Guia 13

Agustin Muñoz Gonzalez

31/5/2020

Preparamos el entorno

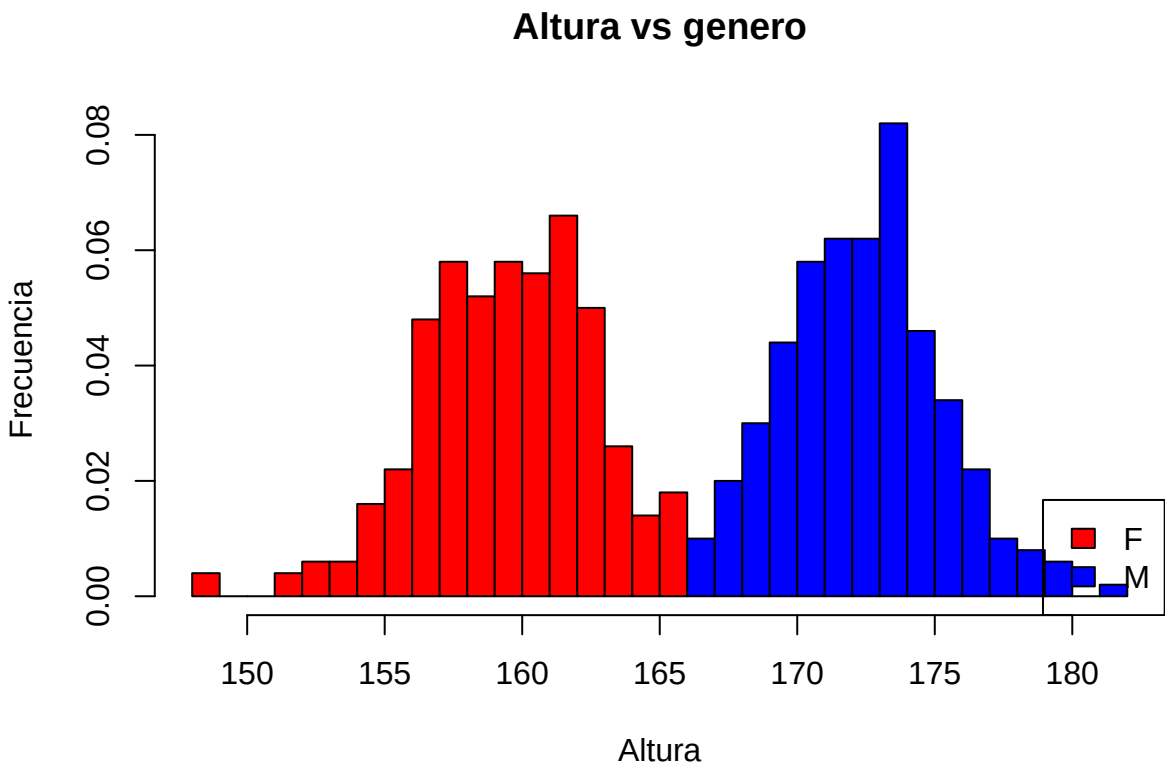
```
rm(list=ls())
```

1. La base.

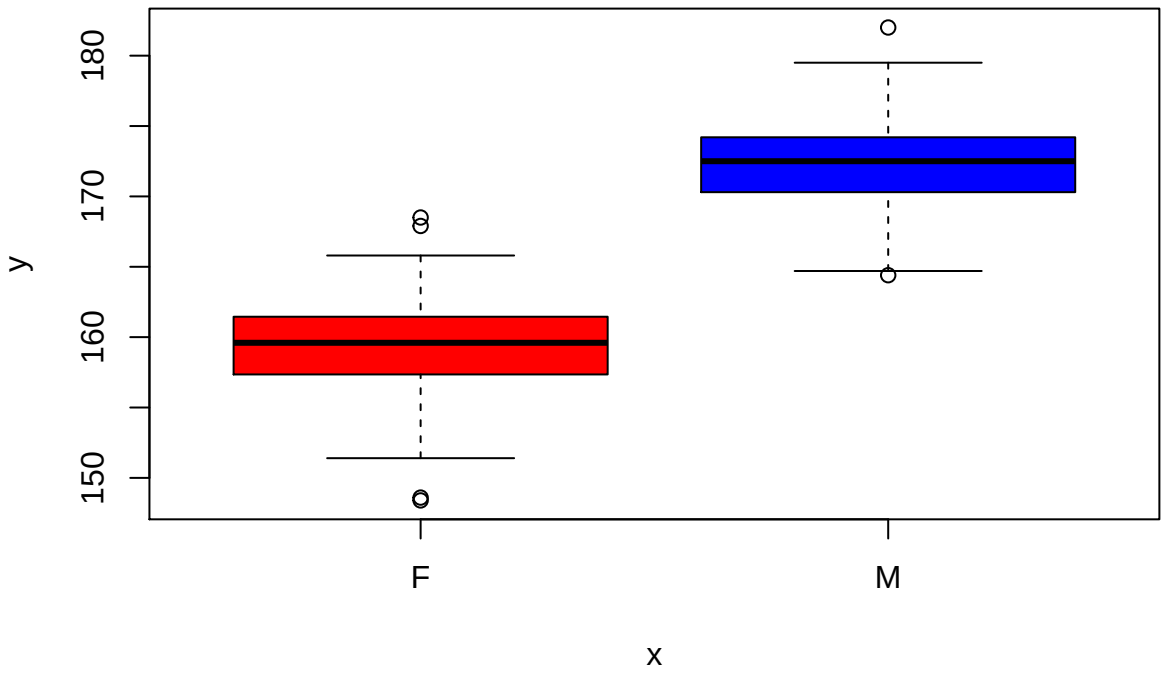
1. Descargar de esta página un conjunto de n = 500 observaciones, con todas las variables y leer el archivo en R. Trabajaremos con las variables altura y genero (codificada como F o M). Graficar un plot que pueda dar información sobre la relación entre estas dos variables.

Resolución:

```
alturas=read.csv('alturas_n_500.csv')
attach(alturas)
hist(altura,col=rep(c('red','blue'),each=18),
     freq=F,
     main='Altura vs genero',
     xlab='Altura',
     ylab='Frecuencia',
     nclass=30)
legend("bottomright",
            c("F","M"),
            fill=c("red","blue") )
```




```
plot(genero,altura,col=c('red','blue'))
```



2. Con la regla de la mayoría vamos a aprender a clasificar el género de un individuo como femenino (1) o masculino (0) cuando su altura x = 165 mediante el método de vecinos. Para ello, considerar los k = 10 vecinos más cercanos y calcular la proporción de 1's. Según este resultado, ¿cómo clasificarías al género de un nuevo individuo con altura igual a 165 cm, F o M? Repetir con x = 175.

Resolución:

Vamos a definir 2 funciones, k_posiciones_cercanas(xCentro,k) y k_vecinos_cercanos(xCentro,k), que devolveran las k posiciones mas cercanas a la de xCentro y los k elementos del vector altura mas cercanos a xCentro, respectivamente. La forma en que haremos esto cera restandole al vector altura el valos x=165 y tomando valor absoluto al vector resultante. Este vector estará formado por la distancia de todos los elementos de altura al valor 165, y de ahí basta tomar las k posiciones mas chicas o los k elementos de altura correspondientes a esas k posiciones mas chicas.

```
k_posiciones_cercanas=function(xCentro,k){
  distancias=abs(altura-xCentro)
  posiciones=c()
  for(i in (1:k)){
    menor_distancia=which.min(distancias)
    posiciones=c(posiciones,menor_distancia)
    distancias[menor_distancia]=1
    # Le pongo un 1 al lugar de la menor distancia, asi en la
    # proxima iteracion la menor distancia cambia.
  }
  posiciones
}
k_vecinos_cercanos=function(xCentro,k){
  datos=altura-xCentro
  altura_aux=altura
  vecinos=c()
  for(i in (1:10)){
    menor_distancia=which.min(abs(datos))
    vecinos=c(vecinos,altura_aux[menor_distancia])
    datos=datos[!datos %in% datos[menor_distancia]]
    altura_aux=altura_aux[!altura_aux %in% altura_aux[menor_distancia]]
  }
  vecinos
}
```

Ahora sí, calculemos la proporción de 1's para x=165 y k=10.

```
# Una forma
proporcion_F=mean(genero[altura %in% k_vecinos_cercanos(165,10)]=='F')
proporcion_F
```

```
## [1] 0.5833333
```

```
# Otra forma
proporcion_F=mean(genero[k_posiciones_cercanas(165,10)]=='F')
proporcion_F
```

```
## [1] 0.6
```

```
# Notar que da 2 valores distintos porque en rigor hay 12 valores
# de altura que coinciden con los de k_vecinos_cercanos(165,10)
# en vez de haber 10.
# La forma correcta es la segunda.
if(proporcion_F>=0.5)
```

```
{'El género de un individuo con altura 165 se clasifica como F'}else
  {'El género de un individuo con altura 165 se clasifica como M'}
```

```
## [1] "El género de un individuo con altura 165 se clasifica como F"
```

calculemos la proporción de 1's para x=175 y k=10.

```
proporcion_F=mean(genero[k_posiciones_cercanas(175,10)]=='F')
proporcion_F
```

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

```
if(proporcion_F>=0.5)
{'El género de un individuo con altura 165 se clasifica como F'}else
  {'El género de un individuo con altura 165 se clasifica como M'}
```

```
## [1] "El género de un individuo con altura 165 se clasifica como M"
```

Vamos a definir el clasificador.

```
clasificador_vecinos=function(xNuevo,k){
  proporcion_F=mean(genero[k_posiciones_cercanas(xNuevo,k)]=='F')
  proporcion_F
  if(proporcion_F>=0.5)
  {sprintf('El género de un individuo con altura %s y ventana %s se clasifica como F',xNuevo,k)}else
  {sprintf('El género de un individuo con altura %s y ventana %s se clasifica como M',xNuevo,k)}
}
```

3. Con la regla de la mayoría vamos a aprender a clasificar el género de un individuo como femenino (1) o masculino (0) cuando su altura x = 165 mediante el método de promedios móviles. Para ello, considerar una ventana de tamaño h = 1,5 alrededor del punto de interés y calcular la proporción de 1's. Según este resultado, ¿cómo clasificarías al género de un nuevo individuo con altura igual a 165 cm, F o M? Repetir con x = 175.

Resolución:

Definimos directamente el clasificador y evaluamos en donde nos pide el ejercicio.

```
clasificador_ventana=function(xNuevo,ventana){
  promedio_F=mean(genero[xNuevo-ventana<=altura
                         & altura <= xNuevo+ventana]=='F')
  if(promedio_F>=0.5)
{sprintf('El género de un individuo con altura %s y ventana %s se clasifica como F',xNuevo,ventana)}els
  {sprintf('El género de un individuo con altura %s y ventana %s se clasifica como M',xNuevo,ventana)}
}
clasificador_ventana(165,1.5)
```

```
## [1] "El género de un individuo con altura 165 y ventana 1.5 se clasifica como F"
```

```
clasificador_ventana(175,1.5)
```

```
## [1] "El género de un individuo con altura 175 y ventana 1.5 se clasifica como M"
```

2. El cuerpo: Regla óptima de Bayes - Método Discriminativo.

4. Clasificar el género de un individuo en F o M conociendo su altura mediante la regla de la mayoría utilizando el método de vecinos más cercanos. Para ello, implementar la función ClasificoVecinos(X, Y, xNuevo, k=10) que tenga por input un conjunto de valores de X, sus correspondientes valores de Y, un nuevo valor x para el que se quiere realizar la clasificación y la cantidad k = 10 de vecinos que vamos a utilizar para calcular la regla de la mayoría. (. . . por ahora te damos el k. . .)

Resolución:

Dado que en el Ejercicio 2 definimos una función análoga a la que nos piden acá pero con los datos X=altura e Y=genero, creamos dos funciones nuevas k_posiciones_cercanas_X(X,xNuevo,k) y ClasificoVecinos(X,Y,xNuevo,k) basandonos en las del Ejercicio 2.

```
k_posiciones_cercanas_X=function(X,xCentro,k){
  distancias=abs(X-xCentro)
  posiciones=c()
  for(i in (1:k)){
    menor_distancia=which.min(distancias)
    posiciones=c(posiciones,menor_distancia)
    distancias[menor_distancia]=1
    # Le pongo un 1 al lugar de la menor distancia, asi en la
    # proxima iteracion la menor distancia cambia.
  }
  posiciones
}
ClasificoVecinos=function(X, Y, xNuevo, k){
  proporcion_F=mean(Y[k_posiciones_cercanas_X(X,xNuevo,k)]=='F')
  proporcion_F
  if(proporcion_F>=0.5)
  {'F'}else{'M'}
}
```

5. Clasificar el género de un individuo en F o M conociendo su altura mediante la regla de la mayoría utilizando el método de promedios móviles. Para ello, implementar la función ClasificoMovil(X, Y, xNuevo, h=1) que tenga por input un conjunto de valores de X, sus correspondientes valores de Y, un nuevo valor x para el que se quiere realizar la clasificación y una ventana h = 1 para calcular la regla de la mayoría. (. . . por ahora te damos la ventana h. . .)

Resolución:

```
ClasificoMovil=function(X, Y, xNuevo, h){
  if(sum(xNuevo-h<=X & X <= xNuevo+h)==0){
NA}else{
  proporcion_F=mean(Y[xNuevo-h<=X
                      & X <= xNuevo+h]=='F')
  if(proporcion_F>=0.5)
{'F'}else{'M'}
}
  }
```

3. El aditivo aromático: Regla óptima de Bayes -Método Generativo.

Recordemos que cuando la covariable es una variable continua la regla de Bayes óptima puede escribirse como:

$$g^{op}(x) = \begin{cases} 1, & \text{si } f_1(X)\mathbb{P}(Y=1) \geq f_0(X)\mathbb{P}(Y=0) \\ 0, & \text{c.c.} \end{cases}$$

donde $X|Y=0$ f_0 y $X|Y=1$ f_1 son las densidades condicionales. Como hemos mencionado en clase, en los métodos generativos la regla de Bayes se implementa en la práctica estimando las densidades f_0 y f_1 y la probabilidad P(Y = 1).

6. Volvamos a la **Guía 7 Predicciones ítems 6 y 7**, donde se realizó un histograma de alturas para cada género y se le superpuso una curva a cada uno de ellos. ¿Qué curva se le superpuso a cada histograma? ¿Con qué parámetros? Realizar nuevamente los histogramas de alturas para cada cada sexo y a cada uno de ellos superponerle la curva como en la Guía 7. **¿Qué relación guardan estas curvas con las densidades f_0 y f_1 ?**

Resolución:

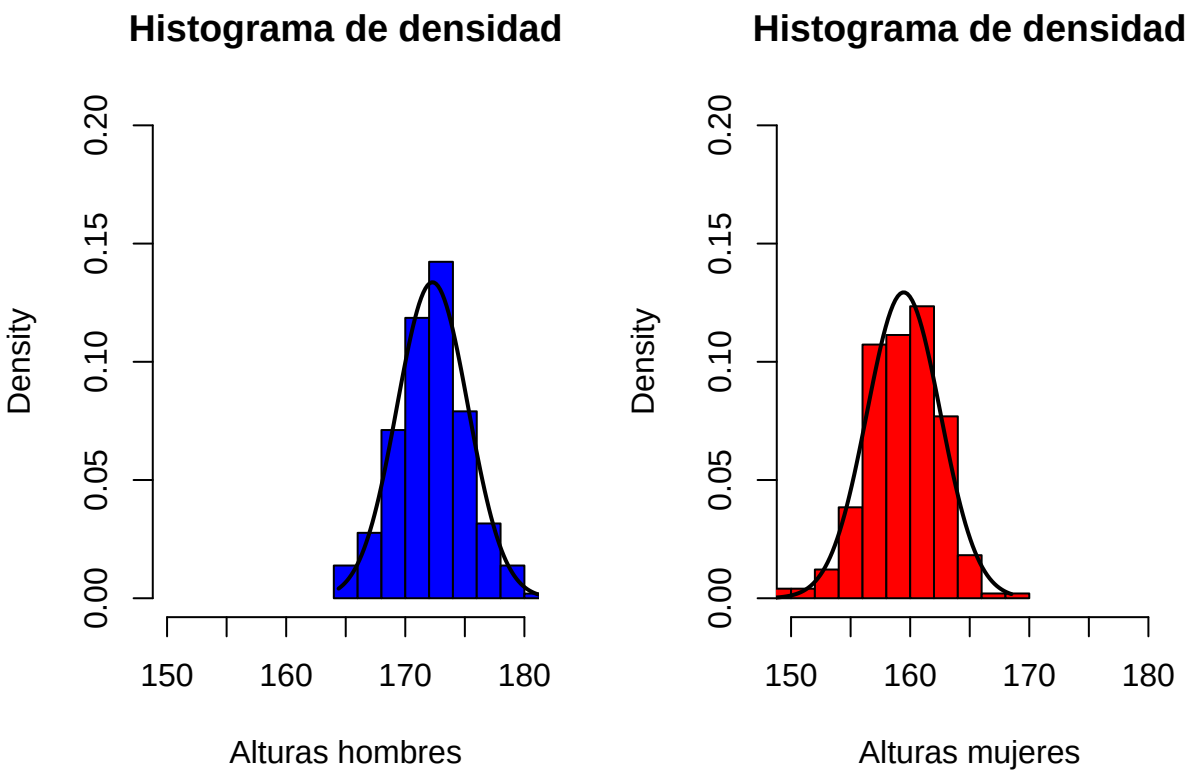
Grafiquemos nuevamente ambos histogramas con las curvas superpuestas.

```
media_M=mean(altura[genero=='M'])
desvio_M=sd(altura[genero=='M'])
grilla_M=seq(range(altura[genero=='M'])[1],
             range(altura[genero=='M'])[2],length=100)
funn_M=dnorm(grilla_M,media_M,desvio_M)

media_F=mean(altura[genero=='F'])
desvio_F=sd(altura[genero=='F'])
grilla_F=seq(range(altura[genero=='F'])[1],
             range(altura[genero=='F'])[2],length=100)
funn_F=dnorm(grilla_F,media_F,desvio_F)

par(mfrow=c(1,2))
hist(altura[genero=='M'],freq=F,main="Histograma de densidad",
     nclass=10, col="blue",xlab="Alturas hombres",
     xlim=c(150,180), ylim=c(0,0.2))
lines(grilla_M,funn_M,lwd=2)

hist(altura[genero=='F'],freq=F,main="Histograma de densidad",
     nclass=10, col="red",xlab="Alturas mujeres",
     xlim=c(150,180), ylim=c(0,0.2))
lines(grilla_F,funn_F,lwd=2)
```



La curva que se le superpuso a cada grafico es la curva de densidad normal con la media y la varianza de altura[genero==‘M’] y altura[genero==‘F’] respectivamente.

f_0 es precisamente la densidad que dibujamos sobre el grafico de los hombres y f_1 es la densidad de las mujeres.

7. ¿Cuál es la proporción de individuos de género femenino en tu conjunto de datos? ¿Cómo estimarías P(Y = 1) a partir de tus datos? ¿Cuánto te da la estimación propuesta?

Resolución:

La proporción de femeninos es

```
proporcion_F=mean(genero=='F')
proporcion_F
```

```
## [1] 0.494
```

¿P(Y=1)=proporcion_F? SI, LA FRECUENCIA RELATIVA!! RDO: 1) PROBA=FREC RELATIVA 2) COMO LOS VALORES SON 0 Y 1 EL PROMEDIO COINCIDE CON LA FREC RELATIVA DEL 1

8. Haciendo un plug-in en $g^{op}(x)$ con las estimaciones de $f_1(x)$, $f_0(x)$, P(Y = 1) y P(Y = 0), vamos a aprender a clasificar el género de un individuo como femenino (1) o masculino (0) cuando su altura x = 165. Usando el método generativo con tus datos, ¿cómo clasificarías a alguien nuevo con altura igual a 165 cm, F o M? Repetir el ítem anterior para x = 175.

Resolución:

Debemos primero definir las funciones estimadas f_0 y f_1. Para ello usaremos las curvas de los graficos del Ejercicio 6.

```
media_M=mean(altura[genero=='M'])
desvio_M=sd(altura[genero=='M'])
f_0=function(x){
  dnorm(x,media_M,desvio_M)
```

```
}

media_F=mean(altura[genero=='F'])
desvio_F=sd(altura[genero=='F'])
f_1=function(x){
  dnorm(x,media_F,desvio_F)
}
```

Ahora que estimamos f_0, f_1, P(Y=1) y P(Y=0) podemos definir el clasificador optimo.

```
clasificador_op=function(xNuevo){
  proporcion_F=mean(genero=='F')
  proporcion_M=1-proporcion_F
  if(f_1(xNuevo)*proporcion_F>f_0(xNuevo)*proporcion_M)
{sprintf('yNuevo=%s, es decir, el nuevo individuo se clasifica como F',1)}else
  {sprintf('yNuevo=%s, es decir, el nuevo individuo se clasifica como M',0)}
}
```

Calculemos lo que nos pide el ejercicio.

```
clasificador_op(165)
```

```
## [1] "yNuevo=1, es decir, el nuevo individuo se clasifica como F"
```

```
clasificador_op(175)
```

```
## [1] "yNuevo=0, es decir, el nuevo individuo se clasifica como M"
```

9. Clasificar el género de un individuo en F o M conociendo su altura mediante el método generativo. Para ello, implementar la función ClasificoGenerativo(X, Y, xNuevo) que tenga por input un conjunto de valores de X, sus correspondientes valores de Y, un nuevo valor x para el que se quiere realizar la clasificación.

Resolución:

Basandonos en la funcion clasificador_op() del ejercicio anterior definimos

```
ClasificoGenerativo=function(X, Y, xNuevo){
  proporcion_F=mean(Y=='F')
  proporcion_M=1-proporcion_F
  if(f_1(xNuevo)*proporcion_F>f_0(xNuevo)*proporcion_M)
{'F'}else{'M'}
}
```

4. A batir!

10. Ahora vamos a testear las reglas. En el archivo alturas.testeo.csv se encuentran 31 datos de altura que separamos para testear como funcionan los tres métodos implementados. Para ello, aplicar a este conjunto de datos cada una de las tres reglas implementadas en los ítems anteriores, calcular el Error de Clasificación Empírico de cada clasificador sobre estos datos y completar la información en el archivo compartido de resultados. ¿Cuál de ellas te parece que clasifica mejor?

Resolución:

```
alturas_testeo=read.csv('alturas.testeo.csv')
ClasificoVecinos_test=function(xNuevo){
  ClasificoVecinos(alturas_testeo$altura,alturas_testeo$genero,xNuevo,10)
}
ClasificoMovil_test=function(xNuevo){
  ClasificoMovil(alturas_testeo$altura,alturas_testeo$genero,xNuevo,1)
}
ClasificoGenerativo_test=function(xNuevo){
  ClasificoGenerativo(alturas_testeo$altura,alturas_testeo$genero,xNuevo)
}
error_empirico_vecinos=mean(lapply(alturas_testeo$altura,ClasificoVecinos_test)
                            !=alturas_testeo$genero)
error_empirico_vecinos
```

```
## [1] 0.03225806
```

```
error_empirico_movil=mean(lapply(alturas_testeo$altura,ClasificoMovil_test)
                            !=alturas_testeo$genero)
error_empirico_movil
```

```
## [1] 0.03225806
```

```
error_empirico_generativo=mean(lapply(alturas_testeo$altura,ClasificoGenerativo_test)
                            !=alturas_testeo$genero)
error_empirico_generativo
```

```
## [1] 0.03225806
```

Las 3 clasifican exactamente igual de bien.

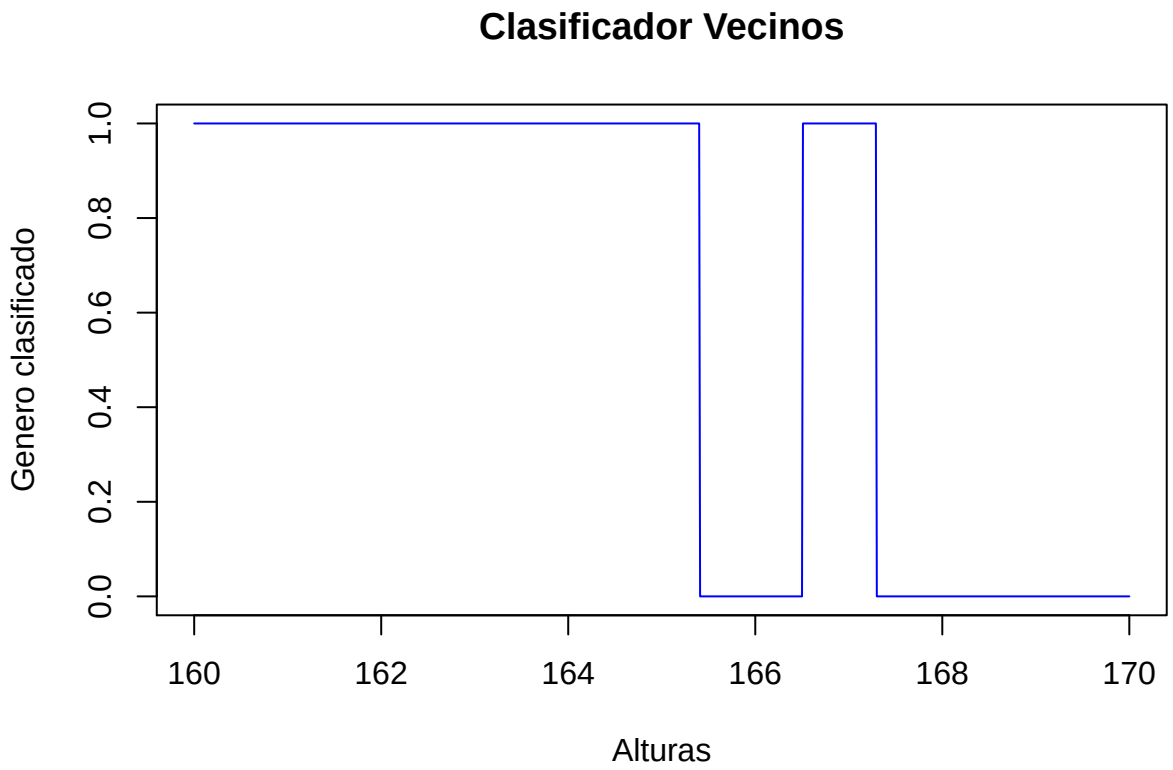
5. Bonus Track

Leyendo el fondo de la copa...

11. Graficar xNuevo (en el eje de abscisas) tomando valores entre 160 y 170 con un paso de 0.01 y en el de ordenadas el valor con el que clasifica a cada valor la regla ClasificoVecinos que implementaste con tus datos (sugerimos representar con línea). Interpretar el criterio con el que clasifica esta regla.

Resolución:

```
grilla=seq(160,170,0.01)
datos_clasificador_vecinos=lapply(grilla,ClasificoVecinos_test)
datos_clasificador_vecinos[datos_clasificador_vecinos=='M']=0
datos_clasificador_vecinos[datos_clasificador_vecinos=='F']=1
plot(grilla,datos_clasificador_vecinos, type='l',
     xlab='Alturas', ylab='Genero clasificado', col='blue',
     main='Clasificador Vecinos')
```



12. Repetir el ítem anterior con ClasificoMovil y ClasificoGenerativo y superponer con otro color al gráfico anterior. Interpretar y comparar el criterio con el que clasifica cada regla.

Resolución:

```
datos_clasificador_movil=lapply(grilla,ClasificoMovil_test)
datos_clasificador_movil[datos_clasificador_movil=='M']=0
datos_clasificador_movil[datos_clasificador_movil=='F']=1
grilla_aux=grilla[!grilla %in%
                    grilla[which(datos_clasificador_movil=='NA')]]
datos_clasificador_movil=datos_clasificador_movil[
  !datos_clasificador_movil %in% datos_clasificador_movil
  [which(datos_clasificador_movil=='NA')]]
# Ver como graficar movil por el tema de los NA.
```

```r
# Graficarlo con la grilla sin los valores problematicos.
datos_clasificador_generativo=lapply(grilla,ClasificoGenerativo_test)
datos_clasificador_generativo[datos_clasificador_generativo=='M']=0
datos_clasificador_generativo[datos_clasificador_generativo=='F']=1
plot(grilla,datos_clasificador_vecinos, type='l',
     xlab='Alturas', ylab='Genero clasificado', col='blue',
     main='Clasificadores')
points(grilla_aux,datos_clasificador_movil, type='l', col='red')
points(grilla,datos_clasificador_generativo, type='l', col='magenta')
```