

# class05.R

chloewelch

2021-10-13

```
#Class 05: Data Visualization
```

```
#Today we are going to use ggplot2 package
```

```
#First we need to load the package!
```

```
#install.packages("ggplot2") - this is a one-time only installation!
```

```
library(ggplot2)
```

```
#We will use this in-built "cars" dataset first
```

```
head(cars)
```

```
##   speed dist
```

```
## 1     4    2
```

```
## 2     4   10
```

```
## 3     7    4
```

```
## 4     7   22
```

```
## 5     8   16
```

```
## 6     9   10
```

```
#All ggplots have at least 3 layers;
```

```
#data + aes + geoms
```

```
ggplot(data=cars) +
```

```
  aes(x=speed, y=dist) +
```

```
  geom_point() +
```

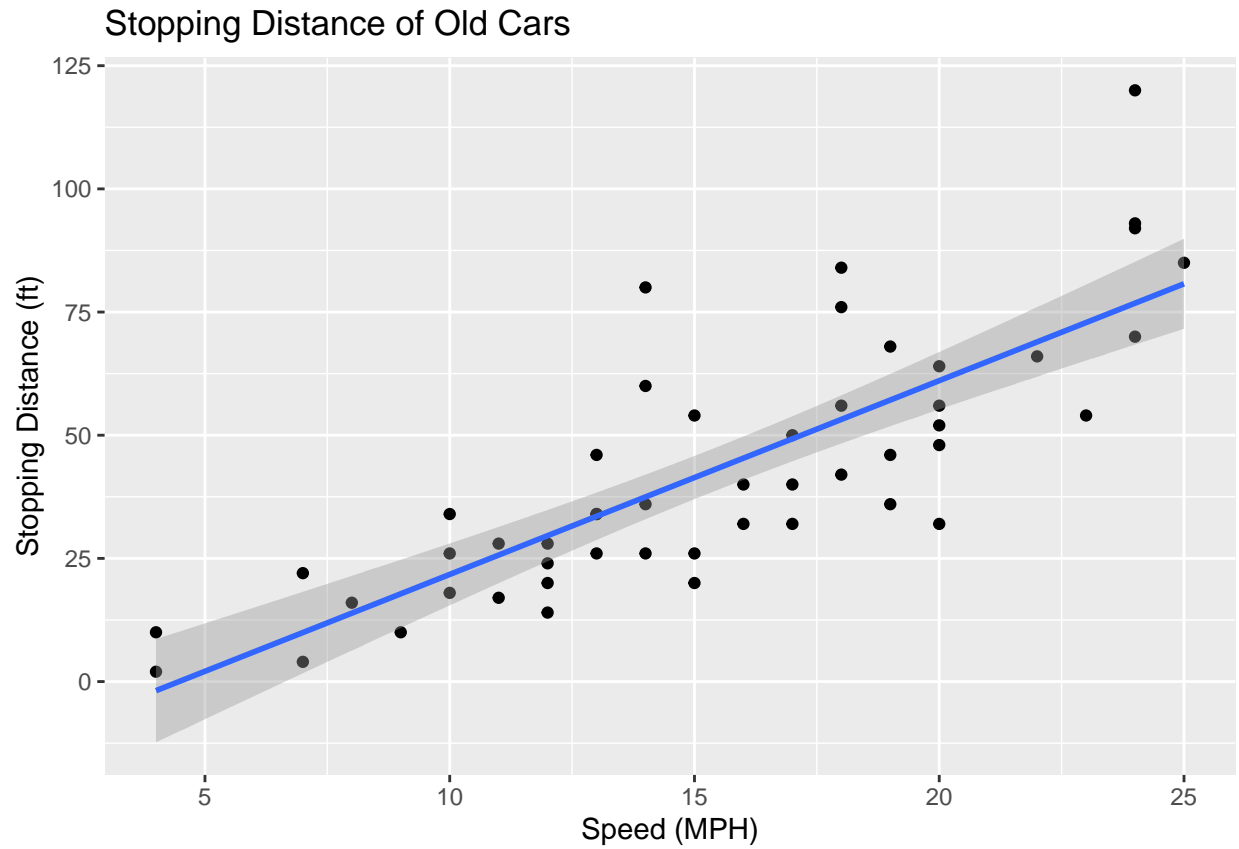
```
  geom_smooth(method="lm") +
```

```
  labs(title="Stopping Distance of Old Cars",
```

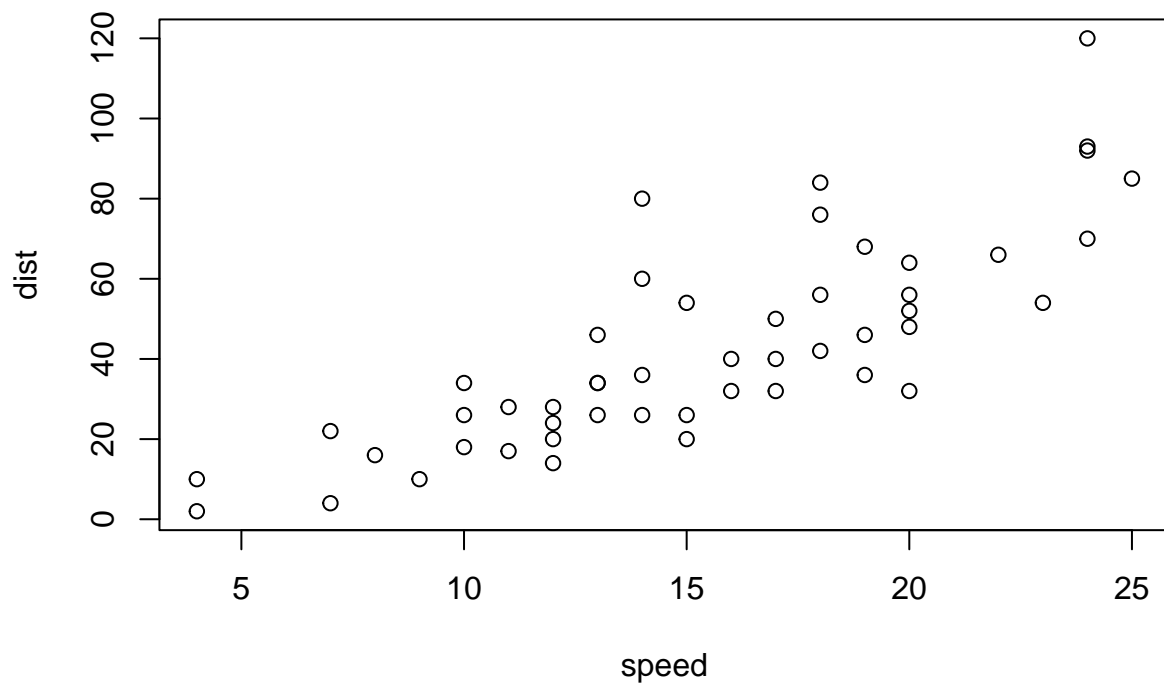
```
        x="Speed (MPH)",
```

```
        y="Stopping Distance (ft)")
```

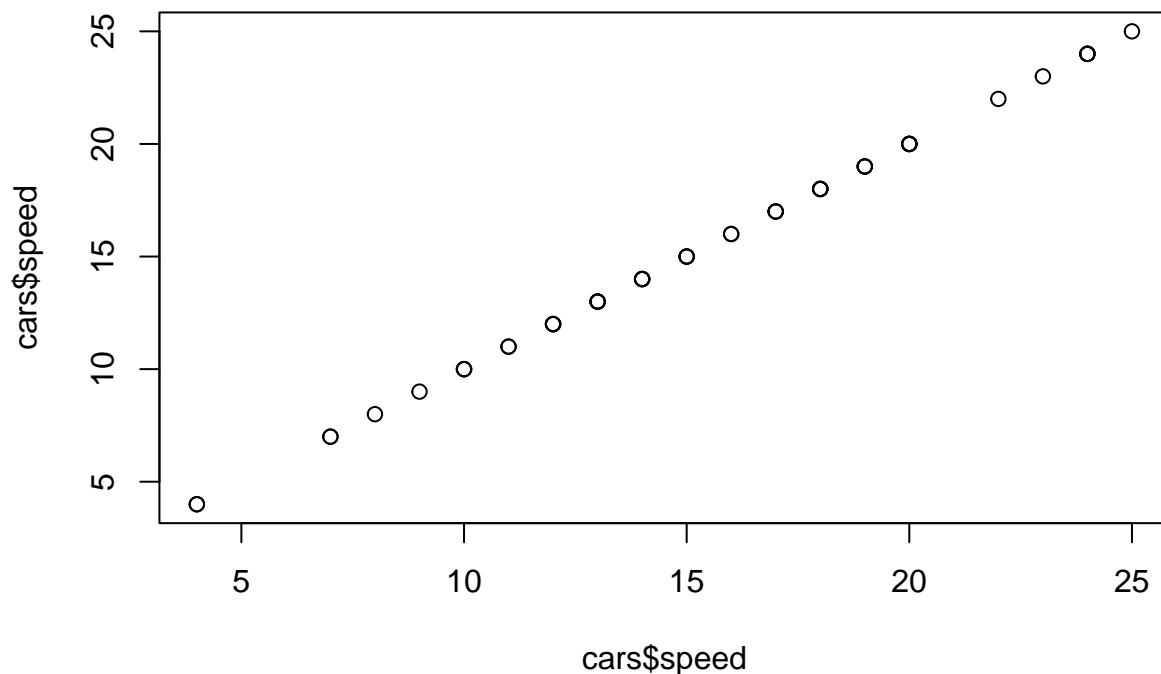
```
## 'geom_smooth()' using formula 'y ~ x'
```



```
#Grey area around line = SEM  
#Side note: ggplot is not the only graphics system  
#A very popular one is good old "base" R graphics  
plot(cars)
```



```
plot(cars$speed, cars$dist)
```



```
#Follow this same process as directed in the worksheet for lab 5. Now, we will be looking at gene expression
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
```

```
p <- ggplot(data=genes) +
  aes(x=Condition1, y=Condition2, col=State) +
  geom_point() +
  scale_color_manual(values=c("blue", "gray", "red")) +
  labs(title="Gene Expression Changes Upon Drug Treatment",
       x="Control (No Drug)",
       y="Drug Treatment")
#Question: How many genes are in this dataset?
nrow(genes)
```

```
## [1] 5196
```

```
#Question: How many genes are "up"?  
table(genes$State)
```

```
##  
##      down  unchanged      up  
##      72      4997      127
```

```
#Question: What % are up?  
table(genes$State)/nrow(genes)
```

```
##  
##      down  unchanged      up  
## 0.01385681 0.96170131 0.02444188
```

```
#Question: How do we report this as a percentage?  
table(genes$State)/nrow(genes) * 100
```

```
##  
##      down  unchanged      up  
##  1.385681  96.170131  2.444188
```

```
#Question: How do we round off to a specific number of sig figs?  
round(table(genes$State)/nrow(genes) * 100, 3)
```

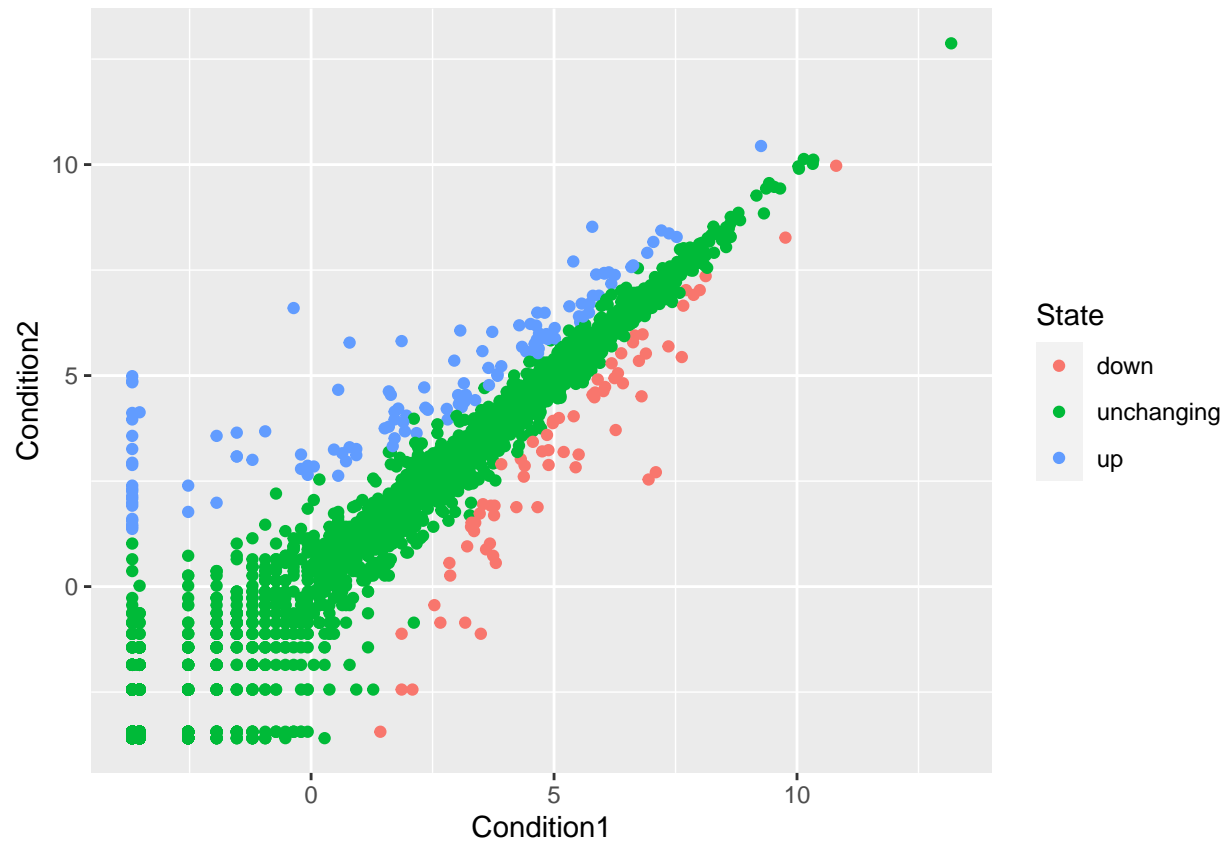
```
##  
##      down  unchanged      up  
##   1.386    96.170    2.444
```

```
#See lines 29-35 to make a figure
```

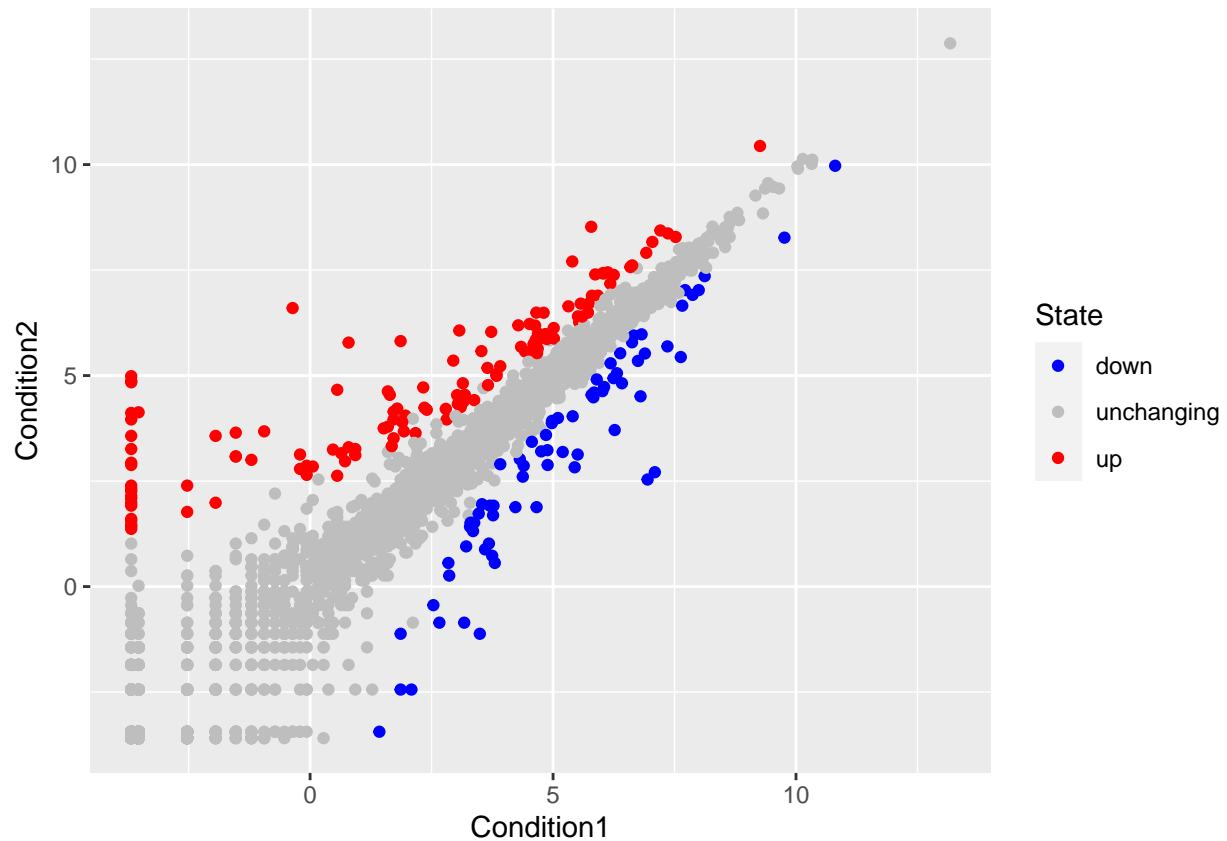
```
#General notes on making the figure:
```

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

```
p
```



```
# I like it but not the default colors, let's change them  
p + scale_color_manual(values=c("blue", "gray", "red"))
```



*#Section of lab: Going Further*

```
library(gapminder)
library(dplyr)
```

```
##
```

```
## Attaching package: 'dplyr'
```

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

```
##
```

```
##   filter, lag
```

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

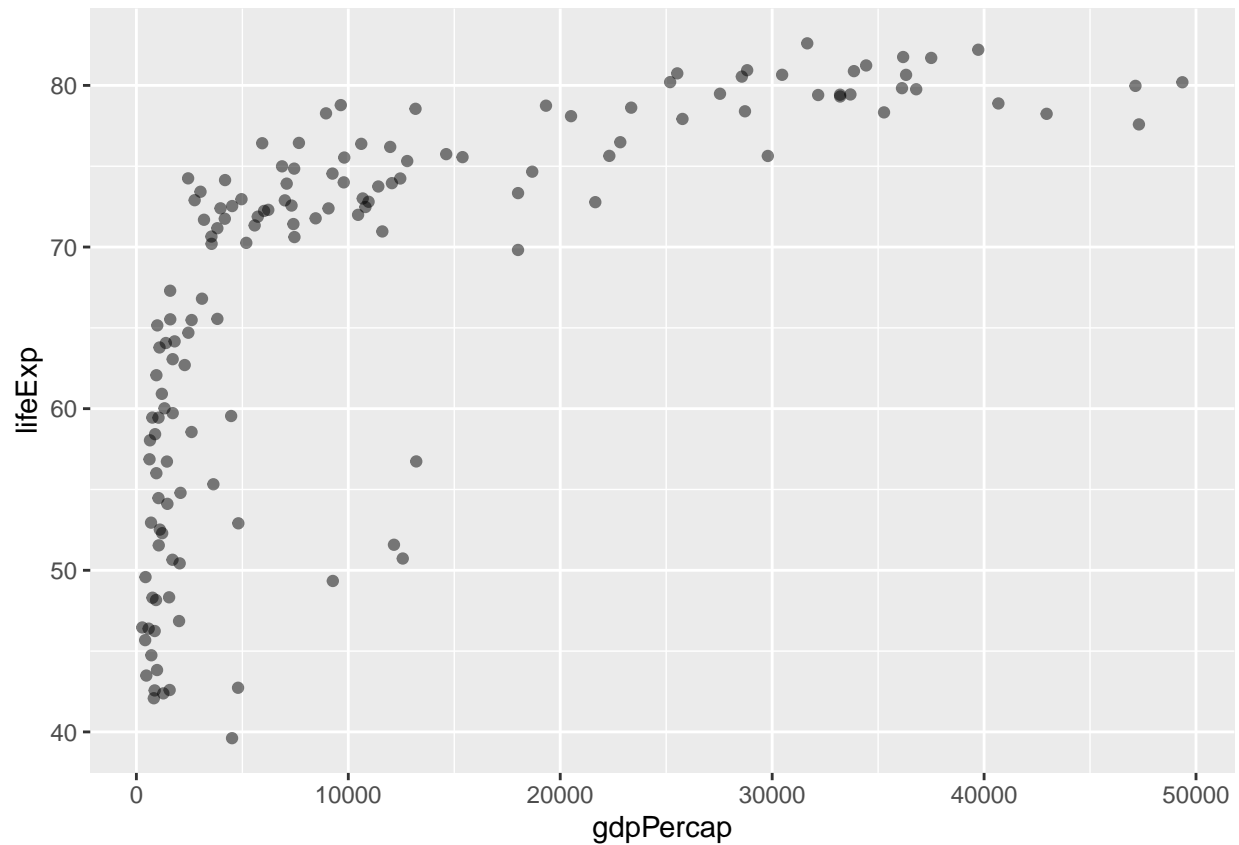
```
##
```

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

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

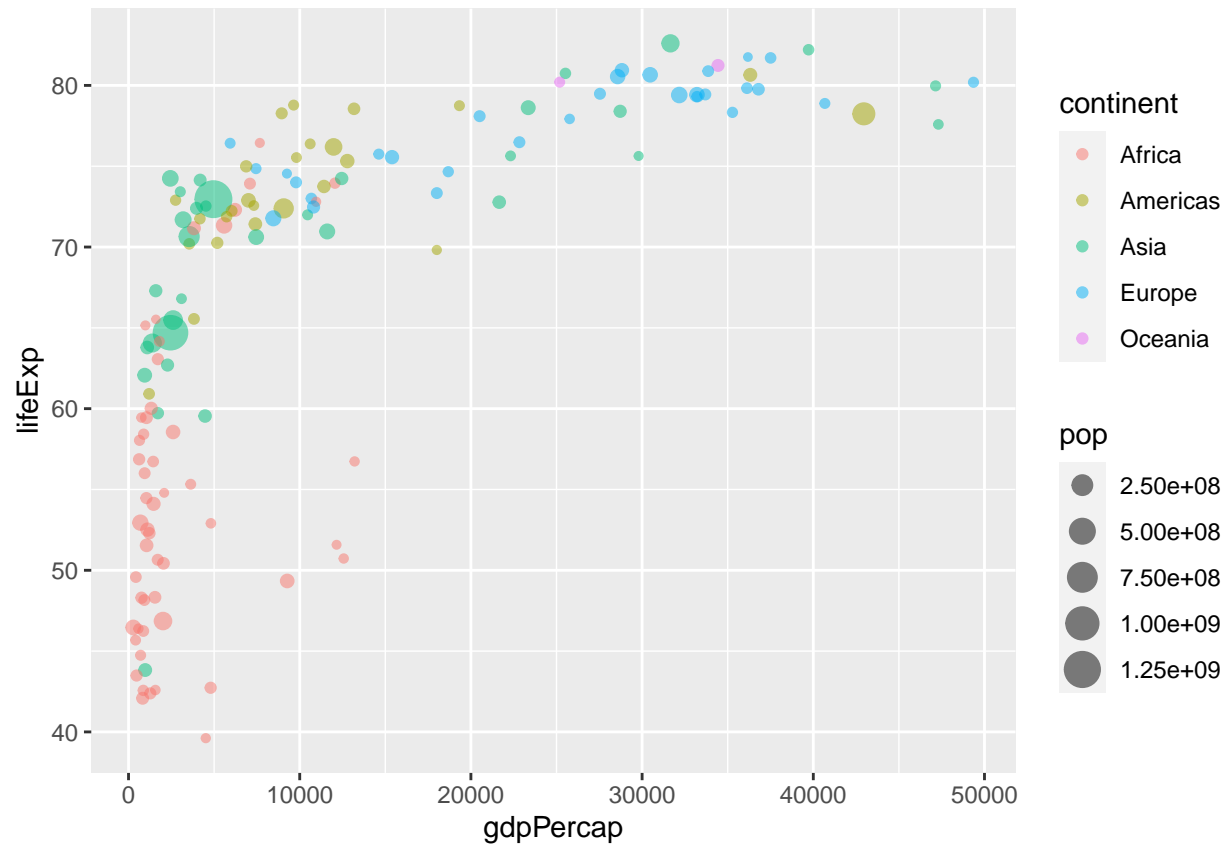
*#Let's make another figure - first, a scatterplot!*

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

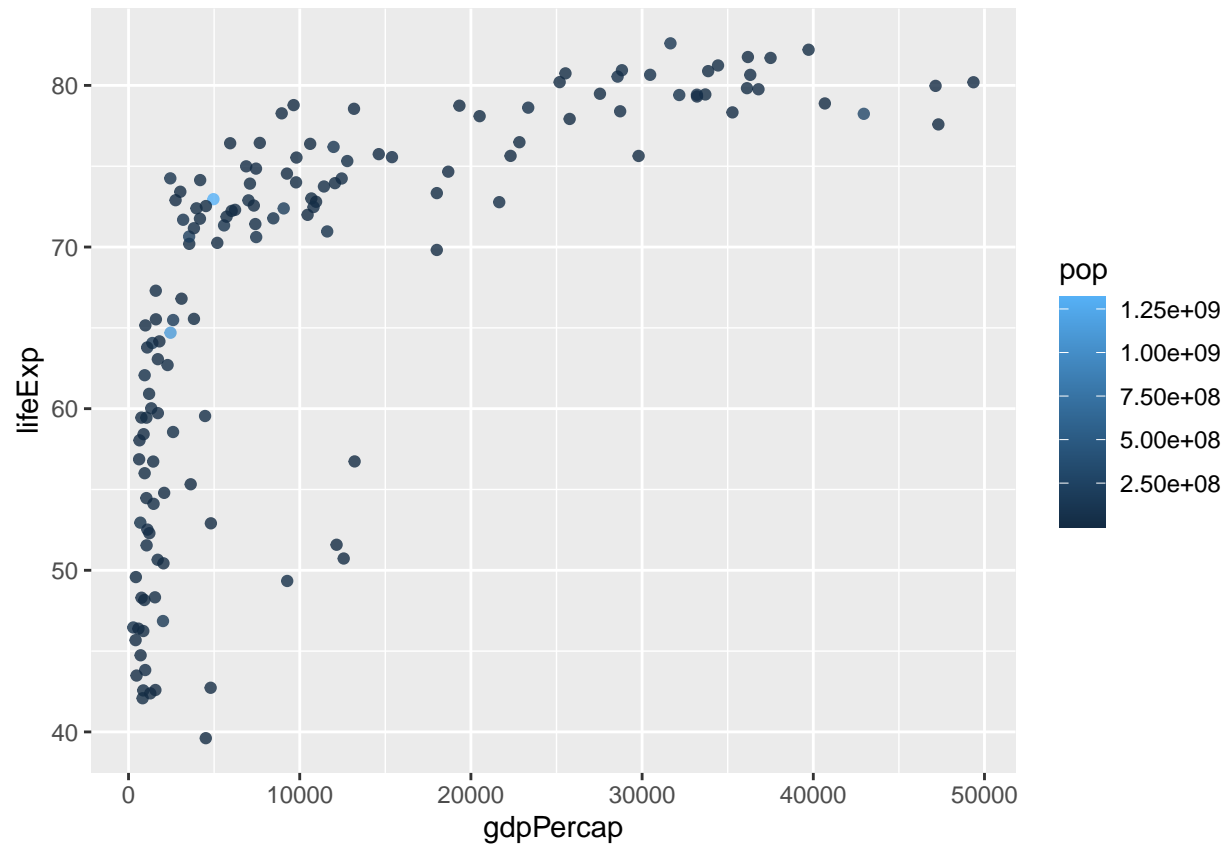


```
#The "alpha" value will make the points on the scatterplot more transparent  
#To add even more complexity to the plot, we will add the following to our calls:  
ggplot(gapminder_2007) +  
  aes(x=gdpPercap, y=lifeExp, color=continent, size=pop) +  
  geom_point(alpha=0.5)
```

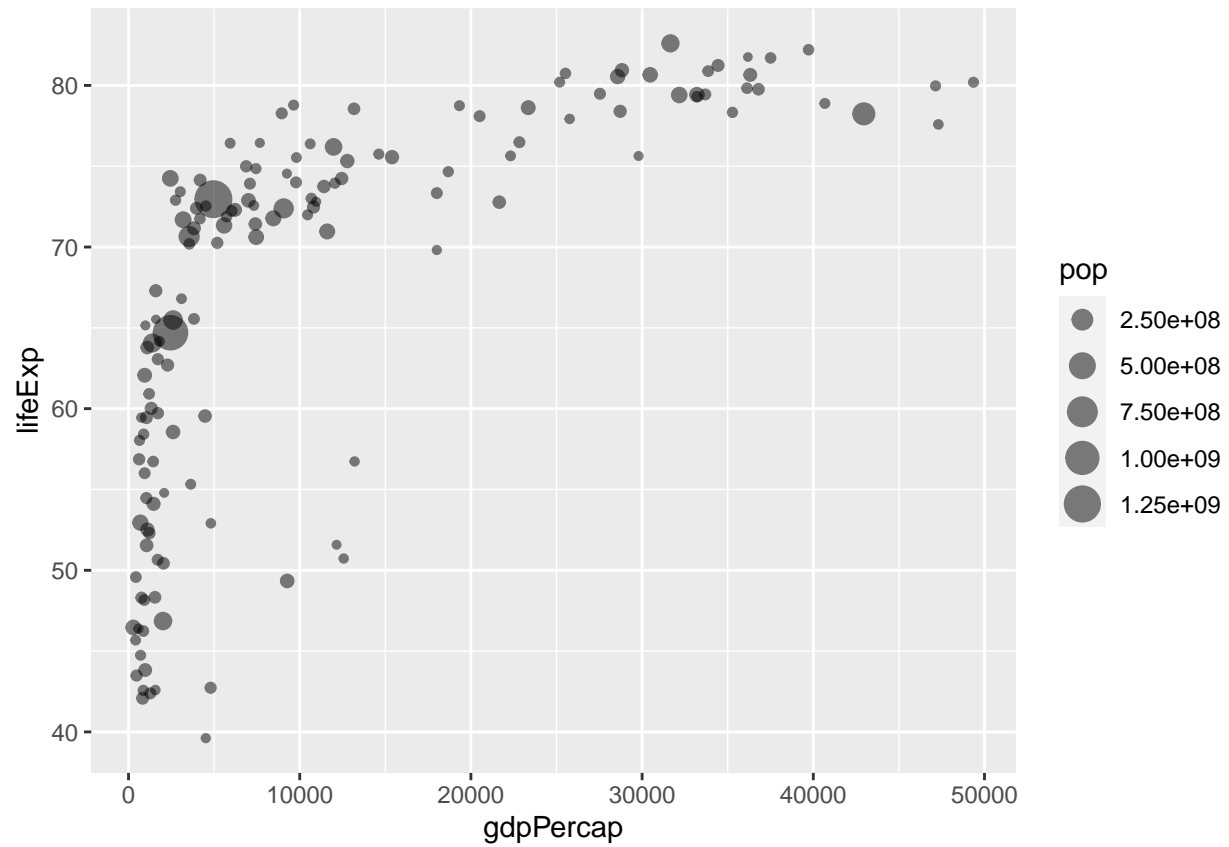




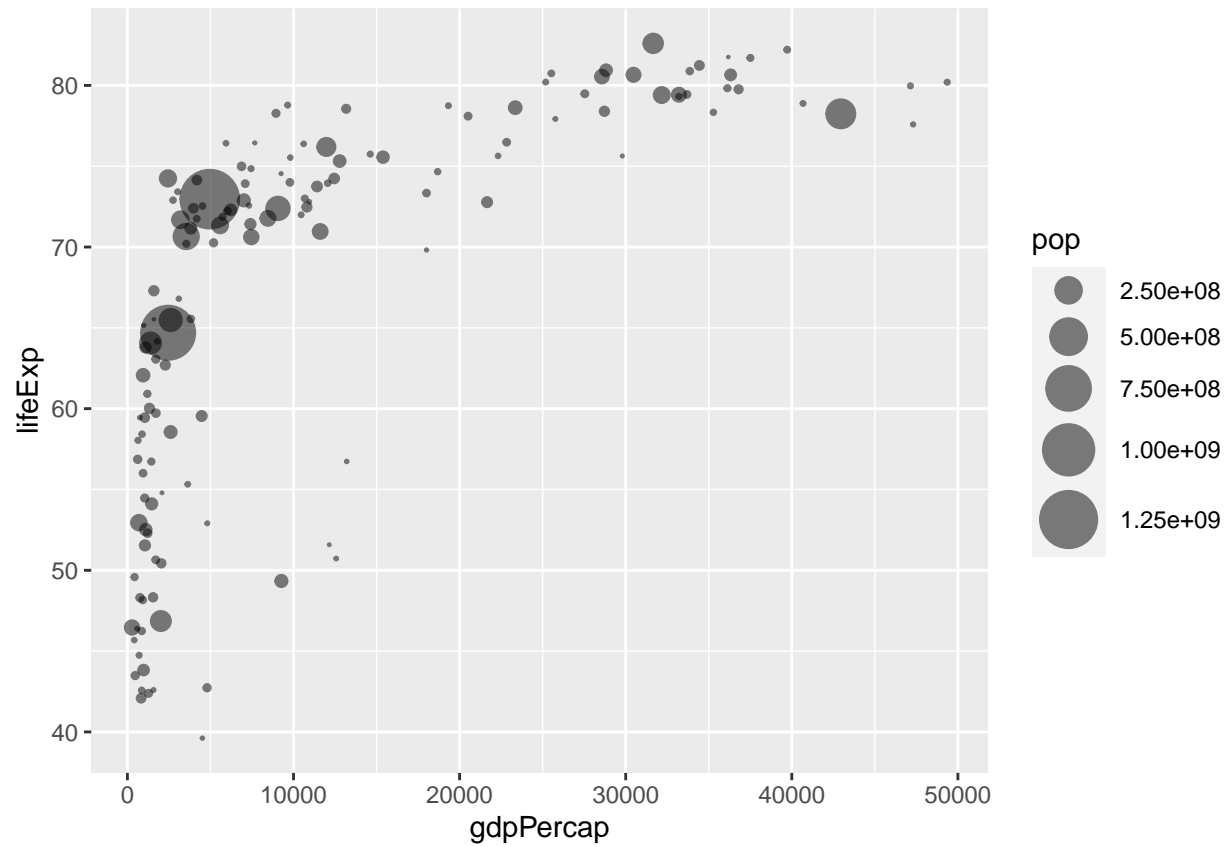
```
#Now try this...  
ggplot(gapminder_2007) +  
  aes(x=gdpPercap, y=lifeExp, color=pop) +  
  geom_point(alpha=0.8)
```



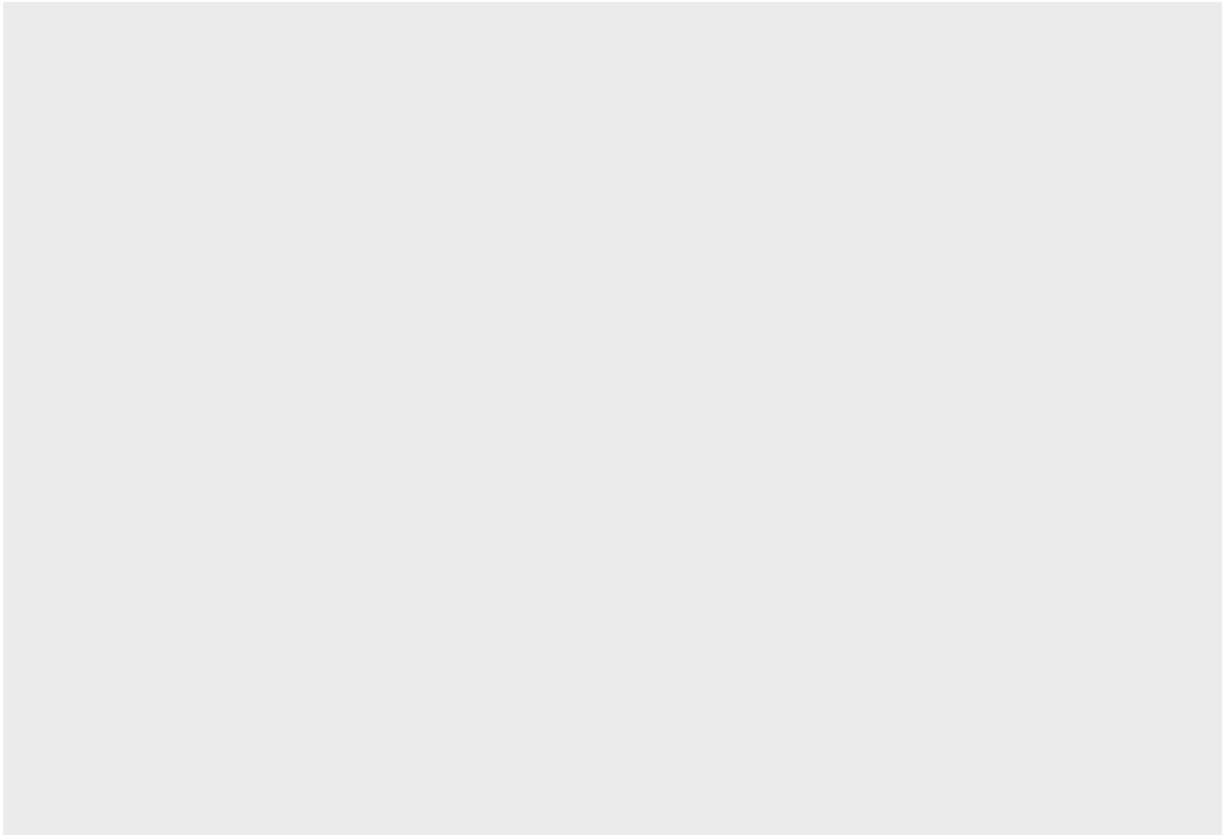
```
#Now try this...  
ggplot(gapminder_2007) +  
  aes(x=gdpPercap, y=lifeExp, size=pop) +  
  geom_point(alpha=0.5)
```



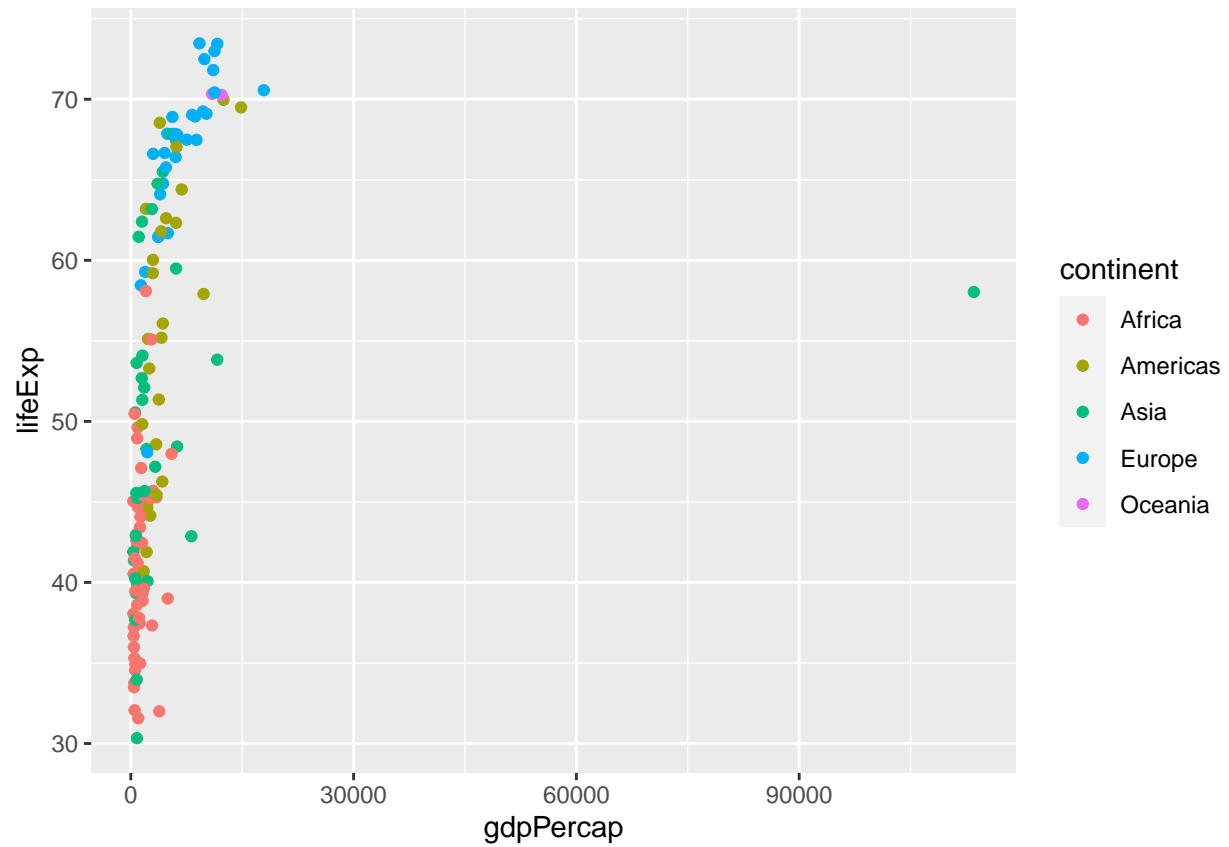
```
#To add scaling information:  
ggplot(gapminder_2007) +  
  aes(x=gdpPercap, y=lifeExp, size=pop) +  
  geom_point(alpha=0.5) +  
  scale_size_area(max_size = 10)
```



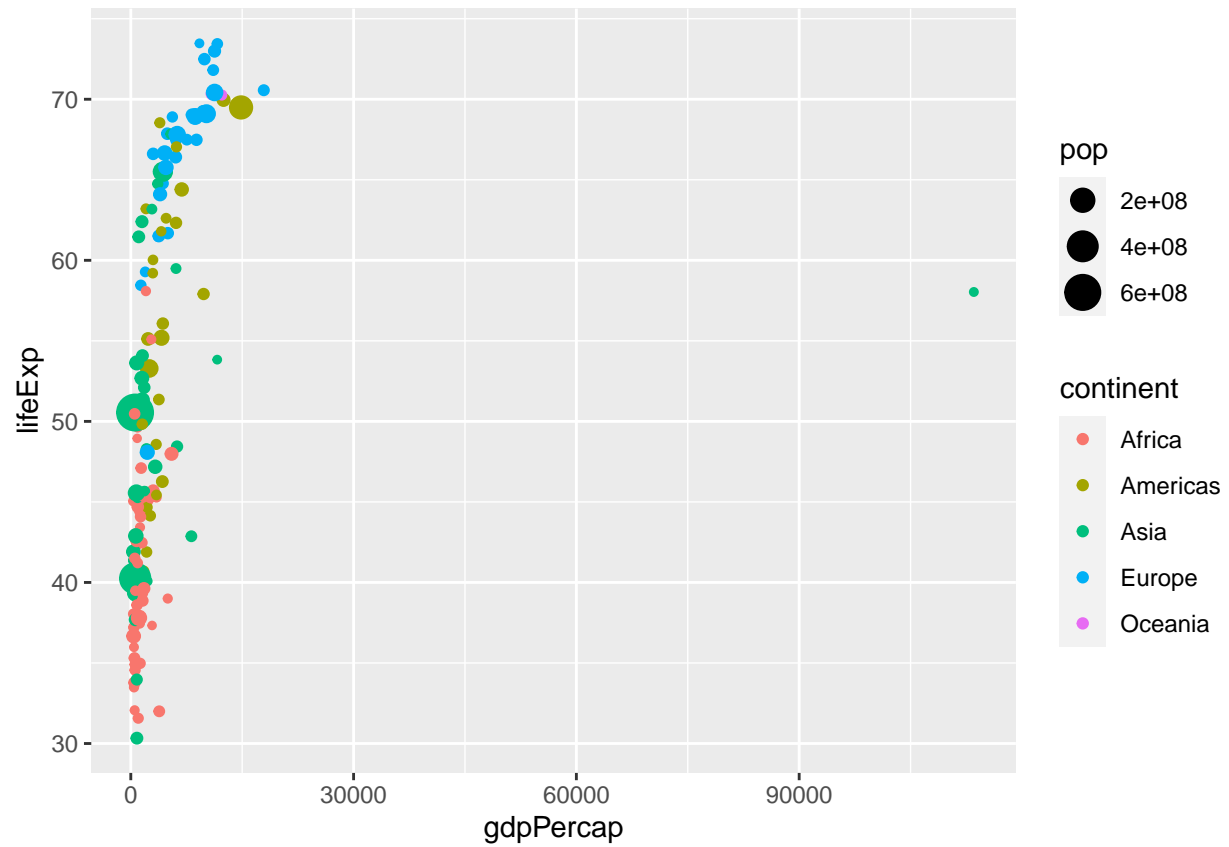
```
#Now, let's try putting this all together to generate a new figure:  
gapminder_1957 <- gapminder %>% filter(year==1957)  
ggplot(gapminder_1957)
```



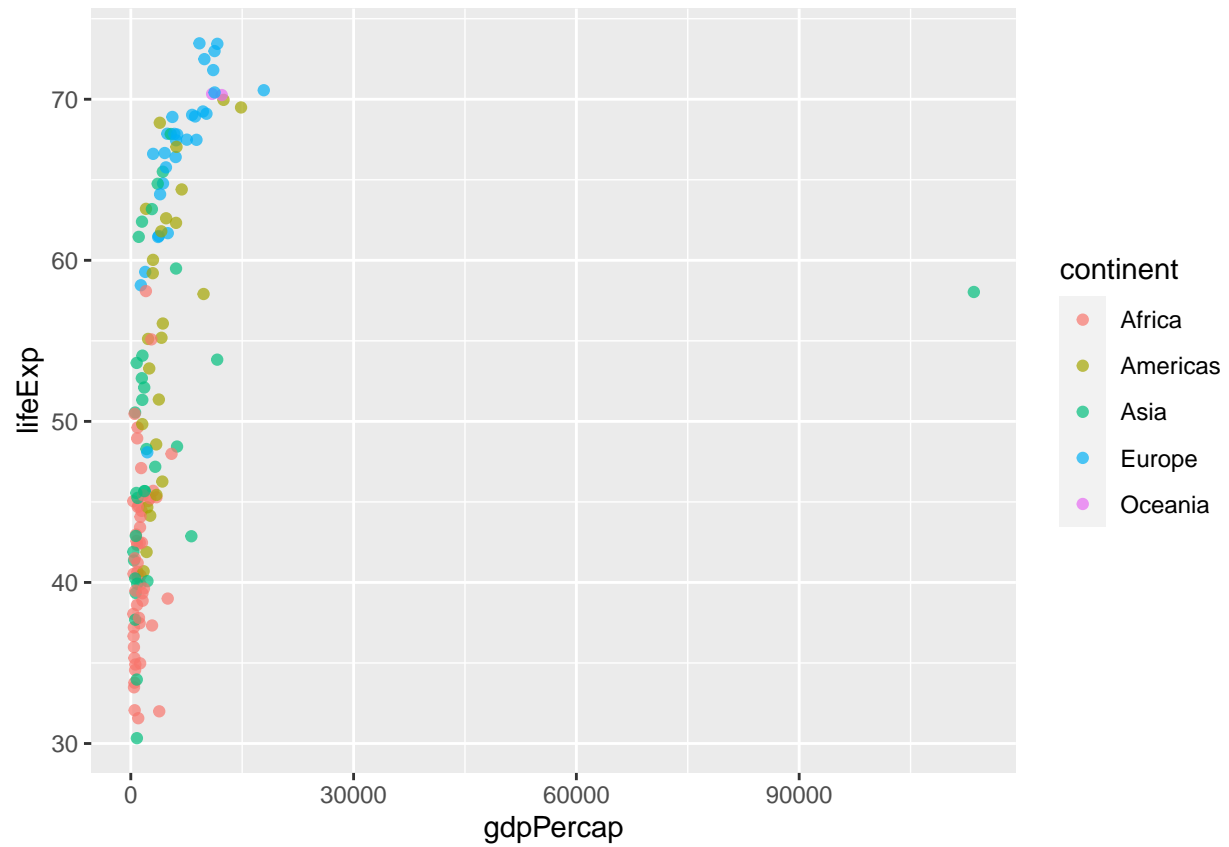
```
ggplot(gapminder_1957) +  
  aes(x=gdpPercap, y=lifeExp, color=continent) +  
  geom_point()
```



```
ggplot(gapminder_1957) +  
  aes(x=gdpPercap, y=lifeExp, color=continent, size=pop) +  
  geom_point()
```

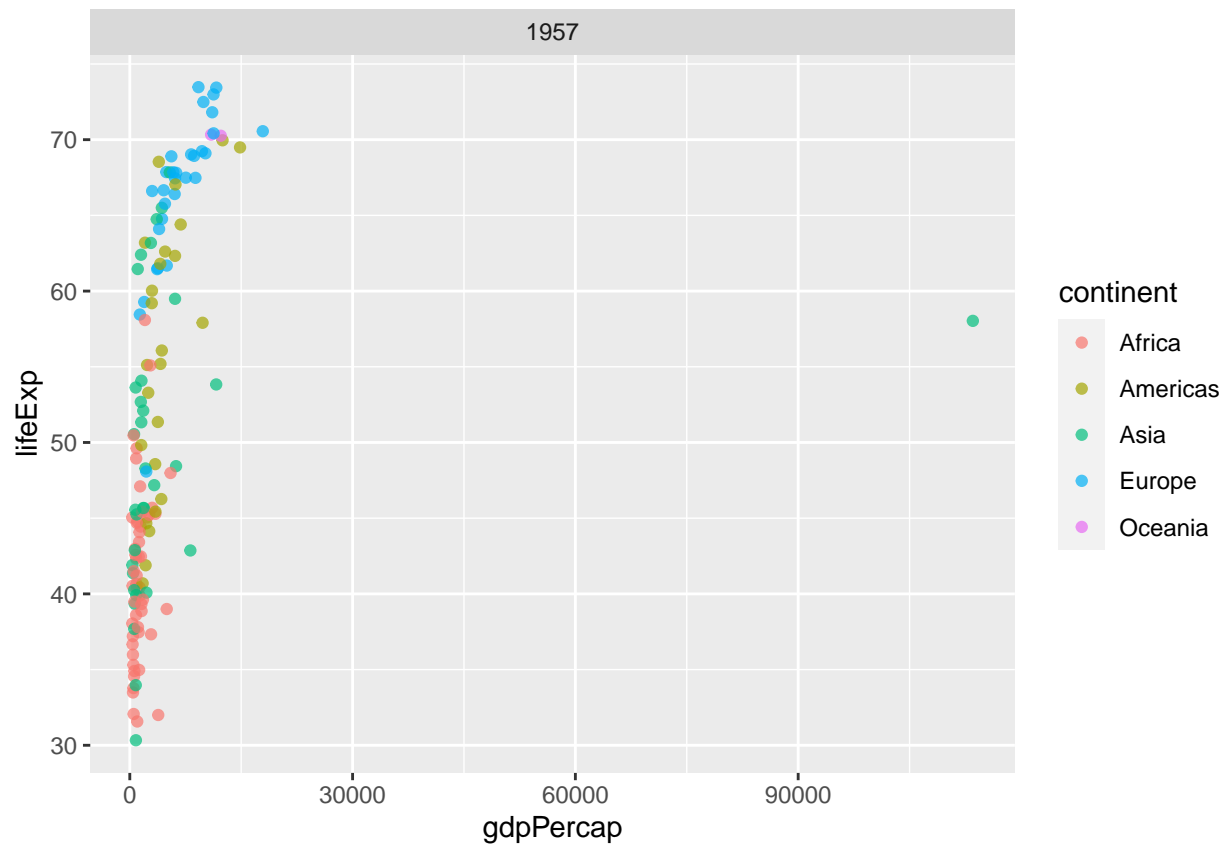


```
ggplot(gapminder_1957) +
  aes(x=gdpPercap, y=lifeExp, color=continent) +
  geom_point(alpha=0.7) +
  scale_size_area(max_size = 15)
```

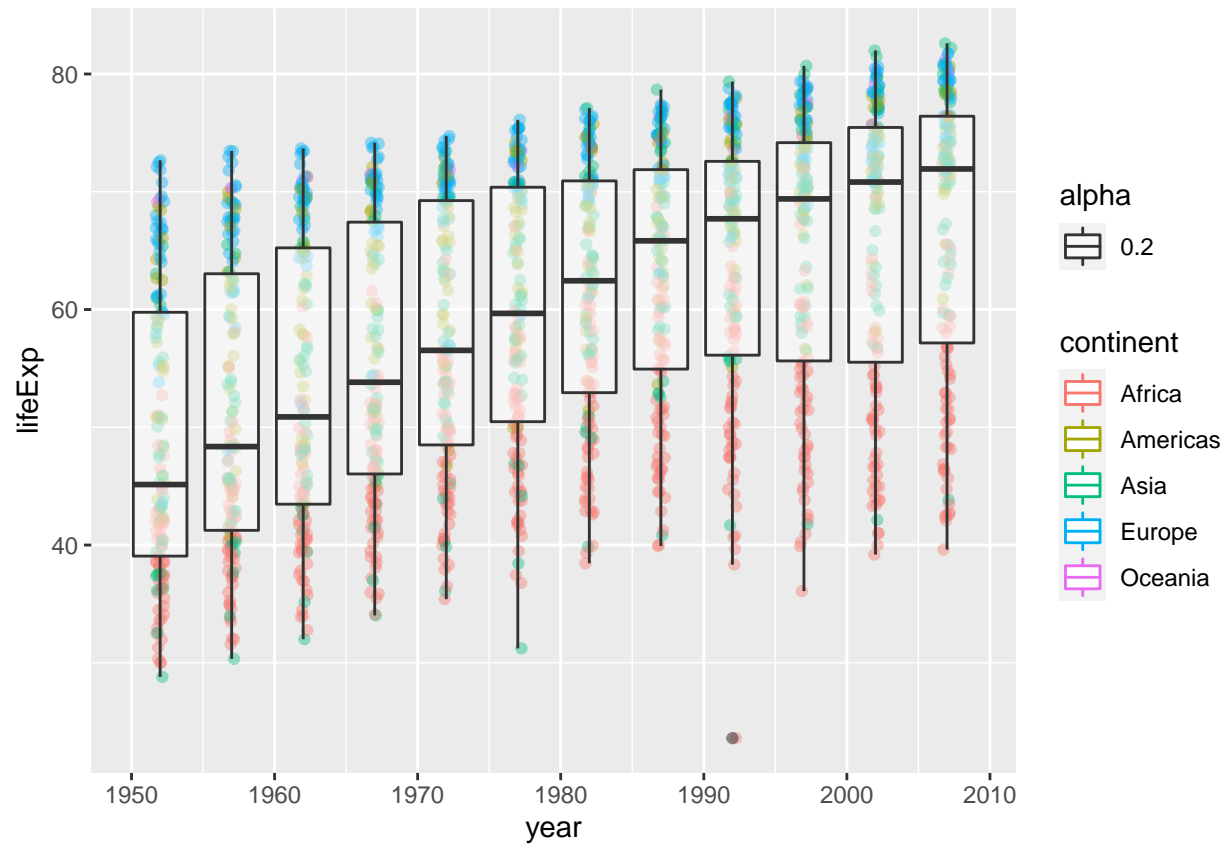


```
ggplot(gapminder_1957) +  
  aes(x=gdpPercap, y=lifeExp, color=continent) +  
  geom_point(alpha=0.7) +  
  scale_size_area(max_size = 15) +  
  facet_wrap(~year)
```

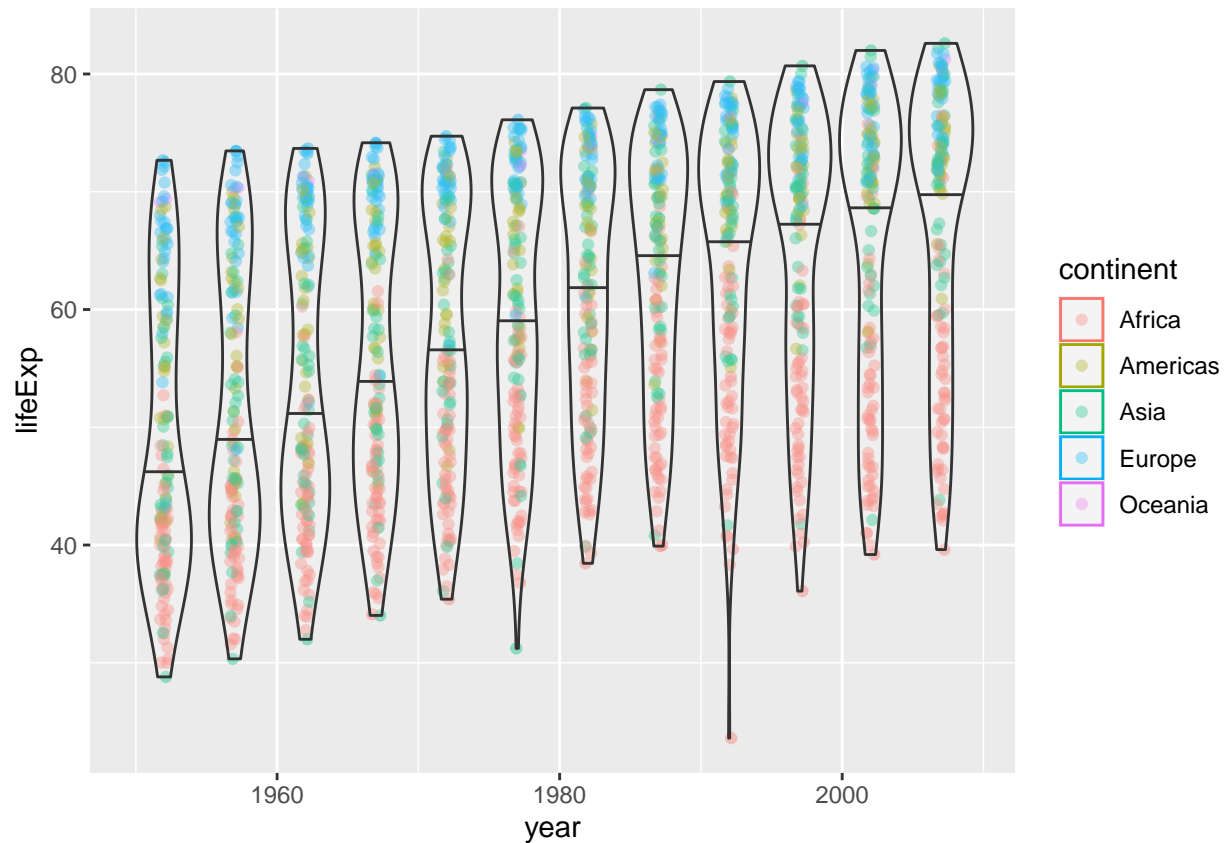




```
#Let's make a new plot of year vs. lifeExp
ggplot(gapminder) +
  aes(x=year, y=lifeExp, col=continent) +
  geom_jitter(width=0.3,alpha=0.4) +
  geom_boxplot(aes(group=year, alpha=0.2))
```



```
ggplot(gapminder) +
  aes(x=year, y=lifeExp, col=continent) +
  geom_jitter(width=0.3,alpha=0.4) +
  geom_violin(aes(group=year), alpha=0.2,
              draw_quantiles=0.5)
```



```
#Install the plotly
#Install.packages("plotly")
library(plotly)
```

```
##
## Attaching package: 'plotly'

## The following object is masked from 'package:ggplot2':
##
##   last_plot

## The following object is masked from 'package:stats':
##
##   filter

## The following object is masked from 'package:graphics':
##
##   layout
```

```
#ggplotly()
```