Tarea Machine Learning

Carlos Manuel Martín Rengel

Introducción

El archivo de datos que le presentamos contiene información sobre la diabetes. Los datos han sido tomados de: https://www.kaggle.com/datasets/tigganeha4/diabetes-dataset-2019.

Para encontrar el archivo de datos que tiene que usar vaya a la carpeta **Machine Learning (asignaciones)** y allí encontrará otra con su nombre donde están los datos que le corresponden. El data_frame que hay dentro del archivo se llama *diabetes*. Todos los archivos son distintos, use exactamente el que le ha correspondido para evitar problemas.

Tenemos dos posibles repuestas, la primera es una variable binaria que indica si el individuo es diabético o no y la segunda es el índice de masa corporal (BMI) que utilizaremos en una regresión.

Disponemos de un total de 906 individuos. Puede encontrar información adicional en:

Tigga, N. P., & Garg, S. (2020). Prediction of Type 2 Diabetes using Machine Learning Classification Methods. Procedia Computer Science, 167, 706-716. DOI: https://doi.org/10.1016/j.procs.2020.03.336

El artículo se proporciona junto con los datos.

Se trata de buscar qué factores influyen en la diabetes y el BMI (body mmass index) y si es posible predecirlas a partir del perfil de los individuos.

En particular, ¿Influyen las variables independientes en la presencia de diabetes?.

¿Es posible predecir aproximadamente el BMI a partir de las variables seleccionadas? ?.

Instrucciones

- A continuación le mostramos, a grandes rasgos, los pasos que tiene que seguir. Le recomendamos que utilice este mismo R Markdown para mostrar los resultados de forma que queden integrados los cálculos y los comentarios. Si lo hace así, le regalo un punto extra.
- Si usa Markdown, suba también el PDF compilado, además del archivo original.
- Si no usa markdown, debe subir también el script de R con el que haga los cálculos y suba también los resultados en PDF convenientemente comentados.
- Las puntuaciones de los apartados son sobre 10.

Limpieza del archivo y colocación de las variables.

Primero abrimos el archivo y cambiamos los nombres de las variables. En mi caso, las he puesto en español para facilitar la interpretación de los resultados.

```
load("MARTIÌN.rda")
head(diabetes)
```

```
##              Age Gender Family_Diabetes highBP     PhysicallyActive BMI Smoking
## 612 less than 40   Male              no     no more than half an hr  22      no
## 615 less than 40 Female             yes     no                 none  21      no
```

```
## 745 less than 40 Female              no     no       one hr or more  32      no
## 19         40-49   Male             yes     no       one hr or more  24      no
## 198 less than 40 Female             yes     no less than half an hr  21      no
## 448 less than 40   Male              no     no more than half an hr  21      no
##     Alcohol Sleep SoundSleep RegularMedicine     JunkFood    Stress BPLevel
## 612      no     6          6              no occasionally sometimes     low
## 615     yes     9          8              no occasionally sometimes  normal
## 745      no     8          8             yes occasionally sometimes  normal
## 19       no    11         11              no occasionally sometimes  normal
## 198      no     8          7              no        often sometimes  normal
## 448      no     7          4              no occasionally sometimes  normal
##     Pregancies Pdiabetes UriationFreq Diabetic
## 612          0         0     not much       no
## 615          0         0     not much       no
## 745          3         0     not much       no
## 19           0         0     not much       no
## 198          0         0     not much       no
## 448          0         0  quite often       no
```

```
names(diabetes)
```

```
##  [1] "Age"              "Gender"           "Family_Diabetes"  "highBP"
##  [5] "PhysicallyActive" "BMI"              "Smoking"          "Alcohol"
##  [9] "Sleep"            "SoundSleep"       "RegularMedicine"  "JunkFood"
## [13] "Stress"           "BPLevel"          "Pregancies"       "Pdiabetes"
## [17] "UriationFreq"     "Diabetic"
```

```
colnames(diabetes)=c('Edad', 'Genero', 'D.Familiar', 'BPAlto', 'Actividad', 'IBM', 'Tabaco',
                     'Alcohol','Sueño', 'SProfundo', 'MedicinaRegular', 'ComidaBasura', 'Estres', 'Nivel
                     'Embarazos', 'Pdiabetes', 'FrecuenciaUrinaria', 'Diabetico')
```

Ahora miraremos el tipo de cada columna.

```
Classes=sapply(diabetes,class)
Classes
```

```
##              Edad            Genero         D.Familiar             BPAlto
##       "character"       "character"       "character"       "character"
##         Actividad               IBM            Tabaco            Alcohol
##       "character"         "integer"       "character"       "character"
##             Sueño          SProfundo    MedicinaRegular       ComidaBasura
##         "integer"         "numeric"       "character"       "character"
##            Estres           NivelBP         Embarazos         Pdiabetes
##       "character"       "character"         "integer"       "character"
## FrecuenciaUrinaria          Diabetico
##       "character"       "character"
```

Algunas de estas variables tienen que ser nominales con varias categorías, como por ejemplo, género o edad. Procedemos a realizar la transformación:

```
diabetes$Edad=factor(diabetes$Edad)
levels(diabetes$Edad)<-c('40-49', '50-59', '60 o mas', 'Menos de 40')

diabetes$Genero=factor(diabetes$Genero)
levels(diabetes$Genero)<-c('Mujer', 'Hombre')

diabetes$D.Familiar=factor(diabetes$D.Familiar)
```

```
levels(diabetes$D.Familiar)<-c('No', 'Si')

diabetes$BPAlto=factor(diabetes$BPAlto)
levels(diabetes$BPAlto)<-c('No', 'Si')

diabetes$Actividad=factor(diabetes$Actividad)
levels(diabetes$Actividad)<-c('-30min', '+30min', 'Nada', '1h')

diabetes$Tabaco=factor(diabetes$Tabaco)
levels(diabetes$Tabaco)<-c('No', 'Si')

diabetes$Alcohol=factor(diabetes$Alcohol)
levels(diabetes$Alcohol)<-c('No', 'Si')

diabetes$MedicinaRegular=factor(diabetes$MedicinaRegular)
levels(diabetes$MedicinaRegular)<-c('No', 'Si')

diabetes$ComidaBasura=factor(diabetes$ComidaBasura)
levels(diabetes$ComidaBasura)<-c('Siempre', 'Ocasional', 'A menudo', 'Casi siempre')

diabetes$Estres=factor(diabetes$Estres)
levels(diabetes$Estres)<-c('Siempre', 'Nada', 'A veces', 'Casi siempre')

diabetes$NivelBP=factor(diabetes$NivelBP)
levels(diabetes$NivelBP)<-c('Alto', 'Bajo', 'Normal')

diabetes$Pdiabetes=factor(diabetes$Pdiabetes)
levels(diabetes$Pdiabetes)<-c('No', 'Si')

diabetes$FrecuenciaUrinaria=factor(diabetes$FrecuenciaUrinaria)
levels(diabetes$FrecuenciaUrinaria)<-c('No mucha', 'A menudo')

diabetes$Diabetico=factor(diabetes$Diabetico)
levels(diabetes$Diabetico)<-c('No', 'Si')
```

Ahora cambiamos las variables denominadas *interger* a *numeric* con el siguiente comando:

```
Classes=sapply(diabetes,class)
for (i in 1:ncol(diabetes))
  if (Classes[i]=='integer') diabetes[[i]]=as.numeric(diabetes[[i]])
Classes=sapply(diabetes,class)
Classes
```

```
##              Edad            Genero        D.Familiar            BPAlto
##          "factor"          "factor"          "factor"          "factor"
##         Actividad               IBM            Tabaco           Alcohol
##          "factor"         "numeric"          "factor"          "factor"
##             Sueño         SProfundo   MedicinaRegular      ComidaBasura
##         "numeric"         "numeric"          "factor"          "factor"
##            Estres           NivelBP         Embarazos         Pdiabetes
##          "factor"          "factor"         "numeric"          "factor"
## FrecuenciaUrinaria         Diabetico
##          "factor"          "factor"
```

Ya tenemos la matriz completa, con toda la información en español y lista para el análisis. Podemos buscar

los datos perdidos, pero en este caso, al estar la matriz completa, devuelve un resultado de 0.

```
which(is.na(diabetes))
```

```
## integer(0)
```

Análisis descriptivo y visual de las vriables.

Una vez están los datos preparados, procedemos a realizar un análisis descriptivo y visual de las variables. Primero, la funcion *summary* nos proporciona el mínimo, el máximo y los percentiles 25, 50 y 75. Tambien nos proporciona la media y la mediana. Solo lo aplicamos a las variables numéricas, ya que, en las otras, al ser nominales, no nos arrojaría unos resultados interpretables.

```
summary(diabetes[,Classes=="numeric"])
```

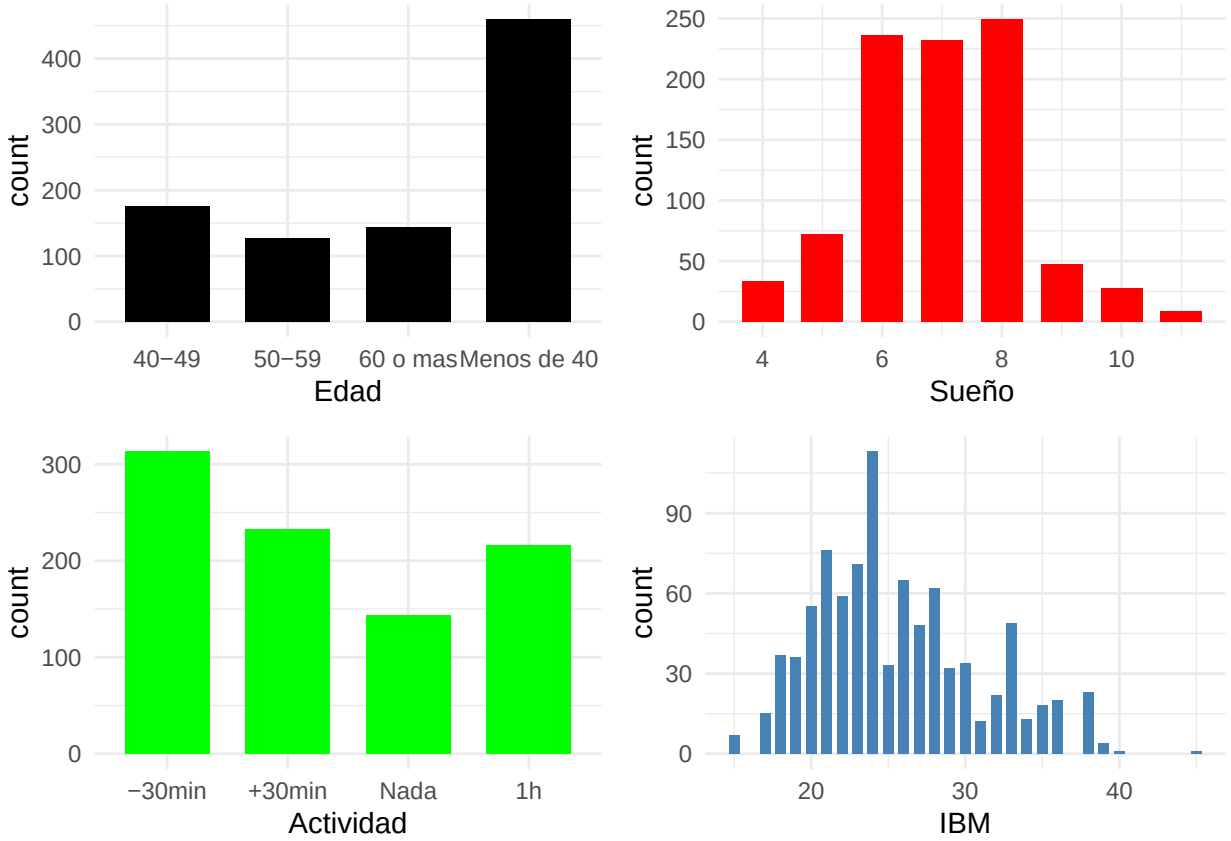
```
##       IBM            Sueño          SProfundo        Embarazos
##  Min.   :15.00   Min.   : 4.000   Min.   : 0.000   Min.   :0.0000
##  1st Qu.:22.00   1st Qu.: 6.000   1st Qu.: 4.000   1st Qu.:0.0000
##  Median :24.00   Median : 7.000   Median : 6.000   Median :0.0000
##  Mean   :25.55   Mean   : 6.982   Mean   : 5.597   Mean   :0.4106
##  3rd Qu.:29.00   3rd Qu.: 8.000   3rd Qu.: 7.000   3rd Qu.:0.0000
##  Max.   :45.00   Max.   :11.000   Max.   :11.000   Max.   :4.0000
```

Hecho esto, utilizamos los paquetes *ggplot2* y *gridExtra* para el análisis visual. Como ejemplo, he seleccionado las variables *Edad*, *Sueño*, *Actividad* e *IBM* para dibujar un diagrama de barras y lo he juntado en un panel.

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.2.3
```

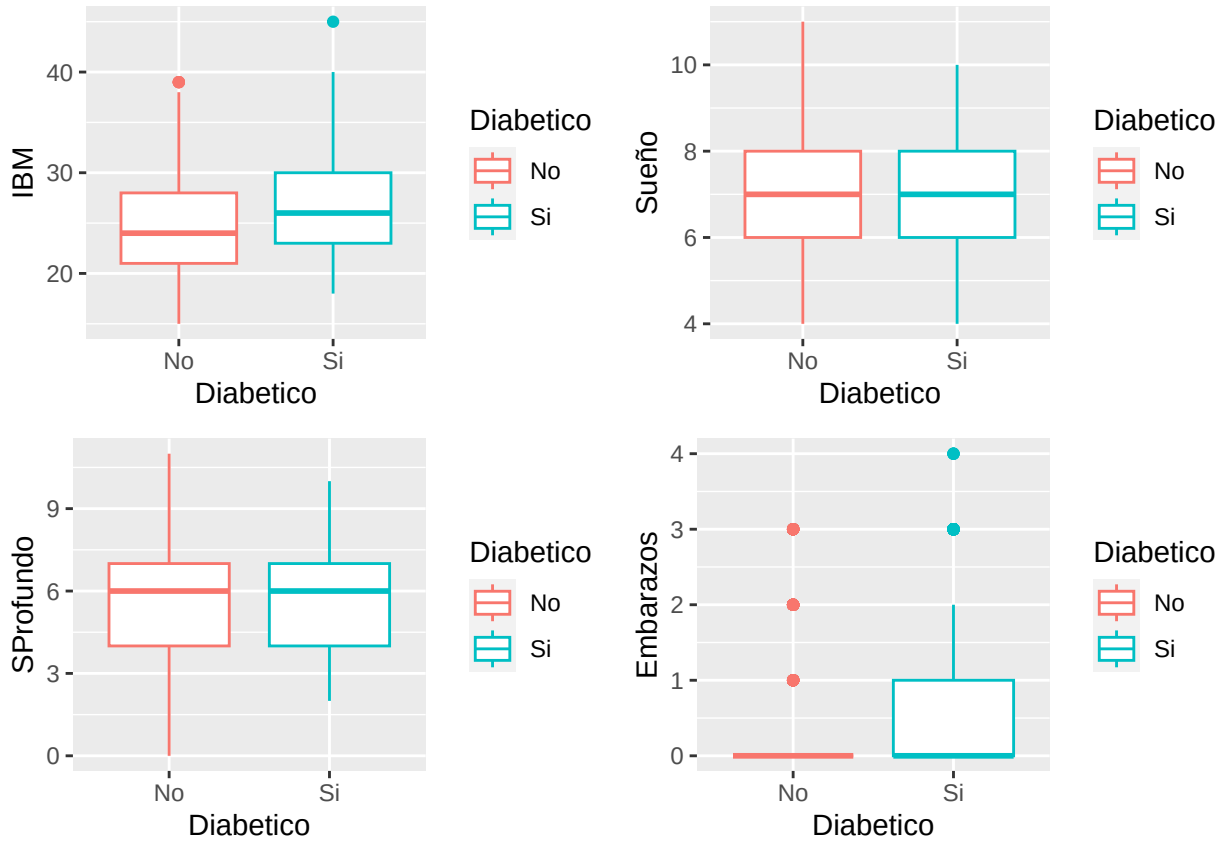
```
library(gridExtra)

g1 = ggplot(diabetes, aes(x=Edad))+
  geom_bar(stat="count", width=0.7, fill="black")+
  theme_minimal()
g2 = ggplot(diabetes, aes(x=Sueño))+
  geom_bar(stat="count", width=0.7, fill="red")+
  theme_minimal()
g3 = ggplot(diabetes, aes(x=Actividad))+
  geom_bar(stat="count", width=0.7, fill="green")+
  theme_minimal()
g4 = ggplot(diabetes, aes(x=IBM))+
  geom_bar(stat="count", width=0.7, fill="steelblue")+
  theme_minimal()
grid.arrange(g1, g2, g3, g4, nrow = 2, ncol=2)
```



Viendo los graficos, observamos que la mayoria de individuos eran menores de 40 años, las horas de sueño suelen ser de 6 a 8 y en la actividad, predomina el realizar menos de una hora de ejercicio al día.

Podemos observar las relaciones entre variables númericas mediante gráficos box-plot.

```
p1 <- ggplot(diabetes, aes(x=Diabetico, y=IBM, color=Diabetico)) +
  geom_boxplot()
p2 <- ggplot(diabetes, aes(x=Diabetico, y=Sueño, color=Diabetico)) +
  geom_boxplot()
p3 <- ggplot(diabetes, aes(x=Diabetico, y=SProfundo, color=Diabetico)) +
  geom_boxplot()
p4 <- ggplot(diabetes, aes(x=Diabetico, y=Embarazos, color=Diabetico)) +
  geom_boxplot()
grid.arrange(p1, p2, p3, p4, nrow = 2, ncol=2)
```



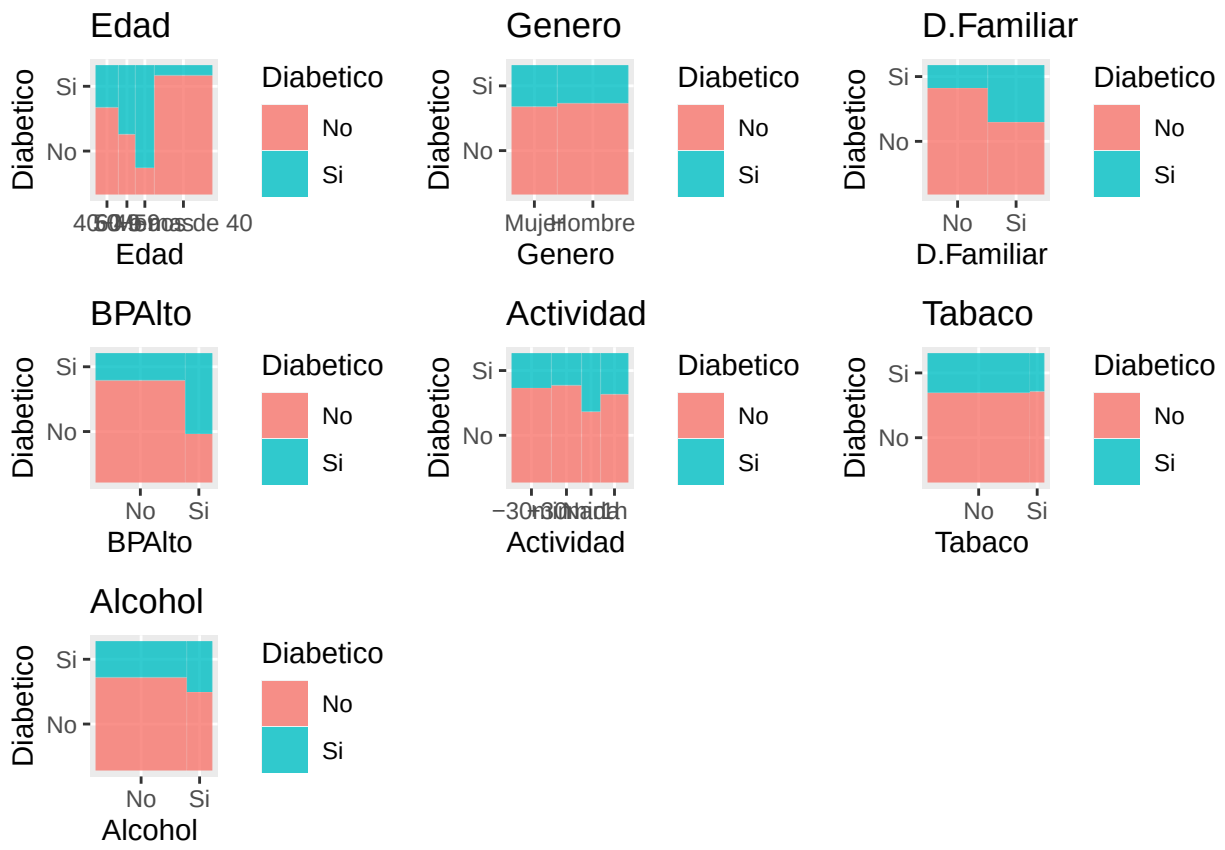
Observamos que no hay mucha diferencia en entre diabéticos y no diabéticos, solo en la variable *IBM* podemos apreciar unos valores mas elevados en los diabéticos.

Relizamos lo mismo para las variables nominales. Al ser muchas, realizamos dos tandas con 7 variables en cada una.

```
library(ggmosaic)
q1=ggplot(data = diabetes) +
  geom_mosaic(aes(x = product(Diabetico, Edad), fill=Diabetico))+
  labs(x = "Edad ", title='Edad')
q2=ggplot(data = diabetes) +
  geom_mosaic(aes(x = product(Diabetico, Genero ), fill=Diabetico))+
  labs(x = "Genero ", title='Genero')
q3=ggplot(data = diabetes) +
  geom_mosaic(aes(x = product(Diabetico,D.Familiar ), fill=Diabetico))+
  labs(x = "D.Familiar ", title='D.Familiar')
q4=ggplot(data = diabetes) +
  geom_mosaic(aes(x = product(Diabetico,BPAlto ), fill=Diabetico))+
  labs(x = "BPAlto ", title='BPAlto')
q5=ggplot(data = diabetes) +
  geom_mosaic(aes(x = product(Diabetico,Actividad ), fill=Diabetico))+
  labs(x = "Actividad ", title='Actividad')
q6=ggplot(data = diabetes) +
  geom_mosaic(aes(x = product(Diabetico,Tabaco ), fill=Diabetico))+
  labs(x = "Tabaco ", title='Tabaco')
q7=ggplot(data = diabetes) +
  geom_mosaic(aes(x = product(Diabetico,Alcohol ), fill=Diabetico))+
```

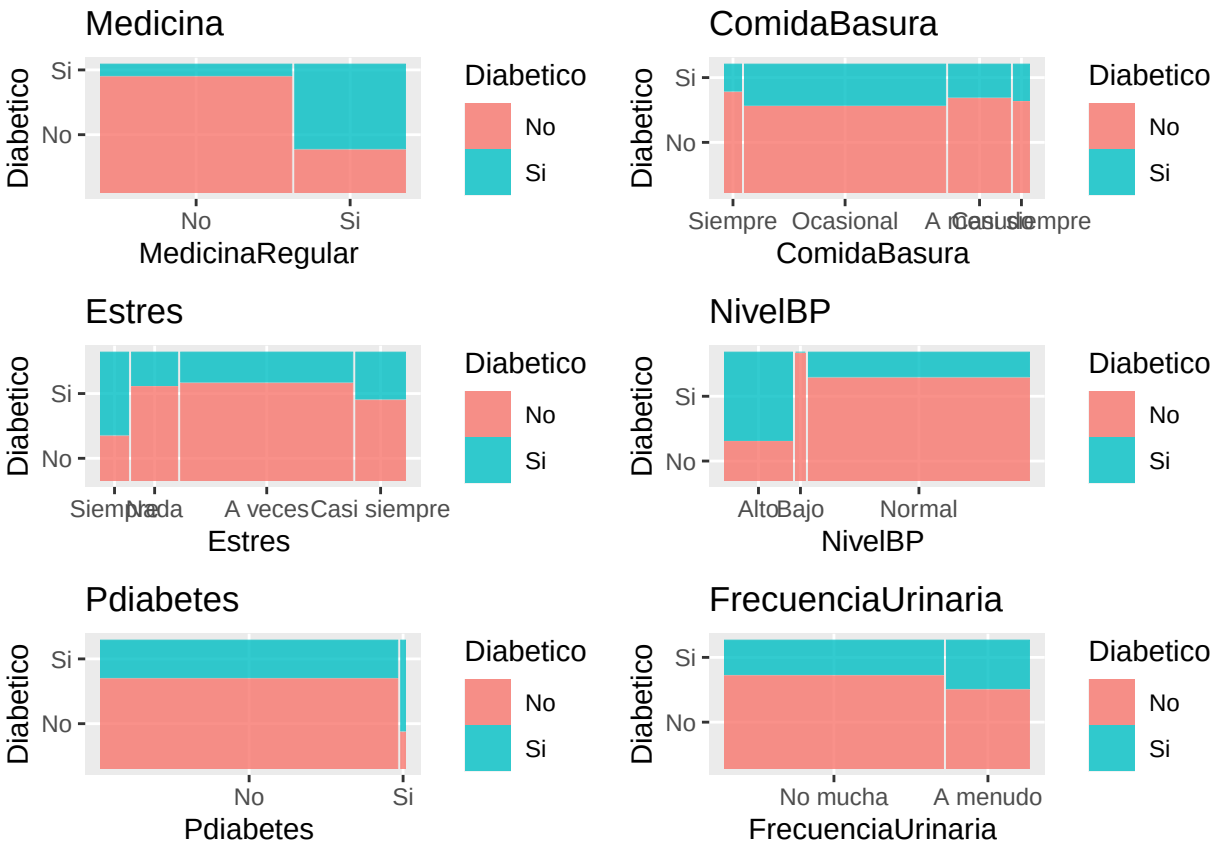
```
  labs(x = "Alcohol ", title='Alcohol')
```

```
grid.arrange(q1, q2, q3, q4, q5, q6, q7, nrow = 3, ncol=3)
```



En este caso, observamos que a más edad, más diabéticos hay. Tambien apreciamos que la variable *BPAlto* esta relacionada con la diabetes. En el resto de variables, no hay una distincion clara o reseñable. Continuamos con el resto de variables.

```
q8=ggplot(data = diabetes) +
  geom_mosaic(aes(x = product(Diabetico, MedicinaRegular), fill=Diabetico))+
  labs(x = "MedicinaRegular ", title='Medicina')
q9=ggplot(data = diabetes) +
  geom_mosaic(aes(x = product(Diabetico, ComidaBasura), fill=Diabetico))+
  labs(x = "ComidaBasura ", title='ComidaBasura')
q10=ggplot(data = diabetes) +
  geom_mosaic(aes(x = product(Diabetico, Estres), fill=Diabetico))+
  labs(x = "Estres ", title='Estres')
q11=ggplot(data = diabetes) +
  geom_mosaic(aes(x = product(Diabetico, NivelBP), fill=Diabetico))+
  labs(x = "NivelBP ", title='NivelBP')
q12=ggplot(data = diabetes) +
  geom_mosaic(aes(x = product(Diabetico, Pdiabetes), fill=Diabetico))+
  labs(x = "Pdiabetes ", title='Pdiabetes')
q13=ggplot(data = diabetes) +
  geom_mosaic(aes(x = product(Diabetico, FrecuenciaUrinaria), fill=Diabetico))+  labs(x = "FrecuenciaUr
grid.arrange(q8, q9, q10, q11, q12, q13, nrow = 3, ncol=2)
```



En este caso, la variable *Medicina Regular* está relacionada con la diabetes, como es lógico, al igual que la variable *NivelBP*, que nos arroja unos resultados similares a la variable *BPAlto*. En el resto de variables, una vez más, no hay nada digno de mención.

Tambien podemos realizar algún gráfico multivariante.

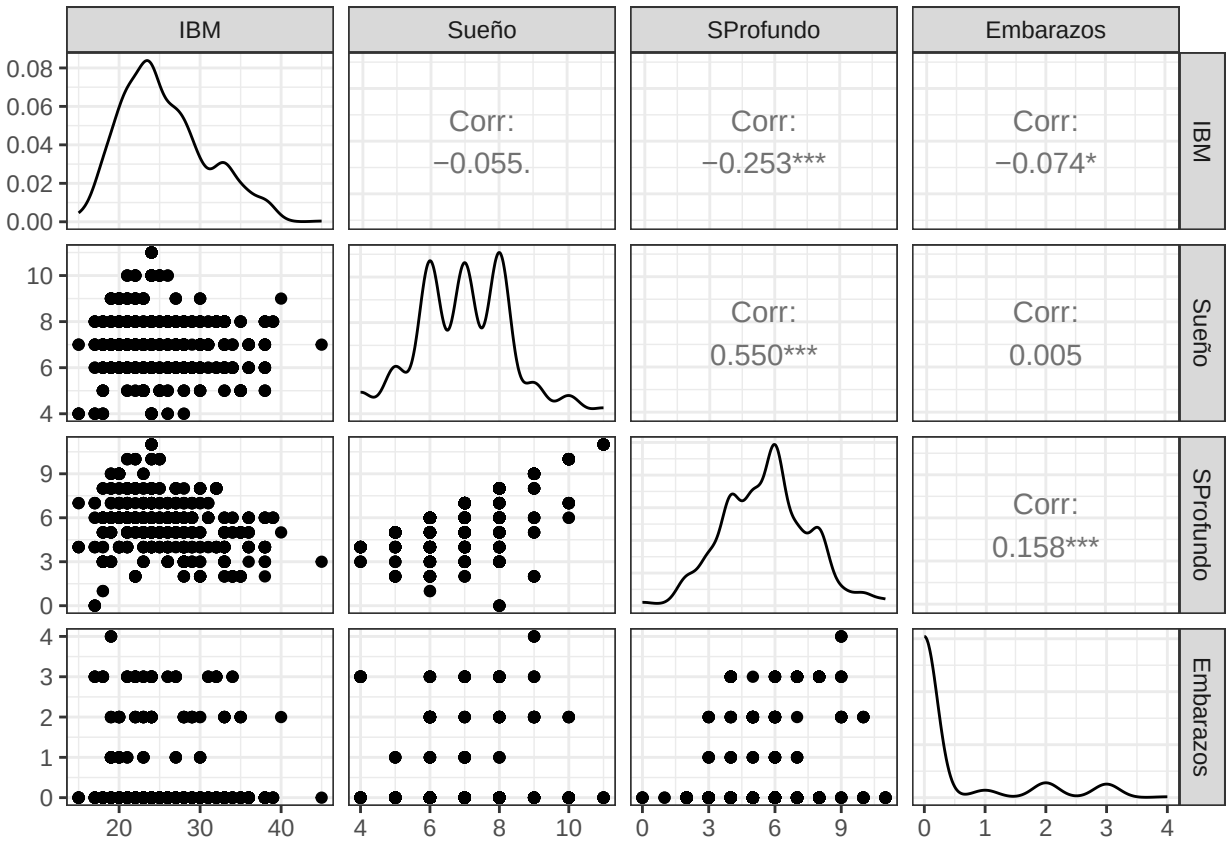
```
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
##
## Attaching package: 'GGally'
```

```
## The following object is masked from 'package:ggmosaic':
##
##     happy
```

```
library(ggplot2)
ggpairs(diabetes[,Classes=="numeric"])+ theme_bw()
```



Predicción del BMI y variables que influyen en el mismo.

En cuanto al modelo para predecir el IBM, realizamos lo siguiente:

```
modIBM=lm(IBM ~ ., data=diabetes)
summary(modIBM)
```

```
##
## Call:
## lm(formula = IBM ~ ., data = diabetes)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.5075  -3.0105  -0.3295   2.9442  15.9657
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 25.8706     1.3903  18.609  < 2e-16 ***
## Edad50-59                    0.6223     0.5873   1.060 0.289626
## Edad60 o mas                -2.0312     0.6179  -3.287 0.001051 **
## EdadMenos de 40             -2.7924     0.4601  -6.069 1.91e-09 ***
## GeneroHombre                -1.0543     0.4115  -2.562 0.010567 *
## D.FamiliarSi                 1.1886     0.3310   3.591 0.000348 ***
## BPAltoSi                     0.4462     0.5461   0.817 0.414140
## Actividad+30min             -0.6108     0.4167  -1.466 0.142992
## ActividadNada               -2.8566     0.5094  -5.608 2.75e-08 ***
## Actividad1h                 -0.8708     0.4264  -2.042 0.041410 *
```

```
## TabacoSi                     0.5633     0.6121   0.920 0.357743
## AlcoholSi                    1.0200     0.4797   2.126 0.033765 *
## Sueño                        0.3309     0.1453   2.278 0.022993 *
## SProfundo                   -0.4767     0.1136  -4.196 2.99e-05 ***
## MedicinaRegularSi           -0.9462     0.4537  -2.086 0.037302 *
## ComidaBasuraOcasional        1.1449     0.7918   1.446 0.148548
## ComidaBasuraA menudo         1.8163     0.8028   2.262 0.023923 *
## ComidaBasuraCasi siempre     0.8803     1.0233   0.860 0.389885
## EstresNada                   0.1626     0.7408   0.220 0.826294
## EstresA veces                0.2560     0.6603   0.388 0.698324
## EstresCasi siempre          -0.3788     0.7030  -0.539 0.590179
## NivelBPBajo                  0.3893     1.0610   0.367 0.713753
## NivelBPNormal                0.4146     0.5774   0.718 0.472973
## Embarazos                   -0.7396     0.2238  -3.304 0.000990 ***
## PdiabetesSi                  1.0058     1.3311   0.756 0.450071
## FrecuenciaUrinariaA menudo   2.5270     0.3949   6.399 2.54e-10 ***
## DiabeticoSi                  1.6212     0.5044   3.214 0.001357 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.613 on 879 degrees of freedom
## Multiple R-squared:  0.264,  Adjusted R-squared:  0.2423
## F-statistic: 12.13 on 26 and 879 DF,  p-value: < 2.2e-16
```

Observamos que hay variables muy significativas, como la edad, si hay antecedentes de diabetes en la familia, la falta de ejercicio, si orina a menudo o si presenta diabetes. Ahora realizamos lo mismo pero con el logaritmo de la variable *IBM*.

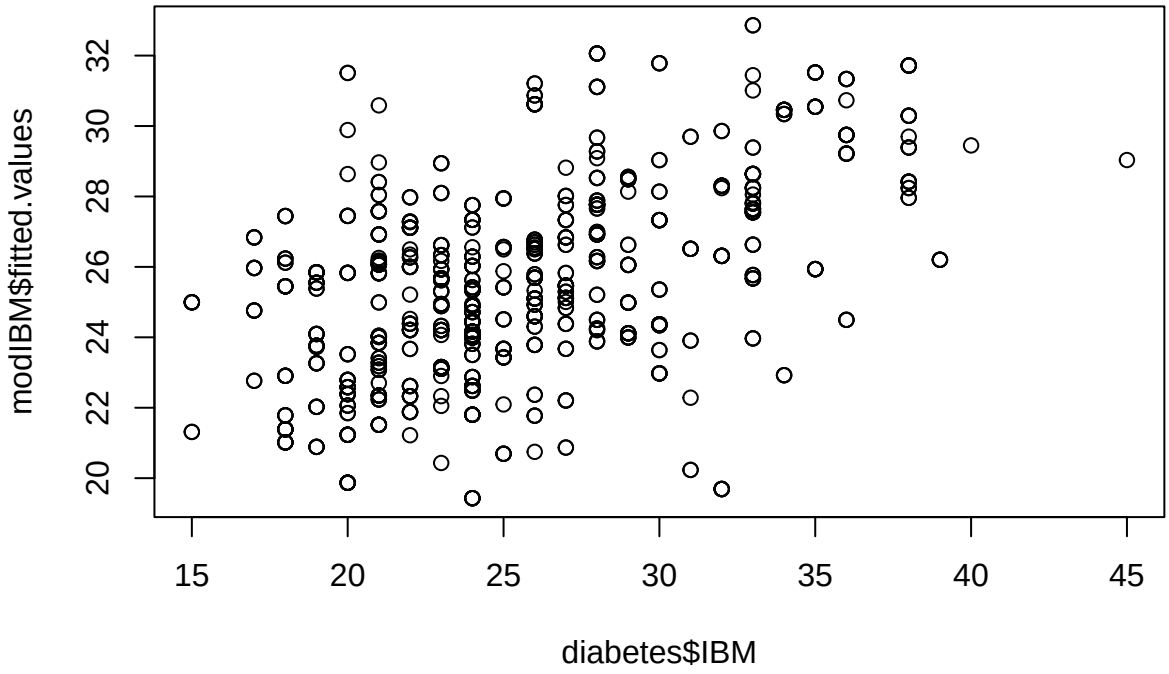
```
modIBM2=lm(log(IBM) ~ ., data=diabetes)
summary(modIBM2)
```

```
##
## Call:
## lm(formula = log(IBM) ~ ., data = diabetes)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.46239 -0.11350 -0.00074  0.12569  0.47179
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 3.232444   0.053777  60.108  < 2e-16 ***
## Edad50-59                   0.024797   0.022716   1.092 0.275316
## Edad60 o mas               -0.076444   0.023901  -3.198 0.001432 **
## EdadMenos de 40            -0.104449   0.017797  -5.869 6.21e-09 ***
## GeneroHombre               -0.038274   0.015916  -2.405 0.016393 *
## D.FamiliarSi                0.047204   0.012803   3.687 0.000241 ***
## BPAltoSi                    0.012882   0.021123   0.610 0.542120
## Actividad+30min            -0.021302   0.016117  -1.322 0.186608
## ActividadNada              -0.119240   0.019705  -6.051 2.12e-09 ***
## Actividad1h                -0.031701   0.016493  -1.922 0.054910 .
## TabacoSi                    0.022126   0.023678   0.934 0.350330
## AlcoholSi                   0.045189   0.018556   2.435 0.015079 *
## Sueño                       0.012387   0.005620   2.204 0.027789 *
## SProfundo                  -0.016084   0.004394  -3.660 0.000267 ***
```

```
## MedicinaRegularSi          -0.045353   0.017550  -2.584 0.009919 **
## ComidaBasuraOcasional       0.043400   0.030628   1.417 0.156832
## ComidaBasuraA menudo        0.069707   0.031055   2.245 0.025040 *
## ComidaBasuraCasi siempre    0.025670   0.039582   0.649 0.516812
## EstresNada                 -0.004765   0.028653  -0.166 0.867962
## EstresA veces              -0.002603   0.025540  -0.102 0.918844
## EstresCasi siempre         -0.034038   0.027194  -1.252 0.211026
## NivelBPBajo                 0.023081   0.041042   0.562 0.573999
## NivelBPNormal               0.016245   0.022335   0.727 0.467233
## Embarazos                  -0.028154   0.008658  -3.252 0.001190 **
## PdiabetesSi                 0.031268   0.051488   0.607 0.543819
## FrecuenciaUrinariaA menudo  0.090612   0.015276   5.932 4.31e-09 ***
## DiabeticoSi                 0.073348   0.019512   3.759 0.000182 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1784 on 879 degrees of freedom
## Multiple R-squared:  0.253,  Adjusted R-squared:  0.2309
## F-statistic: 11.45 on 26 and 879 DF,  p-value: < 2.2e-16
```

Nos da unos resultados similares.Podemos realizar un gráfico para representar los valores observados frente a los valores ajustados.

```
plot(diabetes$IBM, modIBM$fitted.values)
```



Obtenemos el coeficiente de determinación R^2, que sería el cuadrado del coeficiente de correlación entre los valores ajustados y los observados.

```
cor(diabetes$IBM, modIBM$fitted.values)^2
```

```
## [1] 0.2640384
```

Hecho esto, dividimos el conjunto de datos en dos partes, una para entrenar el modelo, con el 70% de los individuos y otra para validarlo con los restantes, el 30%. Seleccionamos una semilla para que salgan siempre los mismos resultados, en este caso es mi cumpleaños.

```
tr=round(nrow(diabetes)*0.7)
set.seed(23111998)
muestra=sample.int(nrow(diabetes), tr)
Train.diab=diabetes[muestra,]
Val.diab=diabetes[-muestra,]
```

Tenemos el conjunto **Train.diab**, que usaremos para ajustar el modelo.

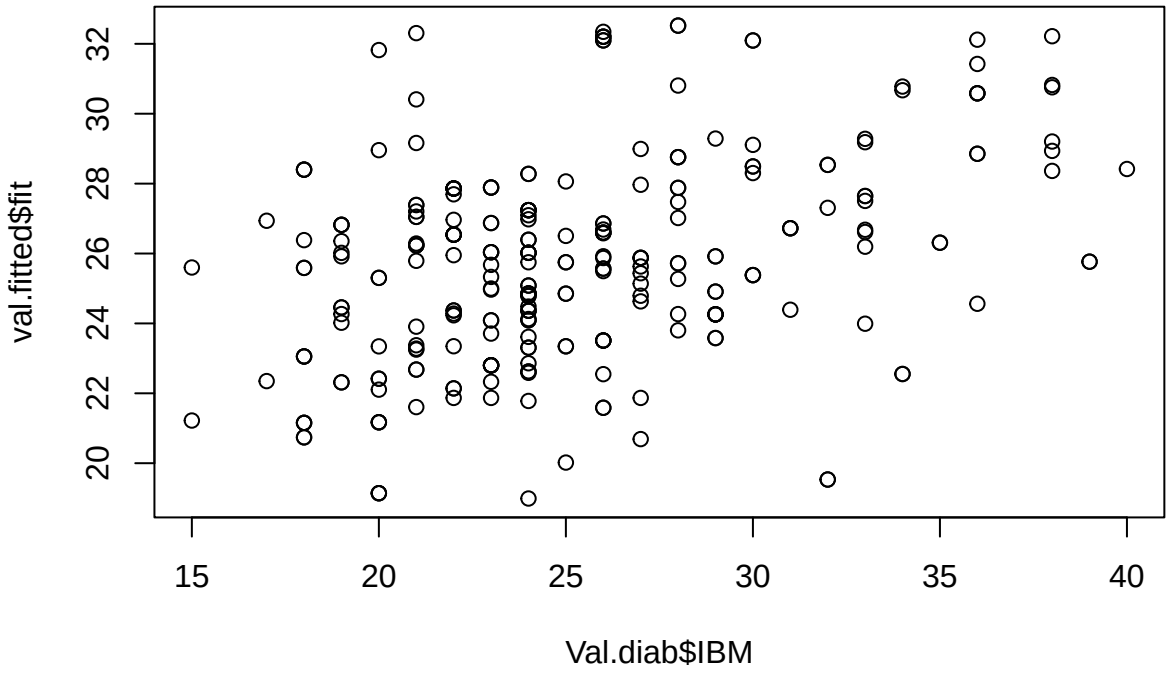
```
modIBMTrain=lm(IBM ~ ., data=Train.diab)
summary(modIBMTrain)
```

```
##
## Call:
## lm(formula = IBM ~ ., data = Train.diab)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.8214  -2.5311  -0.2893   3.2228  15.9465
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  25.08020    1.63639  15.327  < 2e-16 ***
## Edad50-59                    -0.05203    0.71211  -0.073 0.941778
## Edad60 o mas                 -1.85106    0.76788  -2.411 0.016223 *
## EdadMenos de 40              -2.97635    0.54168  -5.495 5.77e-08 ***
## GeneroHombre                 -1.12410    0.48090  -2.338 0.019737 *
## D.FamiliarSi                  1.41297    0.40028   3.530 0.000447 ***
## BPAltoSi                      0.57452    0.65780   0.873 0.382792
## Actividad+30min              -0.79067    0.49462  -1.599 0.110439
## ActividadNada                -3.43261    0.60493  -5.674 2.16e-08 ***
## Actividad1h                  -1.11517    0.50186  -2.222 0.026647 *
## TabacoSi                      0.34280    0.72184   0.475 0.635031
## AlcoholSi                     1.43692    0.57966   2.479 0.013448 *
## Sueño                         0.42108    0.17217   2.446 0.014740 *
## SProfundo                    -0.47148    0.12980  -3.632 0.000305 ***
## MedicinaRegularSi            -0.84704    0.54007  -1.568 0.117311
## ComidaBasuraOcasional         1.65866    0.92844   1.786 0.074517 .
## ComidaBasuraA menudo          2.12653    0.94352   2.254 0.024564 *
## ComidaBasuraCasi siempre      1.00783    1.17235   0.860 0.390311
## EstresNada                   -0.16794    0.89048  -0.189 0.850478
## EstresA veces                 0.20468    0.81631   0.251 0.802105
## EstresCasi siempre           -0.40596    0.84916  -0.478 0.632772
## NivelBPBajo                   0.11795    1.27683   0.092 0.926432
## NivelBPNormal                 0.45593    0.70109   0.650 0.515733
## Embarazos                    -0.84168    0.26186  -3.214 0.001377 **
## PdiabetesSi                   0.18223    1.56016   0.117 0.907055
## FrecuenciaUrinariaA menudo    2.71602    0.46581   5.831 8.97e-09 ***
## DiabeticoSi                   1.89677    0.61043   3.107 0.001976 **
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.552 on 607 degrees of freedom
## Multiple R-squared:  0.3106, Adjusted R-squared:  0.2811
## F-statistic: 10.52 on 26 and 607 DF,  p-value: < 2.2e-16
```

Nos da una R^2 similiar a la del modelo completo, aunque algo más elevada. Finalmente, usaremos el modelo para predecir la muestra de validación y realizamos otro gráfico que muestre las predicciones frente a los valores reales.

```
val.fitted=predict(modIBMTrain, newdata=Val.diab, se.fit = TRUE)
plot(Val.diab$IBM, val.fitted$fit)
```



Podemos observar que, si bien hay algunos puntos que si han tenido una buena predicción, no se ajusta mucho a los valores reales. Nos aseguramos mirando la R^2.

```
cor(Val.diab$IBM, val.fitted$fit)^2
```

```
## [1] 0.1483616
```

Nos da 0,15; un valor que no se asemeja al valor original, por lo que no es un buen modelo predictivo.

Predicción de la diabetes (binaria) y variables que influyen en la misma.

Para este apartado, es necesario convertir las variables nominales en variables binarias. Esto quiere decir que, por ejemplo, la variable *Genero* ha pasado a ser *GeneroHombre* por lo que si vale 1 es que el individuo es un hombre y si vale 0 es mujer. POr otra parte, la variable *Actividad* pasa a ser *Actividad +30min*,

ActividadNada o *Actividad1h*. Estas variables valen 1 cuando el tipo coincide y cero en caso contrario. Si no hay ningún 1, es que la categoría es *Actividad-30min*

```
X=model.matrix(Diabetico~., data=diabetes)
head(X)
```

```
##     (Intercept) Edad50-59 Edad60 o mas EdadMenos de 40 GeneroHombre
## 612           1         0            0               1            1
## 615           1         0            0               1            0
## 745           1         0            0               1            0
## 19            1         0            0               0            1
## 198           1         0            0               1            0
## 448           1         0            0               1            1
##     D.FamiliarSi BPAltoSi Actividad+30min ActividadNada Actividad1h IBM
## 612            0        0               1             0           0  22
## 615            1        0               0             1           0  21
## 745            0        0               0             0           1  32
## 19             1        0               0             0           1  24
## 198            1        0               0             0           0  21
## 448            0        0               1             0           0  21
##     TabacoSi AlcoholSi Sueño SProfundo MedicinaRegularSi ComidaBasuraOcasional
## 612        0         0     6         6                 0                     1
## 615        0         1     9         8                 0                     1
## 745        0         0     8         8                 1                     1
## 19         0         0    11        11                 0                     1
## 198        0         0     8         7                 0                     0
## 448        0         0     7         4                 0                     1
##     ComidaBasuraA menudo ComidaBasuraCasi siempre EstresNada EstresA veces
## 612                    0                        0          0            1
## 615                    0                        0          0            1
## 745                    0                        0          0            1
## 19                     0                        0          0            1
## 198                    1                        0          0            1
## 448                    0                        0          0            1
##     EstresCasi siempre NivelBPBajo NivelBPNormal Embarazos PdiabetesSi
## 612                  0           1             0         0           0
## 615                  0           0             1         0           0
## 745                  0           0             1         3           0
## 19                   0           0             1         0           0
## 198                  0           0             1         0           0
## 448                  0           0             1         0           0
##     FrecuenciaUrinariaA menudo
## 612                          0
## 615                          0
## 745                          0
## 19                           0
## 198                          0
## 448                          1
```

Hecho esto, dividimos una vez más el conjunto en dos partes, la de entrenamiento y la de validación. La semilla vuelve a ser mi cumpleaños.

```
tr=round(nrow(diabetes)*0.7)
set.seed(23111998)
muestra=sample.int(nrow(diabetes), tr)
Train.diab=diabetes[muestra,]
```

```
Val.diab=diabetes[-muestra,]
```

Vamos a ajustar primero el modelo para interpretar los parámetros.Normalmente en la forma estadística tradicional ajustamos el modelo completo a todos los datos sin separar los conjuntos de entrenamiento y validación.

```
gfit1=glm(Diabetico~., data=diabetes, family=binomial)
summary(gfit1)
```

```
##
## Call:
## glm(formula = Diabetico ~ ., family = binomial, data = diabetes)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5520  -0.3133  -0.1184   0.2803   2.6719
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 -5.94772    1.43330  -4.150 3.33e-05 ***
## Edad50-59                    0.73373    0.39282   1.868 0.061785 .
## Edad60 o mas                 1.93578    0.41093   4.711 2.47e-06 ***
## EdadMenos de 40             -1.58779    0.42924  -3.699 0.000216 ***
## GeneroHombre                 0.63132    0.37402   1.688 0.091424 .
## D.FamiliarSi                 1.40603    0.27270   5.156 2.52e-07 ***
## BPAltoSi                    -0.91804    0.42021  -2.185 0.028909 *
## Actividad+30min              1.11060    0.38914   2.854 0.004318 **
## ActividadNada                2.00264    0.41777   4.794 1.64e-06 ***
## Actividad1h                  1.82361    0.38812   4.699 2.62e-06 ***
## IBM                          0.07405    0.02395   3.092 0.001987 **
## TabacoSi                     0.54975    0.55624   0.988 0.322995
## AlcoholSi                    0.27826    0.37543   0.741 0.458586
## Sueño                       -0.07282    0.12362  -0.589 0.555794
## SProfundo                    0.21043    0.10979   1.917 0.055286 .
## MedicinaRegularSi            3.10103    0.33426   9.277  < 2e-16 ***
## ComidaBasuraOcasional       -0.19382    0.83175  -0.233 0.815743
## ComidaBasuraA menudo         0.44350    0.82819   0.536 0.592304
## ComidaBasuraCasi siempre     0.83968    0.99450   0.844 0.398487
## EstresNada                   0.04750    0.58358   0.081 0.935125
## EstresA veces               -0.27595    0.46552  -0.593 0.553325
## EstresCasi siempre          -0.48960    0.50960  -0.961 0.336673
## NivelBPBajo                -14.98614  693.36267  -0.022 0.982756
## NivelBPNormal               -1.62993    0.40213  -4.053 5.05e-05 ***
## Embarazos                    0.06721    0.16772   0.401 0.688627
## PdiabetesSi                  4.63130    1.03016   4.496 6.93e-06 ***
## FrecuenciaUrinariaA menudo   0.18629    0.32359   0.576 0.564808
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1108.91  on 905  degrees of freedom
## Residual deviance:  478.06  on 879  degrees of freedom
## AIC: 532.06
```

```
##
## Number of Fisher Scoring iterations: 16
```

Podemos observar que son significativos los valores para las variables *Edad60 o mas*, *EdadMenos de 40*, *D.FamiliarSi*, *ActividadNada*, *Actividad1h*, *MedicinaRegularSi*, *NivelBPNormal* y *PdiabetesSi*. Ahora ajustamos el modelo con la constante, es decir, el modelo nulo.

```
gfit0=glm(Diabetico~1, data=diabetes, family=binomial)
```

Y ahora los comparamos entre sí con una ANOVA.

```
anova(gfit0, gfit1, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: Diabetico ~ 1
## Model 2: Diabetico ~ Edad + Genero + D.Familiar + BPAlto + Actividad +
##     IBM + Tabaco + Alcohol + Sueño + SProfundo + MedicinaRegular +
##     ComidaBasura + Estres + NivelBP + Embarazos + Pdiabetes +
##     FrecuenciaUrinaria
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1       905    1108.91
## 2       879     478.06 26   630.85 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Nos da un p-valor de 2.2e-16, es decir, es altamente significativo.El modelo con todas las variables es mucho mejor que el modelo que tiene sólo la constante.

```
gfit2=glm(Diabetico~., data=Train.diab, family=binomial)
cbind(gfit1$coefficients, gfit2$coefficients)
```

```
##                                 [,1]         [,2]
## (Intercept)                -5.94771554  -5.30077258
## Edad50-59                   0.73373072   1.17899840
## Edad60 o mas                1.93577960   1.96604662
## EdadMenos de 40            -1.58778772  -1.92272304
## GeneroHombre                0.63131822   0.53306772
## D.FamiliarSi                1.40603371   1.69360099
## BPAltoSi                   -0.91804419  -0.87033938
## Actividad+30min             1.11059816   1.06061680
## ActividadNada               2.00263775   1.40100663
## Actividad1h                 1.82360614   1.35650221
## IBM                         0.07404691   0.07419303
## TabacoSi                    0.54974852   0.38522269
## AlcoholSi                   0.27826006   0.55378327
## Sueño                      -0.07282293  -0.04653857
## SProfundo                   0.21043267   0.18223914
## MedicinaRegularSi           3.10103337   2.85276055
## ComidaBasuraOcasional      -0.19381634  -0.67853500
## ComidaBasuraA menudo        0.44349638   0.26889789
## ComidaBasuraCasi siempre    0.83968058  -0.22227647
## EstresNada                  0.04750261  -0.17939875
## EstresA veces              -0.27595249  -0.48719493
## EstresCasi siempre         -0.48960418  -1.01836615
## NivelBPBajo               -14.98614201 -15.03898679
## NivelBPNormal              -1.62992969  -1.26279081
```

```
## Embarazos                    0.06720743   0.06125257
## PdiabetesSi                  4.63129510   6.11592215
## FrecuenciaUrinariaA menudo   0.18629407   0.17854788
```

Ahora procedemos a realizar la predicción.

```
p=predict(gfit2, Val.diab, type="response")
PredTarget=as.factor(p>0.5)
levels(PredTarget)=c("No", "Si")
library(caret)
```

```
## Loading required package: lattice
```

```
matrizLogis<-confusionMatrix(Val.diab$Diabetico, PredTarget)
matrizLogis
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No  Si
##         No 156  24
##         Si  17  75
##
##                Accuracy : 0.8493
##                  95% CI : (0.8011, 0.8896)
##     No Information Rate : 0.636
##     P-Value [Acc > NIR] : 4.768e-15
##
##                   Kappa : 0.6694
##
##  Mcnemar's Test P-Value : 0.3487
##
##             Sensitivity : 0.9017
##             Specificity : 0.7576
##          Pos Pred Value : 0.8667
##          Neg Pred Value : 0.8152
##              Prevalence : 0.6360
##          Detection Rate : 0.5735
##    Detection Prevalence : 0.6618
##       Balanced Accuracy : 0.8297
##
##        'Positive' Class : No
##
```

Presenta una precisión del 85%. También podemos observar la sensibilidad y la especificidad. La sensibilidad es la probabilidad de que la prueba identifique como enfermo a aquél que efectivamente lo está, en este caso es del 90%, bastante buena. La especificidad es la probabilidad de que calificar como no enfermo a alguien que no lo esté, en este caso es del 76%. Vemos que son unos resultados bastante buenos.

Ahora ajustamos el modelo para nuestros datos con el SVM con kernel *radial*, que es el que hace por defecto.

```
library(e1071)
fitsvm1 <-svm(Diabetico ~., data = Train.diab)
summary(fitsvm1)
```

```
##
## Call:
## svm(formula = Diabetico ~ ., data = Train.diab)
```

```
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##
## Number of Support Vectors:  245
##
##  ( 125 120 )
##
##
## Number of Classes:  2
##
## Levels:
##  No Si
```

```
predictedSVM = predict(fitsvm1,Val.diab)
matrizSVM1<-confusionMatrix(Val.diab$Diabetico, predictedSVM)
matrizSVM1
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No  Si
##         No 168  12
##         Si  15  77
##
##                Accuracy : 0.9007
##                  95% CI : (0.8589, 0.9336)
##     No Information Rate : 0.6728
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.7765
##
##  Mcnemar's Test P-Value : 0.7003
##
##             Sensitivity : 0.9180
##             Specificity : 0.8652
##          Pos Pred Value : 0.9333
##          Neg Pred Value : 0.8370
##              Prevalence : 0.6728
##          Detection Rate : 0.6176
##    Detection Prevalence : 0.6618
##       Balanced Accuracy : 0.8916
##
##        'Positive' Class : No
##
```

En este modelo la precisión es del 90%. La sensibilidad es del 92% y la especificidad es del 86%. Estos resultados son mejores que en el modelo anterior, el de regresión logística. Probamos con otros kernel.

```
fitsvm2 <-svm(Diabetico ~., data = Train.diab, kernel="polynomial")
predictedSVM = predict(fitsvm2,Val.diab)
matrizSVM2<-confusionMatrix(Val.diab$Diabetico, predictedSVM)
matrizSVM2
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No  Si
##         No 180   0
##         Si  67  25
##
##                Accuracy : 0.7537
##                  95% CI : (0.698, 0.8037)
##     No Information Rate : 0.9081
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.3306
##
##  Mcnemar's Test P-Value : 7.433e-16
##
##             Sensitivity : 0.7287
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.2717
##              Prevalence : 0.9081
##          Detection Rate : 0.6618
##    Detection Prevalence : 0.6618
##       Balanced Accuracy : 0.8644
##
##        'Positive' Class : No
##
```

En este caso, usando el kernel polinomial, observamos que la precision y sensibilidad han bajado, aunque la especificidad ha subido hasta el 100%, es decir, que todos los inidividuos sanos han sido calificados como tal.

```
fitsvm3 <-svm(Diabetico ~., data = Train.diab, kernel="sigmoid")
predictedSVM = predict(fitsvm3,Val.diab)
matrizSVM3<-confusionMatrix(Val.diab$Diabetico, predictedSVM)
matrizSVM3
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No  Si
##         No 159  21
##         Si  23  69
##
##                Accuracy : 0.8382
##                  95% CI : (0.789, 0.8799)
##     No Information Rate : 0.6691
##     P-Value [Acc > NIR] : 2.442e-10
##
##                   Kappa : 0.6367
##
##  Mcnemar's Test P-Value : 0.8802
##
##             Sensitivity : 0.8736
##             Specificity : 0.7667
##          Pos Pred Value : 0.8833
```

```
##          Neg Pred Value : 0.7500
##              Prevalence : 0.6691
##          Detection Rate : 0.5846
##    Detection Prevalence : 0.6618
##       Balanced Accuracy : 0.8201
##
##        'Positive' Class : No
##
```

Finalmente, con el kernel sigmoidal, nos da una precisión de 84%, con una sensibilidad del 87% y una especificidad del 77%. Tambien peores que los anteriores. Ahora pondremos la precisión de cada modelo para ver cuales son los mejores.

```
Accuracy=c(matrizLogis$overall[1], matrizSVM1$overall[1], matrizSVM2$overall[1],matrizSVM3$overall[1])
names(Accuracy)=c("Logística","SVM-radial", "SVM-polynomial", "SVM-sigmoid")
Accuracy
```

```
##      Logística     SVM-radial SVM-polynomial    SVM-sigmoid
##      0.8492647      0.9007353      0.7536765      0.8382353
```

Observamos que el modelo con mejor precisión es el radial y el que tiene peor precisión es el polinomial.

Conclusiones.

Como conclusión, observando los gráficos oportunos, podemos afirmar que la presencia de diabetes parece estar relacionada con niveles altos de BP. Tambien esta relacionada con las variables *MedicinaRegular*, ya que los diabéticos tienen que medicarse regularmente y *Edad*, por lo que es probable que la diabetes surja con la misma. El resto de variables no presentan mucha mas relación, si acaso la variable *Actividad*, que indica que las personas que no realizan deporte presentan mas casos de diabetes, si bien no es algo destacable.

En cuanto al modelo de predicción de IBM, podemos ver que hay variables muy significativas, como la edad, si hay antecedentes de diabetes en la familia, la falta de ejercicio, si orina a menudo o si presenta diabetes. Sin embargo, el modelo predictivo no era muy bueno.

Finalmente, se ha realizado el modelo para la predicción de diabetes. En este modelo se ha probado la predicción mediante la Regresión Logarítmica y usando SVM con diversos kernel. Los resultados en general han sido bastante buenos, con porcentajes de precisión, sensibilidad y especificidad bastante elevados, por encima del 75% en todos los casos. El modelo que mejores resultados ha obtenido ha sido el SVM con kernel radial.