Week 6: Graphing trends

EMSE 4197 | John Paul Helveston | February 19, 2020

Quiz 2

25 minutes

R tip of the week: Code outline in RStudio

Today's data

Old:

```
gapminder         <- read_csv(here::here('data', 'gapminder.csv'))
milk_production   <- read_csv(here::here('data', 'milk_production.csv'))
global_temps      <- read_csv(here::here('data', 'nasa_global_temps.csv'))
```

New:

```
internet_country <- read_csv(here::here('data', 'internet_users_country.csv'))
internet_region  <- read_csv(here::here('data', 'internet_users_region.csv'))
hotdogs          <- read_csv(here::here('data', 'hot_dog_winners.csv'))
us_diseases      <- read_csv(here::here('data', 'us_contagious_diseases.csv'))
```

Graphing trends

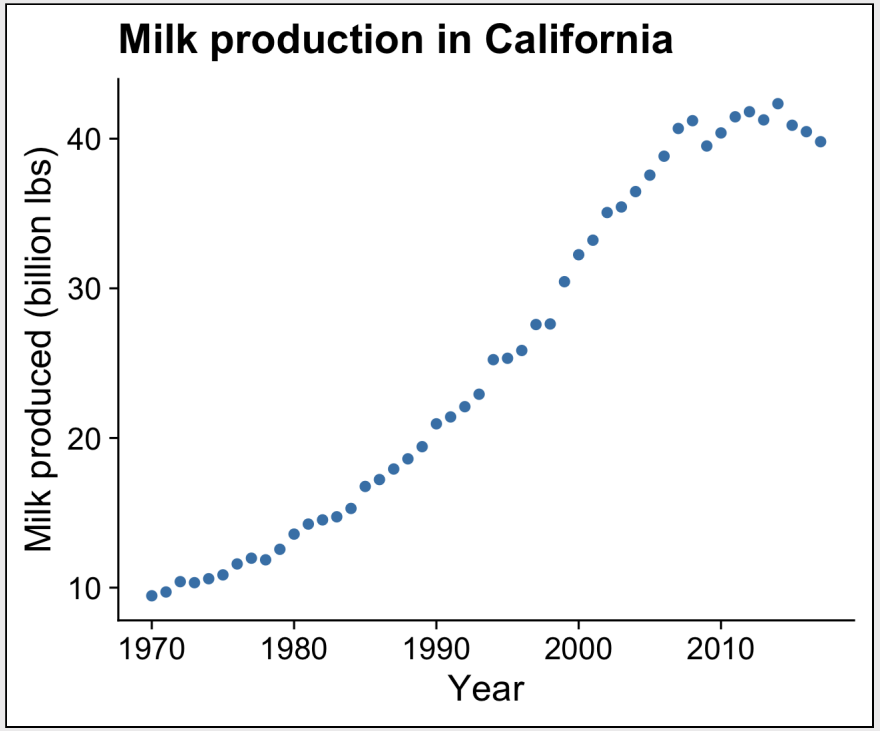
1. Single variables
2. Multiple variables
3. Animations

Graphing trends

1. Single variables
2. Multiple variables
3. Animations

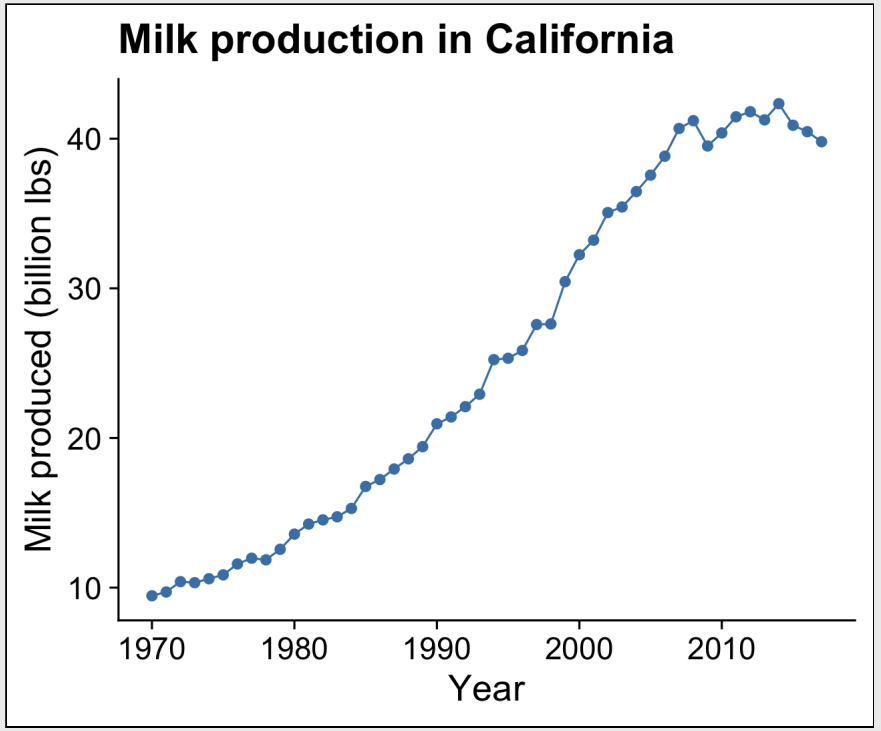
Points

Plotting the data points is a good starting point



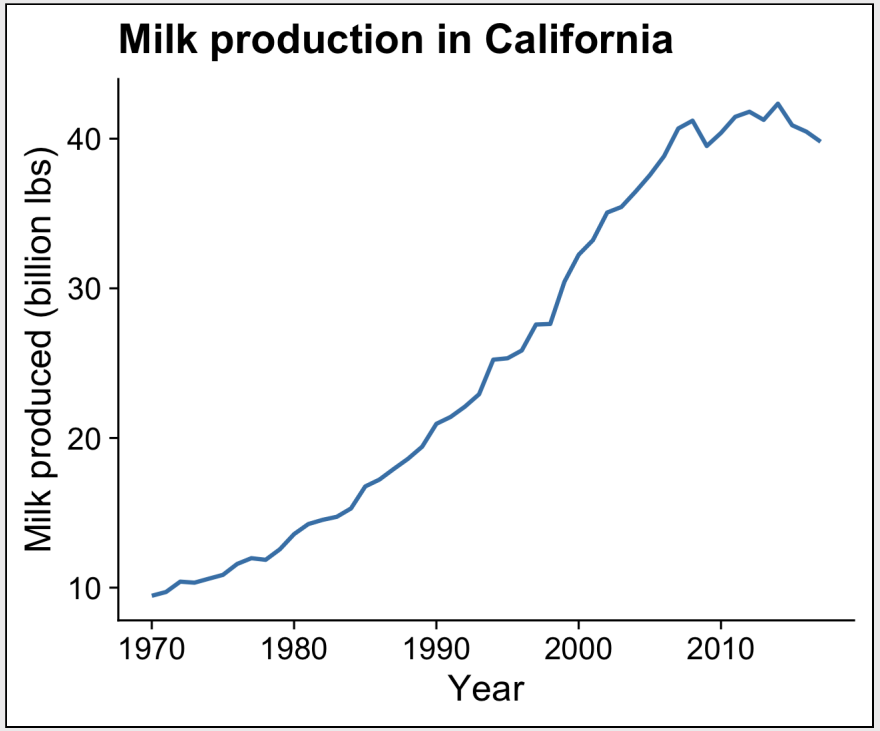
Points + line

Adding lines between the points helps see the overall trend



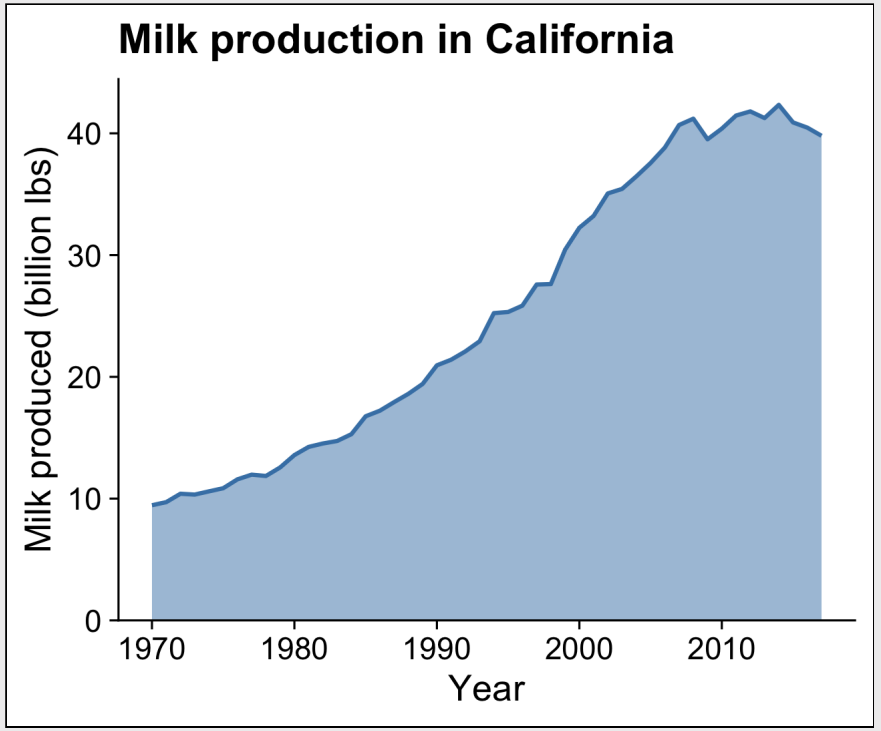
Line

Omitting the points emphasizes the overall trend

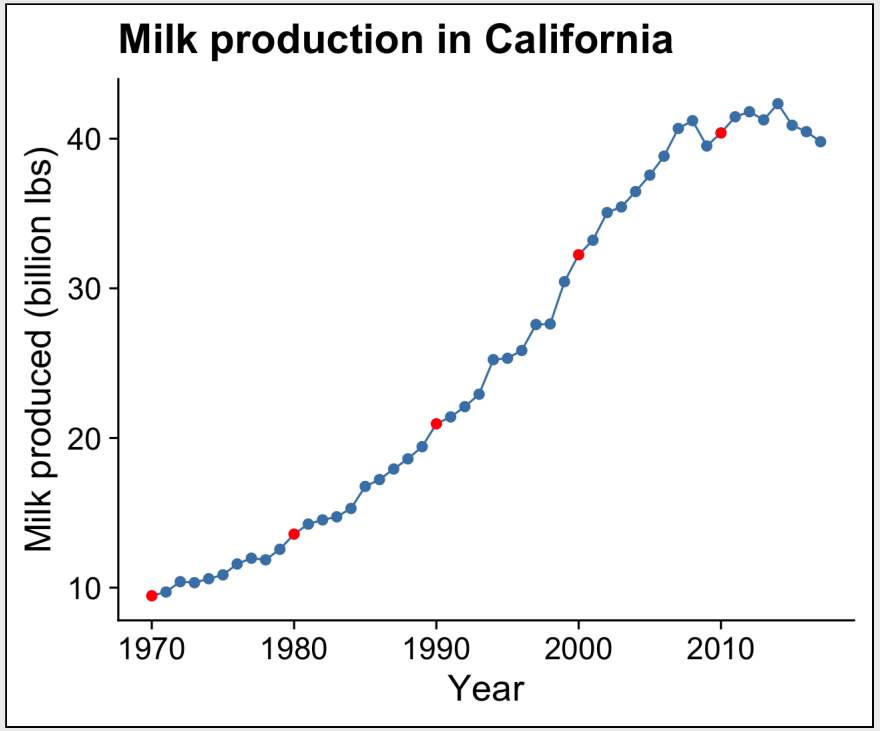


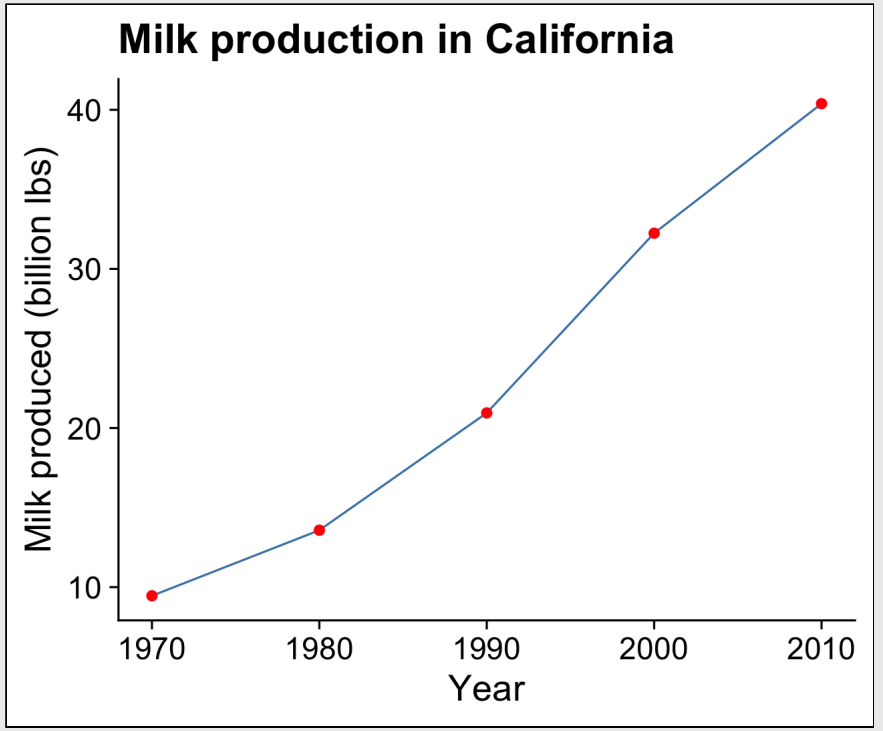
Line + area

Filling area below line further emphasizes the trend, but the y-axis must start at zero



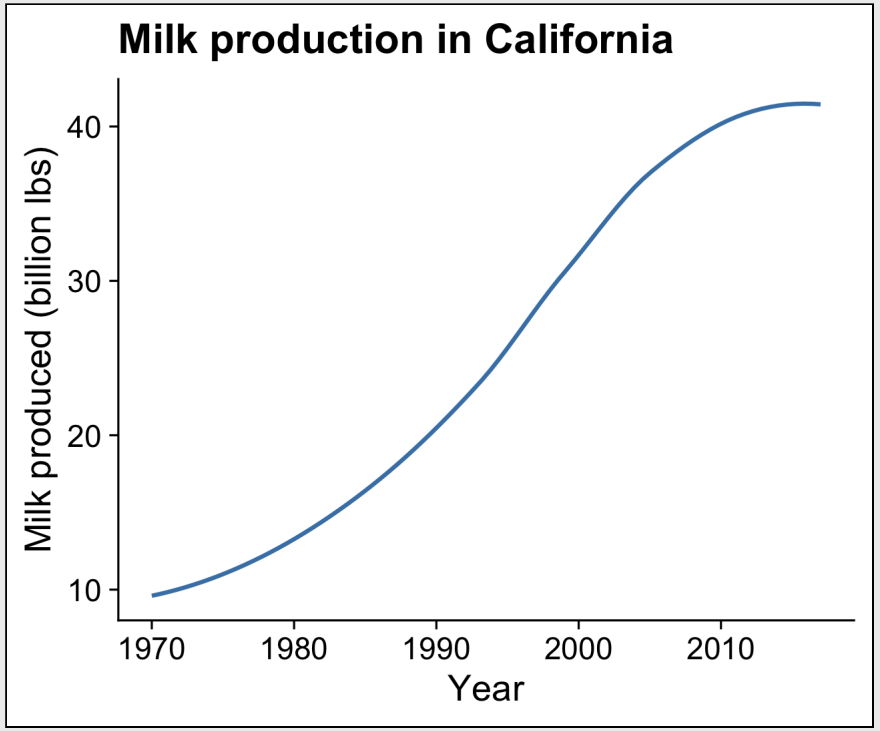
If points are too sparse, a line can be misleading





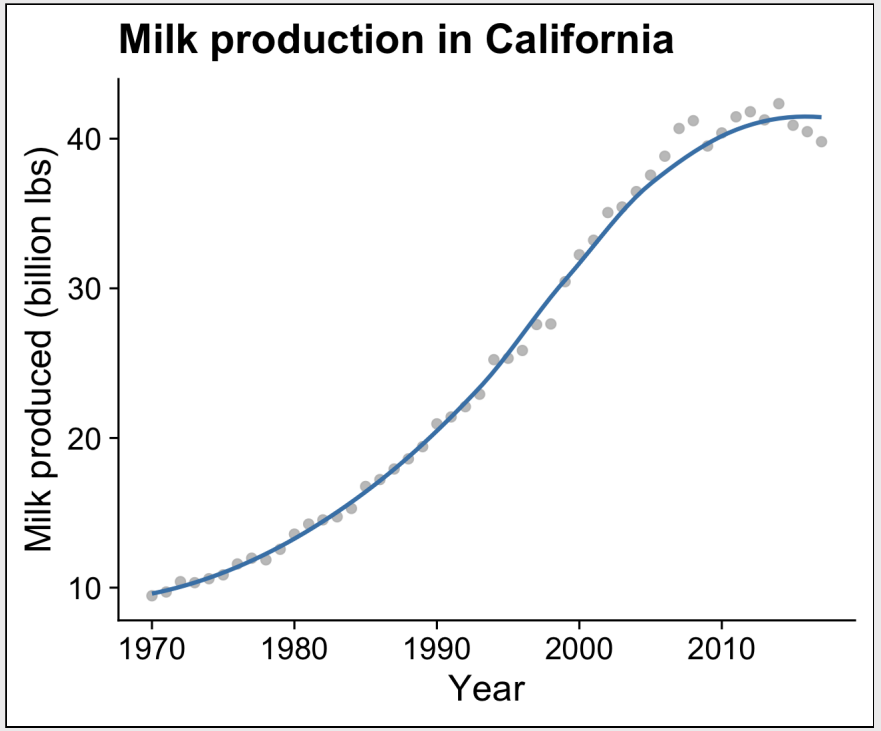
Smoothed line

Adding a "smoothed" line shows a modeled representation of the overall trend

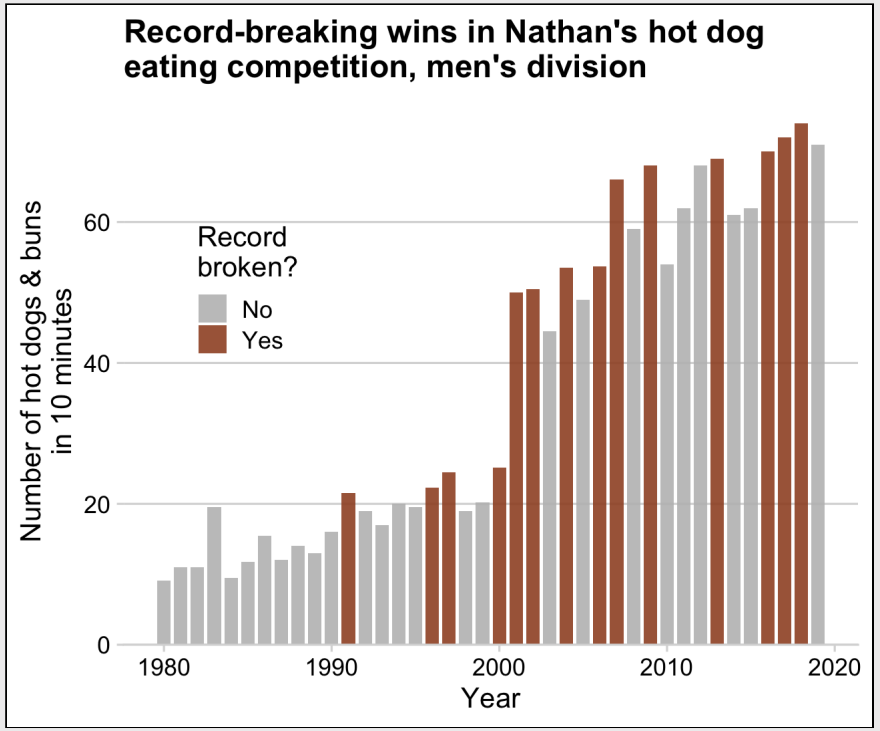


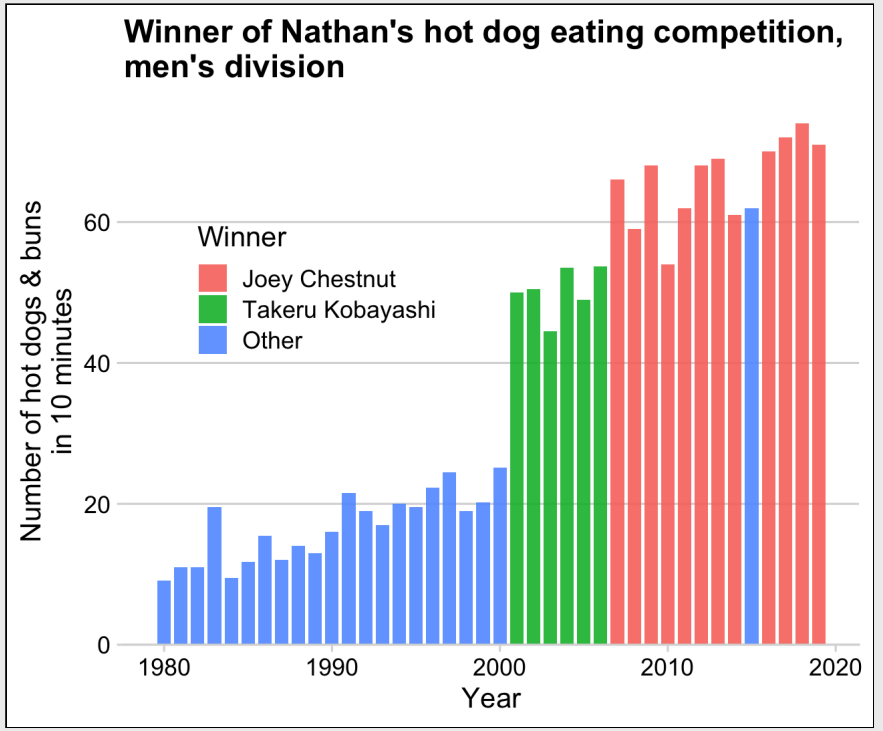
Smoothed line + points

Putting the smoothed line over the data points helps show whether **outliers** are driving the trend line



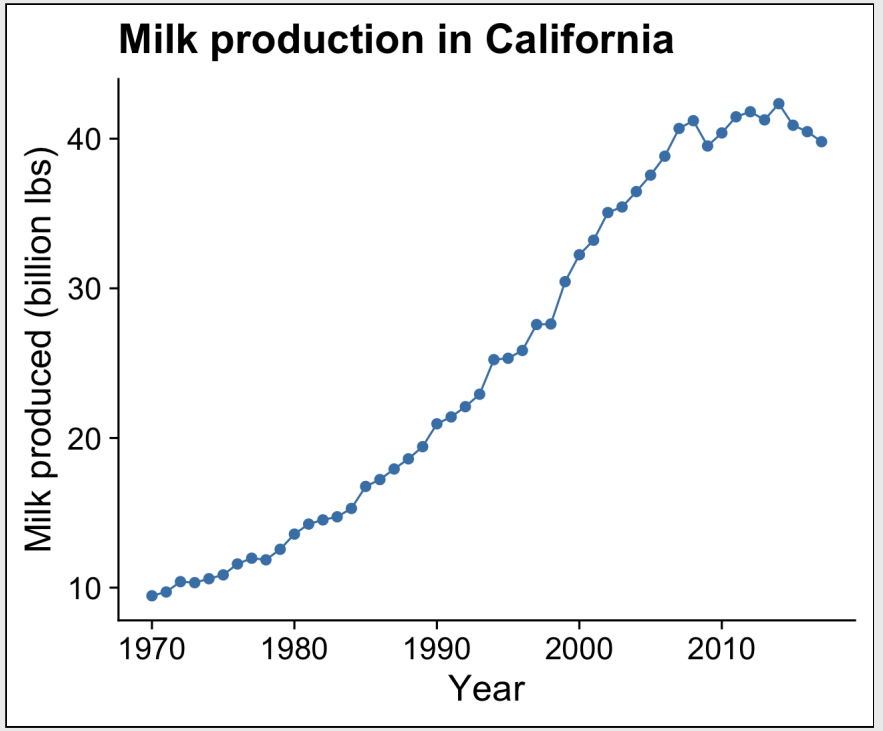
Bars are useful when emphasizing the **data points** rather than the **slope between them**





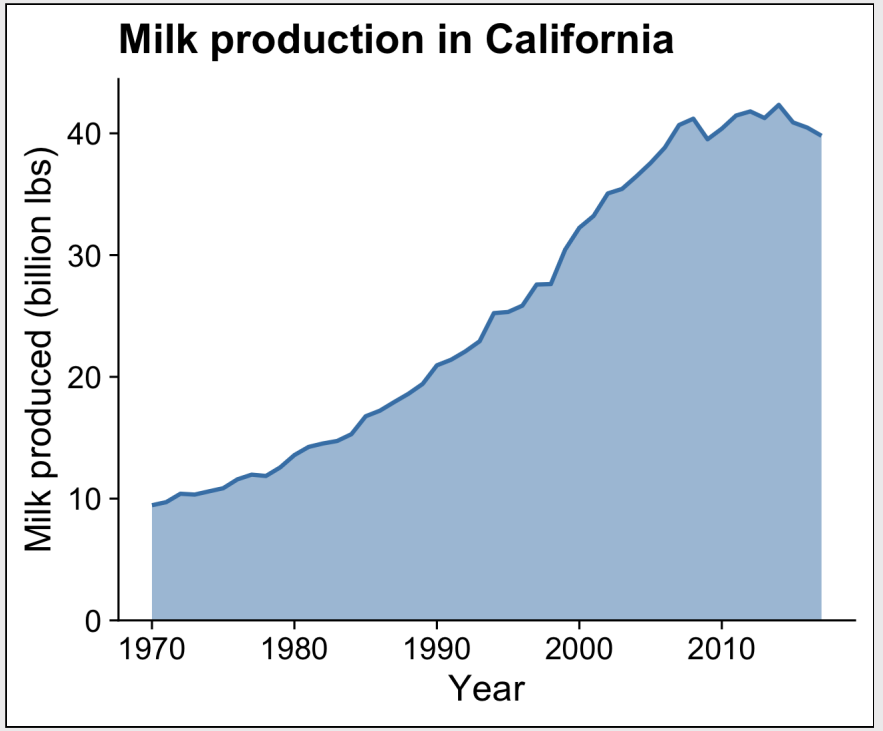
How to: **Points + line**

```
ggplot(milk_ca,
    aes(x = year, y = milk_produced)) +
    geom_point(color = 'steelblue', size = 2) +
    geom_line(color = 'steelblue', size = 0.5) +
    theme_half_open(font_size = 18) +
    labs(x = 'Year',
         y = 'Milk produced (billion lbs)',
         title = 'Milk production in California')
```



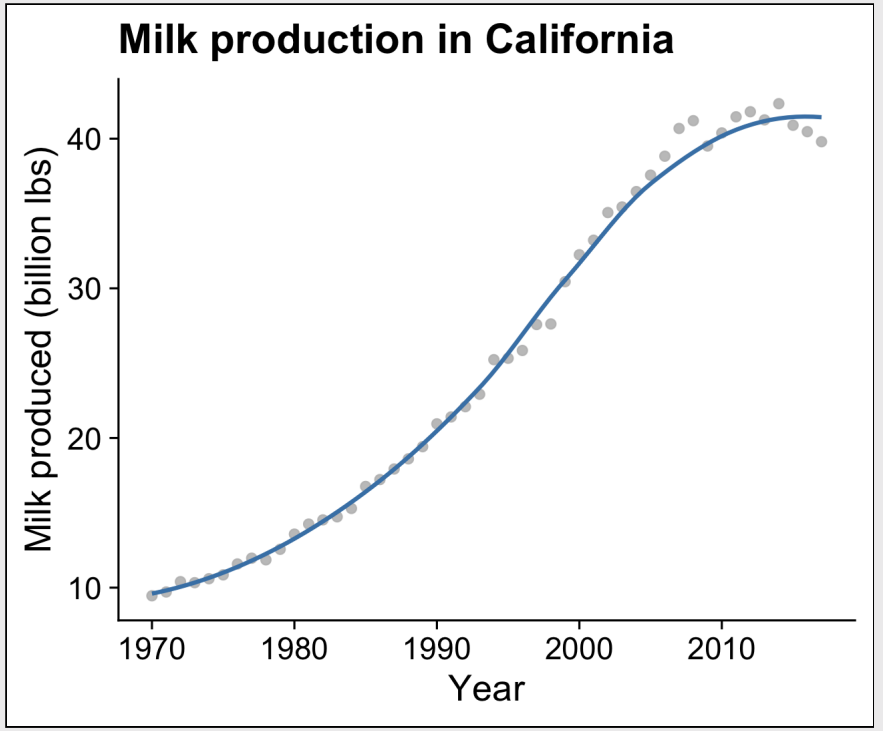
How to: **Line + area**

```
ggplot(milk_ca,
    aes(x = year, y = milk_produced)) +
    geom_line(color = 'steelblue', size = 1) +
    geom_area(fill = 'steelblue', alpha = 0.5) +
    scale_y_continuous(
        expand = expand_scale(mult = c(0, 0.05))) +
    theme_half_open(font_size = 18) +
    labs(x = 'Year',
         y = 'Milk produced (billion lbs)',
         title = 'Milk production in California')
```



How to: **Smoothed line + points**

```
ggplot(milk_ca,
    aes(x = year, y = milk_produced)) +
    geom_point(color = 'grey',
               size = 2,
               alpha = 0.9) +
    geom_smooth(color = 'steelblue',
                size = 1,
                se = FALSE) +
    theme_half_open(font_size = 18) +
    labs(x = 'Year',
         y = 'Milk produced (billion lbs)',
         title = 'Milk production in California')
```

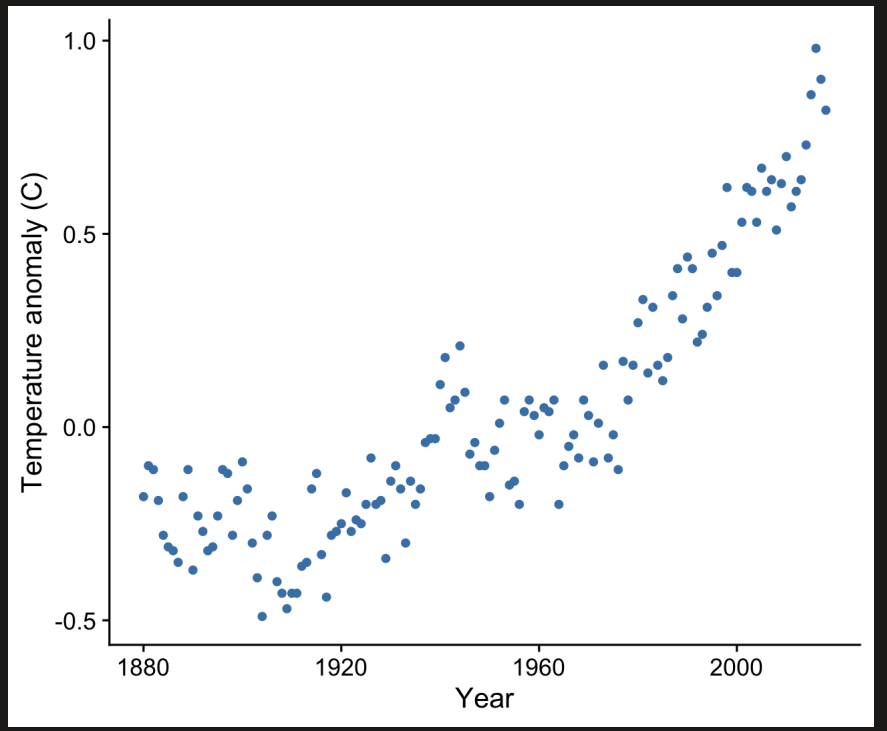


Your turn

Use the `nasa_global_temps.csv` data to explore ways to visualize the change in average global temperatures.

Consider using:

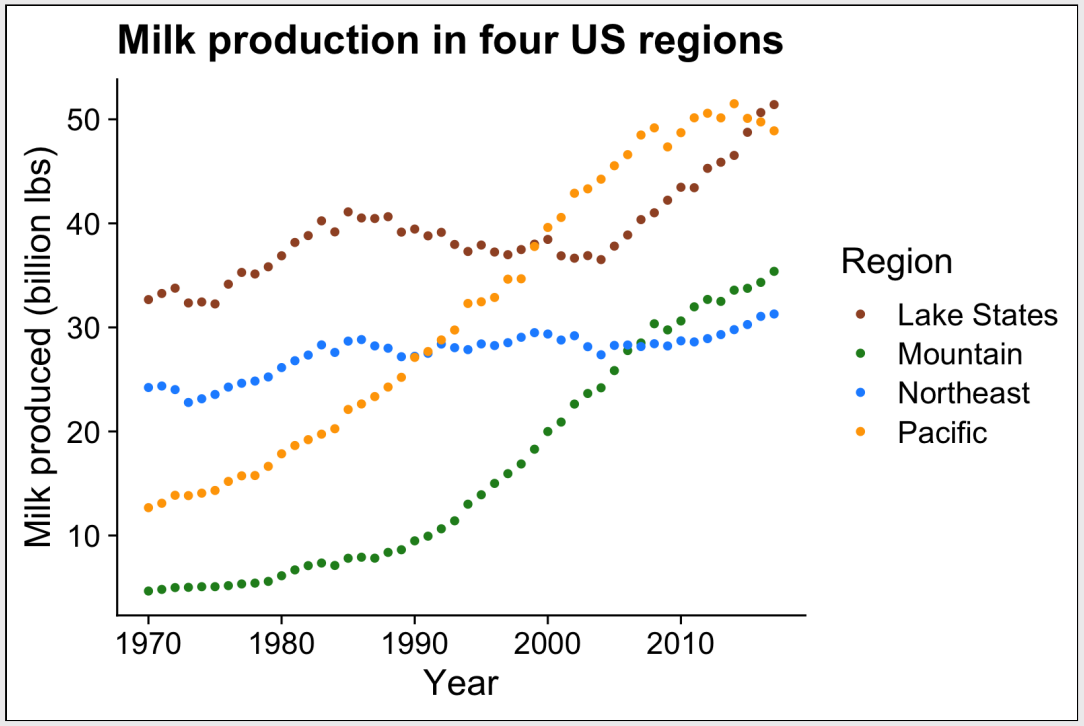
- points
- lines
- areas
- smoothed lines

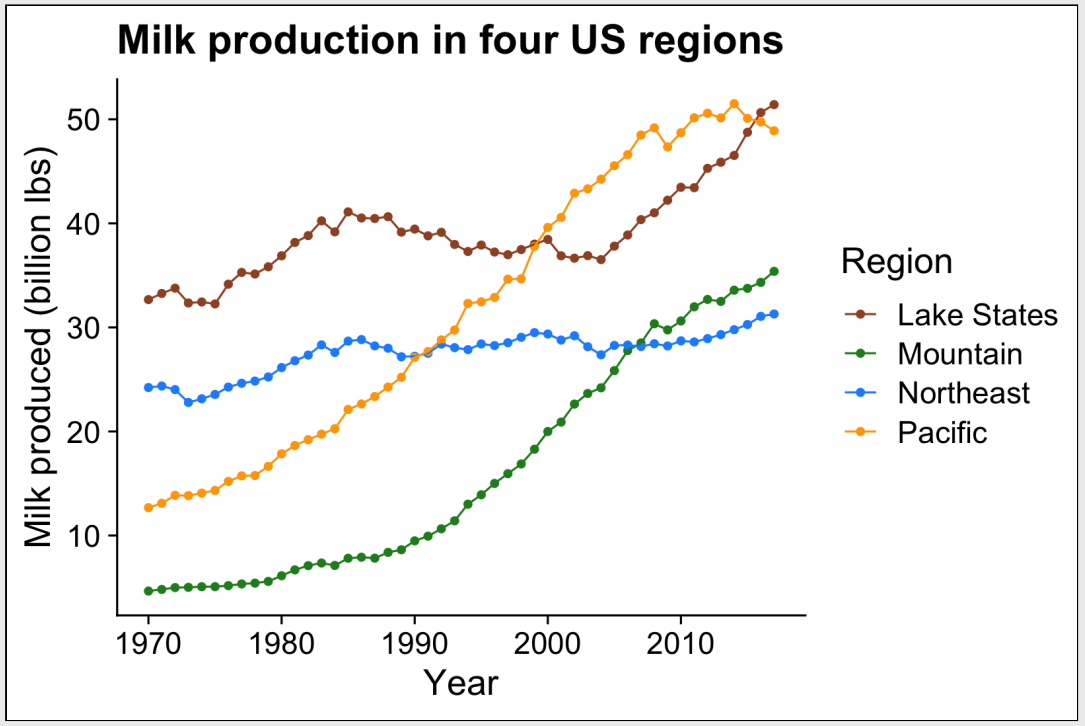


Graphing trends

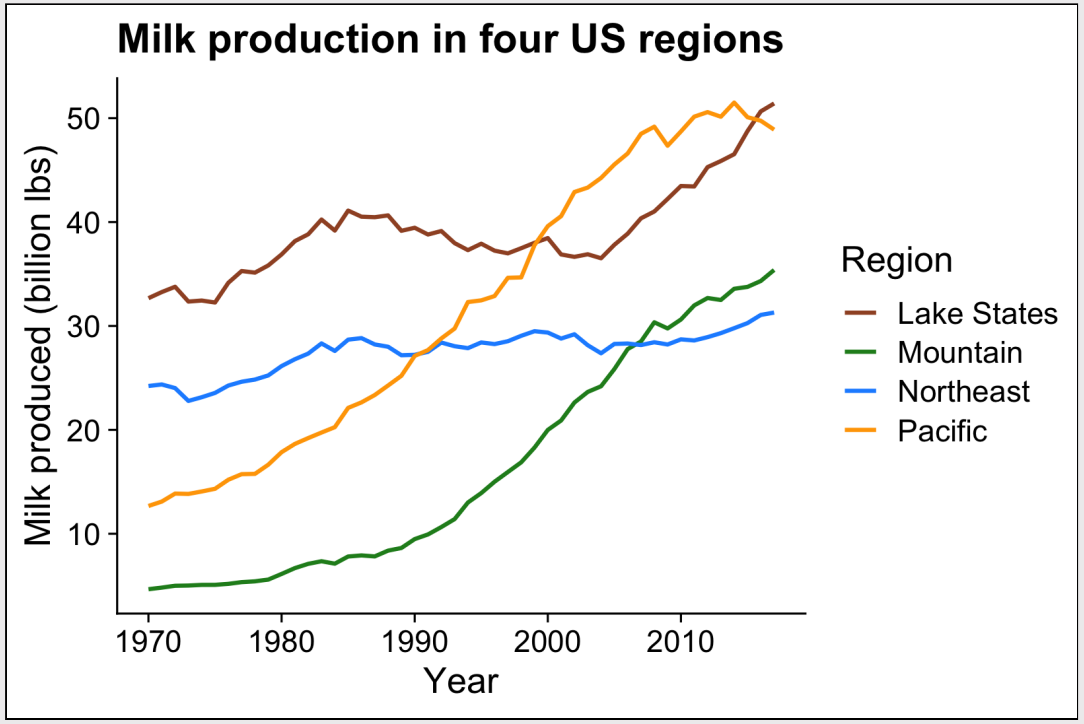
1. Single variables
2. Multiple variables
3. Animations

With multiple categories, points & lines can get messy

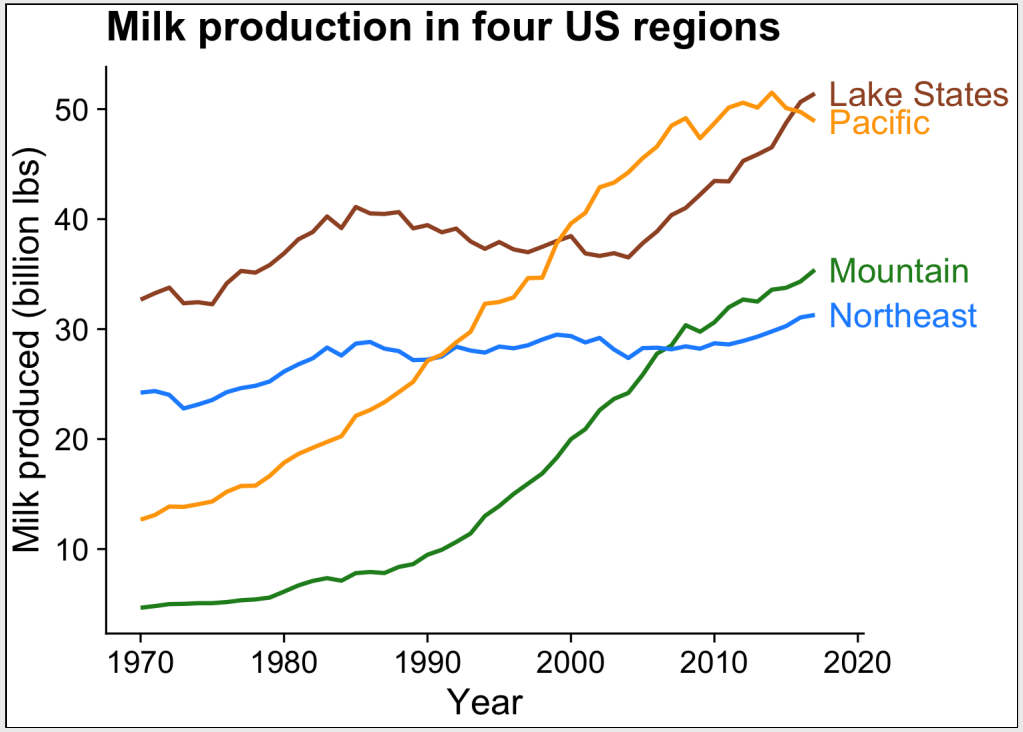




Better: Lines alone makes distinguishing trends easier

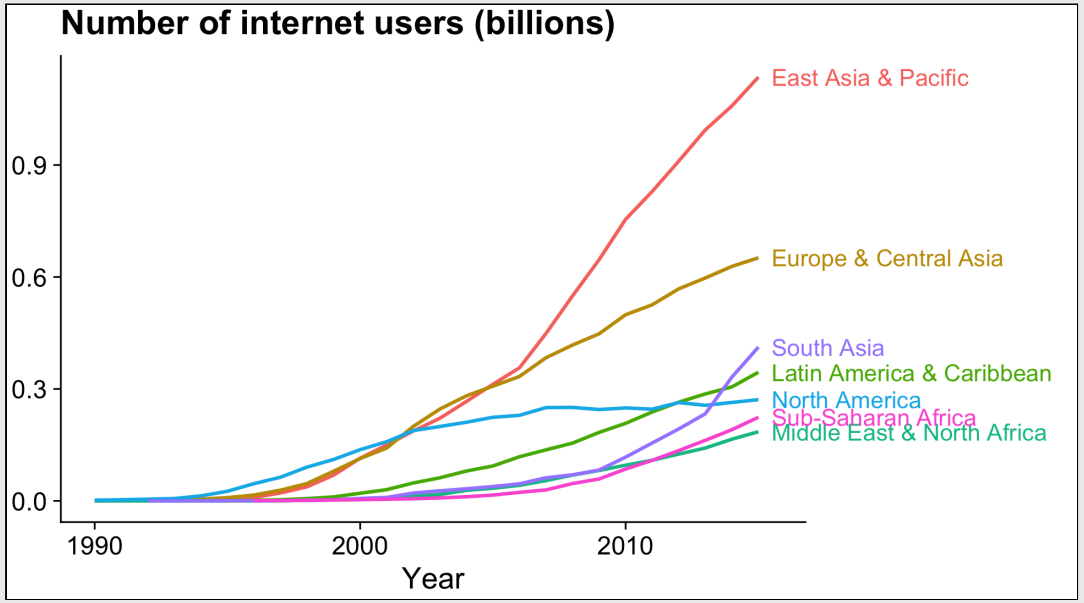


Even better: Directly label lines to remove legend

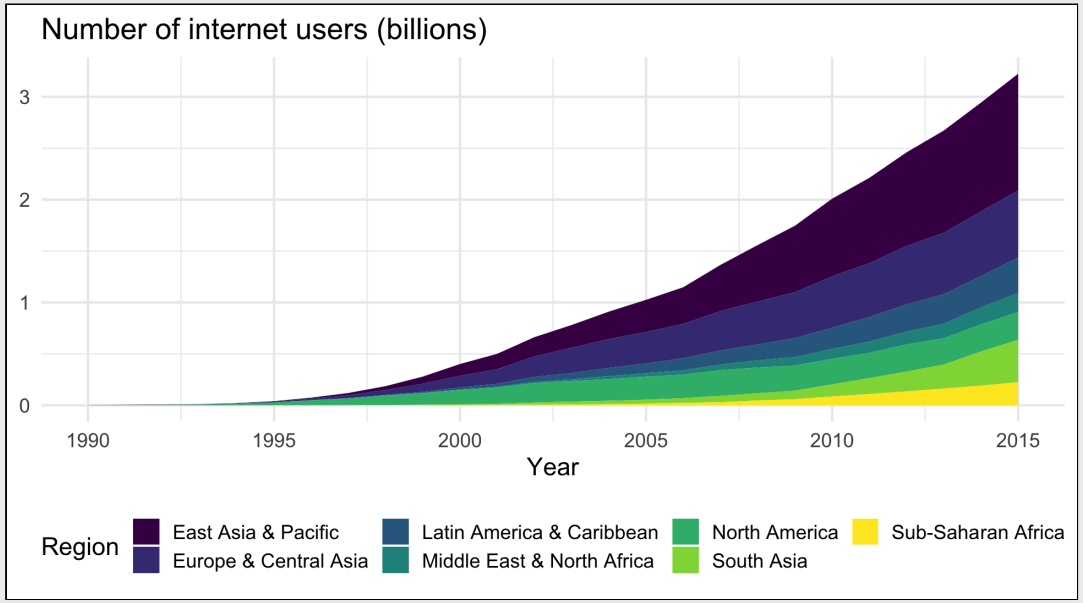


If goal is to communicate the **overall** trend,
consider a stacked area chart

Highlights **regional** trends



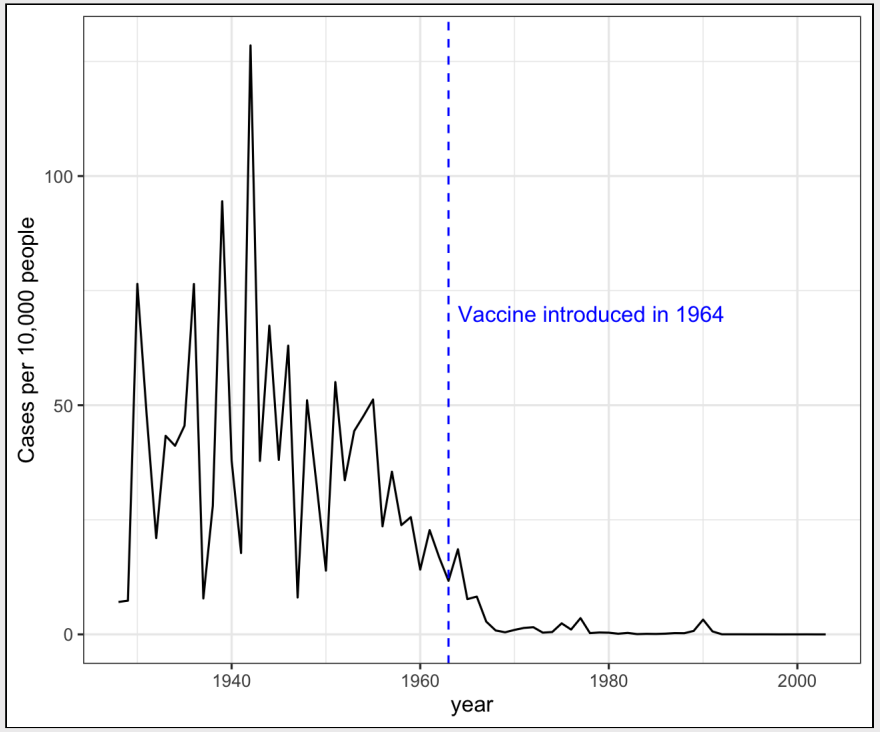
Highlights **overall** trend



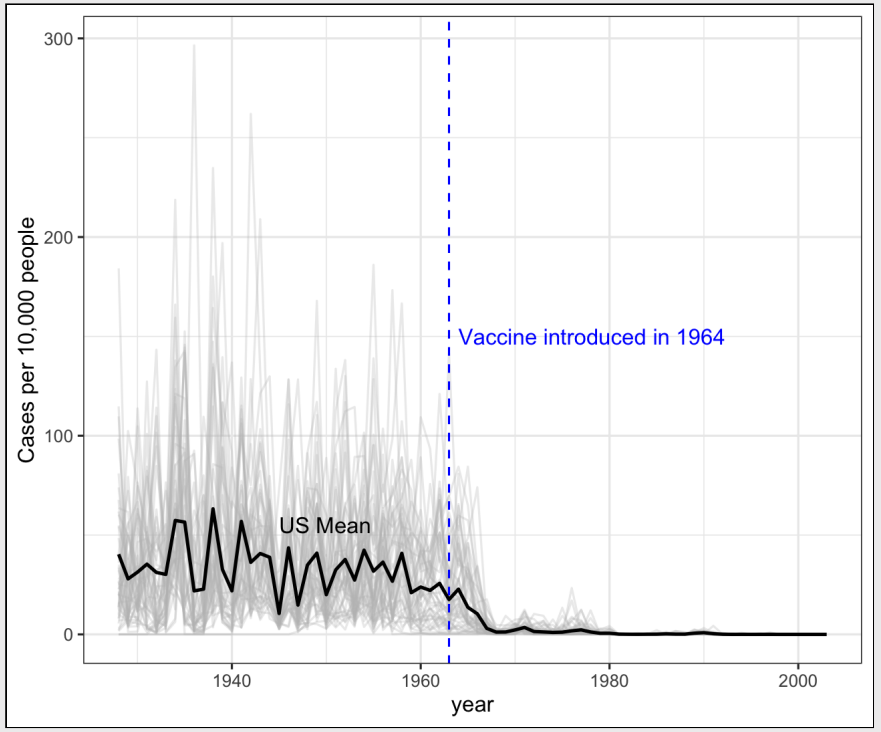
If you have **lots** of categories:

1) Plot all the data with the average highlighted

Measles in California



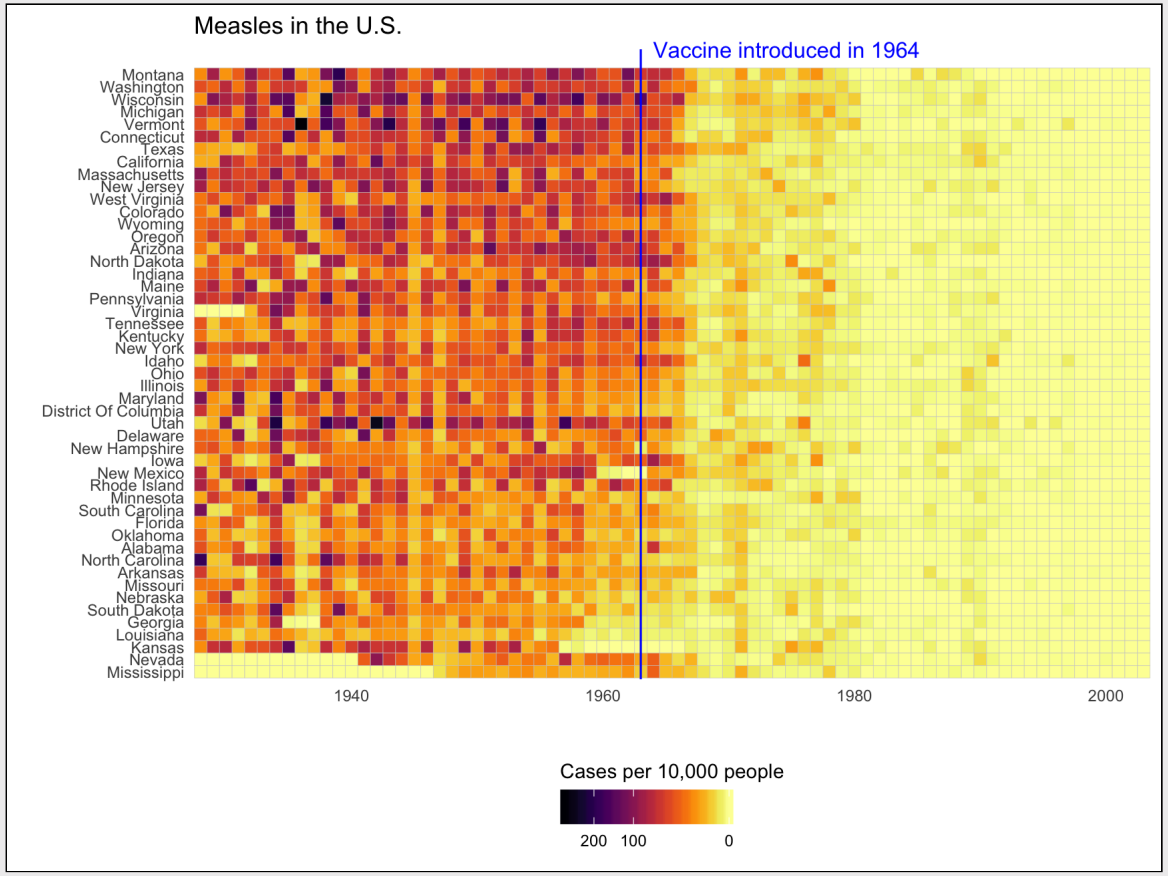
Measles in all 50 states



If you have **lots** of categories:

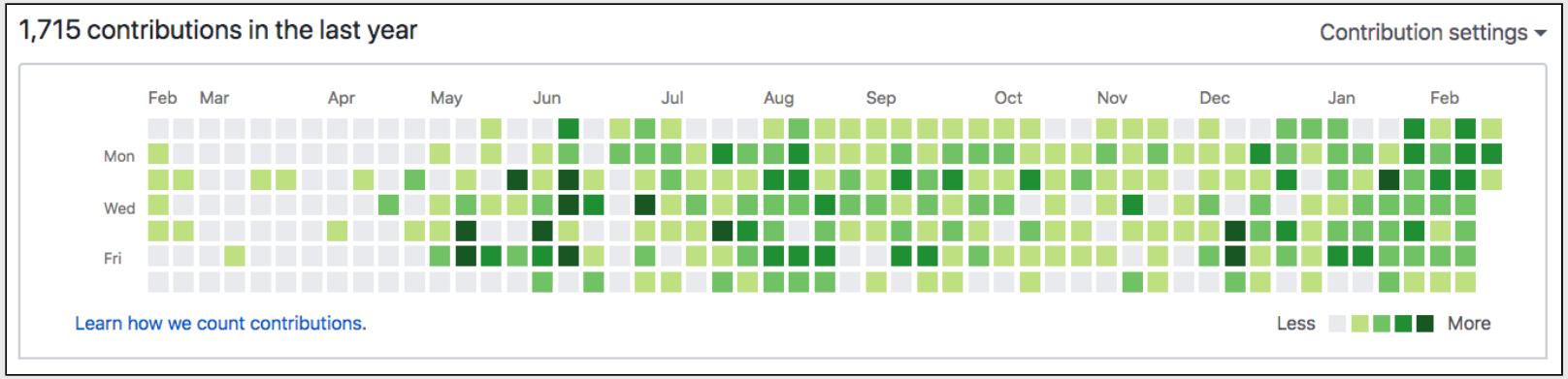
1) Plot all the data with the average highlighted

2) Plot all the data with a heat map



Heatmaps are great for multiple divisions of time

My activity on Github:



Traffic fatalities

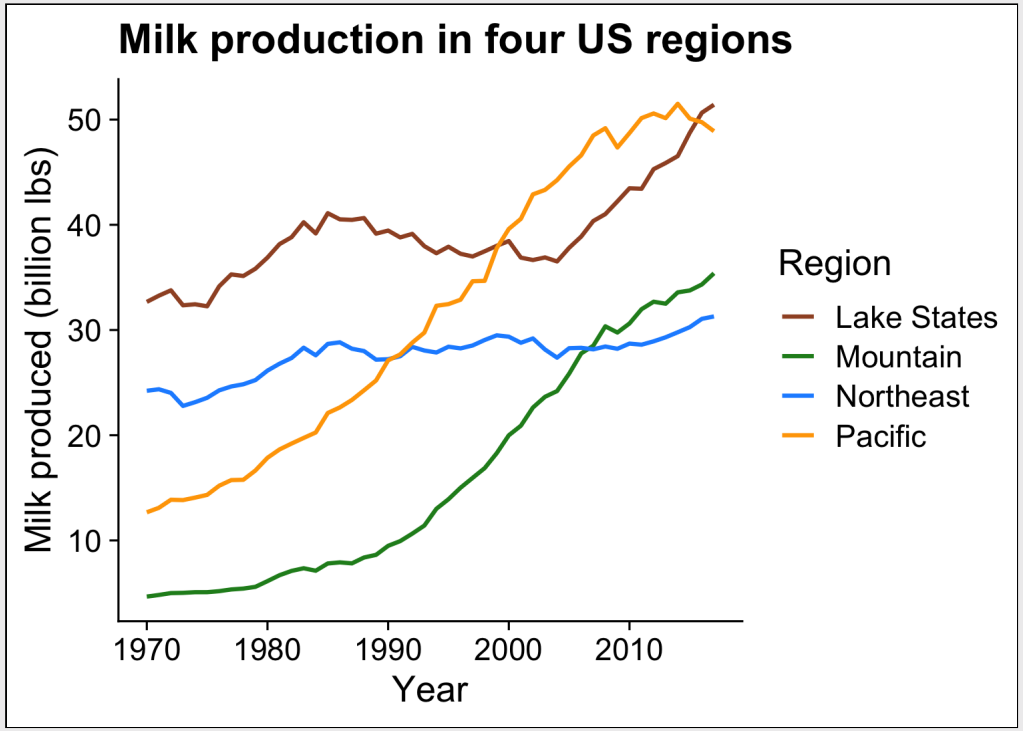
How to:
Directly label lines

Format the data

```
milk_region <- milk_production %>%
    filter(region %in% c(
        'Pacific', 'Northeast', 'Lake States', 'Mountain')) %>%
    group_by(year, region) %>%
    summarise(milk_produced = sum(milk_produced))
```

Make the line plot

```
ggplot(milk_region,
    aes(x = year, y = milk_produced, color = region)) +
    geom_line(size = 1) +
    scale_color_manual(values = c(
        'sienna', 'forestgreen', 'dodgerblue', 'orange')) +
    theme_half_open(font_size = 18) +
    labs(x     = 'Year',
         y     = 'Milk produced (billion lbs)',
         color = 'Region',
         title = 'Milk production in four US regions')
```



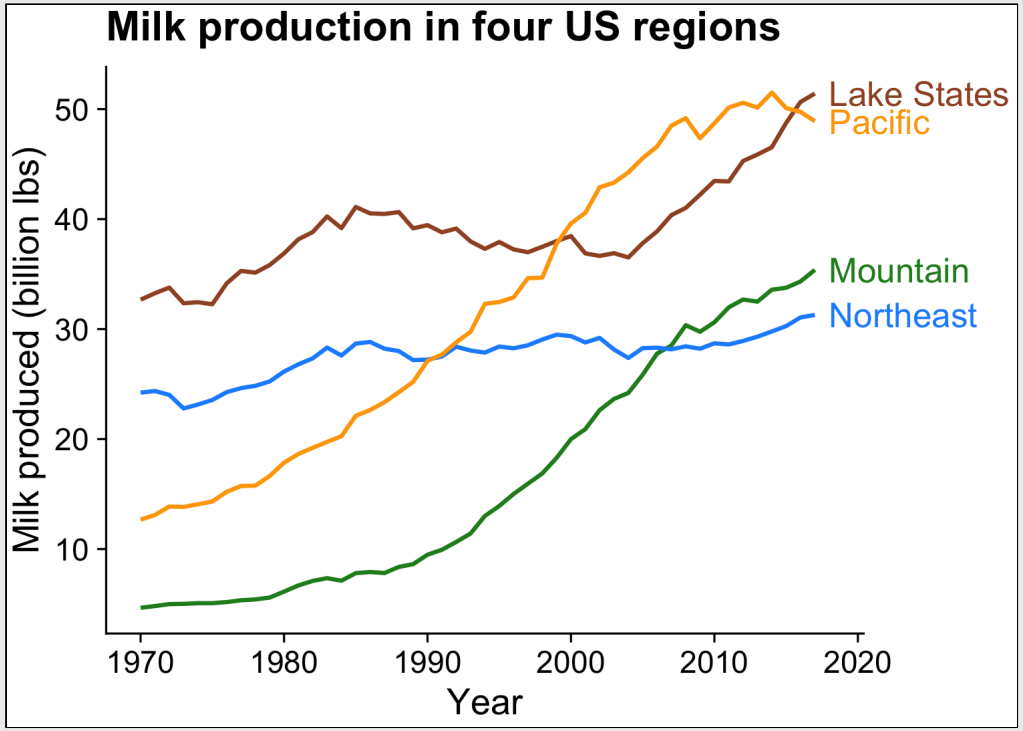
How to:
Directly label lines

Format the data

```
milk_region <- milk_production %>%
    filter(region %in% c(
        'Pacific', 'Northeast', 'Lake States', 'Mountain')) %>%
    group_by(year, region) %>%
    summarise(milk_produced = sum(milk_produced)) %>%
    # Create the labels
    ungroup() %>%
    mutate(label = ifelse(
           year == max(year), region, NA))
```

Make the line plot

```
ggplot(milk_region,
    aes(x = year, y = milk_produced, color = region)) +
    geom_line(size = 1) +
    # Add labels to plot
    geom_text(aes(label = label),
              hjust = 0, nudge_x = 1, size = 6) +
    scale_color_manual(values = c(
        'sienna', 'forestgreen', 'dodgerblue', 'orange')) +
    # Create space for labels on right side
    coord_cartesian(clip = 'off') +
    theme(legend.position = 'none',
          # Remember "trouble": T    R    B    L
          plot.margin = margin(0.1, 2.7, 0.1, 0.1, "cm")) +
    theme_half_open(font_size = 18) +
    labs(x     = 'Year',
         y     = 'Milk produced (billion lbs)',
         title = 'Milk production in four US regions')
```



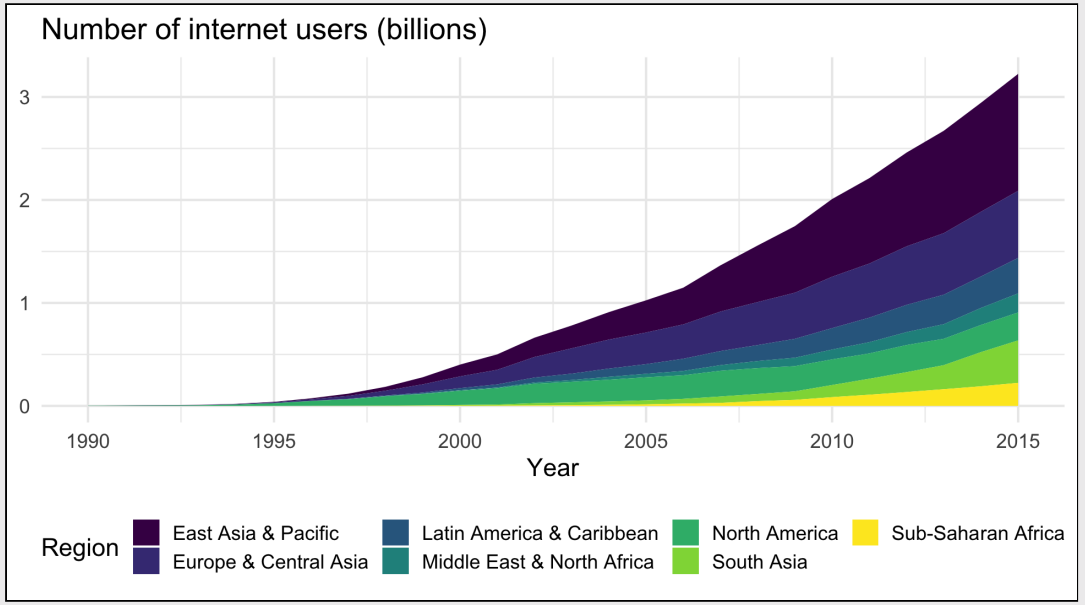
How to: **Stacked area**

Format the data

```
internet_region_summary <- internet_region %>%
    mutate(numUsers = numUsers / 10^9)
```

Make the plot

```
ggplot(internet_region_summary) +
    geom_area(aes(x = year, y = numUsers,
        fill = region)) +
    # Nice colors from "viridis" library:
    scale_fill_viridis(discrete = TRUE) +
    theme_minimal(base_size = 15) +
    theme(legend.position = 'bottom') +
    labs(x = 'Year', y = NULL,
         fill  = 'Region',
         title = 'Number of internet users (billions)'
```



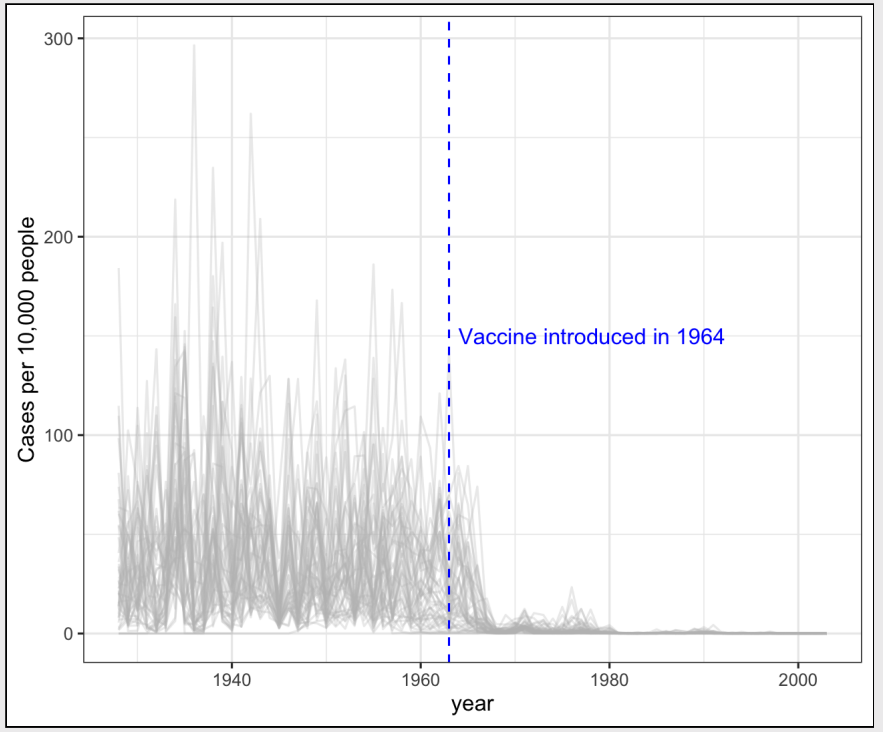
How to: **Average line overlay**

Format the data

```
measles <- us_diseases %>%
    filter(!state %in% c("Hawaii", "Alaska"),
           disease == 'Measles',
           ! is.na(count),
           ! is.na(population)) %>%
    mutate(
        rate = (count / population) * 10000,
        state = fct_reorder(state, rate)) %>%
    filter(! is.na(rate))
```

Make all the state lines in light grey color

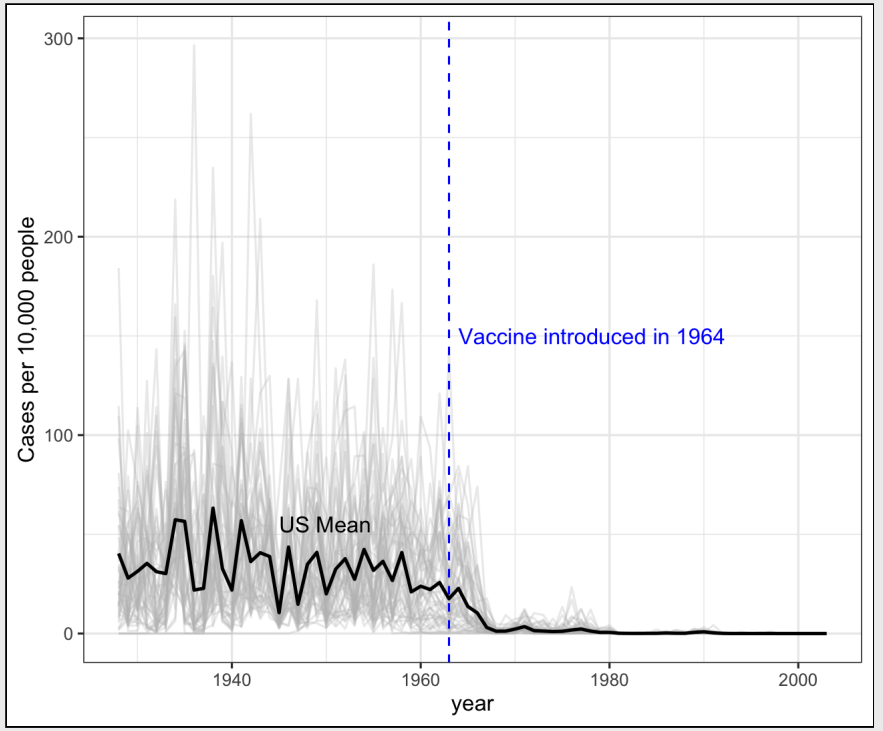
```
ggplot(measles) +
    geom_line(aes(x = year, y = rate, group = state),
              color = 'grey', alpha = 0.3) +
    # Add reference line:
    geom_vline(xintercept = 1963, col = 'blue',
               linetype = 'dashed') +
    annotate('text', x = 1964, y = 150, hjust = 0,
             label = 'Vaccine introduced in 1964',
             color = 'blue') +
    theme_bw() +
    labs(y = 'Cases per 10,000 people')
```



How to: **Average line overlay**

Now overlay the state average line

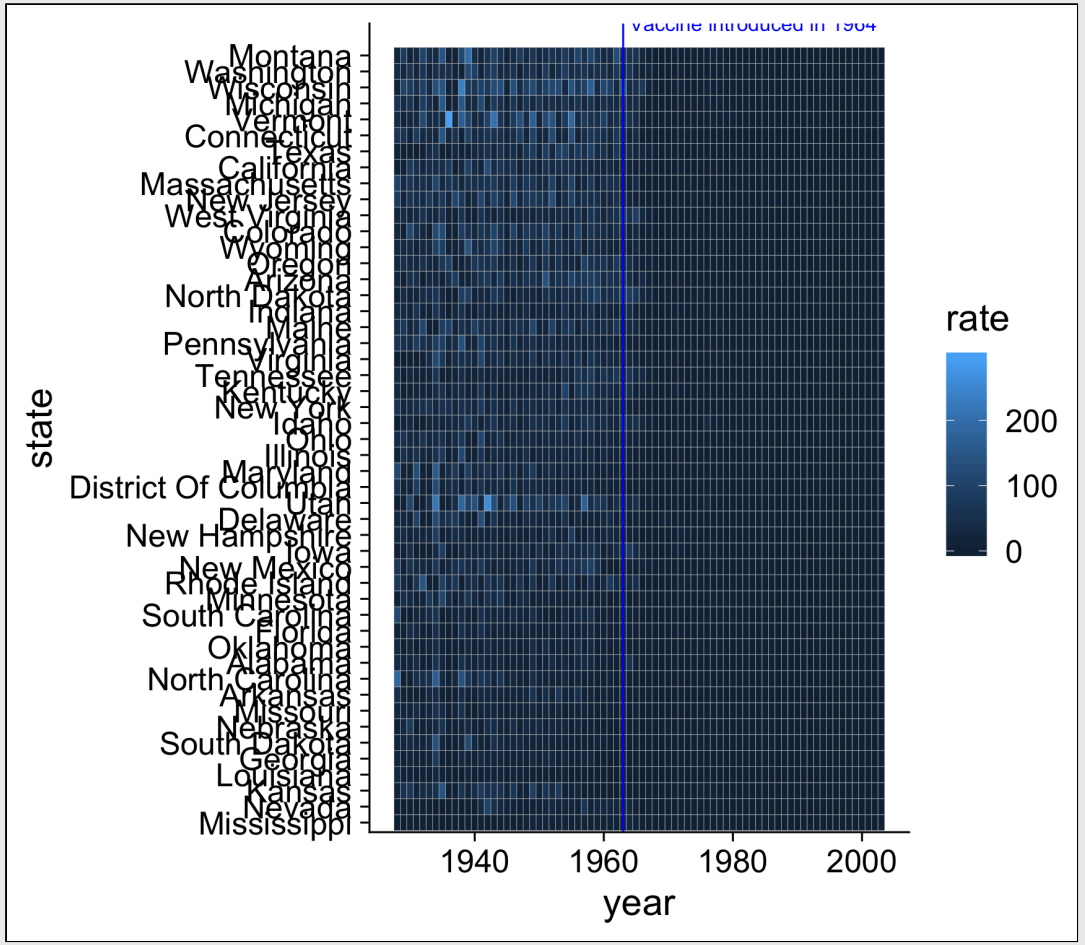
```
measles_us <- measles %>%
    group_by(year) %>%
    summarize(
        rate = sum(count) / sum(population) * 10000)

ggplot(measles) +
    geom_line(aes(x = year, y = rate, group = state),
              color = 'grey', alpha = 0.3) +
    geom_line(data = measles_us,
              aes(x = year, y = rate),
              size = 0.8) +
    # Add reference line:
    geom_vline(xintercept = 1963, col = 'blue',
               linetype = 'dashed') +
    annotate('text', x = 1964, y = 150, hjust = 0,
             label = 'Vaccine introduced in 1964',
             color = 'blue') +
    # Add US average label:
    annotate('text', x = 1945, y = 55, hjust = 0,
             label = 'US Mean') +
    theme_bw() +
    labs(y = 'Cases per 10,000 people')
```



How to: **Heat map**

Create main grid with `geom_tile()`

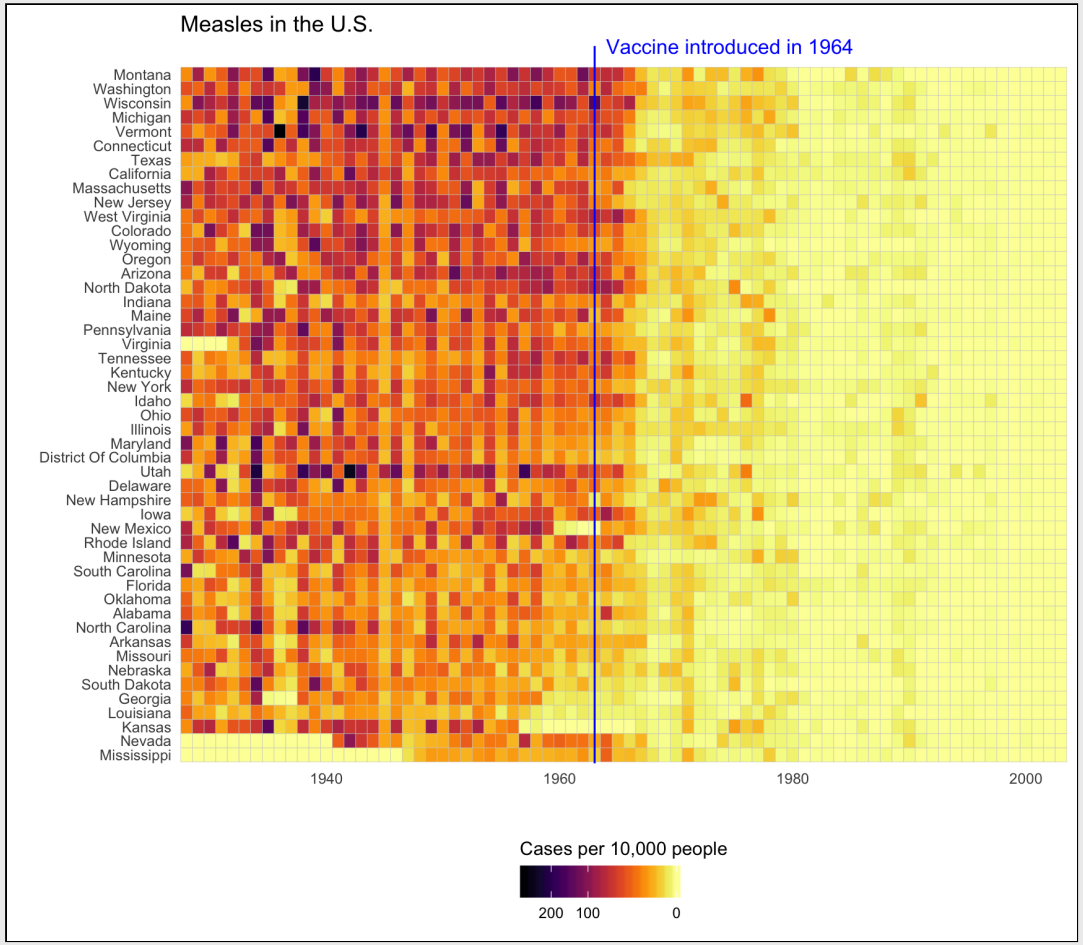
```
ggplot(measles) +
    geom_tile(aes(x = year, y = state, fill = rate),
              color = 'grey80') +
    # Add reference line:
    geom_vline(xintercept = 1963, col = 'blue') +
    annotate('text', x = 1964, y = 51, hjust = 0,
             label = 'Vaccine introduced in 1964',
             color = 'blue')
```



How to: **Heat map**

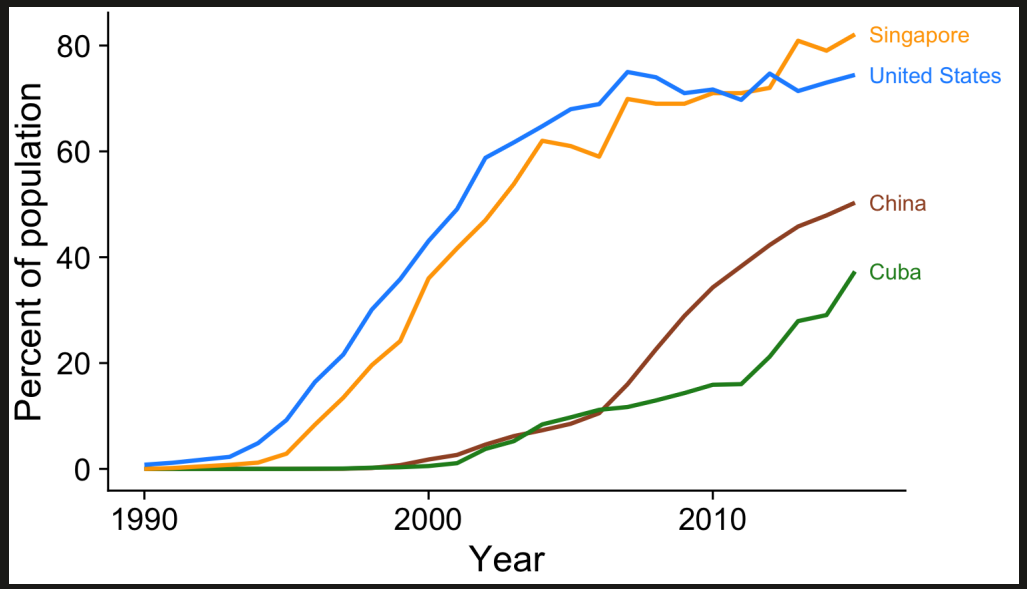
Adjust scales and theme

```
ggplot(measles) +
    geom_tile(aes(x = year, y = state, fill = rate),
              color = 'grey80') +
    # Add reference line:
    geom_vline(xintercept = 1963, col = 'blue') +
    annotate('text', x = 1964, y = 51, hjust = 0,
             label = 'Vaccine introduced in 1964',
             color = 'blue') +
    # Adjust scales
    scale_x_continuous(expand = c(0, 0)) +
    scale_fill_viridis(option = 'inferno', direction = -1,
                       trans = 'sqrt') +
    guides(fill = guide_colorbar(
        title.position = 'top', reverse = TRUE)) +
    coord_cartesian(clip = 'off') +
    # Adjust theme
    theme_minimal() +
    theme(panel.grid = element_blank(),
          legend.position = 'bottom',
          text = element_text(size = 10)) +
    labs(x = '', y = '',
         fill = 'Cases per 10,000 people',
         title = 'Measles')
```

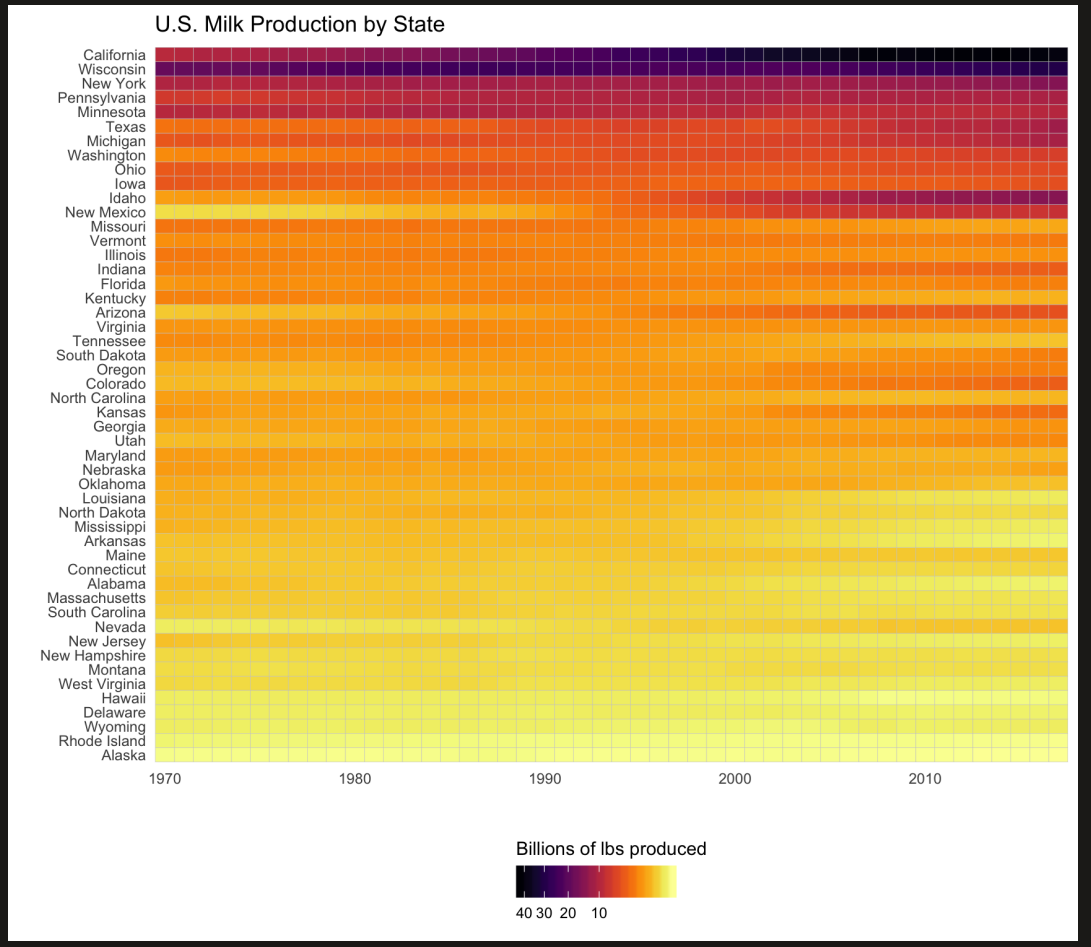


Your turn

Use the `internet_users_country.csv` data to create this chart:

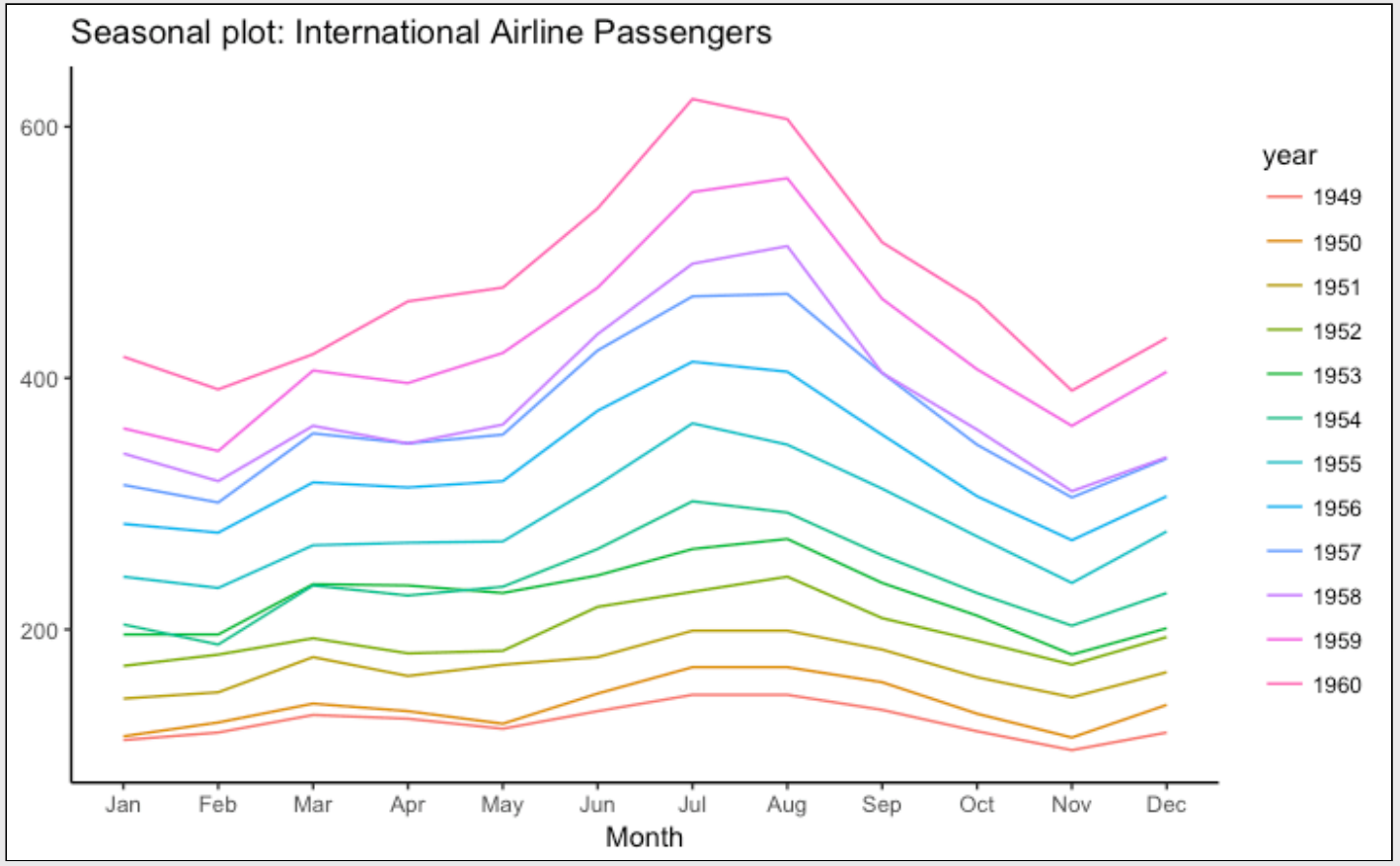


Use the `milk_production.csv` data to create this chart:

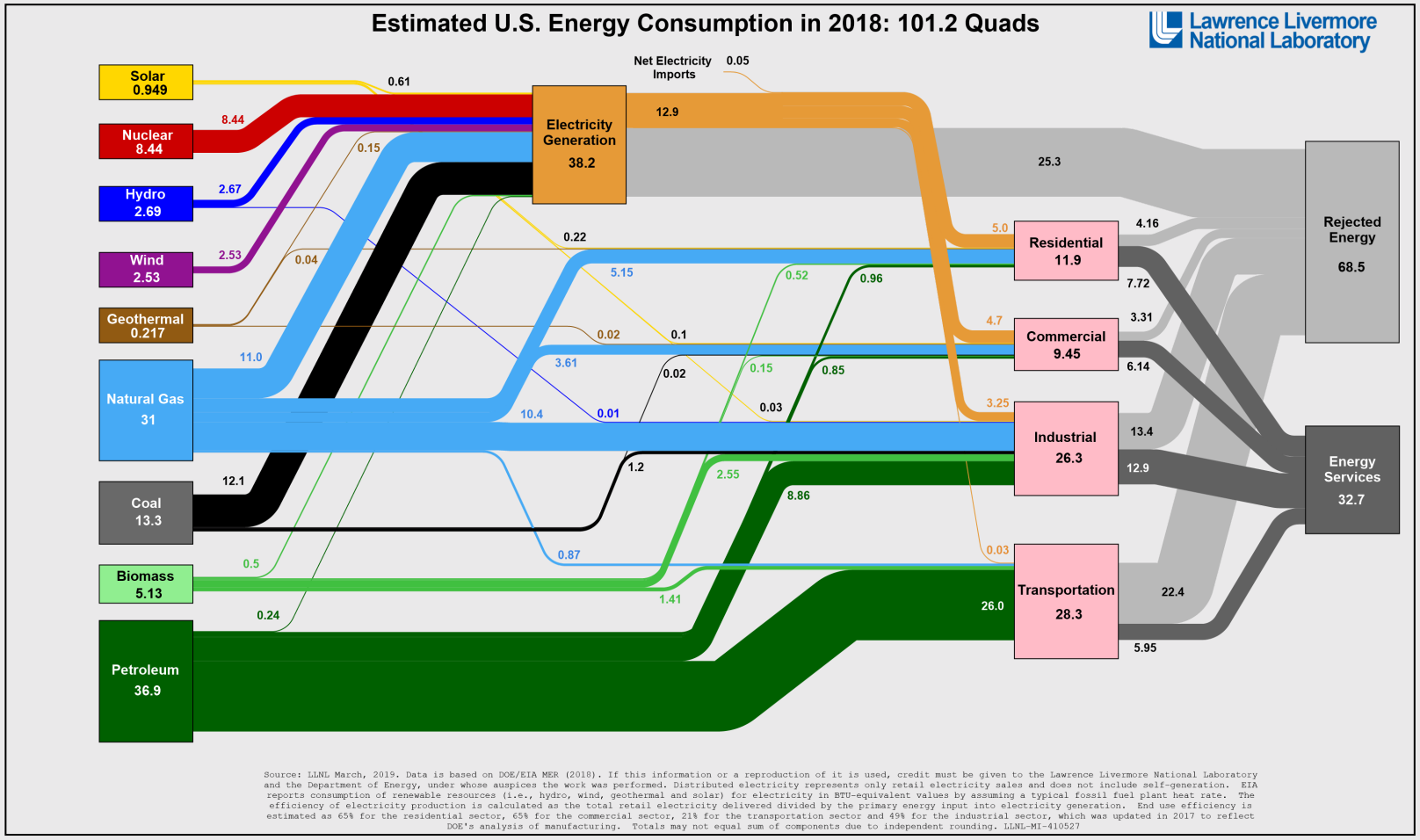


Two other examples for showing change across mutliple categories

Seasonal chart

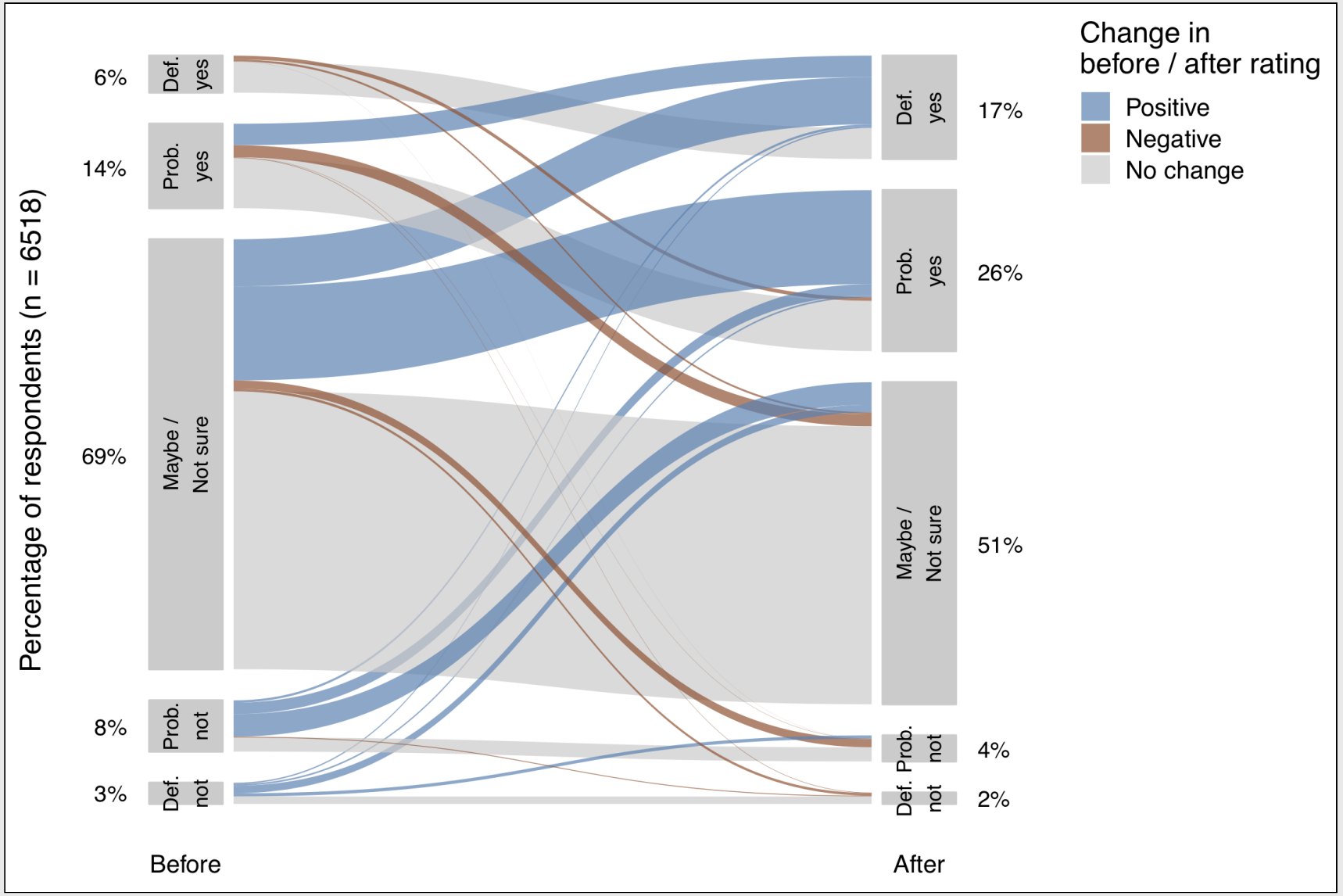


Sankey chart



Source: https://flowcharts.llnl.gov/

Would you consider purchasing an electric car?



5 minute break!

Stand up

Move around

Stretch!

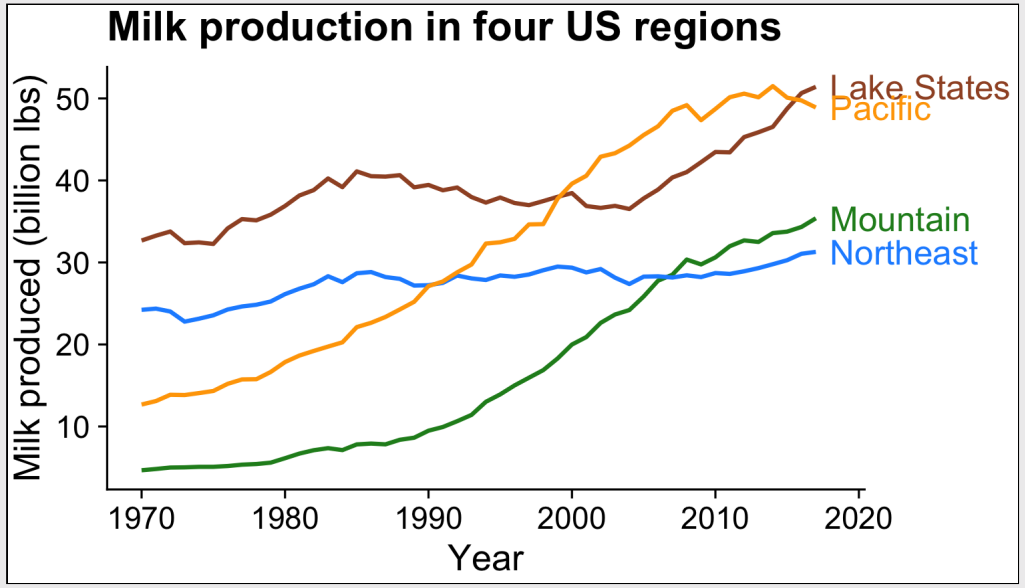
Graphing trends

1. Single variables
2. Multiple variables
3. Animations

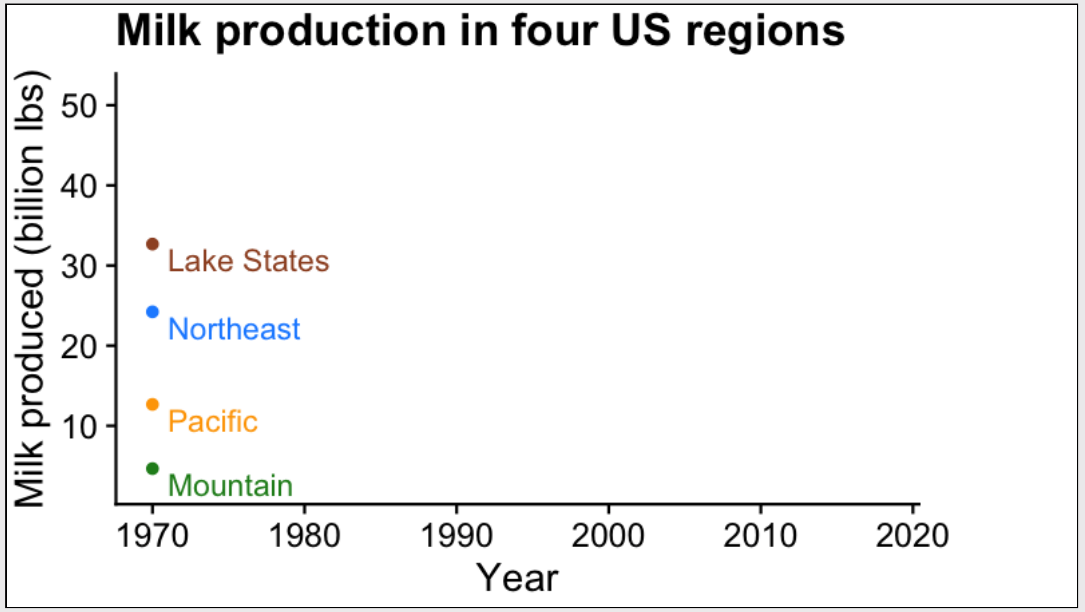
When displaying chart on a webpage,
animation can add emphasis to the change over time

...plus it's fun

Static chart

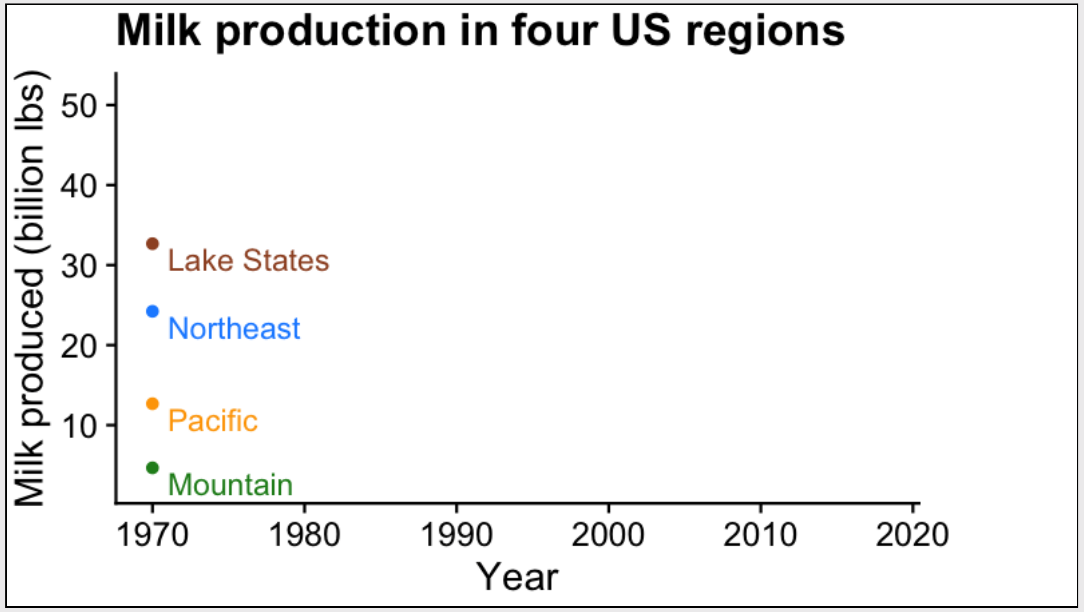


Animated chart

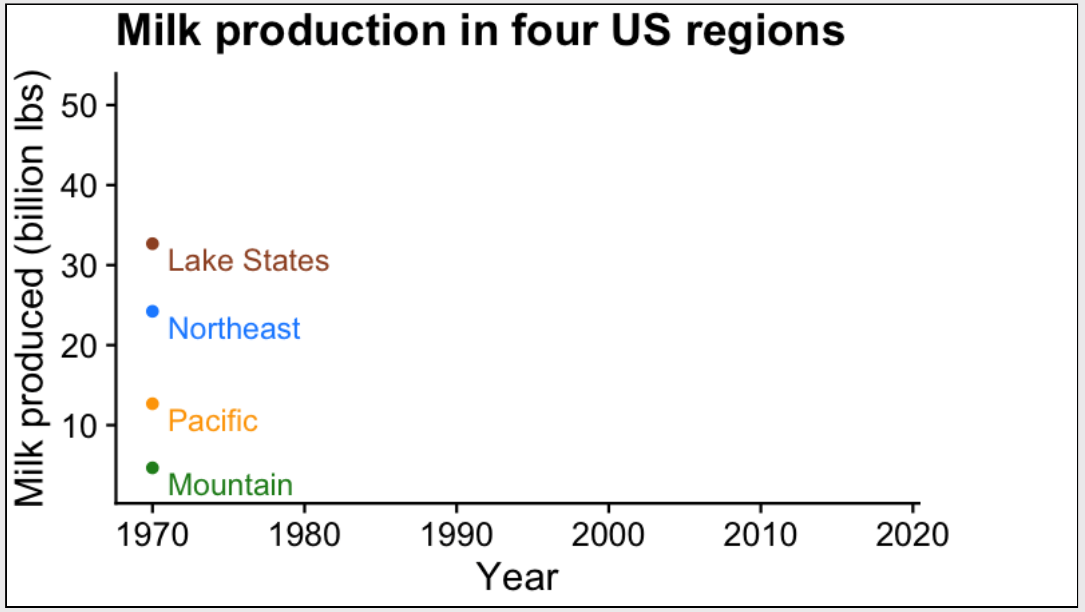


When animating, it's a good idea to **pause on the last frame**

No pause

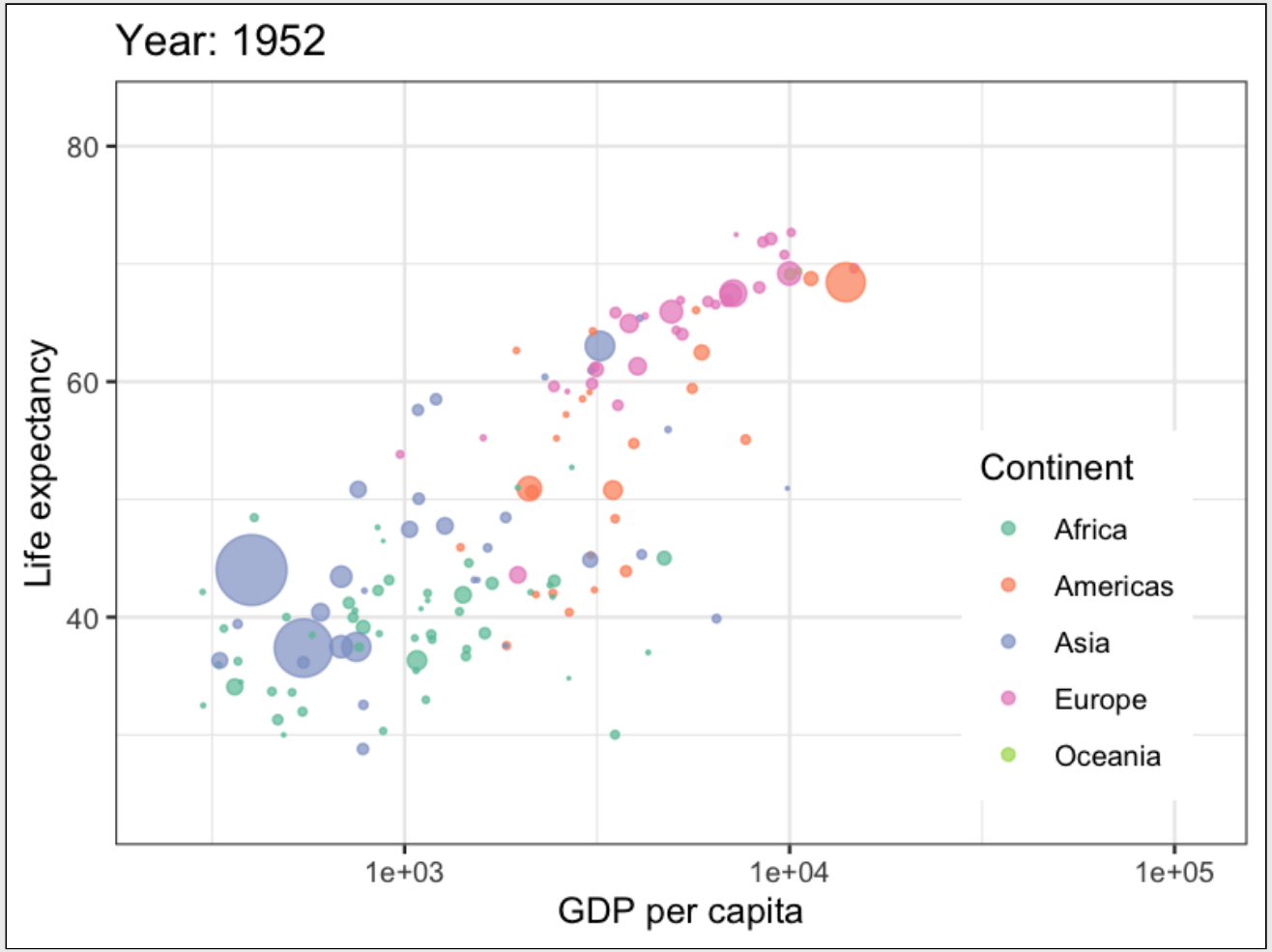


Slight pause



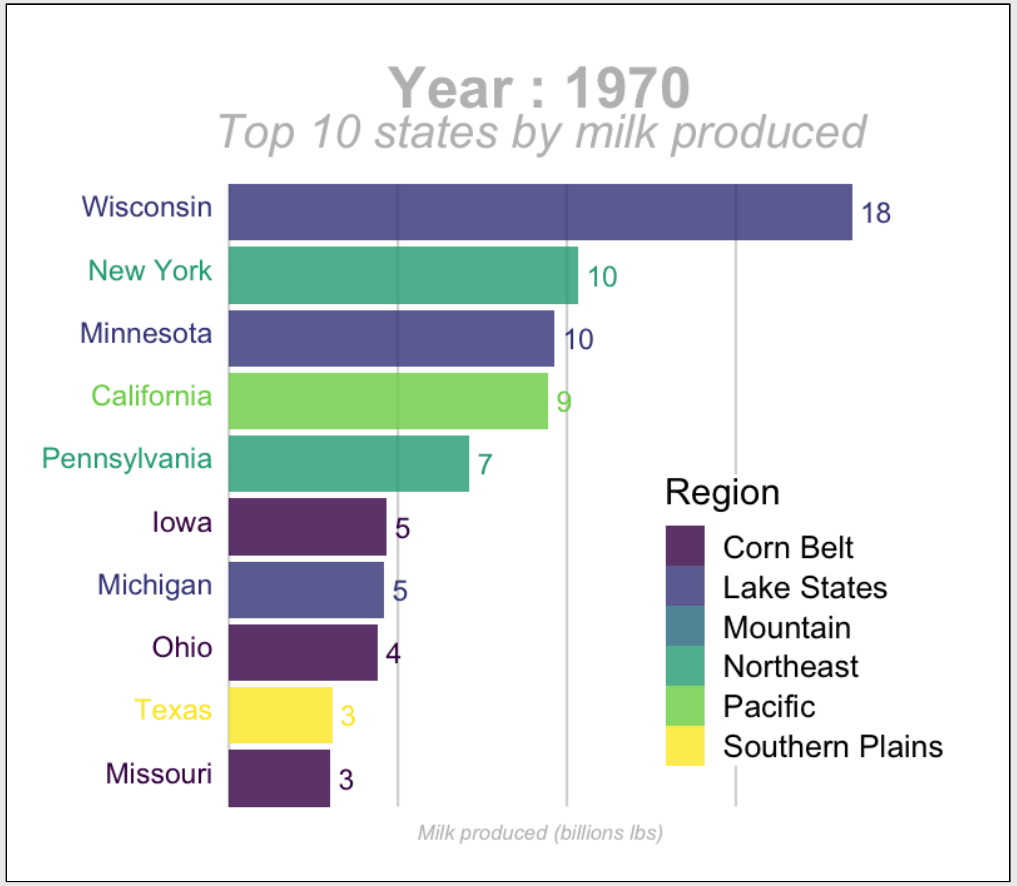
Use animation to show the **time dimension**

"Gapminder" visualization by Hans Rosling



Use animation to show the **time dimension**

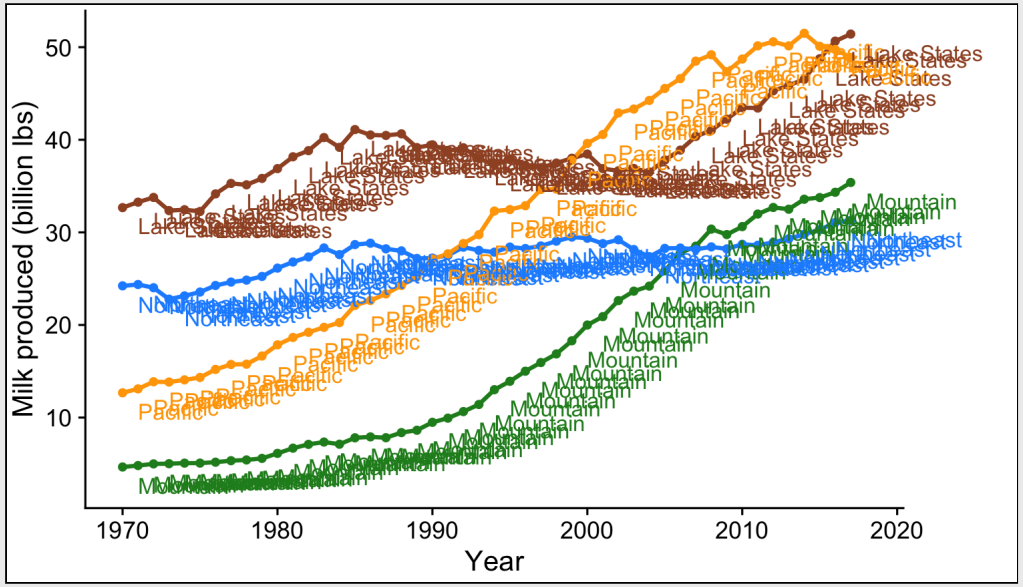
"Bar chart race" of top 10 milk producing states



How to: **Animate a line plot**

First make a static plot with labels for each year

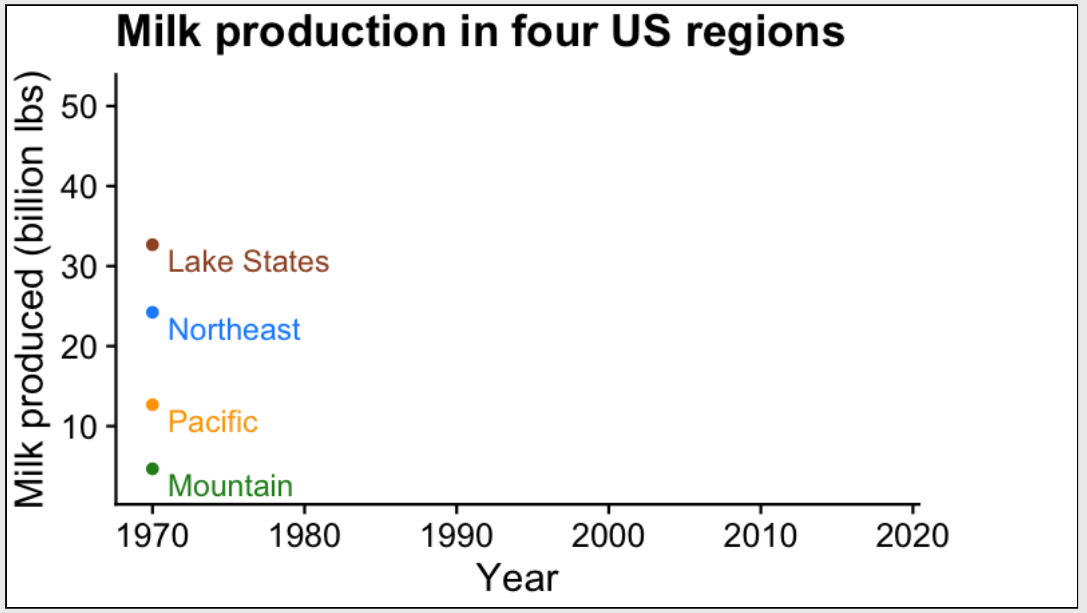
```
milk_region_anim_plot <- ggplot(milk_region,
       aes(x = year, y = milk_produced,
           color = region)) +
    geom_line(size = 1) +
    geom_point() +
    geom_text(aes(label = region),
              hjust = 0,
              nudge_x = 1,
              nudge_y = -2) +
    scale_color_manual(values = c(
        'sienna', 'forestgreen', 'dodgerblue', 'orange')) +
    coord_cartesian(clip = 'off') +
    theme_half_open() +
    theme(legend.position = 'none',
          plot.margin = margin(0.1, 2, 0.1, 0.1, "cm")) +
    labs(x = 'Year',
         y = 'Milk produced (billion lbs)')
```



How to: **Animate a line plot**

Now animate it

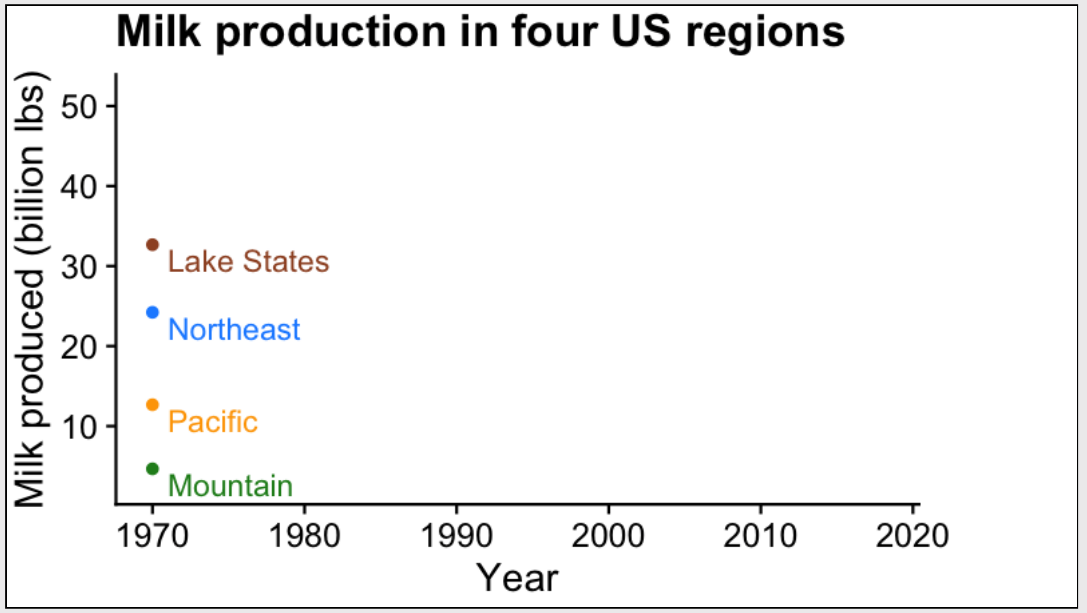
```
milk_region_anim <- milk_region_anim_plot +
    transition_reveal(year)

# Render the animation
animate(milk_region_anim, end_pause = 10,
        width = 800, height = 450, res = 150)
```



How to: **Save an animation**

Now animate it

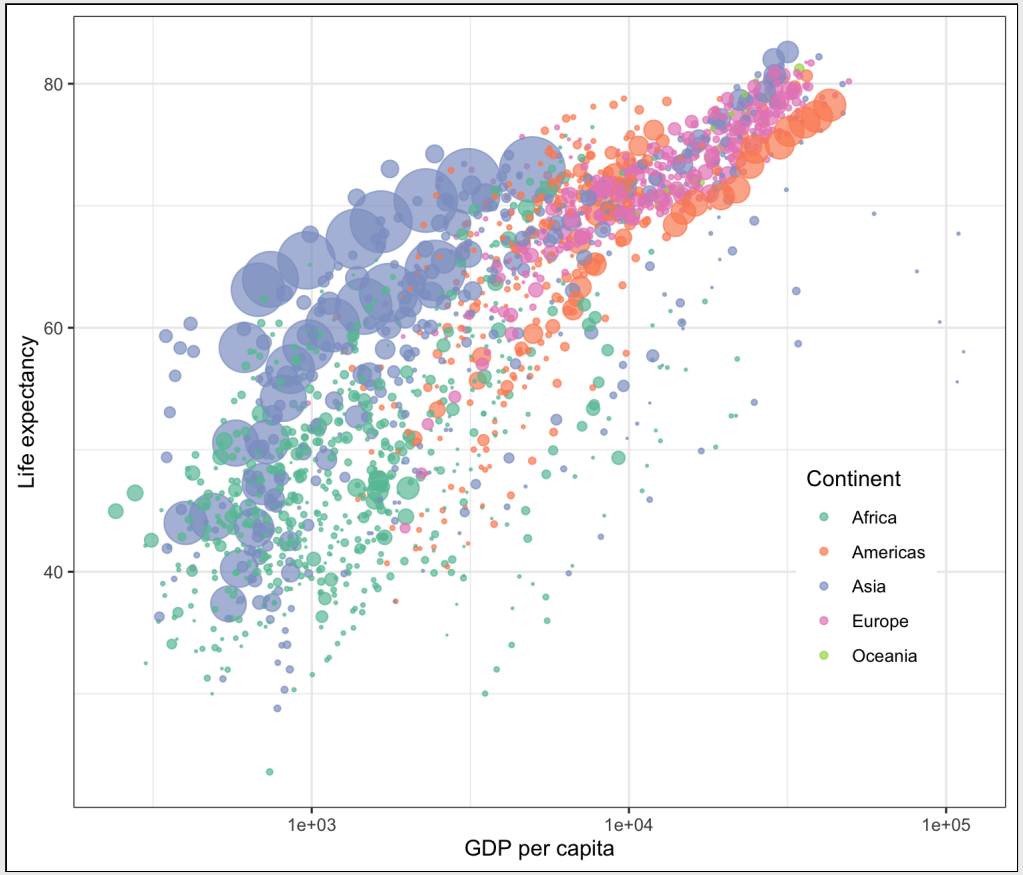
```
milk_region_anim <- milk_region_anim_plot +
    transition_reveal(year)

# Render the animation
animate(milk_region_anim, end_pause = 10,
        width = 800, height = 450, res = 150)

# Save last animation
anim_save(here::here('plots', 'milk_region_animation.gif'))
```



How to: **Change label based on year**

First make a static plot

```
ggplot(gapminder,
    aes(x = gdpPercap, y = lifeExp,
        size = pop, color = continent)) +
    geom_point(alpha = 0.7) +
    scale_size_area(guide = FALSE, max_size = 15) +
    scale_color_brewer(palette = 'Set2') +
    scale_x_log10() +
    theme_bw() +
    theme(legend.position = c(0.85, 0.3)) +
    labs(x = 'GDP per capita',
         y = 'Life expectancy',
         color = 'Continent')
```

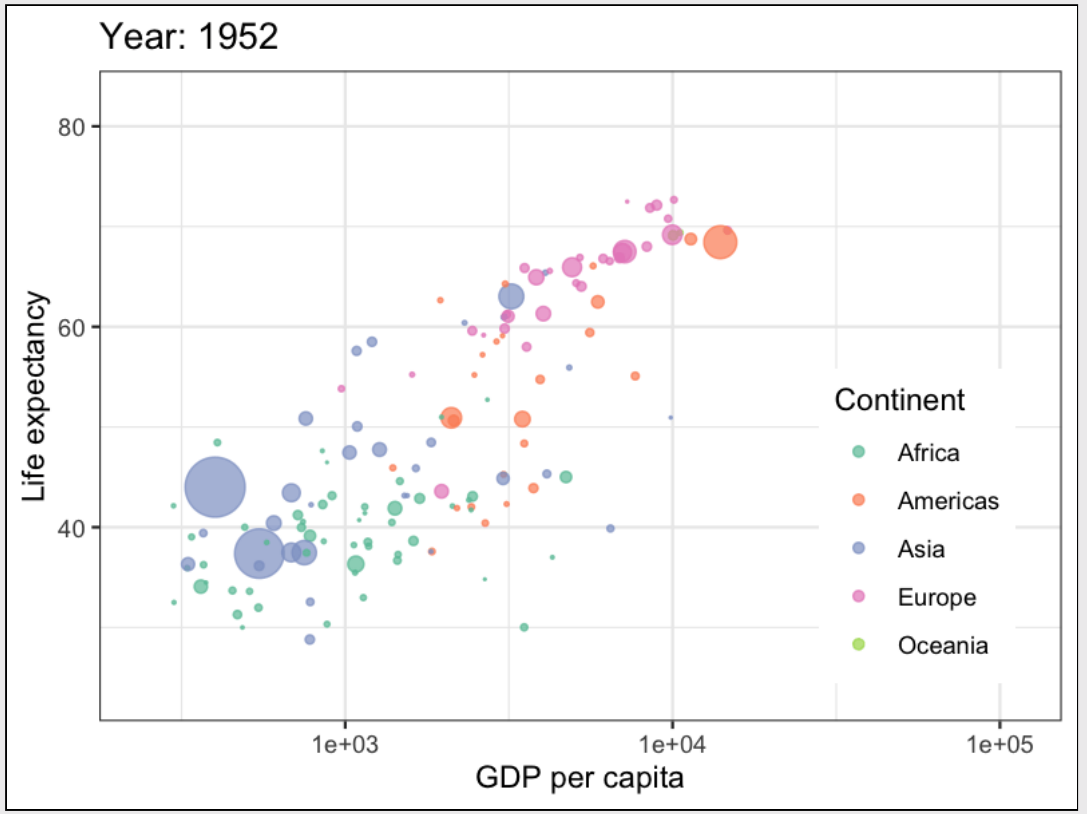


How to: **Change label based on year**

Now animate it

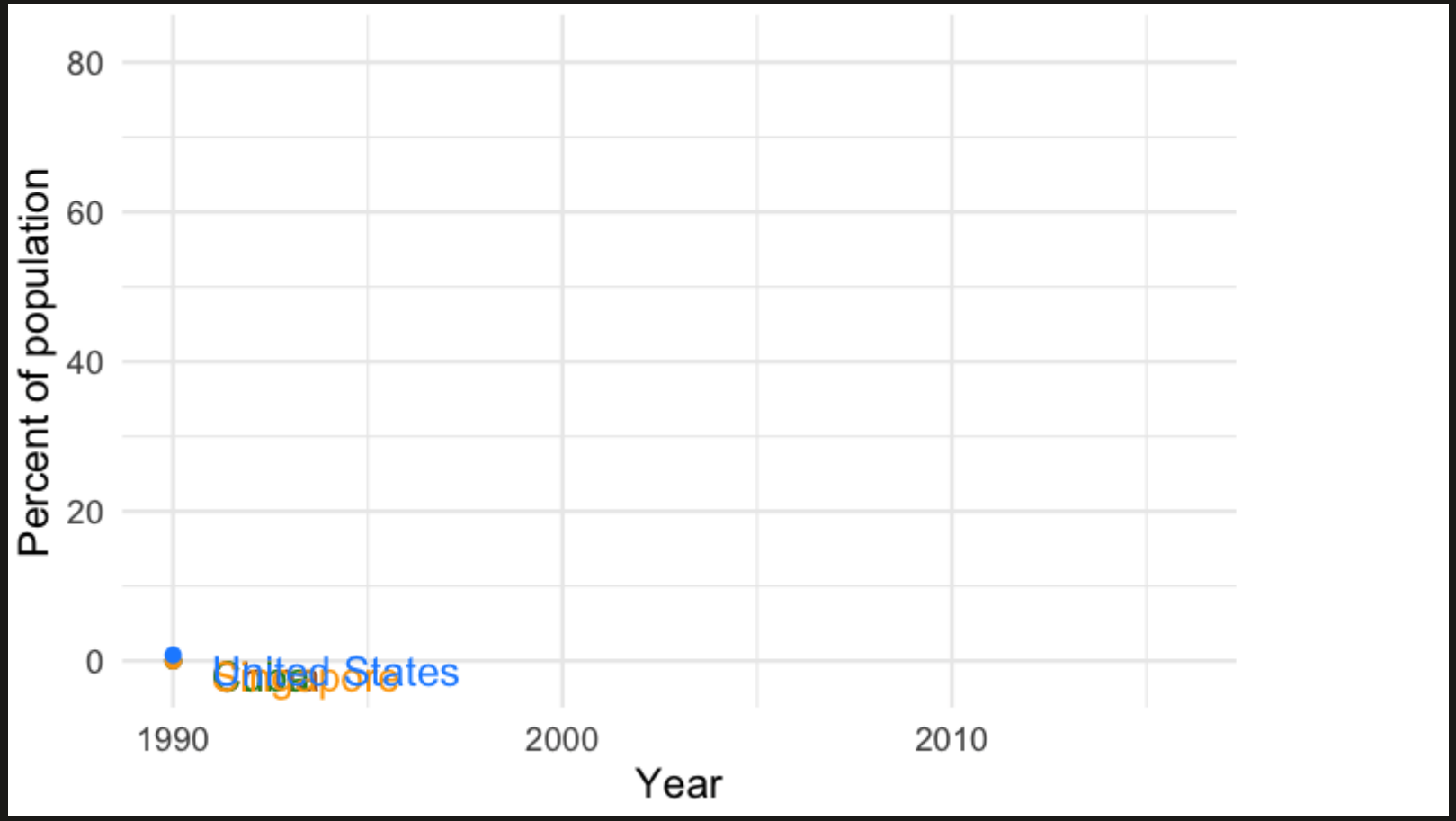
Note: Year must be an integer!

```
gapminder_anim <- gapminder_anim_plot +
    transition_time(year) +
    labs(title = "Year: {frame_time}")

# Render the animation
animate(gapminder_anim, end_pause = 10,
        width = 800, height = 600, res = 150)
```



Your turn

Use the `internet_users_country.csv` data to create the following animation:



One more animation option: **plotly**

Note that it's `plot_ly`, not `plotly`

```
library(plotly)

plot_ly(gapminder,
    x = ~gdpPercap,
    y = ~lifeExp,
    size = ~pop,
    color = ~continent,
    frame = ~year,
    text = ~country,
    hoverinfo = "text",
    type = 'scatter',
    mode = 'markers'
  ) %>%
  layout(
    xaxis = list(
      type = "log"
    )
  )
```