Ejercicios Estadística en R

Mario Pascual González

```
knitr::opts_chunk$set(fig.align='center', fig.width=6, fig.height=4, out.width='70%')
```

Funciones

Aquí incluyo todas las funciones que he ido desarrollando a lo largo de los ejercicios.

```
# Funciones

# Función que calcula las frecuencias absolutas dados unos datos crudos
calc_ni <- function(datos_in, xi_in) {
  ni <- sapply(xi_in, function(valor) {
    sum(datos_in == valor)
  })
  return(ni)
}

# Función que calcula las frecuencias relativas de cada dato
calc_fi <- function(datos_in, ni_in) {
  total <- sum(ni_in)
  fi <- sapply(ni_in, function(valor) {
    valor/total
  })
  return(fi)
}

# Función que calcula la frecuencia acumulada. Se puede dar la relativa o absoluta para este fin, sirve
calc_acumulada <- function(frecuencia_in) {
  frec_acumulada <- c(frecuencia_in[1])
  for (i in 2:length(frecuencia_in)) {
    frec_acumulada <- c(frec_acumulada,
                        frec_acumulada[i-1] + frecuencia_in[i])
  }
  return(frec_acumulada)
}

# Calcula la media ponderada de unos datos dada su tabla de frecuencias
calc_media_ponderada <- function(datos) {
  return(sum(datos$xi * datos$fi))
}

# Calcula el momento de orden "r" (dado) de una tabla de frecuencias
calc_momento <- function(orden, datos) {
  media <- sum(datos$xi * datos$fi)
```

```r
  resultado <- sapply(1:length(datos$xi), function(i) {
    (datos$xi[i] - media)^orden * datos$fi[i]
  })
  return(sum(resultado))
}

# Calcula la desviación típica de una tabla de frecuencias
calc_sd <- function(datos) {
  media <- sum(datos$xi * datos$fi)
  resultado <- sapply(1:length(datos$xi), function(i) {
    (datos$xi[i] - media)^2 * datos$fi[i]
  })
  return(sqrt(sum(resultado)))
}

# Calcula las marcas de clase de una lista de vectores que representan los límites de los intervalos:
# list(c(inf1, max1), c(inf2, max2), ..., c(infN, maxN))
calc_marcas_clase <- function(intervalos){
  marcas <- sapply(intervalos, function(valor) {
    (valor[1] + valor[2])/2
  })
  return(marcas)
}

# Dado un valor C(k) y una tabla de frecuencias con sus intervalos inferiores y superiores en columnas
calc_percentil_k <- function(ck, datos) {
  fila <- which(ck >= datos$vida_media_inf & ck < datos$vida_media_sup)
  li_1 <- datos$vida_media_inf[fila]
  ai <- datos$vida_media_sup[fila] - li_1
  resultado <- ((ck - li_1) * (datos$ni[fila] / ai) + datos$Ni[fila-1] ) * 1/sum(datos$ni)
  return(resultado)
}

# Dato un valor k tal que k < 1 (es decir, no en porcentaje sobre 100, sino en decimal) se calcula el d
calc_decil_k <- function(datos, k) {
  return(min(datos$xi[datos$Fi >= k]))
}

# Dada una tabla de frecuencias y, opcionalmente, un coeficiente diferente a 1,5 podemos calcular todos
calc_outliers <- function(datos, coeficiente = 1.5) {
  Q1 <- calc_decil_k(datos, 0.25)
  Q3 <- calc_decil_k(datos, 0.75)
  IQR <- Q3 - Q1
  IS <- Q3 + coeficiente * IQR
  II <- Q1 - coeficiente * IQR
  outliers <- datos$xi[sapply(datos$ni, function(valor) {
    valor > IS || valor < II
  })]
  return(outliers)
}

# Dadas dos poblaciones como vectores, compara sus dispersiones usando el coeficiente de variación.
```

```r
comparar_dispersion <- function(p1, p2) {
  cv1 <- sd(p1) / abs(mean(p1))
  cv2 <- sd(p2) / abs(mean(p2))
  if (cv1 > cv2) {
    return("p1 tiene una mayor dispersión")
  } else if (cv1 < cv2) {
    return("p2 tiene una mayor dispersión")
  } else {
    return("Ambas poblaciones tienen la misma dispersión")
  }
}
```

Primer Ejercicio

- Se menciona que hay 40 automóviles, por lo que N = 40.
- Debemos representar la distribución de frecuencias en una tabla estadística. Esto incluye calcular la frecuencia absoluta (n_i), la frecuencia relativa (f_i), la frecuencia absoluta acumulada (N_i) y la frecuencia relativa acumulada (F_i).
- Se pide el cálculo de:

  1. *La media.* Que es una medida de tendencia central calculada con el promedio aritmético de un conjunto de datos. No es una medida determinista pero da una idea de la cantidad aproximada de, en este caso, ocupantes por vehículo.

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

  2. *La mediana.* Es otra medida de tendencia central que representa el valor medio de un conjunto de datos cuando estos están ordenados en secuencia. A diferencia de la media, la mediana no se ve afectada por valores extremadamente altos o bajos (valores atípicos), lo que la hace más representativa en ciertas situaciones.
  3. *La moda.* Es otra medida de tendencia central, identifica el valor o valores más frecuentes en un conjunto de datos. A diferencia de la media y la mediana, la moda puede ser utilizada con datos numéricos, categóricos o nominales.

Resolución del ejercicio en R

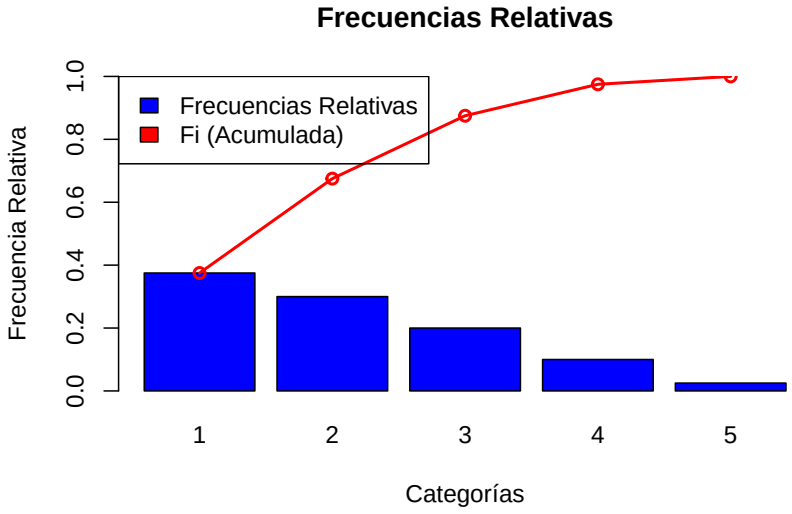
```r
# Main

datos <- c(1,3,2,2,3,1,1,2,2,1,
           1,4,3,1,3,2,3,2,2,2,
           1,2,5,1,3,1,2,1,3,1,
           4,1,1,3,4,2,2,1,1,4)
xi <- sort(unique(datos))
ni <- calc_ni(datos, xi)
fi <- calc_fi(datos, ni)
ni_acumulada <- calc_acumulada(ni)
fi_acumulada <- calc_acumulada(fi)

resultado <- data.frame(
  xi = xi,
  ni = ni,
  fi = fi,
```

```
  Ni = ni_acumulada,
  Fi = fi_acumulada
)

print(resultado)
```

```
##   xi ni    fi Ni    Fi
## 1  1 15 0.375 15 0.375
## 2  2 12 0.300 27 0.675
## 3  3  8 0.200 35 0.875
## 4  4  4 0.100 39 0.975
## 5  5  1 0.025 40 1.000
```

Adicionalmente, podríamos visualizar un gráfico de barras de la frecuencia relativa y la relativa acumulada, como vimos en clase:

```
barp <- barplot(resultado$fi,
        names.arg=resultado$xi,
        col="blue",
        main="Frecuencias Relativas",
        xlab="Categorías", ylab="Frecuencia Relativa", ylim=c(0, 1))
lines(barp,
      resultado$Fi,
      type='o',
      col="red",
      lwd=2) # 'lwd' es el ancho de la línea
legend("topleft",                        # Ubicación de la leyenda
       legend=c("Frecuencias Relativas", "Fi (Acumulada)"), # Textos de la leyenda
       fill=c("blue", "red"))              # Colores correspondientes
```



Se procederá calculando la media, mediana y moda.

```r
# Funciones
calc_moda <- function(tabla_frecuencias) {
  if (any(names(tabla_frecuencias) == "fi")) {
    max_freq <- max(tabla_frecuencias$fi)
    modas <- tabla_frecuencias$xi[tabla_frecuencias$fi == max_freq]
  } else {
    print("El dataset debe incluir la columna de frecuencias relativas")
  }
  return(modas)
}

# Main
media <- mean(datos)
mediana <- median(datos)
moda <- calc_moda(resultado)

print(paste("Media: ", media, "Mediana: ", mediana, "Moda: ", paste(moda, collapse = ", ")))
```

```
## [1] "Media:  2.1 Mediana:  2 Moda:  1"
```

Ejercicio 3

```r
library(ggplot2)
xi <- c(1,2,3,4,5)
fi_x <- c(0.2,0.2,0.2,0.2,0.2)
datos_x <- data.frame(
  xi = xi,
  fi = fi_x
)
yi <- c(-1,2,3,4,7)
fi_y <- c(0.05, 0.2, 0.5, 0.2, 0.05)
datos_y <- data.frame(
  xi = yi,
  fi = fi_y
)

calc_media_ponderada(datos_x)
```

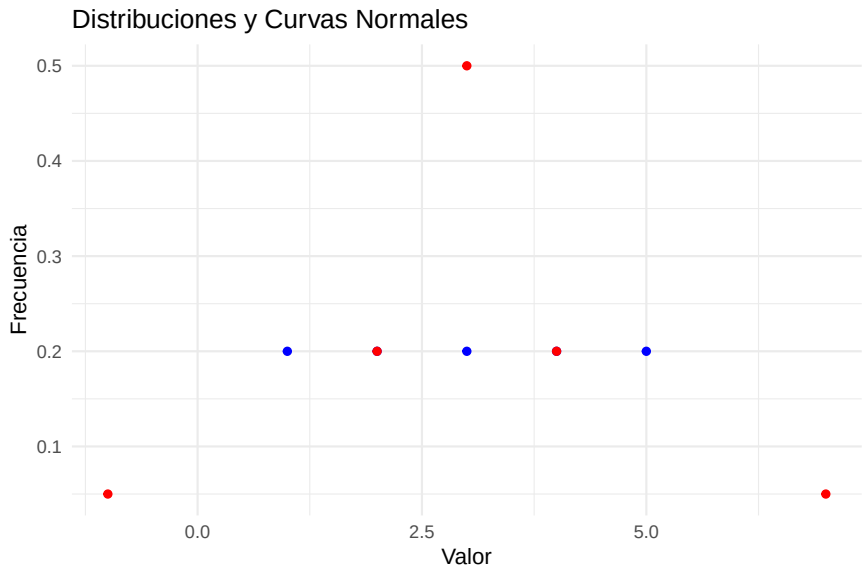
```
## [1] 3
```

```r
apuntamiento_x <- (calc_momento(datos = datos_x, orden = 4) / calc_sd(datos = datos_x)^4 ) - 3
apuntamiento_y <- (calc_momento(datos = datos_y, orden = 4) / calc_sd(datos = datos_y)^4 ) - 3

print(paste("Apuntamiento x: ", apuntamiento_x, "Apuntamiento y: ", apuntamiento_y))
```

```
## [1] "Apuntamiento x:  -1.3 Apuntamiento y:  3.5"
```

```r
# Calcular la media y desviación estándar para los datos_x y datos_y
media_x <- calc_media_ponderada(datos = datos_x)
sd_x <- calc_sd(datos = datos_x)
```

```r
media_y <- calc_media_ponderada(datos = datos_y)
sd_y <- calc_sd(datos = datos_y)

# Crear el gráfico
ggplot() +
  geom_point(data = datos_x, aes(x = xi, y = fi), color = "blue") +
  geom_point(data = datos_y, aes(x = xi, y = fi), color = "red") +
  theme_minimal() +
  labs(x = "Valor", y = "Frecuencia", title = "Distribuciones y Curvas Normales")
```



Ejercicio 4

Se pide calcular el coeficiente de asimetría de Fisher. Esto es el cociente del momento de orden 3 entre la desviación típica al cubo:

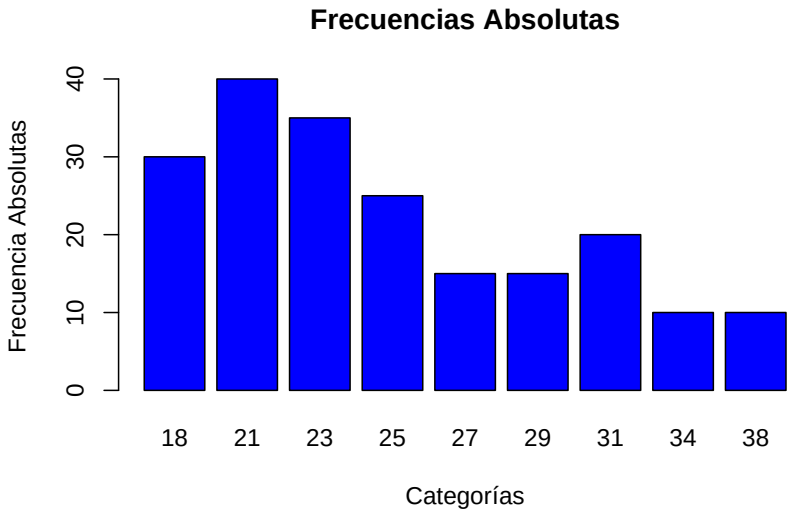
$$g_1 = \frac{\sum_{i=1}^{k} (x_i - \bar{x})^3 \times f_i}{\left(\sqrt{\sum_{i=1}^{k} (x_i - \bar{x})^2 \times f_i}\right)^3}$$

```r
marcas_clase <- c(18,21,23,25,27,29,31,34,38)
ni <- c(30,40,35,25,15,15,20,10,10)
datos = data.frame(
  xi = marcas_clase,
  ni = ni,
  fi = calc_fi(marcas_clase, ni)
)
datos
```

```
##   xi ni    fi
## 1 18 30 0.150
## 2 21 40 0.200
## 3 23 35 0.175
## 4 25 25 0.125
```

```
## 5 27 15 0.075
## 6 29 15 0.075
## 7 31 20 0.100
## 8 34 10 0.050
## 9 38 10 0.050
```

```r
asimetria_fisher <- calc_momento(3, datos) / (calc_sd(datos))^3

barp <- barplot(datos$ni,
        names.arg=datos$xi,
        col="blue",
        main="Frecuencias Absolutas",
        xlab="Categorías", ylab="Frecuencia Absolutas")
```




```r
print(paste("Coeficiente de asimetría de Fisher: ", asimetria_fisher, "> 0 Por lo que asimetría positiva
```

```
## [1] "Coeficiente de asimetría de Fisher:  0.722715758808346 > 0 Por lo que asimetría positiva o a la
```

El coeficiente de Fisher da positivo, por lo que la distribución, comparada con la distribución normal, es asimétrica a la derecha, cosa que podemos observar en el gráfico generado.

Ejercicio 5

```r
vida_media <- list(c(100,300), c(300,400), c(400,500), c(500,600), c(600,700), c(700,900))
vida_media_inferior <- sapply(vida_media, function(x) x[1])
vida_media_superior <- sapply(vida_media, function(x) x[2])
ni <- c(54, 96, 130, 88, 85, 47)
xi <- calc_marcas_clase(vida_media)
fi = calc_fi(ni_in = ni)
Fi = calc_acumulada(frecuencia_in = fi)
Ni = calc_acumulada(frecuencia_in = ni)
```

```r
datos <- data.frame(
  vida_media_inf = vida_media_inferior,
  vida_media_sup = vida_media_superior,
  xi = xi,
  ni = ni,
  fi = fi,
  Fi = Fi,
  Ni = Ni
)

calc_percentil_k(437, datos)
```

```
## [1] 0.3962
```

Ejercicio 8

```r
xi <- c(15,20,21,25,28,29,33,41)
ni <- c(30,25,30,52,45,20,10,1)
fi <- calc_fi(datos_in = xi, ni_in = ni)
Ni <- calc_acumulada(frecuencia_in = ni)
Fi <- calc_acumulada(frecuencia_in = fi)

datos <- data.frame(
  xi = xi,
  ni = ni,
  fi = fi,
  Ni = Ni,
  Fi = Fi
)

calc_outliers(datos)
```

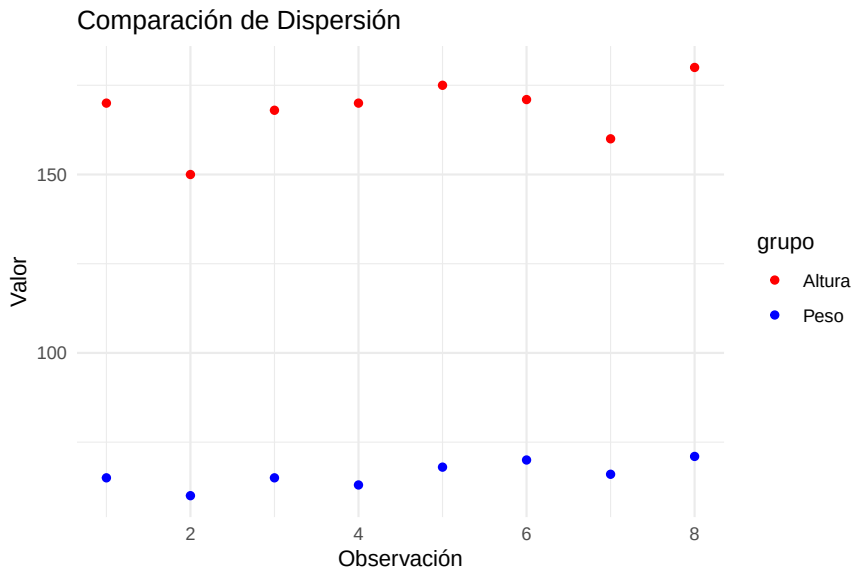
```
## [1] 25 28 41
```

Como podemos observar, se han calculado los outliers con un coeficiente predeterminado de 1.5, sin embargo, la función está hecha para que podamos modificar el intervalo de confianza que tenemos para clasificar un datos como outlier o no.

Ejercicio 9

```r
library(ggplot2)

# Crear un data.frame con los datos
p1 <- c(65, 60, 65, 63, 68, 70, 66, 71)
p2 <- c(170, 150, 168, 170, 175, 171, 160, 180)

# Crear un data.frame con todos los datos y una columna para identificar el grupo
datos <- data.frame(
```

```
  valor = c(p1, p2),
  grupo = rep(c("Peso", "Altura"), each = length(p1)),
  x = rep(1:length(p1), 2)
)

# Gráfico usando ggplot2
ggplot(datos, aes(x = x, y = valor, color = grupo)) +
  geom_point() +
  theme_minimal() +
  labs(x = "Observación", y = "Valor", title = "Comparación de Dispersión") +
  scale_color_manual(values = c("Peso" = "blue", "Altura" = "red"))
```




```
comparar_dispersion(p1, p2)
```

```
## [1] "p2 tiene una mayor dispersión"
```