

Práctica 2: Limpieza y validación de los datos

Autor: Alberto Giménez Aragón

Enero 2021

Contents

1 Descripción del dataset	1
2 Integración y selección de los datos de interés a analizar	3
3 Limpieza de los datos	4
3.1 Preprocesado	4
3.2 Elementos nulos	7
3.3 Valores extremos	9
4 Análisis de los datos	13
4.1 Comparación de la limpieza entre hoteles baratos y de lujo	13
4.2 Comparación de precios entre hoteles de distintos distritos	15
5 Representación de los resultados a partir de tablas y gráficas	17
6 Conclusiones	17

1 Descripción del dataset

Para la realización de esta práctica he escogido el dataset que generamos en la práctica 1. Este dataset recoge una gran variedad de datos sobre los distintos hoteles repartidos a lo largo de la ciudad de Barcelona y alrededores, los cuales fueron obtenidos mediante *web scraping* a fecha de 2 de noviembre de 2020.

A continuación, se detallan todos los atributos de los que dispone el dataset.

- **Name:** Nombre del Hotel
- **Stars:** Estrellas del hotel, entre 1 y 5 estrellas
- **Score:** Calificación del hotel según los usuarios.
- **Score Location:** Calificación de la ubicación del hotel según los usuarios (de 1.0 a 5.0).
- **Score Cleaning:** Calificación de la limpieza del hotel según los usuarios (de 1.0 a 5.0).
- **Score Service:** Calificación del servicio del hotel según los usuarios (de 1.0 a 5.0).
- **Score Value for Money:** Calificación de la relación calidad/precio del hotel según los usuarios (de 1.0 a 5.0).
- **Price:** Precio actual por noche del hotel
- **Price Range:** Rango de precios por noche en el que se encuentra el hotel
- **Ranking:** Ranking Tripadvisor del hotel respecto a otros hoteles de la ciudad
- **Number opinions:** Número de opiniones dejadas por los huéspedes a través de Tripadvisor
- **Number opinions excellent:** Número de opiniones excelentes dejadas por los huéspedes a través de Tripadvisor

- **Number opinions good:** Número de opiniones buenas dejadas por los huéspedes a través de Tripadvisor
- **Number opinions normal:** Número de opiniones normales dejadas por los huéspedes a través de Tripadvisor
- **Number opinions bad:** Número de opiniones malas dejadas por los huéspedes a través de Tripadvisor
- **Number opinions awful:** Número de opiniones pésimas dejadas por los huéspedes a través de Tripadvisor
- **Number QA:** Número de preguntas y respuestas de los usuarios
- **Nearby restaurants:** Número de restaurantes cercanos al hotel
- **Nearby attractions:** Número de atracciones turísticas cercanas al hotel
- **Zone:** Zona en la que se sitúa el hotel
- **Latitude/Longitude:** Latitud y longitud del hotel
- **Swimming pool:** Nos dice si el hotel dispone de este servicio
- **Bar:** Nos dice si el hotel dispone de este servicio
- **Restaurant:** Nos dice si el hotel dispone de este servicio
- **Breakfast:** Nos dice si el hotel dispone de este servicio
- **Gym:** Nos dice si el hotel dispone de este servicio
- **Reception 24h:** Nos dice si el hotel dispone de este servicio
- **Admit pets:** Nos dice si el hotel dispone de este servicio
- **Air conditioning:** Nos dice si el hotel dispone de este servicio
- **Strong box:** Nos dice si el hotel dispone de este servicio
- **Rooms:** Número de habitaciones del hotel
- **Suites:** Nos dice si el hotel dispone de este tipo de habitaciones
- **Sea View Rooms:** Nos dice si el hotel dispone de este tipo de habitaciones
- **Non-smoking Rooms:** Nos dice si el hotel dispone de este tipo de habitaciones
- **Landmark View Rooms:** Nos dice si el hotel dispone de este tipo de habitaciones
- **City View Rooms:** Nos dice si el hotel dispone de este tipo de habitaciones
- **Family Rooms:** Nos dice si el hotel dispone de este tipo de habitaciones
- **Style:** Estilo del hotel
- **Tripadvisor Classification:** Calificación de Tripadvisor sobre la facilidad de realizar actividades y encontrar restaurantes a corta distancia del hotel.
- **Language Spanish:** Indica si el hotel habla este idioma o no
- **Language Catalan:** Indica si el hotel habla este idioma o no
- **Language French:** Indica si el hotel habla este idioma o no
- **Language English:** Indica si el hotel habla este idioma o no
- **Language Italian:** Indica si el hotel habla este idioma o no
- **Language Bulgarian:** Indica si el hotel habla este idioma o no
- **Language Russian:** Indica si el hotel habla este idioma o no
- **Language Portuguese:** Indica si el hotel habla este idioma o no
- **Prat Distance:** Distancia en kilómetros al aeropuerto del Prat
- **Timestamp:** Fecha/hora de recogida de los datos

Con la elección de este dataset, se pretende dar respuesta a diferentes preguntas sobre los hoteles, con el objetivo de ver si algunos atributos varían o no respecto a los barrios donde se sitúan los hoteles o entre diferentes rangos de precio. De esta forma, un posible turista puede ayudarse por este estudio para elegir entre los hoteles que más le convengan. Algunas de estas preguntas podrían ser las siguientes.

- ¿Hay relación entre las estrellas de un hotel y la valoración de los usuarios?
- ¿Hay diferencias en los precios de hoteles entre diferentes barrios de Barcelona?
- ¿Es cierto que los hoteles más caros ofrecen una mejor limpieza de sus instalaciones? ¿Y un mejor servicio?
- ¿Son más caros aquellos hoteles que admiten mascotas de los que no lo hacen? ¿Y los que tienen gimnasio?
- ¿Se hablan más idiomas en los hoteles de un barrio respecto a los demás? ¿Y en los más caros respecto a los demás?

- ¿La relación calidad precio es mayor en hoteles más baratos que en los caros?
- ¿Hay relación entre la puntuación que los usuarios atribuyen a la ubicación del hotel respecto al número de restaurantes y atracciones cercanas?
- ¿Los hoteles con menos estrellas suelen tener más opiniones de usuarios que los de más estrellas?
- ¿Hay más habitaciones disponibles en algún barrio?

A parte, se intentará generar algún modelo de clasificación para predecir los rangos de precios de los hoteles según el resto de características.

2 Integración y selección de los datos de interés a analizar

El primer paso que debemos hacer es cargar el dataset que se va a utilizar. Lo leemos y utilizamos la función `str` para ver su estructura y una muestra de los datos.

```
library(stringr)
library(ggplot2)
library(gridExtra)

df <- read.csv("tripadvisor_barcelona_hotels.csv", encoding = "UTF-8")
str(df, strict.width="wrap")

## 'data.frame': 1619 obs. of 50 variables:
## $ name : chr "Travelodge Barcelona Fira" "Four Points by Sheraton Barcelona
## Diagonal" "Hotel Miramar Barcelona" "W Barcelona" ...
## $ stars : num 3 3 5 5 3 NA 4 3 4 4.5 ...
## $ score : num 3.5 4.5 4 4.5 4.5 4 5 4.5 4 5 ...
## $ score_location : num 4 4 4.5 4.5 4 4.5 5 4.5 4.5 5 ...
## $ score_cleaning : num 4.5 4.5 4.5 4.5 4.5 4 5 4.5 4.5 5 ...
## $ score_service : num 4 4.5 4 4.5 4.5 4 4.5 4.5 4 5 ...
## $ score_value_money : num 4 4.5 4 4 4.5 4.5 4.5 4.5 4 4.5 ...
## $ ranking : num 354 150 198 243 134 80 10 92 313 6 ...
## $ price : num NA 76 184 323 143 NA NA 98 126 160 ...
## $ price_range : chr "52 \200 - 166 \200" "79 \200 - 195 \200" "152 \200 -
## 283 \200" "287 \200 - 813 \200" ...
## $ opinions : num 234 1574 1298 7751 1404 ...
## $ opinions_excellent : num 77 685 708 4598 693 ...
## $ opinions_good : num 72 692 339 1789 566 ...
## $ opinions_normal : num 49 137 107 671 103 29 17 127 200 13 ...
## $ opinions_bad : num 19 31 66 329 31 11 4 21 56 4 ...
## $ opinions_awful : num 17 30 78 364 11 24 2 15 51 3 ...
## $ num_qa : num 15 99 28 182 71 10 23 44 34 94 ...
## $ nearby_restaurants : num 66 170 55 18 468 388 454 566 668 594 ...
## $ nearby_attractions : num 7 11 20 2 48 50 113 111 275 296 ...
## $ zone : chr "Barcelona" "Poblenou / Sant Martí" "Montjuic / Sants-Montjuïc"
## "Ciutat Vella / El Port Vell / Barceloneta" ...
## $ latitude : num 41.4 41.4 41.4 41.4 41.4 ...
## $ longitude : num 2.13 2.2 2.17 2.19 2.15 ...
## $ has_swimming_pool : chr "False" "False" "True" "True" ...
## $ has_bar : chr "True" "True" "True" "True" ...
## $ has_restaurant : chr "True" "True" "True" "True" ...
## $ has_breakfast : chr "True" "True" "True" "True" ...
## $ has_gym : chr "False" "True" "True" "True" ...
## $ has_reception_24h : chr "True" "True" "True" "True" ...
## $ has_ac : chr "True" "True" "True" "True" ...
```

```
## $ has_strongbox : chr "False" "True" "True" "True" ...
## $ admits_pets : chr "False" "False" "False" "True" ...
## $ rooms : num 83 154 75 473 53 16 19 64 81 101 ...
## $ suites : chr "False" "False" "False" "True" ...
## $ sea_views_rooms : chr "False" "False" "True" "True" ...
## $ non_smoking_rooms : chr "True" "True" "True" "True" ...
## $ landmarks_views_rooms : chr "False" "False" "False" "False" ...
## $ city_views_rooms : chr "False" "False" "False" "False" ...
## $ family_rooms : chr "True" "False" "False" "False" ...
## $ style : chr "Familiar" "Ecológico" "Vistas al parque" "Vistas a la bahía" ...
## $ tripadvisor_clasification: chr "51 de 100" "94 de 100" "97 de 100" "75 de
## 100" ...
## $ language_spanish : chr "True" "True" "False" "True" ...
## $ language_catalan : chr "True" "True" "True" "False" ...
## $ language_french : chr "True" "True" "True" "False" ...
## $ language_english : chr "True" "True" "True" "False" ...
## $ language_italian : chr "True" "True" "False" "False" ...
## $ language_bulgarian : chr "True" "False" "False" "False" ...
## $ language_russian : chr "False" "False" "False" "False" ...
## $ language_portuguese : chr "False" "True" "False" "False" ...
## $ prat_distance : int 7 15 11 12 12 11 13 12 12 13 ...
## $ timestamp : chr "2020-11-02 15:37:45.938143" "2020-11-02 15:37:58.762559"
## "2020-11-02 15:38:01.677586" "2020-11-02 15:38:06.467551" ...
```

En cuanto a la selección de datos, vamos a quedarnos con aquellos que tengan un rango de precio informado, ya que como vamos a explicar será una de las variable objetivo del análisis. En cuanto a los diferentes atributos, vamos a empezar eliminando las columnas `name` y `timestamp`, ya que la primera contiene el nombre del hotel, el cual no es necesario para los análisis y el segundo la fecha de la captura de los datos, que tampoco nos aporta información relativa al hotel.

Observamos también dos columnas relativas al precio. `price` contiene el precio por noche en el momento de la extracción del dato proporcionado por terceros (Booking, Expedia...). `price_range` es una estimación más general del precio del hotel y es proporcionado por la propia Tripadvisor. Además, la variable `price`, a parte de tener un carácter temporal, tiene muchos valores nulos, como veremos más tarde. Por este motivo, eliminaremos la variable `price` y obtendremos los precios directamente de la variable `price_range`.

```
# eliminamos las 3 variables mencionadas
df <- df[, !(names(df) %in% c("name", "timestamp", "price"))]
# eliminamos las filas que tengan nulos en price_range (string vacío)
df <- df[df$price_range != "",]
```

3 Limpieza de los datos

3.1 Preprocesado

El primer paso será convertir el tipo de las variables. Vamos a empezar convirtiendo a tipo factor las variables categóricas.

```
categorical_vars <- c("has_swimming_pool", "has_bar", "has_restaurant", "has_breakfast",
  "has_gym", "has_reception_24h", "has_ac", "has_strongbox",
  "admits_pets", "suites", "sea_views_rooms", "non_smoking_rooms",
  "landmarks_views_rooms", "city_views_rooms", "family_rooms",
  "style", "language_spanish", "language_catalan", "language_french",
  "language_english", "language_italian", "language_bulgarian",
```

```

                                "language_russian", "language_portuguese")
for(v in categorical_vars){
  df[[v]] <- factor(df[[v]])
}

```

A continuación, debemos estandarizar la variable `price_range`. Esta variable son rangos de precios medios. El problema es que estos rangos no están predefinidos de ninguna manera. Para solucionar este problema, vamos a transformar la variable para obtener el precio medio del intervalo y luego discretizar la variable en los valores “Barato”, “Medio”, “Caro” y “Lujo” según las siguientes reglas:

- **Barato:** precio < 60 euros
- **Medio:** 60 >= precio < 150
- **Caro:** 150 >= precio < 300
- **Lujo:** precio >= 300

A continuación, se muestra la variable antes y después de esta transformación.

```

head(df$price_range, 3)
price_range_clean <- str_replace_all(df$price_range, "€|[:space:]|\\.", "")
price_range_clean <- str_split(price_range_clean, "-")
new_price_range <- rep(NA, length(price_range_clean))
for(i in 1:length(price_range_clean)){
  range <- as.list(price_range_clean[i][[1]])
  if(range[[1]] != ""){
    m <- mean(c(strtoi(range[[1]]), strtoi(range[[2]])))
    new_price_range[i] <- m
  }
}
df$price_range <- new_price_range
head(df$price_range, 3)

# discretizar
df$price_range_disc[df$price_range < 60] <- "Barato"
df$price_range_disc[(df$price_range >= 60) & (df$price_range < 150)] <- "Medio"
df$price_range_disc[(df$price_range >= 150) & (df$price_range < 300)] <- "Caro"
df$price_range_disc[df$price_range >= 300] <- "Lujo"
df$price_range_disc <- factor(df$price_range_disc,
                             levels = c("Barato", "Medio", "Caro", "Lujo"))
head(df$price_range_disc, 3)

```

```

## [1] "52 \200 - 166 \200" "79 \200 - 195 \200" "152 \200 - 283 \200"
## [1] 109.0 137.0 217.5
## [1] Medio Medio Caro
## Levels: Barato Medio Caro Lujo

```

Ahora, vamos a realizar un procesado para la variable `tripadvisor_clasification`. Es una puntuación sobre 100 y está expresada como un string de la forma “88 de 100”, por ejemplo. El objetivo es obtener la puntuación y convertirla a entero.

```

head(df$tripadvisor_clasification, 5)
new_class <- rep(NA, nrow(df))
for(i in 1:nrow(df)){
  cl <- df$tripadvisor_clasification[i]
  index <- str_locate(cl, " de 100")[1]
  new_class[i] <- as.integer(str_sub(cl, 1, index))
}

```

```
df$tripadvisor_clasification <- new_class
head(df$tripadvisor_clasification, 5)
```

```
## [1] "51 de 100" "94 de 100" "97 de 100" "75 de 100" "100 de 100"
## [1] 51 94 97 75 100
```

Por último, vamos a estandarizar la variable `zone`, la cual contiene la zona de Barcelona donde se encuentra el hotel. Sin embargo, vemos que esta variable no está normalizada y que según el hotel, la ubicación está más o menos detallada.

```
head(df$zone)
```

```
## [1] "Barcelona"
## [2] "Poblenou / Sant Martí"
## [3] "Montjuic / Sants-Montjuïc"
## [4] "Ciutat Vella / El Port Vell / Barceloneta"
## [5] "Sant Gervasi-Galvany"
## [6] "El Ensanche (Eixample) / Sant Antoni"
```

Si nos fijamos en esta muestra de zonas, vemos que el barrio está indicado en la primera parte (a la izquierda de la primera “/”), por lo que transformaremos esta variable para quedarnos solo con este fragmento.

```
df$zone <- factor(substr(df$zone, 1, str_locate(df$zone, " / |$") - 1))
head(df$zone)
```

```
## [1] Barcelona          Poblenou              Montjuic
## [4] Ciutat Vella        Sant Gervasi-Galvany  El Ensanche (Eixample)
## 33 Levels: Barcelona ... Vila de Gràcia
```

Sin embargo, se obtiene un número bastante elevado de barrios diferentes. Lo que haremos será crear una variable nueva a partir de esta (`district`) que indique el distrito al que pertenece cada barrio.

```
df$district[df$zone %in% c("El Ensanche (Eixample)")] <- "Eixample"
df$district[df$zone %in% c("Sants-Montjuïc", "La Marina de Port",
                           "Montjuic")] <- "Sants-Montjuïc"
df$district[df$zone %in% c("Les Corts de Sarrià", "Pedralbes",
                           "La Maternitat i Sant Ramon")] <- "Les Corts"
df$district[df$zone %in% c("Sarrià", "Sant Gervasi-Galvany", "Sant Gervasi-La Bonanova",
                           "El Putxet i el Farró", "Les Tres Torres")] <- "Sarrià"
df$district[df$zone %in% c("Vallcarca i els Penitents", "El Coll", "La Salut",
                           "Vila de Gràcia",
                           "El Camp d'en Grassot i Gràcia Nova")] <- "Gràcia"
df$district[df$zone %in% c("La Vila Olímpica del Poblenou",
                           "El Parc i la Llacuna del Poblenou",
                           "Poblenou", "Diagonal Mar i el Front Marítim del Poblenou",
                           "El Clot", "Provençals del Poblenou",
                           "Sant Martí")] <- "Sant Martí"
df$district[df$zone %in% c("Sant Andreu de Palomar", "Sant Andreu", "La Sagrera",
                           "El Bon Pastor")] <- "Sant Andreu"
df$district[df$zone %in% c("Ciutat Vella")] <- "Ciutat Vella"
df$district[df$zone %in% c("Nou Barris")] <- "Nou Barris"
df$district[df$zone %in% c("La Vall d'Hebron", "Horta-Guinardó" )] <- "Horta-Guinardó"
df$district <- factor(df$district)
summary(df$district)
```

```
## Ciutat Vella          Eixample          Gràcia Horta-Guinardó      Les Corts
##           268             440             72             18             33
## Nou Barris      Sant Andreu      Sant Martí Sants-Montjuïc      Sarrià
```

```
##          4          6          90          105          62
##      NA's
##          15
```

De esta forma, tenemos datos más manejables y hemos conseguido reducir mucho el número de posibles valores para esta variable.

3.2 Elementos nulos

A continuación, vamos a inspeccionar el dataset en busca de valores nulos.

```
colSums(is.na(df))

##          stars          score          score_location
##          154          103          211
##      score_cleaning      score_service      score_value_money
##          208          190          226
##          ranking          price_range          opinions
##          105           0          102
##      opinions_excellent      opinions_good      opinions_normal
##          103          103          103
##          opinions_bad      opinions_awful          num_qa
##          103          103          102
##      nearby_restaurants      nearby_attractions          zone
##           5           5           0
##          latitude          longitude      has_swimming_pool
##           3           3           0
##          has_bar          has_restaurant          has_breakfast
##           0           0           0
##          has_gym      has_reception_24h          has_ac
##           0           0           0
##          has_strongbox          admits_pets          rooms
##           0           0          230
##          suites          sea_views_rooms      non_smoking_rooms
##           0           0           0
##      landmarks_views_rooms      city_views_rooms          family_rooms
##           0           0           0
##          style      tripadvisor_clasificacion      language_spanish
##           0           5           0
##      language_catalan          language_french      language_english
##           0           0           0
##      language_italian          language_bulgarian      language_russian
##           0           0           0
##      language_portuguese          prat_distance      price_range_disc
##           0           3           0
##          district
##          15
```

Dependiendo de la variable, vamos a tratar los nulos de forma diferente.

- Las que hacen referencia al número de opiniones las vamos a imputar con un 0, ya que si es nulo es que no se ha encontrado ninguna opinión.
- Para la distancia al aeropuerto del Prat, latitud y longitud vamos a imputar los nulos con la mediana, ya que al estar en Barcelona es muy probable que estos valores sean similares al resto.
- Para las atracciones y restaurantes cercanos vamos a imputar también con la mediana (solo 5 nulos).

```

# opinión
opinion_vars <- c("opinions", "opinions_excellent", "opinions_good", "opinions_normal",
                 "opinions_bad", "opinions_awesome", "num_qa")
for(var in opinion_vars){
  df[[var]][is.na(df[[var]])] <- 0
}
# prat_distance, latitude, longitude, nearby_restaurants, nearby_attractions
median_vars <- c("prat_distance", "latitude", "longitude", "nearby_restaurants",
                 "nearby_attractions")
for(var in median_vars){
  df[[var]][is.na(df[[var]])] <- median(df[[var]], na.rm = TRUE)
}
colSums(is.na(df))

```

```

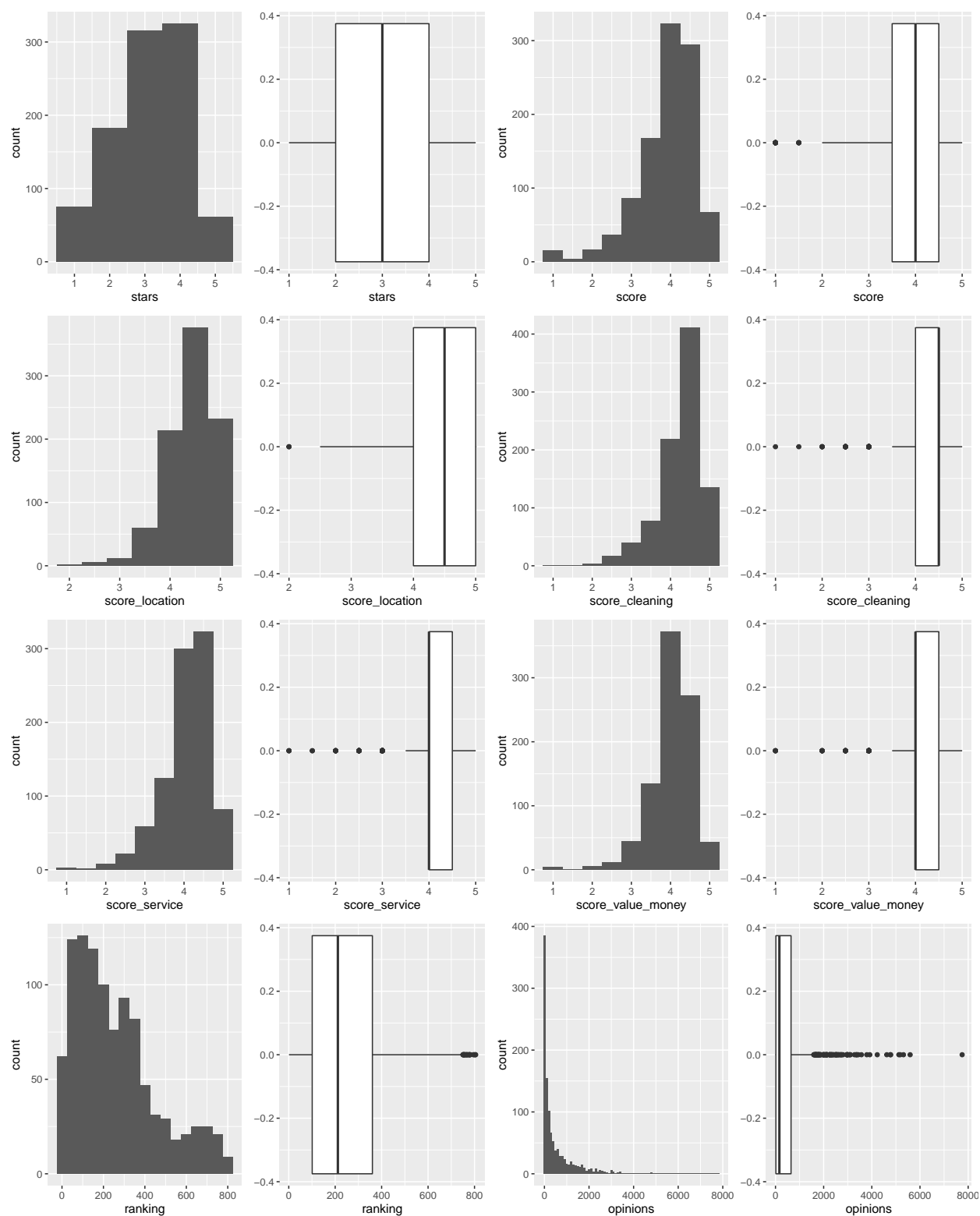
##          stars          score      score_location
##          154          103          211
##      score_cleaning      score_service      score_value_money
##          208          190          226
##          ranking      price_range      opinions
##          105           0           0
##      opinions_excellent      opinions_good      opinions_normal
##           0           0           0
##          opinions_bad      opinions_awesome      num_qa
##           0           0           0
##      nearby_restaurants      nearby_attractions      zone
##           0           0           0
##          latitude      longitude      has_swimming_pool
##           0           0           0
##          has_bar      has_restaurant      has_breakfast
##           0           0           0
##          has_gym      has_reception_24h      has_ac
##           0           0           0
##          has_strongbox      admits_pets      rooms
##           0           0          230
##          suites      sea_views_rooms      non_smoking_rooms
##           0           0           0
##      landmarks_views_rooms      city_views_rooms      family_rooms
##           0           0           0
##          style tripadvisor_clasification      language_spanish
##           0           5           0
##      language_catalan      language_french      language_english
##           0           0           0
##      language_italian      language_bulgarian      language_russian
##           0           0           0
##      language_portuguese      prat_distance      price_range_disc
##           0           0           0
##          district
##          15

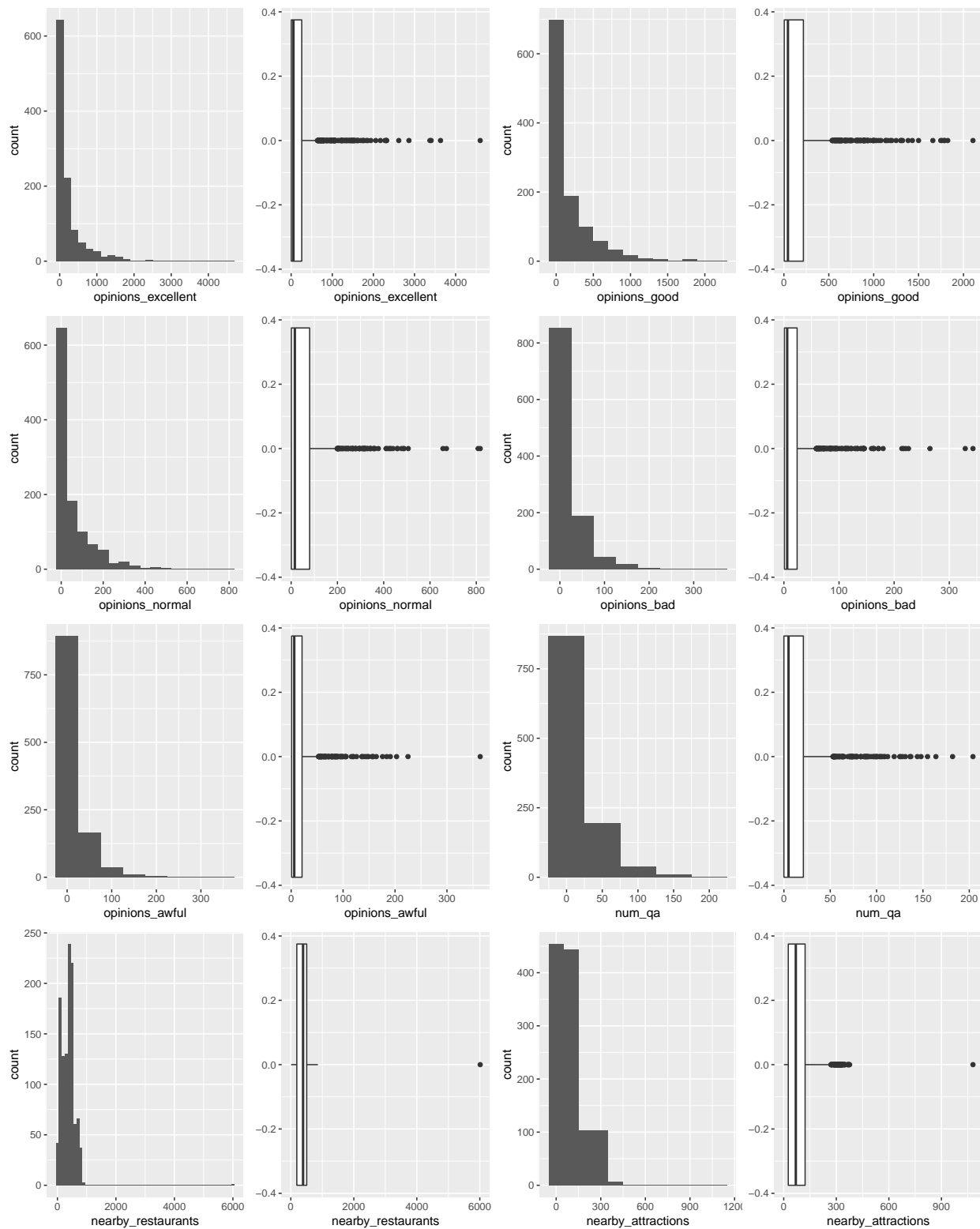
```

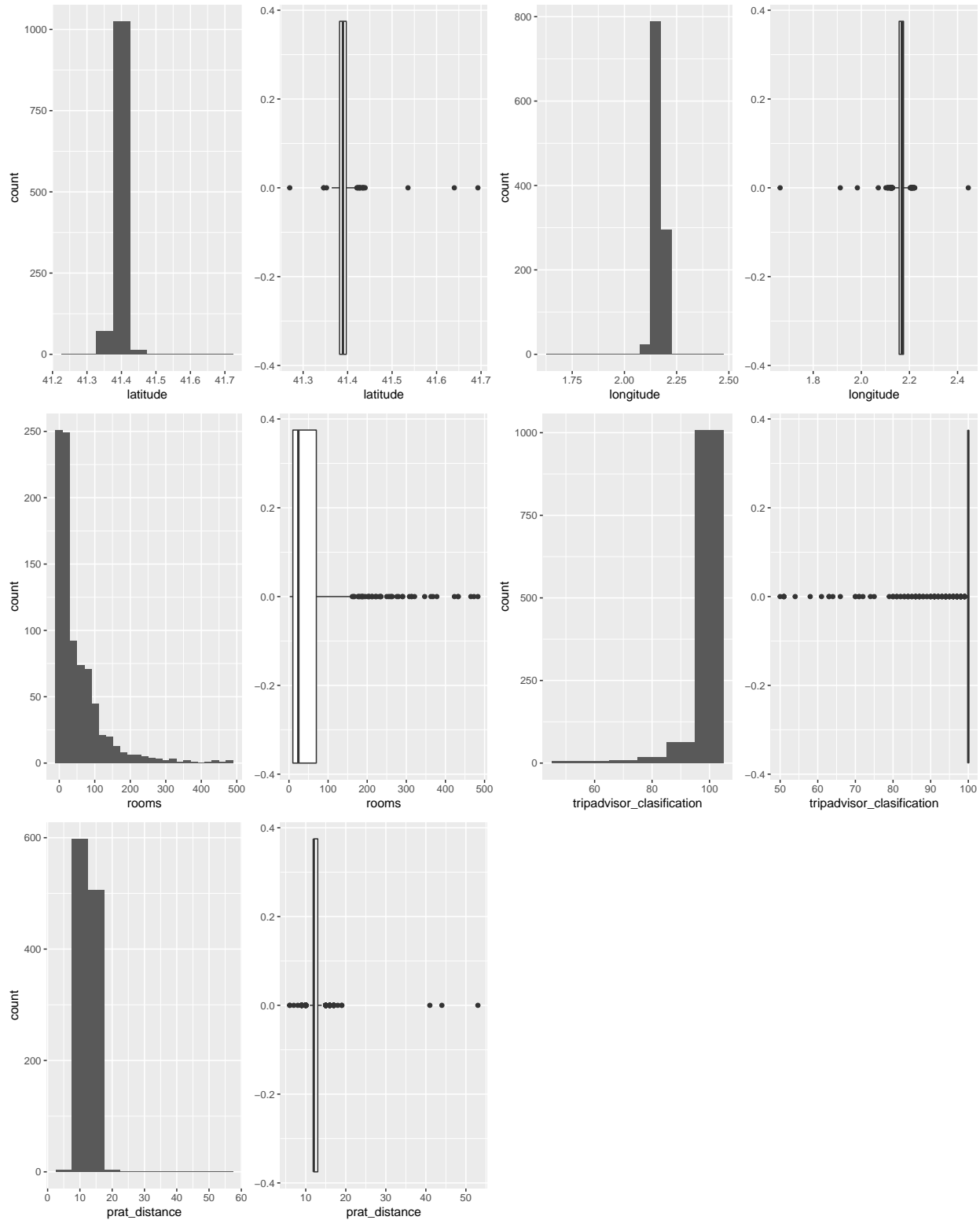
De momento no vamos a imputar el resto de nulos. Será en el momento que necesitemos usar estas columnas cuando eliminaremos aquellas filas que tengan NA's, ya que debido a que en algunas columnas el porcentaje de nulos está cerca del 20%, es mejor eliminarlas para no desvirtuar las posibles conclusiones. Sin embargo, no las vamos a eliminar ahora, ya que pueden tener información importante para el análisis del resto de variables.

3.3 Valores extremos

Vamos a inspeccionar las variables numéricas en busca de outliers. Nos ayudaremos de un histograma y de un boxplot de cada variable.







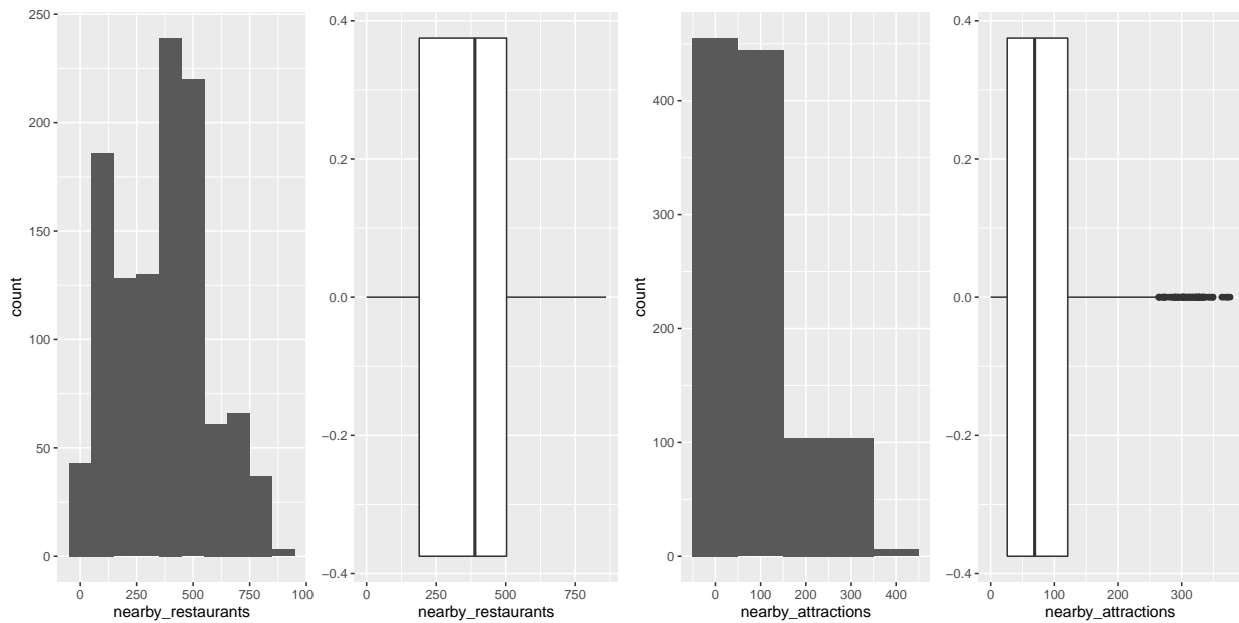
Como observamos, si nos fijamos en los plots que hacen referencia a las variables de estrellas y puntuaciones en diferentes ámbitos observamos una media por encima de 4 sobre 5. Los únicos outliers que vemos son en puntuaciones por debajo de las 3 estrellas en general. Sin embargo, son valores perfectamente válidos, ya que están dentro del rango correcto de valores esperado (entre 1 y 5).

En cuanto a las diferentes variables sobre número de opiniones, se observa una clara asimetría y bastantes outliers por la parte superior. Sin embargo, no parecen valores incorrectos, ya que son valores razonables que y no demasiado alejados del resto.

En cambio, si observamos la cantidad de restaurantes y actividades cercanas, vemos claramente outliers. Si nos fijamos en `nearby_restaurants`, hay una observación por encima de 6000 (la media es 370) y algunas de atracciones por encima de 1000 (la media es 97). Buscando estos casos en web de Tripadvisor, hemos detectado que son hoteles en las afueras de Barcelona, y que en lugar de contabilizar estos números en un radio de 500 metros desde el hotel (como se hace en los demás), estas cifras hacen referencia a un radio de 10km. Por lo tanto, al estar alejados de Barcelona, vamos a imputar estos outliers con el valor 0.

```
df$nearby_restaurants[df$nearby_restaurants > 5000] <- 0
df$nearby_attractions[df$nearby_attractions > 1000] <- 0
```

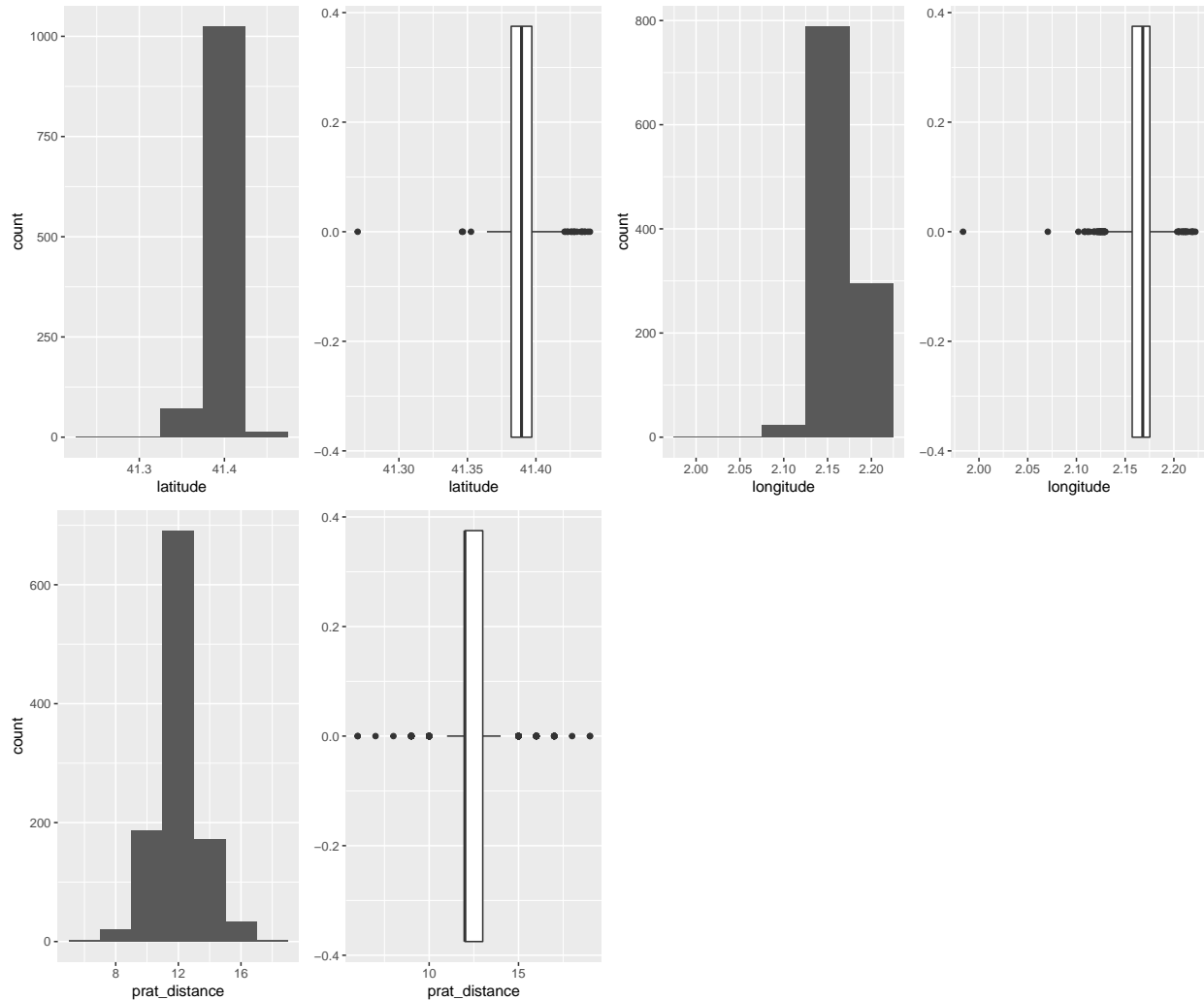
Si volvemos a enseñar los plots de estas variables, vemos que se han eliminado estos valores extremos.



Seguidamente, observamos las variables que hacen referencia a la ubicación del hotel (longitud, latitud y distancia al aeropuerto del Prat). Vemos ciertos outliers en la latitud y longitud. Aquellas observaciones más extremas podrían ser también probablemente los hoteles que estén más lejos del aeropuerto del Prat. Mirando esta variable, observamos que hay hoteles a más de 30 kilómetros del aeropuerto. Sin embargo, estos hoteles los descartaremos, ya que al estar tan alejados de Barcelona no nos interesan. Se ha comprobado que el hotel más lejano se sitúa en Berga, ciudad que está a más de 100km de Barcelona y que obviamente no entra en el objetivo de este análisis. Por lo tanto, vamos a eliminar aquellos hoteles que estén a una distancia superior a 30km del Prat.

```
df <- df[df$prat_distance < 30,]
```

Veamos cómo han quedado estas 3 variables mencionadas.



Finalmente, en las variables `rooms` y `tripadvisor_classification`, vemos que los valores extremos parecen correctos y tienen sentido. Por ejemplo, en cuanto a esta última variable (la cual está sobre 100) vemos que la mayoría de hoteles tienen una puntuación muy alta y que todos están entre 50 y 100, valores que son correctos. Por lo tanto, en este punto ya tenemos el dataset preparado para su análisis.

4 Análisis de los datos

4.1 Comparación de la limpieza entre hoteles baratos y de lujo

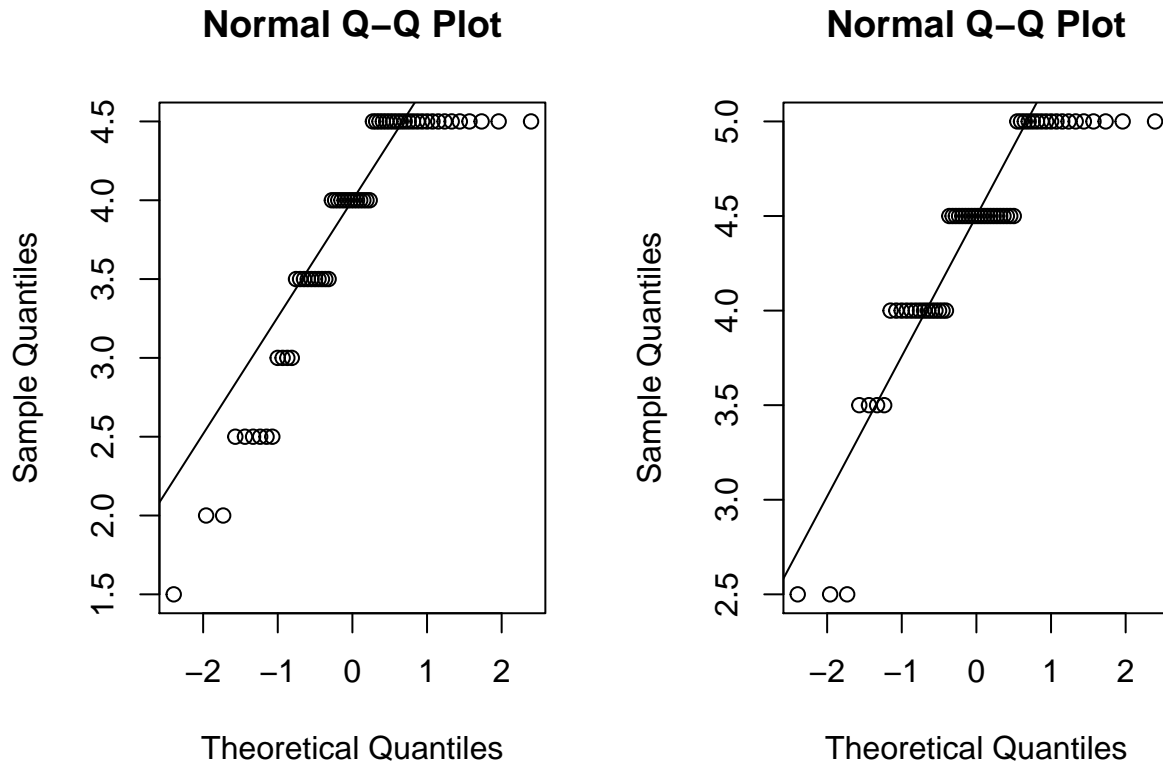
En este primer análisis vamos a ver si la calificación de limpieza de un hotel es superior en hoteles de lujo respecto a los más baratos. Vemos primero cuantos hoteles hay de cada tipo.

```
table(df$price_range_disc)
```

```
##
## Barato Medio Caro Lujo
## 70 661 302 77
```

```
cleaning_cheap <- df$score_cleaning[df$price_range_disc == "Barato"]
cleaning_luxury <- df$score_cleaning[df$price_range_disc == "Lujo"]
```

Vemos que hay una cantidad parecida de hoteles baratos y de lujo. El primer paso será ver si la variable `price_range_disc` sigue o no una distribución normal. Para ello, vamos a visualizar los QQ Plots de cada grupo y hacer el Test de Shapiro-Wilk.



```
##
## Shapiro-Wilk normality test
##
## data:  cleaning_cheap
## W = 0.82755, p-value = 7.042e-07
##
##
## Shapiro-Wilk normality test
##
## data:  cleaning_luxury
## W = 0.82336, p-value = 5.425e-07
```

Ya vemos en los plots (en la izquierda `cleaning_cheap` y derecha `cleaning_luxury`) que no se cumple la normalidad en los datos. La misma conclusión obtenemos con el Shapiro test, en los cuales obtenemos *p-values* muy inferiores a 0.05, con lo que se rechaza la hipótesis nula y se concluye que no siguen una distribución normal.

También debemos comprobar si las varianzas de las dos muestras son iguales o no. Aplicamos el F Test para compararlas.

```
var.test(cleaning_cheap, cleaning_luxury)
```

```
##
## F test to compare two variances
```

```
##
## data:  cleaning_cheap and cleaning_luxury
## F = 1.6415, num df = 59, denom df = 59, p-value = 0.0593
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.9805281 2.7481436
## sample estimates:
## ratio of variances
##          1.641533
```

Observamos que el p-value es 0.0593, el cual es mayor que el nivel de significancia 0.05, por lo que no podemos rechazar la hipótesis nula y consecuentemente se considera que las varianzas son iguales.

Por tanto, para comparar ambas muestras, debemos recurrir a pruebas no paramétricas. Concretamente, aplicaremos el test de suma de rangos de Wilcoxon con las siguientes hipótesis.

$$H_0 : \mu_{cheap} - \mu_{luxury} = 0$$

$$H_1 : \mu_{cheap} - \mu_{luxury} < 0$$

```
wilcox.test(cleaning_cheap, cleaning_luxury , alternative = "less")
```

```
##
## Wilcoxon rank sum test with continuity correction
##
## data:  cleaning_cheap and cleaning_luxury
## W = 1009, p-value = 8.495e-06
## alternative hypothesis: true location shift is less than 0
```

Obtenemos un *p-value* de 8.495e-06, por lo que rechazamos la hipótesis nula y adoptamos la alternativa, de modo que se concluye que los hoteles de lujo tienen mejor limpieza que los más baratos.

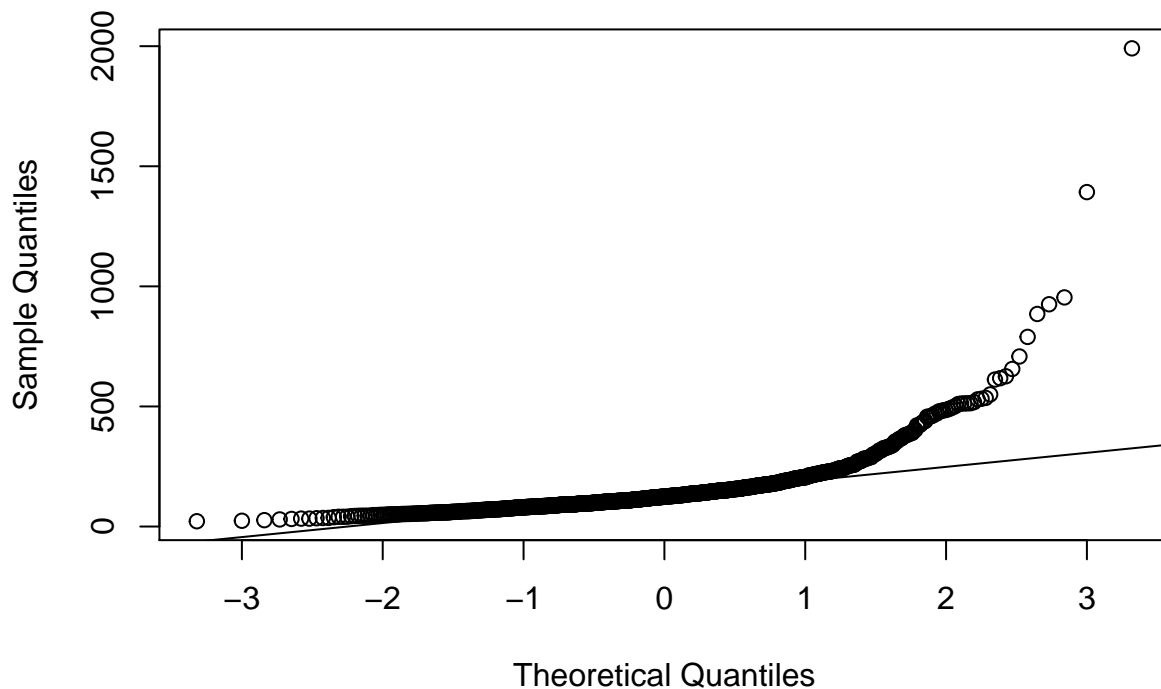
4.2 Comparación de precios entre hoteles de distintos distritos

En este apartado vamos a comprobar si hay diferencia de precios entre diferentes barrios de Barcelona. Como en muchas zonas no hay demasiados hoteles, vamos a comparar solo algunos distritos que aglomeren más hoteles.

Para ver qué test aplicamos, debemos comprobar antes si el precio sigue una distribución normal.

```
qqnorm(df$price_range)
qqline(df$price_range)
```

Normal Q-Q Plot



```
shapiro.test(df$price_range)
```

```
##  
##  Shapiro-Wilk normality test  
##  
## data:  df$price_range  
## W = 0.59891, p-value < 2.2e-16
```

A partir del plot y el Shapiro-Wilk vemos que no se cumple la normalidad en los datos. Antes de aplicar como antes el test de Wilcoxon, debemos asegurarnos de que las varianzas sean iguales.

```
price_sants <- df$price_range[df$district == "Sants-Montjuic"]  
price_ciutat_vella <- df$price_range[df$district == "Ciutat Vella"]  
price_eixample <- df$price_range[df$district == "Eixample"]  
price_sant_marti <- df$price_range[df$district == "Sant Martí"]  
var.test(price_sants, price_eixample)  
var.test(price_sant_marti, price_ciutat_vella)
```

```
##  
##  F test to compare two variances  
##  
## data:  price_sants and price_eixample  
## F = 0.79637, num df = 104, denom df = 439, p-value = 0.1591  
## alternative hypothesis: true ratio of variances is not equal to 1  
## 95 percent confidence interval:  
##  0.5963579 1.0944749  
## sample estimates:  
## ratio of variances
```



```
##          0.7963717
##
##
## F test to compare two variances
##
## data:  price_sant_marti and price_ciutat_vella
## F = 1.1233, num df = 89, denom df = 267, p-value = 0.4803
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.810555 1.603079
## sample estimates:
## ratio of variances
##          1.12328
```

Una vez comprobado que las varianzas de precios en estos distritos son iguales, vamos a realizar dos pruebas de contrastes de hipótesis. En la primera vamos a ver si los precios en el Eixample son mayores que en Sants y en el segundo vamos a comprobar si los precios en Sant Martí son diferentes o no de los de Ciutat Vella.

•

$$H_0 : \mu_{eixample} - \mu_{sants} = 0; H_1 : \mu_{eixample} - \mu_{sants} > 0$$

•

$$H_0 : \mu_{santmarti} - \mu_{ciutatvella} = 0; H_1 : \mu_{santmarti} - \mu_{ciutatvella} \neq 0$$

```
wilcox.test(price_eixample, price_sants, alternative = "greater")
wilcox.test(price_sant_marti, price_ciutat_vella, alternative = "two.sided")
```

```
##
## Wilcoxon rank sum test with continuity correction
##
## data:  price_eixample and price_sants
## W = 28194, p-value = 0.0002217
## alternative hypothesis: true location shift is greater than 0
##
##
## Wilcoxon rank sum test with continuity correction
##
## data:  price_sant_marti and price_ciutat_vella
## W = 13834, p-value = 0.03687
## alternative hypothesis: true location shift is not equal to 0
```

Obtenemos *p-values* inferiores al nivel de significancia 0.05 en ambos casos, por lo que rechazamos la hipótesis nula. Por lo tanto, podemos concluir que los precios de los hoteles del Eixample son más caros que los de Sants y que los del distrito de Sant Martí son diferentes a los de Ciutat Vella.

5 Representación de los resultados a partir de tablas y gráficas

6 Conclusiones

Contribuciones	Firma
Investigación previa	Alberto Giménez
Redacción de las respuestas	Alberto Giménez
Desarrollo código	Alberto Giménez