CT474: Smart Grid

Lecture 1: Introduction to Data Science with R

Dr. Jim Duggan,

School of Engineering & Informatics

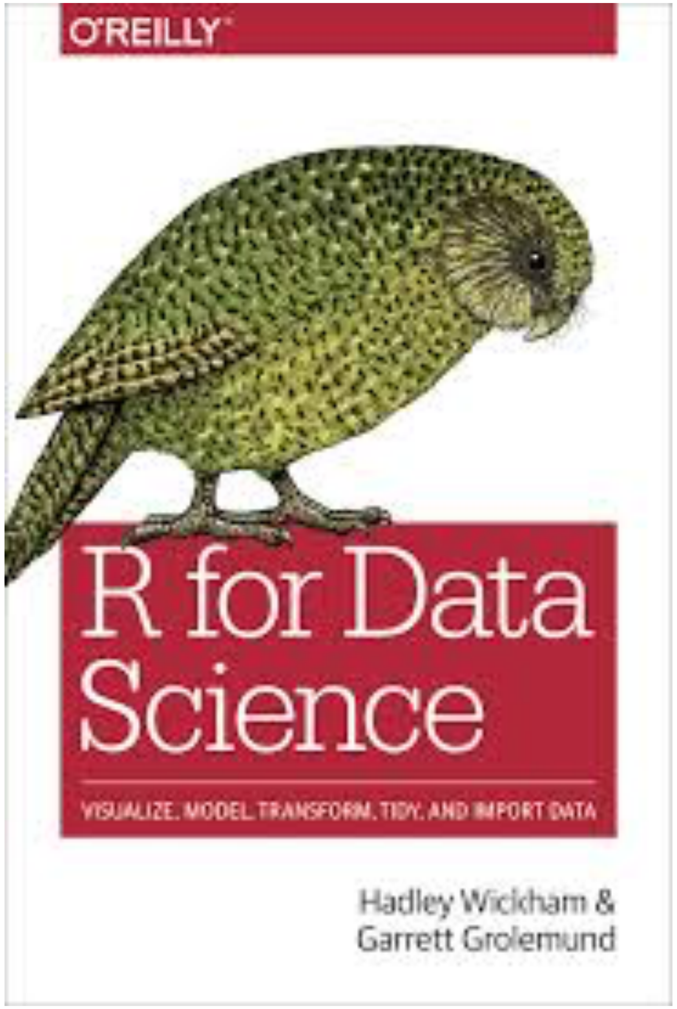
National University of Ireland Galway.

https://github.com/JimDuggan/EnergyAnalytics

https://twitter.com/_jimduggan

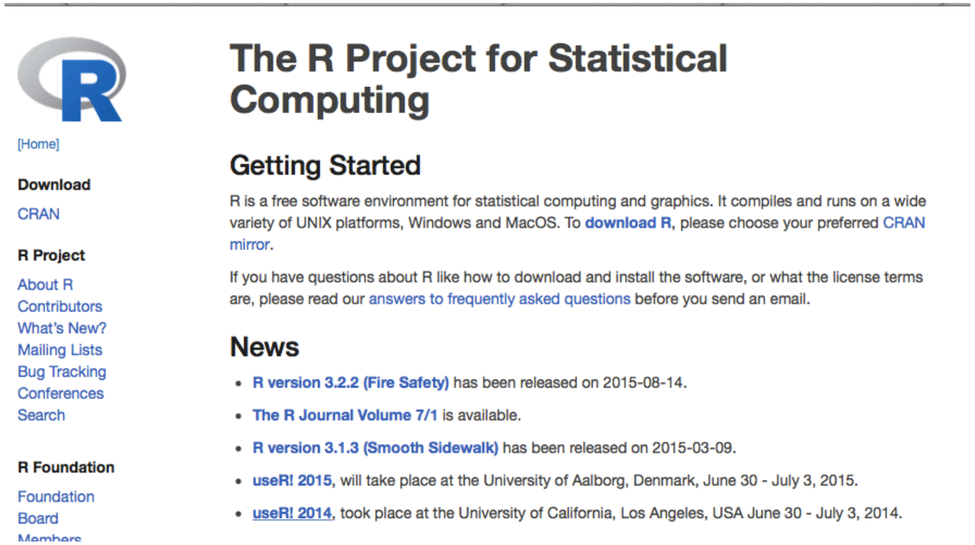
Topic Structure

- Introduction to Data Science and R
  - Data Visualisation
  - Data Transformation
  - Data Modeling
- R Aspects
  - ggplot2
  - dplyr
  - lm
- Energy examples



The R Project for Statistical Computing

- R's *mission* is to enable the best and most thorough exploration of data possible (Chambers 2008).
- It is a dialect of the S language, developed at Bell Laboratories
- ACM noted that *S "will forever alter the way people analyze, visualize, and manipulate data"*



The R Project for Statistical Computing

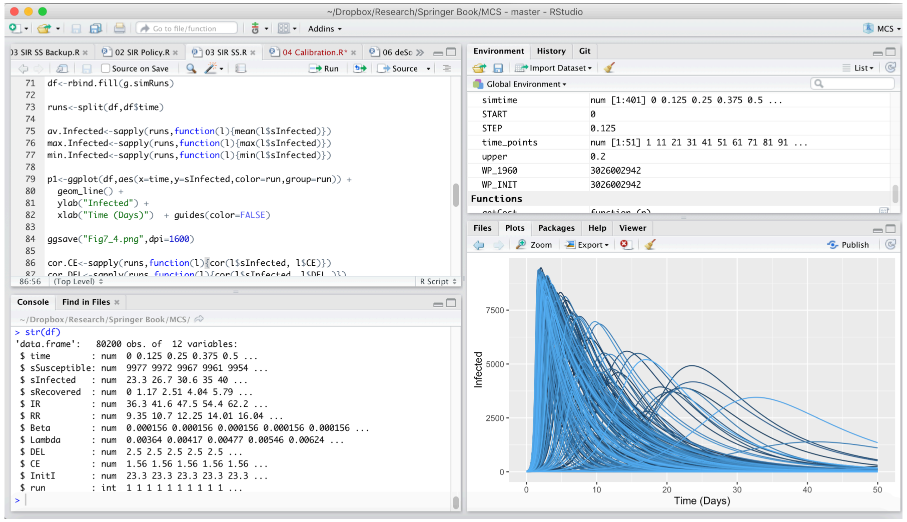
Getting Started

R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. To **download R**, please choose your preferred CRAN mirror.

If you have questions about R like how to download and install the software, or what the license terms are, please read our answers to frequently asked questions before you send an email.

News

- **R version 3.2.2 (Fire Safety)** has been released on 2015-08-14.
- **The R Journal Volume 7/1** is available.
- **R version 3.1.3 (Smooth Sidewalk)** has been released on 2015-03-09.
- **useR! 2015**, will take place at the University of Aalborg, Denmark, June 30 - July 3, 2015.
- **useR! 2014**, took place at the University of California, Los Angeles, USA June 30 - July 3, 2014.



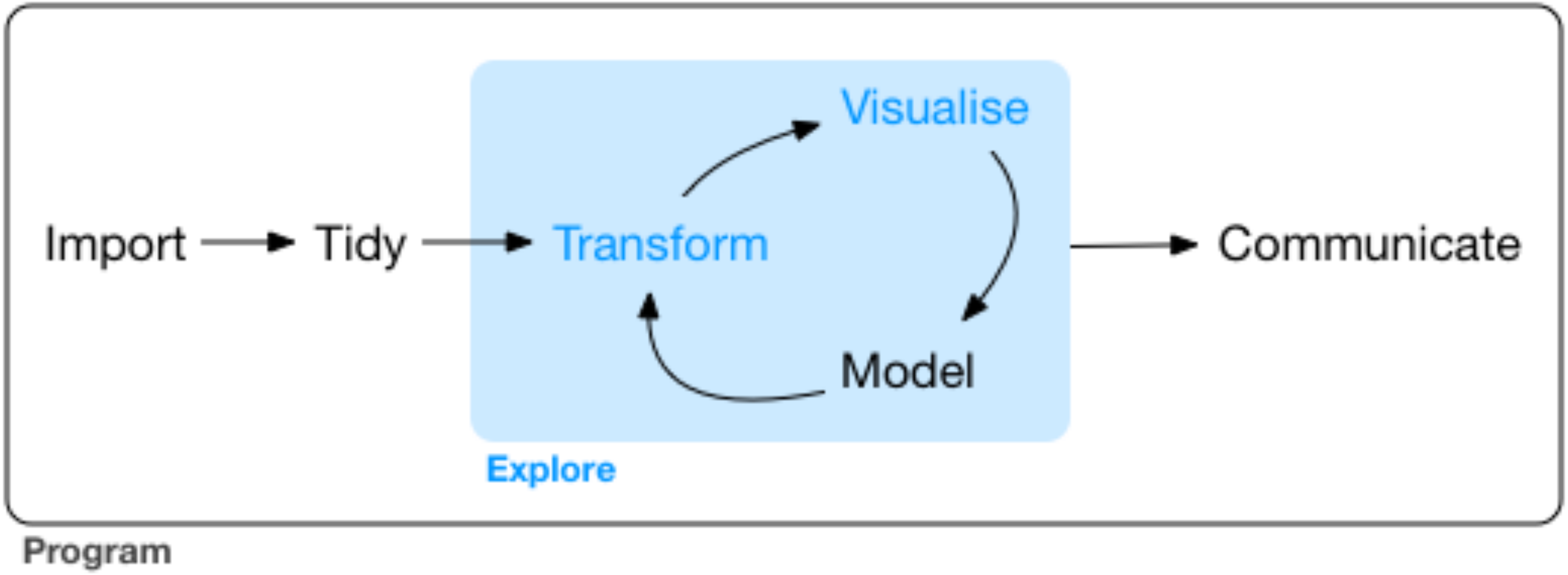
```
df<-rbind.fill(g.simRuns)

runs<-split(df,df$time)

av.Infected<-sapply(runs,function(l){mean(l$sInfected)})
max.Infected<-sapply(runs,function(l){max(l$sInfected)})
min.Infected<-sapply(runs,function(l){min(l$sInfected)})

p1<-ggplot(df,aes(x=time,y=sInfected,color=run,group=run)) +
  geom_line() +
  ylab("Infected") +
  xlab("Time (Days)")  + guides(color=FALSE)

ggsave("Fig7_4.png",dpi=1600)

cor.CE<-sapply(runs,function(l){cor(l$sInfected, l$CE)})
cor.DEL<-sapply(runs,function(l){cor(l$sInfected, l$DEL )})
```

```
> str(df)
'data.frame':   80200 obs. of  12 variables:
 $ time        : num  0 0.125 0.25 0.375 0.5 ...
 $ sSusceptible: num  9977 9972 9967 9961 9954 ...
 $ sInfected   : num  23.3 26.7 30.6 35 40 ...
 $ sRecovered  : num  0 1.17 2.51 4.04 5.79 ...
 $ IR          : num  36.3 41.6 47.5 54.4 62.2 ...
 $ RR          : num  9.35 10.7 12.25 14.01 16.04 ...
 $ Beta        : num  0.000156 0.000156 0.000156 0.000156 0.000156 ...
 $ Lambda      : num  0.00364 0.00417 0.00477 0.00546 0.00624 ...
 $ DEL         : num  2.5 2.5 2.5 2.5 2.5 ...
 $ CE          : num  1.56 1.56 1.56 1.56 1.56 ...
 $ InitI       : num  23.3 23.3 23.3 23.3 23.3 ...
 $ run         : int  1 1 1 1 1 1 1 1 1 1 ...
>
```

```
simtime       num [1:401] 0 0.125 0.25 0.375 0.5 ...
START         0
STEP          0.125
time_points   num [1:51] 1 11 21 31 41 51 61 71 81 91 ...
upper         0.2
WP_1960       3026002942
WP_INIT       3026002942
Functions
getCost       function (n)
```

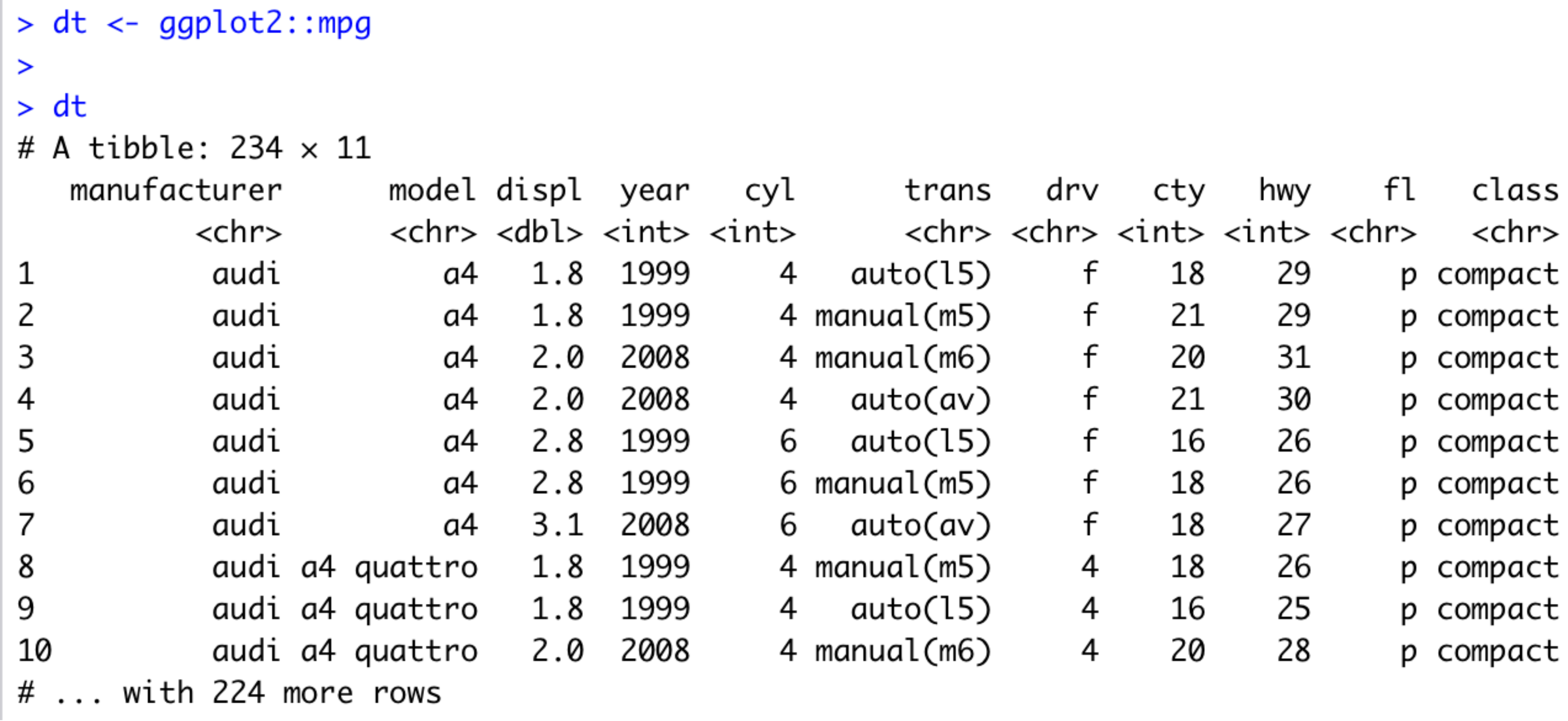
Data Exploration

“Data exploration is the art of looking at your data, rapidly generating hypotheses, quickly testing them, then repeating again and again and again.” (Wickham and Grolemund 2017).



Data Visualisation with **ggplot2**

“The simple graph has brought more information to the data analyst’s mind that any other device.” – John Tukey



```
> dt <- ggplot2::mpg
>
> dt
# A tibble: 234 × 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 compact
2          audi         a4   1.8  1999     4 manual(m5)     f    21    29     p compact
3          audi         a4   2.0  2008     4 manual(m6)     f    20    31     p compact
4          audi         a4   2.0  2008     4   auto(av)     f    21    30     p compact
5          audi         a4   2.8  1999     6   auto(l5)     f    16    26     p compact
6          audi         a4   2.8  1999     6 manual(m5)     f    18    26     p compact
7          audi         a4   3.1  2008     6   auto(av)     f    18    27     p compact
8          audi a4 quattro   1.8  1999     4 manual(m5)     4    18    26     p compact
9          audi a4 quattro   1.8  1999     4   auto(l5)     4    16    25     p compact
10         audi a4 quattro   2.0  2008     4 manual(m6)     4    20    28     p compact
# ... with 224 more rows
```

Fuel Economy Data Set (ggplot2::mpg)

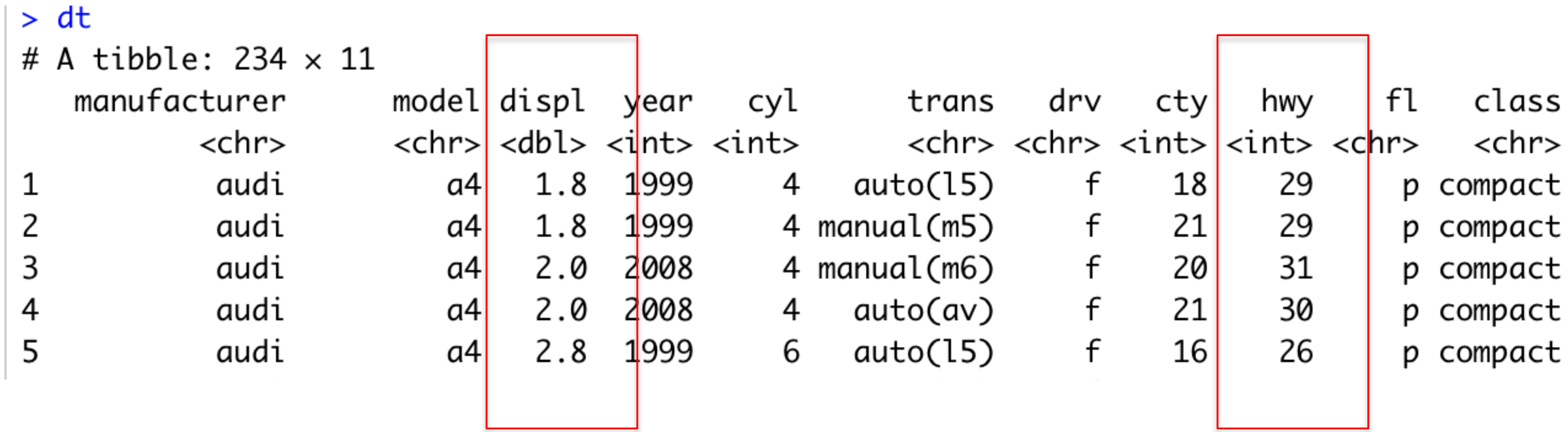
This dataset contains a subset of the fuel economy data that the EPA makes available on http://fueleconomy.gov. It contains only models which had a new release every year between 1999 and 2008 - this was used as a proxy for the popularity of the car.

manufacturer	manufacturer	**drv**	f = front-wheel drive, r = rear wheel drive, 4 = 4wd
model	model name	**cty**	city miles per gallon
displ	engine displacement, in litres	**hwy**	highway miles per gallon
year	year of manufacture	**fl**	fuel type
cyl	number of cylinders	**class**	“type” of car
trans	type of transmission		

First Steps

- Generate a first graph to help answer the following question:
  - *Do cars with big engines use more fuel than cars with small engines*
- What might the relationship between engine size and fuel efficiency look like?
  - Positive or negative?
  - Linear or non-linear?

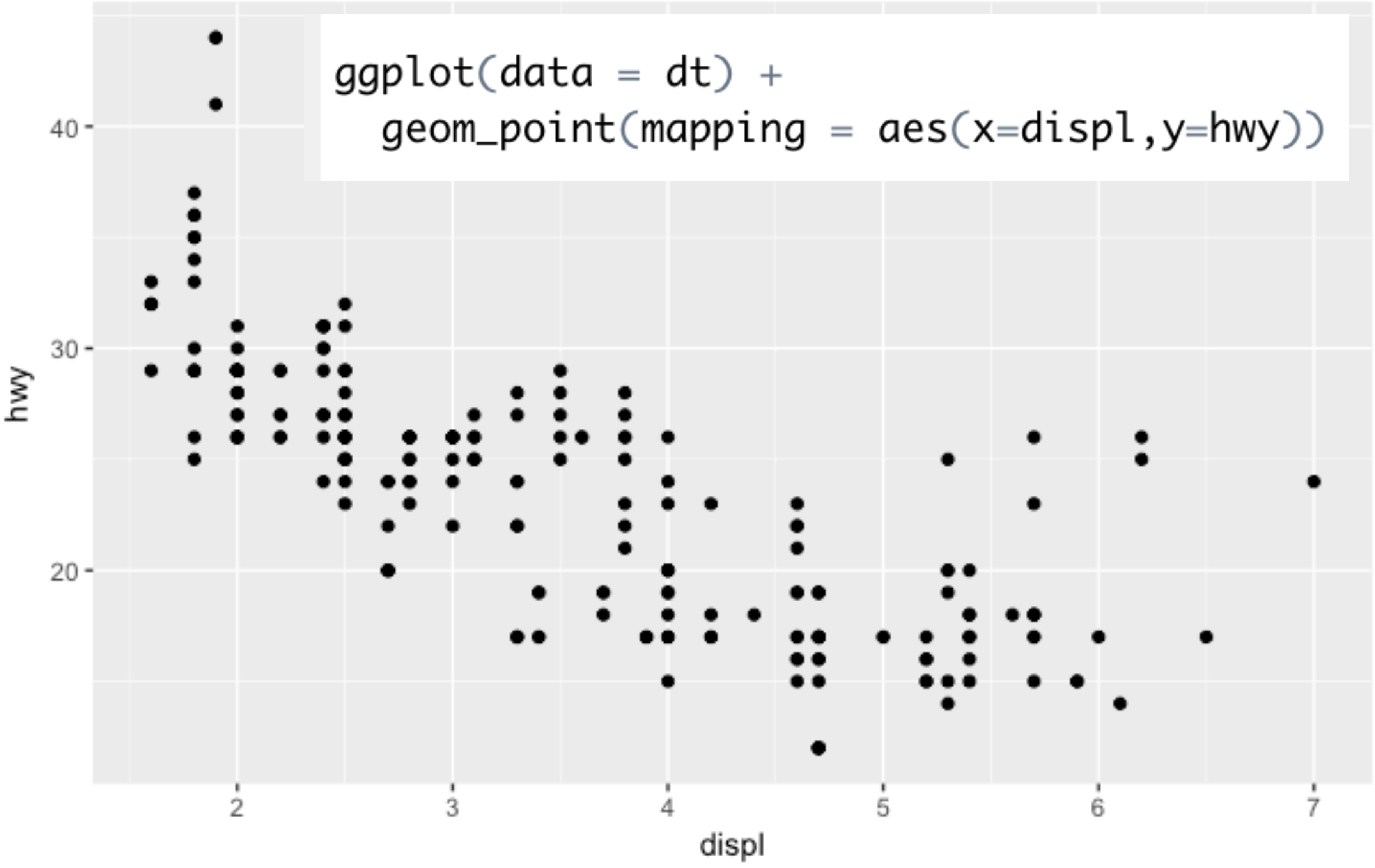
Selecting data



```
> dt
# A tibble: 234 × 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 compact
2          audi         a4   1.8  1999     4 manual(m5)     f    21    29     p compact
3          audi         a4   2.0  2008     4 manual(m6)     f    20    31     p compact
4          audi         a4   2.0  2008     4   auto(av)     f    21    30     p compact
5          audi         a4   2.8  1999     6   auto(l5)     f    16    26     p compact
```

- Among the variables are:
  - **displ**, a car’s engine size in litres
  - **hwy**, a car’s fuel efficiency on the highway in miles per gallon

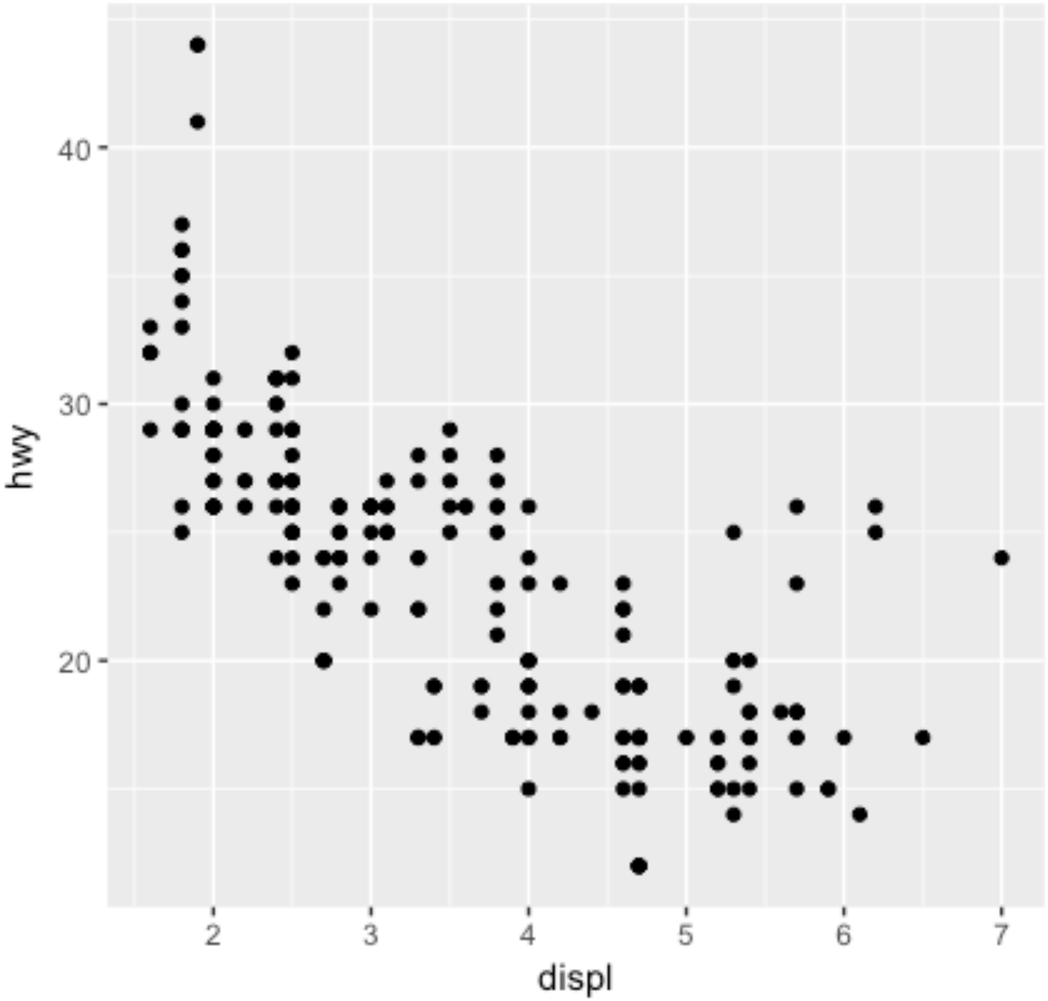
Creating a ggplot



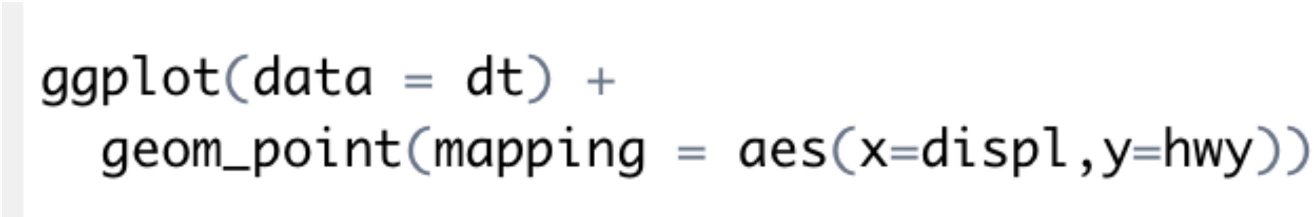
```
ggplot(data = dt) +
  geom_point(mapping = aes(x=displ,y=hwy))
```

Interpreting the plot

- The plot shows a negative relationship between engine size (displ) and fuel efficiency (hwy)
- Cars with big engines use more fuel
- Does this confirm or refute your hypothesis about fuel efficiency and engine size?



A Graphing Template in R



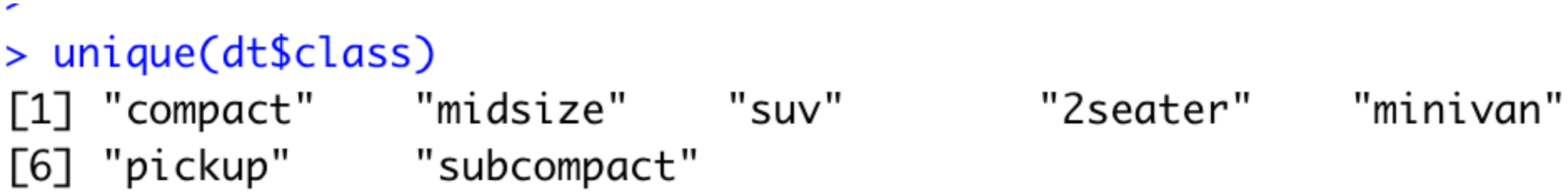
```
ggplot(data = dt) +
  geom_point(mapping = aes(x=displ,y=hwy))
```

- Turn the code into a reusable template for making graphs with ggplot2

```
ggplot(data = <DATA>) +
    <GEOM_FUNCTION>(mapping = aes(<MAPPINGS>))
```

Aesthetic Mappings

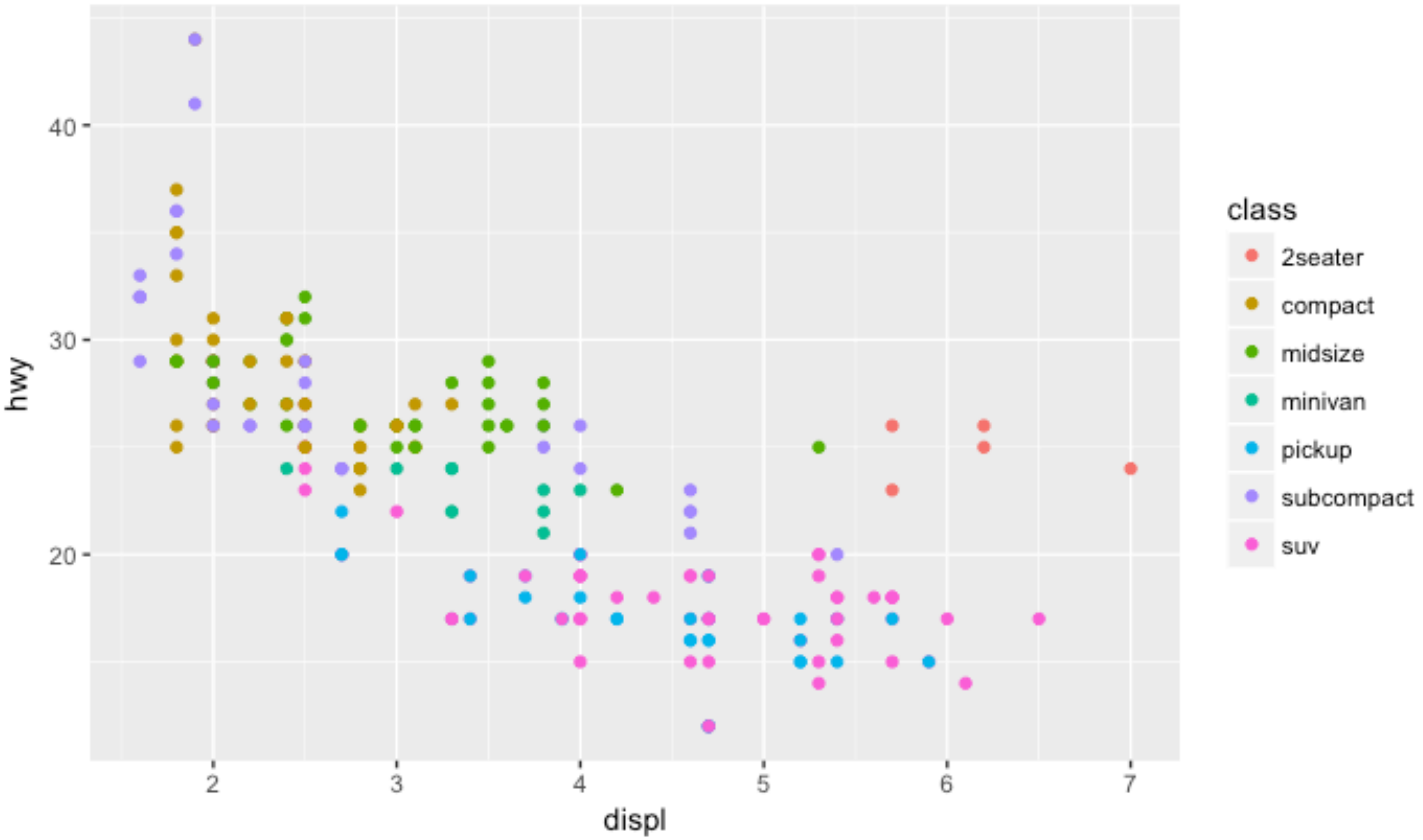
> “The greatest value of a picture is when it forces us to notice what we never expected to see” – John Tukey



```
> unique(dt$class)
[1] "compact"    "midsize"    "suv"        "2seater"    "minivan"
[6] "pickup"     "subcompact"
```

- A third variable can be added to a 2-D plot by mapping it to an aesthetic.
- An aesthetic is a visual property of the plot’s objects.
- An aesthetic’s *level* could be colour, size or shape.

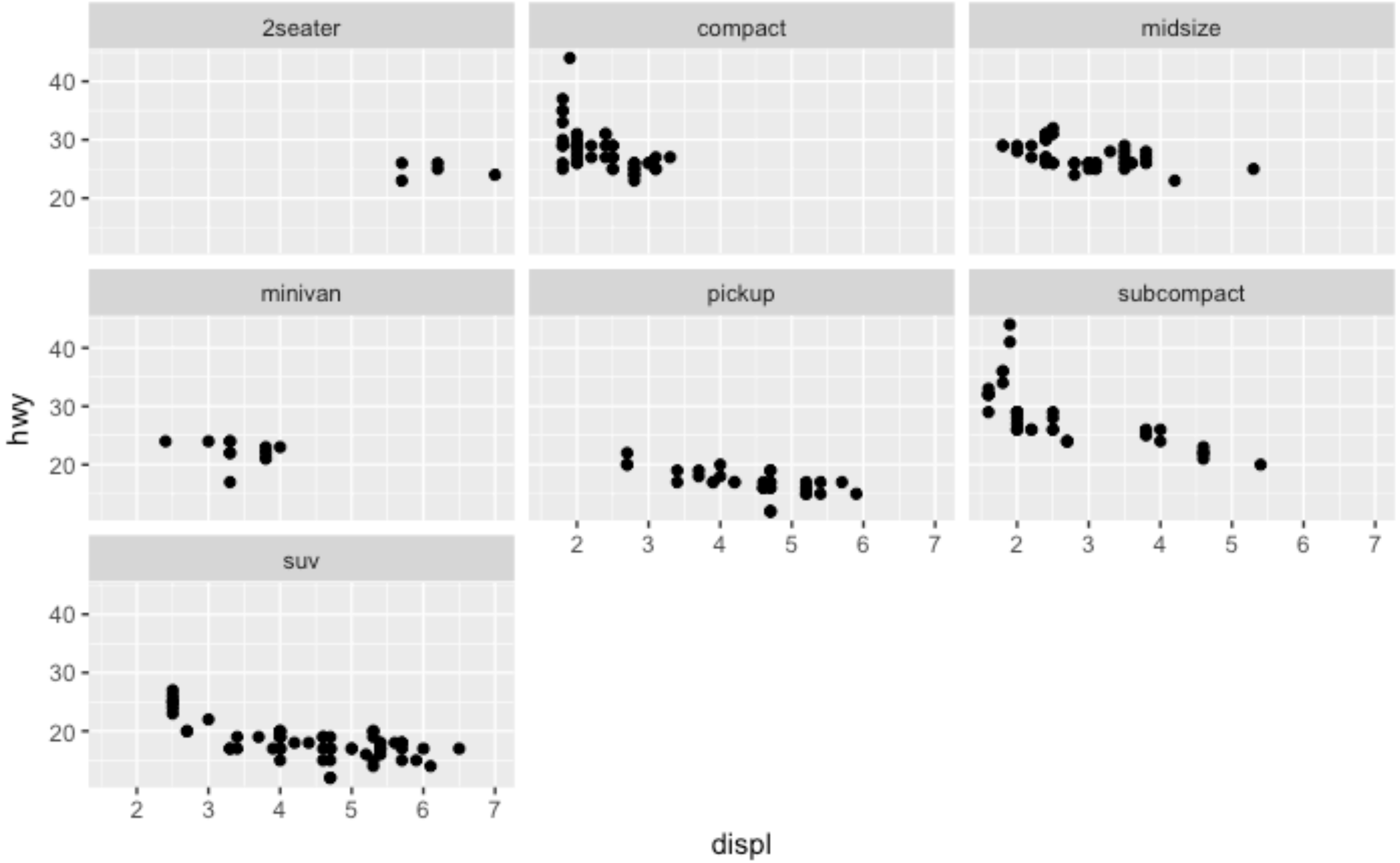
```
ggplot(data = dt) +
  geom_point(mapping = aes(x=displ,y=hwy,colour=class))
```

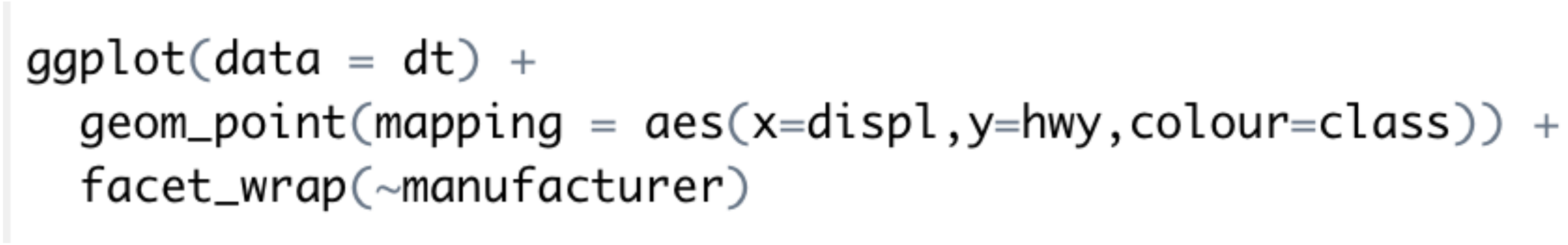


Facets

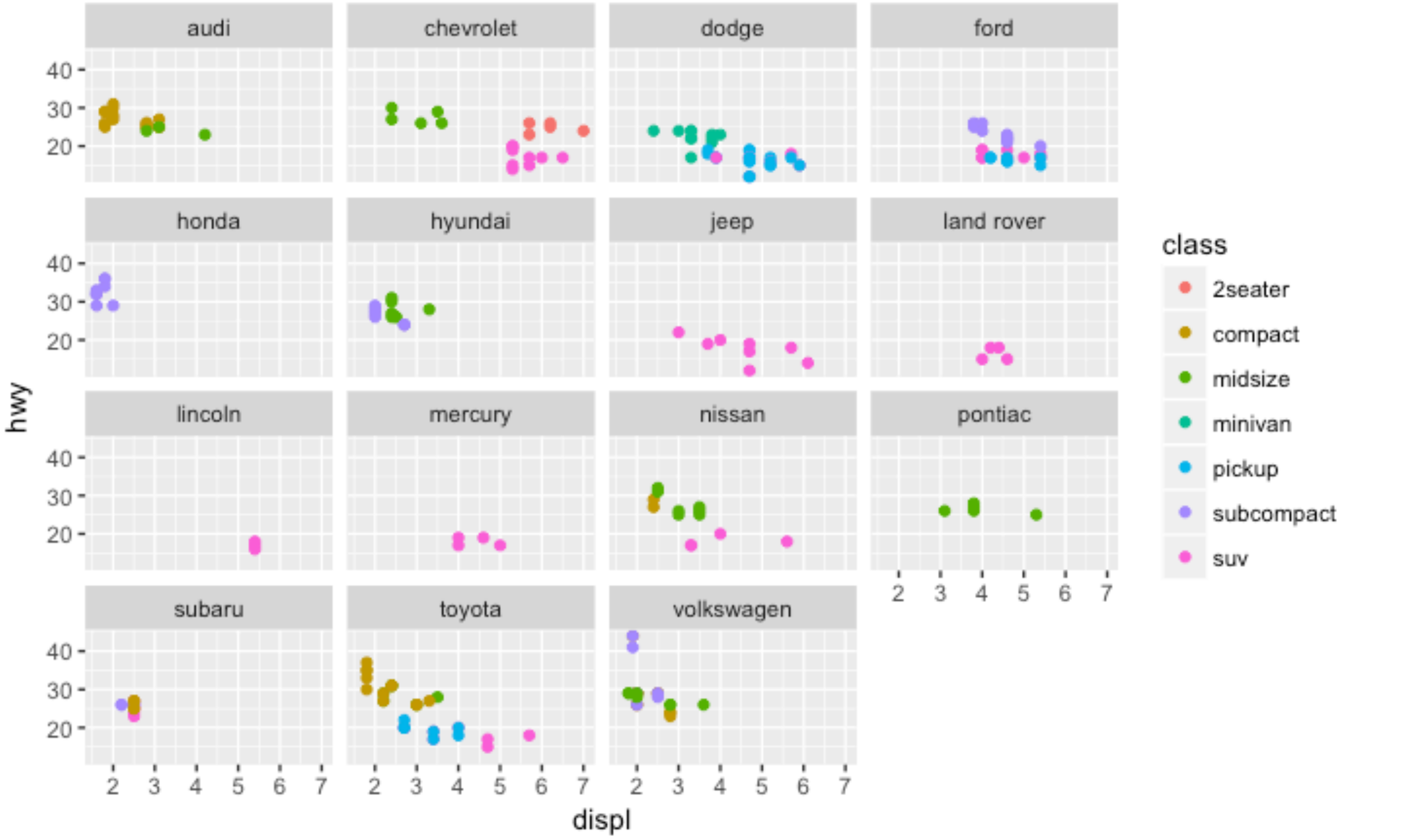
- Another way to add categorical variables is to split a plot into facets, subplots that display one subset of the data.
- To facet your plot by a single variable, use facet_wrap()

```
ggplot(data = dt) +
  geom_point(mapping = aes(x=displ,y=hwy)) + facet_wrap(~class)
```





```
ggplot(data = dt) +
  geom_point(mapping = aes(x=displ,y=hwy,colour=class)) +
  facet_wrap(~manufacturer)
```



Challenge 1.1

- Explore the faithful data set
- x = waiting time
- y = eruption time
- Add a linear model