# Data Analytics
## CS301

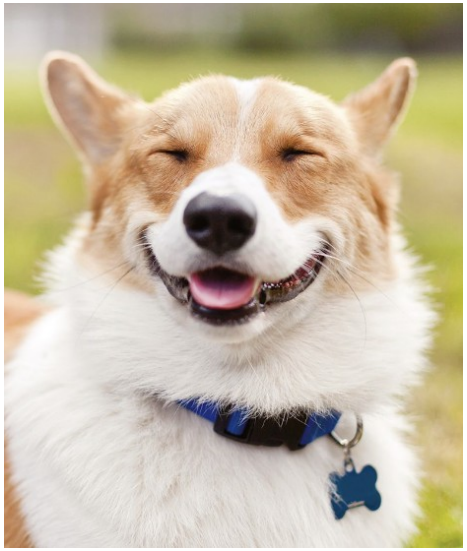
### Exploratory First Steps

Week 5
Spring 2023
Oliver BONHAM-CARTER

# Getting, Printing User Input

- Get some user input:
  - `myVar = readline(prompt = "Enter something :")`
- Print the input
  - `cat("You entered : ",myVar, "\n") #print the output, then drop a line.`
- Find out what kind of data is stored in the variable
  - `typeof(myVar)`

*Typically, many of the inputs are hard-coded into scripts*



# Tidyverse and Ggplot


https://tidyverse.tidyverse.org/


https://ggplot2.tidyverse.org/

```
# Try loading the tidyverse library
library(tidyverse)

#If errors or not working, then try the following instead
library(ggplot2)
```

# Ask Data about Mileage

Ask: *What classes of cars (i.e,. suv's, trucks, etc.) get the best city and highway mileage?*

I know! I will use some MPG data from the Tidyverse library and see what the data says!!

```
library(tidyverse) #not working? Try library(ggplot2)
   # check the data
View(mpg) # note the capital 'V'
   # run simple plot
ggplot(mpg) +
  geom_point(mapping = aes(x = mpg$displ, y = mpg$hwy ))
```

# From Last Time: Code for a Simple GGPlot

- **ggplot(*data* = mpg) + geom_point(mapping = aes(x = displ, y = hwy))**

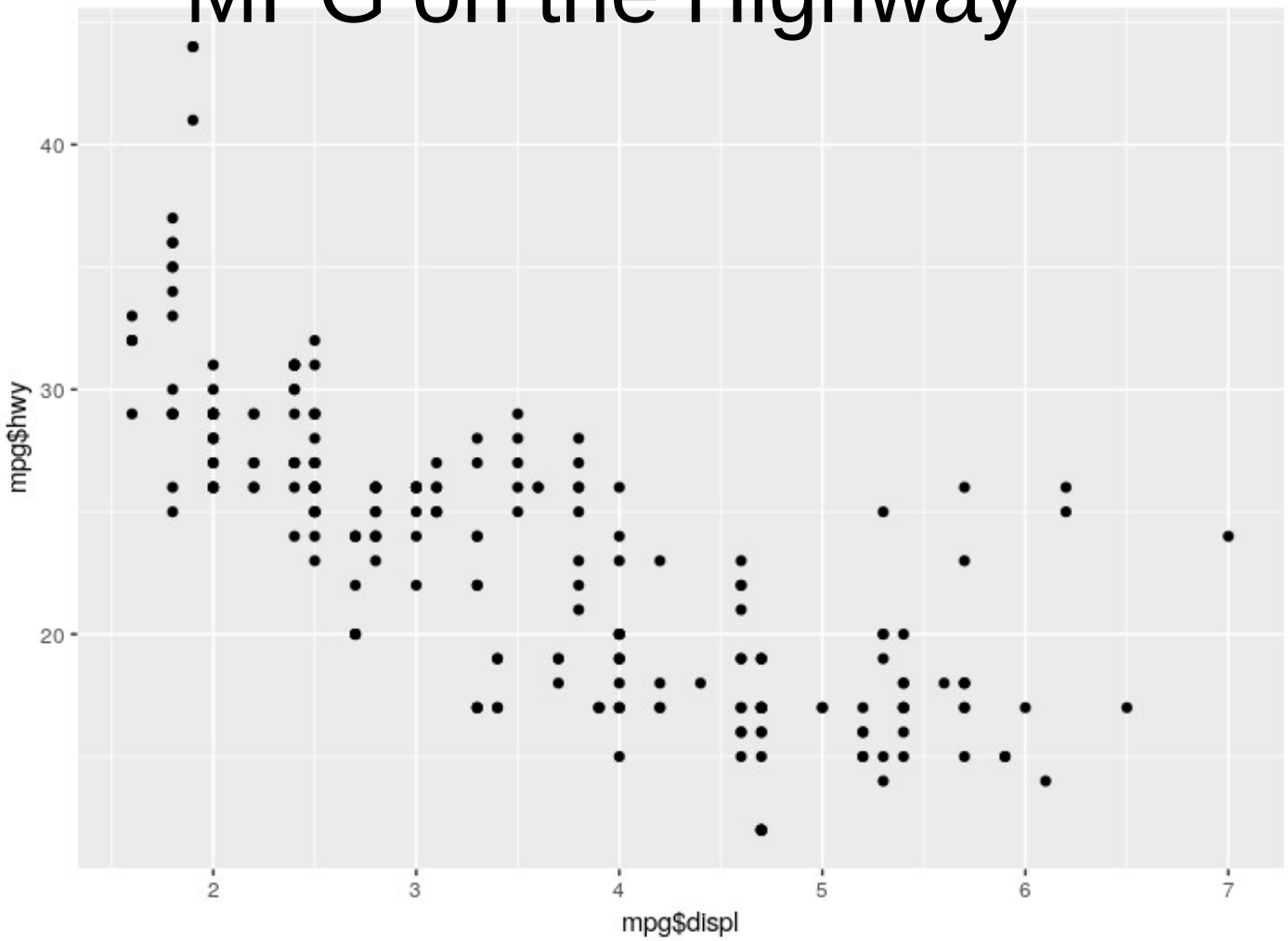
- Establish the *canvas* (where the plot is shown)
- **ggplot()**

- Link to the data (set is called, 'mpg'). We do not need to add "`data =`"
  - **ggplot(data = mpg)**

- Compute the geometery of point placement on canvas
  - **geom_point(mapping = … )**

- Compute the aesthetics of the plot (titles, color, point type, etc)
  - **aes(x = displ, y = hwy)**

# Displacement (Car Weight) by MPG on the Highway

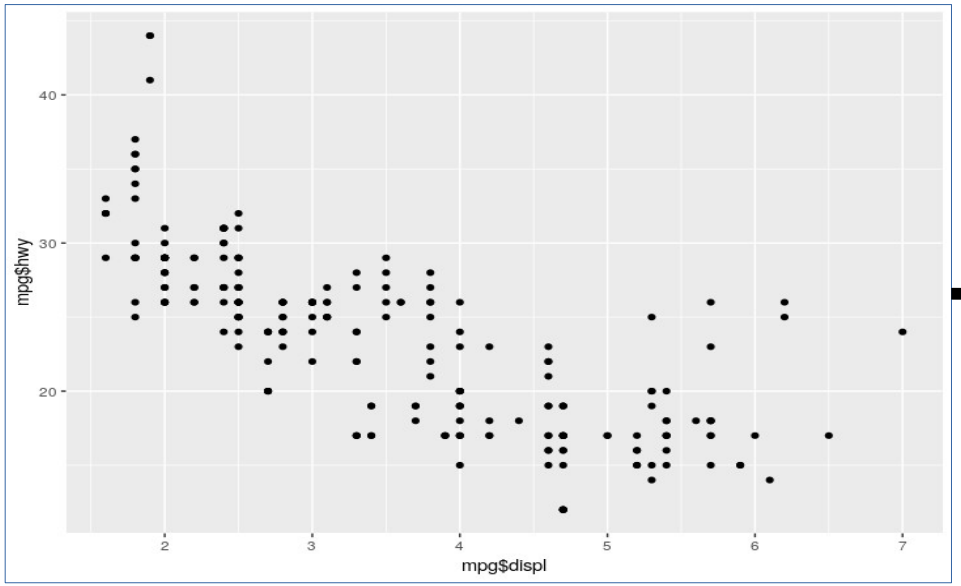



```
ggplot(mpg) +  geom_point(mapping = aes(x = displ, y = hwy ))
```

# Displacement (Car Weight) by MPG on the Highway

Is there more to learn from this data?

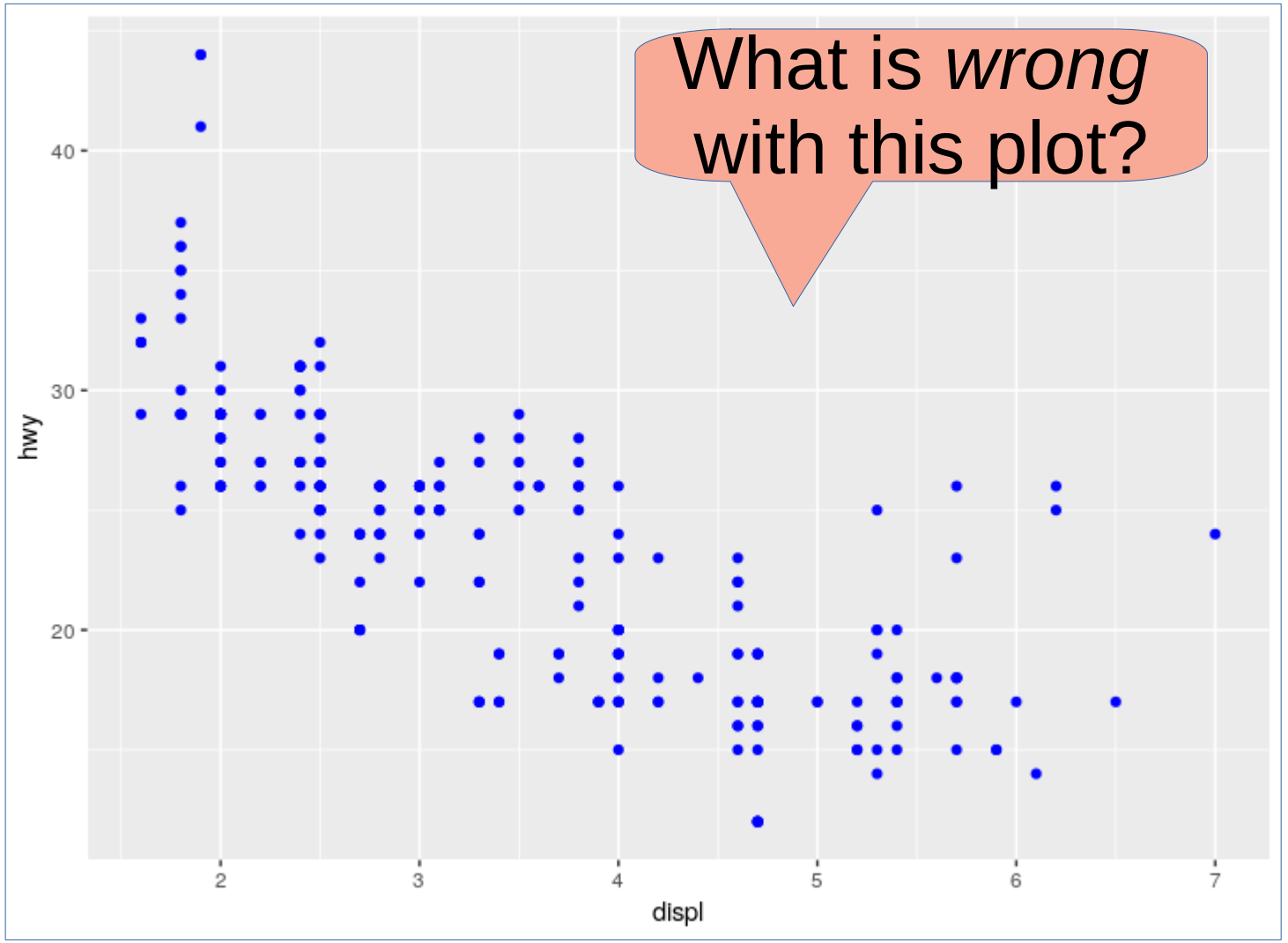
What is *wrong* with this the previous plot?




Yes!

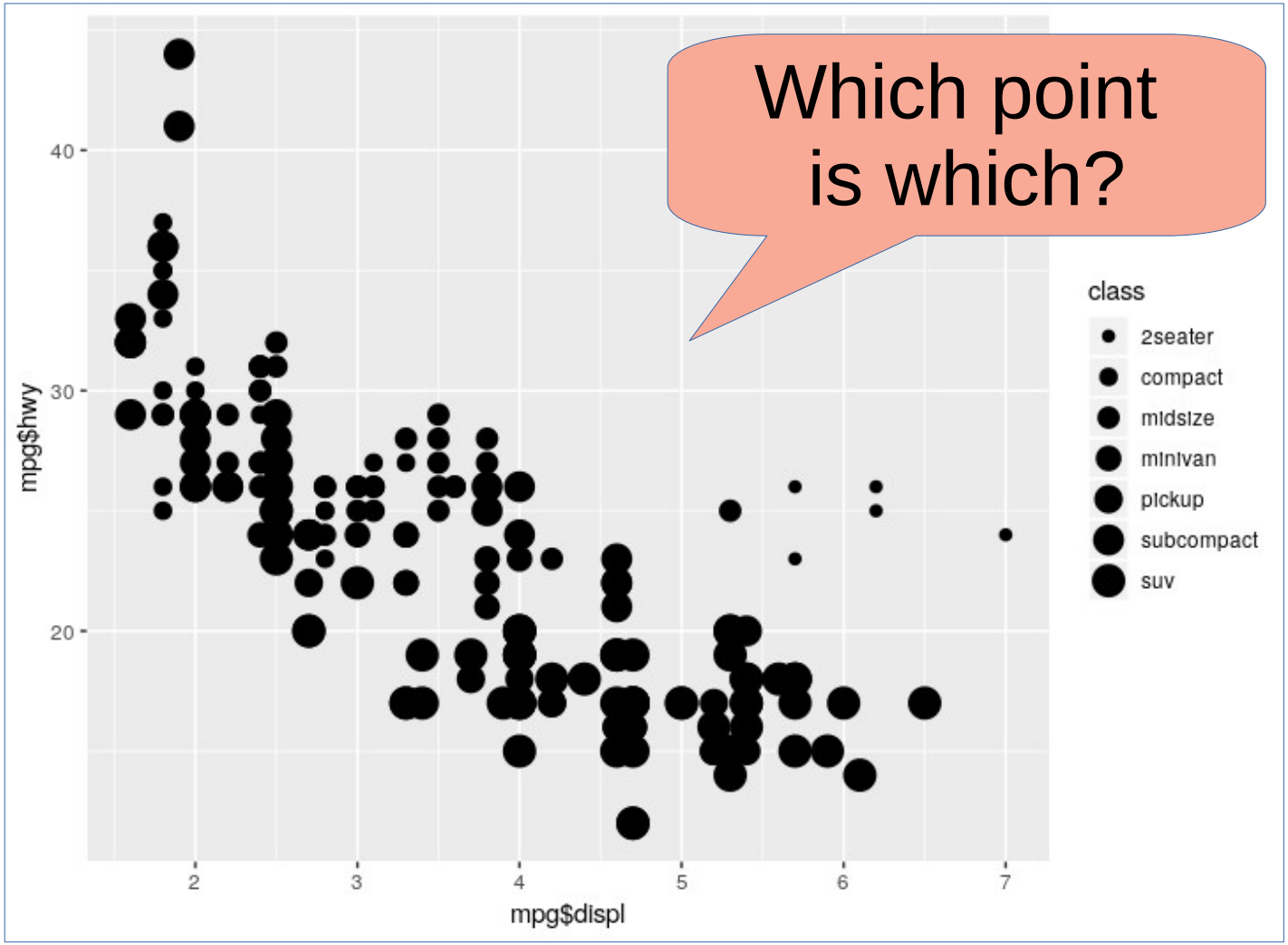
No??

# New Blue Plot?



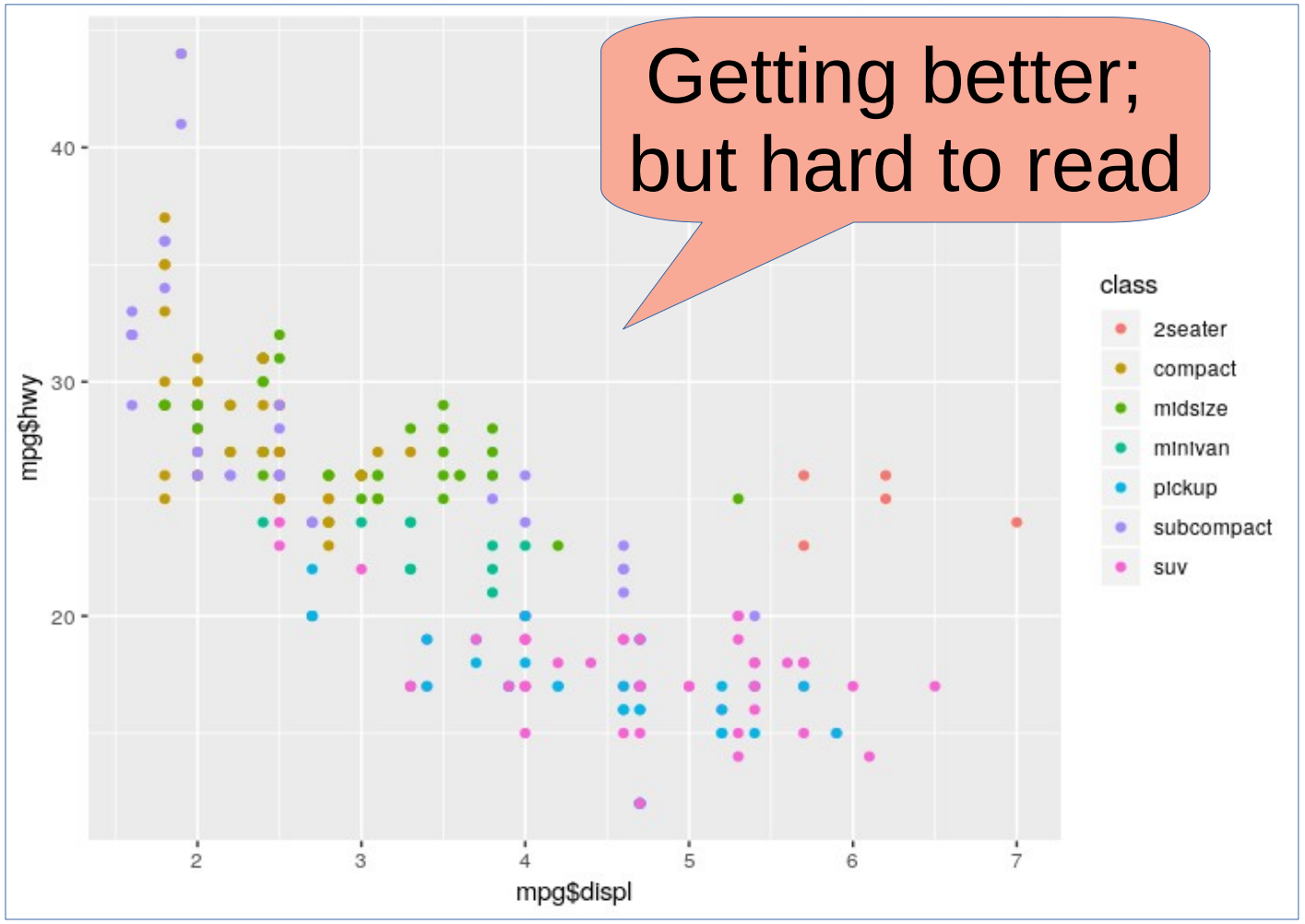

```
ggplot(mpg) + geom_point(mapping = aes(x = displ, y = hwy), color = "blue")
```

# Try Sizing the points for Dimension



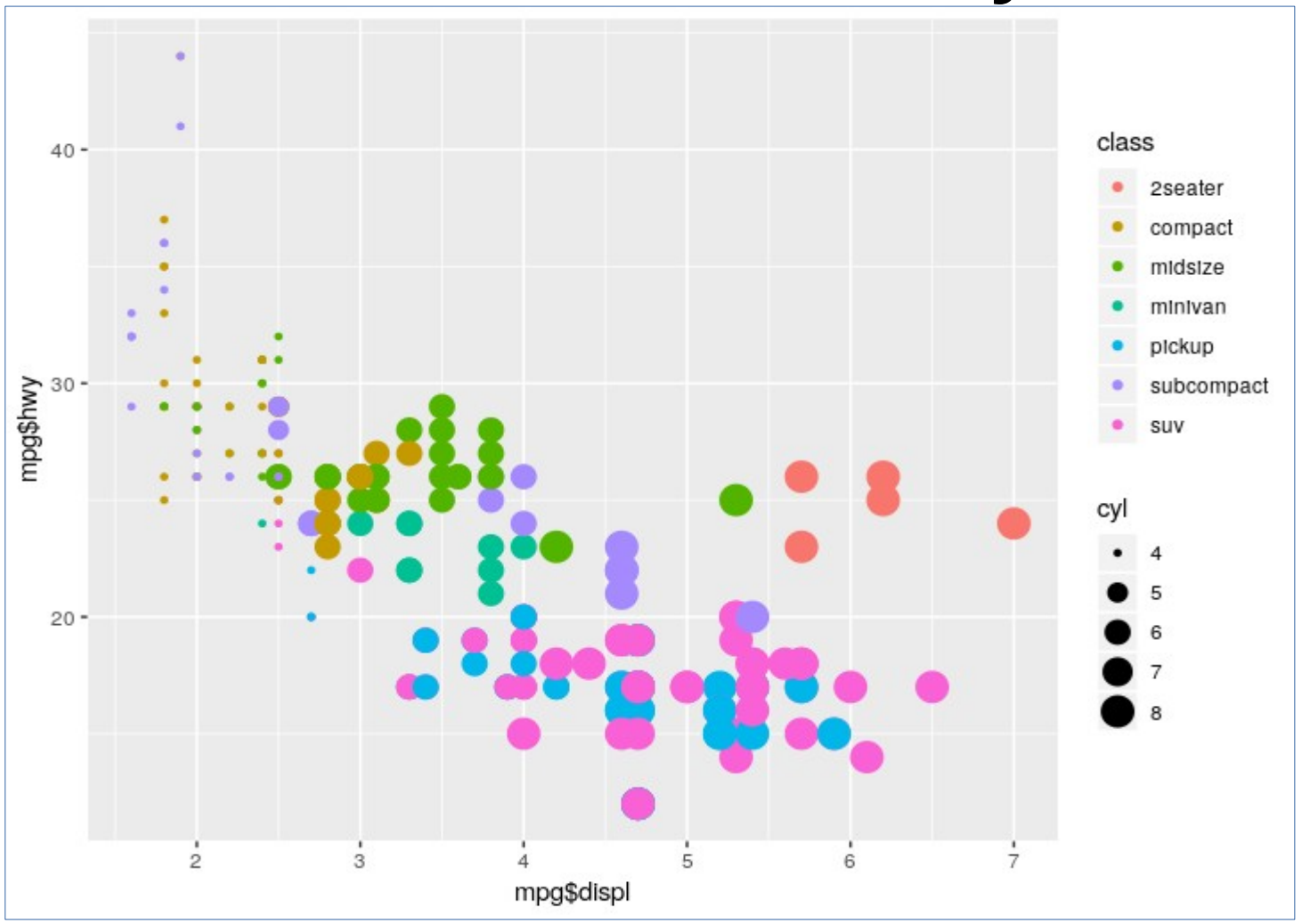

```
ggplot(mpg) + geom_point(mapping = aes(x = displ, y = hwy, size = class))
```

# Try Coloring for Dimension



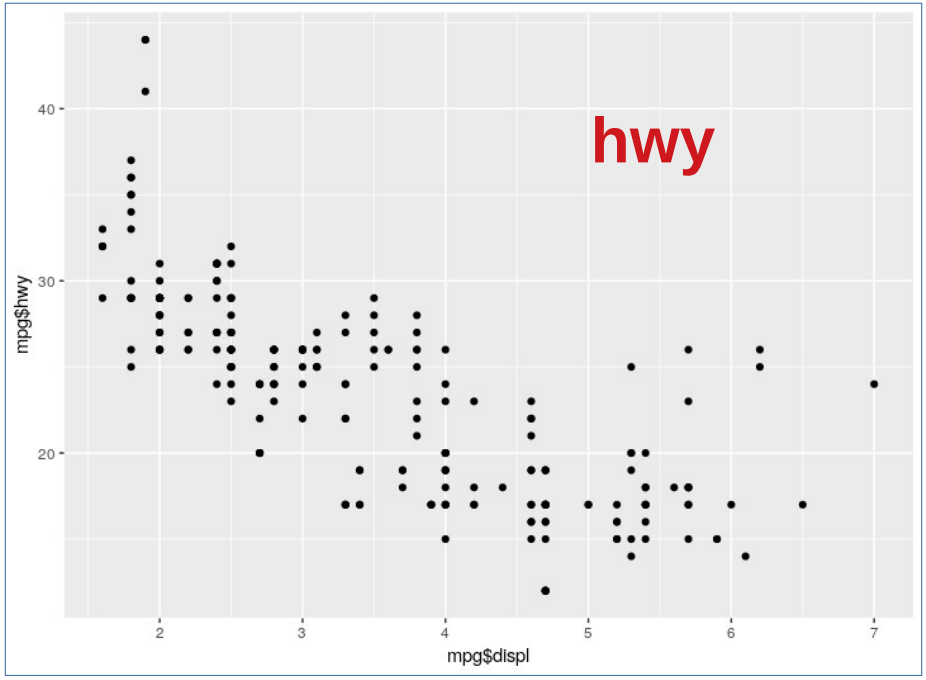

```
ggplot(mpg) + geom_point(mapping = aes(x = displ, y = hwy, color = class))
```

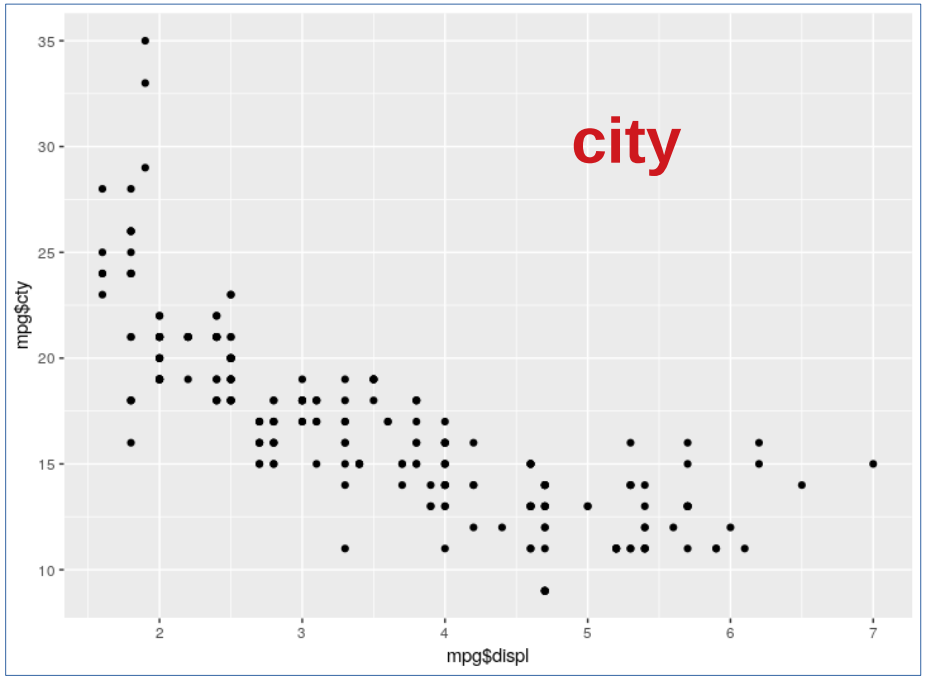
# Combine Color, Sized Points and Cycle




```
ggplot(mpg) + geom_point(mapping = aes(x = displ, y = hwy, color = class, size = cyl))
```

# Comparing Hwy and City Mileage

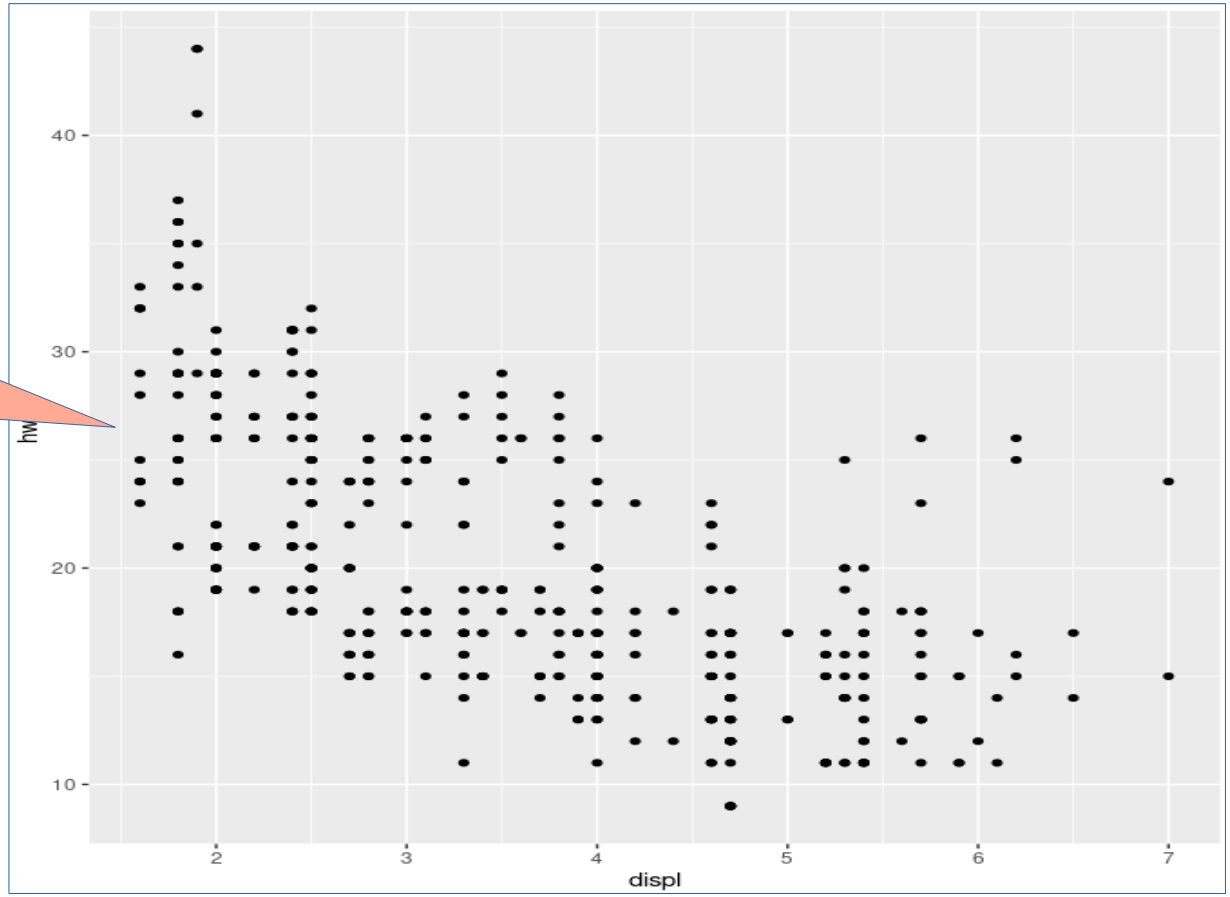



```
# hwy mileage
ggplot(mpg) + geom_point(mapping = aes(x = displ, y = hwy ))

# city mileage
ggplot(mpg) + geom_point(mapping = aes(x = displ, y = cty ))
```

# Comparing Hwy and City Mileage

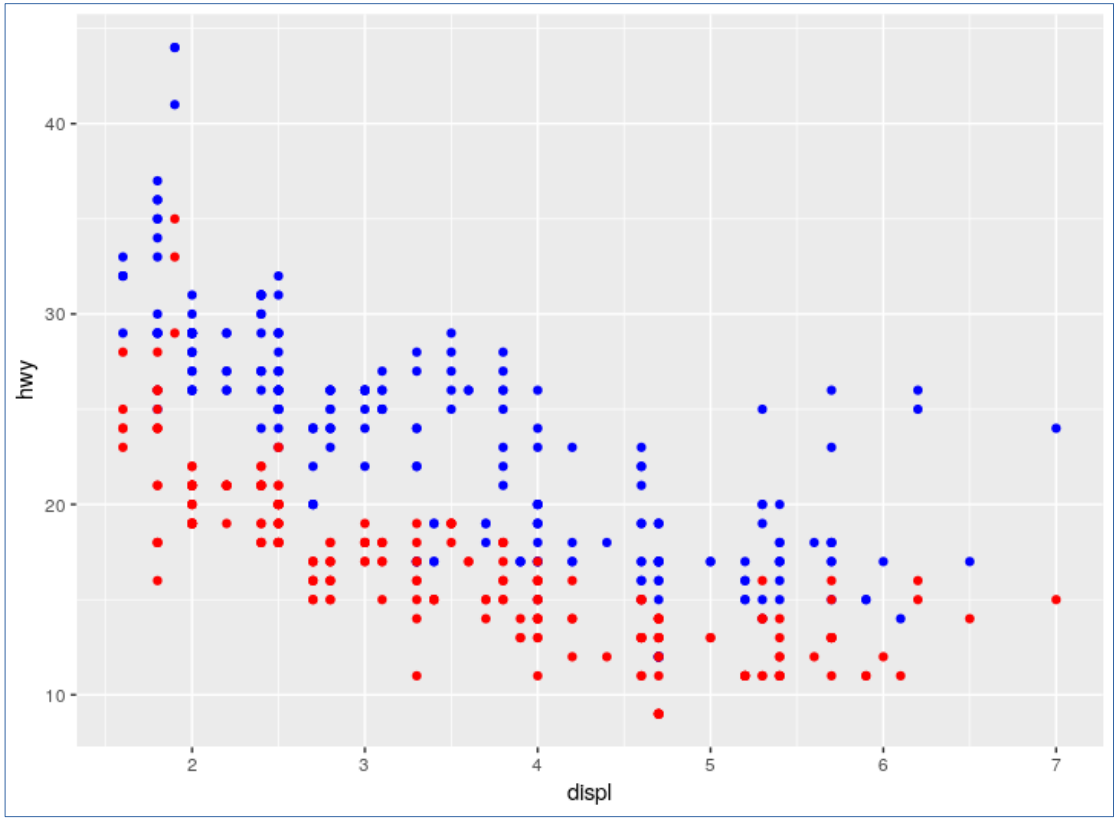
Which Is which?




#Place **hwy and cty** mileage together in same plot

```
ggplot(mpg) + geom_point(mapping = aes(x = displ, y = hwy), ) +
geom_point(mapping = aes(x = displ, y = cty))
```

# Comparing Hwy and City Mileage

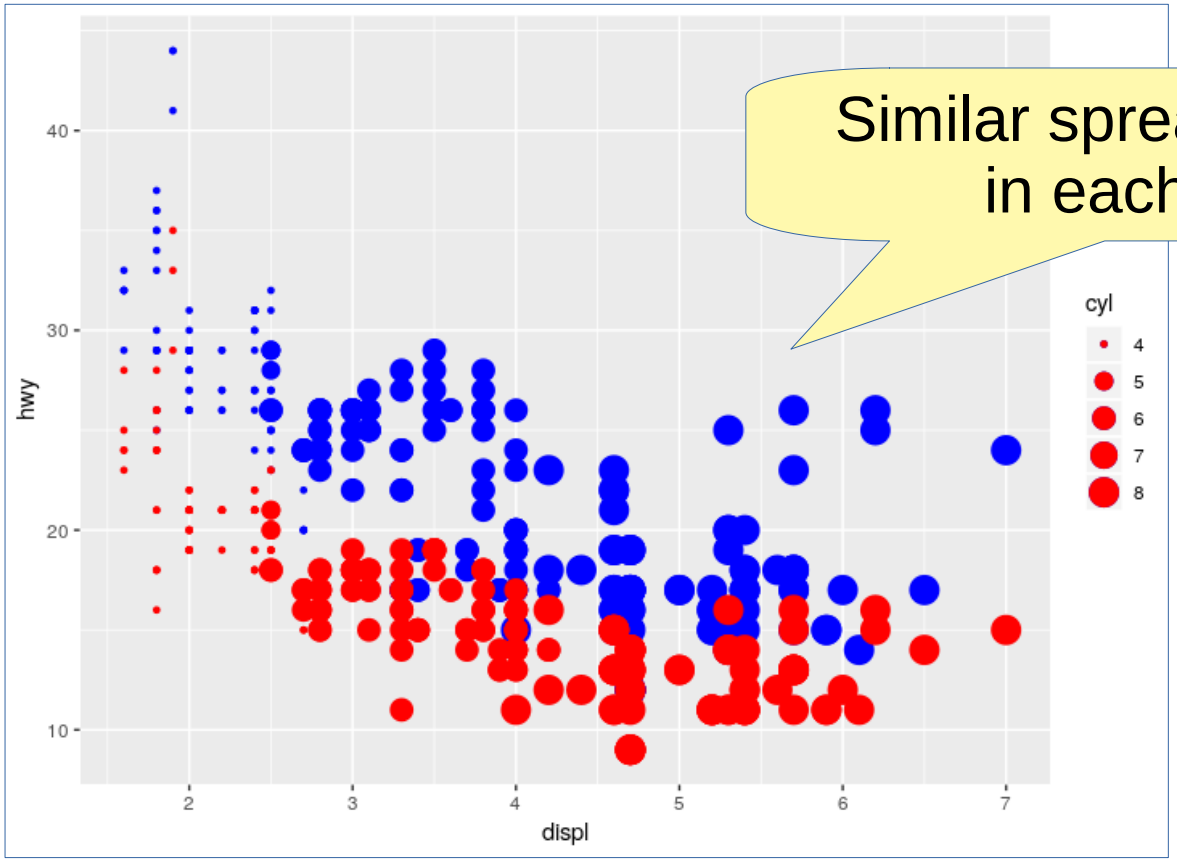



```
#Place hwy and cty mileage together in same plot
ggplot(mpg) +
  geom_point(mapping = aes(x = displ, y = hwy), color = "blue") +
  geom_point(mapping = aes(x = displ, y = cty),   color= "red")
```

# Add Sized Points

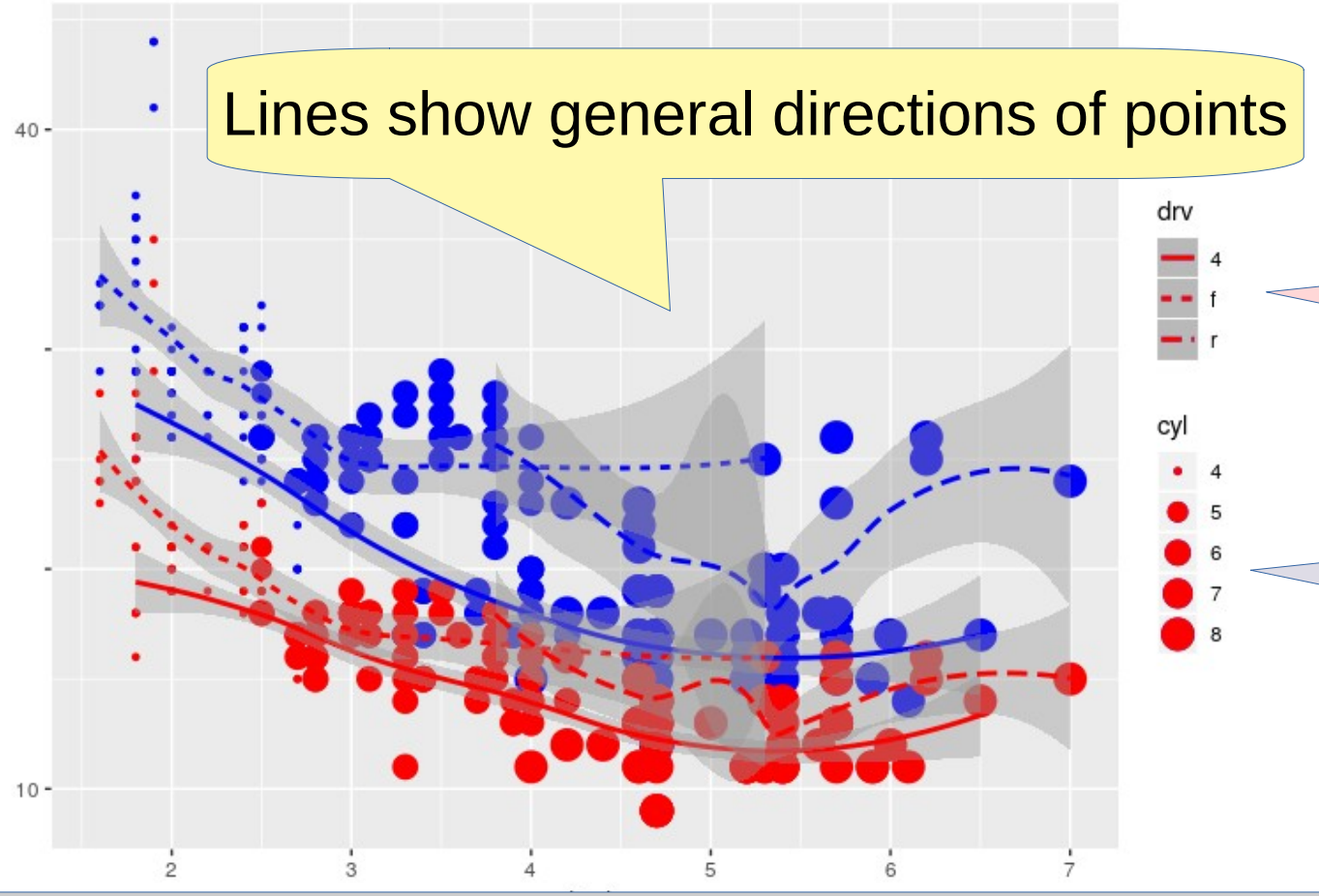
**Hwy**

**City**




```
ggplot(mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, size = cyl), color = "blue") +
  geom_point(mapping = aes(x = displ, y = cty, size = cyl), color= "red")
```

# Add a *Smooth-Line*

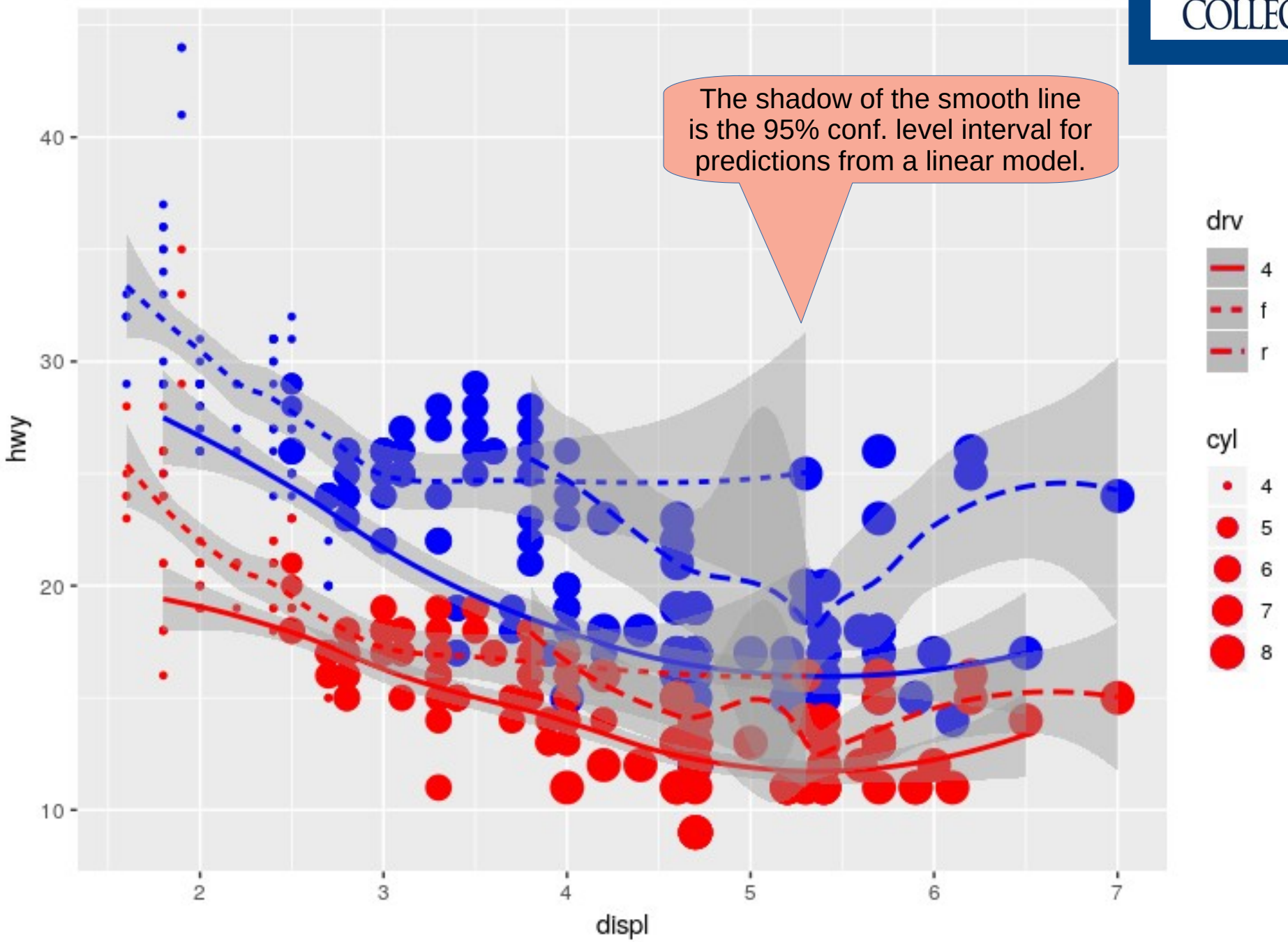



**Line**: Drv type

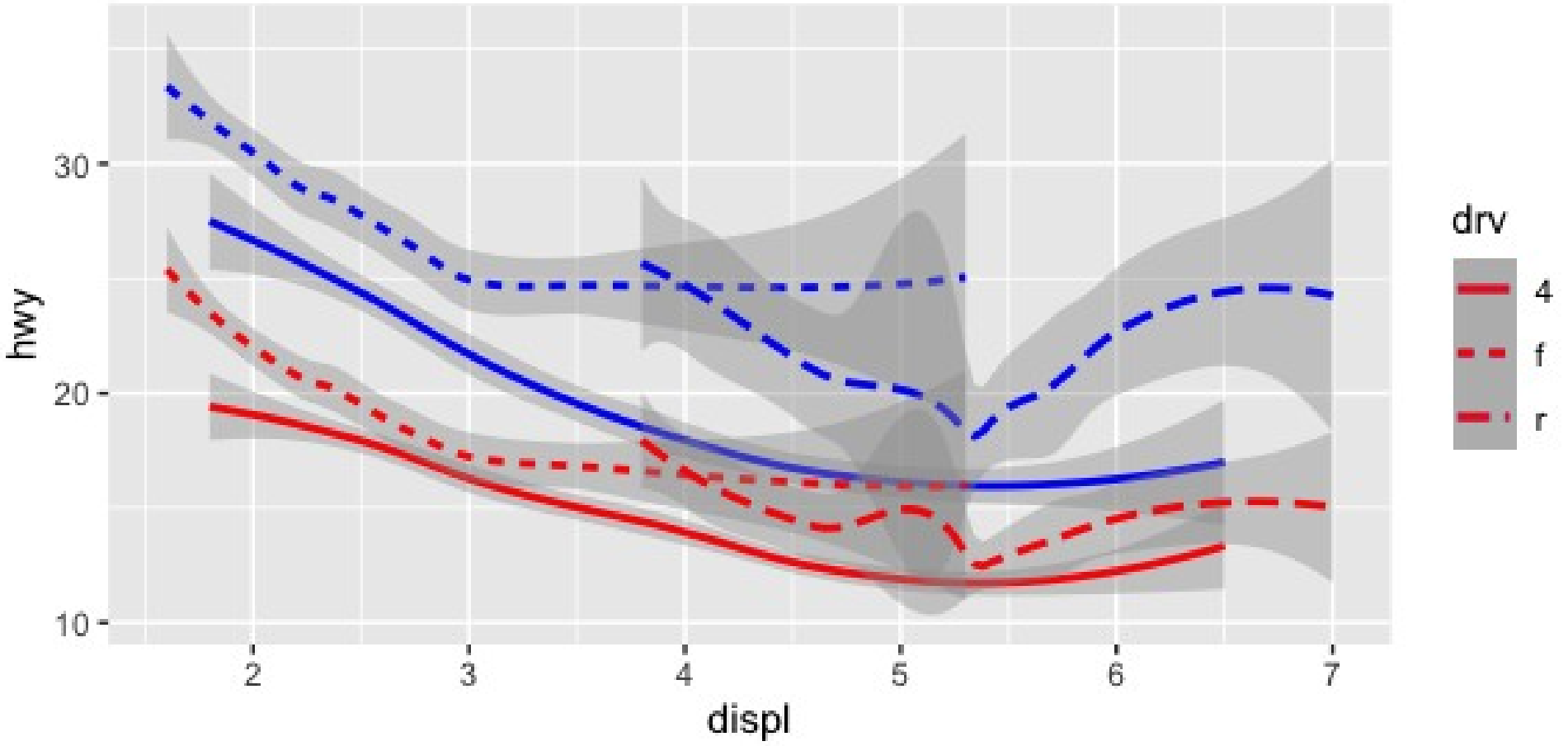
**Size**: Cyl type

```
ggplot(mpg) +
geom_point(mapping = aes(x = displ, y = hwy, size = cyl), color = "blue") +
geom_point(mapping = aes(x = displ, y = cty, size = cyl), color="red") +
geom_smooth(mapping = aes(x = displ, y = hwy, linetype = drv), color = "blue") +
geom_smooth(mapping = aes(x = displ, y = cty, linetype = drv), color = "red")
```

# Bigger Image of the Previous Plot



# Use Lines From Points For Comparison

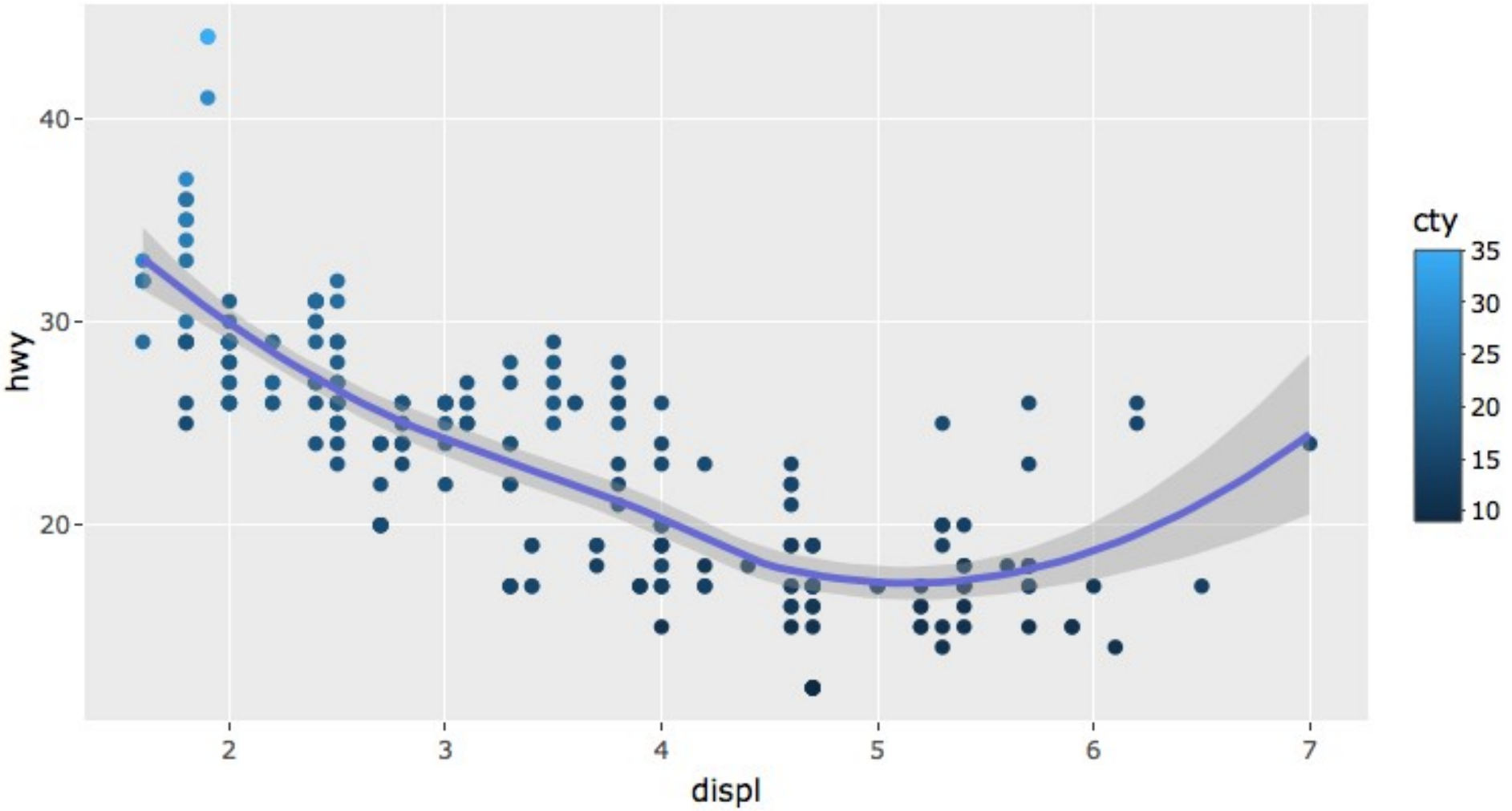



```
ggplot(mpg) +
  geom_smooth(mapping = aes(x = displ, y = hwy, linetype = drv), color = "blue") +
  geom_smooth(mapping = aes(x = displ, y = cty, linetype = drv), color = "red")
```

# Add Plotly For Interaction

```
# ref: https://plot.ly/ggplot2/stat_smooth/
#install.packages("plotly")

library(plotly)

p <- ggplot(mpg, aes(displ, hwy, color = cty))
p <- p + geom_point() + stat_smooth()

p <- ggplotly(p)

p
```

# Interact With Plots



# Add Plotly For Interaction

```
# ref: https://plot.ly/ggplot2/stat_smooth/
#install.packages("plotly")

library(plotly)

p <- ggplot(mpg, aes(displ, hwy, color = cty, size = displ))
p <- p + geom_point() + stat_smooth()

p <- ggplotly(p)

p
```

# Interact With Plots