Pràctica 2: Neteja i anàlisi de les dades

Mireia Olivella i Gabriel Izquierdo

4 de maig de 2020

Índex

1 Descripció del dataset

El conjunt de dades escollit recull informació dels passatgers del titanic, en el que es pot analitzar la supervivència i les característiques d'aquests. Aquest conjunt de dades s'ha obtingut de la web de Kaggle. S'hi pot accedir a partir de l'enllaç que es mostra a continuació:

https://www.kaggle.com/c/titanic

El conjunt de dades utilitzat està format per 1309 registres amb 12 atributs dividit en 2 fitxers CSV, un de *train* i un de *test*, ja que aquest conjunt de dades està preparat per ser utilitzat per tasques de predicció. Els atributs d'aquest conjunt de dades són els següents:

- **passengerId**: identificador dels registres del dataset.
- **survived**: indica si el passatger va sobreviure (0=No, 1=Sí).
- **pclass**: indica la classe en la que viatjava el passatger (1=1a, 2=2a, 3=3a).
- **name**: nom del passatger.
- **sex**: gènere del passatger (*female* o *male*).
- **age**: edat del passatger.
- **sibsp**: número de germans i cònjugues a bord del Titànic.
- **parch**: número de pares i fills a bord del Titànic.
- **ticket**: número del bitllet
- **fare**: preu de compra del bitllet.
- **cabin**: número de cabina on viatjava el passatger.
- **embarked**: port on va embarcar el passatger (C=Cherbourg, Q=Queenstown, S=Southampton).

Aquest conjunt de dades és important perquè representa les dades d'un dels naufragis més infames de la història. A més, ens permet abastir tots els aspectes importants a tenir en compte a l'hora de dur a terme aquesta pràctica.

La pregunta que intenta respondre és la de quins són els factors que van afavorir a un passatger sobreviure al naufragi. Si bé hi havia un element de sort en la supervivència dels passatgers, sembla que alguns grups de persones tenien més probabilitats de sobreviure que d'altres.

2 Neteja de les dades

Abans de començar amb la neteja de les dades, procedim a realitzar les lectures dels fitxers en format CSV en el que es troben. El procediment és el de carregar la informació dels tres fitxers i unir-les posteriorment.

En la secció anterior s'ha parlat de dos fitxers de tipus CSV, i ara se n'ha parlat de tres. Això es deu a que al fitxer `test.csv` li falta un atribut respecte al fitxer `train.csv`, que és el de `Survived`. La informació referent a la supervivència dels passatgers del fitxer `test.csv` es troba en un altre fitxer anomenat `gender_submission.csv` que té només dues columnes: `PassengerId` i `Survived`.

El primer que fem és carregar la informació de tots els fitxers CSV. Després fem un *merge* de les dades de `test.csv` i `gender_submission.csv` utilitzant la funció `merge` amb l'atribut `PassengerId` com a clau comuna entre les dues taules. Per acabar s'uneixen totes les dades de *train* i *test* en un sol *dataframe* utilitzant la funció `rbind`.

```
# Lectura de les dades
titanic_train <- read.csv("../data/train.csv")
titanic_test <- read.csv("../data/test.csv")
titanic_gender_submission <- read.csv("../data/gender_submission.csv")
titanic_test <- merge(titanic_test, titanic_gender_submission, by="PassengerId")
titanic_data <- rbind(titanic_train, titanic_test)
head(titanic_data)
```

```
##   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
```

```
# Tipus de dada assignat a cada camp
sapply(titanic_data, function(x) class(x))
```

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

Podem observar que els tipus de dades assignats automàticament per R a les nostres variables no s'acaben de correspondre amb el domini d'aquestes. Aquest és el cas de l'atribut `Survived`. R detecta que es tracta d'un *integer* quan en realitat es tracta d'un *factor*, pel que procedim a assignar-li el tipus que nosaltres volem.

```
# Canvi del tipus del camp 'Survived'
titanic_data$Survived <- factor(titanic_data$Survived)
```

2.1 Selecció de les dades d'interès

Totes les variables que tenim en el dataset fan referència a característiques dels passatgers del titanic. Tot i això, podem precindir de les columnes *PassengerId*, *Name*, *Ticket* i *Cabin* ja que no aporten informació rellevant de cara a la pregunta que respon aquest conjunt de dades.

```
# Eliminació de les columes 'PassengerId', 'Name', 'Ticket' i 'Cabin'
titanic_data <- select(titanic_data, -c(PassengerId, Name, Ticket, Cabin))
summary(titanic_data)
```

```
##  Survived     Pclass          Sex           Age            SibSp
##  0:815    Min.   :1.000   female:466   Min.   : 0.17   Min.   :0.0000
##  1:494    1st Qu.:2.000   male  :843   1st Qu.:21.00   1st Qu.:0.0000
##           Median :3.000                Median :28.00   Median :0.0000
```

```
##           Mean   :2.295                Mean   :29.88   Mean   :0.4989
##           3rd Qu.:3.000                3rd Qu.:39.00   3rd Qu.:1.0000
##           Max.   :3.000                Max.   :80.00   Max.   :8.0000
##                                        NA's   :263
##      Parch            Fare          Embarked
##  Min.   :0.000   Min.   :  0.000    :  2
##  1st Qu.:0.000   1st Qu.:  7.896   C:270
##  Median :0.000   Median : 14.454   Q:123
##  Mean   :0.385   Mean   : 33.295   S:914
##  3rd Qu.:0.000   3rd Qu.: 31.275
##  Max.   :9.000   Max.   :512.329
##                  NA's   :1
```

2.2 Dades amb elements buits (valors perduts)

Aquest conjunt de dades conté dades amb elements buits representats de dues maneres diferents: amb el valor NA (*Not Available*) i amb un espai en blanc, pel que es procedeix a comprovar quins camps contenen elements buits i en quina quantitat.

```
# Número de valors perduts per camp
colSums(is.na(titanic_data))
```

```
## Survived   Pclass      Sex      Age    SibSp    Parch     Fare Embarked
##        0        0        0      263        0        0        1        0
```

```
colSums(titanic_data == "")
```

```
## Survived   Pclass      Sex      Age    SibSp    Parch     Fare Embarked
##        0        0        0       NA        0        0       NA        2
```

Com es pot observar tenim 2 valors en blanc a la variable *Embarked*, 1 valor NA a *Fare* i 263 valors NA a *Age*.

Primer de tot tractarem els valors en blanc de la variable *Embarked*. Ens basarem en utilitzar una mesura de tendència central i, al tractar-se d'una variable categòrica, utilitzarem la **moda**.

```
# Consulta de la moda de la variable 'Embarked'
mlv(titanic_data$Embarked, method = "mfv")
```

```
## [1] S
## Levels:  C Q S
```

Com es pot observar, al ser *S* la moda prenem aquest valor per omplir als valors buits de la variable.

```
# Imputació dels valors buits de la variable 'Embarked'
titanic_data$Embarked[titanic_data$Embarked == ""] = "S"
```

Per tractar el valor perdut a la variable *Fare* s'utilitzarà la **mitjana**.

```
# Imputació dels valors buits de la variable 'Fare'
titanic_data[is.na(titanic_data$Fare),]$Fare <- mean(titanic_data$Fare, na.rm = TRUE)
```

Per acabar, per tractar els valors perduts de la variable *Age* s'utilitzarà la **mitjana**. Per dur a terme la obtenció d'aquesta mitjana, enlloc d'obtenir la mitjana de l'atribut *Age* sencer, tindrem en compte el gènere (*Sex*) i la classe en la que viatjava (*Pclass*). A la gràfica següent es pot observar la relació entre els atributs *Age* i *Pclass* per dones i per homes.

```
# Visualitzem la relació entre les variables 'Age' i 'Pclass'
par(mfrow = c(1,2))
female_people = titanic_data[titanic_data$Sex == "male",]
male_people = titanic_data[titanic_data$Sex == "female",]
boxplot(female_people$Age~female_people$Pclass, main = "Pclass by age (female)",
        xlab = "Pclass", ylab = "Age")
boxplot(male_people$Age~male_people$Pclass, main = "Pclass by age (male)",
        xlab = "Pclass", ylab = "Age")
```

Per tractar els valors perduts tindrem en compte la informació observada a la gràfica anterior. Per realitzar aquesta tasca s'ha creat una funció `AgeMean` per obtenir la mitjana d'edats de les dones i dels homes segons la classe, i després s'ha creat una altra funció assignar als passatgers que tenen l'edat en blanc la mitjana corresponent al seu gènere i a la classe en la que viatjava.

```
# Funció per obtenir el camp 'Mean' del resultat de la funció 'summary'
AgeMean <- function(age) {
  round(summary(age)['Mean'])
```

```
}

female_mean_ages = tapply(female_people$Age, female_people$Pclass, AgeMean)
male_mean_ages = tapply(male_people$Age, male_people$Pclass, AgeMean)

# Funció per obtenir un valor de mitjana d'edat segons els camps 'Sex' i 'Pclass'
AgeImpute <- function(row) {
  sex <- row['Sex']
  age <- row['Age']
  pclass <- row['Pclass']
  value <- age
  if (is.na(age)) {
    if (sex == "female") {
      value <- female_mean_ages[pclass]
    } else {
      value <- male_mean_ages[pclass]
    }
  }
  return(as.numeric(value))
}

titanic_data$Age <- apply(titanic_data[, c("Sex", "Age", "Pclass")], 1, AgeImpute)
```

2.3 Identificació de valors extrems

3 Anàlisi de les dades

3.1 Comprovació de la normalitat i homogeneïtat de la variància

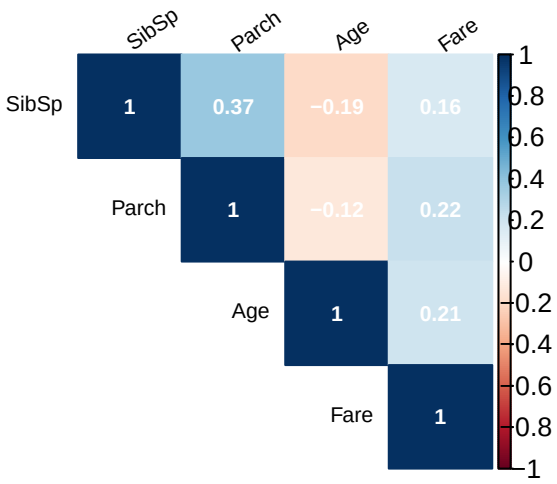
3.2 Aplicació de proves estadístiques

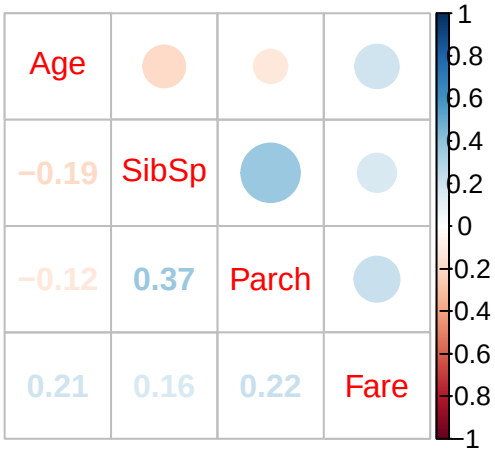
3.2.1 Contrast d'hipòtesis

3.2.2 Matriu de correlació

A continuació podem plasmar la idea anterior gràficament, calculant prèviament la matriu de correlació i guardant-la en un objecte.

```
# Creació de la matriu de correlació
corr_data <- titanic_data[, c("Age", "SibSp", "Parch", "Fare")]
M <- cor(corr_data)
par(mfrow = c(1,2))
corrplot(M, method = "color", type = "upper",
         addCoef.col = "white", number.cex = 0.7,
         tl.col="black", tl.srt=35, tl.cex = 0.7,
         order = "hclust")
corrplot.mixed(M)
```





Però no podem dir si són significativament diferent de 0, és a dir, no tenim evidencies estadístiques. Per saber-ho cal dur a terme una prova de significació. Amb la següent instrucció podem veure la matriu anterior i els p-value, on en la majoria dels casos hi ha correlació, els p-value són especialment petits.

```
rcorr(as.matrix(corr_data))
```

```
##          Age SibSp Parch Fare
## Age    1.00 -0.19 -0.12 0.21
## SibSp -0.19  1.00  0.37 0.16
## Parch -0.12  0.37  1.00 0.22
## Fare   0.21  0.16  0.22 1.00
##
## n= 1309
##
##
## P
##        Age SibSp Parch Fare
## Age         0     0     0
## SibSp  0          0     0
## Parch  0    0           0
## Fare   0    0     0
```

En tots els casos el p-value es (=0) molt petit, és a dir, que és estadísticament significatiu. Destaquen les relacions entre el preu del bitllet i l'edat, i entre el preu del bitllet i la mida de la família. En aquestes relacions la correlació és positiva, de manera que a major edat major ha sigut el preu del bitllet, i el mateix passa amb la mida de la família i el preu del bitllet.

3.2.3 Regressió logística

Volem predir el fet de sobreviure o no, de manera que ens trobem amb una variable discreta, concretament binaria (0,1). Si utilitzéssim un model lineal per predir un grup binàri estariem obtenint un model erroni.

```
# Divisió del conjunt de dades en dos subconjunts, un de train i l'altre de test
train <- titanic_data[1:667,]
test <- titanic_data[668:889,]

# Creació del model de predicció
model <- glm(Survived ~., family = binomial(link = 'logit'), data = train)
summary(model)
```

```
##
## Call:
## glm(formula = Survived ~ ., family = binomial(link = "logit"),
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4606  -0.6121  -0.4273   0.6538   2.4182
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.381022   0.698064   7.708 1.27e-14 ***
## Pclass      -1.194663   0.178218  -6.703 2.04e-11 ***
## Sexmale     -2.719909   0.227150 -11.974  < 2e-16 ***
## Age         -0.034706   0.009187  -3.778 0.000158 ***
## SibSp       -0.259178   0.123884  -2.092 0.036429 *
## Parch       -0.094120   0.142624  -0.660 0.509308
## Fare        -0.002146   0.003049  -0.704 0.481501
## EmbarkedQ    0.072021   0.424418   0.170 0.865251
## EmbarkedS   -0.366620   0.268140  -1.367 0.171541
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 891.99  on 666  degrees of freedom
## Residual deviance: 605.49  on 658  degrees of freedom
## AIC: 623.49
##
## Number of Fisher Scoring iterations: 5
```

```
# Predicció de les dades
result <- predict(model, newdata = test, type = 'response')
result <- ifelse(result > 0.5,1,0)
fitted.probabilities <- predict(model, newdata = test, type = 'response')
fitted.results <- ifelse(fitted.probabilities > 0.5, 1, 0)

# Matriu de confusió
confusionMatrix(table(fitted.results, test$Survived))
```

```
## Confusion Matrix and Statistics
```

```
##
##
## fitted.results   0   1
##              0 126  25
##              1  15  56
##
##                Accuracy : 0.8198
##                  95% CI : (0.7628, 0.868)
##     No Information Rate : 0.6351
##     P-Value [Acc > NIR] : 1.341e-09
##
##                   Kappa : 0.6008
##
##  Mcnemar's Test P-Value : 0.1547
##
##             Sensitivity : 0.8936
##             Specificity : 0.6914
##          Pos Pred Value : 0.8344
##          Neg Pred Value : 0.7887
##              Prevalence : 0.6351
##          Detection Rate : 0.5676
##    Detection Prevalence : 0.6802
##       Balanced Accuracy : 0.7925
##
##        'Positive' Class : 0
##
```

Mitjançant els resultats del model podem veure com el fet de pertanyer a la clase 2 o 3 està relacionat amb el fet de sobreviure, com també el genère, on el fet de ser home té un efecte negatiu.

L'edat també té un efecte negatiu en la supervivencia: a major edat menor probabilitat de sobreviure.

Mitjançant l'intercept, podem veure el que hem anat confirmant amb els test d'independència i els gràfics, i és que el fet de ser dona i viatjar a la classe 1 té una major probabilitat de supervivència.

A través de la matriu de confusió es pot veure com el model té un 82% de precisió en la predicció.

4 Representació dels resultats

5 Resolució del problema

6 Recursos

1. Subirats Maté, L., Pérez Trenanz, D. O., & Calvo González, M. (2019). Introducció a la neteja i anàlisi de dades. UOC.
2. Amat Rodrigo, J. (2016). RPubs - Análisis de Normalidad: gráficos y contrastes de hipótesis. https://rpubs.com/Joaquin_AR/218465
3. Homogeneity of variance. http://www.cookbook-r.com/Statistical_analysis/Homogeneity_of_variance/
4. aggregate function | R Documentation. https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/aggregate
5. Shirokanova, A., & Volchenko, O. (2019). RPubs - Chi squared. https://rpubs.com/ovolchenko/chisq2

6. fisher.test function | R Documentation. https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/fisher.test
7. Amat Rodrigo, J. (2016). RPubs - Test exacto de Fisher, chi-cuadrado de Pearson, McNemar y Q-Cochran. https://rpubs.com/Joaquin_AR/220579
8. Ramos Lorenzo, C. (2019). RPubs - Logistic Regression. https://rpubs.com/MrCristianrl/500969