

Tarea - Geometrías con ggplot2

Jesus Mudarra Luján

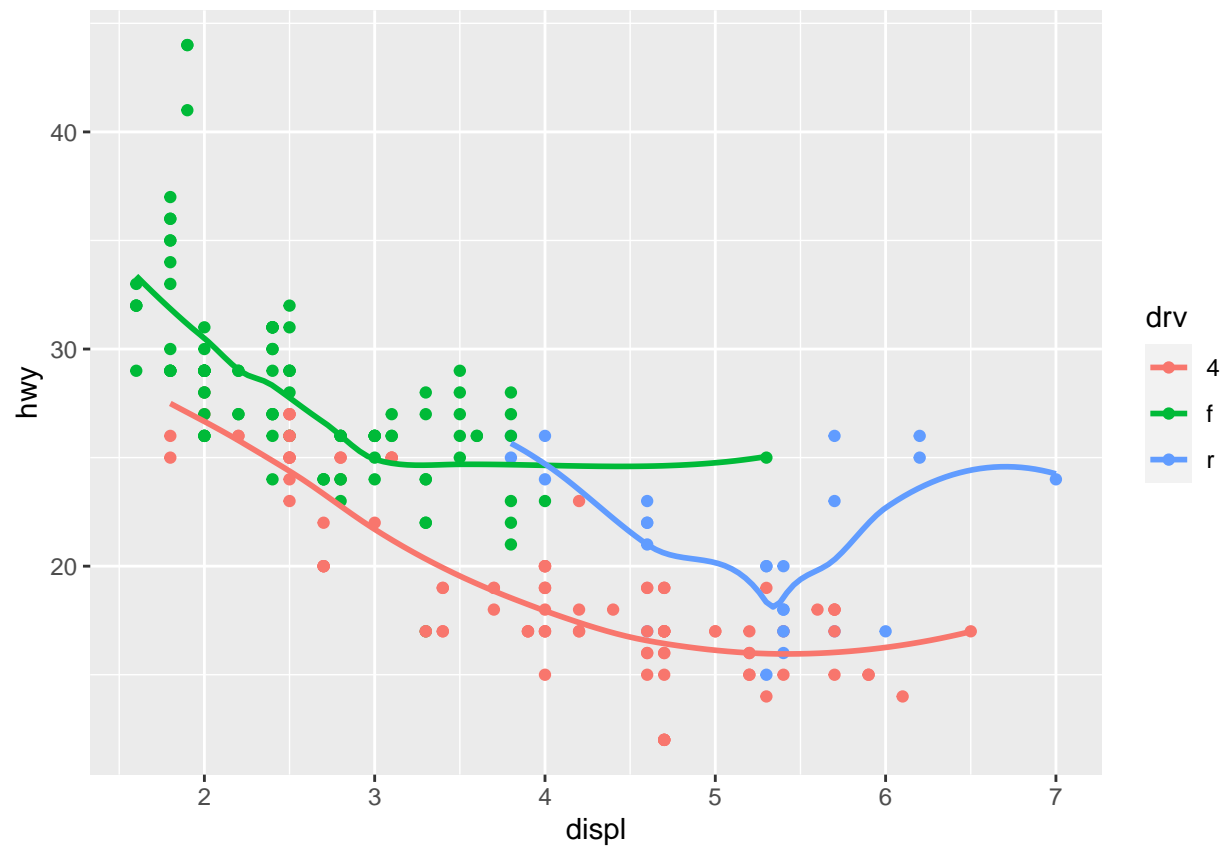
2022-09-06

Ejercicio 1

Ejecuta este código en tu cabeza y predice el resultado. Luego ejecutalo en R y comprueba tu hipótesis:

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

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



Ejercicio 2

¿Qué hace el parámetro `show.legend = F`? ¿Qué pasa si lo eliminamos? ¿Cuándo lo añadirías y cuándo lo quitarías?

Elimina la leyenda. Si lo eliminamos se mostrará la leyenda ya que es un parámetro por defecto. La añadiría cuando sirva de apoyo en la comprensión del gráfico, sobre todo en los casos que hay múltiples puntos o curvas.

Ejercicio 3

¿Qué hace el parámetro `se` de la función `geom_smooth()`? ¿Qué pasa si lo eliminamos? ¿Cuándo lo añadirías y cuándo lo quitarías?

El parámetro `se` incluye o elimina los intervalos de confianzas de las curvas. Lo añadiría siempre que sea relevante mostrar el intervalo de confianza y no interfiera en el análisis del gráfico.

Ejercicio 4

Describe qué hacen los dos siguientes gráficos y di si serán igual y diferente. Justifica tu respuesta.

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

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

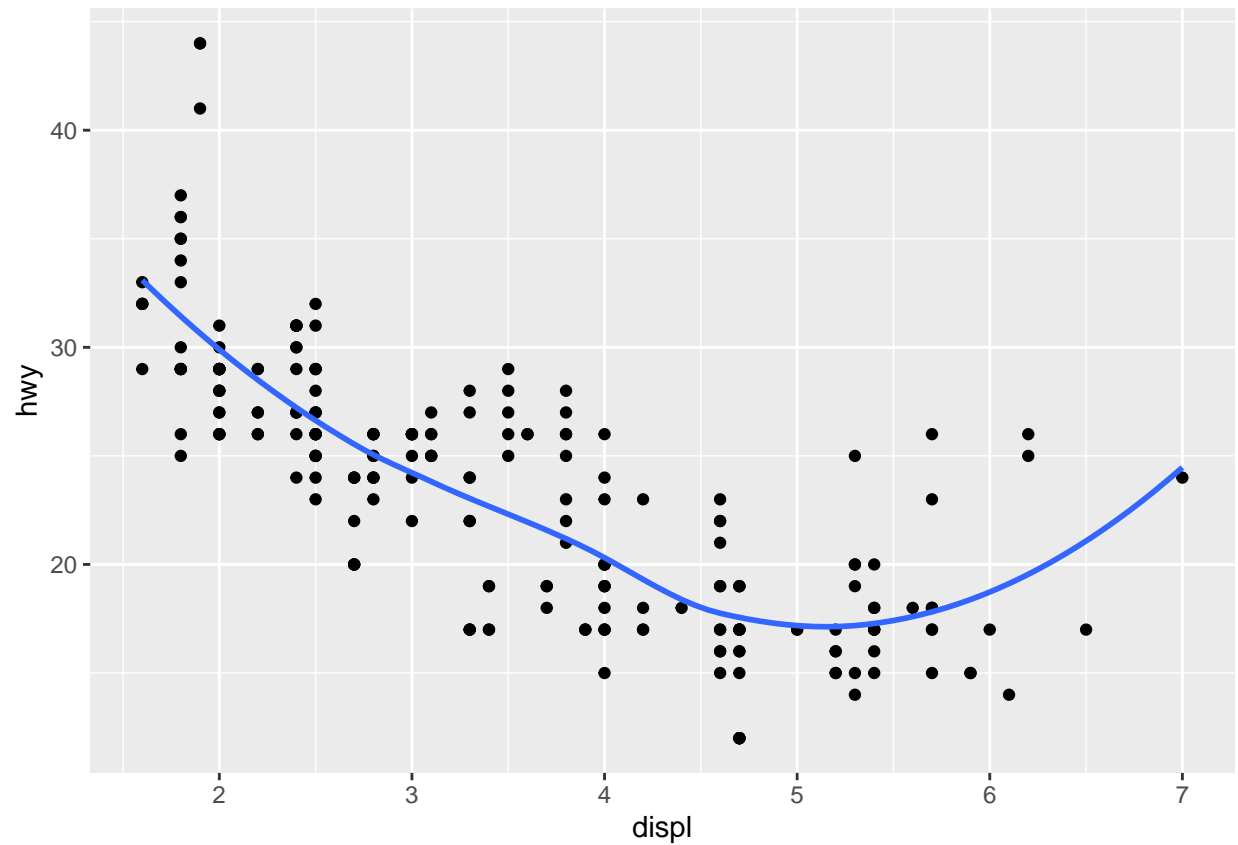
Los dos gráficos hacen exactamente lo mismo. La única diferencia es que en el primer caso se han incluido mappings globales en el `ggplot()` lo cual ayuda a no repetir código y realizar modificaciones de una manera más sencilla.

Ejercicio 5

Reproduce el código de R que te genera el siguiente gráfico.

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

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

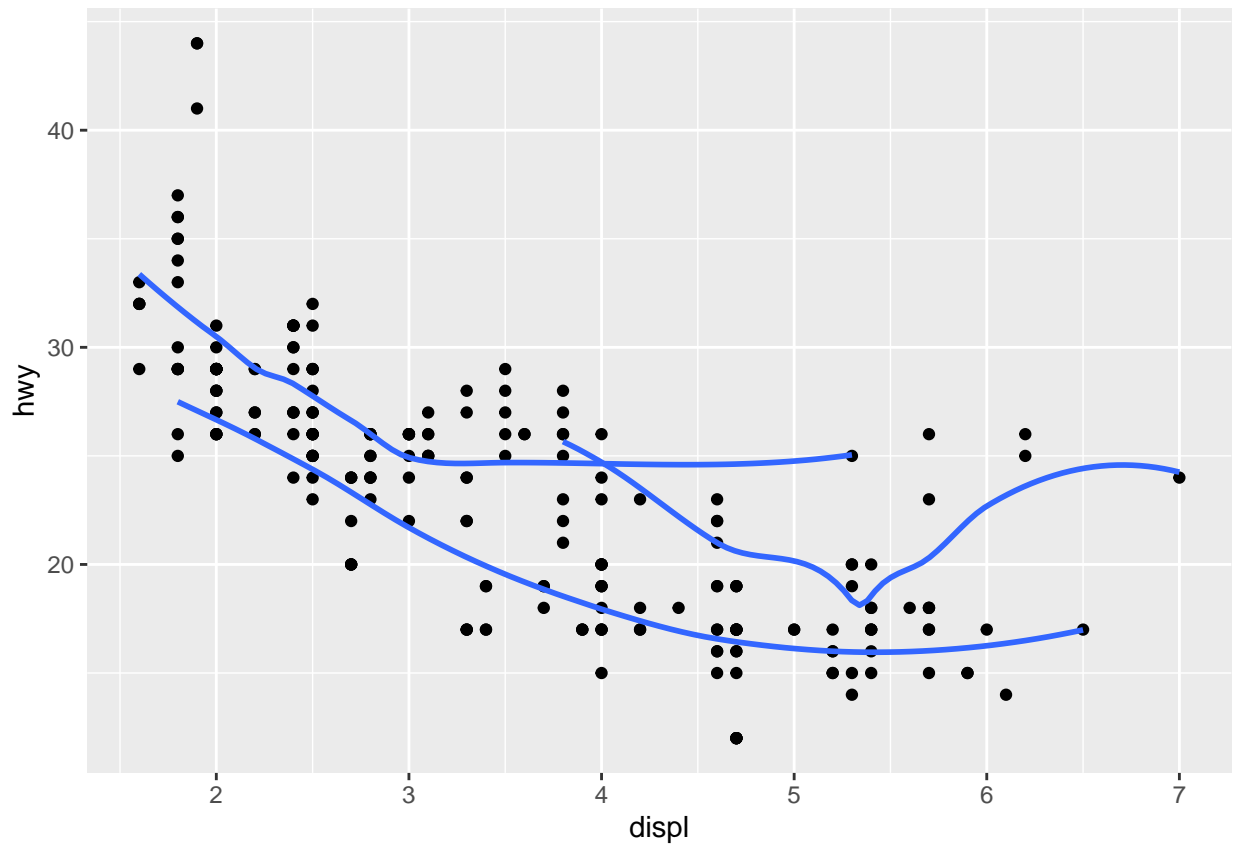


Ejercicio 6

Reproduce el código de R que te genera el siguiente gráfico.

```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +  
  geom_point() +  
  geom_smooth(mapping = aes(group = drv), se = F)
```

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

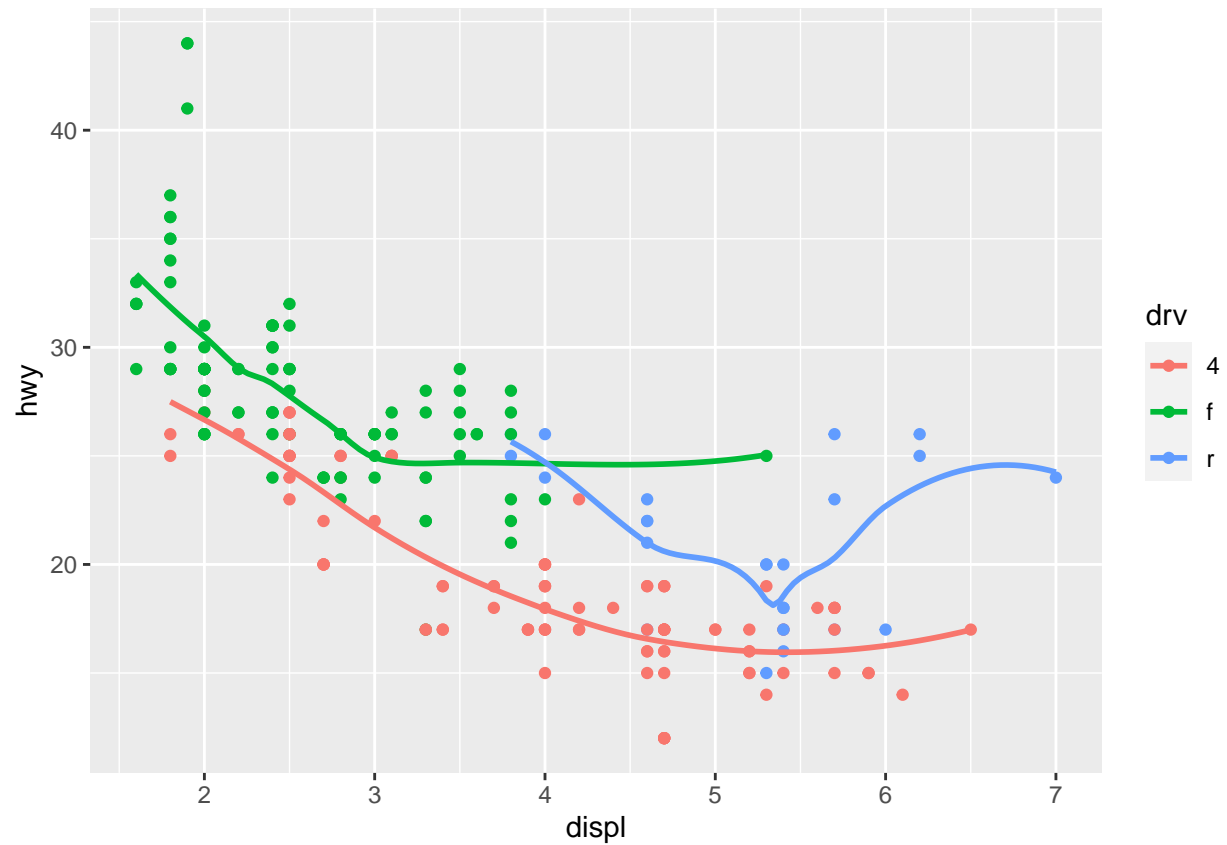


Ejercicio 7

Reproduce el código de R que te genera el siguiente gráfico.

```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy, color = drv)) +  
  geom_point() +  
  geom_smooth(mapping = aes(group = drv), se = F)
```

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

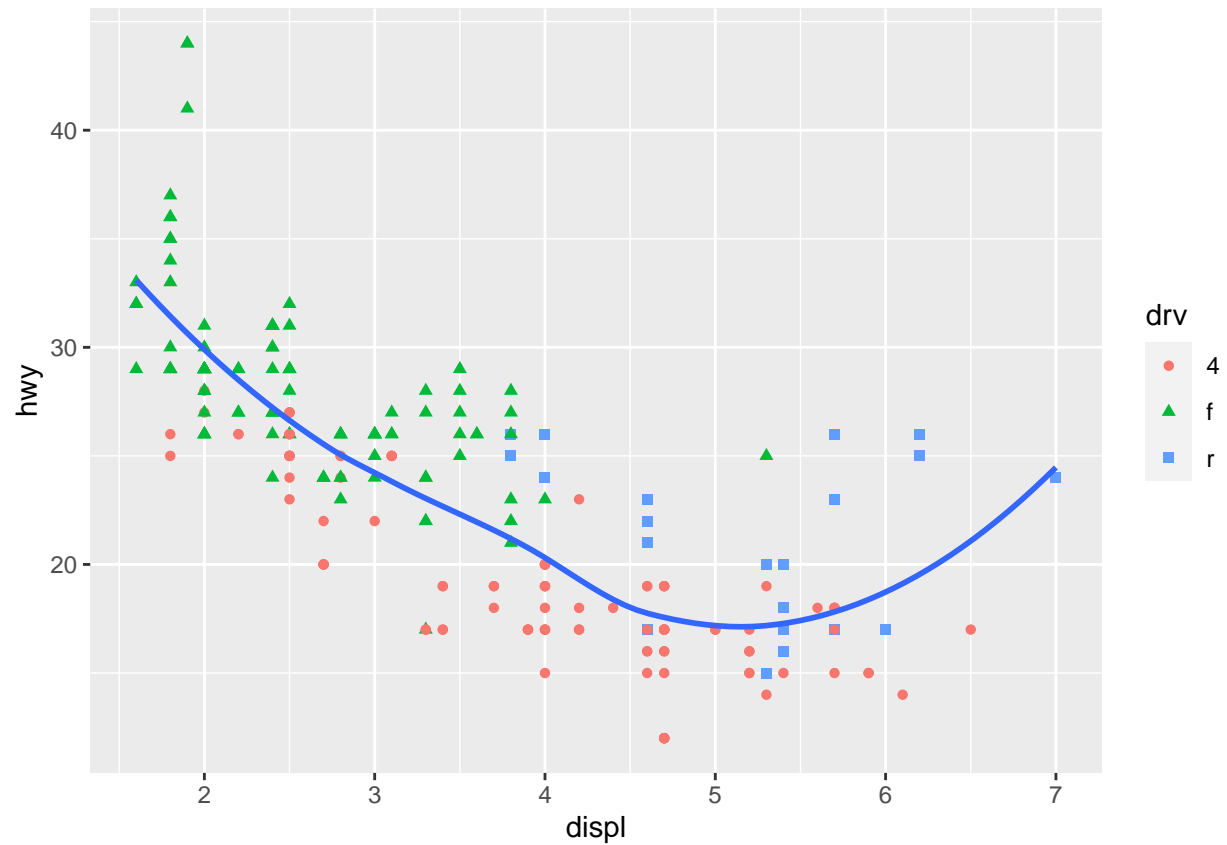


Ejercicio 8

Reproduce el código de R que te genera el siguiente gráfico.

```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point(mapping = aes(shape = drv, color = drv)) +
  geom_smooth(se = F)
```

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

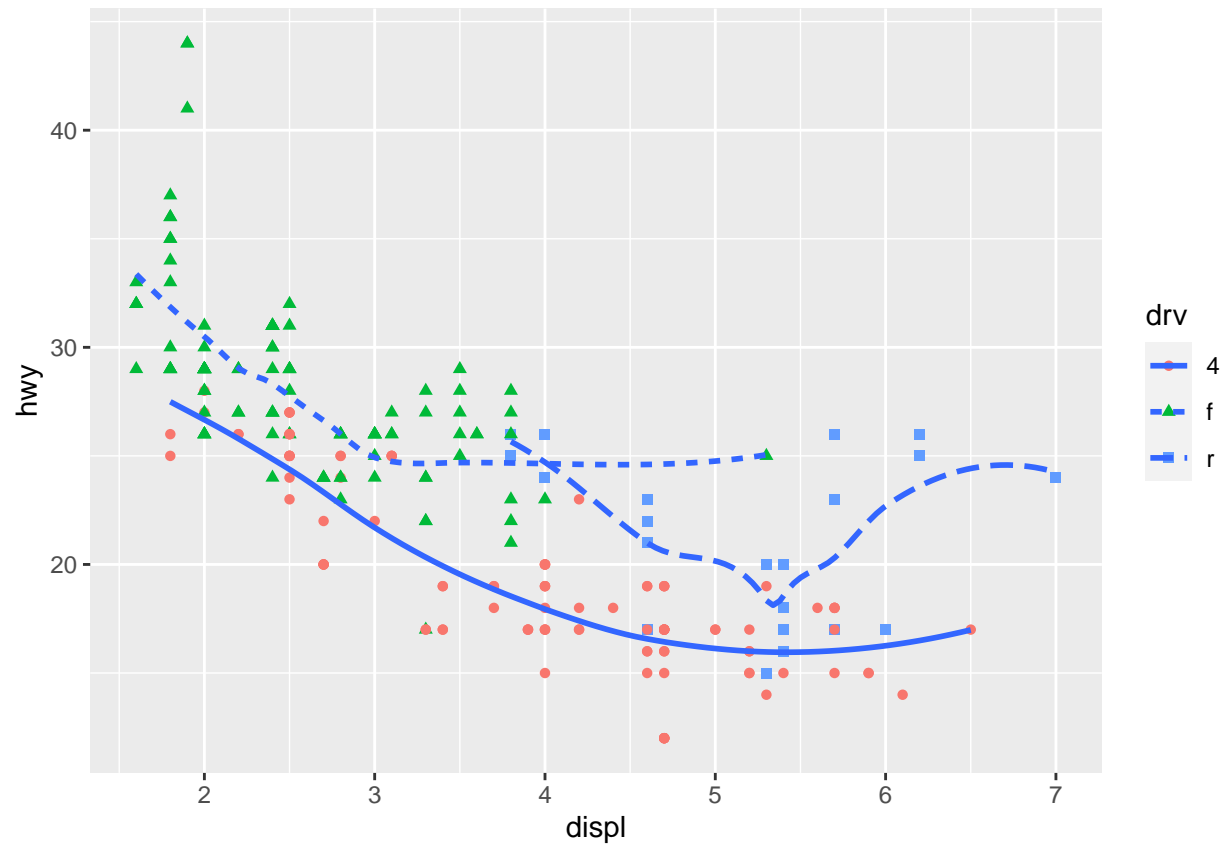


Ejercicio 9

Reproduce el código de R que te genera el siguiente gráfico.

```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point(mapping = aes(shape = drv, color = drv)) +
  geom_smooth(mapping = aes(linetype = drv), se = F)
```

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



Ejercicio 10

¡Este va para nota!

Reproduce el código de R que te genera el siguiente gráfico. Investiga algunos parámetros adicionales que te harán falta de ggplot2 como `stroke` entre otros.

```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy, fill = drv)) +
  geom_point(shape = 23,
             size = 4,
             stroke = 2,
             color = "white")
```

