SOC6707: Intermediate Data Analysis

Monica Alexander

Week 3: Exploratory Data Analysis and Data Visualization

Announcement

- We will be online until 28 February.

Where are we at

- Interested in a population, take a sample, to make inferences about a population
- We can summarize variables in our sample using a number of different useful measures (mean, median, mode, SD, variance, range, IQR)
- The process of taking a sample introduces randomness
- We quantify randomness and uncertainty using probability measures

Where are we at

- Probability distributions summarize the probability/likelihood of an event occurring (e.g. a variable is in a certain range)
- If we plot probability distributions, the probability of an event is the area under the curve
- Probability distributions can be either discrete (e.g. coin toss) or continuous (e.g. height)
- the Normal distribution is a special case of a continuous probability distribution
- the Standard Normal distribution has mean 0 and SD $= 1$
- Any variable that is normally distributed can be converted into a standard normal variable (Z-score)

Where are we going

What we will cover today:

- Why do we care so much about the Normal distribution (or, sampling distributions and the Central Limit Theorem)

Then:

- What is Exploratory Data Analysis (EDA) and why do we do it?
- Steps of EDA
- Data visualization principles

Lab: getting data and reading it into R

Sampling distributions and the central limit theorem

Why do we care about the Normal distribution so much

- It's just one distribution
- It tends to over-simplify the real distribution of outcomes/variables

BUT it turns out that the distribution of summary statistics that we're interested in (e.g. means) tend towards being Normal

- this is important for regression and inference

Sampling distributions

A **sampling distribution** is a probability distribution for a statistic based on repeated samples.

Say we are interested in taking a random sample of people's heights, X and calculating the mean height for that sample. So our statistic of interest is the mean height, $\bar{X}$.

Randomly sampling heights

We first take a random sample of 12 people and get the following heights

```
##  [1] 169.6 188.3 163.0 158.9 162.2 178.7 177.4 175.5 175.4 180.3 155.0 171.4
```

The observed mean height of this sample is 171.3.

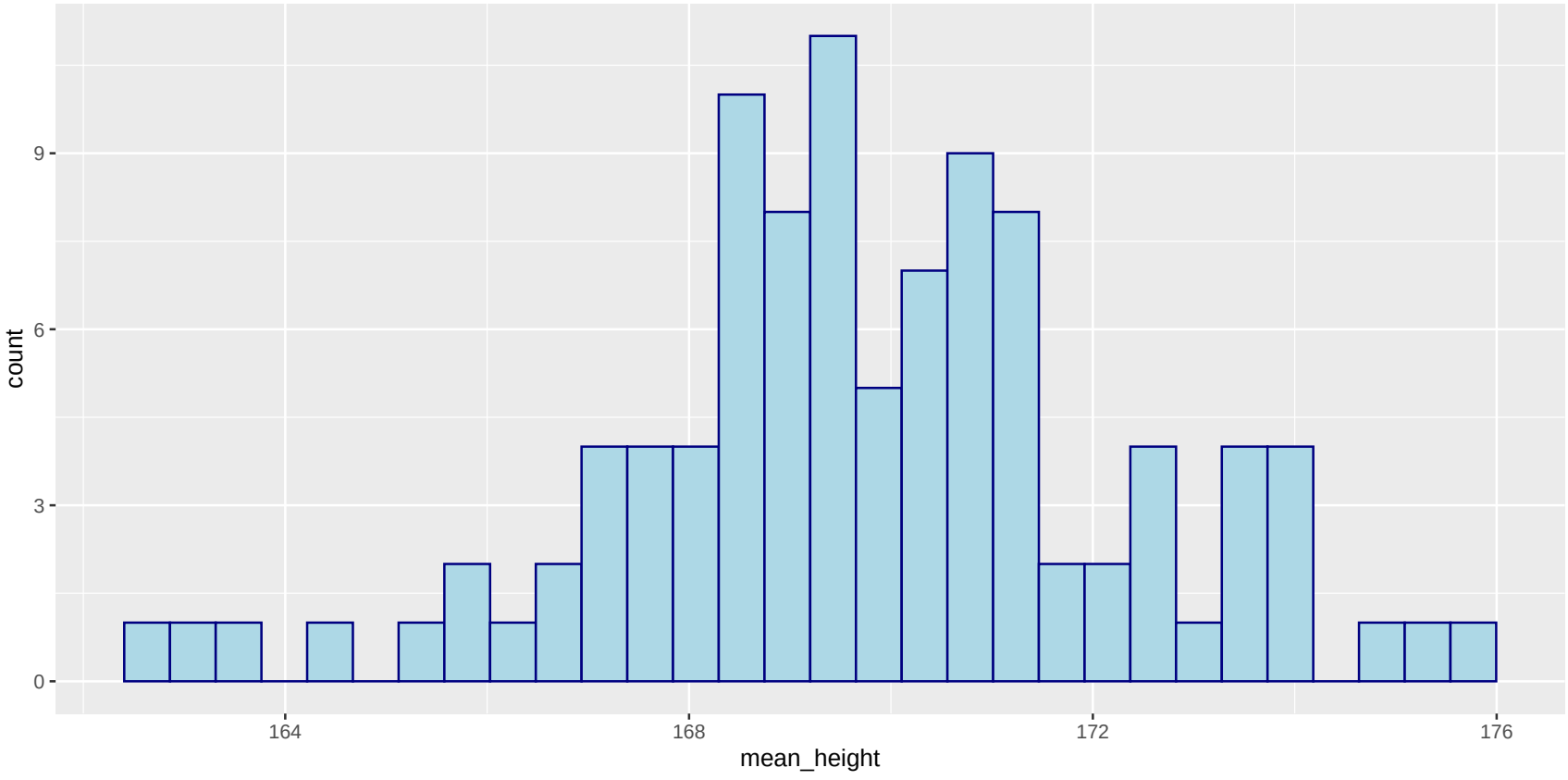
We take a random sample of another 12 people and get the following heights:

```
##  [1] 171.5 180.2 162.6 160.9 159.4 169.3 170.2 176.7 177.8 162.1 173.1 169.3
```

The observed mean height of this sample is 169.4.

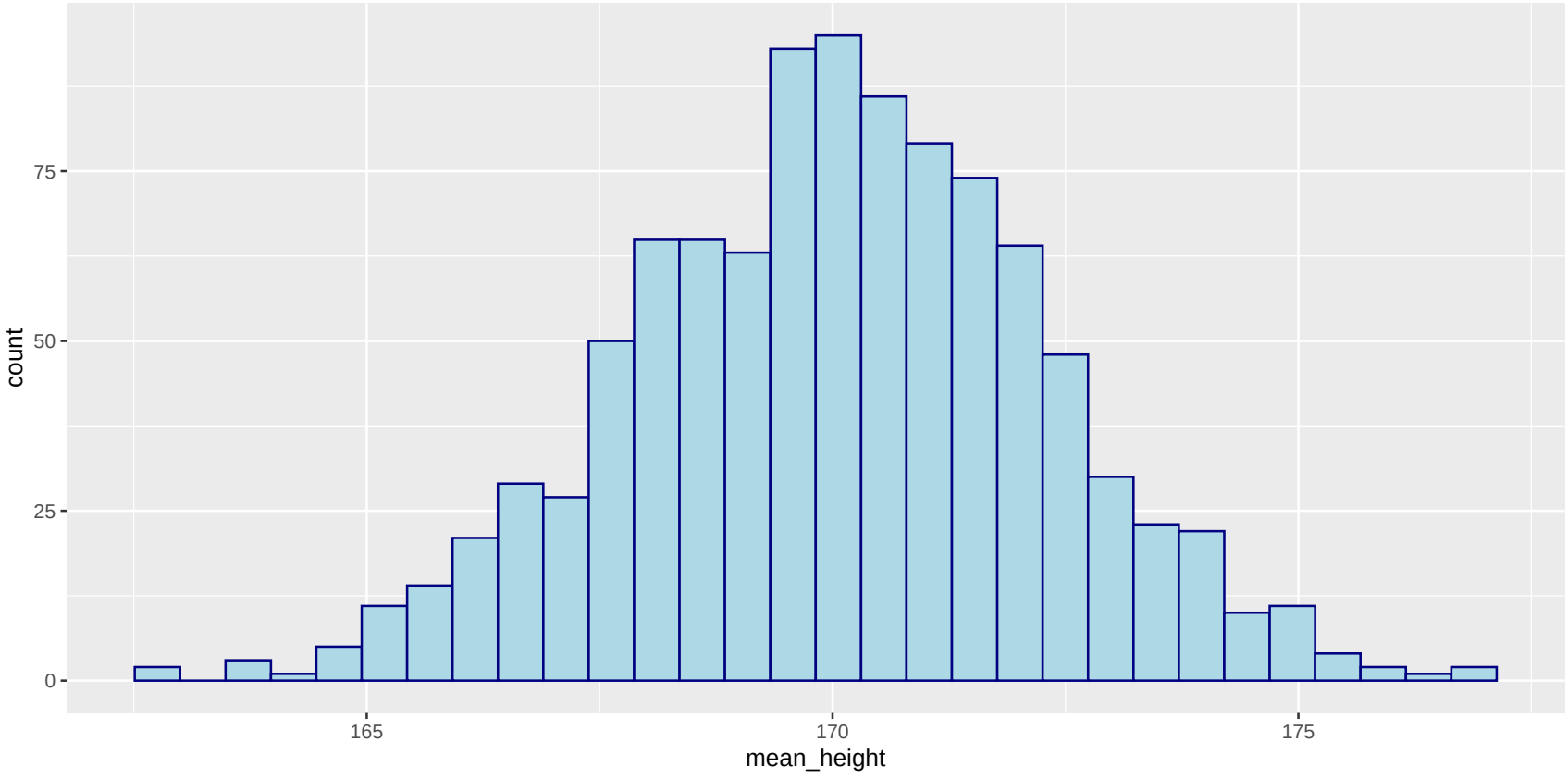
Randomly sampling heights

Say that I keep doing this process again and again and again, and end up with 100 observations of mean height. I can plot a histogram of these means:



Randomly sampling heights

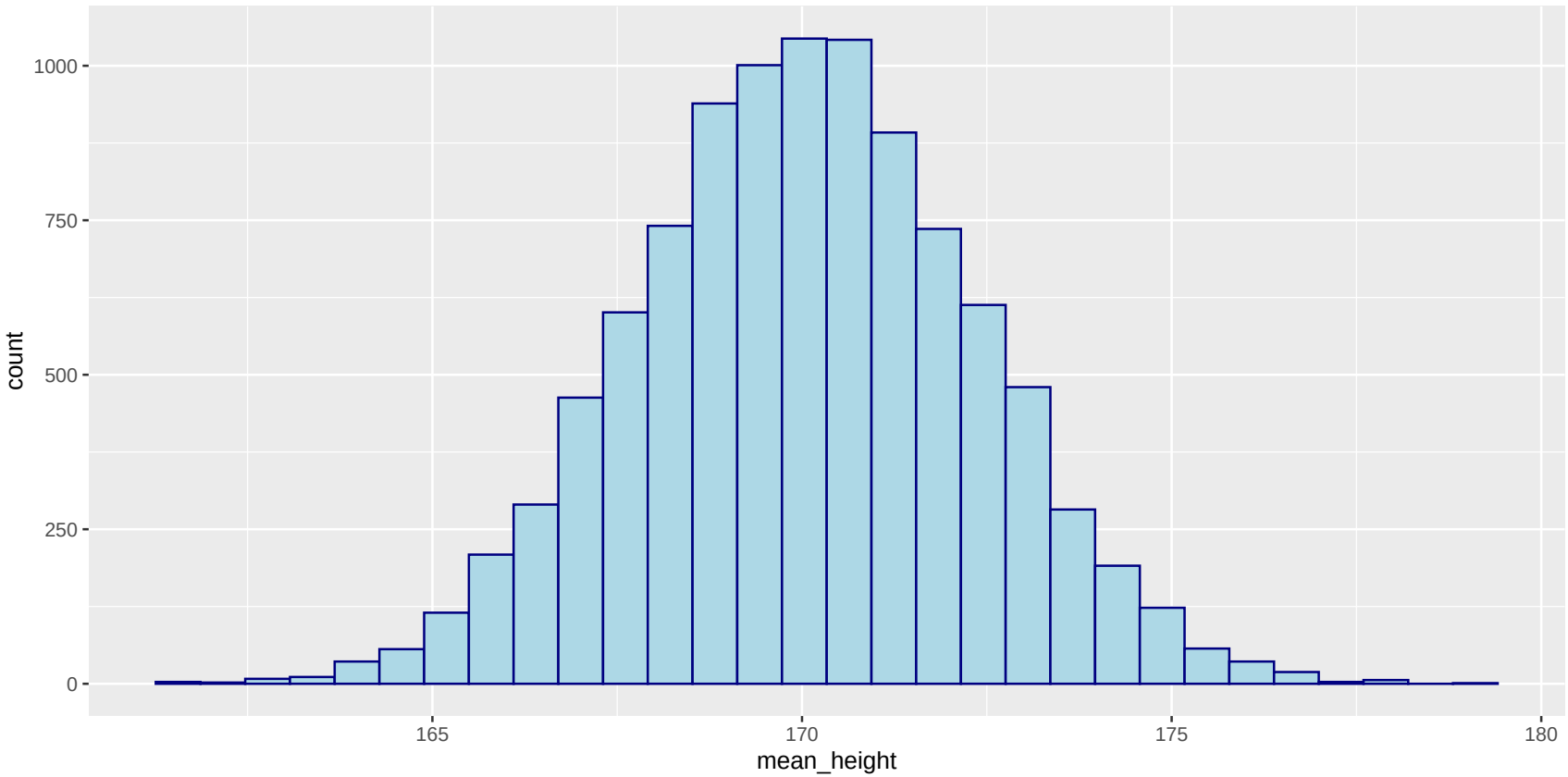
Okay, what if I take 1000 samples. I plot a histogram of the means again



What do you notice?

Randomly sampling heights

What about 10,000 samples?



The central limit theorem

The distribution of the sum (or mean) of a set of independent random variables will **tend towards** a normal distribution.

- "tend towards" means as the number of observations of the sum or mean gets larger, the distribution will become more normal
- The central limit theorem holds even if the original variables themselves are not normally distributed.

The central limit theorem

For a random variable X with $E(X) = \mu$ and $Var(X) = \sigma^2$, the central limit theorem results in the following distribution for the mean $\bar{X}$:

$$\bar{X} \sim N(\mu, \frac{\sigma^2}{n})$$

What does this mean?

- The mean $\bar{X}$ will be centered at the same value as X
- The variance of $\bar{X}$ depends on the variance of the original random variable X and also the number of samples of the mean we have, n.

The quantity $\frac{\sigma}{\sqrt{n}}$ is also called the **standard error of the mean**.

Sampling heights

Before, we were repeatedly taking a random sample of 12 heights.

When we did this 100 times, we had 100 observations of $\bar{X}$. The mean was 170 and the standard error of the mean was 0.081.

When we did this 10000 times, we had 100 observations of $\bar{X}$. The mean was 170 and the standard error of the mean was 8×10^{-4}.

Exploratory Data Analysis (EDA)

What is EDA and why do we do it?

Before we even do any sort of statistical inference, we need to understand the main characteristics of our dataset.

- Helps to identify any potential issues or surprising things about our data
- Helps to check / explore / refine research questions

What is EDA and why do we do it?

EDA is all about asking:

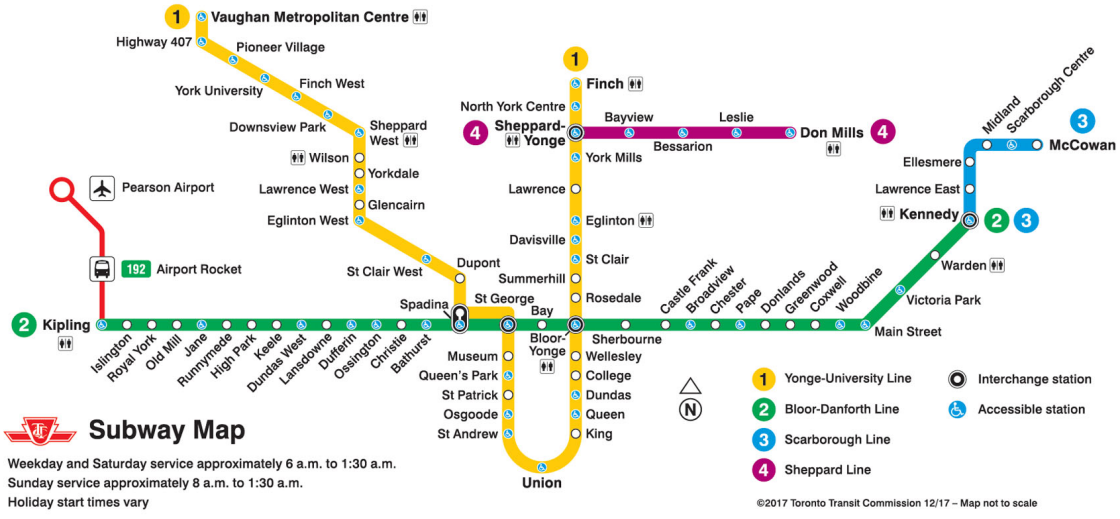
- What types of variables do we have?
- Do we have a complete dataset, or do we have missing data or observations?
- If we have missing data, is it missing equally across observations of different types or concentrated in particular groups?
- Are there any obvious outliers or strange data points?
- What do the data 'look' like?
  - summary measures, measures of centrality, spread
  - Visualizing the data through plots and tables

Steps of EDA

1. Become familiar with size of data set (number of observations and variables available)
2. What kinds of variables are available
3. For the variables that I'm interested in, are there any missing values or other issues?
4. What does the distribution/frequency of observations look like for the variables I'm interested in? (summary measures, tables and graphs)

Example: TTC subway delays in 2019

- Data on TTC subway delay times by station and day available from the Open Data Toronto website: https://open.toronto.ca/



Get familiar with dataset

```
delay_2019
```

```
## # A tibble: 19,222 x 11
##    date                time   day    station code  min_delay min_gap bound line
##    <dttm>              <time> <chr>  <chr>   <chr>     <dbl>   <dbl> <chr> <chr>
##  1 2019-01-01 00:00:00 01:08  Tuesd~ YORK M~ PUSI          0       0 S     YU
##  2 2019-01-01 00:00:00 02:14  Tuesd~ ST AND~ PUMST         0       0 <NA>  YU
##  3 2019-01-01 00:00:00 02:16  Tuesd~ JANE    TUSC          0       0 W     BD
##  4 2019-01-01 00:00:00 02:27  Tuesd~ BLOOR   SUO           0       0 N     YU
##  5 2019-01-01 00:00:00 03:03  Tuesd~ DUPONT  MUATC        11      16 N     YU
##  6 2019-01-01 00:00:00 03:08  Tuesd~ EGLINT~ EUATC        11      16 S     YU
##  7 2019-01-01 00:00:00 03:09  Tuesd~ DUPONT  EUATC         6      11 N     YU
##  8 2019-01-01 00:00:00 03:26  Tuesd~ ST CLA~ EUATC         4       9 N     YU
##  9 2019-01-01 00:00:00 03:37  Tuesd~ KENNED~ TUMVS         0       0 E     BD
## 10 2019-01-01 00:00:00 08:04  Tuesd~ DAVISV~ MUNOA         5      10 S     YU
## # ... with 19,212 more rows, and 2 more variables: vehicle <dbl>,
## #   code_desc <chr>
```

Get familiar with dataset

Dimensions (number of rows x number of columns)

```
dim(delay_2019)
```

```
## [1] 19222    11
```

Variable names

```
colnames(delay_2019)
```

```
##  [1] "date"      "time"      "day"       "station"   "code"      "min_delay"
##  [7] "min_gap"   "bound"     "line"      "vehicle"   "code_desc"
```

The summary function is useful for a quick overview

```
summary(delay_2019)
```

```
##       date                          time             day
##  Min.   :2019-01-01 00:00:00   Length:19222      Length:19222
##  1st Qu.:2019-03-28 00:00:00   Class1:hms        Class :character
##  Median :2019-06-27 00:00:00   Class2:difftime   Mode  :character
##  Mean   :2019-06-27 16:58:00   Mode  :numeric
##  3rd Qu.:2019-09-25 00:00:00
##  Max.   :2019-12-31 00:00:00
##    station              code              min_delay          min_gap
##  Length:19222       Length:19222       Min.   :  0.000   Min.   :  0.000
##  Class :character   Class :character   1st Qu.:  0.000   1st Qu.:  0.000
##  Mode  :character   Mode  :character   Median :  0.000   Median :  0.000
##                                        Mean   :  2.406   Mean   :  3.536
##                                        3rd Qu.:  3.000   3rd Qu.:  6.000
##                                        Max.   :455.000   Max.   :460.000
##     bound               line                vehicle      code_desc
##  Length:19222       Length:19222       Min.   :   0   Length:19222
##  Class :character   Class :character   1st Qu.:   0   Class :character
##  Mode  :character   Mode  :character   Median :5239   Mode  :character
##                                        Mean   :3974
##                                        3rd Qu.:5671
##                                        Max.   :9206
```

Research question?

- What are some good potential research questions with this dataset?

Sanity checks

We need to check variables should be what they say they are. If they aren't, the natural next question is to what to do with issues (recode? remove?)

E.g. check days of week make sense with the `unique` function

```
delay_2019 %>%
  select(day) %>%
  unique()
```

```
## # A tibble: 7 x 1
##   day
##   <chr>
## 1 Tuesday
## 2 Wednesday
## 3 Thursday
## 4 Friday
## 5 Saturday
## 6 Sunday
## 7 Monday
```

Sanity checks

Check lines: oh no. some issues here. Some have obvious recodes, others, not so much.

```
delay_2019 %>%
  select(line) %>%
  unique() %>%
  pull() # turn into a vector for better display
```

```
##  [1] "YU"                  "BD"                  "YU/BD"
##  [4] "SHP"                 "SRT"                 NA
##  [7] "YUS"                 "B/D"                 "BD LINE"
## [10] "999"                 "YU/ BD"              "YU & BD"
## [13] "BD/YU"               "YU\\BD"              "46 MARTIN GROVE"
## [16] "RT"                  "BLOOR-DANFORTH"      "YU / BD"
## [19] "134 PROGRESS"        "YU - BD"             "985 SHEPPARD EAST EXPR"
## [22] "22 COXWELL"          "100 FLEMINGDON PARK" "YU LINE"
```

Data issues

How bad is the mislabeling of lines? look at frequency of cases

NOTE! New very important function: `group_by`

```
delay_2019 %>%
  group_by(line) %>% # group by line label
  tally() %>% # count the number of occurrences
  arrange(-n) # arrange in descending order
```

```
## # A tibble: 24 x 2
##    line        n
##    <chr>   <int>
##  1 YU       9275
##  2 BD       8200
##  3 SRT       699
##  4 SHP       600
##  5 YU/BD     356
##  6 <NA>       50
##  7 YU / BD    16
##  8 YUS         6
##  9 YU/ BD      3
## 10 999         2
## # ... with 14 more rows
```

Missing values

```
delay_2019 %>%
  summarise_all(.funs = funs(sum(is.na(.))))
```

```
## # A tibble: 1 x 11
##    date  time   day station  code min_delay min_gap bound  line vehicle
##   <int> <int> <int>   <int> <int>     <int>   <int> <int> <int>   <int>
## 1     0     0     0       0     0         0       0  4380    50       0
## # ... with 1 more variable: code_desc <int>
```

Summary statistics

Most interested in delay minutes, which is the `min_delay` variable

```
delay_2019 %>%
  summarize(n_obs = n(),
            mean_delay = mean(min_delay),
            median_delay = median(min_delay),
            range_delay = max(min_delay) - min(min_delay),
            iqr_delay = IQR(min_delay))
```

```
## # A tibble: 1 x 5
##   n_obs mean_delay median_delay range_delay iqr_delay
##   <int>      <dbl>        <dbl>       <dbl>     <dbl>
## 1 18697       2.43            0         455         3
```

Summary statistics

Probably more interesting to do these summaries by line (**stratify** by line); easy extension with the `group_by` function

```
delay_2019 %>%
  group_by(line) %>%
  summarize(n_obs = n(),
            mean_delay = mean(min_delay),
            median_delay = median(min_delay),
            range_delay = max(min_delay) - min(min_delay),
            iqr_delay = IQR(min_delay))
```

```
## # A tibble: 4 x 6
##   line  n_obs mean_delay median_delay range_delay iqr_delay
##   <chr> <int>      <dbl>        <dbl>       <dbl>     <dbl>
## 1 BD     8197       2.11            0         180       3
## 2 SHP     598       2.20            0         165       3
## 3 SRT     631       5.79            3         284       5.5
## 4 YU     9271       2.50            0         455       3
```

Summaries

Could also stratify by reason for delay

```
delay_2019 %>%
  group_by(code_desc) %>%
  summarize(n_obs = n(),
            mean_delay = mean(min_delay),
            median_delay = median(min_delay),
            range_delay = max(min_delay) - min(min_delay),
            iqr_delay = IQR(min_delay)) %>%
  arrange(-n_obs)
```

```
## # A tibble: 119 x 6
##    code_desc                n_obs mean_delay median_delay range_delay iqr_delay
##    <chr>                    <int>      <dbl>        <dbl>       <dbl>     <dbl>
##  1 Miscellaneous Speed Cont~  1997    0.186              0          19         0
##  2 Injured or ill Customer ~  1747    0.151              0          54         0
##  3 Operator Overspeeding      1379    0.114              0           8         0
##  4 Passenger Assistance Ala~  1353    0.800              0          12         0
##  5 Disorderly Patron          1147    3.02               3          23         4
##  6 <NA>                        931    4.19               0         284         5
##  7 Injured or ill Customer ~   671    3.92               3          50         5
##  8 Escalator/Elevator Incid~   605    0.00826            0           5         0
##  9 Speed Control Equipment     527    0.436              0          30         0
## 10 ATC Project                 514    3.88               3          28         5
## # ... with 109 more rows
```

Summaries

Arrange by mean delay time

```
delay_2019 %>%
  group_by(code_desc) %>%
  summarize(n_obs = n(),
            mean_delay = mean(min_delay),
            median_delay = median(min_delay),
            range_delay = max(min_delay) - min(min_delay),
            iqr_delay = IQR(min_delay)) %>%
  arrange(-mean_delay)
```

```
## # A tibble: 119 x 6
##    code_desc                n_obs mean_delay median_delay range_delay iqr_delay
##    <chr>                    <int>      <dbl>        <dbl>       <dbl>     <dbl>
##  1 Traction Power Rail Rela~     1      145          145             0       0
##  2 Priority One - Train in ~    24       78.8         80           193      70.2
##  3 Structure Related Problem     4       70.5         27           228      97.5
##  4 Rail Related Problem          8       58.6          3           455       4
##  5 Fire/Smoke Plan A             6       50           11.5        250      17.5
##  6 Bomb Threat                  12       36.7         20           130      32
##  7 Fire/Smoke Plan B - Sour~    84       19.4         11           180      16.2
##  8 Doors Open in Error          11       18.7         16            40       7.5
##  9 Fire/Smoke Plan B - Sour~     2       13.5         13.5          19       9.5
## 10 Suspicious Package           14       13            3.5          67      22
## # ... with 109 more rows
```

EDA: summary so far

- There's no one checklist of things to looks at, depends on your data and research question
- Get familiar with your dataset
- Check for missing values, and that existing values make sense
- Summary statistics depend on your research question of interest
  - stratifying (`group_by`) by important characteristics often useful

Data visualization

Plot your data!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

- We started to compute some summary statistics above, and showed how summaries can be calculated by group and arranged in different ways to get a sense of differences across groups
- However, graphing/plotting your data is usually the best way to visualize patterns, trends, outliers, issues and other surprising points
- The most appropriate types of graph for your data depends on:
  - the type of variable you are interested in (quantitative or qualitative/categorical)
  - your research questions

Plot your data!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

- Before you start to do any statistical analysis, you should always plot your data
- Data visualization is a key part of EDA and essential in understanding the assumptions and outcomes of your eventual statistical analysis

Plot your data!!!!!!!!!!!!!!!!!!!!!!!!!!

Here's a specific example. Imagine we have the following sets of datasets of (x,y) pairs

```
library(tidyverse)
library(datasauRus)
head(datasaurus_dozen)
```

```
## # A tibble: 6 x 3
##   dataset     x     y
##   <chr>   <dbl> <dbl>
## 1 dino     55.4  97.2
## 2 dino     51.5  96.0
## 3 dino     46.2  94.5
## 4 dino     42.8  91.4
## 5 dino     40.8  88.3
## 6 dino     38.7  84.9
```

How many observations?

```
datasaurus_dozen  %>%  count(dataset)
```

```
## # A tibble: 13 x 2
##    dataset        n
##    <chr>      <int>
##  1 away         142
##  2 bullseye     142
##  3 circle       142
##  4 dino         142
##  5 dots         142
##  6 h_lines      142
##  7 high_lines   142
##  8 slant_down   142
##  9 slant_up     142
## 10 star         142
## 11 v_lines      142
## 12 wide_lines   142
## 13 x_shape      142
```

Do some summaries for each dataset

```
datasaurus_dozen %>%
  group_by(dataset) %>%
  summarise(mean_x = mean(x),
            mean_y = mean(y),
            correlation = cor(x,y))
```

```
## # A tibble: 13 x 4
##    dataset    mean_x mean_y correlation
##    <chr>       <dbl>  <dbl>       <dbl>
##  1 away         54.3   47.8     -0.0641
##  2 bullseye     54.3   47.8     -0.0686
##  3 circle       54.3   47.8     -0.0683
##  4 dino         54.3   47.8     -0.0645
##  5 dots         54.3   47.8     -0.0603
##  6 h_lines      54.3   47.8     -0.0617
##  7 high_lines   54.3   47.8     -0.0685
##  8 slant_down   54.3   47.8     -0.0690
##  9 slant_up     54.3   47.8     -0.0686
## 10 star         54.3   47.8     -0.0630
## 11 v_lines      54.3   47.8     -0.0694
## 12 wide_lines   54.3   47.8     -0.0666
## 13 x_shape      54.3   47.8     -0.0656
```

Aside: another summary measure

On the previous slide, x and y were negatively correlated.

Correlation is the statistical measure of the relationship between two variables. **Pearson's correlation coefficient**, r_{xy} summarizes this relationship into one number. For an observation sample of two random variables $x_1, x_2, \ldots, x_n$ and $y_1, y_2, \ldots, y_n$,

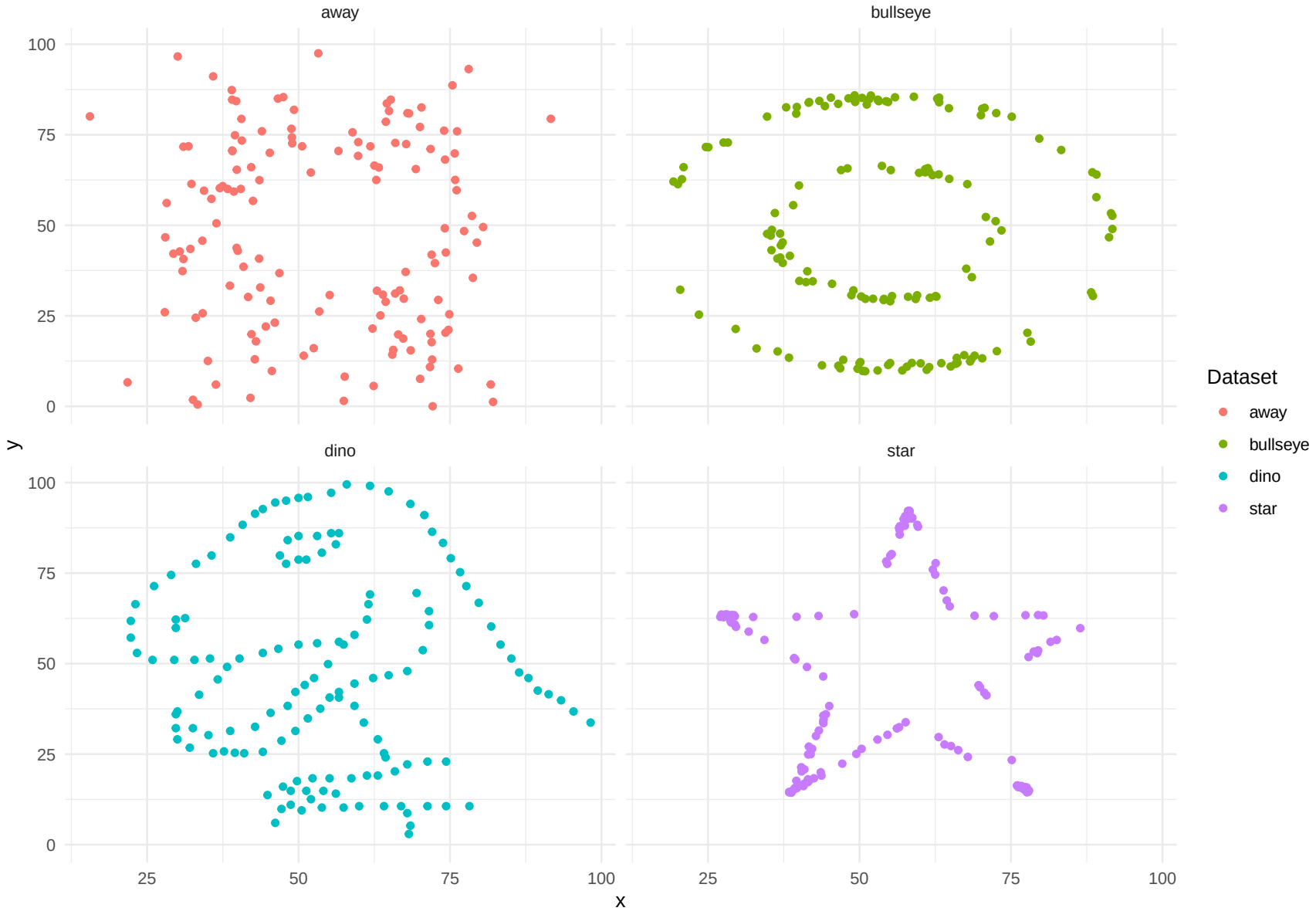
$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Summaries are very similar

```
datasaurus_dozen %>%
  group_by(dataset) %>%
  summarise(mean_x = mean(x),
            mean_y = mean(y),
            correlation = cor(x,y))
```

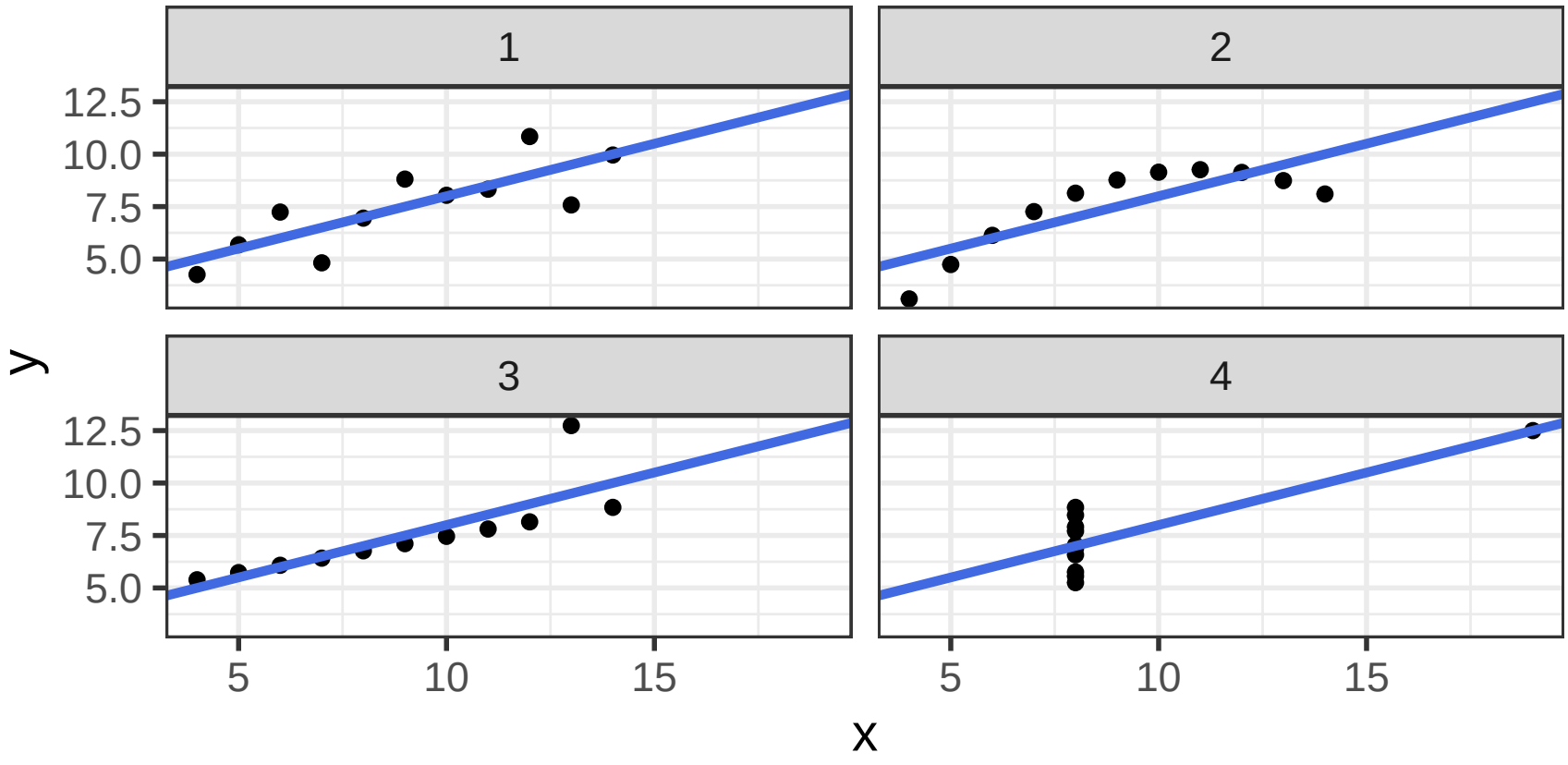
```
## # A tibble: 13 x 4
##    dataset    mean_x mean_y correlation
##    <chr>       <dbl>  <dbl>       <dbl>
##  1 away         54.3   47.8     -0.0641
##  2 bullseye     54.3   47.8     -0.0686
##  3 circle       54.3   47.8     -0.0683
##  4 dino         54.3   47.8     -0.0645
##  5 dots         54.3   47.8     -0.0603
##  6 h_lines      54.3   47.8     -0.0617
##  7 high_lines   54.3   47.8     -0.0685
##  8 slant_down   54.3   47.8     -0.0690
##  9 slant_up     54.3   47.8     -0.0686
## 10 star         54.3   47.8     -0.0630
## 11 v_lines      54.3   47.8     -0.0694
## 12 wide_lines   54.3   47.8     -0.0666
## 13 x_shape      54.3   47.8     -0.0656
```

But now let's plot



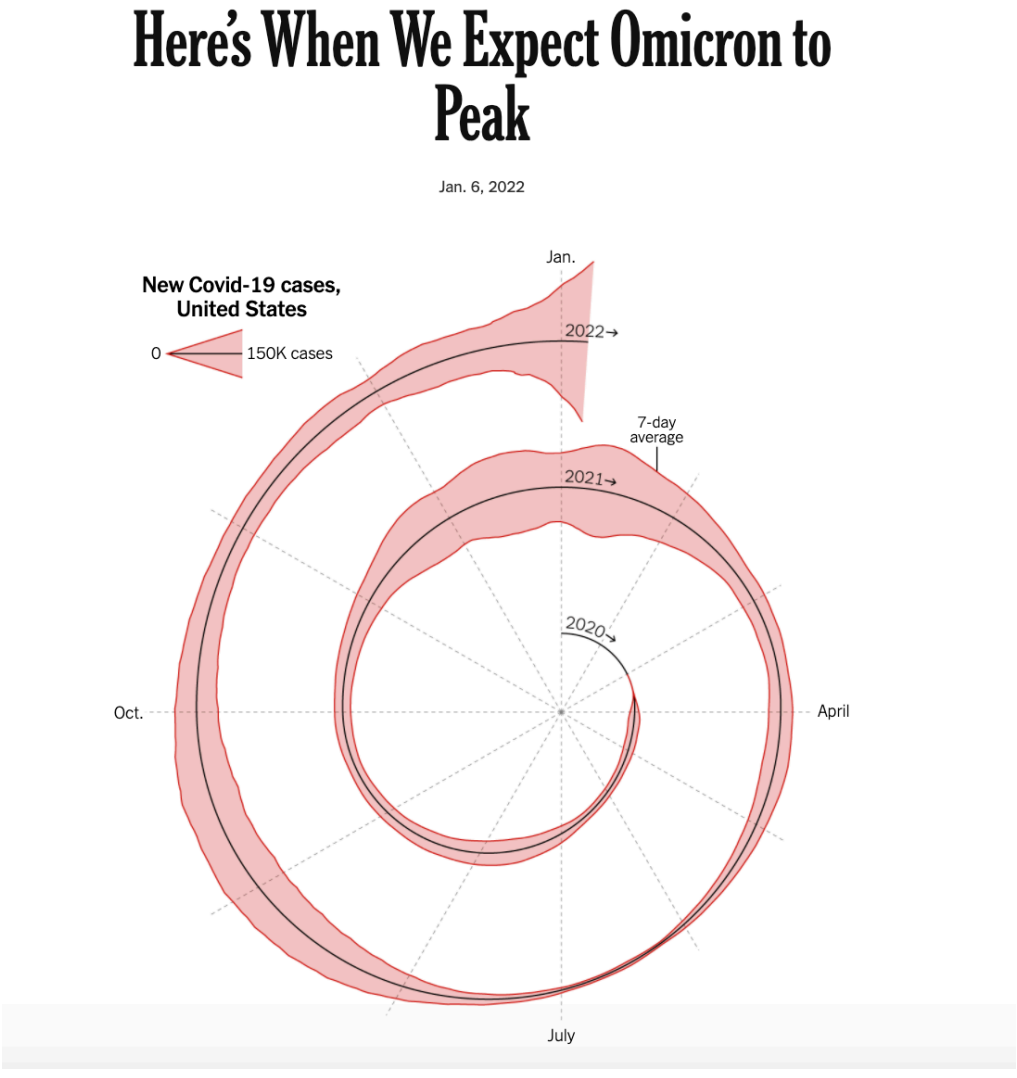
Anscombe's quartet

This is a modern version of a famous plot 'Anscombe's Quartet.' That plot conveys the same message about the importance of plotting the actual data and not relying on summary statistics.

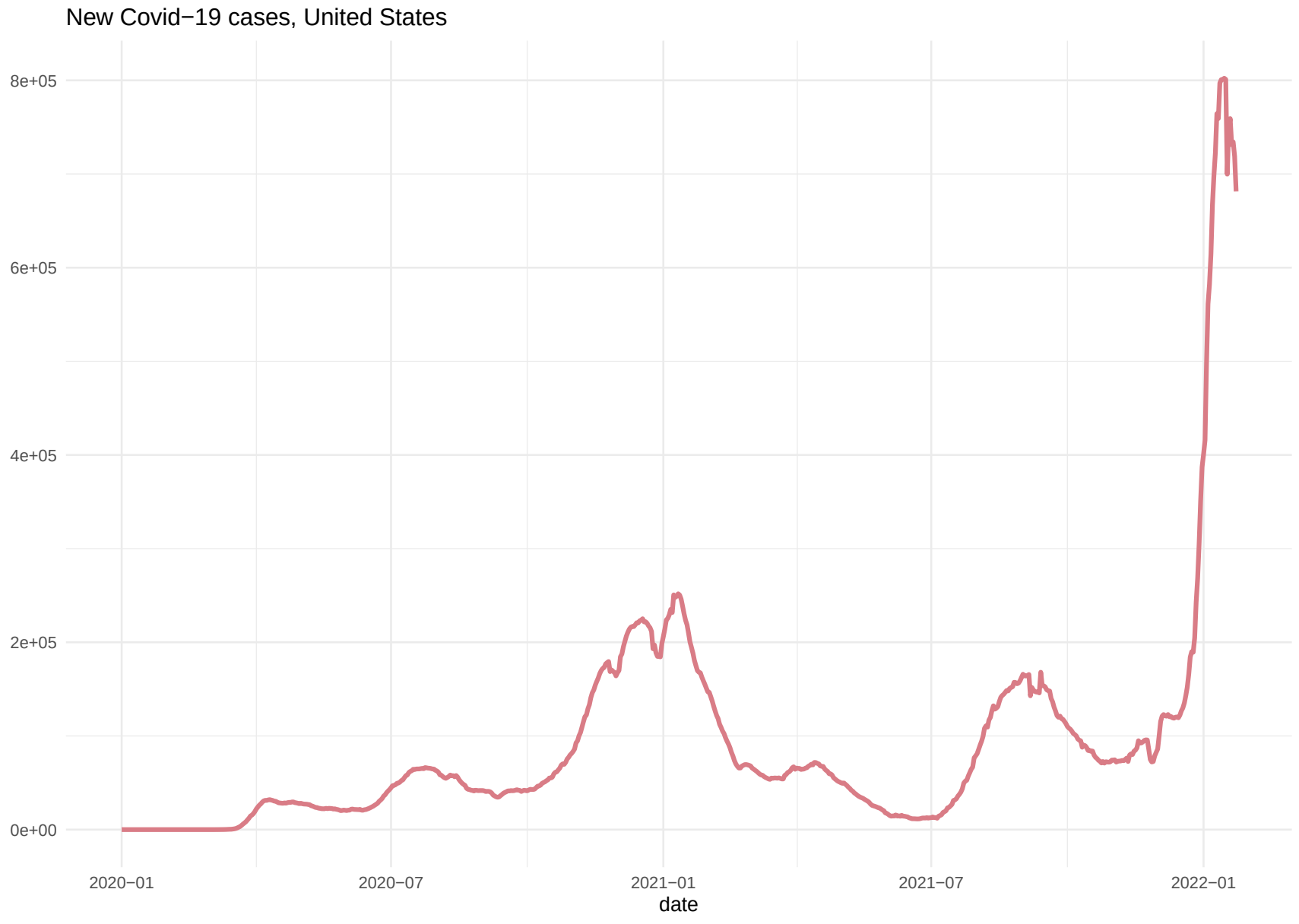


What makes a good plot?

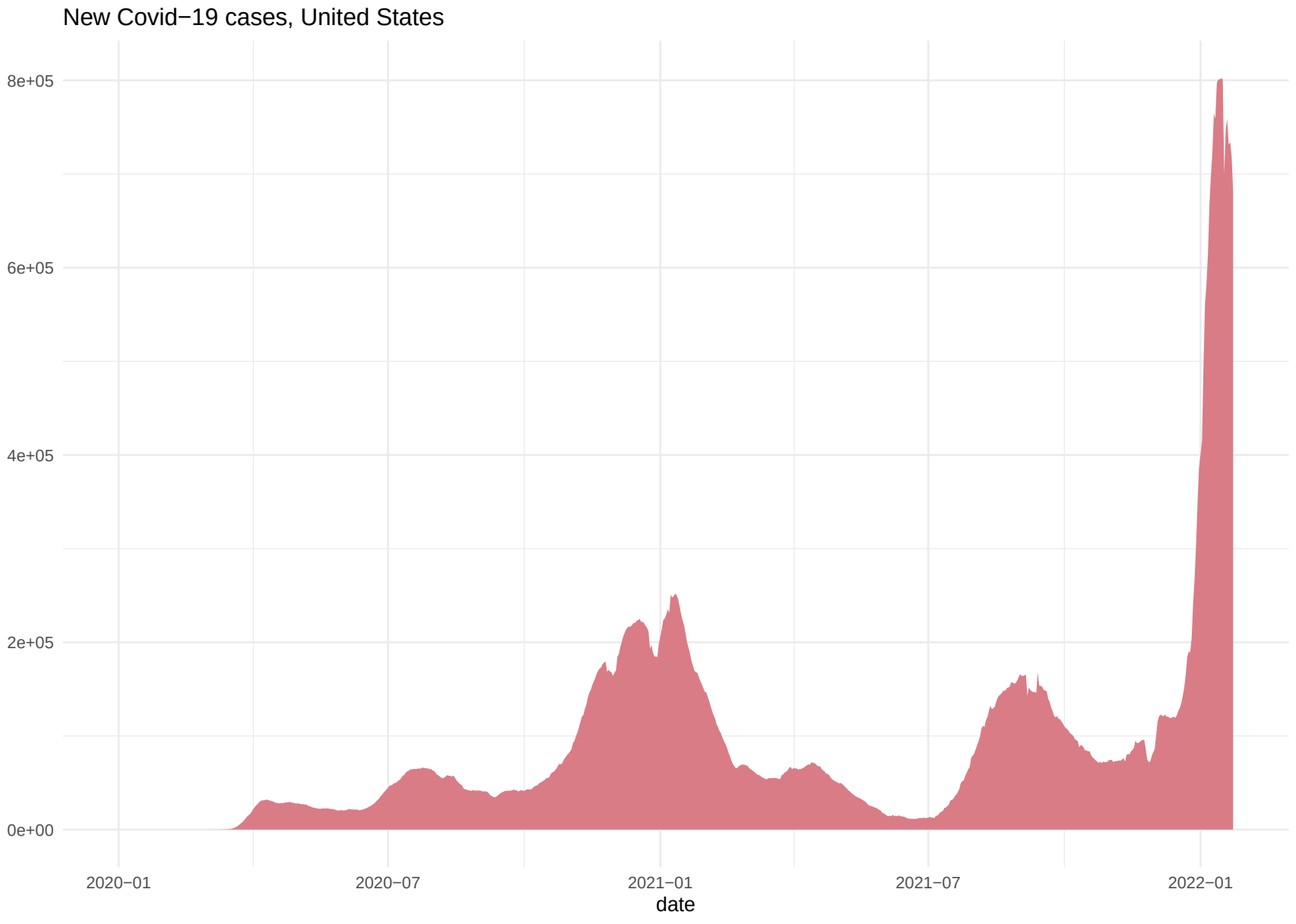
The spiral of doom



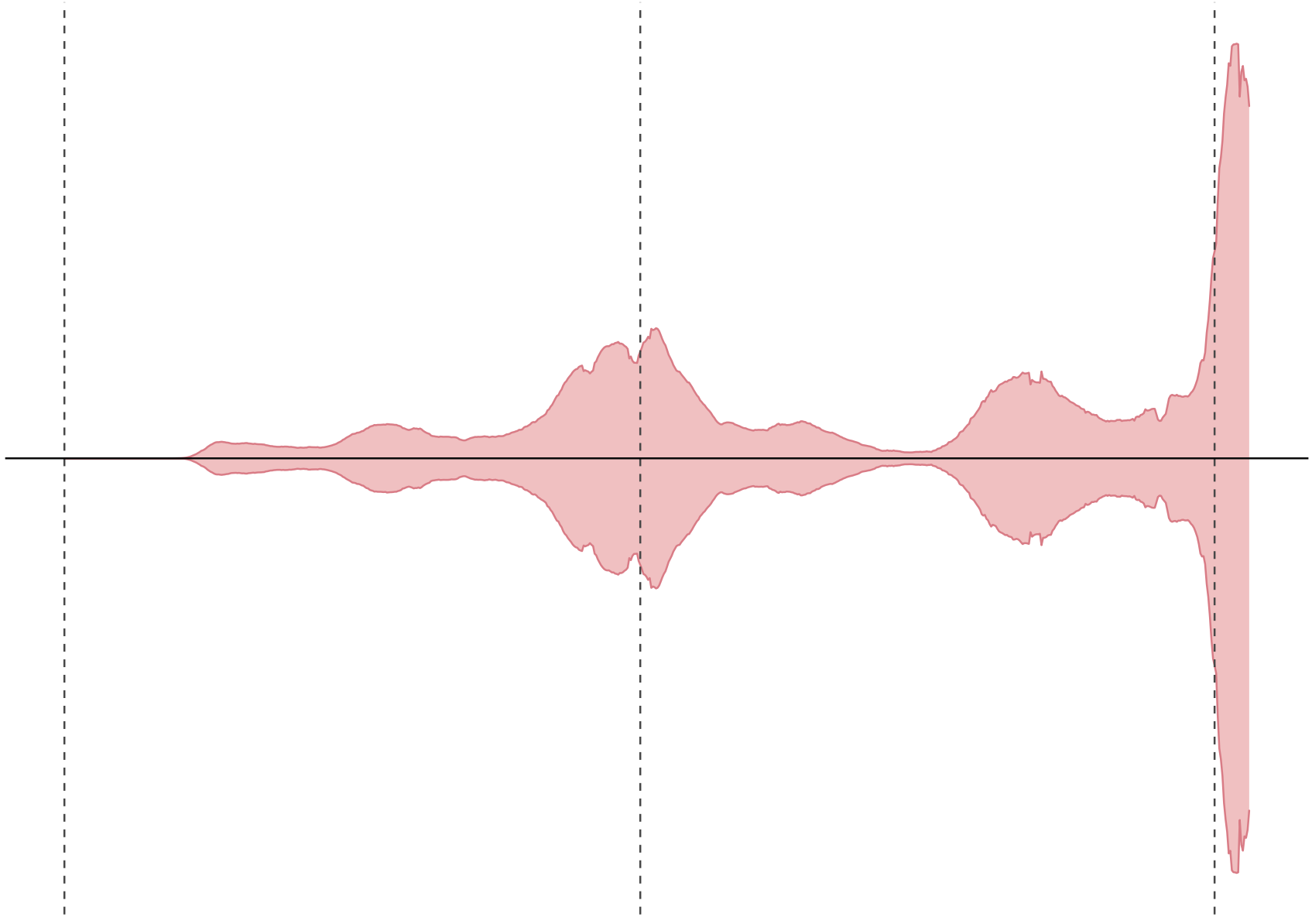
Line plot


New Covid−19 cases, United States
8e+05
6e+05
4e+05
2e+05
0e+00
2020−01
2020−07
2021−01
2021−07
2022−01
date

Area plot


New Covid−19 cases, United States
8e+05
6e+05
4e+05
2e+05
0e+00
2020−01
2020−07
2021−01
2021−07
2022−01
date

Ribbons



What makes a good plot?

- Is the spiral the right graph to use?
- What does right mean?
- Does it effectively portray the information?
- Is it misleading?
- Is it easy to read?
- Is it memorable?

Data visualization principles

- Choose the right graph
- Know your audience
- Emphasize important patterns without being misleading
- Clear, effective designs

Choose the right graph

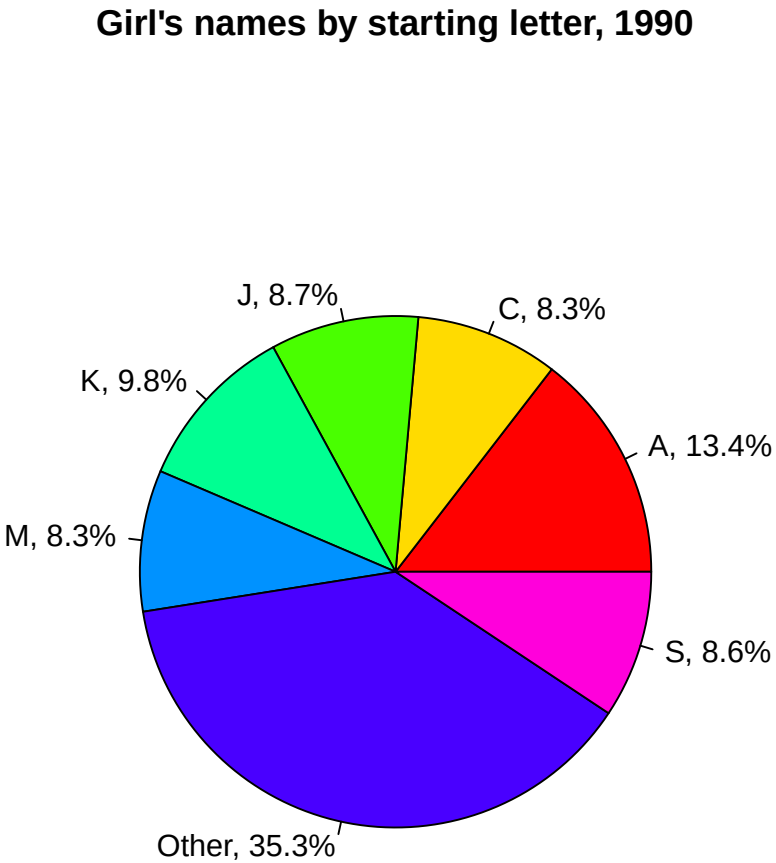
Choosing the right graph primarily depends on the type of variables that you are trying to visualize:

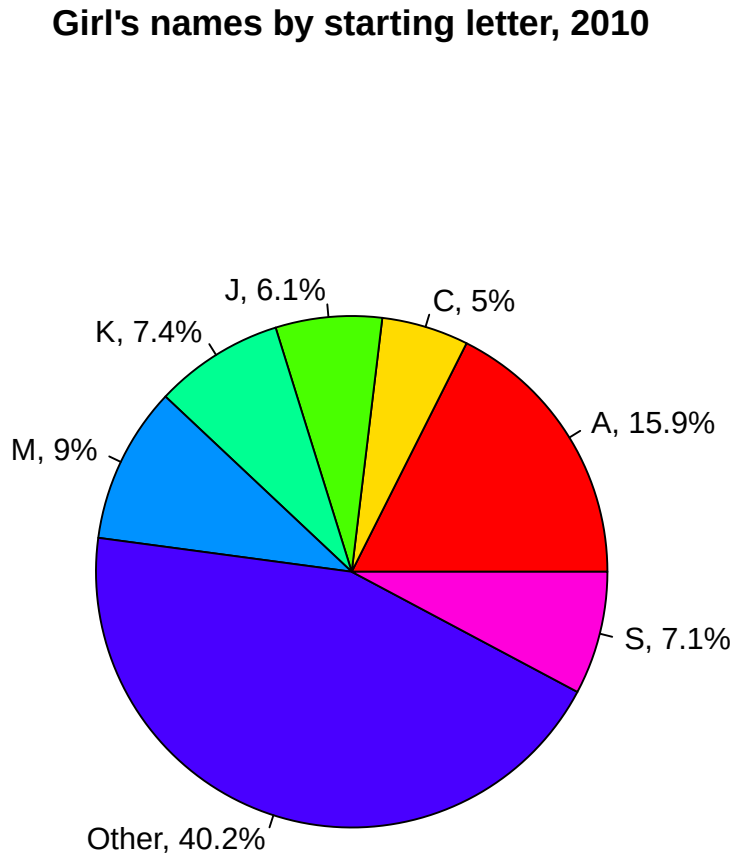
- Quantitative variables e.g. histograms, scatter plots
- Qualitative variables e.g. barcharts

Choose the graph based on the kind of data and the message to be conveyed.

- Do not use different graphs just for variety, as specific graphs convey certain types of information more effectively than others.
- If not required, do not use any chart — show only numbers.

Pie charts


Girl's names by starting letter, 1990
J, 8.7%
C, 8.3%
K, 9.8%
A, 13.4%
M, 8.3%
S, 8.6%
Other, 35.3%

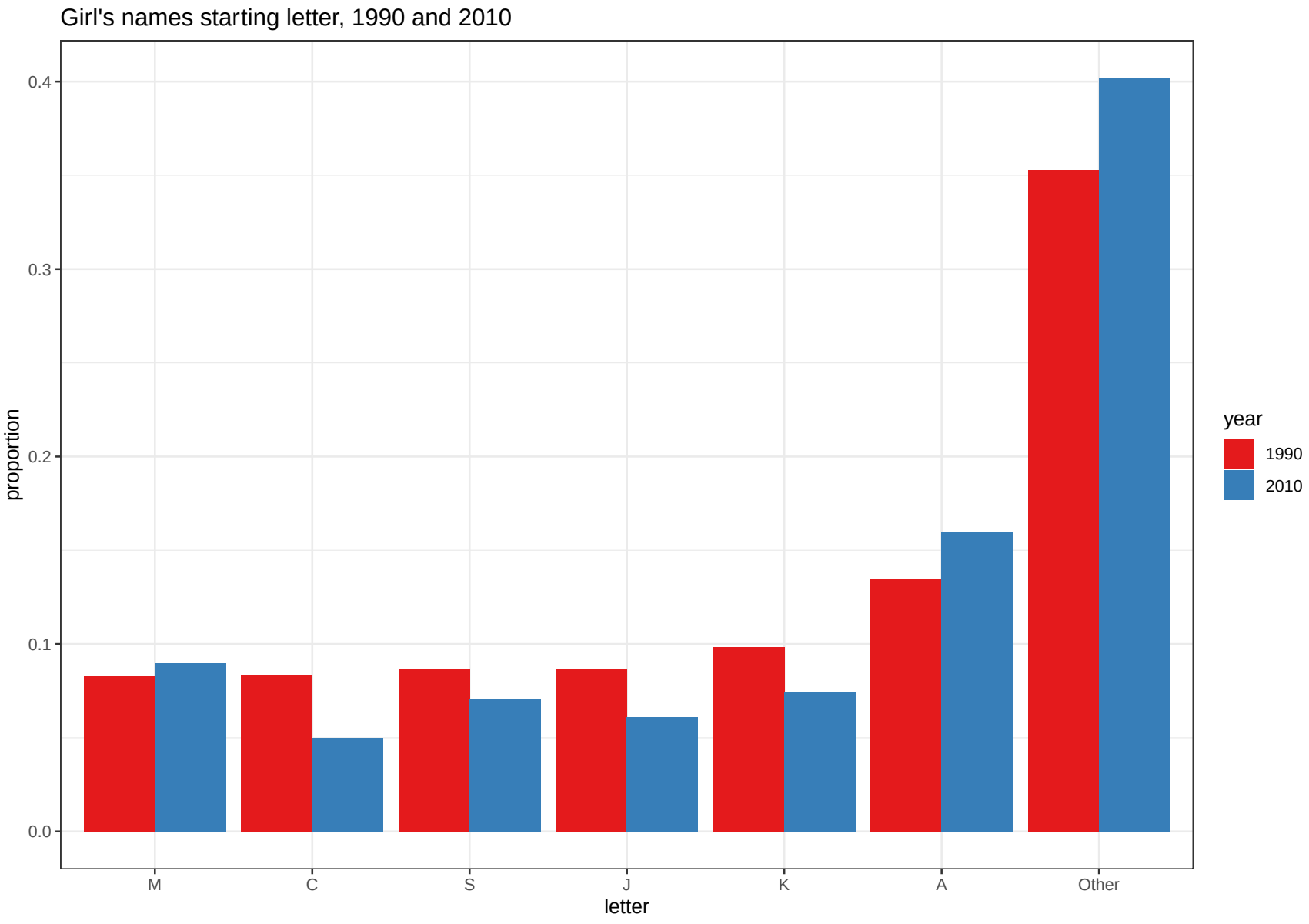

Girl's names by starting letter, 2010
J, 6.1%
C, 5%
K, 7.4%
A, 15.9%
M, 9%
S, 7.1%
Other, 40.2%

Pie charts

```
?pie
```

Pie charts are a very bad way of displaying information. The eye is good at judging linear measures and bad at judging relative areas. A bar chart or dot chart is a preferable way of displaying this type of data.

Alternative



Know your audience

Graphs can be used for

- our own exploratory data analysis
- to convey a message to experts,
- to help tell a story to a general audience.

Make sure that the intended audience understands each element of the plot.

Examples: spiral plot, log scales

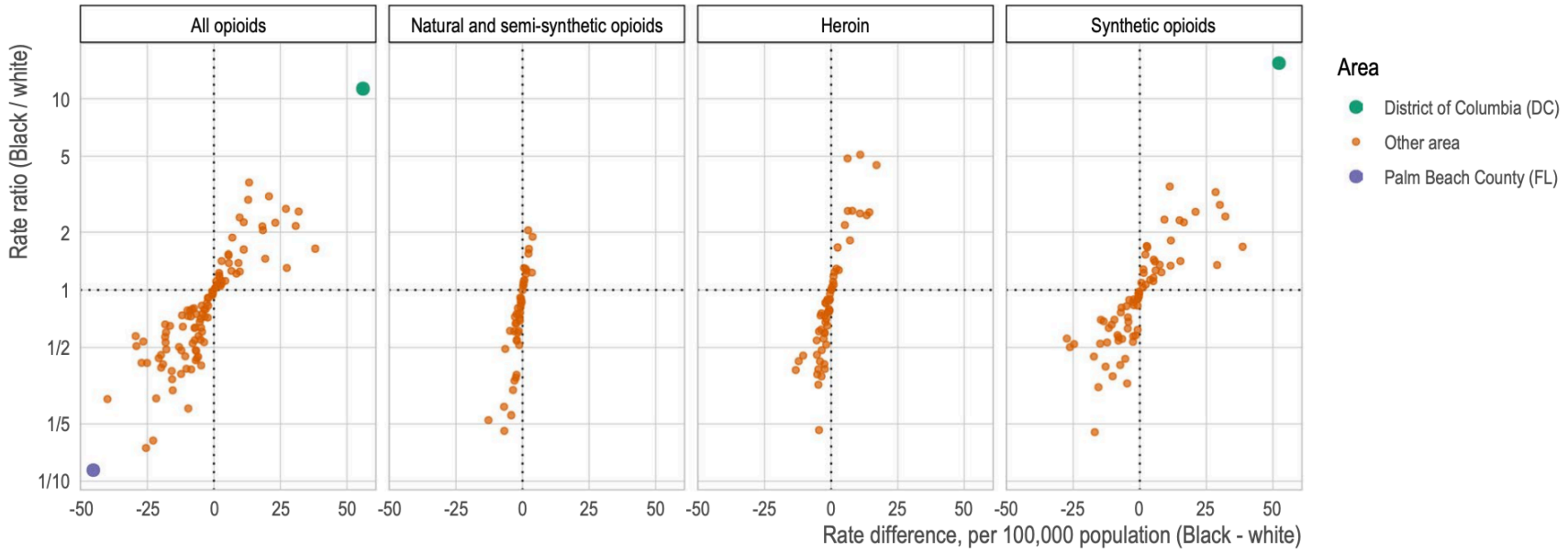
- Think of the color blind. In R, `viridis` and `brewer` palettes give colorblind-friendly options

Emphasize important patterns without being misleading

There is no such thing as information overload. There is only bad design. — Edward Tufte

- Eliminate distractions
- Highlight the essential
- Use color and text strategically
- Avoid pseudo-3D plots

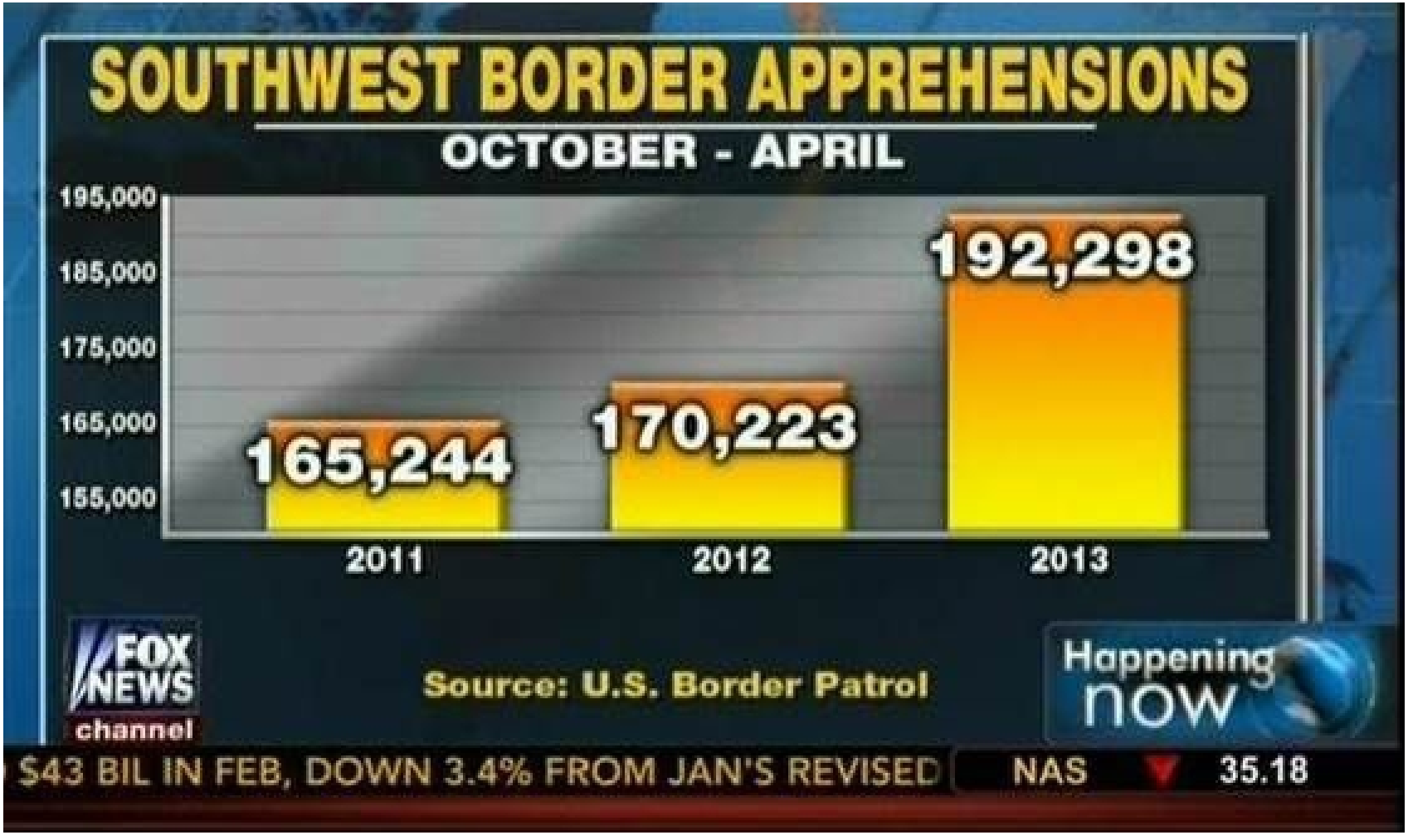
Highlight the essential



Source:
https://link.springer.com/article/10.1007/s11524-021-00573-8

When to start the axis at zero?

When to start the axis at zero?



Source

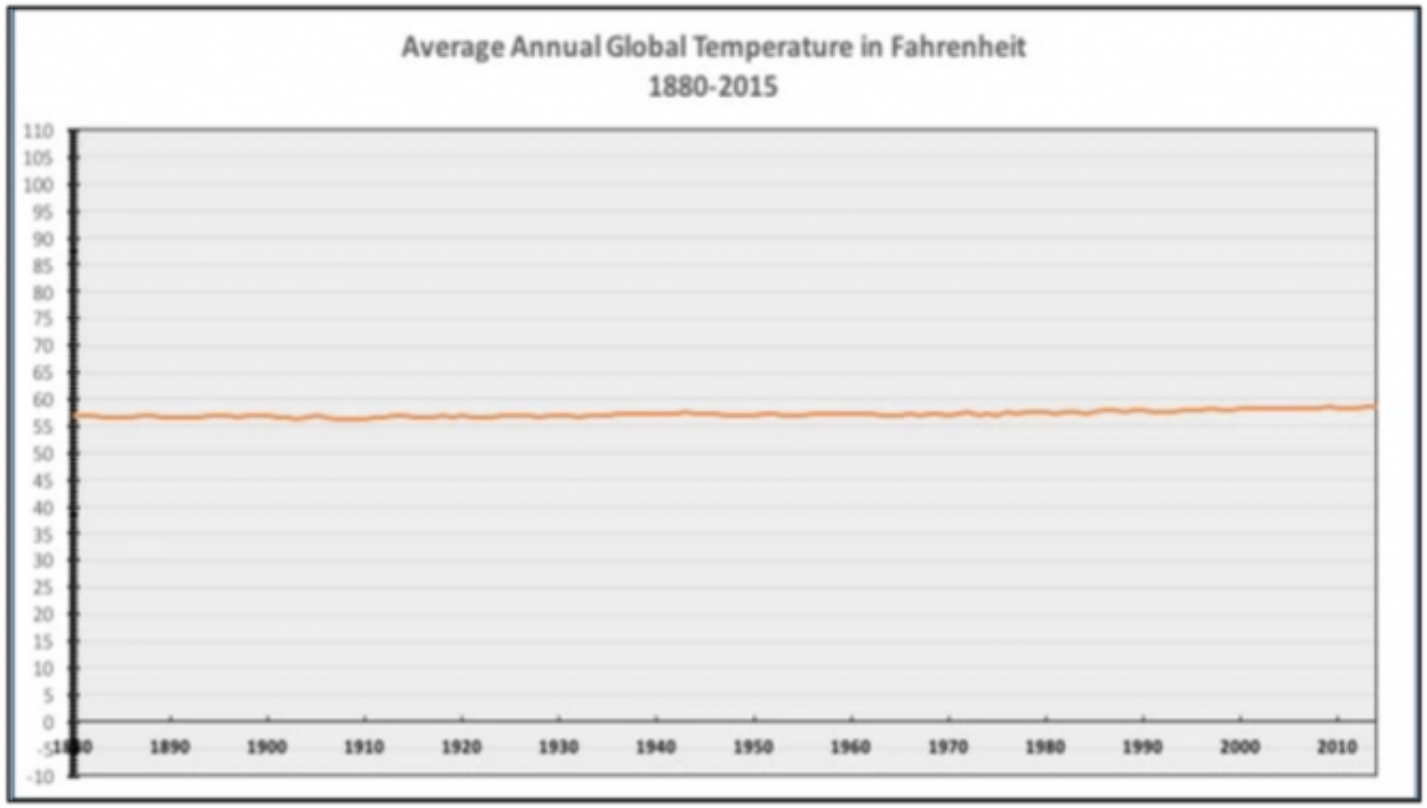
When to start the axis at zero?

National Review ✓
@NRO

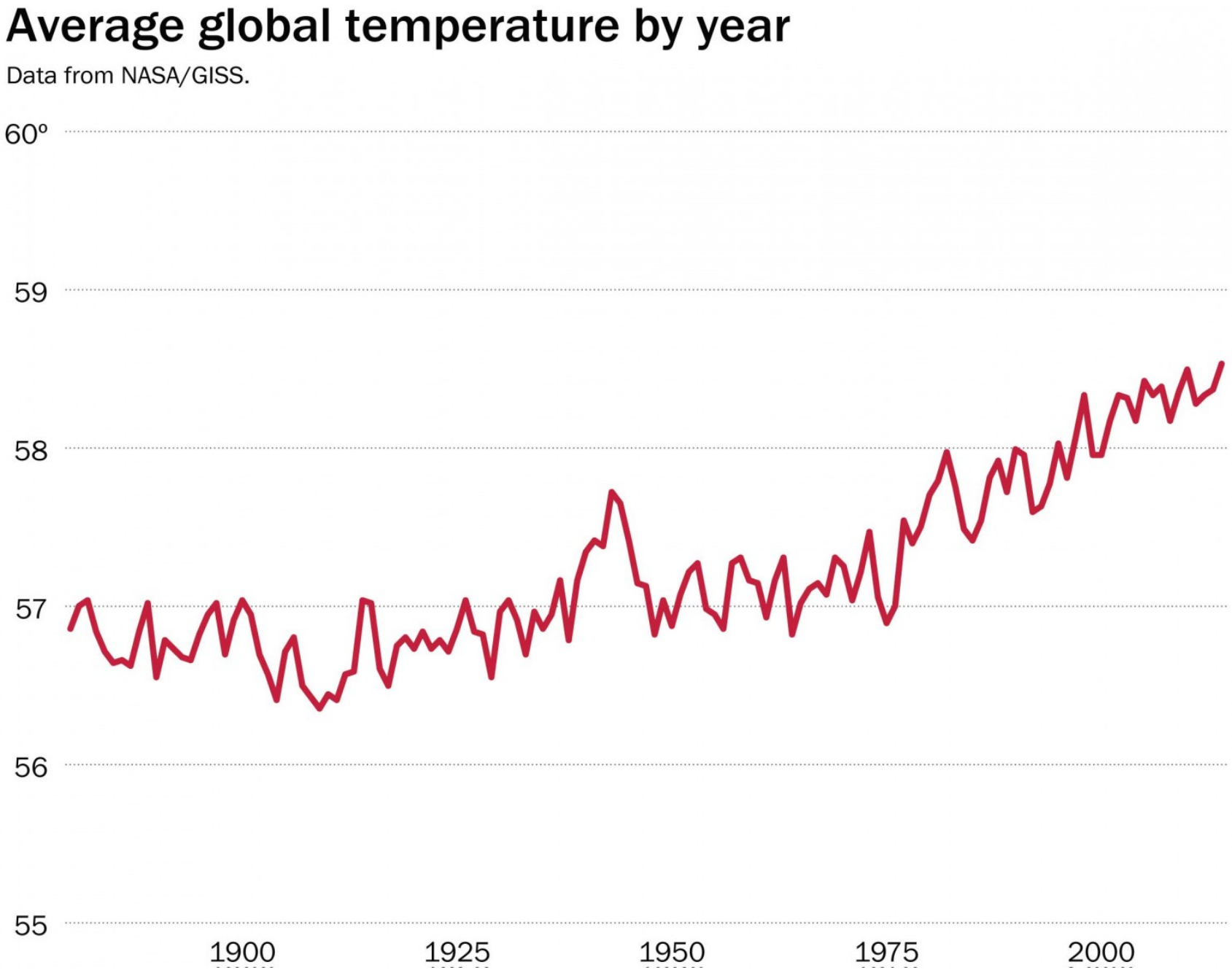
Follow

The only #climatechange chart you need to see. natl.re/wPKpro

(h/t @powerlineUS)

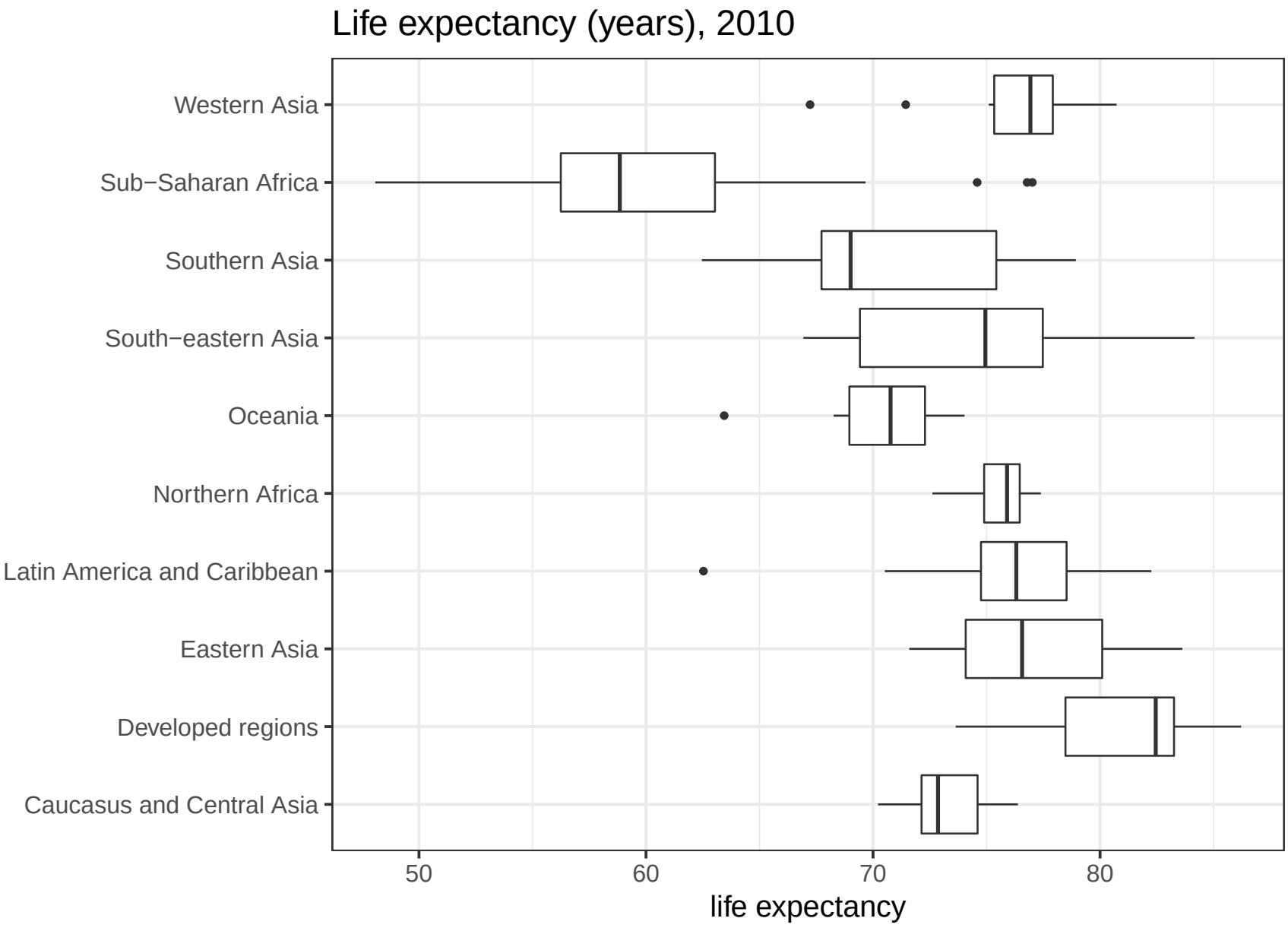


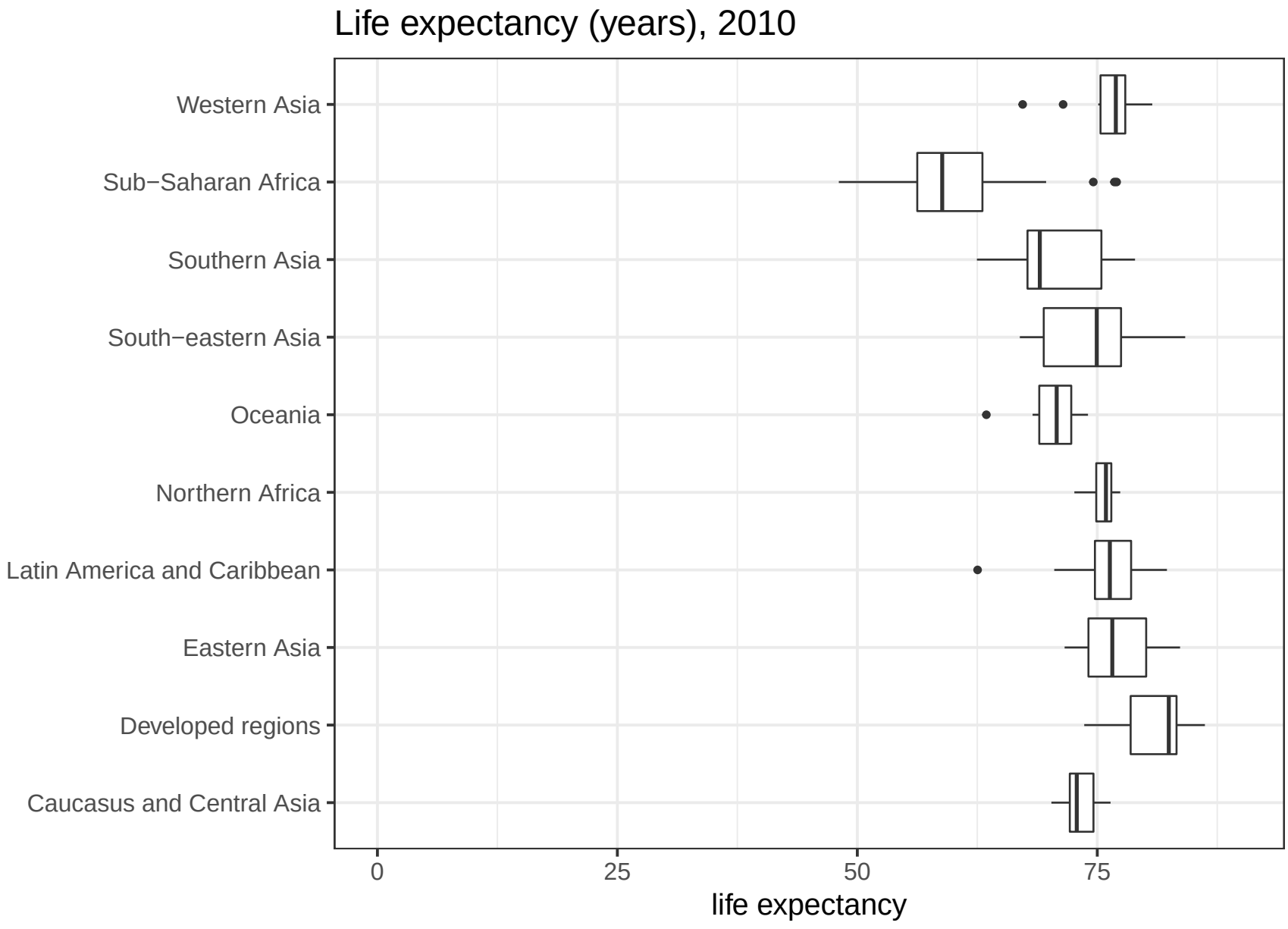
When to start the axis at zero?



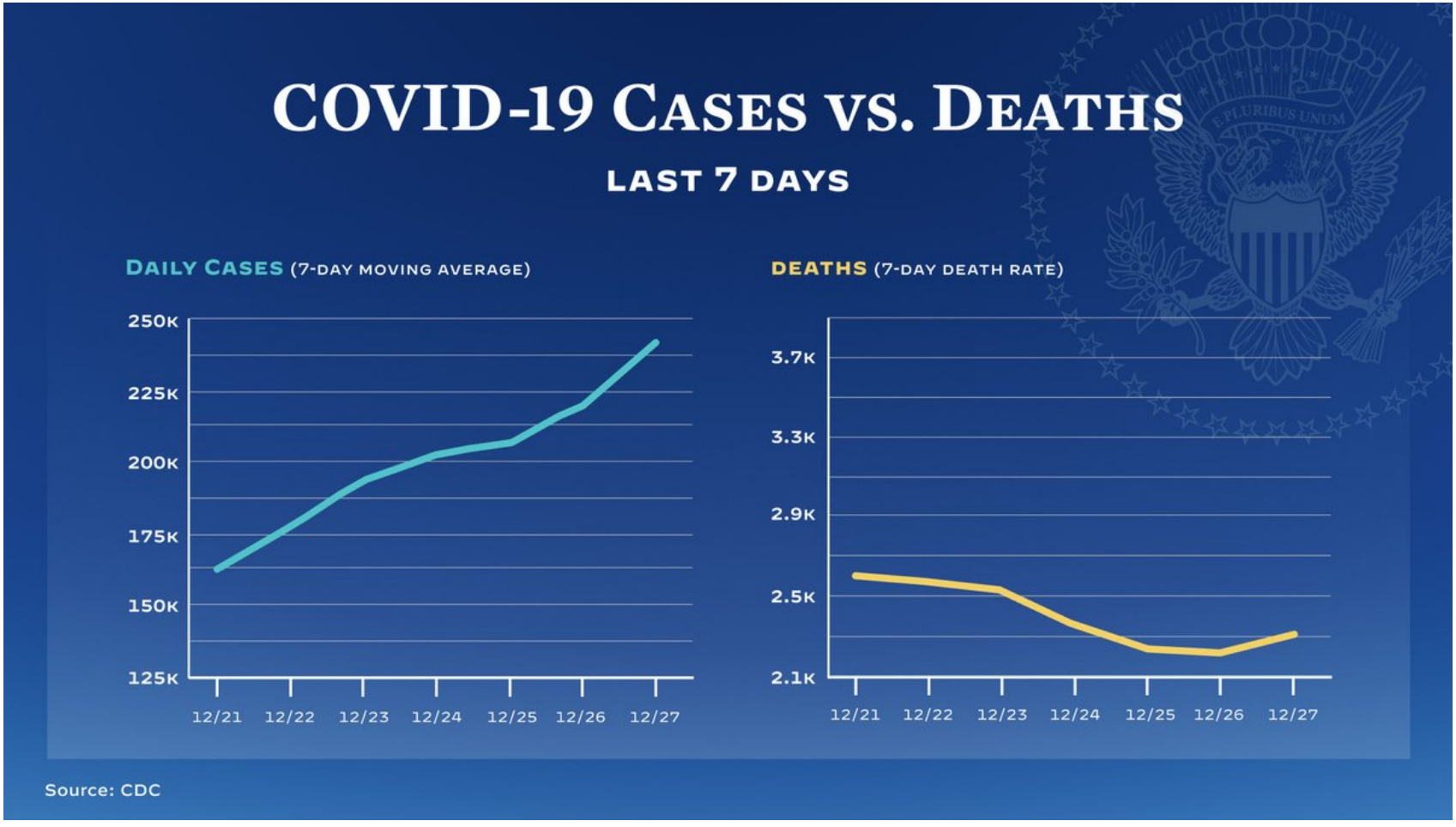
When to include zeroes

- With bar plots, we are implying the length is proportional to the quantities being displayed. By avoiding 0, relatively small differences can be made to look much bigger than they actually are.
- With line plots or plots that use position, it is not neccessary to start the axis at zero (and could be misleading)

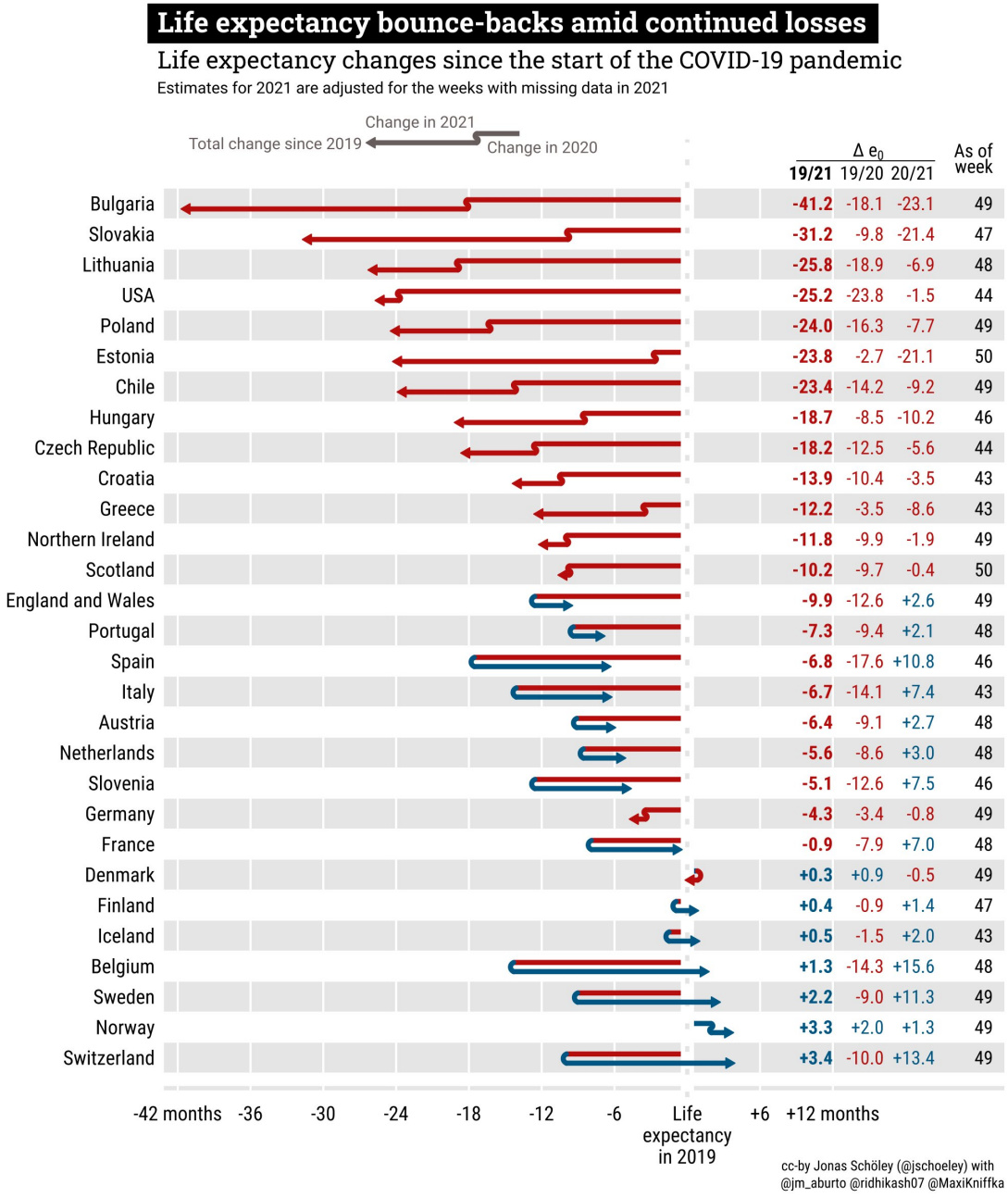

Life expectancy (years), 2010
Western Asia
Sub−Saharan Africa
Southern Asia
South−eastern Asia
Oceania
Northern Africa
Latin America and Caribbean
Eastern Asia
Developed regions
Caucasus and Central Asia
50
60
70
80
life expectancy


Life expectancy (years), 2010
Western Asia
Sub−Saharan Africa
Southern Asia
South−eastern Asia
Oceania
Northern Africa
Latin America and Caribbean
Eastern Asia
Developed regions
Caucasus and Central Asia
0
25
50
75
life expectancy

Emphasize important patterns without being misleading



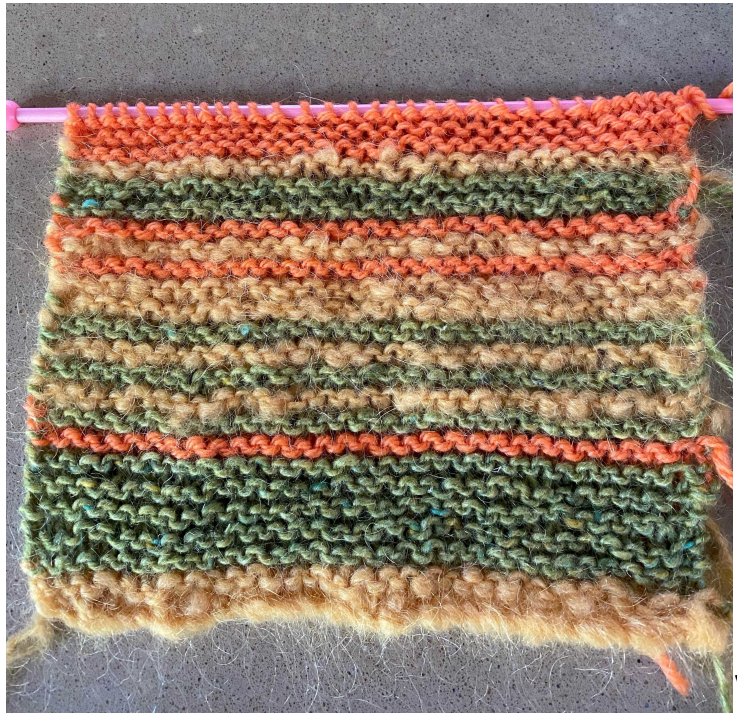
Clear, effective designs



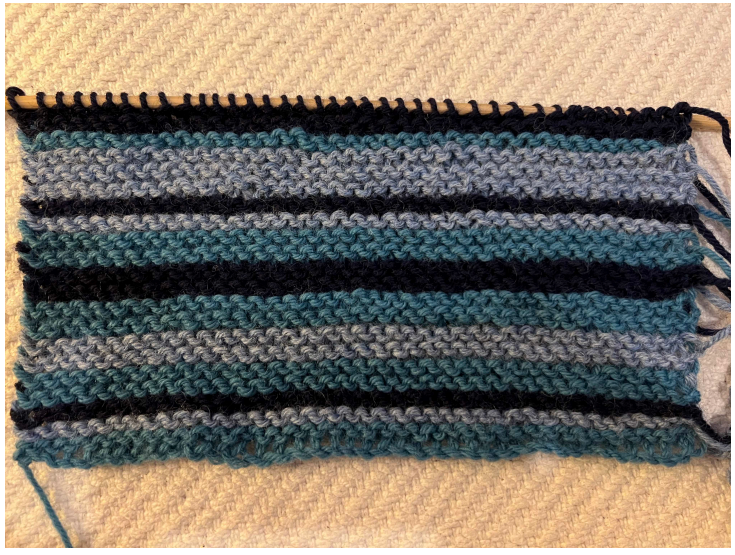
Important types of graphs: next week

- Histograms
- Bar charts
- Boxplots
- Line plots
- Scatter plots

...but data visualization need not be a graph



Hobart, Tasmania, Australia



Toronto, Ontario, Canada

Data ideas

- IPUMS: https://ipums.org/
- ICPSR: https://www.icpsr.umich.edu/web/pages/ICPSR/thematic-collections.html
- CHASS SDA: https://datacentre.chass.utoronto.ca/
- Toronto Open Data Portal: https://open.toronto.ca/ or use `opendatatoronto` R package (ask for code)
- UN WPP: https://population.un.org/wpp/
- NBER: https://www.nber.org/research/data?page=1&perPage=50