Práctica 2: Booking Data Cleaning

Gerard Alcalde and Guillem Rochina

2022-12-24

## DESCRIPCIÓN DEL DATASET

booking <- read.csv("hotels\_data.csv")  
summary(booking)

## name city check.in adults   
## Length:2632 Length:2632 Length:2632 Min. :2.00   
## Class :character Class :character Class :character 1st Qu.:2.00   
## Mode :character Mode :character Mode :character Median :4.00   
## Mean :3.32   
## 3rd Qu.:5.00   
## Max. :5.00   
##   
## children check.out num\_rooms address   
## Min. :1.000 Length:2632 Min. :2.000 Length:2632   
## 1st Qu.:1.000 Class :character 1st Qu.:2.000 Class :character   
## Median :3.000 Mode :character Median :2.000 Mode :character   
## Mean :2.484 Mean :2.268   
## 3rd Qu.:3.000 3rd Qu.:3.000   
## Max. :3.000 Max. :3.000   
##   
## hotel\_coordinates hotel\_score hotel\_scores hotel\_description   
## Length:2632 Min. :-1.00 Length:2632 Length:2632   
## Class :character 1st Qu.: 8.00 Class :character Class :character   
## Mode :character Median : 8.40 Mode :character Mode :character   
## Mean : 8.25   
## 3rd Qu.: 8.80   
## Max. :10.00   
## NA's :5   
## features room\_data page\_count current\_page   
## Length:2632 Length:2632 Min. : 0.00 Min. :1.000   
## Class :character Class :character 1st Qu.: 25.00 1st Qu.:2.000   
## Mode :character Mode :character Median : 54.00 Median :3.000   
## Mean : 57.53 Mean :2.758   
## 3rd Qu.: 88.00 3rd Qu.:4.000   
## Max. :133.00 Max. :5.000   
##   
## in\_page\_count search\_date   
## Min. : 0.00 Length:2632   
## 1st Qu.: 6.00 Class :character   
## Median :13.00 Mode :character   
## Mean :12.55   
## 3rd Qu.:19.00   
## Max. :27.00   
##

## INTEGRACIÓN Y SELECCIÓN

We could add flights data for a number of reasons:

* An higher number of flights in a month might increase the searches of hotels near the airport.
* A higher number of flights in a month indicates more international tourism (rather than just national tourism). A kind of tourism that could be looking for hotels with different characteristics than national tourists.
* A higher number of flights in a month indicates more general tourism, which may influence the promotion strategies and offers of certain hotels, influencing in the final analysis their position in the Booking search engine.

flights <- read.csv("avia\_tf\_apal\_linear.csv.gz")  
summary(flights)

## DATAFLOW LAST.UPDATE freq unit   
## Length:1502955 Length:1502955 Length:1502955 Length:1502955   
## Class :character Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character Mode :character   
##   
##   
##   
## tra\_meas airline rep\_airp TIME\_PERIOD   
## Length:1502955 Length:1502955 Length:1502955 Length:1502955   
## Class :character Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character Mode :character   
##   
##   
##   
## OBS\_VALUE OBS\_FLAG   
## Min. : 0 Mode:logical   
## 1st Qu.: 0 NA's:1502955   
## Median : 264   
## Mean : 75333   
## 3rd Qu.: 3166   
## Max. :289204357

proc\_flights <- flights %>%   
 # Código OACI/ICAO: Barcelona --> ES\_LEBL; Valencia --> ES\_LEVC; Madrid: ES\_LEMD  
 filter(rep\_airp %in% c("ES\_LEBL", "ES\_LEVC", "ES\_LEMD")) %>%   
 filter(tra\_meas == "PAS\_CRD") %>%  
 filter(freq == "M") %>%  
 separate(TIME\_PERIOD, c("YEAR", "MONTH"), sep = "-") %>%  
 filter(YEAR %in% c("2022", "2021", "2020", "2019", "2018") & MONTH %in% c("03", "06", "12")) %>%  
 filter(airline == "TOTAL") %>%  
 mutate(month\_name = ifelse(MONTH == "03", "March", ifelse(MONTH == "06", "June", "December")),  
 city\_airp = ifelse(rep\_airp == "ES\_LEBL", "Barcelona", ifelse(rep\_airp == "ES\_LEVC", "Valencia", "Madrid"))) %>%  
 group\_by(rep\_airp, month\_name) %>%  
 mutate(mean\_fligths = mean(OBS\_VALUE)) %>%  
 ungroup() %>%  
 select(c(city\_airp, month\_name, mean\_fligths))  
  
final\_flights <- unique(proc\_flights)

booking <- booking %>%  
 separate(check.in, c(NA, "month\_in", NA), sep = "-", remove = FALSE) %>%  
 merge(final\_flights, by.x = c("city", "month\_in"), by.y = c("city\_airp", "month\_name"))

## LIMPIEZA DE LOS DATOS

booking <- booking %>%  
 separate(hotel\_coordinates, c("latitude", "longitude"), sep = ",")

columns <- c("staff\_score", "facilities\_score", "cleanliness\_score", "comfort\_score", "value\_for\_money\_score", "location\_score", "free\_wifi\_score")  
  
booking <- booking %>%  
 separate(hotel\_scores, c(NA, "staff\_score", "facilities\_score", "cleanliness\_score", "comfort\_score", "value\_for\_money\_score", "location\_score", "free\_wifi\_score"), sep = ",")

## Warning: Expected 8 pieces. Missing pieces filled with `NA` in 286 rows [9, 80,  
## 105, 125, 141, 202, 203, 218, 223, 224, 225, 226, 227, 247, 303, 311, 322, 327,  
## 336, 370, ...].

booking[columns] <- apply(booking[columns], 2, readr::parse\_number)

booking <- booking %>%  
 separate(check.in, c(NA, "month\_in", NA), sep = "-", remove = FALSE) %>%  
 separate(check.out, c(NA, "month\_out", NA), sep = "-", remove = FALSE) %>%  
 separate(check.in, c("day", "month", "year"), sep = "-") %>%  
 mutate(month = ifelse(month == "March", "03", ifelse(month == "June", "06", "12"))) %>%  
 unite("check\_in", c(day, month, year), sep = "-") %>%  
 separate(check.out, c("day", "month", "year"), sep = "-") %>%  
 mutate(month = ifelse(month == "March", "03", ifelse(month == "June", "06", "12"))) %>%  
 unite("check\_out", c(day, month, year), sep = "-")  
  
booking["check\_in"] <- as.Date.character(booking$check\_in, "%d-%m-%Y")  
booking["check\_out"] <- as.Date(booking$check\_out, "%d-%m-%Y")

Sys.setlocale('LC\_ALL', 'C')

## [1] "C"

booking <- booking %>%  
 extract(address, c("postal\_code"), regex = "( [0-9]\* )")

feature\_list <- c("('Free Wifi')", "('Air conditioning')", "('24-hour front desk')", "('Safe')", "('Heating')", "('Elevator')", "('Private Bathroom')", "('Non-smoking rooms')", "('Aparments')", "('City view')","('Kitchen')", "('Pet Friendly')", "('Swimming pool')", "('Balcony')")  
  
for (feature in feature\_list){  
 col\_name <- str\_replace(str\_replace(str\_to\_lower(str\_extract(feature, "([A-Z][a-z]\*( |-)?[A-Z]?[a-z]\* ? ?[a-z]\*)")), " ", "\_"), "-", "\_")  
 booking <- booking %>%  
 extract(features, c(col\_name), regex = feature, remove = FALSE) %>%  
 mutate\_(.dots = setNames(list(paste0("as.integer(!is.na(",col\_name,"))")), col\_name))  
}

## Warning: `mutate\_()` was deprecated in dplyr 0.7.0.  
## i Please use `mutate()` instead.  
## i See vignette('programming') for more help  
## i The deprecated feature was likely used in the dplyr package.  
## Please report the issue at <]8;;https://github.com/tidyverse/dplyr/issueshttps://github.com/tidyverse/dplyr/issues]8;;>.

## ANÁLISIS DE LOS DATOS

## RESOLUCIÓN DEL PROBLEMA