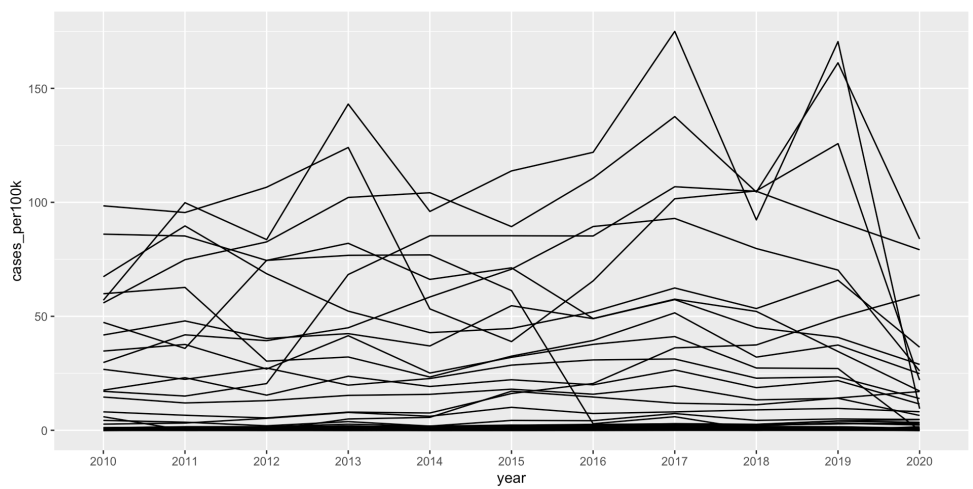


Line plots show the evolution of 1+ numeric variables

Line graphs display the **evolution of one or several *numeric* variables**. They are similar to scatter plots, but the measurement points are **ordered** (typically by their x-axis value) and joined with straight line segments. They are often used to visualize a trend in data over intervals of time. For example, changes in Lyme disease incidence (# cases / 100k people) from 2010 - 2020, by state:

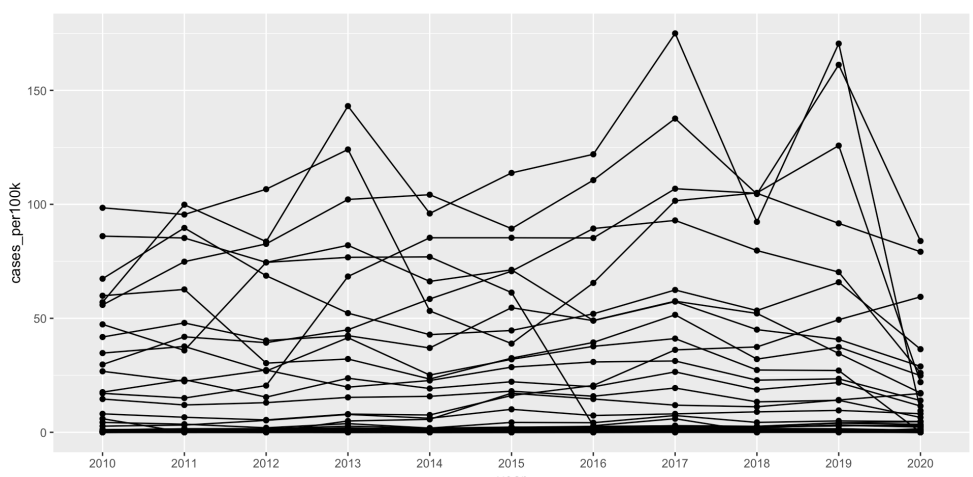
A basic line graph using `geom_line()`

```
1  ggplot(lyme_pop, aes(x = year, y = cas
2    geom_line()
```



A line + scatter plot created by layering `geom_line()` & `geom_point()`

```
1  ggplot(lyme_pop, aes(x = year, y = cas
2    geom_line() +
3    geom_point()
```

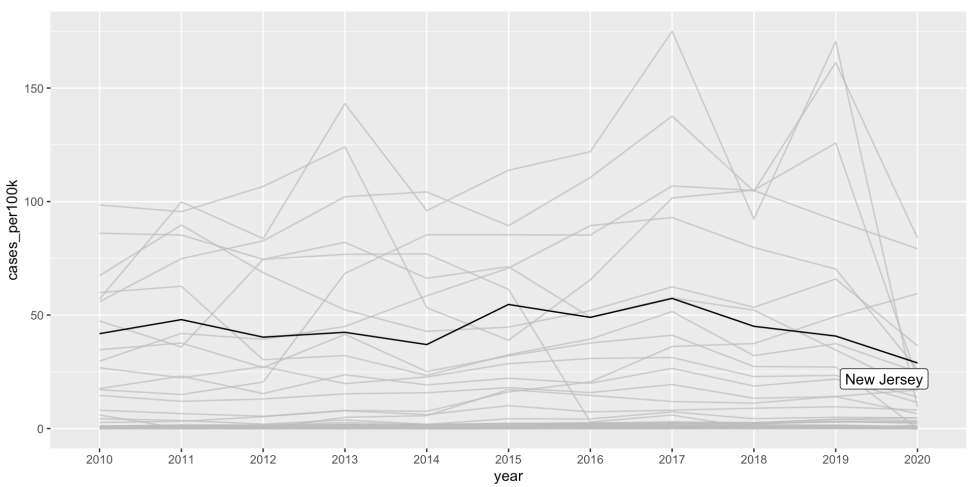


“Spaghetti plots” are hard to read

A line plot with many lines displayed together can be hard to read / overwhelming to interpret. Consider highlighting a group(s) of interest (the `{gghighlight}` package comes in handy):

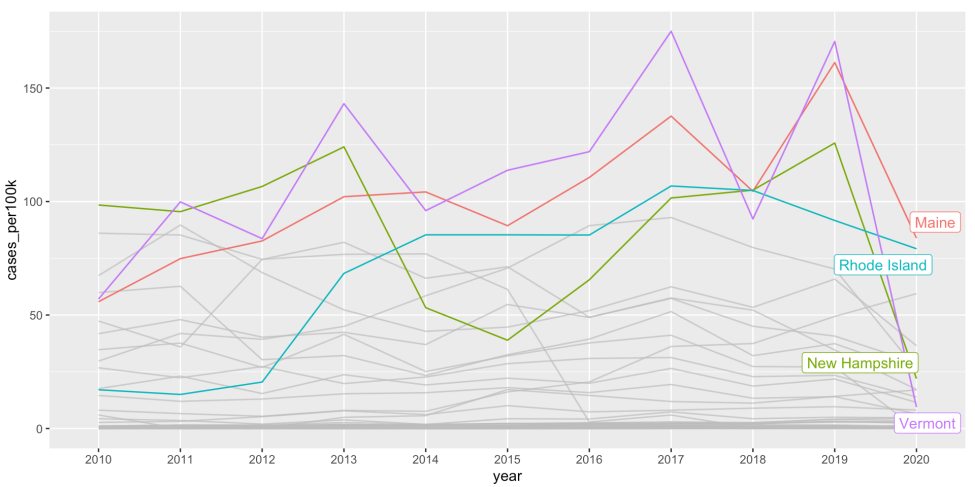
E.g. I’m interested in how Lyme disease in New Jersey compares to other states.

```
1  ggplot(lyme_pop, aes(x = year, y = cas
2    geom_line() +
3    gghighlight::gghighlight(state == "N
```



E.g. I’m interested in states where Lyme disease incidence is (or at one point was) > 100 cases / 100k people.

```
1  ggplot(lyme_pop, aes(x = year, y = cas
2    geom_line() +
3    gghighlight::gghighlight(max(cases_p
```

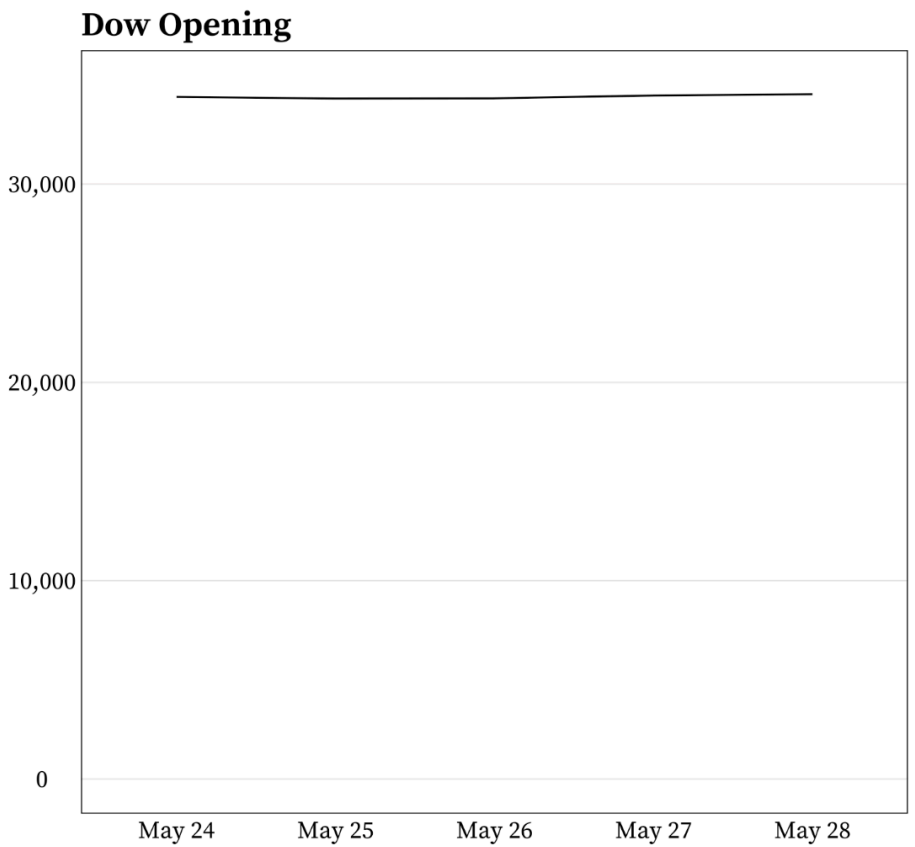


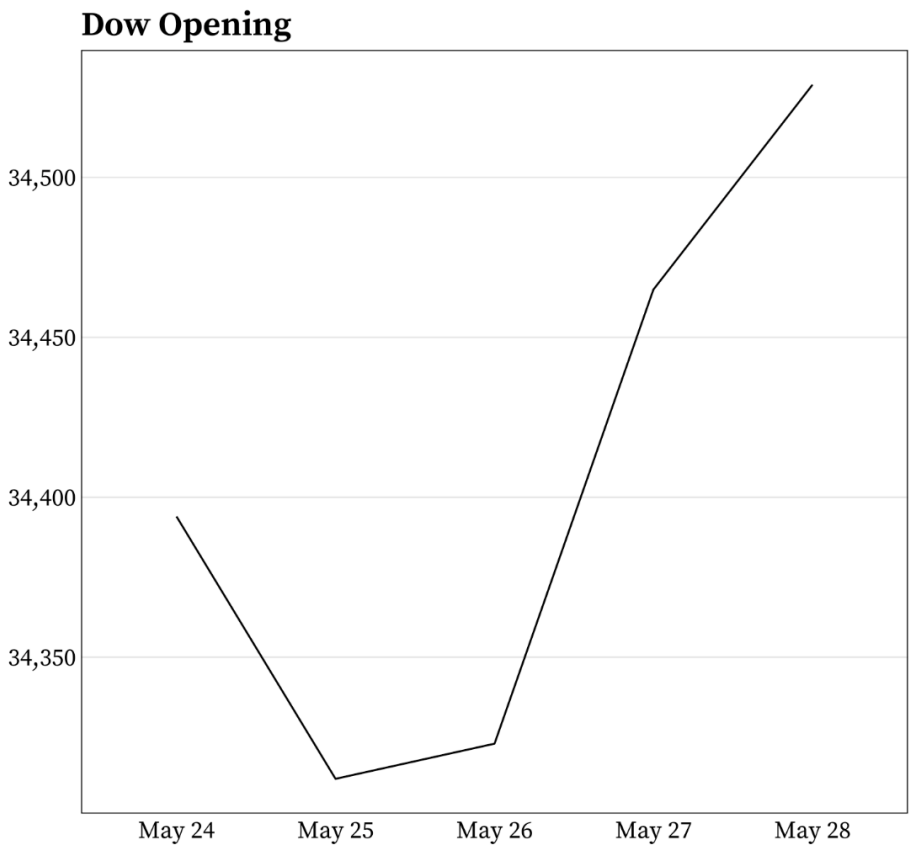
It's okay to cut the y-axis of line graphs

Line graphs encode data **by position** and **not length** (e.g. as in the height of a bar graph), therefore, you can choose to include the 0 origin only if it makes sense.

It's okay to cut the y-axis of line graphs

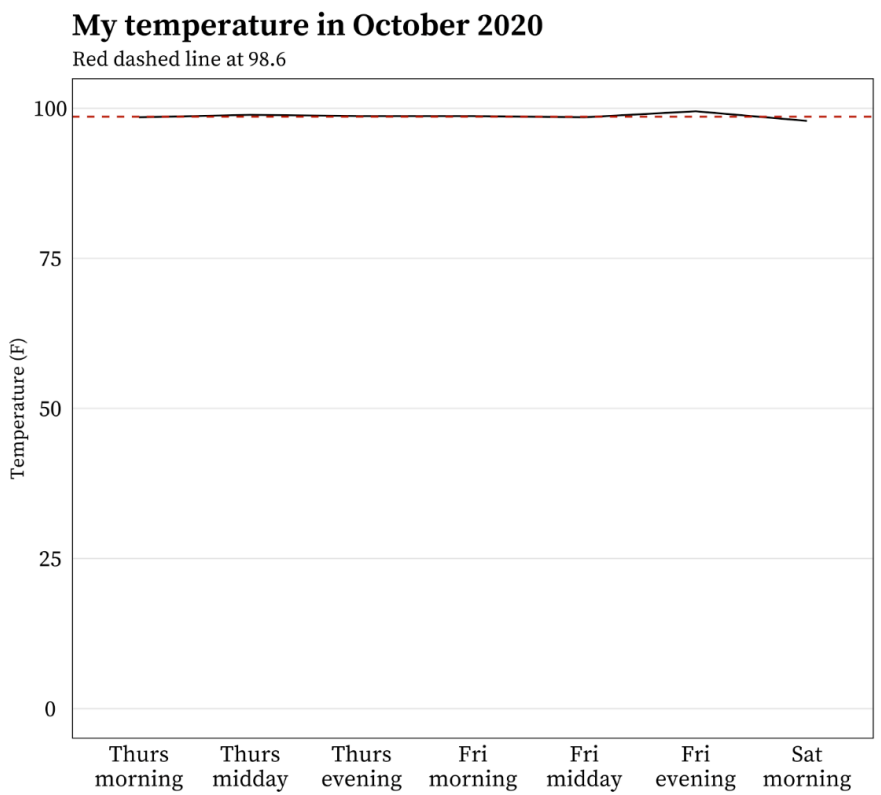
Do not start the y-axis at 0 if the **range of data is small but the distance from the bottom of the range to zero is large**. For example:

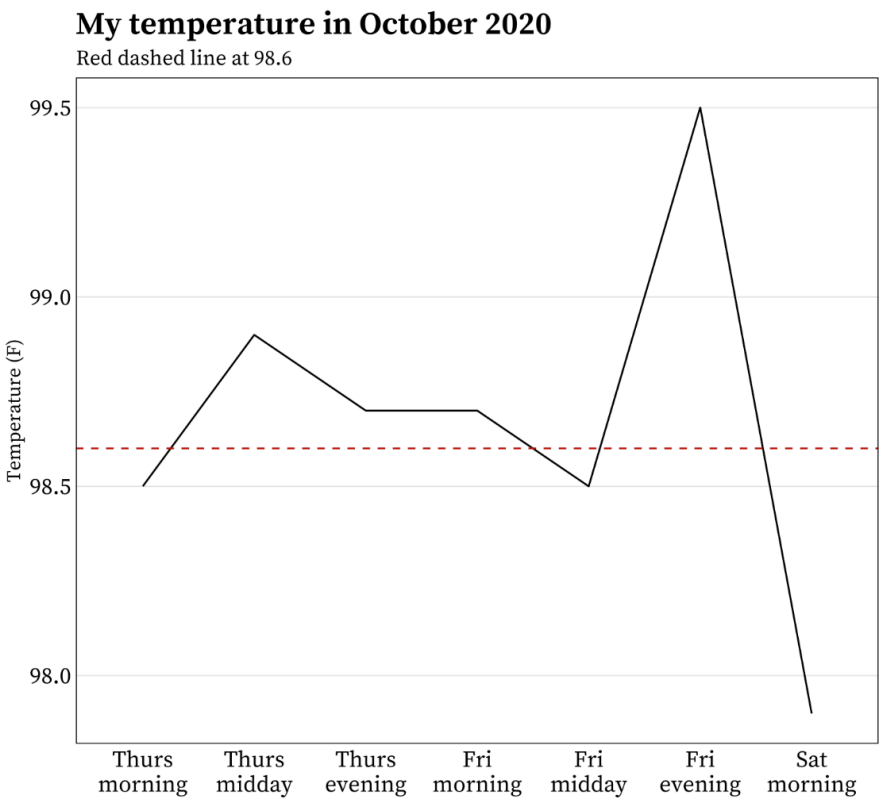




It's okay to cut the y-axis of line graphs

Do not start the y-axis at 0 if the **relationship to zero is insignificant**. For example:





Aspect ratio affects perception of slope

The aspect ratio is the **height:width ratio of a graph**. The **larger the aspect ratio, the steeper changes appear**, which may cause readers to **interpret changes as more important**. The **smaller the aspect ratio, the flatter the line** which may cause readers to **interpret changes as small / insignificant**.

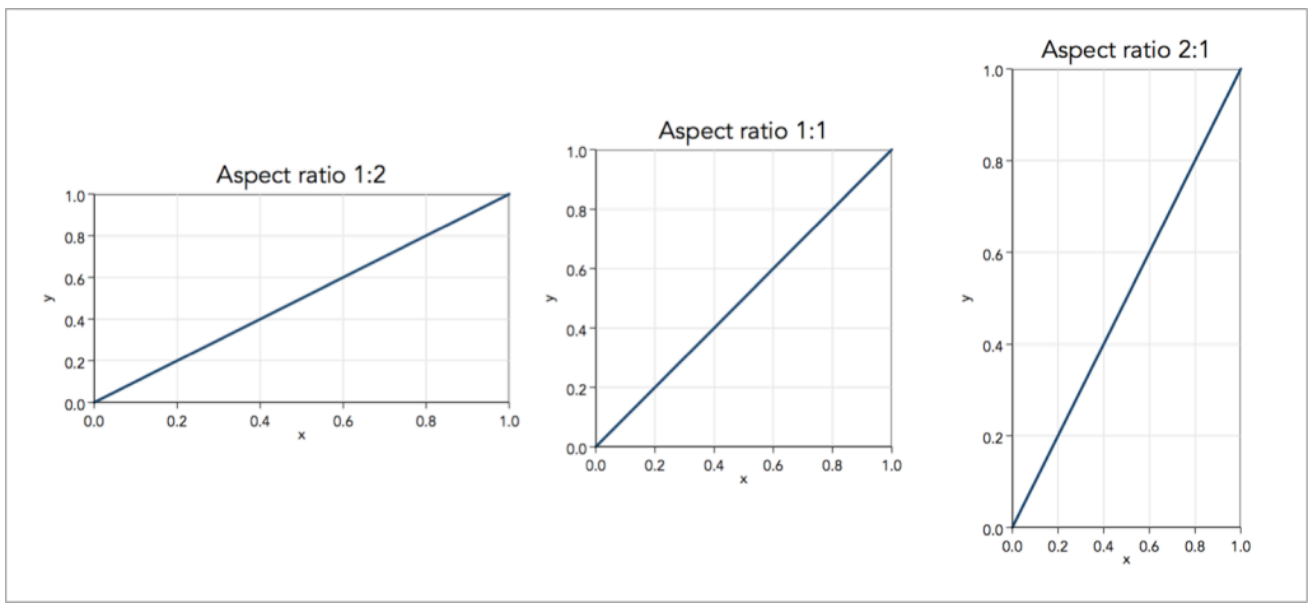


Image source: Graph workflow

There's **no exact rule** for what aspect ratio to use for a given graphic (but see Cleveland et al. 1988 to read about the "banking to 45 degrees" rule) – it depends on the nature of the variable and your goal with the visualization. *However* it's important to keep mind that **manipulating the aspect ratio can mislead readers, and so you should do so carefully**.

Aspect ratio affects perception of slope

Consider this line graph of sunspot activity from 1700 - 2015. It was created using Stata's default aspect ratio. Can you easily identify where in time sunspot activity rises more quickly / sharply than others?

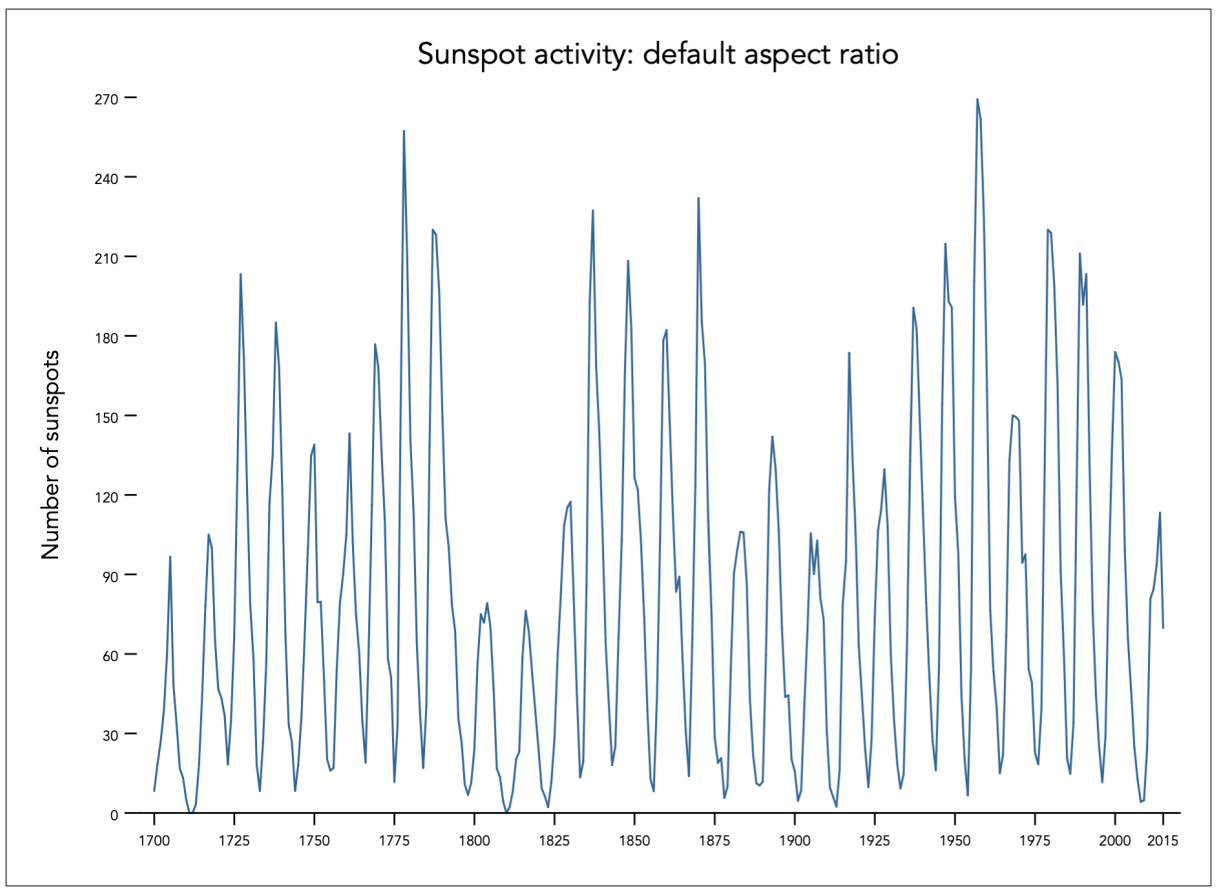
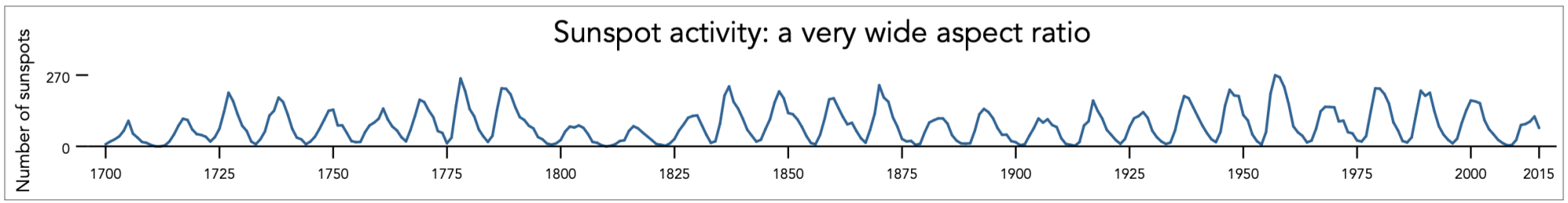
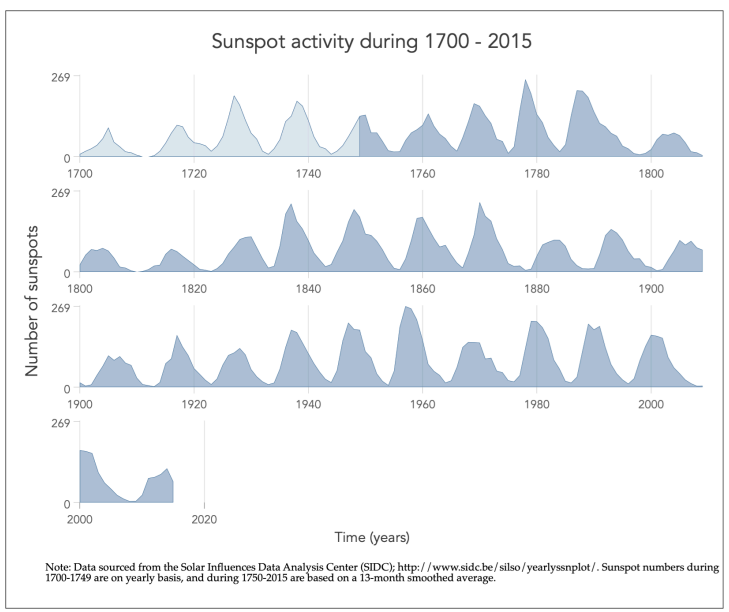


Image source: Graph workflow

Aspect ratio affects perception of slope





Note: The bottom plot is the same as the top, but with the x-axis split into panels / rows, by decade.

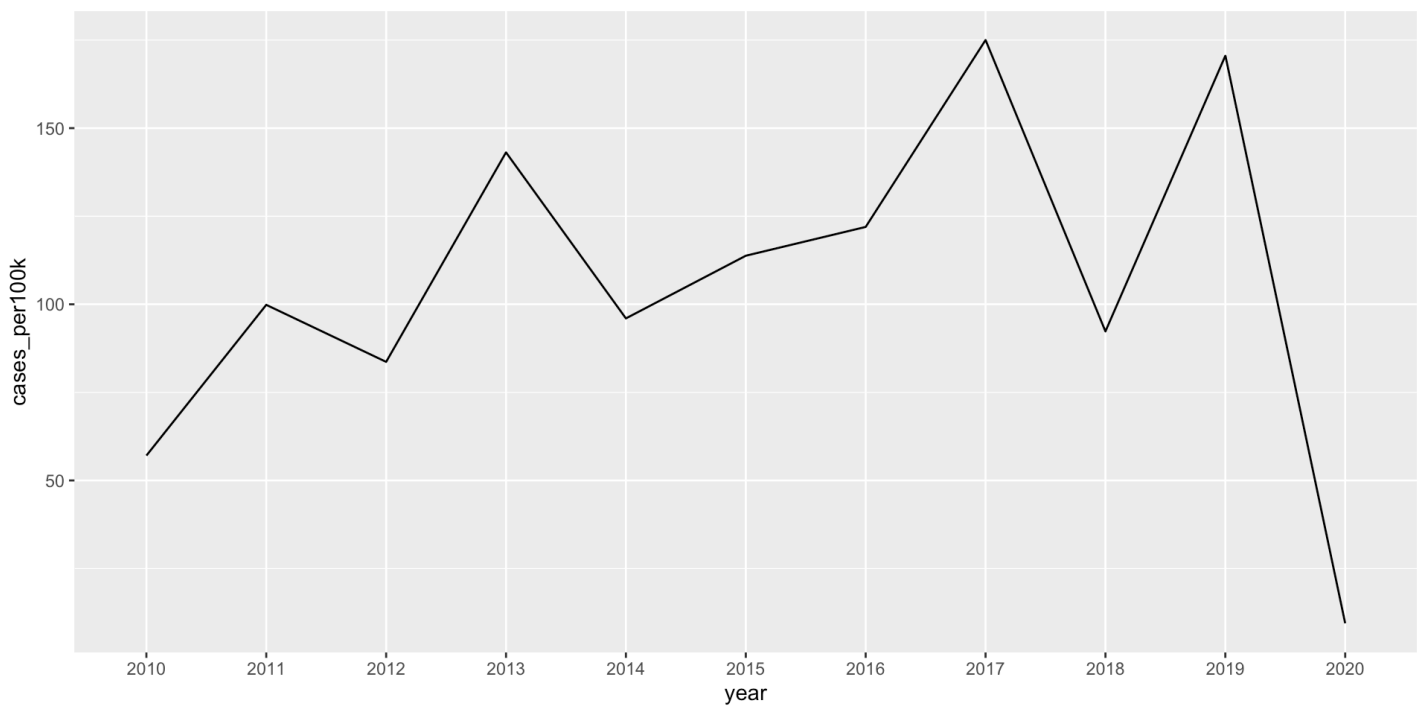
With a wider aspect ratio, we can more clearly see the differences in rates of change (slopes) in sunspot activity through time.

Image source: Graph workflow

ggplot with a default aspect ratio

Let's first look at Lyme disease incidence for Vermont from 2010 - 2020 without adjusting the aspect ratio:

```
lyme_pop |>
  filter(state == "Vermont") |>
  ggplot(aes(x = year, y = cases_per100k, group = state)) +
  geom_line()
```



ggplot with a default aspect ratio

We have 10 units on our x-axis (`year` ranges from 2010 - 2020), and ~175 units (`case_per100k` ranges from 0 to ~175) on our y-axis. By default, ggplot adjusts the space between each x-axis unit so that they are wider apart than each y-axis unit, making the plot easier to read. **Below, we've added in tick marks for each y-axis unit to better highlight this (a single tick already existed for each of our 10 x-axis units).**

Note: This plot doesn't render well on these slides *or* in the RStudio plot pane. I recommend running the code in RStudio, then clicking the **Zoom button** in the **Plot pane** for best viewing.

```
lyme_pop |>
  filter(state == "Vermont") |>
  ggplot(aes(x = year, y = cases_per100k, group = state)) +
  geom_line() +
  scale_y_continuous(breaks = seq(0, 190, by = 1))
```

Adjust the aspect ratio using `coord_fixed()`

We can use `coord_fixed()` to fix the aspect ratio of our plot. The `ratio` argument controls the aspect ratio, which is expressed as `y / x` and by default is set to `1`. This means that the height of one y-unit is equal to the width of one x-unit (paying attention to the grid lines and tick marks here can be helpful). **Because we have 175 y-axis units and only 10 x-axis units, fixing our aspect ratio at 1:1 means our plot gets taller and squished.**

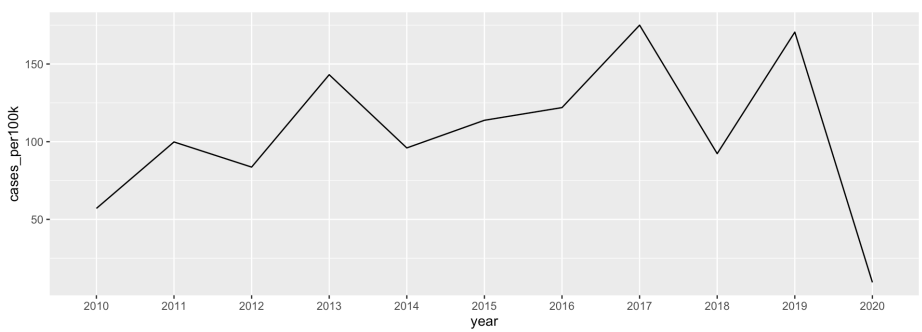
Note: This plot doesn't render well in thse slides *or* in the RStudio plot pane. I recommend running the code in RStudio, then clicking the **Zoom button** in the **Plot pane** for best viewing.

```
lyme_pop |>
  filter(state == "Vermont") |>
  ggplot(aes(x = year, y = cases_per100k, group = state)) +
  geom_line() +
  scale_y_continuous(breaks = seq(0, 190, by = 1)) +
  coord_fixed(ratio = 1)
```

Adjust the aspect ratio using `coord_fixed()`

Ratios > 1 will make units on the y-axis longer than units on the x-axis (resulting in steeper slopes). Ratios < 1 will make units on the x-axis longer than units on the y-axis (resulting in shallower slopes). **If we want to make our graph wider, we'll need to update `ratio` so that it's < 1.** For example:

```
lyme_pop |>
  filter(state == "Vermont") |>
  ggplot(aes(x = year, y = cases_per100k, group = state)) +
  geom_line() +
  coord_fixed(ratio = 1/50)
```



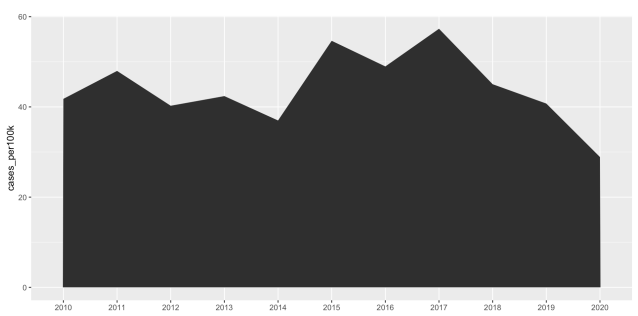
Playing around with the `ratio` value yourself is the best way to get a sense for how the aspect ratio of a given ggplot will change.

Area chart is similar to a line graph, just filled in

Instead of just a line or scatter plot to indicate the change in a numeric variable through time, the space between the line and the x-axis is colored or shaded in. Area plots are sometimes criticized for **violating the data-ink ratio rule**, which argues that any non-data-ink should be omitted wherever possible. If the number of observations is low (as in this example) a connected scatter plot may more clearly show when each observation was made.

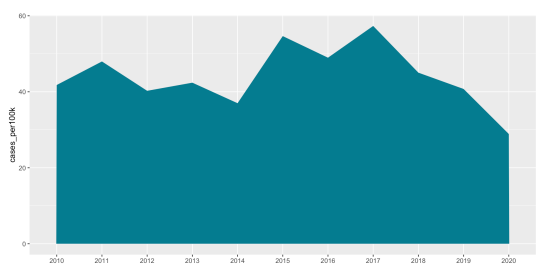
A basic area plot (New Jersey)

```
1  lyme_pop |>
2    filter(state == "New
3    ggplot(aes(x = year,
4    geom_area()
```



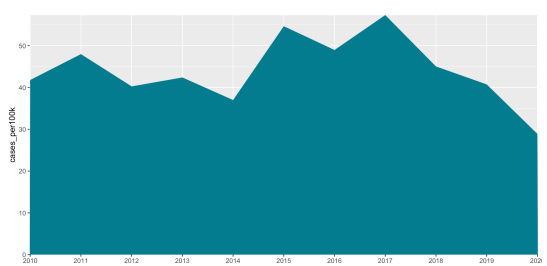
Update the fill color

```
1  lyme_pop |>
2    filter(state == "New
3    ggplot(aes(x = year,
4    geom_area() +
5    scale_fill_manual(val
6    theme(legend.position
```



Expand area to panel margins

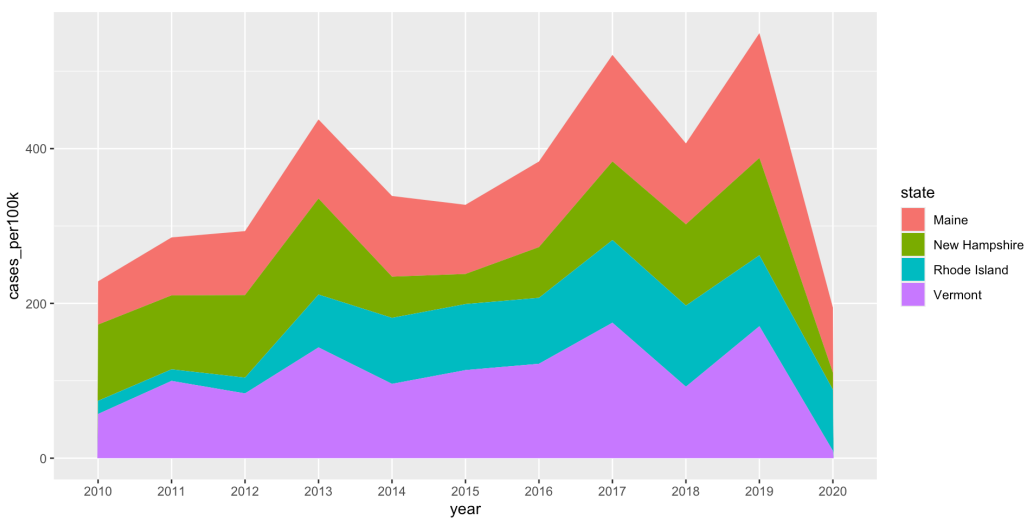
```
1  lyme_pop |>
2    filter(state == "New
3    ggplot(aes(x = year,
4    geom_area() +
5    scale_fill_manual(val
6    scale_x_discrete(expa
7    scale_y_continuous(ex
8    theme(legend.position
```



Stacked area charts show the evolution of a whole + the relative contribution of each group

Stacked area charts are useful for showing the evolution of a whole and the relative proportions of each group that make up the whole. For example, the top of the colored area shows the total Lyme disease incidence (# cases / 100k people) across all groups (notice the difference in y-axis values), while the individual colors are the relative contributions of the top 4 states with the highest lyme disease incidence:

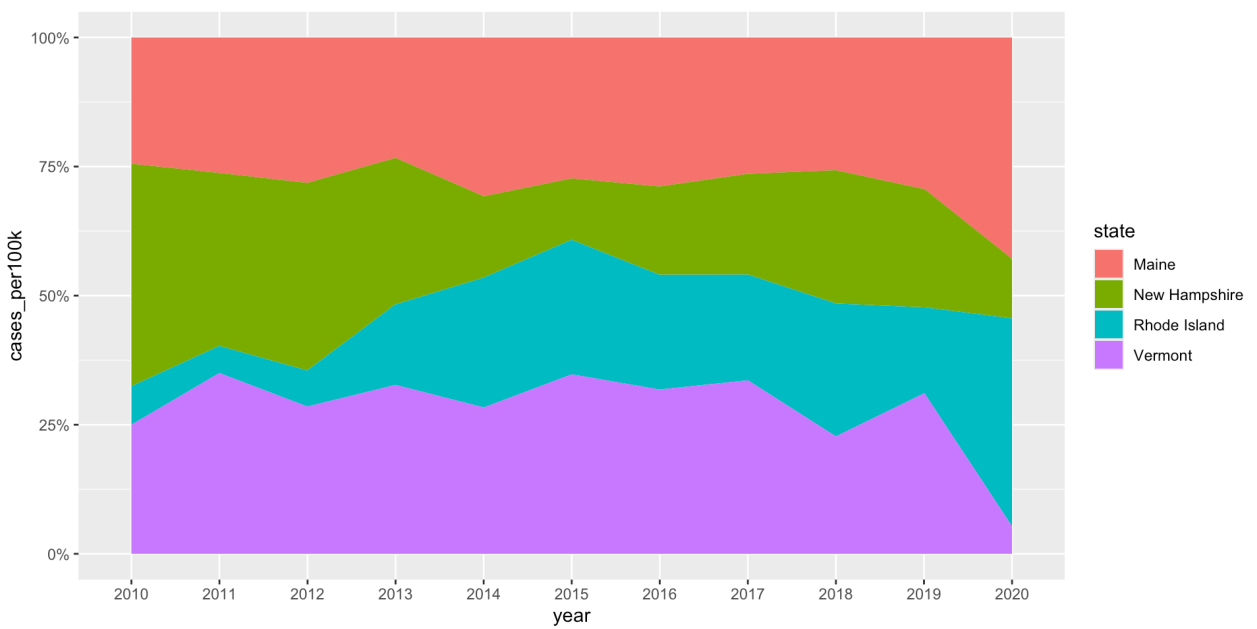
```
lyme_pop |>
  filter(state %in% c("Maine", "Rhode Island", "New Hampshire", "Vermont")) |>
  ggplot(aes(x = year, y = cases_per100k, group = state, fill = state)) +
  geom_area()
```



A variant: proportional stacked area charts

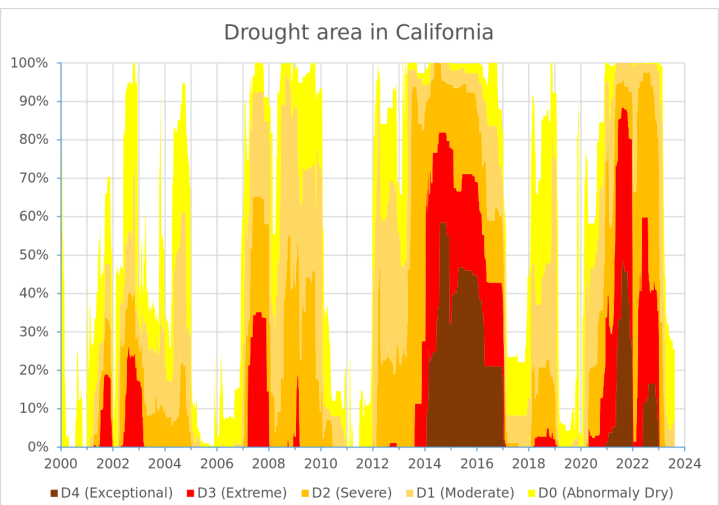
Proportional stacked area charts **plot percentage contribution instead of absolute numbers on the y-axis**. The focus of this version is the proportion of contribution made by each category rather than absolute numbers.

```
lyme_pop |>
  filter(state %in% c("Maine", "Rhode Island", "New Hampshire", "Vermont")) |>
  ggplot(aes(x = year, y = cases_per100k, group = state, fill = state)) +
  geom_area(position = "fill") +
  scale_y_continuous(labels = scales::label_percent(scale = 100))
```



Group order matters!

Group order (from bottom to top) can have an influence – oftentimes, you'll want to **put the most important group on the bottom (closest to the x-axis)**, since your audience will have an easier time reading values for that group. For example, US Drought Monitor likely wanted to draw attention to what percentage of land area in CA experienced the highest-severity drought level (D4, Exceptional). By plotting that group on the bottom of the graph below, we can more easily identify that ~60% of CA experienced the worst level of drought in 2014-2015.



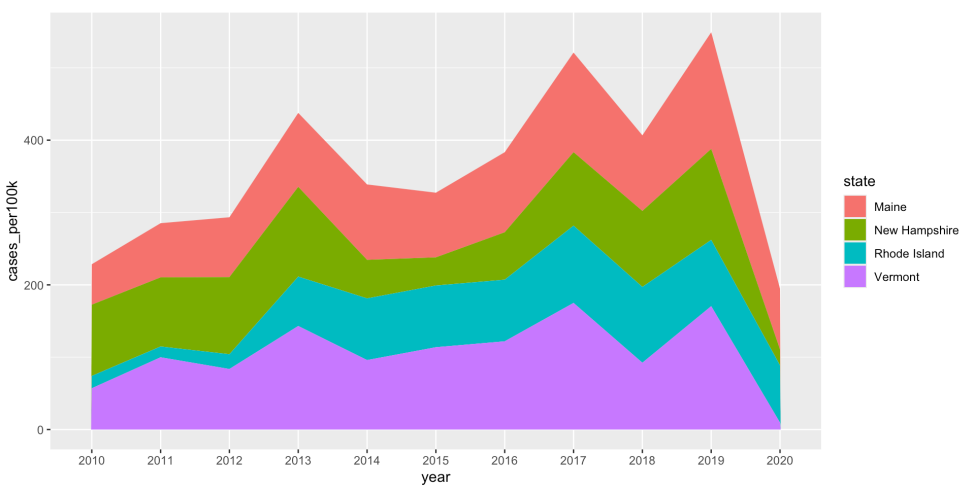
← You'll be recreating this graph (original source US Drought Monitor, via Wikipedia) in discussion section this week!

This article by Info River nicely outlines situations where using a stacked area chart is great, when not to use them, and important considerations.

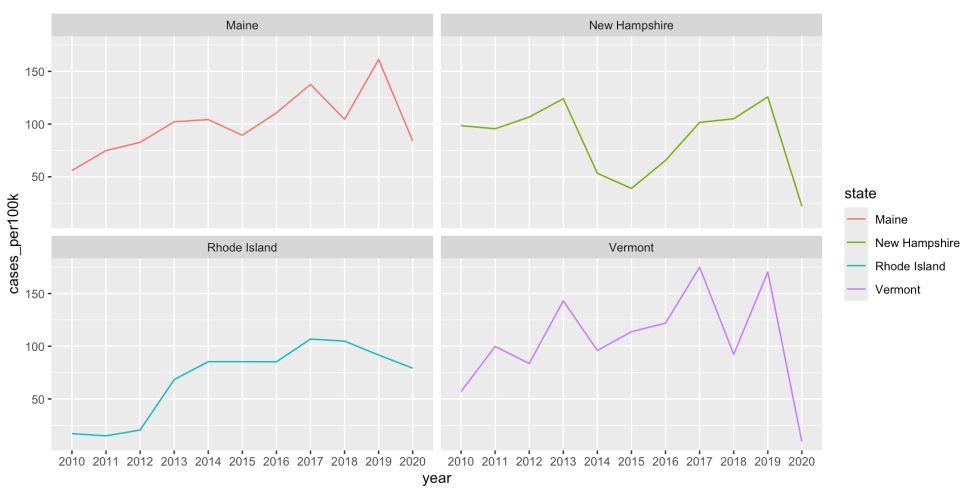
Stacked area charts are *not good* for studying the evolution of individual groups

It is super **challenging to subtract the height of groups from one another at any / each given point in time**. For example, both of the charts below show the same data (Lyme disease incidence (# cases / 100k people) for Maine, New Hampshire, Rhode Island, and Vermont):

```
lyme_pop |>
  filter(state %in% c("Maine", "Rhode
  ggplot(aes(x = year, y = cases_per10
  geom_area()
```




```
lyme_pop |>
  filter(state %in% c("Maine", "Rhode
  ggplot(aes(x = year, y = cases_per10
  geom_line() +
  facet_wrap(~state)
```



See you next week!

~ This is the end of Lesson 3 (of 3) ~