Stat_437 Lecture 2a Notes

John Salmon (011745357)

#Lecture 2a link to slides

Topics: - faceting - annotating a plot - manually setting some scales - adjusting some guides - mathematical expressions in plots

Faceting

- a dataset can be split into subsets based on some criteria. The mpg dataset can be split into 7 subsets according to the levels of class or 3 based on the level of drv.

-faceting takes an alternative approach to aesthetic by creating the same graph for each subset or subgroup instead of changing the color

Two ways to facet 1. facet_grid: 2d grid of panels defined by variables which form the rows and columns 2. facet_wrap: ribbon of 1d panels wrapped into 2d

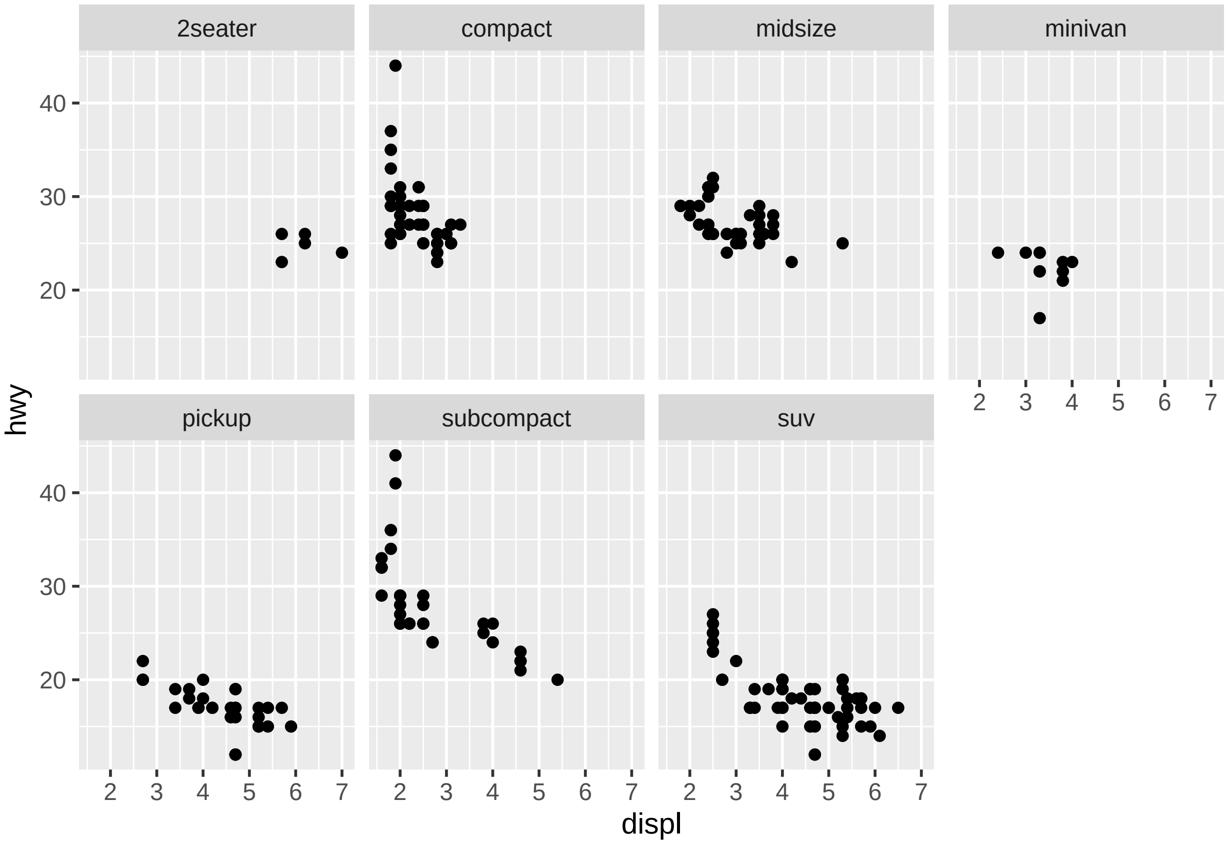
basic syntax facet_wrap(facets, nrow = NULL, ncol = NULL, scales = "fixed", labeller = "label_value")

- facets can be specified by ~variable or vars(variable)
- nrow (or ncol) sets the number of rows or columns in which graphs are displayed
- scales and labeller later
- use ?facet_wrap to get more information

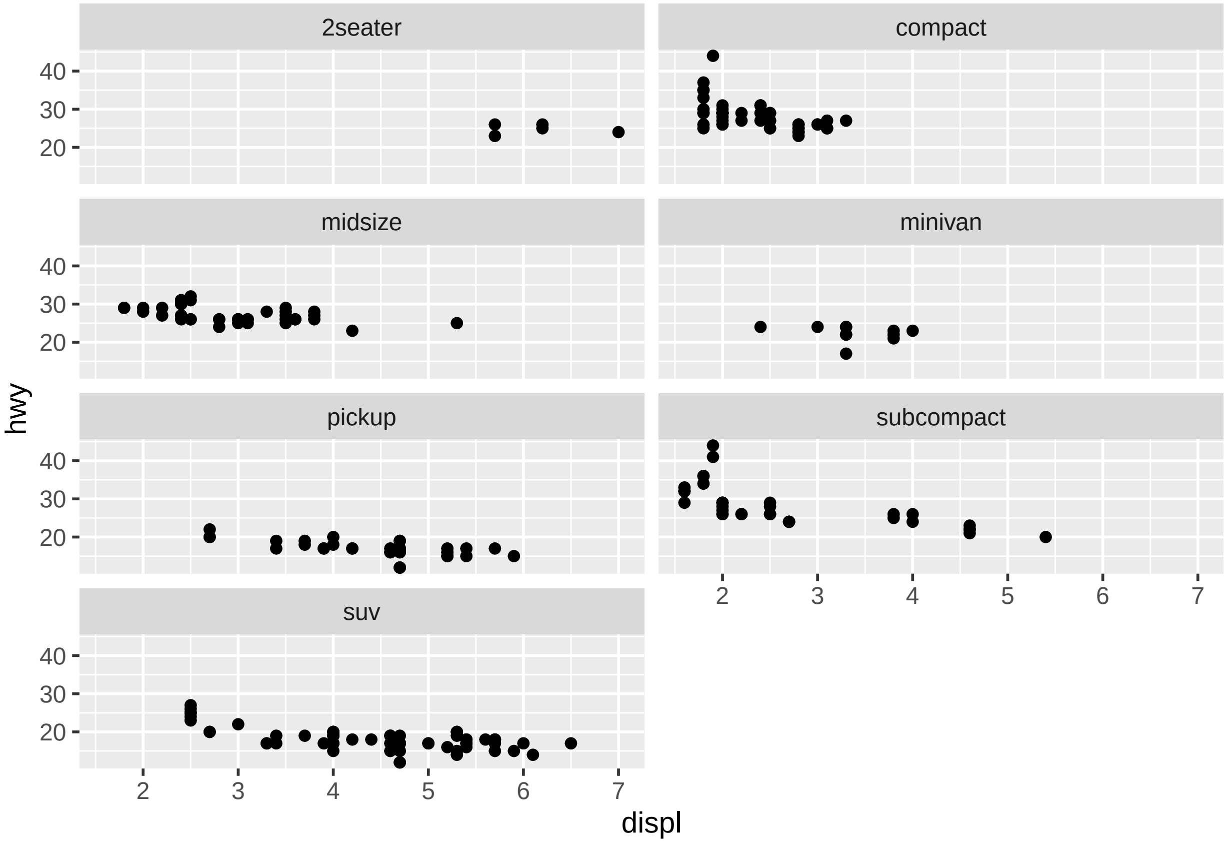
```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.2
```

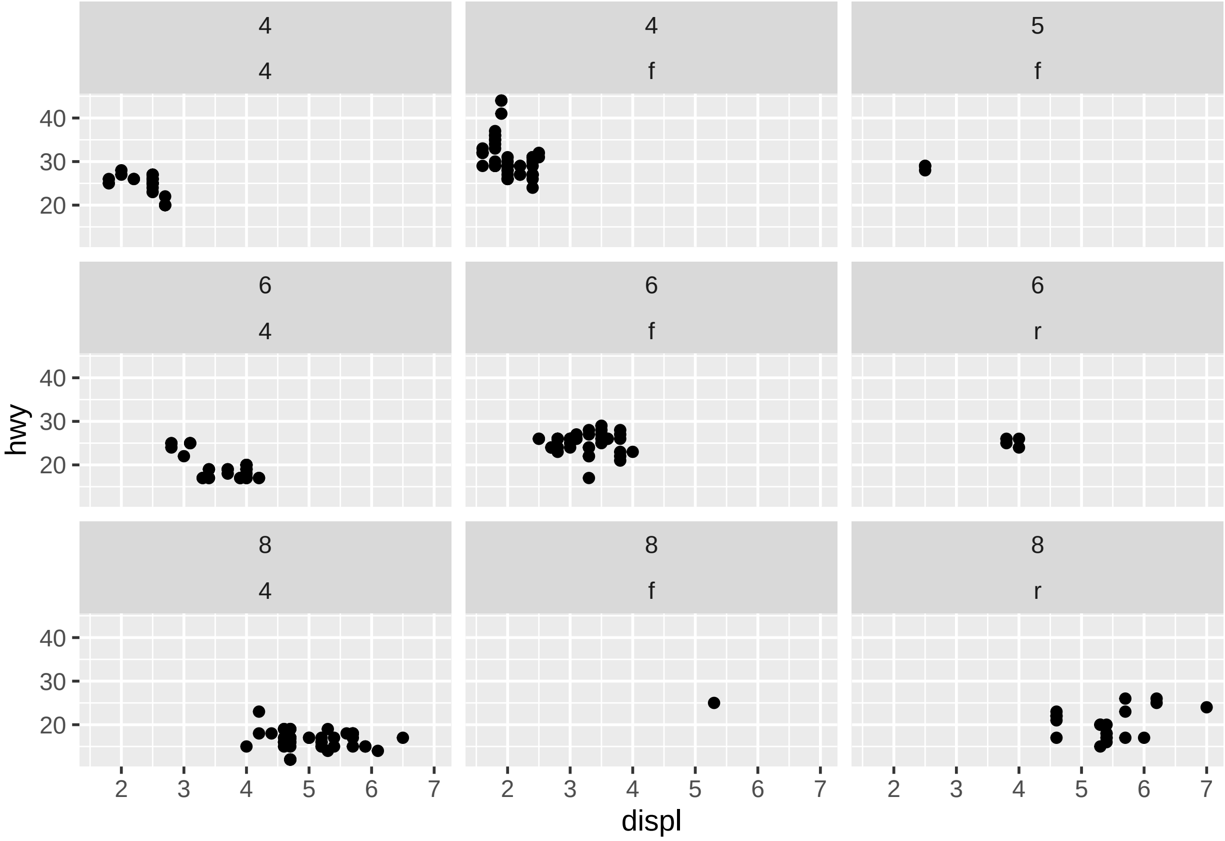
```
# build a base layer
p1= ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy))
# add faceting via `class` to p1
p2 = p1 + facet_wrap(~class, nrow = 2)
p2
```



```
#p1 + facet_wrap(vars(class), nrow = 2)

p2a = p1 + facet_wrap(~class, ncol = 2) #by column instead of row
p2a
```



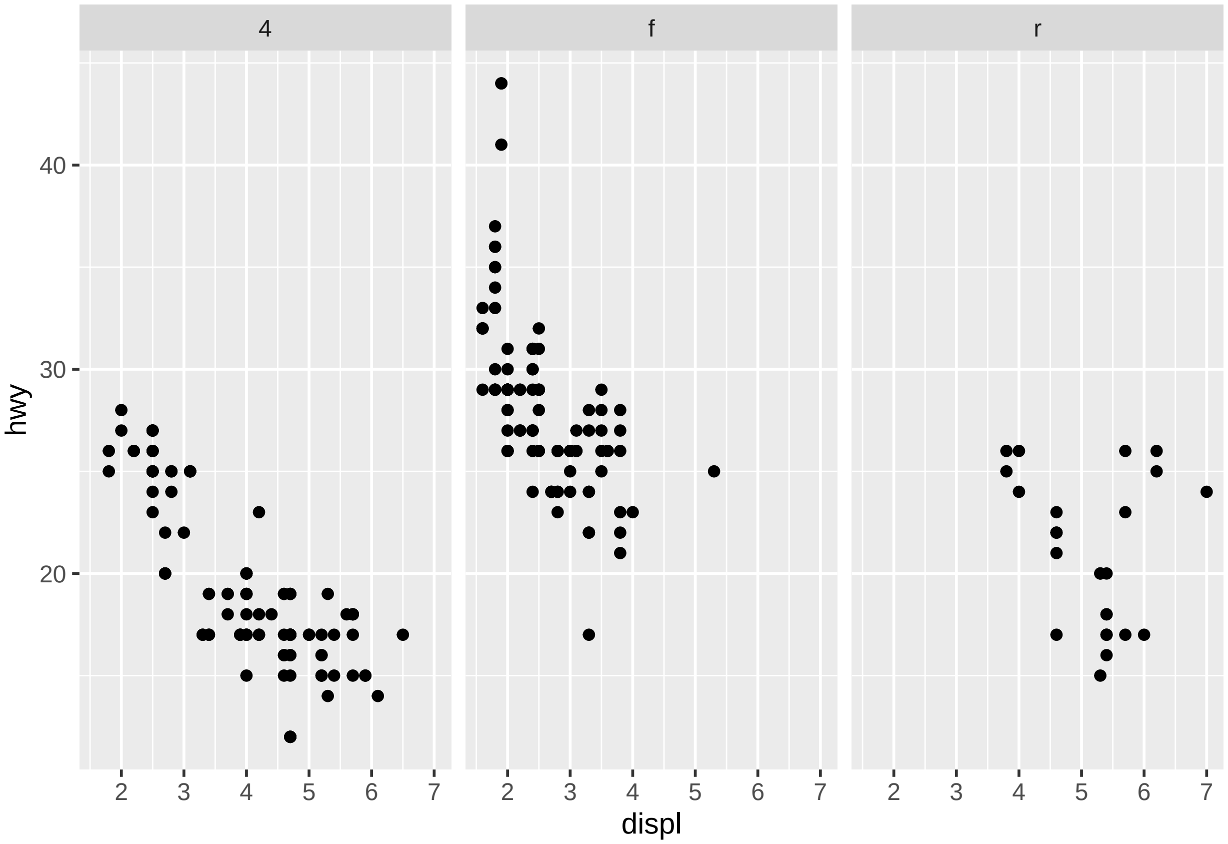
```
p3 = ggplot(mpg, aes(x=displ, y =hwy)) + geom_point() + facet_wrap(cyl~drv)
p3
```



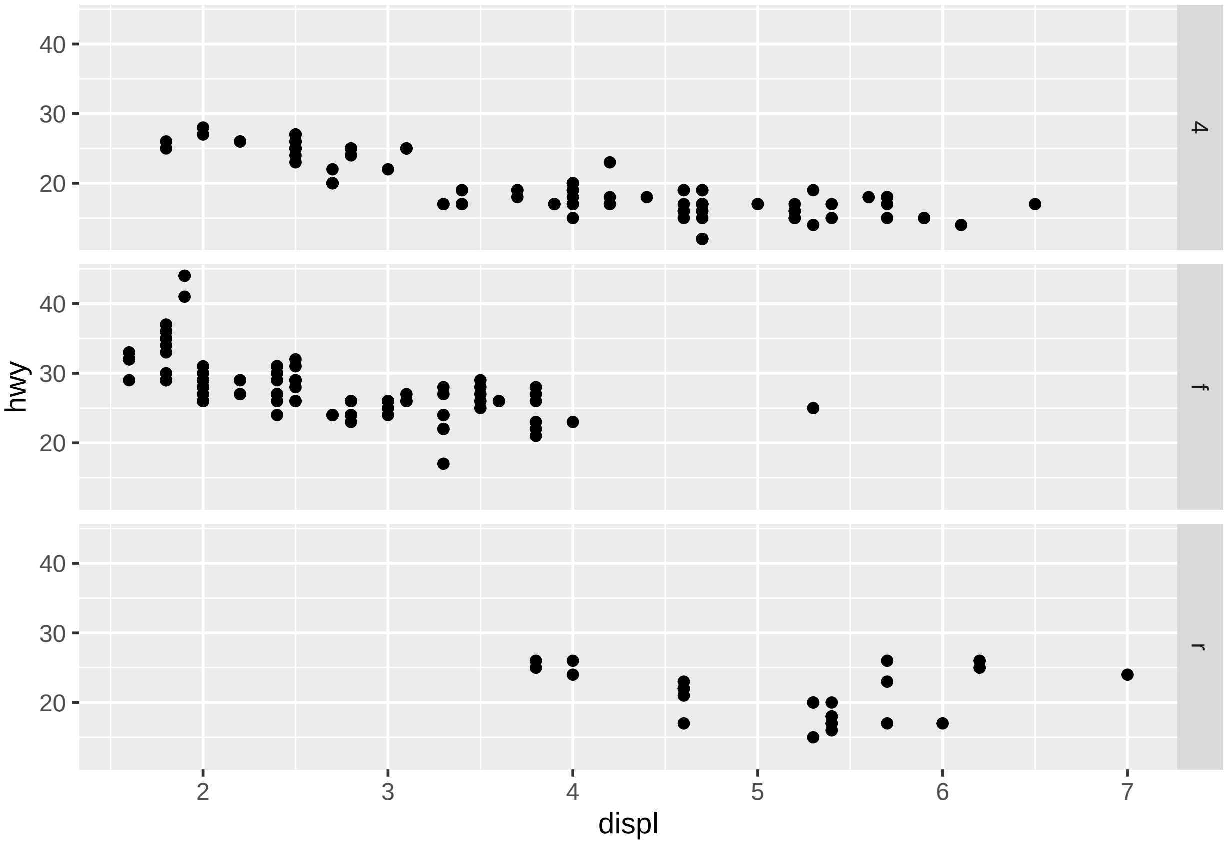
p3 is pretty ugly because the titles end up taking space but it is still useful to know how to do for visualizing multiple levels.

Facet Grid Grid forms a matrix of panels defiined by row and column faceting variables - use ?facet_grid to obtain details -rows = NULL, cols = NULL takes the form of variable1 ~ variable2 .~ variable 2 or variable1 ~. - rows (or cols) variables that define faceting groups on the row(or column) division.

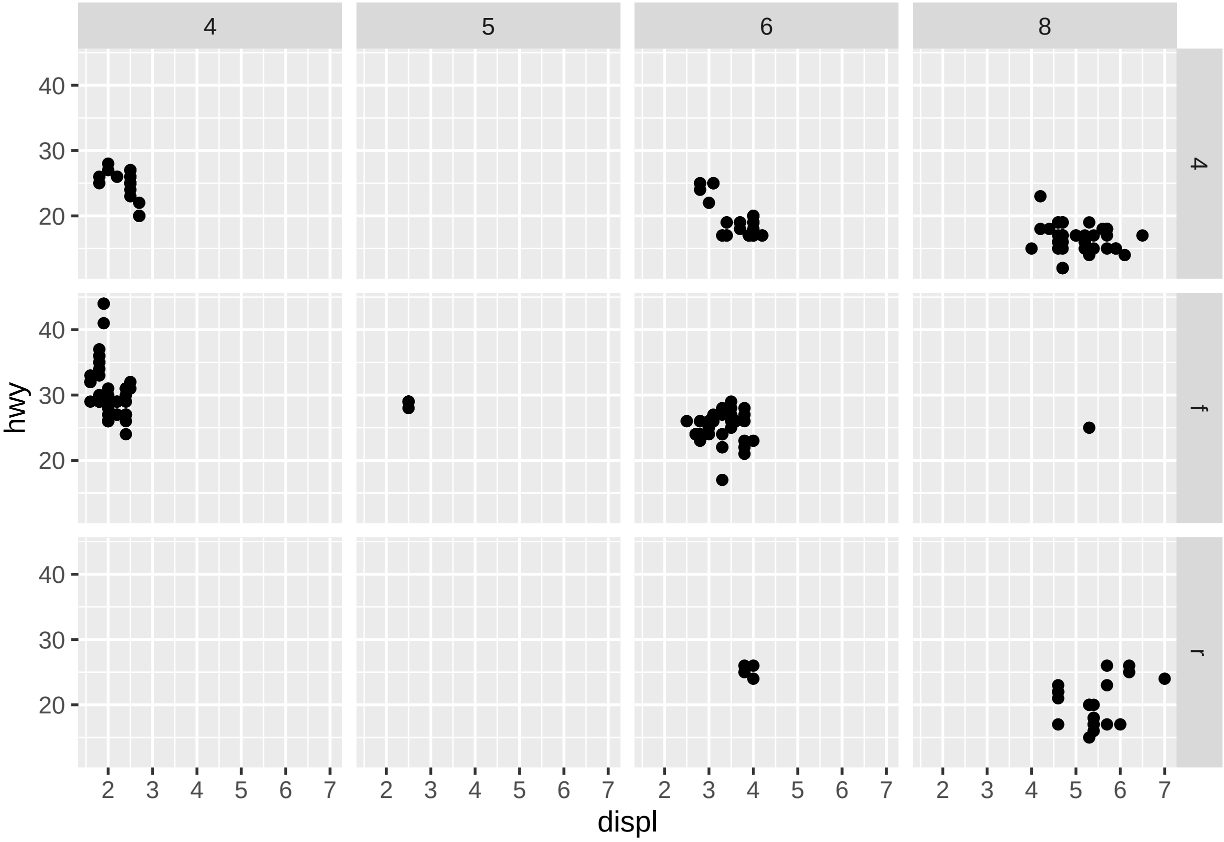
```
p1 + facet_grid(.~ drv) #row wise
```



```
p1 + facet_grid(drv ~ .)#column wise
```



```
#two variable faceting
p1 + facet_grid(drv ~ cyl)
```



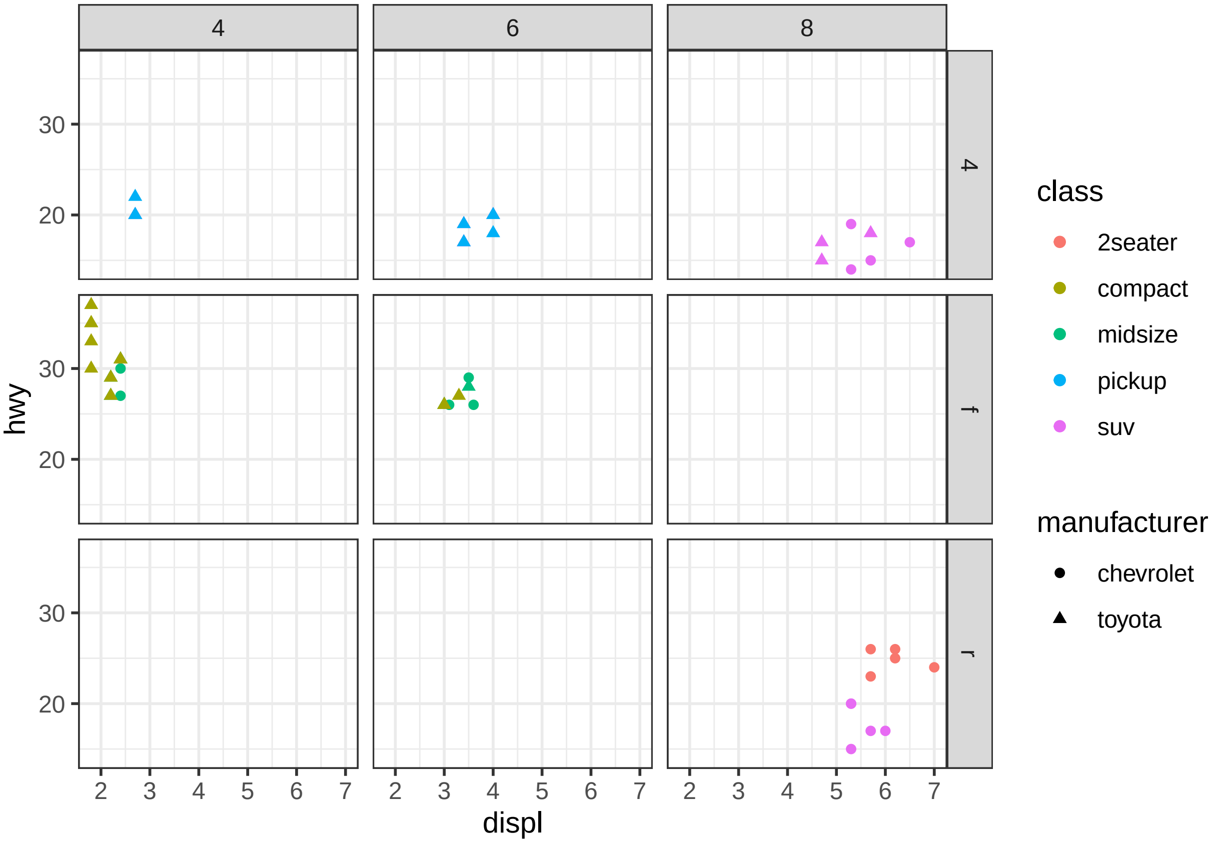
Notice that there are some blank grids where there are not any points, this is fine to leave in for the homework.

```
library(dplyr)
```

Visualization with >=3 factors

```
##
## 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
```

```
mpg1 = mpg %>% filter(manufacturer %in% c("chevrolet", "toyota"))
p1c = ggplot(mpg1, aes(x=displ, y = hwy)) + theme_bw() + geom_point(aes(color=class, shape = manufactur
p1c
```



6-Dimensional Plot! displ x hwy x cyl x drv x class x manufacturer we can see different subcategories and groupings/sub-groupings even better now

real world application of plotting another example(professor wrote it)

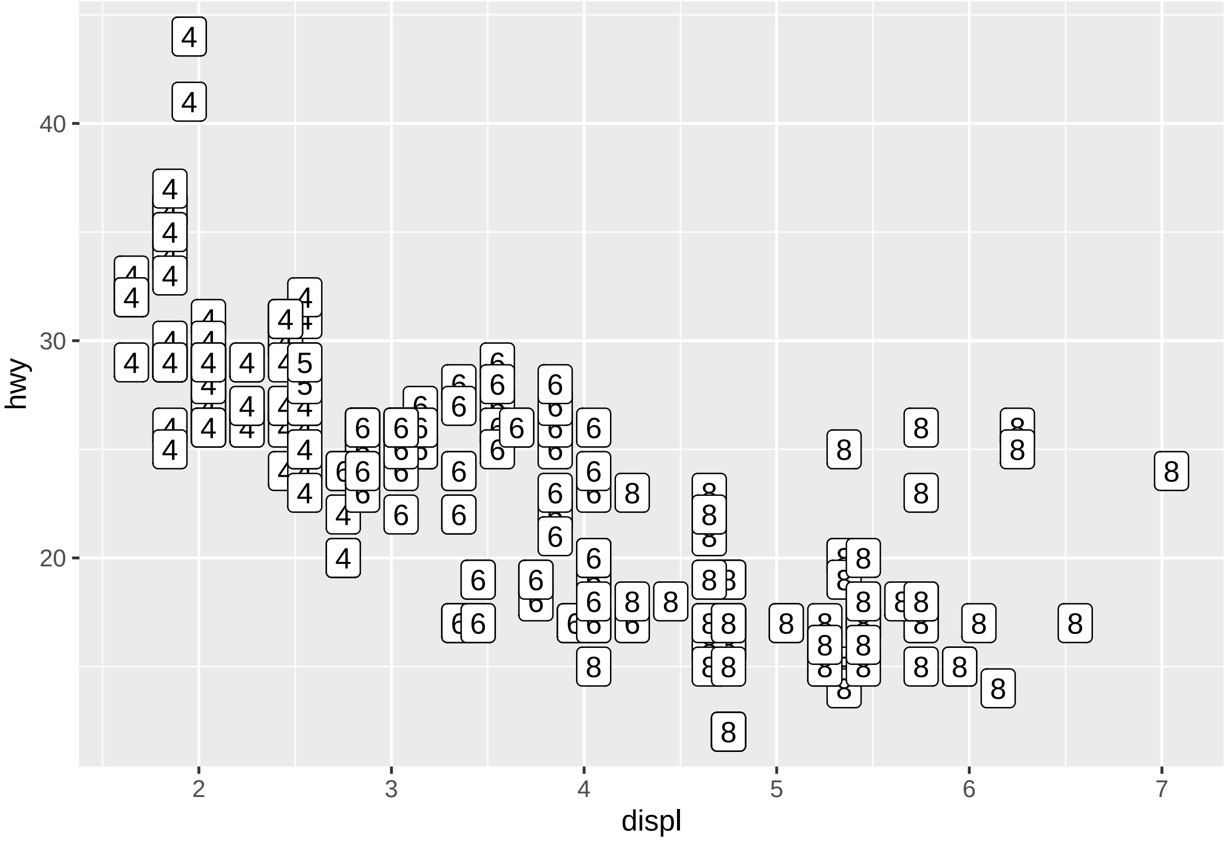
Annotating a plot two common commands for annotation: - geom_text() - geom_label()

the syntax for both are very similar, however there is one line to pay attention to. Mapping: A set of aesthetic mappings created by aes(); if speciified and inherit.aes = TRUE(default), mapping is combined with the default mapping at the top level of the plot. -mapping must be supplied if there is no plot mapping - data: data too be displayed in this layer; if NULL, the default, data are inherited from the plot data as specified in the call to ggplot() - parse: if TRUE labels will (often) be parsed into (math) expressions; FALSE by default

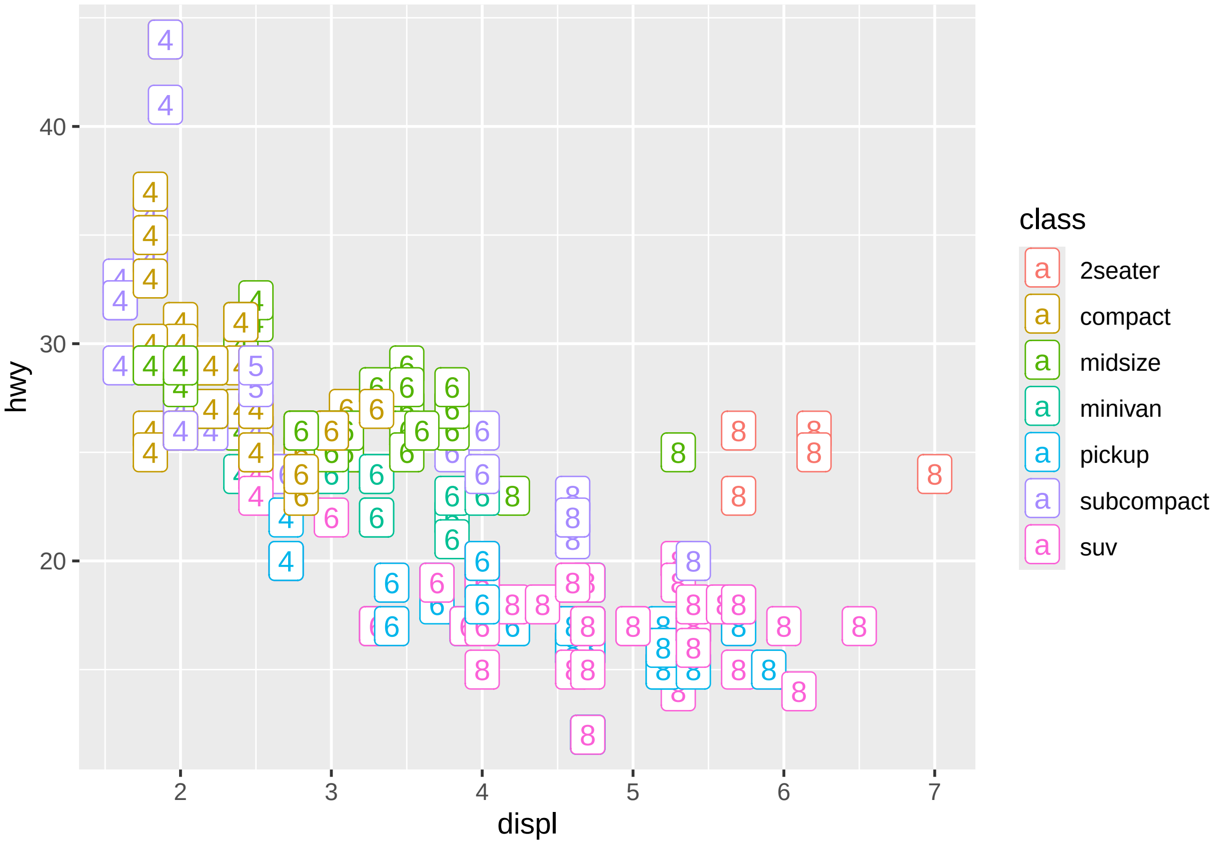
use ?geom_text and ?geom_label for more information

3 hour geom_text command

```
p = ggplot(mpg, aes(displ,hwy))
#add label via cyl; label is an aesthetic
p1 = p + geom_label(aes(label=cyl), nudge_x = 0.05) #nudge moves the label in the x direction
p1
```



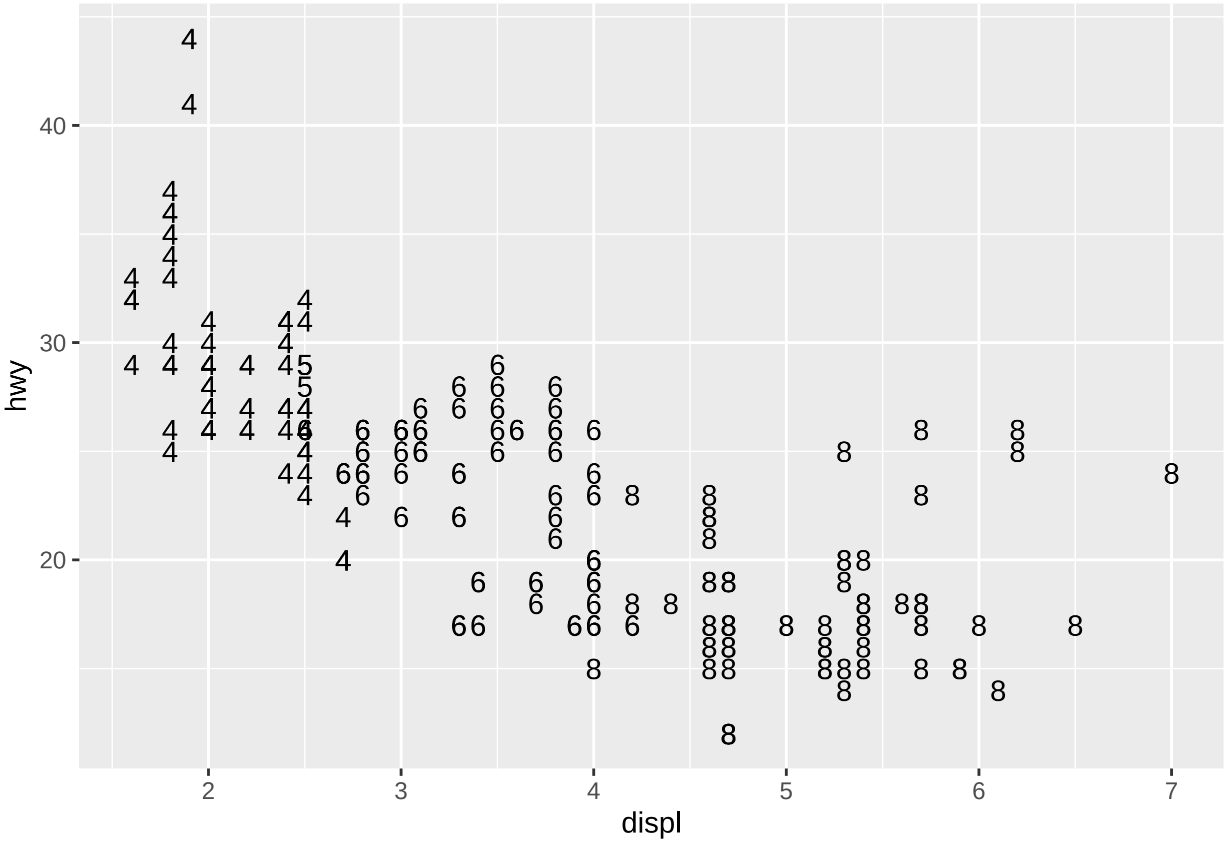
```
p+geom_label(aes(label=cyl, color=class)) #add class
```



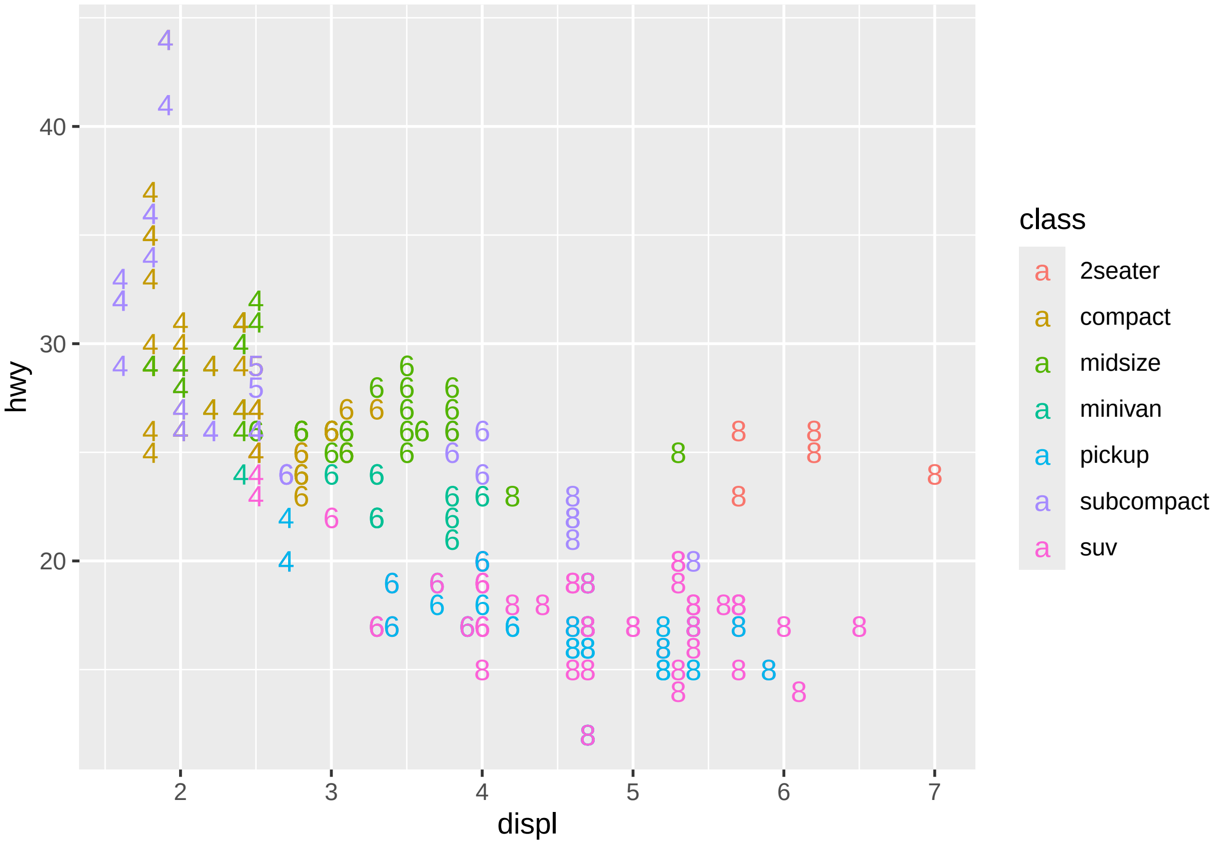
Each observation is a label which corresponds to the number of cylinders Color for class was added on the second plot

geom text will add characters directly to the plot and you have to specify where you want them to be.

```
q = ggplot(mpg,aes(displ,hwy)) + geom_text(aes(label=cyl))
q
```



```
#can also add color
qa = ggplot(mpg,aes(displ,hwy)) + geom_text(aes(label=cyl, color = class))
qa
```



similar to label but just as raw text the second plot also has color added

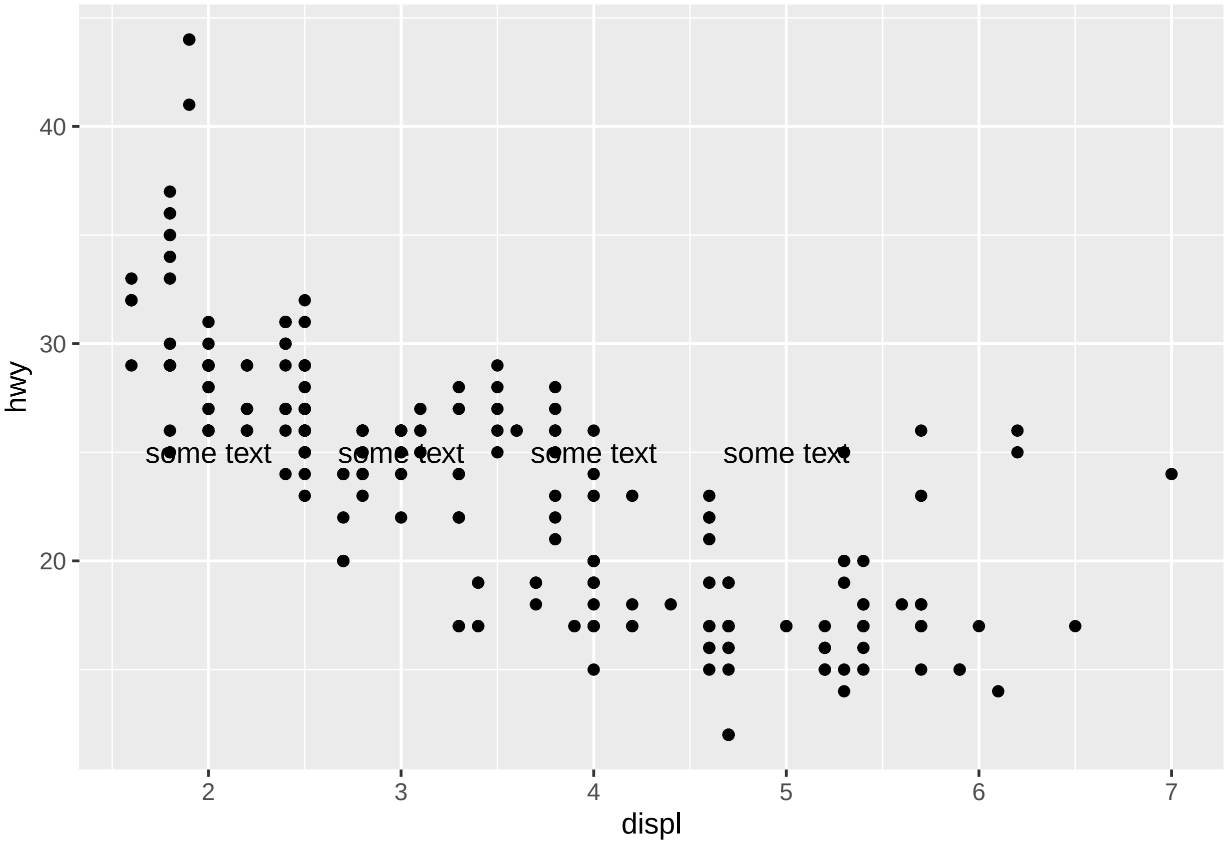
Annotate but harder(math)

Annotate adds geometric objects to a plot. But unlike a typical geom functiion, properties of the geometric objects are not mapped from variables of a datafream, instead they are vectors

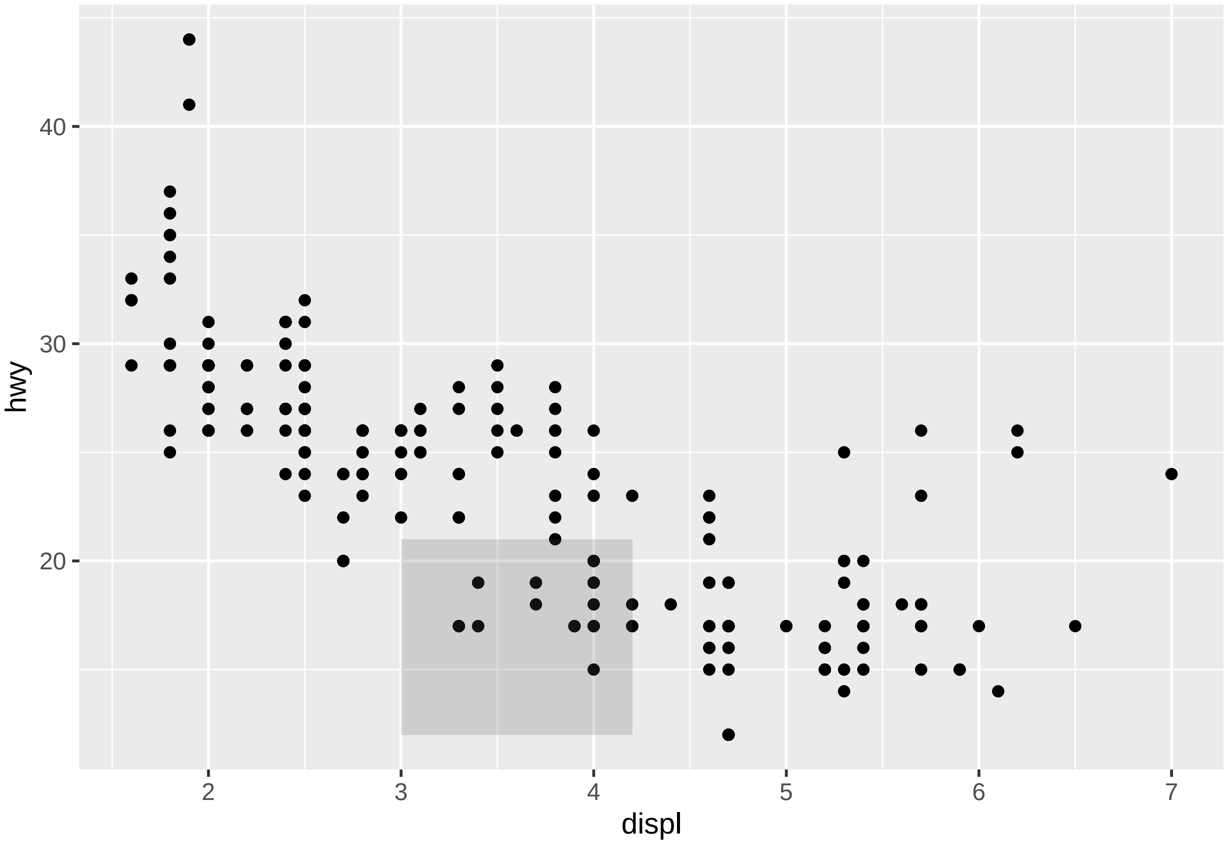
The Annotate Command has a geom argument that states what object is to be used for annotating.

Just look at the slides for the annotating notes, I dont really want to write all of that

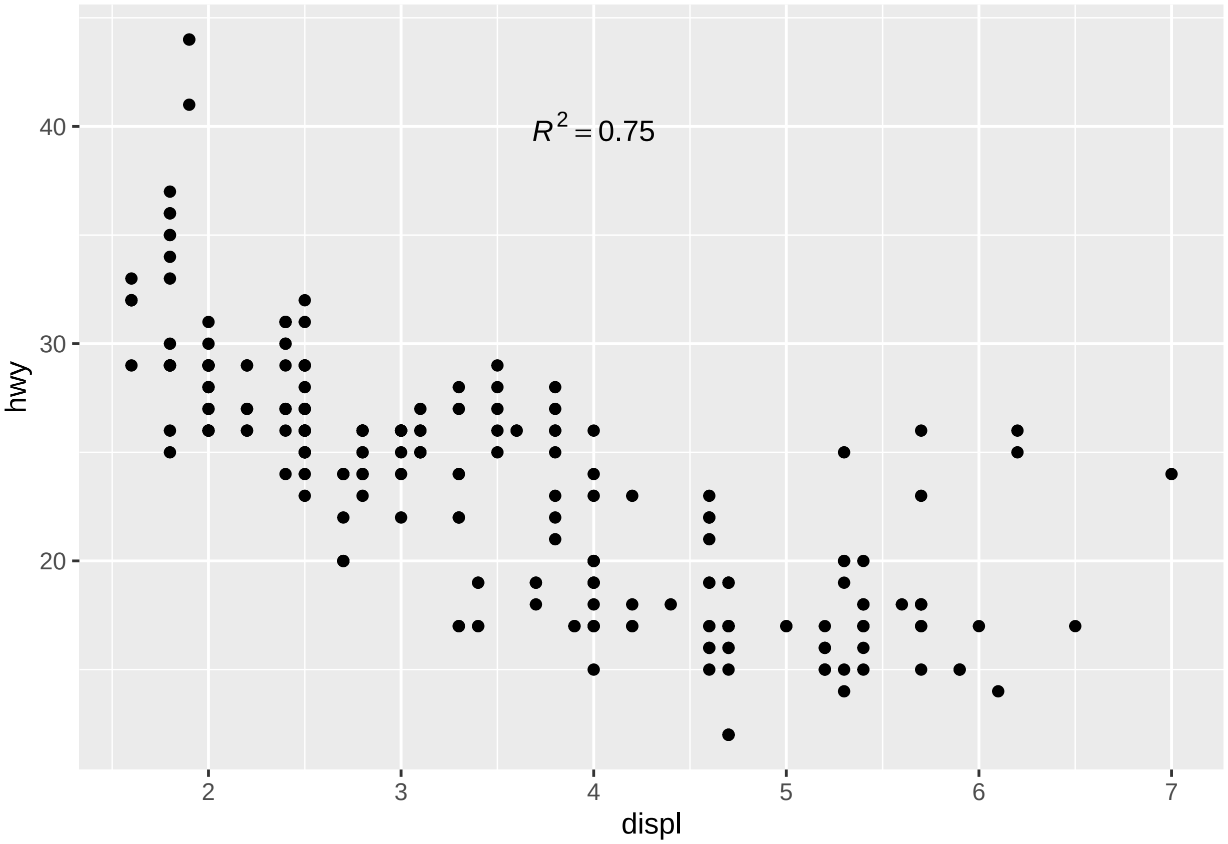
```r
p = ggplot(mpg, aes(x=displ, y = hwy)) + geom_point()

p1 = p + annotate("text", x = 2:5, y = 25, label = "some text")
p1
```



```
p2 = p + annotate("rect", xmin = 3, xmax = 4.2, ymin = 12, ymax = 21, alpha = .2)
p2
```



```
p2 = p + annotate("text", x = 4, y = 40, label = "italic(R) ^ 2 == 0.75", parse = TRUE) #convert to mat
p2
```


$R^2 = 0.75$
hwy
40
30
20
displ
2
3
4
5
6
7