Titanic - Data Wrangling

Terence Fernandez

August 16, 2017

Loading of libraries and Titanic data set.

library("tidyr")  
library("dplyr")

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

train <- read.csv("titanic\_original.csv", header = TRUE)

1: Port of Embarkation

The embarked column has some missing values, which are known to correspond to passengers who actually embarked at Southampton. Find the missing values and replace them with S. (Caution: Sometimes a missing value might be read into R as a blank or empty string.)

Sumarizing the "embarked column" we find 3 records with blank values:

table(train$embarked)

##   
## C Q S   
## 3 270 123 914

View the records of the "embarked" column with blank values:

filter(train,train$embarked=="")

## ï..pclass survived name sex age  
## 1 1 1 Icard, Miss. Amelie female 38  
## 2 1 1 Stone, Mrs. George Nelson (Martha Evelyn) female 62  
## 3 NA NA NA  
## sibsp parch ticket fare cabin embarked boat body home.dest  
## 1 0 0 113572 80 B28 6 NA   
## 2 0 0 113572 80 B28 6 NA Cincinatti, OH  
## 3 NA NA NA NA

Checking if there are NA values in the "embarked" column. Results show no NA values in the embark column.

filter(train,is.na(train$embarked))

## [1] ï..pclass survived name sex age sibsp parch   
## [8] ticket fare cabin embarked boat body home.dest  
## <0 rows> (or 0-length row.names)

Update records with blank values in "embarked" column to value "S" (column 11)

train[(train[,11] ==""),11] <- "S"

Another method to update blank values with "S"

train$embarked[train$embarked ==""] <- "S"

Verify Update. Results show two more records with "S"

table(train$embarked)

##   
## C Q S   
## 0 270 123 917

2: Age

You’ll notice that a lot of the values in the Age column are missing. While there are many ways to fill these missing values, using the mean or median of the rest of the values is quite common in such cases.

* Calculate the mean of the Age column and use that value to populate the missing values
* Think about other ways you could have populated the missing values in the age column. Why would you pick any of those over the mean (or not)?

Number of records with missing values in Age column: 264

filter(train,is.na(train$age)) %>% count()

## # A tibble: 1 x 1  
## n  
## <int>  
## 1 264

Obtain 'Age' column number: 5

colnames(train)

## [1] "ï..pclass" "survived" "name" "sex" "age"   
## [6] "sibsp" "parch" "ticket" "fare" "cabin"   
## [11] "embarked" "boat" "body" "home.dest"

Update NA values with mean value in the 'Age' column (number 5).

train[is.na(train$age),5] <- mean(train$age, na.rm=TRUE)

Another to run the previous operation.

train$age[is.na(train$age)] <- mean(train$age, na.rm=TRUE)

3: Lifeboat

You’re interested in looking at the distribution of passengers in different lifeboats, but as we know, many passengers did not make it to a boat :-( This means that there are a lot of missing values in the boat column. Fill these empty slots with a dummy value e.g. the string 'None' or 'NA'.

Number of records with missing values in the 'boat' column: 824

filter(train, train$boat =="") %>% count()

## # A tibble: 1 x 1  
## n  
## <int>  
## 1 824

Number of records with NA value in the 'boat' column: 0

filter(train, is.na(train$boat)) %>% count()

## # A tibble: 1 x 1  
## n  
## <int>  
## 1 0

Update all blank 'boat' fields to NA. Although a warning is generated, the update is still excuted. The warning is a reference to an invalid factor.

train$boat[train$boat==""] <- ("NA")

## Warning in `[<-.factor`(`\*tmp\*`, train$boat == "", value =  
## structure(c(13L, : invalid factor level, NA generated

The blank fields are replaced with NA. Let's check it. The replaced NA is recognaized by the "is.na" command showing 824 updated fields as expected.

filter(train, is.na(train$boat)) %>% count()

## # A tibble: 1 x 1  
## n  
## <int>  
## 1 824

4. Cabin

You notice that many passengers don’t have a cabin number associated with them.

* Does it make sense to fill missing cabin numbers with a value?
* What does a missing value here mean?

Total data.frame dimension:

dim(train)

## [1] 1310 14

Number of records missing cabin numbers:

filter(train, train$cabin=="") %>% count()

## # A tibble: 1 x 1  
## n  
## <int>  
## 1 1015

1015 records of missing cabin numbers from a total of 1310 entries. This indicate 77% of entries are missing values, therefore it does not make sense to fill up those missing values.

You have a hunch that the fact that the cabin number is missing might be a useful indicator of survival. Create a new column has\_cabin\_number which has 1 if there is a cabin number, and 0 otherwise.

As previously calculated the number of cabins with missing values is 1015.

Update the 'cabin' missing values. Ignore the warning

train$cabin[train$cabin==""] <- "NA"

## Warning in `[<-.factor`(`\*tmp\*`, train$cabin == "", value =  
## structure(c(45L, : invalid factor level, NA generated

Verifying the update to be 1015

filter(train, is.na(train$cabin)) %>% count()

## # A tibble: 1 x 1  
## n  
## <int>  
## 1 1015

Create a column called "has\_cabin\_number" will the value "1" as default.

train$has\_cabin\_number <- "1"

Update 'has\_column\_number' to "0" to indicate the records is missing "cabin" number.

(train$has\_cabin\_number[is.na(train$cabin)] <- "0")

## [1] "0"

Verify the update.

table(train$has\_cabin\_number)

##   
## 0 1   
## 1015 295

Create a CSV file with the cleaned up data set.

write.csv(train, "titanic\_clean.csv", row.names = T)