

Análisis Energy Drink

Frank Casanova

25/1/2022

Cargamos los datos

Cargamos los datos desde la carpeta donde se encuentra el dataset con la función `read.table()`, mostramos información sobre los datos con la función `str()` y mostramos los primeros 5 datos con la función `head()`.

```
Beb_Energ = read.table("../data/EnergyDrink", header = TRUE)
str(Beb_Energ)
```

```
'data.frame':  122 obs. of  3 variables:
 $ estudio: chr  "Informatica" "Mates" "Industriales" "Informatica" ...
 $ bebe   : chr  "No" "No" "Si" "Si" ...
 $ sexo   : chr  "Mujer" "Hombre" "Mujer" "Hombre" ...
```

```
head(Beb_Energ)
```

	estudio	bebe	sexo
1	Informatica	No	Mujer
2	Mates	No	Hombre
3	Industriales	Si	Mujer
4	Informatica	Si	Hombre
5	Industriales	No	Mujer
6	Mates	No	Mujer

Con la función `summary()` mostramos las frecuencias absolutas de cada una de las variables del dataset.

```
summary(Beb_Energ)
```

estudio	bebe	sexo
Length:122	Length:122	Length:122
Class :character	Class :character	Class :character
Mode :character	Mode :character	Mode :character

Podemos crear tablas que nos presenten la información en función de sus variables con la función `table()`.

En este caso se muestran información en relación al sexo y la carrera de estudio.

```
table(Beb_Energ$estudio, Bebe_Energ$sexo)
```

	Hombre	Mujer
Industriales	25	12
Informatica	37	16
Mates	9	7
Telematica	12	4

Acá se muestra informacion acerca de la acreea de estudio, el sexo y como tercer parámetro si la persona bebe o no.

```
table(Beb_Energ$estudio, Beb_Energ$sexo, Beb_Energ$bebe)
```

, , = No

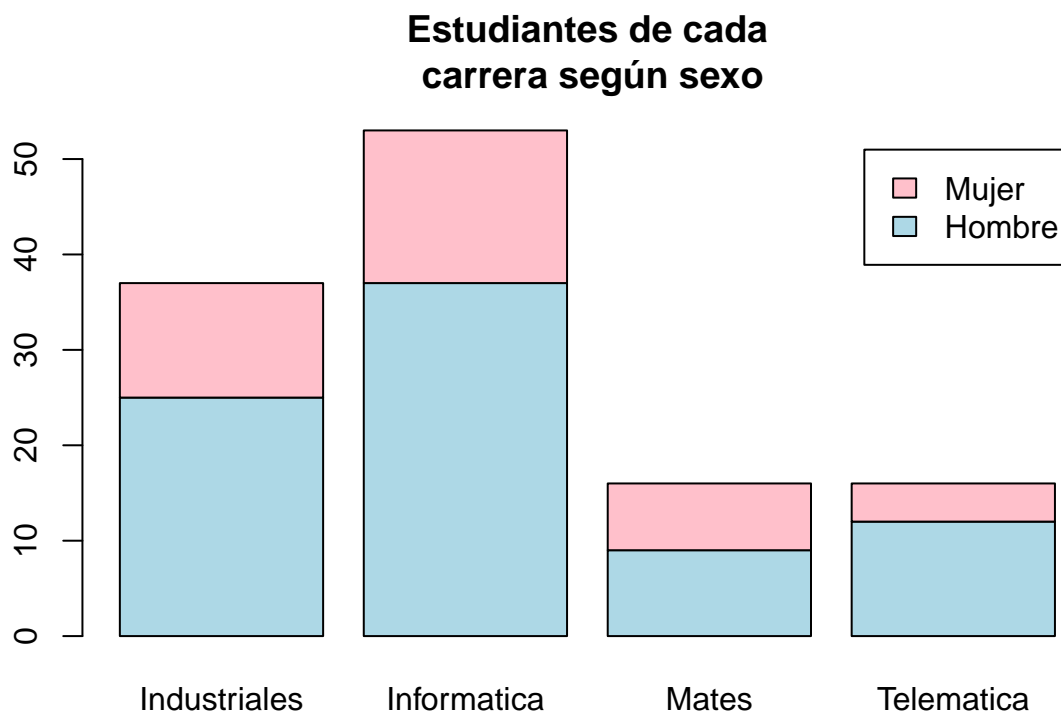
	Hombre	Mujer
Industriales	19	10
Informatica	30	11
Mates	8	6
Telematica	10	3

, , = Si

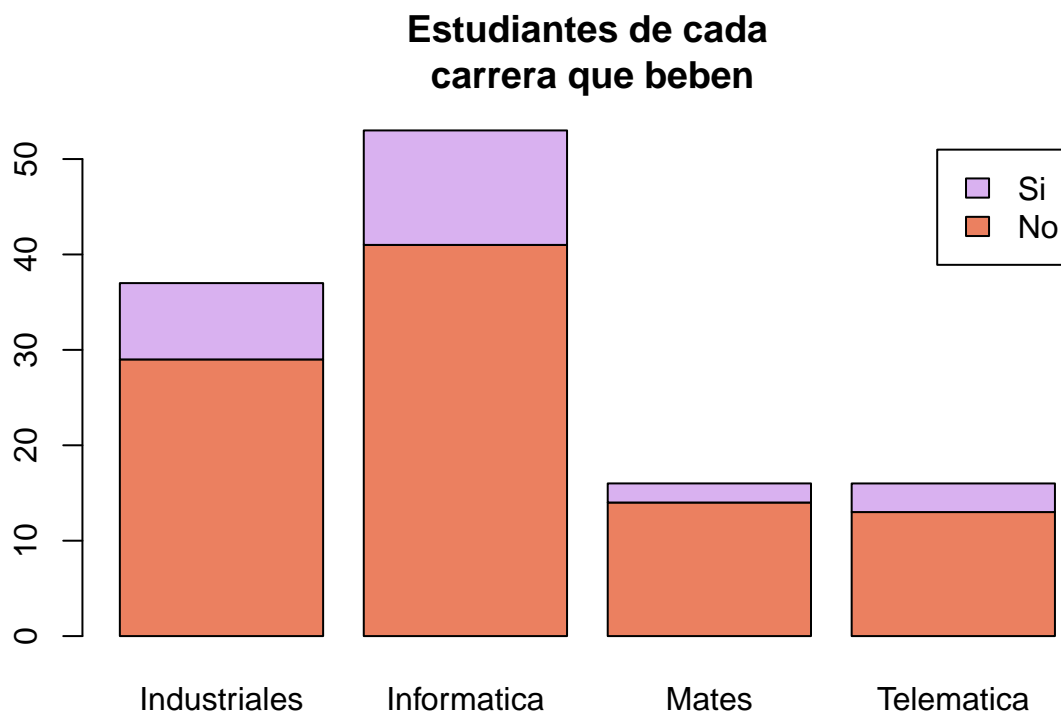
	Hombre	Mujer
Industriales	6	2
Informatica	7	5
Mates	1	1
Telematica	2	1

Probando barplot()

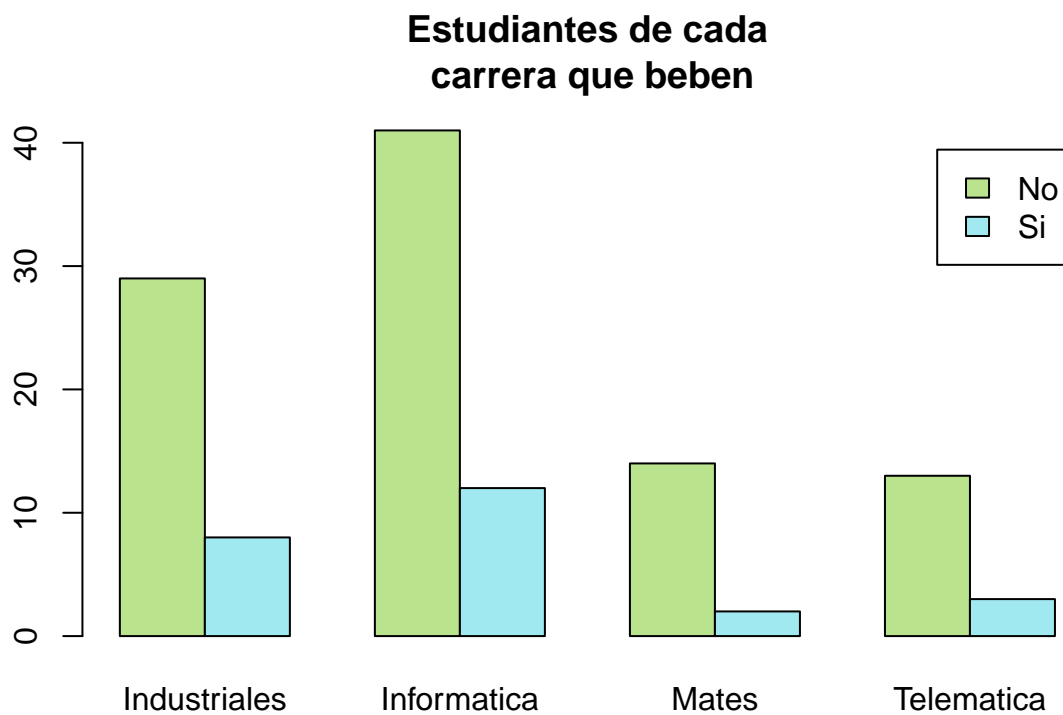
```
barplot(table(Beb_Energ$sexo, Beb_Energ$estudio),
        col=c('lightblue', 'pink'),
        main="Estudiantes de cada\n carrera según sexo",
        legend.text=TRUE )
```



```
barplot(table(Beb_Energ$bebe, Bebe_Energ$estudio),  
        col=c("#eb8060", "#d9b1f0"),  
        main="Estudiantes de cada\n carrera que beben",  
        legend.text=TRUE )
```



```
barplot(table(Beb_Energ$bebe, Beb_Energ$estudio),  
        col=c("#b9e38d", "#a1e9f0"),  
        main="Estudiantes de cada\n carrera que beben",  
        legend.text=TRUE, beside=T )
```



Análisis siguiendo el ejemplo de Joanby

Ejemplo final

```
fable(HairEyeColor)
```

Juntar color de ojos y pelo sin distinguir por sexo

		Sex	Male	Female
Hair	Eye			
Black	Brown		32	36
	Blue		11	9
	Hazel		10	5
	Green		3	2
Brown	Brown		53	66
	Blue		50	34
	Hazel		25	29
	Green		15	14
Red	Brown		10	16

	Blue	10	7
	Hazel	7	7
	Green	7	7
Blond	Brown	3	4
	Blue	30	64
	Hazel	5	5
	Green	8	8

```
male = HairEyeColor[, , "Male"]
female = HairEyeColor[, , "Female"]
datos = as.table(male+female)
datos
```

	Eye			
Hair	Brown	Blue	Hazel	Green
Black	68	20	15	5
Brown	119	84	54	29
Red	26	17	14	14
Blond	7	94	10	16

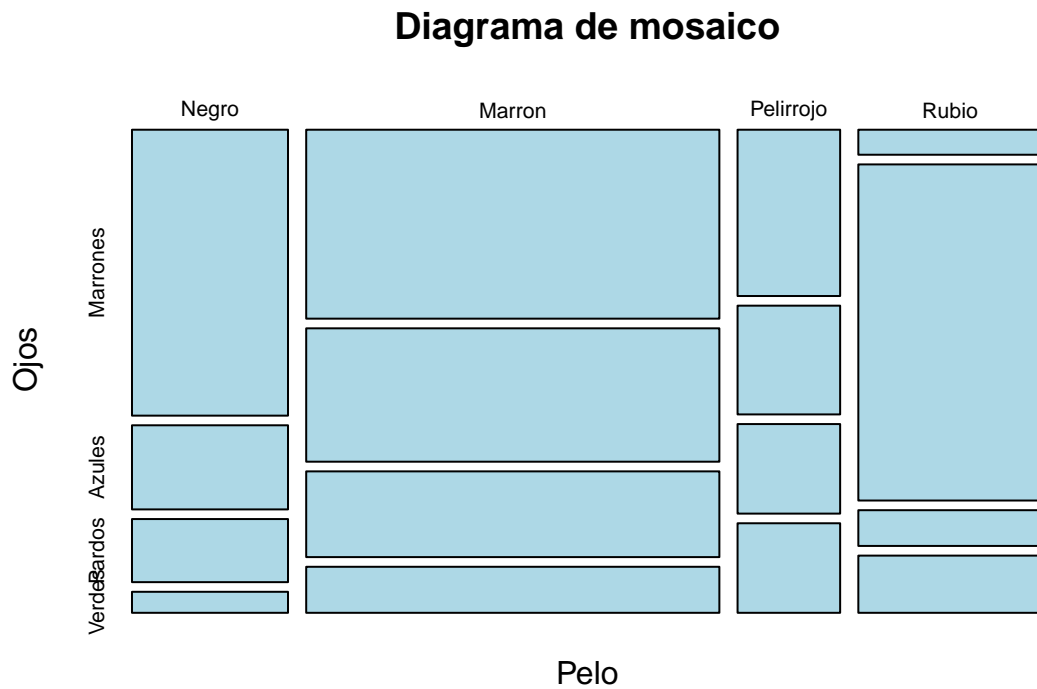
Manipulacion de datos

```
dimnames(datos) = list(
  Pelo = c("Negro", "Marron", "Pelirrojo", "Rubio"),
  Ojos = c("Marrones", "Azules", "Pardos", "Verdes"))
datos
```

	Ojos			
Pelo	Marrones	Azules	Pardos	Verdes
Negro	68	20	15	5
Marron	119	84	54	29
Pelirrojo	26	17	14	14
Rubio	7	94	10	16

Representacion gráfica

```
mosaicplot(datos, col = "lightblue", main = "Diagrama de mosaico")
```



Datos numéricos

Pasamos a calcular el numero total de individuos

```
sum(datos)
```

```
[1] 592
```

```
colSums(datos)
```

```
Marrones  Azules  Pardos  Verdes
      220    215    93    64
```

```
rowSums(datos)
```

```
 Negro  Marron Pelirrojo  Rubio
    108    286      71    127
```

```
round(prop.table(colSums(datos)),3)
```

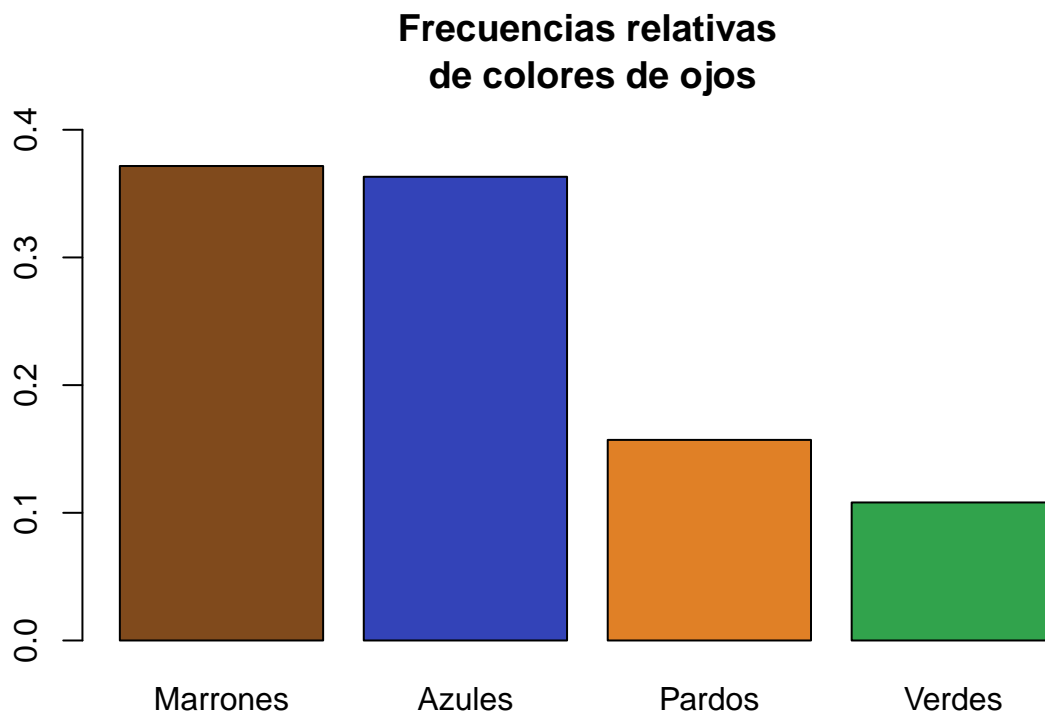
Marrones	Azules	Pardos	Verdes
0.372	0.363	0.157	0.108

```
round(prop.table(rowSums(datos)),3)
```

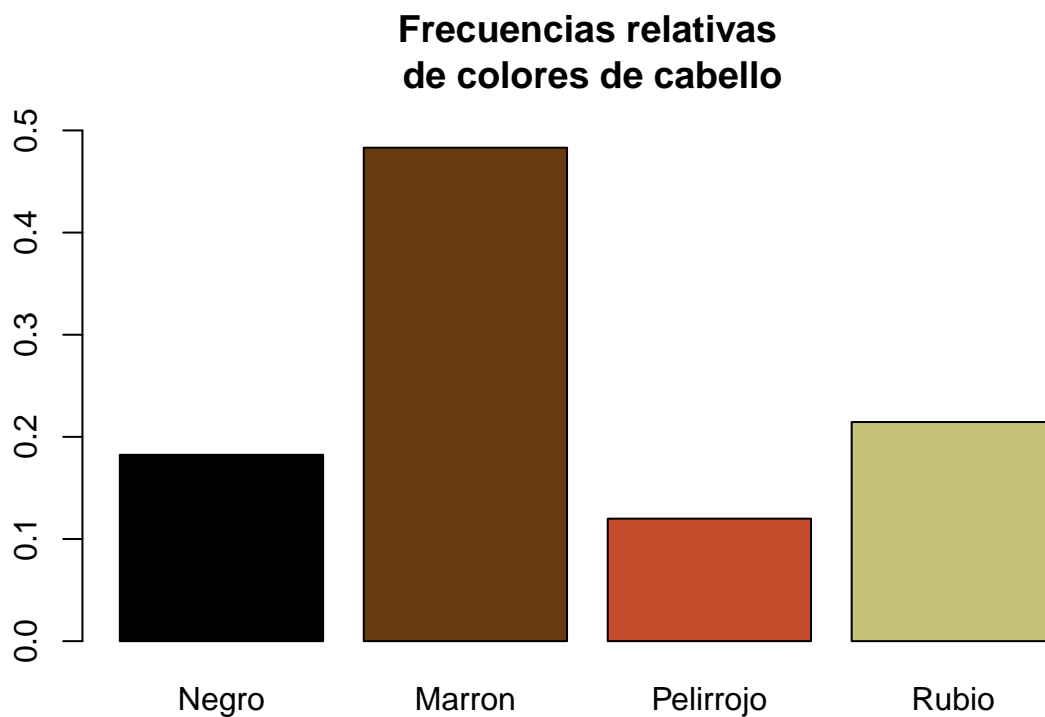
Negro	Marron	Pelirrojo	Rubio
0.182	0.483	0.120	0.215

Representacion gráfica

```
barplot(prop.table(colSums(datos)),
        col = c("#804a1c", "#3343b8", "#e08026", "#31a34c"),
        main = "Frecuencias relativas\n de colores de ojos",
        ylim = c(0,0.4))
```



```
barplot(prop.table(rowSums(datos)),
        col = c("#000000", "#693a0f", "#c24c29", "#c4c278"),
        main = "Frecuencias relativas\n de colores de cabello",
        ylim = c(0,0.5))
```

Frecuencias relativas globales y marginales

```
round(prop.table(datos), 3)
```

Pelo	Ojos			
	Marrones	Azules	Pardos	Verdes
Negro	0.115	0.034	0.025	0.008
Marron	0.201	0.142	0.091	0.049
Pelirrojo	0.044	0.029	0.024	0.024
Rubio	0.012	0.159	0.017	0.027

```
round(prop.table(datos, margin = 1), 3)
```

Pelo	Ojos			
	Marrones	Azules	Pardos	Verdes
Negro	0.630	0.185	0.139	0.046
Marron	0.416	0.294	0.189	0.101
Pelirrojo	0.366	0.239	0.197	0.197
Rubio	0.055	0.740	0.079	0.126

```
round(prop.table(datos, margin = 2), 3)
```

Pelo	Ojos			
	Marrones	Azules	Pardos	Verdes
Negro	0.309	0.093	0.161	0.078
Marron	0.541	0.391	0.581	0.453
Pelirrojo	0.118	0.079	0.151	0.219
Rubio	0.032	0.437	0.108	0.250

Calculando todas las frecuencias con la funcion `crosstable()`.

```
library(gmodels)
CrossTable(datos)
```

```
Cell Contents
|-----|
|              N |
| Chi-square contribution |
|      N / Row Total |
|      N / Col Total |
|      N / Table Total |
|-----|
```

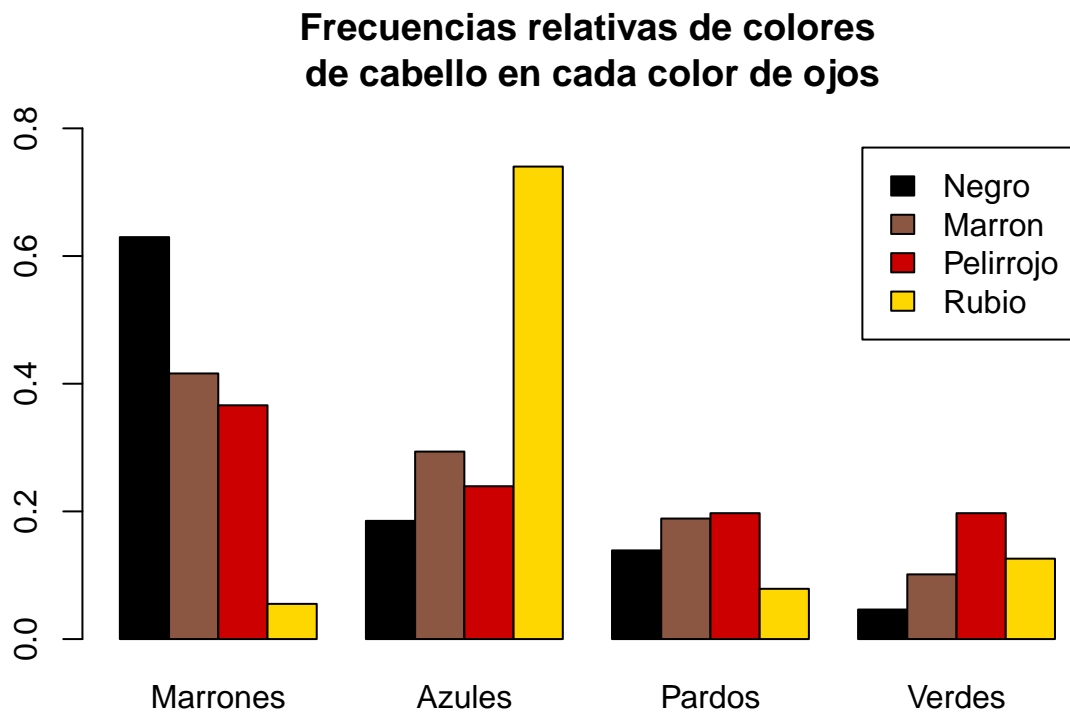
Total Observations in Table: 592

Pelo	Ojos				Row Total
	Marrones	Azules	Pardos	Verdes	
Negro	68	20	15	5	108
	19.346	9.421	0.228	3.817	
	0.630	0.185	0.139	0.046	0.182
	0.309	0.093	0.161	0.078	
	0.115	0.034	0.025	0.008	
Marron	119	84	54	29	286
	1.521	3.800	1.831	0.119	
	0.416	0.294	0.189	0.101	0.483
	0.541	0.391	0.581	0.453	
	0.201	0.142	0.091	0.049	
Pelirrojo	26	17	14	14	71
	0.006	2.993	0.726	5.211	
	0.366	0.239	0.197	0.197	0.120
	0.118	0.079	0.151	0.219	
	0.044	0.029	0.024	0.024	
Rubio	7	94	10	16	127
	34.234	49.697	4.963	0.375	
	0.055	0.740	0.079	0.126	0.215
	0.032	0.437	0.108	0.250	
	0.012	0.159	0.017	0.027	

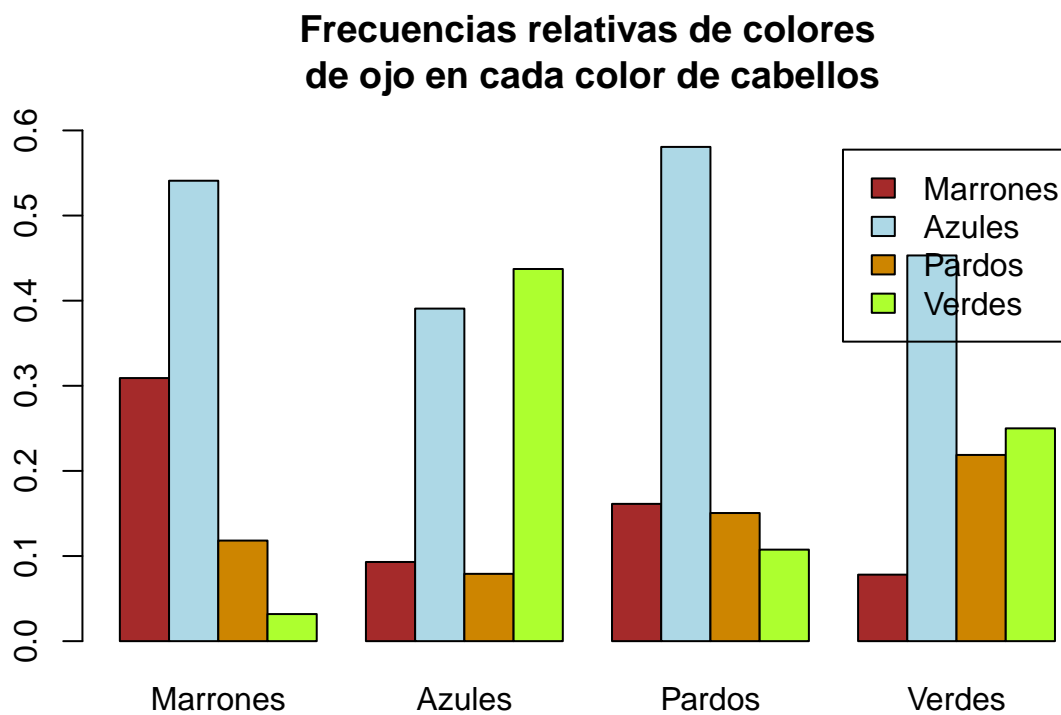
Column Total	220	215	93	64	592
	0.372	0.363	0.157	0.108	
-----	-----	-----	-----	-----	-----

Representacion gráfica de frecuencias relativas

```
barplot(prop.table(datos, margin = 1),
        main="Frecuencias relativas de colores\n de cabello en cada color de ojos",
        legend.text = T,
        beside = T,
        col = c("black", "lightsalmon4", "red3", "gold"),
        ylim = c(0,0.8))
```



```
barplot(prop.table(datos, margin = 2),
        main="Frecuencias relativas de colores\n de ojo en cada color de cabellos",
        legend.text = c("Marrones", "Azules", "Pardos", "Verdes"),
        beside = T,
        col = c("brown", "lightblue", "orange3", "greenyellow"),
        ylim = c(0,0.6))
```



Otra posibles solucion apra la elaboracion del segundo gráfico de frecuencias relativas y así evitar digitar la leyenda, podemos hallar la transpuesta `t()` de la `prop.table()`.

```
barplot(t(prop.table(datos, margin = 2)),
        main="Frecuencias relativas de colores\n de ojo en cada color de cabellos",
        legend.text = T,
        beside = T,
        col = c("brown", "lightblue", "orange3", "greenyellow"),
        ylim = c(0,0.6))
```

**Frecuencias relativas de colores
de ojo en cada color de cabellos**

