Práctica Métodos Clásicos Para La Predicción

Javier Gil Domínguez, David Lazaro Martin, Nicolas Vega Muñoz

2022-11-25

EJERCICIO 1

El Instituto Nacional de la Diabetes y las Enfermedades Digestivas y Renales (EE.UU.) realizó un estudio sobre 768 mujeres indias Pima adultas que vivían cerca de Phoenix. Se registraron las siguientes variables: • número de embarazos, • concentración de glucosa en plasma a las 2 horas en una prueba de tolerancia a la glucosa oral, • presión arterial diastólica (mm Hg), • grosor del pliegue cutáneo del tríceps (mm), • insulina sérica a las 2 horas (mu U/ml), • índice de masa corporal (peso en kg/(altura en m2)), • función de pedigrí de la diabetes, • edad (años) y • una prueba de si la paciente muestra signos de diabetes (codificada como 0 si es negativa, 1 si es positiva). Los datos pueden obtenerse en el repositorio de bases de datos de aprendizaje automático de la UCI en http://www.ics.uci.edu/Ëœmlearn/MLRepository.html. El correspondiente dataset está disponible en el conjunto de datos pima del paquete faraway. Se solicita utilizar la información anterior para construir un modelo de regresión lineal que permita predecir la concentración de glucosa en plasma a las 2 horas en una prueba de tolerancia a la glucosa oral. En concreto, se solicita:

##a) Importar y preparar las variables (variables categóricas como factors y poner etiquetas para los posibles valores de las variables categóricas)##

```
library(faraway)
```

```
## Warning: package 'faraway' was built under R version 4.1.3
```

```
data(pima)
pima$test <- factor(pima$test, labels=c("negative", "positive"))
```

##b) Hacer un estudio de estadística descriptiva sobre las variables disponibles, incluyendo el análisis de valores ausentes y atípicos##

Primero comprobamos la existencia de valores ausentes:

```
sum(is.na(pima)==TRUE)
```

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

```
summary(pima)
```

```
##     pregnant         glucose        diastolic         triceps
##  Min.   : 0.000   Min.   :  0.0   Min.   :  0.00   Min.   : 0.00
##  1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 62.00   1st Qu.: 0.00
##  Median : 3.000   Median :117.0   Median : 72.00   Median :23.00
##  Mean   : 3.845   Mean   :120.9   Mean   : 69.11   Mean   :20.54
##  3rd Qu.: 6.000   3rd Qu.:140.2   3rd Qu.: 80.00   3rd Qu.:32.00
```

```
##  Max.   :17.000   Max.   :199.0   Max.   :122.00   Max.   :99.00
##     insulin           bmi           diabetes           age
##  Min.   :  0.0   Min.   : 0.00   Min.   :0.0780   Min.   :21.00
##  1st Qu.:  0.0   1st Qu.:27.30   1st Qu.:0.2437   1st Qu.:24.00
##  Median : 30.5   Median :32.00   Median :0.3725   Median :29.00
##  Mean   : 79.8   Mean   :31.99   Mean   :0.4719   Mean   :33.24
##  3rd Qu.:127.2   3rd Qu.:36.60   3rd Qu.:0.6262   3rd Qu.:41.00
##  Max.   :846.0   Max.   :67.10   Max.   :2.4200   Max.   :81.00
##        test
##  negative:500
##  positive:268
##
##
##
##
```

Vemos que nuestro dataset está formado por 9 variables de las cuales test es categórica y está compuesta por 500 valores negative y 268 positive. También vemos que tenemos valores iguales a 0 en todas las variables excepto en diabetes y age. Consultando en internet, consideramos que estos valores (menos en la columna pregnant) se deben a un error ya que no es posible tener unos niveles tan bajos de, por ejemplo, glucosa. Por lo tanto, vamos a sustituirlos por NA's (valores ausentes en R). Posteriormente decidiremos como tratar dichos valores ausentes.

```
pima$diastolic[pima$diastolic == 0] <- NA
pima$glucose[pima$glucose == 0] <- NA
pima$triceps[pima$triceps == 0] <- NA
pima$insulin[pima$insulin == 0] <- NA
pima$bmi[pima$bmi == 0] <- NA

print(paste("Valores ausentes en diastolic:",sum(is.na(pima$diastolic)==TRUE)))
```

```
## [1] "Valores ausentes en diastolic: 35"
```

```
print(paste("Valores ausentes en glucose:",sum(is.na(pima$glucose)==TRUE)))
```

```
## [1] "Valores ausentes en glucose: 5"
```

```
print(paste("Valores ausentes en triceps:",sum(is.na(pima$triceps)==TRUE)))
```

```
## [1] "Valores ausentes en triceps: 227"
```

```
print(paste("Valores ausentes en insulin:",sum(is.na(pima$insulin)==TRUE)))
```

```
## [1] "Valores ausentes en insulin: 374"
```

```
print(paste("Valores ausentes en bmi:",sum(is.na(pima$bmi)==TRUE)))
```

```
## [1] "Valores ausentes en bmi: 11"
```

```r
print(paste("Numero total de Valores ausentes:",sum(is.na(pima)==TRUE)))
```

```
## [1] "Numero total de Valores ausentes: 652"
```

```r
which_nas <- apply(pima, 1, function(X) any(is.na(X)))
#eliminamos las filas que esten completamente vacias en caso de que haya
pima = pima[rowSums(is.na(pima)) != ncol(pima),]
print(paste("Numero de filas con algun valor ausente:",length(which(which_nas))))
```

```
## [1] "Numero de filas con algun valor ausente: 376"
```

Observamos que hay una gran cantidad de valores ausentes (652), sobre todo en las variables triceps e insulina. Puesto que estos aparecen repartidos en 376 filas no consideramos oportuno eliminarlas, por lo que procedemos a imputar los valores mediante KNN, estableciendo el numero de vecinos a 10.

```r
library(multiUS)
```

```
## Warning: package 'multiUS' was built under R version 4.1.3
```

```r
pima=KNNimp(pima, k = 10, scale = TRUE)
```

#GRÁFICOS UNIDIMENSIONALES

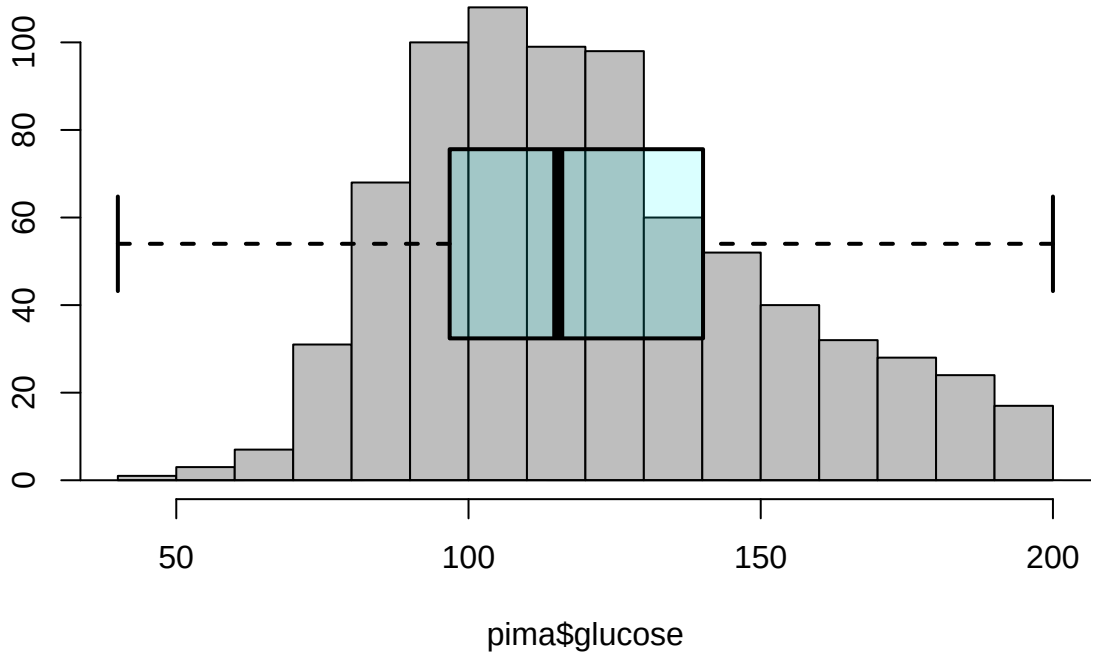
Con estos gráficos podemos hacernos una idea de la distribución de nuestros datos de algunas columnas.

Analizamos la glucosa, que es la variable que tenemos que predecir. Observamos que tiene una distribución similar a una normal de media 121.7 y no hay valores atípicos.

```r
summary(pima$glucose)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    44.0    99.0   117.0   121.7   141.0   199.0
```

```r
hist(pima$glucose, ylab = "", col = "grey", main = "", breaks=20)
lines(density(pima$glucose, na.rm = TRUE))
par(new = TRUE) # Esto indica que lo dibujaremos sobre el gráfico anterior
boxplot(pima$glucose, horizontal = TRUE, axes = FALSE, lwd = 2, col = rgb(0, 1, 1, alpha = 0.15))
```

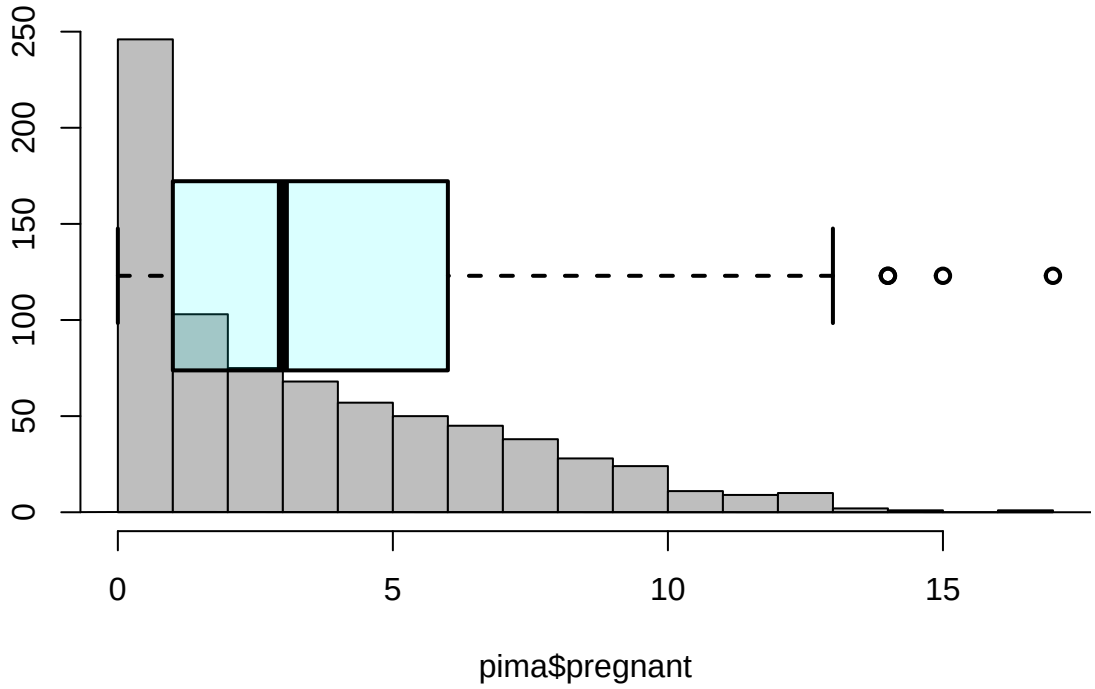


Para el número de embarazos observamos que la mayoria tiene entre 0 y 3 hijos y que existen algunos valores atípicos. Estos datos, a pesar de ser atípicos, no los eliminaremos puesto que no podemos consultar a un experto, son escasos, y tampoco podemos asegurar que sean imposibles.

```
summary(pima$pregnant)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   1.000   3.000   3.845   6.000  17.000
```

```
hist(pima$pregnant, ylab = "", col = "grey", main = "", breaks=20)
lines(density(pima$pregnant)) # lines indica que se va a dibujar una línea sobre el gráfico anterior
par(new = TRUE) # Esto indica que lo dibujaremos sobre el gráfico anterior
boxplot(pima$pregnant, horizontal = TRUE, axes = FALSE, lwd = 2, col = rgb(0, 1, 1, alpha = 0.15))
```

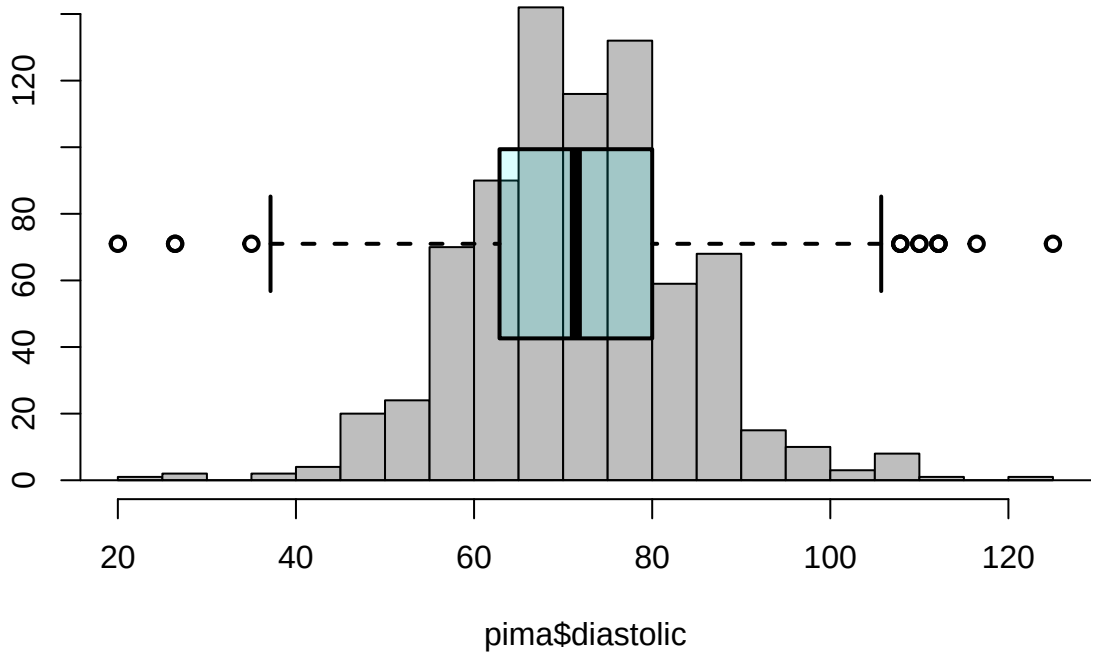


A continuación vamos a analizar la presión arterial diastólica. Tiene una distribución similar a una normal con media entorno a 72, coincidente con la mediana. Vemos que la distribucion concuerda con los valores normales de la presión diastólica (entre 60 y 80) puesto que el primer cuartil es 64 y el tercero 80 exactamente, el 50% de los datos se encuentra entre ellos. Existen pocos valores atipicos por lo que supondremos que los datos son correctos.

```
summary(pima$diastolic)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   24.00   64.00   72.00   72.31   80.00  122.00
```

```
hist(pima$diastolic, ylab = "", col = "grey", main = "", breaks = 20)
lines(density(pima$diastolic, na.rm = TRUE))
par(new = TRUE) # Esto indica que lo dibujaremos sobre el gráfico anterior
boxplot(pima$diastolic, horizontal = TRUE, axes = FALSE, lwd = 2, col = rgb(0, 1, 1, alpha = 0.15))
```

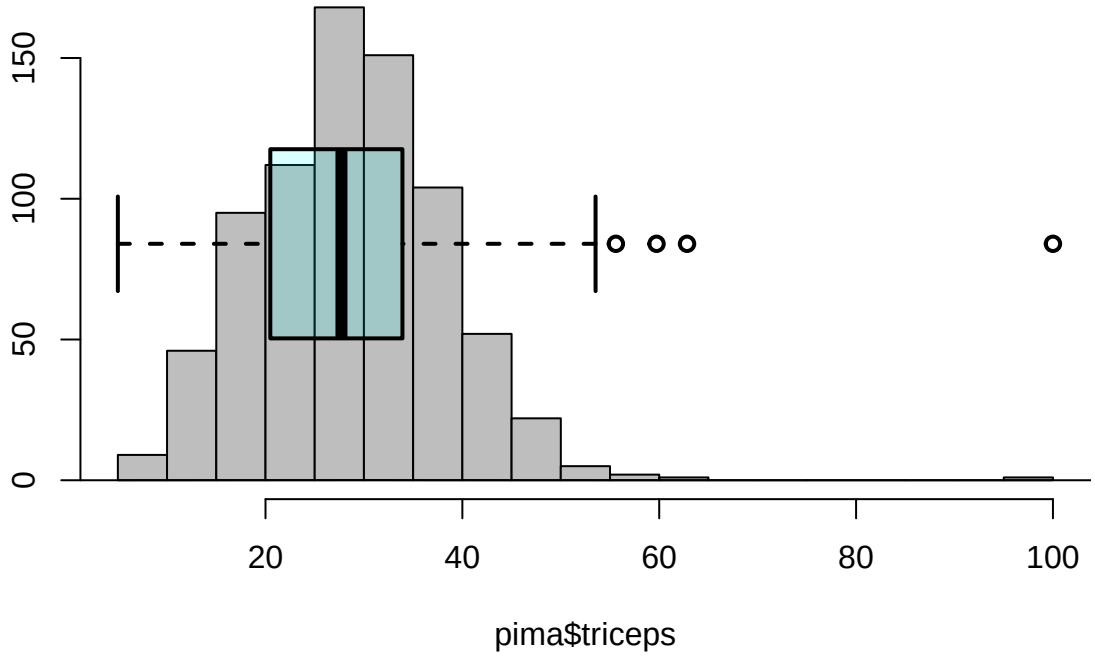


La variable triceps tiene pocos datos atipicos pero uno de ellos está especialmente alejado del resto. Este lo interpretamos como un dato atipico y por tanto procedemos a eliminarlo (nos quedamos arbitrariamente con los valores menores que 80).

```
summary(pima$triceps)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    7.00   22.00   29.00   29.06   35.00   99.00
```

```
hist(pima$triceps, ylab = "", col = "grey", main = "", breaks = 30)
lines(density(pima$triceps, na.rm = TRUE))
par(new = TRUE) # Esto indica que lo dibujaremos sobre el gráfico anterior
boxplot(pima$triceps, horizontal = TRUE, axes = FALSE, lwd = 2, col = rgb(0, 1, 1, alpha = 0.15))
```



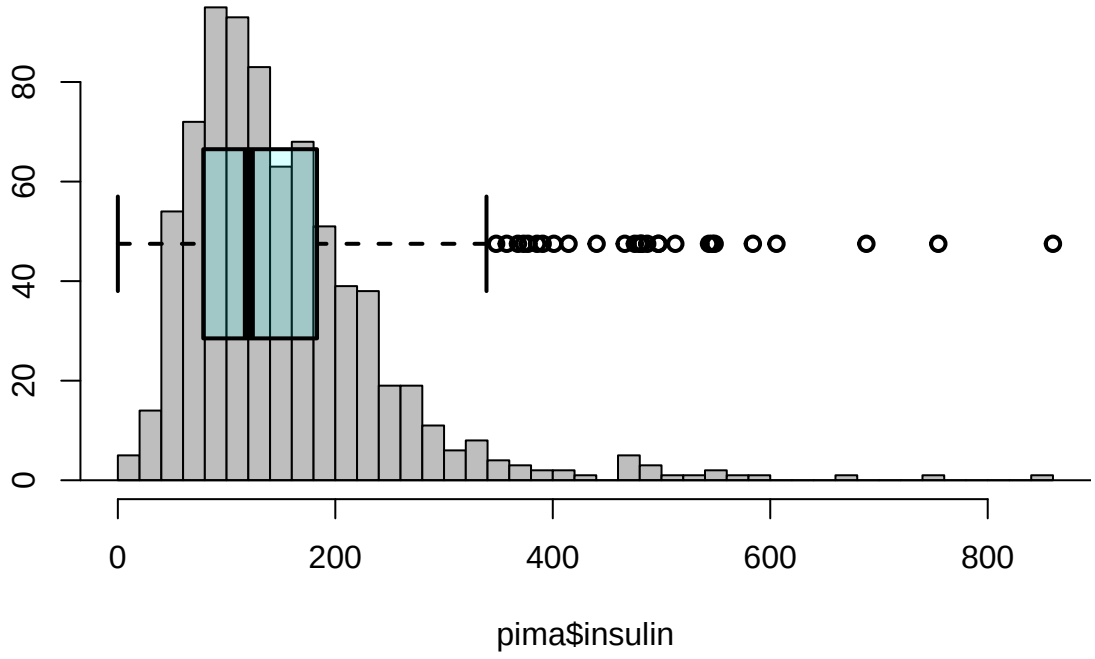

```
pima = pima[pima$triceps < 80, ]
```

Observamos como en la variable insulina hay un gran número de valores atipicos, y por ello, procedemos a eliminar esta variable pues los datos no son fiables.

```
summary(pima$insulin)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    14.0    90.0   130.6   153.2   191.2   846.0
```

```
hist(pima$insulin,  ylab = "", col = "grey", main = "", breaks=30)
lines(density(pima$insulin, na.rm = TRUE))
par(new = TRUE) # Esto indica que lo dibujaremos sobre el gráfico anterior
boxplot(pima$insulin, horizontal = TRUE, axes = FALSE, lwd = 2, col = rgb(0, 1, 1, alpha = 0.15))
```



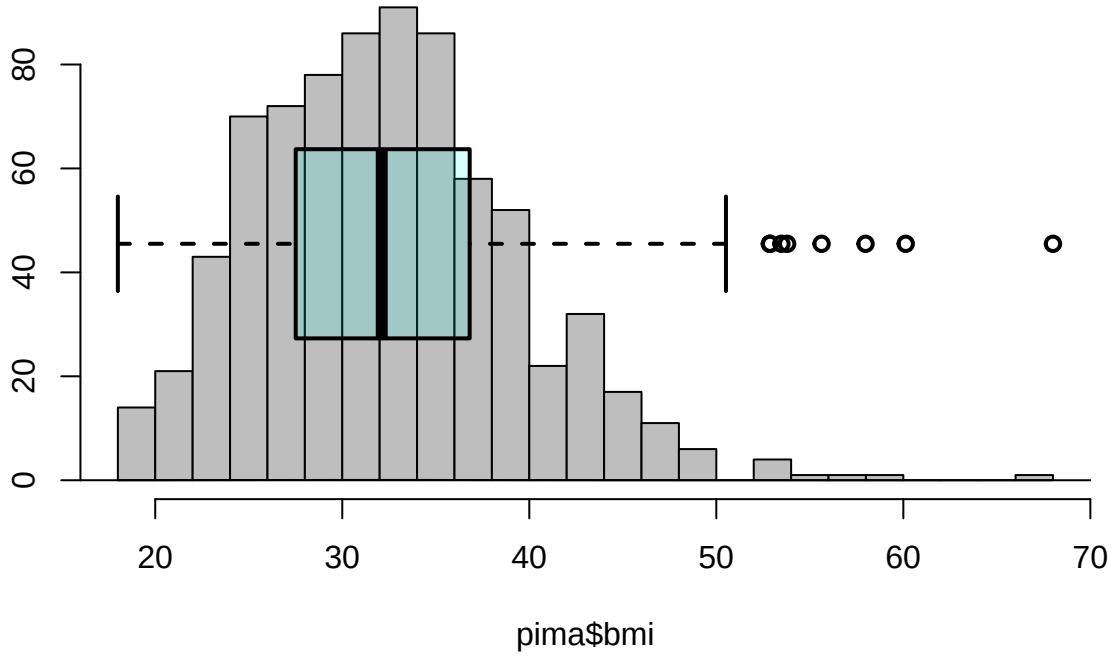

```
pima$insulin <- NULL
```

La variable bmi tiene una distribución similar a la normal con valores atipicos y al igual que antes hay uno particularmente más alejado del resto, por lo que procedemos a eliminarlo. El resto los mantenemos ya que podrian ser reales aunque se deberia consultar con un experto.

```
summary(pima$bmi)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   18.20   27.50   32.00   32.42   36.60   67.10
```

```
hist(pima$bmi, ylab = "", col = "grey", main = "", breaks = 20)
lines(density(pima$bmi, na.rm = TRUE))
par(new = TRUE) # Esto indica que lo dibujaremos sobre el gráfico anterior
boxplot(pima$bmi, horizontal = TRUE, axes = FALSE, lwd = 2, col = rgb(0, 1, 1, alpha = 0.15))
```



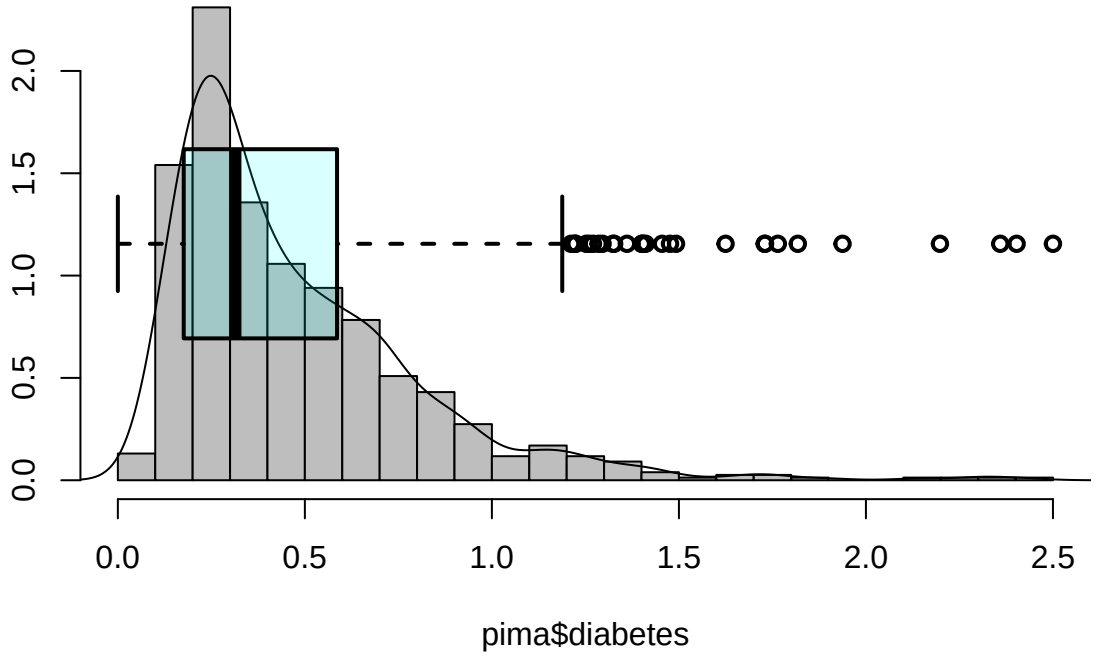

```
pima = pima[pima$bmi < 65, ]
```

La variable diabetes tiene una gran cantidad de valores atipicos, aún así, ante la falta de información en internet sobre qué representa numéricamente esta variable y la imposibilidad de consultar a un experto consideraremos como correctos aquellos valores atipicos que están menos alejados, escogemos "2" como corte en base a lo observado en la gráfica.

```
summary(pima$diabetes)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0780  0.2432  0.3725  0.4719  0.6268  2.4200
```

```
hist(pima$diabetes, probability = TRUE, ylab = "", col = "grey", main = "", breaks = 20)
lines(density(pima$diabetes)) # lines indica que se va a dibujar una línea sobre el gráfico anterior
par(new = TRUE) # Esto indica que lo dibujaremos sobre el gráfico anterior
boxplot(pima$diabetes, horizontal = TRUE, axes = FALSE, lwd = 2, col = rgb(0, 1, 1, alpha = 0.15))
```



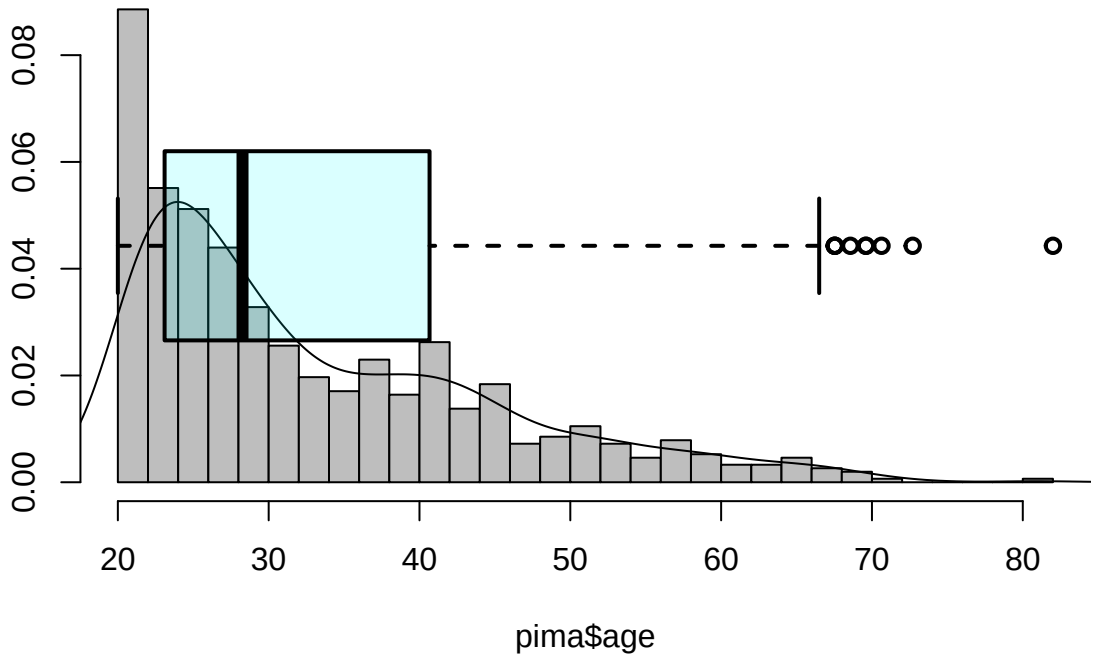

```
pima = pima[pima$diabetes < 2, ]
```

Finalmente observamos que la mayoría de personas son jóvenes. Los datos atipicos no los consideramos como erróneos puesto que sí es posible que haya gente mayor (la edad máxima es 81), pese a no ser lo común en el estudio realizado (la media se sitúa en 33 años).

```
summary(pima$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   21.00   24.00   29.00   33.24   41.00   81.00
```

```
hist(pima$age, probability = TRUE, ylab = "", col = "grey", main = "", breaks = 30)
lines(density(pima$age, na.rm = TRUE))
par(new = TRUE) # Esto indica que lo dibujaremos sobre el gráfico anterior
boxplot(pima$age, horizontal = TRUE, axes = FALSE, lwd = 2, col = rgb(0, 1, 1, alpha = 0.15))
```



Por último, estudiamos la relación lineal entre las variables, para esto eliminamos temporalmente la última columna (test) ya que tiene valores no nominales.

```
cov(pima[, 1:7])
```

```
##              pregnant   glucose    diastolic    triceps        bmi    diabetes
## pregnant  11.3795082  14.19237   8.92371340  4.7596152  0.8946531 -0.01757760
## glucose   14.1923735 914.10192  85.34855369 55.5050797 46.7352227  0.91381998
## diastolic  8.9237134  85.34855 145.32643903 25.2851343 23.1952224  0.05108424
## triceps    4.7596152  55.50508  25.28513433 80.9999074 41.1016192  0.23013218
## bmi        0.8946531  46.73522  23.19522244 41.1016192 45.0140824  0.24069645
## diabetes  -0.0175776   0.91382   0.05108424  0.2301322  0.2406965  0.09301241
## age       21.7223435  96.07282  48.47699847 16.0826558  2.8995285  0.17159048
##                  age
## pregnant   21.7223435
## glucose    96.0728159
## diastolic  48.4769985
## triceps    16.0826558
## bmi         2.8995285
## diabetes    0.1715905
## age       138.0525710
```

Estudiamos las correlaciones: hay principalmente 3 pares de variables correlacionadas positivamente. Grosor del pliegue cutáneo del tríceps con bmi, la edad con el número de embarazos, y en menor medida, la presión diastólica con la edad.

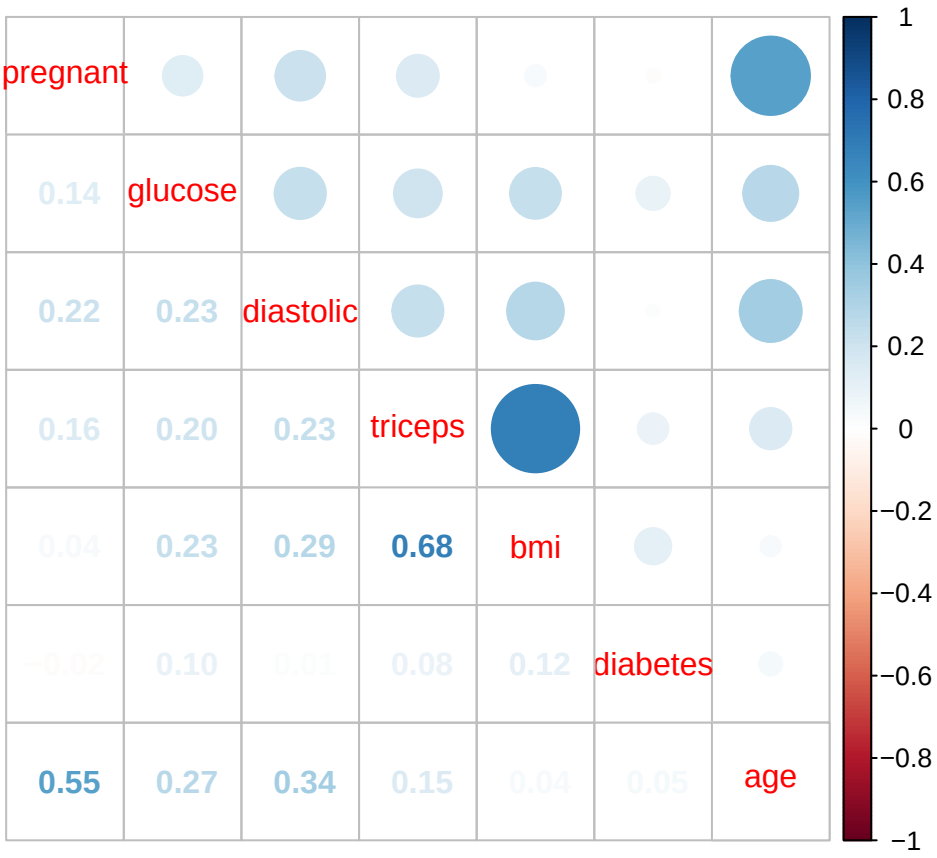
Como podemos ver las correlaciones en general son bajas, también para la variable a predecir (glucose), por lo que nuestro modelo de regresión seguramente no logre buenos resultados a la hora de predecir.

```
par(mfrow = c(1,1))
library('corrplot')
```

```
## Warning: package 'corrplot' was built under R version 4.1.3
```

```
## corrplot 0.92 loaded
```

```
corrplot.mixed(cor(pima[, 1:7], use = "complete.obs"), upper = "circle", lower='number')
```



c) Analizar en primer lugar un modelo de regresión que incluya todas las variables disponibles (describir el modelo ajustado y sus residuos, contrastar la significatividad individual de los parámetros y la calidad del modelo, verificar las hipótesis del modelo, análisis de datos influyentes y atípicos).

Creamos un modelo con todas las variables.

```
modelo <- lm(glucose ~ diabetes + pregnant + age + diastolic + triceps + bmi + test, data=pima)
summary(modelo)
```

```
##
## Call:
## lm(formula = glucose ~ diabetes + pregnant + age + diastolic +
```

```
##     triceps + bmi + test, data = pima)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -66.970 -17.892  -2.496  16.080  90.789
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  72.06686    6.72945  10.709  < 2e-16 ***
## diabetes      0.86811    3.10140   0.280  0.77962
## pregnant     -0.73475    0.33379  -2.201  0.02802 *
## age           0.44450    0.09968   4.459 9.48e-06 ***
## diastolic     0.24756    0.08573   2.888  0.00399 **
## triceps       0.04428    0.14362   0.308  0.75795
## bmi           0.25312    0.19912   1.271  0.20406
## testpositive 27.61835    2.14676  12.865  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 25.58 on 754 degrees of freedom
## Multiple R-squared:  0.2906, Adjusted R-squared:  0.284
## F-statistic: 44.12 on 7 and 754 DF,  p-value: < 2.2e-16
```

El modelo de regresión múltiple generado con todas las variables tiene un R^2 ajustado bajo (0.284), es decir, es capaz de explicar un 28.4% de la variabilidad total. La recta de regresion seria: 0.86811*diabetes* *-0.73475*pregnant + 0.44450*age* + *0.24756*diastolic + 0.04428*triceps* + *0.25312*bmi + 27.61835*testpositive + 72.06686

Como podemos observar la variable que más peso tiene es "testpositive" con su coeficiente asociado igual a 27.6183, que, curiosamene, es la unica variable en la que no pudimos estudiar su correlacion al ser una variable categorica. El resto de variables tienen pesos próximos a 0. Las variables diabetes, tríceps y bmi tienen un p-valor muy superior a 0.05, por lo que no podemos rechazar la hipótesis de que no existe relación lineal significativa. Haremos a continuación los contrastes de significatividad individual correspondientes para verificarlo.

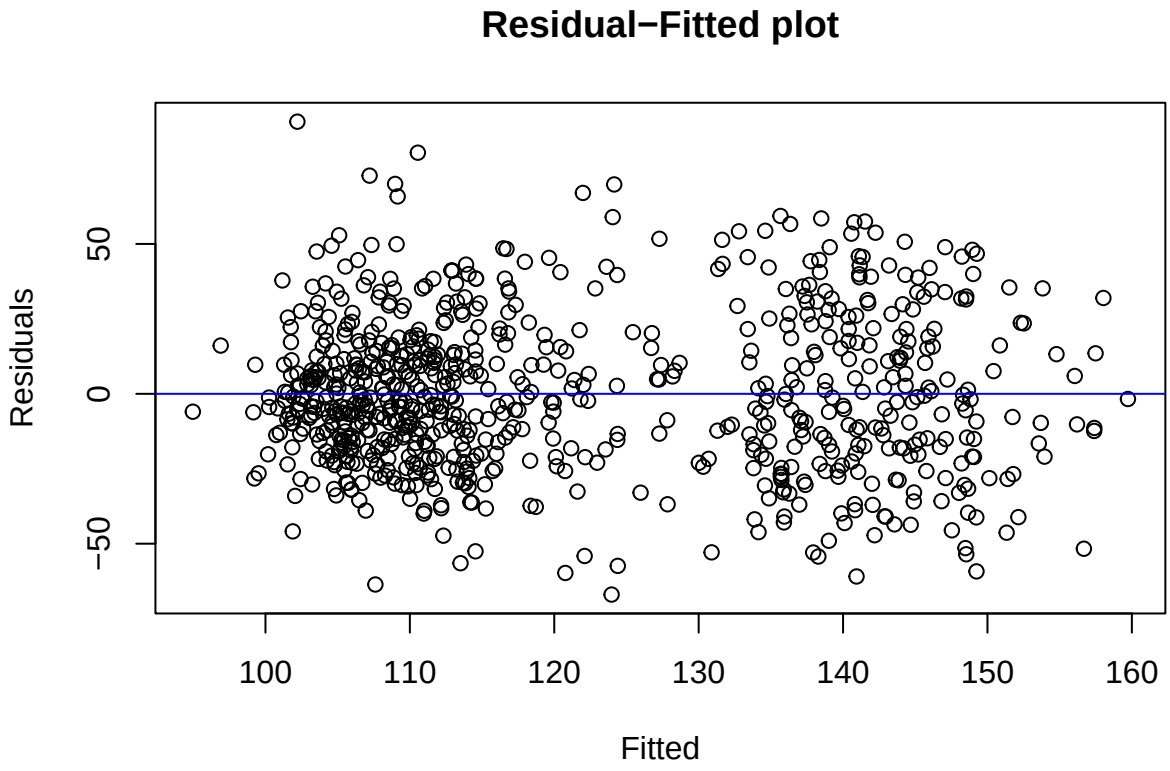
Contraste significatividad individual: Observamos que el "0" entra dentro del intervalo de confianza de las variables que acabamos de ver que tenían un p-valor alto, por tanto no superan el contraste de significatividad individual las variables diabetes, triceps y bmi.

```
confint(modelo)
```

```
##                   2.5 %      97.5 %
## (Intercept)  58.85618557 85.27754231
## diabetes     -5.22029639  6.95651298
## pregnant     -1.39002064 -0.07947956
## age           0.24880501  0.64018862
## diastolic     0.07926518  0.41585133
## triceps      -0.23767131  0.32622714
## bmi          -0.13778574  0.64402341
## testpositive 23.40400790 31.83269058
```

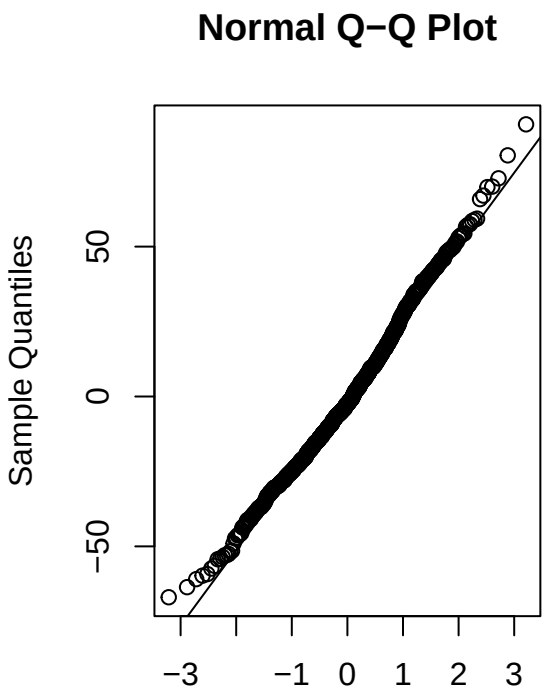
En cuanto a los residuos, vamos a mostrarlos gráficamente. Si la relación es lineal, los residuos deben distribuirse aleatoriamente en torno a 0 con una variabilidad constante a lo largo del eje X. Vemos que esto sí se cumple (hay homocedasticidad).

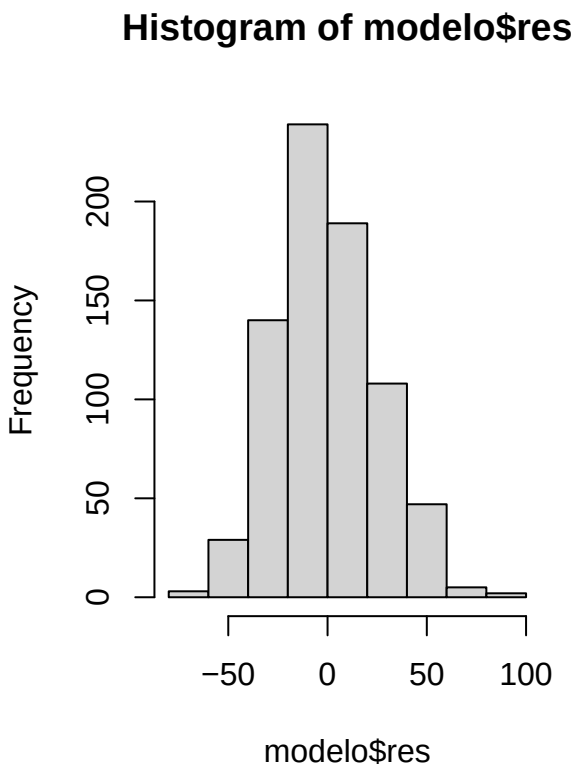
```
par(mfrow=c(1,1))
plot(modelo$fit,modelo$res,xlab="Fitted",ylab="Residuals", main="Residual-Fitted plot")
abline(h=0, col='blue')
```



Vemos que los residuos siguen una distribución normal ya que los puntos se distribuyen aproximadamente sobre la recta del QQ plot y en el histograma podemos ver como parece formar una campana de Gauss.

```
par(mfrow=c(1,2))
qqnorm(modelo$res)
qqline(modelo$res)
hist(modelo$res,10)
```





Comprobamos la independencia mediante el estadístico de Durbin-Watson.

```
library(lmtest)
```

```
## Warning: package 'lmtest' was built under R version 4.1.3
```

```
## Loading required package: zoo
```

```
## Warning: package 'zoo' was built under R version 4.1.3
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

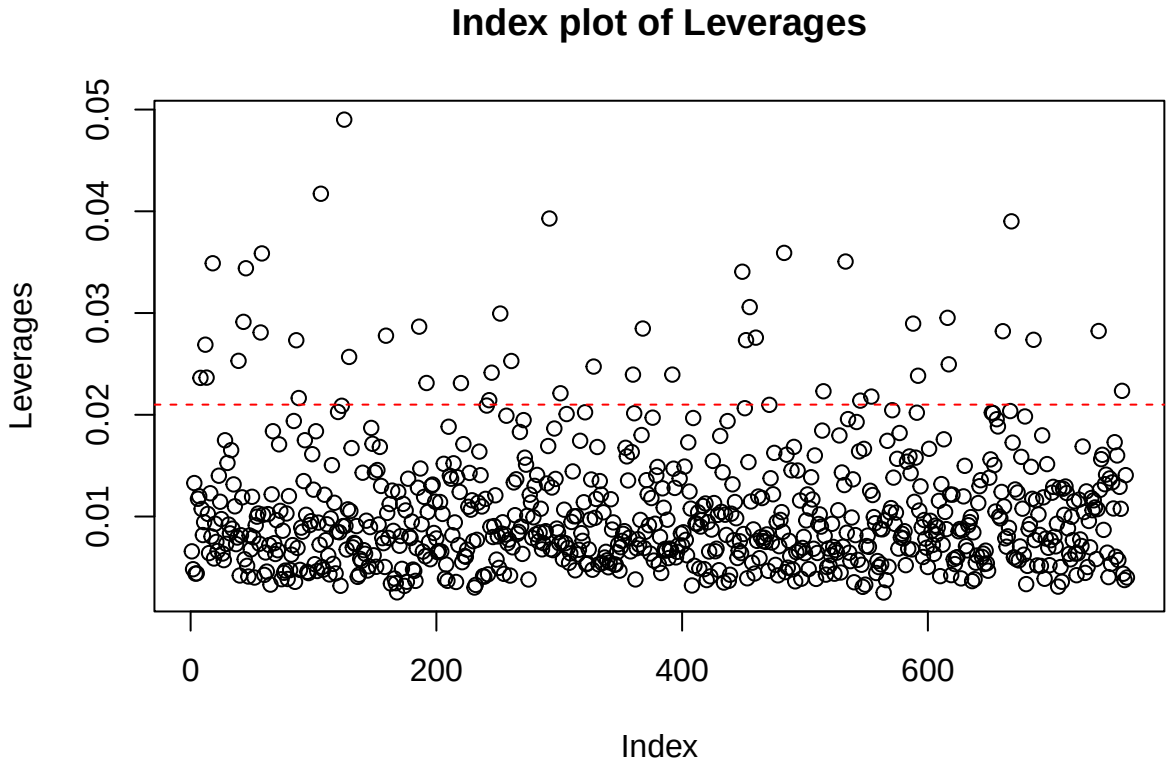
```
dwtest(modelo,alternative ="two.sided",iterations = 1000)
```

```
##
##  Durbin-Watson test
##
## data:  modelo
## DW = 1.9579, p-value = 0.5582
## alternative hypothesis: true autocorrelation is not 0
```

Como 1.5 <= DW=1.9579 <= 2.5, podemos asumir que los valores son independientes.

Estudiamos los puntos palanca. La línea horizontal marca que los valores por encima de ella son al menos dos veces el efecto medio palanca. Observamos que hay bastantes.

```
x <- model.matrix(modelo)
levevageC <- hat(x)
par(mfrow=c(1,1))
plot(levevageC,ylab="Leverages",main="Index plot of Leverages")
abline(h=2*sum(levevageC)/nrow(pima), lty=2, col="red")
```



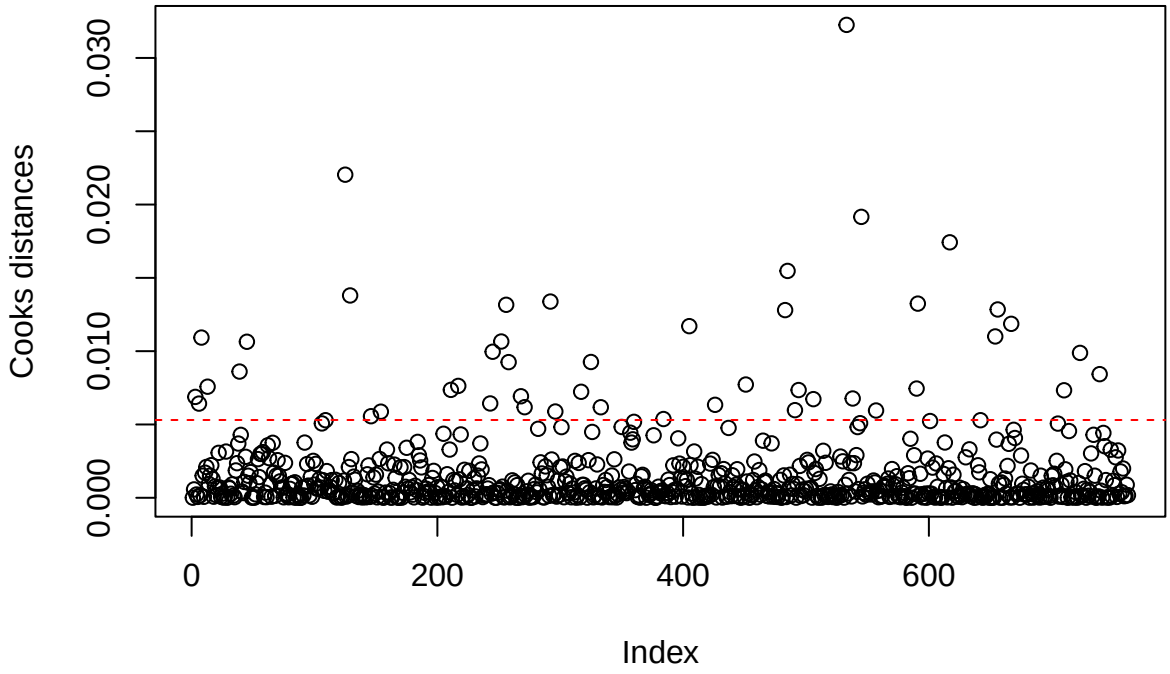
Imprimimos los valores con efecto palanca:

```
levevageC [levevageC > 2*sum(levevageC)/nrow(pima)]
```

```
##  [1] 0.02362657 0.02689051 0.02365283 0.03489092 0.02530087 0.02913213
##  [7] 0.03440206 0.02808891 0.03585921 0.02732507 0.02164610 0.04172544
## [13] 0.04900743 0.02568553 0.02777311 0.02866197 0.02312379 0.02311620
## [19] 0.02143462 0.02413682 0.02994846 0.02528573 0.03928877 0.02211410
## [25] 0.02473935 0.02394854 0.02847609 0.02394915 0.03406203 0.02733408
## [31] 0.03058509 0.02758192 0.03591122 0.02231211 0.03506264 0.02139167
## [37] 0.02178856 0.02896015 0.02383760 0.02952884 0.02495805 0.02821321
## [43] 0.03901290 0.02737005 0.02823079 0.02235373
```

Usamos la distancia de Cook para hallar puntos influyentes. Serán influyentes aquellos valores que tengan una distancia superior a 4/(n_datos-n_variables-1). Observamos que hay una gran cantidad de puntos influyentes.

```
cookC <- cooks.distance(modelo)
plot(cookC,ylab="Cooks distances")
abline (h =4/(nrow(pima)-7-1), lty = 2, col = "red")
```



Mostramos los puntos influyentes y sus distancias de Cook.

```
cookC[cookC > 4/(nrow(pima)-7-1)]
```

```
##           3           7           9          14          40          46
## 0.006879847 0.006424444 0.010926701 0.007577383 0.008608415 0.010642321
##         126         130         147         155         213         219
## 0.022037407 0.013799781 0.005563381 0.005872589 0.007359896 0.007630260
##         246         248         255         259         261         271
## 0.006433559 0.009959210 0.010657908 0.013162850 0.009252488 0.006923824
##         274         295         299         320         328         336
## 0.006177286 0.013382700 0.005883439 0.007229586 0.009260327 0.006174462
##         388         409         430         456         488         490
## 0.005370148 0.011705006 0.006340849 0.007723874 0.012799897 0.015473357
##         496         499         511         538         543         550
## 0.005976656 0.007347947 0.006727615 0.032254989 0.006770703 0.019159753
##         562         596         597         623         660         662
## 0.005951969 0.007450006 0.013241887 0.017423855 0.011003741 0.012843812
##         673         716         729         745
## 0.011864440 0.007332142 0.009880547 0.008432006
```

Mediante un influencePlot podemos analizar el efecto palanca, los puntos influyentes y los valores atípicos simultáneamente. Observamos una gran cantidad de puntos atipicos (aquellos fuera de las bandas horizontales en -2, 2), con distancias de Cook grandes, representada por el tamaño de la burbuja (punto 538); y con efecto palanca, aquellos que superan la primera línea vertical (representa el doble del efecto palanca) y la segunda (representa el triple del efecto palanca).

```
library(car)
```

```
## Warning: package 'car' was built under R version 4.1.3
```

```
## Loading required package: carData
```

```
## Warning: package 'carData' was built under R version 4.1.3
```

```
##
## Attaching package: 'car'
```

```
## The following objects are masked from 'package:faraway':
##
##     logit, vif
```

```
influencePlot(modelo, id.method="identify")
```

```
## Warning in plot.window(...): "id.method" is not a graphical parameter
```

```
## Warning in plot.xy(xy, type, ...): "id.method" is not a graphical parameter
```

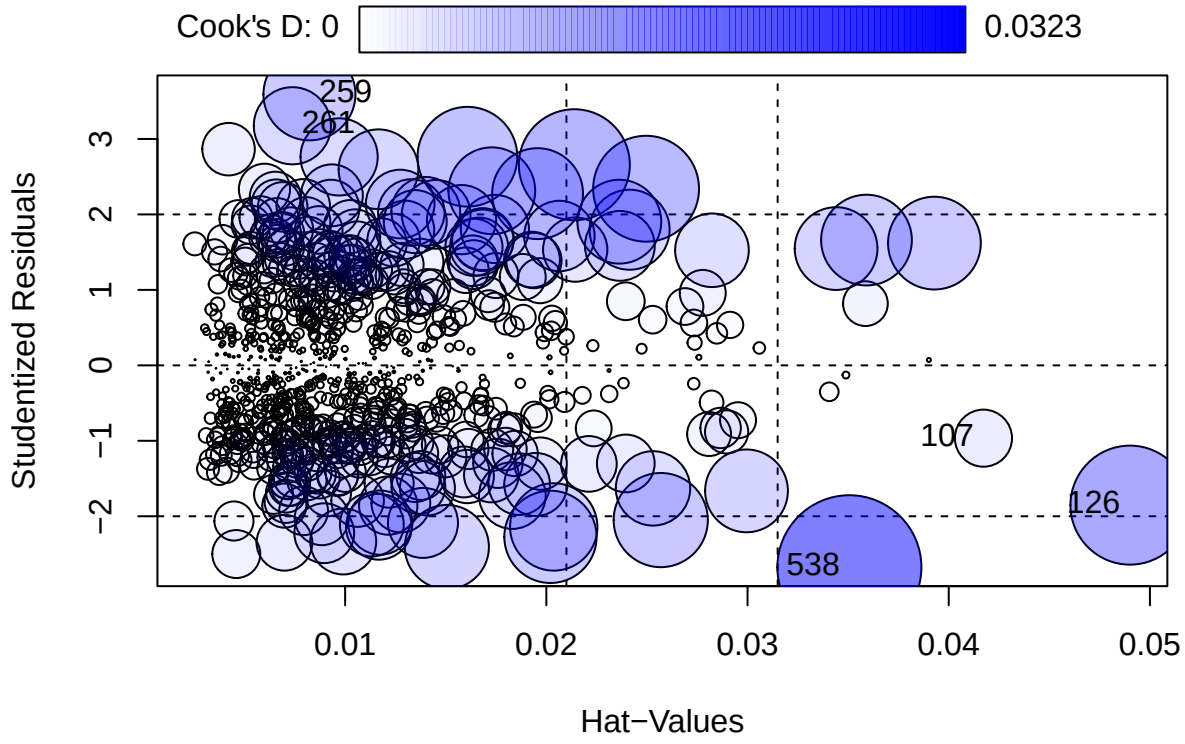
```
## Warning in axis(side = side, at = at, labels = labels, ...): "id.method" is not
## a graphical parameter
```

```
## Warning in axis(side = side, at = at, labels = labels, ...): "id.method" is not
## a graphical parameter
```

```
## Warning in box(...): "id.method" is not a graphical parameter
```

```
## Warning in title(...): "id.method" is not a graphical parameter
```

```
## Warning in plot.xy(xy.coords(x, y), type = type, ...): "id.method" is not a
## graphical parameter
```




```
##         StudRes         Hat       CookD
## 107 -0.9651675 0.041725441 0.005070676
## 126 -1.8526021 0.049007430 0.022037407
## 259  3.5914150 0.008224732 0.013162850
## 261  3.1750220 0.007376305 0.009252488
## 538 -2.6756996 0.035062638 0.032254989
```

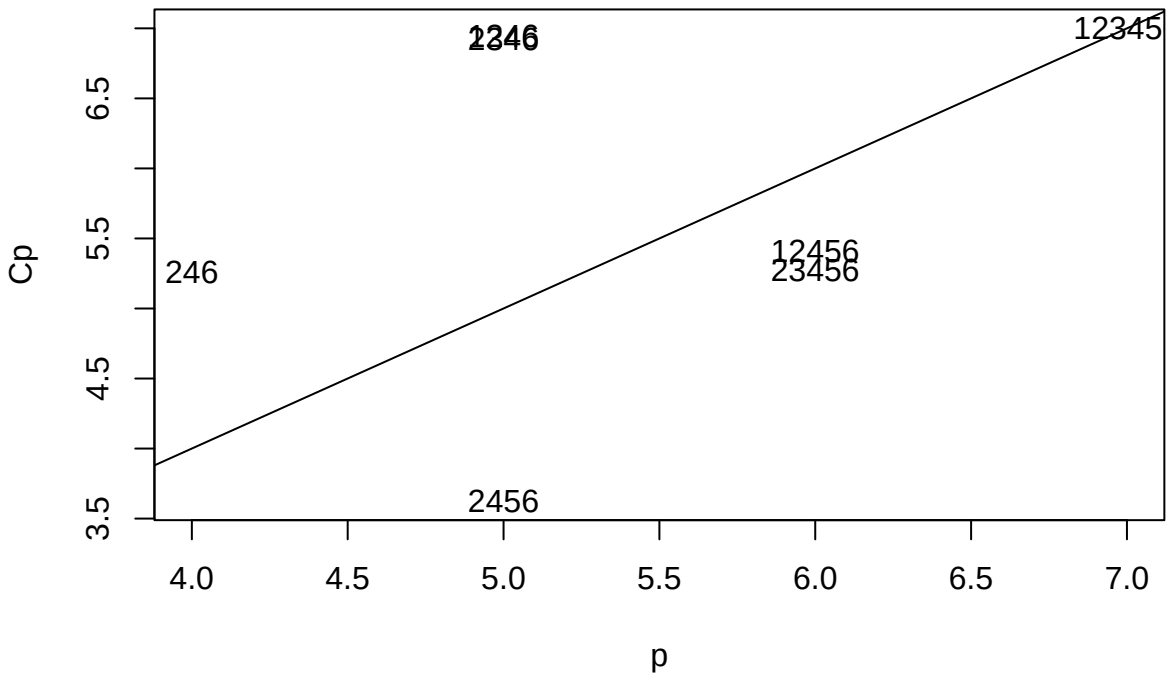
##d) Identificar el mejor modelo de regresión lineal y analizarlo.##

Usamos el estadistico Cp de Mallow vemos que el mejor modelo corresponde al 2456, que correspondería en nuestro modelo a las variables 3567.

```
y<- pima$glucose[cooks.distance(modelo) < 0.2]
Y <- y[complete.cases(y)]
x <- pima[,c(1,3,4,5,6,7)]
x <- x[cooks.distance(modelo) < 0.2,]
library(leaps)
```

```
## Warning: package 'leaps' was built under R version 4.1.3
```

```
a <- leaps(x,Y)
Cpplot(a)
```



Vemos que subconjunto de variables propone el criterio de R^2 ajustado. Este concuerda con lo observado con el estadístico Cp de Mallow proponiendo como mejor modelo el 2456. Vemos aún así que el valor de R^2 es muy bajo (0.13).

```
library(leaps)
adjr <- leaps(x,Y,method="adjr2")
maxadjr(adjr,8)
```

```
##     2,4,5,6   2,3,4,5,6   1,2,4,5,6 1,2,3,4,5,6       2,4,6     2,3,4,6
##       0.130       0.129       0.129       0.128       0.127       0.126
##     1,2,4,6   1,2,3,4,6
##       0.126       0.125
```

Buscamos el mejor modelo para las variables explicativas disponibles.

Usando el criterio AIC en combinación con backward/forward/both (no analiza todos los posibles modelos y no considera los p-valores dudosos) nos quedaremos con el modelo que menor AIC tenga pues combina los 2 criterios anteriores.

Como podemos ver los AIC son muy similares, el mejor modelo es el que usa todas las variables menos diabetes y triceps con AIC=4944.8:

```
back <- lm(glucose ~ ., pima)
sm<-step(back, direction = "both")
```

```
## Start:  AIC=4948.69
```

```
## glucose ~ pregnant + diastolic + triceps + bmi + diabetes + age +
##     test
##
##             Df Sum of Sq    RSS    AIC
## - diabetes   1        51 493560 4946.8
## - triceps    1        62 493571 4946.8
## - bmi        1      1058 494567 4948.3
## <none>                   493509 4948.7
## - pregnant   1      3171 496680 4951.6
## - diastolic  1      5458 498967 4955.1
## - age        1     13014 506523 4966.5
## - test       1    108331 601840 5097.9
##
## Step:  AIC=4946.77
## glucose ~ pregnant + diastolic + triceps + bmi + age + test
##
##             Df Sum of Sq    RSS    AIC
## - triceps    1        62 493622 4944.9
## - bmi        1      1090 494651 4946.5
## <none>                   493560 4946.8
## + diabetes   1        51 493509 4948.7
## - pregnant   1      3244 496805 4949.8
## - diastolic  1      5426 498986 4953.1
## - age        1     13146 506707 4964.8
## - test       1    111092 604652 5099.5
##
## Step:  AIC=4944.87
## glucose ~ pregnant + diastolic + bmi + age + test
##
##             Df Sum of Sq    RSS    AIC
## <none>                   493622 4944.9
## - bmi        1      2512 496134 4946.7
## + triceps    1        62 493560 4946.8
## + diabetes   1        51 493571 4946.8
## - pregnant   1      3188 496810 4947.8
## - diastolic  1      5414 499037 4951.2
## - age        1     13406 507029 4963.3
## - test       1    111361 604983 5097.9
```

Creamos el modelo con las variables pregnant, diastolic, bmi, age, test:

```
modelo1 <- lm(glucose ~ pregnant + diastolic + bmi + age + test, pima)
summary(modelo1)
```

```
##
## Call:
## lm(formula = glucose ~ pregnant + diastolic + bmi + age + test,
##     data = pima)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -66.525 -17.805  -2.614  16.123  90.721
##
```

```
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  72.24965    6.62164  10.911  < 2e-16 ***
## pregnant     -0.73147    0.33103  -2.210  0.02743 *
## diastolic     0.24637    0.08556   2.880  0.00409 **
## bmi           0.29623    0.15103   1.961  0.05020 .
## age           0.44873    0.09903   4.531 6.81e-06 ***
## testpositive 27.72140    2.12269  13.060  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 25.55 on 756 degrees of freedom
## Multiple R-squared:  0.2904, Adjusted R-squared:  0.2857
## F-statistic: 61.88 on 5 and 756 DF,  p-value: < 2.2e-16
```

Observamos que el valor de R^2 ajustado sigue siendo muy bajo e ínfimamente superior al del anterior modelo (ahora es 0.2857 y antes 0.284) aunque no nos sorprende puesto que inicialmente ya vimos como las variables no estaban apenas correlacionadas. La recta de regresión sería: -0.73147*pregnant* + *0.44873*age + 0.24637*diastolic* + *0.29623*bmi + 27.72140*testpositive + 72.24965

Los p-valores son todos menores de 0.05 salvo para bmi que es 0.05 prácticamente, para el resto podemos rechazar la hipótesis de que no hay relación lineal significativa. Haremos a continuación los contrastes de significatividad individual correspondientes para verificarlo y ver si la variable bmi es o no significativa.

Vemos que se corrobora lo anterior, todas las variables son significativas salvo bmi pues el "0" pertenece a su intervalo de confianza.

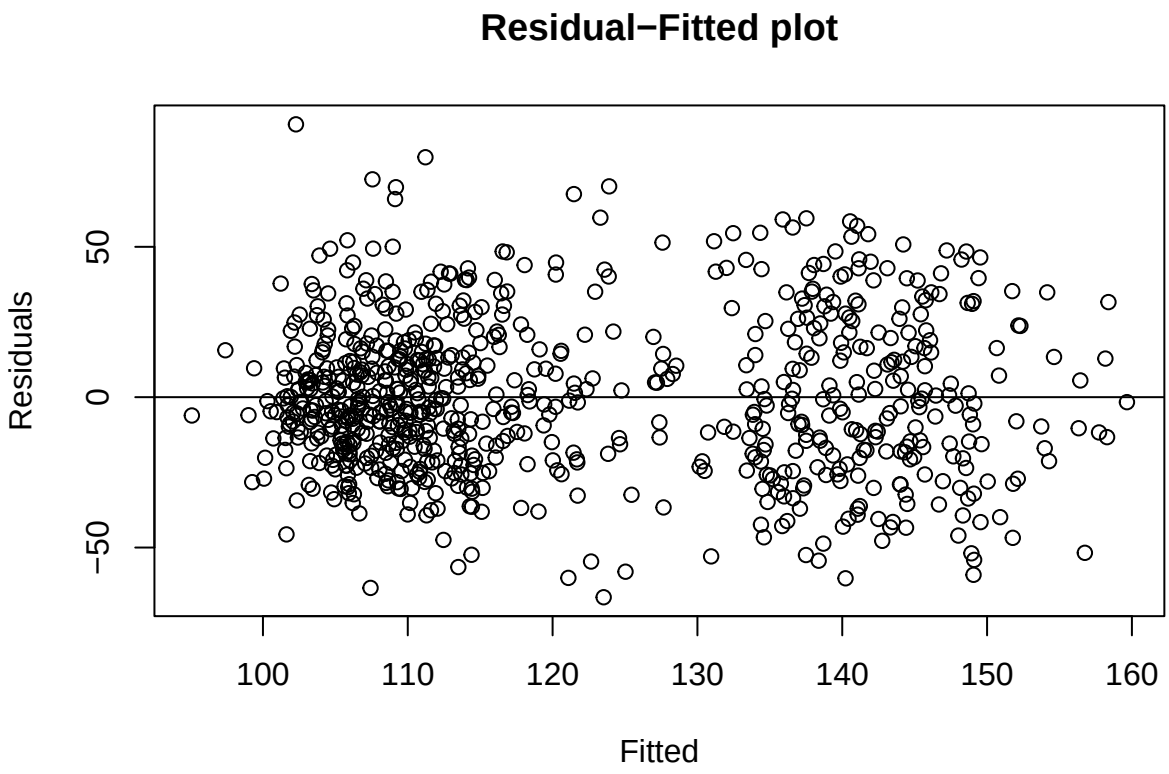
```
confint(modelo1)
```

```
##                    2.5 %      97.5 %
## (Intercept)  59.250674904 85.24862901
## pregnant     -1.381328362 -0.08161696
## diastolic     0.078412746  0.41433051
## bmi          -0.000255742  0.59271763
## age           0.254321624  0.64313338
## testpositive 23.554339758 31.88846551
```

Analizamos las hipótesis del modelo.

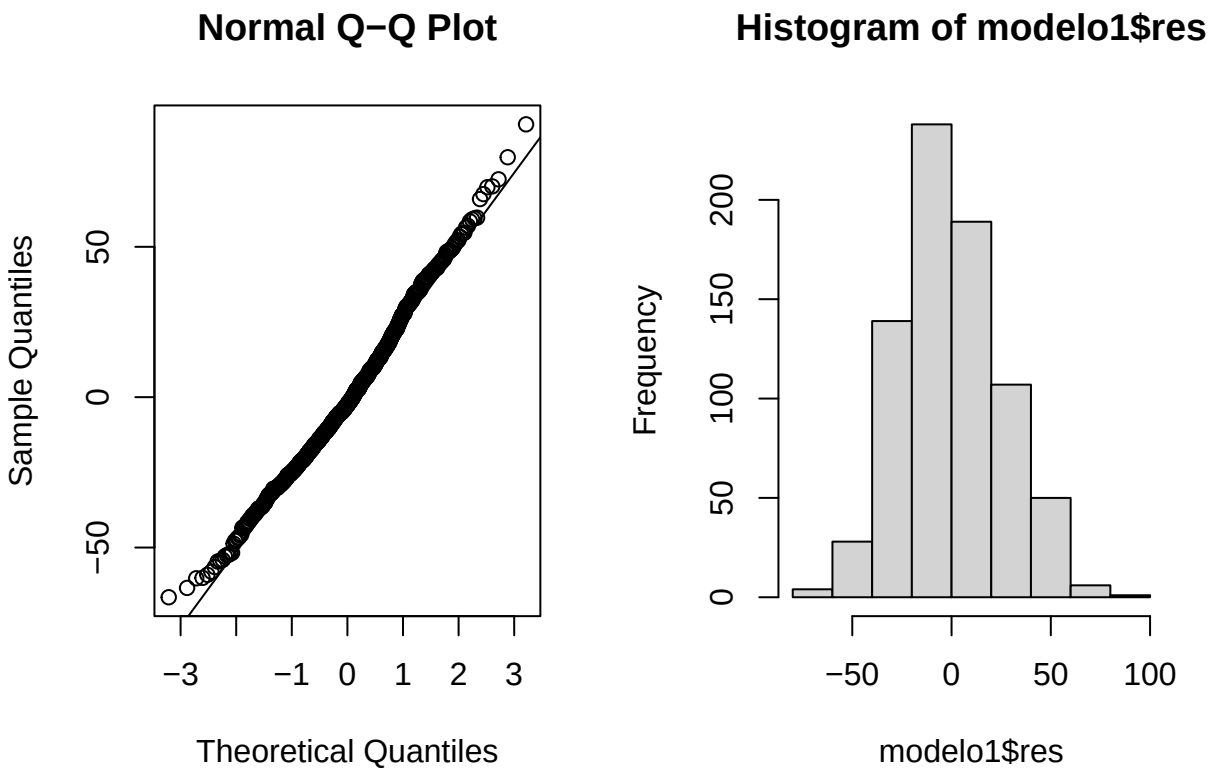
Homocedasticidad: analizamos los residuos y vemos que su varianza es constante gráficamente.

```
plot(modelo1$fit, modelo1$res,xlab="Fitted",ylab="Residuals", main="Residual-Fitted plot")
abline(h=0)
```



Vemos que los residuos siguen una distribución normal ya que los puntos se distribuyen aproximadamente sobre la recta del QQ plot y en el histograma podemos ver como parece formar una campana de Gauss.

```
par(mfrow=c(1,2))
qqnorm(modelo1$res)
qqline(modelo1$res)
hist(modelo1$res,10)
```



Vemos la independencia con el estadístico de Durbin-Watson.

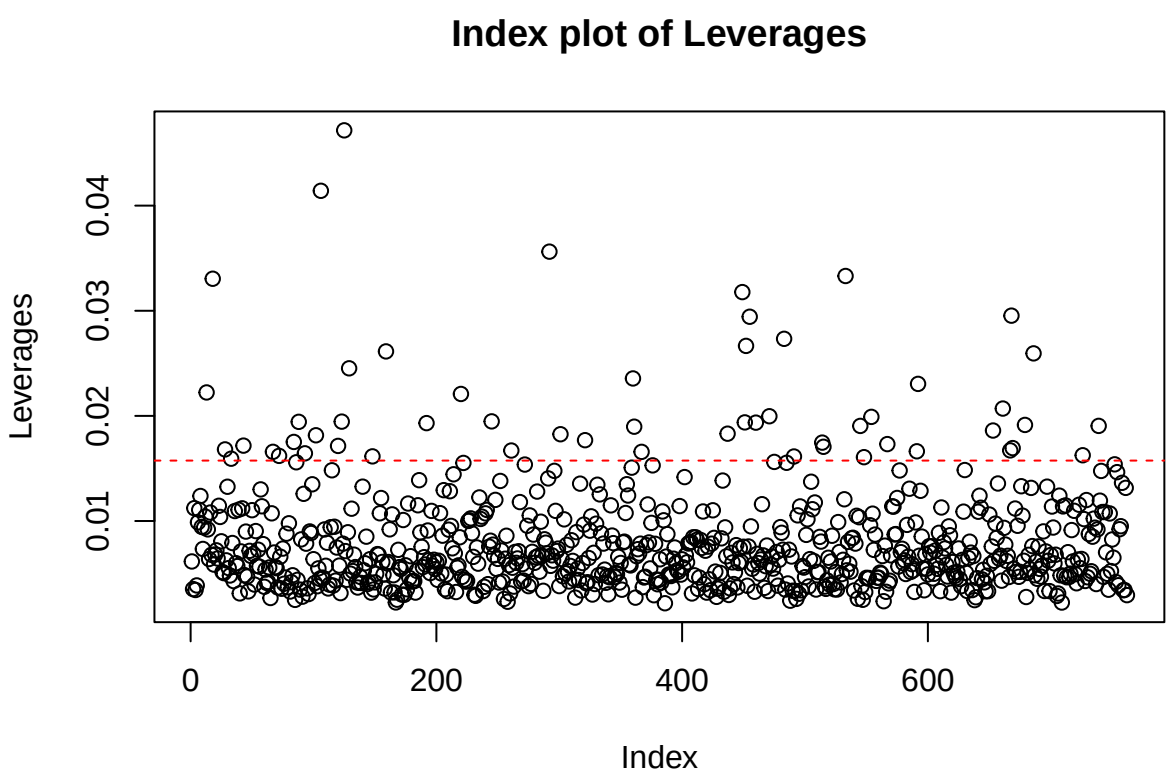
```
dwtest(modelo1,alternative ="two.sided",iterations = 1000)
```

```
##
##  Durbin-Watson test
##
## data:  modelo1
## DW = 1.9551, p-value = 0.5332
## alternative hypothesis: true autocorrelation is not 0
```

Como 1.5 <= DW = 1.9551 <= 2.5, podemos asumir que los valores son independientes.

Analizamos los puntos palanca, influyentes y atípicos.

```
x <- model.matrix(modelo1)
levevageC <- hat(x)
par(mfrow=c(1,1))
plot(levevageC,ylab="Leverages",main="Index plot of Leverages")
abline(h=2*sum(levevageC)/nrow(pima), lty=2, col="red")
```



La línea horizontal marca que los valores por encima de ella son al menos dos veces el efecto medio palanca. Observamos que hay bastantes.

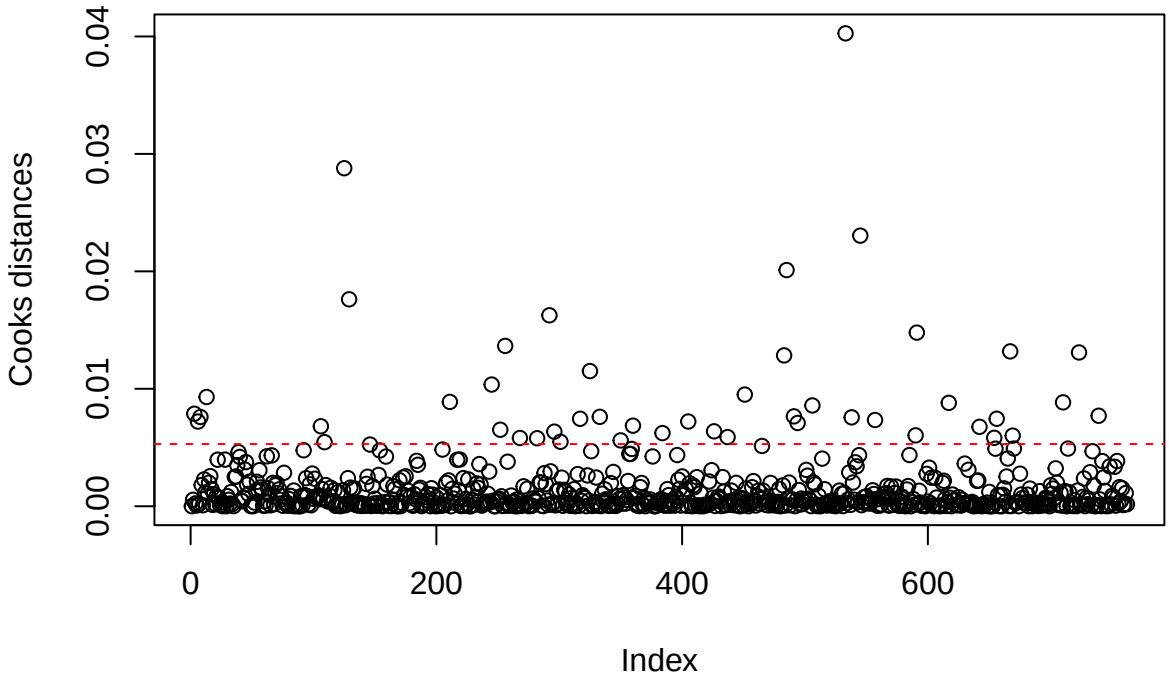
Imprimimos los valores con efecto palanca:

```
levevageC [levevageC > 2*sum(levevageC)/nrow(pima)]
```

```
##  [1] 0.02222892 0.03304474 0.01681220 0.01592827 0.01717154 0.01658165
##  [7] 0.01619787 0.01752125 0.01943438 0.01643751 0.01815501 0.04141163
## [13] 0.01714451 0.01945430 0.04716123 0.02452430 0.01615017 0.02613422
## [19] 0.01931138 0.02208403 0.01947797 0.01670399 0.03562415 0.01825470
## [25] 0.01769013 0.02356216 0.01897360 0.01657855 0.01830739 0.03177365
## [31] 0.01938095 0.02665815 0.02943048 0.01935756 0.01996598 0.02733116
## [37] 0.01613704 0.01745746 0.01706086 0.03330806 0.01904707 0.01607370
## [43] 0.01990580 0.01731575 0.01663076 0.02304345 0.01860726 0.02069865
## [49] 0.01671723 0.02953467 0.01693541 0.01913905 0.02594573 0.01625610
## [55] 0.01904894
```

Usamos la distancia de Cook para hallar puntos influyentes. Serán influyentes aquellos valores que tengan una distancia superior a 4/(n_datos-n_variables-1). Observamos que hay una gran cantidad de puntos influyentes.

```
cookC <- cooks.distance(modelo1)
plot(cookC,ylab="Cooks distances")
abline (h =4/(nrow(pima)-5-1), lty = 2, col = "red")
```



Mostramos los puntos influyentes y sus distancias de Cook.

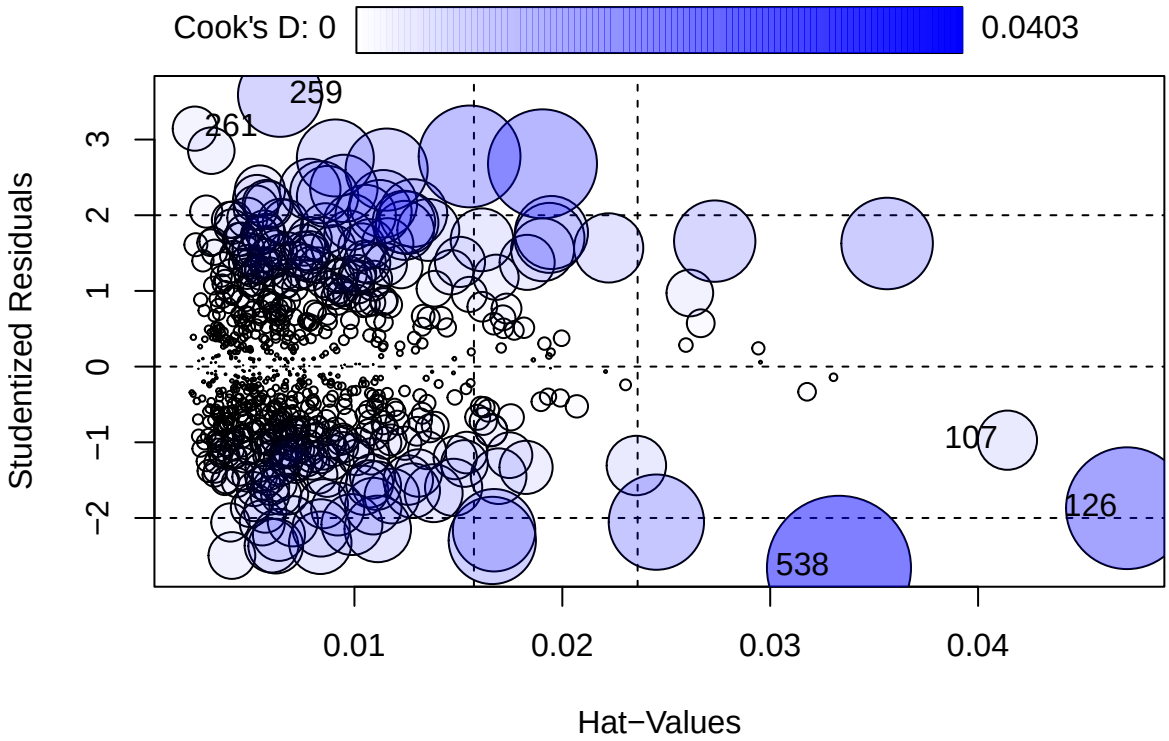
```
cookC[cookC > 4/(nrow(pima)-5-1)]
```

```
##           3           7           9          14         107         110
## 0.007875817 0.007209848 0.007597938 0.009300730 0.006807715 0.005455413
##         126         130         213         248         255         259
## 0.028781535 0.017619651 0.008884442 0.010362123 0.006512854 0.013650879
##         271         285         295         299         304         320
## 0.005811562 0.005788508 0.016257679 0.006349376 0.005488258 0.007446037
##         328         336         353         363         388         409
## 0.011509261 0.007612576 0.005606184 0.006861230 0.006227182 0.007212977
##         430         441         456         488         490         496
## 0.006379103 0.005888444 0.009507697 0.012842543 0.020109041 0.007654318
##         499         511         538         543         550         562
## 0.007087886 0.008581982 0.040264040 0.007567998 0.023040884 0.007342928
##         596         597         623         648         660         662
## 0.006036277 0.014785260 0.008801886 0.006763927 0.005825612 0.007448085
##         673         675         716         729         745
## 0.013184698 0.006011509 0.008844446 0.013093099 0.007705945
```

Mediante un influencePlot podemos analizar el efecto palanca, los puntos influyentes y los valores atípicos simultáneamente. Observamos una gran cantidad de puntos atipicos (aquellos fuera de las bandas horizontales en -2, 2), con distancias de Cook grandes, representada por el tamaño de la burbuja (punto 538); y con efecto palanca, aquellos que superan la primera línea vertical (representa el doble del efecto palanca) y la segunda (representa el triple del efecto palanca). El punto con mayor efecto palanca sería el 295.

```
library(car)
influencePlot(modelo1, id.method="identify")
```

```
## Warning in plot.window(...): "id.method" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "id.method" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "id.method" is not
## a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "id.method" is not
## a graphical parameter

## Warning in box(...): "id.method" is not a graphical parameter

## Warning in title(...): "id.method" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "id.method" is not a
## graphical parameter
```




```
##         StudRes         Hat       CookD
## 107 -0.9723341 0.041411632 0.006807715
## 126 -1.8709698 0.047161235 0.028781535
```

```
## 259  3.5896995 0.006414691 0.013650879
## 261  3.1441725 0.002321872 0.003789958
## 538 -2.6585178 0.033308059 0.040264040
```

##e) Analizar el modelo de regresión lineal tras la eliminación de valores influentes y atípicos.##

Hacemos un nuevo modelo eliminando los valores influentes y atípicos:

```
modelo2 <- lm(glucose ~ pregnant + diastolic + bmi + age + test, pima, subset=((cookC <4/(nrow(pima)-5-1
summary(modelo2)
```

```
##
## Call:
## lm(formula = glucose ~ pregnant + diastolic + bmi + age + test,
##     data = pima, subset = ((cookC < 4/(nrow(pima) - 5 - 1)) &
##         sort(abs((rstudent(modelo) < qt(0.975, 761))))))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -61.963 -16.226  -2.318  14.730  80.627
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  67.65676    6.37167  10.618  < 2e-16 ***
## pregnant     -0.77276    0.32721  -2.362  0.01847 *
## diastolic     0.27747    0.08577   3.235  0.00127 **
## bmi           0.35497    0.14563   2.438  0.01504 *
## age           0.44701    0.10009   4.466 9.31e-06 ***
## testpositive 28.73669    2.02601  14.184  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22.76 on 692 degrees of freedom
## Multiple R-squared:  0.3569, Adjusted R-squared:  0.3523
## F-statistic: 76.82 on 5 and 692 DF,  p-value: < 2.2e-16
```

Vemos que en este modelo la variabilidad total explicada (R^2 ajustado) aumenta considerablemente a un 34.92%. La recta de regresión sería: -0.67814*pregnant* + *0.40397*age + 0.24152*diastolic* + *0.31952*bmi + 29.17013*testpositive + 71.98805.

Los p-valores son inferiores todos a 0.05, por lo que se podría rechazar que no existe relación lineal significativa.

Hacemos los contrastes de significatividad individual y observamos como todos lo pasan al no tener ninguno incluido el "0" en su intervalo de confianza.

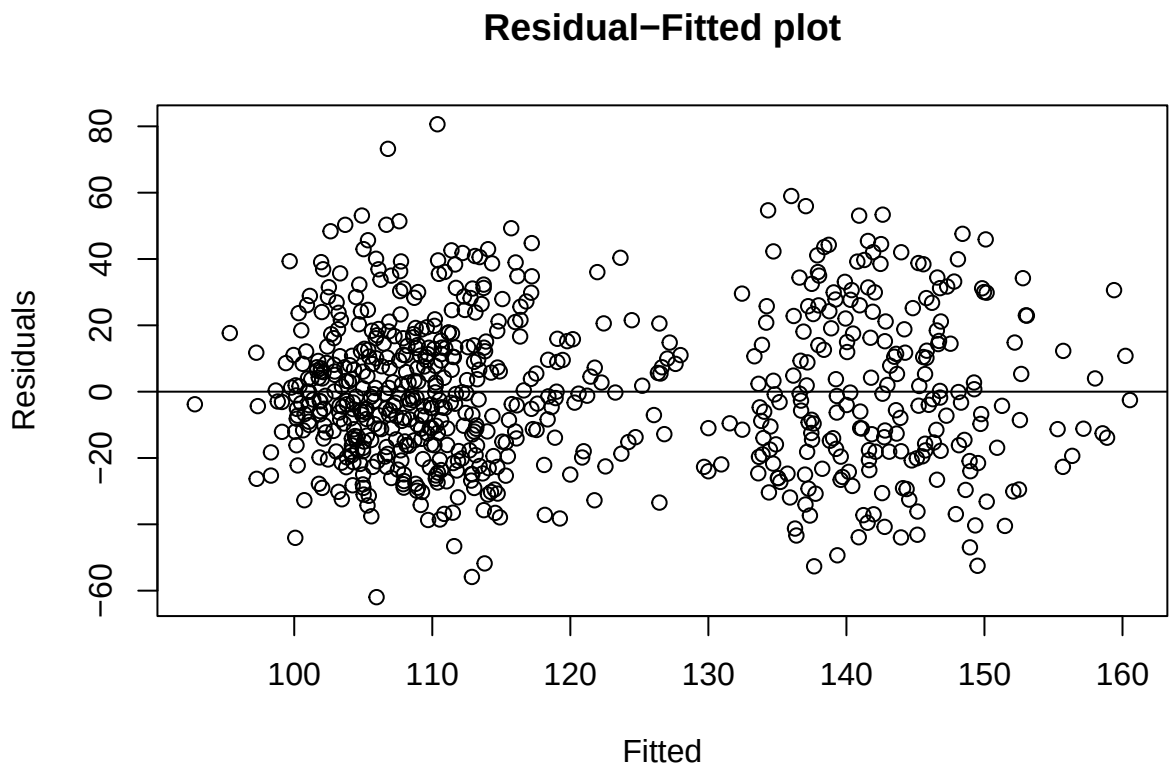
```
confint(modelo2)
```

```
##                    2.5 %     97.5 %
## (Intercept)  55.14663180 80.1668977
## pregnant     -1.41519323 -0.1303209
## diastolic     0.10907050  0.4458673
## bmi           0.06904573  0.6409006
## age           0.25049169  0.6435335
## testpositive 24.75882710 32.7145459
```

Analizamos las hipótesis del modelo.

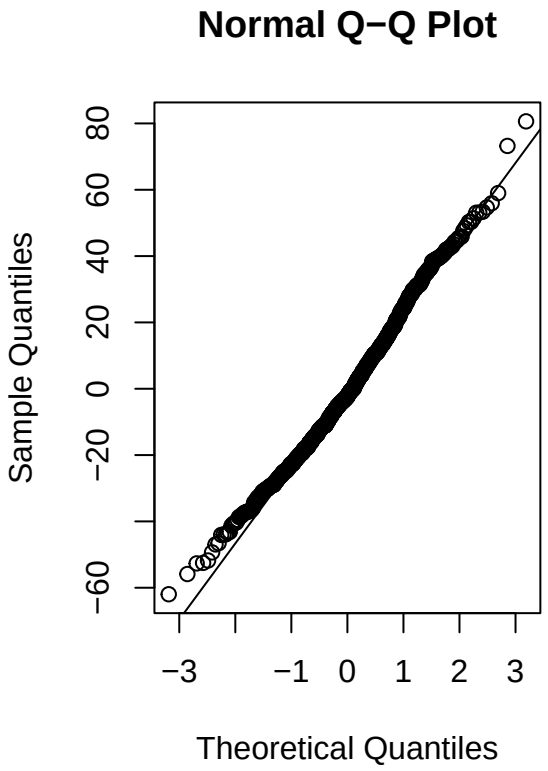
Homocedasticidad: analizamos los residuos y vemos que su varianza es constante gráficamente.

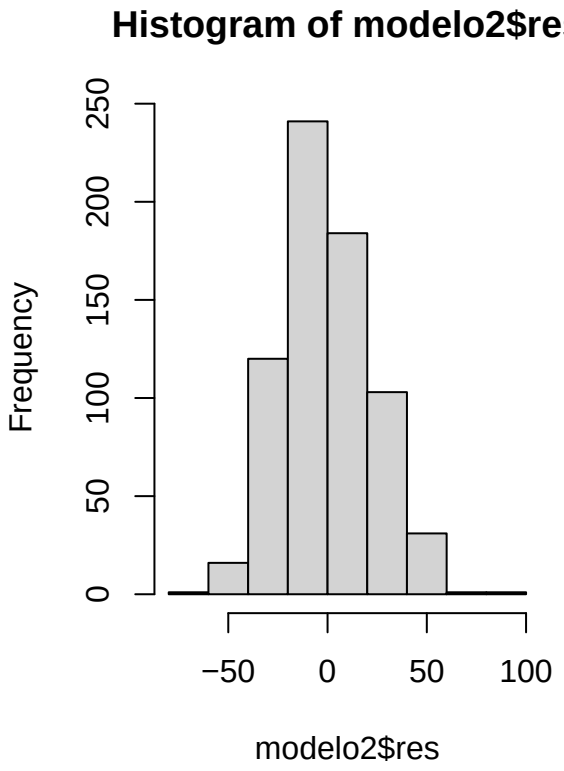
```
plot(modelo2$fit, modelo2$res,xlab="Fitted",ylab="Residuals", main="Residual-Fitted plot")
abline(h=0)
```



Vemos que los residuos siguen una distribución normal ya que los puntos se distribuyen aproximadamente sobre la recta del QQ plot y en el histograma podemos ver cómo parece formar una campana de Gauss.

```
par(mfrow=c(1,2))
qqnorm(modelo2$res)
qqline(modelo2$res)
hist(modelo2$res,10)
```





Vemos la independencia con el estadístico de Durbin-Watson.

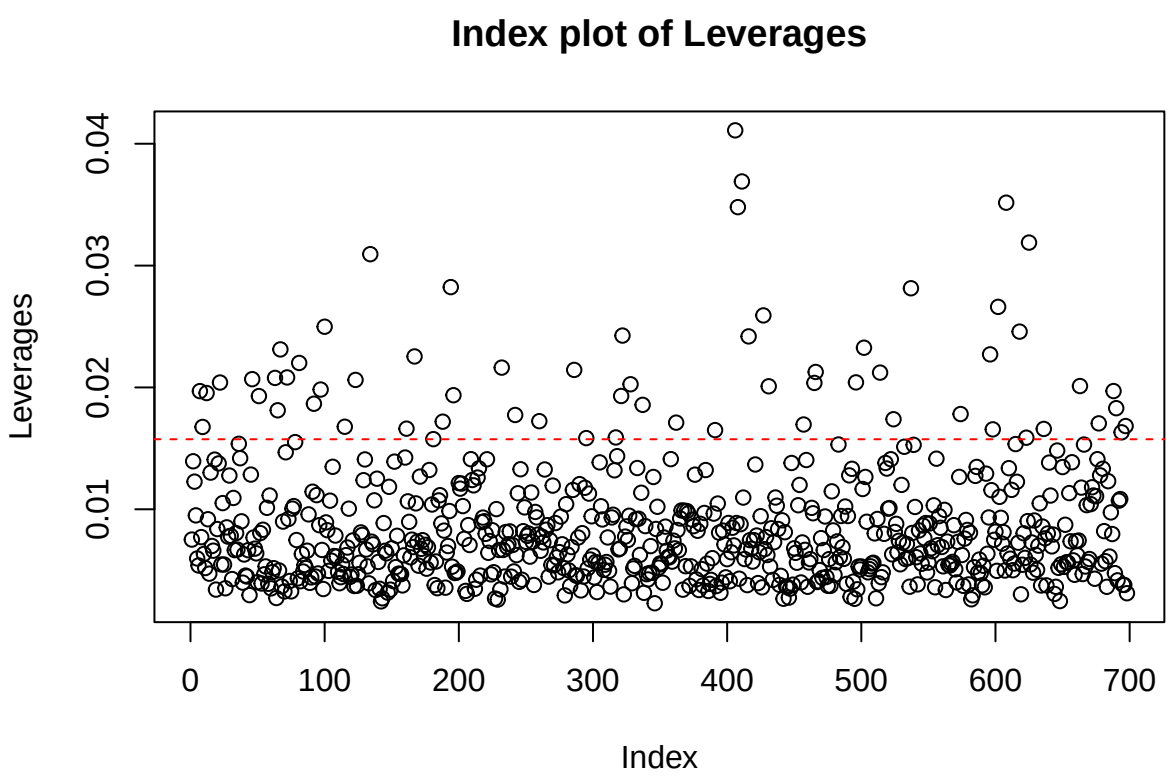
```
dwtest(modelo2,alternative ="two.sided",iterations = 1000)
```

```
##
##  Durbin-Watson test
##
## data:  modelo2
## DW = 1.9264, p-value = 0.3286
## alternative hypothesis: true autocorrelation is not 0
```

Como 1.5 <= DW = 1.9551 <= 2.5, podemos asumir que los valores son independientes.

Analizamos los puntos palanca, influyentes y atípicos.

```
x <- model.matrix(modelo2)
levevageC <- hat(x)
par(mfrow=c(1,1))
plot(levevageC,ylab="Leverages",main="Index plot of Leverages")
abline(h=2*sum(levevageC)/nrow(pima), lty=2, col="red")
```



La línea horizontal marca que los valores por encima de ella son al menos dos veces el efecto medio palanca. Observamos que hay bastantes.

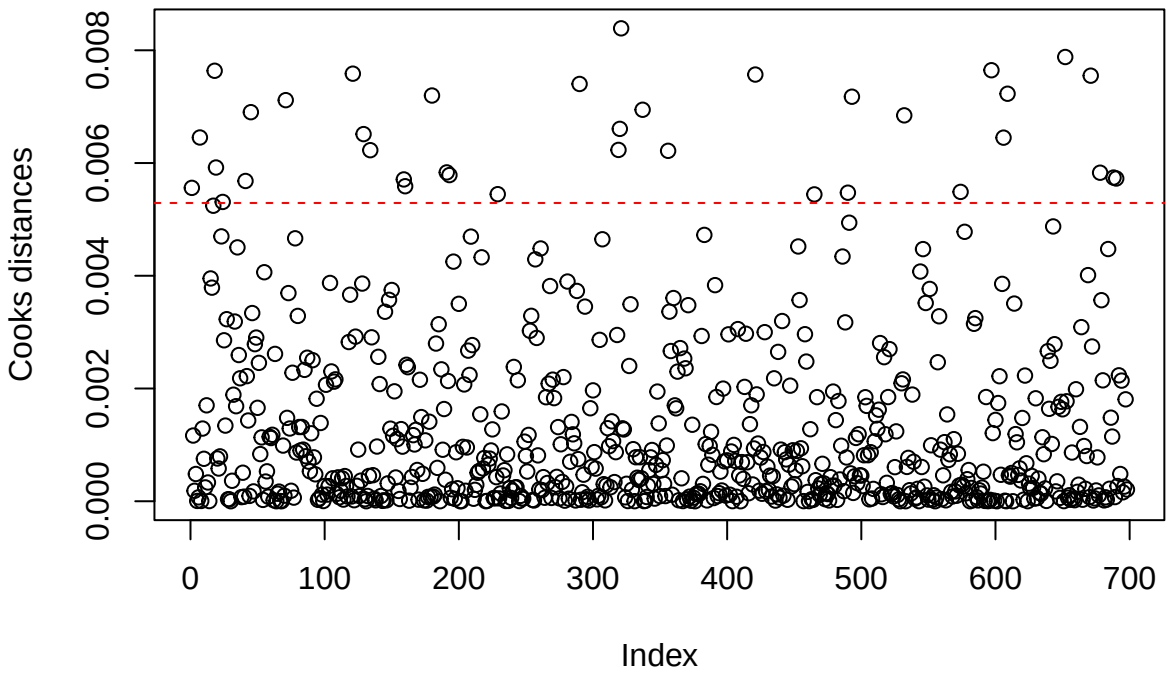
Imprimimos los valores con efecto palanca:

```
levevageC [levevageC > 2*sum(levevageC)/nrow(pima)]
```

```
##  [1] 0.01969816 0.01675689 0.01954078 0.02040213 0.02067907 0.01929155
##  [7] 0.02078013 0.01812314 0.02312086 0.02081990 0.02200984 0.01866145
## [13] 0.01982651 0.02498238 0.01677336 0.02061789 0.03093491 0.01661724
## [19] 0.02254299 0.01575561 0.01718863 0.02823341 0.01936114 0.02162606
## [25] 0.01773794 0.01723042 0.02143734 0.01584059 0.01589380 0.01929787
## [31] 0.02425747 0.02025972 0.01857137 0.01710989 0.01649729 0.04109888
## [37] 0.03479558 0.03690349 0.02418806 0.02591014 0.02009364 0.01695779
## [43] 0.02036910 0.02126981 0.02041246 0.02325924 0.02121617 0.01738800
## [49] 0.02813923 0.01781557 0.02271297 0.01655973 0.02661226 0.03516282
## [55] 0.02458126 0.01587511 0.03188897 0.01659440 0.02011378 0.01704854
## [61] 0.01970130 0.01829674 0.01631704 0.01683074
```

Usamos la distancia de Cook para hallar puntos influyentes. Serán influyentes aquellos valores que tengan una distancia superior a 4/(n_datos-n_variables-1). Observamos que hay una gran cantidad de puntos influyentes.

```
cookC <- cooks.distance(modelo2)
plot(cookC,ylab="Cooks distances")
abline (h =4/(nrow(pima)-5-1), lty = 2, col = "red")
```



Mostramos los puntos influyentes y sus distancias de Cook.

```
cookC[cookC > 4/(nrow(pima)-5-1)]
```

```
##          23          29          40          41          46          63
## 0.005559425 0.006452976 0.007636799 0.005920469 0.005309666 0.005681532
##          67          93         147         155         160         186
## 0.006902178 0.007115447 0.007585203 0.006513033 0.006227907 0.005708733
##         187         207         219         221         261         329
## 0.005588322 0.007197502 0.005833658 0.005786243 0.005447306 0.007402354
##         360         361         362         380         400         470
## 0.006232662 0.006606021 0.008389261 0.006944875 0.006216719 0.007569354
##         519         546         549         591         636         661
## 0.005443483 0.005473903 0.007176464 0.006845728 0.005487911 0.007645989
##         671         676         720         740         748         758
## 0.006450256 0.007229260 0.007880864 0.007549897 0.005828117 0.005742231
##         760
## 0.005725352
```

EJERCICIO 2

Ahora vamos a realizar el análisis de componentes principales. Para ello importamos las librerías necesarias.

```
library(ISLR)
```

```
## Warning: package 'ISLR' was built under R version 4.1.3
```

```
library(factoextra)
```

```
## Warning: package 'factoextra' was built under R version 4.1.3
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 4.1.3
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

El análisis de componentes principales es una técnica de reducción de la dimensionalidad que consiste en transformar nuestras variables en otro conjunto llamado componentes principales que son combinación lineal de las variables iniciales. Las técnicas de reducción de dimensionalidad son muy útiles en el caso de que tengamos un gran número de variables. En nuestro caso, vamos a utilizarlas aunque nuestro número de variables no sea demasiado grande y comparar los modelos que resultan. Cabe destacar que nunca vamos a obtener un modelo mejor que el que tenemos con todas las variables ya que, al quedarnos con sólo algunas componentes, estamos perdiendo siempre información.

Para utilizar esta técnica, es importante cumplir las siguientes dos condiciones:

- Las variables son numéricas.
- Las variables están correlacionadas.

Para cumplir la primera condición, simplemente vamos a retirar de nuestro conjunto de variables la variable test, que es cualitativa.

```
variables <- pima[,c(1,3,4,5,6,7)]
```

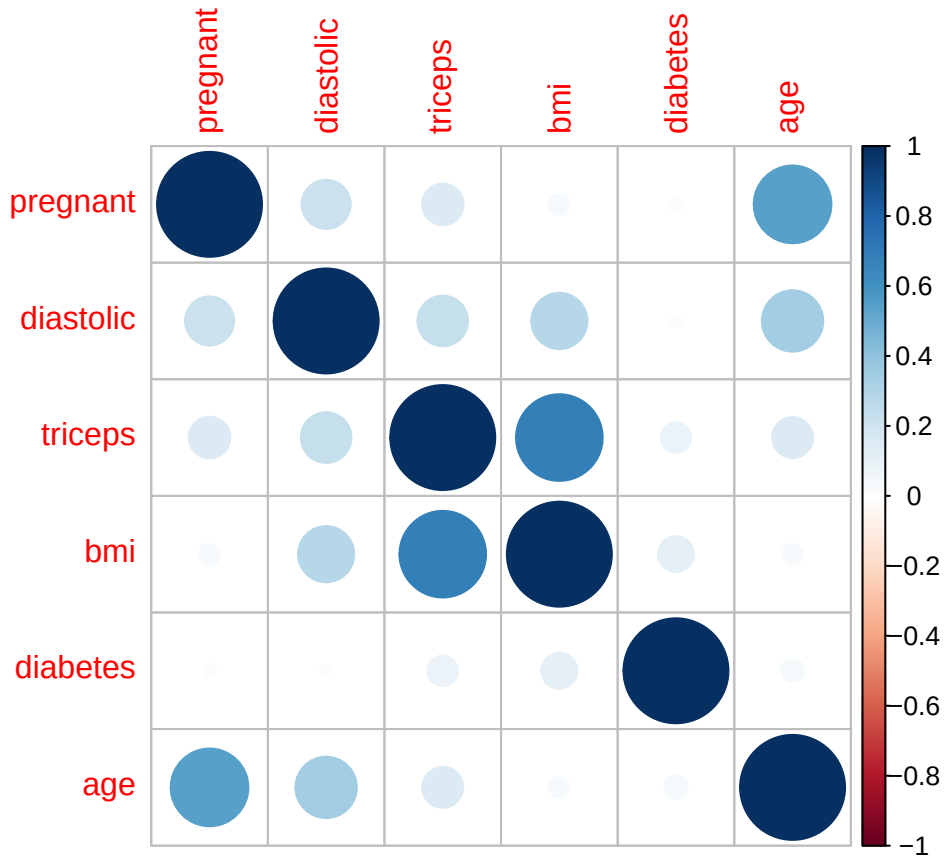
Para la segunda condición, vamos a imprimir los valores de correlación entre nuestras variables. Para ello utilizamos la matriz de correlación de Pearson.

```
Cor<-cor(variables,method="pearson",use="pairwise.complete.obs")
Cor
```

```
##              pregnant  diastolic    triceps        bmi    diabetes        age
## pregnant   1.00000000 0.21943786 0.15677176 0.03952926 -0.01708549 0.54805315
## diastolic  0.21943786 1.00000000 0.23305085 0.28678209  0.01389454 0.34224819
## triceps    0.15677176 0.23305085 1.00000000 0.68067918  0.08384258 0.15208741
## bmi        0.03952926 0.28678209 0.68067918 1.00000000  0.11763188 0.03678163
## diabetes  -0.01708549 0.01389454 0.08384258 0.11763188  1.00000000 0.04788511
## age        0.54805315 0.34224819 0.15208741 0.03678163  0.04788511 1.00000000
```

En esta matriz vemos cómo no encontramos casi ningún valor próximo a 1, lo que indicaría una correlación perfecta entre variables. Aún así, vamos a mostrar un gráfico para ver estas correlaciones de manera más visual.

```
library('corrplot')
corrplot(Cor,method="circle")
```



Como podíamos ver anteriormente, el gráfico nos muestra que no hay dos variables que estén muy correlacionadas entre sí. Sí que podemos ver que hay algunas más similares a otras como el bmi y la medida del grosor del pliegue cutáneo en el tríceps, lo cual tiene sentido (una persona con mayor bmi tendrá más sobrepeso por lo que el grosor de su pliegue cutáneo será mayor) o entre el número de embarazos y la edad.

Una vez vistos estos resultados, vemos como esa segunda condición que necesitamos no se da en nuestros datos o se da de manera muy leve así que nuestro análisis no va a ser bueno.

Aún así vamos a comprobarlo a continuación.

Esta es la función que nos genera el objeto PCA. Con esta función hacemos todo el análisis y sacamos nuestras componentes principales. Ponemos scale = True para que nuestras variables estén escaladas y no influyan más algunas de ellas por tener un rango de valores mayor. .

```
pca_pima<-prcomp(variables,scale=TRUE)
```

Vamos a ver alguna de las características principales del objeto que nos devuelve.

```
pca_pima$rotation
```

```
##                 PC1        PC2         PC3         PC4         PC5         PC6
## pregnant  -0.3845910  0.5137146 -0.02903367 -0.41703900  0.64229235 -0.02971727
## diastolic -0.4385160  0.1060106  0.12421232  0.84374803  0.21321336  0.15383794
## triceps   -0.5043988 -0.4046392  0.12060607 -0.31412613 -0.14595179  0.66882429
## bmi       -0.4671072 -0.5117475  0.10735551 -0.07826305  0.05094033 -0.70688050
## diabetes  -0.1065416 -0.1733887 -0.97106477  0.07960083  0.08478997  0.04574731
## age       -0.4193269  0.5188226 -0.12120448 -0.05505550 -0.71478278 -0.16233293
```

Con la columna rotation de nuestro objeto vemos el peso que tiene cada una de nuestras variables en cada una de las componentes principales. Cuanto mayor sea el valor absoluto de un coeficiente, mayor contribución tendrá esa variable en dicha componente. Por ejemplo, vemos como en la componente 4, tenemos la mayoría de la información proveniente de las variables insulin y diabetes.

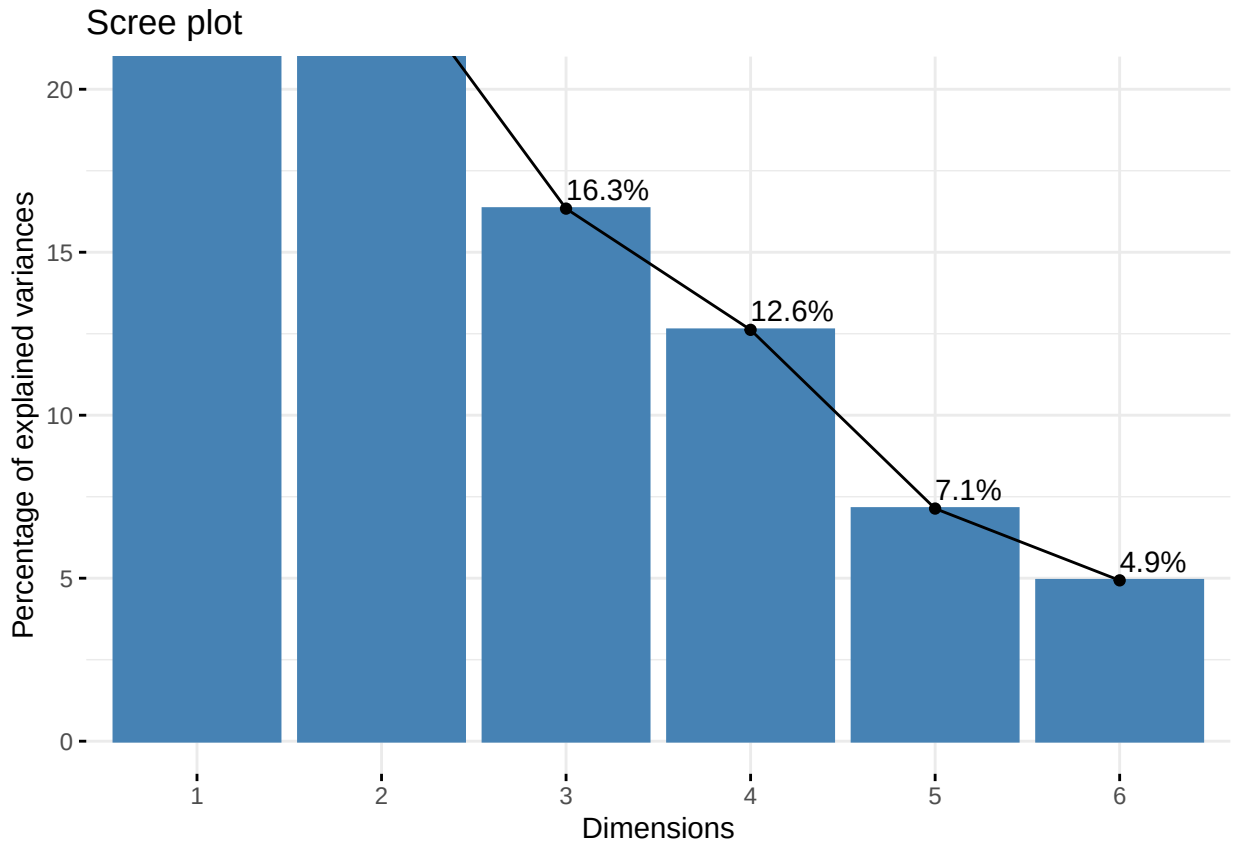
```
summary(pca_pima)
```

```
## Importance of components:
##                           PC1    PC2    PC3    PC4     PC5     PC6
## Standard deviation     1.4480 1.2007 0.9901 0.8701 0.65436 0.54410
## Proportion of Variance 0.3494 0.2403 0.1634 0.1262 0.07137 0.04934
## Cumulative Proportion  0.3494 0.5897 0.7531 0.8793 0.95066 1.00000
```

Utilizando la función summary podemos ver la información principal como el porcentaje de la varianza explicada. Este valor indica qué porcentaje de la variabilidad de los datos explica cada una de nuestras variables. La suma de todas ellas resulta 1, es decir, el 100%.

Podemos observar que no tenemos una PC (componente príncipal) que nos explique la mayor parte de los datos si no que esa ganacia en la varianza explicada se da poco a poco. Esto corrobora que nuestro análisis de componentes principales no va a ser bueno otra vez ya que necesitamos más de 5 componentes para explicar el 90% de la variabilidad.

```
fviz_eig(pca_pima, addlabels = TRUE, ylim = c(0, 20))
```



El scree plot o gráfico de sedimentos nos muestra de manera gráfica esos datos de variabilidad explicada. Con este gráfico se ve mucho más claro cómo no vemos un gran porcentaje en ninguna de las barras de

nuestro histograma y además no hay un cambio significativo entre la variabilidad explicada de las diferentes componentes para poder aplicar el método del codo de la curva y obtener el número de componentes adecuadas.

```
get_eig(pca_pima)
```

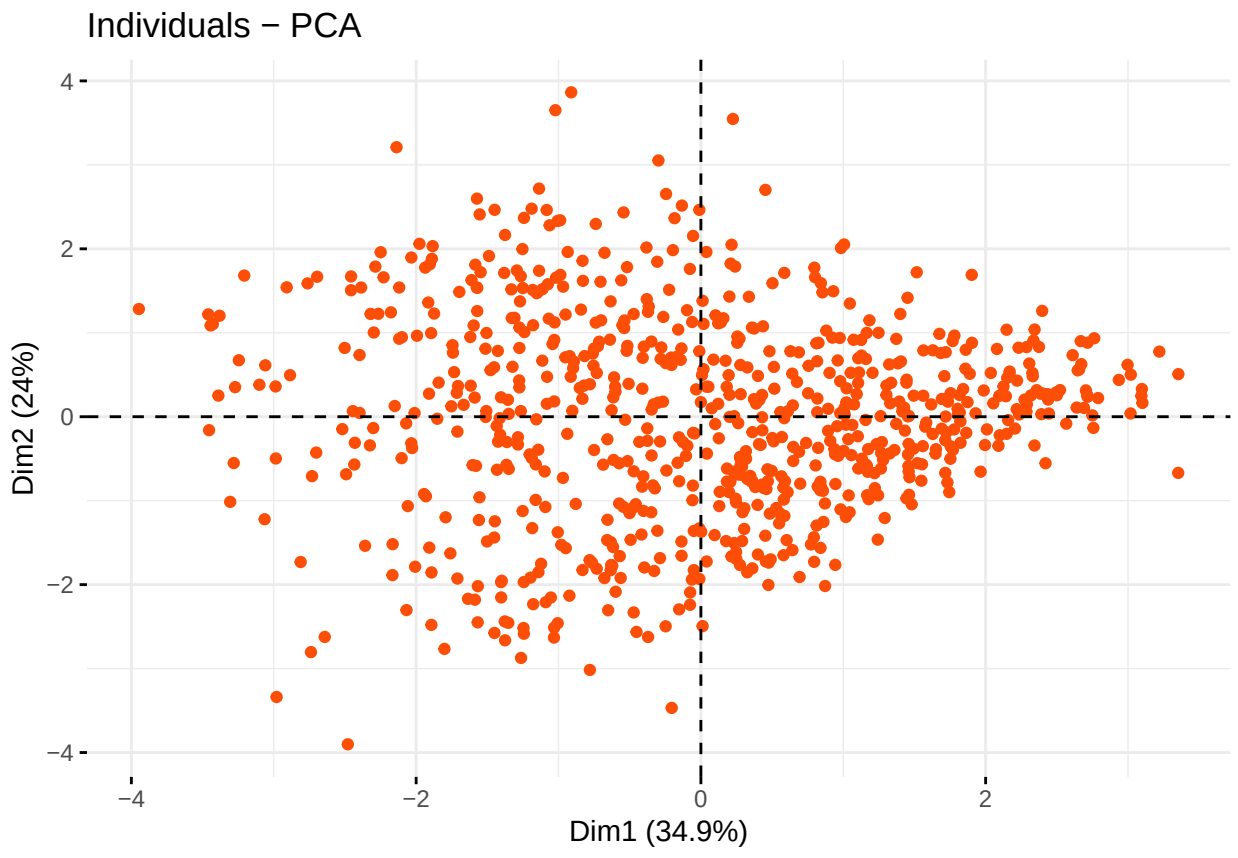
```
##        eigenvalue variance.percent cumulative.variance.percent
## Dim.1  2.0966456        34.944094                    34.94409
## Dim.2  1.4416899        24.028164                    58.97226
## Dim.3  0.9802707        16.337846                    75.31010
## Dim.4  0.7571519        12.619199                    87.92930
## Dim.5  0.4281918         7.136530                    95.06583
## Dim.6  0.2960500         4.934167                   100.00000
```

Según el método de los autovalores, nos deberíamos quedar con aquellos mayores a 1.

En nuestro caso son las dos primeras componentes. A pesar de esto, antes hemos visto como las tres primeras componentes explicaban apenas un 70% de la variabilidad, una cifra muy pobre. En nuestro caso vamos a elegir un porcentaje mínimo de varianza explicada de un 90 % con el que nos quedaríamos con 5 componentes. Esta aproximanción la deberíamos hacer con un experto en los datos ya depende mucho de nuestro objetivo y el objetivo del modelo.

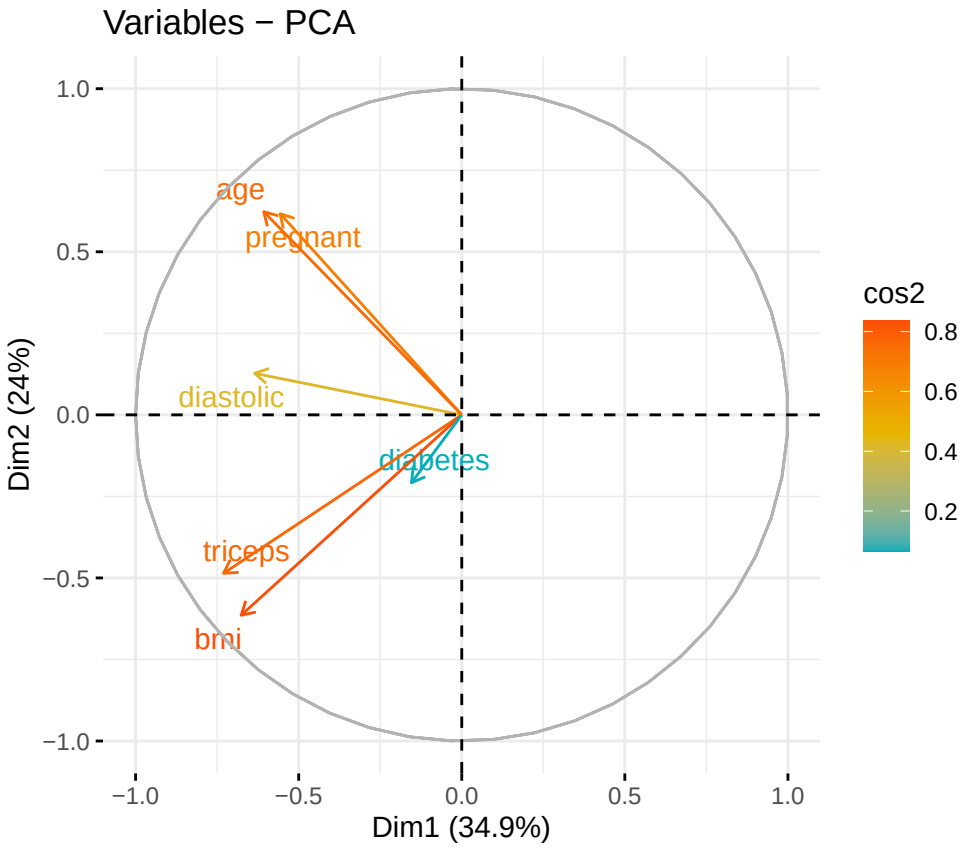
Una vez visto el número de PC que queremos, vamos a empezar a analizar cómo se ven los datos representados respecto de ellas.

```
fviz_pca_ind(pca_pima, geom.ind = "point", col.ind = "#FC4E07", axes = c(1, 2), pointsize = 1.5)
```



Con este gráfico podemos ver los datos representados a partir de nuestras dos primeras componentes principales. Vemos como los datos tienen valores un poco más altos en la dimensión uno mientras que, con respecto a la dos, están un poco más aplanados. En el siguiente gráfico vamos a ver cómo interviene cada variable en las dos primeras componentes.

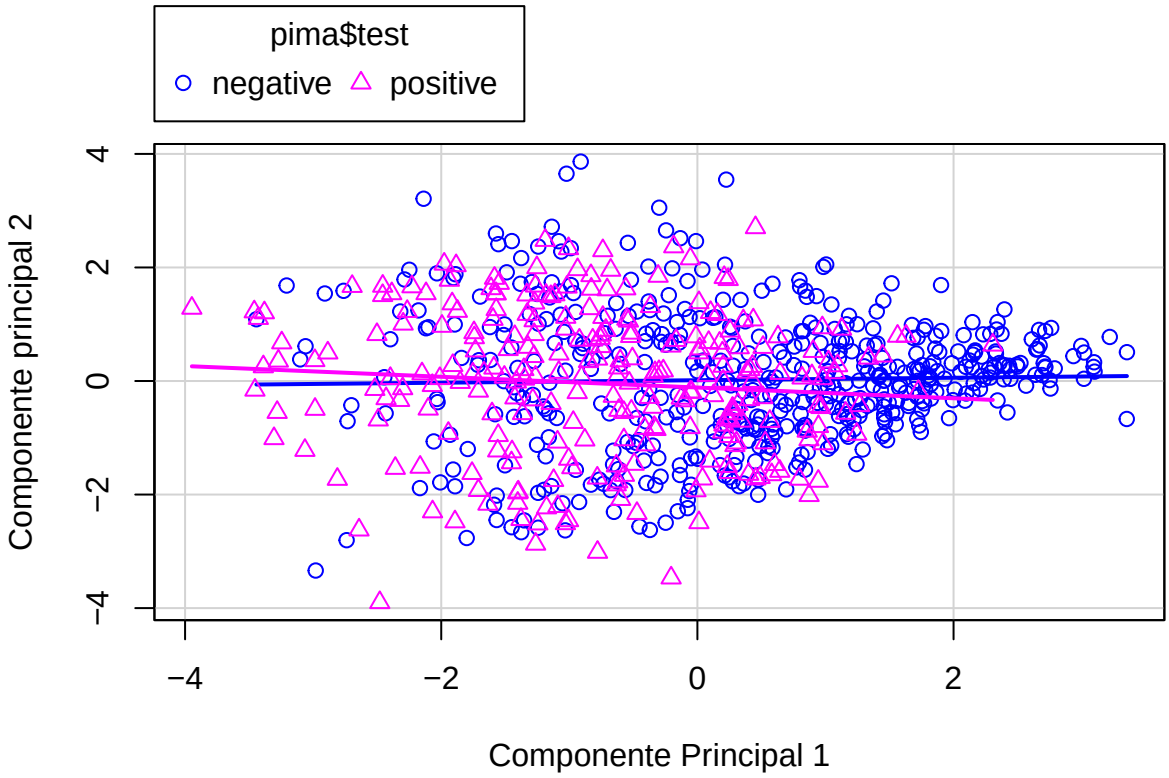
```
fviz_pca_var(pca_pima, col.var = "cos2", gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"), repel = TRU
```



Vemos cómo intervienen nuestras variables iniciales con el gráfico de saturaciones. Las variables que más se alejan del eje vertical son aquellas que intervienen más en la dimensión 1. En este caso son los embarazos, la edad, bmi, presión diastólica y el pliegue cutáneo del triceps. Con respecto a la segunda dimensión, las variables que más se alejan del eje horizontal son las mismas excepto la presión diastólica. Si volvemos a los datos de rotation que hemos impreso antes, los resultados concuenrdas ya que en las dos primeras dimensiones, el peso de las variables es muy similar excepto en la presion diastólica que tiene un valor bastante mayor en la primera.

A continuación, vamos a ver la distribución de los casos de diabéticos y no diabéticos con respecto a las dos primeras componentes. Con este gtráfico podemos comprobar si hay alguna diferencia sustancial entre los pacientes con diabetes y aquellos que no la tienen con respecto a las primeras componentes principales.

```
dataLL<-data.frame(pca_pima$x)
library(car)
scatterplot(dataLL[,2]~ dataLL[,1], xlab= "Componente Principal 1", ylab="Componente principal 2", leger
```



Como podemos ver, las dos primeras componentes no diferencian en el valor de test de diabetes.

Con esto damos por terminado nuestro análisis. Hemos podido ver muchas características y gráficos de nuestras PC que nos han indicado que no vamos a mejorar el modelo con esta técnica. Esto ya lo preveíamos antes de empezar al no cumplir una de las dos condiciones iniciales. A pesar de esto, vamos a comprobar igualmente cómo se comporta el modelo con las PC (las nuevas variables) y ver si el análisis que hemos realizado es acertado.

Lo primero vamos a añadir las 5 primeras componentes a nuestro dataframe. Hemos decidido que sean 5 ya que son el mínimo número de componentes con las cuales conseguimos explicar un 90% de la variabilidad.

```
pima$PC1 <- dataLL$PC1
pima$PC2 <- dataLL$PC2
pima$PC3 <- dataLL$PC3
pima$PC4 <- dataLL$PC4
pima$PC5 <- dataLL$PC5
```

Una vez hecho esto imprimimos las correlaciones de nuestro nuevo DataFrame y observamos algunas de sus características. Esto no haría falta ya que son las mismas que en el apartado anterior pero nos sirve para verificar la ortogonalidad de las componentes.

```
cor(pima[,c(1,3,4,5,6,7,9,10,11,12,13)])
```

```
##              pregnant   diastolic     triceps         bmi    diabetes
## pregnant   1.00000000  0.21943786  0.15677176  0.03952926 -0.01708549
## diastolic  0.21943786  1.00000000  0.23305085  0.28678209  0.01389454
```

```
## triceps    0.15677176  0.23305085  1.00000000  0.68067918  0.08384258
## bmi        0.03952926  0.28678209  0.68067918  1.00000000  0.11763188
## diabetes  -0.01708549  0.01389454  0.08384258  0.11763188  1.00000000
## age        0.54805315  0.34224819  0.15208741  0.03678163  0.04788511
## PC1       -0.55687999 -0.63496227 -0.73035933 -0.67636188 -0.15427008
## PC2        0.61681915  0.12728740 -0.48585193 -0.61445727 -0.20818853
## PC3       -0.02874584  0.12298090  0.11941041  0.10629121 -0.96143786
## PC4       -0.36288431  0.73418295 -0.27333521 -0.06810018  0.06926425
## PC5        0.42029278  0.13951908 -0.09550555  0.03333350  0.05548347
##                   age           PC1           PC2           PC3           PC4
## pregnant   0.54805315 -5.568800e-01  6.168192e-01 -2.874584e-02 -3.628843e-01
## diastolic  0.34224819 -6.349623e-01  1.272874e-01  1.229809e-01  7.341829e-01
## triceps    0.15208741 -7.303593e-01 -4.858519e-01  1.194104e-01 -2.733352e-01
## bmi        0.03678163 -6.763619e-01 -6.144573e-01  1.062912e-01 -6.810018e-02
## diabetes   0.04788511 -1.542701e-01 -2.081885e-01 -9.614379e-01  6.926425e-02
## age        1.00000000 -6.071769e-01  6.229523e-01 -1.200029e-01 -4.790626e-02
## PC1       -0.60717692  1.000000e+00 -8.372929e-16  5.565151e-17  6.303175e-16
## PC2        0.62295234 -8.372929e-16  1.000000e+00 -2.827232e-15 -8.278720e-16
## PC3       -0.12000289  5.565151e-17 -2.827232e-15  1.000000e+00  1.020631e-17
## PC4       -0.04790626  6.303175e-16 -8.278720e-16  1.020631e-17  1.000000e+00
## PC5       -0.46772787 -1.058953e-15  4.359618e-16  1.962398e-16  8.572252e-16
##                    PC5
## pregnant   4.202928e-01
## diastolic  1.395191e-01
## triceps   -9.550555e-02
## bmi        3.333350e-02
## diabetes   5.548347e-02
## age       -4.677279e-01
## PC1       -1.058953e-15
## PC2        4.359618e-16
## PC3        1.962398e-16
## PC4        8.572252e-16
## PC5        1.000000e+00
```

Nos vamos a centrar en analizar las PC. Como podemos ver, la correlaión entre las diferentes PC es prácti-acamente nula. Esto concuerda con la idea de que las PC tienen que ser ortogonales entre sí.

Ahora vamos a analizar el modelo con las PC.

```
modelo1 <- lm(glucose ~ PC1 + PC2 + PC3+ PC4 +PC5, data=pima)
summary(modelo1)
```

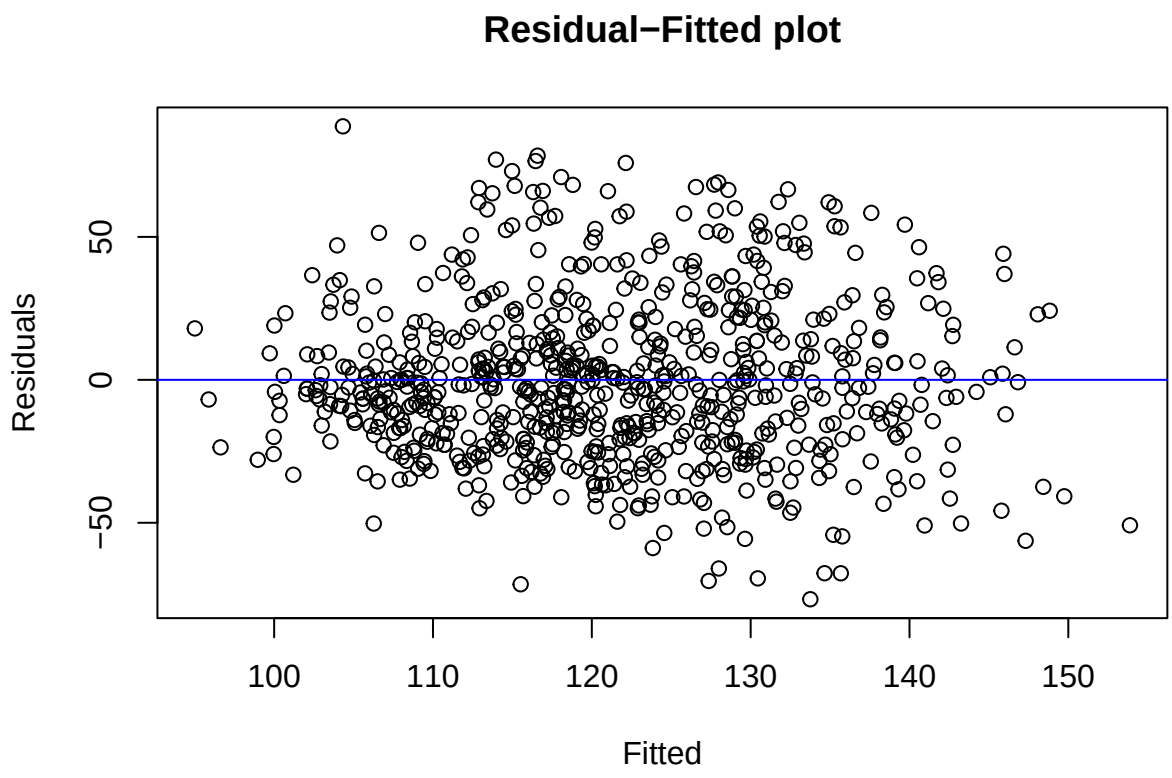
```
##
## Call:
## lm(formula = glucose ~ PC1 + PC2 + PC3 + PC4 + PC5, data = pima)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -76.756 -19.268  -2.797  16.492  88.671
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 121.2886     1.0244 118.398  < 2e-16 ***
## PC1          -7.0757     0.7079  -9.995  < 2e-16 ***
```

```
## PC2           0.3984     0.8537   0.467  0.64090
## PC3          -1.6851     1.0353  -1.628  0.10404
## PC4           2.0140     1.1781   1.710  0.08775 .
## PC5          -4.4934     1.5665  -2.868  0.00424 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 28.28 on 756 degrees of freedom
## Multiple R-squared:  0.1309, Adjusted R-squared:  0.1252
## F-statistic: 22.78 on 5 and 756 DF,  p-value: < 2.2e-16
```

El modelo de regresión resultante es Y = -7.4743 * PC1 -0.8204 * PC2 -2.3835 * PC3 -1.6623 * PC4 -2.3452*PC5 El modelo de regresión múltiple generado con ambas combinaciones de PC tienen un R2 muy bajo (0.1243), es decir, es capaz de explicar un porcentaje muy bajo de la variabilidad observada. El p-valor es muy bajo, por lo que se puede aceptar que el modelo no es fruto del azar. En cuanto al p_valor del test de significatividad individual vemos como solo dos de estos resultan significativos. Esto no tendría mucho sentido en una análisis de componentes principales pero puede ser resultado de lo que hemos dicho antes sobre las variables no correlacionadas.

En cuanto a los residuos, vamos a mostrarlos gráficamente. Si son homocedásticos, los residuos deben de distribuirse aleatoriamente en torno a la recta de regresión con una variabilidad constante a lo largo del eje X.

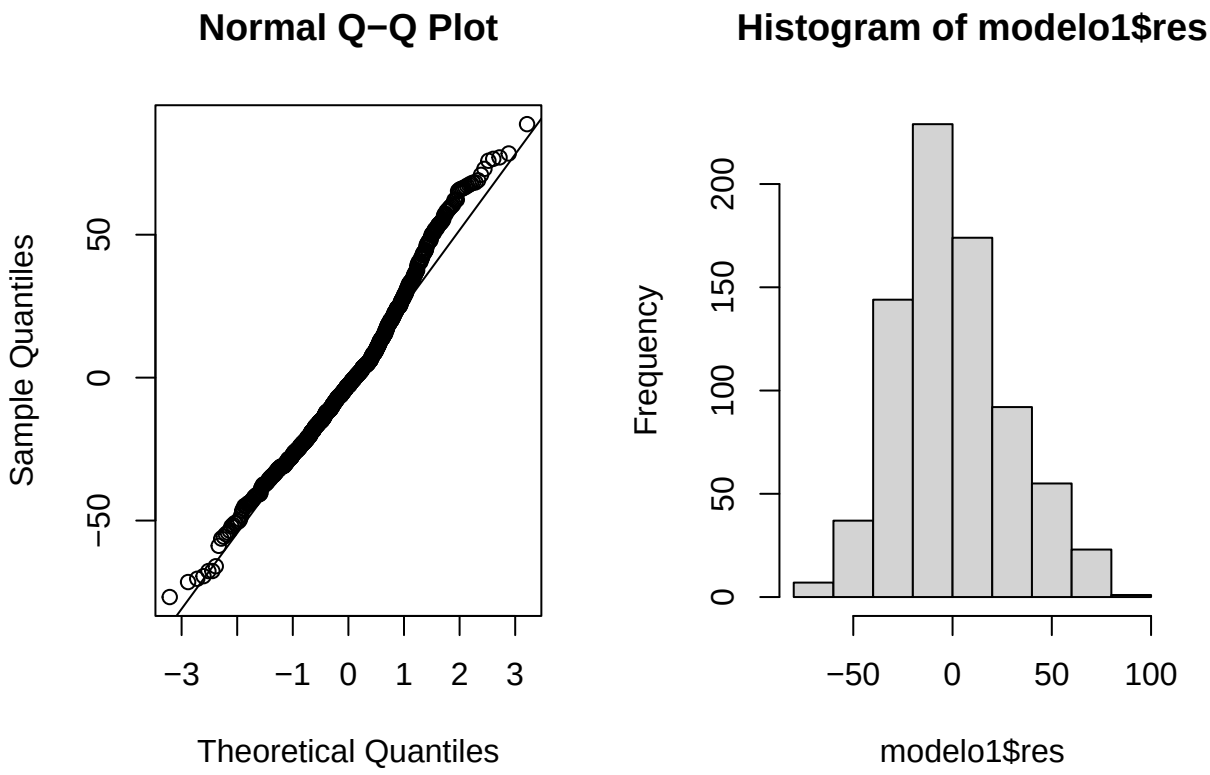
```
par(mfrow=c(1,1))
plot(modelo1$fit,modelo1$res,xlab="Fitted",ylab="Residuals", main="Residual-Fitted plot")
abline(h=0, col='blue')
```



Vemos que esto sí se cumple ya que no hay una tendencia o patrón en la variabilidad en ninguno de los modelos

Comprobamos la normalidad de los residuos con el qq-plot y el hostograma. Los residuos son normales si se ajustan a la recta del qq-plot y siguen una distribución similar a una campana en el histograma. Vemos que se cumple también.

```
par(mfrow=c(1,2))
qqnorm(modelo1$res)
qqline(modelo1$res)
hist(modelo1$res,10)
```



Y por último, comprobamos la independencia mediante el estadístico de Durbin-Watson y que la esperanza de los residuos sea 0.

```
library(lmtest)
dwtest(modelo1,alternative ="two.sided",iterations = 1000)
```

```
##
##  Durbin-Watson test
##
## data:  modelo1
## DW = 1.9531, p-value = 0.5151
## alternative hypothesis: true autocorrelation is not 0
```

```
mean(modelo1$res)
```

```
## [1] -7.376991e-16
```

Podemos comprobar que ambos modelos cumplen todas las condiciones para ser válidos ya que el estadístico de Durbin-Watson está entre 1.5 y 2.5 y la media de los residuos es casi 0.

Aunque el modelo supera todas las condiciones, como hemos dicho antes, este modelo no mejoran para nada los conseguidos en el apartado 1, debido a que no cumplen las condiciones iniciales por lo que era de esperar que no funcionara correctamente. (el R2 es muy bajo)

Como conclusión, determinamos que un modelo de regresión no es el más adecuado para estos datos ya que tenemos una capacidad predictiva baja debido a un bajo R2. Si tenemos en cuenta esto, era obvio que el análisis de componentes principales no nos iba a ayudar a mejorar el modelo, sumado a que hemos visto que los datos no están correlacionados.