

DataViz - Gráficos con ggplot2

Escuela de Métodos de Análisis Sociopolítico

Pablo Cabrera Álvarez

Departamento de Sociología y Comunicación (USAL)

19 junio 2018

Agenda día 2

- ▶ **15.30 - 16.30** Gramática de gráficos y ggplot2
- ▶ **16.30 - 17.45** SP: Gráficos con ggplot2
- ▶ **17.45 - 18.15** *Coffee break*
- ▶ **18.15 - 18.45** ggplot2: stats
- ▶ **18.45 - 19.30** SP: Proyecto

Introducción a ggplot2: data, aesthetics y geoms

La gramática

<i>Sujeto</i>	<i>Artículo</i> <i>Sustantivo</i>	El perro	El profesor
<i>Verbo</i>	<i>Verbo</i>	ladra	enseña
<i>Complemento</i>	<i>Artículo</i> <i>Sustantivo</i>	al dueño	

Figure 1

La gramática de gráficos

La gramática de gráficos es una herramienta que permite describir de manera concisa los elementos de un gráfico

(Wickham, 2010)

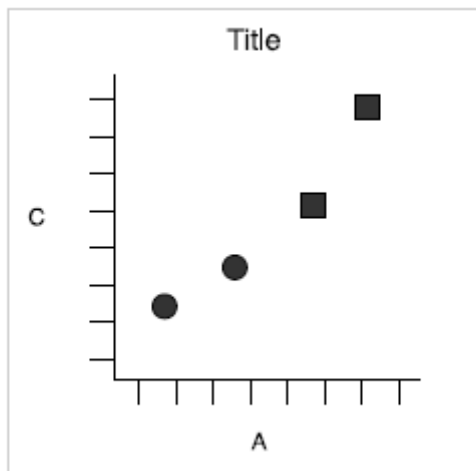


Figure 2

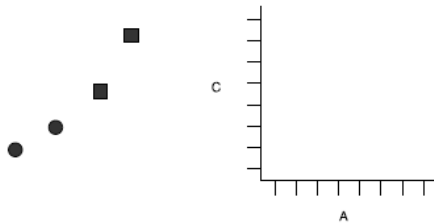


Figure 3

¿Qué elementos son imprescindibles en un gráfico?

Elemento	Descripción
Data	Los datos necesarios para crear el gráfico
Aesthetics	Las características en las que los datos son representados
Geometries	Los elementos visuales que dan formato al gráfico (p. ej. puntos, barras. . .)

¿Qué elementos pueden incluirse en un gráfico?

Elemento	Descripción
Data	Los datos necesarios para crear el gráfico
Aesthetics	Las características en las que los datos son representados
Geometries	Los elementos visuales que dan formato al gráfico (p. ej. puntos, barras. . .)
Statistics	Es posible representar transformaciones estadísticas de los datos
Facets	Generar gráficos de manera simultánea a partir de una o varias variables
Coordinates	Sistema de coordenadas a utilizar
Theme	Todos los elementos estéticos adicionales

Aspectos de la gramática

Elemento Descripción

Data Base de datos

Aesthetics Eje X, eje Y, color, relleno, etiquetas, tamaño, forma, transparencia, ancho de línea, tipo de línea. . .

Geometries Barras, puntos, líneas, rectángulos. . .

Statistics Recta de regresión, estadísticos descriptivos, nivel de error. . .

Facets Filas y columnas

Coordinates Cartesianas, polar. . .

Theme Color de fondo, líneas de grid, aspecto línea de los ejes, aspecto leyenda. . .

Pero la gramática no lo es todo. . .

Puede haber muchos gráficos con la gramática correcta que no tengan sentido. La gramática es solo el primer paso para crear una buena oración.

(Healy y Moody, 2014)

ggplot2 y la gramática de gráficos

- ▶ ggplot2 es un paquete de R utilizado para generar gráficos
- ▶ Su estructura está basada en la gramática de gráficos (Wickham, 2010)
- ▶ Elementos de la gramática de gráficos en ggplot2:
 - ▶ **Data**
 - ▶ **Aesthetics**
 - ▶ **Geometries**
 - ▶ Statistics
 - ▶ Facets
 - ▶ Coordinates
 - ▶ Theme

Estructura ggplot2

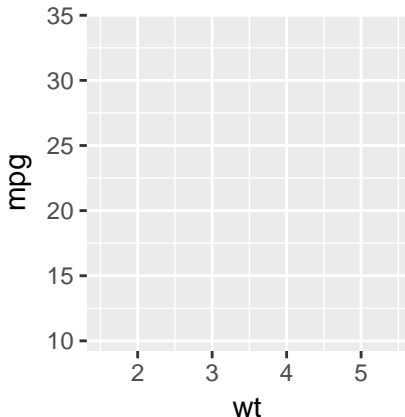
- ▶ **ggplot2** se estructura basado en la gramática de gráficos a través de capas, cada capa añade un elemento nuevo
- ▶ Cada capa se añade utilizando el signo “+”
- ▶ Estructura básica:

```
myplot <- ggplot(data=mydata, aes(aes1=var1,  
                                  aes2=var2, aesN=varN)) +  
  geom_xxx() +  
  stat_xxx() +  
  facet_XXX() +  
  theme()  
  
myplot
```

Ejemplo ggplot2 I

En la primera capa se establecen data y aestehtics:

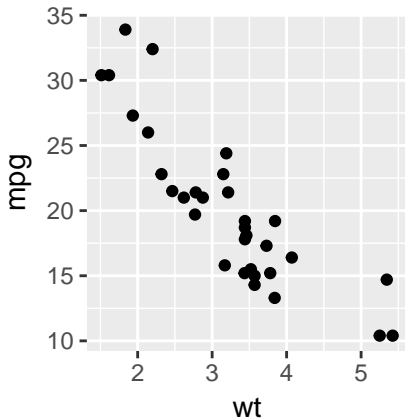
```
g1 <- ggplot(data = mtcars, aes(x = wt, y = mpg))  
g1
```



Ejemplo ggplot2 II

En la segunda se añade “+” la geom:

```
g1 <- g1 + geom_point()  
g1
```



Datos

- ▶ Piensa en el gráfico y en la estructura de **datos** que necesitas para llevarlo a cabo
- ▶ Primero lleva a cabo las operaciones como recodificaciones, creación de nuevas variables. . .
- ▶ La transformación de datos pueden ahorrar mucho tiempo

Ejemplo datos

Queremos realizar un gráfico de la relación entre peso (wt) del coche y aceleración (qsec), pero solo para los automáticos

```
head(mtcars[, c("am", "wt", "qsec")], n=10)
```

##	am	wt	qsec
## Mazda RX4	Manual	2.620	16.46
## Mazda RX4 Wag	Manual	2.875	17.02
## Datsun 710	Manual	2.320	18.61
## Hornet 4 Drive	Automatico	3.215	19.44
## Hornet Sportabout	Automatico	3.440	17.02
## Valiant	Automatico	3.460	20.22
## Duster 360	Automatico	3.570	15.84
## Merc 240D	Automatico	3.190	20.00
## Merc 230	Automatico	3.150	22.90
## Merc 280	Automatico	3.440	18.30

Ejemplo datos 2

Para seleccionar solo los casos `am == "Automatico"` hay dos opciones:

#Opción 1

```
mtcars2 <- filter(mtcars, am == "Automatico")
```

```
myplot <- ggplot(data = mtcars2, aes(x= wt, y=qsec)) +  
  geom_point()
```

#Opción 2

```
myplot <- ggplot(data = filter(mtcars2, am == "Automatico"),  
  aes(x= wt, y=qsec)) +  
  geom_point()
```

Aesthetics I

La **aesthetics** se refiere a las características del gráfico que utilizamos para representar (*mapear*) información.

Aspectos que pueden ser utilizados para representar información (cambia según la geometría):

- ▶ Eje x (x)
- ▶ Eje y (y)
- ▶ Color (col)
- ▶ Forma (shape)
- ▶ Tamaño (size)
- ▶ Transparencia (alpha)
- ▶ Relleno (fill)
- ▶ Tipo de línea (linetype)
- ▶ Anchura (width)
- ▶ Máximos y mínimos (xmax, ymax, xmin, y min)
- ▶ ...

Aesthetics II

- ▶ No confundir aestehtics con atributos
- ▶ La aesthetics se incorpora de manera preferible dentro de `ggplot()` -> Efecto herencia
- ▶ En caso de que quieras combinar más de un gráfico, puedes incorporar la aestehtics a un `geom_xxx()`

Por ejemplo:

#Opción 1

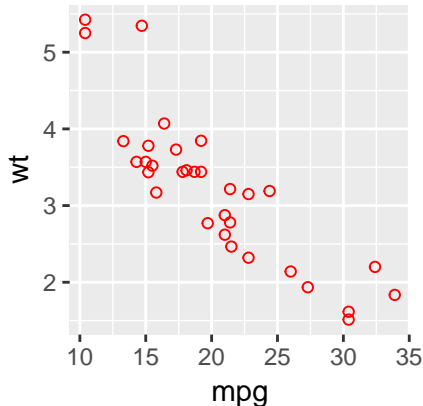
```
myplot <- ggplot(data = mtcars,  
                 aes(x= wt, y=qsec, col=am, shape=cyl)) +  
  geom_point()
```

#Opción 2

```
myplot <- ggplot(data = mtcars) +  
  geom_point(aes(x= wt, y=qsec,  
                 col=am, shape=cyl))
```

Ejemplo aestehctics: atributos vs aesthetics

```
myplot <- ggplot(data = mtcars, aes(x = mpg, y=wt)) +  
  geom_point( shape=1, col="red")  
myplot
```

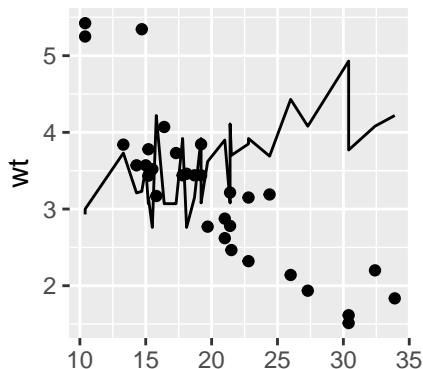


Ejemplo aesthetics: efecto herencia

Gráfico en el eje X **mpg** (consumo), y en el eje Y tanto **drat** (ratio eje - línea) como **wt** (peso - dispersión)

```
myplot <- ggplot(data = mtcars, aes(x = mpg)) +  
  geom_point(aes(y = wt)) +  
  geom_line(aes(y = drat))
```

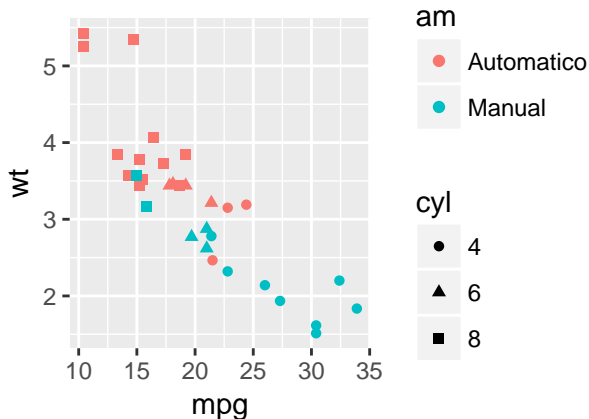
myplot



Ejemplo aestehtics: límites de la aestehtics

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

myplot



Aesthetics: escalas I

Cada elemento de la aesthetics que utilizamos está sujeto a una escala. Las escalas ponen en relación los valores de los datos con los valores de los elementos visuales de la aestehtics, por ejemplo:

- ▶ Valores de los *datos*: "PP", "PSOE", "Podemos"
- ▶ Valores de la *aestehtics*: "Azul", "Rojo", "Morado"

```
myplot + scale_fill_manual(  
  values = c("Blue", "Red", "Purple"),  
  labels = c("PP", "PSOE", "Podemos"))
```


Aesthetics: escalas II

La "aes" se sustituye por el tipo de aesthetics (e.g. x, y, col, shape, size...)

```
# para valores continuos  
scale_aes_continuous(limits = c("límite1", "límite2"),  
                      breaks = c("break1", "break2"),  
                      name = "minombredeaestehtics",  
                      labels = c("Etiqueta1", "Etiqueta2"))
```

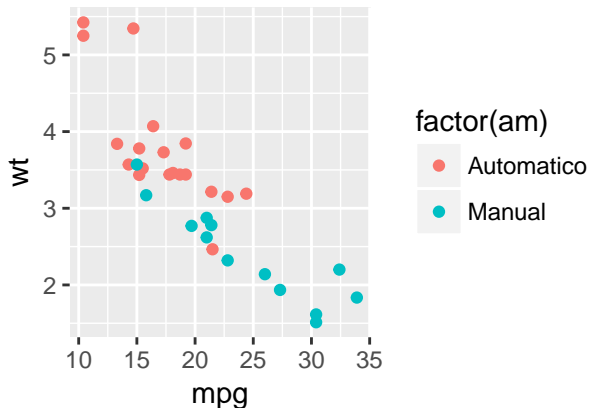
```
# para valores discretos  
scale_aes_discrete()
```

```
# usa los valores de los datos como visuales  
scale_aes_identity()
```

```
#mapea valores discreto manual  
scale_aes_manual(values = c())
```

Ejemplo escalas I

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

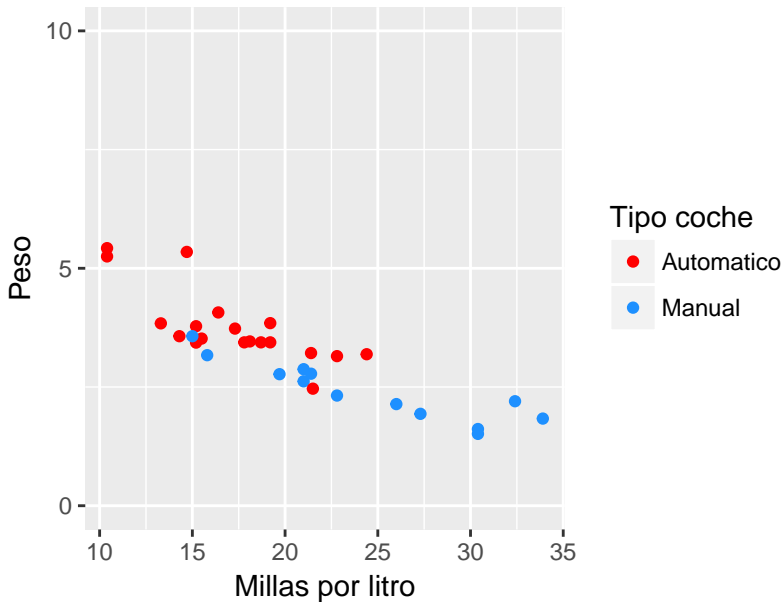


Ejemplo escalas II (a)

```
myplot <- ggplot(data = mtcars, aes(x = mpg, y=wt,
                                     col=factor(am))) +
  geom_point() +
  scale_y_continuous(limits = c(0, 10),
                    breaks = c(0, 5, 10),
                    name = "Peso") +
  scale_x_continuous(name =
                    "Millas por litro") +
  scale_color_manual(values = c("red", "dodgerblue"),
                    labels = c("Automatico", "Manual"),
                    name = "Tipo coche")
```

```
myplot
```

Ejemplo escalas II (b)



labs()

Otra forma de etiquetar los elementos de la aesthetics, así como dar un título al gráfico es utilizando la función **labs()**

```
myplot <- myplot + labs( x = "NombreEjeX",  
                        y = "NomberEjeY",  
                        title = "Título",  
                        subtitle = "Subtítulo",  
                        caption = "NotaAlPie",  
                        MiAes = "MiAes_título")
```

Geoms

- ▶ Los **geoms** son las geometrías disponibles para crear gráficos que existen en ggplot
- ▶ Cada geom está relacionado con una determinada aesthetics (y atributos)
- ▶ Varios geoms se pueden combinar en un mismo gráfico
- ▶ Los atributos de cada elemento geométrico se definen dentro del mismo
- ▶ Los geoms admiten data, aesthetics, y stats dentro

Geoms & aesthetics

Una *ayuda* para tener un listado de los elementos de la gramática. . .

Geoms con los que vamos a trabajar...

- ▶ Dispersión - `geom_point()`
- ▶ Barras - `geom_bar()`
- ▶ Histograma - `geom_histogram()`
- ▶ Densidad - `geom_density()`
- ▶ Línea de tendencia - `geom_smooth()`
- ▶ Línea - `geom_line()`
- ▶ Cajas y bigotes - `geom_boxplot()`
- ▶ Rectángulos - `geom_rect()`

Ejemplo geoms I - data and aesthetics

#Opcion 1

```
myplot <- ggplot(data = mtcars, aes(x = mpg, y=wt,  
                                     col=factor(am))) +  
  geom_point(alpha=0.8, shape=1)  
myplot
```

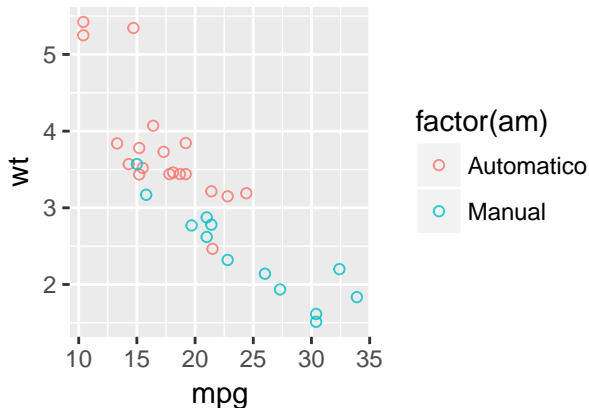
#Opcion 2

```
myplot <- ggplot() +  
  geom_point(data = mtcars, aes(x = mpg, y=wt,  
                                col=factor(am)), alpha=0.8, shape=1)  
myplot
```

Ejemplo geoms I - Opción 1

#Opcion 1

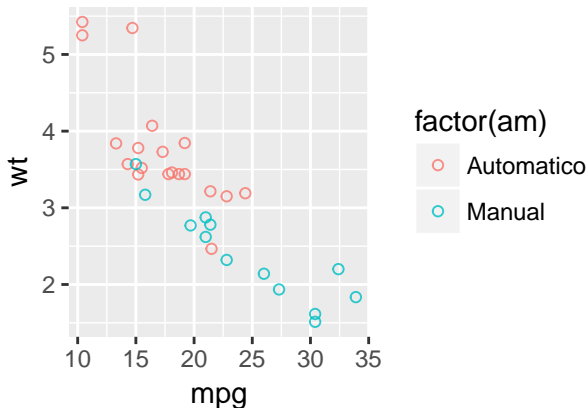
```
myplot <- ggplot(data = mtcars, aes(x = mpg, y=wt,  
                                     col=factor(am))) +  
  geom_point(alpha=0.8, shape=1)  
myplot
```



Ejemplo geoms I - Opción 2

#Opcion 2

```
myplot <- ggplot() +  
  geom_point(data = mtcars, aes(x = mpg, y=wt,  
                                col=factor(am)), alpha=0.8, shape=1)  
myplot
```

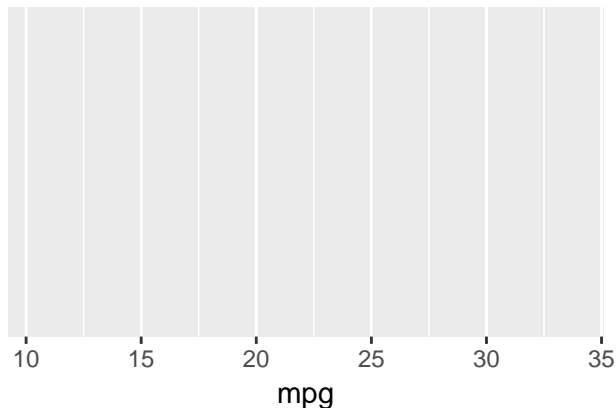


Ejemplo geoms II - base

```
#Establecer la base
```

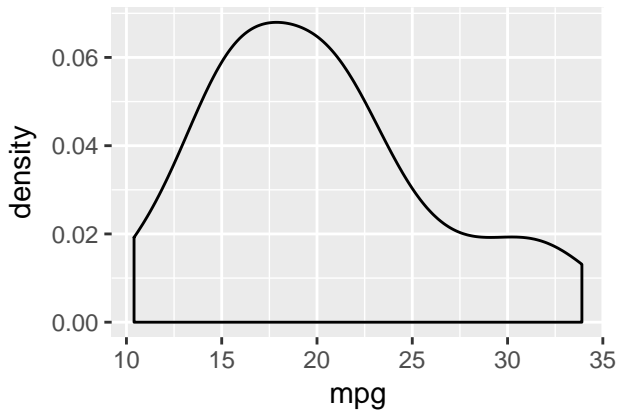
```
myplot <- ggplot(data = mtcars, aes(x = mpg))
```

```
myplot
```



Ejemplo geoms II - geom_density()

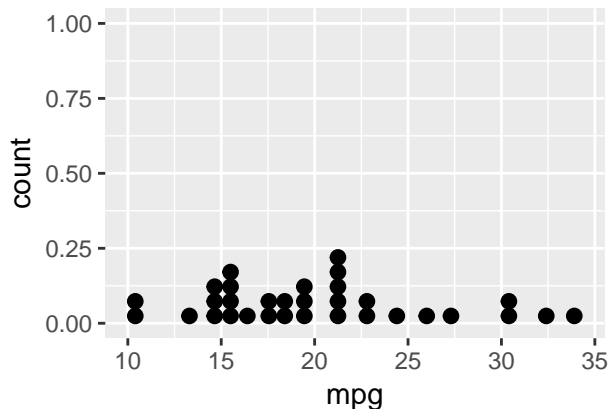
```
myplot + geom_density()
```



Ejemplo geoms II - geom_dotplot()

```
myplot + geom_dotplot()
```

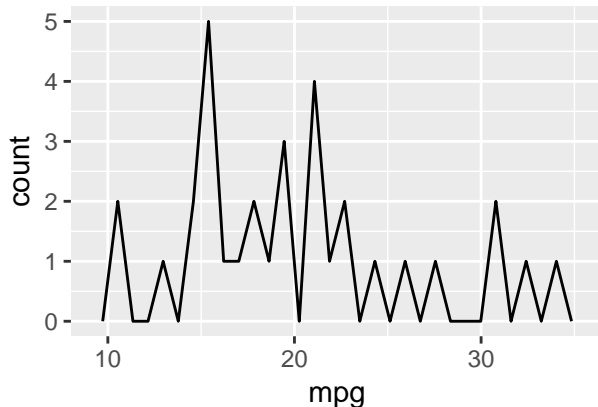
```
## `stat_bindot()` using `bins = 30`. Pick better value with
```



Ejemplo geoms II - geom_freqpoly()

```
myplot + geom_freqpoly()
```

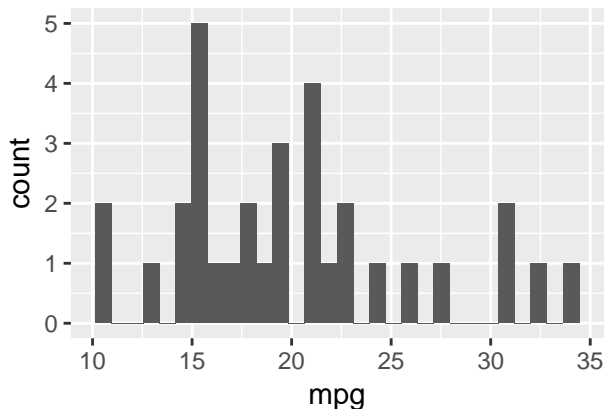
```
## `stat_bin()` using `bins = 30`. Pick better value with `
```



Ejemplo geoms II - geom_histogram()

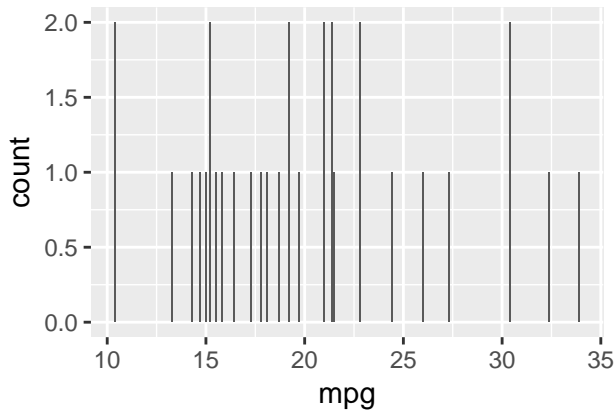
```
myplot + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `
```



Ejemplo geoms II - geom_bar()

```
myplot + geom_bar()
```

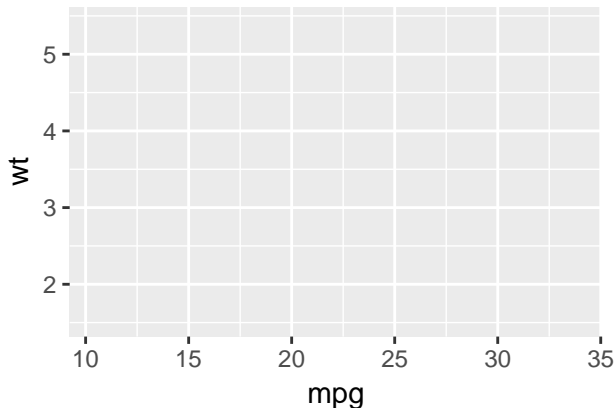


Ejemplo geoms III - base

```
#Establecer la base
```

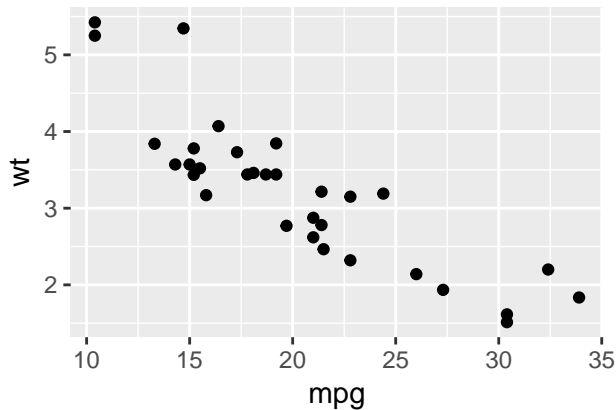
```
myplot <- ggplot(data = mtcars, aes(x = mpg, y=wt))
```

```
myplot
```



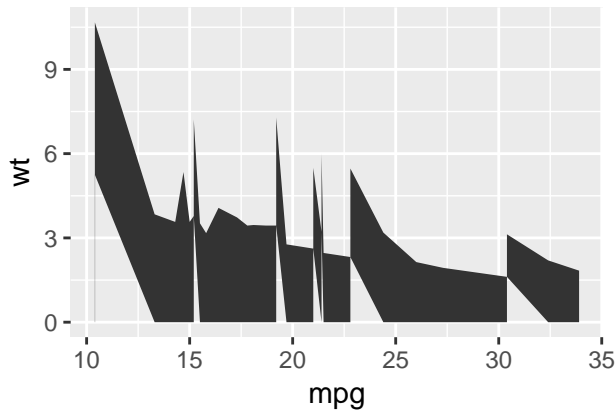
Ejemplo geoms III - geom_point()

```
myplot + geom_point()
```



Ejemplo geoms III - geom_area()

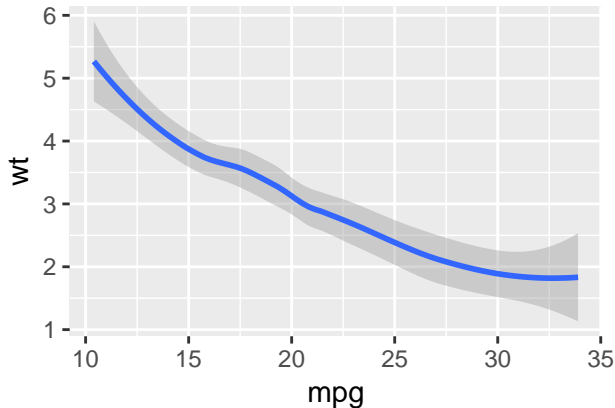
```
myplot + geom_area()
```



Ejemplo geoms III - geom_smooth()

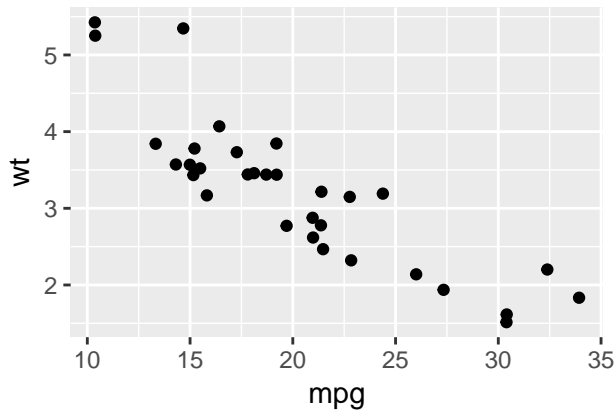
```
myplot + geom_smooth()
```

```
## `geom_smooth()` using method = 'loess'
```



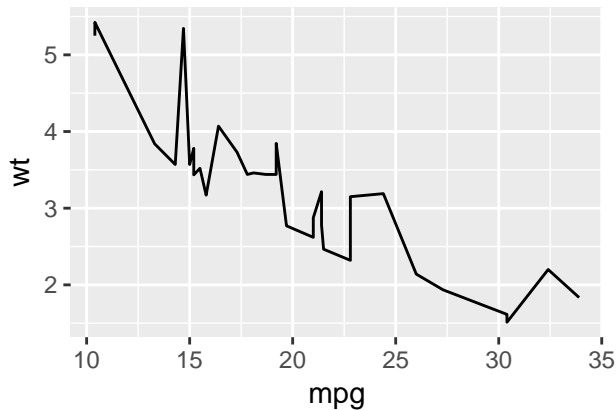
Ejemplo geoms III - geom_jitter()

```
myplot + geom_jitter()
```



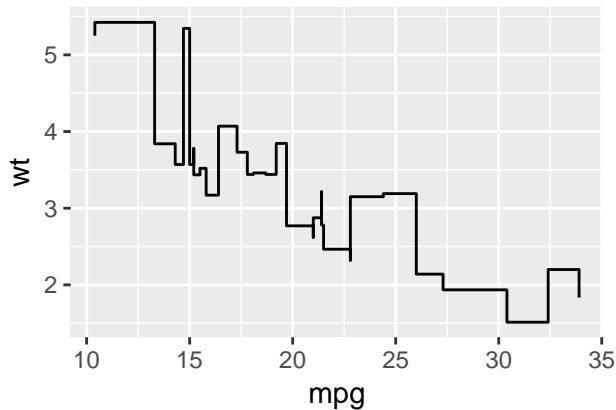
Ejemplo geoms III - geom_line()

```
myplot + geom_line()
```



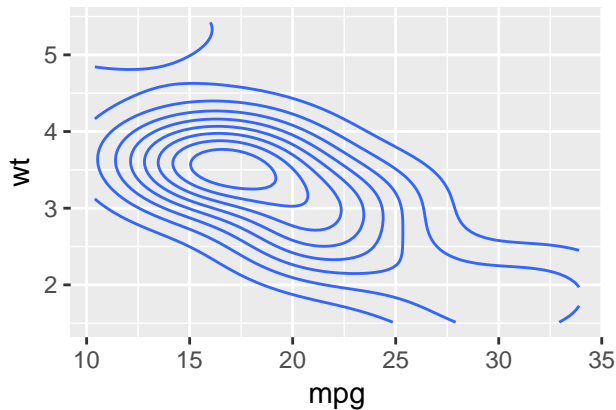
Ejemplo geoms III - geom_step()

```
myplot + geom_step()
```



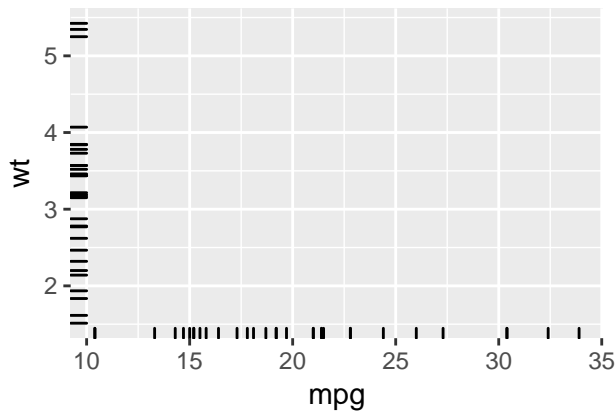
Ejemplo geoms III - geom_density2d()

```
myplot + geom_density2d()
```



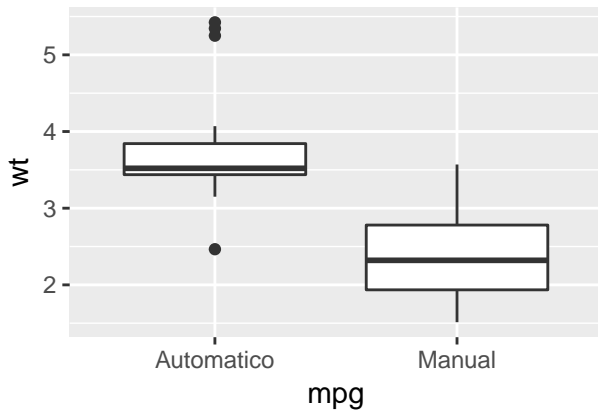
Ejemplo geoms III - geom_rug()

```
myplot + geom_rug()
```



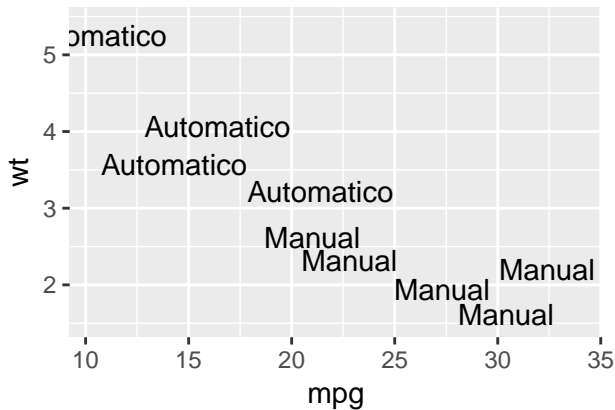
Ejemplo geoms II - geom_boxplot()

```
myplot + geom_boxplot(aes(x=am))
```



Ejemplo geoms III - geom_text()

```
myplot + geom_text(aes(label=am), check_overlap = T)
```



Guía de referencia ggplot2

<http://ggplot2.tidyverse.org/reference/index.html>

¡Práctica!

La gramática de vuelta

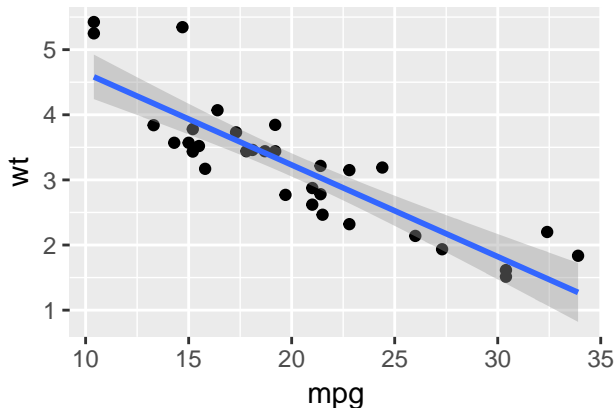
- ▶ Elementos de la gramática de gráficos en ggplot2:
 - ▶ Data
 - ▶ Aesthetics
 - ▶ Geometries
 - ▶ **Statistics**
 - ▶ Facets
 - ▶ Coordinates
 - ▶ Theme

Statistics

- ▶ Para hacer gráficos sobre estadísticos: medias, intervalos de confianza
- ▶ En otras ocasiones los estadísticos son computados directamente por el programa en el *background*, al estar asociados a un determinado geom (e.g. `geom_smooth`, `geom_histogram...`)
- ▶ La capa stats puede ser utilizada de forma independiente o desde dentro de un geom
- ▶ La capas de stats se llaman a través de las funciones `stat_*()`

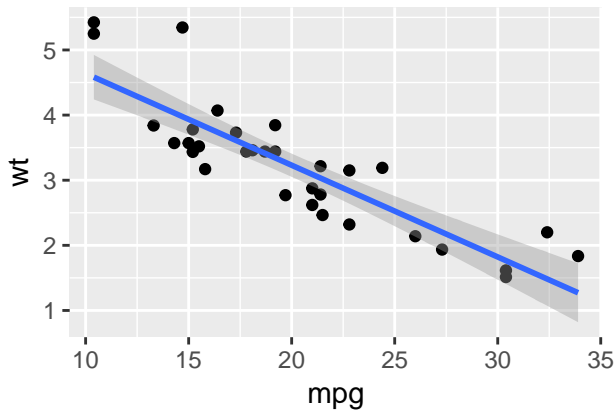
Ejemplo statistics I

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



Ejemplo statistics II

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



stat_summary()

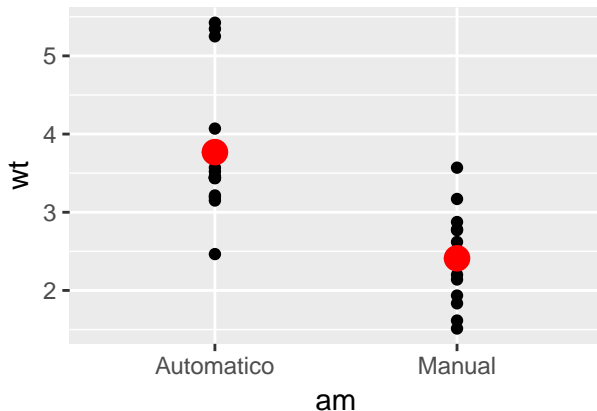
- **stat_summary()** es una función que permite resumir los valores de y para x.

```
myplot + stat_summary(fun.y = "MiFunction",  
                      fun.args = list(ArgumentoA = A,  
                                      ArgumentoB= B),  
                      geom = "MiGeom")
```

- Por ejemplo, para representar los valores medios de wt para las categorías de am

Ejemplo stat_summary()

```
myplot <- ggplot(mtcars, aes(x=am, y=wt)) +  
  geom_point()  
myplot + stat_summary(fun.y = "mean",  
  geom = "point", size=4, col="red")
```



¡Proyecto!