STATS 2107
Statistical Modelling and Inference II
Practical 3: Graphics

Sharon Lee, Matt Ryan

Semester 2 2022

In this practical, we start to look at how to use ggplot2 to visualise relationships between variables.

The mpg dataset is a fuel economy dataset from 1999 and 2008 for 38 popular models of cars. This dataset is included in the ggplot2 package.

Variable	Description	Details
manufacturer	car manufacturer	15 manufacturers
model	mode; name	38 models
displ	engine displacement in litres	
year	year of manufacturing	
cyl	number of cylinders	4,5,6,8
trans	type of transmission	automatic, manual (many sub types)
drv	drive type	f=front wheel, r=rear wheel, 4=4 wheel
cty	city mileage	miles per gallon
hwy	highway mileage	miles per gallon
fl	fuel type	5 fuel types (diesel, petrol, electric, etc.)
class	vehicle class	7 types (compact, SUV, minivan etc.)

My first canvas

Load the `MPG` dataset.

```
library(tidyverse)
data(mpg)
head(mpg)
```

```
## # A tibble: 6 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 audi         a4      1.8  1999     4 auto(l5)   f        18    29 p     compa~
## 2 audi         a4      1.8  1999     4 manual(m5) f        21    29 p     compa~
## 3 audi         a4      2    2008     4 manual(m6) f        20    31 p     compa~
## 4 audi         a4      2    2008     4 auto(av)   f        21    30 p     compa~
## 5 audi         a4      2.8  1999     6 auto(l5)   f        16    26 p     compa~
## 6 audi         a4      2.8  1999     6 manual(m5) f        18    26 p     compa~
```

Quiz questions

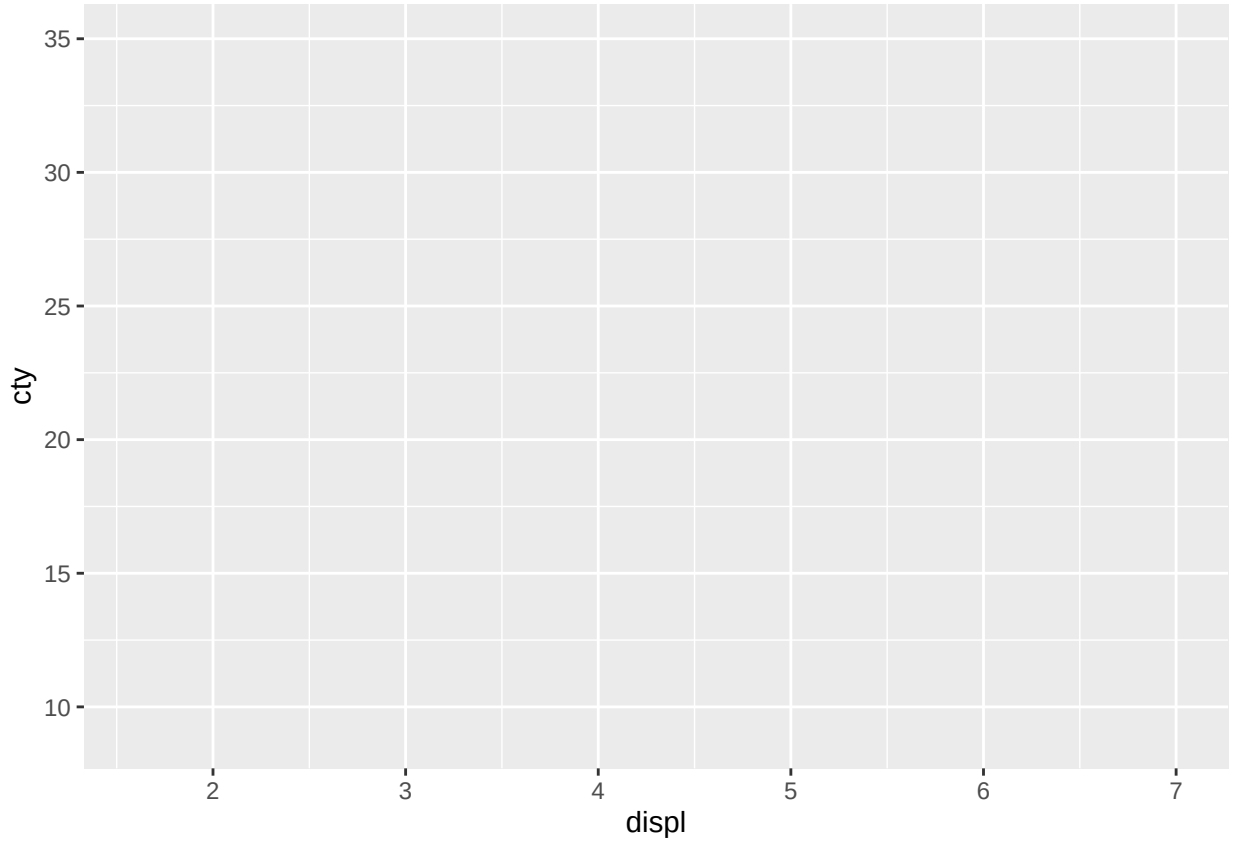
1. What type of variable is `cty`?

2. What type of variable is `displ`?

So, we will start with a scatterplot of city miles per gallon (`cty`) against engine displacement (`displ`).

Start by setting up the canvas:

```
ggplot(mpg,aes(x = displ, y = cty))
```

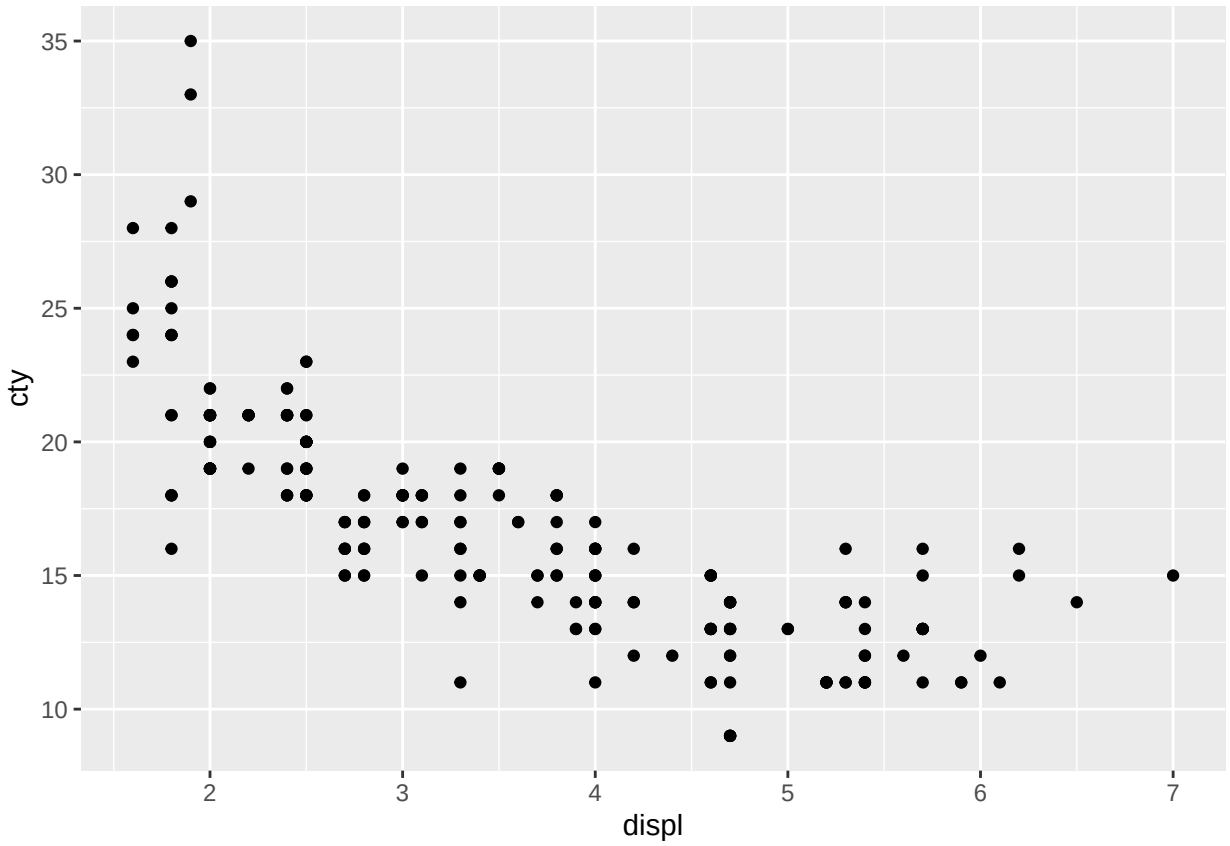


The general form is

- `ggplot()`: the basic command,
- `mpg`: the first argument is the dataframe,
- `aes()`: this command tells ggplot how you are going to match variables to aesthetics, in this case, disp is the x-axis, and cty is the y-axis.

Next we add something to it, in a scatterplot - points:

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



Quiz questions

3. Describe the relationship between `displ` and `cty`.

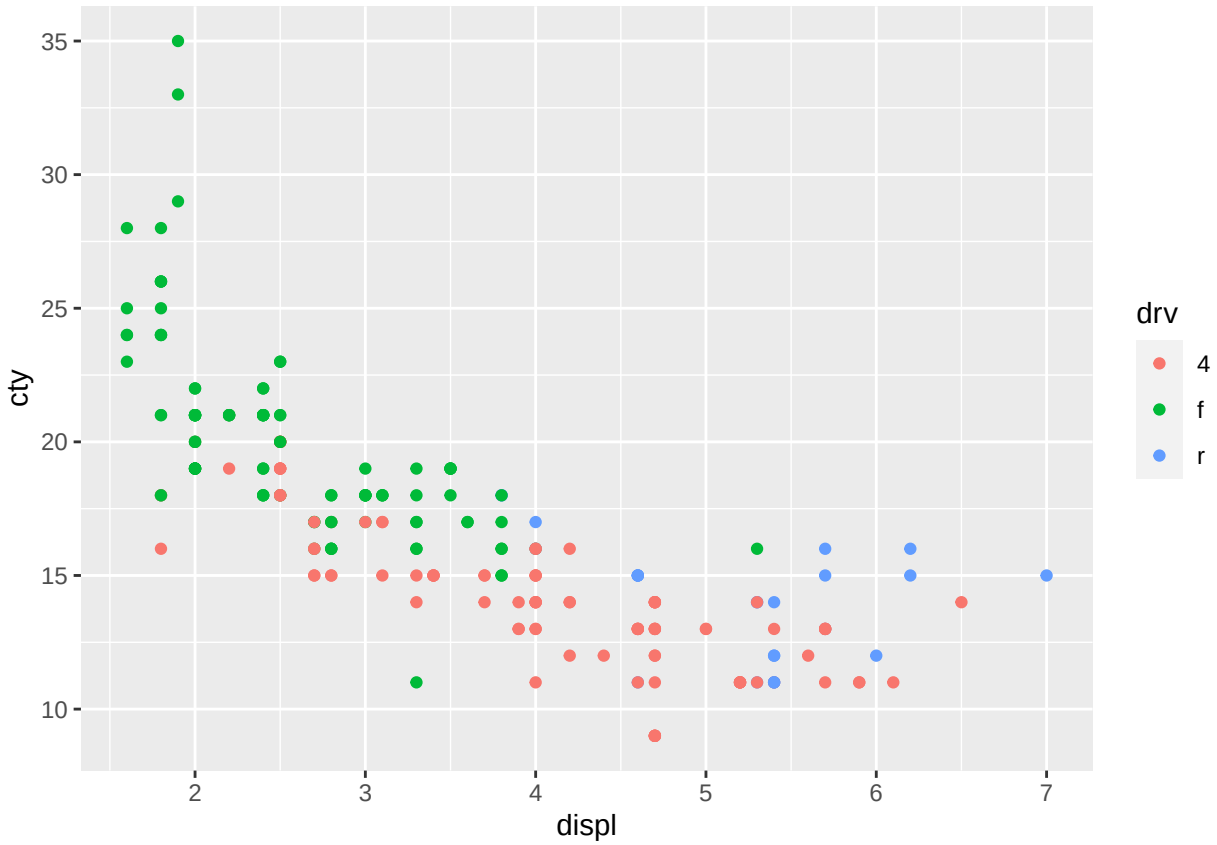
Colour me pink

The first adaption that we can make is to add colour. Let's look at the relationship between `cty` and `displ` for the different types of drives (`drv`).

Quiz questions

4. What type of variable is drive?
5. Describe the relationship between `displ` and `cty` for each of the drives.

```
ggplot(mpg,aes(x = displ, y = cty, col = drv)) +
  geom_point()
```



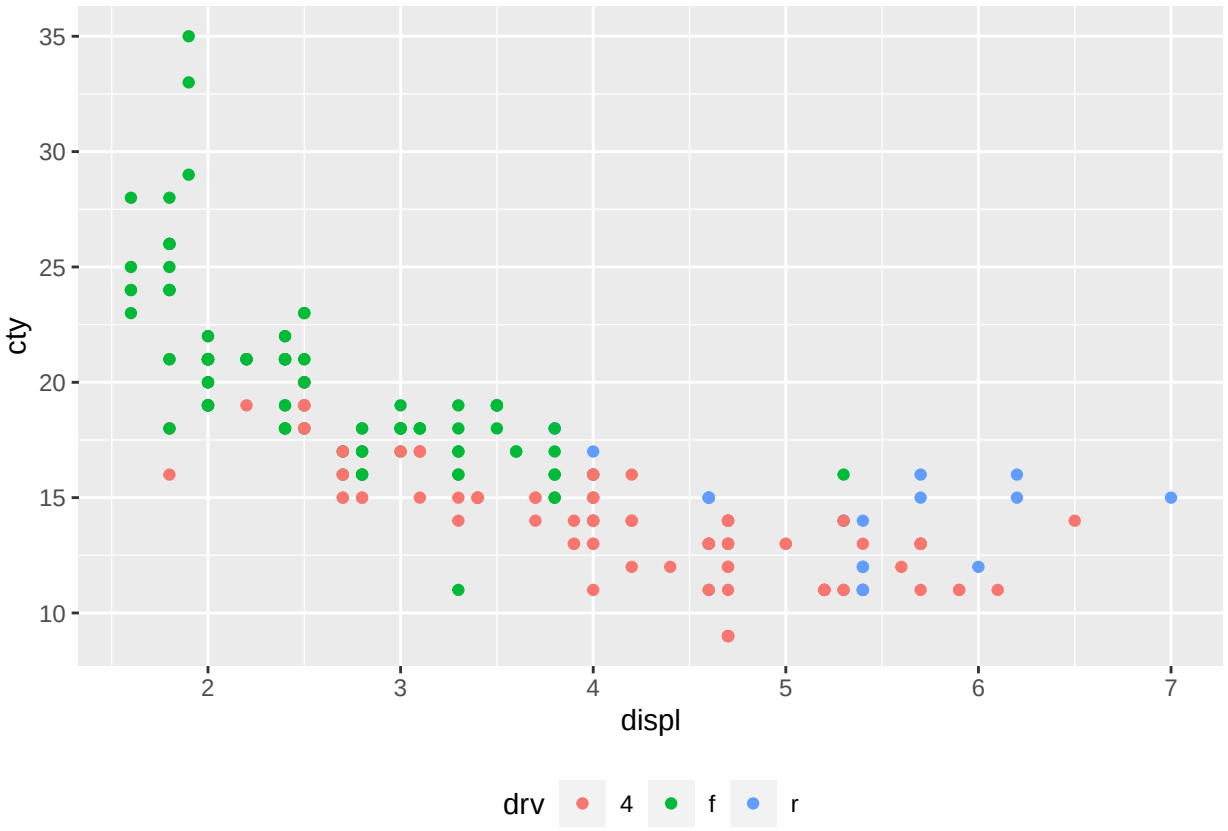
So we want to map the column called `drv` to the aesthetic colour. Note how we do this by adding it into the `aes()` part.

You are a legend

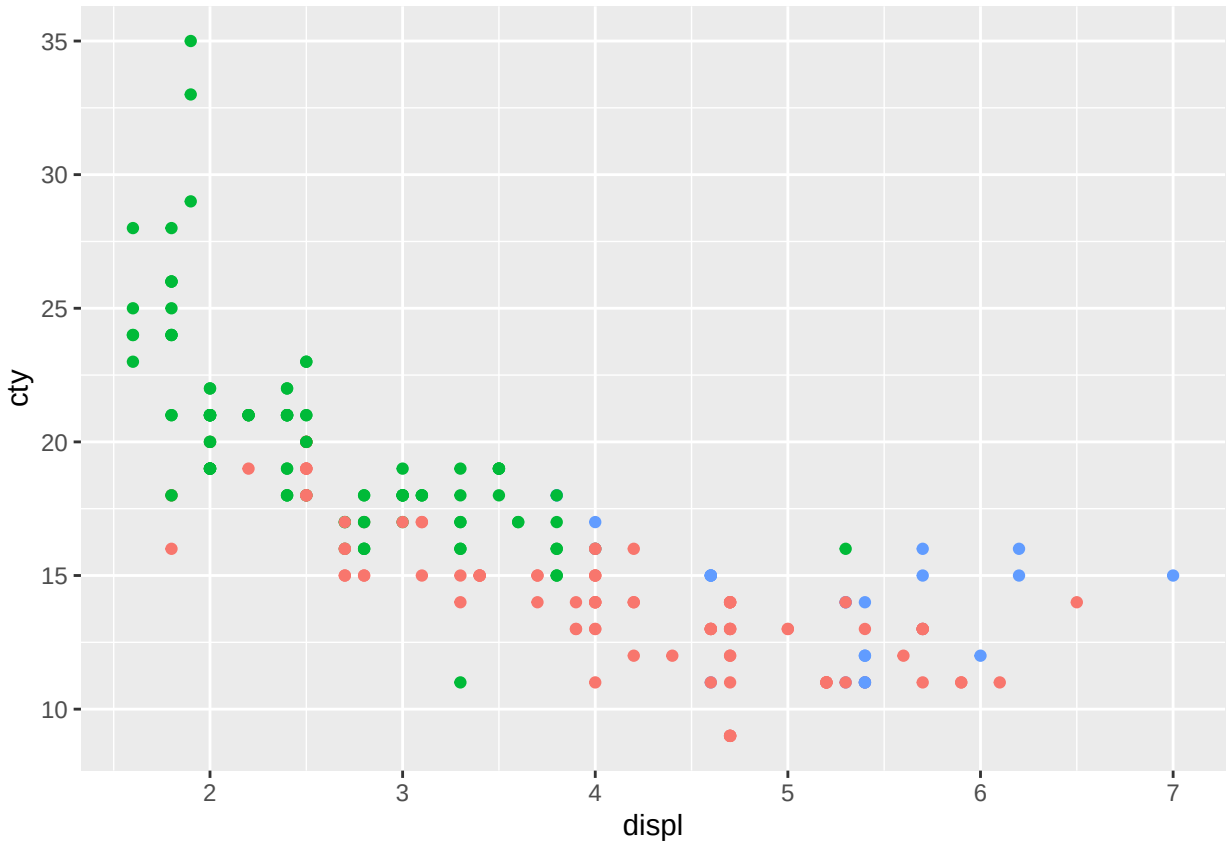
Notice how the previous commands automatically gave us a legend. This is very nice.

Quiz questions

6. Have a Google search and see if you can find how to move it to the bottom.



7. Try to remove the legend.

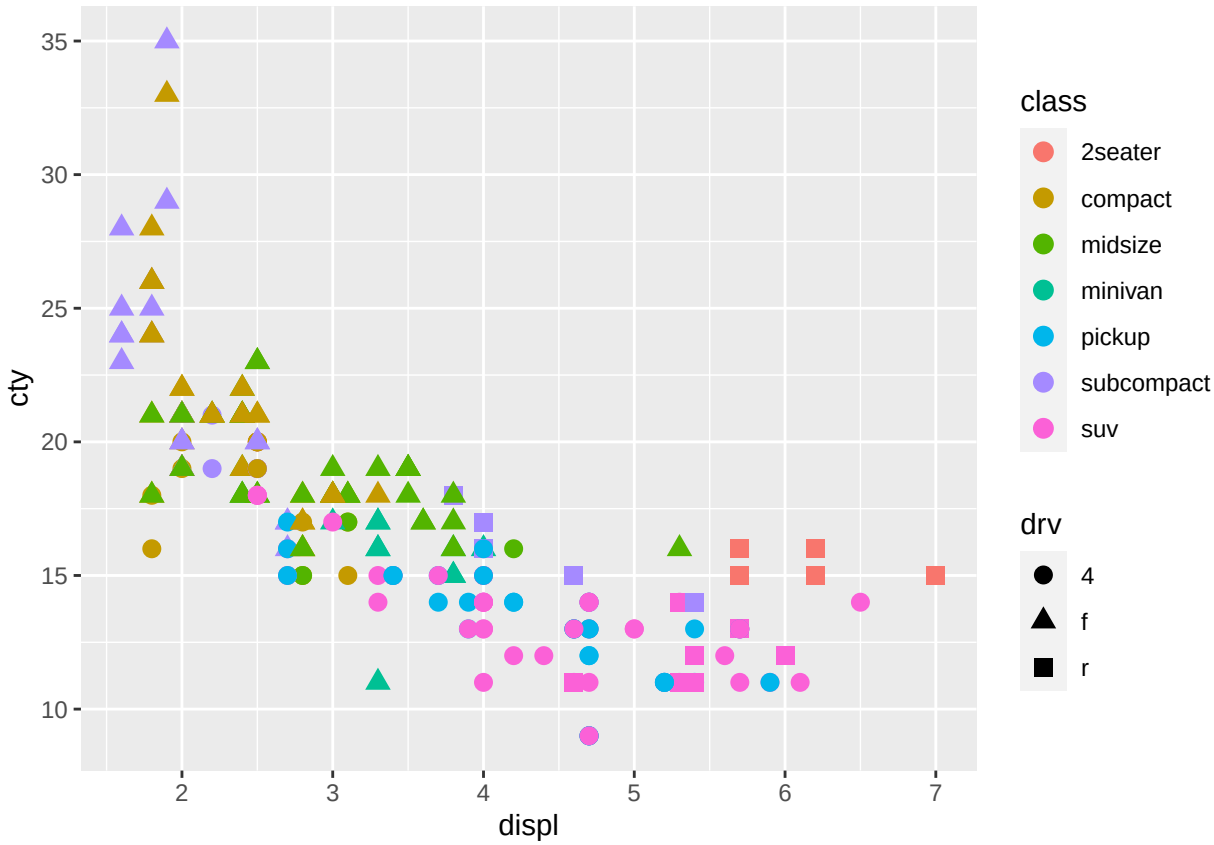


Round dot in a square canvas

Colour is not the only aesthetic of the plot that we can change. For example, we can also change the shapes of the data points.

Quiz questions

8. See if you can produce the following plot:

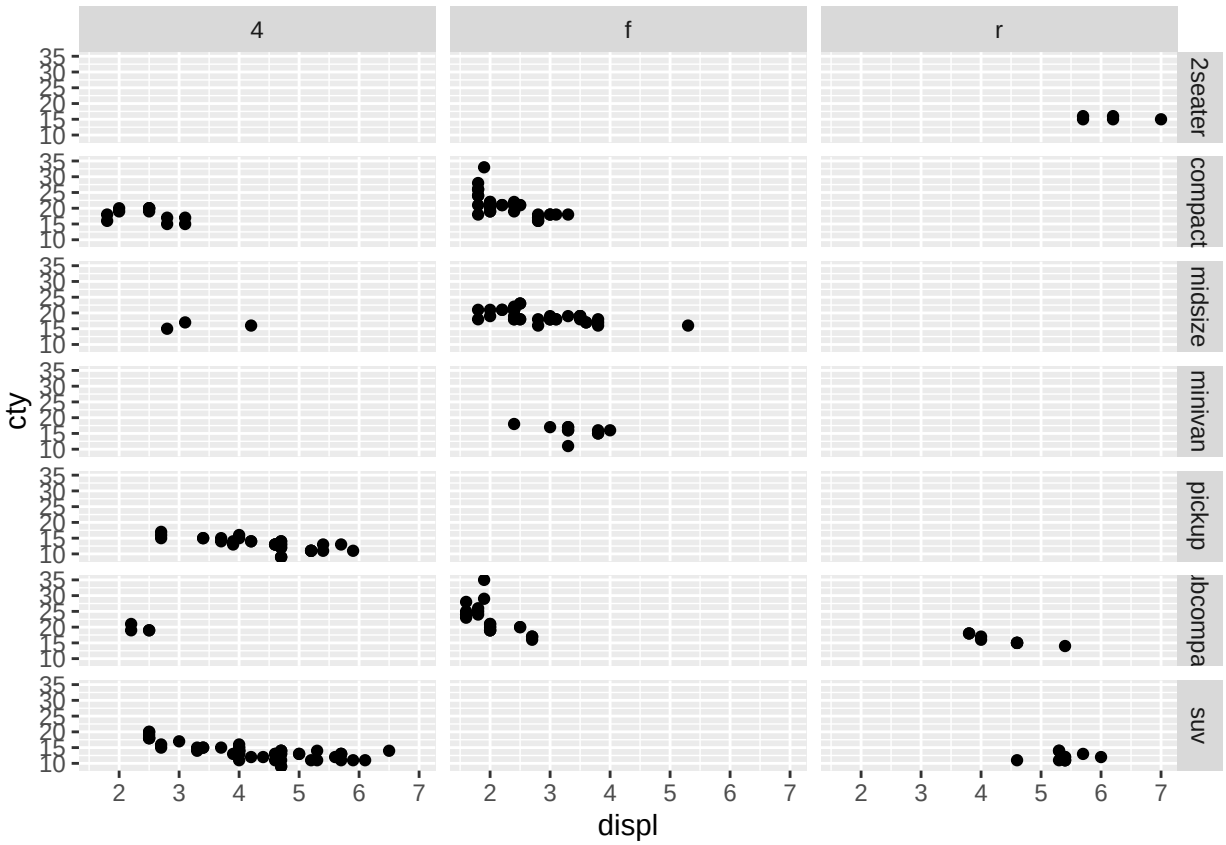


Plots and plots as far as the eye can see

The last plot is far too messy, so to help, we are going to split the plot into multiple plots. There are two commands that help with this - `facet_wrap()` and `facet_grid()`.

Let's illustrate with code.

```
ggplot(mpg,aes(x = displ, y = cty)) + geom_point() +
  facet_grid(class~drv)
```



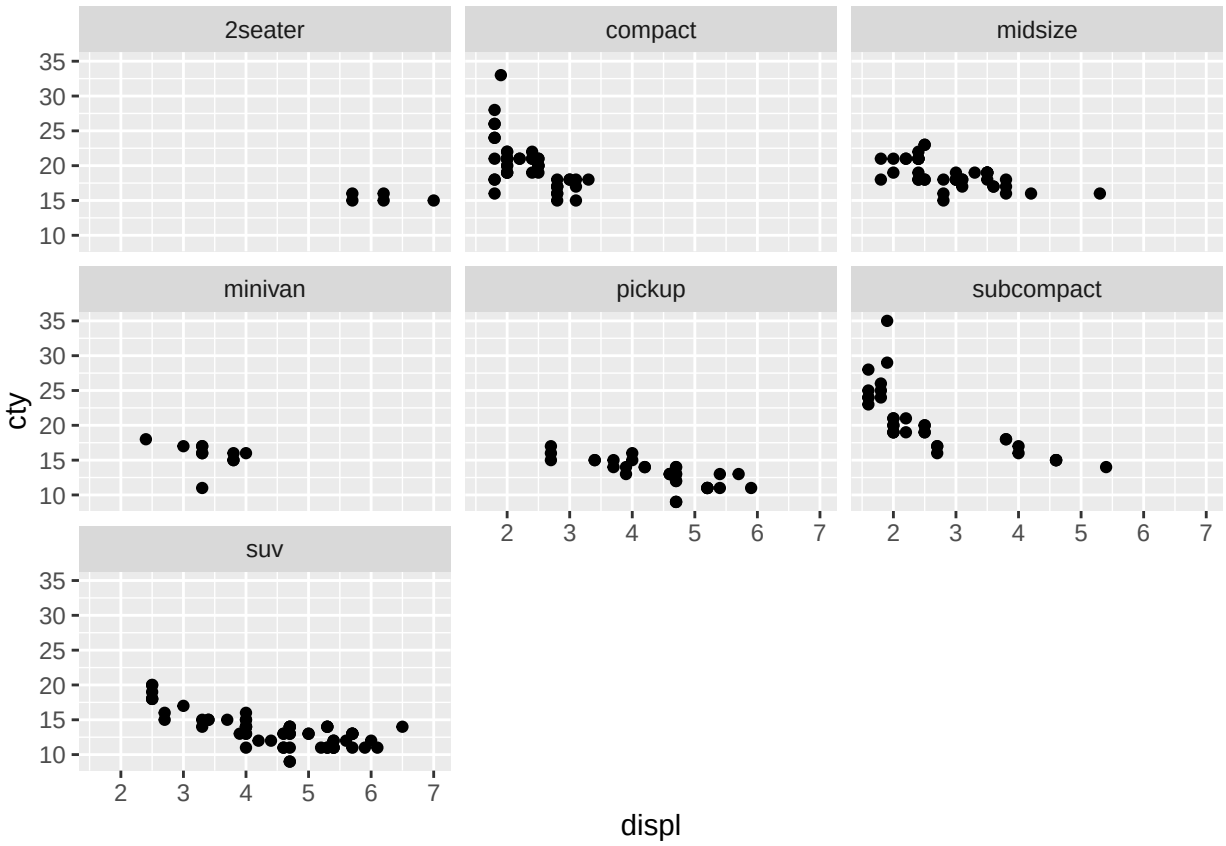
We now have a grid of plots. The command `class ~ drv` is like the formula you used for linear models - the variable before the `~` splits the plots in the y-direction, while the variable after the `~` splits the plots in the x-direction.

Quiz questions

9. Describe the scatterplots.

If we only want to split one variable, then we use `facet_wrap()`:

```
ggplot(mpg,aes(x = displ, y = cty)) + geom_point() +
  facet_wrap(~class)
```

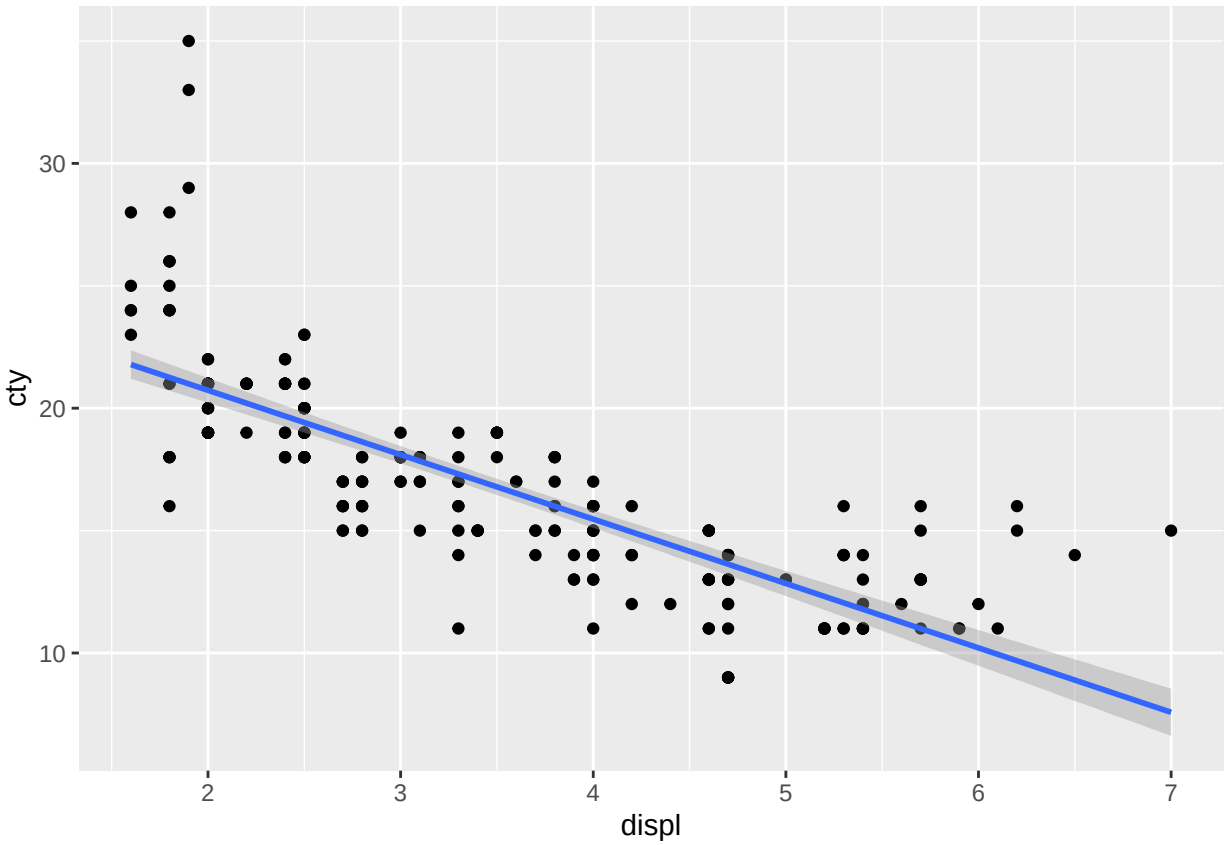


Notice that we still use the ~.

To summarise

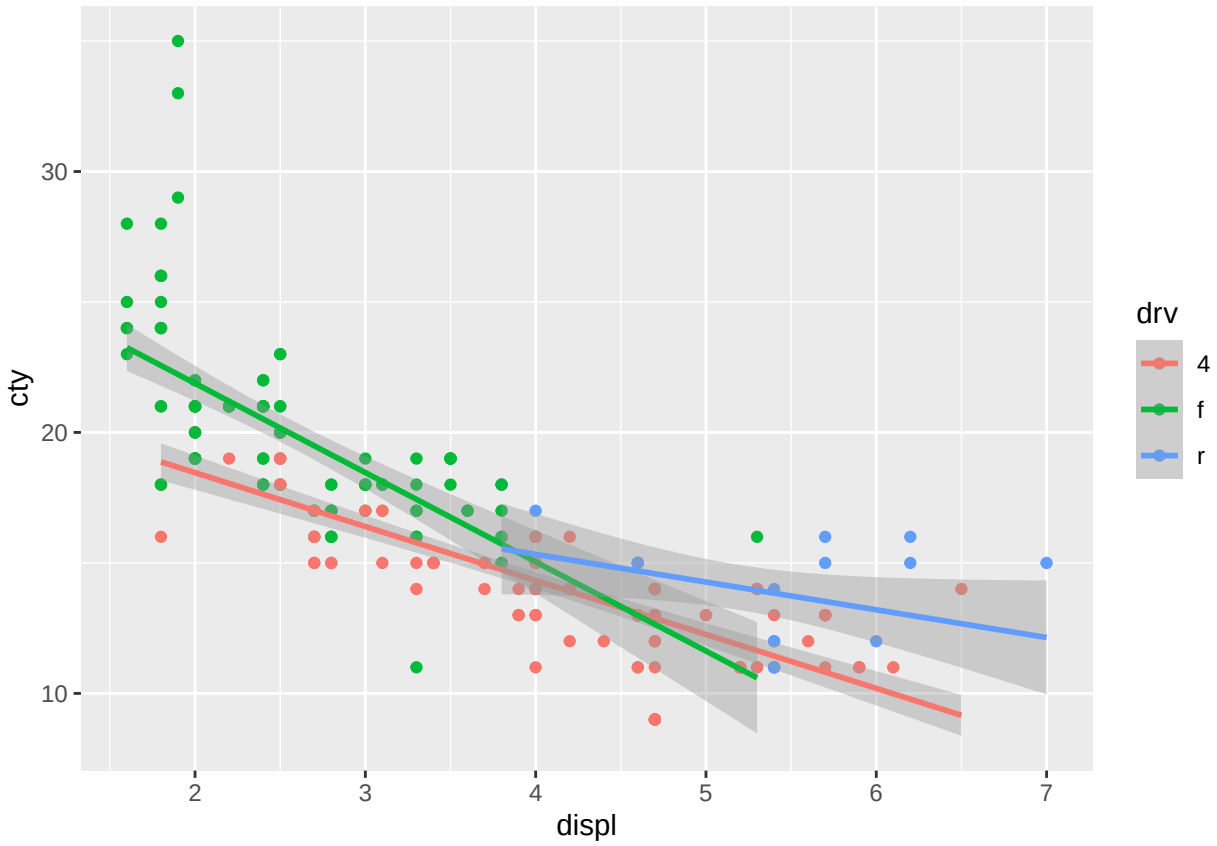
What about adding some summary statistics to plots? First we will add lines of best fit to our scatterplot of cty on displ:

```
ggplot(mpg,aes(x = displ, y = cty)) +
  geom_point() +
  geom_smooth(method = "lm")
```



Now add a line of best fit for each type of drive in the scatterplot of cty on displ.

```
ggplot(mpg,aes(x = displ, y = cty, col  = drv)) +
  geom_point() +
  geom_smooth(method = "lm")
```

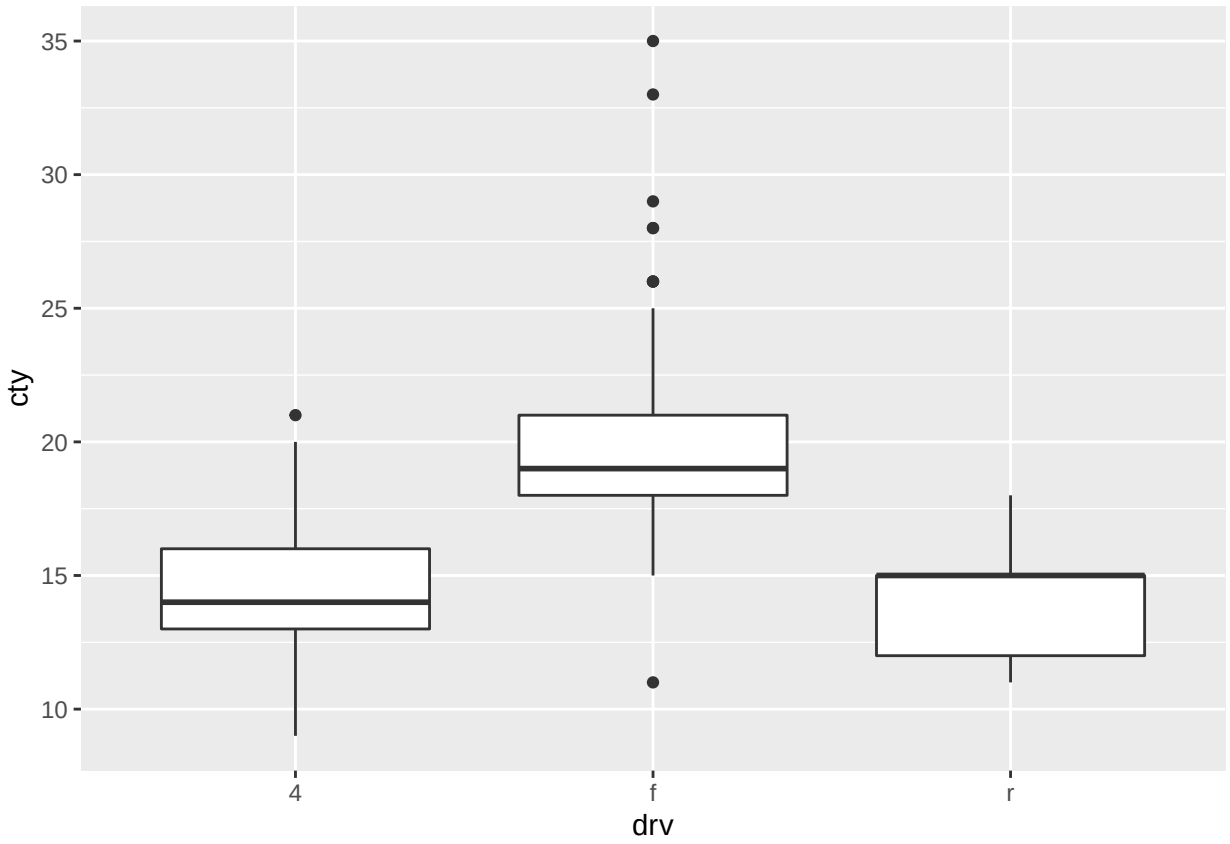


Quiz questions

10. Comment on what you see.

Finally, boxplots are just summaries of the data. We will now produce boxplots of cty for each type of drive.

```
ggplot(mpg,aes(x = drv, y = cty)) + geom_boxplot()
```

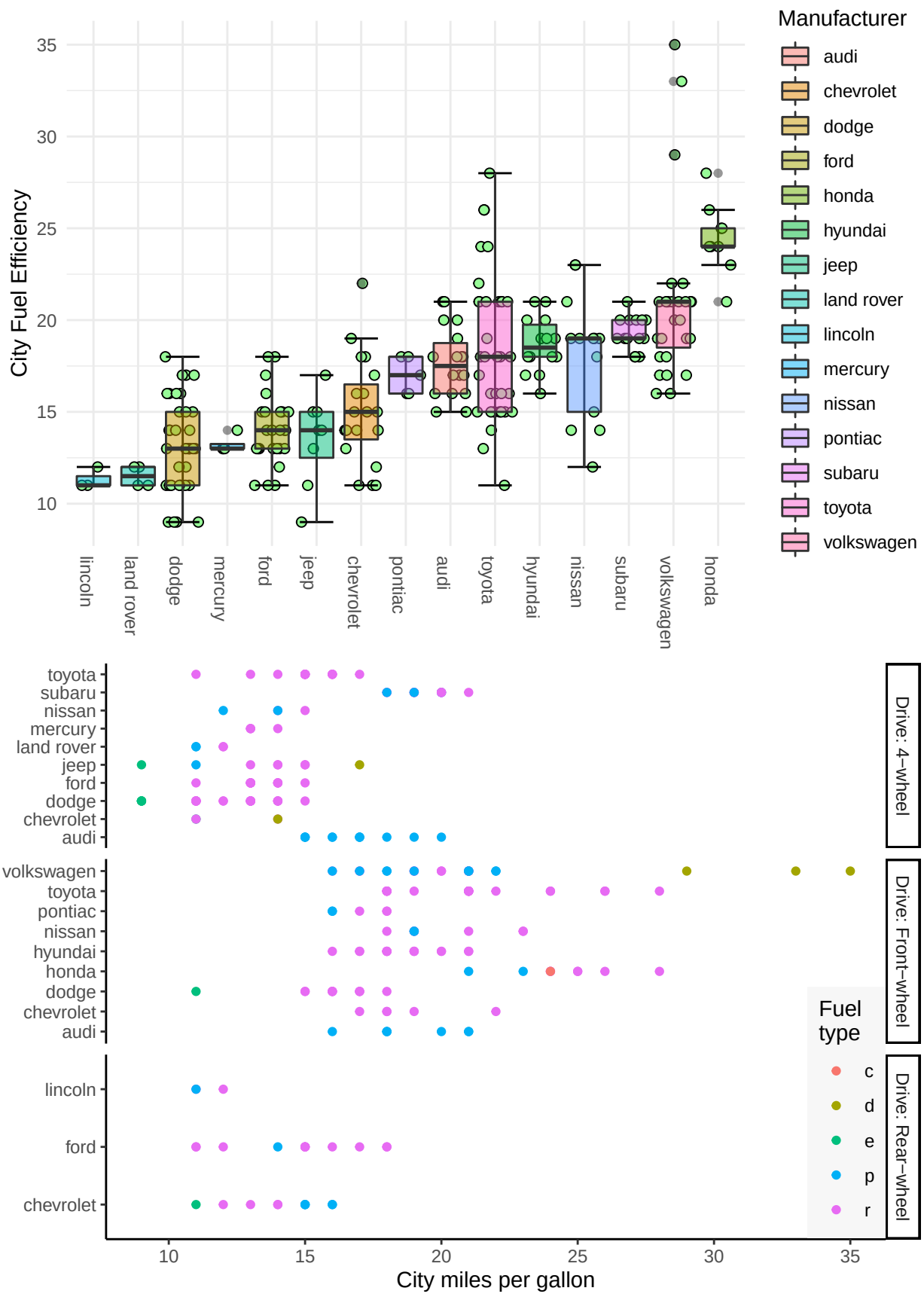


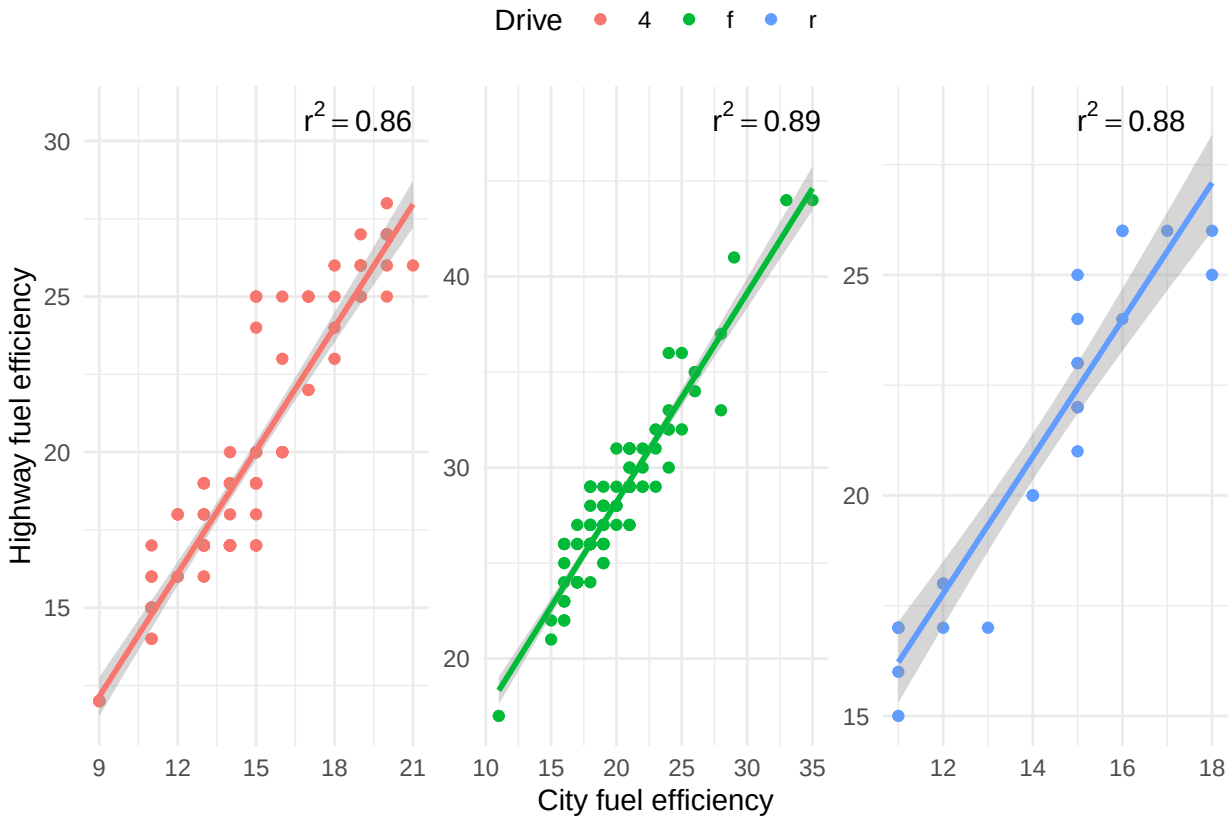
Quiz questions

11. Compare the distributions of `cty` for each type of drive.

Challenges

Now try and recreate the following plots:


City Fuel Efficiency
35
30
25
20
15
10
lincoln
land rover
dodge
mercury
ford
jeep
chevrolet
pontiac
audi
toyota
hyundai
nissan
subaru
volkswagen
honda
Manufacturer
audi
chevrolet
dodge
ford
honda
hyundai
jeep
land rover
lincoln
mercury
nissan
pontiac
subaru
toyota
volkswagen
Drive: 4−wheel
Drive: Front−wheel
Drive: Rear−wheel
Fuel type
c
d
e
p
r
City miles per gallon


Drive
4
f
r
$r^2 = 0.86$
$r^2 = 0.89$
$r^2 = 0.88$
Highway fuel efficiency
City fuel efficiency
30
25
20
15
40
30
20
25
20
15
9
12
15
18
21
10
15
20
25
30
35
12
14
16
18