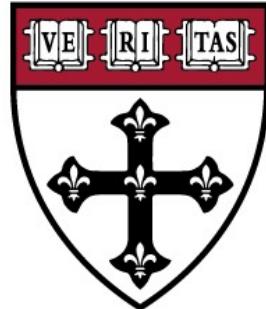


# Data Visualization Principles

Summer Institute in Data Science

Rolando J. Acosta

1



**HARVARD**  
SCHOOL OF PUBLIC HEALTH

June 25, 2020



@RJANunez

# What to expect today

- Today we will show some bad examples of data visualization, discuss how we can improve on them, and use these as motivation for a list of principles
- The principles we will discuss today are mostly based on research related to how humans detect patterns and make visual comparisons
- Things to consider:
  1. Choose a visualization technique that best fit the way our brains process visual information
  2. Have a goal in mind
  3. Know your audience

# Encoding data using visual cues

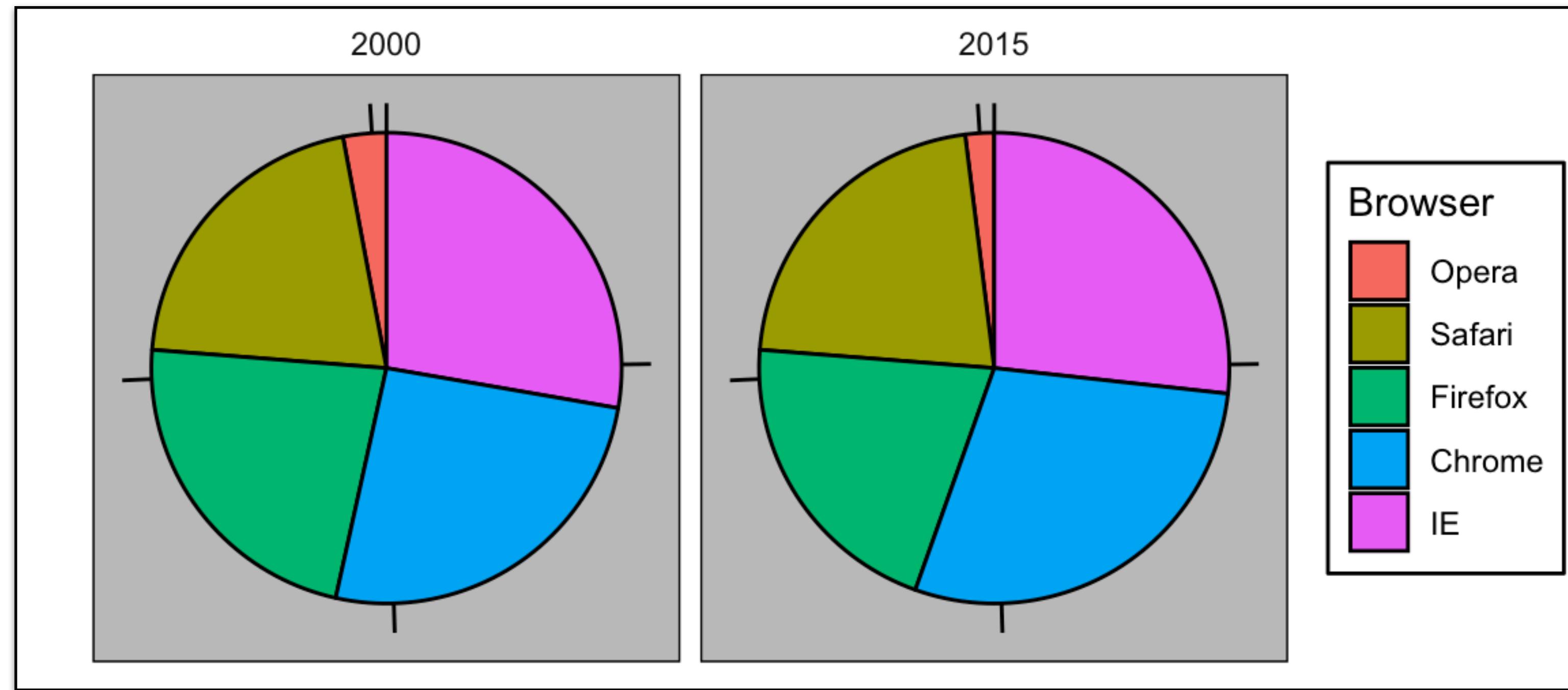
- There are several approaches at our disposal including:
  1. Position
  2. Aligned lengths
  3. Angles
  4. Area
  5. Brightness
  6. Color hue

# First example

- Suppose we want to report the results from two hypothetical regarding browser preference taken in 2000 and then 2015
- The options were:
  1. Opera
  2. Safari
  3. Firefox
  4. Chrome
  5. Internet explorer
- Note that for each year we are simply comparing 5 quantities (the 5 percentages)

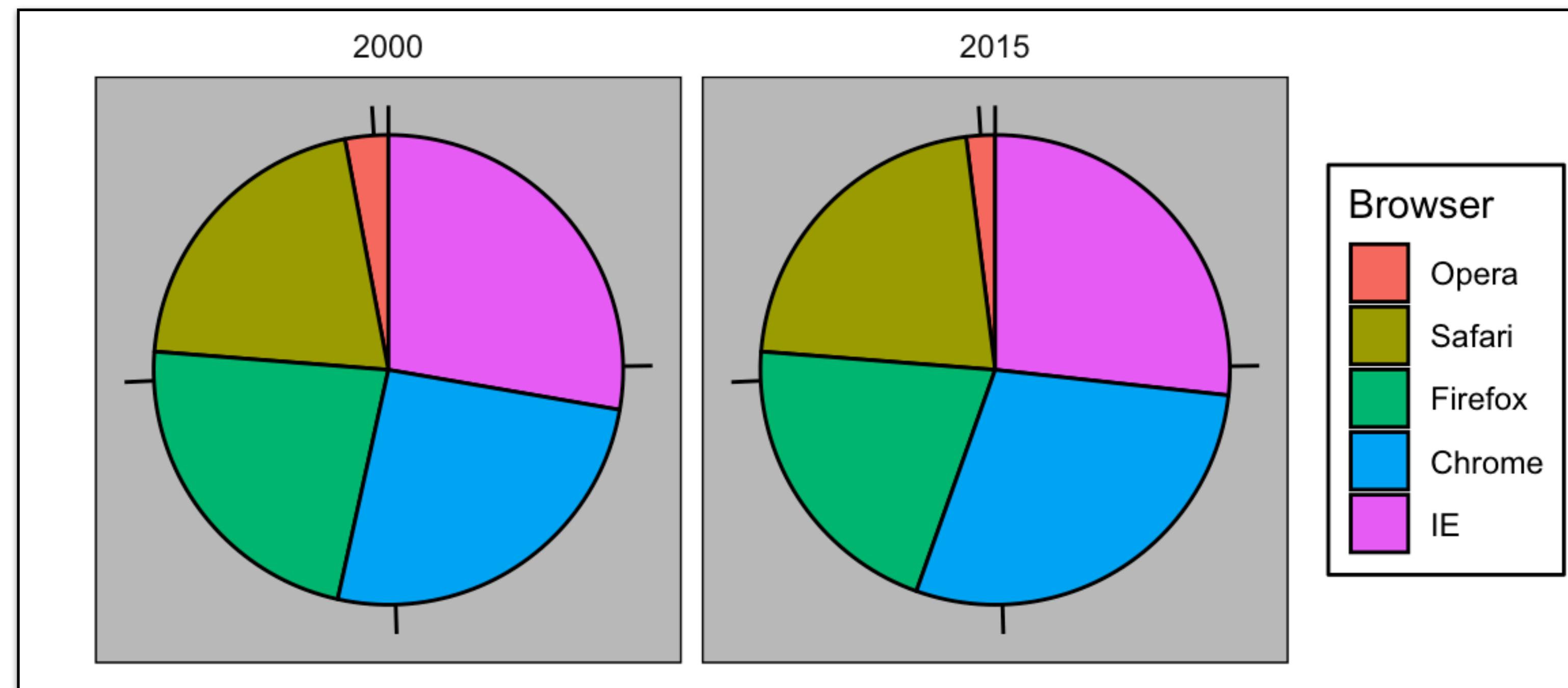
# First example

- A common representation of percentages, popularized by Microsoft Excel, is the pie chart:



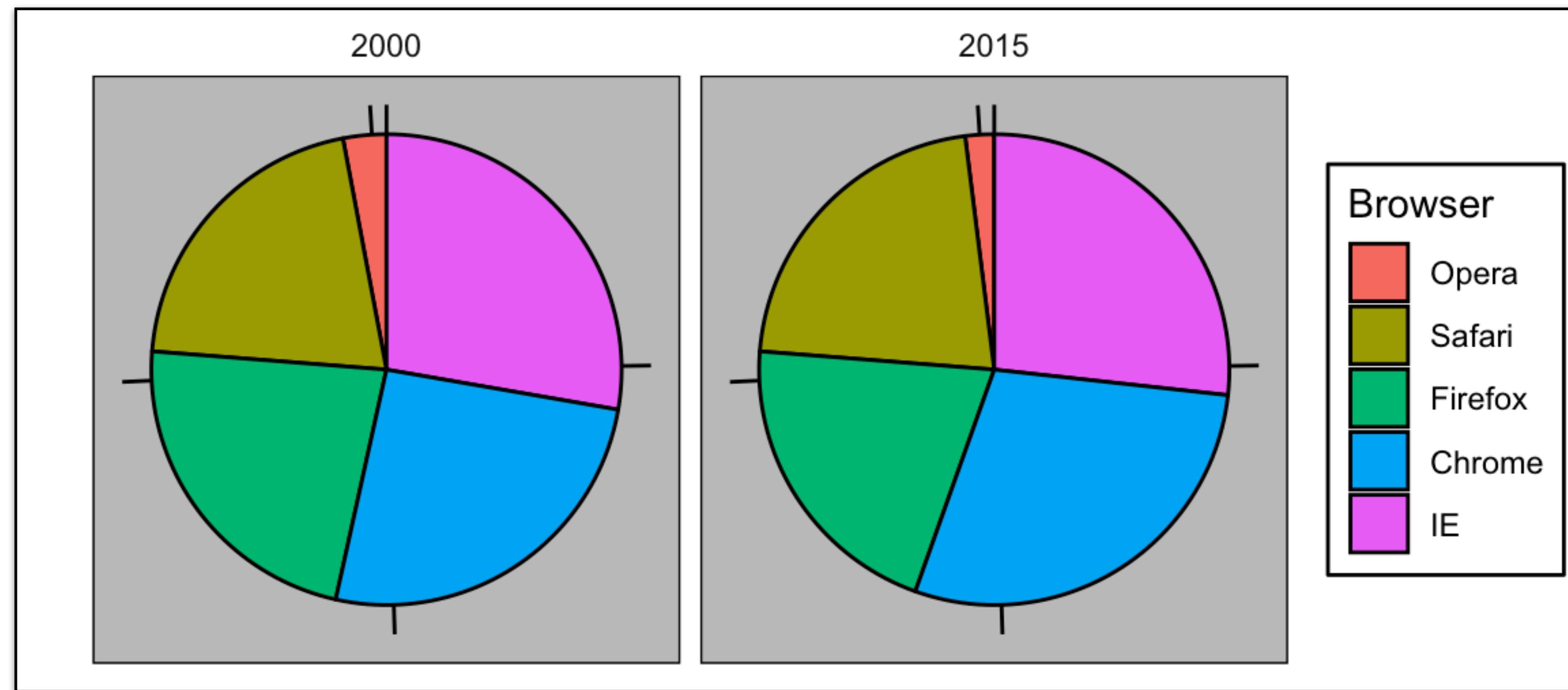
# First example

- Here we are representing quantities using both areas and angles, since both the angle and area of each slice is proportional to the percentage it represents
- It turns out that perception studies conclude that humans are not good at quantifying angles and even worse at quantifying areas



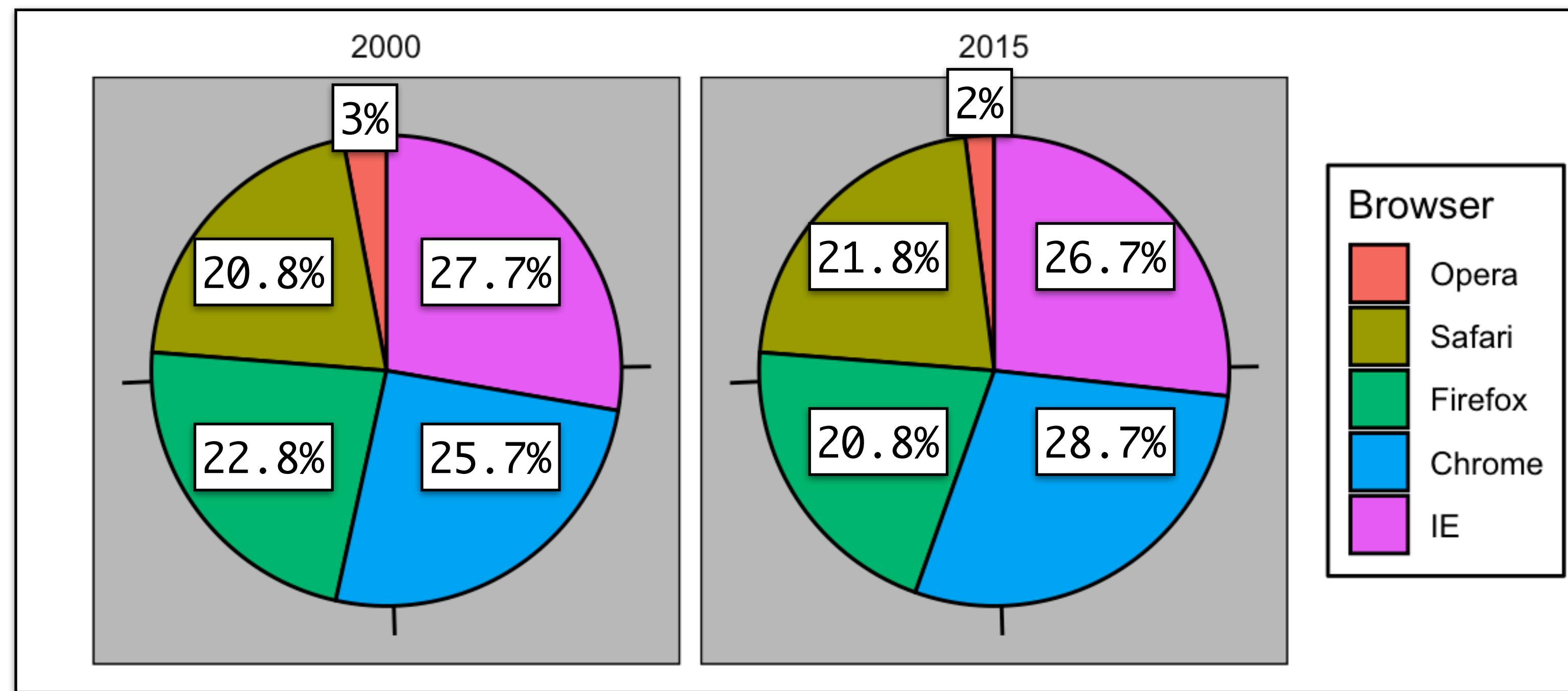
# First example

- To see how bad we are at visually quantifying angles and areas let's try to estimate the percentage of respondents that prefer Firefox
- Any guesses?



# First example

- To see how bad we are at visually quantifying angles and areas let's try to estimate the percentage of respondents that prefer Firefox
- Any guesses?



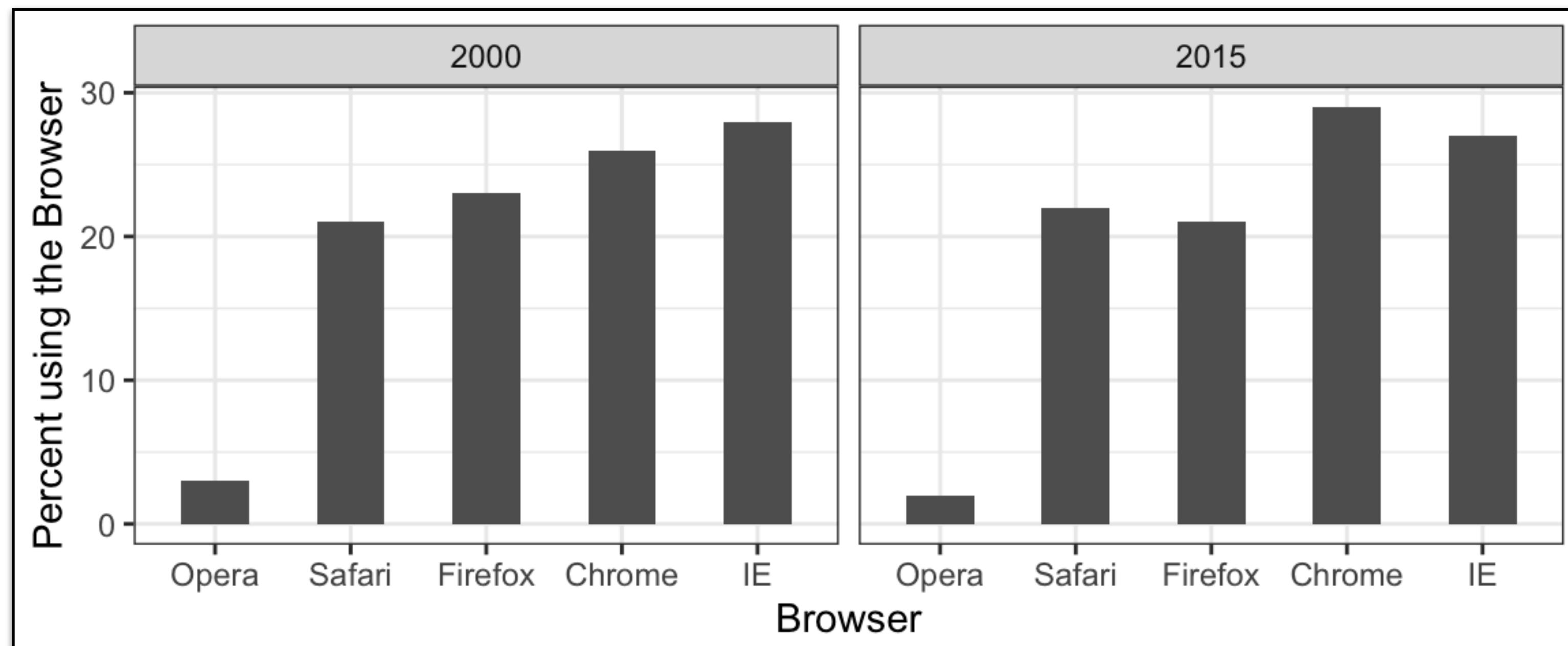
# First example

- Here is the data
- In this case, its better to just present a table

Browser	2000	2010
Opera	3	2
Safari	21	22
Firefox	23	21
Chrome	26	29
Internet explorer	28	27

# First example

- The preferred way to plot these data is with a barplot
- Here we use length and position as visual cues



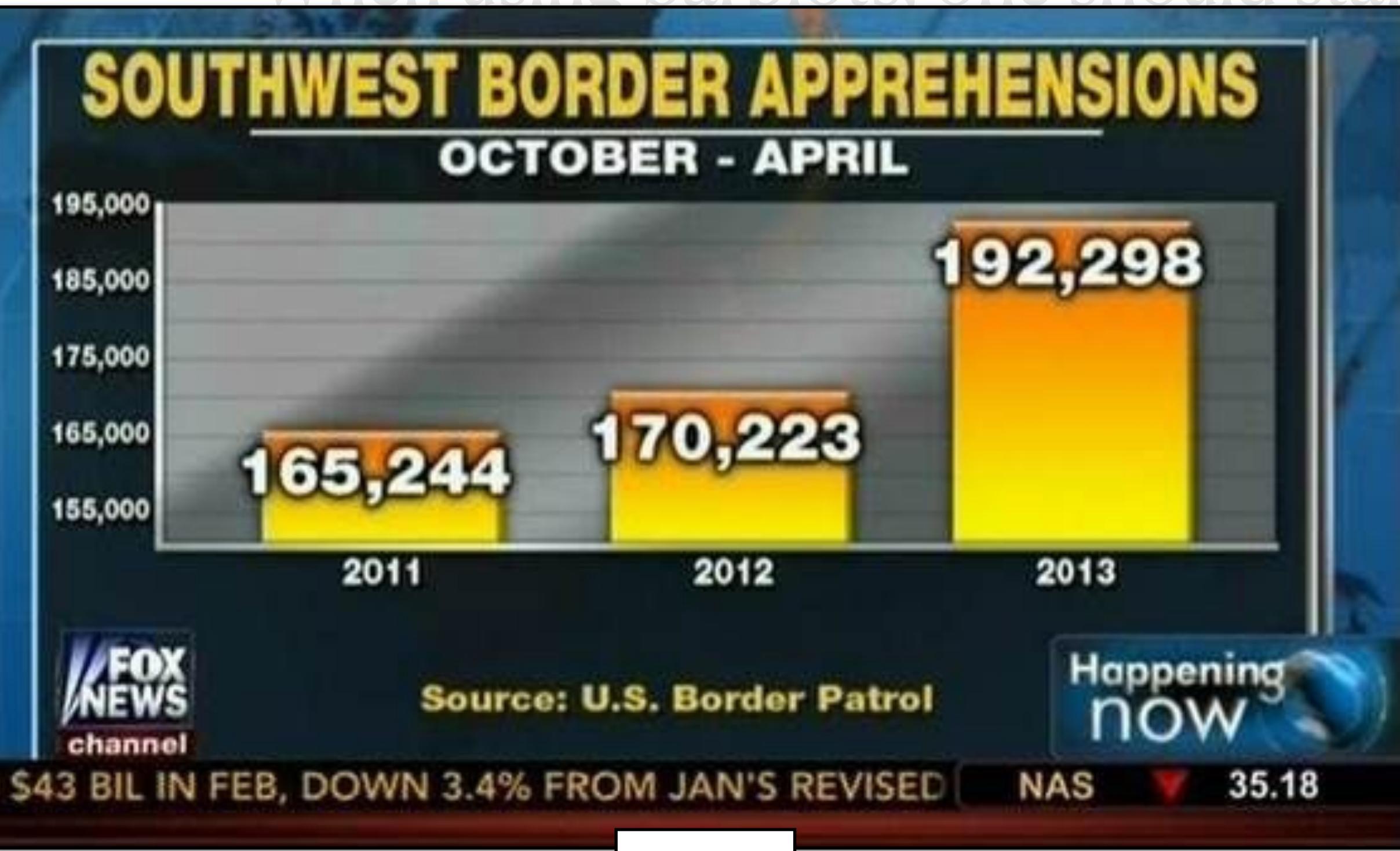
- If for some reason you have to use pie charts, label each slice with the corresponding numerical value

# Know when to include 0

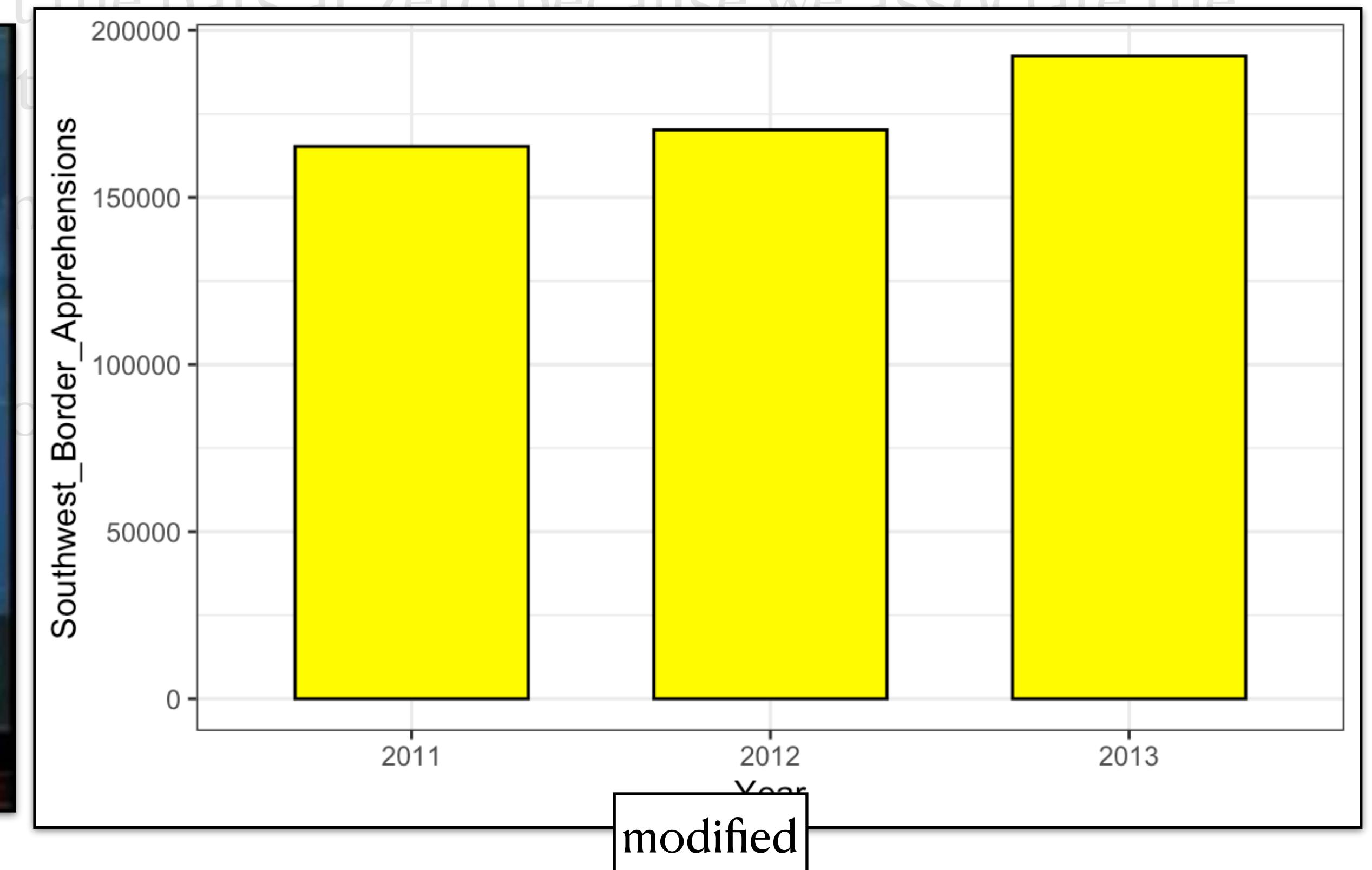
- When using barplots, one should start the bars at zero because we associate the length to be proportional to the quantities being displayed
- By avoiding zero, small differences can be made to look bigger than they actually are
- Take this example used by Peter Aldhous [here](#)

# Know when to include 0

- When using barplots, one should start the bars at zero because we associate the



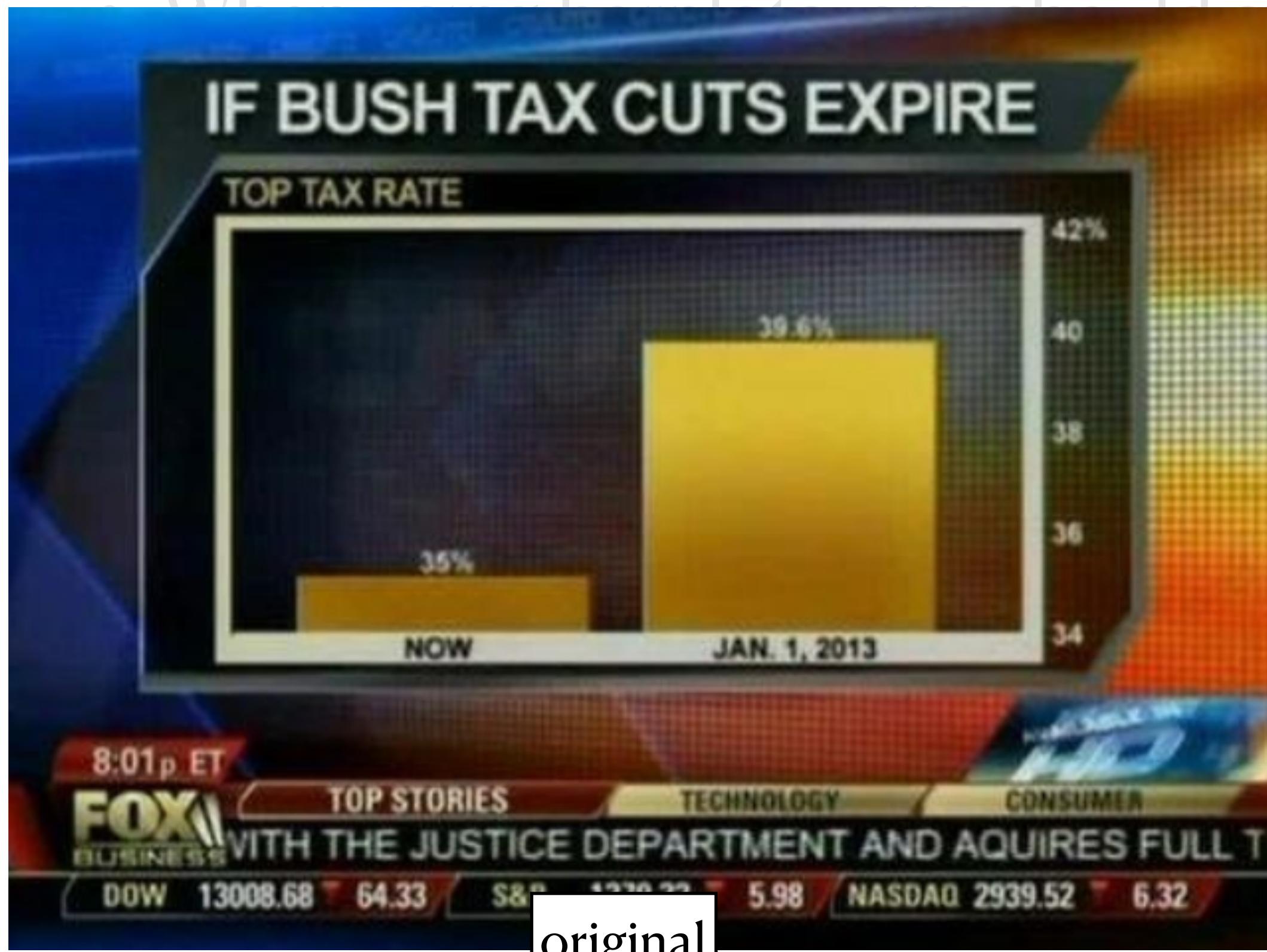
original



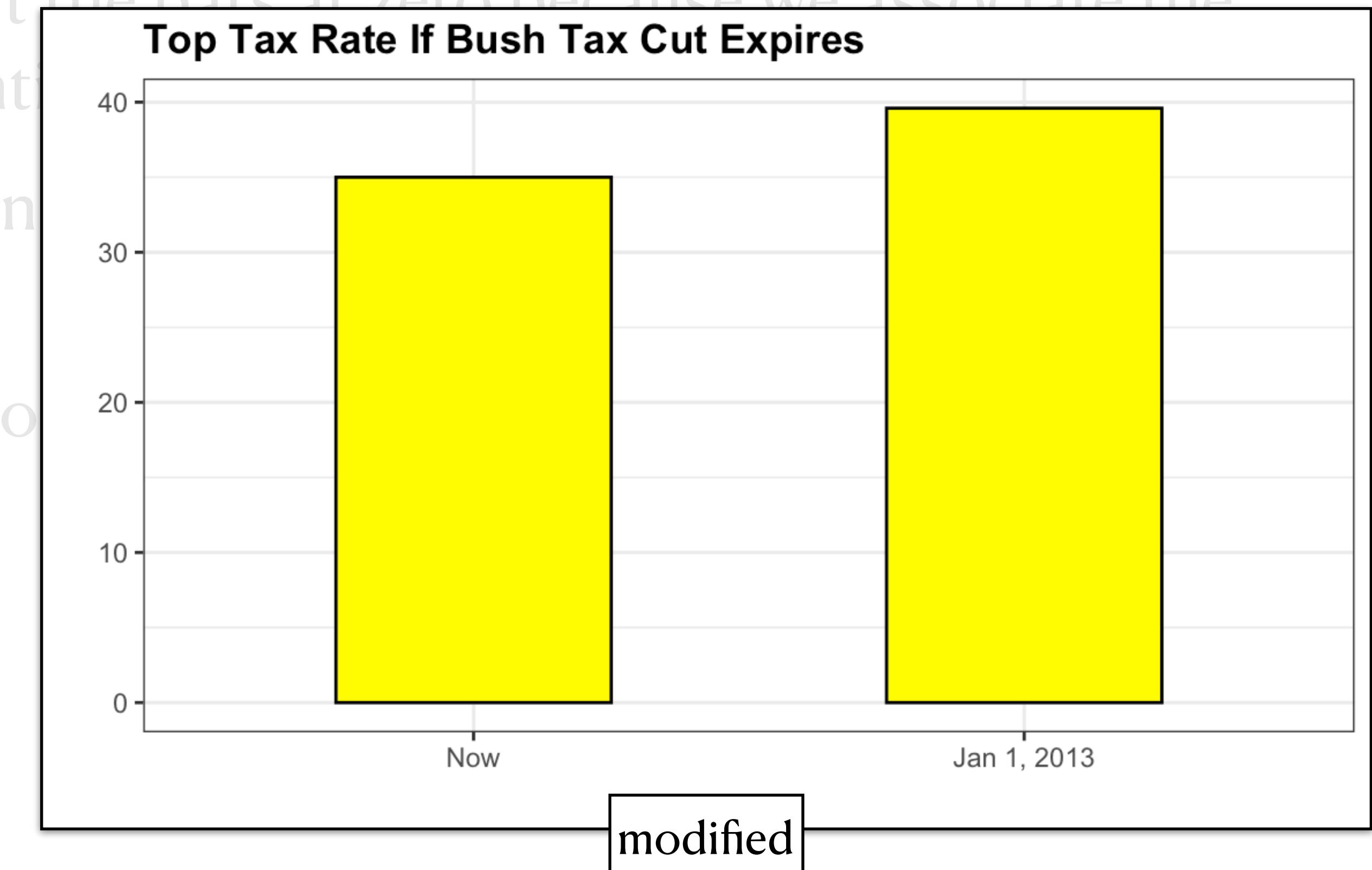
modified

- The original figure makes a 16% increase look a 3-fold change

# Know when to include 0



original

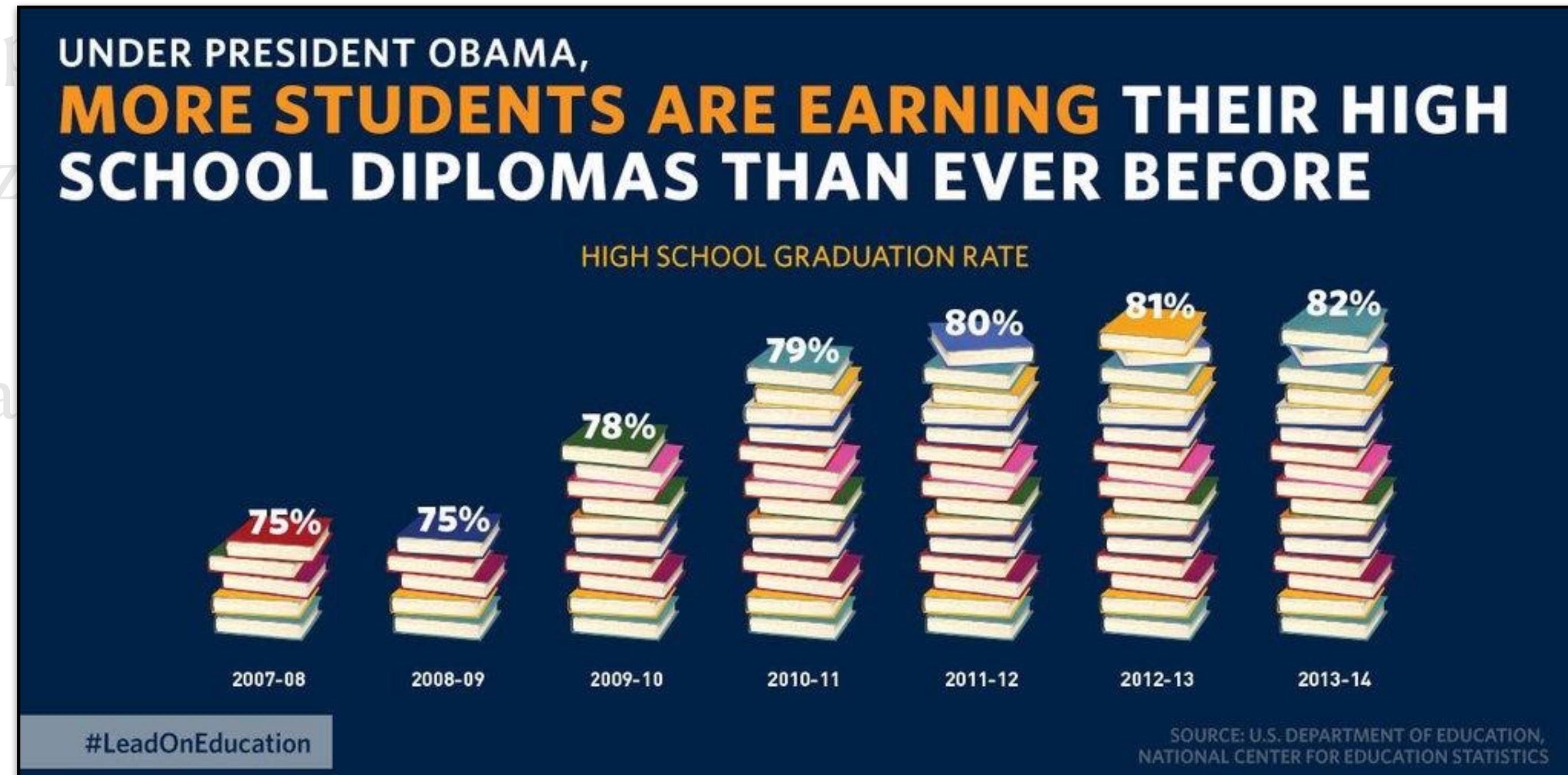


modified

- This one makes a 13% increase look like a 5-fold change!

# Know when to include 0

- When using barplots, one should start the bars at zero because we associate the length to be proportional to the value.
- By avoiding zero, it makes the values look like they are increasing when they actually are.
- Take this example:

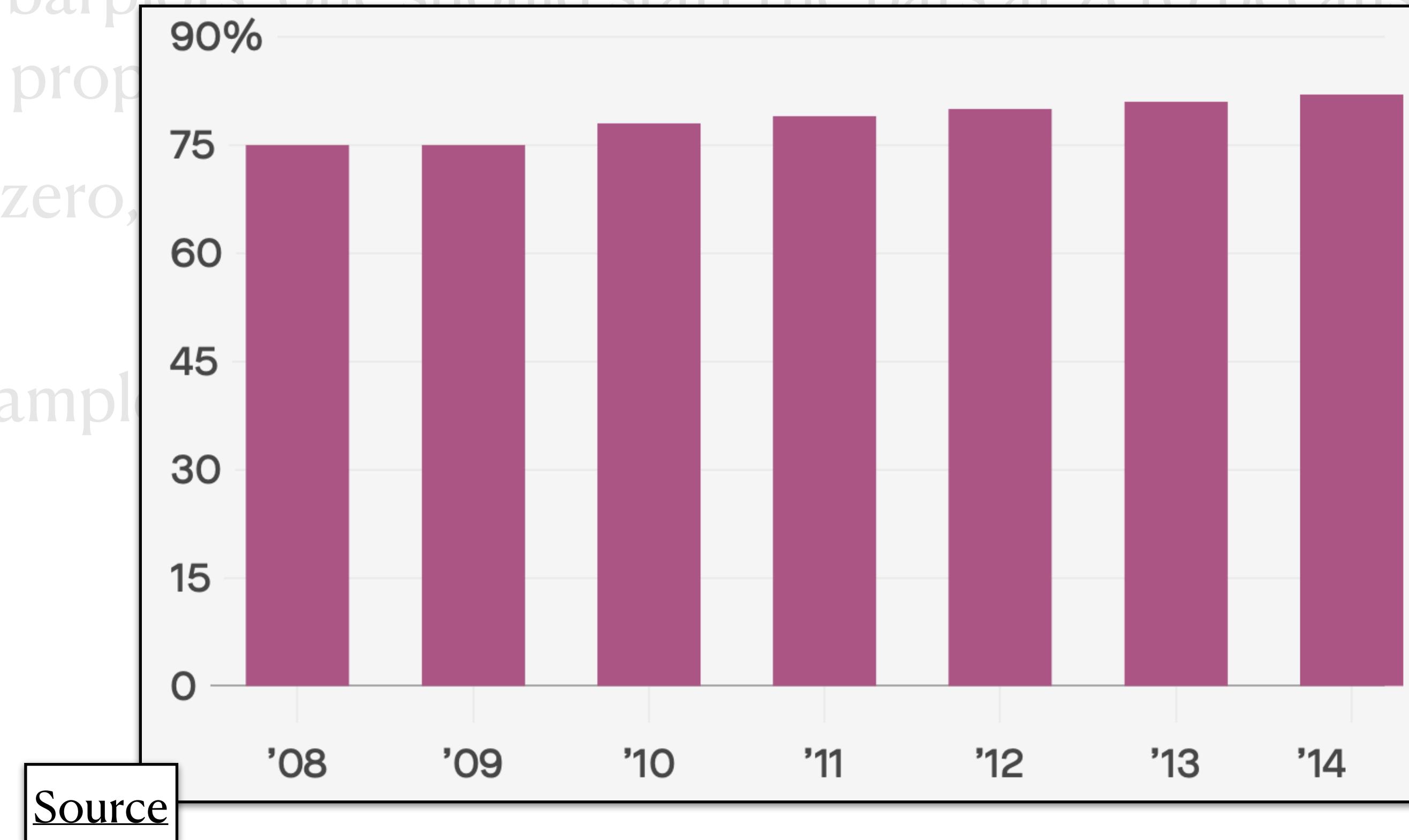


Source

- What does 5 books (75%) mean? What about 16 books(82%)?

# Know when to include 0

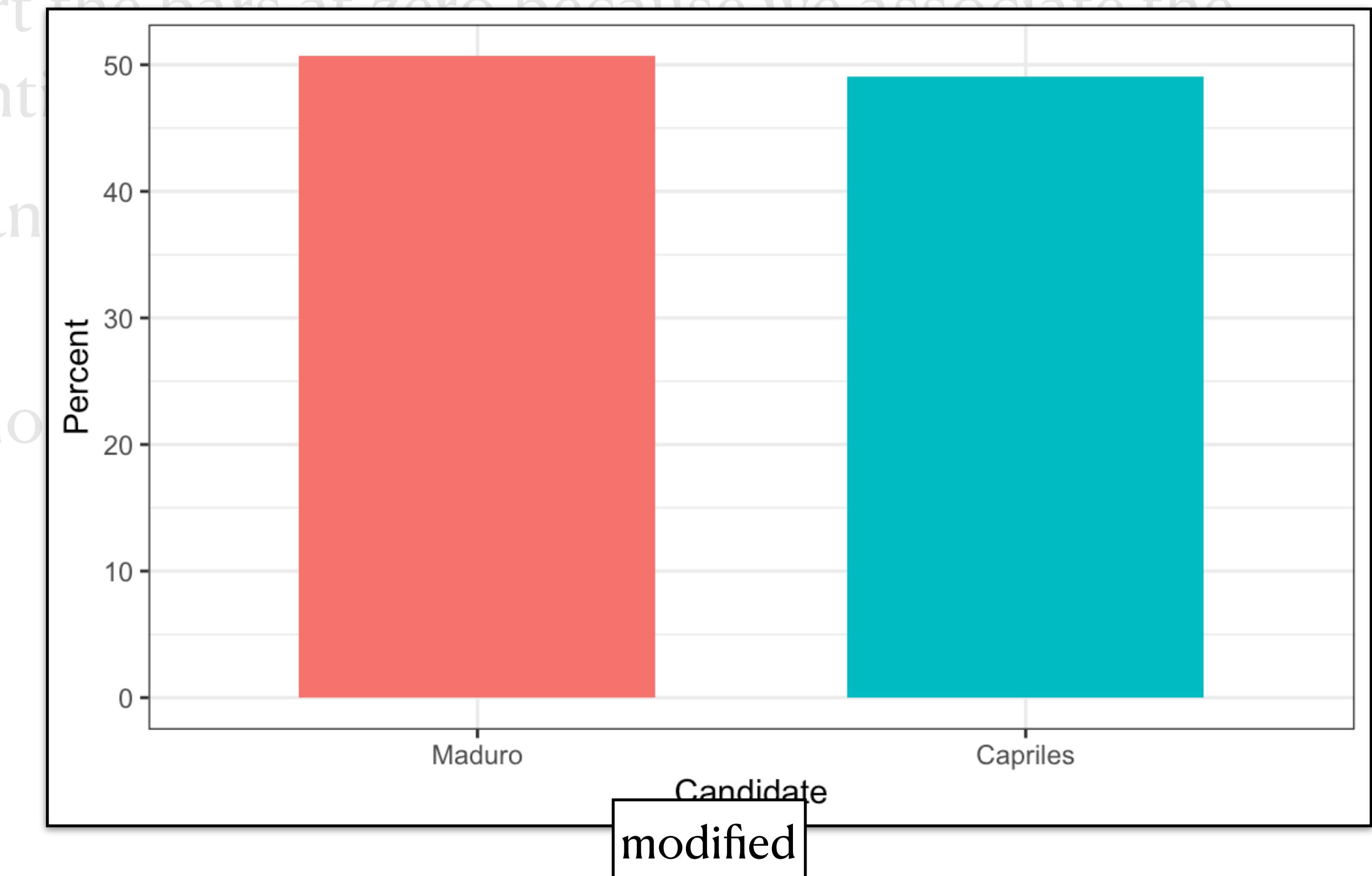
- When using barplots, one should start the bars at zero because we associate the length to be proportional to the value.
- By avoiding zero, we make values look larger than they actually are.
- Take this example:



- A better version

# Know when to include 0

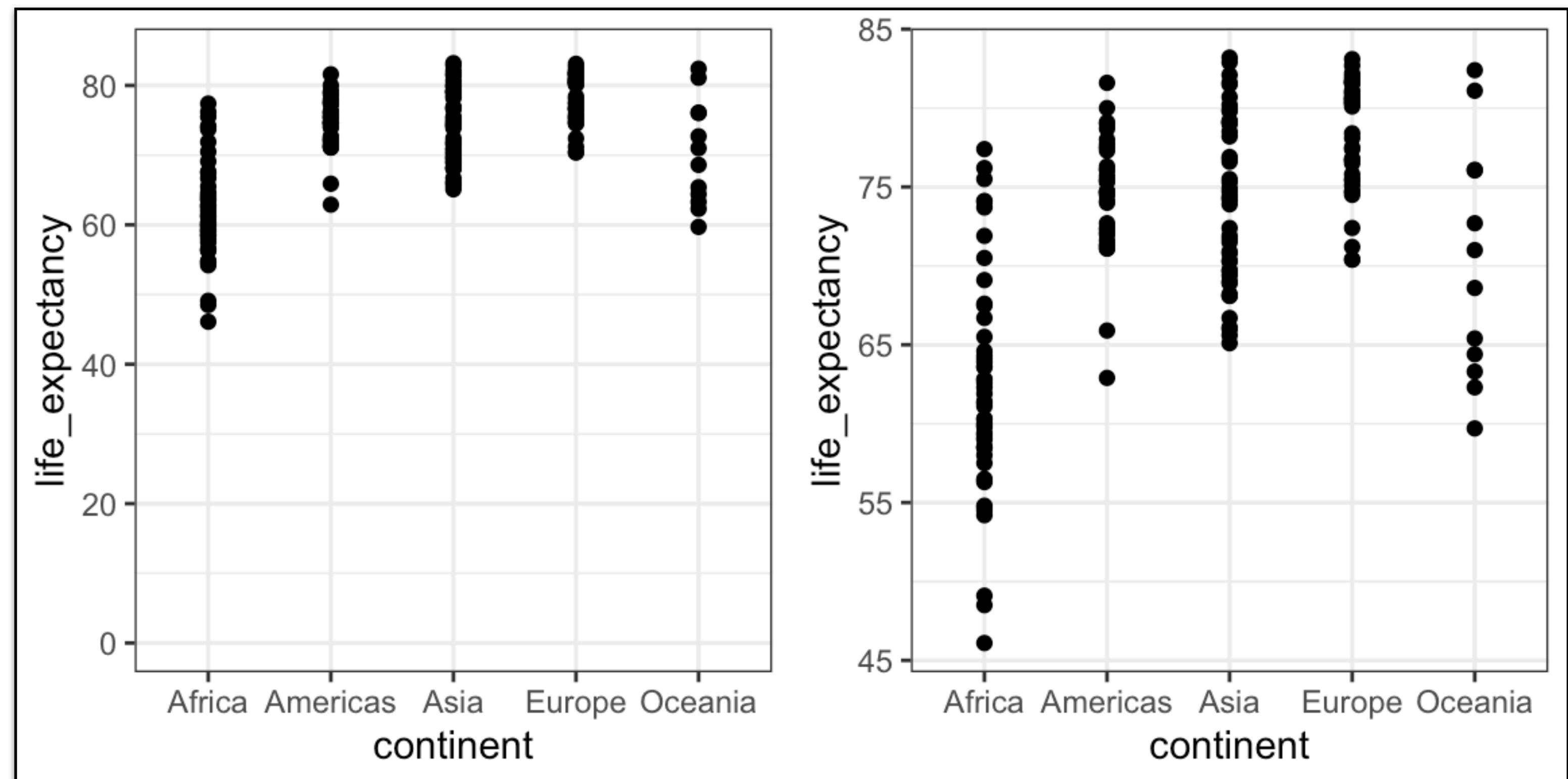
- When using barplots, one should start the bars at zero because we associate the



- This one makes a 3% increase look like all of Venezuela voted for Maduro

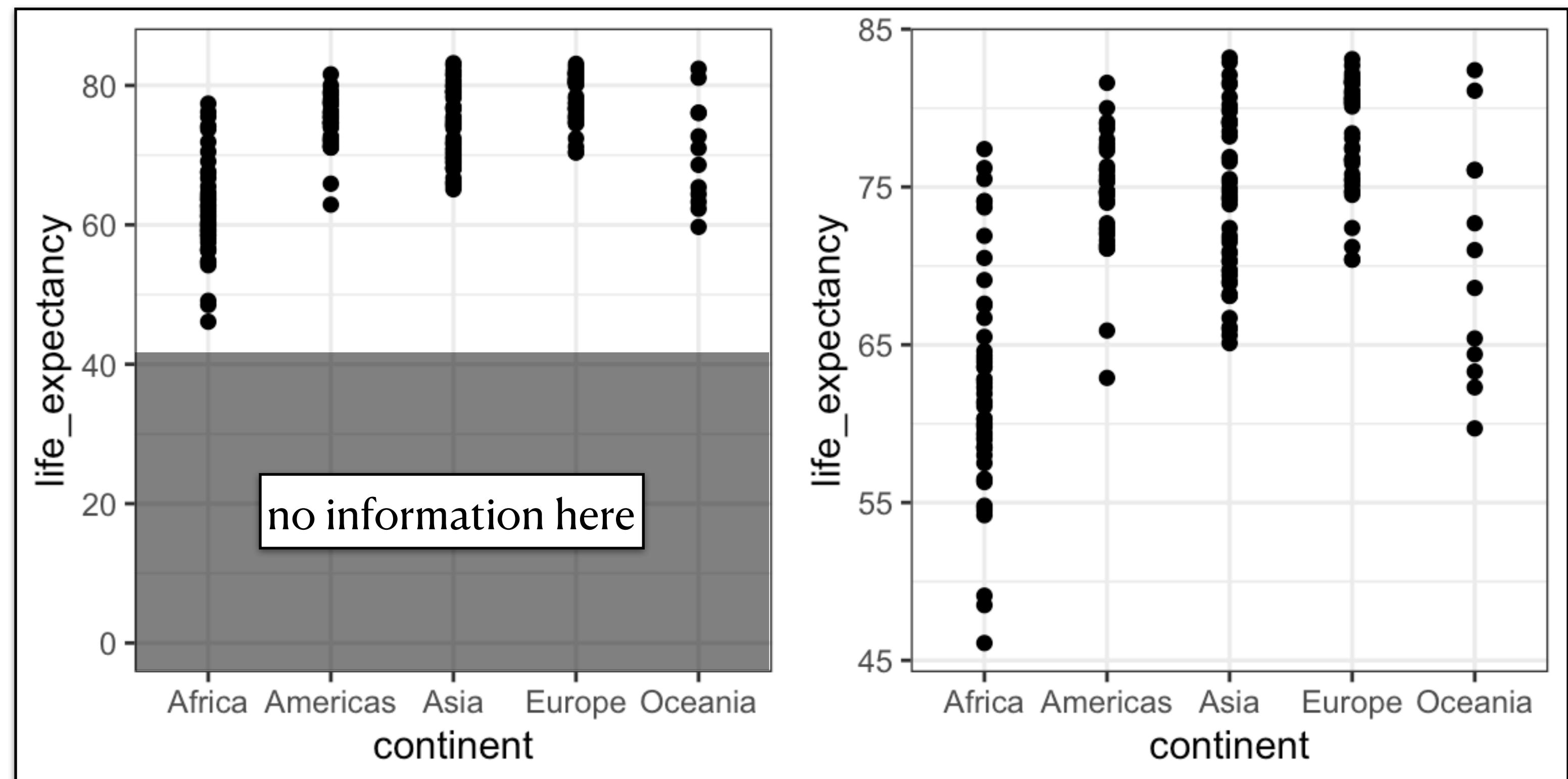
# Know when to include 0

- If, for example, one uses position instead of length as a visual cue then its not necessary to include 0
- This is particularly the case when we want to compare between-group to within-group variability



# Know when to include 0

- If, for example, one uses position instead of length as a visual cue then its not necessary to include 0
- This is particularly the case when we want to compare between-group to within-group variability



# Do not distort quantities

- During President Obama's 2011 State of the Union Address, the following chart was used to show the US GDP against four competing nations
- Judging by the area of the circles, it seems that the US has an economy over five times larger than China's economy and over 20 plus times that of France



# Do not distort quantities

- But look at the numbers:

US/China  $14.6/5.7 = 2.6$

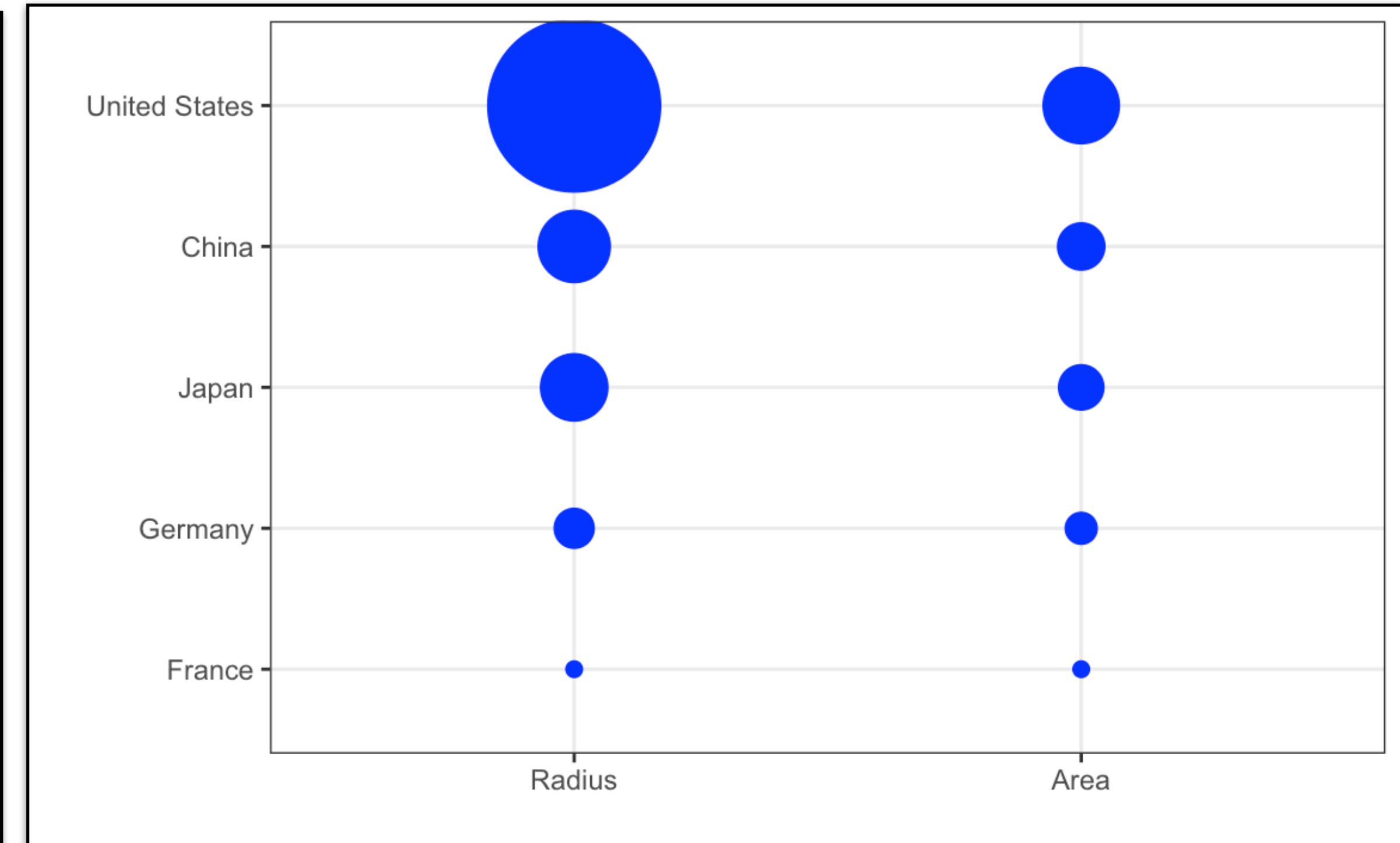
US/France  $14.6/2.5 = 5.8$

- Why the distortion? It turns out that the radius of the circle, instead of the area, is proportional to the quantity



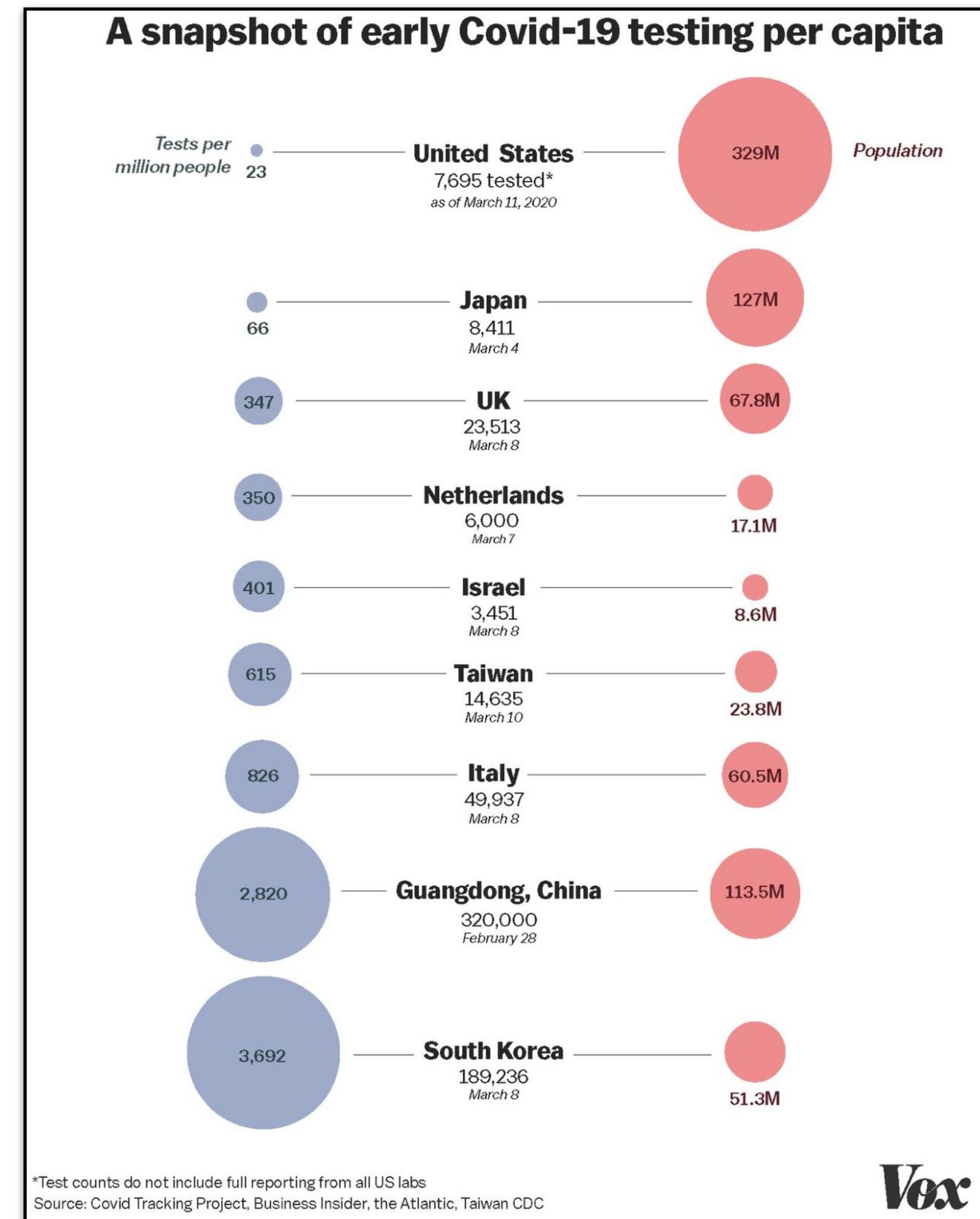
# Do not distort quantities

- Creating a better version

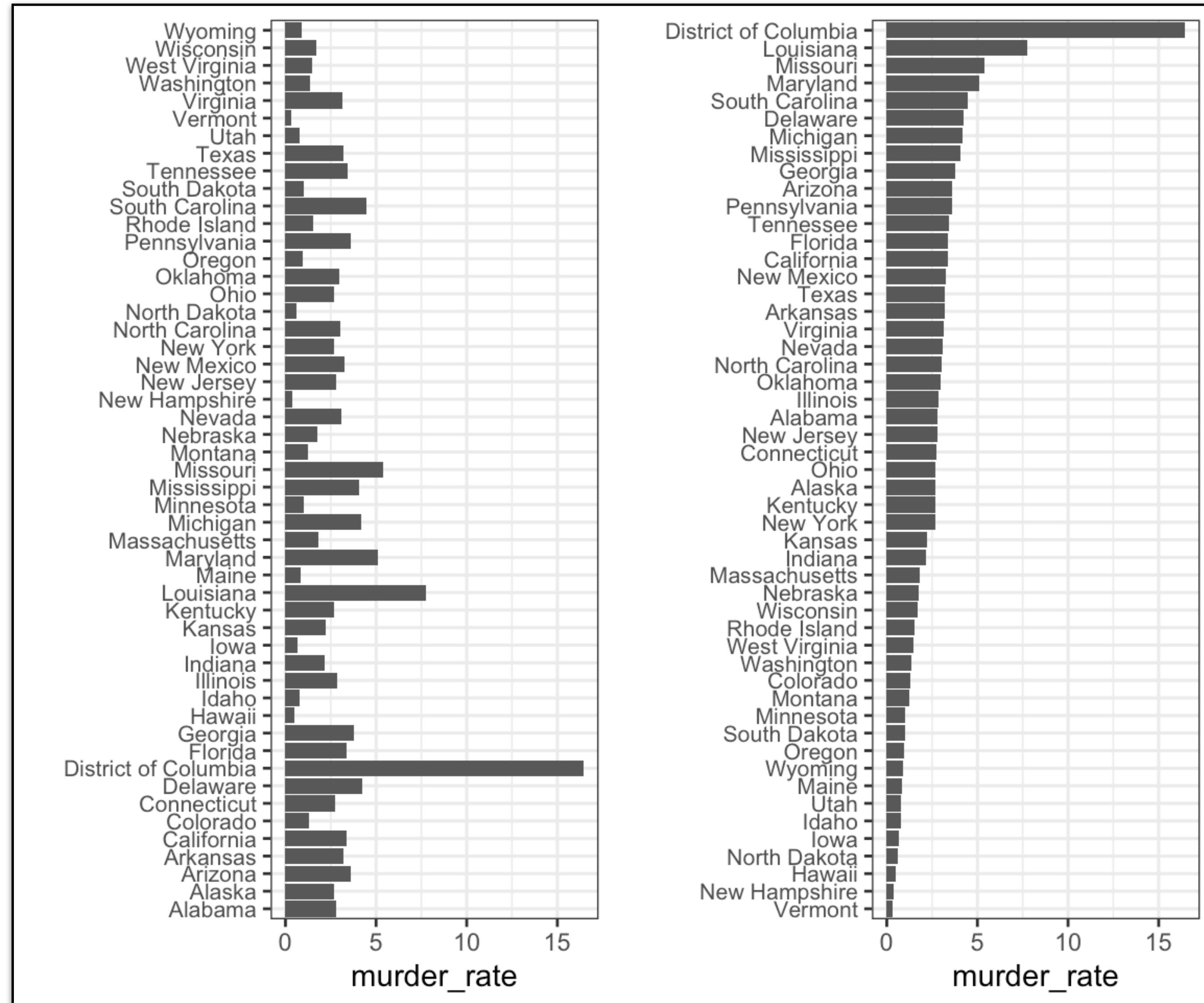


# Do not distort quantities

- Here is a good example from Vox

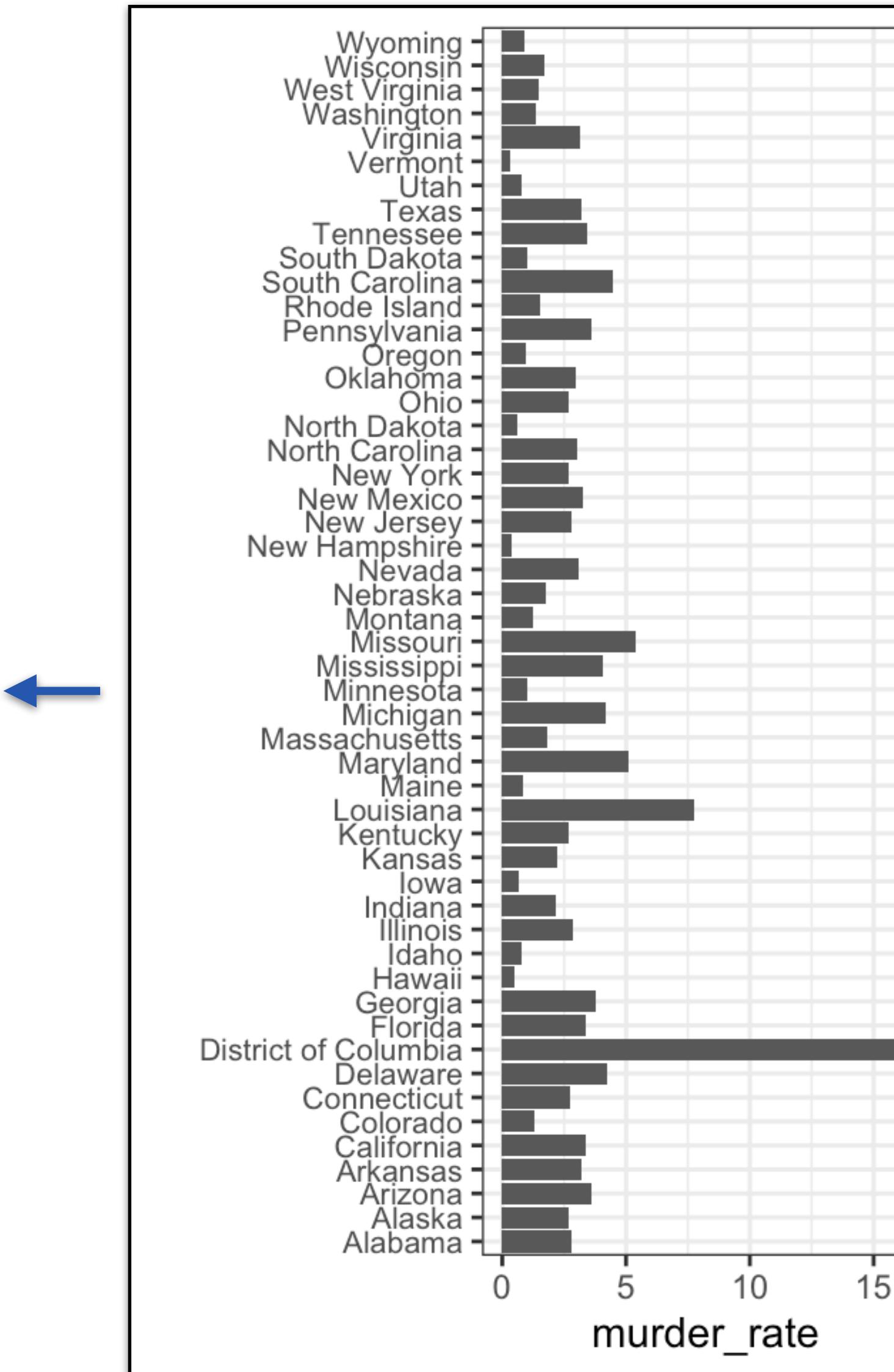


# Order categories by a meaningful value

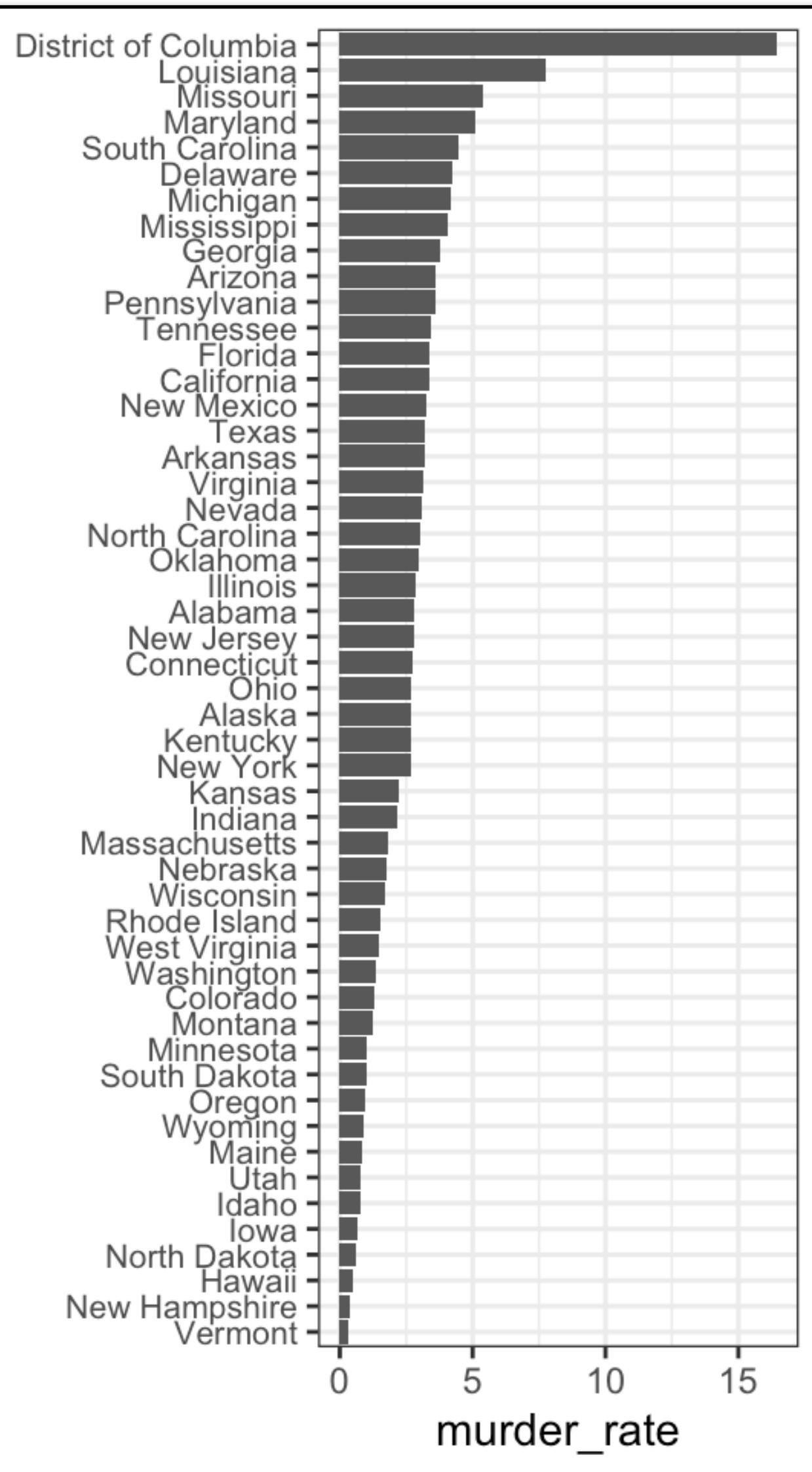


# Order categories by a meaningful value

Ordered alphabetically



Ordered by rate



# Show the data

- We should always strive to show the data
- To motivate this principle, here is an example from the heights dataset

```
library(tidyverse)
library(dslabs)
data("heights")

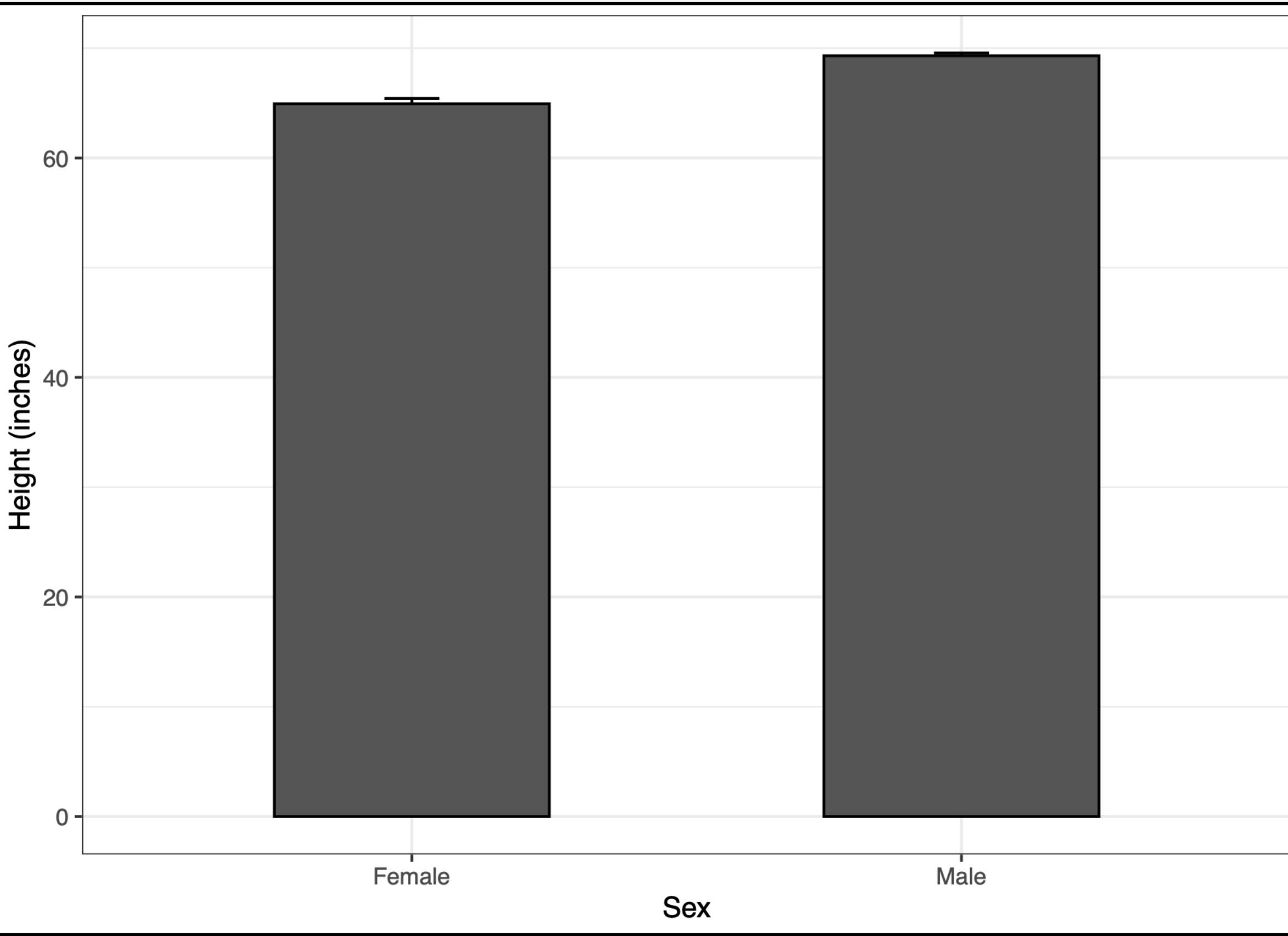
heights %>%
  group_by(sex) %>%
  summarize(a = mean(height),
            s = sd(height)) %>%
  ungroup() %>%
  ggplot(aes(sex, a)) +
  geom_errorbar(aes(ymin=a-s, ymax=a+s), width = 0.10) +
  geom_col(width = 0.50, color="black") +
  ylab("Height (inches)") +
  xlab("Sex") +
  theme_bw()
```

# Show the data

- To motivate this principle, here is an example from the heights dataset

```
library(tidyverse)
library(dslabs)
data("heights")

heights %>%
  group_by(sex) %>%
  summarize(a = mean(height),
            s = sd(height))
ungroup() %>%
  ggplot(aes(sex, a))
  geom_errorbar(aes(ymin = a - s, ymax = a + s),
                 geom_col(width = 0.5)
  ylab("Height (inches")
  xlab("Sex") +
  theme_bw()
```



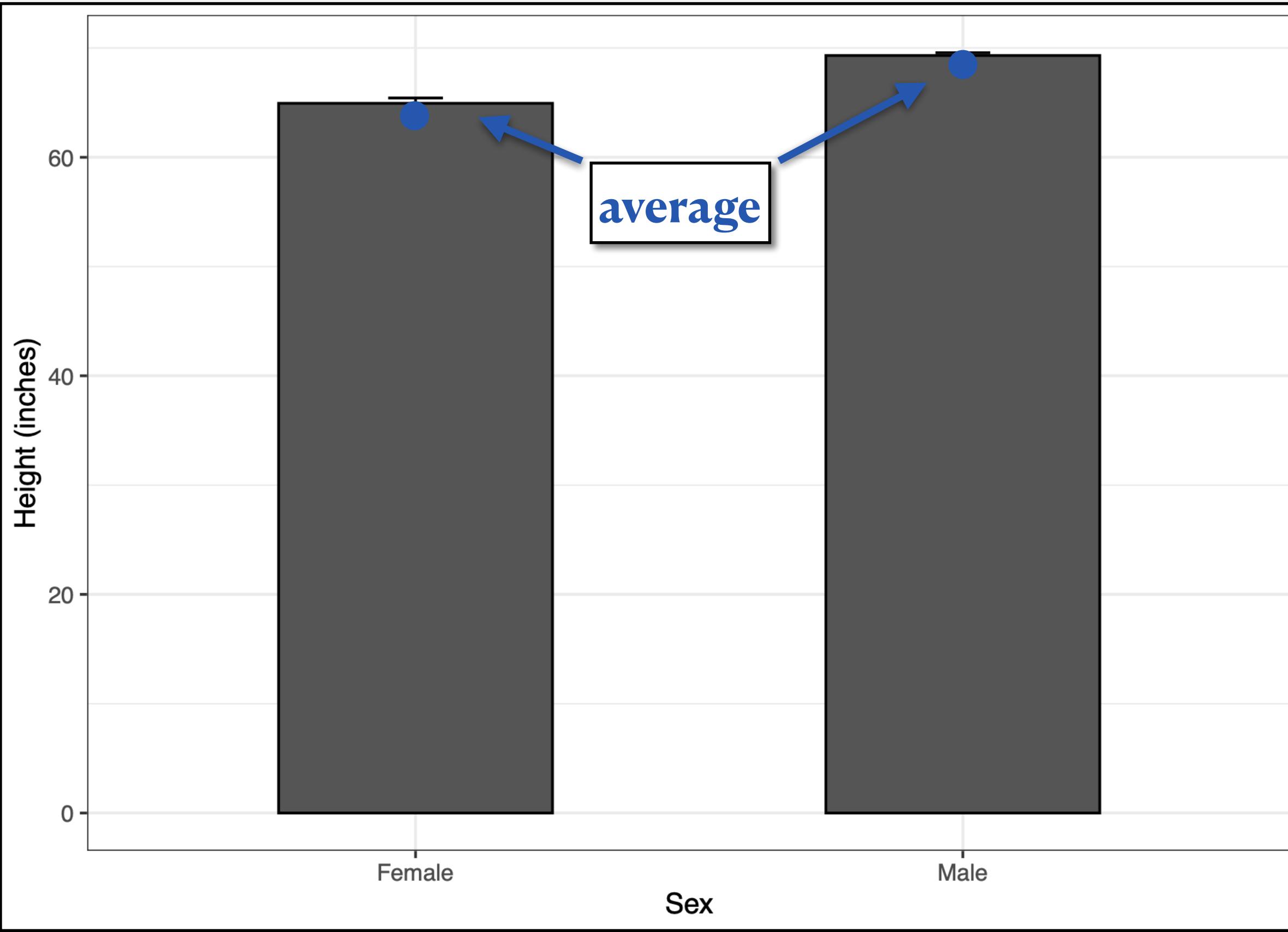
- This is known as a dynamite plot

# Show the data

- To motivate this principle, here is an example from the heights dataset

```
library(tidyverse)
library(dslabs)
data("heights")

heights %>%
  group_by(sex) %>%
  summarize(a = mean(height),
            s = sd(height))
ungroup() %>%
  ggplot(aes(sex, a))
  geom_errorbar(aes(ymin = a - s,
                     ymax = a + s),
                 geom_col(width = 0.5)
  ylab("Height (inches")
  xlab("Sex") +
  theme_bw()
```



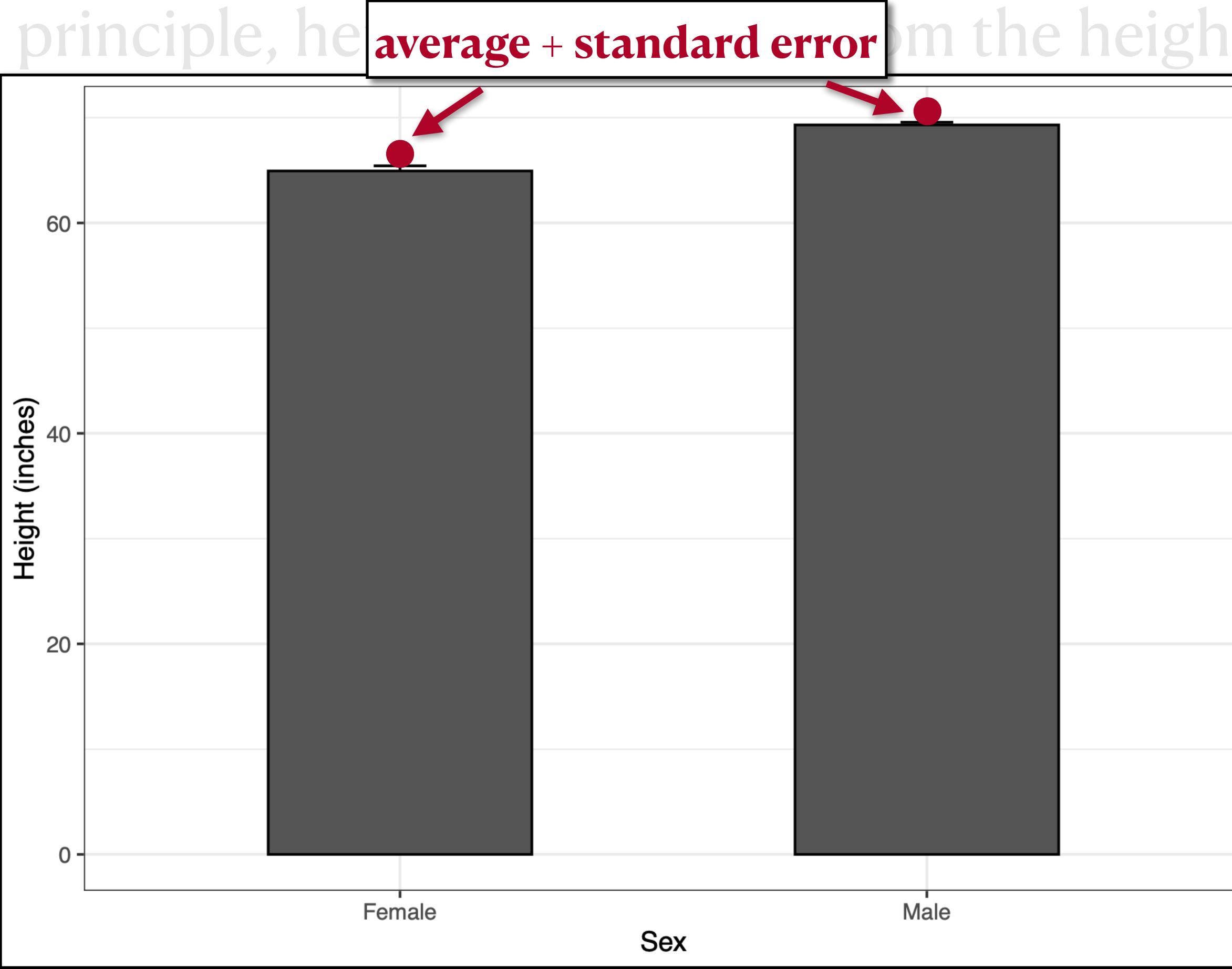
- This is known as a dynamite plot

# Show the data

- To motivate this principle, here's a plot from the heights dataset

```
library(tidyverse)
library(dslabs)
data("heights")

heights %>%
  group_by(sex) %>%
  summarize(a = mean(height),
            s = sd(height))
ungroup() %>%
  ggplot(aes(sex, a))
  geom_errorbar(aes(ymin = a - s,
                     ymax = a + s),
                 geom_col(width = 0.5))
  ylab("Height (inches")
  xlab("Sex")
  theme_bw()
```



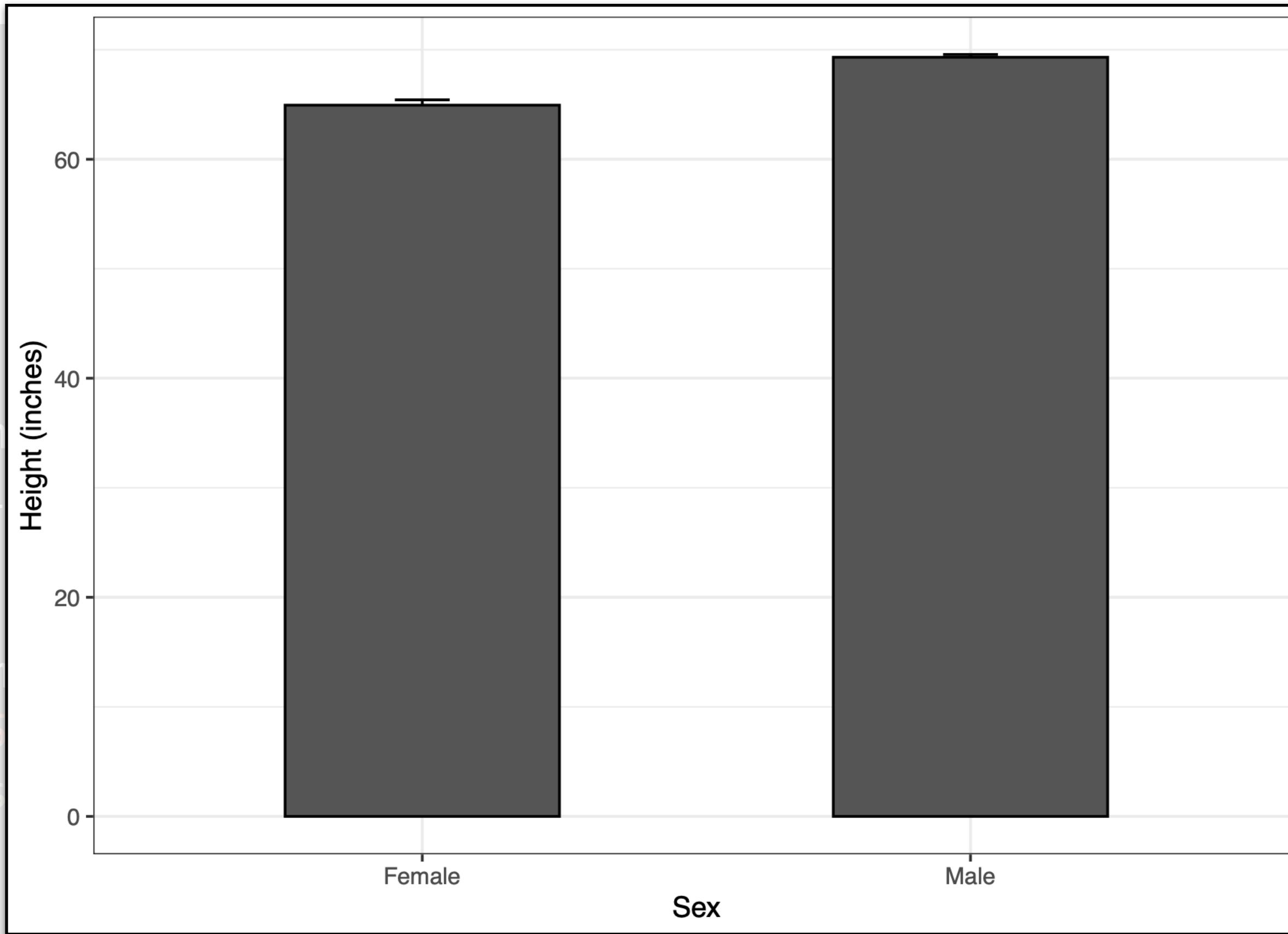
- This is known as a dynamite plot

# Show the data

- To motivate this principle, here is an example from the heights dataset

```
library(tidyverse)
library(dslabs)
data("heights")

heights %>%
  group_by(sex) %>%
  summarize(a = mean(height),
            s = sd(height))
ungroup() %>%
  ggplot(aes(sex, a))
  geom_errorbar(aes(ymin = a - s,
                     ymax = a + s),
                 geom_col(width = 0.5))
  ylab("Height (inches)")
  xlab("Sex") +
  theme_bw()
```



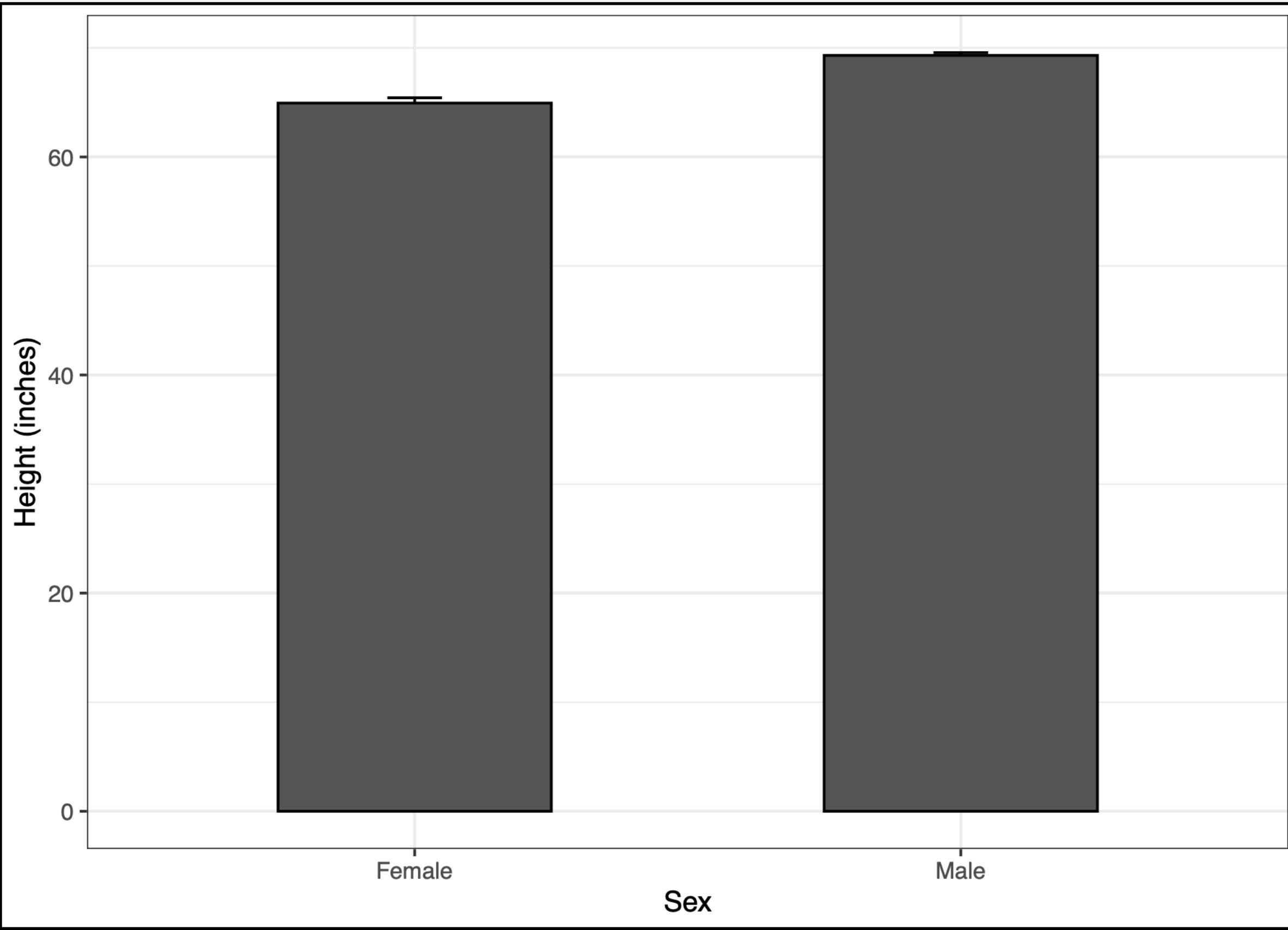
- This figure says very little about the data.
- All we can see is the mean for each group <sup>29</sup> and the corresponding standard errors

# Show the data

- To motivate this principle, here is an example from the heights dataset

```
library(tidyverse)
library(dslabs)
data("heights")

heights %>%
  group_by(sex) %>%
  summarize(a = mean(height),
            s = sd(height))
ungroup() %>%
  ggplot(aes(sex, a)) +
  geom_errorbar(aes(ymin = a - s, ymax = a + s),
                width = 0.5) +
  ylab("Height (inches)") +
  xlab("Sex") +
  theme_bw()
```



- Let's go back to our principle, show the data!

# Show the data

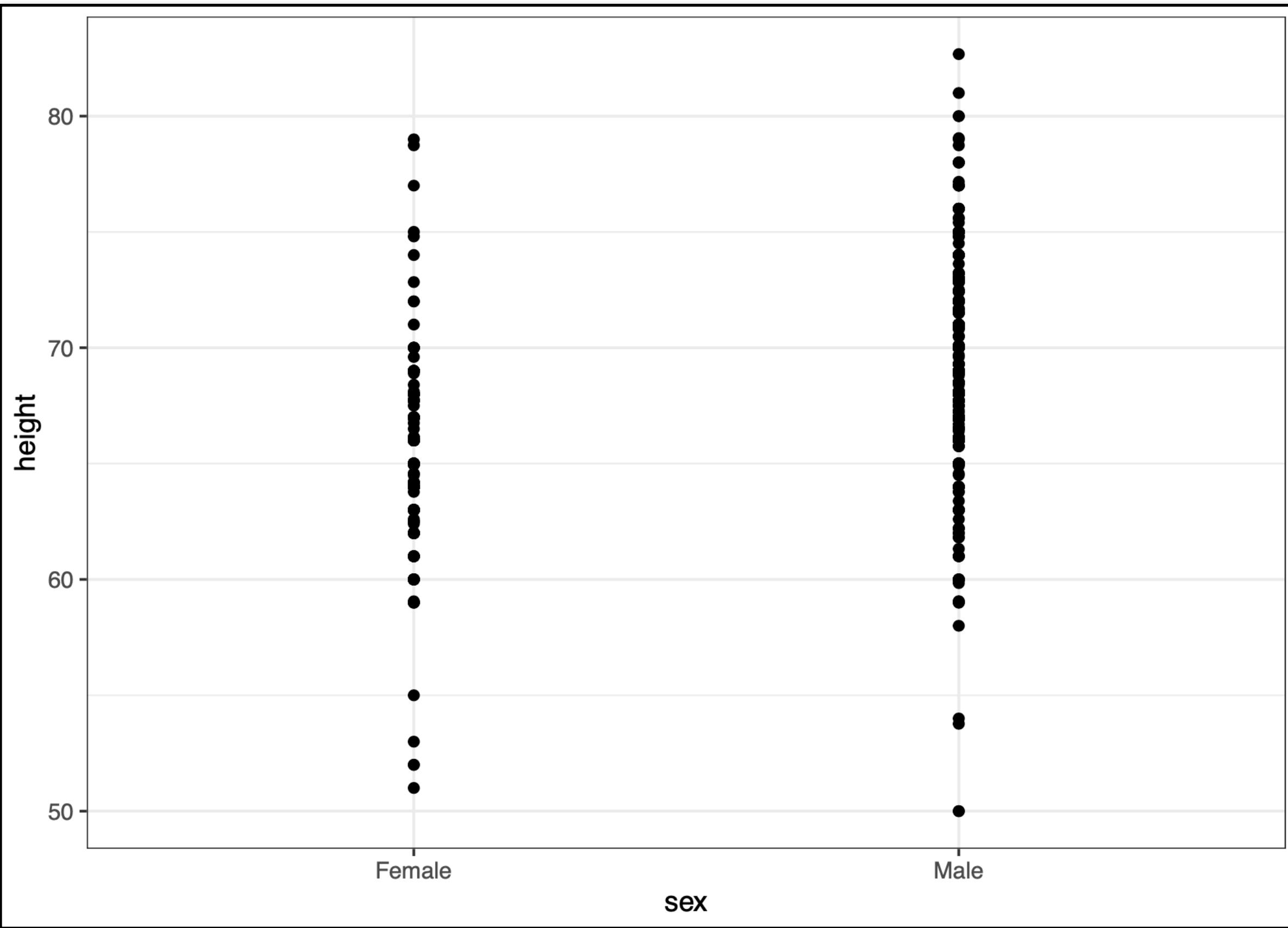
- This simple code generates a more informative figure:

```
heights %>%  
  ggplot(aes(sex, height)) +  
  geom_point() +  
  theme_bw()
```

# Show the data

- This simple code generates a more informative figure:

```
heights %>%  
  ggplot(aes(sex, height)) +  
  geom_point() +  
  theme_bw()
```



- From this we can see the range of the data
- Let's improve on this

# Show the data

- This simple code generates a more informative figure:

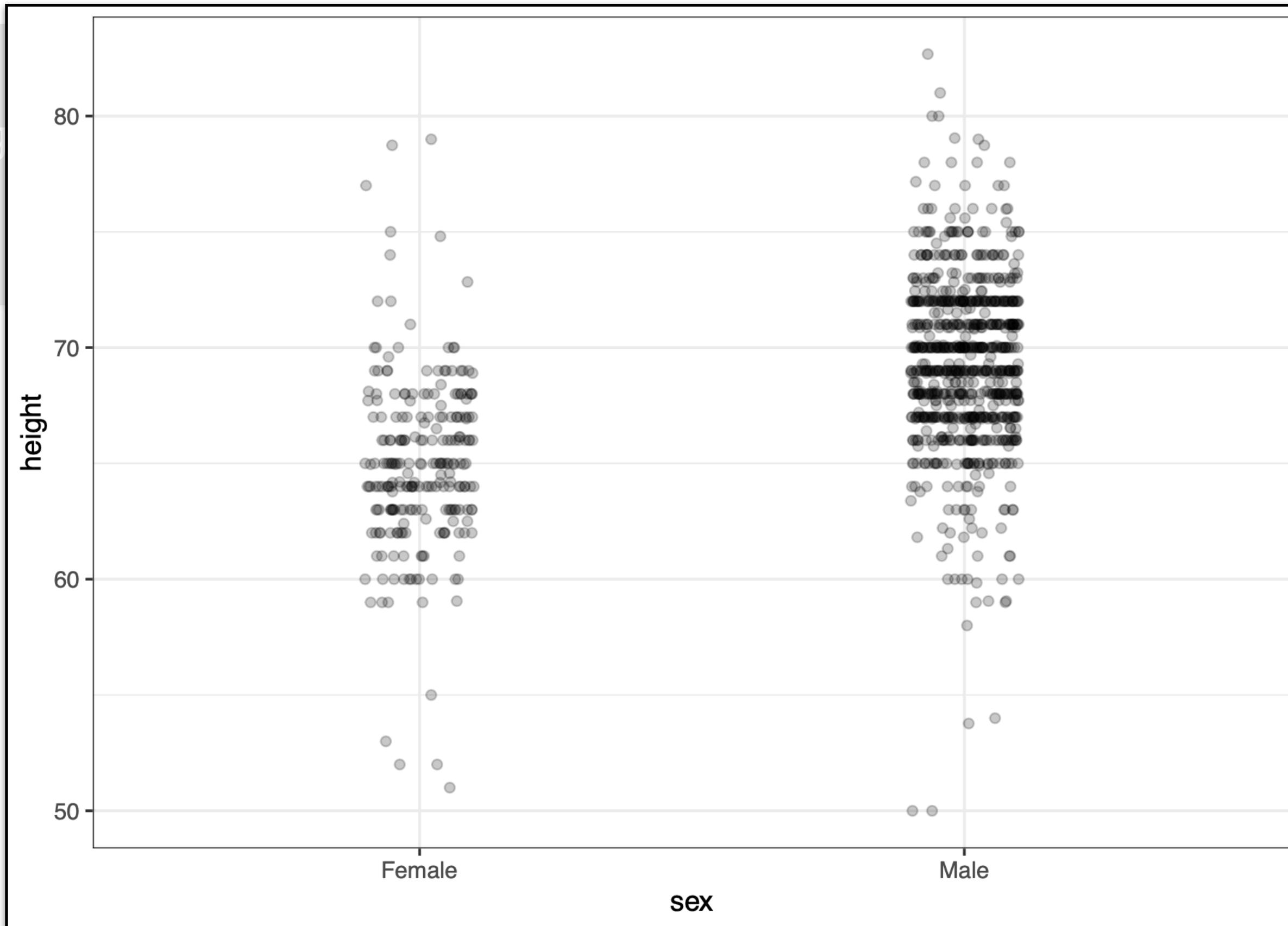
```
heights %>%  
  ggplot(aes(sex, height)) +  
  geom_jitter(width = 0.1, alpha = 0.2) +  
  theme_bw()
```

- Here we add a *jitter* – a small random shift to each point

# Show the data

- This simple code generates a more informative figure:

```
heights %>%  
  ggplot(aes(sex, height)) +  
  geom_jitter(width = 0.5) +  
  theme_bw()
```

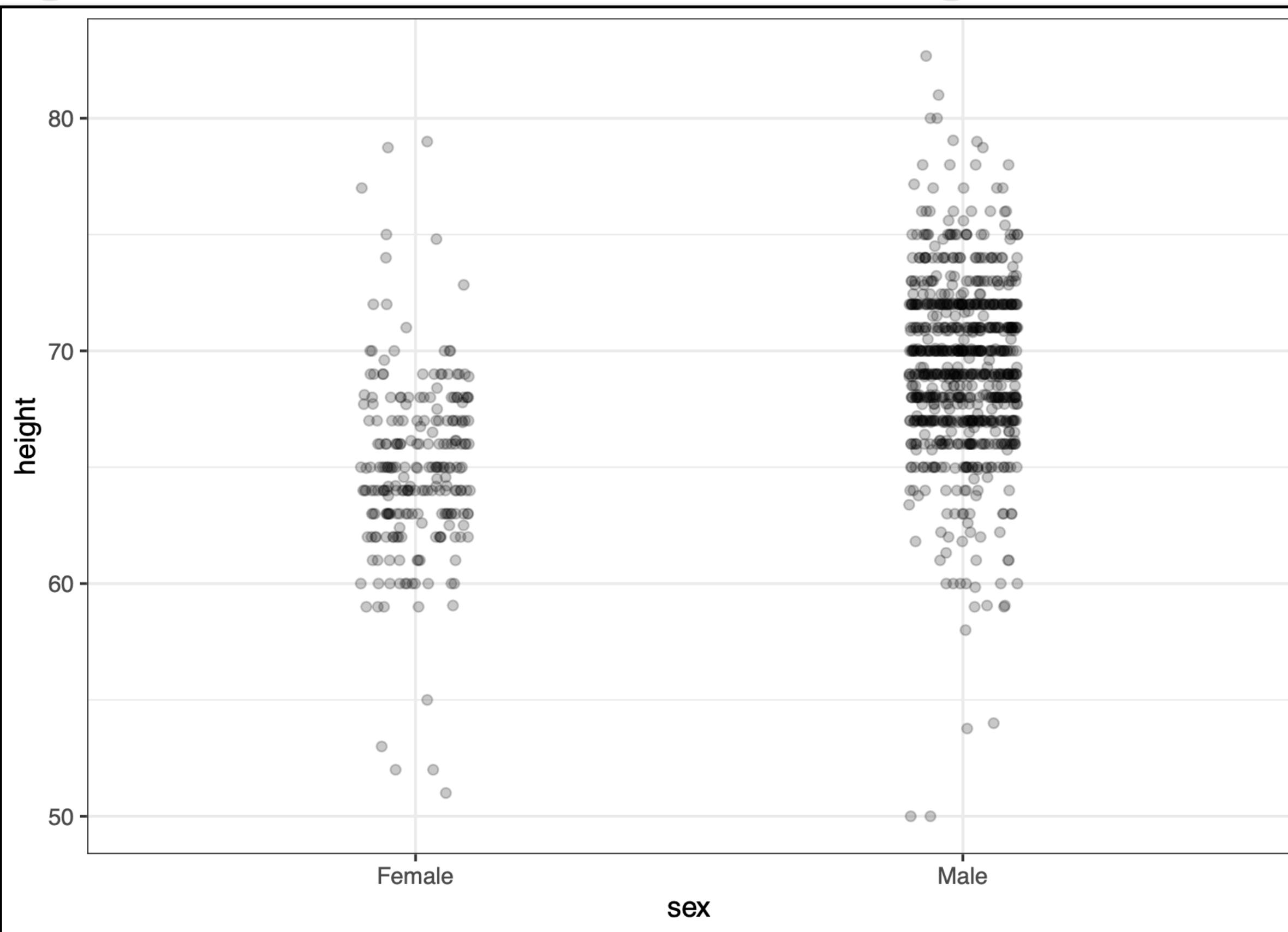


- Note that we add a horizontal *jitter*, which does not affect the interpretation

# Show the data

- This simple code generates a more informative figure:

```
heights %>%  
  ggplot(aes(sex, height)) +  
  geom_jitter(width = 0.5) +  
  theme_bw()
```



- Further, the alpha-blend gives us a better understanding of the distributions

# Use common axes

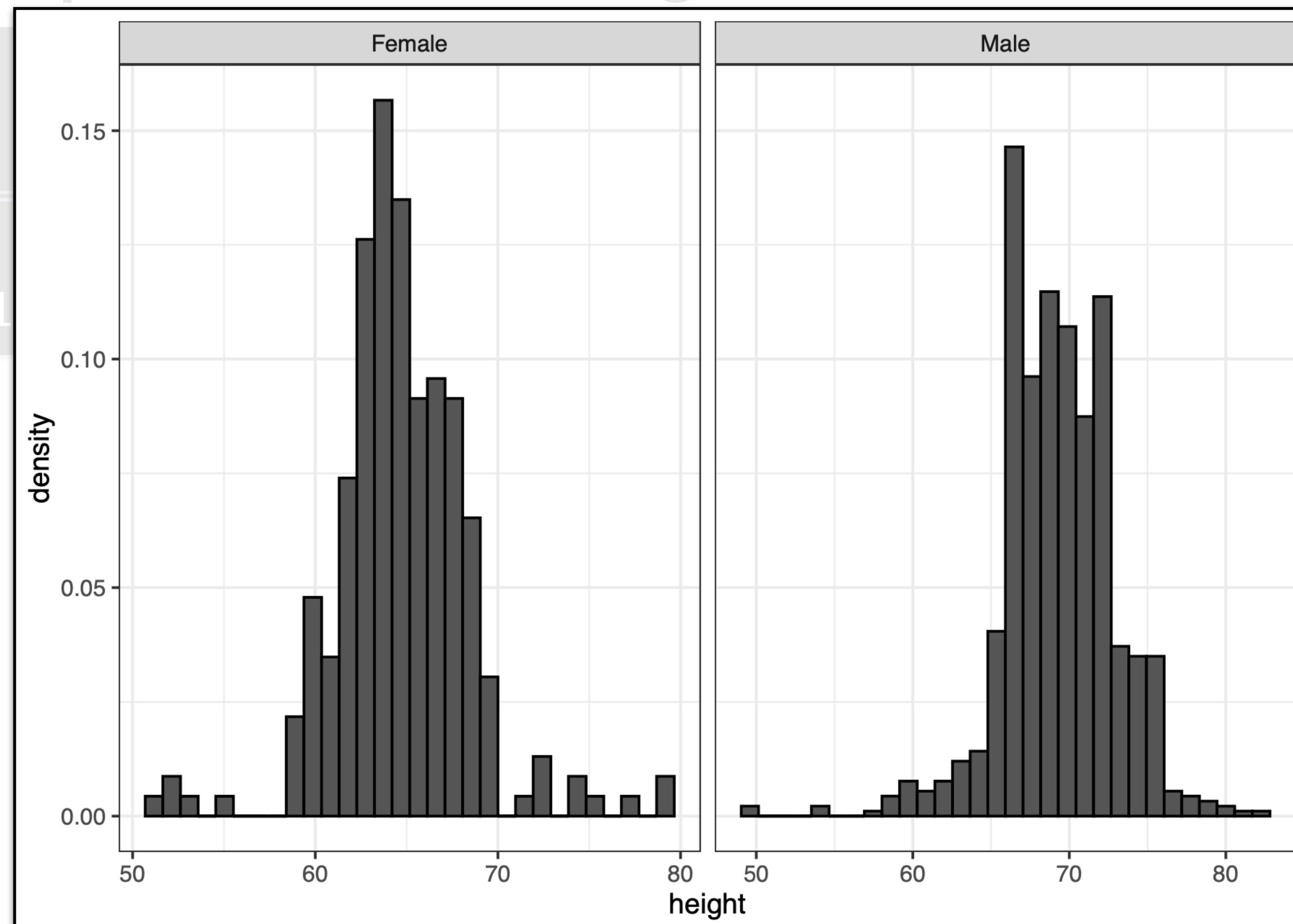
- There are a lot of points, let's use histograms instead:

```
heights %>%
  ggplot(aes(x=height, y=..density..)) +
  geom_histogram(color="black") +
  theme_bw() +
  facet_wrap(~sex, scales = "free_x")
```

# Use common axes

- There are a lot of points, let's use histograms instead:

```
heights %>%
  ggplot(aes(x=height,
  geom_histogram(color=
  theme_bw() +
  facet_wrap(~sex, scale="free_x")
```

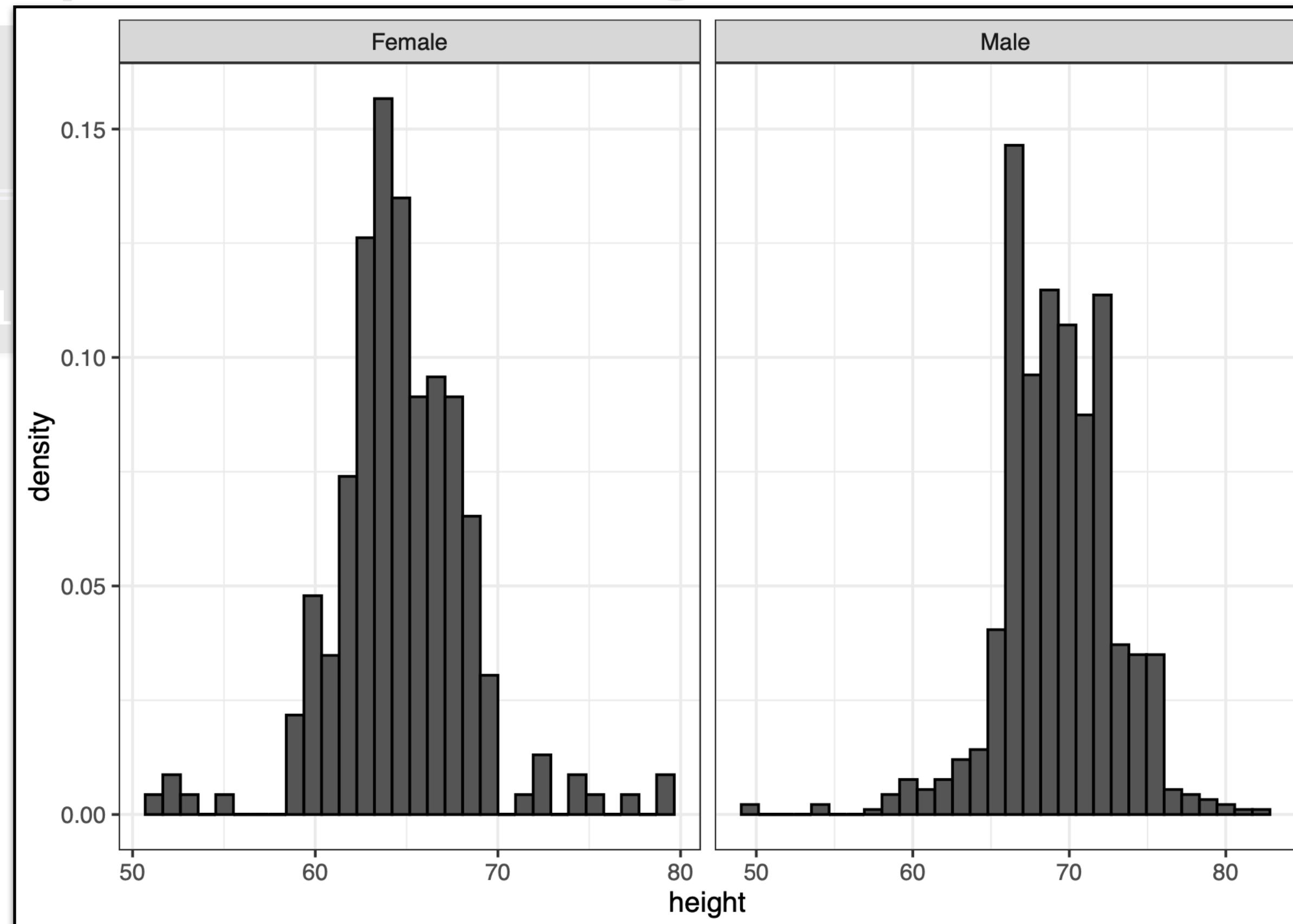


- It's not immediately clear that males, on average, are taller than females
- The *x-axis* is not the same for the two groups<sup>37</sup>

# Use common axes

- There are a lot of points, let's use histograms instead:

```
heights %>%
  ggplot(aes(x=height,
  geom_histogram(color=
  theme_bw() +
  facet_wrap(~sex, scale="free_x")
```



- Use common x axes!

# Use common axes

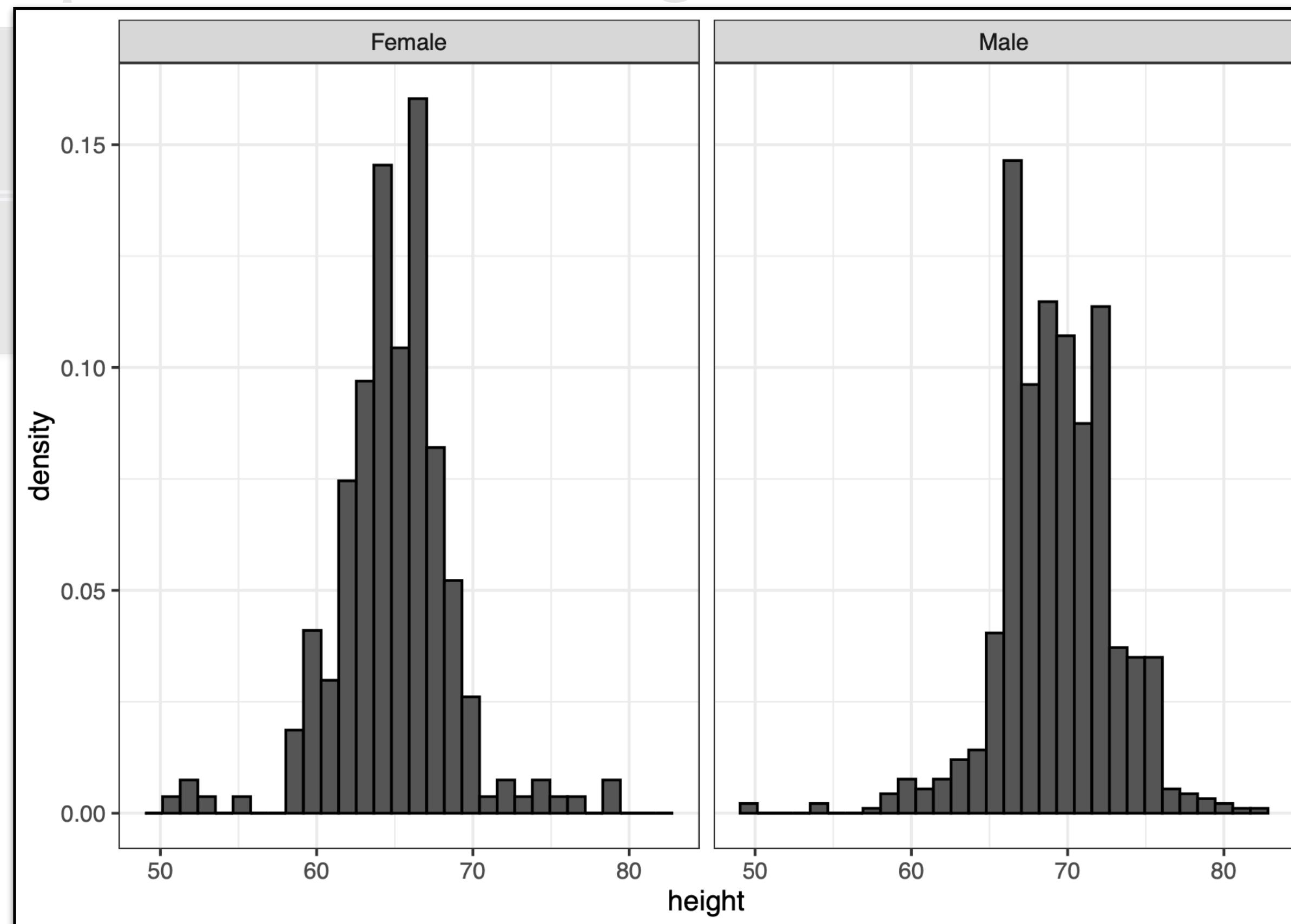
- The default in `facet_wrap` is to have common axes

```
heights %>%
  ggplot(aes(x=height, y=..density..)) +
  geom_histogram(color="black") +
  theme_bw() +
  facet_wrap(~sex)
```

# Use common axes

- There are a lot of points, let's use histograms instead:

```
heights %>%
  ggplot(aes(x=height,
  geom_histogram(color=
  theme_bw() +
  facet_wrap(~sex)
```



- This is better

# Use common axes

- Another principle:
  - Align plots vertically to see horizontal changes and horizontally to see vertical changes

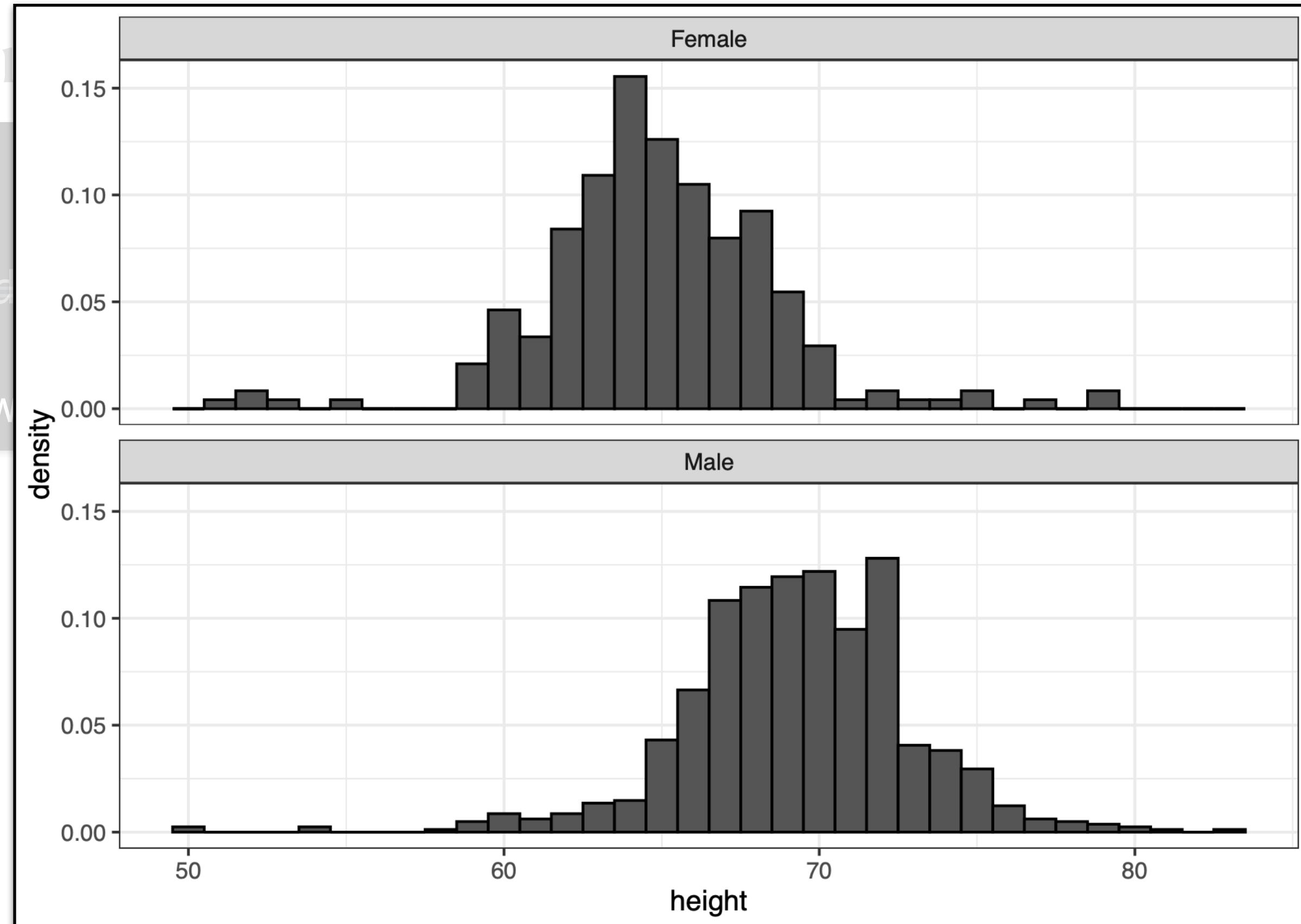
```
heights %>%
  ggplot(aes(x=height, y=..density..)) +
  geom_histogram(binwidth=1,color="black") +
  theme_bw() +
  facet_wrap(~sex, nrow=2)
```

# Use common axes

- Another principle:

- Align plots vertically

```
heights %>%  
  ggplot(aes(x=height,  
             geom_histogram(bins=15)) +  
  theme_bw() +  
  facet_wrap(~sex, nrow=2)
```



- This makes it easier to see the difference

# Use common axes

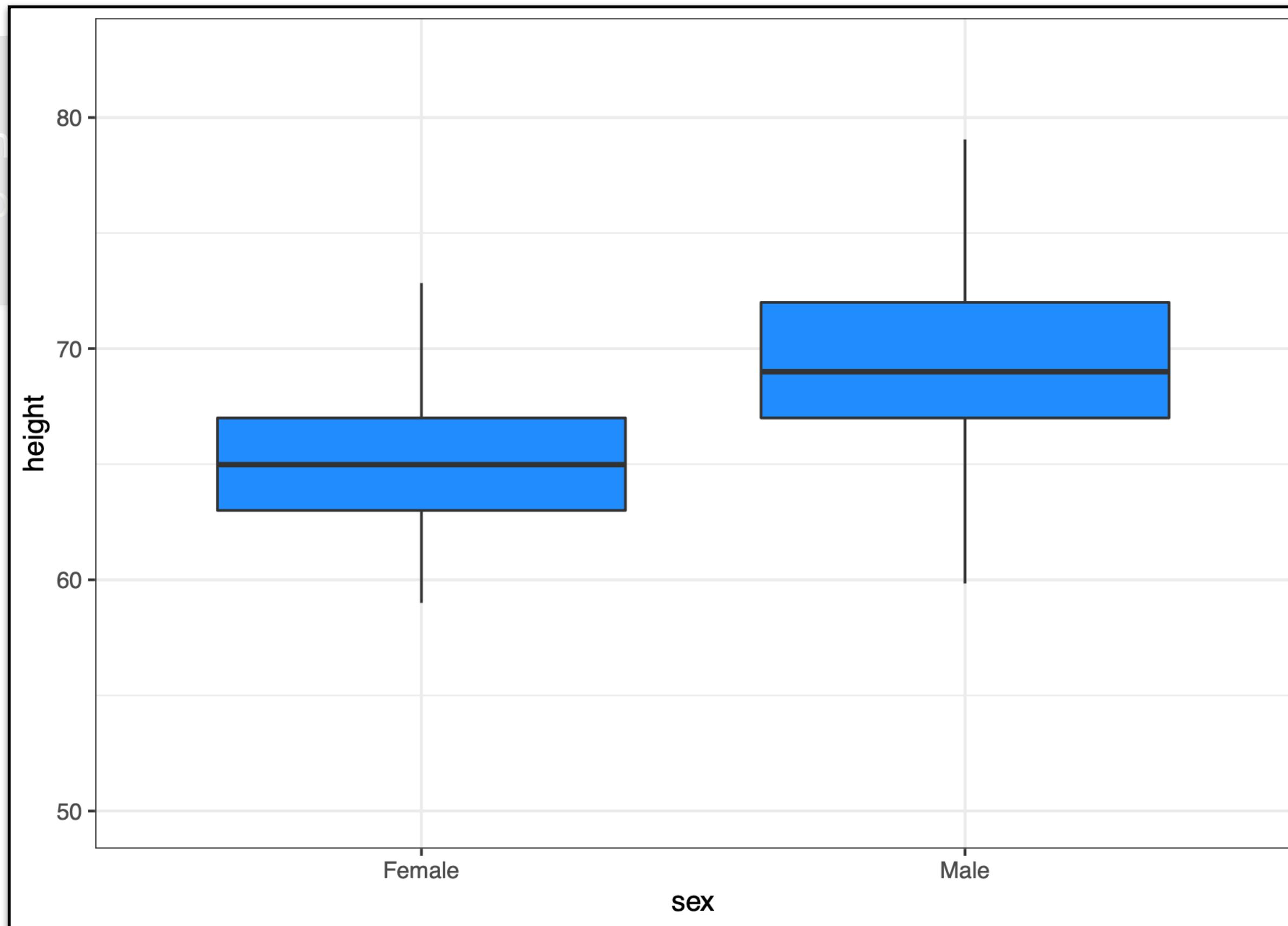
- A boxplot would also be informative here

```
heights %>%  
  ggplot(aes(x=sex, y=height)) +  
  geom_boxplot(fill="dodgerblue", outlier.alpha = 0) +  
  theme_bw()
```

# Use common axes

- A boxplot would also be informative here

```
heights %>%  
  ggplot(aes(x=sex, y=height)) +  
  geom_boxplot(fill="steelblue") +  
  theme_bw()
```



- Show the data!

# Use common axes

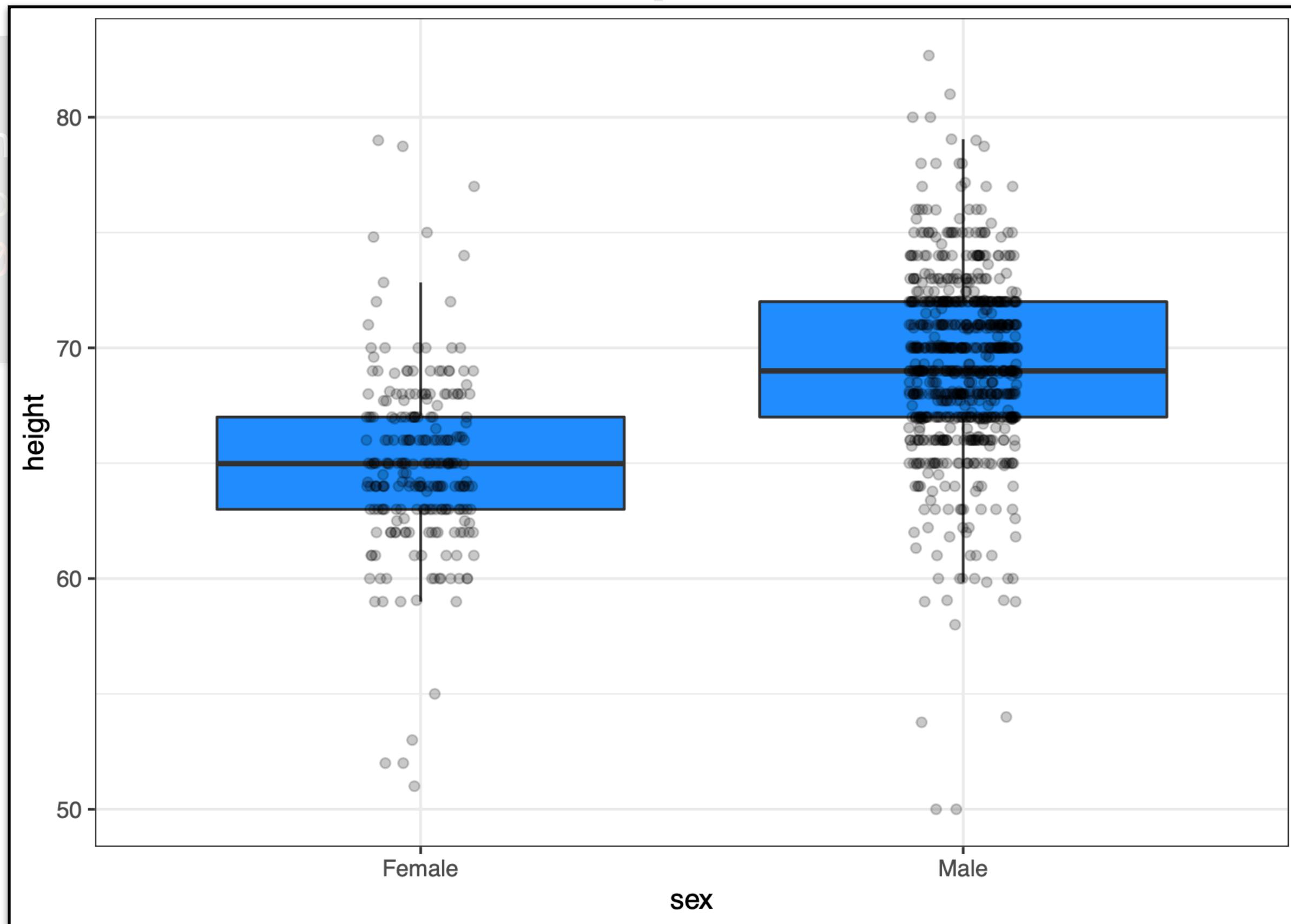
- We use `geom_jitter` to add the data points

```
heights %>%  
  ggplot(aes(x=sex, y=height)) +  
  geom_boxplot(fill="dodgerblue", outlier.alpha = 0) +  
  geom_jitter(width = 0.1, alpha = 0.2) +  
  theme_bw()
```

# Use common axes

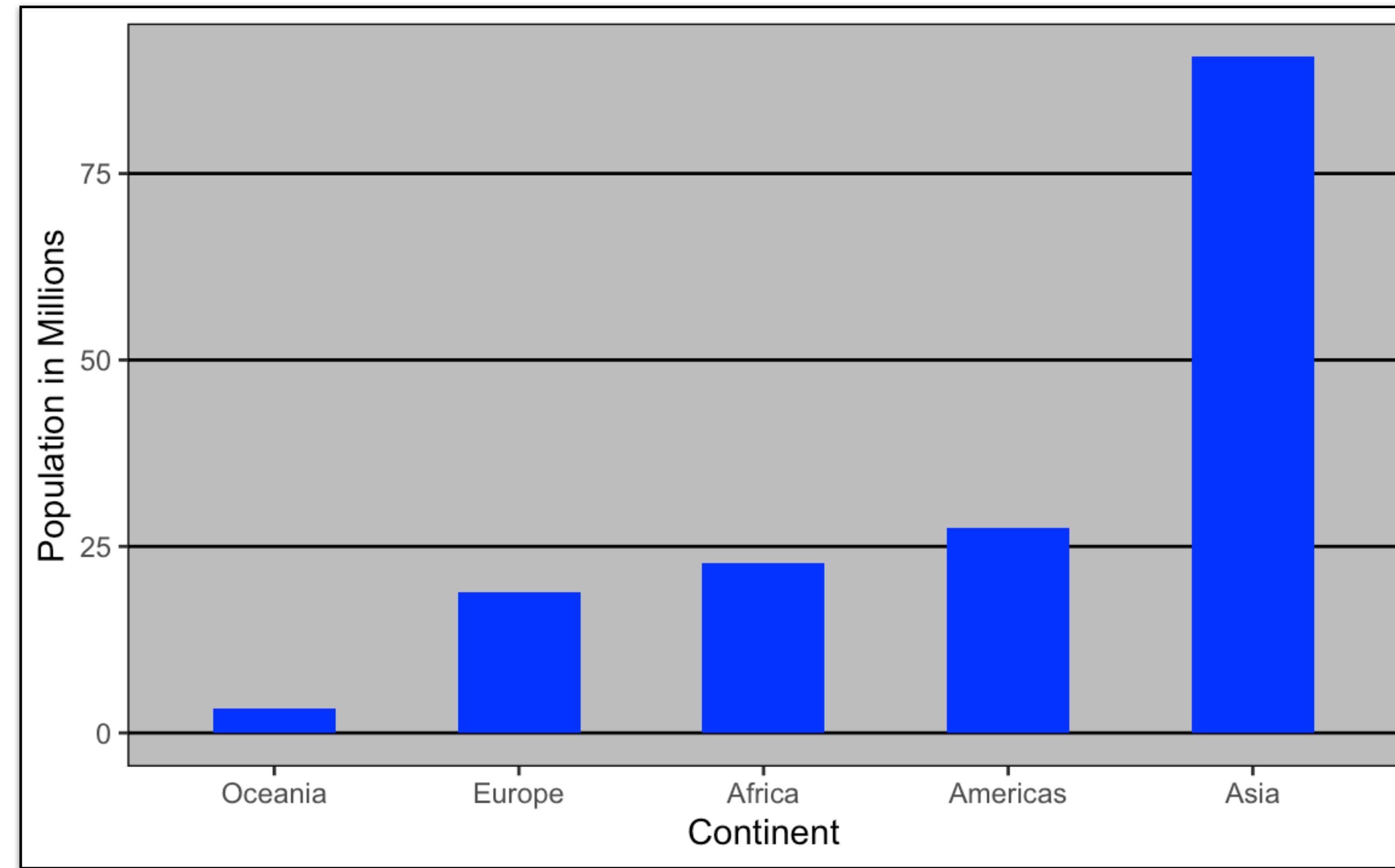
- We use `geom_jitter` to add the data points

```
heights %>%  
  ggplot(aes(x=sex, y=height)) +  
  geom_boxplot(fill="darkblue") +  
  geom_jitter(width = 0.5) +  
  theme_bw()
```



# Consider transformations

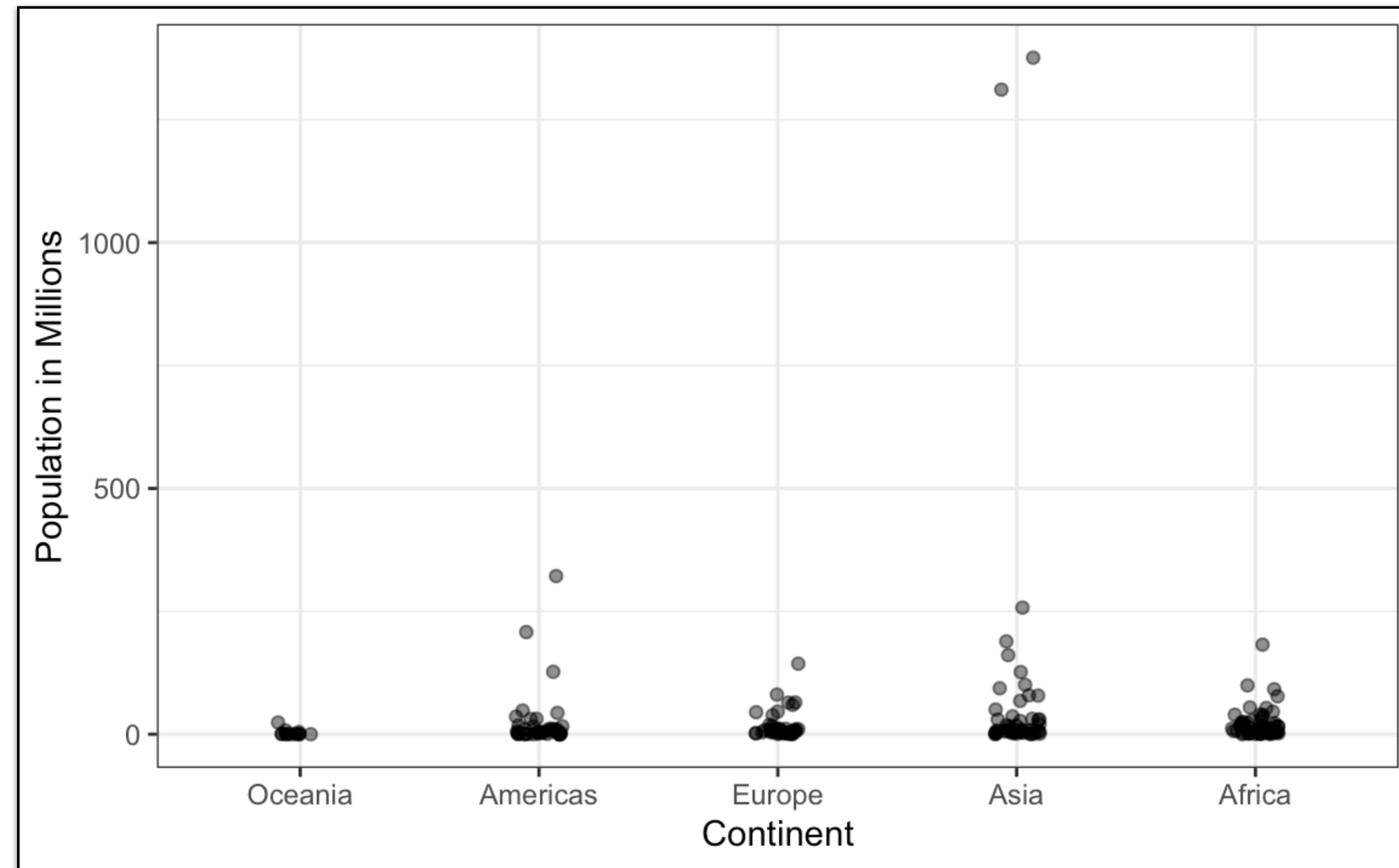
- Consider the following barplot of average population for each continent in 2015



- Any comments?

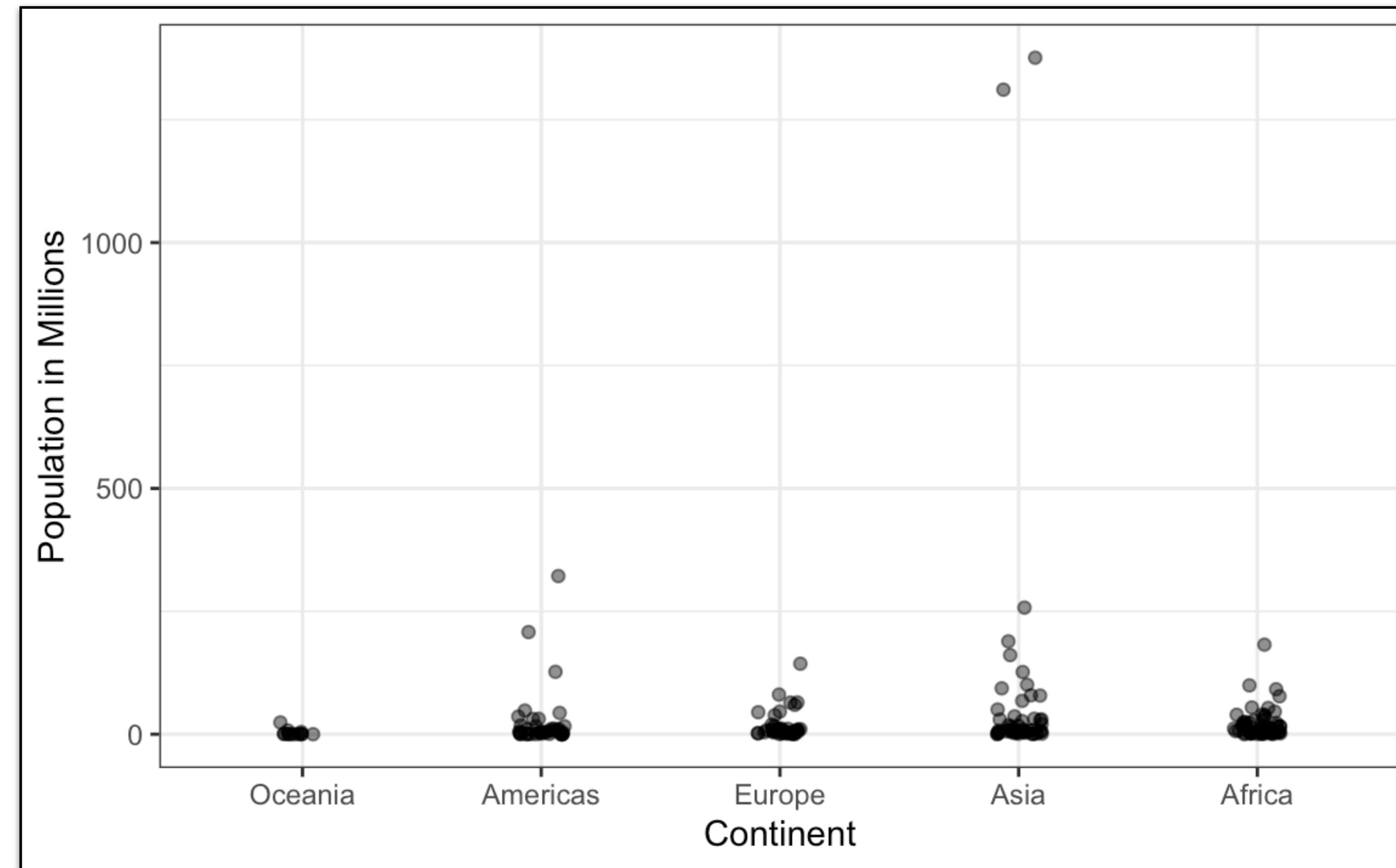
# Consider transformations

- Consider the following barplot of average population for each continent in 2015



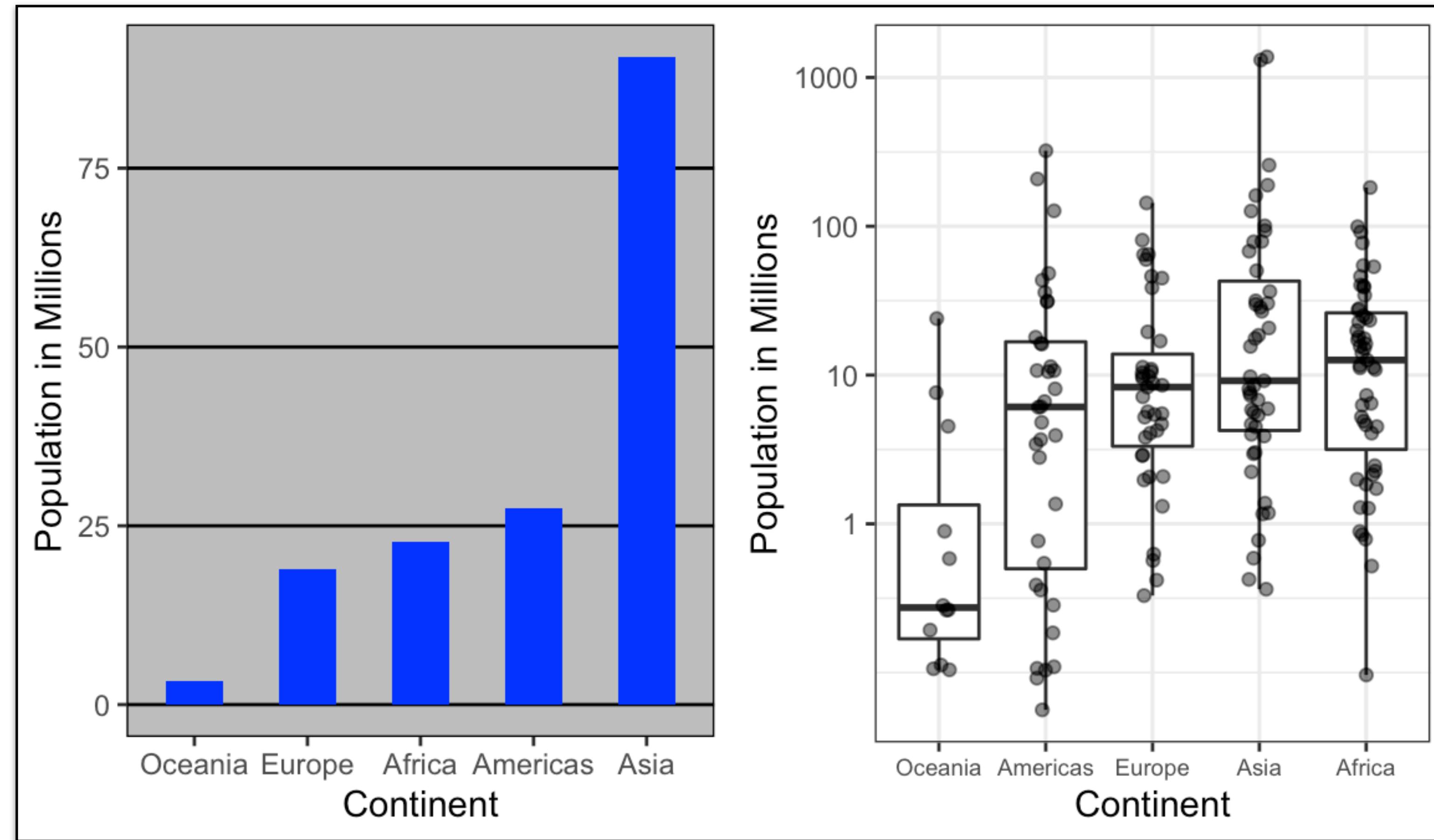
# Consider transformations

- Consider the following barplot of average population for each continent in 2015



- Any ideas on what the two outliers are?<sup>49</sup>

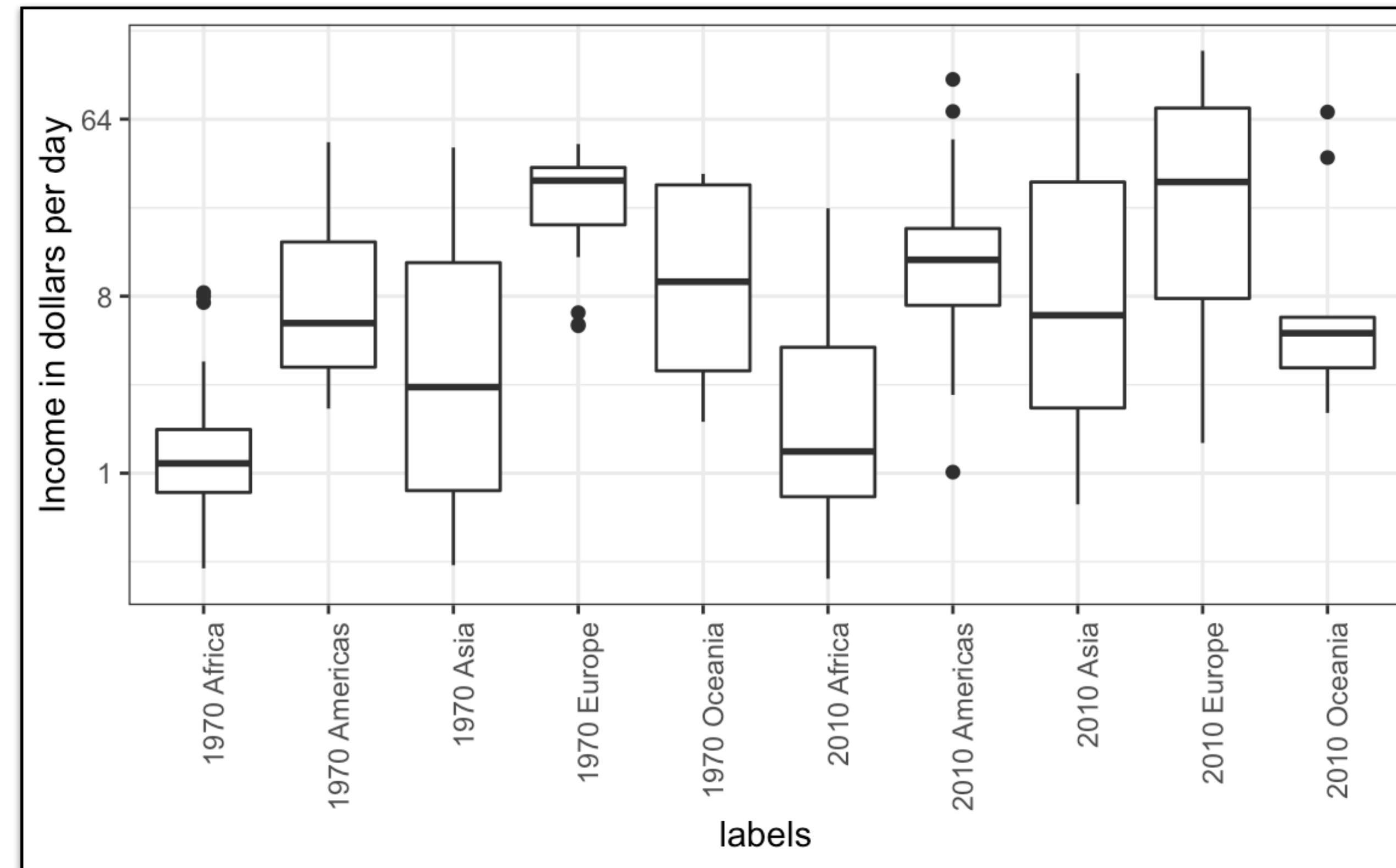
# Consider transformations



- $\log_{10}$  transformation of the *y-axis*
- It turns out that Africa has a higher median population than Asia

# Use color to codify a third variable

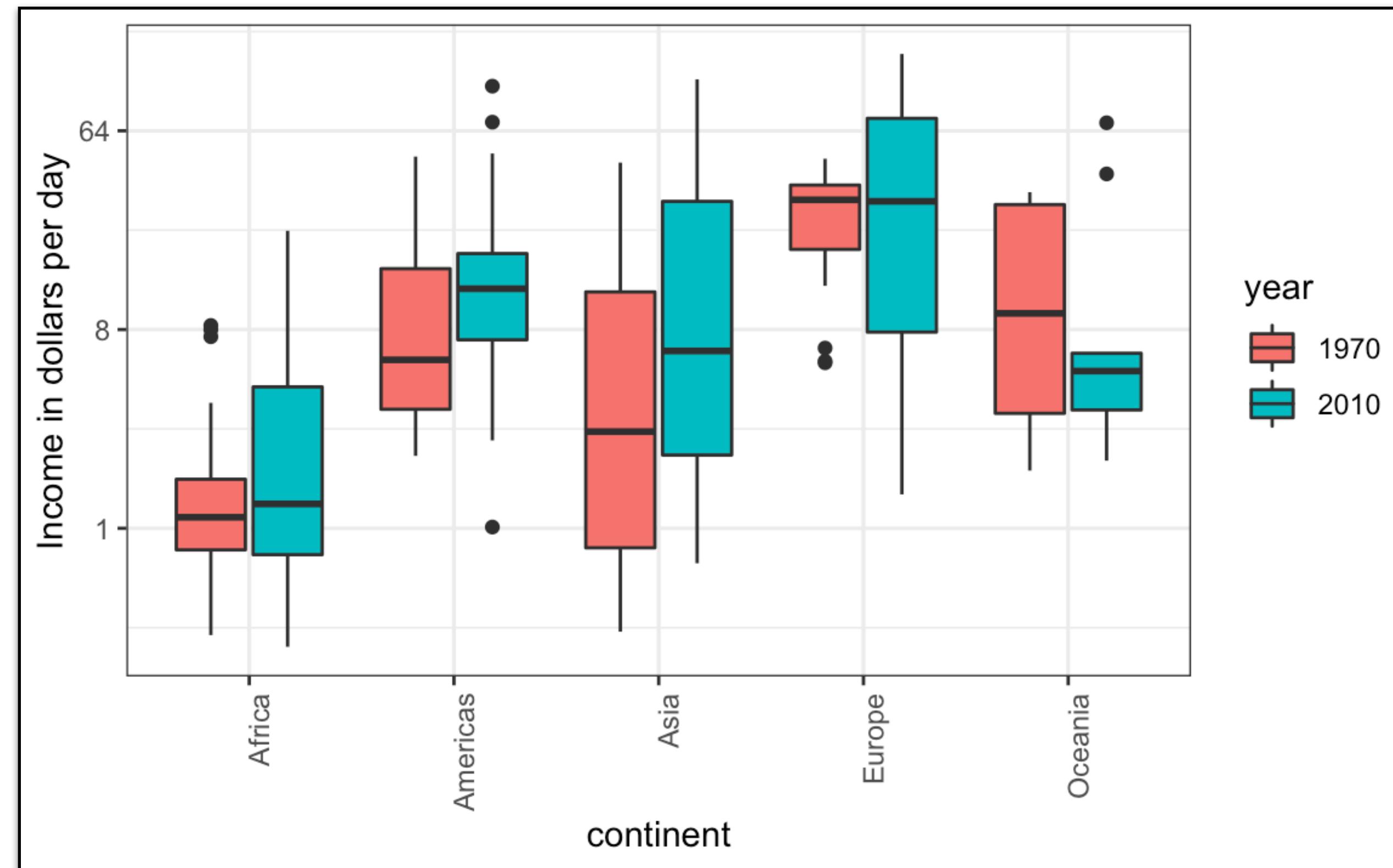
- This figure shows income per continent for 1970 and 2010:



- It's hard to compare, for example, Africa in 1970 to Africa in 2010 because the corresponding boxplots are visually far apart.

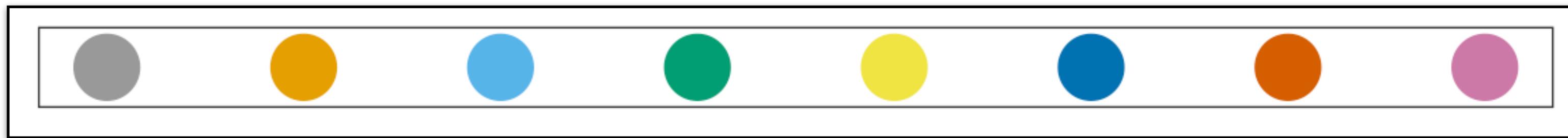
# Use color to codify a third variable

- By coding year with color, its easier to make the comparison



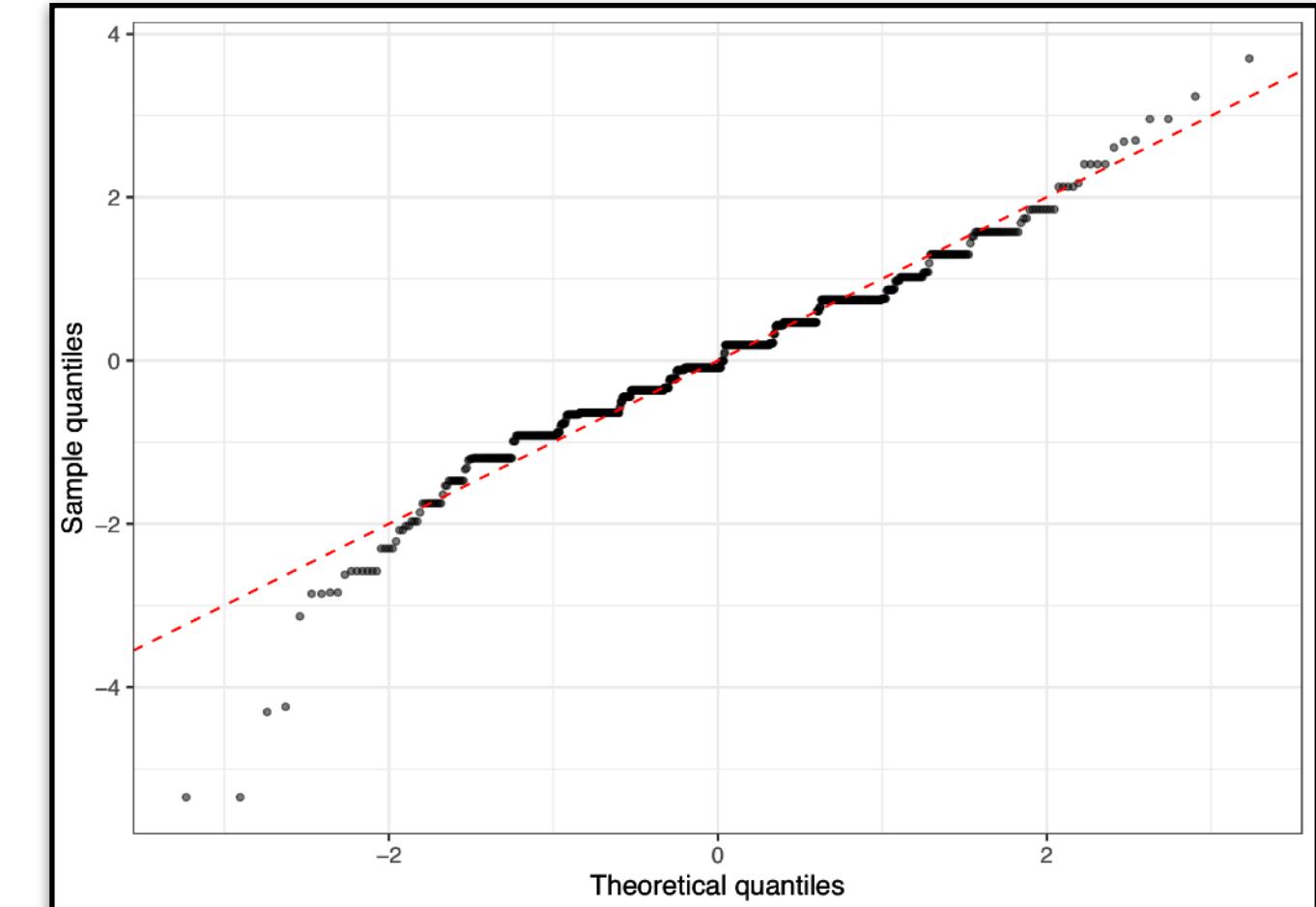
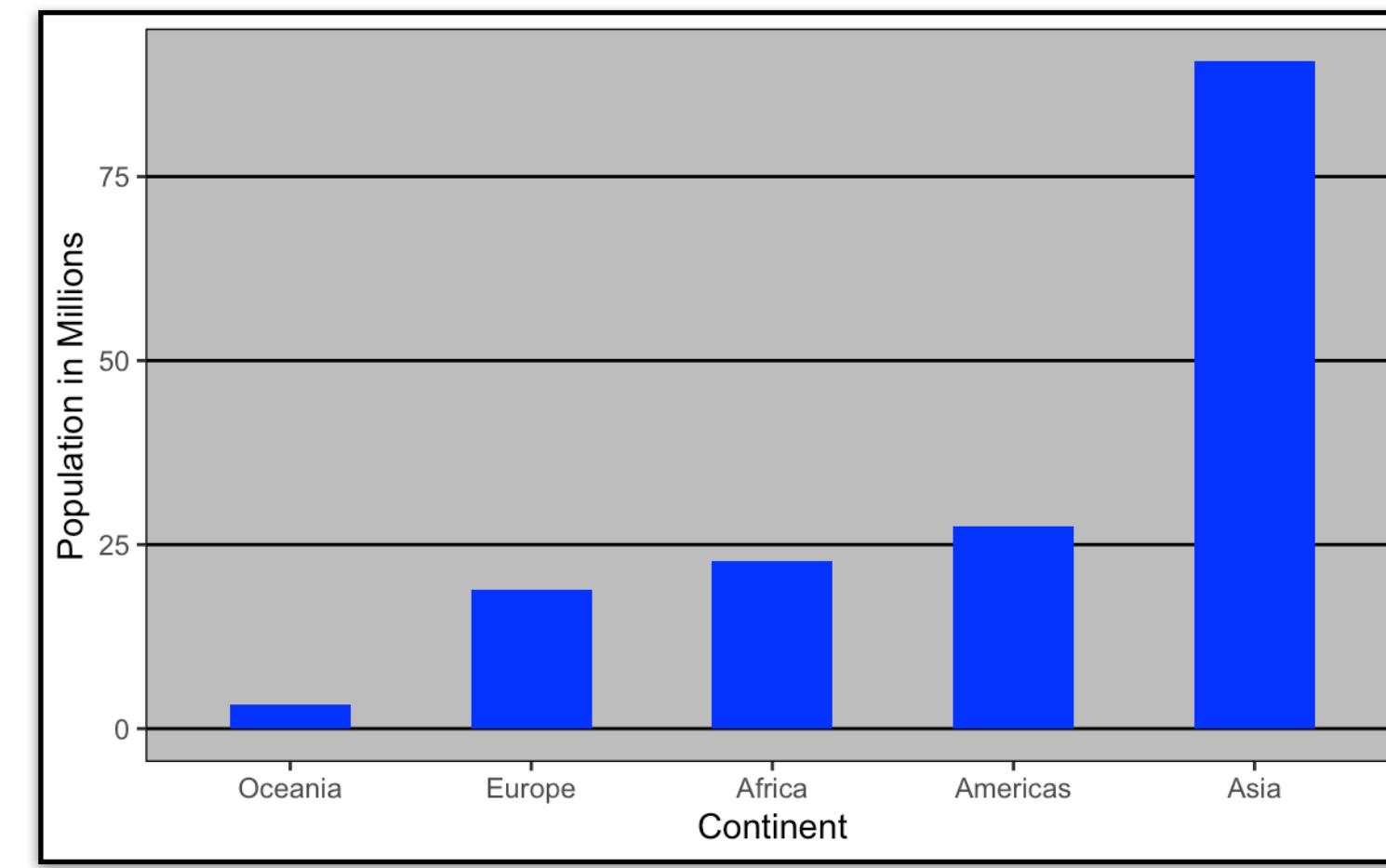
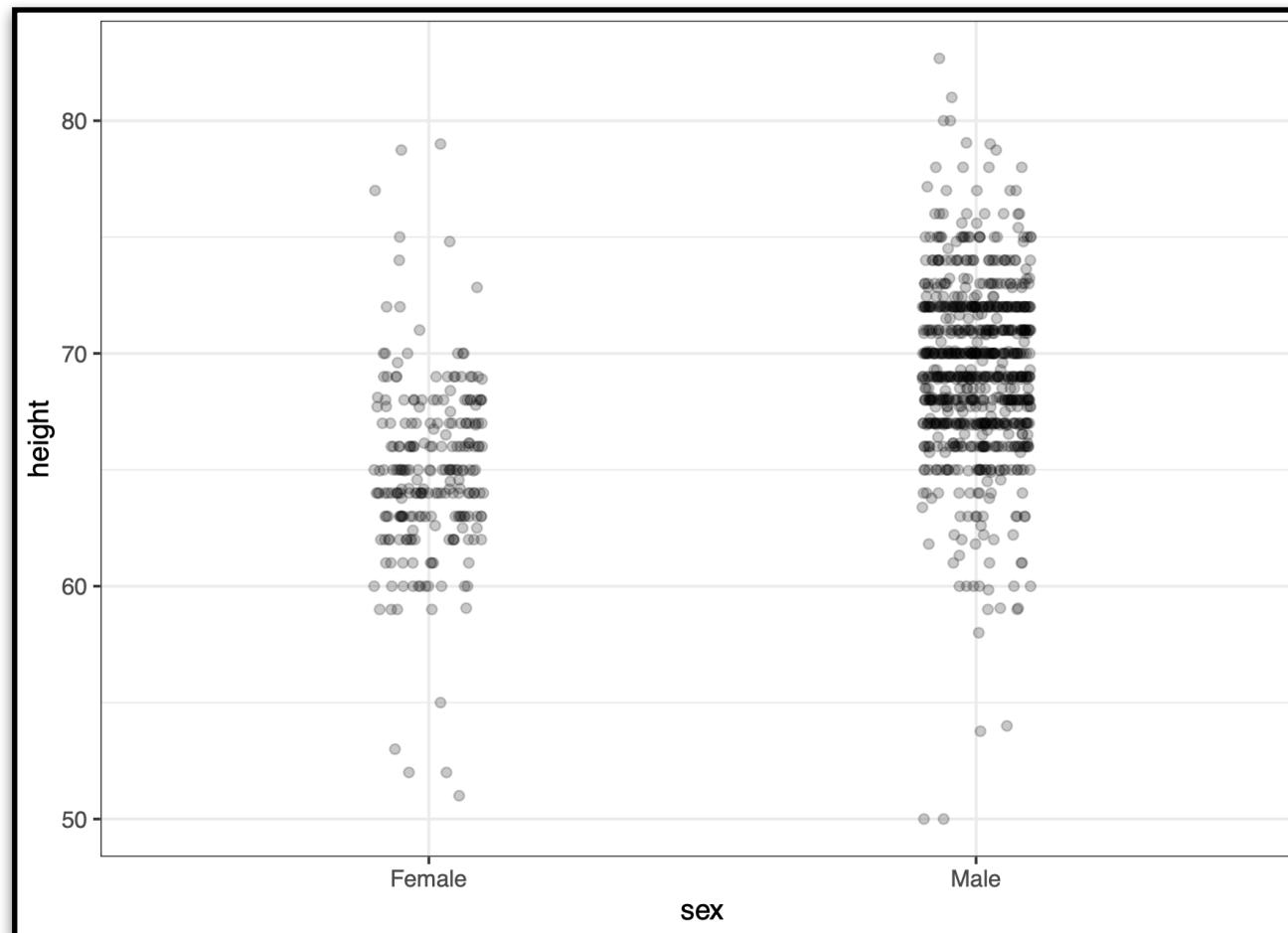
# Think of the color blind

- About 10% of the population is color blind
- The default color scheme in ggplot2 is not optimal for this population
- You can find a colorblind-friendly color palette [here](#)
- Here is one example:



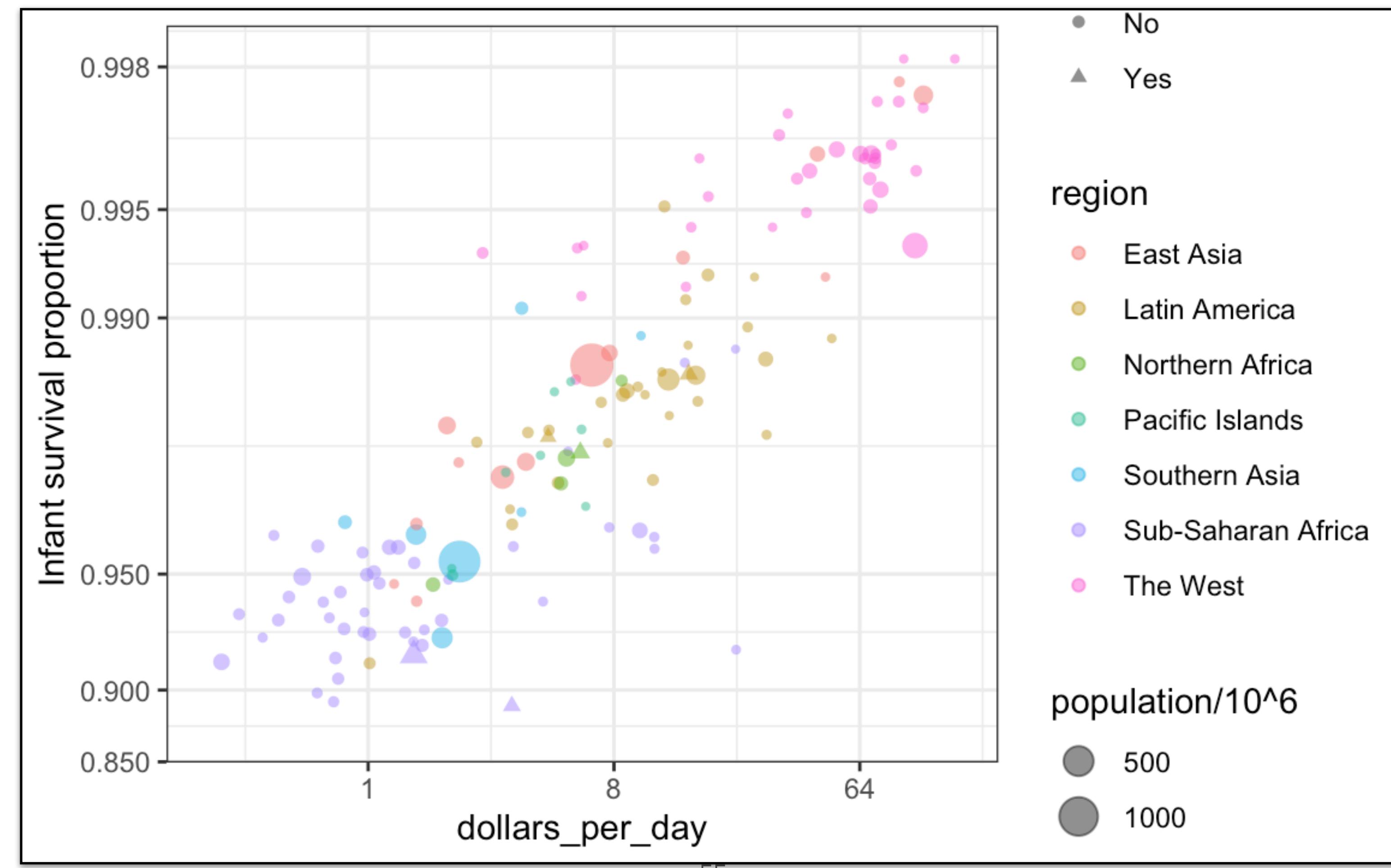
# Encoding more than two variables

- Most of the figures we have seen so far encode information on two variables:



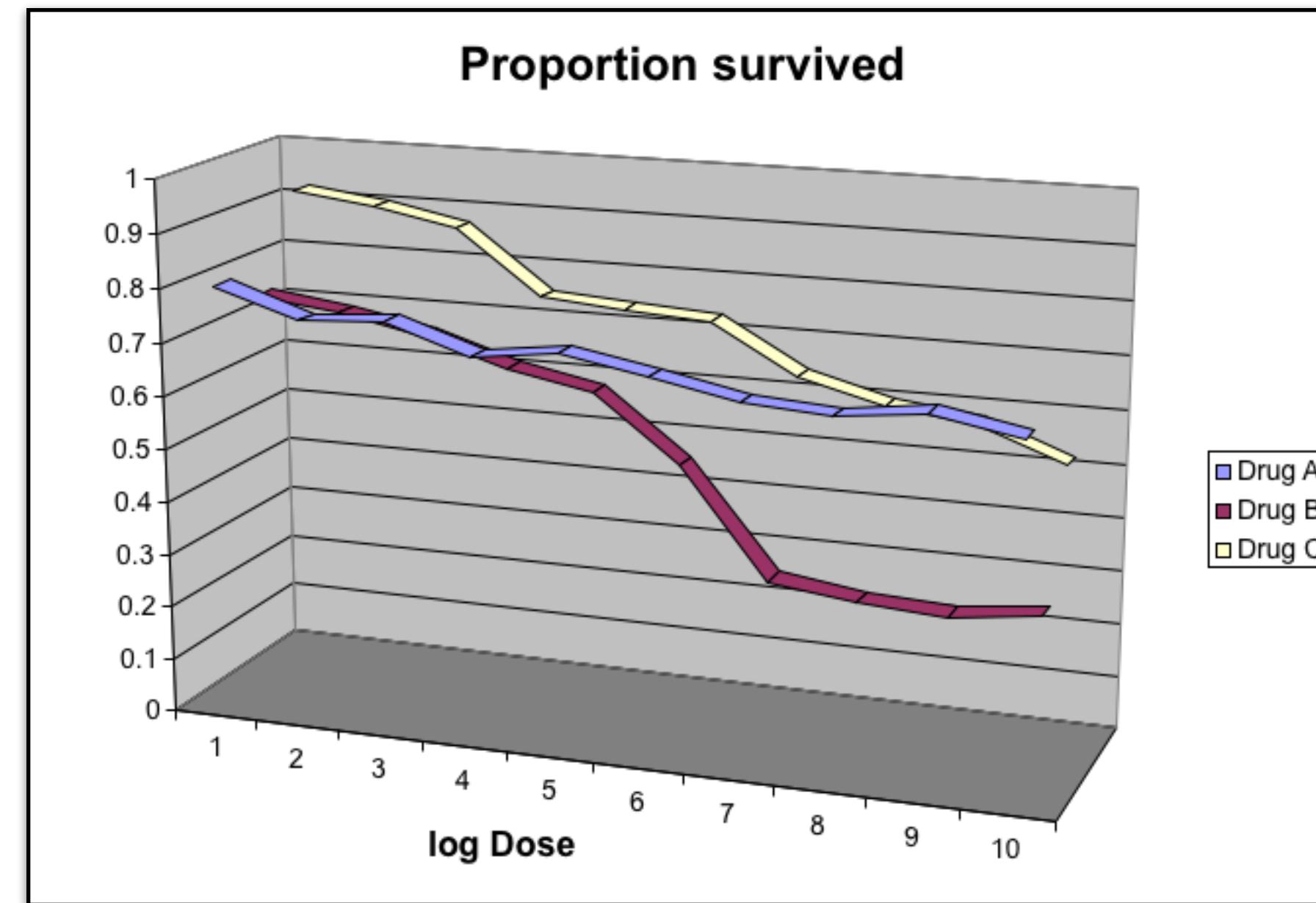
# Encoding more than two variables

- In this example, we encode 3 variables: OPEC membership, region, and population



# Avoid pseudo-three dimensional plots

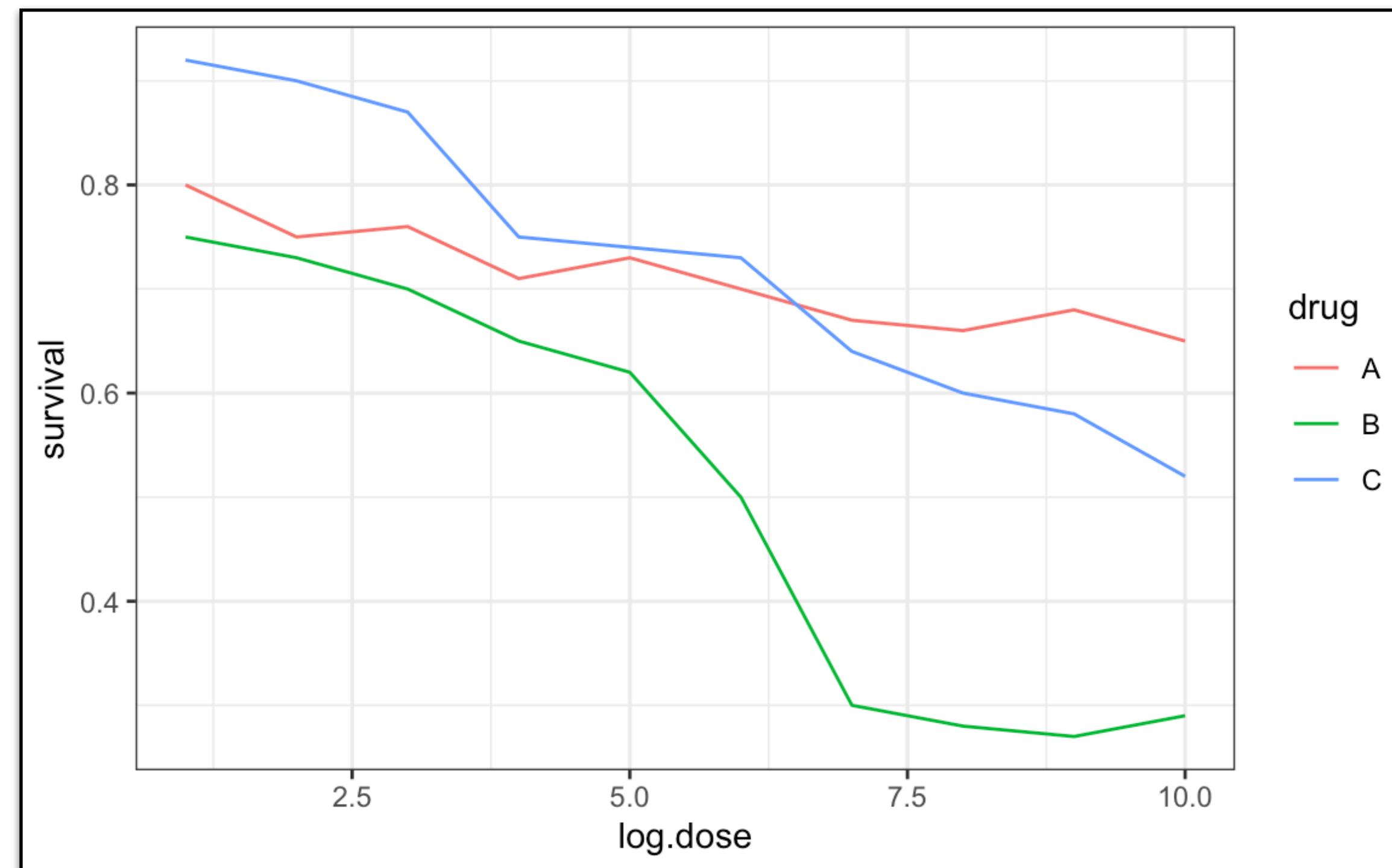
- Humans are not well-equipped to understand 3d figures
- Take the following example:



- Can you tell when the purple ribbon intersects the red one?

# Avoid pseudo-three dimensional plots

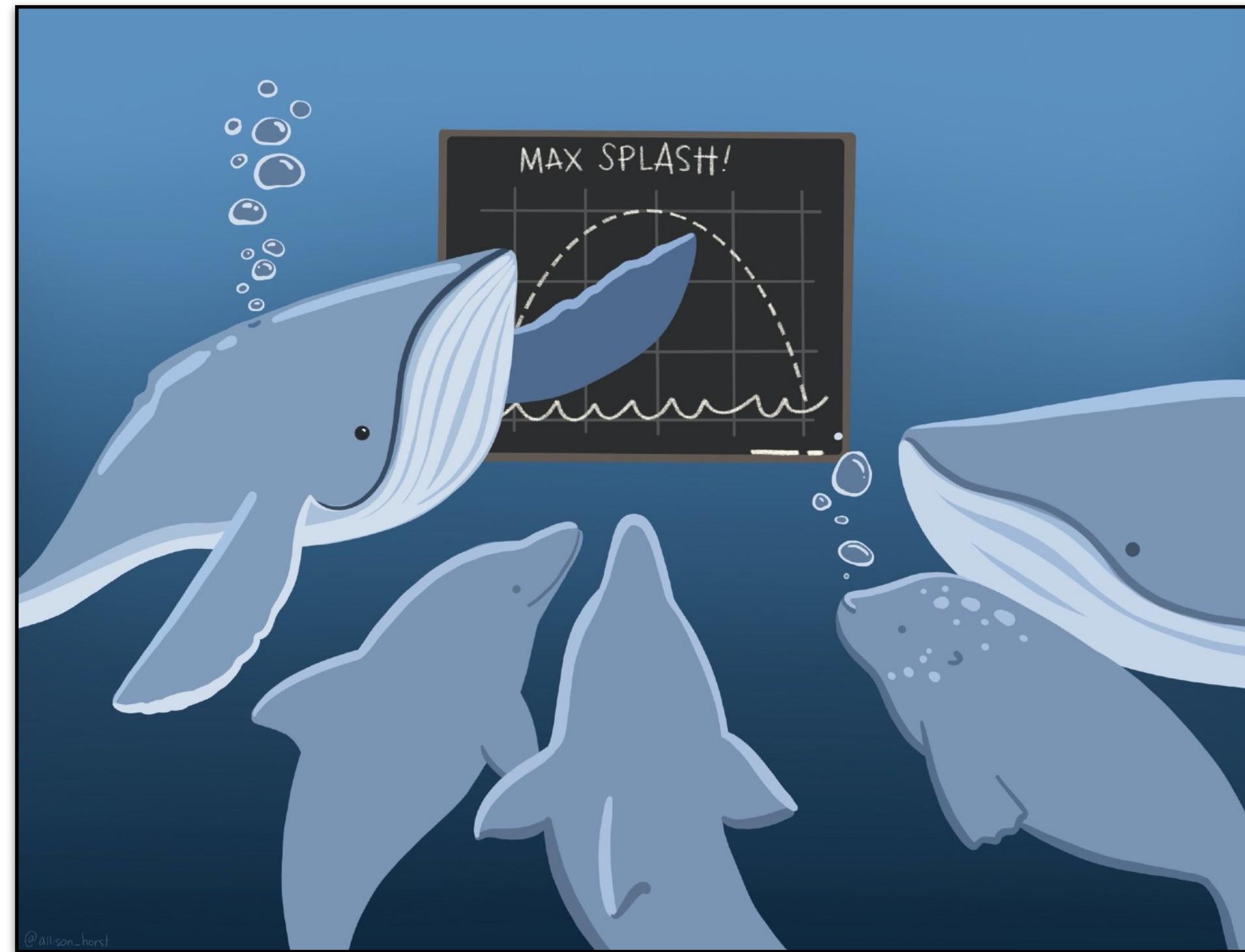
- Humans are not well-equipped to understand 3d figures
- Take the following example:



- They never do!

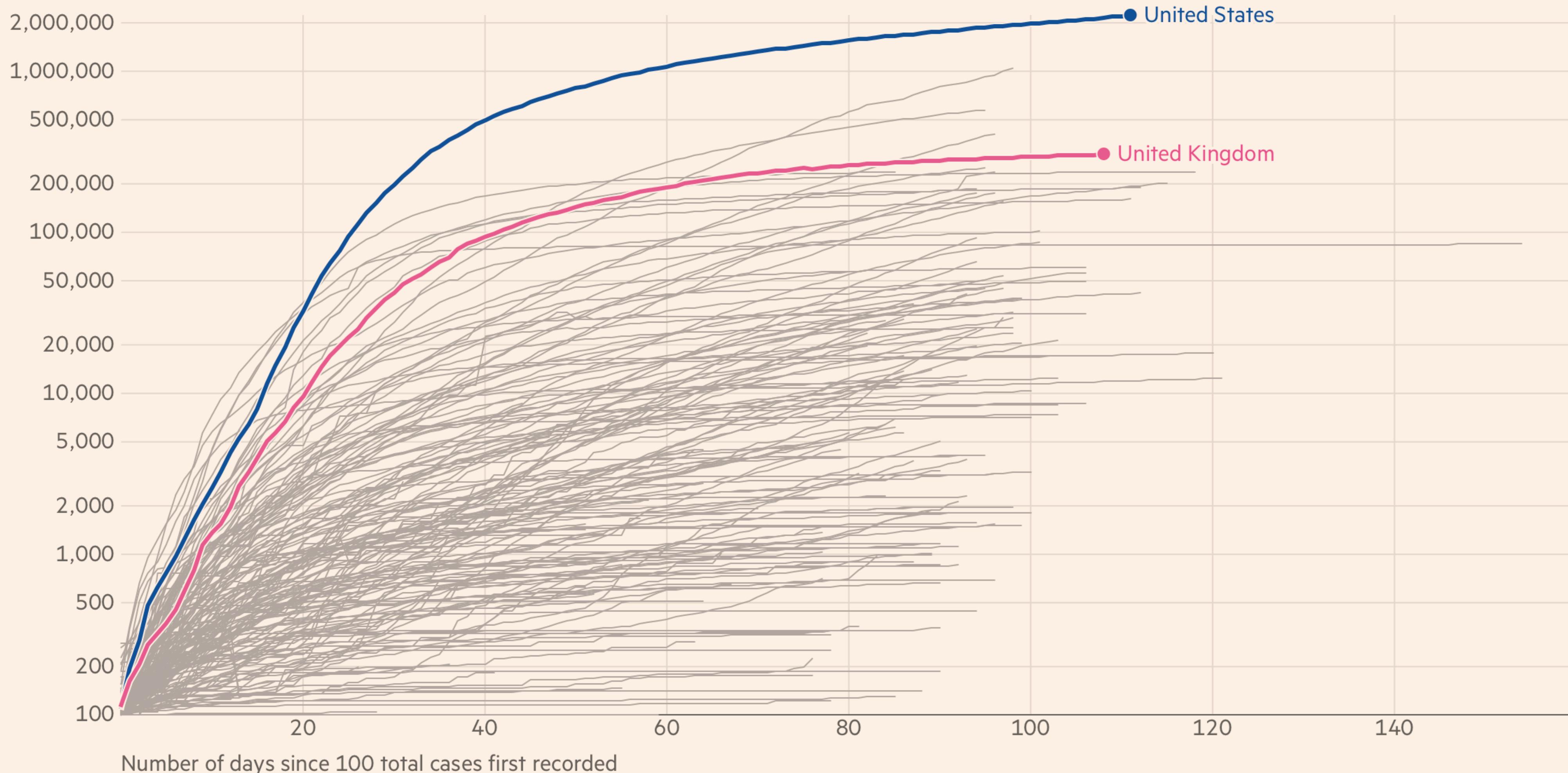
# Know your audience

- Graphs can be used for:
  1. Our own exploratory data analysis
  2. To convey a message to experts
  3. To help tell a story to a general audience
- For example, we may use a log transform of the *y-axis* for our EDA but may advise against it when presenting to a general audience



## Cumulative confirmed cases of Covid-19 in United States and United Kingdom

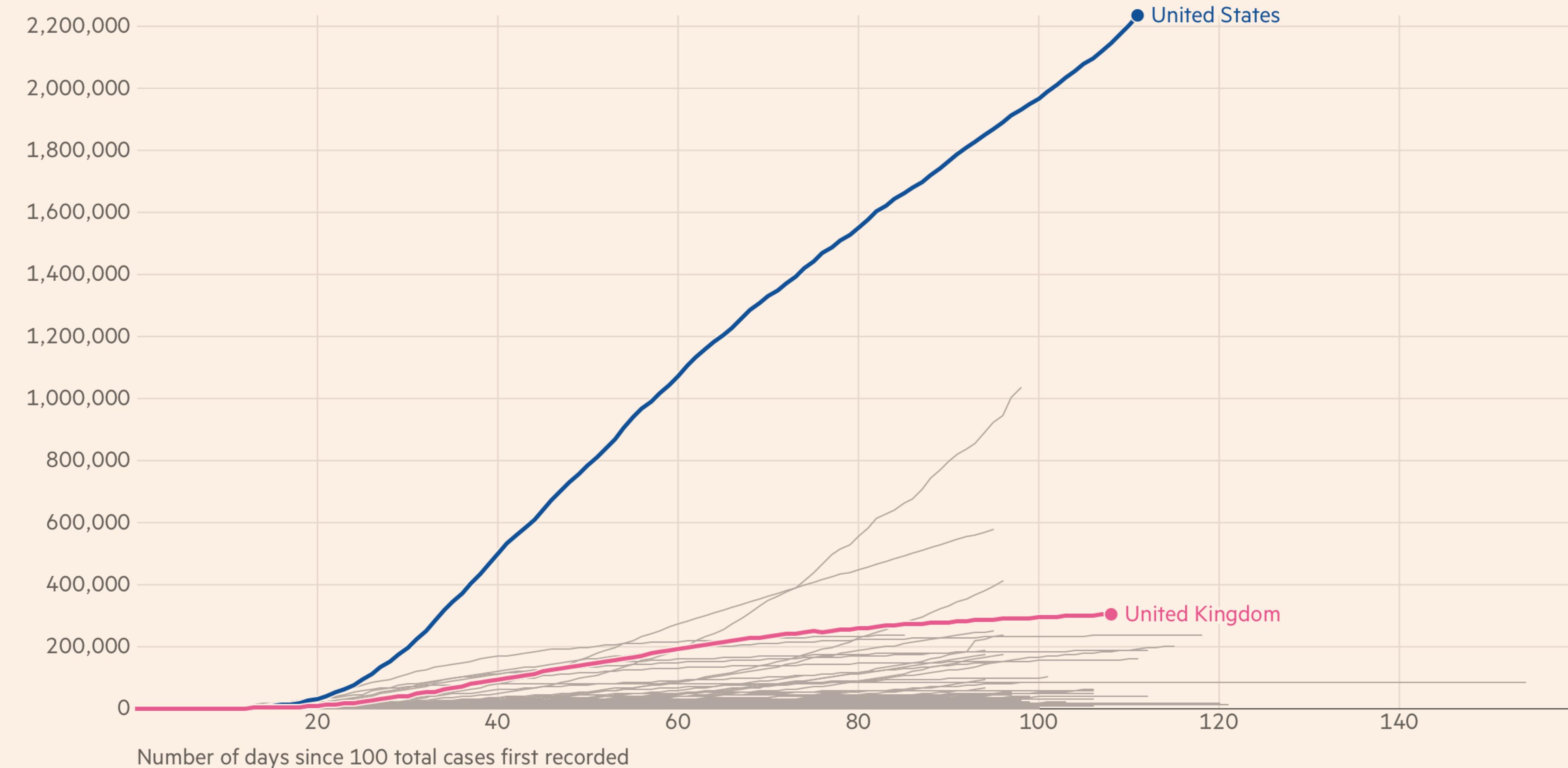
Cumulative cases, by number of days since 100 total cases first recorded



Source: FT analysis of data from the European Centre for Disease Prevention and Control and the Covid Tracking Project. Data updated June 21 2020 5.43pm BST

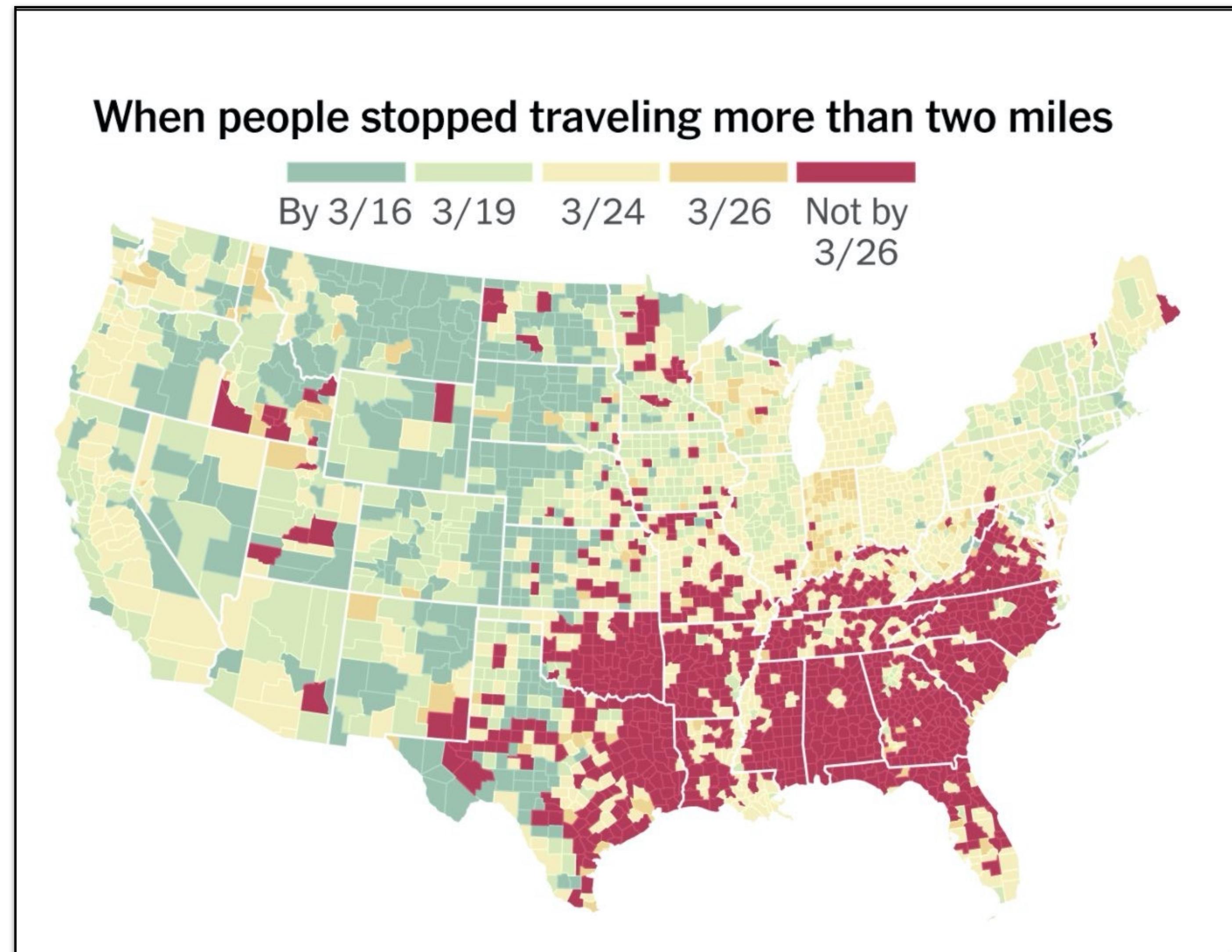
## Cumulative confirmed cases of Covid-19 in United States and United Kingdom

Cumulative cases, by number of days since 100 total cases first recorded



Source: FT analysis of data from the European Centre for Disease Prevention and Control and the Covid Tracking Project. Data updated June 21 2020 5.43pm BST

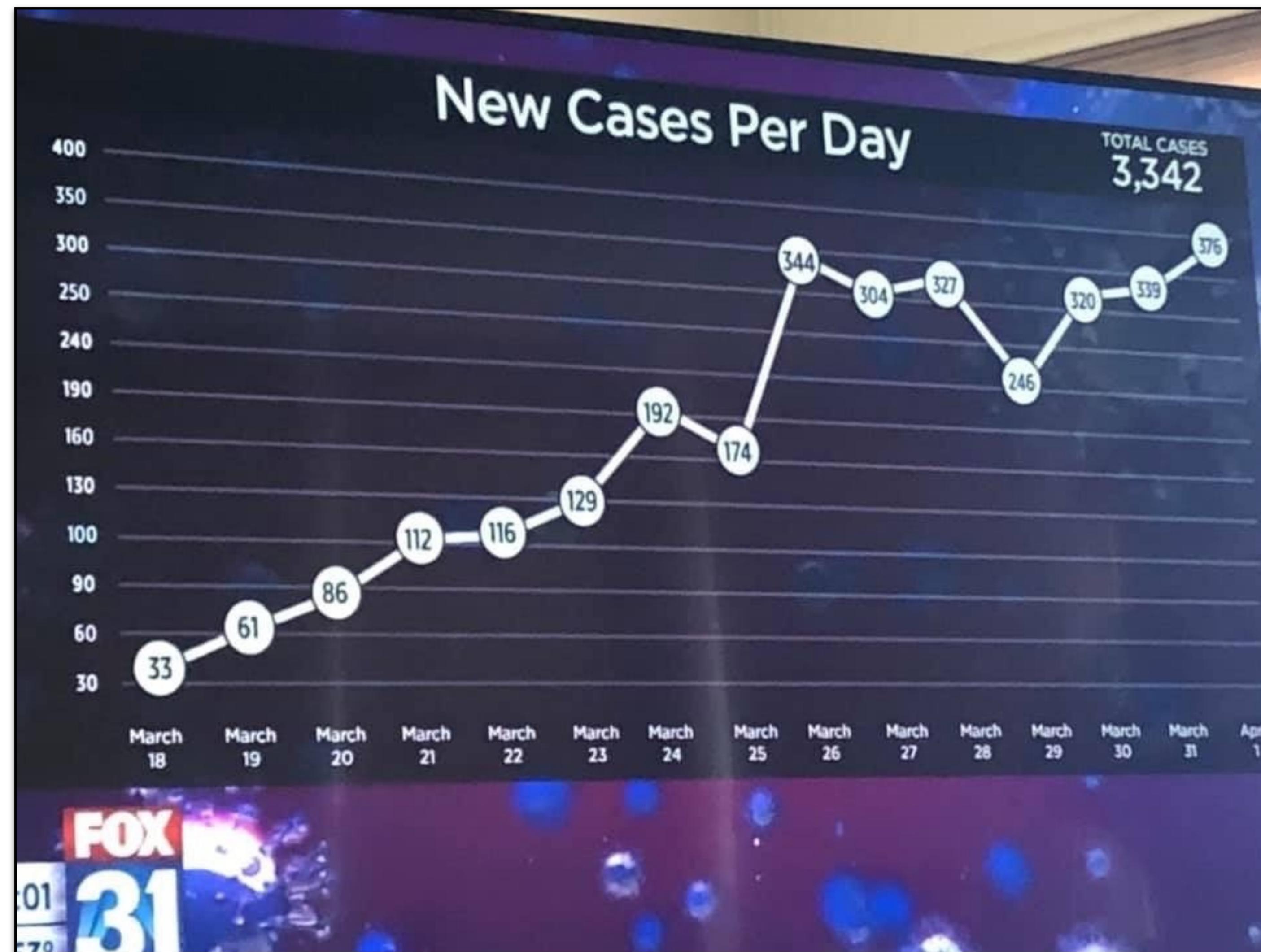
# Beware of biases when creating figures



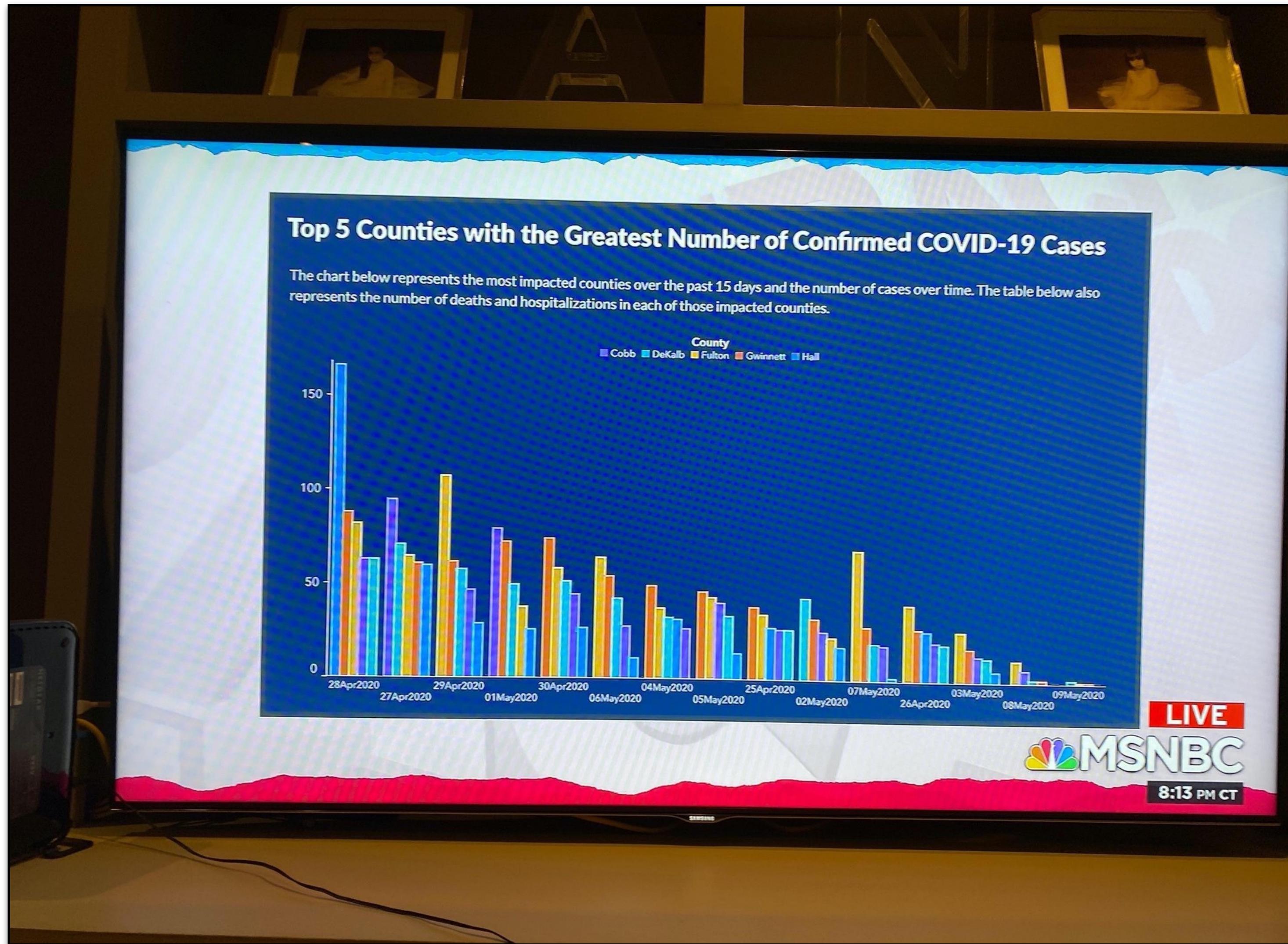
# Beware of biases when creating figures



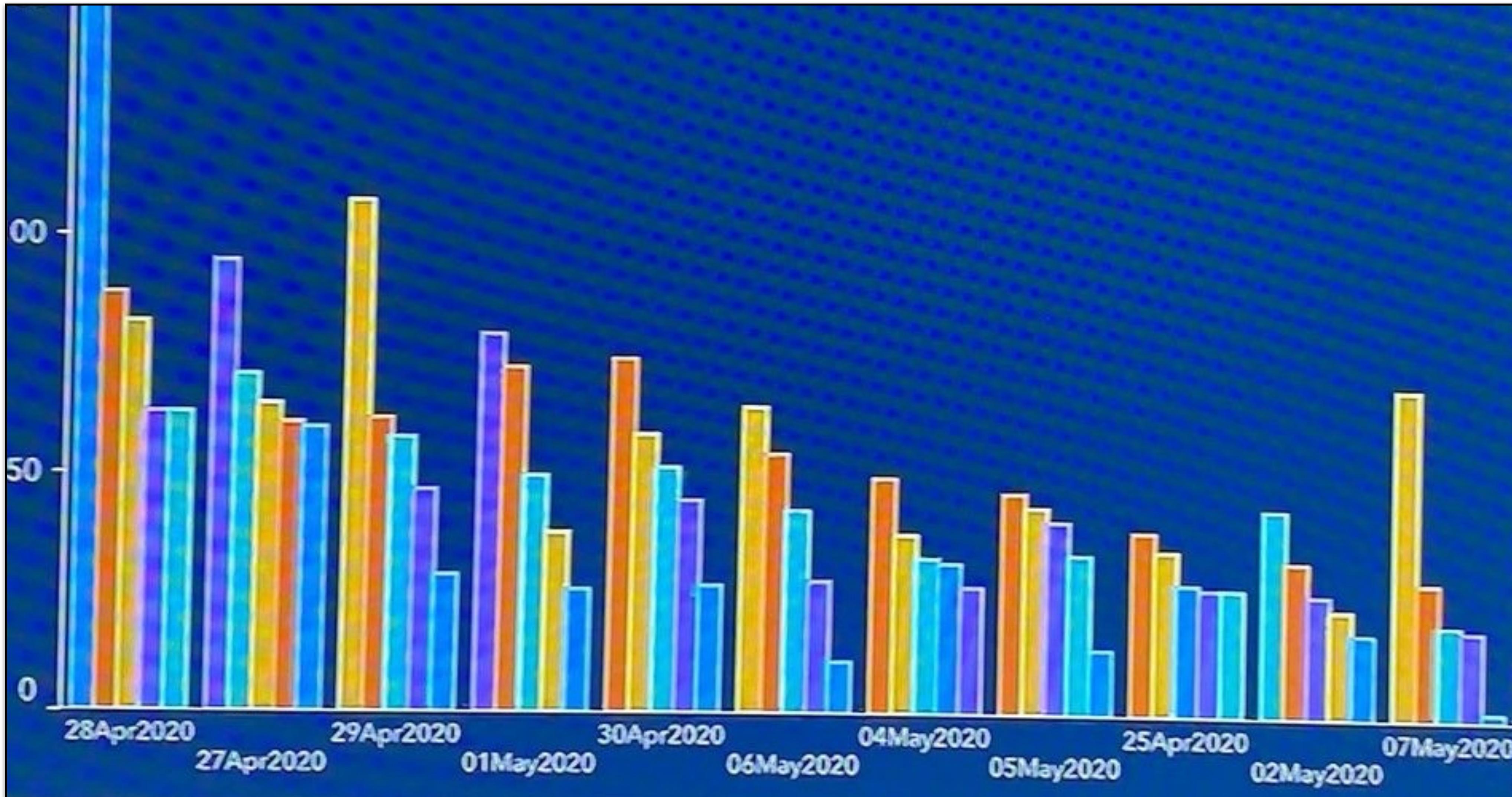
# Beware of misleading practices



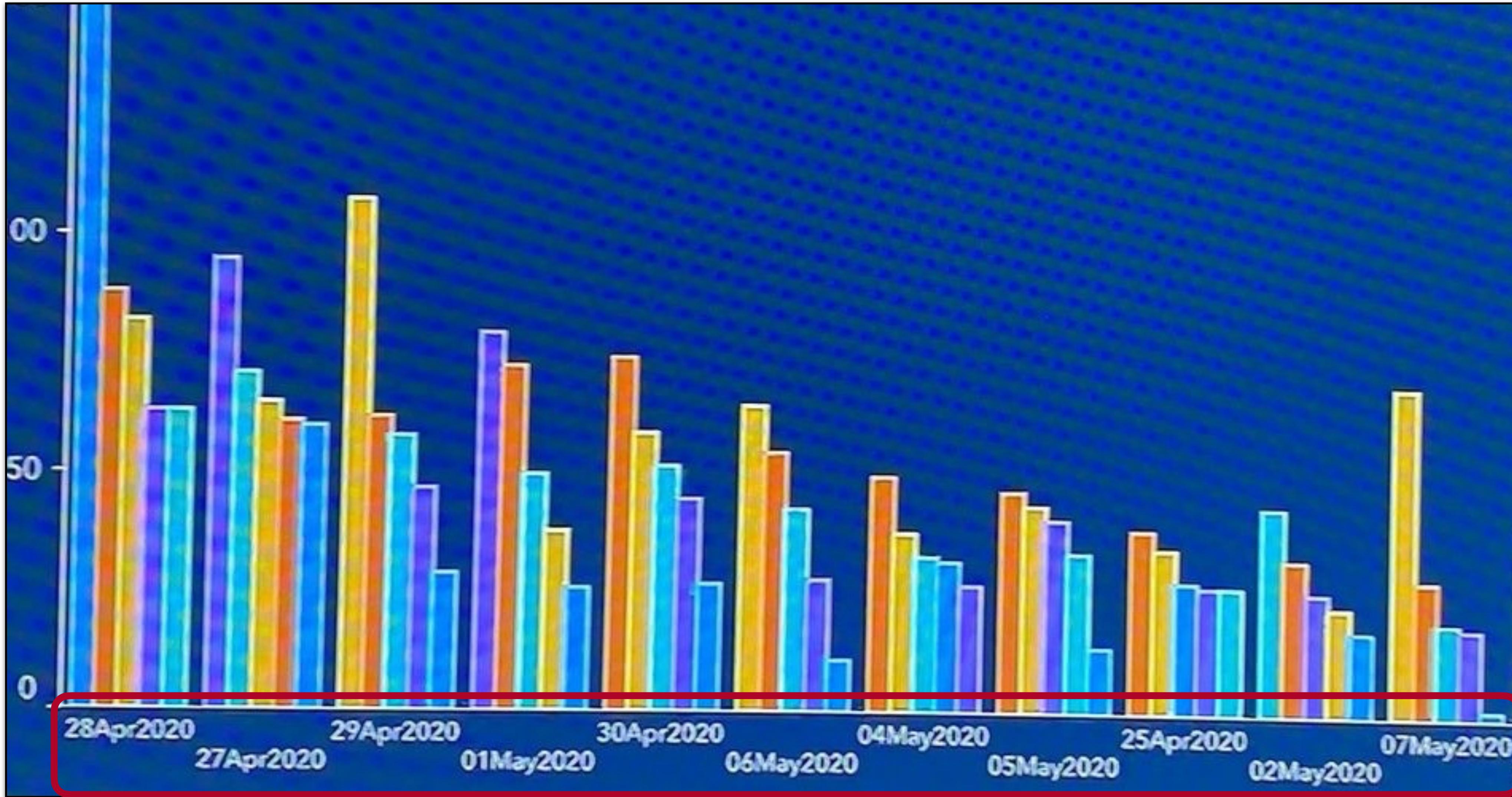
# Beware of misleading practices



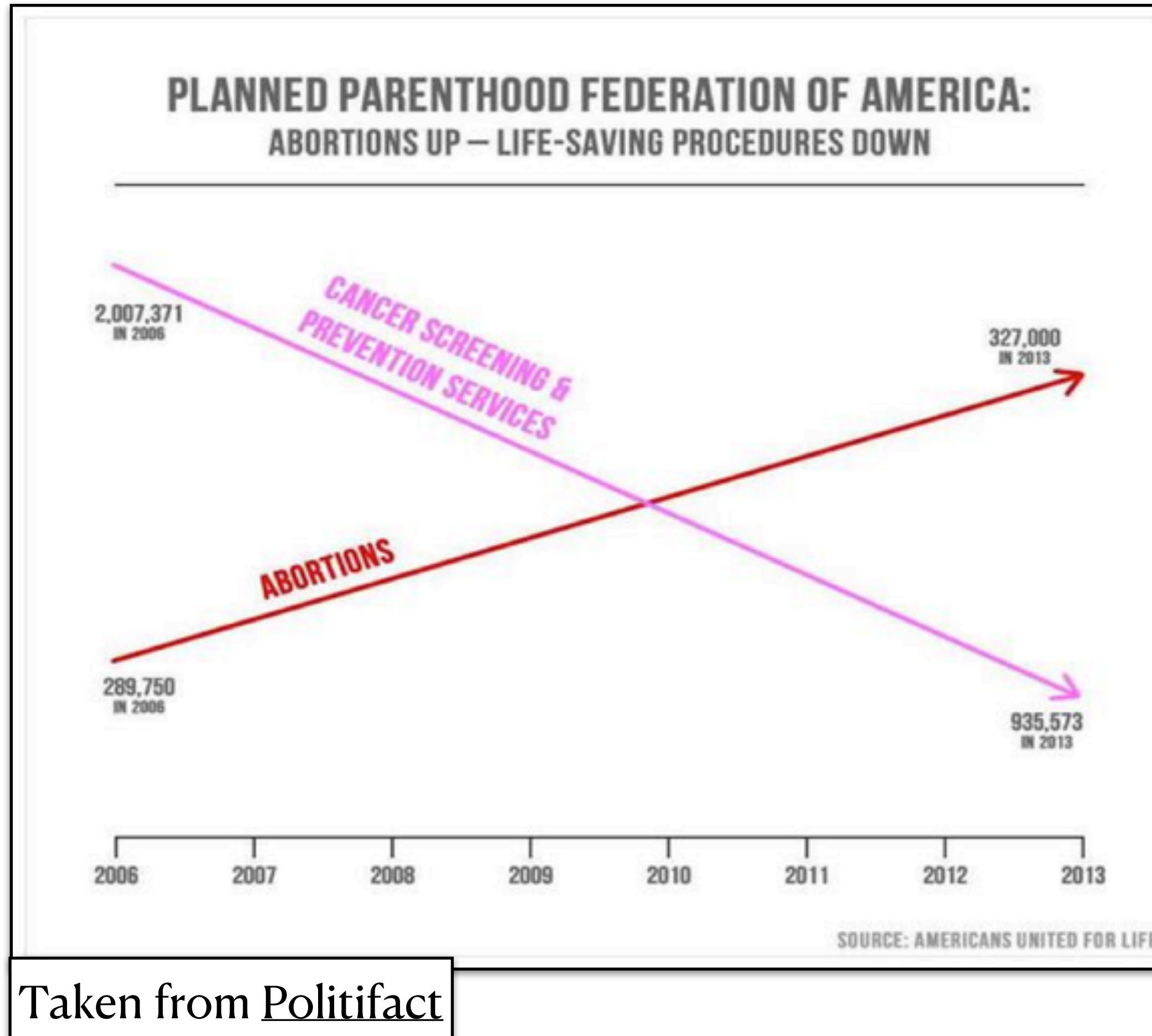
# Beware of misleading practices



# Beware of misleading practices



# Beware of misleading practices



Increase in abortions from 2006 to 2013

$$327,000 - 289,750 = 37,250$$

Decrease in other services from 2006 to 2013

$$2,007,371 - 935,573 = 1,071,798$$

# References

1. Introduction to Data Science: Data analysis and prediction algorithms with R by Rafael A. Irizarry, Chapter 10. <https://rafalab.github.io/dsbook/>
2. Creating effective figures and tables by Karl W. Brown. <https://www.biostat.wisc.edu/~kbroman/presentations/graphs2017.pdf>

## Referencias en español:

1. Introducción a la Ciencia de Datos: Análisis de datos y algoritmos de predicción con R por Rafael A. Irizarry, Capítulo 10. <https://rafalab.github.io/dslibro/>