Estadístca descriptiva

Ingeniería comercial

Joaquin Cavieres G

1. Introducción

- Para realizar un correcto análisis de datos nosotros necesitamos tener una mezcla de conocimientos estadísticos y del problema que queremos resolver. No existen unas pautas constantes para obtener unos resultados óptimos ya que cada base de datos es un problema diferente. Sin embargo, como tarea inicial, debemos estudiar la estructura de nuestros datos, hacer un análisis descriptivo, crear gráficos y resuménes númericos para así generar un reporte con la información relevante que estamos estudiando.
- Dado lo anterior es que el análisis exploratorio es fundamental en cualquier estudio que contemple el análisis de datos.
- Este análisis inicial proporciona una idea de la distribución de nuestras observaciones, cálcular estadísticos de tendencia central (media, mediana y moda), de dispersión (varianza, desviación típica, rango), de forma (asimetría, curtosis), de posición (percentiles), así como la creación de gráficos y/o histogramas.
- El software estadístico R contiene una serie de librerías que nos ayudan a desarrollar este anásis sencillo de nuestros datos.

2. Estadística descriptiva

2.1. Tabla de frecuencias

Para crear tablas de frecuencia en R se hace mediante la función `table` o la función `prop.table`. La línea de código es la siguiente:

- `table(x)` para frecuencias absolutas
- `prop.table(tab)` para las frecuencias relativas

La diferencia entre una y otra es el argumento dado para la función:

`table()` construye la tabla de frecuencias absolutas a partir de la variable que recibe como argumento, en cambio, `prop.table()` recibe como argumento una tabla o una matriz que representa una tabla de frecuencias absolutas, y a partir de ella construye la tabla de frecuencias relativas asociada. Es decir, `prop.table()` recibe como argumento el resultado que devuelve la función `table()`.

La tabla contiene 14 observaciones sobre el peso, altura, edad, sexo y nombres de cada persona.

```
# Calculamos las frecuencias absolutas y relativas de las variables peso y nombre
tabla_peso <- table(data1$Peso)
tabla_peso
```

47 54 55 58 60 65 75 77 82 85 89 1 1 1 2 1 2 2 1 1 1 1

Calculamos la frecuencia relativa

```
prop.table(tabla_peso)
```

```
    47         54         55         58         60         65         75
```

0.07142857 0.07142857 0.07142857 0.14285714 0.07142857 0.14285714 0.14285714 77 82 85 89 0.07142857 0.07142857 0.07142857 0.07142857

2.2. Gráficos descriptivos

Generalmente dentro de una base de datos nosotros contamos con:

a) Variables cualitativas o variables cuantitativas de tipo discreto: Se pueden considerar gráficos de sectores o gráficos de barras, los cuales se obtienen en R mediante las funciones pie y barplot, respectivamente. Los argumentos más importantes de estas funciones son: `pie(x, labels = names(x), clockwise = FALSE, init.angle = if(clockwise) 90 else 0, col = NULL, main = NULL)`

`barplot(x, horiz = FALSE, height, col = NULL, width space, names.arg, beside, main = NULL, sub = NULL, xlab = NULL, ylab = NULL)`

donde

`x` es un vector con las frecuencias de las observaciones. Igualmente, puede ser una tabla de frecuencia (de las obtenidas con table o prop.table)

`labels` es un vector de cadenas de caracteres que indican los nombres de cada una de las categorías que aparecen en el gráfico de sectores

`clockwise` es un argumento lógico que indica si los sectores se dibujan en sentido horario (clockwise = TRUE) o en sentido antihorario (clockwise = FALSE, que es la opción por defecto).

`init.angle` es un valor numérico que indica el ángulo (en grados) en el que se sitúa el primer sector. Por defecto, el primer sector empieza a dibujarse a los 90 grados (- a las 12 en punto -, cuando clockwise es igual a TRUE) o a los 0 grados (- a las 3 en punto -, cuando clockwise es igual a FALSE)

`horiz` es un argumento lógico que indica si las barras del gráfico de barras se dibujan de forma vertical (horiz = FALSE, que es la opción por defecto) u horizontal (horiz = TRUE)

`height`: vector de frecuencias para cada valor

`width`: especifica mediante un vector el ancho de las barras

`space`: fija el espacio entre las barras

`names.arg`: vector de nombres para colocarlos bajo las barras

`beside`: valor lógico, FALSE indica barras apiladas y TRUE yuxtapuestas

`col`: se indica los colores de las barras o los sectores del gráfico

`main` y `sub`: son cadenas de caracteres en la que se especifican el título y el subtítulo del gráfico

`xlab` e `ylab`: son cadenas de caracteres en las que se especifican los nombres de los ejes X e Y.

`NULL` indica que no se da ningun valor ese carácter en especial.

b) Variables cuantitativas: Los gráficos que se suelen emplear con más frecuencia son el histograma, el diagrama de tallos y hojas y el diagrama de caja y bigotes. En R, se utilizan las órdenes `hist` y `boxplot`

para la obtención de histogramas y de diagramas de caja y bigotes, respectivamente. Éstas son las principales opciones de estas funciones:

```
hist(x, breaks = "Sturges", freq=TRUE, right = TRUE, col = NULL, main = ("Histogram of"))

boxplot(x, range = 1.5, col = NULL, main = NULL)
```

donde, en este caso,

`x`: es el vector de valores de la variable a partir de los cuales se dibujará el gráfico.

`breaks`: indica la forma en la que se calcularán los intervalos en el histograma. Las opciones disponibles para este parámetro son "Sturges" (que es la opción por defecto) "Scott" y "FD" "Freedman-Diaconis".

`freq` si es TRUE determina que el intervalo se represente con las frecuencias absolutas

`range` es un valor numérico que determina la extensión de los bigotes de la caja. Para un valor positivo de range, los bigotes se extienden hasta el último dato que no supere 1.5 veces la longitud de la caja (el rango intercuartílico). Para un valor de 0, los bigotes se extienden hasta el dato más lejano

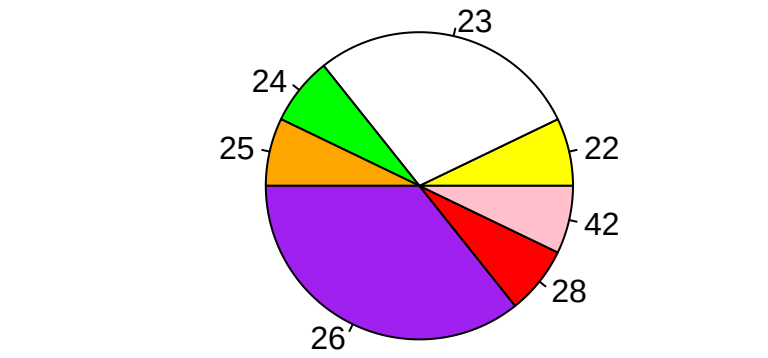
`right` es un argumento lógico que indica si los intervalos son cerrados por la izquierda y abiertos por la derecha (en cuyo caso, right = TRUE, que es la opción por defecto) o viceversa (right = FALSE).

`main` es un argumento para especificar el título dea la variable en el histograma

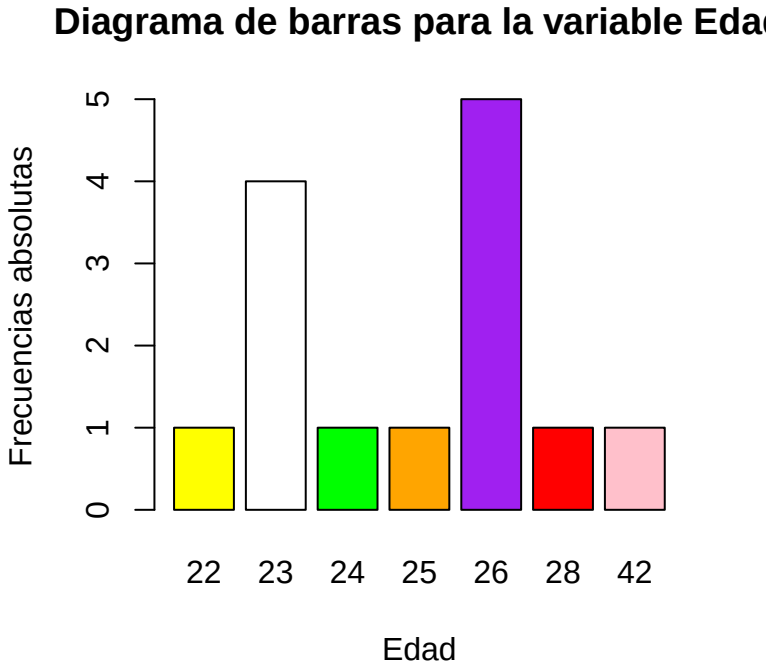
Las opciones `col` y `paste` funcionan igual que en los gráficos de barras y sectores.

Ejemplos

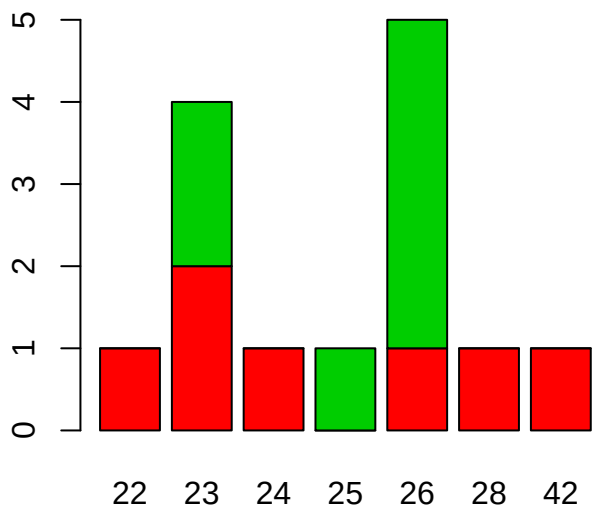
```
pie(table(data1$Edad), col = c("yellow", "white", "green", "orange", "purple", "red", "pink"),
main = "Diagrama de sectores para la variable Edad")
```



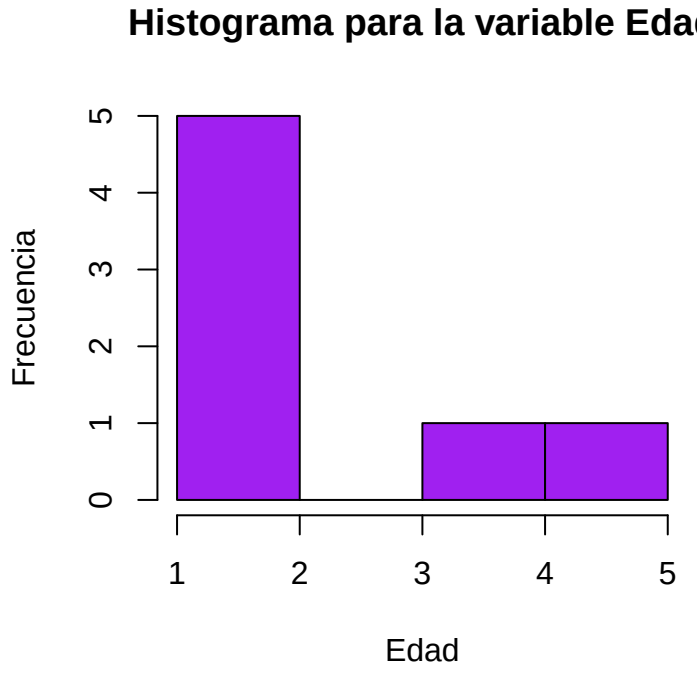

```
barplot(table(data1$Edad), col=c("yellow", "white", "green", "orange","purple", "red", "pink"),
xlab="Edad", ylab="Frecuencias absolutas", main ="Diagrama de barras para la variable Edad")
```



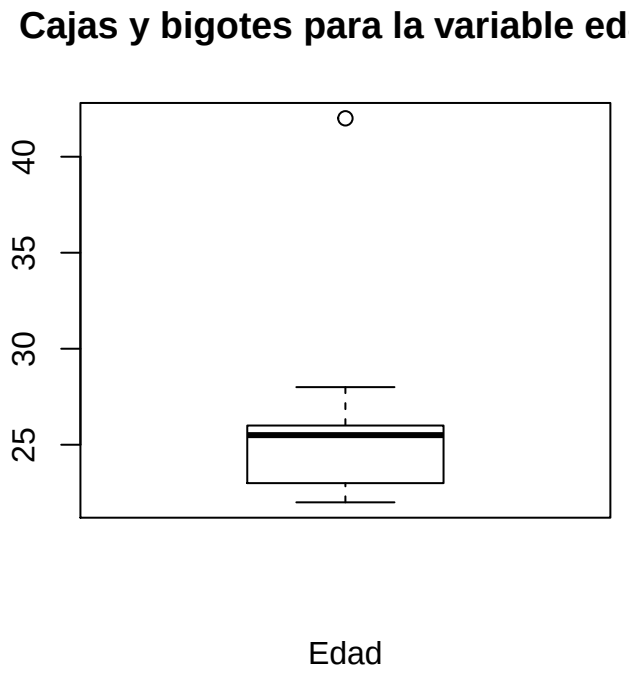
```r
barplot(table(data1$Sexo,data1$Edad),col=c(2,3))
```




```r
hist(table(data1$Edad), col = "purple", main = "Histograma para la variable Edad",
     xlab="Edad", ylab="Frecuencia")
```



```r
boxplot(data1$Edad, xlab="Edad", main = "Cajas y bigotes para la variable edad")
```



2.3. Estadísticos de resumen

2.3.1. Media, Mediana y Moda Los estadísticos de resumen hacen lo que la misma palabra describe, "resumen" las observaciones de cualquier variable número medida. Estos estadísticos entregan información relacionada a los puntos relevantes de la distribución que la variable tiene y complementan a la información descrita en los puntos anteriores.

a) Medidas de posición

Generalmente, las medidas de posición comunmente presentadas son 3: la media, la mediana y la moda, que son estadísticos de medida central en cualquier distribución de una variable número. En R se pueden utilizar directamente la media (`mean`)y la mediana `median`.

```r
mean(x, na.rm = FALSE)

median (x, na.rm = FALSE)
```

donde:

`x`: vector con los valores de la variable

`na.rm`: un argumento lógico que indica si hay que eliminar los valores faltantes del conjunto de datos.

Asignando el valor `TRUE` al argumento `na.rm` se pueden eliminar los valores faltantes (`NA`) para así calcular un valor para la media o la mediana, basado en las observaciones restantes (si exsiten `NA` y no se otorga `na.rm` = `TRUE` la función no dara el resultado de la media o mediana).

```r
mean(data1$Peso)
```

[1] 67.5

```r
median(data1$Peso)
```

[1] 65

No existe implementada una función para estimar la moda en R pero nosotros aquí podemos crea una para así calcularla

```r
moda <- function(x) {
ux <- unique(x)
ux[which.max(tabulate(match(x, ux)))]
}
```

Ya creada la función ahora podemos calcular la moda de la variable "Edad" en nuestros datos mediante la siguiente expresión:

```r
moda(data1$Edad)
```

[1] 26

Entre las medidas de posición de tendencia no central (como lo son la media, mediana o moda), los cuantiles figuran entre las más utilizadas para representar la información en las "colas" de nuestras observaciones. Para obtener los cuantiles en R se llama a la función `quantile`.

```r
quantile(x, probs = seq(0, 1, 0.25), na.rm = FALSE)
```

donde:

`x`: vector que incluye los valores de la variable númerica.

`seq`: Argumento que indica los cuantiles que se van a calcular. Por defecto, se muestran los siguiente cuantiles:

- 0: igual al valor mínimo
- 25: Coincide con el primer cuartil de los la distribución de los datos
- 50: Coincide con el segundo cuartil y con la mediana
- 75: Coincide con el tercer cuartil de los la distribución de los datos
- 100: Coincide con el valor máximo

`na.rm`: un argumento lógico que indica si hay que eliminar los valores faltantes del conjunto de datos.

```r
# Cuantiles para la variable Peso en nuestra base de datos
quantile = quantile(data1$Peso, probs = c(0.25, 0.75))
quantile
```

25% 75% 58.0 76.5

2.3.1. Varianza y desviación estándar Estas medidas cuantifican la variabilidad de las observaciones e indican la mayor o menor representatividad de las medidas de tendencia central (Media, Mediana y Moda). La varianza se puede cálcular mediante la función `var` y la desviación estándar mediante la función `sd`

```r
var(x, na.rm = FALSE)
```

```r
sd(x, na.rm = FALSE)
```

`x` es el vector de observaciones de la variable que se está estudiando y `na.rm` indica si los valores faltantes han de ser eliminados antes del análisis.

En estricto rigor, las funciones `var` y `sd` no calculan la varianza y la desviación estándar directamente, sino su cuasi-varianza y su cuasi-desviación estándar. En caso de necesitar la varianza o la desviación típica, basta con multiplicar el resultado de las funciones nombradas anteriormente por (n – 1)/n, siendo n el número total de datos con el que se está trabajando.

```
var(data1$Peso, na.rm = TRUE)  # cuasi-varianza de la variable Peso
```

[1] 168.4231

```
sd(data1$Peso, na.rm = TRUE)   # cuasi-desviación estándar de la variable Peso
```

[1] 12.97779

```
var_Edad <- 13/14 * var(data1$Edad) # Varianza de la variable edad
desvt_Edad <- sqrt(var_Edad)        # Desviación estándar de la variable edad

var_Edad
```

[1] 22.63776

```
desvt_Edad
```

[1] 4.757915