Ejercicio 1. Lab 4

Para realizar esta tarea tendremos que instalar las librerías animation y pyramid. El primer paso es la lectura correcta de los datos, que pasa por un previo tratamiento de los mismos. Cuando los datos se descargan del Istac es recomendable eliminar el separador de los miles y agregarle , en el separador decimal. Esto es fácil de realizar usando ms excel.

Una vez terminado esto tendremos que poner las opciones de la animación y usaremos la función saveGIF para crear nuestra imagen.

Dentro de esta función recorreremos del 19 al 1 debido a que los años están al revés en los datos descargados e iremos confeccionando nuestra pirámide.

```
setwd("C:/Users/jrpen/OneDrive/Escritorio")
poblac <- read.csv2("istac.csv", header=F, col.names=c("Edad", rep(c("Hombre", "Mujer"), 20)), skip=10,

library(animation)
library(pyramid)

ani.options(interval=.5)

saveGIF({
  for (i in 19:1){
    piramide <- as.data.frame((poblac[,c((2*i), (2*i)+1, 1)]))
    names(piramide)<- c("Hombres", "Mujeres", "Edad")
    pyramid(piramide, Llab = "Hombres", Rlab = "Mujeres", Clab = "Edad",
            main=paste("Poblacion Canarias", 2019-i),
            Laxis=seq(0,100000, len=5), AxisFM = "d", Csize=0.8)
  }
}, movie.name="prueba.gif")
```

```
## Output at: prueba.gif
```

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

Ejercicio 3. Lab 4

setup)

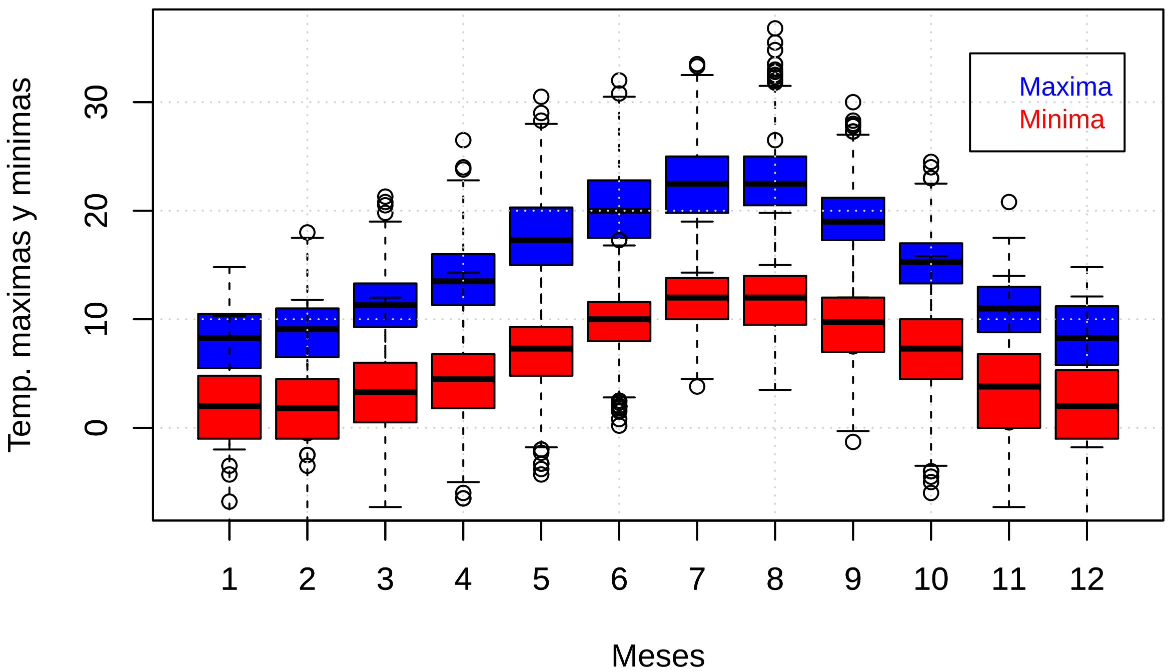
```
setwd("~/Escritorio/R/Datasets")
library(knitr)
library(ggplot2)
datos_c <- read.table("Tiempo_Clima.txt", sep="\t", dec = ".", header=T)
attach(datos_c)
kable(datos_c[c(1:3, 150:152, 1300:1304), ])
```

	T_Maxima	T_Minima	Precipitacion	Mes	Year
1	10.8	6.5	12.2	1	1987
2	10.5	4.5	1.3	1	1987
3	7.5	-1.0	0.1	1	1987
150	18.3	12.5	0.0	5	1987
151	19.3	10.8	11.7	5	1987
152	19.8	10.3	0.0	6	1987
1300	23.8	12.5	0.0	7	1990
1301	25.0	7.3	0.0	7	1990
1302	25.0	6.8	0.0	7	1990
1303	26.3	8.4	0.0	7	1990
1304	24.0	13.3	0.0	7	1990

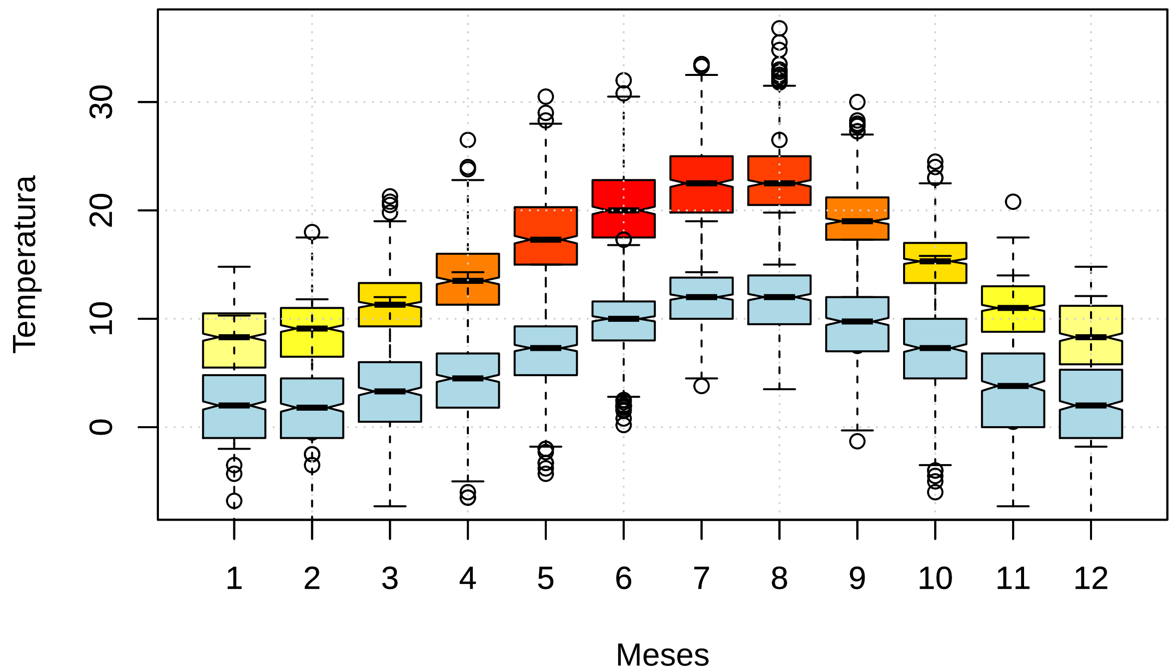
Visualizar con R básico y con ggplot2 los cambios de temperatura máxima y mínima en función del mes de año.

Para ello realizaremos una primera visión conjunta usando el paquete de R básico, realizaremos dos plot de las temperaturas máximas y las mínimas con una leyenda explicativa (se pueden apreciar dos, una mas personalizada y otra menos). Y a continuación usaremos ggplot2 para reproducir la misma situación.

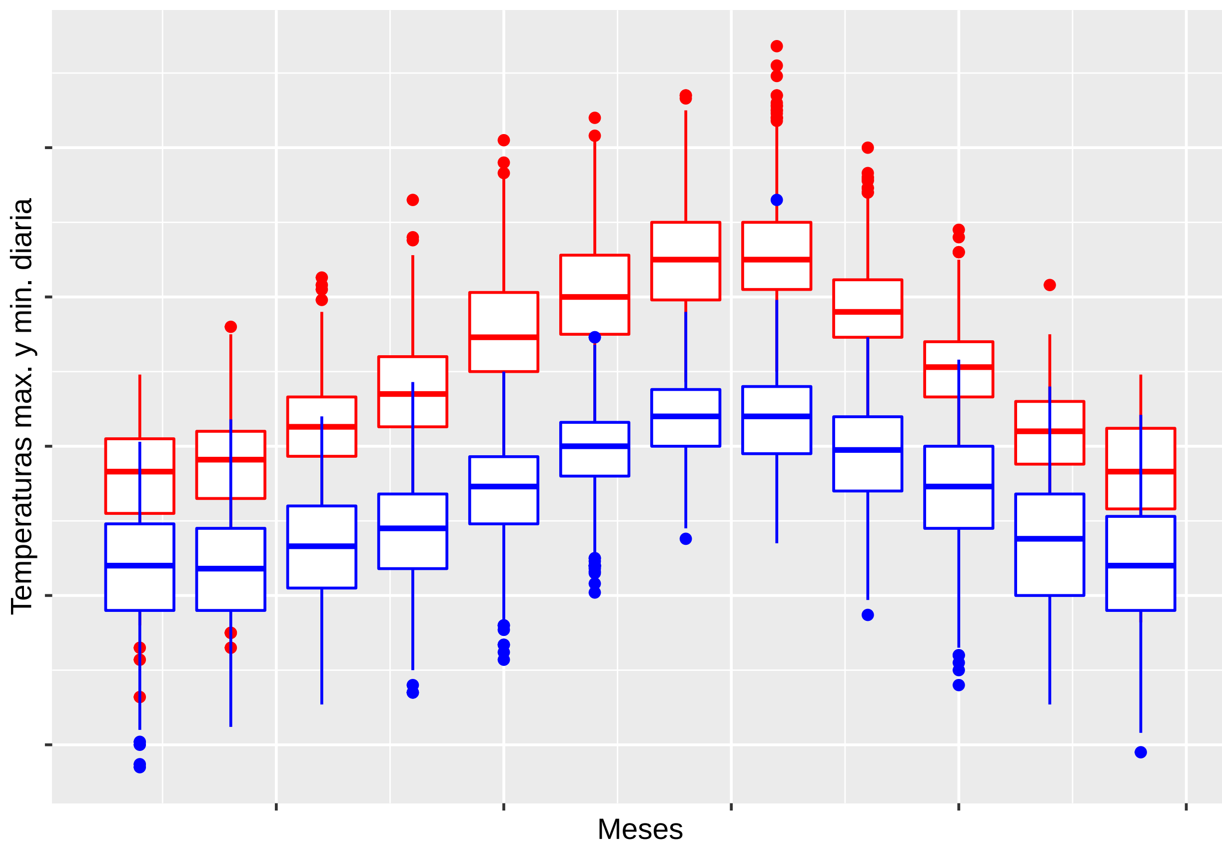
```
Mes<-factor(Mes)
plot(Mes, T_Maxima, col="blue",  xlab="Meses", ylab="Temp. maximas y minimas");grid()
plot(Mes, T_Minima, col="red", add=T)
legend(10.5, 34.5, legend=c("Maxima", "Minima"), cex=0.85, text.col=c("blue", "red"))
```




```
estacion <- heat.colors(12)
paleta_sel <- c(11,10,8,5,3,1,2,3,5,8,10,11)
plot(Mes, T_Maxima, col=estacion[paleta_sel], xlab="Meses", ylab="Temperatura", notch=T); grid()
plot(Mes, T_Minima, col="lightblue", notch=T, add=T)
```




```
g<- ggplot(datos_c, aes(x=Mes, y=T_Maxima))
g+geom_boxplot(aes(group=Mes), col="red")+
  ylab("Temperaturas max. y min. diaria")+xlab("Meses")+
  theme(axis.text=element_blank())+
  geom_boxplot(aes(y=T_Minima, group=Mes), col="blue")
```



Idem a) para el nivel de precipitaciones a lo largo de los meses de cada año.

Para ello lo primero que realizaremos es tratado de datos, usando la función par, la función aggregate… Y a continuación mostraremos usando R básico las precipitaciones a lo largo de los meses.

```
par(mfrow=(c(1,1)))
max_p <- aggregate(Precipitacion~Year, datos_c, max);max_p
```

```
##    Year Precipitacion
## 1  1987          54.3
## 2  1988          23.7
## 3  1989          29.0
## 4  1990          15.0
## 5  1991          30.3
## 6  1992          44.5
## 7  1993          59.5
## 8  1994          38.8
## 9  1995          32.5
## 10 1996          22.0
## 11 1997          32.0
## 12 1998          30.9
## 13 1999          42.5
## 14 2000          41.4
## 15 2001          26.3
## 16 2002          43.0
## 17 2003          29.5
## 18 2004          24.5
## 19 2005          22.0
```

```r
plot(max_p$Year, max_p$Precipitacion, ylab="Precipitaciones maximas anuales", xlab="A;os", col="red", t
```

c) Crear un data frame con los temperaturas máximas y mínimas por año.

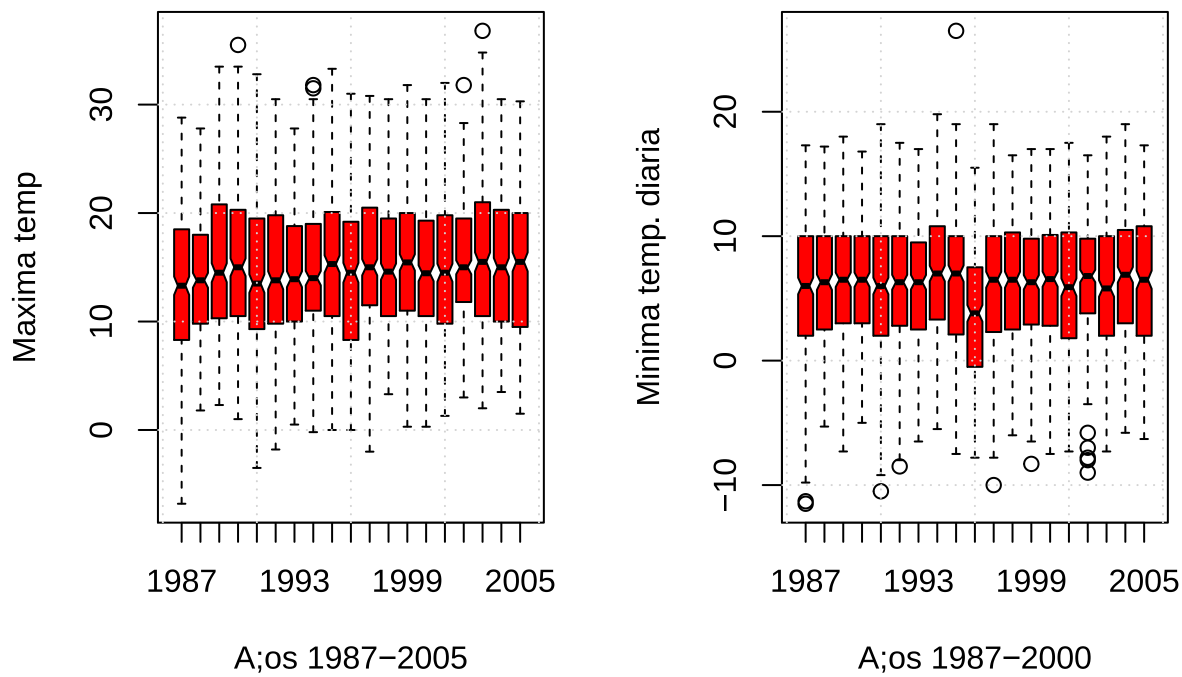
Lo primero que realizaremos es el data frame usando el comando aggregate y creando un nuevo data frame con data.frame. Por último lo mostraremos usando R básico.

```r
max_temp <- aggregate(T_Maxima~Year, datos_c, max)
min_temp <- aggregate(T_Minima~Year, datos_c, min)
temp <- data.frame(Year=unique(Year), T_Maxima=max_temp[,2], T_Minima=min_temp[,2])
kable(temp)
```

Year	T_Maxima	T_Minima
1987	28.8	-11.5
1988	27.8	-5.3
1989	33.5	-7.3
1990	35.5	-5.0
1991	32.8	-10.5
1992	30.5	-8.5
1993	27.8	-6.5
1994	31.8	-5.5
1995	33.3	-7.5
1996	31.0	-7.8
1997	30.8	-10.0
1998	30.5	-6.0
1999	31.8	-8.3
2000	30.5	-7.5

Year	T_Maxima	T_Minima
2001	32.0	-7.3
2002	31.8	-9.0
2003	36.8	-7.3
2004	30.5	-5.8
2005	30.3	-6.3

```
Year<- factor(Year)
par(mfrow=c(1,2))
plot(Year, T_Maxima, ylab="Maxima temp", xlab="A;os 1987-2005", col="red", notch=TRUE);grid()
plot(Year, T_Minima,  ylab="Minima temp. diaria", xlab="A;os 1987-2000", col="red", notch=TRUE); grid()
```

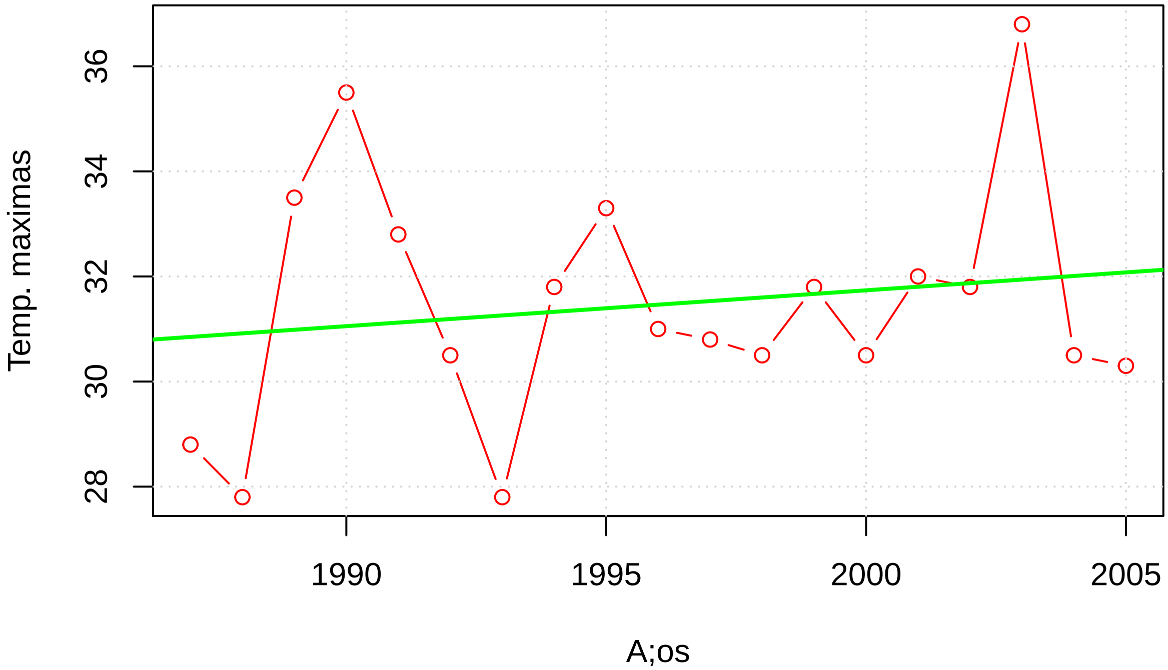


d) Utilizar la función lm() para ver la tendencia e influencia del cambio de temperaturas a lo largo de los años. Analizar si existe una relación con el fenómeno del calentamiento global. Razonar y justificar las respuestas.

Se puede observar gracias a las aproximaciones realizadas y los datos mostrados que la temperatura a lo largo de los años esta aumentando, parece que es un cambio poco drástico y aun así no tenemos suficientes datos para dar una opinión rigurosa y certera. Sin embargo un cambio en la temperatura, por muy pequeño que sea, influye de forma bestial en la biosfera terrestre. También cabe mencionar que a lo largo de los años el aumento de la población humana, el uso de recursos fósiles que contienen dióxido de carbono, etc… puede haber influido significativamente.

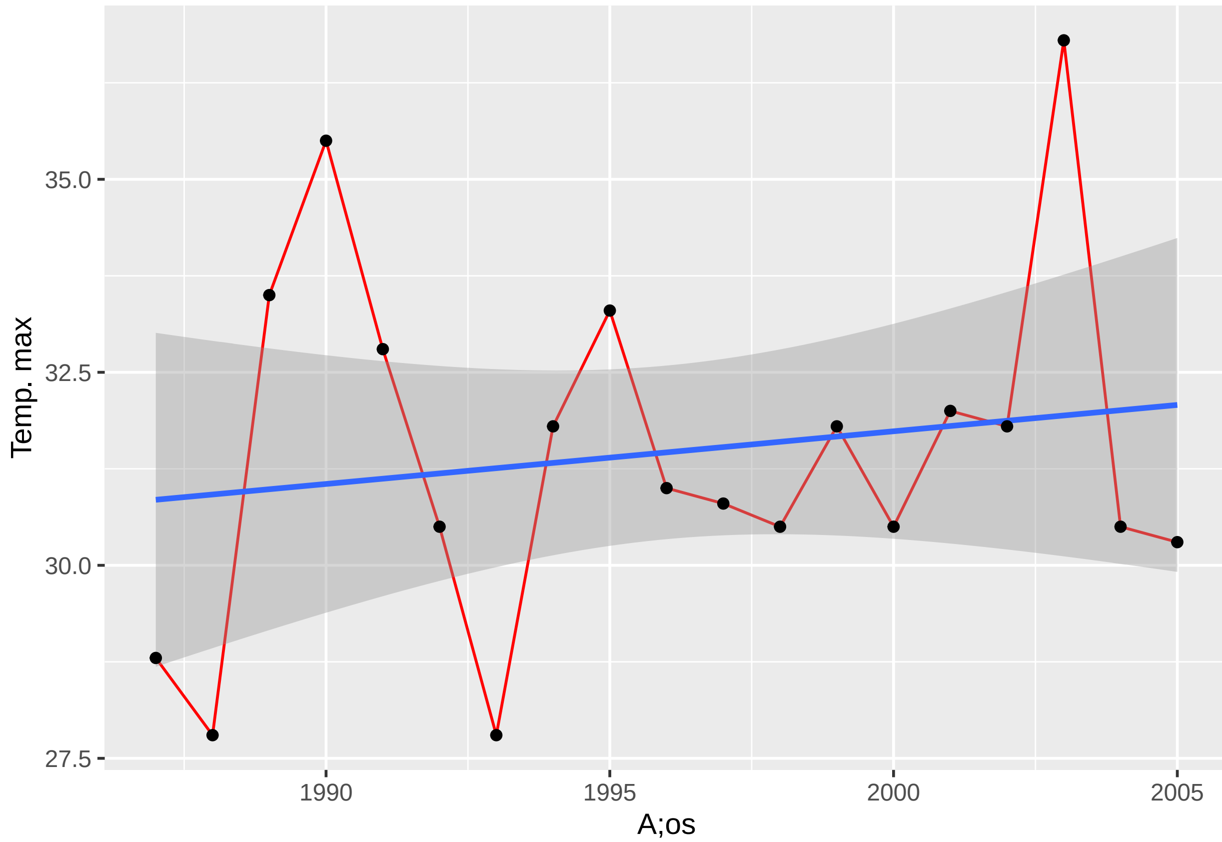
```
par(mfrow=c(1,1))
plot(max_temp$Year, max_temp$T_Maxima, ylab="Temp. maximas", xlab="A;os", col="red", type="b");grid()
```

```r
mod<- lm(max_temp$T_Maxima~max_temp$Year)
abline(mod, col="green", lwd=2)
```

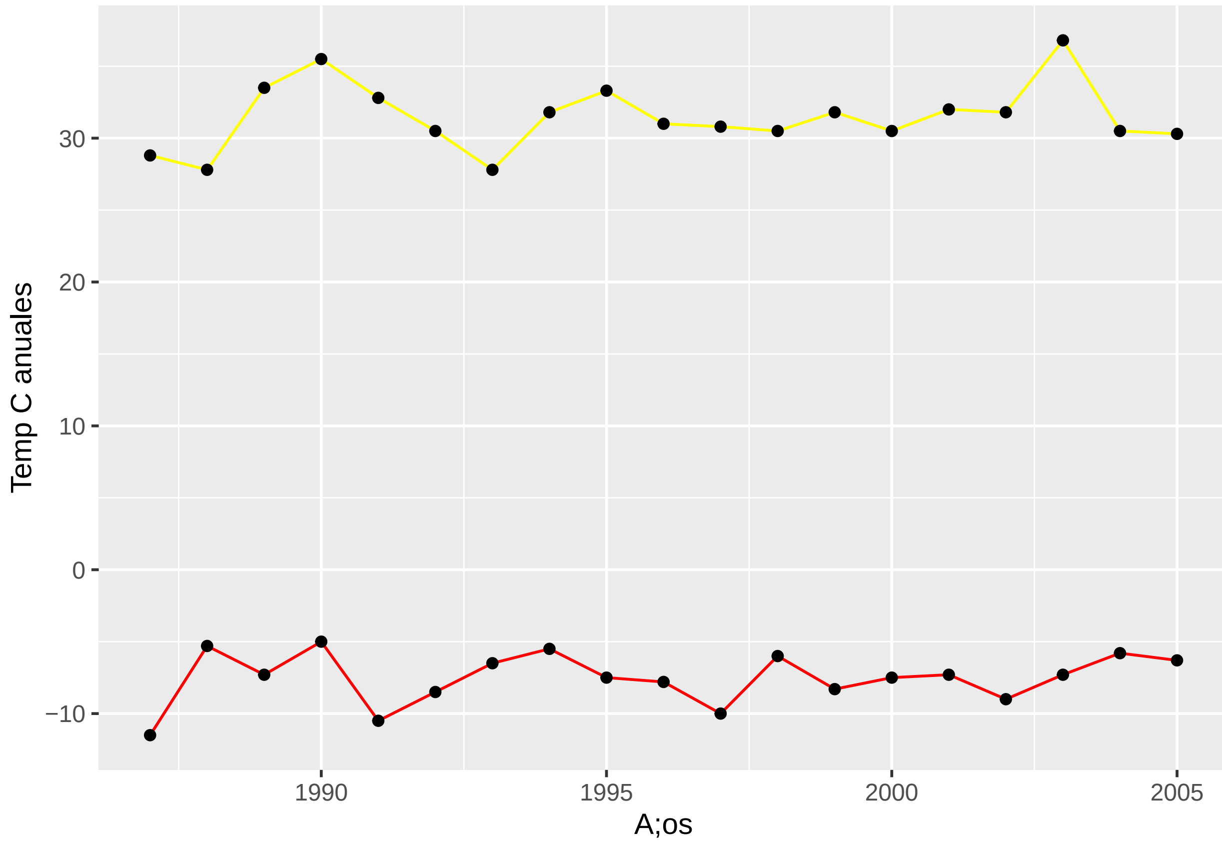



```r
g2<-ggplot(max_temp, aes(x=Year, y=T_Maxima))+ylab("Temp. max")+xlab("A;os")
g2+geom_line(col="red")+geom_smooth(method="lm")+geom_point()
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
g2<-ggplot(max_temp, aes(x=Year, y=T_Maxima))+
  ylab("Temp C anuales")+xlab("A;os")
g3<-g2+geom_line(col="yellow")+geom_point()
g4<-g3+geom_line(aes(x=min_temp$Year, y=min_temp$T_Minima), col="red")+
  geom_point(aes(x=min_temp$Year, y=min_temp$T_Minima));g4
```




```r
detach(datos_c)
```

Ejercicio 4. Lab 4

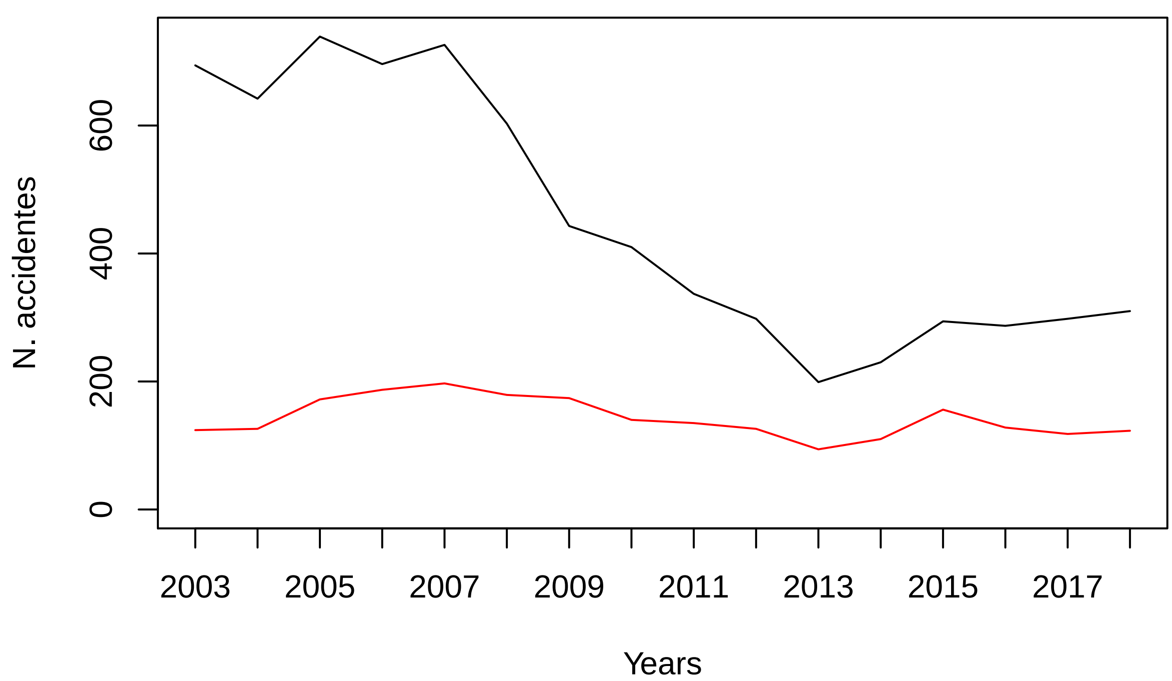
```
setwd("~/Escritorio/")
poblac <- read.csv("ejercicio4.csv", header=F, col.names=c("Edad", "Hombre", "Mujer"))

attach(poblac)
```

a)

Para realizar este trabajo tendremos que tener en cuenta 2 cosas primero que los datos de los años no se han descargado ordenados (tendremos que tenerlo en consideración a la hora de realizar los gráficos) y la comparativa a realizar entre hombres y mujeres.

A su vez hemos visualizado una tabla con los años máximos y mínimos segun mujer y hombre.

```
plot(Edad[16:1], Hombre[16:1], xaxt="n", type="l",
     ylab="N. accidentes", xlab="Years", ylim=c(0, max(Hombre)))
years <- as.character(seq(2003, 2018, 1))
axis(side = 1, at=Edad[16:1], labels=years)
points(Edad[16:1], Mujer[16:1], col="red", type="l")
```



```r
H_max <- Edad[which.max(Hombre)]; H_max
```

```
## [1] 2005
```

```r
H_min <- Edad[which.min(Hombre)]; H_min
```

```
## [1] 2013
```

```r
M_max <- Edad[which.max(Mujer)]; M_max
```

```
## [1] 2007
```

```r
M_min <- Edad[which.min(Mujer)]; M_min
```

```
## [1] 2013
```

```r
v_sing<- data.frame(Sexo=c("hombres", "mujeres"
), max_year=c(H_max, M_max), min_year=c(H_min, M_min));v_sing
```

```
##      Sexo max_year min_year
## 1 hombres     2005     2013
## 2 mujeres     2007     2013
```

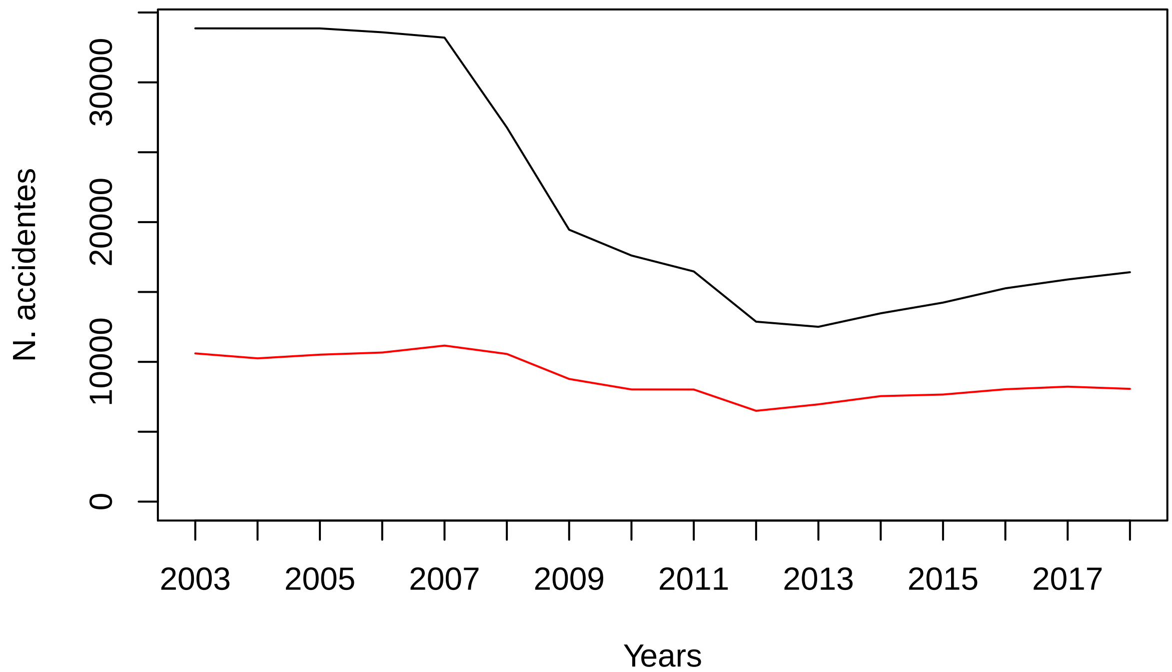
b)

Hemos realizado el mismo proceso anteriormente descrito pero con los datos totales proporcionados por la ISTAC.

```r
poblac <- read.csv("Ejercicio41.csv", header=F, col.names=c("Edad", "Hombre", "Mujer"))

attach(poblac)
```

```
## The following objects are masked from poblac (pos = 3):
##
##     Edad, Hombre, Mujer
```

```r
plot(Edad[16:1], Hombre[16:1], xaxt="n", type="l",
     ylab="N. accidentes", xlab="Years", ylim=c(0, max(Hombre)))
years <- as.character(seq(2003, 2018, 1))
axis(side = 1, at=Edad[16:1], labels=years)
points(Edad[16:1], Mujer[16:1], col="red", type="l")
```




```r
H_max <- Edad[which.max(Hombre)]; H_max
```

```
## [1] 2003
```

```r
H_min <- Edad[which.min(Hombre)]; H_min
```

```
## [1] 2013
```

```r
M_max <- Edad[which.max(Mujer)]; M_max
```

```
## [1] 2007
```

```r
M_min <- Edad[which.min(Mujer)]; M_min
```

```
## [1] 2012
```

```r
v_sing<- data.frame(Sexo=c("hombres", "mujeres"
), max_year=c(H_max, M_max), min_year=c(H_min, M_min));v_sing
```

```
##      Sexo max_year min_year
## 1 hombres     2003     2013
## 2 mujeres     2007     2012
```

c)

Podemos observar que definitivamente se han producido bastantes mejoras en los accidentes sobretodo en el entorno masculino puesto que la diferencia entre los datos observadoa antes de 2007 y los de 2017 es de casi 10.000 accidentes. Esto es un claro signo de que se esta abordando el problema con soluciones eficaces.

d)

Hemos usado la librería mgcv para predecir cuales son los sucesos que pasarán en 2019 y 2020 como podemos observar hemos realizado dos modelos uno lineal y otro cuadrático.

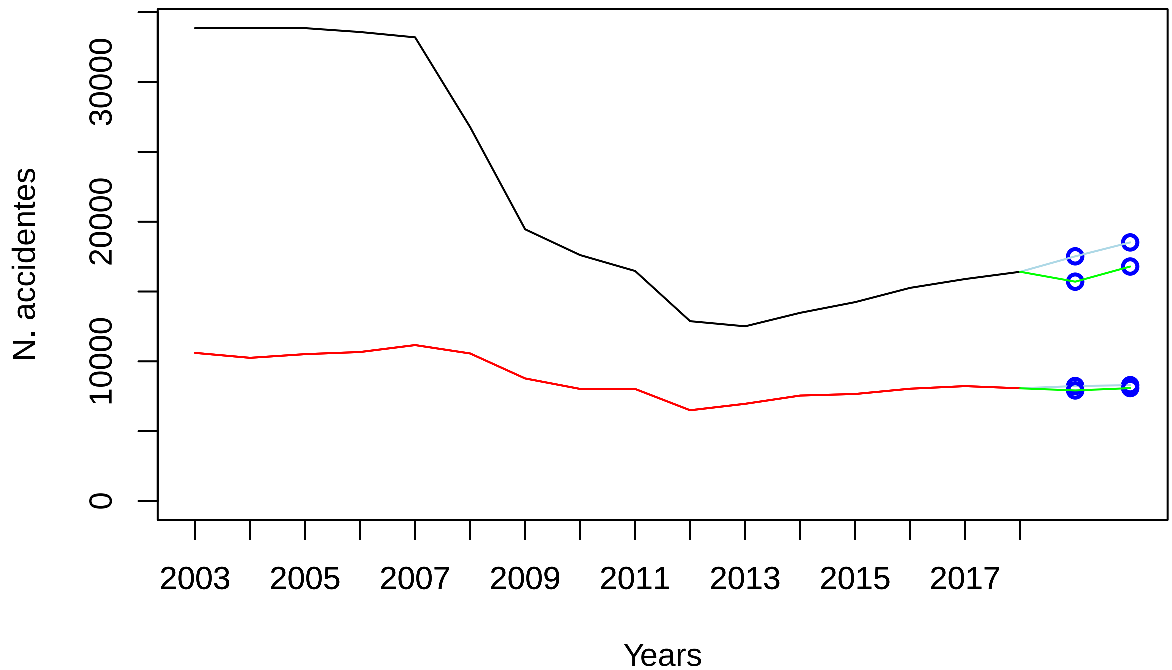
Vemos una tendencia ascendente en las predicciones salvo en la cuadrática masculina, que primero realiza un descenso y después vuelve a pronunciarse.

```
library(mgcv)
```

```
## Loading required package: nlme
```

```
## This is mgcv 1.8-33. For overview type 'help("mgcv-package")'.
```

```
modelo1<-gam(Hombre~s(Edad))
xv<-c(2019, 2020)
yv<-predict(modelo1, list(Edad=xv))
plot(Edad[16:1], Hombre[16:1], xaxt="n", type="l",
     ylab="N. accidentes", xlab="Years", ylim=c(0, max(Hombre)), xlim=c(2003, 2020))
years <- as.character(seq(2003, 2018, 1))
axis(side = 1, at=Edad[16:1], labels=years)
points(Edad[16:1], Mujer[16:1], col="red", type="l")
points(xv, yv, col="blue", lwd=2)

lines(c(2018, xv), c(Hombre[Edad==2018], yv), col='lightblue')

modelo2<-gam(Mujer~s(Edad))
xv<-c(2019, 2020)
yv<-predict(modelo2, list(Edad=xv))
points(xv, yv, col="blue", lwd=2)
lines(c(2018, xv), c(Mujer[Edad==2018], yv), col='lightblue')



modelo1<-lm(Hombre~Edad+I(Edad^2))
xv<-c(2019, 2020)
yv<-predict(modelo1, list(Edad=xv))
years <- as.character(seq(2003, 2018, 1))
axis(side = 1, at=Edad[16:1], labels=years)
points(Edad[16:1], Mujer[16:1], col="red", type="l")
points(xv, yv, col="blue", lwd=2)

lines(c(2018, xv), c(Hombre[Edad==2018], yv), col='green')

modelo2<-lm(Mujer~Edad+I(Edad^2))
xv<-c(2019, 2020)
yv<-predict(modelo2, list(Edad=xv))
points(xv, yv, col="blue", lwd=2)
lines(c(2018, xv), c(Mujer[Edad==2018], yv), col='green')
```


N. accidentes
0
10000
20000
30000
2003
2005
2007
2009
2011
2013
2015
2017
Years

Ejercicio 1. Lab 3

a y b)

Para realizar estos dos apartados usaremos la librería dplyr que nos permitirá crear una tabla con todos los valores que buscamos, ordenandolos y agrupandolos.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
head(toyota)
```

```
##   Precio Antiguedad_Meses    KM Combustible    CV Metalizado Automatico   CC
## 1   9091               23 46986      Diesel 88.74         No         Si 2000
## 2   9318               23 72937      Diesel 88.74         No         Si 2000
## 3   9500               24 41711      Diesel 88.74         No         Si 2000
## 4  10409               26 48000      Diesel 88.74         Si         Si 2000
## 5   9318               30 38500      Diesel 88.74         Si         Si 2000
## 6   8591               32 61000      Diesel 88.74         Si         Si 2000
##   N_Puertas Peso
## 1         3 1165
## 2         3 1165
## 3         3 1165
## 4         3 1165
## 5         3 1170
## 6         3 1170
```

```
summary(toyota)
```

```
##      Precio      Antiguedad_Meses       KM          Combustible
##  Min.   :  773   Min.   : 1.00    Min.   :     1   Length:1436
##  1st Qu.: 4500   1st Qu.:44.00    1st Qu.: 43000   Class :character
##  Median : 5818   Median :61.00    Median : 63390   Mode  :character
##  Mean   : 6574   Mean   :55.95    Mean   : 68533
##  3rd Qu.: 7682   3rd Qu.:70.00    3rd Qu.: 87021
##  Max.   :26364   Max.   :80.00    Max.   :243000
##        CV          Metalizado         Automatico              CC
##  Min.   : 68.03   Length:1436        Length:1436        Min.   :1300
##  1st Qu.: 88.74   Class :character   Class :character   1st Qu.:1400
##  Median :108.46   Mode  :character   Mode  :character   Median :1600
##  Mean   :100.08                                         Mean   :1567
##  3rd Qu.:108.46                                         3rd Qu.:1600
```

```
##  Max.   :189.31                                         Max.   :2000
##    N_Puertas          Peso
##  Min.   :2.000   Min.   :1000
##  1st Qu.:3.000   1st Qu.:1040
##  Median :4.000   Median :1070
##  Mean   :4.033   Mean   :1072
##  3rd Qu.:5.000   3rd Qu.:1085
##  Max.   :5.000   Max.   :1615
```

```r
tail(toyota)
```

```
##      Precio Antiguedad_Meses    KM Combustible      CV Metalizado Automatico
## 1431   4500               80 23000    Gasolina  84.796         Si         Si
## 1432   3636               69 20544    Gasolina  84.796         No         Si
## 1433   6677               72 19000    Gasolina  84.796         Si         Si
## 1434   4545               71 17016    Gasolina  84.796         Si         Si
## 1435   3409               70 16916    Gasolina  84.796         No         Si
## 1436   3136               76     1    Gasolina 108.460         Si         Si
##        CC N_Puertas Peso
## 1431 1300         3 1015
## 1432 1300         3 1025
## 1433 1300         3 1015
## 1434 1300         3 1015
## 1435 1300         3 1015
## 1436 1600         5 1114
```

```r
agg <- toyota %>%
    group_by(Combustible, Automatico) %>%
    summarise(mean = mean(Precio), std = sd(Precio), mediana = median(Precio), quantile=quantile(Precio)
```

```
## `summarise()` regrouping output by 'Combustible', 'Automatico' (override with `.groups` argument)
```

```r
agg
```

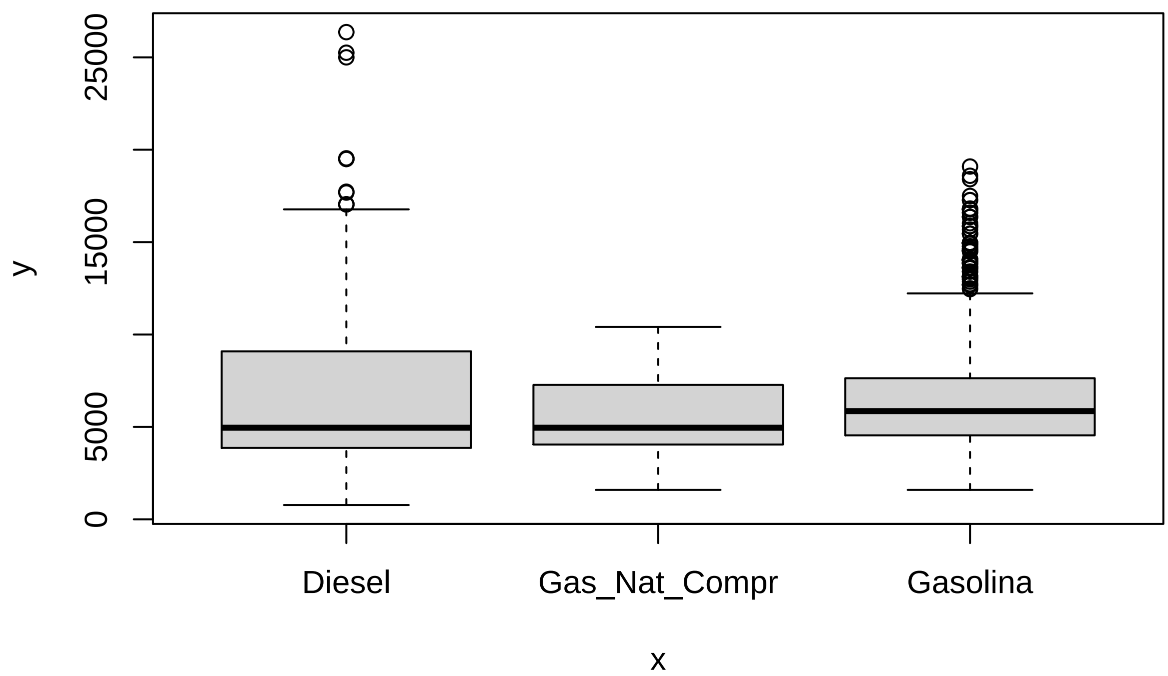
```
## # A tibble: 25 x 6
## # Groups:   Combustible, Automatico [5]
##    Combustible   Automatico  mean   std mediana quantile
##    <chr>         <chr>      <dbl> <dbl>   <dbl>    <dbl>
##  1 Diesel        Si         7086. 5033.    4955      773
##  2 Diesel        Si         7086. 5033.    4955     3864
##  3 Diesel        Si         7086. 5033.    4955     4955
##  4 Diesel        Si         7086. 5033.    4955     9091
##  5 Diesel        Si         7086. 5033.    4955    26364
##  6 Gas_Nat_Compr No         4045    NA     4045     4045
##  7 Gas_Nat_Compr No         4045    NA     4045     4045
##  8 Gas_Nat_Compr No         4045    NA     4045     4045
##  9 Gas_Nat_Compr No         4045    NA     4045     4045
## 10 Gas_Nat_Compr No         4045    NA     4045     4045
## # ... with 15 more rows
```

c)

A continuación observaremos una seríe de gráficos de comparación entre los precios de coches con sus respectivos combustibles y si estos son automáticos o no.

Como podemos observar vemos que los coches mas caros suelen ser los de Gassolina y aquellos que son automáticos. A su vez podemos destacar la gran presencia de outliers de los que tienen gasolina, aumentando su promedio de forma considerable siendo este mayor que aquellos con diesel.

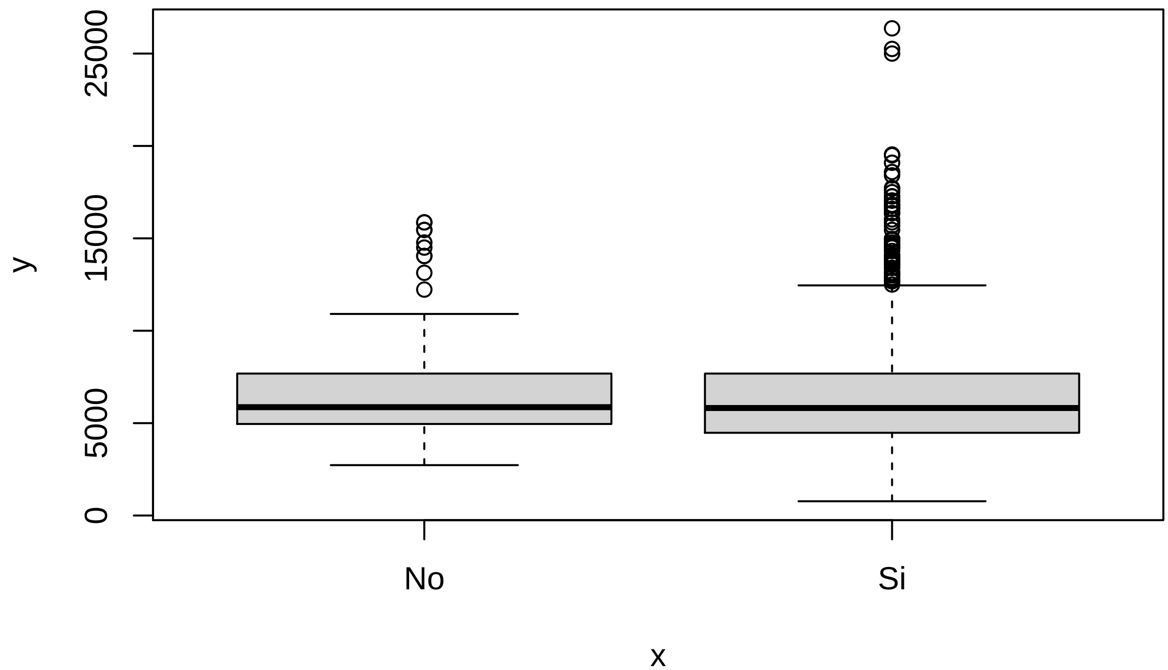
```
n <- plot(as.factor(toyota$Combustible),toyota$Precio)
```




```
n$out
```

```
##  [1] 17045 17045 26364 25000 25250 19500 19500 17682 19536 17045 17727 16364
## [13] 15864 14955 14636 16364 17273 16818 17500 13136 12723 13136 16773 15455
## [25] 16773 14045 13864 13136 13136 14045 14045 14955 13864 13591 14045 12500
## [37] 13136 12682 14000 13864 12682 14082 13636 13636 14500 13909 12682 13136
## [49] 13091 14045 14045 16591 13136 13591 14909 18591 14955 13636 14045 15455
## [61] 19091 14500 15864 12455 14955 13591 14545 16591 14545 14000 14773 14773
## [73] 14045 15682 14636 14545 12864 14955 14955 15864 15455 12995 13405 18409
## [85] 14545 14045 16773 14955 14045 14955 16773 17273 13636 13818 16023 16364
## [97] 12995 13405 14045
```

```
p <- plot(as.factor(toyota$Automatico),toyota$Precio)
```



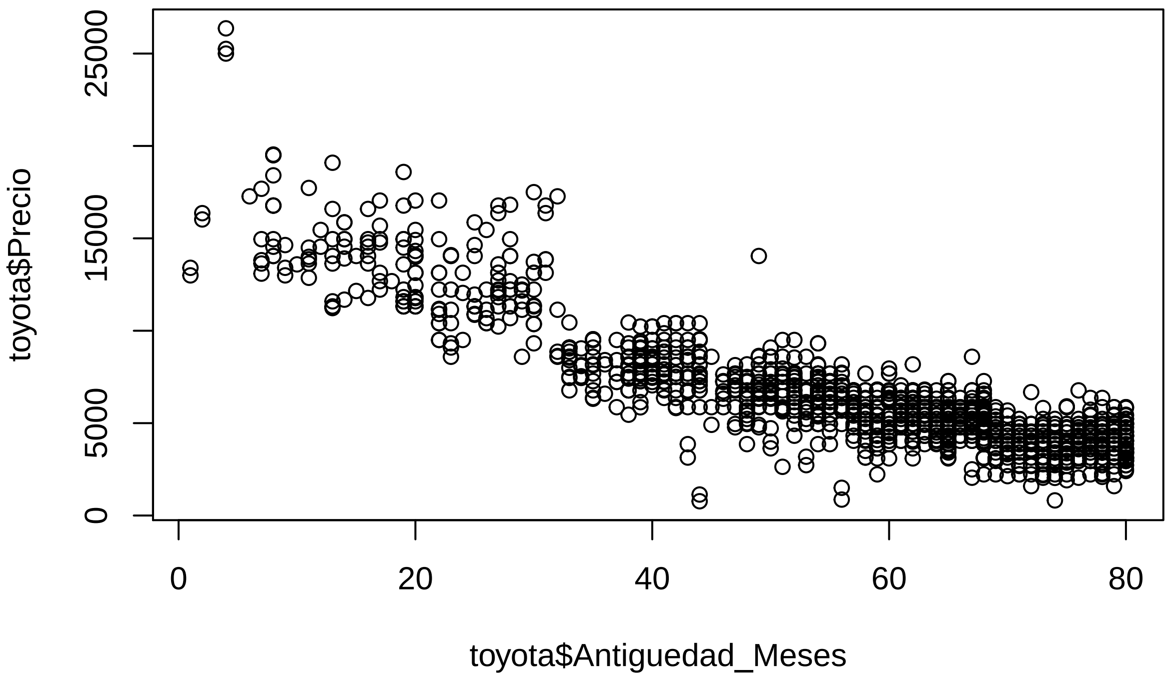

```r
p$out
```

```
##   [1] 12227 15455 14045 14045 13136 14500 15455 15864 14773 15864 13727 16364
##  [13] 15864 14955 14636 16364 17273 16818 17500 13136 12723 13136 14091 13136
##  [25] 13136 16773 13136 16773 13864 13136 13136 14045 17045 14045 14955 13864
##  [37] 13591 12500 12682 13136 16773 17045 14955 14000 14955 13864 12682 14082
##  [49] 13636 13636 13909 12682 13136 26364 25000 25250 19500 19500 17682 19536
##  [61] 16773 13091 14318 17045 14045 14955 14045 16591 13136 13591 17727 14909
##  [73] 18591 14955 13636 14045 19091 14500 14955 13591 14545 16591 14545 14000
##  [85] 14773 14045 15682 14636 14545 12864 14955 14955 15455 12995 13405 18409
##  [97] 14545 14045 16773 14955 14045 14955 16773 17273 13636 13818 16023 16364
## [109] 12995 13405 14045
```

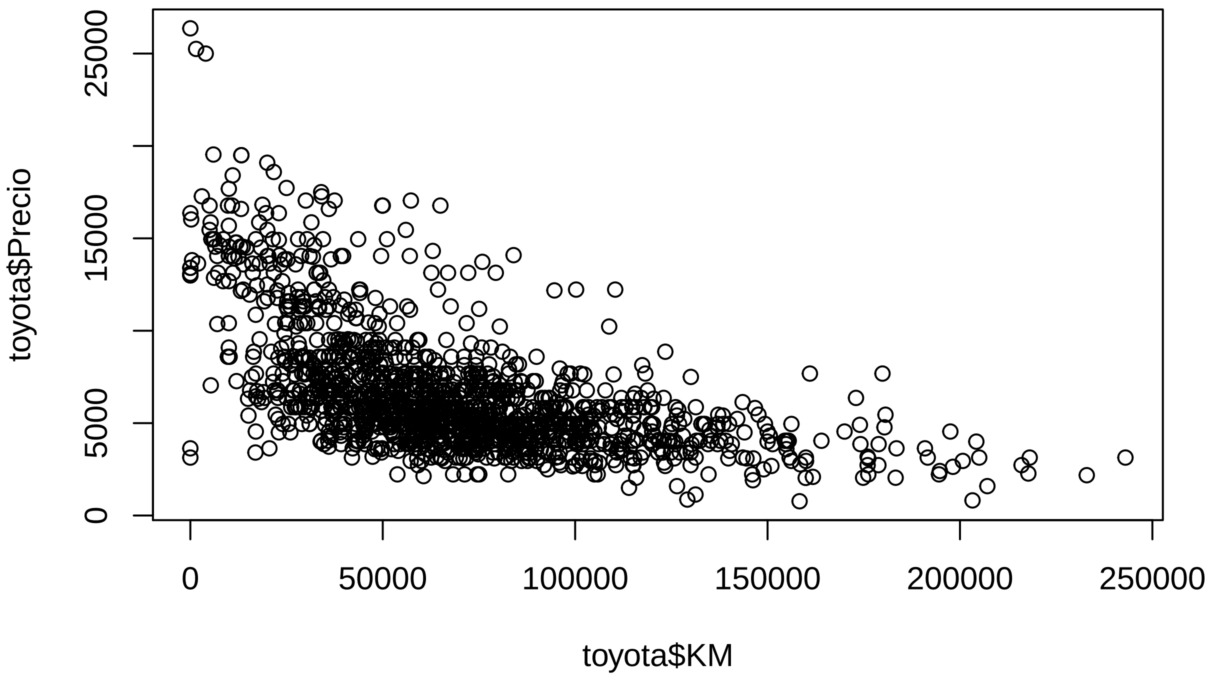
d)

Como observamos hay una fuerte correlación entre los km recorridos con el coche en su edad como es obvio. También podemos observar que estos dos factores influyen de manera inversamente proporcional en el precio es decir mientras mayores son los km y los años mas barato saldrá el coche.

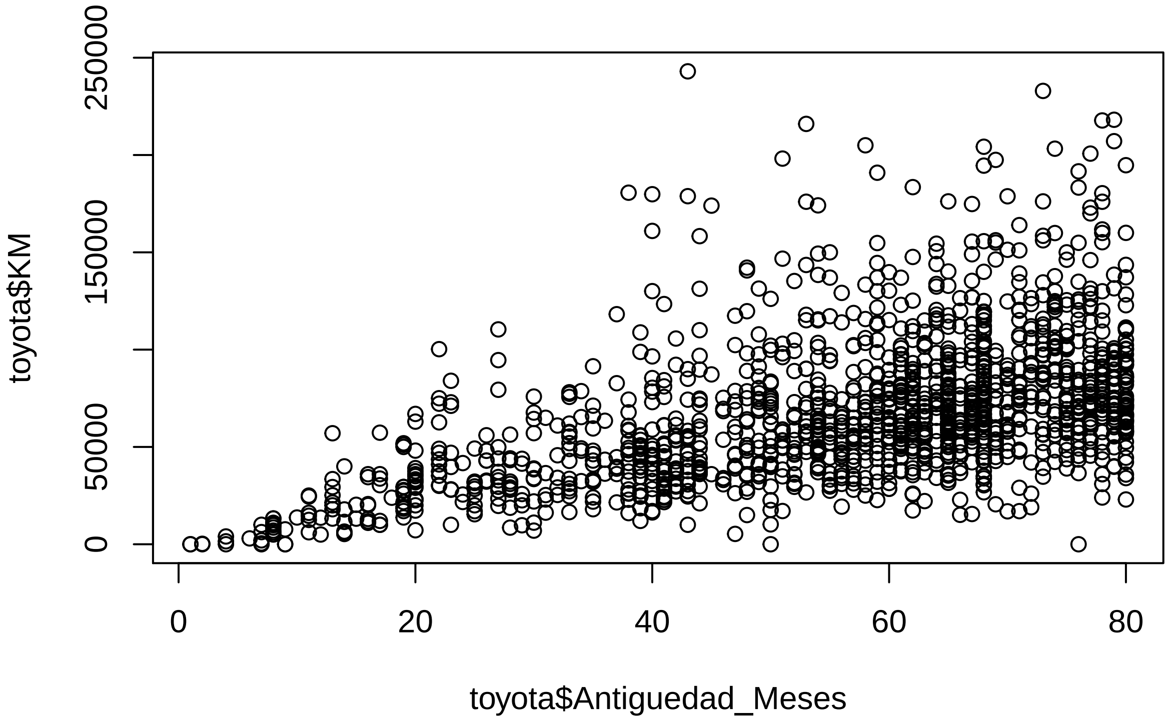
```r
plot(toyota$Antiguedad_Meses, toyota$Precio)
```




```
plot(toyota$KM, toyota$Precio)
```



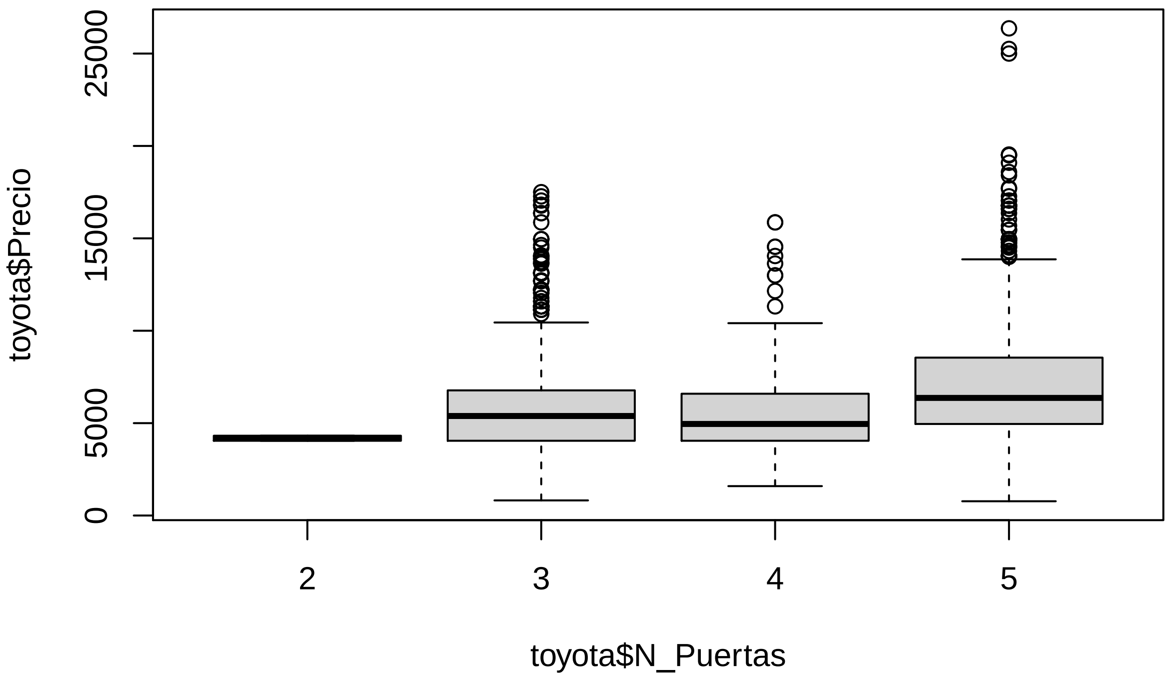

```r
plot(toyota$Antiguedad_Meses, toyota$KM)
```



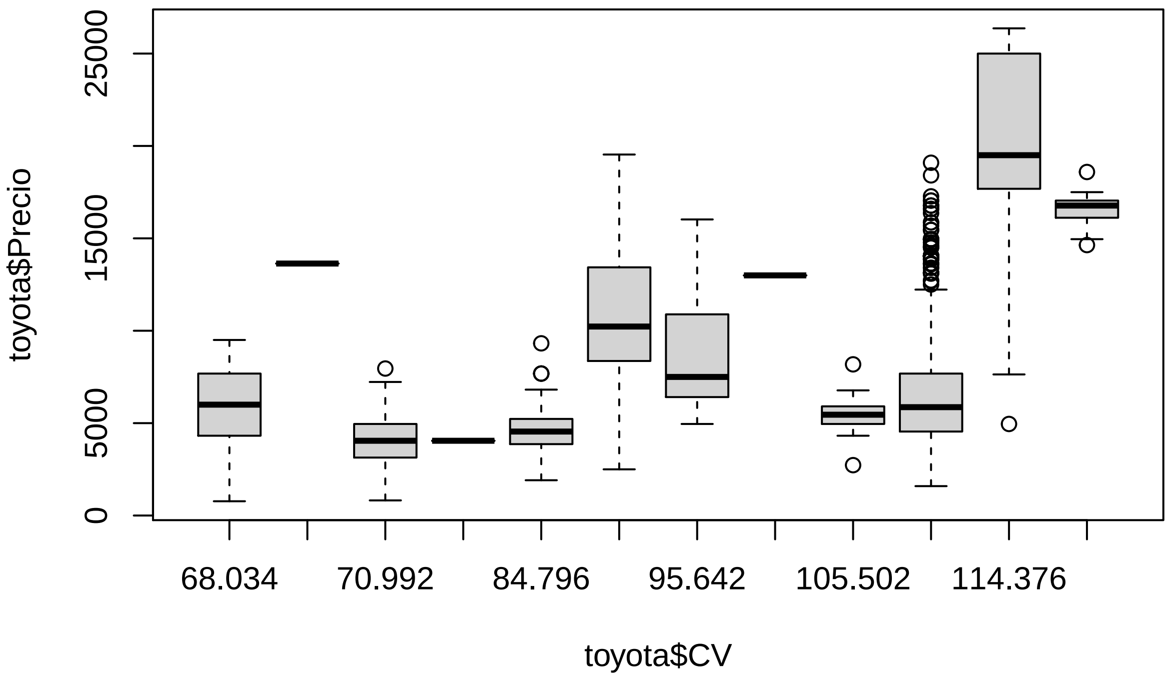
e)

Como podemos ver, los coches que mas puertas tienen se ven superiores al resto en cuanto precio y aunque menos acentuado pasa lo mismo con la potencia de los mismos.

```
boxplot(toyota$Precio~toyota$N_Puertas);
```




```r
boxplot(toyota$Precio~toyota$CV);
```



g)

Sin duda los criterios más destacados a la hora de comprar un coche son el kilometraje y los años de antiguedad que disminuyen el precio base de forma considerable. A su vez es obvio que aquellos coches con mas puertas y con mas potencia serán mas caros, seguramente su precio base también sería mas caro que el resto.

Ejercicio 3. Lab 3

Analizar el tipo de función de probabilidad subyacente y explicar sus características.

Se puede observar claramente que la probabilidad subyacente es la binomial. Esta es una distribución de probabilidad discreta que describe el numero de exitos al realizar un numero x de experimentos que son independientes entre sí a travez de una variable aleatoria.

Esto se ve claro en este problema al ser tener dos opciones o bien que la fruta este estropeada o bien que no. A su vez que una fruta este podrida no afecta al resto son fenomenos como hemos dicho antes independientes.

Calcular la probabilidad de que el número de kg defectuosos sea de 15 a la semana

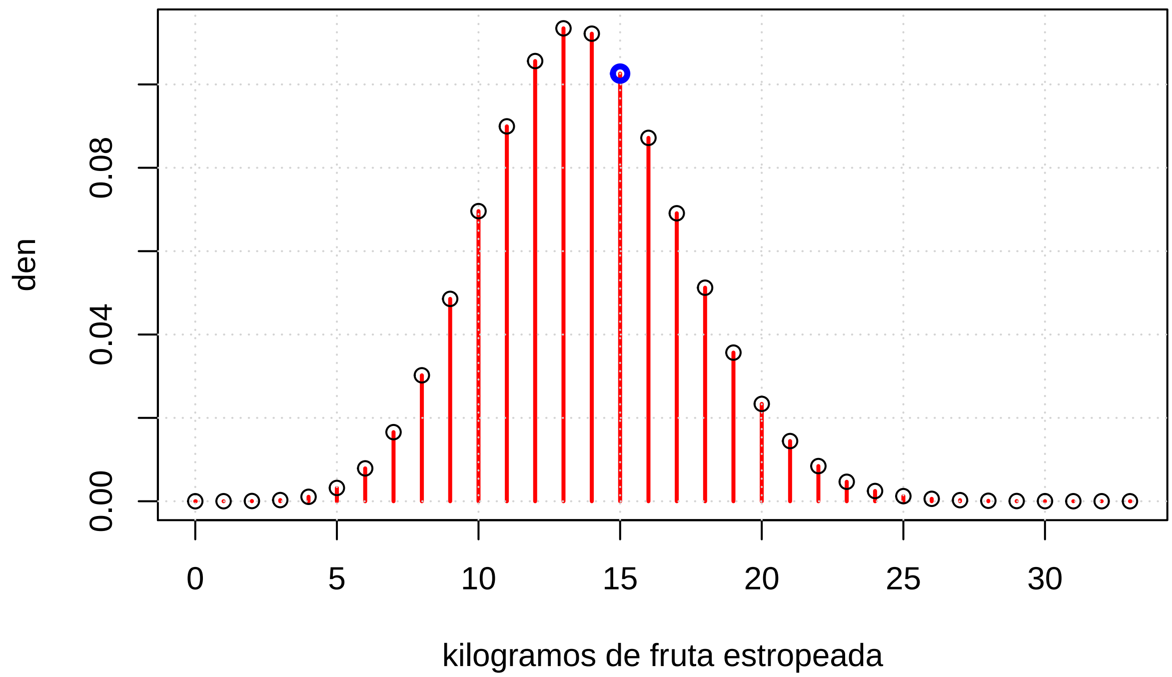
Para realiar este calculo primero definiremos la distribución a travez de los parametros 125 kg y 11% de probabilidad de defecto.

Mostraremos con plot la distribución de forma gráfica.

Usando dbinom calcularemos la probabilidad de que el numero de defectuosos sean 15 concretamente. Por ultimo lo mostraremos en nuestra grafica usando point

```
n <- 125; p<-0.11; x <- seq(0,  33, 1); den <- dbinom(x, n, p)
plot(x, den, type="h", lwd=2, col="red", xlab="kilogramos de fruta estropeada"); grid(); points(x, den)

#a) PROBABILIDAD DE 15 DEFECTUOSOS

p1 <- dbinom(15, n, p); p1; points(15, p1, col="blue", lwd=3)
```

```
## [1] 0.1026005
```



c) Si se paga el kilo de fruta a 3.5 euros, encontrar la probabilidad de que las pérdidas por este concepto asciendan a más de 55 euros/semana.

Si el kg de fruta cuesta 3.5 euros para calcular la cantidad de fruta con la que perderemos 55 simplemente realizamos la division entre ambos. El resultado de esta no es un numero entero por lo tanto lo aproximamos al siguiente natural el cual es 16.

La probabilidad de perder mas de 55 euros sera 1 menos la probabilidad de perder menos de 55. Esta segunda la calcularemos usando pbinom.

Por ultimo mostraremos por pantalla visualmente las probabilidades obtenidas.

```
#b) Si se paga el kg de fruta a 3.5 euros encontrar la probabilidad...

k <- ceiling(55/3.5);k
```

```
## [1] 16
```

```
s <- 1 - pbinom(k, n, p);p
```

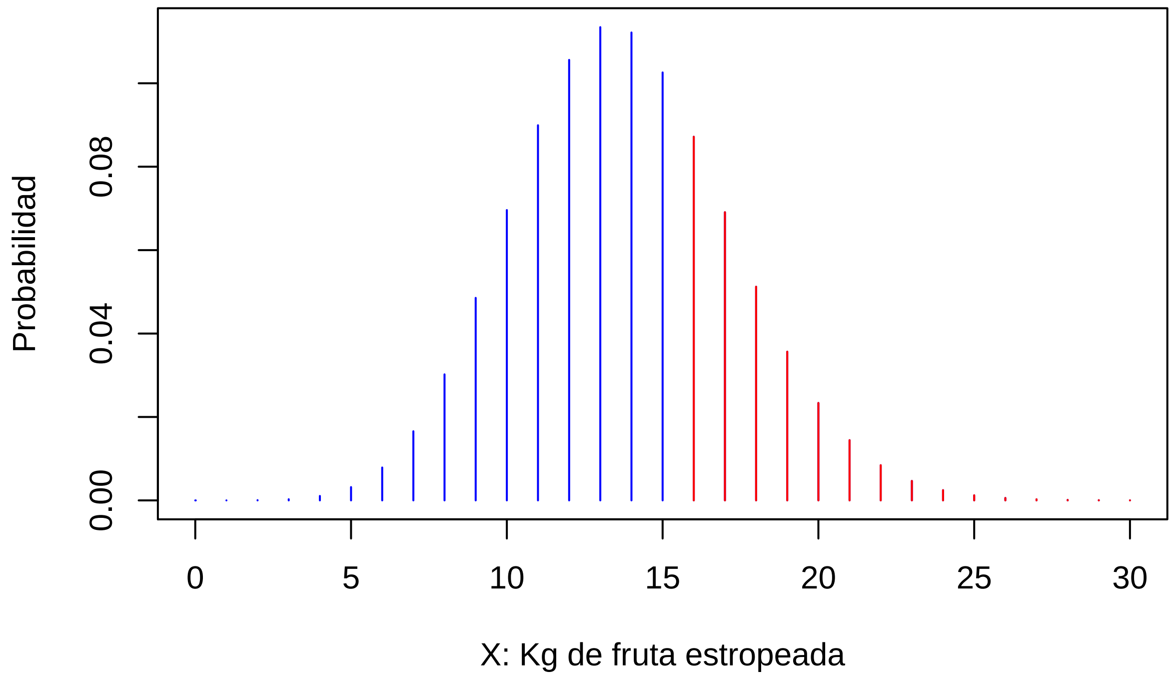
```
## [1] 0.11
```

```
poi <- dbinom(0:30, n, p); poi
```

```
##  [1] 4.717923e-07 7.288926e-06 5.585446e-05 2.830378e-04 1.066957e-03
##  [6] 3.191281e-03 7.888559e-03 1.657484e-02 3.021649e-02 4.855008e-02
## [11] 6.960664e-02 8.994116e-02 1.056051e-01 1.134547e-01 1.121799e-01
## [16] 1.026005e-01 8.718163e-02 6.908842e-02 5.123411e-02 3.566088e-02
## [21] 2.335988e-02 1.443588e-02 8.434447e-03 4.668407e-03 2.452225e-03
## [26] 1.224459e-03 5.820679e-04 2.637836e-04 1.141086e-04 4.717315e-05
```

```
## [31] 1.865725e-05
```

```r
plot(0:30, poi, xlab="X: Kg de fruta estropeada", ylab="Probabilidad", col="blue", type="h")
points(k:30, dbinom(k:30, n, p), type="h", col="red")
```



d) ¿Qué valor de la variable deja por debajo de sí el 75% de la probabilidad?

Para calcular esto es tan sencillo como ver el Q75 calculandolo con qbinom. Como podemos observar en las gráficas mostradas a continuación vemos que este valor es el 16.

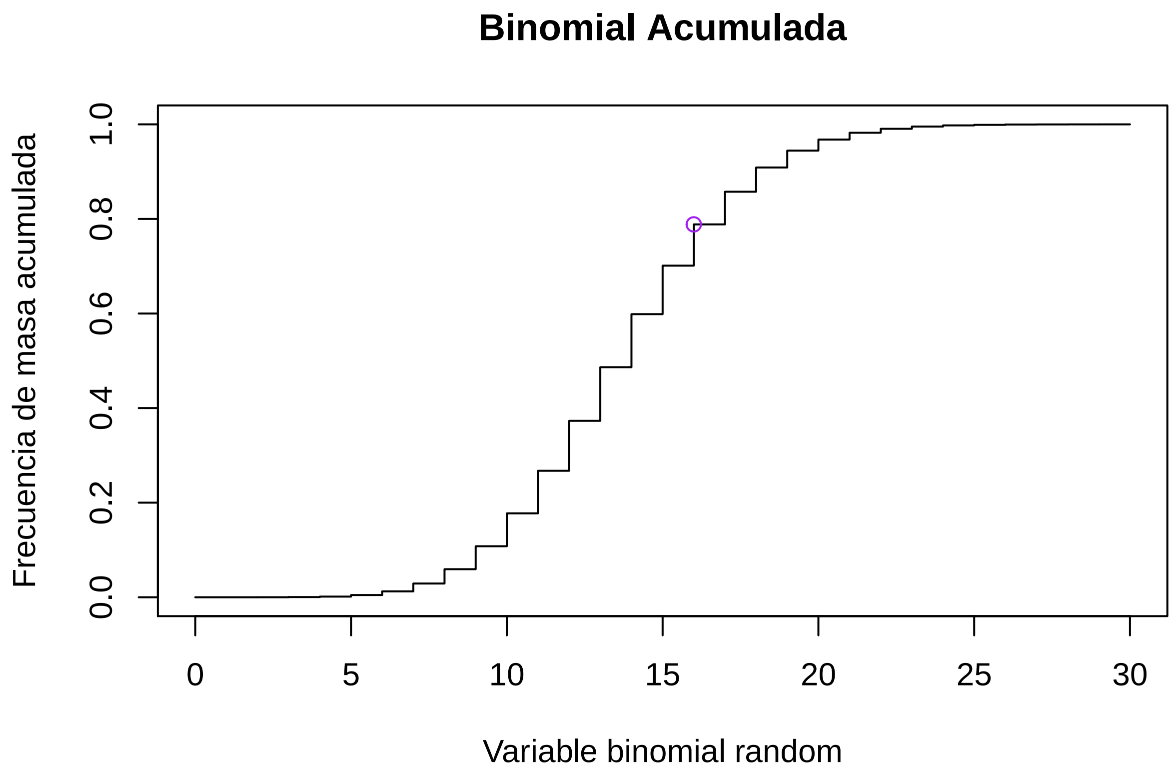
Lo primero que nos da a pensar esque es herroneo puesto que el dieciseis como hemos visto anteriormente deja por debajo aproximadamente el 80% de la probabilidad sin embargo al ser discreta seguramente el 75 no alcanzaba el 75% por lo tanto cogio el siguiente valor.

Esto lo observamos claramente en la gráfica acumulada.

```r
#probabilidad acumulada
x <- 0:30
c<- pbinom(x, 125, 0.11)
plot(x, c, xlab="Variable binomial random", ylab="Frecuencia de masa acumulada", main="Binomial Acumulad
#c) que  de la variable deja por debajo de si el 75 por cient
v <- qbinom(0.75, n, p);v
```

```
## [1] 16
```

```r
points(v, pbinom(v, n, p), col="purple")
```




```
dinero <- v*3.5; dinero
```

```
## [1] 56
```

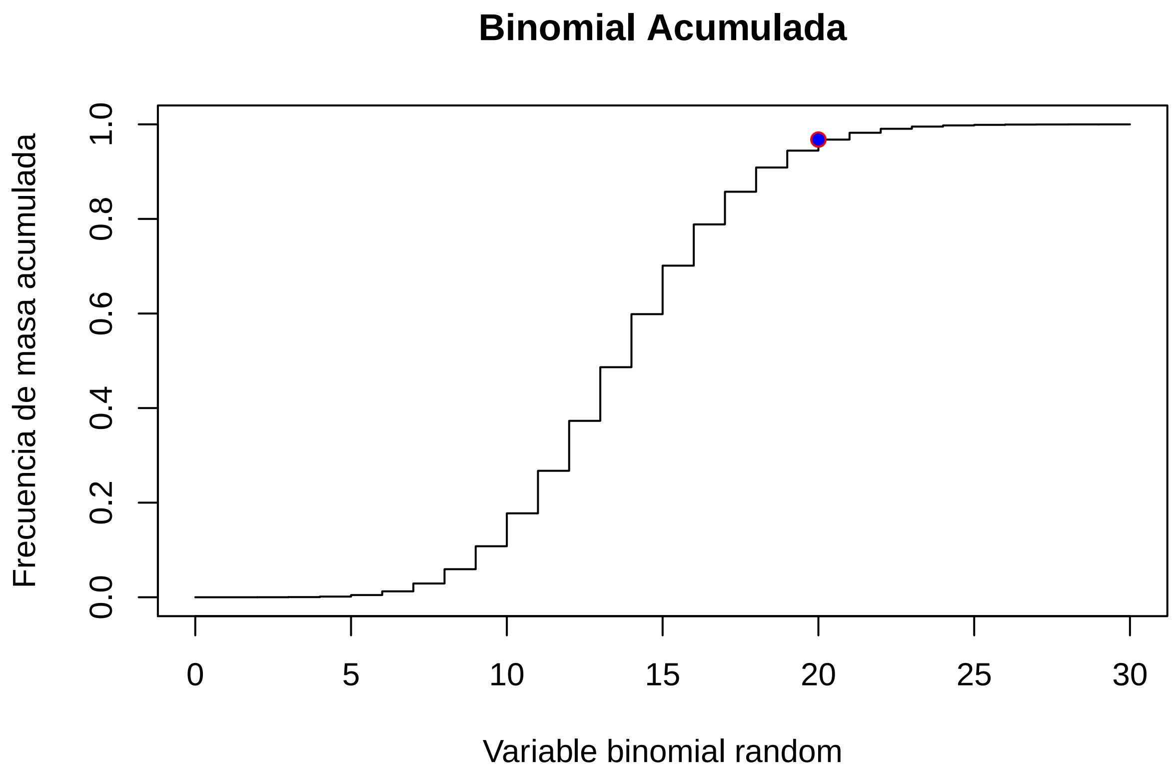
e) Calcular el percentil 95% de la distribución y estimar las pérdidas en euros en este caso

Para ello usaremos la funcion qbinom a la que le pasaremos 0.95. Una vez realizado esto simlemente para calcular las predidas multiplicaremos el dato obtenido por 3.5eur/kg

```
p95 <-qbinom(0.95, n, p);p95
```

```
## [1] 20
```

```
plot(x, c, xlab="Variable binomial random", ylab="Frecuencia de masa acumulada", main="Binomial Acumula
points(p95, pbinom(p95, n, p), col="red", ylab="f(x)", pch= 21, bg="blue")
```



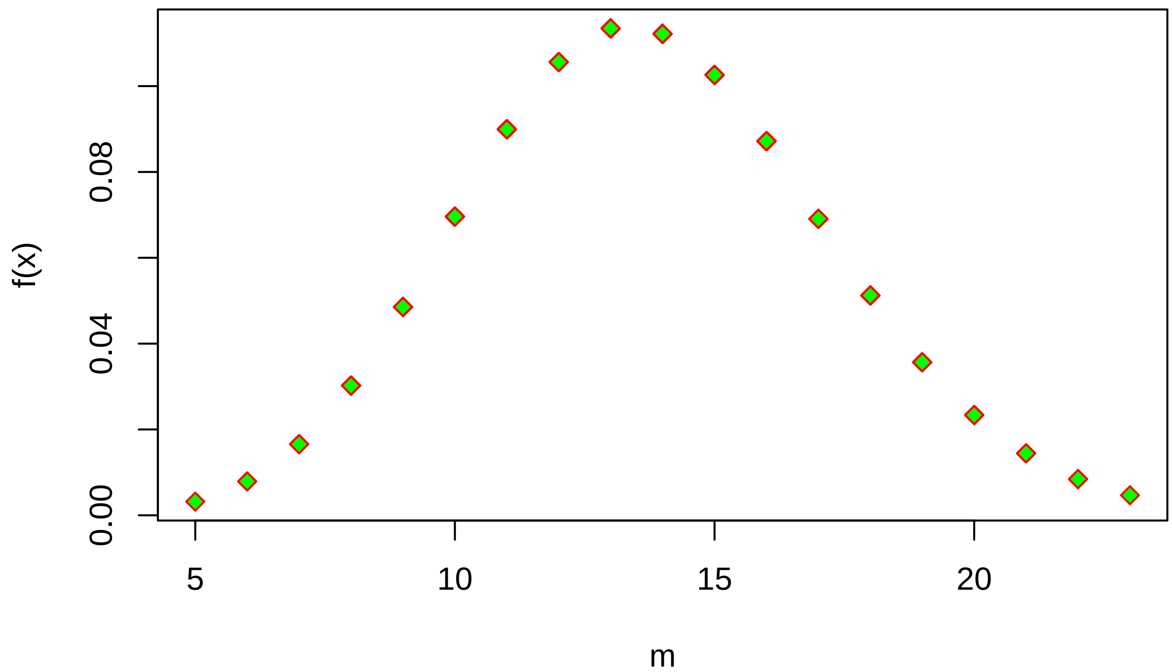

```
perd <- p95*3.5; perd
```

```
## [1] 70
```

##f) Obtener una muestra de tamaño 250 de esta distribución.g) Representar gráficamente la muestra de f) mediante un diagrama de barras y comparar éste con las frecuencias esperadas según el modelo que genera los datos.

Para ello pondremos como semilla de generacion la 32420, a su vez calcularemos la muestra con rbinom y por ultimo usando plot, hist y points superpondremos la distribucion sobre la muestra calculada y como podemos observar coinciden en gran medida.

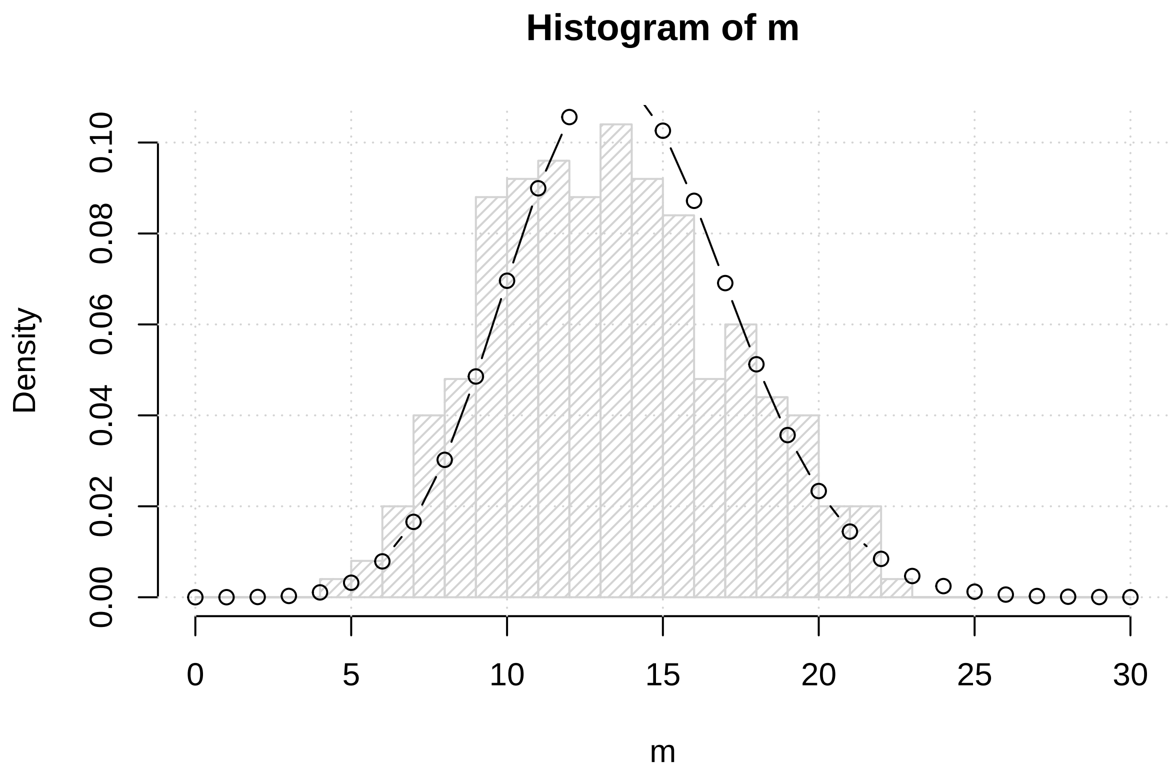
```
set.seed(32420)
t <-250; m <- rbinom(t, n, p)
plot(m, dbinom(m, n, p),col="red", ylab="f(x)", pch=23, bg="green")
```




```r
hist(m, dp1="blue", density=25, breaks=x, freq = F)
```

```
## Warning in plot.window(xlim, ylim, "", ...): "dp1" is not a graphical parameter

## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...): "dp1"
## is not a graphical parameter

## Warning in axis(1, ...): "dp1" is not a graphical parameter

## Warning in axis(2, ...): "dp1" is not a graphical parameter
```

```r
grid()
points(x, dbinom(x, n, p), type="b")
```

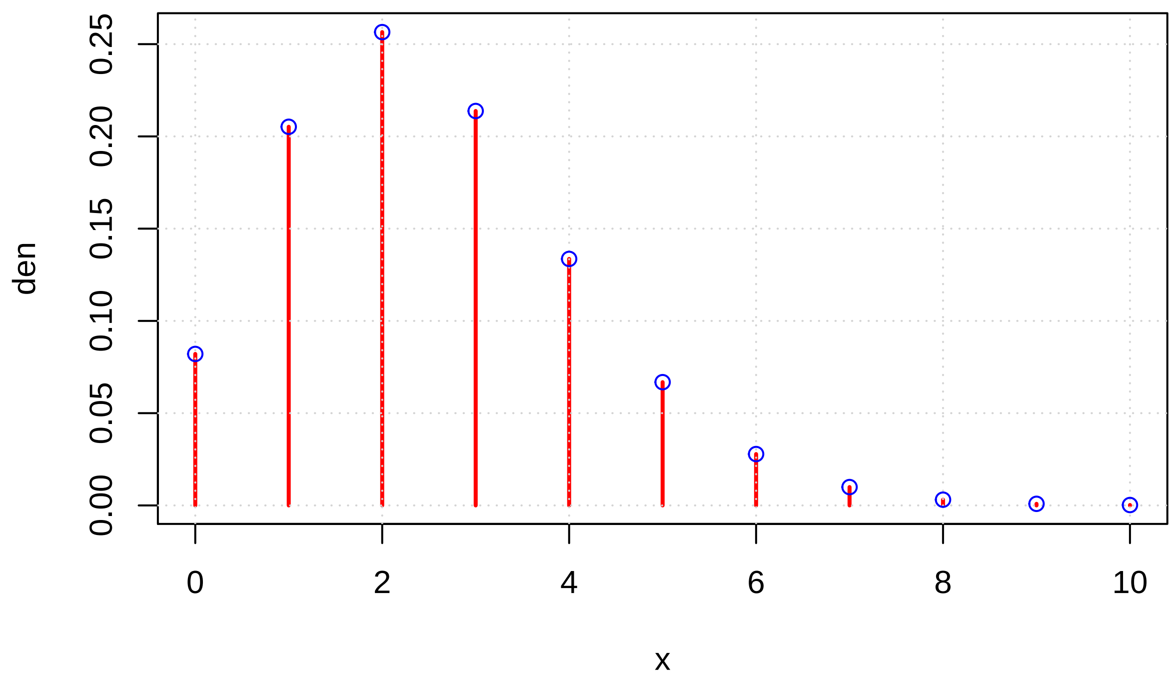

Histogram of m
Density
0.00
0.02
0.04
0.06
0.08
0.10
0
5
10
15
20
25
30
m

Ejercicio 4. Lab 3

Setup

Lo primero que realizaremos sera un almacenamiento de todas las variables requeridas por la distribución y a su vez las mostraremos gráficamente

```
lambda <- 2.5;
x <- seq(0, 10, 1)
den <- dpois(x, lambda)

plot(x, den, type="h", lwd=2, col="red")
grid()
points(x, den, col="blue")
```



Calcular la probabilidad de que sea mayor o igual que 4.5

Para ello usaremos la función ppois que nos calculara la acumulada menor a 4.5 y le restaremos 1 para obtener la inversa. Al ser la distribución discreta como podemos observar en prob y prob1 el valor de que sea mayor o igual a 4.5 y a 4 es el mismo, pues nuestra función no puede tomar valores intermedios.

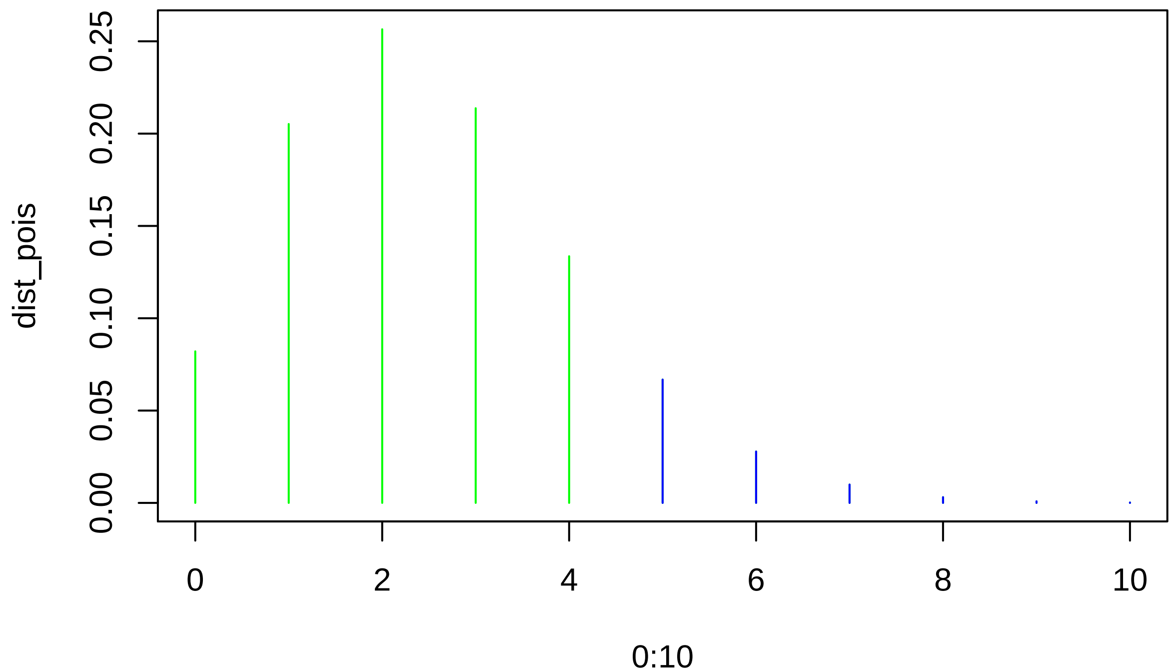
```
prob <- 1 - ppois(4, lambda); prob
```

```
## [1] 0.108822
```

```
prob1<- 1 - ppois(4.5, lambda); prob1
```

```
## [1] 0.108822
```

```
dist_pois <- dpois(0:10, 2.5)
plot(0:10, dist_pois, col="green", type="h")
points(5:10, dpois(5:10, 2.5), type="h", col="blue")
```



Calcular la probabilidad de sus valores mayores o iguales a 1 y menores o iguales a 5

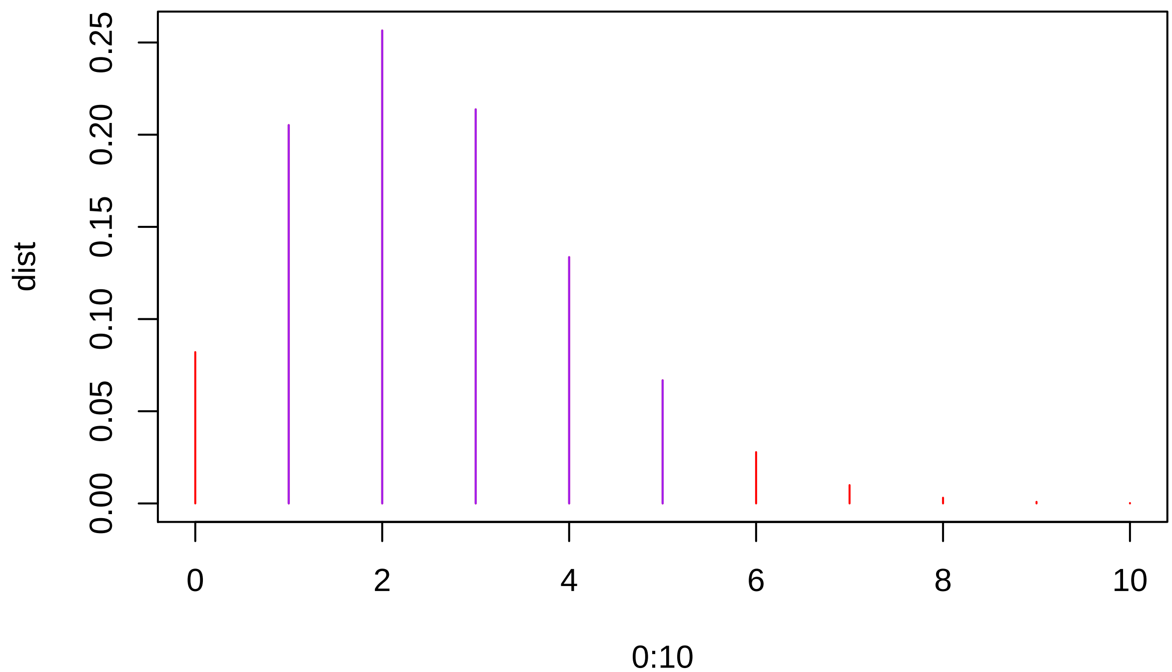
Para ello usaremos ppois para calcular la probabilidad acumulada de 0 a 5 y restarle la de 0 a 1 de esa manera encontramos la probabilidad que deseamos.

A su vez podemos observar en color lila la parte que corresponde a la probabilidad que buscamos. Y a su vez también aquella perteneciente a la frecuencia acumulada.

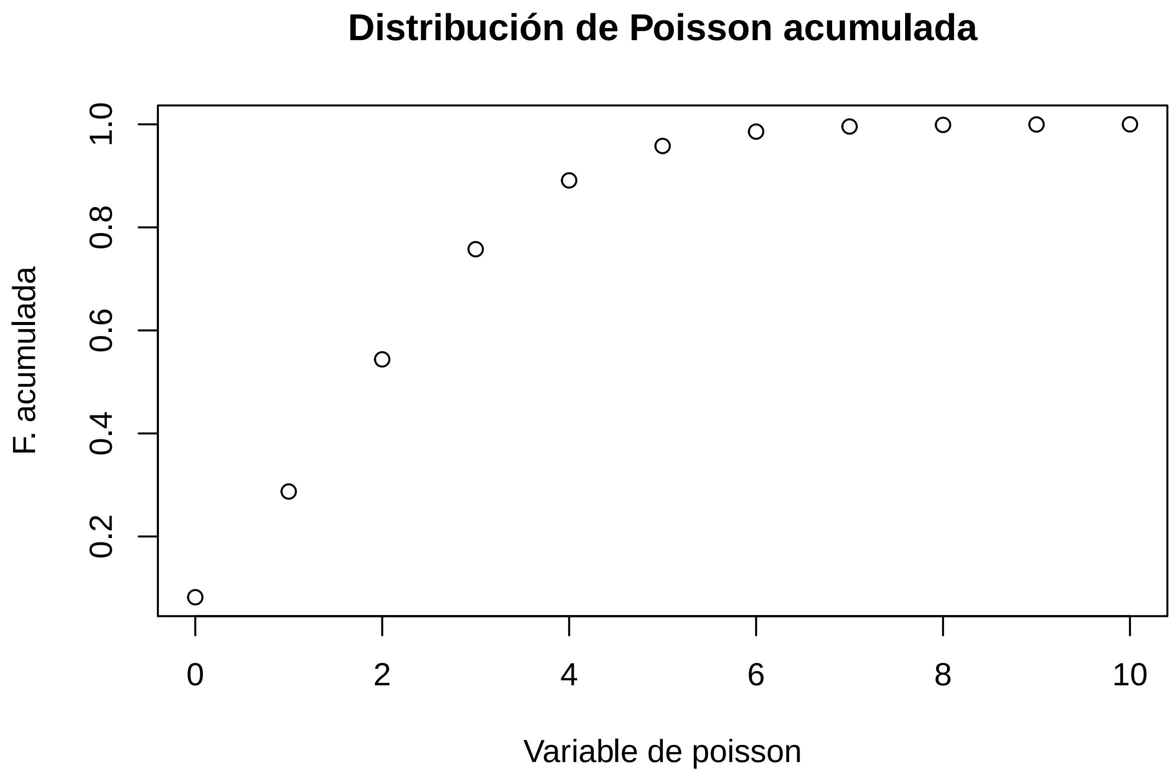
```
p5 <- ppois(5, lambda) - ppois(1, lambda); p5
```

```
## [1] 0.6706815
```

```
dist <- dpois(0:10, lambda)
plot(0:10, dist, col="red", type="h")
points(1:5, dpois(1:5, lambda), type="h", col="purple")
```




```
x <- 0:10
dist1 <- ppois(x, lambda)
plot(x, dist1, xlab="Variable de poisson", ylab="F. acumulada", main = "Distribución de Poisson acumula
```



Objeter el percentil 75 de la distribución

Esto nos será sencillo usando ppois que nos permitira calcular directamente ese percentil.

```
per <- ppois(0.75, lambda); per
```

```
## [1] 0.082085
```

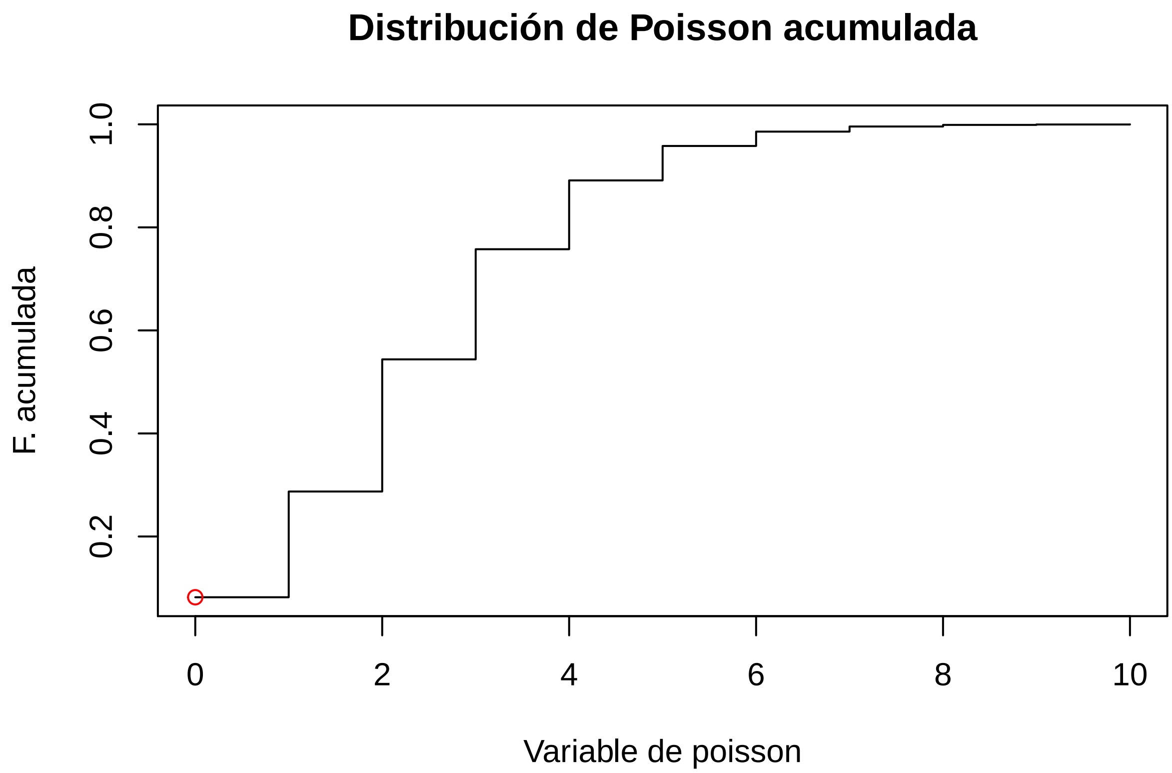
¿Qué valor es el que deja por debajo de si el 5% de los valores mas bajos de la variable?

Para ello usaremos qpois de 0.05 que nos dará el cuantil 0.05.

```
p6 <- qpois(0.05, lambda); p6
```

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

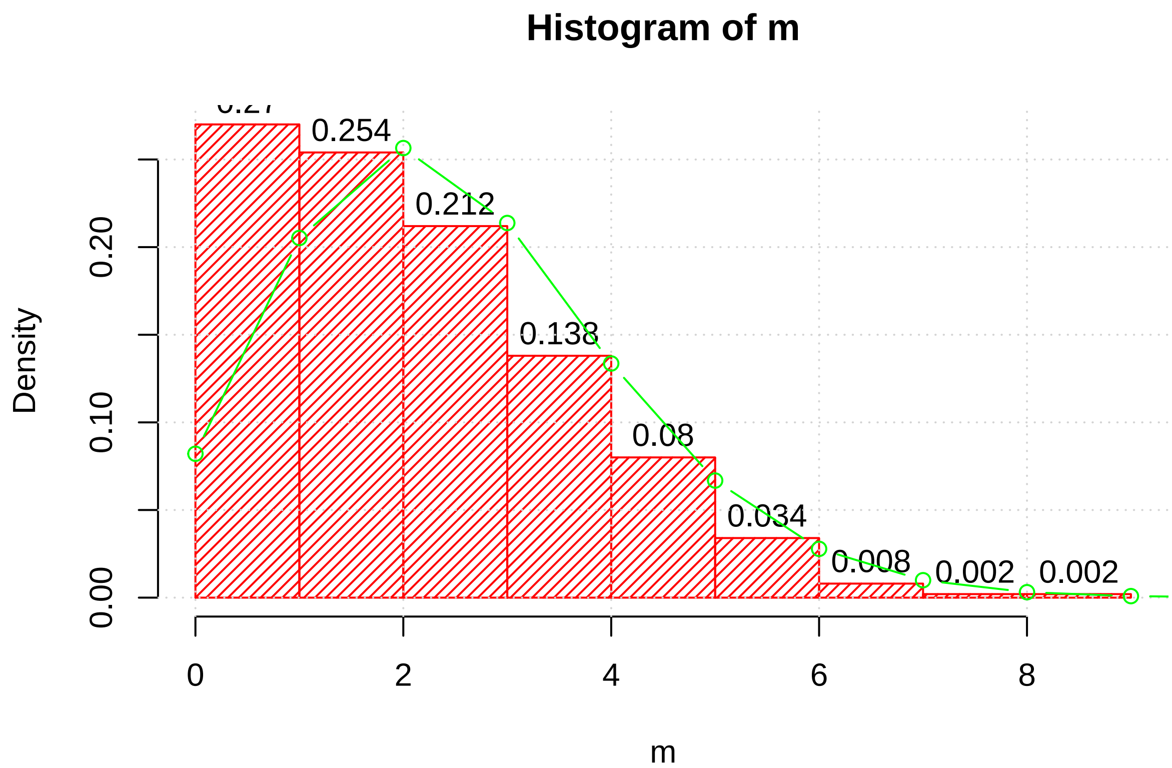
```
plot(x, dist1, xlab="Variable de poisson", ylab="F. acumulada", main = "Distribución de Poisson acumulad
points(p6, ppois(p6, lambda), col="red")
```



Obtener una muestra de tamaño 500 de la distribución, representarla gráficamente mediante un diagrama de barras y comparar éste con las frecuencias esperadas según el modelo que genera los datos.

Como podemos observar en general se ajusta a la distribución que hemos generado.

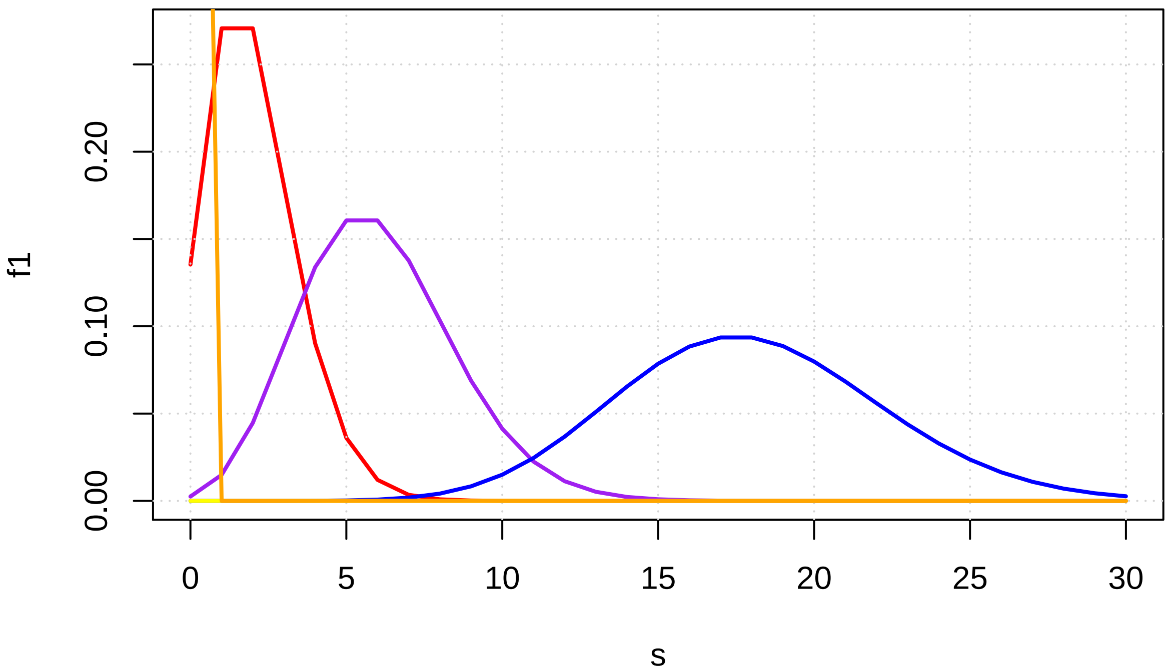
```
set.seed(1200)
size <- 500
m <- rpois(size, lambda)


hist(m, col="red", density=25, freq= F, labels= T)
grid()
points(x, den, type="b", col="green")
```



Explicar la influencia del parámetro lambda en la distribución y visualizar los diferentes resultados superpuestos.

Como podemos observar en las diferentes gráficas mostradas conforme el valor de lambda aumenta la altura disminuye y a su vez su dispersión es mayor y el valor medio de la distribución se acerca mas al 0 en el eje x. Esto sucede debido a que la distribución de poisson es una de las únicas donde el parametro lambda coincide con la media y la varianza.

```r
s <- seq(0, 30)
lambda <- 2
f1 <- dpois(s, lambda)
f2 <- dpois(s, 3*lambda)
f3 <- dpois(s, 9* lambda)
f4 <- dpois(s, 100* lambda)
f5 <- dpois(s, 0* lambda)

plot(s, f1, col="red", lwd = 2, type="l", ylim=c(0, max(f1)))
grid()
points(s, f2, col="purple", lwd = 2, type="l", ylim=c(0, max(f1)))
points(s, f3, col="blue", lwd = 2, type="l")
points(s, f4, col="yellow", lwd = 2, type="l")
points(s, f5, col="orange", lwd = 2, type="l")
```


f1
0.00
0.10
0.20
0
5
10
15
20
25
30
s

Ejercicio 5. Lab 3

Setup

Primero mostraremos gráficamente la distribución marcando la media con una linea horizontal. Definiendo tanto los limites como la sigma y media.

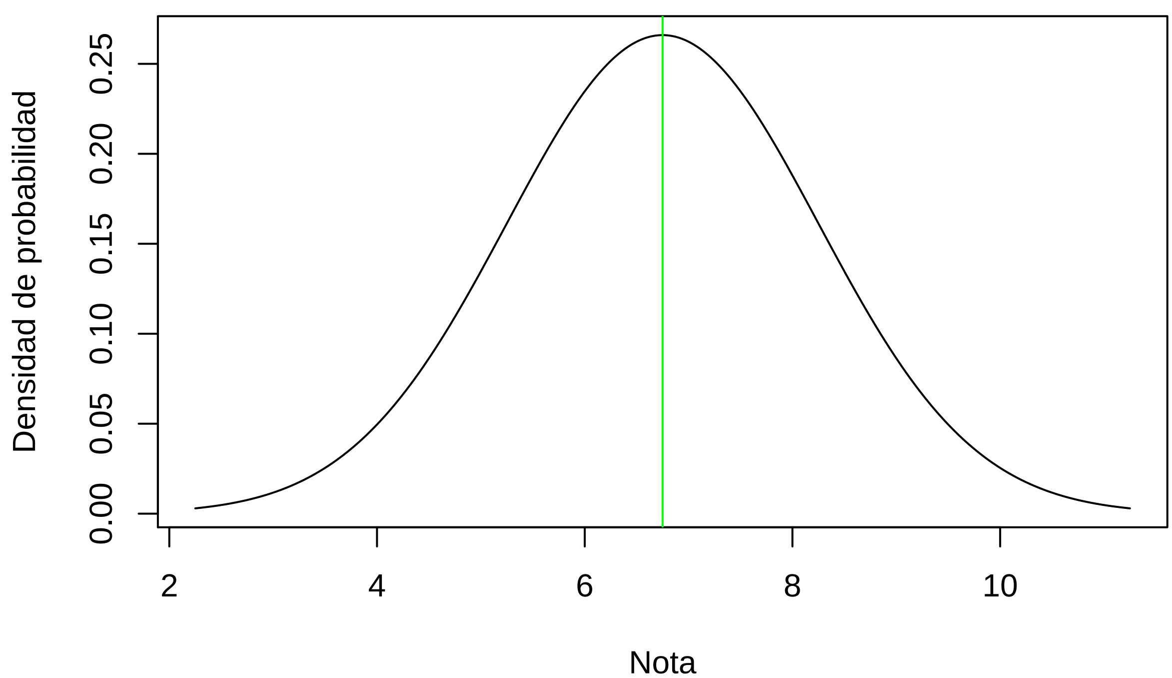
```
mu <- 6.75
sigma <- 1.5
z_max <- mu+3*sigma; z_max
```

```
## [1] 11.25
```

```
z_min <- mu-3*sigma; z_min
```

```
## [1] 2.25
```

```
x <- seq(z_min, z_max, 0.01)
plot(x, dnorm(x, mu, sigma), type= "l", xaxt="l", ylab="Densidad de probabilidad", xlab="Nota")
abline(v=mu, col="green")
```



P[5 < Nota <= 8]

Esta probabilidad es igual a la probabilidad acumulada hasta 8 menos la acumulada hasta 5. Como podemos observar en las distribuciones de probabilidad que no son discretas la probabilidad en un punto es cero. Esto es debido a que la probabilidad en este tipo de funciones es igual a el area comprendida entre dos puntos (la integral) como se ve en la gráfica.

```
p1 <- pnorm(8, mu, sigma)-pnorm(5,mu, sigma); p1
```

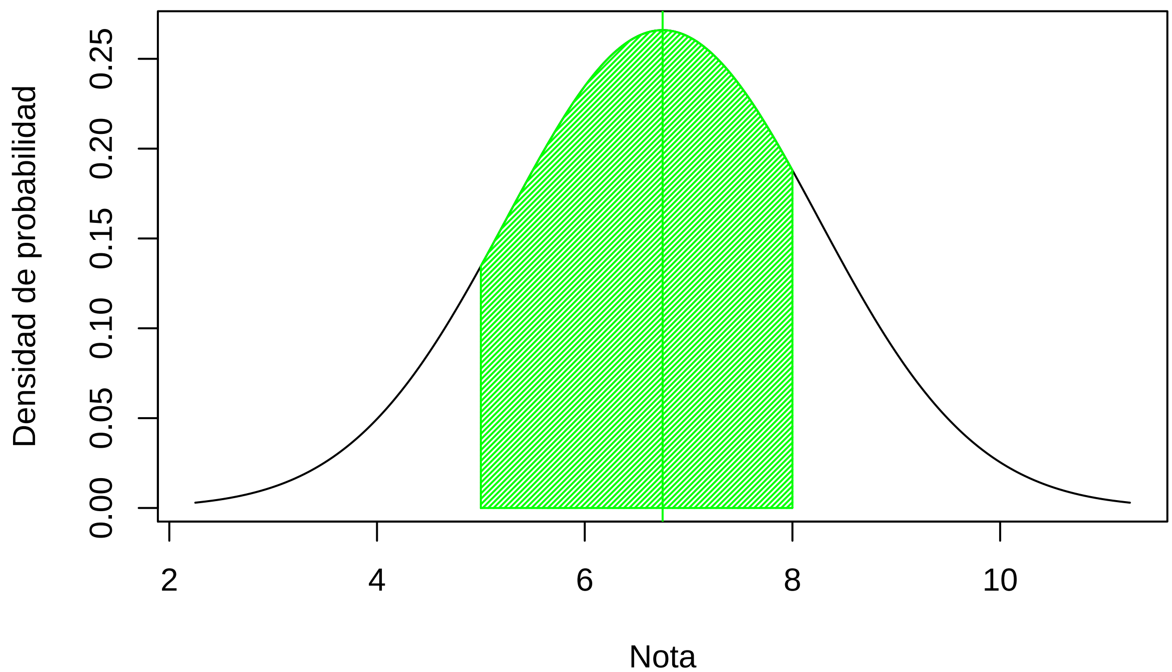
```
## [1] 0.6759991
```

```
z <- seq(5,8, 0.01)
p <- dnorm(z,mu,sigma); p
```

```
##   [1] 0.1346658 0.1357143 0.1367648 0.1378174 0.1388719 0.1399282 0.1409863
##   [8] 0.1420461 0.1431075 0.1441704 0.1452348 0.1463005 0.1473675 0.1484357
##  [15] 0.1495049 0.1505752 0.1516464 0.1527185 0.1537912 0.1548647 0.1559387
##  [22] 0.1570131 0.1580880 0.1591631 0.1602384 0.1613138 0.1623892 0.1634645
##  [29] 0.1645397 0.1656145 0.1666889 0.1677629 0.1688363 0.1699090 0.1709809
##  [36] 0.1720519 0.1731219 0.1741909 0.1752586 0.1763251 0.1773902 0.1784537
##  [43] 0.1795157 0.1805760 0.1816344 0.1826910 0.1837455 0.1847979 0.1858481
##  [50] 0.1868959 0.1879413 0.1889841 0.1900242 0.1910616 0.1920962 0.1931277
##  [57] 0.1941562 0.1951814 0.1962034 0.1972219 0.1982369 0.1992483 0.2002559
##  [64] 0.2012597 0.2022595 0.2032553 0.2042469 0.2052342 0.2062171 0.2071955
##  [71] 0.2081693 0.2091384 0.2101026 0.2110619 0.2120162 0.2129653 0.2139092
##  [78] 0.2148477 0.2157807 0.2167082 0.2176299 0.2185459 0.2194560 0.2203600
##  [85] 0.2212580 0.2221497 0.2230352 0.2239142 0.2247866 0.2256525 0.2265116
##  [92] 0.2273639 0.2282092 0.2290475 0.2298787 0.2307026 0.2315192 0.2323283
##  [99] 0.2331300 0.2339240 0.2347102 0.2354887 0.2362592 0.2370217 0.2377761
## [106] 0.2385223 0.2392602 0.2399897 0.2407108 0.2414233 0.2421271 0.2428222
## [113] 0.2435084 0.2441858 0.2448542 0.2455134 0.2461635 0.2468044 0.2474359
## [120] 0.2480580 0.2486707 0.2492737 0.2498672 0.2504509 0.2510248 0.2515888
## [127] 0.2521429 0.2526870 0.2532210 0.2537449 0.2542585 0.2547619 0.2552549
## [134] 0.2557375 0.2562097 0.2566712 0.2571222 0.2575626 0.2579922 0.2584111
## [141] 0.2588191 0.2592163 0.2596025 0.2599778 0.2603420 0.2606951 0.2610372
## [148] 0.2613680 0.2616877 0.2619960 0.2622931 0.2625789 0.2628533 0.2631163
## [155] 0.2633678 0.2636079 0.2638365 0.2640535 0.2642589 0.2644528 0.2646350
## [162] 0.2648056 0.2649646 0.2651118 0.2652473 0.2653712 0.2654832 0.2655835
## [169] 0.2656721 0.2657488 0.2658138 0.2658670 0.2659083 0.2659379 0.2659556
## [176] 0.2659615 0.2659556 0.2659379 0.2659083 0.2658670 0.2658138 0.2657488
## [183] 0.2656721 0.2655835 0.2654832 0.2653712 0.2652473 0.2651118 0.2649646
## [190] 0.2648056 0.2646350 0.2644528 0.2642589 0.2640535 0.2638365 0.2636079
## [197] 0.2633678 0.2631163 0.2628533 0.2625789 0.2622931 0.2619960 0.2616877
## [204] 0.2613680 0.2610372 0.2606951 0.2603420 0.2599778 0.2596025 0.2592163
## [211] 0.2588191 0.2584111 0.2579922 0.2575626 0.2571222 0.2566712 0.2562097
## [218] 0.2557375 0.2552549 0.2547619 0.2542585 0.2537449 0.2532210 0.2526870
## [225] 0.2521429 0.2515888 0.2510248 0.2504509 0.2498672 0.2492737 0.2486707
## [232] 0.2480580 0.2474359 0.2468044 0.2461635 0.2455134 0.2448542 0.2441858
## [239] 0.2435084 0.2428222 0.2421271 0.2414233 0.2407108 0.2399897 0.2392602
## [246] 0.2385223 0.2377761 0.2370217 0.2362592 0.2354887 0.2347102 0.2339240
## [253] 0.2331300 0.2323283 0.2315192 0.2307026 0.2298787 0.2290475 0.2282092
## [260] 0.2273639 0.2265116 0.2256525 0.2247866 0.2239142 0.2230352 0.2221497
## [267] 0.2212580 0.2203600 0.2194560 0.2185459 0.2176299 0.2167082 0.2157807
## [274] 0.2148477 0.2139092 0.2129653 0.2120162 0.2110619 0.2101026 0.2091384
## [281] 0.2081693 0.2071955 0.2062171 0.2052342 0.2042469 0.2032553 0.2022595
```

```
## [288] 0.2012597 0.2002559 0.1992483 0.1982369 0.1972219 0.1962034 0.1951814
## [295] 0.1941562 0.1931277 0.1920962 0.1910616 0.1900242 0.1889841 0.1879413
```

```
z <- c(5,z,8)
p <- c(0,p,0)
plot(x, dnorm(x, mu, sigma), type= "l", xaxt="l", ylab="Densidad de probabilidad", xlab="Nota")
abline(v=mu, col="green")
polygon(z,p,col="green", density=50)
```



P [Nota >= 8]

La probabilidad de que sea mayor que 8 es la acumulada hasta 8 menos 1. La podemos calcular facilmente con pnorm. Por último la probabilidad la podemos observar en la gráfica con el area comprendida entre 8 y 10

```
p2 <- 1 - pnorm(8, mu, sigma); p2
```

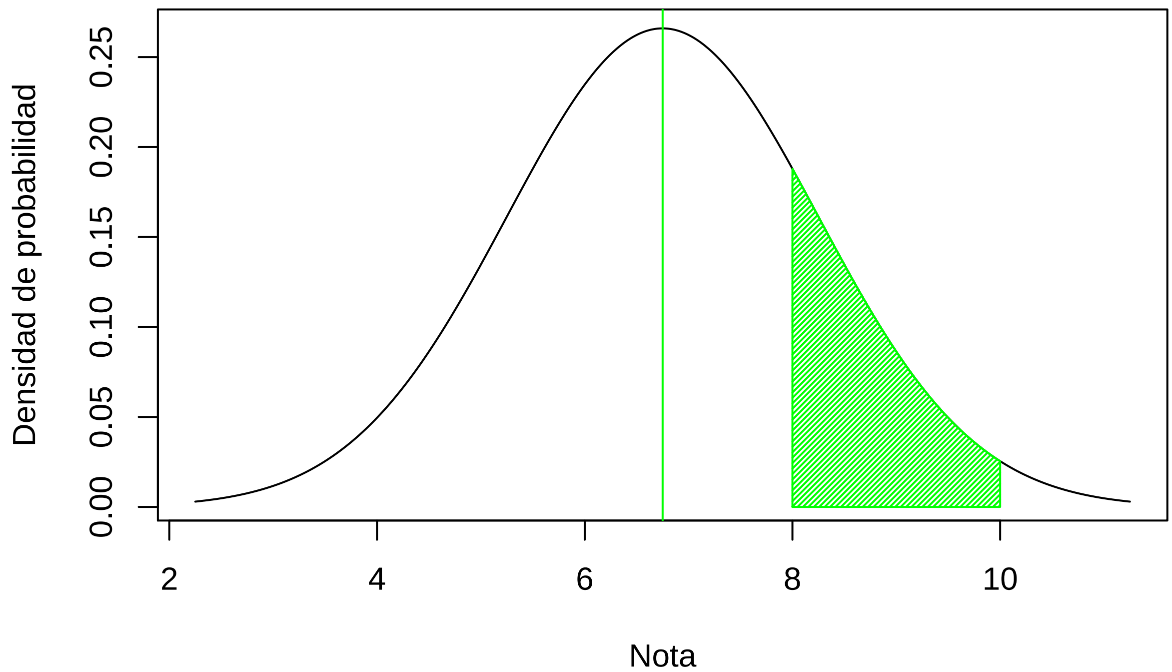
```
## [1] 0.2023284
```

```
z <- seq(8, 10, 0.01)
p <- dnorm(z,mu,sigma); p
```

```
##  [1] 0.18794125 0.18689587 0.18584805 0.18479789 0.18374550 0.18269098
##  [7] 0.18163444 0.18057598 0.17951571 0.17845374 0.17739017 0.17632509
## [13] 0.17525863 0.17419087 0.17312192 0.17205188 0.17098086 0.16990896
## [19] 0.16883627 0.16776289 0.16668894 0.16561449 0.16453966 0.16346454
## [25] 0.16238923 0.16131382 0.16023841 0.15916309 0.15808797 0.15701313
```

```
##  [31] 0.15593866 0.15486467 0.15379124 0.15271846 0.15164642 0.15057522
##  [37] 0.14950494 0.14843567 0.14736749 0.14630050 0.14523478 0.14417042
##  [43] 0.14310750 0.14204610 0.14098630 0.13992820 0.13887186 0.13781737
##  [49] 0.13676481 0.13571426 0.13466579 0.13361948 0.13257541 0.13153365
##  [55] 0.13049428 0.12945737 0.12842299 0.12739121 0.12636211 0.12533574
##  [61] 0.12431219 0.12329152 0.12227379 0.12125908 0.12024744 0.11923894
##  [67] 0.11823365 0.11723162 0.11623292 0.11523760 0.11424573 0.11325736
##  [73] 0.11227255 0.11129136 0.11031384 0.10934005 0.10837004 0.10740386
##  [79] 0.10644156 0.10548319 0.10452881 0.10357846 0.10263219 0.10169005
##  [85] 0.10075207 0.09981831 0.09888881 0.09796361 0.09704275 0.09612628
##  [91] 0.09521423 0.09430664 0.09340356 0.09250500 0.09161103 0.09072165
##  [97] 0.08983692 0.08895687 0.08808152 0.08721091 0.08634506 0.08548402
## [103] 0.08462779 0.08377642 0.08292993 0.08208835 0.08125169 0.08041999
## [109] 0.07959326 0.07877153 0.07795482 0.07714315 0.07633653 0.07553500
## [115] 0.07473855 0.07394722 0.07316102 0.07237996 0.07160405 0.07083331
## [121] 0.07006776 0.06930740 0.06855224 0.06780230 0.06705758 0.06631809
## [127] 0.06558385 0.06485485 0.06413110 0.06341261 0.06269938 0.06199143
## [133] 0.06128874 0.06059132 0.05989918 0.05921231 0.05853071 0.05785439
## [139] 0.05718335 0.05651757 0.05585707 0.05520183 0.05455185 0.05390713
## [145] 0.05326766 0.05263344 0.05200446 0.05138071 0.05076218 0.05014888
## [151] 0.04954078 0.04893788 0.04834016 0.04774763 0.04716026 0.04657805
## [157] 0.04600098 0.04542904 0.04486222 0.04430051 0.04374388 0.04319232
## [163] 0.04264583 0.04210437 0.04156795 0.04103653 0.04051011 0.03998867
## [169] 0.03947218 0.03896063 0.03845400 0.03795227 0.03745543 0.03696344
## [175] 0.03647630 0.03599398 0.03551645 0.03504371 0.03457572 0.03411246
## [181] 0.03365392 0.03320006 0.03275087 0.03230632 0.03186638 0.03143104
## [187] 0.03100028 0.03057405 0.03015235 0.02973514 0.02932240 0.02891410
## [193] 0.02851022 0.02811074 0.02771562 0.02732484 0.02693837 0.02655619
## [199] 0.02617826 0.02580457 0.02543508
```

```r
z <- c(8,z,10)
p <- c(0,p,0)
plot(x, dnorm(x, mu, sigma), type= "l", xaxt="l", ylab="Densidad de probabilidad", xlab="Nota")
abline(v=mu, col="green")
polygon(z,p,col="green", density=50)
```



Si queremos desechar el 5% de valores más altos de la distribución y el 5% de los valores más bajos, ¿Con qué intervalo de valores nos quedaríamos?

Para ello usaremos qnorm que nos dara el cuantil 0.05 y 0.95. El valor devolvido sera los puntos que tenemos que quedarnos, y el area entre estos puntos la probabilidad de estar entre ambas notas como se observa en la gráfica de abajo.

```
p <- qnorm(0.05, mu, sigma); p
```

```
## [1] 4.28272
```

```
p1 <- qnorm(0.95, mu, sigma); p1
```

```
## [1] 9.21728
```

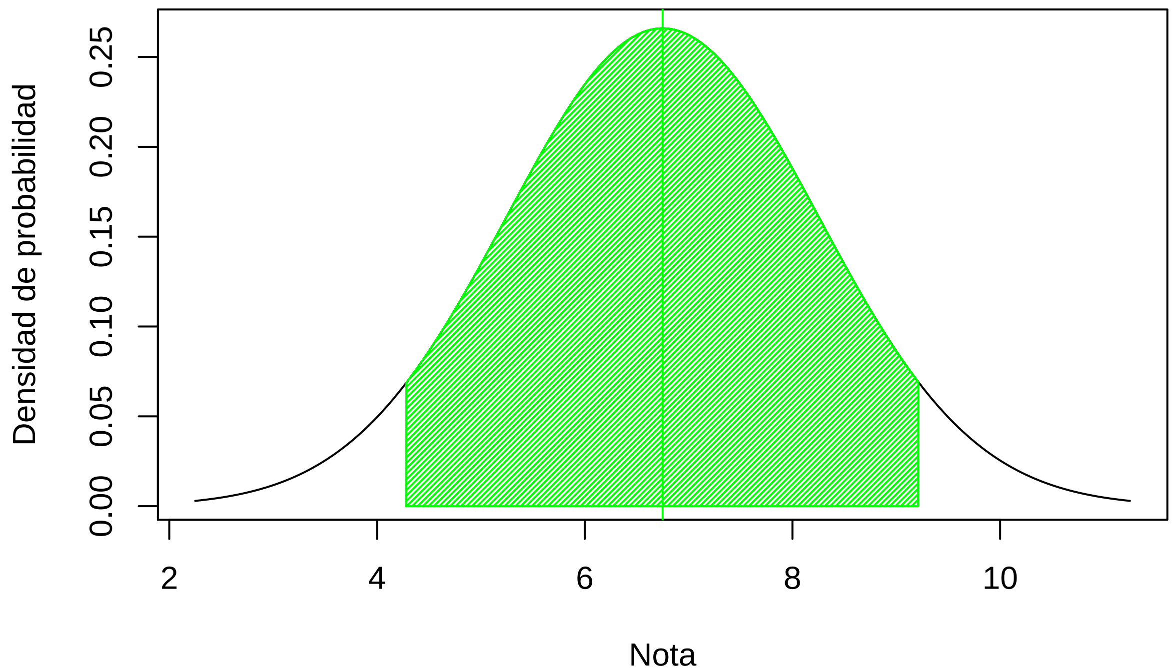
```
z <- seq(p, p1, 0.01)
p <- dnorm(z,mu,sigma); p
```

```
##   [1] 0.06875709 0.06951367 0.07027544 0.07104241 0.07181455 0.07259186
##   [7] 0.07337432 0.07416192 0.07495465 0.07575248 0.07655540 0.07736339
##  [13] 0.07817643 0.07899451 0.07981760 0.08064568 0.08147874 0.08231674
##  [19] 0.08315966 0.08400748 0.08486017 0.08571771 0.08658006 0.08744721
##  [25] 0.08831911 0.08919574 0.09007707 0.09096307 0.09185369 0.09274892
##  [31] 0.09364871 0.09455302 0.09546183 0.09637508 0.09729275 0.09821480
##  [37] 0.09914117 0.10007183 0.10100674 0.10194586 0.10288913 0.10383651
```

```
##  [43] 0.10478796 0.10574343 0.10670287 0.10766623 0.10863346 0.10960451
##  [49] 0.11057932 0.11155784 0.11254002 0.11352580 0.11451513 0.11550794
##  [55] 0.11650419 0.11750380 0.11850672 0.11951290 0.12052225 0.12153474
##  [61] 0.12255028 0.12356881 0.12459027 0.12561459 0.12664171 0.12767155
##  [67] 0.12870404 0.12973912 0.13077671 0.13181673 0.13285913 0.13390381
##  [73] 0.13495072 0.13599976 0.13705087 0.13810396 0.13915896 0.14021578
##  [79] 0.14127436 0.14233460 0.14339642 0.14445974 0.14552448 0.14659056
##  [85] 0.14765787 0.14872636 0.14979591 0.15086645 0.15193789 0.15301014
##  [91] 0.15408310 0.15515670 0.15623083 0.15730540 0.15838033 0.15945552
##  [97] 0.16053087 0.16160628 0.16268168 0.16375695 0.16483200 0.16590674
## [103] 0.16698106 0.16805487 0.16912806 0.17020055 0.17127223 0.17234299
## [109] 0.17341274 0.17448137 0.17554879 0.17661489 0.17767956 0.17874271
## [115] 0.17980423 0.18086402 0.18192196 0.18297797 0.18403192 0.18508371
## [121] 0.18613325 0.18718042 0.18822511 0.18926722 0.19030664 0.19134327
## [127] 0.19237699 0.19340771 0.19443530 0.19545966 0.19648069 0.19749828
## [133] 0.19851231 0.19952268 0.20052928 0.20153200 0.20253073 0.20352537
## [139] 0.20451580 0.20550192 0.20648362 0.20746078 0.20843331 0.20940108
## [145] 0.21036401 0.21132196 0.21227485 0.21322255 0.21416497 0.21510199
## [151] 0.21603351 0.21695942 0.21787962 0.21879399 0.21970244 0.22060485
## [157] 0.22150113 0.22239116 0.22327485 0.22415208 0.22502276 0.22588678
## [163] 0.22674405 0.22759445 0.22843788 0.22927425 0.23010346 0.23092540
## [169] 0.23173998 0.23254710 0.23334665 0.23413855 0.23492270 0.23569899
## [175] 0.23646734 0.23722766 0.23797984 0.23872379 0.23945943 0.24018665
## [181] 0.24090538 0.24161553 0.24231699 0.24300969 0.24369354 0.24436846
## [187] 0.24503435 0.24569114 0.24633874 0.24697707 0.24760604 0.24822559
## [193] 0.24883563 0.24943608 0.25002687 0.25060791 0.25117915 0.25174050
## [199] 0.25229189 0.25283325 0.25336451 0.25388560 0.25439646 0.25489701
## [205] 0.25538720 0.25586697 0.25633624 0.25679495 0.25724306 0.25768049
## [211] 0.25810720 0.25852312 0.25892820 0.25932240 0.25970565 0.26007790
## [217] 0.26043912 0.26078925 0.26112824 0.26145605 0.26177264 0.26207796
## [223] 0.26237198 0.26265466 0.26292595 0.26318583 0.26343425 0.26367119
## [229] 0.26389662 0.26411050 0.26431280 0.26450350 0.26468258 0.26485001
## [235] 0.26500576 0.26514982 0.26528217 0.26540279 0.26551167 0.26560878
## [241] 0.26569412 0.26576767 0.26582943 0.26587939 0.26591754 0.26594387
## [247] 0.26595839 0.26596108 0.26595196 0.26593101 0.26589825 0.26585368
## [253] 0.26579730 0.26572913 0.26564916 0.26555742 0.26545391 0.26533864
## [259] 0.26521164 0.26507292 0.26492249 0.26476039 0.26458662 0.26440122
## [265] 0.26420420 0.26399560 0.26377544 0.26354374 0.26330055 0.26304590
## [271] 0.26277981 0.26250232 0.26221347 0.26191329 0.26160184 0.26127914
## [277] 0.26094524 0.26060019 0.26024402 0.25987680 0.25949855 0.25910934
## [283] 0.25870922 0.25829823 0.25787644 0.25744389 0.25700065 0.25654676
## [289] 0.25608230 0.25560732 0.25512187 0.25462604 0.25411987 0.25360344
## [295] 0.25307680 0.25254004 0.25199322 0.25143640 0.25086967 0.25029309
## [301] 0.24970673 0.24911068 0.24850501 0.24788979 0.24726510 0.24663103
## [307] 0.24598764 0.24533504 0.24467329 0.24400248 0.24332269 0.24263402
## [313] 0.24193654 0.24123034 0.24051552 0.23979215 0.23906034 0.23832017
## [319] 0.23757173 0.23681512 0.23605042 0.23527774 0.23449716 0.23370879
## [325] 0.23291271 0.23210903 0.23129784 0.23047925 0.22965334 0.22882022
## [331] 0.22798000 0.22713276 0.22627861 0.22541766 0.22455001 0.22367575
## [337] 0.22279499 0.22190784 0.22101440 0.22011477 0.21920907 0.21829738
## [343] 0.21737983 0.21645651 0.21552753 0.21459301 0.21365304 0.21270773
## [349] 0.21175720 0.21080154 0.20984087 0.20887529 0.20790492 0.20692986
## [355] 0.20595022 0.20496610 0.20397762 0.20298489 0.20198801 0.20098710
## [361] 0.19998225 0.19897359 0.19796121 0.19694523 0.19592576 0.19490291
```

```
## [367] 0.19387677 0.19284747 0.19181511 0.19077979 0.18974163 0.18870073
## [373] 0.18765720 0.18661115 0.18556268 0.18451190 0.18345892 0.18240384
## [379] 0.18134677 0.18028781 0.17922707 0.17816465 0.17710066 0.17603520
## [385] 0.17496837 0.17390027 0.17283102 0.17176071 0.17068944 0.16961731
## [391] 0.16854442 0.16747088 0.16639678 0.16532222 0.16424730 0.16317212
## [397] 0.16209677 0.16102135 0.15994596 0.15887068 0.15779563 0.15672088
## [403] 0.15564653 0.15457268 0.15349942 0.15242683 0.15135502 0.15028405
## [409] 0.14921404 0.14814506 0.14707720 0.14601055 0.14494519 0.14388121
## [415] 0.14281869 0.14175772 0.14069837 0.13964074 0.13858490 0.13753093
## [421] 0.13647891 0.13542891 0.13438102 0.13333532 0.13229187 0.13125075
## [427] 0.13021204 0.12917581 0.12814213 0.12711107 0.12608271 0.12505710
## [433] 0.12403433 0.12301445 0.12199754 0.12098365 0.11997286 0.11896523
## [439] 0.11796082 0.11695969 0.11596190 0.11496751 0.11397659 0.11298918
## [445] 0.11200535 0.11102515 0.11004864 0.10907587 0.10810690 0.10714177
## [451] 0.10618053 0.10522325 0.10426996 0.10332071 0.10237556 0.10143454
## [457] 0.10049771 0.09956510 0.09863677 0.09771275 0.09679308 0.09587780
## [463] 0.09496696 0.09406060 0.09315874 0.09226143 0.09136870 0.09048059
## [469] 0.08959712 0.08871835 0.08784428 0.08697496 0.08611042 0.08525068
## [475] 0.08439578 0.08354573 0.08270057 0.08186033 0.08102501 0.08019466
## [481] 0.07936929 0.07854892 0.07773358 0.07692328 0.07611805 0.07531789
## [487] 0.07452284 0.07373290 0.07294809 0.07216843 0.07139393 0.07062460
## [493] 0.06986046 0.06910151
```

```r
z <- c(4.28,z,9.21)
p <- c(0,p,0)
plot(x, dnorm(x, mu, sigma), type= "l", xaxt="l", ylab="Densidad de probabilidad", xlab="Nota")
abline(v=mu, col="green")
polygon(z,p,col="green", density=50)
```

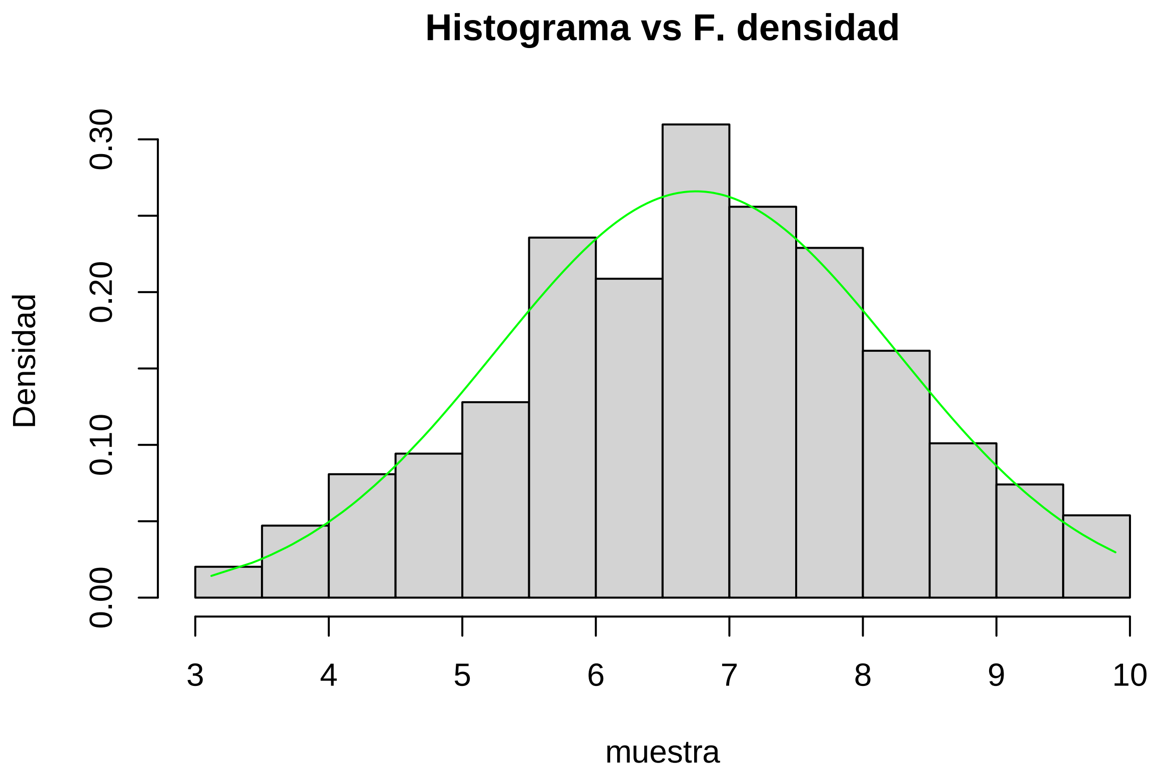


Obtener una muestra de tamaño 300 de la distribución, representar la función de densidad de esta distribución y compararla con el histograma de la muestra obtenida (limitar las calificaciones al intervalo 0, 10)

Para ello como siempre definiremos una semilla de generación, crearemos una distribución aleatoria con rnorm y limitaremos esta entre 0 y 10. Por ultimo mostraremos un histograma y encima de ella los puntos de la d. de probabilidad.

Podemos observar que en su mayoría conincide correctamente el histograma y la función.

```
set.seed(2000)
x <- rnorm(300, 6.75, 1.5)
x <- x[(0 <= x)&(x<=10)]
muestra <- sort(x)
i <- hist(muestra, freq=FALSE, main="Histograma vs F. densidad", xlab="muestra", ylab="Densidad")
lines(muestra, dnorm(muestra, mu , sigma), col="green")
```



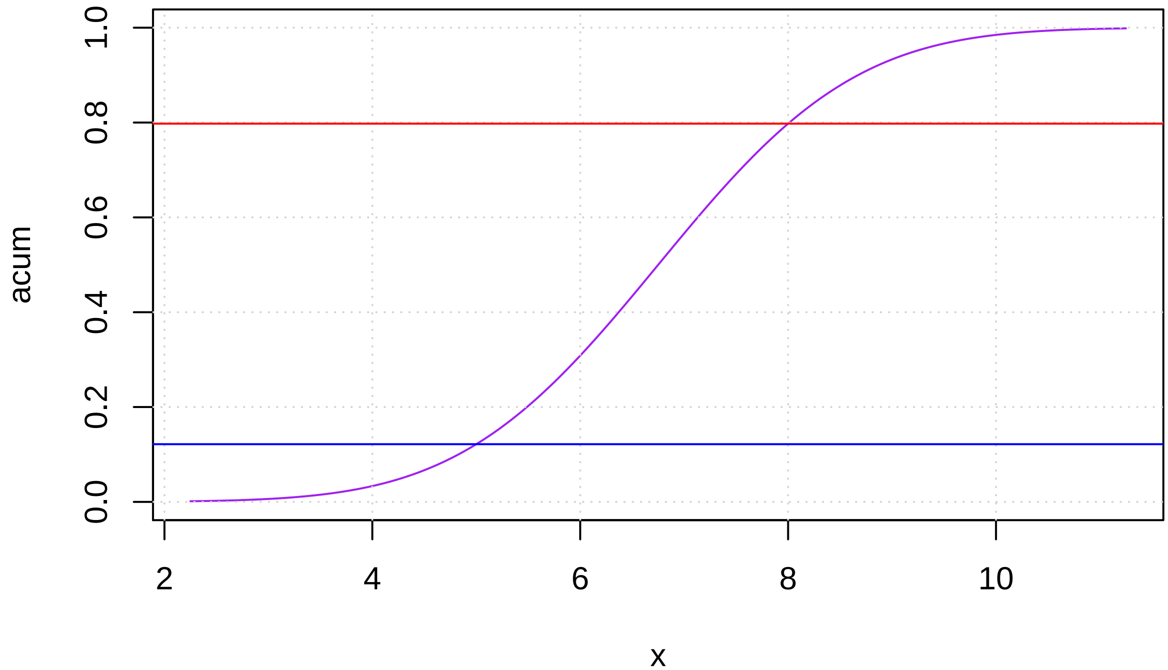
Obtener y visualizar la función de distribución acumulada y situar sobre ella los resultados de a) y b)

Para ello primero visualizaremos la función de la distribución acumulada y dibujaremos lineas paralelas al eje x en el pnorm 5 y 8. La distancia entre ambas lineas vendría a representar la probabilidad entre 5 y 8 la parte inferior la probabilidad de ser menor que cinco y la superior la mayor. Esto es gracias a las propiedades de la función de distribución acumulada.

```
x <- seq(z_min, z_max, 0.01)
acum <- pnorm(x, mu, sigma)
plot(x, acum, type="line", col = "purple")
```

```
## Warning in plot.xy(xy, type, ...): gráfico de tipo 'line' va a ser truncado al
## primer carácter
```

```
grid()
abline(h=pnorm(5,mu, sigma), col = "blue")
abline(h=pnorm(8,mu, sigma), col="red")
```



Calcular los coeficientes que definen los factores de forma de la distribución (Curtosis, y Asimetría)

Por último para calcular la curtosis y asimetría nos es tan sencillo como importar la libreria e1071 y usar las funciones correspondientes. Como bien nos indica la curtosis tenemos una función platicúrtica, esto quiere decir que tenemos muy poca concentración de datos en la media (esta es mas achatada que la d. norm).

Y la Asimetría nos indica que es asimétrica negativa y la media esta desplazada hacia la derecha.

```
library(e1071)
kurtosis(muestra, type=1)
```

```
## [1] -0.3690067
```

```
skewness(muestra)
```

```
## [1] -0.1051452
```