

Visualización de datos con ggplot2

Extraído del Capítulo 1 de: [R for Data Science, Wickham & Grolemund, 2017](#)

https://diegokoz.github.io/intro_ds/fuentes/ggplot2-cheatsheet-2.1-Spanish.pdf

<https://www.maths.usyd.edu.au/u/UG/SM/STAT3022/r/current/Misc/data-visualization-2.1.pdf>

Introducción: Instalación de ggplot2

- > `install.packages("tidyverse")` #instala el paquete tidyverse en su computadora
- > `library(tidyverse)` #trae a tidyverse al ambiente de trabajo de RStudio

- > Sólo se instala una vez.

- > `mpg` # es una de las data.frames que trae ggplot2 (mirar en el Environment)

Introducción: Instalación de ggplot2

> mpg # es una de las data.frames que trae ggplot2 (mirar en el Environment)

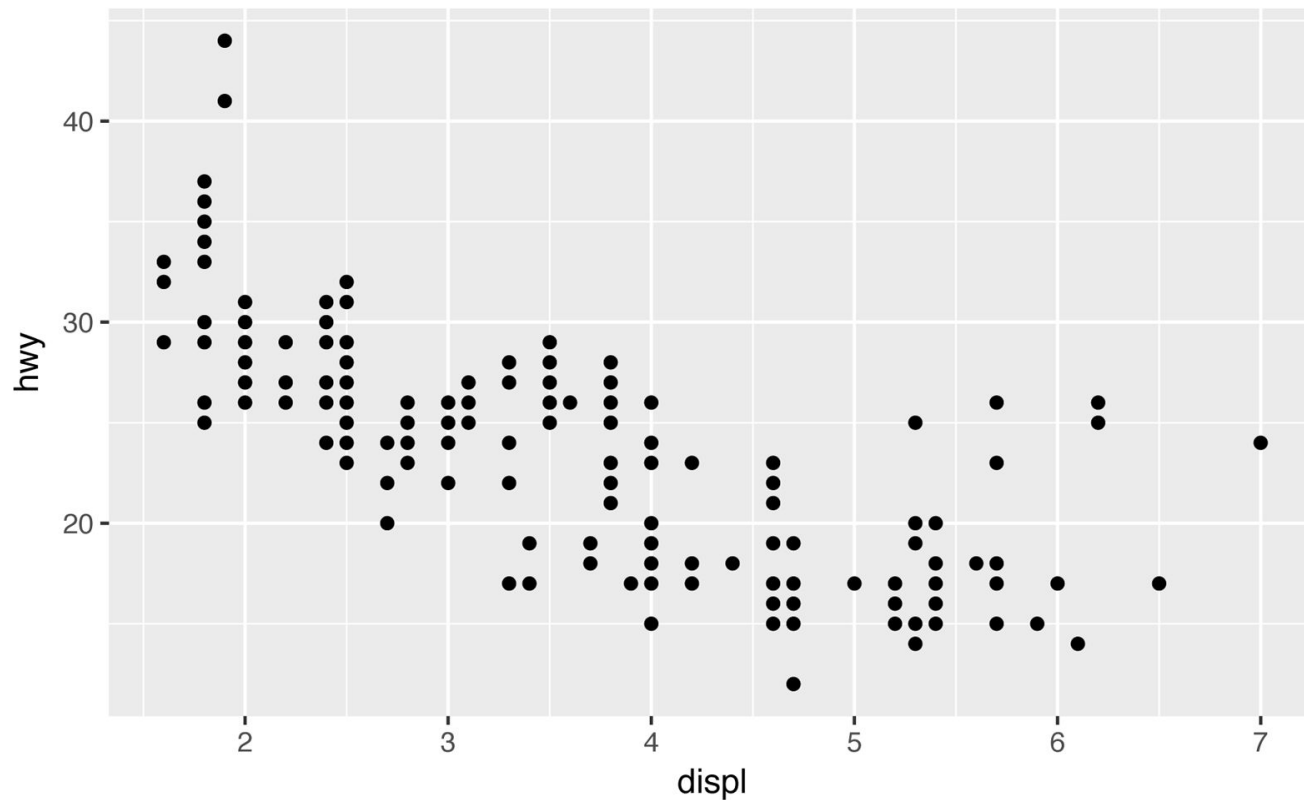
```
# A tibble: 234 x 11
  manufacturer model      displ  year   cyl trans      drv    cty   hwy fl      class
  <chr>         <chr>    <dbl> <int> <int> <chr>    <chr> <int> <int> <chr> <chr>
1 audi         a4        1.8   1999     4 auto(l5)  f      18    29 p      compact
2 audi         a4        1.8   1999     4 manual(m5) f      21    29 p      compact
3 audi         a4        2     2008     4 manual(m6) f      20    31 p      compact
4 audi         a4        2     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     2008     4 manual(m6) 4      20    28 p      compact
# ... with 224 more rows
```

> ?mpg # se explica el significado de los datos de mpg.

- **Pregunta:** ¿Los automóviles con motores grandes consumen más combustible que los que tienen motores pequeños?

¿Los automóviles con motores grandes...?

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



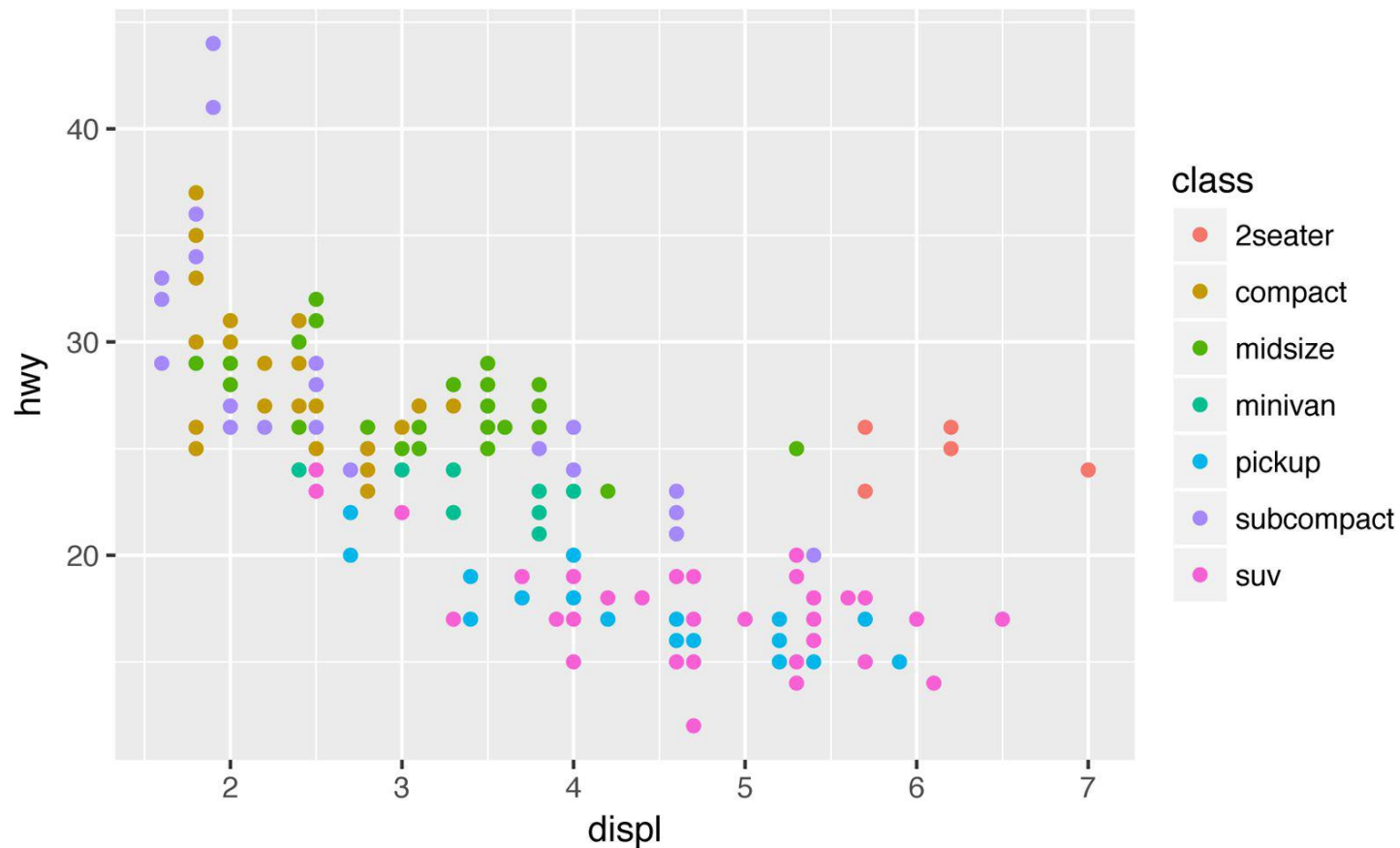
Estructura general de ggplot

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

Probar con otros <GEOM_FUNCTION>, con otros <DATA> y con otros <MAPPINGS>

¿Los automóviles con motores grandes...?

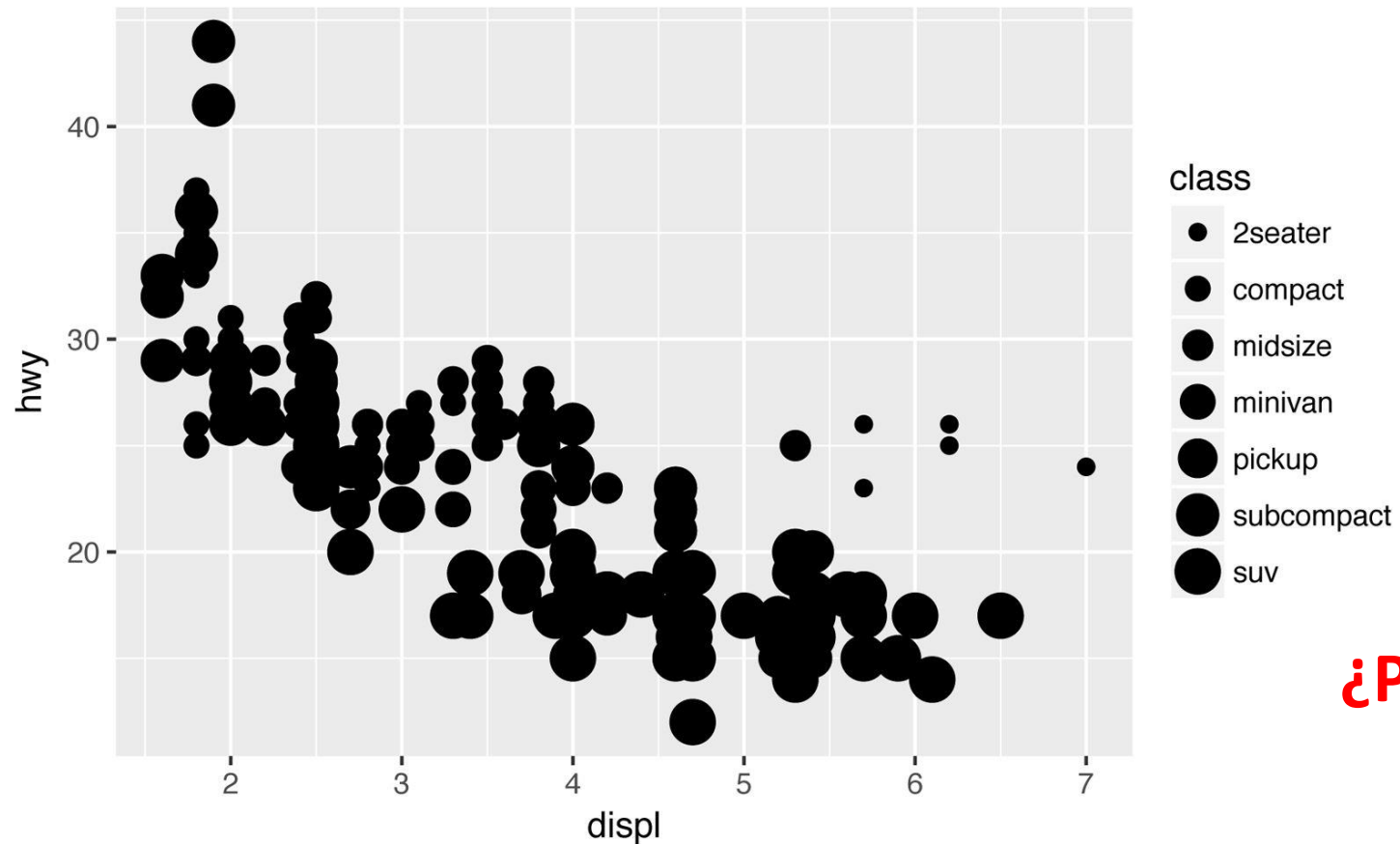
```
> ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy, color = class))
```



¿Los automóviles con motores grandes...?

```
> ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy, size = class))
```

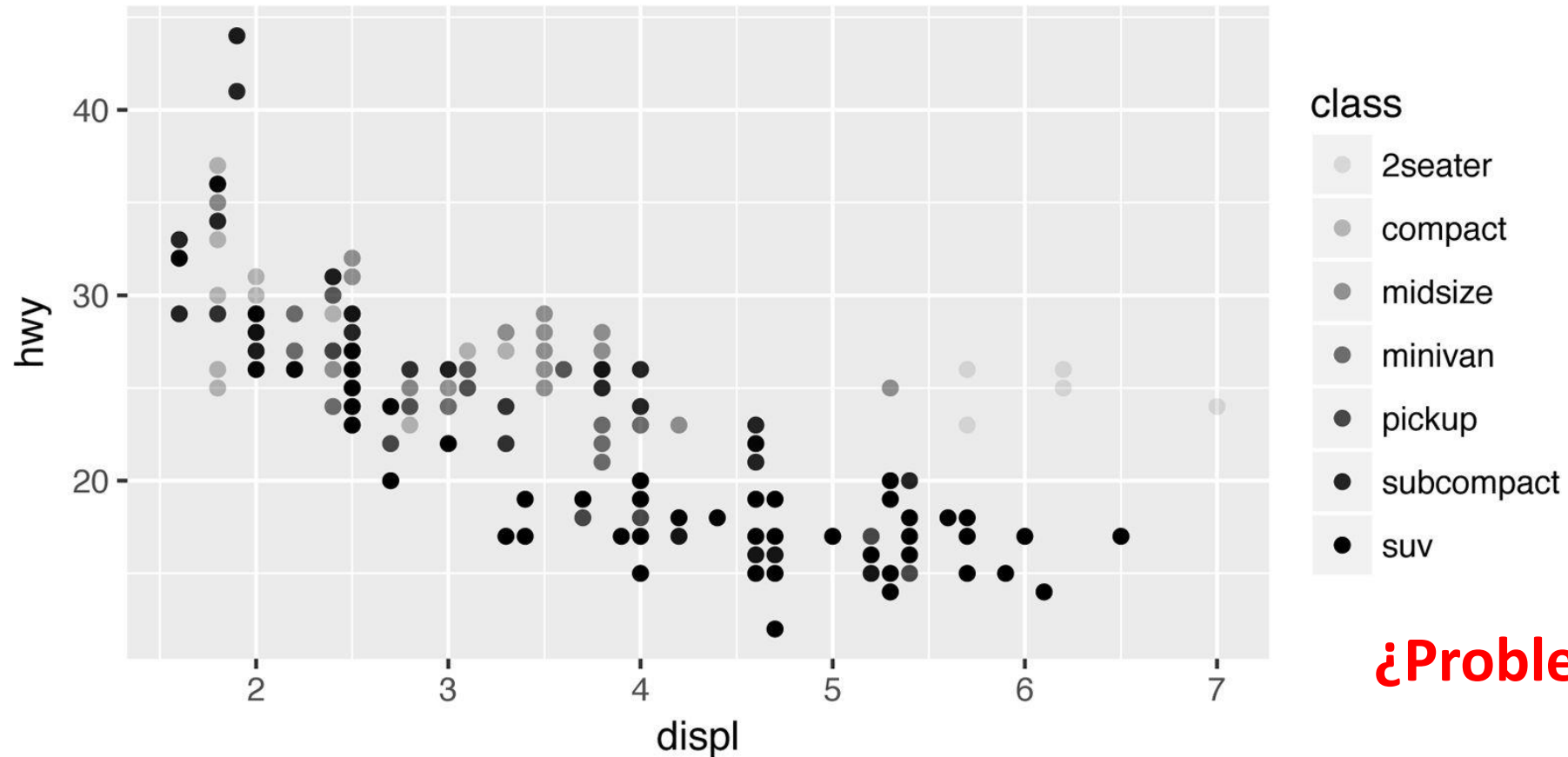
#> Warning: Using size for a discrete variable is not advised.



¿Problemas?

¿Los automóviles con motores grandes...?

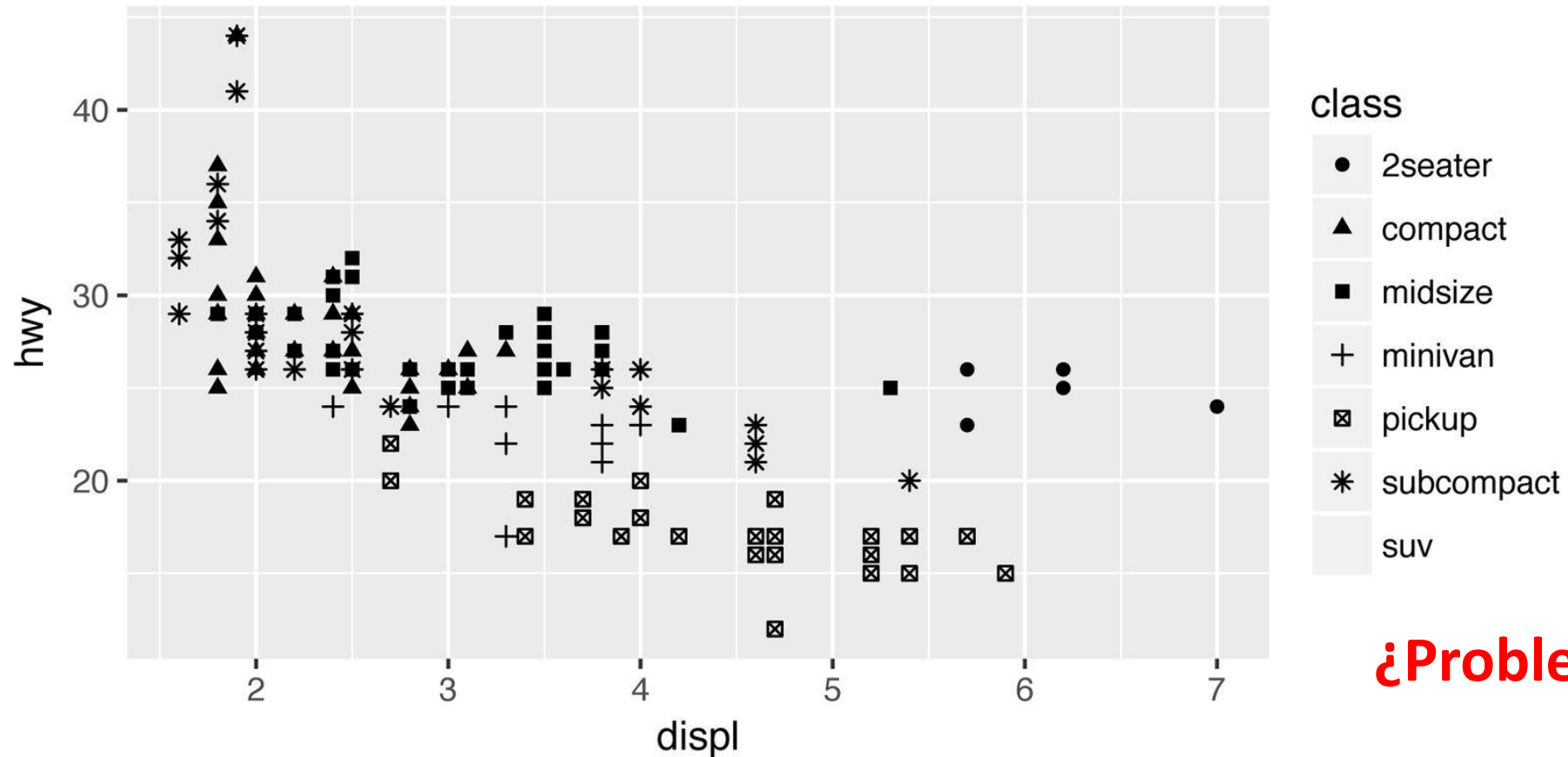
```
> ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy, alpha = class))
```



¿Problemas?

¿Los automóviles con motores grandes...?

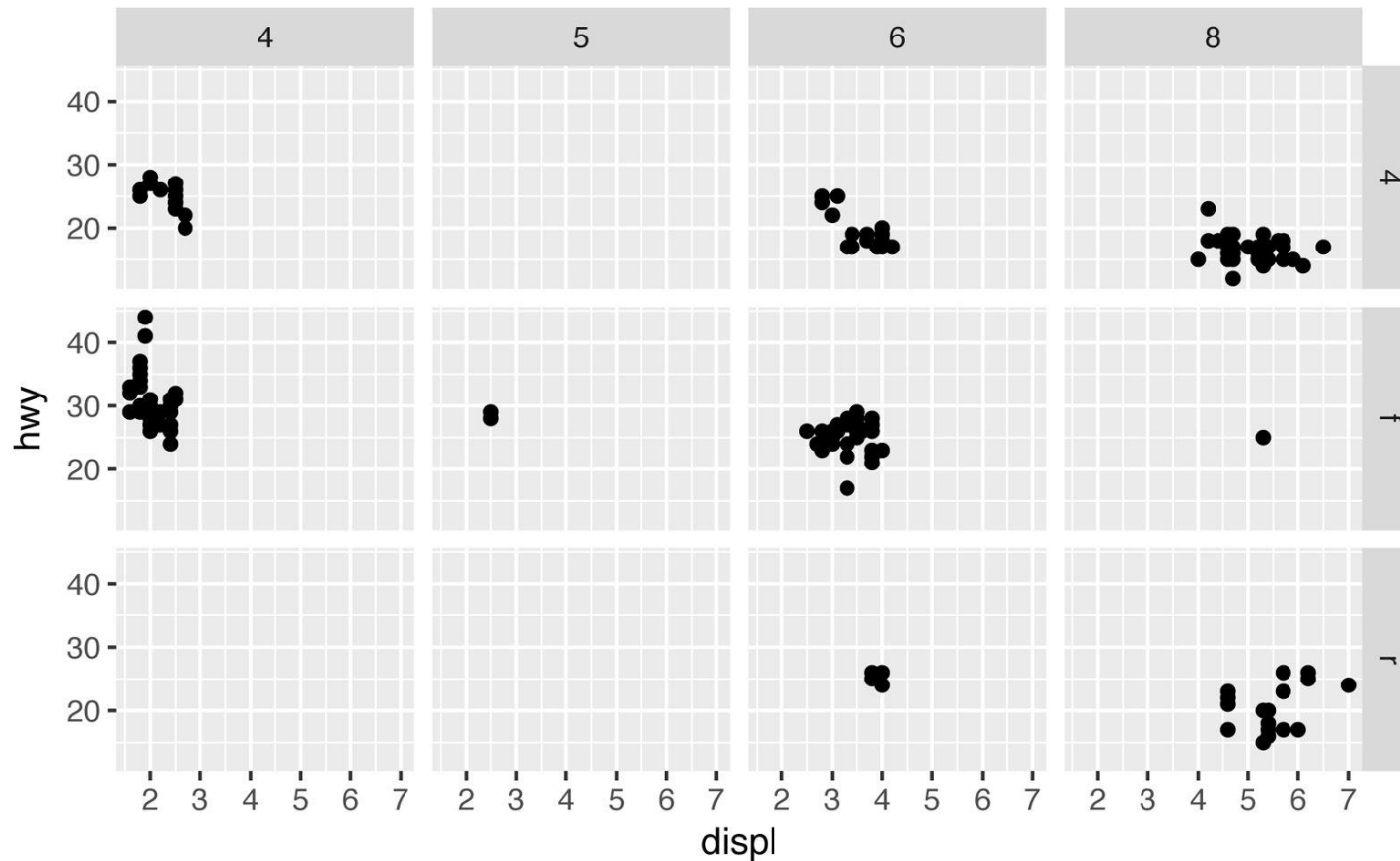
```
> ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy, shape = class))
```



¿Problemas?

FACETS

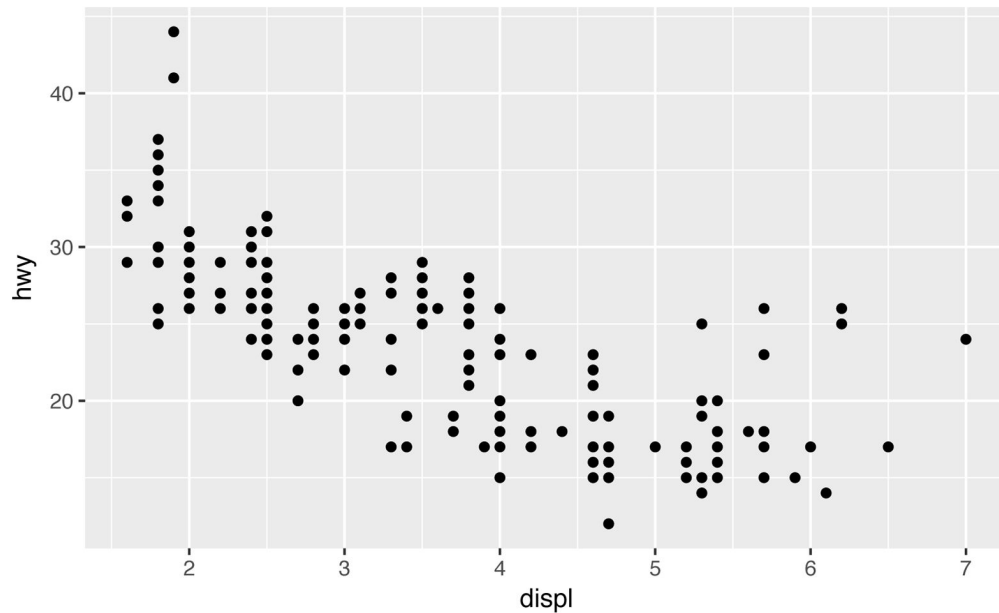
```
> ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy)) +  
  facet_grid(drv ~ cyl)
```



Objetos Geométricos: geom_*()

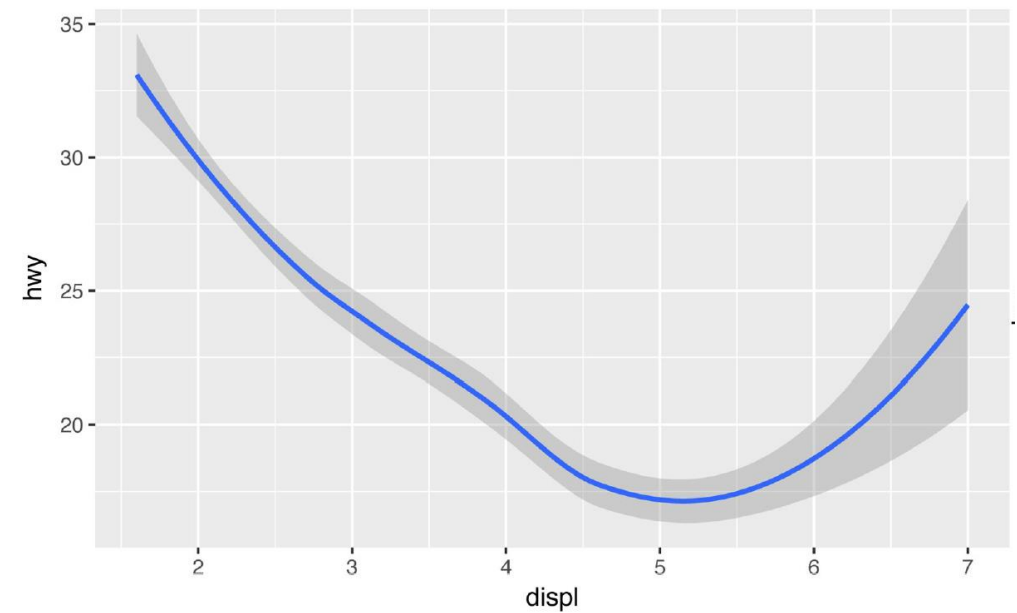
```
> ggplot(data = mpg) +
```

```
  geom_point(mapping = aes(x = displ, y = hwy))
```



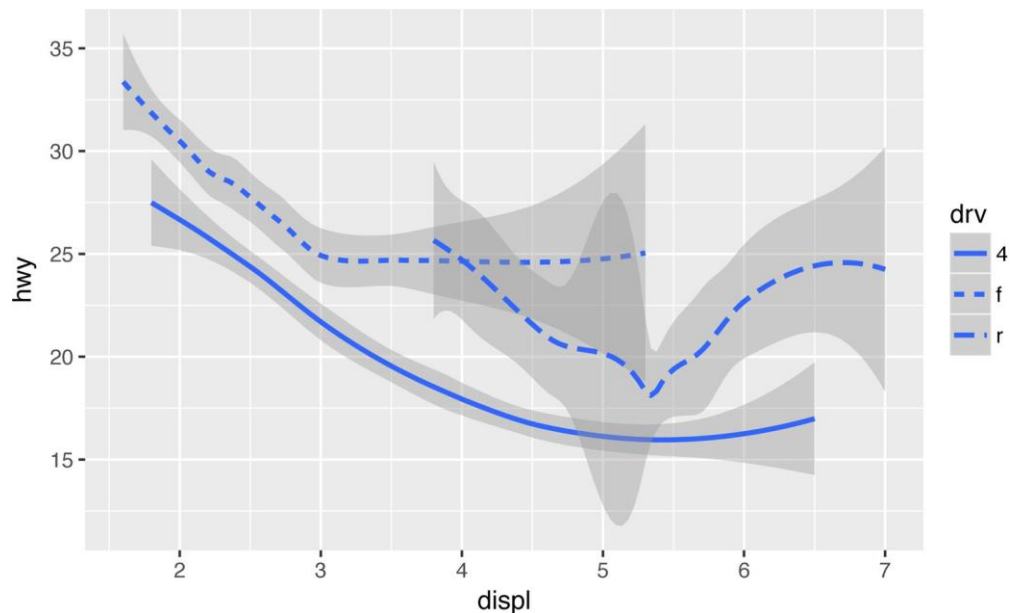
```
> ggplot(data = mpg) +
```

```
  geom_smooth(mapping = aes(x = displ, y = hwy))
```

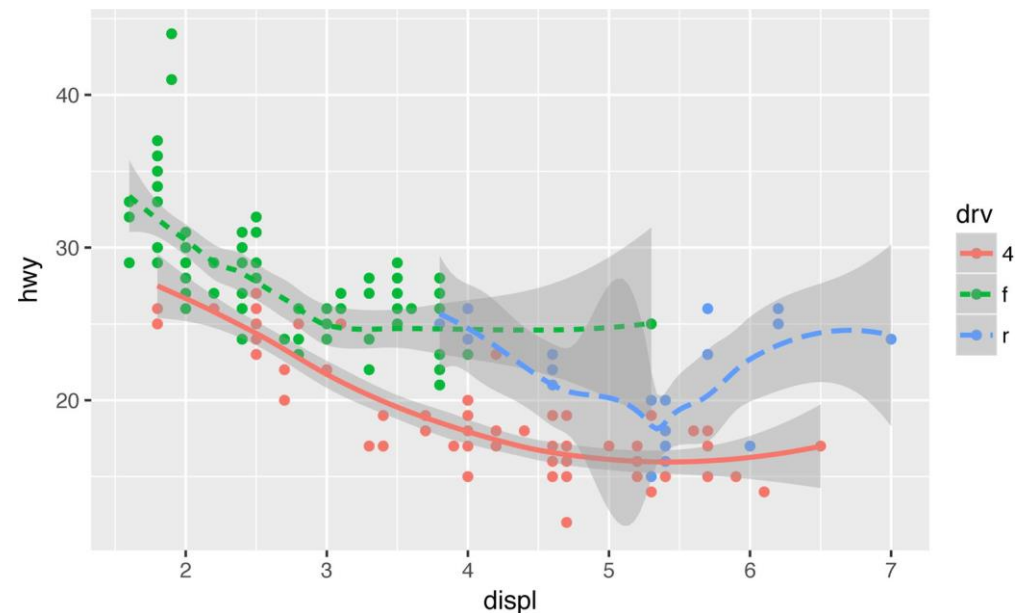


Objetos Geométricos: geom_*()

```
> ggplot(data = mpg) +  
  geom_smooth(mapping = aes(x = displ, y =  
    hwy, linetype = drv))
```

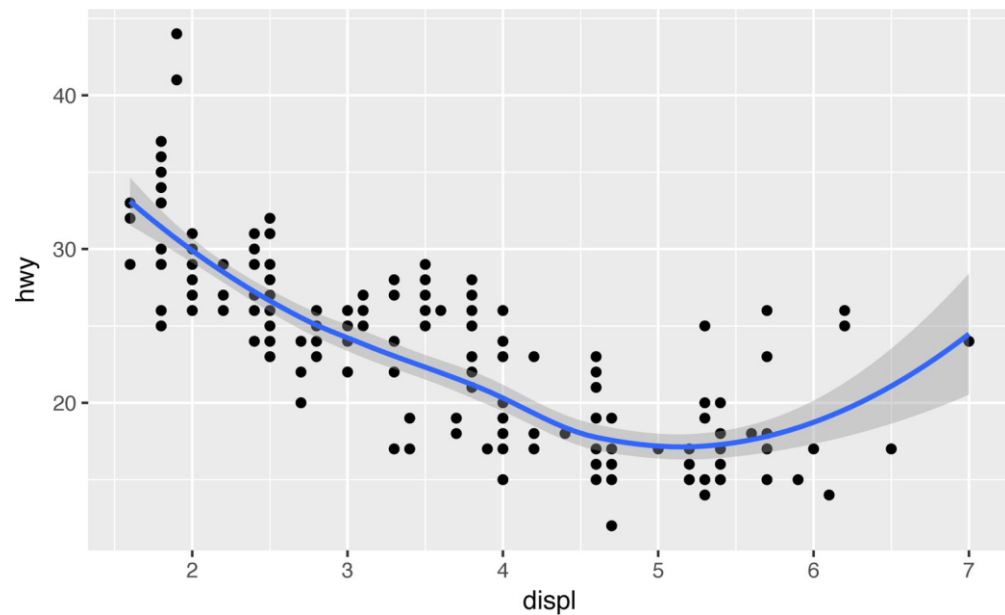


```
> ggplot(data = mpg) +  
  geom_smooth(mapping = aes(x = displ, y =  
    hwy, linetype = drv, color=drv)) +  
  geom_point(mapping = aes(x = displ, y = hwy,  
    color = drv))
```

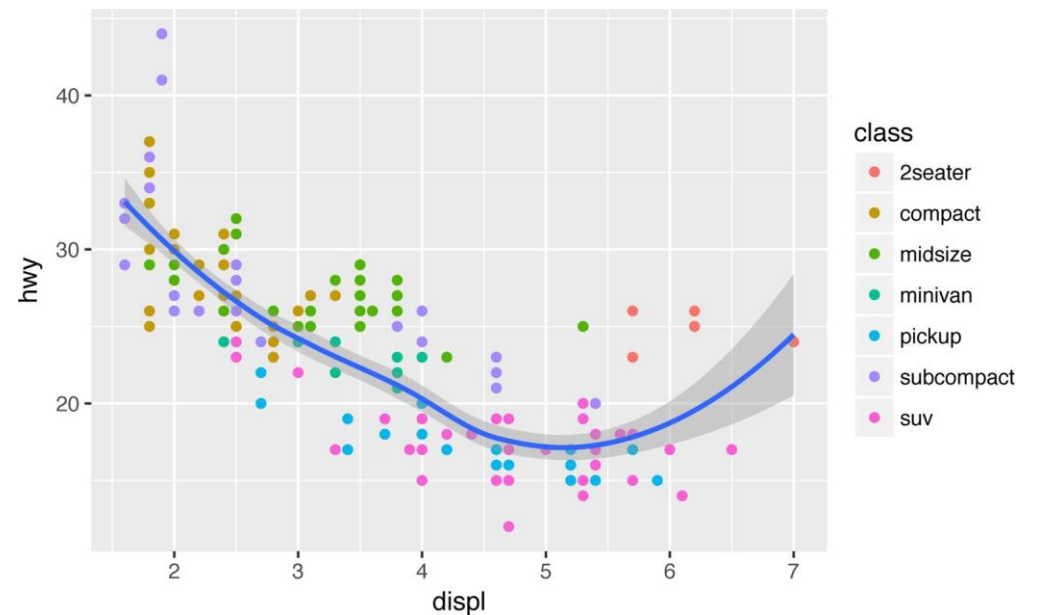


Objetos Geométricos: geom_*()

```
> ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +  
  geom_point() +  
  geom_smooth()
```

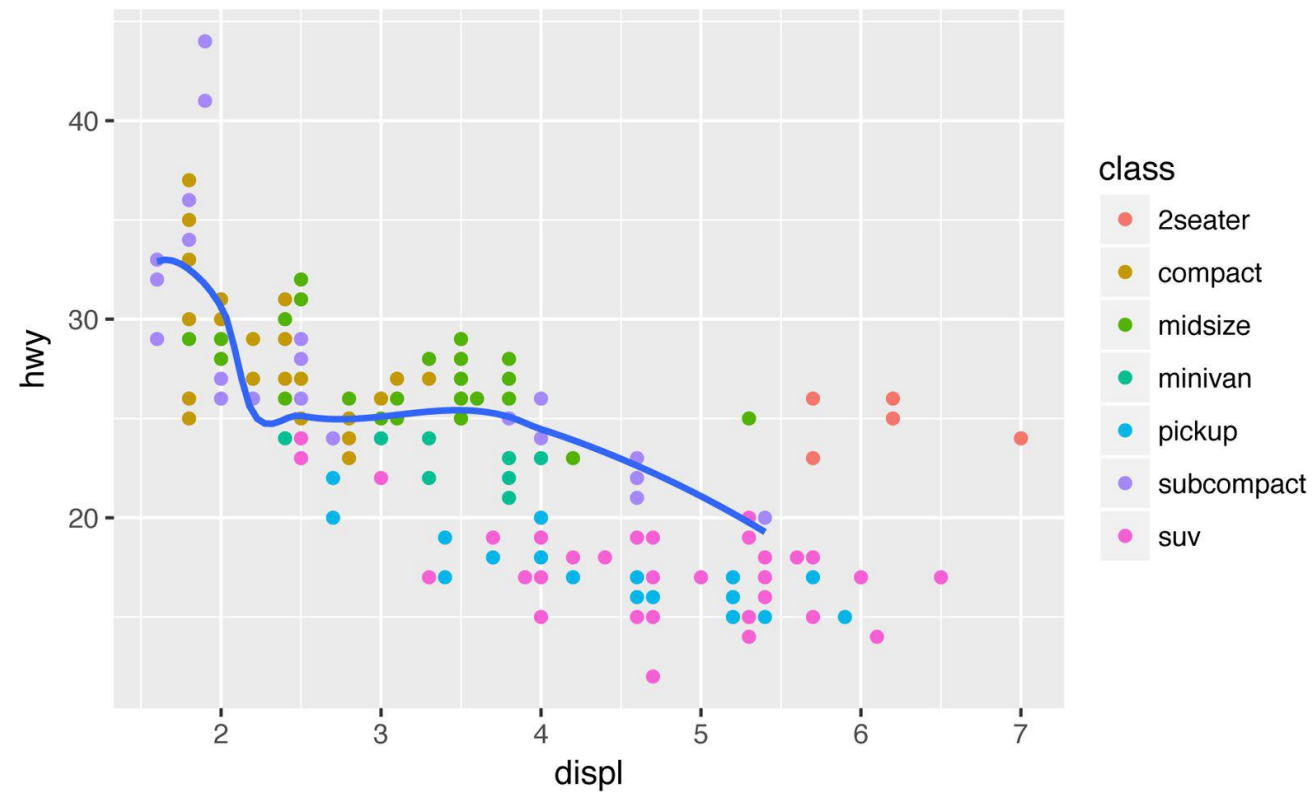


```
> ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +  
  geom_point(mapping = aes(color = class)) +  
  geom_smooth()
```



Objetos Geométricos: geom_*()

```
> ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +  
  geom_point(mapping = aes(color = class)) +  
  geom_smooth(data = filter(mpg, class == "subcompact"), se = FALSE)
```



Transformaciones Estadísticas

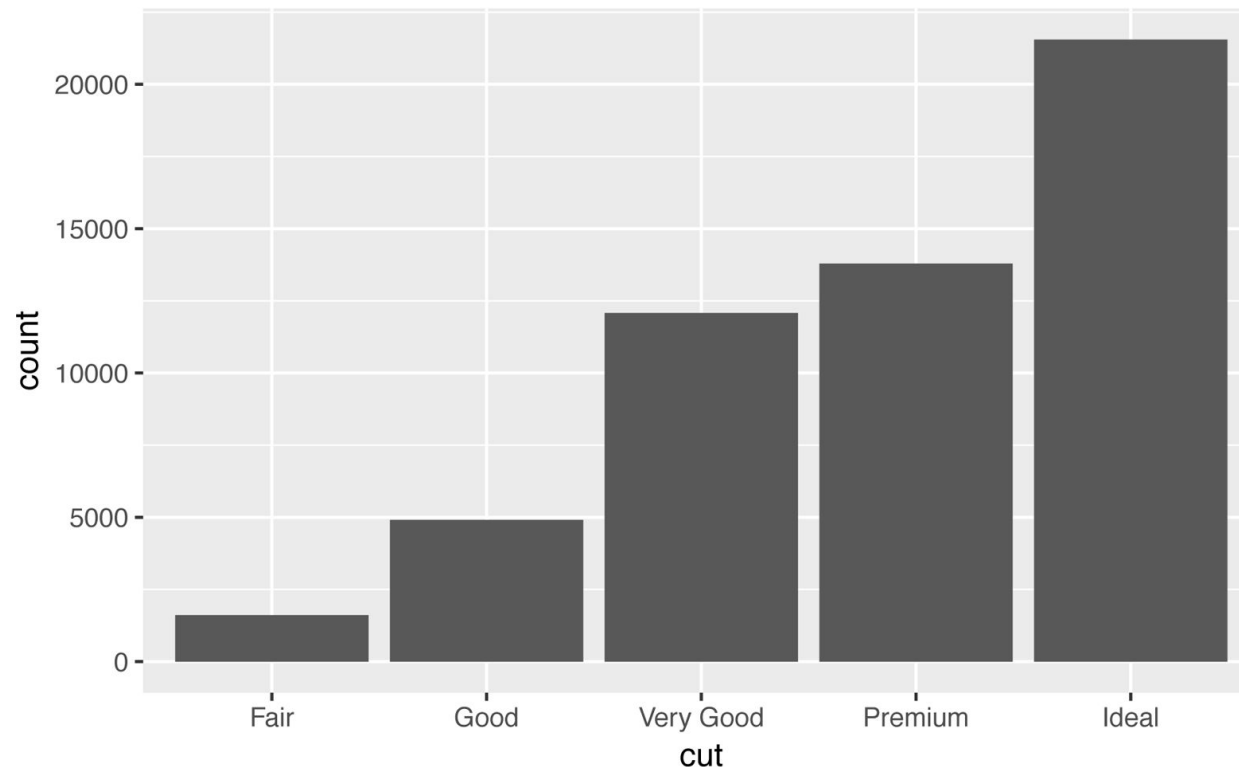
Extraído del Capítulo 1 de: [R for Data Science, Wickham & Grolemund, 2017](#)

Analizando diamonds, con 54000 tuplas

> ?diamonds #no es muy útil visualizar datos de cada diamante.

> ggplot(data = diamonds) +

geom_bar(mapping = aes(x = cut)) # también funciona con **stat_count()**



Cómo funciona geom_bar()

1. `geom_bar()` begins with the **diamonds** data set

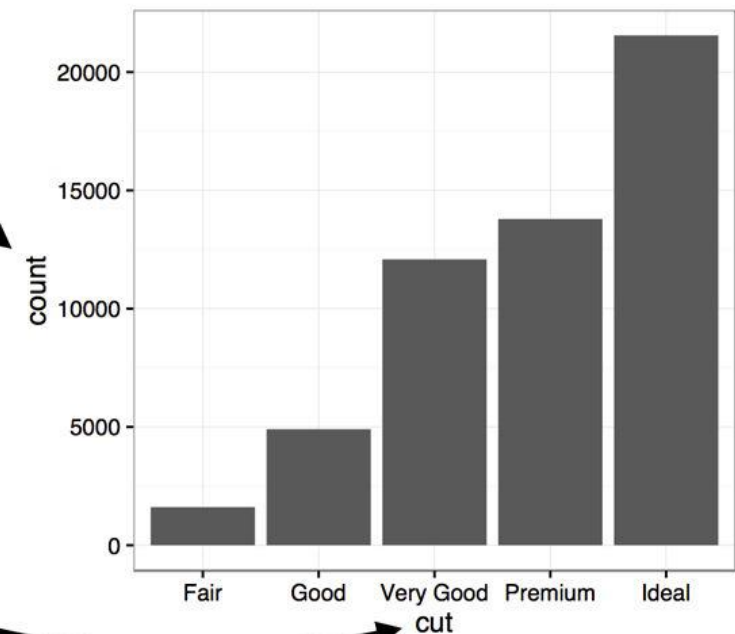
carat	cut	color	clarity	depth	table	price	x	y	z
0.23	Ideal	E	SI2	61.5	55	326	3.95	3.98	2.43
0.21	Premium	E	SI1	59.8	61	326	3.89	3.84	2.31
0.23	Good	E	VS1	56.9	65	327	4.05	4.07	2.31
0.29	Premium	I	VS2	62.4	58	334	4.20	4.23	2.63
0.31	Good	J	SI2	63.3	58	335	4.34	4.35	2.75
...

`stat_count()`

2. `geom_bar()` transforms the data with the "count" stat, which returns a data set of cut values and counts.

cut	count	prop
Fair	1610	1
Good	4906	1
Very Good	12082	1
Premium	13791	1
Ideal	21551	1

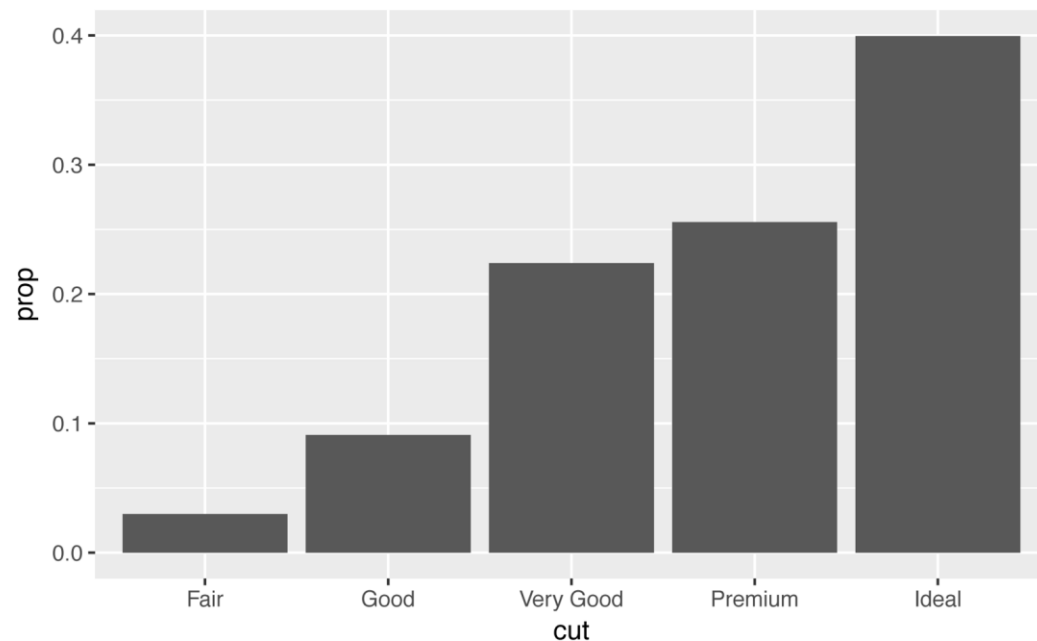
3. `geom_bar()` uses the transformed data to build the plot. cut is mapped to the x axis, count is mapped to the y axis.



Cómo funciona geom_bar()

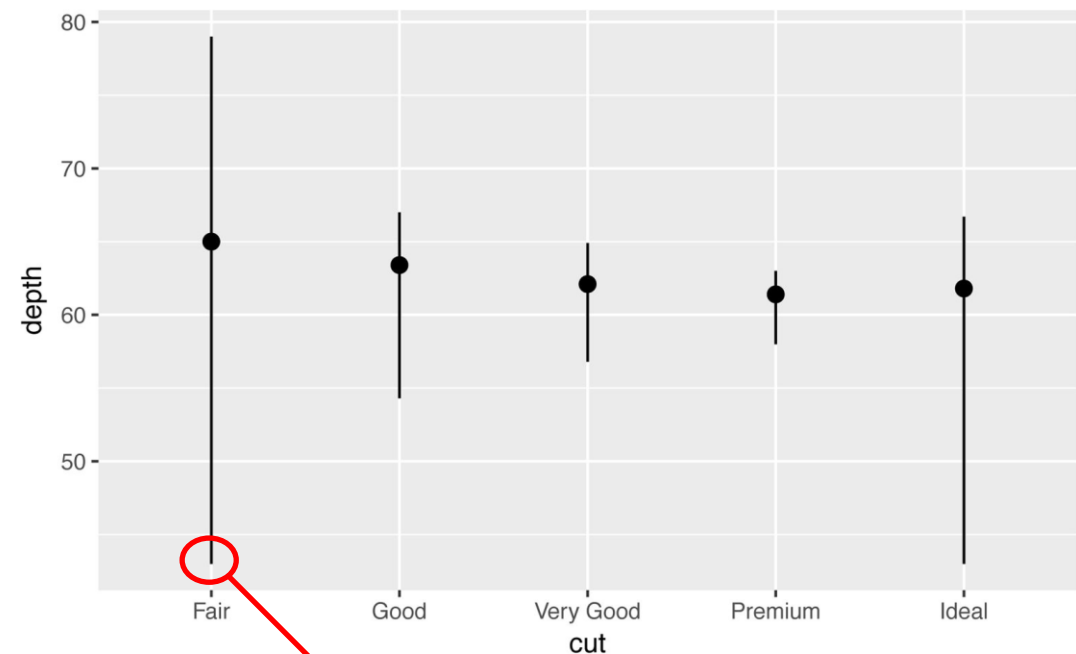
```
> ggplot(data = diamonds) +
```

```
  geom_bar(mapping = aes(x = cut, y = ..prop..,  
    group = 1))
```



```
> ggplot(data = diamonds) +
```

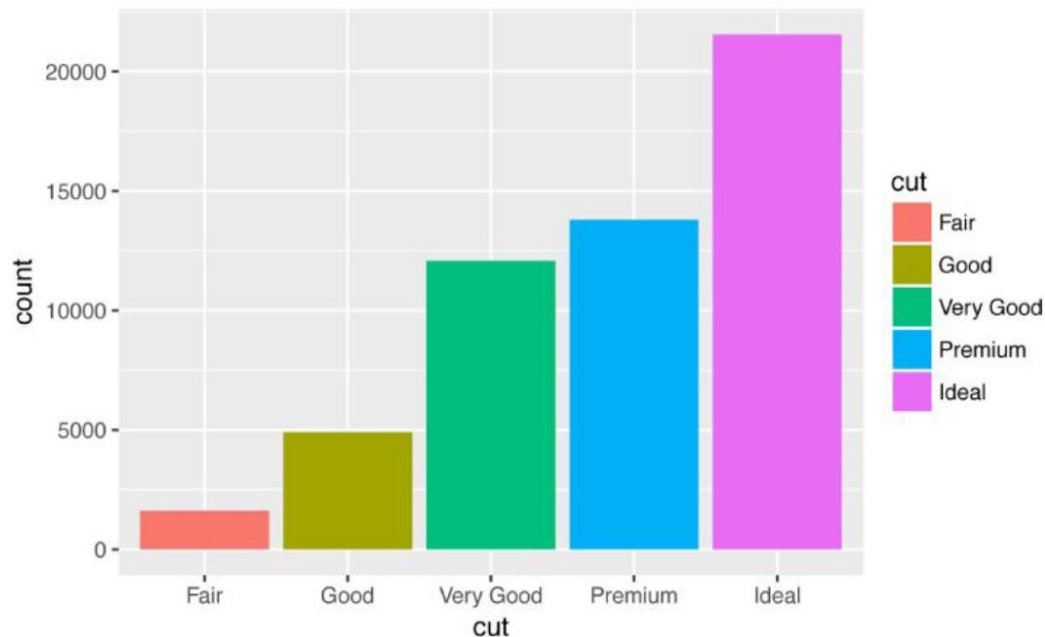
```
  stat_summary(mapping = aes(x = cut, y = depth),  
    fun.ymin = min, fun.ymax = max, fun.y = median)
```



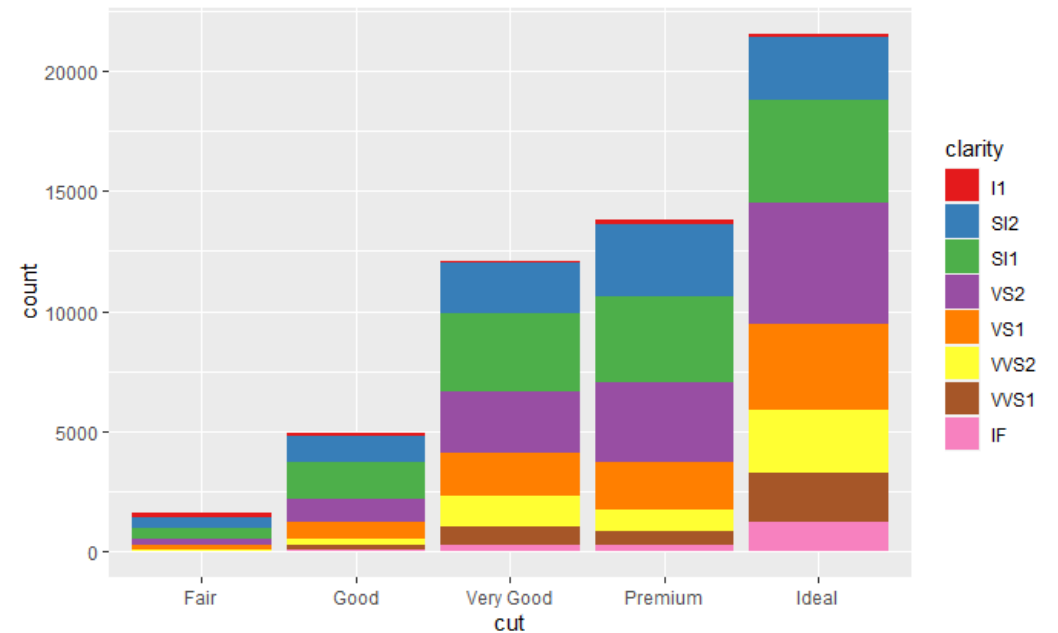
```
diamonds %>% filter(cut=="Fair") %>% select(depth) %>% min()
```

Cómo funciona geom_bar()

```
> ggplot(data = diamonds) +  
  geom_bar(mapping = aes(x = cut, fill = cut))
```



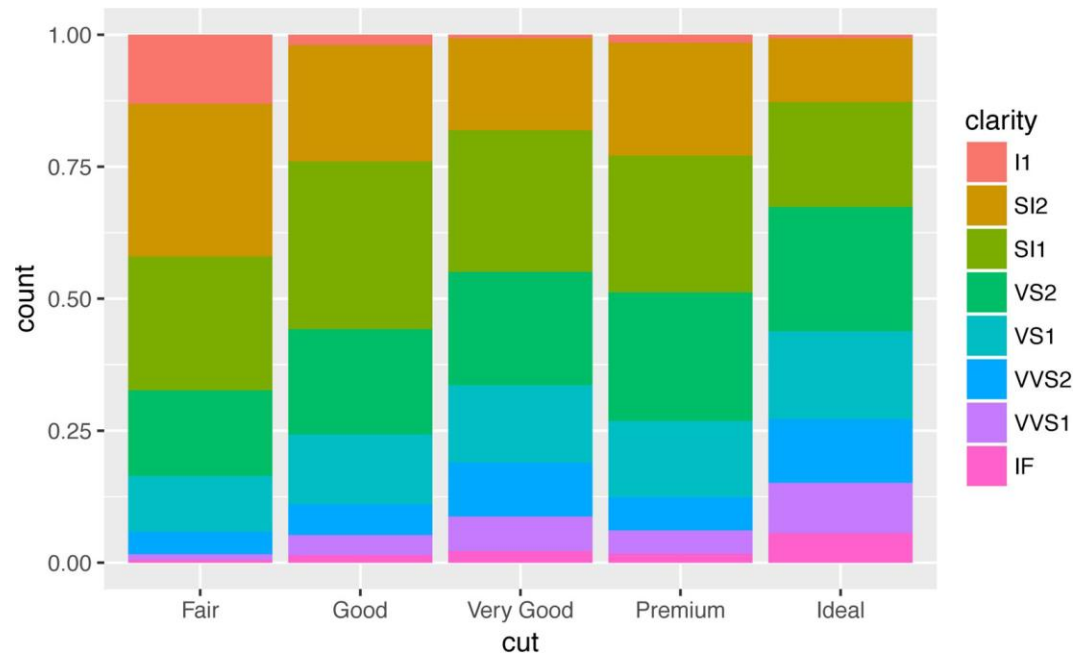
```
> ggplot(data = diamonds) +  
  geom_bar(mapping = aes(x = cut, fill = clarity)) +  
  scale_fill_brewer(palette = "Set1")
```



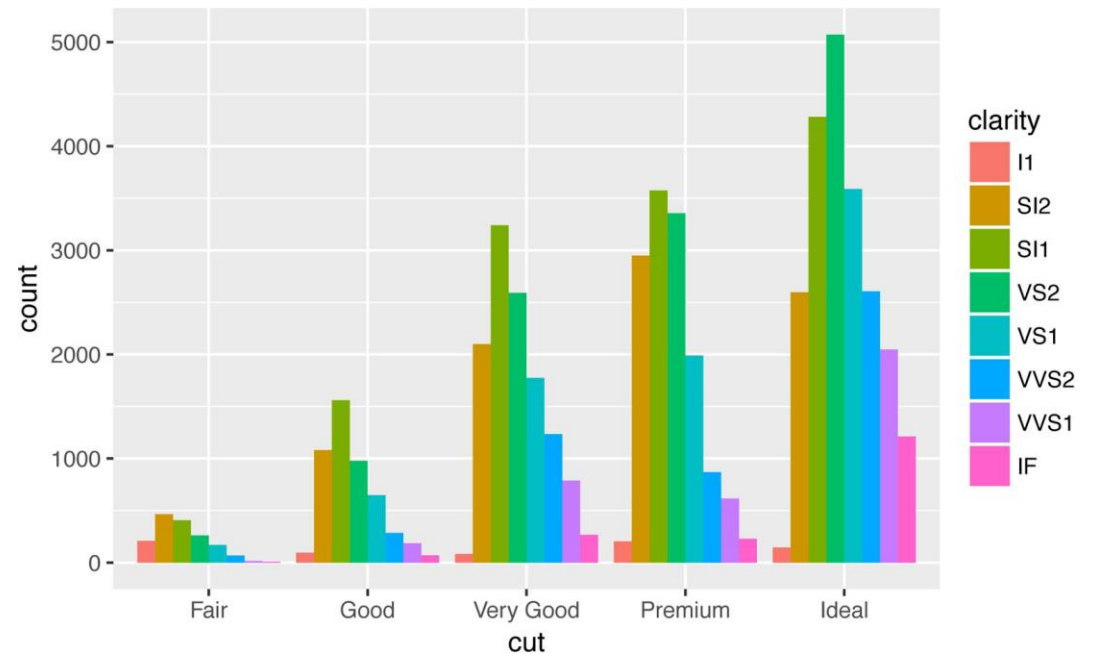
Ojo: si se usa color = cut es otra cosa.

Cómo funciona geom_bar()

```
> ggplot(data = diamonds) +  
  geom_bar(mapping = aes(x = cut, fill = clarity),  
           position = "fill")
```



```
> ggplot(data = diamonds) +  
  geom_bar(mapping = aes(x = cut, fill = clarity),  
           position = "dodge")
```

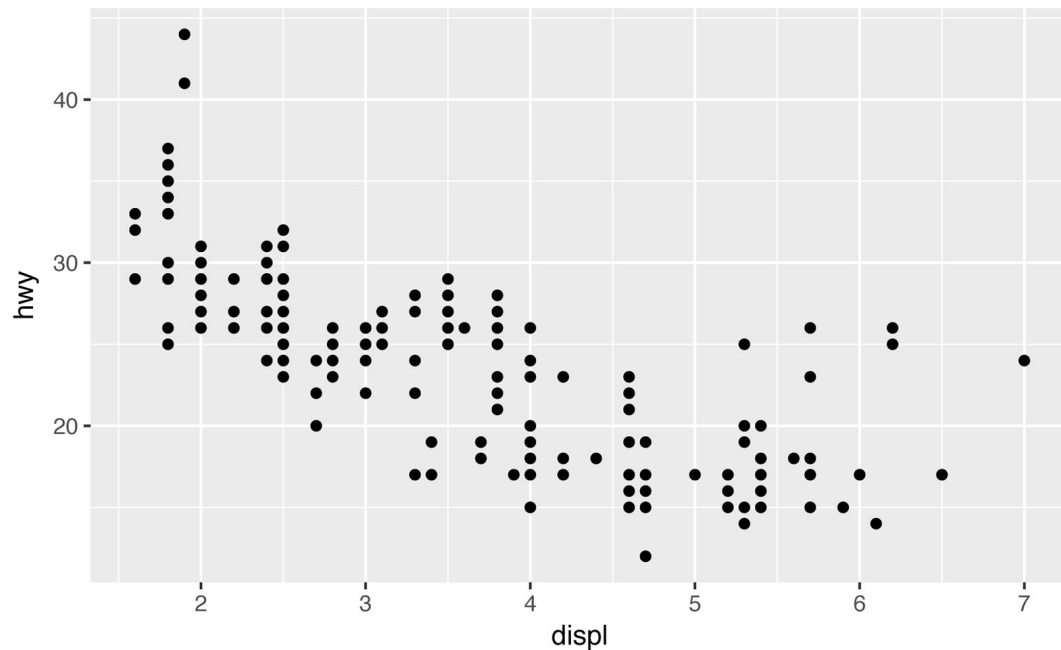


Ajustes: Jittering

```
> ggplot(data = mpg) +
```

```
  geom_point(mapping = aes(x = displ, y = hwy))
```

#Se supone que hay 234 filas, pero no hay 234 puntos.



```
> ggplot(data = mpg) +
```

```
  geom_point(mapping = aes(x = displ, y = hwy),  
                    position = "jitter")
```

