Proportions and distributions

Week 8

AEM 2850 / 5850 : R for Business Analytics

Cornell Dyson

Spring 2025

Acknowledgements: Andrew Heiss, Claus Wilke

Announcements

Prelim 1 grades are posted on canvas

Questions about your grade? Please:

1. Meet with Victor first for clarification regarding any grade deductions
2. If you have further questions, schedule a meeting with me
3. Email me if the available meeting times do not work for you

Prelim 2 may be more difficult, may be a different format, or both

We will provide details on the group project in the next 1-2 weeks

Questions before we get started?

Plan for this week

Tuesday

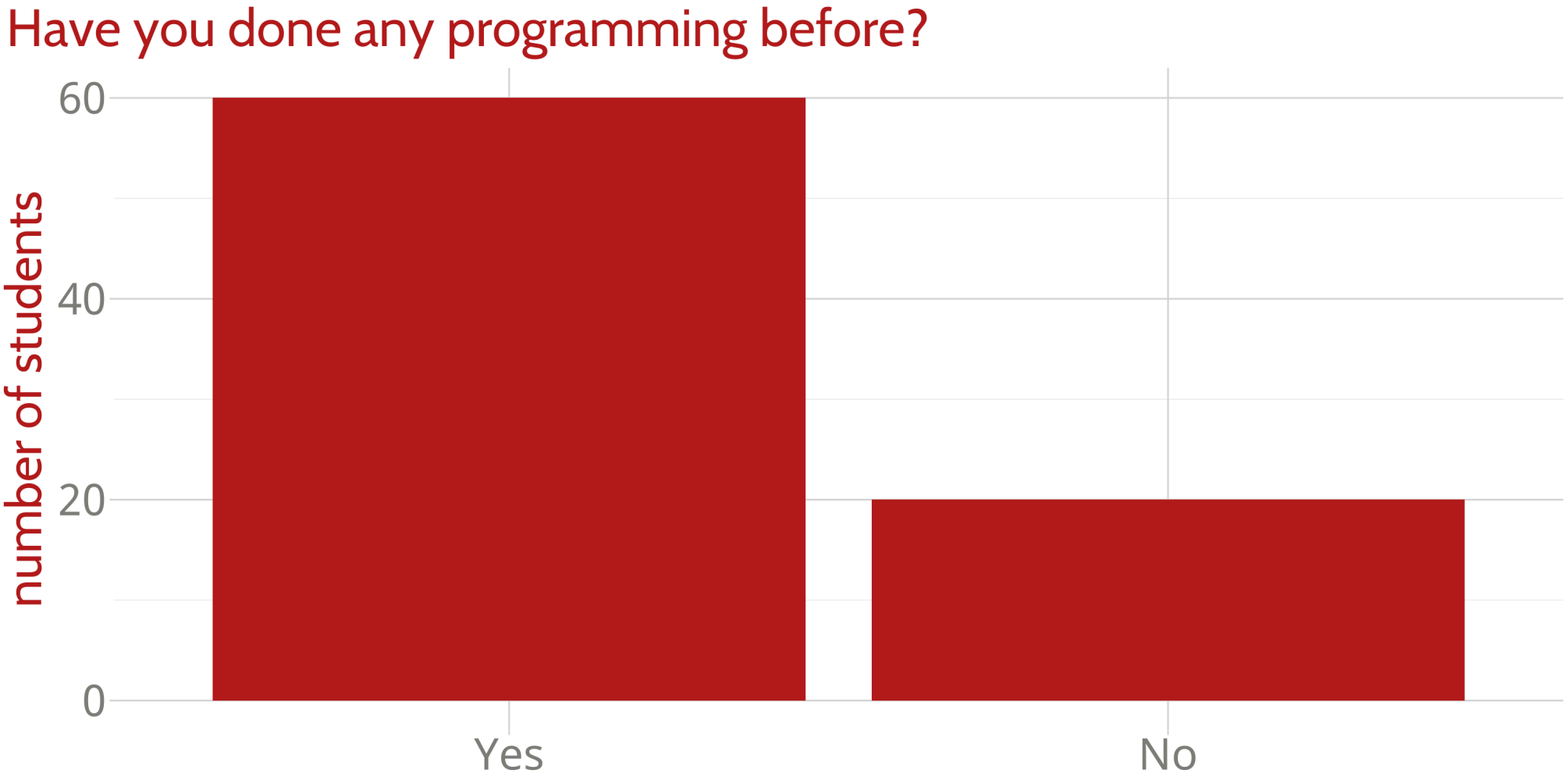
- Proportions
- example-08-1

Thursday

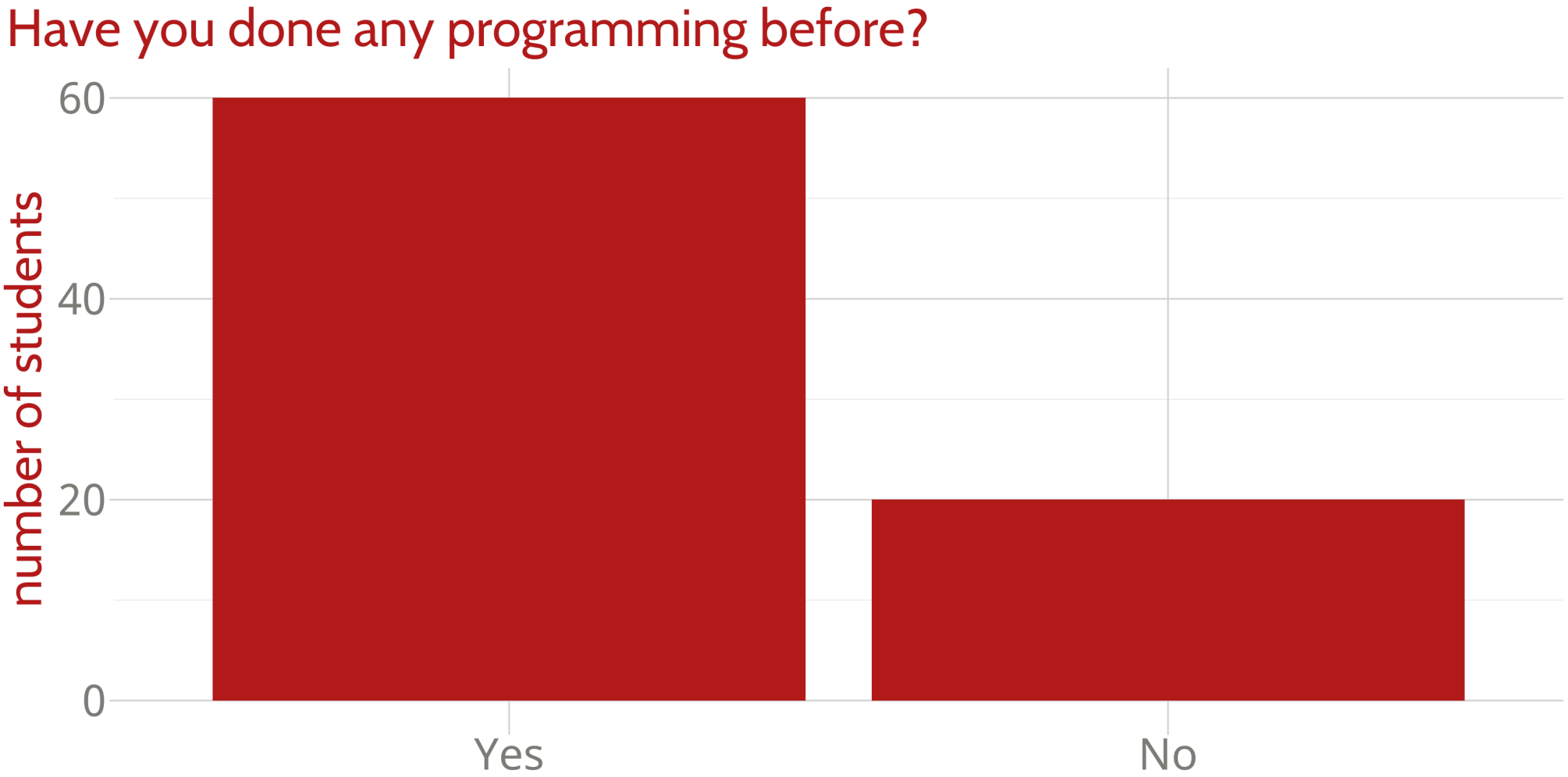
- Distributions
- example-08-2

Proportions

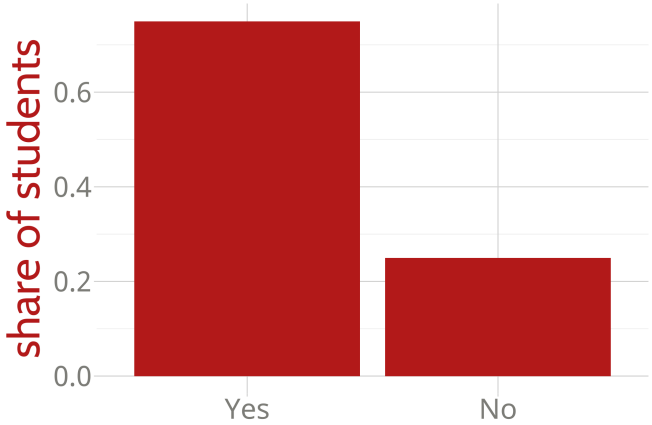
Last week we plotted amounts

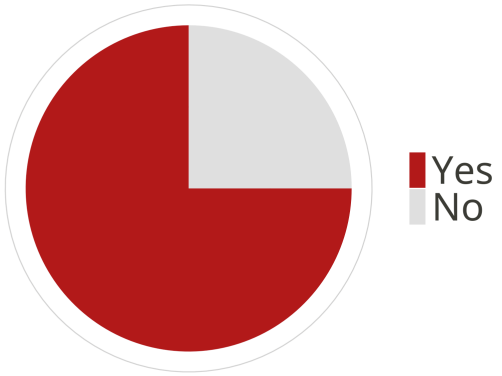


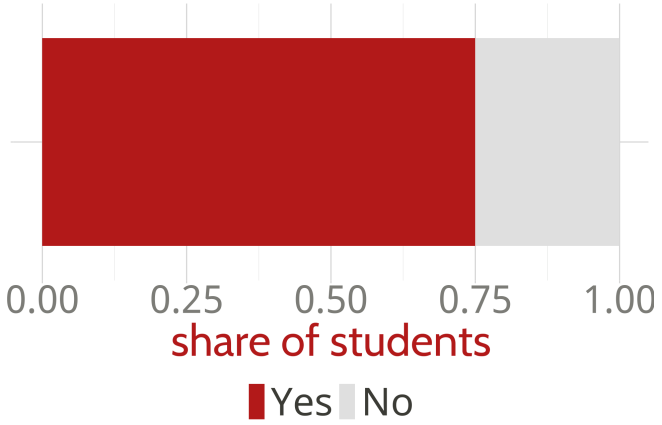
How else could we visualize these data?



Have you done any programming before?







Which do you think is best?

Does it depend on what you want to communicate?

Pros and cons of different approaches

	Pie chart	Stacked bars	Side-by-side bars
Allows easy comparison of relative proportions			

Pros and cons of different approaches

	Pie chart	Stacked bars	Side-by-side bars
Allows easy comparison of relative proportions	✖	✖	✔

Pros and cons of different approaches

	Pie chart	Stacked bars	Side-by-side bars
Allows easy comparison of relative proportions	✖	✖	✔
Shows data as proportions of a whole			

Pros and cons of different approaches

	Pie chart	Stacked bars	Side-by-side bars
Allows easy comparison of relative proportions	✖	✖	✔
Shows data as proportions of a whole	✔	✔	✖

Pros and cons of different approaches

	Pie chart	Stacked bars	Side-by-side bars
Allows easy comparison of relative proportions	✖	✖	✔
Shows data as proportions of a whole	✔	✔	✖
Emphasizes simple fractions (1/2, 1/3, ...)			

Pros and cons of different approaches

	Pie chart	Stacked bars	Side-by-side bars
Allows easy comparison of relative proportions	✖	✖	✔
Shows data as proportions of a whole	✔	✔	✖
Emphasizes simple fractions (1/2, 1/3, ...)	✔	✖	✖

Pros and cons of different approaches

	Pie chart	Stacked bars	Side-by-side bars
Allows easy comparison of relative proportions	✖	✖	✔
Shows data as proportions of a whole	✔	✔	✖
Emphasizes simple fractions (1/2, 1/3, ...)	✔	✖	✖
Visually appealing for small datasets			

Pros and cons of different approaches

	Pie chart	Stacked bars	Side-by-side bars
Allows easy comparison of relative proportions	✖	✖	✔
Shows data as proportions of a whole	✔	✔	✖
Emphasizes simple fractions (1/2, 1/3, ...)	✔	✖	✖
Visually appealing for small datasets	✔	✖	✔

Pros and cons of different approaches

	Pie chart	Stacked bars	Side-by-side bars
Allows easy comparison of relative proportions	✖	✖	✔
Shows data as proportions of a whole	✔	✔	✖
Emphasizes simple fractions (1/2, 1/3, ...)	✔	✖	✖
Visually appealing for small datasets	✔	✖	✔
Works well for a large number of subsets			

Pros and cons of different approaches

	Pie chart	Stacked bars	Side-by-side bars
Allows easy comparison of relative proportions	✖	✖	✔
Shows data as proportions of a whole	✔	✔	✖
Emphasizes simple fractions (1/2, 1/3, ...)	✔	✖	✖
Visually appealing for small datasets	✔	✖	✔
Works well for a large number of subsets	✖	✖	✔

Pros and cons of different approaches

	Pie chart	Stacked bars	Side-by-side bars
Allows easy comparison of relative proportions	✖	✖	✔
Shows data as proportions of a whole	✔	✔	✖
Emphasizes simple fractions (1/2, 1/3, ...)	✔	✖	✖
Visually appealing for small datasets	✔	✖	✔
Works well for a large number of subsets	✖	✖	✔
Works well for time series and similar			

Pros and cons of different approaches

	Pie chart	Stacked bars	Side-by-side bars
Allows easy comparison of relative proportions	✖	✖	✔
Shows data as proportions of a whole	✔	✔	✖
Emphasizes simple fractions (1/2, 1/3, ...)	✔	✖	✖
Visually appealing for small datasets	✔	✖	✔
Works well for a large number of subsets	✖	✖	✔
Works well for time series and similar	✖	✔	✖

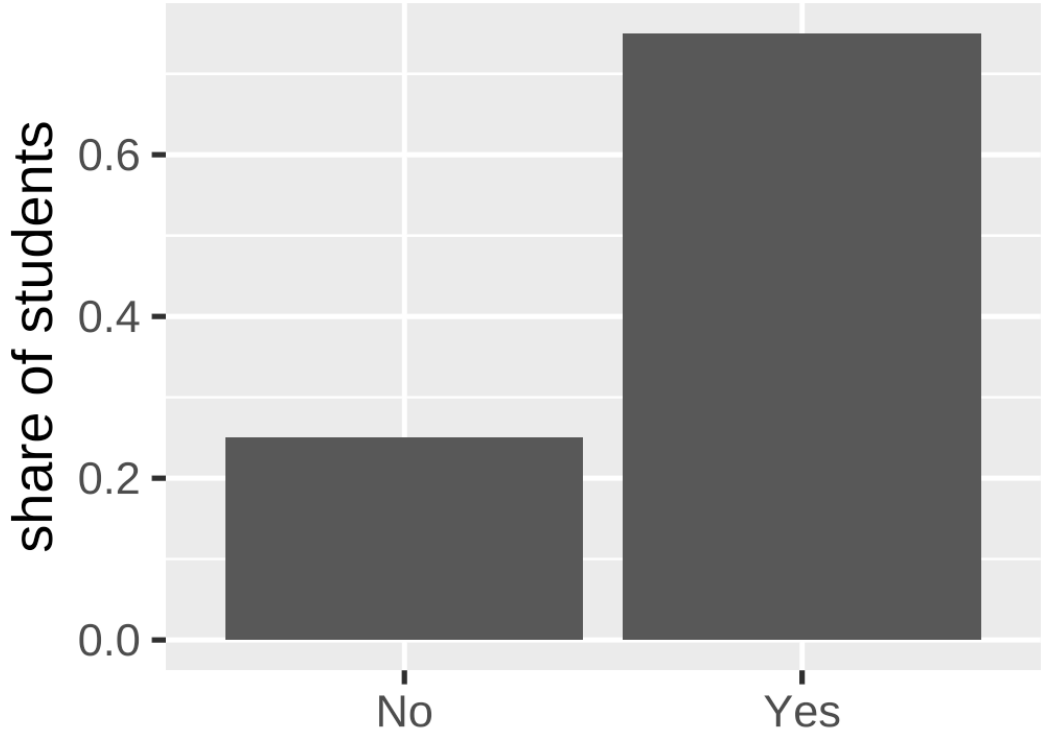
No one visualization fits all scenarios!

Side-by-side bars using ggplot

How could we use ggplot to visualize *proportions* using side-by-side bars?

We could do it manually:

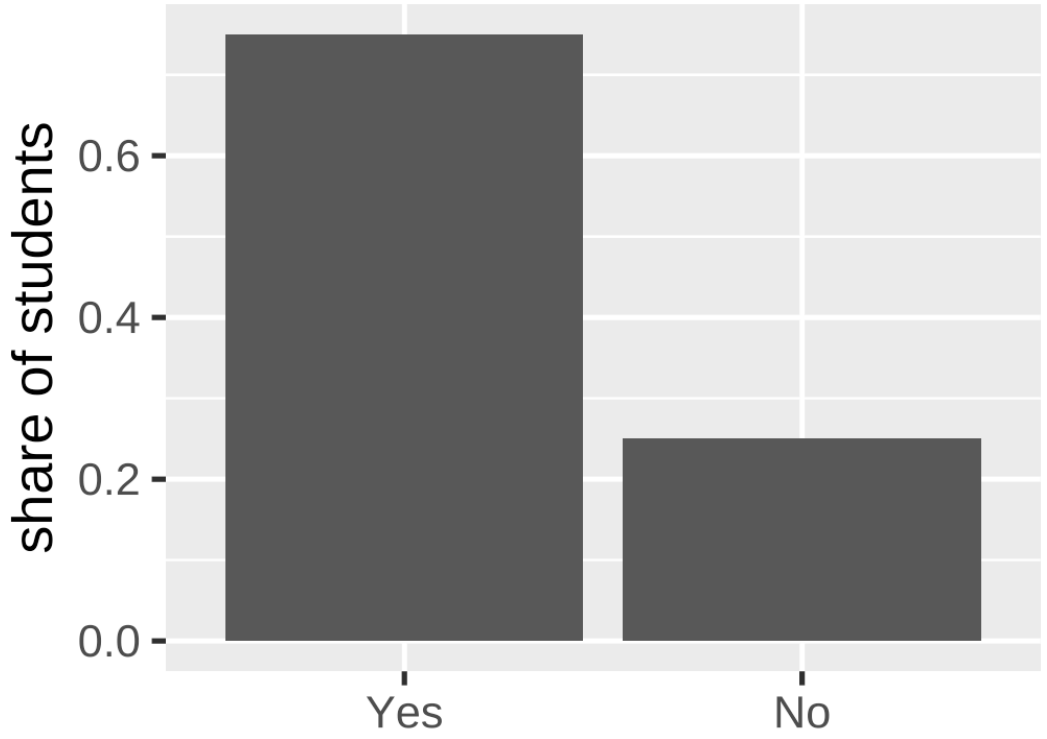
```
prior_programming |>
    count(prior_programming) |>
    mutate(share = n / sum(n)) |>
    ggplot(aes(
      x = prior_programming,
      y = share
      )) +
    geom_col() +
    labs(x = NULL,
         y = "share of students")
```



How could we reverse the bars' order?

Side-by-side bars using ggplot

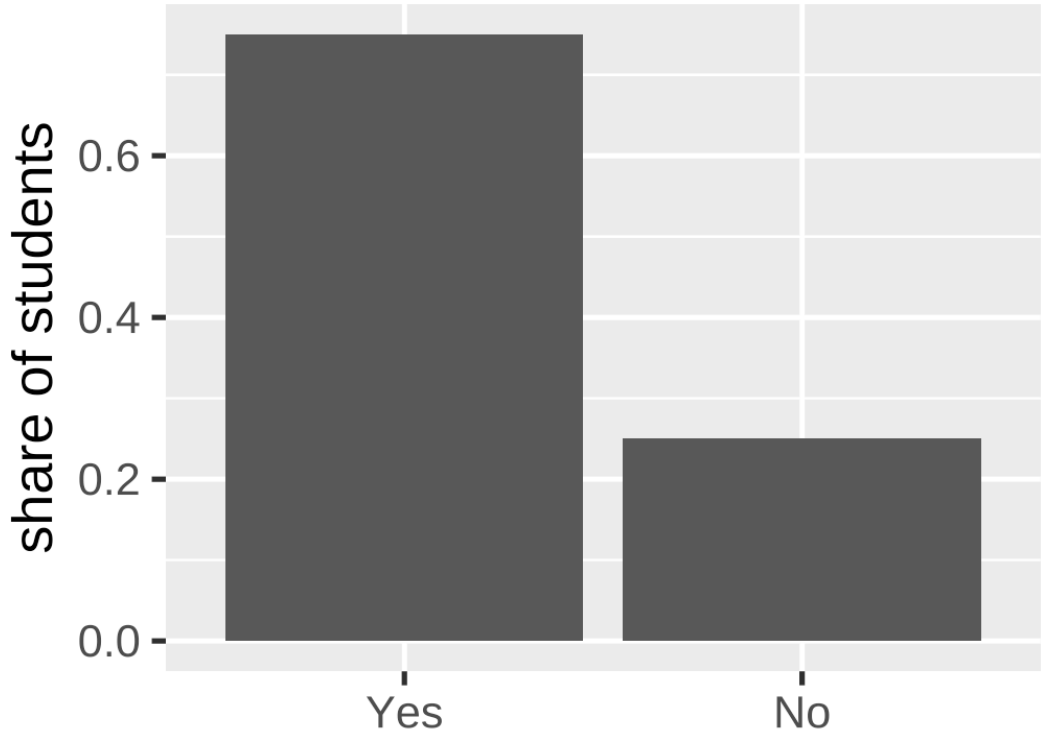
```
prior_programming |>
    count(prior_programming) |>
    mutate(share = n / sum(n)) |>
    ggplot(aes(
      x = fct_reorder(
        prior_programming,
        -share
        ),
      y = share
      )) +
    geom_col() +
    labs(x = NULL,
         y = "share of students")
```



Side-by-side bars using ggplot

`fct_rev()` also works well since there are only two categories:

```
prior_programming |>
    count(prior_programming) |>
    mutate(share = n / sum(n)) |>
    ggplot(aes(
      x = fct_rev(prior_programming),
      y = share
      )) +
    geom_col() +
    labs(x = NULL,
         y = "share of students")
```

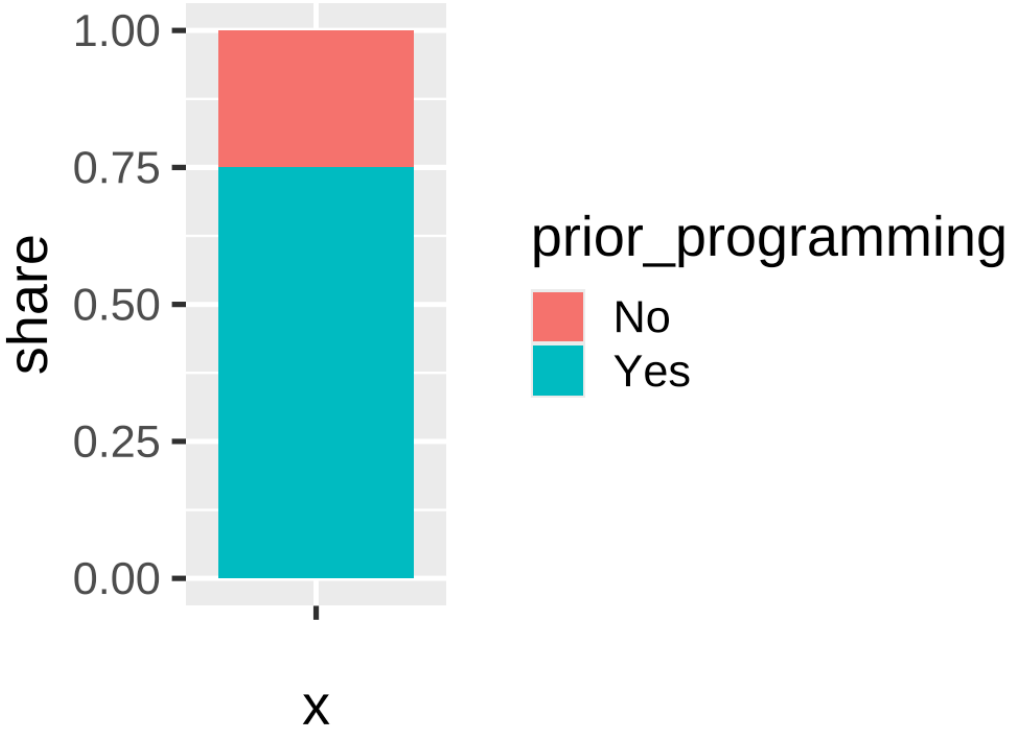


Stacked bars using ggplot

How could we use ggplot to visualize *proportions* using stacked bars?

Again, we could do it manually:

```
prior_programming |>
    count(prior_programming) |>
    mutate(share = n / sum(n)) |>
    ggplot(aes(
        x = "",     # provide dummy to x
        y = share, # plot shares on y
        fill = prior_programming
    )) +
    geom_col()
```

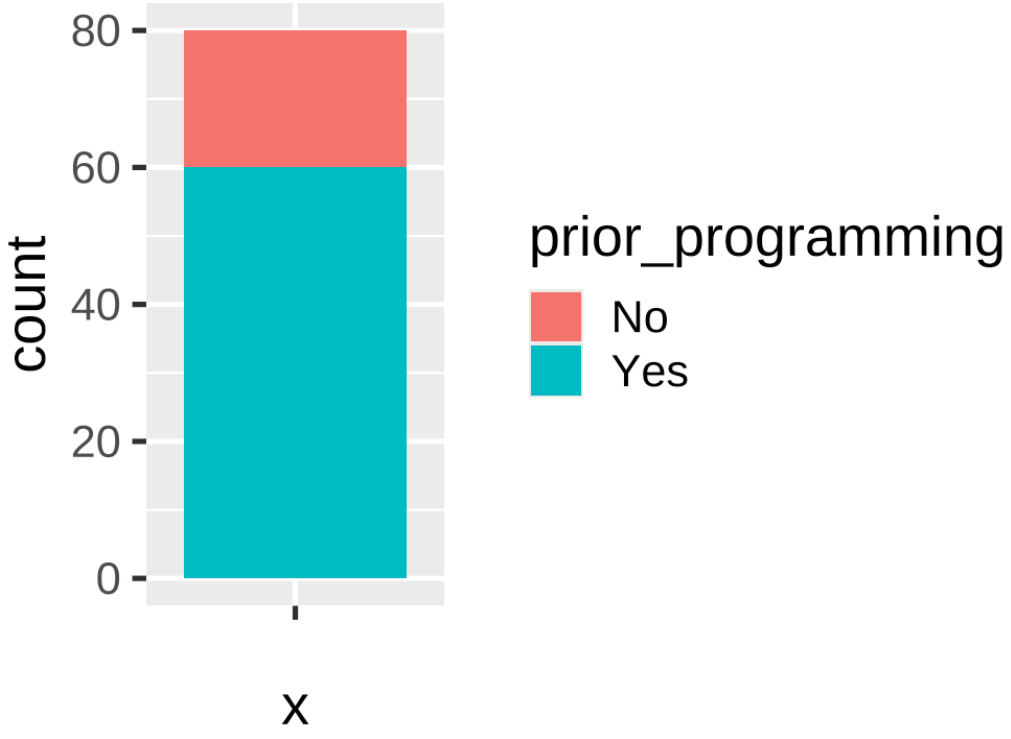


By default, `geom_col` stacks bars if they fall in the same place (`x`)

Stacked bars using ggplot

Alternatively, we could use `geom_bar()` to count and plot the data for us

```
prior_programming |>
  ggplot(aes(
    x = "",
    fill = prior_programming
  )) +
  geom_bar()
```

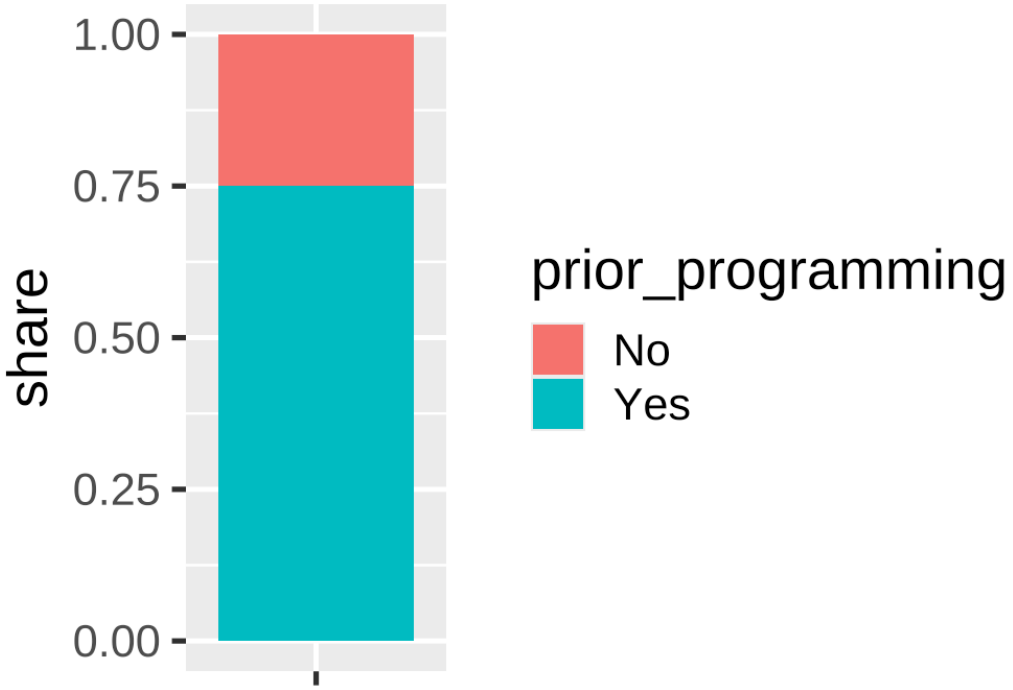


But this gives us *counts*. We want *shares*!

Stacked bars using ggplot

The argument `position = "fill"` scales everything to sum to 1

```
prior_programming |>
  ggplot(aes(
    x = "",
    fill = prior_programming
  )) +
  geom_bar(position = "fill") +
  labs(x = NULL, y = "share")
```

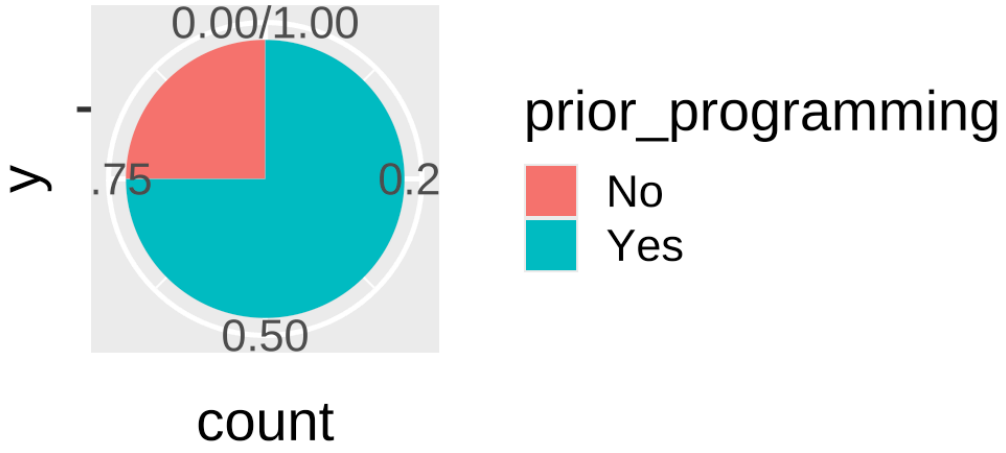


Pie charts using ggplot

How could we use ggplot to visualize *proportions* using stacked bars?

Pie charts are just stacked bars in polar coordinates

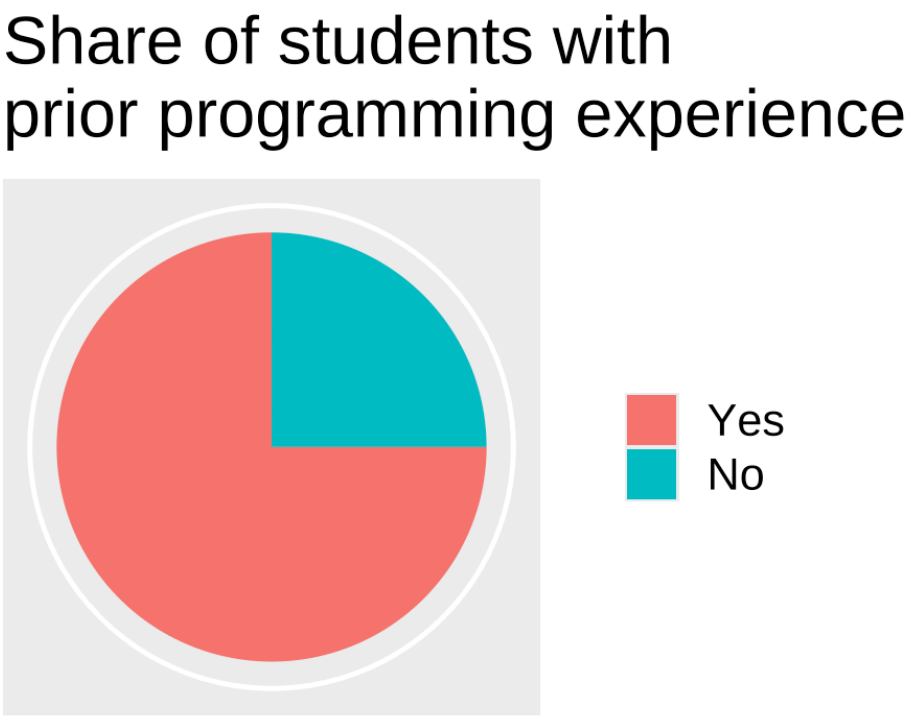
```
prior_programming |>
  ggplot(aes(
    y = "", # x, not y
    fill = prior_programming
  )) +
  geom_bar(position = "fill") +
  coord_polar() # convert to polar coordinates
```



Pie charts using ggplot

It takes some work to create a clean pie chart using ggplot

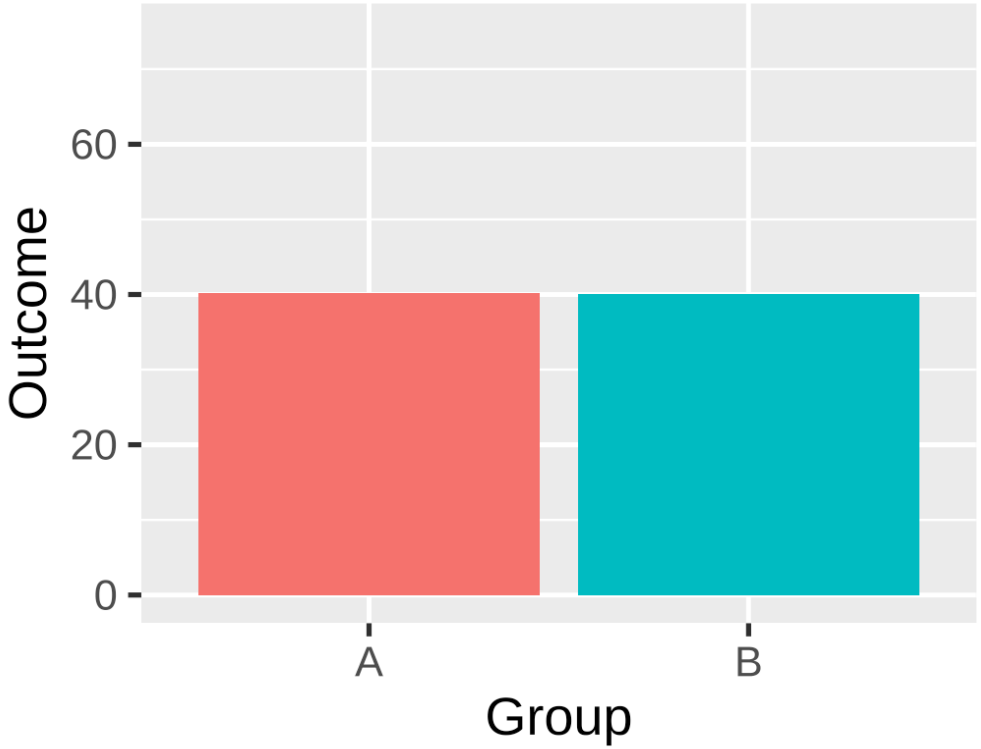
```
prior_programming |>
  ggplot(aes(
    y = "",
    fill = fct_rev(prior_programming)
  )) +
  geom_bar(position = "fill") +
  coord_polar() +
  scale_x_continuous(
    name = NULL, breaks = NULL
  ) +
  scale_y_discrete(
    name = NULL, breaks = NULL
  ) +
  labs(
    title = "Share of students with\nprior prog
    fill = NULL
  )
```

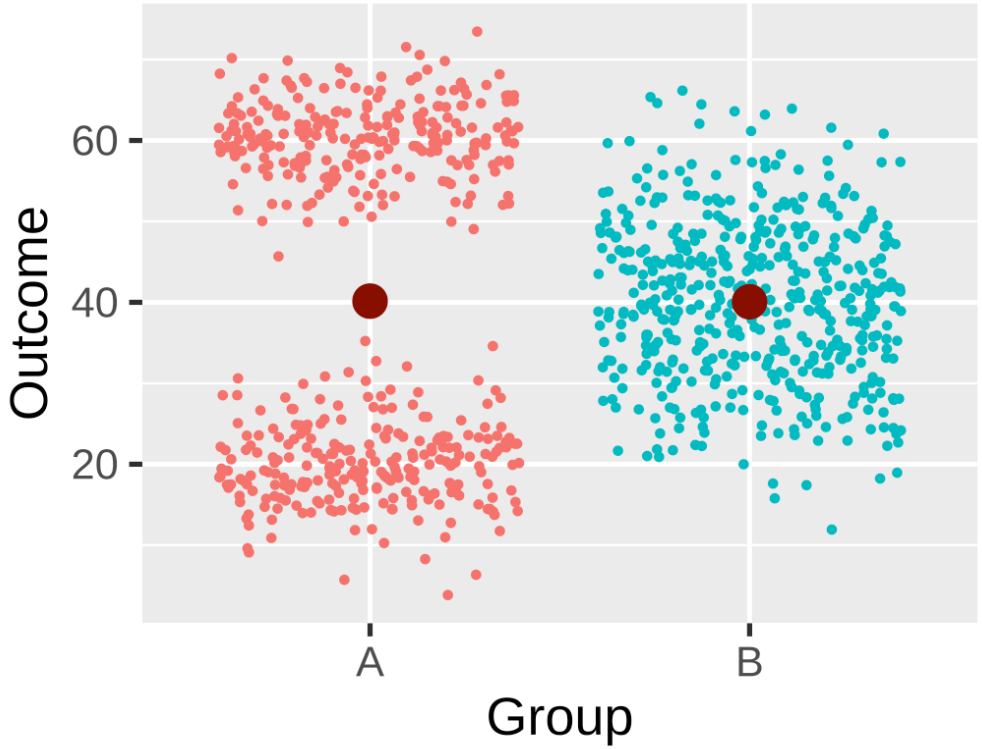


example-08-1: proportions-practice.R

Distributions

Problems with single numbers





More information is (almost) always better

Avoid visualizing single numbers when you have a whole range or distribution of numbers

Uncertainty in single variables

Uncertainty across multiple variables

Uncertainty in models and simulations

What are some common methods for visualizing distributions?

Histograms, densities, box plots, etc.

Histograms

What are they?

Put data into equally spaced buckets (or "bins") based on values of a variable, plot how many rows of the data frame are in each bucket

Histograms

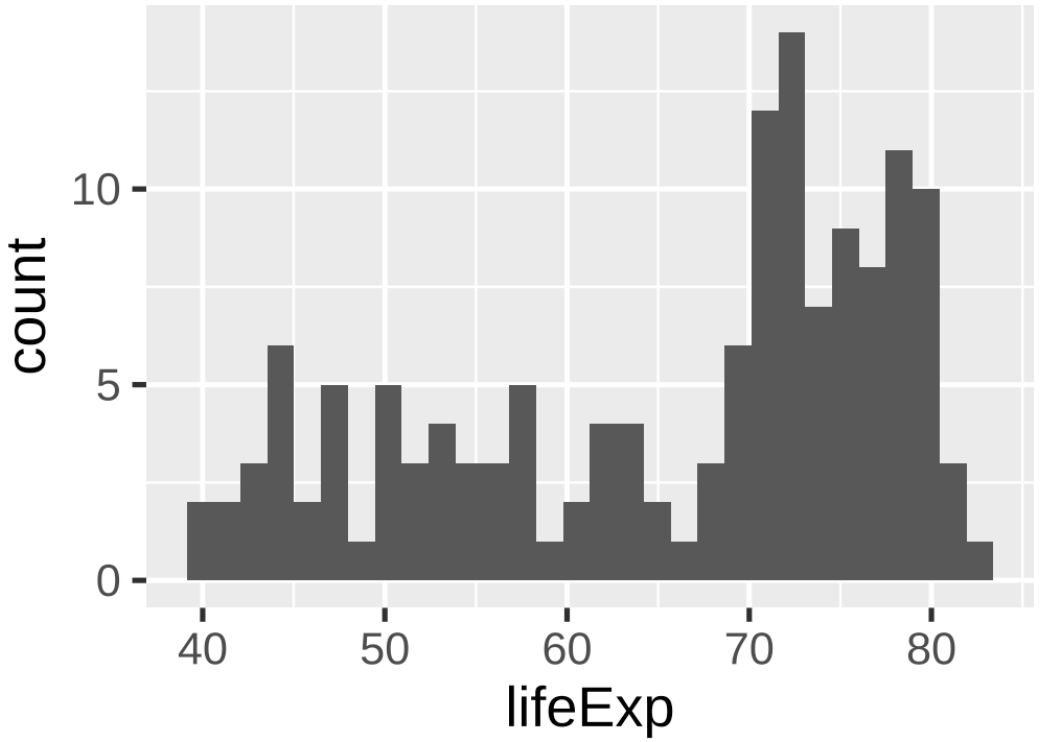
How would we use the grammar of graphics to make a histogram of `lifeExp`?

```
library(gapminder)

gapminder_2002 <- gapminder |>
  filter(year == 2002)

head(gapminder_2002)
```

```
## # A tibble: 6 × 6
##   country     continent  year lifeExp      pop gdpPercap
##   <fct>       <fct>     <int>   <dbl>    <int>     <dbl>
## 1 Afghanistan Asia       2002    42.1 25268405      727.
## 2 Albania     Europe     2002    75.7  3508512     4604.
## 3 Algeria     Africa     2002    71.0 31287142     5288.
## 4 Angola      Africa     2002    41.0 10866106     2773.
## 5 Argentina   Americas   2002    74.3 38331121     8798.
## 6 Australia   Asia       2002    80.4 19546792    30688.
```

Histograms

```
gapminder_2002 |>
  ggplot(aes(x = lifeExp)) +
  geom_histogram()
```

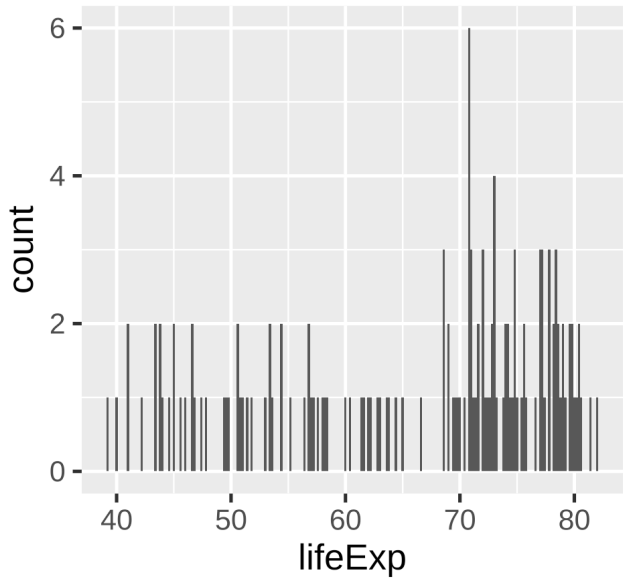


Histograms: binwidth argument

No official rule for what makes a good bin width

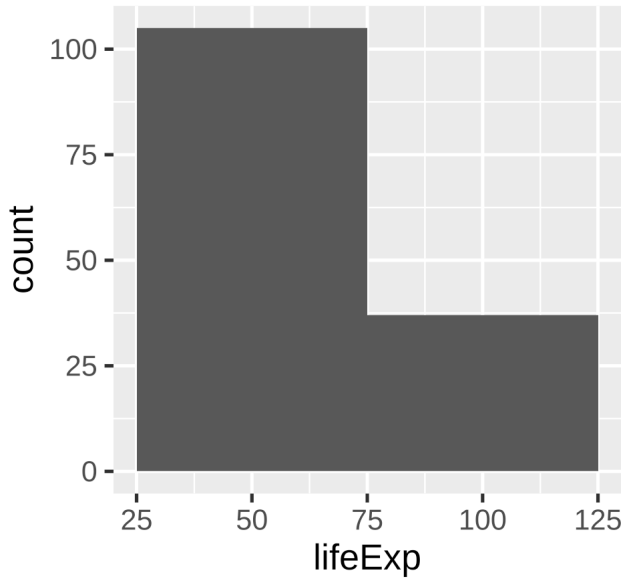
Too narrow:

```
geom_histogram(binwidth = .2)
```



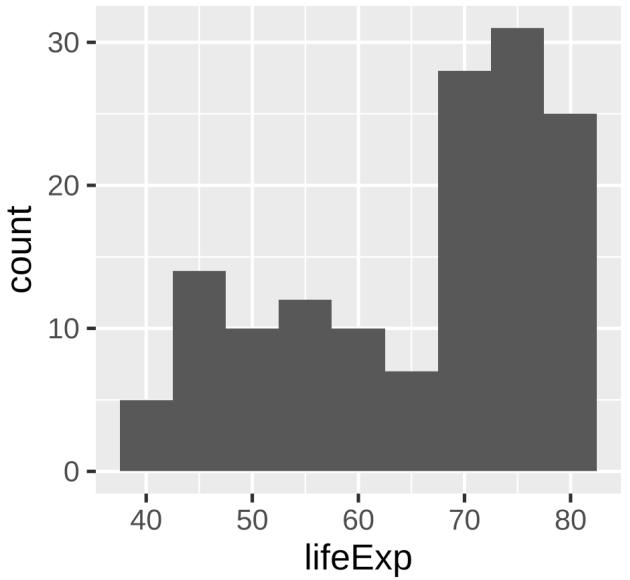
Too wide:

```
geom_histogram(binwidth = 50)
```



(One type of) just right:

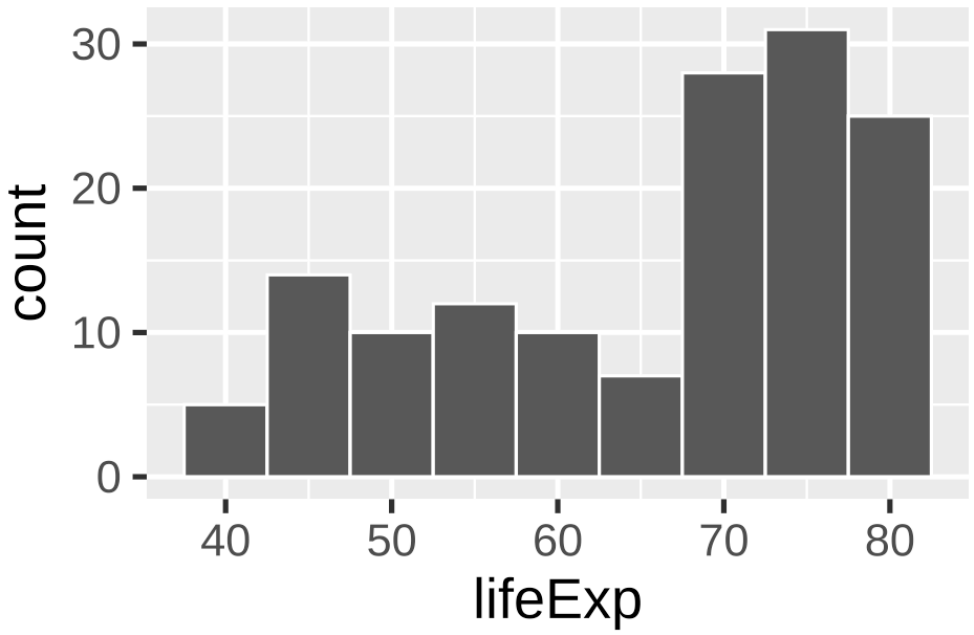
```
geom_histogram(binwidth = 5)
```



Histograms: tips using other arguments

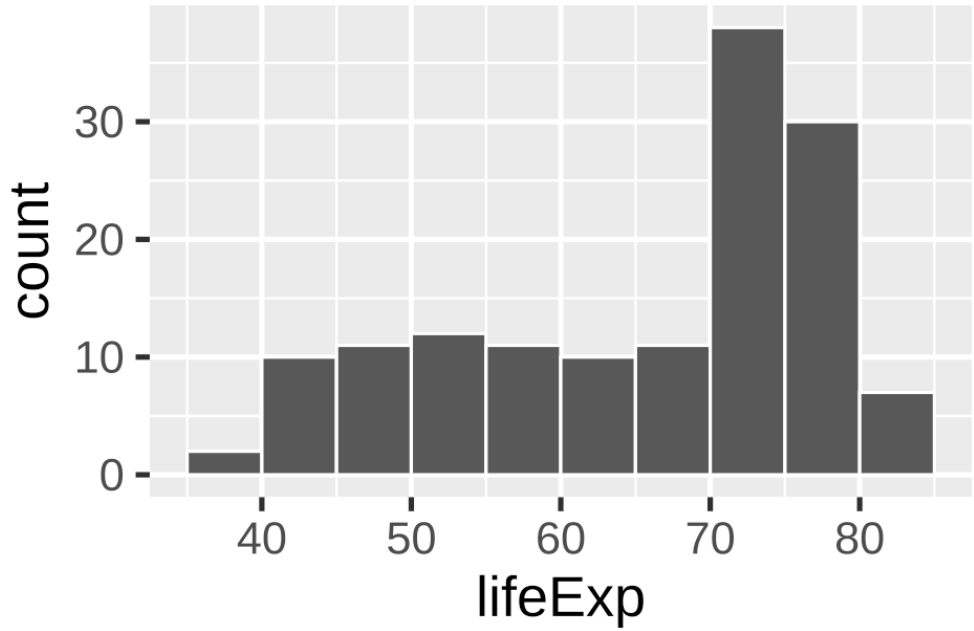
Add a border to the bars
for readability

```
geom_histogram(..., color = "white")
```



Set the boundary;
bucket now 50–55, not 47.5–52.5

```
geom_histogram(..., boundary = 50)
```



Density plots

What are they?

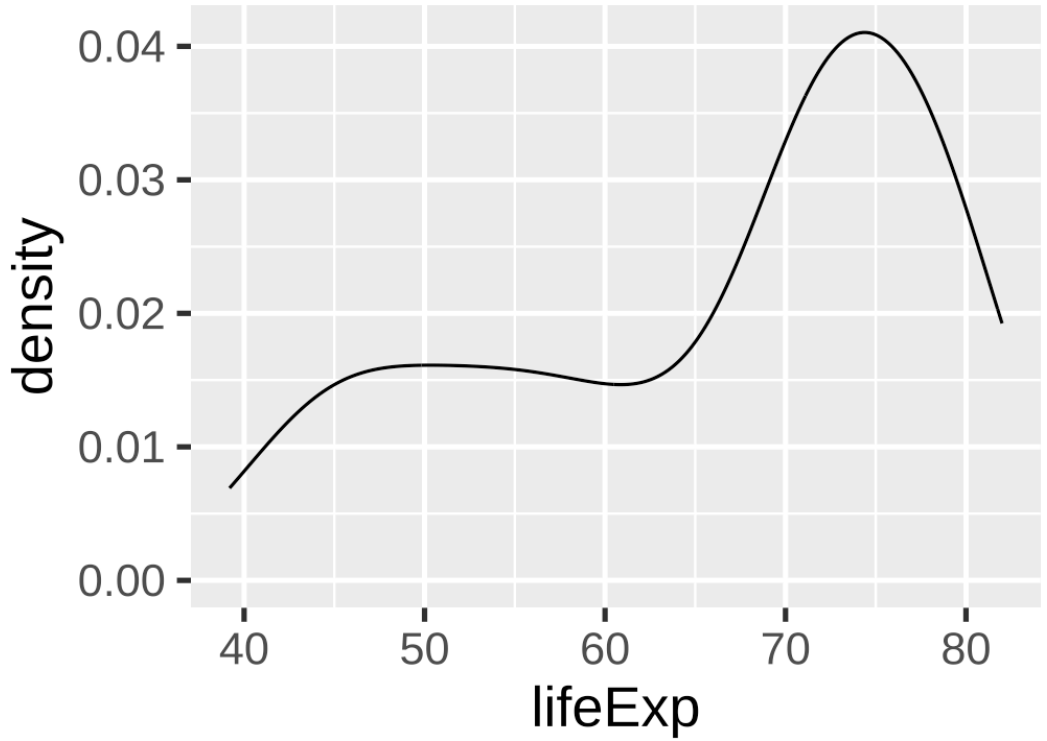
Estimates of the **probability *density* function** of a random variable

Histograms show raw counts; density plots show proportions (integrate to 1)

How would we use the grammar of graphics to make a density plot of `lifeExp`?

Density plots

```
gapminder_2002 |>
  ggplot(aes(x = lifeExp)) +
  geom_density()
```

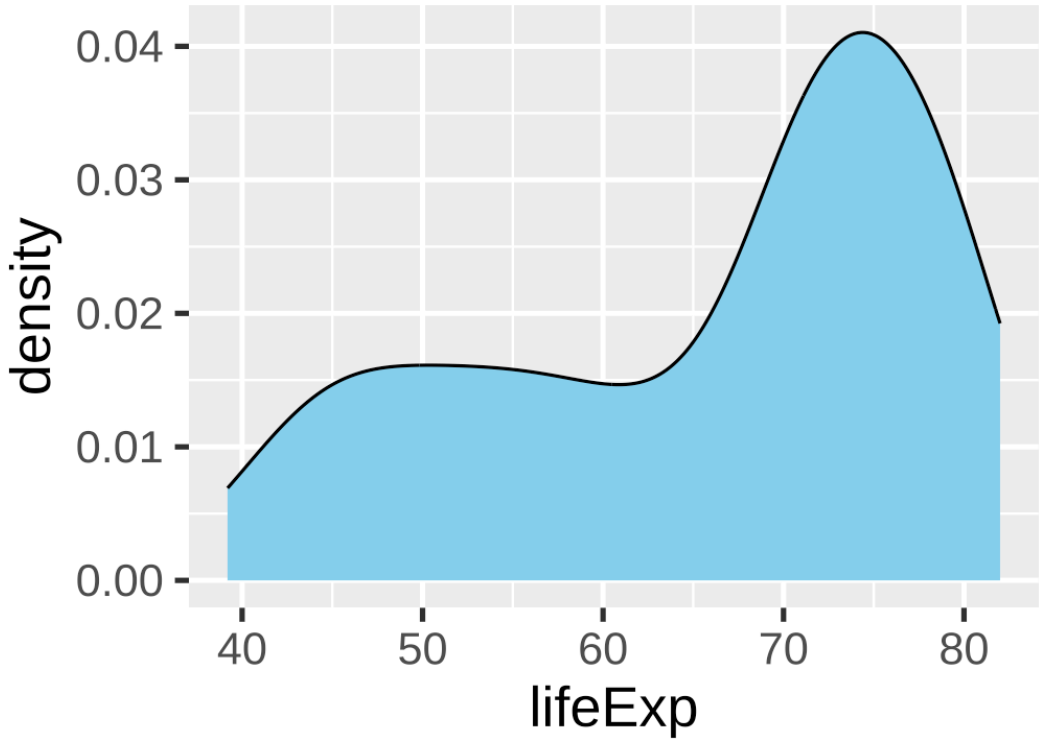


Density plots: add some color

```
gapminder_2002 |>
  ggplot(aes(x = lifeExp)) +
  geom_density(fill = "skyblue")
```

We can use aesthetics as parameters inside a geom rather than inside an **aes()** statement

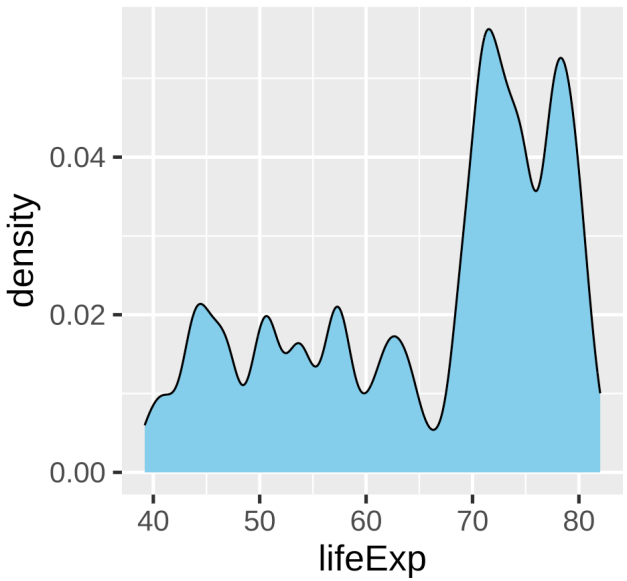
Here we used **fill = "skyblue"**



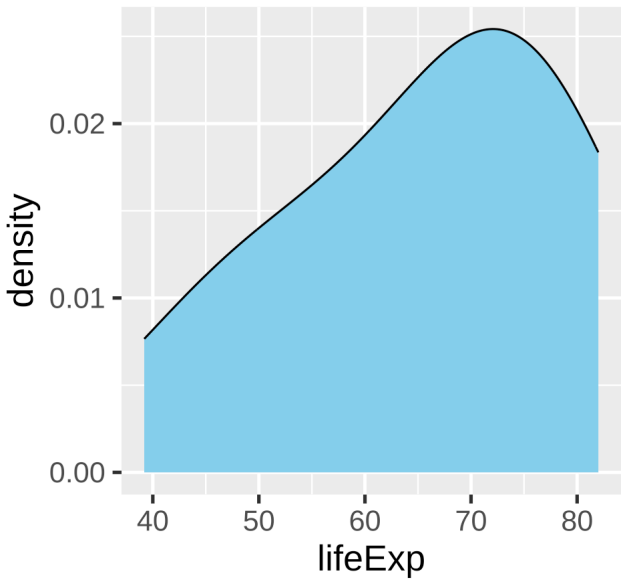
Density plots: bandwidths

Different options for calculus change the plot shape

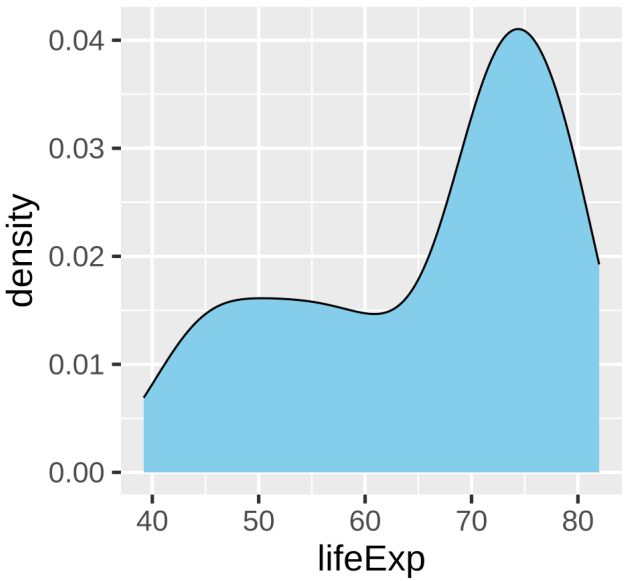
`bw = 1`



`bw = 10`



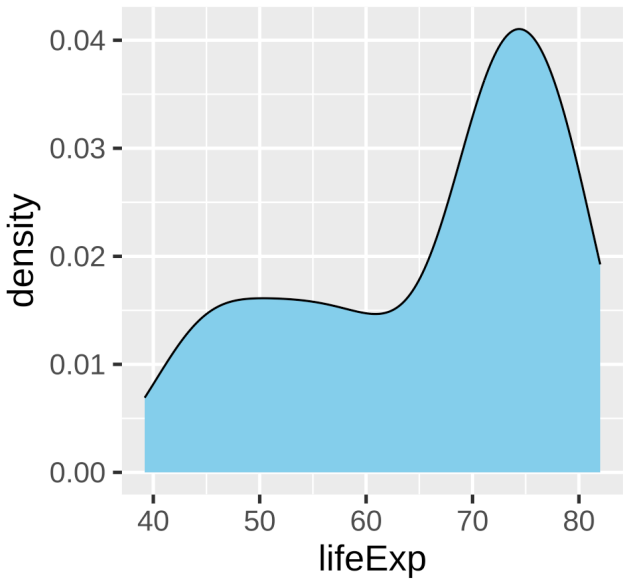
`bw = "nrd0"` (default)



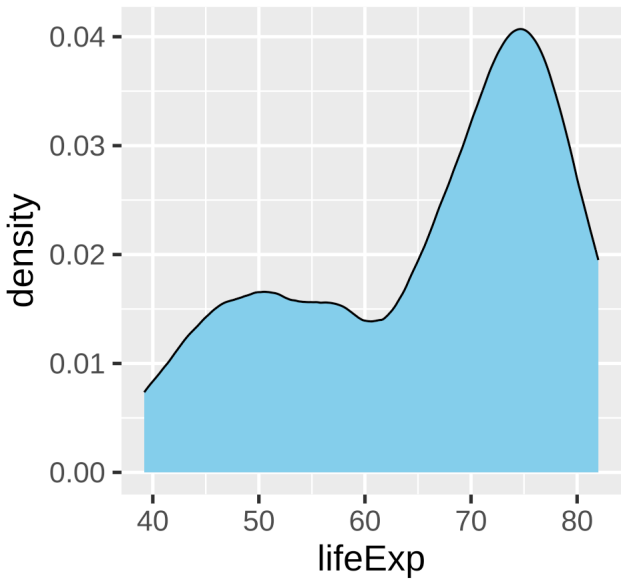
Density plots: kernels

Different options for calculus change the plot shape

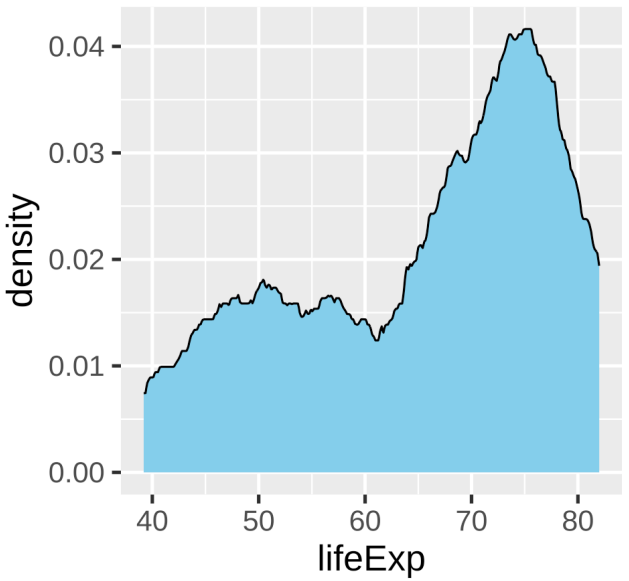
`kernel = "gaussian"`



`"epanechnikov"`



`"rectangular"`

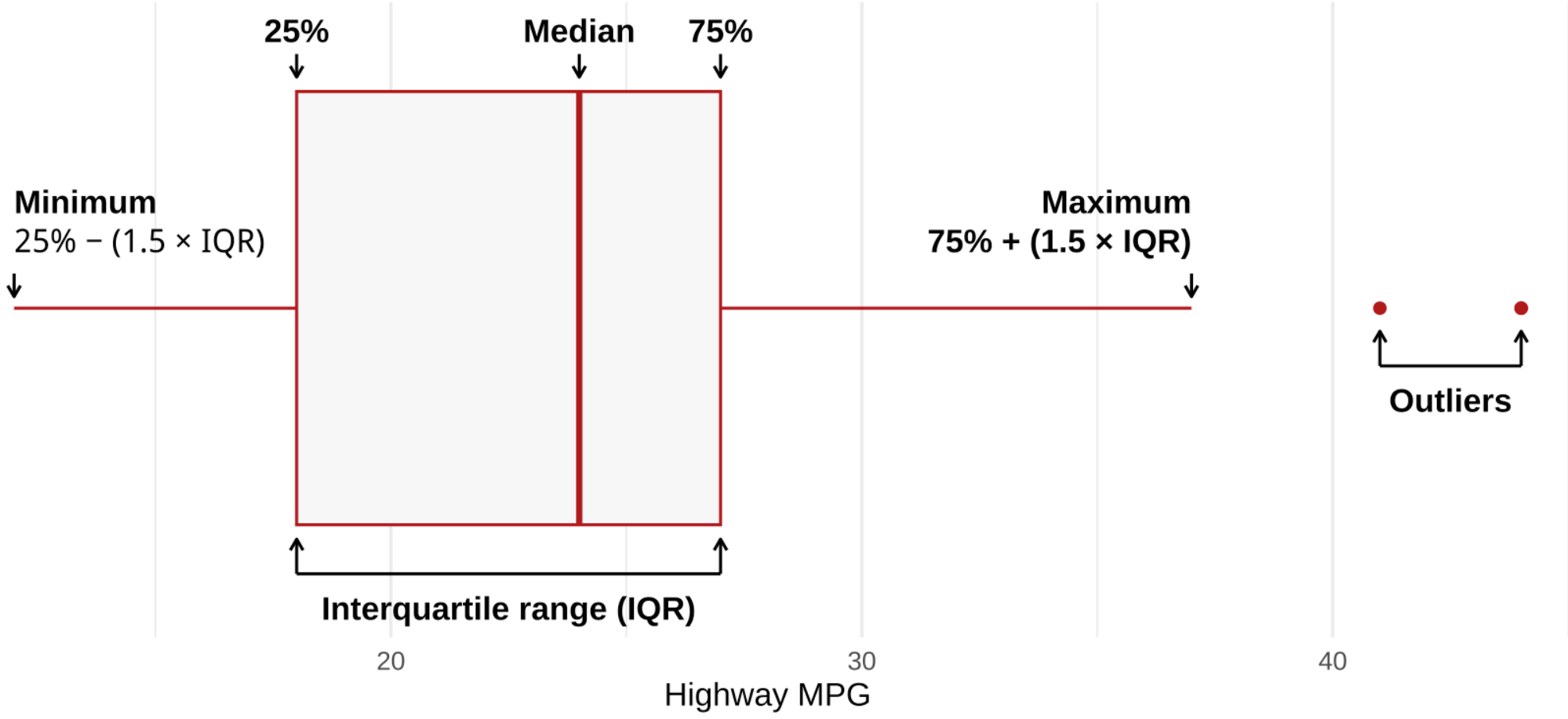


Box and whisker plots

What are they?

Graphical representations of specific points in a distribution

Box and whisker plots



Box and whisker plots

What are they?

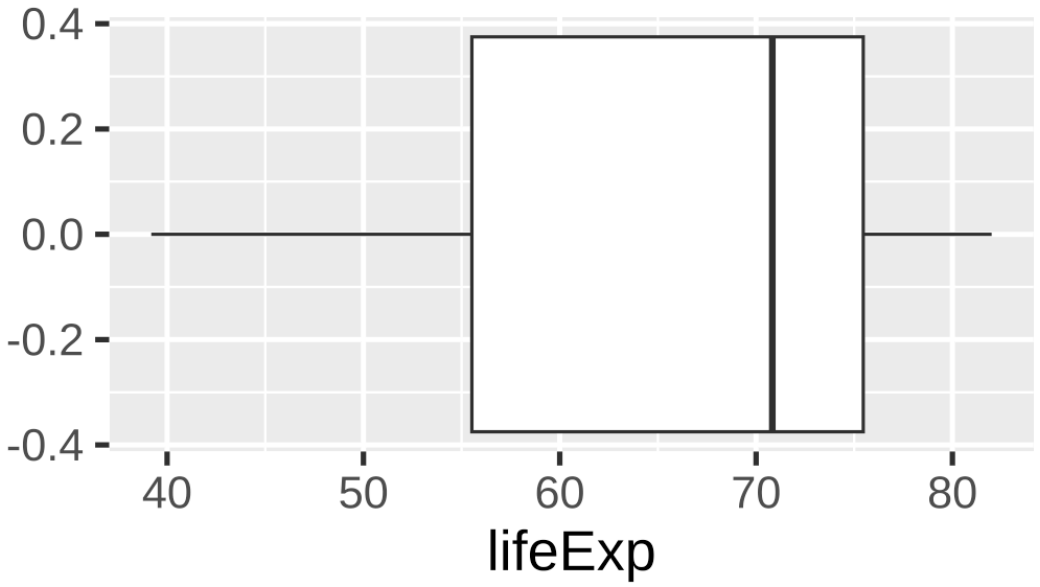
Graphical representations of specific points in a distribution

How could we use ggplot to make a boxplot of `lifeExp`?

Box and whisker plots

```
gapminder_2002 |>
  ggplot(aes(x = lifeExp)) +
  geom_boxplot()
```

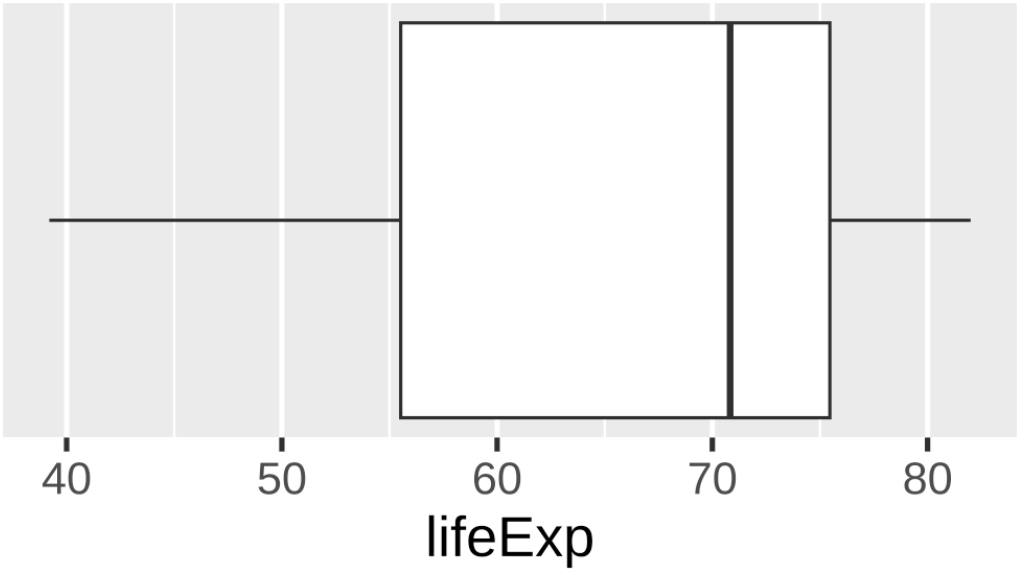
What do the y axis numbers mean?



Box and whisker plots

Use `theme()` to customize the plot for this geom

```
gapminder_2002 |>
  ggplot(aes(x = lifeExp)) +
  geom_boxplot() +
  theme(
    axis.text.y = element_blank(),
    axis.ticks.y = element_blank(),
    panel.grid.major.y = element_blank(),
    panel.grid.minor.y = element_blank()
  )
```



Uncertainty across multiple variables

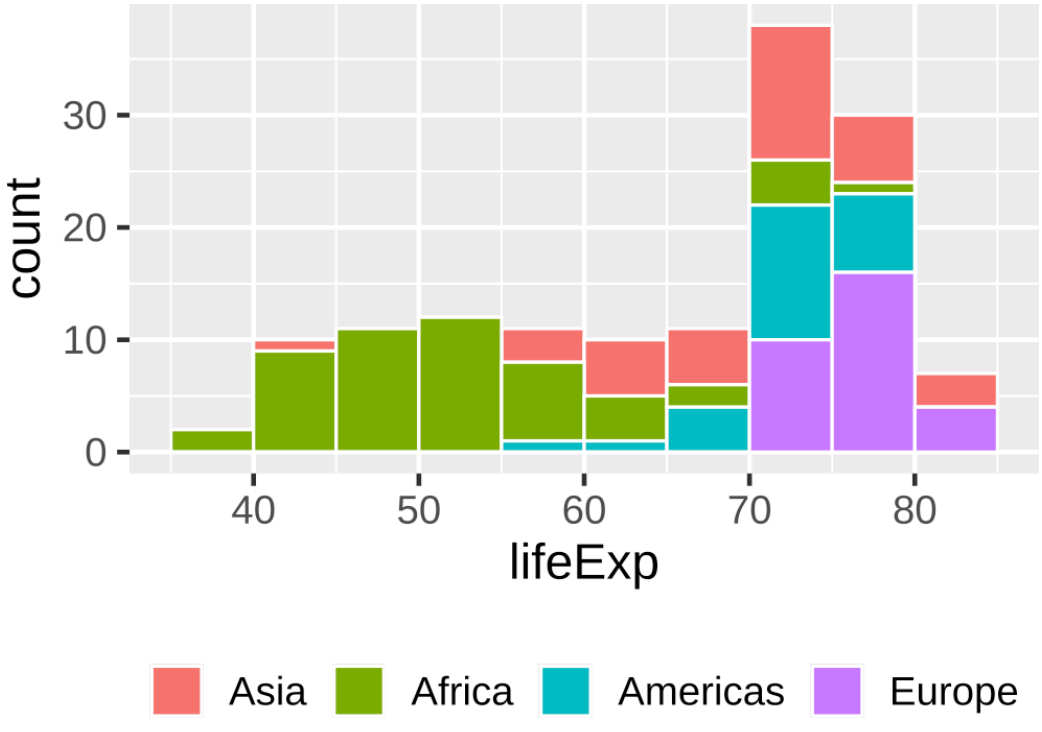
How could we visualize the distribution of a single variable across groups?

Add a `fill` aesthetic or use facets!

Multiple histograms

Fill with a different variable

```
gapminder_2002 |>
  ggplot(aes(x = lifeExp,
             fill = continent)) +
  geom_histogram(binwidth = 5,
                 color = "white",
                 boundary = 50) +
  theme(legend.position = "bottom") +
  labs(fill = NULL)
```

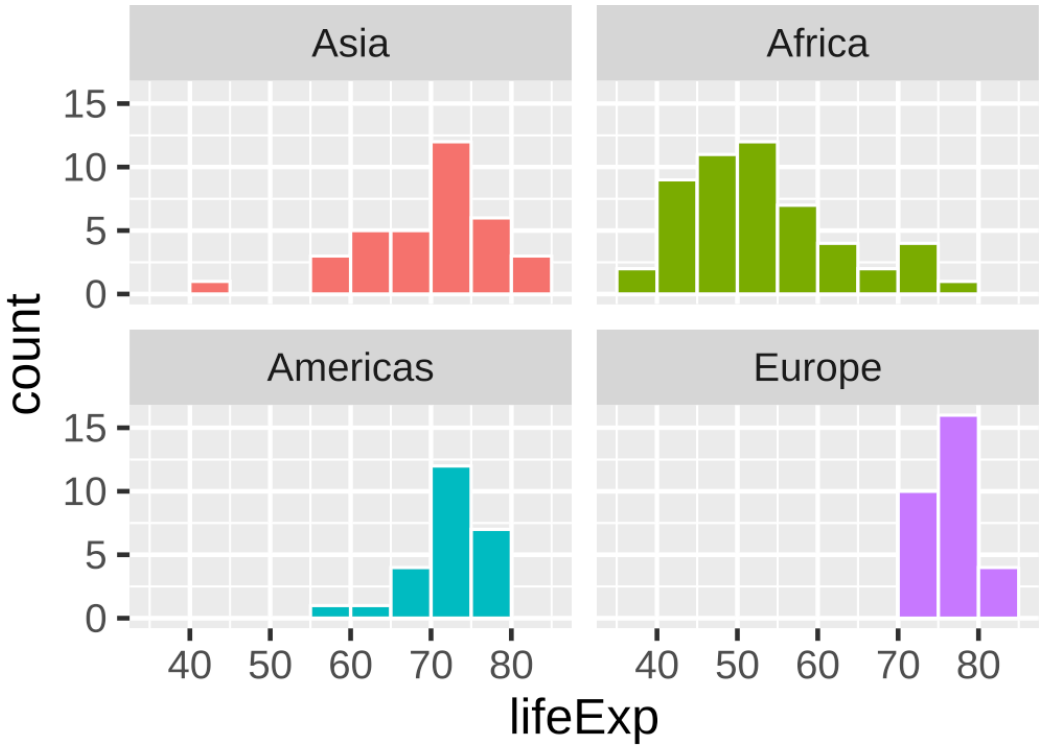


This stacked histogram is bad and hard to read though

Multiple histograms

Facet with a different variable

```
gapminder_2002 |>
  ggplot(aes(x = lifeExp,
             fill = continent)) +
  geom_histogram(binwidth = 5,
                 color = "white",
                 boundary = 50) +
  facet_wrap(vars(continent)) +
  guides(fill = "none")
```

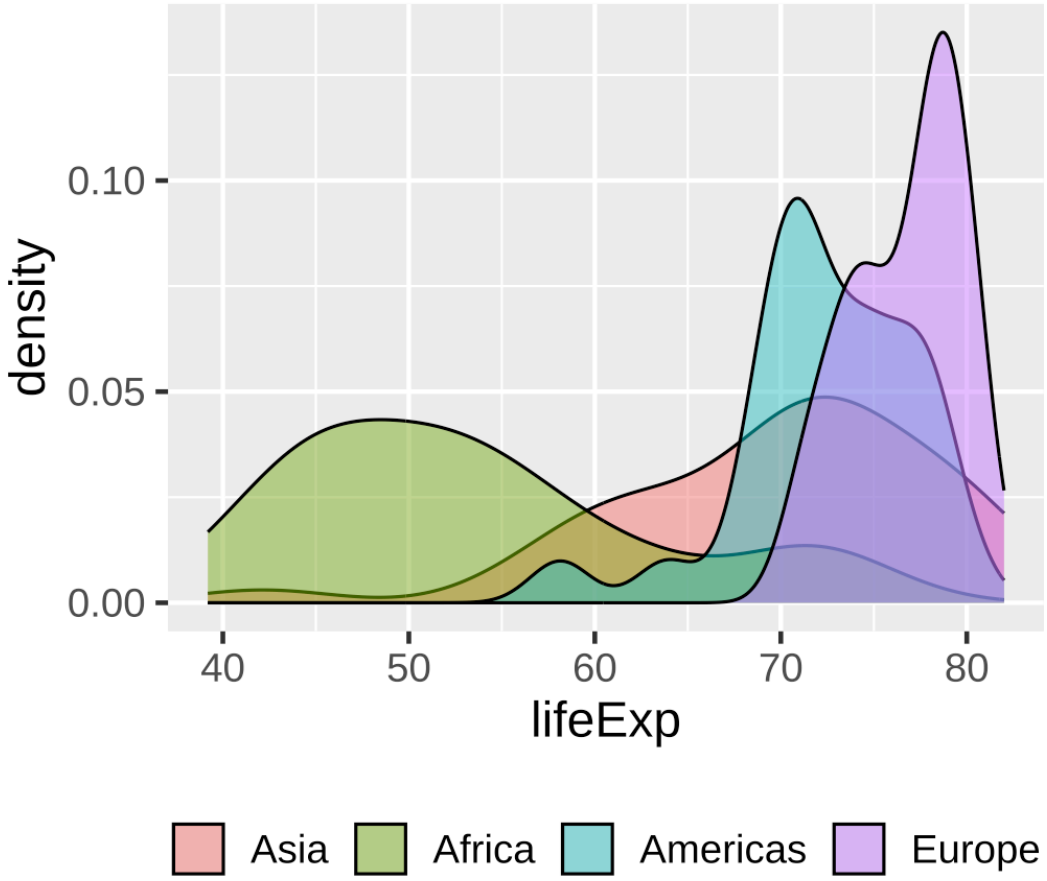


Multiple densities: Transparency

```
gapminder_2002 |>
  ggplot(aes(x = lifeExp,
             fill = continent)) +
  geom_density(alpha = 0.5) +
  theme(legend.position = "bottom") +
  labs(fill = NULL)
```

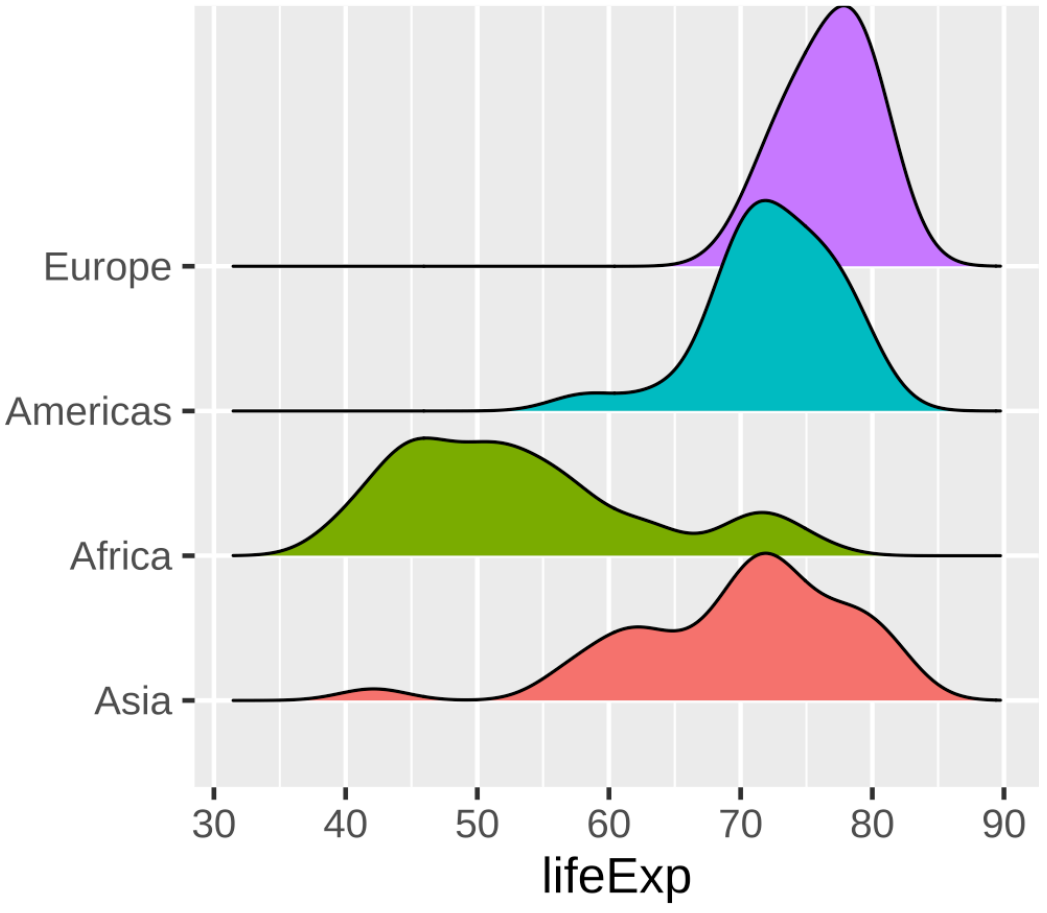
But be careful, these can get confusing quickly

With many groups, better to space them out using ridgeline plots



Multiple densities: Ridgeline plots

```
library(ggridges)

gapminder_2002 |>
  ggplot(aes(x = lifeExp,
             fill = continent,
             y = continent)) +
  guides(fill = "none") +
  labs(y = NULL) +
  geom_density_ridges()
```

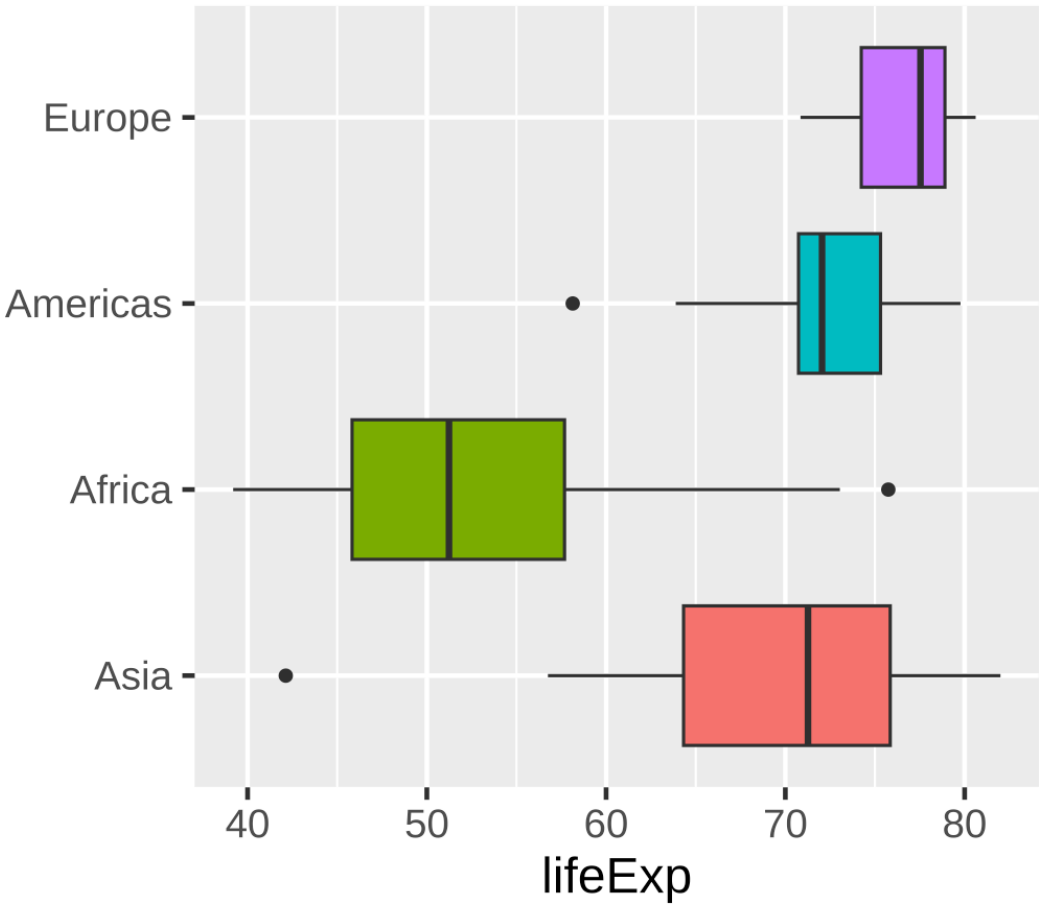


There is no explicit scale for the densities anymore (it is shared with y)

With many densities, use a single fill color to prevent distraction

Multiple box and whisker plots

```
gapminder_2002 |>
  ggplot(aes(
    x = lifeExp,
    fill = continent,
    y = continent
  )) +
  guides(fill = "none") +
  labs(y = NULL) +
  geom_boxplot()
```



example-08-2: distributions-practice.R