Tipología y ciclo de vida de los datos. Práctica 2

Baltasar Boix / Yago Ezcurra

13/5/2021

Contents

1 Titanic - Machine Learning from Disaster. Kaggle competition.

```
require(tidyverse)
require(lares)
require(GGally)
require(knitr)
require(kableExtra)
require(gridExtra)
require(DescTools)
require(caret)
```

2 Descripción del dataset. ¿Por qué es importante y qué pregunta/problema pretende responder?

2.1 Lectura y analisis previo del dataset.

Data Dictionary

Variable	Definition	Key
survival	Survival	0 = No, 1 = Yes
pclass	Ticket class	1 = 1st, 2 = 2nd, 3 = 3rd
sex	Sex	
Age	Age in years	
sibsp	# of siblings / spouses aboard the Titanic	
parch	# of parents / children aboard the Titanic	
ticket	Ticket number	
fare	Passenger fare	
cabin	Cabin number	
embarked	Port of Embarkation	C = Cherbourg, Q = Queenstown, S = Southampton

Variable Notes

pclass: A proxy for socio-economic status (SES)
1st = Upper
2nd = Middle
3rd = Lower

age: Age is fractional if less than 1. If the age is estimated, is it in the form of xx.5

sibsp: The dataset defines family relations in this way...
Sibling = brother, sister, stepbrother, stepsister
Spouse = husband, wife (mistresses and fiancés were ignored)

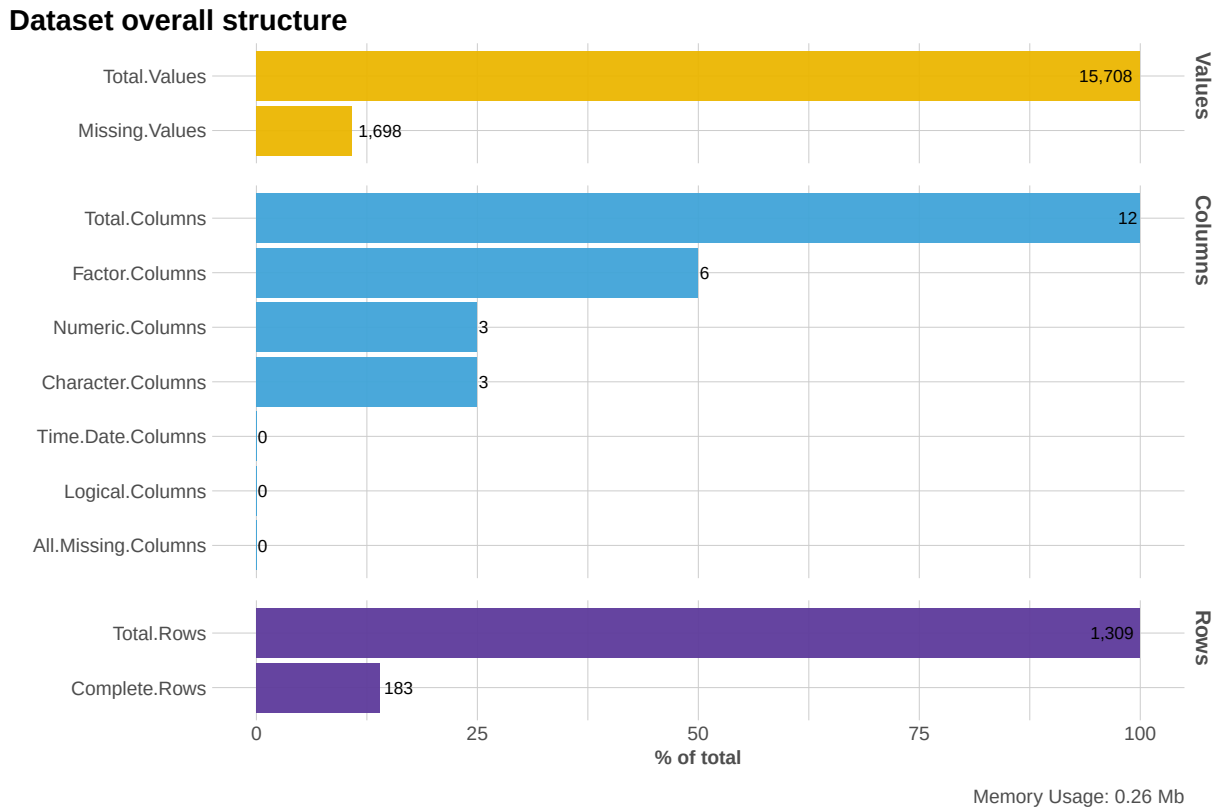
parch: The dataset defines family relations in this way...
Parent = mother, father
Child = daughter, son, stepdaughter, stepson
Some children travelled only with a nanny, therefore parch=0 for them.

Figure 1: Descripción del dataset obtenido en Kaggle

```r
df <- read_csv('../data/train.csv')
df_t <- read_csv('../data/test.csv')

df_t$Survived <- NA
df <- bind_rows(df, df_t)

df$Survived <- factor(df$Survived)
df$Pclass <- factor(df$Pclass)
df$Sex <- factor(df$Sex)
df$SibSp <- factor(df$SibSp)
df$Parch <- factor(df$Parch)
df$Embarked <- factor(df$Embarked)

summary(df)
```

```
##   PassengerId   Survived   Pclass      Name              Sex
##  Min.   :   1   0   :549   1:323   Length:1309        female:466
##  1st Qu.: 328   1   :342   2:277   Class :character   male  :843
##  Median : 655   NA's:418   3:709   Mode  :character
##  Mean   : 655
##  3rd Qu.: 982
##  Max.   :1309
##
##       Age         SibSp       Parch         Ticket               Fare
##  Min.   : 0.17   0:891   0      :1002   Length:1309        Min.   :  0.000
##  1st Qu.:21.00   1:319   1      : 170   Class :character   1st Qu.:  7.896
##  Median :28.00   2: 42   2      : 113   Mode  :character   Median : 14.454
##  Mean   :29.88   3: 20   3      :   8                      Mean   : 33.295
##  3rd Qu.:39.00   4: 22   4      :   6                      3rd Qu.: 31.275
##  Max.   :80.00   5:  6   5      :   6                      Max.   :512.329
##  NA's   :263     8:  9   (Other):   4                      NA's   :1
##     Cabin           Embarked
##  Length:1309        C   :270
##  Class :character   Q   :123
##  Mode  :character   S   :914
##                     NA's:  2
##
##
##
```

```r
df_str(df)
```



Creamos la variable dicotómica `Child` para diferenciar los niños de los adultos (>12años).

Creamos la variable `n_ticket` con el número de peraonas que viajan con el mismo ticket.

Separamos del `Name` el título (`title_name`) y el primer apellido (`first_name`).

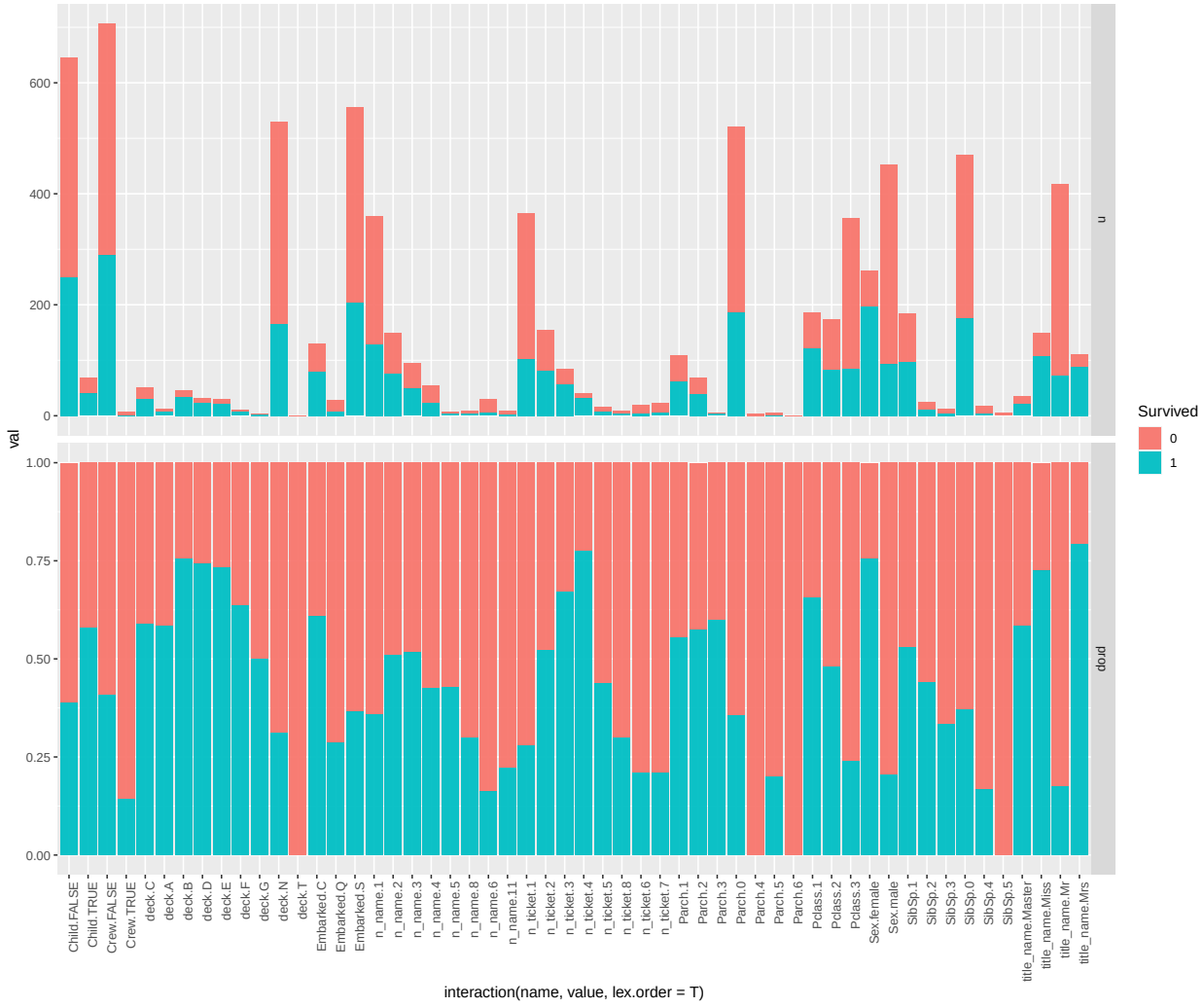
Simplificamos `title_name` en cuatro niveles.

Creamos la variable `Crew`. TRUE para la personas con Fare==0 que consideramos que viajan como tripulación.

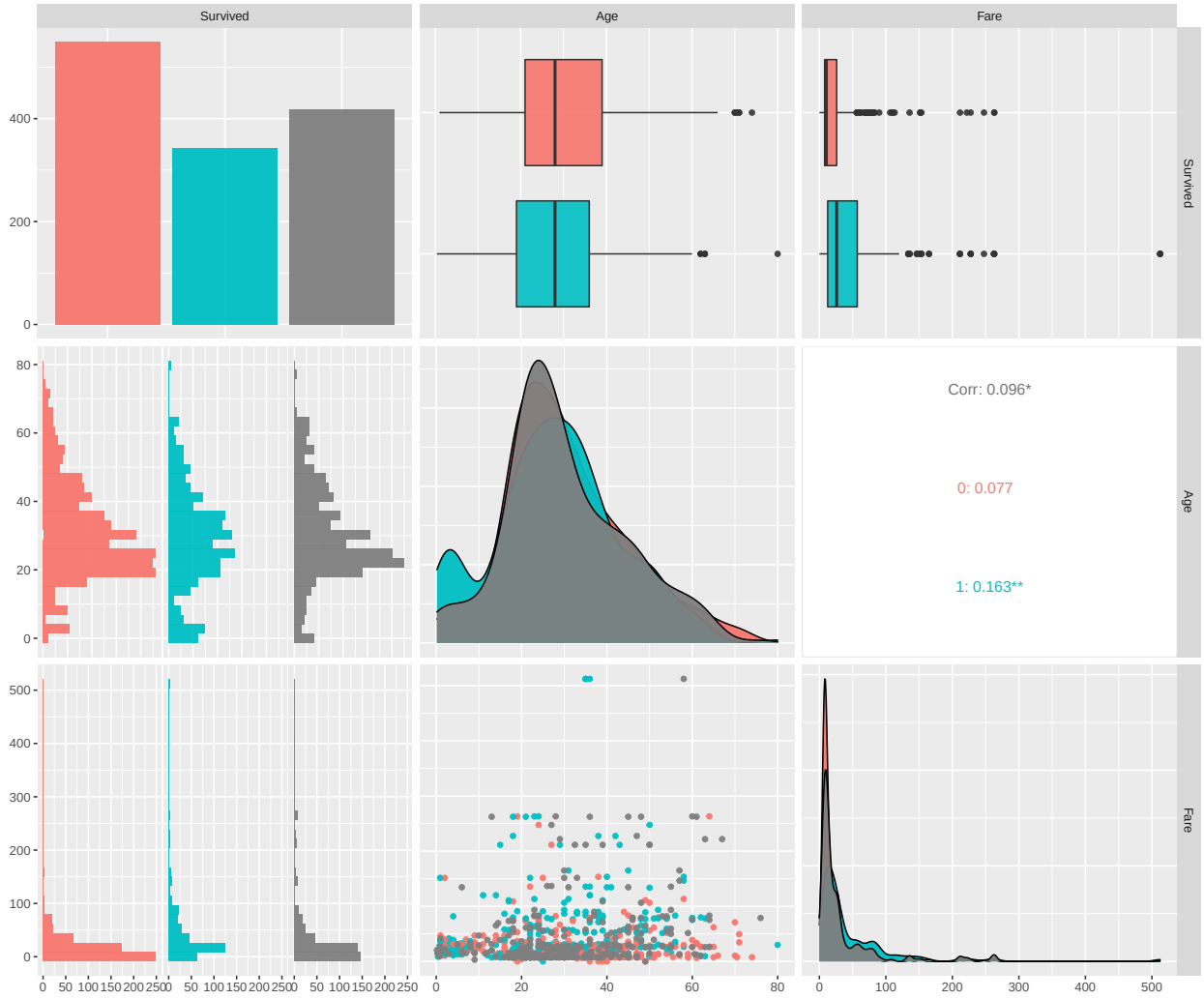
3 Integración y selección de los datos de interés a analizar.

```
df <- df %>%
  mutate(Child=factor(Age<=12))

df <- left_join(df, df %>%
                  group_by(Ticket) %>%
                  summarize(n_ticket=n())) %>%
  mutate(n_ticket=factor(n_ticket))

df <- df %>%
  separate(Name, c('first_name', 'rest_name'), sep=', ', remove=F) %>%
  separate(rest_name, c('title_name', 'rest_name'), sep='\\.') %>%
  select(-rest_name)

df$title_name[df$title_name %in% c('Capt', 'Col', 'Don', 'Dr', 'Jonkheer',
                                   'Major', 'Rev', 'Sir')] <- 'Mr'
df$title_name[df$title_name %in% c('Lady', 'Mme','the Countess', 'Dona')] <- 'Mrs'
df$title_name[df$title_name %in% c('Mlle', 'Ms')] <- 'Miss'
df$title_name <- factor(df$title_name)

df <- left_join(df, df %>%
```

```
                  group_by(Pclass, first_name) %>%
                  summarize(n_name=n())) %>%
  mutate(n_name=factor(n_name))

df <- df %>%
  mutate(deck=if_else(is.na(Cabin), 'N', str_sub(Cabin, 1, 1) )) %>%
  mutate(deck=factor(deck))

df$Fare[is.na(df$Fare)] <- df %>% filter(Pclass==3) %>% summarize(mean(Fare, na.rm=T)) %>% pull()

df$Embarked[is.na(df$Embarked)] <- 'S'

df <- df %>%
  mutate(Crew=factor(if_else(Fare==0, TRUE, FALSE)))

df %>%
  select(where(is.factor)) %>%
  na.omit() %>%
  pivot_longer(-Survived) %>%
  group_by(name, value, Survived) %>%
  summarize(n=n()) %>%
  mutate(prop=prop.table(n)) %>%
  pivot_longer(c(n,prop), names_to='tipo', values_to='val') %>%
  ggplot(aes(x=interaction(name,value, lex.order = T), y=val, fill=Survived)) +
    geom_bar(stat='identity', position='stack') +
    facet_grid(tipo ~ ., scale='free_y') +
    theme(axis.text.x = element_text(angle = 90, vjust = 1, hjust=1))
```




```
df %>%
  select(Survived, where(is.numeric), -PassengerId) %>%
  ggpairs(aes(color=Survived))
```



4 Limpieza de los datos.

4.1 ¿Los datos contienen ceros o elementos vacíos? ¿Cómo gestionarías cada uno de estos casos?

Sustituimos los NA's de la variable Child con los siguientes criterios:

Asumimos que los viajeros con ticket unipersonal con SibSp==0 (sin hermanos o esposa a bordo) y Parch==0 (sin hermanos o padres a bordo) no son niños.

Asumimos que las personas con title_name=='Mrs' (mujeres casadas) no son niños.

Asumimos que las personas con title_name=='Master' son niños.

Asumimos que las personas con SibSp>0 y Parch>0 son niños.

```
df <- df %>%
  mutate(Child = if_else(SibSp == 0 & Parch == 0 & is.na(Child) & n_ticket == 1,
                         FALSE, as.logical(Child) )) %>%
  mutate(Child = if_else(title_name == 'Master' & is.na(Child),
                         TRUE, as.logical(Child))) %>%
  mutate(Child = if_else(title_name == 'Mrs' & is.na(Child),
                         FALSE, as.logical(Child))) %>%
  mutate(Child = if_else(SibSp != '0' & Parch != '0' & is.na(Child),
```

```
                        TRUE, as.logical(Child))) %>%
  mutate(Child = if_else(is.na(Child), FALSE, as.logical(Child))) %>%
  mutate(Child=factor(Child))



prop.table(table(df$Survived, df$Child, dnn = c('Survived', 'Child')), margin = 2)
```

```
##          Child
## Survived     FALSE      TRUE
##        0 0.6290323 0.4941176
##        1 0.3709677 0.5058824
```

4.2 Identificación y tratamiento de valores extremos.

5 Análisis de los datos.

5.1 Selección de los grupos de datos que se quieren analizar/comparar (planificación de los análisis a aplicar).

5.2 Comprobación de la normalidad y homogeneidad de la varianza.

5.3 Aplicación de pruebas estadísticas para comparar los grupos de datos. En función de los datos y el objetivo del estudio, aplicar pruebas de contraste de hipótesis, correlaciones, regresiones, etc. Aplicar al menos tres métodos de análisis diferentes.

Para determinar la asociación de la variable `Survived` con el resto de variables categóricas utilizamos el test χ^2 y el coeficiente CramerV.

Para determinar la asociación de la variable `Survived` con las variables numéricas utilizaremos el t.test (H_0: la media del subset con `Survived==1` es igual a la media con `Survived==0`).

Ajustamos un modelo GLM y minimizamos el Aic con la función `step` (búsqueda de las variables más significativas).

Los dos primeros métodos identifican la asociación de cada variable individualmente. GLM tiene en cuenta la colinearidad de las variables.

Las más significativas son: `Pclass`, `title_name`, `Age` y `n_ticket`.

```
df %>%
  select(Survived, where(is.factor)) %>%
  filter(!is.na(Survived)) %>%
  pivot_longer(-Survived) %>%
  group_by(name) %>%
  summarize(cramerv=CramerV(x=Survived, y=factor(value)),
            chisq.pvalue=chisq.test(Survived, factor(value))$p.value) %>%
  mutate(`signif 95%`=chisq.pvalue < 0.05) %>%
  arrange(chisq.pvalue) %>%
  kable(format='latex', digits=4, caption='CramerV y chisq.test') %>%
  kable_styling(full_width = F, latex_options = "HOLD_position")
```

Table 1: CramerV y chisq.test

name	cramerv	chisq.pvalue	signif 95%
title_name	0.5708	0.0000	TRUE
Sex	0.5434	0.0000	TRUE
Pclass	0.3398	0.0000	TRUE
n_ticket	0.3388	0.0000	TRUE
deck	0.3336	0.0000	TRUE
SibSp	0.2045	0.0000	TRUE
Embarked	0.1707	0.0000	TRUE
n_name	0.1979	0.0000	TRUE
Parch	0.1770	0.0001	TRUE
Child	0.0815	0.0206	TRUE
Crew	0.0853	0.0226	TRUE

```
df %>%
  select(Survived, where(is.numeric), -PassengerId) %>%
  filter(!is.na(Survived)) %>%
  pivot_longer(-Survived) %>%
  group_by(name, Survived) %>%
  summarize(value_list = list(value)) %>%
  pivot_wider(names_from=Survived, values_from=value_list) %>%
  mutate(vartest.pval=var.test(unlist(`0`), unlist(`1`))$p.val) %>%
  mutate(ttest.pval=t.test(unlist(`0`), unlist(`1`),
                           var.equal=vartest.pval>0.05)$p.val) %>%
  mutate(`signif 95%`=ttest.pval < 0.05) %>%
  select(-`0`, -`1`)  %>%
  kable(format='latex', digits=4, caption='t.test') %>%
  kable_styling(full_width = F, latex_options = "HOLD_position")
```

Table 2: t.test

name	vartest.pval	ttest.pval	signif 95%
Age	0.317	0.0391	TRUE
Fare	0.000	0.0000	TRUE

```
# GLM
df0 <- df %>%
  select(-PassengerId, -Name, -first_name, -Ticket, -Cabin) %>%
  na.omit()

fit <- glm(data=df0, Survived ~ ., family=binomial(link = "logit"))

step_fit <- step(fit, direction='both', trace=0)

summary(step_fit)
```

```
##
## Call:
## glm(formula = Survived ~ Pclass + title_name + Age + SibSp +
##     Parch + Fare + n_ticket + n_name, family = binomial(link = "logit"),
##     data = df0)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3009  -0.4797  -0.2710   0.4781   2.7265
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)     5.511e+00  9.725e-01   5.666 1.46e-08 ***
```

```
## Pclass2        -1.497e+00  3.996e-01  -3.747 0.000179 ***
## Pclass3        -2.469e+00  4.302e-01  -5.738 9.60e-09 ***
## title_nameMiss -1.518e+00  8.371e-01  -1.813 0.069772 .
## title_nameMr   -4.587e+00  8.751e-01  -5.242 1.59e-07 ***
## title_nameMrs  -7.098e-01  8.948e-01  -0.793 0.427618
## Age            -3.372e-02  1.092e-02  -3.088 0.002012 **
## SibSp1          1.808e-01  3.831e-01   0.472 0.636885
## SibSp2          6.417e-01  8.061e-01   0.796 0.425990
## SibSp3          4.091e-01  1.183e+00   0.346 0.729391
## SibSp4         -1.838e+01  8.315e+02  -0.022 0.982365
## SibSp5         -4.443e+01  1.696e+03  -0.026 0.979105
## Parch1          8.129e-02  4.350e-01   0.187 0.851774
## Parch2          9.620e-01  6.991e-01   1.376 0.168783
## Parch3          1.591e+00  1.646e+00   0.967 0.333690
## Parch4         -1.336e+01  1.539e+03  -0.009 0.993071
## Parch5         -1.557e+01  8.315e+02  -0.019 0.985059
## Parch6         -4.253e+01  4.043e+03  -0.011 0.991606
## Fare            9.308e-03  5.319e-03   1.750 0.080151 .
## n_ticket2      -6.119e-01  4.101e-01  -1.492 0.135627
## n_ticket3      -1.752e-01  5.963e-01  -0.294 0.768940
## n_ticket4       3.973e-01  8.386e-01   0.474 0.635659
## n_ticket5      -2.548e+00  9.856e-01  -2.585 0.009735 **
## n_ticket6      -5.102e+00  1.468e+00  -3.476 0.000509 ***
## n_ticket7      -4.545e+00  2.077e+00  -2.188 0.028672 *
## n_ticket8       3.115e+00  1.217e+00   2.559 0.010492 *
## n_name2        -3.150e-01  4.163e-01  -0.757 0.449254
## n_name3        -1.142e+00  5.409e-01  -2.111 0.034789 *
## n_name4        -1.366e+00  6.890e-01  -1.983 0.047334 *
## n_name5        -1.467e+00  1.576e+00  -0.930 0.352162
## n_name6        -1.217e+00  1.097e+00  -1.109 0.267408
## n_name8         2.035e+01  8.315e+02   0.024 0.980476
## n_name11        1.684e+01  8.315e+02   0.020 0.983837
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 964.52  on 713  degrees of freedom
## Residual deviance: 522.04  on 681  degrees of freedom
## AIC: 588.04
##
## Number of Fisher Scoring iterations: 16
```

6 Representación de los resultados a partir de tablas y gráficas.

7 Resolución del problema. A partir de los resultados obtenidos, ¿cuáles son las conclusiones? ¿Los resultados permiten responder al problema?

```
df0 <- df %>%
  select(-PassengerId, -Name, -Cabin) %>%
  rowwise() %>%
  mutate(Age=if_else(is.na(Age), if_else(as.logical(Child), runif(1, 0, 12), runif(1, 13, 60)), Age)) %>%
  mutate(first_name=factor(first_name)) %>%
  mutate(Ticket=factor(Ticket)) %>%
  na.omit()

set.seed(200560)
train_control <- trainControl(method="cv", number=10)

# Fit the model
svm1 <- train(Survived ~ ., data = df0, method = 'svmLinear', trControl = train_control,
              tuneGrid = expand.grid(C = seq(0, 2, length = 10)))
```

```
svm1
```

```
## Support Vector Machines with Linear Kernel
##
## 891 samples
##  15 predictor
##   2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 802, 801, 802, 802, 803, 802, ...
## Resampling results across tuning parameters:
##
##   C          Accuracy   Kappa
##   0.0000000        NaN        NaN
##   0.2222222  0.8328294  0.6414743
##   0.4444444  0.8407201  0.6598885
##   0.6666667  0.8440912  0.6688588
##   0.8888889  0.8429804  0.6671013
##   1.1111111  0.8452026  0.6718010
##   1.3333333  0.8440662  0.6690223
##   1.5555556  0.8451898  0.6716414
##   1.7777778  0.8440662  0.6693570
##   2.0000000  0.8429551  0.6667851
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was C = 1.111111.
```

```
confusionMatrix(predict(svm1), df0$Survived)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 547   3
##          1   2 339
##
##                Accuracy : 0.9944
##                  95% CI : (0.987, 0.9982)
##     No Information Rate : 0.6162
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9881
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9964
##             Specificity : 0.9912
##          Pos Pred Value : 0.9945
##          Neg Pred Value : 0.9941
##              Prevalence : 0.6162
##          Detection Rate : 0.6139
##    Detection Prevalence : 0.6173
##       Balanced Accuracy : 0.9938
##
##        'Positive' Class : 0
##
```

```
df1 <- df %>%
  rowwise() %>%
  mutate(Age=if_else(is.na(Age), if_else(as.logical(Child), runif(1, 0, 12), runif(1, 13, 60)), Age)) %>%
  filter(is.na(Survived))

df1$Survived <- predict(svm1, df1)
```

```
write_csv((df1 %>% select(PassengerId, Survived)), file='My_submission.csv')
```