Práctica 3: ggplot2 data, aesthetics, geoms y stats

Abre el script Practica3.R, en esta práctica también utilizaremos los datos del archivo "**cis3145t.sav**". Este ejercicio consiste en la generación de cuatro gráficos centrados en los aspectos de datos, aesthetics, y geoms. Las soluciones al ejercicio las tienes en el fichero PracticaDia3_resuelto.R.

1. En primer lugar, abre el script Pracitica3.R . Carga los paquetes que necesitas para realizar la práctica, ejecutando las líneas de **library**(), en caso de que no estén cargados ya (e.g. ggplot2). En caso de que alguno de ellos no esté instalado, quita la # de delante del comando **install.packages**(), y añade comillas al nombre del paquete.

2. Una vez cargadas las librerías, limpia el espacio de datos utilizando **rm(list=ls())**, con esto quitarás 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, en caso de que no lo hayas hecho ya. 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")
   ```

3. Carga el archivo "**cis3145t.sav**" como un data frame y asígnalo al objeto **d**.

Gráfico 1.

1. Este primer ejercicio consiste en generar dos índices, uno de confianza en instituciones públicas, y otro de confianza en instituciones privadas. Estos índices son variables cuantitativas, y queremos observar si están relacionados, es decir, si a mayor nivel de confianza en instituciones públicas, mayor nivel también de confianza en instituciones privadas. También se quiere observar a través de la visualización si hay alguna relación entre la afección política, medido como votar o no votar en las elecciones de 2016 y la relación entre estas dos variables de confianza. Con esta información, completa la siguiente tabla:

Aspecto	Descripción
Datos	
Aesthetics	
Geoms	

2. El primer paso será preparar las variables. Ten en cuenta que necesitamos tres variables: **confpub** (compuesta por confparl, confpart y confjudic), **confpriv** (compuesta por confbanco, confmedia y confong), y votó/no votó (**urnas16r**) en las elecciones de 2016 (recodificación de voto16). La sintaxis está lista, solo tienes que ejecutarla.

3. Para comenzar el gráfico establece la primera capa: datos y aesthetics. Completa los huecos en la sintaxis. Ten en cuenta que "col" se refiere al color Imprime el gráfico.

4. El siguiente paso será añadir la geom principal, en este caso para hacer un gráfico de dispersión entre **confpriv** y **confpub** utiliza **geom_point**(). Imprime el gráfico, ¿qué problemas presenta?

5. Para superar el problema de que los **NA** y los NC de **urnas16r** han sido incluidos en el gráfico, comienza el gráfico de nuevo (**g1b**), pero usando un **filter**() para seleccionar los casos que no tengan **NA** en **voto16r**. En este caso, los datos en **ggplot**(), el primer argumento, tiene que ser sustituidos por:

```
filter(d, !is.na(urnas16r), urnas16r != "NC")
```

6. Ahora se puede añadir una nueva capa, el geom: **geom_point**(). Dentro de **geom_point**() vas a añadir tres atributos para solucionar el problema de los puntos superpuestos. Primero, `position = position_jitter(0.1)` añade a los puntos un componente de variación aleatoria, que mitiga el problema de la superposición. Segundo, `alpha` va de 0 a 1, y establece el nivel de transparencia, establécelo en 0.3. Por último, `shape=1` es una forma redonda hueca recomendada para cuando hay una densidad alta de puntos. Imprime el gráfico.

7. En el siguiente paso vas a añadir una línea de tendencia general a los datos que ajuste los puntos. Para ellos, añade una nueva capa con la geom **geom_smooth**(). Dentro del paréntesis especifica el método que debe usar el programa para crear la línea, en este caso es método lineal: `method="lm"`. Imprime el gráfico, ¿qué problema hay?

8. Lo que ha ocurrido es que, para cada geom, a no ser que se especifique lo contrario, hereda la aesthetics que especificamos al principio en **ggplot**(). En esa aesthetics se establecieron las variables `x=confpub`, `y=confpriv`, y `col=unrnas16r`. Con esa aesthetics, al aplicar **geom_smooth**(), se crean dos líneas, una para cada valor de la variable `urnas16r`. Para crear una línea de tendencia para todos los datos, sin diferencias por subgrupos, hay que especificar una nueva aesthetics dentro de **geom_smooth**(), completa la sintaxis:

```
g1b <- g1b + geom_smooth(aes(col=factor(1)), method="lm")
```

Imprime el gráfico, como verás ahora hay tres líneas de ajuste.

9. Lo siguiente que puedes hacer es cambiar los colores de la escala utilizada, y añadir las etiquetas que aparecerán en la leyenda, así como el nombre de esta. Para ello trabaja con las funciones **scale_*_***(). Recuerda que las escalas están relacionadas con aspectos de la aesthetics, en este caso hemos utilizado el color, por lo tanto, utiliza **scale_colour_***(). Por último, en este caso, vas a establecer la escala de forma manual, por lo tanto la función a utilizar es: **scale_colour_manual**(). Esta función tiene dos argumentos principales, primero `values`, en el que se establecen los colores que tomarán los elementos afectados por la aesthetics color. Después, el argumento `labels` se utiliza para establecer las etiquetas de los valores que aparecen en la leyenda. En nuestro caso completa con:

```
g1b <- g1b + scale_colour_manual(values=c("red", "black", "skyblue"), labels=c("Total", "No votó", "Votó"))
g1b
```

Imprime el gráfico.

TIP! Para saber los nombres de los colores puedes usar esta web: http://sape.inf.usi.ch/quick-reference/ggplot2/colour

10. Por último, cambia los títulos de los ejes X, Y, y la leyenda. Además, hay que añadir un título al gráfico. Para ello añade otra capa con la función **labs**(). Dentro de la función, los argumentos son los diferentes aspectos de la aesthetics, en este caso, `x="título eje X"`, `y="Título eje Y"`, `col="Título de la leyenda"`, y además, `title="Título del gráfico"`. Añade las labs e imprime el gráfico.

Gráfico 2.

1. El segundo gráfico tiene como objetico crear una representación de los porcentajes de recuerdo de voto que en la encuesta tiene cada partido, referido a las elecciones de 2015. El porcentaje que se busca es el de voto válido, excluyendo abstención y NC. Completa la tabla:

Aspecto	Descripción
Datos	
Aesthetics	
Geoms	

2. Lo primero es realizar la preparación de los datos. Para ellos se ha creado un nuevo conjunto de datos, `d2`, en el que solo están los casos que no son NR, NC, y NoVoto en la variable `voto15` del data frame `d`.

3. Haz el set – up de **`ggplot`**(), los datos son `d2`, y la aes solo necesitamos especificar `x=voto15`.

4. A continuación añade el geom **`geom_bar`**() e imprime el gráfico. Como ves el eje Y representa las frecuencias, pero en este caso se buscan los porcentajes. No existe una opción para añadir los porcentajes de forma automática, sino que hay que calcularlo de forma explícita. Si lo piensas, aunque en aesthetics solo has especificado el argumento `x=voto15`, el sistema calcula el elemento `y` en el *background*. En este caso el sistema calcula la frecuencia para cada categoría de `x`, pero podemos cambiar esto. Dentro de **`geom_bar`**(), como ya has visto, puedes especificar la aesthetics, así que allí podemos cambiar el eje Y. Para ello utilizaremos variables internas del sistema, que se escriben con dos puntos delante y dos detrás (e.g. "`..count..`"). Añade dentro de **`geom_bar`**() la siguiente información:

```
aes(y=..count../sum(..count..)*100)
```

Además de modificar la aesthetics, añade otros dos atributos a **`geom_bar`**(): `fill="gray", width=.75`. Recuerda que al ser atributos van fuera del paréntesis de aes. Imprime el gráfico.

5. Ahora modifica la escala del eje Y. En este caso usa **`scale_y_continuous`**(). Esta función tiene dos argumentos principales, `limits=c(min, max)`, en el que min y max deben ser sustituidos por el máximos y el mínimo que quieres para la escala. El segundo argumento es `breaks =c()`, en el que se establecen los puntos que quieres que aparezcan señalados en la escala. En este caso, establece una escala del 0 al 40, con breaks cada 10 puntos.

6. Por último, añade un título al gráfico, y en este caso, deja los títulos del eje X e Y en blanco, ya que no son necesarios. Utiliza la función **`labs`**(). Imprime el gráfico.

Gráfico 3.

1. En este gráfico vas a comparar la distribución de dos variables cuantitativas de forma independiente, confpub y confpriv. Completa la tabla:

Aspecto	Descripción
Datos	
Aesthetics	
Geoms	

2. En este caso los datos están preparados. Por lo tanto, pasa a establecer la base del gráfico usando **`ggplot`**(). Vas a realizar dos histogramas superpuestos, uno para `confpub` y otro para `confpriv`. Los histogramas solo requieren de un elemento de aesthetics para funcionar, `x`. Como vas a hacer dos histogramas separados, la aesthitics no se establece en la función **`ggplot`**(), sino dentro de cada

geom_histogram(). Por lo tanto, en este caso, **ggplot**() solo contendrá los datos. La sintaxis está lista, ejecutala.

3. En la siguiente capa añadirás dos geoms, una por cada histograma que vamos a producir, para ello se utiliza el **geom_hist**(). Cada histograma necesita su aesthetics, uno tendrá **x=confpub**, otro **x=confpriv**. El histograma, como el gráfico de barras, representa la cuenta de casos en el eje Y. Sin embargo, queremos realizar el histograma utilizando la densidad en vez de la frecuencia de casos. Para representar la densidad, dentro de aesthetics, establece **y=..density..** . Density es una variable del sistema que calcula la densidad. Dado que vamos a superponer dos histogramas, hay que diferenciarlos utilizando diferentes colores. Los colores también los estableceremos dentro de la aesthetics, para poder crear una leyenda, así añade el argumento **fill="color"**, sustituyendo "color" por "skyblue" para **confpub**, y "red" para **confpriv**. Fuera de la aesthetics, como atributos, establece **alpha=.2**, para que los histogramas sean parcialmente transparentes. También fuera de la aesthetics establece **bins=30**. Lo que hace el histograma es partir la variable cuantitativa en grupos ("bins"), en este caso 30, dado que nuestras variables tienen 30 valores, si bins es 30, habrá un valor por grupo. Este argumento puede necesitar ser cambiada en otro escenario. Ejecuta e imprime el gráfico.

4. Además de añadir las **geom_histogram**(), añade unas líneas de densidad. Para ello habrá que incluir dos funciones más en la siguiente capa, una para **confpub** y otra para **confpriv**: **geom_density**(). En la aesthetics dentro de **geom_density**(), **x** corresponderá a **confpub** y **confpriv** respectivamente. También dentro de la aestehtics, el color será "skyblue" en **confpub**, y "red" en **confpriv**. Ejecuta e imprime el gráfico.

5. Ahora establece las etiquetas de la leyenda utilizando las escalas. En este gráfico has utilizado dos elementos en la aestehtics: fill en los histogramas, y color en los gráficos de densidad. Por ello utilizaremos dos funciones:

```
g3 <- g3 + scale_fill_discrete(labels=c("Privadas",
"Públicas"), name="" ) +
  scale_color_discrete(labels=c("Privadas",
"Públicas"), name="")
```

6. Por último establece las etiquetas utilizando la función **labs**(). Este gráfico solo necesita título, puedes dejar en blanco X e Y. Imprime el gráfico. Prueba a rehacer el gráfico reduciendo el bins de **geom_histogram**().

Gráfico 4.

1. El cuarto gráfico es una representación de la variación del nivel de confianza en las instituciones pública para los diferentes niveles de recuerdo de voto en 2016:

Aspecto	Descripción
Datos	
Aesthetics	
Geoms	

2. La preparación de la variable **voto16** para transformarla en **voto16r** (agrupa a podemos y las confluencias en una categoría), ya está realizada. Ejecutalo.

3. Establece la base del gráfico usando **ggplot**(). En la aesthetics habrá dos elementos, uno el eje **x**, que en este caso será **voto16r**, y el elemento del eje **y** será **confpub**. Ejecuta e imprime.

4. Añade el geoms del gráfico: **geom_boxplot**(), establece el atributo **fil="skyblue"**. Ejecuta e imrpime.

5. Retira los títulos de los ejes X e Y. Crea el título del gráfico usando **labs**(). Ejecuta e imprime.

6. Como está el gráfico hay un diagrama de cajas que muestra la distribución de la variable **confpub** para cada categoría de **voto16r**. Intenta ahora incluir una caja adicional que sea la media española de **confpub**, es decir, un resumen de todas las anteriores. Cambia el color para diferencias esa caja del resto.

Gráfico 5.

1. En este ejercicio vas a utilizar la función **stat_summary**() para crear un gráfico de medias con sus intervalos de confianza.

2. Los dos primeros pasos del ejercicio están listos para ejecutar. En el tercero tienes que añadir la capa **stat_summary**(). Esta función es muy útil porque nos permite graficar cualquier resultado que sea asumible por ggplot2, que consista en un resumen de la variable y para los valores de x. En este caso la función que queremos graficar es **mean_cl_normal**(). Esta función ofrece la media y los intervalos de confianza de la misma. En este caso tu objetivo es representar la media de **confpub** para cada grupo de **urnas16r**, con sus barras de error. Para ello habrá que añadir:

```
stat_sumary(fun.data=mean_cl_normal)
```

   Si te fijas fun.data sirve para llamar cualquier función, en nuestro caso **mean_cl_normal**. Si ejecutas en la consola la función **mean_cl_normal**(**d$confpub**), te devolverá el resultado con la media, el ymin y el ymax, que marcan el principio y final del intervalo de confianza. Ahora ejecuta la sintaxis, e imprime el resultado.

3. En el siguiente paso del ejercicio vas a incorporar los límites del eje Y. Para ello añade la función **coord_cartesian**(), añadiendo el argumento, **ylim = c(min, max)**, En este caso queremos ver una escala del 0 al 15.

4. Por último, ejecuta las líneas que añaden las etiquetas, e imprime el resultado final del gráfico.