# Class 5 Data Visualization with ggplot

Job Rocha (A59023124)

### Using ggplot

To use ggplot2 we first need to install it on our computers. To do this we will use the function `install.packages()`.

Before I use any package functions I have to load them up with a `libray()` call, like so:

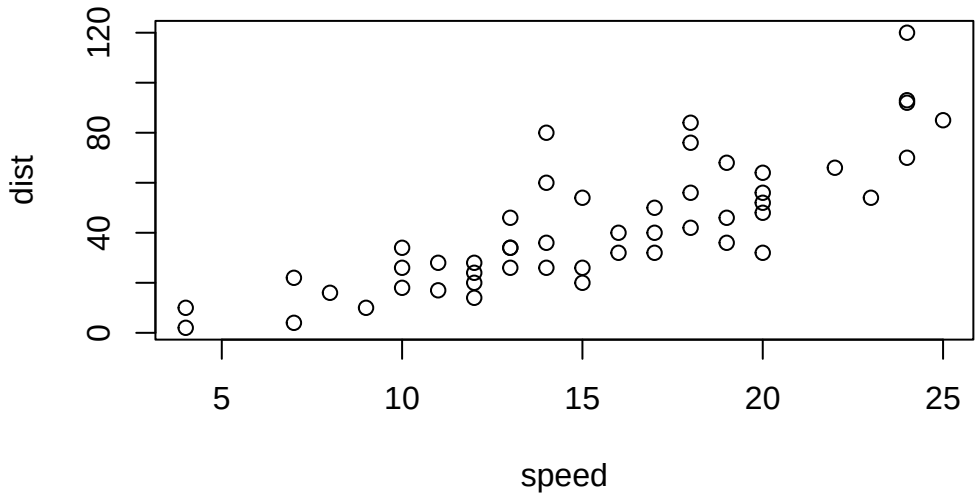
```
library(ggplot2)

ggplot(cars)
```

```
head(cars)
```

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

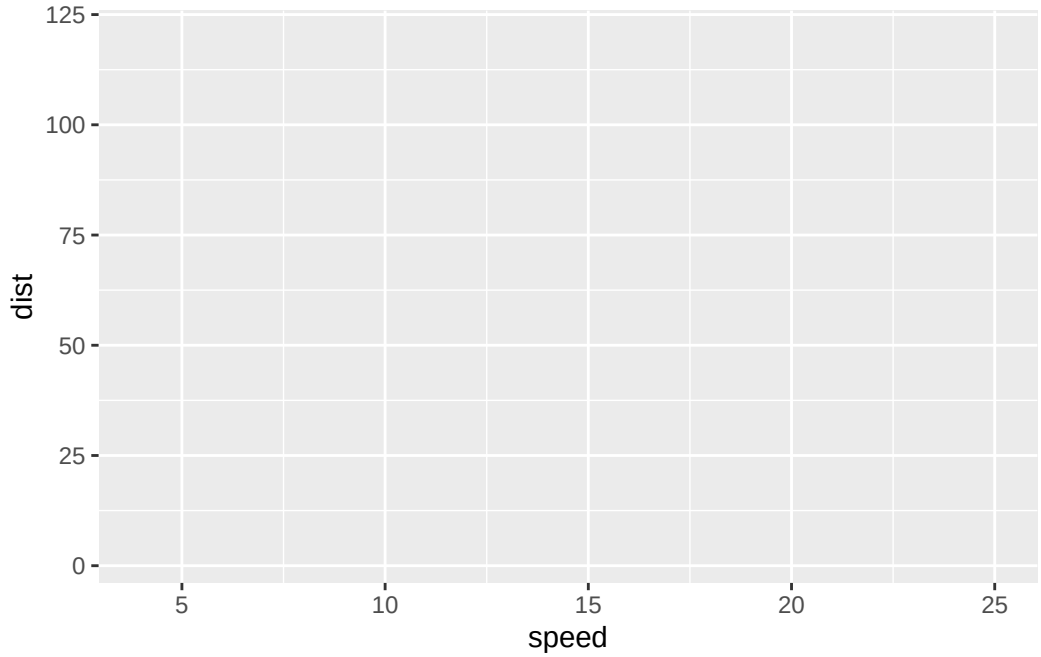
There is always the "base R" graphics system, i.e. `plot()`

```
plot(cars)
```

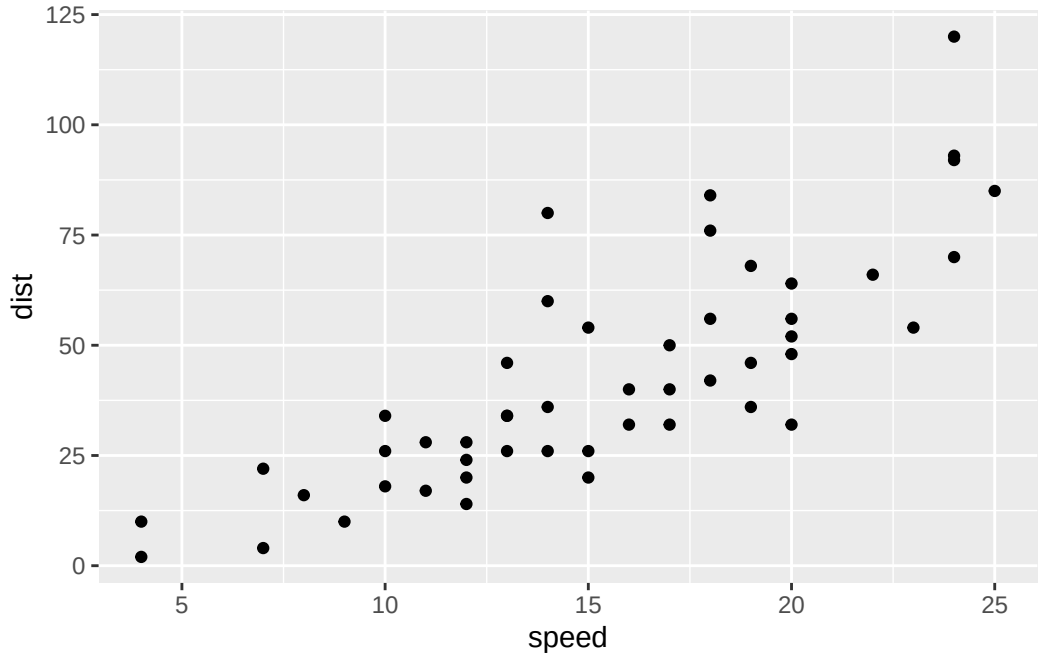



To use ggplot I need to spell out at least 3 things: - data (the stuff I want to plot as a data.frame) - aesthetics(aes() values - how the data map to the plot) - geoms (how I want things drawn)

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

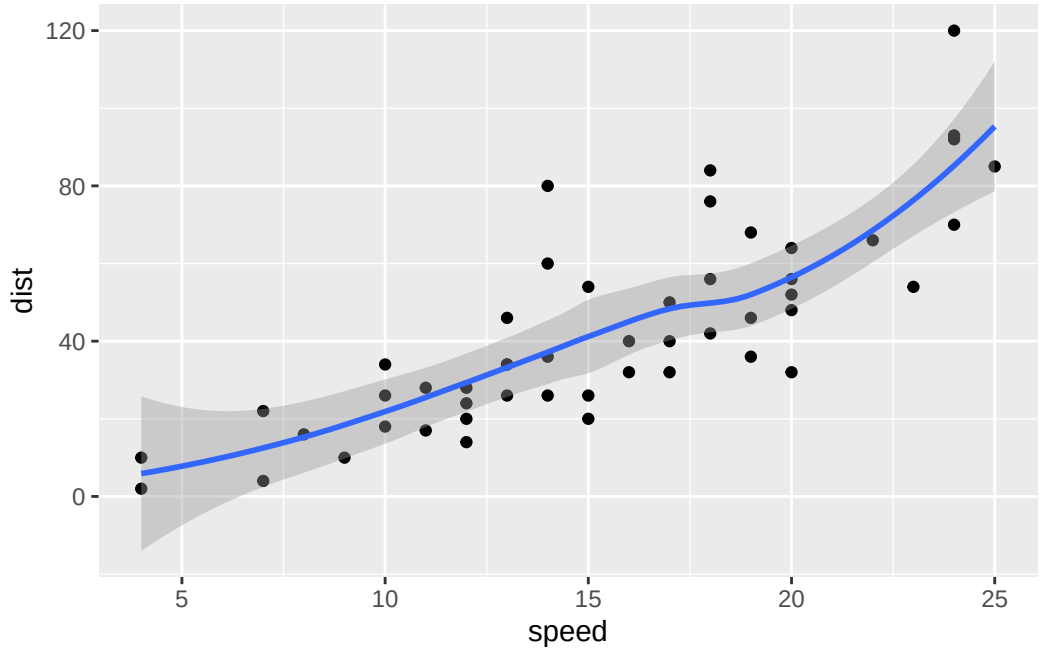



```
ggplot(cars) +
  aes(x=speed, y=dist) +
  geom_point()
```



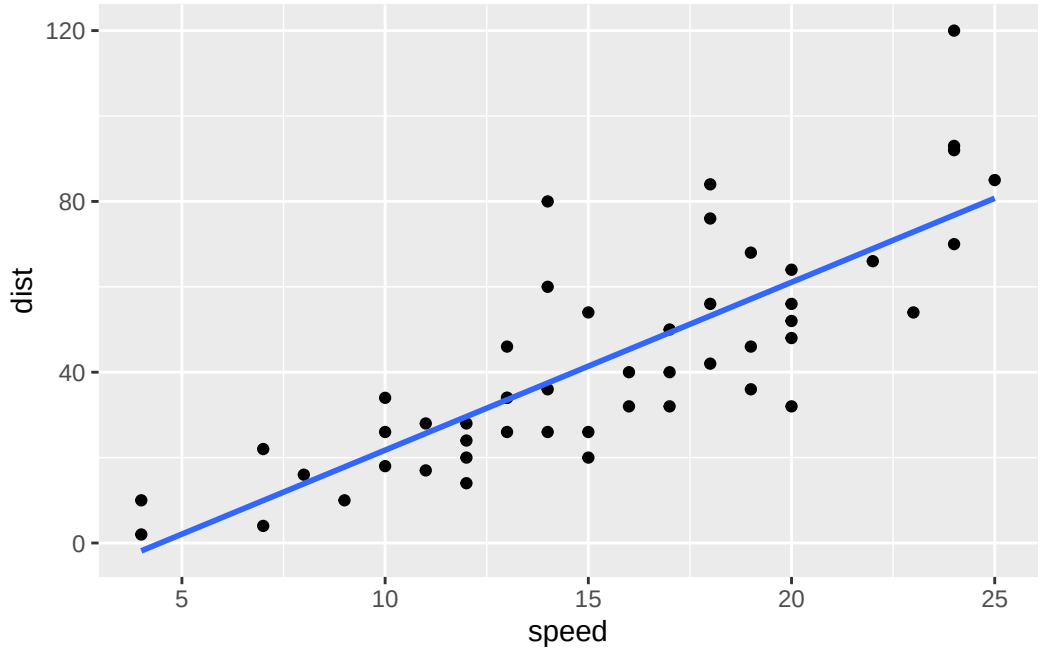

```
ggplot(cars) +
  aes(x=speed, y=dist) +
  geom_point() +
  geom_smooth()
```

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



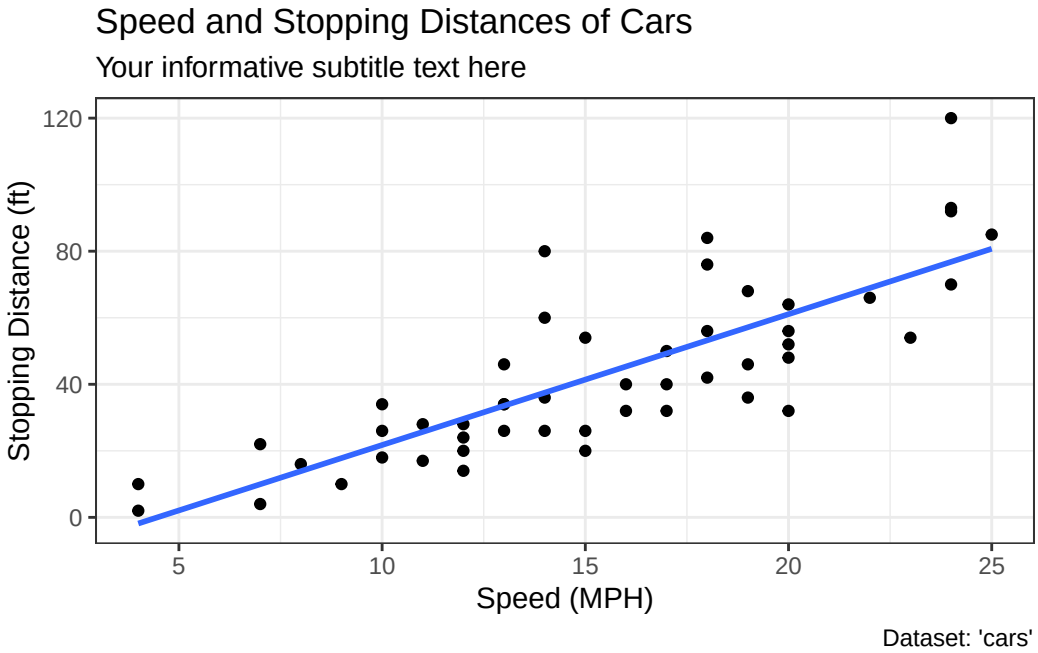

```
ggplot(cars) +
  aes(x=speed, y=dist) +
  geom_point() +
  geom_smooth(method="lm", se=FALSE)
```

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




```
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 = "Your informative subtitle text here",
       caption="Dataset: 'cars'") +
  theme_bw()
```

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




** Questions **

```
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
```

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

```
# Result = 5196
nrow(genes)
```

```
[1] 5196
```

Q. 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?

```
colnames(genes)
```

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

```
# Result = 4
ncol(genes)
```

```
[1] 4
```

Q. 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?

```
# Result = 127
table(genes$State)
```

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

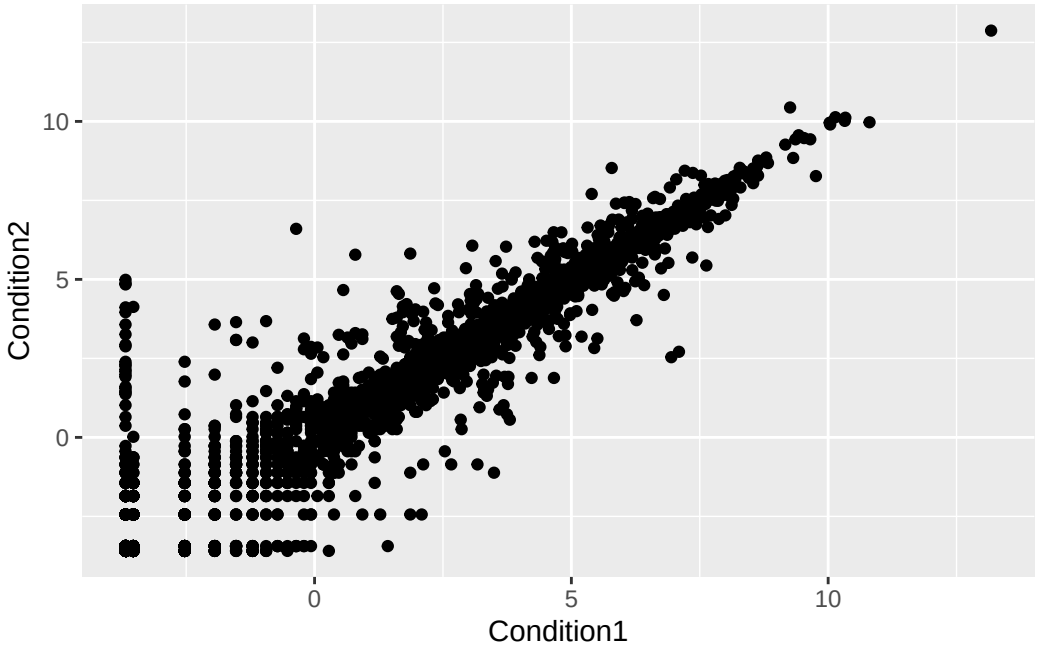
Q. Using your values above and 2 significant figures. What fraction of total genes is up-regulated in this dataset?

```
# Result = 0.0244
sum(genes$State == "up") / nrow(genes)
```

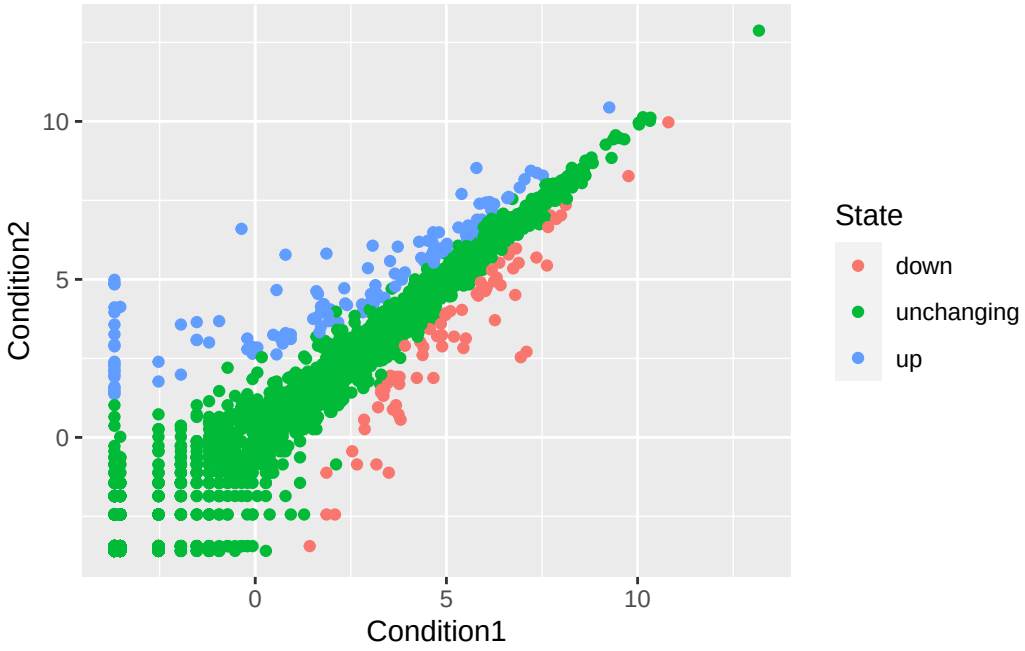
```
[1] 0.02444188
```

Q. Complete the code below to produce the following plot ggplot*() + aes(x=Condition1, y=*) _____

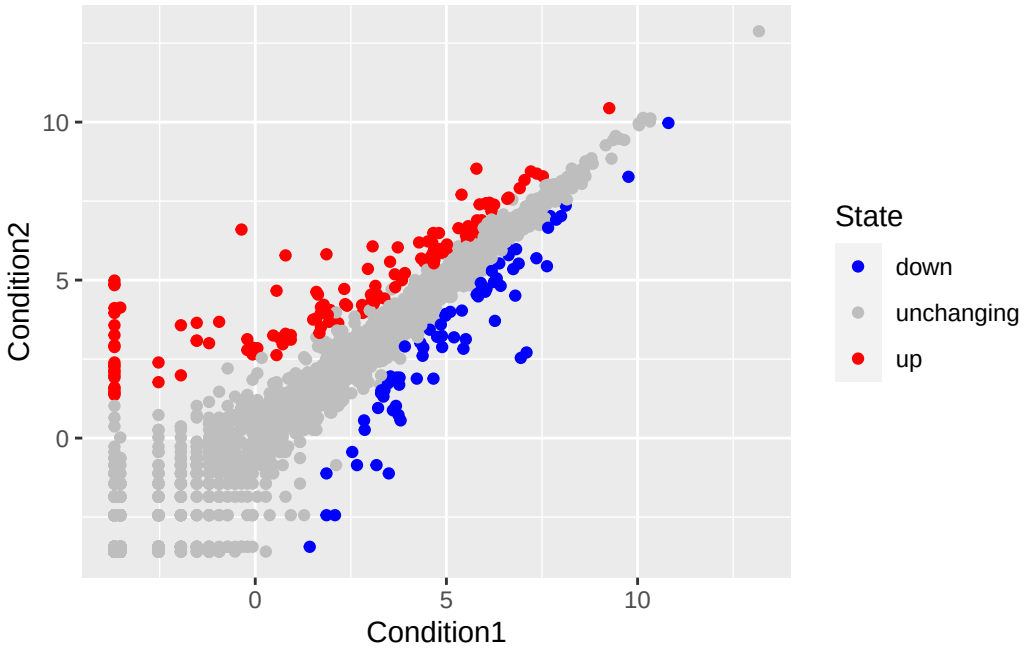
```
ggplot(genes) +
    aes(x=Condition1, y=Condition2) +
    geom_point()
```




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

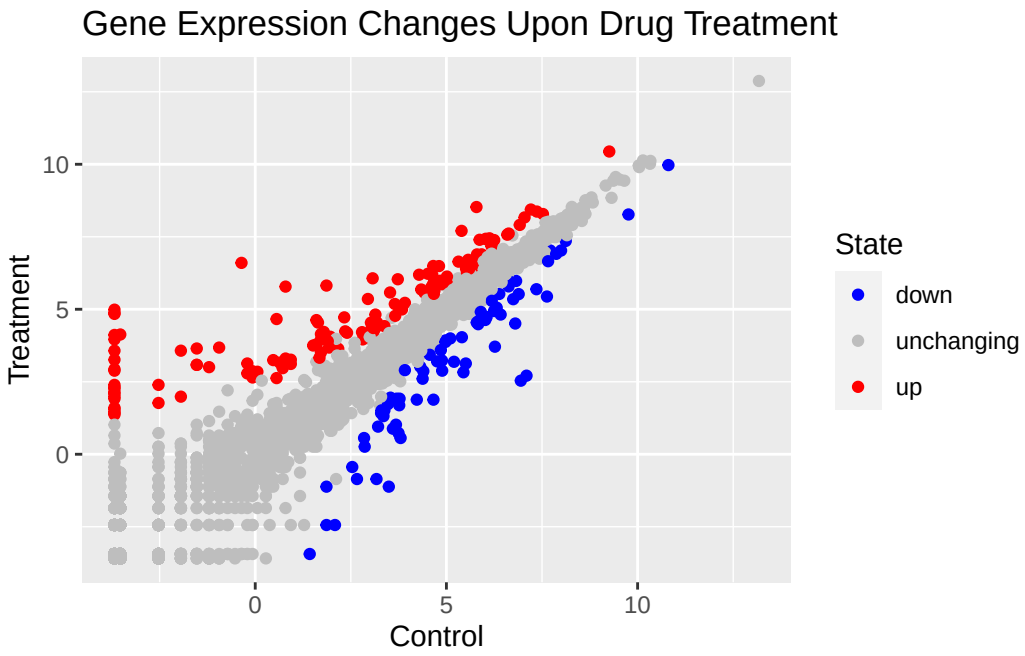



```
p + scale_colour_manual( values=c("blue","gray","red") )
```



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

```
p + scale_colour_manual( values=c("blue","gray","red") ) +
  labs(title="Gene Expression Changes Upon Drug Treatment",
       x="Control",
       y="Treatment")
```




```
# File location online
url <- "https://raw.githubusercontent.com/jennybc/gapminder/master/inst/extdata/gapminder.

gapminder <- read.delim(url)
```

```
# install.packages("dplyr")  ## un-comment to install if needed
library(dplyr)
```

```
Attaching package: 'dplyr'

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

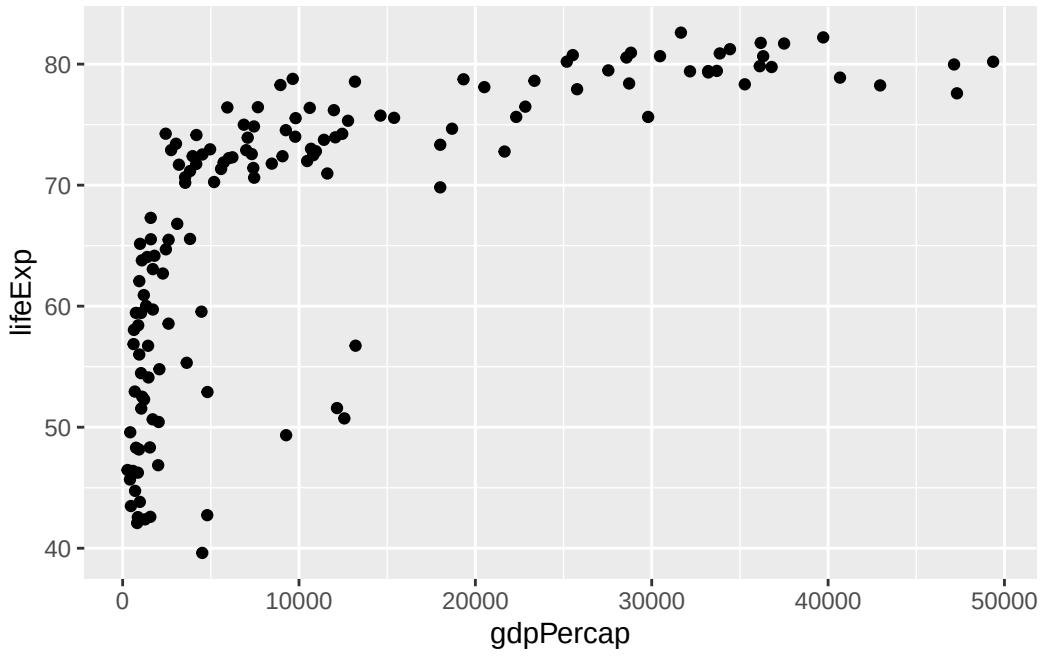
```
    filter, lag

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

    intersect, setdiff, setequal, union
```

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

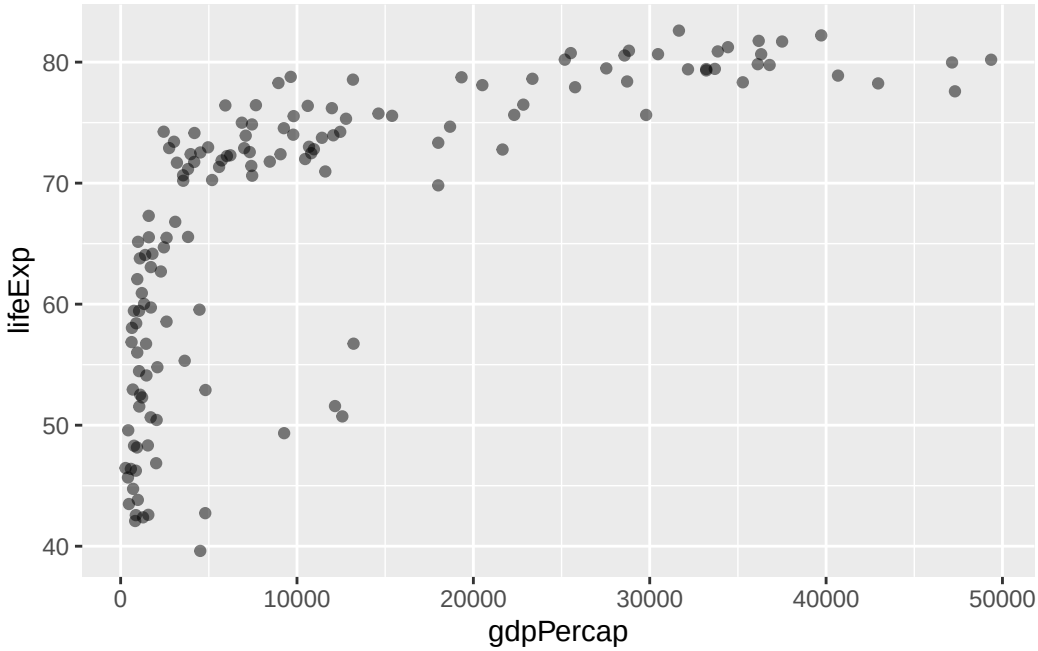
Let's consider the gapminder_2007 dataset which contains the variables GDP per capita gdpPercap and life expectancy lifeExp for 142 countries in the year 2007

Q. Complete the code below to produce a first basic scater plot of this gapminder_2007 dataset: ggplot(gapminder_2007) + aes(x=, ***y=***) + ____

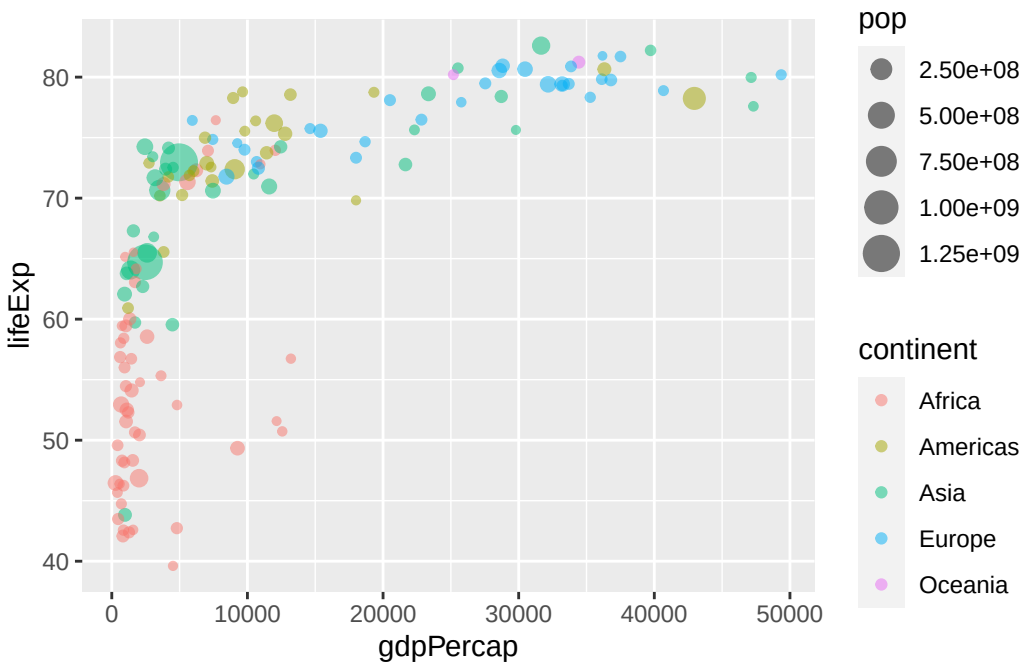
```
ggplot(gapminder_2007, aes(x=gdpPercap, y=lifeExp)) +
  geom_point()
```



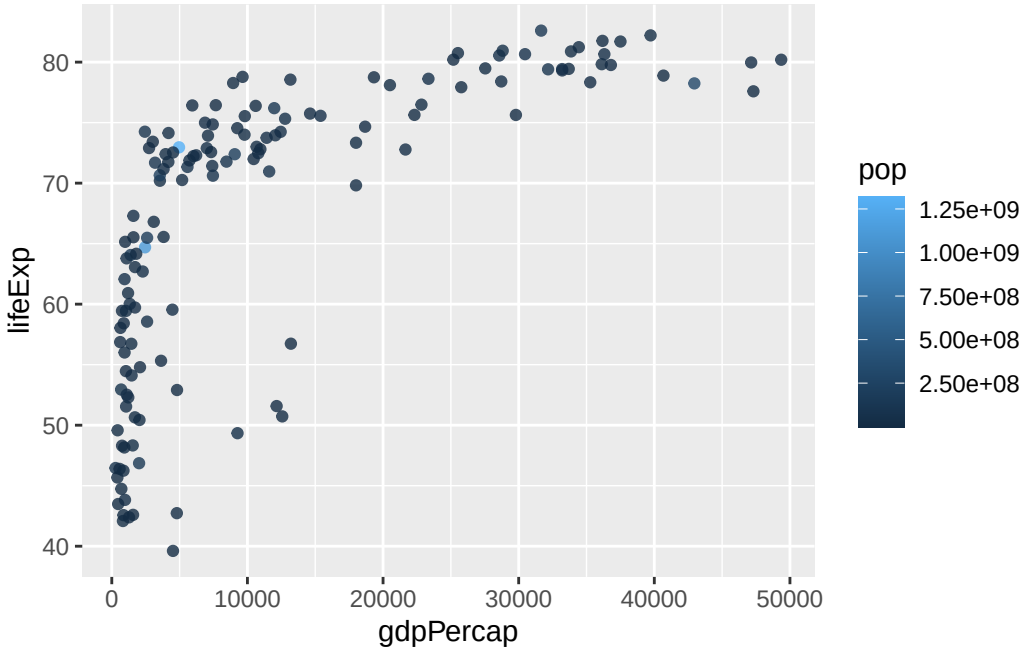

```
ggplot(gapminder_2007) +
  aes(x=gdpPercap, y=lifeExp) +
  geom_point(alpha=0.5)
```



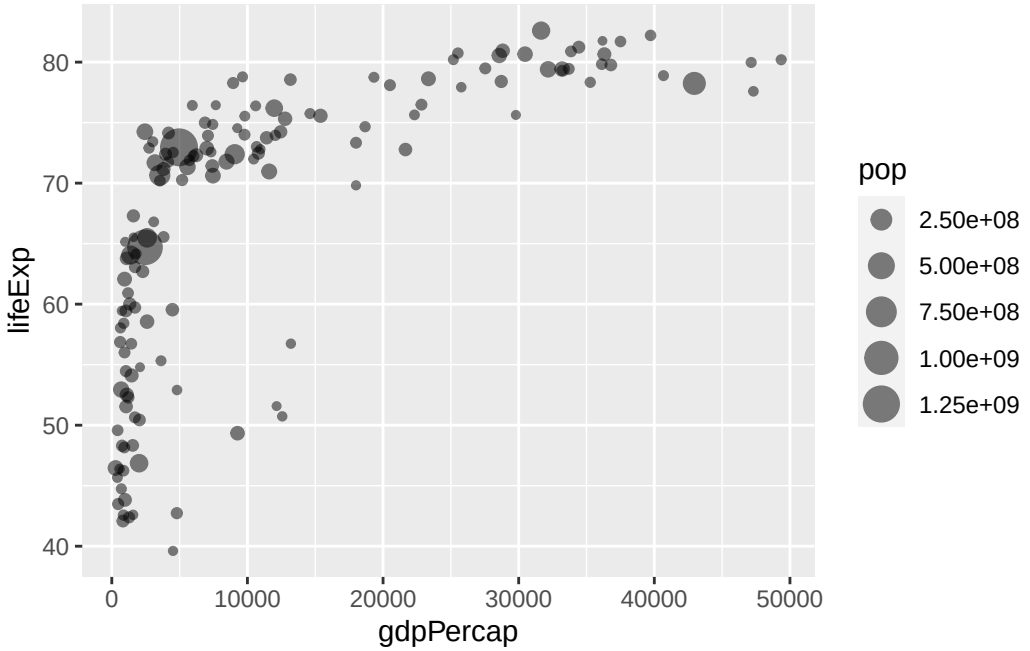

```
ggplot(gapminder_2007) +
  aes(x=gdpPercap, y=lifeExp, color=continent, size=pop) +
  geom_point(alpha=0.5)
```



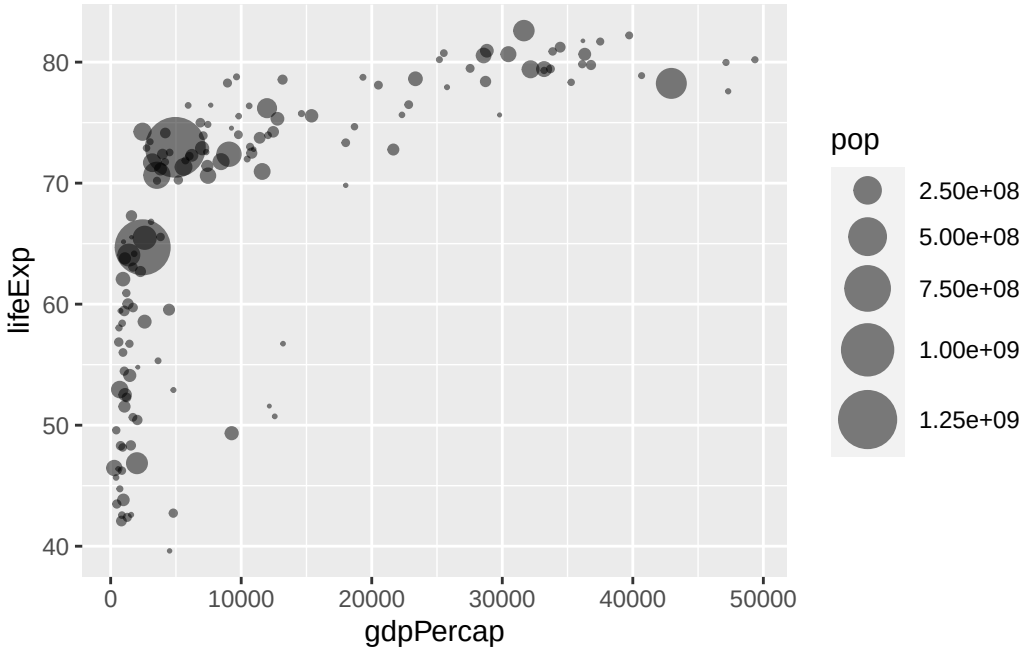

```
ggplot(gapminder_2007) +
  aes(x = gdpPercap, y = lifeExp, color = pop) +
  geom_point(alpha=0.8)
```




```
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)
```



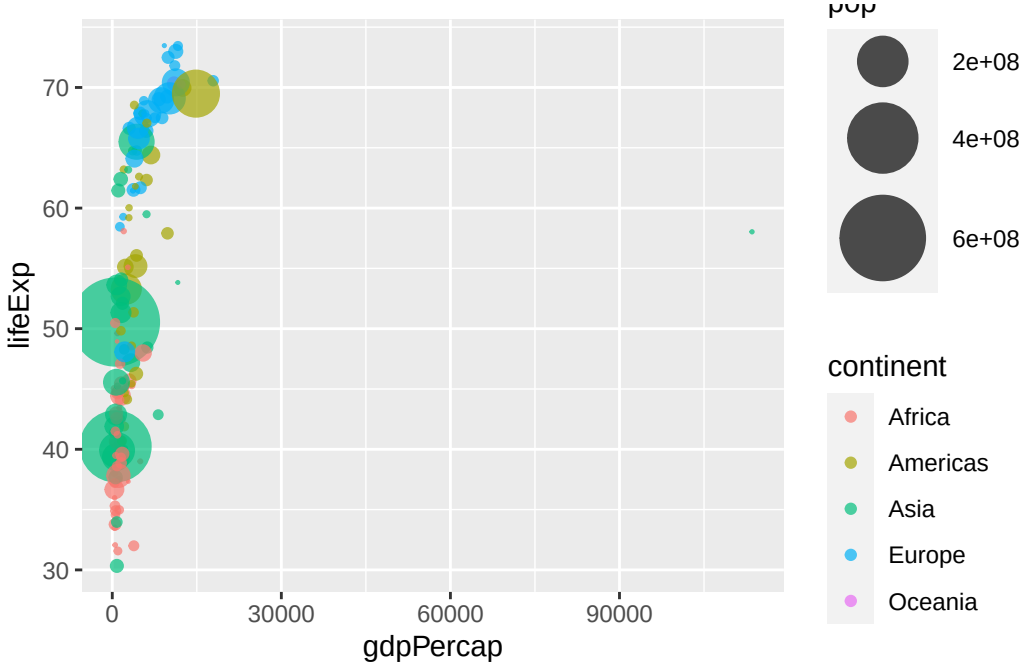

```
gapminder_1957 <- gapminder %>% filter(year==1957)
```

Q. 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?

Steps to produce your 1957 plot should include:

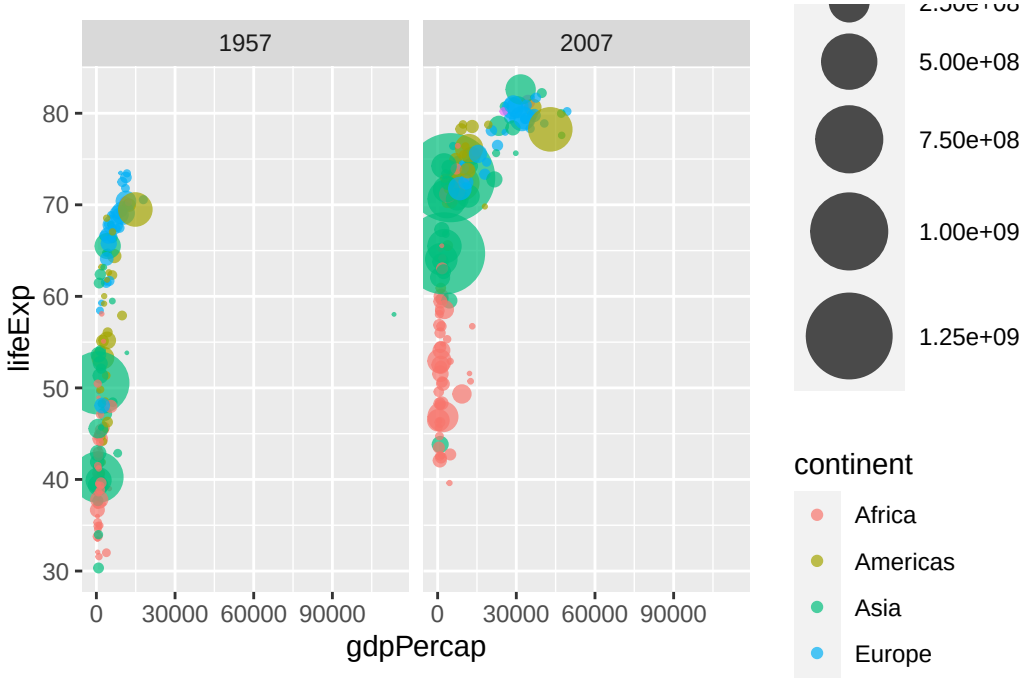
Use dplyr to filter the gapmider dataset to include only the year 1957 (check above for how we did this for 2007). Save your result as gapminder_1957. Use the ggplot() function and specify the gapminder_1957 dataset as input Add a geom_point() layer to the plot and create a scatter plot showing the GDP per capita gdpPercap on the x-axis and the life expectancy lifeExp on the y-axis Use the color aesthetic to indicate each continent by a different color Use the size aesthetic to adjust the point size by the population pop Use scale_size_area() so that the point sizes reflect the actual population differences and set the max_size of each point to 15 -Set the opacity/transparency of each point to 70% using the alpha=0.7 parameter

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




Q. 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_2007_1957 <- rbind(gapminder_2007, gapminder_1957)
ggplot(gapminder_2007_1957, aes(x=gdpPercap, y=lifeExp, color=continent, size=pop)) +
  geom_point(alpha=0.7) +
  scale_size_area(max_size = 15) +
  facet_wrap(~year)
```


1957
2007
80
70
60
50
40
30
lifeExp
0
30000
60000
90000
gdpPercap
5.00e+08
7.50e+08
1.00e+09
1.25e+09
continent
Africa
Americas
Asia
Europe