Notas para el curso de Métodos Numéricos

Carlos E Martínez-Rodríguez

01 September, 2025

Índice general

Capítulo 1

Introducción al uso de R

1.1. Sesión 1

1.1.1. Operadores Lógicos

```
17<5
```

```
## [1] FALSE
```

```
17>5
```

```
## [1] TRUE
```

```
17<=5
```

```
## [1] FALSE
```

```
17>=5
```

```
## [1] TRUE
```

```
17!=5
```

```
## [1] TRUE
```

```
17==5
```

```
## [1] FALSE
```

1.2. Operadores Aritméticos

1.2.1. Sums, Resta, Multiplicación, División, Potencia, Modulo, División Entera

```
17+5
```

```
## [1] 22
```

```
17*5
```

```
## [1] 85
```

```
17*5
```

```
## [1] 85
```

```
17^5
```

```
## [1] 1419857
```

```
17%/%5
```

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

```
17%%5
```

```
## [1] 2
```

1.2.2. Logaritmos y exponenciales

```
log(1)
```

```
## [1] 0
```

```
log(12)
```

```
## [1] 2.484907
```

```
log(12,2)
```

```
## [1] 3.584963
```

```
exp(12)
```

```
## [1] 162754.8
```

```
exp(1)
```

```
## [1] 2.718282
```

1.2.3. Funciones Trigonométicas

```
sin(45)
```

```
## [1] 0.8509035
```

```
cos(45)
```

```
## [1] 0.525322
```

```
tan(45)
```

```
## [1] 1.619775
```

```
asin(0.96)
```

```
## [1] 1.287002
```

```
acos(0.97)
```

```
## [1] 0.2455655
```

```
atan(0.45)
```

```
## [1] 0.4228539
```

1.2.4. Funciones varias

```
abs(-34)
```

```
## [1] 34
```

```
sqrt(8)
```

```
## [1] 2.828427
```

```
floor(1.56)
```

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

```
ceiling(1.56)
```

```
## [1] 2
```

```
max(4,7,2,12)
```

```
## [1] 12
```

```
min(4,7,2,12)
```

```
## [1] 2
```

```
sign(-45)
```

```
## [1] -1
```

1.2.5. Ejercicios de práctica

1. calcular la expresion cos(pi/6+pi/2)+eˆ2 2 calcular la expresion cos(pi/6+pi/2)+eˆ2*log(5)+arc cos(1/raiz(2)) 3 introducir las siguientes expresiones:

a) 1/7
b) options(digits=3); 1/7
c) options(digits=6); 1/7
d) round(67.45)
e) round(75.324568,2)
f) options(digits=7);
g) signif(56.345458234234,2)
h) signif(56.345458234234)
i) exp(-30)
j) options(scipen= 999)
k) exp(-30)
l) options(scipen=0)

1.3. Sesión 2

1.3.1. Definición de constantes

```
e = exp(1);
x = 0.0034
e <- exp(1)
x <- 0.034;
x0 = e^(2*x)
```

1.3.2. Concatenar y pegar expresiones

```
txt = "El valor de x0 es _"
cat(txt, x0)
```

```
## El valor de x0 es _ 1.070365
```

```
paste(txt,x0)
```

```
## [1] "El valor de x0 es _ 1.07036530847877"
```

```
paste0(txt,x0)
```

```
## [1] "El valor de x0 es _1.07036530847877"
```

1.3.3. Asignación e impresión

```
x0 <- 1
x1 <- x0 - pi*x0 + 1
(x1 <- x0 - pi*x0 + 1 )
```

```
## [1] -1.141593
```

```
print(x1)
```

```
## [1] -1.141593
```

1.3.4. Listado de objetos definidos

```
ls()
```

```
## [1] "e"   "txt" "x"   "x0"  "x1"
```

Eliminar todos los objetos

```
rm(list= ls())
ls()
```

```
## character(0)
```

1.3.5. Imprimir pegar avanzado

```
x0 <- 1
x1 <- x0 - pi*x0 + 1
cat("x0 =", x0, "\n","x1 =", x1)
```

```
## x0 = 1
##  x1 = -1.141593
```

1.4. Sesión 3

1.4.1. Definición de funciones

nombre_funcion <- function(param1,param2,param3,. . . ,paramn){ instruccion 1 instruccion 2 return(valor_de_retorno) }

Ejemplo 1

```
fun1 <- function(x,a,b,h,k){
  res <- a+b*cos(hx+k)
  return(res)
}
```

Ejemplo 2

```
Discriminante <- function(a,b,c){
  res <- b^2-4*a*c
  return(res)
}
```

Capítulo 2

Gráficas en R

Primero limpiamos nuestro ambiente, eliminamos variables y cerramos las gráficas generadas con anterioridad.

```
## null device
##           1
```

Generemos los datos correspondientes a los pesos registrados de 90 personas

```
kgs <- c(100.5, 91.3, 60.4, 83.2, 49.6, 103.1, 60.3, 109.3, 83.5,
         63.9, 106, 50, 47, 89.7, 108.5, 78.9, 82.7, 60.7, 98.7, 85.2, 48.7, 106.7, 63.9, 84.1,
         69.5, 53.3, 108.9, 91.8, 108.6, 54.5, 95.1, 90.6, 115.9, 88.5, 67.7, 115.1, 108.3, 76.8,
         81.4, 102.6, 63.9, 105.9, 106.7, 76.3, 113.7, 50.3, 105.8, 81.4, 67.9, 91.3, 68.9, 93.9,
         113.7, 87.7, 92.8, 76.2, 104.7, 109.7, 72.6, 81.6, 112.2, 79.8, 60.7, 95.7, 100.1, 94,
         60.5, 117.1, 45.5, 112.7, 51.7, 107.8, 86.6, 90.3, 105.9, 64.7, 48, 55.4, 52.9, 58.2,
         117.1, 59.6, 69.9, 96.9, 97, 66.5, 67.4, 77.2, 73.7, 113)
```

2.1. Procesamiento de datos

Primero obtenemos un resumen de los datos generados

```
summary(kgs)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   45.50   65.15   84.65   83.86  104.30  117.10
```

Ordenamos los datos

```
kgs_ord <- sort(kgs)
cte <- 9
```

Definimos un vector que nos servirá para calcular los deciles:

```
indices <- 1:10; (indices)
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```
t <- cte*indices; (t)
```

```
##  [1]  9 18 27 36 45 54 63 72 81 90
```

con base en lo anterior ahora podemos extraer los cuartiles

```
mis_deciles <- kgs_ord[t]; (mis_deciles)
```

```
##  [1]  52.9  60.7  67.9  77.2  84.1  91.3  98.7 106.0 109.7 117.1
```

```
cuantiles <- quantile(kgs); (cuantiles)
```

```
##     0%    25%    50%    75%   100%
##  45.50  65.15  84.65 104.30 117.10
```

```
Q1 <- cuantiles[2]; (Q1)
```

```
##   25%
## 65.15
```

```
Q2 <- cuantiles[3]; (Q2)
```

```
##   50%
## 84.65
```

```
Q3 <- cuantiles[4]; (Q3)
```

```
##   75%
## 104.3
```

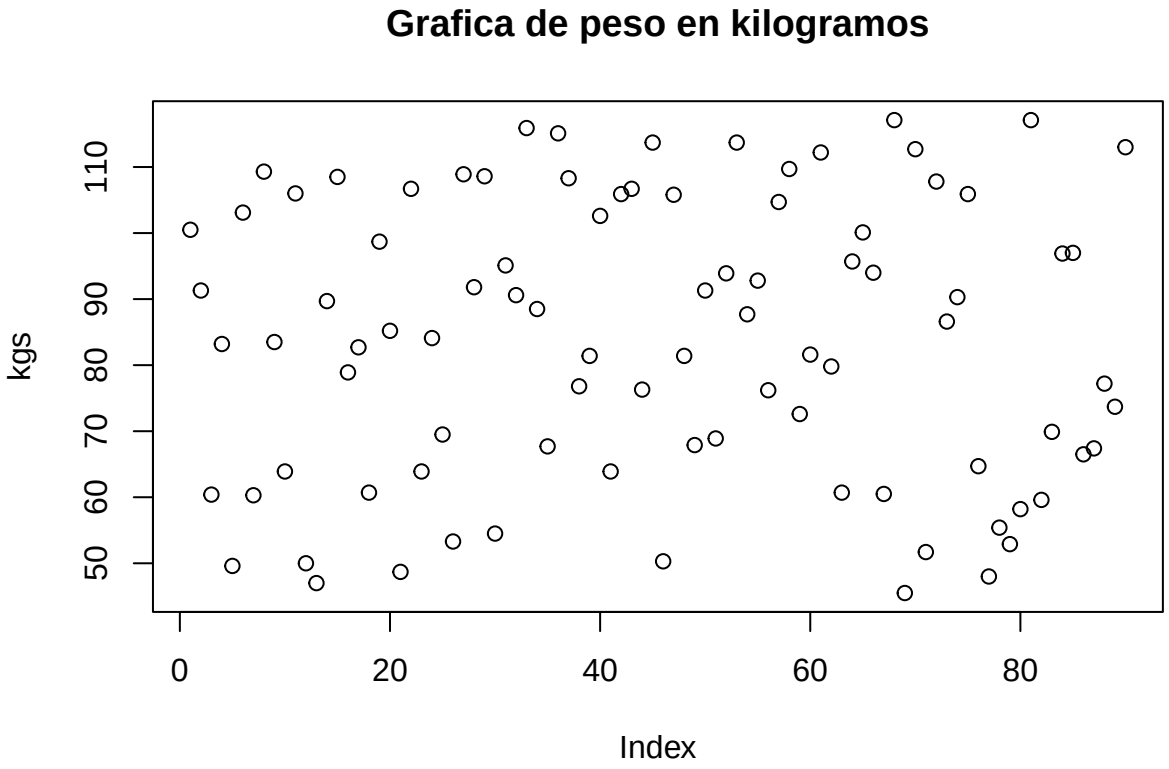
Calculemos la media, el valor máximo y el valor mínimo para los datos

```
x_media <- mean(kgs)
mi_min <- min(kgs);
mi_max <- max(kgs);
```

2.2. Gráficas con el comando *plot*

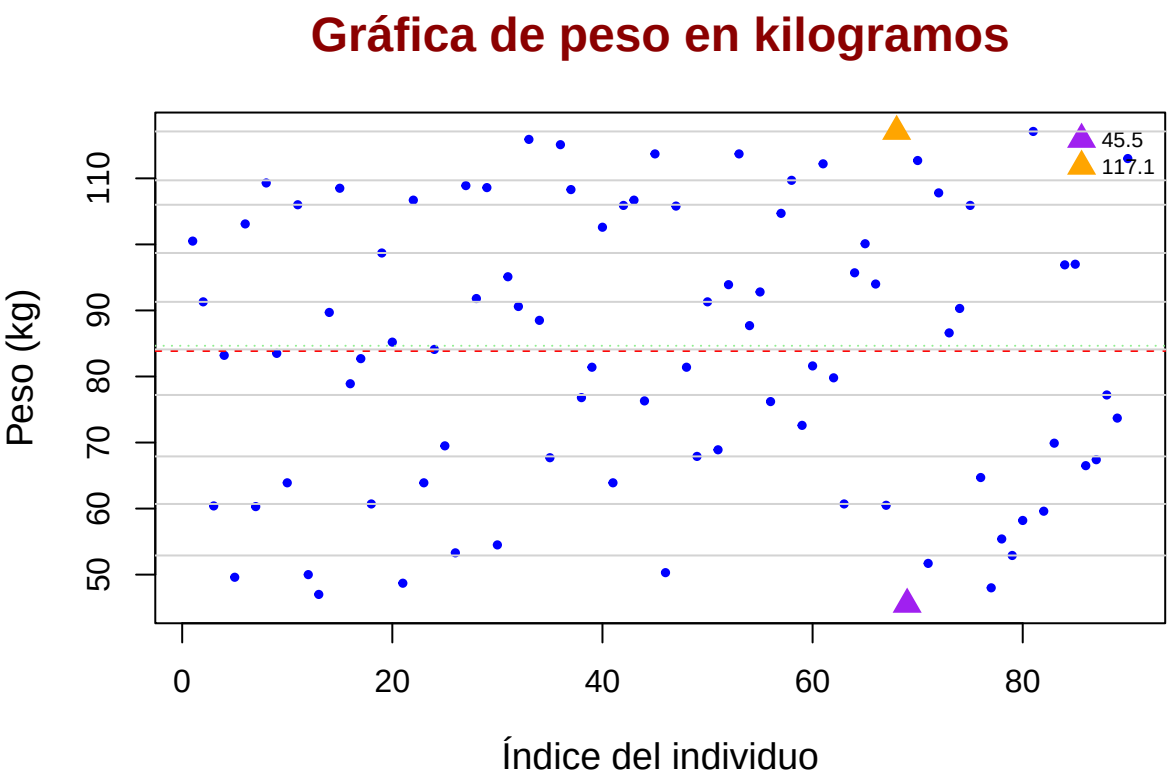
Generemos una gráfica simple de los datos

```
plot(kgs,main="Grafica de peso en kilogramos")
```



Ahora agreguemos detalles, etiquetas, título, colores, etc a la misma gráfica

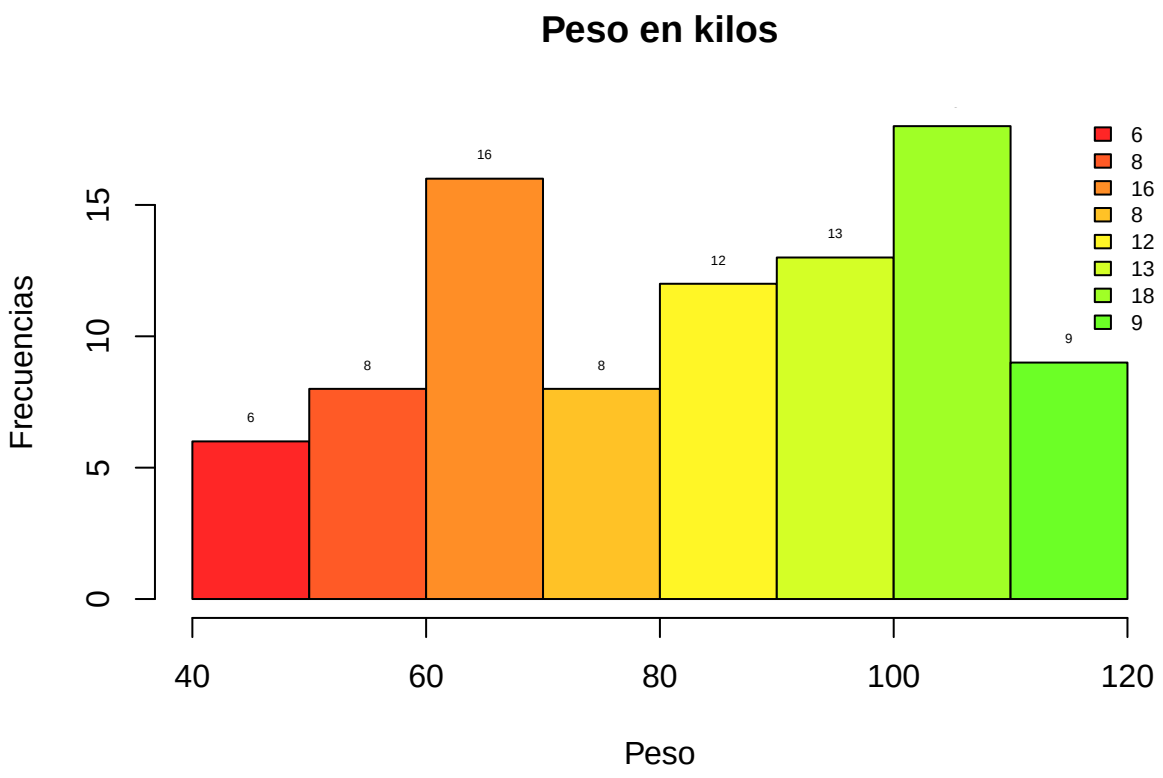
```
plot(
  kgs,
  main = "Gráfica de peso en kilogramos",
  xlab = "Índice del individuo",
  ylab = "Peso (kg)",
  col = "blue",
  pch = 19,
  cex = 0.5,
  cex.main = 1.5,
  cex.lab = 1.2,
  col.main = "darkred")
legend(
  "topright",                   # ubicación
  legend = c(mi_min,mi_max),    # etiquetas
  pch = 17,
  col=c("purple","orange"),
  pt.cex = 1.5,                 # tamaño de los símbolos
  cex = 0.7,
  bty = "n"                     # sin borde en la caja
)
abline(h = x_media, col = "red", lwd = 1, lty = 2)
abline(h = Q2, col = "lightgreen", lwd = 1, lty = 3)
abline(h = mis_deciles, col = "lightgray", lty = 1.5)
points(which.min(kgs), min(kgs), col = "purple", pch = 17, cex = 1.5)
points(which.max(kgs), max(kgs), col = "orange", pch = 17, cex = 1.5)
```



2.3. Generación de *Histogramas*

Con la misma idea que la gráfica anterior, generemos un histograma de los mismos datos

```
nbreaks=10;
miscolores <- rainbow(25,0.85);
h <- hist(kgs, breaks = nbreaks, col= miscolores,
          main = 'Peso en kilos',
          xlab="Peso",
          ylab="Frecuencias")
text(h$mids, h$counts, labels = h$counts, pos = 3, cex = 0.4,
     col = "black")
nl <- length(h$counts);
legend(
  "topright",                  # ubicación
  legend = h$counts,    # etiquetas
  fill = miscolores,
  col = miscolores,    # colores (uno por cada símbolo)
  pt.cex = 1.5,                # tamaño de los símbolos
  cex = 0.7,
  bty = "n"                    # sin borde en la caja
)
```



Mejoremos el gráfico agregando detalles, leyendas e información en la parte lateral de la gráfica

```
##---- Mejora del grafico ----
h2 <- hist(kgs, breaks = nbreaks, col= miscolores,
          main = 'Peso en kilos',
          xlab="Peso",
          ylab="Frecuencias",
          ylim=c(0,max(h$counts)*1.5))
text(h$mids, h$counts, labels = h$counts, pos = 3, cex = 0.8,
     col = "black")
nl <- length(h$counts);
legend(
  "topright",                  # ubicación
  legend = h$counts,     # etiquetas
  fill = miscolores,
  col = miscolores,    # colores (uno por cada símbolo)
  pt.cex = 1.5,                # tamaño de los símbolos
  cex = 0.5,
  bty = "n"                    # sin borde en la caja
)
```

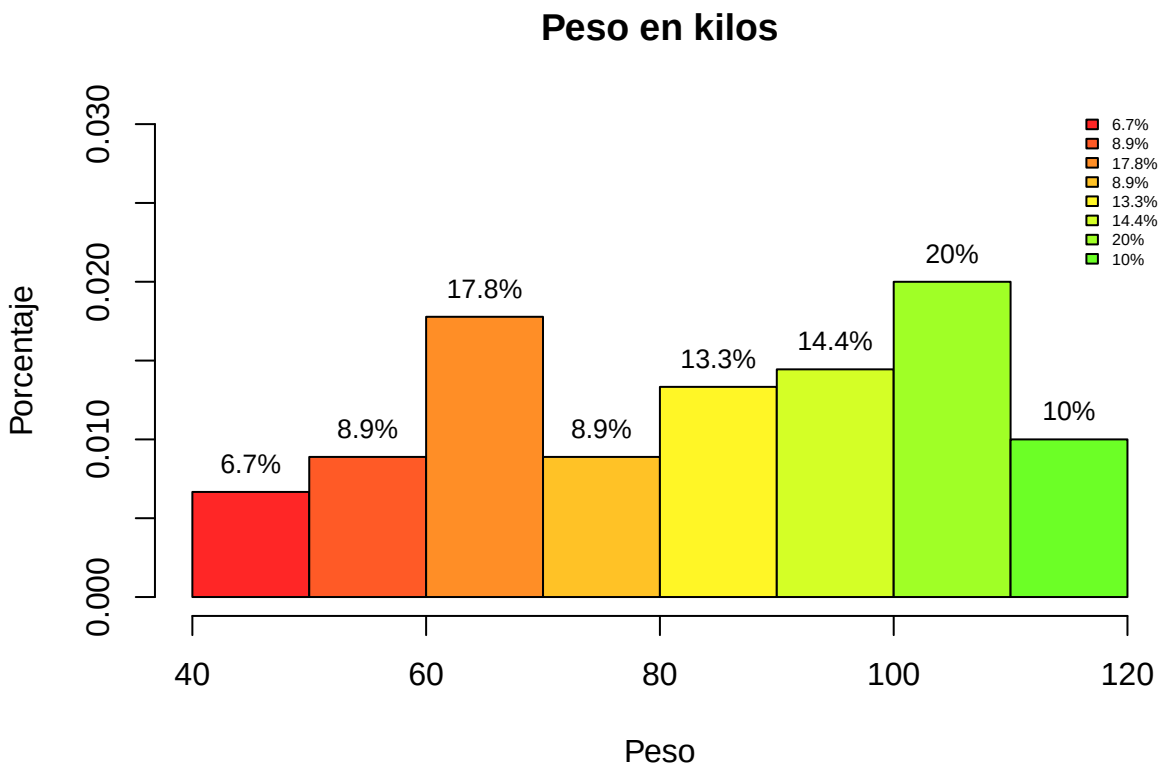
2.4. Histograma con porcentajes

```
##---- Grafico con porcentajes ----
h3 <- hist(kgs,
           breaks = nbreaks,
           col= miscolores,
           main = 'Peso en kilos',
           xlab="Peso",
           ylab="Porcentaje",
           ylim=c(0,max(h$density)*1.5),
           probability=TRUE)
porcentajes <- h3$counts/sum(h3$counts)*100

text(h3$mids,
     h3$density,
     labels = paste0(round(porcentajes,1),"%"),
     pos = 3,
     cex = 0.8,
     col = "black")

nl <- length(h3$counts); (nl)
```

```
## [1] 8
```

```
legend(
  "topright",                  # ubicación
  legend = paste0(round(porcentajes,1),"%"),
```

```
  fill = miscolores,
  cex = 0.5,
  bty = "n"                    # sin borde en la caja
)
```



2.5. Gráficos de pastel

```
#---- Graficos de pastel ----
# Tus datos
nbreaks <- 10
intervalos <- cut(kgs, breaks = nbreaks); (intervalos)
```

```
##  [1] (95.6,103]   (88.5,95.6] (59.8,67]   (81.3,88.5] (45.4,52.7] (103,110]
##  [7] (59.8,67]    (103,110]   (81.3,88.5] (59.8,67]   (103,110]   (45.4,52.7]
## [13] (45.4,52.7]  (88.5,95.6] (103,110]   (74.1,81.3] (81.3,88.5] (59.8,67]
## [19] (95.6,103]   (81.3,88.5] (45.4,52.7] (103,110]   (59.8,67]   (81.3,88.5]
## [25] (67,74.1]    (52.7,59.8] (103,110]   (88.5,95.6] (103,110]   (52.7,59.8]
## [31] (88.5,95.6]  (88.5,95.6] (110,117]   (88.5,95.6] (67,74.1]   (110,117]
## [37] (103,110]    (74.1,81.3] (81.3,88.5] (95.6,103]  (59.8,67]   (103,110]
## [43] (103,110]    (74.1,81.3] (110,117]   (45.4,52.7] (103,110]   (81.3,88.5]
## [49] (67,74.1]    (88.5,95.6] (67,74.1]   (88.5,95.6] (110,117]   (81.3,88.5]
## [55] (88.5,95.6]  (74.1,81.3] (103,110]   (103,110]   (67,74.1]   (81.3,88.5]
## [61] (110,117]    (74.1,81.3] (59.8,67]   (95.6,103]  (95.6,103]  (88.5,95.6]
## [67] (59.8,67]    (110,117]   (45.4,52.7] (110,117]   (45.4,52.7] (103,110]
## [73] (81.3,88.5]  (88.5,95.6] (103,110]   (59.8,67]   (45.4,52.7] (52.7,59.8]
## [79] (52.7,59.8]  (52.7,59.8] (110,117]   (52.7,59.8] (67,74.1]   (95.6,103]
```

```
## [85] (95.6,103]  (59.8,67]   (67,74.1]   (74.1,81.3] (67,74.1]   (110,117]
## 10 Levels: (45.4,52.7] (52.7,59.8] (59.8,67] (67,74.1] ... (110,117]
```

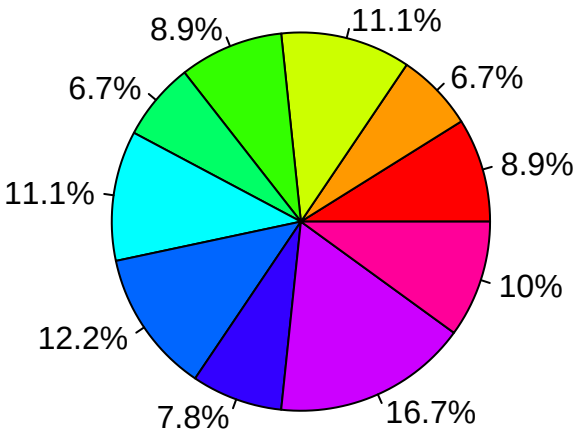
```
tabla <- table(intervalos); (tabla)
```

```
## intervalos
## (45.4,52.7] (52.7,59.8]   (59.8,67]   (67,74.1] (74.1,81.3] (81.3,88.5]
##           8           6          10           8           6          10
## (88.5,95.6]  (95.6,103]   (103,110]   (110,117]
##          11           7          15           9
```

```
porcentajes <- round(prop.table(tabla) * 100, 1); (porcentajes)
```

```
## intervalos
## (45.4,52.7] (52.7,59.8]   (59.8,67]   (67,74.1] (74.1,81.3] (81.3,88.5]
##         8.9         6.7        11.1         8.9         6.7        11.1
## (88.5,95.6]  (95.6,103]   (103,110]   (110,117]
##        12.2         7.8        16.7        10.0
```

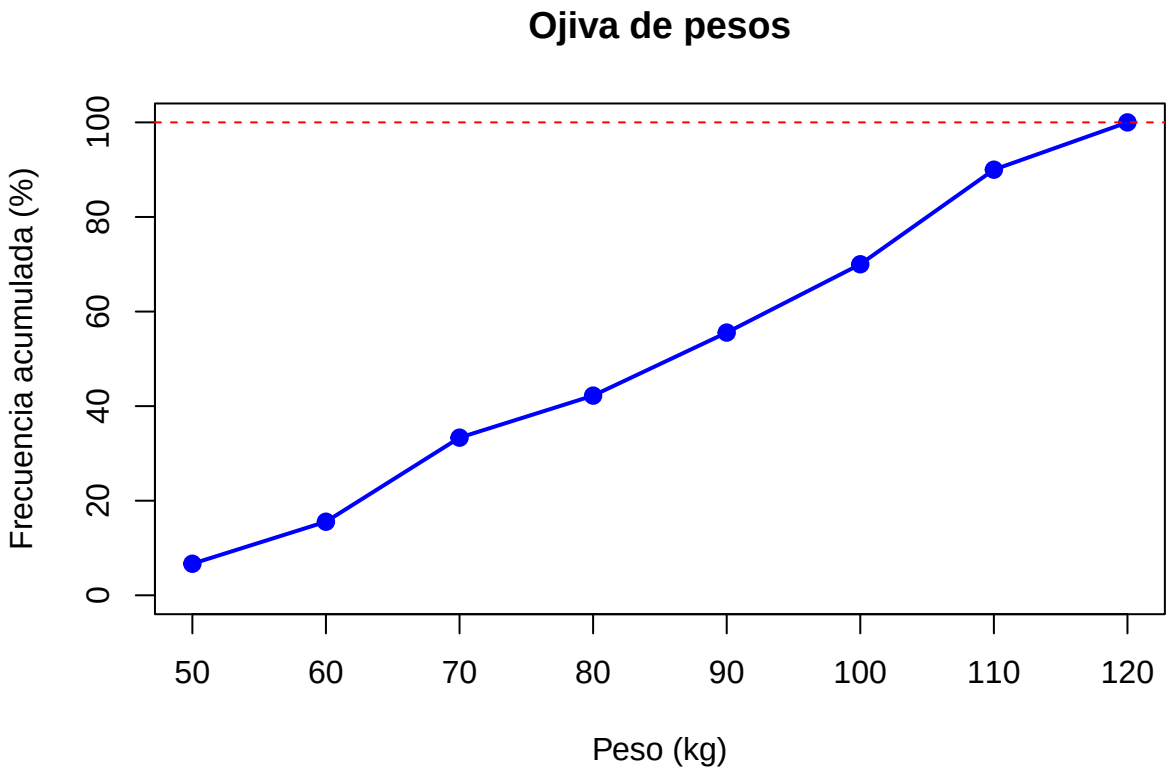
```
pie(tabla,
    main = "Distribución de pesos (10 intervalos)",
    col = rainbow(length(tabla)),
    labels = paste0(porcentajes, "%"))
```



2.6. Gráfica de Ojiva

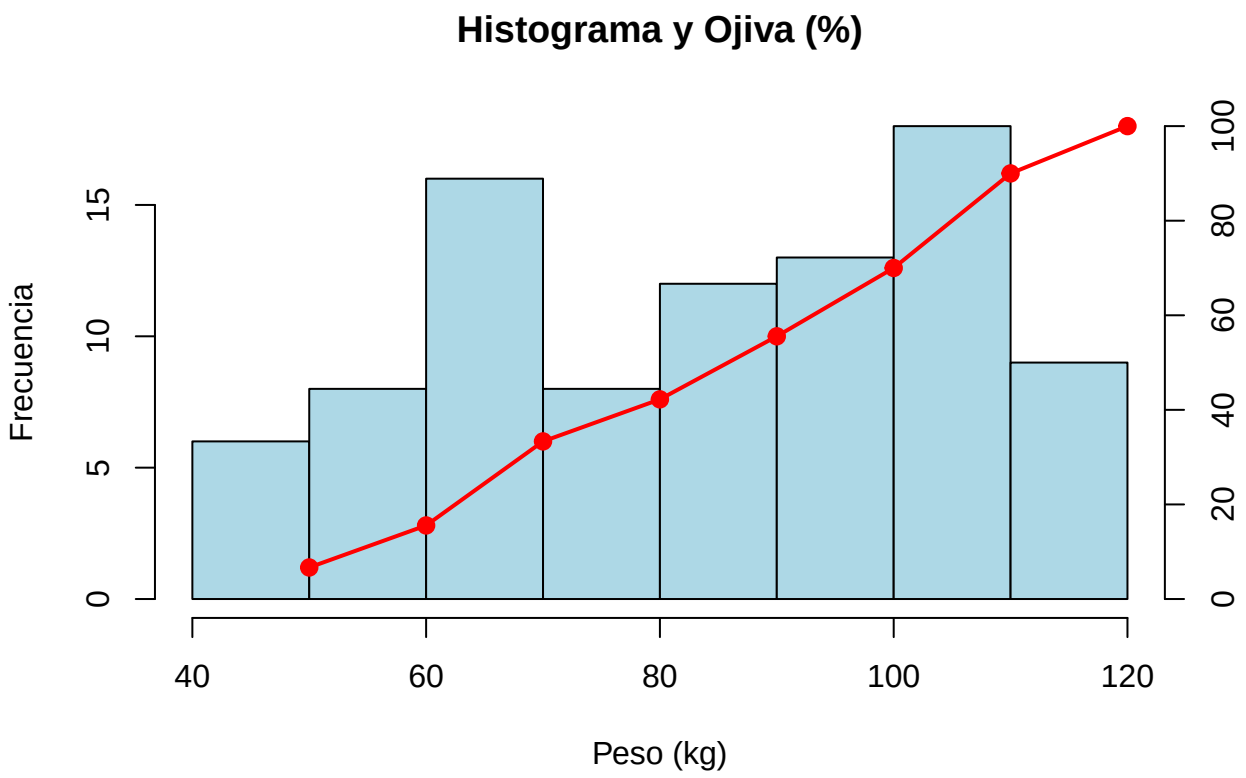
```
#---- Grafica de Ojiva ----
##---- Grafica sencilla ----
nbreaks <- 10
h <- hist(kgs, breaks = nbreaks, plot = FALSE)
freq_acum <- cumsum(h$counts) / sum(h$counts) * 100
x_vals <- h$breaks[-1]
plot(x_vals, freq_acum, type = "o",
     main = "Ojiva de pesos",
```

```
     xlab = "Peso (kg)",
     ylab = "Frecuencia acumulada (%)",
     col = "blue", pch = 19, lwd = 2,
     ylim = c(0, 100))
abline(h = 100, col = "red", lty = 2)
```



Mejoramos la gráfica anterior.

```
##---- Grafica combinada ----
nbreaks <- 10
h <- hist(kgs, breaks = nbreaks, plot = FALSE)
hist(kgs,
     breaks = nbreaks,
     col = "lightblue",
     main = "Histograma y Ojiva (%)",
     xlab = "Peso (kg)",
     ylab = "Frecuencia")
freq_acum_pct <- cumsum(h$counts) / sum(h$counts) * 100
x_vals <- h$breaks[-1]
par(new = TRUE)
plot(x_vals, freq_acum_pct,
     type = "o", pch = 19, lwd = 2, col = "red",
     axes = FALSE, xlab = "", ylab = "",
     xlim = range(h$breaks), ylim = c(0, 100))
axis(4)                          # eje derecho
```


Histograma y Ojiva (%)
Frecuencia
0
5
10
15
0
20
40
60
80
100
40
60
80
100
120
Peso (kg)

```
#---- Fin ----
```