Análisis gráfico con ggplot2

Introducción a ggplot2

ggplot2 es un paquete de R para producir gráficos estadísticos o de datos, pero es diferente a la mayoría de los otros paquetes de gráficos porque tiene una gramática particular. Esta gramática está formada por un conjunto de componentes independientes que pueden componerse de muy diversas formas. Esto hace que ggplot2 sea una herramienta muy potente pues no está limitada a un conjunto de gráficos preespecificados, sino que puede crear nuevos gráficos que se adapten con precisión a cada problema. ggplot2 también es fácil de aprender (aunque puede tomar un poco de tiempo olvidar sus ideas preconcebidas de otras herramientas gráficas).

- Ventajas de ggplot2

- Dispone de una gramática de gráficos consistente (Wilkinson, 2005);
- Especificación de gráficos con un alto nivel de versatilidad;
- Amplio sistema temático de gráficos;
- Sistema gráfico maduro y completo;
- Muchos usuarios, e innovaciones muy frecuentes.

- A qué nos referimos con la gramática de gráficos

La idea básica consiste en especificar de forma independiente los diferentes bloques que dan lugar al gráfico. Estos bloques incluyen:

- data
- aesthetic mapping
- geometric object
- statistical transformations
- scales
- coordinate system
- position adjustments
- faceting

ggplot2 es un paquete elaborado por Hadley Wickham y Winston Chang que implementa la Gramática de Gráficos de Wilkinson. El énfasis de ggplot2 está en la exploración rápida de datos, y especialmente de datos de alta dimensión. En el siguiente link puedes encontrar un completo repositorio de posibilidades gráficas con ggplot2, vale la pena que lo explores con detenimiento.

https://r-graph-gallery.com/ggplot2-package.html

Geoms_ Cómo seleccionar el tipo de gráfico

Geoms se refiere al objeto geométrico, algunos ejemplos:

- geom_bar() Bars
- geom_point() Points
- geom_line() Lines
- geom_ribbon() Ribbons, y range with continuous x values
- geom_polygon() Polygon, a filled path
- geom_pointrange() Vertical line with a point in the middle
- geom_linerange() An interval represented by a vertical line
- geom_path() Connect observations in original order
- geom_histogram() Histograms
- geom_text() Text annotations
- geom_violin() Violin plot (another name for a beanplot)
- geom_map() Polygons on a map

Normalmente los diferentes elementos del gráfico se van añadiendo de forma consecutiva en distintas capas (layers). Para añadir una nueva capa se usa el signo +.

Para utilizar ggplot2 deberemos instalar previamente el paquete ggplot2, install.packages("ggplot2"), y cargarlo cada vez que lo queramos a utilizar. Aprovecho y cargo otros paquetes importantes para algunas especificaciónes de los datos.

```
library(ggplot2)
library(tibble)
library(scales)
library(cowplot)
```

Gráficos de columnas/barras

Comenzamos representando gráficamente el número de colegiados en 2021 en la provincia de Las Palmas en seis profesiones sanitarias. De acuerdo con los datos disponibles en el INE construyo los siguientes vectores. Vectores que posteriormente introduciré en una tabla con el paquete "tibble".

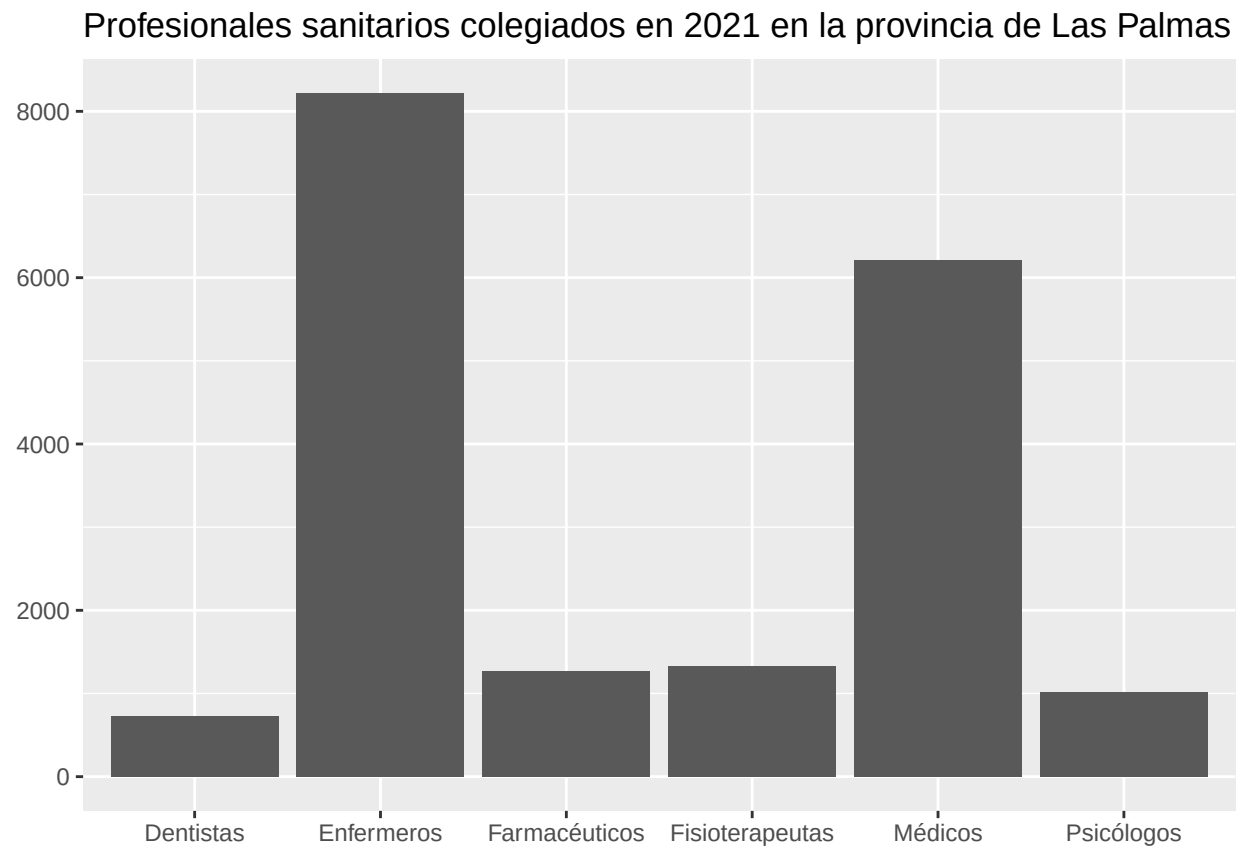
```
titulo <- c("Médicos", "Dentistas", "Farmacéuticos",
            "Psicólogos", "Enfermeros", "Fisioterapeutas")

colegiados <- c(6216, 730, 1264, 1014, 8220, 1323)


Tabla <- tibble(
  rank = 1:6,
  titulo <- c("Médicos", "Dentistas", "Farmacéuticos",
              "Psicólogos", "Enfermeros", "Fisioterapeutas"),
  colegiados <- c(6216, 730, 1264, 1014, 8220, 1323)
)

Tabla
```

```
## # A tibble: 6 x 3
##    rank `titulo <- ...` `colegiados <- c(6216, 730, 1264, 1014, 8220, 1323)`
##   <int> <chr>                                                          <dbl>
## 1     1 Médicos                                                         6216
## 2     2 Dentistas                                                        730
## 3     3 Farmacéuticos                                                   1264
```

```
## 4     4 Psicólogos                                                1014
## 5     5 Enfermeros                                                8220
## 6     6 Fisioterapeutas                                           1323
```

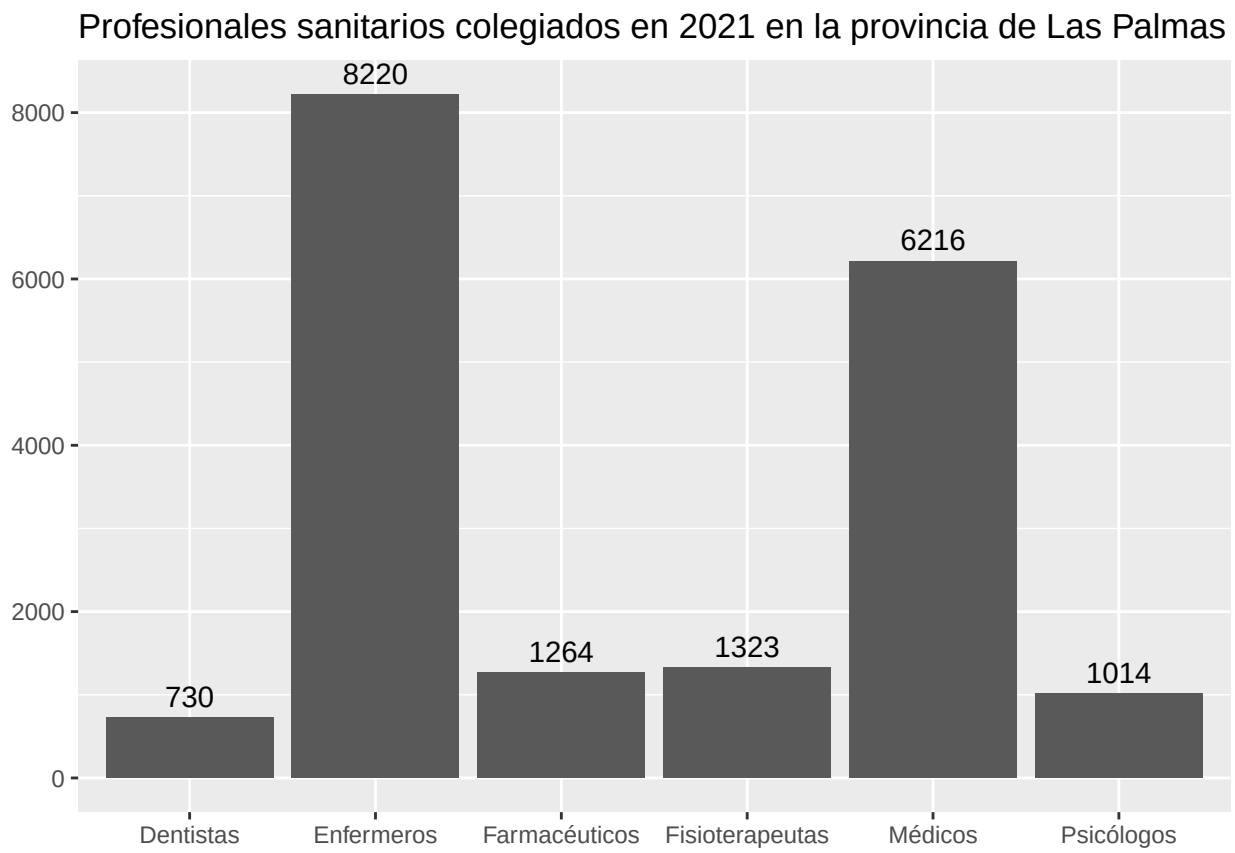
Utilizando geom_col(...)

```
ggplot(
  Tabla,
  aes(
    x =  titulo,
    y = colegiados)) +
  geom_col() +
  ggtitle("Profesionales sanitarios colegiados en 2021 en la provincia de Las Palmas") +
  xlab(NULL) +
  ylab(NULL)
```



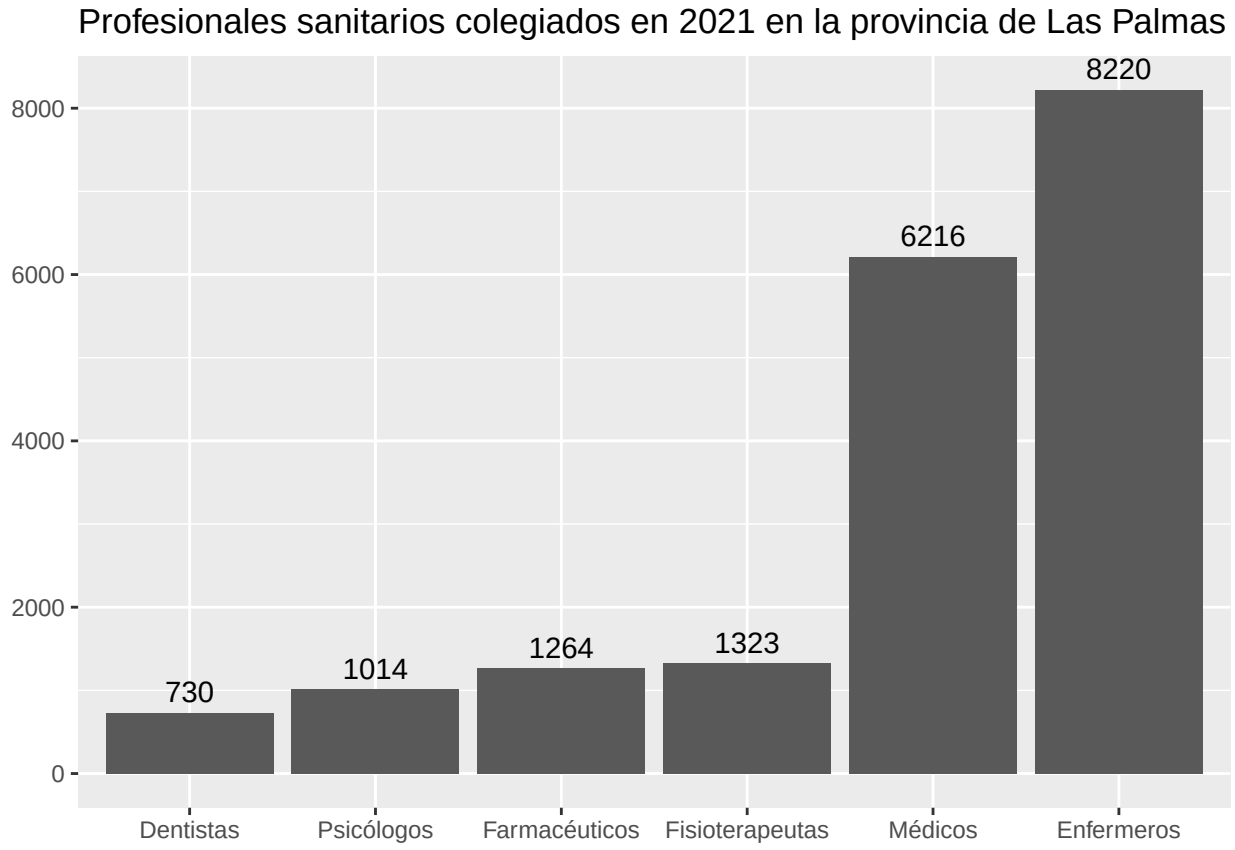

```
# Añadiendo las etiquetas de datos
 ggplot(
   Tabla,
   aes(
     x =  titulo,
     y = colegiados)) +
 geom_col() +
```

```
  geom_text(aes(label = colegiados), vjust = -0.5) +
  ggtitle("Profesionales sanitarios colegiados en 2021 en la provincia de Las Palmas") +
  xlab(NULL) +
  ylab(NULL)
```

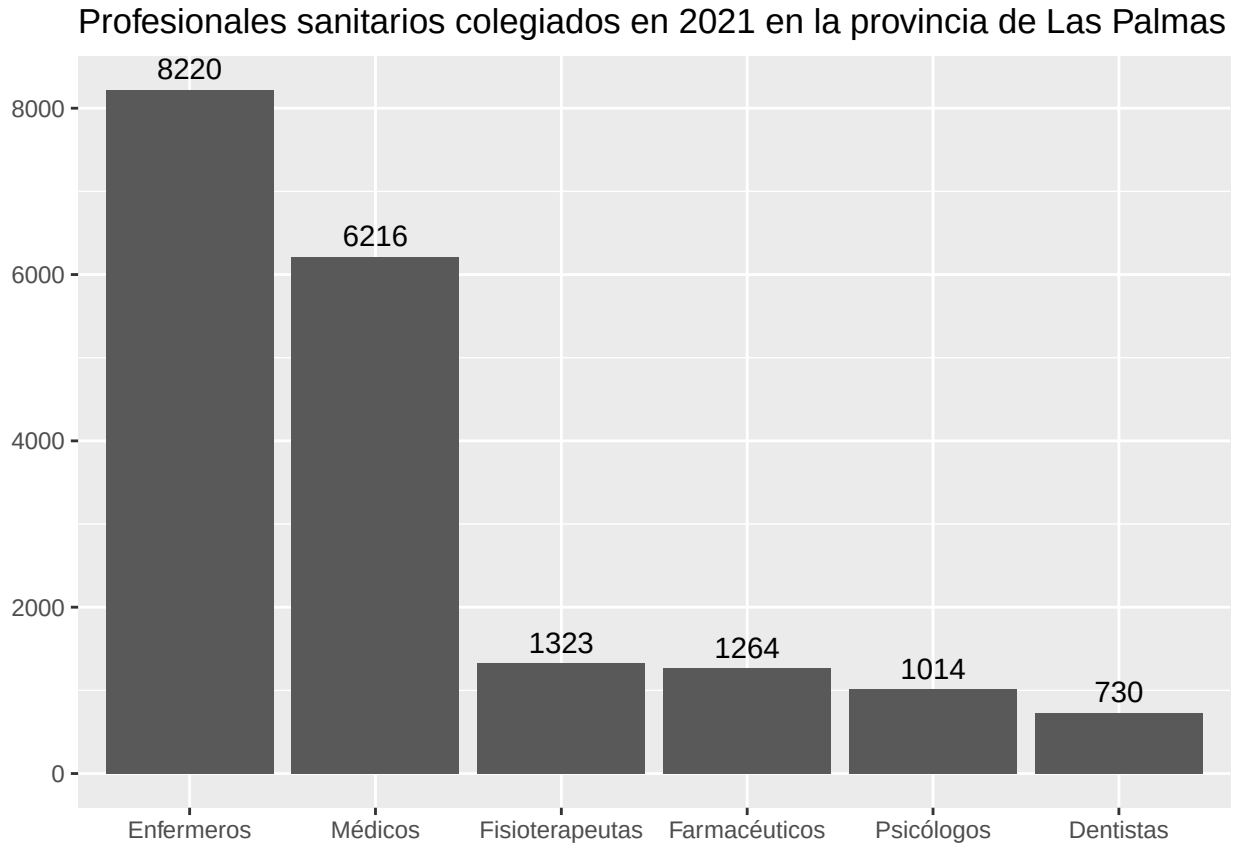


A continuación ordenados por frecuencias ascendente/descendente

```
ggplot(Tabla,
       aes(x = reorder(titulo, colegiados),
           y = colegiados )) +
  geom_col() +
  geom_text(aes(label = colegiados), vjust = -0.5) +
  ggtitle("Profesionales sanitarios colegiados en 2021 en la provincia de Las Palmas") +
  xlab(NULL) +
  ylab(NULL)
```

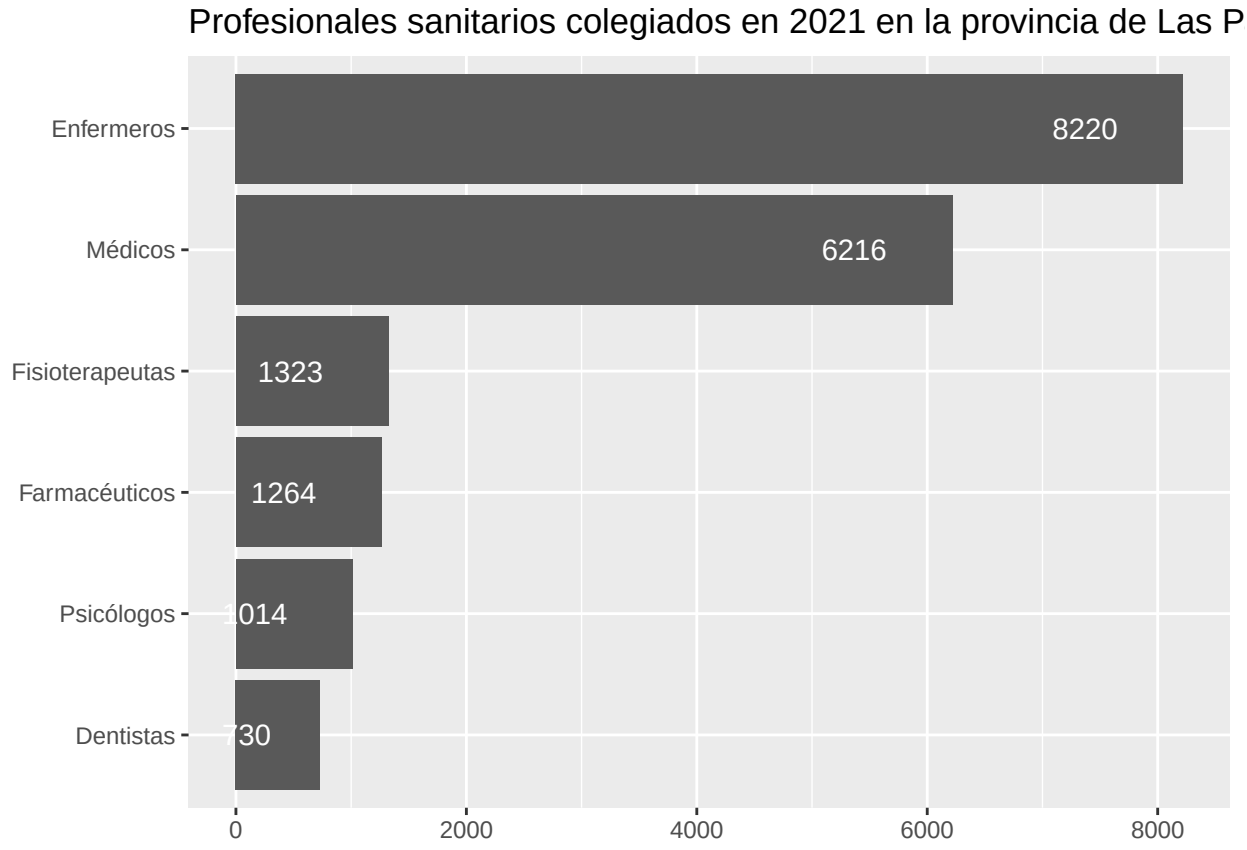



```
ggplot(Tabla,
       aes(x = reorder(titulo, -colegiados),
           y = colegiados )) +
  geom_col() +
  geom_text(aes(label = colegiados), vjust = -0.5) +
  ggtitle("Profesionales sanitarios colegiados en 2021 en la provincia de Las Palmas") +
  xlab(NULL) +
  ylab(NULL)
```



Finalmente con columnas horizontales

```
ggplot(Tabla,
       aes(x = reorder(titulo, colegiados),
           y = colegiados )) +
  geom_col() +
  geom_text(aes(label = colegiados), hjust = 2, colour="white") +
  ggtitle("Profesionales sanitarios colegiados en 2021 en la provincia de Las Palmas") +
  coord_flip()+
  xlab(NULL) +
  ylab(NULL)
```



Utilizamos geom_bar(...)

Con este geom_ podemos leer datos desde un fichero externo, y que ggplot cuente la frecuencia de repetición de cada caracter. Por ejemplo, a continuación genero una muestra de profesionales sanitarios de tamaño n=100, de acuerdo con las proporciones de los datos del INE. A partir de la muestra anterior creo el data frame DF.

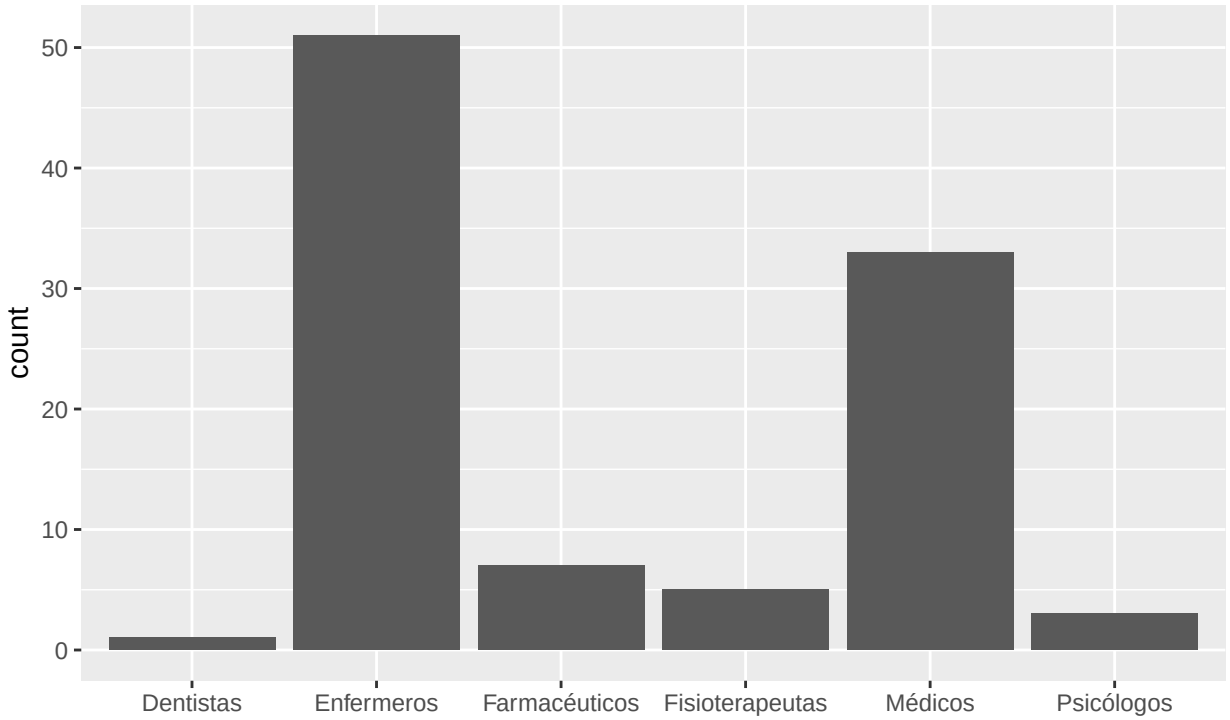
```
set.seed(1234)
prof.sanita <- sample(titulo, 100, replace=TRUE, prob=colegiados/sum(colegiados))

DF <-data.frame(prof.sanita)
```

Con geom_bar(...) construyo el gráfico de barras sobre dicho data frame,

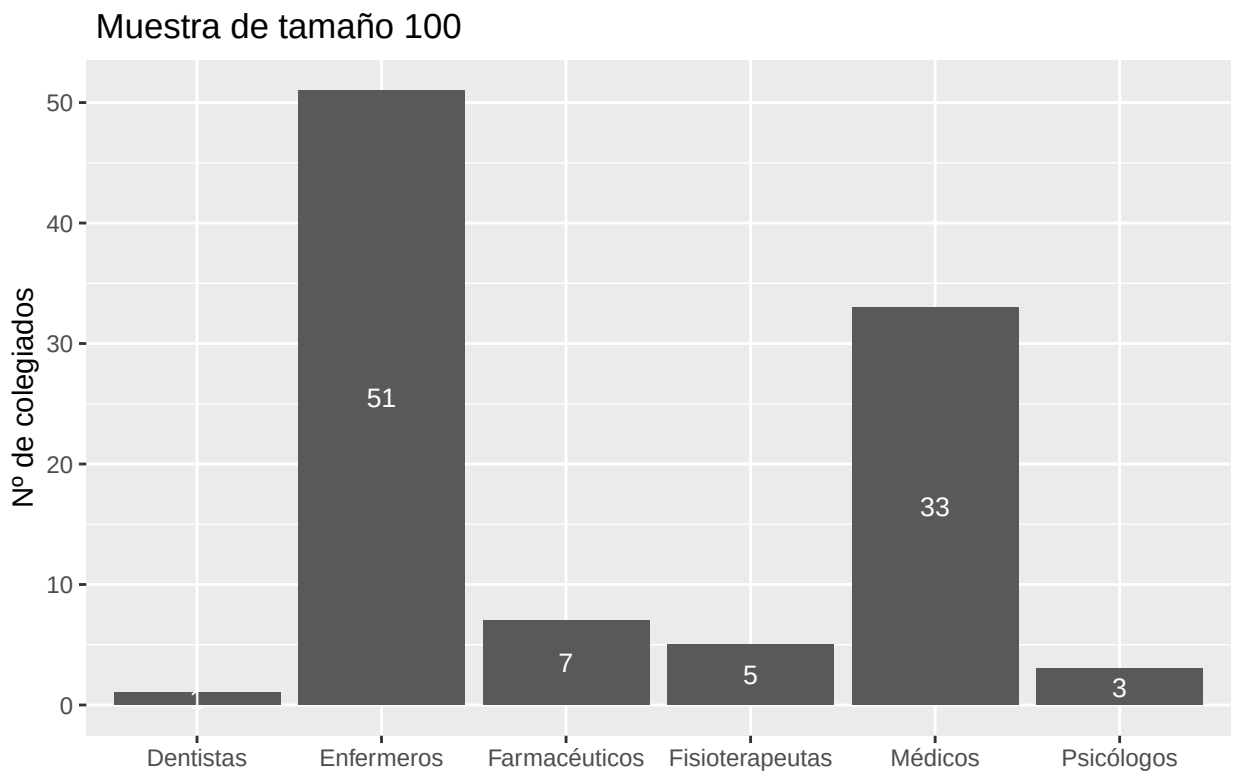
```
ggplot(DF, aes(x = prof.sanita)) + # y no está definida en el aesthetic
  geom_bar() +
  ggtitle("Profesionales sanitarios colegiados en 2021 en la provincia de Las Palmas:
          \n Muestra de tamaño 100") +
  xlab(NULL)
```

Profesionales sanitarios colegiados en 2021 en la provincia de Las Palmas:

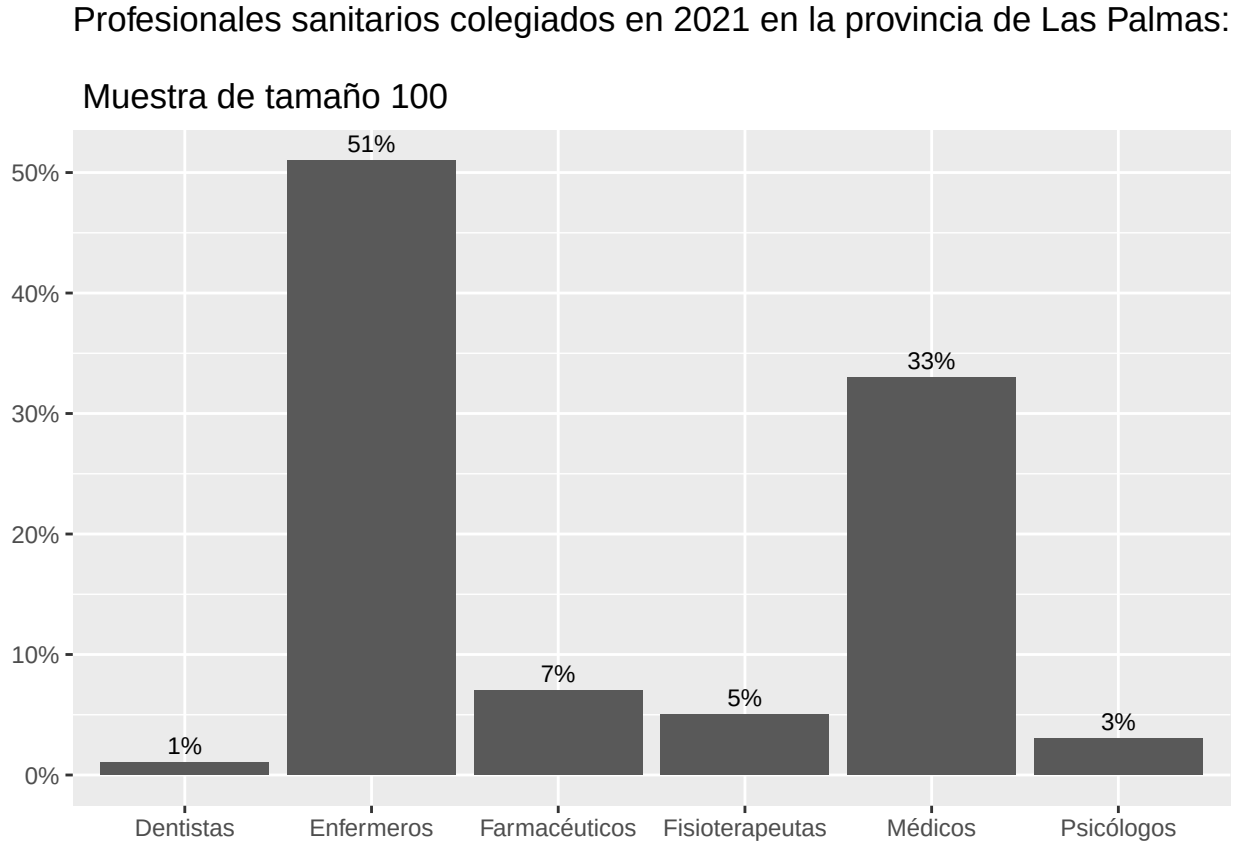
Muestra de tamaño 100




```
# Con etiquetas de valores
ggplot(data = DF, aes(x = prof.sanita)) +
  geom_bar(stat = "count") +
  ggtitle("Profesionales sanitarios colegiados en 2021 en la provincia de Las Palmas:
          \n Muestra de tamaño 100") +
  stat_count(geom = "text", colour = "white", size = 3.5,
             aes(label = ..count..),position=position_stack(vjust=0.5)) +
  xlab(NULL) +
  ylab("Nº de colegiados")
```




```
# Con etiquetas de valores en porcentajes

  ggplot(data = DF, aes(x = prof.sanita,
                        y = prop.table(stat(count)),
                        label = scales::percent(prop.table(stat(count))))) +
    geom_bar(position = "dodge") +
    ggtitle("Profesionales sanitarios colegiados en 2021 en la provincia de Las Palmas:
          \n Muestra de tamaño 100") +
    geom_text(stat = 'count',
              position = position_dodge(.9),
              vjust = -0.5,
              size = 3) +
    scale_y_continuous(labels = scales::percent) +
    xlab(NULL) +
    ylab(NULL)
```



Ejercicio:

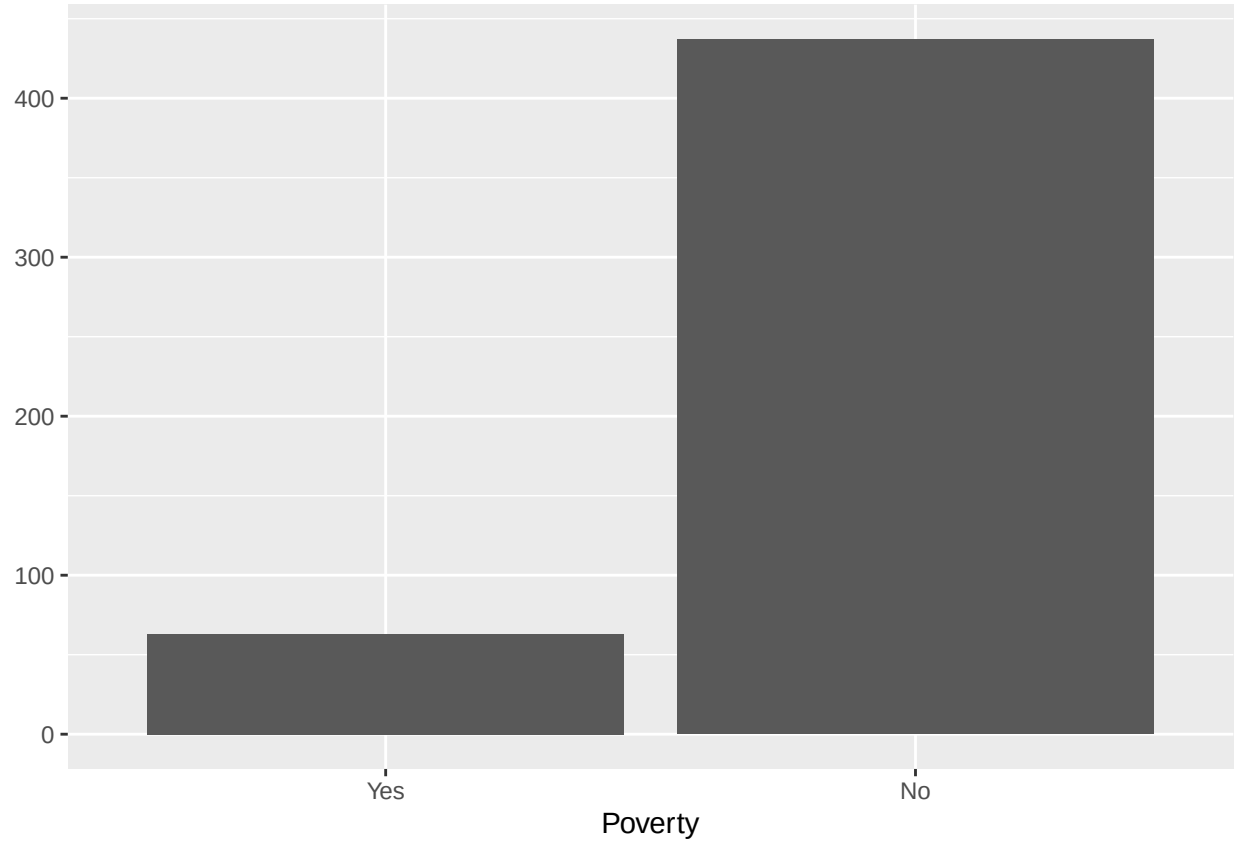
Para practicar las posibilidades de geom_bar(...) te propongo analizar los datos del fichero "healthdisparities.dta". de nuestro repositorio. Carga los datos y etiqueta los valores en las variables cualitativas.

```
library(haven)
HD <- read_dta("healthdisparities.dta")

HD$gender <- factor(HD$gender,
                    levels = c(1,2),
                    labels = c("Male", "Female"))

HD$race <- factor(HD$race,
                    levels = c(1:7),
                    labels = c("latino", "pacific islander", "american Indian",
                               "asian", "african American", "white", "otros"))

HD$poverty <- factor(HD$poverty,
                    levels = c(1, 0),
                    labels = c("Yes", "No"))

HD$doctor <- factor(HD$doctor,
                     levels = c(1, 0),
                     labels = c("Yes", "No"))

HD$racecat <- factor(HD$racecat,
                    levels = c(1:3),
```

```
                    labels = c("White non-Hispanic", "Hispanic", "All other races"))

HD$urban <- factor(HD$urban,
                    levels = c(1, 0),
                    labels = c("Urban", "Rural"))

# DF$race <- ordered( , levels = c(, labels = c())
```

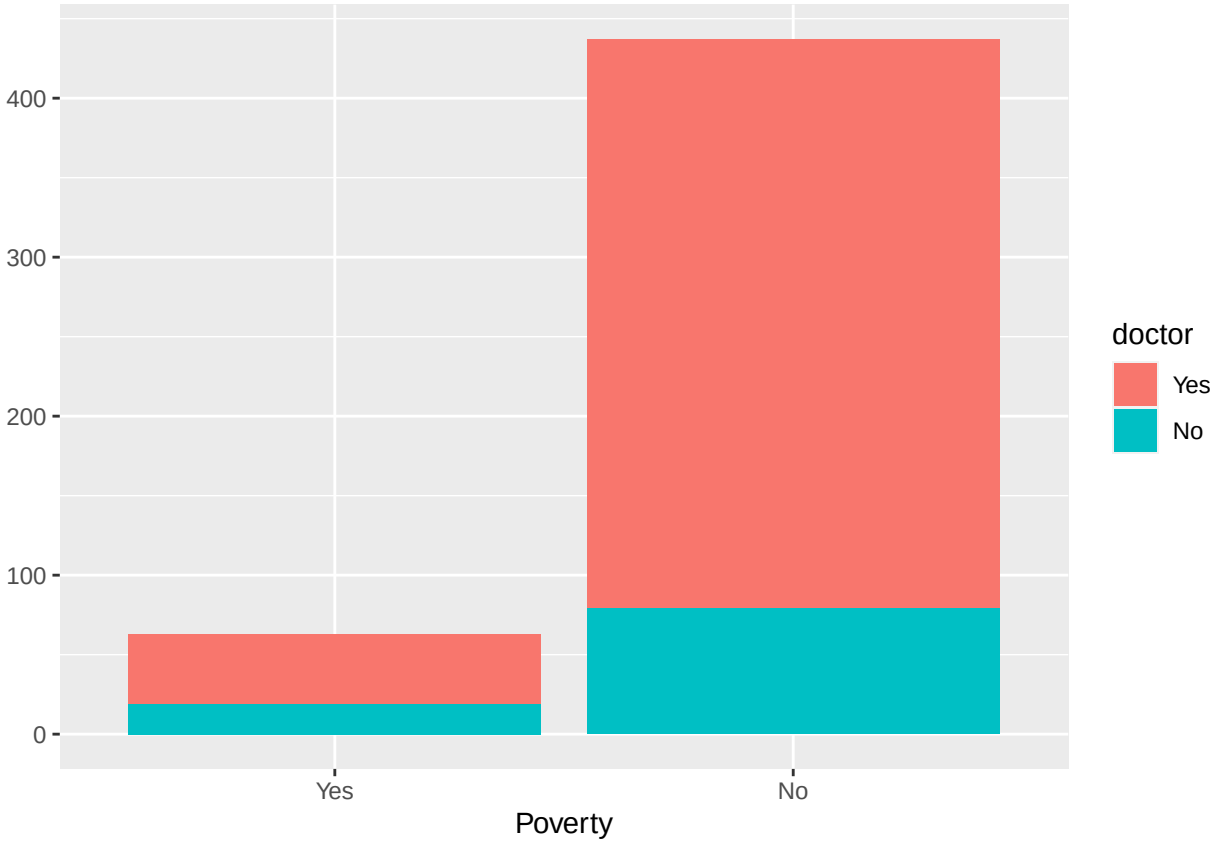
Comenzaremos representando la distribución de los individuos en situación de pobreza.

```
ggplot(HD, aes(poverty)) +
  geom_bar() +
  xlab("Poverty") +
  ylab(NULL)
```



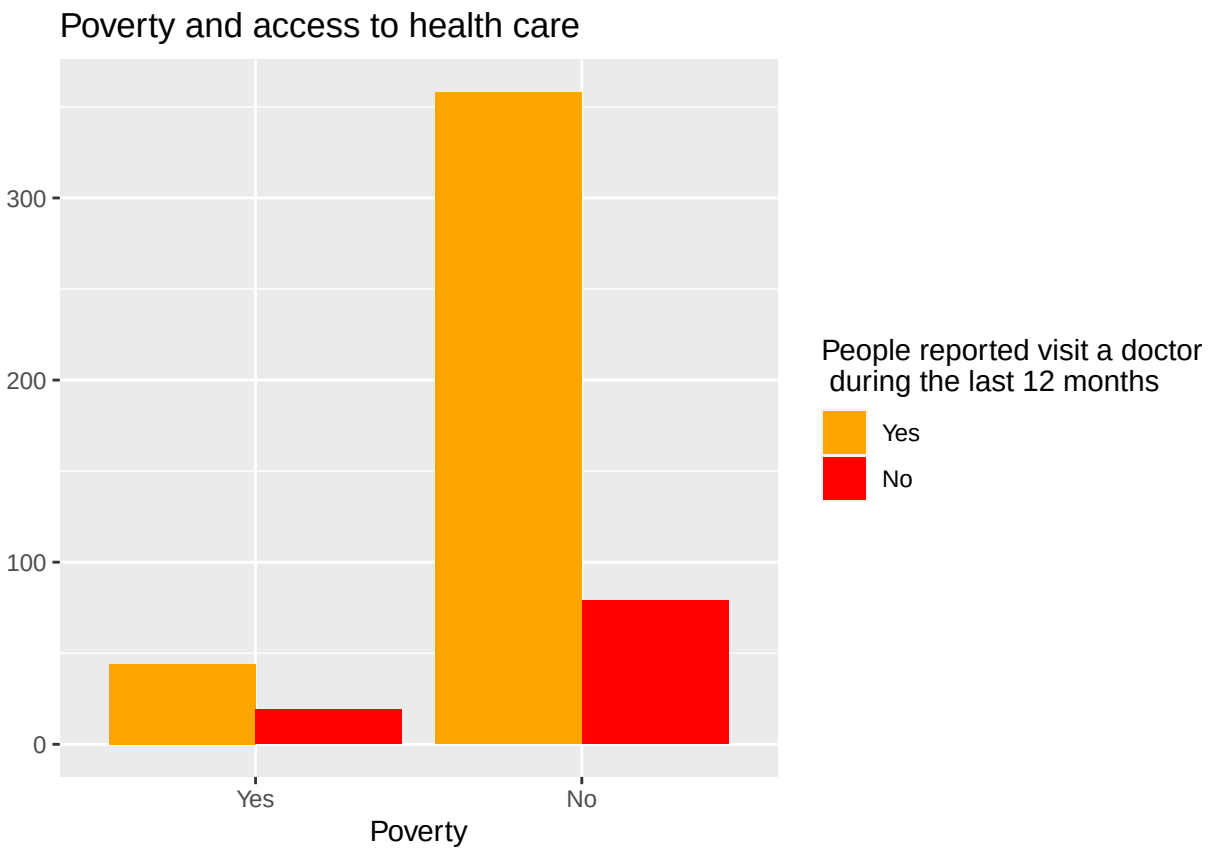
Añadimos una nueva variable. Diferenciamos la distribución de la variable pobreza en función de la visita Yes/No al médico durante el último año.

```
ggplot(HD, aes(poverty, fill = doctor)) +
  geom_bar() +
  xlab("Poverty") +
  ylab(NULL)
```



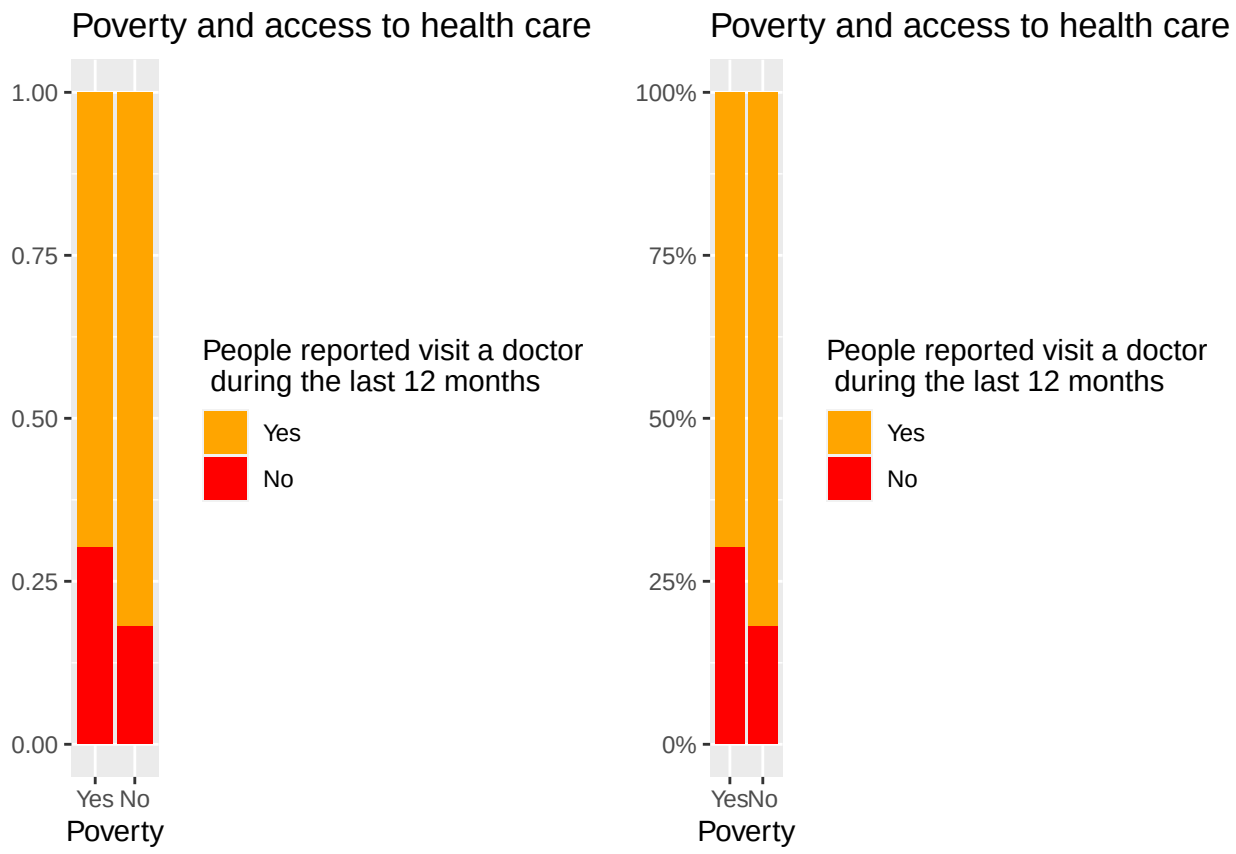
Con el argumento "dodge" presento las barras por separado. Aprovecho para añadir un título más informativo a la leyenda y cambiar los colores.

```
legend_title <- c("People reported visit a doctor \n during the last 12 months")

ggplot(HD, aes(poverty, fill = doctor)) +
  geom_bar(position = "dodge") +
  ggtitle("Poverty and access to health care") +
  xlab("Poverty") +
  ylab(NULL) +
  scale_fill_manual(legend_title,
                    values=c("orange","red"))
```



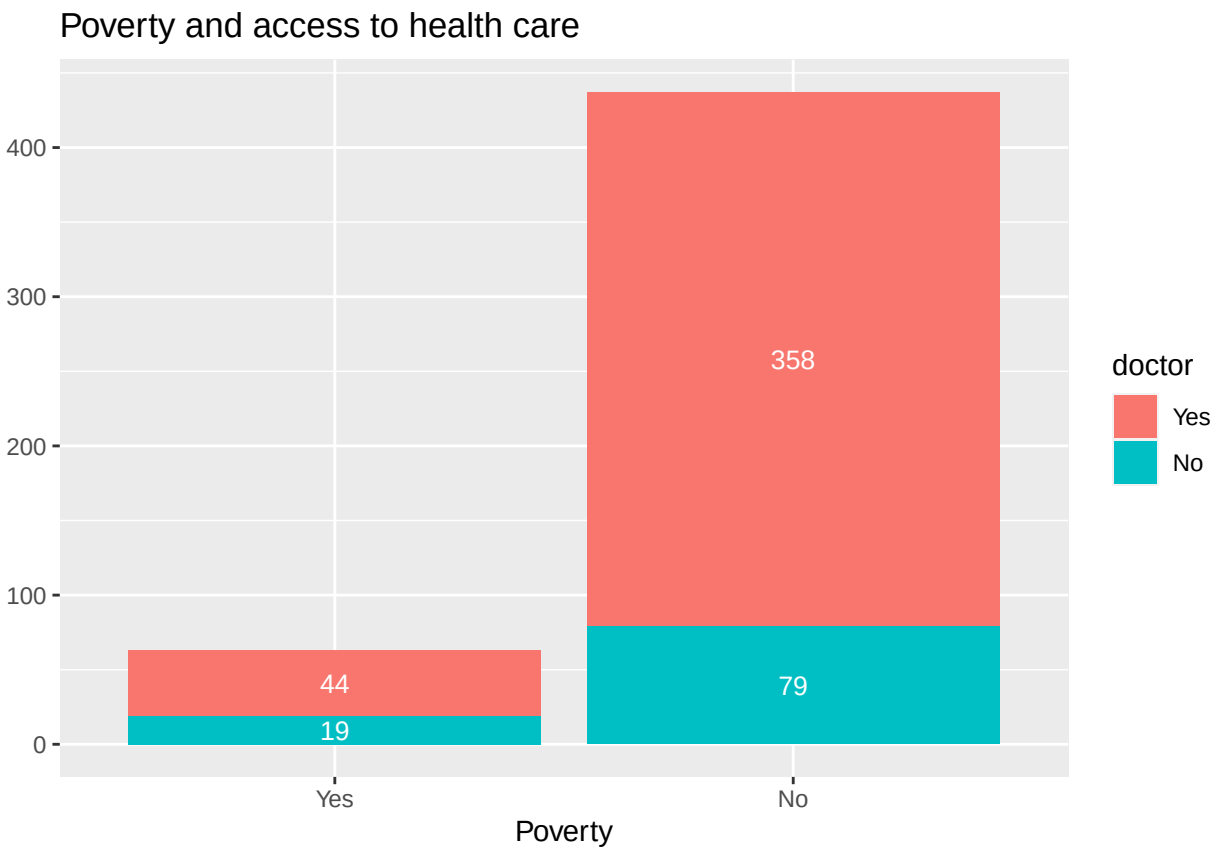
Mostramos conjuntamente el mismo gráfico en proporciones y porcentaje:

```
p1 <- ggplot(HD, aes(poverty, fill = doctor)) +
  geom_bar(position = "fill") +
  ggtitle("Poverty and access to health care") +
  xlab("Poverty") +
  ylab(NULL) +
  scale_fill_manual(legend_title,
                    values=c("orange","red"))

p2 <-ggplot(HD, aes(poverty, fill = doctor)) +
  geom_bar(position = "fill") +
  scale_y_continuous(labels = percent) +
  ggtitle("Poverty and access to health care") +
  xlab("Poverty") +
  ylab(NULL) +
  scale_fill_manual(legend_title,
                    values=c("orange","red"))
plot_grid(p1,p2)
```



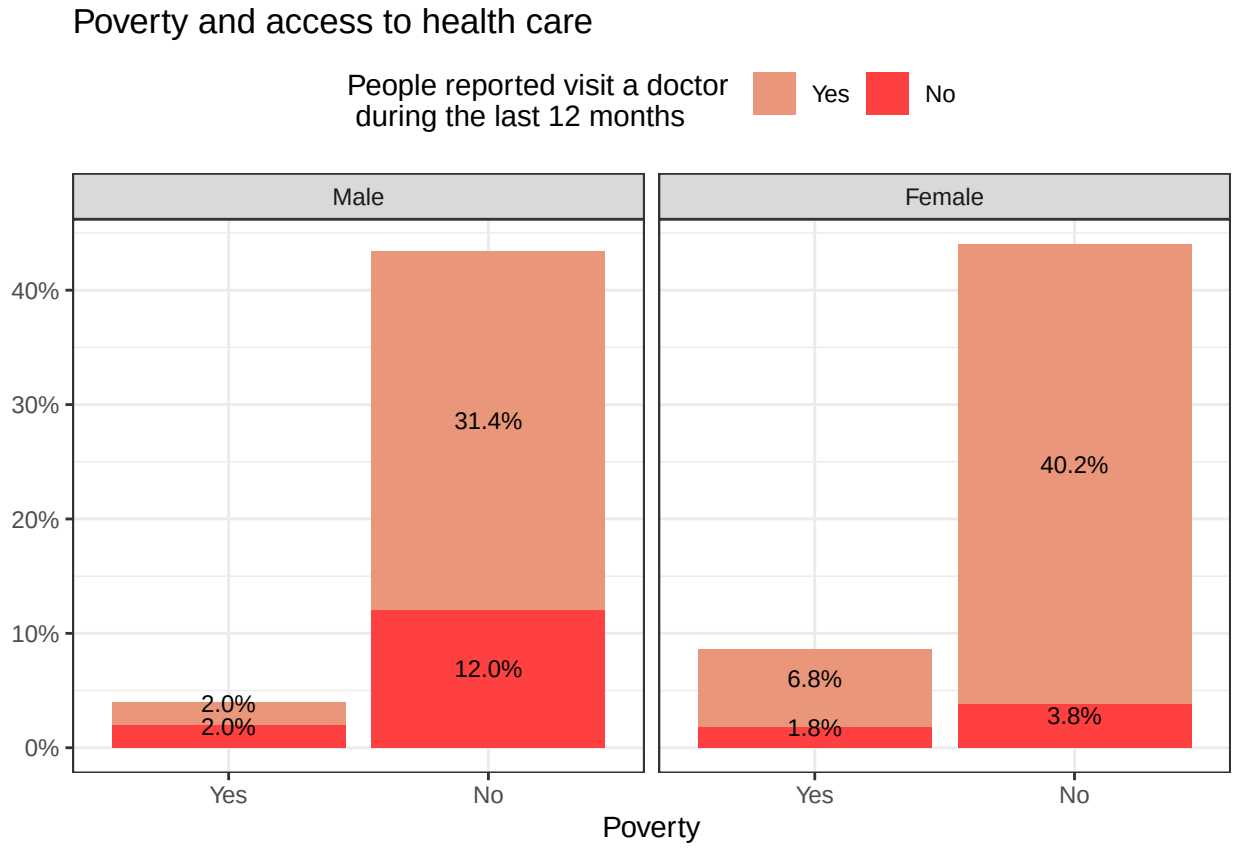
Muestro las etiquetas de valores

```
ggplot(HD, aes(poverty, fill = doctor)) +
  geom_bar(stat = "count") +
  stat_count(geom = "text", colour = "white", size = 3.5,
             aes(label = ..count..),
             position=position_stack(vjust=0.5)) +
  ggtitle("Poverty and access to health care") +
  xlab("Poverty") +
  ylab(NULL)
```

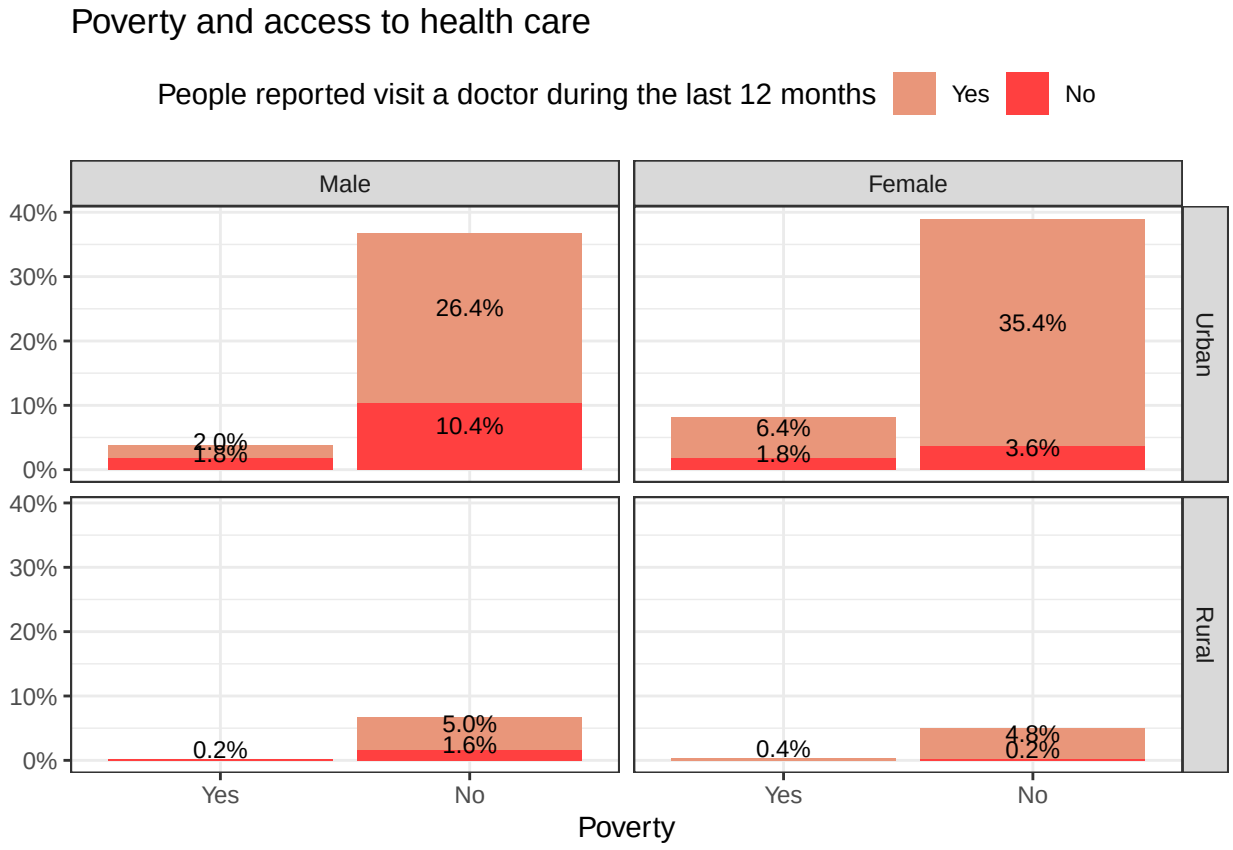


Finalizamos con la función facet_, la cual nos permite construir un panel de gráficos por categorías de otras variables.

```
ggplot(HD, aes(x = poverty,
               y = prop.table(stat(count)),
               fill = doctor,
               label = scales::percent(prop.table(stat(count))))) +
  geom_bar(aes(y = (..count..)/sum(..count..))) +
  geom_text(stat = 'count',
            position = position_stack(.5),
            vjust = -0.1,
            size = 3) +
  scale_y_continuous(labels = scales::percent) +
  ggtitle("Poverty and access to health care") +
  xlab("Poverty") +
  ylab(NULL) +
  scale_fill_manual(legend_title,
                    values=c("darksalmon","brown1")) +
  theme_bw() +
  theme(legend.position = "top")+
  facet_grid(~gender)
```




```
legend_title <- c("People reported visit a doctor during the last 12 months")

ggplot(HD, aes(x = poverty,
               y = prop.table(stat(count)),
               fill = doctor,
               label = scales::percent(prop.table(stat(count))))) +
  geom_bar(aes(y = (..count..)/sum(..count..))) +
  geom_text(stat = 'count',
            position = position_stack(.5),
            vjust = -0.1,
            size = 3) +
  scale_y_continuous(labels = scales::percent) +
  ggtitle("Poverty and access to health care") +
  xlab("Poverty") +
  ylab(NULL) +
  scale_fill_manual(legend_title,
                    values=c("darksalmon","brown1")) +
  theme_bw() +
  theme(legend.position = "top") +
  facet_grid(urban~gender)
```

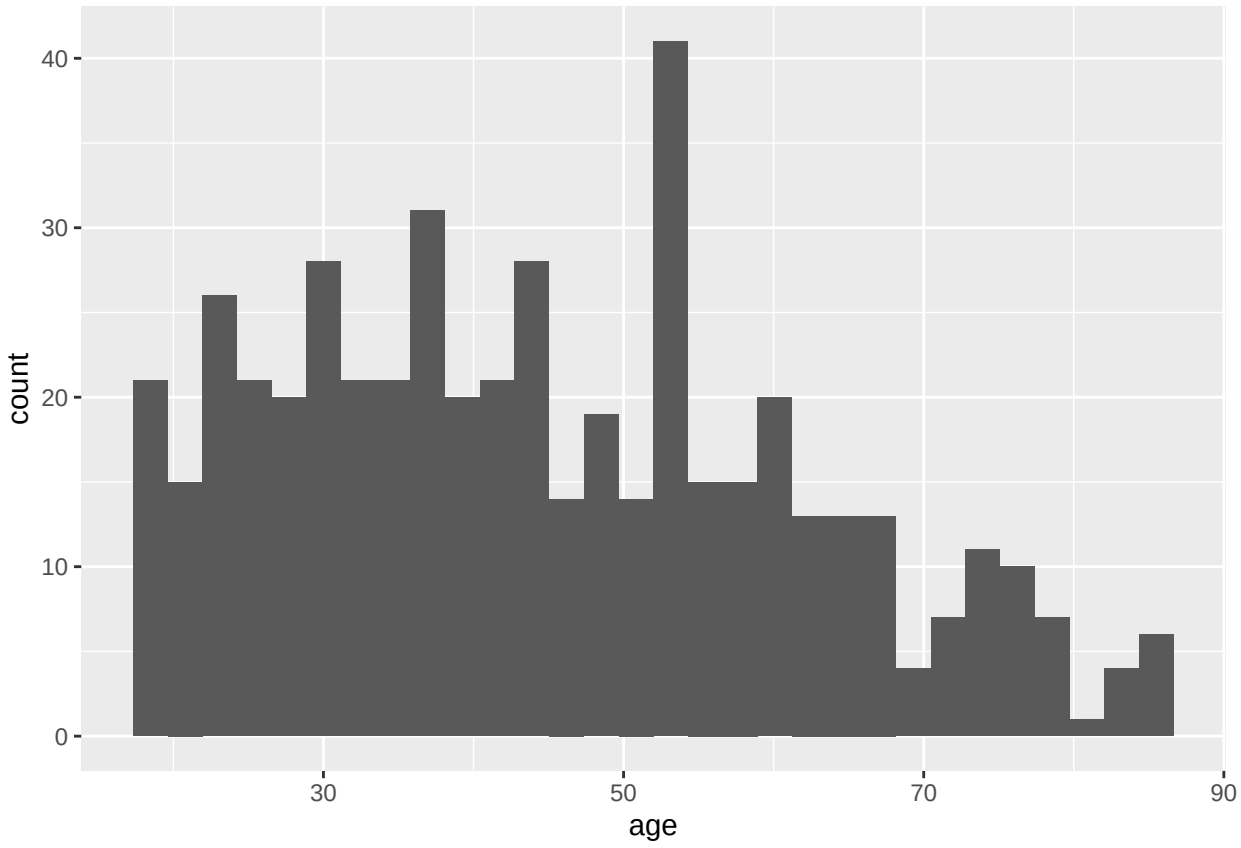


Ejercicio propuesto:

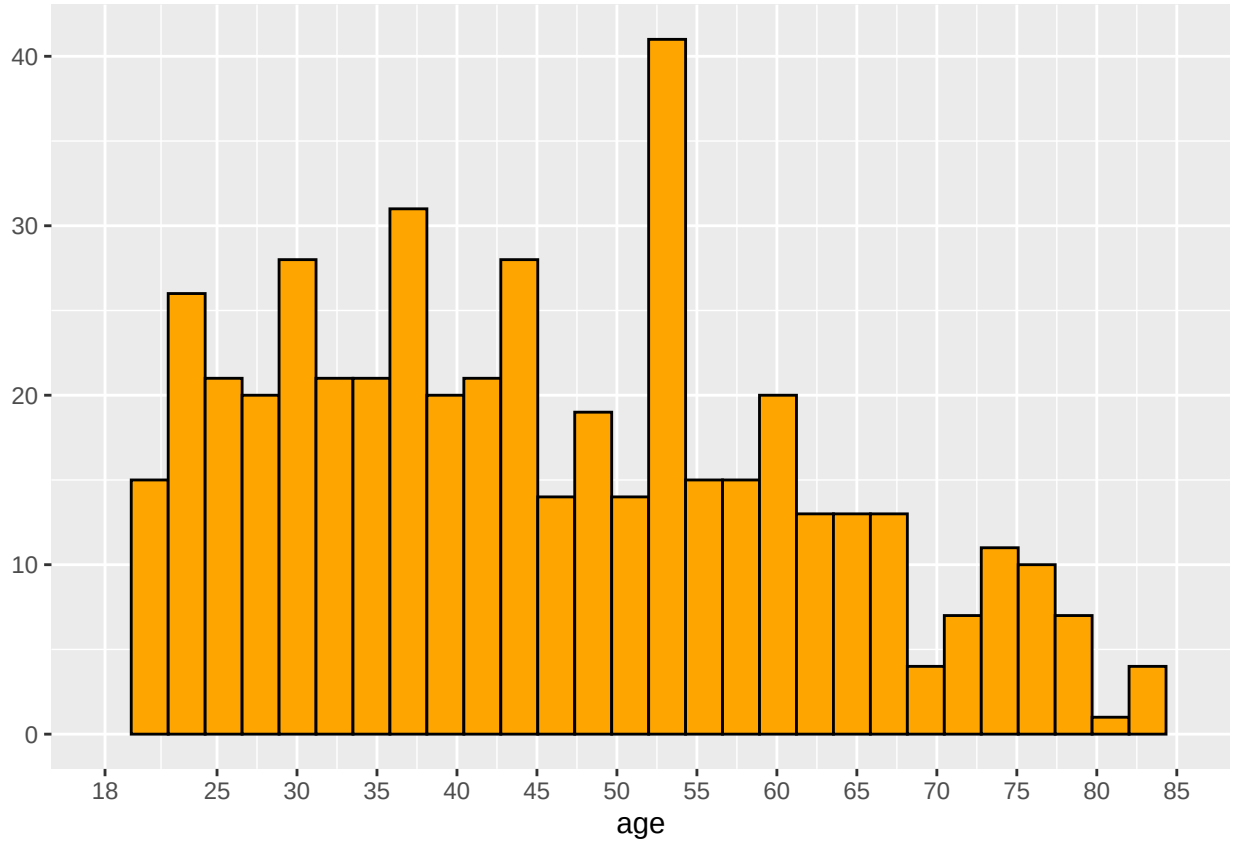
Con la misma base de datos, construye un gráfico que muestre las diferencias en visitas al médico (Yes/No) en función de la variable "racecat" y por género.

Histogramas de frecuencias con geom_histogram(...)

```
ggplot(HD, aes(x = age)) +
  geom_histogram()
```

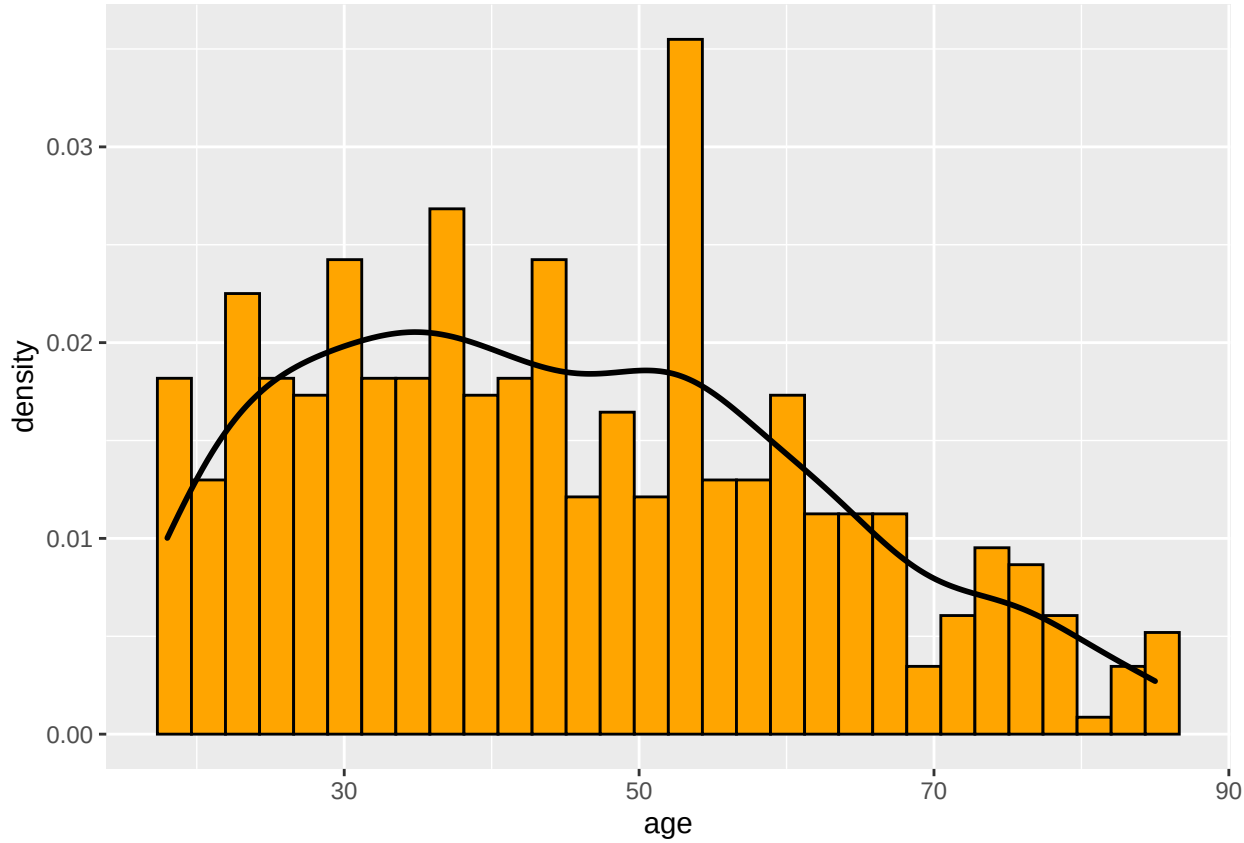



```
# Indicando unos límites concreto en el eje x
ggplot(HD, aes(x = age)) +
geom_histogram(fill="orange", colour="black") +
  scale_x_continuous(limits = c(18, 85),
                     breaks = c(18, seq(25,85,5))) +
  ylab(NULL)
```



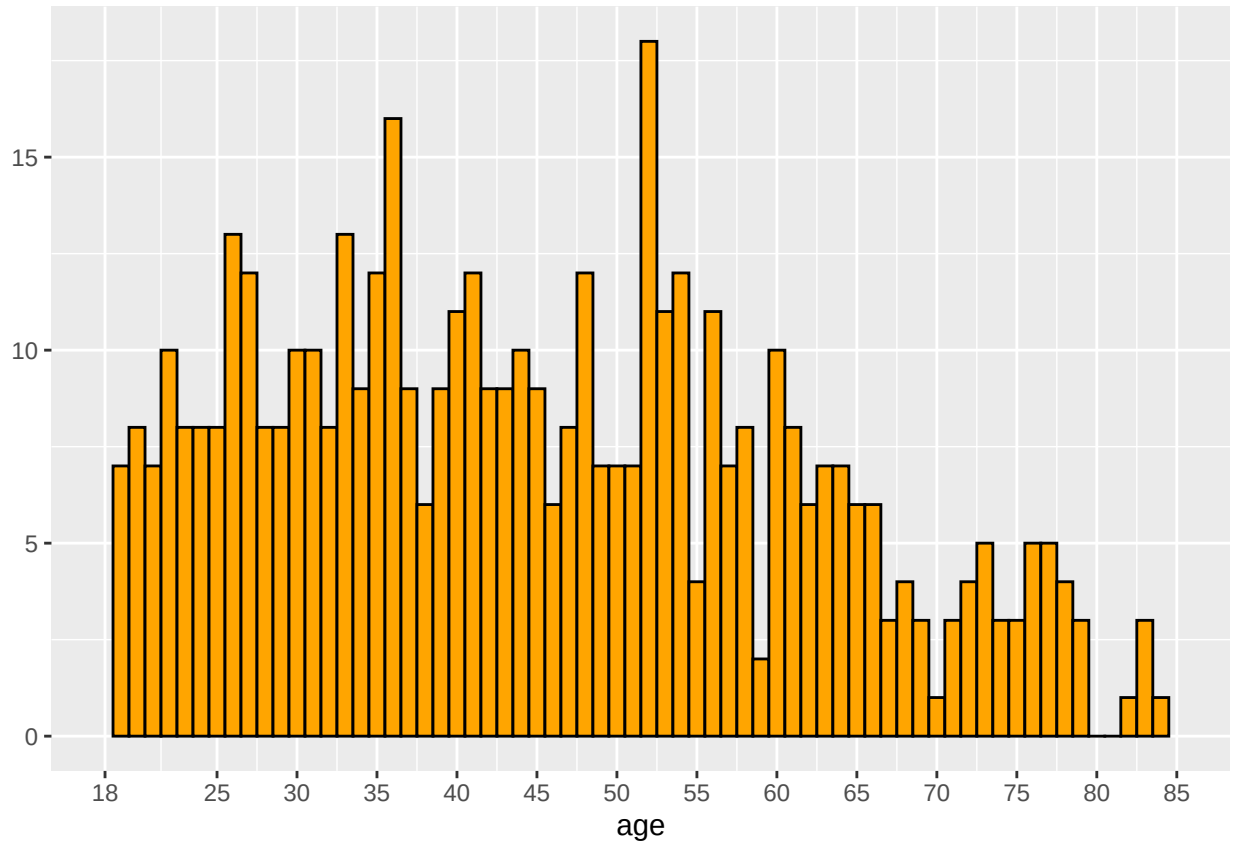
Al histograma le podemos ajustar una función de densidad de kernel

```
ggplot(HD, aes(x = age)) +
  geom_histogram(aes(y = ..density..),
                 colour = "black", fill = "orange") +
  geom_density(lwd = 1,
               linetype = 1,
               colour = "black")
```

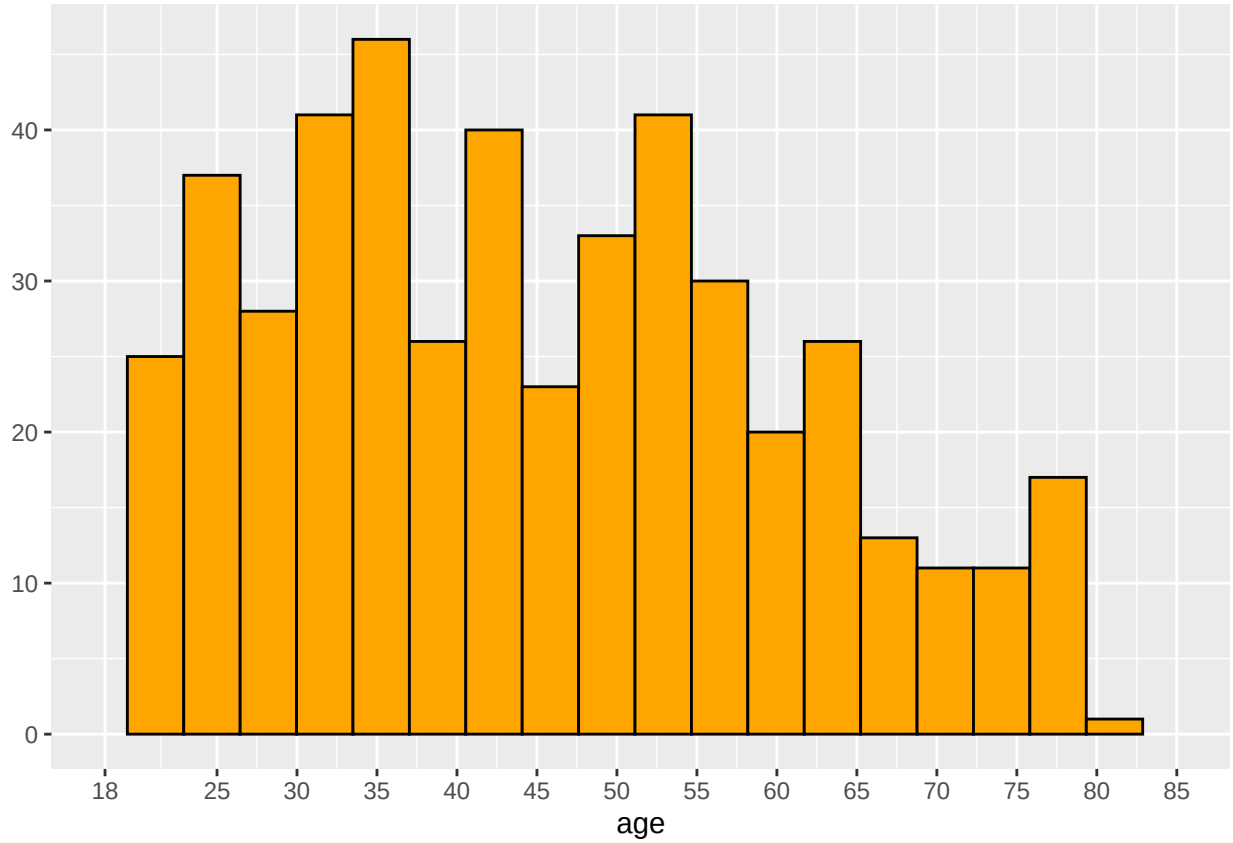


También puedo modificar el ancho o el número de intervalos

```
ggplot(HD, aes(x = age)) +
  geom_histogram(binwidth = 1, fill="orange", colour="black") +
scale_x_continuous(limits = c(18, 85),
                   breaks = c(18, seq(25,85,5))) +
  ylab(NULL)
```




```
ggplot(HD, aes(x = age)) +
  geom_histogram(bins = 20, fill="orange", colour="black") +
  scale_x_continuous(limits = c(18, 85),
                     breaks = c(18, seq(25,85,5))) +
  ylab(NULL)
```

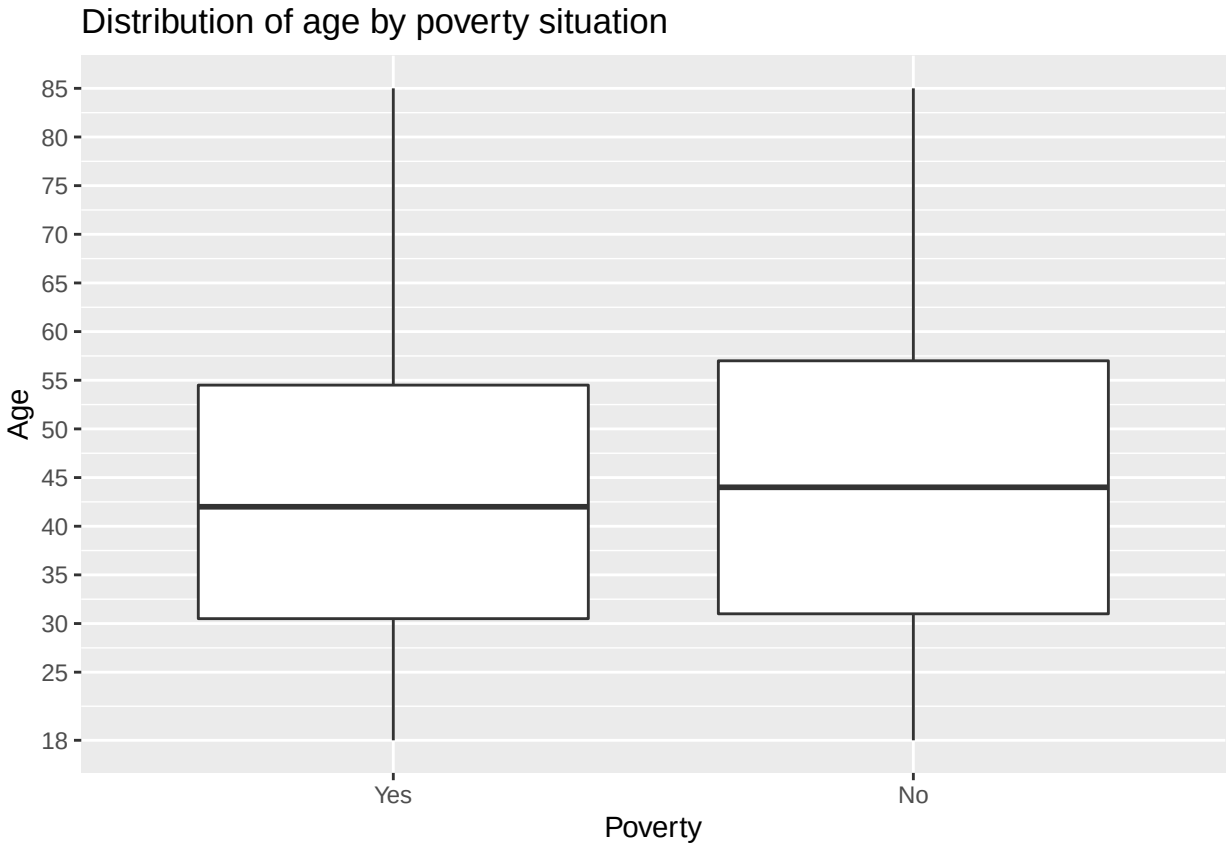


Ejercicio propuesto: Comprueba gráficamente si la distribución de la edad es igual en los individuos pobres y no pobres.

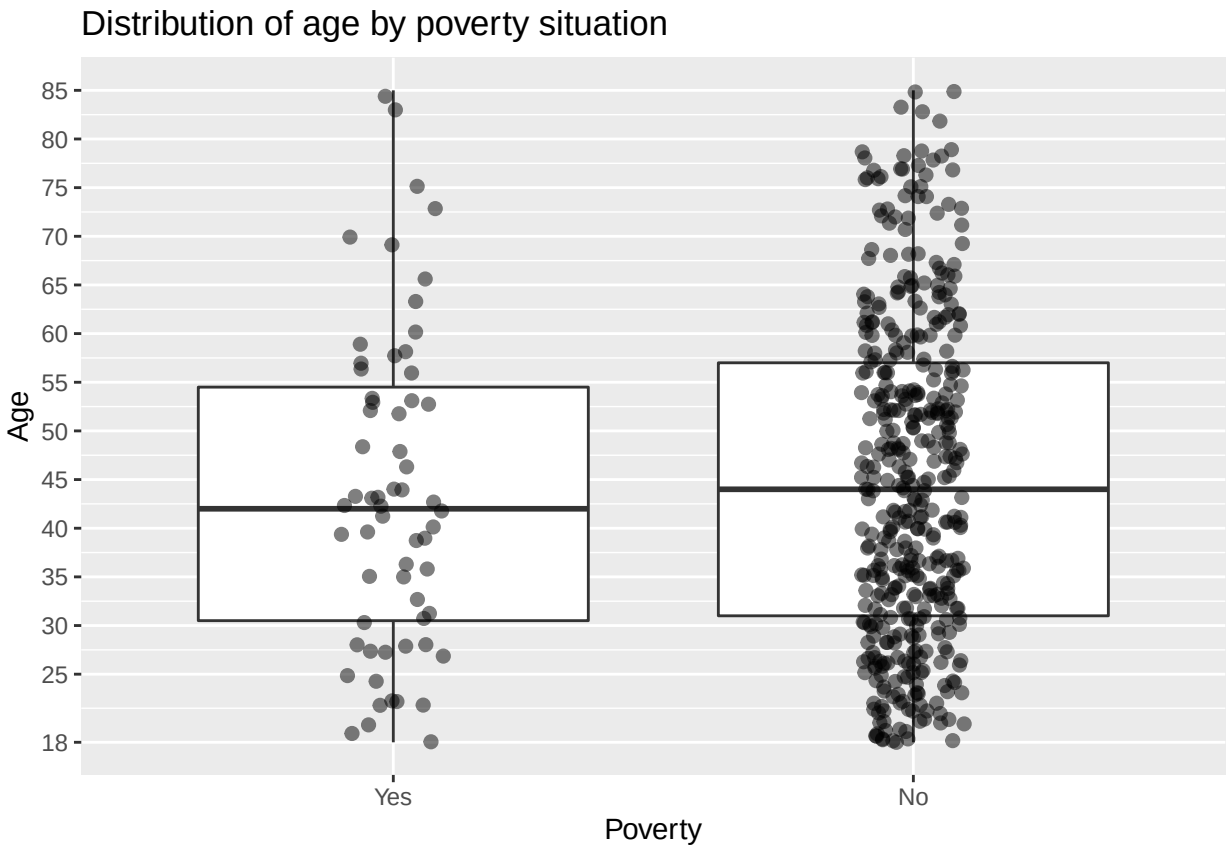
Diagrama de caja y bigotes, geom_boxplot(...)

Gráfico utilizado para representar gráficamente la distribución de una variable cuantitativa continua a través de sus cuartiles. Muy útil también para identificar outliers. Seguimos trabajando con la variable age del data frame "healthdisparities.dta". Dibujamos el box-plot sin y con los puntos (jitter).

```
ggplot(HD, aes(x = poverty, y = age )) +
  geom_boxplot() +
  scale_y_continuous(limits = c(18, 85),
                     breaks = c(18, seq(25,85,5))) +
  ggtitle("Distribution of age by poverty situation") +
  ylab("Age") +
  xlab("Poverty")
```

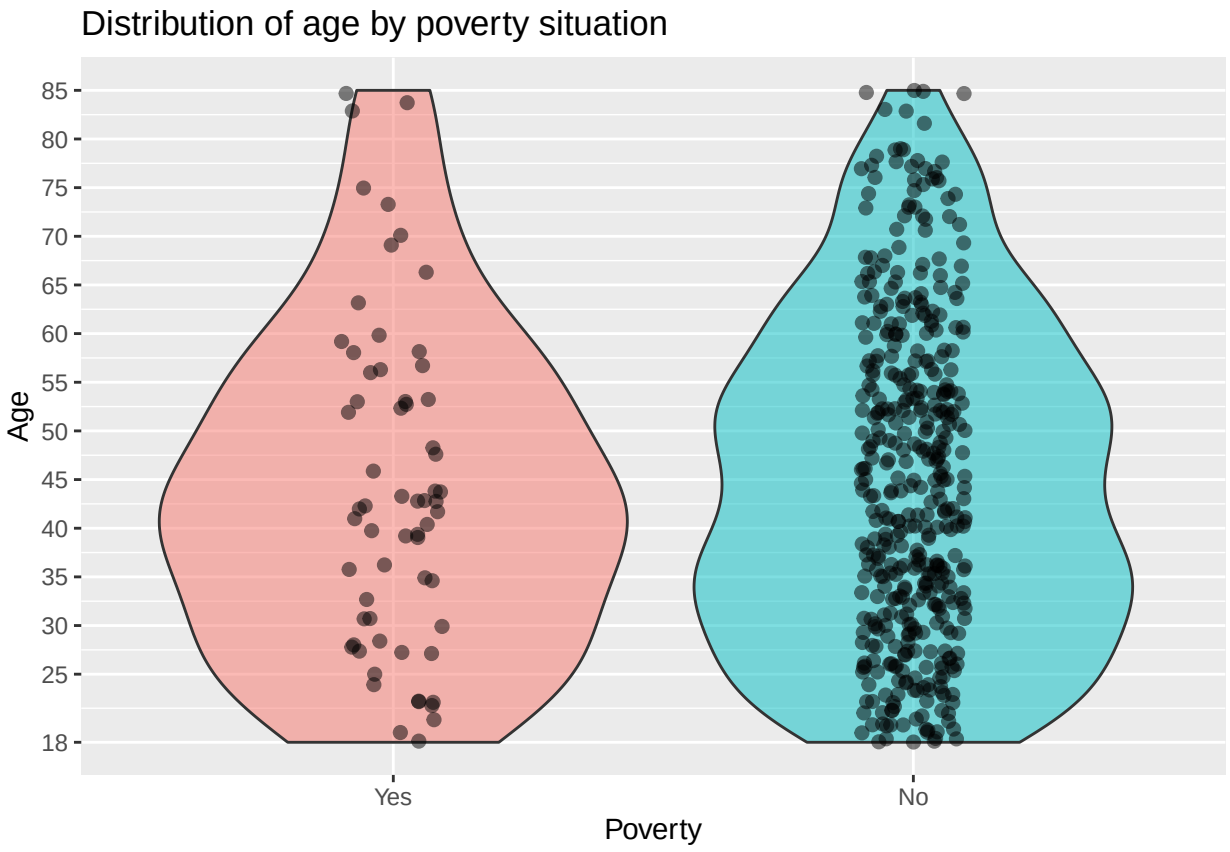



```
ggplot(HD, aes(x = poverty, y = age )) +
  geom_boxplot() +
  scale_y_continuous(limits = c(18, 85),
                     breaks = c(18, seq(25,85,5))) +
  ggtitle("Distribution of age by poverty situation") +
  ylab("Age") +
  xlab("Poverty") +
  geom_jitter(aes(x = poverty, y = age),
              size = 2,
              alpha = 0.5,
              width = 0.1)
```



Una altenativa muy interesante al box-plot son los gráficos violín. Los gráficos box-plot, debido a su simplicidad visual, tienden a ocultar detalles significativos sobre cómo se distribuyen los valores en los datos. Por ejemplo, no puedes ver si la distribución es bimodal o multimodal. Los gráficos violín son una buena alternativa.

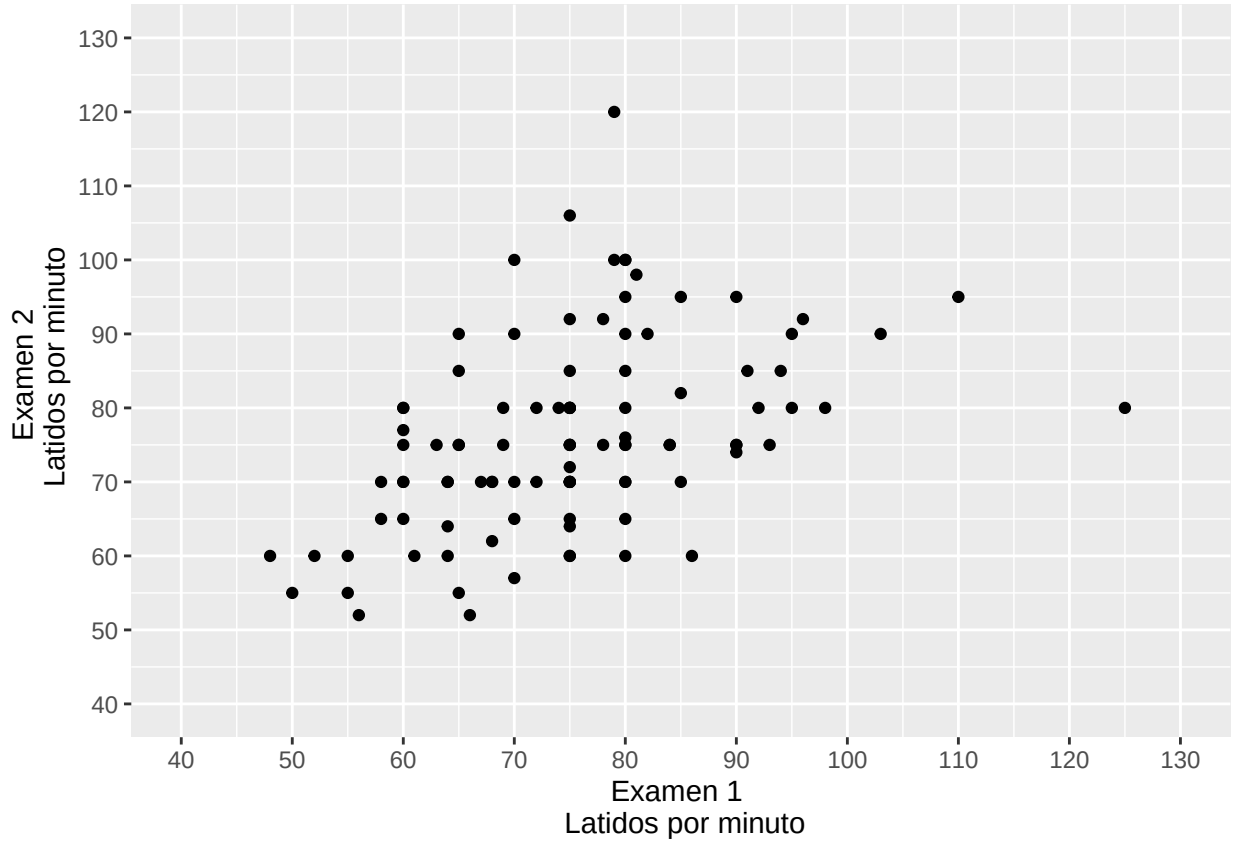
```
ggplot(HD, aes(x = poverty, y = age )) +
  geom_violin( aes(x = poverty, y = age ,fill=poverty), alpha=0.5) +
  scale_y_continuous(limits = c(18, 85),
                     breaks = c(18, seq(25,85,5))) +
  ggtitle("Distribution of age by poverty situation") +
  ylab("Age") +
  xlab("Poverty") +
  geom_jitter(aes(x = poverty, y = age),
              size = 2,
              alpha = 0.5,
              width = 0.1) +
  theme(legend.position = "none")
```



Scatter plot o diagrama de dispersión, geom_point(...)

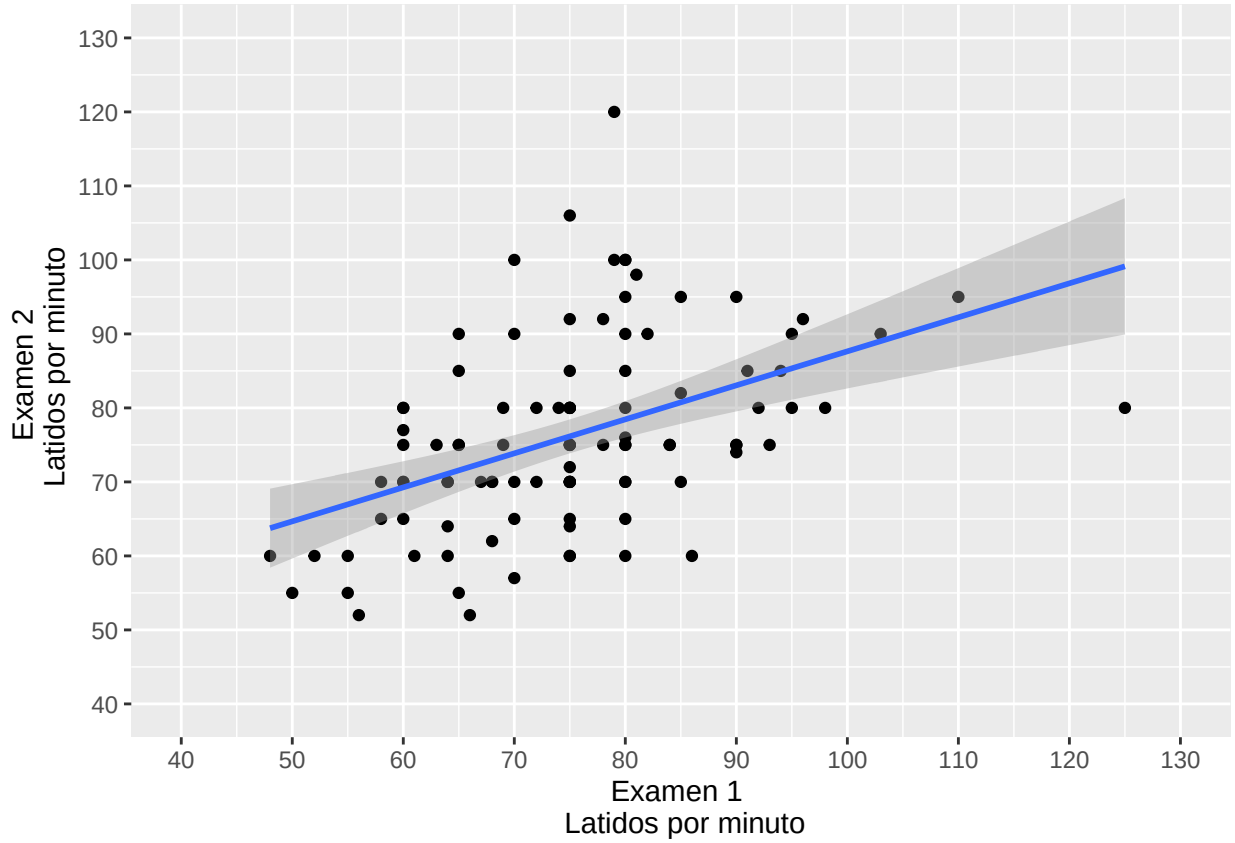
Para ilustrar este tipo de gráficos trabajaremos con las variables del dataset "heartrate.dta". Comenzaremos graficando la relación entre las pulsaciones del corazón en el primer examen y las pulsaciones en el segundo examen.

```
library(haven)
HR <- read_dta("heartrate.dta")

ggplot(data = HR, aes(x = heartrte1, y = heartrte2)) +
  geom_point() +
  scale_x_continuous(limits = c(40, 130),
                     breaks = seq(40, 130,10)) +
  scale_y_continuous(limits = c(40, 130),
                     breaks = seq(40, 130,10)) +
  xlab("Examen 1 \n Latidos por minuto") +
  ylab("Examen 2 \n Latidos por minuto")
```

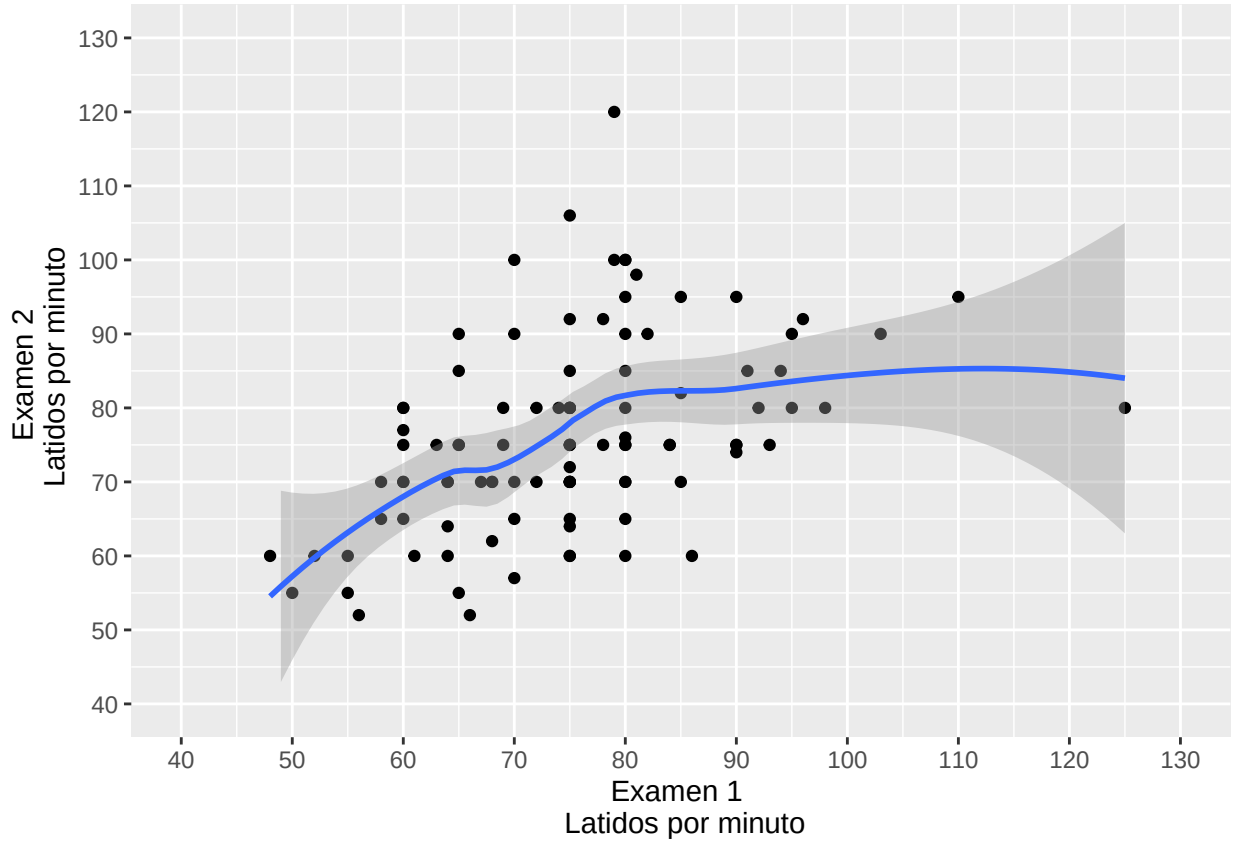


Con la función geom_smooth(...) podemos dibujar el ajuste a nuestra nube de puntos mediante métodos lineales y no-lineales.

```
ggplot(data = HR, aes(x = heartrte1, y = heartrte2)) +
  geom_point() +
  geom_smooth(method = lm, se = TRUE) +
  scale_x_continuous(limits = c(40, 130),
                     breaks = seq(40, 130,10)) +
  scale_y_continuous(limits = c(40, 130),
                     breaks = seq(40, 130,10)) +
  xlab("Examen 1 \n Latidos por minuto") +
  ylab("Examen 2 \n Latidos por minuto")
```

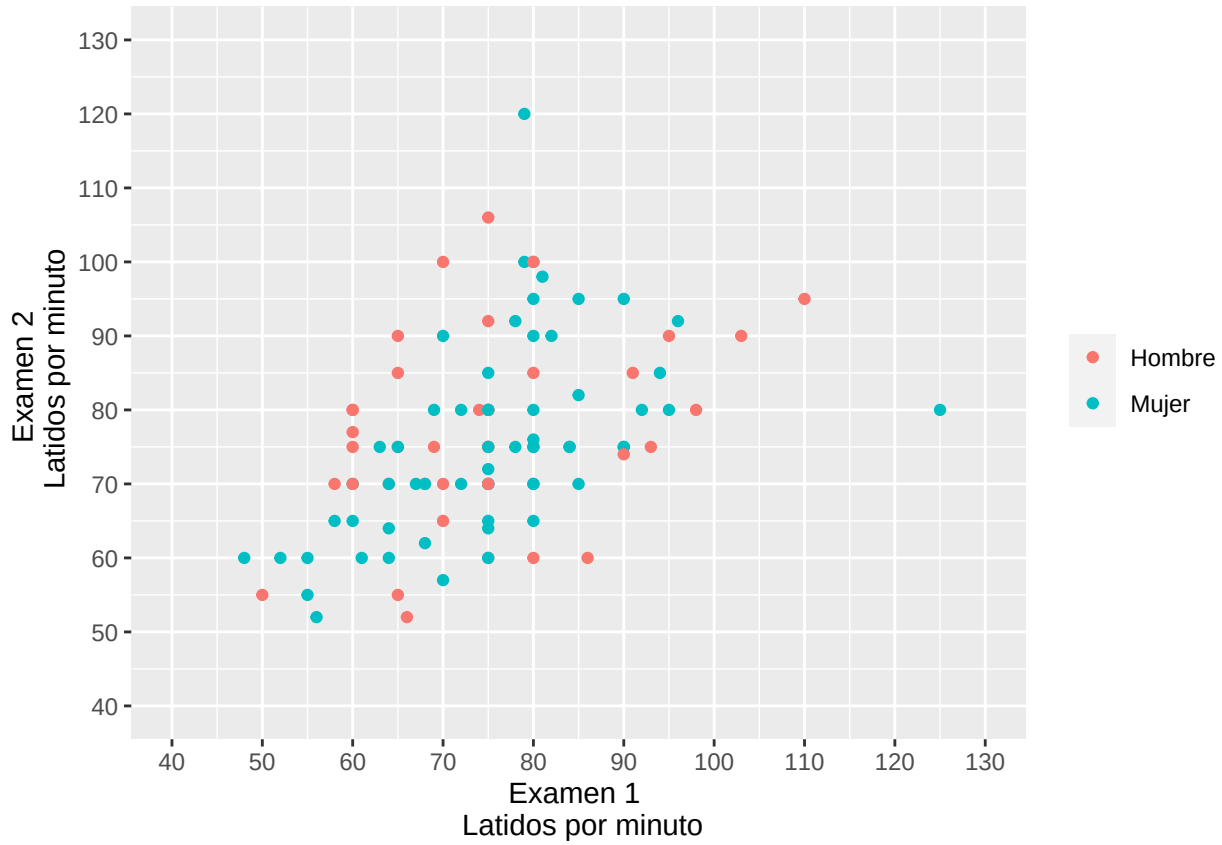



```
ggplot(data = HR, aes(x = heartrte1, y = heartrte2)) +
  geom_point() +
  geom_smooth(method = loess, se = TRUE) +
  scale_x_continuous(limits = c(40, 130),
                     breaks = seq(40, 130,10)) +
  scale_y_continuous(limits = c(40, 130),
                     breaks = seq(40, 130,10)) +
  xlab("Examen 1 \n Latidos por minuto") +
  ylab("Examen 2 \n Latidos por minuto")
```

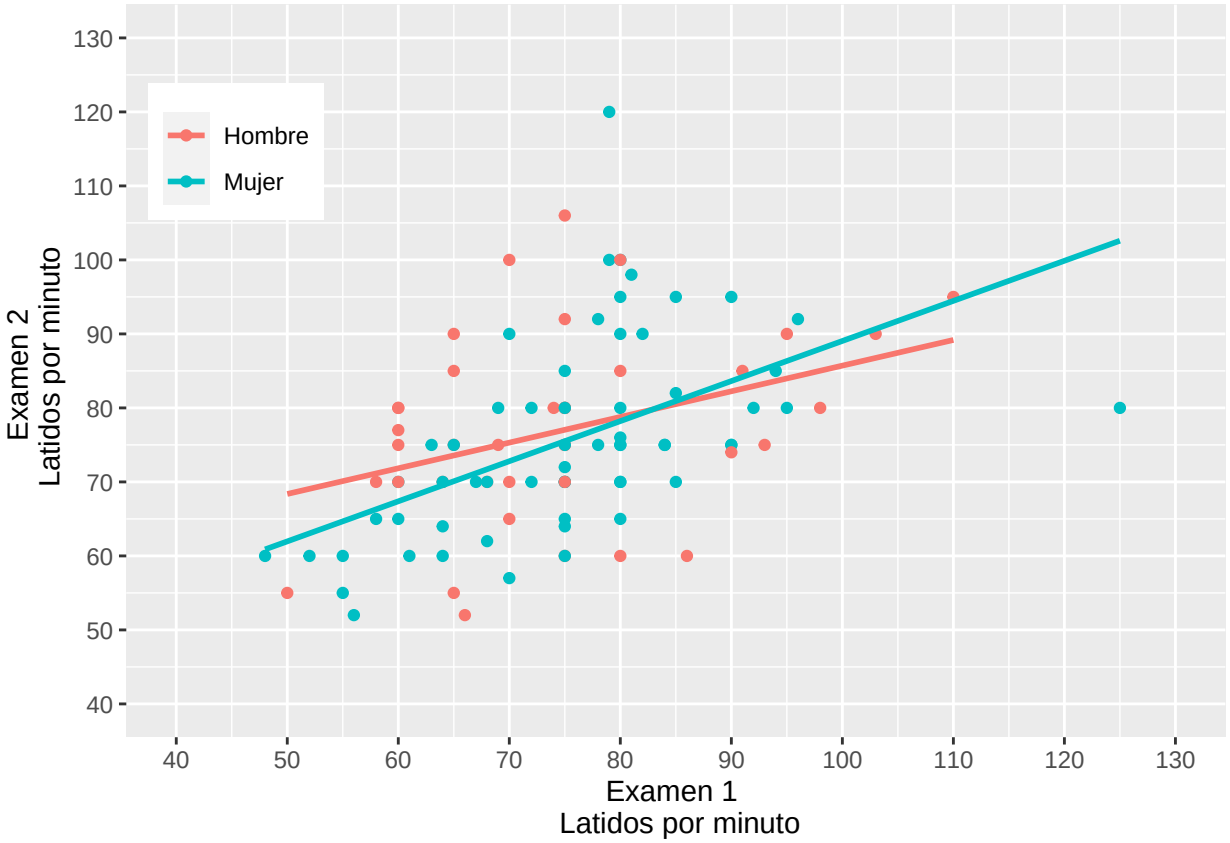


Podemo incorporar a nuestro gráfico una o varias variables categóricas, marcando con colores o figuras las diferentes categorías. En este ejemplo incoporamos la variable sex, previamente etiquetada como 1 Hombre , 2 Mujer.

```
HR$sex <- factor(HR$sex,
                    levels = c(1,2),
                    labels = c("Hombre", "Mujer"))

ggplot(data = HR, aes(x = heartrte1, y = heartrte2, color = sex)) +
  geom_point() +
  scale_x_continuous(limits = c(40, 130),
                     breaks = seq(40, 130,10)) +
  scale_y_continuous(limits = c(40, 130),
                     breaks = seq(40, 130,10)) +
  xlab("Examen 1 \n Latidos por minuto") +
  ylab("Examen 2 \n Latidos por minuto") +
  theme(legend.title = element_blank())
```




```
# Con ajuste lineal
ggplot(HR, aes(heartrte1, heartrte2, colour = sex)) +
  geom_point() +
  geom_smooth(se = FALSE, method = lm) +
  scale_x_continuous(limits = c(40, 130),
                     breaks = seq(40, 130,10)) +
  scale_y_continuous(limits = c(40, 130),
                     breaks = seq(40, 130,10)) +
  xlab("Examen 1 \n Latidos por minuto") +
  ylab("Examen 2 \n Latidos por minuto") +
  theme(legend.title = element_blank(),
        legend.position = c(0.1, 0.8))
```



Ejercicio 1:

Utiliza el análisis gráfico con ggplot2 para adelantar los resultados esperados de este ejercicio del Módulo2.

El dataset hiaa.dta (en formato Stata) incluye mediciones del examen A5-HIA en la orina en 40 pacientes, con el objeto de medir la cantidad de ácido 5-hidroxindolacético (A5-HIA). Este ácido es un producto de degradación de la serotonina.

Ejercicio 2:

Utiliza el análisis gráfico con ggplot2 para analizar las relaciones entre las variables del fichero “Medidas.xlsx”.