cm006 Exercises: Exploring Geometric Objects

Plotting in R

- base R (came with R)
- lattice
- ggplot2 (an R package)
  - Part of the tidyverse

When using ggplot2 for generating figure, one most specific the following things:

- Data
- Aesthetic mapping (horizontal axis, vertical axis, . . .)
- Geometric objects

The ggplot2 **ggplot()** qplot()

In this worksheet, we'll be exploring various plot types (i.e., geometric objects), only using the `x` and `y` aesthetics (and `group`).

1. To get started, load the `tidyverse` and `gapminder` R packages.

```
#library(tidyverse)
library(gapminder)
library(ggplot2)
```

Scatterplot

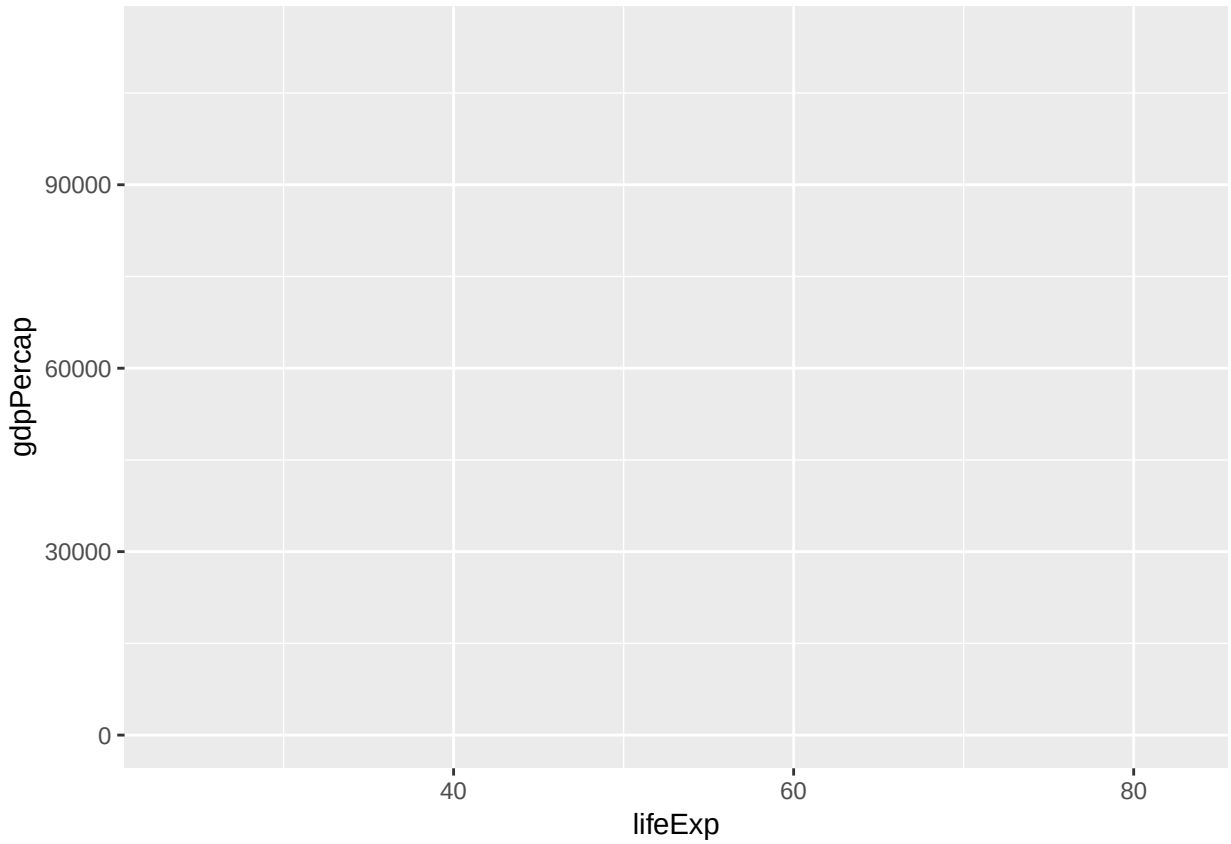
Let's look at a *scatterplot* of `gdpPercap` vs. `lifeExp`.

1. Fill out the grammar components below. Again, bold *must* be specified to make a `ggplot2` plot.
   - We'll ignore "coordinate system" and "facetting" after this.

Grammar Component	Specification
data	`gapminder`
aesthetic mapping	`x` and `y`
geometric object	point
scale	linear
statistical transform	none
coordinate system	rectangular
facetting	none

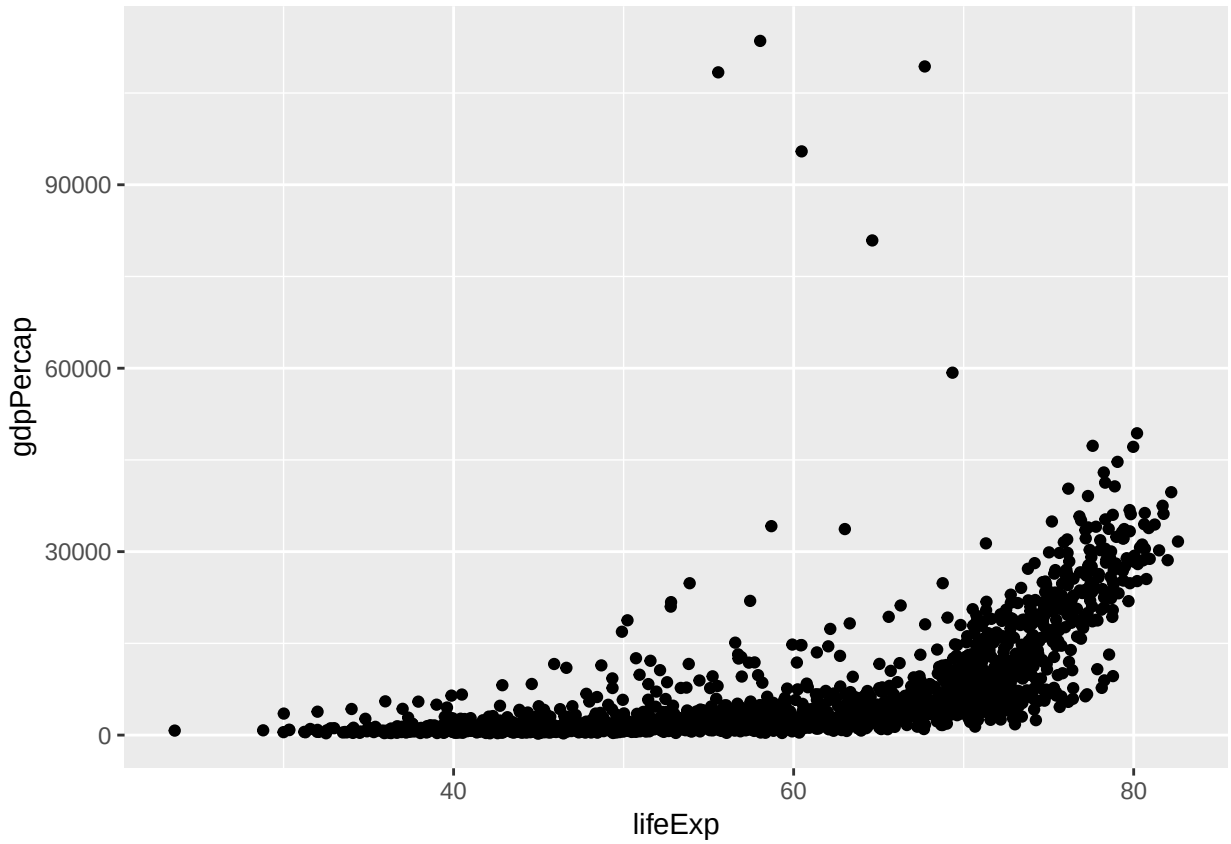
2. Populate the data and aesthetic mapping in `ggplot`. What is returned? What's missing?

```
ggplot(gapminder,aes(x=lifeExp,y=gdpPercap))
```



3. Add the missing component as a *layer*.

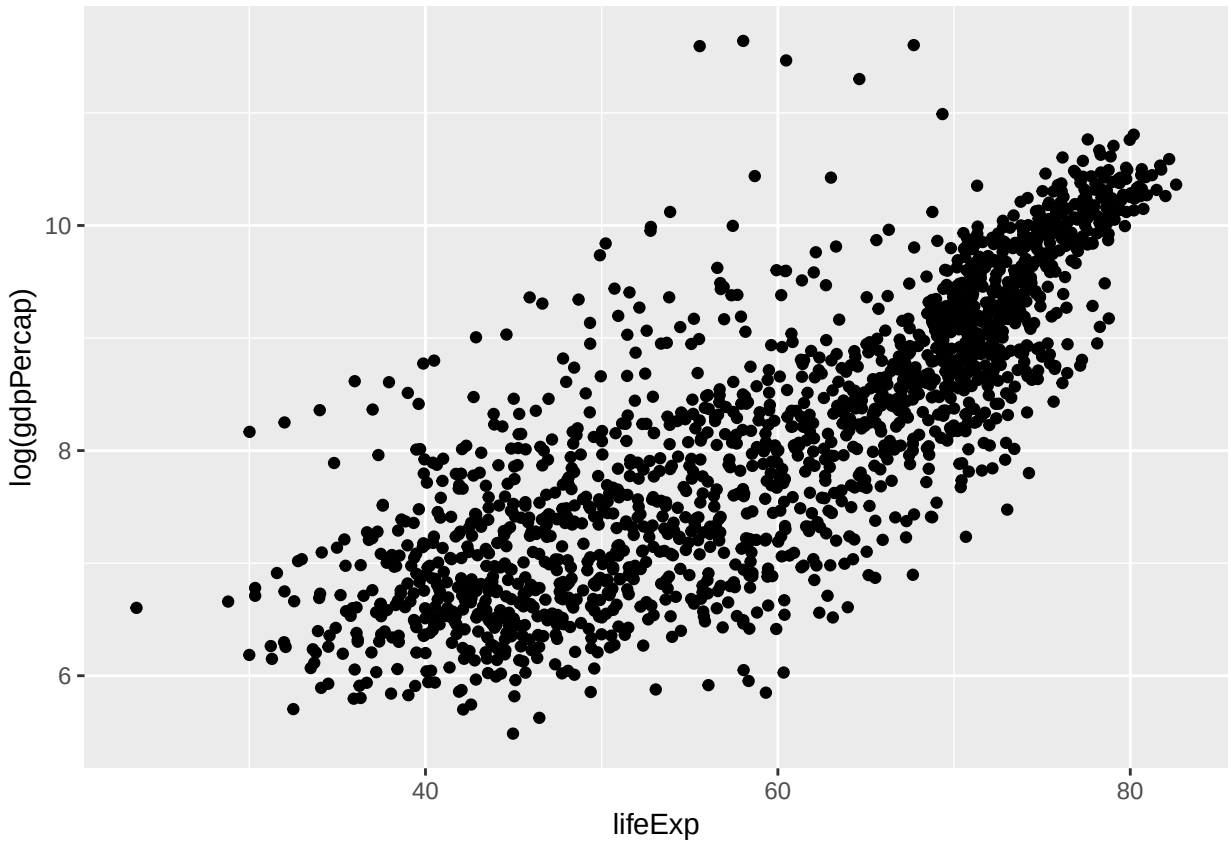
```
ggplot(gapminder,aes(x=lifeExp,y=gdpPercap))+ geom_point()
```



Notice the "metaprogramming" again!

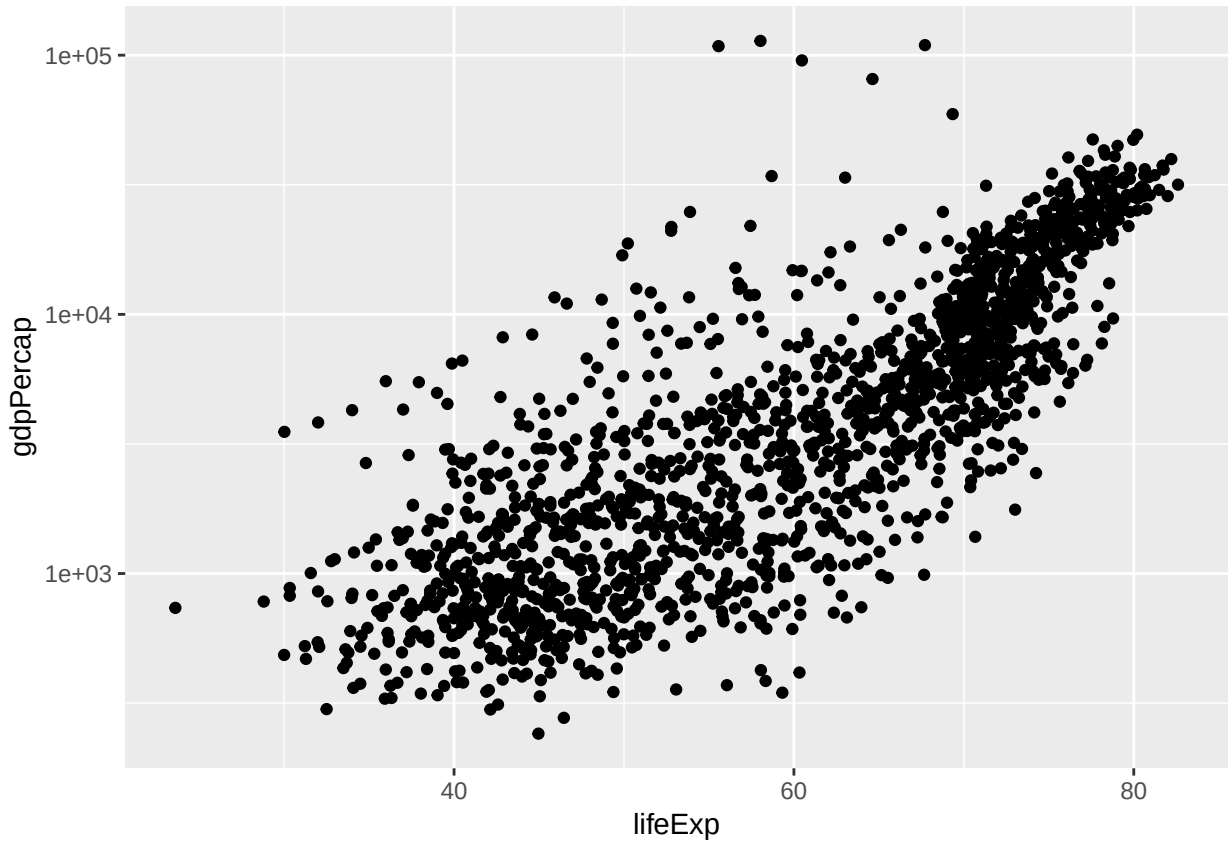
4. You *must* remember to put the aesthetic mappings in the `aes` function! What happens if you forget?
5. Put the x-axis on a log scale, first by transforming the x variable.
    - Note: `ggplot2` does some data wrangling and computations itself! We don't always have to modify the data frame.

```
ggplot(gapminder,aes(x=lifeExp,y=log(gdpPercap))) +
       geom_point()
```



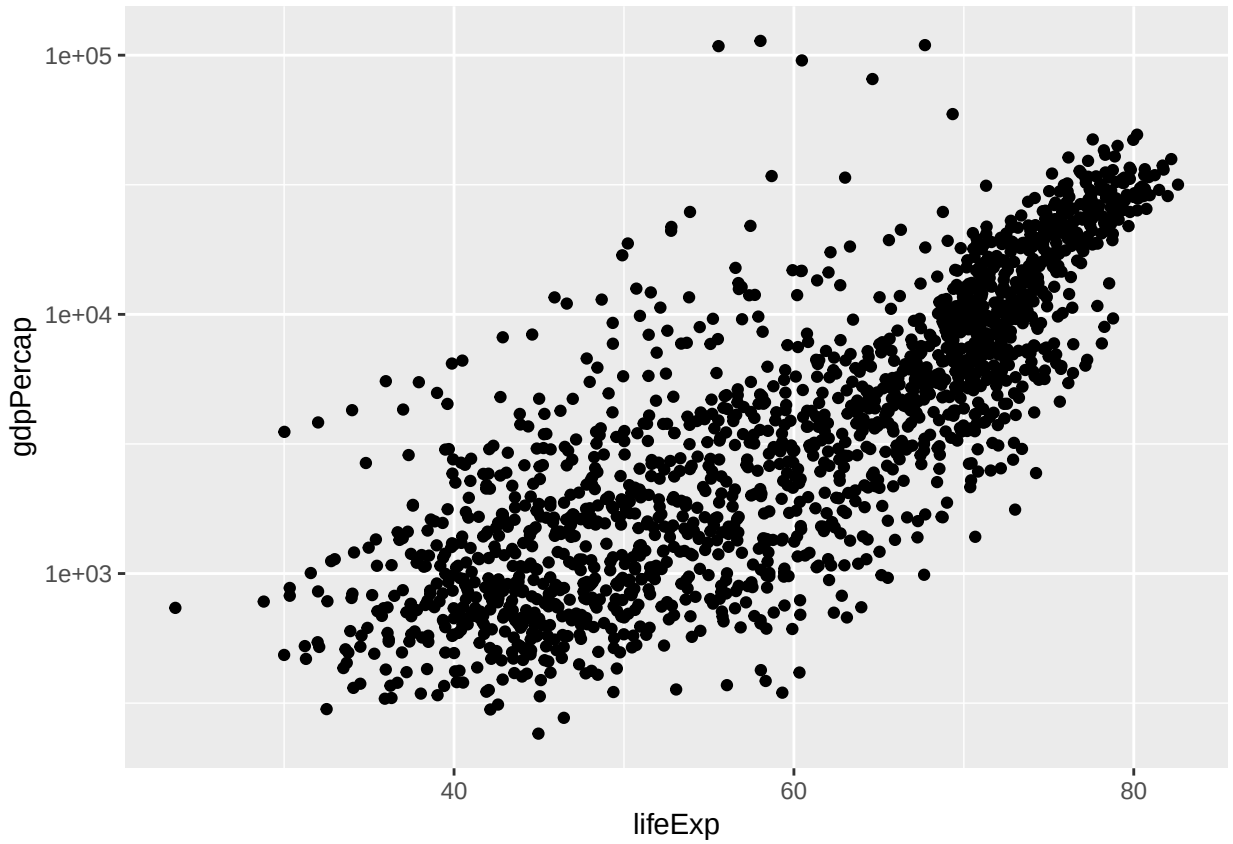
6. Try again, this time by changing the *scale* (this way is better).

```
ggplot(gapminder,aes(x=lifeExp,y=gdpPercap))+
  geom_point() +
  scale_y_log10()
```



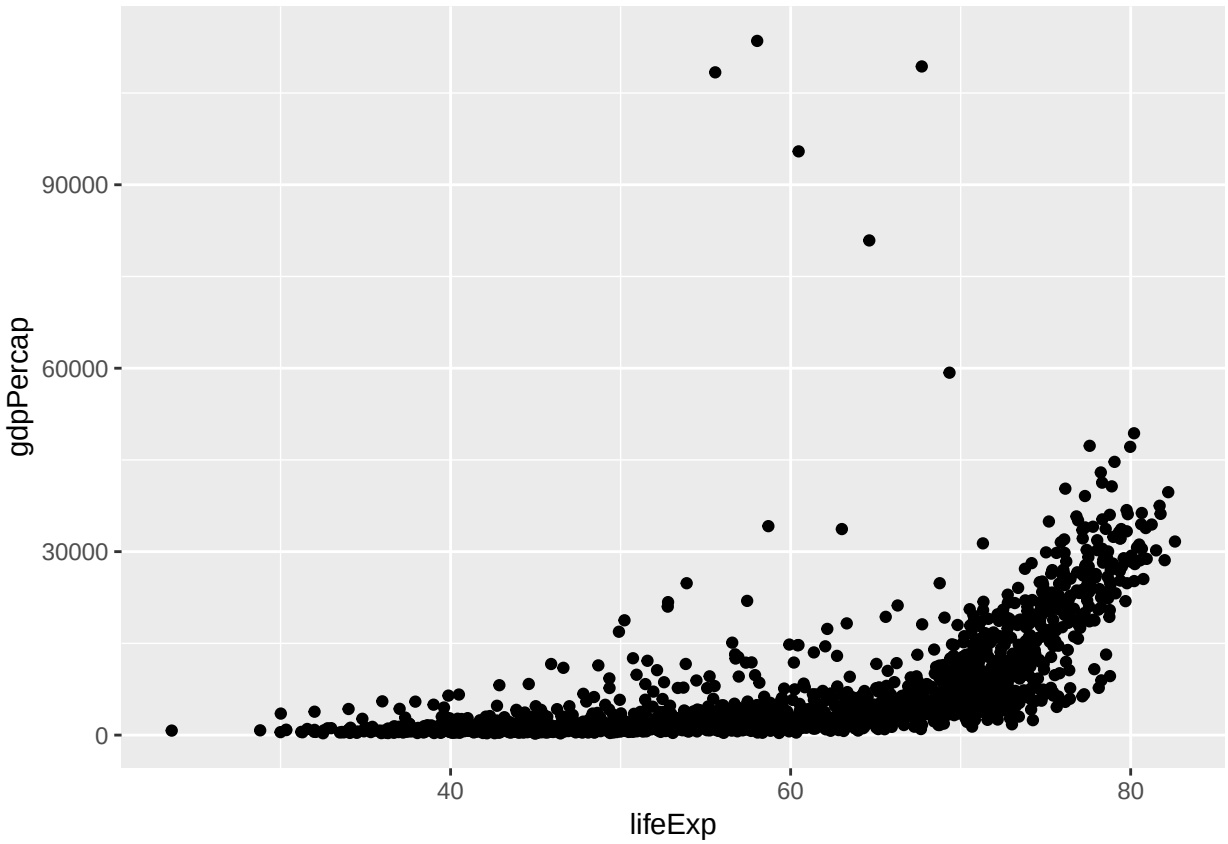
Inter-changing the order of **scale** and **geom_point**

```
ggplot(gapminder,aes(x=lifeExp,y=gdpPercap))+
  scale_y_log10() +
  geom_point()
```



7. The aesthetic mappings can be specified on the geom layer if you want, instead of the main `ggplot` call. Give it a try:

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



8. Optional: git stage and commit

Uses of a scatterplot:

- Visualize 2-dimensional distributions; dependence.
- 2 numeric variables

Histograms, and Kernel Density Plots

Let's build a histogram of life expectancy.

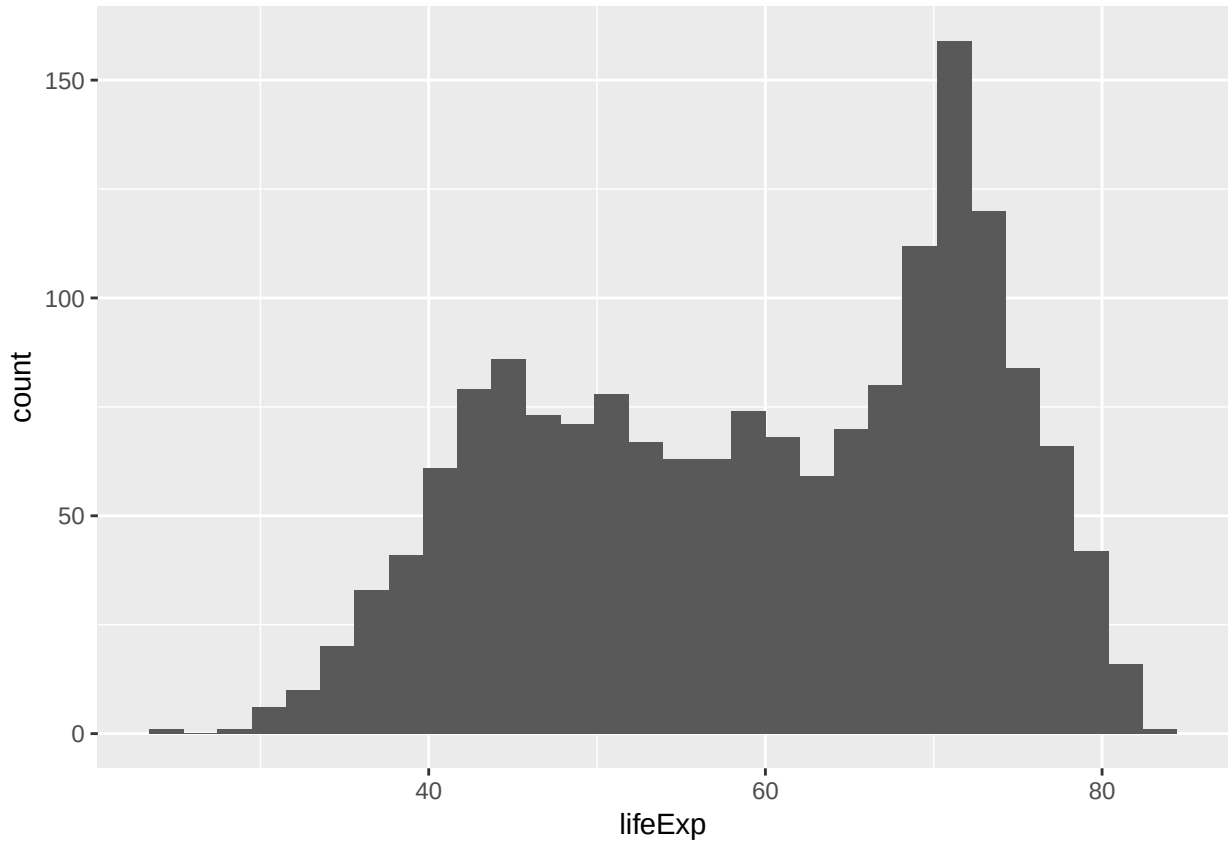
1. Fill out the grammar components below. Again, bold *must* be specified to make a `ggplot2` plot.

Grammar Component	Specification
data	`gapminder`
aesthetic mapping	`x`
geometric object	histogram
scale	linear
statistical transform	none

2. Build the histogram of life expectancy.

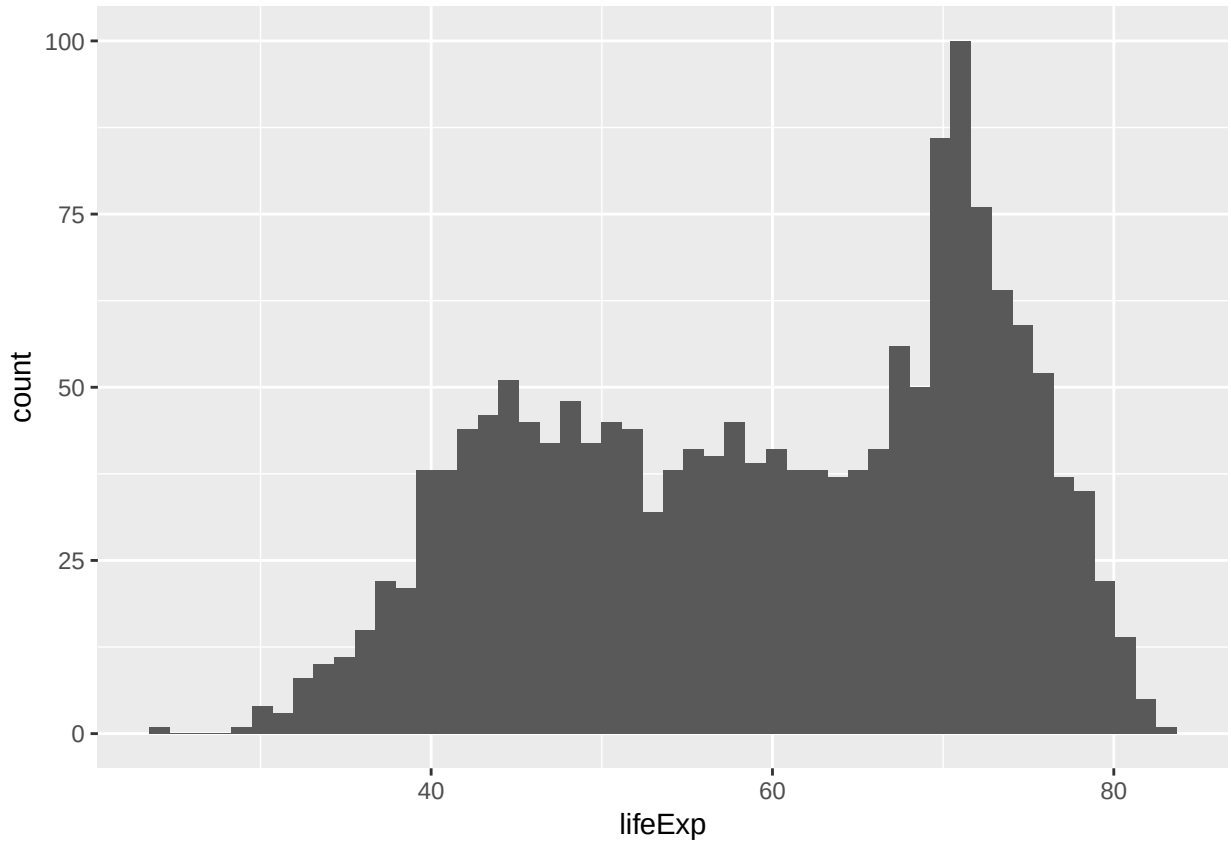
```
ggplot(gapminder, aes(lifeExp)) +
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



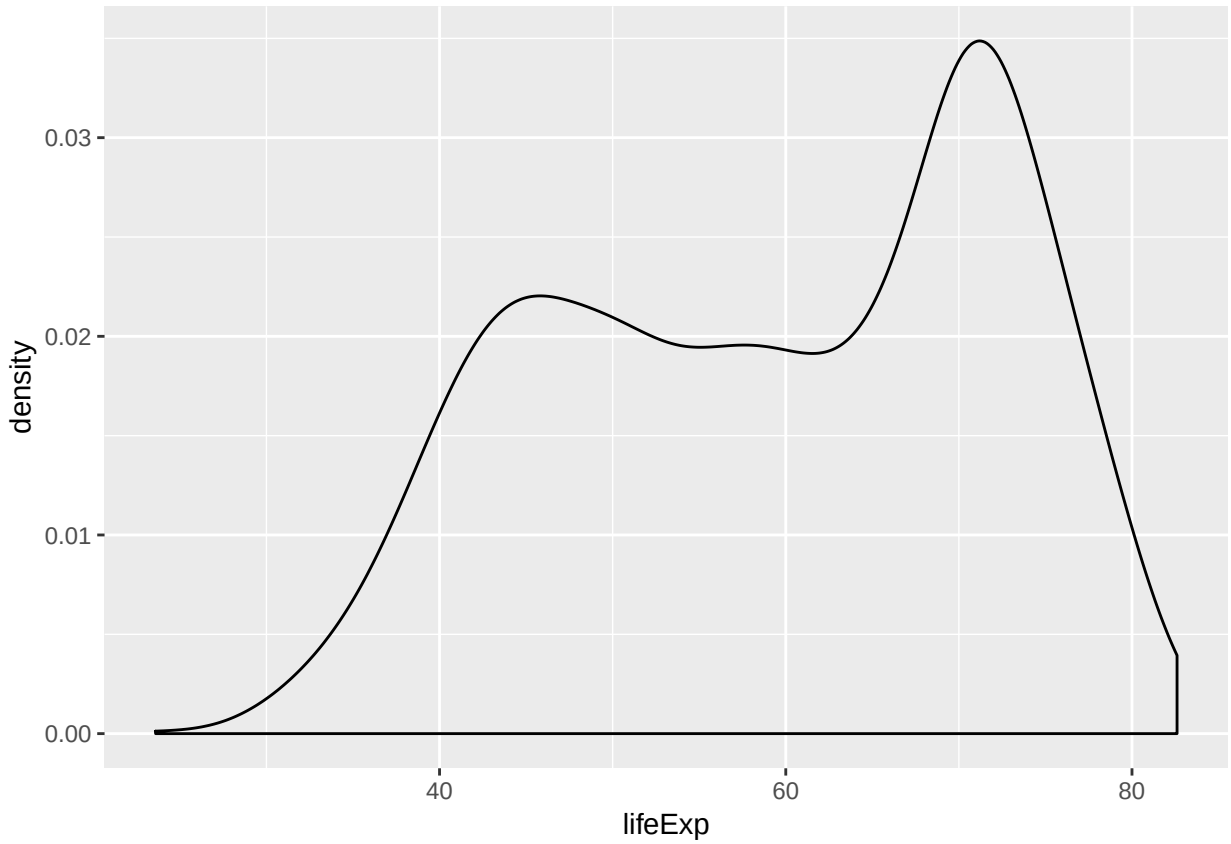
3. Change the number of bins to 50.

```
ggplot(gapminder, aes(lifeExp)) +
  geom_histogram(bins=50)
```

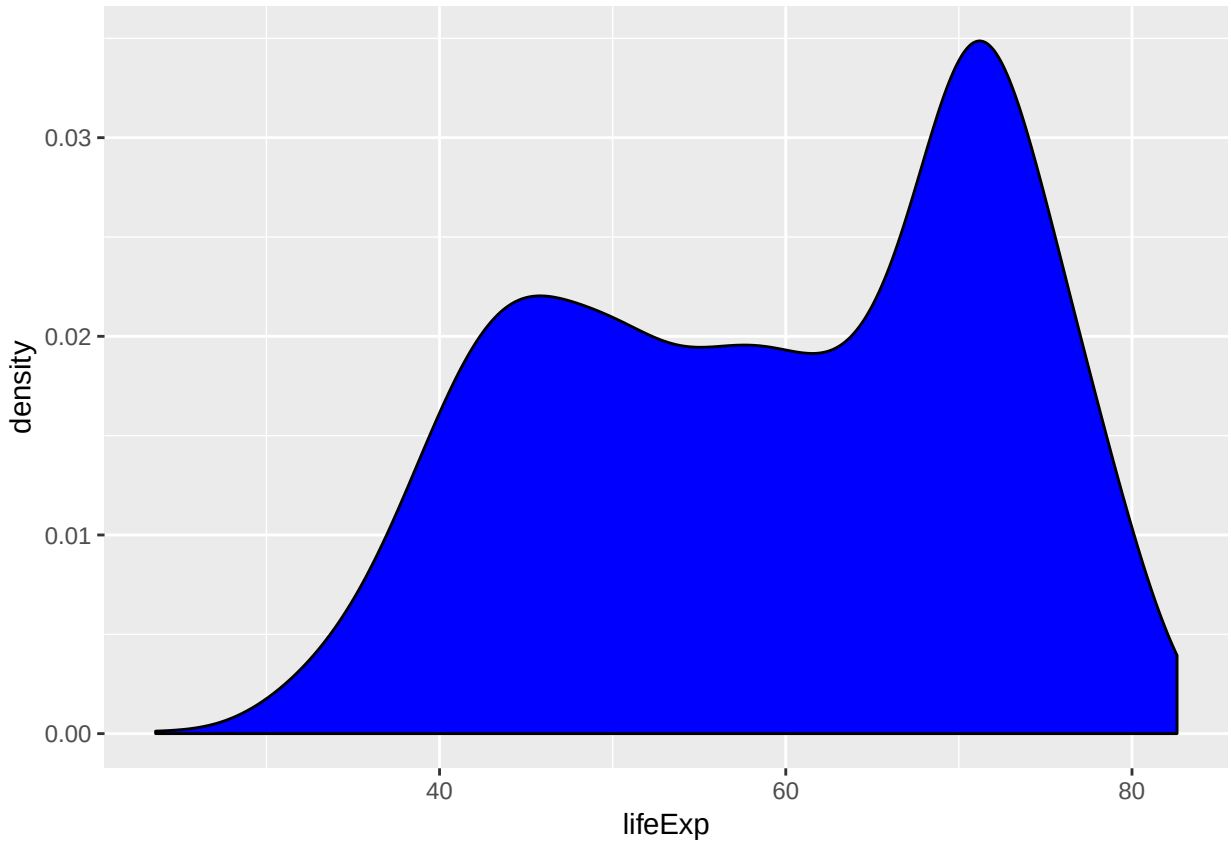


4. Instead of a histogram, let's create a kernel density plot.

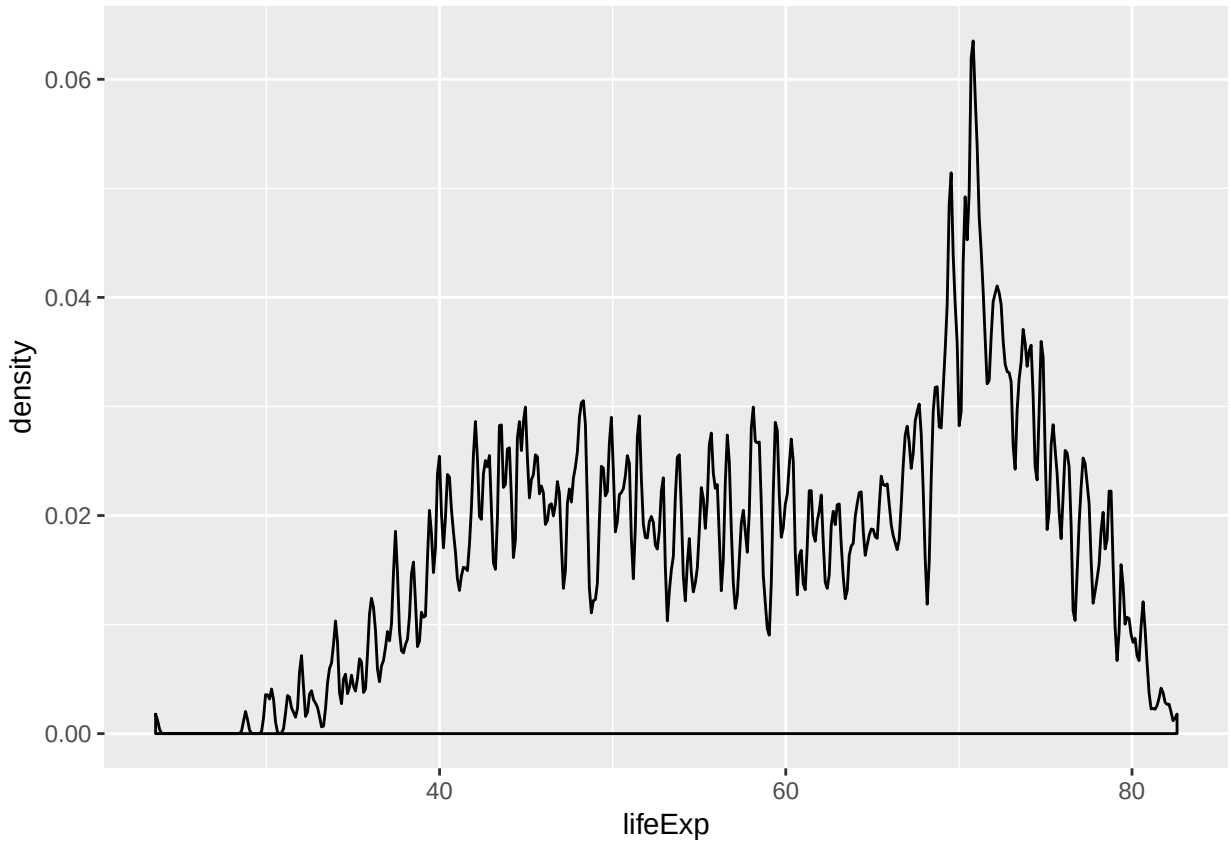
```
ggplot(gapminder, aes(lifeExp)) +
  geom_density()
```




```
ggplot(gapminder, aes(lifeExp)) +
  geom_density(fill = "blue")
```

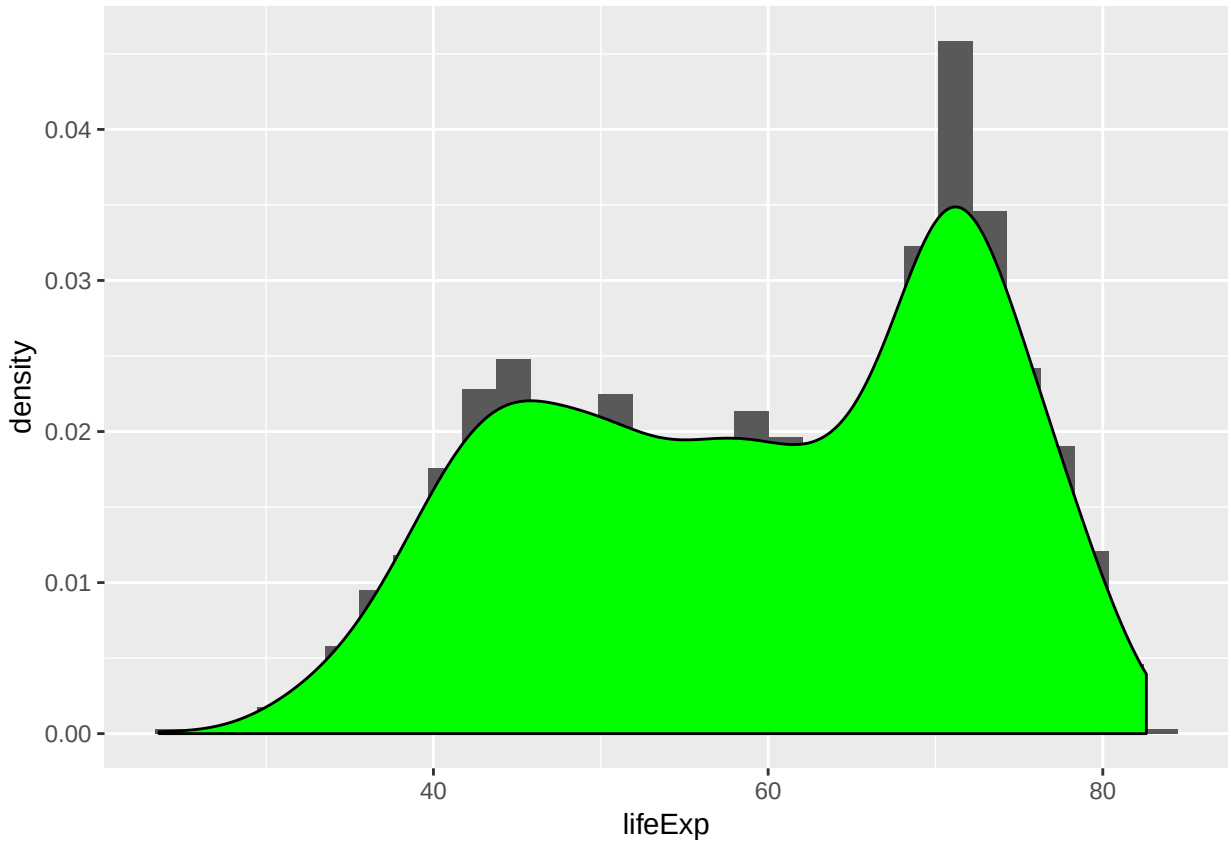



```
ggplot(gapminder, aes(lifeExp)) +
  geom_density(bw=0.1)
```



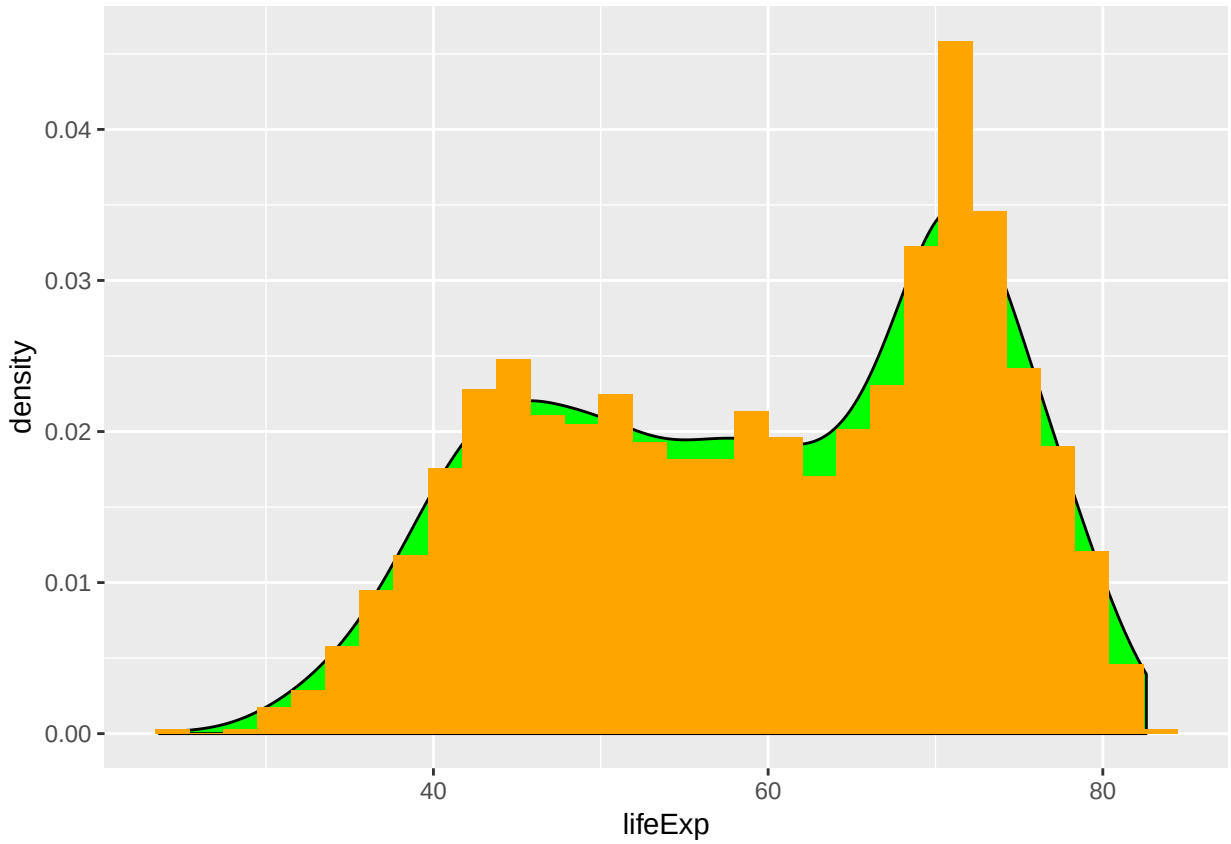

```
ggplot(gapminder, aes(lifeExp)) +
  geom_histogram(aes(y=..density..)) +
  geom_density(fill='green')
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
ggplot(gapminder, aes(lifeExp)) +
  geom_density(fill='green') +
  geom_histogram(aes(y=..density..), fill='orange')
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



5.

Optional: git stage and commit

Uses of a histogram: Explore the distribution of a single numeric variable.

Box plots, and violin plots

Let's make *box plots* of population for each continent. Note: y-axis is much better on a log scale!

1. Fill out the grammar components below. Again, bold *must* be specified to make a `ggplot2` plot.

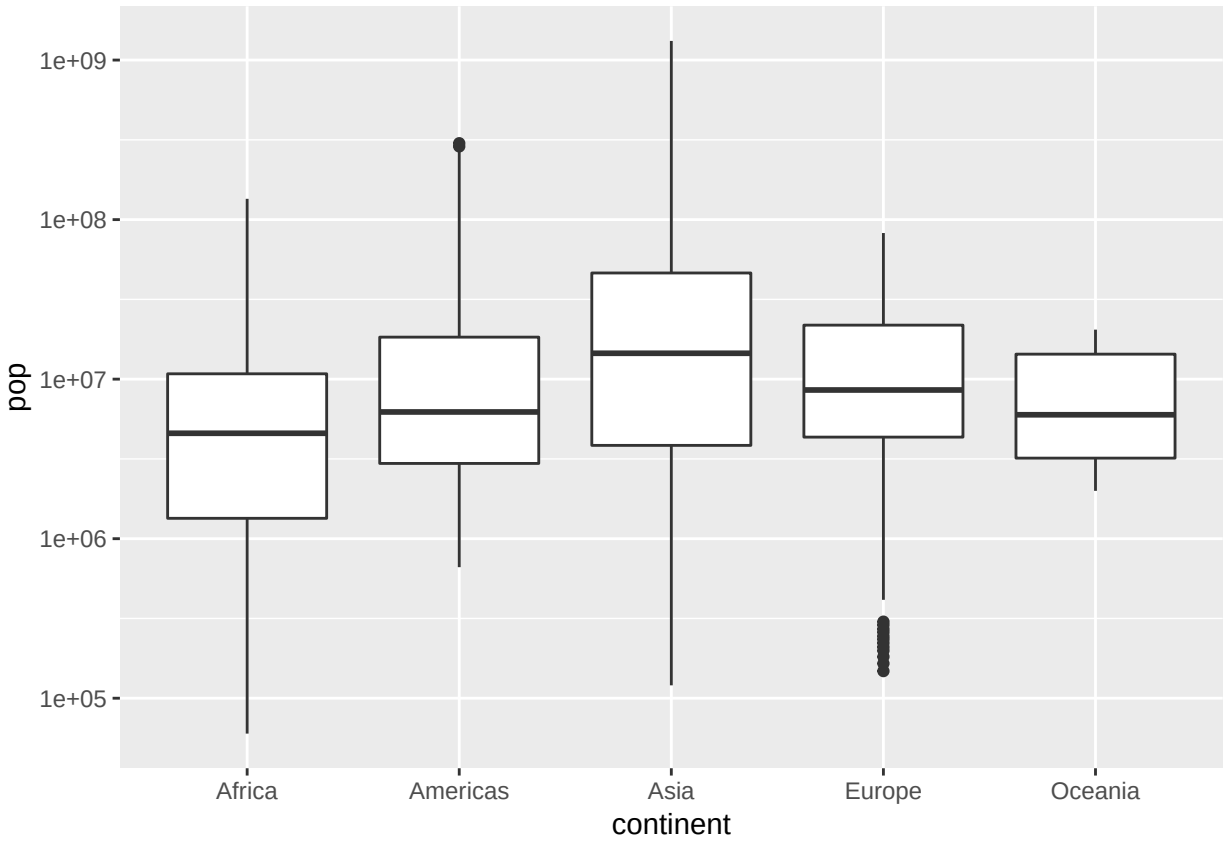
Grammar Component	Specification
data	`gapminder`
aesthetic mapping	`x` and `y`
geometric object	boxplot
scale	log-y
statistical transform	5-number summary

2. Initiate the `ggplot` call, with the log y scale, and store it in the variable `a`. Print out `a`.

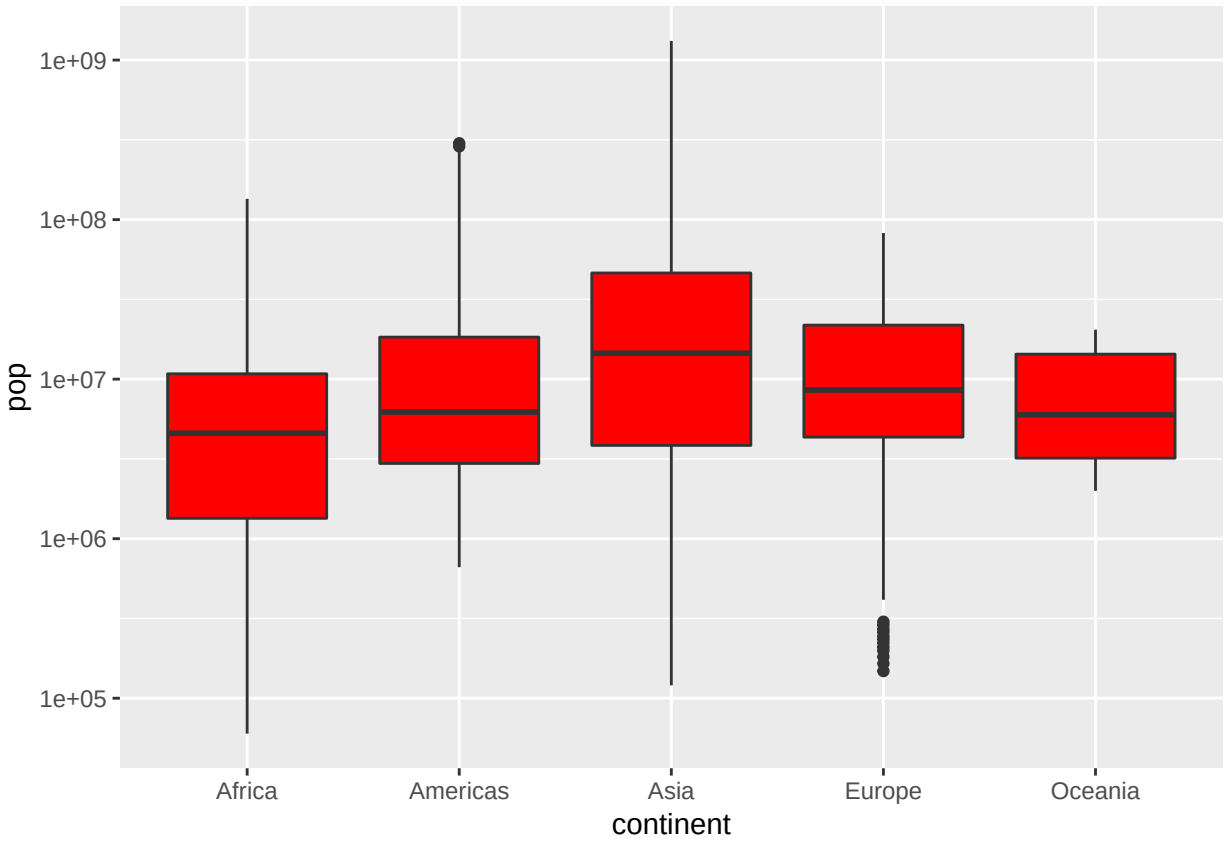
```
A <- ggplot(gapminder, aes(continent,pop)) +
    scale_y_log10()
```

3. Add the boxplot geom to `a`.

```
A +
  geom_boxplot()
```

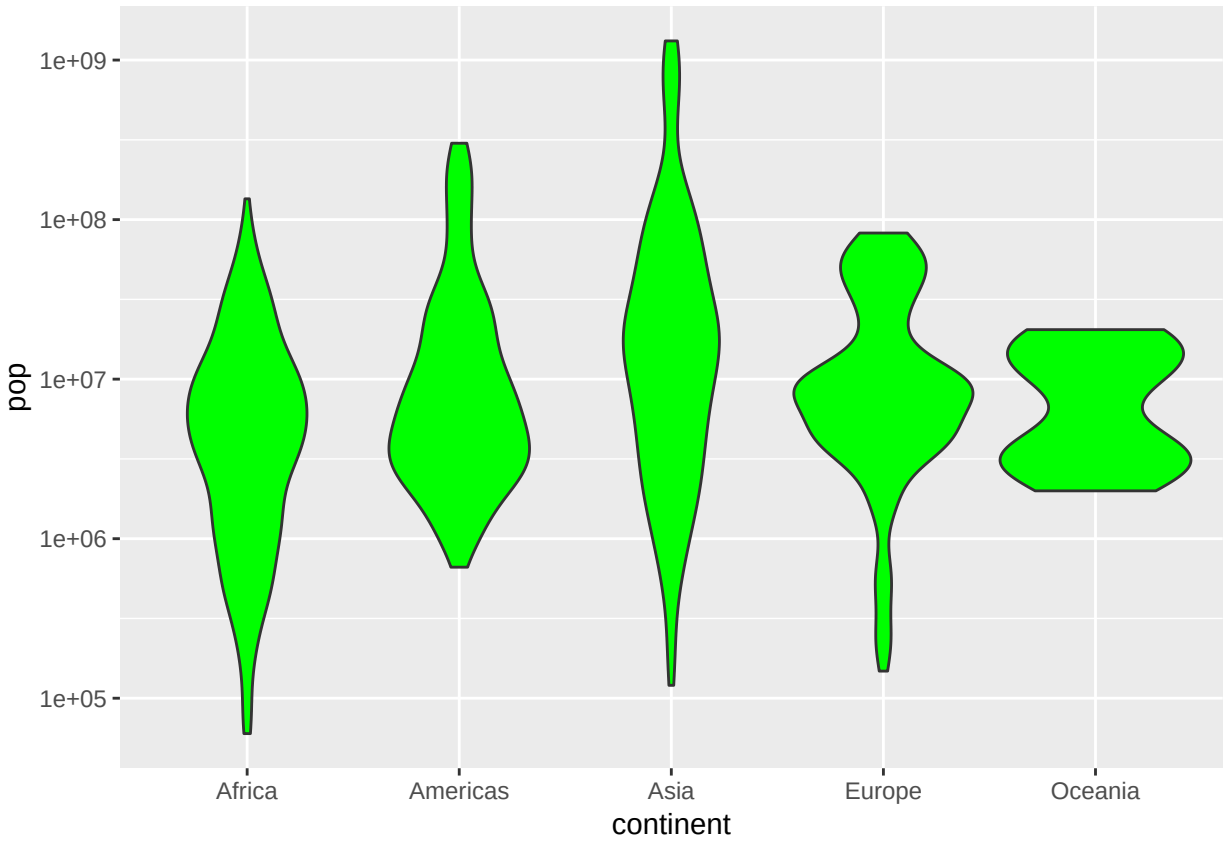



```
A +
  geom_boxplot(fill = 'red')
```



4. A violin plot is a kernel density on its side, made symmetric. Add that geom to `a`.
   - What's better here, boxplots or violin plots? Why?

```
A + geom_violin(fill='green')
```



This gives us more information about the data. The thickness show the density of data with each value.

5. Optional: git stage and commit

Use of boxplot: Visualize 1-dimensional distributions (of a single numeric variable).

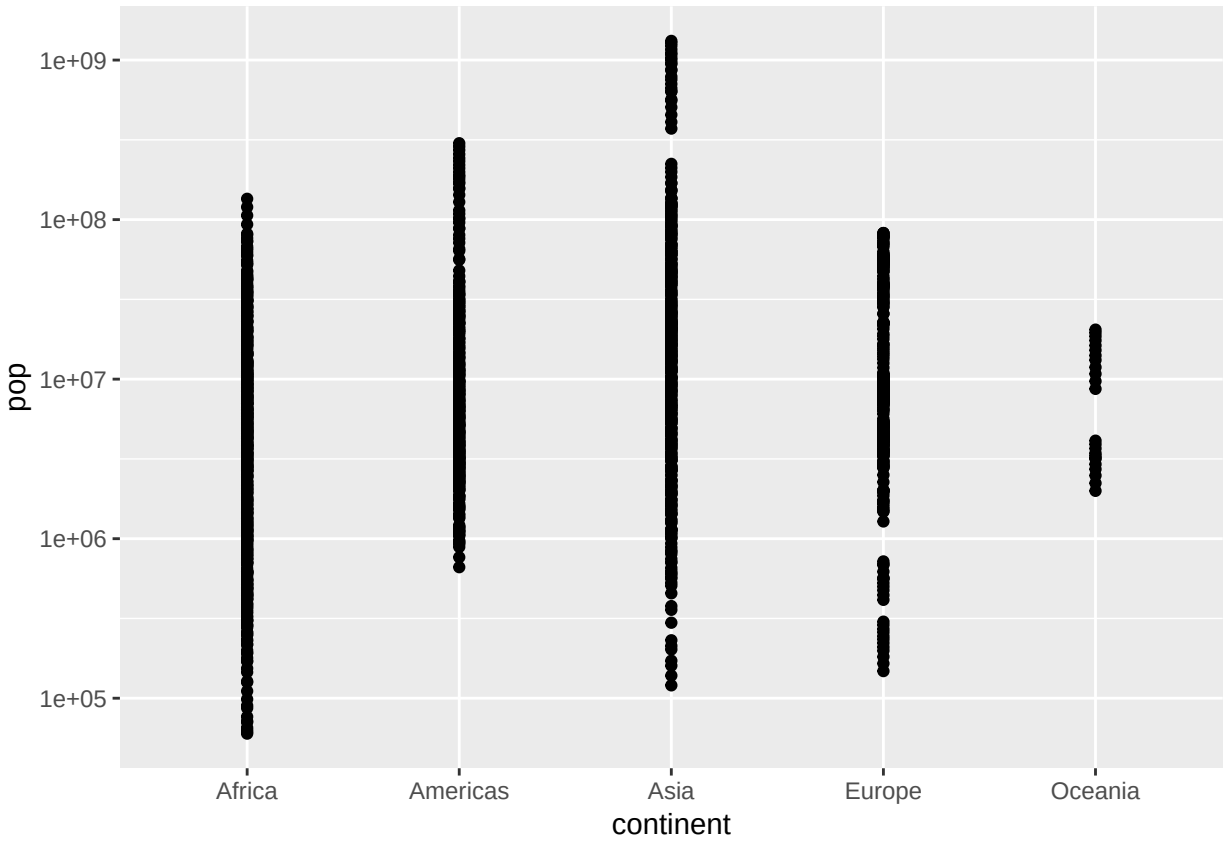
Jitter plots

Let's work up to the concept of a *jitter plot*. As above, let's explore the population for each continent, but using points (again, with the y-axis on a log scale).

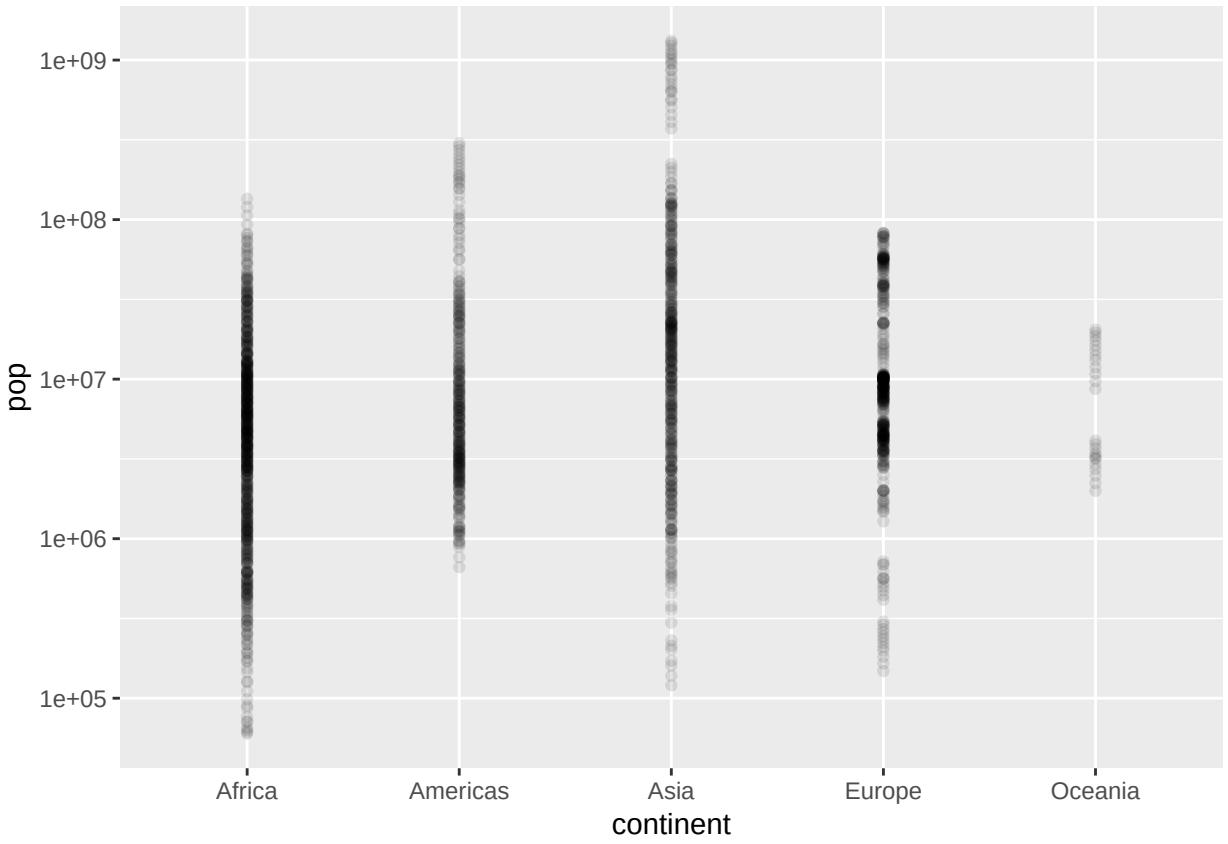
Let's hold off on identifying the grammar.

1. Initiate the `ggplot` call to make a scatterplot of `continent` vs `pop`; initiate the log y scale. Store the call in the variable `b`.

```
A + geom_point()
```



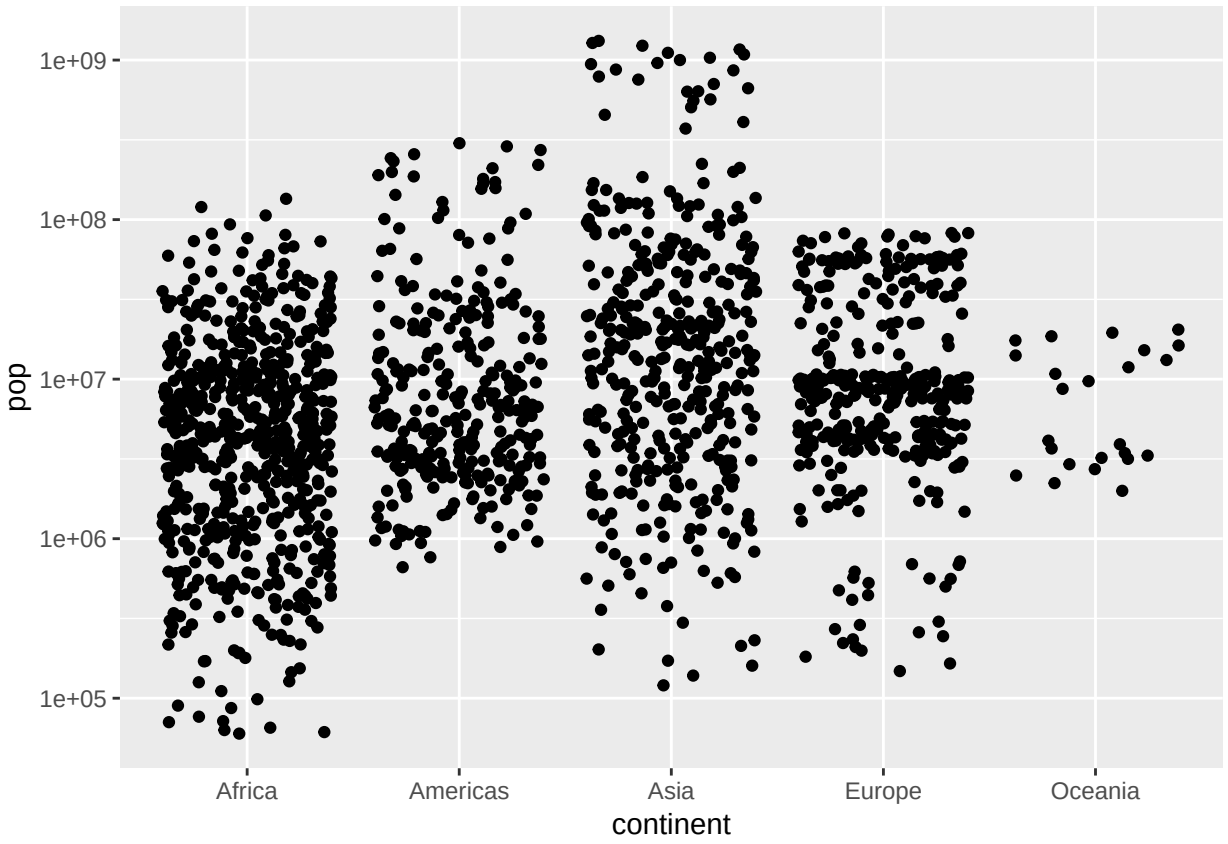

```
A + geom_point(alpha=0.1)
```



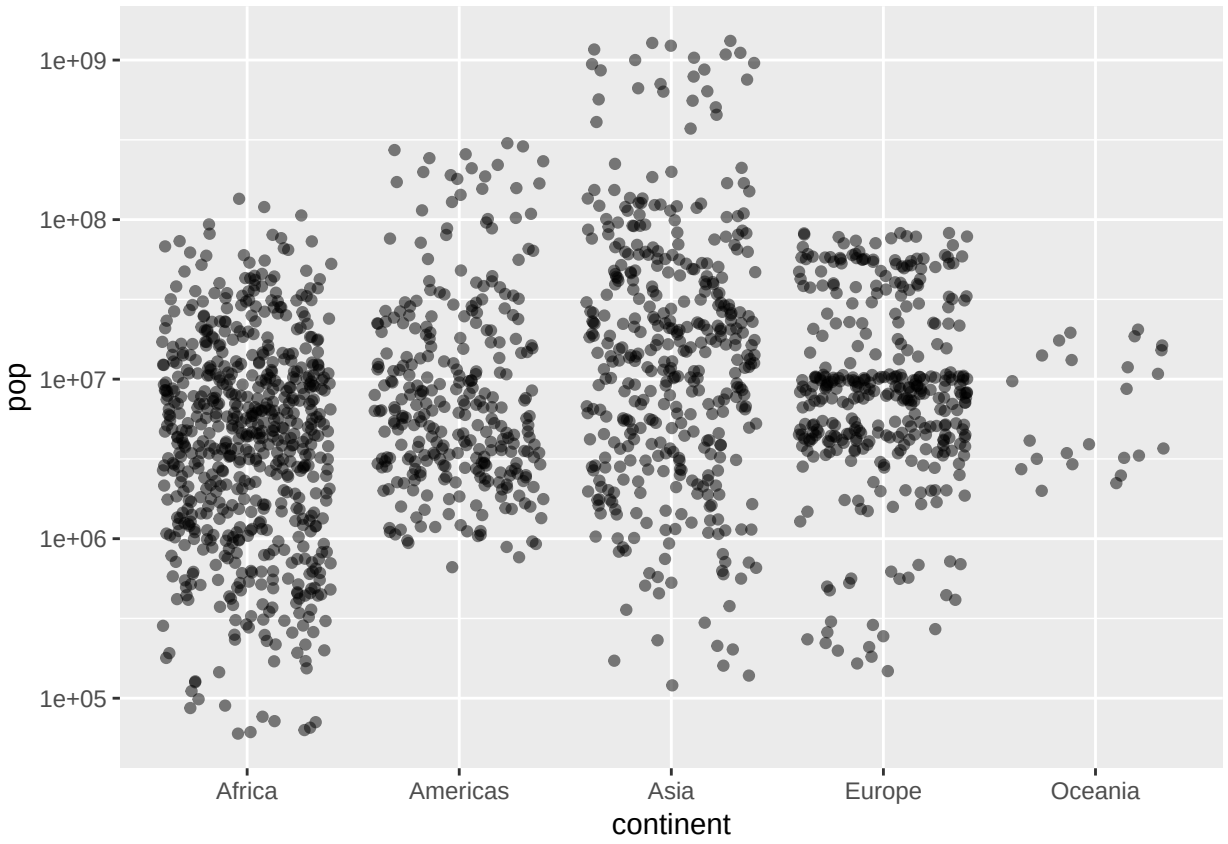
The thickness shows the density.

2. Add the point geom to `b`. Why is this an ineffective plot?
3. A solution is to jitter the points. Add the jitter geom. Re-run the command a few times – does the plot change? Why?

```
A + geom_jitter()
```




```
A + geom_jitter(alpha = 0.5)
```

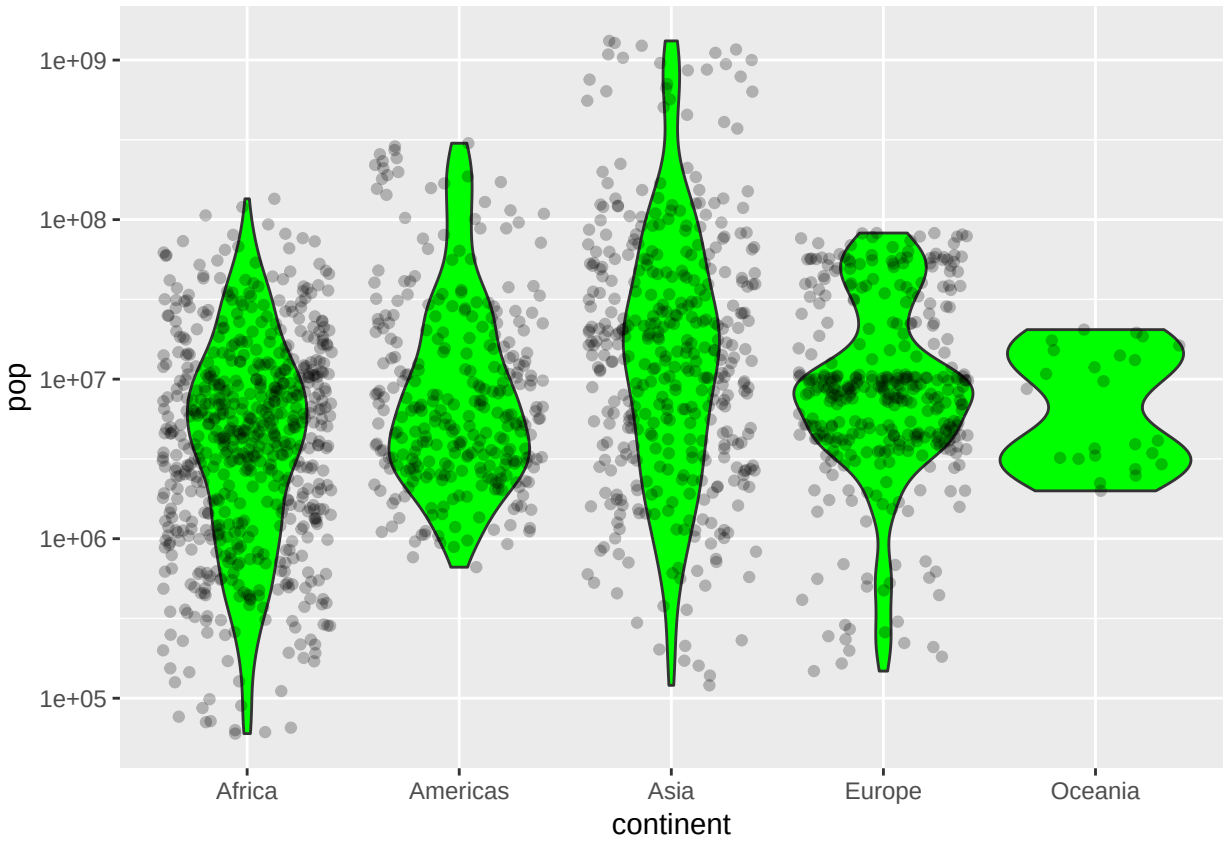



```
head(gapminder)
```

```
## # A tibble: 6 x 6
##   country     continent  year lifeExp      pop gdpPercap
##   <fct>       <fct>     <int>   <dbl>    <int>     <dbl>
## 1 Afghanistan Asia       1952    28.8  8425333      779.
## 2 Afghanistan Asia       1957    30.3  9240934      821.
## 3 Afghanistan Asia       1962    32.0 10267083      853.
## 4 Afghanistan Asia       1967    34.0 11537966      836.
## 5 Afghanistan Asia       1972    36.1 13079460      740.
## 6 Afghanistan Asia       1977    38.4 14880372      786.
```

4. How does the grammar differ from a box plot or violin plot?
    - ANSWER:

```
A + geom_violin(fill = 'green') + geom_jitter(alpha = 0.25, fill = 'red')
```



5. We can add multiple geom *layers* to our plot. Put a jitterplot overtop of the violin plot, starting with our base `b`. Try vice-versa.
6. Optional: git stage and commit

Uses of jitterplot: Visualize 1-dimensional distributions, AND get a sense of the sample size.

Time/Line Plots

Let's make some time/line plot, starting with Canada's life expectancy over time.

1. Fill out the grammar components below. Again, bold *must* be specified to make a `ggplot2` plot.

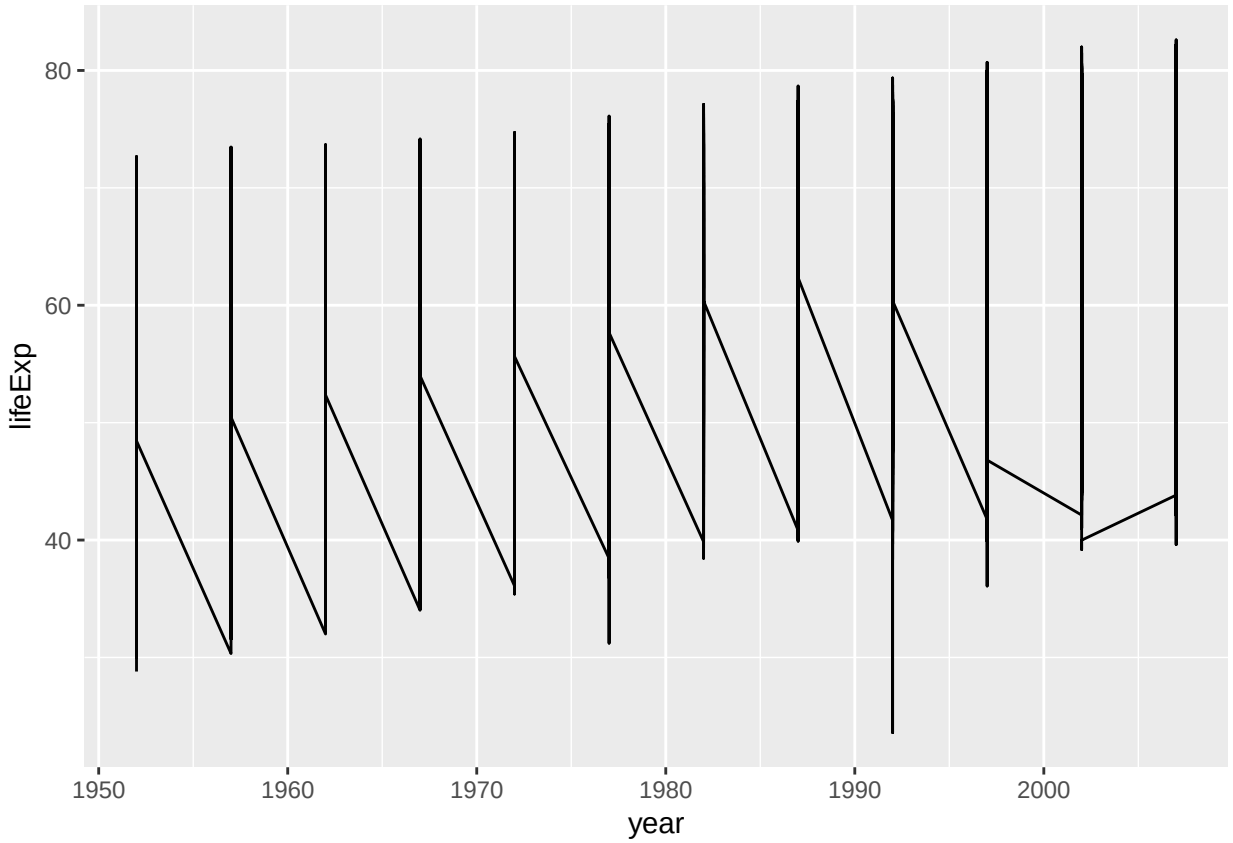
Grammar Component	Specification
data	`gapminder`
aesthetic mapping	
geometric object	
scale	
statistical transform	

2. In one readable call, write code that:
    1. Filters the data to Canada only
    2. Pipes the filtered data into `ggplot`
    3. Makes the time plot of `lifeExp` over time
    4. Also displays the points

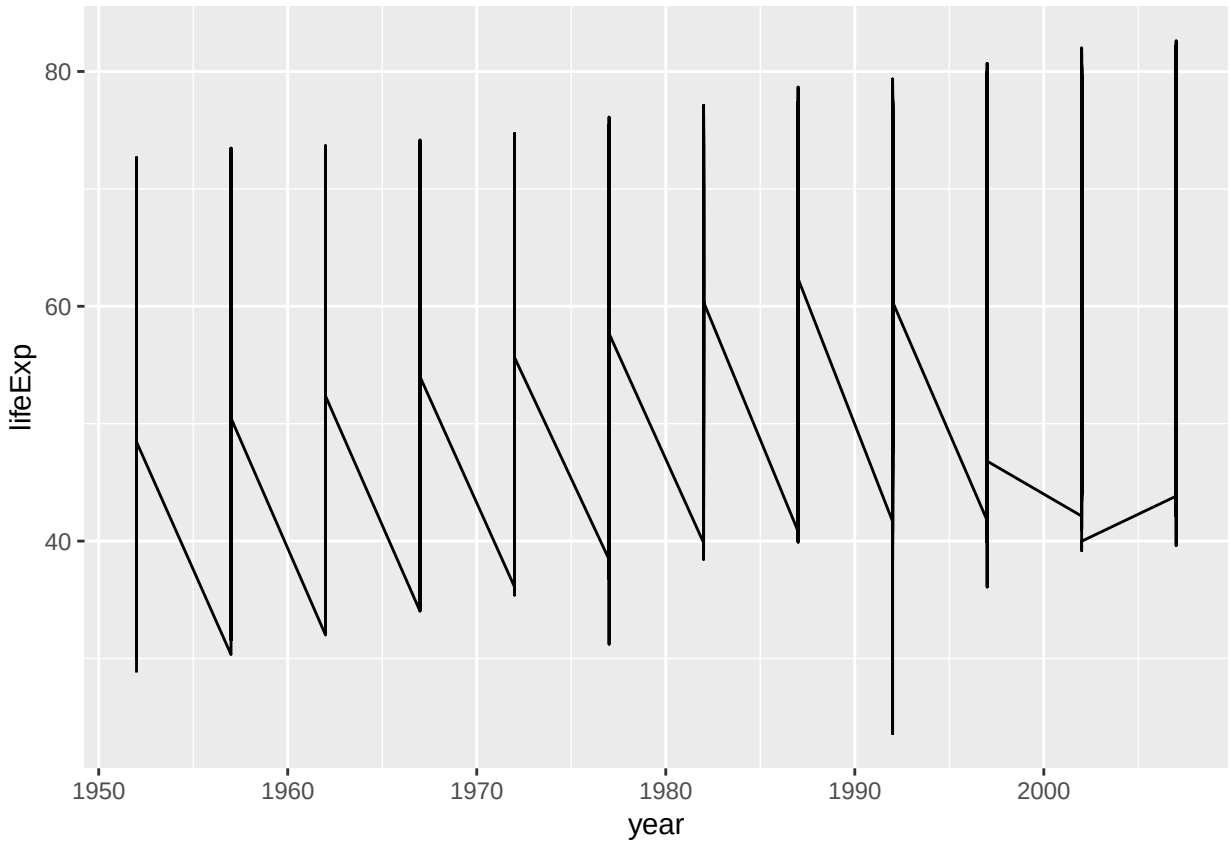
```
#gapminder %>%
  #filter(country == "Canada") %>%
  #  ggplot(aes(year,lifeExp)) +
   #   geom_line() +
        #geom_point()
```

3. Attempt to overlay line plots for all countries. That is, repeat the above code, but don't filter. What's wrong here?

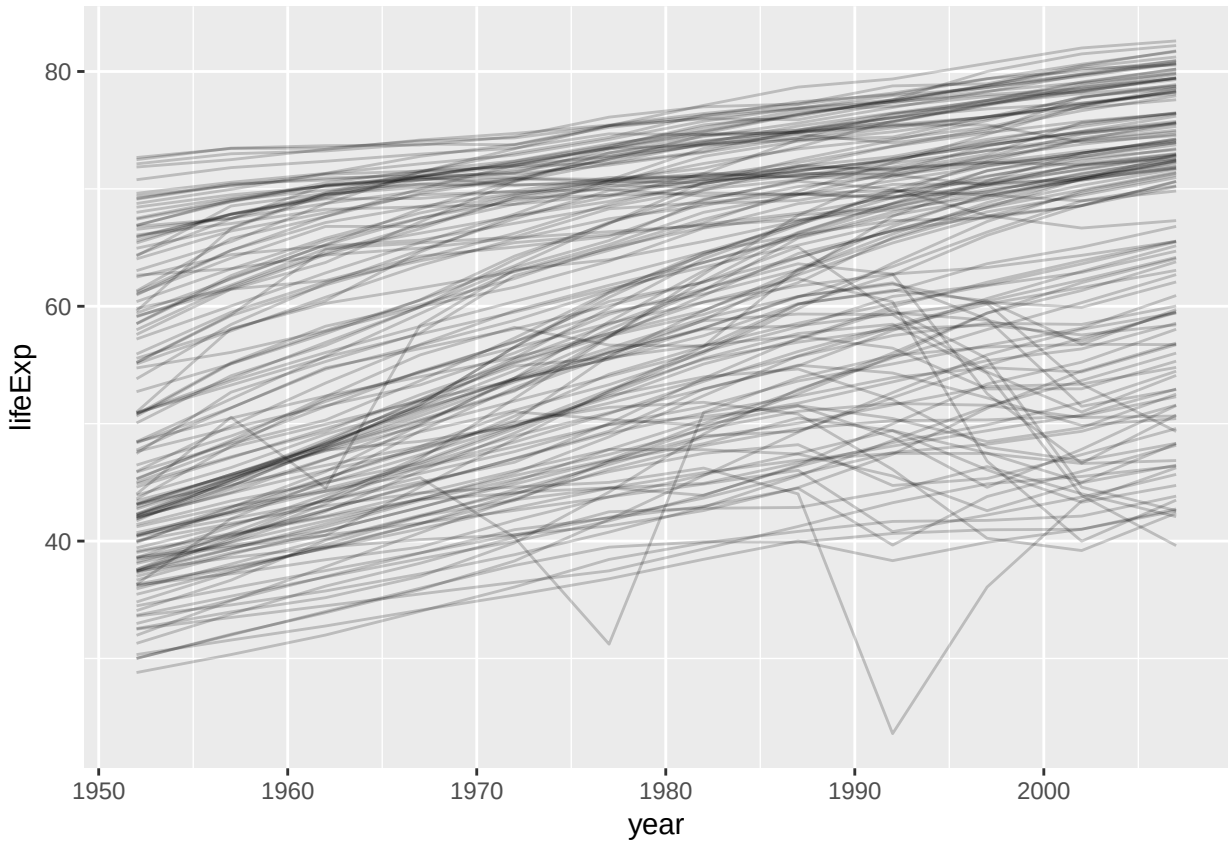
```
ggplot(gapminder, aes(year, lifeExp)) + geom_line()
```




```
c <- ggplot(gapminder, aes(year, lifeExp))
c + geom_line()
```




```
c + geom_line(aes(group=country), alpha= 0.2)
```



4. Use the `group` aesthetic to fix the problem.
5. Optional: git stage and commit

Uses of time/line plots: Visualize trends of a numeric variable over time.

Path plots

Let's see how Rwanda's life expectancy and GDP per capita have evolved over time, using a path plot.

1. Make a scatterplot. Store it in the variable `c`.
2. We want to connect the dots from earliest point to latest. What happens if we add the "line" geom to `c`?
3. Add the appropriate geom to `c`. In that geom, specify a property of the geom: `arrow=arrow()`.
4. Optional: git stage and commit

Uses of path plots: The four "corners" of the plot usually indicate different qualities. This plot allows you to see how Rwanda (or some entity) evolves over these qualities.

Bar plots

How many countries are in each continent? Use the year 2007.

1. Fill out the grammar components below. Again, bold *must* be specified to make a `ggplot2` plot.

Grammar Component	Specification
data	`gapminder`
aesthetic mapping	
geometric object	
scale	
statistical transform	

2. After filtering the gapminder data to 2007, make a bar chart of the number of countries in each continent. Store everything except the geom in the variable `d`.
3. Notice the y-axis. Oddly, `ggplot2` doesn't make it obvious how to change to proportion. Try adding a `y` aesthetic: `y=..count../sum(..count..)`.
4. Optional: git stage, commit, and push!

Uses of bar plots: Get a sense of relative quantities of categories, or see the probability mass function of a categorical random variable.