



UPC
Universidad Peruana
de Ciencias Aplicadas

Ciencias de la Computación 2022-01

Administración de la Información

TA1

Integrantes :

Salinas Roca, Antonio - U201924931

Guillén Rojas, Daniel Carlos - U201920113

Wu Pan, Tito Peng - U201921200

Profesora :

Reyes Silva, Patricia Daniela

Lima, 2022

Índice

1. Caso de análisis.....	2
2. Conjunto de datos (data set).....	2
3. Análisis exploratorio de los datos.....	3
4. Conclusiones preliminares.....	8
5. Bibliografía.....	15

1. Caso de análisis

Los datos utilizados en este proyecto fueron extraídos del artículo *Hotel booking demand datasets* desarrollado por Nuno Antonio, Anade Almeida y Luis Nunes, publicado el 2018 por la línea editorial Elsevier, una empresa mundialmente reconocida, dirigida a la literatura científica, ubicada en Ámsterdam.

En este artículo se describen dos grupos de datos de demanda hotelera, un hotel resort y uno urbano. Debido a que se tratarán con datos reales, se eliminaron datos que vulneren la privacidad de los establecimientos y sus clientes. Debido a no contar con datos comerciales reales, no pueden ser tratados con fines científicos o educativos. Sin embargo, sí pueden usarse con fines educativos en la gestión de ingresos, la minería de datos o el aprendizaje autónomo, así como en otros campos.

2. Conjunto de datos (data set)

Los datos están organizados en 31 variables que describen 40 060 observaciones del hotel resort y 79 330 del hotel urbano. Los datos son provenientes de las reservas de ambos hoteles entre el 1 de julio de 2015 y el 31 de agosto de 2017, incluyendo reservas efectuadas y canceladas.

3. Análisis exploratorio de datos

Cargar datos:



Para cargar los datos usamos la función `read.csv`, esta función nos permitirá tanto leer los datos y cargarlos. Los organizamos en dos tablas, una tabla con los archivos por defecto y otra que contará con los datos a los cuales modificaremos para hacer una comparación al final.

```
#-----  
#CARGAR DATOS  
#-----  
library(dplyr)  
Tabla_modificada <- read.csv("hotel_bookings_miss.csv", header = TRUE, stringsAsFactors = FALSE)  
Tabla_original <- read.csv("hotel_bookings_miss.csv", header = TRUE, stringsAsFactors = FALSE)  
|
```

Inspeccionar datos:

Utilizamos “`view()`” para ver la cantidad de objetos y variables que contiene nuestra data.

```
#-----  
#INSPECCIONAR DATOS  
#-----  
view(Tabla_original)  
|
```

Data		
🔍 Tabla_modificada	119390 obs. of 32 variables	
🔍 Tabla_original	119390 obs. of 32 variables	

id	hotel	is_cancelled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	stays_in_week_nights	adults	children	babies	meal	country	market_segment	distribution_channel
1	Resort Hotel	0	342	2015	July		27	1	NA	0	2	0	0	BB	PRT	Direct
2	Resort Hotel	0	737	2015	July		27	1	0	0	2	0	0	BB	PRT	Direct
3	Resort Hotel	0	7	2015	July		27	1	0	1	1	0	0	BB	GBR	Direct
4	Resort Hotel	0	13	2015	July		27	1	0	1	1	0	0	BB	GBR	Corporate
5	Resort Hotel	0	14	2015	July		27	1	0	2	2	0	0	BB	GBR	Online TA
6	Resort Hotel	0	14	2015	July		27	1	0	2	2	0	0	BB	GBR	Online TA
7	Resort Hotel	0	0	2015	July		27	1	0	2	2	0	0	BB	PRT	Direct
8	Resort Hotel	0	9	2015	July		27	1	0	2	2	0	0	BB	PRT	Direct
9	Resort Hotel	1	85	2015	July		27	1	0	3	2	0	0	BB	PRT	Online TA
10	Resort Hotel	1	75	2015	July		27	1	0	3	2	0	0	BB	PRT	Online TA
11	Resort Hotel	1	23	2015	July		27	1	0	4	2	0	0	BB	PRT	Online TA
12	Resort Hotel	0	35	2015	July		27	1	0	4	2	0	0	BB	PRT	Online TA
13	Resort Hotel	0	68	2015	July		27	1	0	4	2	0	0	BB	USA	Online TA
14	Resort Hotel	0	18	2015	July		27	1	0	4	2	1	0	BB	ESP	Online TA
15	Resort Hotel	0	37	2015	July		27	1	0	4	2	0	0	BB	PRT	Online TA
16	Resort Hotel	0	68	2015	July		27	1	0	4	2	0	0	BB	IRL	Online TA
17	Resort Hotel	0	37	2015	July		27	1	0	4	2	0	0	BB	PRT	Online TA
18	Resort Hotel	0	12	2015	July		27	1	0	1	2	0	0	BB	IRL	Online TA
19	Resort Hotel	0	0	2015	July		27	1	0	1	2	0	0	BB	FRA	Corporate
20	Resort Hotel	0	7	2015	July		27	1	0	4	2	0	0	BB	GBR	Direct
21	Resort Hotel	0	37	2015	July		27	1	1	4	1	0	0	BB	GBR	Online TA
22	Resort Hotel	0	72	2015	July		27	1	2	4	2	0	0	BB	PRT	Direct
23	Resort Hotel	0	72	2015	July		27	1	2	4	2	0	0	BB	PRT	Direct
24	Resort Hotel	0	72	2015	July		27	1	2	4	2	0	0	BB	PRT	Direct
25	Resort Hotel	0	127	2015	July		27	1	2	5	2	0	0	BB	GBR	Online TA
26	Resort Hotel	0	78	2015	July		27	1	2	5	2	0	0	BB	PRT	Online TA
27	Resort Hotel	0	48	2015	July		27	1	2	5	2	0	0	BB	IRL	Online TA
28	Resort Hotel	1	60	2015	July		27	1	2	5	2	0	0	BB	PRT	Online TA
29	Resort Hotel	0	77	2015	July		27	1	2	5	2	0	0	BB	PRT	Online TA
30	Resort Hotel	0	99	2015	July		27	1	2	5	2	0	0	BB	PRT	Online TA
31	Resort Hotel	0	118	2015	July		27	1	4	10	1	0	0	BB	NULL	Direct
32	Resort Hotel	0	95	2015	July		27	1	4	11	2	0	0	BB	GBR	Online TA
33	Resort Hotel	1	96	2015	July		27	1	4	8	2	0	0	BB	PRT	Direct

Preprocesar datos:

Para realizar el preprocesado de los datos, tendremos que aplicar métodos para reemplazar tanto a los datos que no cuenten con valores (*NA*) y para los datos atípicos existentes dentro de nuestro dataset.

Primero aplicaremos el método de reemplazo a los valores *NA*, les añadiremos valores que se obtengan mediante la media poblacional de las columnas que contuviesen datos numéricos.

```
# VERIFICAR VALORES NA
#-----
verificar_NA <- function(df){
  for (variable in names(df)) {
    print(variable)
    var <- is.na(df[,c(variable)])
    print(table(var))
  }
}

# REEMPLAZAR VALORES NA x MEDIA POBLACIONAL
#-----
Reemplazar_NA <- function(columna){
  colum <- Tabla_modificada[,c(columna)]
  Tabla_modificada[,c(columna)][colum == 0] <- NA
  temp.mean <- ifelse(is.na(colum), mean(colum, na.rm = TRUE), colum)
  temp.mean <- round(temp.mean, digits = 0)
  return(temp.mean)
}
```

Veremos los valores *NA*.

```
# VER DATOS NA
#-----
verificar_NA(Tabla_modificada)
```

```
Console Terminal x Jobs x
R 3.6.3 · ~/Rstudio/ea-2021-1-cc51/ ↻
> verificar_NA(Tabla_modificada)
[1] "i..hotel"
var
  FALSE
119390
[1] "is_canceled"
var
  FALSE
119390
[1] "lead_time"
var
  FALSE TRUE
119369 21
[1] "arrival_date_year"
var
  FALSE TRUE
119384 6
[1] "arrival_date_month"
var
  FALSE
119390
[1] "arrival_date_week_number"
var
  FALSE TRUE
119365 25
[1] "arrival_date_day_of_month"
var
  FALSE TRUE
119383 7
[1] "stays_in_weekend_nights"
var
  FALSE TRUE
119365 25
[1] "stays_in_week_nights"
var
  FALSE TRUE
119378 12
[1] "adults"
var
  FALSE TRUE
119378 12
[1] "children"
var
  FALSE TRUE
119386 4
[1] "babies"
var
  FALSE TRUE
119358 32
[1] "meal"
var
  FALSE
119390
[1] "country"
var
  FALSE
119390
[1] "market_segment"
```

Modificamos los datos *NA* de la tabla en modificación.

```
# CAMBIAR NA
#-----

# stays_in_weekend_nights
Tabla_modificada$stays_in_weekend_nights.mean <- Reemplazar_NA("stays_in_weekend_nights")
Tabla_modificada$stays_in_weekend_nights <- NULL
names(Tabla_modificada)[32] <- "stays_in_weekend_nights"

# lead_time
Tabla_modificada$lead_time.mean <- Reemplazar_NA("lead_time")
Tabla_modificada$lead_time <- NULL
names(Tabla_modificada)[32] <- "lead_time"

# arrival_date_year
Tabla_modificada$arrival_date_year.mean <- Reemplazar_NA("arrival_date_year")
Tabla_modificada$arrival_date_year <- NULL
names(Tabla_modificada)[32] <- "arrival_date_year"

# arrival_date_week_number
Tabla_modificada$arrival_date_week_number.mean <- Reemplazar_NA("arrival_date_week_number")
Tabla_modificada$arrival_date_week_number <- NULL
names(Tabla_modificada)[32] <- "arrival_date_week_number"

# stays_in_week_nights
Tabla_modificada$stays_in_week_nights.mean <- Reemplazar_NA("stays_in_week_nights")
Tabla_modificada$stays_in_week_nights <- NULL
names(Tabla_modificada)[32] <- "stays_in_week_nights"

# adults
Tabla_modificada$adults.mean <- Reemplazar_NA("adults")
Tabla_modificada$adults <- NULL
names(Tabla_modificada)[32] <- "adults"

# children
Tabla_modificada$children.mean <- Reemplazar_NA("children")
Tabla_modificada$children <- NULL
names(Tabla_modificada)[32] <- "children"

# babies
Tabla_modificada$babies.mean <- Reemplazar_NA("babies")
Tabla_modificada$babies <- NULL
names(Tabla_modificada)[32] <- "babies"

# | days_in_waiting_list
Tabla_modificada$days_in_waiting_list.mean <- Reemplazar_NA("days_in_waiting_list")
Tabla_modificada$days_in_waiting_list <- NULL
names(Tabla_modificada)[32] <- "days_in_waiting_list"

# arrival_date_day_of_month
Tabla_modificada$arrival_date_day_of_month.mean <- Reemplazar_NA("arrival_date_day_of_month")
Tabla_modificada$arrival_date_day_of_month <- NULL
names(Tabla_modificada)[32] <- "arrival_date_day_of_month"
```

Ahora modificaremos los valores atípicos que encontramos inmersos en nuestro dataset.

Crearemos una función para buscar si en cada columna logramos encontrar datos atípicos.

```
# VERIFICAMOS VALORES ATÍPICOS
#-----
verificar_outliers <- function(df, df_names){
  for(variable in df_names) {
    outliers.values <- boxplot(df[,c(variable)], main = paste(variable,"con outliers"))$out
    outliers.values
  }
}
#-----
```

Crearemos una función la cual nos permitirá modificar esos datos atípicos encontrados.

```
# MODIFICAMOS VALORES ATÍPICOS
#-----

Corregir_outliers <- function(x, removeNA = TRUE){
  quantiles <- quantile(x, c(0.05, 0.95), na.rm = removeNA)
  x[x<quantiles[1]] <- mean(x, na.rm = removeNA)
  x[x>quantiles[2]] <- median(x, na.rm = removeNA)
  x
}
#-----
```

Por último, guardamos la data corregida.

```
# GUADAMOS DATOS PROCESADOS:
#-----
write.csv(Tabla_modificada, "hotel_bookings_miss_procesado.csv")
#-----
```

Visualizar datos:

En esta sección compararemos los datos iniciales que se nos brindó.

En esta primera sección veremos que se eliminaron los datos *NA* presentes en las columnas “adults”, “babies” y “children” del archivo original.

```
> verificar_NA(Tabla_Original, "adults")
[1] "adults"
var
  FALSE  TRUE
119378   12
> verificar_NA(Tabla_procesada, "adults")
[1] "adults"
var
  FALSE
119390

> verificar_NA(Tabla_procesada, "babies")
[1] "babies"
var
  FALSE
119390
> verificar_NA(Tabla_Original, "children")
[1] "children"
var
  FALSE  TRUE
119386   4
```

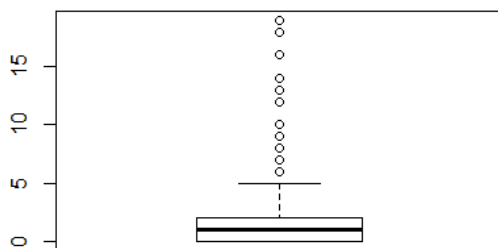
```

> verificar_NA(Tabla_Original, "children")
[1] "children"
var
  FALSE  TRUE
119386   4
> verificar_NA(Tabla_procesada, "children")
[1] "children"
var
  FALSE
119390

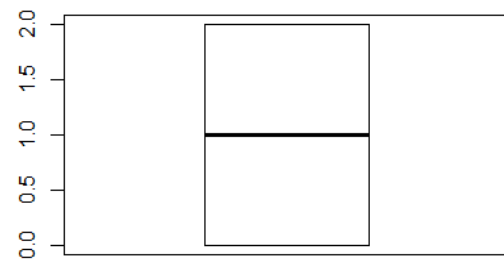
```

En esta segunda sección observamos la eliminación de datos atípicos de las columnas “stays_in_weekend_nights” y “stays_in_wee_nights” del archivo original.

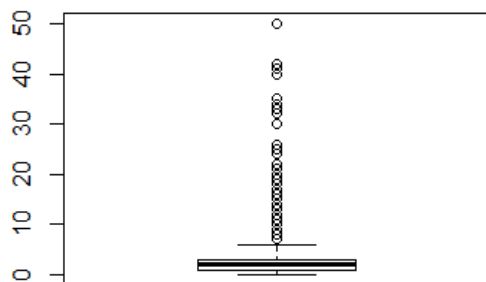
stays_in_weekend_nights con Outliers



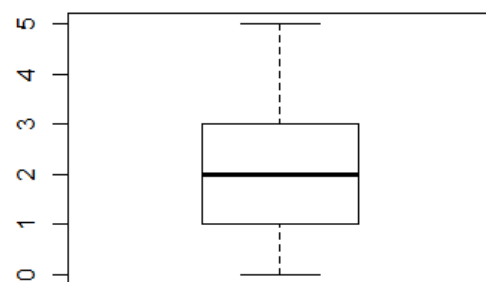
stays_in_weekend_nights sin outliers



stays_in_week_nights con Outliers



stays_in_week_nights sin outliers



4. Conclusiones preliminares

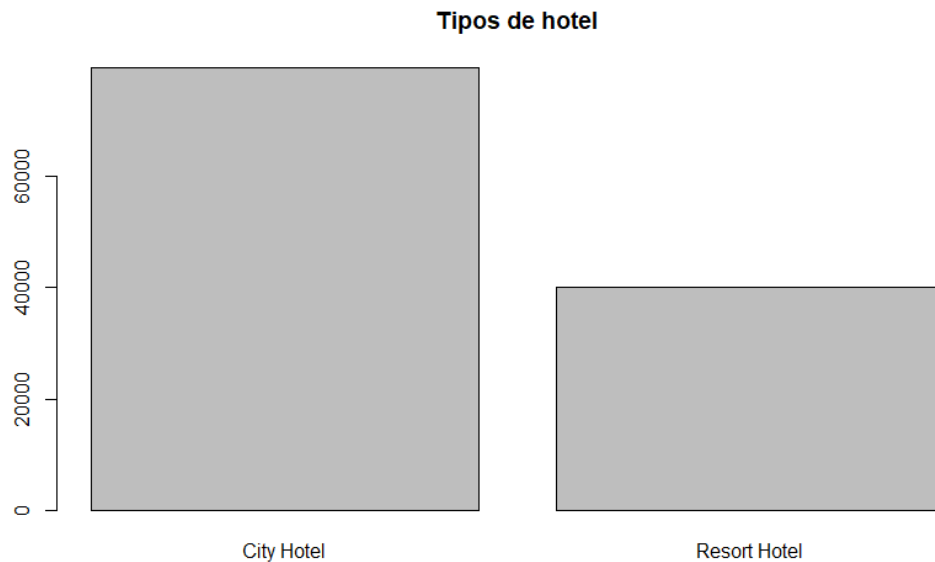
- a. ¿Cuántas reservas se realizan por tipo de hotel? o ¿Qué tipo de hotel prefiere la gente?

El tipo de hotel de preferencia vendría a ser el City Hotel. Como se puede ver en el gráfico de barras, la demanda por City Hotel es casi el doble que la de Resort Hotel.

```
library(dplyr)

hotel_data <- read.csv("Data/hotel_bookings_miss_processed.csv")
hotel_table <- table(hotel_data$ï..hotel)

hotel <- barplot(hotel_table,main="Tipos de hotel")
```



- b. ¿Está aumentando la demanda con el tiempo?

No, como se puede apreciar tan solo viendo la línea transversal a las barras del gráfico de barras, que básicamente define una tendencia de la demanda en el tiempo, esta está cayendo, por ende, está bajando la demanda en el tiempo.

```
library("dplyr")
library("lubridate")

hotel_data <- read.csv("Data/hotel_bookings_miss_processed.csv")

m_x <-
month(as.POSIXlt(hotel_data$reservation_status_date,format="%d/%m/%Y"))
```

```

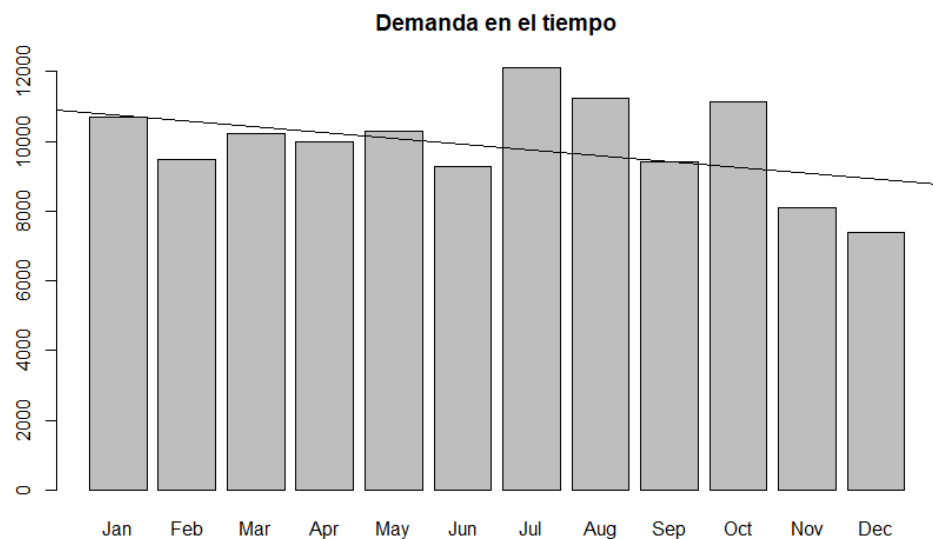
hotel_data$mon <- m_x

hotel_data.grp <- hotel_data %>%group_by(mon) %>% summarise(n = n())

regresion <- lm(hotel_data.grp$n ~ hotel_data.grp$mon,col="red")

barplot(hotel_data.grp$n, names.arg=month.abb, main="Demanda en el
tiempo")
abline(regresion)

```



- c. ¿Cuándo se producen las temporadas de reservas: alta, media y baja?
- La temporada alta se produce en los meses de Julio, Agosto, Octubre y Enero.
- La temporada media se produce en Febrero, Marzo, Abril, Mayo, Junio y Septiembre.
- La temporada baja se produce en Noviembre y Diciembre.

```

library("dplyr")
library("lubridate")

hotel_data <- read.csv("Data/hotel_bookings_miss_processed.csv")

m_x <- month(as.POSIXlt(hotel_data$reservation_status_date,
format="%d/%m/%Y"))
hotel_data$mon <- m_x

hotel_data.grp <- hotel_data %>% group_by(mon) %>% summarise(n =
n())

hotel_menor_d <- min(hotel_data.grp$n)

```

```

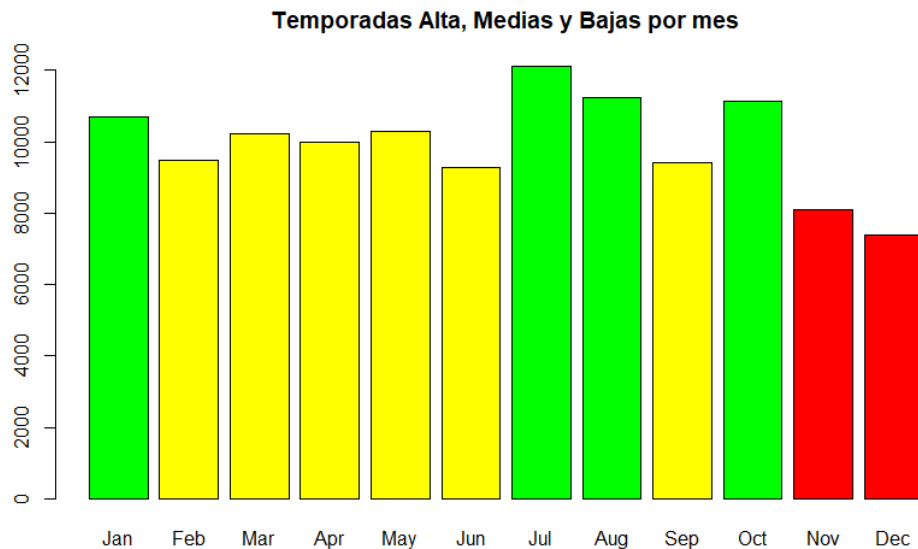
hotel_mayor_d <- max(hotel_data.grp$n)
d = (hotel_mayor_d - hotel_menor_d)/3

colors_menor <- (hotel_data.grp$n >= hotel_menor_d & hotel_data.grp$n <
hotel_menor_d + d)
colors_mayor <- (hotel_data.grp$n <= hotel_mayor_d & hotel_data.grp$n >
hotel_mayor_d - d)
colors_medio <- ( hotel_data.grp$n < hotel_mayor_d - d & hotel_data.grp$n
> hotel_menor_d + d)

colores <- ifelse(colors_mayor , "green" ,ifelse (colors_menor, "red"
,ifelse(colors_medio,"yellow", "gray")))

barplot(hotel_data.grp$n, names.arg=month.abb, col = colores,
main="Temporadas Alta, Medias y Bajas por mes")

```



d. ¿Cuándo es menor la demanda de reservas?

La demanda de reservas será menor en el mes de diciembre.

```

library("dplyr")
library("lubridate")
library("purrr")

hotel_data <- read.csv("Data/hotel_bookings_miss_processed.csv")

m_x <- month(as.POSIXlt(hotel_data$reservation_status_date,
format="%d/%m/%Y"))
hotel_data$mon <- m_x

```

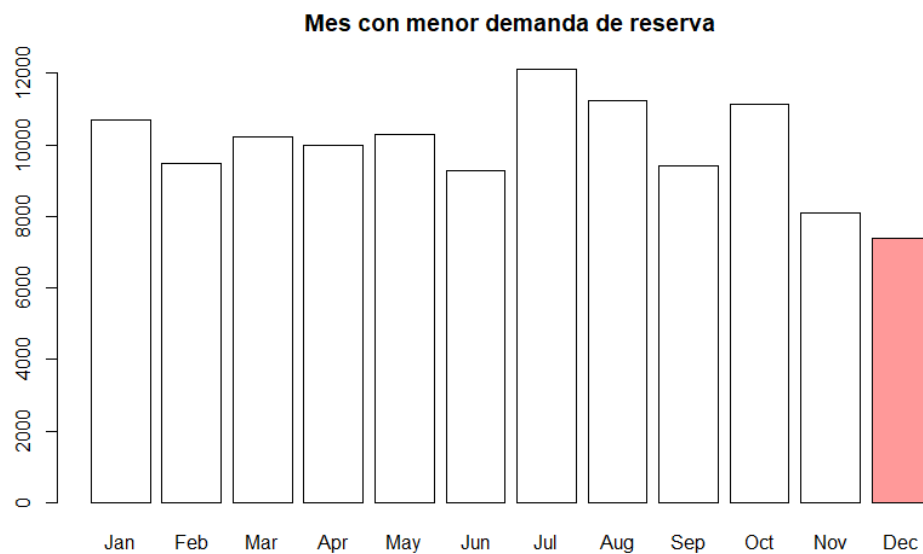
```

hotel_data.grp <- hotel_data %>% group_by(mon) %>% summarise(n =
n())
hotel_menor_d <- min(hotel_data.grp$n)

colors <- hotel_data.grp$n == hotel_menor_d
colors <- ifelse(colors, "#ff9999", "white")

barplot(hotel_data.grp$n, names.arg=month.abb, col=colors, main="Mes
con menor demanda de reserva")

```



- e. ¿Cuántas reservas incluyen niños y/o bebés?
Existen 4861 reservas que incluyen niños y/o bebés.

```

hotel_data <- read.csv("Data/hotel_bookings_miss_processed.csv")

hotel_data.babies <- hotel_data[hotel_data$babies>0,]
hotel_data.babies <- hotel_data.babies[is.na(hotel_data.babies$children) ==
0,]

hotel_data.children <- hotel_data[hotel_data$children>0,]
hotel_data.children <-
hotel_data.children[is.na(hotel_data.children$children) == 0,]

hotel_data.all <-
hotel_data[(hotel_data$children==0)&(hotel_data$babies==0),]
hotel_data.all <-
hotel_data.all[(is.na(hotel_data.all$children)|is.na(hotel_data.all$babies))==
0,]

```

```

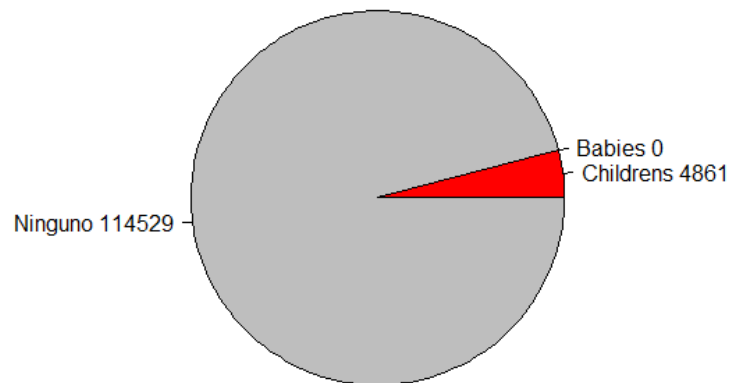
n_children <- nrow(hotel_data.children)
n_babies <- nrow(hotel_data.babies)
n_all <- nrow(hotel_data.all)

colors <- c("red", "yellow", "gray")
labels <- c("Childrens", "Babies", "Ninguno")
values <- c(n_children, n_babies, n_all)
etiquetas <- paste0(labels, " ", values)

pie(values, labels = etiquetas, col = colors, main="Reservas incluyen niños
y/o bebes")

```

Reservas incluyen niños y/o bebes



- f. ¿Es importante contar con espacios de estacionamiento?
- Para la mayoría de inquilinos no lo es. Se podría decir que es importante para una minoría, cosa que no es determinante, dado que el porcentaje del total no llega ni al 10%. Entonces, contar con espacios de estacionamiento no es realmente importante si hablamos en términos generales.

```

hotel_data <- read.csv("Data/hotel_bookings_miss_processed.csv")

hotel_data.tr <- hotel_data[hotel_data$required_car_parking_spaces==1,]
hotel_data.tr <-
hotel_data.tr[is.na(hotel_data.tr$required_car_parking_spaces) == 0,]

hotel_data.fl <- hotel_data[hotel_data$required_car_parking_spaces==0,]

```

```

hotel_data.fl <-
hotel_data.fl[is.na(hotel_data.fl$required_car_parking_spaces) == 0,]

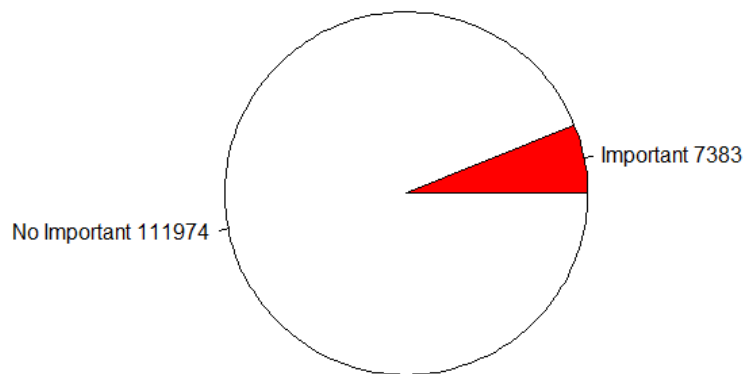
n_tr <- nrow(hotel_data.tr)
n_fl <- nrow(hotel_data.fl)

colors <- c("red", "white")
labels <- c("Important", "No Important")
values <- c(n_tr, n_fl)
etiquetas <- paste0(labels, " ", values)

pie(values, labels = etiquetas, col = colors,
main="Reservas con importaciona de espacios de estacionamiento")

```

Reservas con importaciona de espacios de estacionamiento



- g. ¿En qué meses del año se producen más cancelaciones de reservas?

En el mes de enero, en este se producen cerca de 6000 cancelaciones, el máximo de cancelaciones en un mes, en el año.

```

library("dplyr")
library("lubridate")

hotel_data <- read.csv("Data/hotel_bookings_miss_processed.csv")

hotel_data.sts <- hotel_data[hotel_data$is_canceled == 1,]

```

```

m_x <-
month(as.POSIXlt(hotel_data.sts$reservation_status_date,format="%d/%m/
%Y"))
hotel_data.sts$mon <- m_x

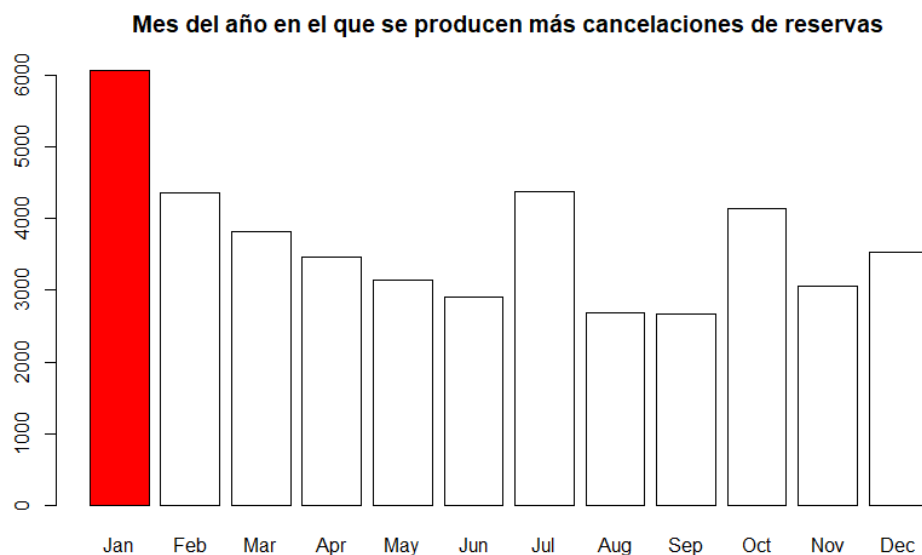
hotel_data.grp <- hotel_data.sts %>% group_by(mon) %>% summarise(n =
n())
hotel_data.grp$monName <- month.abb[hotel_data.grp$mon]

# Obtener Mes con mayores cancelamientos
hotel_data.maximo <- max(hotel_data.grp$n)
hotel_data.r <- hotel_data.grp[hotel_data.grp$n == hotel_data.maximo, ]

colors <- hotel_data.grp$n == hotel_data.maximo
colors <- ifelse(colors, "red", "white")

barplot(hotel_data.grp$n, names.arg=hotel_data.grp$monName, col=colors,
main="Mes del año en el que se producen más cancelaciones de reservas")

```



Repositorio GitHub

<https://github.com/DanielCGR/ea-2021-1-cc51.git>

5. Bibliografía

Antonio, N., de Almeida, A., & Nunes, L. (2019). Hotel booking demand datasets. Data in Brief, 22, 41–49. <https://doi.org/10.1016/j.dib.2018.11.126>