DATA VIZ

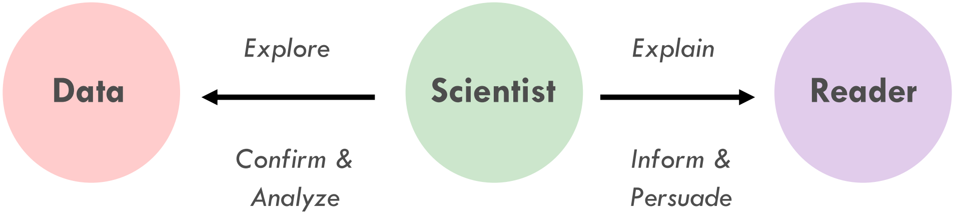
Research Methods in Psychology I & II ▪ Department of Psychology ▪ Colorado State University

BY THE END OF THIS SECTION YOU WILL:

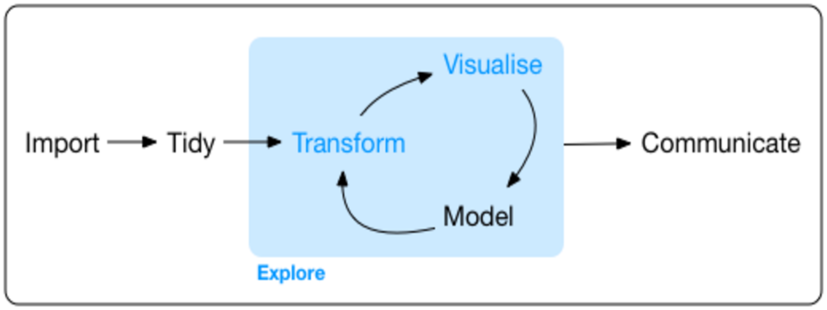
1. Know the grammar of graphics.
2. Be comfortable with using ggplot2 to create graphs in R.
3. Be familiar with a variety of different useful plots for visualizing data.
4. Have a sense of the importance of visualizing data.

What is Data Visualization?

Data visualization is the act of producing meaningful and aesthetically pleasing graphics of your data. In this class, we will use data visualization for exploration (for ourselves/our research team) and explanation (for the readers and consumers of our research).



Rick Scavetta, Data Visualization with ggplot2



Wickham & Grolemund—R for Data Science

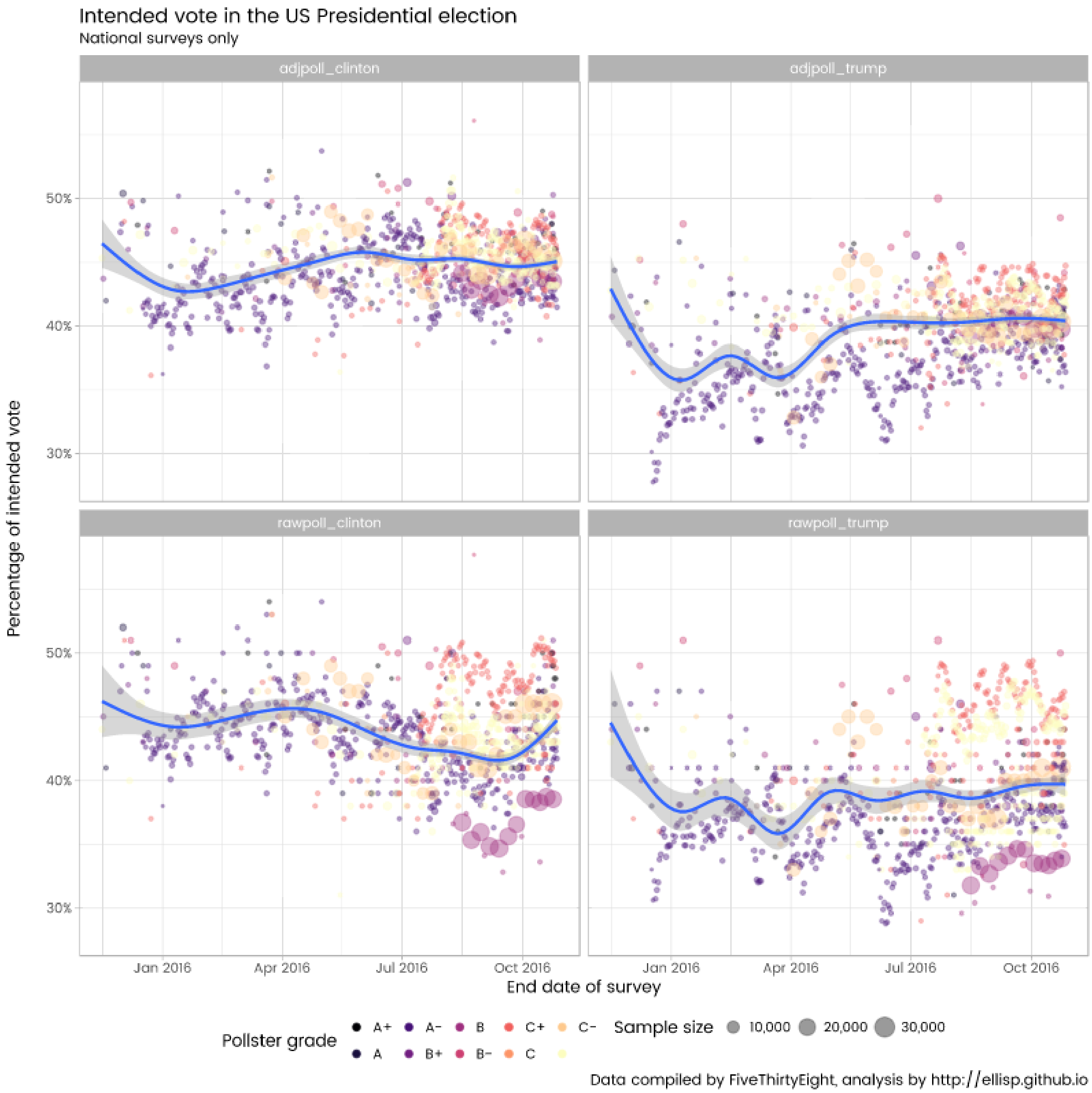
The Grammar of Graphics with ggplot2

ggplot2 is a package for R, it was automatically loaded when we installed the tidyverse. There are a number of plotting systems in R (including base R as well as other packages), but ggplot2 is the best, all purpose plotting system. It was developed to implement the graphing framework put forth by Leland Wilkinson in his book entitled "The Grammar of Graphics" (http://www.springer.com/us/book/9780387245447).

The grammar of graphics and ggplot2 are built around 7 elements (listed in the table below) of a graph — think about each element/layer as providing a piece of the graphic —and when they come together they tell the complete story of the data.

Layers	Description
Data	The data to be plotted.
Aesthetics	The mapping of variables to elements of the plot.
Geometrics	The visual elements used to display the data — for example, the data points in a scatterplot, the lines in a line chart, the bars in a bar plot.
Facets	Plotting of small groupings across tiles — i.e., creation of subplots within the main plot.
Statistics and Statistical Transformations	Representations of the data to aid understanding — e.g., plot means rather than raw data. We can also use this layer to plot transformed scores (e.g., log).
Coordinates	The space on which the data are plotted — that is, the coordinate system — usually we use a Cartesian coordinate system, but others are possible (e.g., pie charts are an example of polar coordinates). We can also use this element to flip the x and y coordinates.
Themes	All non-data elements—e.g., axis labels, background color.

The Grammar of Graphics with ggplot2 — an example of layers

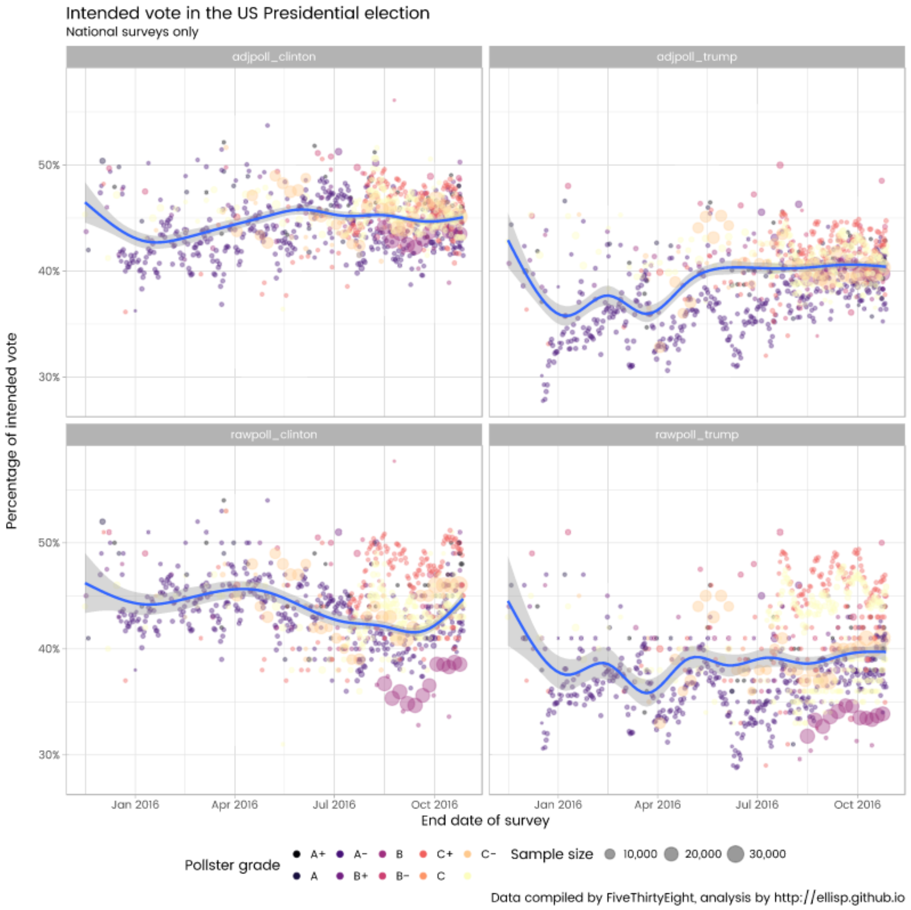


This graph presents polling data from over 3000 voting-intention polls collated by FiveThirtyEight during the 2016 presidential election. These polls were used to make predictions about the results of the USA presidential election. Data are included for Hillary Clinton and Donald Trump — and both raw (unadjusted) and adjusted values (percentages) are considered. Adjusted values adjust for the historical statistical bias of the individual polls. If the pollster consistently overestimated party X (e.g., favored Democrats) in the past, the reported percentage for party X was adjusted downwards in the adjusted values.

The Grammar of Graphics with ggplot2 — an example of layers

Facets: Two grouping variables are used here — Clinton vs Trump — and whether the polls are adjusted or unadjusted.

Aesthetics: Month of survey is mapped to the x-axis, percentage of vote is mapped to the y-axis, pollster grade (color) and poll size (size) are mapped to data points, the size is also mapped to the smooth (the loess smooth line that runs through the data points) to upweight larger polls.



Data compiled by FiveThirtyEight, analysis by http://ellisp.github.io

Themes: Titles, subtitle, labels for axes, etc. are included to aid interpretation and quick digestion of the material.

Coordinates: This graph uses Cartesian coordinates (i.e., it has an x and y axis) — the default in ggplot.

Geometry: The *points* to represent polls, the *loess smooth* to capture the time trend.

Data: This graph uses a dataframe compiled by FiveThirtyEight.

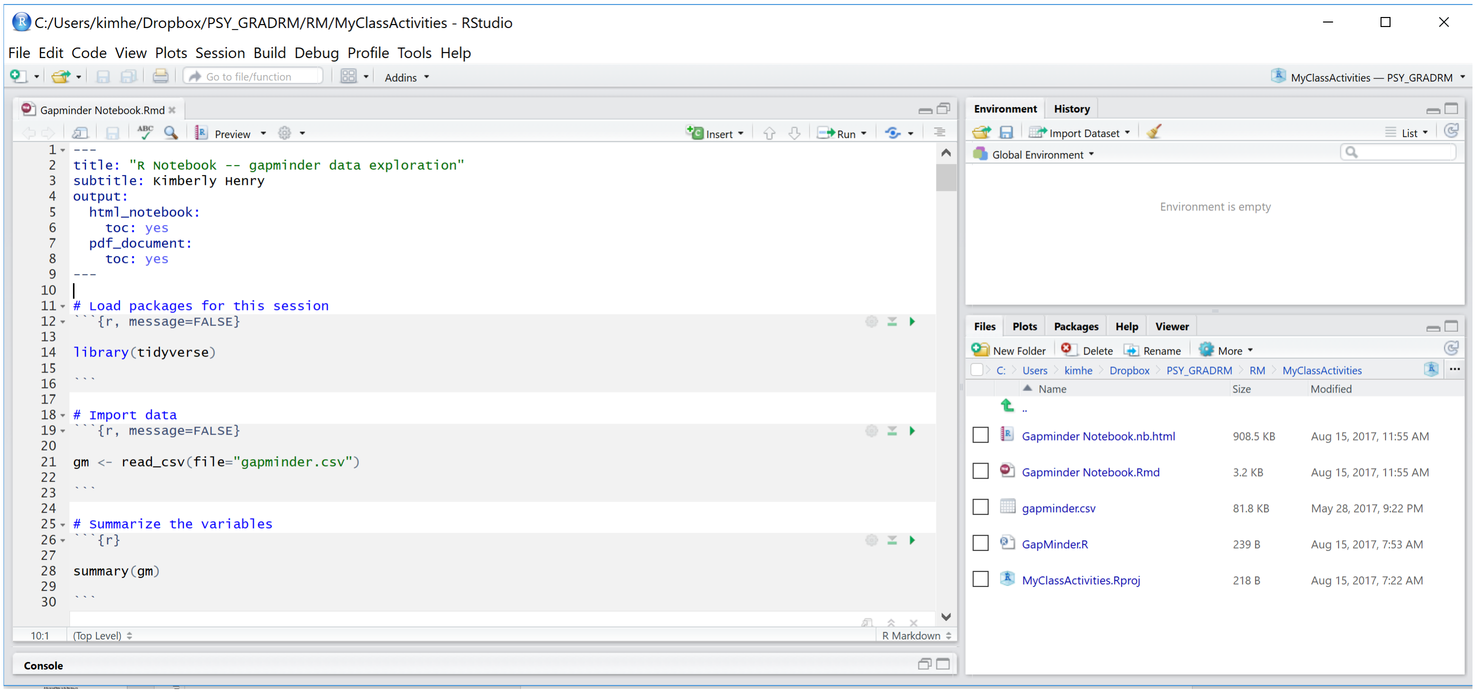
Exploration of the Grammar of Graphics with Gapminder Data

The gapminder dataset is provided by an organization called Gapminder.org. For each of 142 countries, the package provides values for life expectancy, GDP per capita, and population every five years, from 1952 to 2007.

The gapminder dataset has 1704 rows and 6 variables:

- **country** factor with 142 levels
- **continent** factor with 5 levels
- **year** ranges from 1952 to 2007 in increments of 5 years
- **lifeExp** life expectancy at birth, in years
- **pop** population
- **gdpPercap** GDP per capita

To begin, open up your Gapminder notebook: Open RStudio, click FILE > OPEN PROJECT, navigate to the MyClassActivities project that you created for Unit 1 in your RM folder. Once the project is open, then open the Gapminder Notebook (you will see it listed under the FILES tab in the lower right quadrant of RStudio, or you can use FILE > OPEN FILE.


C:/Users/kimhe/Dropbox/PSY_GRADRM/RM/MyClassActivities - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
MyClassActivities — PSY_GRADRM
Gapminder Notebook.Rmd
Preview
Insert
Run
Environment
History
Import Dataset
List
Global Environment
Environment is empty
Files
Plots
Packages
Help
Viewer
New Folder
Delete
Rename
More
C: > Users > kimhe > Dropbox > PSY_GRADRM > RM > MyClassActivities
10:1
(Top Level)
R Markdown
Console

~~~
---
title: "R Notebook -- gapminder data exploration"
subtitle: Kimberly Henry
output:
  html_notebook:
    toc: yes
  pdf_document:
    toc: yes
---

# Load packages for this session
```{r, message=FALSE}

library(tidyverse)

```

# Import data
```{r, message=FALSE}

gm <- read_csv(file="gapminder.csv")

```

# Summarize the variables
```{r}

summary(gm)

```
~~~

Name	Size	Modified
..		
□ Gapminder Notebook.nb.html	908.5 KB	Aug 15, 2017, 11:55 AM
□ Gapminder Notebook.Rmd	3.2 KB	Aug 15, 2017, 11:55 AM
□ gapminder.csv	81.8 KB	May 28, 2017, 9:22 PM
□ GapMinder.R	239 B	Aug 15, 2017, 7:53 AM
□ MyClassActivities.Rproj	218 B	Aug 15, 2017, 7:22 AM

The ggplot Function.

Let's begin with a very basic graph, it's actually the graph we created in Unit 1. It is a density plot of one of the variables in the dataframe — life expectancy (lifeExp). A density plot displays the distribution of data over a continuous interval, the density function describes the relative likelihood of a variable to take on a given value. The peaks of a density plot help display where values are concentrated over the interval.

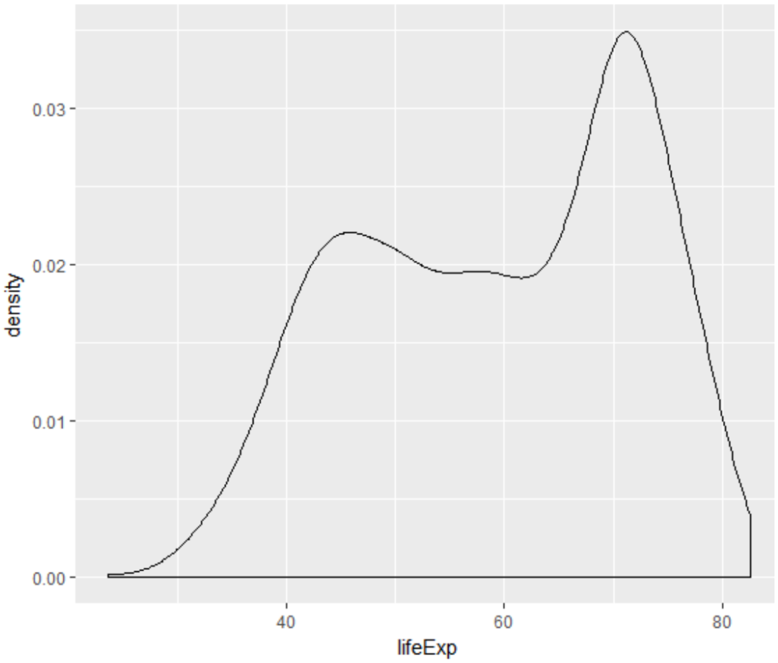
Aesthetics are the visual properties of the objects in your plot. For this simple density plot, we simply have a variable on the x axis — that will be life expectancy. The y-axis of a density plot is the density.

As the first argument to the function, we list the dataframe to plot.

ggplot is the function name

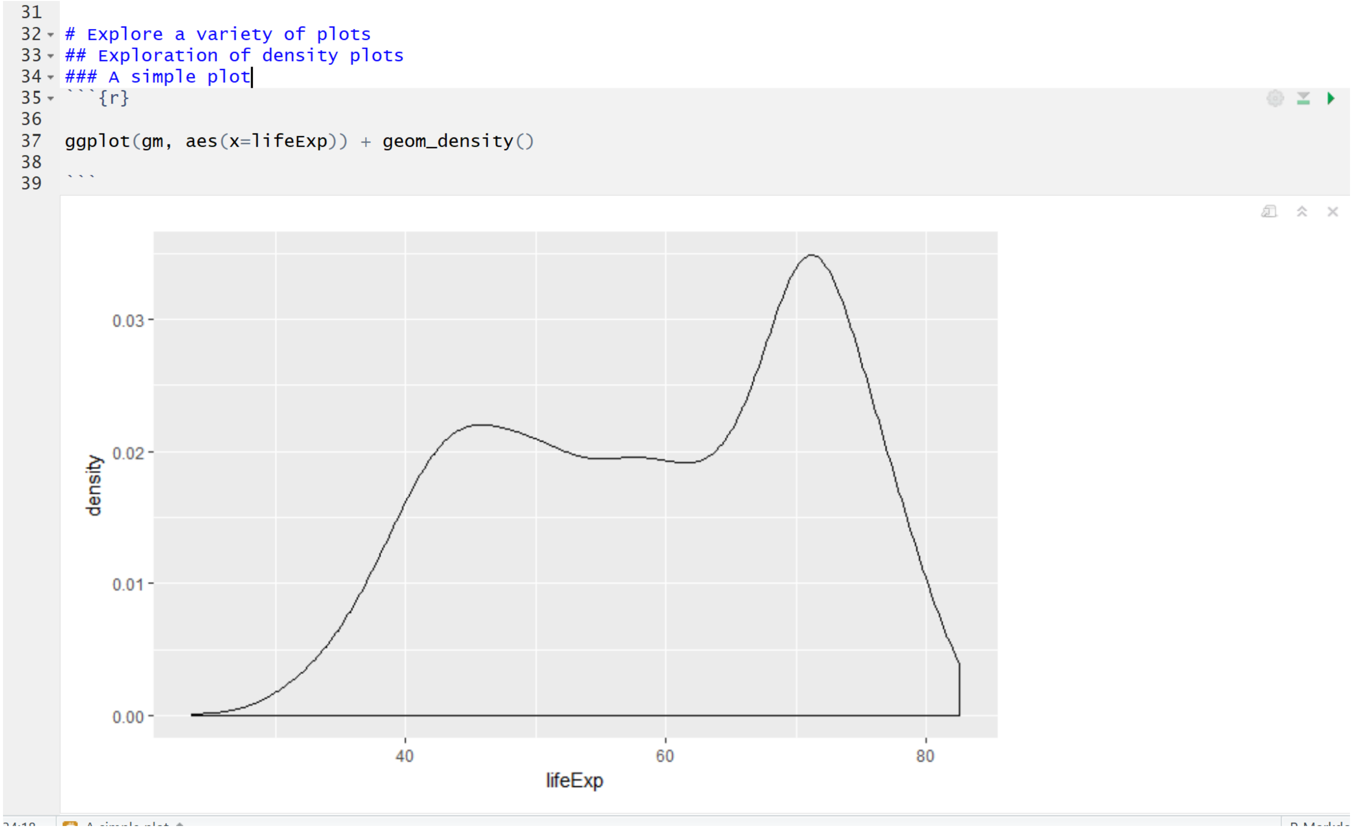
```
ggplot(gm, aes(x = lifeExp)) + geom_density()
```

As the second layer to the function, we specify the geometry of the plot. Common types include density, histogram, bar, point (for scatterplots), line. Each geom function can have aesthetics as well. Here we start simple and do not request any additional aesthetics.



Plot Exploration

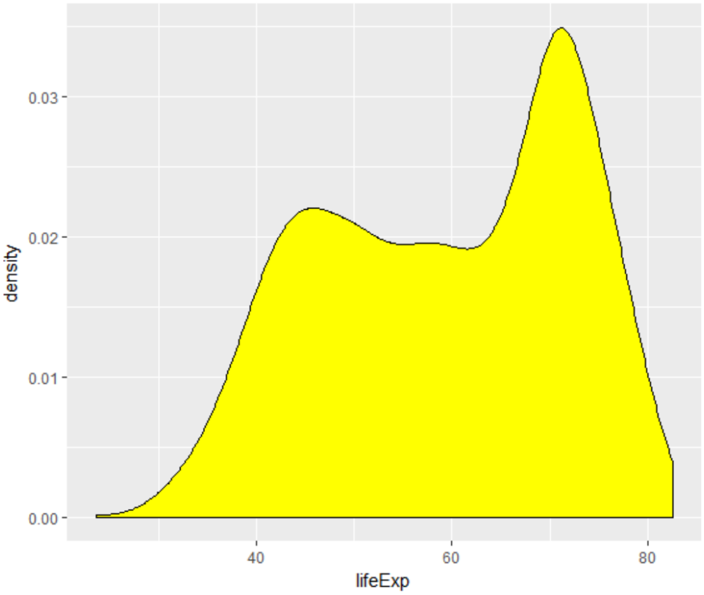
Let's start adding to our Gapminder Notebook. We're going to estimate a series of different plots. In order to keep them nicely organized in our notebook, we'll use a few headings. Enhance your notebook by adding the three heading lines below. Recall that the hashtags (outside of a code chunk) refer to a first level (#), a second level (##), and a third level (###) heading.



~~~

Explore a variety of plots
Exploration of density plots
A simple plot
```{r}

ggplot(gm, aes(x=lifeExp)) + geom_density()

```
~~~

Enhancements to the Density Plot

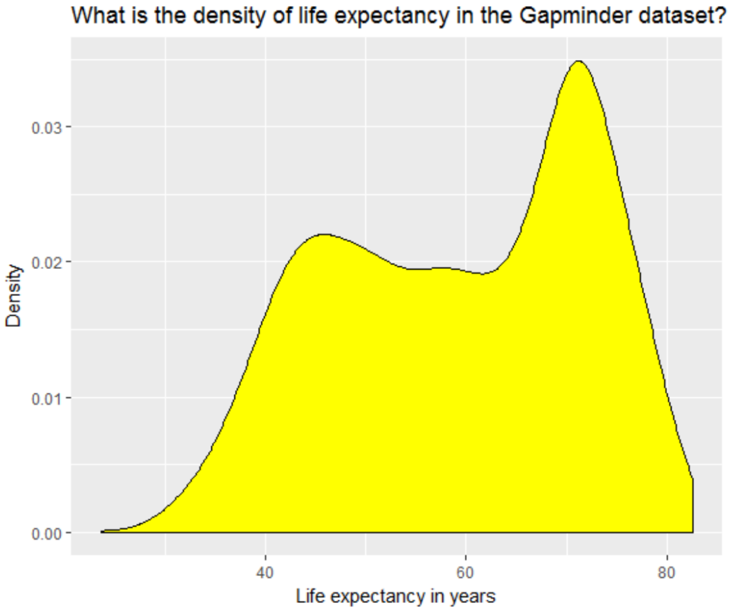
```
ggplot(gm, aes(x = lifeExp)) +
  geom_density(fill = "yellow")
```



Let's change the color of the density plot. See the Rcolors.pdf in the Dropbox Other Resources folder for a list of colors and color names readily available.

When using ggplot, if you go to a new line, do it AFTER a + sign. With ggplot or any R function, you can go to the next line after a comma.

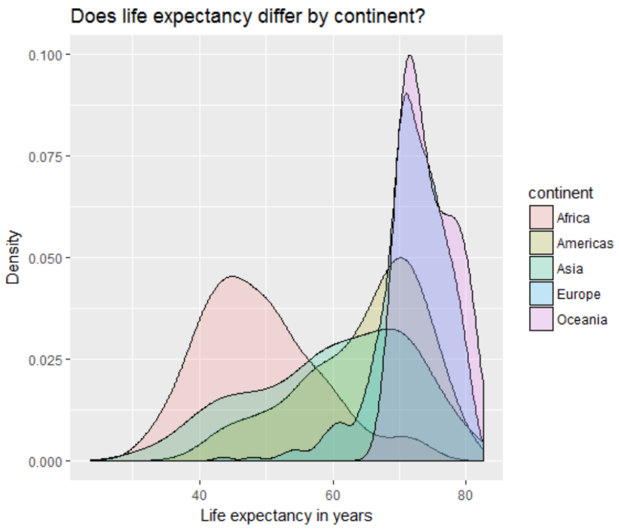
```
ggplot(gm, aes(x = lifeExp)) +
  geom_density(fill = "yellow") +
  labs(title = "What is the density of life expectancy in the Gapminder dataset?",
       x = "Life expectancy in years", y = "Density")
```



Now, we enhance the graph to include a title and axis labels.

Enhancements to the Density Plot

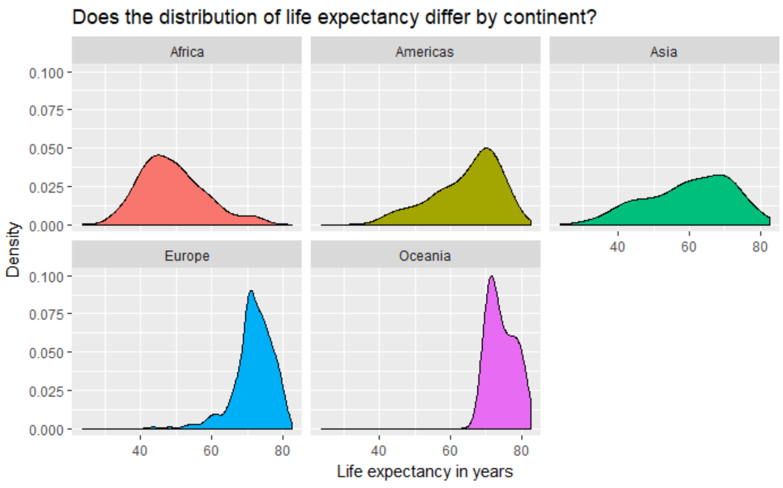
```
ggplot(gm, aes(x = lifeExp, group = continent, fill=continent)) +
  geom_density(alpha=.2) +
  labs(title = "Does the distribution of life expectancy differ by continent?",
     x = "Life expectancy in years", y = "Density")
```



In the first density plot, we lumped together all continents. We can separate them if we like. By specifying group = continent, we can create a separate density plot for each continent. The fill = continent indicates we want to fill the density plots with a different color for each continent.

The alpha argument controls the transparency of the element. This is helpful here because it's easier to see each density.

```
ggplot(gm, aes(x = lifeExp, group =continent, fill = continent)) +
  geom_density(show.legend = FALSE) +
  facet_wrap(~ continent) +
  labs(title = "Does the distribution of life expectancy differ by continent?",
     x = "Life expectancy in years", y = "Density")
```



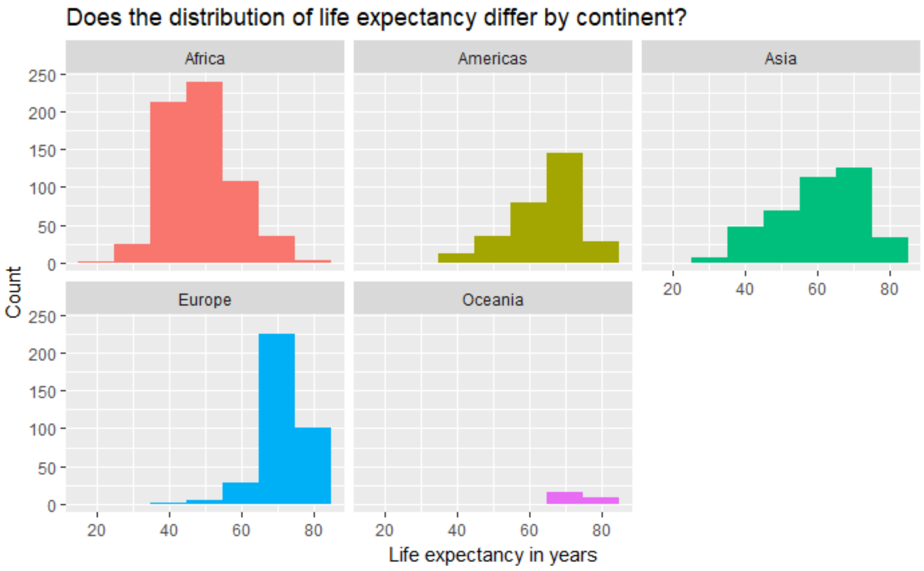
Instead of putting all of the density plots on one plot, let's use facet_wrap to have a separate plot tile for each continent. In the fact_wrap argument, you can list a tilde (~) and then the variable that denotes the groupings (continent in this case). Another argument, facet_grid, works in a similar way but allows for two grouping variables. We'll see examples of this later in the semester.

Notice that I use show.legend = FALSE to suppress the legend, which would be redundant in this case.

Let’s Look at a Similar Plot Type– Histograms

A histogram consists of rectangles in which the area is proportional to the frequency of the specified variable and the width is equal to the class interval. The binwidth denotes the desired bins or chunks for the variable — here we use 10, which will chunk in 10 years increments(i.e, how many people are expected to have a life expectancy in each 10 year chunk — e.g., 20’s, 60’s, 80’s etc.).

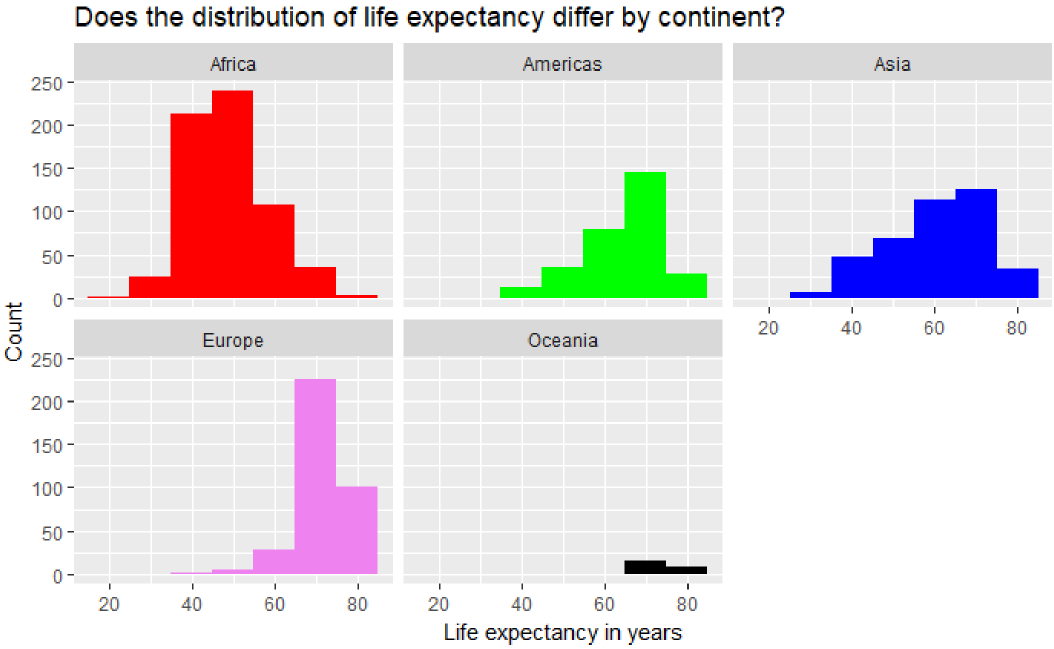
```
ggplot(gm, aes(x = lifeExp, group = continent, fill = continent)) +
  geom_histogram(show.legend = FALSE, binwidth=10) +
  facet_wrap(~ continent) +
  labs(title = "Does the distribution of life expectancy differ by continent?",
     x = "Life expectancy in years", y = "Count")
```



Enhancements to the Density Plot

We can pick our own fill colors with the scale_fill_manual option.

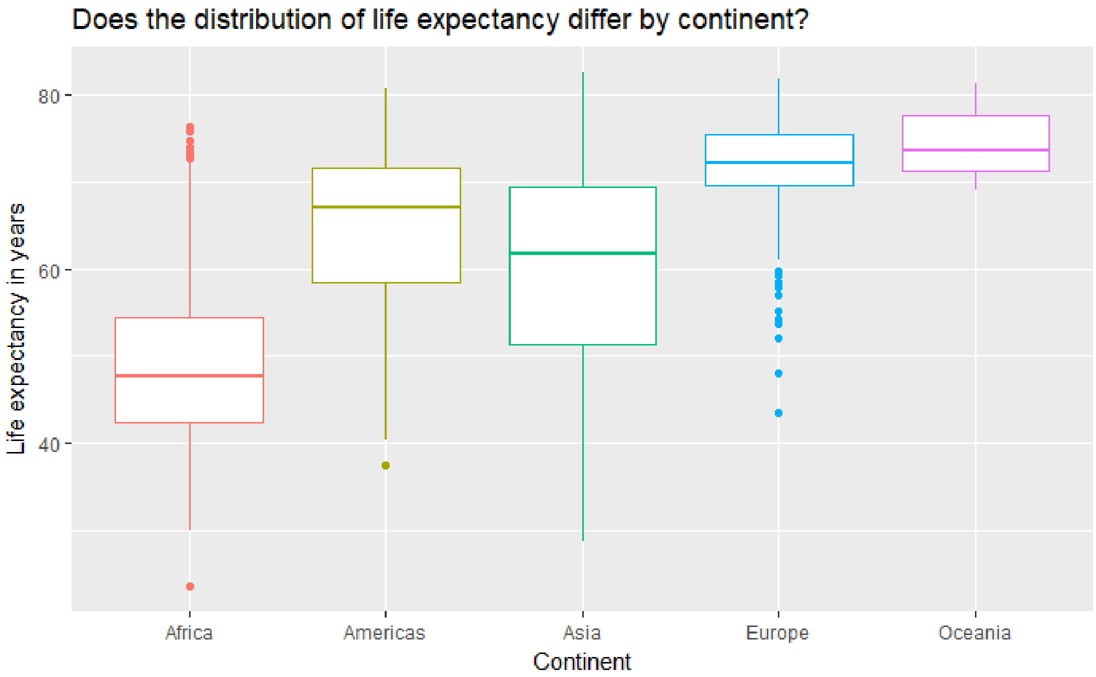
```
ggplot(gm, aes(x = lifeExp, group = continent, fill= continent)) +
  geom_histogram(show.legend = FALSE, binwidth=10) +
  facet_wrap(~ continent) +
  labs(title = "Does the distribution of life expectancy differ by continent?",
     x = "Life expectancy in years",  y = "Count") +
  scale_fill_manual(values = c("red","green","blue","violet","black"))
```



Now, Let's Consider Boxplots

A boxplot displays the distribution of data based on the five number summary: minimum, first quartile, median, third quartile, and maximum. The "interquartile range", abbreviated "IQR", is the length of the box, or IQR = Q3 – Q1 . An outlier is a value that lies more than one and a half times the length of the box (either below or above).

```
ggplot(gm, aes(x = continent, y = lifeExp, group =continent, color = continent)) +
  geom_boxplot(show.legend = FALSE) +
  labs(title = "Does the distribution of life expectancy differ by continent?",
       x = "Continent", y = "Life expectancy in years")
```

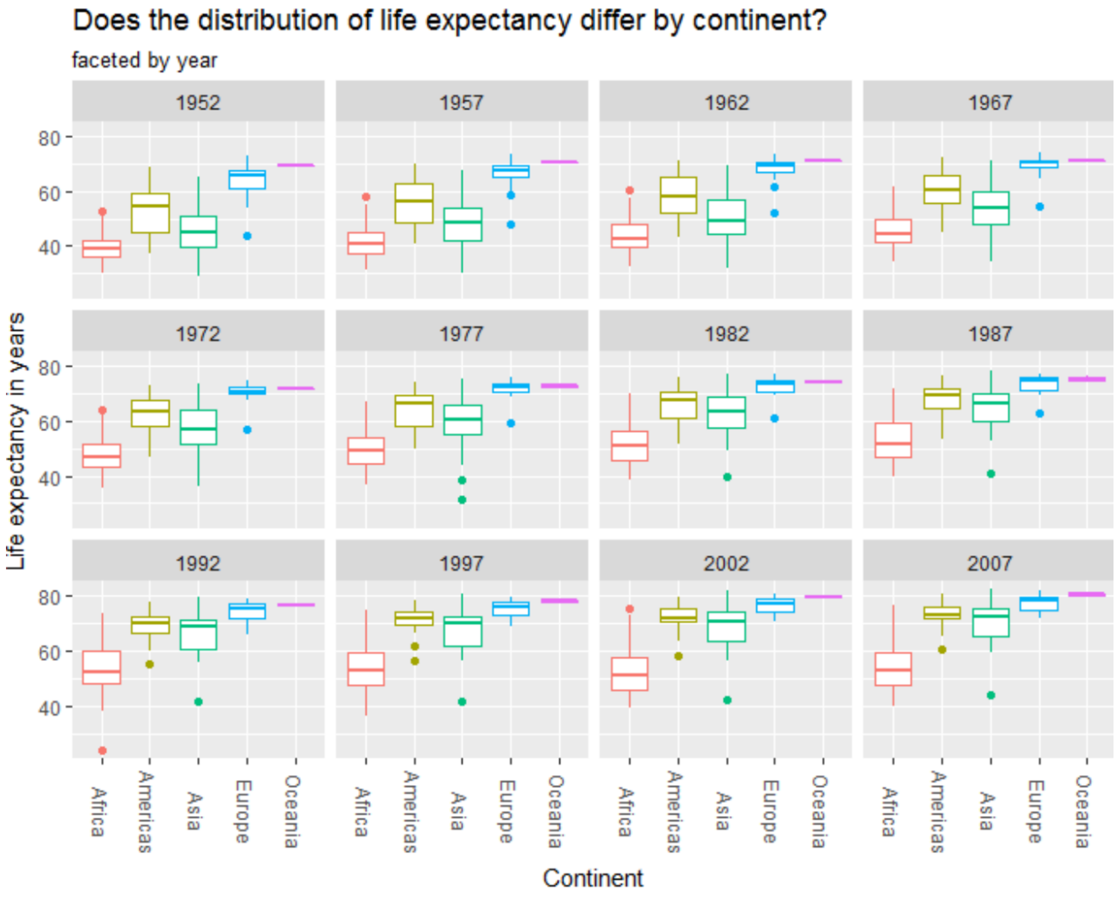


Notice that in this code chunk, rather than using fill to set the color, I use the argument "color" to set the color — this sets the border of the element rather than the fill color.

Enhancements to Boxplots

So far in our explorations, we have ignored the fact that there are multiple years worth of data. Now, let's facet our boxplot by year. Here, I use the facet_wrap argument to provide a boxplot for each year of available data. Notice also that I am using the theme function to rotate the x-axis text. Now the continent names are rotated and easier to read. Vjust moves the title between left justified (0) and right justified (1) — I pick .5 so that it lines up in the middle.

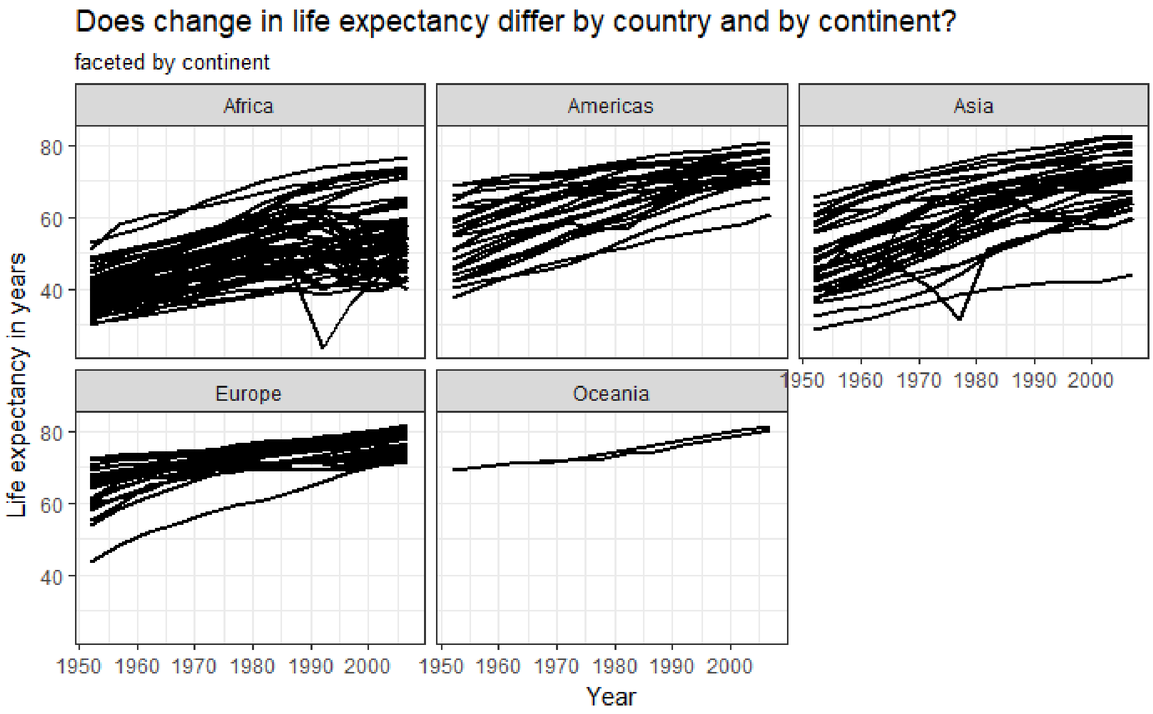
```
ggplot(gm, aes(x = continent, y = lifeExp, group =continent, color = continent)) +
  geom_boxplot(show.legend = FALSE) +
  facet_wrap(~ year) +
  labs(title = "Does the distribution of life expectancy differ by continent?",
      subtitle = "faceted by year", x = "Continent", y = "Life expectancy in years") +
  theme(axis.text.x = element_text(angle=-90, vjust=.5))
```



Let’s Consider Line Plots

Line plots can be useful for examining change over some continuous x variable. Let’s examine how life expectancy changes across years for each country within each continent. The group aesthetic requests a separate line for each country. The argument lwd sets the line width.

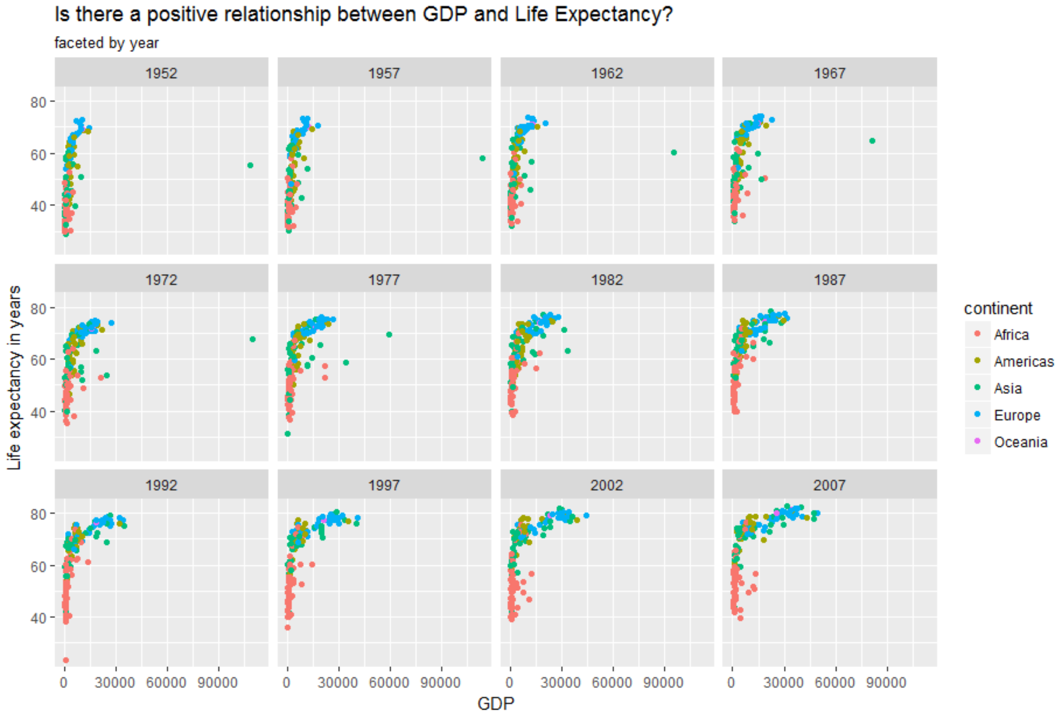
```
ggplot(gm, aes(x = year, y = lifeExp, group = country)) +
  geom_line(lwd = 1, show.legend = FALSE) +
  facet_wrap(~ continent) +
  labs(title = "Does change in life expectancy differ by country and by continent?",
      subtitle = "faceted by continent", x = "Year", y = "Life expectancy in years")
```



Let’s Consider Scatterplots

Scatterplots are a useful graph for examining the relationship between two continuous variables. Each point on the graph represents a case — and each case’s score on x and y are plotted on the graph. Let’s make a scatterplot of life expectancy and GDP. We’ll color the points by continent, and request a separate time by year (facet_wrap).

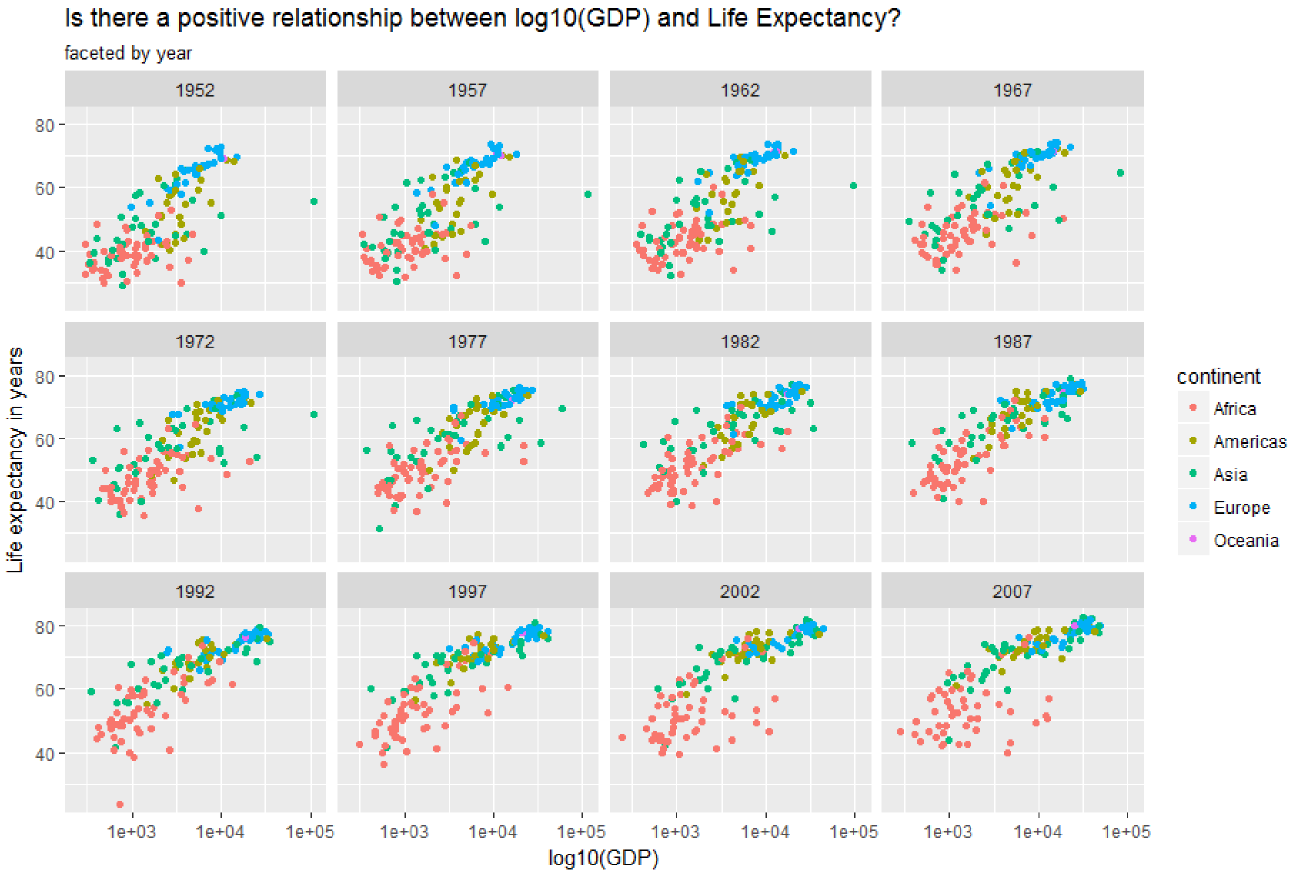
```
ggplot(gm, aes(x = gdpPercap, y = lifeExp, group = continent, color = continent)) +
  geom_point() +
  facet_wrap(~ year) +
  labs(title = "Is there a positive relationship between GDP and Life Expectancy?",
     subtitle = "faceted by year", x = "GDP", y = "Life expectancy in years")
```



Enhancements to Scatterplots

In the previous plot, we see that our measure of GDP is highly skewed; it has a very long right hand tail. We can apply a log transformation to pull in the tail (don't worry if this idea of non-linear transformations is unfamiliar to you — we will spend time learning and exploring these types of concepts later this semester). We can use the scale_x_log10 function to apply a log base 10 transformation to gdpPercap.

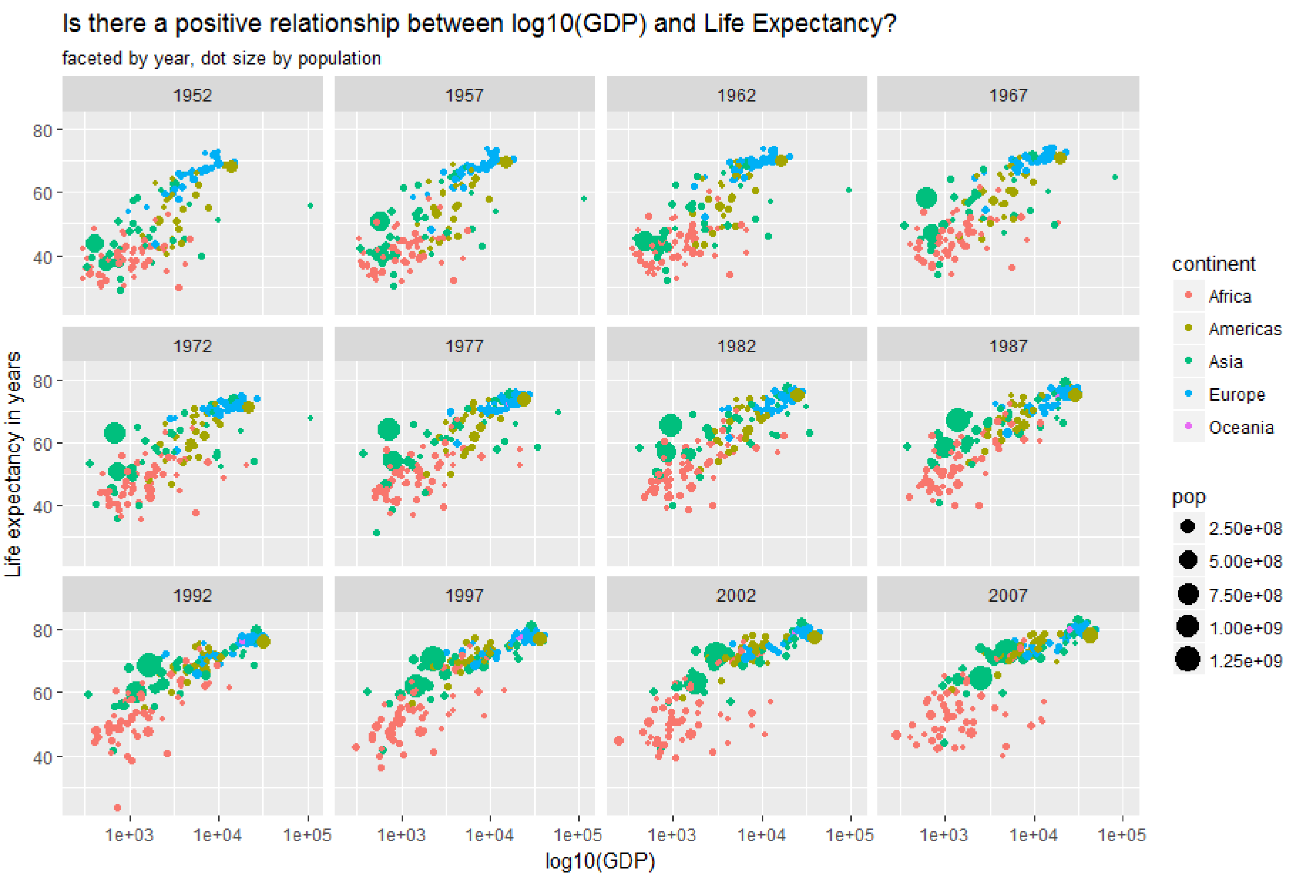
```
ggplot(gm, aes(x = gdpPercap, y = lifeExp, group = continent, color = continent)) +
  geom_point() +
  facet_wrap(~ year) +
  scale_x_log10() +
  labs(title = "Is there a positive relationship between log10(GDP) and Life Expectancy?",
       subtitle = "faceted by year", x = "log10(GDP)", y = "Life expectancy in years")
```



Enhancements to Scatterplots

Let's enhance our scatterplot by requesting that the datapoints are sized based on population, see the size = pop statement in the geom_point argument, here we indicate that we want the size of the points to depend on the variable pop (larger populations constitute larger points).

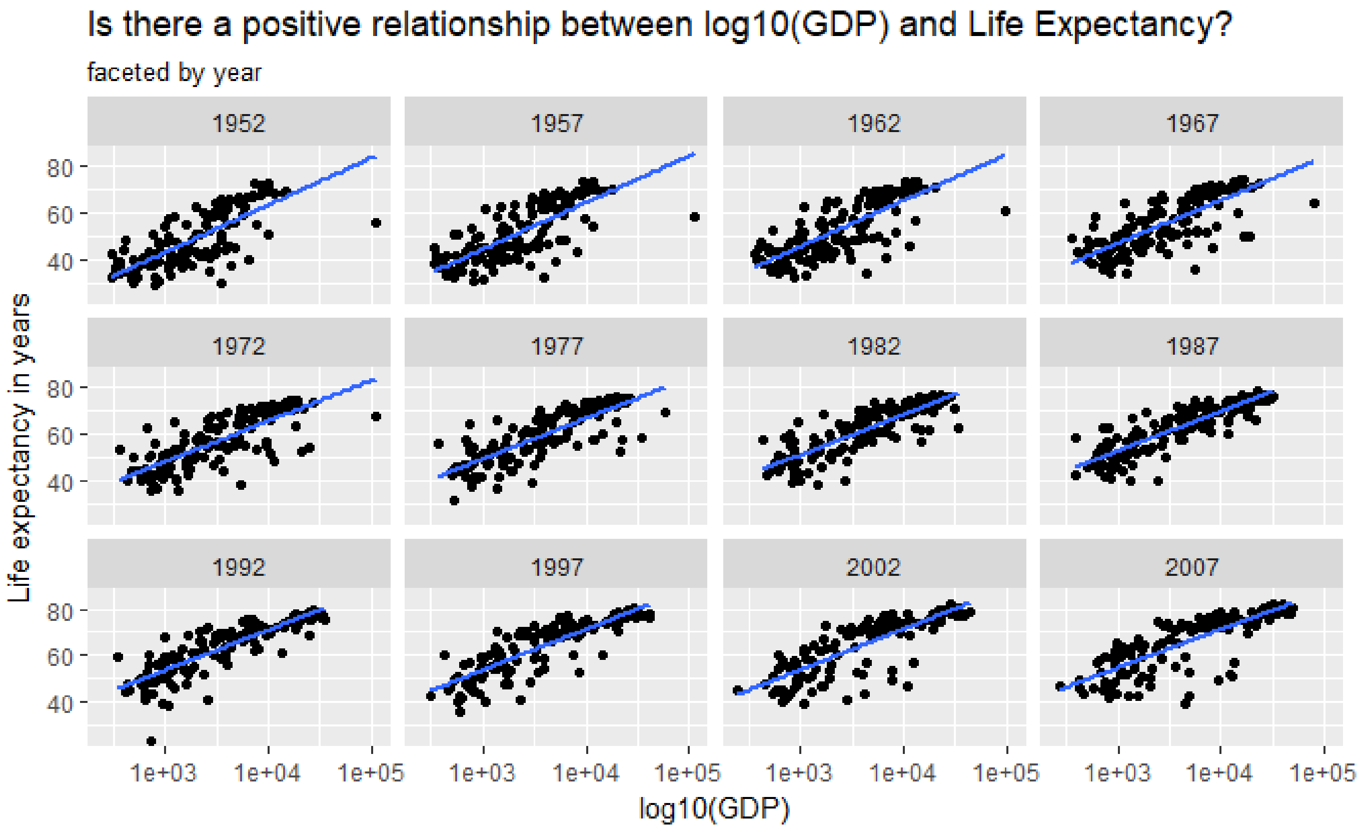
```
ggplot(gm, aes(gdpPercap, lifeExp, group = continent, color = continent)) +
  geom_point(aes(size = pop)) +
  facet_wrap(~ year) +
  scale_x_log10() +
  labs(title = "Is there a positive relationship between log10(GDP) and Life Expectancy?",
     subtitle = "faceted by year, dot size by population", x = "log10(GDP)", y = "Life expectancy in years")
```



Enhancements to Scatterplots

Let's implement our last enhancement, and add a best fit line through the cloud of datapoints. We do this with geom_smooth argument. Here we ask for a linear model (lm — a straight line) and se=FALSE to indicate that we don't want standard errors.

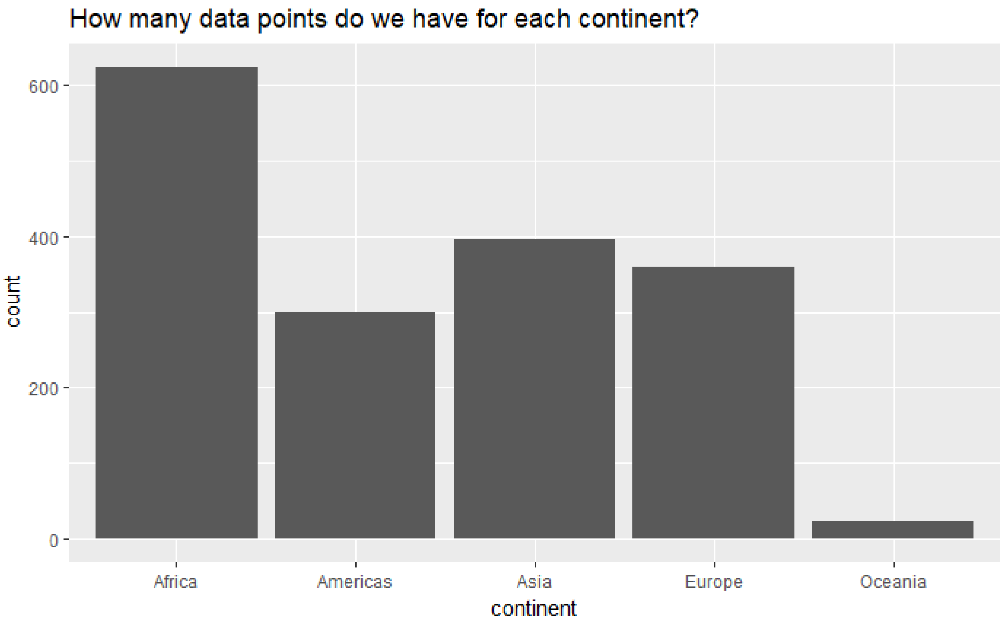
```
ggplot(gm, aes(x = gdpPercap, y = lifeExp)) +
  geom_point() +
  geom_smooth(method="lm", se=FALSE) +
  facet_wrap(~ year) +
scale_x_log10() +
  labs(title = "Is there a positive relationship between log10(GDP) and Life Expectancy?",
       subtitle = "faceted by year", x = "log10(GDP)", y = "Life expectancy in years")
```



And Finally, Let's Take a Look at a Bar Chart

A bar chart is a graph that presents grouped data with rectangular bars. The lengths of the bar charts are proportional to the values that they represent (for example, the count of cases).

```
ggplot(gm, aes(x = continent)) + geom_bar() +
  labs(title = "How many data points do we have for each continent?")
```


How many data points do we have for each continent?
count
Africa
Americas
Asia
Europe
Oceania
continent