

Visualizacion de Datos con ggplot2

Lavinia Lavin

2024-08-06

##Este es el script para el taller de Introduccion a visualizacion de datos: Comenzamos con los requisitos

```
install.packages("tidyverse")
install.packages("ggplot2")
```

Llamar de la libreria ggplot2, con este paquete estaremos trabajando:

```
library(ggplot2)
```

Vamos a trabajar con el data frame que ggplot ofrece donde podemos observar una serie de datos que corresponden a las características de autos, incluyendo displ (tamaño de motor) y hwy (eficiencia de kilometraje)

```
ggplot2::mpg
```

```
## # A tibble: 234 x 11
##   manufacturer model    displ  year  cyl trans drv    cty   hwy fl    class
##   <chr>         <chr>    <dbl> <int> <int> <chr> <chr> <int> <int> <chr> <chr>
## 1 audi         a4         1.8  1999    4 auto~ f      18    29 p    comp~
## 2 audi         a4         1.8  1999    4 manu~ f      21    29 p    comp~
## 3 audi         a4         2    2008    4 manu~ f      20    31 p    comp~
## 4 audi         a4         2    2008    4 auto~ f      21    30 p    comp~
## 5 audi         a4         2.8  1999    6 auto~ f      16    26 p    comp~
## 6 audi         a4         2.8  1999    6 manu~ f      18    26 p    comp~
## 7 audi         a4         3.1  2008    6 auto~ f      18    27 p    comp~
## 8 audi         a4 quattro  1.8  1999    4 manu~ 4      18    26 p    comp~
## 9 audi         a4 quattro  1.8  1999    4 auto~ 4      16    25 p    comp~
## 10 audi        a4 quattro  2    2008    4 manu~ 4      20    28 p    comp~
## # i 224 more rows
```

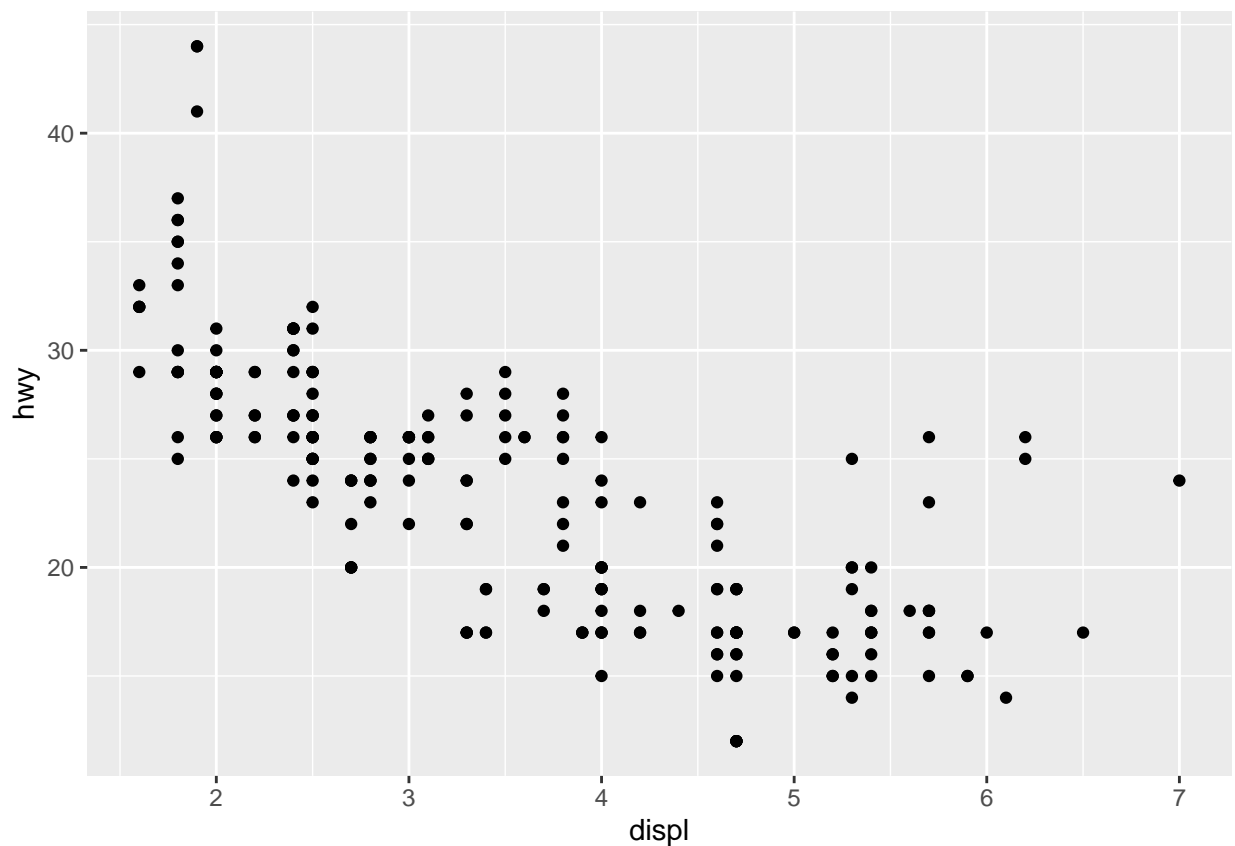
```
print(mpg)
```

```
## # A tibble: 234 x 11
##   manufacturer model    displ  year  cyl trans drv    cty   hwy fl    class
##   <chr>         <chr>    <dbl> <int> <int> <chr> <chr> <int> <int> <chr> <chr>
## 1 audi         a4         1.8  1999    4 auto~ f      18    29 p    comp~
## 2 audi         a4         1.8  1999    4 manu~ f      21    29 p    comp~
## 3 audi         a4         2    2008    4 manu~ f      20    31 p    comp~
## 4 audi         a4         2    2008    4 auto~ f      21    30 p    comp~
```

```
## 5 audi      a4      2.8 1999    6 auto~ f      16    26 p    comp~
## 6 audi      a4      2.8 1999    6 manu~ f      18    26 p    comp~
## 7 audi      a4      3.1 2008    6 auto~ f      18    27 p    comp~
## 8 audi      a4 quattro 1.8 1999    4 manu~ 4      18    26 p    comp~
## 9 audi      a4 quattro 1.8 1999    4 auto~ 4      16    25 p    comp~
## 10 audi     a4 quattro 2    2008    4 manu~ 4      20    28 p    comp~
## # i 224 more rows
```

Comenzaremos con nuestra primera tabla, simple; comenzamos con la función `ggplot()` SIEMPRE, esto nos permite coordinar sistemas para añadir capas. Los datos los llamamos con la función `data=mpg` ya que estamos trabajando con la base de datos `mpg`, `geom_point()` nos da la capacidad de trabajar con puntos en la tabla, esto nos crea un scatter plot (o un gráfico disperso). El argumento `mapping` siempre va acompañado de `aes()`, esto nos va a permitir definir el eje x y y

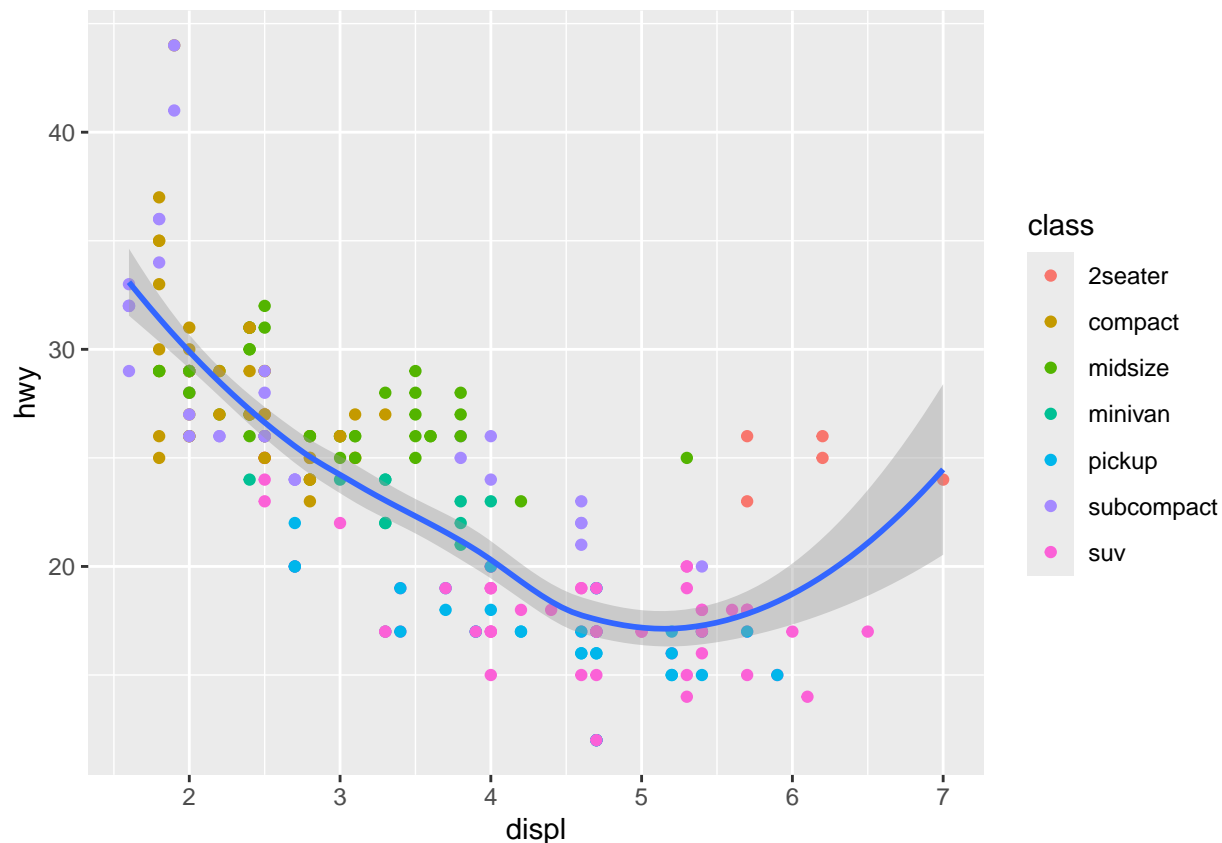
```
ggplot(data=mpg)+
  geom_point(mapping = aes(x=displ, y=hwy))
```



Ahora vamos a pasar a hacer un plot un poco más complejo donde `geom_smooth()` es una capa que permite diferenciar la tendencia de la distribución de los datos

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

```
## 'geom_smooth()' using method = 'loess' and formula = 'y ~ x'
```



Pero con R nos podemos facilitar la vida, existe un paquete llamado `esquisse` que permite jugar con lo datos, vamos a probarlo usando la funcion `esquisser()`

```
install.packages("esquisse")
library(esquisse)
esquisser()
```

Podemos hacer graficos dinamicos usando la paqueteria de “plotly” llamamos al objeto de la grafica que vamos a crear, en este caso vamos a añadir titulos a los ejes El objeto se llamara `p`, para observarlo solo incluimos el objeto `p`

```
install.packages("plotly")
library(plotly)
p <- ggplot(mpg, aes(x = displ, y = hwy, color = class, text = paste("Fabricante: ", manufacturer, "<br>
  geom_point(size = 3) +
  labs(
    title = "Relación entre Desplazamiento del Motor y Eficiencia de Combustible",
    x = "Desplazamiento del Motor (litros)",
    y = "Millas por Galón en Carretera (hwy)",
    color = "Clase de Vehículo"
  ) +
  theme_minimal()
p
```

Nos da como resultado esta grafica, entonces para hacerla dinamica usamos el comando `p_interactive` y `tooltip` para mostrar lo que queremos que se vea cuando movamos el cursor

```
p_interactive <- ggplotly(p, tooltip = "text")
```

Llamamos al objeto que creamos y tenemos nuestro grafico dinamico!

```
p_interactive
```