

DataViz - ggplot2+

Escuela de Métodos de Análisis Sociopolítico

Pablo Cabrera Álvarez
Departamento de Sociología y Comunicación (USAL)

20 junio 2018

Agenda día 3

- ▶ **15.30 - 16.30** Facets y theme
- ▶ **16.30 - 17.45** SP: Aplicar facets y theme
- ▶ **17.45 - 18.15** *Coffee break*
- ▶ **18.15 - 18.45** Ampliaciones de ggplot2
- ▶ **18.45 - 19.30** SP: Proyecto

Más ggplot2: stats, facets, coordinates y theme

La gramática de vuelta

- ▶ Elementos de la gramática de gráficos en ggplot2:
 - ▶ Data
 - ▶ Aesthetics
 - ▶ Geometries
 - ▶ Statistics
 - ▶ **Facets**
 - ▶ **Coordinates**
 - ▶ **Theme**

Facets

- ▶ En ocasiones para incluir una o más variables en el gráfico, además de la aesthetics se puede optar por crear un gráfico para cada categoría de la variable a incluir: facets
- ▶ Existen dos funciones:

```
#1D paneles a 2D: facet_wrap()
```

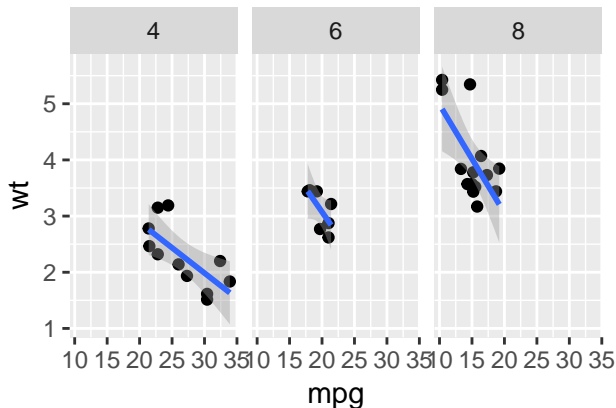
```
myplot + facet_wrap(~ MyFacetvar, ncol = n)
```

```
#2D panel por 1 o 2 var: facet_grid()
```

```
myplot + facet_grid(MyFacetVar1 ~ MyFacetVar2)
```

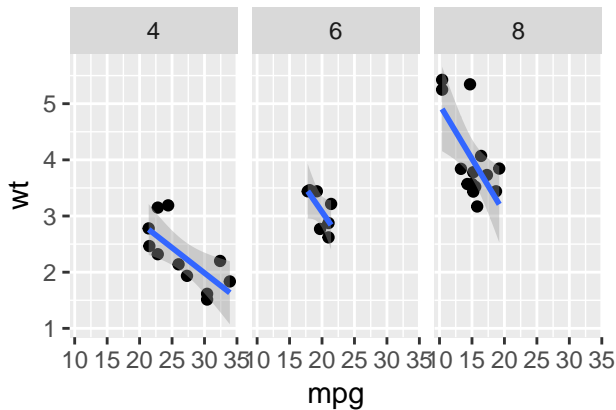
Ejemplo facet_wrap()

```
myplot <- ggplot(mtcars, aes(x=mpg, y=wt)) +  
  geom_point() +  
  geom_smooth(method="lm")  
myplot + facet_wrap(~ cyl, ncol=3)
```



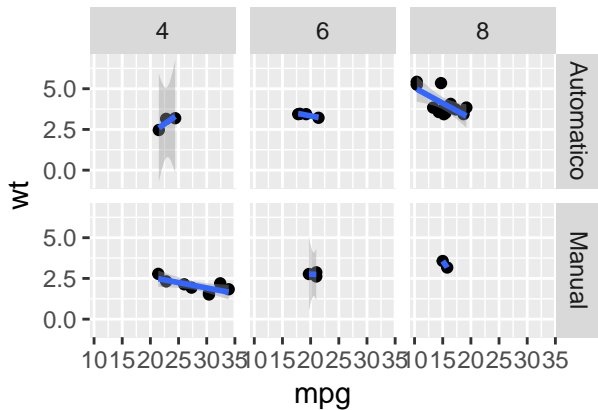
Ejemplo facet_grid() I

```
myplot + facet_grid(. ~ cyl)
```



Ejemplo facet_grid() II

```
myplot + facet_grid(am ~ cyl)
```

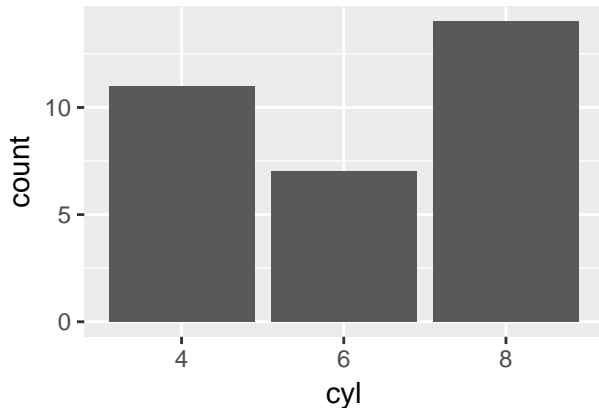


Coordinates

- ▶ La mayoría de gráficos utilizan coordenadas cartesianas. Hay diseños en los que también hay que modificar el tipo de eje. En ggplot2 es posible:
- ▶ Dos transformaciones susales:
 - ▶ `coord_flip()` - Para dar la vuelta a los ejes de coordenadas (barras a columnas)
 - ▶ `coord_polar()` - crear gráfico de sectores

Ejemplo coordinates la

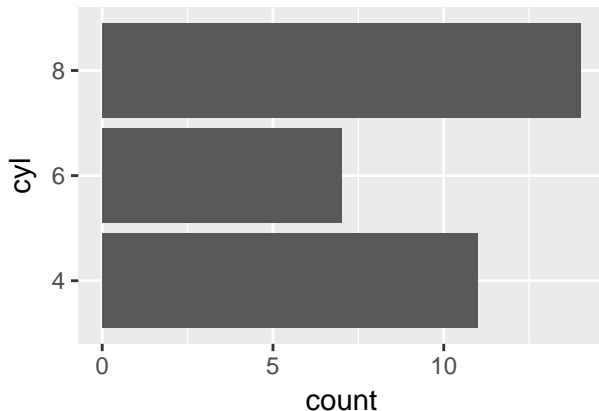
```
myplot <- ggplot(mtcars, aes(cyl)) +  
  geom_bar()  
myplot
```



Ejemplo coordinates lb

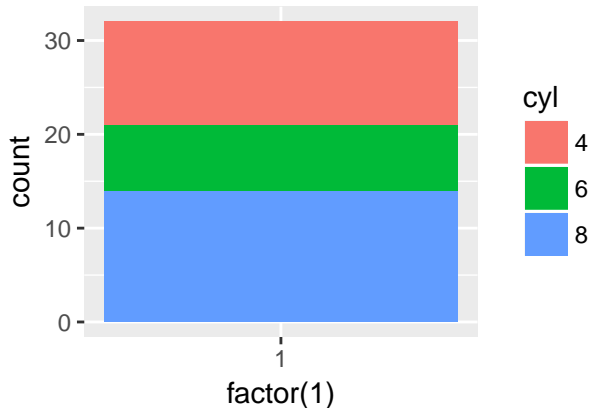
```
myplot <- ggplot(mtcars, aes(cyl)) +  
  geom_bar() +  
  coord_flip()
```

myplot



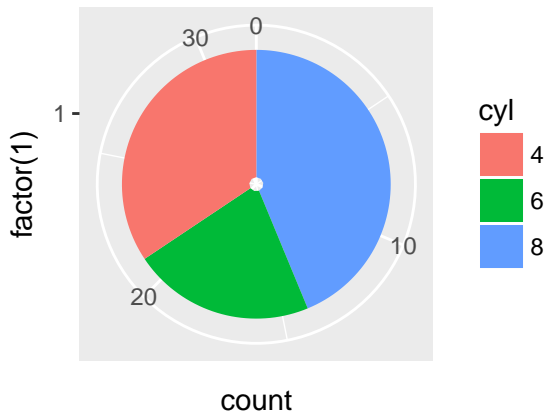
Ejemplo coordinates IIa

```
myplot <- ggplot(mtcars, aes(factor(1), fill=cyl)) +  
  geom_bar()  
myplot
```



Ejemplo coordinates 1lb

```
myplot <- ggplot(mtcars, aes(factor(1), fill=cyl)) +  
  geom_bar() +  
  coord_polar(theta="y")  
myplot
```



Theme

- ▶ Todos los elementos que no están relacionados con los datos “tinta extra” se engloban en theme
- ▶ Elementos modificables en la capa theme:
 - ▶ Los ejes
 - ▶ La leyenda
 - ▶ El área del gráfico y sus atributos
 - ▶ Los atributos de faceting

Ejes

- ▶ `axis.text.x` (text)
- ▶ `axis.text.y` (text)
- ▶ `axis.title.x` (text)
- ▶ `axis.title.y` (text)
- ▶ `axis.line` (segment)
- ▶ `axis.ticks` (segment)
- ▶ `axis.ticks.length` (unit)
- ▶ `axis.ticks.margin` (unit)

Leyenda

- ▶ legend.background (rect)
- ▶ legend.key (rect)
- ▶ legend.key.size (unit)
- ▶ legend.key.width (unit)
- ▶ legend.text (text)
- ▶ legend.title (text)
- ▶ legend.position ("bottom", "top", "left", "right", "none")
- ▶ legend.justification ("bottom", "top", "left", "right", "none")
- ▶ legend.direction (number)
- ▶ legend.box ("horizontal", "vertical")

Área del gráfico y atributos

- ▶ `panel.background` (rect)
- ▶ `panel.border` (rect)
- ▶ `panel.margin` (unit)
- ▶ `panel.grid.major` (line)
- ▶ `panel.grid.minor` (line)
- ▶ `plot.background` (rect)
- ▶ `title` (see `plot.title`)
- ▶ `plot.title` (text)
- ▶ `plot.margin` (unit)

Atributos del faceting

- ▶ `strip.background` (rect)
- ▶ `strip.text.x` (text)
- ▶ `strip.text.y` (text)

¿Cómo modificar los atributos?

- ▶ `element_rect()`
- ▶ `element_line()`
- ▶ `element_text()`
- ▶ `element_blank()`

element_rect()

- ▶ colour
- ▶ fill
- ▶ size
- ▶ linetype

`element_line()`

- ▶ colour
- ▶ size
- ▶ linetype

element_text()

- ▶ colour
- ▶ size
- ▶ angle
- ▶ hjust
- ▶ vjust
- ▶ face = 'bold' 'ital'

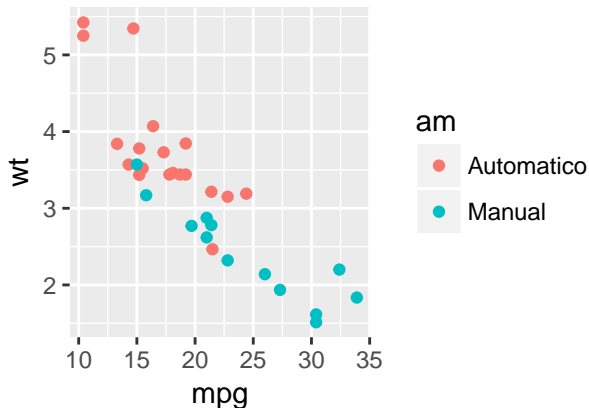
¿Cómo se cambia un elemento?

- ▶ ¿Qué elemento quiero cambiar?
 - ▶ La línea del eje = `axis.line`
- ▶ ¿Qué tipo de elemento es la línea del eje?
 - ▶ `element_line()`
- ▶ ¿Qué atributo del elemento quiero cambiar?
 - ▶ El color a negro = `colour="black"`

```
myplot + theme(axis.line =  
                element_line(colour=black))
```

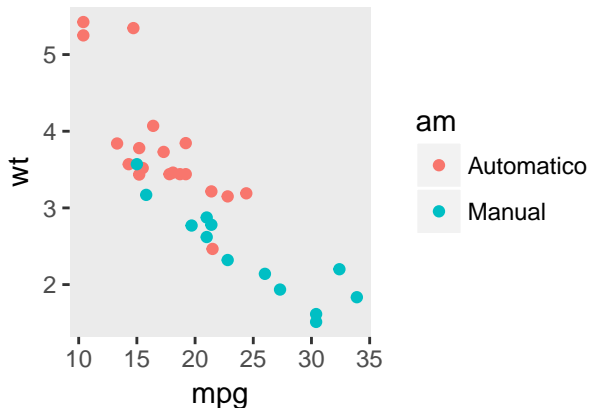
Ejemplo theme - base

```
myplot <- ggplot(mtcars, aes(x=mpg, y=wt,  
                             col=am)) +  
  geom_point()  
myplot
```



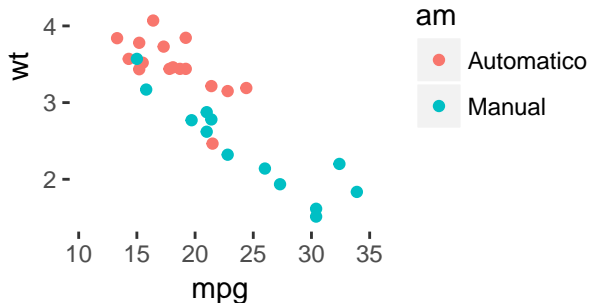
Ejemplo theme - quitar grid

```
myplot <- myplot +  
  theme(panel.grid = element_blank())  
myplot
```



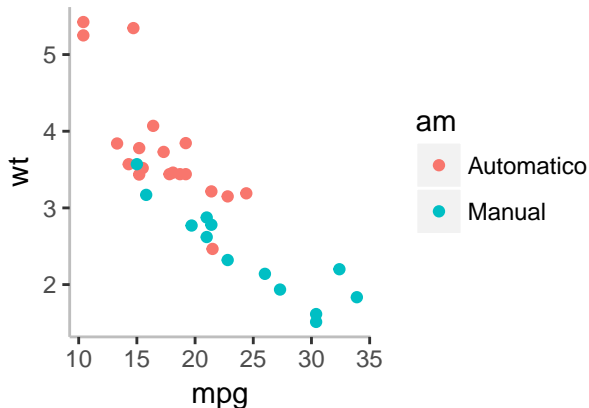
Ejemplo theme - color background a blanco

```
myplot <- myplot +  
  theme(panel.background =  
    element_rect(fill="white"))  
myplot
```



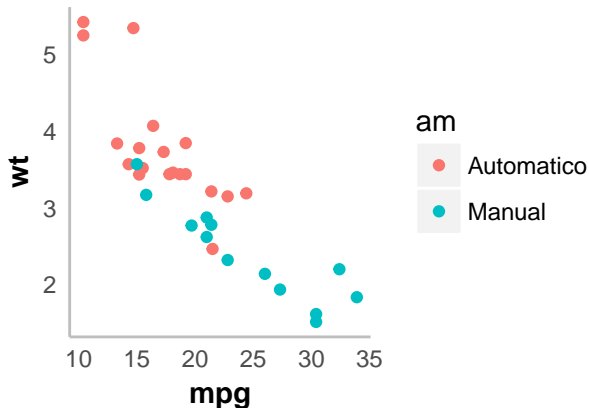
Ejemplo theme - añadir líneas ejes grises

```
myplot <- myplot +  
  theme(axis.line = element_line(color="gray"))  
myplot
```



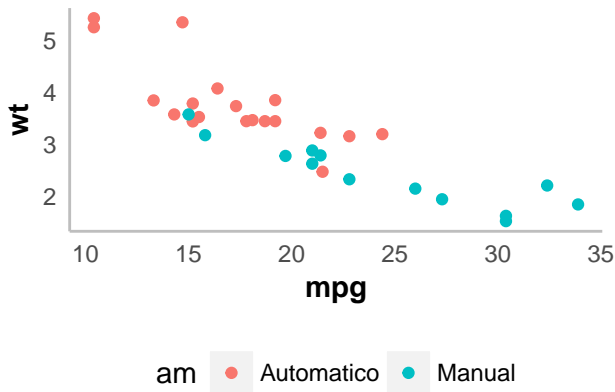
Ejemplo theme - quitar ticks

```
myplot <- myplot +  
  theme(axis.ticks = element_blank())  
myplot
```



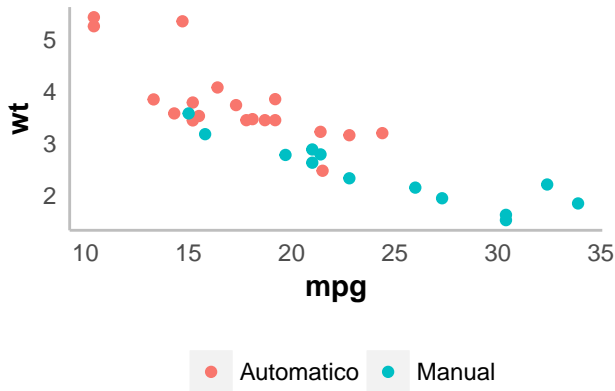
Ejemplo theme - poner leyenda abajo

```
myplot <- myplot +  
  theme(legend.position = "bottom")  
myplot
```



Ejemplo theme - retirar título leyenda

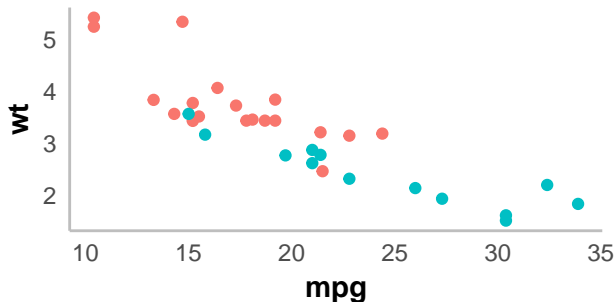
```
myplot <- myplot +  
  theme(legend.title = element_blank())  
myplot
```



Ejemplo theme - cambio ancho leyenda

```
myplot +
```

```
theme(legend.key.width = unit(.5, "cm"))
```



Automatic

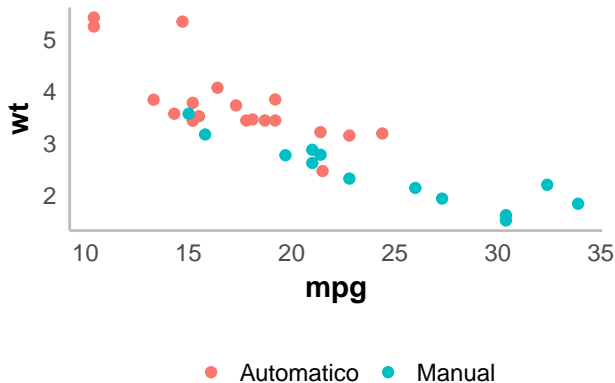


Manual

```
myplot
```


Ejemplo theme - cambio color background elementos leyenda

```
myplot +  
  theme(legend.key = element_rect(fill="white"))
```



```
myplot
```

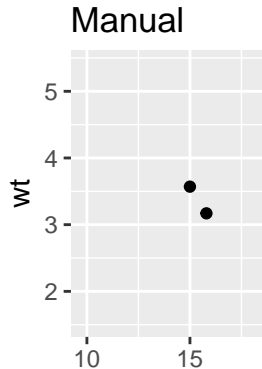
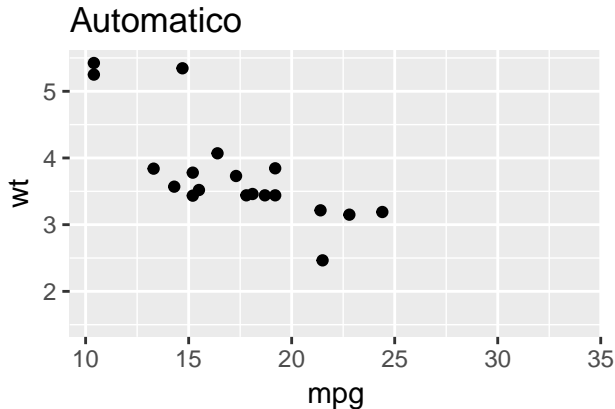
Y guardar el gráfico...

```
ggsave(myplot.png, plot=myplot)
```

Extensiones de ggplot2

Ejemplo gganimate()

```
animplot <- ggplot(mtcars, aes(mpg, wt, frame = am)) + geom_point()
gganimate(animplot)
```



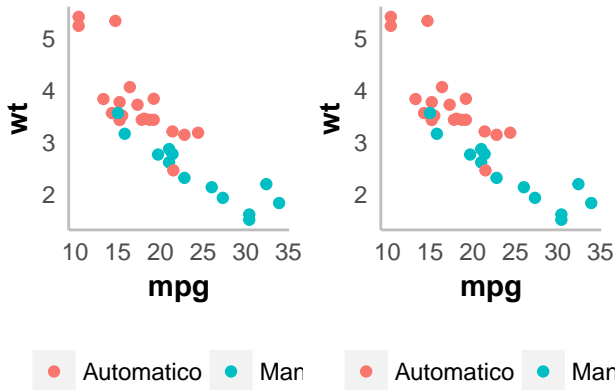
Pack: grid

- ▶ En ocasiones no se trata de crear un *facet* añadiendo nuevas variables, sino de combinar difernetes gráficos. Para ello:

```
install.packages("grid")  
library(grid)  
  
grid <- grid.arrange(graf1, graf2, ncol=n)
```

Ejemplo grid.arrange()

```
g1 <- myplot  
g2 <- myplot  
grid <- grid.arrange(g1, g2, ncol=2)
```



grid

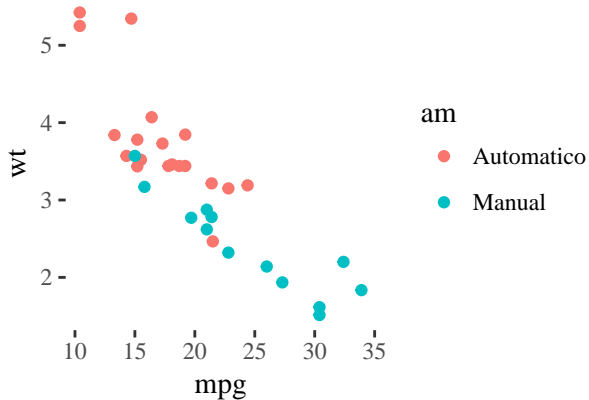
Pack: ggthemes

- En vez de configurar todos los elementos de un tema, se puede comenzar con una composición ofrecida por ggthemes

```
install.packages("ggthemes")  
library(ggthemes)  
  
plot <- plot + theme_tufte()
```

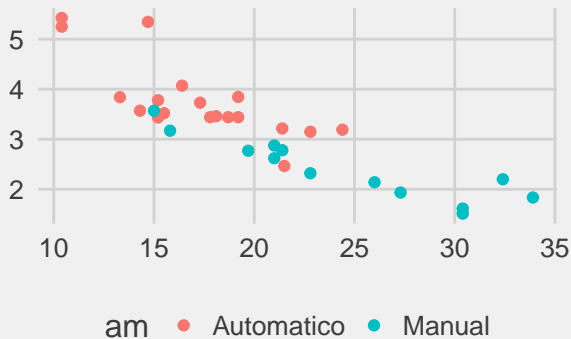

Ejemplo themes_tufte()

```
myplot + theme_tufte()
```



Ejemplo themes_fivethirtyeight()

```
myplot + theme_fivethirtyeight()
```



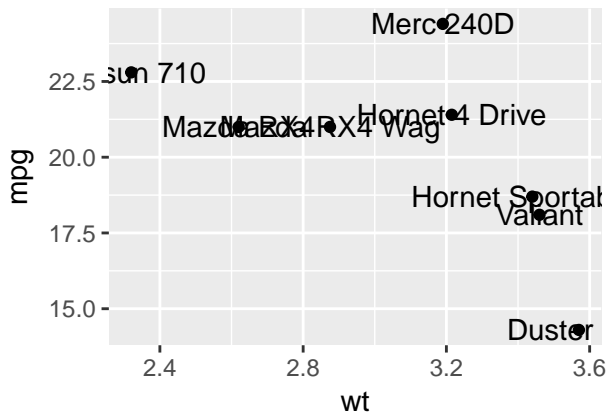
Pack: ggrepel

- En ocasiones puede haber problemas con las etiquetas superpuestas, ggrepel es una forma de evitar este problema

```
install.packages("ggrepel")  
library(ggrepel)  
  
plot <- plot + geom_text_repel(aes(label = varname))
```

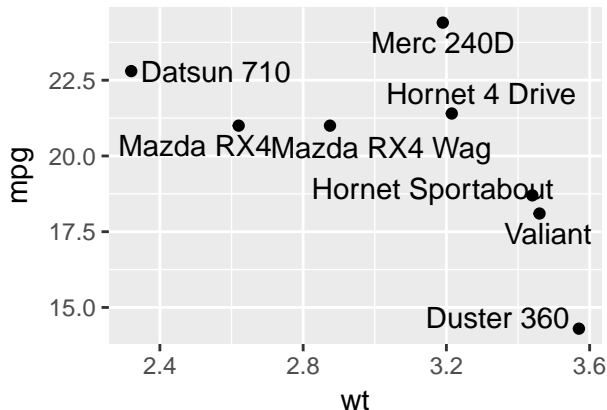
Ejemplo geom_text()

```
smallcars <- mtcars[1:8, ]  
ggplot(smallcars, aes(x = wt, y = mpg,  
                      label = rownames(smallcars))) +  
  geom_point( ) +  
  geom_text()
```



Ejemplo geom_text_repel()

```
ggplot(smallcars, aes(x = wt, y = mpg,  
                      label = rownames(smallcars))) +  
  geom_point( ) +  
  geom_text_repel()
```



Proyectos