Week 3 Lab

Visualization Using ggplot2

Micaela Wood
01/20/2022

ggplot2

- Today we will learn how to present data visually with `ggplot2`
- This comes with the `tidyverse` package so you can install it in one of two ways

```
install.packages("tidyverse")
library(tidyverse)
```

or

```
install.packages("ggplot2")
library(ggplot2)
```

Notes about ggplot2

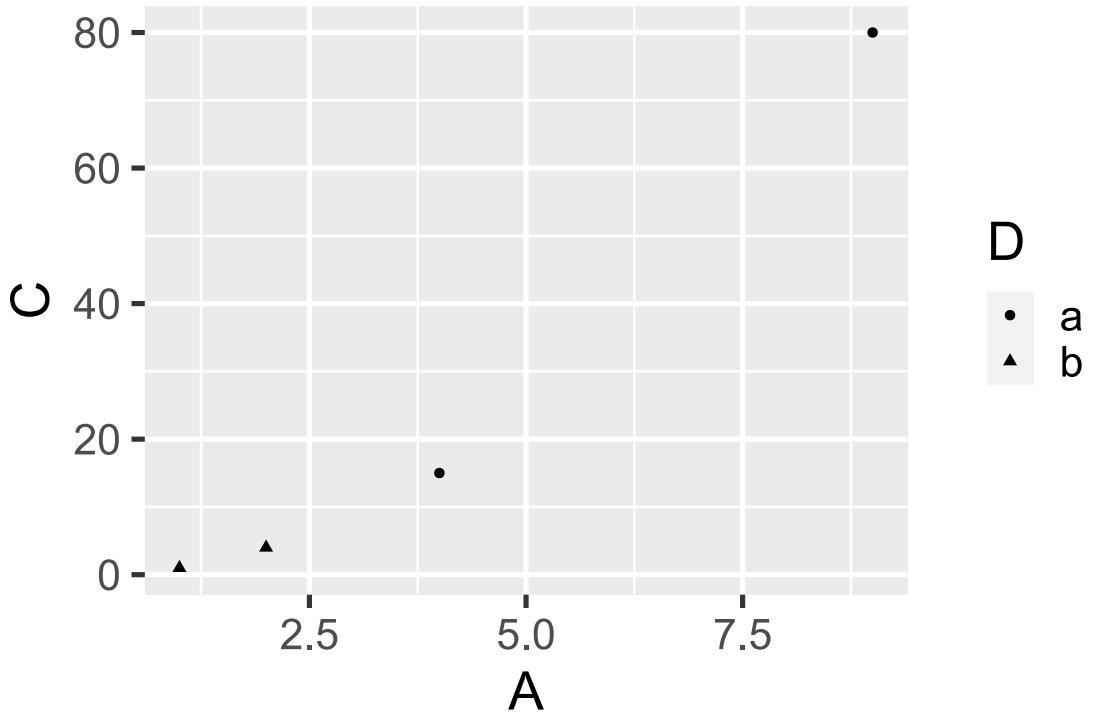
- Data needs to be in a data frame.
- To change data into a data frame use `new data ← as.data.frame(old data)` or use `new data ← as_tibble(old data)`
- First, we will start with a small data frame

```
data ← tribble(
  ~"A", ~"B", ~"C", ~"D",
  #---/-----/-----/-----
  2, 3, 4, "b",
  1, 2, 1, "b",
  4, 5, 15, "a",
  9, 10, 80, "a"
)
```

Basic Plot

- First let's make a simple scatter plot of A versus C

```
ggplot(data=data, mapping=aes(x=A, y=C, shape=D))+
  geom_point()
```

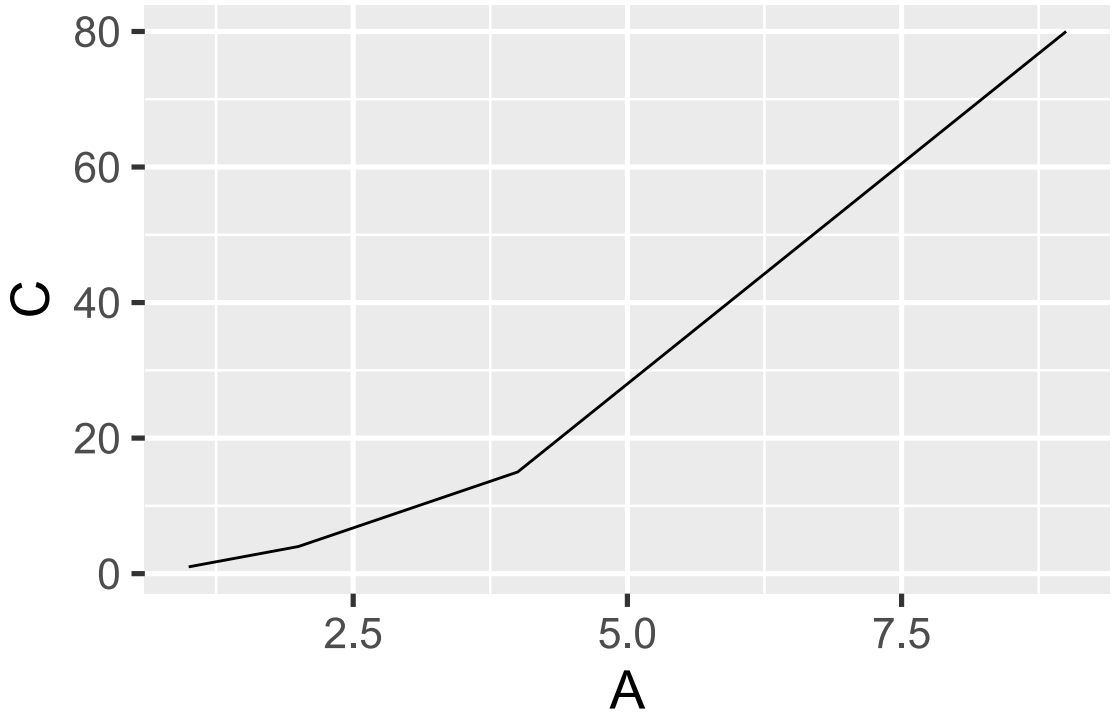


Basic Plot

- Notice, `ggplot` mapped "a" to a circle and "b" to a triangle. This transformation is the responsibility of **scales**. So if you want the mapping to be different, specify a different scale.
- We can create lots of different types of plots using this same basic specification. For example, if we drew lines instead of points, we would get a line plot.
- Or, if we used bars, we would get a bar plot. Bars, lines, and points are all examples of **geometric objects**, `geom` for short.
- `ggplot2` has a ton of options for geoms, but we'll be using `geom_point`, `geom_line`, `geom_bar`, etc.

Line Example

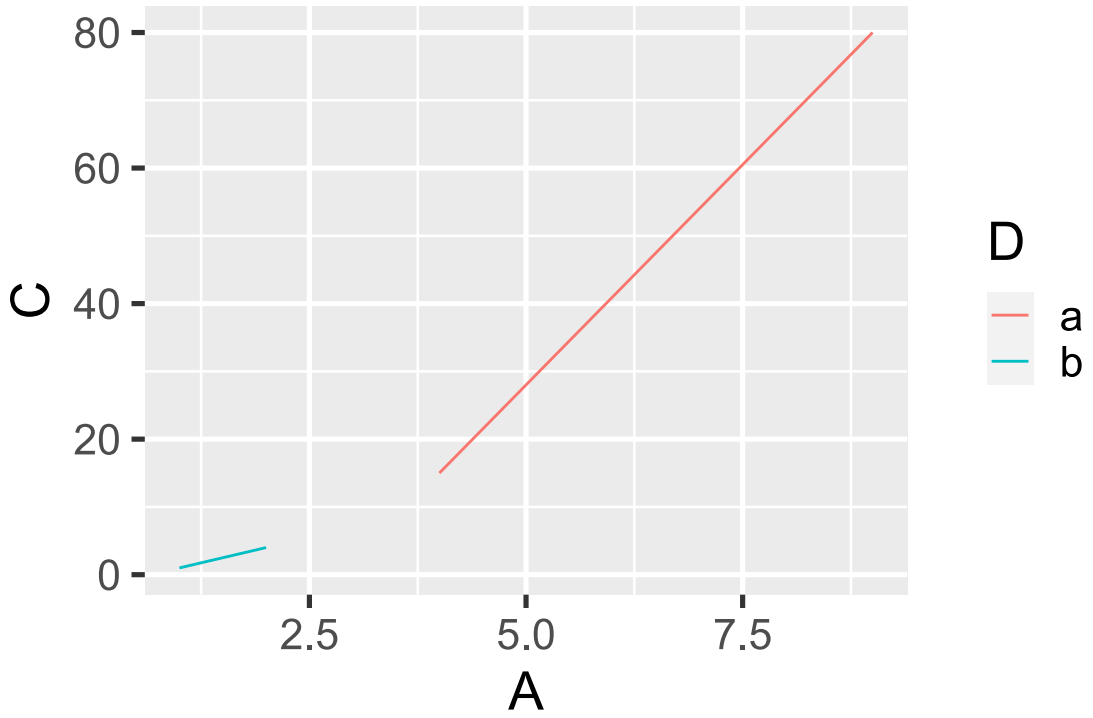
```
ggplot(data=data, mapping=aes(x=A, y=C))+
  geom_line()
```



Line Example

- We can still separate by column D in a line plot

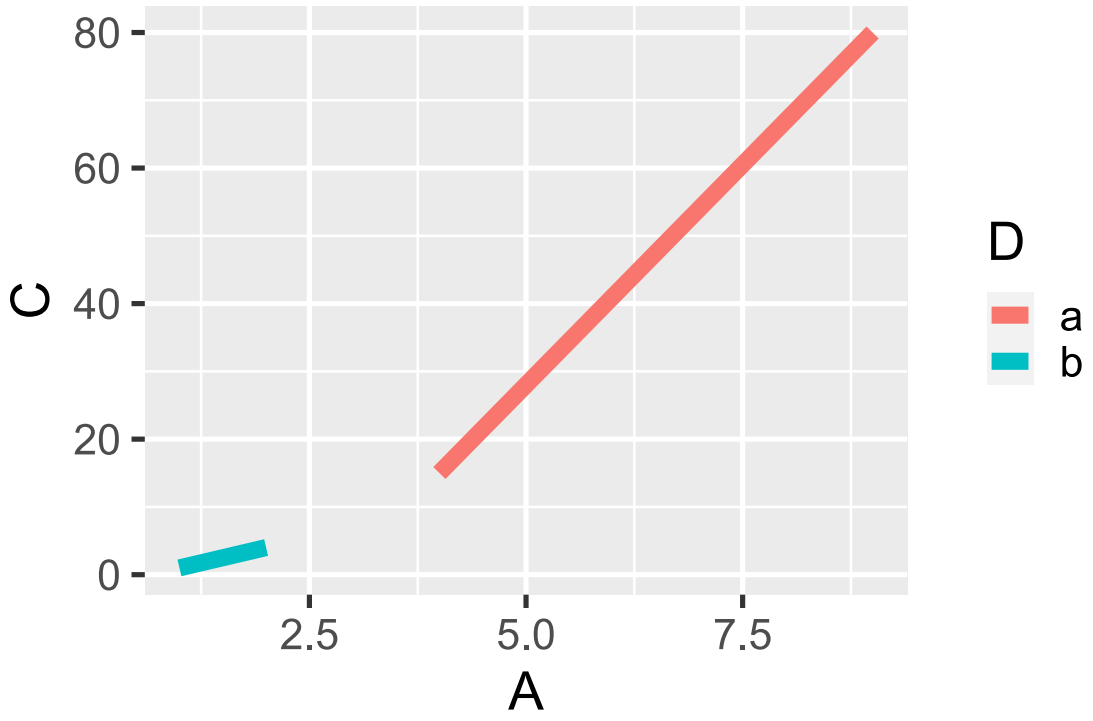
```
ggplot(data=data, mapping=aes(x=A, y=C, color=D))+
  geom_line()
```



Line Example

- These lines are a little thin. Let's make them bolder

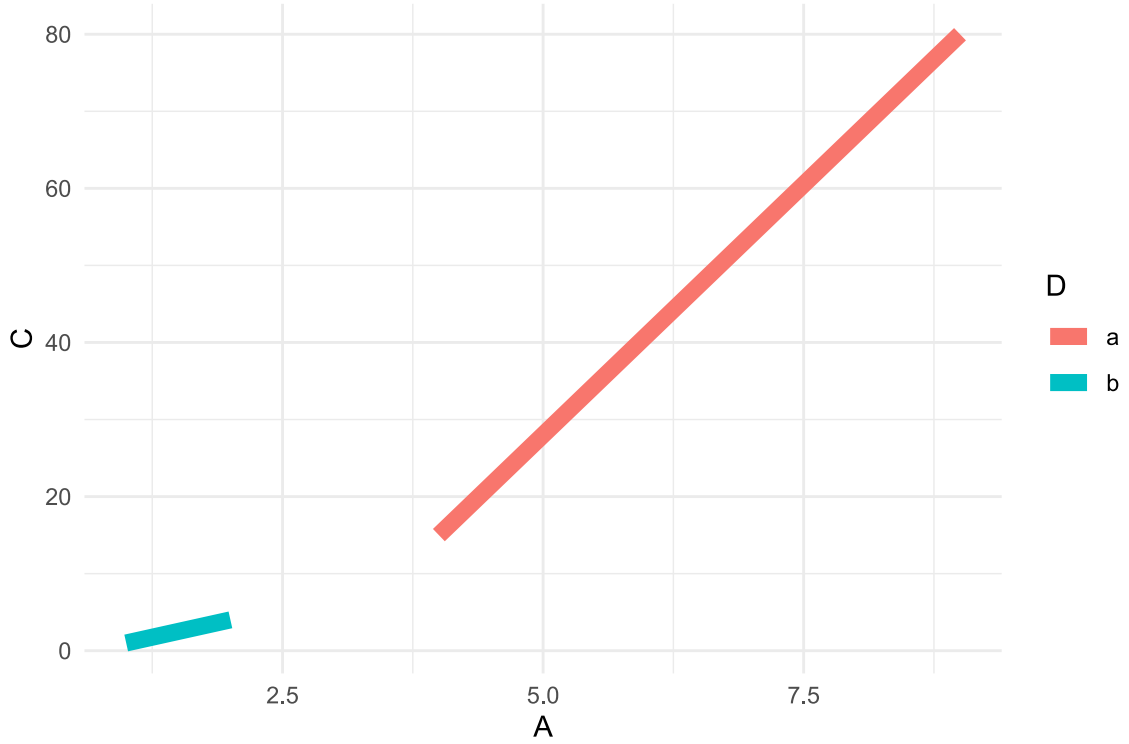
```
ggplot(data=data, mapping=aes(x=A, y=C, color = D))+
  geom_line(size = 3)
```



Line Example

- I don't like the background. Let's change the theme.

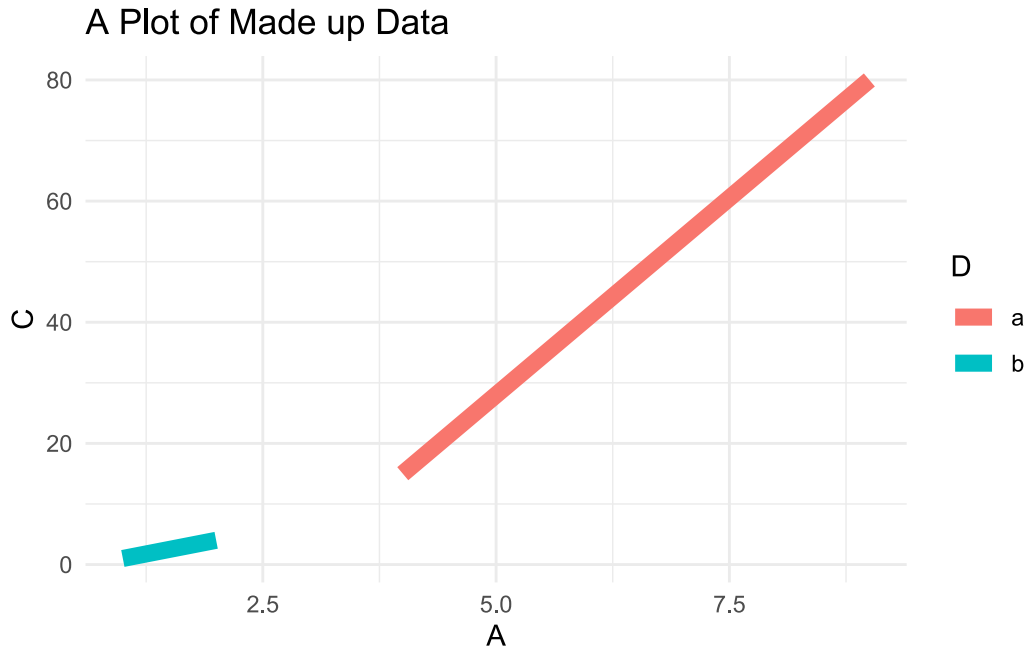
```
ggplot(data=data, mapping=aes(x=A, y=C, color = D))+
  geom_line(size = 3)+
  theme_minimal()
```



Line Example

- Finally, let's add some labels

```
ggplot(data=data, mapping=aes(x=A, y=C, color = D))+
  geom_line(size = 3)+
  theme_minimal()+
  ggtitle("A Plot of Made up Data")
```



Faceting

- Sometimes we may want to display multiple charts with similar data.

--

- For this we can use faceting

--

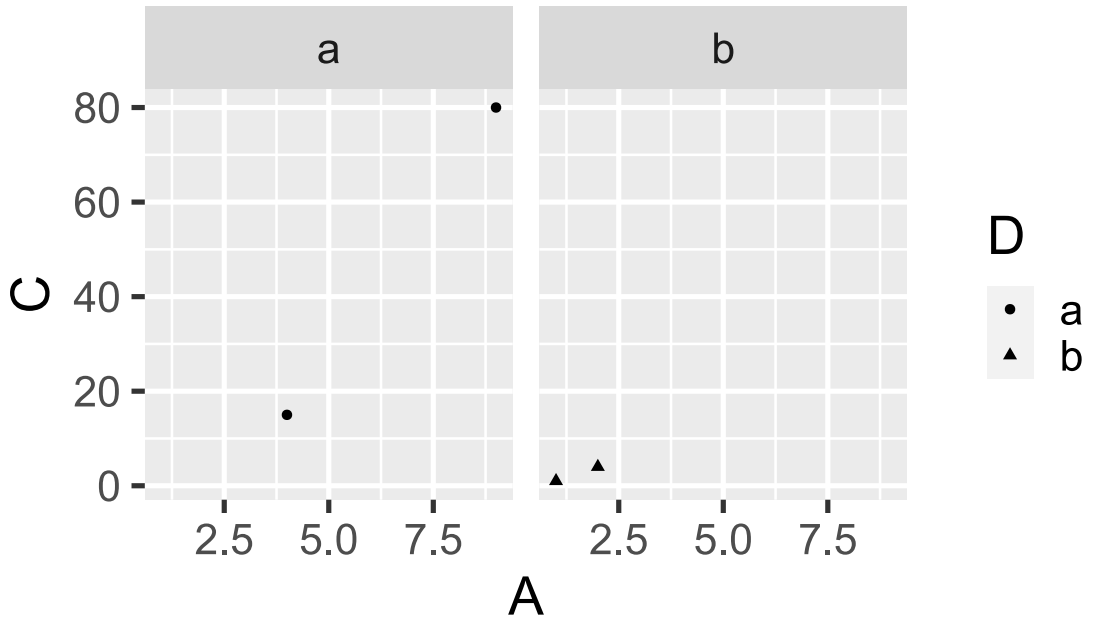
- This is especially useful whenever the Y-axis is the same.

Faceting

Faceting your plot into groups specified by D.

Each value of D (a or b) will be displayed on a different panel.

```
ggplot(data=data) +
  geom_point(mapping = aes(x = A, y = C, shape = D)) +
  facet_wrap(~D)
```



More Plots

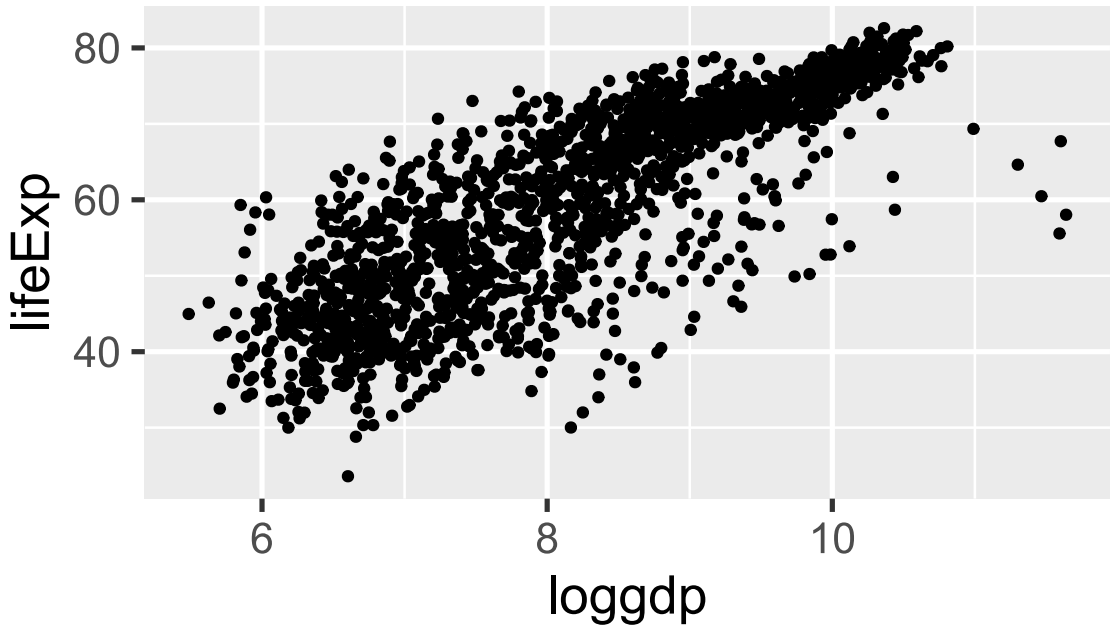
- Now let's use `gapminder` to make some more informative plots
- Suppose we are interested in the relationship between GDP per capita and life expectancy
- First, we should use `mutate` from last week to get log GDP per capita

```
gapminder ← gapminder %>% mutate(
  loggdp = log(gdpPercap)
)
```

More Plots

- now we can make a scatter plot

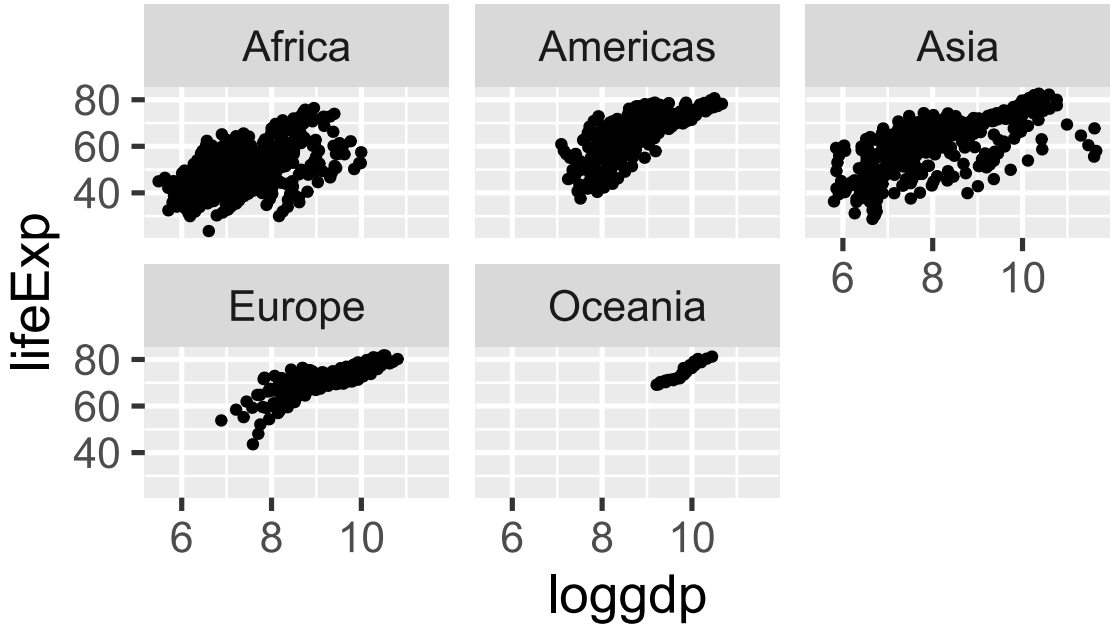
```
ggplot(data=gapminder, aes(x=loggdp, y=lifeExp))+
  geom_point()
```



Faceting again

- Let's separate this by continent

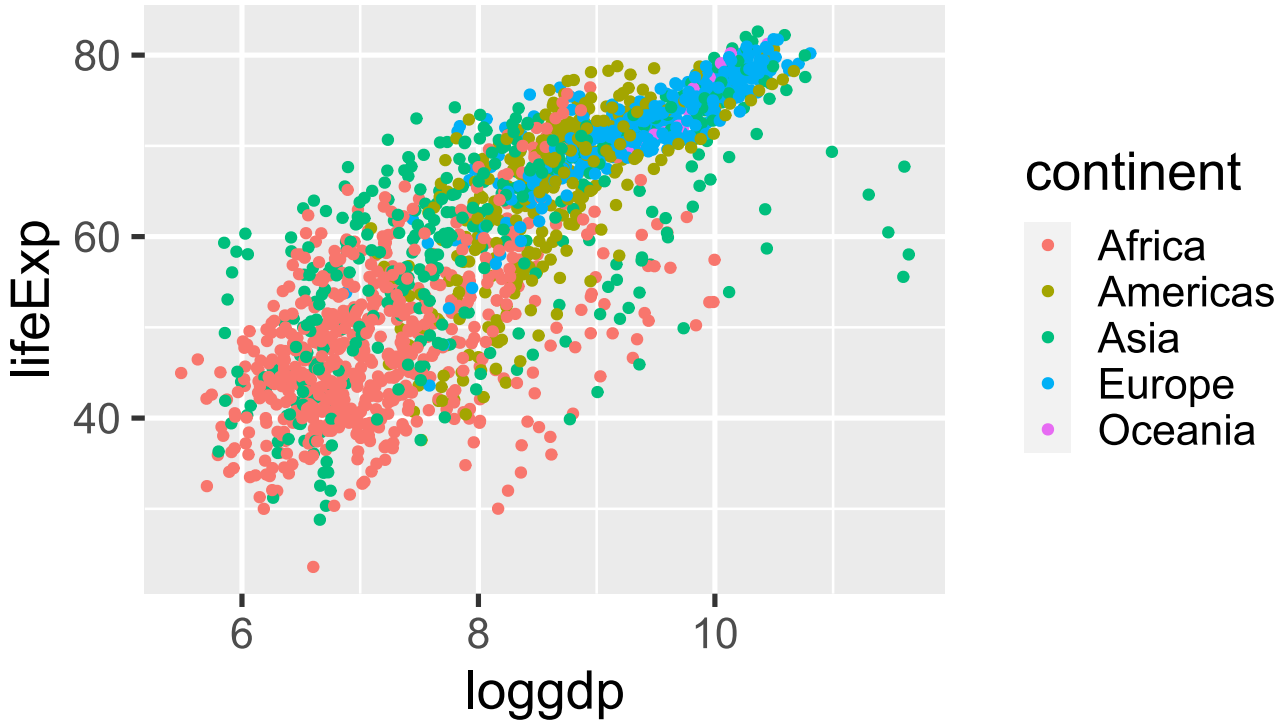
```
ggplot(data=gapminder, aes(x=loggdp, y=lifeExp))+
  geom_point()+
  facet_wrap(~continent)
```



More Plots

- Now let's separate by continent but on the same graph

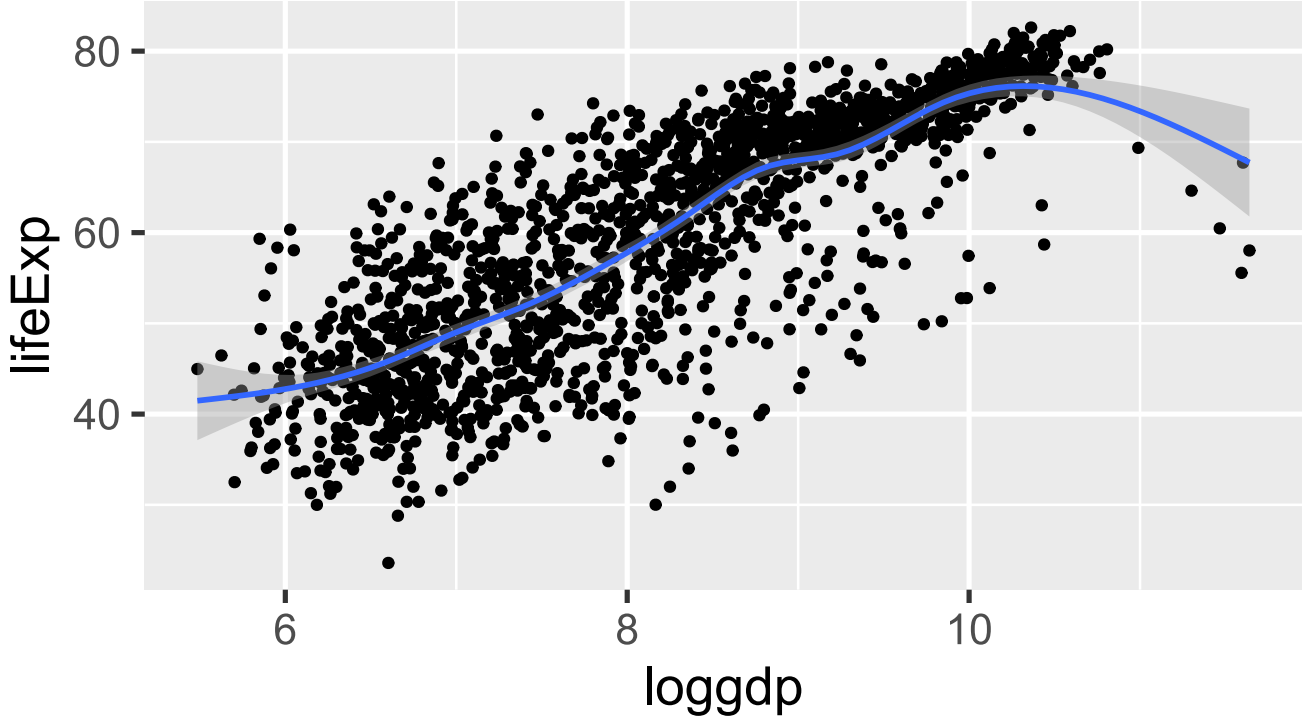
```
ggplot(data=gapminder, aes(x=loggdp, y=lifeExp, color=continent))+
  geom_point()
```



Layering Graphs

- Suppose we want to add a line to our plot

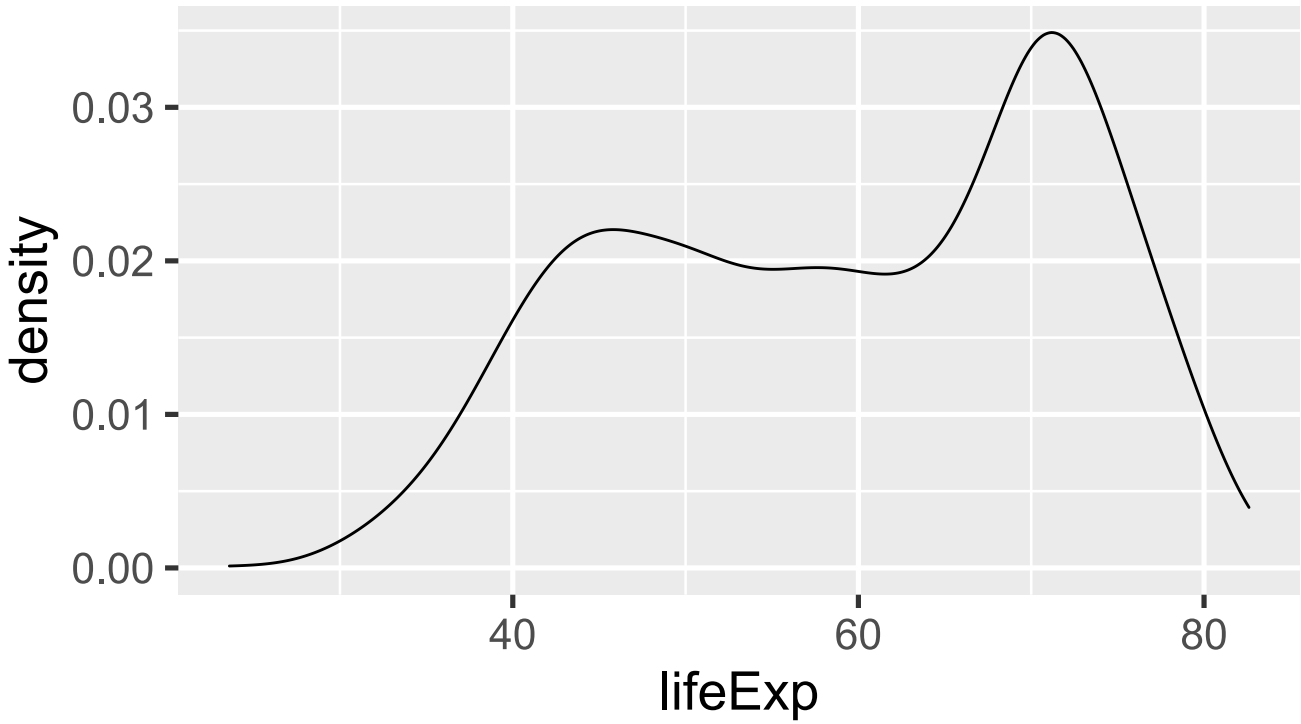
```
ggplot(data=gapminder, aes(x=loggdp, y=lifeExp))+
  geom_point()+
  geom_smooth()
```



Density Plots

- You might also be interested in the distribution of a variable. Let's do a density plot of Life Expectancy.

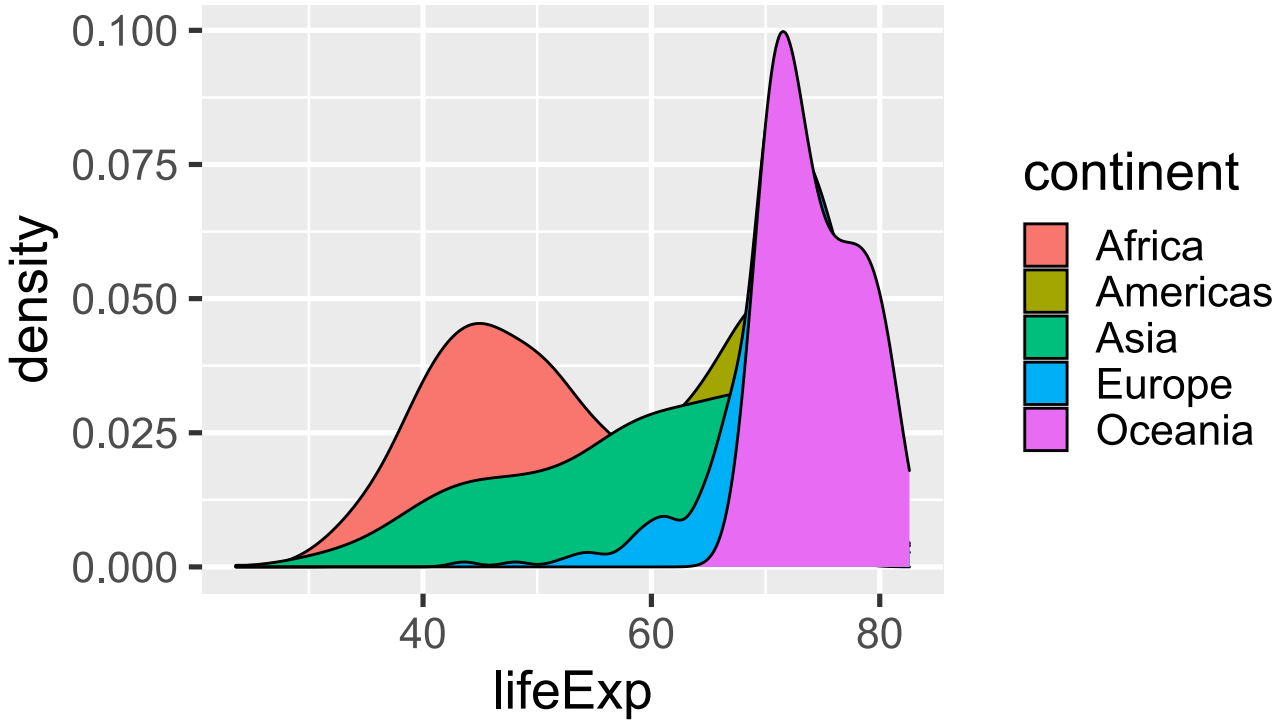
```
ggplot(data=gapminder, aes(x=lifeExp))+
  geom_density()
```



Density Plots

- Now lets separate the variable by continent.

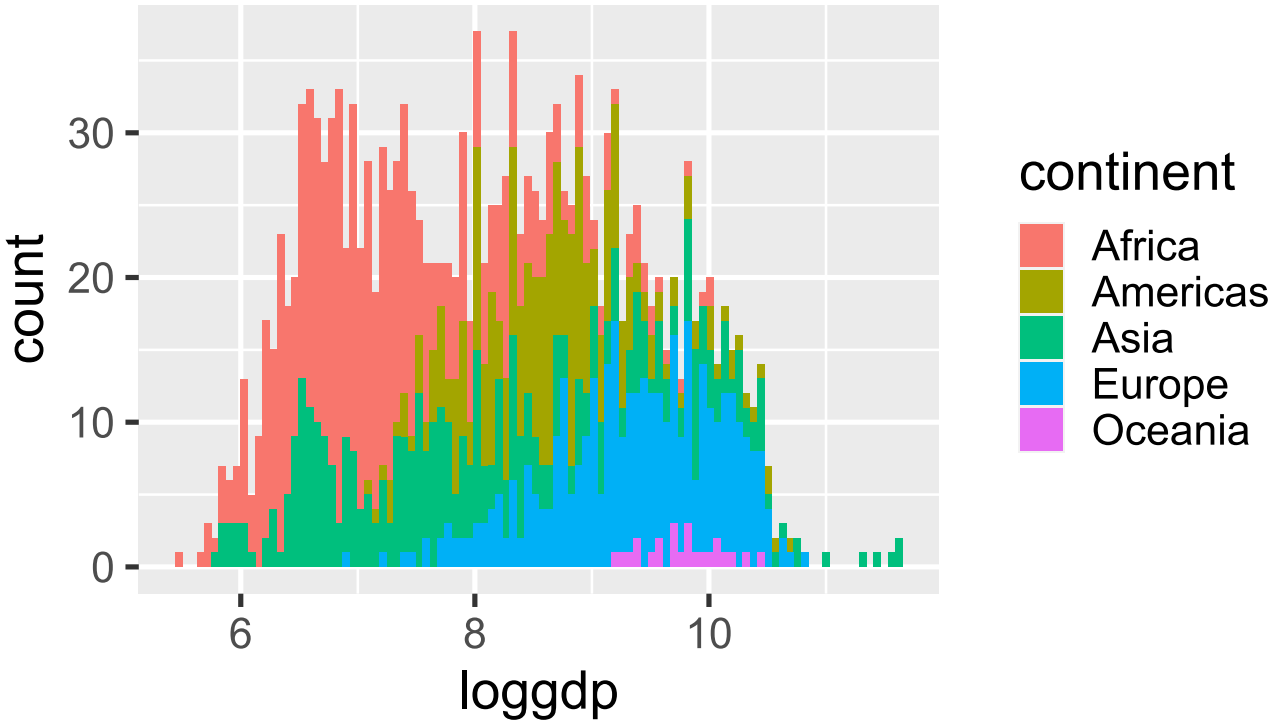
```
ggplot(data=gapminder, aes(x=lifeExp, fill=continent))+
  geom_density()
```



Histograms

- Let's make a histogram of log GDP per Capita on top of our density plot

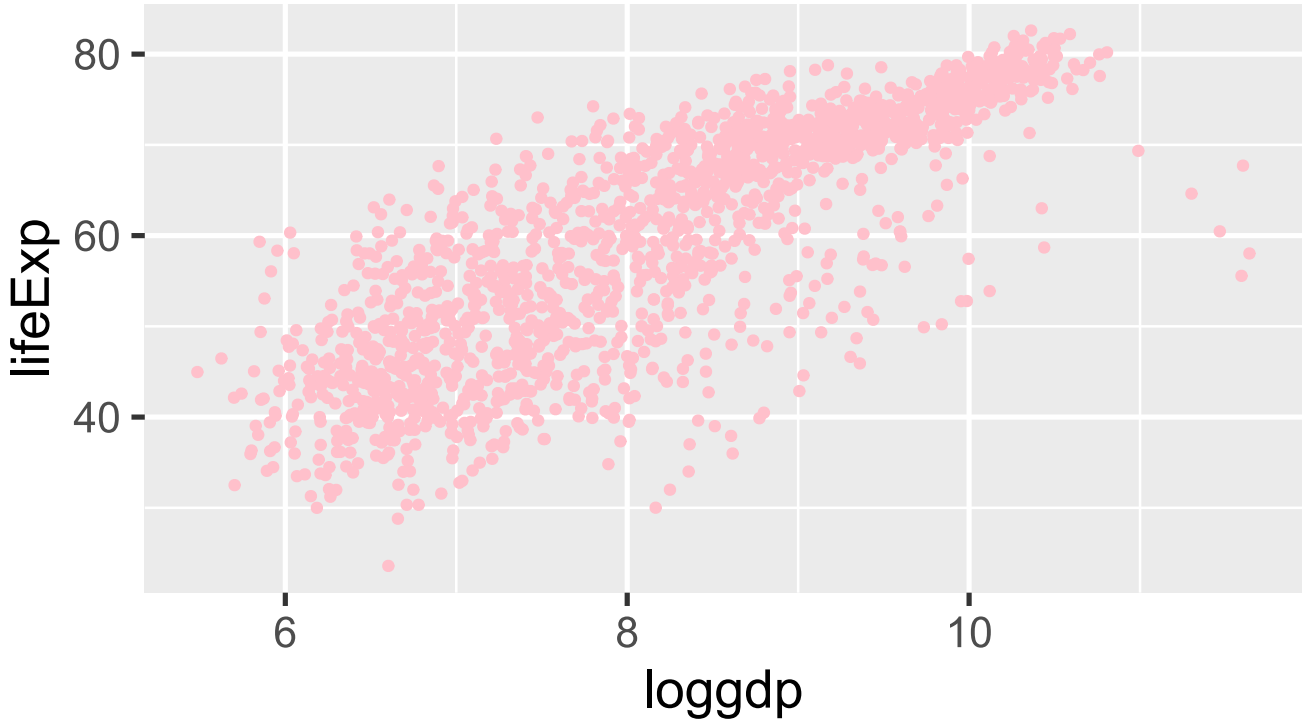
```
ggplot(data=gapminder, aes(x=loggdp, fill=continent))+
  geom_histogram(bins = 100)
```



More Fun with GGPlot2

1. Manually set colors

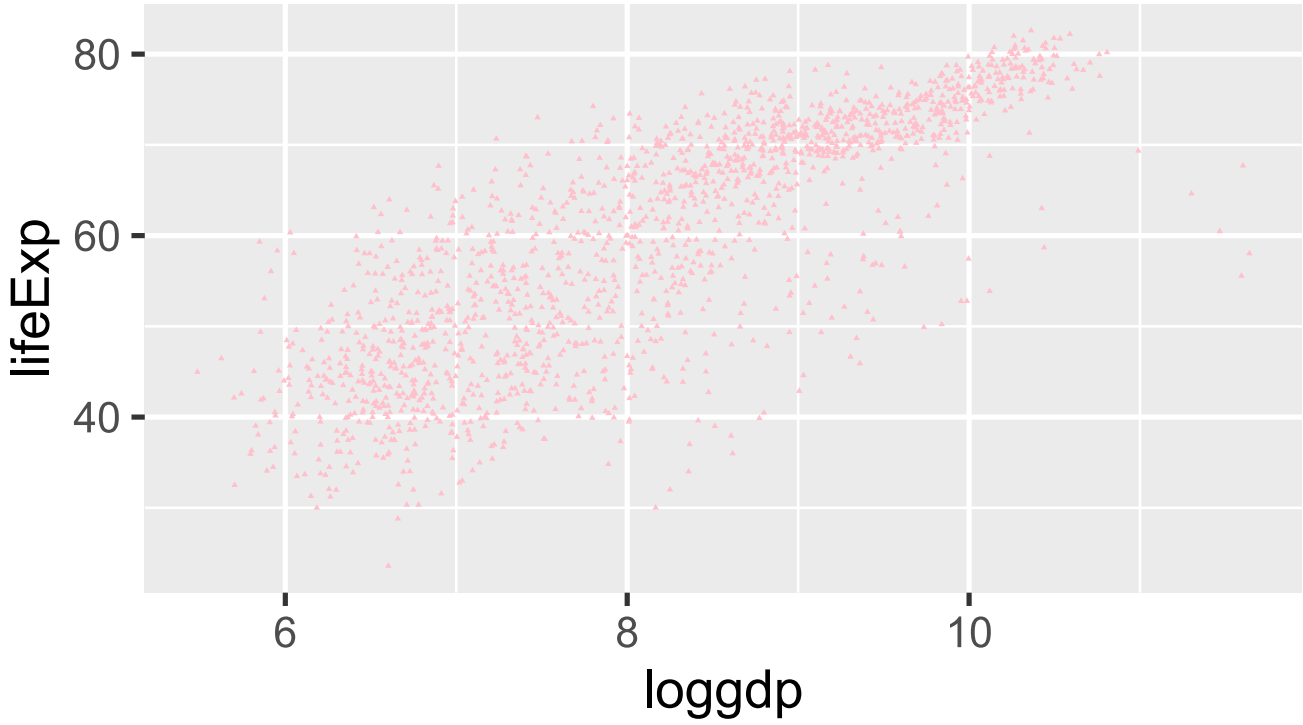
```
ggplot(data=gapminder, aes(x=loggdp, y=lifeExp))+
  geom_point(color="pink")
```



More Fun with GGPlot2

2. Change size and Shape of Points

```
ggplot(data=gapminder, aes(x=loggdp, y=lifeExp))+
  geom_point(color="pink", size = 0.5, shape="triangle")
```

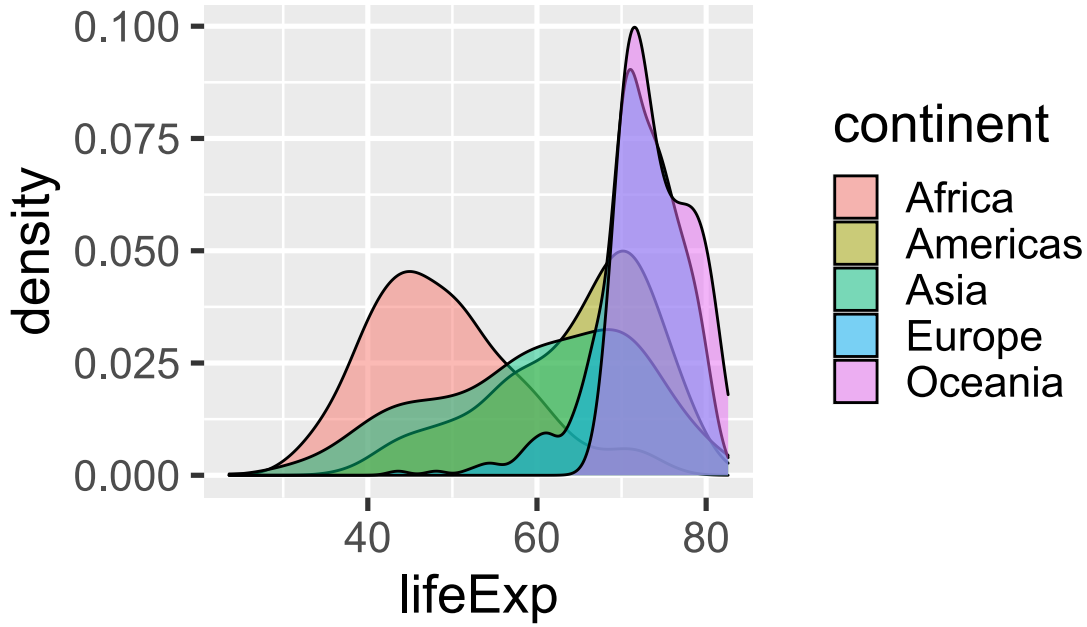


More Fun with GGPlot2

3. Change Transpancy

- Very helpful for density plots

```
ggplot(data=gapminder, aes(x=lifeExp, fill=continent))+
  geom_density(alpha = 0.5)
```

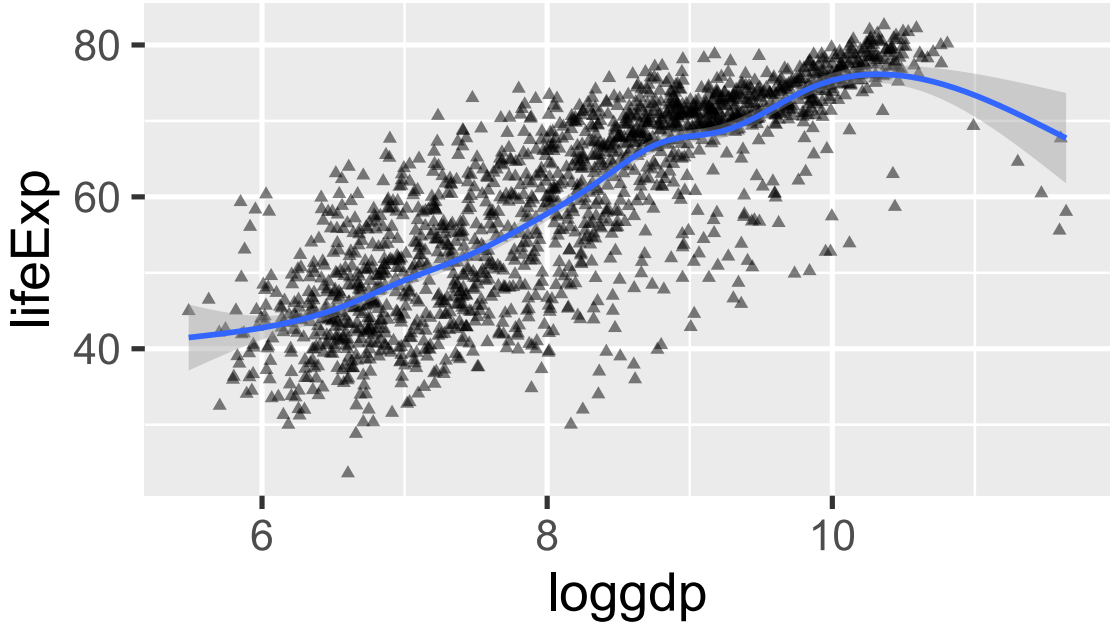


More Fun with GGPlot2

3. Change Transpancy

- Very helpful for layering

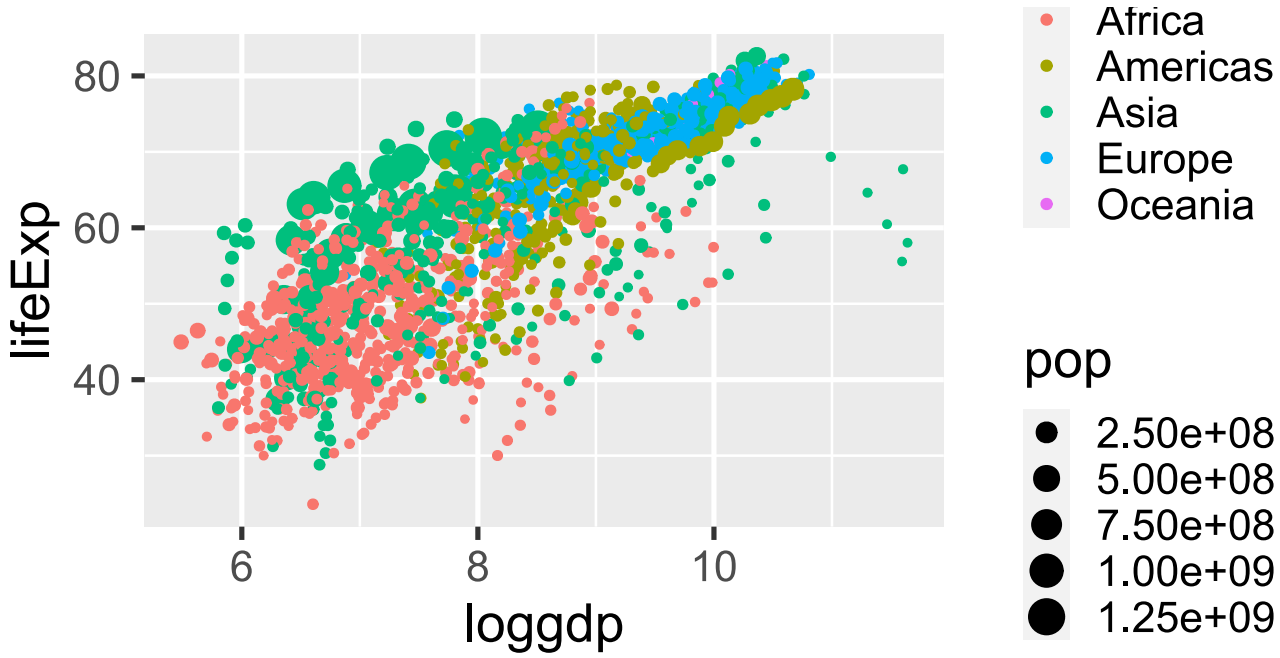
```
ggplot(data=gapminder, aes(x=loggdp, y=lifeExp))+
  geom_point(shape="triangle", alpha = 0.5) +
  geom_smooth()
```



More Fun with GGPlot2

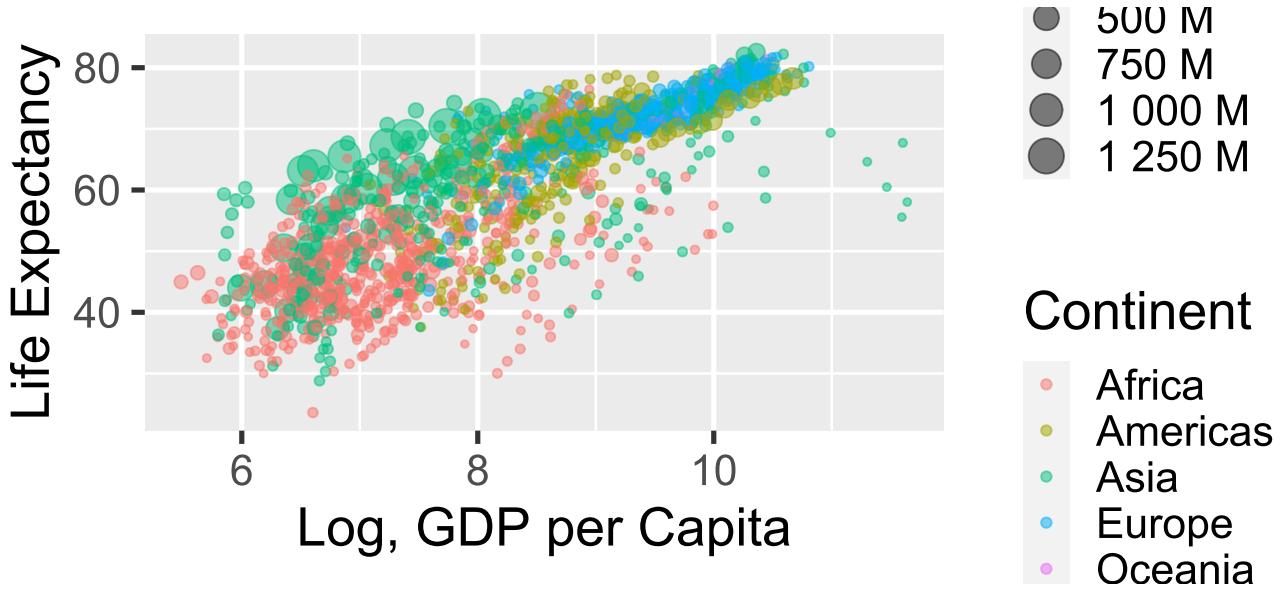
4. Vary Point Size

```
ggplot(data=gapminder, aes(x=loggdp, y=lifeExp))+
  geom_point(aes(color=continent, size=pop))
```



More Fun with GGPlot2

```
ggplot(data=gapminder, aes(x=loggdp, y=lifeExp))+
  geom_point(aes(color = continent, size = pop), alpha=0.5)+
  scale_size_continuous(labels = scales::unit_format(unit="M", scale=1e-6))+
  xlab("Log, GDP per Capita")+ ylab("Life Expectancy")+
  labels(size="Population", colour="Continent")
```



Tools for Using ggplot2

- A Layered Grammar of Graphics
- R4DS
- Datacamp
- R Graph Gallery

Final Thoughts

- There are so many more thing you can create using ggplot2
- Start with simple graphs and then work your way up to more sophisticated ones
- Don' forget to use help in **R** to know what can go into each different `geom` function