

Class 5 Data Visualization with ggplot2

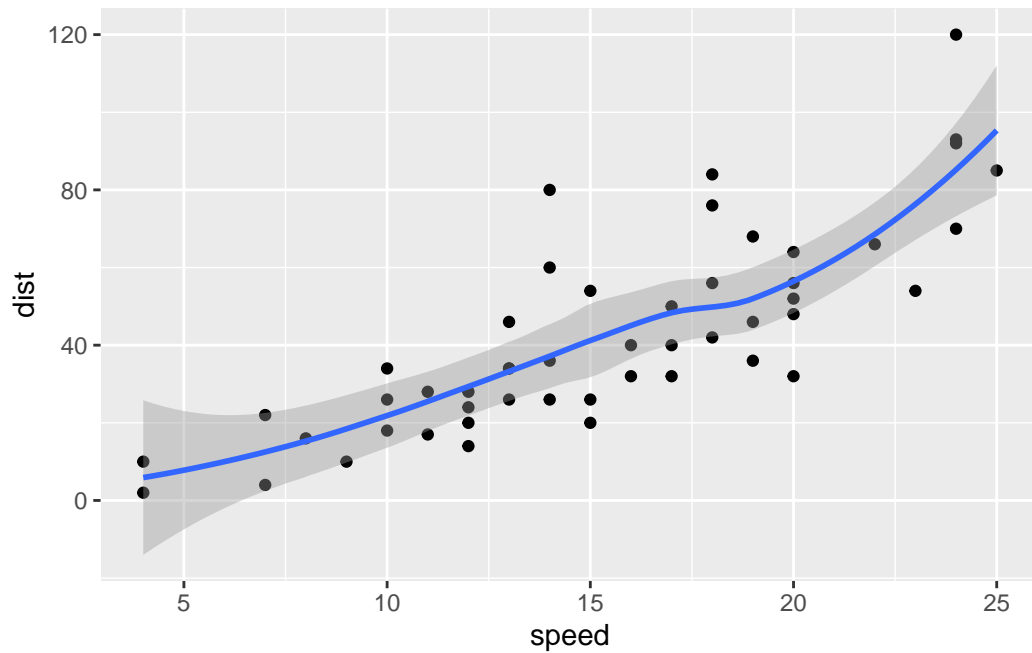
Jason (PID: A15796973)

Questions

- Q1. For which phases is data visualization important in our scientific workflows? All of the above
- Q2. True or False? The ggplot2 package comes already installed with R? FALSE
- Q3. Which plot types are typically NOT used to compare distributions of numeric variables? Network graphs
- Q4. Which statement about data visualization with ggplot2 is incorrect? ggplot2 is the only way to create plots in R
- Q5. Which geometric layer should be used to create scatter plots in ggplot2? `geom_point()`
- Q6. In your own RStudio can you add a trend line layer to help show the relationship between the plot variables with the `geom_smooth()` function?

```
library(ggplot2)
ggplot(cars) +
  aes(x=speed, y=dist) +
  geom_point() +
  geom_smooth()
```

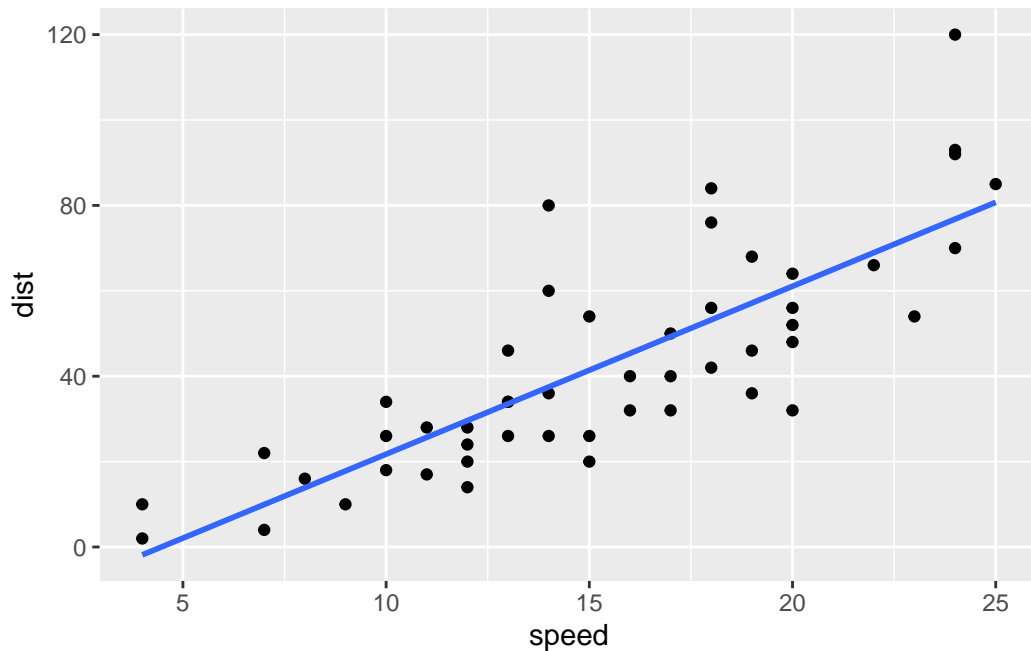
``geom_smooth()`` using method = 'loess' and formula = 'y ~ x'



Q7. Argue with `geom_smooth()` to add a straight line from a linear model without the shaded standard error region?

```
ggplot(cars) +  
  aes(x=speed, y=dist) +  
  geom_point() +  
  geom_smooth(method="lm", se=FALSE)
```

``geom_smooth()`` using formula = `'y ~ x'`



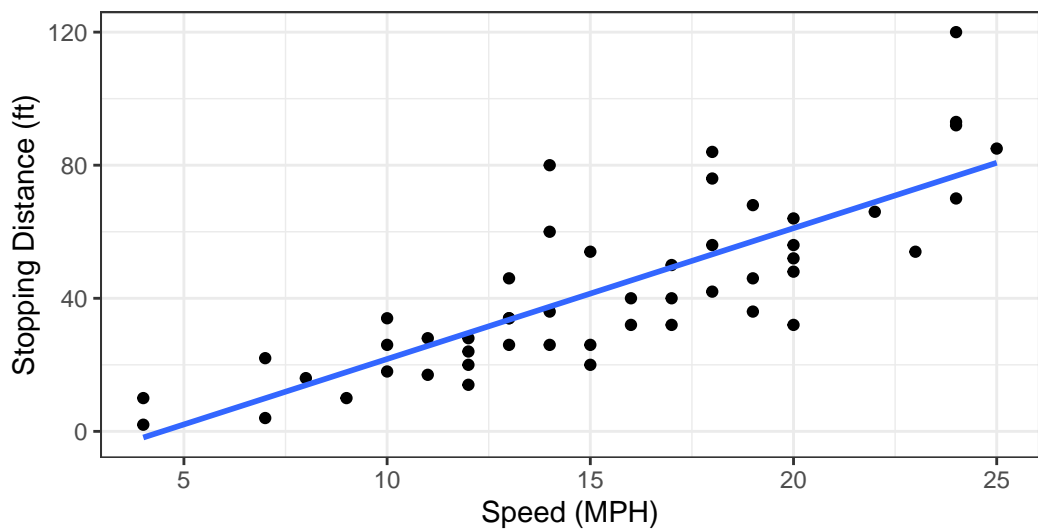
Q8. Can you finish this plot by adding various label annotations with the `labs()` function and changing the plot look to a more conservative “black & white” theme by adding the `theme_bw()` function:

```
ggplot(cars) +
  aes(x=speed, y=dist) +
  geom_point() +
  geom_smooth(method="lm", se=FALSE) +
  labs(title="Speed and Stopping Distances of Cars",
        x="Speed (MPH)",
        y="Stopping Distance (ft)",
        subtitle = "This subtitle is very informative super informative I have never seen a",
        caption="Dataset: 'cars'") +
  theme_bw()
```

``geom_smooth()`` using formula = 'y ~ x'

Speed and Stopping Distances of Cars

This subtitle is very informative super informative I have never seen anythi



Dataset: 'cars'

For questions 9-12

```
url <- "https://bioboot.github.io/bimm143_S20/class-material/up_down_expression.txt"
genes <- read.delim(url)
head(genes)
```

	Gene	Condition1	Condition2	State
1	A4GNT	-3.6808610	-3.4401355	unchanging
2	AAAS	4.5479580	4.3864126	unchanging
3	AASDH	3.7190695	3.4787276	unchanging
4	AATF	5.0784720	5.0151916	unchanging
5	AATK	0.4711421	0.5598642	unchanging
6	AB015752.4	-3.6808610	-3.5921390	unchanging

```
nrow(genes)
```

```
[1] 5196
```

```
colnames(genes)
```

```
[1] "Gene"          "Condition1" "Condition2" "State"
```

```
ncol(genes)
```

```
[1] 4
```

```
table(genes["State"])
```

```
State
down  unchanging      up
   72      4997     127
```

```
round( table(genes["State"])/nrow(genes) * 100, 2 )
```

```
State
down  unchanging      up
 1.39      96.17     2.44
```

Q9. Use the `nrow()` function to find out how many genes are in this dataset. What is your answer? 5196

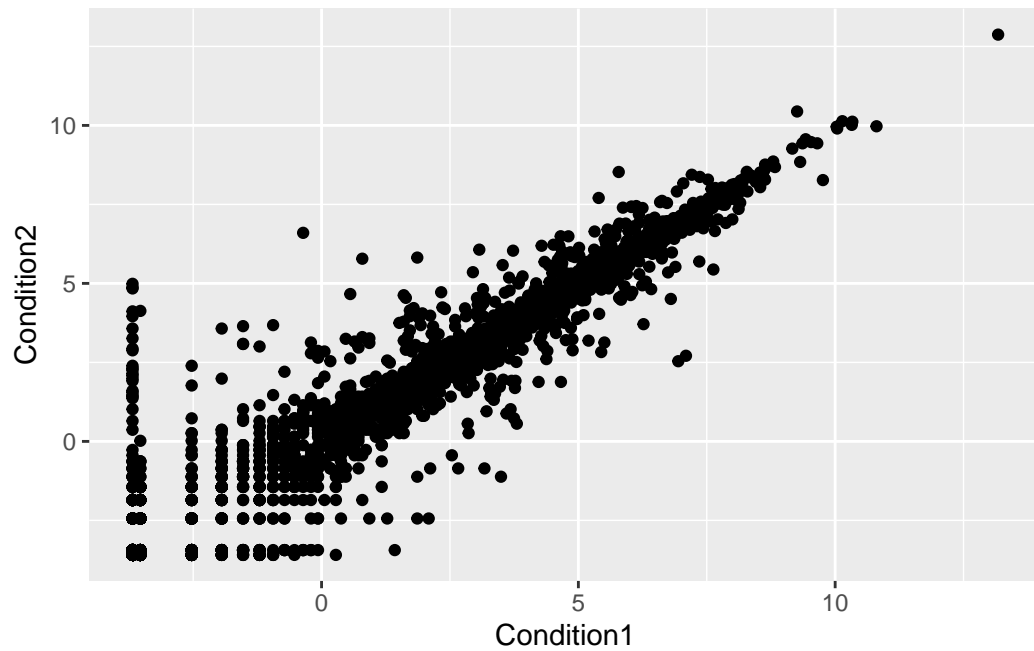
Q10. Use the `colnames()` function and the `ncol()` function on the `genes` data frame to find out what the column names are (we will need these later) and how many columns there are. How many columns did you find? 4

Q11. Use the `table()` function on the `State` column of this `data.frame` to find out how many 'up' regulated genes there are. What is your answer? 127

Q12. Using your values above and 2 significant figures. What fraction of total genes is up-regulated in this dataset? 2.44

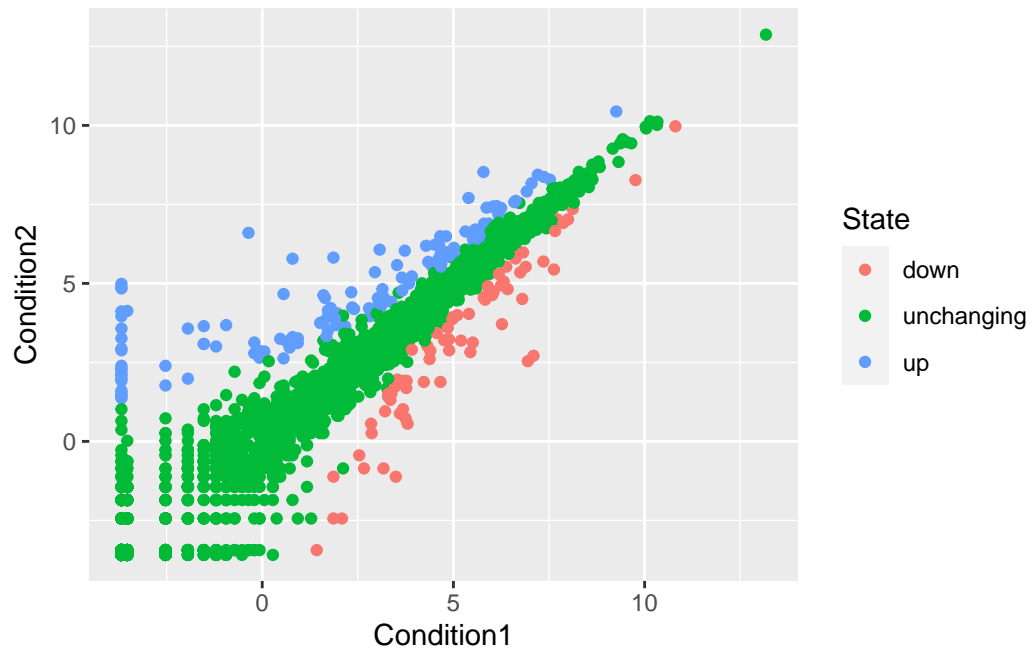
Q13. Complete the code below to produce the following plot

```
ggplot(genes) +
  aes(x=Condition1, y=Condition2) +
  geom_point()
```

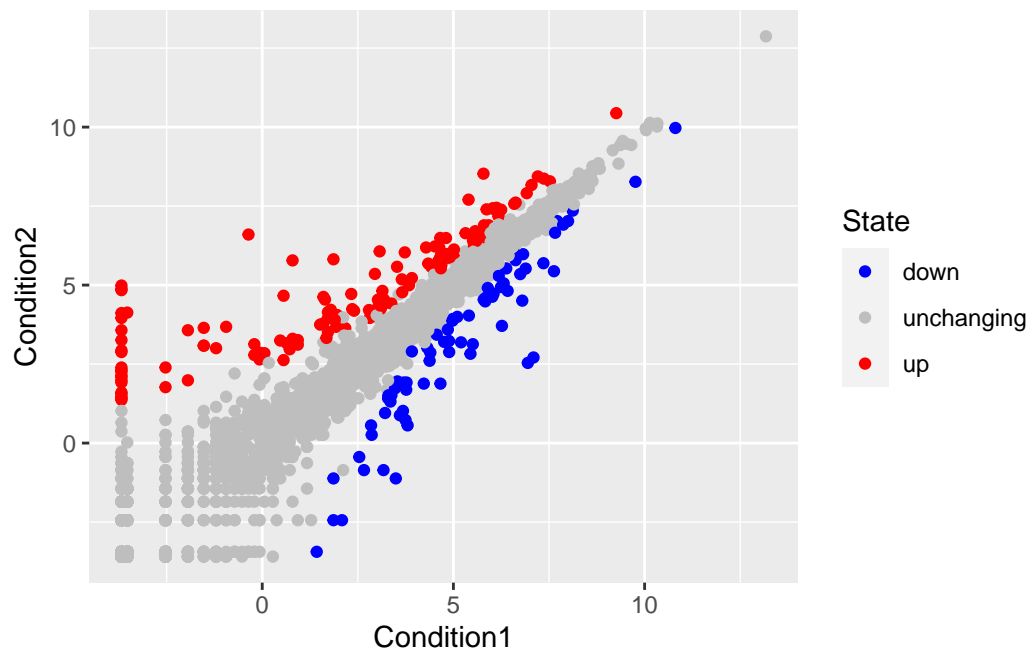


Post question 13 graphs followed by question 14

```
p <- ggplot(genes) +  
  aes(x=Condition1, y=Condition2, col=State) +  
  geom_point()  
p
```

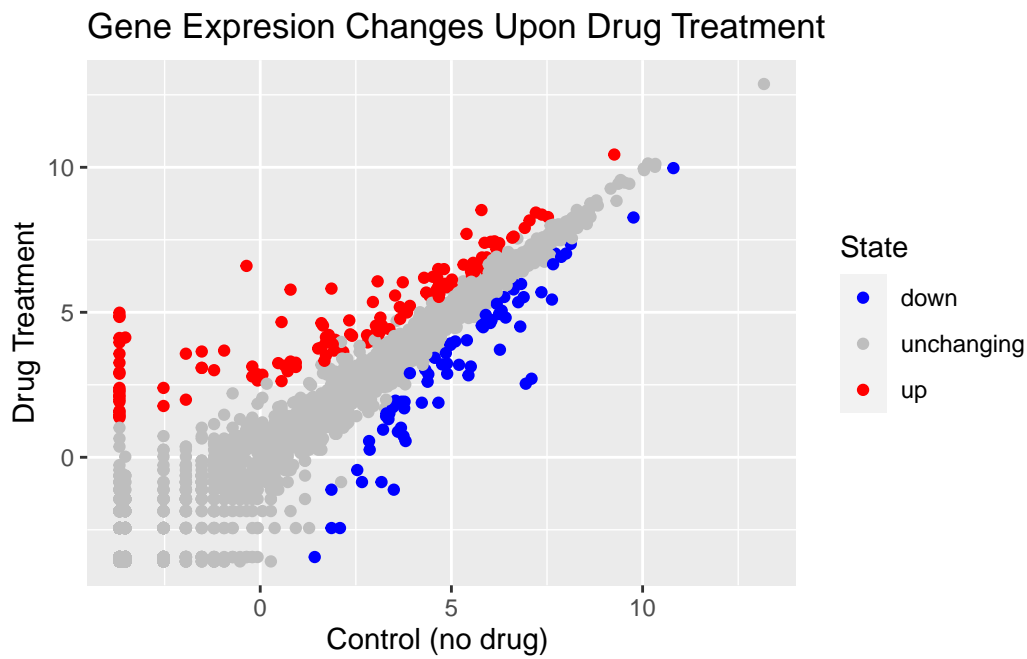


```
p + scale_colour_manual( values=c("blue","gray","red") )
```



Q14. Nice, now add some plot annotations to the p object with the labs() function so your plot looks like the following:

```
p + labs(title="Gene Expression Changes Upon Drug Treatment",
         x="Control (no drug) ",
         y="Drug Treatment") +
  scale_colour_manual( values=c("blue","gray","red") )
```



Section 7: Going Further

```
##install.packages("gapminder")
library(gapminder)

##install.packages("dplyr")
library(dplyr)
```

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':


```
filter, lag
```

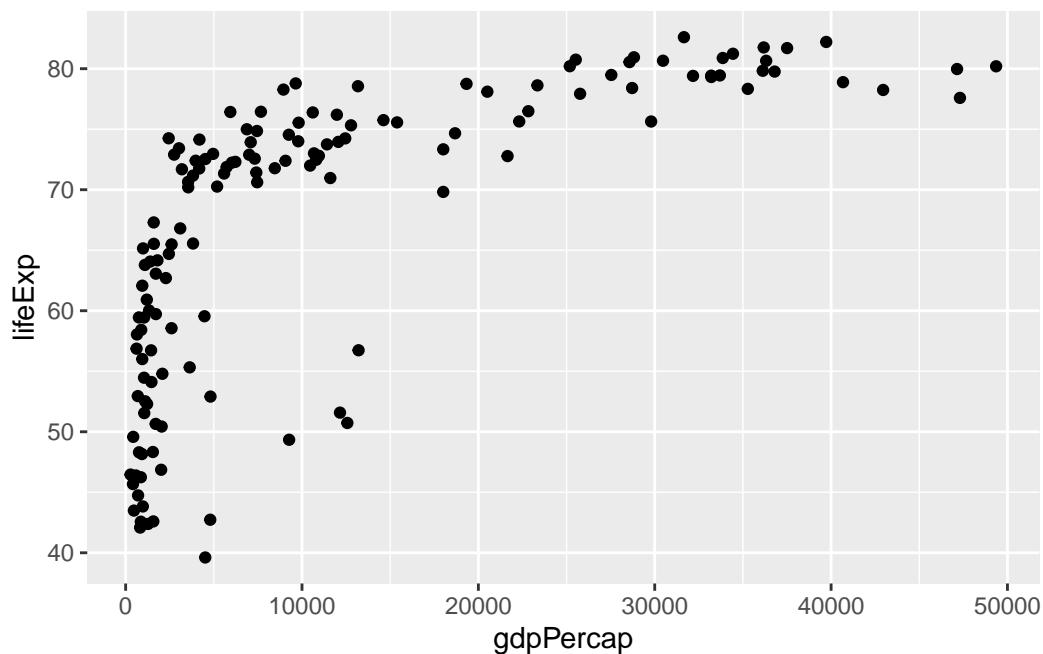
The following objects are masked from 'package:base':

```
intersect, setdiff, setequal, union
```

```
gapminder_2007 <- gapminder %>% filter(year==2007)
```

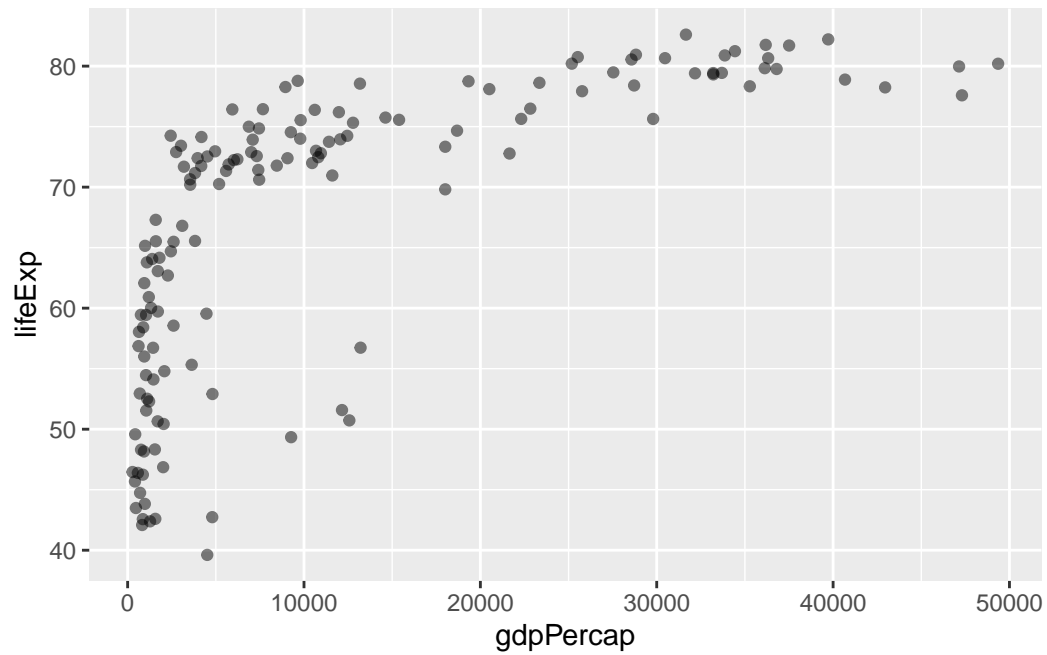
Q15. Complete the code below to produce a first basic scatter plot of this gapminder_2007 dataset:

```
ggplot(gapminder_2007) +  
  aes(x=gdpPercap, y=lifeExp) +  
  geom_point()
```

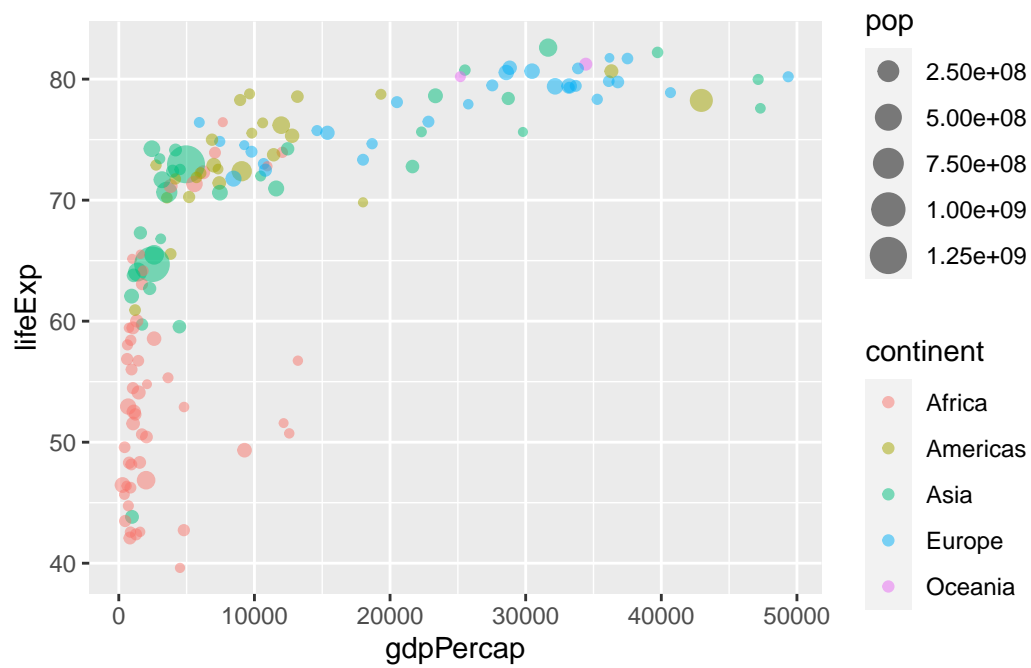


****Post Question 15 Stuff****

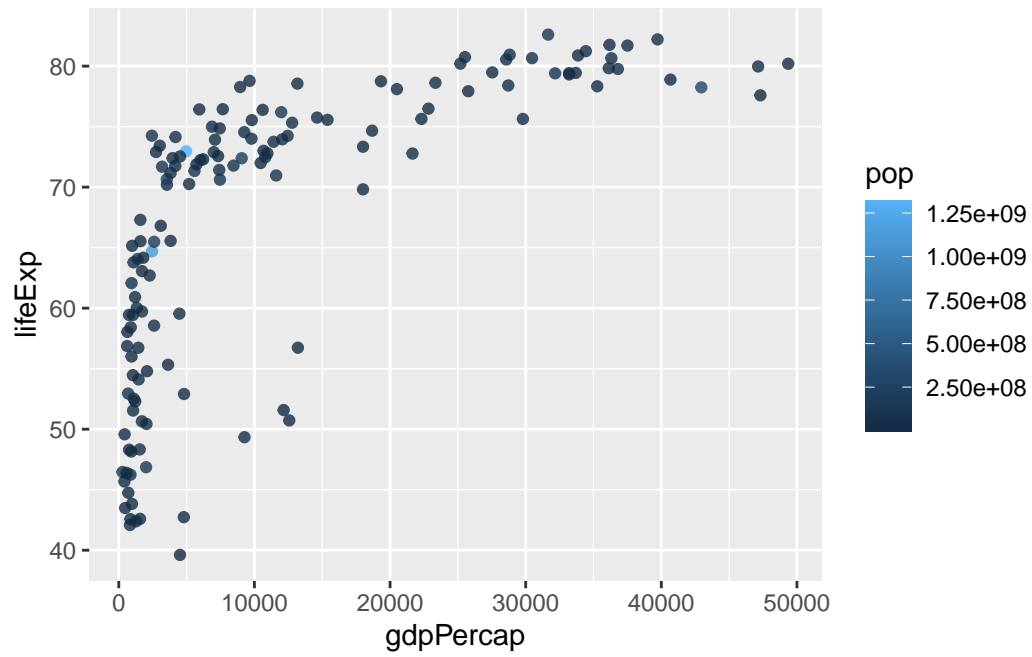
```
ggplot(gapminder_2007) +  
  aes(x=gdpPercap, y=lifeExp) +  
  geom_point(alpha=0.5)
```



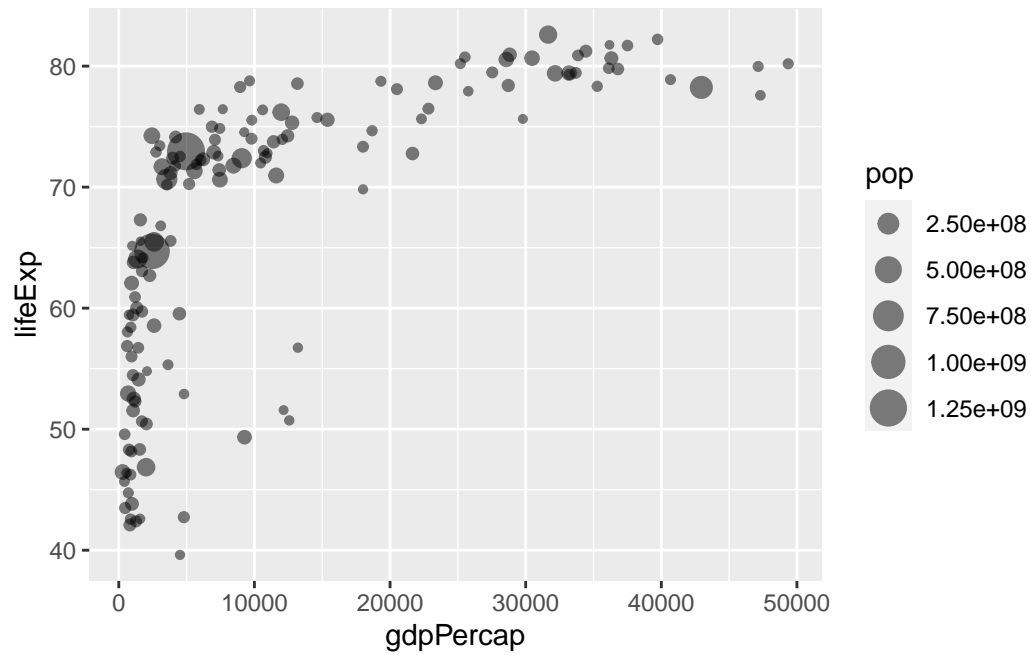
```
ggplot(gapminder_2007) +  
  aes(x=gdpPerCap, y=lifeExp, color=continent, size=pop) +  
  geom_point(alpha=0.5)
```



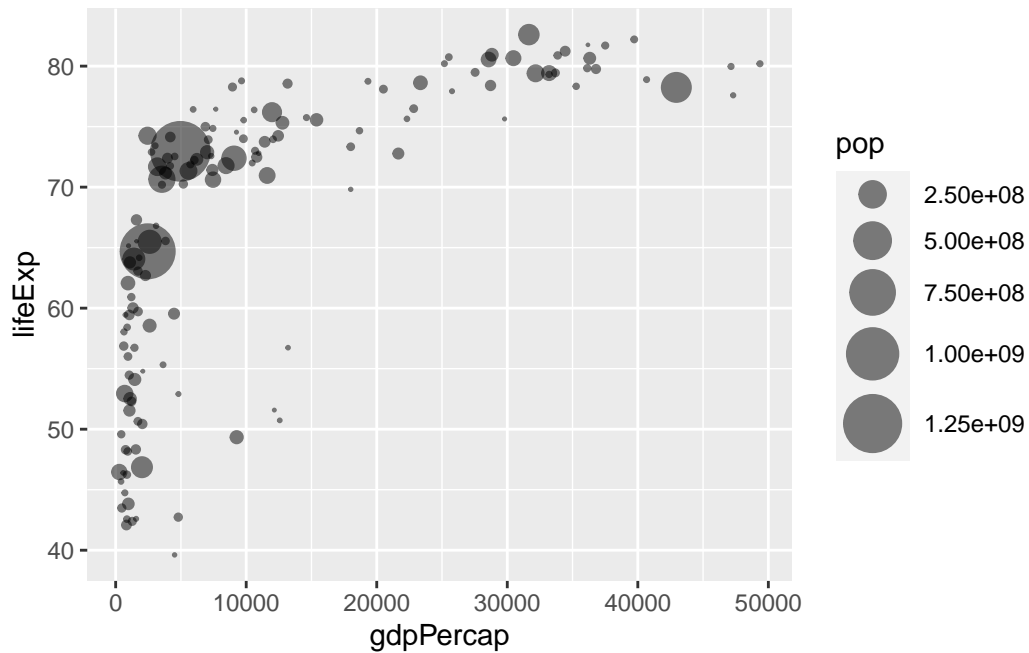
```
ggplot(gapminder_2007) +  
  aes(x = gdpPercap, y = lifeExp, color = pop) +  
  geom_point(alpha=0.8)
```



```
ggplot(gapminder_2007) +  
  aes(x = gdpPercap, y = lifeExp, size = pop) +  
  geom_point(alpha=0.5)
```



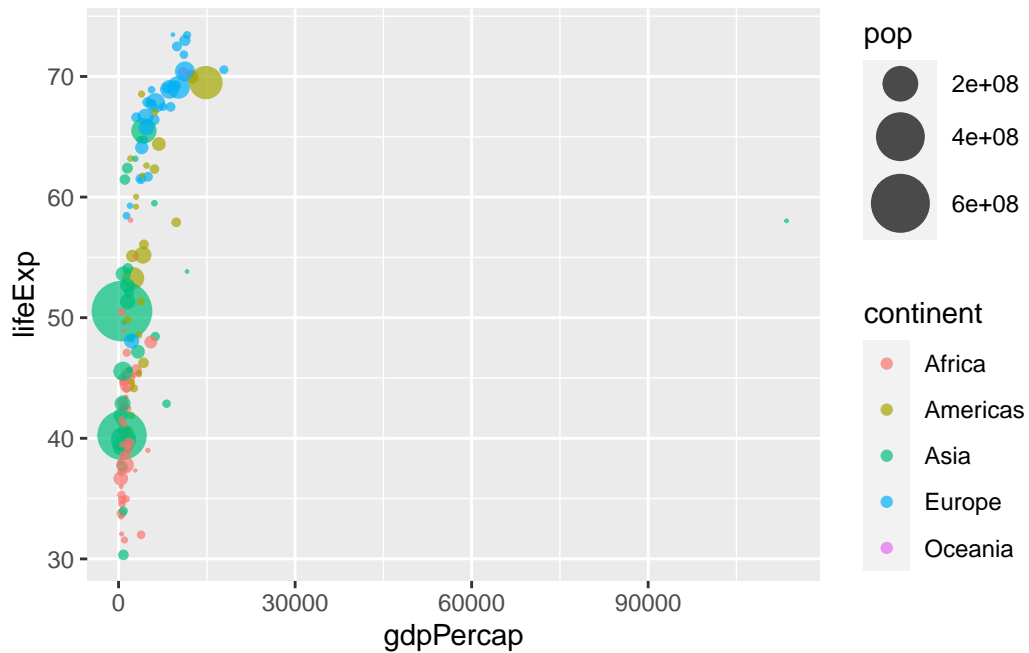
```
ggplot(gapminder_2007) +  
  geom_point(aes(x = gdpPercap, y = lifeExp,  
                 size = pop), alpha=0.5) +  
  scale_size_area(max_size = 10)
```



Q16. Can you adapt the code you have learned thus far to reproduce our gapminder scatter plot for the year 1957? What do you notice about this plot is it easy to compare with the one for 2007?

```
gapminder_1957 <- gapminder %>% filter(year==1957)

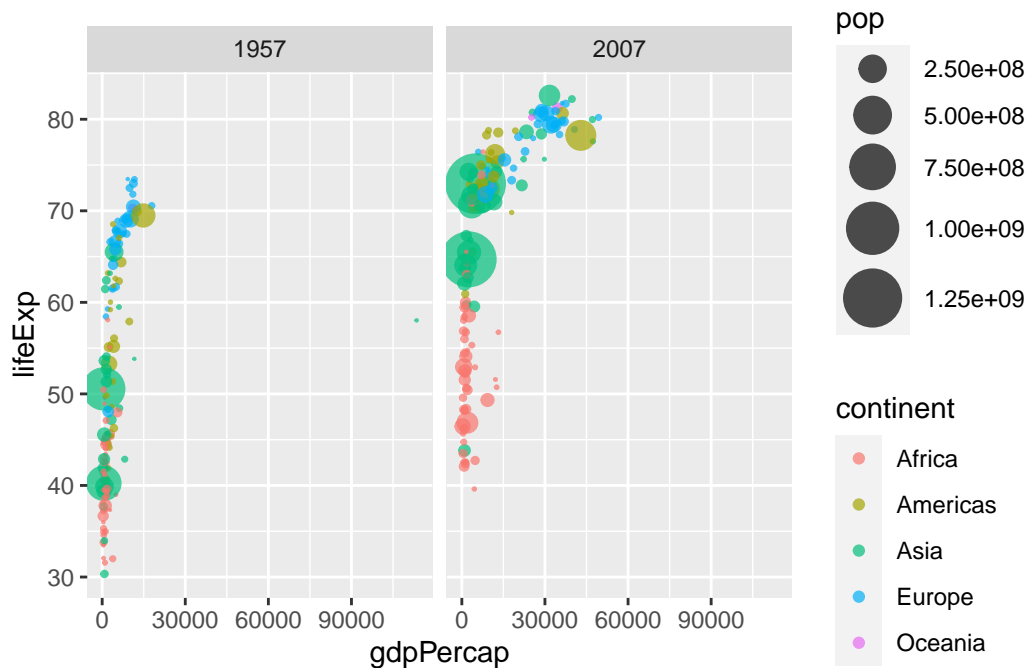
ggplot(gapminder_1957) +
  aes(x = gdpPercap, y = lifeExp, color=continent,
      size = pop) +
  geom_point(alpha=0.7) +
  scale_size_area(max_size = 10)
```



Q17. Do the same steps above but include 1957 and 2007 in your input dataset for ggplot(). You should now include the layer `facet_wrap(~year)` to produce the following plot:

```
gapminder_1957 <- gapminder %>% filter(year==1957 | year==2007)

ggplot(gapminder_1957) +
  geom_point(aes(x = gdpPercap, y = lifeExp, color=continent,
                 size = pop), alpha=0.7) +
  scale_size_area(max_size = 10) +
  facet_wrap(~year)
```



Using GGPLOT

The ggplot2 package needs to be installed as it does not come with R “out of the box”.

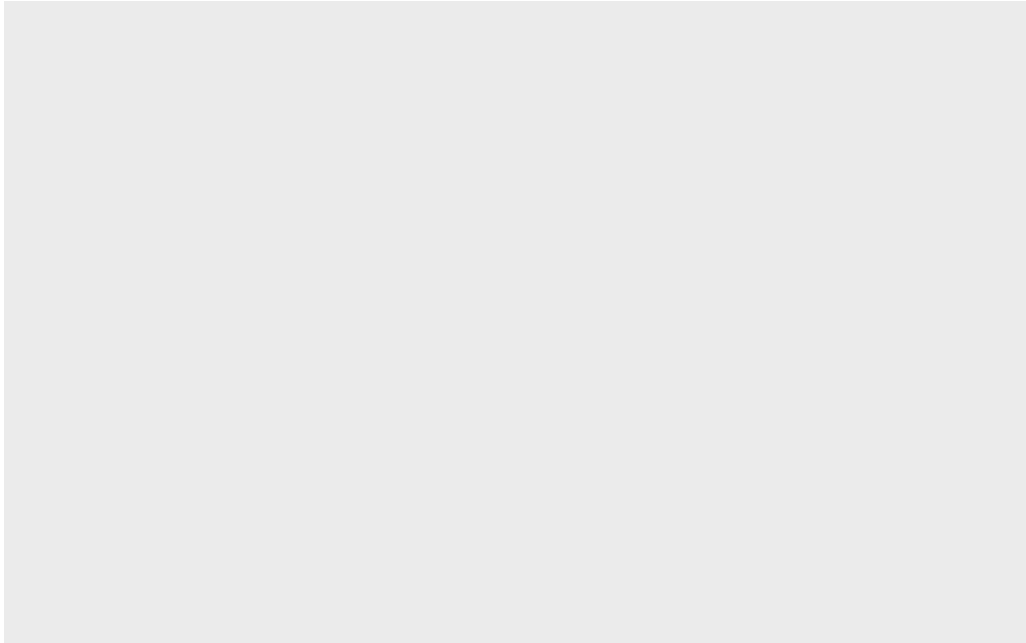
We use the `install.packages()` function to do this.

```
head(cars)
```

```
  speed dist
1     4    2
2     4   10
3     7    4
4     7   22
5     8   16
6     9   10
```

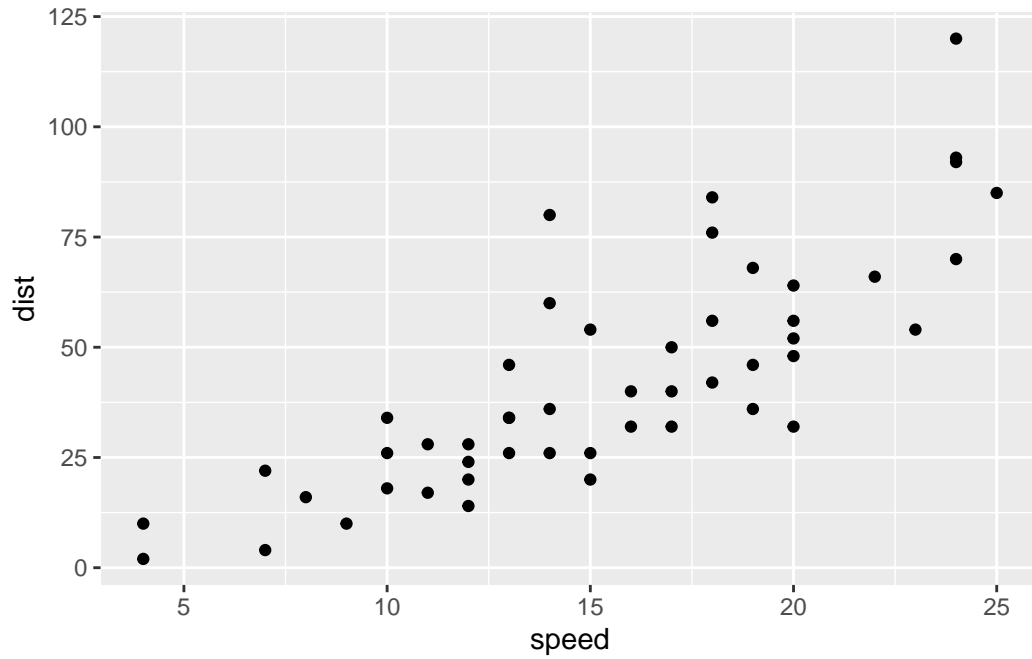
To use ggplot I need to load it up before I can call any of the functions in the package. I need to do this with the `library()` function.

```
library(ggplot2)
ggplot()
```

All ggplot figures have at least 3 things: - data (the stuff we want to plot) - aesthetics mapping (aes vales) -geoms

```
ggplot(cars) +  
  aes(x=speed, y=dist) +  
  geom_point()
```



ggplot is not the only graphing system in R there are lots of others. There is even “base R” graphics.

```
plot(cars)
```

