1.4 Data Visualization with ggplot2 - R Practice (Answers)

Ryan Safner

ECON 480 - Fall 2019

Getting Set Up

Before we begin, start a new file with `File` → `New File` → `R Script`. As you work through this sheet in the console in `R`, also add (copy/paste) your commands that work into this new file. At the end, save it, and run to execute all of your commands at once.

Exploring the Data

1. We will look at GDP per Capita and Life Expectancy using some data from the gapminder project. There is a handy package called `gapminder` that uses a small snippet of this data for exploratory analysis. Install and load the package `gapminder`. Type `?gapminder` and hit enter to see a description of the data.

```
# first time only
# install.packages("gapminder")

# load gapminder
library(gapminder)

# get help
?gapminder
```

2. Let's get a quick look at `gapminder` to see what we're dealing with.

a. Get the `structure` of the `gapminder` data.

```
str(gapminder)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    1704 obs. of  6 variables:
##  $ country  : Factor w/ 142 levels "Afghanistan",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ continent: Factor w/ 5 levels "Africa","Americas",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ year     : int  1952 1957 1962 1967 1972 1977 1982 1987 1992 1997 ...
##  $ lifeExp  : num  28.8 30.3 32 34 36.1 ...
##  $ pop      : int  8425333 9240934 10267083 11537966 13079460 14880372 12881816 13867957 16317921 222
##  $ gdpPercap: num  779 821 853 836 740 ...
```

b. What variables are there?

```
# - country: a factor
# - continent: a factor
# - year: an integer
# - lifeExp: a number
# - gdpPercap: a number
```

c. Look at the `head` of the dataset to get an idea of what the data looks like.

```
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.
```

d. Get `summary` statistics of all variables.

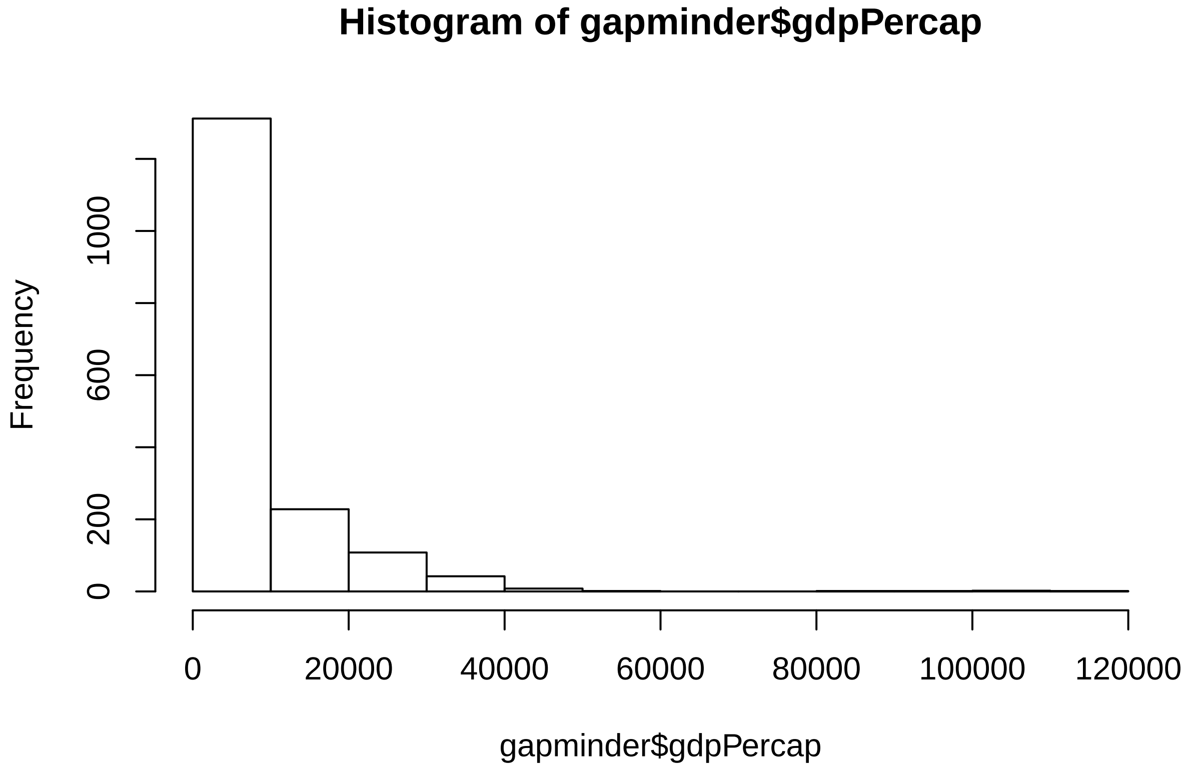
```
summary(gapminder)
```

```
##         country        continent        year         lifeExp
##  Afghanistan:  12   Africa  :624   Min.   :1952   Min.   :23.60
##  Albania    :  12   Americas:300   1st Qu.:1966   1st Qu.:48.20
##  Algeria    :  12   Asia    :396   Median :1980   Median :60.71
##  Angola     :  12   Europe  :360   Mean   :1980   Mean   :59.47
##  Argentina  :  12   Oceania : 24   3rd Qu.:1993   3rd Qu.:70.85
##  Australia  :  12                  Max.   :2007   Max.   :82.60
##  (Other)    :1632
##       pop              gdpPercap
##  Min.   :6.001e+04   Min.   :   241.2
##  1st Qu.:2.794e+06   1st Qu.:  1202.1
##  Median :7.024e+06   Median :  3531.8
##  Mean   :2.960e+07   Mean   :  7215.3
##  3rd Qu.:1.959e+07   3rd Qu.:  9325.5
##  Max.   :1.319e+09   Max.   :113523.1
##
```

Simple Plots in Base R

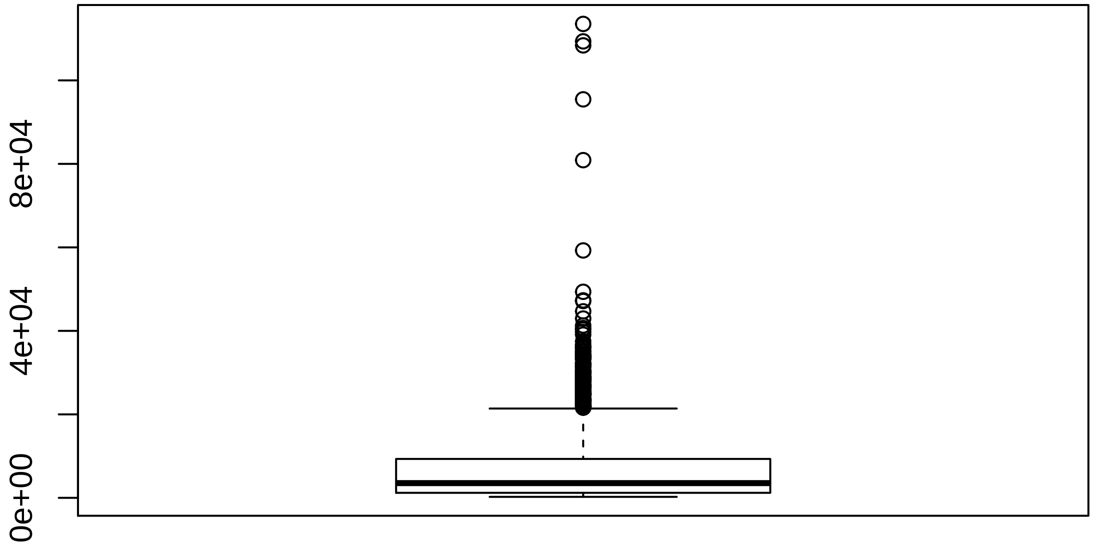
3. Let's make sure you can do some basic plots before we get into the gg. Use base `R`'s `hist()` function to plot a ***histogram*** of `gdpPercap`.

```
hist(gapminder$gdpPercap)
```



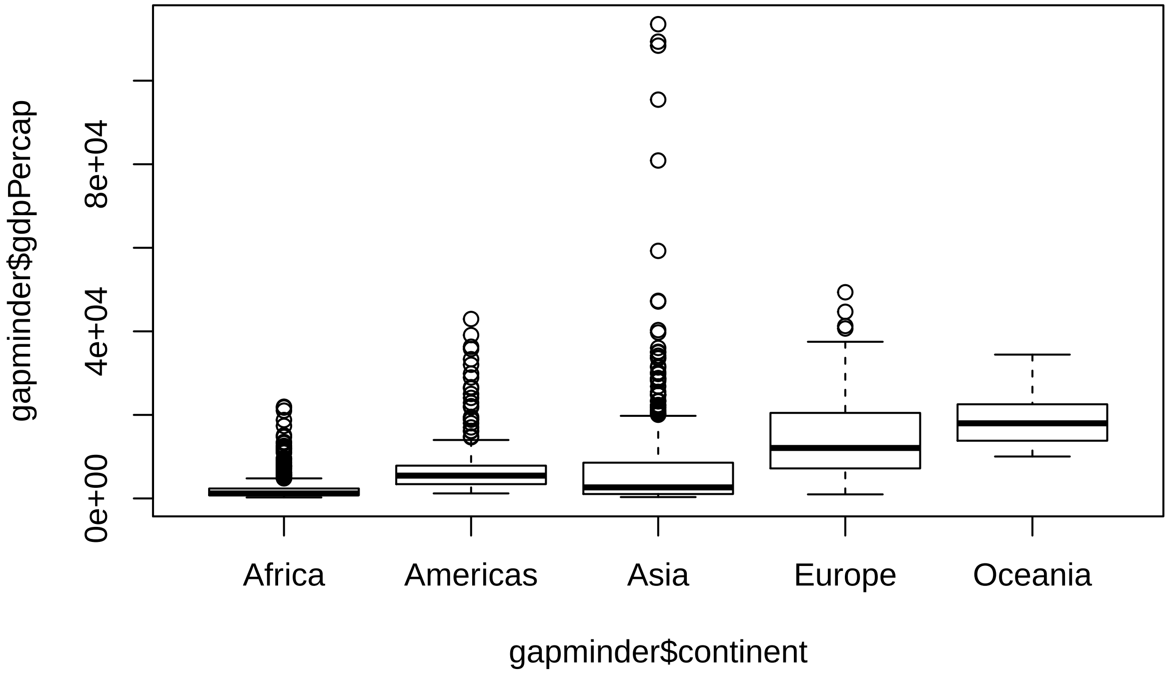
4. Use base R's `boxplot()` function to plot a *boxplot* of `gdpPercap`.

```
boxplot(gapminder$gdpPercap)
```



5. Now make it a *boxplot* by `continent`.[1]

```
boxplot(gapminder$gdpPercap~gapminder$continent)
```

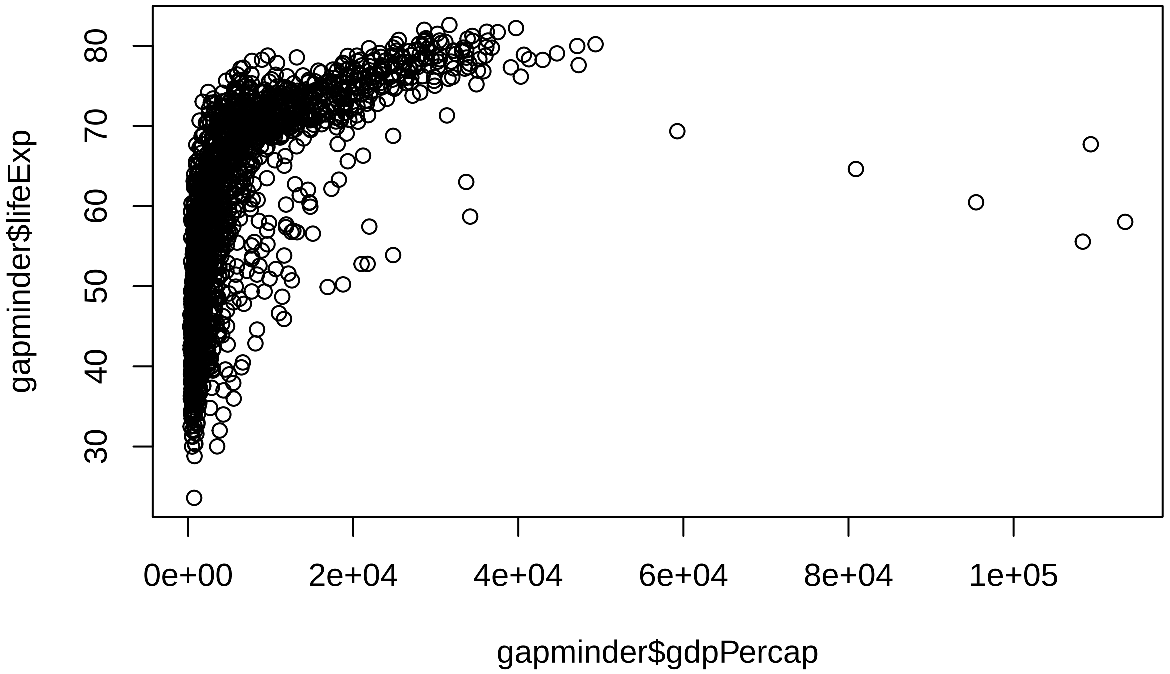


```
# alternate method
# boxplot(gdpPercap~continent, data = gapminder)
```

6. Now make a *scatterplot* of `gdpPercap` on the x-axis and `LifeExp` on the y-axis.

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

[1] Hint: use formula notation with~.



```
# alternate method
# boxplot(lifeExp~gdpPercap, data = gapminder)
```

Plots with ggplot2

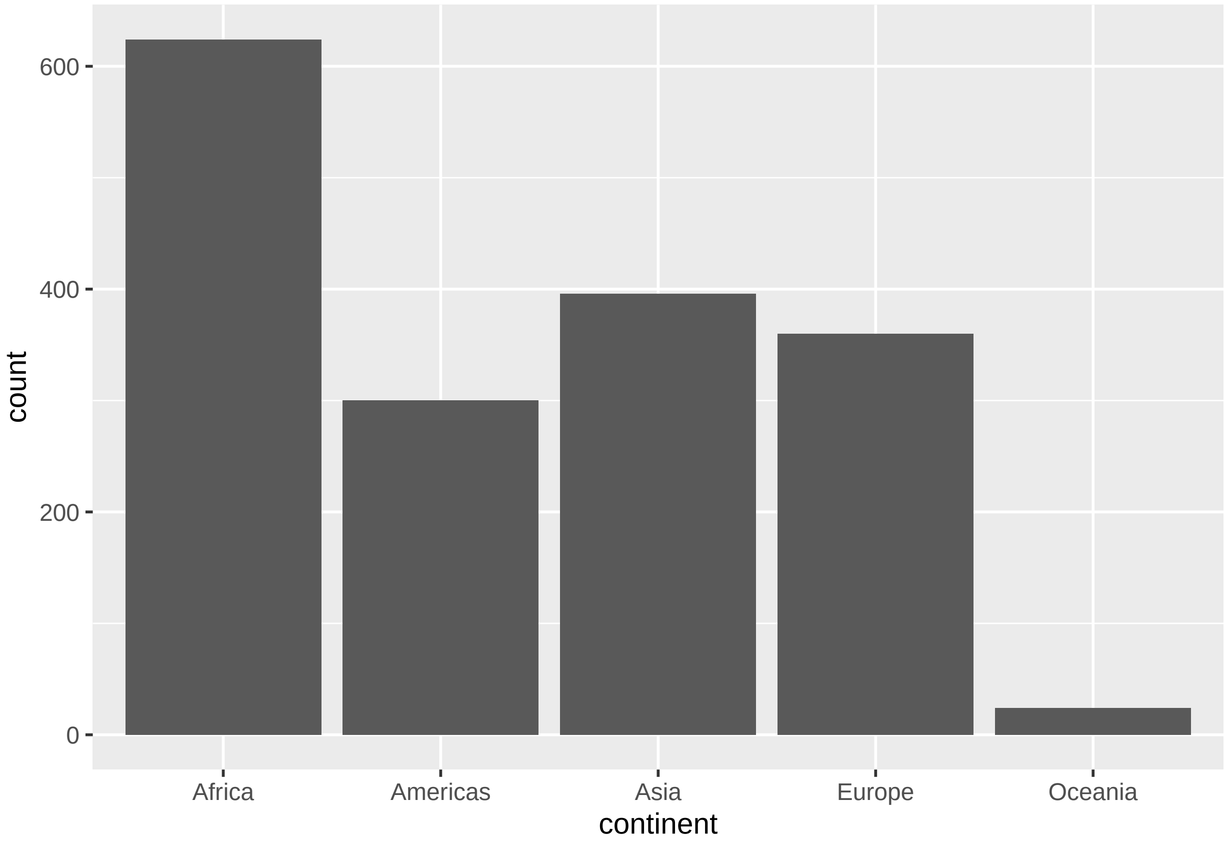
7. Load the package `ggplot2` (you should have installed it previously. If not, install first with `install.packages("ggplot2")`).

```
# install if you don't have
# install.packages("ggplot2")

# load ggplot2
library(ggplot2)
```

8. Let's first make a `bar` graph to see how many countries are in each continent. The only `aesthetic` you need is to map `continent` to `x`. Bar graphs are great for representing categories, but not quantitative data.

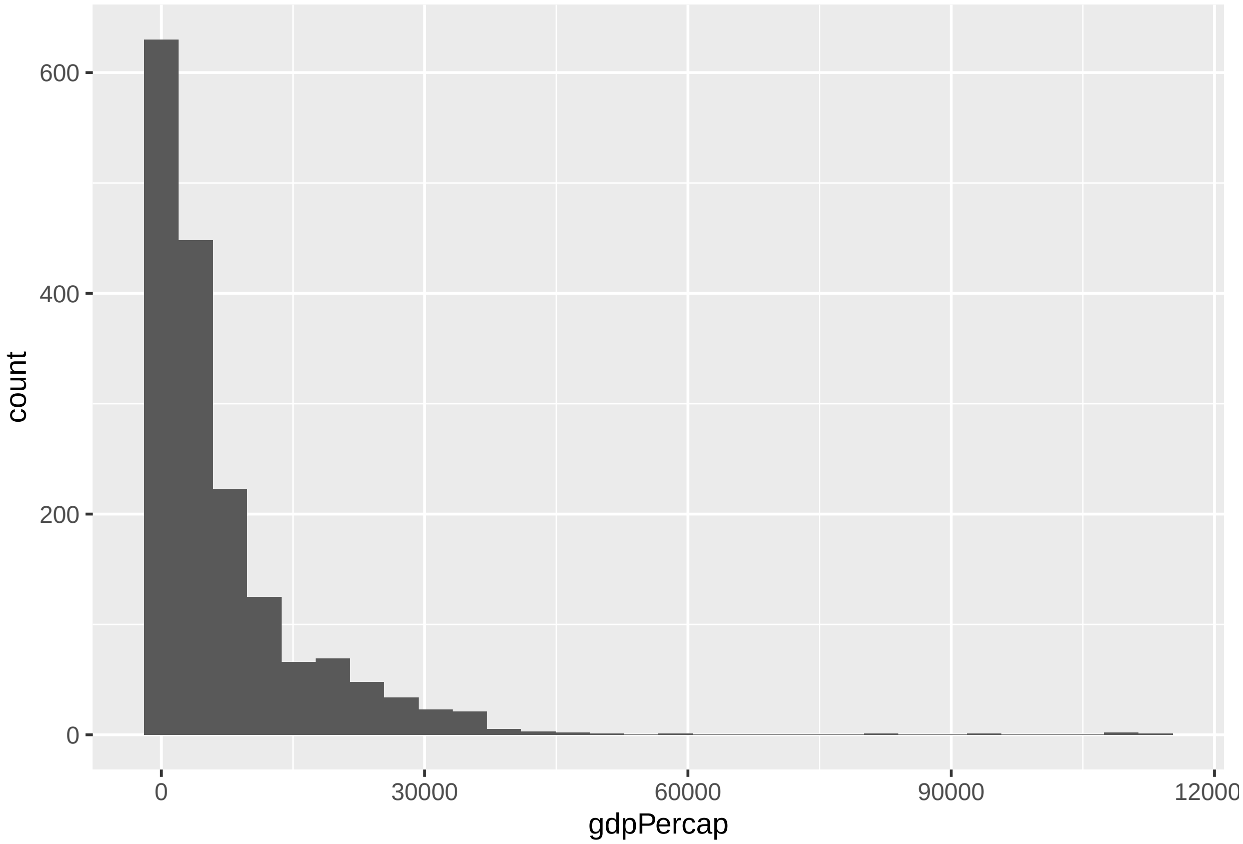
```
ggplot(data = gapminder,
       aes(x = continent))+
  geom_bar()
```



9. For quantitative data, we want a `histogram` to visualize the distribution of a variable. Make a `histogram` of `gdpPercap`. Your only aesthetic here is to map `gdpPercap` to `x`.

```
ggplot(data = gapminder,
       aes(x = gdpPercap))+
  geom_histogram()
```

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

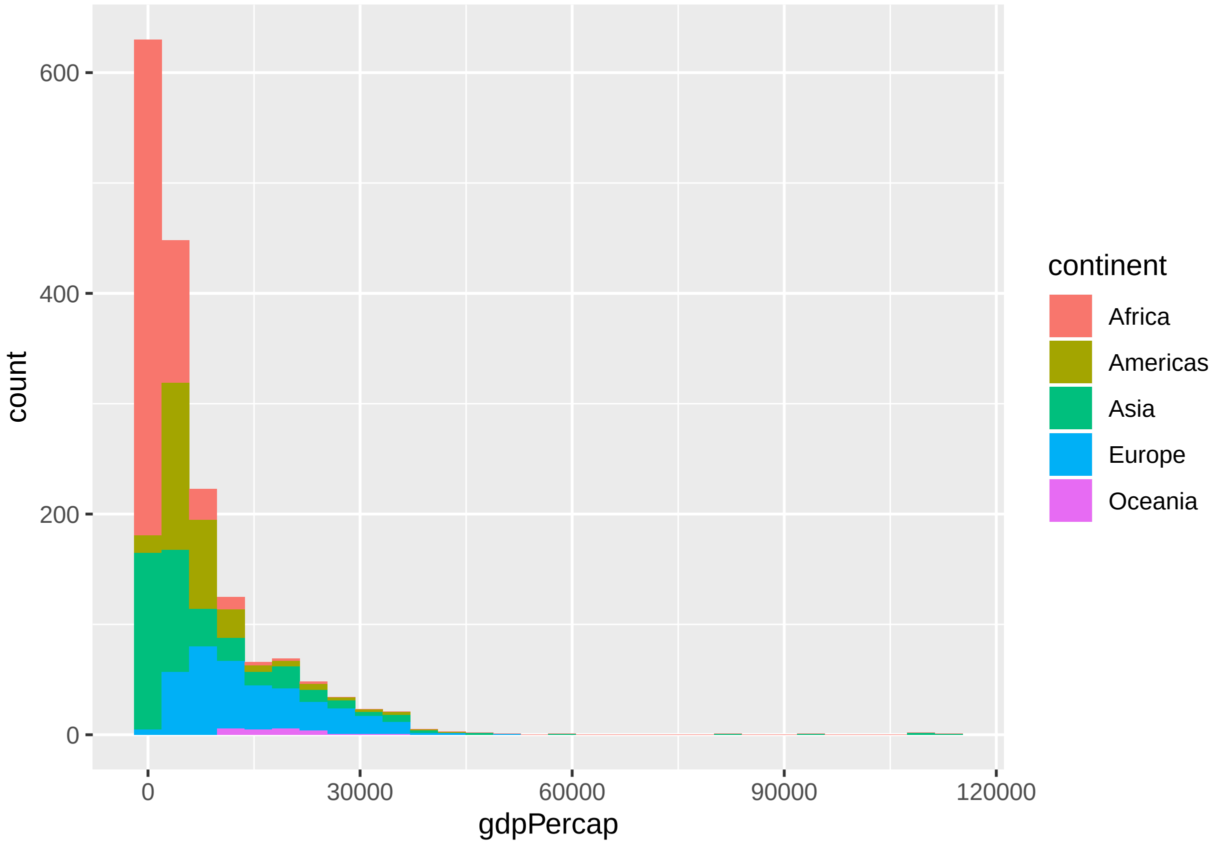


10. Now let's try adding some color, specifically, add an aesthetic that maps `continent` to `fill`.[2]

```
ggplot(data = gapminder,
       aes(x = gdpPercap,
           fill = continent))+
  geom_histogram()
```

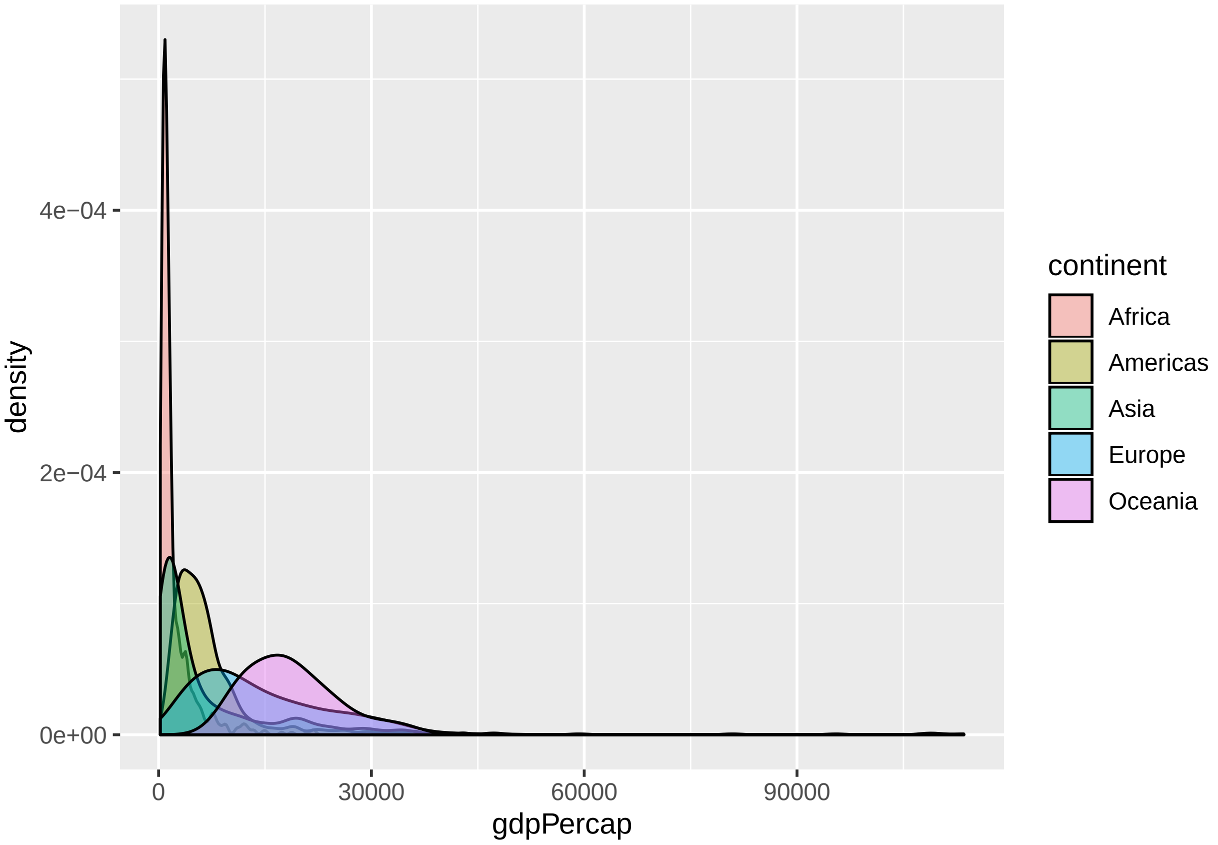
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

[2] In general, `color` refers to the outside borders of a `geom` (except points), `fill` is the interior of an object.



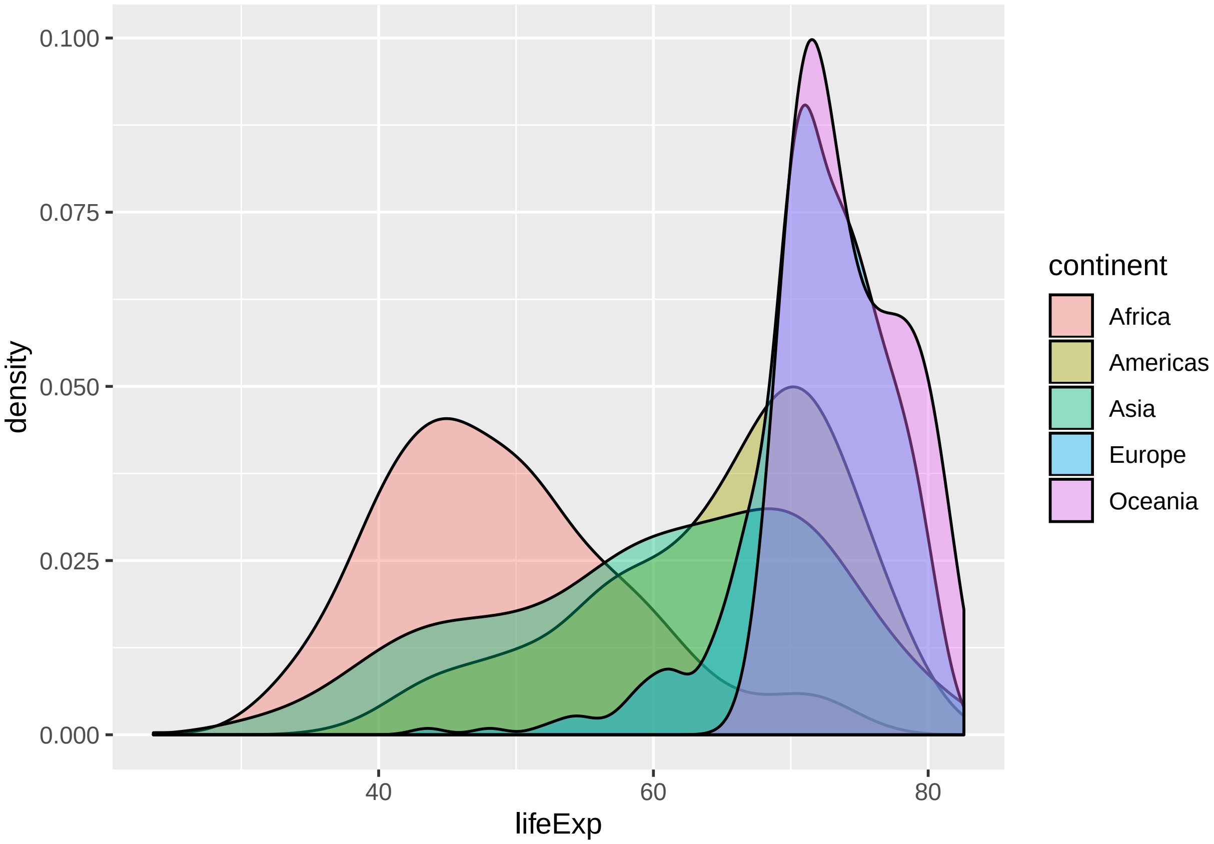
11. Instead of a `histogram`, change the `geom` to make it a `density` graph. To avoid overplotting, add `alpha=0.4` to the `geom` argument (alpha changes the *transparency* of a `fill`).

```
ggplot(data = gapminder,
       aes(x = gdpPercap,
           fill = continent))+
  geom_density(alpha=0.4)
```



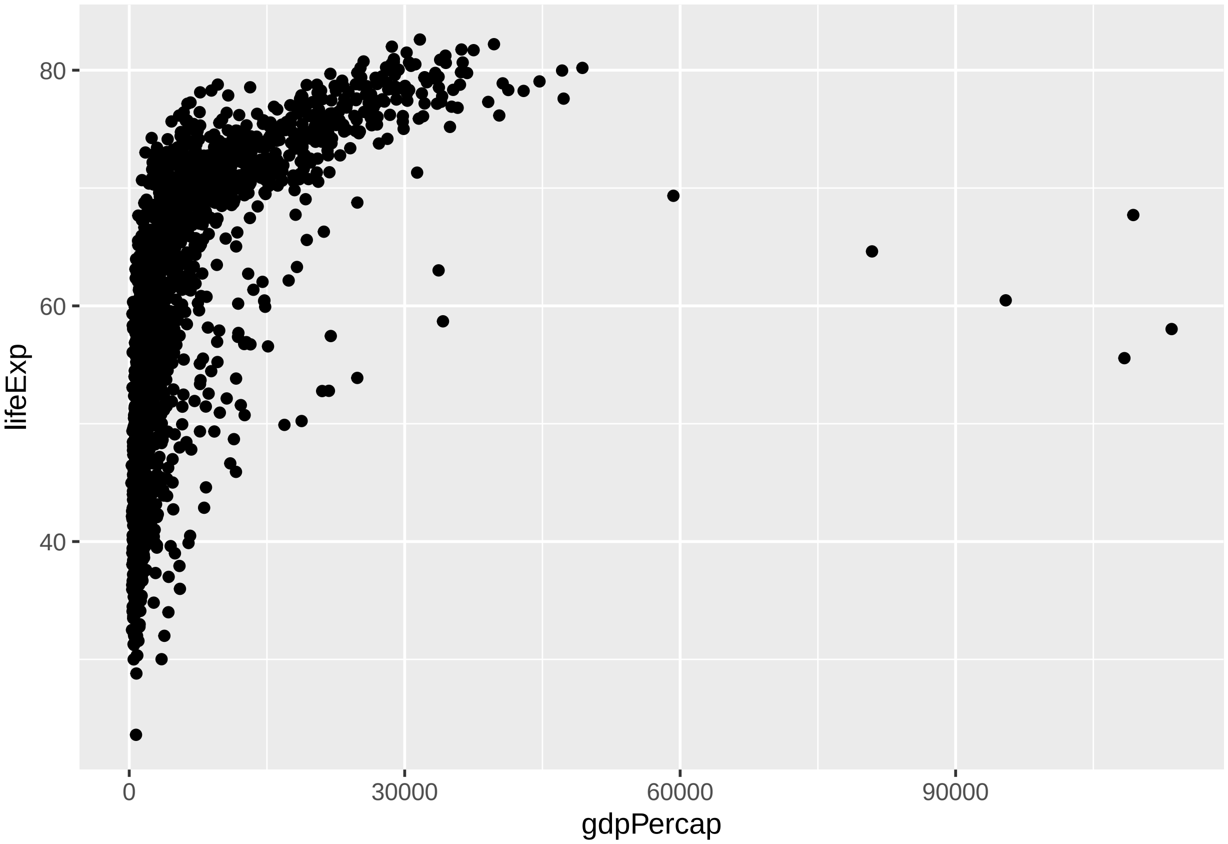
12. Redo your plot from 11 for `lifeExp` instead of `gdpPercap`.

```
ggplot(data = gapminder,
       aes(x = lifeExp,
           fill = continent))+
  geom_density(alpha=0.4)
```



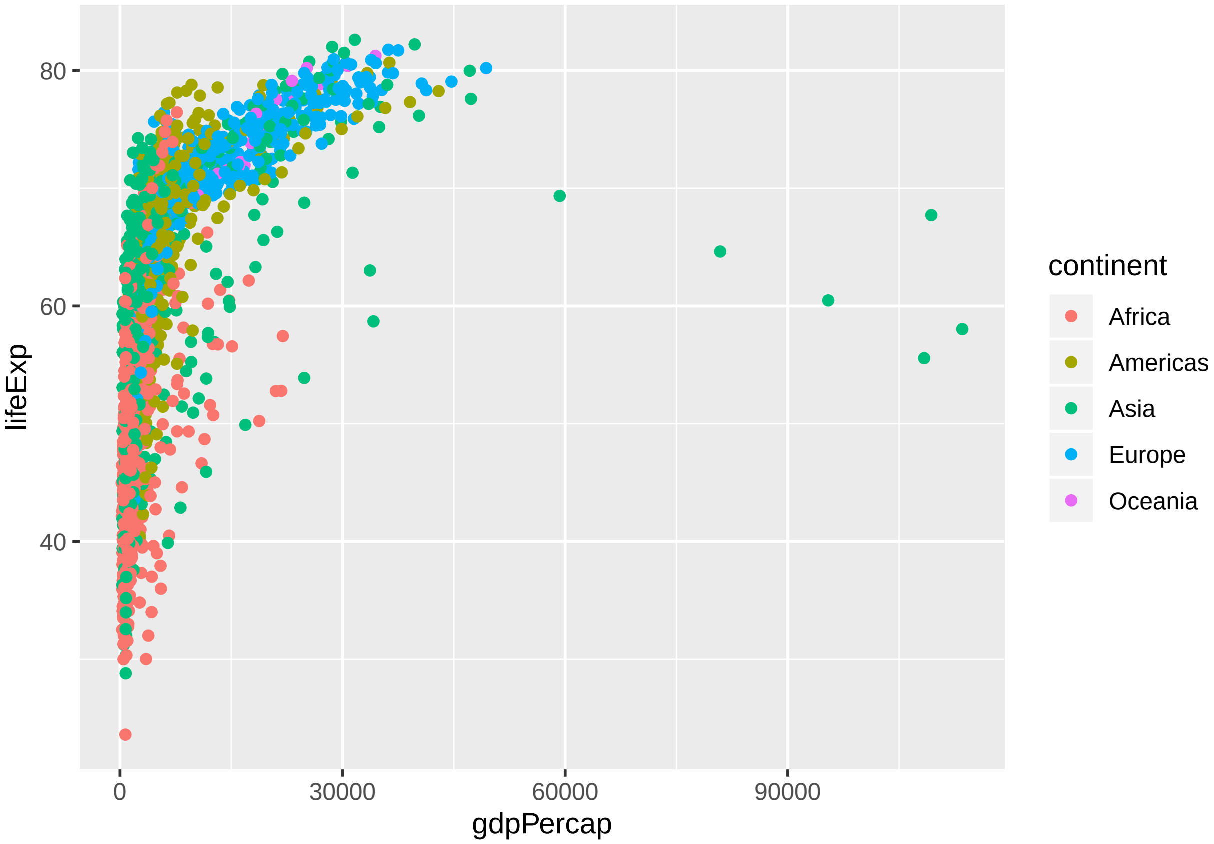
13. Now let's try a scatterplot for `lifeExp` (as `y`) on `gdpPercap` (as `x`). You'll need both for `aesthetics`. The `geom` here is `geom_point()`.

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



14. Add some color by mapping `continent` to `color` in your `aesthetics`.

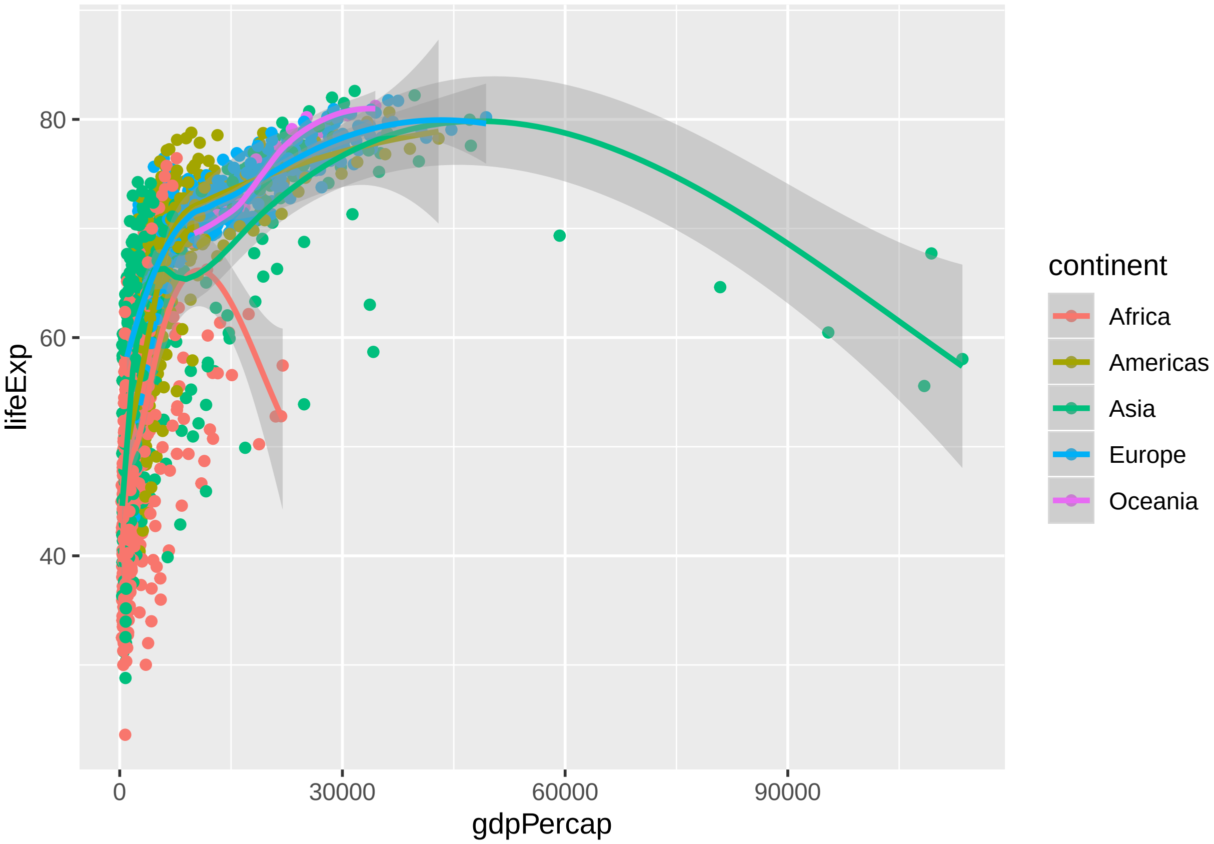
```
ggplot(data = gapminder,
       aes(x = gdpPercap,
           y = lifeExp,
           color = continent))+
  geom_point()
```



15. Now let's try adding a regression line with `geom_smooth()`. Add this layer on top of your `geom_point()` layer.

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

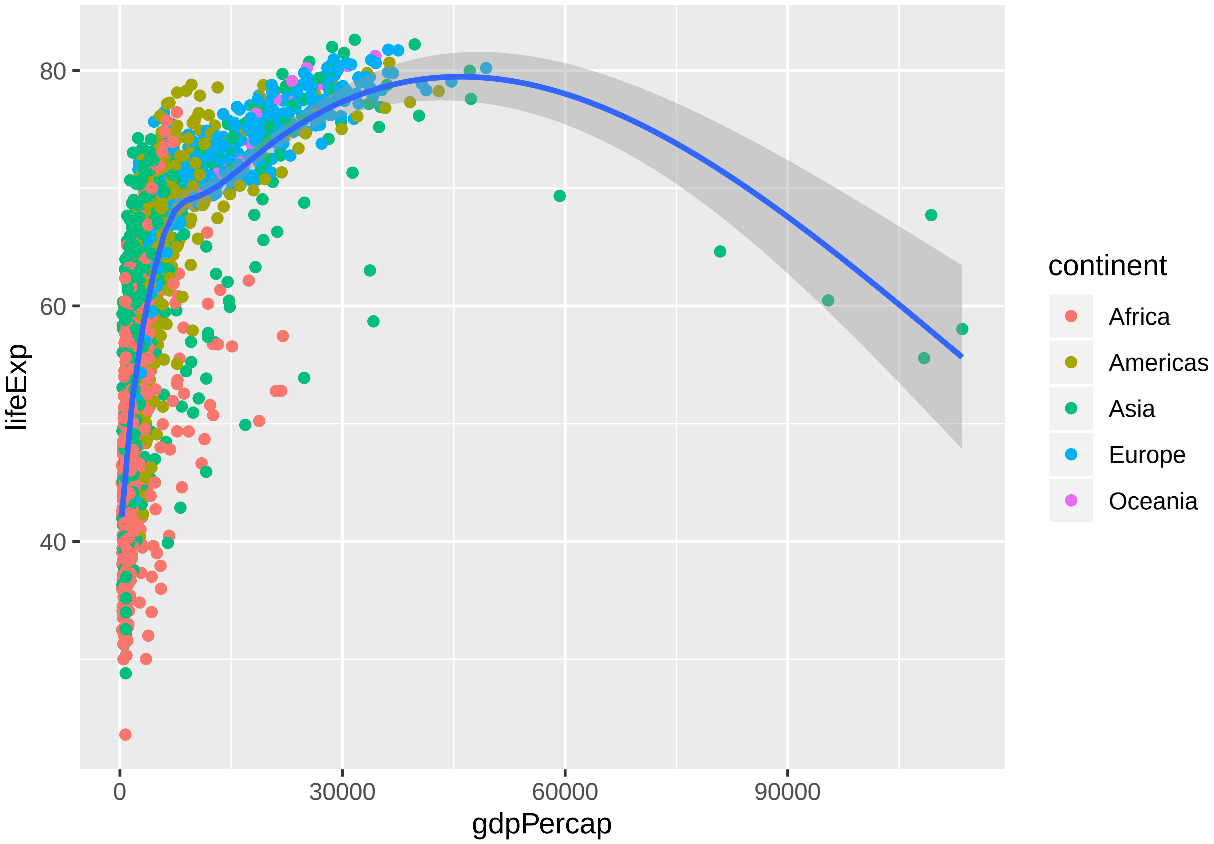
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



16. Did you notice that you got multiple regression lines (colored by continent)? That's because we set a `global` `aesthetic` of mapping `continent` to `color`. If we want just *one* regression line, we need to instead move the `color = continent` inside the `aes` of `geom_point`. This will only map `continent` to `color` for points, not for anything else.

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

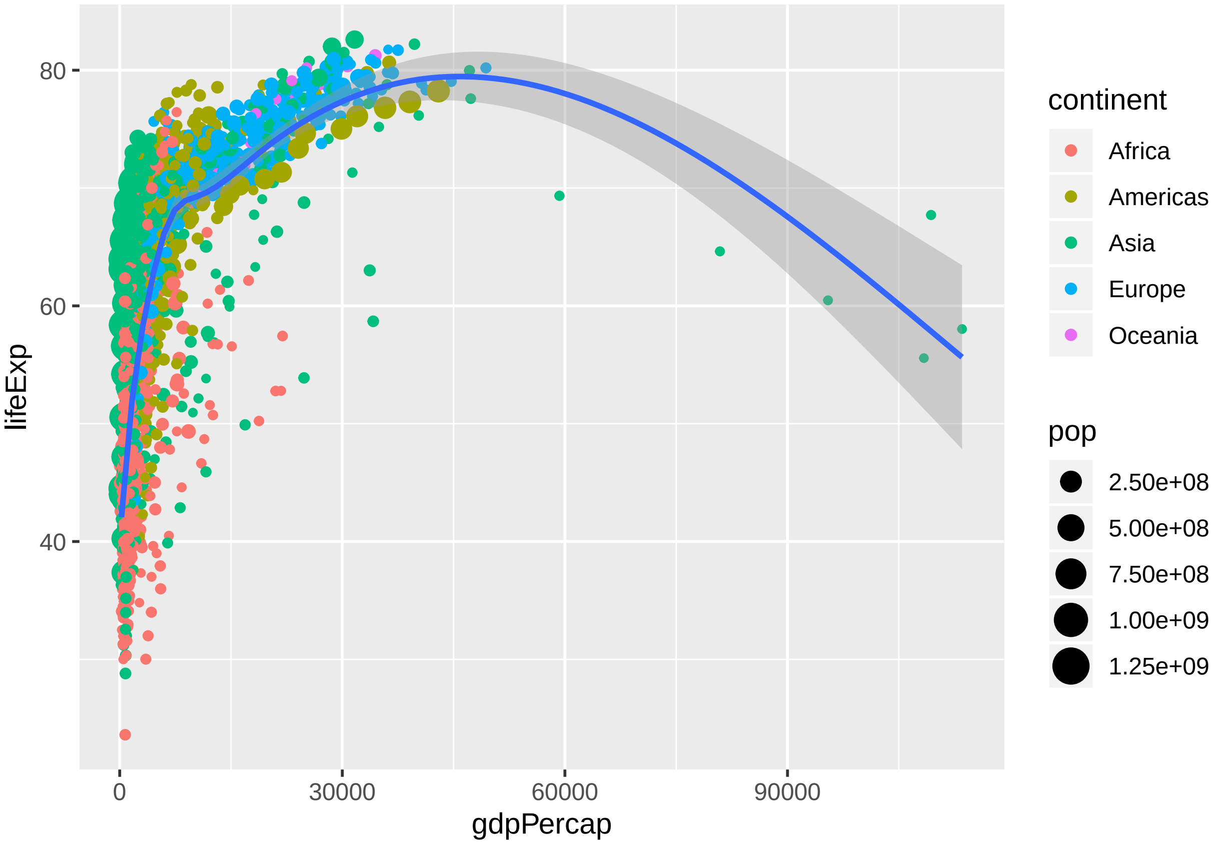
```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



17. Now add an aesthetic to your points to map `pop` to `size`.

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

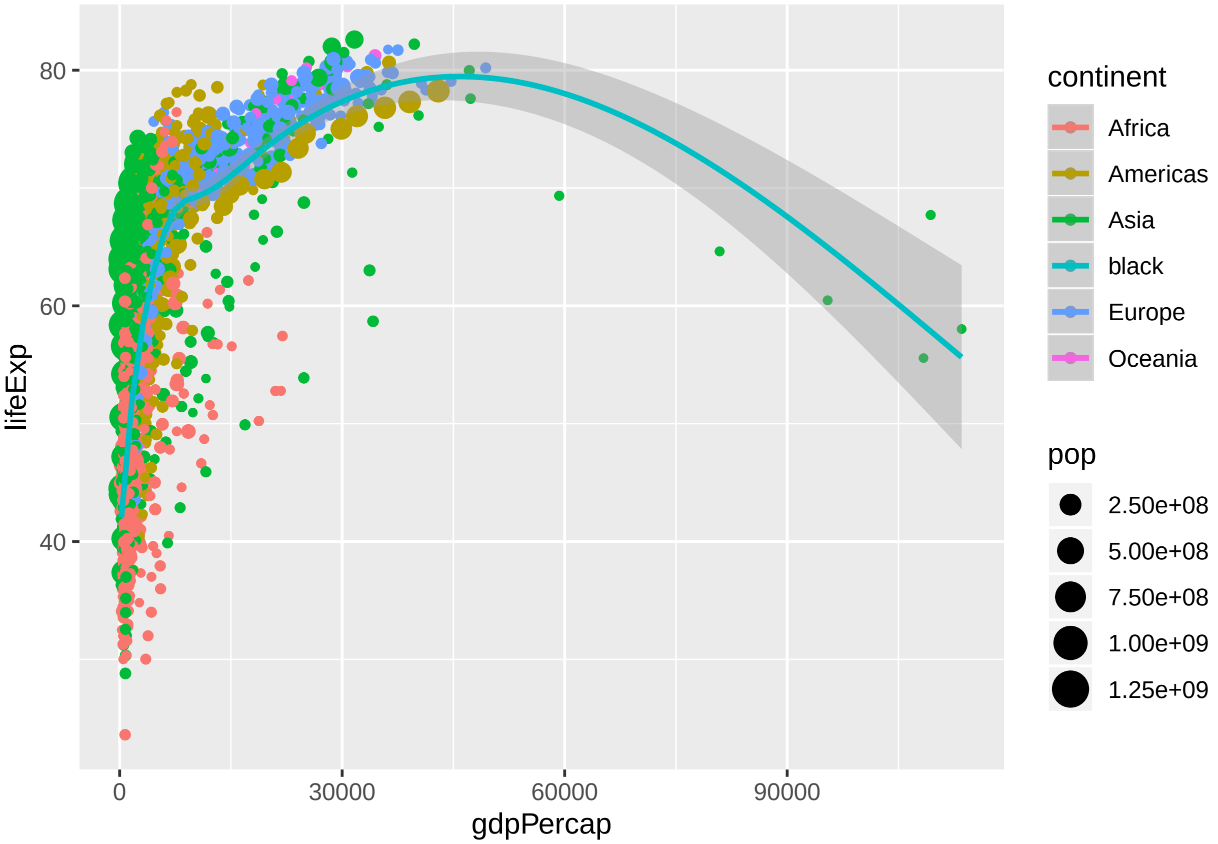
```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



18. Change the color of the regression line to `"black"`. Try first by putting this inside an `aes()` in your `geom_smooth`, and try a second time by just putting it inside `geom_smooth` without an `aes()`. What's the difference, and why?

```
ggplot(data = gapminder,
       aes(x = gdpPercap,
           y = lifeExp))+
  geom_point(aes(color = continent,
                 size = pop))+
  geom_smooth(aes(color = "black"))
```

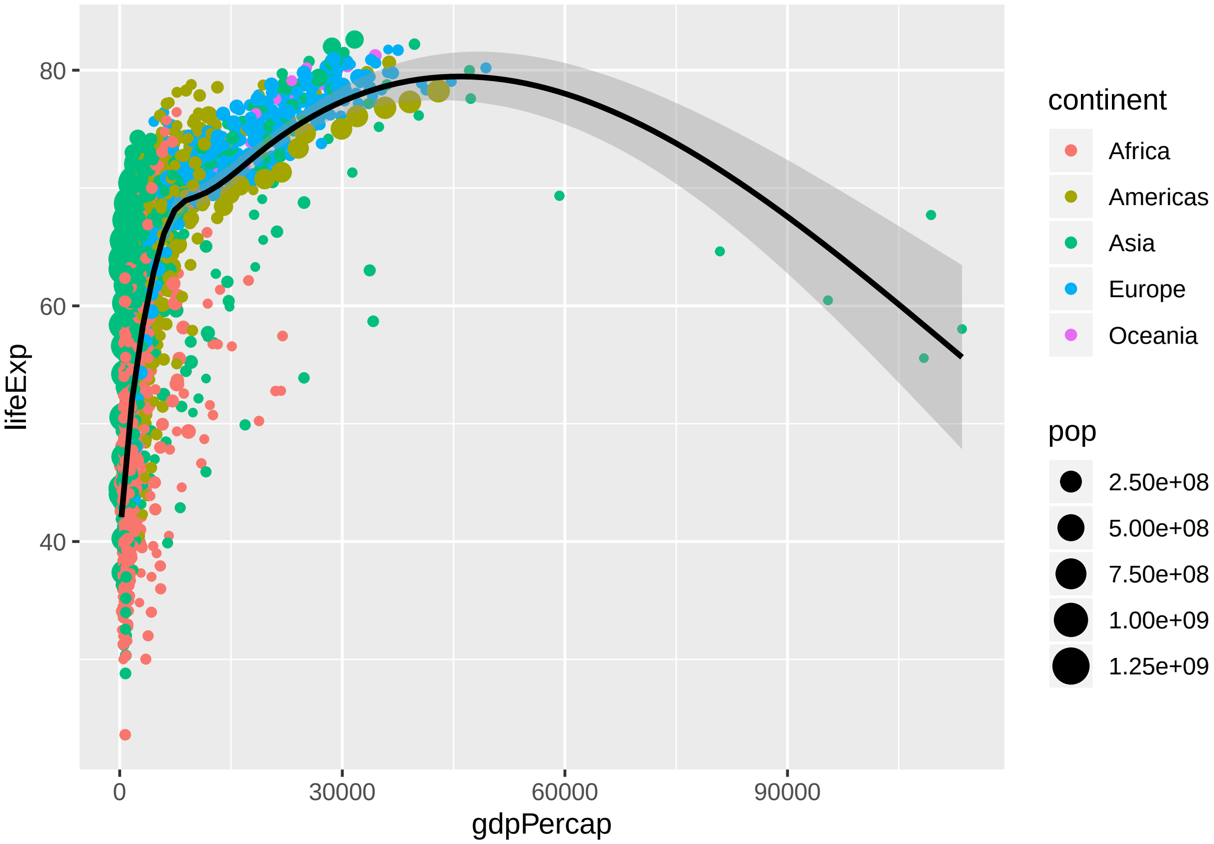
```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



```
# putting it inside aesthetics tries to map color to something
# in the da ta called "black", since R can't find "black",
# it will produce some random color

ggplot(data = gapminder,
       aes(x = gdpPercap,
           y = lifeExp))+
  geom_point(aes(color = continent,
                 size = pop))+
  geom_smooth(color = "black")
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

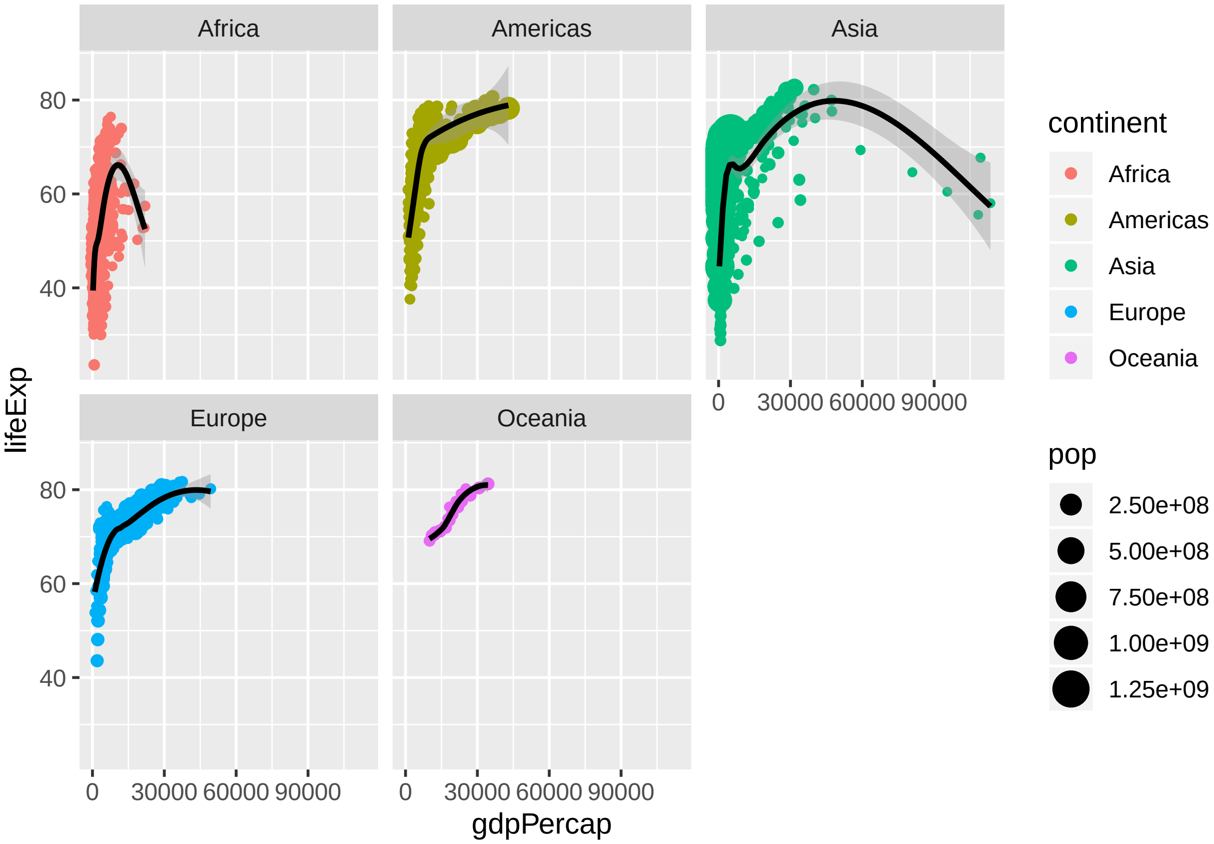


```
# putting it outside aesthetics (correctly) sets color to black
```

19. Another way to separate out continents is with `faceting`. Add `+facet_wrap(~continent)` to create subplots by `continent`.

```
ggplot(data = gapminder,
       aes(x = gdpPercap,
           y = lifeExp))+
  geom_point(aes(color = continent,
                 size = pop))+
  geom_smooth(color = "black")+
  facet_wrap(~continent)
```

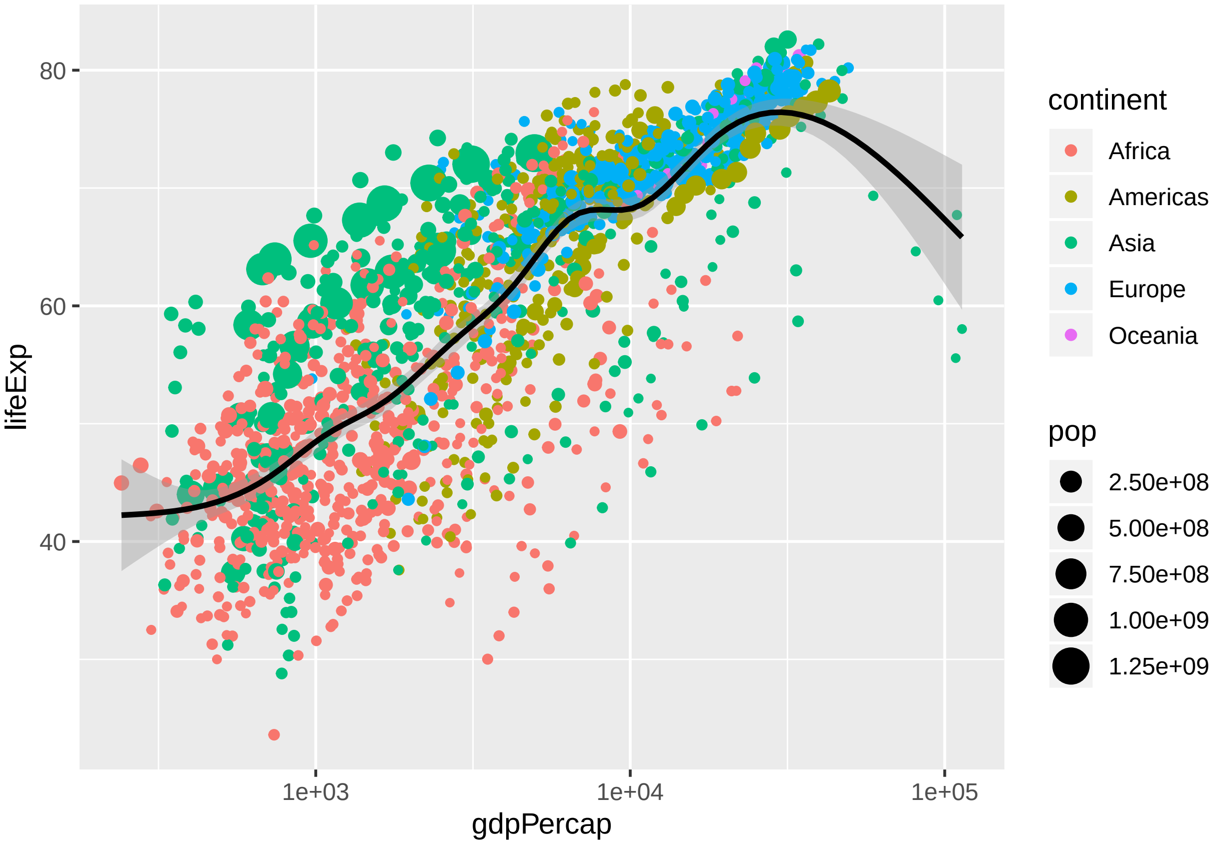
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



20. Remove the `facet` layer. The `scale` is quite annoying for the `x`-axis, a lot of points are clustered on the lower level. Let's try changing the scale by adding a layer: `+scale_x_log10()`.

```r
ggplot(data = gapminder,
       aes(x = gdpPercap,
           y = lifeExp))+
  geom_point(aes(color = continent,
                 size = pop))+
  geom_smooth(color="black")+
  scale_x_log10()
```

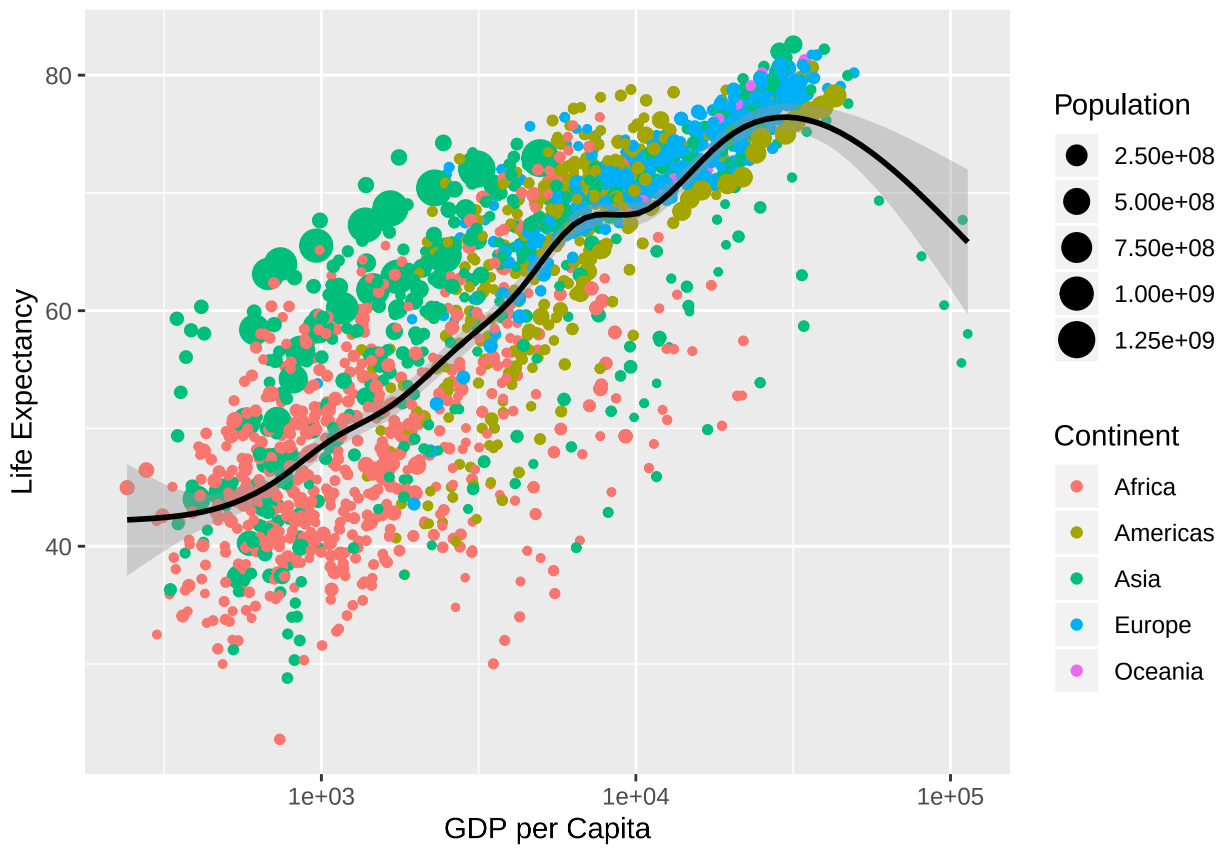
```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



21. Now let's fix the labels by adding `+labs()`. Inside `labs`, make proper axes titles for `x`, `y`, and a `title` to the plot. If you want to change the name of the legends (continent color), add one for `color` and `size`.

```
ggplot(data = gapminder,
       aes(x = gdpPercap,
           y = lifeExp))+
  geom_point(aes(color = continent,
                 size = pop))+
  geom_smooth(color="black")+
  scale_x_log10()+
  labs(x = "GDP per Capita",
       y = "Life Expectancy",
       color = "Continent",
       size = "Population")
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



22. Now let's try subsetting by looking only at North America. Take the `gapminder` dataframe and subset it to only look at `continent=="Americas"`). Assign this to a new dataframe object (call it something like `america`.) Now, use *this* as your `data`, and redo the graph from question 17. (You might want to take a look at your new dataframe to make sure it worked first!)

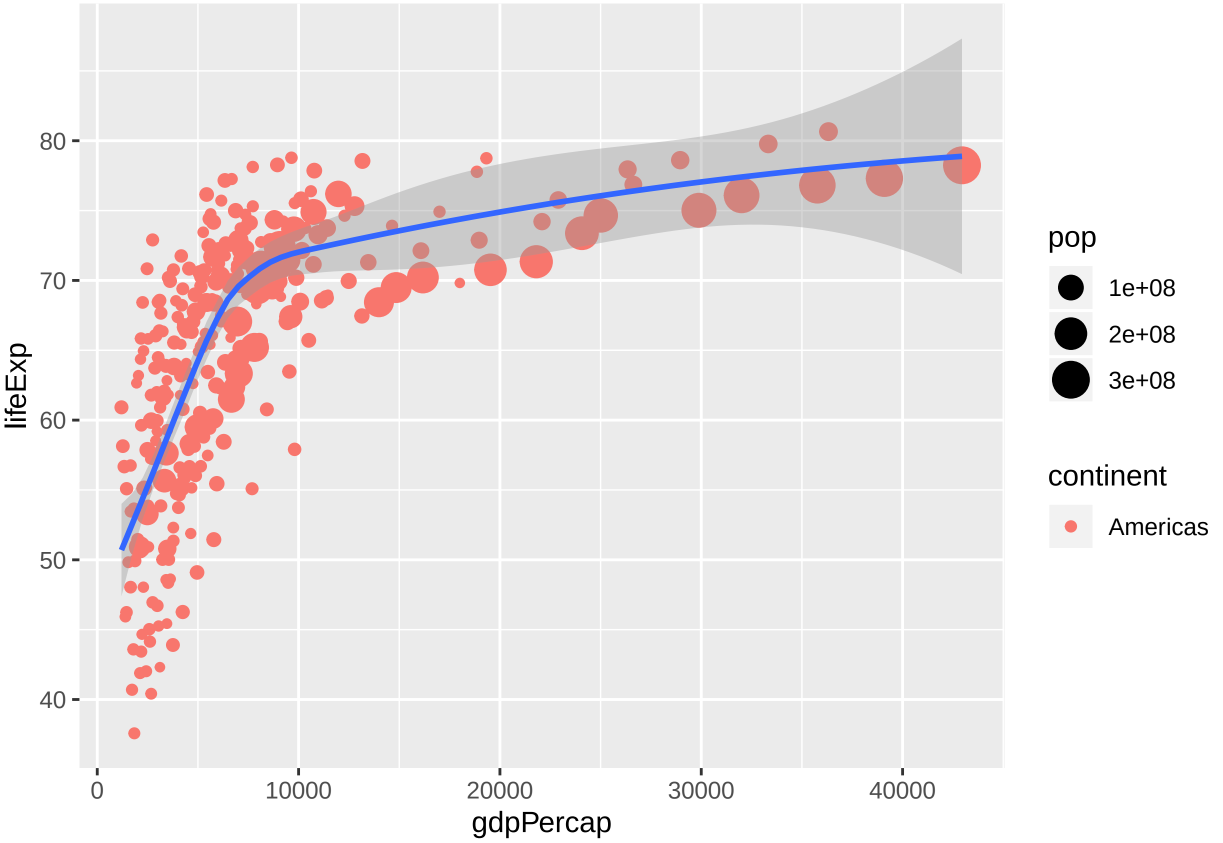
```
america<-gapminder[gapminder$continent=="Americas",]

# verify this worked
america
```

```
## # A tibble: 300 x 6
##    country   continent  year lifeExp      pop gdpPercap
##    <fct>     <fct>     <int>   <dbl>    <int>     <dbl>
##  1 Argentina Americas   1952    62.5 17876956     5911.
##  2 Argentina Americas   1957    64.4 19610538     6857.
##  3 Argentina Americas   1962    65.1 21283783     7133.
##  4 Argentina Americas   1967    65.6 22934225     8053.
##  5 Argentina Americas   1972    67.1 24779799     9443.
##  6 Argentina Americas   1977    68.5 26983828    10079.
##  7 Argentina Americas   1982    69.9 29341374     8998.
##  8 Argentina Americas   1987    70.8 31620918     9140.
##  9 Argentina Americas   1992    71.9 33958947     9308.
## 10 Argentina Americas   1997    73.3 36203463    10967.
## # ... with 290 more rows
```

```
ggplot(data = america,
       aes(x = gdpPercap,
           y = lifeExp))+
  geom_point(aes(color = continent,
                 size = pop))+
  geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



23. Try this again for the *whole* world, but just for observations in the year 2002.

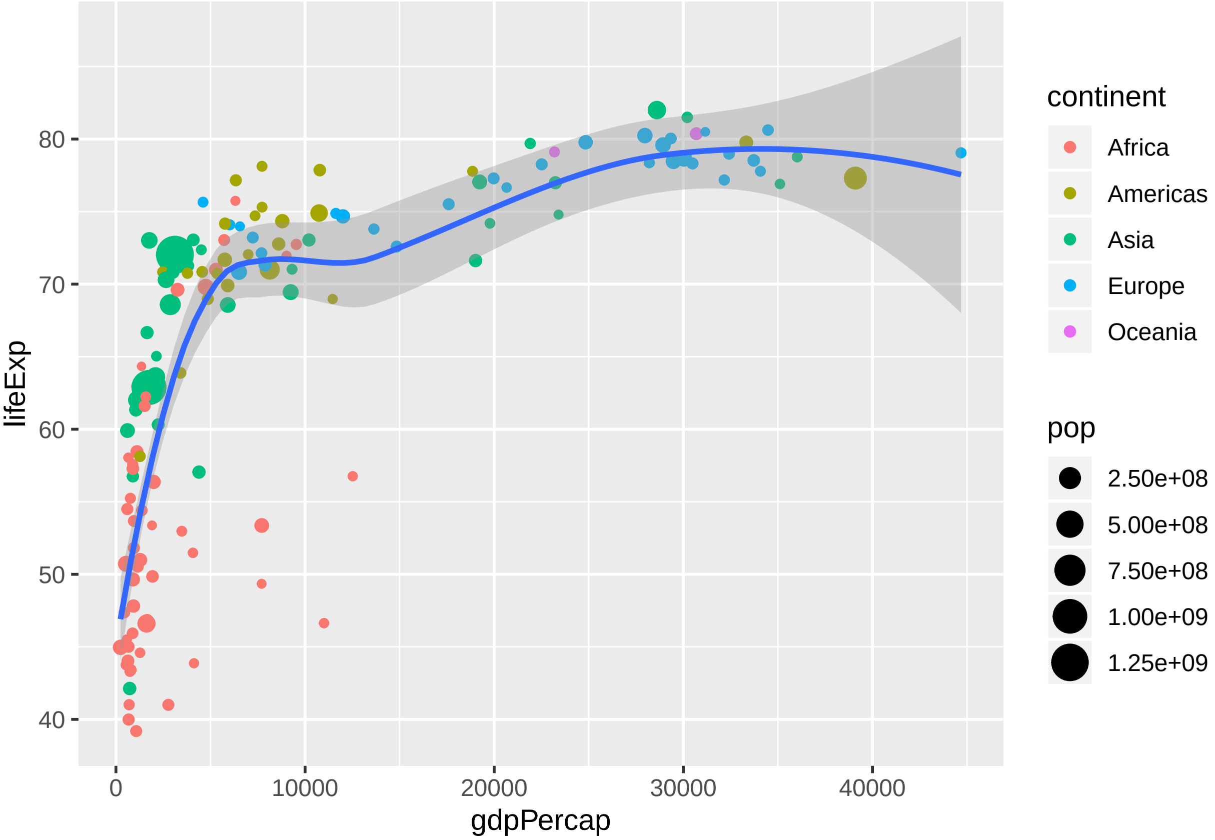
```
gap_2002<-gapminder[gapminder$year==2002,]

# verify this worked
gap_2002
```

```
## # A tibble: 142 x 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   Oceania    2002    80.4  19546792    30688.
##  7 Austria     Europe     2002    79.0   8148312    32418.
##  8 Bahrain     Asia       2002    74.8    656397    23404.
##  9 Bangladesh  Asia       2002    62.0 135656790     1136.
## 10 Belgium     Europe     2002    78.3  10311970    30486.
## # ... with 132 more rows
```

```
ggplot(data = gap_2002,
       aes(x = gdpPercap,
           y = lifeExp))+
  geom_point(aes(color = continent,
                 size = pop))+
  geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```


lifeExp
gdpPercap
continent
Africa
Americas
Asia
Europe
Oceania
pop
2.50e+08
5.00e+08
7.50e+08
1.00e+09
1.25e+09