Introduction to R and the tidyverse

– plotting part 1 –

Paolo Crosetto

Why plot?

Why do we plot

> *Why do we want to `plot` data?*

- we are human beings – we are `pattern recognizers`
- we can *see* things we are not able to grasp from data
- good to `explore` a dataset and look for regularities
- good to `convey` *a clear message*
- to have `fun`

Why plot? Eyeballing

> `Eyeballing` the data first is **always** a good idea

- data could look `similar` at a first glance
- and even have similar descriptive statistics
- but still be *`very`*`different` in practice

An example

- data contains vars `x` and `y`, over 13 different conditions
- import `data/plotme.tsv`, compute μ, σ by `dataset`

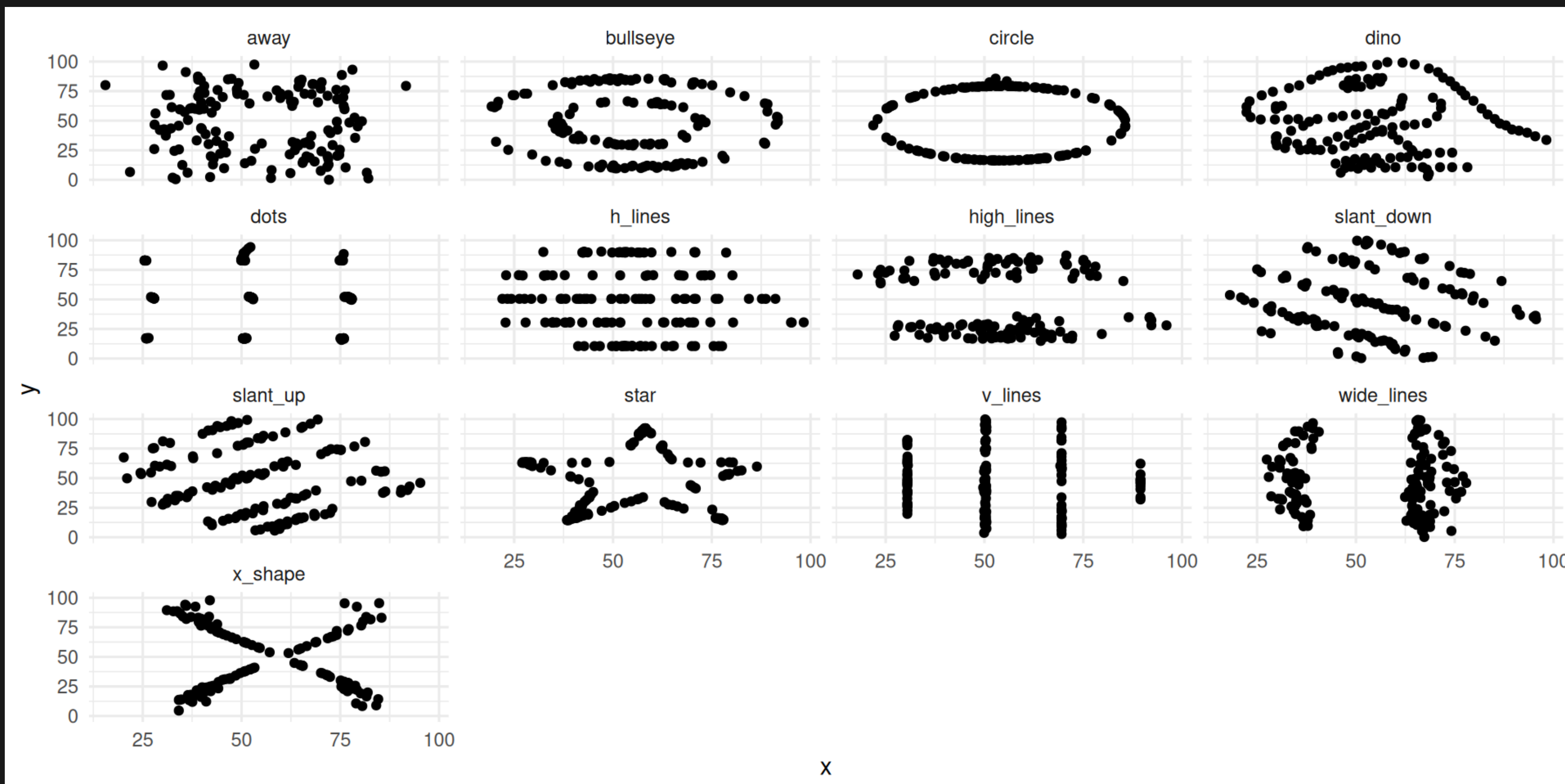
An example

- data contains vars `x` and `y`, over 13 different conditions
- import `data/plotme.tsv`, compute μ, σ by `dataset`

dataset	mean_x	sd_x	mean_y	sd_y
away	54.27	16.77	47.83	26.94
bullseye	54.27	16.77	47.83	26.94
circle	54.27	16.76	47.84	26.93
dino	54.26	16.77	47.83	26.94
dots	54.26	16.77	47.84	26.93

Now let's plot this!

> But if you `plot` it, you'll see **stark** differences



Why plot? Compact information

> Plotting allows you to convey a `lot of info`

- humans are `pattern recognizers`
- several `geometric` objects can convey meaning
  - position (x,y,z)
  - color, size, shape
- you can combine multiple plots into `infographics`

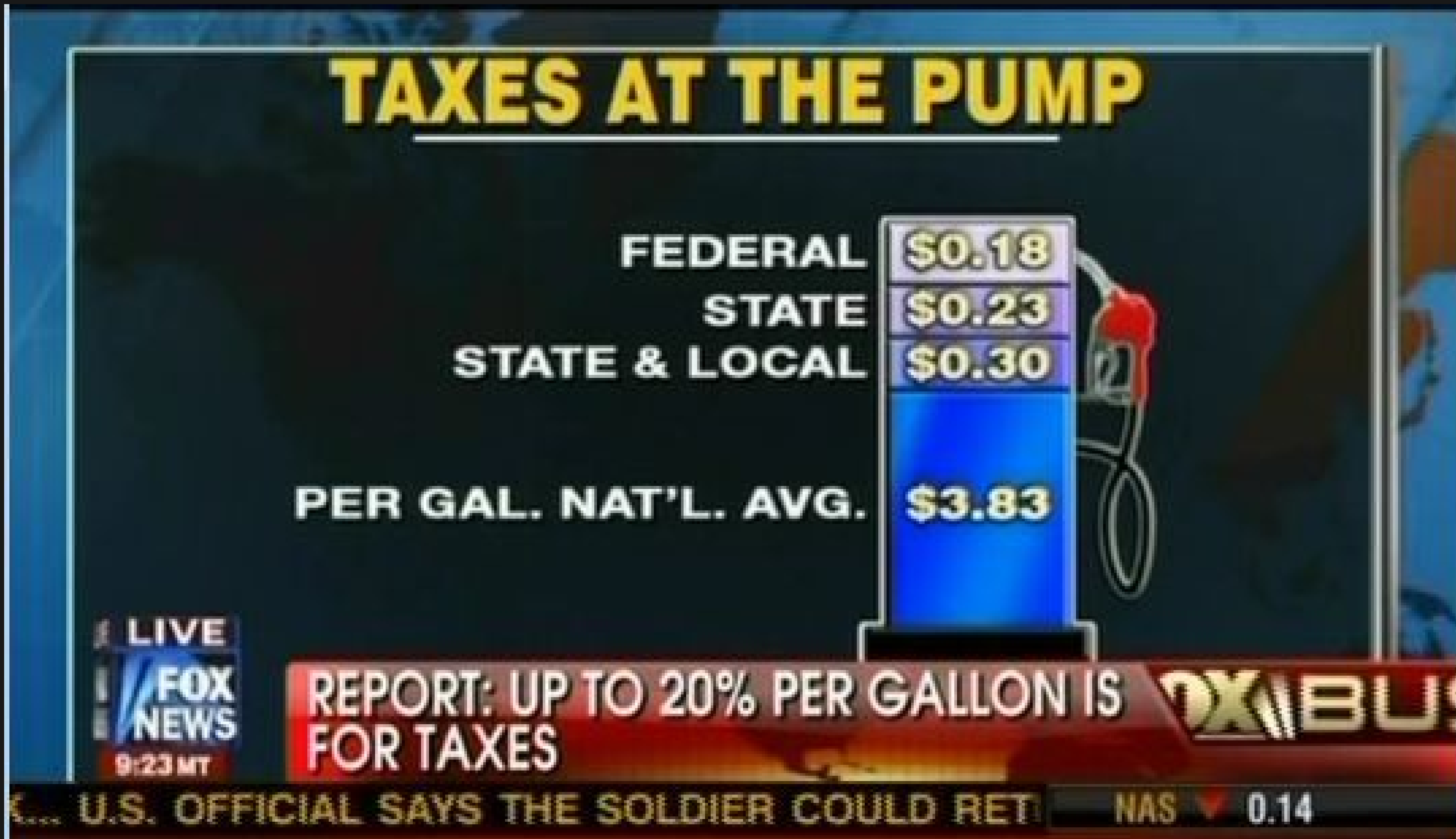
Good and bad plots

Good plots, bad plots

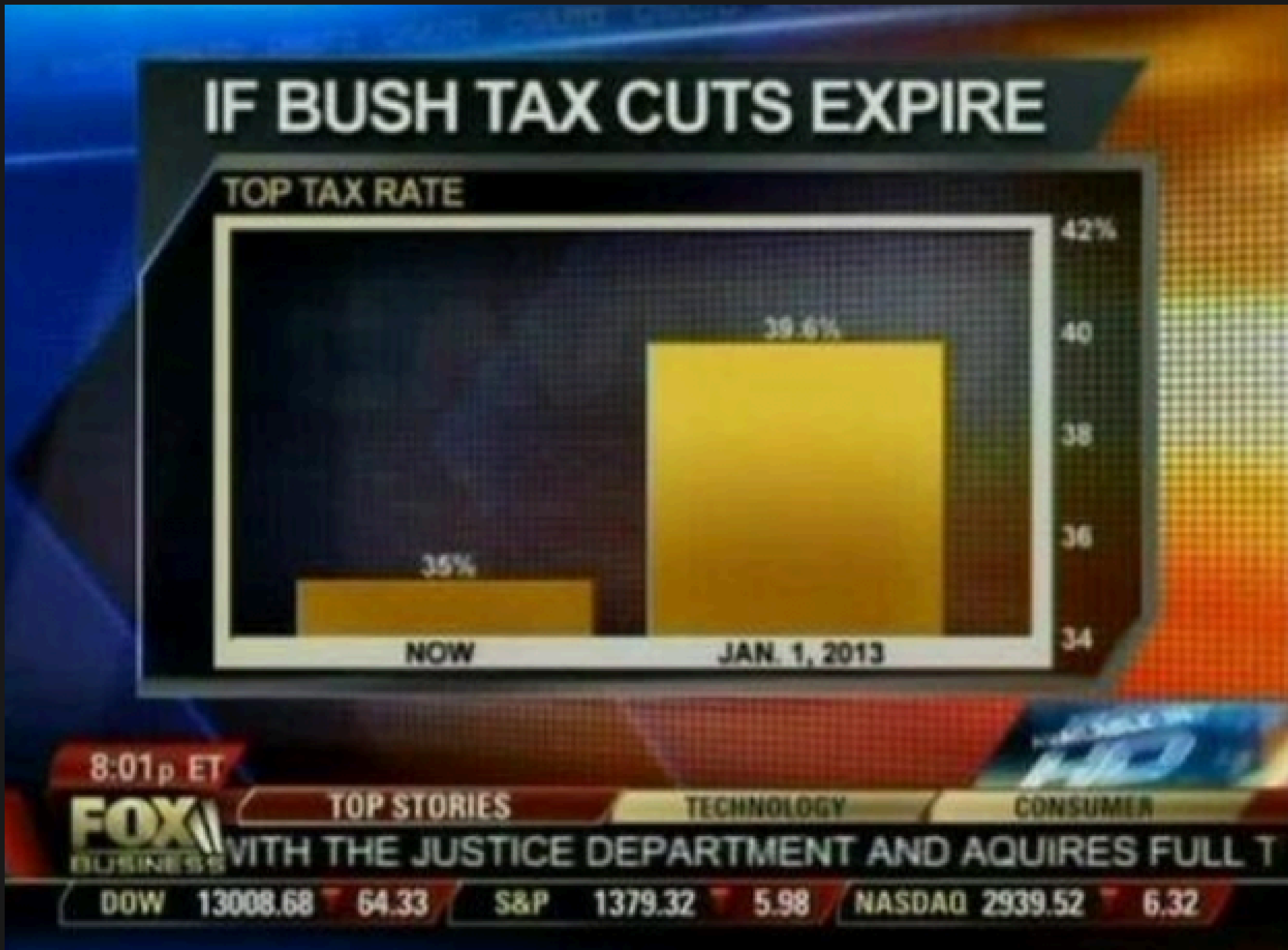
- It is important to make *`good`* plots
- i.e., plots that *look good*…
- …and are *`honest`* to the data
- it is `very easy to hide` the message rather than *highlighting* it
- it is `very easy to mislead` with a plot

> let's start with **bad plots**. *Why* are they bad?

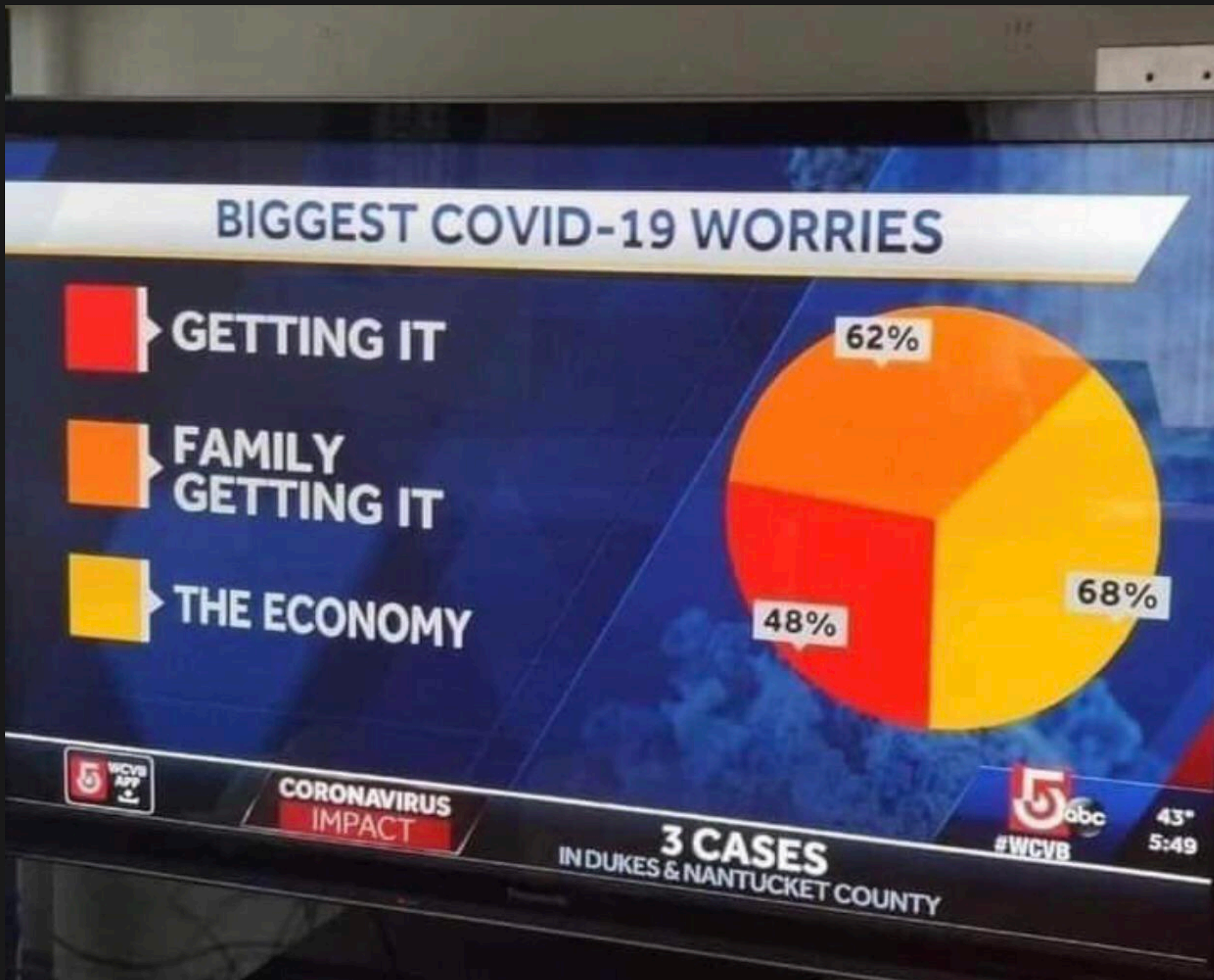
Bad plotting 1



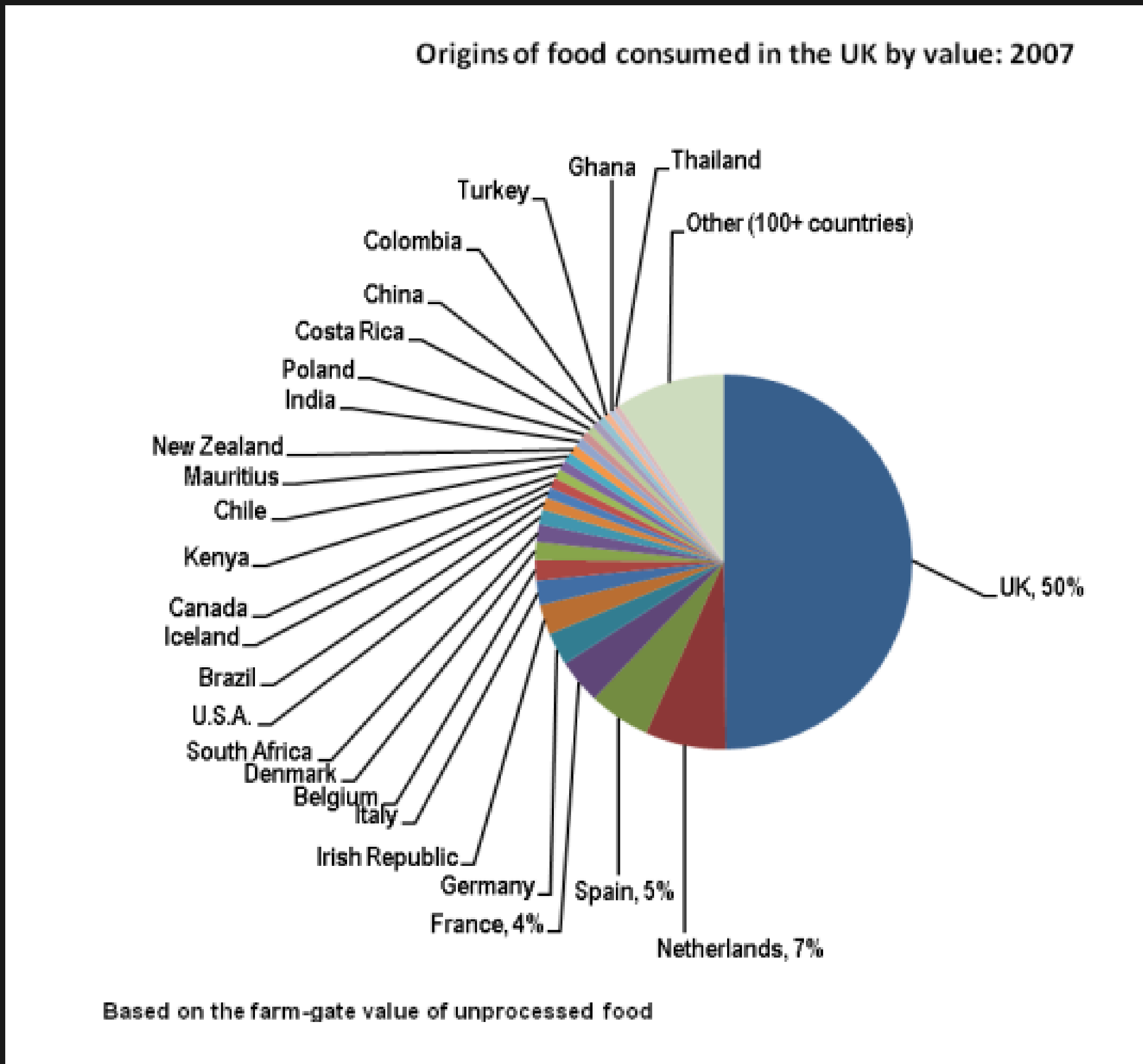
Bad plotting 2



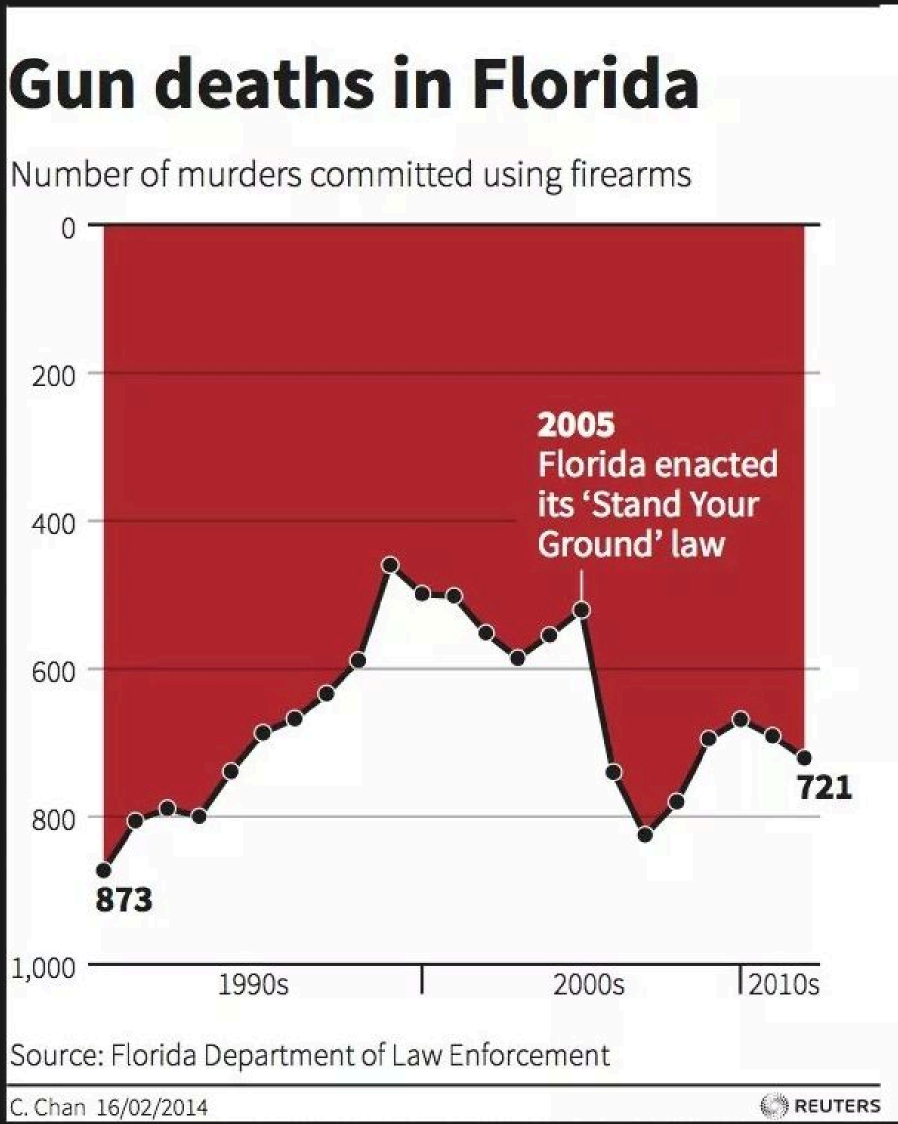
Bad plotting 3



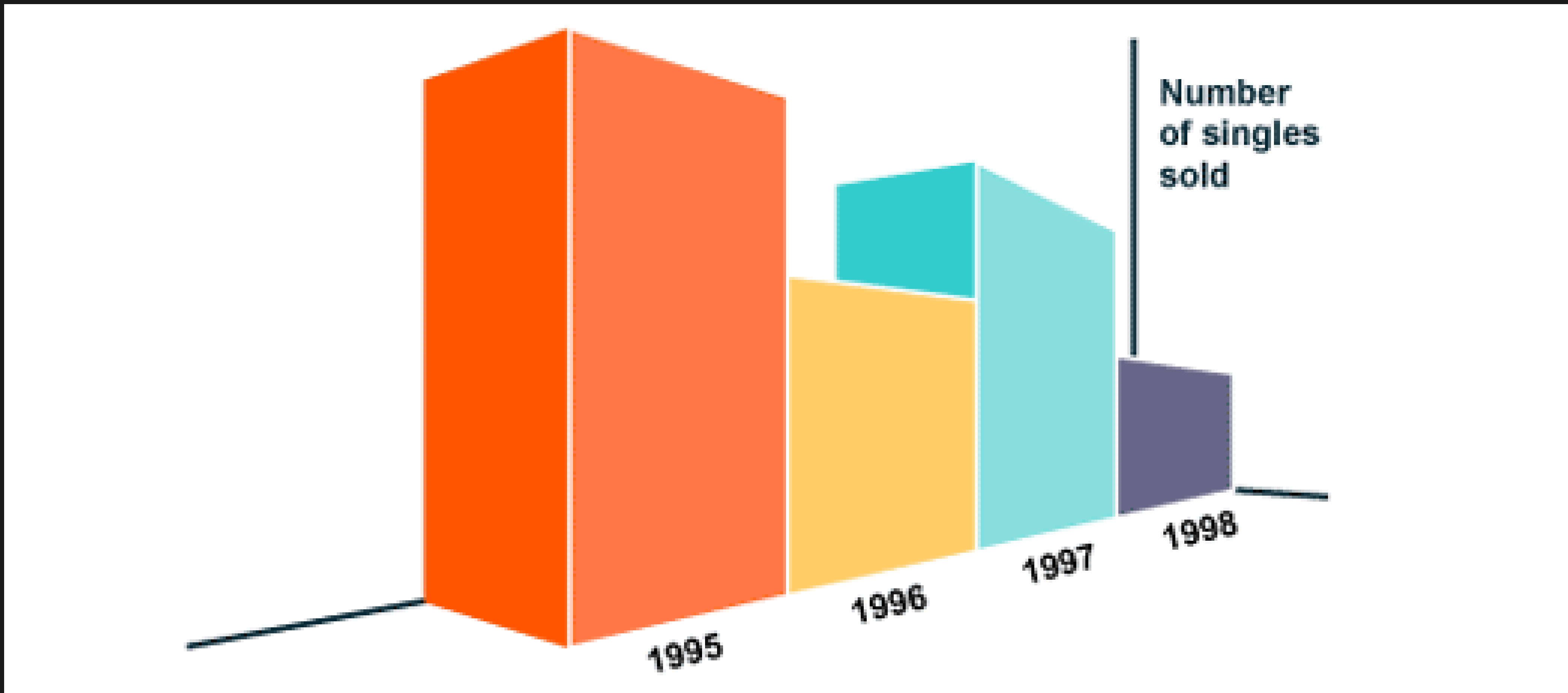
Bad plotting 4



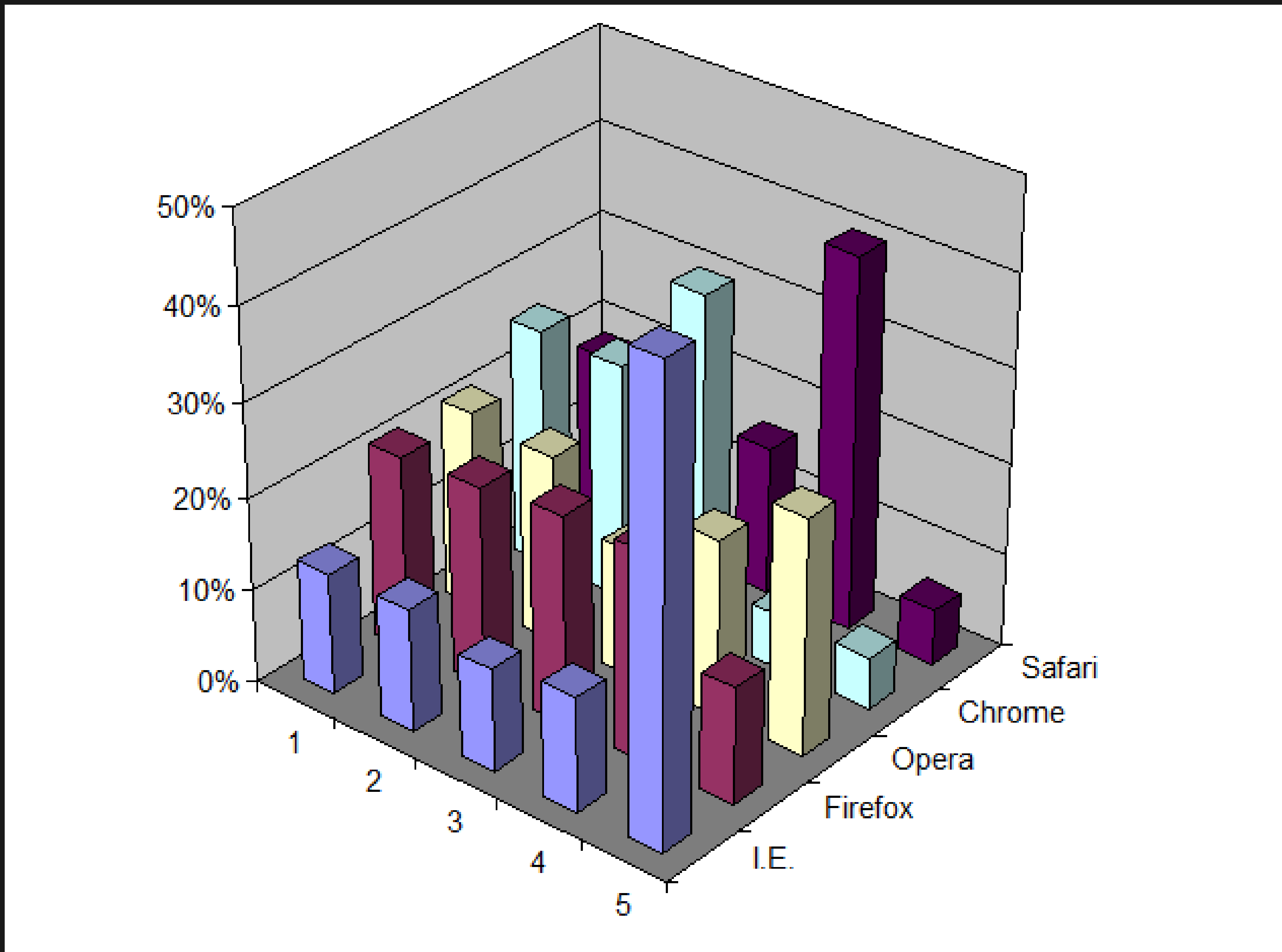
Bad plotting 5



Bad plotting 6



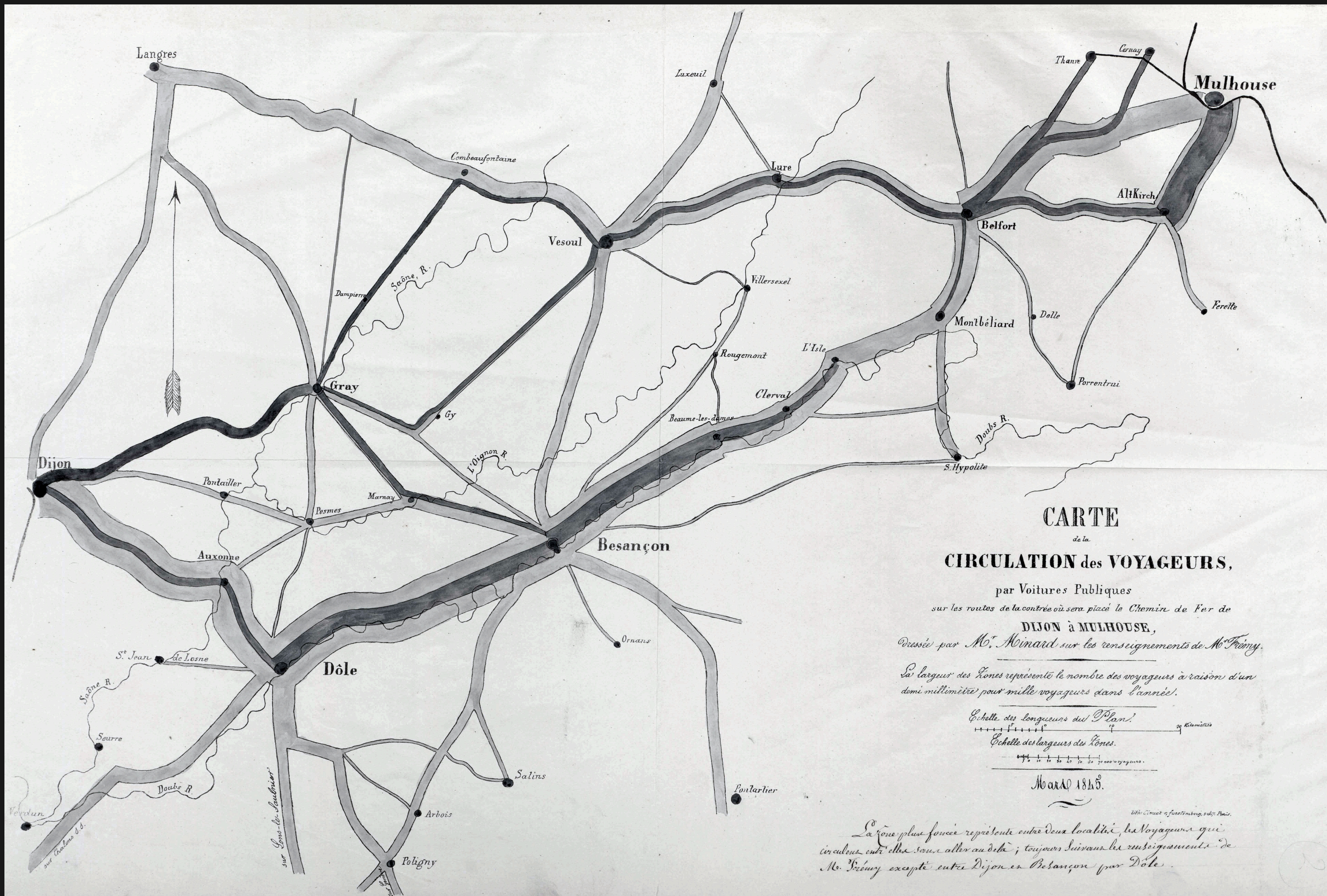
Bad plotting 7 (really, **NO** 3D plots)



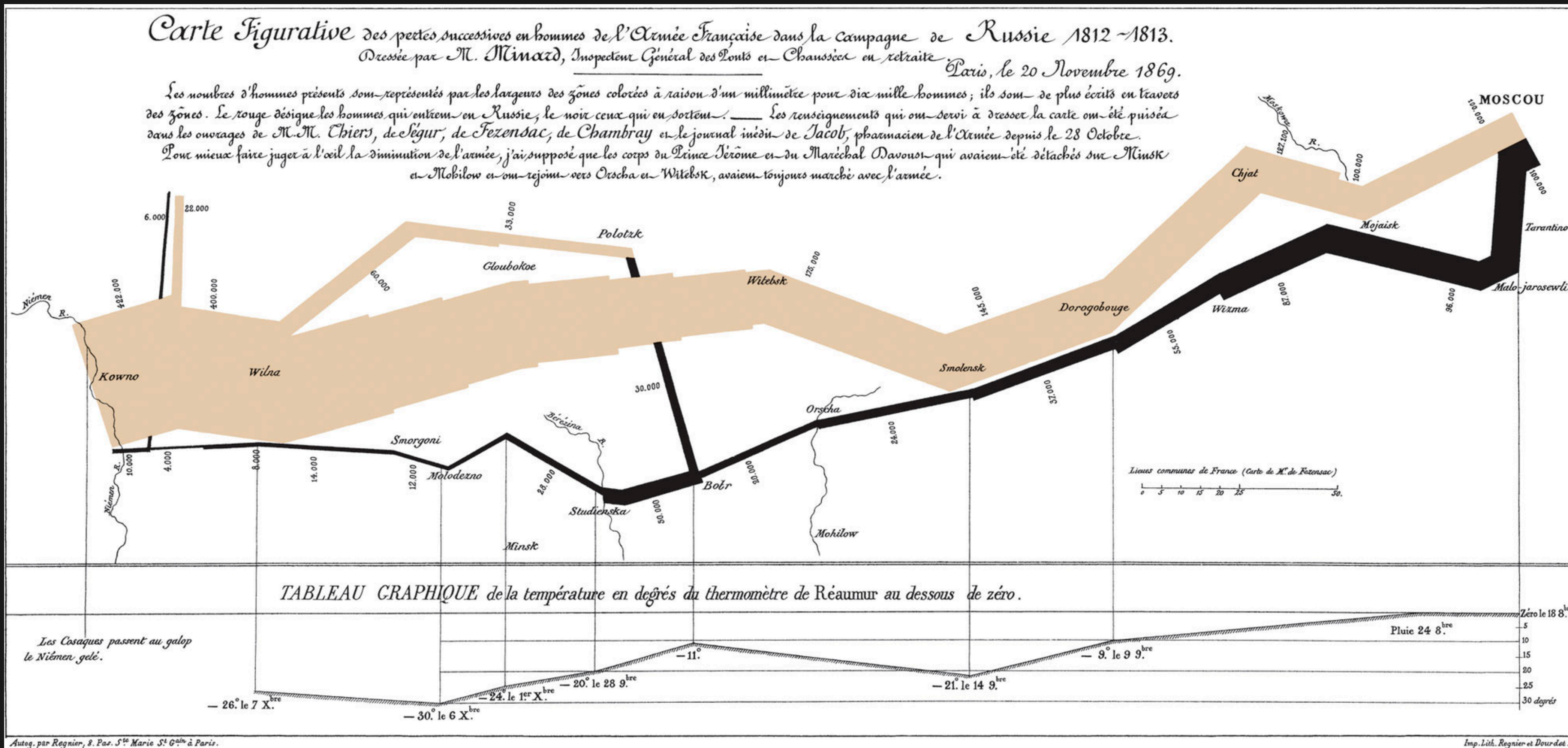
The road to good plotting

- `know` your data
- `think` before you hit the enter button
- `sketch` on paper first
- be `honest`
- draw your `axis` first
- choose your `visualization` wisely
- a good plot: lots of precise information in a concise way.

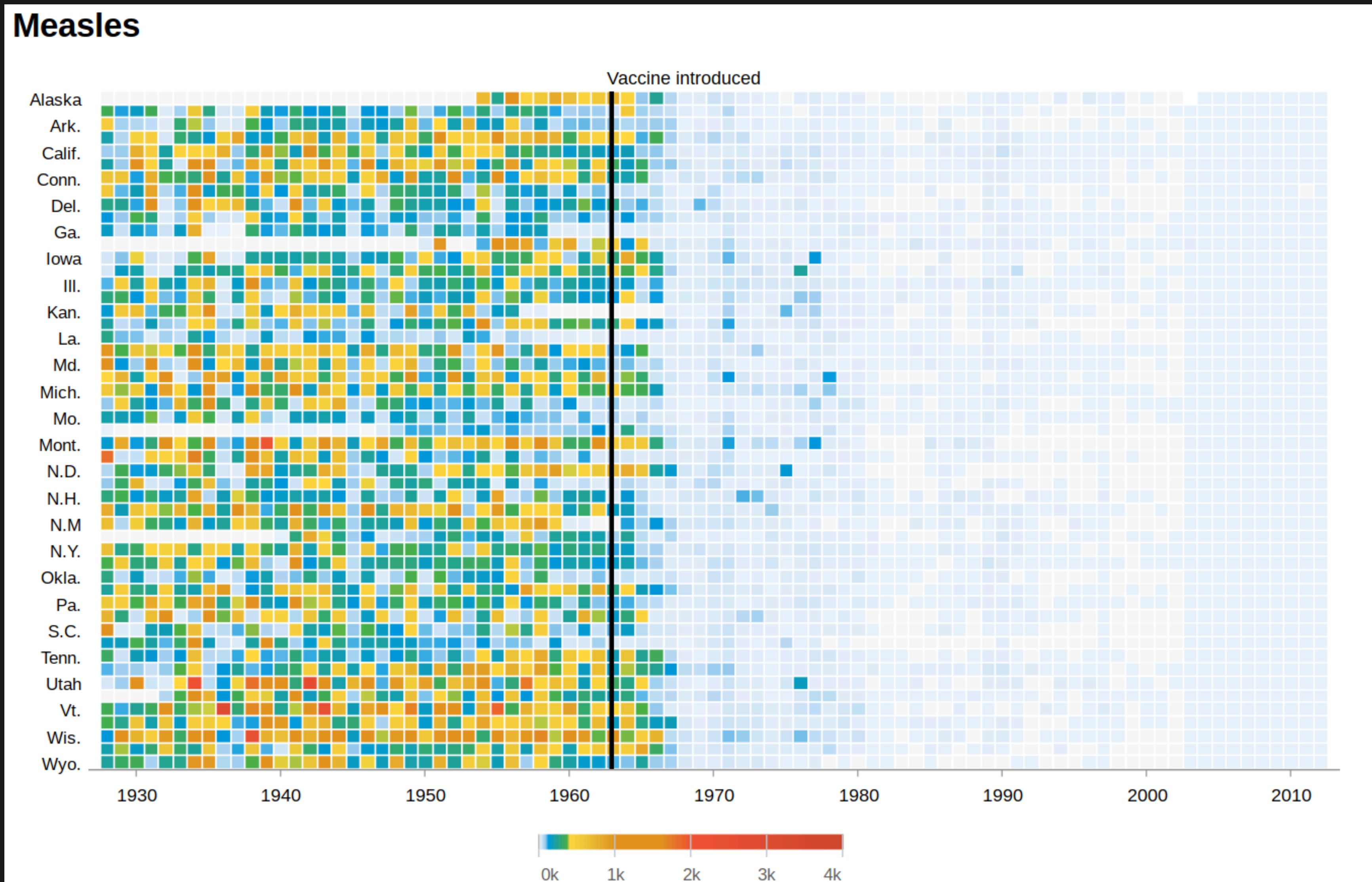
Good plots, 1


CARTE
de la
CIRCULATION des VOYAGEURS,
par Voitures Publiques
sur les routes de la contrée où sera placé le Chemin de Fer de
DIJON à MULHOUSE,
Dressée par Mr. Minard sur les renseignements de Mr Frémy.
La largeur des Zones représente le nombre des voyageurs à raison d'un demi millimètre pour mille voyageurs dans l'année.
Echelle des longueurs du Plan.
Echelle des largeurs des Zones.
Mars 1845.
La Zone plus foncée représente entre deux localités les Voyageurs qui circulent entre elles sans aller au delà ; toujours suivant les renseignements de M. Frémy excepté entre Dijon et Besançon par Dôle.
Langres
Luxeuil
Thann
Cernay
Mulhouse
Combeaufontaine
Lure
Vesoul
Belfort
AltKirch
Villersexel
Montbéliard
Dolle
Ferette
Dampierre
Saône R.
Rougemont
L'Isle
Porrentrui
Gray
Gy
Clerval
Beaume-les-dames
Doubs R.
Dijon
L'Oignon R.
S. Hypolite
Pontailler
Marnay
Pesmes
Besançon
Auxonne
Ornans
St. Jean de Losne
Dôle
Sourre
Salins
Pontarlier
Doubs R.
Arbois
Poligny

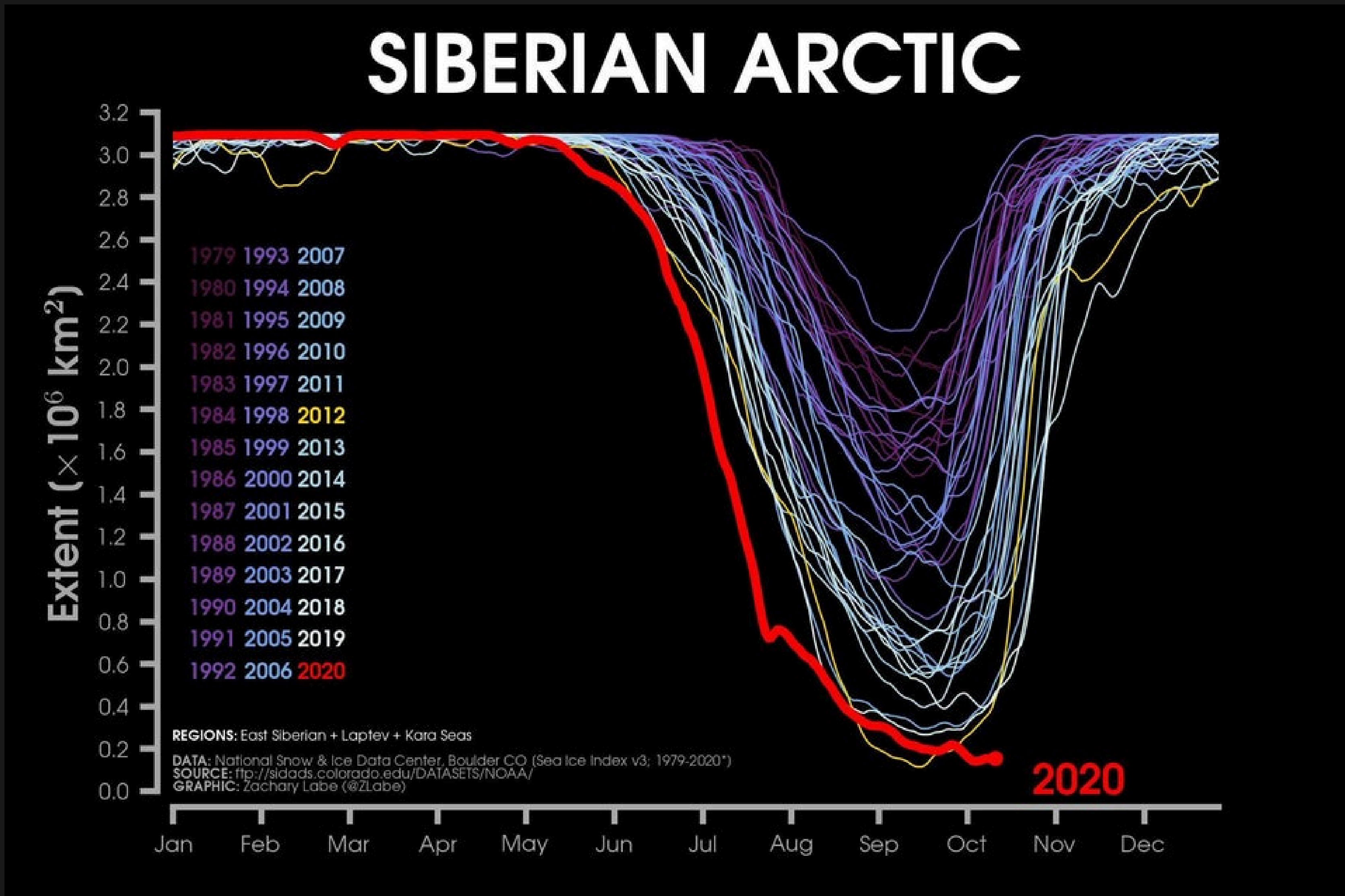
Good plots, 2


Carte Figurative des pertes successives en hommes de l'Armée Française dans la campagne de Russie 1812–1813.
Dressée par M. Minard, Inspecteur Général des Ponts et Chaussées en retraite.
Paris, le 20 Novembre 1869.
Les nombres d'hommes présents sont représentés par les largeurs des zônes colorées à raison d'un millimètre pour dix mille hommes; ils sont de plus écrits en travers des zônes. Le rouge désigne les hommes qui entrent en Russie, le noir ceux qui en sortent. —— Les renseignements qui ont servi à dresser la carte ont été puisés dans les ouvrages de M.M. Thiers, de Ségur, de Fezensac, de Chambray et le journal inédit de Jacob, pharmacien de l'Armée depuis le 28 Octobre.
Pour mieux faire juger à l'œil la diminution de l'armée, j'ai supposé que les corps du Prince Jérôme et du Maréchal Davoust qui avaient été détachés sur Minsk et Mobilow et ont rejoint vers Orscha et Witebsk, avaient toujours marché avec l'armée.
MOSCOU
Chjat
Mojaisk
Tarantino
Malo-jarosewli
Wixma
Dorogobouge
Smolensk
Orscha
Witebsk
Polotzk
Gloubokoe
Kowno
Wilna
Smorgoni
Molodezno
Studienska
Minsk
Bobr
Mohilow
Niémen R.
Bérézina R.
Moskowa R.
Lieues communes de France (Carte de M. de Fezensac)
TABLEAU GRAPHIQUE de la température en degrés du thermomètre de Réaumur au dessous de zéro.
Les Cosaques passent au galop le Niémen gelé.
Zéro le 18 8.bre
Pluie 24 8.bre
— 9.° le 9 9.bre
— 21.° le 14 9.bre
— 11.°
— 20.° le 28 9.bre
— 24.° le 1.er X.bre
— 30.° le 6 X.bre
— 26.° le 7 X.bre
30 degrés
Autog. par Regnier, 8. Pas. S.te Marie S.t G.ain à Paris.
Imp. Lith. Regnier et Dourdet.

Good plots, 3


Measles
Vaccine introduced
Alaska
Ark.
Calif.
Conn.
Del.
Ga.
Iowa
Ill.
Kan.
La.
Md.
Mich.
Mo.
Mont.
N.D.
N.H.
N.M
N.Y.
Okla.
Pa.
S.C.
Tenn.
Utah
Vt.
Wis.
Wyo.
1930
1940
1950
1960
1970
1980
1990
2000
2010
0k
1k
2k
3k
4k

Good plots, 4


SIBERIAN ARCTIC
Extent (× 10^6 km^2)
3.2
3.0
2.8
2.6
2.4
2.2
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
1979 1993 2007
1980 1994 2008
1981 1995 2009
1982 1996 2010
1983 1997 2011
1984 1998 2012
1985 1999 2013
1986 2000 2014
1987 2001 2015
1988 2002 2016
1989 2003 2017
1990 2004 2018
1991 2005 2019
1992 2006 2020
REGIONS: East Siberian + Laptev + Kara Seas
DATA: National Snow & Ice Data Center, Boulder CO (Sea Ice Index v3; 1979-2020*)
SOURCE: ftp://sidads.colorado.edu/DATASETS/NOAA/
GRAPHIC: Zachary Labe (@ZLabe)
2020
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec

`ggplot2`: the basics

Some data

We will start by using the *built-in dataset* **mpg**

```
1  mpg
```

```
# 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… f        18    29 p     
comp…
 2 audi         a4           1.8  1999     4 manu… f        21    29 p     
comp…
 3 audi         a4           2    2008     4 manu… f        20    31 p     
comp…
 4 audi         a4           2    2008     4 auto… f        21    30 p     
comp…
 5 audi         a4           2.8  1999     6 auto… f        16    26 p     
```

A look at the data

A look at the data

```
skimr::skim(mpg)
```

Data summary

Name	mpg
Number of rows	234
Number of columns	11
_______________________	
Column type frequency:	
character	6
numeric	5

____

Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	e
manufacturer	0	1	4	10	
model	0	1	2	22	
trans	0	1	8	10	
drv	0	1	1	1	
fl	0	1	1	1	
class	0	1	3	10	

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd
displ	0	1	3.47	1.29
year	0	1	2003.50	4.51
cyl	0	1	5.89	1.61
cty	0	1	16.86	4.26
hwy	0	1	23.44	5.95

Why `ggplot2`?

Advantages of `ggplot2`

- consistent underlying `grammar of graphics`
- plot specification at a high level of `abstraction`
- very flexible
- mature and complete graphics system
- `theme` system for polishing plot appearance
- many users, active, fast & competent `support`
- arguably the `best` plotting system on the planet

Grammar of graphics

> Independently specify plot `building blocks` & combine them to create *any* plot.

Starting from the basics

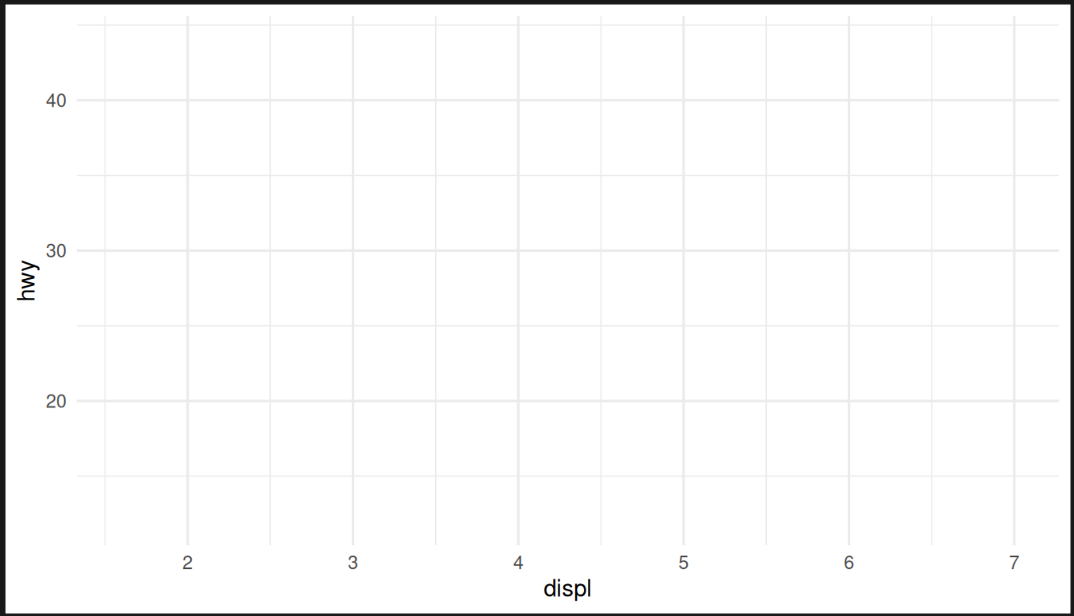
As in a grammar the minimal sentence is a subject in a plot the minimal object is data

```r
p <- ggplot(mpg)
```

basics

In a grammar, you need a verb. In plots, this is axis

```
p <- p + aes(x = displ, y =hwy)
p
```

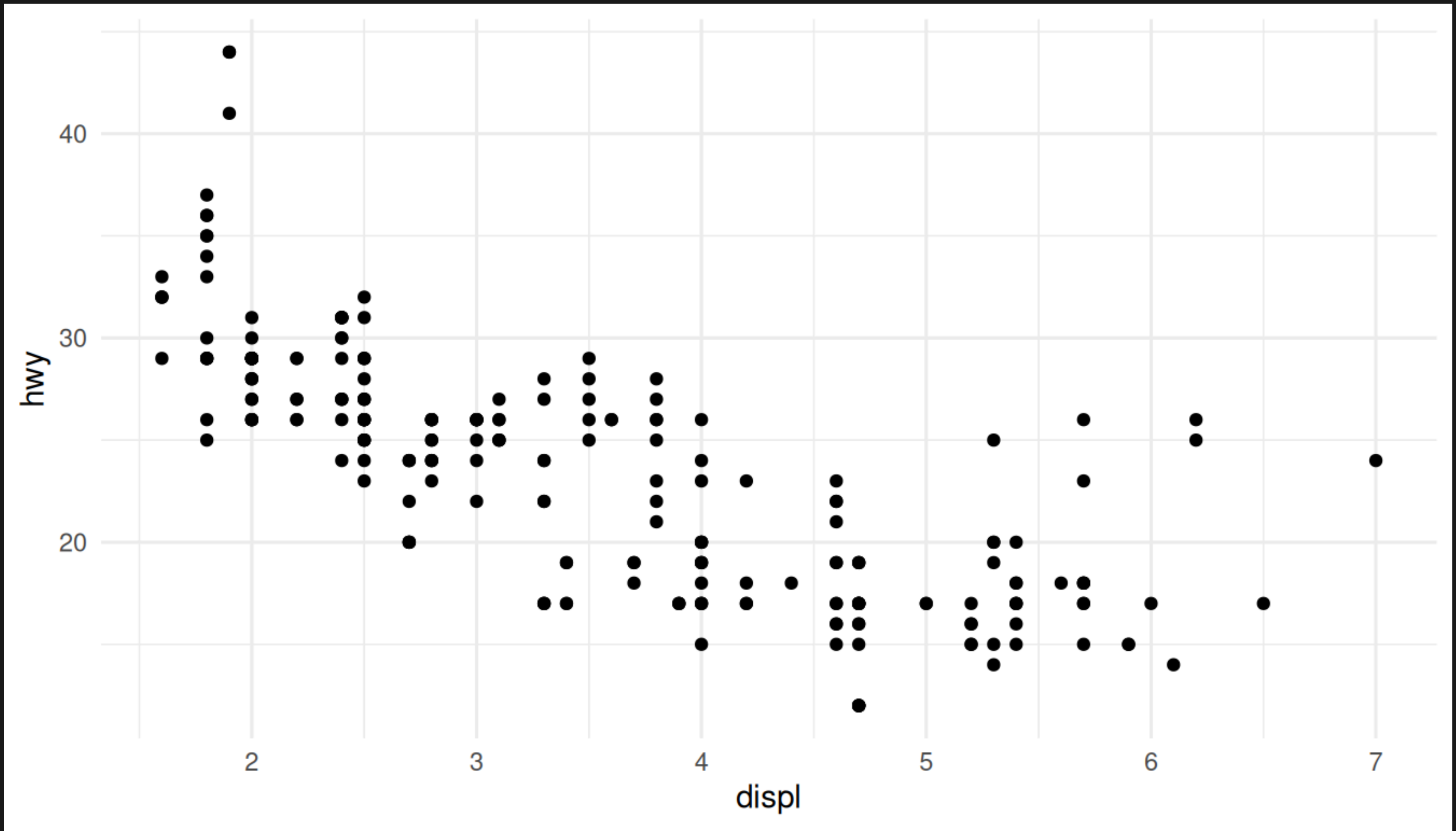


> Still no real plot generated!

Generating a plot

But you also need an object. In ggplot, this is *geoms*

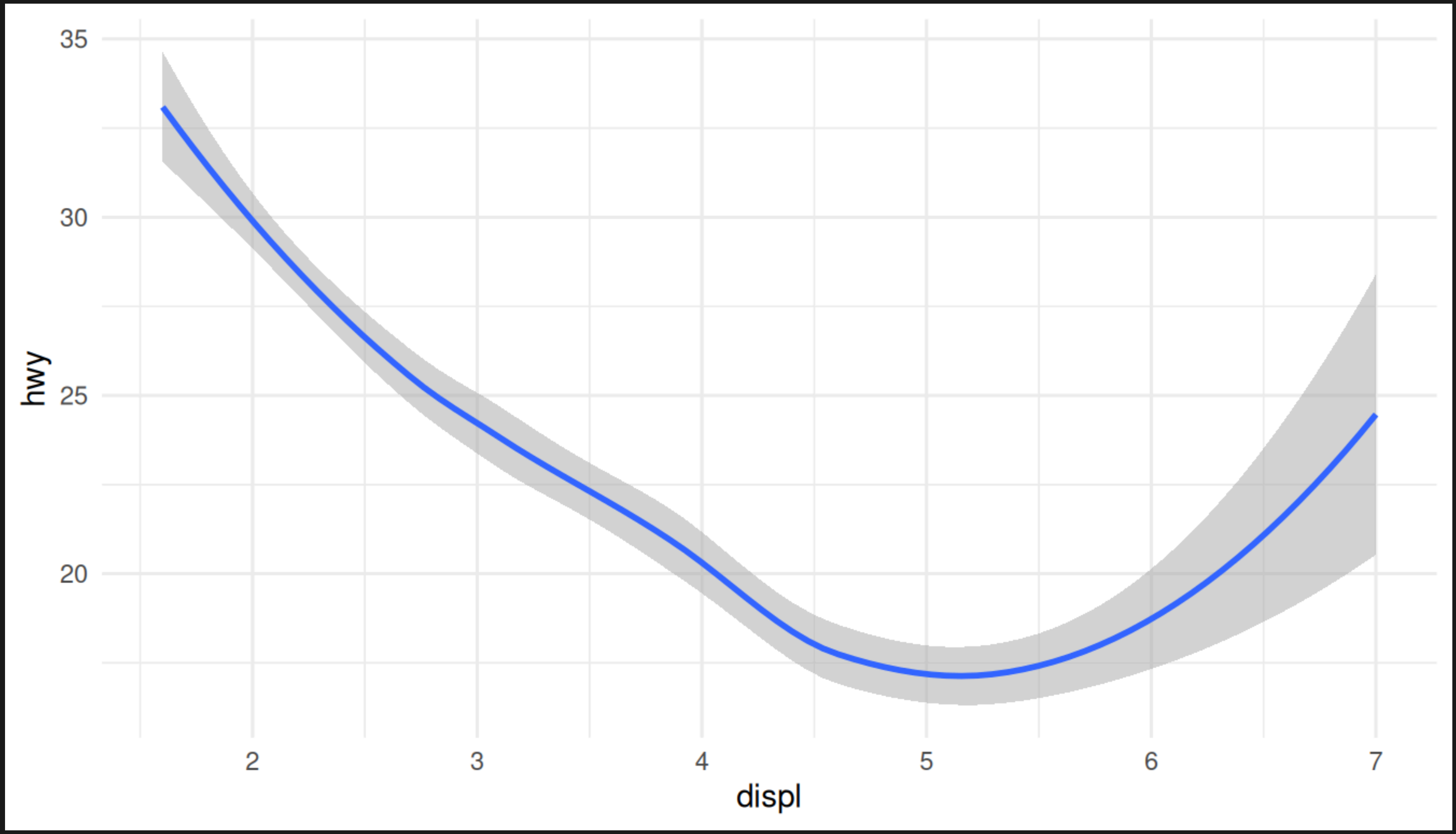
```
p + geom_point()
```



Generating a plot, 2

But you also need an object. In ggplot, this is *geoms*

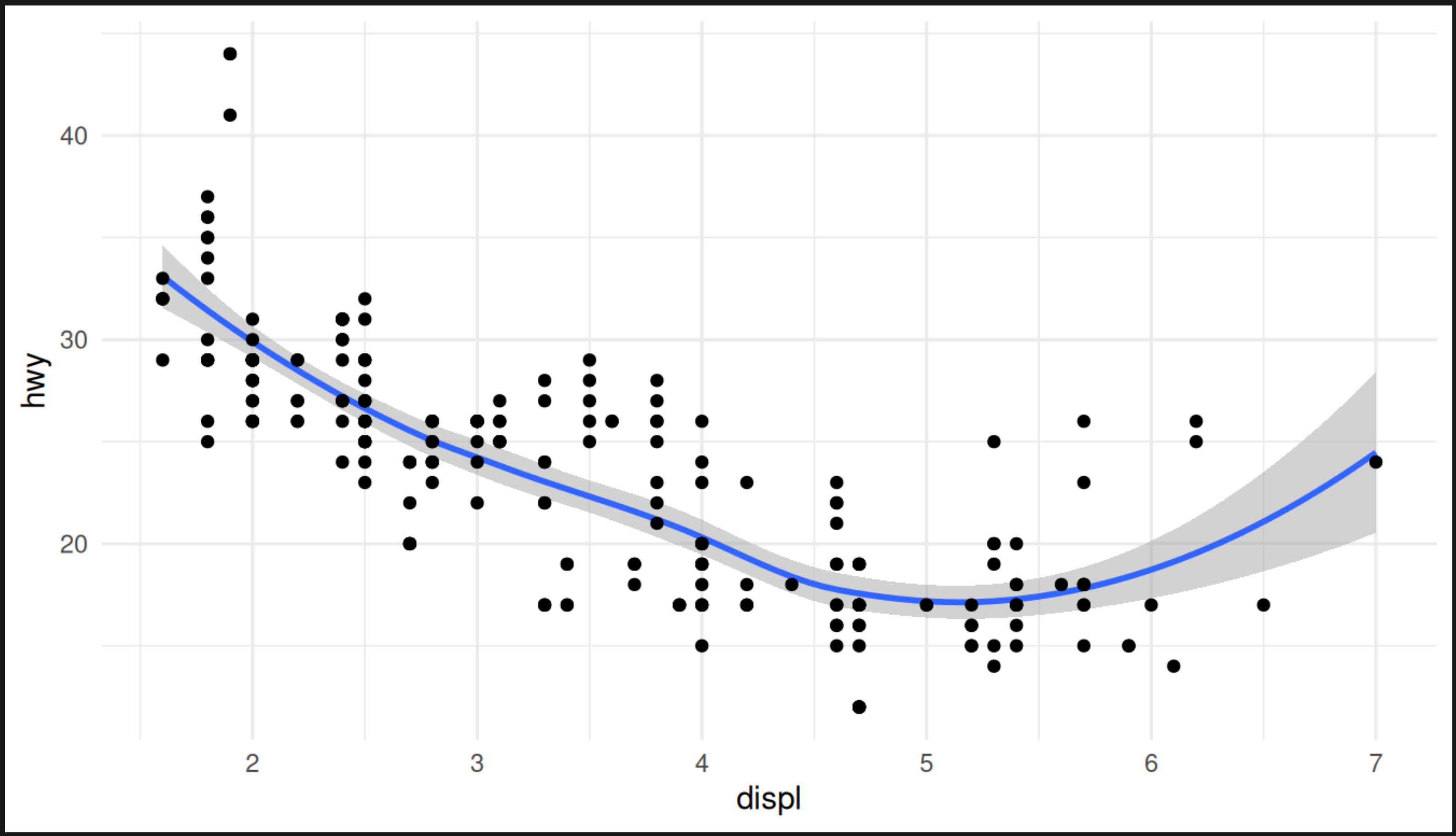
```
p + geom_smooth()
```



Generating a plot, 3

You can add (+) as many *geoms* as you wish

```
p + geom_smooth()+geom_point()
```

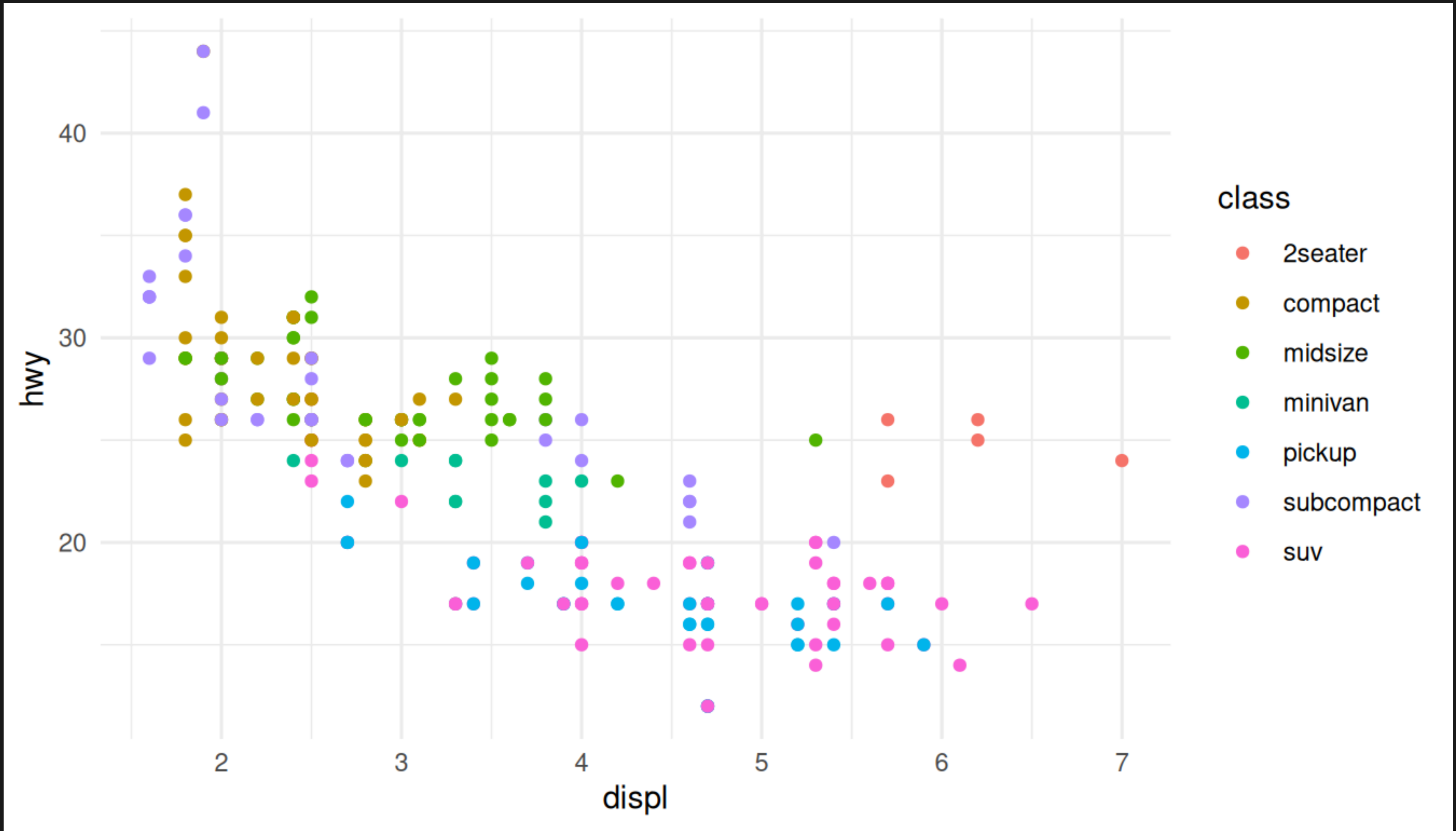


The beauty of a grammar `metaphor`

- once you get the main idea, `adding` things is easy
- a plot is a **sentence made with data**
- you add layers with `+`
- as you would add words to a sentence
- as in grammar you use adjectives to give more nuanced meaning, in plots you could use `+` to add color, fill, size, shape, etc…

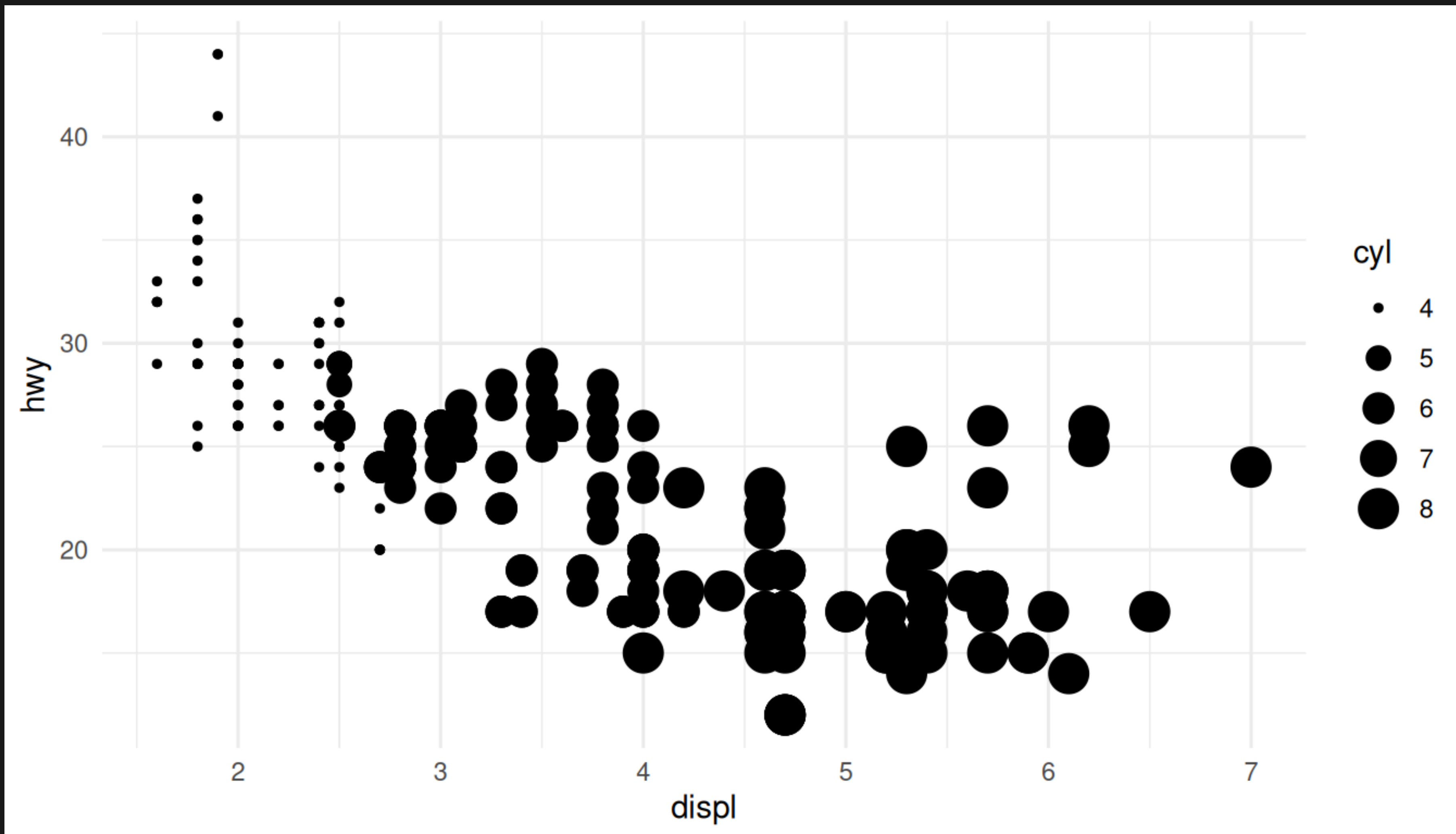
Adding meaning: color

```
p + geom_point(aes(color=class))
```



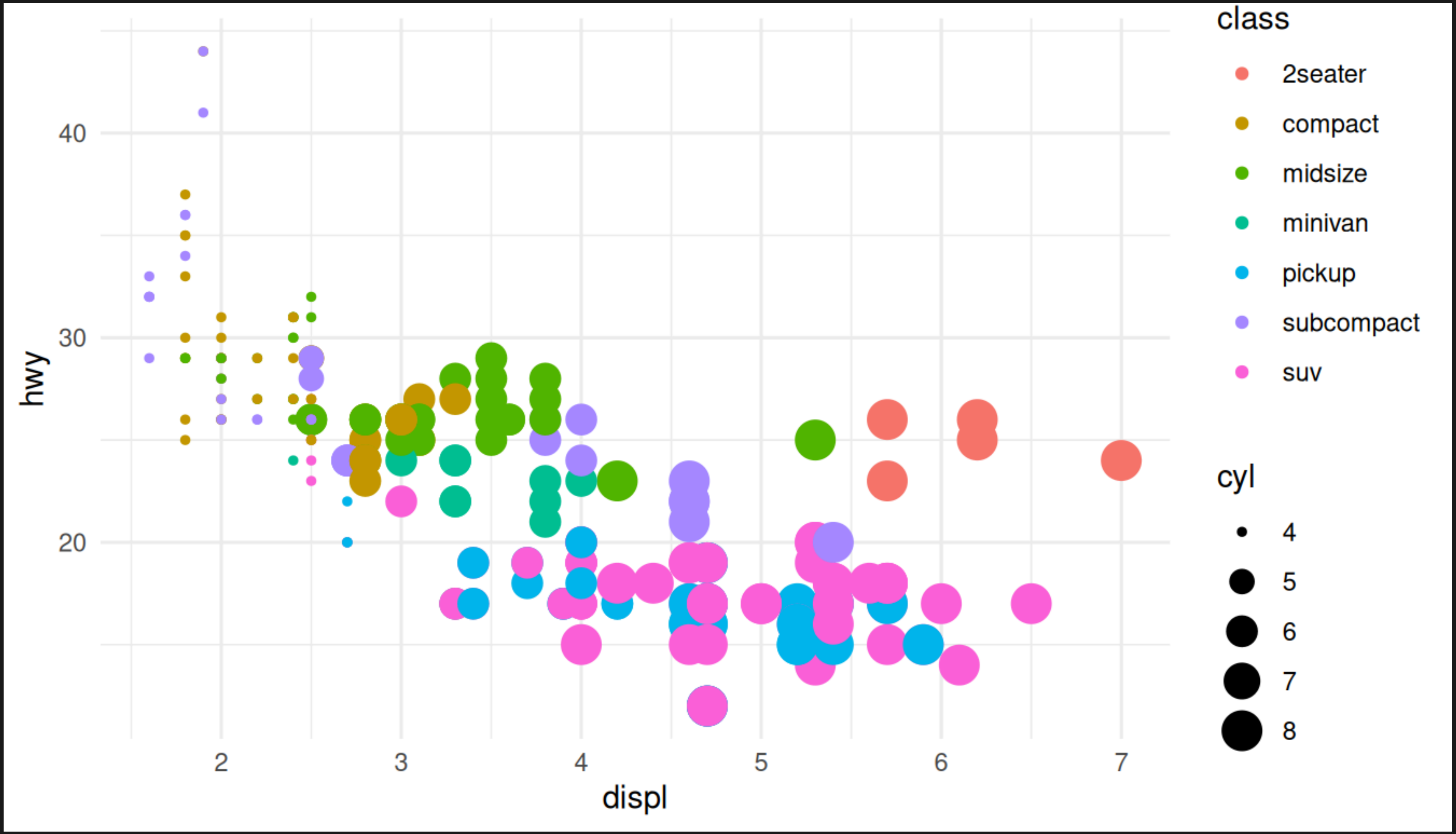
Adding meaning: size

```
p + geom_point(aes(size=cyl))
```



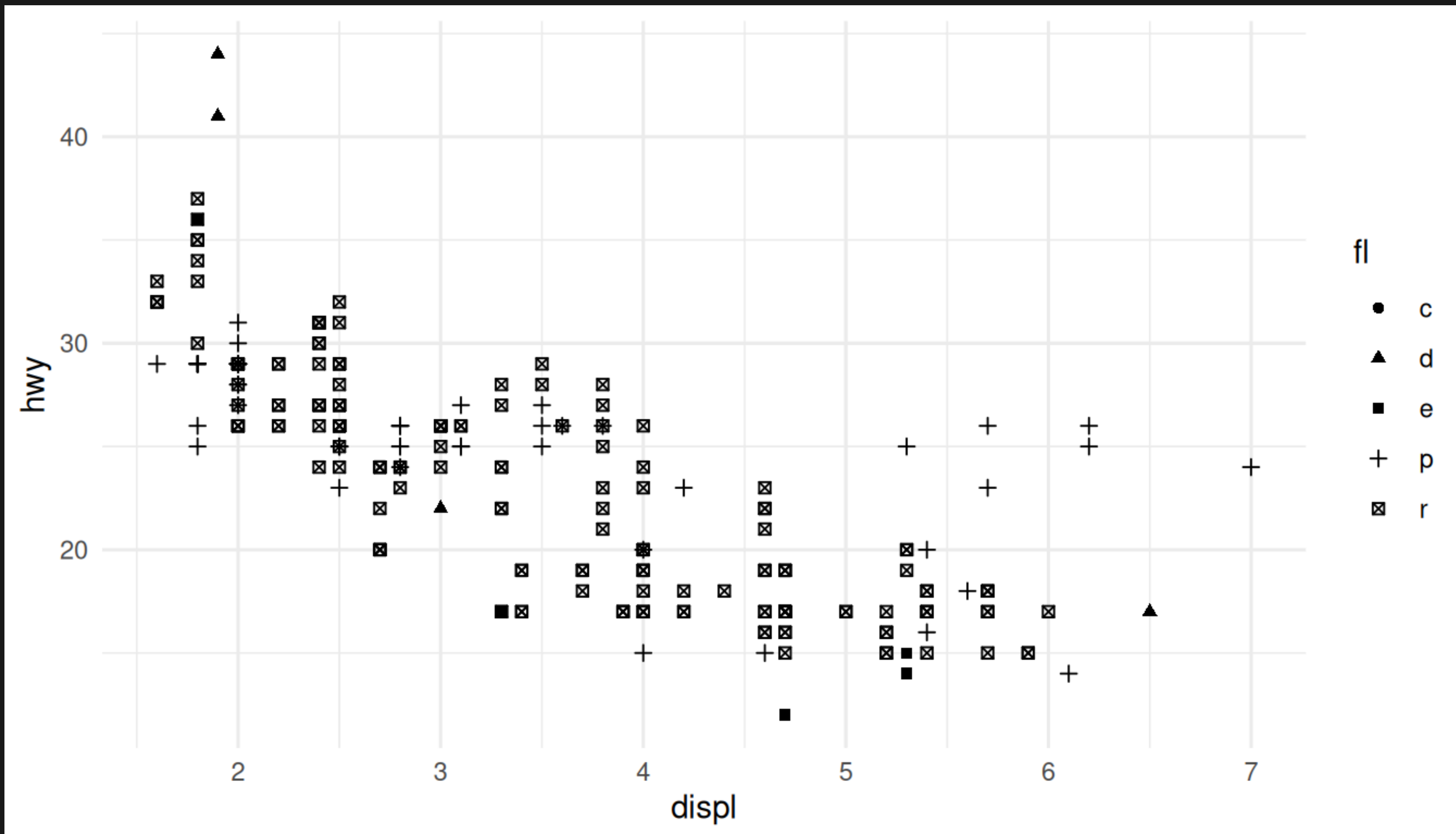
Adding meaning: color AND size

```
1 p + geom_point(aes(size = cyl, color=class))
```



Adding meaning: shape

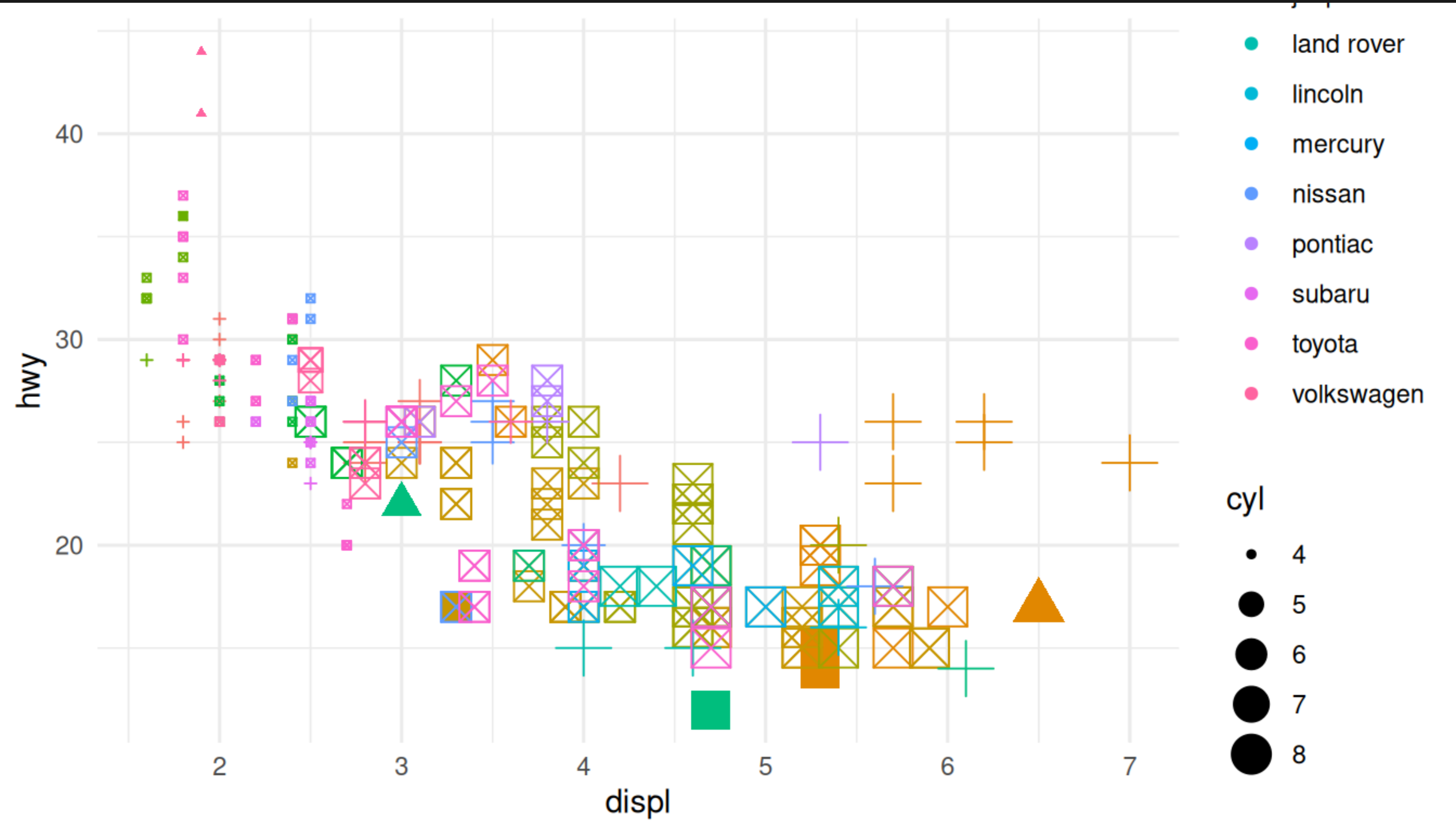
```
p + geom_point(aes(shape=fl))
```



Adding meaning: all together (…)

Possibly not a good idea though

```
p + geom_point(aes(color=manufacturer, shape =fl, size = cyl))
```



Recap so far

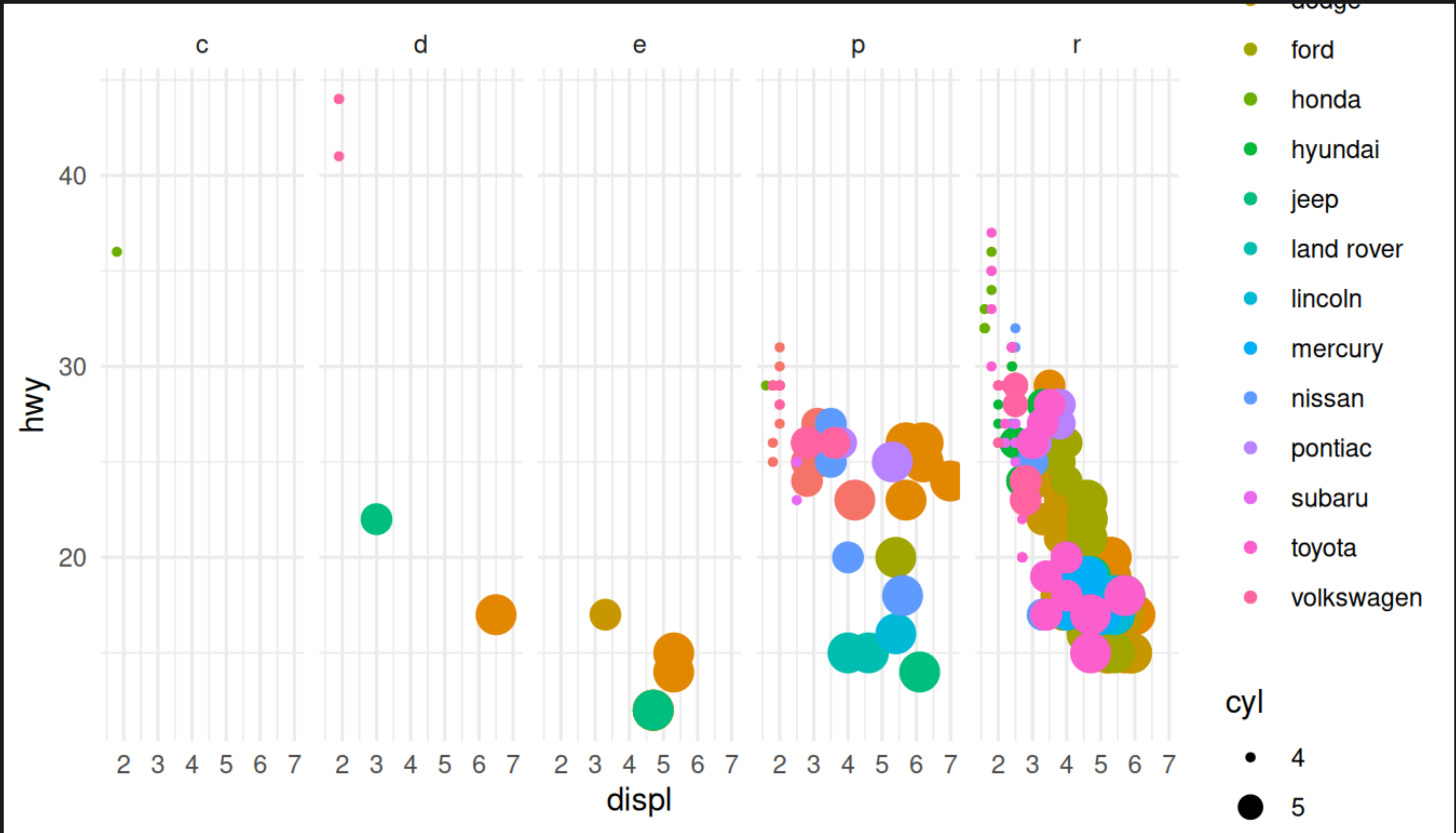
- ggplot works like a grammar
- start with `ggplot()`
- first argument: *data*: `ggplot(df, ...)`
- then map variables to `aes`thetics (x, y, color, fill, …)
- `ggplot(df, aes(dimension = variable))`
- then add meaning with geometric objects: `geom_*`
- notes:
  - `geom`s inherit the `aes` of the plot if not specified
  - all variables mapped to `aes` vary with the data

Facets

- sometimes sentences become too long
- it is useful to `split` them up in shorter sentences
- you could first talk about a car, *then* another one
- in plots, you can split up the plot along a `variable`
- one plot is drawn `for each level` of a given variable

Facets

```r
p + geom_point(aes(color=manufacturer, size = cyl))+facet_grid(.~fl)
```



More details on the grammar

> Once your main plot is done, you can tweak it

- coordinate functions (changing the axis)
- scale functions (changing how geoms look)
- theme functions (changing how the plot looks)

We will do this in Lecture 5 – advanced plotting

`ggplot2`: gallery

Exploring data: one variable

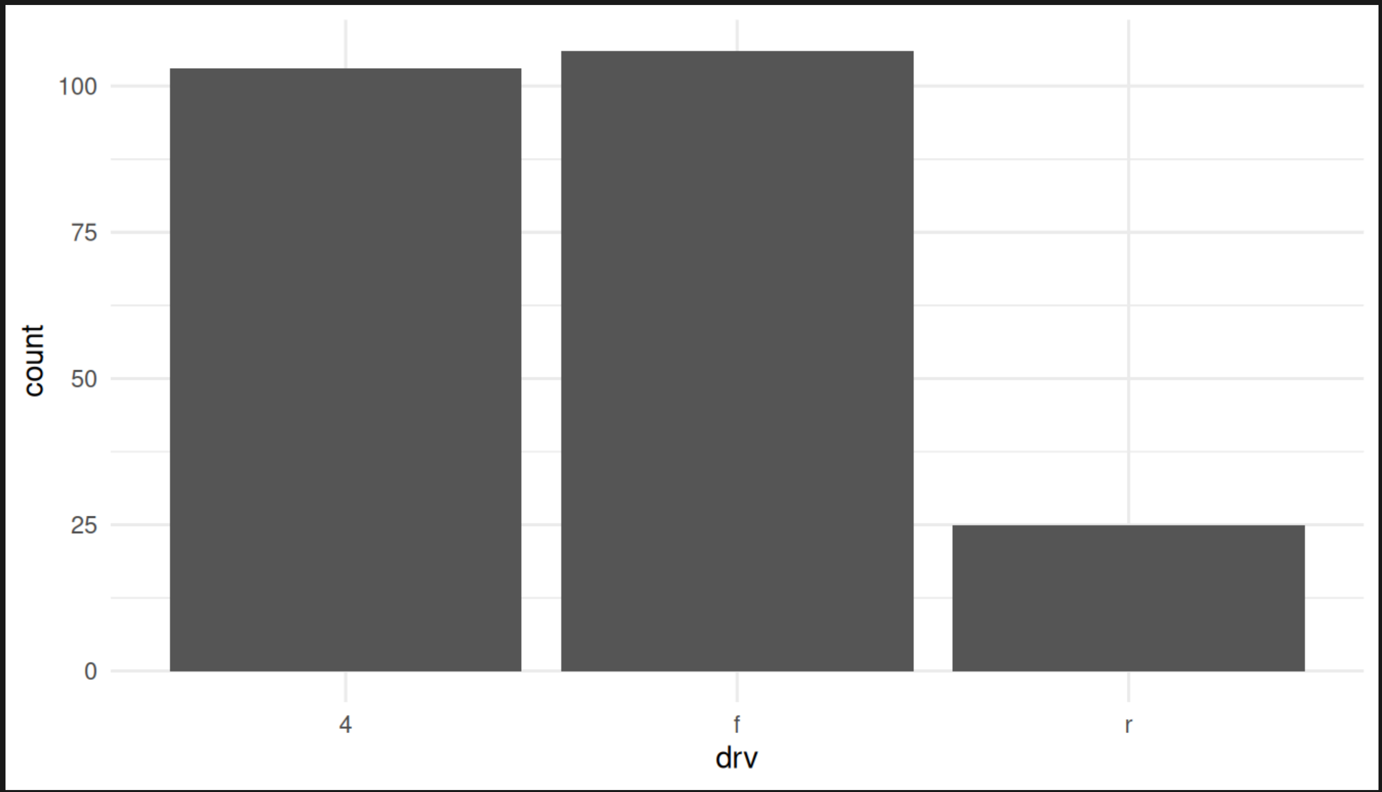
Plot types depend on the variable type

- *one-variable plots, discrete variable*: `barplot`
- *one-variable plots, continuous variable*: `distribution`, `density`

Barplots

- let's look at the drive type of the cars: front, rear, or 4wd

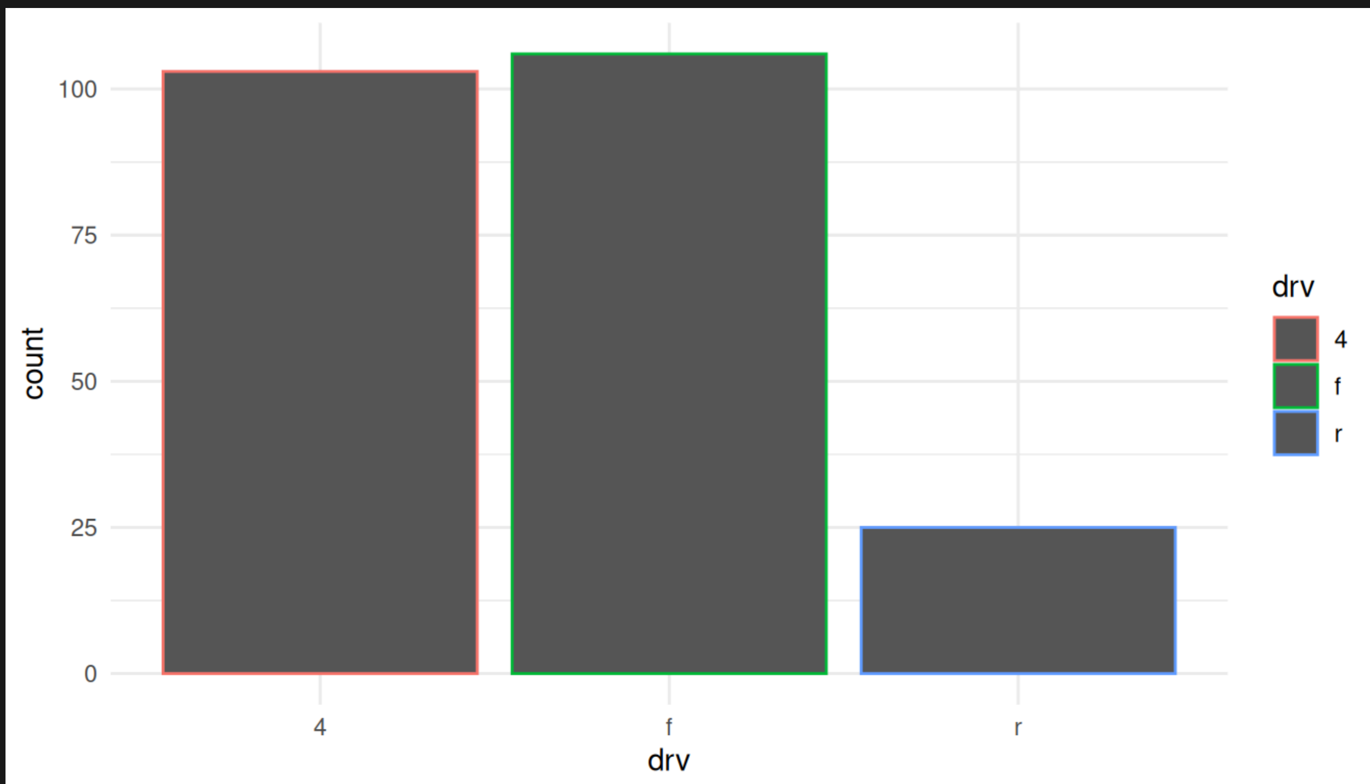
```r
p <- ggplot(mpg, aes(drv))
p + geom_bar()
```



Barplots

- not so fancy. should we add color?

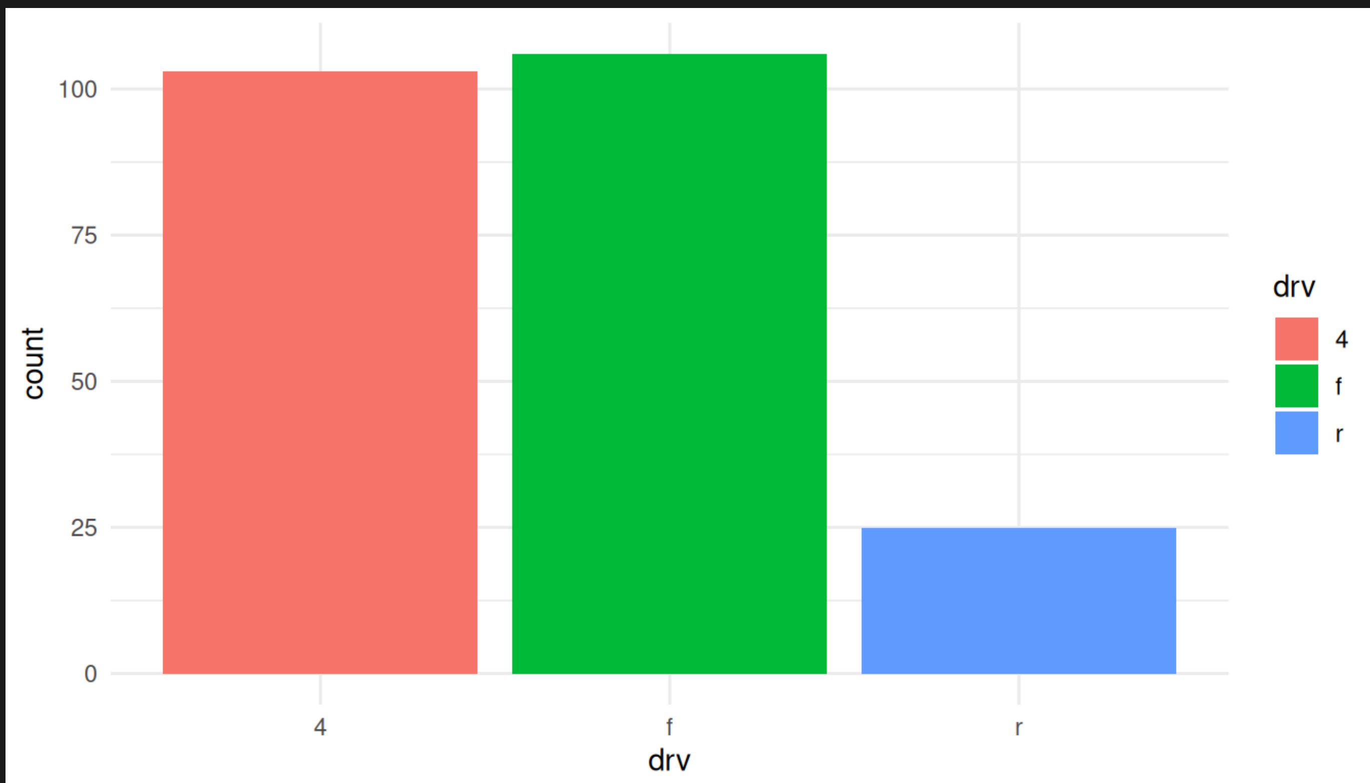
```
p <- ggplot(mpg, aes(drv))
p + geom_bar(aes(color=drv))
```



Barplots

- ups. Maybe we meant fill?

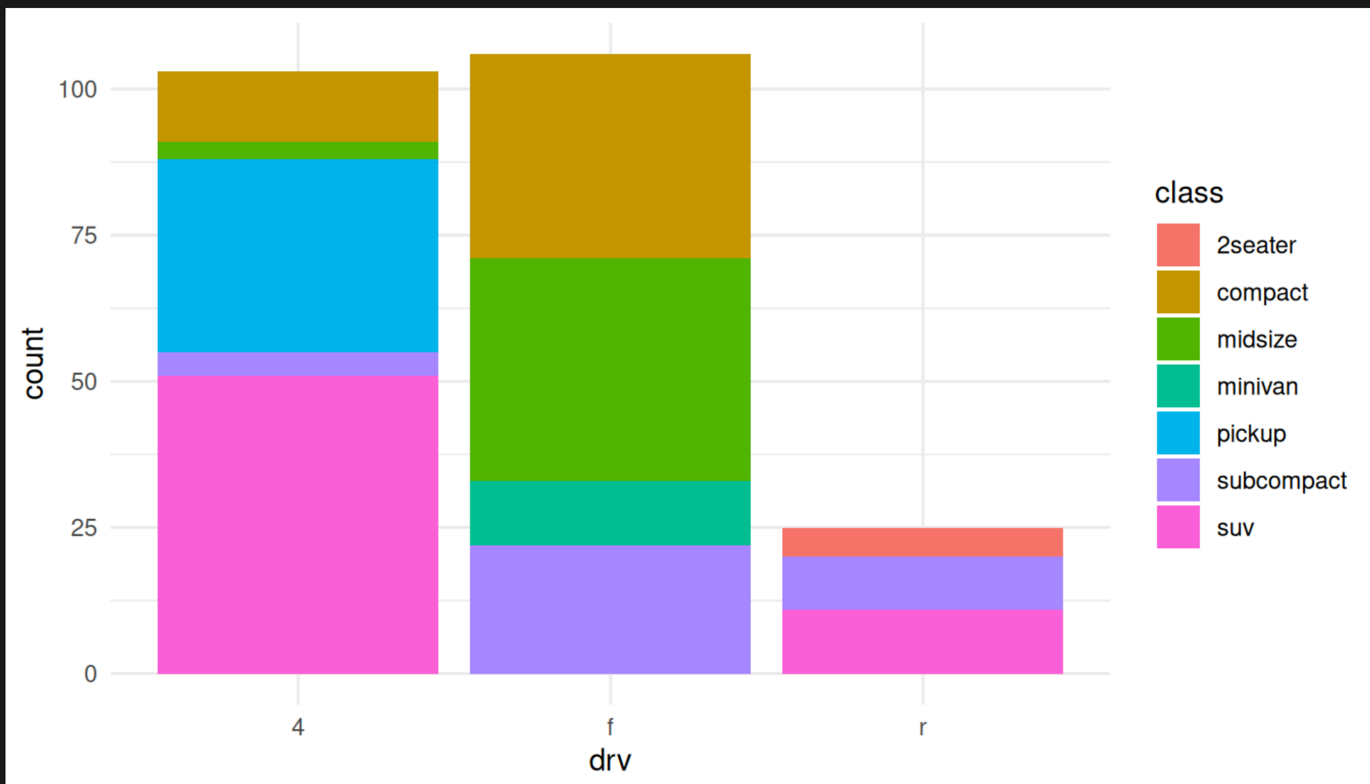
```
p <- ggplot(mpg, aes(drv))
p + geom_bar(aes(fill=drv))
```



Barplots

- what if we cross it with another variable?

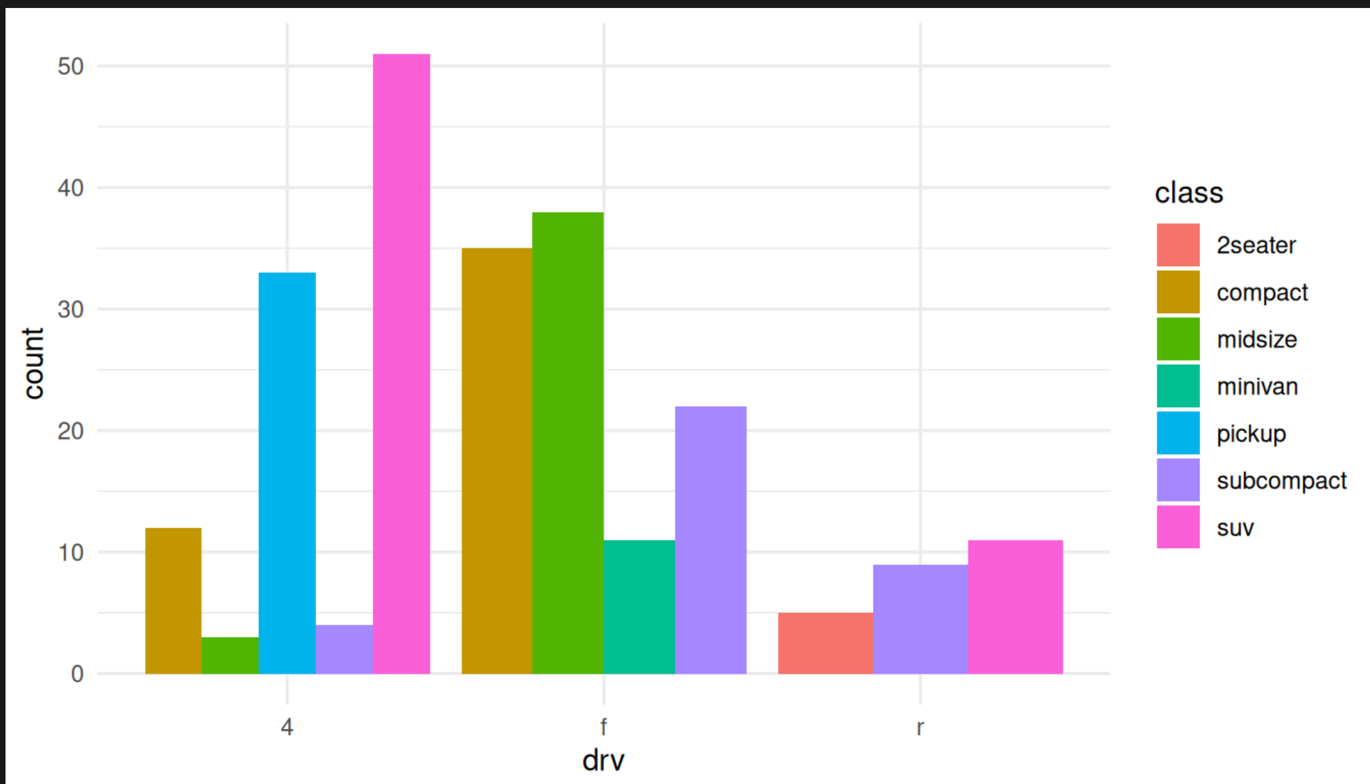
```
p <- ggplot(mpg, aes(drv))
p + geom_bar(aes(fill=class))
```



Barplots

- By default stacked. How to unstack?

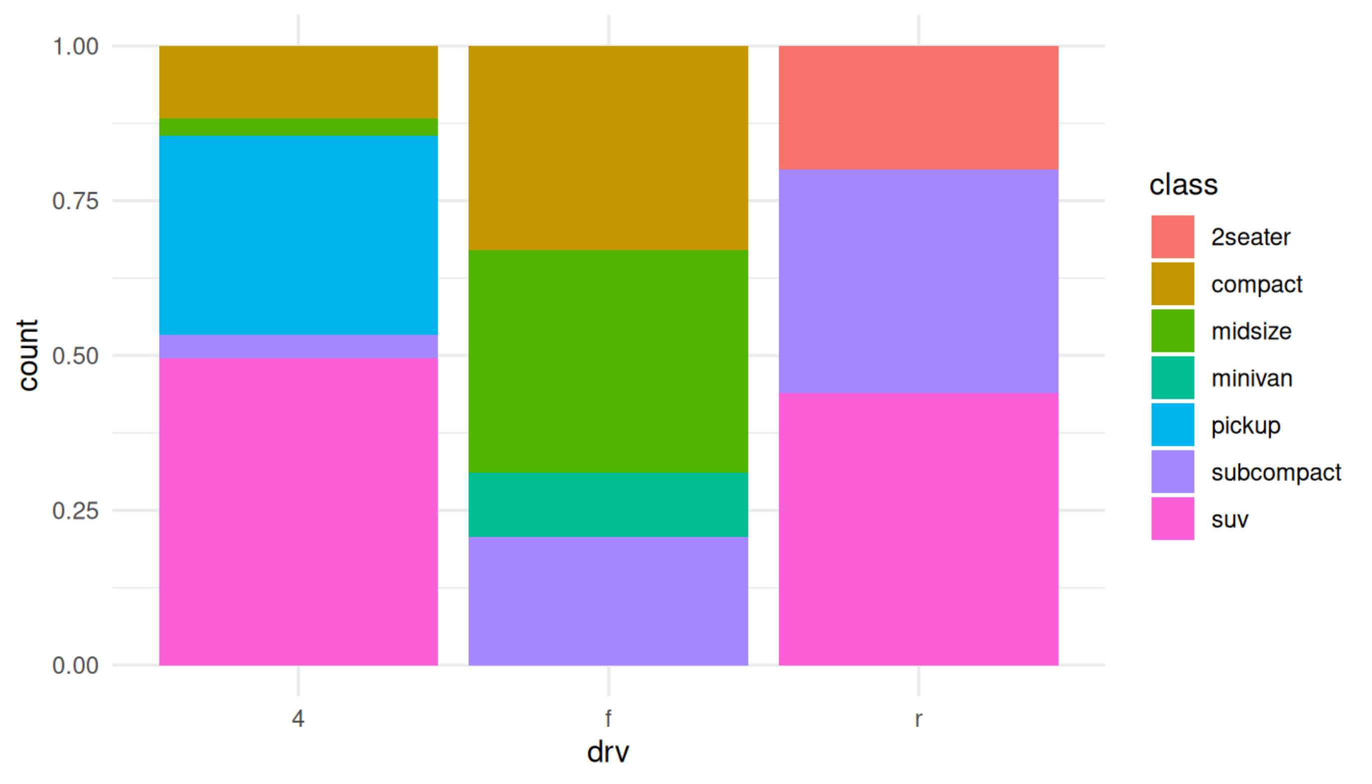
```
p <- ggplot(mpg, aes(drv))
p + geom_bar(aes(fill=class), position = position_dodge())
```



Barplots

- By default stacked. How to show relative weight?

```
p <- ggplot(mpg, aes(drv))
p + geom_bar(aes(fill=class), position = position_fill())
```



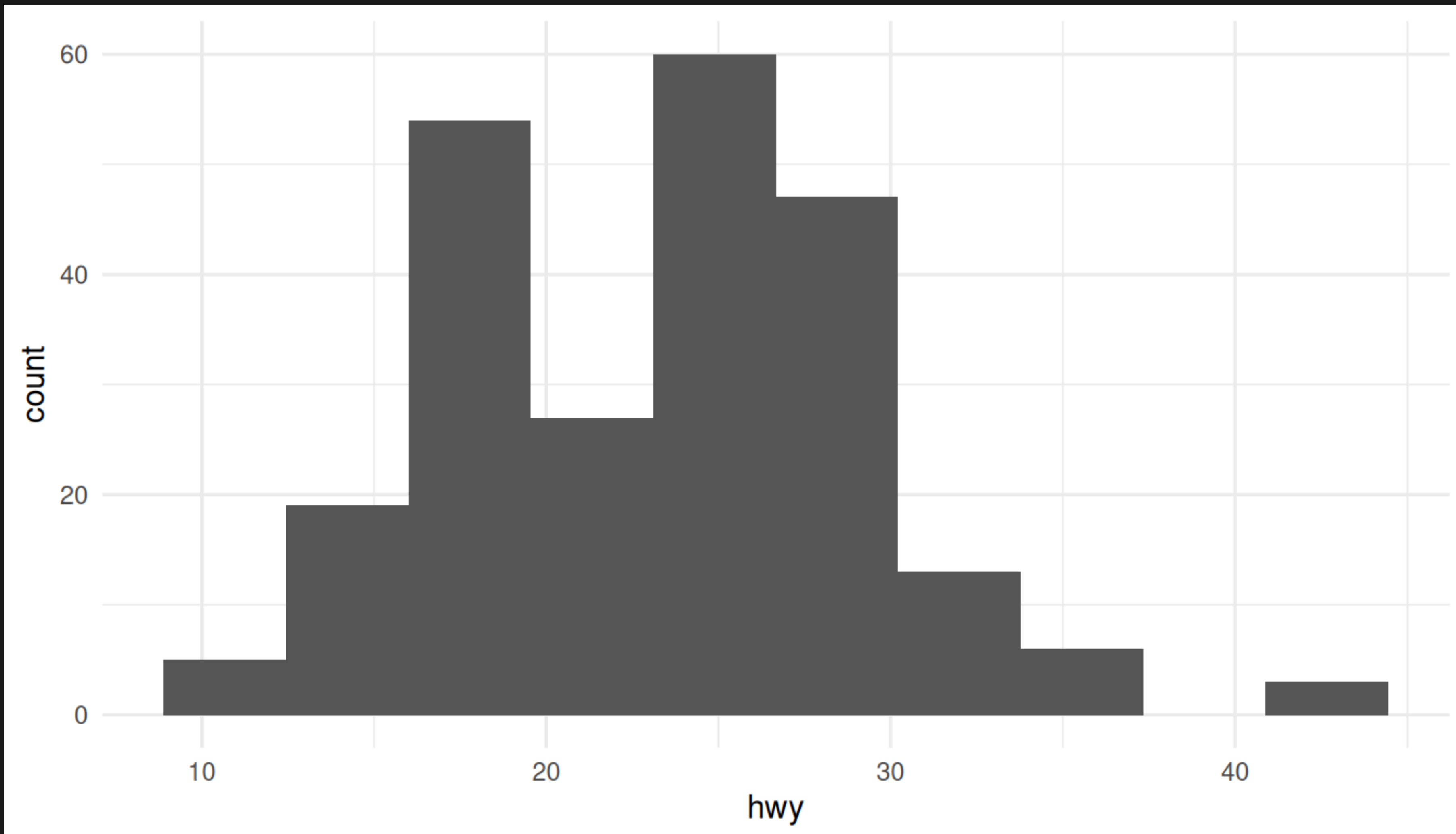
One variable, continuous

- If var is continuous, it makes more sense to show `distributions`

```
p <- ggplot(mpg, aes(hwy))
p + geom_histogram()
```

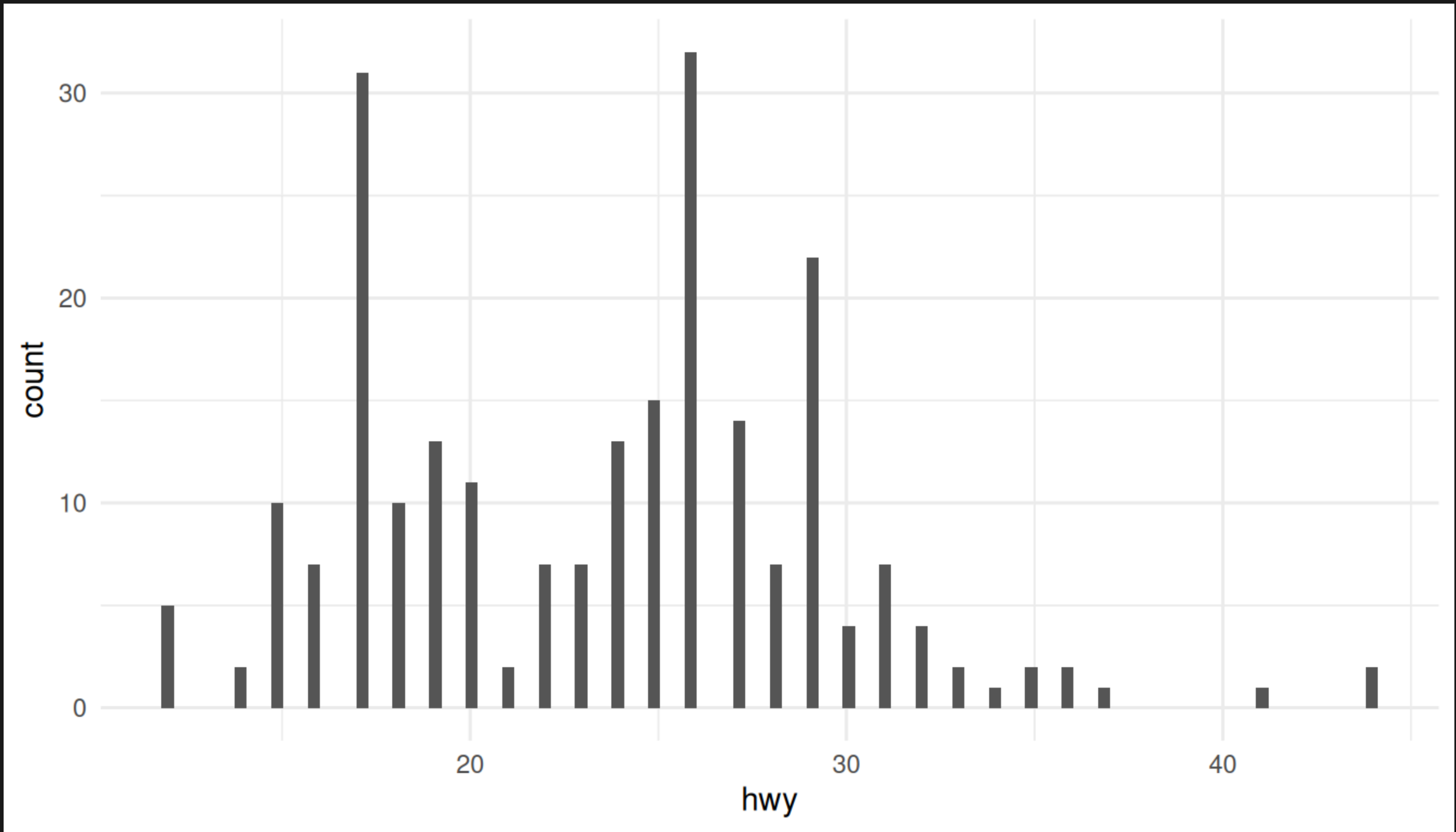
Histograms: binwidth

```
1 p + geom_histogram(bins = 10)
```



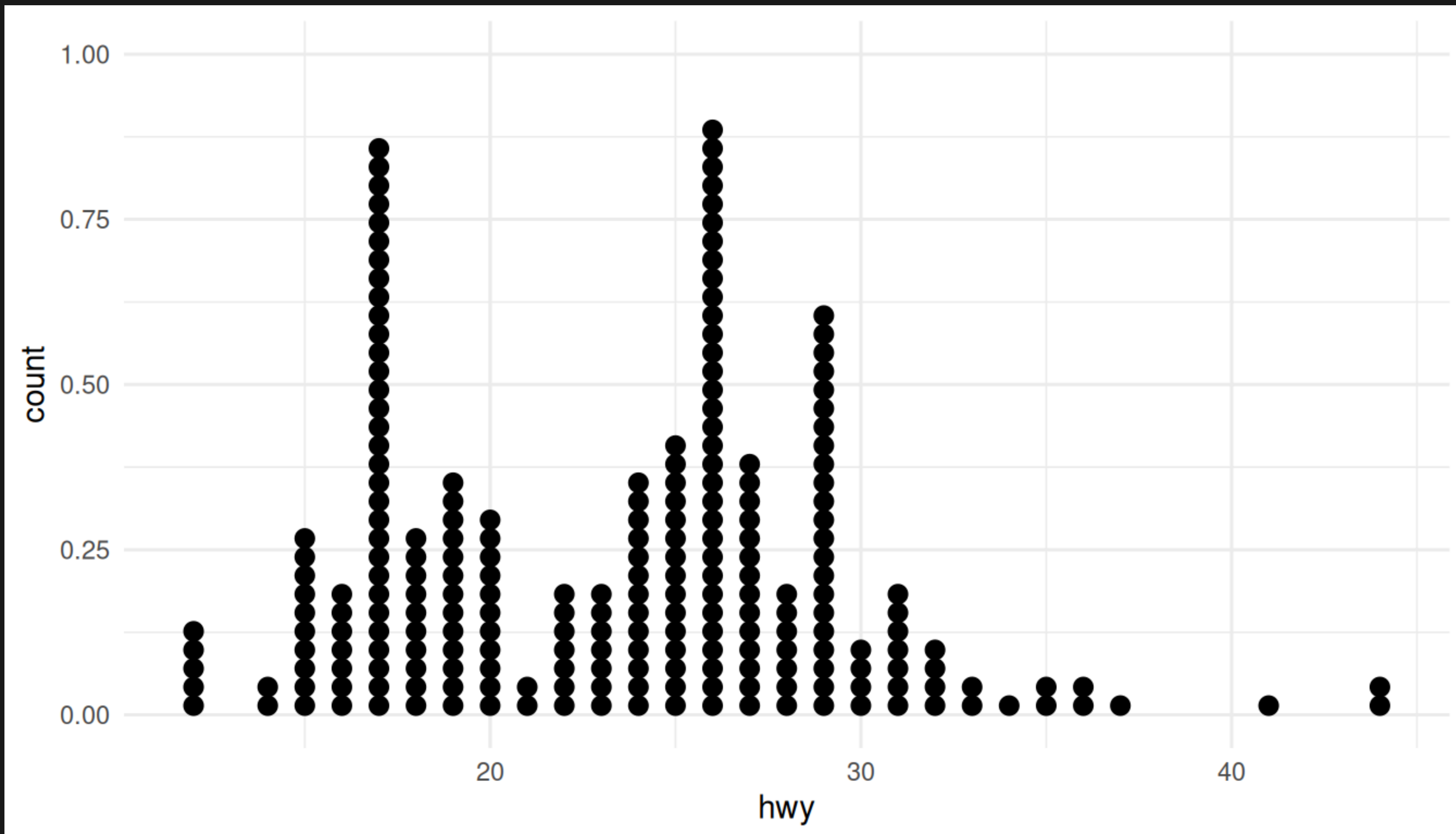
Histograms: binwidth

```r
p + geom_histogram(bins = 100)
```



An alternative to histogram: `dotplot`

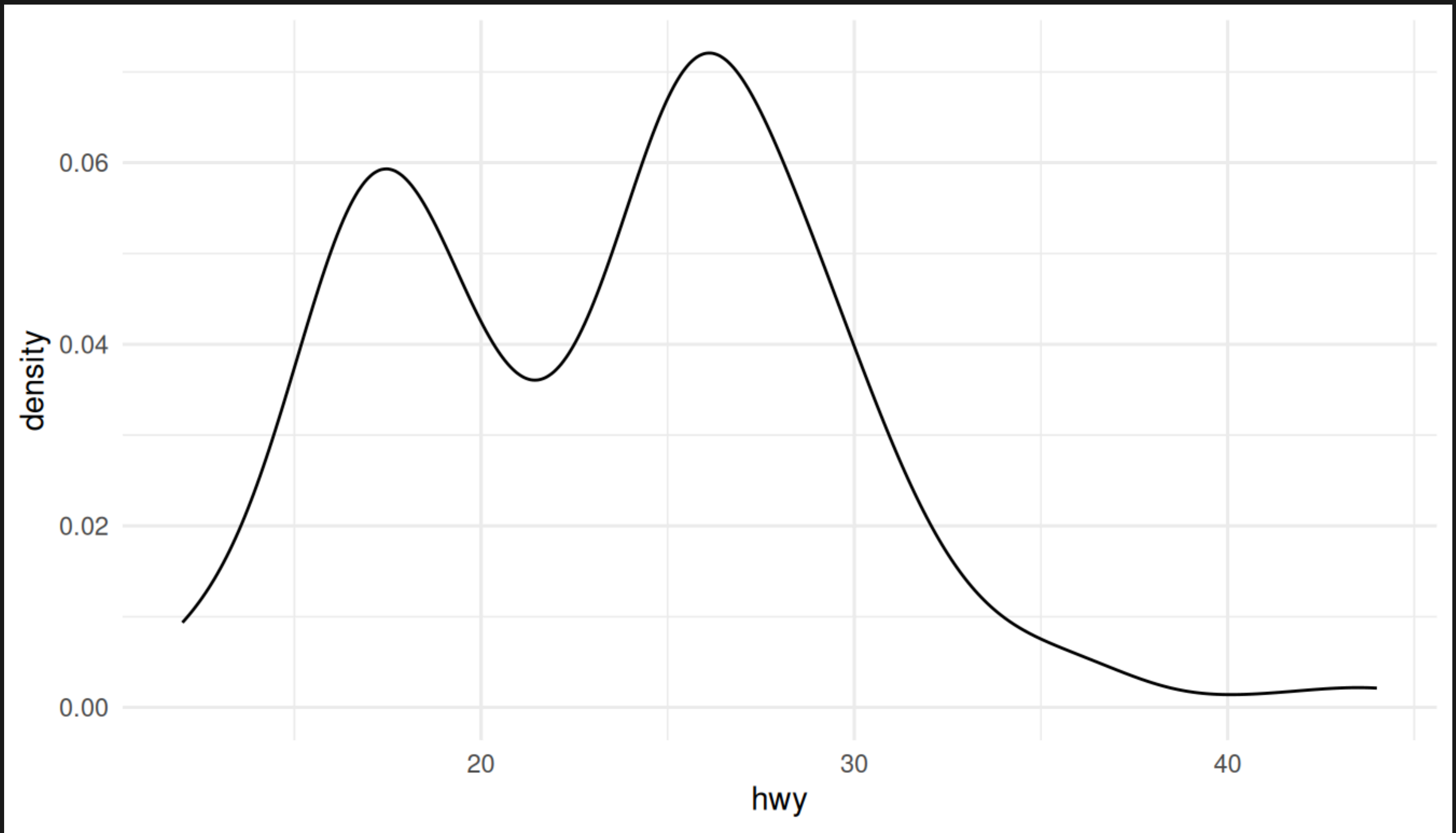
```r
p + geom_dotplot(binwidth = 0.5)
```



Continuous distributions

In this case use `kernel density estimation`

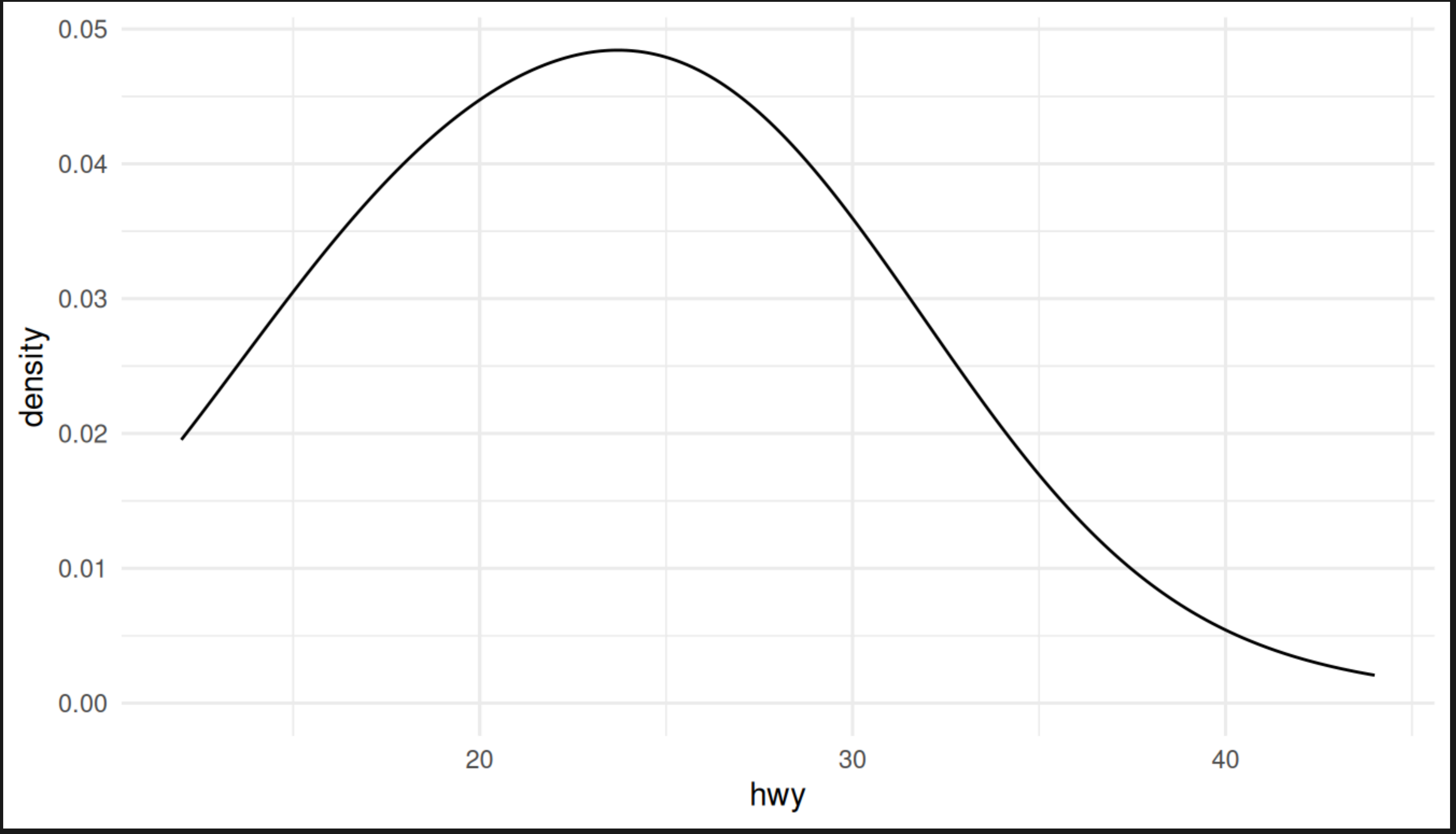
```
p + geom_density()
```



Continuous distribution

In this this case use `kernel density estimation`

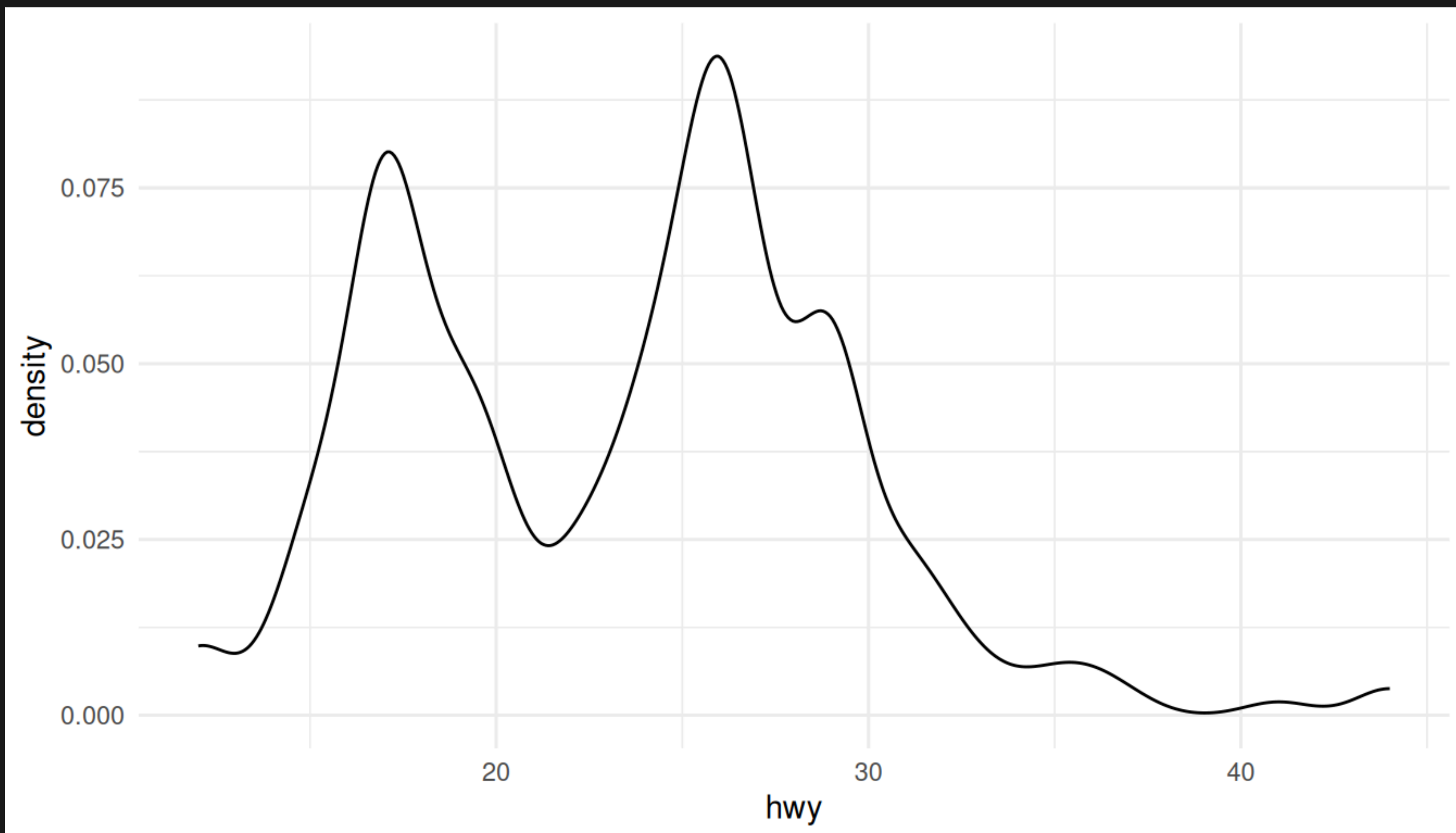
```r
p + geom_density(adjust = 3)
```



Continuous distribution

In this case use `kernel density estimation`

```
p + geom_density(adjust = 0.5)
```



Exploring data: `two` variables

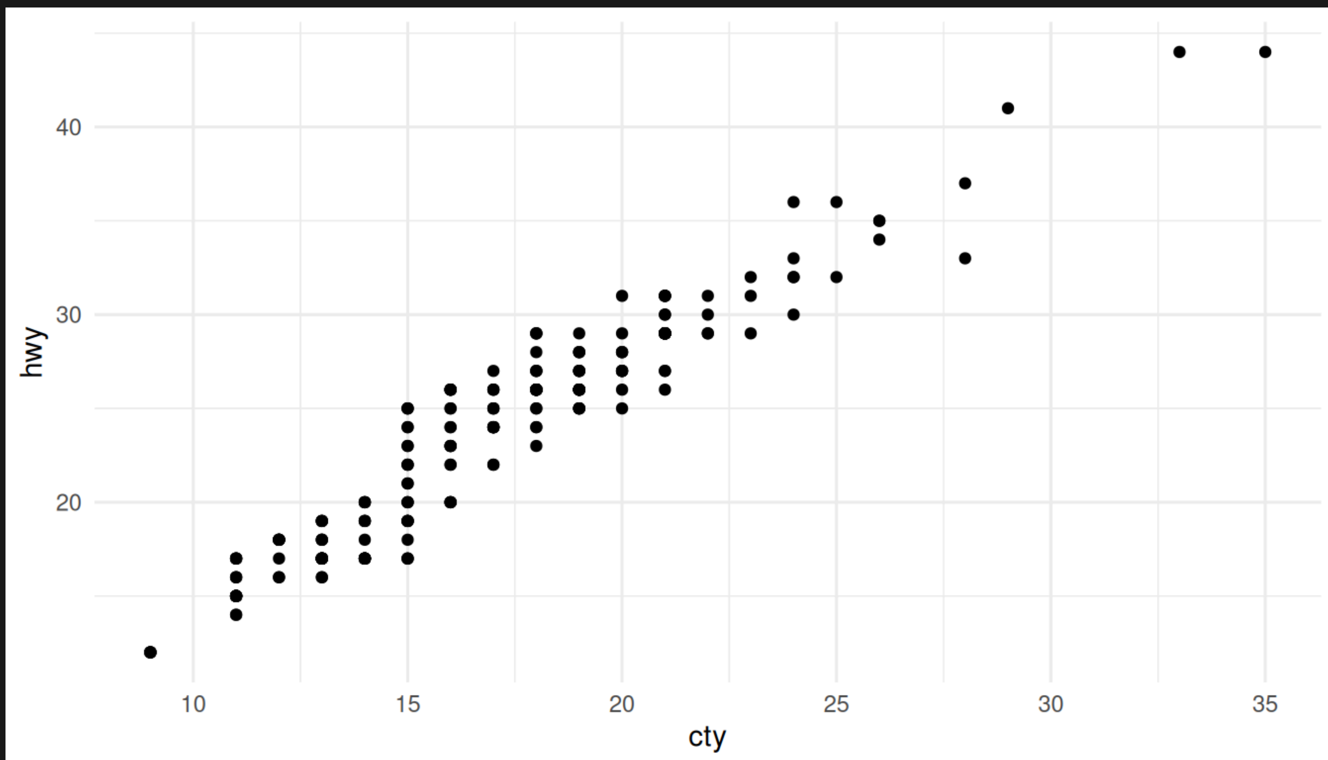
Plot types depend on the variable type

- *both vars continuous*: scatter, smooth
- *one continuous, one discrete*: columns, boxplot, violins
- *both discrete*: count

Scatter

> if two variables are continuous, your choice is `scatter`

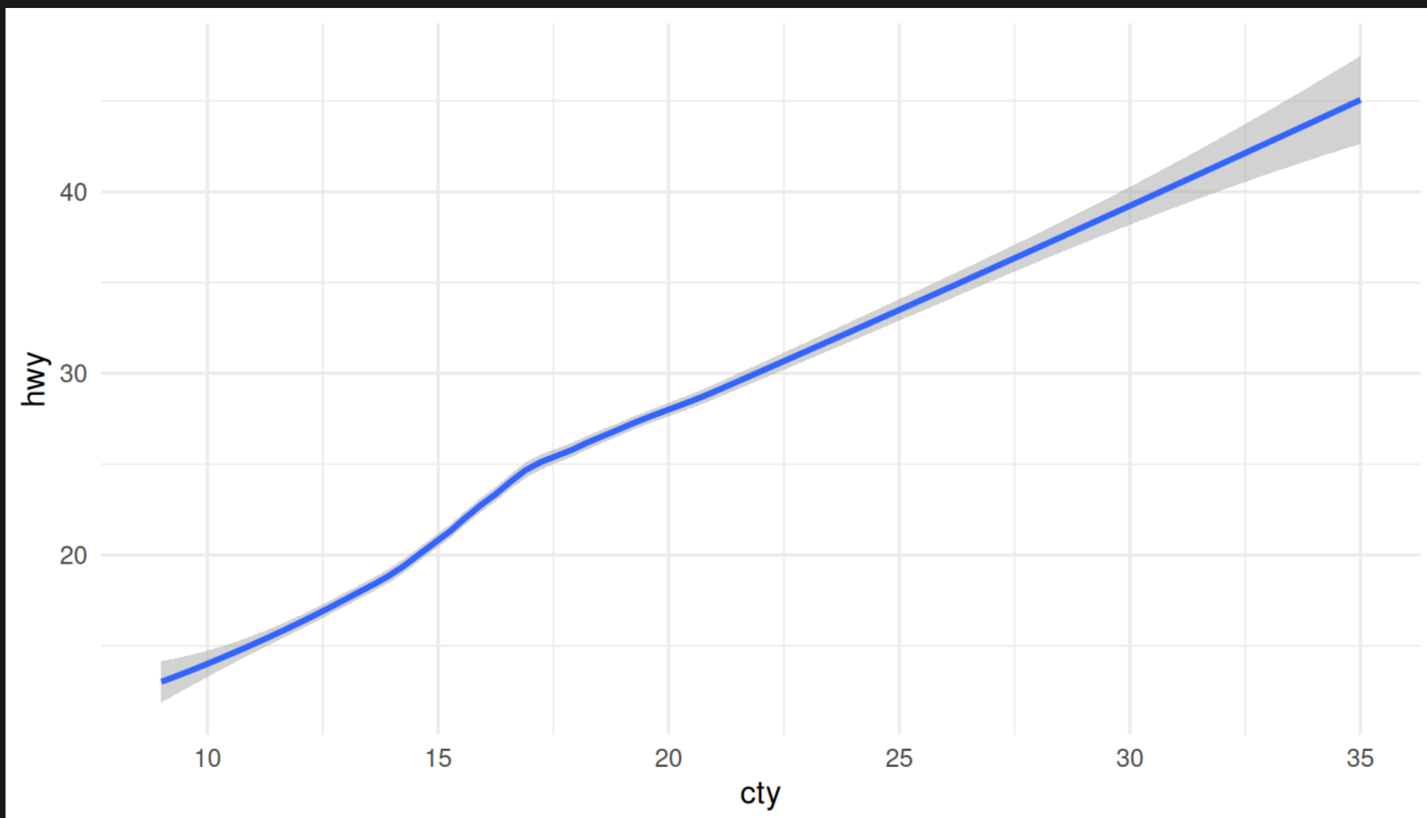
```
p <- ggplot(mpg, aes(x = cty, y = hwy))
p + geom_point()
```



Smooth

> still, you might just want to show the general tendency

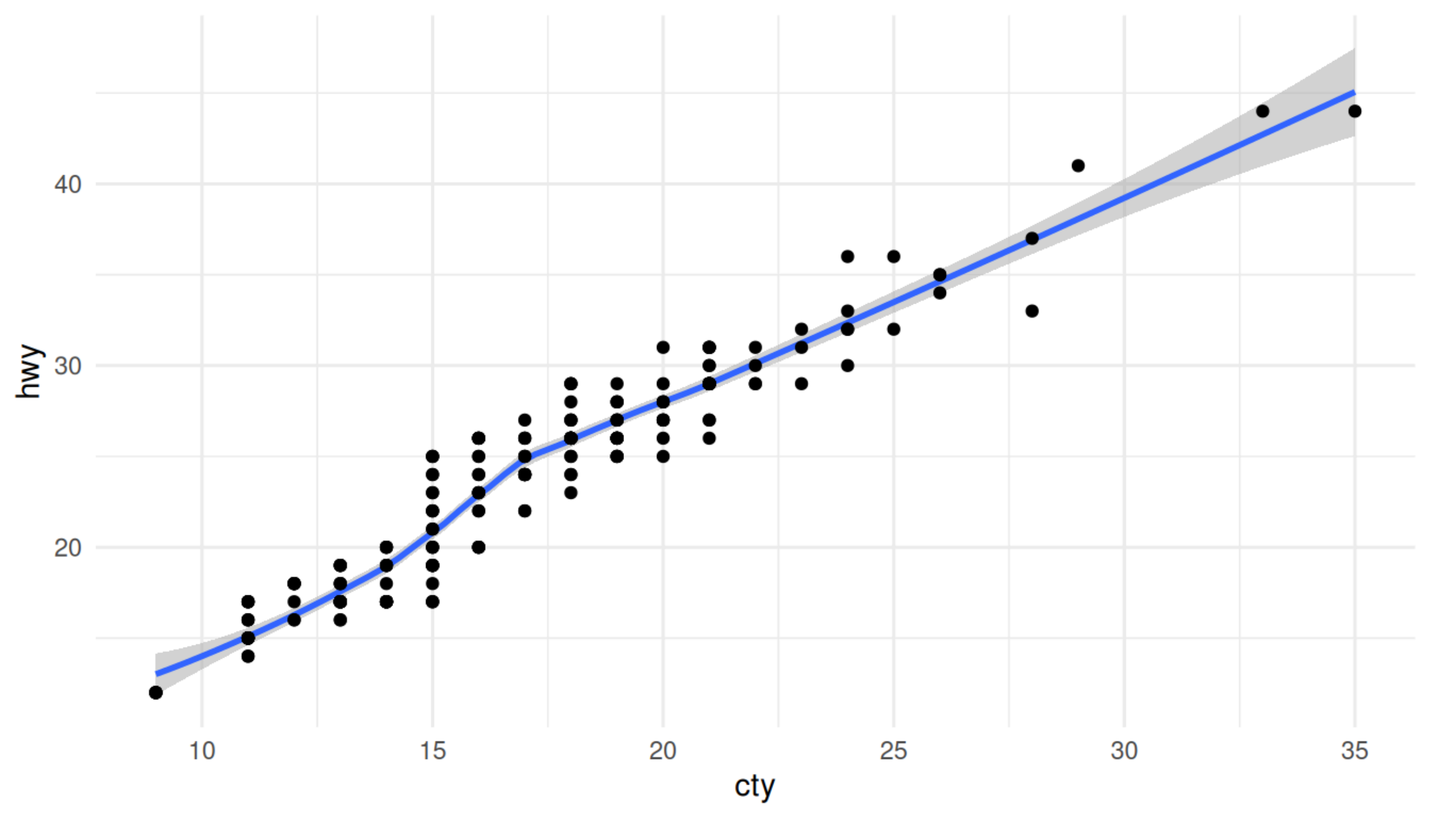
```
p + geom_smooth()
```



Scatter + smooth

> or both

```
p + geom_smooth() + geom_point()
```



Columns: a special type of bars

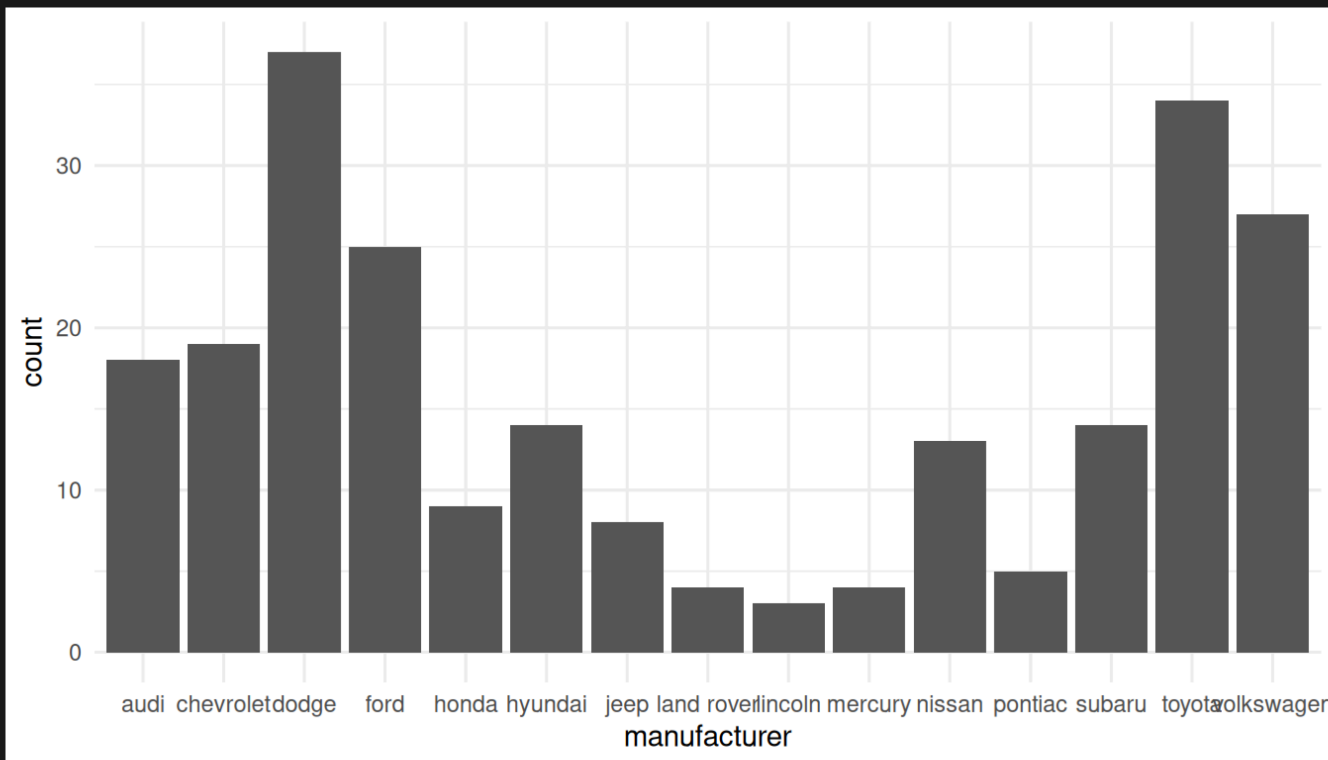
> one variable discrete, one continuous (needs `summarise()`!)

```
mpg %>% group_by(manufacturer) %>% summarise(n = n()) %>%
ggplot(aes(manufacturer, n))+
  geom_col()
```

Columns: why bother?

> we could have used `geom_bar` (that counts for us)

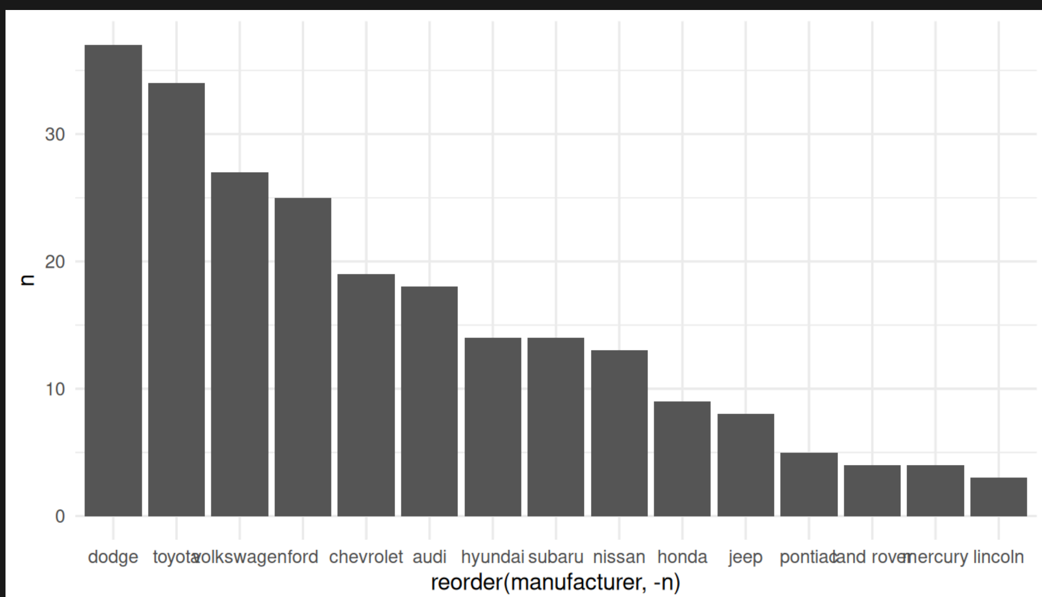
```
mpg %>% ggplot(aes(manufacturer))+
  geom_bar()
```



Columns: a special type of bars

> but `geom_col` gives more options, since now you condition on a proper variable (n). For instance: order by n

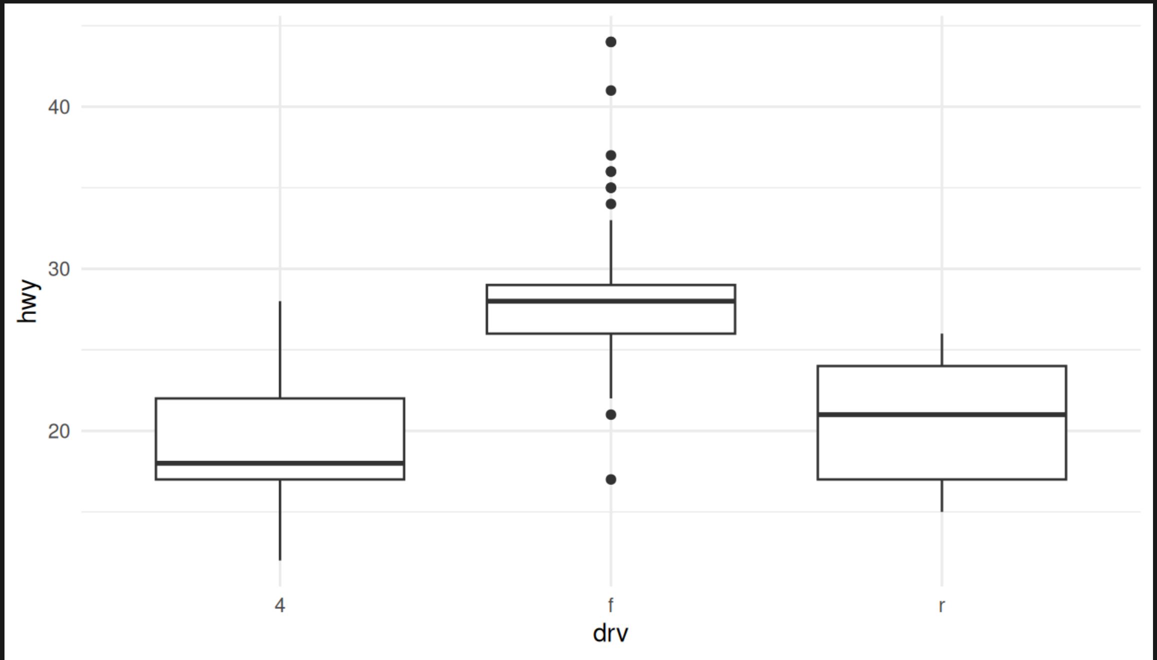
```
mpg %>% group_by(manufacturer) %>% summarise(n = n()) %>%
ggplot(aes(reorder(manufacturer, -n), n))+
  geom_col()
```



Boxplots

> boxplots show a distribution but can do so over different levels of a categorical var

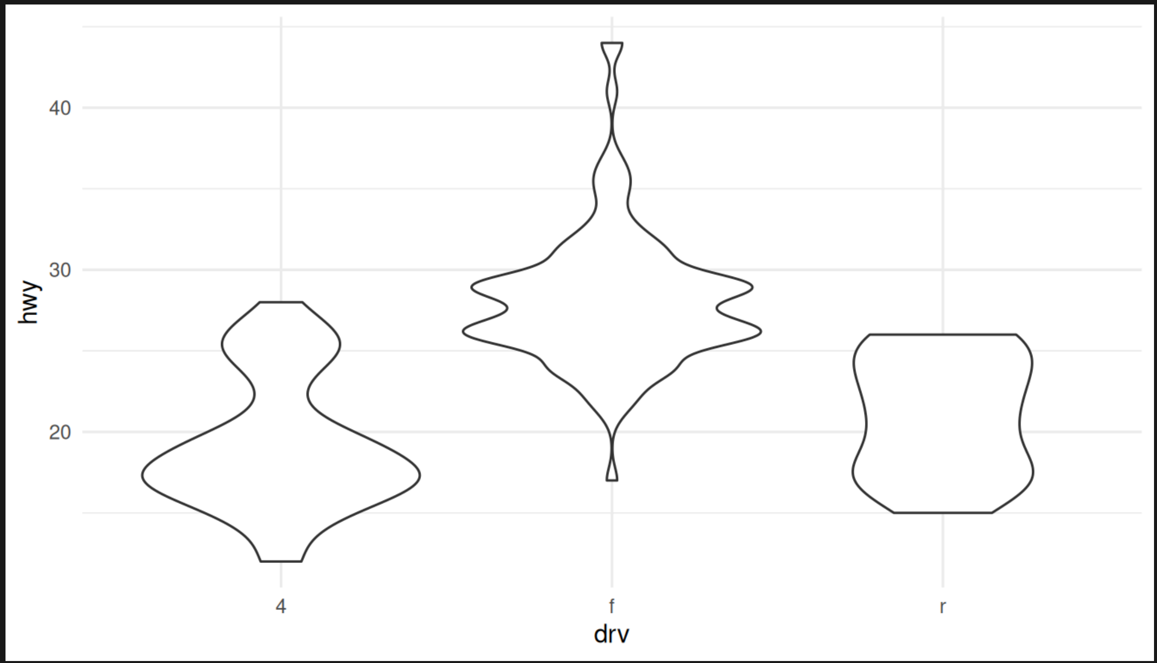
```r
mpg %>% ggplot(aes(drv, hwy))+
  geom_boxplot()
```



An alternative to boxplot: violin

> boxplots are bulky and only show relevant info. Want full distribution? Use violins

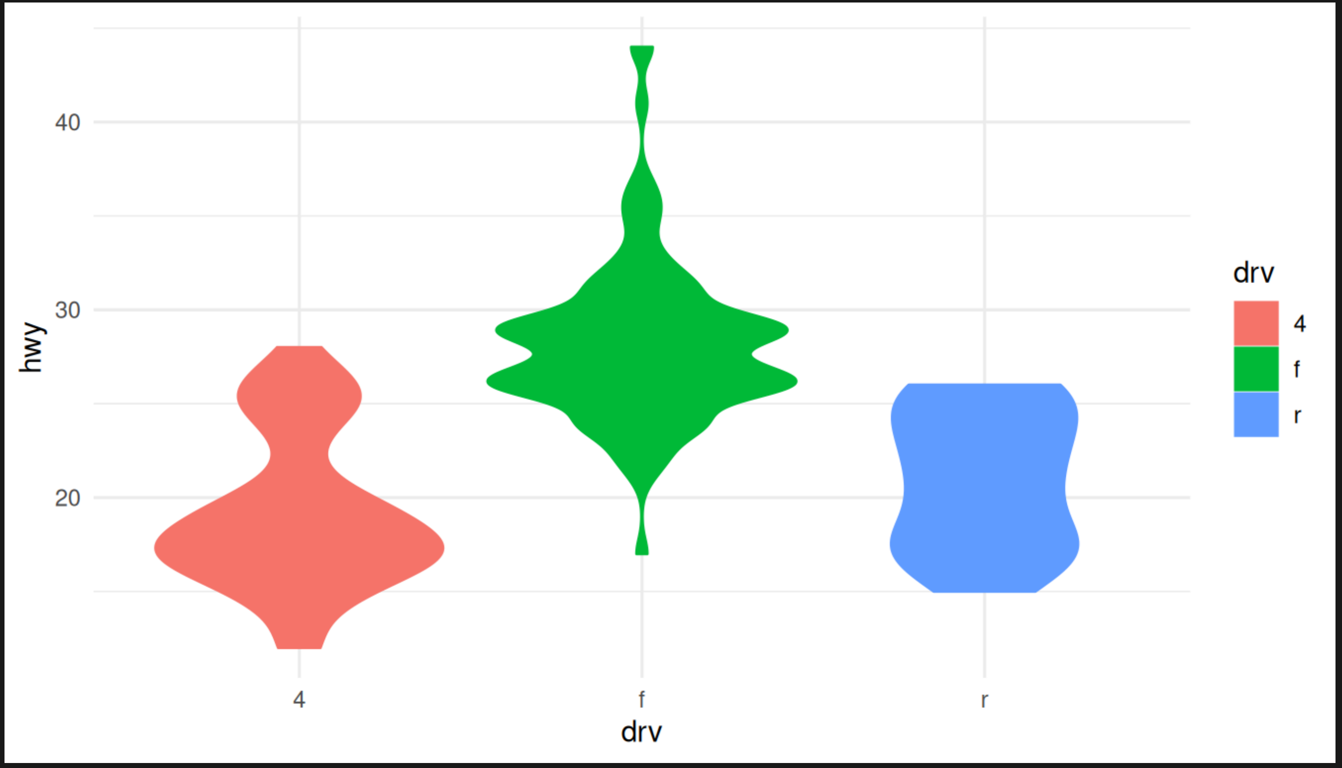
```r
mpg %>% ggplot(aes(drv, hwy))+
  geom_violin()
```



An alternative to boxplot: violin

> remember: all is modular. We can always add color, fill…

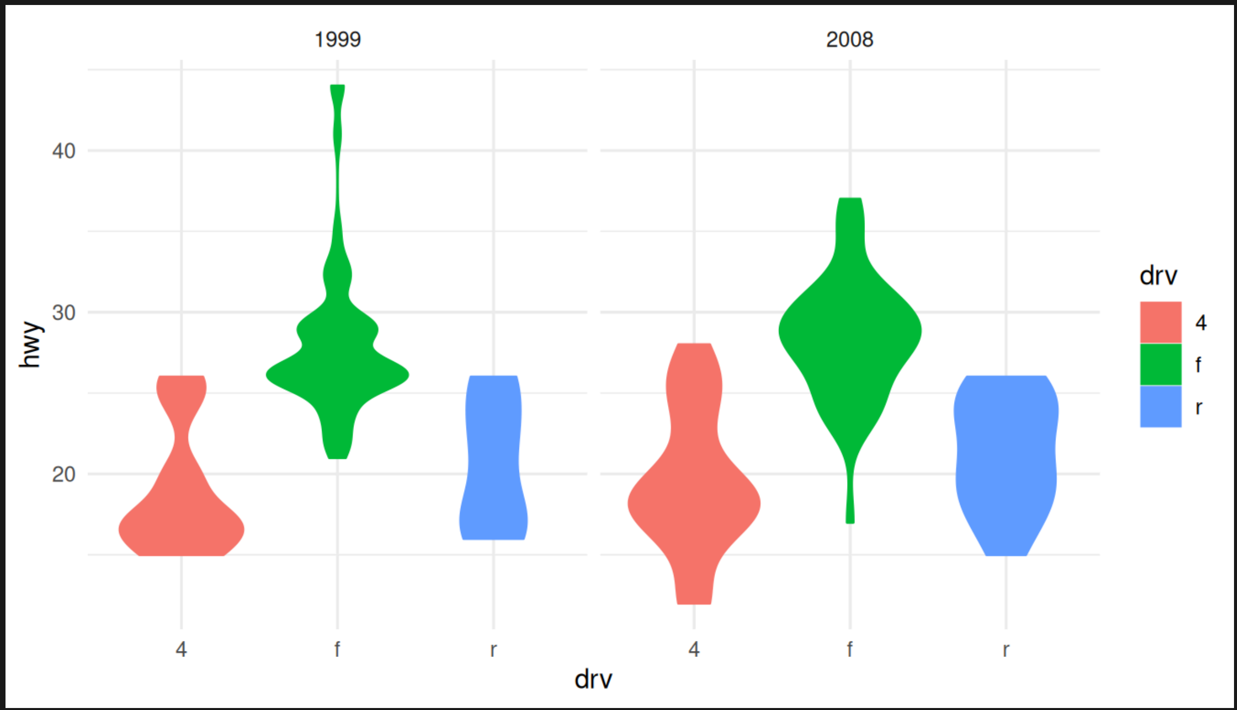
```
mpg %>% ggplot(aes(drv, hwy, color = drv, fill = drv))+
  geom_violin()
```



An alternative to boxplot: violin

> remember: all is modular. …facets

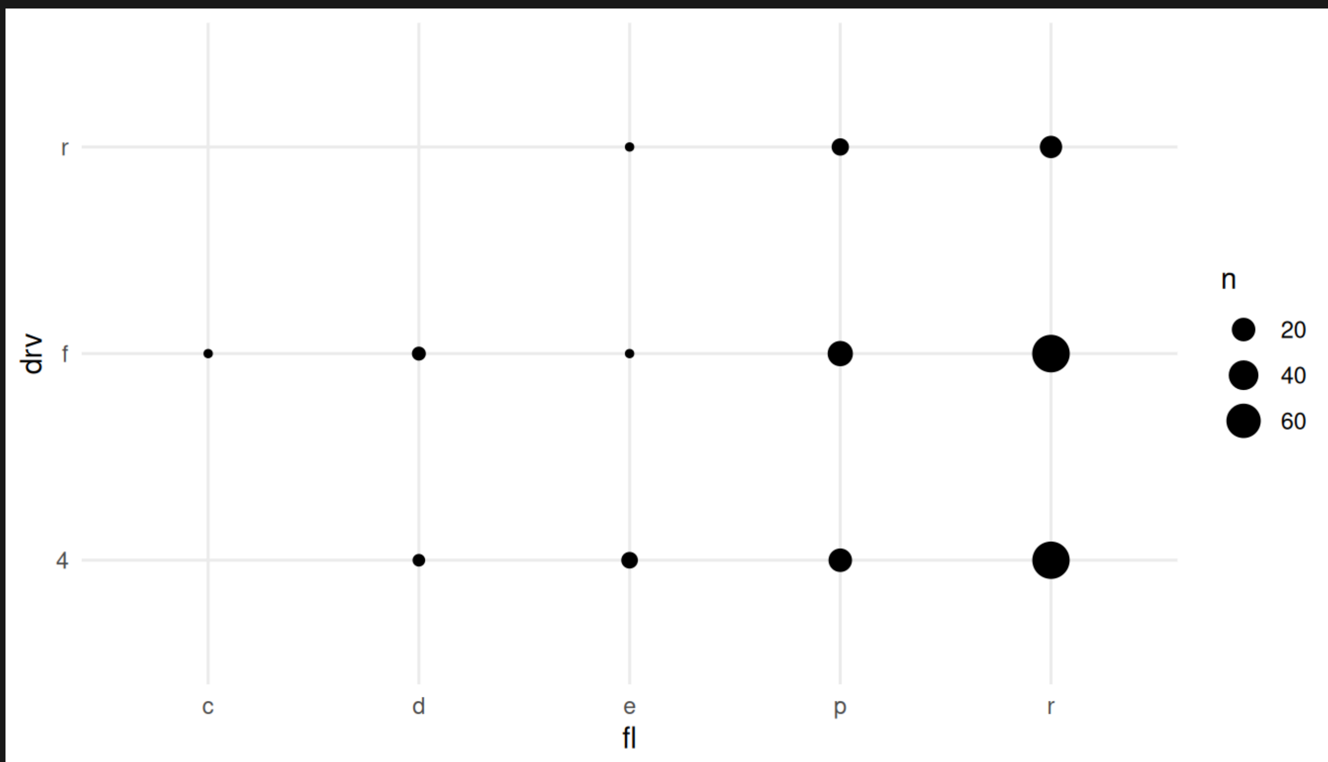
```r
mpg %>% ggplot(aes(drv, hwy, color = drv, fill = drv))+
  geom_violin()+
  facet_grid(.~year)
```



Counts

> two categorical variables: count their cross-tabulation

```
mpg %>% ggplot(aes(fl, drv))+
  geom_count()
```



Exploring data: `three` variables

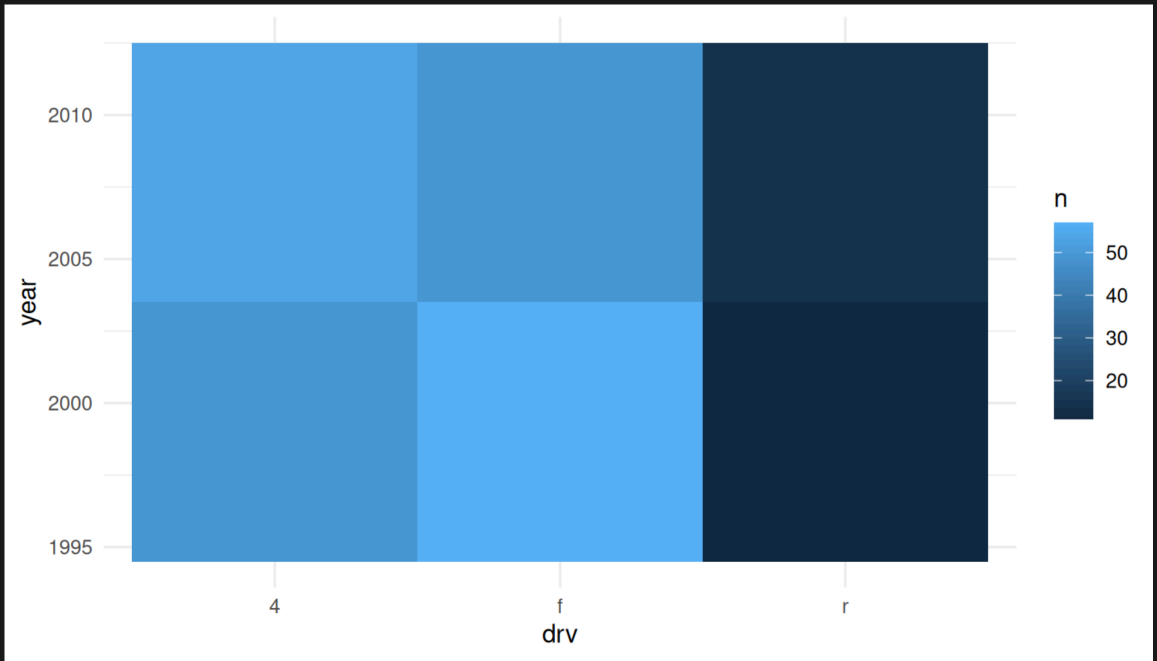
Plot types depend on the variable type

- *all continuous*: contour plot (think: elevation in maps)
- *some discrete*: tile

Tile

> two variables for the x,y grid. A third the color of the cell. (needs `summarise()`!)

```
mpg %>% group_by(year, drv) %>% summarise(n = n()) %>%
  ggplot(aes(x = drv, y = year, fill = n)) +  geom_tile()
```



Additional resources

- the ggplot `cheatsheet` is your friend (Help -> cheatsheets)
- `stack overflow` helps out for trickier questions
- `chatGPT` is your friend, too (but beware)
- not feeling inspired?
  - 50 cool visualisations
  - a complete list of possibilities in R and python