CUNEF
COLEGIO UNIVERSITARIO DE ESTUDIOS FINANCIEROS

ggplot2

ggplot2: Aesthetics y sintaxis

Conceptos clave:

- Aesthetics/mapping
  - Es el «mapeado» o rol que tiene cada variable visualmente.
    - La variable sexo está en el eje X
    - La variable altura está en el eje Y
    - La variable peso es el color
    - …
- Geoms
  - Son objetos geométricos o marcas
    - Barras, líneas, puntos…
- Statistics
  - Funciones agregadoras
    - Media, mediana, max…
- Scales
  - Las leyendas visuales
- Facets
  - Agrupaciones de los datos

ggplot2: Aesthetics y sintaxis

Dataset: mpg

	manufacturer	model	displ	year	cyl	trans	drv	cty	hwy	fl	class
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
11	audi	a4 quattro	2.0	2008	4	auto(s6)	4	19	27	p	compact
12	audi	a4 quattro	2.8	1999	6	auto(l5)	4	15	25	p	compact
13	audi	a4 quattro	2.8	1999	6	manual(m5)	4	17	25	p	compact

ggplot2: Aesthetics y sintaxis

La base para cualquier gráfico es la función ggplot.

```
ggplot(datos, aes)
```

En nuestro caso el dataset es «mpg». Los aesthetics o mappings se ponen dentro de la función aes:

```
ggplot(mpg, aes(…))
```

Por ejemplo…

```
ggplot(mpg, aes(class))
```

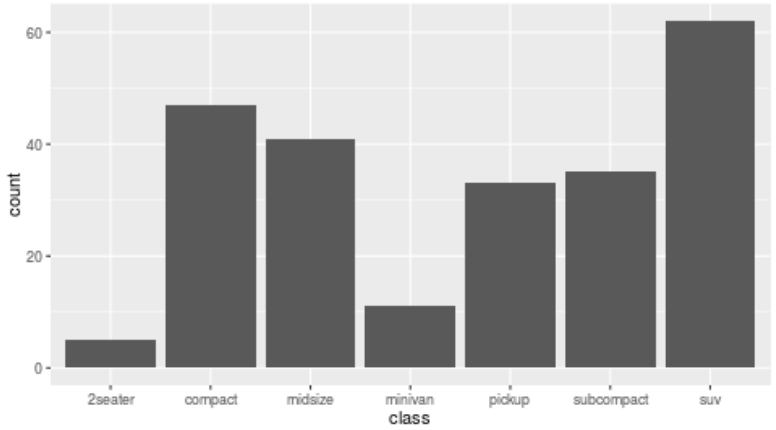
ggplot2: Aesthetics y sintaxis: Ejercicio 1

Una gráfica básica está compuesta por:

- Dataset
- Mappings
- Una geometría

Para añadir la geometría simplemente sumamos (+) la geom_ que corresponda.

```
ggplot(mpg, aes(class)) + geom_bar()
```



ggplot2: Aesthetics y sintaxis: Ejercicio 1

Hay muchas geometrías, probemos algunos (Utilizad Ctrl+Space)

```
ggplot(cars, aes(class)) + geom_...
```

Probemos a cambiar el ejercicio de geom_bar a geom_point…

Falla. ¿Por qué?

ggplot2: Aesthetics y sintaxis: Ejercicio 1

- Cada geometría acepta aes distintos.

- Por ejemplo, en un scatterplot (geom_point) puedes usar shape. En un diagrama de barras no puedes.

- Algunas geometrías comparten aes
  - P.e.: tanto geom_point como geom_bar tienen x

ggplot2: Aesthetics y sintaxis: Ejercicio 1

- Documentación de geom_bar
- Documentación de geom_point
- Mirad la sección Aesthetics:

geom_bar understands the following aesthetics (required aesthetics are in bold):

- **x**
- alpha
- colour
- fill
- linetype
- size

geom_point understands the following aesthetics (required aesthetics are in bold):

- **x**
- **y**
- alpha
- colour
- fill
- shape
- size
- stroke

ggplot2: Aesthetics y sintaxis: Ejercicio 1

- geom_bar funciona porque sólo requiere un mapping (x)
- Este mapping es x=class

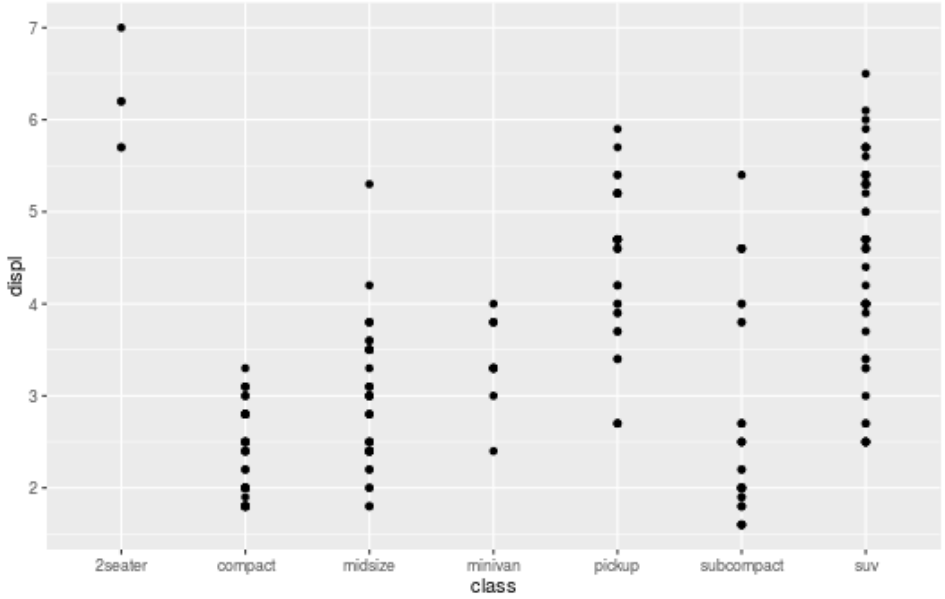
```
ggplot(cars, aes(class)) + geom_bar()
```

```
ggplot(cars, aes(x=class)) + geom_bar()
```

- En cambio geom_point requiere dos (x e y)

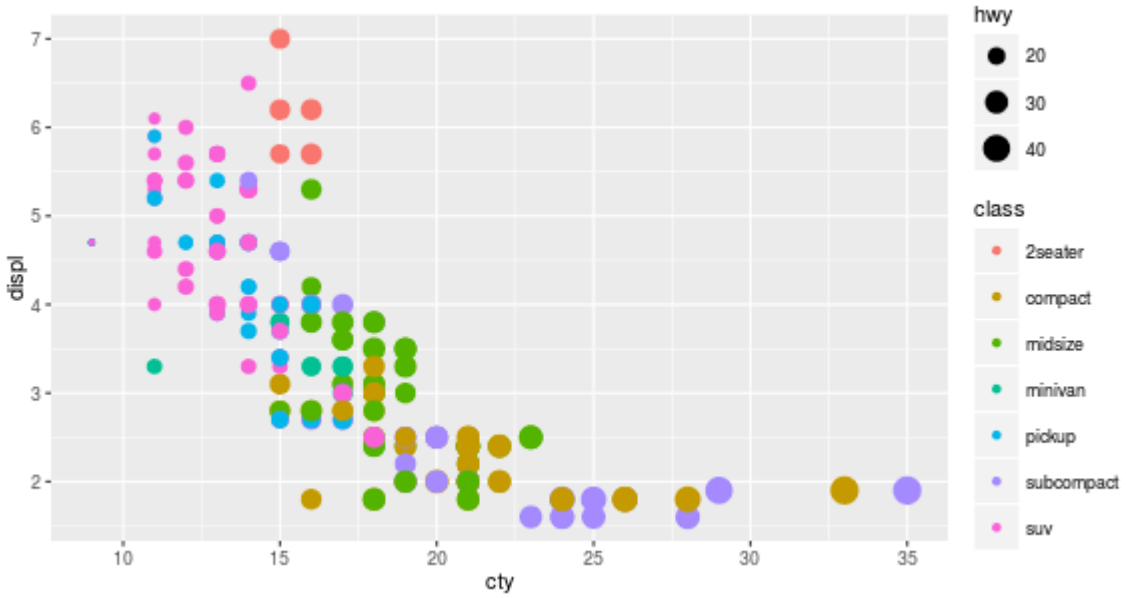
ggplot2: Aesthetics y sintaxis: Ejercicio 2

- Corregid el ejercicio anterior cambiando los mappings. Haced un geom_point con dos mappings:
  - En el eje X la variable class
  - En el eje Y la variable displ

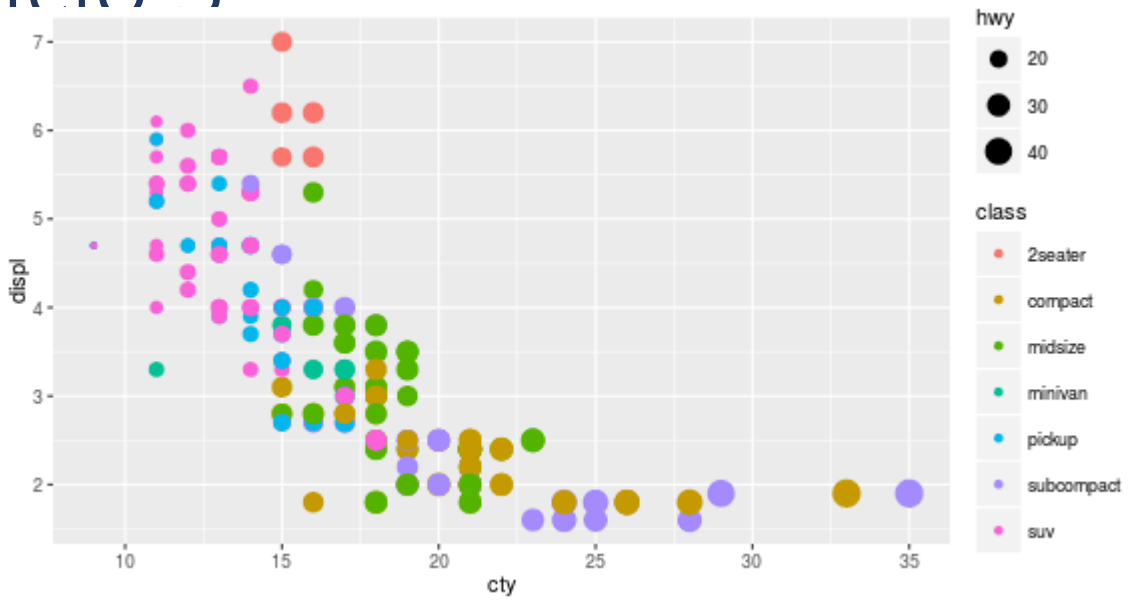


ggplot2: Aesthetics y sintaxis: Ejercicio 3

- Conseguid este gráfico:
  Mappings de: tamaño, color, x e y
  Pista: mirad la documentación de geom_point



ggplot2: Aesthetics y sintaxis: Ejercicio 3




```
ggplot(mpg,
       aes(x=cty, y=displ, colour=class, size=hwy)) +
  geom_point()
```

ggplot2: Aesthetics y sintaxis

- Puedes tener más de una geometría en una misma gráfica
  - Se ordenan en capas (layers). Siempre conservando en orden en el que tu las escribas. ¡Cuidado! Se van tapando unas a otras

- Esto permite crear una gran cantidad de gráficos avanzados

- Nunca penséis en los gráficos por su tipo.
  - Pensad en mappings y geoms. Esa es la potencia de grammar of graphics

ggplot2: Aesthetics y sintaxis

¿Cómo añadir múltiples capas?

```
ggplot(data,
       aes(x=variableuno)) +
  geom_bar() +
  geom_point() +
   . . .
```

¡Cuidado! Geom_point estará sobre geom_bar

ggplot2: Aesthetics y sintaxis

Los aesthetics son jerárquicos

```
ggplot(data,
       aes(x=variableuno)) +
  geom_bar() +
  geom_point(aes(y=variabledos))
```

Aesthetic global.
Todas las capas lo usarán

Aesthetic de la geometría.
Sólo esta capa lo utilizará

Los aes se pueden definir globales (en la función ggplot) o para cada geometría.

El «geom» busca los aes primero en el nivel más bajo y luego en el general.

La geometría point encuentra en aes x en el global.

Lo hemos hecho así porque si no geom_bar falla.

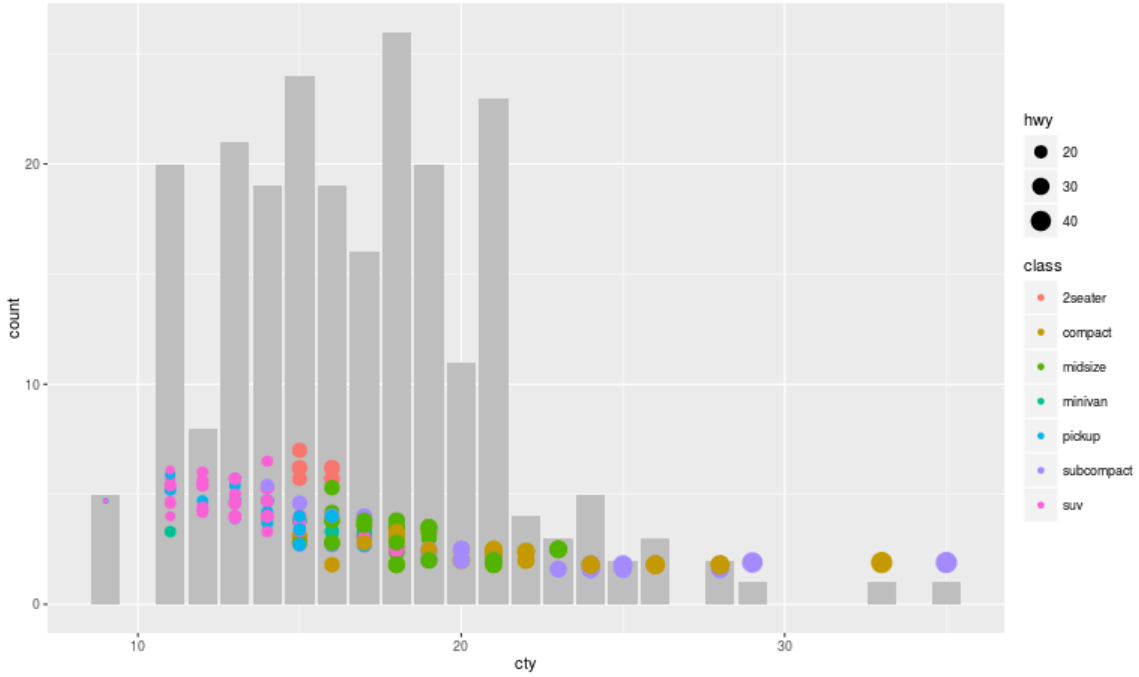
ggplot2: Aesthetics y sintaxis

- Puedes establecer constantes.
- ¡Cuidado! Ya no está dentro de aes, ya que no mapea una variable. Es un parámetro que se ha fijado

```
ggplot(datos,
       aes(x=variableuno)) +
  geom_bar(fill="grey")
```

ggplot2: Aesthetics y sintaxis: Ejercicio 4

Obtén este gráfico:



Pistas

El mismo mapping que el anterior ejercicio

Hemos añadido una capa más (fijaos en el orden)

El ejercicio debe resolverse con la menor cantidad de código. **Utilizad los mappings globales cuando sea posible.**

ggplot2: Aesthetics y sintaxis: Ejercicio 4

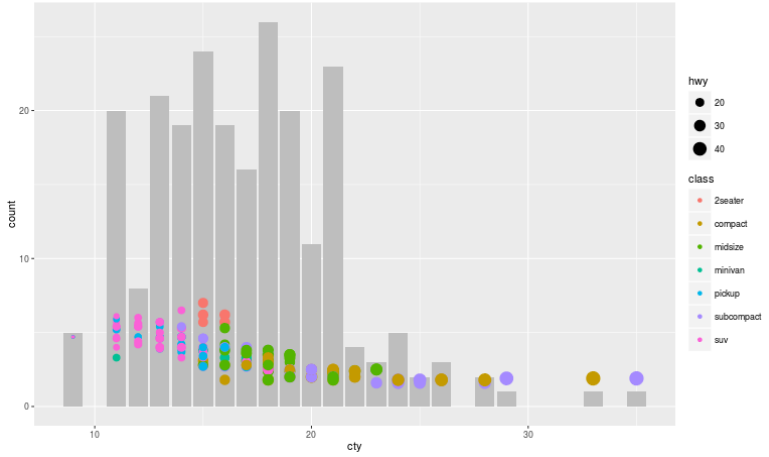
Obtén este gráfico:

```
ggplot(mpg,
       aes(x=cty)) +
  geom_bar(fill="grey") +
  geom_point(aes(y=displ,
colour=class, size=hwy))
```

El único mapping común entre ambas capas es la x.

Primera capa.
El fill es constante.
Está fuera de aes()

Segunda capa.
Añadimos el resto de aes



ggplot2: Aesthetics y sintaxis: Ejercicio 5

Evitar over plotting:

Apartado a) Utilizando opacity (alpha) de 0.2

Apartado b) Utilizando Jittering

ggplot2: Aesthetics y sintaxis: Ejercicio 5

Apartado a) Utilizando opacity (alpha) de 0.2

```
ggplot(mpg,
       aes(x=cty)) +
  geom_bar(fill="grey") +
  geom_point(aes(y=displ, colour=class, size=hwy),
alpha=0.3)
```

Apartado b) Utilizando Jittering

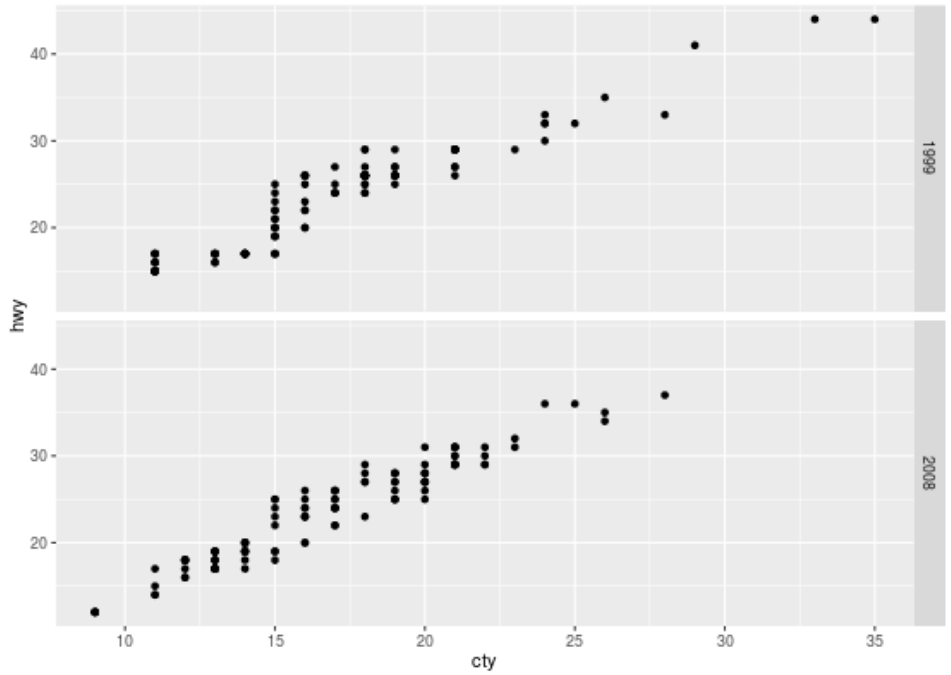
```
ggplot(mpg,
       aes(x=cty)) +
  geom_bar(fill="grey") +
  geom_jitter(aes(y=displ,
colour=class, size=hwy))
```

```
ggplot(mpg,
       aes(x=cty)) +
  geom_bar(fill="grey") +
  geom_point(aes(y=displ,
colour=class, size=hwy),
position="jitter")
```

ggplot2: Facets

¿Qué es un facet?

Es una agrupación de los datos que se refleja visualmente mediante separaciones y divisiones del eje.



ggplot2: Facets

¿Qué es un facet?

```
ggplot(datos, aes(…)) +
  geom_bar() +
  facet_grid(variable1~variable2)
```

Facet_grid hace divisiones de los ejes de forma ortogonal.

De manera que puedes dividir por dos variables.

Variable1 divide en filas

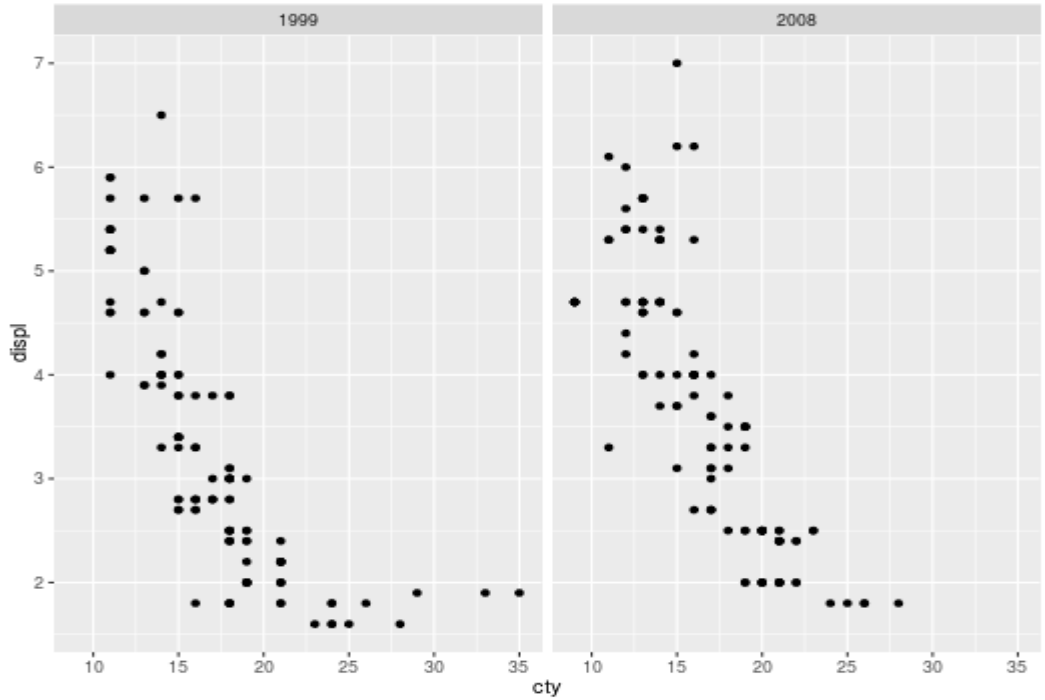
Variable2 divide en columnas

Si sólo quieres dividir por una variable, sustituye una por un punto:

```
ggplot(datos, aes(…)) +
  geom_bar() +
  facet_grid(.~variable2)
```

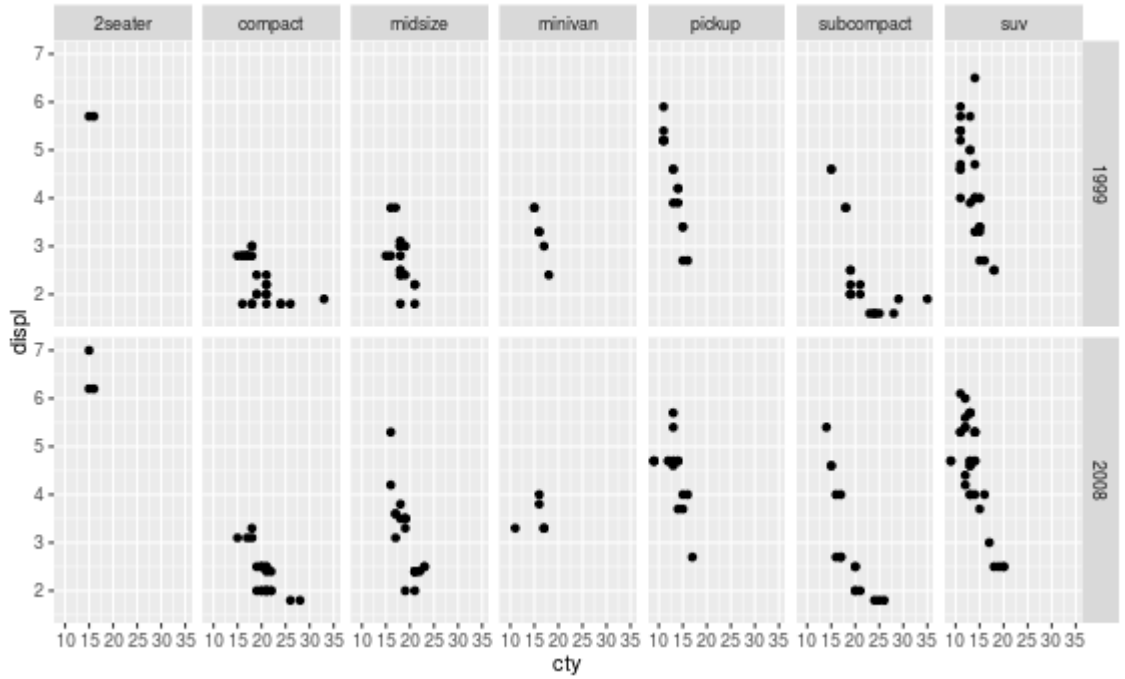
ggplot2: Facets : Ejercicio 6

a) Realiza un scatterplot de cty x displ. Dividelo por columnas por el año



ggplot2: Facets : Ejercicio 6

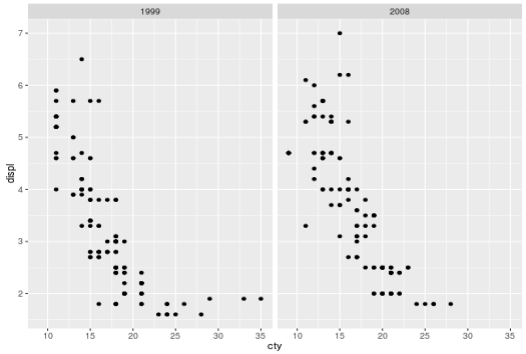
b) Realiza un scatterplot de cty x displ. Dividelo por filas por year y columnas por class



ggplot2: Facets : Ejercicio 6

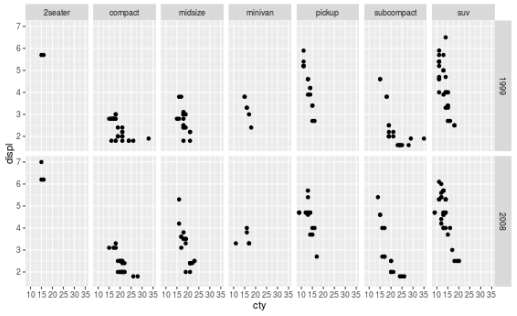
a)

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



b)

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



ggplot2: Facets: Cuándo usarlos

Los facets son útiles cuando quieres visualizar una misma interacción a lo largo de distintos grupos

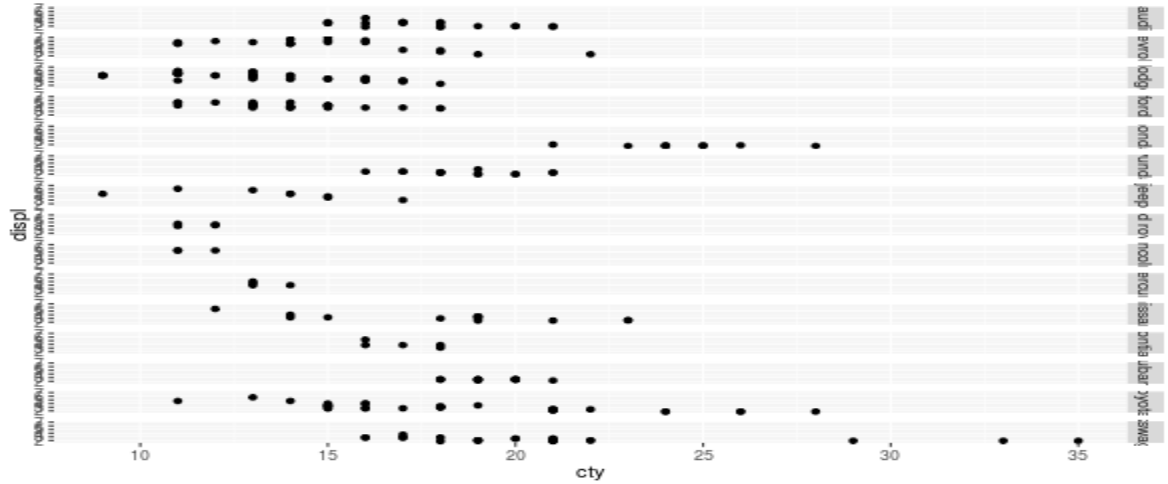
Ejs:

- ¿La relación entre estos medicamentos es igual para todos los grupos de edad?
- ¿La criminalidad de estos distritos ha cambiado a lo largo de los años?
- ¿La correlación entre potencia y cilindrada es igual para cada fabricante de coches?

ggplot2: Facets: Ejercicio 7

Cuando hay demasiadas categorías se desperdicia espacio.

Por ejemplo, vamos a hacer el mismo scatterplot haciendo facet por filas por la variable manufacturer



ggplot2: Facets: Ejercicio 7

Perdemos totalmente la capacidad de visualizar (no hay espacio suficiente) y desperdiciamos el eje Y

Para esto existe facet_wrap

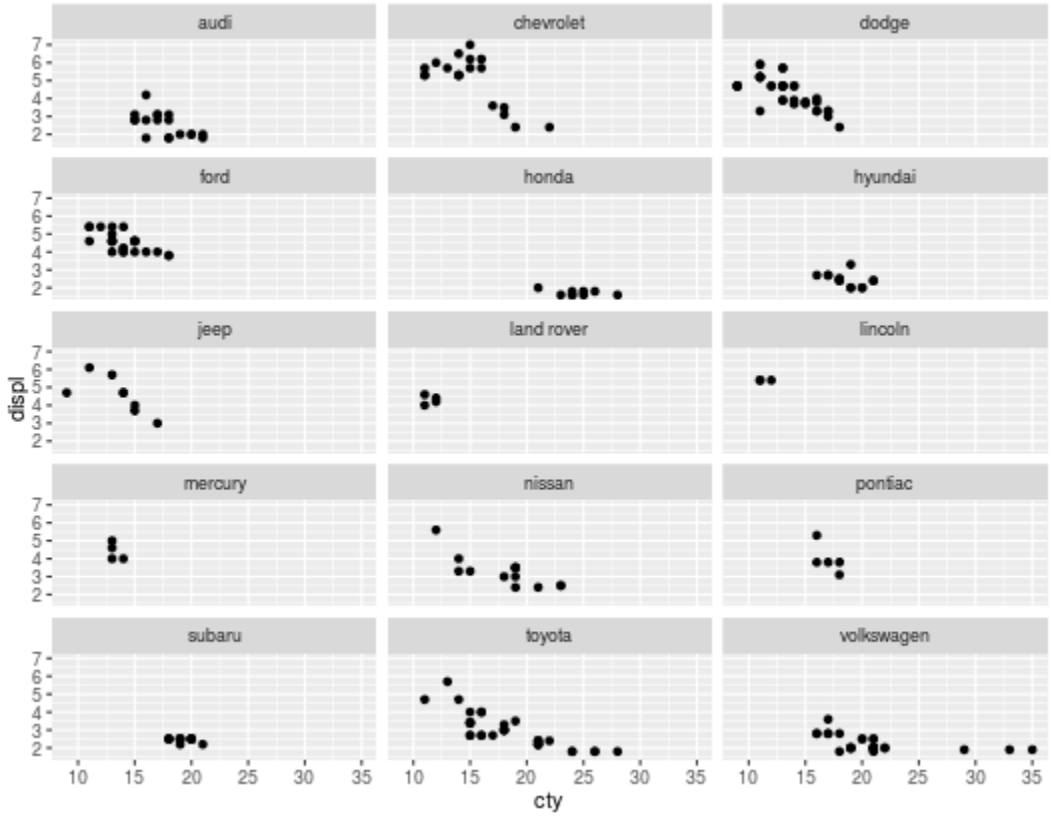
```
ggplot(datos,
       aes(…) +
  geom_point() +
  facet_wrap(~variable, ncol=3)
```

Limitas el número de columnas (ncol) o filas (nrow)

¡Cuidado!
Siempre hay que poner la dieresis por delante sin punto

ggplot2: Facets: Ejercicio 7

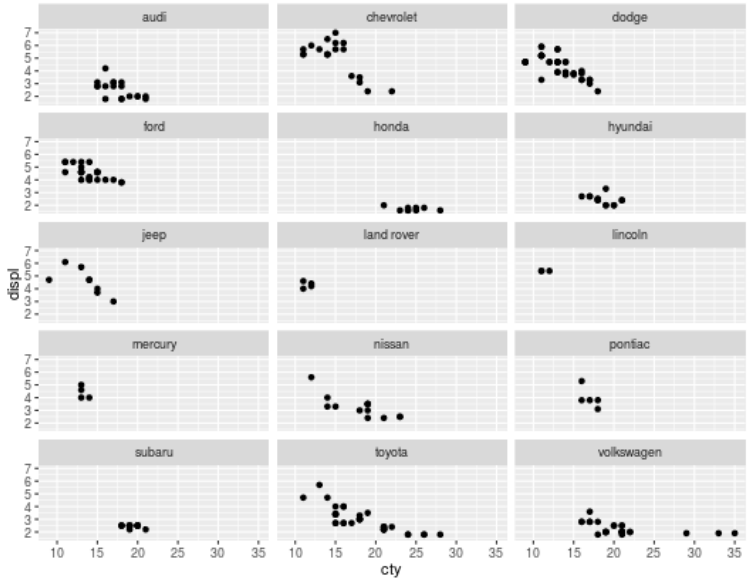
Obtened este gráfico (número de columnas limitado a 3)



ggplot2: Facets: Ejercicio 7

Obtened este gráfico (número de columnas limitado a 3)

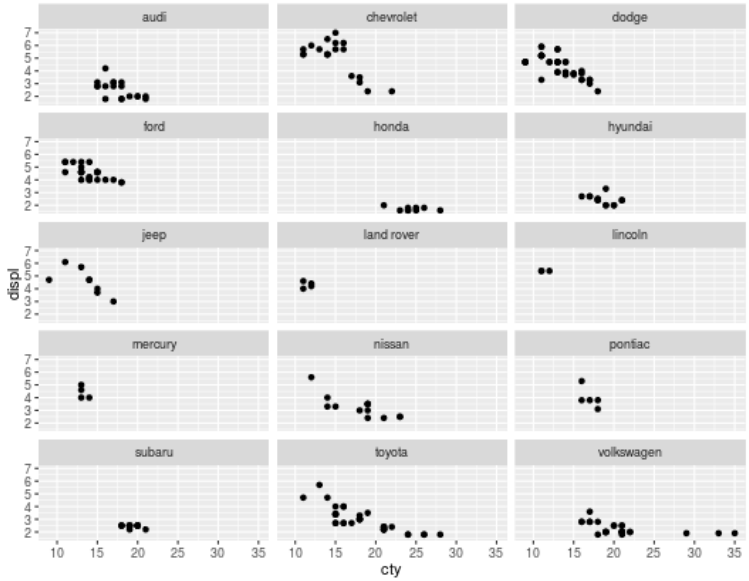
```
ggplot(mpg,
       aes(x=cty, y=displ)) +
  geom_point() +
  facet_wrap(~manufacturer,
ncol=3))
```



ggplot2: Facets: Ejercicio 7

Obtened este gráfico (número de columnas limitado a 3)

```
ggplot(mpg,
       aes(x=cty, y=displ)) +
  geom_point() +
  facet_wrap(~manufacturer,
ncol=3))
```



ggplot2: Estilado

El cambio de estilo en ggplot se hace mediante themes ya predefinidos.

- theme_grey
- theme_bw
- theme_linedraw
- theme_light
- theme_minimal
- theme_classic
- theme_dark

Hay paquetes de terceros que tienen otros temas (p.e: ggthemes)

ggplot2: Estilado

Un estilo se aplica simplemente sumando el tema a la gráfica

```
ggplot(datos,
       aes(…)) +
  geom_point() +
  theme_bw()
```

ggplot2: Estilado: Ejercicio 8

Aplicar al scatterplot cty x displ x manufacturer (ejericio 7) un estilo distinto.

ggplot2: Estilado: Personalización

ggplot2 permite muchísima personalización del diseño mediante la función theme.

Documentación oficial
Vignette (Tutorial)

```
theme_bw() +
  theme(axis.text = element_text(size = 14),
    legend.key = element_rect(fill = "navy"),
    legend.background = element_rect(fill = "white"),
    legend.position = c(0.14, 0.80),
    panel.grid.major = element_line(colour = "grey40"),
    panel.grid.minor = element_blank()
  )
```

ggplot2: Estadísticas

ggplot2 es una herramienta pensada para el trabajo diario de un Data Scientist o un investigador.

Por ello también permite hacer estadísticas básicas.

Algunos de los más importantes son:

```
+ stat_smooth()
+ geom_density2d()
+ geom_hex()
```

ggplot2: Estadísticas: Ejercicio 9

Haz un scatterplot cty x displ y…

a) Añade una capa de geom_density2d

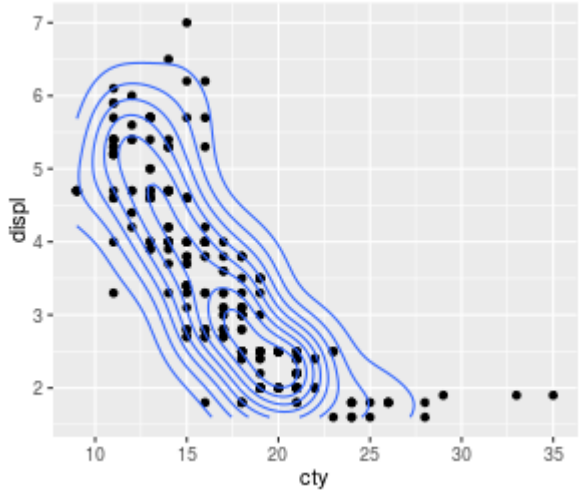
b) Añade una capa de geom_hex

ggplot2: Estadísticas: Ejercicio 9

Haz un scatterplot cty x displ y…

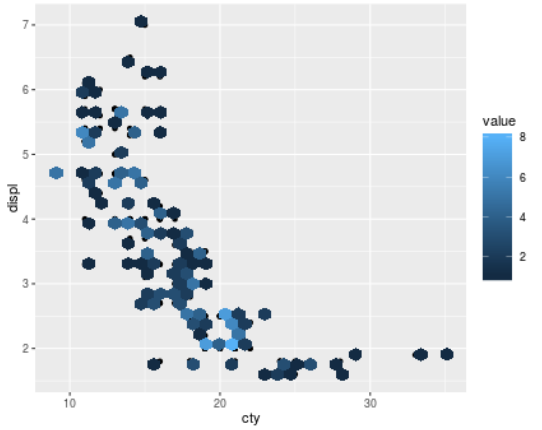
a) Añade una capa de geom_density2d

```
ggplot(mpg,
       aes(x=cty, y=displ)) +
  geom_point() +
  geom_density2d()
```



b) Añade una capa de geom_hex

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

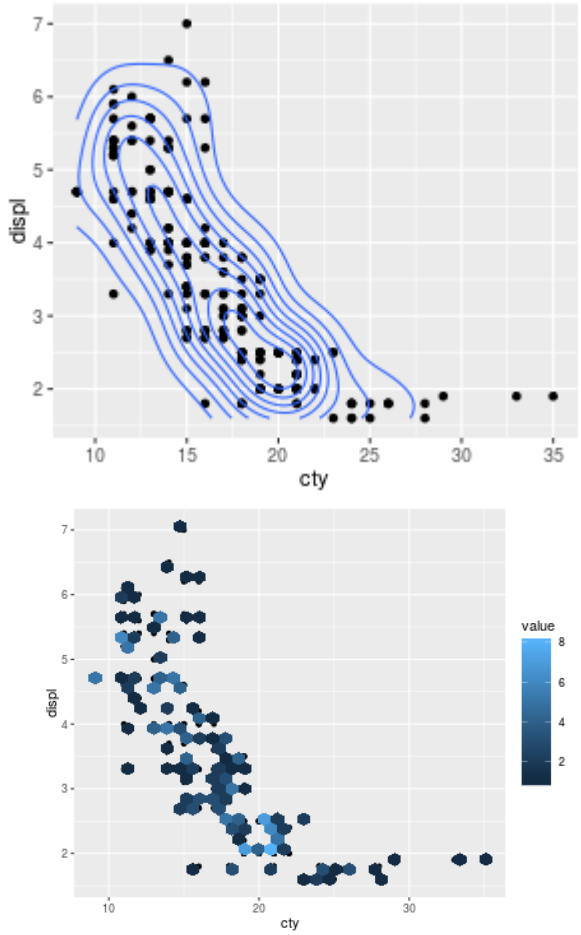


ggplot2: Estadísticas: Ejercicio 9

Haz un scatterplot cty x displ y…

Aunque no se aprecie a primera vista ambas operaciones han hecho agregaciones y cálculos estadísticos.

- Suavizado de las líneas
- Binarización e Histograma 2d
- Cálculo de la rejilla hexagonal



ggplot2: Estadísticas

Otra estadística muy útil y frecuente en ggplot2 es stat_smoth.

Permite realizar suavizados

```
+ stat_smooth()
```

O modelos

```
+ stat_smooth(method = "lm")


+ stat_smooth(method = "lm")
```