Gráficos avanzados con R

CNE/ISCIII

Estructura del curso

1. Gráficos básicos
2. Gráficos avanzados
3. Mapas
4. Informes automatizados

¿Porqué usar gráficos?

"Una imagen vale más que mil palabras" (Napoleón)

Motivos:

- Potente herramienta de síntesis de la información estadística
- Agiliza la exploración, modelización y comunicación en el análisis de datos

Objetivos del curso:

- Manejar el paquete `ggplot2` de R que implementa una "gramática" de diseño de los gráficos
- Controlar el proceso de elaboración de gráficos, que va desde la preparación de los datos hasta la publicación de los resultados del análisis

Gráficos básicos

Una función si no hay tiempo para pensar: `qplot`

Instalación del paquete `ggplot2` ("gg" para "Grammar of Graphics").

```
#install.packages("ggplot2")
library(ggplot2) #carga la librería ggplot2
```

Empezaremos con la función básica `qplot` ("quick plot") de este paquete:

```
qplot(x, y=NULL, data, geom="auto")
```

- `x` : valores en el eje de abscisas.
- `y` : valores en el eje de ordenadas (opcional).
- `data` : data.frame de donde salen los datos (opcional).
- `geom` : elementos gráficos o geometrías ("point", "line", "bar", ..). Por defecto, "point" si `y` viene especificado, e "histogram" si sólo se especifica `x`.
- … y otros argumentos relacionados con los ejes del gráfico (`xlab`, `ylab`: etiquetas de los ejes; `xlim`, `ylim`: limites en los ejes de `x` e `y`; `log`: eje en escala log, "x", "y" o bien "xy").

Distribución de una variable continua

Un ejemplo

Para ilustrar la descripción gráfica de la distribución de una variable numérica, se utiliza la base de datos `msleep` que contiene información sobre el tiempo de sueño (en horas) de mamíferos:

```
str(msleep) # ?msleep para más detalles
```

```
## tibble [83 × 11] (S3: tbl_df/tbl/data.frame)
##  $ name        : chr [1:83] "Cheetah" "Owl monkey" "Mountain beaver" "Greater short-tailed shrew" ...
##  $ genus       : chr [1:83] "Acinonyx" "Aotus" "Aplodontia" "Blarina" ...
##  $ vore        : chr [1:83] "carni" "omni" "herbi" "omni" ...
##  $ order       : chr [1:83] "Carnivora" "Primates" "Rodentia" "Soricomorpha" ...
##  $ conservation: chr [1:83] "lc" NA "nt" "lc" ...
##  $ sleep_total : num [1:83] 12.1 17 14.4 14.9 4 14.4 8.7 7 10.1 3 ...
##  $ sleep_rem   : num [1:83] NA 1.8 2.4 2.3 0.7 2.2 1.4 NA 2.9 NA ...
##  $ sleep_cycle : num [1:83] NA NA NA 0.133 0.667 ...
##  $ awake       : num [1:83] 11.9 7 9.6 9.1 20 9.6 15.3 17 13.9 21 ...
##  $ brainwt     : num [1:83] NA 0.0155 NA 0.00029 0.423 NA NA NA 0.07 0.0982 ...
##  $ bodywt      : num [1:83] 50 0.48 1.35 0.019 600 ...
```

Histograma

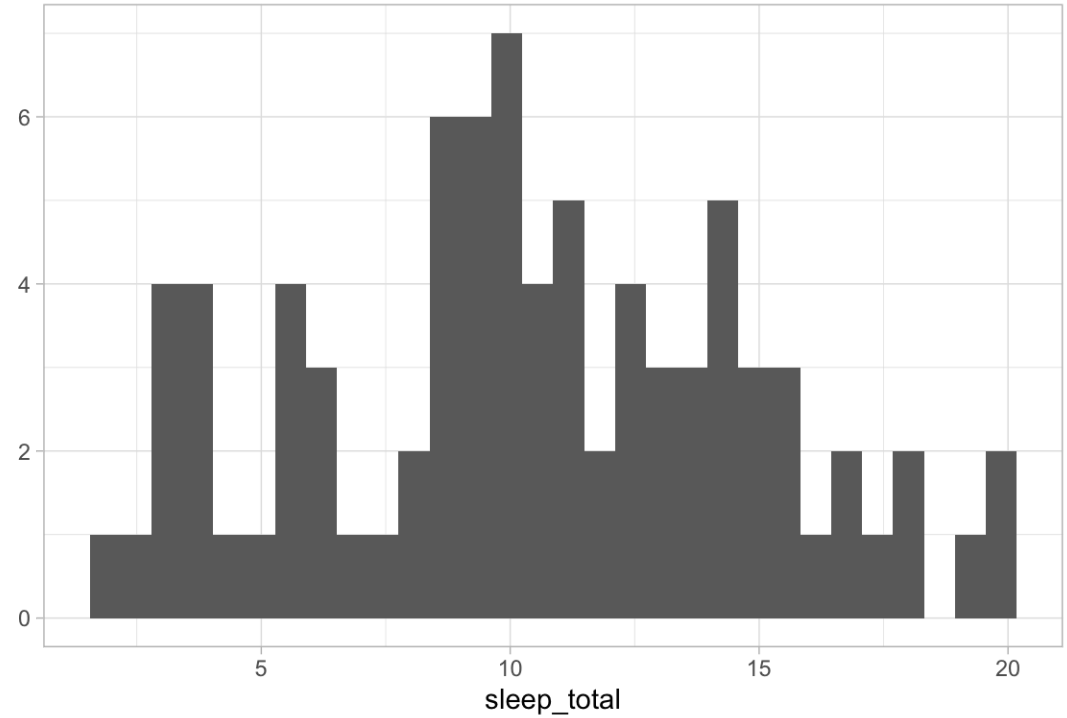
A mano

```
#summary(msleep$sleep_total) #tiempo total de sueño (en horas)
stem(msleep$sleep_total)
```

```
##
##   The decimal point is at the |
##
##    0 | 9
##    2 | 79013589
##    4 | 0423346
##    6 | 23307
##    8 | 03446779114456788
##   10 | 01113346900135
##   12 | 15555880578
##   14 | 234456996889
##   16 | 604
##   18 | 01479
```

… y con `qplot`

```
qplot(sleep_total, data=msleep) #histograma
```

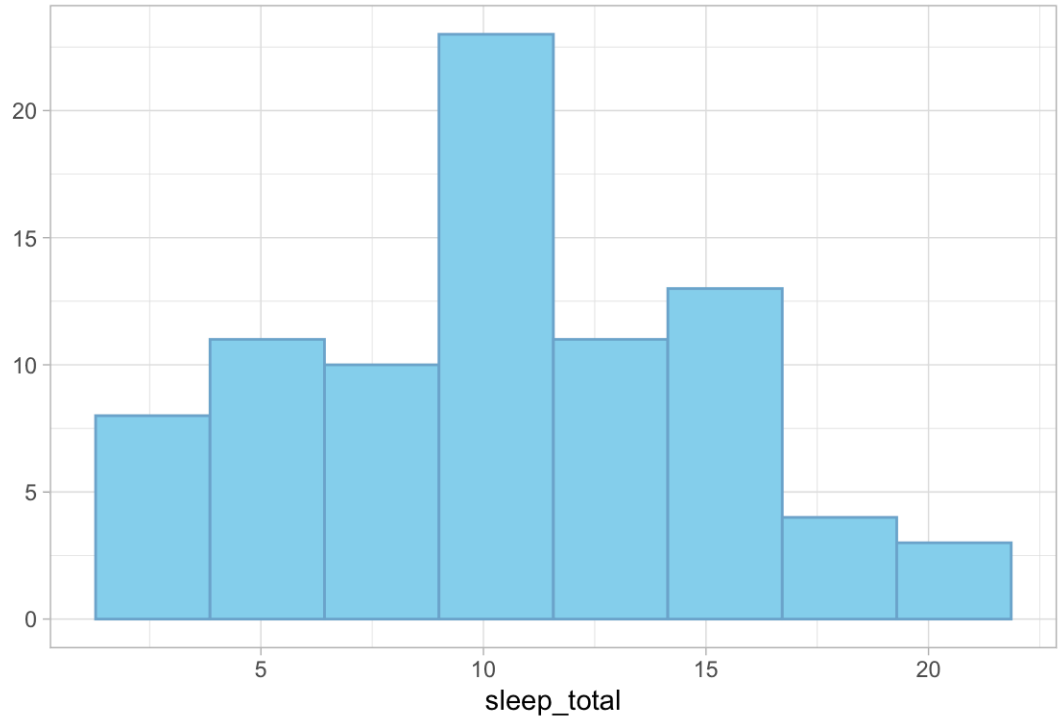


La altura de cada barra en el histograma es proporcional a la frecuencia de datos que caen en el intervalo correspondiente. Por defecto, el número de barras es igual al valor arbitrario `bins=30`.

Número de intervalos

Otra alternativa consiste en elegir un número k de barras en función del tamaña muestral n, como por ejemplo, el criterio de Sturges ($k = 1 + \log_2(n)$) o el criterio de Rule ($k = 2n^{1/3}$).

```
qplot(sleep_total, data=msleep, bins=8, color=I("skyblue3"), fill=I("skyblue")) #con criterio de Rule
```



Una dimensión más

El argumento `fill` controla el color de relleno de las barras y el argumento `color` el color del borde. Para especificar un color concreto se utiliza la función `I()`. Si el color varía con otra variable `z`, se especifica esta dependencia escribiendo `fill=z`.

```
qplot(sleep_total, data=msleep, bins=12, fill=vore) #distribución del tiempo de sueño según dieta del mamífero.
```

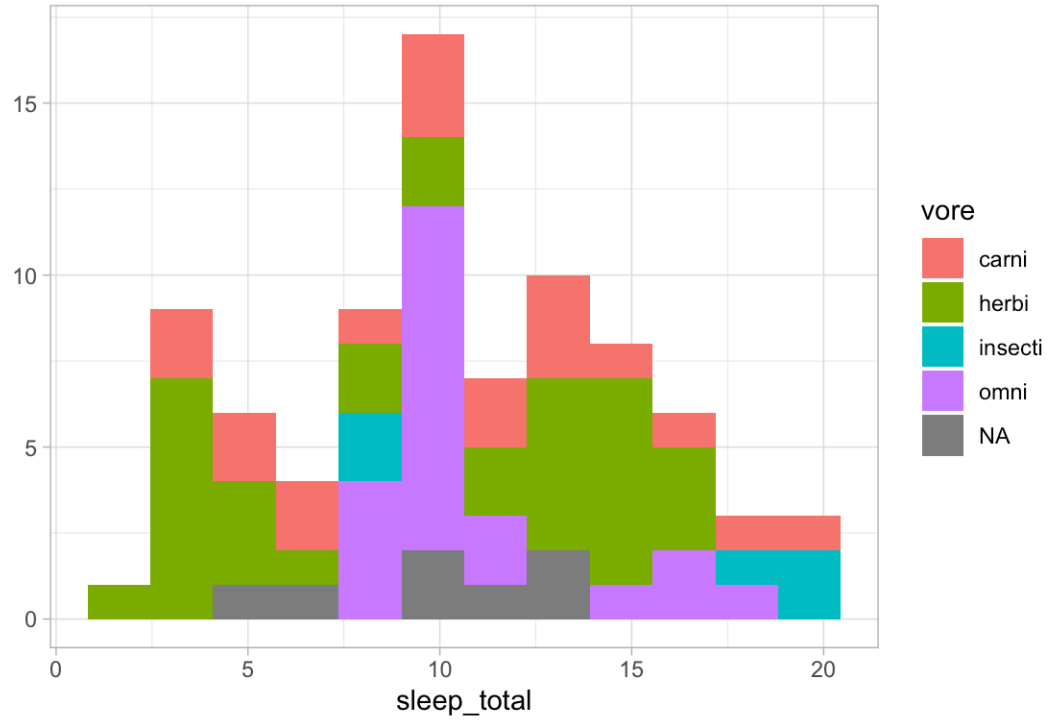
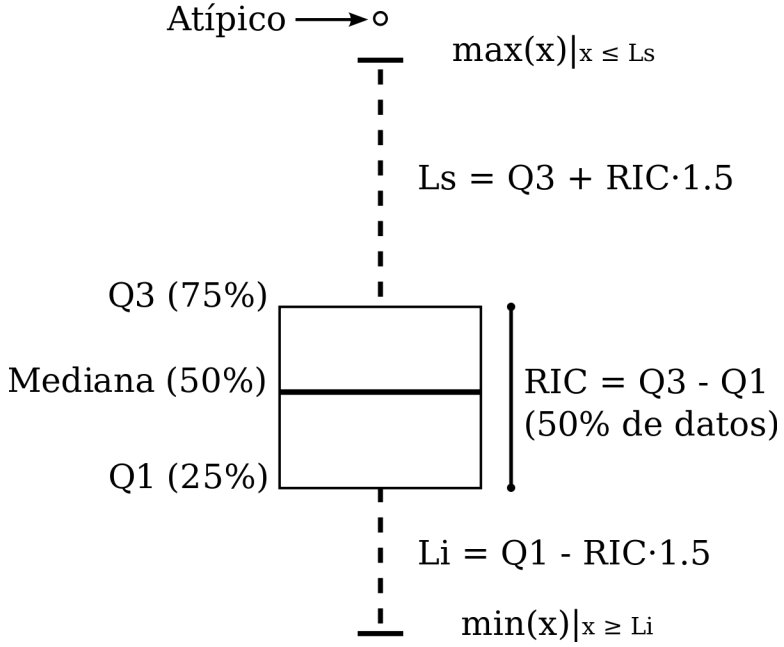


Diagrama de caja (boxplot)

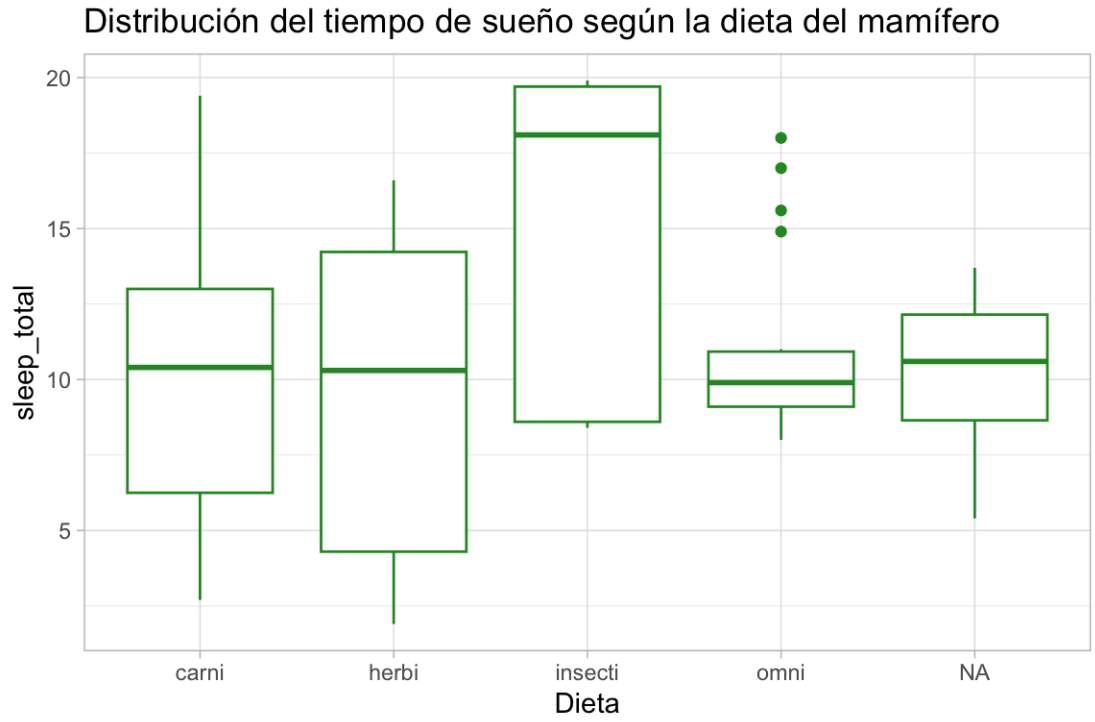
Describe la distribución de una variable numérica mediante una caja y unos segmentos que acotan las regiones donde tiene el grueso de sus valores.

- Menos fina que el histograma pero más robusta.
- Adecuada para representar dependencia con otra variable (categórica).



Una dimensión más

```
qplot(vore,sleep_total,data=msleep,geom="boxplot",xlab="Dieta",color=I("forestgreen"))
```

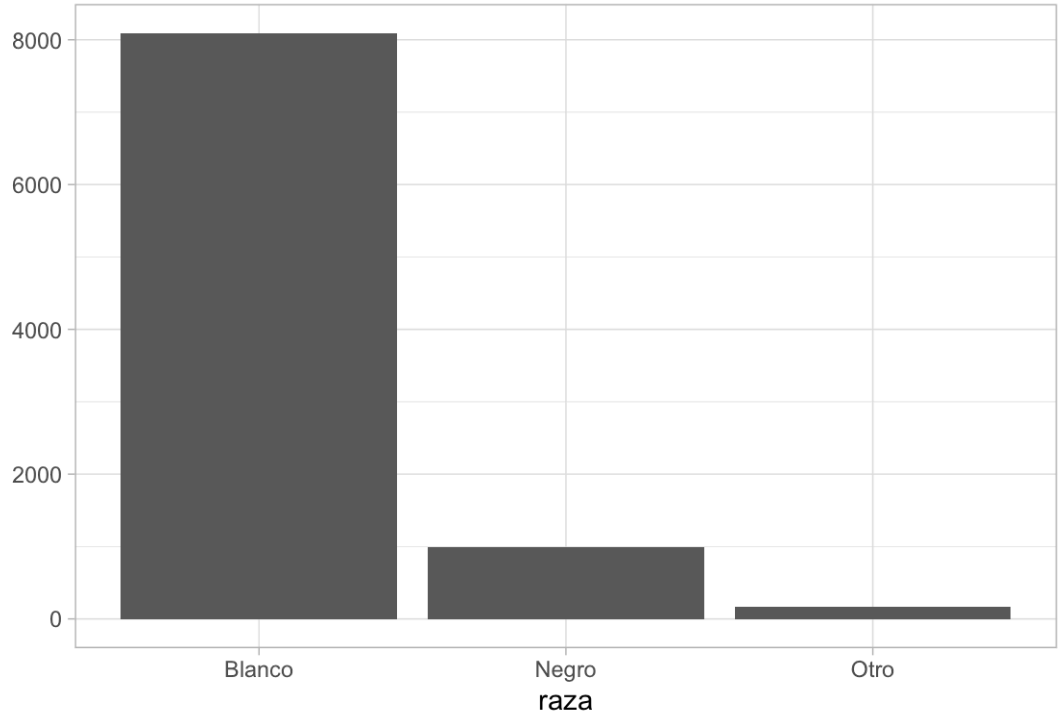


Ejercicio: Cargar la base de datos de la encuesta nacional americana `nhs` y representar la distribución del índice de masa corporal (imc) según el sexo y la raza.

Diagrama de barras

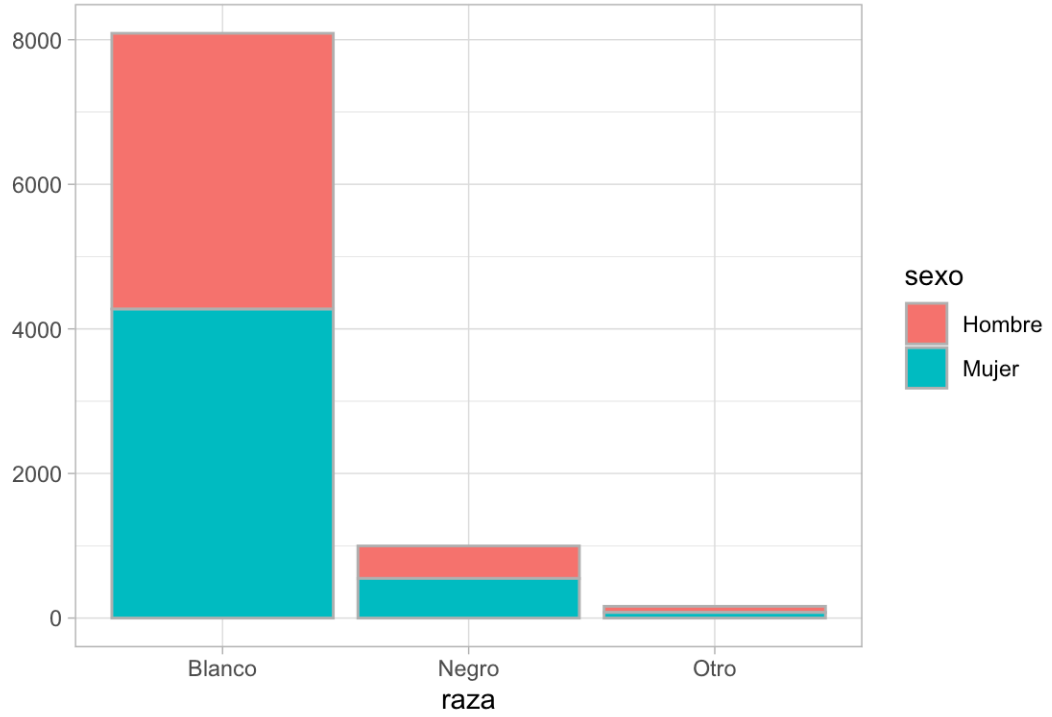
Los diagramas de barras permiten representar la distribución de una variable categórica. En esta representación, cada categoría viene representada por una barra cuya altura es proporcional a su frecuencia en la muestra.

```
qplot(raza,data=nhs) #Distribución de las razas en la muestra de la encuesta americana
```



Una dimensión más

```
qplot(raza,data=nhs,fill=sexo,color=I("grey70")) #Distribución del sexo segun la raza
```

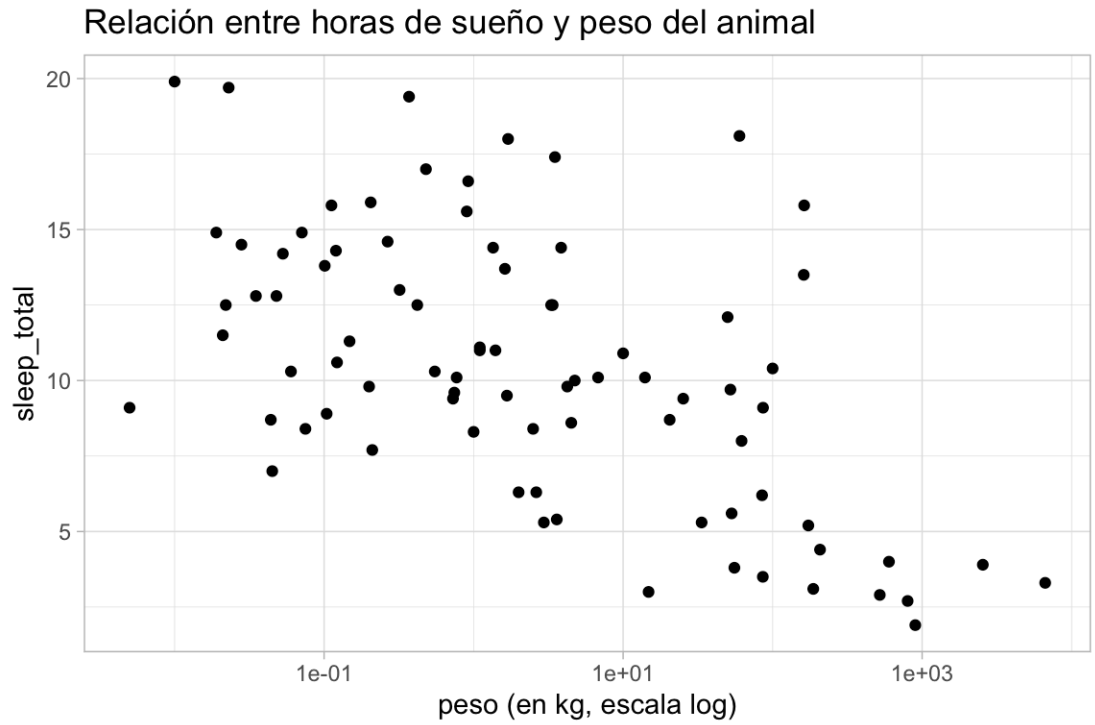


Con el argumento `fill` se puede ver como varia la distribución de una variable respecto a otra (aquí el sexo según la raza). El gráfico obtenido resulta poco claro y veremos más adelante como mejorarlo.

Relación entre dos variables cuantitativas

Diagrama de dispersión

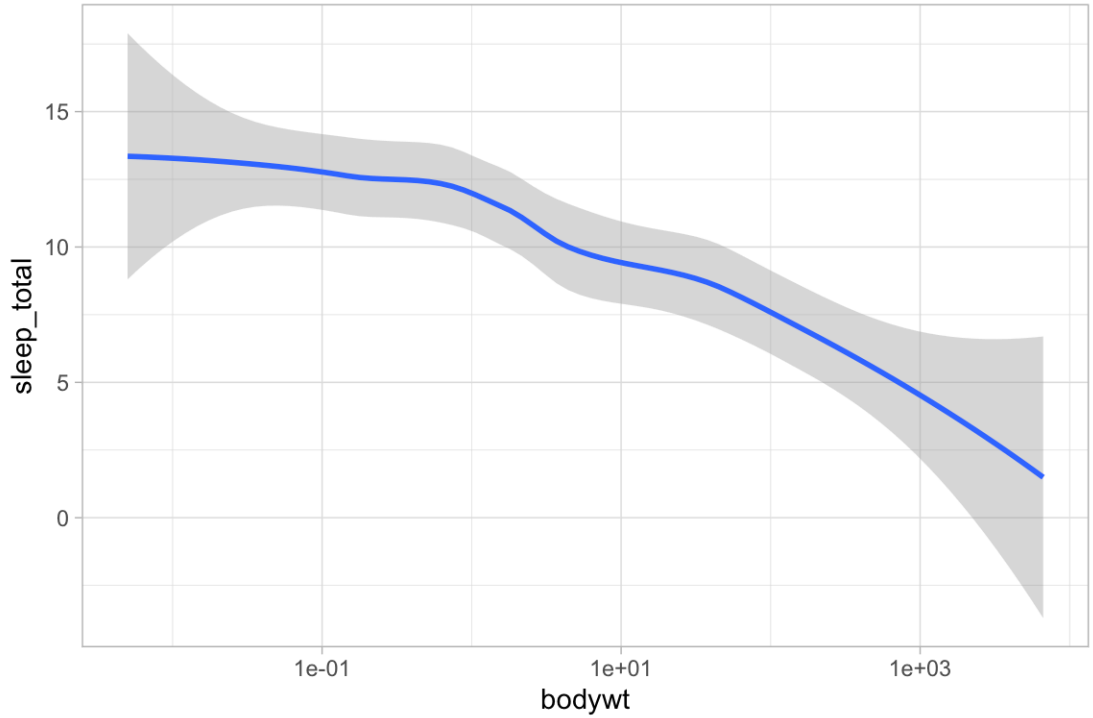
```
qplot(bodywt,sleep_total,data=msleep,xlab="peso (en kg, escala log)",log="x")
```



Ajuste

Para apreciar mejor la tendencia en la nube de puntos, se puede ajustar una curva ("smooth"):

```
qplot(bodywt, sleep_total, data=msleep, log="x", geom="smooth")
# utilizar el argumento method="lm" para ajustar una recta
```

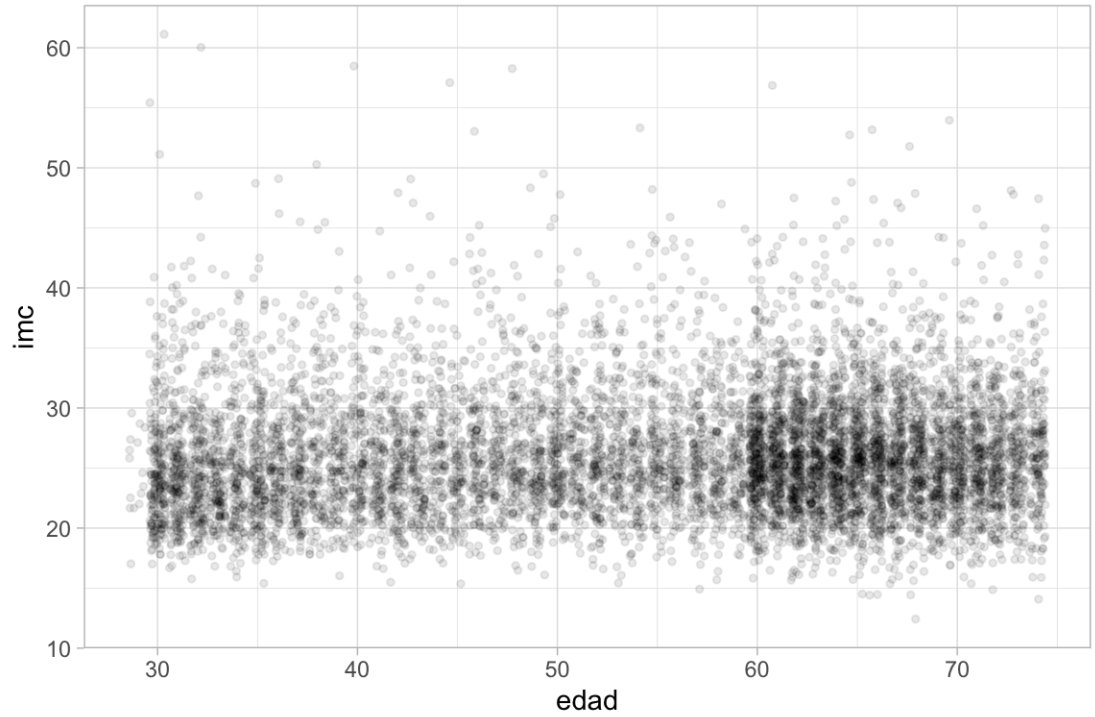


Ejercicio: Representar la tendencia de la esperanza de vida por continentes utilizando la base de datos `gapminder`.

Problema de solapamiento

El solapamiento de puntos puede ser minimizado insertando algo de ruido en los datos (`geom="jitter"`), reduciendo el tamaño de los puntos (`size`) o recurriendo a la transparencia (`alpha`):

```
qplot(edad, imc, data=nhs, alpha=I(.1), size=I(1), geom="jitter")
```



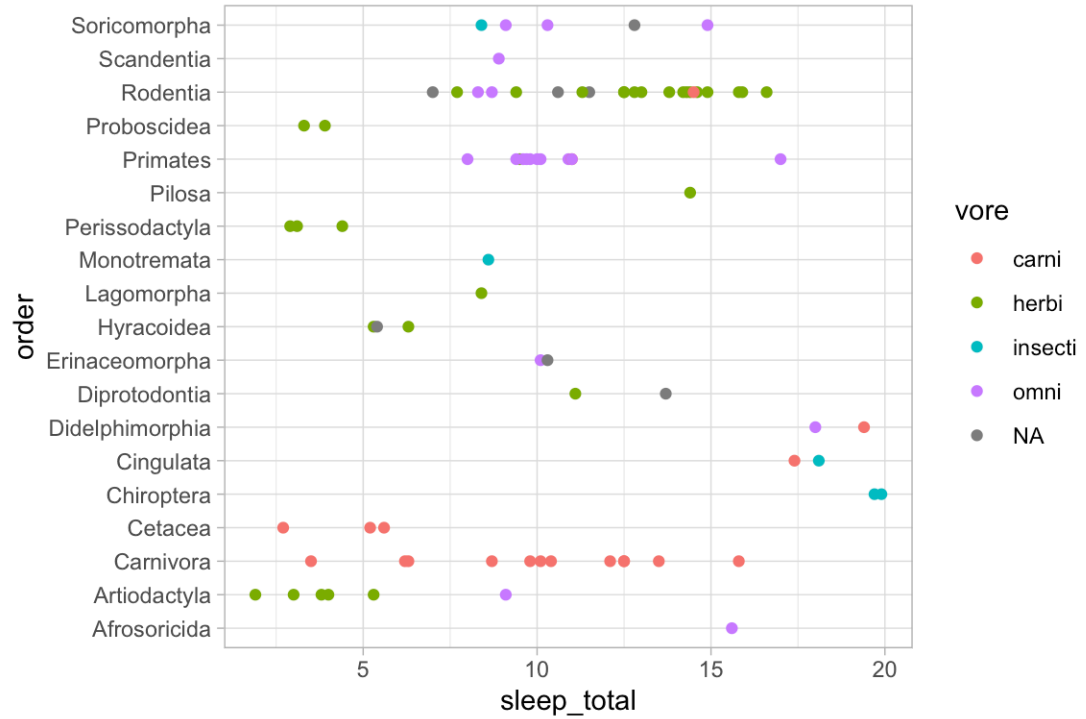
Ejercicio: Describir la relación entre edad y presión arterial sistólica, y como esta relación cambia con el sexo.

Dotchart

Relación con una variable categorica

Si una de las variables es categórica y tiene muchas categorías, el gráfico de dispersión puede ser también apropiado:

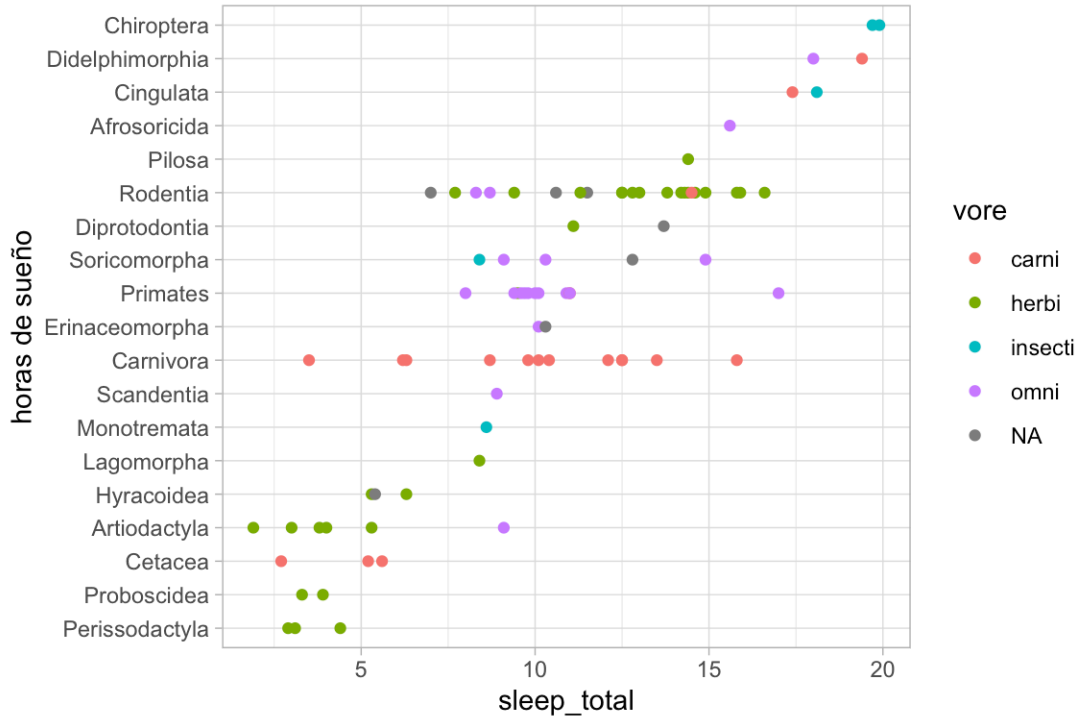
```
qplot(sleep_total,order,data=msleep,col=vore)
```



Poniendo orden

Para mayor claridad, es recomendable ordenar la variable categórica de acuerdo a la variable numérica:

```
qplot(sleep_total,reorder(order,sleep_total),data=msleep,col=vore,ylab="horas de sueño")
```

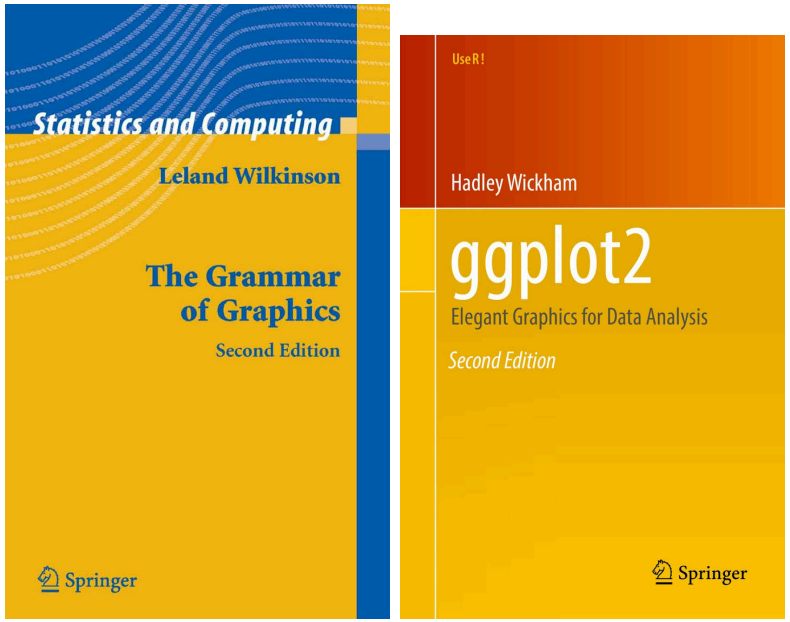


Ejercicio: Describir con un gráfico similar al anterior, los datos de la base de datos `islands` sobre superficies de islas (es aconsejable escoger una escala `log`).

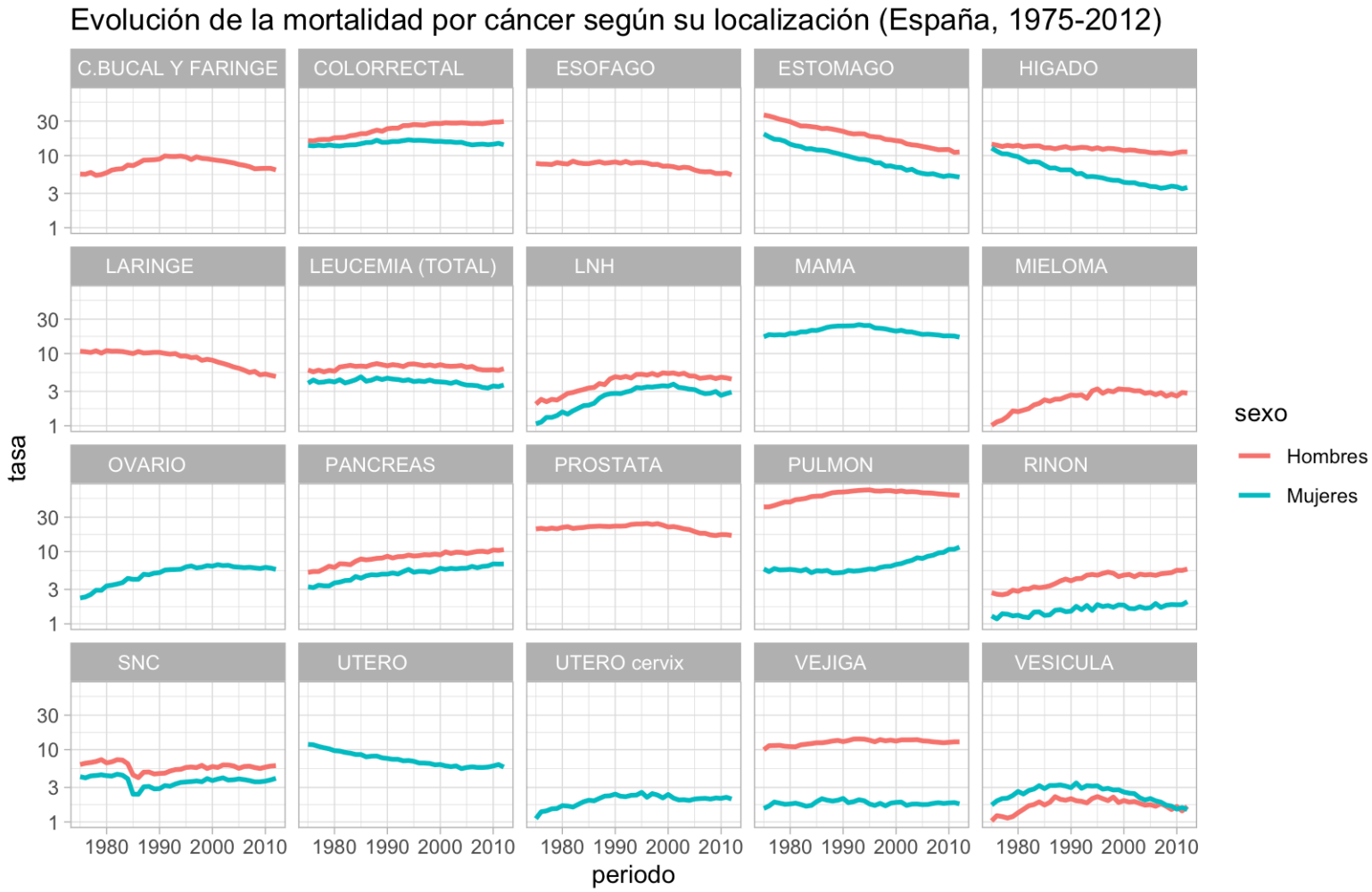
Gráficos avanzados

Gramática de los graficos

Introducción a *The Grammar of Graphics* de Leland Wilkinson (2005) tal y como viene implementada en el paquete `ggplot2` de Hadley Wickham (2009).



Un ejemplo


Evolución de la mortalidad por cáncer según su localización (España, 1975-2012)
C.BUCAL Y FARINGE
COLORRECTAL
ESOFAGO
ESTOMAGO
HIGADO
LARINGE
LEUCEMIA (TOTAL)
LNH
MAMA
MIELOMA
OVARIO
PANCREAS
PROSTATA
PULMON
RINON
SNC
UTERO
UTERO cervix
VEJIGA
VESICULA
30
10
3
1
1980 1990 2000 2010
tasa
periodo
sexo
Hombres
Mujeres

Un ejemplo

El código

El código de `ggplot2` que generó el gráfico anterior fue:

```
ggplot(cancer) + #carga la base de datos cancer
  aes(x = periodo, y = tasa, col = sexo) +
  geom_line(size=1) +
  scale_y_log10() +
  facet_wrap( ~ tumor)
```

Esta expresión combina varios elementos:

- **Datos**: siempre un "data.frame"
- **Estéticas**: columnas del data.frame representables gráficamente (coordenadas x e y, el color, …)
- **Geometrías** (o capas): puntos, rectas, áreas, histogramas, …, que pueden superponerse.
- **Facetas**: parten un gráfico en sublienzos preservando el diseño original

Elementos de un gráfico

Datos

La base del gráficos son los datos:

- El primer argumento de la función `ggplot` es un `data.frame`.
- **Formato alargado**: una columna para cada dimensión y una fila para cada observación.

```
load("data/cancer.RData") #carga los datos
cancer
```

```
##          sexo periodo             tumor     tasa
##    1: Hombres    1975 C.BUCAL Y FARINGE 5.541879
##    2: Hombres    1976 C.BUCAL Y FARINGE 5.512168
##    3: Hombres    1977 C.BUCAL Y FARINGE 5.827874
##    4: Hombres    1978 C.BUCAL Y FARINGE 5.323451
##    5: Hombres    1979 C.BUCAL Y FARINGE 5.461059
##   ---
## 1174: Mujeres    2008 LEUCEMIA (TOTAL)  3.402265
## 1175: Mujeres    2009 LEUCEMIA (TOTAL)  3.337342
## 1176: Mujeres    2010 LEUCEMIA (TOTAL)  3.558769
## 1177: Mujeres    2011 LEUCEMIA (TOTAL)  3.494782
## 1178: Mujeres    2012 LEUCEMIA (TOTAL)  3.675811
```

Formato alargado

La siguiente base de datos no viene en un formato alargado:

```
VADeaths # mortalidad (por 1000 p.a) según grupos socio-demográficos y de edad (Virgina, 1940):
```

```
##       Rural Male Rural Female Urban Male Urban Female
## 50-54       11.7          8.7       15.4          8.4
## 55-59       18.1         11.7       24.3         13.6
## 60-64       26.9         20.3       37.0         19.3
## 65-69       41.0         30.9       54.6         35.1
## 70-74       66.0         54.3       71.1         50.0
```

Conversión al formato alargado (función `melt`):

```
temp=data.table(VADeaths,keep.rownames=TRUE) #require(data.table)
mortalidad=melt(temp,id.vars="rn") #formato alargado
names(mortalidad) <- c("edad","grupo","tasa")
str(mortalidad)
```

```
## Classes 'data.table' and 'data.frame':   20 obs. of  3 variables:
##  $ edad : chr  "50-54" "55-59" "60-64" "65-69" ...
##  $ grupo: Factor w/ 4 levels "Rural Male","Rural Female",..: 1 1 1 1 1 2 2 2 2 2 ...
##  $ tasa : num  11.7 18.1 26.9 41 66 8.7 11.7 20.3 30.9 54.3 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

Estéticas (**aes**)

El siguiente código crea un "protográfico" `p` que contiene los datos que vamos a utilizar:

```
p <- ggplot(mortalidad)
```

Pero, este código es insuficiente para dibujar un gráfico ya que no hemos indicado que dimensión de la base de datos se va a representar.

Para ello, se añade al objeto `p` información sobre las "estéticas" (las coordenadas, el color, la forma o el tamaño de un punto, …) y su relación con las variables de la base de datos:

```
p <- p + aes(x = edad, y = tasa, colour = grupo)

p$mapping # relación (o mapeo) entre estéticas y columnas de la base de datos
```

```
## Aesthetic mapping:
## * `x`      -> `edad`
## * `y`      -> `tasa`
## * `colour` -> `grupo`
```

¿Cuántas estéticas existen?

Alrededor de una docena, aunque se utilizan, generalmente, menos:

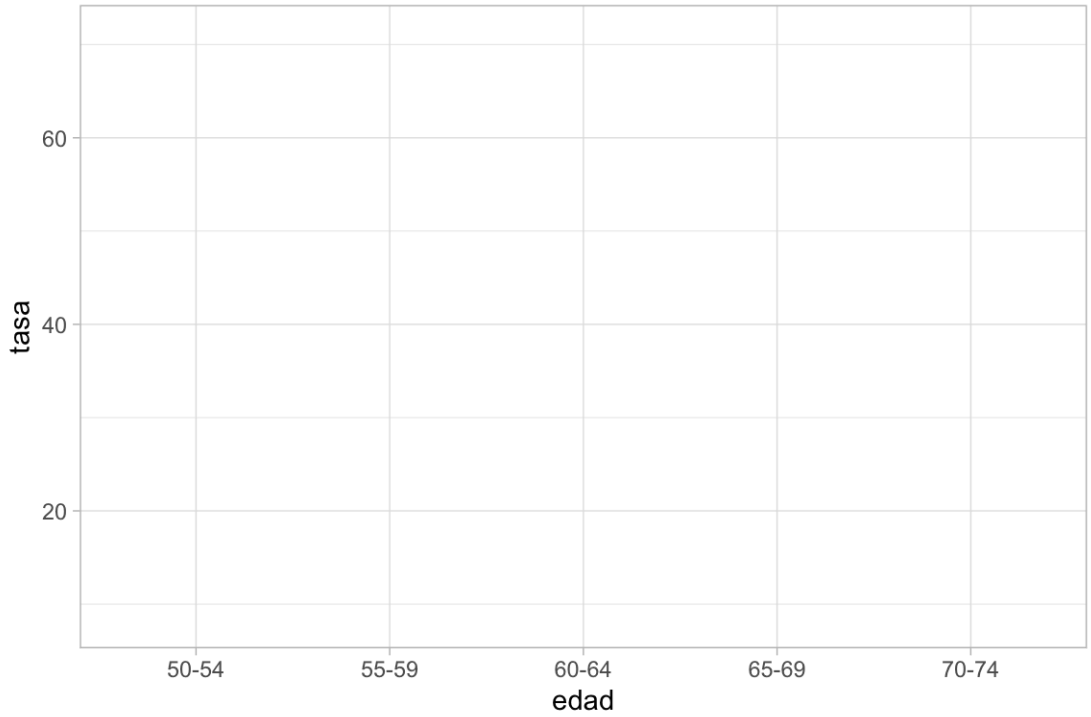
- `x` e `y`, coordenadas horizontal y vertical.
- `colour`, color de lineas y bordes.
- `size`, para el tamaño.
- `shape`, que indica la forma de los puntos (cuadrados, triángulos, etc.) de los puntos o del trazo (continuo, punteado) de las líneas.
- `alpha` para la transparencia: los valores más altos tendrían formas opacas y los más bajos, casi transparentes. También muy útil para el solapamiento de puntos.
- `fill`, para el color de relleno de las formas sólidas (barras, etc.).

No todas las *estéticas* tienen la misma potencia en un gráfico. El ojo humano percibe fácilmente colores y longitudes, pero tiene problemas para comparar áreas. Se recomienda usar las estéticas más potentes para representar las variables más importantes.

Un lienzo

El objeto p resultante aún no permite representar los datos (le falta capas):

```
p
```

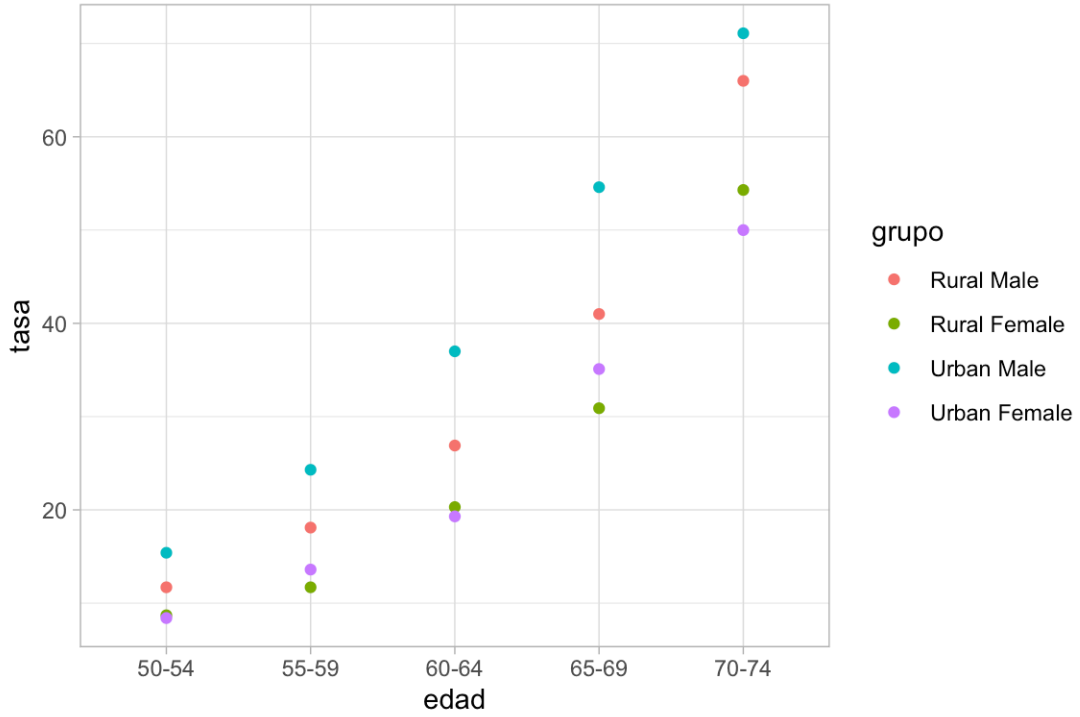


… no obstante, ya se puede apreciar los ejes.

Capas (**geoms**)

Las capas (o `geoms` para `ggplot2`) son los verbos del lenguaje de los gráficos. Indican como representar los datos mediante las estéticas en un lienzo. Una vez añadida una capa al gráfico, este puede pintarse:

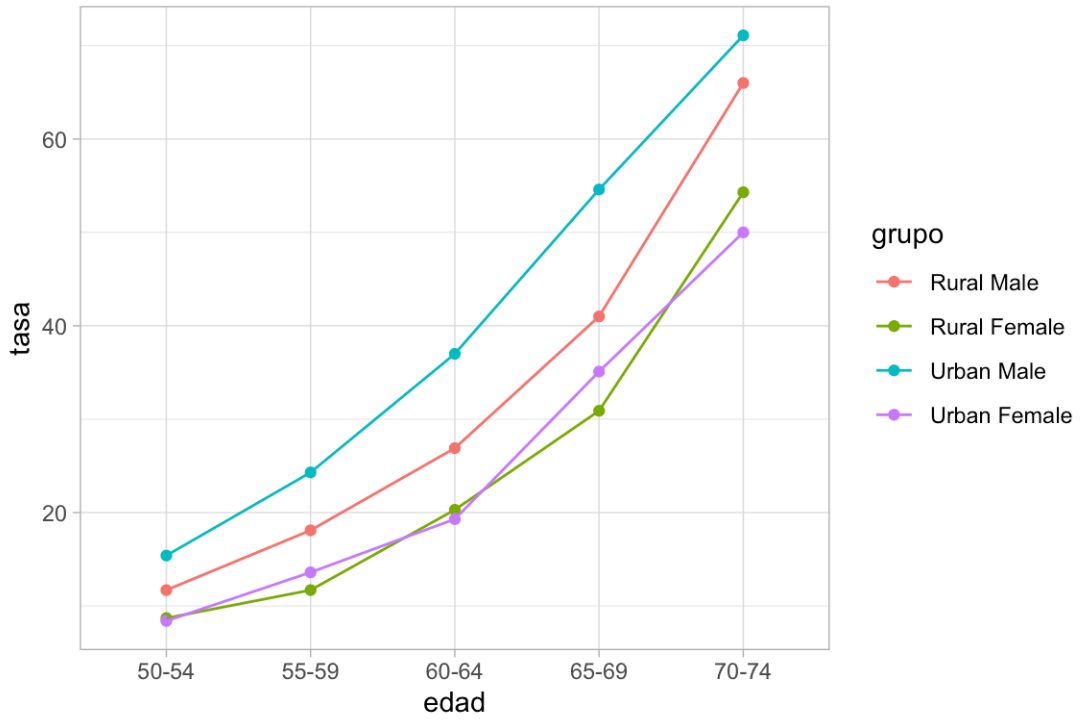
```
p <- p + geom_point()
p #ggplot(mortalidad, aes(x = edad, y = tasa, colour = grupo)) + geom_point()
```



Varias capas

Una característica de las capas, y de ahí su nombre, es que pueden superponerse:

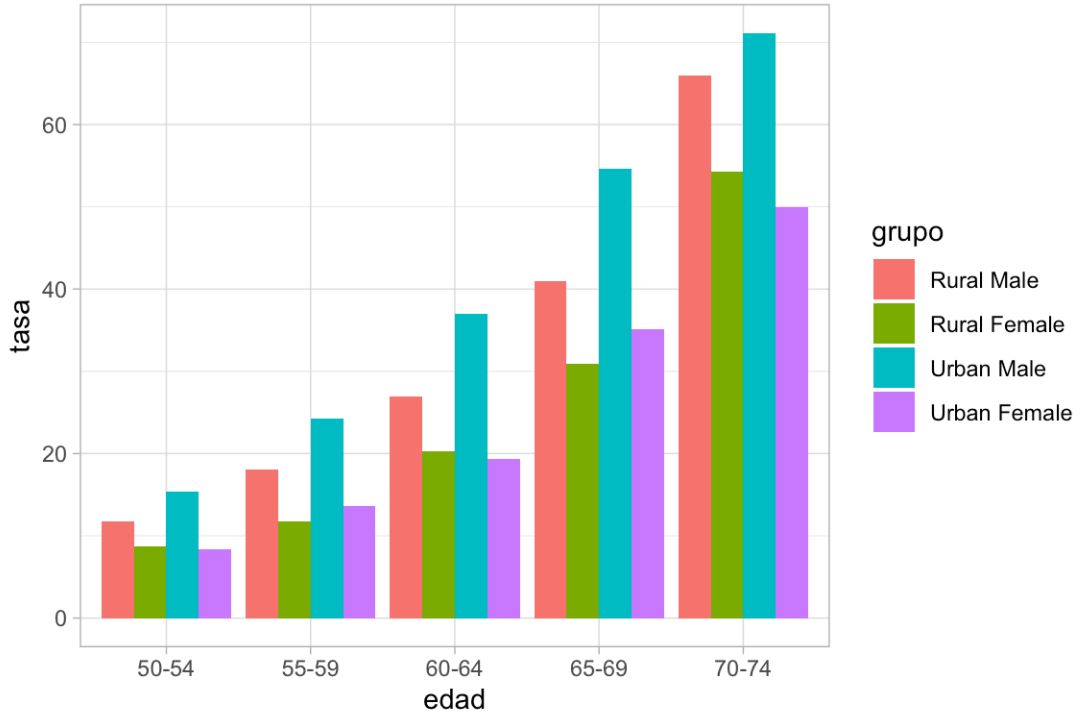
```
# Se requiere la estética `group` para conectar los puntos de una linea
# cuando la variable en abscisa es un factor.
ggplot(mortalidad, aes(x = edad, y = tasa, colour = grupo, group= grupo)) +
  geom_point() +
  geom_line()
```



Más capas

Abajo una representación mediante un diagrama de barra de los datos anteriores:

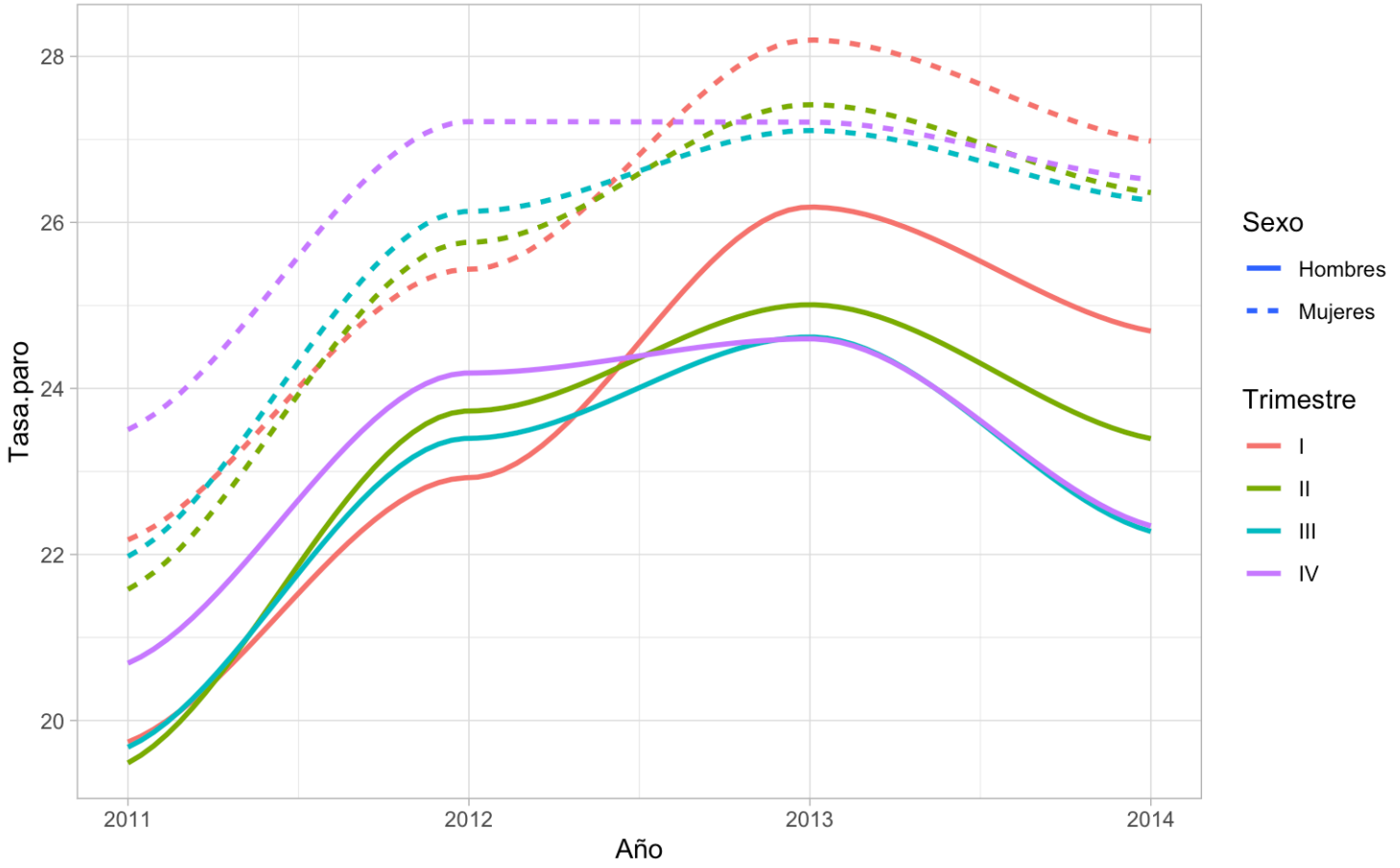
```
ggplot(mortalidad, aes(x = edad, y = tasa, fill = grupo)) +
  geom_bar(stat="identity",position="dodge")
```



Existen muchos tipos de capas. Los más usuales son `geom_point`, `geom_line`, `geom_histogram`, `geom_bar` y `geom_boxplot` (ver https://ggplot2.tidyverse.org/reference/) para una lista actualizada.

Ejercicio

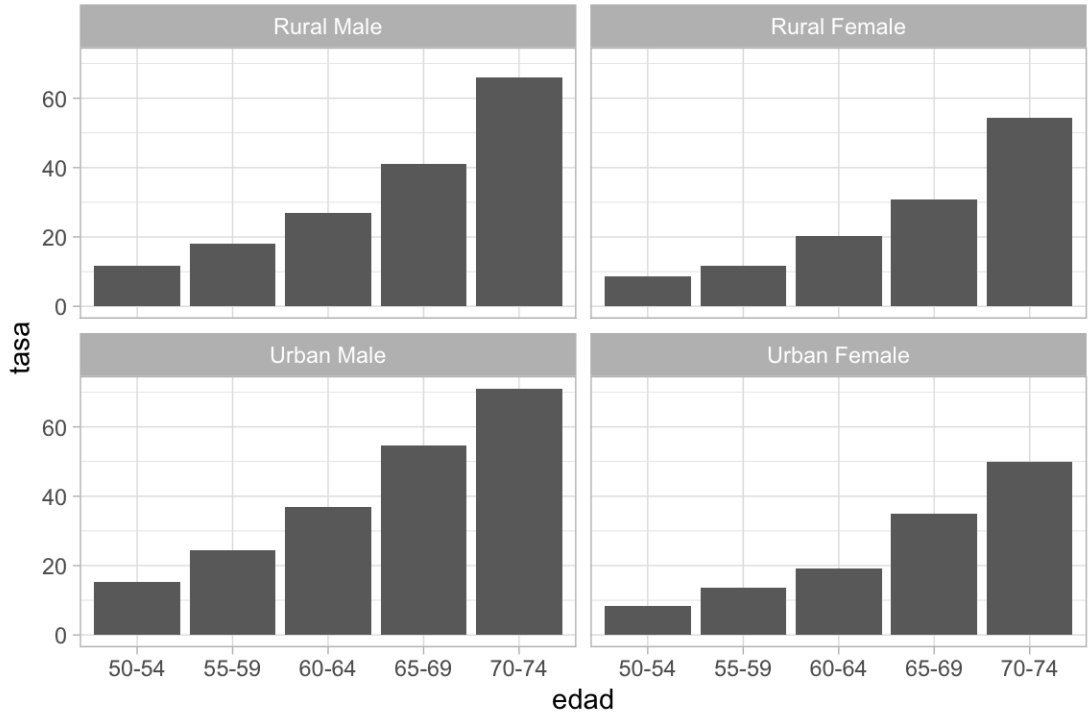
Elaborar el siguientes gráfico sobre la evolución del paro en España. Utilizar la capa `geom_smooth` para suavizar la tendencia y la estética `linetype` para distintos tipos de curvas.



Facetas

Las facetas permiten subdividir un gráfico. Suele ser un recurso muy eficiente para añadir otra dimensión al gráfico:

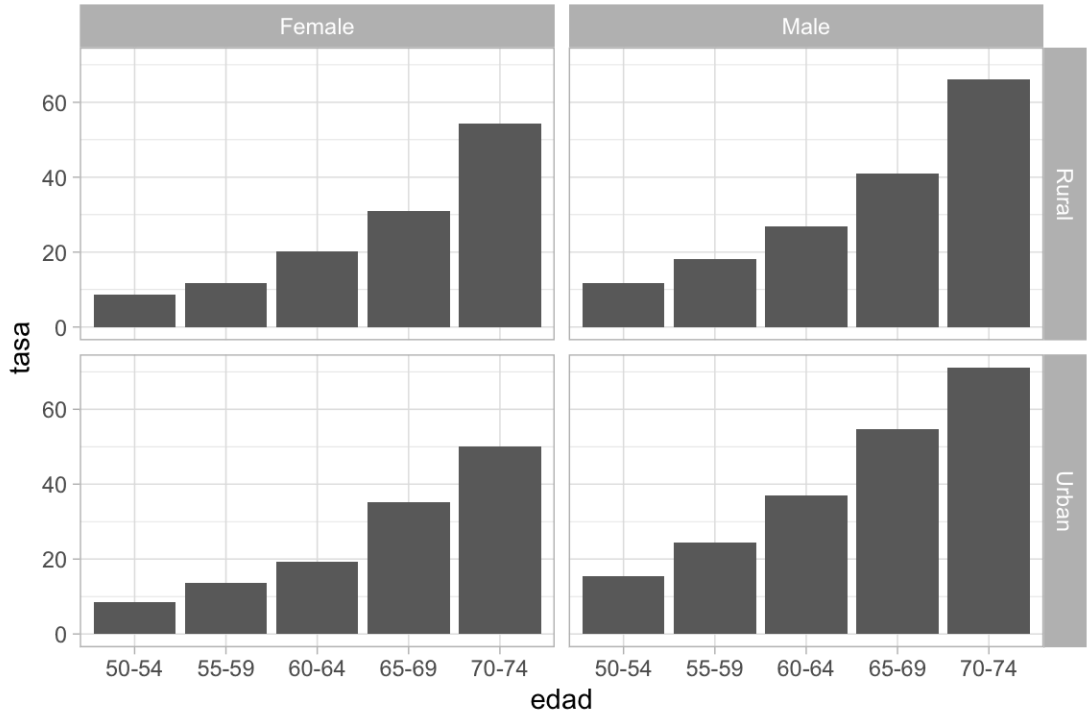
```
ggplot(mortalidad, aes(x = edad, y = tasa)) +
  geom_bar(stat="identity") +
  facet_wrap(~grupo)
```



facetas cruzadas (`facet_grid`)

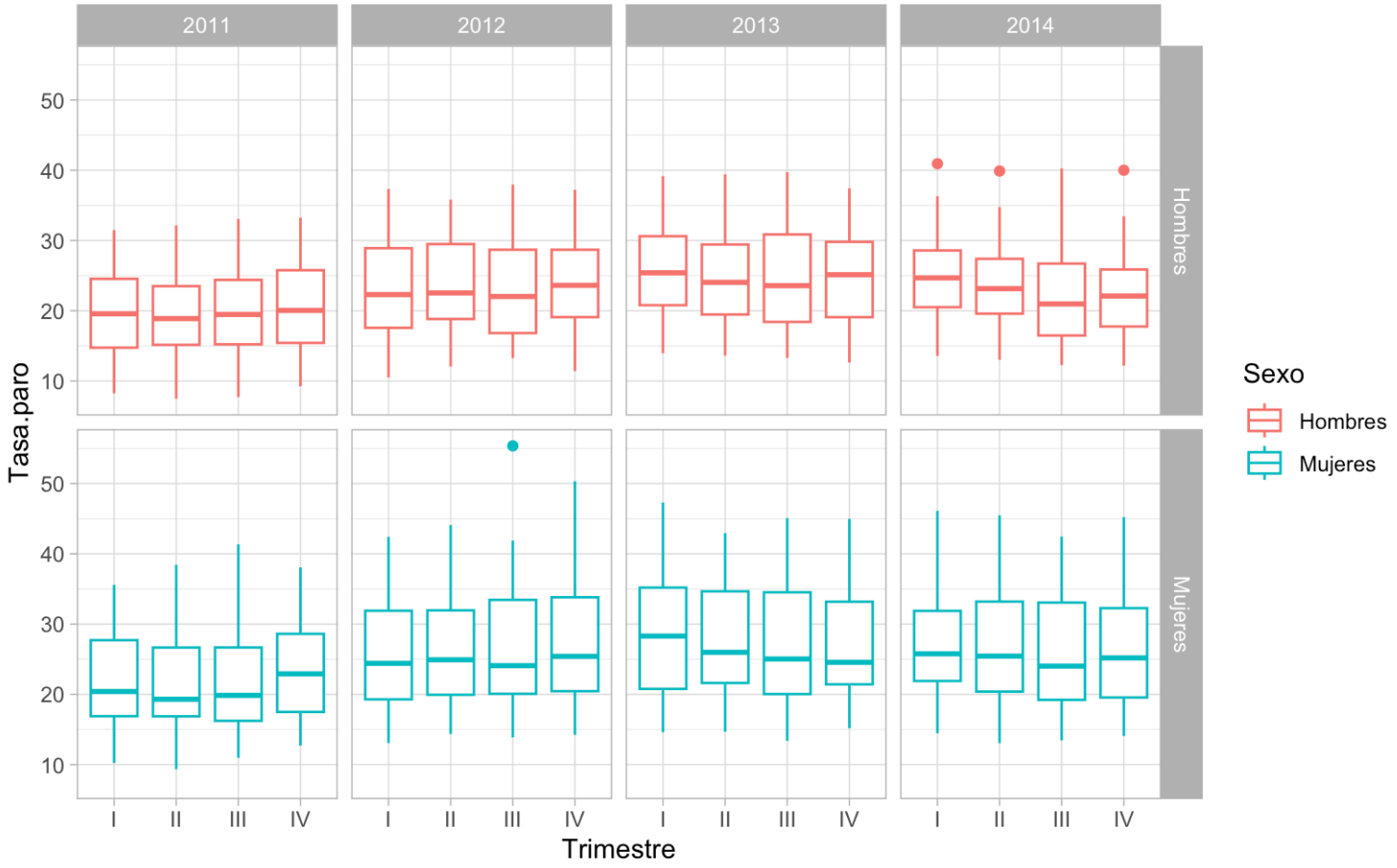
Se puede subdividir el lienzo de acuerdo a dos (¡o más!) variables:

```
mortalidad[,c("zone", "sex"):= tstrsplit(grupo, " ")]
ggplot(mortalidad, aes(x = edad, y = tasa)) +
  geom_bar(stat="identity") +
  facet_grid(zone ~ sex)
```



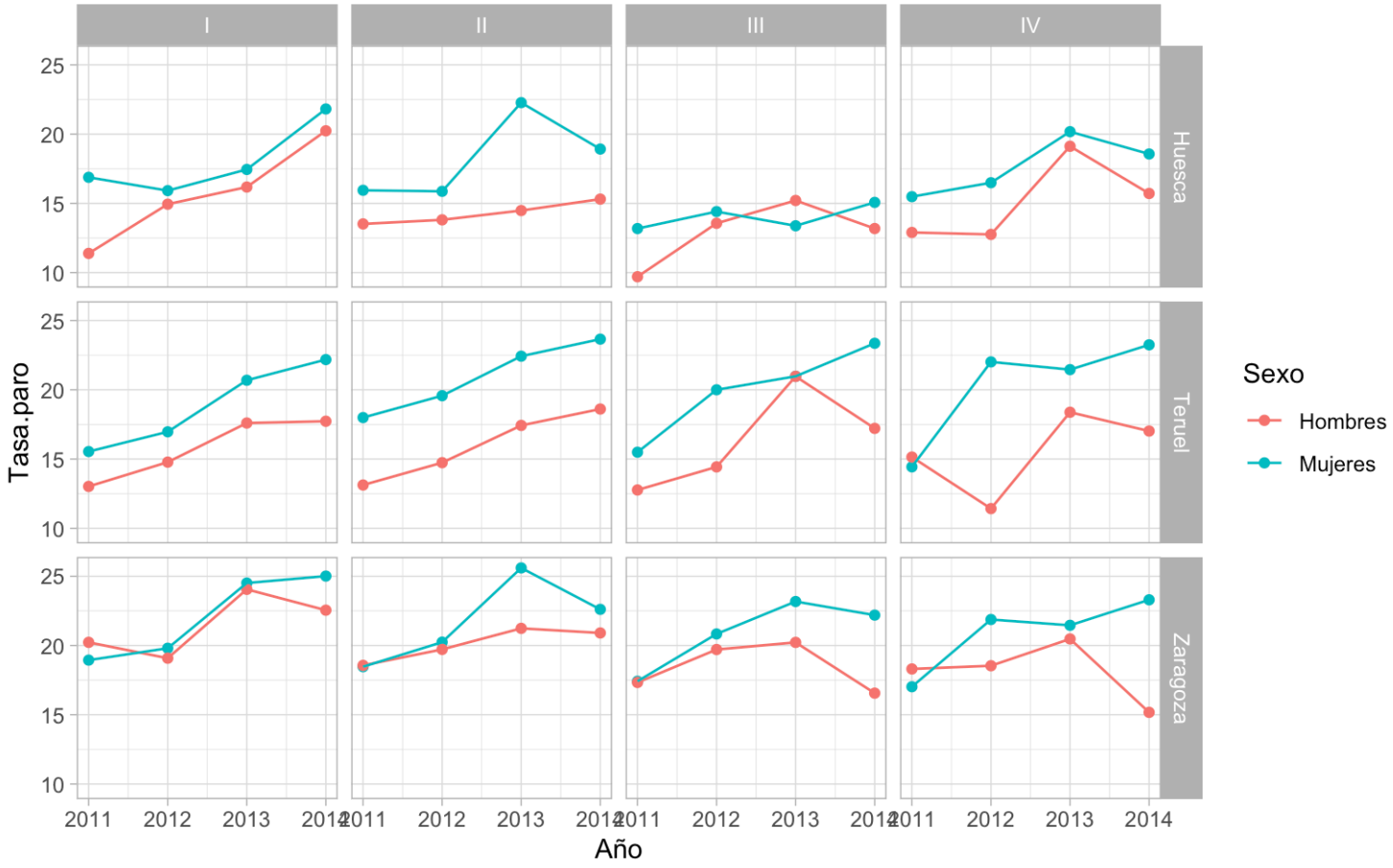
Ejercicio

Elaborar el siguiente gráficos sobre la evolución del paro



Otro ejercicio más

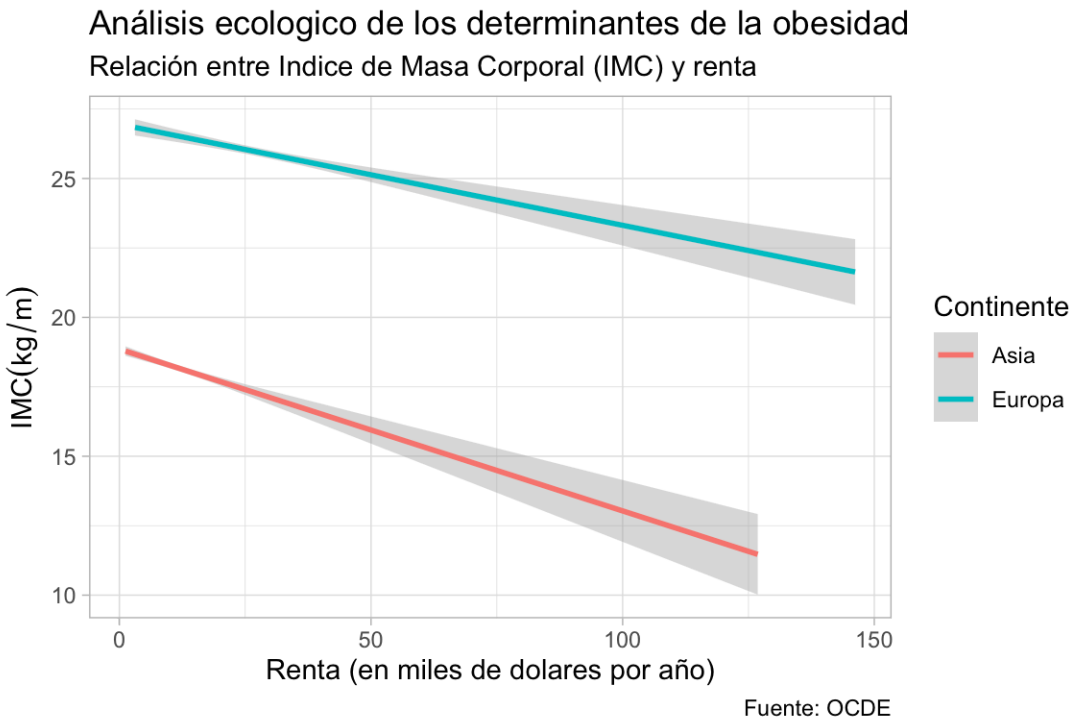
Para este segundo gráfico, limitar la base a las provincias de Zaragoza, Huesca y Teruel.



Etiquetas

Las estéticas se pueden etiquetar con la función `labs`. Esta misma función se puede usar para añadir un título, un subtitulo o una nota al pie del gráfico:

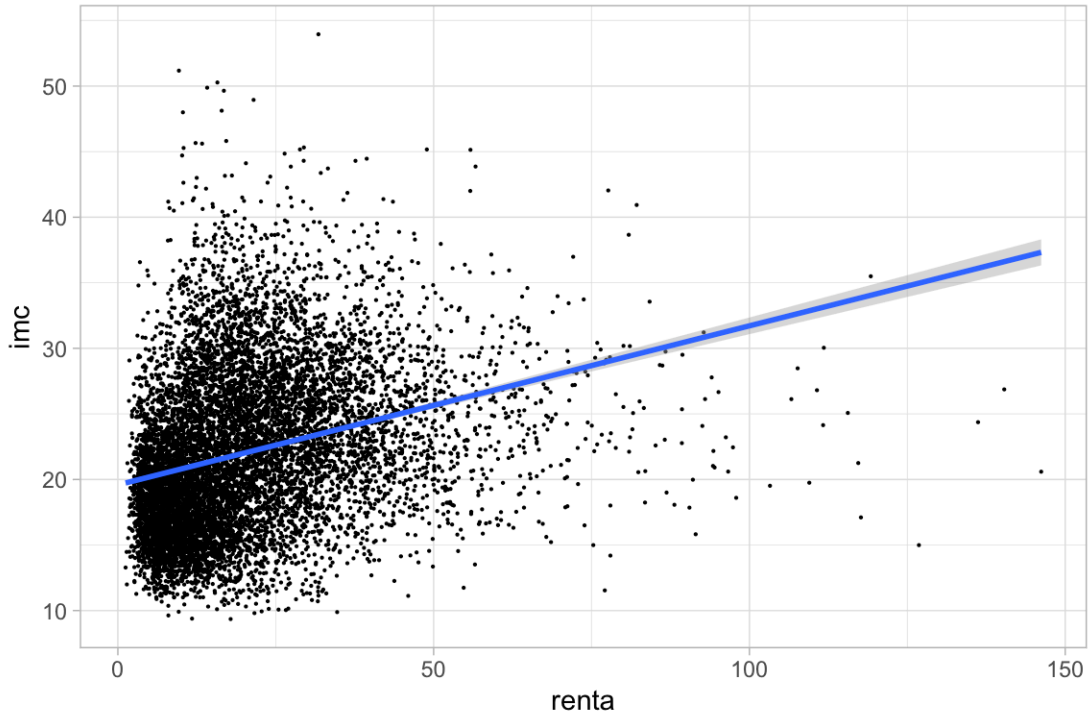
```
obesidad<-fread("data/obesidad.csv")
p<-ggplot(obesidad,aes(x=renta,y=imc,color=region))+geom_smooth(method="lm")
p + labs(x = "Renta (en miles de dolares por año)", y = quote(IMC (kg/m)), color = "Continente",
            title="Análisis ecologico de los determinantes de la obesidad",
            subtitle="Relación entre Indice de Masa Corporal (IMC) y renta", caption = "Fuente: OCDE")
```



Escalas

La escala por defecto de una estética no es siempre la adecuada para una buena representación:

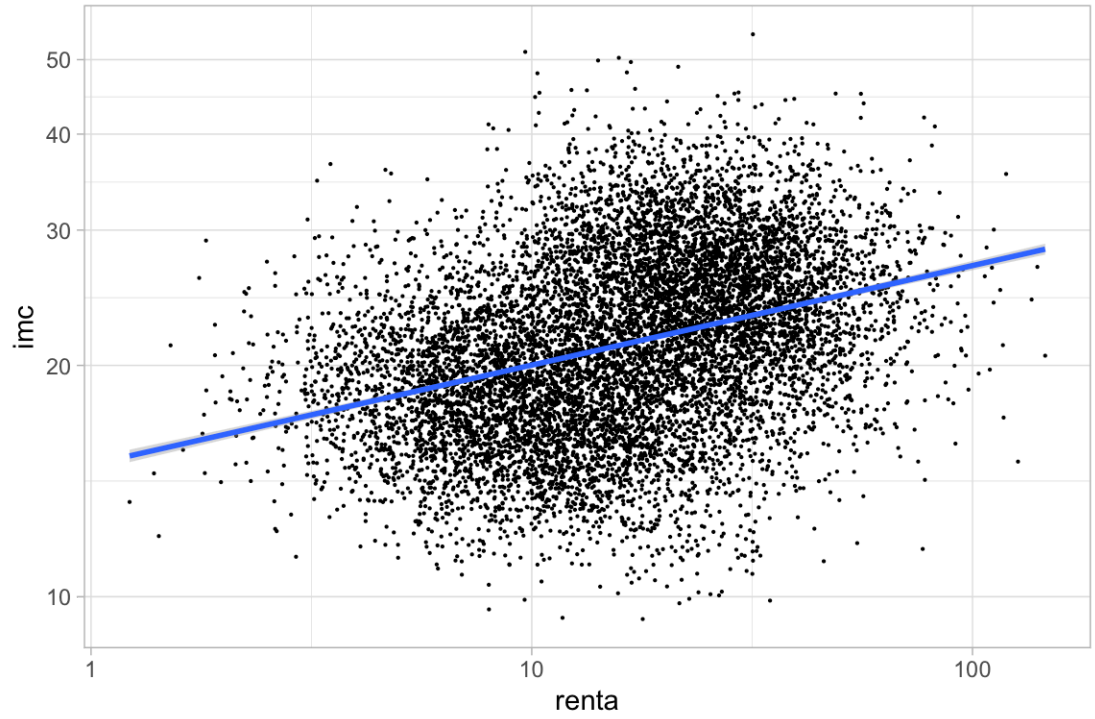
```
p<-ggplot(obesidad,aes(x=renta,y=imc))+geom_point(size=.1)+geom_smooth(method="lm")
p
```



Transformación

La escala de una estética puede ser modificada para mejorar la claridad de la representación:

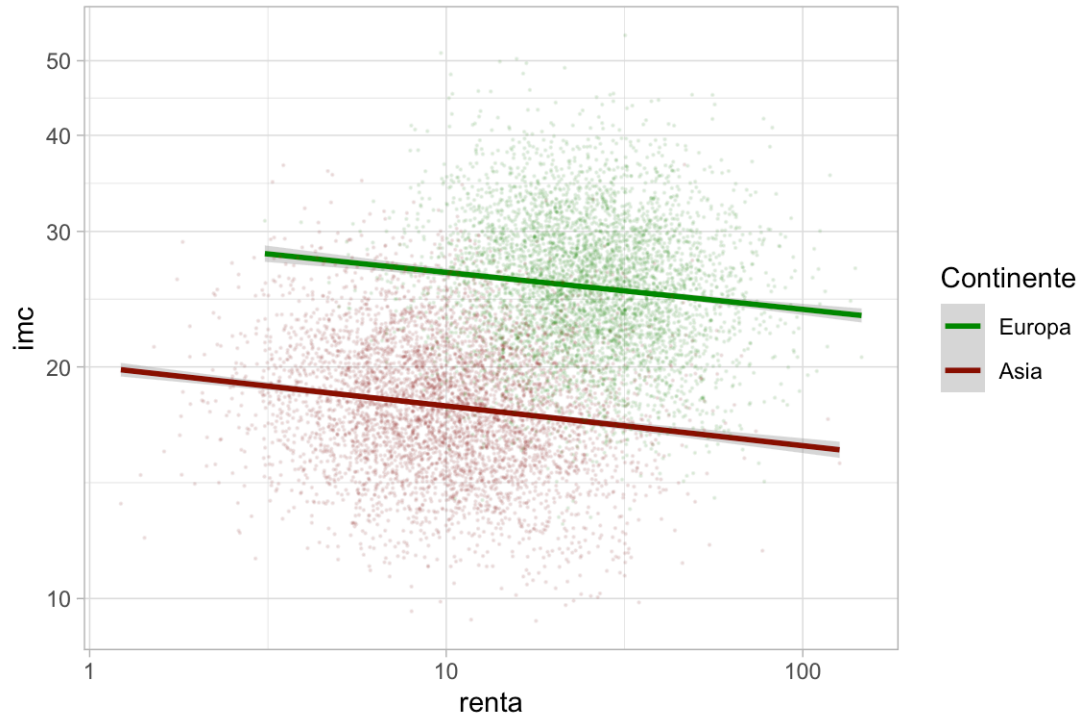
```
p+scale_x_log10()+scale_y_continuous(breaks=seq(10,50,10),trans="log")
```



Transformación

Cada una de las estéticas (incluido el color, la transparencia, …) tiene escala que puede ser configurada:

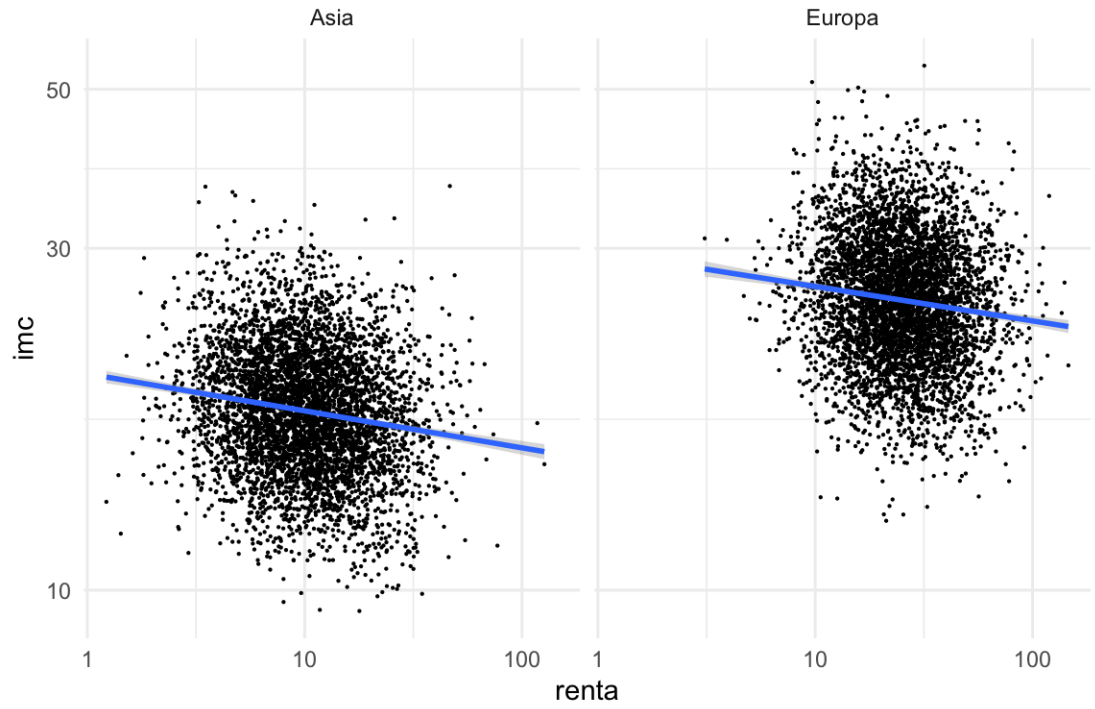
```
ggplot(obesidad,aes(x=renta,y=imc,color=region))+geom_smooth(method="lm") +
  geom_point(size=.1,alpha=.1)+
  scale_x_log10()+
  scale_y_continuous(breaks=seq(10,50,10),trans="log")+
  scale_color_manual("Continente",values=c("green4","red4"),limits=c("Europa","Asia"))
```



Temas

- Los *temas* de `ggplot2` permiten modificar aspectos estéticos del gráfico (ejes, colores de fondo, tamaño de los caracteres, …) para adecuarse a criterios de estilo de publicación.
- Existen muchos temas predefinidos (ver https://ggplot2.tidyverse.org/reference/ggtheme.html) y el paquete `ggthemes` para temas que se ajustan al estilo de reconocidas revistas científicas.
- El tema que usa `ggplot2` por defecto es `theme_grey`, aquí otro más minimalista:

```
p + facet_grid(~region) + scale_y_log10() + scale_x_log10() + theme_minimal()
```



Exportación de los graficos

Una vez creado un gráfico, es posible exportarlo en diversos formatos:

- Imagen tipo bitmap (`jpeg`, `png`, `bmp`, `tiff`,...)
- Imagen vectorial (`pdf`, `svg`,...)

La función `ggsave` guarda en un fichero el último gráfico generado con `ggplot2` con el formato indicado en la extensión del nombre del fichero que se quiere generar:

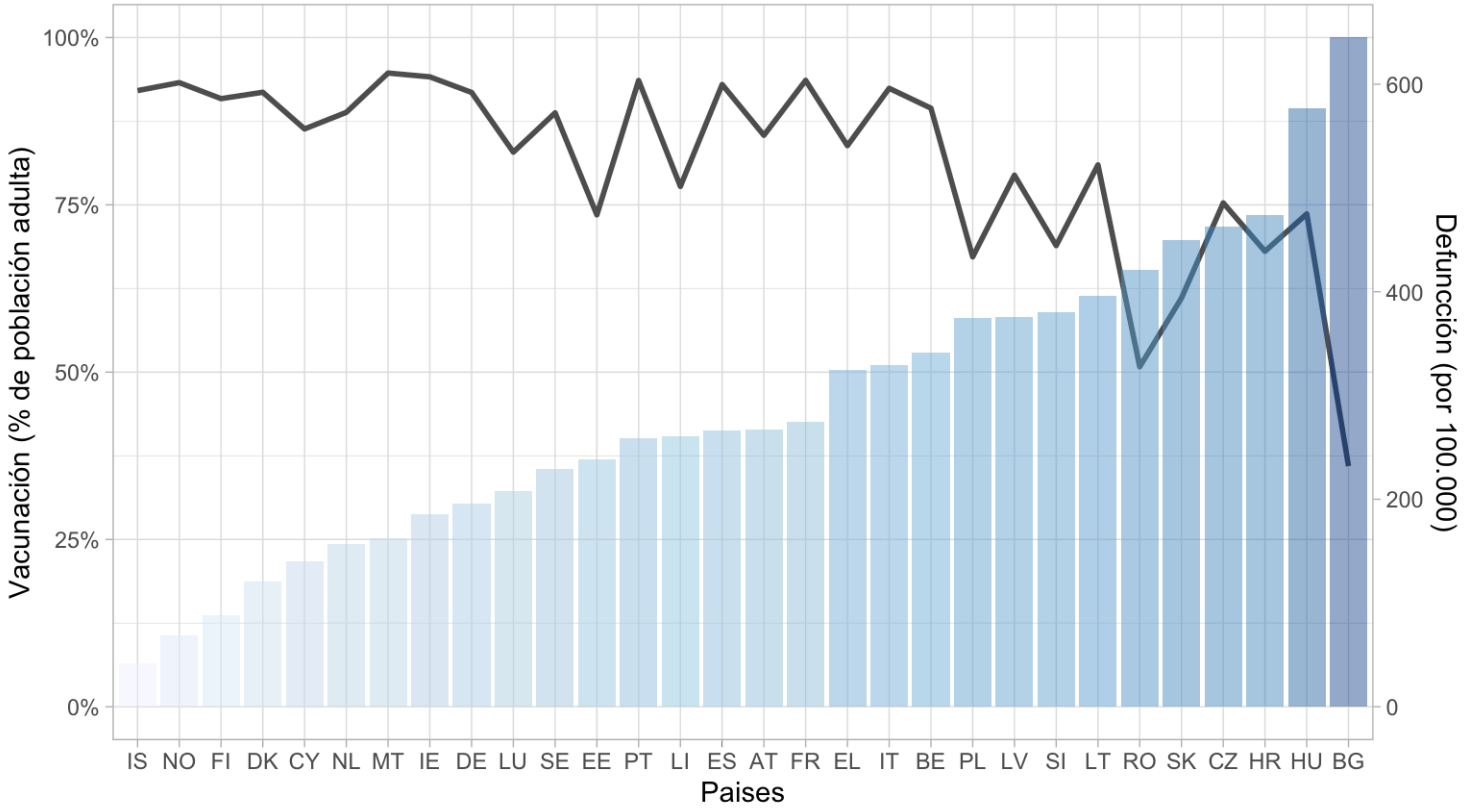
```
ggplot(obesidad,aes(x=renta,y=imc,color=region))+geom_smooth(method="lm")
ggsave("obesidad.pdf")
#ggsave("mortalidad.pdf", width = 20, height = 20, units = "cm")
ggsave("obesidad.png")
```

Las imágenes vectoriales tienen una resolución “infinita” y suelen ocupar poca memoria. Sin embargo, no todos los editores de texto admiten este tipo de formato.

Unos graficos destacados

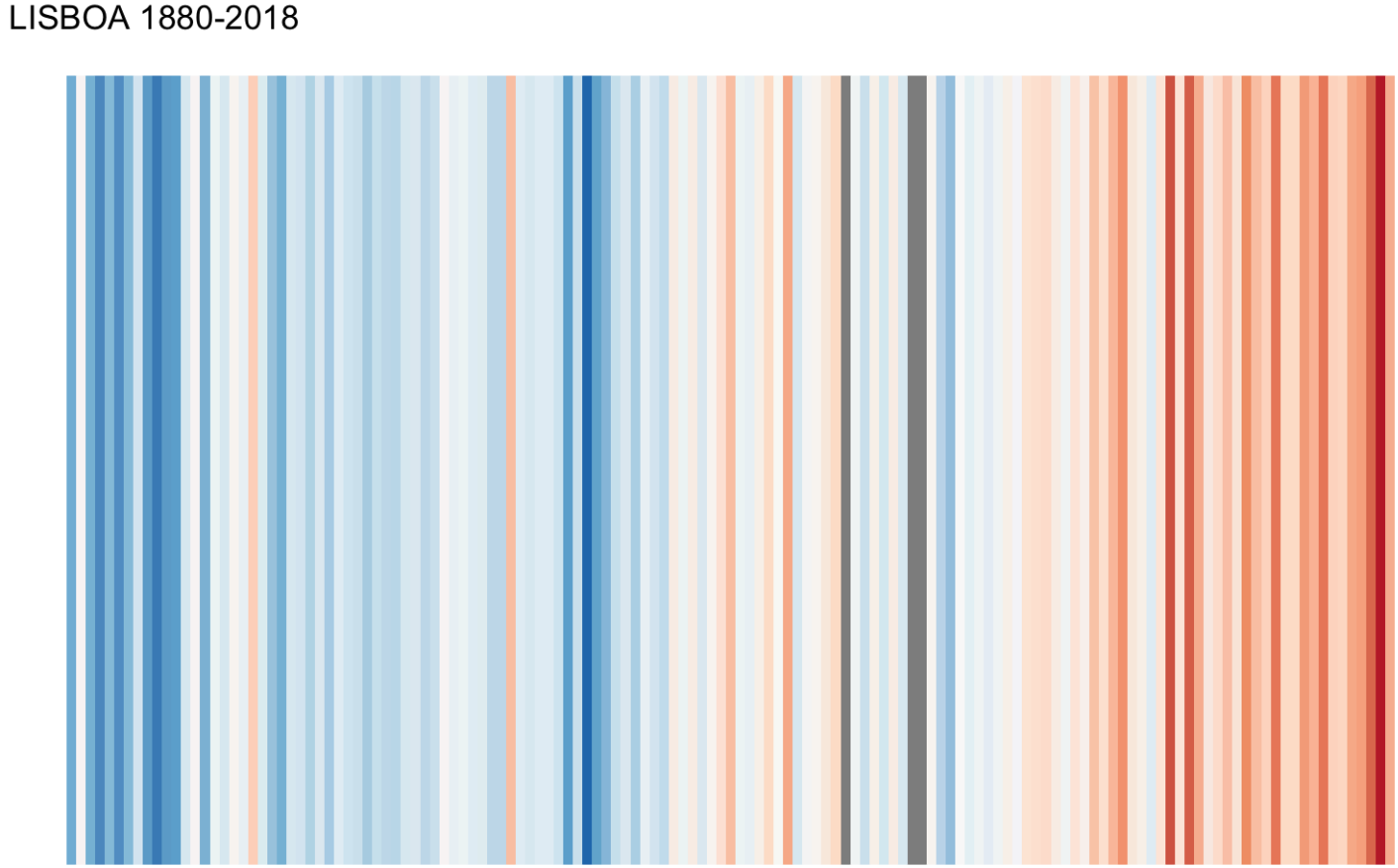
Dos ejes

Representar en un sólo gráfico las variaciones de las tasas de mortalidad por COVID y vacunación entre países de la UE (ver `?sec_axis`).



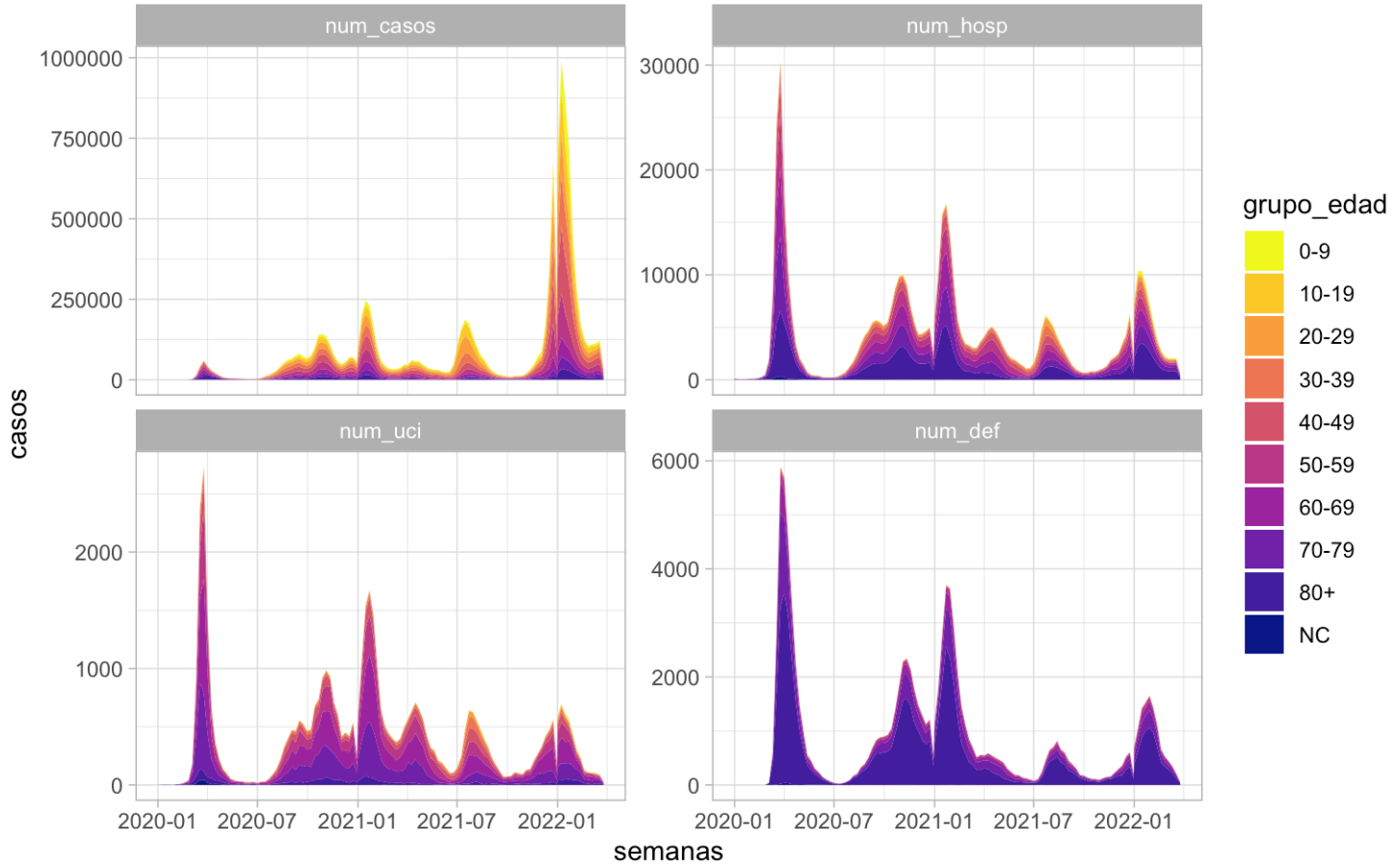
Warming stripes

A partir de los datos de temperaturas anuales en Lisboa de 1880 a 2008 (base de datos `temp_lisboa.csv`), crear y exportar el siguiente gráfico, utilizando la capa `geom_tile`, el tema vacio `theme_void()` y la paleta de colores “RdBu” `scale_fill_distiller(palette = 'RdBu')`.



Representación por áreas

Utilizando la base de COVID, describir en un único gráfico la evolución de esta pandemia en España de acuerdo a la edad y gravedad (se sugiere utilizar la geometría `geom_area`).



Comparación anual de series

Describir en un único gráfico la evolución de las defunciones en España por año respecto a las defunciones esperadas (base de datos del MOMO).