

Práctica 4: más ggplot2

En los materiales de hoy encontrarás el fichero cis3145t.sav. Se trata de un fichero del estudio poselectoral de las elecciones de junio de 2016. También encontrarás en la carpeta un fichero Excel con la descripción de las variables.

En esta vas a trabajar sobre aspectos adicionales de ggplot2 como las facets y theme. Las soluciones al ejercicio las tienes en el fichero Practica4_resuelta.R.

1. En primer lugar, abre el fichero de código Practica4.R . Carga los paquetes que necesitas para realizar la práctica, ejecutando las líneas de `library()`. En caso de que alguno de ellos no esté instalado, retira la `#` de delante del comando `install.packages()`, y ejecuta esa línea del código.
2. Una vez cargadas las librerías, limpia el espacio de datos utilizando `rm(list=ls())`, con esto eliminarás del espacio de trabajo todos los datos que estén abiertos, y así evitarás confusiones. Después utiliza la función `setwd()` para establecer tu carpeta de trabajo, donde has copiado los datos "cis3145t.sav". Recuerda que las barras para indicar la ruta deben ser `"/"`, y que la ruta a la carpeta debe ir entre comillas `" "`.

Ejemplo:

```
setwd("C:/Mis Documentos/Curso R")
```

Gráfico 1.

1. En este gráfico vas a trabajar sobre la relación entre `confpub` y `edad`, añadiendo una tercera variable: estatus social (`esta`). Esta tercera variable la vamos añadir utilizando las facets.

2. Para empezar, ejecuta la transformación de la variables y crea un `ggplot()` utilizando los datos de `d`, en el que la variable x sea `edad`, y la variable y sea `confpub`.
3. En el siguiente paso añade un `geom_point()`, para crear el diagrama de dispersión. En este `geom_point()`, especifica los atributos: `position="jitter"`, `alpha=.3`, y `shape=1`. Estos tres argumentos, que ya vimos la práctica 3, ayudan a mejorar la visualización de los puntos superpuestos.
4. En el tercer paso, añade al gráfico una línea de regresión, ya sabes que puedes hacerlo añadiendo en la nueva capa `geom_smooth()`. Dentro de la función, especifica el argumento `method="lm"`, para que la línea sea una recta de regresión.
5. En el cuarto paso vas a añadir el `facet_wrap()`. El objetivo es que la tercera variable, estatus social (`esta`), haga de divisora, creándose un gráfico de dispersión para cada categoría de estatus social. Para ello incluye en la nueva capa:

`facet_wrap(~ esta)`
6. Por último, añade las etiquetas correspondientes. Eje X "Edad", eje Y "Confianza en las instituciones públicas", y el título del gráfico "G1. Relación entre la edad y la confianza en las instituciones públicas por clase social". Para ello puedes utilizar la función `labs()`. Imprime el gráfico final.

Gráfico 2.

1. En el gráfico 2 vas a comparar diferentes tipos de gráficos de barras. Estos gráficos van a representar, para cada categoría de estatus social (`esta`), la proporción de votantes y abstencionistas en las elecciones de 2016 (urnas16r). Para ello se ha creado un nuevo data frame, `d2`, en el

que se excluyen los casos perdido en urnas16r. En este gráfico también vas a ver como con la misma base se pueden construir varios gráficos.

2. En primer lugar establece la base del gráfico usando ggplot(), en la aes, x = esta y fill = urnas16r. De esta forma estamos pidiendo que en el eje X se represente la variable estatus social, y que estas barras estén rellenas según los valores de urnas16r. Dicho de otra forma, dentro de cada grupo de estatus social vamos a ver en qué medida fueron o no a votar en 2016.
3. Una vez que has creado el objeto g2 – la base del gráfico, vas a producir el objeto:

```
g2a <- g2 + geom_bar() +  
  scale_x_discrete(labels = c("Alta",  
    "Viejas Med", "Nuevas Med.", "Obr. cual", "Obr.  
    no cual", "NC")) +  
  labs(x="", y="", fill="")
```

Como ves, en el objeto **g2a** incluimos el geom bar, para crear el gráfico de barras, cambiamos las etiquetas de las categorías de esta a través de **scale_x_discrete()**, y además retiramos todas las etiquetas de los ejes utilizando **labs()**. Crea el objeto **g2a** e imprime el resultado, ¿cómo se representa urnas16r dentro de las categorías de estatus social?



4. Hay diferentes formas en las que se puede representar el relleno "fill" en el gráfico de barras. Estas representaciones vienen dadas por el argumento **position=**, que se coloca dentro de la función **geom_bar()**. Los valores que puede tomar el argumento position son tres: **"stack"**, **"dodge"**, o **"fill"**. En el paso G2.C) vas a crear el

objeto **g2b**, otro gráfico como el anterior con la única salvedad de que dentro de **geom_bar()** vas a incluir el argumento **position="stack"**. Imprime el gráfico.

5. En el siguiente paso vas a crear el objeto **g2c**, en el que vas a cambiar el argumento position a **"dodge"**. Imprime el gráfico.
6. En este paso vas a crear el gráfico **g2d**, en el que el argumento position será igual a **"fill"**. Imprime el gráfico.
7. Como has visto hay diferentes formas de representar una variable nominal dentro de la variable principal (x) del gráfico de barras. Para acabar vamos a fusionar los cuatro gráficos que hemos creado, para ello vas a utilizar la función **grid.arrange()**, para crear el objeto **g2comb**:

```
g2comb <- grid.arrange(g2a, g2b, g2c, g2d,  
  ncol=2, top=quote("G4. Voto en 2016 por estatus  
  social "))
```

Ejecútalo e imprime. Ahora que puedes comparar los cuatro gráficos, ¿cuál te parece más adecuado para alcanzar el objetivo que nos planteamos al principio del ejercicio?

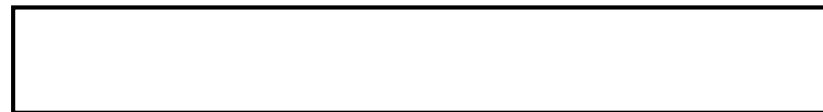


Gráfico 3.

1. El gráfico 3 ya lo has realizado con anterioridad, por ello tienes el código completo que puedes ejecutar hasta el paso G3.b). Este gráfico representa

la relación entre `confpub` y `confpriv`. Para completar el gráfico de dispersión vamos a crear unas líneas de regresión, pero en vez de utilizar `geom_smooth()`, como en la práctica anterior, vas a usar `stat_smooth()`.

- En el paso G3.c) tienes que completar el código, de forma que añadas una línea de regresión, para ello añade:

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

- Existe el problema de que se ha dibujado una línea de regresión para cada subgrupo, pero no una general. Para añadir la general, habrá que crear otra capa con `stat_smooth()`, en la que además se agregue el argumento `aes(col=factor(1))`. Añade la nueva capa de `stat_smooth` e imprime el resultado.
- En el paso G3.e) los colores de la escala son modificados, así como las etiquetas. Date cuenta de que al primer grupo (Total) se le asigna el color rojo, al segundo (No votó) el negro, y al tercero (Votó) el "skyblue". Si ejecutas `g3`, cada línea de regresión habrá tomado un color.
- En el paso G3.f) se añaden las etiquetas, solo tienes que ejecutarlo.
- En el paso G3.g) vas a ir incluyendo diferentes aspectos de `theme`, la recomendación es que lo vayas haciendo uno a uno, e imprimas el gráfico tras cada transformación para que te vayas dando cuenta de cómo se van produciendo los cambios. Esto serían los primeros dos pasos:

```
g1b <- g1b + theme(legend.position= "bottom")  
g1b
```

```
g1b <- g1b + theme(legend.title=element_blank() )  
g1b
```

En el primer paso cambiar la posición de la leyenda a la parte inferior del gráfico, esta parte es la única que no utiliza algunos de los argumentos del tipo `element_*`. En segundo lugar, vas a quitar el título. Puedes copiar estos dos trozos de sintaxis y ejecutarlos. Imprime el gráfico.

- Lo siguiente será quitar el grid, esto se especifica a través del argumento `panel.grid`. En este caso, como queremos quitarlo, se aplicará `element_blank()`. Imprime el gráfico.

- Lo siguiente es cambiar el color de los ejes y sus etiquetas a gris. Los ejes se dividen en cuatro elementos:

```
axis.text = element_text()  
axis.title = element_text()  
axis.ticks = element_line()  
axis.line = element_line()
```

Para estos cuatro elementos, especifica `colour="gray"`. Con ello habrás cambiado la apariencia, en este caso el color, de ambos ejes.

- Para continuar vas a cambiar el color del fondo del gráfico de gris a blanco, para ello, el elemento a modificar es `panel.background()`. Este elemento es un rectángulo, por lo tanto el argumento es `fill="white"`. Imprime el gráfico.
- Por último vas a cambiar el color del título del gráfico, para ello puede usar el argumento `plot.title()`, que se trata de un elemento de texto. Haz que el color sea igual a `"gray31"`.

Gráfico 4.

- En este gráfico vuelves a tener el código preparado, a excepción del último paso: `themes`. En este caso, en vez de ser tu quien establece el tema manualmente, prueba diferentes temas preparados. Puedes probar los

que vienen con ggplot, pero también los del paquete ggtheme. No olvides, que aunque utilices uno de los temas predefinidos, puedes seguir utilizando theme para cambiar aquello que quieras.

2. Esta es la lista de themes que puedes probar, no te olvide de imprimir los gráficos cada vez que cambies el tema:

- `theme_base()`
- `theme_calc()`
- `theme_economist()`
- `theme_excel()`
- `theme_few()`
- `theme_fivethirtyeight()`
- `theme_gdocs()`
- `theme_hc()`
- `theme_par()`
- `theme_pander()`
- `theme_solarized()`
- `theme_stata()`
- `theme_tufte()`
- `theme_wsj()`