Tipologia i cicle de vida de les dades

Pràctica 2: Neteja i anàlisi de les dades

Autors: Adrià Tarradas i Aleix Arnau Soler

Desembre 2020

Contents

1. Presentació del projecte de ciència de dades

1.1. Background

L'enfonsament del Titanic és un dels naufragis més famosos de la història. El 15 d'abril de 1912, durant el seu viatge inaugural, el considerat "inenfonsable" RMS Titanic es va enfonsar després de xocar amb un iceberg. Malauradament, no hi havia prou bots salvavides per a tothom a bord, cosa que va provocar la mort de 1502 de 2224 passatgers i tripulants. Tot i que hi va haver algun element de sort per sobreviure, sembla ser que alguns grups de persones tenien més probabilitats de sobreviure que d'altres.

1.2. Objectiu

L'objectiu d'aquesta pràctica és la creació final d'un o varis models predictius que responguin a la pregunta: **"quin tipus de persones tenien més probabilitats de sobreviure?"**. Per això, caldrà primer netejar i analitzar les dades dels passatgers a bord del RMS Titanic (*i.e.* nom, edat, sexe, classe socioeconòmica, etc.) abans de construir els models per a predir quins passatgers van sobreviure a l'enfonsament del Titanic.

1.3. Descripció del dataset

El dataset que utilitzarem en aquesta pràctica és *Titanic: Machine Learning from Disaster*, disponible clicant a *aquí*.

Aquest conjunt de dades està compost per diversos atributs/característiques dels passatgers del titanic distribuïts en 2 fitxers: un dataset per entrenar els models i un altre per a testar-los. A més, s'incou un fitxer amb la predicció de la supervivencia dels passatgers (codificat com a 0: mort o 1: sobreviu) que assumeix que totes les dones, i només les dones, sobreviuen. Obviament, aquestes prediccions són irreals pero ens serveixen com a exemple de com hauria de ser el fitxer final amb les prediccions de supervivencia en cas de que es participés a una de les competició de *Kaggle*. És per aquest motiu que les dades es proporcionen ja semparades en un dataset d'entrenament i un dataset de testeig, ja que així tots els membres que participen a la cometició de la web Kaggle parteixen de la mateixa informació i dades, i per tant es poden comparar entre ells els resultats obtinguts amb els diferents models implentats.

Els fitxers que componen el dataset són:

- **train.csv**: Conté totes les dades i variables d'un subgrup dels passatgers a bord del RMS Titanic (891 passatgers i tripulants), a més de la variable que ens indica si aquell passatger va morir o sobreviure. Aquest dataset serà el que s'utilitzarà per l'entrenament d'un model.
- **test.csv**: Conté totes les dades i variables d'un subgrup més petit que l'anterior dels passatgers a bord del RMS Titanic (en aquest cas 418 passatgers i tripulants) amb l'excepció de que aquest dataset no conté la variable que ens indica si el passatger va sobreviure o no. Aquestes dades s'utilitzaran per a poder testejar els models creats amb les dades del dataset d'entrenament.
- **gender_submission.csv**: Conté la classe dels passatgers (si sobreviuen o no) vinculada a l'identificador de cada passatger, assumint que totes les dones, i només les dones, haguessin sobreviscut.

Cada passatger conté informació de les següents variables:

Table 1: Data diccionary: resum de les variables del dataset 'Titanic: Machine Learning from Disaster'.

	Variables	Definició	Codificació
	PassangerID	Número identificador del passatger	
	Survived	Enter que indica si el passatger va sobreviure l'enfonsament o no	0 = No, 1 = Sí
	Pclass	Enter que indica el tipus de tiquet del passatger	1 = 1a classe, 2 = 2a classe, 3 = 3a classe
	Name	Títol/Nom del passatger	
	Sex	Sexe del passatger	
	Age	Edat del passatger (anys)	
	SibSp	Número de cònjuges i germans del passatger a bord	
	Parch	Número de pares i fills del passatger a bord	
	Ticket	Número del tiquet del passatger	
	Fare	Preu pagat/Tarifa del viatge	
	Cabin	Número de cabina	C = Cherbourg, Q = Queenstown, S = Southampton
	Embarked	Port en el que ha embarcat el passatger	PassangerID

Notes a tenir en compte: * La variable **pclass** ens serveix com a indicador indirecte del nivell socioeconomic del passatger (alt: 1a classe, mitjà: 2a classe, baix: 3a classe). * L'edat dels menors d'1 any està codificada com la fracció d'any que tenen. Les edats és mostren com a fraccions on en alguns casos s'indica' el 'i mig' (e.g. 35.50 = 35 anys i mig) * Alguns nens viatjaven amb una mainadera, per tant és possible que per algun d'ells 'parch' sigui 0.

2. Integració i comprobació

Un cop les dades s'han descarregat des de l'enllaç proporcionat, carreguem les dades a l'entorn de treball.

```
train_raw <- read.csv("data/train.csv", sep=',', stringsAsFactors = FALSE)
test_raw <- read.csv("data/test.csv", sep=',', stringsAsFactors = FALSE)
test_class_raw <- read.csv("data/gender_submission.csv", sep=',', stringsAsFactors = FALSE)
```

Primer inspeccionem els datasets

```
head(train_raw)
```

```
##   PassengerId Survived Pclass
## 1           1        0      3
## 2           2        1      1
## 3           3        1      3
## 4           4        1      1
## 5           5        0      3
## 6           6        0      3
##                                                 Name    Sex Age SibSp Parch
## 1                            Braund, Mr. Owen Harris   male  22     1     0
## 2 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female  38     1     0
## 3                             Heikkinen, Miss. Laina female  26     0     0
## 4       Futrelle, Mrs. Jacques Heath (Lily May Peel) female  35     1     0
## 5                           Allen, Mr. William Henry   male  35     0     0
## 6                                   Moran, Mr. James   male  NA     0     0
##             Ticket    Fare Cabin Embarked
## 1        A/5 21171  7.2500              S
## 2         PC 17599 71.2833   C85        C
## 3 STON/O2. 3101282  7.9250              S
## 4           113803 53.1000  C123        S
## 5           373450  8.0500              S
## 6           330877  8.4583              Q
```

```
head(test_raw)
```

```
##   PassengerId Pclass                                          Name    Sex  Age
## 1         892      3                              Kelly, Mr. James   male 34.5
## 2         893      3              Wilkes, Mrs. James (Ellen Needs) female 47.0
## 3         894      2                     Myles, Mr. Thomas Francis   male 62.0
## 4         895      3                              Wirz, Mr. Albert   male 27.0
## 5         896      3 Hirvonen, Mrs. Alexander (Helga E Lindqvist) female 22.0
## 6         897      3                    Svensson, Mr. Johan Cervin   male 14.0
##   SibSp Parch  Ticket    Fare Cabin Embarked
## 1     0     0  330911  7.8292              Q
## 2     1     0  363272  7.0000              S
## 3     0     0  240276  9.6875              Q
## 4     0     0  315154  8.6625              S
## 5     1     1 3101298 12.2875              S
## 6     0     0    7538  9.2250              S
```

```
head(test_class_raw)
```

```
##   PassengerId Survived
## 1         892        0
## 2         893        1
## 3         894        0
```

```
## 4          895        0
## 5          896        1
## 6          897        0
```

Primer de tot extreurem la variable 'Survived' del dataset d'entrenament en un format igual al dataset 'gender_submission.csv' ja que aquesta no ens farà falta de moment i així podem integrar els dos datasets en un de sol per a dur a terme la neteja, inspecció i analisis en un sol dataset.

```
train_predictions<-train_raw[,c("PassengerId","Survived")]
train_raw<-train_raw[,-2]
```

Comprovem que les dades s'han carregat correctamen i les inspeccionem. Dataset d'entrenament:

```
# Comprovem que les dimensions del dataset siguin les esperades
dim(train_raw)
```

```
## [1] 891  11
```

```
# Comprovem en quina classe es troben les variables
sapply(train_raw,class)
```

```
## PassengerId      Pclass        Name         Sex         Age       SibSp
##   "integer"   "integer" "character" "character"   "numeric"   "integer"
##       Parch      Ticket        Fare       Cabin    Embarked
##   "integer" "character"   "numeric" "character" "character"
```

```
# Comprovem l'estructura
str(train_raw)
```

```
## 'data.frame':    891 obs. of  11 variables:
##  $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Pclass     : int  3 1 3 1 3 3 1 3 3 2 ...
##  $ Name       : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)"
##  $ Sex        : chr  "male" "female" "female" "female" ...
##  $ Age        : num  22 38 26 35 35 NA 54 2 27 14 ...
##  $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
##  $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
##  $ Ticket     : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
##  $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
##  $ Cabin      : chr  "" "C85" "" "C123" ...
##  $ Embarked   : chr  "S" "C" "S" "S" ...
```

```
# Inspeccionem alguns parametres bàsics
summary(train_raw)
```

```
##   PassengerId        Pclass          Name              Sex
##  Min.   :  1.0   Min.   :1.000   Length:891         Length:891
##  1st Qu.:223.5   1st Qu.:2.000   Class :character   Class :character
##  Median :446.0   Median :3.000   Mode  :character   Mode  :character
##  Mean   :446.0   Mean   :2.309
##  3rd Qu.:668.5   3rd Qu.:3.000
##  Max.   :891.0   Max.   :3.000
##
##       Age            SibSp           Parch           Ticket
##  Min.   : 0.42   Min.   :0.000   Min.   :0.0000   Length:891
##  1st Qu.:20.12   1st Qu.:0.000   1st Qu.:0.0000   Class :character
##  Median :28.00   Median :0.000   Median :0.0000   Mode  :character
##  Mean   :29.70   Mean   :0.523   Mean   :0.3816
```

```
##  3rd Qu.:38.00   3rd Qu.:1.000   3rd Qu.:0.0000
##  Max.   :80.00   Max.   :8.000   Max.   :6.0000
##  NA's   :177
##       Fare           Cabin             Embarked
##  Min.   :  0.00   Length:891         Length:891
##  1st Qu.:  7.91   Class :character   Class :character
##  Median : 14.45   Mode  :character   Mode  :character
##  Mean   : 32.20
##  3rd Qu.: 31.00
##  Max.   :512.33
##
```

```
# Inspeccionem les dades
glimpse(train_raw)
```

```
## Rows: 891
## Columns: 11
## $ PassengerId <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, ...
## $ Pclass      <int> 3, 1, 3, 1, 3, 3, 1, 3, 3, 2, 3, 1, 3, 3, 3, 2, 3, 2, 3...
## $ Name        <chr> "Braund, Mr. Owen Harris", "Cumings, Mrs. John Bradley ...
## $ Sex         <chr> "male", "female", "female", "female", "male", "male", "...
## $ Age         <dbl> 22, 38, 26, 35, 35, NA, 54, 2, 27, 14, 4, 58, 20, 39, 1...
## $ SibSp       <int> 1, 1, 0, 1, 0, 0, 0, 3, 0, 1, 1, 0, 0, 1, 0, 0, 4, 0, 1...
## $ Parch       <int> 0, 0, 0, 0, 0, 0, 0, 1, 2, 0, 1, 0, 0, 5, 0, 0, 1, 0, 0...
## $ Ticket      <chr> "A/5 21171", "PC 17599", "STON/O2. 3101282", "113803", ...
## $ Fare        <dbl> 7.2500, 71.2833, 7.9250, 53.1000, 8.0500, 8.4583, 51.86...
## $ Cabin       <chr> "", "C85", "", "C123", "", "", "E46", "", "", "", "G6",...
## $ Embarked    <chr> "S", "C", "S", "S", "S", "Q", "S", "S", "S", "C", "S", ...
```

Dataset de testeig:

```
# Comprovem que les dimensions del dataset siguin les esperades
dim(test_raw)
```

```
## [1] 418  11
```

```
# Comprovem en quina classe es troben les variables
sapply(test_raw,class)
```

```
## PassengerId      Pclass        Name         Sex         Age       SibSp
##   "integer"   "integer" "character" "character"   "numeric"   "integer"
##       Parch      Ticket        Fare       Cabin    Embarked
##   "integer" "character"   "numeric" "character" "character"
```

```
# Comprovem l'estructura
str(test_raw)
```

```
## 'data.frame':    418 obs. of  11 variables:
##  $ PassengerId: int  892 893 894 895 896 897 898 899 900 901 ...
##  $ Pclass     : int  3 3 2 3 3 3 3 2 3 3 ...
##  $ Name       : chr  "Kelly, Mr. James" "Wilkes, Mrs. James (Ellen Needs)" "Myles, Mr. Thomas Francis
##  $ Sex        : chr  "male" "female" "male" "male" ...
##  $ Age        : num  34.5 47 62 27 22 14 30 26 18 21 ...
##  $ SibSp      : int  0 1 0 0 1 0 0 1 0 2 ...
##  $ Parch      : int  0 0 0 0 1 0 0 1 0 0 ...
##  $ Ticket     : chr  "330911" "363272" "240276" "315154" ...
##  $ Fare       : num  7.83 7 9.69 8.66 12.29 ...
##  $ Cabin      : chr  "" "" "" "" ...
```

```
##  $ Embarked   : chr  "Q" "S" "Q" "S" ...
```

```
# Inspeccionem alguns parametres bàsics
summary(test_raw)
```

```
##   PassengerId         Pclass          Name               Sex
##  Min.   : 892.0   Min.   :1.000   Length:418         Length:418
##  1st Qu.: 996.2   1st Qu.:1.000   Class :character   Class :character
##  Median :1100.5   Median :3.000   Mode  :character   Mode  :character
##  Mean   :1100.5   Mean   :2.266
##  3rd Qu.:1204.8   3rd Qu.:3.000
##  Max.   :1309.0   Max.   :3.000
##
##       Age            SibSp            Parch           Ticket
##  Min.   : 0.17   Min.   :0.0000   Min.   :0.0000   Length:418
##  1st Qu.:21.00   1st Qu.:0.0000   1st Qu.:0.0000   Class :character
##  Median :27.00   Median :0.0000   Median :0.0000   Mode  :character
##  Mean   :30.27   Mean   :0.4474   Mean   :0.3923
##  3rd Qu.:39.00   3rd Qu.:1.0000   3rd Qu.:0.0000
##  Max.   :76.00   Max.   :8.0000   Max.   :9.0000
##  NA's   :86
##       Fare            Cabin             Embarked
##  Min.   :  0.000   Length:418         Length:418
##  1st Qu.:  7.896   Class :character   Class :character
##  Median : 14.454   Mode  :character   Mode  :character
##  Mean   : 35.627
##  3rd Qu.: 31.500
##  Max.   :512.329
##  NA's   :1
```

```
# Inspeccionem les dades
glimpse(test_raw)
```

```
## Rows: 418
## Columns: 11
## $ PassengerId <int> 892, 893, 894, 895, 896, 897, 898, 899, 900, 901, 902, ...
## $ Pclass      <int> 3, 3, 2, 3, 3, 3, 3, 2, 3, 3, 3, 1, 1, 2, 1, 2, 2, 3, 3...
## $ Name        <chr> "Kelly, Mr. James", "Wilkes, Mrs. James (Ellen Needs)",...
## $ Sex         <chr> "male", "female", "male", "male", "female", "male", "fe...
## $ Age         <dbl> 34.5, 47.0, 62.0, 27.0, 22.0, 14.0, 30.0, 26.0, 18.0, 2...
## $ SibSp       <int> 0, 1, 0, 0, 1, 0, 0, 1, 0, 2, 0, 0, 1, 1, 1, 1, 0, 0, 1...
## $ Parch       <int> 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ Ticket      <chr> "330911", "363272", "240276", "315154", "3101298", "753...
## $ Fare        <dbl> 7.8292, 7.0000, 9.6875, 8.6625, 12.2875, 9.2250, 7.6292...
## $ Cabin       <chr> "", "", "", "", "", "", "", "", "", "", "", "", "B45", ...
## $ Embarked    <chr> "Q", "S", "Q", "S", "S", "S", "Q", "S", "C", "S", "S", ...
```

Les dimensions són les esperades i el format és equivalent entre els dos datasets.

```
head(train_raw)
```

```
##   PassengerId Pclass                                                Name    Sex
## 1           1      3                             Braund, Mr. Owen Harris   male
## 2           2      1 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female
## 3           3      3                              Heikkinen, Miss. Laina female
## 4           4      1        Futrelle, Mrs. Jacques Heath (Lily May Peel) female
## 5           5      3                            Allen, Mr. William Henry   male
```

```
## 6           6      3                                     Moran, Mr. James   male
##   Age SibSp Parch           Ticket    Fare Cabin Embarked
## 1  22     1     0        A/5 21171  7.2500              S
## 2  38     1     0         PC 17599 71.2833   C85        C
## 3  26     0     0 STON/O2. 3101282  7.9250              S
## 4  35     1     0           113803 53.1000  C123        S
## 5  35     0     0           373450  8.0500              S
## 6  NA     0     0           330877  8.4583              Q
```

```
head(test_raw)
```

```
##   PassengerId Pclass                                          Name    Sex  Age
## 1         892      3                              Kelly, Mr. James   male 34.5
## 2         893      3              Wilkes, Mrs. James (Ellen Needs) female 47.0
## 3         894      2                     Myles, Mr. Thomas Francis   male 62.0
## 4         895      3                               Wirz, Mr. Albert   male 27.0
## 5         896      3 Hirvonen, Mrs. Alexander (Helga E Lindqvist) female 22.0
## 6         897      3                     Svensson, Mr. Johan Cervin   male 14.0
##   SibSp Parch  Ticket    Fare Cabin Embarked
## 1     0     0  330911  7.8292              Q
## 2     1     0  363272  7.0000              S
## 3     0     0  240276  9.6875              Q
## 4     0     0  315154  8.6625              S
## 5     1     1 3101298 12.2875              S
## 6     0     0    7538  9.2250              S
```

Integrem les dades dels dos datasets en un de sol.

```
dataset<-rbind(train_raw,test_raw)
head(dataset)
```

```
##   PassengerId Pclass                                              Name    Sex
## 1           1      3                           Braund, Mr. Owen Harris   male
## 2           2      1 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female
## 3           3      3                            Heikkinen, Miss. Laina female
## 4           4      1    Futrelle, Mrs. Jacques Heath (Lily May Peel) female
## 5           5      3                          Allen, Mr. William Henry   male
## 6           6      3                                 Moran, Mr. James   male
##   Age SibSp Parch           Ticket    Fare Cabin Embarked
## 1  22     1     0        A/5 21171  7.2500              S
## 2  38     1     0         PC 17599 71.2833   C85        C
## 3  26     0     0 STON/O2. 3101282  7.9250              S
## 4  35     1     0           113803 53.1000  C123        S
## 5  35     0     0           373450  8.0500              S
## 6  NA     0     0           330877  8.4583              Q
```

```
dim(dataset)
```

```
## [1] 1309   11
```

A partir de la informació observada decidim canviar les classes d'algunes variables.

```
dataset <- within(dataset, {
  Pclass <- factor(Pclass)
  Sex <- factor(Sex)
  Age <- as.integer(Age)
  Embarked <- factor(Embarked)
})
```

Tornem a inspeccionar:

```
# Inspeccionem les dades
glimpse(dataset)
```

```
## Rows: 1,309
## Columns: 11
## $ PassengerId <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, ...
## $ Pclass      <fct> 3, 1, 3, 1, 3, 3, 1, 3, 3, 2, 3, 1, 3, 3, 3, 2, 3, 2, 3...
## $ Name        <chr> "Braund, Mr. Owen Harris", "Cumings, Mrs. John Bradley ...
## $ Sex         <fct> male, female, female, female, male, male, male, male, f...
## $ Age         <int> 22, 38, 26, 35, 35, NA, 54, 2, 27, 14, 4, 58, 20, 39, 1...
## $ SibSp       <int> 1, 1, 0, 1, 0, 0, 0, 3, 0, 1, 1, 0, 0, 1, 0, 0, 4, 0, 1...
## $ Parch       <int> 0, 0, 0, 0, 0, 0, 0, 1, 2, 0, 1, 0, 0, 5, 0, 0, 1, 0, 0...
## $ Ticket      <chr> "A/5 21171", "PC 17599", "STON/O2. 3101282", "113803", ...
## $ Fare        <dbl> 7.2500, 71.2833, 7.9250, 53.1000, 8.0500, 8.4583, 51.86...
## $ Cabin       <chr> "", "C85", "", "C123", "", "", "E46", "", "", "", "G6",...
## $ Embarked    <fct> S, C, S, S, S, Q, S, S, S, C, S, S, S, S, S, S, Q, S, S...
```

Observacions:

- Les dimensions són les esperades.
- De la variable 'Name' es poden extreure els títols d'algunes persones.
- La variable 'Sex' es podria codificar de forma binaria.
- La variable 'Fare' sembla representar el preu total pagat alhora de comprar més d'un tiquet junts (e.g. families) envés del tiquet individual.
- Podem crear 'dummy' variables per a les variables 'Pclass', 'Sex' i 'Embarked'
- De la variable 'cabin' es pot extreure la lletra que segurament representa diferentes zones del vaixell.
- Dels resultats de la funció summary() i glimpse() es pot veure com tenim varis valors nuls o mencants en les variables: Age, Fare, Cabin i Embarked

3. Neteja de dades

3.1. Gestió de valors buits o nul

Creem una funció que indica les variables que contenen valors nul i l'executem passant-li el dataset que hem carregat.

```
has_na <- function(dades) {
  no_na <- TRUE
  for (i in names(dades)) {
    a <- sum(is.na(dades[[i]]))
    if (a != 0) {
      no_na <- FALSE
      print(paste("La variable ", i, " té ", a, " valors nul"))
    }
    else if (a == 0 & is.character(dades[[i]])) {
      b <- length(dades[[i]][which(dades[[i]]=="")])
      if (b != 0) {
        no_na <- FALSE
        print(paste("La variable ", i, " té ", b, " valors buits"))
      }
    }
  }
  if (no_na) {
    print("No hi ha cap variable amb valors nul")
```

```
  }
}
```

Un cop creada la funció l'executem per saber quines variables tenen valors nul en els diferents datasets. Dataset d'entrenament:

```
has_na(dataset)
```

```
## [1] "La variable  Age  té  263  valors nul"
## [1] "La variable  Fare  té  1  valors nul"
## [1] "La variable  Cabin  té  1014  valors buits"
```

La variable Cabin i Embarket també tenen valors nuls pero com que son empty strings la funcio no els reconeix. ** S'hauria de fer aixo: https://www.kaggle.com/c/titanic/discussion/62321

3.2. Creació de variables

3.2.1. Creació variable 'Title'

A partir del nom dels passatgers es poden extreure els titols d'algun d'ells.

```
# Extreiem el 'titol' de cada passatger. Com que el format dels nomes sempre es el mateix podem seguir u
dataset$title <- str_sub(dataset$Name, str_locate(dataset$Name, ",")[ , 1] + 2, str_locate(dataset$Name

# combines els títols dels passatgers en grups segons si considerem que formen part de la noblesa (i.e.
# Titols nobles per homes
names_noblesa <- c("Capt", "Col", "Don", "Dr", "Jonkheer", "Major", "Rev", "Sir","Lady", "Mlle", "Mme",
dataset$title[dataset$title %in% names_noblesa] <- "noble"
dataset$title<-factor(dataset$title)
# Comprobem el resultat final
table(dataset$title)
```

```
##
## Master   Miss     Mr    Mrs  noble
##     61    260    757    197     34
```

3.2.2. Creació variable 'Zona'

De la variable 'cabin' extreiem la lletra que segurament representa la zona del vaixell on es troba la cabina.

```
Zona <- dataset %>%
  select(Cabin) %>%
  mutate(Zona = factor(str_extract(Cabin, pattern = "^.")))

dataset$Zona<-Zona$Zona
```

3.2.3. Creació variable 'Familiy_Size'

El tamany de les families es pot inferir a partir de les variables 'SibSp' i 'Parch'. A més, tota aquella gent que comparteixi cognom és probable que formin part de la mateixa familia. Dataset d'entrenament:

```
#Creem la variable 'Familiy_name"
dataset$Family_name <- str_replace(string = dataset$Name, pattern = ",.*", replacement = "")

#Visualitzem les dades
dataset %>%
  select(Name, Family_name) %>%
  head(10)
```

```
##                                                  Name Family_name
## 1                             Braund, Mr. Owen Harris      Braund
## 2  Cumings, Mrs. John Bradley (Florence Briggs Thayer)     Cumings
## 3                              Heikkinen, Miss. Laina   Heikkinen
## 4        Futrelle, Mrs. Jacques Heath (Lily May Peel)    Futrelle
## 5                            Allen, Mr. William Henry       Allen
## 6                                    Moran, Mr. James       Moran
## 7                             McCarthy, Mr. Timothy J    McCarthy
## 8                      Palsson, Master. Gosta Leonard     Palsson
## 9    Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)     Johnson
## 10                  Nasser, Mrs. Nicholas (Adele Achem)      Nasser
```

```
#Ceem la variable 'Family_size' (sumem 1 per assegurar-nos de que el passatger també es té en compte en
dataset <- dataset %>%
  mutate(Family_size = SibSp + Parch + 1)

#Visualitzem les dades
dataset %>%
  select(Family_name, SibSp, Parch, Family_size) %>%
   arrange(Family_name)%>%
  head(10)
```

```
##    Family_name SibSp Parch Family_size
## 1       Abbing     0     0           1
## 2       Abbott     1     1           3
## 3       Abbott     1     1           3
## 4       Abbott     0     2           3
## 5     Abelseth     0     0           1
## 6     Abelseth     0     0           1
## 7      Abelson     1     0           2
## 8      Abelson     1     0           2
## 9  Abrahamsson     0     0           1
## 10     Abrahim     0     0           1
```

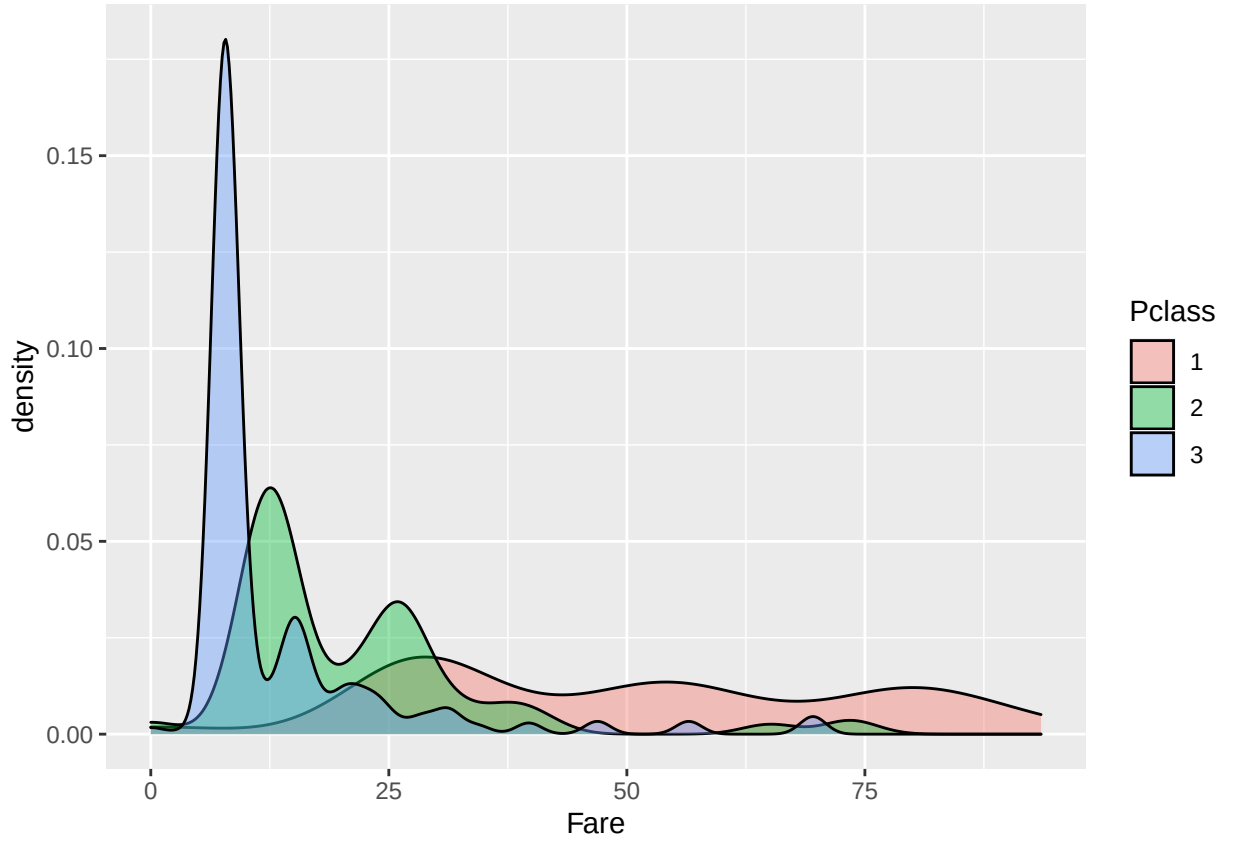
Categorizem el tamany de les families segons si el passatger va sol, en familia/grup gran o petit

```
dataset <- dataset %>%
  mutate(FamilyCat = factor(case_when(Family_size == 1 ~ 'sol'
                             ,Family_size > 1 & Family_size < 5 ~ 'petita'
                             ,Family_size >= 5 ~ 'gran')))
```

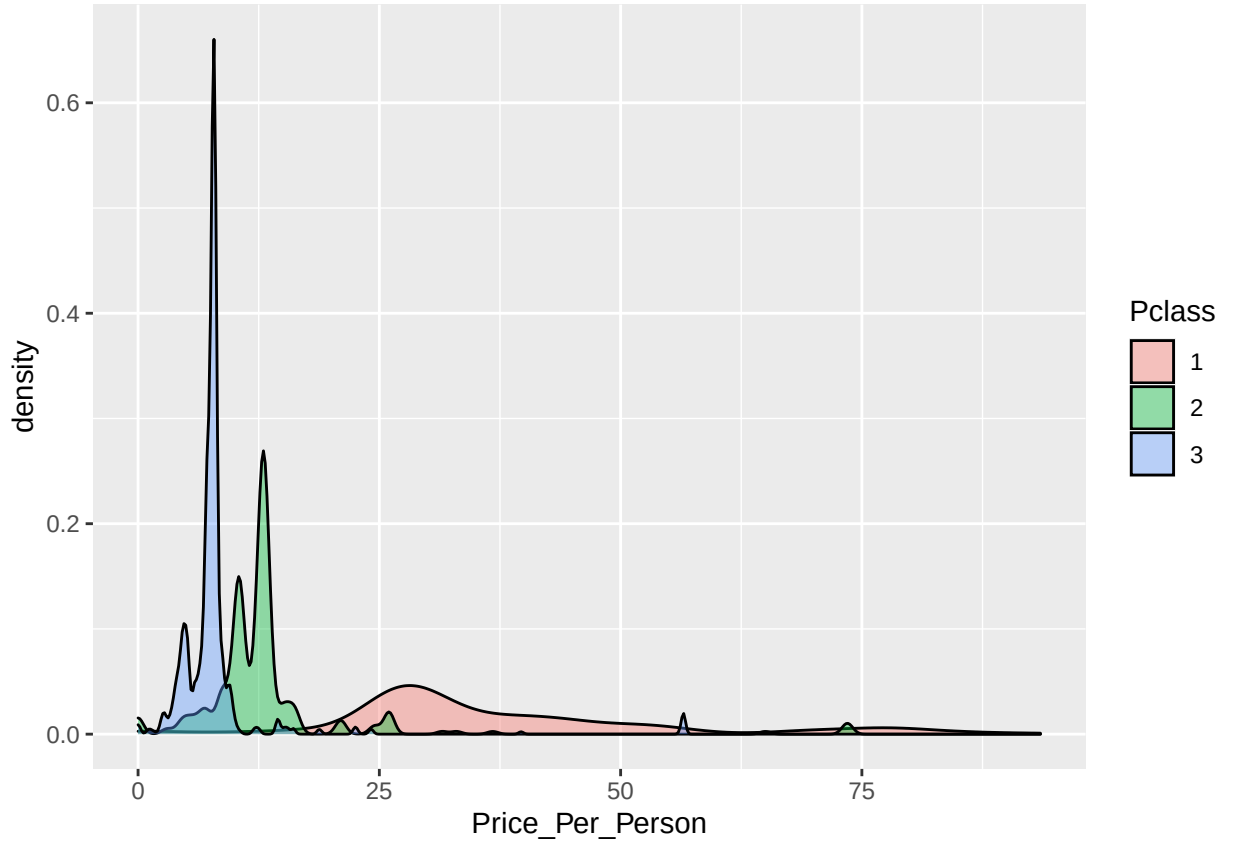
3.2.4. Creació variable 'Price_Person'

```
dataset$Price_Per_Person<-dataset$Fare/dataset$Family_size
```

```
# Use semi-transparent fill
p_fare<-ggplot(dataset[which(dataset$Fare<100),], aes(x=Fare, fill=Pclass)) +
  geom_density(alpha=0.4)
p_fare
```




```
# Use semi-transparent fill
p_priceperson<-ggplot(dataset[which(dataset$Price_Per_Person<100),], aes(x=Price_Per_Person, fill=Pclass
  geom_density(alpha=0.4)
p_priceperson
```



3.3. Exploració i transformació de variables

3.4. Gestió de valors extrems (outliers)

També cal observar els valors extrems de les variables numèriques i considerar si es corresponen a valors errònis que cal evitar o bé són casos extrems que convé mantenir. Com hem vist enteriorment, tenim 4 variables a considerar: Age, SibSp, Parch i Fare.

Per començar l'anàlisi de valors atípics hem creat una funció que mostra els diagrames de caixa de les diferents variables numèriques:

```
mostrar_diag_caixa <- function(dades, variables_numeriques) {
  mida <- ceiling(sqrt(length(variables_numeriques)))
  par(mfrow=c(2,2))
  for (i in variables_numeriques) {
    boxplot(dades[[i]], main=i, horizontal = TRUE)
  }
}
```

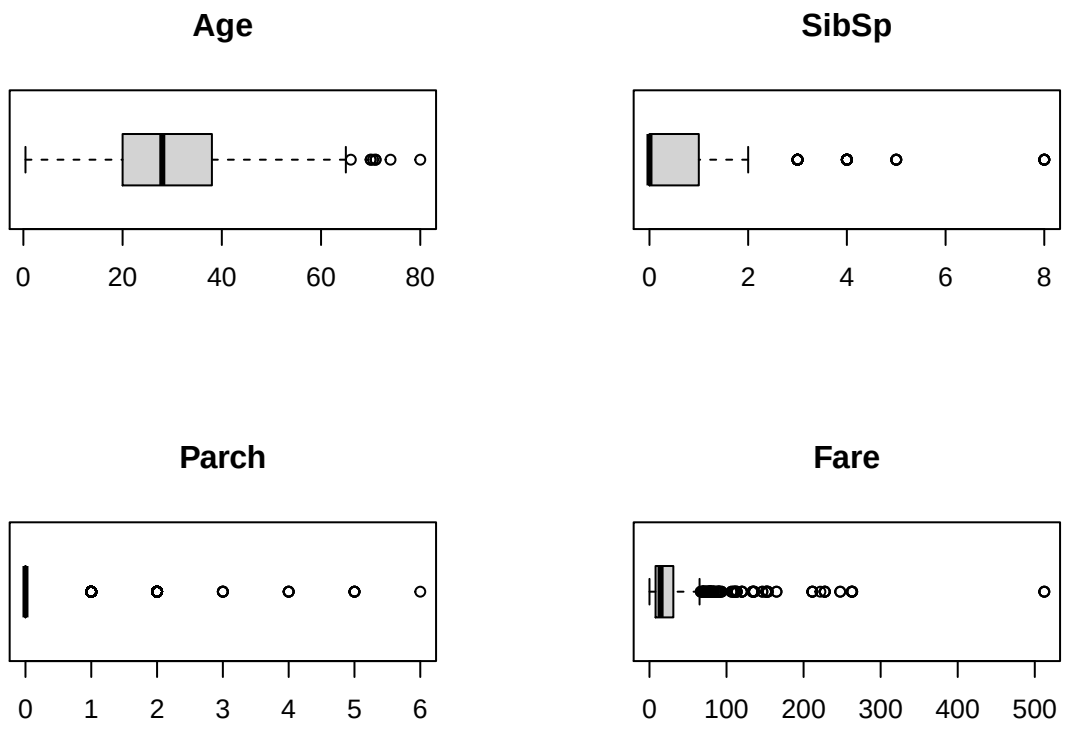
I una altra que indica textualment els valors atípics d'una variable:

```
deteccio_valors_atipics <- function(dades,variables_numeriques) {
  for (i in variables_numeriques) {
    valors_atipics <- boxplot.stats(dades[[i]])$out
    if (length(valors_atipics != 0)) {
      print(paste("La variable", i, "té els valors atípics: "))
      print(valors_atipics)
    }
```

```
  }
}
```

Un cop creades les funcions, les cridem passant-los hi les dades de cada fitxer i un vector amb el nom de les variables numèriques que contenen:

```
mostrar_diag_caixa(train_raw,c("Age","SibSp","Parch","Fare"))
```




```
deteccio_valors_atipics(train_raw,c("Age","SibSp","Fare"))
```

```
## [1] "La variable Age té els valors atípics: "
## [1] 66.0 71.0 70.5 71.0 80.0 70.0 70.0 74.0
## [1] "La variable SibSp té els valors atípics: "
##  [1] 3 4 3 3 4 5 3 4 5 3 3 4 8 4 4 3 8 4 8 3 4 4 4 4 8 3 3 5 3 5 3 4 4 3 3 5 4 3
## [39] 4 8 4 3 4 8 4 8
## [1] "La variable Fare té els valors atípics: "
##   [1]  71.2833 263.0000 146.5208  82.1708  76.7292  80.0000  83.4750  73.5000
##   [9] 263.0000  77.2875 247.5208  73.5000  77.2875  79.2000  66.6000  69.5500
##  [17]  69.5500 146.5208  69.5500 113.2750  76.2917  90.0000  83.4750  90.0000
##  [25]  79.2000  86.5000 512.3292  79.6500 153.4625 135.6333  77.9583  78.8500
##  [33]  91.0792 151.5500 247.5208 151.5500 110.8833 108.9000  83.1583 262.3750
##  [41] 164.8667 134.5000  69.5500 135.6333 153.4625 133.6500  66.6000 134.5000
##  [49] 263.0000  75.2500  69.3000 135.6333  82.1708 211.5000 227.5250  73.5000
##  [57] 120.0000 113.2750  90.0000 120.0000 263.0000  81.8583  89.1042  91.0792
##  [65]  90.0000  78.2667 151.5500  86.5000 108.9000  93.5000 221.7792 106.4250
##  [73]  71.0000 106.4250 110.8833 227.5250  79.6500 110.8833  79.6500  79.2000
##  [81]  78.2667 153.4625  77.9583  69.3000  76.7292  73.5000 113.2750 133.6500
```

```
##  [89]  73.5000 512.3292  76.7292 211.3375 110.8833 227.5250 151.5500 227.5250
##  [97] 211.3375 512.3292  78.8500 262.3750  71.0000  86.5000 120.0000  77.9583
## [105] 211.3375  79.2000  69.5500 120.0000  93.5000  80.0000  83.1583  69.5500
## [113]  89.1042 164.8667  69.5500  83.1583
```

3.4. Datasets final

```
# El dataset final
dataset_final<-merge(dataset,train_predictions,by = "PassengerId", all = T)

# El dataset final d'entrenament
dataset_train<-dataset_final[which(!(is.na(dataset_final$Survived))),]
dim(dataset_train)
```

```
## [1] 891  18
```

```
# El dataset final de test
dataset_test<-dataset_final[which(is.na(dataset_final$Survived)),]
dim(dataset_test)
```

```
## [1] 418  18
```

4. Anàlisis de les dades

5. Resultats finals

6. Conclusions

7. Bibliografia

- Gibergans, J. (2019). Regressió lineal simple. Editorial UOC.
- Gibergans, J. (2019). Regressió lineal múltiple. Editorial UOC.
- Guillén, M., Alonso, M. T. (2019). Models de regressió logística. Editorial UOC.
- Liviano, D., Pujol, M. (2019). Models de regressió i anàlisi multivariant amb R-Commander. Editorial UOC.

Integrem les dades del dataset de test amb les dades de les seves corresponents prediccions de supervivencia si les dones, i només les dones, haguessin sobreviscut.

```
names(test_class_raw)[2]<-"AllWomenSurvived"
test_dataset<-merge(test_raw,test_class_raw,by = 'PassengerId')
head(test_dataset)
```

```
##   PassengerId Pclass                                          Name    Sex  Age
## 1         892      3                              Kelly, Mr. James   male 34.5
## 2         893      3              Wilkes, Mrs. James (Ellen Needs) female 47.0
## 3         894      2                     Myles, Mr. Thomas Francis   male 62.0
## 4         895      3                              Wirz, Mr. Albert   male 27.0
## 5         896      3 Hirvonen, Mrs. Alexander (Helga E Lindqvist) female 22.0
## 6         897      3                    Svensson, Mr. Johan Cervin   male 14.0
##   SibSp Parch  Ticket    Fare Cabin Embarked AllWomenSurvived
## 1     0     0  330911  7.8292              Q                0
## 2     1     0  363272  7.0000              S                1
## 3     0     0  240276  9.6875              Q                0
## 4     0     0  315154  8.6625              S                0
## 5     1     1 3101298 12.2875              S                1
## 6     0     0    7538  9.2250              S                0
```

..... Compare the passenger Class with their position on the Deck table(cabin_deck$Deck, $cabin_deck$Pcl

#Calculate the correlation of the 2 variables cabin_deck %>% select(Pclass, Deck) %>% map_df(as.numeric) %>% cor(., use = 'complete.obs',method = 'pearson')

.....

..... #plot the data
ggplot(missing_fare_subset, aes(x = Fare)) + geom_density(fill = 'forestgreen') + scale_x_continuous(label = dollar_format(), breaks = seq(0, 60, by = 10)) + ggtitle("Density Distribution of Fares for 3rd Class Male Passengers Embarking from Southampton", subtitle = "n = 365") .....

Mètodes de predicció Arbres de decisió Mètodes d'agregació