Pràctica 2 Tipologia i cicle de vida de les dades

Joan Ginard Miquel Piña

17 - 12 - 2020

Contents

1 Descripció del dataset

Hem triat un joc de dades disponible al conegut repositori de dades d' UCI, en concret el que du per nom "diabetes-130-US hospitals for years 1999-2008 Data set", a algunes bandes anomenat diabetes-130. S'ha de distingir aquest joc de dades sobre diabetis d'un altre prou conegut que es diu PIMA ó PIMA-INDIANS

El joc de dades conté 101766 observacions i 50 variables i es corresponen a dades de pacients ingressats a un Hospital i que pateixen diabetis, però aquesta afecció no és, en general, la causa de l'ingrés. Es a dir, el joc de dades fa seguiment d'un grup de pacients que a banda del motiu principal de l'ingrés tenen diabetis.

Les dades procedeixen de diferents hospital dels EUA i van ser recollides entre els anys 1999 i 2008.

L'**objectiu**, el problema que es vol resoldre, és ser capaç de trobar un model que ens pugui predir els pacients que tornaran a ingressar.

Els atributs que component el dataset són els següents (tot i que traduïm els valors que poden prendre els atributs, estan en anglès a l'original):

1. **encounter_id**: identificador de la visita/ingrés.
2. **patient_nbr**: identificador del pacient.
3. **race**: Origen ètnic del pacient: Caucàsic, Asiàtic, Africà-Americà, Hispà i altres.
4. **gender**: masculí, femení i desconegut/no-vàlid.
5. **age**: Edat agrupada en intervals de 10 anys.
6. **weight**: Pes del pacient mesurat en lliures.
7. **admission_type_id**: Codi (nº enter) que identifica el motiu d'admissió, per exemple, urgència, programat, nou-nat.
8. **discharge_disposition_id**: Codi (nº enter) que identifica el motiu d'abandonament de l'hospital, per exemple, alta a casa, exitus, alta amb atenció mèdica.
9. **admission_source_id**: Codi (nº enter) que identifica la font d'admissió, per exemple, urgència, trasllat d'un altre hospital, recomanació mèdica.
10. **time_in_hospital**: Número de dies des de l'ingrés fins que abandona l'hospital
11. **payer_code**: Codi (nº enter) que identifica el tipus d'assegurança del pacient
12. **medical_specialty**: Especialitat del metge que atendrà al pacient: cardiologia, cirurgia...
13. **num_lab_procedures**: Número de proves de laboratori que es fan al pacient durant l'estada.
14. **num_procedures**: Número de procediments, diferents de test de laboratori, que es fan al pacient durant l'estada.
15. **num_medications**: Número de medicacions *diferents* que s'administren al pacient durant l'estada.
16. **number_outpatient**: Número de visites mèdiques del pacient l'any anterior a l'ingrés.
17. **number_emergency**: Número de visites del pacient a urgències l'any anterior a l'ingrés.
18. **number_inpatient**: Número d'ingressos hospitalaris del pacient l'any anterior a l'ingrés
19. **diag_1**, **diag_2** i **diag_3**: Són tres atributs corresponents a diagnòstics del pacient, diag_1 és el diagnòstic primari i els altres dos diagnòstics secundaris. Estan codificats segons uns codis estandarditzats.
20. **number_diagnoses**: Número de diagnòstics que consten al sistema.
21. **max_glu_serum**: Resultats del test de glucosa. Expressat com "normal", ">200", ">300" o "none" si no es va fer o no consta la mesura
22. **A1Result**: Resultat de la prova de hemoglobina glicosilada. Expressat en valors com ">8" (major del 8%), ">7" (que indica major que 7% però menor que 8%), "normal" i "none" si no es va fer o no consta.
23. **Conjunt de MEDICACIONS**: (Atributs 25 al 47) Al dataset ara tenim tot un seguit de medicacions relacionades amb la diabetes i on figura si la dosi es puja, es manté o es baixa o si no es pren el medicament. Com veurem més endavant la major part de medicacions apenes si tenen variació i en molts casos els pacients no en prenen.
24. **change**: Indica si va haver algun canvi en les medicacions relacionades amb la diabetis.
25. **diabetesMed** : Indica si el pacient pren alguna medicació contra la diabetis.
26. **readmitted**: Dies fins a la readmissió del pacient a l'hospital. Dividit en tres valors: "<30" si reingressa en menys de 30 dies, ">30", si tarda més de 30 dies i "No" en cas de que no hagi registre de readmissió. Aquest seguiment es va fer durant 1 any, per tant un ingrés de ">30" és un ingrés entre un mes després i un any després de l'alta.

Els codis als que fan referència els atributs 7, 8 i 9 es troben a un arxiu separat anomenat *IDs_mapping.csv* i que es descarrega juntament amb el dataset a la pàgina indicada.

Els diagnòstics (atributs *diag_1*, *diag_2* i *diag_3*) estan codificats per tres xifres (i si és necessari un punt i una o dues xifres més) segons un manual de codificació estàndard i que per exemple es pot consultar aquí

Donada la naturalesa de les dades hem consultat moltes informacions a metges del nostre entorn, per tal d'aclarir o entendre alguns detalls.

2 Integració i selecció de les dades d'interès a analitzar

Recordem que el nostre objectiu es tractar de predir el possible reingrés dels pacients s a partir de les diferents dades que tenim.

Donada l'enorme quantitat de variables que tenim està clar que haurem de realitzar una selecció dels atributs a considerar però part d'aquesta selecció s'haurà de realitzar durant el procés de neteja de les dades, donat que, com veurem, atributs que *a priori* resulten molt importants com el pes estan bàsicament buits. No obstant ja podem veure que dades com l'identificador del pacient o de la visita/ingrés no resultaran interessants, ni tampoc l'assegurança del pacient.

També farem constrast d'hipòtesis per veure si determinats factors com el sexe o la medicació de la diabetis presenten una proporció major de reingrés.

Una altra cosa a considerar és la divisió de la variable readmissió en tres factors: menys de 30 dies, més de 30 i no readmissió. Habitualment la divisió és menys de 30 dies o no readmissió, donat que si la readmissió és posterior al mes no sempre es considera relacionada.

3 Neteja de les dades

Procedim a carregar les dades del csv i veiem els diferents atributs

```
#Carreguem les dades

dataDiabetes <- read.csv("dataset_diabetes/diabetic_data.csv")

#Observem els camps que tenim
str(dataDiabetes)
```

```
## 'data.frame':    101766 obs. of  50 variables:
##  $ encounter_id            : int  2278392 149190 64410 500364 16680 35754 55842 63768 12522 15738 ..
##  $ patient_nbr             : int  8222157 55629189 86047875 82442376 42519267 82637451 84259809 11488
##  $ race                    : chr  "Caucasian" "Caucasian" "AfricanAmerican" "Caucasian" ...
##  $ gender                  : chr  "Female" "Female" "Female" "Male" ...
##  $ age                     : chr  "[0-10)" "[10-20)" "[20-30)" "[30-40)" ...
##  $ weight                  : chr  "?" "?" "?" "?" ...
##  $ admission_type_id       : int  6 1 1 1 1 2 3 1 2 3 ...
##  $ discharge_disposition_id: int  25 1 1 1 1 1 1 1 1 3 ...
##  $ admission_source_id     : int  1 7 7 7 7 2 2 7 4 4 ...
##  $ time_in_hospital        : int  1 3 2 2 1 3 4 5 13 12 ...
##  $ payer_code              : chr  "?" "?" "?" "?" ...
##  $ medical_specialty       : chr  "Pediatrics-Endocrinology" "?" "?" "?" ...
##  $ num_lab_procedures      : int  41 59 11 44 51 31 70 73 68 33 ...
##  $ num_procedures          : int  0 0 5 1 0 6 1 0 2 3 ...
##  $ num_medications         : int  1 18 13 16 8 16 21 12 28 18 ...
##  $ number_outpatient       : int  0 0 2 0 0 0 0 0 0 0 ...
##  $ number_emergency        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ number_inpatient        : int  0 0 1 0 0 0 0 0 0 0 ...
##  $ diag_1                  : chr  "250.83" "276" "648" "8" ...
##  $ diag_2                  : chr  "?" "250.01" "250" "250.43" ...
##  $ diag_3                  : chr  "?" "255" "V27" "403" ...
##  $ number_diagnoses        : int  1 9 6 7 5 9 7 8 8 8 ...
##  $ max_glu_serum           : chr  "None" "None" "None" "None" ...
##  $ A1Cresult               : chr  "None" "None" "None" "None" ...
##  $ metformin               : chr  "No" "No" "No" "No" ...
##  $ repaglinide             : chr  "No" "No" "No" "No" ...
```

```
##  $ nateglinide             : chr  "No" "No" "No" "No" ...
##  $ chlorpropamide          : chr  "No" "No" "No" "No" ...
##  $ glimepiride             : chr  "No" "No" "No" "No" ...
##  $ acetohexamide           : chr  "No" "No" "No" "No" ...
##  $ glipizide               : chr  "No" "No" "Steady" "No" ...
##  $ glyburide               : chr  "No" "No" "No" "No" ...
##  $ tolbutamide             : chr  "No" "No" "No" "No" ...
##  $ pioglitazone            : chr  "No" "No" "No" "No" ...
##  $ rosiglitazone           : chr  "No" "No" "No" "No" ...
##  $ acarbose                : chr  "No" "No" "No" "No" ...
##  $ miglitol                : chr  "No" "No" "No" "No" ...
##  $ troglitazone            : chr  "No" "No" "No" "No" ...
##  $ tolazamide              : chr  "No" "No" "No" "No" ...
##  $ examide                 : chr  "No" "No" "No" "No" ...
##  $ citoglipton             : chr  "No" "No" "No" "No" ...
##  $ insulin                 : chr  "No" "Up" "No" "Up" ...
##  $ glyburide.metformin     : chr  "No" "No" "No" "No" ...
##  $ glipizide.metformin     : chr  "No" "No" "No" "No" ...
##  $ glimepiride.pioglitazone: chr  "No" "No" "No" "No" ...
##  $ metformin.rosiglitazone : chr  "No" "No" "No" "No" ...
##  $ metformin.pioglitazone  : chr  "No" "No" "No" "No" ...
##  $ change                  : chr  "No" "Ch" "No" "Ch" ...
##  $ diabetesMed             : chr  "No" "Yes" "Yes" "Yes" ...
##  $ readmitted              : chr  "NO" ">30" "NO" "NO" ...
```

Sembla que els camps buits s'assenyalen amb un "?". Aprofitem per canviar els "?" per NAs

```
dataDiabetes[dataDiabetes == "?"] <- NA
```

IDs de pacients i visites

Com hem comentat els identificadors dels pacients no ens resulten interessants però abans d'eliminar els atributs, hem d'observar si tenim pacients repetits.

```
length(unique(dataDiabetes$patient_nbr))
```

```
## [1] 71518
```

Veiem que efectivament tenim 101766 registres però nomes 71518 pacients. Aquestes visites o ingressos repetits no es poden considerar independents que és una de les suposicions habituals de molts models, per tant ens quedem només amb una ocurrència de cada pacient i eliminem els camps d'identificació i de tipus d'assegurança del pacient

```
#Aprofitem per passar les dades a un nou dataframe amb el que farem feina
dades <- dataDiabetes[!duplicated(dataDiabetes$patient_nbr),]

dades$encounter_id <- NULL
dades$patient_nbr <- NULL
dades$payer_code <- NULL
```

3.1 Tractament de zeros i elements buits

Fem una exploració dels camps que tenen valors buits

```
colSums(is.na(dades)) #Número de valors a cada columna
```

```
##                     race                   gender                      age
##                     1948                        0                        0
##                   weight        admission_type_id  discharge_disposition_id
##                    68665                        0                        0
##      admission_source_id         time_in_hospital         medical_specialty
##                        0                        0                    34477
##       num_lab_procedures           num_procedures           num_medications
##                        0                        0                        0
##        number_outpatient         number_emergency         number_inpatient
##                        0                        0                        0
##                   diag_1                   diag_2                   diag_3
##                       11                      294                     1225
##         number_diagnoses            max_glu_serum               A1Cresult
##                        0                        0                        0
##                metformin              repaglinide              nateglinide
##                        0                        0                        0
##           chlorpropamide              glimepiride            acetohexamide
##                        0                        0                        0
##                glipizide                glyburide              tolbutamide
##                        0                        0                        0
##             pioglitazone            rosiglitazone                  acarbose
##                        0                        0                        0
##                miglitol             troglitazone               tolazamide
##                        0                        0                        0
##                  examide              citoglipton                  insulin
##                        0                        0                        0
##      glyburide.metformin      glipizide.metformin glimepiride.pioglitazone
##                        0                        0                        0
##  metformin.rosiglitazone   metformin.pioglitazone                   change
##                        0                        0                        0
##              diabetesMed              readmitted
##                        0                        0
```

PES I ESPECIALITAT

Observem que el pes està gairebé completament buit i medical_speciality està buit a mitges.

Tot i que es veu prou bé mostrem el percentatge de valors buits en aquests dos casos.

```
perc <- 100*sum(is.na(dades$weight))/nrow(dades)
print(sprintf("El percentatge de valors buits al pes és: %.2f%%", perc))
```

```
## [1] "El percentatge de valors buits al pes és: 96.01%"
```

```
perc<- 100*sum(is.na(dades$medical_specialty))/nrow(dades)
print(sprintf("El percentatge de valors buits a l'especialitat és: %.2f%%", perc))
```

```
## [1] "El percentatge de valors buits a l'especialitat és: 48.21%"
```

El pes és un atribut amb un 96% de valors buits, cosa que resulta molt estranya perquè la diabetis en adults està molt associada a l'excés de pes i l'obesitat. No obstant l'article original que estudia el joc de dades (Strack et al. (2014), veure bilbiografia) assenyala que fins al 2009 els hospitals no estaven obligats a recollir

aquesta dada en un format estructurat. Potser podria haver estat un atribut important però haurem de prescindir d'ell,

D'altra banda i respecte a l'especialitat mèdica, mentre que a l'article esmentat es decideix substituir els valors nuls de l'especialitat per un valor "desconegut", nosaltres considerem que són massa valors per fer això, ja que generaria una categoria amb "dret propi" per dir-ho d'alguna manera. També hem pensat que ni sabem ni podem saber si aquests valors es corresponen sobretot a unes poques especialitats o més aviat es distribueixen més uniformement, per tant, al igual que el pes, tot i que podria ser una característica important, no podem mantenir la variable

```
dades$weight <- NULL
dades$medical_specialty <- NULL
```

DIAGNÒSTICS

Observem que tenim 11 valors absents al diagnòstic primari (*diag_1*), observem aquests valors juntament amb els diagnòstics secundaris

```
dades[is.na(dades$diag_1), c("diag_1", "diag_2", "diag_3")]
```

```
##       diag_1 diag_2 diag_3
## 519     <NA>    780    997
## 1007    <NA>    595  250.6
## 1268    <NA> 250.82    401
## 1489    <NA>    276    594
## 3198    <NA> 250.01    428
## 37694   <NA>    780    295
## 57059   <NA>    V63    414
## 57738   <NA>    276    V08
## 60315   <NA>    427    486
## 86019   <NA> 250.02    438
## 87182   <NA>   <NA>   <NA>
```

Com en aquests valors falta el diagnòstic primari, però no els secundaris, no ens queda més que concloure que són valors perduts. Podríem fer hipòtesi sobre si les dades es van introduir malament i en realitat el diagnòstic secundari és el primari, però són tan pocs valors que simplement eliminem els registres corresponents

```
dades <- dades[!is.na(dades$diag_1),]
```

Els diagnòstics secundaris poden estar presents o no, en el sentit de que un pacient pot tenir només un diagnòstic, per tant que hagi valors buits no és un problema.

RAÇA

En aquesta variable tenim 1948 valors buits que, de moment, desarem com a "desconeguts". En el següent apartat el tractarem amb més detall.

```
dades$race[is.na(dades$race)] <- "desconegut"
```

ALTRES

Un altre tipus de dades "absents" són aquelles marcades als camps "admission_type_id", "discharge_dispostion_id" i "admission_source_id" amb el codi corresponent al valor: NULL, Not Available, Unknown/Invalid...

Mirem quantes dades d'aquest tipus tenim

```
abs <- sum(dades$admission_type_id == "6" | dades$admission_type_id == "8" | dades$admission_type_id ==
print(sprintf("Dades que es poden considerar absents a admission_source_id: %i ", abs))
```

```
## [1] "Dades que es poden considerar absents a admission_source_id: 8052 "
```

```
abs <- sum(dades$discharge_disposition_id == "18" | dades$discharge_disposition_id == "25" | dades$disch
print(sprintf("Dades que es poden considerar absents a discharge_disposition: %i ", abs))
```

```
## [1] "Dades que es poden considerar absents a discharge_disposition: 3251 "
```

```
abs <- sum(dades$admission_source_id == "17" | dades$admission_source_id == "20" | dades$admission_sourc
print(sprintf("Dades que es poden considerar absents a admission_source_id: %i ", abs))
```

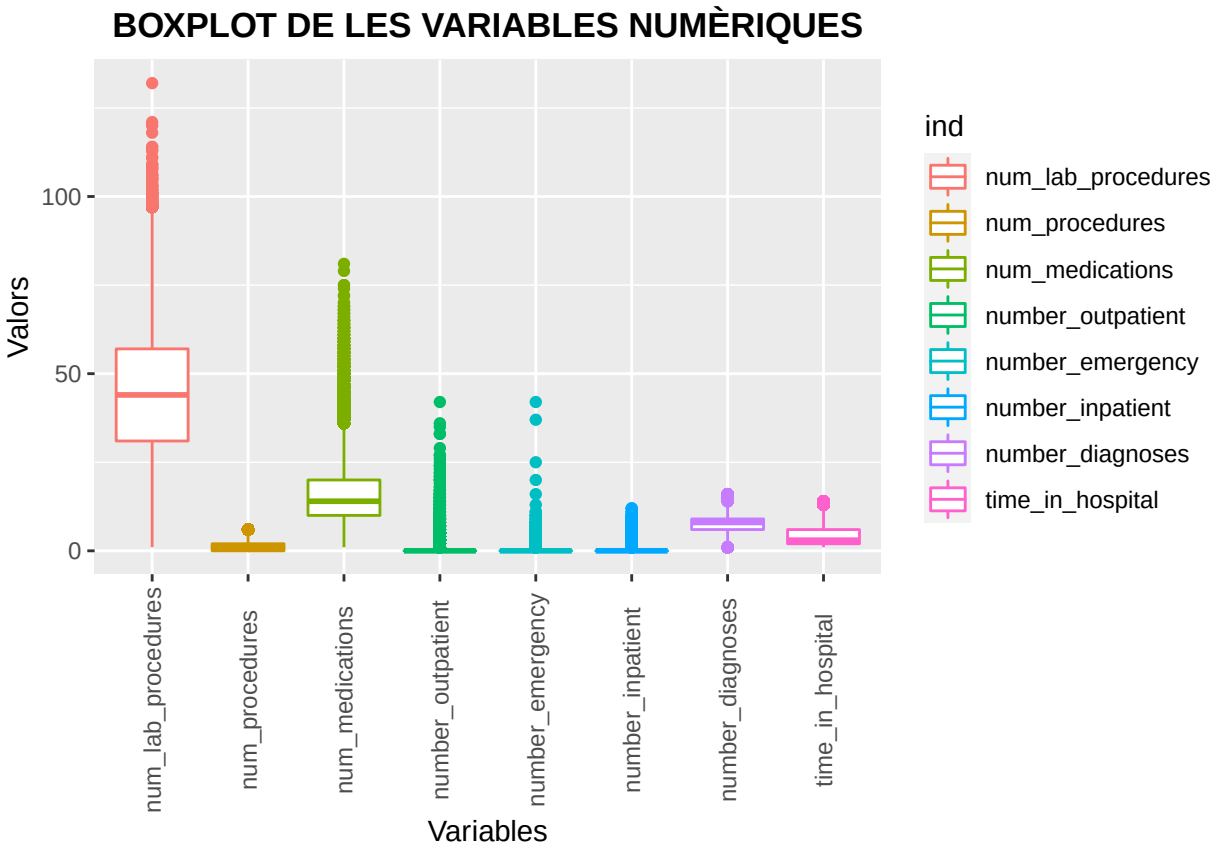
```
## [1] "Dades que es poden considerar absents a admission_source_id: 5199 "
```

És un nombre no menyspreable de dades. A una fase posterior considerarem que fem amb ells, ja que segurament haurem de fer agrupament de categories.

3.2 Identificació de valors extrems

Fem boxplots dels atributs numèrics (a tenir en compte que alguns camps són numèrics però representen codis)

```
dades_numeriques <- select(dades, c("num_lab_procedures", "num_procedures", "num_medications", "number_

#Preparem el dataframe per fer el gràfic amb ggplot
stacked_df <- stack(dades_numeriques)

#Fem la gràfica, ajustem alguns paràmetres com la ortientació del text a l'eix i els títols
ggplot(stacked_df, aes(x = ind, y = values, color = ind ))+geom_boxplot()+
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5),
        plot.title = element_text(hjust = 0.5, face = "bold"))+
  labs(title = "BOXPLOT DE LES VARIABLES NUMÈRIQUES", x = "Variables", y = "Valors")
```



Les variables *number_outpatient*, *number_emergency* i *number_inpatient* semblen tenir molts de valors a zero o al voltant i segurament per això tenen tot un continu de valors fora del boxplot.

A la resta d'atributs veiem que tenim punts fora dels "bigotis" del boxplot, però cap d'ells sembla que sigui una dada clarament errònia.

3.2.1 Altres

Tenim altres valors que no són buits o que no són extrems però que també haurem de tractar.

Al camp gènere tenim alguns valors que són desconeguts o invàlids.

```
table(dades$gender)
```

```
##
##          Female            Male Unknown/Invalid
##           38023           33481               3
```

Són tan pocs que els llevem directament de les dades

```
dades <- dades[!(dades$gender == "Unknown/Invalid"), ]
```

4 Anàlisi de les dades

4.1 Selecció de grups de dades. Planificació de l'anàlisi.

4.1.1 Reducció de la dimensionalitat.

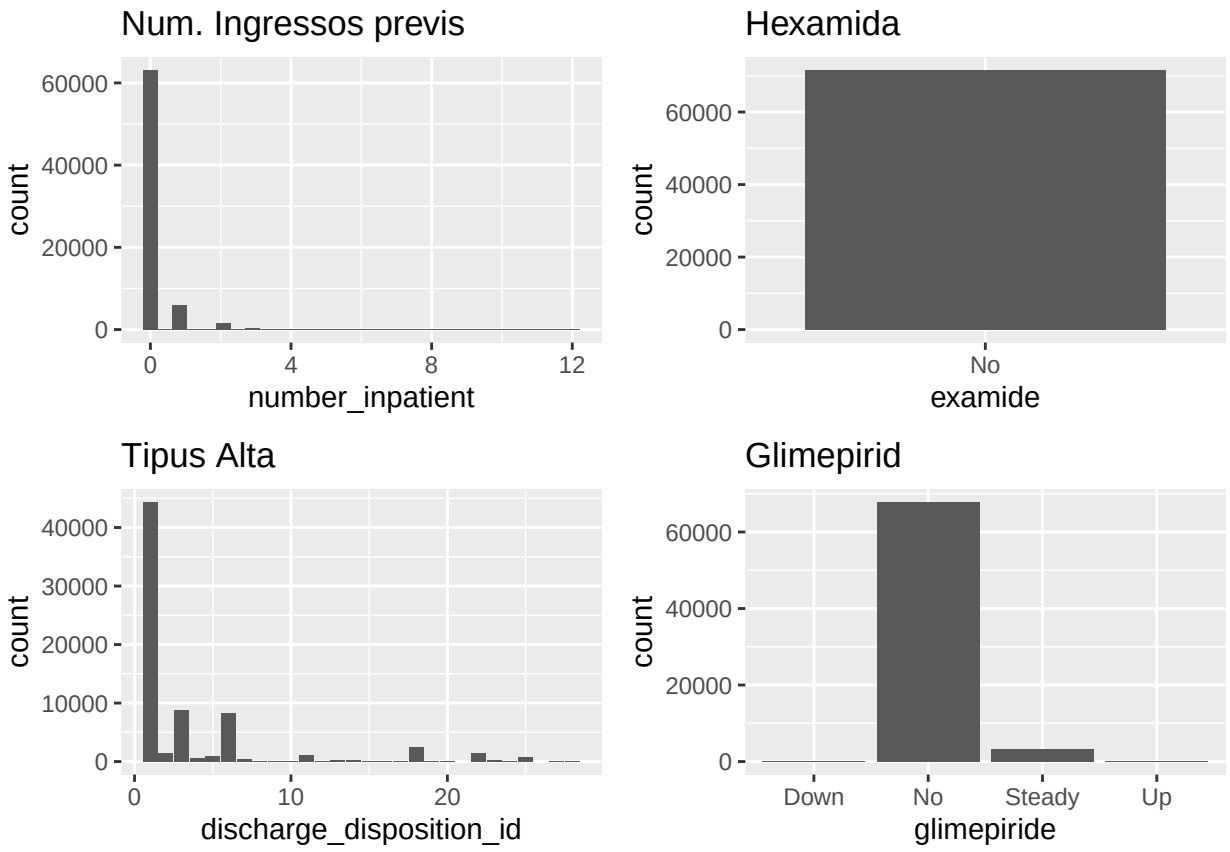
ATRIBUTS AMB POCA VARIACIÓ

En el joc de dades tenim moltes variables que gairebé no presenten variació, la gran majoria dels valors són el mateix.

Anem a veure una representació d'alguns camps per a que es vegi a que ens referim

```
p1 <- ggplot(dades, aes(x = number_inpatient))+geom_histogram()+ggtitle("Num. Ingressos previs")
p2 <- ggplot(dades, aes(x = examide))+geom_bar()+ggtitle("Hexamida")
p3 <- ggplot(dades, aes(x = discharge_disposition_id))+geom_bar()+ggtitle("Tipus Alta")
p4 <- ggplot(dades, aes(x =glimepiride))+geom_bar()+ggtitle("Glimepirid")

grid.arrange(p1,p2,p3,p4)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Això que es veu als gràfic passa a més atributs, fem servir la funció *nearzerovar()* de la llibreria *caret* per veure quins són els atributs no numèrics que bàsicament no varien i que, per tant, ja podem eliminar sense cap més anàlisi.

```
#18:45
dades_no_numeriques <- select(dades, -c("num_lab_procedures", "num_procedures", "num_medications", "numb
ZV <- nearZeroVar(dades_no_numeriques[,], saveMetrics = TRUE)
```

```r
#Imprimir aquells valors que tenen variació gairebé zero.
print(ZV[ZV$nzv == TRUE, c("zeroVar", "nzv")])
```

```
##                           zeroVar  nzv
## max_glu_serum               FALSE TRUE
## repaglinide                 FALSE TRUE
## nateglinide                 FALSE TRUE
## chlorpropamide              FALSE TRUE
## glimepiride                 FALSE TRUE
## acetohexamide               FALSE TRUE
## tolbutamide                 FALSE TRUE
## acarbose                    FALSE TRUE
## miglitol                    FALSE TRUE
## troglitazone                FALSE TRUE
## tolazamide                  FALSE TRUE
## examide                      TRUE TRUE
## citoglipton                  TRUE TRUE
## glyburide.metformin         FALSE TRUE
## glipizide.metformin         FALSE TRUE
## glimepiride.pioglitazone     TRUE TRUE
## metformin.rosiglitazone     FALSE TRUE
## metformin.pioglitazone      FALSE TRUE
```

```r
#Guardem el noms dels valors en un vector per després
nzv <- row.names(ZV[ZV$nzv==TRUE, ])
```

Notem que per ser diabètics una de les variables que no té variació és la de mesura de la glucosa. Procedim a llevar aquestes variables.

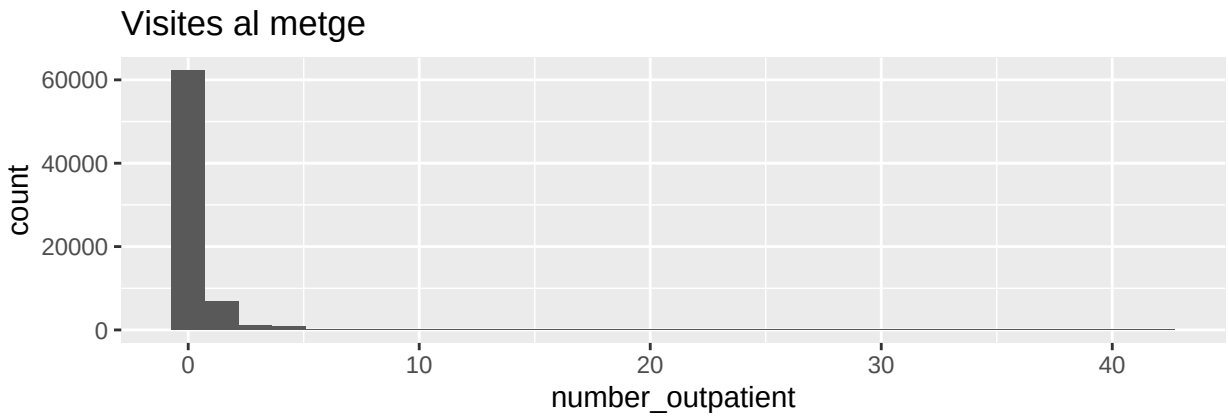
```r
dades <- select(dades, -all_of(nzv))
```

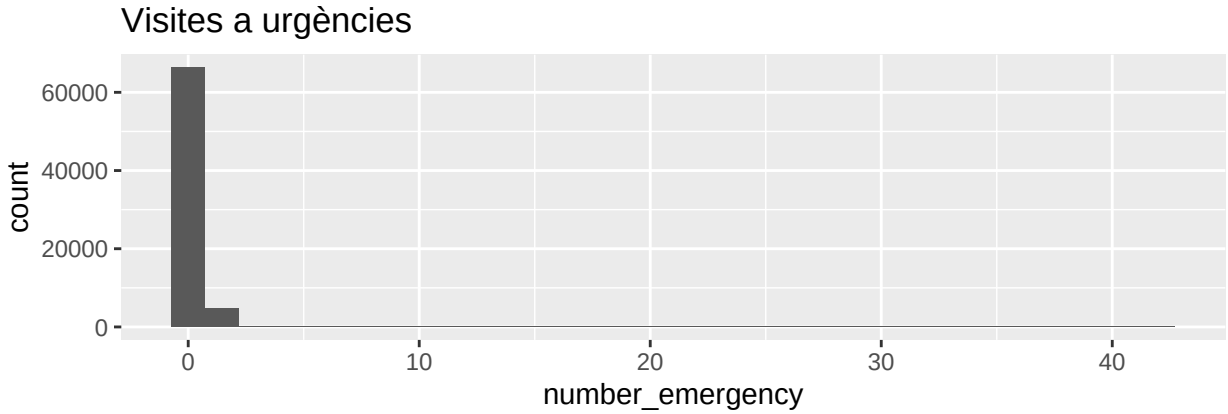
Aquesta funció només ens lleva variables que és molt clar que no varien, per tant encara poden quedar algunes variables que varien molt poc, on la majoria dels valors són els mateixos.

Anem a veure aquesta situació mitjançant una combinació de taules i gràfics

```r
p1 <- ggplot(dades, aes(x = number_outpatient))+geom_histogram()+ggtitle("Visites al metge")
p2 <- ggplot(dades, aes(x =number_emergency))+geom_histogram()+ggtitle("Visites a urgències")
grid.arrange(p1,p2)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```





Aquests dos camps, juntament amb el camp *number_inpatient* que hem representat abans no mostren gairebé variació i els eliminarem.

Revisem també la resta de medicacions

```
cols <- seq(19,24)

df <- data.frame()

for (i in cols) {
    registre <- c((names(dades)[i]),table(dades[,i]))
    df <-rbind(df,registre)

}

colnames(df) <- c("Medicament", "Down", "No", "Steady", "Up")

df
```

```
##      Medicament Down    No Steady   Up
## 1     metformin  435 56517  13714  838
## 2     glipizide  378 62400   8148  578
## 3     glyburide  421 63651   6811  621
## 4  pioglitazone   81 66198   5045  180
## 5 rosiglitazone   75 66807   4486  136
## 6       insulin 7504 34913  22125 6962
```

Notem com excepte en el cas de la insulina on hi ha certa varietat, la major part de medicacions no semblen administrar-se. I per tant també les llevarem.

Finalment notem que tenim dues columnes *admission_type_id* i *admission_source_id* que s'assemblen molt. La primera té 8 codis que fan referència si el pacient ingressa per urgències o emergències (dos codis diferents), ho fa per elecció (es suposa que ingrés programat), és nounat, per un trauma o per motius desconeguts. Mentre que la segona inclou els mateixos o semblants però també si ve d'un altre hospital (i el tipus d'hospital) i altres. Per tant decidim eliminar el primer

```
cols_a_llevar <- c("number_outpatient", "number_inpatient", "number_emergency", "metformin", "glipizide
                   "rosiglitazone", "admission_type_id")

dades <- select(dades, -all_of(cols_a_llevar))
```

4.1.2 Reducció de la quantitat i dels factors

A alguns atributs com a diagnòstics o tipus d'alta tenim un nombre molt elevat de categories. Intentem reduir-les per tal que la informació quedi més agrupada.

Ens guiem per l'article original per reduir els codis dels diferents diagnòstics a categories més simples. Concretament fent servir aquesta taula.

```
dades1 <- as.data.frame(dades) #Fem una còpia per si durant la realització de l'activitat necessitem to



###Diagnòstic 1

##Passem a factor la columna i desem els factors en una variable
dades$diag_1 <- as.factor(dades$diag_1)
cols <- levels(dades$diag_1)

# Desem els diagnostics propis de la diabetes
diag_diabet <- c("250", "250.01", "250.02", "250.03", "250.1"  ,"250.11", "250.12", "250.13", "250.2",
 "250.3",  "250.31", "250.32", "250.33", "250.4",  "250.41" ,"250.42", "250.43", "250.5",  "250.51", "25
 "250.7" , "250.8",  "250.81" ,"250.82" ,"250.83", "250.9",  "250.91", "250.92", "250.93")

## Amb un bucle convertim tots els valors que corresponen a certs diagnòstic a la seva nova categoria.
for (i in cols) {
   if (i %in% as.character(seq(390,459)) | i == "785" ) {
    levels(dades$diag_1)[levels(dades$diag_1) == i] <- "Circulatory"
  } else if (i %in% as.character(seq(460,519))| i == "786"){
     levels(dades$diag_1)[levels(dades$diag_1) == i] <- "Respiratory"
  } else if (i %in% as.character(seq(520,579))| i == "787"){
    levels(dades$diag_1)[levels(dades$diag_1) == i] <- "Digestive"
  } else if (i %in% diag_diabet){
    levels(dades$diag_1)[levels(dades$diag_1) == i] <- "Diabetes"
  } else if (i %in% as.character(seq(800,999))){
    levels(dades$diag_1)[levels(dades$diag_1) == i] <- "Injury"
  } else if (i %in% as.character(seq(710,739))){
    levels(dades$diag_1)[levels(dades$diag_1) == i] <- "Musculoeskeletal"
  } else if (i %in% as.character(seq(580,629))| i == "788"){
    levels(dades$diag_1)[levels(dades$diag_1) == i] <- "Genitourinary"
  } else if (i %in% as.character(seq(140,239))){
    levels(dades$diag_1)[levels(dades$diag_1) == i] <- "Neoplasms"
  } else{
    levels(dades$diag_1)[levels(dades$diag_1) == i] <- "Other"
```

```
    }
}

#Procedim de la mateixa manera amb els nous diagnostics
###Diagnòstic 2

dades$diag_2 <- as.factor(dades$diag_2)
cols <- levels(dades$diag_2)

for (i in cols) {
   if (i %in% as.character(seq(390,459)) | i == "785" ) {
    levels(dades$diag_2)[levels(dades$diag_2) == i] <- "Circulatory"
  } else if (i %in% as.character(seq(460,519))| i == "786"){
     levels(dades$diag_2)[levels(dades$diag_2) == i] <- "Respiratory"
  } else if (i %in% as.character(seq(520,579))| i == "787"){
    levels(dades$diag_2)[levels(dades$diag_2) == i] <- "Digestive"
  } else if (i %in% diag_diabet){
    levels(dades$diag_2)[levels(dades$diag_2) == i] <- "Diabetes"
  } else if (i %in% as.character(seq(800,999))){
    levels(dades$diag_2)[levels(dades$diag_2) == i] <- "Injury"
  } else if (i %in% as.character(seq(710,739))){
    levels(dades$diag_2)[levels(dades$diag_2) == i] <- "Musculoeskeletal"
  } else if (i %in% as.character(seq(580,629))| i == "788"){
    levels(dades$diag_2)[levels(dades$diag_2) == i] <- "Genitourinary"
  } else if (i %in% as.character(seq(140,239))){
    levels(dades$diag_2)[levels(dades$diag_2) == i] <- "Neoplasms"
  } else{
    levels(dades$diag_2)[levels(dades$diag_2) == i] <- "Other"


    }
}

###Diagnòstic 3


dades$diag_3 <- as.factor(dades$diag_3)
cols <- levels(dades$diag_3)

for (i in cols) {
   if (i %in% as.character(seq(390,459)) | i == "785" ) {
    levels(dades$diag_3)[levels(dades$diag_3) == i] <- "Circulatory"
  } else if (i %in% as.character(seq(460,519))| i == "786"){
     levels(dades$diag_3)[levels(dades$diag_3) == i] <- "Respiratory"
  } else if (i %in% as.character(seq(520,579))| i == "787"){
    levels(dades$diag_3)[levels(dades$diag_3) == i] <- "Digestive"
  } else if (i %in% diag_diabet){
    levels(dades$diag_3)[levels(dades$diag_3) == i] <- "Diabetes"
  } else if (i %in% as.character(seq(800,999))){
    levels(dades$diag_3)[levels(dades$diag_3) == i] <- "Injury"
  } else if (i %in% as.character(seq(710,739))){
    levels(dades$diag_3)[levels(dades$diag_3) == i] <- "Musculoeskeletal"
```

```
  } else if (i %in% as.character(seq(580,629))| i == "788"){
    levels(dades$diag_3)[levels(dades$diag_3) == i] <- "Genitourinary"
  } else if (i %in% as.character(seq(140,239))){
    levels(dades$diag_3)[levels(dades$diag_3) == i] <- "Neoplasms"
  } else{
    levels(dades$diag_3)[levels(dades$diag_3) == i] <- "Other"


    }
}

# Cada vegada que fem això hem d'indicar que els nivells antics i que no es fan servir s'han d'eliminar
dades <- droplevels(dades)

#Representem el diagnostic principal per a que es vegi com resulta
table(dades$diag_1)
```

```
##
##            Other        Neoplasms         Diabetes      Circulatory
##            12347             2742             5805            21893
##      Respiratory        Digestive    Genitourinary Musculoeskeletal
##             9776             6570             3514             4080
##           Injury
##             4777
```

Procedim ara de la mateixa manera però amb els camps d'admissió i d'alta.

Per reduir els valors d'aquests camps notem el següent:

1. Es nota que les dades han estat recollides en diferents moments i diferents hospitals i segurament havia criteris diferents. Per exemple a font d'ingrés tenim la identificació "part normal" per nounats, però també "naixement dins l'hospital" i "naixement fora de l'hospital", situacions que no són excloents amb el part normal
2. Els criteris en algun cas reflecteixen que estem tractant amb dades d'un sistema de salut diferent i resulta difícil entendre si les diferències entre els casos són importants (han de quedar) o si són més subtils i per tant no són necessàries.

En qualsevol cas en la nostra opinió aquests dos camps tenen massa descriptors i han de ser molt més ser reduïts (de fet no tots estan utilitzats a les dades). Per exemple gairebé tots els valors d'alta que no són "alta a casa" són transferències del pacient a hospitals, residències, alta sota atencions d'infermeres a casa i similars. Tots aquests valors han de quedar reduïts a una sola categoria.

En algun cas hem de prendre decisions subjectives, per exemple, un alta voluntària ("against medical advice" o AMA) com la classifiquem? Són molt pocs casos i hem optat per ajuntar-la amb aquelles altes que no són "completes" es a dir que suposen tornar a l'hospital amb cites programades, amb cures externes...

D'altra banda hem d'eliminar els pacients que moren, doncs no els hem de contemplar de cara a la readmissió. També fiquem en aquest grup els pacients que són transferits a una unitat/hospital per pacients terminals ("hospice" a l'original)

De tal manera que ens quedem amb dos grups d'altes: ALTA i ALTRES (on aquest indica que els pacients són donats d'alta amb algun tipus d'atenció o seguiment) i no disponibles

Respecte a la font d'admissió ens quedem amb: Urgències i Programat o Trasllat (que són gairebé tots els altres codis) i també conservem els no disponibles.

```r
#Convertim en factors
dades$admission_source_id <- as.factor(dades$admission_source_id)
dades$discharge_disposition_id <- as.factor(dades$discharge_disposition_id)


#Procedim de forma semblant a abans

cols = levels(dades$discharge_disposition_id)

for (i in cols) {
   if (i %in% c("1") ) {
    levels(dades$discharge_disposition_id)[levels(dades$discharge_disposition_id) == i] <- "Alta"
  } else if (i %in% c("11", "13", "14","19", "20", "21")){
     levels(dades$discharge_disposition_id)[levels(dades$discharge_disposition_id) == i] <- "Exitus"
  } else if (i %in% c("18", "25")){
    levels(dades$discharge_disposition_id)[levels(dades$discharge_disposition_id) == i] <- "No disponibl
  } else{
    levels(dades$discharge_disposition_id)[levels(dades$discharge_disposition_id) == i] <- "Altres"


    }
}

#Eliminem les observacions corresponents als morts
dades <- dades[dades$discharge_disposition_id != "Exitus", ]

cols = levels(dades$admission_source_id)

for (i in cols){
  if (i %in% c("7")){
    levels(dades$admission_source_id)[levels(dades$admission_source_id) == i] <- "Urgencia"
  }else if(i %in% c("9", "15", "17", "20", "21")){
    levels(dades$admission_source_id)[levels(dades$admission_source_id) == i] <- "No disponible"
  }else{
    levels(dades$admission_source_id)[levels(dades$admission_source_id) == i] <- "Programat_Trasllat"
  }
}


#Recordem que eliminar els valors no els elimina dels nivells, hem de eliminarl-los explicitament

dades <- droplevels(dades)

#Comprovem els nivells que tenim ara i la suma dels dos

unique(dades$discharge_disposition_id)
```

```
## [1] No disponible Alta          Altres
## Levels: Alta Altres No disponible
```

```r
unique(dades$admission_source_id)
```

```
## [1] Programat_Trasllat Urgencia           No disponible
## Levels: Programat_Trasllat Urgencia No disponible
```

```
b <- sum(dades$admission_source_id == "No disponible")
c <- sum(dades$admission_type_id == "No disponible")

print(sprintf("El valors catalogats com a no disponibles entre els dos grups són: %i", b+c))
```

```
## [1] "El valors catalogats com a no disponibles entre els dos grups són: 5068"
```

4.1.3 Revisió de les dades

Fins ara només ens hem centrat en revisar valors buits, llevar dades (atributs i registres) que a priori no semblen rellevants i/o pertinents i a reduir el nombre de categories, però encara no hem fet un cop d'ull a les dades, per dir-ho d'alguna manera.

Comencem revisant les dades que tenim hores d'ara

```
str(dades)
```

```
## 'data.frame':    69960 obs. of  18 variables:
##  $ race                    : chr  "Caucasian" "Caucasian" "AfricanAmerican" "Caucasian" ...
##  $ gender                  : chr  "Female" "Female" "Female" "Male" ...
##  $ age                     : chr  "[0-10)" "[10-20)" "[20-30)" "[30-40)" ...
##  $ discharge_disposition_id: Factor w/ 3 levels "Alta","Altres",..: 3 1 1 1 1 1 1 1 1 2 ...
##  $ admission_source_id     : Factor w/ 3 levels "Programat_Trasllat",..: 1 2 2 2 2 1 1 2 1 1 ...
##  $ time_in_hospital        : int  1 3 2 2 1 3 4 5 13 12 ...
##  $ num_lab_procedures      : int  41 59 11 44 51 31 70 73 68 33 ...
##  $ num_procedures          : int  0 0 5 1 0 6 1 0 2 3 ...
##  $ num_medications         : int  1 18 13 16 8 16 21 12 28 18 ...
##  $ diag_1                  : Factor w/ 9 levels "Other","Neoplasms",..: 3 1 1 1 2 4 4 4 4 4 ...
##  $ diag_2                  : Factor w/ 9 levels "Other","Neoplasms",..: NA 3 3 3 2 4 4 5 4 2 ...
##  $ diag_3                  : Factor w/ 9 levels "Other","Neoplasms",..: NA 1 1 4 3 3 1 3 1 5 ...
##  $ number_diagnoses        : int  1 9 6 7 5 9 7 8 8 8 ...
##  $ A1Cresult               : chr  "None" "None" "None" "None" ...
##  $ insulin                 : chr  "No" "Up" "No" "Up" ...
##  $ change                  : chr  "No" "Ch" "No" "Ch" ...
##  $ diabetesMed             : chr  "No" "Yes" "Yes" "Yes" ...
##  $ readmitted              : chr  "NO" ">30" "NO" "NO" ...
```

Passem a factor les variables que encara ens queden com a caràcter

```
cols <- c("race", "gender", "age", "A1Cresult", "insulin", "change", "diabetesMed", "readmitted")
for(i in cols){
  dades[,i] <- as.factor(dades[,i])
}
str(dades[,cols])
```

```
## 'data.frame':    69960 obs. of  8 variables:
##  $ race       : Factor w/ 6 levels "AfricanAmerican",..: 3 3 1 3 3 3 3 3 3 3 ...
##  $ gender     : Factor w/ 2 levels "Female","Male": 1 1 1 2 2 2 2 2 1 1 ...
##  $ age        : Factor w/ 10 levels "[0-10)","[10-20)",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ A1Cresult  : Factor w/ 4 levels ">7",">8","None",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ insulin    : Factor w/ 4 levels "Down","No","Steady",..: 2 4 2 4 3 3 3 2 3 3 ...
##  $ change     : Factor w/ 2 levels "Ch","No": 2 1 2 1 1 2 1 2 1 1 ...
##  $ diabetesMed: Factor w/ 2 levels "No","Yes": 1 2 2 2 2 2 2 2 2 2 ...
##  $ readmitted : Factor w/ 3 levels "<30",">30","NO": 3 2 3 3 3 2 3 2 3 3 ...
```

VARIABLE OBJECTIU

A continuació examinem el camp **readmitted** per veure com s'agrupen els valors

```
table(dades$readmitted)
```

```
##
##   <30   >30    NO
##  6275 22220 41465
```

Observem que tenim molt pocs valors per a la readmissió en menys de 30 dies, cosa que segurament dificultarà una bona predicció per part dels models.

Aprofitarem per crear una variables binària: ens dona 1 si el pacient és admès en menys de 30 dies i 0 en cas contrari

Normalment només s'ha de considerar el reingrés en menys d'un mes, però donat que tenim tan pocs casos farem també una altra variable binària que ens doni 1 si el pacient és readmés en menys d'un any (grup "<30" i ">30") i zero si no ho és.

```
dades$readmittedMES <- as.factor(ifelse(dades$readmitted == "<30", 1, 0))
dades$readmittedANY <- as.factor(ifelse(dades$readmitted == "NO", 0, 1))



perc <- sum(dades$readmittedMES == 1)/nrow(dades)

print(sprintf("El percentatge de pacients que reingressen en menys d'un MES es: %.2f%%", perc*100))
```

```
## [1] "El percentatge de pacients que reingressen en menys d'un MES es: 8.97%"
```

```
perc <- sum(dades$readmittedANY == 1)/nrow(dades)

print(sprintf("El percentatge de pacients que reingressen en menys d'un ANY es: %.2f%%", perc*100))
```

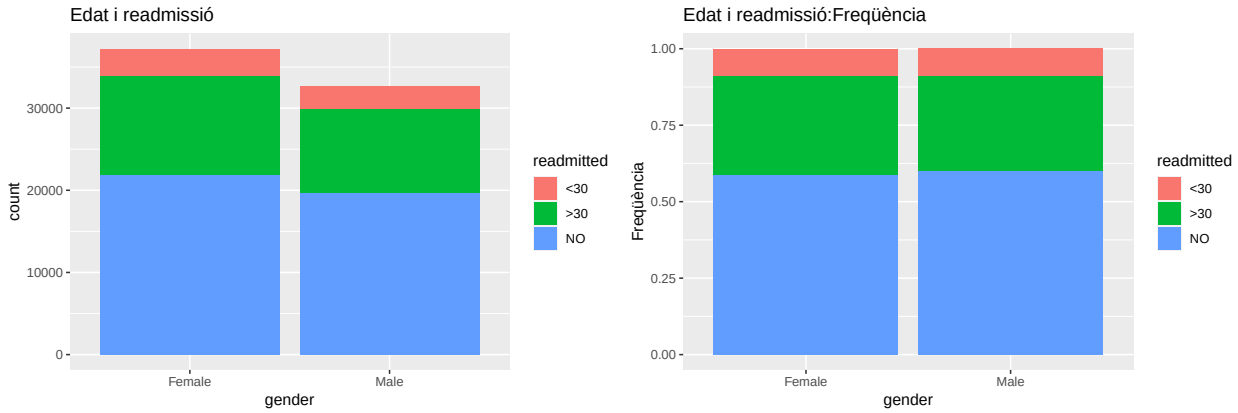
```
## [1] "El percentatge de pacients que reingressen en menys d'un ANY es: 40.73%"
```

Comprovem com gairebé no tenim pacients que reingressen en menys d'un mes, mentre que pel total dels que reingressen en menys d'un any les dades estan més equilibrades.

Anem a veure com es comporten algunes de la resta de variables en funció d'aquesta

GÈNERE

```
p1 <- ggplot(dades, aes(x = gender, fill = readmitted)) + geom_bar()+ ggtitle("Edat i readmissió")

p2 <- ggplot(dades, aes(x = gender, fill = readmitted)) + geom_bar(position = "fill")+ ggtitle("Edat i r

grid.arrange(p1,p2, nrow = 1)
```



Notem com sembla que la proporció de readmesos sembla la mateixa en homes que en dones.

Aquesta serà una de les **hipòtesis** que comprovarem en el punt següent.

RAÇA

Si recordem tenim alguns valors desconeguts en aquesta variable. Mirem la quantitat de valors que tenim en cada una de les races

```
table(dades$race)
```

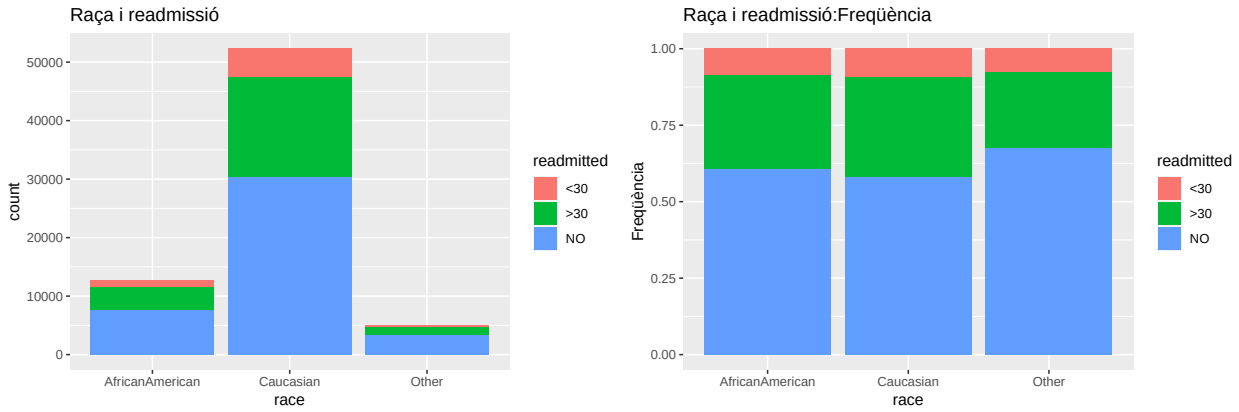
```
##
## AfricanAmerican           Asian       Caucasian      desconegut        Hispanic
##           12621             488           52289            1915            1498
##           Other
##            1149
```

Com es pot apreciar les ètnies que són diferents de blanca i negre tenen una presència més aviat testimonial. Ajuntarem totes aquestes i els valors desconeguts en la categoria "Other"

```
dades$race[dades$race == "Asian" | dades$race == "desconegut" | dades$race == "Hispanic"] <- "Other"

dades <- droplevels(dades)
```

Representem ara la raça comparant-la amb la variable "readmitted" igual que amb sexe

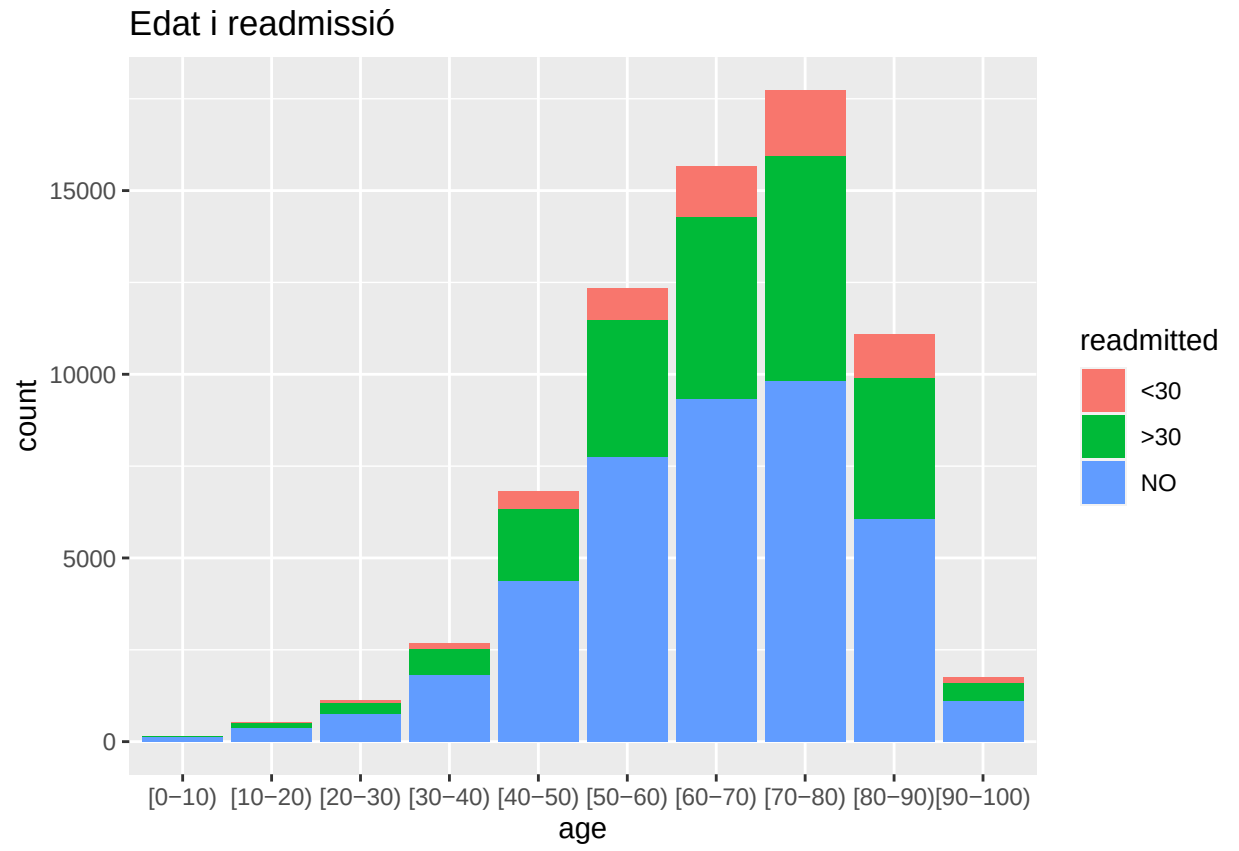
```
p1 <- ggplot(dades, aes(x = race, fill = readmitted)) + geom_bar()+ ggtitle("Raça i readmissió")

p2 <- ggplot(dades, aes(x = race, fill = readmitted)) + geom_bar(position = "fill")+ ggtitle("Raça i rea

grid.arrange(p1,p2, nrow = 1)
```



EDAT

Representem els valors de la variable edat

```
ggplot(dades, aes(age, fill =readmitted))+geom_bar()+ggtitle("Edat i readmissió")
```



Com era d'esperar hi ha més persones a mesura que l'edat puja fins al segment de 70 a 80 anys, moment en que comença a davallar la quantitat ja que hi ha menys població dins aquest grup d'edat.

A1Cresult

En aquest camp tenim la prova de l'hemoglobina glicosilada que és una prova que es fa als pacients per

conèixer si el control de la diabetis és el correcte.

Si mirem el percentatge de proves que s'han realitzat al conjunt dels pacients que tenim:

```
round(100*sum(dades$A1Cresult != "None")/nrow(dades),2)
```

```
## [1] 18.36
```

Veiem que la prova només s'ha fet a un 18.36 % dels pacients. Tot i que no és una prova necessària en un pacient agut, als hospitals espanyols sí que es du a terme quan s'ingressa a un pacient diabètic.

Resulta molt curiós que aquest joc de dades sigui sobre pacients diabètics i la mesura de la glucosa l'haguem descartada (tenia variació gairebé zero) i la de la prova de l'hemoglobina glicosilada només s'hagi realitzat a un percentatge inferior al 20 %.

Una altra de les **hipòtesis** que farem és veure si aquells pacients sobre els que es fa aquesta mesura reingressen menys, podent ser això un indicador de que el control de la malaltia és important.

CAMPS NUMÈRICS

No representem ni retoquem els camps numèrics perquè, tot i que no tots els utilitzarem en l'elaboració dels models, els podrem fer servir per a la comprovació de la normalitat de les dades al següent punt i també per a veure correlaciones entre les variables.

DESAR L'ARXIU DE DADES

```
write.csv(dades, file = "./data/dadesDiabetis.csv")
```

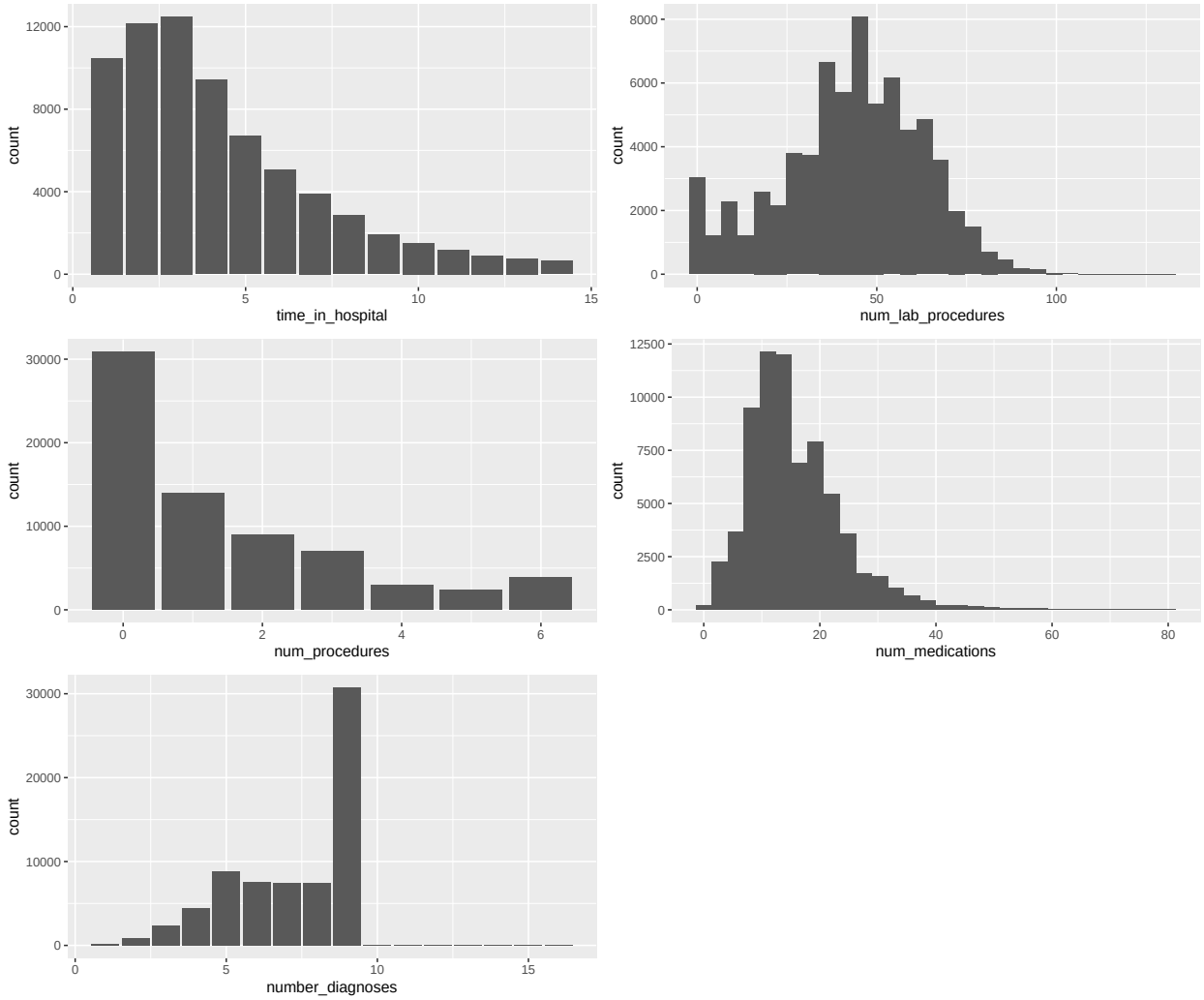
4.2 Comprovació de la normalitat i homogeneïtat de la variància

NORMALITAT

Representem les nostres variables numèriques abans de fer cap comprovació de normalitat. És necessari esmentar que les variables tot i numèriques, no són continues sinó senceres..

```
p1 <- ggplot(dades, aes(time_in_hospital)) + geom_bar()
p2 <- ggplot(dades, aes(num_lab_procedures)) + geom_histogram()
p3 <- ggplot(dades, aes(num_procedures)) + geom_bar()
p4 <- ggplot(dades, aes(num_medications)) + geom_histogram()
p5 <- ggplot(dades, aes(number_diagnoses)) + geom_bar()

grid.arrange(p1, p2, p3, p4, p5)
```

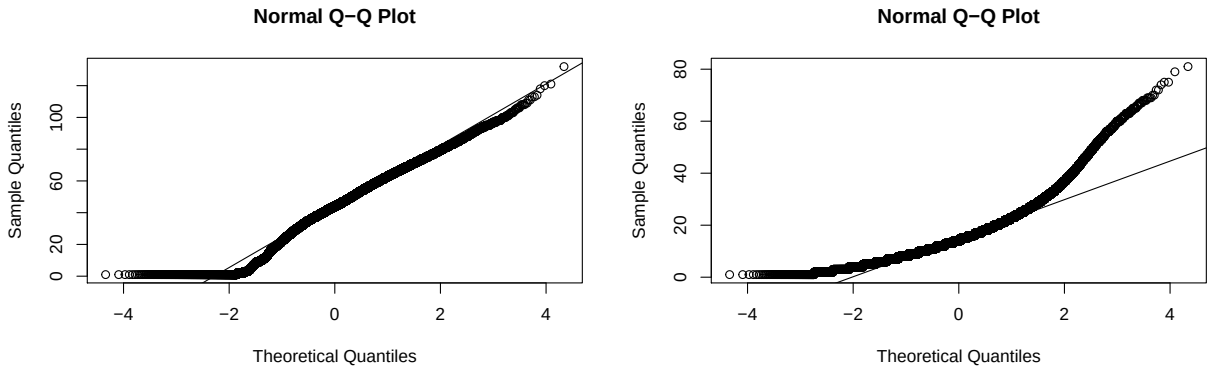
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Simplement observant les gràfiques podem veure que les variables de la columna de l'esquerra no segueixen una distribució normal.

Las variables de la columna de la dreta també tenen biaix cap a un dels costats i per tant segurament no seran normals, però fem servir una gràfica Q-Q per veure si es compleix la normalitat.

```
par(mfrow = c(1,2))
qqnorm(dades$num_lab_procedures)
qqline(dades$num_lab_procedures)

qqnorm(dades$num_medications)
qqline(dades$num_medications)
```



Com havíem anticipat les dues gràfiques es separen significativament de la línia en els extrems i per tant tampoc podríem considerar que aquestes variables segueixen una distribució normal.

HOMOGENEITAT DE LA VARIÀNCIA

En el nostre cas no hi ha lloc a comprovar aquesta homogeneïtat donat que no compararem grups de variables numèriques.

4.3 Proves estadístiques.

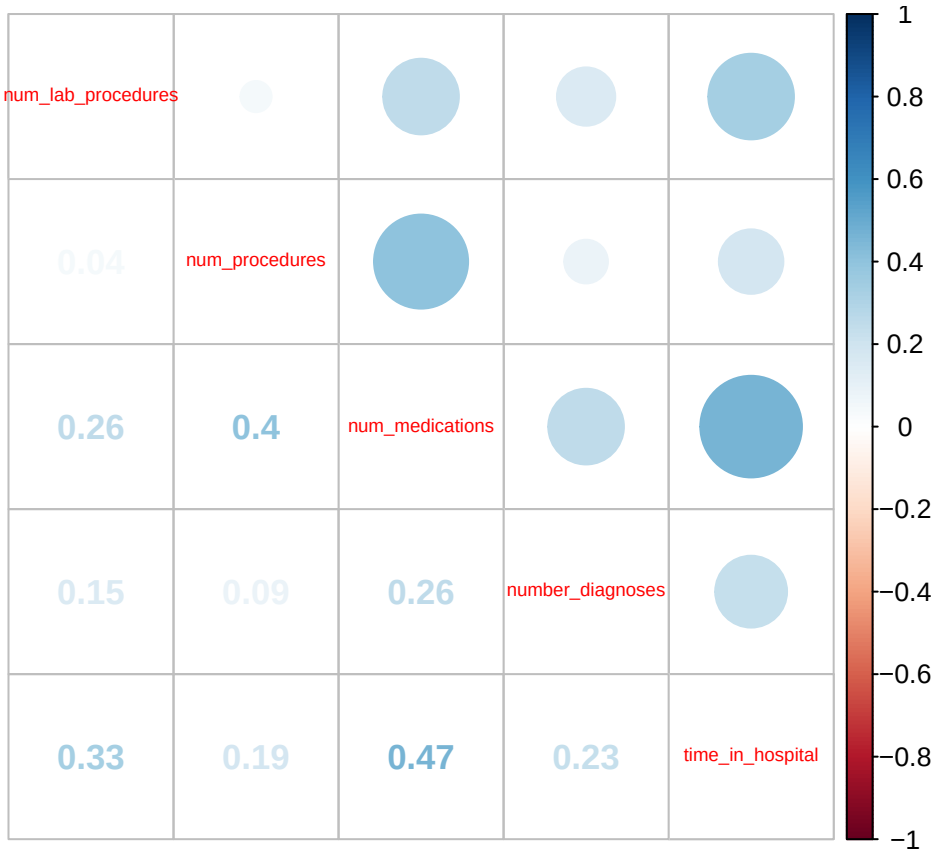
4.3.1 Correlació entre variables numèriques

Aprofitarem per veure si tenim correlació entre les variables numèriques, més dies a l'hospital impliquen més procediments? o més medicacions?

```
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
dades_numeriques <- select(dades, c("num_lab_procedures", "num_procedures", "num_medications", "number_
                                    "time_in_hospital") )
correlacions <- cor(dades_numeriques)

corrplot.mixed(correlacions, upper = "circle", number.cex = 1.1, tl.cex = .6)
```



Observem que la correlació més important és entre el temps a l'hospital i el nombre de medicacions, tot i així no arriba a 0.5. El que sí s'observa és que totes les correlacions són positives com sembla lògic.

4.3.2 És major la proporció d'homes o dones que reingressen?

Volem saber si s'observen diferències per sexe en la taxa de reingrés. Abans de fer res observem els percentatges de cada cas.

```
#Separem les dades en fos dataframes, perquè els necessitarem després
dHomes <- dades[dades$gender == "Male",]
dDones <- dades[dades$gender == "Female", ]

pHomes <- sum(dHomes$readmittedMES == 1)/nrow(dHomes)
pDones <- sum(dDones$readmittedMES == 1)/nrow(dDones)

print(sprintf(" El percentatge d'homes que reingressen en menys de 30 dies és: %.2f%%", pHomes*100))
```

```
## [1] " El percentatge d'homes que reingressen en menys de 30 dies és: 8.90%"
```

```
print(sprintf(" El percentatge de dones que reingressen en menys de 30 dies és: %.2f%%", pDones*100))
```

```
## [1] " El percentatge de dones que reingressen en menys de 30 dies és: 9.03%"
```

```
pHomes <- sum(dHomes$readmittedANY == 1)/nrow(dHomes)
pDones <- sum(dDones$readmittedANY == 1)/nrow(dDones)

print(sprintf(" El percentatge d'homes que reingressen en menys d'un any és: %.2f%%", pHomes*100))
```

```
## [1] " El percentatge d'homes que reingressen en menys d'un any és: 39.88%"
```

```
print(sprintf(" El percentatge de dones que reingressen en menys d'un anys és: %.2f%%", pDones*100))
```

```
## [1] " El percentatge de dones que reingressen en menys d'un anys és: 41.48%"
```

Tant en el cas de reingrés en menys d'un mes com en el de més d'un any la diferència de percentatges és molt petita, però superior per les dones. Hem de recordar que tenim un nombre molt elevat de dades, i això podria fer que tot i petita, la diferència fos estadísticament significativa.

Fem un test d'hipòtesi per comprovar si la proporció de dones que reingressen és superior la dels els homes amb un nivell de confiança del 95 %:

$$H_0: \ p_{dones} = p_{homes}$$
$$H_1: \ p_{dones} > p_{homes}$$

Farem el test tant pel cas de reingrés en un mes com en un any.

CAS 1 MES

```
nHomes <- nrow(dHomes)
nDones <- nrow(dDones)

pHomes <- sum(dHomes$readmittedMES == 1)/nHomes
pDones <- sum(dDones$readmittedMES == 1)/nDones

success <- c(pDones*nDones, pHomes*nHomes )
nn <- c(nDones, nHomes)

prop.test(success,nn, alternative = "greater", correct = FALSE)
```

```
##
##  2-sample test for equality of proportions without continuity
##  correction
##
## data:  success out of nn
## X-squared = 0.33901, df = 1, p-value = 0.2802
## alternative hypothesis: greater
## 95 percent confidence interval:
##  -0.002299196  1.000000000
## sample estimates:
##     prop 1     prop 2
## 0.09028393 0.08902331
```

En aquest cas veiem que p > 0.05 i que per tant no podem rebutjar la hipòtesi nul · la. Es a dir, no podem concloure que en menys d'un mes reingressen mes dones.

CAS 1 ANY

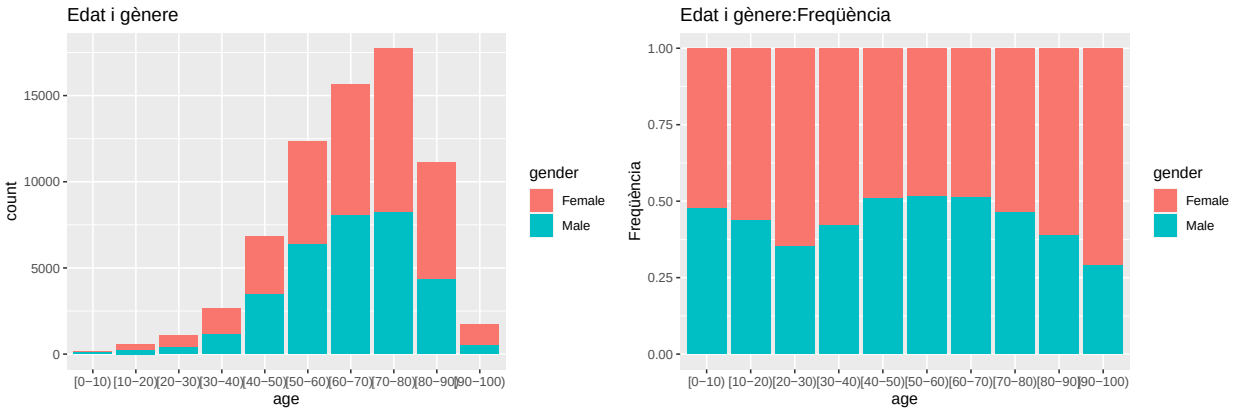
```
pHomes <- sum(dHomes$readmittedANY == 1)/nHomes
pDones <- sum(dDones$readmittedANY == 1)/nDones

success <- c(pDones*nDones, pHomes*nHomes )
nn <- c(nDones, nHomes)
```

```r
prop.test(success,nn, alternative = "greater", correct = FALSE)
```

```
##
##  2-sample test for equality of proportions without continuity
##  correction
##
## data:  success out of nn
## X-squared = 18.418, df = 1, p-value = 8.869e-06
## alternative hypothesis: greater
## 95 percent confidence interval:
##  0.009856814 1.000000000
## sample estimates:
##    prop 1    prop 2
## 0.4147796 0.3988024
```

Per contra, pel cas d'un any, veiem que podem refusar la hipòtesi nul · la i concloure que la proporció de dones que reingressen és major. Un dels motius podria ser que sol haver més dones als grups de més edat. Ho comprovem a les següents gràfiques.

```r
p1 <- ggplot(dades, aes(x = age, fill = gender)) + geom_bar()+ ggtitle("Edat i gènere")

p2 <- ggplot(dades, aes(x = age, fill = gender)) + geom_bar(position = "fill")+ ggtitle("Edat i gènere:

grid.arrange(p1,p2, nrow = 1)
```



De totes maneres, recordar que com es tracta d'un estudi observacional només podem comprovar associacions estadístiques però no relacions causa-efecte.

4.3.3 És millor mesurar l'hemoglobina glicosilada o no?

Com es tracta de pacients diabètics, una de les mesures més importants que podem fer amb aquest joc de dades és si el control de la malaltia afecta a la taxa de reingrés. La forma de revisar si la diabetis està ben controlada pel pacient és amb la prova de l'hemoglobina glicosilada o HbA1c.

Podem repetir un test d'hipòtesi semblant a l'anterior però ara mirant si la proporció de pacients als quals se'ls ha fet la prova i que reingressen és menor que aquells que no.

$$H_0: \quad p_{HbA1c} = p_{No\_HbA1c}$$

$$H_1 : \ p_{HbA1c} < p_{No\_HbA1c}$$

```
# Procedim de forma semblant. Primer separem les dades en dos dataframes

dA1C <- dades[dades$A1Cresult != "None",]
dNone <- dades[dades$A1Cresult == "None",]



pA1C <- sum(dA1C$readmittedMES == 1)/nrow(dA1C)
pNone <- sum(dNone$readmittedMES == 1)/nrow(dNone)

print(sprintf("La proporcio de pacient que reingressen al mes i que SÍ tenen la prova feta és: %.2f%%",
```

```
## [1] "La proporcio de pacient que reingressen al mes i que SÍ tenen la prova feta és: 8.39%"
```

```
print(sprintf("La proporcio de pacient que reingressen al mes i que NO tenen la prova feta és: %.2f%%",
```

```
## [1] "La proporcio de pacient que reingressen al mes i que NO tenen la prova feta és: 9.10%"
```

Veiem que al igual que abans la diferència és molt petita però podem fer el test i comprovar-ho

```
exit <- c(sum(dA1C$readmittedMES == 1), sum(dNone$readmittedMES == 1))
numeros <- c(nrow(dA1C), nrow(dNone))
prop.test(exit, numeros, alternative = "less", correct = FALSE)
```

```
##
##  2-sample test for equality of proportions without continuity
##  correction
##
## data:  exit out of numeros
## X-squared = 6.5448, df = 1, p-value = 0.00526
## alternative hypothesis: less
## 95 percent confidence interval:
##  -1.000000000 -0.002655252
## sample estimates:
##     prop 1     prop 2
## 0.08386544 0.09100459
```

Obtenim un p < 0.05 i podem rebutjar la hipòtesi nul · la i concloure que la proporció de pacients que reingressen en un mes és menor si durant la seva estança s'ha fet aquesta prova.

Com s'ha comentat no podem assignar una relació causa-efecte, però la hipòtesi a comprovar en aquest cas seria que quan es descarta la prova perquè no sembla necessària se'ns estan escapant casos de diabètics que podrien tenir un millor control.

En aquest cas és possible que això sigui així, perquè hem de recordar que als hospitals espanyols aquesta mesura sí que es fa de forma habitual en els pacients diabètics.

4.3.4 MODEL DE REGRESSIÓ LOGíSTICA

Provarem ara un model de regressió logística que ens doni una probabilitat de ser readmès, en menys d'un mes i en menys d'un any (tot i que el primer model és important, fem el segon de cara a la comparació)

MODEL READMISSIÓ EN UN MES

Abans de fer el model separem les mostres en una mostra d'entrenament i una altra de test fent servir la funció *createDataPartition* del paquet *caret*, que ja ens distribueix la variable a predir de forma equilibrada entre les dues mostres. Com tenim moltes dades ens reservem només el 65 % per l'entrenament

```
intrain <- createDataPartition(y = dades$readmittedMES, p = 0.65, list = F)
train <- dades[intrain,]
test <- dades[-intrain,]
```

Per realitzar el model tenim moltes variables que podem incloure i s'hauria de fer una exploració entre les diferents combinacions i anar provant diferents interaccions entre variables. A la bibliografia (James et al.) s'assenyala que hi ha 3 mètodes habituals: selecció cap endavant (*forward selection*, començar sense variables i anar afegint), selecció cap enrera (*backward selection*, el contrari que l'anterior cas) i seleccio barrejada (*mixed selection*, combinació del anteriors).

Nosaltres hem considerat que l'objectiu de la pràctica no era tant trobar el millor model com provar i triar els adequats al joc de dades, i que, per tant, l'exploració exhaustiva de les diferents combinacions quedava fora de l'abast de la pràctica. Hem fet algunes proves i ens quedem amb una combinació prou bona (d'entre les proves realitzades)

(Nota: Gairebé no hem explorat termes d'interacció, i no s'inclou cap)

```
model.Log.MES <- glm(readmittedMES ~ gender+race + age + +A1Cresult +diag_1 + discharge_disposition_id+




summary(model.Log.MES)
```

```
##
## Call:
## glm(formula = readmittedMES ~ gender + race + age + +A1Cresult +
##     diag_1 + discharge_disposition_id + num_lab_procedures +
##     num_medications, family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.6787  -0.4674  -0.4000  -0.3570   2.6911
##
## Coefficients:
##                                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)                       -3.8003587  0.5971865  -6.364 1.97e-10
## genderMale                         0.0660652  0.0336007   1.966 0.049277
## raceCaucasian                      0.0058475  0.0448144   0.130 0.896184
## raceOther                         -0.1384216  0.0775313  -1.785 0.074202
## age[10-20)                         0.5222574  0.6333150   0.825 0.409576
## age[20-30)                         1.0816851  0.6016484   1.798 0.072198
## age[30-40)                         0.9694775  0.5940397   1.632 0.102678
## age[40-50)                         0.9210622  0.5901525   1.561 0.118590
## age[50-60)                         0.8513803  0.5893738   1.445 0.148584
## age[60-70)                         1.0682778  0.5890041   1.814 0.069724
## age[70-80)                         1.1323591  0.5889179   1.923 0.054508
## age[80-90)                         1.1055674  0.5894953   1.875 0.060731
## age[90-100)                        0.9161274  0.5967818   1.535 0.124756
## A1Cresult>8                       -0.1367653  0.0988979  -1.383 0.166697
## A1CresultNone                     -0.0008728  0.0817191  -0.011 0.991478
## A1CresultNorm                     -0.0186329  0.1055801  -0.176 0.859916
## diag_1Neoplasms                   -0.0590279  0.0972042  -0.607 0.543680
```

```
## diag_1Diabetes                          0.0857367  0.0700921   1.223 0.221254
## diag_1Circulatory                       0.0084623  0.0498544   0.170 0.865214
## diag_1Respiratory                      -0.2971070  0.0637703  -4.659 3.18e-06
## diag_1Digestive                        -0.1060418  0.0694200  -1.528 0.126627
## diag_1Genitourinary                    -0.1400734  0.0856195  -1.636 0.101840
## diag_1Musculoeskeletal                 -0.3408427  0.0847751  -4.021 5.81e-05
## diag_1Injury                            0.0100249  0.0713481   0.141 0.888260
## discharge_disposition_idAltres          0.5832313  0.0372626  15.652  < 2e-16
## discharge_disposition_idNo disponible   0.2728654  0.0783250   3.484 0.000494
## num_lab_procedures                      0.0043801  0.0009190   4.766 1.88e-06
## num_medications                         0.0046977  0.0020805   2.258 0.023946
##
## (Intercept)                            ***
## genderMale                             *
## raceCaucasian
## raceOther                              .
## age[10-20)
## age[20-30)                             .
## age[30-40)
## age[40-50)
## age[50-60)
## age[60-70)                             .
## age[70-80)                             .
## age[80-90)                             .
## age[90-100)
## A1Cresult>8
## A1CresultNone
## A1CresultNorm
## diag_1Neoplasms
## diag_1Diabetes
## diag_1Circulatory
## diag_1Respiratory                      ***
## diag_1Digestive
## diag_1Genitourinary
## diag_1Musculoeskeletal                 ***
## diag_1Injury
## discharge_disposition_idAltres         ***
## discharge_disposition_idNo disponible  ***
## num_lab_procedures                     ***
## num_medications                        *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 27452  on 45474  degrees of freedom
## Residual deviance: 26933  on 45447  degrees of freedom
## AIC: 26989
##
## Number of Fisher Scoring iterations: 5
```

Notem que el valor p es superior a 0.05 en gairebé tots els factors, cosa que segurament indica que el model no resulta gaire bo.

Representem en una taula els resultats del model amb les dades del conjunt d'entrenament. Triem un valor del 50 % per decidir si classifiquem un resultat com postitiu o negatiu

```
tb <- table(predict(model.Log.MES, data = train, type = "response") > 0.5, train$readmittedMES)

tb
```

```
##
##              0     1
##   FALSE 41396  4079
```

El model ens classifica TOTS els valors com no reingrés, cosa que farà que la seva precisió sigui del voltant del 90% ja que només un 10 % reingressa.

```
prec <- sum(diag(tb))/nrow(train)

print(sprintf("La precisió del model és: %.2f%%", prec*100))
```

```
## [1] "La precisió del model és: 91.03%"
```

Si fem el mateix però amb les dades de test

```
prediccio <- predict(model.Log.MES, test, type = "response")

tb <- table(prediccio > 0.5, test$readmittedMES)

tb
```

```
##
##              0     1
##   FALSE 22289  2196
```

```
prec <- sum(diag(tb))/nrow(test)

print(sprintf("La precisió del model és: %.2f%%", prec*100))
```

```
## [1] "La precisió del model és: 91.03%"
```

Obtenim una precisió molt semblant a l'anterior.

NOTEM com ja hem assenyalat que aquest model no ens aporta res. Si classifiquem tots els casos com de NO reingrés tindrem el mateix encert que el model.

MODEL READMISSIÓ EN UN ANY

```
#Tornem a crear els conjunts d'entenament i de test per a que el criteri d'estratificacio sigui la nova
intrain <- createDataPartition(y = dades$readmittedANY, p = 0.65, list = F)
train <- dades[intrain,]
test <- dades[-intrain,]

model.Log.ANY <- glm(readmittedANY ~ gender + race + age + A1Cresult + diag_1 + discharge_disposition_i



summary(model.Log.ANY)
```

```
##
## Call:
## glm(formula = readmittedANY ~ gender + race + age + A1Cresult +
##     diag_1 + discharge_disposition_id + num_lab_procedures +
##     num_medications, family = binomial, data = train)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.394  -1.032  -0.895   1.279   1.975
##
## Coefficients:
##                                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)                          -2.0216796  0.2637669  -7.665 1.79e-14
## genderMale                           -0.0646030  0.0196058  -3.295 0.000984
## raceCaucasian                         0.0450144  0.0257649   1.747 0.080616
## raceOther                            -0.3657235  0.0441493  -8.284  < 2e-16
## age[10-20)                            0.7217398  0.2794209   2.583 0.009795
## age[20-30)                            0.8898184  0.2668737   3.334 0.000855
## age[30-40)                            0.8899405  0.2605435   3.416 0.000636
## age[40-50)                            0.9927874  0.2577581   3.852 0.000117
## age[50-60)                            1.0521779  0.2571728   4.091 4.29e-05
## age[60-70)                            1.1634556  0.2570885   4.526 6.03e-06
## age[70-80)                            1.3330787  0.2570728   5.186 2.15e-07
## age[80-90)                            1.3112916  0.2575668   5.091 3.56e-07
## age[90-100)                           0.8692926  0.2638533   3.295 0.000986
## A1Cresult>8                           0.0925211  0.0588430   1.572 0.115872
## A1CresultNone                         0.2209274  0.0499264   4.425 9.64e-06
## A1CresultNorm                        -0.0936568  0.0645142  -1.452 0.146578
## diag_1Neoplasms                      -0.3721758  0.0575551  -6.466 1.00e-10
## diag_1Diabetes                        0.1870205  0.0420251   4.450 8.58e-06
## diag_1Circulatory                     0.0290283  0.0298035   0.974 0.330061
## diag_1Respiratory                     0.0712048  0.0351813   2.024 0.042977
## diag_1Digestive                      -0.0511784  0.0396986  -1.289 0.197337
## diag_1Genitourinary                  -0.1194251  0.0498948  -2.394 0.016687
## diag_1Musculoeskeletal               -0.2570627  0.0483743  -5.314 1.07e-07
## diag_1Injury                         -0.1471243  0.0443251  -3.319 0.000903
## discharge_disposition_idAltres        0.1683582  0.0227191   7.410 1.26e-13
## discharge_disposition_idNo disponible -0.1883963  0.0477574  -3.945 7.98e-05
## num_lab_procedures                    0.0049929  0.0005358   9.318  < 2e-16
## num_medications                       0.0043588  0.0012769   3.414 0.000641
##
## (Intercept)                          ***
## genderMale                           ***
## raceCaucasian                        .
## raceOther                            ***
## age[10-20)                           **
## age[20-30)                           ***
## age[30-40)                           ***
## age[40-50)                           ***
## age[50-60)                           ***
## age[60-70)                           ***
## age[70-80)                           ***
## age[80-90)                           ***
## age[90-100)                          ***
```

```
## A1Cresult>8
## A1CresultNone                          ***
## A1CresultNorm
## diag_1Neoplasms                        ***
## diag_1Diabetes                         ***
## diag_1Circulatory
## diag_1Respiratory                      *
## diag_1Digestive
## diag_1Genitourinary                    *
## diag_1Musculoeskeletal                 ***
## diag_1Injury                           ***
## discharge_disposition_idAltres         ***
## discharge_disposition_idNo disponible ***
## num_lab_procedures                     ***
## num_medications                        ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 61470  on 45474  degrees of freedom
## Residual deviance: 60576  on 45447  degrees of freedom
## AIC: 60632
##
## Number of Fisher Scoring iterations: 4
```

Veiem que en aquesta ocasió tenim molts més coeficients amb $p < 0.05$, de fet gairebé tots. Notar com la variable sexe ara sí té un $p < 0.05$, recordar que en el test d'hipòtesi havíem vist com el percentatge de reingrés entre homes i dones era diferent pel cas d'un any.

Ara fem la taula amb els valors igual que abans, fixem un límit del 50 % en el criteri de separació.

Pel conjunt de dades de train la nostra precisió és:

```
tb <- table(predict(model.Log.ANY, data = train, type = "response") > 0.5, train$readmittedANY)

tb
```

```
##
##             0     1
##   FALSE 24927 16526
##   TRUE   2026  1996
```

```
prec <- sum(diag(tb))/nrow(train)

print(sprintf("La precisió del model és: %.2f%%", prec*100))
```

```
## [1] "La precisió del model és: 59.20%"
```

Mentre que pel nostre conjunt de test serà

```
prediccio <- predict(model.Log.ANY, test, type = "response")

tb <- table(prediccio > 0.5, test$readmittedANY)
```

```
tb
```

```
##
##              0     1
##   FALSE 13488  8886
##   TRUE   1024  1087
```

```
prec <- sum(diag(tb))/nrow(test)

print(sprintf("La precisió del model és: %.2f%%", prec*100))
```

```
## [1] "La precisió del model és: 59.53%"
```

Comprovem que la precisió del model és tampoc és bona i difícilment podem predir la taxa de reingrés amb aquest model.

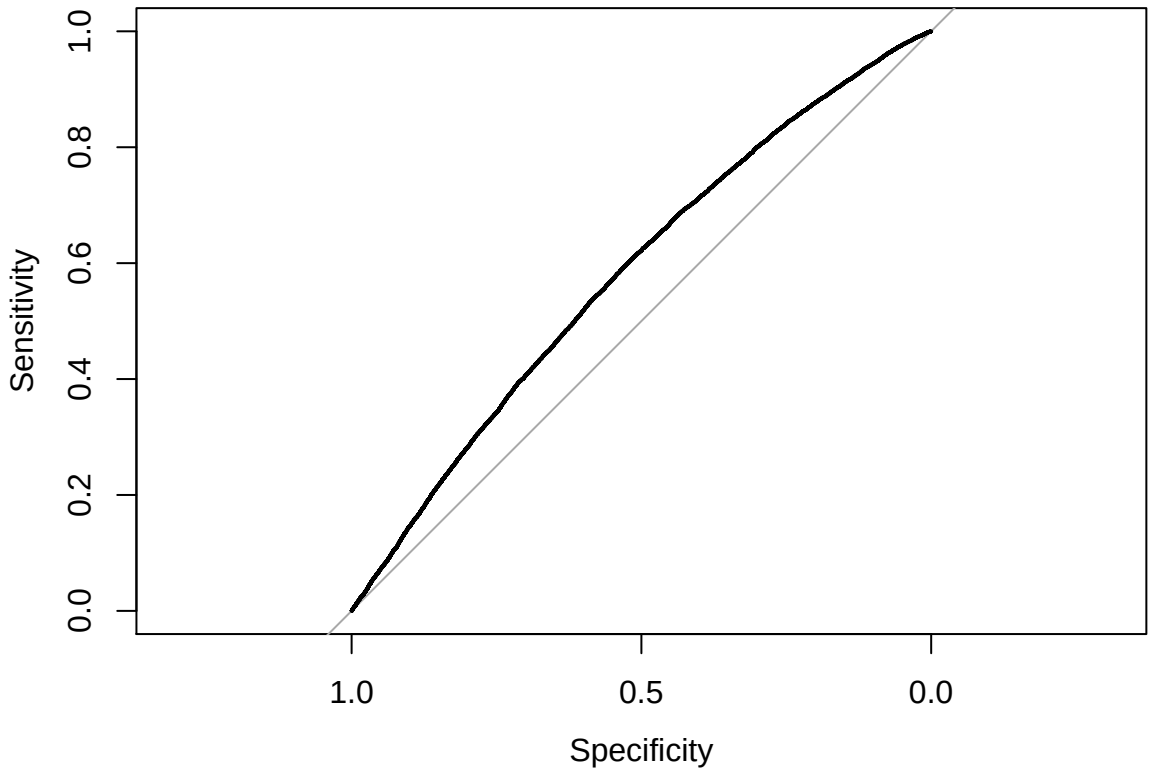
Tot i que ja sabem que el model no és gaire bo fem la representació de la corba roc fent servir *roc()* del paquet *pROC* i calculem l'àrea sota la corba

```
r <- pROC::roc(train$readmittedANY, predict(model.Log.ANY, data = train, type = "response"), data = tra
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
plot(r)
```



```
pROC::auc(r)
```

```
## Area under the curve: 0.5815
```

La corba està molt a prop de la diagonal indicant el que ja hem comentat de que el model no és gaire bo. I el valor de AUC és proper a 0.5

4.3.5 ARBRES DE DECISIÓ - RANDOM FOREST

El model random Forest es basa en crear tot un conjunt d'arbres de decisió (d'aquí el nom de forest: bosc) i decidir la classificació segons una "votació" entre arbres.

Seleccionem les mateixes dades que abans per tal de poder comparar una mica els models i dividim les dades en entrenament i test

```
set.seed(3) #fixem el valor per obtenir sempre els mateixos resultats

dades_RF <- select(dades, c("gender", "race", "age", "A1Cresult", "diag_1", "discharge_disposition_id",
                            "num_lab_procedures", "num_medications", "readmittedMES"))


intrain <- createDataPartition(y = dades_RF$readmittedMES, p = 0.65, list = F)
train <- dades_RF[intrain,]
test <- dades_RF[-intrain,]

model.RF.1 <- randomForest(train$readmittedMES ~., data = train, importante = TRUE, ntree = 30)


model.RF.1
```

```
##
## Call:
##  randomForest(formula = train$readmittedMES ~ ., data = train,      importante = TRUE, ntree = 30)
##                Type of random forest: classification
##                      Number of trees: 30
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 9.01%
## Confusion matrix:
##       0  1  class.error
## 0 41372 24 0.0005797662
## 1  4074  5 0.9987742094
```

L'error calculat és del 9.01%. *Molt breument, "OOB" o "One out of Bag" és un sistema en el qual les files de les dades que no s'han utilitzat en construir un arbre es fan servir com a test de l'arbre en qüestió i en d'altres en les que no s'ha fet servir.*

Fem la taula completa de predicció sobre les dades d'entrenament.

```
#Com ara tractem de predir una classe hem de posar "class" a type
predTrain <- predict(model.RF.1, train, type = "class")
table(predTrain, train$readmittedMES)
```

```
##
## predTrain     0     1
```

```
##          0 41396  3928
##          1     0   151
```

I ara sobre el joc de dades de test

```
predTest <- predict(model.RF.1, test, type = "class")
pred <- sum(predTest == test$readmittedMES)/nrow(test)
table(predTest, test$readmittedMES)
```

```
##
## predTest     0     1
##        0 22288  2195
##        1     1     1
```

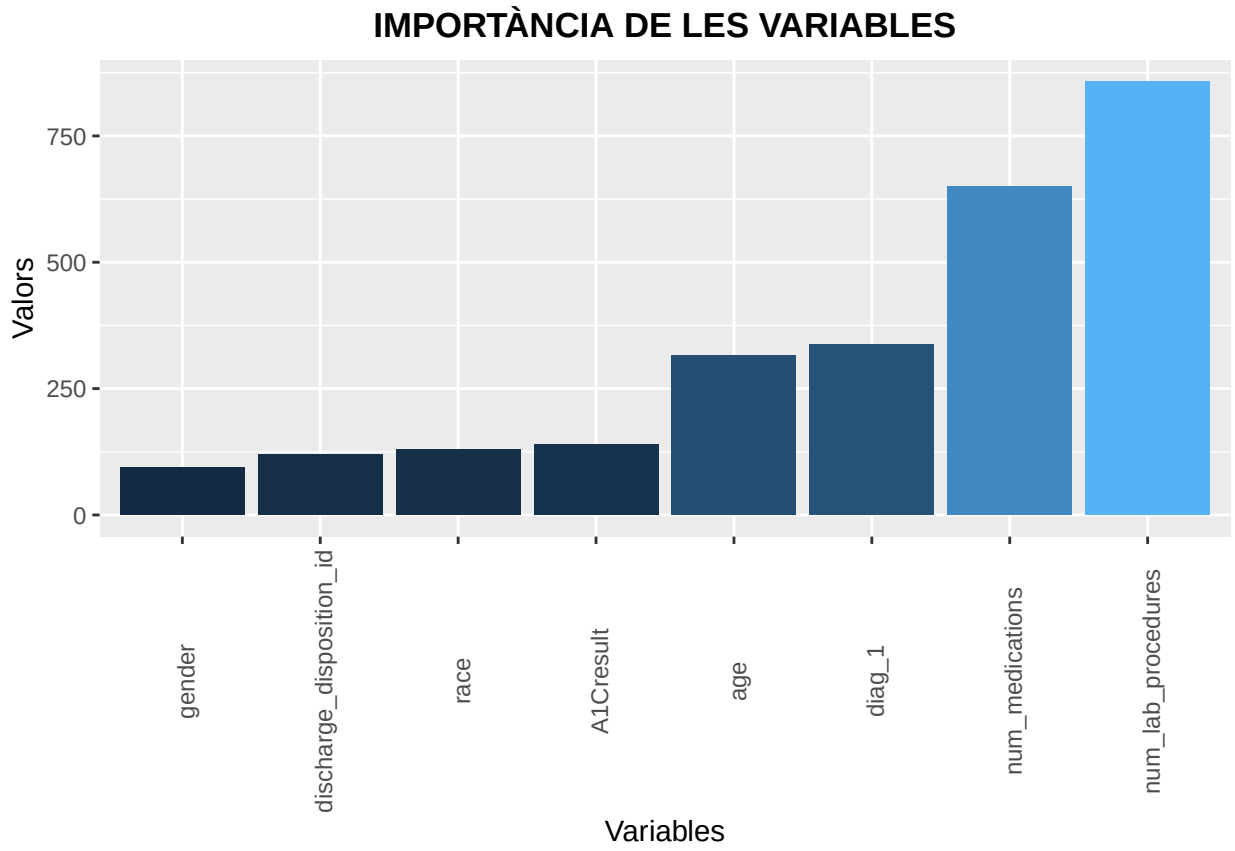
```
print(sprintf("La precisió del model és: %.2f%%", pred*100))
```

```
## [1] "La precisió del model és: 91.03%"
```

La precisió que obtenim és semblant a la de la regressió logística i és semblant a la que obtindríem si decidíssim classificar tots els casos com a no-readmesos, per tant, no guanyem gaire.

Com no es tracta d'un arbre sinó d'un sistema que fa servir diversos arbres, resulta difícil representar-ho gràficament, per això fem una representació de la importància de les diferents variables, segons el criteri de valor mig de decreixement en l'índex de GINI

```
# Necessitem reordenar les dades com un dataframe per poder fer la gràfica
df <- as.data.frame(model.RF.1$importance)
df$nomsVar <- rownames(df)
rownames(df) <- NULL

## Fem la gràfica
ggplot(df, aes(x = reorder(nomsVar, MeanDecreaseGini) , weight = MeanDecreaseGini))+
  geom_bar(aes(fill = MeanDecreaseGini), show.legend = FALSE)+
   theme(axis.text.x = element_text(angle = 90, vjust = 0.7),
        plot.title = element_text(hjust = 0.5, face = "bold"))+
  labs(title = "IMPORTÀNCIA DE LES VARIABLES", x = "Variables", y = "Valors")
```



Segons les dades tenim que les variables més importants són el nombre de proves de laboratori i de medicacions i amb bastant diferència.

5 Conclusions

En l'estudi d'aquest joc de dades ens hem fixat en diversos aspectes:

Per començar ens hem demanat si el sexe i la mesura de l'hemoglobina glicosilada es traduïen en una proporció diferent de reingrés, una diferència estadísticament significativa.

El que hem vist és que en el cas del **gènere** la proporció de dones que ingressava era major que la d'homes però només de forma estadísticament significativa ($p > 0.05$) pel cas de reingrés en un termini d'un any. En el cas de considerar només el mes següent a l'alta la diferència no era significativa.

Per la **mesura de HbA1c** hem comprovat que la proporció de persones que reingressaven un mes després és menor en el cas dels pacients als que se'ls ha fet la prova, indicant que possiblement és convenient fer aquesta prova que ajuda a detectar pacients diabètics descompensats.

Després hem intentant entrenar dos models: **regressió logística i random Forest**

En cap dels dos casos hem trobat bons resultats en els models, en el sentit de que si coneixem el percentatge de reingrés el nostre model no millora respecte aquest coneixement, encerta el mateix.

Hi ha diversos motius que poden explicar això:

1. Que el joc de dades en el format actual no permeti aquesta predicció, hi ha molts de factors i influeixen moltes coses i pot ser es necessiten aquests atributs i encara més, o és més convenient centrar-se en algun tipus de malaltia...

2. Revisar les nostres suposicions i desfer alguns canvis que hem fet en el procés. Nosaltres pensem que no hem fet cap canvi massa "agosarat" i que per tant aquest no deu ser el motiu.
3. Fer més proves i combinacions d'atributs per obtenir els models. Aquest podria ser el cas, donat que amb aquest joc de dades nosaltres tenim una precisió al voltant del 90 % i si fem un procés d'ajust potser pugem un parell de punts percentuals més cosa que tot i no ser espectacular sí que podria ser suficient en un tema tan complex com aquest.
4. Òbviament no podem descartar que en aquests moments aquestes dades siguin massa complicades pel nostre coneixement i que som nosaltres el que no som capaços d'obtenir un *insight* major.

6 Bibliografia

A més de la proposada a l'enunciat de la pràctica.

Beata Strack, Jonathan P. DeShazo, Chris Gennings, et al., "Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records," BioMed Research International, vol. 2014, Article ID 781670, 11 pages, 2014. https://doi.org/10.1155/2014/781670.

Bernadó, Ester, "Disseny experimental en analítica de dades" (PID_00247922).UOC.

Brownlee, Jason, "Tune Machine Learning Algorithms in R (random forest case study)" (Machine Learning Mastery) [en línia][Data d'actualització: 05-02-2016] [Data de consulta: 15-12-2020]. Disponible a: https://machinelearningmastery.com/tune-machine-learning-algorithms-in-r/

James, G., Witten, D., Hastie, T., Tibshirani, R. "An Introduction to Statistical Learning with Applications in R".Springer Texts in Statistics.Springer.2013