Práctica 3. Estadística descriptiva en R.

Pedro M. Carmona Sáez, Christian J. Acal González,
Miguel Ángel Montero Alonso y Grupo BioestadísticaR

UNIVERSIDAD
DE GRANADA

Todo el material para el conjunto de actividades de este curso ha sido elaborado y es propiedad intelectual del grupo **BioestadisticaR** formado por:
Juan de Dios Luna del Castillo,
Pedro Femia Marzo,
Miguel Ángel Montero Alonso,
Christian José Acal González,
Pedro María Carmona Sáez,
Juan Manuel Melchor Rodríguez,
José Luis Romero Béjar,
Manuela Expósito Ruíz,
Juan Antonio Villatoro García,
Juan Manuel Praena Fernández,
Miguel Ángel Luque Fernández,
Francisco Javier Arnedo Fernández.

Todos los integrantes del grupo han participado en todas las actividades, en su elección, construcción, correcciones o en su edición final, no obstante, en cada una de ellas, aparecerán uno o más nombres correspondientes a las personas que han tenido la máxima responsabilidad de su elaboración junto al grupo de **BioestadisticaR**.

Práctica 3. Estadística descriptiva en R

Pedro M. Carmona Sáez, Christian J. Acal González y Miguel Ángel Montero Alonso

Los objetivos de este tema son los métodos relacionados con la estadística descriptiva: tablas de frecuencias, gráficos y resumen de datos.

3.1 Distribuciones de frecuencias en R

Las tablas de frecuencias son una herramienta muy utilizada en estadística para apreciar cómo están distribuidas las características de la población. Se utilizan para resumir la información de las observaciones de una variable/característica con el fin de obtener las primeras conclusiones acerca de ella. Una tabla de frecuencias suele estar formada por cuatro componentes: frecuencia absoluta, frecuencia absoluta acumulada, frecuencia relativa y frecuencia relativa acumulada. R permite obtenerlas de una forma muy rápida:

- **Frecuencia absoluta**. Se extrae mediante la función **table()**. Dentro de los paréntesis se coloca la variable que se desee analizar. Esta función devuelve una tabla que representa el número total de individuos que presenta cada modalidad de la variable
- **Frecuencia absoluta acumulada**. Es obtenida usando la función **cumsum(x)** donde x es un objeto de tipo *table*. En este caso, o se guarda en un objeto el resultado de aplicar la función **table()** y posteriormente se ejecuta **cumsum()**, o bien, se ordena directamente **cumsum(table())**. Esto muestra una tabla que representa el número total de individuos cuya modalidad es menor o igual a ella. Esta medida no tiene sentido extraerla para variables cualitativas nominales.
- **Frecuencia relativa**. Se consigue empleando la función **prop.table(x)** donde x es un objeto de tipo *table*. Asimismo, o se guarda en un objeto el resultado de aplicar la función **table()** y posteriormente se ejecuta **prop.table()**, o bien, se ordena directamente **prop.table(table())**. Esto muestra una tabla que representa el porcentaje de individuos que presenta cada modalidad.
- **Frecuencia relativa acumulada**. Se alcanza mediante la orden **cumsum(prop.table(table()))**. Se puede ir ejecutando cada función de una en una e ir guardando el resultado en un objeto o aplicar las tres al mismo tiempo. Esto devuelve una tabla que representa el porcentaje de individuos cuya modalidad es menor o igual a ella.

Cada uno de las opciones comentadas anteriormente, mostrará un resultado por separado en la *consola* de R. Si se quieren mostrar todos los resultados en un mismo objeto, se puede utilizar la función **cbind()**, de tal forma que R fusione todos los objetos creados por separado en uno solo. En los siguientes comandos se muestra un caso de tabla de frecuencias para la variable *grupo_edad* agrupada en tres intervalos del fichero **osteo.sav**. Se actuaría de la misma forma para cualquier tipo de variable.

```
## re-encoding from UTF-8
```

```
#Creamos un data frame a partir de los datos contenidos en el fichero osteo
library(foreign)
osteo=read.spss("osteo.sav",to.data.frame = TRUE)
attach(osteo)
```

```
frec=table(grupo_edad)
frec
```

```
## grupo_edad
##    < 25 25 - 33    > 33
##      32      32      30
```

```
frec.acum=cumsum(table(grupo_edad))
frec.acum=cumsum(frec) #Otra opción si se han calculado las frecuencias absolutas
frec.acum
```

```
##    < 25 25 - 33    > 33
##      32      64      94
```

```
prop=prop.table(table(grupo_edad))
prop=prop.table(frec) #Otra opción si se han calculado las frecuencias absolutas
prop
```

```
## grupo_edad
##      < 25   25 - 33      > 33
## 0.3404255 0.3404255 0.3191489
```

```
prop.acum=cumsum(prop.table(table(grupo_edad)))
prop.acum=cumsum(prop) #Otra opción si se han calculado las frecuencias relativas
prop.acum
```

```
##      < 25   25 - 33      > 33
## 0.3404255 0.6808511 1.0000000
```

```
tabla.frecuencias=cbind(frec,frec.acum,prop,prop.acum)
tabla.frecuencias
```

```
##         frec frec.acum      prop prop.acum
## < 25      32        32 0.3404255 0.3404255
## 25 - 33   32        64 0.3404255 0.6808511
## > 33      30        94 0.3191489 1.0000000
```

Del mismo modo que cualquier otro objeto en R, los resultados obtenidos en el ejemplo pueden ser manipulados. Si se aplica la función **class()** sobre ellos, se puede apreciar que los objetos 'frec', 'frec.acum', 'prop' y 'prop.acum' son de tipo *tabla*, mientras que el objeto 'tabla.frecuencias' es de tipo *matriz*. En consecuencia, el objeto 'tabla.frecuencias' puede ser manipulado de igual forma que un *dataframe*. Con respecto a los objetos de tipo *tabla*, su manejo es similar al de un vector pero siempre acompañado de un nombre. Por ejemplo:

1. `prop[1]`. Extrae el primer elemento del objeto que es 0.3404255. La diferencia con los vectores radica que en este objeto los elementos que los conforman vienen acompañados de un nombre, en este caso < 25.

```
prop[1]
```

```
##      < 25
## 0.3404255
```

2. `prop[1]+1`. Se pone de manifiesto que con los elementos con los que se trabaja son con los números y no con los nombres, ya que 1 más el valor del primer elemento que era 0.3404255 resulta ser 1.3404255.

```
prop[1]+1
```

```
##     < 25
## 1.340426
```

3. `round(prop,2)`. Se pueden realizar las mismas operaciones que en un vector pero los resultados siempre vendrán acompañados del nombre en cuestión.

```
round(prop,2)
```

```
## grupo_edad
##    < 25 25 - 33    > 33
##    0.34    0.34    0.32
```

4. `sort(prop,decreasing = FALSE)`. Siguiendo el hilo del punto anterior, se permite ordenar los valores de la tabla, de manera que el nombre que le sigue a cada elemento cambie automáticamente de posición quedando siempre encuadrado con su elemento correspondiente.

```
sort(prop,decreasing = FALSE)
```

```
## grupo_edad
##      > 33      < 25   25 - 33
## 0.3191489 0.3404255 0.3404255
```

5. `names(prop)`. Se puede acceder a los nombres de la tabla.

```
names(prop)
```

```
## [1] "< 25"    "25 - 33" "> 33"
```

6. `names(sort(prop,decreasing = T))[1]`. Obtiene el valor modal. Tener sumo cuidado en que la orden *decreasing* debe ser igual a *TRUE* para ordenar los valores de mayor a menor. En lugar de usar el objeto 'prop' que denota las frecuencias relativas se podría haber usado el objeto 'frec' que corresponde con las frecuencias absolutas.

```
names(sort(prop,decreasing = T))[1]
```

```
## [1] "< 25"
```

Otra posibilidad para extraer la tabla de frecuencias es a través de la función **freq()** que viene implementada en el paquete **summarytools**. Este paquete hay que instalarlo porque no viene integrado en R por defecto. La forma de la tabla con la función freq puede considerarse más visual, tiene en cuenta los valores perdidos y calcula la frecuencia relativa y la frecuencia relativa acumulada tanto considerando los valores perdidos (*% Total* y *% Total Cum.*) como ignorándolos (*% Valid* y *% Valid Cum.*). Si no hay presencia de datos faltantes, las columnas *% Valid* y *% Total* serán iguales y lo mismo sucederá con *% Valid Cum.* y *% Total Cum.* Por contra, esta función no calcula la frecuencia absoluta acumulada.

```
library(summarytools)
freq(grupo_edad)
```

```
## Frequencies
## grupo_edad
## Type: Factor
##
##                 Freq   % Valid   % Valid Cum.   % Total   % Total Cum.
## ------------- ------ --------- -------------- --------- --------------
##          < 25     32     34.04          34.04     34.04          34.04
##       25 - 33     32     34.04          68.09     34.04          68.09
##          > 33     30     31.91         100.00     31.91         100.00
##          <NA>      0                               0.00         100.00
##         Total     94    100.00         100.00    100.00         100.00
```

```
#Ejemplo para mostrar el funcionamiento con valores perdidos
prueba=factor(c("1","1","2","2","1",NA,"3","4","2","4"))
freq(prueba)
```

```
## Frequencies
## prueba
## Type: Factor
##
##               Freq   % Valid   % Valid Cum.   % Total   % Total Cum.
## ----------- ------ --------- -------------- --------- --------------
##           1      3     33.33          33.33     30.00          30.00
##           2      3     33.33          66.67     30.00          60.00
##           3      1     11.11          77.78     10.00          70.00
##           4      2     22.22         100.00     20.00          90.00
##        <NA>      1                               10.00         100.00
##       Total     10    100.00         100.00    100.00         100.00
```

3.2 Síntesis de datos

En cualquier estudio estadístico es interesante utilizar medidas de resumen o descriptivas. Estas medidas exploran los datos a fin de identificar sus principales características. Solo pueden ser utilizadas con variables numéricas.

Tenemos de dos tipos:

- De Posición: describen cómo se encuentra el resto de la muestra con respecto a ella (como la mediana, que indica que en la muestra hay tantos individuos por debajo como por encima de ella).
- De Dispersión: describen cómo de variables o dispersos son los datos (la muestra 3, 4, 5 es menos dispersa que la muestra 1, 4, 7).

3.2.1 Medidas de Posición

Entre ellas tenemos las medidas de tendencia central como:

- **Media aritmética**. Se define como el cociente que se obtiene al dividir la suma de los valores de la variable por el número de observaciones. En R se obtiene mediante la función **mean()**.
- **Mediana**. Se define como el número que divide a la muestra (ordenada de menor a mayor) en dos partes iguales. En R se obtiene mediante la función **median()**.
- **Moda**. La clase cuya frecuencia alcanza un máximo absoluto (si la cualidad es nominal) o relativo (en otro caso). El cálculo de la moda puede hacerse con la función implementada en el paquete **modeest**

y las de tendencia no central como los cuantiles:

- **Cuantiles**. Cumplen con la condición de superar al menos un cierto porcentaje de los datos y de ser superado a lo sumo por el porcentaje restante de las observaciones, supuesto que están ordenadas por valor creciente del carácter. Los cuantiles se dividen en tres grandes bloques: cuartiles (dividen a la población en cuatro partes iguales), deciles (dividen a la población en diez partes iguales) y percentiles (dividen a la población en cien partes iguales). En R se calculan usando la función **quantile(x,p)**, siendo 'x' la variable y 'p' el percentil que se esté deseando determinar. Dado que Cuartiles y Deciles son un caso particular de Percentiles (por ejemplo, el cuartil 2 coincide con el decil 5 que a su vez es igual que el percentil 50), basta utilizar la función quantile en el que se indica la variable en estudio y el porcentaje que queda acumulado por debajo. Si no se indica el valor de 'p' en la función, R devuelve un vector con los cuartiles.

```
library(modeest)
mfv(peso) #Indica el o los valores con más frecuencia
```

```
## [1] 62
```

```
mean(peso)  #media de los valores de peso
```

```
## [1] 63.83936
```

```
median(peso)  #mediana de los valores de peso
```

```
## [1] 62
```

```
quantile(peso)   #Cuartiles
```

```
##    0%   25%   50%   75%  100%
## 44.60 56.25 62.00 69.75 99.00
```

```
quantile(peso,0.90)   #Percentil 90
```

```
##   90%
## 82.38
```

3.2.2 Medidas de dispersión

Nos sirven para cuantificar la variabilidad de los datos informando acerca de la mayor o menor representatividad de las medidas de tendencia central.

- **Recorrido, Rango o Amplitud (A)**. Es la diferencia entre los valores más grande y más pequeño de la muestra. Se calcula usando las funciones **min()** y **max()**
- **Varianza**. Es la media aritmética de los cuadrados de las desviaciones. En R se determina a partir de la función **var()**. Sin embargo, por defecto, R calcula la varianza muestral. Por tanto, en R habría que realizar $var() * (n-1)/n$ para obtener la varianza que es dividida por 'n'.
- **Desviación típica**. Se define como la raíz cuadrada de la varianza. Por defecto R calcula con la función **sd()** la desviación típica muestral, es decir, la raíz cuadrada de la varianza que es obtenida dividiendo por 'n-1' en lugar de por 'n'.
- **Coeficiente de variación**: se define como la desviación típica dividida entre la media. En R se puede calcular directamente dividiendo el resultado de la función **sd()** entre la de **mean()**.

```
var(peso)*(length(peso)-1)/length(peso) #Varianza
```

```
## [1] 137.8579
```

```
sd(peso)  #Desviación típica
```

```
## [1] 11.80425
```

```
max(peso)-min(peso) #rango
```

```
## [1] 54.4
```

```
sd(peso) / mean(peso) #Coeficiente de variación
```

```
## [1] 0.1849055
```

Existe una función, **summary()**, que devuelve un objeto de tipo *tabla* que contiene todas las medidas definidas anteriormente, salvo la varianza y la desviación típica.

```
n=length(peso)  #Longitud del vector. nº de observaciones
summary(peso)  #Resumen de las medidas
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   44.60   56.25   62.00   63.84   69.75   99.00
```

```
summary(peso)[4]  #Media a través de la función summary
```

```
##     Mean
## 63.83936
```

Al igual que sucedía con las tablas de frecuencias, todas las medidas descriptivas pueden ser obtenidas a través de la función **descr()** del paquete **summarytools**. En esta función hay que indicar la variable de interés y las medidas que se quieren calcular. Para más información acerca de esta función se invita consultar el CRAN del paquete summarytools y leer la descripción de la función descr.

```
library(summarytools)
descr(peso,stats=c("mean","sd","q1","med","q3","min","max","cv"))
```

```
## Descriptive Statistics
## peso
## N: 94
##
##                  peso
## ------------- -------
##          Mean   63.84
##       Std.Dev   11.80
##            Q1   56.00
##        Median   62.00
##            Q3   70.00
##           Min   44.60
##           Max   99.00
##            CV    0.18
```

```
medidas=descr(peso,stats=c("mean","sd","q1","med","q3","min","max","cv"))
medidas[1] #Media mediante la función descr
```

```
## [1] 63.83936
```

Finalmente, también es posible obtener las medidas de resumen distinguiendo por grupos. Hay situaciones en las que el estudio desemboca en separar los individuos según una o varias características. Un ejemplo claro puede ser separar a la población según el sexo, y estudiar el peso en cada grupo. Esto en R se puede realizar a través de la función **subset()** que fue vista en el apartado [1.1.1 Almacenamiento de datos en R mediante dataframes], creando dos subconjuntos de datos (uno para hombres y otro para mujeres) y calcular las medidas descriptivas que se quieran en cada grupo. Otra posibilidad es usar la función **ddply()** que viene implementada en el paquete **plyr** que habrá que instalar previamente. Con esta función se puede dividir el archivo en tantas variables de tipo *factor* como se desee, pero hay que tener sumo cuidado porque la forma de actuar cambia ligeramente si se introduce más de una variable factor. En las siguientes líneas de comandos, primero se calcula la media y la desviación típica del peso según el *sexo* y seguidamente, según el *sexo* y el *grupo de edad*. Recalcar que para el primer caso la variable *sexo* fue introducida mediante `~sexo`, y en el segundo caso las variables fueron añadidas con `.(sexo,grupo_edad)`. Además, esta función no requiere el uso del símbolo **$**.

```
library(plyr)
ddply(osteo,~sexo,summarise,media=mean(peso),desv.tip=sd(peso))
```

```
##     sexo    media desv.tip
## 1 Hombre 66.93111 10.84678
## 2  Mujer 61.00000 12.03877
```

```
ddply(osteo,.(sexo,grupo_edad),summarise,media=mean(peso),desv.tip=sd(peso))
```

```
##     sexo grupo_edad    media  desv.tip
## 1 Hombre       < 25 65.50000  7.390788
## 2 Hombre    25 - 33 63.20667 11.427315
## 3 Hombre       > 33 73.10000 12.031071
## 4  Mujer       < 25 63.45333  9.452201
## 5  Mujer    25 - 33 53.77647  9.769502
## 6  Mujer       > 33 66.05882 13.149242
```

3.3 Representaciones gráficas con R

Las representaciones gráficas son una herramienta estadística que se utiliza para proporcionar la misma información que las tablas de frecuencias, pero de manera visual con el objetivo de facilitar la interpretación de los datos y que complementan los resultados obtenidos a través de las medidas de resumen. Las más utilizadas son:

- **Diagrama de barras**. Sobre unos ejes cartesianos se representan, en el eje de abscisas, las modalidades de la variable y, sobre el eje de ordenadas, las frecuencias (relativas o absolutas). Sobre cada modalidad se levanta una barra de altura igual a la frecuencia correspondiente.
- **Diagrama de sectores**. Consiste en dividir un círculo en tantos sectores según el número de modalidades que tenga la variable. El área de cada sector tiene que ser proporcional a la frecuencia absoluta o relativa de cada modalidad.
- **Histograma**. Se construye mediante un conjunto de rectángulos yuxtapuestos cuyas bases son los diferentes intervalos de clase. La altura de cada rectángulo es proporcional a la frecuencia absoluta o relativa de cada intervalo.
- **Polígono de frecuencias**. Consiste en unir mediante una línea (conocida como poligonal) los centros de los techos de los rectángulos del histograma o diagrama de barras.
- **Gráfico de cajas y bigotes o Boxplot**. Este gráfico está formado por un rectángulo o caja, cuyos bordes inferior y superior corresponden al primer y tercer cuartil de la variable graficada, y una linea horizontal en su interior que representa la mediana de la variable. Los bigotes son segmentos que unen cada extremo de la caja con el último valor de la variable que esté a una distancia máxima de un rango intercuartílico (RIQ=Q3-Q1). Los valores más alejados de la distancia 1.5RIQ, se consideran datos atípicos, y se representan mediante puntos arriba o abajo de los bigotes de la caja. Serán atípicos extremos si superan 3 veces el RIQ.

En R existen diversas formas de obtener estos gráficos. Las más usuales son a través de las funciones básicas implementadas en el propio programa (hist,barplot,pie, etc.). Sin embargo, los gráficos resultantes no son del todo vistosos y carecen de 'profesionalidad'. Una herramienta disponible en R capaz de realizar gráficos de alta calidad es el paquete **ggplot2**. La librería ggplot2 de R es un sistema organizado de visualización de datos que cuenta con diversas posibilidades para realizar gráficos complejos. Este paquete provee una estructura para que se pueda especificar qué variables representar, cómo deben ser presentadas y otras propiedades visuales generales. De esta manera, sólo se necesita realizar cambios mínimos si los datos sufren alguna modificación o si se decide pasar, por ejemplo, de un gráfico de barras a un diagrama de dispersión. Esto ayuda a crear gráficos con calidad para informes con pocos ajustes adicionales. Los elementos necesarios para representar un gráfico mediante ggplot2 son:

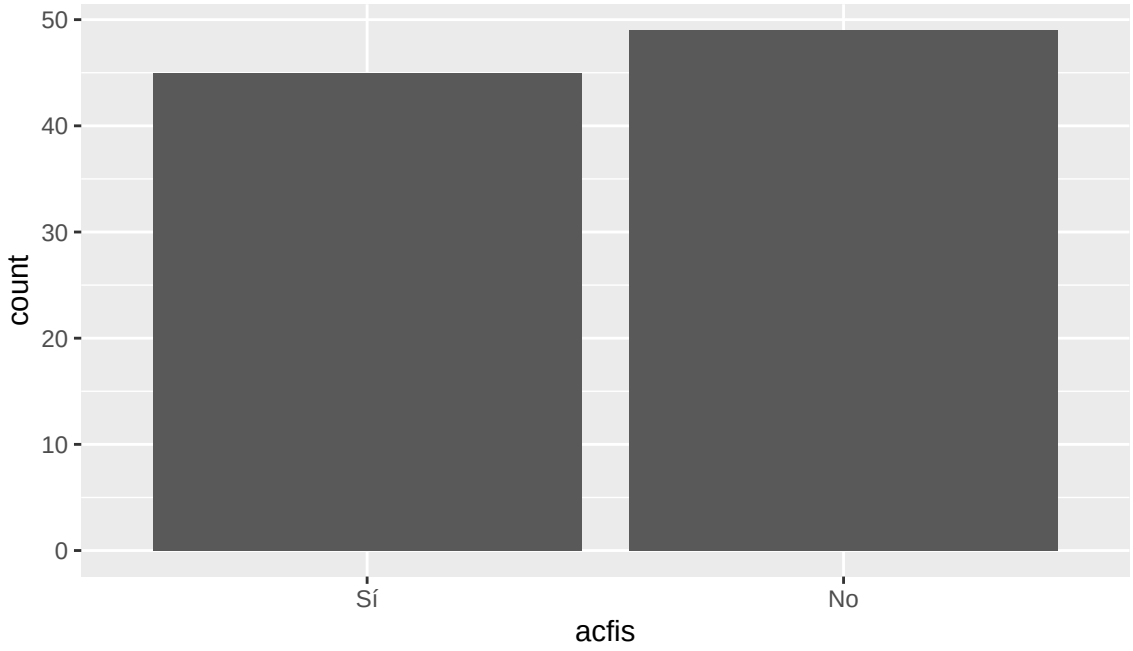
1. Un *dataframe* que contengan los datos que se quieran representar.
2. Una lista de relaciones que permitan interactuar a las variables entre sí.
3. Los *geoms* que representan los elementos geométricos del gráfico.

El esquema básico por el que se rige el paquete ggplot2 es `ggplot(data.frame, aes(x = variable)) + geom_forma()`. A este esquema se le puede añadir un sinfín de elementos (denominadas capas) que modifiquen la forma y el aspecto de la visualización. Al ser el presente un curso básico de estadística y, en particular, de R, se dará las nociones principales para que el estudiantado sea capaz de hacer representaciones básicas con el programa y tenga los conocimientos suficientes que le permitan adentrarse en la representación de datos más sofisticadas a través de ggplot2. En la red existen múltiples recursos para profundizar en este tema.

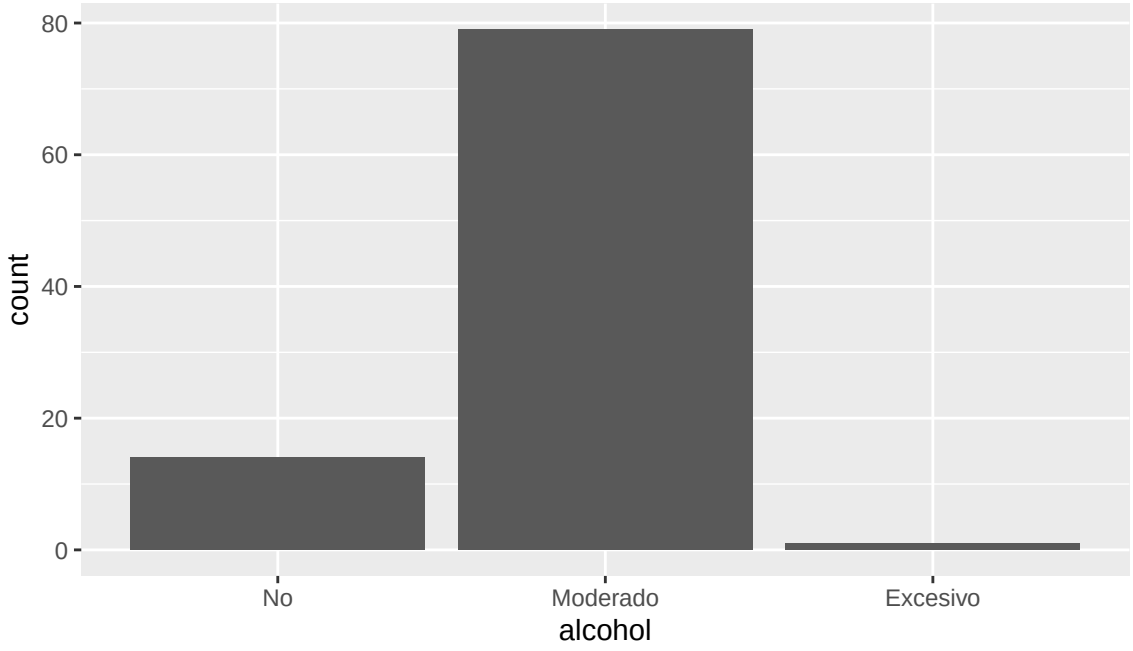
3.3.1 Diagrama de barras

Para realizar un diagrama de barras, se introduce el nombre del fichero con el que se está trabajando, la variable que se quiera representar en el gráfico y se indica el tipo de gráfico. A continuación figuran el gráfico de barras de la variable *acfis* y *alcohol* de la base de datos **osteo.sav**.

```
library(ggplot2)
ggplot(osteo, aes(x = acfis)) + geom_bar()
```

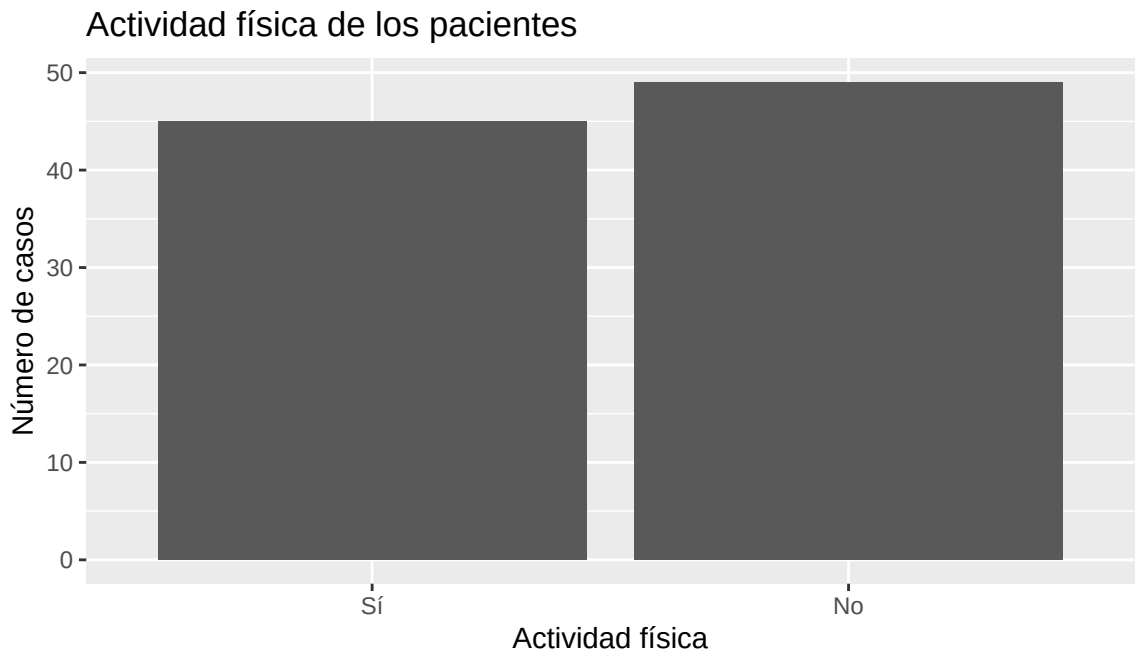



```
ggplot(osteo, aes(x = alcohol)) + geom_bar()
```



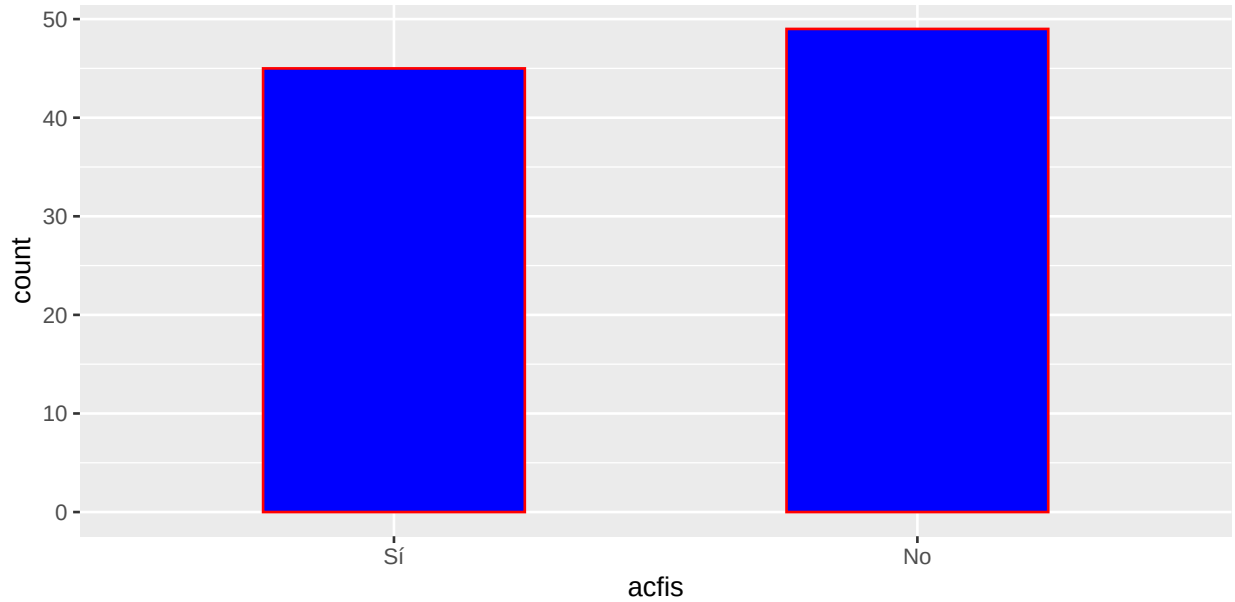
Los gráficos anteriores son las versiones más simples de un gráfico de barras con ggplot2. Se pueden añadir otras capas básicas como pueden ser establecer etiquetas a los ejes y añadir un título. Para ello, lo ideal es guardar el gráfico en su forma más simple (tal cual se ha definido en los ejemplos anteriores) en un objeto e ir añadiendo las capas correspondientes.

```
barras.acfis=ggplot(osteo, aes(x = acfis)) + geom_bar()
barras.acfis + xlab("Actividad física") + ylab ("Número de casos") +
    ggtitle("Actividad física de los pacientes")
```



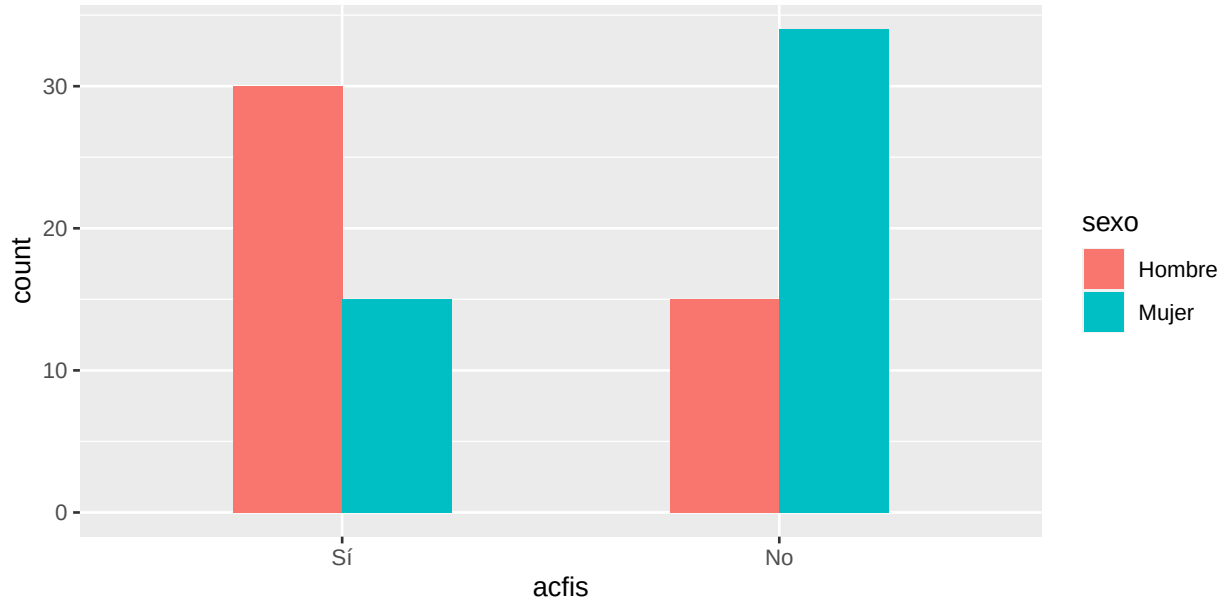
También se pueden modificar el estilo y el color de las barras, así como el color del fondo del gráfico. El paquete ggplot dibuja establece por defecto el gráfico sobre un fondo gris pero se puede cambiar a blanco y negro añadiendo el comando **theme_bw()** y **theme_dark()** respectivamente.

```
ggplot(osteo, aes(x = acfis)) + geom_bar(width=0.5, colour="red", fill="blue")
```

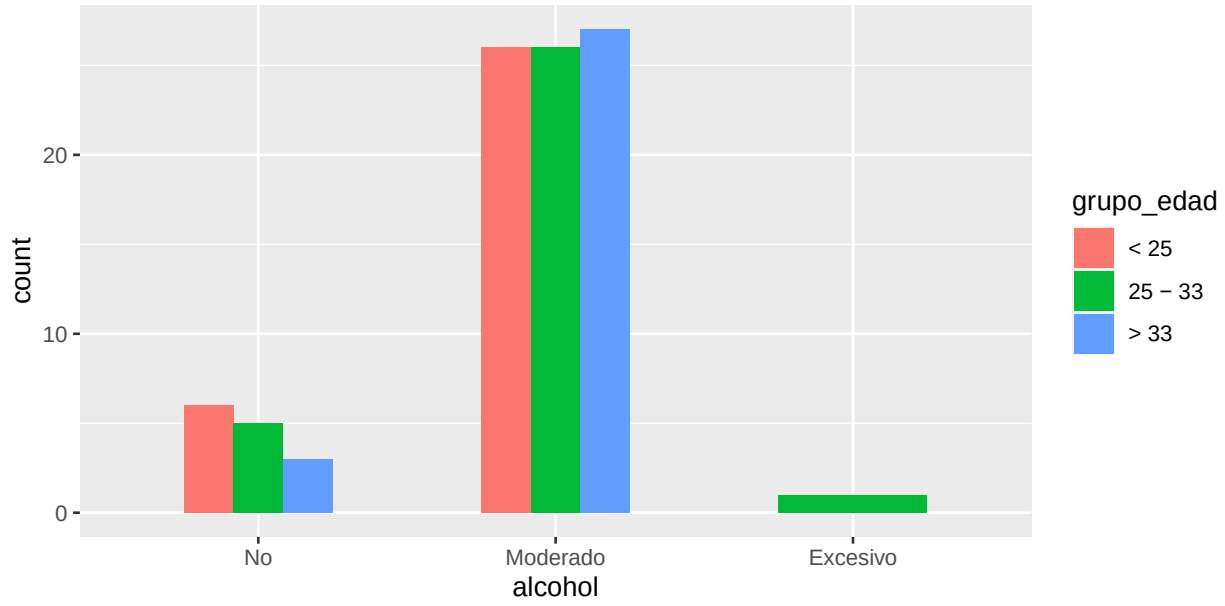


Otra opción interesante es representar un gráfico de barras de una variable para cada una de las categorías de otra variable.

```
ggplot(osteo, aes(x = acfis, fill=sexo)) +
    geom_bar(width=0.5, position="dodge")
```



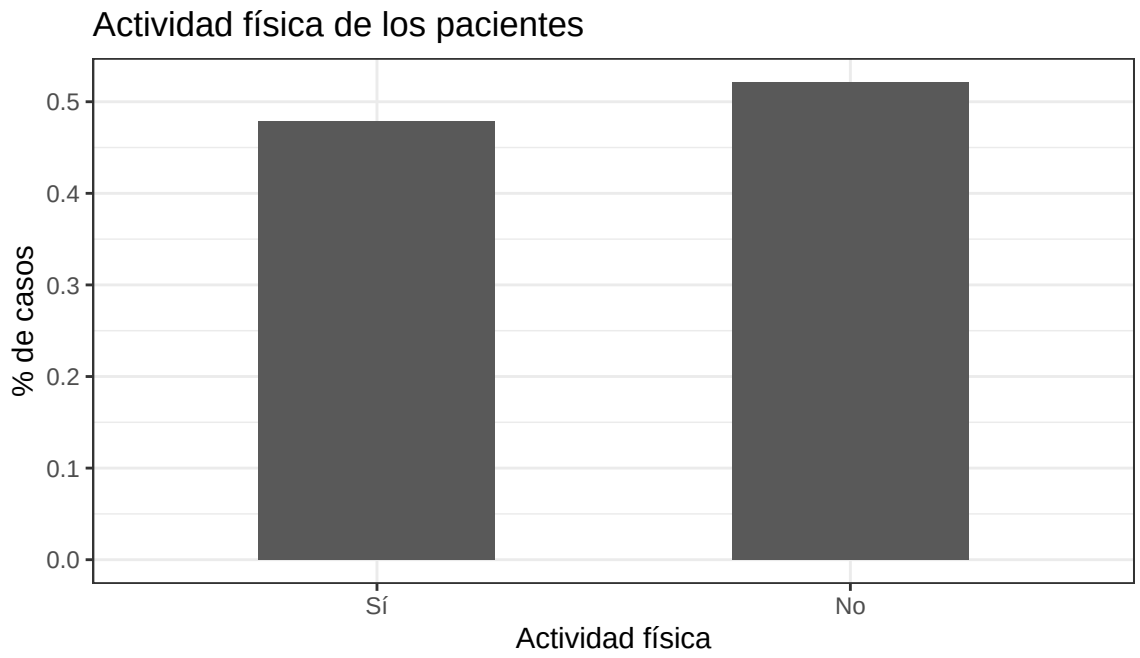

```
ggplot(osteo, aes(x = alcohol, fill=grupo_edad)) +
    geom_bar(width=0.5, position="dodge")
```



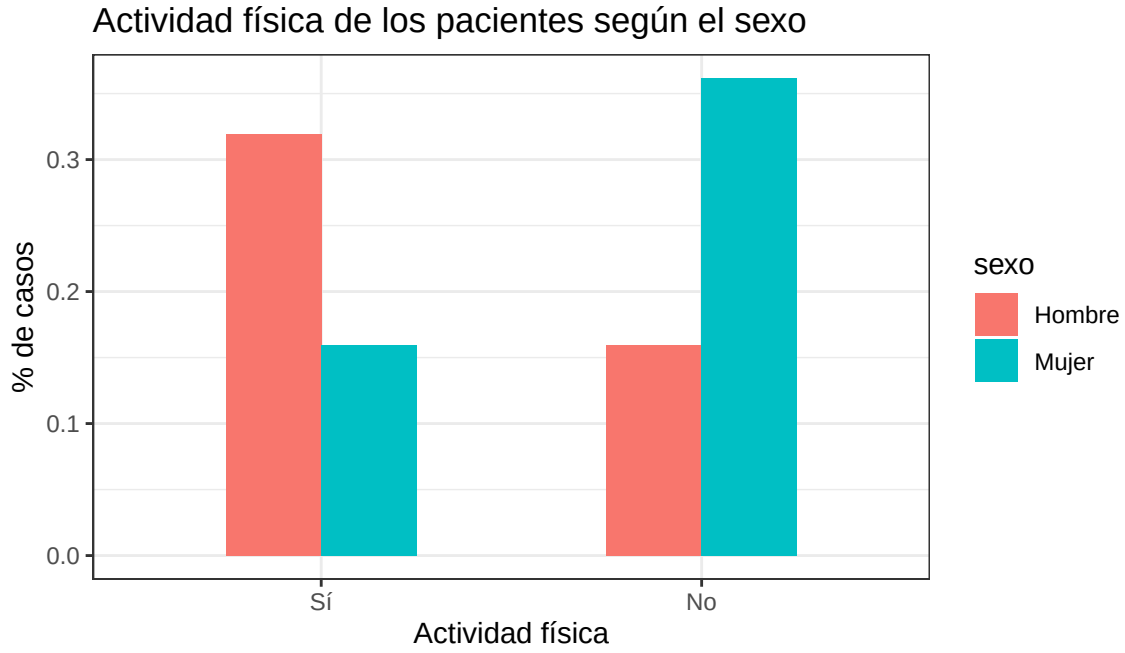
Si en lugar de querer representar el número de casos se está interesado en representar el porcentaje de casos, se procede de la siguiente manera

```
barras.acfis=ggplot(osteo, aes(x = acfis,(..count..)/sum(..count..))) +
    geom_bar(width=0.5) + theme_bw()
```

```
barras.acfis + xlab("Actividad física") + ylab ("% de casos") +
    ggtitle("Actividad física de los pacientes")
```



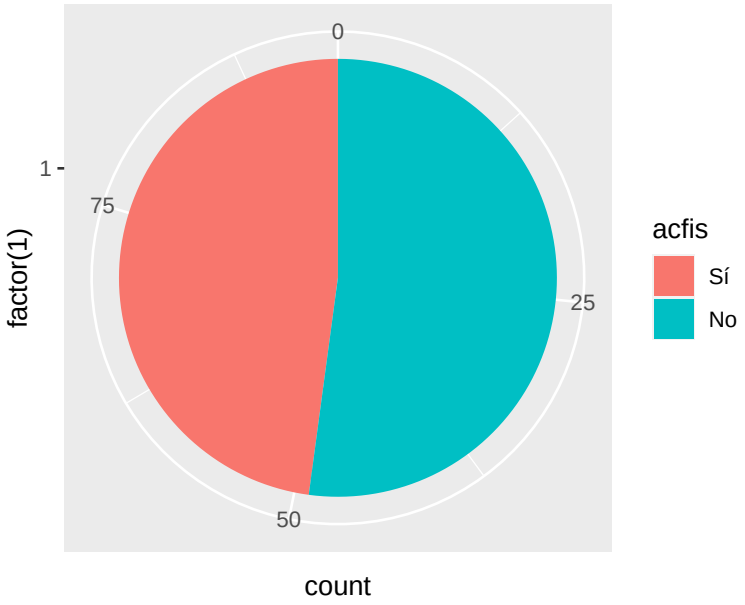

```
barras.acfis.sexo=ggplot(osteo,
    aes(x = acfis, fill=sexo,(..count..)/sum(..count..))) +
    geom_bar(width=0.5, position="dodge") + theme_bw()
barras.acfis.sexo + xlab("Actividad física") + ylab ("% de casos") +
    ggtitle("Actividad física de los pacientes según el sexo")
```



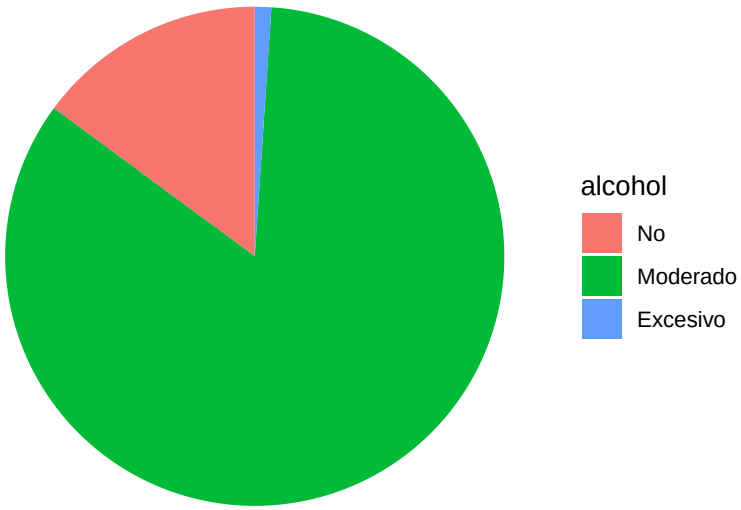
3.3.2 Diagrama de sectores

Para realizar un diagrama de sectores, se repite todo el proceso del gráfico de barras con ligeras modificaciones en la introducción de la variable de interés y se añade la capa `coord_polar("y")`. A continuación figuran el gráfico de sectores de la variable *acfis* y *alcohol* de la base de datos **osteo.sav**. El comando `theme_void()` se utiliza en el segundo ejemplo para suprimir el fondo.

```
ggplot(osteo,aes(x=factor(1),fill=acfis))+geom_bar()+coord_polar("y")
```

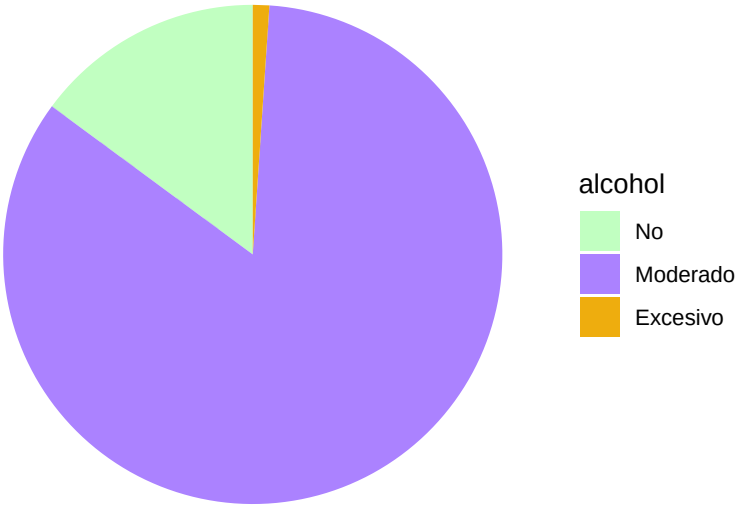



```
ggplot(osteo,aes(x=factor(1),fill=alcohol))+geom_bar()+coord_polar("y")+
    theme_void()
```



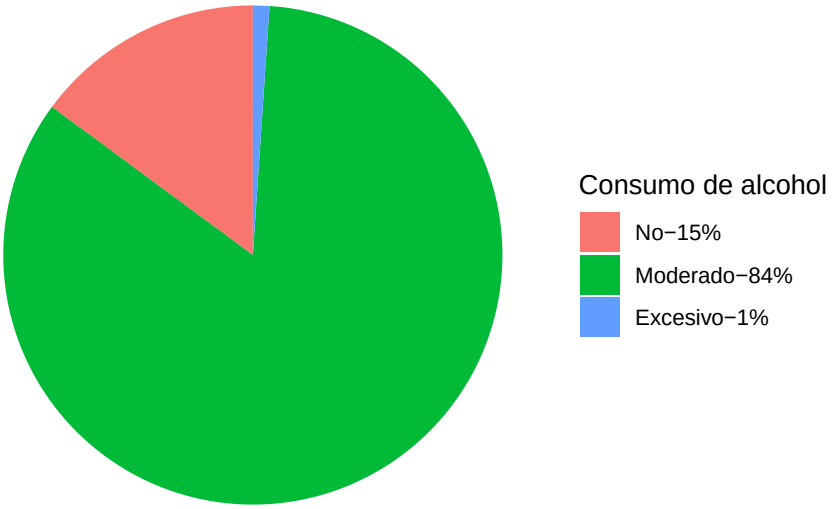
Se pueden modificar los colores de las distintas porciones que conforman el diagrama. En la siguiente dirección web figuran todos los colores disponibles en R http://www.stat.columbia.edu/~tzheng/files/Rcolor.pdf.

```
sectores=ggplot(osteo,aes(x=factor(1),fill=alcohol))+geom_bar()+
    coord_polar("y")+theme_void()
sectores+scale_fill_manual(values=c("darkseagreen1", "mediumpurple1",
    "darkgoldenrod2"))
```



Introducir el porcentaje de casos dentro del gráfico no es una tarea sencilla ni inmediata. Este procedimiento requiere el uso de un paquete que no está disponible para todas las versiones de R, y por ende, puede provocar problemas a la hora de elaborar este tipo de gráficos. Una posibilidad es modificar la leyenda e introducir en ella los porcentajes de casos.

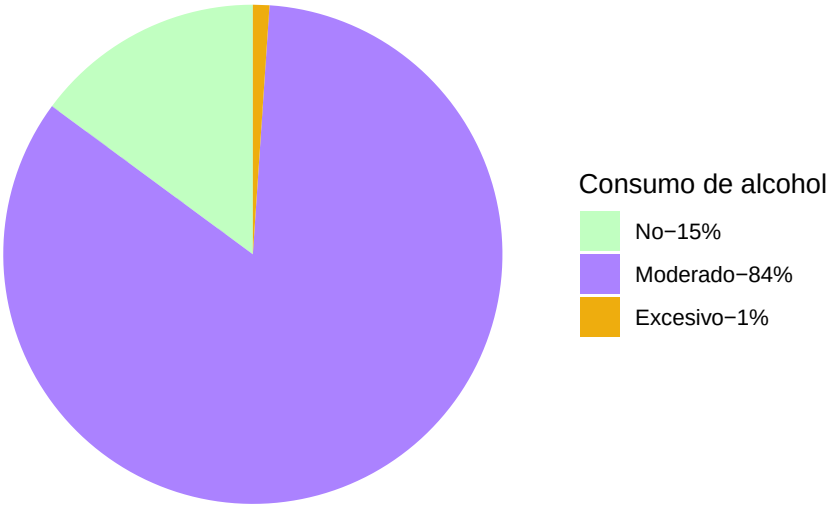
```
sectores + scale_fill_discrete(name="Consumo de alcohol",
    labels=c("No-15%", "Moderado-84%", "Excesivo-1%"))
```



NOTA: Los porcentajes de casos han sido calculados previamente usando la función **prop.table(table())**.

En cambio, si se quiere modificar la leyenda y los colores de las porciones del gráfico, se debe recurrir al elemento **scale_fill_manual** en lugar de **scale_fill_discrete**.

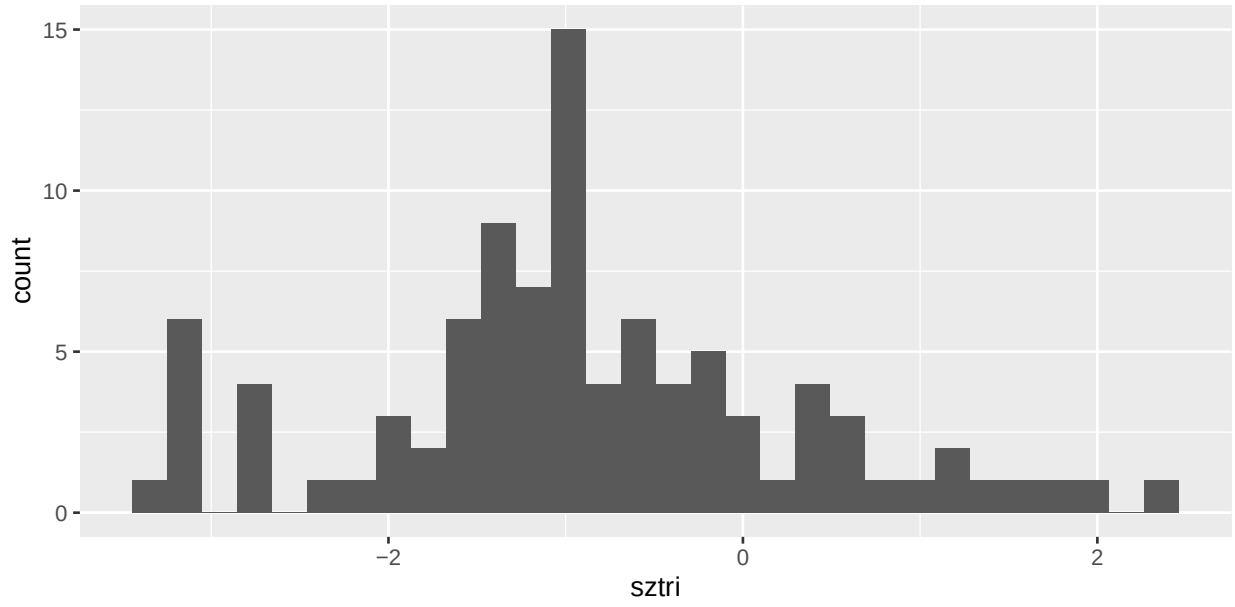
```
sectores + scale_fill_manual(
  values=c("darkseagreen1", "mediumpurple1", "darkgoldenrod2"),
  name="Consumo de alcohol",
  labels=c("No-15%", "Moderado-84%", "Excesivo-1%"))
```



3.3.3 Histograma

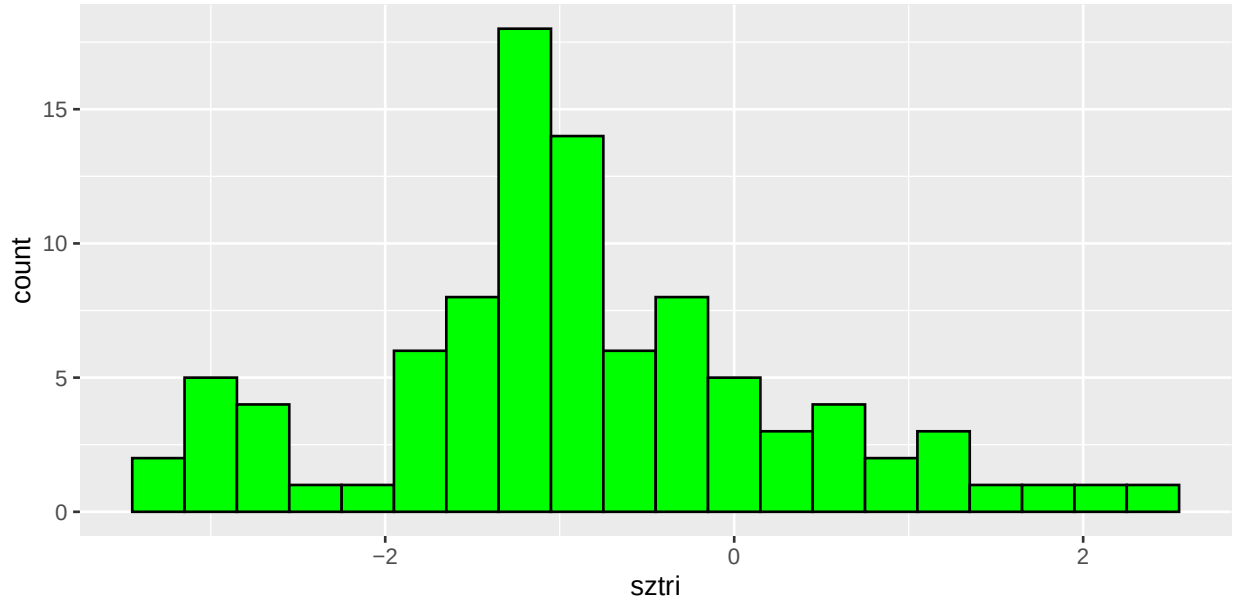
Representar un histograma de una variable continua con ggplot2 es muy sencillo de conseguir. La forma de actuar se asemeja a la mecánica seguida en los gráficos de diagramas de barras salvo con ligeras modificaciones.

```
ggplot(osteo, aes(x=sztri)) + geom_histogram()
```

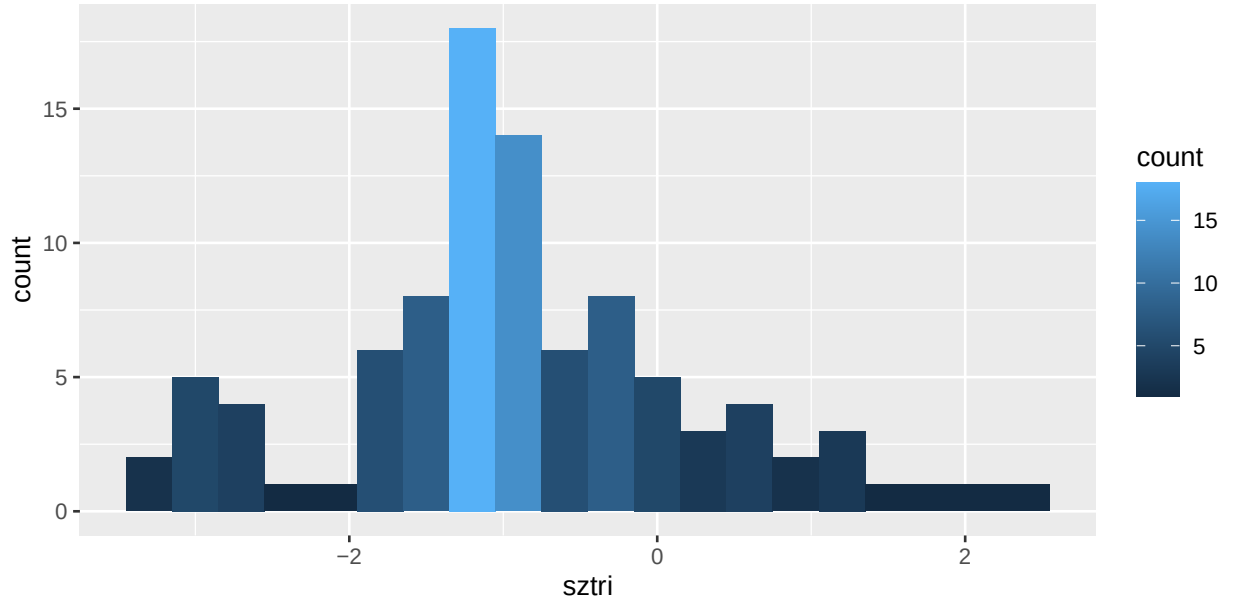


Para modificar el color y forma de las barras se deben añadir los elementos oportunos dentro del argumento **geom_histogram**.

```
ggplot(osteo, aes(x=sztri)) + geom_histogram(binwidth = 0.3,
    col="black", fill="green")
```

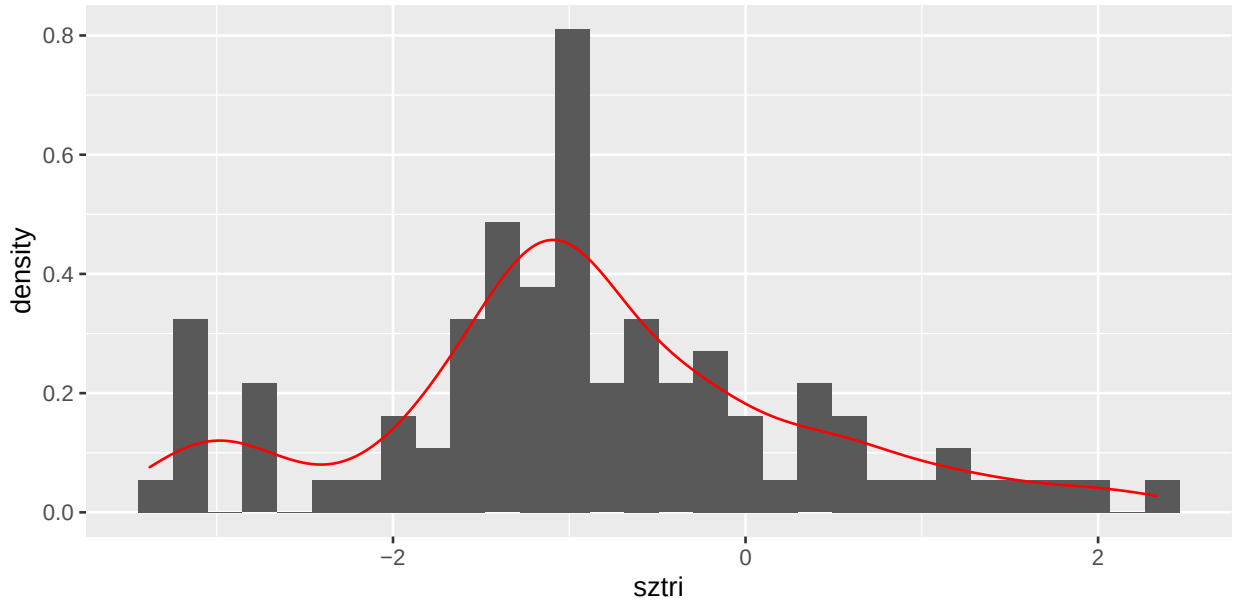


```
ggplot(osteo, aes(x=sztri)) + geom_histogram(binwidth = 0.3,
    aes(fill=..count..))
```



También se puede mostrar en el eje de ordenadas la densidad en lugar del número de casos en cada intervalo.

```
ggplot(osteo, aes(x=sztri)) + geom_histogram(aes(y=..density..)) +
    geom_density(col="red")
```



3.3.4 Polígono de frecuencias

Para representar el polígono de frecuencias en el ámbito de conteo de casos, no se puede actuar de forma inmediata como en el caso del diagrama de barras. En este contexto hay que calcular las frecuencias absolutas

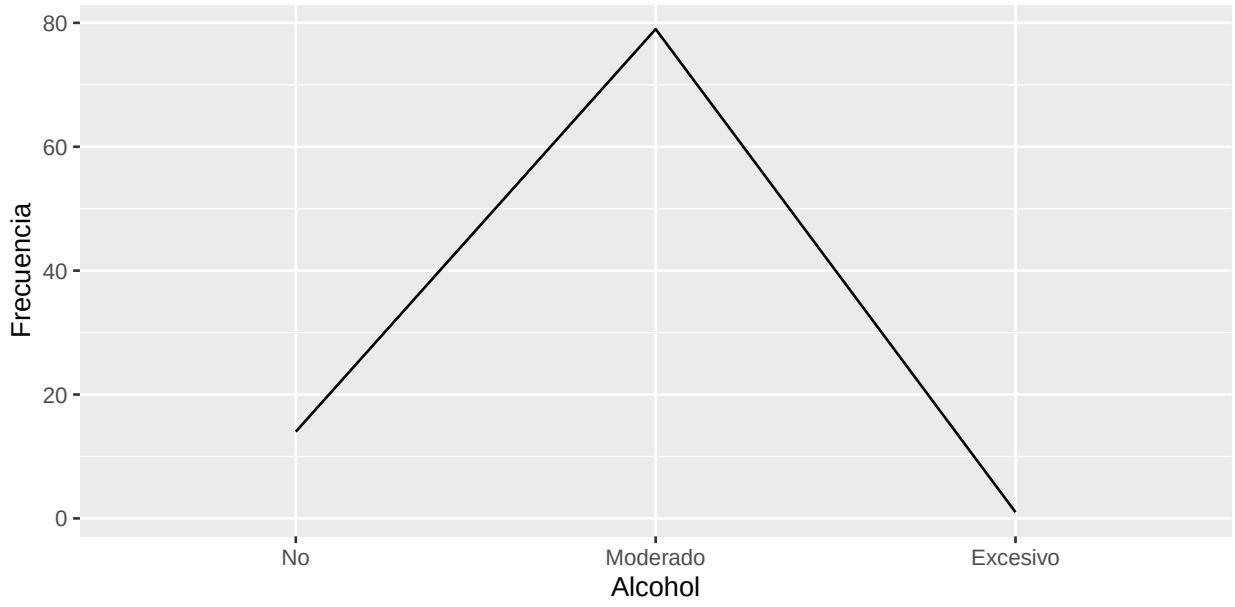
o relativas de la variable de interés y pasar el objeto creado de tipo *tabla* a un *dataframe*. si se tiene un objeto de un determinado tipo y se pretende cambiar a otro (si resulta posible porque no siempre lo será), la función a utilizar será **as.** seguida del tipo deseado.

Como ejemplo, se calcula las frecuencias absolutas de la variable *alcohol* del fichero **osteo.sav**. Como se vio en el apartado [1.2.1 Distribuciones de frecuencias en R], el objeto creado es de tipo *tabla*. Entonces, para modificar la tipología de este objeto a *dataframe* se utiliza la función **as.data.frame()**.

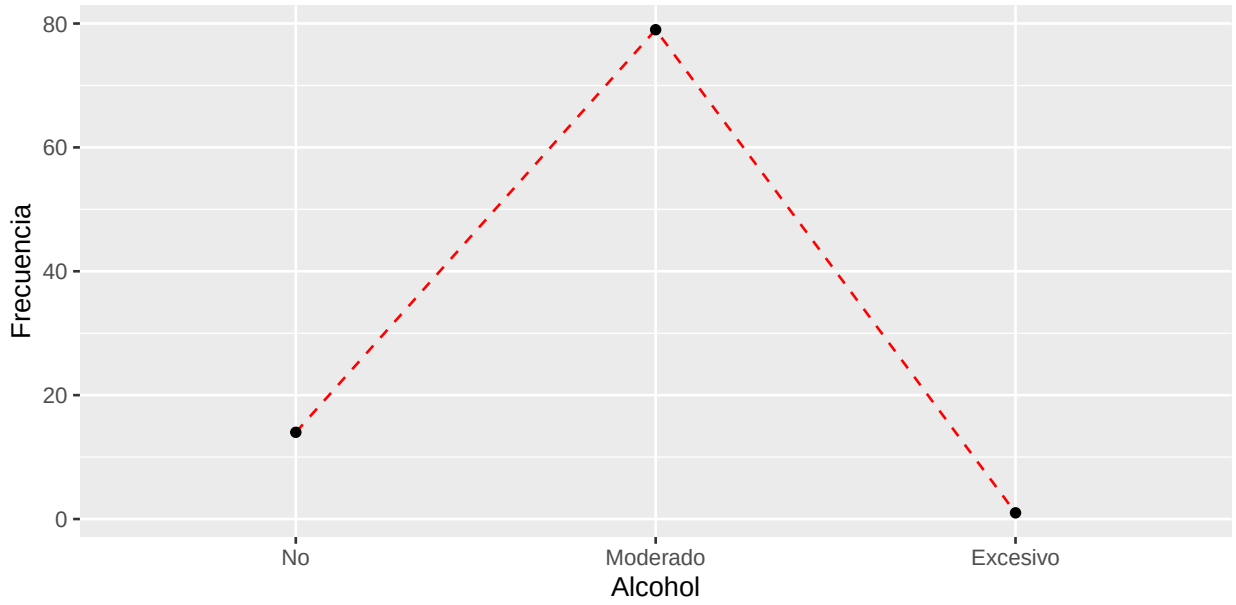
```
df=table(alcohol)
df=as.data.frame(df)
colnames(df)=c("Alcohol","Frecuencia")
```

A partir de este objeto ya en formato *dataframe*, ya se pueden añadir las líneas.

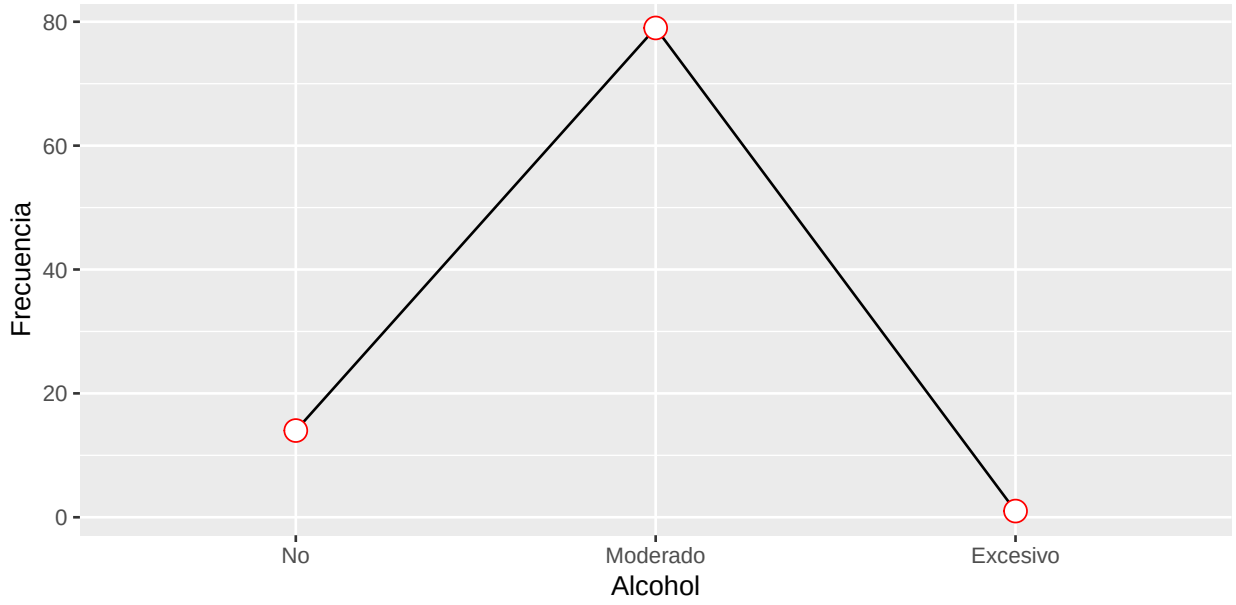
```
ggplot(df, aes(x=Alcohol,y=Frecuencia,group=1)) + geom_line()
```




```
ggplot(df, aes(x=Alcohol,y=Frecuencia,group=1)) +
    geom_line(linetype = "dashed",col="red")+geom_point()
```

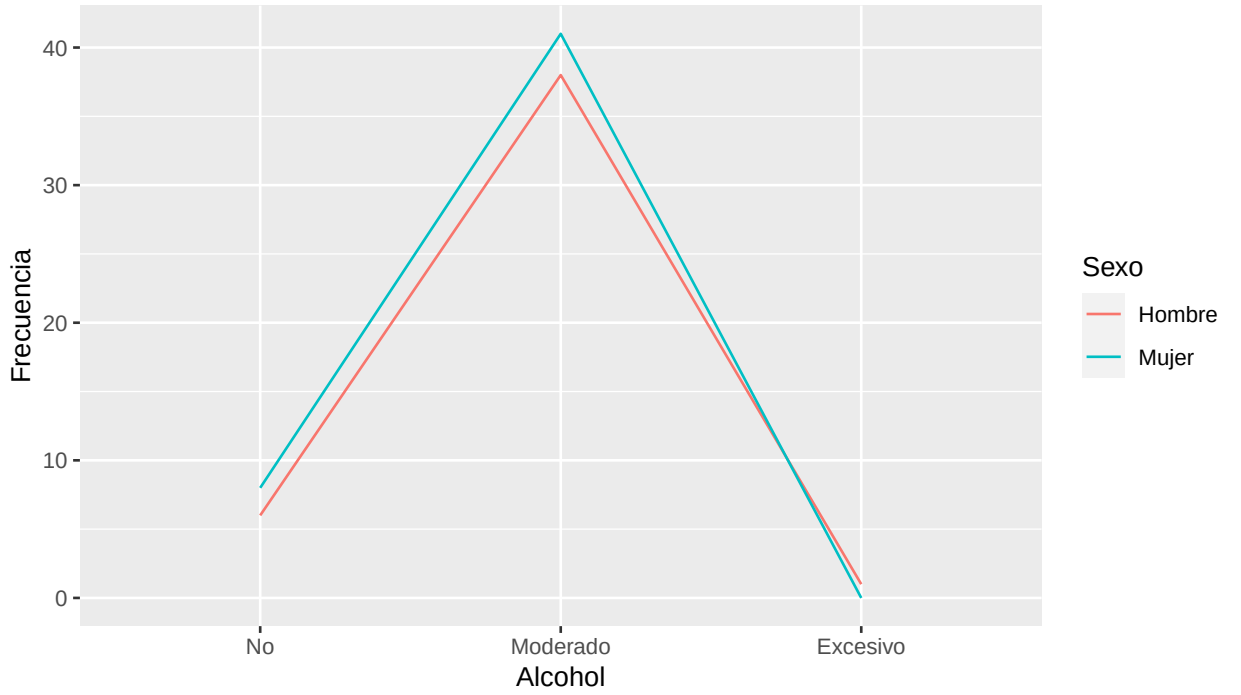



```
ggplot(df, aes(x=Alcohol,y=Frecuencia,group=1)) + geom_line()+
    geom_point(colour="red", size=4, shape=21, fill="white")
```



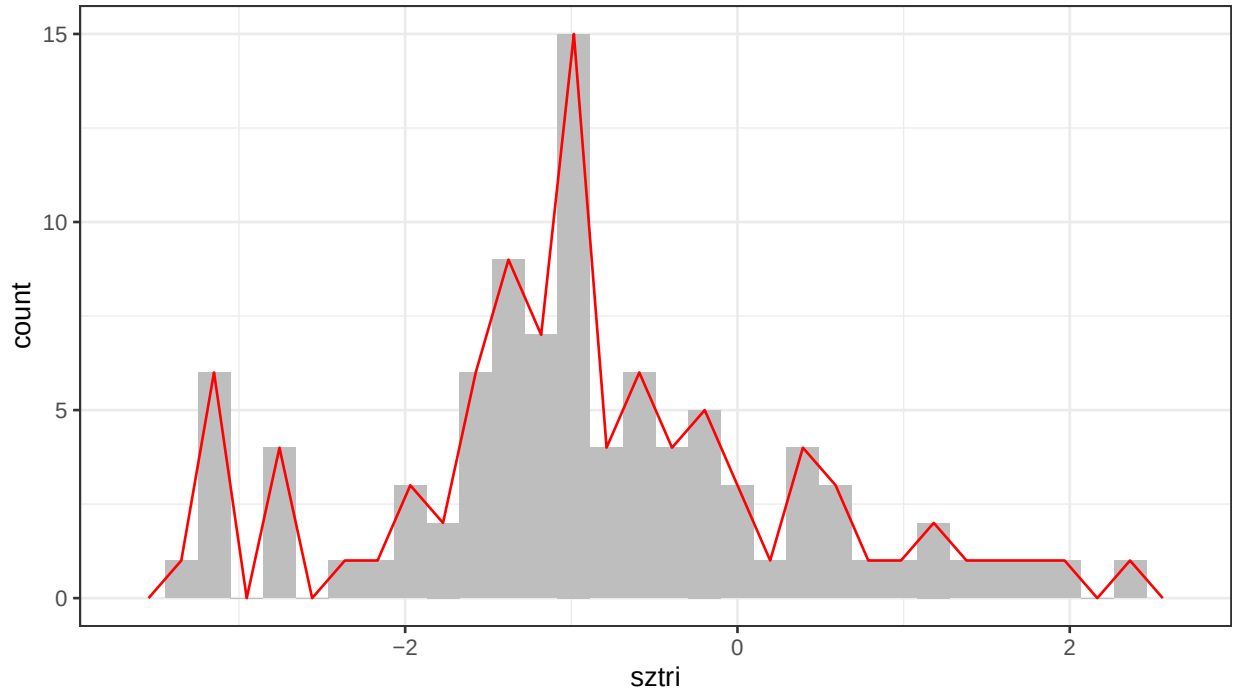
Si se quiere realizar el mismo gráfico pero distinguiendo los casos en función de otra variable, hay que añadir esta segunda variable al *dataframe* desde el comienzo del análisis. Por ejemplo, si se añade la variable *sexo* se tiene lo siguiente.

```
df=table(alcohol,sexo)
df=as.data.frame(df)
colnames(df)=c("Alcohol","Sexo","Frecuencia")
ggplot(df, aes(x=Alcohol,y=Frecuencia,group=Sexo,colour=Sexo)) + geom_line()
```



En el caso del histograma, el polígono de frecuencias se añade al gráfico con la capa **geom_freqpoly**.

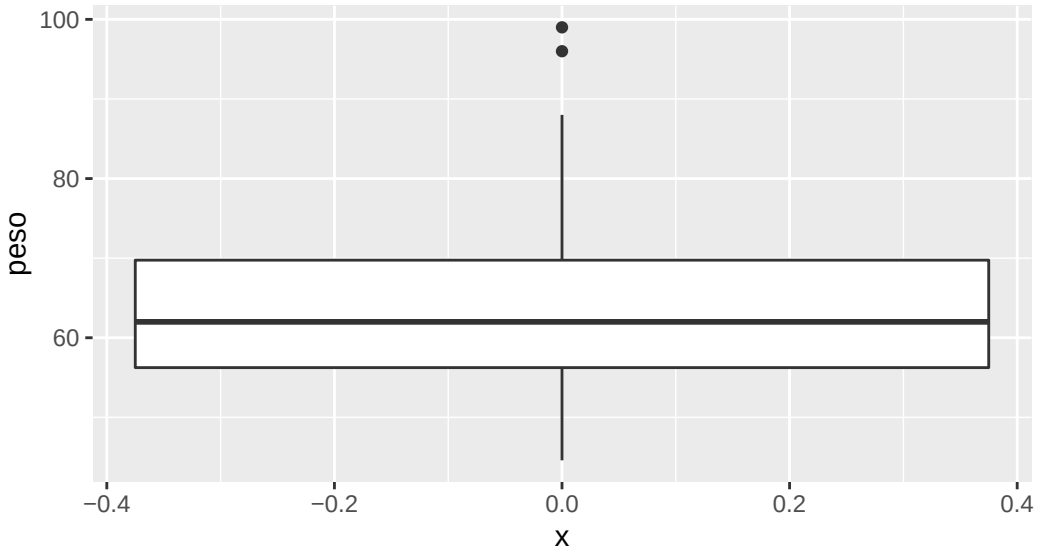
```
ggplot(osteo, aes(x=sztri)) +  geom_histogram(fill="grey") +
    geom_freqpoly(col="red") +theme_bw()
```



3.3.5 Gráfico de cajas y bigotes (Boxplot)

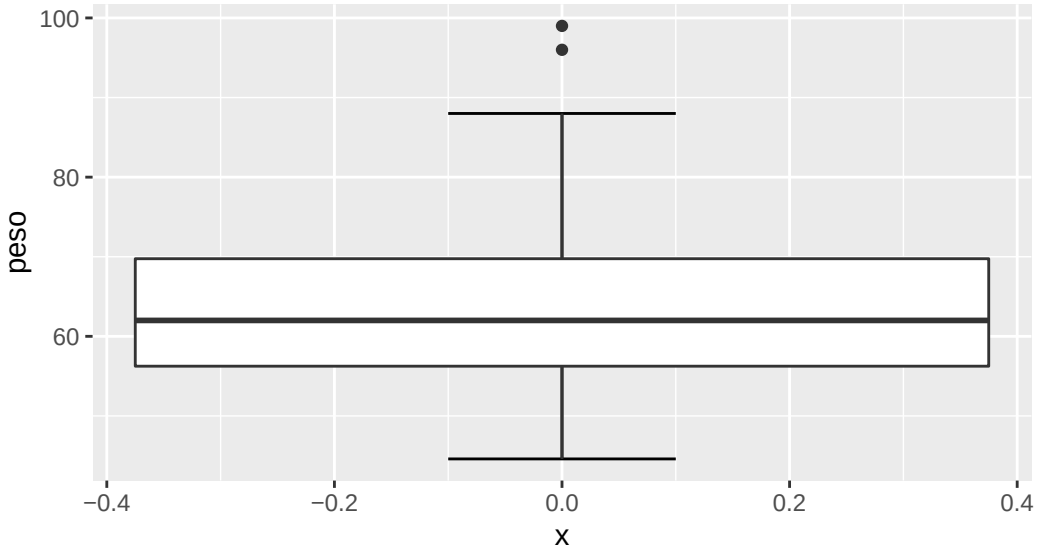
Para realizar el gráfico de cajas y bigotes con la función ggplot, procedemos de forma similar a la de los casos anteriores, esta vez añadiendo **geom_boxplot**

```
ggplot(osteo, aes(x =0, y=peso)) + geom_boxplot()
```



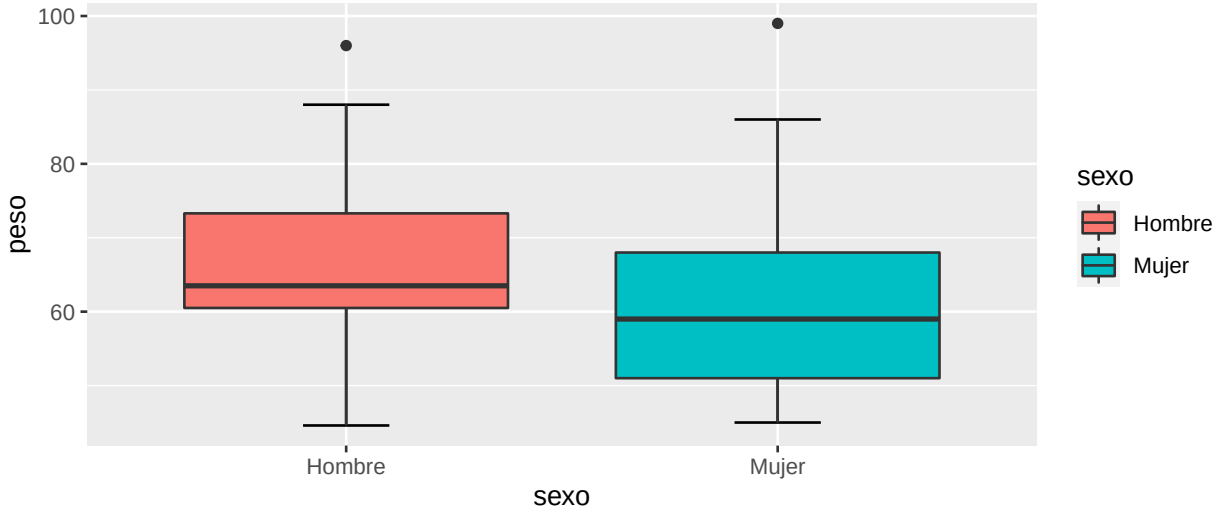
Al igual que con los gráficos anteriores, existen diversas opciones que nos permiten modificar y mejorar la visualización del boxplot, como por ejemplo, añadir barras de error.

```
ggplot(osteo, aes(x =0, y=peso)) +
stat_boxplot(geom = "errorbar",
             width = 0.20) +
  geom_boxplot()
```



También es posible modificar colores, asi como realizar el gráfico por grupos

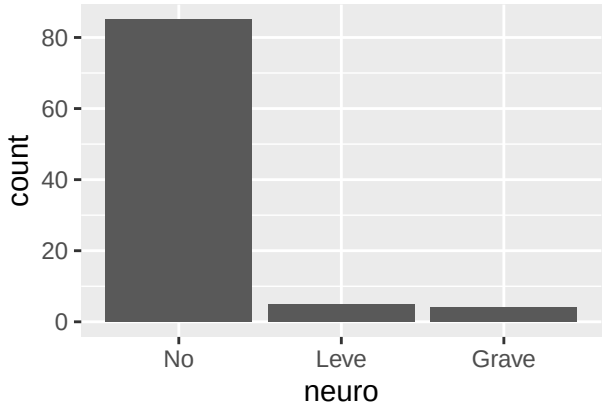
```
ggplot(osteo, aes(x =sexo, y=peso, fill=sexo)) +
stat_boxplot(geom = "errorbar",
             width = 0.20) +
  geom_boxplot()
```

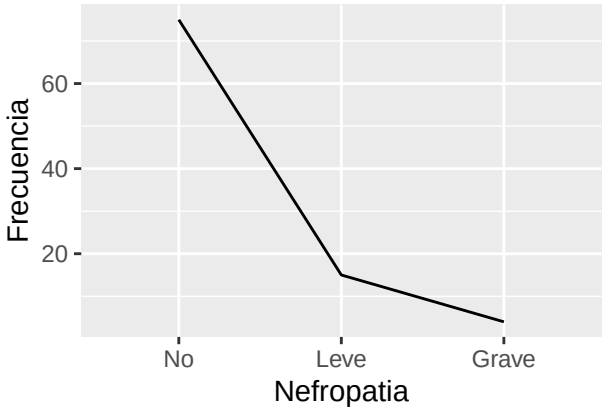


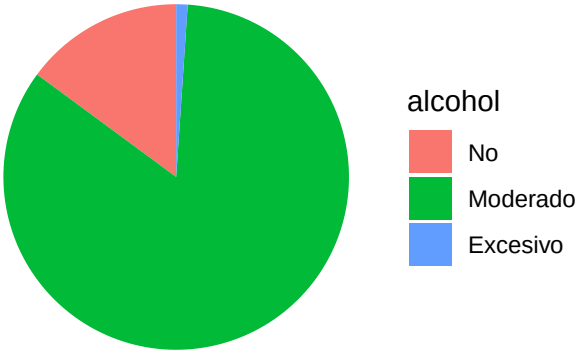
3.3.6 Representaciones múltiples

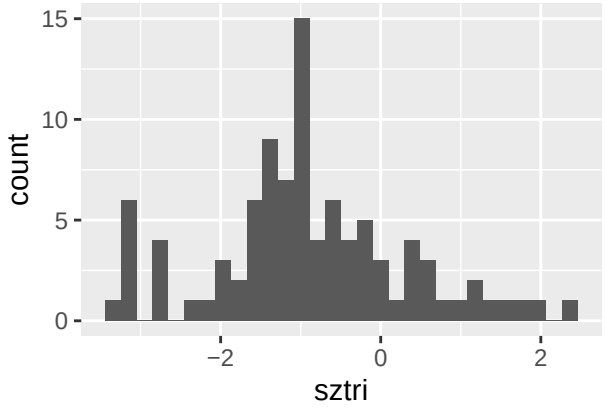
Hasta el momento solo se han representado una única figura en cada gráfico. Sin embargo, en ocasiones puede ser interesante representar varios gráficos en el mismo 'lienzo' o 'panel' de forma que el informe resulte más vistoso y quede más formal que realizar múltiples figuras por separados. Aunque existen una gran cantidad de paquetes en R para realizar representaciones múltiples, quizás el más sencillo para ello sea el paquete **gridExtra**. Una vez instalado y cargado dicha librería, se utiliza la función **grid.arrange()** en la que hay que introducir los distintos gráficos que se quieren representar, seguido del número de filas y columnas. El número de filas y columnas determinará la forma en la que se distribuirán los gráficos sobre el lienzo: si se tiene cuatro gráficos y se quieren representar todos en la misma línea, habrá que indicar que *ncol=4* y *nrow=1*; en cambio, si el objetivo es colocar cada figura una debajo de la otra los parámetros tomarán los valores *ncol=1* y *nrow=4*.

```
barras=ggplot(osteo, aes(x=neuro)) +  geom_bar()
sectores=ggplot(osteo,aes(x=factor(1),fill=alcohol))+geom_bar()+
    coord_polar("y")+theme_void()
histograma=ggplot(osteo, aes(x=sztri)) + geom_histogram()

df=table(nefro)
df=as.data.frame(df)
colnames(df)=c("Nefropatia","Frecuencia")
lineas=ggplot(df, aes(x=Nefropatia,y=Frecuencia,group=1)) + geom_line()

library(gridExtra)
grid.arrange(barras,lineas,sectores,histograma, nrow = 2,ncol=2)
```


80
60
40
20
0
count
No
Leve
Grave
neuro


60
40
20
Frecuencia
No
Leve
Grave
Nefropatia


alcohol
No
Moderado
Excesivo


15
10
5
0
count
−2
0
2
sztri