Amounts

Week 6

AEM 2850: R for Business Analytics

Cornell Dyson

Spring 2022

Acknowledgements: Andrew Heiss, Claus Wilke

Announcements

I hope you had a restorative February break

Today we will cover slides and an example

- Next lab is due Monday

We will provide details on the first project in the next 0-1 weeks

Questions before we get started?

Plan for today

Prologue

Amounts

Plotting amounts using ggplot

example-06

Prologue

R Markdown in real life

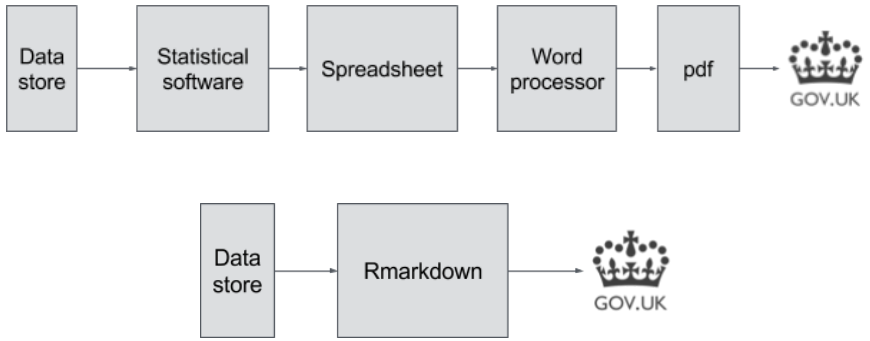
3.1.2 Data Visualization

We use ggplot2 as our main package to create ad-hoc exploratory graphics as well as polished-looking customized visualizations. When combined with tools to clean and transform data, ggplot2 allows analysts to quickly translate insights into high quality, compelling visualizations. In addition to the static graphics of `ggplot2`, we often make interactive visualizations or dashboards using R packages such as `plotly` (Sievert et al. 2017), `leaflet` (Cheng et al. 2017), `dygraphs` (Vanderkam et al. 2017), `DiagrammeR` (Sveidqvist et al. 2017), and `shiny` (Chang et al. 2017).

3.1.3 Reproducible Research

At Airbnb, all R analyses are documented in `rmarkdown`, where code and visualizations are combined within a single written report. Posts are carefully reviewed by experts in the content area and techniques used, both in terms of methodologies and code style, before publishing and sharing with the business partners. The peer review process is

Airbnb, ggplot, and rmarkdown

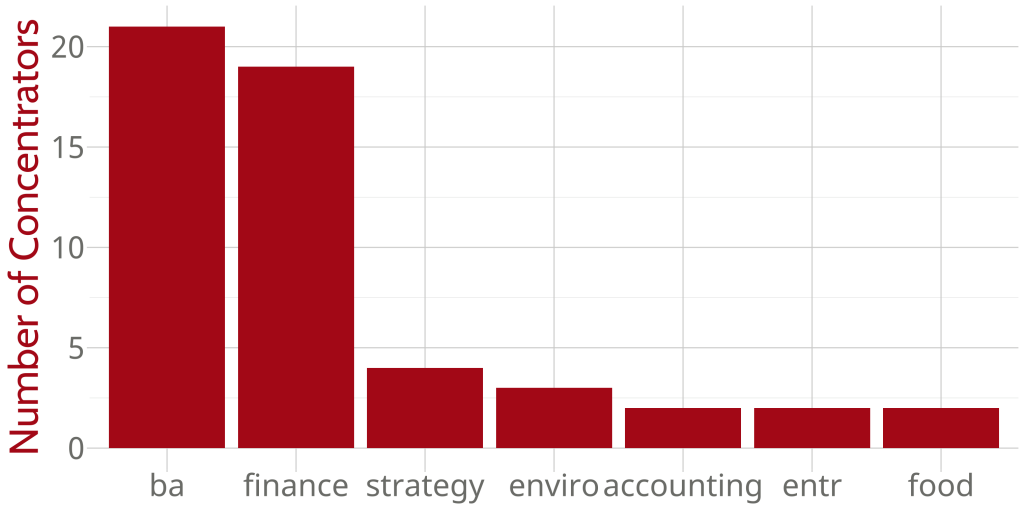


The UK's reproducible analysis pipeline

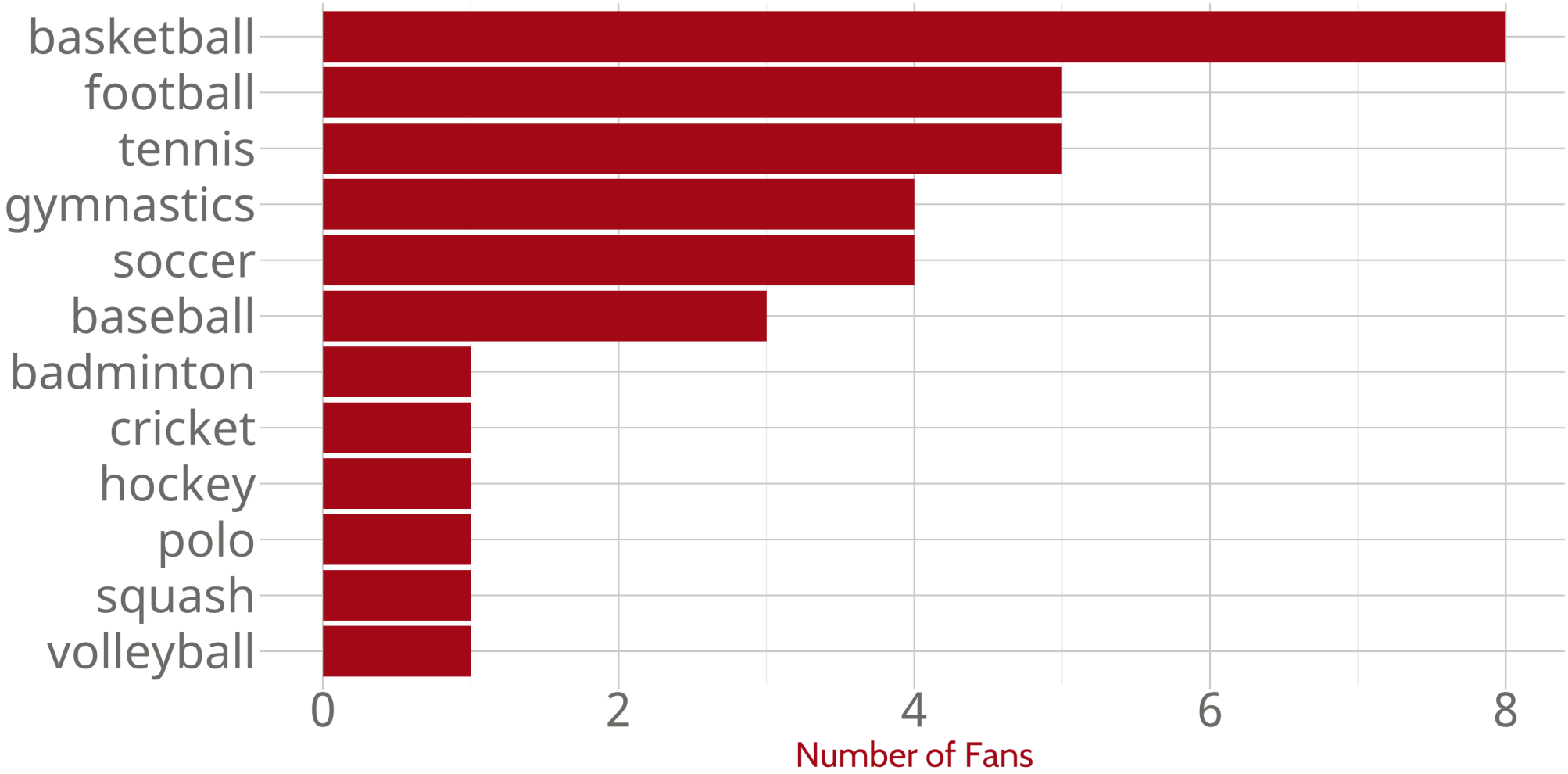
Remember our concentrations?

How might we visualize these data?

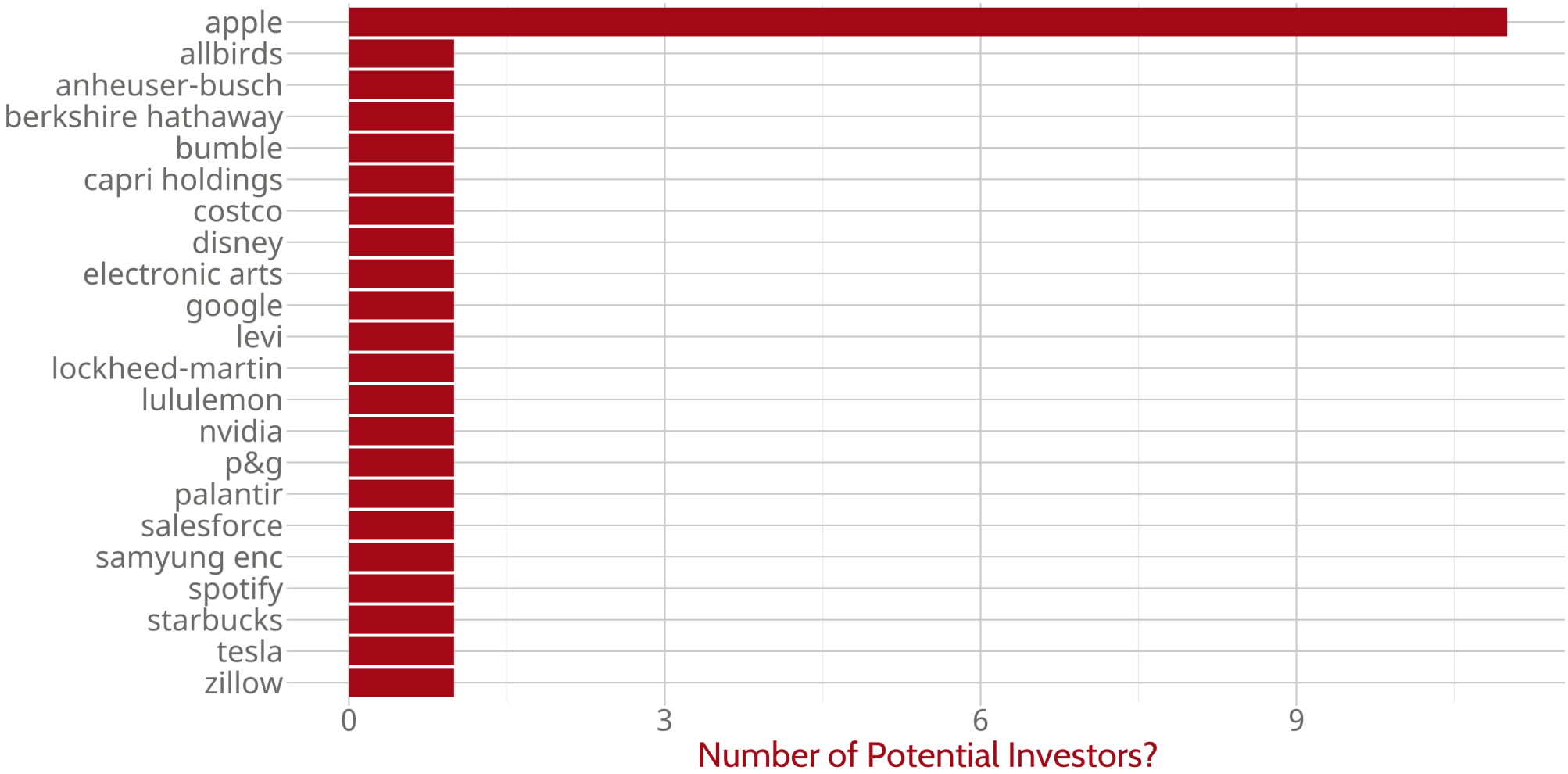
```
## # A tibble: 7 × 2
##   concentration count
##   <chr>         <int>
## 1 ba               21
## 2 finance          19
## 3 strategy          4
## 4 enviro            3
## 5 accounting        2
## 6 entr              2
## 7 food              2
```



We could do the same thing with sports



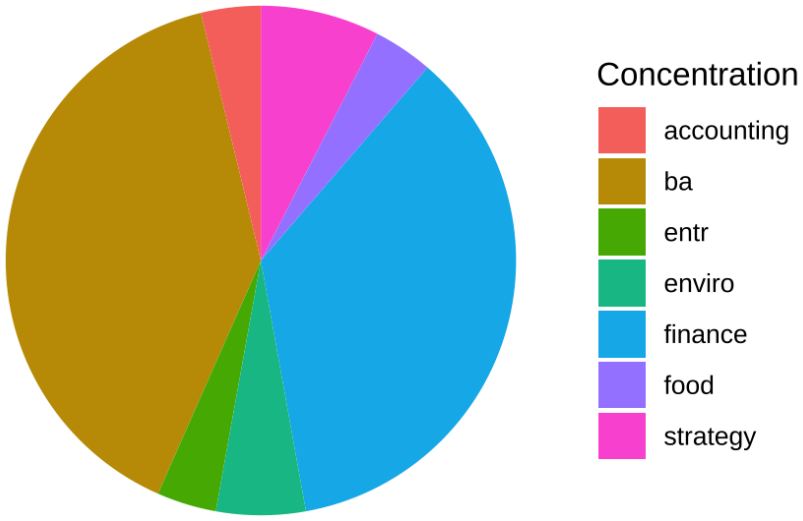
What are our favorite public companies?

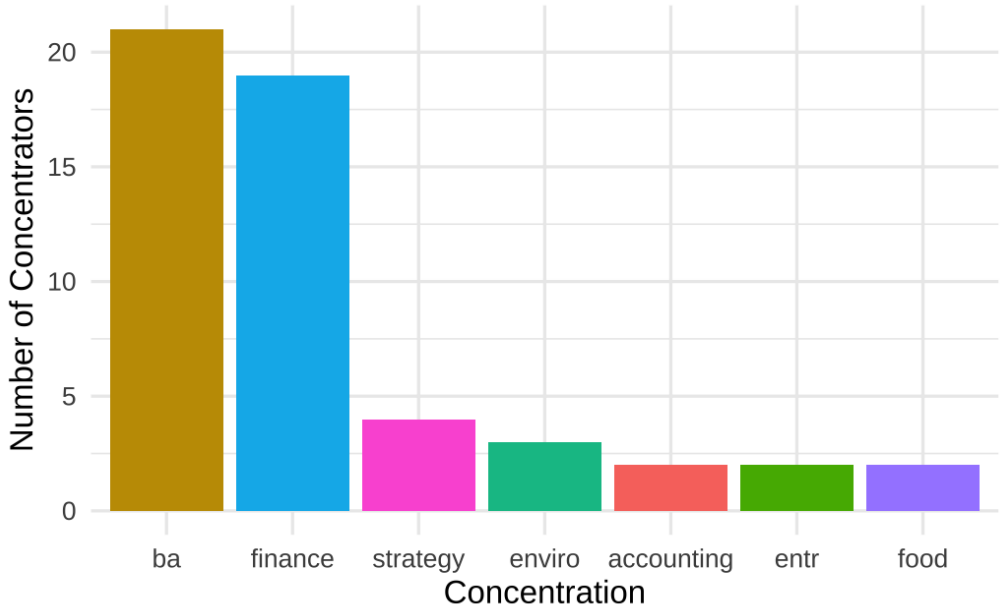


Amounts

Yay bar plots!

We are a lot better at visualizing line lengths than angles and areas

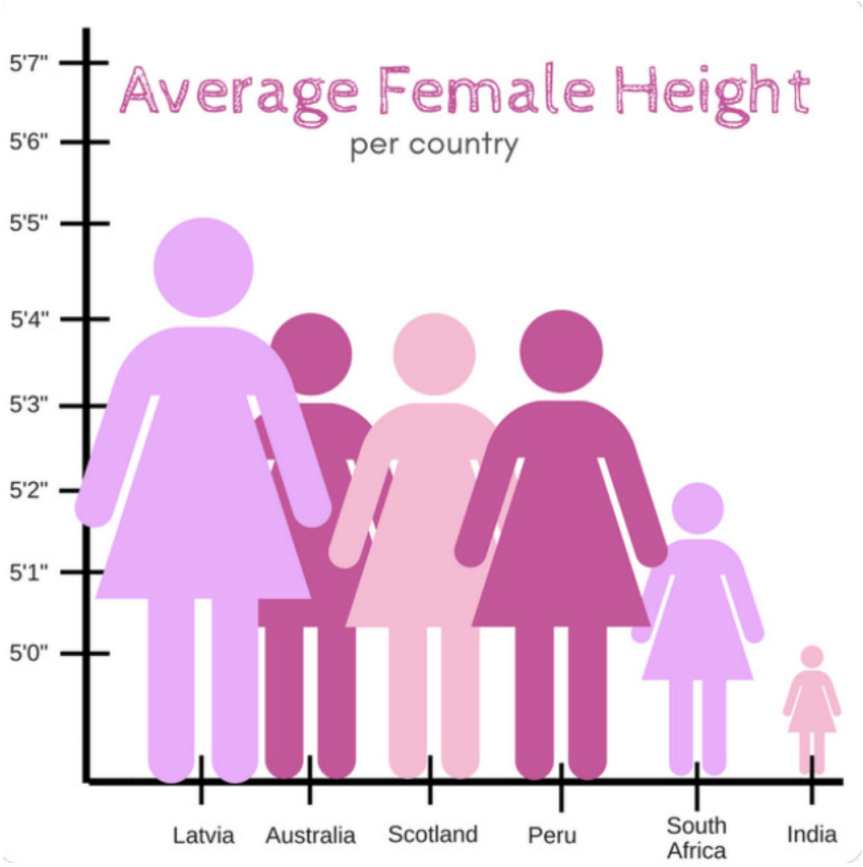




Oh no bar plots!

What went wrong?

What went wrong?

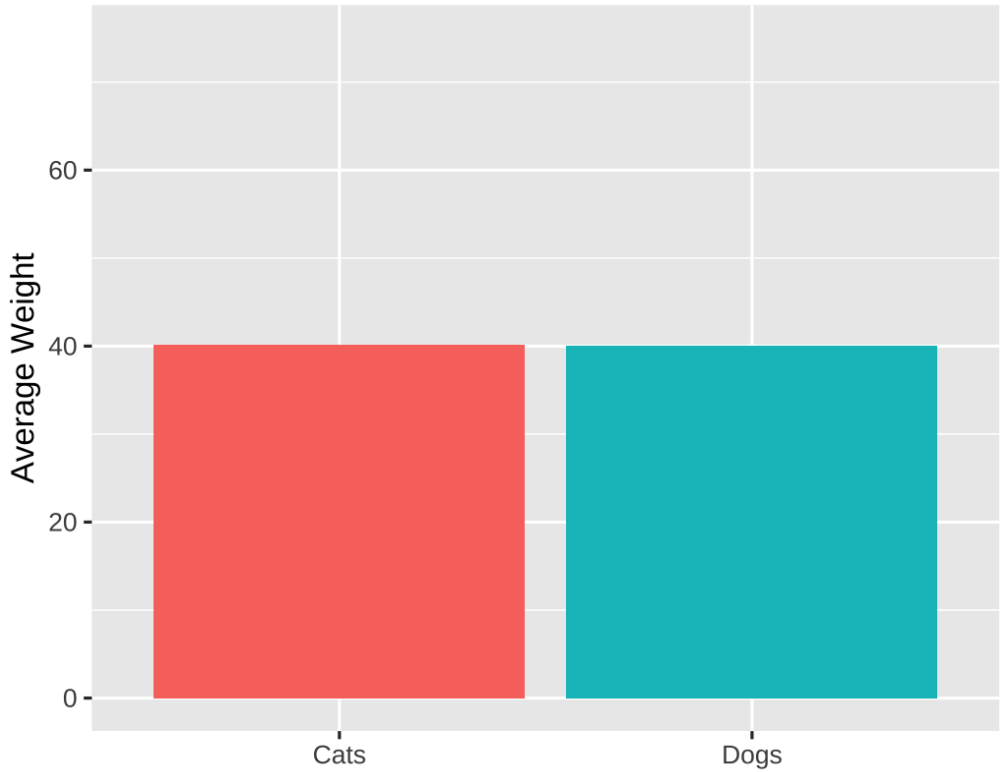


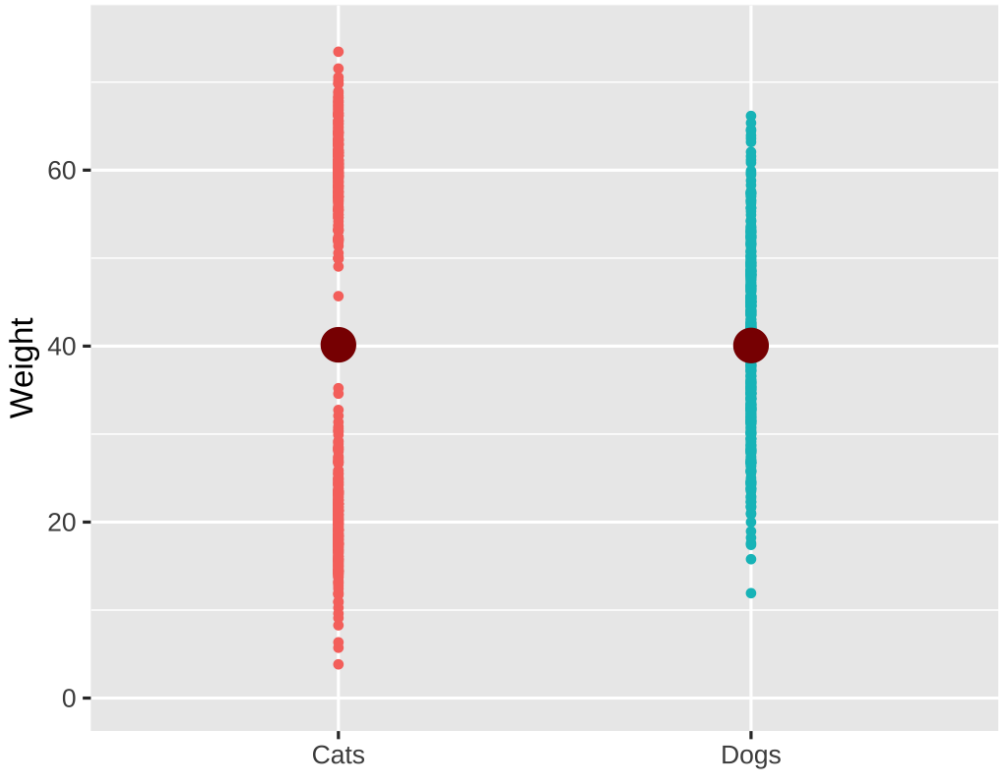
At least two problems:

1. truncated y axis
2. area scales faster than height

This terrible figure was brought to us by one of you, thanks!

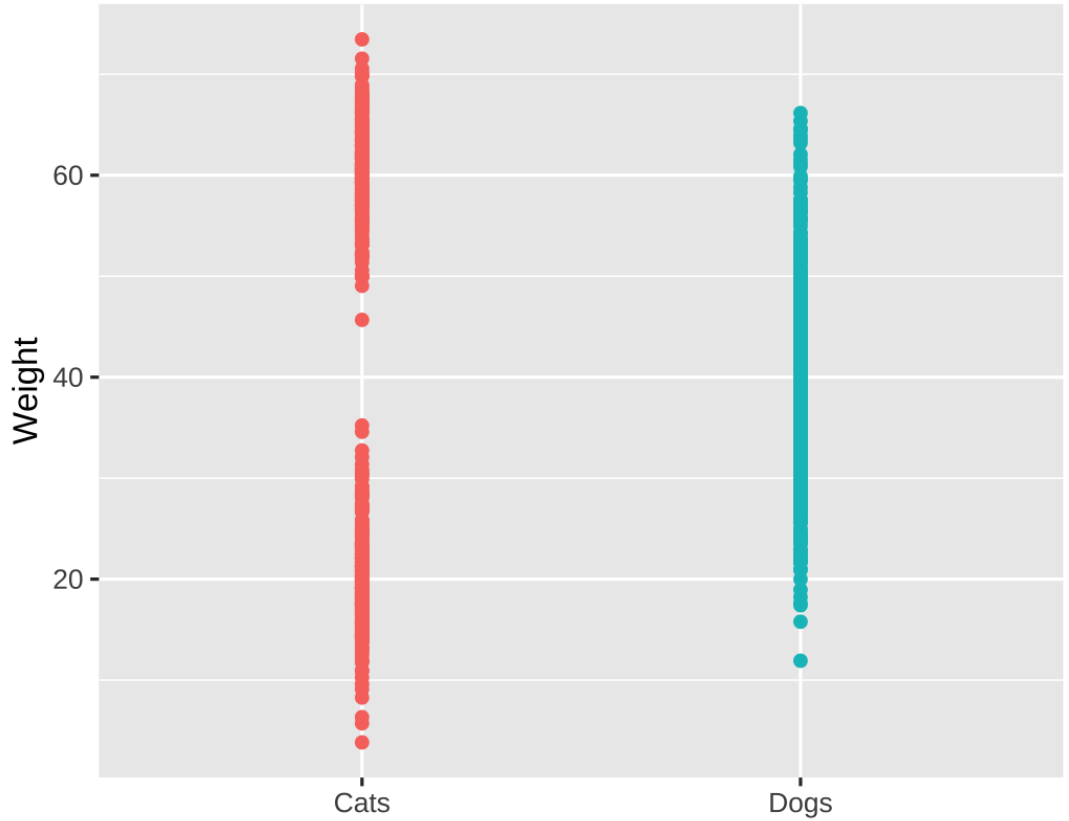
Bar plots and summary statistics





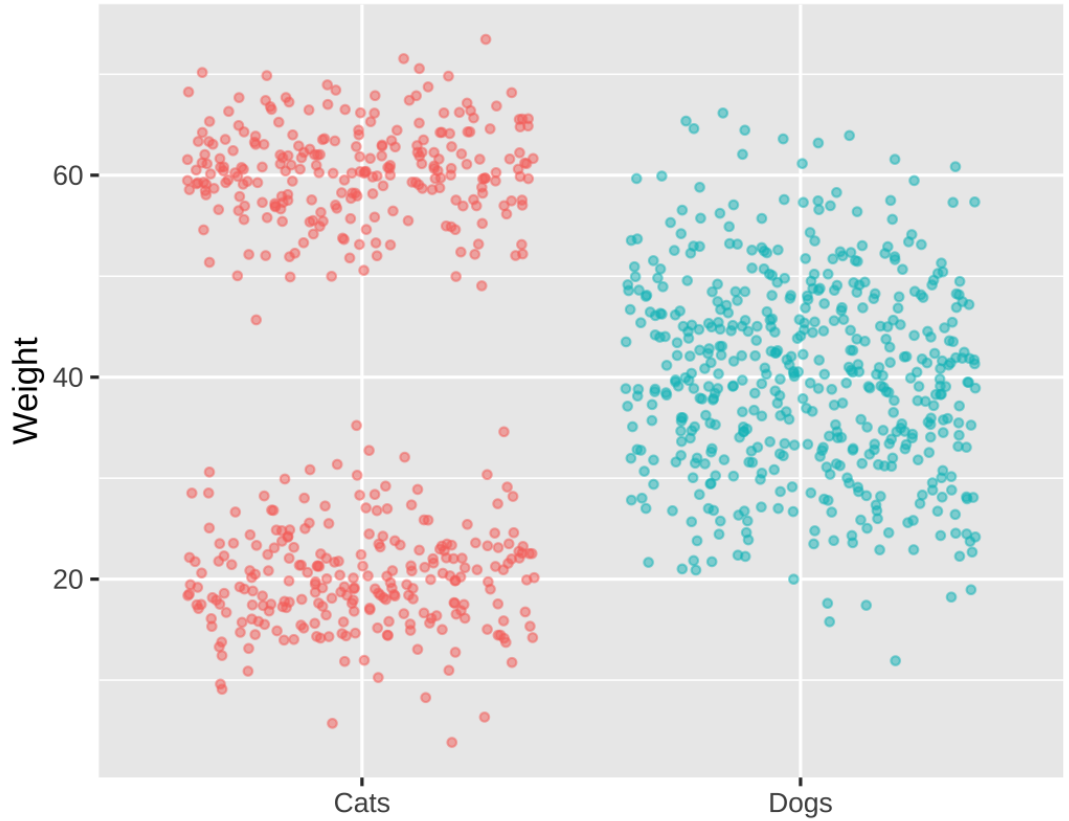
Show more data with strip plots

```
ggplot(animals,
       aes(x = animal_type,
           y = weight,
           color = animal_type)) +
  geom_point() +
  labs(x = NULL, y = "Weight") +
  guides(color = "none")
```



Jittering and transparency can also help

```
ggplot(animals,
       aes(x = animal_type,
           y = weight,
           color = animal_type)) +
  geom_point(position = position_jitter(),
             size = 1,
             alpha = 0.5) +
  labs(x = NULL, y = "Weight") +
  guides(color = "none")
```



General rules for bar charts

Useful when the length of the bar is all that matters

Bar charts should always start at zero

- Or: don't use bars!

Don't use bars for summary statistics. You throw away too much information.

- We will come back to visualizing distributions / uncertainty

Plotting amounts using ggplot

Plotting amounts using ggplot

We'll use a summarized version of the gapminder dataset for examples

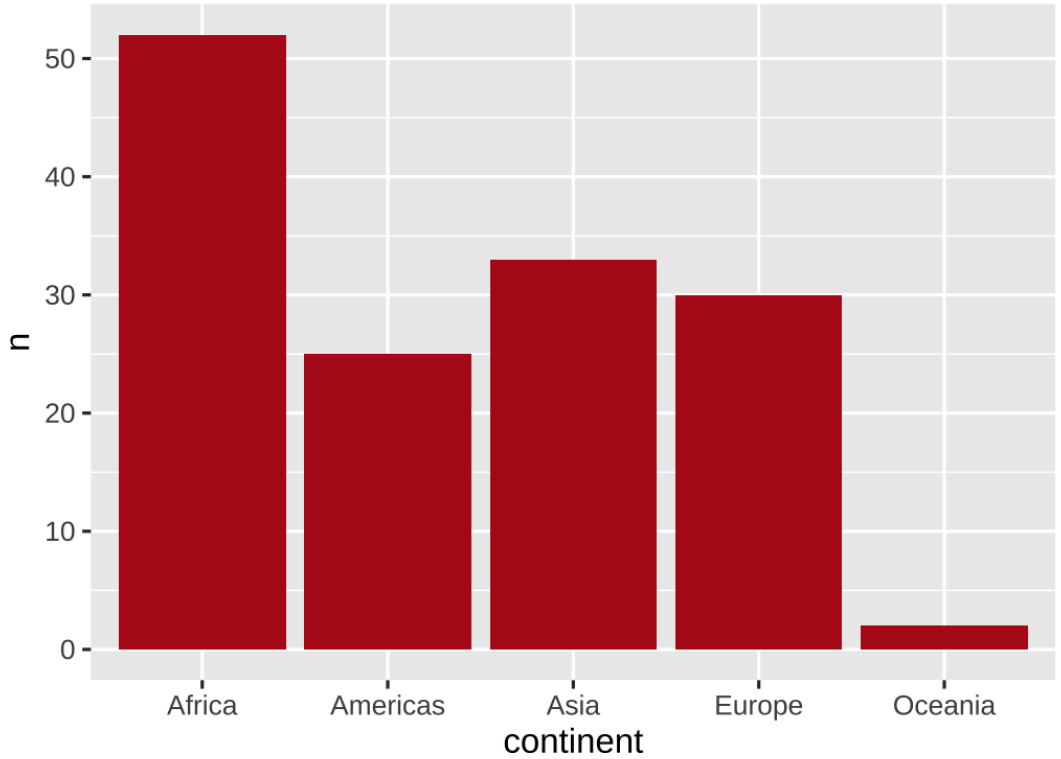
```
library(gapminder)
gapminder_continents <- gapminder %>%
  filter(year == 2007) %>%  # only look at 2007
  count(continent) %>%  # get a count of continents
  arrange(desc(n))  # sort by count, descending
gapminder_continents
```

```
## # A tibble: 5 × 2
##   continent     n
##   <fct>     <int>
## 1 Africa       52
## 2 Asia         33
## 3 Europe       30
## 4 Americas     25
## 5 Oceania       2
```

Start with a simple bar plot

```
ggplot(data = gapminder_continents,
       aes(x = continent, # map continent to x
           y = n)) + # map n (num countries) to y
  geom_col() # add bars
```

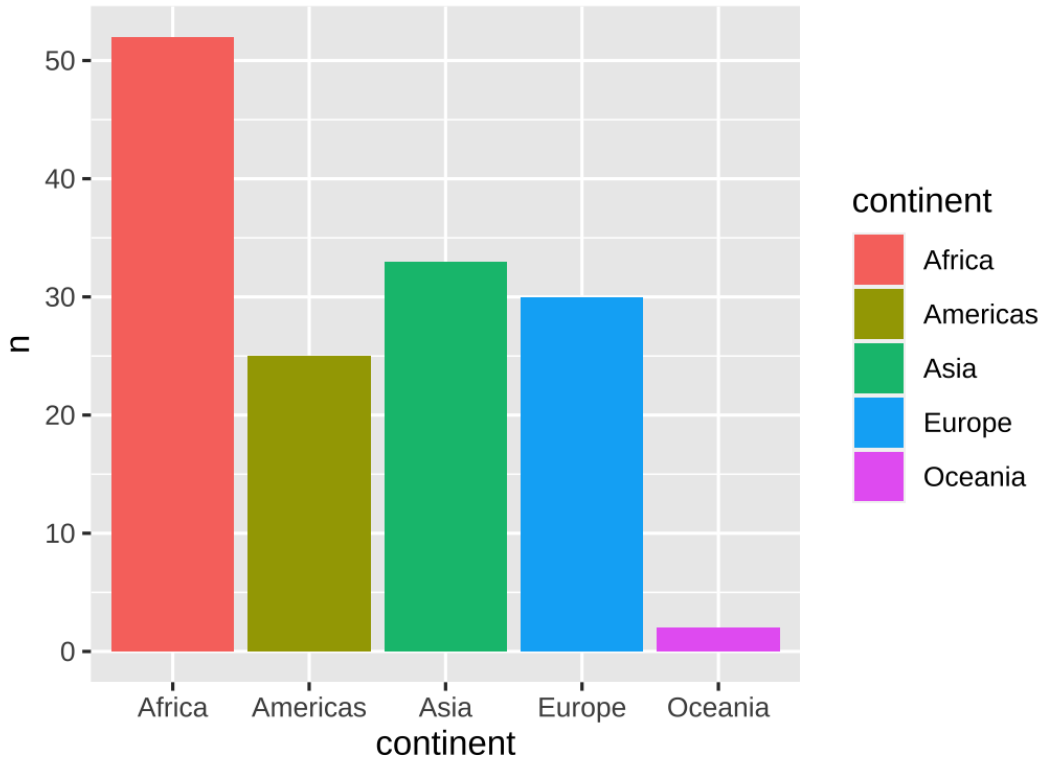
How could we improve this?



Add some color

```
ggplot(gapminder_continents,
       aes(x = continent, y = n,
           fill = continent)) + # color bars
  geom_col()
```

Do we need the fill legend?

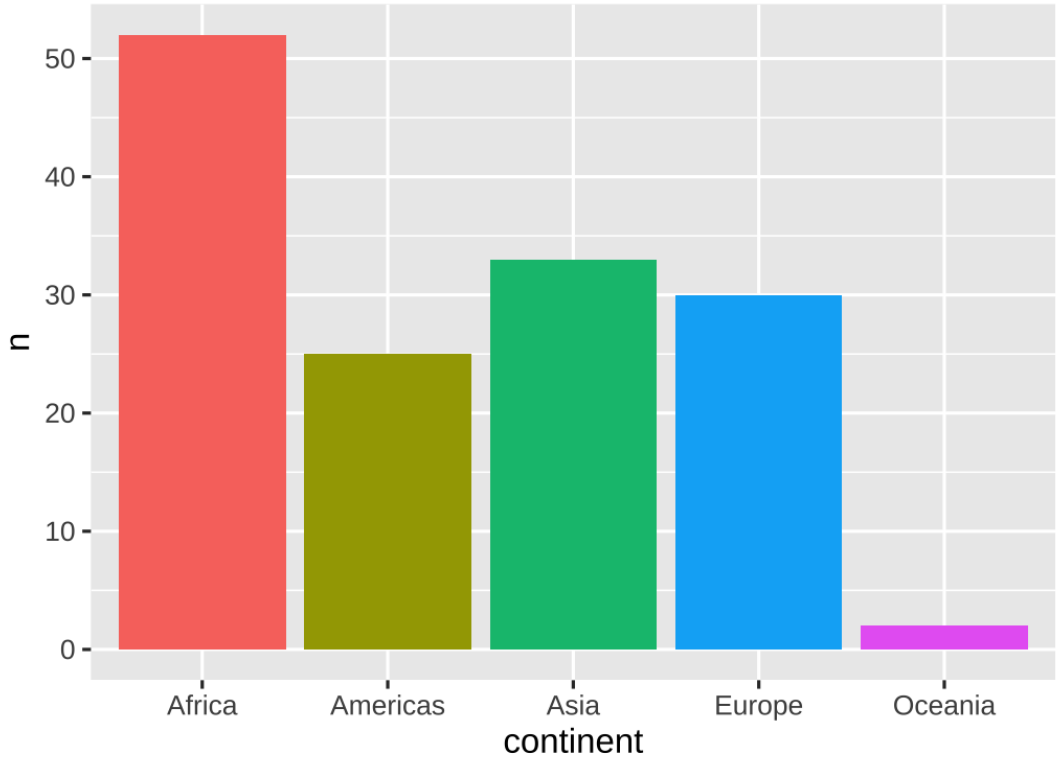


Add some color

```
ggplot(gapminder_continents,
       aes(x = continent, y = n,
           fill = continent)) +
  geom_col() +
  guides(fill = "none") # omit fill legend
```

Is "n" a good axis title?

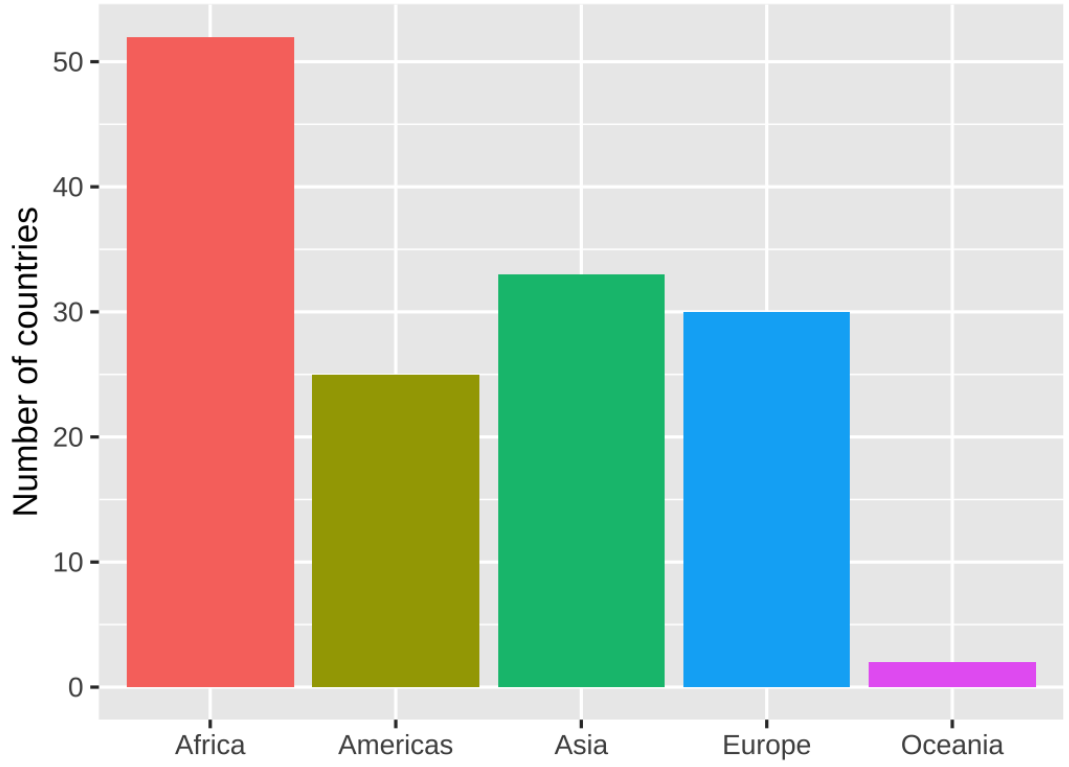
Do we need "continent" at all?



Add some labels

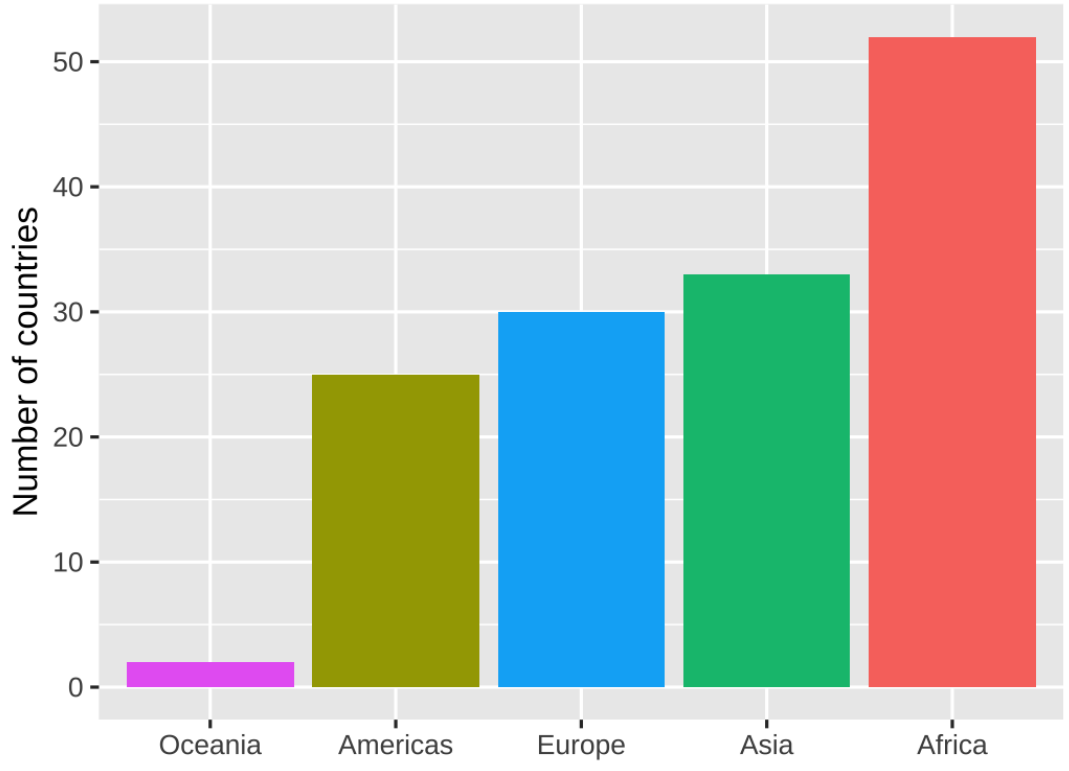
```
ggplot(gapminder_continents,
       aes(x = continent, y = n,
           fill = continent)) +
  geom_col() +
  guides(fill = "none") +
  labs(x = NULL, y = "Number of countries")
```

Is alphabetical the best ordering?



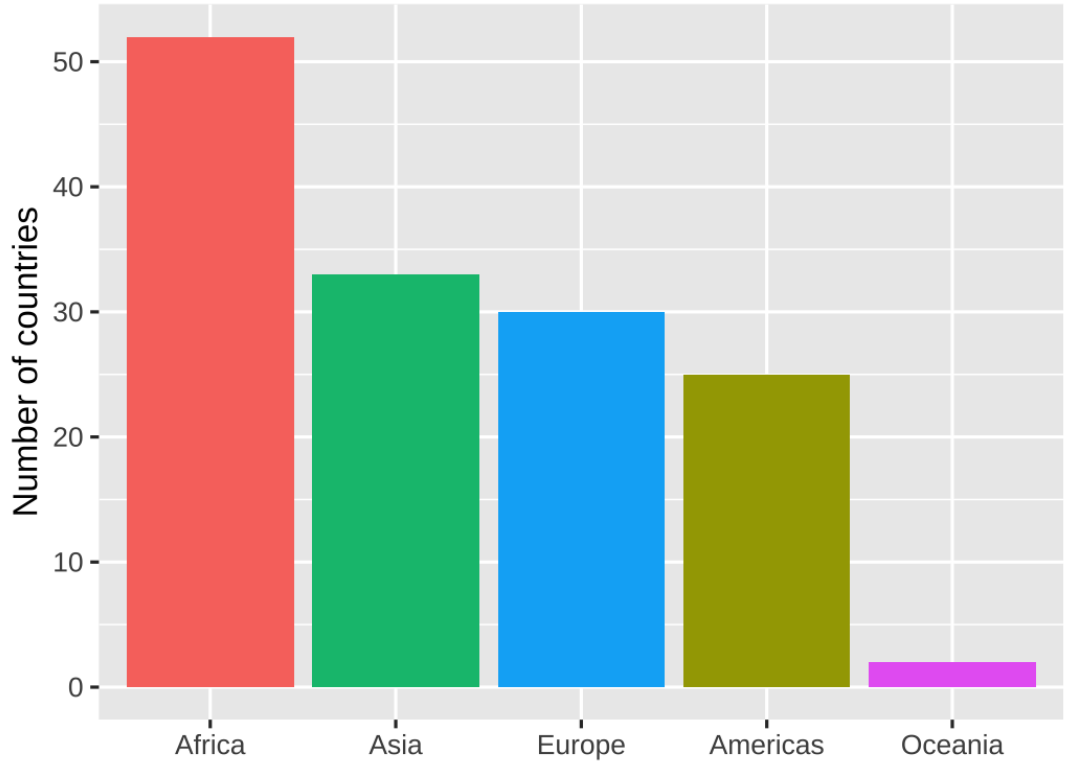
Order by data value

```
ggplot(gapminder_continents,
       aes(x = fct_reorder(continent, n),
           y = n, fill = continent)) +
  geom_col() +
  guides(fill = "none") +
  labs(x = NULL, y = "Number of countries")
```



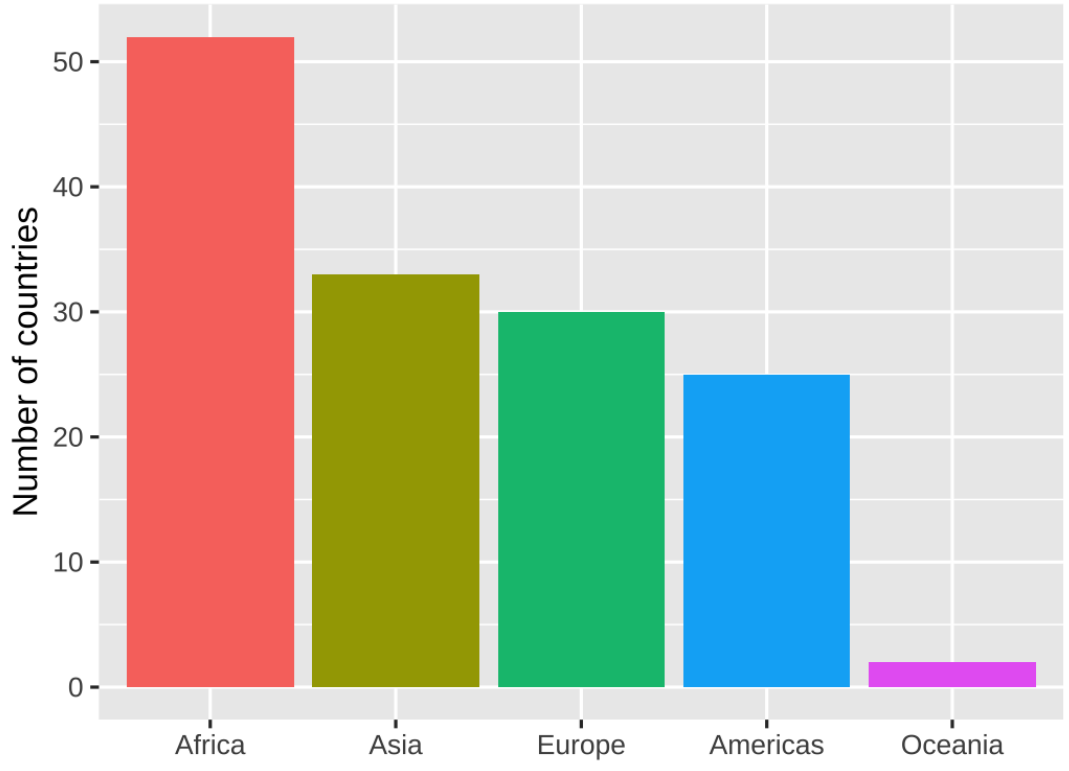
Order by data value, descending

```
ggplot(gapminder_continents,
       aes(x = fct_reorder(continent, -n),
           y = n, fill = continent)) +
  geom_col() +
  guides(fill = "none") +
  labs(x = NULL, y = "Number of countries")
```



Another option is to encode order in the data

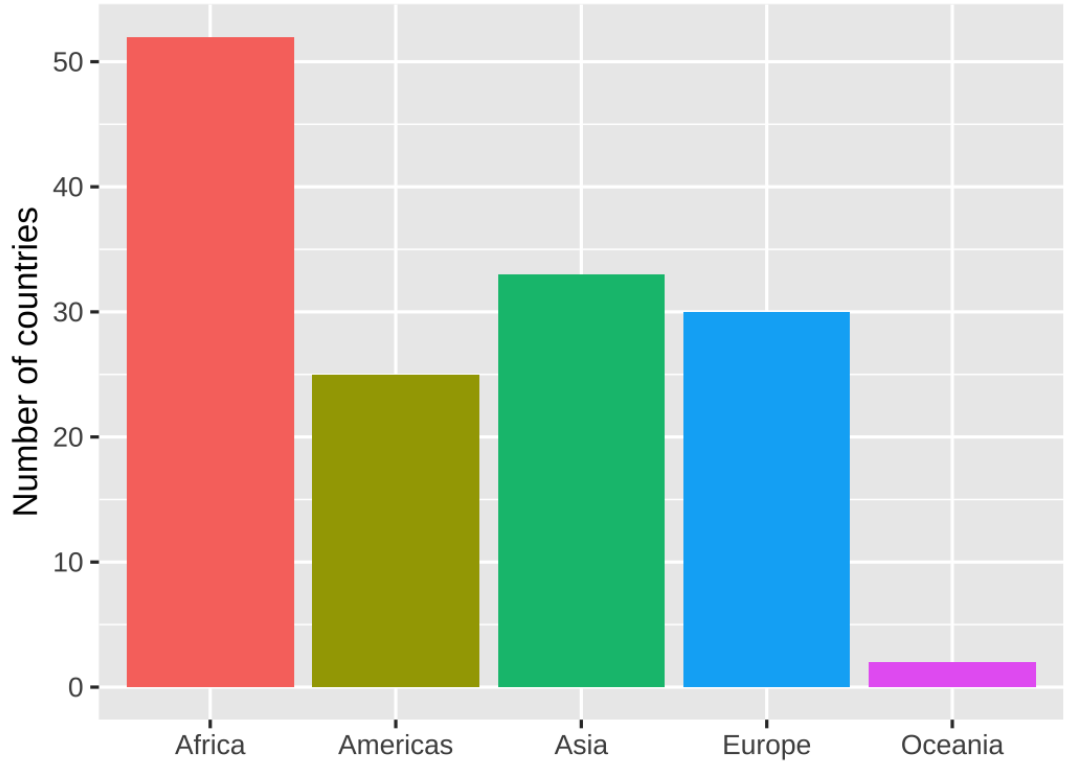
```
# use `fct_inorder` to order levels of continent
# based on how they appear in the data
gapminder_ordered <- gapminder_continents %>%
  arrange(desc(n)) %>% # sort by count, descending
  mutate(continent = fct_inorder(continent))
ggplot(gapminder_ordered,
       aes(x = continent,
           y = n, fill = continent)) +
  geom_col() +
  guides(fill = "none") +
  labs(x = NULL, y = "Number of countries")
```



Wait, what about geom_bar?

Use `geom_bar` to count and plot in one step

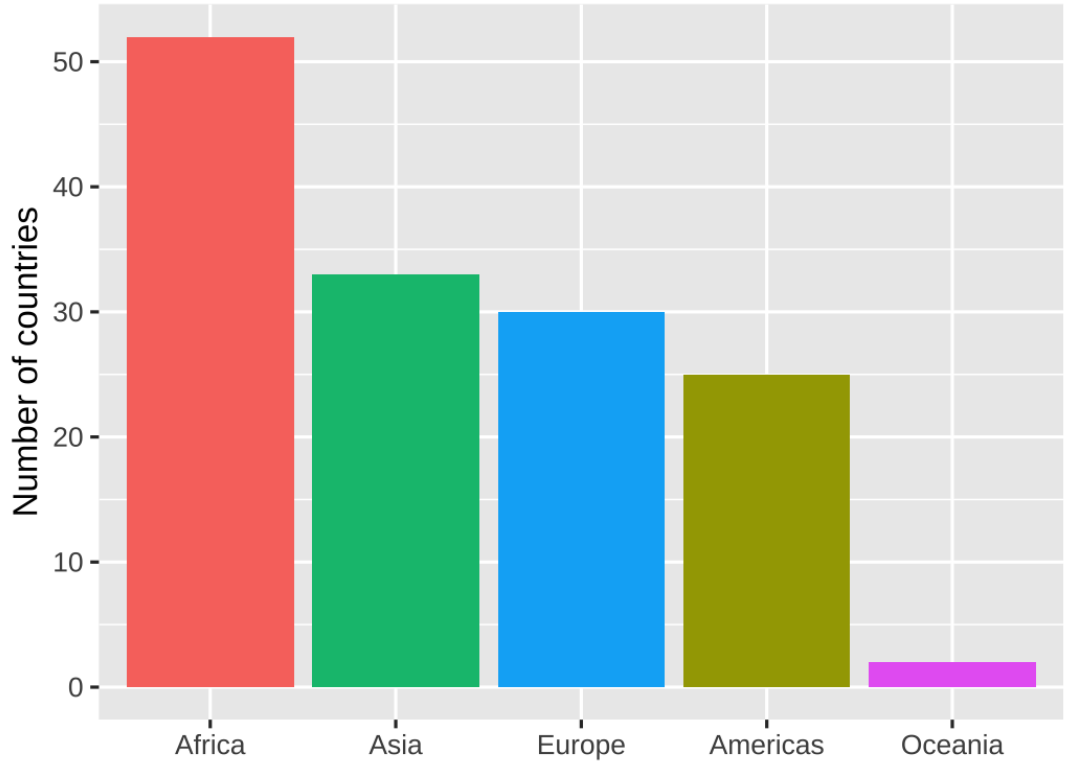
```
gapminder %>%
  filter(year == 2007) %>%
  ggplot(aes(x = continent, # note: no y aesthetic
             fill = continent)) +
  geom_bar() +
  guides(fill = "none") +
  labs(x = NULL, y = "Number of countries")
```



Wait, what about geom_bar?

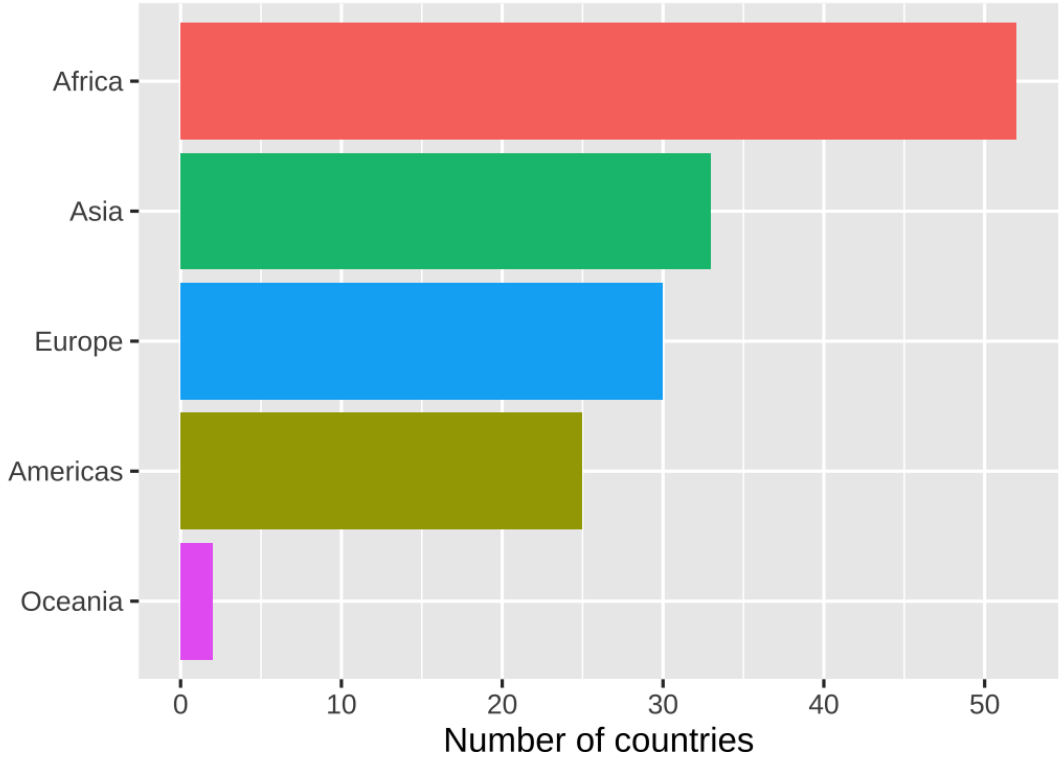
Here we can reorder by frequency using `fct_infreq`

```
gapminder %>%
  filter(year == 2007) %>%
  ggplot(aes(x = fct_infreq(continent),
             fill = continent)) +
  geom_bar() +
  guides(fill = "none") +
  labs(x = NULL, y = "Number of countries")
```



We can also flip geom_col/bar axes

```
gapminder %>%
  filter(year == 2007) %>%
  ggplot(aes(y = fct_rev(fct_infreq(continent)),
             fill = continent)) +
  geom_bar() +
  guides(fill = "none") +
  labs(x = "Number of countries", y = NULL)
```



Grouped bar charts

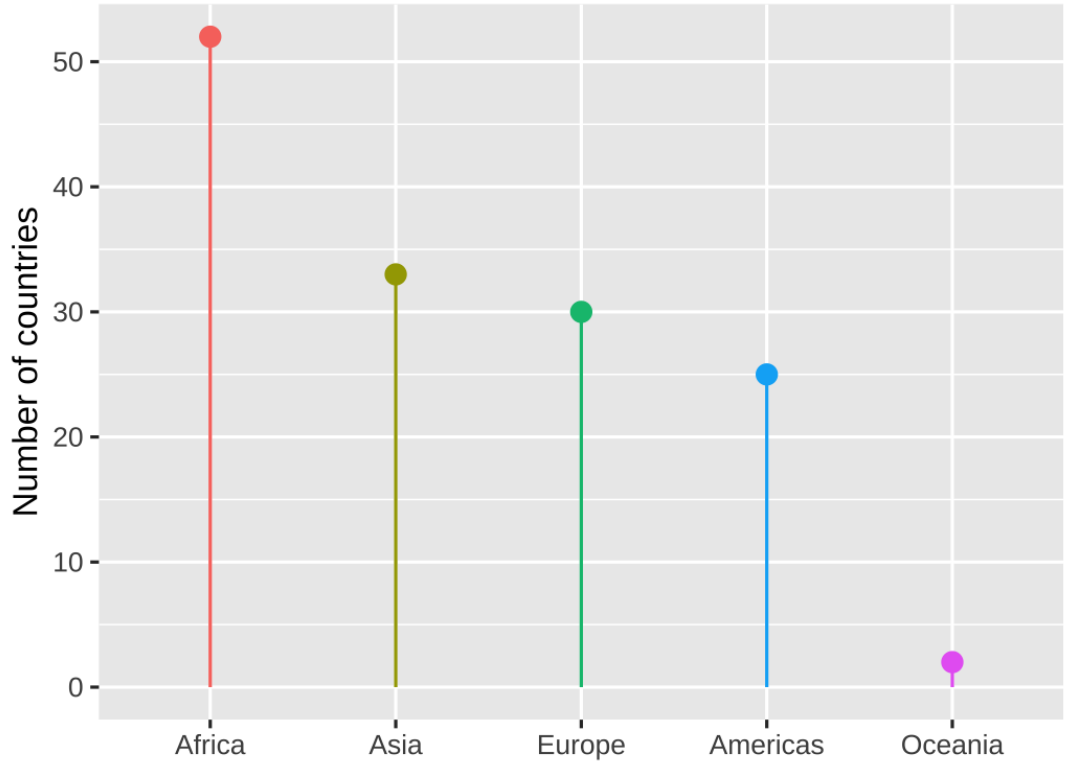
Use grouped bars for higher dimensional datasets

Facets are another option we saw last week

Alternatives: Lollipop charts

Since the end of the bar is important, emphasize it the most

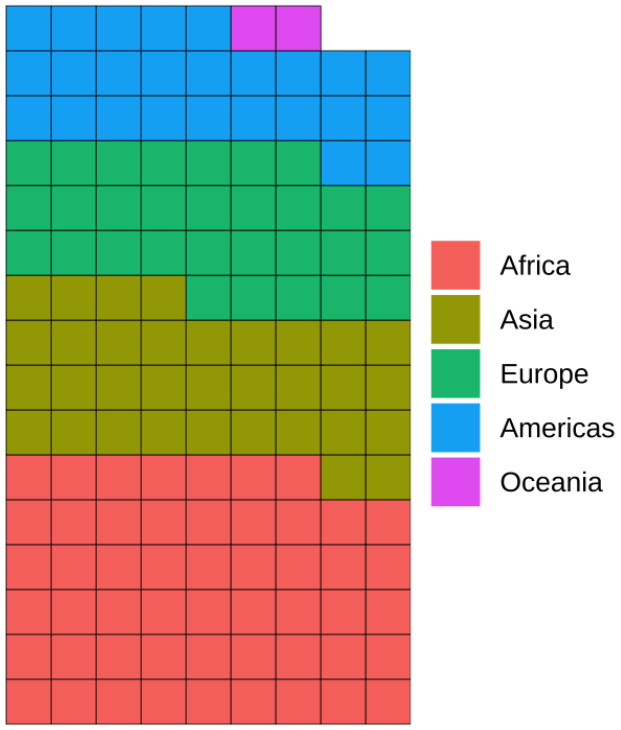
```
ggplot(gapminder_ordered,
       aes(x = continent, y = n,
           color = continent)) +
  geom_pointrange(aes(ymin = 0, ymax = n)) +
  guides(color = "none") +
  labs(x = NULL, y = "Number of countries")
```



Alternatives: Waffle charts

Show individual observations as squares

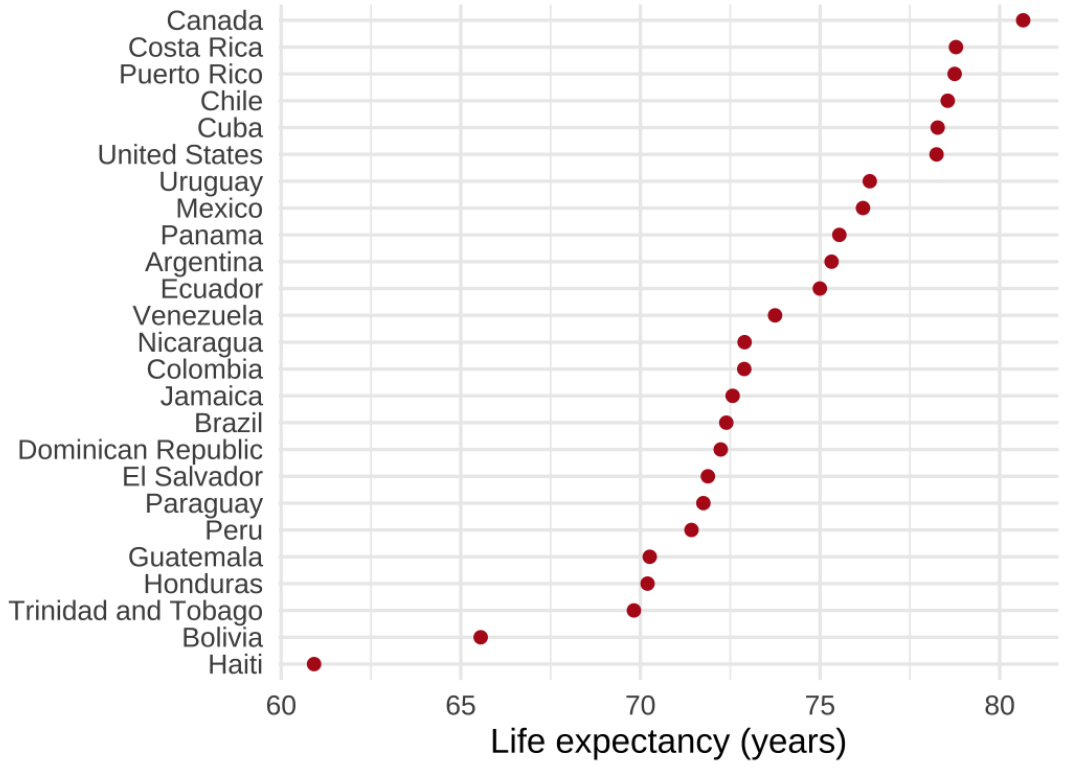
```
# This has to be installed in a special way
# install.packages("devtools") # do this once
# devtools::install_github("hrbrmstr/waffle") # do
library(waffle) # for geom_waffle

ggplot(gapminder_ordered,
       aes(x = continent, y = n,
           fill = continent)) +
  geom_waffle(aes(values = n),  # waffle aesthetic
              n_rows = 9,  # waffle options
              flip = TRUE) +
  labs(fill = NULL) +
  coord_equal() +  # make all the squares square
  theme_void()  # use a completely empty theme
```



Alternatives: Dots instead of bars

Dots are preferable if we want to truncate the axes

```
gapminder %>%
  filter(year == 2007, continent == "Americas") %>%
  ggplot(aes(x = lifeExp,
             y = fct_reorder(country, lifeExp))) +
  geom_point() +
  guides(color = "none") +
  labs(x = "Life expectancy (years)", y = NULL) +
  theme_minimal()
```



example-06