Pràctica 2 - Neteja i anàlisi de les dades

M2.951 - Tipologia i cicle de vida de les dades

Roger Ribas Gimeno i Xabier Urra Nuin

Desembre 2021

---

1. Introducció i descripció del dataset

El conjunt de dades healthcare-dataset-stroke-data.csv l'hem extret de la plataforma Kaggle a través del següent enllaç: https://www.kaggle.com/fedesoriano/stroke-prediction-dataset (https://www.kaggle.com/fedesoriano/stroke-prediction-dataset). D'acord amb l'Organització Mundial de la Salut (OMS), l'ictus o stroke en anglès és la segona causa principal de defuncions a nivell global, responsable d'aproximadament un 11% de les morts totals.

Aquest conjunt de dades s'utilitza per predir si un pacient té probabilitats de sofrir un ictus, basant-se en paràmetres d'entrada com el sexe, l'edat, altres malalties, i si la persona és fumadora o no, entre d'altres. Cada fila o observació del conjunt de dades proporciona informació rellevant sobre el pacient. El conjunt de dades conté 5110 observacions i 12 variables.

Les variables del conjunt de dades i que s'usaran en aquesta activitat són:

- id: identificador únic
- gender: "Male", "Female" o "Other"
- age: edat del pacient
- hypertension: 0 si el pacient no té hipertensió, 1 si el pacient té hipertensió
- heart_disease: 0 si el pacient no té cap malaltia cardiovascular, 1 si el pacient té alguna malaltia cardiovascular
- ever_married: "No" si el pacient no ha estat casat, "Yes" si el pacient ha estat casat
- work_type: "children" si el pacient és un infant, "Govt_jov" si el pacient és funcionari del govern, "Never_worked" si el pacient no ha treballat mai, "Private" si treballa en l'empresa privada, o "Self-employed" si el pacient és un treballador autònom.
- Residence_type: "Rural" si el pacient viu en una zona rural, o "Urban" si el pacient viu en una zona urbana
- avg_glucose_level: nivell mitjà de glucosa en sang del pacient
- bmi: índex de massa corporal del pacient
- smoking_status: "formerly smoked" si el pacient havia fumat però ja no ho fa, "never smoked" si el pacient mai no ha fumat, "smokes" si el pacient fuma actualment, o "Unknown" si no es tenen dades.
- stroke: 1 si el pacient ja ha patit ictus o 0 si no l'ha patit.

---

2. Integració i selecció de les dades

A continuació llegim el fitxer healthcare-dataset-stroke-data.csv i guardem les dades en un objecte amb identificador denominat stroke_dataset. Seguidament, verificarem que les dades s'han carregat correctament.

```
# Carreguem el dataset tenint en compte que ja disposa d'una capçalera amb els noms dels atributs, i que
 els caràcters separadors són comes
stroke <- read.csv("healthcare-dataset-stroke-data.csv",header = TRUE,sep = ",")
attach(stroke)

# Guardem una còpia del dataset original
copiastroke <- stroke

# Vegem com és el dataset revisant-ne les primeres 10 files
head(stroke,10)
```

```
##       id gender age hypertension heart_disease ever_married     work_type
## 1   9046   Male  67            0             1          Yes       Private
## 2  51676 Female  61            0             0          Yes Self-employed
## 3  31112   Male  80            0             1          Yes       Private
## 4  60182 Female  49            0             0          Yes       Private
## 5   1665 Female  79            1             0          Yes Self-employed
## 6  56669   Male  81            0             0          Yes       Private
## 7  53882   Male  74            1             1          Yes       Private
## 8  10434 Female  69            0             0           No       Private
## 9  27419 Female  59            0             0          Yes       Private
## 10 60491 Female  78            0             0          Yes       Private
##    Residence_type avg_glucose_level  bmi  smoking_status stroke
## 1           Urban            228.69 36.6 formerly smoked      1
## 2           Rural            202.21  N/A    never smoked      1
## 3           Rural            105.92 32.5    never smoked      1
## 4           Urban            171.23 34.4          smokes      1
## 5           Rural            174.12   24    never smoked      1
## 6           Urban            186.21   29 formerly smoked      1
## 7           Rural             70.09 27.4    never smoked      1
## 8           Urban             94.39 22.8    never smoked      1
## 9           Rural             76.15  N/A         Unknown      1
## 10          Urban             58.57 24.2         Unknown      1
```

```
# Ara comprovem l'estructura del joc de dades:
str(stroke)
```

```
## 'data.frame':    5110 obs. of  12 variables:
##  $ id               : int  9046 51676 31112 60182 1665 56669 53882 10434 27419 60491 ...
##  $ gender           : chr  "Male" "Female" "Male" "Female" ...
##  $ age              : num  67 61 80 49 79 81 74 69 59 78 ...
##  $ hypertension     : int  0 0 0 0 1 0 1 0 0 0 ...
##  $ heart_disease    : int  1 0 1 0 0 0 1 0 0 0 ...
##  $ ever_married     : chr  "Yes" "Yes" "Yes" "Yes" ...
##  $ work_type        : chr  "Private" "Self-employed" "Private" "Private" ...
##  $ Residence_type   : chr  "Urban" "Rural" "Rural" "Urban" ...
##  $ avg_glucose_level: num  229 202 106 171 174 ...
##  $ bmi              : chr  "36.6" "N/A" "32.5" "34.4" ...
##  $ smoking_status   : chr  "formerly smoked" "never smoked" "never smoked" "smokes" ...
##  $ stroke           : int  1 1 1 1 1 1 1 1 1 1 ...
```

```
# Com observem al resultat, comprovem que tenim 5110 observacions o files i 12 variables o columnes. Vege
m ara el resum del dataset:
summary(stroke)
```

```
##        id           gender               age         hypertension
##  Min.   :   67   Length:5110        Min.   : 0.08   Min.   :0.00000
##  1st Qu.:17741   Class :character   1st Qu.:25.00   1st Qu.:0.00000
##  Median :36932   Mode  :character   Median :45.00   Median :0.00000
##  Mean   :36518                      Mean   :43.23   Mean   :0.09746
##  3rd Qu.:54682                      3rd Qu.:61.00   3rd Qu.:0.00000
##  Max.   :72940                      Max.   :82.00   Max.   :1.00000
##  heart_disease     ever_married        work_type         Residence_type
##  Min.   :0.00000   Length:5110        Length:5110        Length:5110
##  1st Qu.:0.00000   Class :character   Class :character   Class :character
##  Median :0.00000   Mode  :character   Mode  :character   Mode  :character
##  Mean   :0.05401
##  3rd Qu.:0.00000
##  Max.   :1.00000
##  avg_glucose_level     bmi            smoking_status         stroke
##  Min.   : 55.12    Length:5110        Length:5110        Min.   :0.00000
##  1st Qu.: 77.25    Class :character   Class :character   1st Qu.:0.00000
##  Median : 91.89    Mode  :character   Mode  :character   Median :0.00000
##  Mean   :106.15                                          Mean   :0.04873
##  3rd Qu.:114.09                                          3rd Qu.:0.00000
##  Max.   :271.74                                          Max.   :1.00000
```

3. Neteja i preparació de les dades

Tractament general de les dades

Observant la natura de les dades amb què tractem, veiem que tenim una variable objectiu (stroke), binària, i la resta són variables predictores. D'aquestes, haurem de transformar algunes d'elles. Per exemple haurem de convertir en factor variables com heart_disease o hypertension, i convertir a tipus numèric la variable bmi, la qual està emmagatzemada com a a factor.

```
# Transformem la variable bmi a numèric com hem comentat abans, i els seus valors NA els tractarem en el
 següent apartat

str(stroke$bmi)
```

```
##  chr [1:5110] "36.6" "N/A" "32.5" "34.4" "24" "29" "27.4" "22.8" "N/A" ...
```

```
stroke$bmi <- as.numeric(as.character(stroke$bmi))
str(stroke$bmi)
```

```
##  num [1:5110] 36.6 NA 32.5 34.4 24 29 27.4 22.8 NA 24.2 ...
```

```
# Transformem les variables heart_disease i hypertension a factorial com hem comentat abans

stroke$heart_disease <- factor(stroke$heart_disease, levels = c(0,1), labels =c('No','Yes'))
stroke$hypertension <- factor(stroke$hypertension,levels = c(0,1), labels = c('No','Yes'))
str(stroke$heart_disease)
```

```
##  Factor w/ 2 levels "No","Yes": 2 1 2 1 1 1 2 1 1 1 ...
```

```
str(stroke$hypertension)
```

```
##  Factor w/ 2 levels "No","Yes": 1 1 1 1 2 1 2 1 1 1 ...
```

```
head(stroke,4)
```

```
##      id gender age hypertension heart_disease ever_married     work_type
## 1  9046   Male  67           No           Yes          Yes       Private
## 2 51676 Female  61           No            No          Yes Self-employed
## 3 31112   Male  80           No           Yes          Yes       Private
## 4 60182 Female  49           No            No          Yes       Private
##   Residence_type avg_glucose_level  bmi  smoking_status stroke
## 1          Urban            228.69 36.6 formerly smoked      1
## 2          Rural            202.21   NA    never smoked      1
## 3          Rural            105.92 32.5    never smoked      1
## 4          Urban            171.23 34.4          smokes      1
```

Valors absents

Depenent del conjunt de dades amb què tractem, s'utilitzen uns o altres mètodes per indicar els valors absents. Hi ha datasets en els quals aquests valors són representats per zeros, per interrogants, o també pot ser que hi hagi camps que continguin elements buits. Com comprovem a continuació, no és el cas ja que no trobem elements buits en cap de les columnes del nostre conjunt de dades.

```
library(dplyr)

sapply(stroke, function(x) sum(is.na(x)))
```

```
##               id           gender              age     hypertension
##                0                0                0                0
##    heart_disease     ever_married        work_type   Residence_type
##                0                0                0                0
## avg_glucose_level              bmi   smoking_status           stroke
##                0              201                0                0
```

Amb una ullada ràpida al fitxer de dades, observem que hi ha observacions amb la categoria "N/A" en la variable 'bmi', referent a l'índex de massa corporal. Comprovarem la proporció d'observacions que tenen valors absents i treurem conclusions sobre què hem de fer amb aquestes dades, si eliminarles o bé si podem aplicar algun mètode d'imputació de valors.

```
nrow(filter(stroke, is.na(bmi)))
```

```
## [1] 201
```

Veiem que tenim 201 observacions amb valors absents a la columna 'bmi'. Decidim que el fet d'esborrar aquestes observacions que contenen aquestes categories "N/A" ens faria perdre informació rellevant, i per tant és millor emprar un mètode d'imputació de valors, com ara la mitjana dels valors numèrics, excloent els NA, de la columna d'índex de massa corporal. Per poder fer això, substituirem aquests NA pel valor de la mitjana comentat abans i finalment comprovarem que, un cop aplicada aquesta tècnica d'imputació, no queda cap valor NA en el dataset.

```
stroke$bmi[is.na(stroke$bmi)] <- mean (as.numeric(stroke$bmi),na.rm=TRUE )
sapply(stroke, function(x) sum(is.na(x)))
```

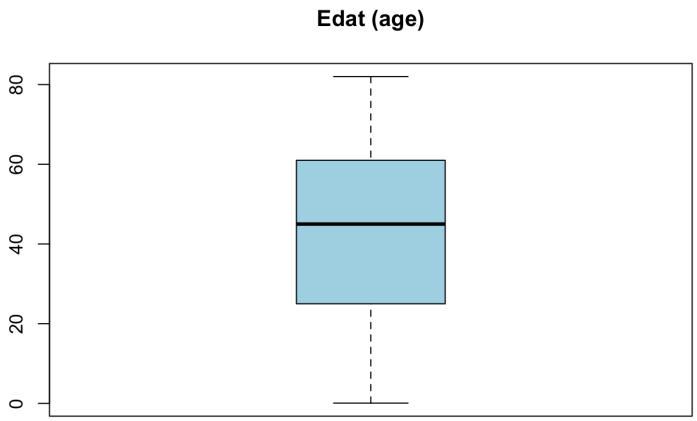
```
##               id           gender              age     hypertension
##                0                0                0                0
##    heart_disease     ever_married        work_type   Residence_type
##                0                0                0                0
## avg_glucose_level              bmi   smoking_status           stroke
##                0                0                0                0
```

Valors extrems

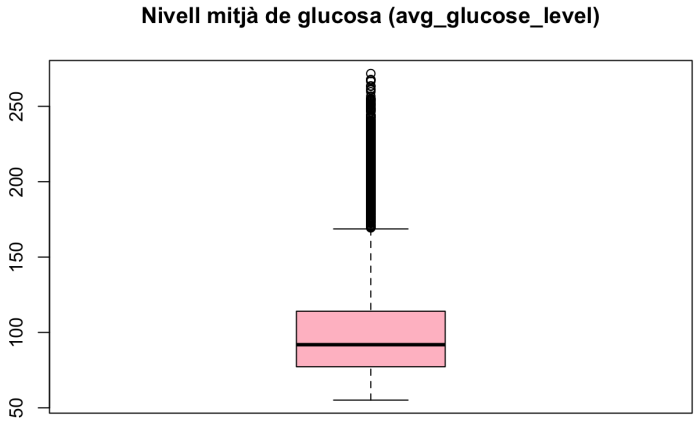
Els valos extrems, també anomenats valors atípics o outliers, són observacions que es troben a una distància força allunyada de la majoria de les altres observacions en una mateixa població de dades.

Per identificar-ne, si n'hi ha, utilitzarem gràfiques de tipus boxplot, les quals detecten outliers com tots aquells valors més enllà dels anomenats bigotis. Aquests són les línies que es determinen com el tercer quartil + 1.5 vegades el rang interquartílic (tercer quartil menys el primer quartil) i el primer quartil - 1.5 vegades el rang interquartílic. Analitzem els valors extrems en les variables numèriques:

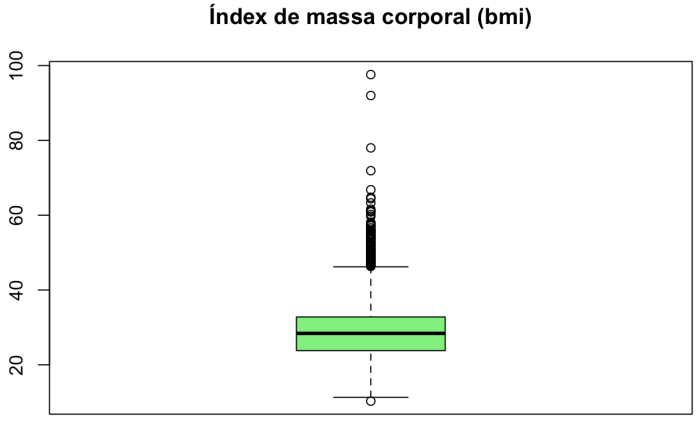
```
primer_plot <- boxplot(stroke$age,
        main = "Edat (age)",
        boxwex = 0.5,col="light blue")
```




```
segon_plot <- boxplot(stroke$avg_glucose_level,
        main = "Nivell mitjà de glucosa (avg_glucose_level)",
        boxwex = 0.5,col="pink")
```




```
tercer_plot <- boxplot(as.numeric(stroke$bmi),
        main = "Índex de massa corporal (bmi)",
        boxwex = 0.5,col="light green")
```



Veiem com la variable Edat no presenta outliers, mentre que les altres dues variables sí que en tenen, com ja anticipàvem en els histogrames anteriors. No obstant, decidim que no els eliminarem ni els tractarem perquè es tracta de valors biològicament plausibles que a més a méspoden contenir informació molt important per identificar els casos amb ictus.

Pel que fa a la variable sobre fumadors/no-fumadors:

```
unique(stroke$smoking_status)
```

```
## [1] "formerly smoked" "never smoked"    "smokes"          "Unknown"
```

```
nrow(filter(stroke, smoking_status=="formerly smoked"))
```

```
## [1] 885
```

```
nrow(filter(stroke, smoking_status=="never smoked"))
```

```
## [1] 1892
```

```
nrow(filter(stroke, smoking_status=="smokes"))
```

```
## [1] 789
```

```
nrow(filter(stroke, smoking_status=="Unknown"))
```

```
## [1] 1544
```

Veiem que hi ha un percentatge molt alt d'observacions en la categoria "Unknown", és a dir que se'n desconeix l'status de fumador o no fumador. Però com que es tracta de 1544 observacions (aprox. 30%), no les eliminarem, ho deixarem tal com està.

Per acabar aquest apartat de valors extrems, ens fixem en la variable de gènere:

```
unique(stroke$gender)
```

```
## [1] "Male"   "Female" "Other"
```

```
nrow(filter(stroke, gender=="Male"))
```

```
## [1] 2115
```

```
nrow(filter(stroke, gender=="Female"))
```

```
## [1] 2994
```

```
nrow(filter(stroke, gender=="Other"))
```

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

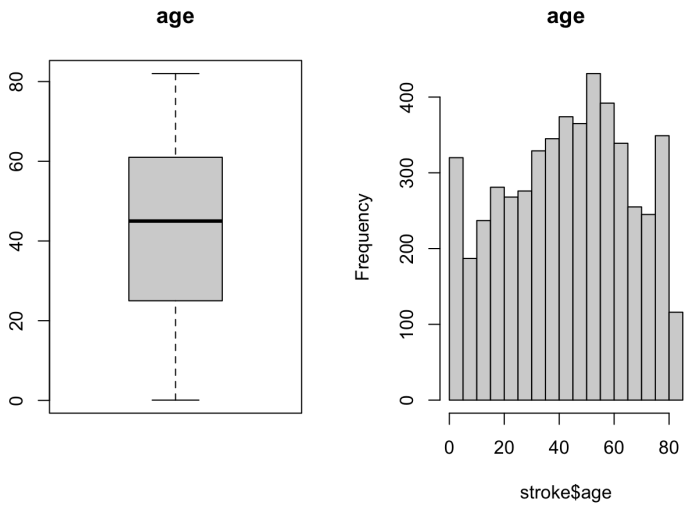
Veiem que hi ha una categoria de "Other", que podria fer referència a un gènere no binari, però en aquest cas, com que només es disposa d'una observació sota aquest gènere, podem determinar que es tracta d'un outlier, i decidim eliminar la fila.

```
stroke <- stroke[-which(stroke$gender=='Other'),]
dim(stroke)
```

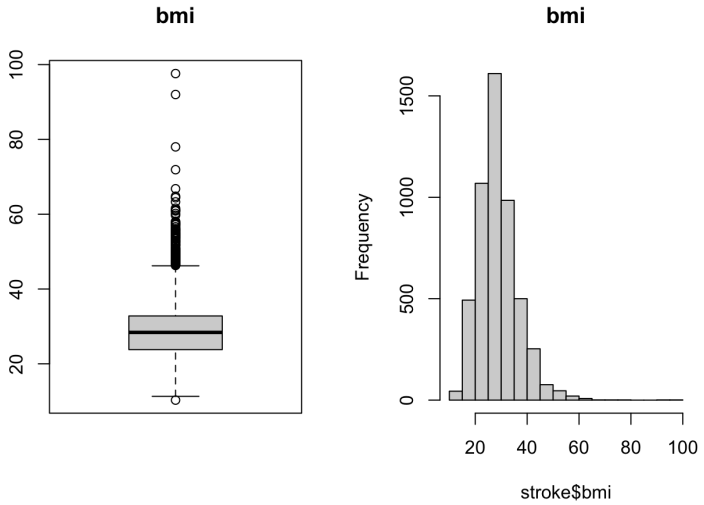
```
## [1] 5109   12
```

Normalitat: realitzem una inspecció visual de normalitat de les variables quantitatives

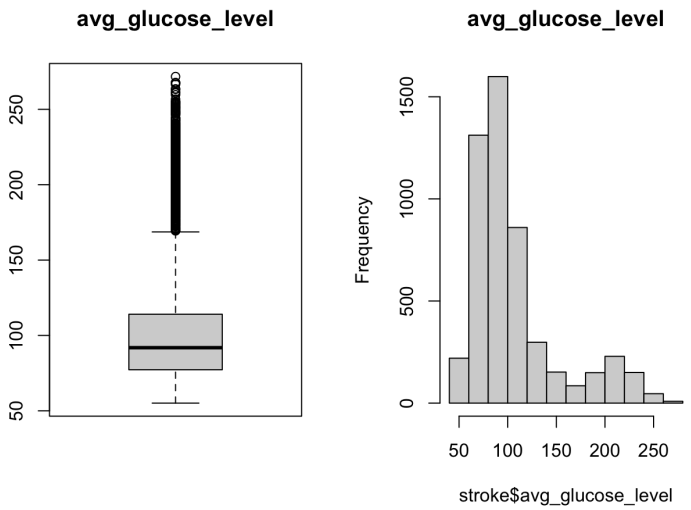
```
par(mfrow=c(1,2))
boxplot(stroke$age, main="age")
hist(stroke$age, main="age")
```




```
par(mfrow=c(1,2))
boxplot(stroke$bmi, main="bmi")
hist(stroke$bmi, main="bmi")
```




```
par(mfrow=c(1,2))
boxplot(stroke$avg_glucose_level, main="avg_glucose_level")
hist(stroke$avg_glucose_level, main="avg_glucose_level")
```

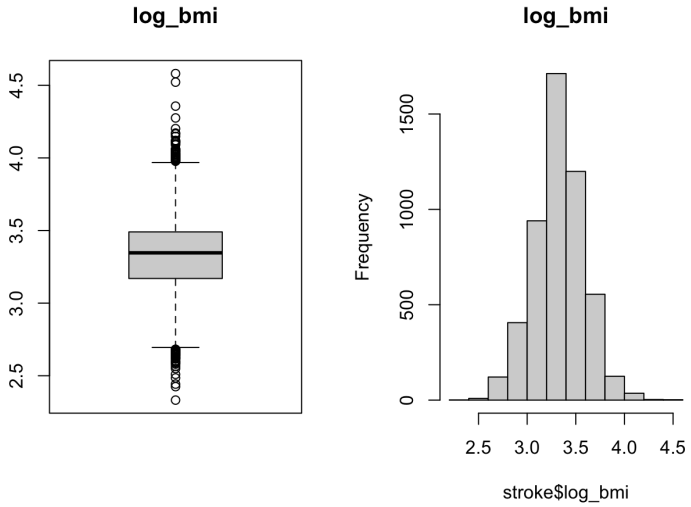


Els resultats del diagrama de caixes i de l'histograma suggereixen que les variables *bmi* i *avg_glucose_level* no tenen una distribució normal, sinó una marcada asimetria positiva.

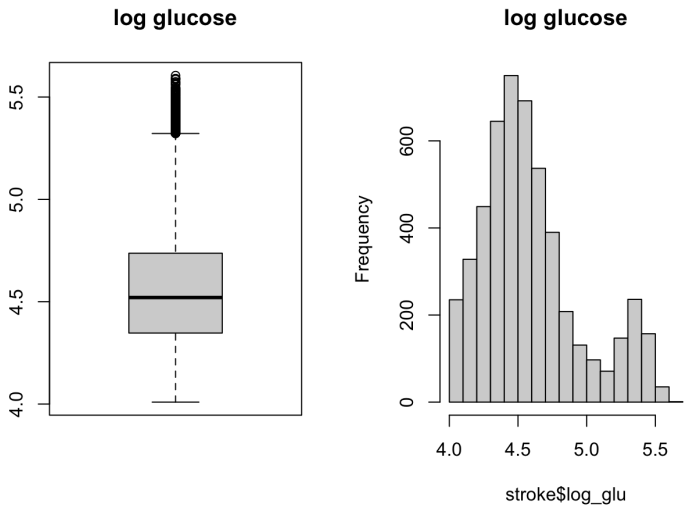
Provem de corregir-ho amb una transformació logarítmica i mostrem els resultats:

```
stroke$log_bmi=log(stroke$bmi)
stroke$log_glu=log(stroke$avg_glucose_level)
```

```
par(mfrow=c(1,2))
boxplot(stroke$log_bmi, main="log_bmi")
hist(stroke$log_bmi, main="log_bmi")
```




```
par(mfrow=c(1,2))
boxplot(stroke$log_glu, main="log glucose")
hist(stroke$log_glu, main="log glucose")
```



La transformació logarítmica ha resolt parcialment l'asimetria, especialment per la variable *bmi*.

Completem l'anàlisi de normalitat amb el contrast de normalitat de Lilliefors:

```
# Carreguem la llibreria nortest:
if (!require('nortest')) install.packages('nortest'); library(nortest)
```

```
## Loading required package: nortest
```

```
# Executem el nest de normalitat:
lillie.test(stroke$age)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  stroke$age
## D = 0.05075, p-value < 2.2e-16
```

```
lillie.test(stroke$log_bmi)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  stroke$log_bmi
## D = 0.034054, p-value = 8.735e-15
```

```
lillie.test(stroke$log_glu)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  stroke$log_glu
## D = 0.10283, p-value < 2.2e-16
```

El resultat altament significatiu ens confirma que cap d'aquestes variables no segueix una distribució normal. Malgrat això, serà un problema menor considerant la gran mida del joc de dades.

4. Anàlisi de les dades

Anàlisi exploratòria

Primer de tot procedim a efectuar una anàlisi exploratòria de les dades. Ens desfem de la columna id ja que no ens aportarà res per a aquest anàlisi exploratori.

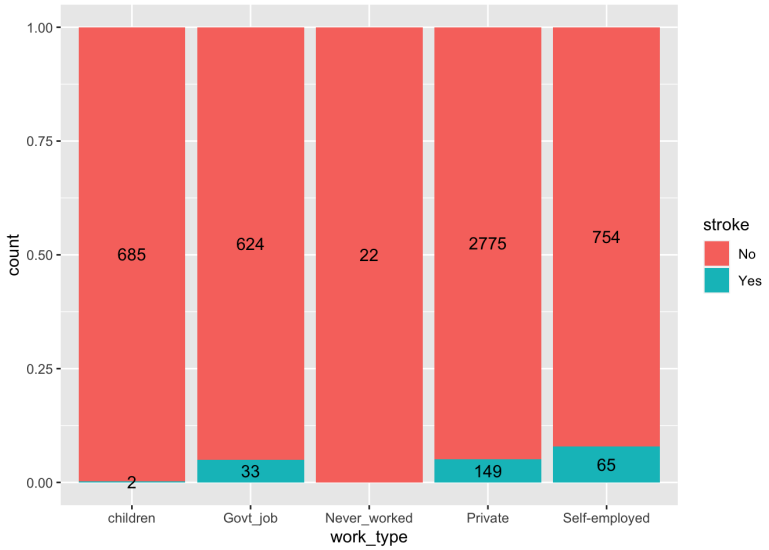
```
#install.packages("GGally")
#install.packages("mice")
#install.packages("ROSE")
library(ggplot2)
library(caret)
library(mice)
library(GGally)
library(dplyr)
library(ROSE)
library(randomForest)
library(e1071)

stroke<-stroke[2:14]
stroke$stroke <- as.character(stroke$stroke)
stroke$stroke <- gsub('1', 'Yes', stroke$stroke)
stroke$stroke <- gsub('0', 'No', stroke$stroke)
```

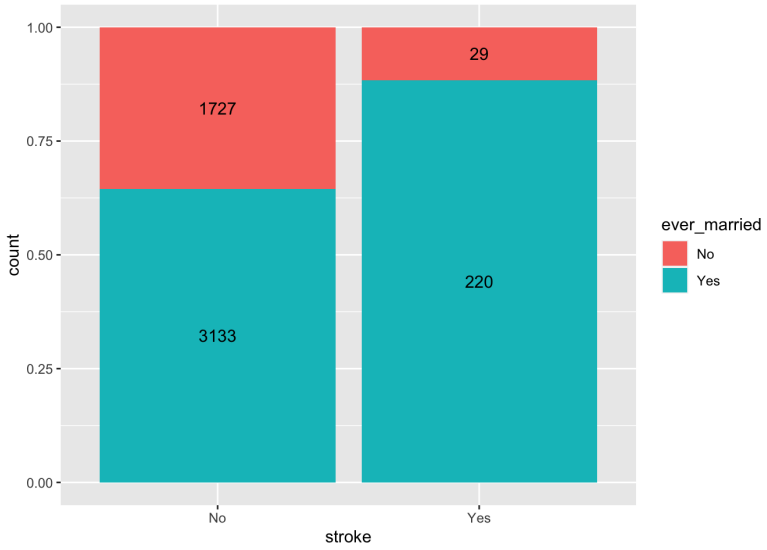
Amb això, ens diposem a analitzar la relació de diverses variables amb la variable stroke.

Primer de tot ho fem amb les variables categòriques:

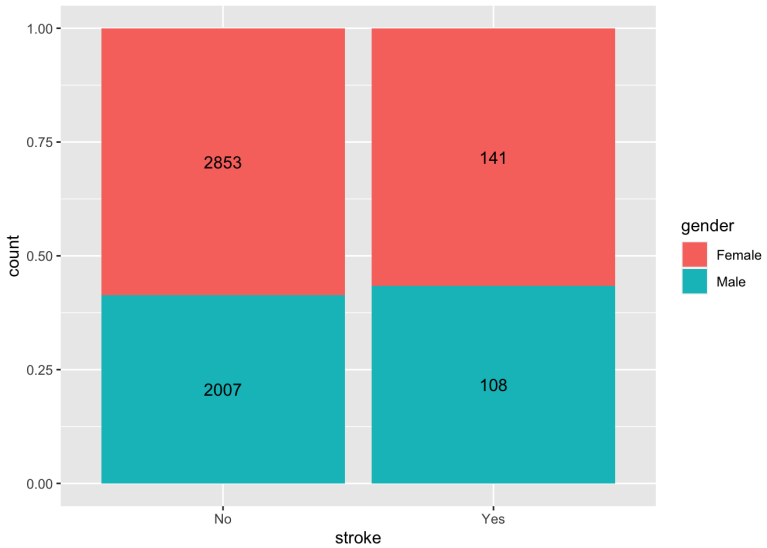
```
g1<-ggplot(stroke, aes(x = work_type, fill = stroke))+
  geom_bar(position = "fill")+
  stat_count(geom = "text",
             aes(label = stat(count)),
             #position = "fill", color = "black"
             position = position_fill(vjust = 0.5), color = "black")
g1
```



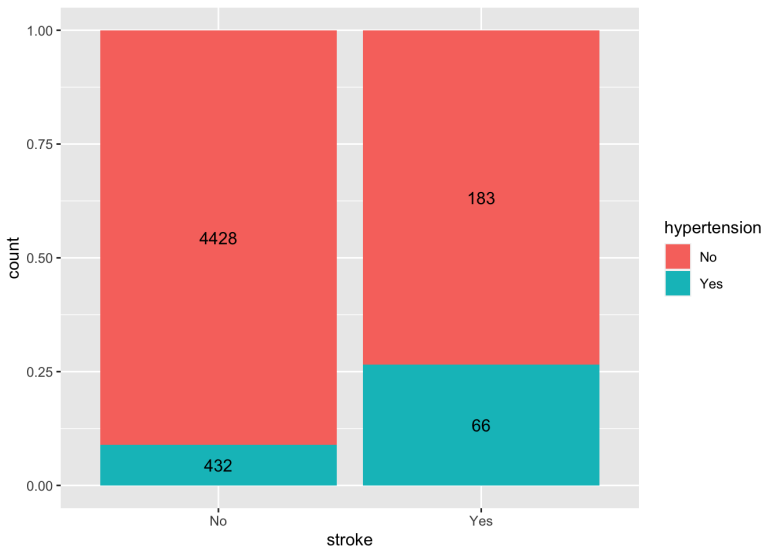

```
g2<-ggplot(stroke, aes(x = stroke, fill = ever_married))+
  geom_bar(position = "fill")+
  stat_count(geom = "text",
             aes(label = stat(count)),
             #position = "fill", color = "black"
             position = position_fill(vjust = 0.5), color = "black")
g2
```



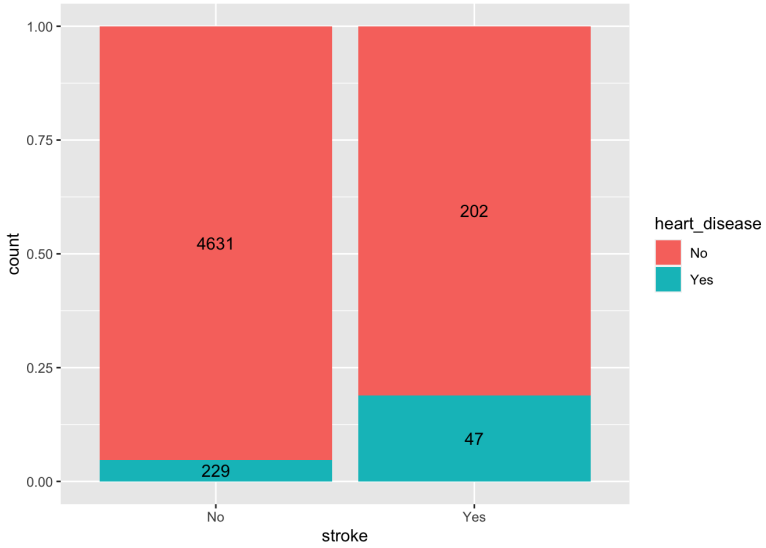

```
g3<-ggplot(stroke, aes(x = stroke, fill = gender))+
  geom_bar(position = "fill")+
  stat_count(geom = "text",
             aes(label = stat(count)),
             #position = "fill", color = "black"
             position = position_fill(vjust = 0.5), color = "black")
g3
```



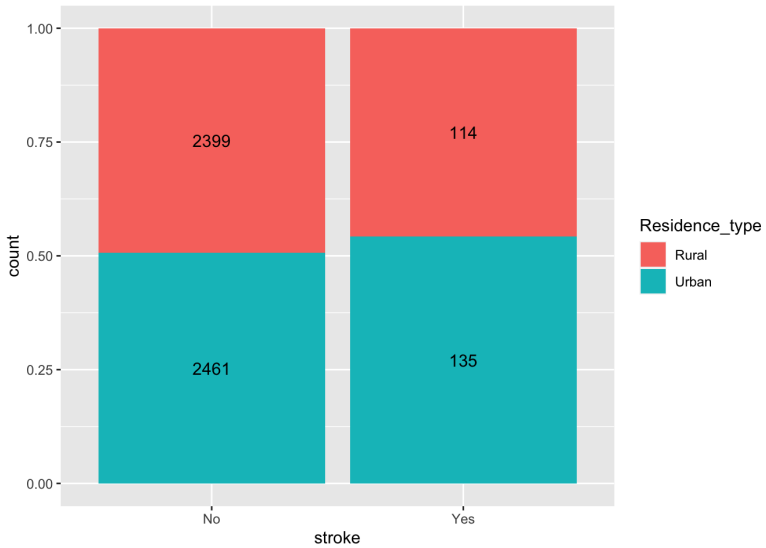

```
g4<-ggplot(stroke, aes(x = stroke, fill = hypertension))+
  geom_bar(position = "fill")+
  stat_count(geom = "text",
             aes(label = stat(count)),
             #position = "fill", color = "black"
             position = position_fill(vjust = 0.5), color = "black")
g4
```



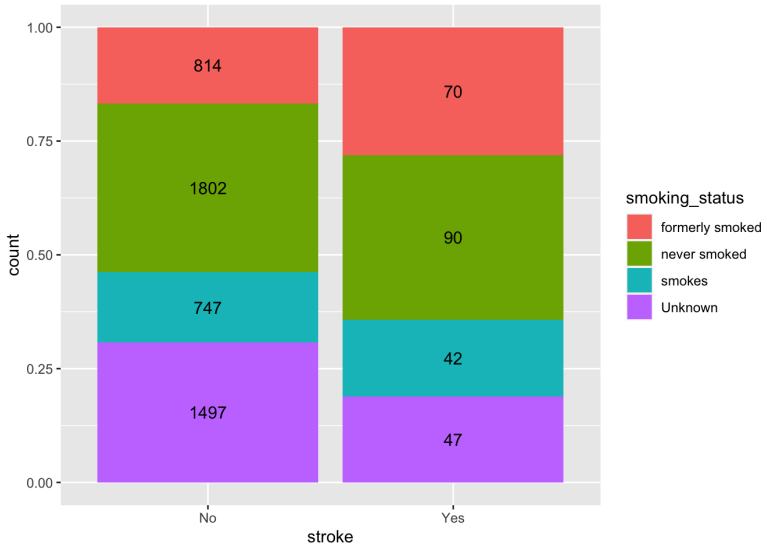
```
g5<-ggplot(stroke, aes(x = stroke, fill = heart_disease))+
  geom_bar(position = "fill")+
  stat_count(geom = "text",
             aes(label = stat(count)),
             #position = "fill", color = "black"
             position = position_fill(vjust = 0.5), color = "black")
g5
```




```
g6<-ggplot(stroke, aes(x = stroke, fill = Residence_type))+
  geom_bar(position = "fill")+
  stat_count(geom = "text",
             aes(label = stat(count)),
             #position = "fill", color = "black"
             position = position_fill(vjust = 0.5), color = "black")
g6
```



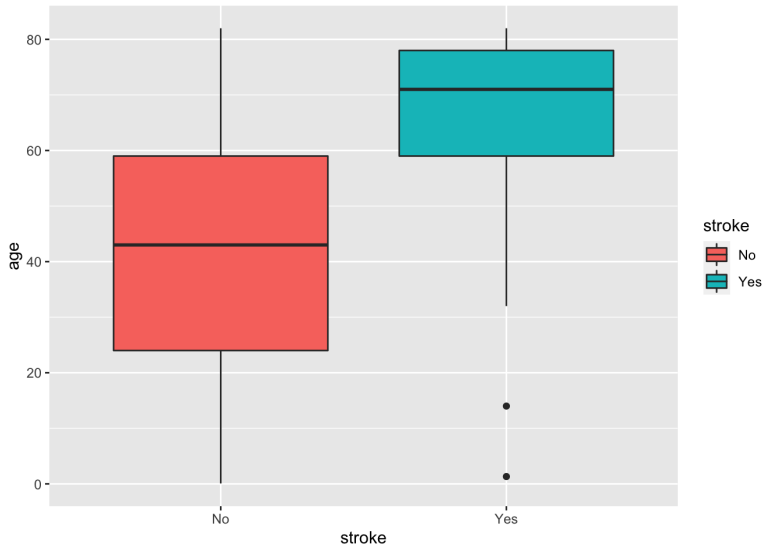

```
g7<-ggplot(stroke, aes(x = stroke, fill = smoking_status))+
  geom_bar(position = "fill")+
  stat_count(geom = "text",
             aes(label = stat(count)),
             #position = "fill", color = "black"
             position = position_fill(vjust = 0.5), color = "black")
g7
```



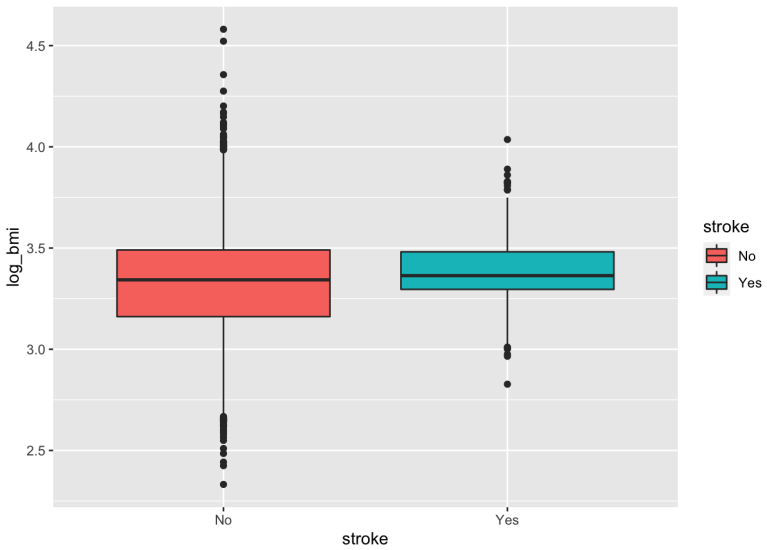
Veiem que les prinipals diferències de proporció d'ictus es troben als atributs *ever_married*, *hypertension* i *heart_disease*.

Explorem a continuació les variables quantitatives:

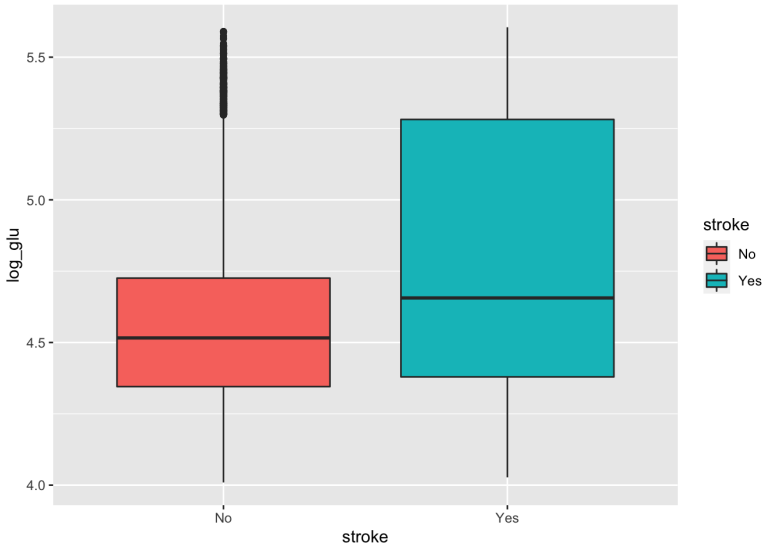
```
g8<-ggplot(data = stroke, aes(x = stroke, y = age, fill = stroke))+geom_boxplot()
g8
```




```
g9<-ggplot(data = stroke, aes(x = stroke, y = log_bmi, fill = stroke))+geom_boxplot()
g9
```



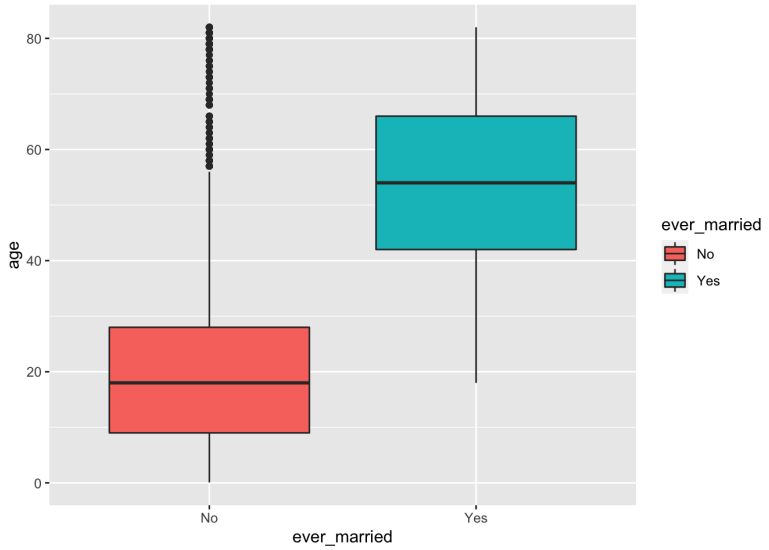

```
g10<-ggplot(data = stroke, aes(x = stroke, y = log_glu, fill = stroke))+geom_boxplot()
g10
```



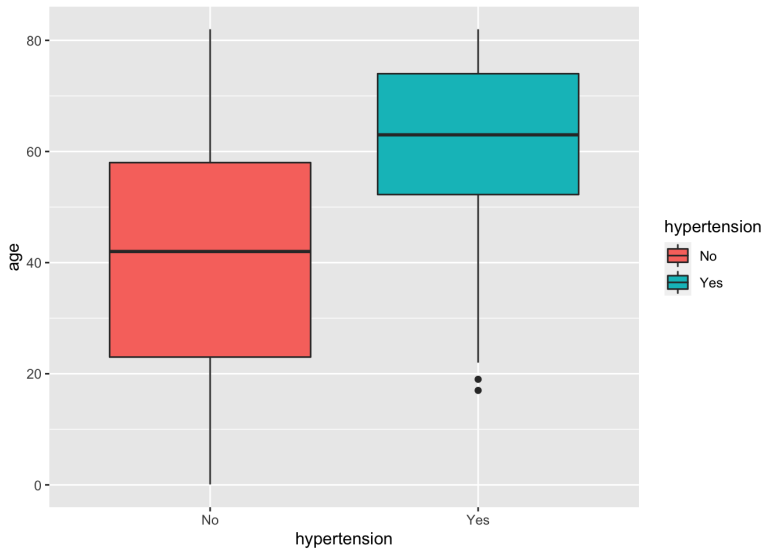
Veiem que la diferència més significativa es dona per l'atribut *age*, sent els pacients que desenvolupen un ictus molt més grans que els que no el desenvolupen. També sembla haver-hi certa diferència en les xifres mitges de glucosa.

Donada la gran diferència d'edat entre pacients amb i sense ictus, explorem si poden haver-hi diferències d'edat en les variables categòriques que semblen associar-se al risk d'ictus (*hipertensió*, *ever_married* i *heart_disease*):

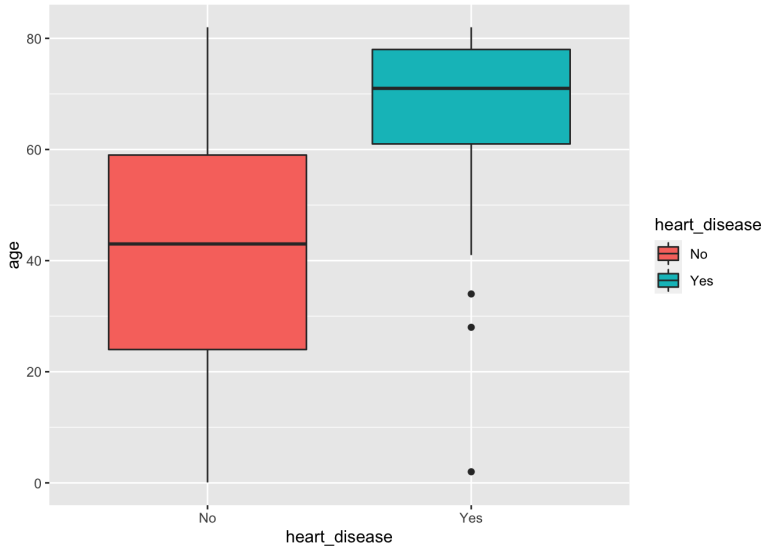
```
g11<-ggplot(data = stroke, aes(x = ever_married, y = age, fill = ever_married))+geom_boxplot()
g11
```




```
g12<-ggplot(data = stroke, aes(x = hypertension, y = age, fill = hypertension))+geom_boxplot()
g12
```




```
g13<-ggplot(data = stroke, aes(x = heart_disease, y = age, fill = heart_disease))+geom_boxplot()
g13
```



Efectivament, veiem que totes aquestes variables s'asocien a una major edat.

Finalment calculem la matriu de correlacions entre variables numèriques:

```
stroke_num <- select(stroke, "age", "log_bmi", "log_glu")
res <- cor(stroke_num)
round(res, 2)
```

```
##          age log_bmi log_glu
## age     1.00    0.39    0.21
## log_bmi 0.39    1.00    0.16
## log_glu 0.21    0.16    1.00
```

Hi ha una correlació moderada entre l'edat i el bmi, motiu pel qual podria ser preferible no incloure ambdues variables als models de regressió que farem a continuació.

Split en conjunt d'entrenament i test

Abans d'iniciar el modelatge, separem el dataset en conjunt d'entrenament i conjunt de test. Dividirem el conjunt de dades en un conjunt d'entrenament (3/4) i un conjunt de test (1/4). Seleccionem el conjunt de dades de la variable classificadora i el de la resta de variables:

```
# Desarem el joc de dades com un arxiu format csv:
write.csv(stroke,'joc_analitzat.csv')

# Seleccionem les variables dependent i independents:
set.seed(666)
y <- stroke[,"stroke"]
X <- stroke[,!names(stroke) %in% c("stroke", "bmi", "avg_glucose_level")]
```

Creem un rang utilitzant el paràmetre *split_prop* (en aquest cas = 4):

```
split_prop <- 4
indexes = sample(1:nrow(stroke), size=floor(((split_prop-1)/split_prop)*nrow(stroke)))
trainX<-X[indexes,]
trainy<-y[indexes]
testX<-X[-indexes,]
testy<-y[-indexes]
```

Fem una anàlisi de dades mínim per a assegurar-nos de no obtenir classificadors esbiaixats pels valors que conté cada mostra. Verificarem que la proporció de valors "Yes" i "No" de *stroke* és semblant en els dos conjunts:

```
summary(trainy)
```

```
##    Length     Class      Mode
##      3831 character character
```

```
summary(testy)
```

```
##    Length     Class      Mode
##      1278 character character
```

Un cop confirmat que la proporció de valors és semblant, explorem també que no hi hagi diferències significatives a les variables independents:

```
summary(trainX);
```

```
##     gender              age        hypertension heart_disease
##  Length:3831        Min.   : 0.08   No :3455     No :3638
##  Class :character   1st Qu.:25.00   Yes: 376     Yes: 193
##  Mode  :character   Median :45.00
##                     Mean   :43.32
##                     3rd Qu.:61.00
##                     Max.   :82.00
##  ever_married        work_type         Residence_type     smoking_status
##  Length:3831        Length:3831        Length:3831        Length:3831
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##     log_bmi         log_glu
##  Min.   :2.332   Min.   :4.012
##  1st Qu.:3.172   1st Qu.:4.341
##  Median :3.350   Median :4.520
##  Mean   :3.334   Mean   :4.591
##  3rd Qu.:3.497   3rd Qu.:4.732
##  Max.   :4.522   Max.   :5.605
```

```
summary(testX)
```

```
##     gender              age        hypertension heart_disease
##  Length:1278        Min.   : 0.24   No :1156     No :1195
##  Class :character   1st Qu.:25.00   Yes: 122     Yes:  83
##  Mode  :character   Median :44.00
##                     Mean   :42.95
##                     3rd Qu.:61.00
##                     Max.   :82.00
##  ever_married        work_type         Residence_type     smoking_status
##  Length:1278        Length:1278        Length:1278        Length:1278
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##     log_bmi         log_glu
##  Min.   :2.549   Min.   :4.010
##  1st Qu.:3.158   1st Qu.:4.355
##  Median :3.336   Median :4.523
##  Mean   :3.317   Mean   :4.598
##  3rd Qu.:3.475   3rd Qu.:4.753
##  Max.   :4.581   Max.   :5.590
```

Modelatge amb Regressió Logística

Utilitzarem un model de regressió logística per dur a terme la classificació, atès que la variable objectiu (*stroke*) és categòrica binària i les variables predictores contenen valors tant categòrics com numèrics. Com que els models de regressió logística s'avaluen amb el criteri d'informació d'Akaike (AIC), compararem amb aquest fins a 4 models diferents creats amb els següents criteris:

1- Model més senzill que només inclou la variable edat, que sembla tan important a l'anàlisi exploratòria de les dades:

```
trainy = as.factor(trainy)
mrlog1 <- glm(trainy ~ age, data = trainX, family = binomial)
summary(mrlog1)
```

```
##
## Call:
## glm(formula = trainy ~ age, family = binomial, data = trainX)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.7580  -0.3223  -0.1766  -0.0795   3.8100
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -7.358226   0.397743  -18.50   <2e-16 ***
## age          0.076316   0.005834   13.08   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1470.3  on 3830  degrees of freedom
## Residual deviance: 1190.5  on 3829  degrees of freedom
## AIC: 1194.5
##
## Number of Fisher Scoring iterations: 7
```

2- Model que inclou també les xifres de glucosa perquè també semblen estar associades amb el resultat i no eren correlacionades amb l'edat:

```
# Logistics Regression
mrlog2 <- glm(trainy ~ age + log_glu, data = trainX, family = binomial)
summary(mrlog2)
```

```
##
## Call:
## glm(formula = trainy ~ age + log_glu, family = binomial, data = trainX)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9214  -0.3189  -0.1730  -0.0780   3.8421
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -10.109584   0.902847 -11.197  < 2e-16 ***
## age           0.073760   0.005916  12.468  < 2e-16 ***
## log_glu       0.618747   0.180226   3.433 0.000597 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1470.3  on 3830  degrees of freedom
## Residual deviance: 1178.9  on 3828  degrees of freedom
## AIC: 1184.9
##
## Number of Fisher Scoring iterations: 7
```

3- Model que inclou totes les variables categòriques i quantitatives que semblen estar associades a l'ictus a la inspecció de les dades:

```
# Logistics Regression
mrlog3 <- glm(trainy ~ age + log_glu + ever_married + hypertension + heart_disease, data = trainX, family
= binomial)
summary(mrlog3)
```

```
##
## Call:
## glm(formula = trainy ~ age + log_glu + ever_married + hypertension +
##     heart_disease, family = binomial, data = trainX)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2457  -0.3111  -0.1681  -0.0873   3.7462
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -9.43322    0.91914 -10.263  < 2e-16 ***
## age               0.07141    0.00605  11.804  < 2e-16 ***
## log_glu           0.54609    0.18340   2.978  0.00291 **
## ever_marriedYes  -0.39208    0.24211  -1.619  0.10536
## hypertensionYes   0.54245    0.18531   2.927  0.00342 **
## heart_diseaseYes  0.31888    0.22605   1.411  0.15834
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1470.3  on 3830  degrees of freedom
## Residual deviance: 1166.1  on 3825  degrees of freedom
## AIC: 1178.1
##
## Number of Fisher Scoring iterations: 7
```

4- Model que inclou totes les variables:

```
# Logistics Regression
mrlog4 <- glm(trainy ~ age + log_glu + ever_married + hypertension + heart_disease + log_bmi + smoking_st
atus + Residence_type + gender + work_type, data = trainX, family = binomial)
summary(mrlog4)
```

```
##
## Call:
## glm(formula = trainy ~ age + log_glu + ever_married + hypertension +
##     heart_disease + log_bmi + smoking_status + Residence_type +
##     gender + work_type, family = binomial, data = trainX)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1989  -0.3073  -0.1603  -0.0923   3.4970
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 -8.244713   1.569918  -5.252 1.51e-07 ***
## age                          0.075152   0.006803  11.047  < 2e-16 ***
## log_glu                      0.547591   0.188417   2.906  0.00366 **
## ever_marriedYes             -0.347350   0.250395  -1.387  0.16538
## hypertensionYes              0.565591   0.188203   3.005  0.00265 **
## heart_diseaseYes             0.300196   0.230708   1.301  0.19319
## log_bmi                     -0.110823   0.412314  -0.269  0.78810
## smoking_statusnever smoked  -0.154777   0.202507  -0.764  0.44469
## smoking_statussmokes        -0.001316   0.261682  -0.005  0.99599
## smoking_statusUnknown       -0.056846   0.244989  -0.232  0.81651
## Residence_typeUrban          0.133652   0.161921   0.825  0.40914
## genderMale                  -0.033169   0.166068  -0.200  0.84169
## work_typeGovt_job           -1.018857   0.875773  -1.163  0.24468
## work_typeNever_worked      -10.677531 351.161784  -0.030  0.97574
## work_typePrivate            -1.048229   0.857940  -1.222  0.22178
## work_typeSelf-employed      -1.254756   0.883417  -1.420  0.15551
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1470.3  on 3830  degrees of freedom
## Residual deviance: 1161.7  on 3815  degrees of freedom
## AIC: 1193.7
##
## Number of Fisher Scoring iterations: 14
```

Veiem que els millors models semblen el primer i l'últim. Comprovarem ara la seva precisió al conjunt de dades de test:

```
glm.probs <- predict (mrlog1, newdata = testX, type = "response")
glm.pred <- ifelse(glm.probs > 0.5, "Yes", "No")
table(glm.pred, testy)
```

```
##         testy
## glm.pred   No  Yes
##       No 1212   66
```

```
mean(glm.pred == testy)
```

```
## [1] 0.9483568
```

El model 1 encerta el 94,8% dels casos, amb la limitació de que només prediu "no ictus", és a dir, erra en tots els casos d'ictus.

```
glm.probs2 <- predict (mrlog4, newdata = testX, type = "response")
glm.pred2 <- ifelse(glm.probs2 > 0.5, "Yes", "No")
table(glm.pred2, testy)
```

```
##          testy
## glm.pred2   No  Yes
##        No 1212   66
```

```
mean(glm.pred2 == testy)
```

```
## [1] 0.9483568
```

Veiem que el model 4 té exactament la mateixa precisió.

Per aquest motiu, té sentit escollir el model més senzill. Calculem ara l'associació entre edat i ictus en terme d'Odds Ratio:

```
# Logistics Regression
testy = as.factor(testy)
mrlog5 <- glm(testy ~ age, data = testX, family = binomial)

exp(coefficients(mrlog5))
```

```
## (Intercept)         age
## 0.001019766 1.073019114
```

Per cada any addicional el risk d'ictus s'incrementa en un 7%.

Modelatge amb Arbres de decisió:

Donada l'important limitació de manca de sensibilitat del model anterior, comprovarem la utilitat d'un arbre de decisió.

Creem l'arbre de decisió usant les dades d'entrenament:

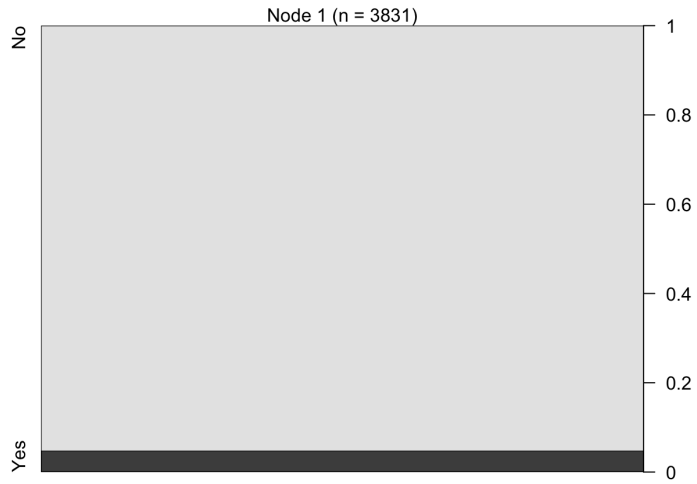
```
trainy = as.factor(trainy)
model <- C50::C5.0(trainX, trainy)
summary(model)
```

```
##
## Call:
## C5.0.default(x = trainX, y = trainy)
##
##
## C5.0 [Release 2.07 GPL Edition]      Mon Jan  3 18:52:03 2022
## -------------------------------
##
## Class specified by attribute `outcome'
##
## Read 3831 cases (11 attributes) from undefined.data
##
## Decision tree:
##  No (3831/183)
##
##
## Evaluation on training data (3831 cases):
##
##      Decision Tree
##    ----------------
##    Size      Errors
##
##       1  183( 4.8%)   <<
##
##
##     (a)   (b)    <-classified as
##    ----  ----
##    3648          (a): class No
##     183          (b): class Yes
##
##
## Time: 0.0 secs
```

L'arbre obtingut classifica erròniament 3648 dels 3831 casos donats, una taxa d'error del 4,8%.

A continuació, procedim a mostrar l'arbre obtingut.

```
model <- C50::C5.0(trainX, trainy)
plot(model)
```



Novament, sembla que aquest model adjudica a tots els casos la classe "no ictus", motiu pel qual erra en els casos positius.

Validem el model amb les dades reservades (test):

```
predicted_model <- predict( model, testX, type="class" )
print(sprintf("La precissió de l'arbre és: %.4f %%",100*sum(predicted_model == testy) / length(predicted_
model)))
```

```
## [1] "La precissió de l'arbre és: 94.8357 %"
```

Analitzem la qualitat de la predicció mitjançant una matriu de confusió que ens facilitarà la interpretació de la sensibilitat i especificitat de les prediccions:

```
if (!require('gmodels')) install.packages('gmodels'); library(gmodels)
```

```
## Loading required package: gmodels
```

```
CrossTable(testy, predicted_model,prop.chisq  = FALSE, prop.c = FALSE, prop.r =FALSE,dnn = c('Reality',
'Prediction'))
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  1278
##
##
##              | predicted_model
##        testy |        No | Row Total |
## -------------|-----------|-----------|
##           No |      1212 |      1212 |
##              |     0.948 |           |
## -------------|-----------|-----------|
##          Yes |        66 |        66 |
##              |     0.052 |           |
## -------------|-----------|-----------|
## Column Total |      1278 |      1278 |
## -------------|-----------|-----------|
##
##
```

A les dades de test, sembla que la precisió és força alta, semblant a la de les dades d'entrenament. Malgrat això, novament, a la matriu de confusió veiem que donat que el resultat "no ictus" és més freqüent, els errors del model són falsos negatius, és a dir, el model prediu que no tindran ictus pacients que si que el pateixen. Això és novament una limitació important per l'eina que estem cercant.

Per tant, explorarem finalment si és possible minimitzar aquest error. Per fer el model més sensible, jugarem amb el paràmetre cost del mètode C5.0 [1,2]. Especificarem quins errors s'han d'evitar, i per això creem una matriu de costos, a la qual indiquem que els falsos negatius ponderen 100 vegades més que els falsos positius:

```
matrix_dimensions <- list(c("Yes", "No"), c("Yes", "No"))
names(matrix_dimensions) <- c("reference", "prediction")
error_cost100 <- matrix(c(0, 1, 100, 0), nrow = 2, dimnames = matrix_dimensions)
error_cost100
```

```
##          prediction
## reference Yes  No
##       Yes   0 100
##       No    1   0
```

Creem el model mantenint la resta de paràmetres:

```
if (!require('modeldata')) install.packages('modeldata'); library(modeldata)
```

```
## Loading required package: modeldata
```

```
if (!require('C50')) install.packages('C50'); library(C50)
```

```
## Loading required package: C50
```

```
model2 <- C50::C5.0(trainX, trainy, control = C5.0Control(), trials = 10, costs = error_cost100)
summary(model2)
```

```
##
## Call:
## C5.0.default(x = trainX, y = trainy, trials = 10, control =
##  C5.0Control(), costs = error_cost100)
##
##
## C5.0 [Release 2.07 GPL Edition]      Mon Jan  3 18:52:04 2022
## -------------------------------
##
## Class specified by attribute `outcome'
##
## Read 3831 cases (11 attributes) from undefined.data
## Read misclassification costs from undefined.costs
##
## -----  Trial 0:  -----
##
## Decision tree:
##  No (3831/183)
##
## -----  Trial 1:  -----
##
## Decision tree:
##  No (94073.7/91337.3)
##
## *** boosting reduced to 1 trial since last classifier is very inaccurate
##
## *** boosting abandoned (too few classifiers)
##
##
## Evaluation on training data (3831 cases):
##
##         Decision Tree
##    ----------------------
##    Size      Errors   Cost
##
##       1  183( 4.8%)   0.05   <<
##
##
##     (a)   (b)    <-classified as
##    ----  ----
##    3648          (a): class No
##     183          (b): class Yes
##
##
## Time: 0.0 secs
```

Veiem com són les prediccions del model 2:

```
predicted_model2 <- predict( model2, testX, type="class" )
print(sprintf("La precissió de l'arbre és: %.4f %%",100*sum(predicted_model2 == testy) / length(predicted
_model2)))
```

```
## [1] "La precissió de l'arbre és: 94.8357 %"
```

```
CrossTable(testy, predicted_model2,prop.chisq  = FALSE, prop.c = FALSE, prop.r =FALSE,dnn = c('Reality',
'Prediction'))
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  1278
##
##
##              | predicted_model2
##        testy |        No | Row Total |
## -------------|-----------|-----------|
##           No |      1212 |      1212 |
##              |     0.948 |           |
## -------------|-----------|-----------|
##          Yes |        66 |        66 |
##              |     0.052 |           |
## -------------|-----------|-----------|
## Column Total |      1278 |      1278 |
## -------------|-----------|-----------|
##
##
```

Veiem que malgrat aquest intent de millorar la sensibilitat de l'arbre, els resultats són idèntics i no hem pogut millorar la precisió, específicament la capacitat de predir ictus.

---

5. Conclusions

En aquesta pràctica hem treballat amb un joc de dades obtingut a la web Kaggle de pacients amb ictus i sense ictus amb la intenció final de trobar models que puguin predir quins subjectes patiran un ictus, estudiant diversos atributs basals demogràfics, socioeconòmics i biològics.

Després de la neteja de dades, hem fet una avaluació visual de les mateixes, observant certes associacions amb el risc d'ictus. L'associació més robusta es veu amb l'edat dels subjectes, i és possible que les associacions amb altres variables puguin estar confoses per diferències d'edat.

Finalment, hem intentat construir models de classificació que puguin predir el risc d'ictus. Hem creat models de regressió logística i arbres de decisió que en tots els casos han obtingut una precisió alta però amb una sensibilitat nul·la per l'ictus, segurament degut a un important inbalanç en la distribució de casos (ictus) i controls. Es podria plantejar treballar amb un subconjunt del joc de dades en què la proporció d'ictus i controls fos semblant, encara que aquest exercici tindria la limitació de no reflectir bé la distribució real de casos i controls.

Bibliografia

[1] Package 'C50' [en línia]. [Data de consulta: 22 de desembre de 2021]. Disponible a: https://cran.r-project.org/web/packages/C50/C50.pdf (https://cran.r-project.org/web/packages/C50/C50.pdf)

[2] David García Sabaté. Arboles de decisión C5.0 [en línia]. [Data de consulta: 22 de desembre de 2021]. Disponible a: https://rpubs.com/DavidGS/c50 (https://rpubs.com/DavidGS/c50)