Class 1 Notes

Combining visualization and transformation (r4ds - Section I)

For the first 2 classes, we will take the tools in r4ds section 1 - Explore, and walk through scenarios:

Load the tidyverse library, and get the mpg data *(included in the base package)*. Put that in a dataframe. Look at the structure of the data, and the first 10 rows:

```
library(tidyverse)

str(mpg) # this gives you the structure of the dataframe
```

```
Classes 'tbl_df', 'tbl' and 'data.frame':   234 obs. of  11 variables:
 $ manufacturer: chr  "audi" "audi" "audi" "audi" ...
 $ model       : chr  "a4" "a4" "a4" "a4" ...
 $ displ       : num  1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 ...
 $ year        : int  1999 1999 2008 2008 1999 1999 2008 1999 1999 2008 ...
 $ cyl         : int  4 4 4 4 6 6 6 4 4 4 ...
 $ trans       : chr  "auto(l5)" "manual(m5)" "manual(m6)" "auto(av)" ...
 $ drv         : chr  "f" "f" "f" "f" ...
 $ cty         : int  18 21 20 21 16 18 18 18 16 20 ...
 $ hwy         : int  29 29 31 30 26 26 27 26 25 28 ...
 $ fl          : chr  "p" "p" "p" "p" ...
 $ class       : chr  "compact" "compact" "compact" "compact" ...
```

```
top_n(mpg, 10)
```

```
# A tibble: 62 x 11
   manufacturer model      displ  year   cyl trans drv     cty   hwy fl    class
   <chr>        <chr>      <dbl> <int> <int> <chr> <chr> <int> <int> <chr> <chr>
 1 chevrolet    c1500 sub...   5.3  2008     8 auto... r        14    20 r     suv
 2 chevrolet    c1500 sub...   5.3  2008     8 auto... r        11    15 e     suv
 3 chevrolet    c1500 sub...   5.3  2008     8 auto... r        14    20 r     suv
 4 chevrolet    c1500 sub...   5.7  1999     8 auto... r        13    17 r     suv
 5 chevrolet    c1500 sub...   6    2008     8 auto... r        12    17 r     suv
 6 chevrolet    k1500 tah...   5.3  2008     8 auto... 4        14    19 r     suv
 7 chevrolet    k1500 tah...   5.3  2008     8 auto... 4        11    14 e     suv
 8 chevrolet    k1500 tah...   5.7  1999     8 auto... 4        11    15 r     suv
 9 chevrolet    k1500 tah...   6.5  1999     8 auto... 4        14    17 d     suv
10 dodge        durango 4...   3.9  1999     6 auto... 4        13    17 r     suv
# ... with 52 more rows
```

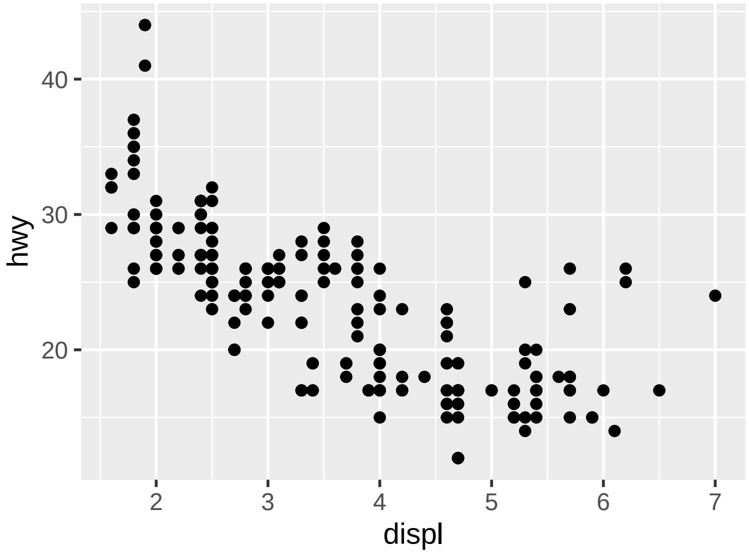
What do the datatypes in str tell you. What's an int? a chr? a num?

Now that we have some data, let's work with visualization in ggplot:

(Note that I use a different syntax from Hadley - I store the plots in "objects" - this time I named it "p", but you can name it anything. Then I can add whatever I like later without having to recreate the whole plot). Also, I use a less verbose syntax. Either way is fine - up to you.

Note: with the asesthtics *(aes)*, x comes first.

```
p = ggplot(mpg, aes(displ,hwy))
p = p + geom_point()
p
```

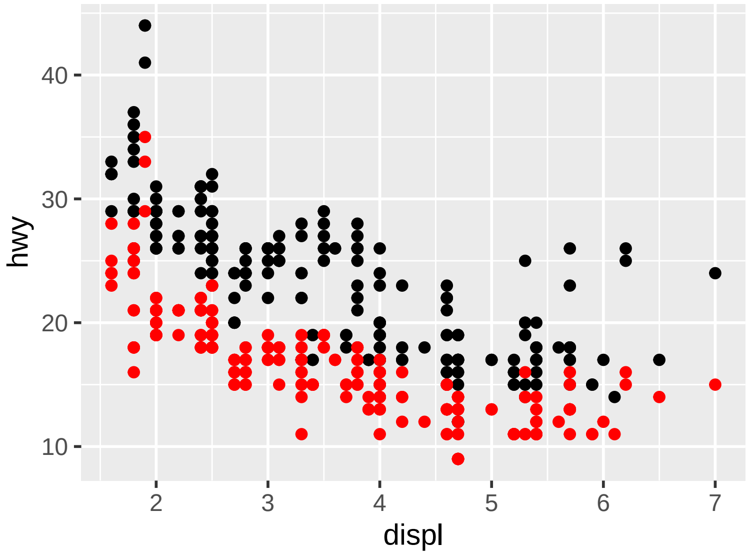


So, what does this tell us? Does it make sense? If it doesn't, go back and figure out. It's not about drawing pictures, it's about understanding the data. *(Bots can draw pictures - do you want to compete with bots?)*

Now that we have highway mileage, let's add city and compare *(notice how we are just adding a **layer** - one of the beauties of ggplot - it really is well designed)*.

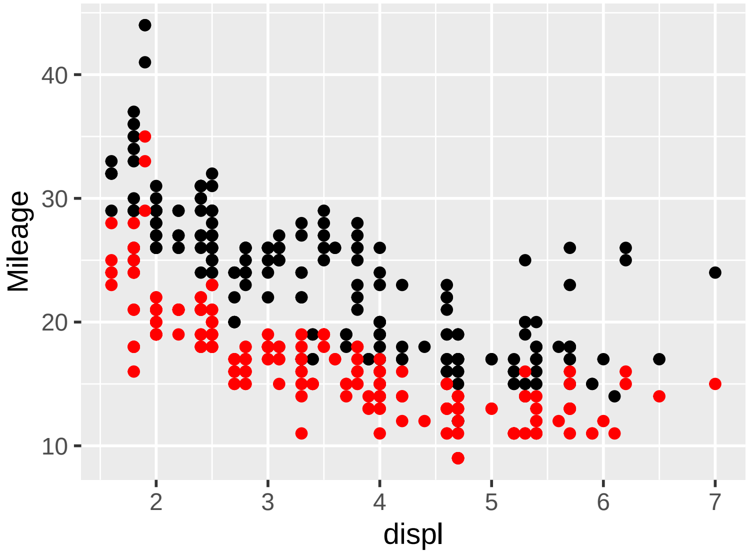
Does this all make sense? Always ask! So, we'll add a new layer and color it red:

```
p = p + geom_point(aes(displ, cty), color = "red")
p
```



There's something wrong with this plot, though - can you see it? it's not hwy mileage anymore, let's correct it:

```
p <- p + ylab("Mileage")
p
```

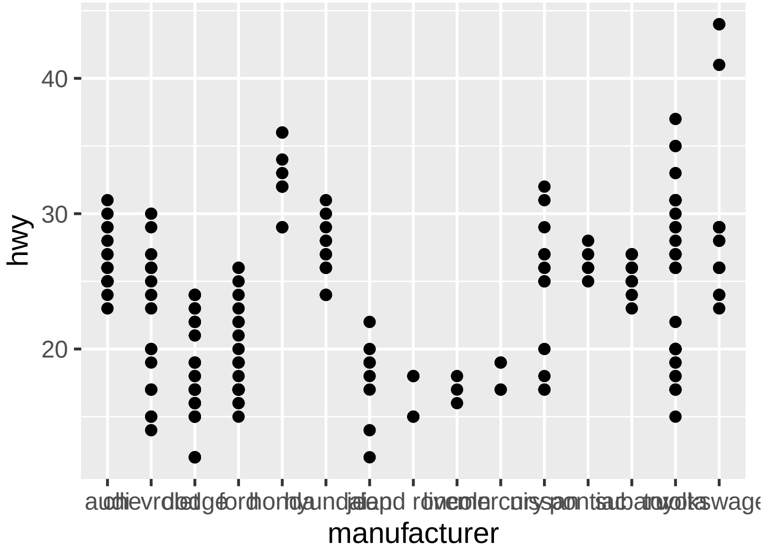


Notice again how we didn't have to recreate the whole plot - we just added a layer.

What if we wanted to see milege by manufacturer? How would we do it? **Think**

Let's put manufacturer on the X axis instead of displ:

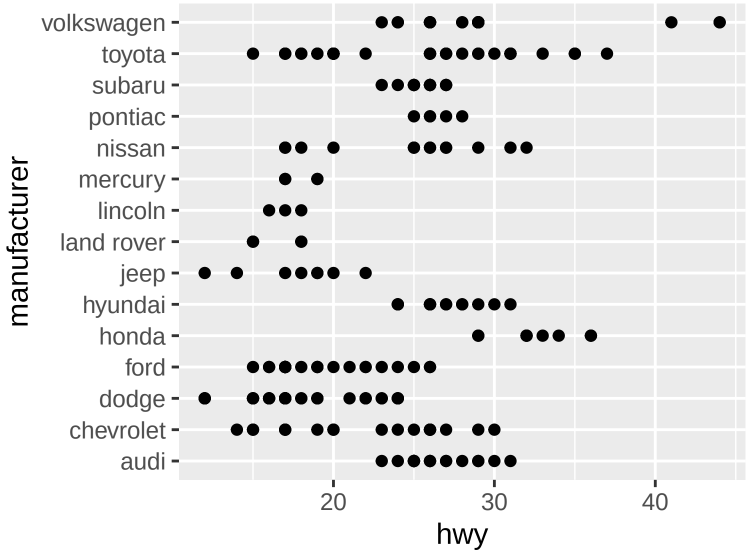
```
p = ggplot(mpg, aes(manufacturer,hwy))
p = p + geom_point()
p
```



What does each of the points represent? You should know. Also, notice that we DID need to create a new plot here - we're no longer building.

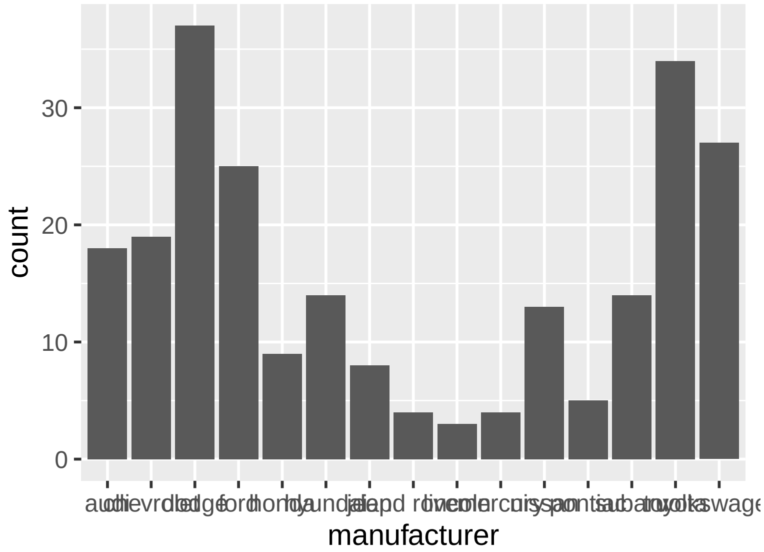
This is hard to read. We can fix that by flipping the axis *(add a layer)*:

```
p = p + coord_flip()
p
```



This might look better as a bar *(we'll create a new plot)*

```
p = ggplot(mpg, aes(manufacturer)) + geom_bar()
p
```



Wait! what is this? Does this make sense? **THINK**. geom_bar defaults to count. Is that what we want?

We want to know hwy mileage by maufacturer, but the sample had a different number of cars by manufacturer *(look at it - know your data)* We can verify that with the function count.

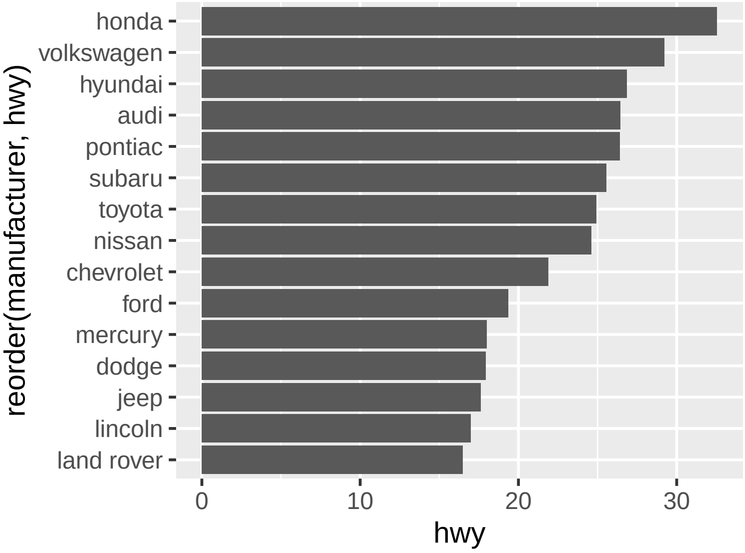
```
count(mpg, manufacturer)
```

```
# A tibble: 15 x 2
   manufacturer     n
   <chr>        <int>
```

```
 1 audi            18
 2 chevrolet       19
 3 dodge           37
 4 ford            25
 5 honda            9
 6 hyundai         14
 7 jeep             8
 8 land rover       4
 9 lincoln          3
10 mercury          4
11 nissan          13
12 pontiac          5
13 subaru          14
14 toyota          34
15 volkswagen      27
```

So, is that what we want to know? **NO!** We want to see mileage, right? Here's one way:

```
p = ggplot(mpg) +
  geom_bar(aes(reorder(manufacturer, hwy), hwy), stat = "summary", fun.y = "mean") +
  coord_flip()
p
```



A little explaination:

reorder(manufacturer, hwy) reorders the x axis by hwy *(that way, we can see the ranking)*

stat = "summary", fun.y = "mean" tells ggplot that we want a summary function and that function is the mean. *(geom_bar and geom_histogram will normally default to count. This command will override that - but you can also do your own math here)*

That said, This is not usually the way you would build this in practice. That's because you are usually working with a particular stat, and you're going to continue your analysis beyond the exploratory visualization. So, normally, you would create the stat, and **store** it, so you can use it later - something like:

```
mpg %>% group_by(manufacturer) %>% summarise(AvgHwy = mean(hwy))
```

```
# A tibble: 15 x 2
   manufacturer AvgHwy
   <chr>          <dbl>
 1 audi            26.4
 2 chevrolet       21.9
 3 dodge           17.9
 4 ford            19.4
 5 honda           32.6
 6 hyundai         26.9
 7 jeep            17.6
 8 land rover      16.5
 9 lincoln         17
10 mercury         18
11 nissan          24.6
12 pontiac         26.4
13 subaru          25.6
14 toyota          24.9
15 volkswagen      29.2
```

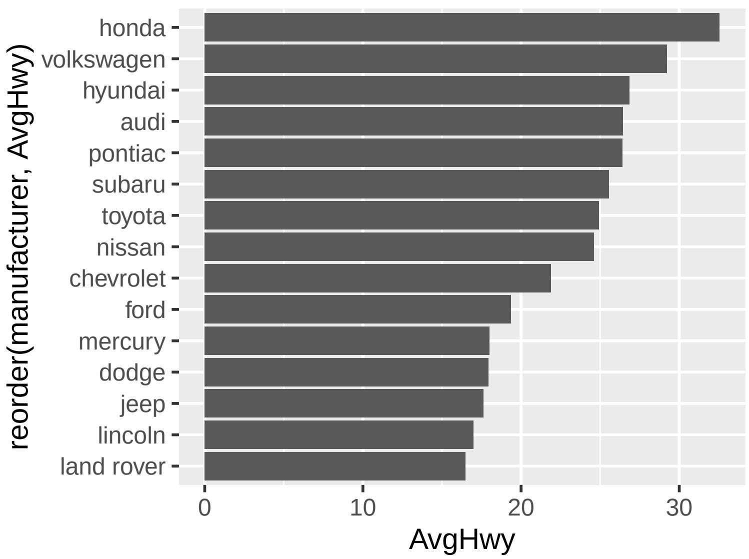
And then we can plot that. BUT, you have to remember to save *(persist)* the analysis to an object *(it hasn't been saved - you just asked R to do a group_by AND THEN do a summarise, but you did not save it)*. To save it *(and use it later)*, you can:

```
dfAvgHwy = mpg %>% group_by(manufacturer) %>% summarise(AvgHwy = mean(hwy))
head(dfAvgHwy)
```

```
# A tibble: 6 x 2
  manufacturer AvgHwy
  <chr>         <dbl>
1 audi           26.4
2 chevrolet      21.9
3 dodge          17.9
4 ford           19.4
5 honda          32.6
6 hyundai        26.9
```

And then plot it:

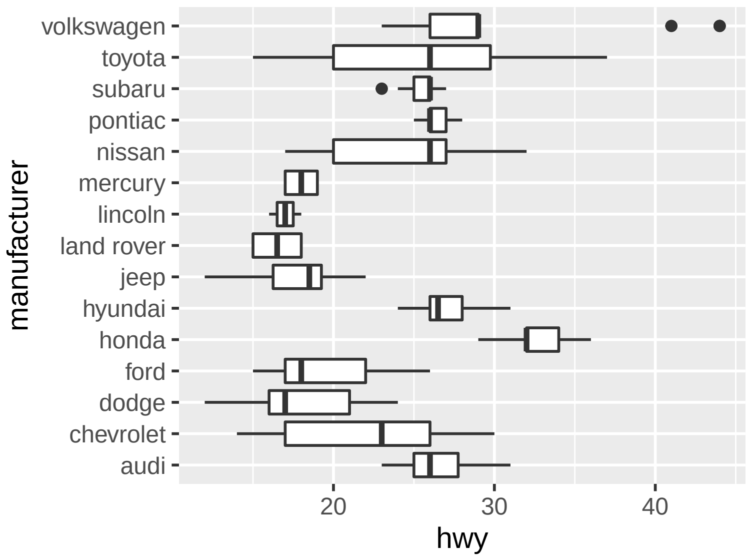
```
p = ggplot(dfAvgHwy, aes(reorder(manufacturer, AvgHwy), AvgHwy)) +
  geom_bar(stat = "identity") +
  coord_flip()
p
```



Notice how we have changed datasource and y axis *(which becomes the x axis after we do a coordinate flip)*. Please understand this process. This is a very basic and expected skill.

There are better ways to look at all this. Boxplots are great:

```
p = ggplot(data = mpg, aes(x = manufacturer, y = hwy)) +
  geom_boxplot() +
  coord_flip()
p
```



The boxplot gives us a visual of the distribution of data by manufacturer. It also shows the mean, AND the first and third quartiles.

Look at volkswagen. That should get your attention - you should notice that the mean and third quartile are really close, with 2 "outliners" beyond 40 hwy. Let's take a summary look at the data:

```
summary(filter(mpg, manufacturer == "volkswagen")$hwy)
```

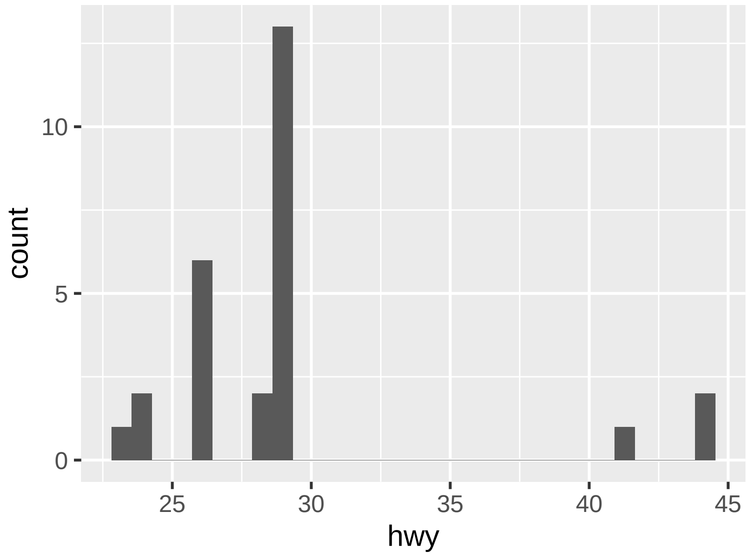
```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
```

```
  23.00   26.00   29.00   29.22   29.00   44.00
```

(summary can be applied to a range of objects - we'll use it extensively)

So far, its hard to tell why the mean is right next to the 3rd quartile. Let's look at a histogram:

```
p = ggplot(filter(mpg, manufacturer == "volkswagen"), aes(hwy)) + geom_histogram()
p
```



What does a quantile, or quartile mean? It's how the data are ordered. What is a mean? Could it be that there's a group of observations that are "pulling" the mean up - skewing it? Let's take a closer look:

```
filter(mpg, manufacturer == "volkswagen") %>% arrange(desc(hwy))
```

```
# A tibble: 27 x 11
   manufacturer model   displ  year   cyl trans   drv     cty   hwy fl    class
   <chr>        <chr>   <dbl> <int> <int> <chr>   <chr> <int> <int> <chr> <chr>
 1 volkswagen   jetta     1.9  1999     4 manual... f        33    44 d     compa...
 2 volkswagen   new be...   1.9  1999     4 manual... f        35    44 d     subco...
 3 volkswagen   new be...   1.9  1999     4 auto(l... f        29    41 d     subco...
 4 volkswagen   gti        2    1999     4 manual... f        21    29 r     compa...
 5 volkswagen   gti        2    2008     4 manual... f        21    29 p     compa...
 6 volkswagen   gti        2    2008     4 auto(s... f        22    29 p     compa...
 7 volkswagen   jetta      2    1999     4 manual... f        21    29 r     compa...
 8 volkswagen   jetta      2    2008     4 auto(s... f        22    29 p     compa...
 9 volkswagen   jetta      2    2008     4 manual... f        21    29 p     compa...
10 volkswagen   jetta      2.5  2008     5 auto(s... f        21    29 r     compa...
# ... with 17 more rows
```

Comparing jettas and new beetles

```
filter(mpg, manufacturer == "volkswagen", model %in% c("jetta", "new beetle"))
```

```
# A tibble: 15 x 11
   manufacturer model   displ  year   cyl trans   drv     cty   hwy fl    class
   <chr>        <chr>   <dbl> <int> <int> <chr>   <chr> <int> <int> <chr> <chr>
 1 volkswagen   jetta     1.9  1999     4 manual... f        33    44 d     compa...
 2 volkswagen   jetta      2    1999     4 manual... f        21    29 r     compa...
 3 volkswagen   jetta      2    1999     4 auto(l... f        19    26 r     compa...
 4 volkswagen   jetta     2    2008     4 auto(s... f        22    29 p     compa...
 5 volkswagen   jetta     2    2008     4 manual... f        21    29 p     compa...
 6 volkswagen   jetta     2.5  2008     5 auto(s... f        21    29 r     compa...
 7 volkswagen   jetta     2.5  2008     5 manual... f        21    29 r     compa...
 8 volkswagen   jetta     2.8  1999     6 auto(l... f        16    23 r     compa...
 9 volkswagen   jetta     2.8  1999     6 manual... f        17    24 r     compa...
10 volkswagen   new be...   1.9  1999     4 manual... f        35    44 d     subco...
11 volkswagen   new be...   1.9  1999     4 auto(l... f        29    41 d     subco...
12 volkswagen   new be...   2    1999     4 manual... f        21    29 r     subco...
13 volkswagen   new be...   2    1999     4 auto(l... f        19    26 r     subco...
14 volkswagen   new be...   2.5  2008     5 manual... f        20    28 r     subco...
15 volkswagen   new be...   2.5  2008     5 auto(s... f        20    29 r     subco...
```

So, These “outliers” are not really outliers. They’re diesels. And that would explain the wide gap.

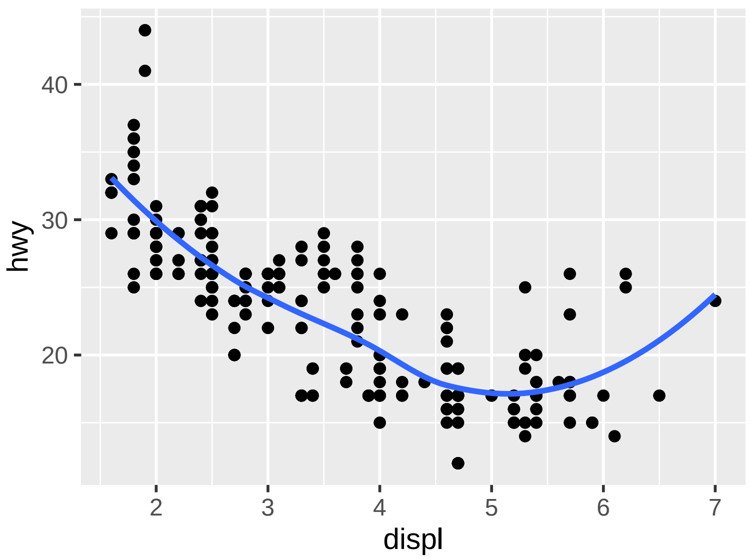
```r
filter(mpg, manufacturer == "volkswagen", model %in% c("jetta", "new beetle"))
```

```
# A tibble: 15 x 11
   manufacturer model   displ  year   cyl trans   drv     cty   hwy fl    class
   <chr>        <chr>   <dbl> <int> <int> <chr>   <chr> <int> <int> <chr> <chr>
 1 volkswagen   jetta     1.9  1999     4 manual... f        33    44 d     compa...
 2 volkswagen   jetta     2    1999     4 manual... f        21    29 r     compa...
 3 volkswagen   jetta     2    1999     4 auto(l... f        19    26 r     compa...
 4 volkswagen   jetta     2    2008     4 auto(s... f        22    29 p     compa...
 5 volkswagen   jetta     2    2008     4 manual... f        21    29 p     compa...
 6 volkswagen   jetta     2.5  2008     5 auto(s... f        21    29 r     compa...
 7 volkswagen   jetta     2.5  2008     5 manual... f        21    29 r     compa...
 8 volkswagen   jetta     2.8  1999     6 auto(l... f        16    23 r     compa...
 9 volkswagen   jetta     2.8  1999     6 manual... f        17    24 r     compa...
10 volkswagen   new be...   1.9  1999     4 manual... f        35    44 d     subco...
11 volkswagen   new be...   1.9  1999     4 auto(l... f        29    41 d     subco...
12 volkswagen   new be...   2    1999     4 manual... f        21    29 r     subco...
13 volkswagen   new be...   2    1999     4 auto(l... f        19    26 r     subco...
14 volkswagen   new be...   2.5  2008     5 manual... f        20    28 r     subco...
15 volkswagen   new be...   2.5  2008     5 auto(s... f        20    29 r     subco...
```

We might consider segregating the fuel. Worry about that later.

One last thing, let’s look at a regression line and see if there’s a trend:

```r
p = ggplot(mpg, aes(displ,hwy)) +
  geom_point() +
  geom_smooth(se = F)
p
```



We'll get into regression modeling this semester, but for now, you can see there's a trend where hwy mileage decreases with the size of the engine *(displ)*. Duh.