Class 5: Data Viz with ggplot

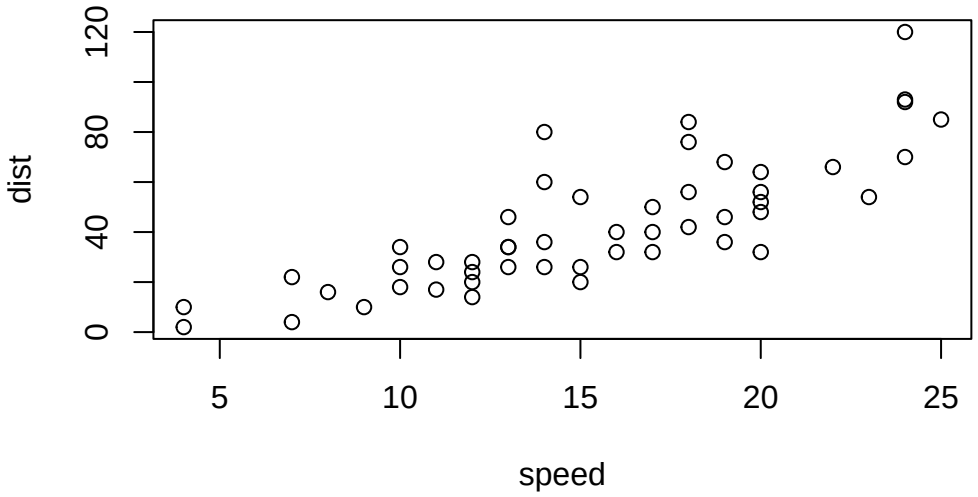
Renny (PID: A98061553)

2024-01-24

Graphics systems in R

There are many graphics systems for R. These include so-called "*base R*" and those in add-on packages like `ggplot2`.

```
plot(cars)
```



How can we make this with `ggplot2`?

This is an add-on package and I first need to install it on my computer.

To install any package I use the `install.packages()` function.

To use it, we need to load up the package from our library of installed packages.

```
library(ggplot2)
```

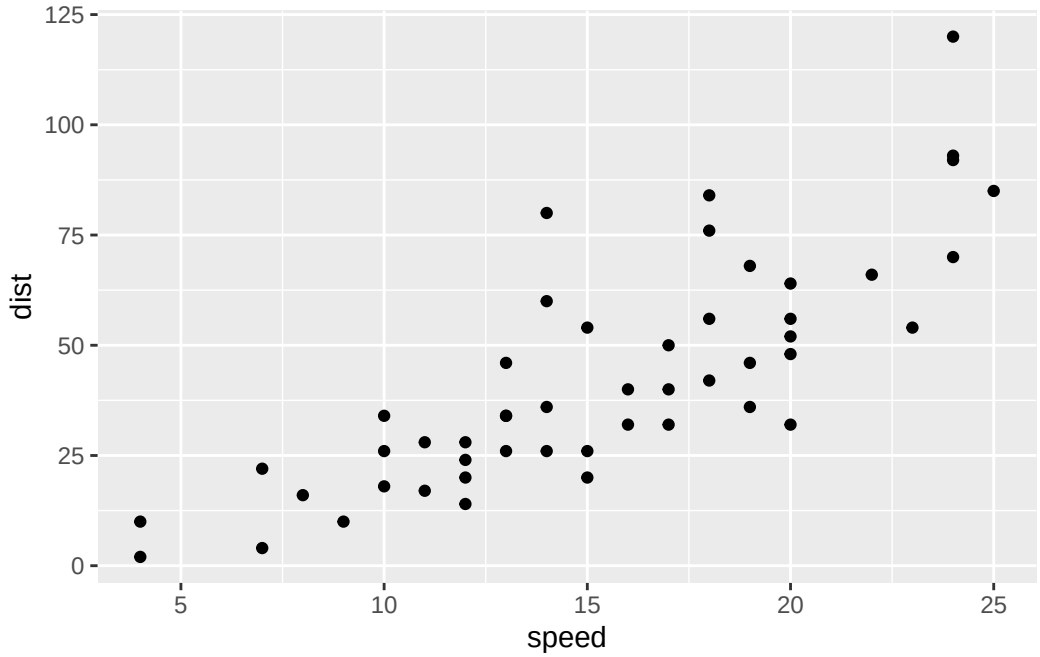
```
ggplot(cars)
```


Using ggplot is not as straightforward as base R plot for basic plots. I have some more typing to do.

Every ggplot has at least three things (layers):

- **data** (data.frame)
- **aes** (how the data maps to the plot)
- **geoms** (think of this as the type of plot, e.g. points, lines, etc.)

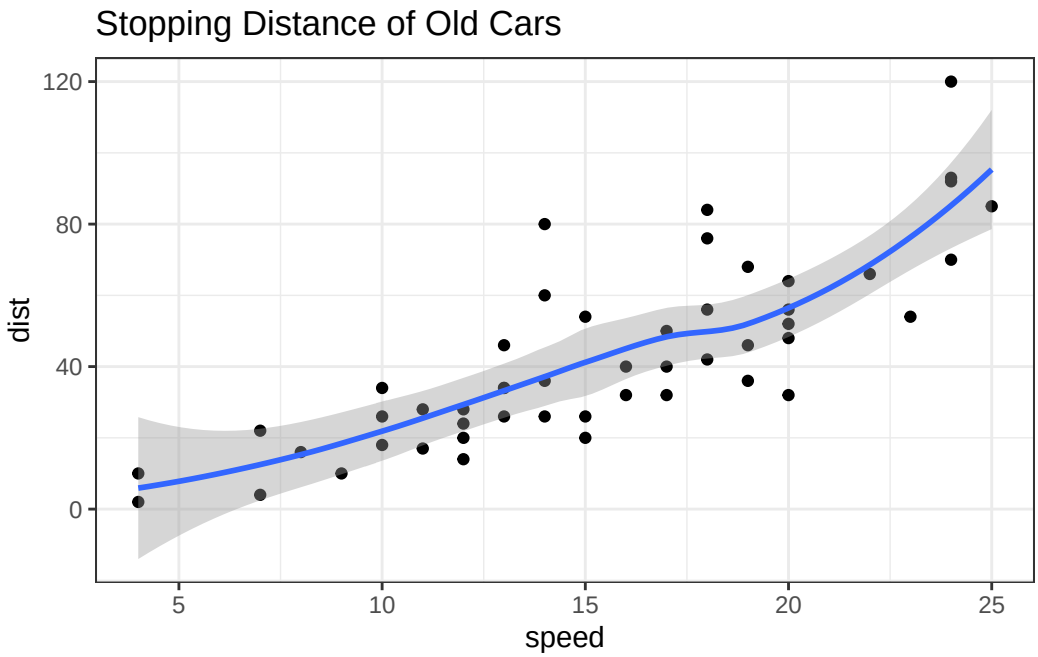
```
ggplot(cars) +
  aes(x=speed, y=dist) +
  geom_point()
```



Here ggplot was more verbose - i.e. I had more typing to do - than base R. However, I can add more layers to make nicer and more complicated plots in an easy way with ggplot.

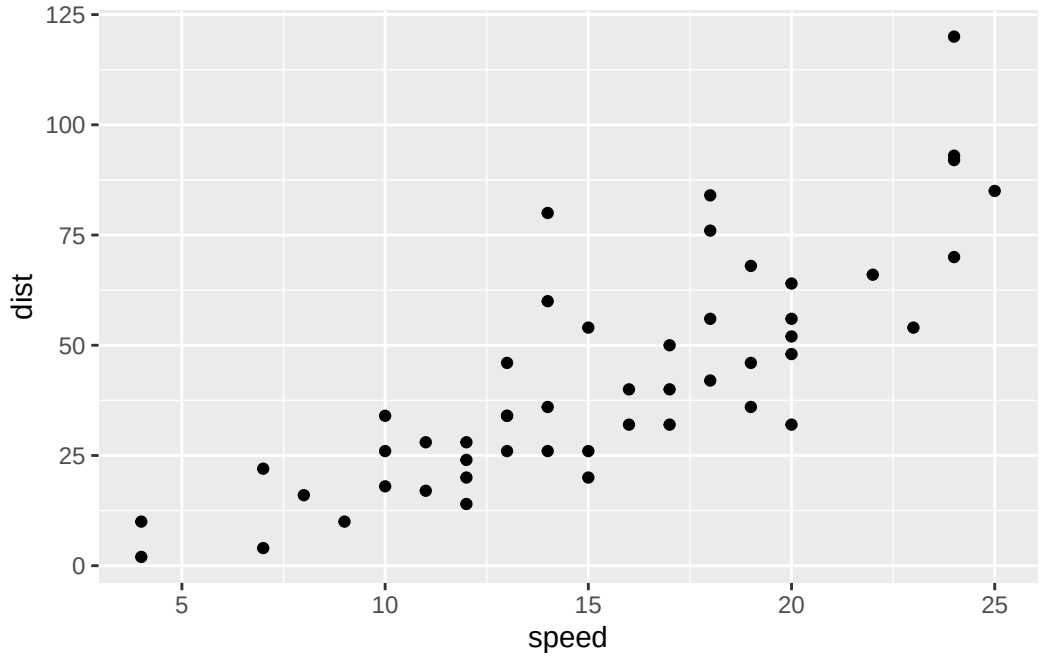
```
ggplot(cars) +
  aes(speed, dist) +
  geom_point() +
  geom_smooth() +
  labs(title="Stopping Distance of Old Cars") +
  theme_bw()
```

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



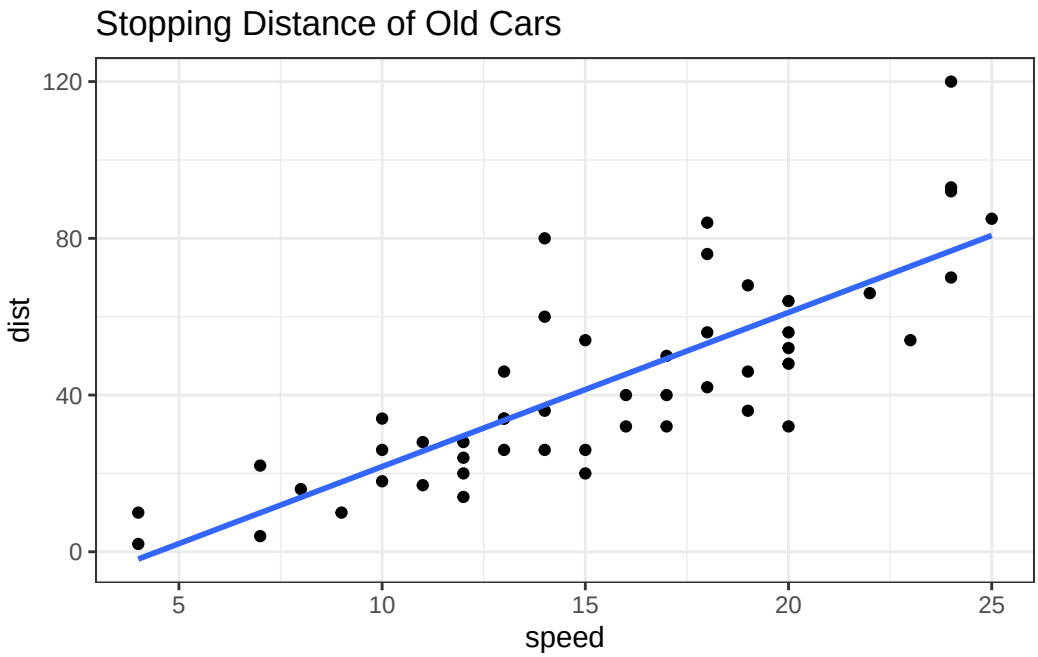

```
pl <- ggplot(cars) +
  aes(x=speed, y=dist) +
  geom_point()
```

```
pl
```




```
ggplot(cars) +
  aes(speed, dist) +
  geom_point() +
  geom_smooth(method=lm, se=FALSE) +
  labs(title="Stopping Distance of Old Cars") +
  theme_bw()
```

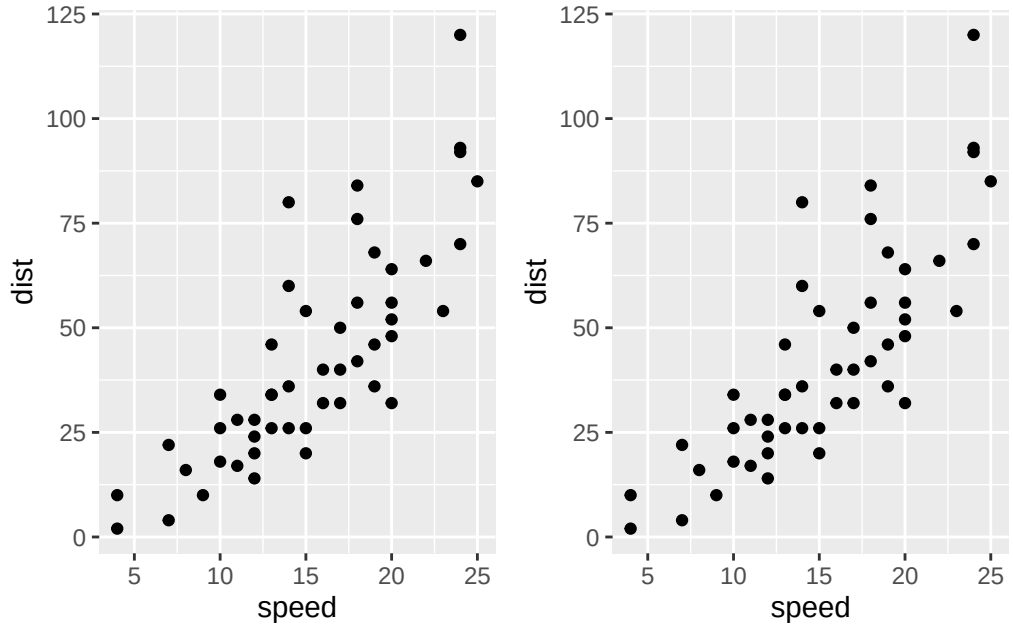
```
`geom_smooth()` using formula = 'y ~ x'
```




```
library(patchwork)
```

Using patchwork with "|" sets panels side by side. Other formats are available.

```
pl | pl
```



Lab Sheet secetion 6 onward

```
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
```

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

```
nrow(genes)
```

```
[1] 5196
```

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"
```

```
ncol(genes)
```

```
[1] 4
```

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?

```
table(genes$S)
```

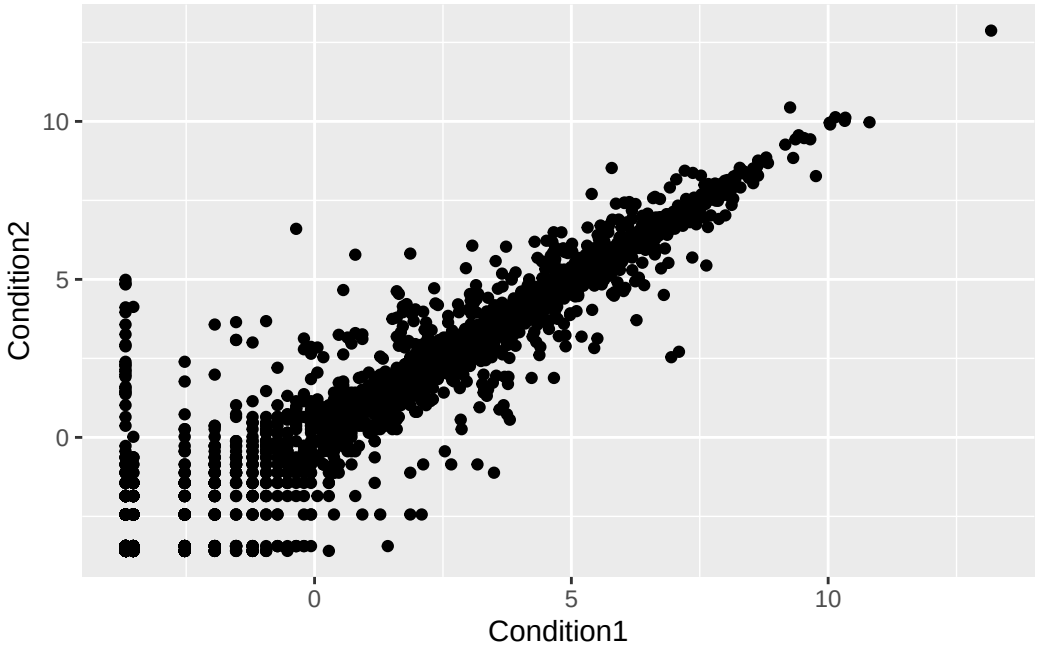
```
      down unchanging         up
        72       4997        127
```

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

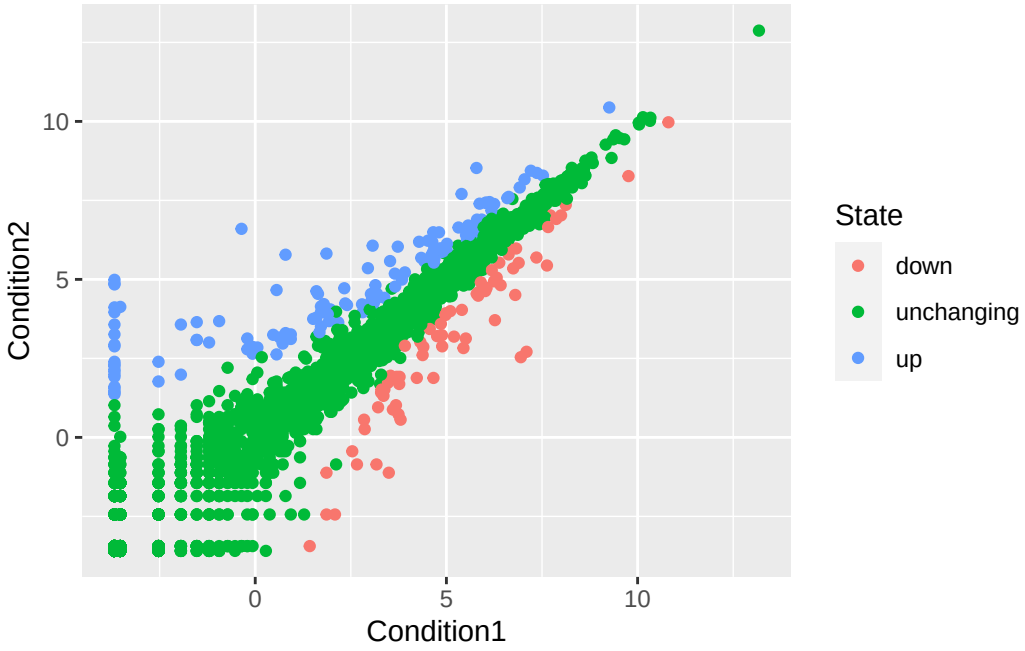
```
round(table(genes$State)/nrow(genes)*100,2)
```

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

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

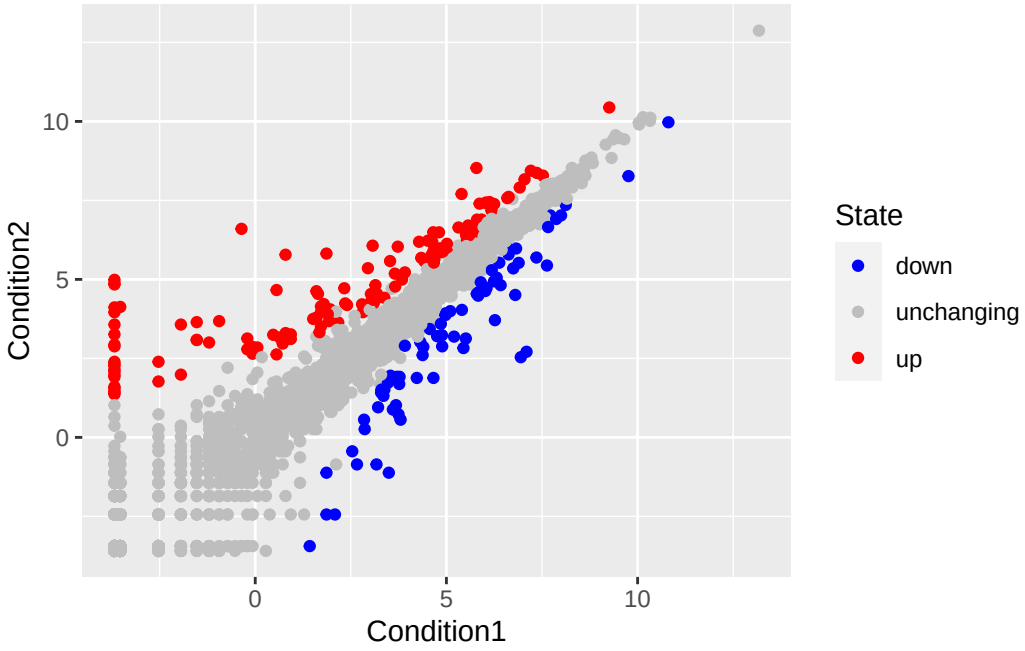



```
ggplot(genes) +
    aes(x=Condition1, y=Condition2, col=State) +
    geom_point()
```

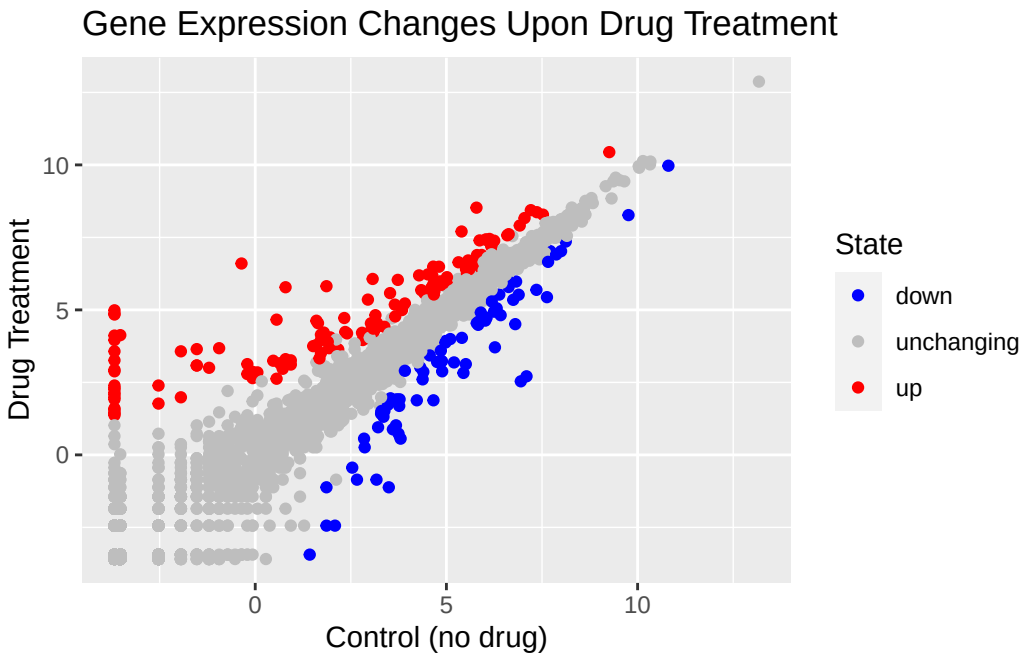



```
p <-ggplot(genes) +
    aes(x=Condition1, y=Condition2, col=State) +
    geom_point()
```

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




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



Inetarction version with `plotly`

```
library(plotly)
#ggplotly(p)
#Note that this is incompatible with PDF; plotly animation is only compatible with HTML.
#For PDF purposes, I will not plot this graph with plotly.
```

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

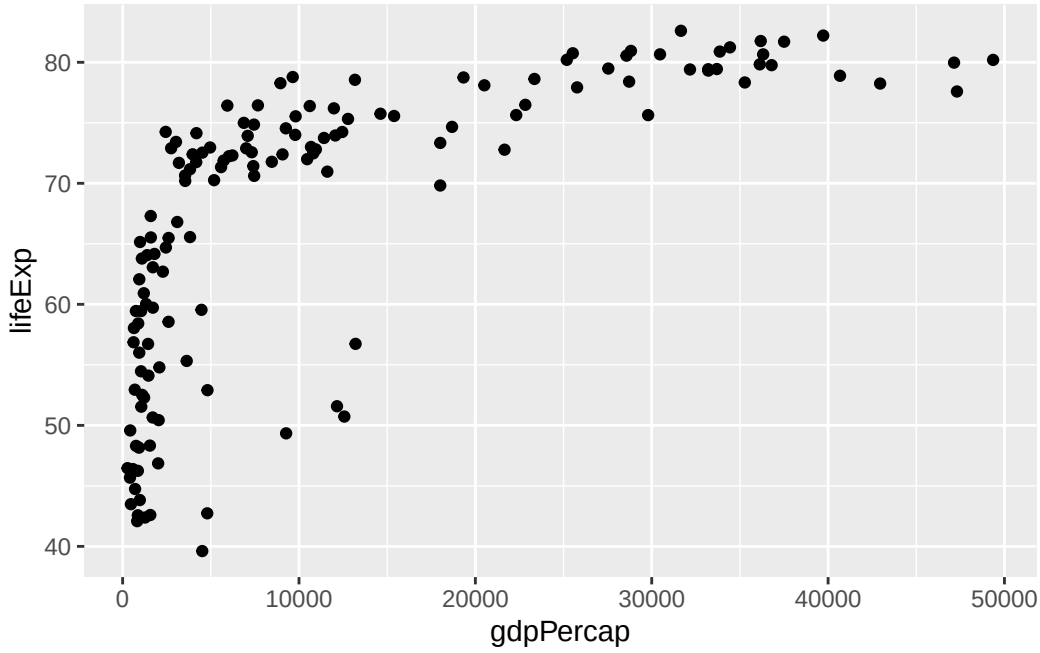
```
# install.packages("dplyr")  ## un-comment to install if needed
library(dplyr)
```

```
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
```

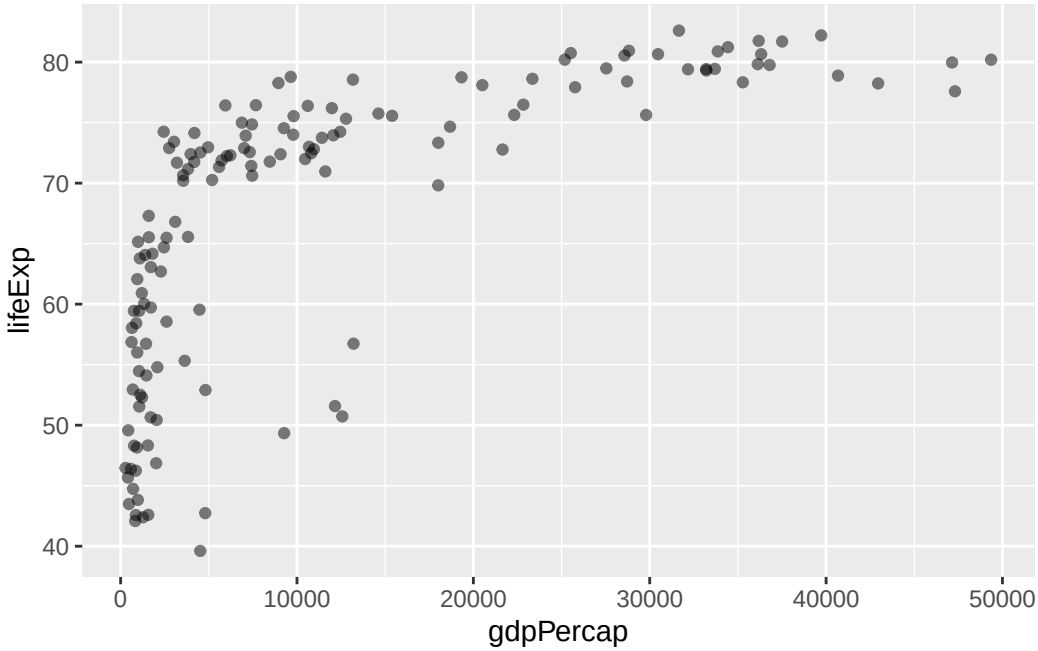
```
gapminder_2007 <- gapminder %>% filter(year==2007)
```

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



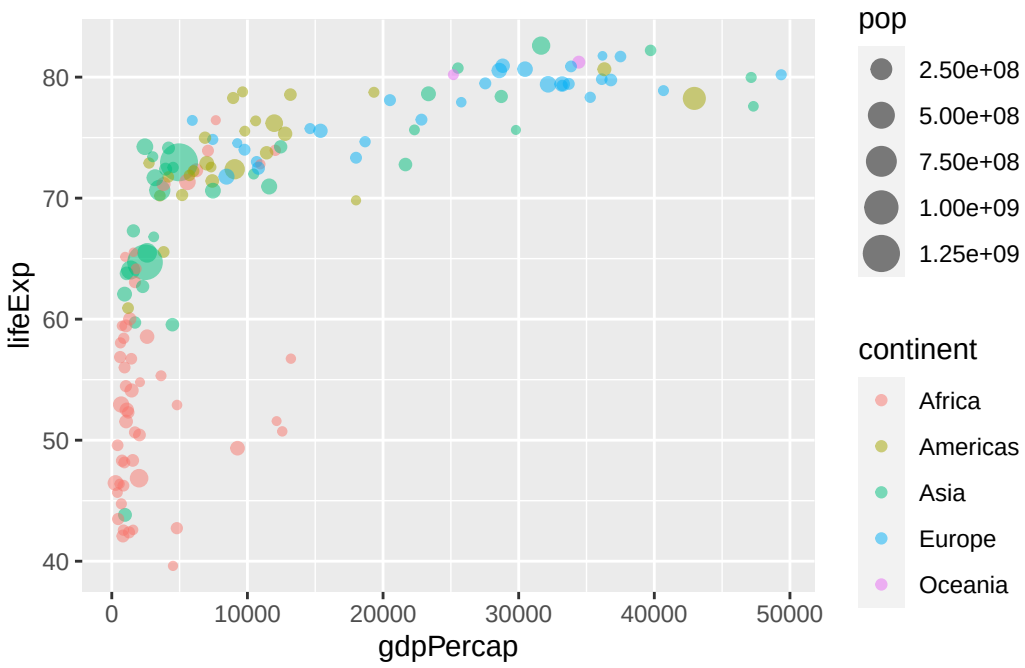
Adjusting alpha allows transparency to change, so that overplotted points do not occlude one another.

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



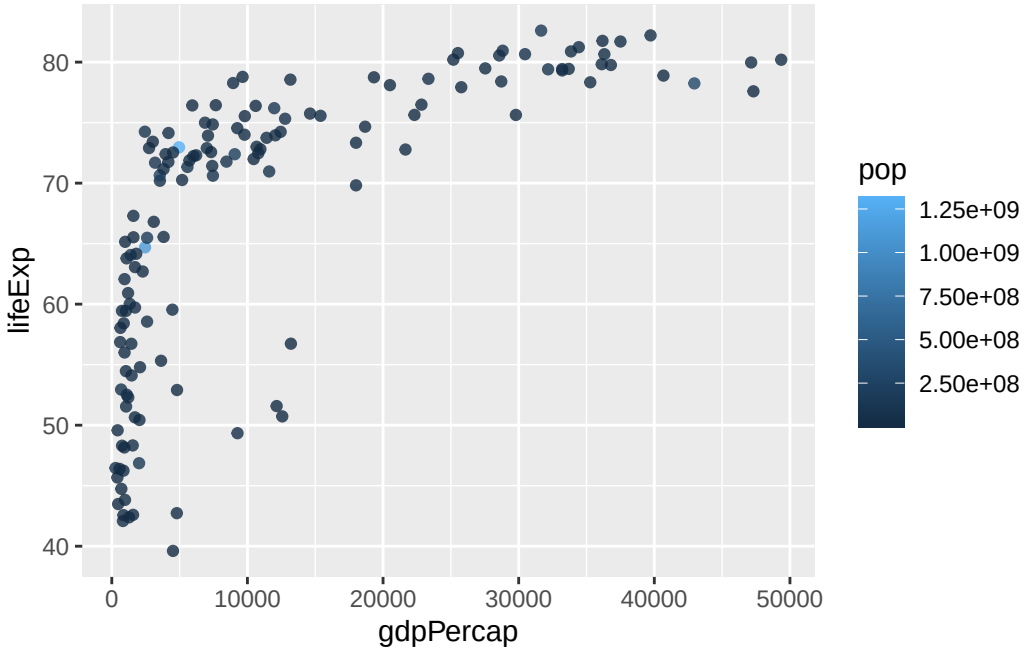
Map the continent variable to the point color aesthetic Map the population pop (in millions) through the point size argument to aes():

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



By contrast:the plot looks like if points were colored by the numeric variable population pop:

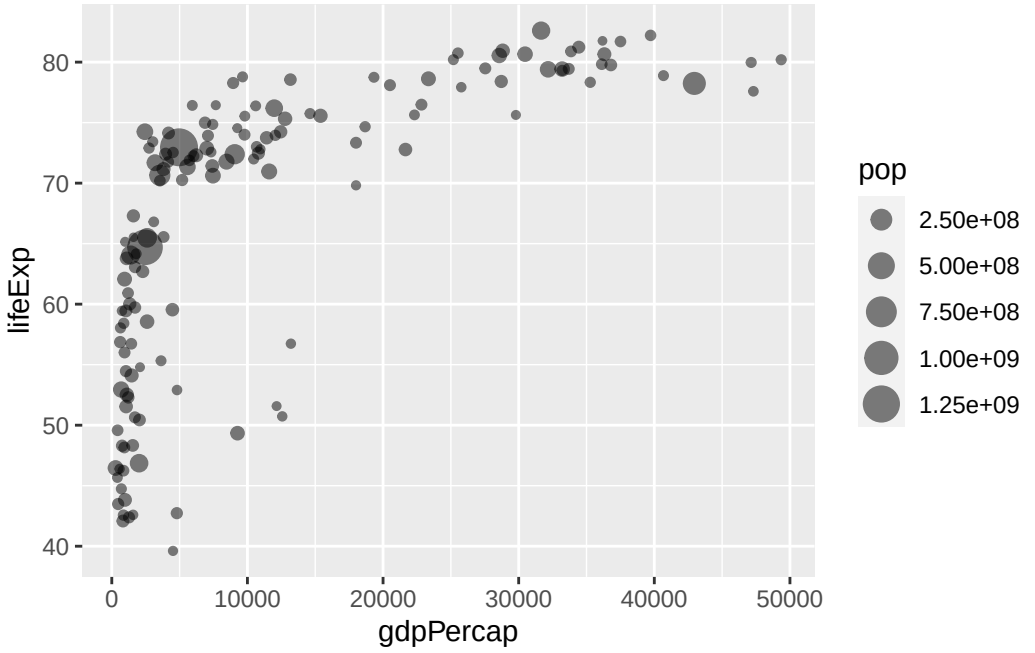
```
ggplot(gapminder_2007) +
  aes(x = gdpPercap, y = lifeExp, color = pop) +
  geom_point(alpha=0.8)
```



For the gapminder_2007 dataset:

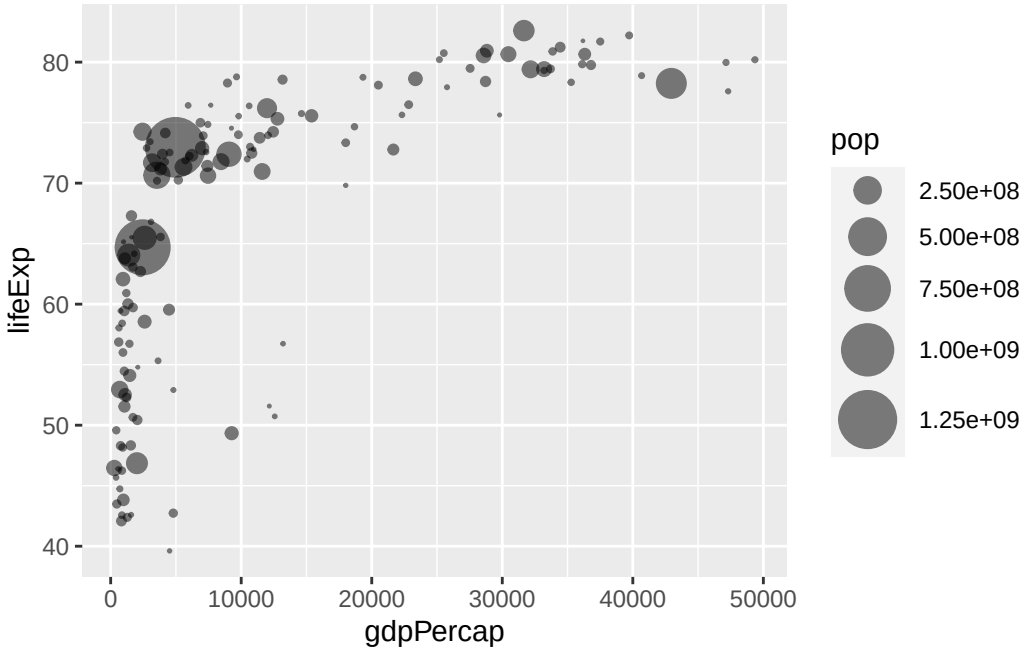
- Plot the GDP per capita (x=gdpPercap) vs. the life expectancy (y=lifeExp)
- Set the point size based on the population (size=pop) of each country

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



From above: the point sizes are binned by default. To reflect the actual population differences by the point size we can use the scale_size_area() function instead. The scaling information can be added like any other ggplot object with the + operator:

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



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?

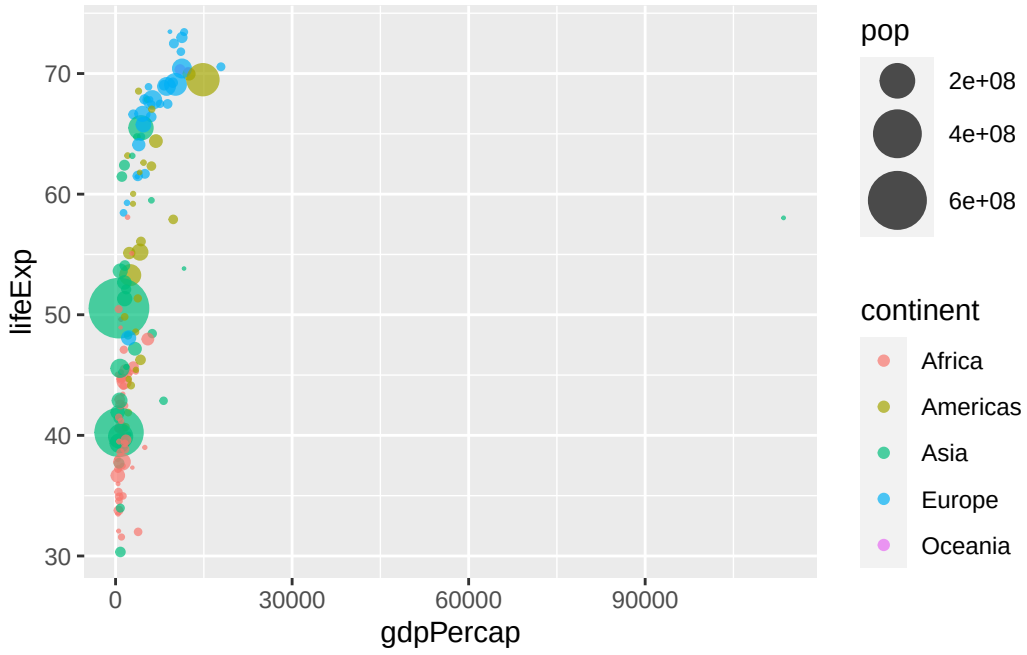
Steps to produce your 1957 plot should include:

- Use dplyr to `filter` the `gapminder` 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

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

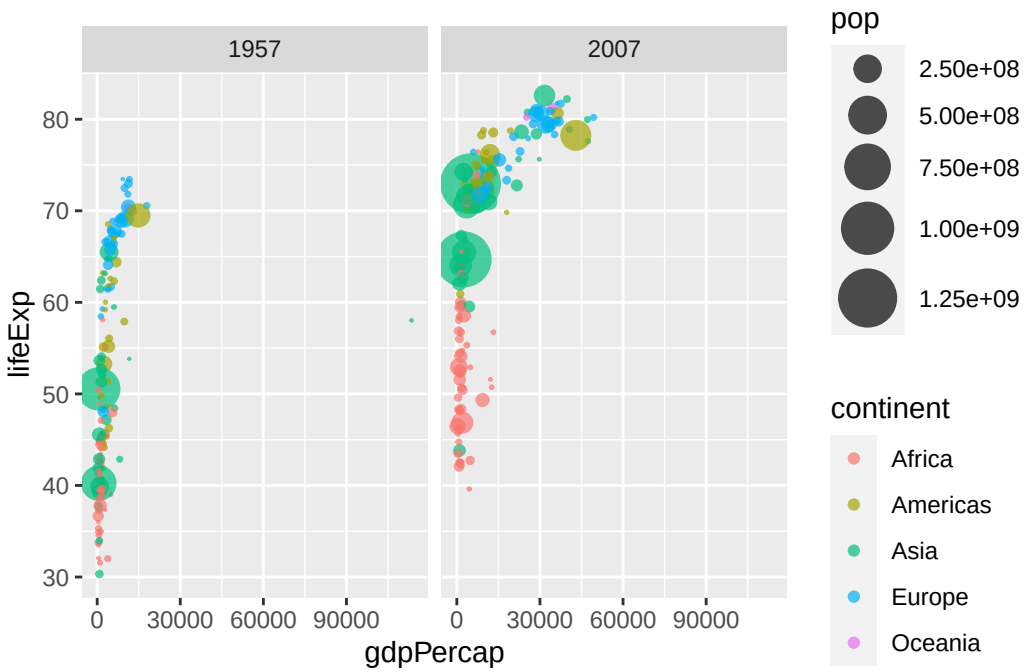
```
ggplot(data=gapminder_1957) +
  aes(x=gdpPercap, y=lifeExp, color=continent,size=pop) +
```

```
  geom_point(alpha=0.7)+
  scale_size_area(max_size = 10)
```



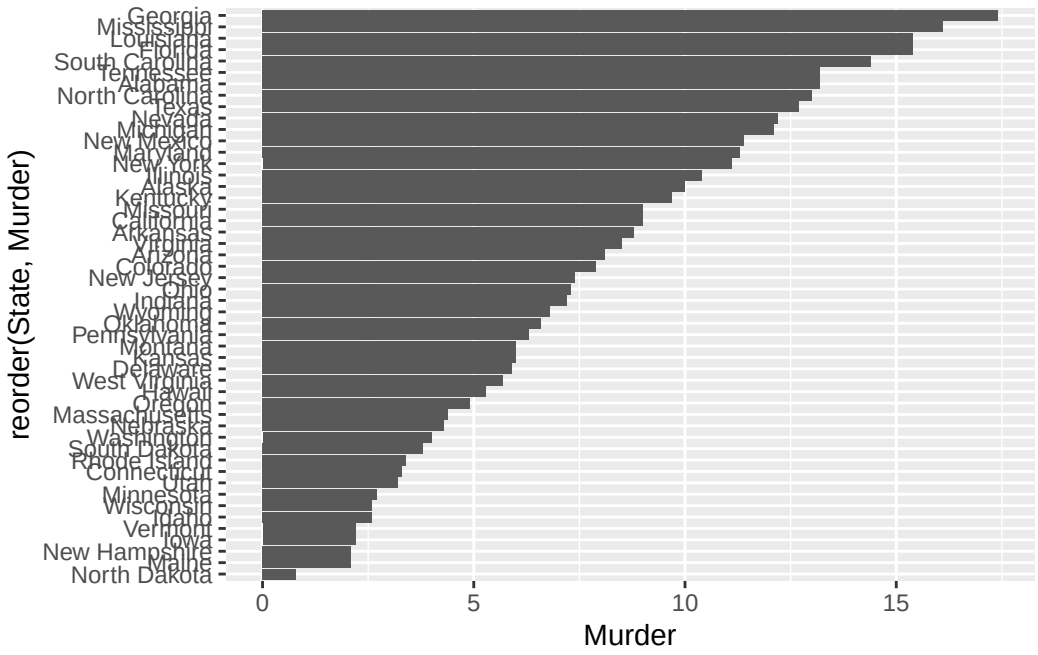

```
gapminder_1957_2007 <- gapminder %>% filter(year==1957 | year==2007)
```

```
ggplot(gapminder_1957_2007) +
  geom_point(aes(x = gdpPercap, y = lifeExp, color=continent,
                 size = pop), alpha=0.7) +
  scale_size_area(max_size = 10) +
  facet_wrap(~year)
```

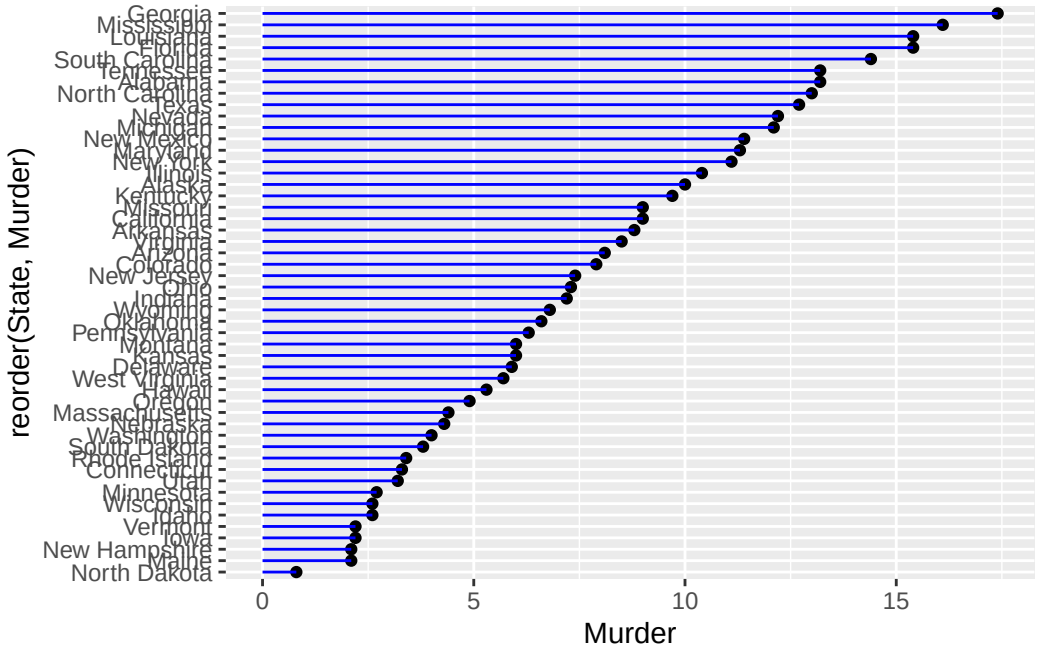


When making Bar Graphs: use geom_col(). The other option (geom_bar()) processes the data in additional aways.

```
USArrests$State <- rownames(USArrests)
ggplot(USArrests) +
  aes(x=reorder(State,Murder), y=Murder) +
  geom_col() +
  coord_flip()
```




```
ggplot(USArrests) +
  aes(x=reorder(State,Murder), y=Murder) +
  geom_point() +
  geom_segment(aes(x=State,
                   xend=State,
                   y=0,
                   yend=Murder), color="blue") +
  coord_flip()
```


reorder(State, Murder)
Georgia
Mississippi
Louisiana
Florida
South Carolina
Tennessee
Alabama
North Carolina
Texas
Nevada
Michigan
New Mexico
Maryland
New York
Illinois
Alaska
Kentucky
Missouri
California
Arkansas
Virginia
Arizona
Colorado
New Jersey
Ohio
Indiana
Wyoming
Oklahoma
Pennsylvania
Montana
Kansas
Delaware
West Virginia
Hawaii
Oregon
Massachusetts
Nebraska
Washington
South Dakota
Rhode Island
Connecticut
Utah
Minnesota
Wisconsin
Idaho
Vermont
Iowa
New Hampshire
Maine
North Dakota
0
5
10
15
Murder