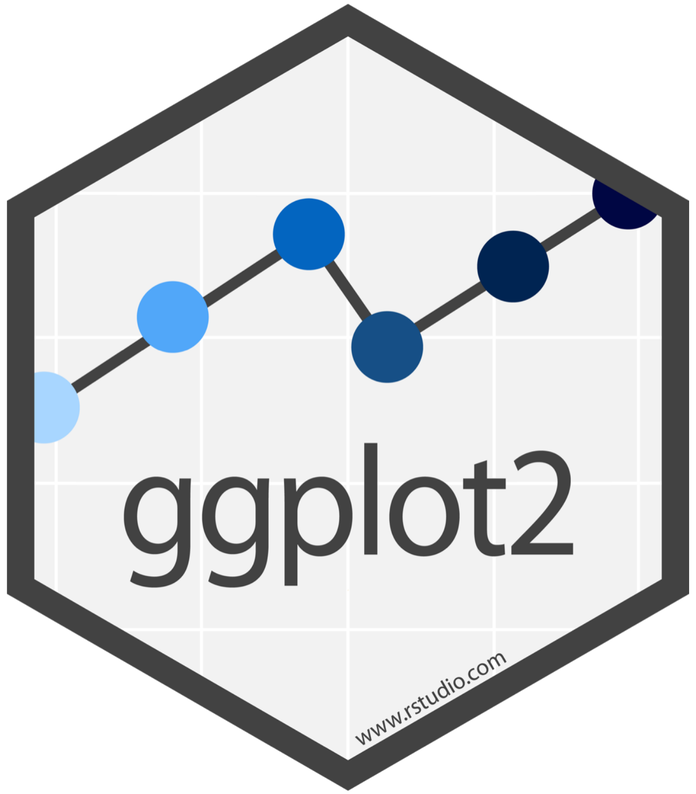
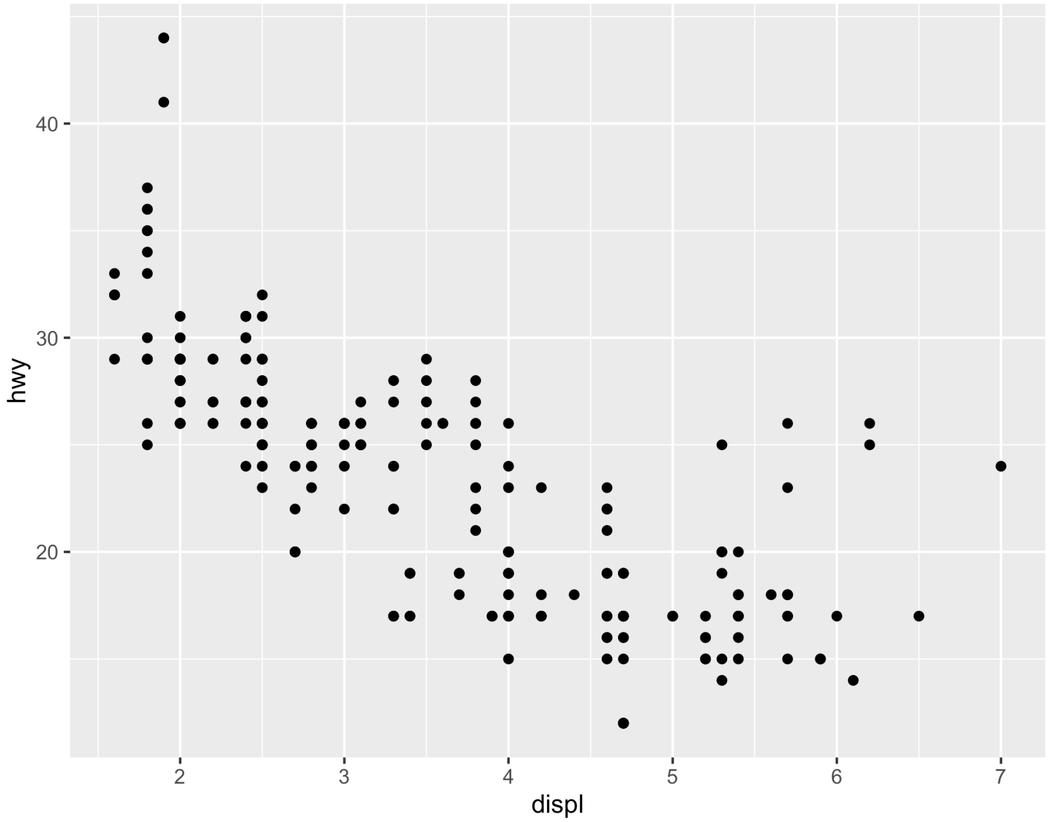
Visualise Data with



Per altro materiale (in inglese): potete andare qui

Quale è più facile capire?



	manufacturer	model	displ	year	cyl	trans	drv	cty	hwy	fl	class
16	audi	a6 quattro	2.8	1999	6	auto(l5)	4	15	24	p	midsize
17	audi	a6 quattro	3.1	2008	6	auto(s6)	4	17	25	p	midsize
18	audi	a6 quattro	4.2	2008	8	auto(s6)	4	16	23	p	midsize
19	chevrolet	c1500 suburban 2wd	5.3	2008	8	auto(l4)	r	14	20	r	suv
20	chevrolet	c1500 suburban 2wd	5.3	2008	8	auto(l4)	r	11	15	e	suv
21	chevrolet	c1500 suburban 2wd	5.3	2008	8	auto(l4)	r	14	20	r	suv
22	chevrolet	c1500 suburban 2wd	5.7	1999	8	auto(l4)	r	13	17	r	suv
23	chevrolet	c1500 suburban 2wd	6.0	2008	8	auto(l4)	r	12	17	r	suv
24	chevrolet	corvette	5.7	1999	8	manual(m6)	r	16	26	p	2seater
25	chevrolet	corvette	5.7	1999	8	auto(l4)	r	15	23	p	2seater
26	chevrolet	corvette	6.2	2008	8	manual(m6)	r	16	26	p	2seater
27	chevrolet	corvette	6.2	2008	8	auto(s6)	r	15	25	p	2seater
28	chevrolet	corvette	7.0	2008	8	manual(m6)	r	15	24	p	2seater
29	chevrolet	k1500 tahoe 4wd	5.3	2008	8	auto(l4)	4	14	19	r	suv
30	chevrolet	k1500 tahoe 4wd	5.3	2008	8	auto(l4)	4	11	14	e	suv
31	chevrolet	k1500 tahoe 4wd	5.7	1999	8	auto(l4)	4	11	15	r	suv
32	chevrolet	k1500 tahoe 4wd	6.5	1999	8	auto(l4)	4	14	17	d	suv
33	chevrolet	malibu	2.4	1999	4	auto(l4)	f	19	27	r	midsize
34	chevrolet	malibu	2.4	2008	4	auto(l4)	f	22	30	r	midsize
35	chevrolet	malibu	3.1	1999	6	auto(l4)	f	18	26	r	midsize
36	chevrolet	malibu	3.5	2008	6	auto(l4)	f	18	29	r	midsize
37	chevrolet	malibu	3.6	2008	6	auto(s6)	f	17	26	r	midsize
38	dodge	caravan 2wd	2.4	1999	4	auto(l3)	f	18	24	r	minivan
39	dodge	caravan 2wd	3.0	1999	6	auto(l4)	f	17	24	r	minivan

Il dataset

Dati di risparmio di carburante per 38 modelli di auto.

```
mpg
```

Quiz

In gruppi discutete:

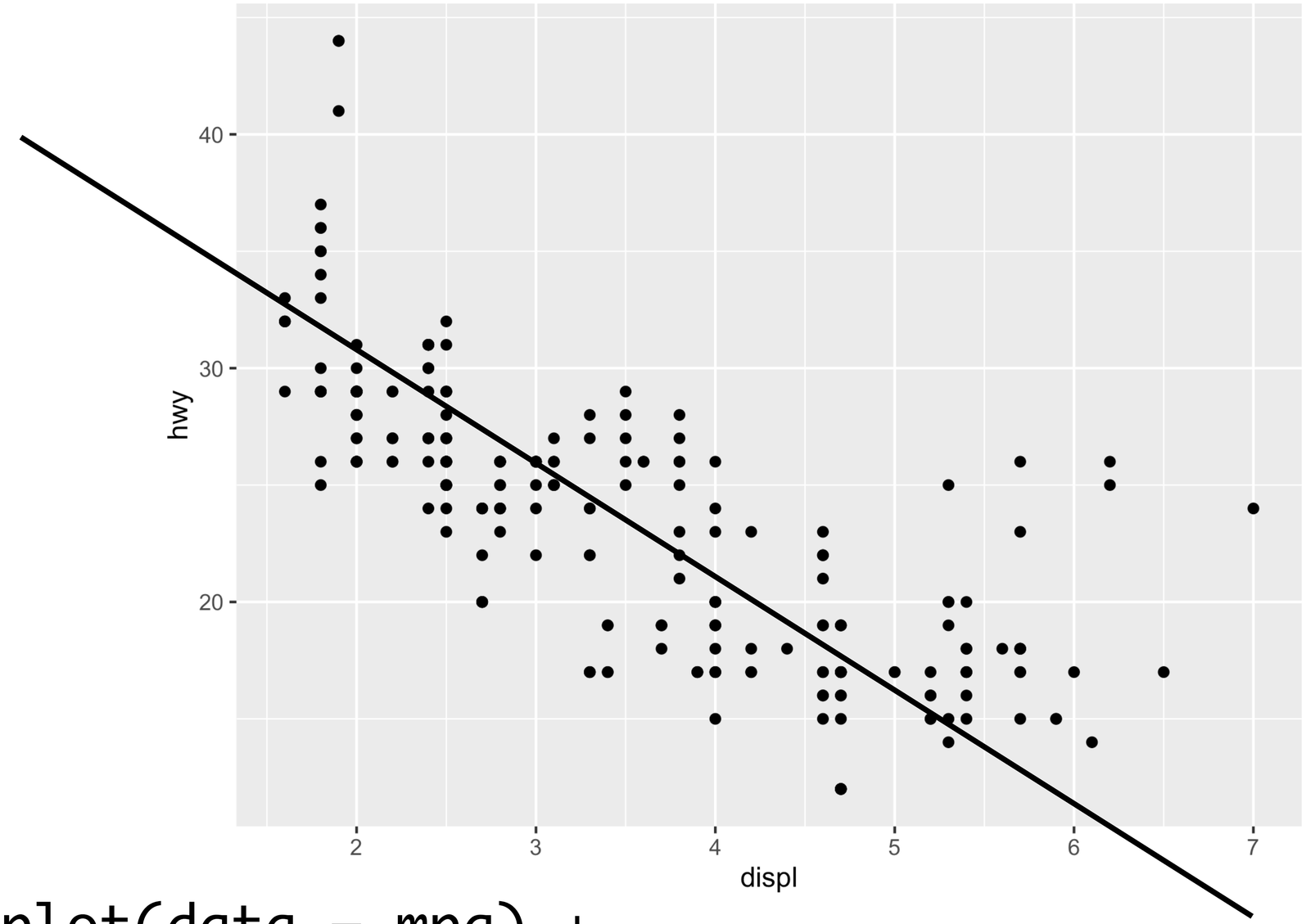
Quale relazione vi aspettate tra le dimensioni del motore(nome variable: displ) e l'efficenza del motore (km per litro, nome variable: hwy)?

Tocca a te !

Copiate e “run” the code

Quando si scrive un codice (in qualsiasi linguaggio di programmazione) è importantissimo stare attenti agli errori di ortografia e punteggiatura!

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




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

1. Inizia il grafico (plot) con ggplot()
2. Aggiungi i diversi strati (layers) con geom_ functions

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

Suggerimento: Sempre aggiungere “+” alla fine della riga mai all’inizio

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

data

+ prima della nuova riga

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

Tipologia

aes()

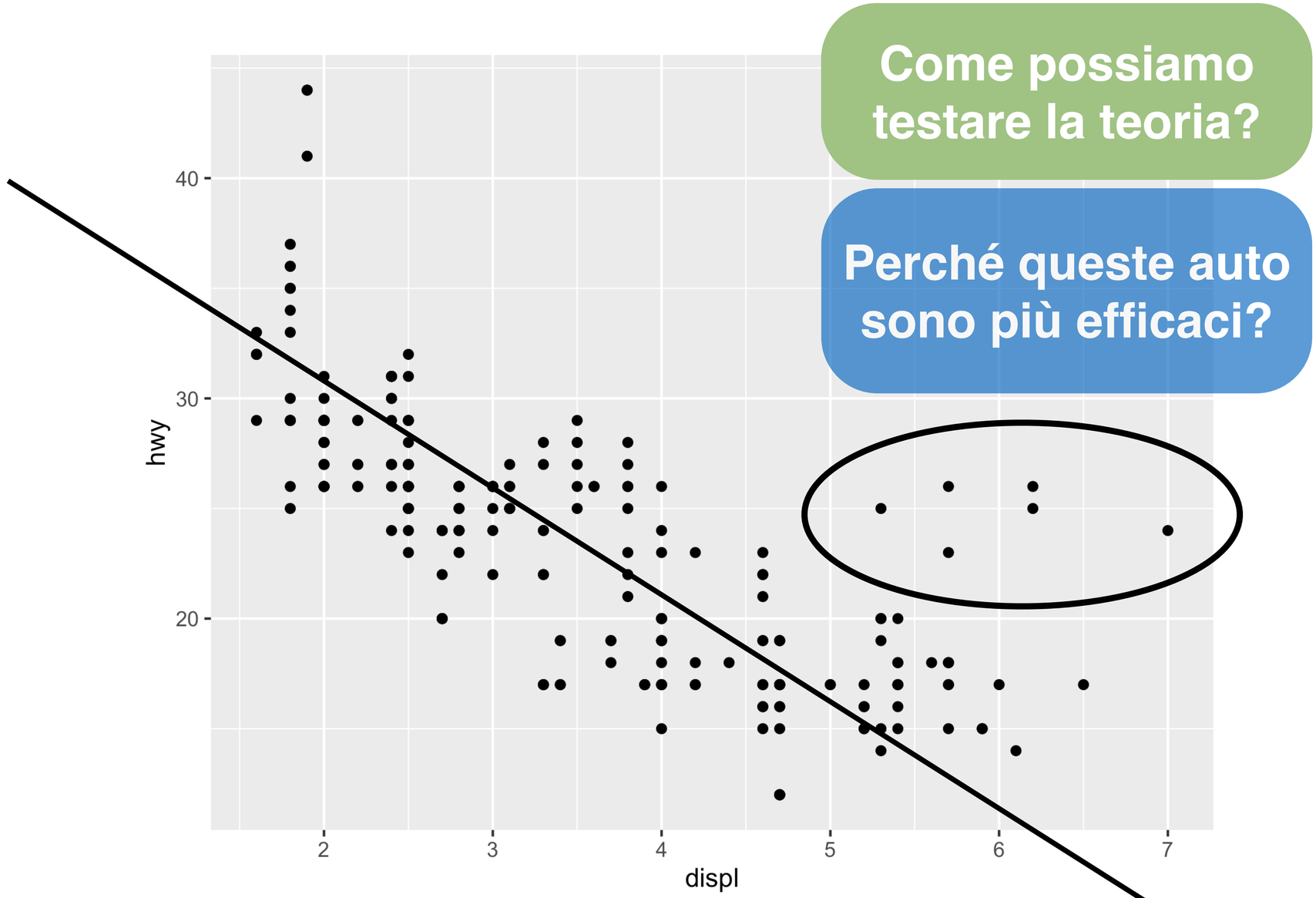
var x

var y

Mappings

"The greatest value of a picture is when it forces us to notice what we never expected to see."

- John Tukey




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

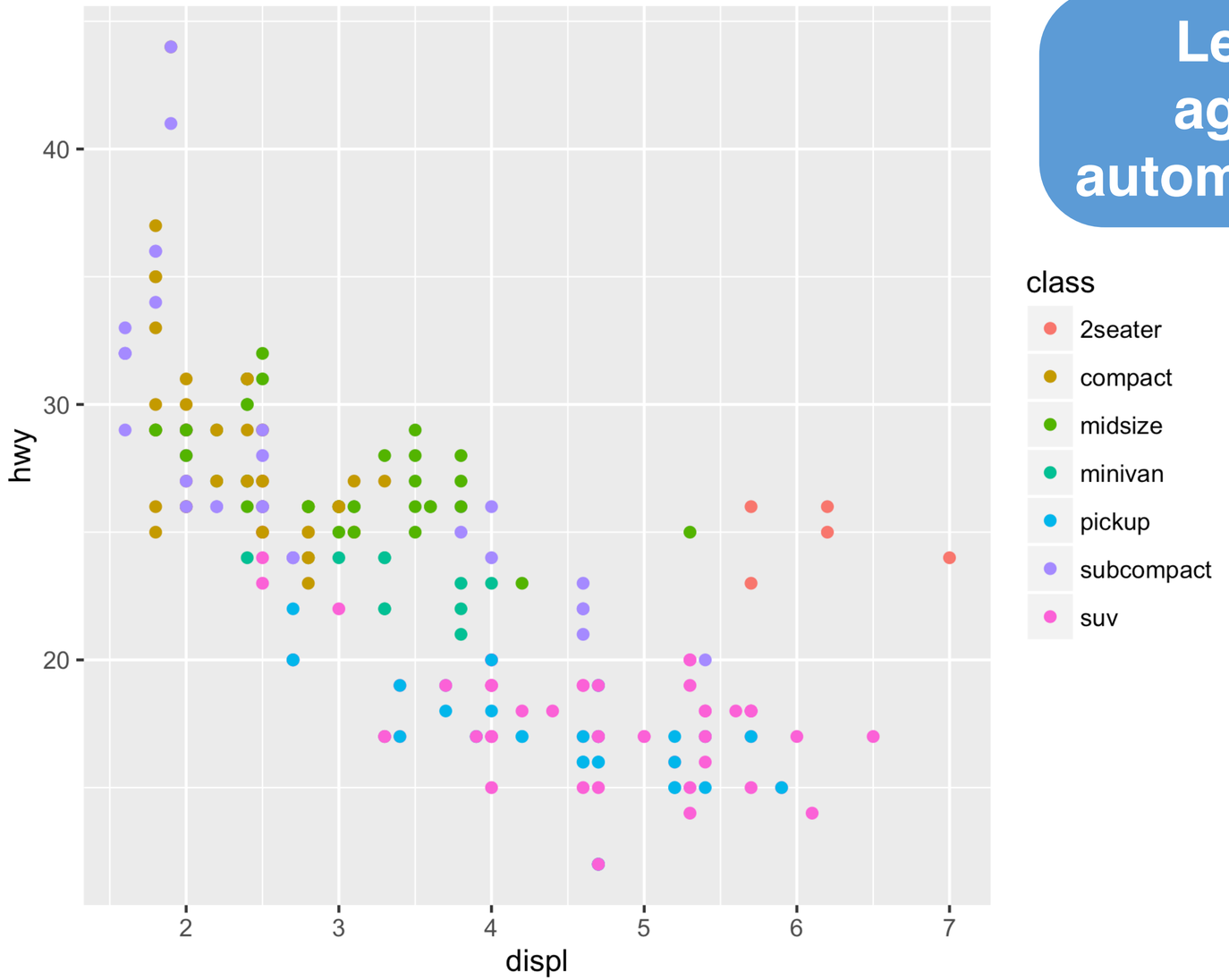

Spazio Visivo
Spazio Dati
colore
classi
Rosso
Marrone
Verde
Acqua
Blu
Viola
Rosa
2seater
compact
midsize
minivan
pickup
subcompact
suv

Aesthetics

proprietà estetiche

Variabile per

```
ggplot(mpg) + geom_point(aes(x = displ, y = hwy, color = class))
ggplot(mpg) + geom_point(aes(x = displ, y = hwy, size = class))
ggplot(mpg) + geom_point(aes(x = displ, y = hwy, shape = class))
ggplot(mpg) + geom_point(aes(x = displ, y = hwy, alpha = class))
```



Legenda aggiunta automaticament

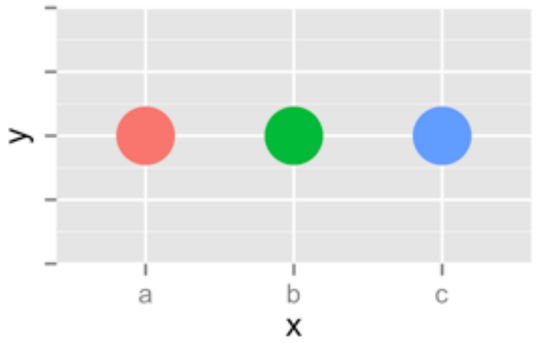
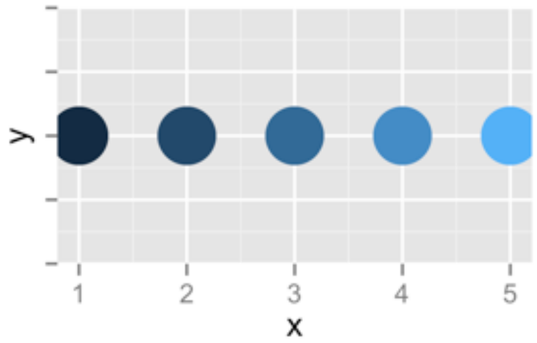
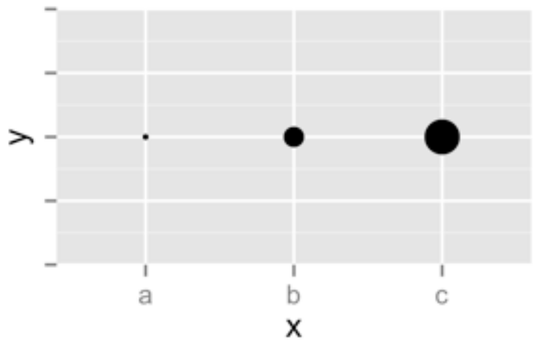
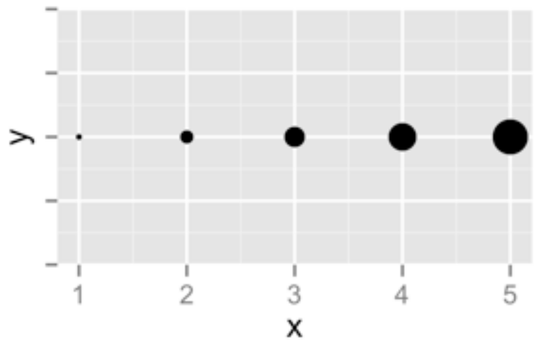
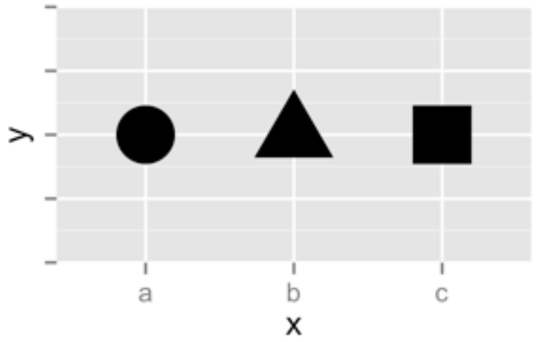
```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, color = class))
```

Tocca a te II

Ora aggiungete il colore (color), dimensioni (size), alpha, elementi estetici (aesthetics). Provate a disegnare diversi grafici

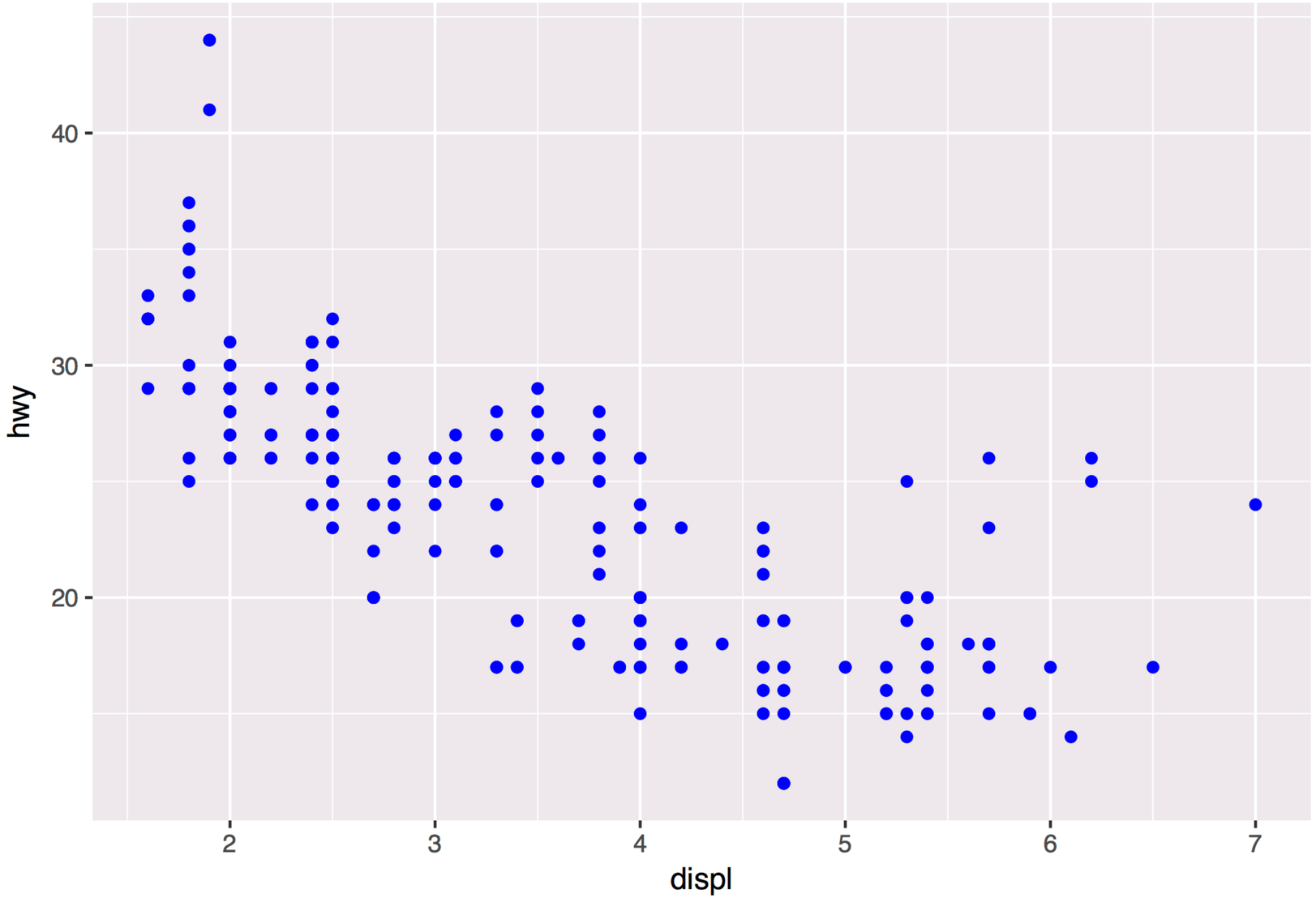
Cosa succede se provate a mappare “aesthetics” con variabili discrete e continue?

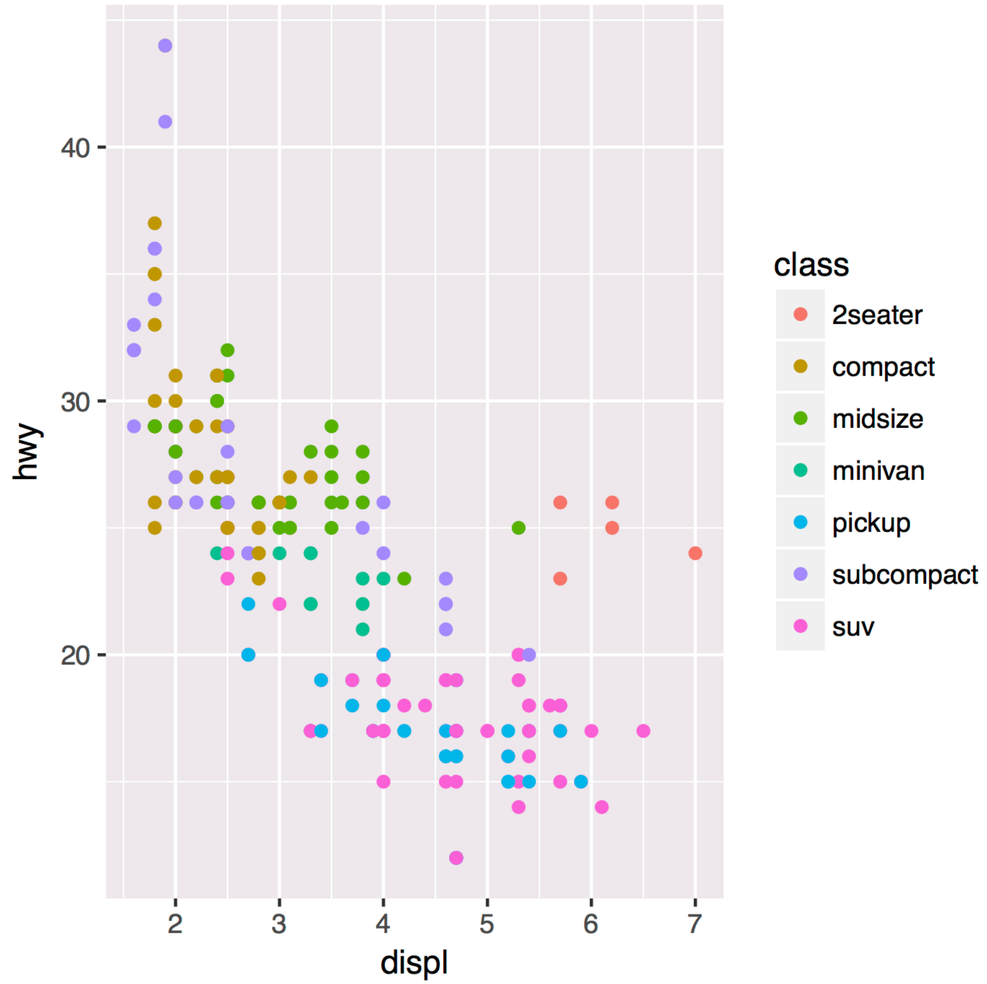
Cosa succede se usate più di un elemento “aesthetics”?

	Discrete	Continuous
Colore		
Dimensione (size)		
Forma (Shape)		

set vs. map

Come si può disegnare questo grafico?

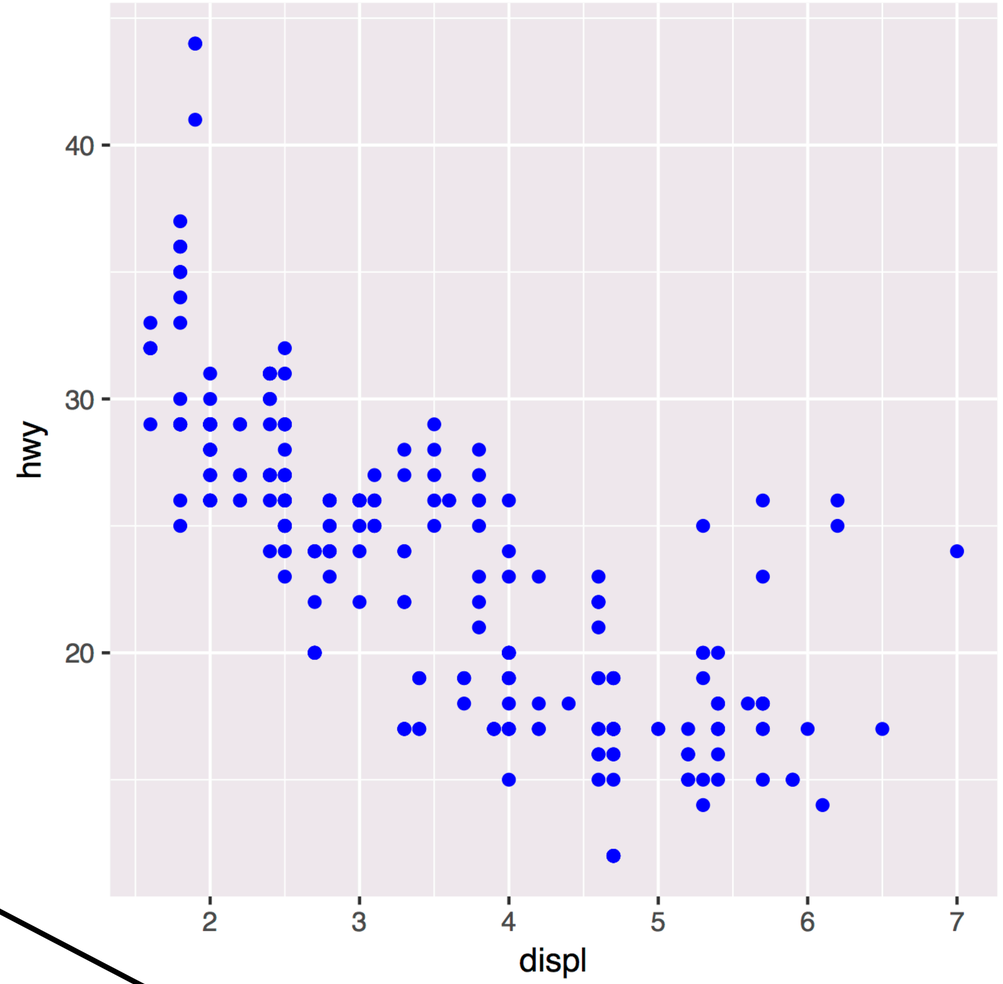




Dentro aes(): l'elemento estetico mappa una variable

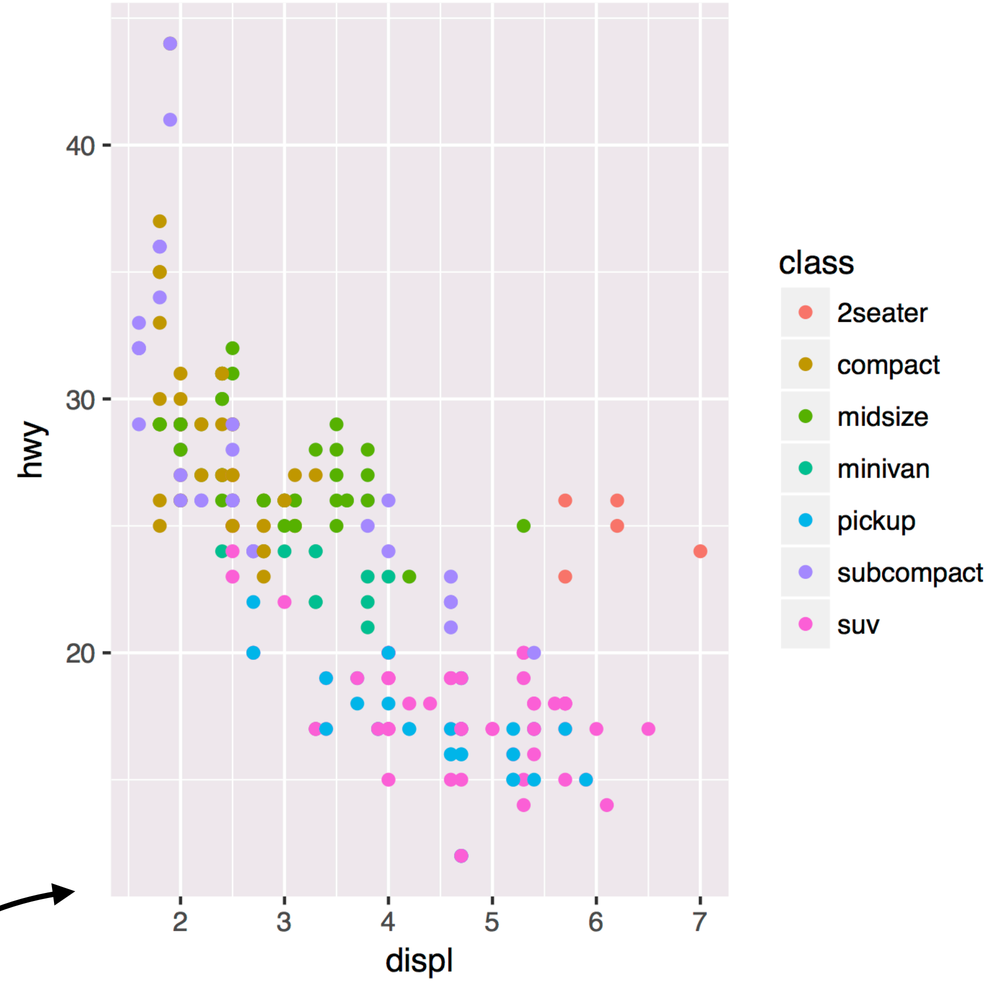
```
ggplot(mpg) + geom_point(aes(x = displ, y = hwy, color = class))
```

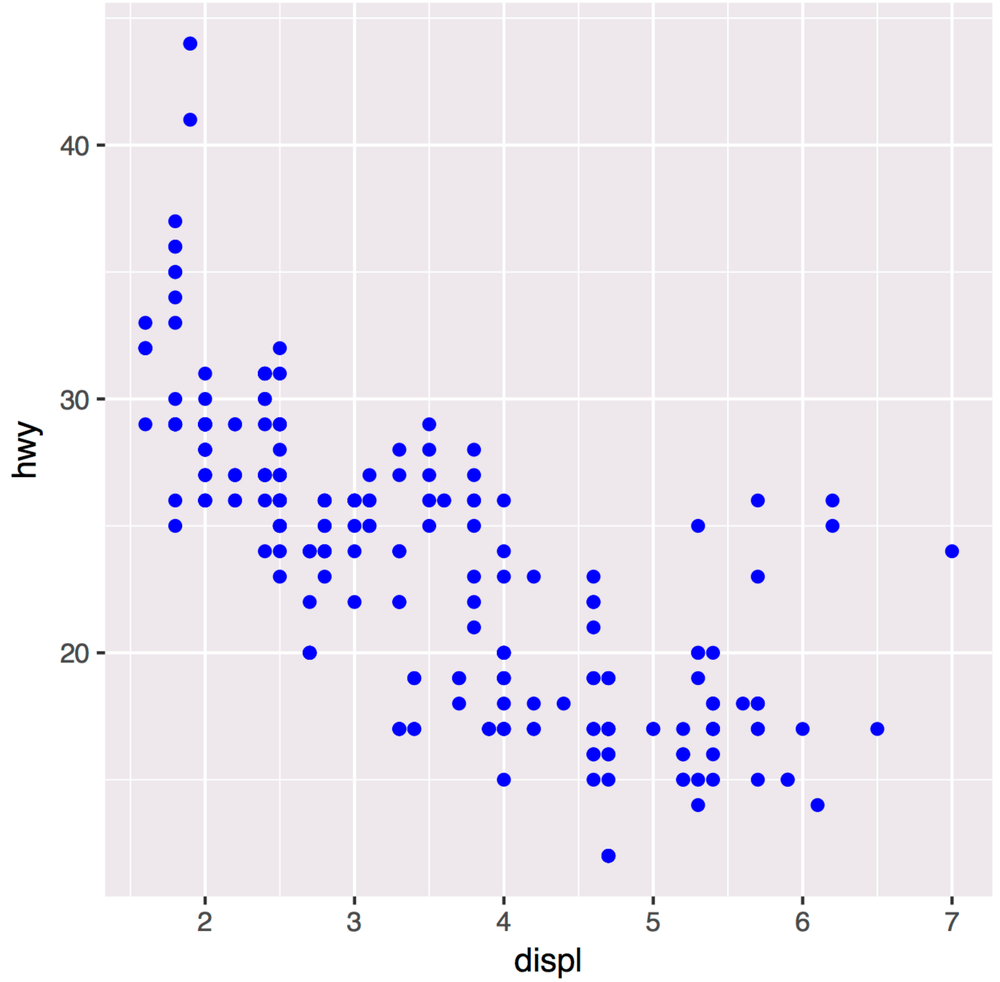
Outside of aes(): imposta l'elemento estetico su un valore




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

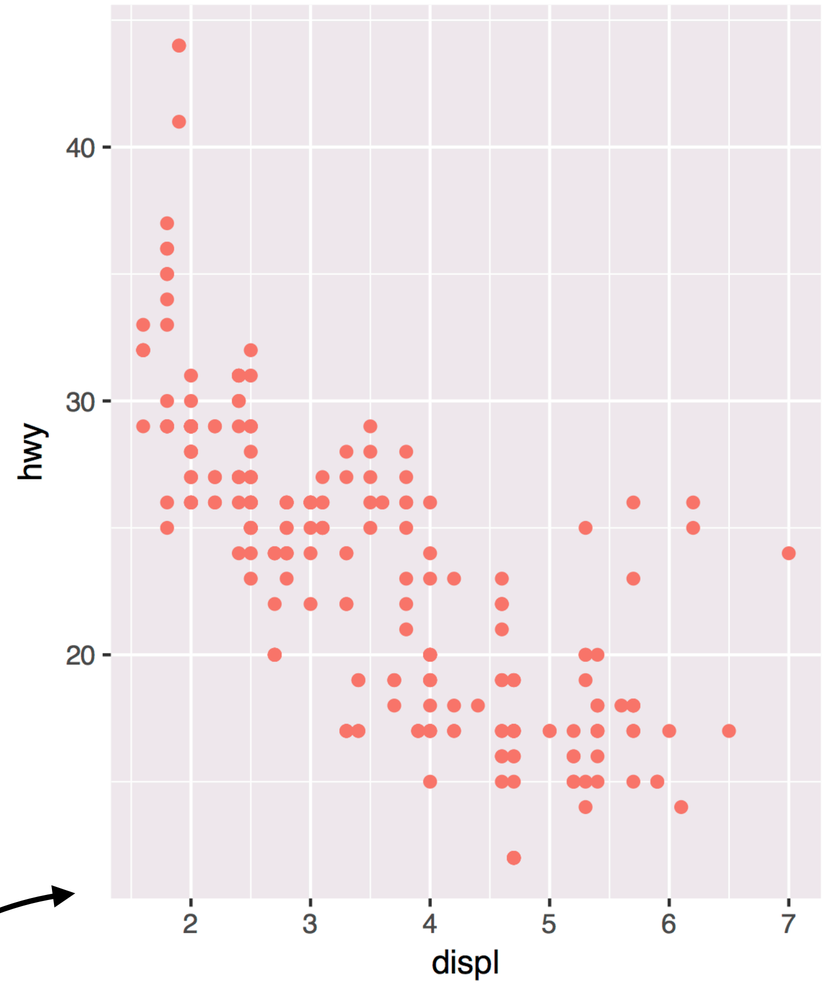
```
ggplot(mpg) + geom_point(aes(x = displ, y = hwy), color = "blue")
```

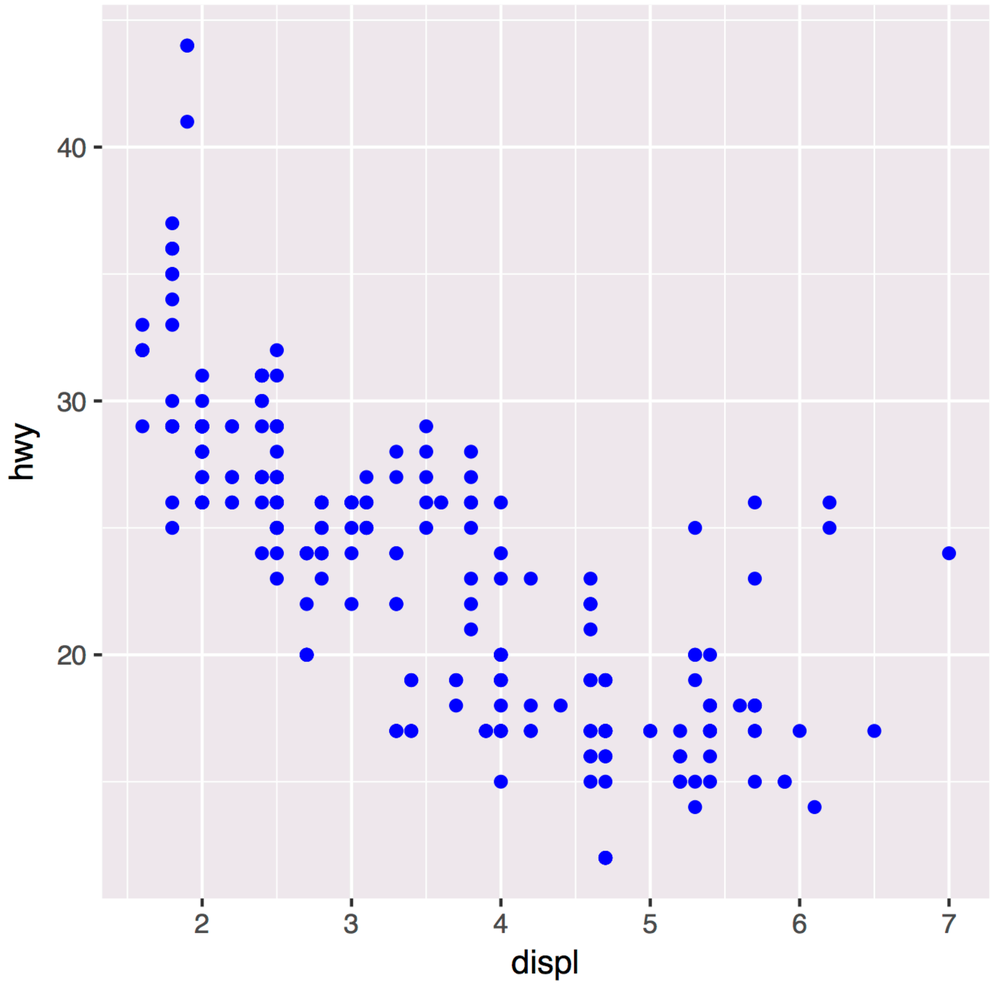





```
ggplot(mpg) + geom_point(aes(x = displ, y = hwy, color = class))
```

```
ggplot(mpg) + geom_point(aes(x = displ, y = hwy), color = "blue")
```





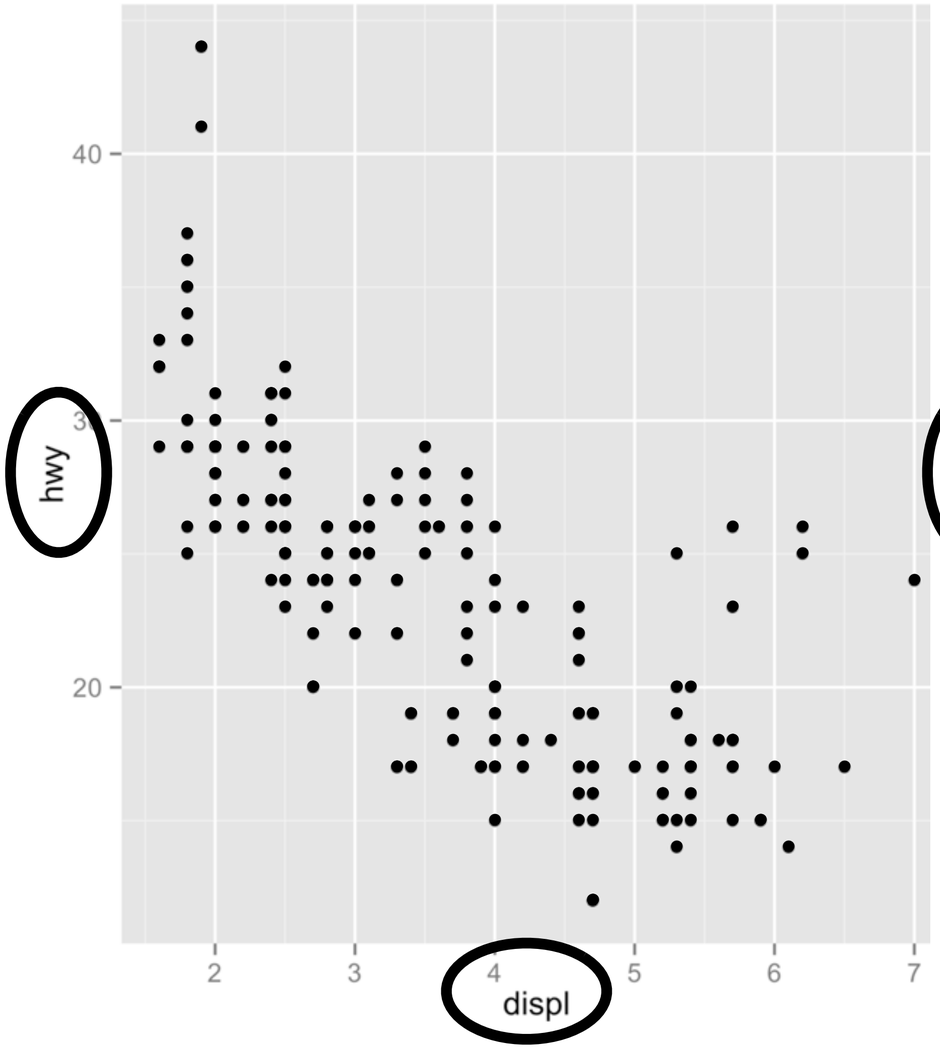

```
ggplot(mpg) + geom_point(aes(x = displ, y = hwy, color = "blue"))
```

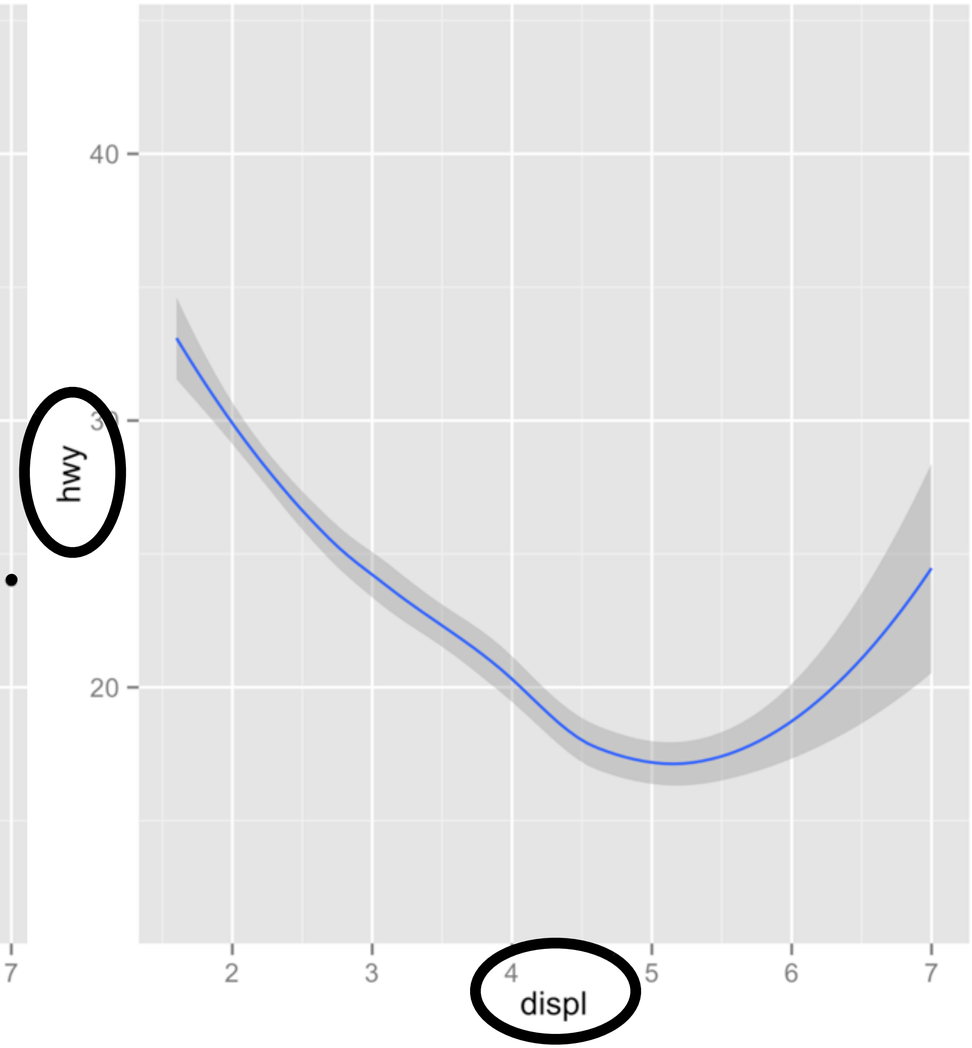
```
ggplot(mpg) + geom_point(aes(x = displ, y = hwy), color = "blue")
```

Geoms

Come è possibile che questi grafici siano

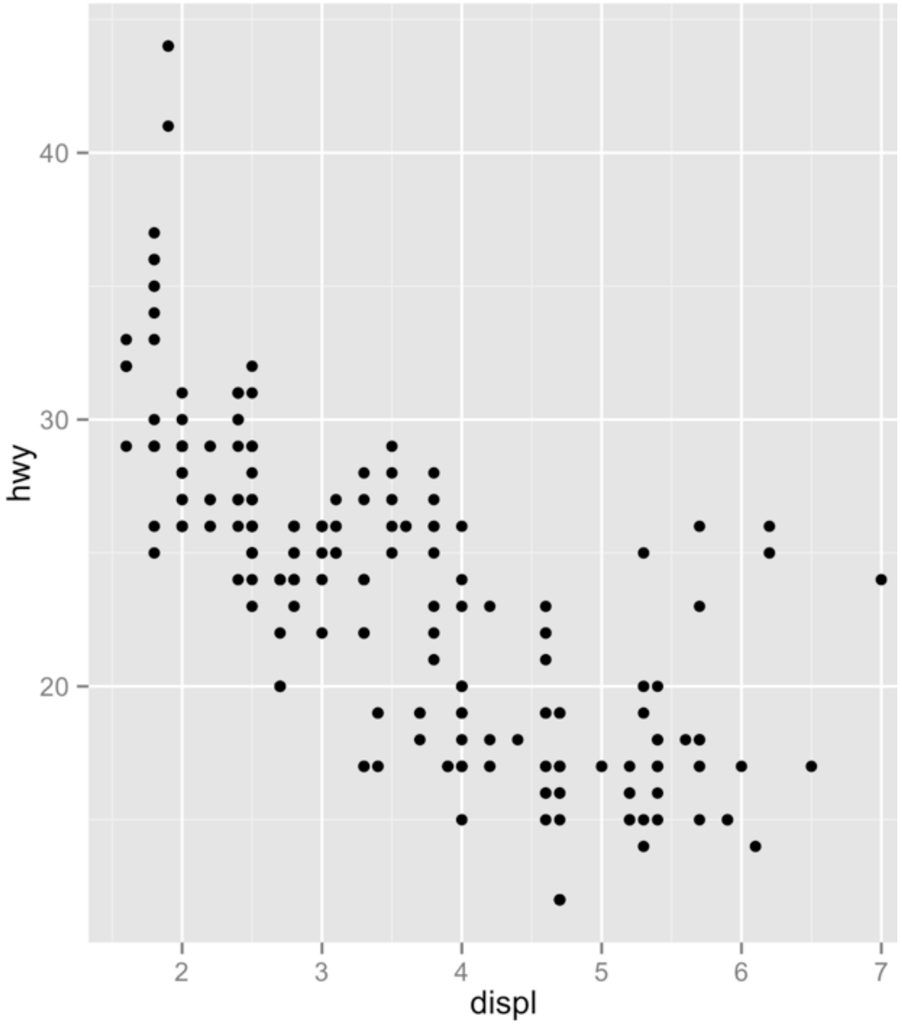
Stesso: x, y, data

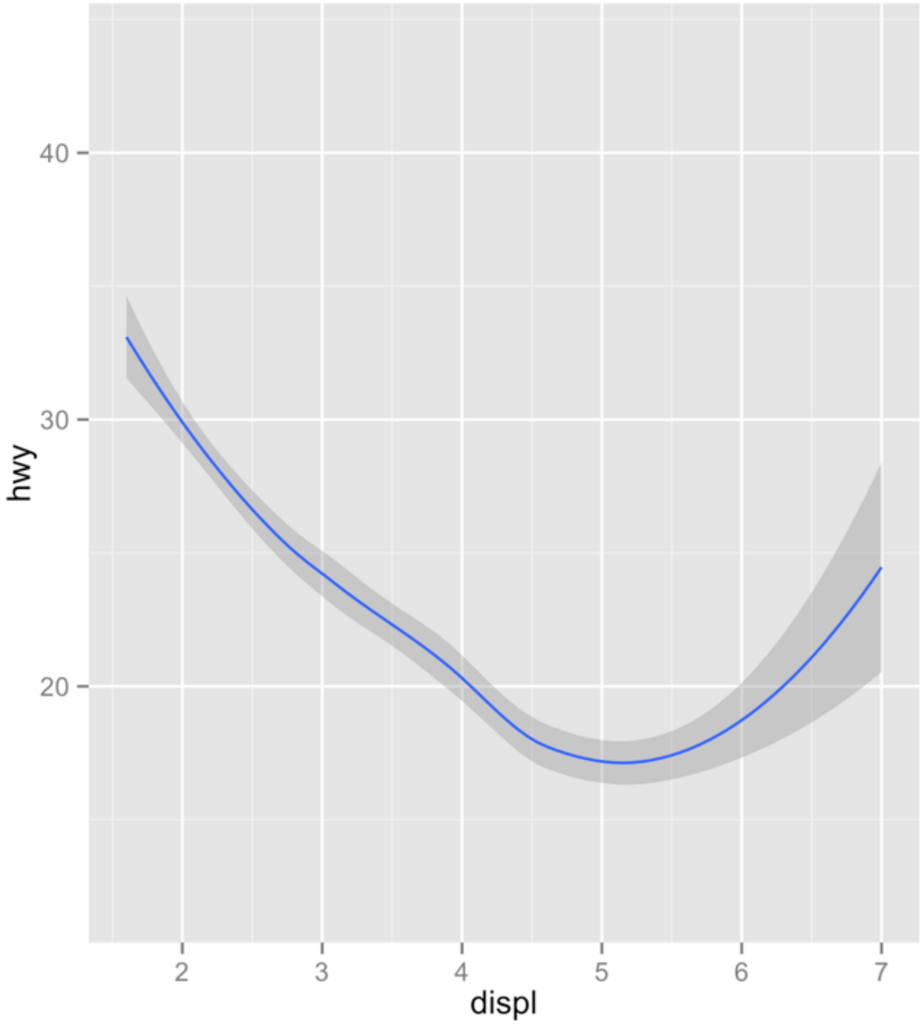




Cosa differenzia questi grafici?

Differenze: elementi geometrici (geom),
E.s. l'oggetto visivo utilizzato per rappresentare i dati



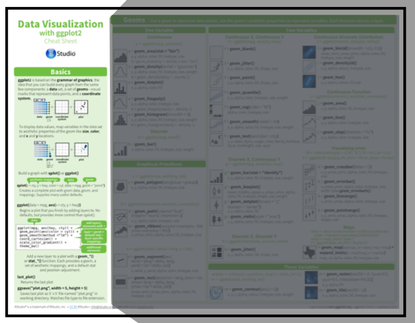


geoms

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

geom_ functions

Ognuno richiede un argomento di mappatura.



Geoms - Use a geom to represent data points, use the geom's aesthetic properties to represent variables. Each function returns a layer.

One Variable

Continuous

```
a <- ggplot(mpg, aes(hwy))
```

a + geom_area(stat = "bin")
x, y, alpha, color, fill, linetype, size
b + geom_area(aes(y = ..density..), stat = "bin")

a + geom_density(kernel = "gaussian"**)**
x, y, alpha, color, fill, linetype, size, weight
b + geom_density(aes(y = ..county..))

a + geom_dotplot()
x, y, alpha, color, fill

a + geom_freqpoly()
x, y, alpha, color, linetype, size
b + geom_freqpoly(aes(y = ..density..))

a + geom_histogram(binwidth = 5**)**
x, y, alpha, color, fill, linetype, size, weight
b + geom_histogram(aes(y = ..density..))

Discrete

```
b <- ggplot(mpg, aes(fl))
```

b + geom_bar()
x, alpha, color, fill, linetype, size, weight

Graphical Primitives

```
map <- map_data("state")
c <- ggplot(map, aes(long, lat))
```

c + geom_polygon(aes(group = group)**)**
x, y, alpha, color, fill, linetype, size

```
d <- ggplot(economics, aes(date, unemploy))
```

d + geom_path(lineend="butt", linejoin="round', linemitre=1**)**
x, y, alpha, color, linetype, size

d + geom_ribbon(aes(ymin=unemploy - 900, ymax=unemploy + 900)**)**
x, ymax, ymin, alpha, color, fill, linetype, size

```
e <- ggplot(seals, aes(x = long, y = lat))
```

e + geom_segment(aes(
xend = long + delta_long,
yend = lat + delta_lat)**)**
x, xend, y, yend, alpha, color, linetype, size

e + geom_rect(aes(xmin = long, ymin = lat, xmax= long + delta_long, ymax = lat + delta_lat)**)**
xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size

Two Variables

Continuous X, Continuous Y

```
f <- ggplot(mpg, aes(cty, hwy))
```

f + geom_blank()
(Useful for expanding limits)

f + geom_jitter()
x, y, alpha, color, fill, shape, size

f + geom_point()
x, y, alpha, color, fill, shape, size

f + geom_quantile()
x, y, alpha, color, linetype, size, weight

f + geom_rug(sides = "bl"**)**
alpha, color, linetype, size

f + geom_smooth(method = lm**)**
x, y, alpha, color, fill, linetype, size, weight

f + geom_text(aes(label = cty)**)**
x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

Continuous Bivariate Distribution

```
i <- ggplot(movies, aes(year, rating))
```

i + geom_bin2d(binwidth = c(5, 0.5)**)**
xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size, weight

i + geom_density2d()
x, y, alpha, colour, linetype, size

i + geom_hex()
x, y, alpha, colour, fill size

Continuous Function

```
j <- ggplot(economics, aes(date, unemploy))
```

j + geom_area()
x, y, alpha, color, fill, linetype, size

j + geom_line()
x, y, alpha, color, linetype, size

j + geom_step(direction = "hv"**)**
x, y, alpha, color, linetype, size

Visualizing error

```
df <- data.frame(grp = c("A", "B"), fit = 4:5, se = 1:2)
k <- ggplot(df, aes(grp, fit, ymin = fit-se, ymax = fit+se))
```

k + geom_crossbar(fatten = 2**)**
x, y, ymax, ymin, alpha, color, fill, linetype, size

k + geom_errorbar()
x, ymax, ymin, alpha, color, linetype, size, width (also **geom_errorbarh()**)

k + geom_linerange()
x, ymin, ymax, alpha, color, linetype, size

k + geom_pointrange()
x, y, ymin, ymax, alpha, color, fill, linetype, shape, size

Discrete X, Continuous Y

```
g <- ggplot(mpg, aes(class, hwy))
```

g + geom_bar(stat = "identity")
x, y, alpha, color, fill, linetype, size, weight

g + geom_boxplot()
lower, middle, upper, x, ymax, ymin, alpha, color, fill, linetype, shape, size, weight

g + geom_dotplot(binaxis = "y", stackdir = "center"**)**
x, y, alpha, color, fill

g + geom_violin(scale = "area"**)**
x, y, alpha, color, fill, linetype, size, weight

Discrete X, Discrete Y

```
h <- ggplot(diamonds, aes(cut, color))
```

h + geom_jitter()
x, y, alpha, color, fill, shape, size

Maps

```
data <- data.frame(murder = USArrests$Murder,
state = tolower(rownames(USArrests)))
map <- map_data("state")
l <- ggplot(data, aes(fill = murder))
```

l + geom_map(aes(map_id = state), map = map**) + expand_limits(**x = map$long, y = map$lat**)**
map_id, alpha, color, fill, linetype, size

Three Variables

```
seals$z <- with(seals, sqrt(delta_long^2 + delta_lat^2))
m <- ggplot(seals, aes(long, lat))
```

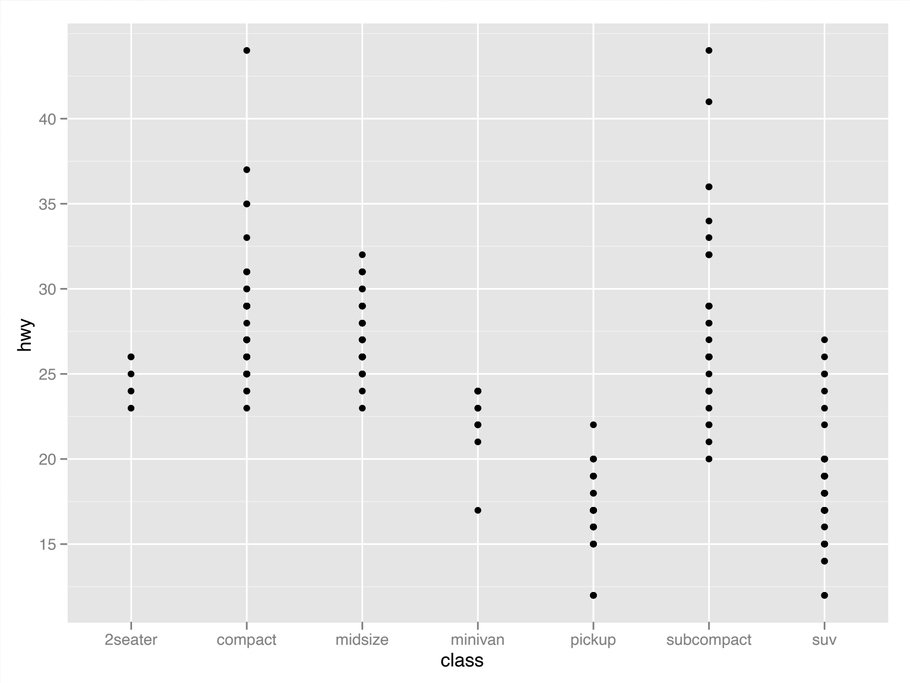
m + geom_contour(aes(z = z)**)**
x, y, z, alpha, colour, linetype, size, weight

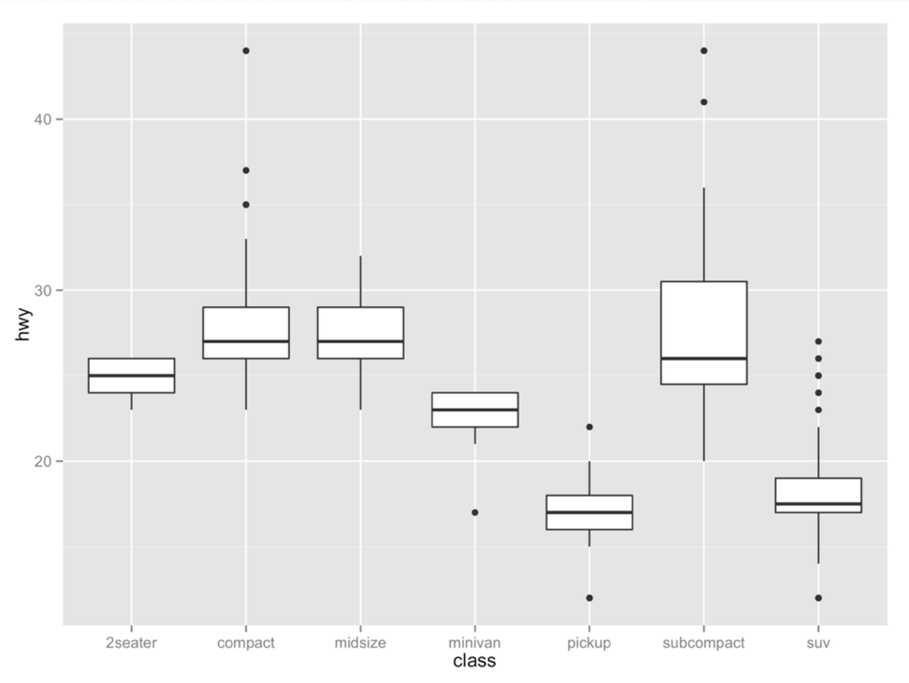
m + geom_raster(aes(fill = z), hjust=0.5, vjust=0.5, interpolate=FALSE**)**
x, y, alpha, fill (fast)

m + geom_tile(aes(fill = z)**)**
x, y, alpha, color, fill, linetype, size (slow)

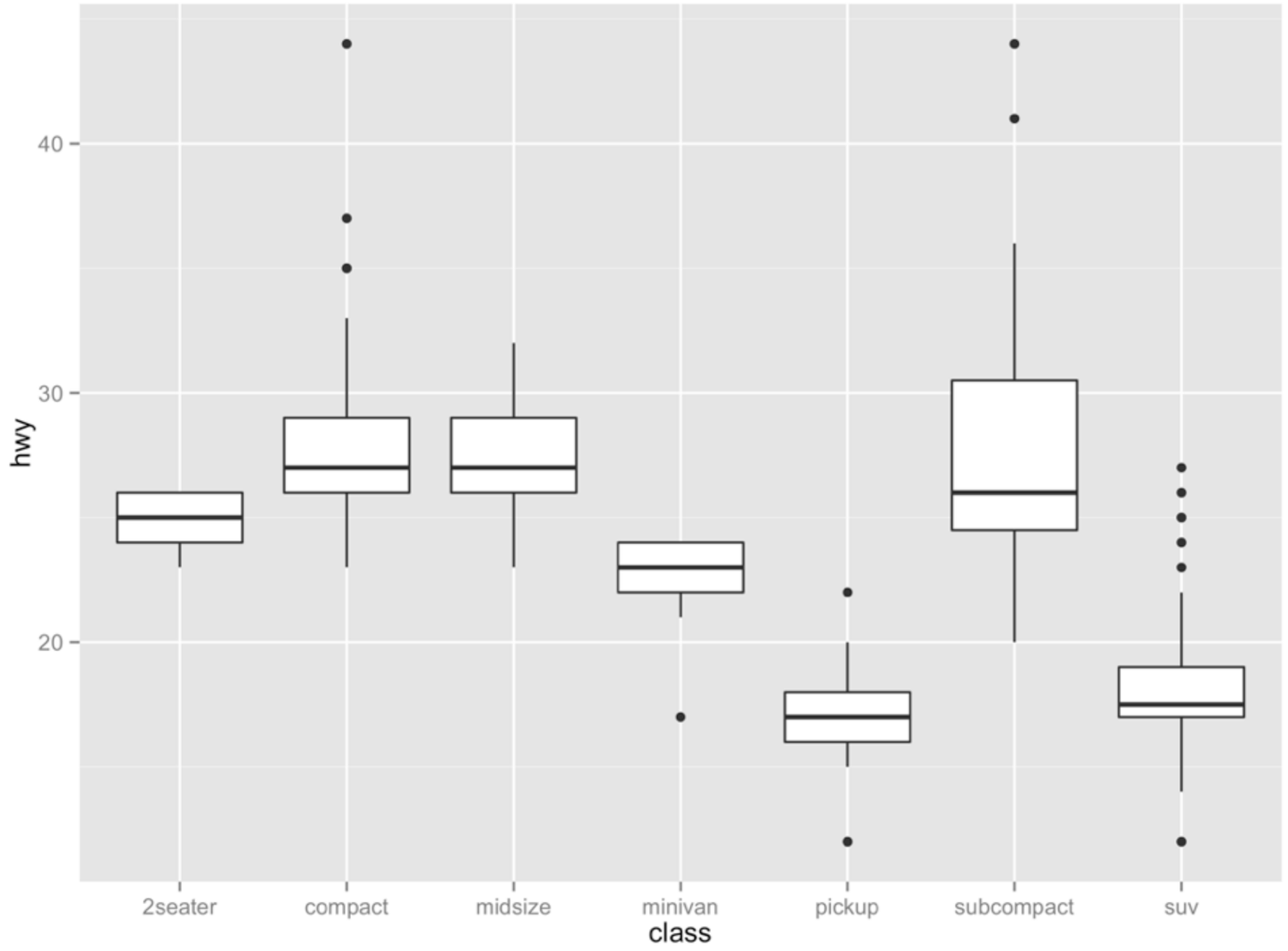
Tocca a te III

Con il tuo partner, pensate come sostituire questo grafico a dispersione (scatterplot) con uno che disegna grafici a scatole (boxplot). Usa il cheatsheet di prima. Prova la tua ipotesi migliore





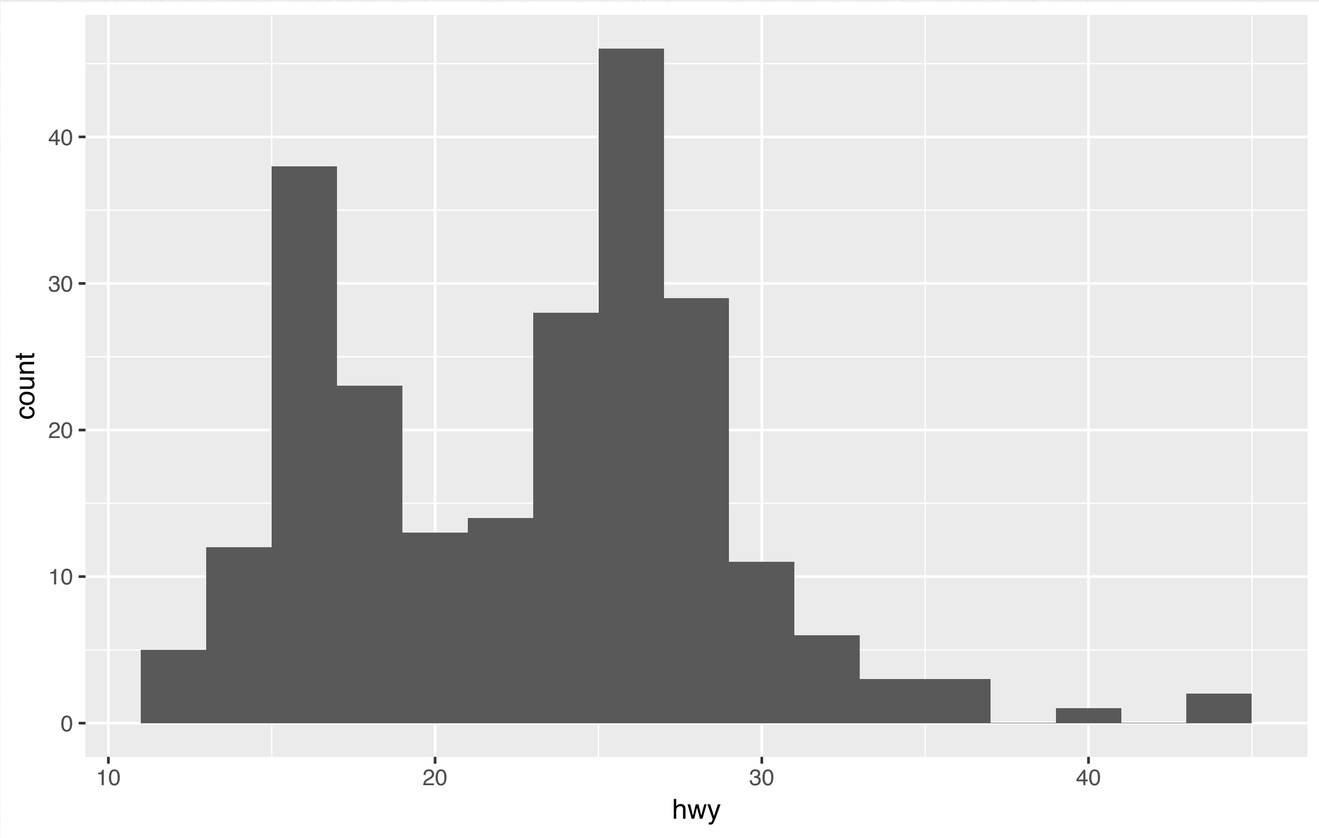

```
ggplot(mpg) + geom_point(aes(class, hwy))
```

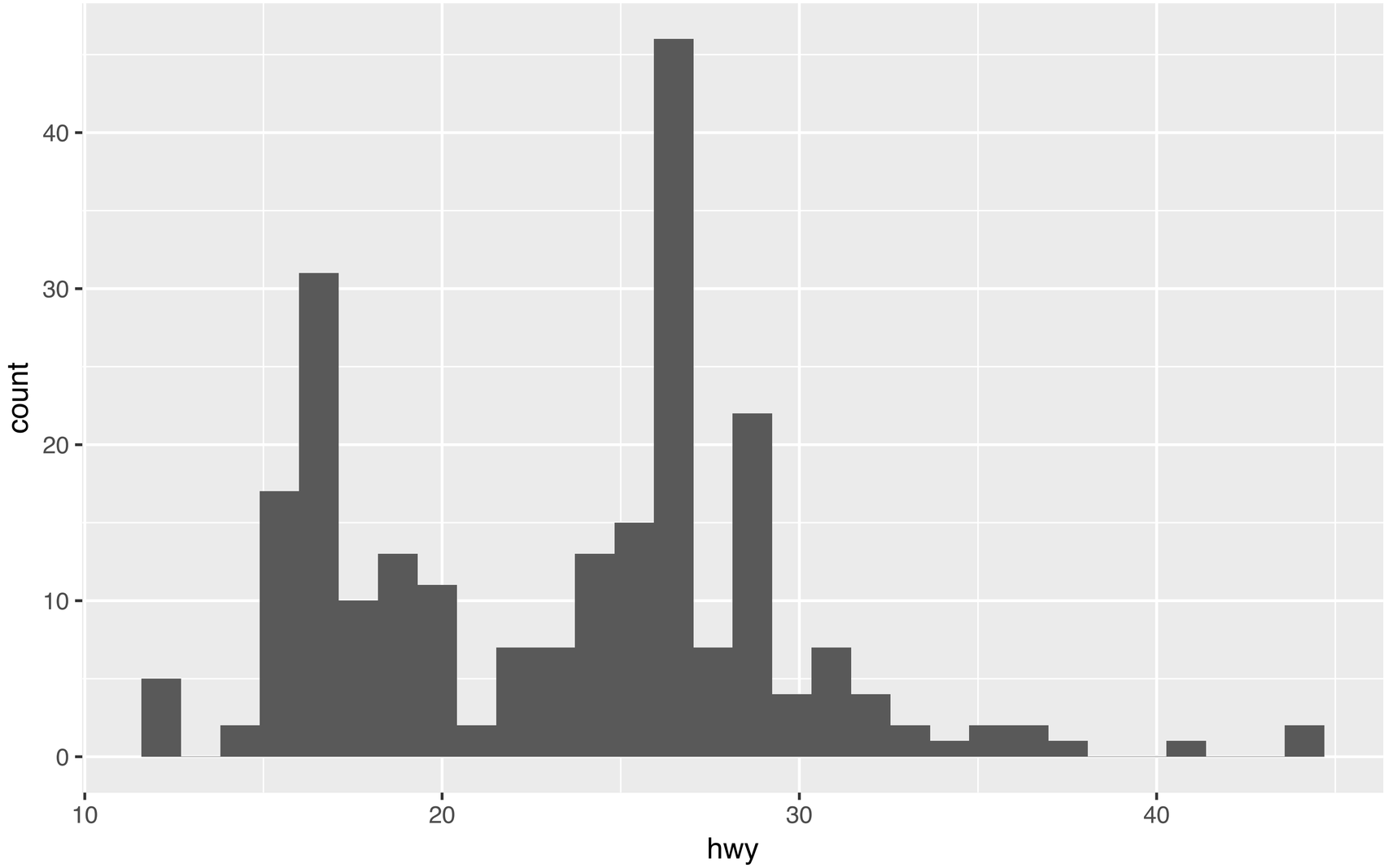



```
ggplot(data = mpg) +
  geom_boxplot(mapping = aes(x = class, y = hwy))
```

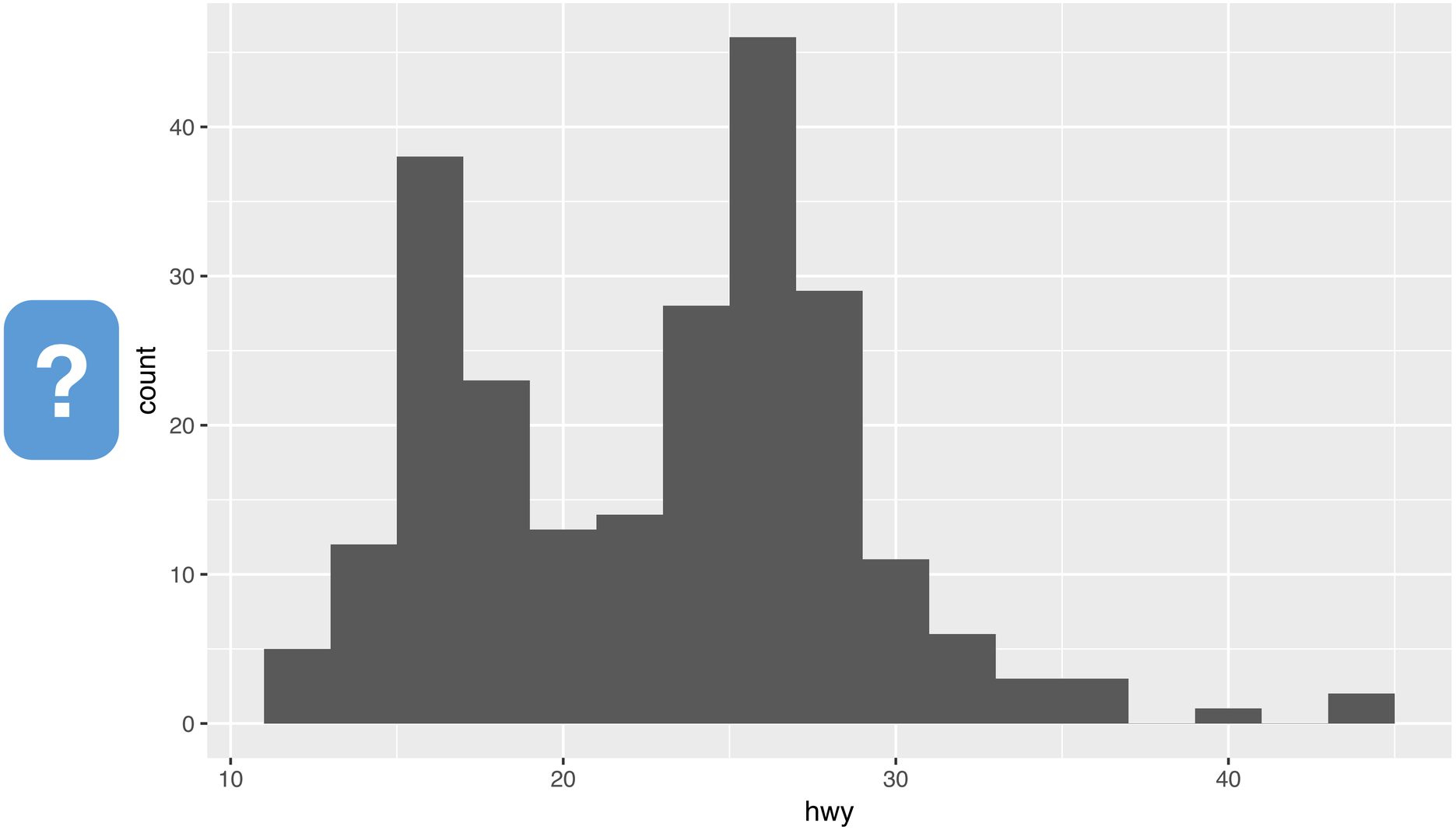
Tocca a te IV

Con il tuo partner, crea l'istogramma di hwy di seguito. Usa il cheatsheet. Suggerimento: non fornire alcuna variabile.





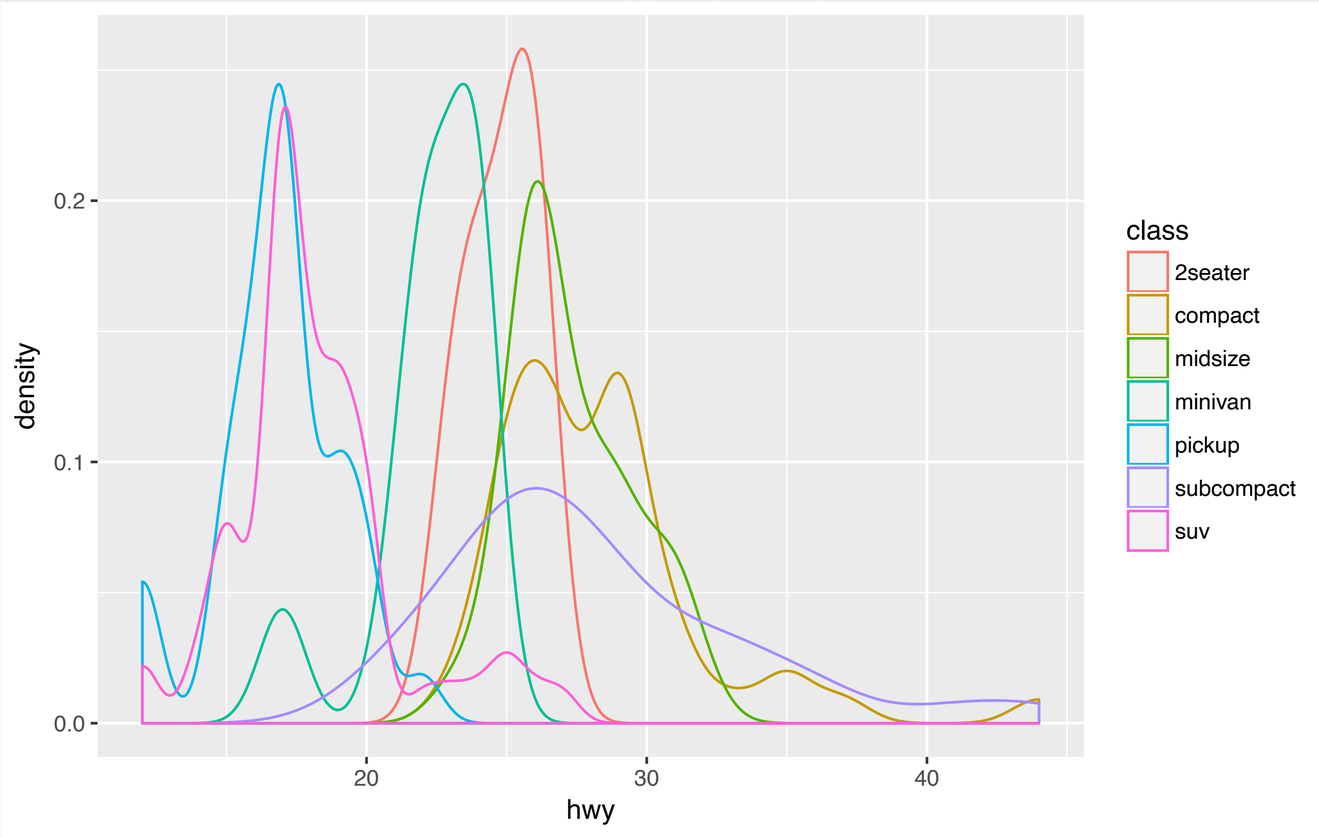

```
ggplot(data = mpg) +
  geom_histogram(mapping = aes(x = hwy))
```

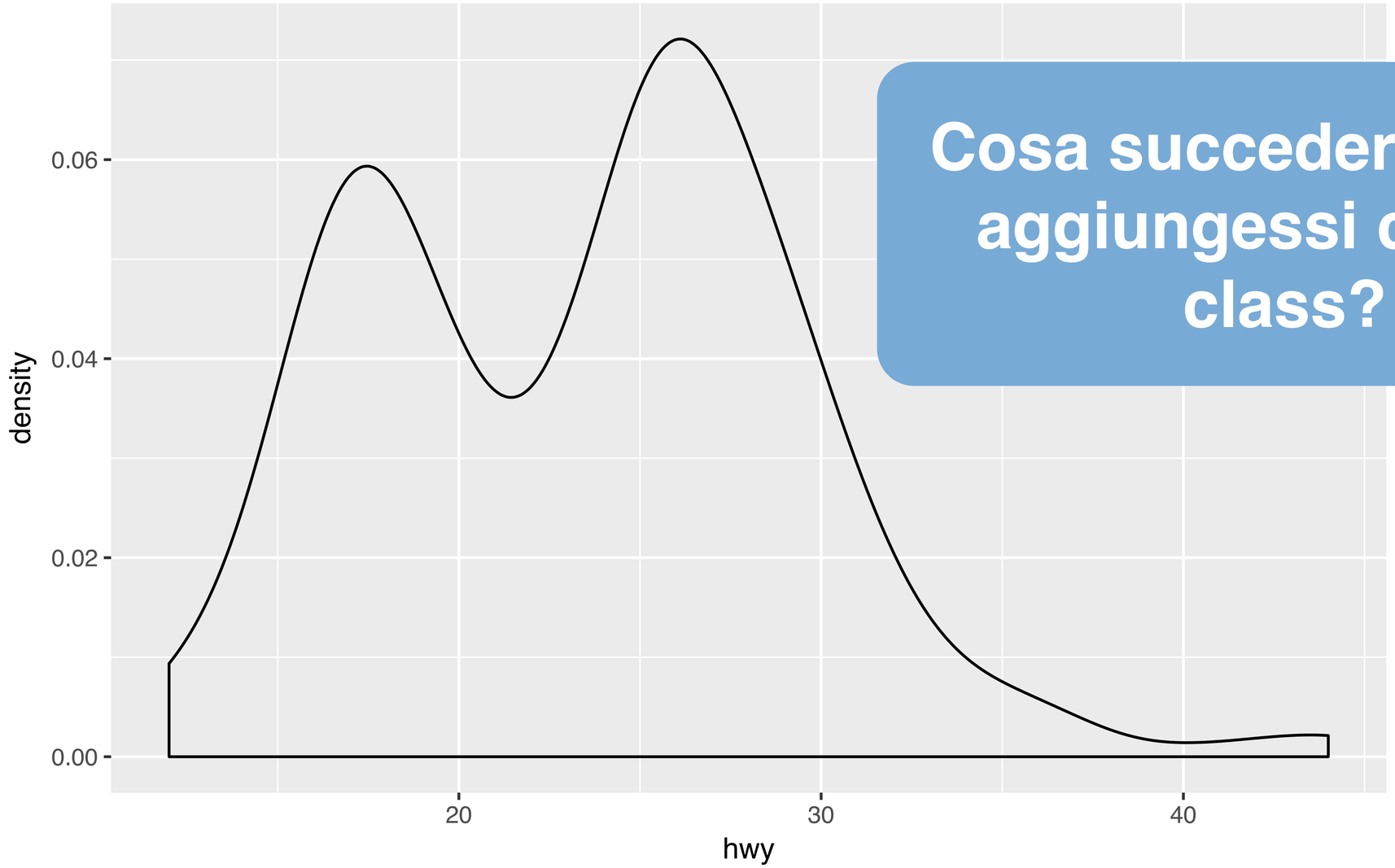



```
ggplot(data = mpg) +
  geom_histogram(mapping = aes(x = hwy), binwidth = 2)
```

Tocca a te V

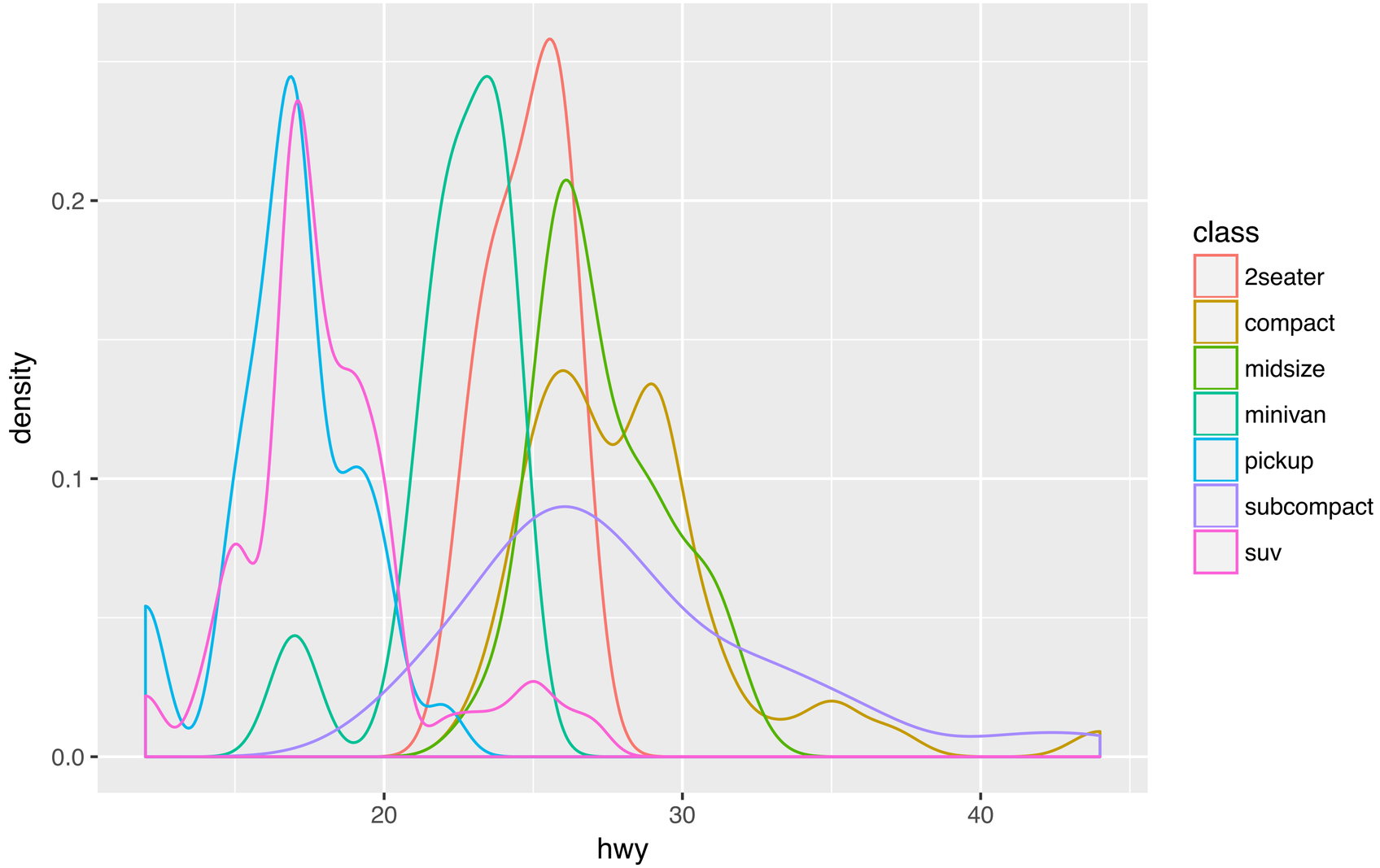
Con il tuo partner, crea la trama della densità di hwy colorata per classe di seguito. Usa il cheatsheet. Prova la tua ipotesi migliore.



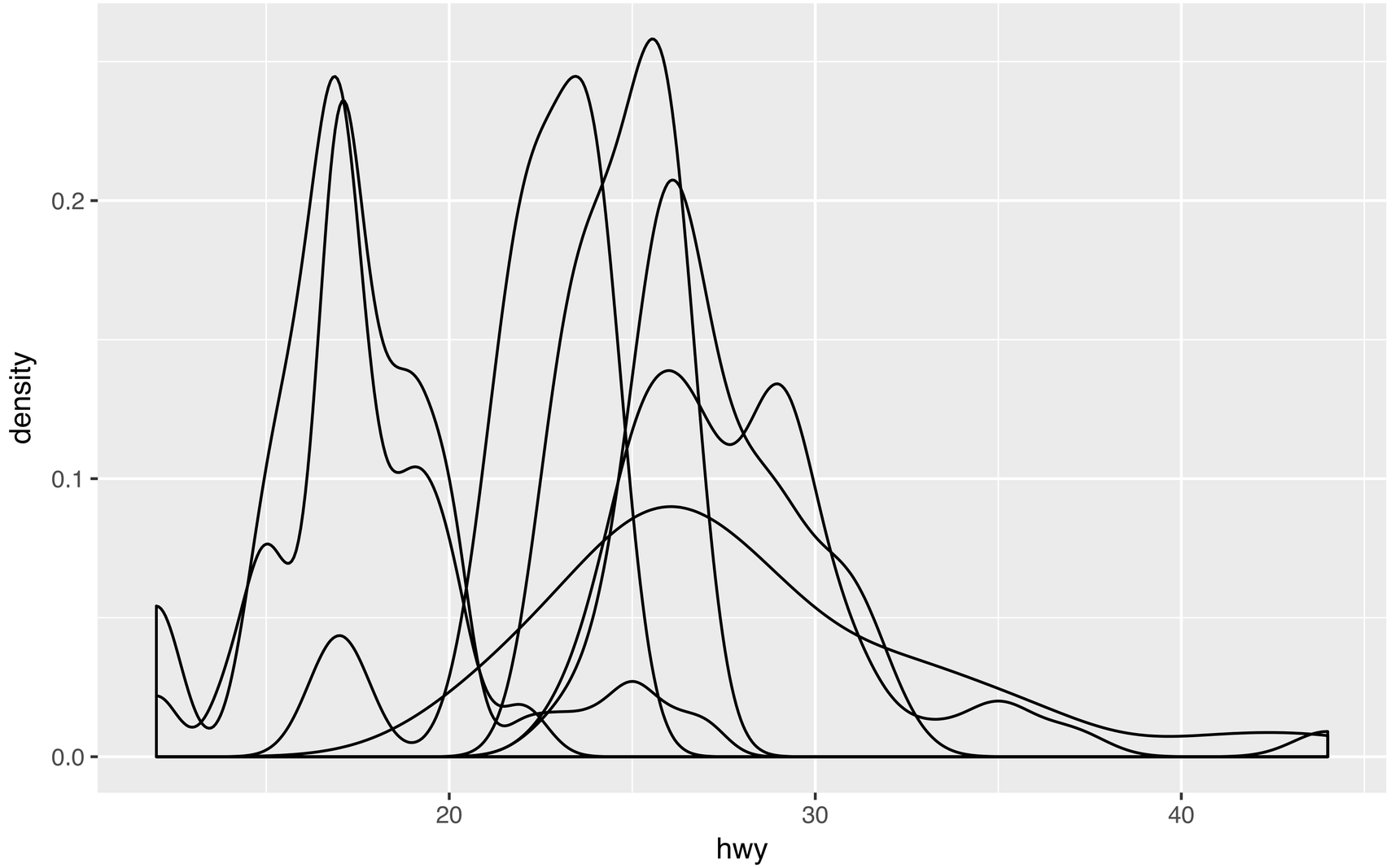


Cosa succederebbe se aggiungessi color = class?

```
ggplot(data = mpg) +
  geom_density(mapping = aes(x = hwy))
```



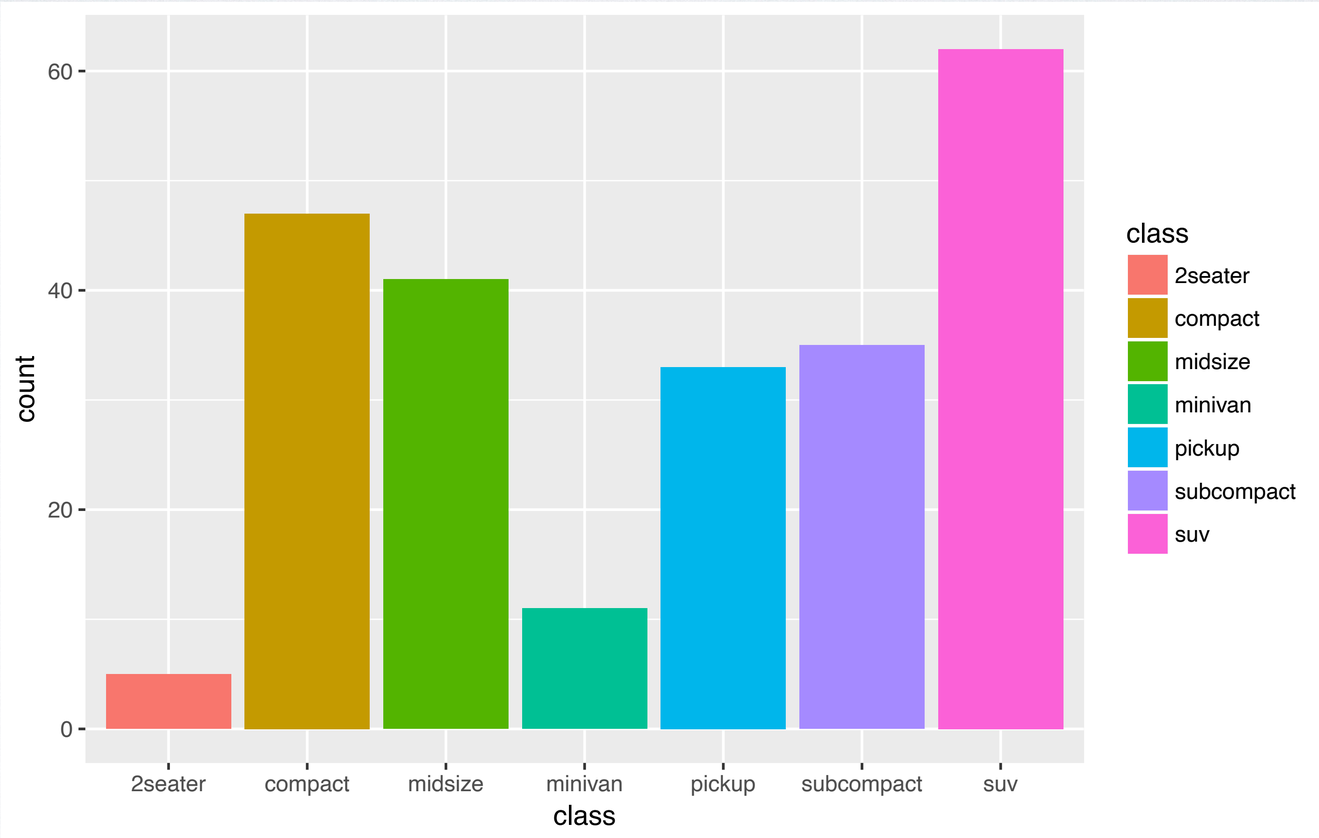

```
ggplot(data = mpg) +
  geom_density(mapping = aes(x = hwy, color = class))
```

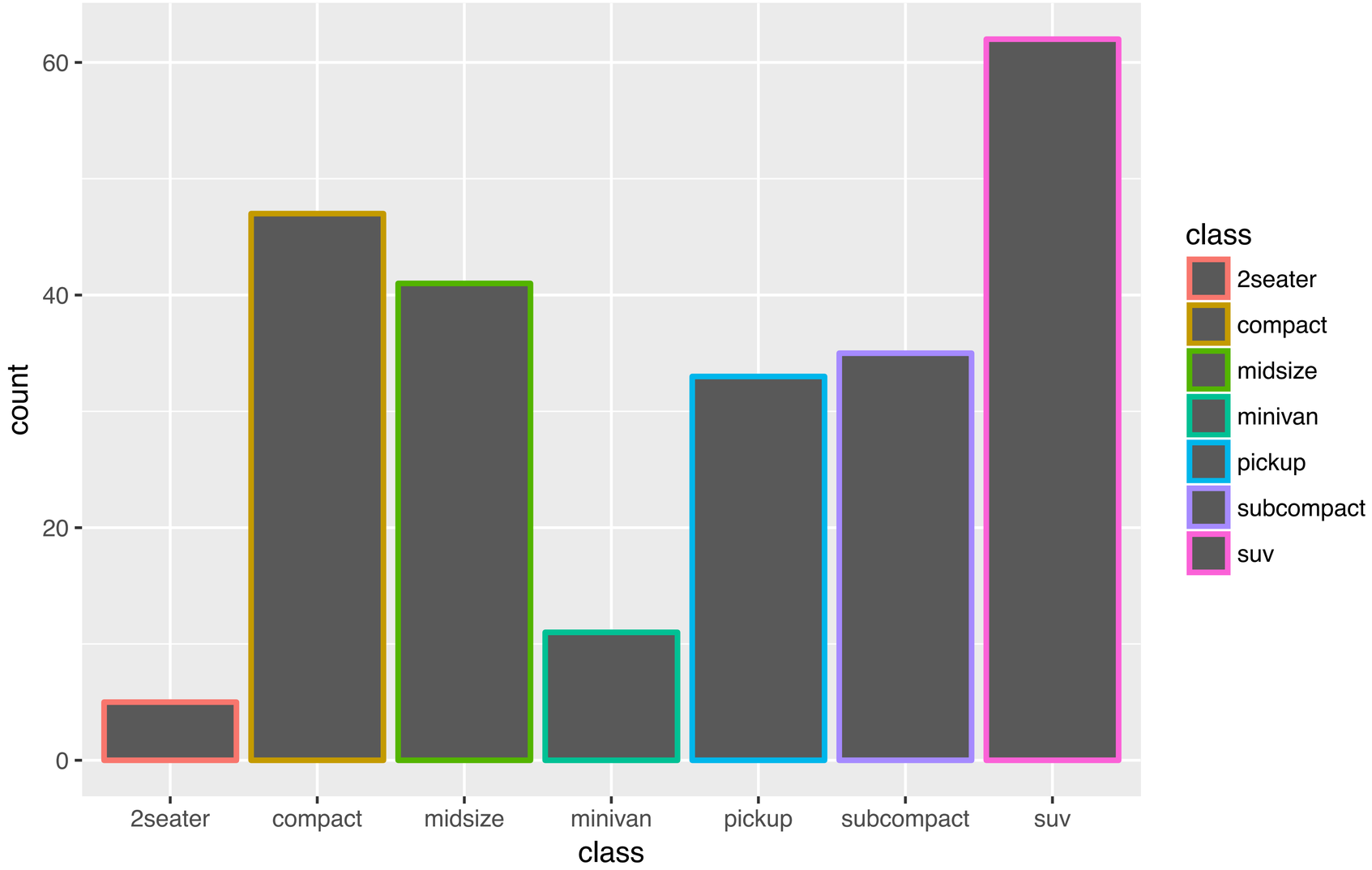



```
ggplot(data = mpg) +
  geom_density(mapping = aes(x = hwy, group = class))
```

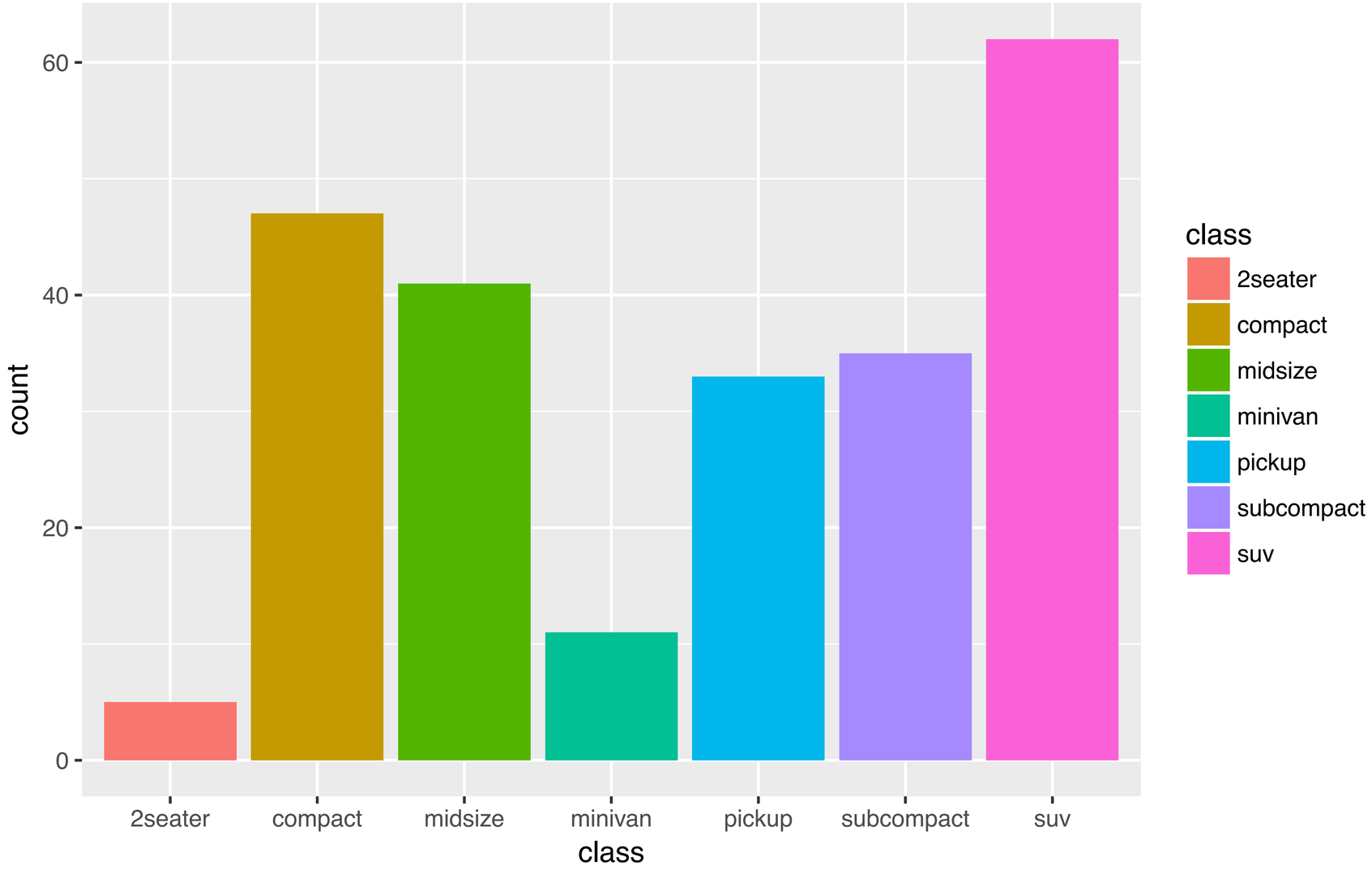
Tocca a te VI

Con il tuo partner, crea il grafico a barre della classe colorato per classe di seguito. Usa il cheatsheet. Prova la tua ipotesi migliore.

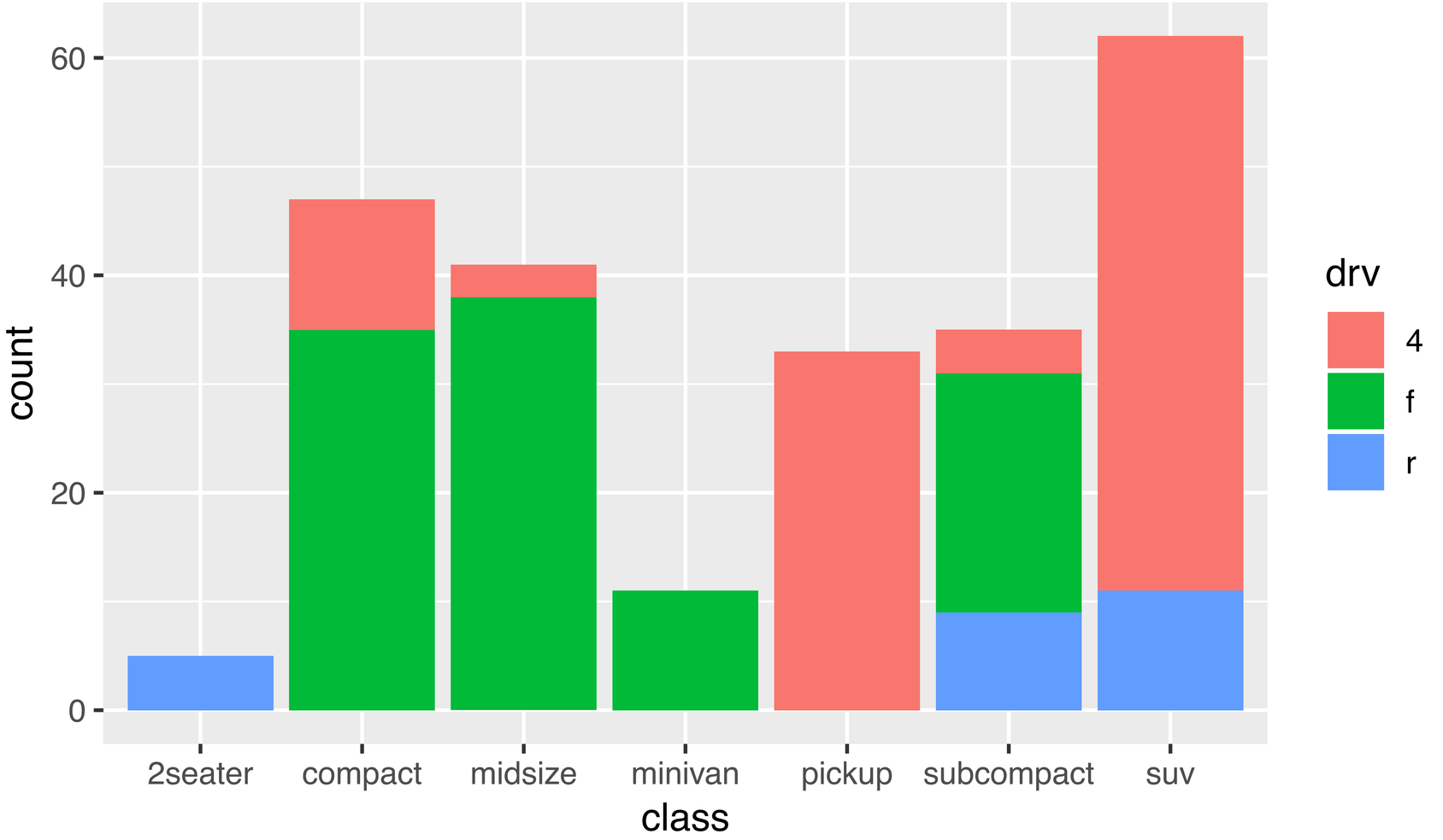





```
ggplot(data = mpg) +
  geom_bar(mapping = aes(x = class, color = class))
```




```
ggplot(data = mpg) +
  geom_bar(mapping = aes(x = class, fill = class))
```

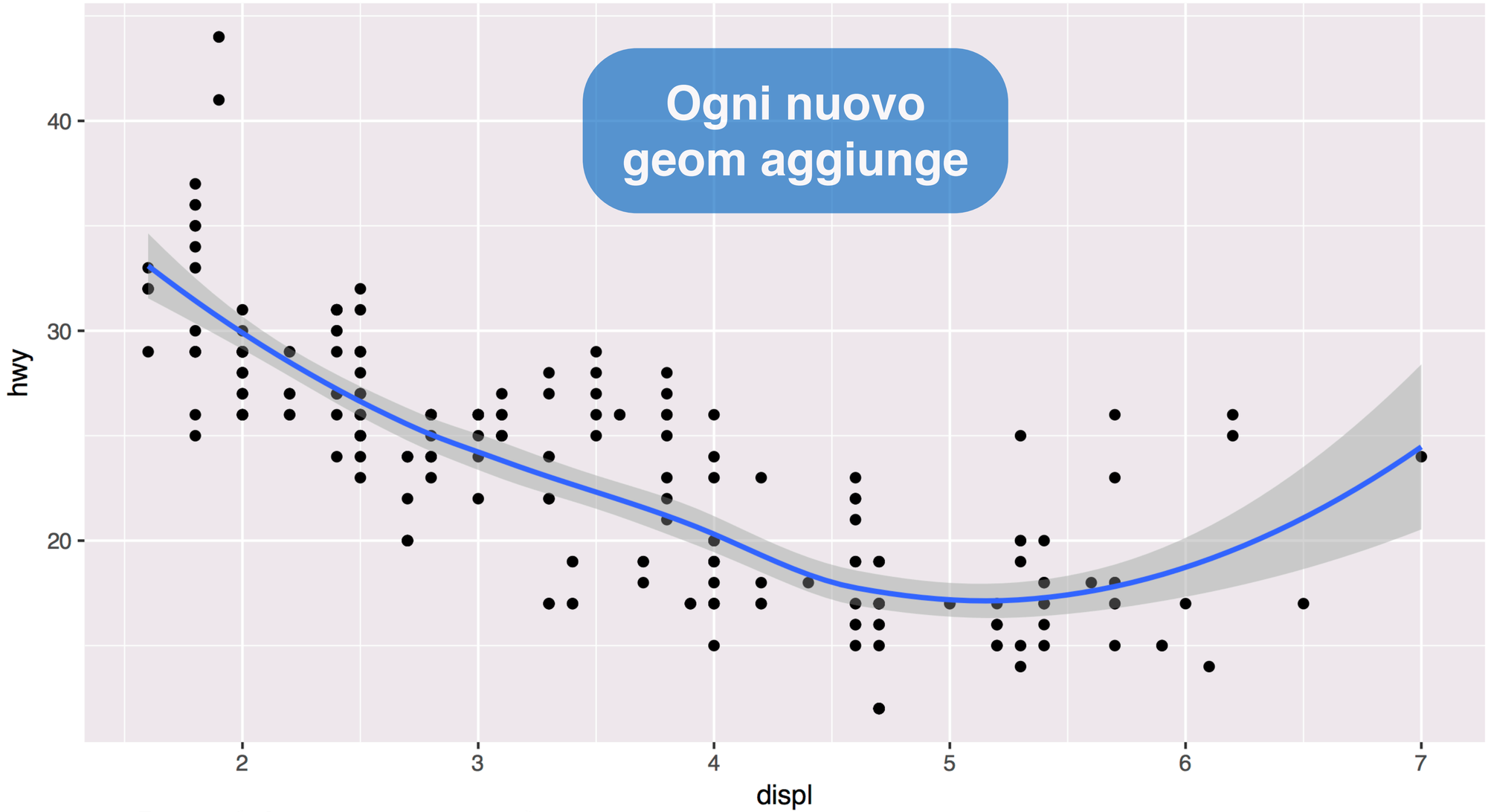



```
ggplot(data = mpg) +
  geom_bar(mapping = aes(x = class, fill = drv))
```

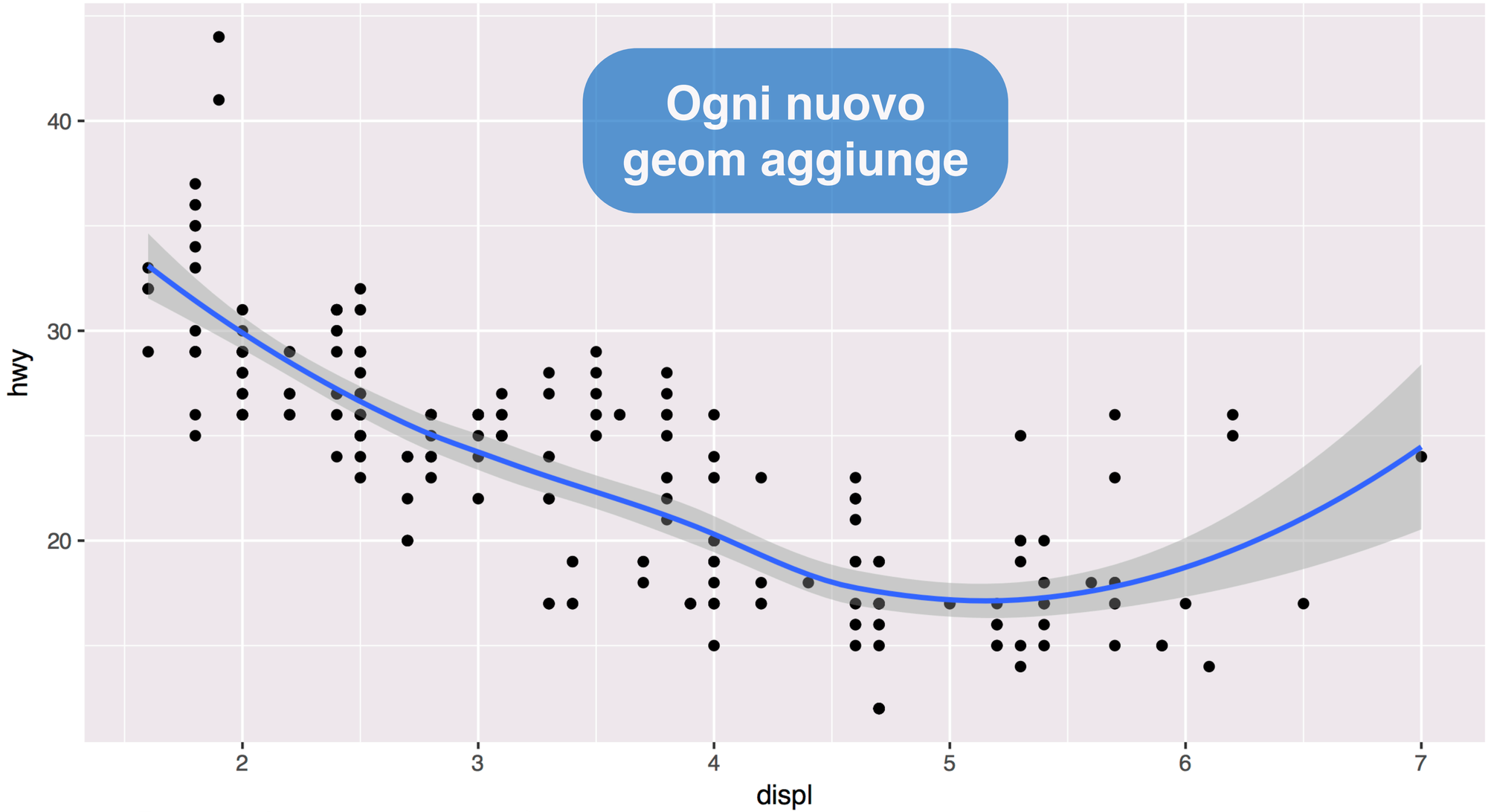
Quiz?

Cosa fa questo codice?

```
ggplot(mpg) +
  geom_point(aes(displ, hwy)) +
  geom_smooth(aes(displ, hwy))
```

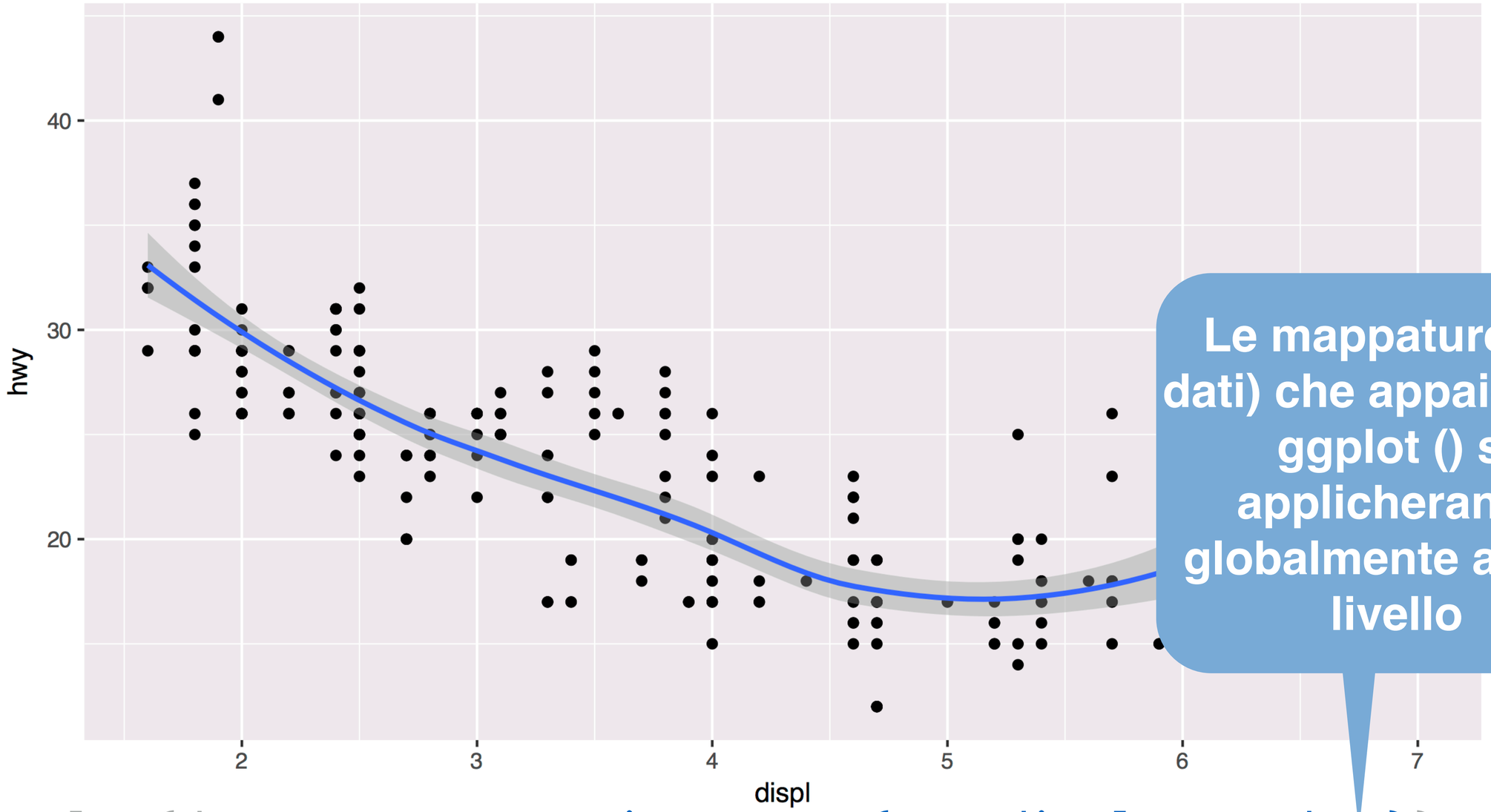



```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  geom_smooth(mapping = aes(x = displ, y = hwy))
```



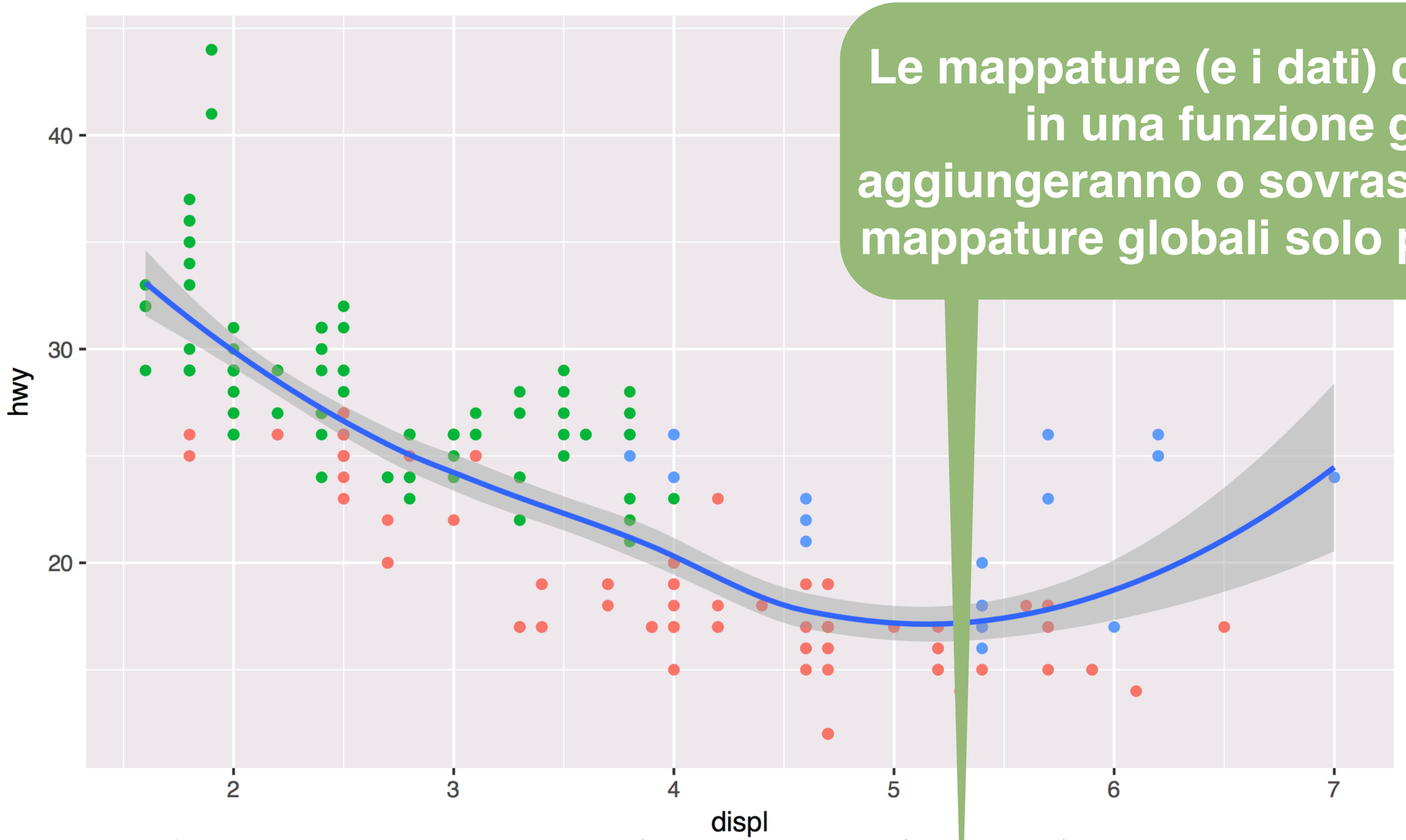

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  geom_smooth(mapping = aes(x = displ, y = hwy))
```

global vs. local



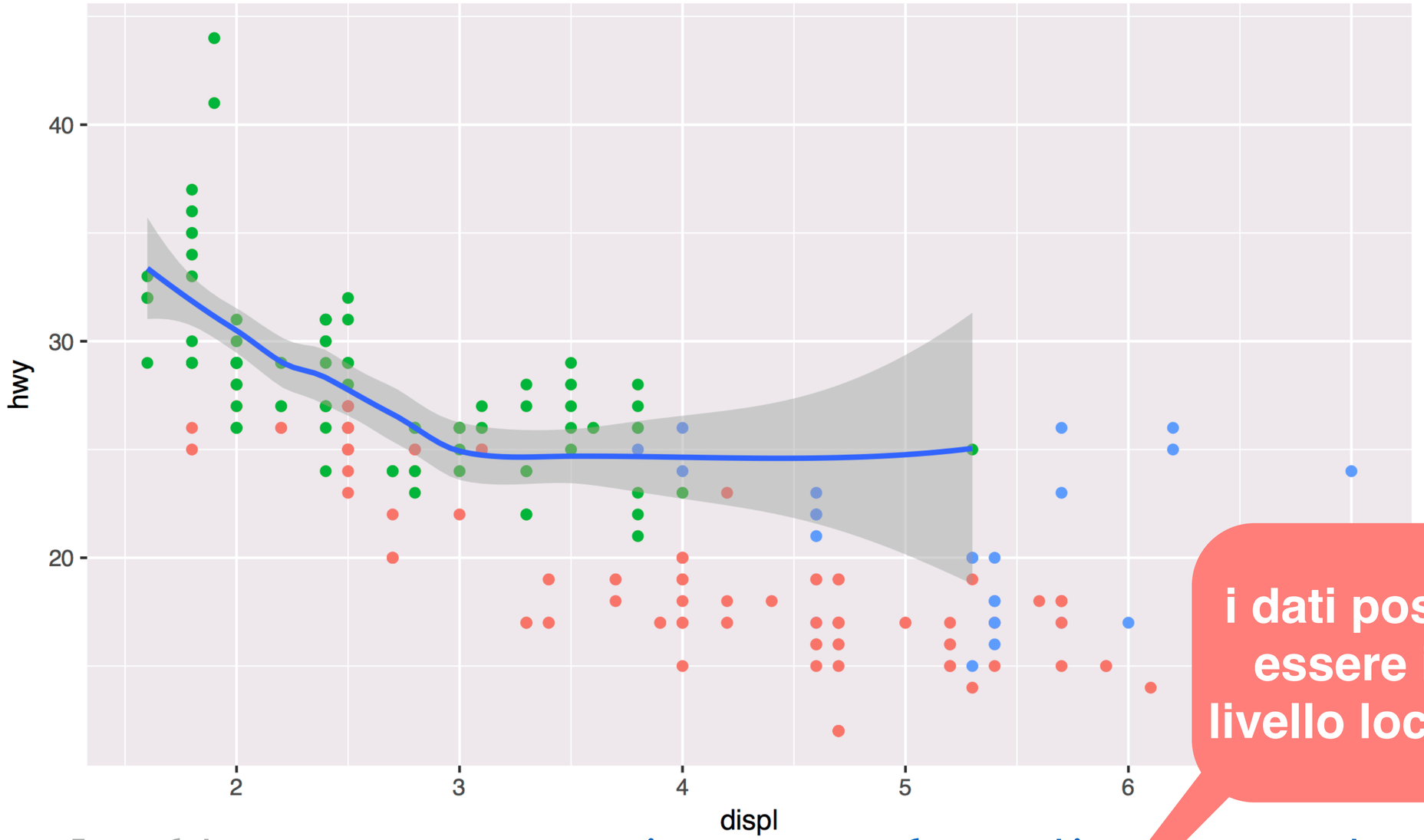
Le mappature (e i dati) che appaiono in ggplot () si applicheranno globalmente a ogni livello

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



Le mappature (e i dati) che appaiono in una funzione geom_ aggiungeranno o sovrascriveranno le mappature globali solo per quel layer

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

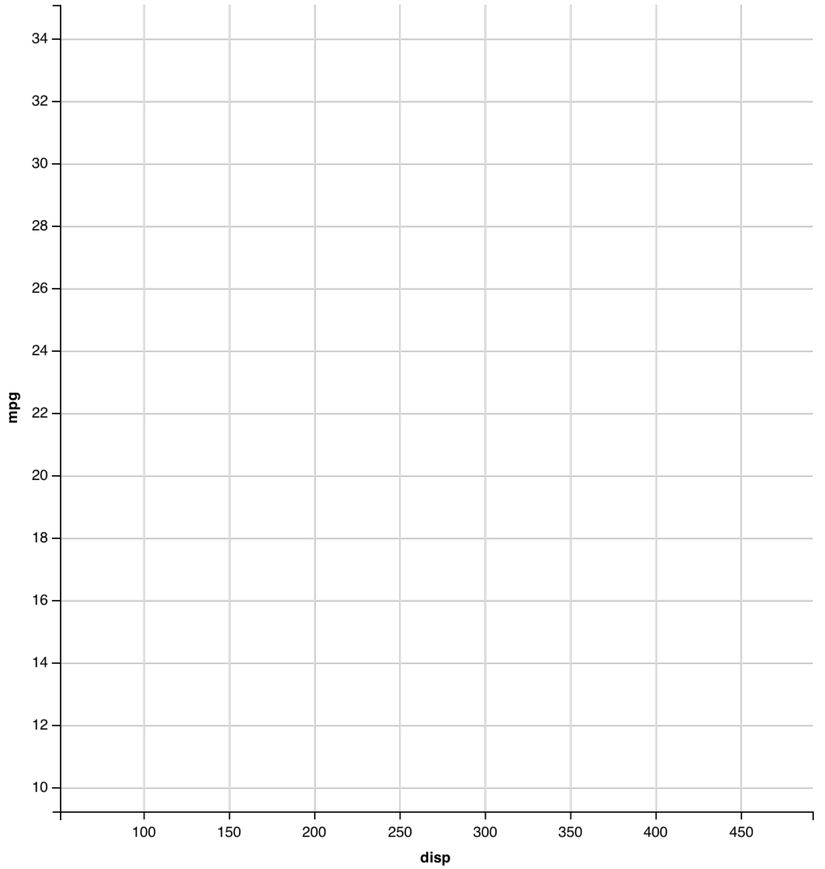


i dati possono anche essere impostati a livello locale o globale

```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point(mapping = aes(color = drv)) +
  geom_smooth(data = filter(mpg, drv == "f"))
```

Grammatica di grafica

mpg	cyl	disp	hp
21,0	6	160,0	2
21,0	6	160,0	2
22,8	4	108,0	1
21,4	6	258,0	2
18,7	8	360,0	3
18,1	6	225,0	2
14,3	8	360,0	5
24,4	4	146,7	1
22,8	4	140,8	1
19,2	6	167,6	2
17,8	6	167,6	2
16,4	8	275,8	3
17,3	8	275,8	3
15,2	8	275,8	3
10,4	8	472,0	4
10,4	8	460,0	4
14,7	8	440,0	4
32,4	4	78,7	1
30,4	4	75,7	1
33,9	4	71,1	1

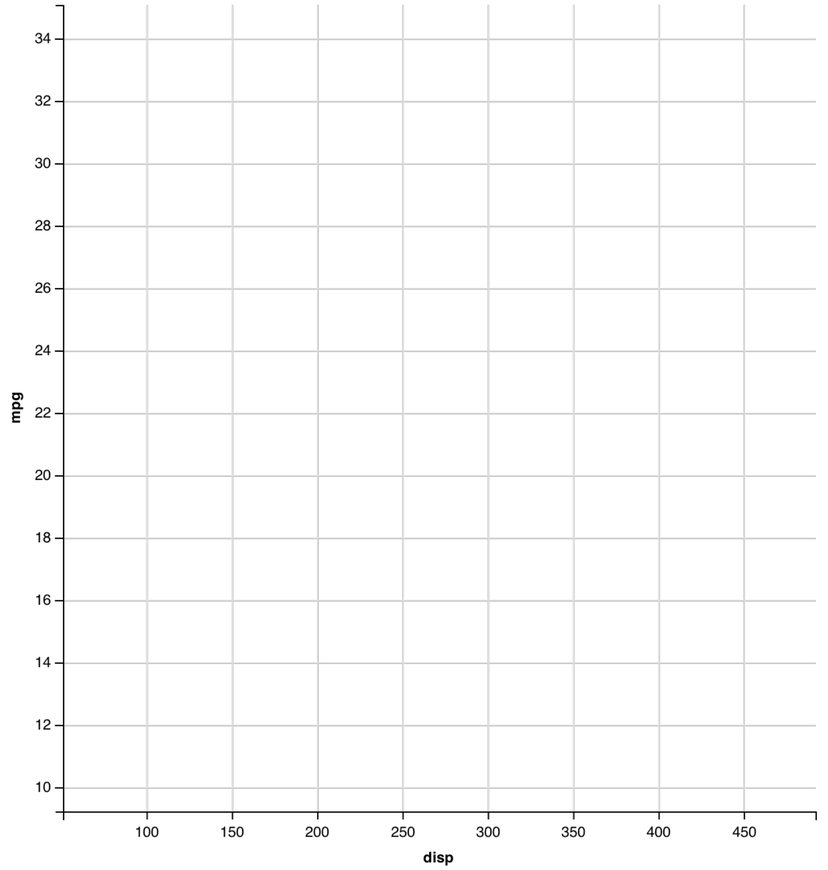


data **geom**

mappings

fill

mpg	cyl	disp	hp
21,0	6	160,0	2
21,0	6	160,0	2
22,8	4	108,0	1
21,4	6	258,0	2
18,7	8	360,0	3
18,1	6	225,0	2
14,3	8	360,0	5
24,4	4	146,7	1
22,8	4	140,8	1
19,2	6	167,6	2
17,8	6	167,6	2
16,4	8	275,8	3
17,3	8	275,8	3
15,2	8	275,8	3
10,4	8	472,0	4
10,4	8	460,0	4
14,7	8	440,0	4
32,4	4	78,7	1
30,4	4	75,7	1
33,9	4	71,1	1

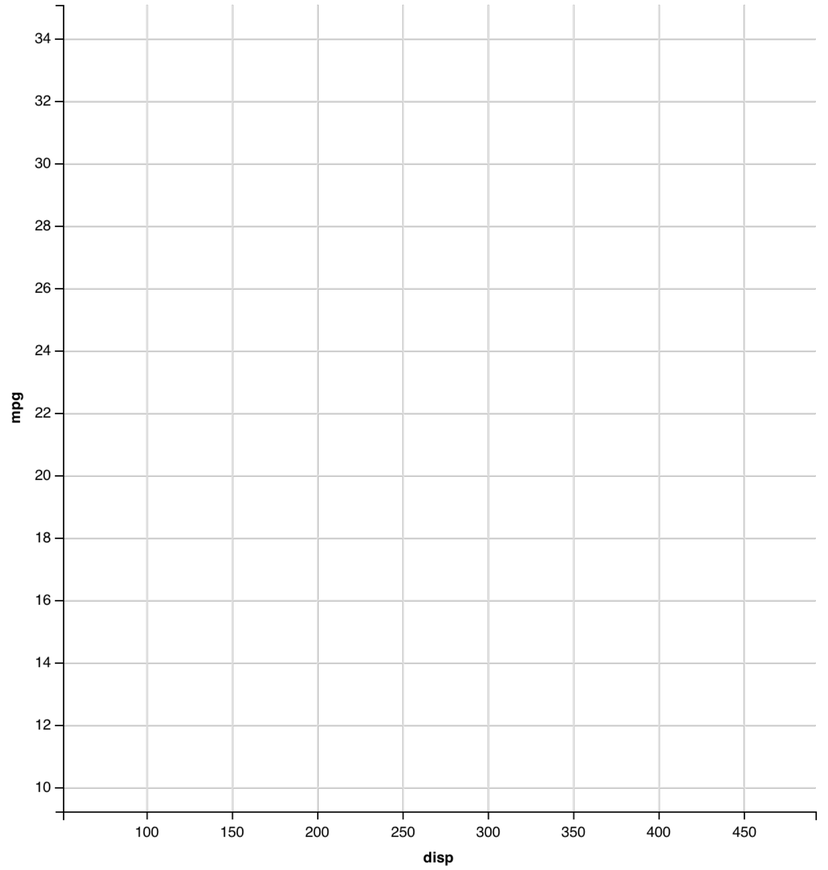


data **geom**

mappings

shape fill

mpg	cyl	disp	hp
21,0	6	160,0	2
21,0	6	160,0	2
22,8	4	108,0	1
21,4	6	258,0	2
18,7	8	360,0	3
18,1	6	225,0	2
14,3	8	360,0	5
24,4	4	146,7	1
22,8	4	140,8	1
19,2	6	167,6	2
17,8	6	167,6	2
16,4	8	275,8	3
17,3	8	275,8	3
15,2	8	275,8	3
10,4	8	472,0	4
10,4	8	460,0	4
14,7	8	440,0	4
32,4	4	78,7	1
30,4	4	75,7	1
33,9	4	71,1	1

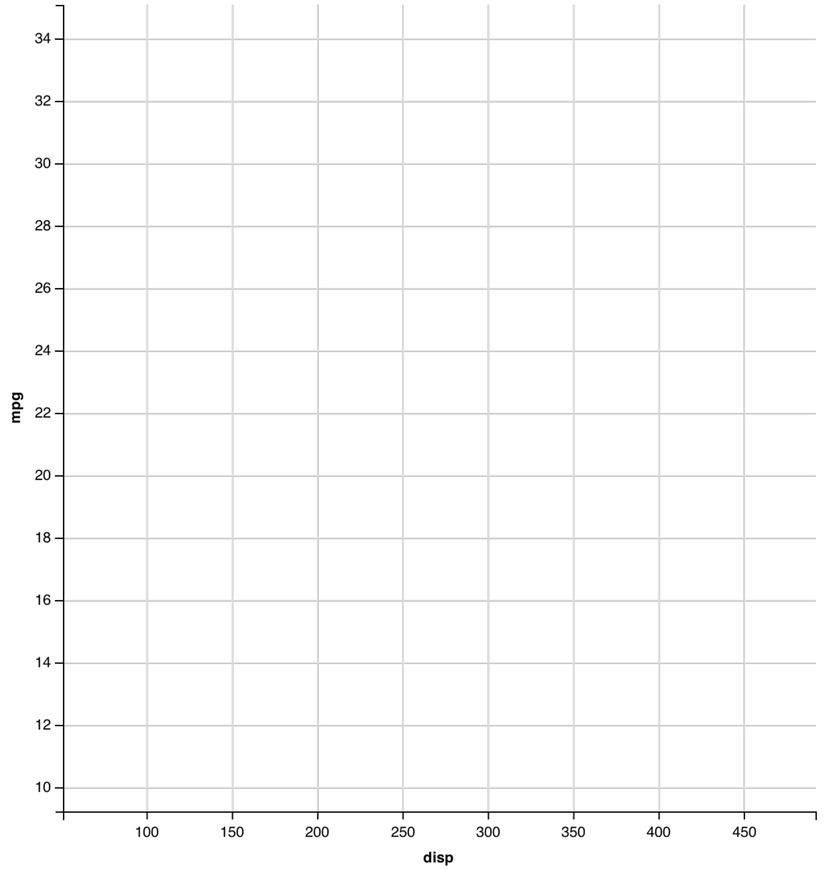


data **geom**

mappings

shape x fill

mpg	cyl	disp	hp
21,0	6	160,0	2
21,0	6	160,0	2
22,8	4	108,0	1
21,4	6	258,0	2
18,7	8	360,0	3
18,1	6	225,0	2
14,3	8	360,0	5
24,4	4	146,7	1
22,8	4	140,8	1
19,2	6	167,6	2
17,8	6	167,6	2
16,4	8	275,8	3
17,3	8	275,8	3
15,2	8	275,8	3
10,4	8	472,0	4
10,4	8	460,0	4
14,7	8	440,0	4
32,4	4	78,7	1
30,4	4	75,7	1
33,9	4	71,1	1

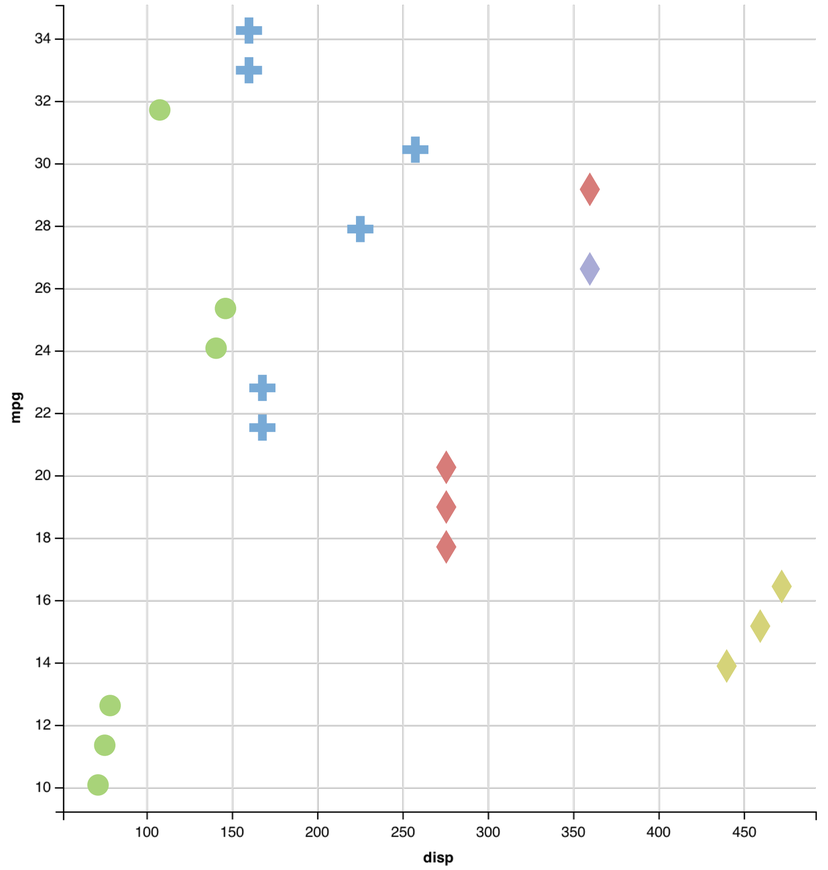


data **geom**

mappings

y shape x fill

mpg	cyl	disp	hp
21,0	6	160,0	2
21,0	6	160,0	2
22,8	4	108,0	1
21,4	6	258,0	2
18,7	8	360,0	3
18,1	6	225,0	2
14,3	8	360,0	5
24,4	4	146,7	1
22,8	4	140,8	1
19,2	6	167,6	2
17,8	6	167,6	2
16,4	8	275,8	3
17,3	8	275,8	3
15,2	8	275,8	3
10,4	8	472,0	4
10,4	8	460,0	4
14,7	8	440,0	4
32,4	4	78,7	1
30,4	4	75,7	1
33,9	4	71,1	1

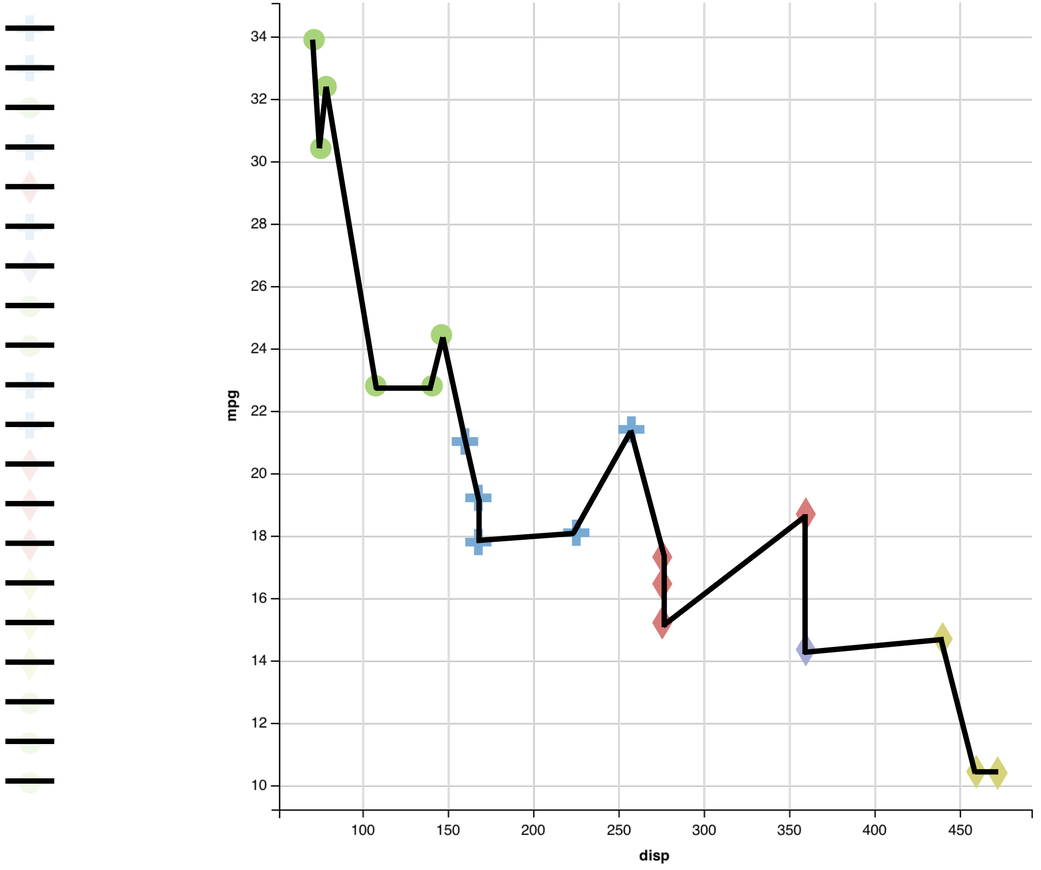


data **geom**

mappings

y shape x fill

mpg	cyl	disp	hp
21,0	6	160,0	2
21,0	6	160,0	2
22,8	4	108,0	1
21,4	6	258,0	2
18,7	8	360,0	3
18,1	6	225,0	2
14,3	8	360,0	5
24,4	4	146,7	1
22,8	4	140,8	1
19,2	6	167,6	2
17,8	6	167,6	2
16,4	8	275,8	3
17,3	8	275,8	3
15,2	8	275,8	3
10,4	8	472,0	4
10,4	8	460,0	4
14,7	8	440,0	4
32,4	4	78,7	1
30,4	4	75,7	1
33,9	4	71,1	1



data

geom

points

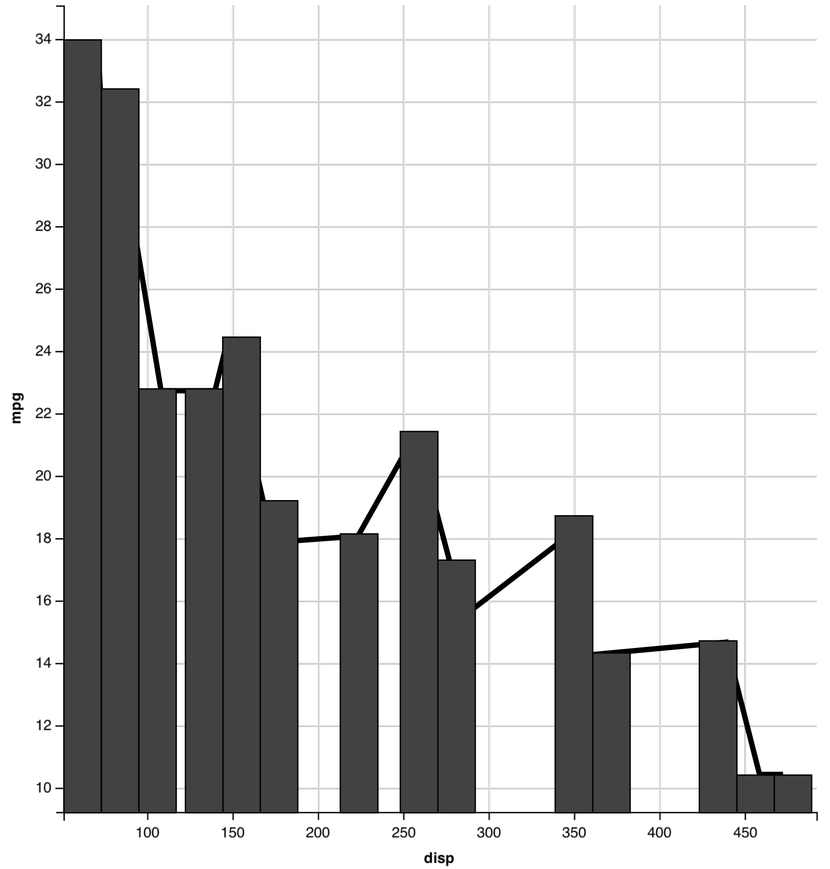
lines

ggplot2

mappings

y x

mpg	cyl	disp	hp
21,0	6	160,0	2
21,0	6	160,0	2
22,8	4	108,0	1
21,4	6	258,0	2
18,7	8	360,0	3
18,1	6	225,0	2
14,3	8	360,0	5
24,4	4	146,7	1
22,8	4	140,8	1
19,2	6	167,6	2
17,8	6	167,6	2
16,4	8	275,8	3
17,3	8	275,8	3
15,2	8	275,8	3
10,4	8	472,0	4
10,4	8	460,0	4
14,7	8	440,0	4
32,4	4	78,7	1
30,4	4	75,7	1
33,9	4	71,1	1



data **geom**

points

lines

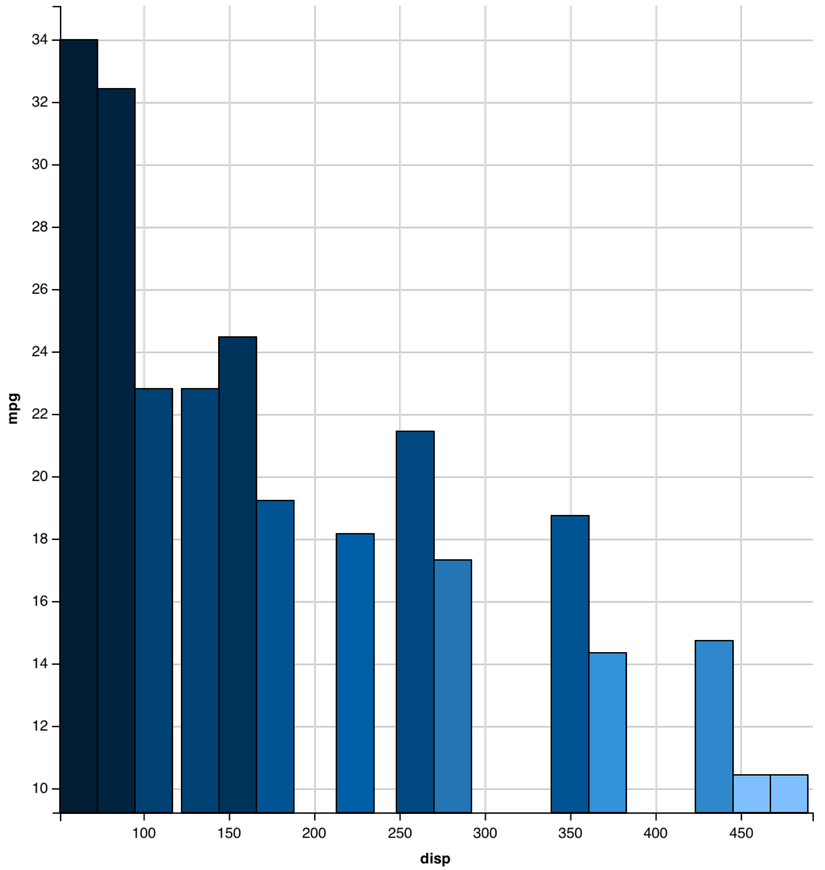
bars

mappings

y

fill

mpg	cyl	disp	hp
21,0	6	160,0	2
21,0	6	160,0	2
22,8	4	108,0	1
21,4	6	258,0	2
18,7	8	360,0	3
18,1	6	225,0	2
14,3	8	360,0	5
24,4	4	146,7	1
22,8	4	140,8	1
19,2	6	167,6	2
17,8	6	167,6	2
16,4	8	275,8	3
17,3	8	275,8	3
15,2	8	275,8	3
10,4	8	472,0	4
10,4	8	460,0	4
14,7	8	440,0	4
32,4	4	78,7	1
30,4	4	75,7	1
33,9	4	71,1	1



data

geom

points

lines

bars

To make a graph

[modello]

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

To make a graph

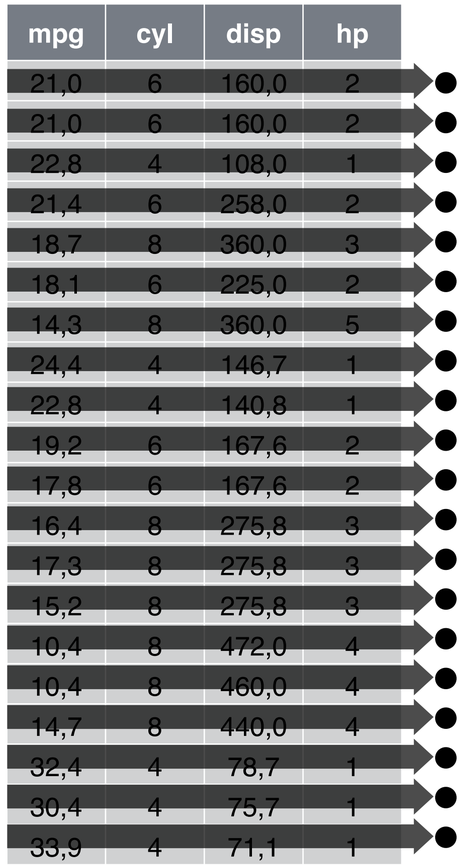
mpg	cyl	disp	hp
21,0	6	160,0	2
21,0	6	160,0	2
22,8	4	108,0	1
21,4	6	258,0	2
18,7	8	360,0	3
18,1	6	225,0	2
14,3	8	360,0	5
24,4	4	146,7	1
22,8	4	140,8	1
19,2	6	167,6	2
17,8	6	167,6	2
16,4	8	275,8	3
17,3	8	275,8	3
15,2	8	275,8	3
10,4	8	472,0	4
10,4	8	460,0	4
14,7	8	440,0	4
32,4	4	78,7	1
30,4	4	75,7	1
33,9	4	71,1	1

data

1. Scegli un set di dati

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

To make a graph



mpg	cyl	disp	hp
21,0	6	160,0	2
21,0	6	160,0	2
22,8	4	108,0	1
21,4	6	258,0	2
18,7	8	360,0	3
18,1	6	225,0	2
14,3	8	360,0	5
24,4	4	146,7	1
22,8	4	140,8	1
19,2	6	167,6	2
17,8	6	167,6	2
16,4	8	275,8	3
17,3	8	275,8	3
15,2	8	275,8	3
10,4	8	472,0	4
10,4	8	460,0	4
14,7	8	440,0	4
32,4	4	78,7	1
30,4	4	75,7	1
33,9	4	71,1	1

data **geom**

1. Scegli un set di dati

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

2. Scegli un geom per visualizzare i casi

To make a graph

mappings

fill

mpg	cyl	disp	hp
21,0	6	160,0	2
21,0	6	160,0	2
22,8	4	108,0	1
21,4	6	258,0	2
18,7	8	360,0	3
18,1	6	225,0	2
14,3	8	360,0	5
24,4	4	146,7	1
22,8	4	140,8	1
19,2	6	167,6	2
17,8	6	167,6	2
16,4	8	275,8	3
17,3	8	275,8	3
15,2	8	275,8	3
10,4	8	472,0	4
10,4	8	460,0	4
14,7	8	440,0	4
32,4	4	78,7	1
30,4	4	75,7	1
33,9	4	71,1	1

data **geom**

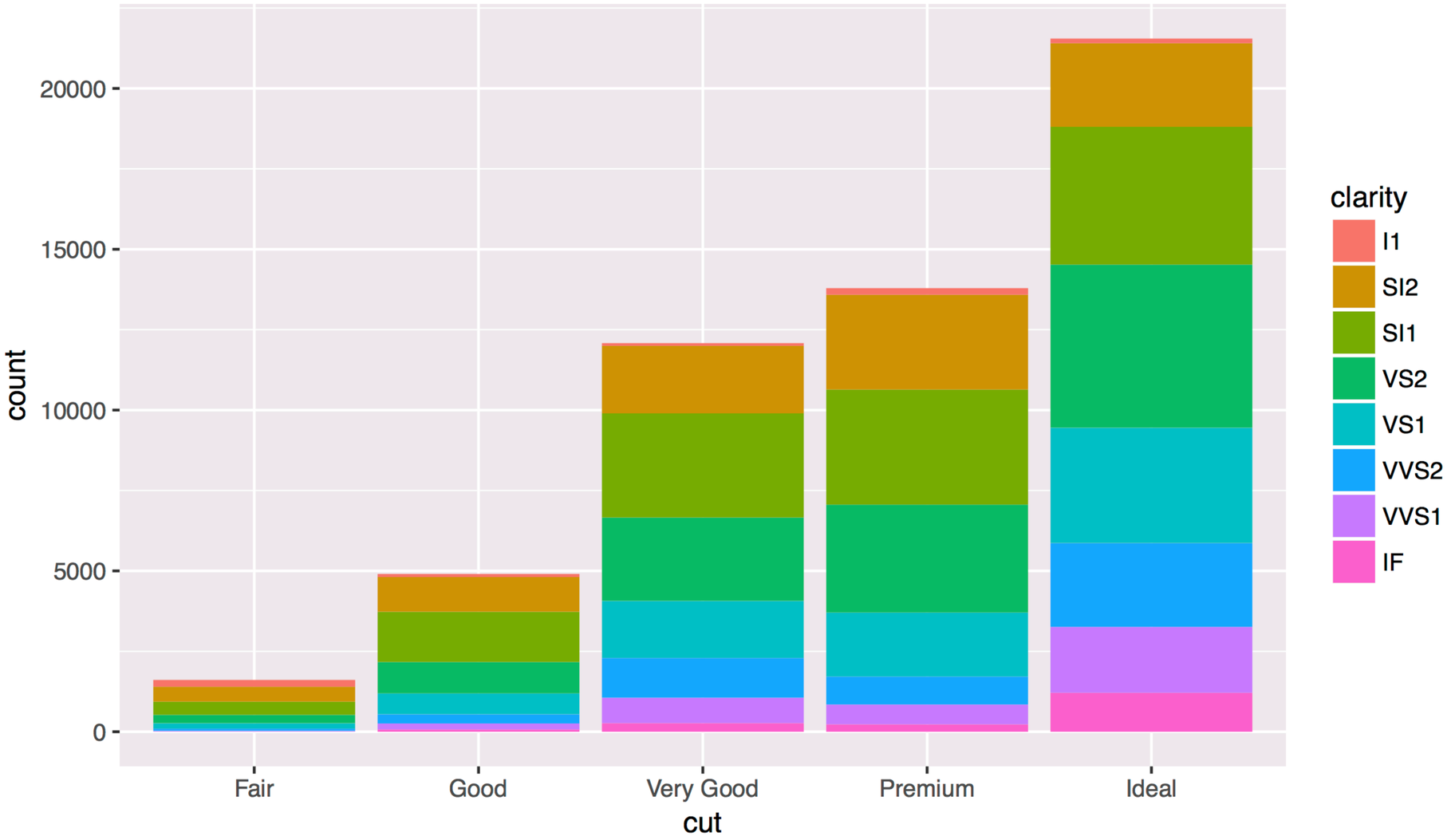
1. Scegli un set di dati

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

2. Scegli un geom per visualizzare i casi

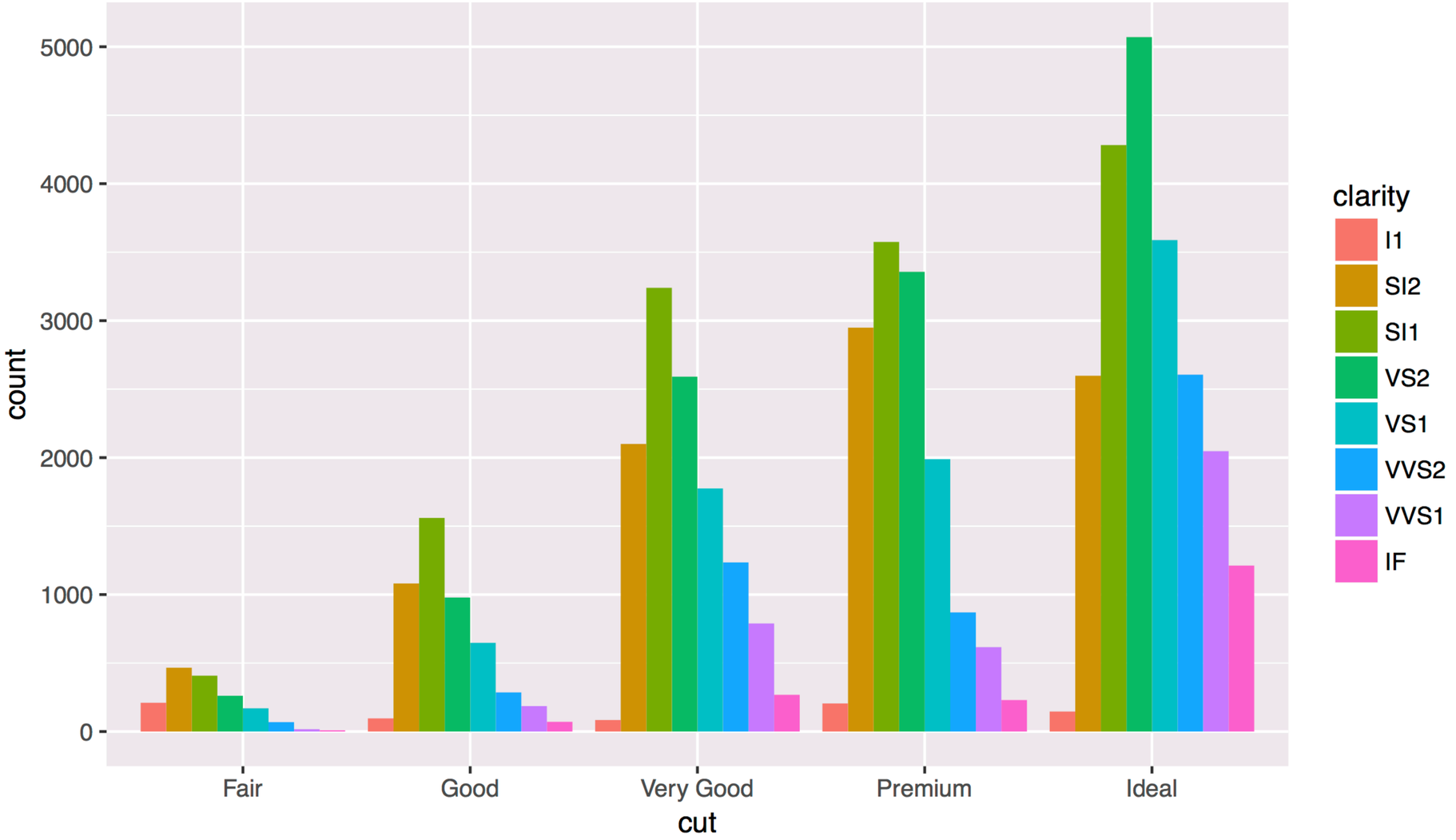
3. **Mappa le proprietà estetiche alle variabili**

Altro?


20000
15000
10000
5000
0
count
Fair
Good
Very Good
Premium
Ideal
cut
clarity
I1
SI2
SI1
VS2
VS1
VVS2
VVS1
IF

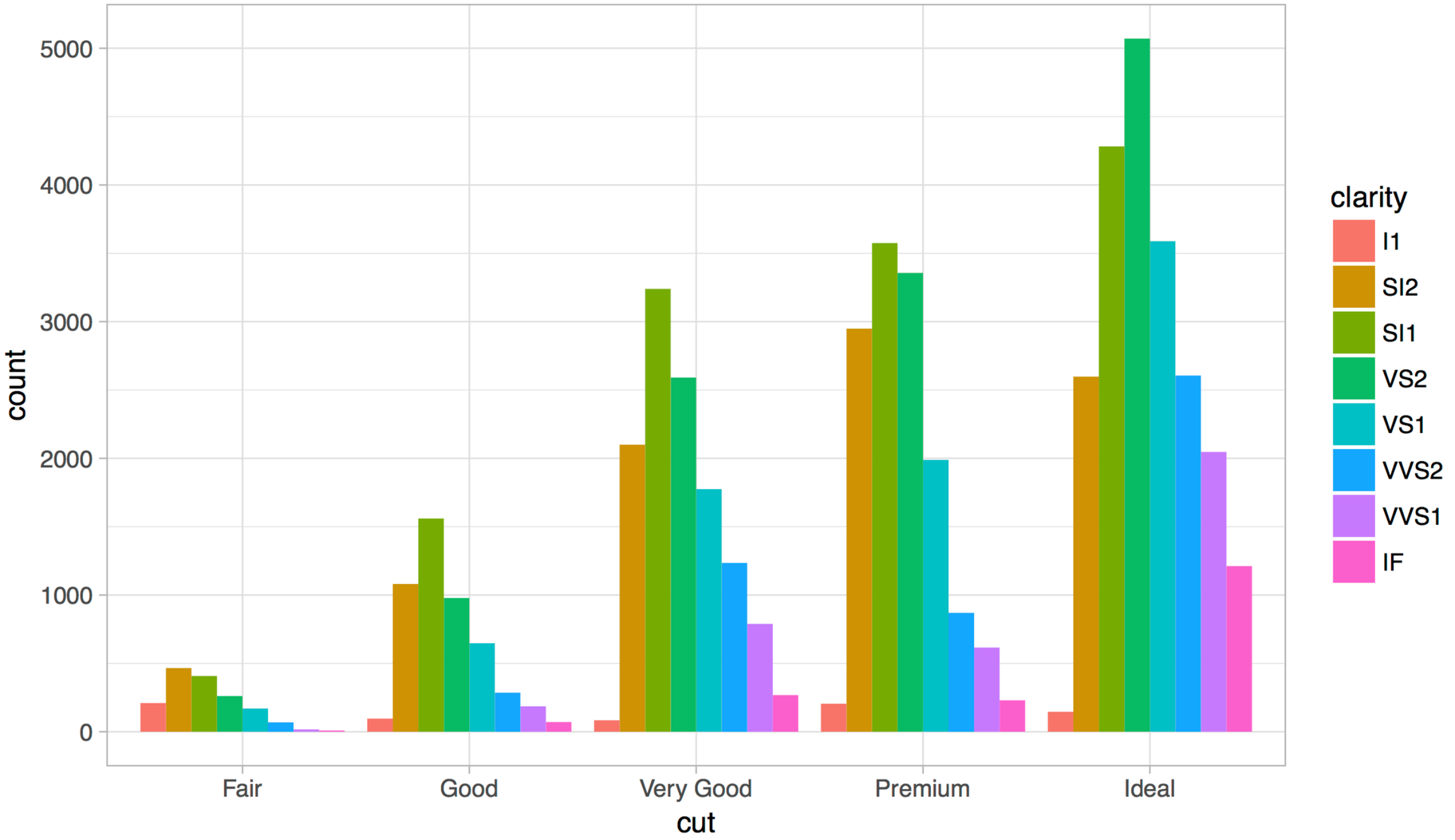
Aggiustare la posizione

Come sono disposti gli oggetti sovrapposti



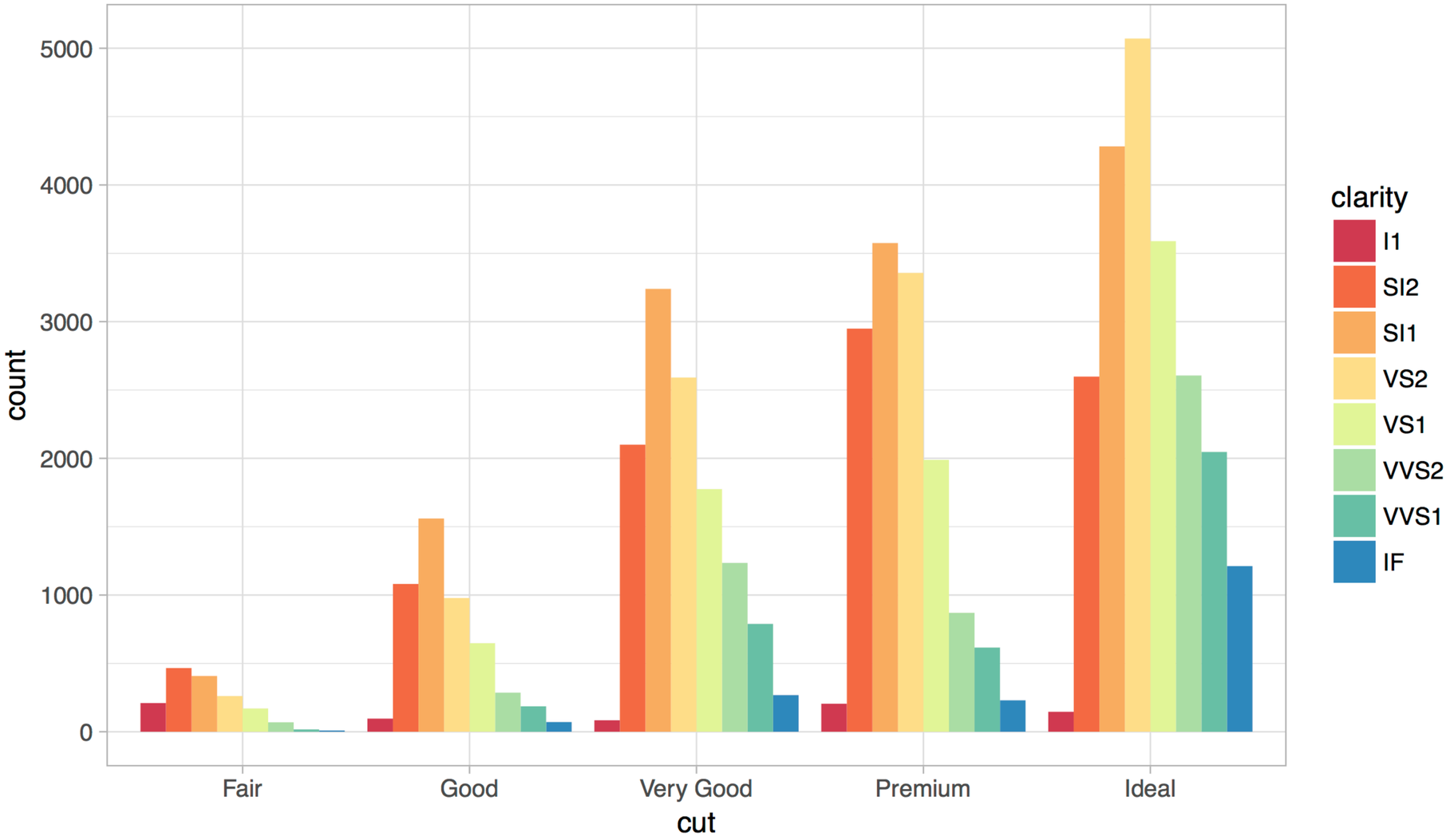
Themes

Aspetto visivo di elementi non di dati



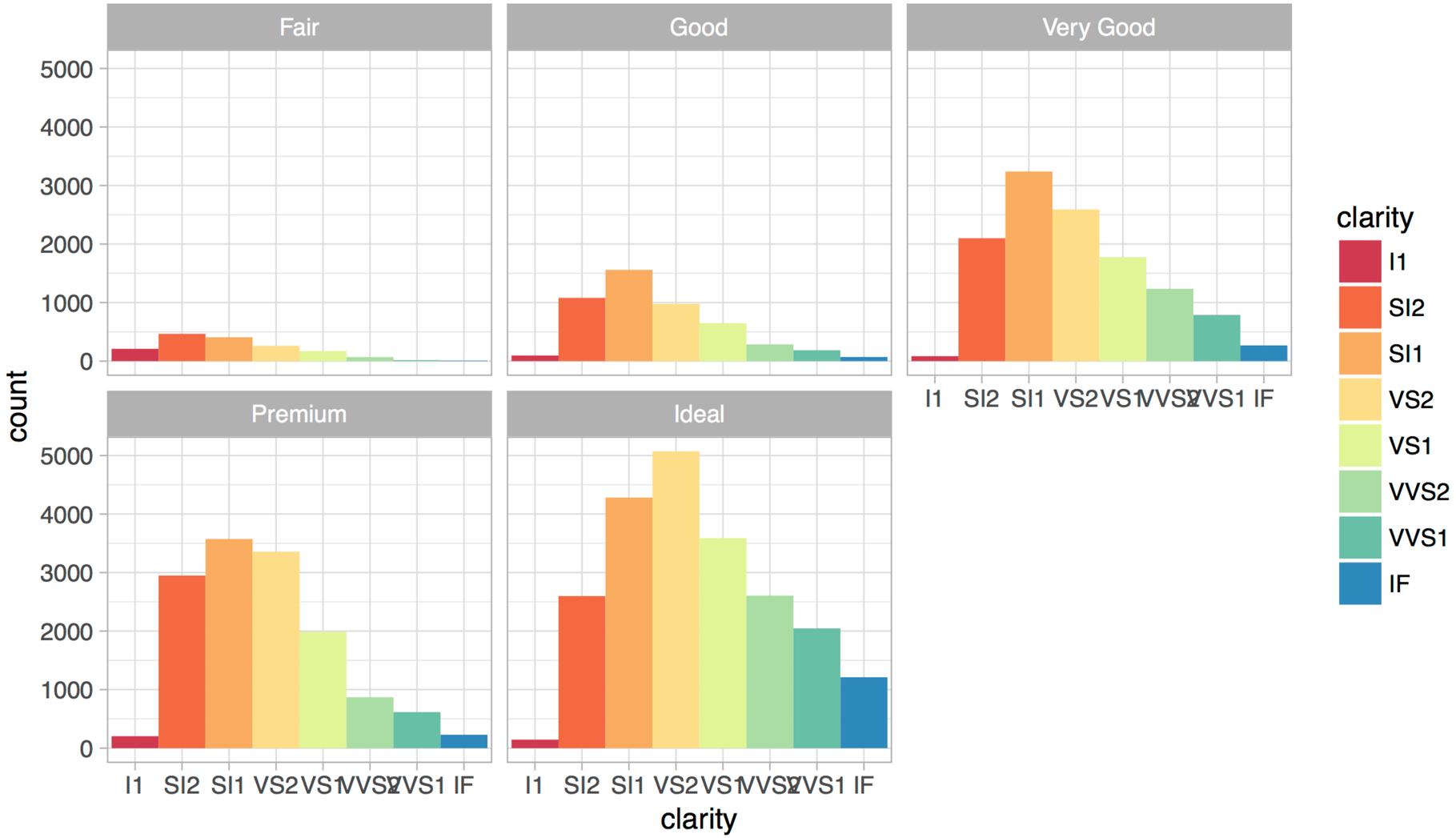
Scales

Personalizza le scale di colore, altre mappature

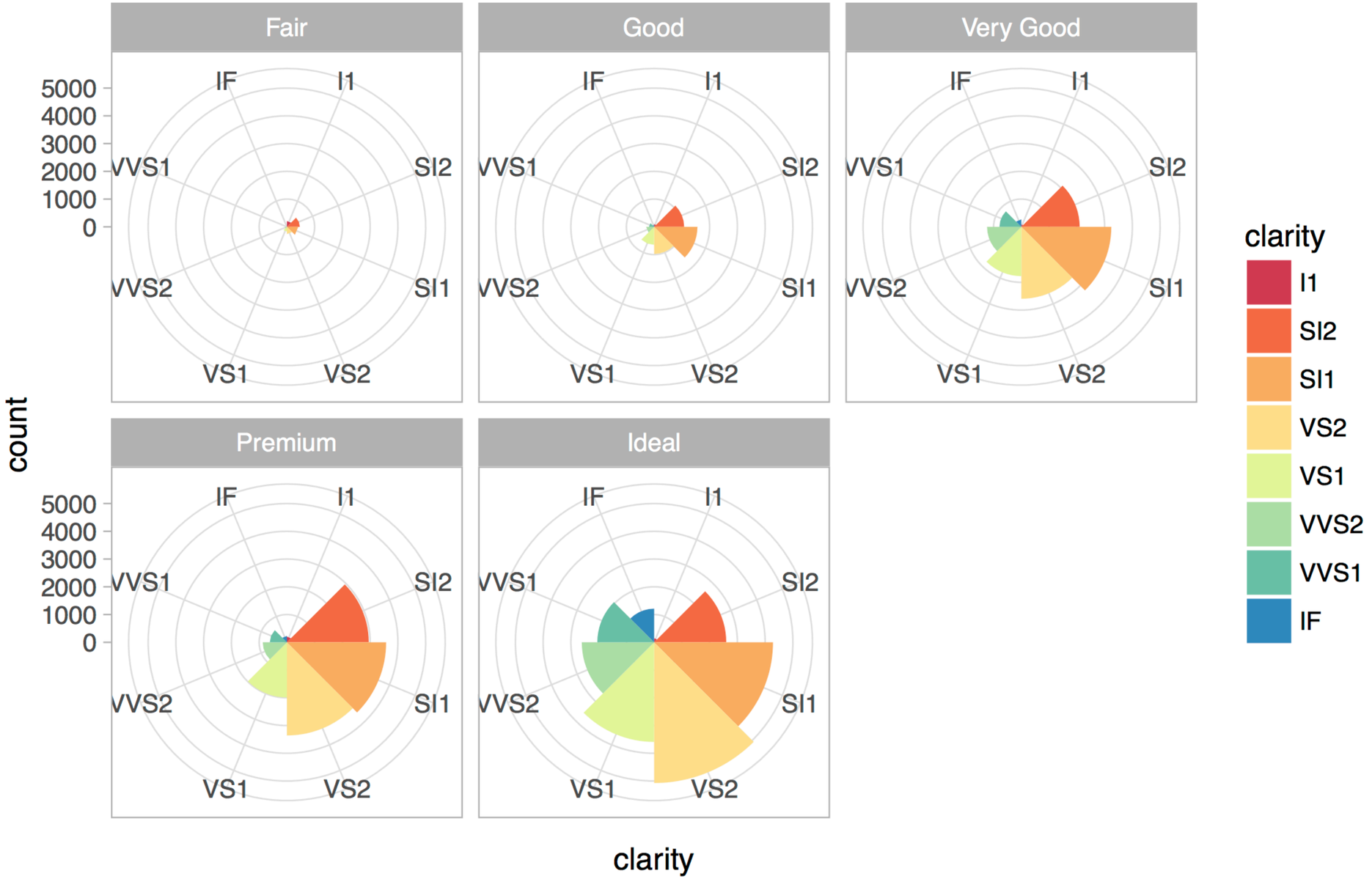


Facets

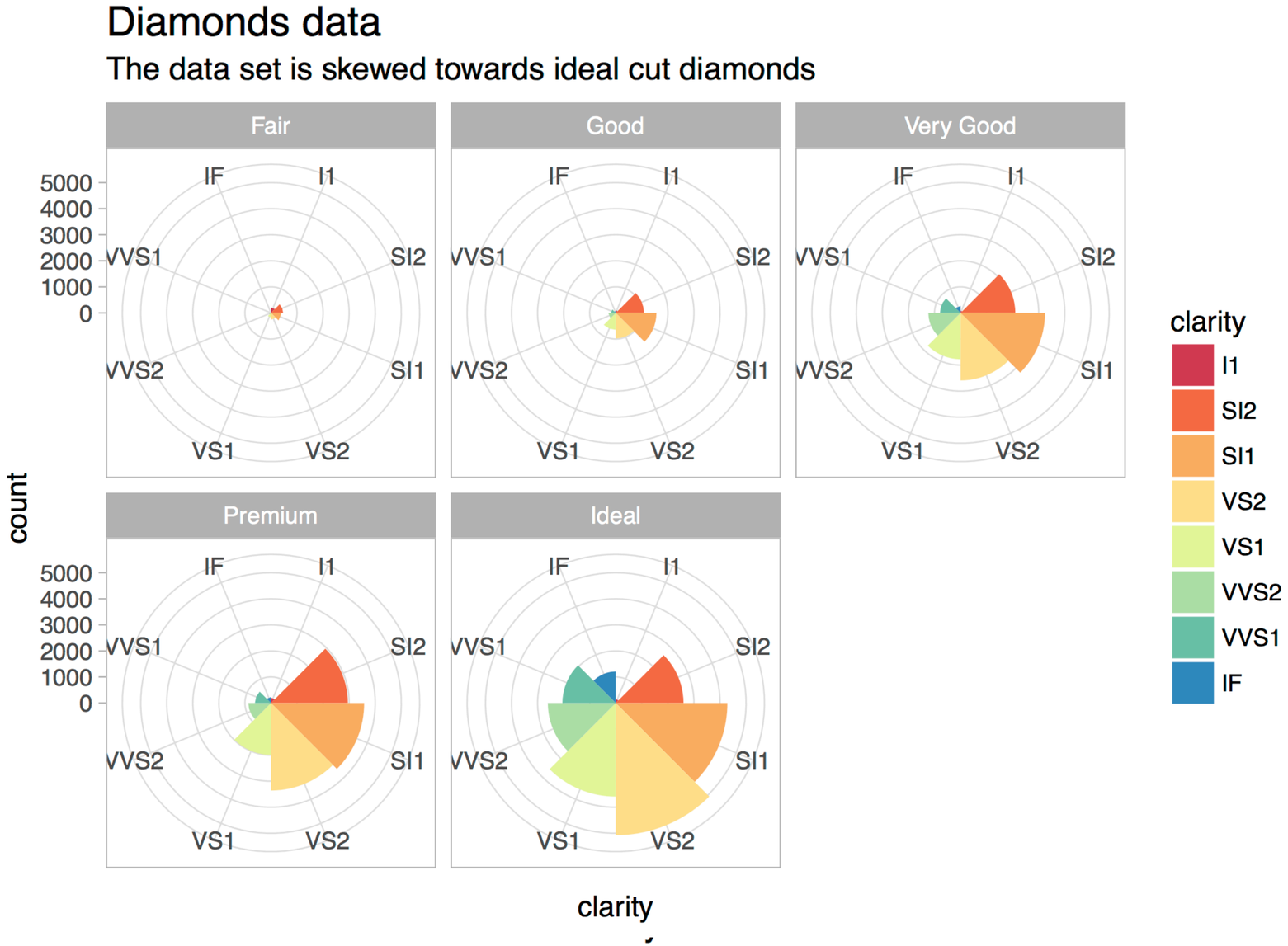
Sottotrame che visualizzano sottoinsiemi di dati.



Sistemi di coordinate



Titoli e sottotitoli



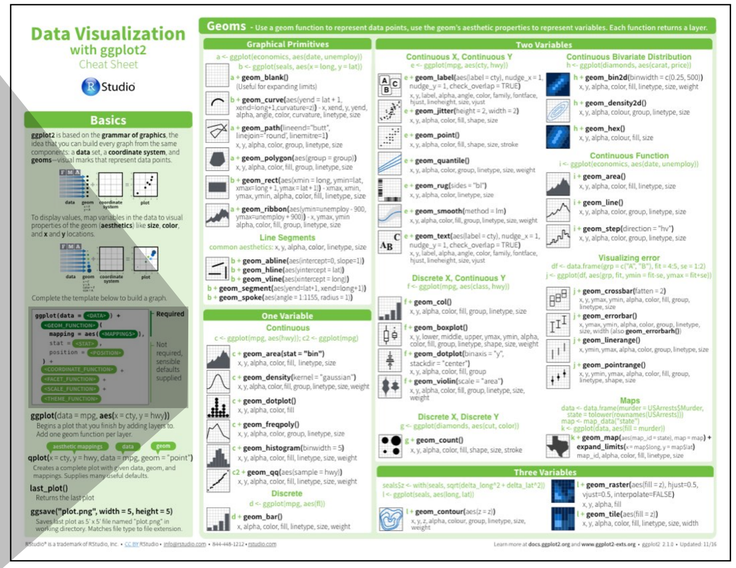
Il modello ggplot2

Crea qualsiasi grafico compilando i parametri di questo modello

```
ggplot(data = <DATA>) +
  <GEOM_FUNCTION>(
    mapping = aes(<MAPPINGS>),
    stat = <STAT>,
    position = <POSITION>
  ) +
  <COORDINATE_FUNCTION> +
  <FACET_FUNCTION> +
  <SCALE_FUNCTION> +
  <THEME_FUNCTION>
```

Required

Not required, sensible defaults supplied



ggplot2

ggplot2.tidyverse.org

Create Elegant Data Visualisati ×

Garrett

ggplot2.tidyverse.org

ggplot2 part of the tidyverse

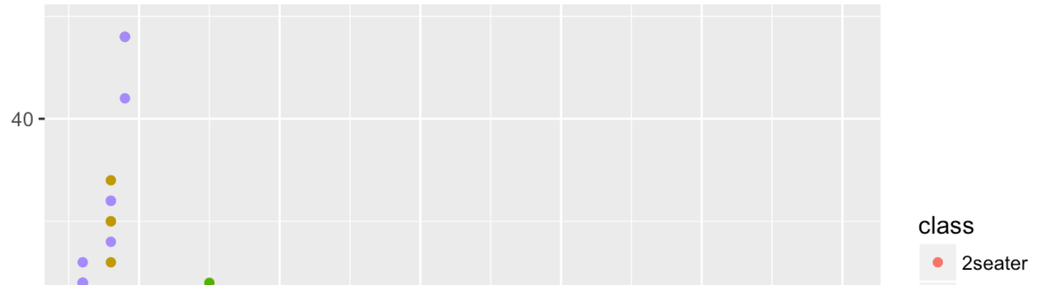
Reference Articles News

Usage

It's hard to succinctly describe how ggplot2 works because it embodies a deep philosophy of visualisation. However, in most cases you start with `ggplot()`, supply a dataset and aesthetic mapping (with `aes()`). You then add on layers (like `geom_point()` or `geom_histogram()`), scales (like `scale_colour_brewer()`), faceting specifications (like `facet_wrap()`) and coordinate systems (like `coord_flip()`).

```
library(ggplot2)

ggplot(mpg, aes(displ, hwy, colour = class)) +
  geom_point()
```



Links

Download from CRAN at https://cran.r-project.org/package=ggplot2

Browse source code at https://github.com/tidyverse/ggplot2

Report a bug at https://github.com/tidyverse/ggplot2/issues

Learn more at http://r4ds.had.co.nz/data-visualisation.html

License

GPL-2 | file LICENSE

Developers

Hadley Wickham

Author, maintainer

Visualize Data with