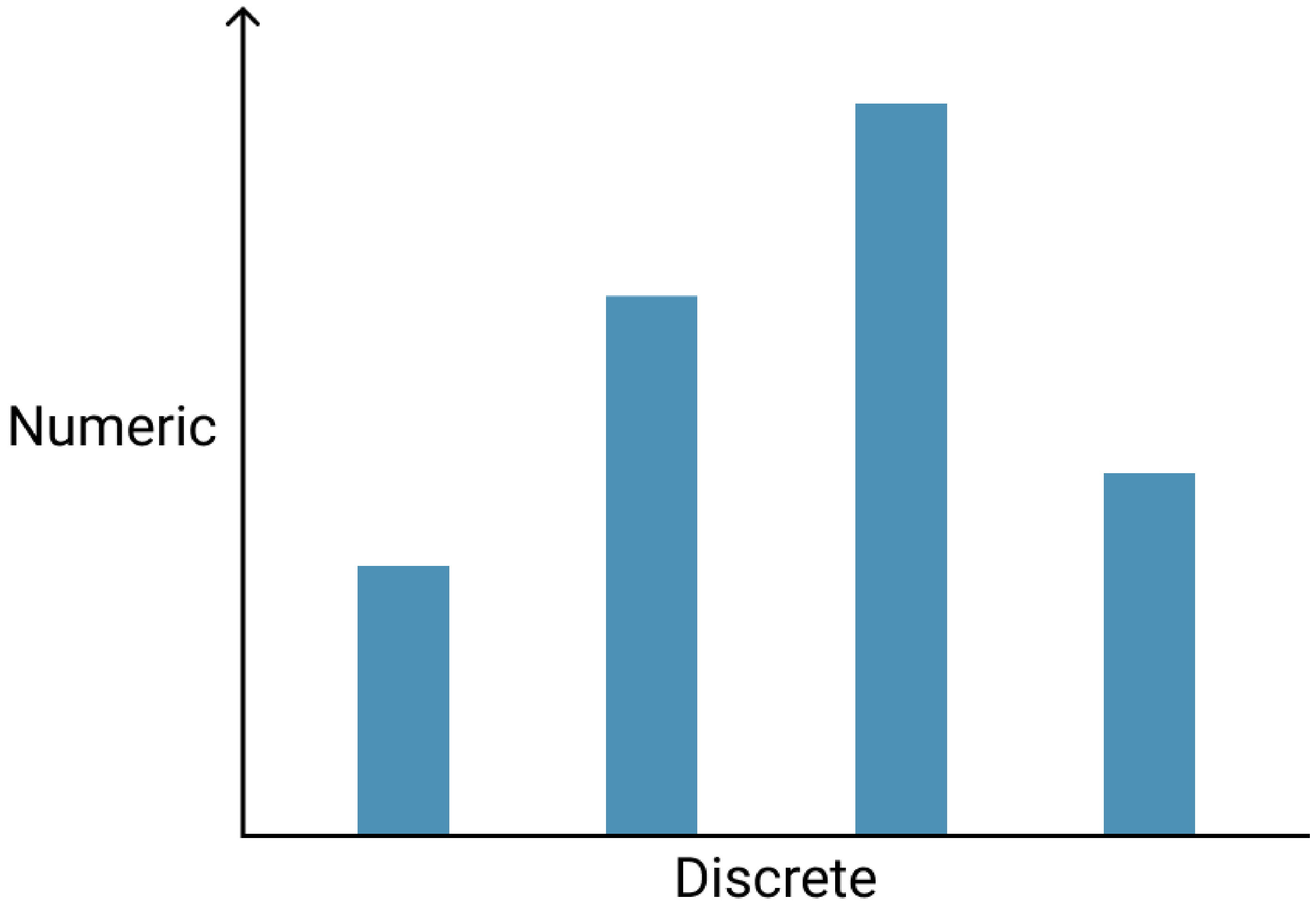
Bars and dots: point data

VISUALIZATION BEST PRACTICES IN R

Nick Strayer
Instructor

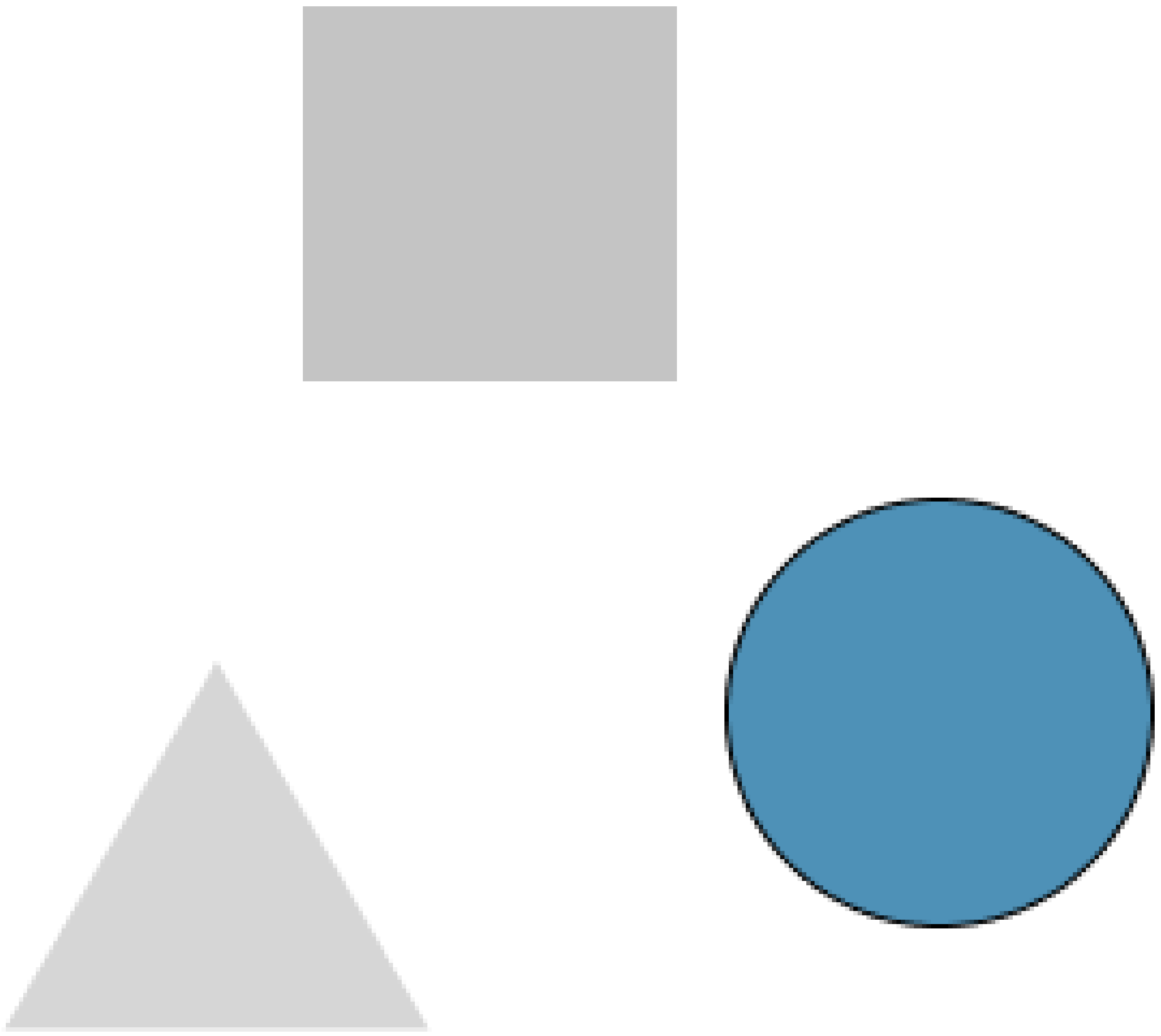
What is point data?

- One categorical axis, one numeric
- Counts, averages, rates, etc.



A single observation

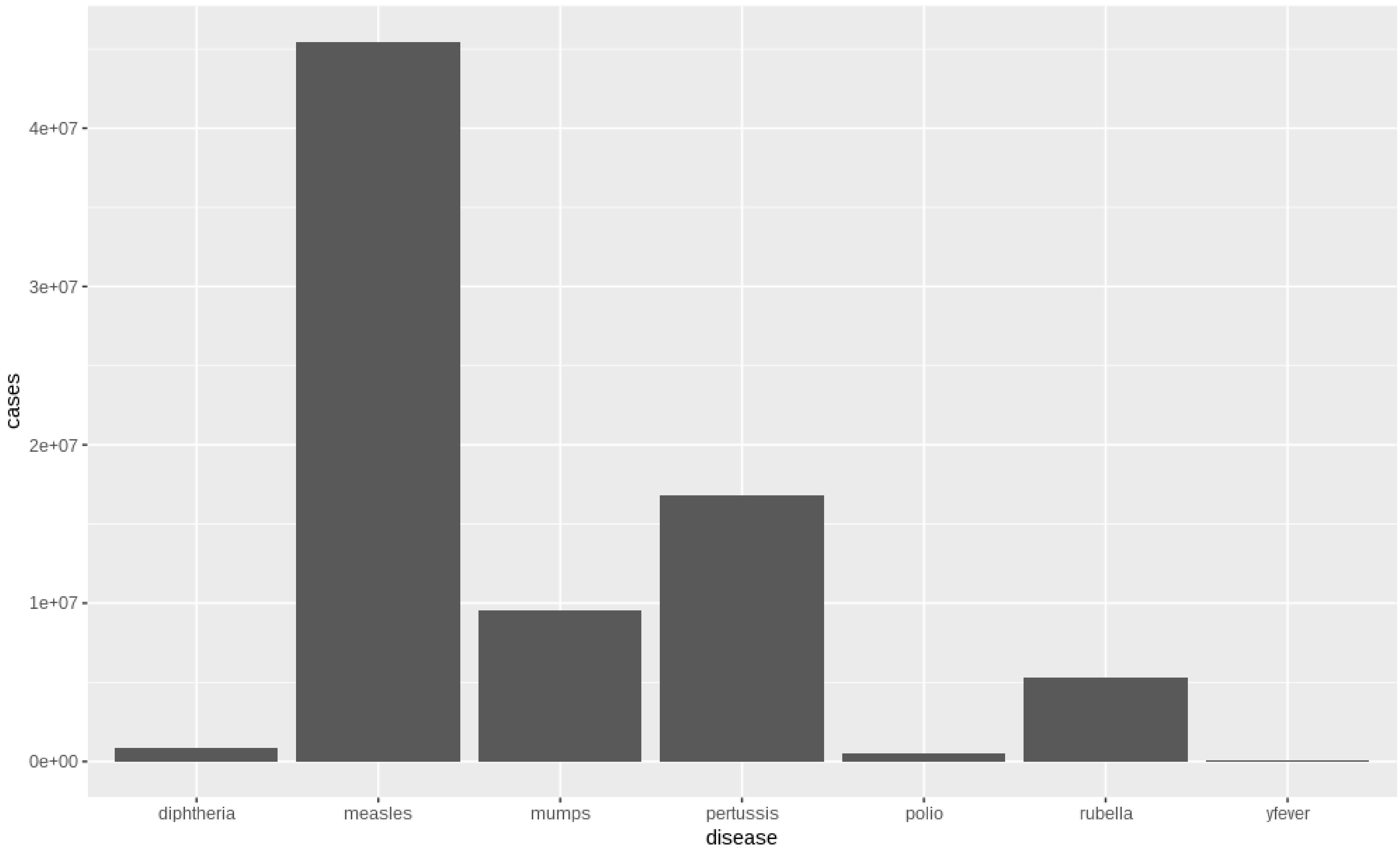
- Represents a singular observation of something
- E.g. population of a state, rate of cell growth



The bar chart

- Popular
- Simple
- Accurate

```
ggplot(who_disease) +
  geom_col(aes(x = disease, y = cases))
```


4e+07
3e+07
2e+07
1e+07
0e+00
cases
diphtheria
measles
mumps
pertussis
polio
rubella
yfever
disease

Not always the best

- Bar charts are frequently used when other charts are more appropriate
- A few principles can be followed to help avoid this

The stacking principle

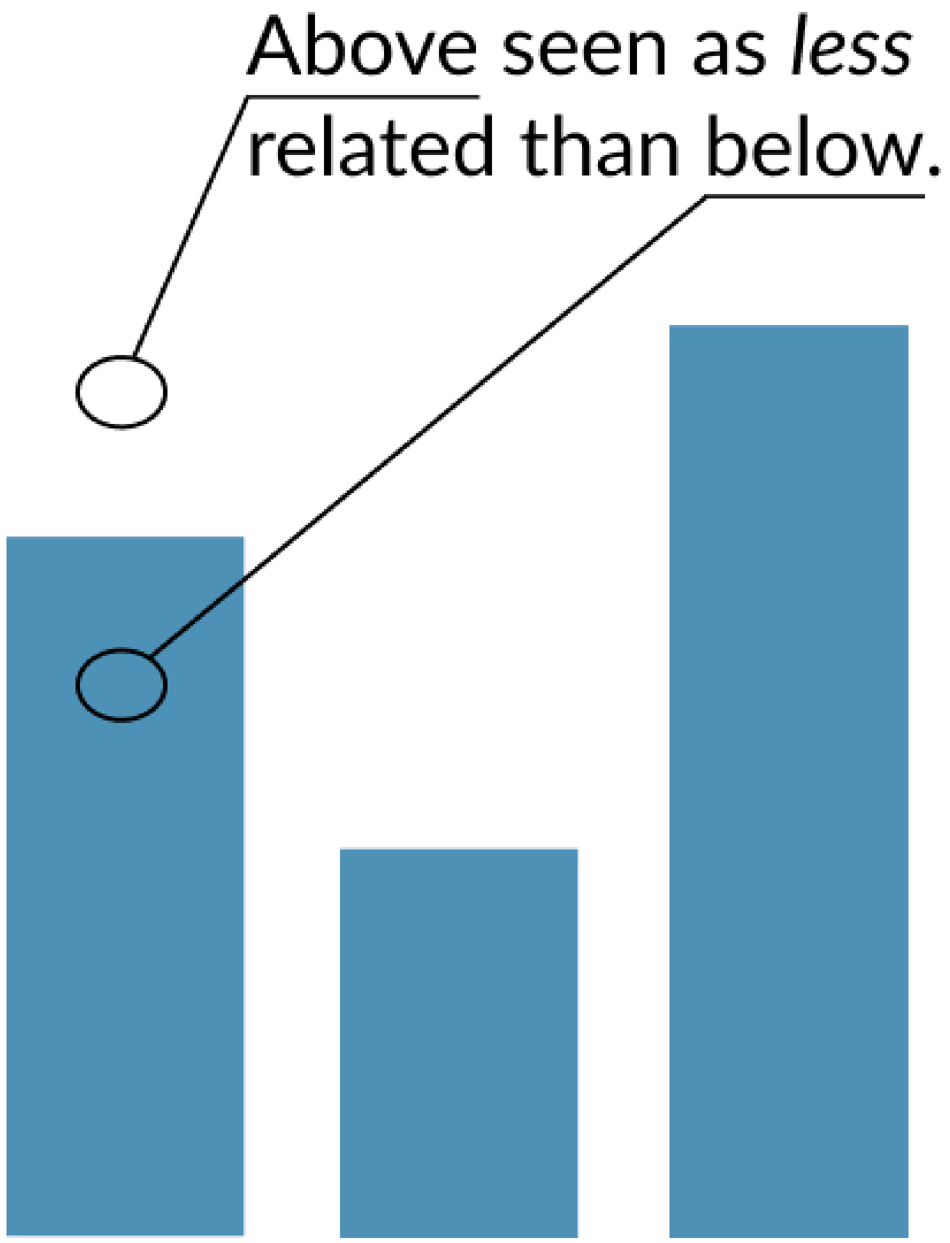
- Should be used for data that represents a meaningful quantity
- Ask: 'Could I stack what I'm measuring to make the bars?'

Why quantities?

> *"...viewers judge points that fall within the bar as being more likely than points equidistant from the mean, but outside the bar..."*

- Scholl & Newman, 2012

- People view the bar as 'containing' the values below top
- Quantities fulfill this assumption



A big deal?

- Not really...
- ... but alternatives are not *worse,* so they may as well be used

¯_(ツ)_/¯

Let's practice!

VISUALIZATION BEST PRACTICES IN R

Point charts

VISUALIZATION BEST PRACTICES IN R

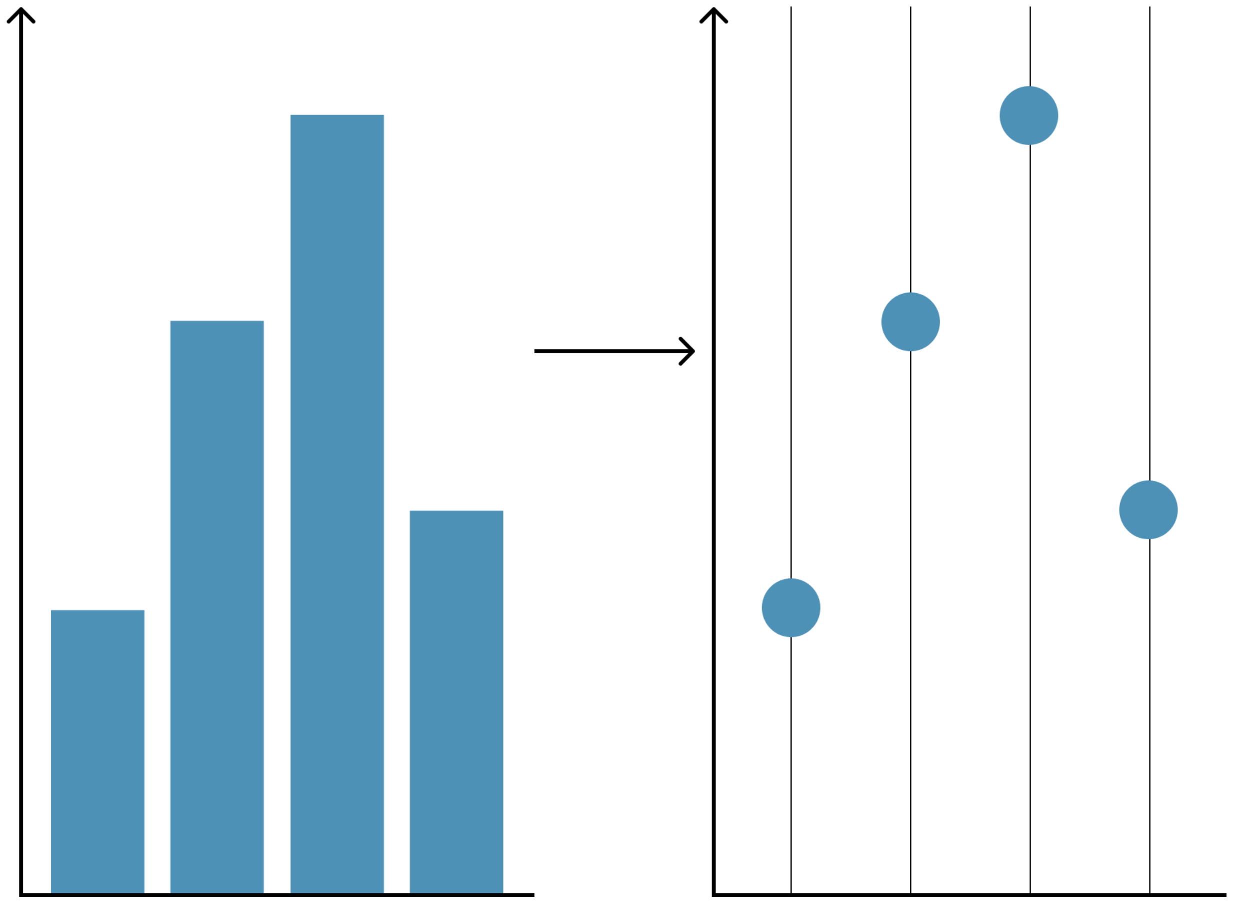
Nick Strayer
Instructor

When a bar chart isn't ideal

- Not a quantity
- Non-linear transformations

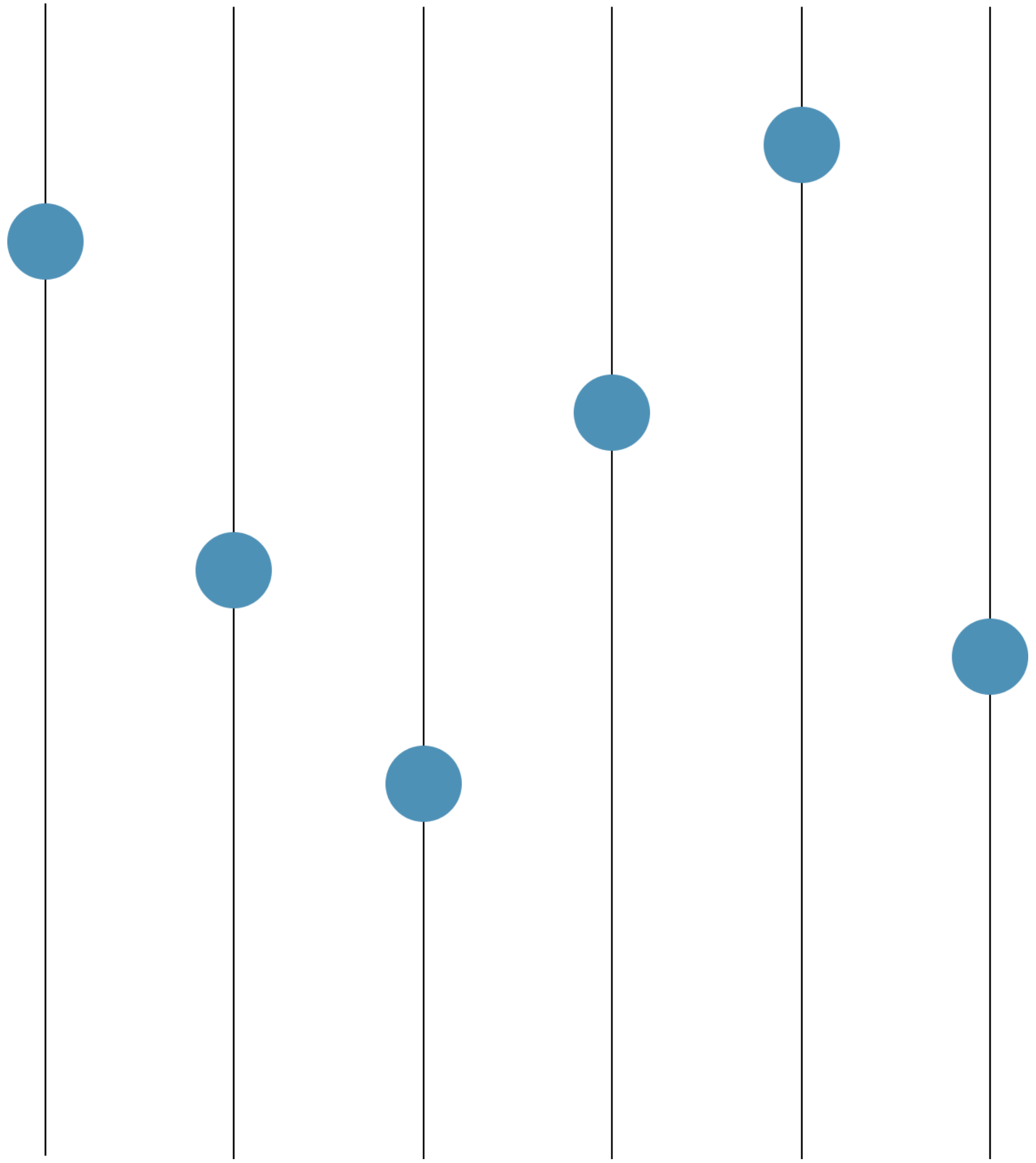
Point charts

- Simply replace bar with a point
- Sometimes called point charts or dot plots



Benefits of point charts

- High precision
- Efficient representation
- Simple



Data for lesson

- Working with a subset of WHO data
- Countries are an 'interesting' subset -- let's see if we can find out why

```r
interestingCountries <- c("NGA", "SDN", "FRA", "NPL", "MYS", "TZA", "YEM", "UKR", "BGD", "VNM")

who_subset <- who_disease %>%
  filter(
    countryCode %in% interestingCountries,
    disease == 'measles',
    year %in% c(2006, 2016)
  ) %>%
  mutate(year = paste0('cases_', year)) %>%
  spread(year, cases)
```

who_subset

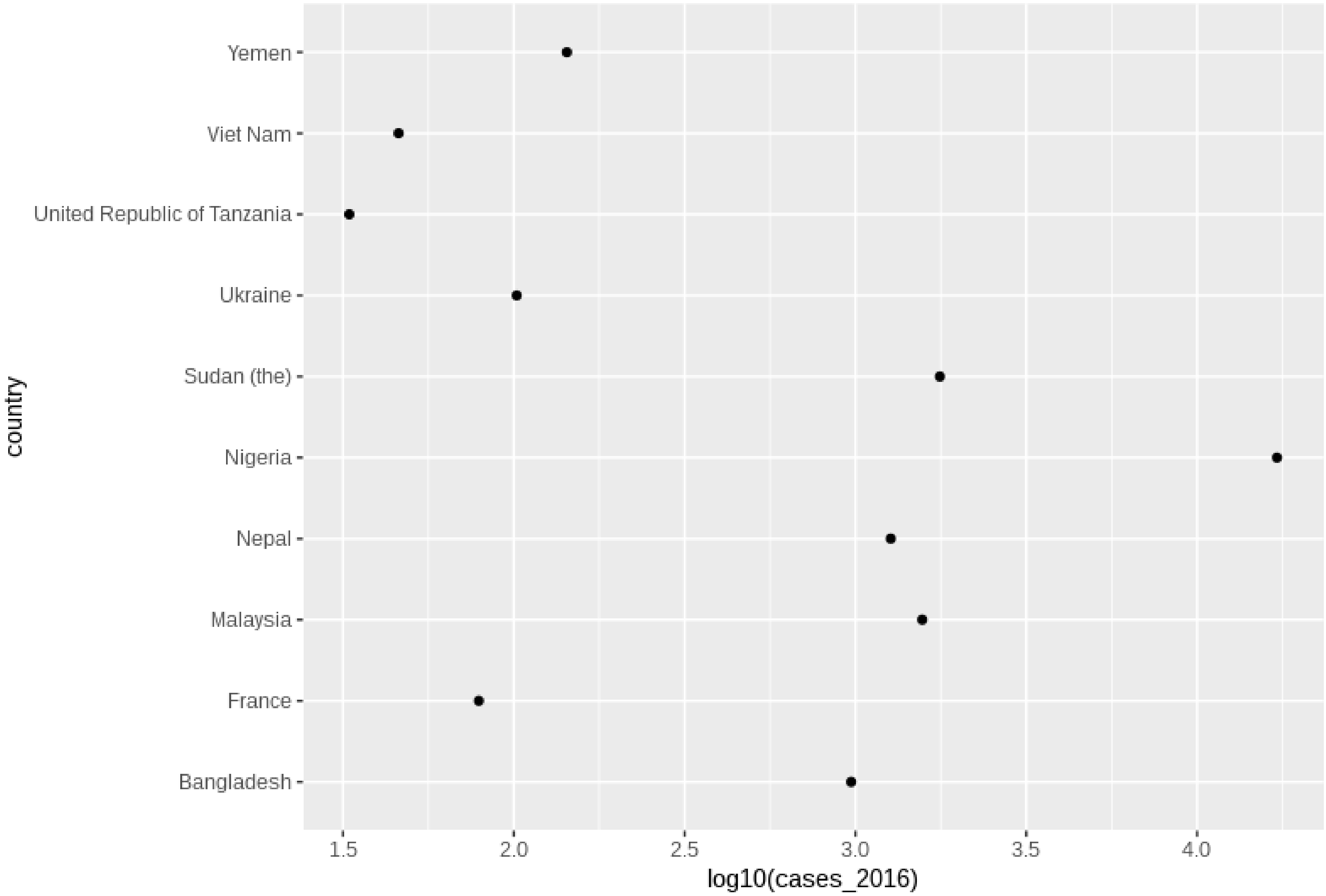
```
who_subset
```

```
# A tibble: 10 x 6
   region countryCode country     disease cases_2006 cases_2016
   <chr>  <chr>       <chr>       <chr>        <dbl>      <dbl>
 1 AFR    NGA         Nigeria     measles        704      17136
 2 AFR    TZA         Tanzania    measles       2362         33
 3 EMR    SDN         Sudan (the) measles        228       1767
 4 EMR    YEM         Yemen       measles       8079        143
 5 EUR    FRA         France      measles         40         79
 6 EUR    UKR         Ukraine     measles      42724        102
 7 SEAR   BGD         Bangladesh  measles       6192        972
 8 SEAR   NPL         Nepal       measles       2838       1269
 9 WPR    MYS         Malaysia    measles        564       1569
10 WPR    VNM         Viet Nam    measles       1978         46
```

Code for point charts

- `geom_point()` with one categorical and one numerical axis

```r
who_subset %>%
  # We log transform our values here so bars are not appropriate
  ggplot(aes(y = country, x = log10(cases_2016))) +
  # Simple geom_point()
  geom_point()
```

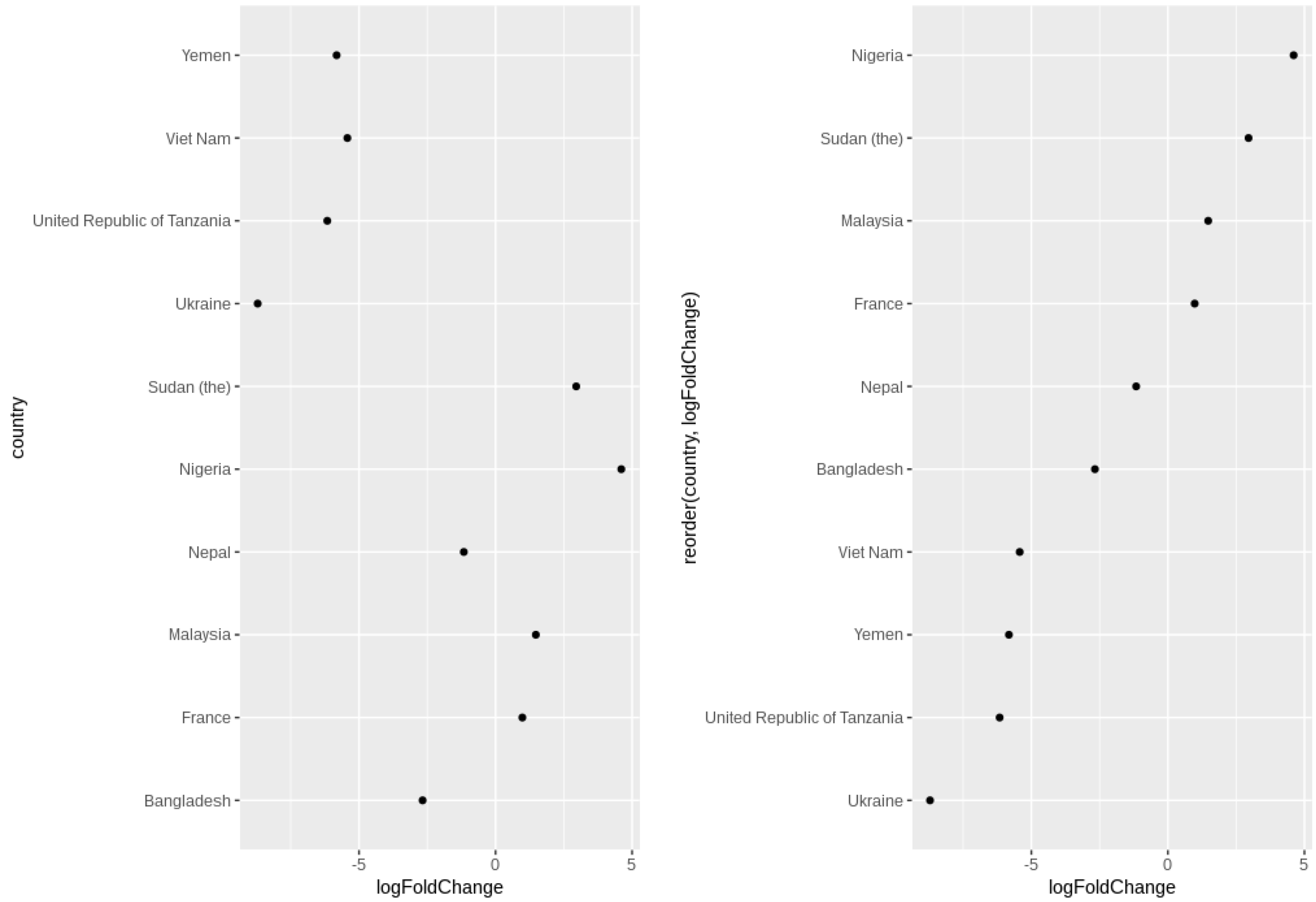

Yemen
Viet Nam
United Republic of Tanzania
Ukraine
Sudan (the)
Nigeria
Nepal
Malaysia
France
Bangladesh
country
1.5
2.0
2.5
3.0
3.5
4.0
log10(cases_2016)

Ordering your point charts

- Ordering can vastly help legibility
- Use the `reorder()` function in the aesthetic assignment

```
who_subset %>%
  # calculate the log fold change between 2016 and 2006
  mutate(logFoldChange = log2(cases_2016/cases_2006)) %>%
  ggplot(aes(x = logFoldChange, y = reorder(country, logFoldChange))) +
  geom_point()
```

reorder



country
Yemen
Viet Nam
United Republic of Tanzania
Ukraine
Sudan (the)
Nigeria
Nepal
Malaysia
France
Bangladesh
-5
0
5
logFoldChange
reorder(country, logFoldChange)
Nigeria
Sudan (the)
Malaysia
France
Nepal
Bangladesh
Viet Nam
Yemen
United Republic of Tanzania
Ukraine
-5
0
5
logFoldChange

Let's practice!

VISUALIZATION BEST PRACTICES IN R

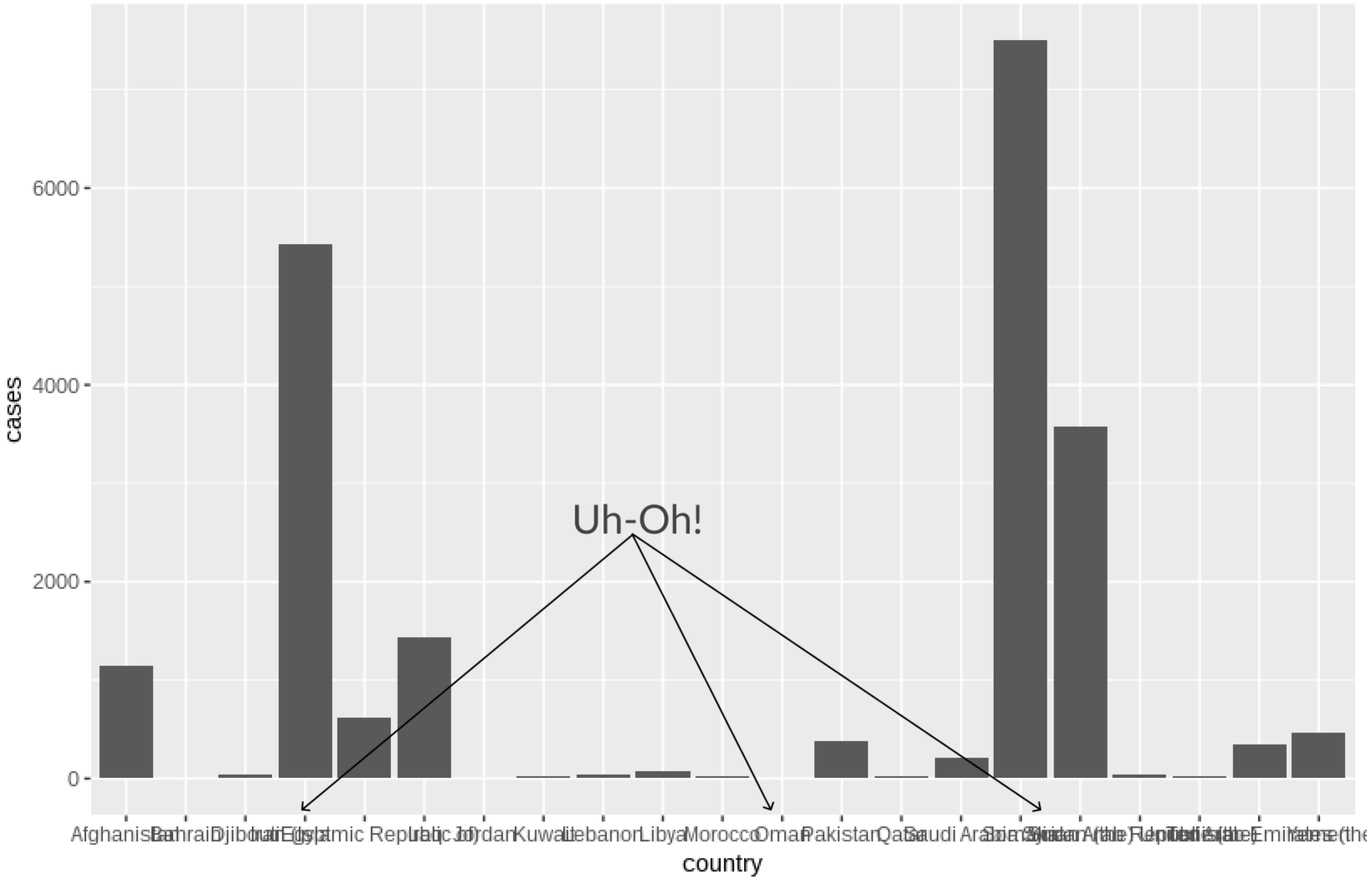
Tuning your bar and point charts

VISUALIZATION BEST PRACTICES IN R

Nick Strayer
Instructor

A busy bar chart

```r
who_disease %>%
  filter(region == 'EMR', disease == 'measles', year == 2015) %>%
  ggplot(aes(x = country, y = cases)) +
    geom_col()
```


cases
6000
4000
2000
0
country
Afghanistan
Bahrain
Djibouti
Egypt
Iran (Islamic Republic of)
Iraq
Jordan
Kuwait
Lebanon
Libya
Morocco
Oman
Pakistan
Qatar
Saudi Arabia
Somalia
Sudan (the)
Syrian Arab Republic
Tunisia
United Arab Emirates (the)
Yemen

Uh-Oh!

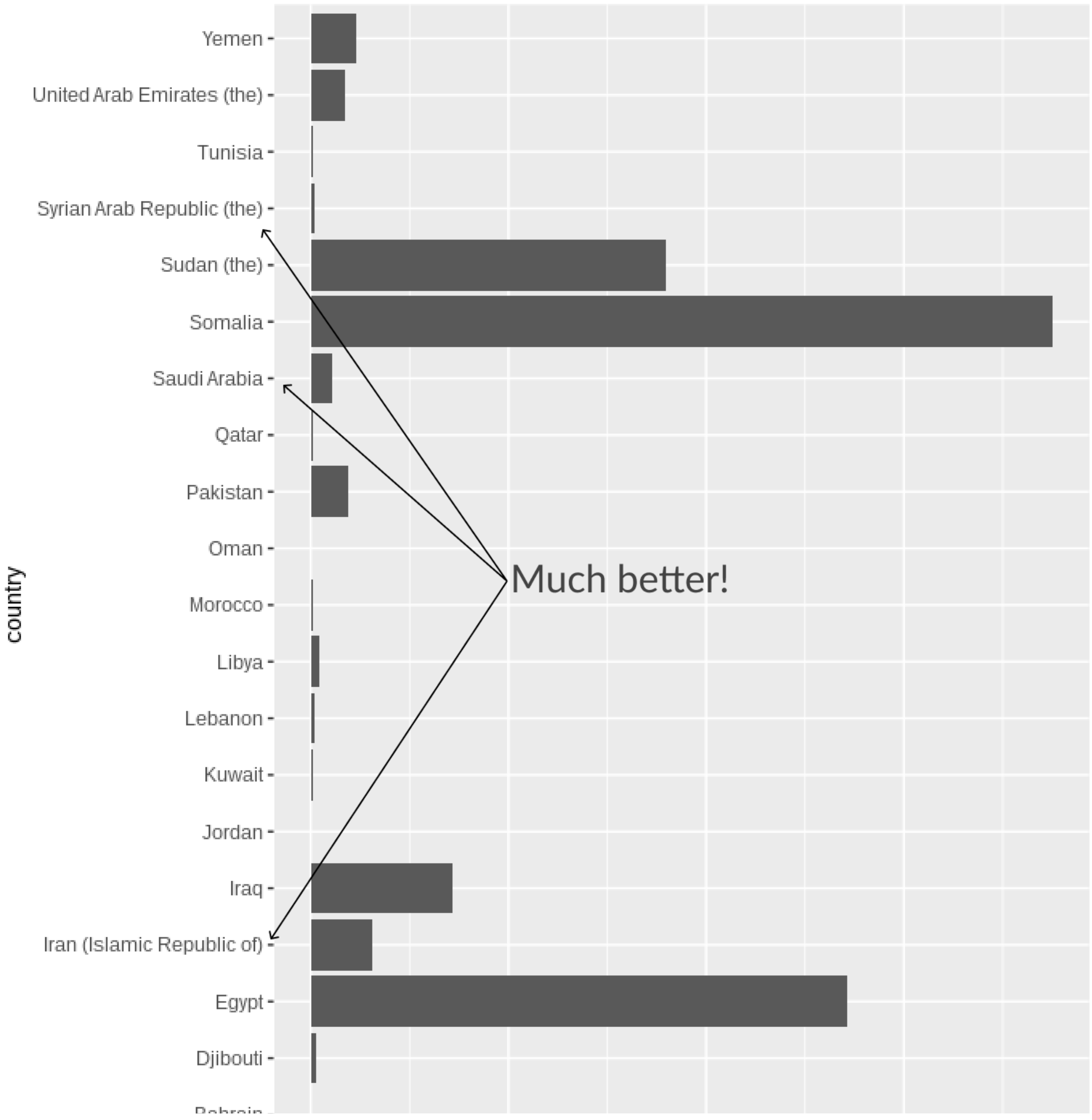
Flipping the bar

- `geom_bar()` and `geom_col()` don't allow categories on y-axis

  ```
  busy_bars <- who_disease %>%
  filter(region == 'EMR', disease == 'measles', year == 2015) %>%
  ggplot(aes(x = country, y = cases)) +
  geom_col()
  ```

- So we have to flip!

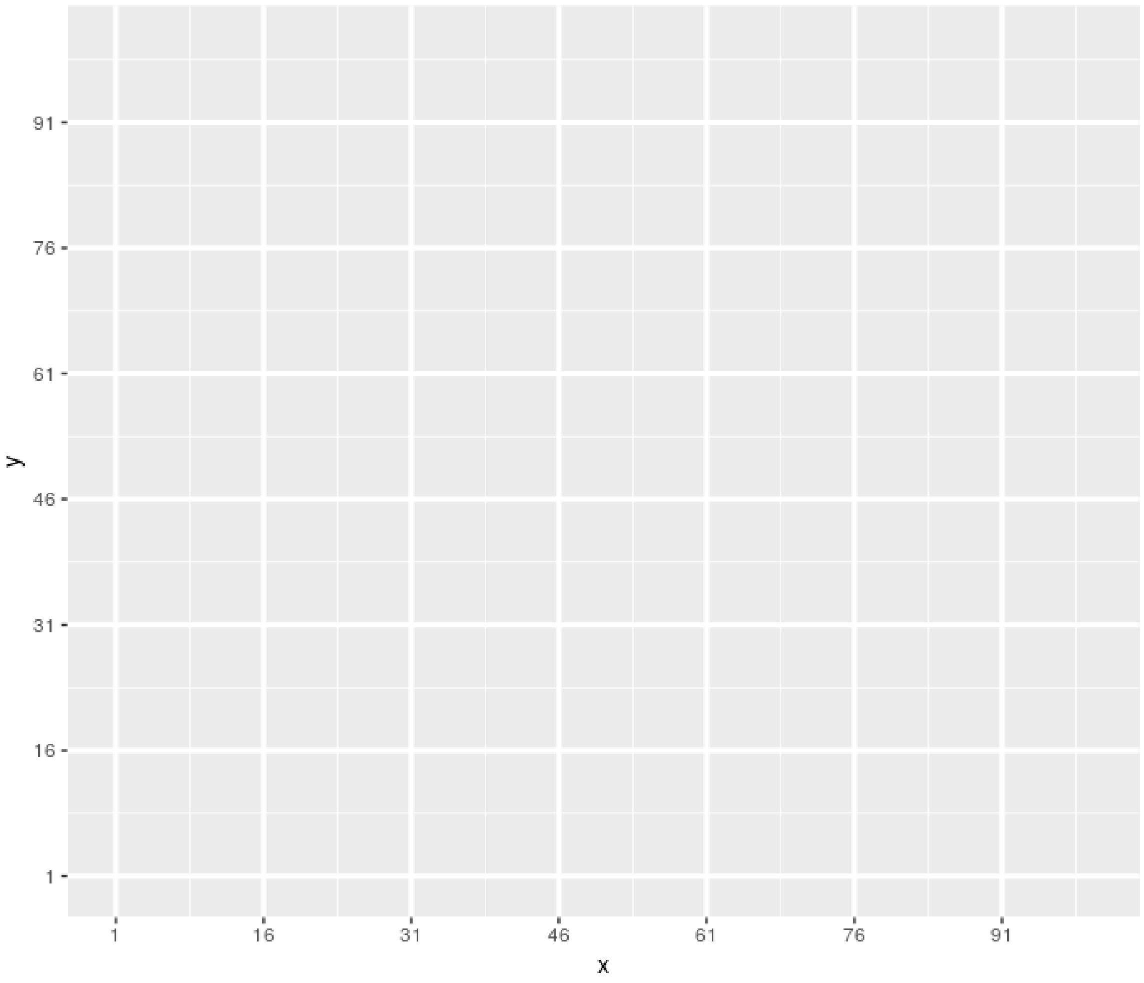
```
busy_bars + coord_flip() # swap x and y axes!
```

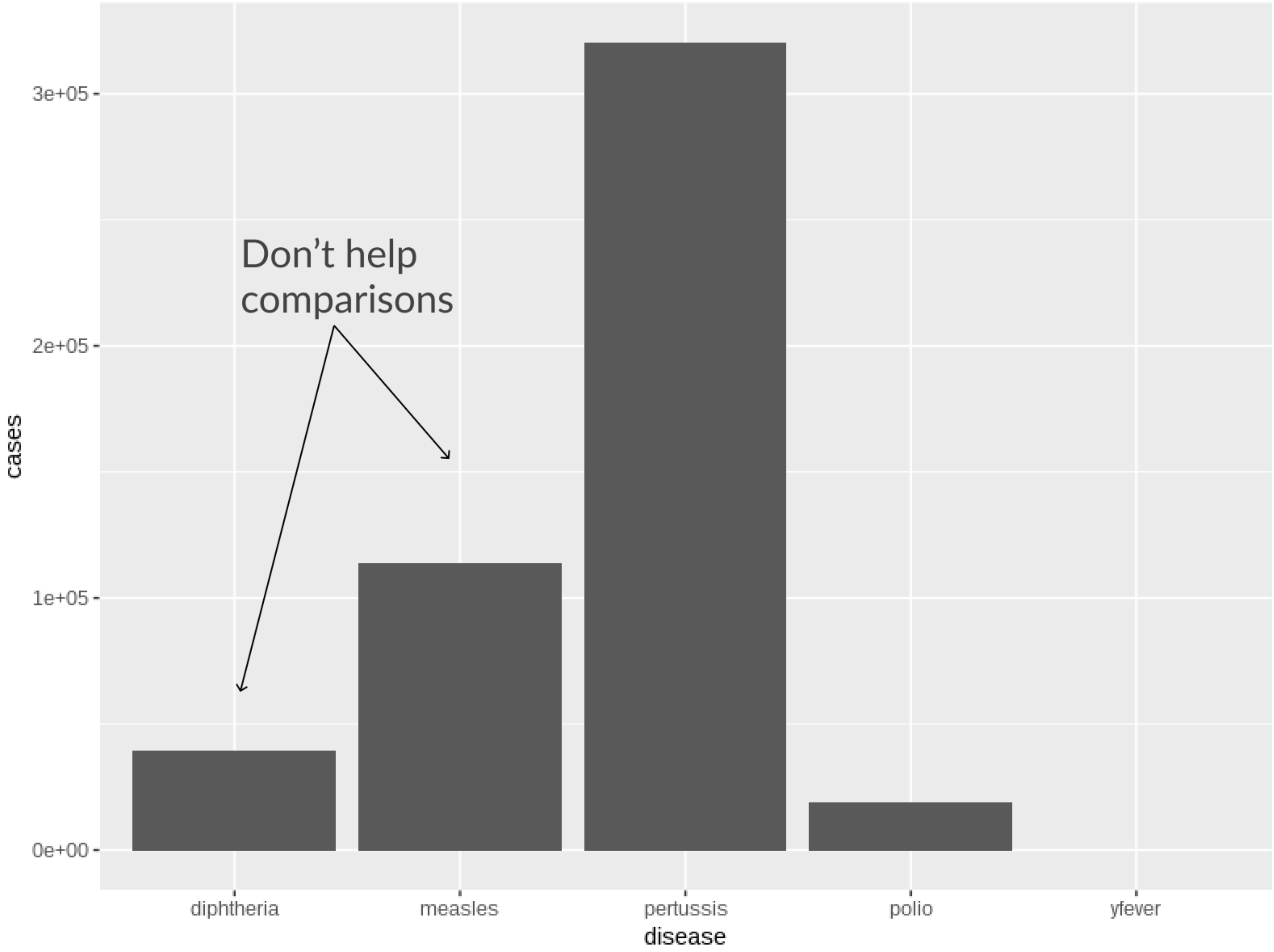

Yemen
United Arab Emirates (the)
Tunisia
Syrian Arab Republic (the)
Sudan (the)
Somalia
Saudi Arabia
Qatar
Pakistan
Oman
Morocco
Libya
Lebanon
Kuwait
Jordan
Iraq
Iran (Islamic Republic of)
Egypt
Djibouti
Bahrain
country

Much better!

Excess grid

- No need for parallel grid lines in bars
- In point charts, only grids in line with point locations are needed



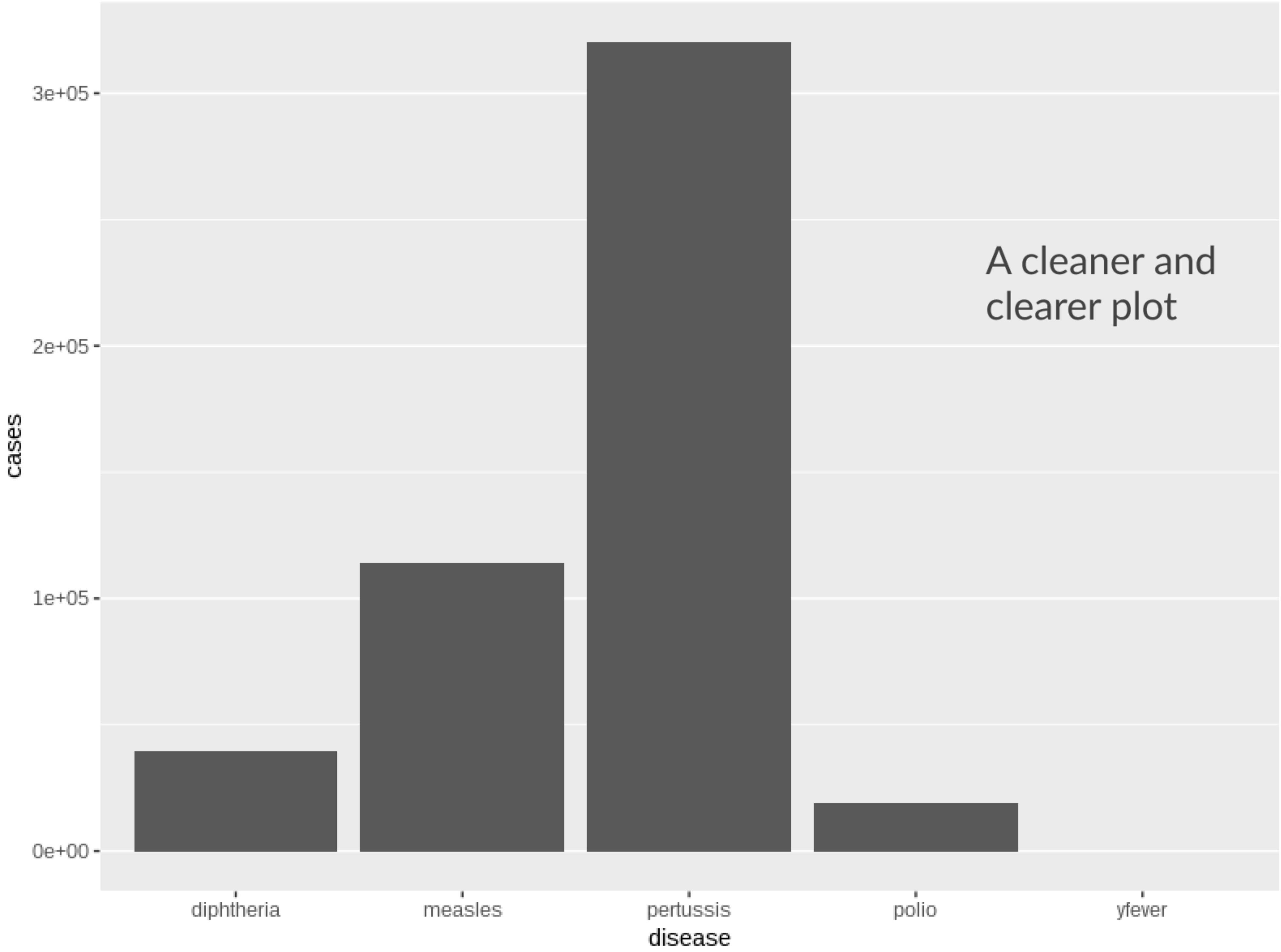


Don’t help comparisons

Removing vertical grid

```
plot <- who_disease %>%
  filter(country == "India", year == 1980) %>%
  ggplot(aes(x = disease, y = cases)) +
    geom_col()
```

```
# Remove vertical grid lines
 plot + theme(
    panel.grid.major.x = element_blank()
  )
```



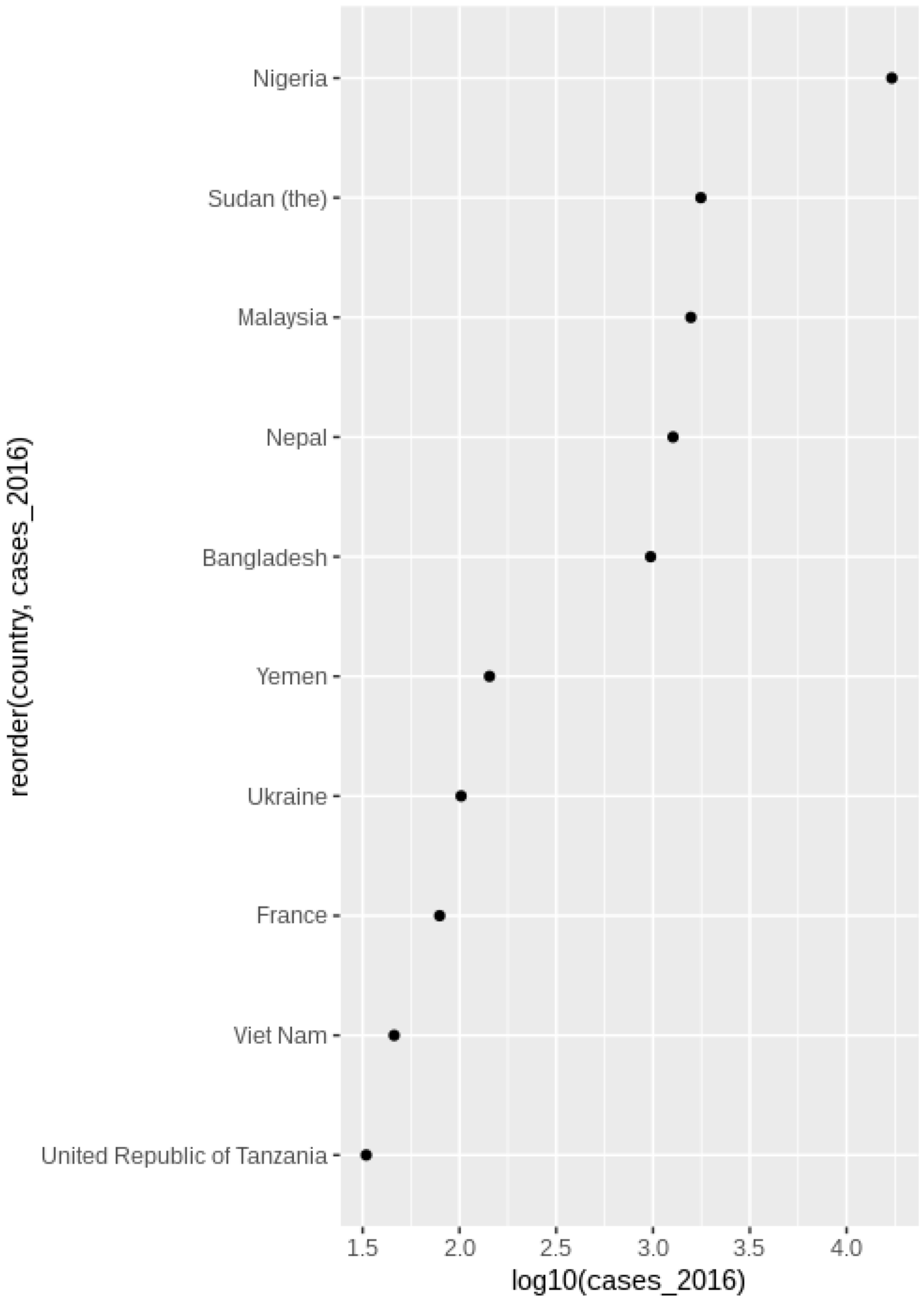
A cleaner and clearer plot

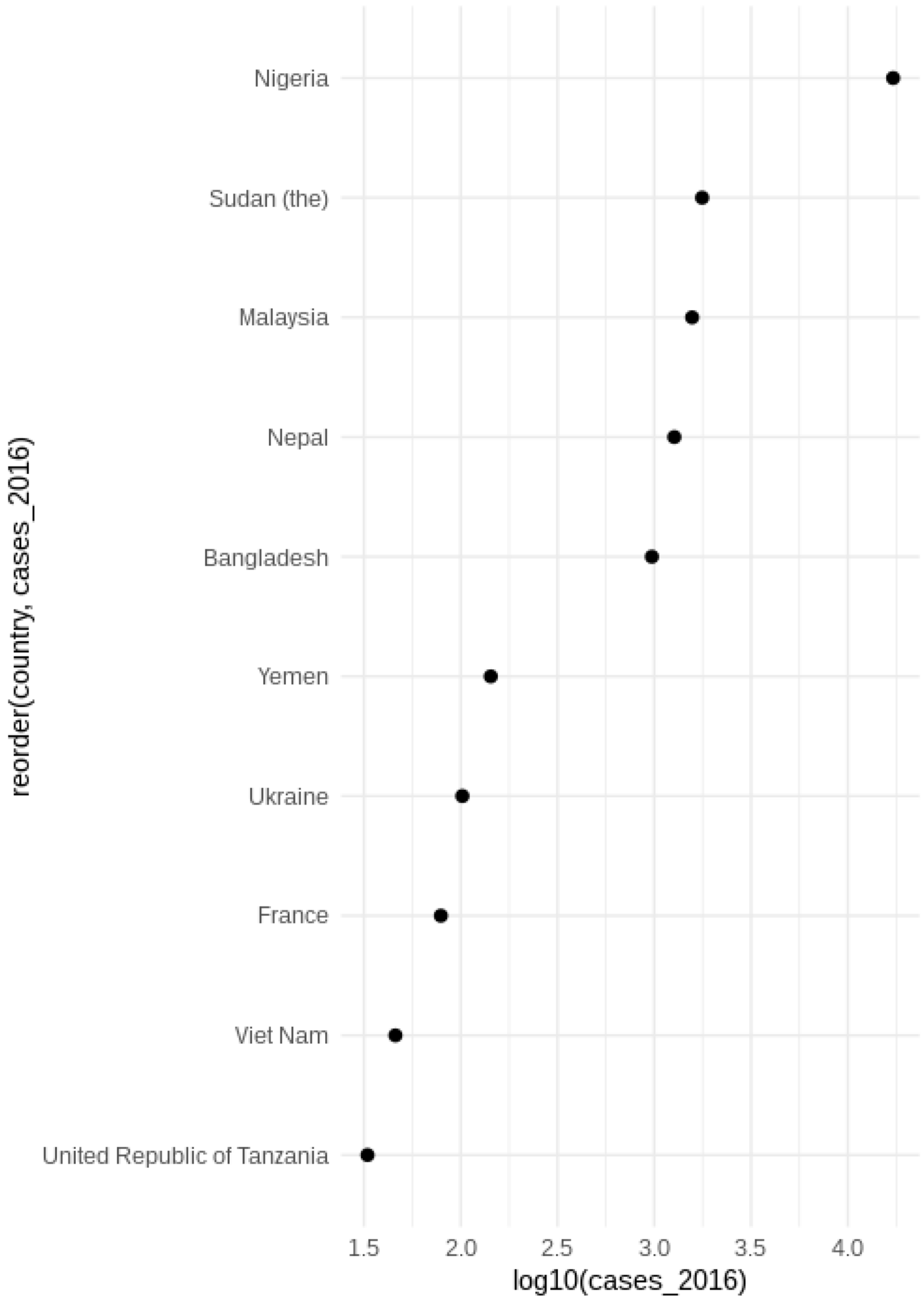
Lighter background for point charts

- Default gray background can be too low-contrast for points
- `theme_minimal()` is a quick fix
- Making points bigger helps too

```
who_subset %>%
  ggplot(aes(y = reorder(country, cases_2016), x = log10(cases_2016))) +
  # Point size increased
  geom_point(size = 2) +
  # Theme minimal for light background
  theme_minimal()
```

```
size=2 +
theme_minimal()
```





Let's try it out!

VISUALIZATION BEST PRACTICES IN R