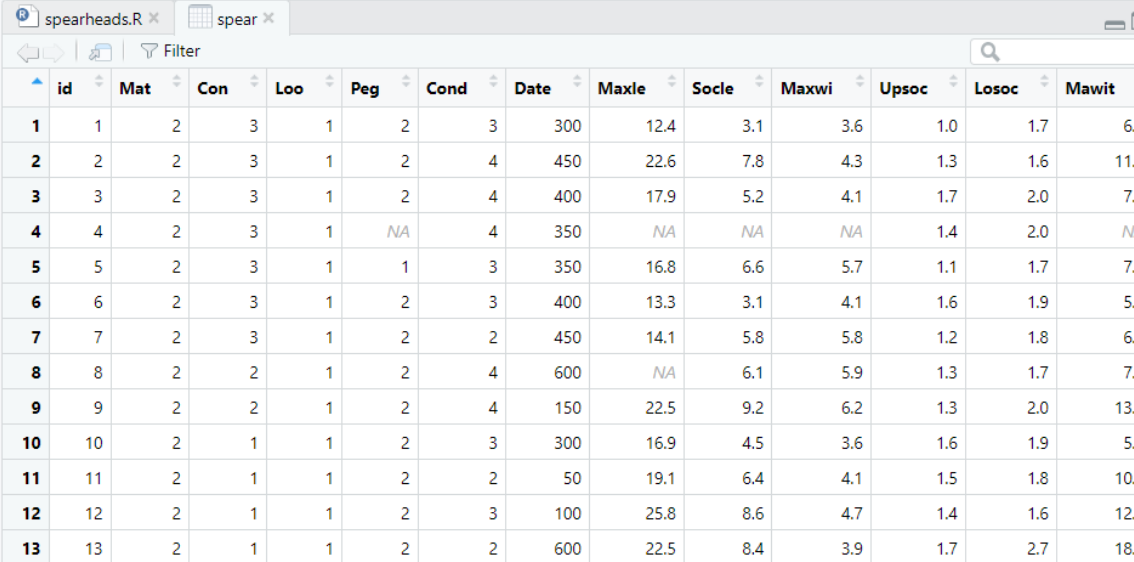


Practica 4. Spearheads.

Manuel Tristán Martín Ruiz

```
library(readxl)
#Ej.1.
#Para importar los datos del archivo excel utilizamos la función
"read_excel" y dentro de los paréntesis definimos la "url" de nuestro
archivo Excel. Para ello debemos tener descargado el paquete "readxl"
en el gestor del directorio y después tenemos que activarlo mediante
la función "library" y entre paréntesis escribimos en nombre de la
librería que queremos activar (readxl). Es importante diferenciar
entre lo que es descargar el paquete y lo que es activarlo dentro del
editor de códigos.
```

```
spear <-
read_excel("C:/Users/manueltristan/Documents/spearheads/spearheads.xls
x")
View(spear)
str(spear)
class(spear)
```



	id	Mat	Con	Loo	Peg	Cond	Date	Maxle	Socle	Maxwi	Upsoc	Losoc	Mawit
1	1	2	3	1	2	3	300	12.4	3.1	3.6	1.0	1.7	6.
2	2	2	3	1	2	4	450	22.6	7.8	4.3	1.3	1.6	11.
3	3	2	3	1	2	4	400	17.9	5.2	4.1	1.7	2.0	7.
4	4	2	3	1	NA	4	350	NA	NA	NA	1.4	2.0	NA
5	5	2	3	1	1	3	350	16.8	6.6	5.7	1.1	1.7	7.
6	6	2	3	1	2	3	400	13.3	3.1	4.1	1.6	1.9	5.
7	7	2	3	1	2	2	450	14.1	5.8	5.8	1.2	1.8	6.
8	8	2	2	1	2	4	600	NA	6.1	5.9	1.3	1.7	7.
9	9	2	2	1	2	4	150	22.5	9.2	6.2	1.3	2.0	13.
10	10	2	1	1	2	3	300	16.9	4.5	3.6	1.6	1.9	5.
11	11	2	1	1	2	2	50	19.1	6.4	4.1	1.5	1.8	10.
12	12	2	1	1	2	3	100	25.8	8.6	4.7	1.4	1.6	12.
13	13	2	1	1	2	2	600	22.5	8.4	3.9	1.7	2.7	18.

```
#Para convertir los datos a un data.frame, utilizamos la función
"as.data.frame" y le asignamos al data.frame con el nombre "spear".
```

```
spear <- as.data.frame(spear)
class(spear)
```

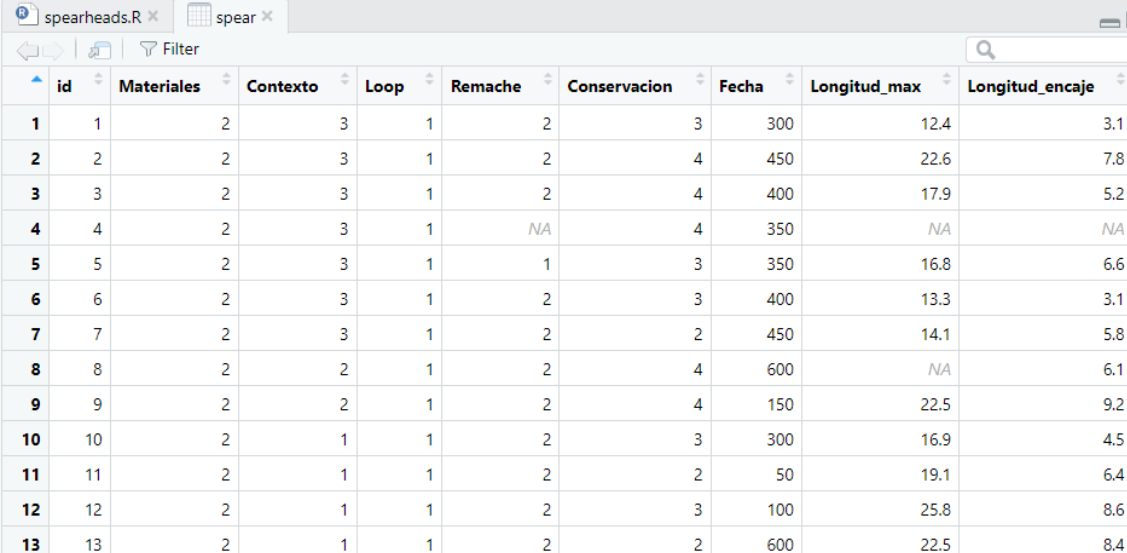
```
#Ej.2.
#Para renombrar las variables utilizamos la función "names", entre
paréntesis definimos de donde son los nombres "(spear)" y con la
función lógica "==" compara con el nombre de las columnas en el
data.frame, por ejemplo "Mat", después le asignamos el nombre por el
que lo queremos cambiar, por ejemplo "Materiales".
```

```
names(spear)[names(spear) == "Mat"] <- "Materiales"
names(spear)[names(spear) == "Con"] <- "Contexto"
names(spear)[names(spear) == "Cond"] <- "Conservacion"
names(spear)[names(spear) == "Loo"] <- "Loop"
names(spear)[names(spear) == "Peg"] <- "Remache"
names(spear)[names(spear) == "Date"] <- "Fecha"
```

```

names(spear)[names(spear) == "Maxle"] <- "Longitud_max"
names(spear)[names(spear) == "Socle"] <- "Longitud_encaje"
names(spear)[names(spear) == "Maxwi"] <- "Ancho_max"
names(spear)[names(spear) == "Upsoc"] <- "Ancho_encaje"
names(spear)[names(spear) == "Mawit"] <- "Ancho_max_encaje"
names(spear)[names(spear) == "Weight"] <- "Peso"
spear
View(spear)

```



	id	Materiales	Contexto	Loop	Remache	Conservacion	Fecha	Longitud_max	Longitud_encaje
1	1	2	3	1	2	3	300	12.4	3.1
2	2	2	3	1	2	4	450	22.6	7.8
3	3	2	3	1	2	4	400	17.9	5.2
4	4	2	3	1	NA	4	350	NA	NA
5	5	2	3	1	1	3	350	16.8	6.6
6	6	2	3	1	2	3	400	13.3	3.1
7	7	2	3	1	2	2	450	14.1	5.8
8	8	2	2	1	2	4	600	NA	6.1
9	9	2	2	1	2	4	150	22.5	9.2
10	10	2	1	1	2	3	300	16.9	4.5
11	11	2	1	1	2	2	50	19.1	6.4
12	12	2	1	1	2	3	100	25.8	8.6
13	13	2	1	1	2	2	600	22.5	8.4

```

#Ej.3
# Para este ejercicio queremos asignar etiquetas que sustituyan los
valores numéricos que tenemos. Para ello, con la operación lógica $,
decimos que dentro del data.frame (spear) que seleccione la columna
concreta en la que queremos hacer el cambio, y le decimos que esos
datos sean factores, y le decimos que los factores 1,2 y 3 los
transforme en convierta en unos factores determinados, en los que
definimos con el tipo "cadena de texto" las nuevas etiquetas.

```

```

spear$Contexto=factor(spear$Contexto, levels=c('1','2','3'),
labels=c("s/c", "Habitacional", "Funerario"))
spear$Conservacion=factor(spear$Conservacion, levels=c(1,2,3,4),
labels=c('Excelente', 'Bueno', 'Regular', 'Malo'))
spear$Remache=factor(spear$Remache, levels=c(1,2), labels=c('Si',
'No'))
spear$Materiales=factor(spear$Materiales, levels=c(1,2),
labels=c('Bronce', 'Hierro'))
View(spear)

```

	id	Materiales	Contexto	Loop	Remache	Conservacion	Fecha	Longitud_max	Longitud_encaje
1	1	Hierro	Funerario	1	No	Regular	300	12.4	3.1
2	2	Hierro	Funerario	1	No	Malo	450	22.6	7.8
3	3	Hierro	Funerario	1	No	Malo	400	17.9	5.2
4	4	Hierro	Funerario	1	NA	Malo	350	NA	NA
5	5	Hierro	Funerario	1	Si	Regular	350	16.8	6.6
6	6	Hierro	Funerario	1	No	Regular	400	13.3	3.1
7	7	Hierro	Funerario	1	No	Bueno	450	14.1	5.8
8	8	Hierro	Habitacional	1	No	Malo	600	NA	6.1
9	9	Hierro	Habitacional	1	No	Malo	150	22.5	9.2
10	10	Hierro	s/c	1	No	Regular	300	16.9	4.5
11	11	Hierro	s/c	1	No	Bueno	50	19.1	6.4
12	12	Hierro	s/c	1	No	Regular	100	25.8	8.6
13	13	Hierro	s/c	1	No	Bueno	600	22.5	8.4

#Ej.4

#En este ejercicio empleamos la función "table" para generar una tabla con los datos de nuestro data.set "spear" y con la función "\$" seleccionamos la columna de la que queremos esos datos. A esta tabla le asignamos un nombre y después con la función "View" podemos ver la tabla generada.

```
freq.mat=table(spear$Materiales)
View(freq.mat)
```

	Var1	Freq
1	Bronce	21
2	Hierro	19

```
freq.con=table(spear$Contexto)
View(freq.con)
```

	Var1	Freq
1	s/c	27
2	Habitacional	6
3	Funerario	7

```
freq.cond=table(spear$Conservacion)
View(freq.cond)
```

spearheads.R x		freq.cond x
Filter		
	Var1	Freq
1	Excelente	8
2	Bueno	18
3	Regular	9
4	Malo	5

#Ej.5

#En este ejercicio empleamos la misma función "table" que, en el anterior para generar una tabla de datos, pero ahora incluimos dos columnas de datos mediante la función "\$" en vez de solo una, para que sean los datos cruzados. De nuevo, a esta tabla le asignamos un nombre y después con la función "View" podemos ver la tabla generada.

```
materiales_contexto <- table(spear$Materiales, spear$Contexto)
View(materiales_contexto)
```

spearheads.R x materiales_contexto x

Filter

	Var1	Var2	Freq
1	Bronce	s/c	19
2	Hierro	s/c	8
3	Bronce	Habitacional	2
4	Hierro	Habitacional	4
5	Bronce	Funerario	0
6	Hierro	Funerario	7

```
materiales_conservacion <-table(spear$Materiales, spear$Conservacion)
View(materiales_conservacion)
```

spearheads.R x materiales_conservacion x

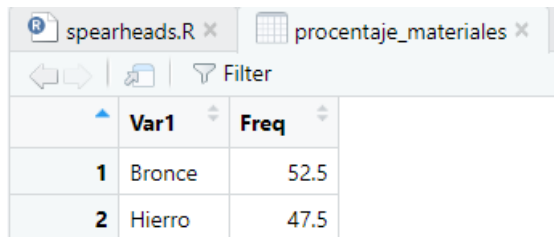
Filter

	Var1	Var2	Freq
1	Bronce	Excelente	8
2	Hierro	Excelente	0
3	Bronce	Bueno	11
4	Hierro	Bueno	7
5	Bronce	Regular	2
6	Hierro	Regular	7
7	Bronce	Malo	0
8	Hierro	Malo	5

#Ej.6

#Para este ejercicio empleamos la función "prop.table", lo que genera una tabla de porcentajes y seleccionamos la columna de los datos que nos interesan mediante la función "\$", y multiplicando por *100 para que aparezca el porcentaje (60 en vez de 0.6). Por último, de nuevo le asignamos un nombre y creamos el objeto y después con la función "View" podemos ver la tabla generada.

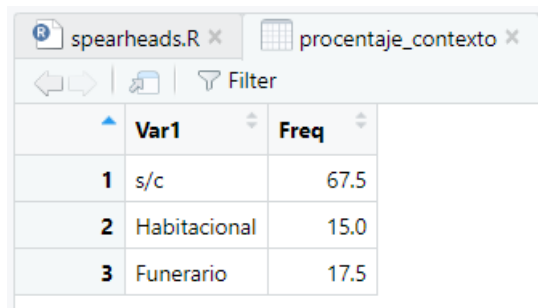
```
porcentaje_materiales <- prop.table(table(spear$Materiales)) * 100  
View(porcentaje_materiales)
```



The screenshot shows the RStudio interface with a tab for 'spearheads.R' and a 'View' window for 'porcentaje_materiales'. The window displays a table with two columns: 'Var1' and 'Freq'. The data is as follows:

	Var1	Freq
1	Bronce	52.5
2	Hierro	47.5

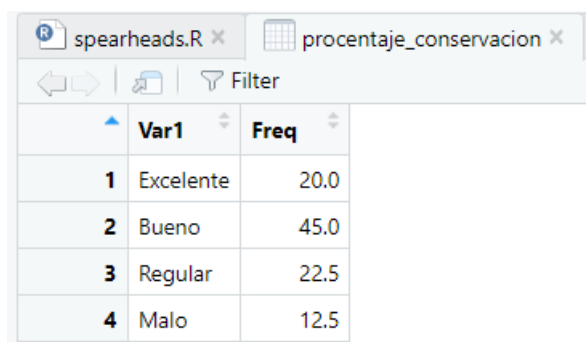
```
porcentaje_contexto <- prop.table(table(spear$Contexto)) * 100  
View(porcentaje_contexto)
```



The screenshot shows the RStudio interface with a tab for 'spearheads.R' and a 'View' window for 'porcentaje_contexto'. The window displays a table with two columns: 'Var1' and 'Freq'. The data is as follows:

	Var1	Freq
1	s/c	67.5
2	Habitacional	15.0
3	Funerario	17.5

```
porcentaje_conservacion <- prop.table(table(spear$Conservacion)) * 100  
View(porcentaje_conservacion)
```



The screenshot shows the RStudio interface with a tab for 'spearheads.R' and a 'View' window for 'porcentaje_conservacion'. The window displays a table with two columns: 'Var1' and 'Freq'. The data is as follows:

	Var1	Freq
1	Excelente	20.0
2	Bueno	45.0
3	Regular	22.5
4	Malo	12.5

#Ej.7

#En este ejercicio también empleamos la función "prop.table" para hacer una tabla en porcentaje, incluyendo dos columnas de datos que seleccionamos mediante el uso de la función "\$". La función "margin" sirve para especificar la dimensión en la que queremos hacer el cálculo. Margin = 1 significa que el porcentaje se calcule por filas. Por último, de nuevo le asignamos un nombre y creamos el objeto y después con la función "View" podemos ver la tabla generada.

```
porcentaje_materiales_contexto <- prop.table(table(spear$Materiales,  
spear$Contexto), margin = 1) * 100  
View(porcentaje_materiales_contexto)
```

	Var1	Var2	Freq
1	Bronce	s/c	90.47619
2	Hierro	s/c	42.10526
3	Bronce	Habitacional	9.52381
4	Hierro	Habitacional	21.05263
5	Bronce	Funerario	0.00000
6	Hierro	Funerario	36.84211

```

porcentaje_materiales_conservacion <-
prop.table(table(spear$Materiales, spear$Conservacion), margin = 1) *
100
View(porcentaje_materiales_conservacion)

```

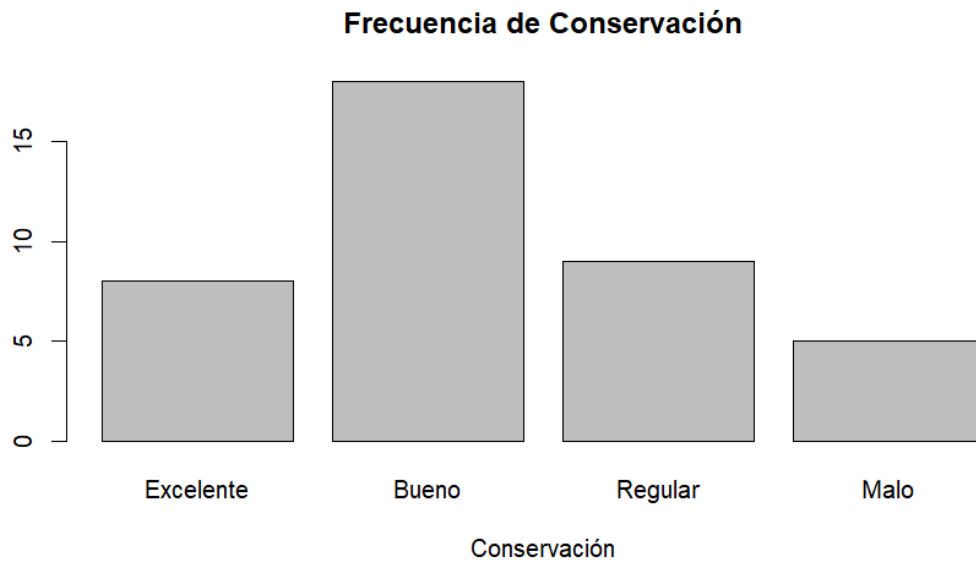
	Var1	Var2	Freq
1	Bronce	Excelente	38.09524
2	Hierro	Excelente	0.00000
3	Bronce	Bueno	52.38095
4	Hierro	Bueno	36.84211
5	Bronce	Regular	9.52381
6	Hierro	Regular	36.84211
7	Bronce	Malo	0.00000
8	Hierro	Malo	26.31579

#Ej.8
 #Para elaborar los gráficos de barras verticales utilizamos la función "barplot", seleccionando los datos concretos de una columna de nuestro data.set empleando la función \$. Por último, le asignamos un título a nuestro gráfico mediante la función "main" y con una igualdad "=" definimos su nombre. Lo mismo para nombrar el eje X, con la función "xlab".

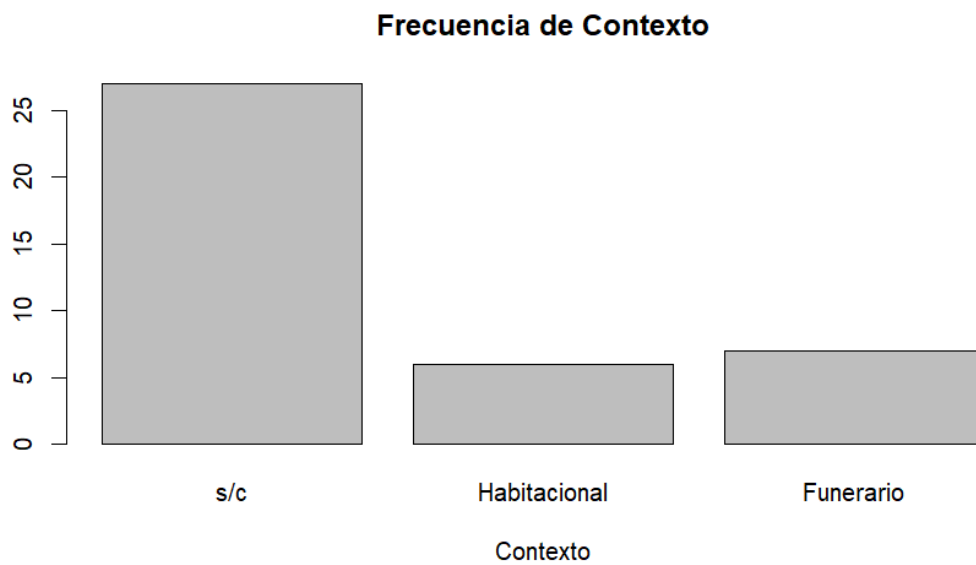
```

barras_verticales_conservacion <- barplot(table(spear$Conservacion),
main = "Frecuencia de Conservación",
xlab = "Conservación")

```

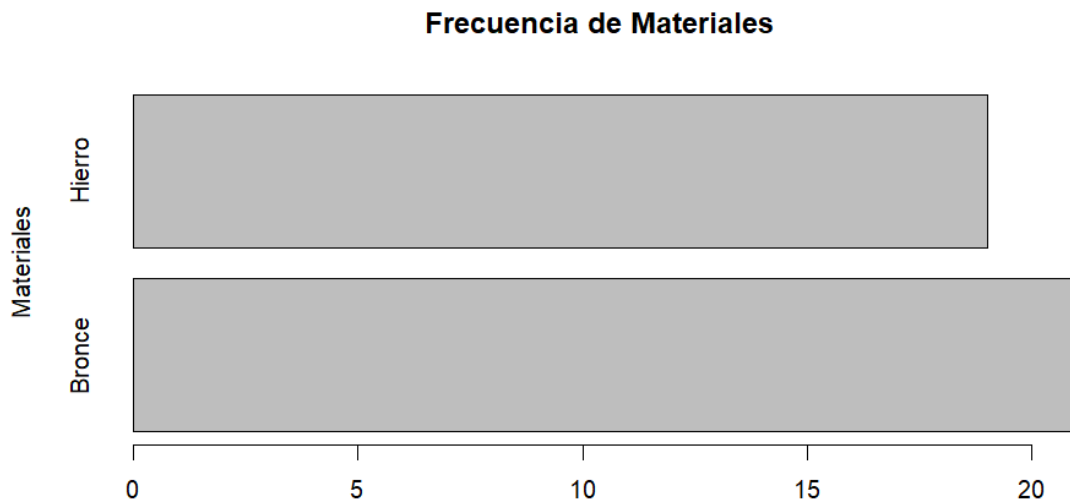


```
barras_verticales_contexto <- barplot(table(spear$Contexto), main =
"Frecuencia de Contexto",
xlab = "Contexto")
```

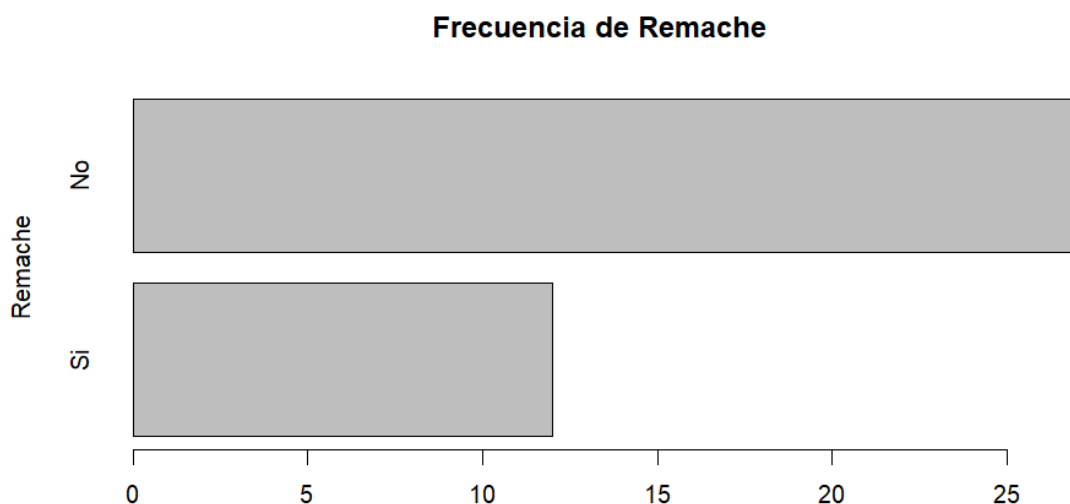


#Ej.9
 #Para elaborar estos gráficos utilizamos la misma función que en Ej.8, para generar gráficos de barras, pero añadimos la función "horiz" = True para que los haga horizontales. El resto es idéntico al ejercicio 8.

```
barras_horizontales_materiales <- barplot(table(spear$Materiales),
horiz = TRUE, main = "Frecuencia de Materiales", ylab = "Materiales")
```

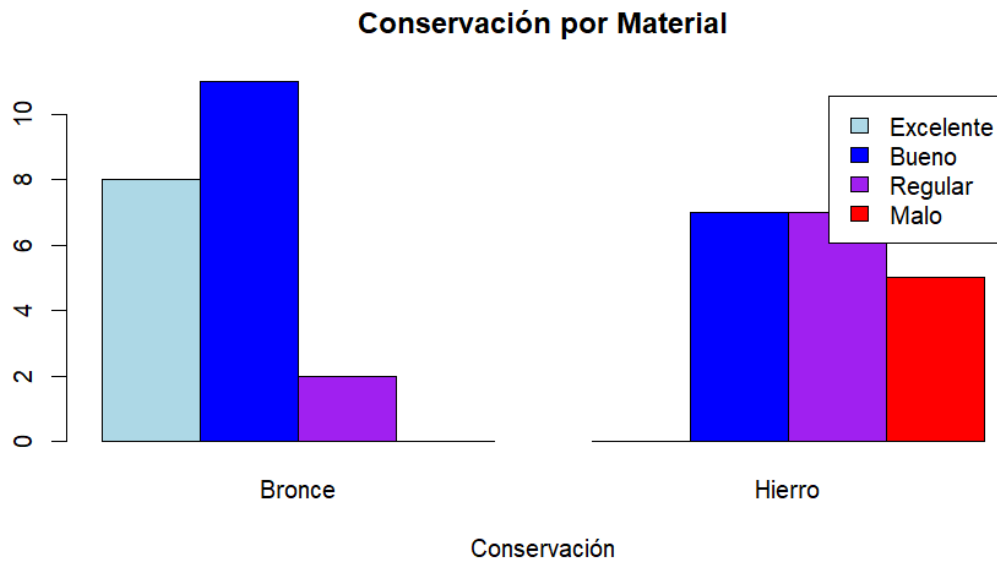


```
barras_horizontales_remache <- barplot(table(spear$Remache), horiz =
TRUE, main = "Frecuencia de Remache", ylab = "Remache")
```



#Ej.10
 #En este ejercicio empleamos la misma función de "barplot", que nos permite generar gráficos de barras y seleccionamos las dos columnas de los datos que queremos agrupar, e incluimos una leyenda para el título, para el eje X y además en este caso definimos los colores que queremos para las barras, empleando la función "col" y mediante el empleo de una igualdad, para asignarle a cada variable un color.

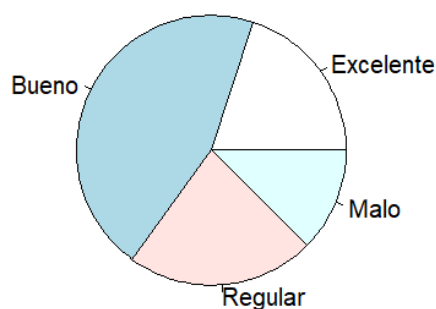
```
barras_material_conservacion <- barplot(table(spear$Conservacion,
spear$Materiales), beside = TRUE, legend = TRUE, main = "Conservación
por Material", xlab = "Conservación", col = c("lightblue",
"blue", "purple", "red"))
```

#Ej.11
 #En este caso para generar un gráfico de sectores usamos la función "pie", y utilizamos la misma función \$, para seleccionar la columna de datos que nos interesa. Aquí también definimos un título para la gráfica y en este caso nos quedamos con los colores predeterminados, aunque también podríamos seleccionarlos nosotros.

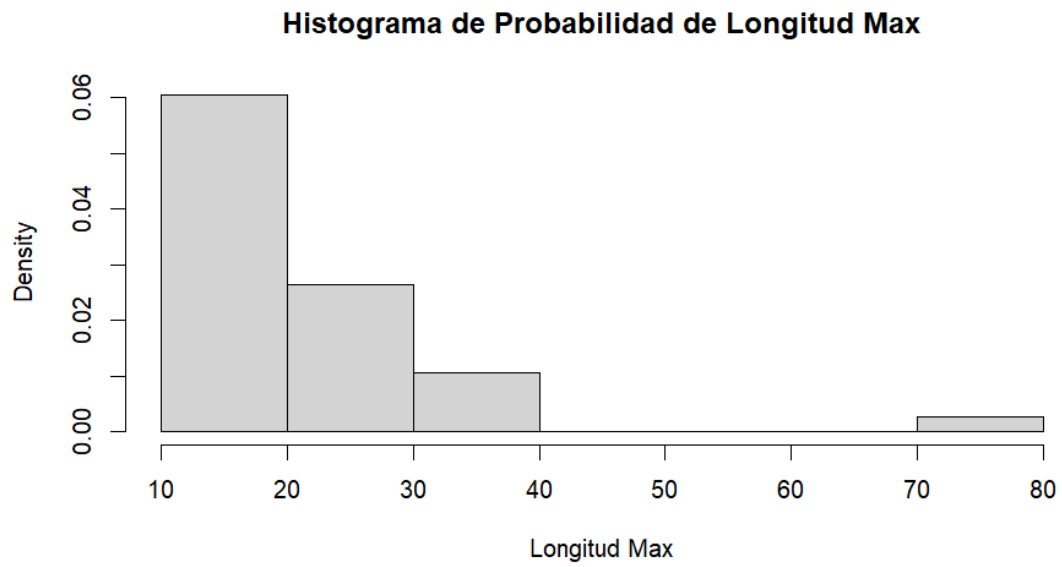
```
sectores_conservacion <- pie(table(spear$Conservacion), main =
"Porcentaje de Conservación")
```

Porcentaje de Conservación



#Ej.12
 #En el último ejercicio empleamos la función "hist" para generar un histograma, con dos variables continuas como es el ancho y la longitud máximos que son variables continuas. Usamos la función "prob" que nos va a permitir calcular las probabilidades.

```
hist(spear$Longitud_max, prob = TRUE, main = "Histograma de
Probabilidad de Longitud Max", xlab = "Longitud Max")
```



```
hist(spear$Ancho_max, prob = TRUE, main = "Histograma de Probabilidad  
de Ancho Max", xlab = "Ancho Max")
```

