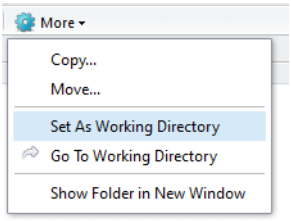
Proyecto Diabetes

- ✓ **Cargar los datos en R y eliminar los missing values, que están codificados como -9999.00.**

  Configuro como carpeta de trabajo el directorio donde tengo el fichero con los datos. De esta forma no tendré que usar la ruta entera a la hora de cargar.



A continuación cargo los datos y lo asigno como valor de una variable

Opción 1:

```
diabetes <- read.table("diabetes.data",header = TRUE) # importo datos con cabeceros, los valores no válidos por defecto son NA

#eliminar NA: como en el dataset todos los valores válidos son positivos pongo NA a los valores menores de cero

diabetes [diabetes$BMI < 0, 3] <- NA
diabetes [diabetes$BP < 0, 4] <- NA
diabetes [diabetes$S1 < 0 , 5] <- NA
diabetes [diabetes$S2 < 0 , 6] <- NA
diabetes [diabetes$S3 < 0 , 7]<- NA
diabetes [diabetes$S4 < 0 , 8]<- NA
diabetes [diabetes$S5 < 0 , 9]<- NA
diabetes [diabetes$S6 < 0 , 10]<- NA
diabetes [diabetes$Y < 0 , 11]<- NA


diabetes_sin_na <- diabetes[complete.cases(diabetes), ] #eligo solo los registros completos, sin NA
```

Opción 2:

```
# importo datos con cabeceros, los    valores no válidos por defecto son NA
diabetes <- read.table("diabetes.data",header = TRUE)

# con una función que busca valores menores de cero identifico los valores no válidos (menores de cero), al aplicar la función omito columna 2 que no es numérica
buscar_na <- function(x){x<0}
na <- apply(diabetes[ ,-2], 2, buscar_na)

# eligo solo los registros en los que no hay ningún valoe inválido, es decir: lo contrario al resultado de la función buscar_na
diabetes_sin_na <- diabetes[!apply(na, 1, any),]
```

Opción 3:

```
# importo datos con cabeceros y configuro los valores -9999.00 como NA
diabetes <- read.table("diabetes.data",header = TRUE, na.strings="-9999.0")

# eligo solo los registros completos, sin NA
diabetes_sin_na <- diabetes[complete.cases(diabetes), ]
```

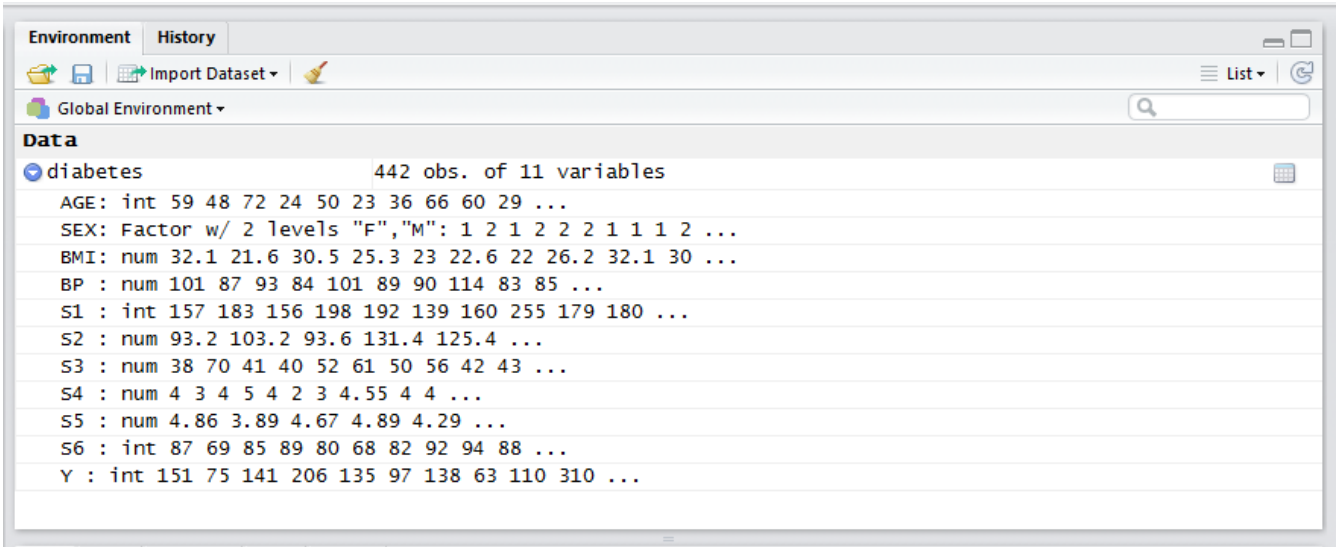
Filter

	AGE	SEX	BMI	BP	S1	S2	S3	S4	S5	S6	Y
1	59	F	32.1	101.00	157	93.2	38	4.00	4.8598	87	151
2	48	M	21.6	87.00	183	103.2	70	3.00	3.8918	69	75
3	72	F	30.5	93.00	156	93.6	41	4.00	4.6728	85	141
4	24	M	25.3	84.00	198	131.4	40	5.00	4.8903	89	206
5	50	M	23.0	101.00	192	125.4	52	4.00	4.2905	80	135
6	23	M	22.6	89.00	139	64.8	61	2.00	4.1897	68	97
7	36	F	22.0	90.00	160	99.6	50	3.00	3.9512	82	138
8	66	F	26.2	114.00	255	185.0	56	4.55	4.2485	92	63
9	60	F	32.1	83.00	179	119.4	42	4.00	4.4773	94	110
10	29	M	30.0	85.00	180	93.4	43	4.00	5.3845	88	310
11	22	M	18.6	97.00	114	57.6	46	2.00	3.9512	83	101
12	56	F	28.0	85.00	184	144.8	32	6.00	3.5835	77	69
13	53	M	23.7	92.00	186	109.2	62	3.00	4.3041	81	179
14	50	F	26.2	97.00	186	105.4	49	4.00	5.0626	88	185
15	61	M	24.0	91.00	202	115.4	72	3.00	4.2905	73	118
16	34	F	24.7	118.00	254	184.2	39	7.00	5.0370	81	171
17	47	M	30.3	109.00	207	100.2	70	3.00	5.2149	98	166
18	68	F	27.5	111.00	214	147.0	39	5.00	4.9416	91	144
19	38	M	25.4	84.00	162	103.0	42	4.00	-9999.0000	87	97
20	41	M	24.7	83.00	187	108.2	60	3.00	4.5433	78	168

Showing 1 to 20 of 442 entries

✔ **Ver el tipo de cada una de las variables.**

Puedo ver el tipo de cada variable en la ventana a la derecha:



También puedo obtener la información con:

str(diabetes_sin_na)

```
> str(diabetes_sin_na)
'data.frame':   433 obs. of  11 variables:
 $ AGE: int  59 48 72 24 50 23 36 66 60 29 ...
 $ SEX: Factor w/ 2 levels "F","M": 1 2 1 2 2 2 1 1 1 2 ...
 $ BMI: num  32.1 21.6 30.5 25.3 23 22.6 22 26.2 32.1 30 ...
 $ BP : num  101 87 93 84 101 89 90 114 83 85 ...
 $ S1 : int  157 183 156 198 192 139 160 255 179 180 ...
 $ S2 : num  93.2 103.2 93.6 131.4 125.4 ...
 $ S3 : num  38 70 41 40 52 61 50 56 42 43 ...
 $ S4 : num  4 3 4 5 4 2 3 4.55 4 4 ...
 $ S5 : num  4.86 3.89 4.67 4.89 4.29 ...
 $ S6 : int  87 69 85 89 80 68 82 92 94 88 ...
 $ Y  : int  151 75 141 206 135 97 138 63 110 310 ...
```

#también puedo ver tipo de cada variable por separado
class(diabetes_sin_na [[1]])
class(diabetes_sin_na [[2]])
class(diabetes_sin_na [[3]])
class(diabetes_sin_na [[4]])
class(diabetes_sin_na [[5]])
class(diabetes_sin_na [[6]])
class(diabetes_sin_na [[7]])
class(diabetes_sin_na [[8]])
class(diabetes_sin_na [[9]])
class(diabetes_sin_na [[10]])
class(diabetes_sin_na [[11]])

✔ **Realizar un an´alisis estadístico de las variables: calcular la media, varianza, rangos, etc. ¿Tienen las distintas variables rangos muy diferentes?.**

calculo min, max, media, mediana y los cuartiles para cada variable

summary(diabetes_sin_na)

```
> summary(diabetes_sin_na)          # calculo min, max, media, mediana y los cuartiles par
      AGE        SEX          BMI              BP              S1              S2
 Min.   :19.00   F:203   Min.   :18.00   Min.   : 62.00   Min.   : 97.0   Min.   : 41.6
 1st Qu.:38.00   M:230   1st Qu.:23.10   1st Qu.: 84.00   1st Qu.:164.0   1st Qu.: 95.4
 Median :50.00           Median :25.70   Median : 93.00   Median :186.0   Median :113.0
 Mean   :48.48           Mean   :26.35   Mean   : 94.65   Mean   :189.3   Mean   :115.4
 3rd Qu.:59.00           3rd Qu.:29.20   3rd Qu.:105.00   3rd Qu.:210.0   3rd Qu.:134.2
 Max.   :79.00           Max.   :42.20   Max.   :133.00   Max.   :301.0   Max.   :242.4
       S3              S4              S5              S6              Y
 Min.   :22.00   Min.   :2.000   Min.   :3.258   Min.   : 58.00   Min.   : 25.0
 1st Qu.:40.00   1st Qu.:3.000   1st Qu.:4.277   1st Qu.: 83.00   1st Qu.: 85.0
 Median :48.00   Median :4.000   Median :4.635   Median : 91.00   Median :140.0
 Mean   :49.86   Mean   :4.071   Mean   :4.645   Mean   : 91.25   Mean   :152.2
 3rd Qu.:58.00   3rd Qu.:5.000   3rd Qu.:4.997   3rd Qu.: 98.00   3rd Qu.:214.0
 Max.   :99.00   Max.   :9.090   Max.   :6.107   Max.   :124.00   Max.   :346.0
```

calculo la varianza para cada variable, excluyendo la segunda que no es numérica

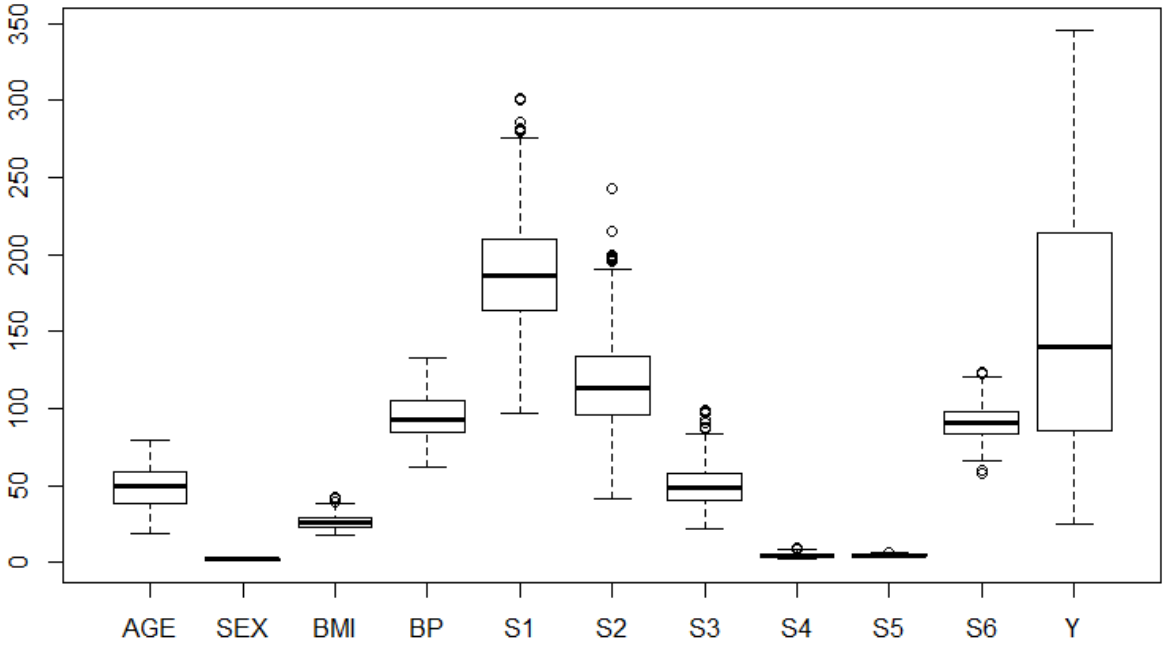
apply(diabetes_sin_na[-2],2,var)

```
> apply(diabetes_sin_na[-2],2,var)  # calculo la varianza para cada
le, excluyendo la segunda que no es numérica
        AGE         BMI          BP          S1          S2
173.4814708  19.6089824 194.2240393 1210.2951629 936.8608596
         S3          S4          S5          S6           Y
169.9568204   1.6894878   0.2757557 131.6506073 6013.5473441
```

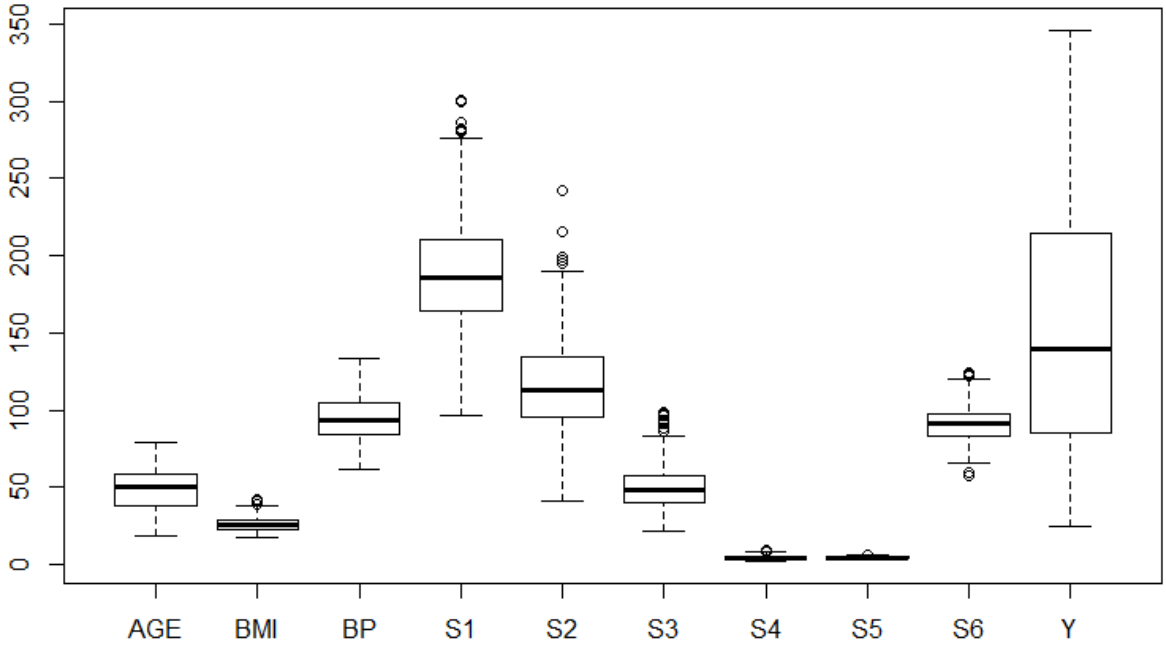
Observamos que las variables tienen rangos muy diferentes: el mínimo y máximo difieren mucho según variable.

✔ **Hacer un gráfico de cajas (boxplot) dónde se pueda ver la información anterior de forma gráfica.**

boxplot(diabetes_sin_na)



boxplot(diabetes_sin_na[-2]) # excluyendo variable SEX que son factores



✔ **Calcular la media para las filas que tienen SEX=M y la media para las filas que tienen SEX=F, utilizando la función tapply.**

sapply(diabetes_sin_na[, -2], tapply, diabetes_sin_na$SEX,mean)

```
> sapply(diabetes_sin_na[ , -2], tapply, diabetes_sin_na$SEX,mean)
       AGE      BMI       BP       S1       S2       S3       S4       S5       S6        Y
F 50.86700 26.80099 98.17562 190.6552 120.1103 44.54187 4.537882 4.731533 93.86207 155.8079
M 46.36522 25.95609 91.53474 188.0304 111.1726 54.56304 3.658217 4.569392 88.94348 149.0391
```

También podemos obtener las medias para cada variable por separado

tapply(diabetes_sin_na$AGE,diabetes_sin_na$SEX,mean)
tapply(diabetes_sin_na$BMI,diabetes_sin_na$SEX,mean)
tapply(diabetes_sin_na$BP,diabetes_sin_na$SEX,mean)
tapply(diabetes_sin_na$S1,diabetes_sin_na$SEX,mean)
tapply(diabetes_sin_na$S2,diabetes_sin_na$SEX,mean)
tapply(diabetes_sin_na$S3,diabetes_sin_na$SEX,mean)
tapply(diabetes_sin_na$S4,diabetes_sin_na$SEX,mean)
tapply(diabetes_sin_na$S5,diabetes_sin_na$SEX,mean)
tapply(diabetes_sin_na$S6,diabetes_sin_na$SEX,mean)
tapply(diabetes_sin_na$Y,diabetes_sin_na$SEX,mean)

```
> tapply(diabetes_sin_na$AGE,diabetes_sin_na$SEX,mean)
       F        M
50.86700 46.36522
```

✔ **Calcular la correlación de todas las variables numéricas con la variable Y.**

```
cor(diabetes_sin_na[,-2],diabetes_sin_na$Y) # excluyo la segunda columna que no es numérica
```

```
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
```

```
# también puedo calcular cada correlacion por separado

cor(diabetes_sin_na$AGE,diabetes_sin_na$Y)
cor(diabetes_sin_na$BMI,diabetes_sin_na$Y)
cor(diabetes_sin_na$BP,diabetes_sin_na$Y)
cor(diabetes_sin_na$S1,diabetes_sin_na$Y)
cor(diabetes_sin_na$S2,diabetes_sin_na$Y)
cor(diabetes_sin_na$S3,diabetes_sin_na$Y)
cor(diabetes_sin_na$S4,diabetes_sin_na$Y)
cor(diabetes_sin_na$S5,diabetes_sin_na$Y)
cor(diabetes_sin_na$S6,diabetes_sin_na$Y)
```

✔ **Transformar la variable SEX, que es un factor, en una variable numérica utilizando, por ejemplo, la codificación M=1 y F=2.**

```
diabetes_sin_na$SEX<- as.numeric(diabetes_sin_na$SEX)
```

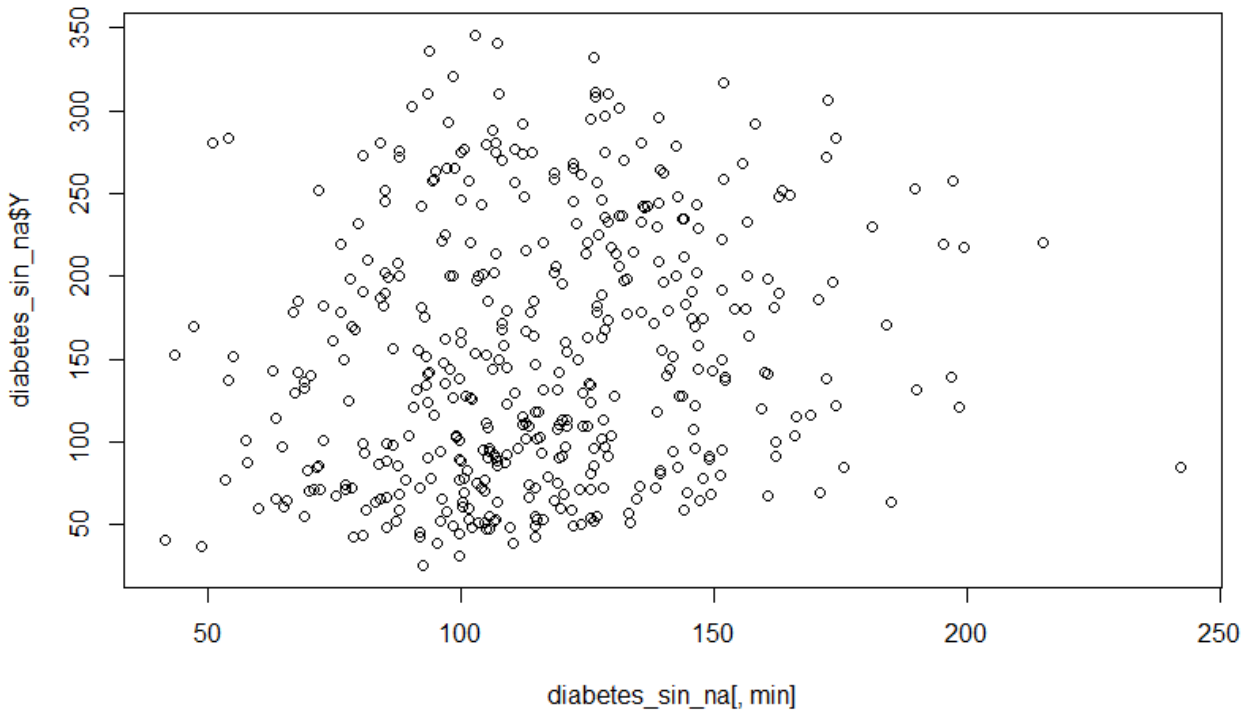
```
> head(diabetes_sin_na)
  AGE SEX  BMI  BP  S1    S2 S3 S4     S5 S6   Y
1  59   1 32.1 101 157  93.2 38  4 4.8598 87 151
2  48   2 21.6  87 183 103.2 70  3 3.8918 69  75
3  72   1 30.5  93 156  93.6 41  4 4.6728 85 141
4  24   2 25.3  84 198 131.4 40  5 4.8903 89 206
5  50   2 23.0 101 192 125.4 52  4 4.2905 80 135
6  23   2 22.6  89 139  64.8 61  2 4.1897 68  97
```

✔ **Realizar un gráfico de dispersión para las variables que tienen más y menos correlación con Y y comentar los resultados. ¿Como sería el gráfico de dispersión entre dos cosas con correlación 1?**

```
# obtenemos la correlación mínima y máxima de todas las correlaciones con y. Excluyo la columna de sexo y la Y.
range(abs(cor(diabetes_sin_na[,c(1,3:10)],diabetes_sin_na$Y)))
```

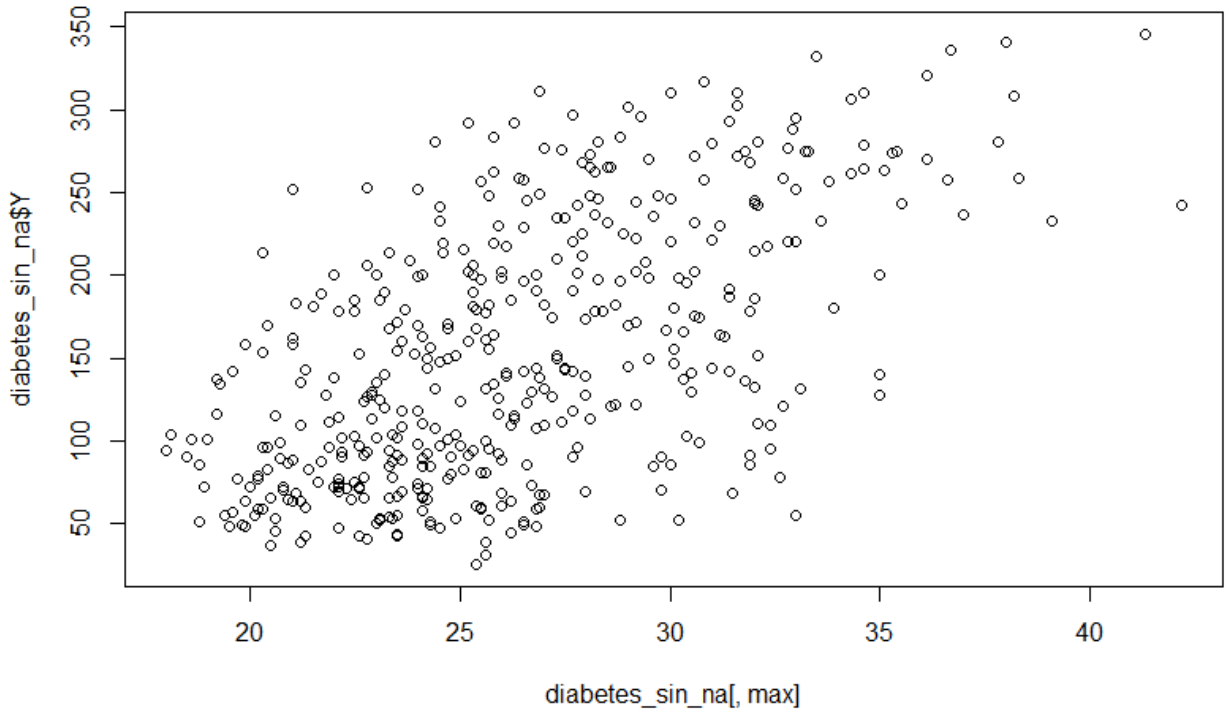
```
[1] 0.1747189 0.5863673
```

```
# obtenemos el índice de las columnas que tienen correlación minima y maxima

 minmax <-

 match(range(abs(cor(diabetes_sin_na[,c(1,3:10)],diabetes_sin_na$Y))),abs(cor(diabetes_si
 n_na,diabetes_sin_na$Y)))

# obtenemos el índice en diabetes_sin_na de las columnas que tienen correlación mínima y
máxima

minmax
[1] 6 3

# vemos los índices que tienen las columnas con correlación mínima y máxima con y: son
las variables BMI (max) y s2 (min) 3 y 6

min <- minmax[1]                    # indice de la variable con menos correlación con Y
max <- minmax [2]                   # indice de la variable con mas correlación con Y


# grafico de la dispersion de la variable con la correlacon minima con y
plot(diabetes_sin_na[ ,min],diabetes_sin_na$Y)
```



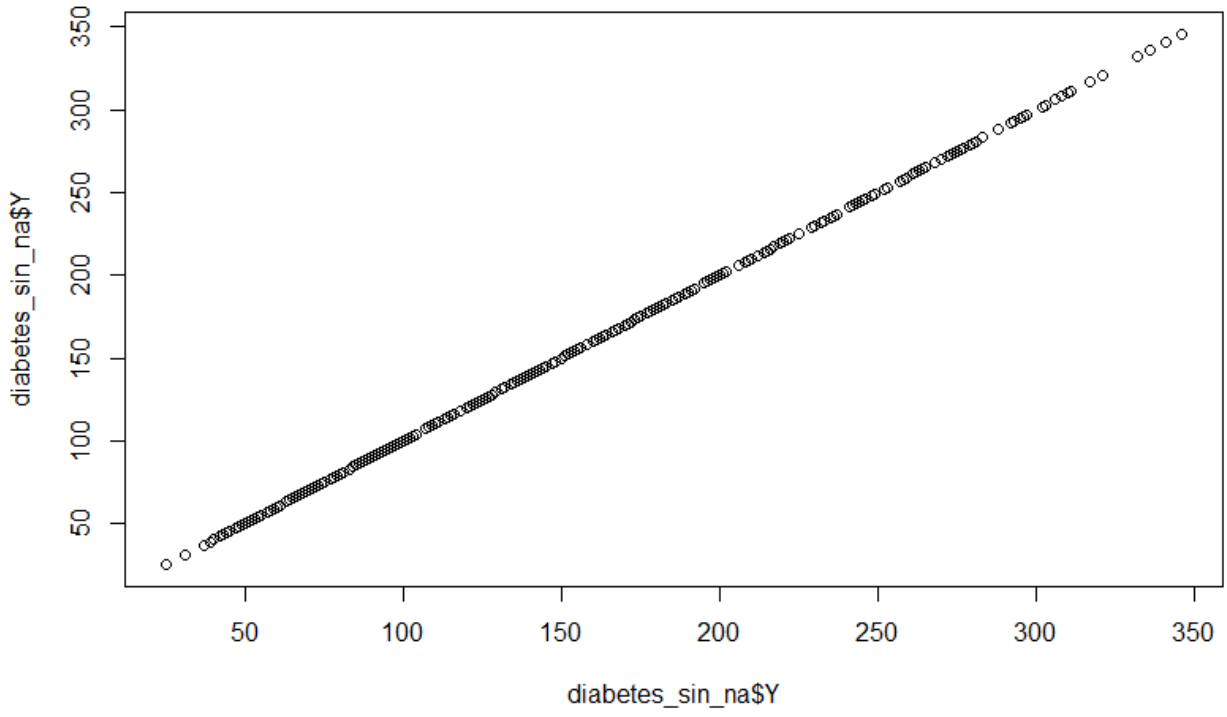
Se puede ver en el gráfico que los puntos están dispersos y no se observa correlación.

```
# grafico de la dispersion de la variable con la correlacon maxima con y
plot(diabetes_sin_na[ ,max],diabetes_sin_na$Y)
```



Se puede ver que los puntos muestran mayor correlación que en le punto anterior.

```
# corelación 1 tiene y consigo mismo, el gráfio es una recta
plot(diabetes_sin_na$Y,diabetes_sin_na$Y)
```




```
# correlación 1 tiene y consigo mismo, el gráfico es una recta
```

✔ **Definimos los outliers como los elementos (filas) de los datos para los que cualquiera de las variables est´a por encima o por debajo de la mediana m´as/menos 3 veces el MAD (Median Absolute Deviation). Identificar estos outliers y quitarlos.**

```
apply(diabetes_sin_na, 2, median, na.rm = TRUE)     # calcula mediana
apply(diabetes_sin_na, 2, mad, na.rm = TRUE)   # calcula median absolute deviation(MAD)

# definimos que es outlier
outlier <-
function(x, const=3){x < median(x) - const*mad(x) | x > median(x) + const*mad(x)}

# buscar outlier en los datos
buscar_outlier <- apply(diabetes_sin_na[ ,-2], 2, outlier)

# seleccionamos outlier
si_outliers <- diabetes_sin_na[apply(buscar_outlier, 1, any),]

# seleccionamos todos menos outlier
no_outliers <- diabetes_sin_na[!apply(buscar_outlier, 1, any),]
```

✔ **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.**

```
test<-diabetes_sin_na[sample(nrow(diabetes_sin_na),nrow(diabetes_sin_na)*0.3,
replace=FALSE), ]
entrenamiento<-diabetes_sin_na[sample(nrow(diabetes_sin_na),nrow(diabetes_sin_na)*0.7,
replace=FALSE), ]
```

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

```
# escalando el conjunto test:

media_test<-apply(test,2,mean)                  # calculo la media por variable
sd_test<-apply(test,2,sd)                       # calculo desviación típica
test_normal<-scale(test,media_test,sd_test)     # cescalando a la distribución normal N(0,1)

summary(test_normal)                            # veo que las medias son 0
```

```
      AGE                SEX                BMI                BP                 S1                 S2          
 Min.   :-2.06467   Min.   :-1.0196   Min.   :-1.7802   Min.   :-2.2094   Min.   :-2.40488   Min.   :-2.10955  
 1st Qu.:-0.71159   1st Qu.:-1.0196   1st Qu.:-0.6762   1st Qu.:-0.7502   1st Qu.:-0.72320   1st Qu.:-0.66907  
 Median : 0.07177   Median : 0.9732   Median :-0.1784   Median :-0.1713   Median :-0.02918   Median :-0.03576  
 Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.00000  
 3rd Qu.: 0.78391   3rd Qu.: 0.9732   3rd Qu.: 0.6442   3rd Qu.: 0.7091   3rd Qu.: 0.53138   3rd Qu.: 0.44854  
 Max.   : 2.20820   Max.   : 0.9732   Max.   : 2.7872   Max.   : 2.5852   Max.   : 3.04055   Max.   : 3.09976  
       S3                 S4                 S5                 S6                 Y          
 Min.   :-1.9157   Min.   :-1.6415   Min.   :-2.143162   Min.   :-2.78302   Min.   :-1.63462  
 1st Qu.:-0.7142   1st Qu.:-0.8708   1st Qu.:-0.636168   1st Qu.:-0.67927   1st Qu.:-0.83555  
 Median :-0.1135   Median :-0.1001   Median : 0.004454   Median : 0.05247   Median :-0.04916  
 Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.000000   Mean   : 0.00000   Mean   : 0.00000  
 3rd Qu.: 0.3370   3rd Qu.: 0.6706   3rd Qu.: 0.719874   3rd Qu.: 0.60127   3rd Qu.: 0.58502  
 Max.   : 3.7161   Max.   : 2.9826   Max.   : 2.857412   Max.   : 2.88795   Max.   : 2.31000  
```

```
apply(test_normal,2,sd)                    # veo que las varianzas son 1
```

```
> apply(test_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
```

```
# escalando el conjunto entrenamiento:

media_entrenamiento<-apply(entrenamiento,2,mean)  # calculo la media por variable
sd_entrenamiento<-apply(entrenamiento,2,sd)       # calculo desviación típica

# cescalando a la distribución normal N(0,1)
entrenamiento_normal<-scale(entrenamiento,media_entrenamiento,sd_entrenamiento)

summary(entrenamiento_normal)  # veo que las medias son 0
```

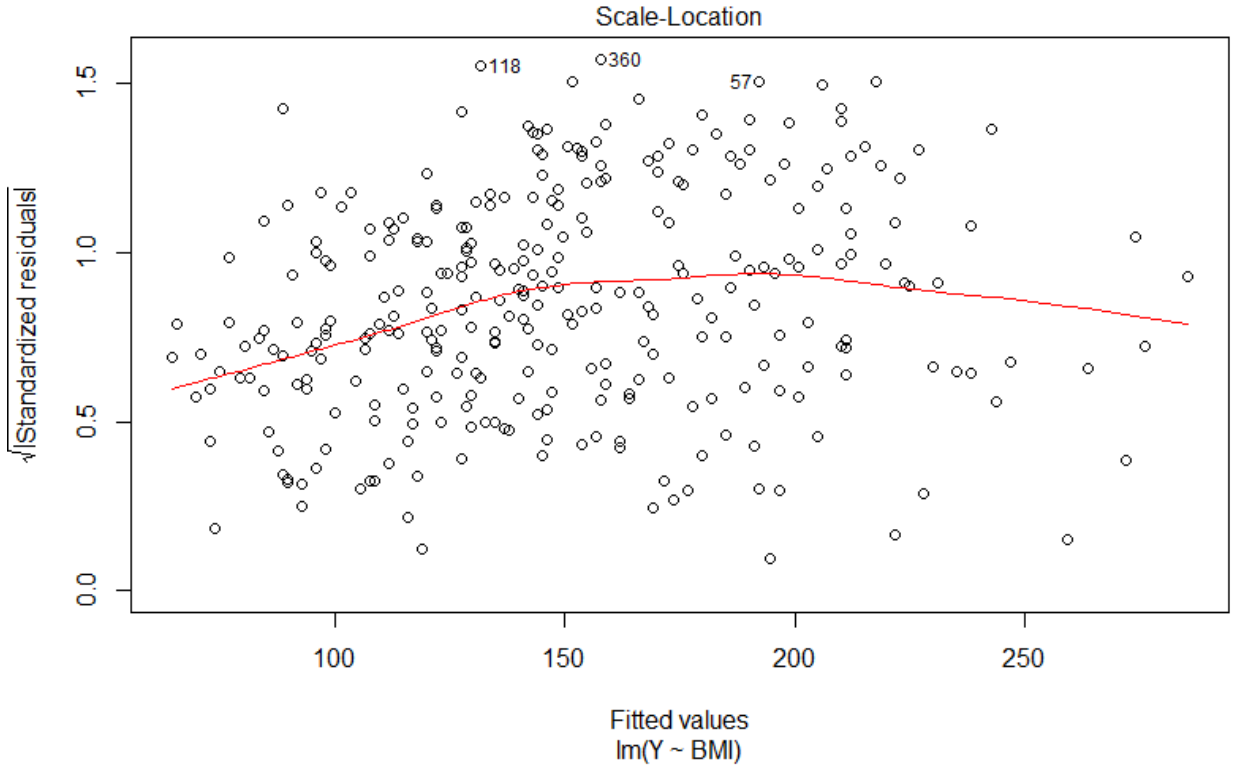
```
> summary(entrenamiento_normal)
      AGE                SEX                BMI                BP                S1                S2
 Min.   :-2.18444   Min.   :-1.0772   Min.   :-1.8897   Min.   :-2.3108   Min.   :-2.63018   Min.   :-2.41311
 1st Qu.:-0.76425   1st Qu.:-1.0772   1st Qu.:-0.7139   1st Qu.:-0.7449   1st Qu.:-0.69294   1st Qu.:-0.63114
 Median : 0.05797   Median : 0.9252   Median :-0.1203   Median :-0.1044   Median :-0.06619   Median :-0.07449
 Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.00000
 3rd Qu.: 0.84282   3rd Qu.: 0.9252   3rd Qu.: 0.6713   3rd Qu.: 0.7497   3rd Qu.: 0.63178   3rd Qu.: 0.53487
 Max.   : 2.30039   Max.   : 0.9252   Max.   : 3.0228   Max.   : 2.6002   Max.   : 3.18153   Max.   : 4.20091
       S3                 S4                 S5                 S6                 Y
 Min.   :-2.1270   Min.   :-1.58721   Min.   :-2.205089   Min.   :-2.858355   Min.   :-1.6298
 1st Qu.:-0.7411   1st Qu.:-0.80032   1st Qu.:-0.710987   1st Qu.:-0.696178   1st Qu.:-0.8450
 Median :-0.1043   Median :-0.01343   Median :-0.006833   Median :-0.004281   Median :-0.1387
 Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.000000   Mean   : 0.000000   Mean   : 0.0000
 3rd Qu.: 0.6073   3rd Qu.: 0.77347   3rd Qu.: 0.694449   3rd Qu.: 0.601128   3rd Qu.: 0.6658
 Max.   : 3.6414   Max.   : 3.99186   Max.   : 2.794082   Max.   : 2.849792   Max.   : 2.5036
```

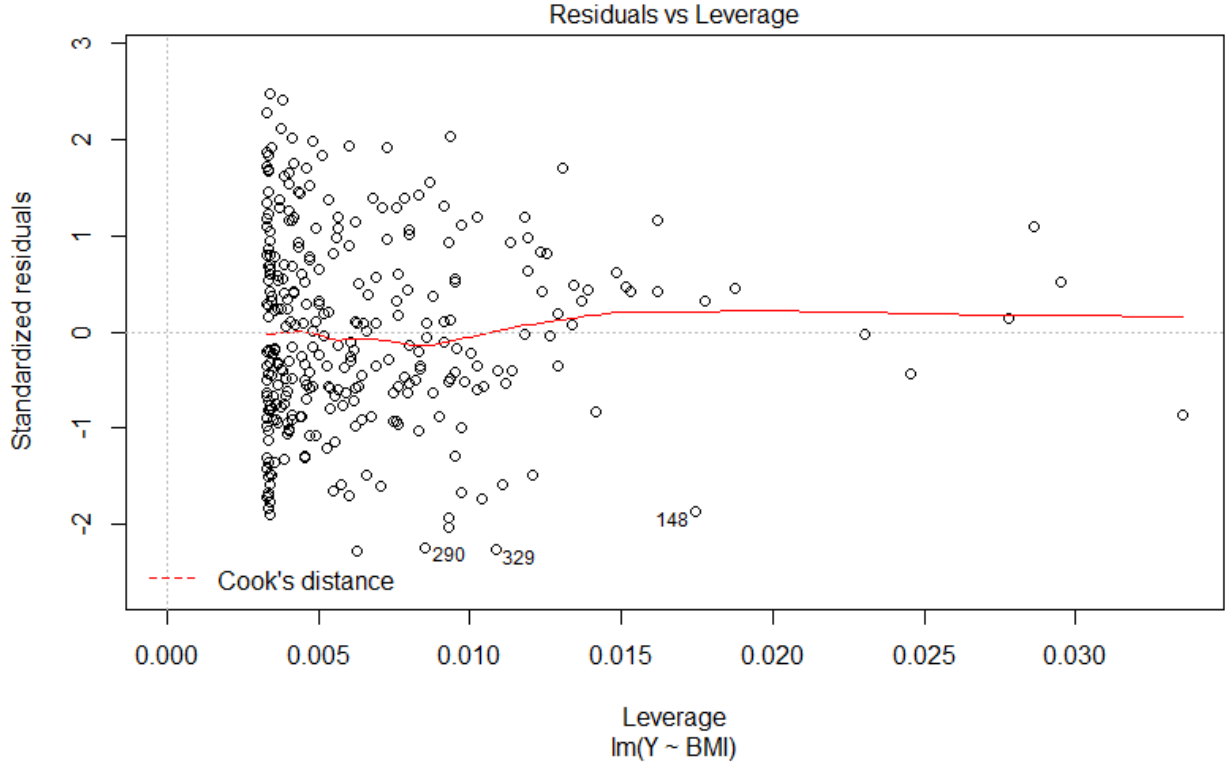
```
apply(entrenamiento_normal,2,sd)  # veo que las varianzas son 1
```

```
> apply(entrenamiento_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
```

✔ **Realizar un modelo de regresión lineal de la variable de respuesta sobre el resto y ajustarlo por mínimos cuadrados usando únicamente los datos del conjunto de entrenamiento.**

```
# modelo de regresión de la variable BMI sobre Y
regresion_entrenamiento <- lm(Y ~ BMI, data=entrenamiento)

#puedo ver los coeficientes de la regresión:

Coefficients:
(Intercept)          BMI
   -123.89        10.47

# puedo graficar la¡s características de la regresión:
plot(regresion_entrenamiento)
```


Scale-Location
√|Standardized residuals|
118
360
57
0.0
0.5
1.0
1.5
100
150
200
250
Fitted values
lm(Y ~ BMI)


Residuals vs Leverage
Standardized residuals
148
290
329
Cook's distance
-2
-1
0
1
2
3
0.000
0.005
0.010
0.015
0.020
0.025
0.030
Leverage
lm(Y ~ BMI)

✔ **Calcular el error cuadrático medio de los datos del conjunto de entrenamiento y de los datos del conjunto de test, definido como**

$$\mathrm{ECM} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

donde y es el vector de respuesta de los datos y ˆ y es el vector que predice el modelo (para los mismos datos).

```
# conjunto entrenamiento

# Para el cálculo necesito los valores predichos por el modelo y los valores reales.  Los valores reales los tengo en  entrenamiento predichos los calculo de la siguiente forma:

predicho_entrenamiento<- predict(regresion_entrenamiento)

# el error cuadratico medio será una media de las diferencias entre valores reales y predicciones al cuadrado

ecm_entrenamiento <- mean((entrenamiento-predicho_entrenamiento)²)
ecm_entrenamiento

[1] 12075.61

# conjunto test: hago los mismos cálculos

regresion_test <- lm(Y ~ BMI, data=test)
predicho_test <- predict(regresion_test)
ecm_test <- mean((test-predicho_test)^2)

ecm_test

[1] 13129.76
```