Mejorando los Gráficos con ggplot2

Antonio Miñarro
aminarro@ub.edu

UNIVERSITAT DE BARCELONA

Departament de Genètica, Microbiologia i Estadística

15/06/2021

Esquema del tema

Esquema del tema

Esquema del tema

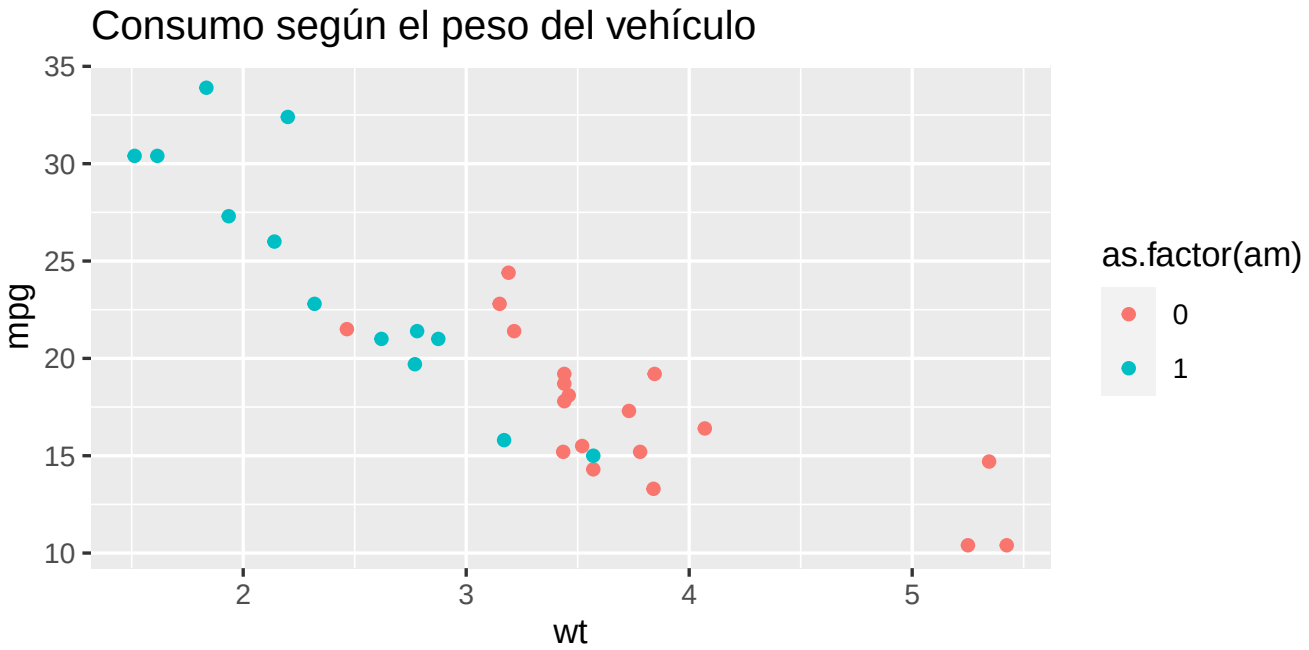
Esquema del tema

Esquema del tema

Etiquetas

Podemos añadir etiquetas con la función *labs()*.

```
> ggplot(mtcars,aes(wt,mpg))+
+  geom_point(aes(color=as.factor(am)))+
+  labs(title='Consumo según el peso del vehículo')
```



Etiquetas (2)

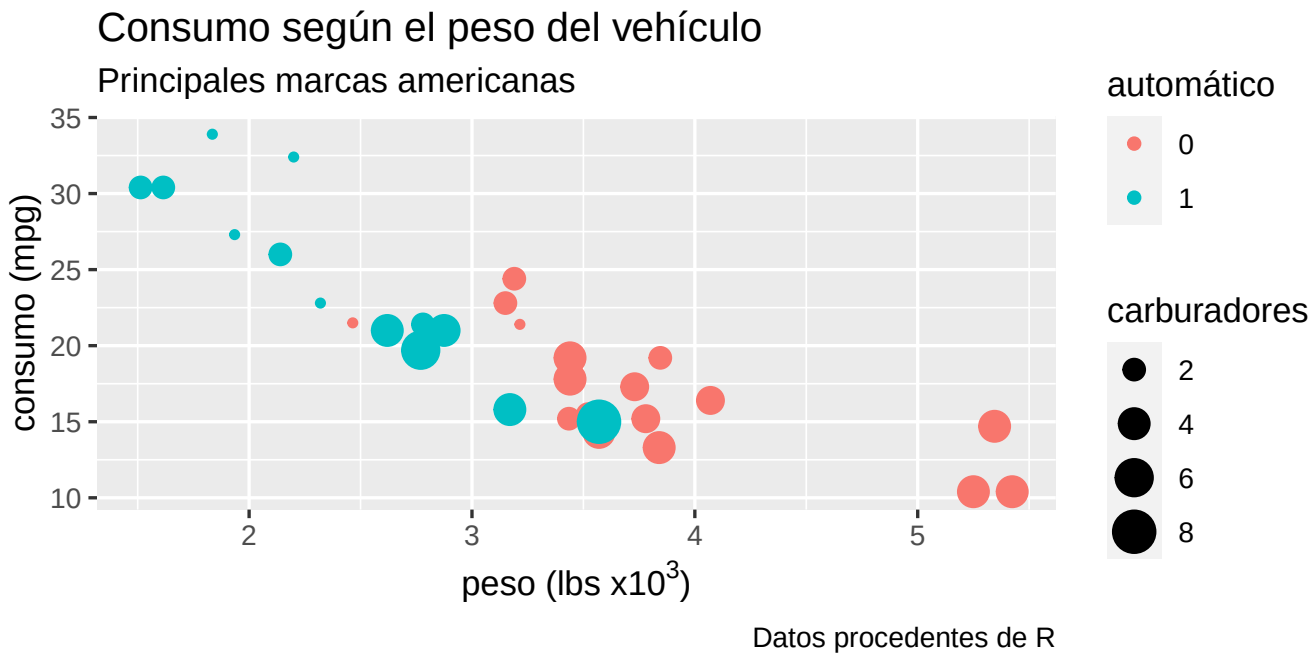
Podemos añadir las siguientes etiquetas:

1. title: título del gráfico
2. subtitle: subtítulo (fuente menor bajo el título)
3. caption: texto en la parte inferior derecha
4. x: título del eje X
5. y: título del eje Y
6. colour/fill/...: títulos de las leyendas creadas por cada propiedad

Es posible añadir expresiones matemáticas en las etiquetas con la función *expression()*.

Etiquetas (3)

```
> ggplot(mtcars,aes(wt,mpg))+
+  geom_point(aes(color=as.factor(am),size=carb))+
+  labs(title='Consumo según el peso del vehículo',
+       subtitle='Principales marcas americanas',
+       caption='Datos procedentes de R',
+       x=expression(paste('peso (lbs x', 10^3,')')),
+       y='consumo (mpg)',
+       color='automático',
+       size='carburadores'
+  )
```

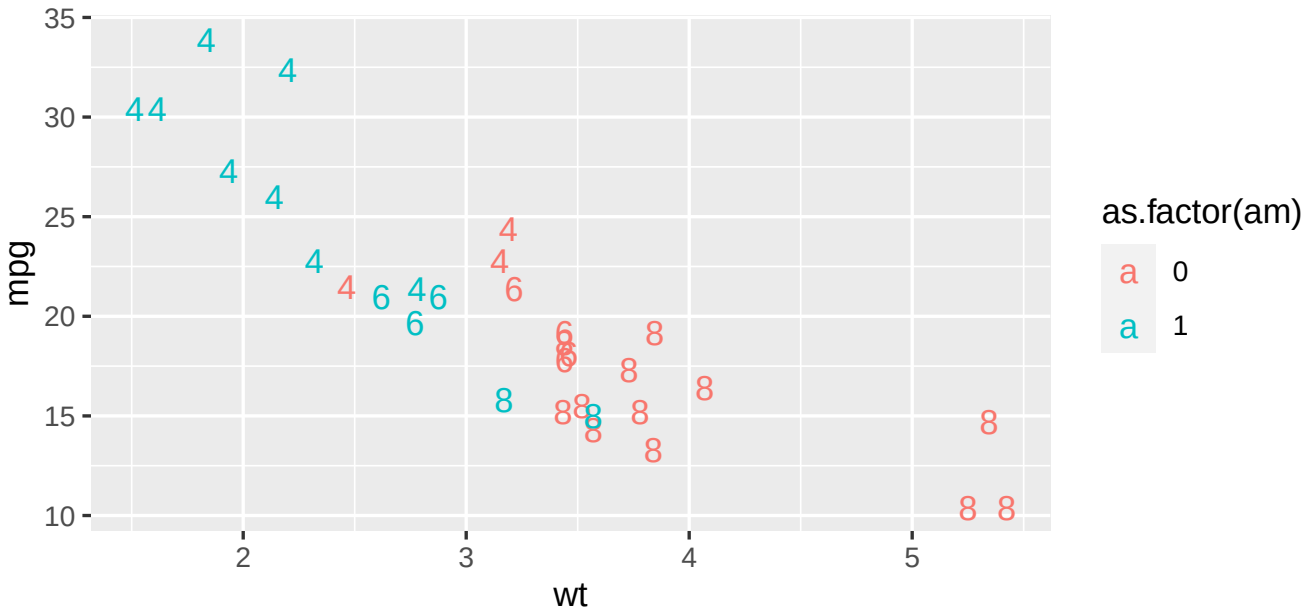


Esquema del tema

Anotación de observaciones individuales

Podemos utilizar diferentes geoms para añadir etiquetas o las observaciones: geom_text(), geom_label()

```
> ggplot(mtcars,aes(wt,mpg))+
+   geom_text(aes(label=cyl,color=as.factor(am)))+
+  labs(caption='Los números representan el número de cilindros')
```

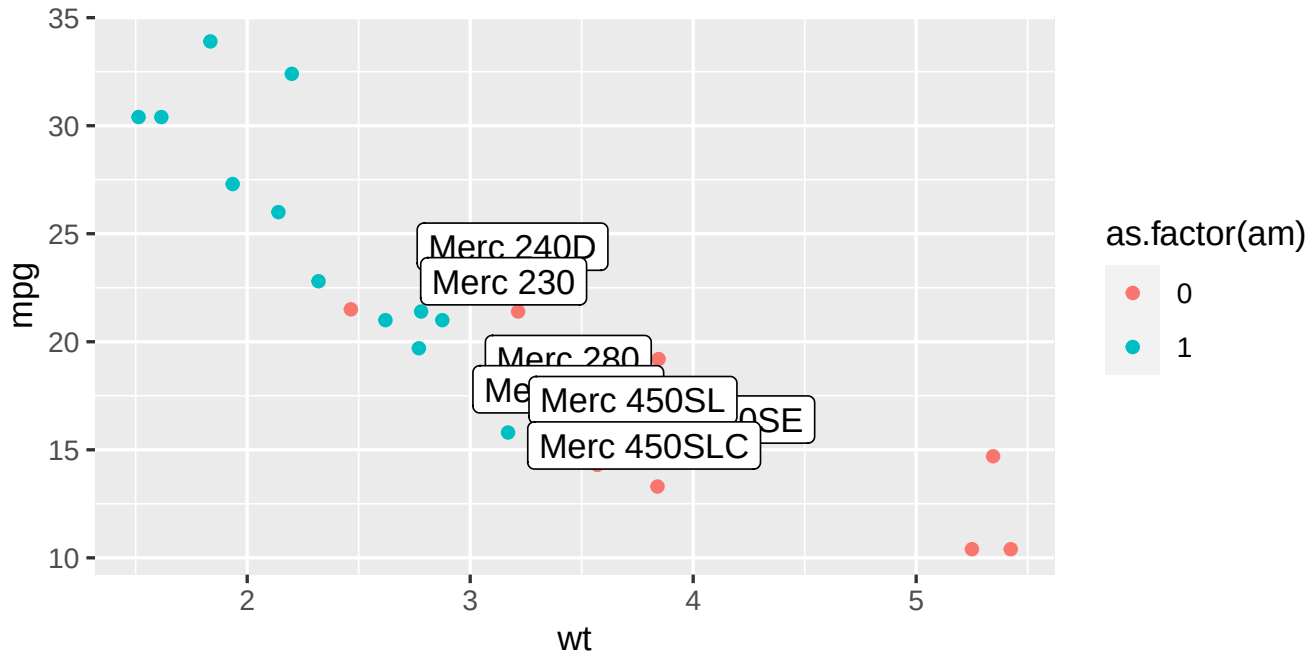


Los números representan el número de cilindros

Anotaciones (2)

Creamos un subconjunto de los datos con los vehículos Mercedes.

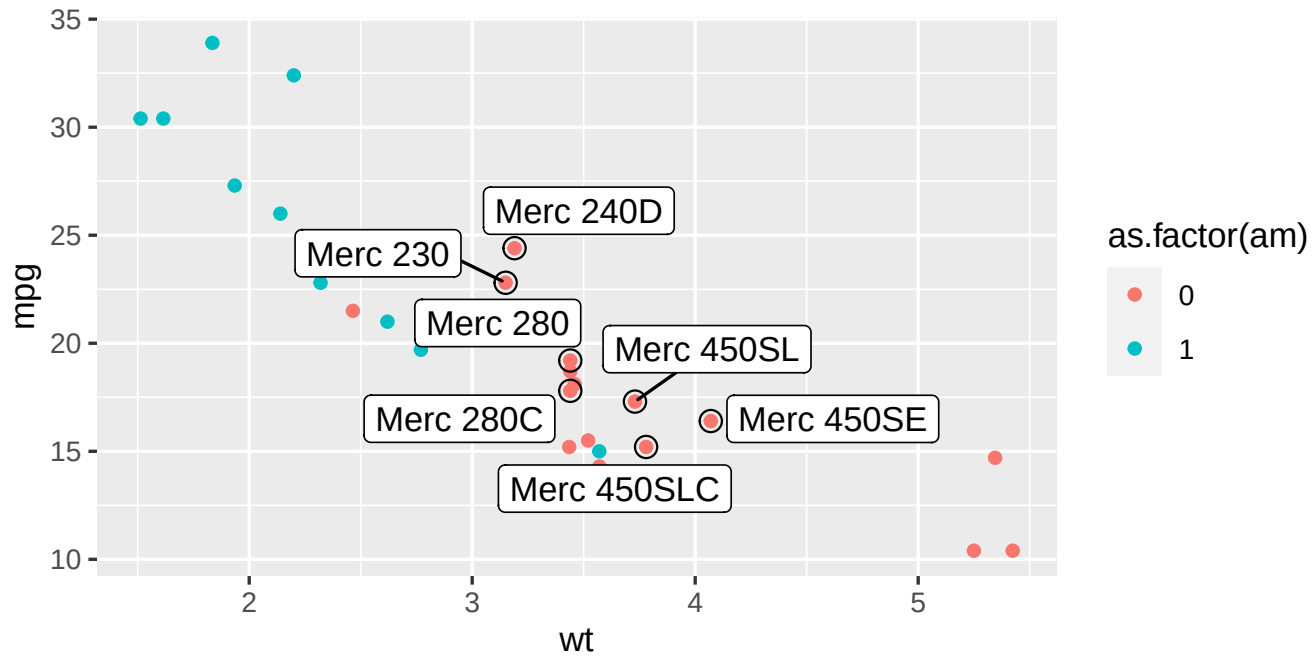
```
> sub.mtcars<-subset(mtcars,substr(rownames(mtcars),1,3)=='Mer')
> ggplot(mtcars,aes(wt,mpg))+
+  geom_point(aes(color=as.factor(am)))+
+  geom_label(aes(label=rownames(sub.mtcars)),data=sub.mtcars)
```



Anotaciones (3)

Veremos dos novedades, primero la utilización del package ggrepel, que supone una extensión de las capacidades de ggplot2; segundo la superposición de dos geom_point() para destacar los puntos del segundo data frame.

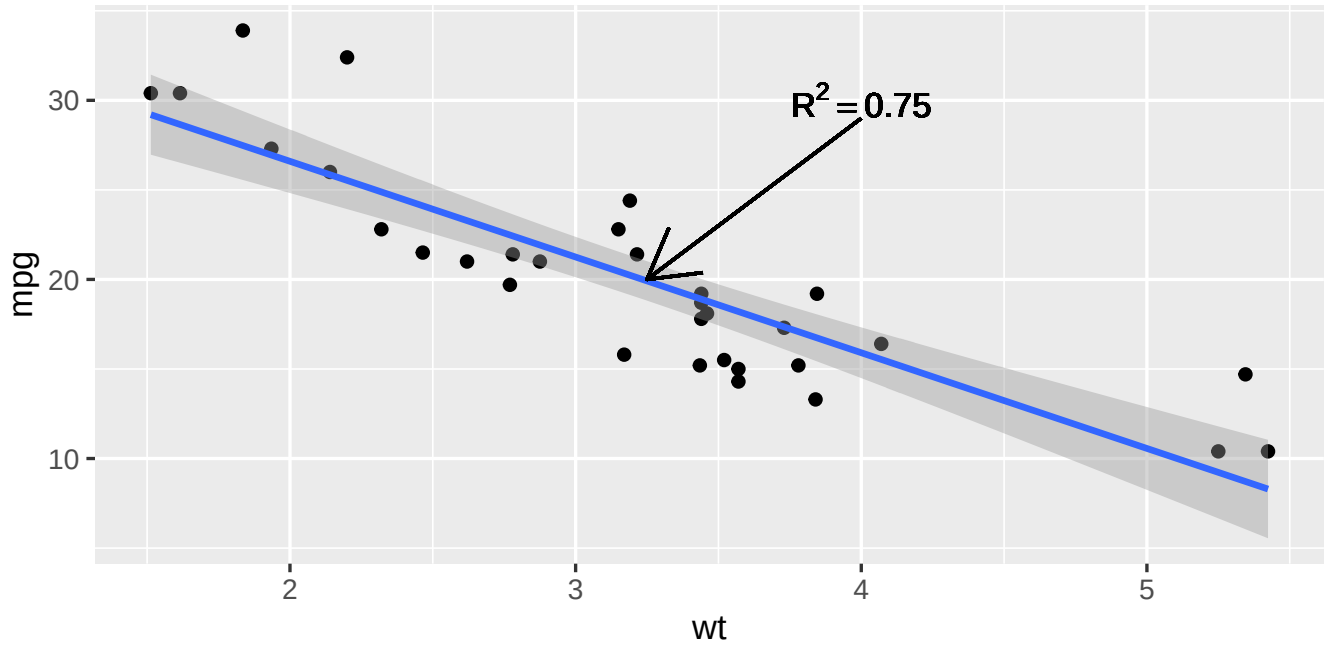
```
> ggplot(mtcars,aes(wt,mpg))+
+  geom_point(aes(color=as.factor(am)))+
+  geom_point(size=3,shape=1,data=sub.mtcars)+
+  ggrepel::geom_label_repel(aes(label=rownames(sub.mtcars))
+                            ,data=sub.mtcars)
```



Anotaciones (4)

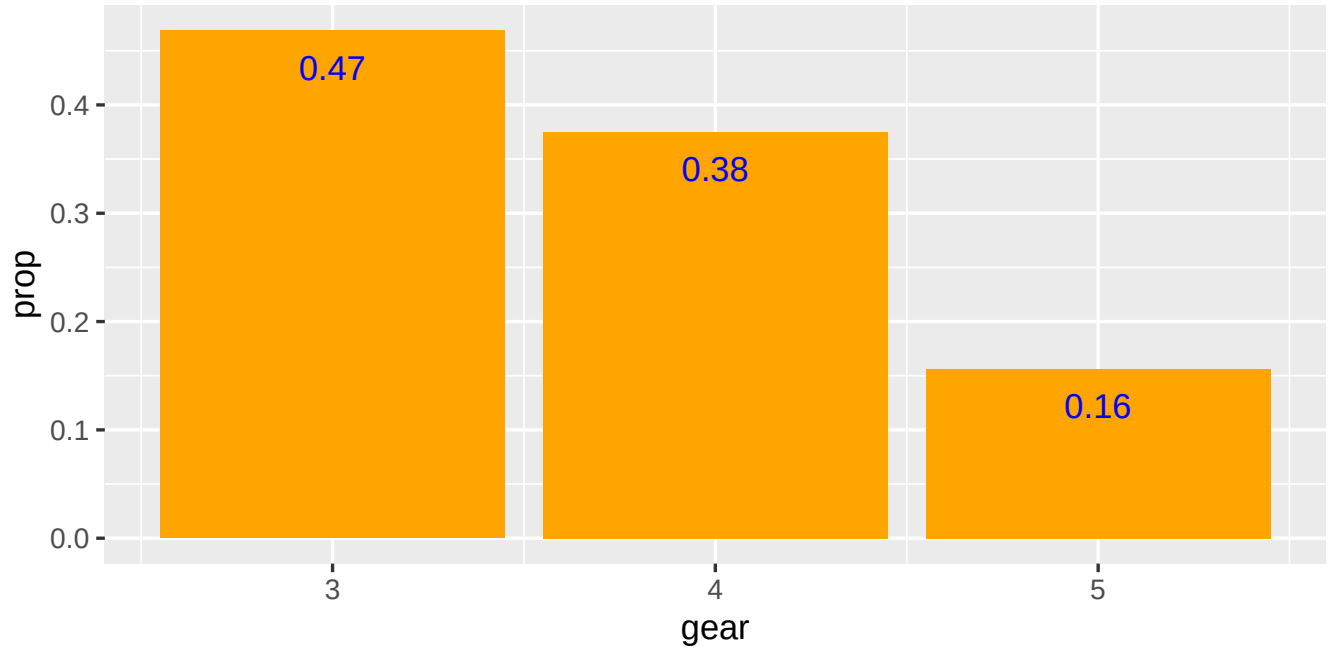
Con la geom_text() podemos también añadir texto en cualquier parte de un gráfico.

```
> p<-ggplot(mtcars,aes(x=wt,y=mpg))+geom_point()+
+  geom_smooth(method='lm')
> r2<-round(summary(lm(mpg~wt,mtcars))$r.squared,2)
> label<-paste('R^2==',r2)
> p+geom_text(x=4,y=30,label=label,parse=T)+
+  geom_segment(x=4,y=29,xend=3.25,yend=20,arrow=arrow())
```



Anotaciones (5)

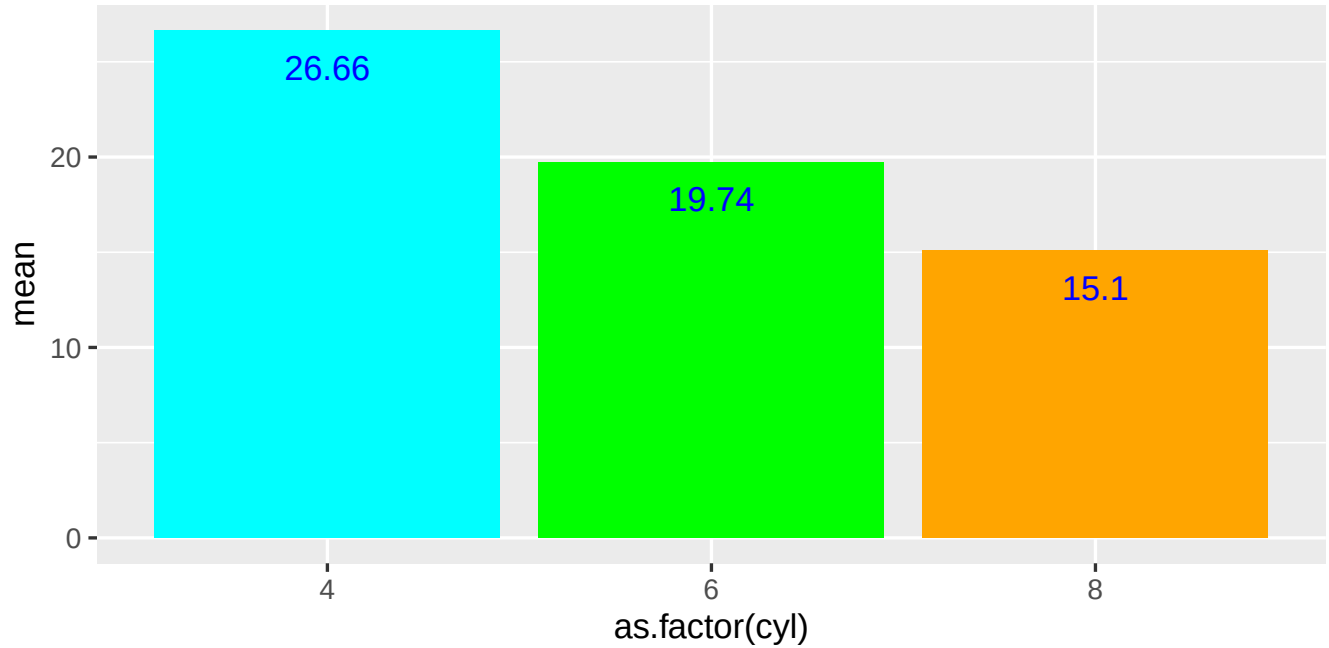
```
> ggplot(mtcars)+geom_bar(aes(gear,y=..prop..),fill='orange')+
+  geom_text(aes(x=gear,y=..prop..,label=round(..prop..,2)),
+            stat='count',vjust=2,color='blue')
```



Anotaciones (6)

	cyl	mean	sd	se	ci
1	4.00	26.66	4.51	1.36	2.67
2	6.00	19.74	1.45	0.55	1.08
3	8.00	15.10	2.56	0.68	1.34

```
> p1<-ggplot(car.new,aes(x=as.factor(cyl),y=mean))
> p2<-p1+geom_bar(stat='identity',fill=c('cyan','green','orange'))
> p2+geom_text(aes(label=round(mean,2)),vjust=2,col='blue')
```

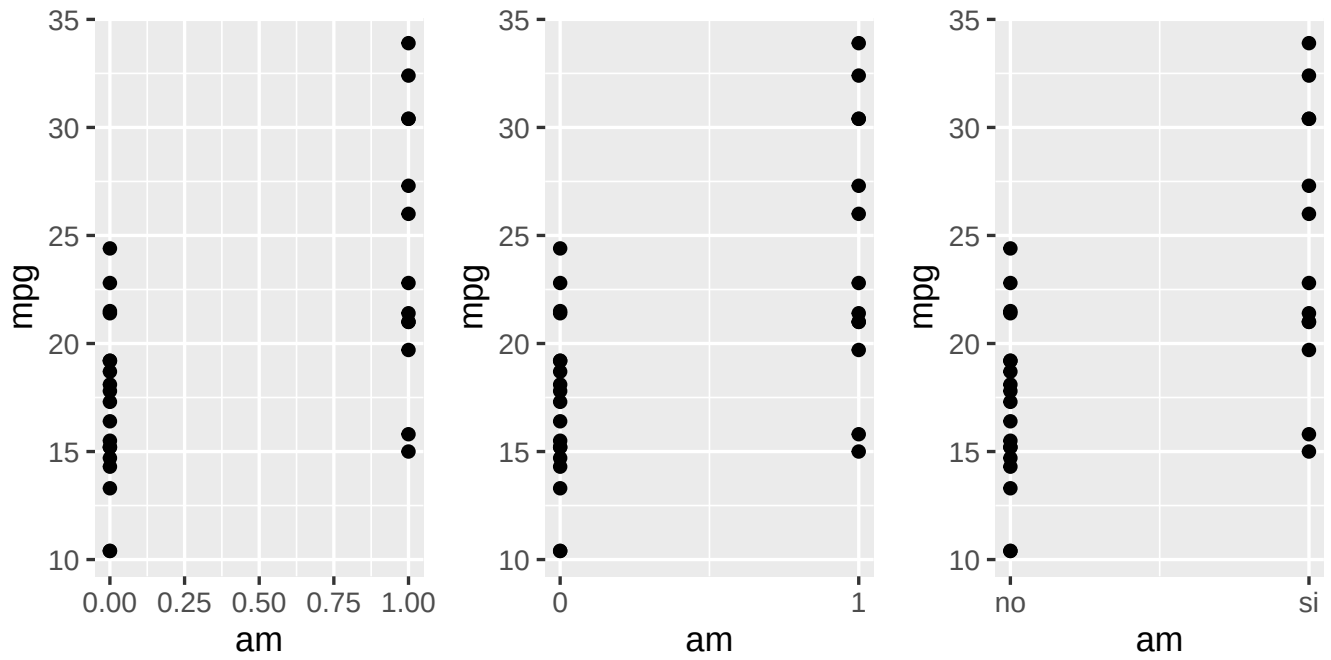


Esquema del tema

Scales

Las **scales** controlan la forma cómo se representan los valores de los datos.
Generalmente se aplican unas scales por defecto pero, como casi todo, puede personalizarse. Sirven para controlar desde si un eje es tratado como continuo o discreto, hasta si la escala es logarítmica, exponencial o con cualquier otra transformación.
Veamos algunos ejemplos

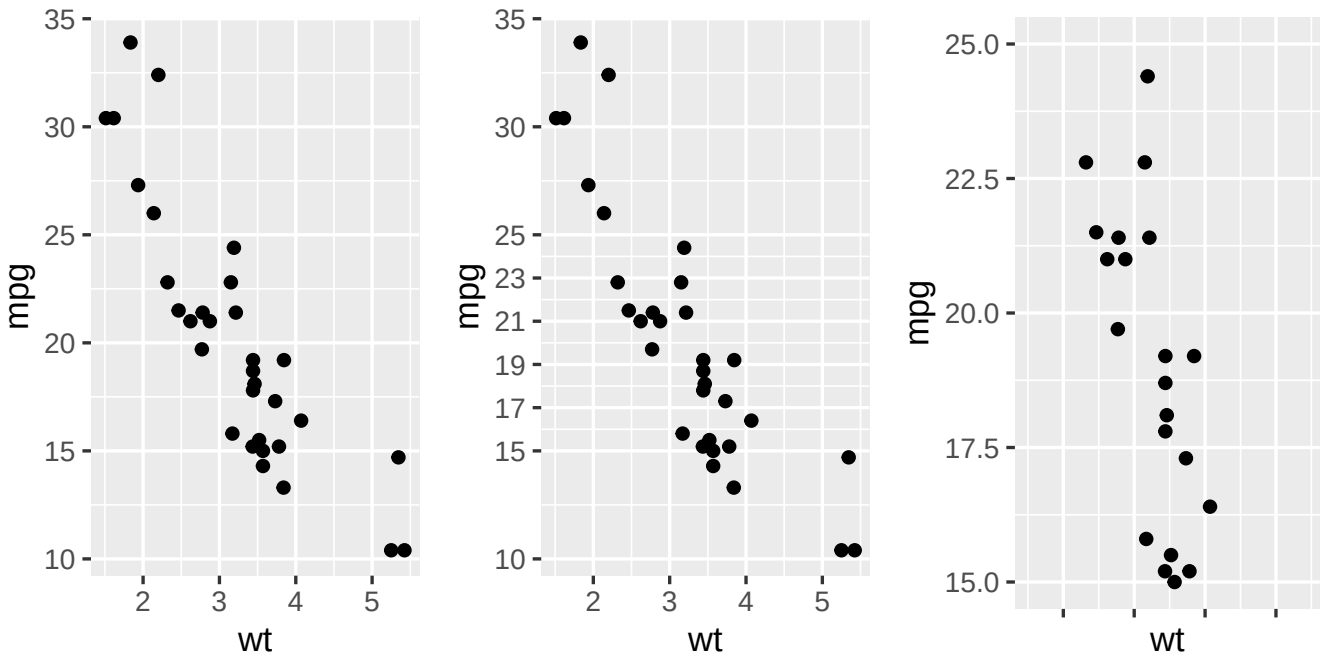
```
> p<-ggplot(mtcars,aes(x=am,y=mpg))
> grid.arrange(p+geom_point(),
+ p+geom_point()+scale_x_continuous(breaks=seq(0,1,1)),
+ p+geom_point()+scale_x_continuous(breaks=seq(0,1,1),
+                labels=c('no','si')),ncol=3)
```



Scales (2)

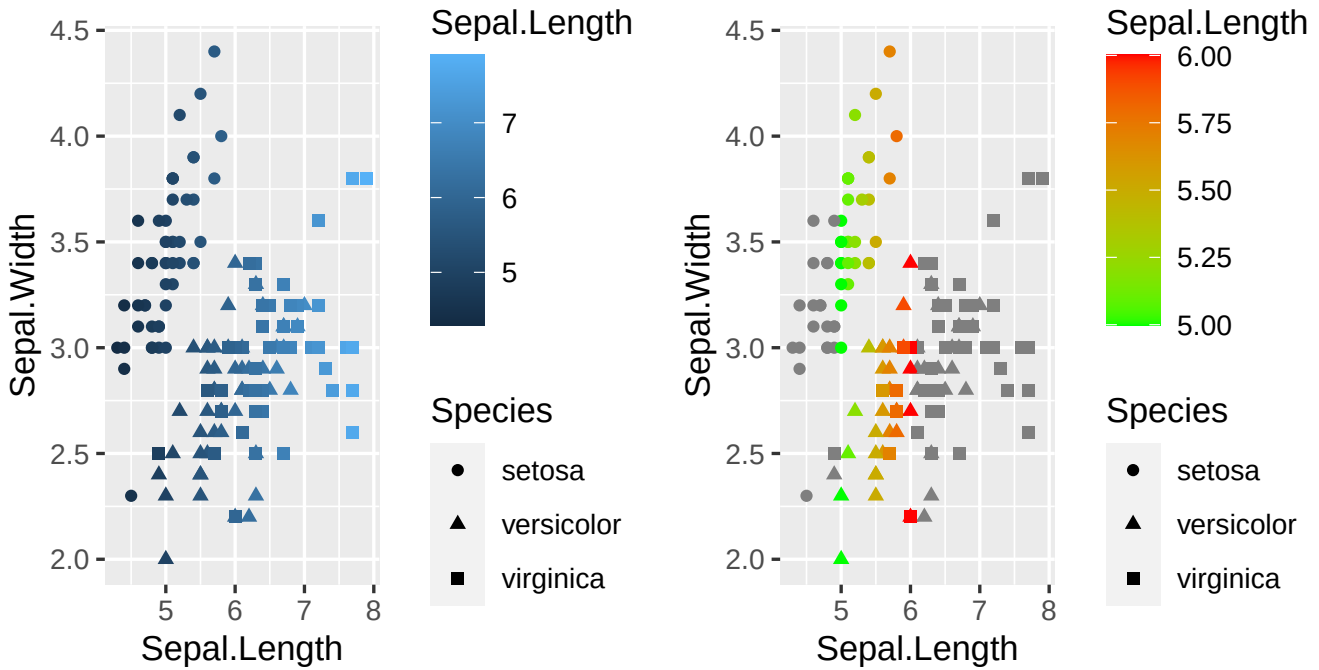
Podemos controlar de forma muy precisa la representación y las marcas de los ejes.

```
> p<-ggplot(mtcars,aes(x=wt,y=mpg))
> grid.arrange(p+geom_point(),
+ p+geom_point()+scale_y_continuous(breaks=c(10,seq(15,25,2),30,35)),
+ p+geom_point()+scale_x_continuous(labels=NULL)+
+  scale_y_continuous(limits=c(15,25)),ncol=3)
```



Scales (3)

```
> p<-ggplot(iris,aes(x=Sepal.Length,y=Sepal.Width))
> grid.arrange(p+geom_point(aes(color=Sepal.Length,shape=Species)),
+              p+geom_point(aes(color=Sepal.Length,shape=Species))+
+          scale_color_gradient(limits=c(5,6),low='green',high='red'), ncol=2)
```

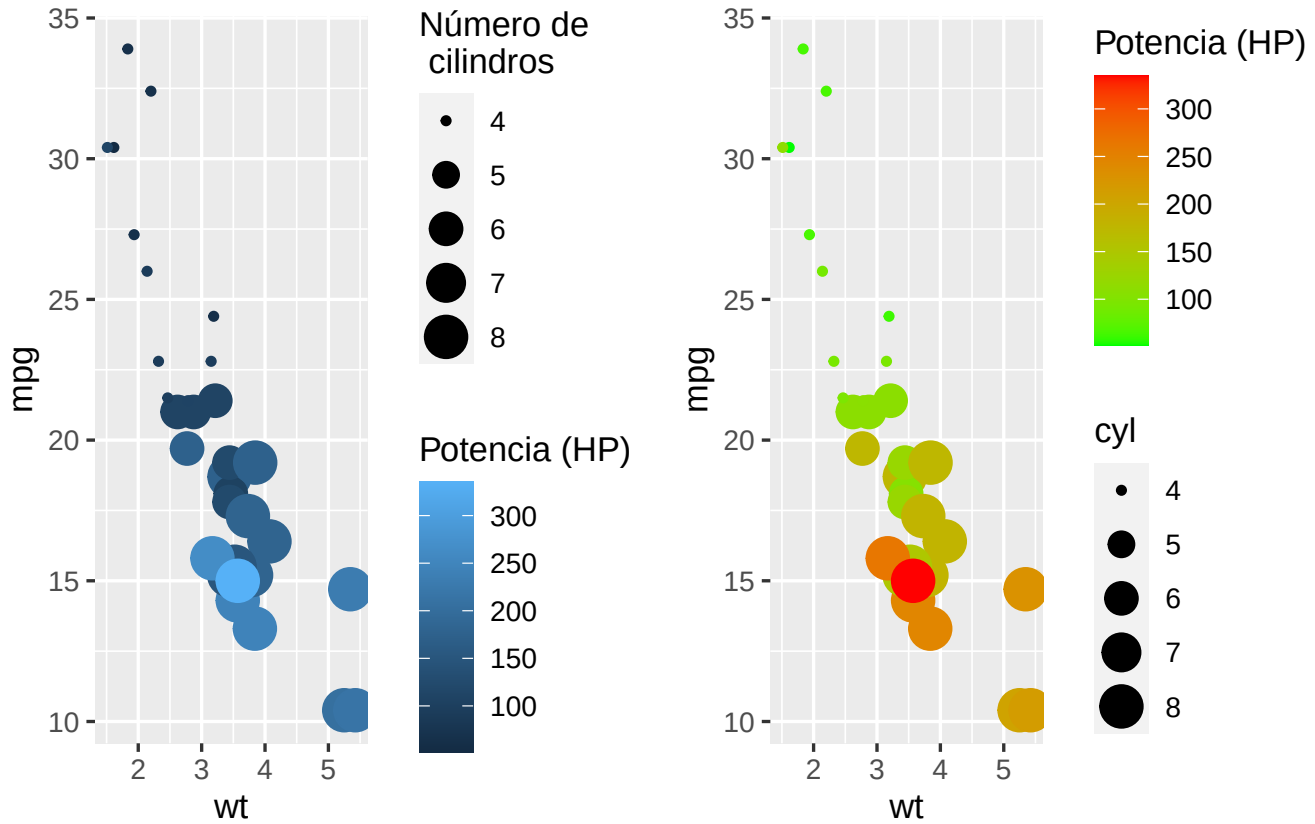


Tenemos todo tipo de funciones para modificar scales:scale_color, scale_fill, scale_alpha, scale_size,...

En general **scale_<aesthetic>_<type>**

Scales (4)

```
> p<-ggplot(mtcars,aes(x=wt,y=mpg))
> p1<-p+geom_point(aes(color=hp,size=cyl))
> grid.arrange(
+  p1+scale_size('Número de \n cilindros')+scale_color_continuous('Potencia (HP)'),
+  p1+scale_color_continuous('Potencia (HP)',low='green',high='red')
+  ,ncol=2
+ )
```



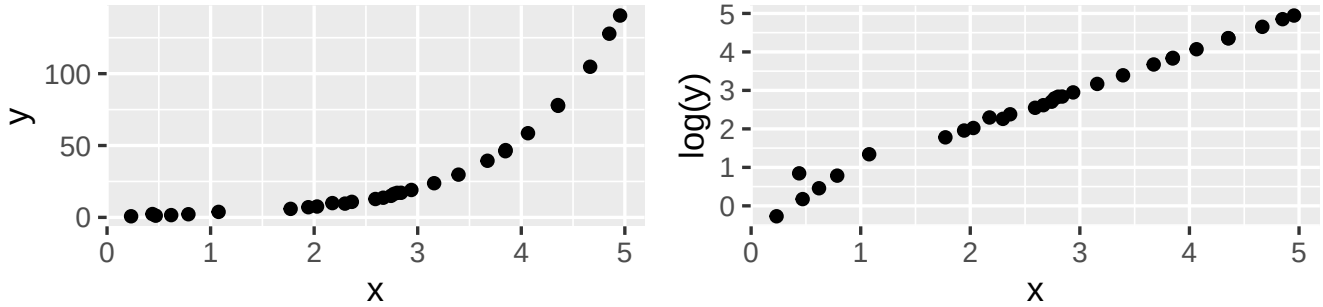
Ejercicio

Modificar el gráfico anterior ajustando el tamaño según hp y el color según el número de cilindros pero transformado en factor. Modificar la escala de color del cilindro para que utilize la paleta de colores Pastel1. Nota: buscar ayuda sobre *scale_color_brewer()*

Transformación de las variables con **scales**

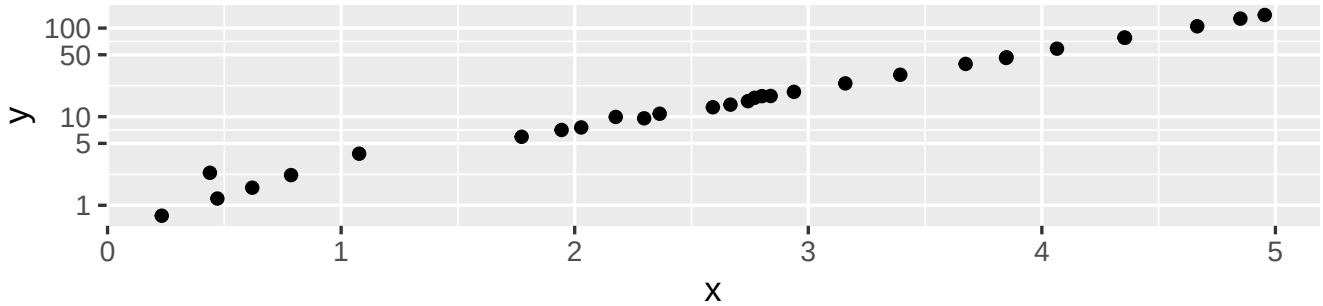
Siempre es posible transformar una variable antes de representarla.

```
> set.seed(12244)
> x<-runif(30,0,5)
> y<-exp(x)+rnorm(30,0,0.5)
> dades<-data.frame(x,y)
> grid.arrange(
+  ggplot(dades,aes(x,y))+geom_point(),
+  ggplot(dades,aes(x,log(y)))+geom_point(),
+  ncol=2)
```



Como vemos al transformar la variable las unidades y las escalas lógicamente se modifican. Como alternativa:

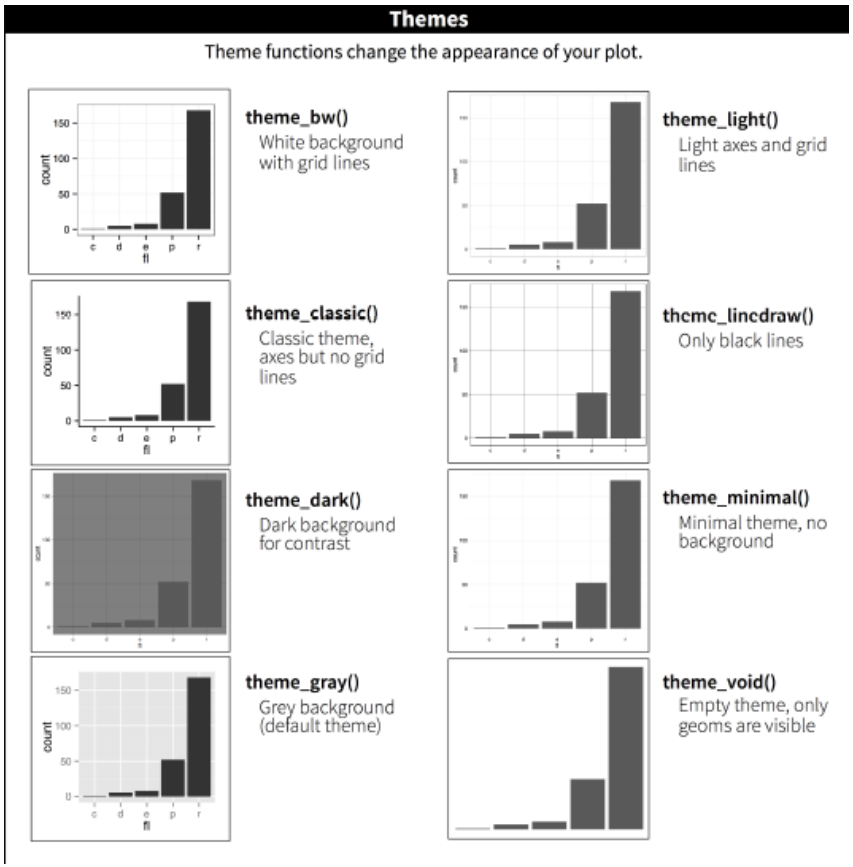
```
> ggplot(dades,aes(x,y))+geom_point()+scale_y_continuous(,breaks=c(0,1,5,10,50,100),trans='log')
```



Esquema del tema

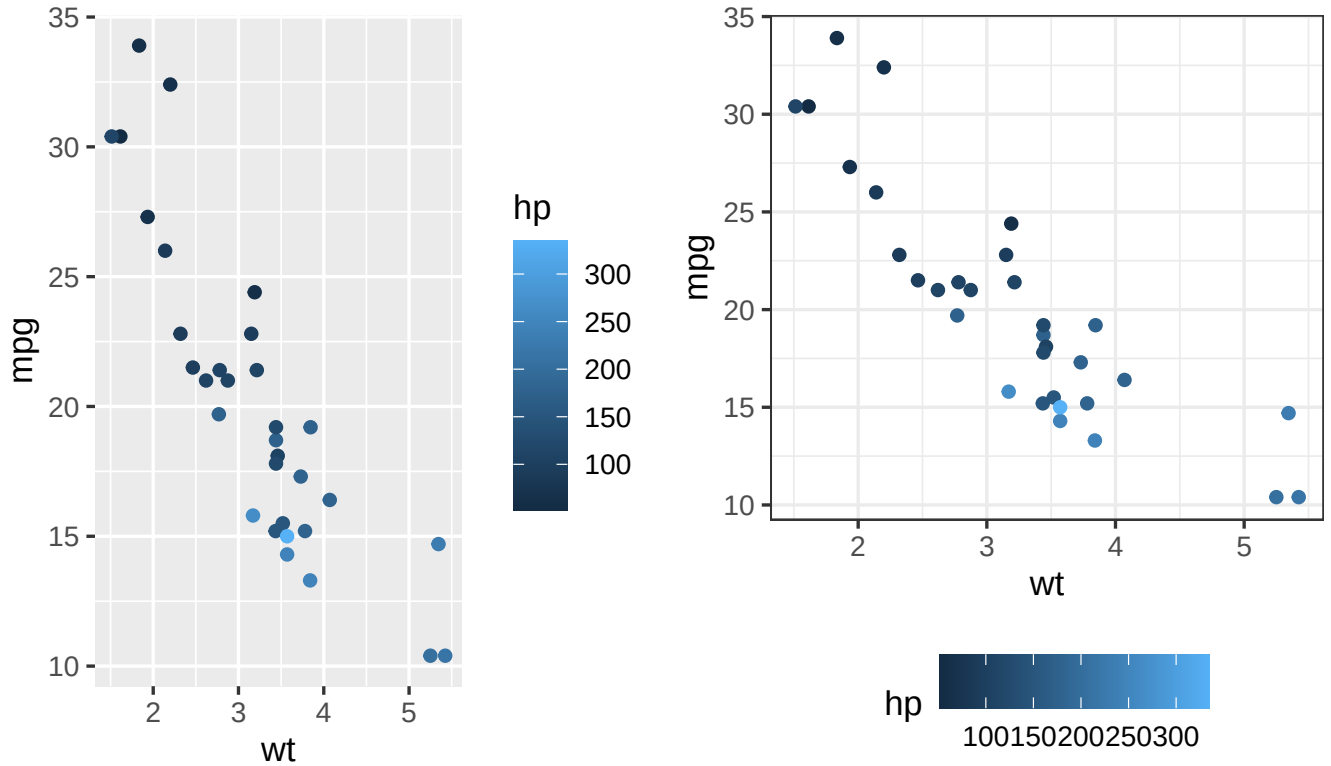
Themes

Los **themes** (temas) son las herramientas para personalizar los elementos que no están directamente relacionados con los datos: fondo, posición de la leyenda, etc.
Existen 8 temas por defecto en **ggplot2** y muchos más en extensiones como **ggthemes**.



Modificaciones de los **themes**

```
> p<-ggplot(mtcars,aes(wt,mpg,color=hp))+geom_point()
> grid.arrange(p,p+theme_bw()+theme(legend.position='bottom'),
+ ncol=2)
```



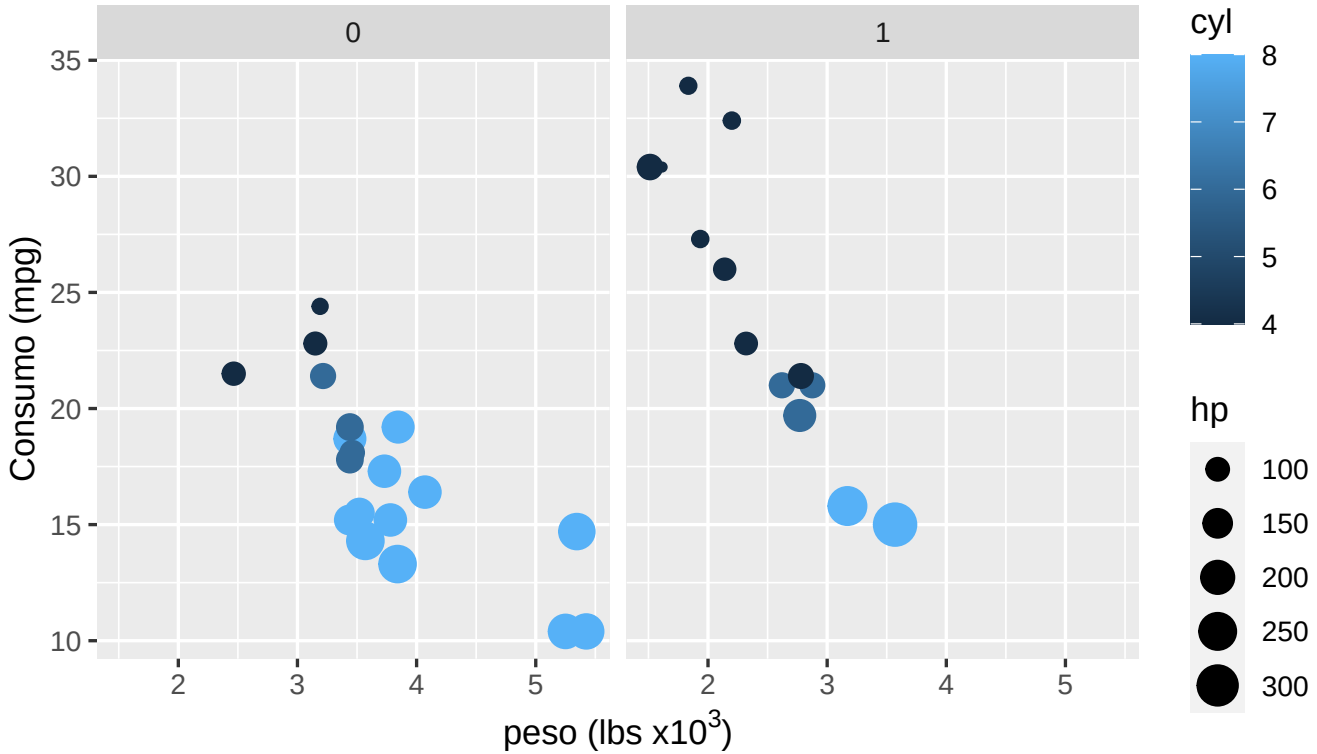
Opciones de los **themes**

```
theme(line, rect, text, title, aspect.ratio, axis.title, axis.title.x,
  axis.title.x.top, axis.title.y, axis.title.y.right, axis.text, axis.text.x,
  axis.text.x.top, axis.text.y, axis.text.y.right, axis.ticks, axis.ticks.x,
  axis.ticks.y, axis.ticks.length, axis.line, axis.line.x, axis.line.y,
  legend.background, legend.margin, legend.spacing, legend.spacing.x,
  legend.spacing.y, legend.key, legend.key.size, legend.key.height,
  legend.key.width, legend.text, legend.text.align, legend.title,
  legend.title.align, legend.position, legend.direction, legend.justification,
  legend.box, legend.box.just, legend.box.margin, legend.box.background,
  legend.box.spacing, panel.background, panel.border, panel.spacing,
  panel.spacing.x, panel.spacing.y, panel.grid, panel.grid.major,
  panel.grid.minor, panel.grid.major.x, panel.grid.major.y, panel.grid.minor.x,
  panel.grid.minor.y, panel.ontop, plot.background, plot.title, plot.subtitle,
  plot.caption, plot.margin, strip.background, strip.placement, strip.text,
  strip.text.x, strip.text.y, strip.switch.pad.grid, strip.switch.pad.wrap, ...,
  complete = FALSE, validate = TRUE)
```

La función *qplot*

La función *qplot()* permite crear un gráfico ggplot2 de forma simplificada. El control sobre las opciones es menor.

```
> qplot(wt,mpg,data=mtcars,facets=~am,col=cyl,size=hp,
+       xlab=expression(paste('peso (lbs x', 10^3,')')),
+       ylab='Consumo (mpg)')
```



Ejercicio final

Se trata de recuperar el gràfico que presentamos a continuación a partir de los datos contenidos en el fichero *HDR2013.csv*

Notas: point shape = 1 ; text label size = 3 ; smooth method = lm ; text legend size = 7