Práctica 3 - Aplicaciones Informáticas en Arqueología y Estadística

Álvaro Miranda García

2023-03-12

Para esta práctica, he tratado de crear la memoria junto al script utilizando Rmarkdown. El script, por su lado, se enviará también con el enlace de GitHub.

1. Crea un vector llamado 'numArtefactos' a partir de los siguientes valores referidos al número de artefactos por yacimiento: '17, 54, 10, 34, 90, 33, 49, 82, 12, 23, 56, 78, 44, 102, 10, 53, 4, 28, 37, 95'.

```
numArtefactos = c(17, 54, 10, 34, 90, 33, 49, 82, 12, 23, 56, 78, 44,
102, 10, 53, 4, 28, 37, 95)
```

¿Cómo almacena los valores numéricos: integer o double? Transforma el tipo de dato a número entero llamando al objeto 'numArtefactos_int'

En doble. Usando as.integer () podemos pasarlo a este formato, y con is.integer () se comprueba.

```
numArtefactos_int = as.integer (numArtefactos)
is.integer (numArtefactos_int)

## [1] TRUE
```

2. Calcula la media del objeto 'numArtefactos_int'. La función para calcular la media es mean (x). El resultado es 45.55. Creamos, además, un vector para poder trabajar con este valor más adelante.

```
mean (numArtefactos_int)

## [1] 45.55

media1 = mean (numArtefactos_int)
```

3. Calcula la mediana del objeto 'numArtefactos_int'. Define brevemente la mediana: concepto y cálculo. La función para calcular la mediana es median (x). El resultado es 40.5. Creamos, además, un vector para poder trabajar con este valor más adelante.

```
median (numArtefactos_int)

## [1] 40.5

mediana1 = median (numArtefactos_int)
```

4. Calcula la moda del objeto 'numArtefactos_int'. Explica detalladamente el procedimiento para su cálculo: empleo de funciones, operadores etc.

R no tiene función integrada para encontrar la moda, para esto creamos una función. A través de la función unique () se seleccionan los valores únicos. Tabulando se puede hacer un conteo de ocurrencias. Con match, se casa la posición de los valores en el vector de valores únicos con el vector con los valores indexados. Con == max (tab), se genera el valor con más ocurrencias dentro de los valores únicos. La moda, entonces, es 10, que se repite dos veces.

```
encontrar_moda = function (x) {

u = unique (x)
tab = tabulate (match(x, u))
u [tab == max (tab)]
}

moda1 = encontrar_moda (numArtefactos_int)

moda1 = 10
```

5. Calcula el número de veces que se repite el valor correspondiente con la moda.

```
table (numArtefactos_int)

## numArtefactos_int
##   4  10  12  17  23  28  33  34  37  44  49  53  54  56  78  82  90
95 102
##   1   2   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
1   1
```

Aquí vemos como el valor 10 aparece 2 veces.

6. Calcula los cuartiles del objeto 'numArtefactos_int'.

Los Cuantiles son unas medidas de posición que dividen a la distribución en un cierto número de partes, resultando en que en cada una de ellas hay el mismo de valores de la variable. Los cuartiles dividen a la distribución en cuatro partes iguales. Se calculan con la función quantile ().

```
quantnumArtefactos_int = quantile (numArtefactos_int)
quantnumArtefactos_int

##    0%   25%   50%   75%  100%
##   4.0  21.5  40.5  61.5 102.0
```

7. Calcula el rango intercuartílico del objeto 'numArtefactos_int'. Interpreta el resultado.

El rango intercuartílico de estos valores es un rango en el que se corta el 25% a cada lado. Estadísticamente, el rango intercuartílico es la diferencia entre el cuartil superior y el cuartil inferior. 40 es la diferencia entre 102 y 4.

```
IQR (quantnumArtefactos_int) #El resultado es 40

## [1] 40
```

8. Calcula el rango del objeto 'numArtefactos_int'. Almacena el rango en un vector denominado 'rango_artefactos'.

Con la función range () sacamos el valor más alto y el más bajo. El rango, entonces, es la diferencia entre estos dos valores, lo que nos da como resultado: 98

```
range (numArtefactos_int)

## [1]   4 102

rango_artefactos = max (numArtefactos_int) - min (numArtefactos_int)
```

9. Calcula la varianza del objeto 'numArtefactos_int'. Emplea 2 funciones para su cálculo.

```
var1 = var (numArtefactos_int) #927.1026

#Otra forma de calcular la varianza:

sd (numArtefactos_int) #desviación típica

## [1] 30.44836

sd (numArtefactos_int) ^2 #la varianza es también el cuadrado de la
desviacion tipica = 927.1026

## [1] 927.1026
```

10. Calcula la desviación estándar del objeto 'numArtefactos_int'. Emplea 2 funciones para su cálculo.

```
#Dos formas de calcular la desviación típica o estándar:

sd (numArtefactos_int) #30.44836

## [1] 30.44836

sqrt (var (numArtefactos_int)) #30.44836

## [1] 30.44836

#Creamos un vector para trabajar después con él

sd1 = sd(numArtefactos_int)
```

11. ¿En qué se diferencia la desviación estándar de la varianza?

Ambas vienen a medir lo mismo. El uso de una u otra depende de los cálculos que necesitemos y la comodidad para usar los datos, por ejemplo, para calcular la covarianza se necesita la varianza.

La desviación típica o estándar es la medida de la dispersión de una distribución de frecuencias respecto de su media. Equivale a la raíz cuadrada de la varianza.

Como definiciones, la desviación típica es una medida estadística que analiza la distancia que separa un grupo de números de la media, es decir, la desviación típica mide la distancia entre los números de un conjunto de datos.

La varianza es la desviación típica al cuadrado, o bien, la desviación típica es la raiz cuadrada de la varianza.

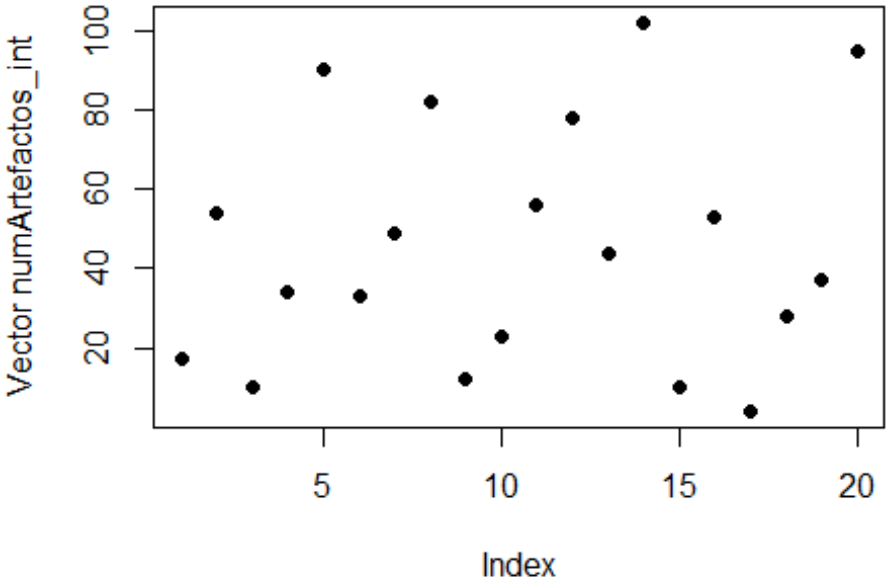
Las diferencias clave son:

- La desviación típica mide la distancia entre los números de un conjunto de datos. La varianza, por su parte, da un valor real a cuánto varían los números de un conjunto de datos con respecto a la media.
- La desviación típica se expresa en las mismas unidades que el conjunto de datos. La varianza puede expresarse en unidades al cuadrado o en porcentaje (usado mucho en el contexto de las finanzas).
- La desviación típica puede ser mayor que la varianza, ya que la raíz cuadrada de un decimal es mayor (y no menor) que el número original cuando la varianza es menor que uno.
- La desviación típica es menor que la varianza cuando esta es mayor que uno.

12. Visualiza gráficamente de manera horizontal la dispersión del objeto 'numArtefactos_int'.

Con la función plot () se puede representar la dispersión. Podemos establecer etiquetas de los ejes con xlab e ylab, al igual que con gráficos de barras. Con pch determinamos los símbolos que aparecen, siendo el 19 un círculo. Con col se establece el color.

```
plot(numArtefactos_int, xlim =, xlab = "Index", ylab = "Vector
numArtefactos_int", pch = 19, col = "black")
```



13. Crea un vector llamado 'vector3' a partir de la siguiente secuencia de valores: '21, 45, 33, 98, 34, 90, 67, 87, 45, 11, 73, 38, 28, 15, 50, 57, 12, 87, 29, 1'

Lo hacemos de la misma manera que con numArtefactos. Asimismo, generamos vectores para los estadísticos descriptivos de vector3.

```
vector3 = c (21, 45, 33, 98, 34, 90, 67, 87, 45, 11, 73, 38, 28, 15, 50,
57, 12, 87, 29, 1)

View (vector3)

vector3 = as.integer (vector3)

is.integer(vector3)

## [1] TRUE

media2 = mean (vector3)
mediana2 = median (vector3)
range (vector3)

## [1]  1 98

rango_vector3 = max (vector3) - min (vector3)
var2 = var (vector3)
sd2 = sd(vector3)
```

14. Calcula el coeficiente de variación de los objetos: 1)'numArtefactos_int' y 2) 'vector3'. Emplea 2 funciones para su cálculo. Compara e interpreta los resultados.

El coeficiente de variación (CV) es la relación entre la desviación estándar y la media.

```
#Primera forma de hacerlo:
cv_numart <- sd(numArtefactos_int) / mean(numArtefactos_int) * 100
View (cv_numart) #66.84602

cv_v3 <- sd(vector3) / mean(vector3) * 100
View (cv_v3) #63.59067


#Segunda forma: creamos data frame
coefvar <- data.frame (a =numArtefactos_int,
                       b = vector3)

#calculamos CV para cada columna en el dataframe
sapply (coefvar, function (x) sd (x) / mean (x) * 100 )

##        a        b
## 66.84602 63.59067

#a        b
#66.84602 63.59067
```

Esto nos indica que el CV de numArtefactos es 66.8%, y el de Vector3, 63.5%. De acuerdo con estos datos, numArtefactos tiene, aunque por poco, mayor dispersión que Vector3.

15. Genera una tabla-resumen de los estadísticos descriptivos expuestos: media, mediana, desviación estándar etc.

Creamos dataframes con los estadísticos creados para cada vector, después, creamos otro df combinando los dos que tenemos; cambiamos los nombres de las filas y las columnas, y ya tenemos una tabla-resumen.

```
df1 = data.frame(estadisticos_numart = c(media1, mediana1,
rango_artefactos, var1, sd1, cv_numart))
df2 = data.frame(estadisticos_v3 = c(media2, mediana2, rango_vector3,
var2, sd2, cv_v3))


dataf = data.frame(df1, df2)

row.names(dataf) = c ("Media", "Mediana", "Rango", "Varianza","Desviacion
Estandar", "Coeficiente de Variacion")
colnames(dataf) = c("NumArtefactos", "Vector3")
```

```
View (dataf)

#Aquí, en RMarkdown, podemos instalar el package DT, entre otros, para
crear tablas en base a dataframes. En mi caso, instlado el package DT (en
la consola), y con la función datatable (df) se genera una tabla, que
además es interactiva.

library (DT)
datatable (dataf)
```

16. Calcula el coeficiente de asimetría del objeto 'vector3'. Interpreta su resultado. Exponga ejemplos de distribuciones de variables con asimetría positiva y negativa y simétricas. Explique cada uno de estos escenarios.

La Asimetría es la Simetría con respecto a su media. Mediante el coeficiente de asimetría (CA) podemos tener un valor numérico de este fenómeno, sin que sea necesaria una representación gráfica, como un histograma.

Para calcular el coeficiente de asimetría, primero instalamos el package moments en la consola, que contiene la función skewness (), con la cual calculamos el coeficiente de asimetría, con 0.3389539 como resultado.

Hay tres tipos de curva de distribución según su asimetría:

Asimetría negativa: la cola de la distribución se alarga para valores inferiores a la media. Simétrica: hay el mismo número de elementos a izquierda y derecha de la media. Aquí coinciden la media, la mediana y la moda. La distribución simétrica es lo que conocemos como la campana de Gauss, o distribución normal. Asimetría positiva: la cola de la distribución se alarga (a la derecha) para valores superiores a la media.

Si el CA = <0: la distribución tiene una asimetría negativa. Si el CA = 0: la distribución es simétrica. Si el CA = >0: la distribución tiene una asimetría positiva.

En nuestro caso, el CA es 0.3389539; es mayor que 0. Por tanto, la distribución tiene una asimetría positiva y se alarga a valores mayores que la media.

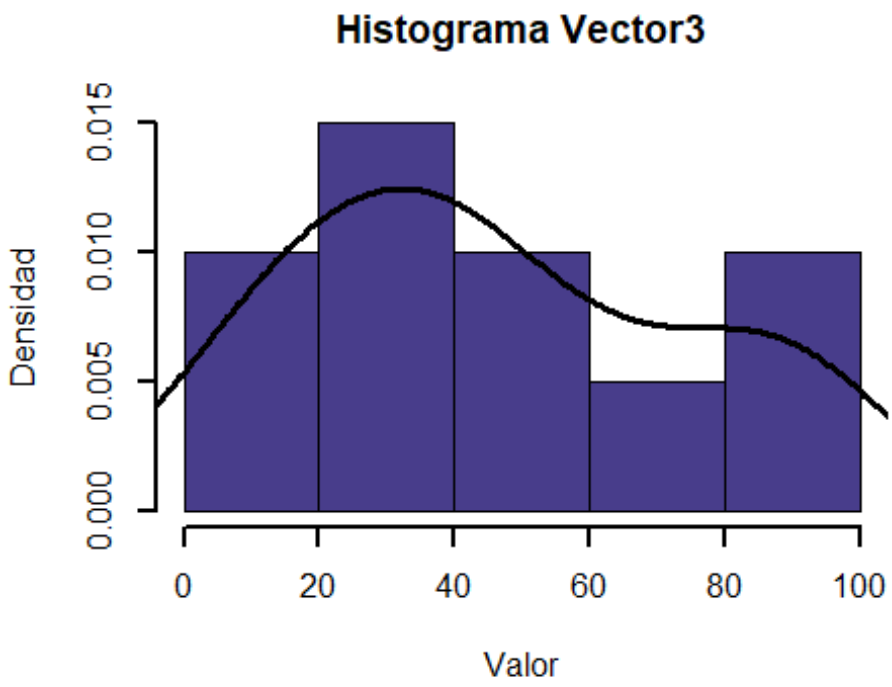
```
library(moments)

skewness(vector3) #0.3389539

## [1] 0.3389539

#Podemos crear un histograma y comprobar visualmente como la distribución
se alarga hacia valores mayores que la media

DV3 <- density(na.omit (vector3))
hist(vector3, prob = TRUE, main = "Histograma Vector3", col =
"darkslateblue", ylab = "Densidad", xlab = "Valor", lty =1, lwd = 2)
lines (DV3, lwd = 3, col = "black")
```



17. Calcula la curtosis del objeto 'vector3'. ¿Qué tipo de curtosis se encuentra asociada al anterior objeto? Justifica tu respuesta.

La Curtosis(Apuntamiento) mide cómo de achatada o apuntada es la curva y cómo se agrupan valores en torno a la media.

El valor obtenido indica la cantidad de datos que hay cercanos a la media, de manera que a mayor grado de curtosis, más escarpada (o apuntada) será la forma de la curva.

Se calcula con la función kurtosis (), dándonos 1.952376.

Una curtosis positiva indica una distribución relativamente elevada (leptocúrtica), mientras que una curtosis negativa indica una distribución relativamente plana (platicúrtica). Por su lado, la distribución normal es llamada mesocúrtica.

La interpretación de la curtosis es similar al CA. Podemos tomar 0 como la curtosis o apuntamiento de una distribución normal (realmente la curtosis en un distribución normal es 3) . Si la curtosis es menor que 0, esta es menos apuntada que la normal. Si es mayor que 0, es más apuntada.

En nuestro caso, al ser 1.952376, es una curtosis positiva, más apuntada que la distribución normal.

```
kurtosis (vector3) #1.952376
```

```
## [1] 1.952376
```

Como bonus, a los conjuntos de datos podemos realizar tests de normalidad como el de Jarque-Bera, para comprobar si los datos de la muestra tienen asimetría y curtosis que coinciden con una distribución normal. Se hace con la función jarque.test ().

```
jarque.test(vector3)

##
##  Jarque-Bera Normality Test
##
## data:  vector3
## JB = 1.2976, p-value = 0.5227
## alternative hypothesis: greater

#Jarque-Bera Normality Test

#data:  vector3
#JB = 1.2976, p-value = 0.5227
#alternative hypothesis: greater
```

Esto indica que es probable que tengamos una hipótesis alternativa (el conjunto de datos tiene una asimetría y una curtosis que no coincide con una distribución normal), y no una hipótesis nula (asimetría y una curtosis que coincide con una distribución normal). Sin embargo, se necesitarían más evidencias.