ggplot

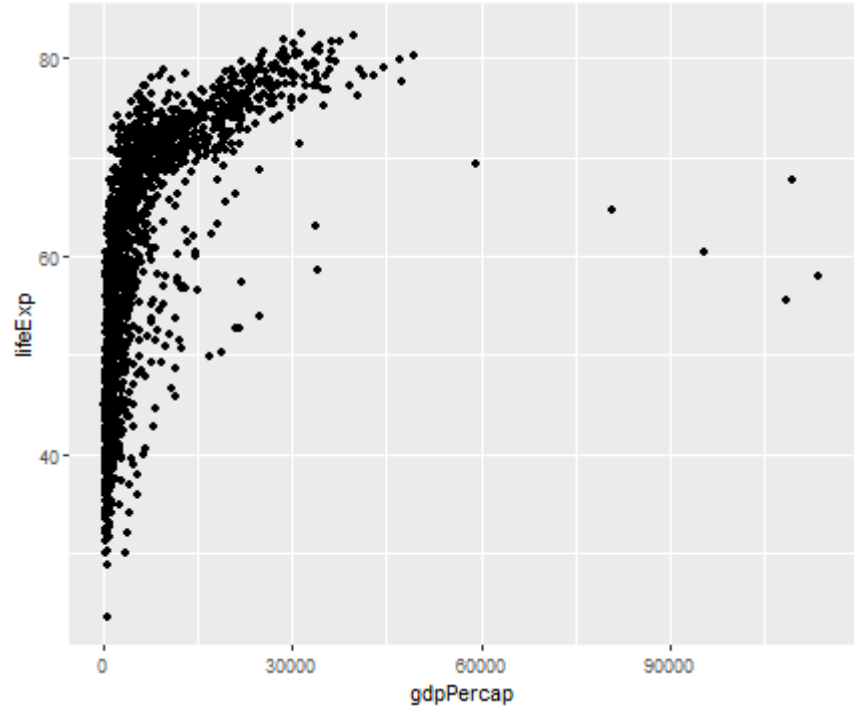
Amanda and Pilar

09/10/2020

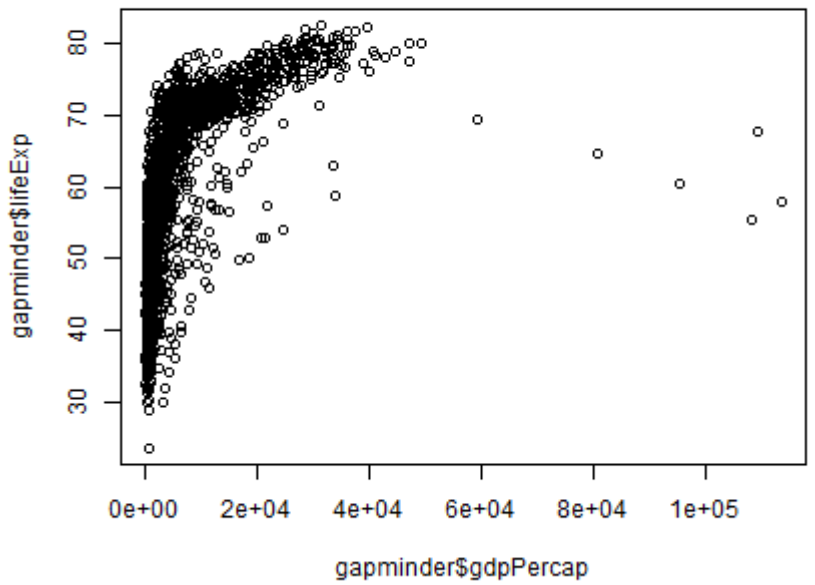
Intro to ggplot

Compare a simple ggplot with a base R plot.

```
gapminder %>% ggplot(aes(gdpPercap, lifeExp))
```




```
plot(gapminder$gdpPercap, gapminder$lifeExp)
```



Intro to Ggplot

Now compare it with this:

Getting started

First, make sure you have the package installed and loaded. Remember that by loading 'tidyverse' you'll be including both ggplot and dplyr (and the other packages from the collection).

Next, install and load the package "gapminder".

Gapminder is a reduced version of the full data found at gapminder.org. Take a first look at the data with some of these commands: glimpse(), View(), names(), head().

```
glimpse(gapminder)
```

```
## Rows: 1,704
## Columns: 6
## $ country   <fct> Afghanistan, Afghanistan, Afghanistan, Afghanistan, Afgha...
## $ continent <fct> Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asi...
## $ year      <int> 1952, 1957, 1962, 1967, 1972, 1977, 1982, 1987, 1992, 199...
## $ lifeExp   <dbl> 28.801, 30.332, 31.997, 34.020, 36.088, 38.438, 39.854, 4...
## $ pop       <int> 8425333, 9240934, 10267083, 11537966, 13079460, 14880372,...
## $ gdpPercap <dbl> 779.4453, 820.8530, 853.1007, 836.1971, 739.9811, 786.113...
```

The Basic Structure

ggplot works by combining several functions using the + operator. This creates *layers* in an additive fashion. Each function does something specific: provide a dataset, create a geometric object, add labels, add scales, change the coordinate system or layout, change the color palette, etc.

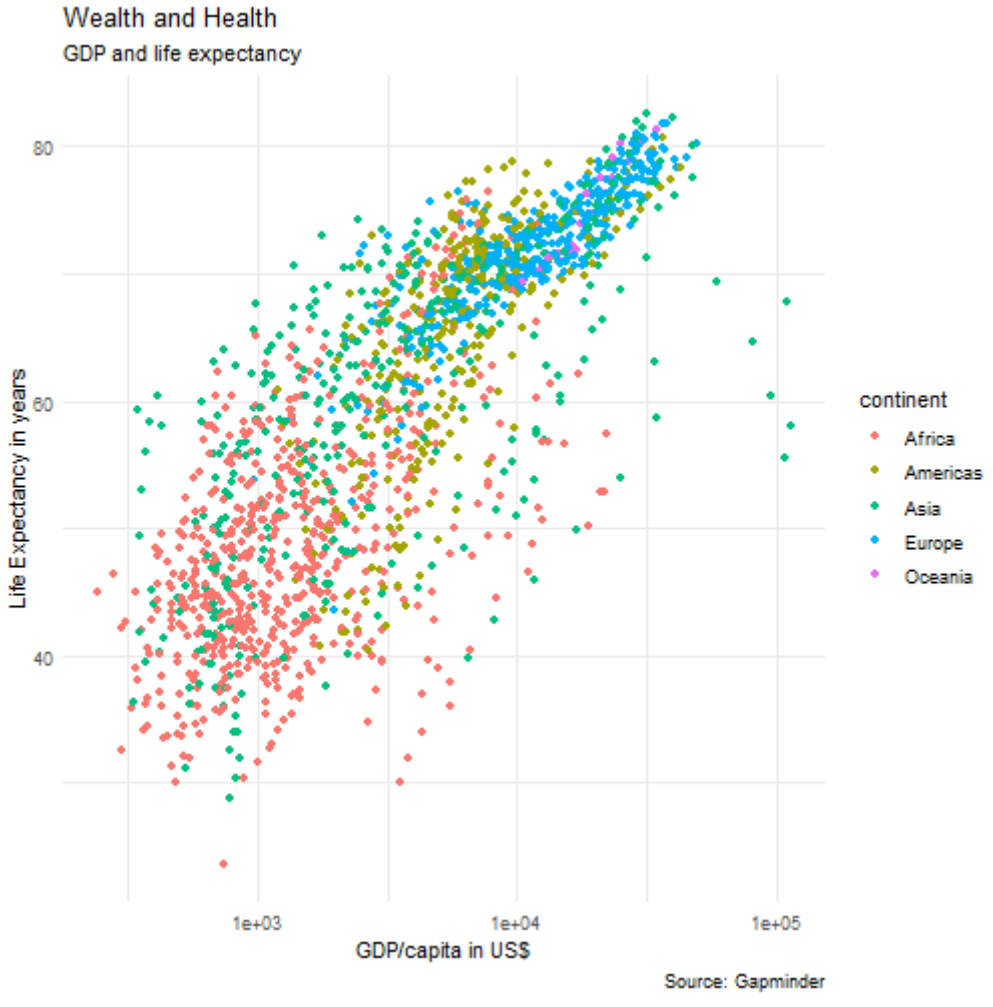
ggplot(name of dataset), aes (x= name variable x, y= name variable y, ...) + geom_something()

The Basic Structure

ggplot(name of dataset), aes (x= name variable x, y= name variable y, ...) + geom_something()

- Name of dataset: just type the name of the dataset that contains the variables you are interested in graphing.
- aes(): *what* you want to plot (variables) and what you want to *see* (color, fill, shape, linetype, and size. aes stands for 'aesthetics'.
- geom_something: *how* you want to plot it. Geometric objects are the actual marks we put on a plot. You need at least one but can have as many as you want! Common examples:
    - geom_point: points for scatter plots, dot plots, etc.
    - geom_line: lines for time series, trend lines, etc.
    - geom_boxplot: boxplots
- other options: titles, themes, axis adjustment, legends, etc.

```
gapminder %>%
  ggplot() +
  aes(x=gdpPercap) +
  aes(y=lifeExp)+
  scale_x_log10() +
  geom_point() +
  geom_point(aes(color=continent)) +
  labs(title="Wealth and Health") +
  labs(subtitle = "GDP and life expec
  labs(caption="Source: Gapminder") +
  xlab("GDP/capita in US$")+
  ylab("Life Expectancy in years")+
  theme_minimal()
```



Step by Step

The example we just saw is fleshed out at its maximum to illustrate what each step does. In practice, you can get this done in fewer lines.

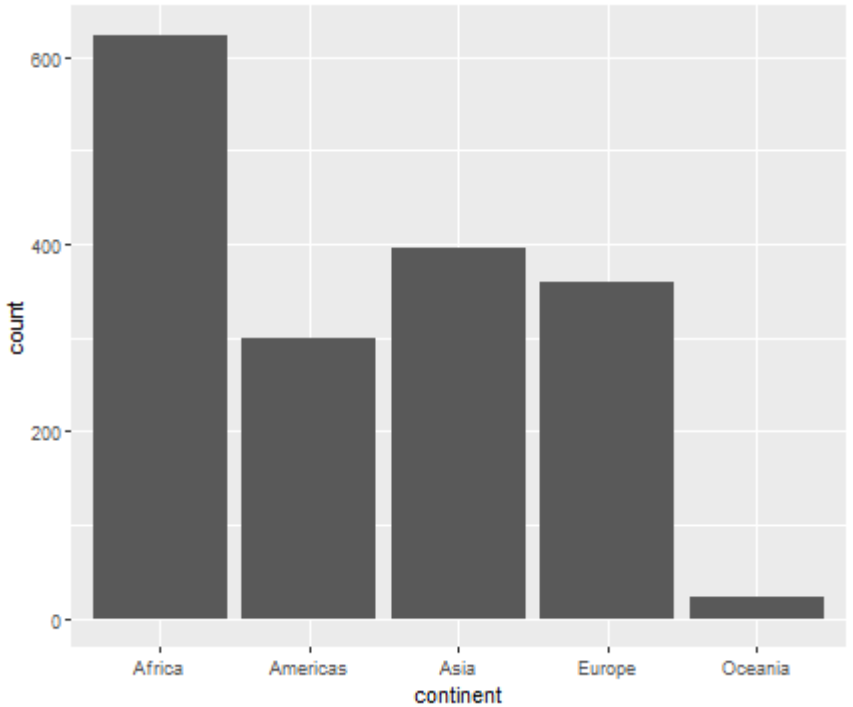
Both of these codes produce the exact same graph. Take a second to identify what's different from them:

```
gapminder %>%
  ggplot(aes(gdpPercap, lifeExp)) +
  scale_x_log10() +
  geom_point(aes(color = continent)) +
  labs(
    title = "Wealth and Health",
    subtitle = "GDP and life expectancy",
    caption = "Source: Gapminder",
    x = "GDP/capita in US$",
    y = "Life Expectancy in years"
  ) +
  theme_minimal()
```

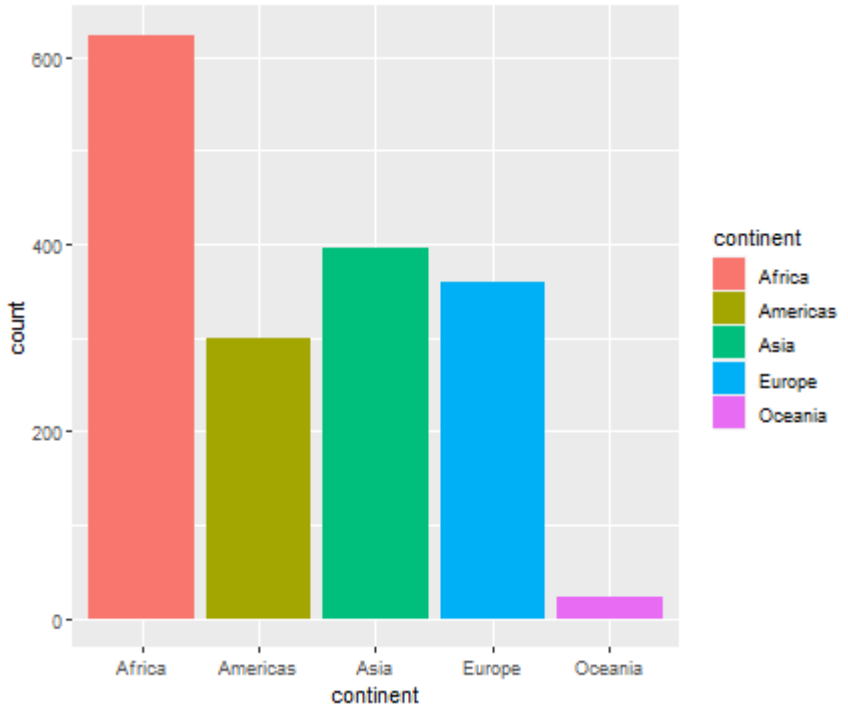
```
gapminder %>%
  ggplot() +
  aes(x=gdpPercap) +
  aes(y=lifeExp)+
  scale_x_log10() +
  geom_point() +
  geom_point(aes(color=continent)) +
  labs(title="Wealth and Health") +
  labs(subtitle = "GDP and life expectancy")
  labs(caption="Source: Gapminder") +
  xlab("GDP/capita in US$")+
  ylab("Life Expectancy in years")+
  theme_minimal()
```

ggplot: one categorical variable

```
ggplot(gapminder, aes(x=continent)) + geom_ba
```

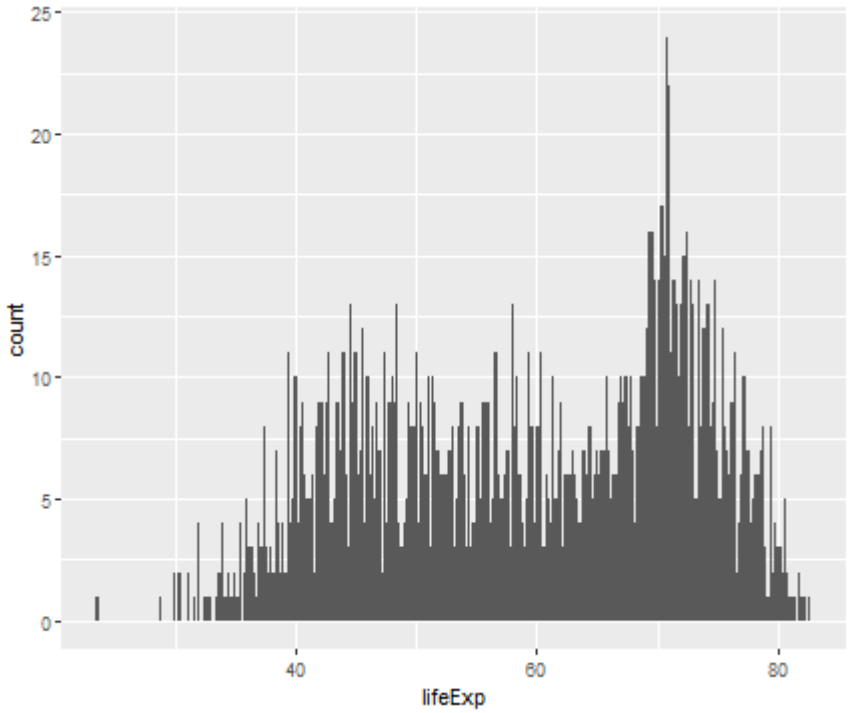



```
ggplot(gapminder, aes(x=continent,fill= conti
```



Plotting a continuous variable

```
ggplot(data=gapminder, aes(x=lifeExp)) +
    geom_histogram(binwidth = .2)
```

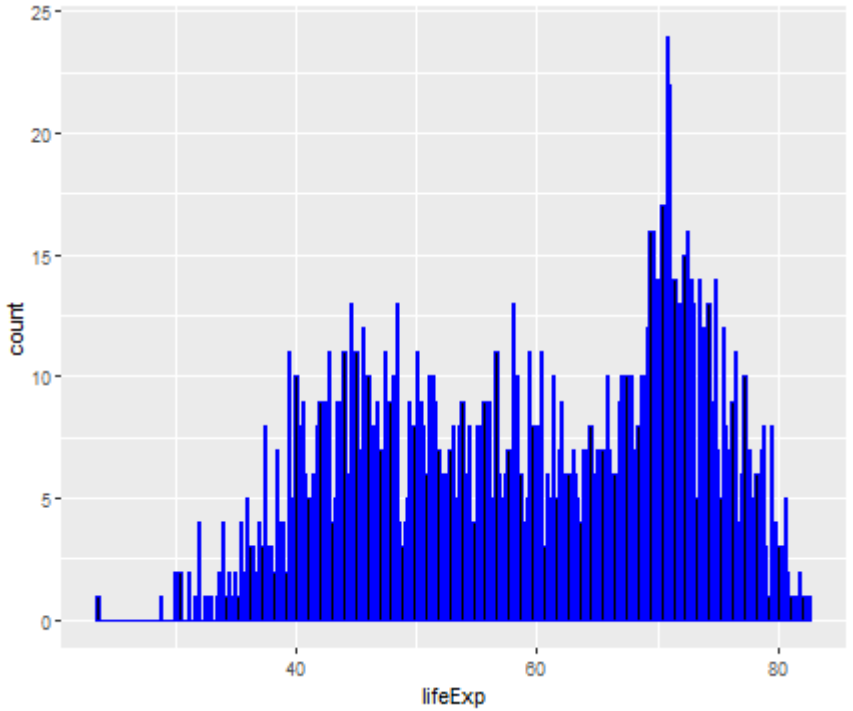


PRACTICE: Change the binwidth to different values. What does this do?

Plotting a continuous variable

Now let's add some color.

```
ggplot(data=gapminder, aes(x=lifeExp)) +
    geom_histogram(binwidth = .2, color= "blue", fill="black")
```



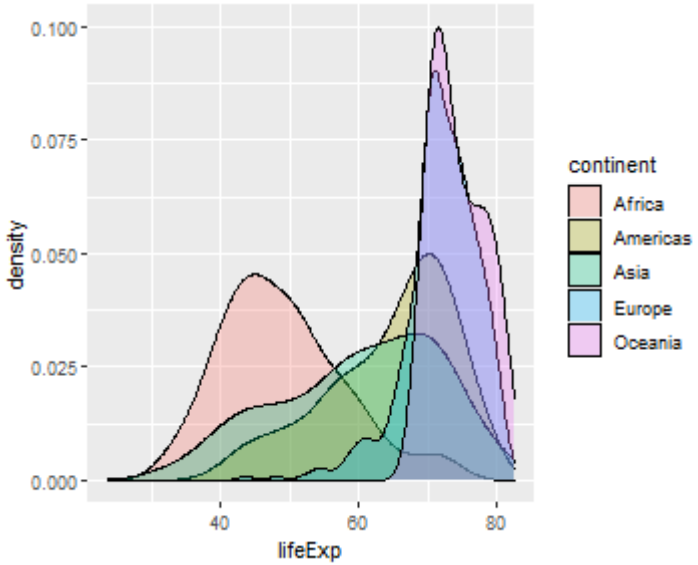
ggplot: Plotting a continuous variable

PRACTICE: What if we want a density plot instead of a histogram? Google what geom_xxx we need to get this done?

ggplot: Plotting a continuous variable

How about one density plot per continent?

```
ggplot(data=gapminder, aes(x=lifeExp, fill=co
    geom_density(alpha=0.3)
```

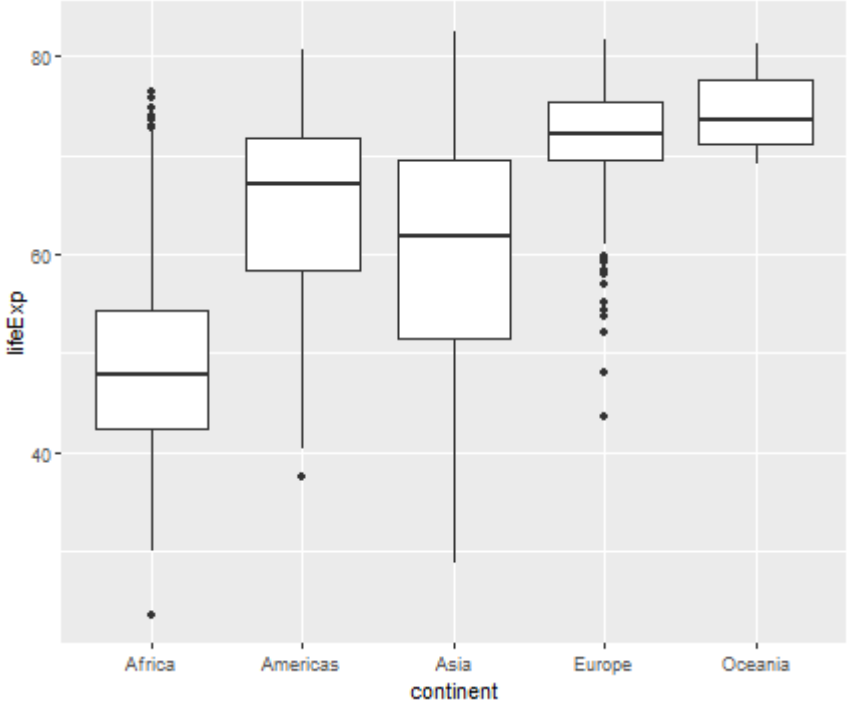


PRACTICE: Change the alpha value and see what that does.

Plotting a continuous vs a categorical

Boxplots are useful when we have a categorical X and a continuous Y (or viceversa)

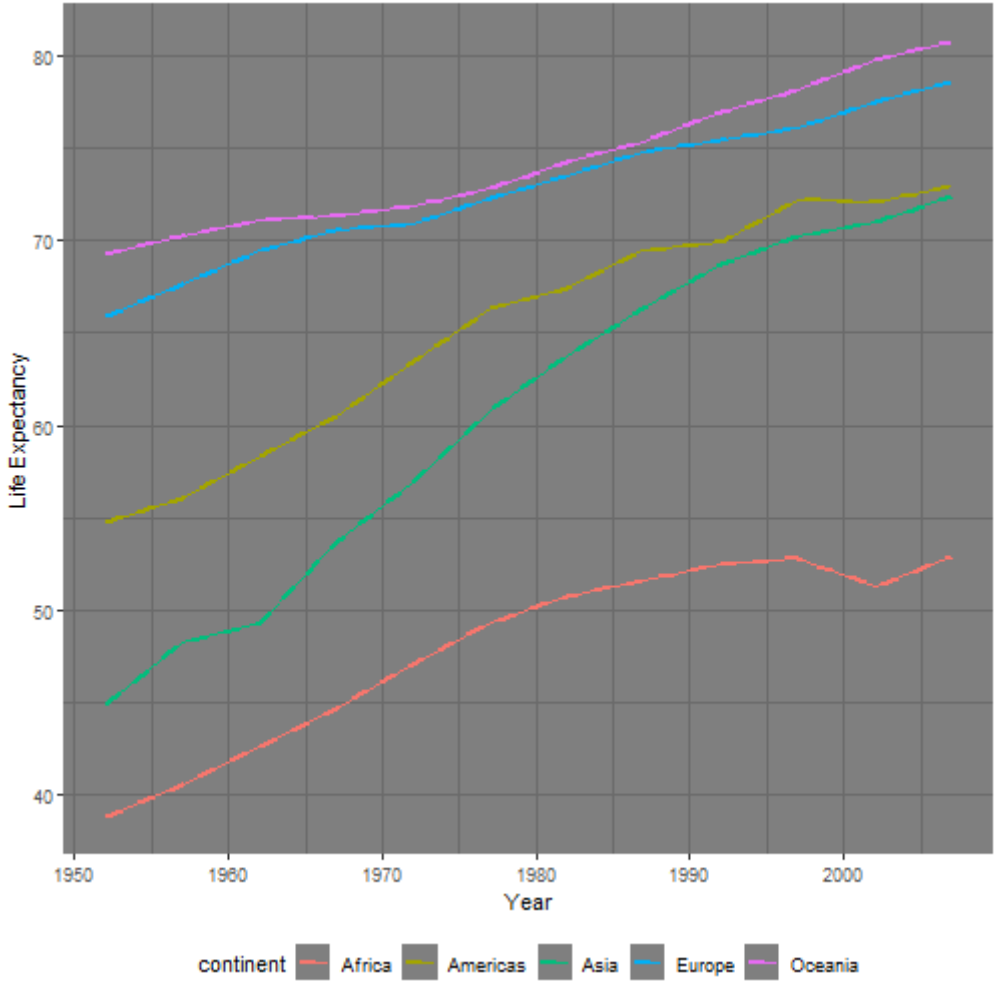
```
ggplot(data=gapminder, aes(x=continent, y=li
 geom_boxplot()
```



PRACTICE: Give it some color (hint: see the colorful bar graph above).

Check out the 'violin' plot in https://www.r-graph-gallery.com/ and try to make one with this same data.

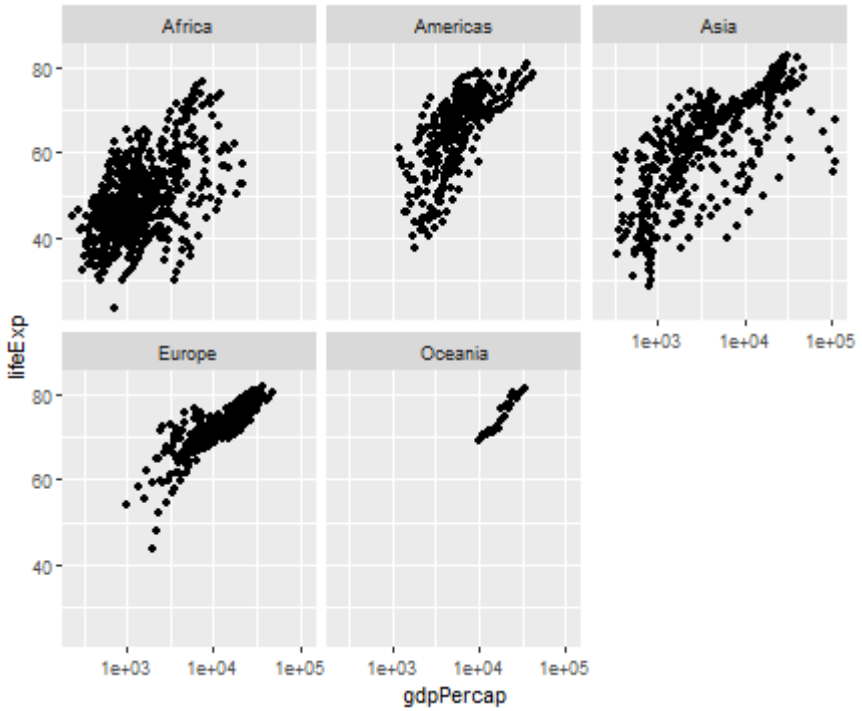
```
gapminder %>%
  group_by(continent, year) %>%
  summarise(lifeExp = median(lifeExp)
  ggplot() +
  aes(x= year) +
  aes(y=lifeExp) +
  aes(color = continent) +
  geom_line() +
  geom_line(size = 1, linetype="dashe
  labs(x = "Year", y = "Life Expectar
  theme_dark() +
  theme(legend.position = "bottom")
```



ggplot: Multiple Plots

Sometimes we have too much on a single graph, and that makes things messy. We can use the 'facet_wrap' option:

```
gapminder %>%
  ggplot(aes(gdpPercap, lifeExp)) +
  scale_x_log10() +
  geom_point() +
  facet_wrap(~continent)
```



PRACTICE

Make the above plot nicer:

1) Fix the axis labels

2) Add some color

3) Change the axis so that it's not in scientific notation (google: options(scipen =))

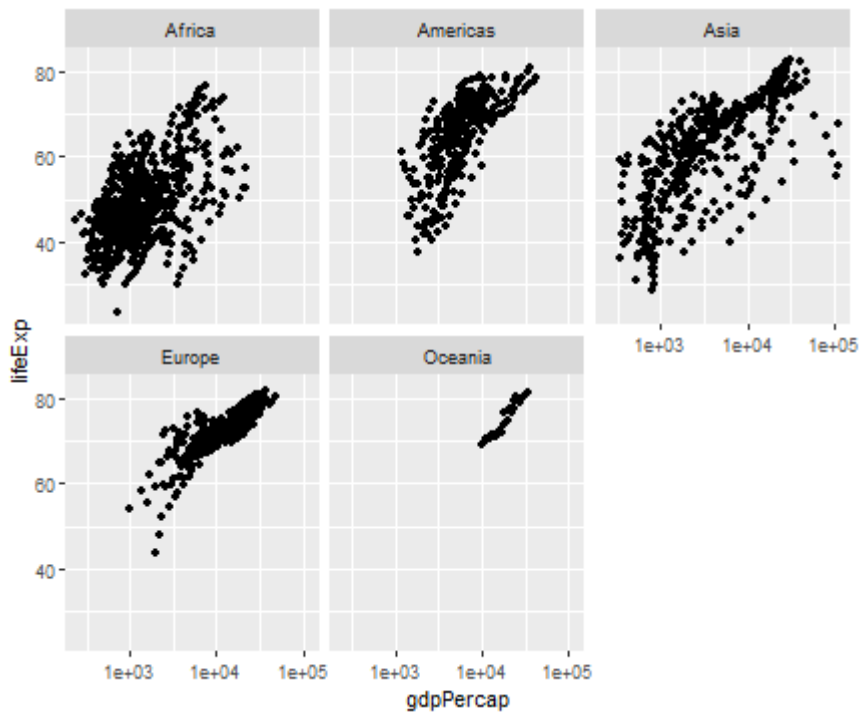
4) Add a theme

Saving graphs

1) Create an object with the plot

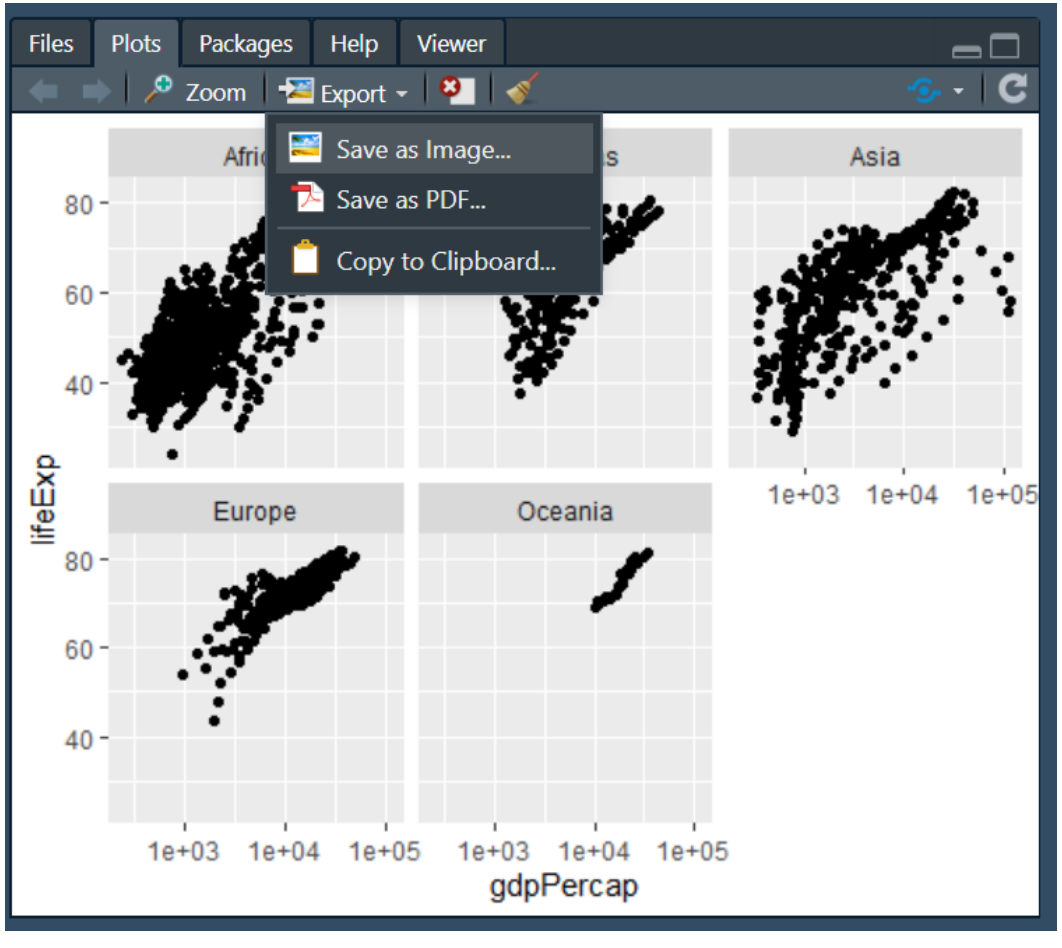
```
plot1 <- gapminder %>%
  ggplot(aes(gdpPercap, lifeExp)) +
  scale_x_log10() +
  geom_point() +
  facet_wrap( ~ continent)
```

```
plot1
```



Saving graphs

2) Save it from the Viewer Panel



Saving graphs

3) Use the 'ggsave' function.

```
plot1 <- gapminder %>%
  ggplot(aes(gdpPercap, lifeExp)) +
  scale_x_log10() +
  geom_point() +
  facet_wrap( ~ continent) +
  ggsave("plot1.png")
```

PRACTICE: Find this plot in your files.

MORE PRACTICE 

Using the World Value Survey:

1) Create a subset with the following variables: B_COUNTRY_ALPHA, A_YEAR, Q106-Q109, Q240, Q184-Q186. (Refer to codebook to see what these variables are.)

2) Keep only observations from the following countries: US, Colombia, Tajikistan, Japan, Zimbabwe

3) Check the dimensions of this new dataset

4) Rename the following variables:

- Q106 to "income_eq"
- Q240 to "left_right"
- Q184 to "just_abort"

5) Make a table of "left_right" for the whole sample, then for each country (hint: make a crosstab of B_COUNTRY_ALPHA and left_right). BONUS: convert this crosstab into percentage per column.

6) Find the mean of "left_right" for the total sample. Then find the mean of "left_right" in each country.

7) Now calculate the mean of "income_eq" and "just_abort" for each country. What does this tell us about liberal-conservative values?

MORE PRACTICE 

With the subset of the wvs we've created:

1) Plot a bar graph with the number of observations per country

2) Plot a histogram with the distribution of left_right for each country (all in the same plot)

3) Make a violin plot of the distribution of "income_eq" for each country (all in same plot).

4) Make a box plot of the distribution of "just_abort" for each country (all in same plot).

With the complete dataset:

6) Make a scatter plot of q106(income_eq) and q184(just_abort). Hint: calculate the mean for each country first.

7) Google how to add the country labels to the scatter plot.