Week 10: Trends

October 28, 2026

SHOWING CHANGE OVER TIME

Time is the most common x-axis in data analysis. The question a trend chart answers is *how did this change?*, and the right geom depends on what part of the change you want the reader to see: the individual values, the slope between them, the cumulative total, or the pattern across many groups at once. This week covers three cases: **one variable** over time, **animating** the time dimension, and **many categories** over time.

SINGLE VARIABLES

Geom	Emphasizes	Watch out
Points	The observed values	Trend can be hard to see
Points + line	Values *and* trend	Draw line first
Line	The overall trend	Hides sparse data
Line + area	Cumulative amount	y-axis must start at 0
Smoothed line	Modeled trend	Can hide outliers
Smooth + points	Trend vs. outliers	Fade the points
Bars	Each data point	Not the slope

- **Sparse points mislead as a line.** A line drawn through five years of data implies you know what happened in between. With few points, show the points.
- **Bars** say "compare these values"; **lines** say "follow the slope."
- **Layer order matters.** ggplot draws geoms in the order you add them: put the line before the points, and the area before the line.

```
ggplot(milk_ca, aes(x = year, y = milk_produced)) +
  geom_area(fill = 'steelblue', alpha = 0.5) +
  geom_line(color = 'steelblue', size = 1) +
  theme_half_open()
```

```
ggplot(milk_ca, aes(x = year, y = milk_produced)) +
  geom_point(color = 'grey', alpha = 0.9) +
  geom_smooth(color = 'steelblue', se = FALSE)
```

Smoothing over the points lets the reader check whether a few outliers are driving the trend line.

ANIMATION

Animation emphasizes the **change over time** itself: Gapminder's moving bubbles, climate spirals, bar chart races. Use it for the time dimension, and always pair it with a static version for print.

The `{gganimate}` recipe: build a normal static ggplot, add a `transition_*()` layer, then `animate()` and `anim_save()`.

Function	Use for
`transition_reveal(year)`	Draw a line in over time
`transition_time(year)`	Move points between years
`labs(title = "{frame_time}")`	Year label that updates
`view_follow(fixed_x = TRUE)`	Rescale axes each frame
`animate(..., end_pause = 15)`	Hold the final frame
`anim_save(path)`	Save the last animation

```
library(gganimate)

anim <- milk_region_anim_plot +
  transition_reveal(year)

animate(
  anim,
  end_pause = 15,
  duration = 10,
  renderer = magick_renderer()
)
anim_save(file.path('figs', 'milk_animation.gif'))
```

- **Add an end pause.** Without one the GIF loops straight back to the start and the reader never gets to study the final state.
- **The time variable must be an integer** for `transition_time()`.
- New packages this week: `viridis`, `gganimate`, `magick`.

MANY CATEGORIES

With more than a few groups, points plus lines turn into spaghetti. Options, roughly from few to many categories:

Chart	Best when
Lines only, directly labeled	A handful of groups
Stacked area	The **total** trend matters most
All lines grey + average	Dozens of groups, one story
Heat map	Many groups, or nested time

Directly label the lines

Drop the legend and put each label at the end of its line. Label only the last year, and widen the x-axis to make room:

```
milk_region %>%
  ggplot(aes(x = year, y = milk_produced,
             color = region)) +
  geom_line(size = 1) +
  geom_text_repel(
    data = milk_region %>% filter(year == max(year)),
    aes(label = region),
    hjust = 0, nudge_x = 1, direction = "y",
    segment.color = NA
  ) +
  scale_x_continuous(
    expand = expansion(add = c(1, 13))
  ) +
  theme(legend.position = 'none')
```

Alternative: `geom_textline()` from `{geomtextpath}` writes the label *into* the line itself.

Stacked area

Each band is a group; the top edge is the total. Only the bottom band sits on a common baseline, so use it when the *total* is the story.

```
internet_region %>%
  ggplot() +
  geom_area(aes(x = year, y = numUsers,
                fill = region)) +
  scale_fill_viridis(discrete = TRUE)
```

Highlight the average

Draw every state in light grey with `group = state`, then overlay the national mean in black. Mark events with `geom_vline()` plus `annotate()` (e.g., the 1963 measles vaccine).

```
measles <- us_diseases %>%
  filter(disease == 'Measles') %>%
  mutate(rate = (count / population) * 10000) %>%
  group_by(year) %>%
  mutate(
    mean_rate = sum(count) / sum(population) * 10000
  )

ggplot(measles) +
  geom_line(aes(x = year, y = rate, group = state),
            color = 'grey', alpha = 0.3) +
  geom_line(aes(x = year, y = mean_rate),
            size = 0.8) +
  geom_vline(xintercept = 1963, linetype = 'dashed') +
  annotate('text', x = 1964, y = 150, hjust = 0,
           label = 'Vaccine introduced in 1964')
```

The mean is computed from totals, `sum(count) / sum(population)`, not by averaging the state rates, so big and small states are weighted correctly.

Heat maps
`geom_tile(aes(x = year, y = state, fill = rate))`. Great for multiple divisions of time (day × hour, week × year, like a GitHub activity grid).

- Sort rows with `fct_reorder(state, rate)`, not alphabetically.
- `scale_fill_viridis(option = 'inferno', direction = -1)` gives a perceptually even, color-blind-safe scale.
- **Large variation?** Add `trans = 'sqrt'`. A linear color scale lets a few extreme values wash out everything else.

```
measles %>%
  mutate(state = fct_reorder(state, rate)) %>%
  ggplot() +
  geom_tile(aes(x = year, y = state, fill = rate),
            color = 'grey80') +
  scale_x_continuous(expand = c(0, 0)) +
  scale_fill_viridis(option = 'inferno',
                     direction = -1, trans = 'sqrt') +
  guides(fill = guide_colorbar(reverse = TRUE)) +
  theme_minimal()
```

Pick the geom for the part of the change you want seen. Points for values, lines for slope, area for totals, a grey crowd plus one dark line for "the typical case," a heat map when there are too many groups to follow. An agent will default to a colored line per group with a legend; it's your job to ask for the right one.

DIRECTING AN AGENT

- *"Plot these as lines, label each line at its last year, and remove the legend."*
- *"Show all 50 states as faint grey lines with the national average in black, and mark 1963 with a dashed vertical line."*
- *"Make a heat map of rate by year and state, sorted by rate, with a square-root viridis fill scale."*
- *"Animate this line chart with `gganimate`, revealing by year, with a pause at the end."*

Then check: does a line imply data you don't have? Does the area start at 0? Are outliers driving the smooth?

TREND CHART SYMPTOMS & FIXES

Symptom	Fix
Line connects years with no data	Show points, or bars
Points drawn under the line	Add `geom_point()` last
Area chart y-axis starts above 0	Start at 0, or use a line
Smooth trend hides odd years	Overlay faded points
Spaghetti of colored lines	Grey crowd + one dark line
Eyes bounce to the legend	`geom_text_repel()` labels
Labels cut off at the right	Widen `expand` on x
Heat map is one color	`trans = 'sqrt'` on the fill
Heat map rows alphabetical	`fct_reorder()` the rows
GIF loops before you can read it	`end_pause` in `animate()`

PRACTICE

The practice file uses two data sets: `nasa_global_temps.csv` (one variable: plot points, line, smooth, and area, then animate one) and `us_covid.csv` (many categories: a labeled CA vs. NY line chart, a stacked area of CA, NY, and all other states via `fct_other()`, and a heat map).

BEFORE NEXT CLASS

- **HW9: Polishing Charts**, due **Nov 2**.
- **Initial Report**, due **Nov 1** (6% of grade).
- **Quiz 5** on Nov 11, covering Weeks 10–11.

You should be able to:

- Choose among points, lines, areas, smooths, and bars for one trend.
- Explain why sparse data shouldn't be drawn as a line.
- Animate a chart with `gganimate` and save it as a GIF.
- Directly label lines and drop the legend.
- Show many categories with a grey crowd plus average, or a heat map.
- Pick a linear vs. square-root color scale for a heat map.