

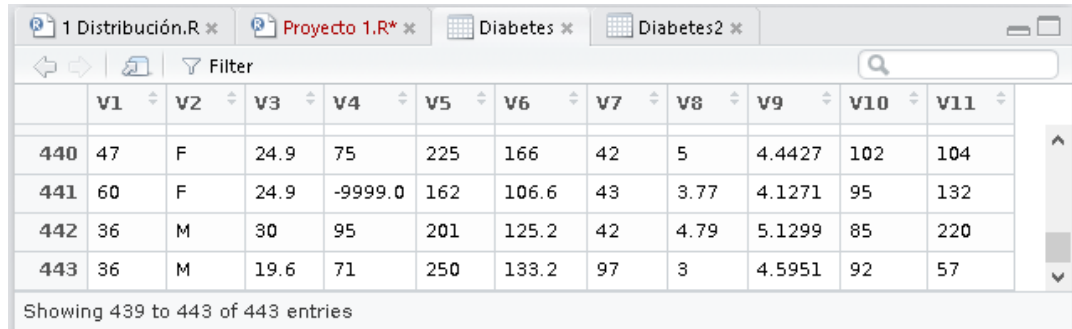
Proyecto fin R

Con el conjunto de datos diabetes.data

1 Cargar los datos en R.

```
Diabetes <- read.table ("diabetes.data", header=TRUE)
```

(NO OLVIDAR indicar que tiene cabecera el fichero con header=TRUE que si no te importa todo como texto y después no te sale nada.

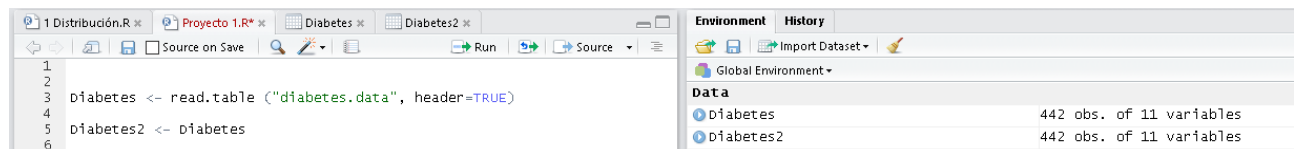


	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11
440	47	F	24.9	75	225	166	42	5	4.4427	102	104
441	60	F	24.9	-9999.0	162	106.6	43	3.77	4.1271	95	132
442	36	M	30	95	201	125.2	42	4.79	5.1299	85	220
443	36	M	19.6	71	250	133.2	97	3	4.5951	92	57

Showing 439 to 443 of 443 entries

Voy a crear un data frame identico, para tener como backup/comparar el Dataframe Diabetes

```
Diabetes2 <- Diabetes
```



```
1
2
3 Diabetes <- read.table ("diabetes.data", header=TRUE)
4
5 Diabetes2 <- Diabetes
6
```

Global Environment	
Diabetes	442 obs. of 11 variables
Diabetes2	442 obs. of 11 variables

1 Eliminar los missing values, que estan codicados como -9999.00.

```
Diabetes2[Diabetes2==-9999.0]<-NA
Diabetes2
```

Diabetes2 En diabetes 2

```
[Diabetes2==-9999.0] Donde sea el valor -9999.0
```

```
<-NA Ahora sera NA
```

```
438 60 F 28.2 112.00 185 113.8 42.0 4.00 4.9836 93 178
439 47 F 24.9 75.00 225 166.0 42.0 5.00 4.4427 102 104
440 60 F 24.9 NA 162 106.6 43.0 3.77 4.1271 95 132
441 36 M 30.0 95.00 201 125.2 42.0 4.79 5.1299 85 220
442 36 M 19.6 71.00 250 133.2 97.0 3.00 4.5951 92 57
> |
```

Y ahora las eliminamos

```
Diabetes2 <- na.omit(Diabetes2)
```

```
Diabetes2
```

```
438 60 F 28.2 112.00 185 113.8 42.0 4.00 4.9836 93 178
439 47 F 24.9 75.00 225 166.0 42.0 5.00 4.4427 102 104
441 36 M 30.0 95.00 201 125.2 42.0 4.79 5.1299 85 220
442 36 M 19.6 71.00 250 133.2 97.0 3.00 4.5951 92 57
> |
```

¡ Ver el tipo de cada una de las variables.

`sapply(Diabetes2, class)`

```
> sapply(Diabetes2, class)
      AGE      SEX      BMI      BP      S1      S2      S3
"integer" "factor" "numeric" "numeric" "integer" "numeric" "numeric"
      S4      S5      S6      Y
"numeric" "numeric" "integer" "integer"
> |
```

`sapply` Aplicame una función a
(`Diabetes2`, A mi data frame de diabetes 2
`class`) la función `class`

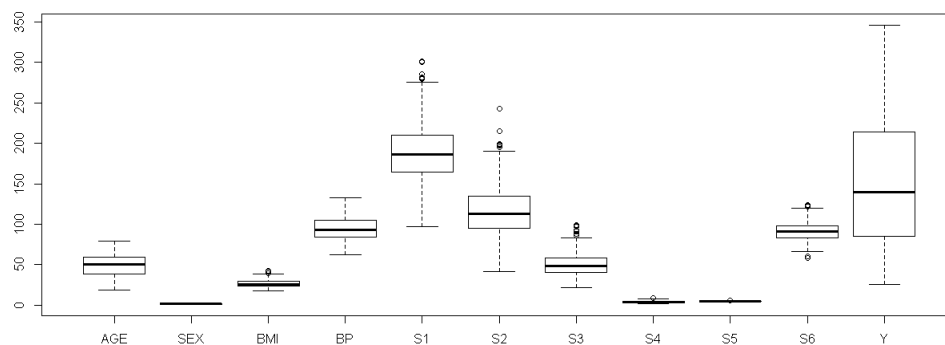
¡ Realizar un análisis estadístico de las variables: calcular la media, varianza, rangos, etc. > Tienen las distintas variables rangos muy diferentes?.

`summary(Diabetes2)`

```
> summary(Diabetes2)
      AGE      SEX      BMI      BP      S1      S2      S3      S4
Min.   :19.00  F:203  Min.   :18.00  Min.   : 62.00
1st Qu.:38.00  M:230  1st Qu.:23.10  1st Qu.: 84.00
Median :50.00          Median :25.70  Median : 93.00
Mean   :48.48          Mean  :26.35  Mean   : 94.65
3rd Qu.:59.00          3rd Qu.:29.20  3rd Qu.:105.00
Max.   :79.00          Max.   :42.20  Max.   :133.00
      S5      S6      Y
Min.   : 3.258  Min.   : 58.00  Min.   : 25.0
1st Qu.:4.277  1st Qu.: 83.00  1st Qu.: 85.0
Median :4.635  Median : 91.00  Median :140.0
Mean   :4.645  Mean   : 91.25  Mean   :152.2
3rd Qu.:4.997  3rd Qu.: 98.00  3rd Qu.:214.0
Max.   :6.107  Max.   :124.00  Max.   :346.0
> |
```

¡ Hacer un gráfico de cajas (boxplot) donde se pueda ver la información anterior de forma gráfica.

`boxplot(Diabetes2)`



¡ Calcular la media para las las que tienen SEX=M y la media para las las que tienen SEX=F, utilizando la funcion tapply.

```
test <- tapply(Diabetes2$AGE,Diabetes2$SEX == "M", mean)
test1 <- tapply(Diabetes2$BMI,Diabetes2$SEX == "M", mean)
test2 <- tapply(Diabetes2$BP,Diabetes2$SEX == "M", mean)
test3 <- tapply(Diabetes2$S1,Diabetes2$SEX == "M", mean)
test4 <- tapply(Diabetes2$S2,Diabetes2$SEX == "M", mean)
test5 <- tapply(Diabetes2$S3,Diabetes2$SEX == "M", mean)
test6 <- tapply(Diabetes2$S4,Diabetes2$SEX == "M", mean)
test7 <- tapply(Diabetes2$S5,Diabetes2$SEX == "M", mean)
test8 <- tapply(Diabetes2$S6,Diabetes2$SEX == "M", mean)
test9 <- tapply(Diabetes2$Y,Diabetes2$SEX == "M", mean)
```

```
test1
test2
test3
test4
test5
test6
test7
test8
test9
```

```
> test1
  FALSE      TRUE
26.80099 25.95609
> test2
  FALSE      TRUE
98.17562 91.53474
> test3
  FALSE      TRUE
190.6552 188.0304
> test4
  FALSE      TRUE
120.1103 111.1726
> test5
  FALSE      TRUE
44.54187 54.56304
> test6
  FALSE      TRUE
4.537882 3.658217
> test7
  FALSE      TRUE
4.731533 4.569392
> test8
  FALSE      TRUE
93.86207 88.94348
~ |
```

test <- Lleva a Test el resultado de
tapply(Aplicar la funcion
Diabetes2\$AGE,Diabetes2\$SEX == "M", r
mean) Mean es decir calcular la media es la funcion a aplicar

```
> test
  FALSE      TRUE
50.86700 46.36522
```

Esto significa que con sexo M la media de edad es 50,86 y con sexo F la media de edad es: 46,36522

Calcular la correlacion de todas las variables numericas con la variable Y.

```
correlación <- cor(Diabetes2[,-2],Diabetes2$Y)
```

```
> correlación <- cor(Diabetes2[,-2],Diabetes2$Y)
> correlación
      [,1]
AGE  0.1889540
BMI  0.5863673
BP   0.4398515
S1   0.2133325
S2   0.1747189
S3  -0.3963076
S4   0.4325640
S5   0.5703164
S6   0.3892246
Y    1.0000000
```

```
correlación <- llevo a correlación (para poder ver el resultado después)
cor( el resultado de la función correlación
Diabetes2 aplicada a diabetes
[, -2] de todas las columnas pero quitando la 2 (la de Sexo porque es un texto)
,Diabetes2$Y) correlación respecto a la columna Y
```

Realizar un grafico de dispersion para las variables que tienen mas y menos correlacion con Y y comentar los resultados. > Como sera el grafico de dispersion entre dos cosas con correlacion 1?.

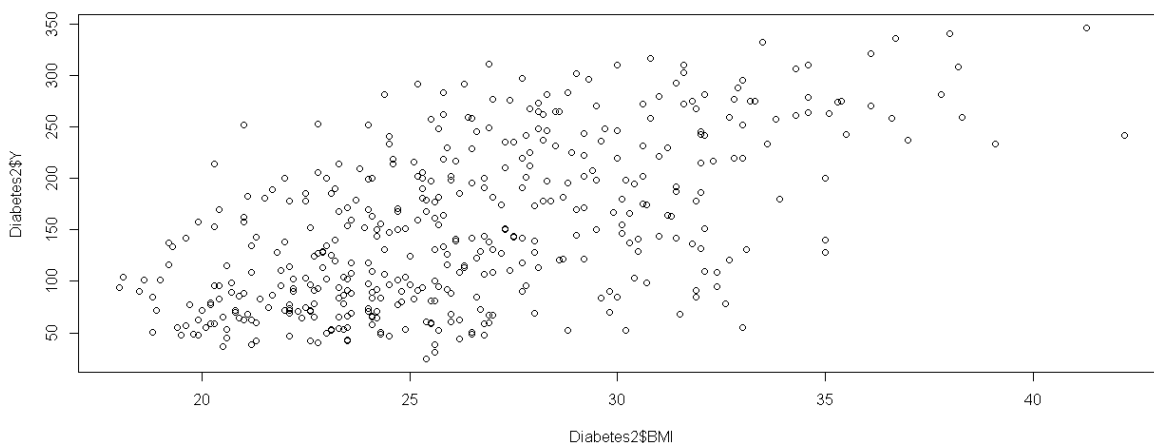
La relacion directa más amplia:

BMI 0.5863673

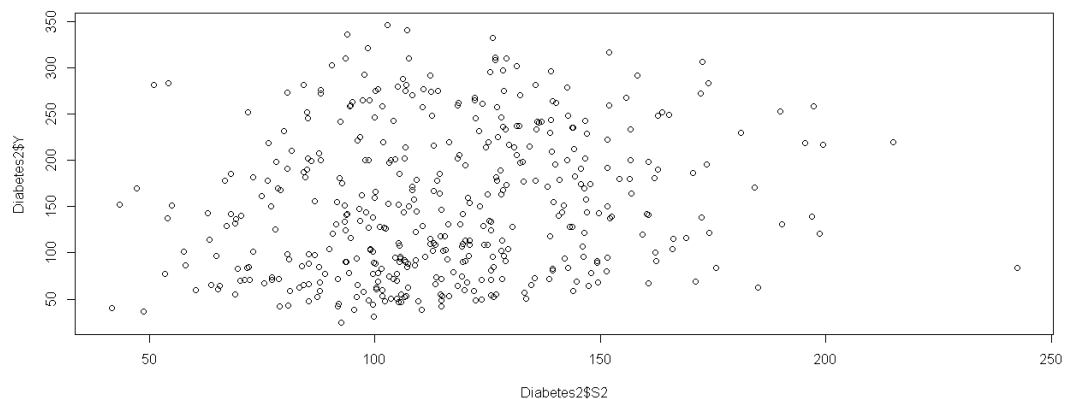
La relación inversa más cercana a 0:

S2 0.1747189

```
plot(Diabetes2$BMI, Diabetes2$Y)
```



```
plot(Diabetes2$S2, Diabetes2$Y)
```



Transformar la variable SEX, que es un factor, en una variable numerica utilizando, por ejemplo, la codicacion M=1 y F=2.

```
Diabetes2$SEX <- as.numeric(Diabetes2$SEX, "M" == 1, "F" == 2 )
```

```
Console C:/Asier/MASTER/Clases/R/Entregables/
442 30 M 19.0 71.00 230 133.2 97.0 3.00 4.3931 92 37
> Diabetes2$SEX <- as.numeric(Diabetes2$SEX, "M" == 1, "F" == 2 )
> Diabetes2
      AGE SEX  BMI      BP  S1      S2  S3  S4      S5  S6  Y
1    59  1 32.1 101.00 157  93.2 38.0 4.00 4.8598 87 151
2    48  2 21.6  87.00 183 103.2 70.0 3.00 3.8918 69  75
3    72  1 30.5  93.00 156  93.6 41.0 4.00 4.6728 85 141
4    24  2 25.3  84.00 198 131.4 40.0 5.00 4.8903 89 206
5    50  2 23.0 101.00 192 125.4 52.0 4.00 4.2905 80 135
```

```
Diabetes2$SEX <- Pon en diabetes2 2 en su columna SEX
as.numeric( Como numero
Diabetes2$SEX, Lo que hay en diabetes2 columna SEX
"M" == 1, "F" == 2 ) Si es M un 1 y si es F un 2
```

Definimos los outliers como los elementos (las) de los datos para los que cualquiera de las variables esta por encima o por debajo de la mediana mas/menos 3 veces el MAD (Median Absolute Deviation). Identificar estos outliers y quitarlos.

Buscamos ayuda para la mediana

```
87
88
89 ?median
90
91
92
93
97:1 (Top Level)
R Script
```

```
Console C:/Asier/MASTER/Clases/R/Entregables/
416 47 2 27.2 80.00 208 145.6 38.0 6.00 4.8040 92 174
417 41 2 33.8 123.33 187 127.0 45.0 4.16 4.3175 100 257
418 34 2 33.0 73.00 178 114.6 31.0 3.49 4.1271 92 55
419 51 2 24.1 87.00 261 175.6 69.0 4.00 4.4067 93 84
420 43 2 21.3 79.00 141 78.8 53.0 3.00 3.8286 90 42
422 59 1 27.9 101.00 218 144.2 38.0 6.00 5.1874 95 212
423 27 1 33.6 110.00 246 156.6 57.0 4.00 5.0876 89 233
424 51 1 22.7 103.00 217 162.4 30.0 7.00 4.8122 80 91
```

R: Median Value Find in Topic

median (stats)

Median Value

Description

Compute the sample median.

Usage

```
median(x, na.rm = FALSE)
```

Diabetes3 <- Diabetes2 Copia de seguridad una vez más :o)

Calculamos mediana de cada columna

```
mean0 <- median(Diabetes3$AGE, na.rm = FALSE)
mean1 <- median(Diabetes3$BMI, na.rm = FALSE)
mean2 <- median(Diabetes3$BP, na.rm = FALSE)
mean3 <- median(Diabetes3$S1, na.rm = FALSE)
mean4 <- median(Diabetes3$S2, na.rm = FALSE)
mean5 <- median(Diabetes3$S3, na.rm = FALSE)
mean6 <- median(Diabetes3$S4, na.rm = FALSE)
mean7 <- median(Diabetes3$S5, na.rm = FALSE)
mean8 <- median(Diabetes3$S6, na.rm = FALSE)
mean9 <- median(Diabetes3$Y, na.rm = FALSE)
```

Calculamos Mad de cada columna

```
mad0 <- mad(Diabetes3$AGE, na.rm = FALSE)
mad1 <- mad(Diabetes3$BMI, na.rm = FALSE)
mad2 <- mad(Diabetes3$BP, na.rm = FALSE)
mad3 <- mad(Diabetes3$S1, na.rm = FALSE)
mad4 <- mad(Diabetes3$S2, na.rm = FALSE)
mad5 <- mad(Diabetes3$S3, na.rm = FALSE)
mad6 <- mad(Diabetes3$S4, na.rm = FALSE)
mad7 <- mad(Diabetes3$S5, na.rm = FALSE)
mad8 <- mad(Diabetes3$S6, na.rm = FALSE)
mad9 <- mad(Diabetes3$Y, na.rm = FALSE)
```

Y ahora para cada columna:

Guardamos en Val0 el valor que es 3 desviaciones más alto que el de la mediana

```
Val0 <- mean0 + 3*mad0
```

Y los valores de diabetes 3 que sean mayor que val0 los cambiamos por NA

```
Diabetes3[Diabetes3$AGE > Val0]<-NA
```

y después los quitamos

```
Diabetes3 <- na.omit(Diabetes3)
```

Guardamos en Valn0 el valor que es 3 desviaciones más bajo que el de la mediana

```
Valn0 <- mean0 - 3*mad0
```

Y los valores de diabetes 3 que sean menor que valn0 los cambiamos por NA

```
Diabetes3[Diabetes3$AGE < Valn0]<-NA
```

y después los quitamos

```
Diabetes3 <- na.omit(Diabetes3)
```

```
#
```

```
Val1 <- mean1 + 3*mad1
```

```
Diabetes3[Diabetes3$BMI > Val1]<-NA  
Diabetes3 <- na.omit(Diabetes3)
```

```
Valn1 <- mean1 - 3*mad1
```

```
Diabetes3[Diabetes3$BMI < Valn1]<-NA  
Diabetes3 <- na.omit(Diabetes3)
```

```
#
```

```
Val2 <- mean2 + 3*mad2
```

```
Diabetes3[Diabetes3$BP > Val2]<-NA  
Diabetes3 <- na.omit(Diabetes3)
```

```
Valn2 <- mean2 - 3*mad2
```

```
Diabetes3[Diabetes3$BP < Valn2]<-NA  
Diabetes3 <- na.omit(Diabetes3)
```

```
#
```

```
Val3<- mean3 + 3*mad3
```

```
Diabetes3[Diabetes3$s1 > Val3]<-NA  
Diabetes3 <- na.omit(Diabetes3)
```

```
Valn3 <- mean3 - 3*mad3
```

```
Diabetes3[Diabetes3$s1 < Valn3]<-NA  
Diabetes3 <- na.omit(Diabetes3)
```

```
#
```

```
Val4<- mean4 + 3*mad4
```

```
Diabetes3[Diabetes3$s2 > Val4]<-NA  
Diabetes3 <- na.omit(Diabetes3)
```

```
Valn4 <- mean4 - 3*mad4
```

```
Diabetes3[Diabetes3$s2 < Valn4]<-NA  
Diabetes3 <- na.omit(Diabetes3)
```

```
#
```

```
Val5<- mean5 + 3*mad5
```

```
Diabetes3[Diabetes3$s3 > Val5]<-NA  
Diabetes3 <- na.omit(Diabetes3)
```

```
Valn5 <- mean5 - 3*mad5  
  
Diabetes3[Diabetes3$s3 < Valn5]<-NA  
Diabetes3 <- na.omit(Diabetes3)
```

```
#
```

```
Val5<- mean5 + 3*mad5  
  
Diabetes3[Diabetes3$s3 > Val5]<-NA  
Diabetes3 <- na.omit(Diabetes3)
```

```
Valn5 <- mean5 - 3*mad5  
  
Diabetes3[Diabetes3$s3 < Valn5]<-NA  
Diabetes3 <- na.omit(Diabetes3)
```

```
#
```

```
Val6<- mean6 + 3*mad6  
  
Diabetes3[Diabetes3$s4 > Val6]<-NA  
Diabetes3 <- na.omit(Diabetes3)
```

```
Valn6 <- mean6 - 3*mad6  
  
Diabetes3[Diabetes3$s4 < Valn6]<-NA  
Diabetes3 <- na.omit(Diabetes3)
```

```
#
```

```
Val7 <- mean7 + 3*mad7  
  
Diabetes3[Diabetes3$s5 > Val7]<-NA  
Diabetes3 <- na.omit(Diabetes3)
```

```
Valn7 <- mean7 - 3*mad7  
  
Diabetes3[Diabetes3$s5 < Valn7]<-NA  
Diabetes3 <- na.omit(Diabetes3)
```

```
#
```

```
Val8 <- mean8 + 3*mad7  
  
Diabetes3[Diabetes3$s6 > Val8]<-NA  
Diabetes3 <- na.omit(Diabetes3)
```

```
Valn8 <- mean8 - 3*mad8  
  
Diabetes3[Diabetes3$s6 < Valn8]<-NA  
Diabetes3 <- na.omit(Diabetes3)
```

```
#
```



```
Val9<- mean9 + 3*mad9
```

```
Diabetes3[Diabetes3$Y > Val9]<-NA  
Diabetes3 <- na.omit(Diabetes3)
```

```
Valn9 <- mean9 - 3*mad9
```

```
Diabetes3[Diabetes3$Y < Valn9]<-NA  
Diabetes3 <- na.omit(Diabetes3)
```

Separar el conjunto de datos en dos, el primero (entrenamiento) conteniendo un 70% de los datos y el segundo (test) un 30%, de forma aleatoria.

Si tenemos 433 registros --> 70% son 303 y el 30% son 130

```
sample(diabetes, 130)
```

```
datos130 <- Diabetes3[sample(nrow(Diabetes3), 130), ]  
datos303 <- Diabetes3[sample(nrow(Diabetes3), 303), ]
```

```
datos303  
datos130
```

```
nrow(datos303)  
nrow(datos130)
```

Escalar los datos para que tengan media 0 y varianza 1, es decir, restar a cada variable numerica su media y dividir por la desviacion tpica. Calcular la media y desviacion en el conjunto de train, y utilizar esa misma media y desviacion para escalar el conjunto de test.

```
media_datos130<-apply(datos130,2,mean)  
sd_datos130<-apply(datos130,2,sd)  
datos130_normal<-scale(datos130,media_datos130,sd_datos130)  
summary(datos130_normal)  
apply(datos130_normal,2,sd)
```

```
> media_datos130<-apply(datos130,2,mean)  
> sd_datos130<-apply(datos130,2,sd)  
> datos130_normal<-scale(datos130,media_datos130,sd_datos130)  
> summary(datos130_normal)  
      AGE      SEX      BMI      BP  
Min.   :-2.1520  Min.   :-1.1099  Min.   :-1.8217  Min.   :-2.1323  
1st Qu.: -0.7990 1st Qu.: -1.1099 1st Qu.: -0.6965 1st Qu.: -0.7249  
Median :  0.1029 Median :  0.8941 Median : -0.1449 Median : -0.1773  
Mean    :  0.0000 Mean    :  0.0000 Mean    :  0.0000 Mean    :  0.0000  
3rd Qu.:  0.7042 3rd Qu.:  0.8941 3rd Qu.:  0.5169 3rd Qu.:  0.8648  
Max.    :  1.9068 Max.    :  0.8941 Max.    :  2.6570 Max.    :  2.6487  
      S1      S2      S3  
Min.   :-2.52419  Min.   :-2.15784  Min.   :-2.0986  
1st Qu.: -0.66917 1st Qu.: -0.66242 1st Qu.: -0.8029  
Median : -0.08775 Median : -0.06233 Median : -0.1932  
Mean    :  0.00000 Mean    :  0.00000 Mean    :  0.0000  
3rd Qu.:  0.52136 3rd Qu.:  0.58750 3rd Qu.:  0.6452  
Max.    :  2.70862 Max.    :  2.72633 Max.    :  3.6176  
      S4      S5      S6  
Min.   :-1.55306  Min.   :-2.05904  Min.   :-2.55722  
1st Qu.: -0.81307 1st Qu.: -0.67699 1st Qu.: -0.63792  
Median : -0.07309 Median :  0.03015 Median :  0.08182  
Mean    :  0.00000 Mean    :  0.00000 Mean    :  0.00000  
3rd Qu.:  0.66690 3rd Qu.:  0.62293 3rd Qu.:  0.72158  
Max.    :  3.09405 Max.    :  2.77661 Max.    :  2.40097  
      Y  
Min.   :-1.5870  
1st Qu.: -0.8402  
Median : -0.1120  
Mean    :  0.0000  
3rd Qu.:  0.7334  
Max.    :  2.2392  
`
```

```
> apply(datos130_normal,2,sd)
AGE SEX BMI BP S1 S2 S3 S4 S5 S6 Y
 1 1 1 1 1 1 1 1 1 1 1 1
> |
```

`media_datos130<-apply(datos130,2,mean)` calculo la media del grupo del 30%

`sd_datos130<-apply(datos130,2,sd)` aquí la desviación típica

`datos130_normal<-scale(datos130,media_datos130,sd_datos130)` Y por último el escalado de la distribución normal

`summary(datos130_normal)` Aquí tenemos los datos estadísticos

`apply(datos130_normal,2,sd)` Y por último la desviación típica

`media_datos330<-apply(datos330,2,mean)`

`sd_datos330<-apply(datos330,2,sd)`

`datos330_normal<-scale(datos330,media_datos330,sd_datos330)`

`summary(datos330_normal)`

`apply(datos330_normal,2,sd)`

```
> media_datos330<-apply(datos330,2,mean)
> sd_datos330<-apply(datos330,2,sd)
> datos330_normal<-scale(datos330,media_datos330,sd_datos330)
> summary(datos330_normal)
      AGE      SEX      BMI
Min.   :-2.24997 Min.   :-1.0353 Min.   :-1.8677
1st Qu.: -0.69987 1st Qu.: -1.0353 1st Qu.: -0.7373
Median :  0.09408 Median :  0.9627 Median : -0.1269
Mean    :  0.00000 Mean    :  0.0000 Mean    :  0.0000
3rd Qu.:  0.77461 3rd Qu.:  0.9627 3rd Qu.:  0.6644
Max.    :  2.28690 Max.    :  0.9627 Max.    :  2.7217
      BP      S1      S2
Min.   :-2.2644 Min.   :-2.62872 Min.   :-2.3839
1st Qu.: -0.7738 1st Qu.: -0.73215 1st Qu.: -0.6446
Median : -0.1349 Median : -0.06278 Median : -0.1136
Mean    :  0.0000 Mean    :  0.00000 Mean    :  0.0000
3rd Qu.:  0.6576 3rd Qu.:  0.60660 3rd Qu.:  0.6941
Max.    :  2.7043 Max.    :  3.06098 Max.    :  4.0010
      S3      S4      S5
Min.   :-2.1359 Min.   :-1.55479 Min.   :-2.56587
1st Qu.: -0.7143 1st Qu.: -0.80808 1st Qu.: -0.68840
Median : -0.1158 Median : -0.06136 Median : -0.06574
Mean    :  0.0000 Mean    :  0.00000 Mean    :  0.00000
3rd Qu.:  0.6324 3rd Qu.:  0.68535 3rd Qu.:  0.68044
Max.    :  3.6251 Max.    :  3.73941 Max.    :  2.72150
      S6      Y
Min.   :-2.83261 Min.   :-1.6132
1st Qu.: -0.72857 1st Qu.: -0.8517
Median :  0.02889 Median : -0.1663
Mean    :  0.00000 Mean    :  0.0000
3rd Qu.:  0.61802 3rd Qu.:  0.7728
Max.    :  2.72206 Max.    :  2.3974
> apply(datos330_normal,2,sd)
AGE SEX BMI BP S1 S2 S3 S4 S5 S6 Y
 1 1 1 1 1 1 1 1 1 1 1 1
> |
```