UNIVERSIDAD PERUANA DE CIENCIAS APLICADAS

FACULTAD DE INGENIERÍA

CARRERA PROFESIONAL DE Ciencias de la Computación

Asignatura:

CC50- Administración de la Información

Autores

Arana del Carpio, Sebastián Alonso

Mosqueira Chacón, César Manuel

Profesor

Reyes Silva, Patricia

2021-2

CONTENIDO

Control de código usando Github:

https://github.com/Cesarmosqueira/HotelBookingDemand-AnalysisCC50

Caso de análisis

El conjunto de datos se obtuvo de Kaggle; sin embargo, el dataset usado fue modificado, se le añadió ruido en los datos faltantes (NA) y datos atípicos (outliers). El dataset refleja dos conjuntos de datos de demanda hotelera. El primero es un hotel resort (H1) y el otro es un hotel urbano (H2), de los cuales ambos conjuntos de datos comparten la misma estructura, con 31 variables que describen las 40.060 observaciones de H1 y 79.330 observaciones de H2. Cada observación representa una reserva de hotel. Además, comprenden las reservas que deben llegar entre el 1 de julio de 2015 y el 31 de agosto de 2017, incluidas las reservas que llegaron efectivamente y las reservas que se cancelaron. La información del dataset es verídica, por lo cual se optó por eliminar la identificación del hotel o del cliente.

Debido a la escasez de datos comerciales reales para fines educativos, este dataset ayudará mucho para la educación en gestión de datos y minería de datos. Por otro lado, se tiene que esta investigación ayudará a los clientes a saber más sobre cuándo ir a estos hoteles debido a que tienen más información, por lo cual pueden tomar mejores decisiones.

Conjunto de datos

A continuación se mostrará todas las variables que contiene nuestro dataset:

#Item	Variable	Descripción
1	hotel	Variable que muestra el nombre del hotel, que en este conjunto de datos serán dos: Resort Hotel y City Hotel.
2	is_canceled	Indica si la reserva fue cancelada en el hotel
3	lead_time	Diferencia en días entre fecha de reserva y entrada
4	arrival_date_year	El año de la fecha de entrada
5	arrival_date_month	El mes de la fecha de entrada
6	arrival_date_week_number	El número de la semana del año de la fecha de la entrada
7	arrival_date_day_of_month	El día de la fecha de entrada
8	stays_in_weekend_nights	El número de noches de fin de semana que se reservaron
9	stays_in_week_nights	El número de noches de día de semana que se

		reservaron
10	adults	Número de adultos en la reserva
11	children	Número de niños en la reserva
12	babies	Número de bebés en la reserva
13	meal	Número de comidas en la reserva
14	country	País de origen de los clientes
15	market_segment	La designación del mercado. Puede ser por agentes de viajes, operadores turísticos u otros.
16	distribution_channel	Muestra canal de reserva. Puede ser por agentes de viajes, operadores turísticos o otro forma.
17	is_repeated_guest	Muestra si la reserva está hecha por un cliente repetido
18	previous_cancellations	Número de reservas canceladas anteriormente por un cliente antes de la reserva actual
19	previous_bookings_not_canceled	Número de reservas no canceladas anteriormente por un cliente antes de la reserva actual
20	reserved_room_type	Tipo de habitación reservada
21	assigned_room_type	Código de asignación de las habitaciones
22	booking_changes	Número de cambios hechos en el momento de la reserva
23	deposit_type	Depósito de la reserva
24	agent	Nombre del agente de viajes que realizó la reserva
25	company	Nombre de la compañía que realizó la reserva
26	days_in_waiting_list	Número de días que espera la confirmación de su reserva
27	customer_type	Tipo de cliente que hace la reserva
28	adr	Tarifa promedio diaria de una reserva
29	required_car_parking_spaces	Número de espacios de estacionamientos puestas en la reserva
30	total_of_special_requests	Número de pedidos especiales hechos por los clientes en la reserva

31	reservation_status	El estado de la reserva hecha por el cliente
32	reservation_status_date	El día que se actualizó el último estado de reserva del cliente

Análisis de datos

I. CARGAR DATOS

Para adquirir los datos usamos la función read.csv():
La siguiente instrucción indica la creación de una variable, la cual almacenará el dataset

```
na_values = c("", "Undefined", "NA")
# read csv ignoring {na_values}
hotel <- read.csv(path, na.strings=na_values)
```

II. INSPECCIONAR DATOS

> View(hotel)
>names(hotel)
>str(hotel)
>summary(hotel)

III. PREPROCESAR DATOS

En primer lugar, identificamos filas que no necesitamos

```
> sum(is.na(hotel))
[1] 1327
```

Luego, ignoramos las filas que no necesitamos

```
hotel = hotel[!is.na(hotel)]
```

Resultado

```
> sum(is.na(hotel))
[1] 0
```

Se eliminaron 1319 filas

```
# new dim = 118071 32
dim(hotel)
```

Combinar fechas de llegada

Originalmente:

arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of_month	st
2015	July	27	1	

Se combinaron las columnas en un arreglo

```
# Merge dates
# Retreieve info from columns
# Year : Integer
Year<-hotel$arrival_date_year
# Month = 'January' regular expresion -> 'Jan'
Month <- sub('(^.{3}).*','\\1', hotel$arrival_date_month)
# Day : Integer
Day<-hotel$arrival_date_day_of_month
dates <- paste(Year,Month,Day,sep="-")
```

```
> dates
   [1] "2015-Jul-1"  "2015-Jul-1"  "2015-Jul-1"  "2015-Jul-1"  "2015-Jul-1"
   [6] "2015-Jul-1"  "2015-Jul-1"  "2015-Jul-1"  "2015-Jul-1"  "2015-Jul-1"
  [11] "2015-Jul-1"  "2015-Jul-1"  "2015-Jul-1"  "2015-Jul-1"  "2015-Jul-1"
  [16] "2015-Jul-1"  "2015-Jul-1"  "2015-Jul-1"  "2015-Jul-1"  "2015-Jul-1"
  [21] "2015-Jul-1"  "2015-Jul-1"  "2015-Jul-1"  "2015-Jul-1"  "2015-Jul-1"
  [26] "2015-Jul-1"  "2015-Jul-1"  "2015-Jul-1"  "2015-Jul-1"  "2015-Jul-1"
  [31] "2015-Jul-1"  "2015-Jul-1"  "2015-Jul-1"  "2015-Jul-2"  "2015-Jul-2"
  [36] "2015-Jul-2"  "2015-Jul-2"  "2015-Jul-2"  "2015-Jul-2"  "2015-Jul-2"
  [41] "2015-Jul-2"  "2015-Jul-2"  "2015-Jul-2"  "2015-Jul-2"  "2015-Jul-2"
  [46] "2015-Jul-2"  "2015-Jul-2"  "2015-Jul-2"  "2015-Jul-2"  "2015-Jul-2"
  [51] "2015-Jul-2"  "2015-Jul-2"  "2015-Jul-2"  "2015-Jul-2"  "2015-Jul-2"
  [56] "2015-Jul-2"  "2015-Jul-2"  "2015-Jul-2"  "2015-Jul-2"  "2015-Jul-2"
  [61] "2015-Jul-2"  "2015-Jul-2"  "2015-Jul-2"  "2015-Jul-2"  "2015-Jul-2"
  [66] "2015-Jul-3"  "2015-Jul-3"  "2015-Jul-3"  "2015-Jul-3"  "2015-Jul-3"
  [71] "2015-Jul-3"  "2015-Jul-3"  "2015-Jul-3"  "2015-Jul-3"  "2015-Jul-3"
  [76] "2015-Jul-3"  "2015-Jul-3"  "2015-Jul-3"  "2015-Jul-3"  "2015-Jul-3"
  [81] "2015-Jul-3"  "2015-Jul-3"  "2015-Jul-3"  "2015-Jul-3"  "2015-Jul-3"
  [86] "2015-Jul-3"  "2015-Jul-3"  "2015-Jul-3"  "2015-Jul-3"  "2015-Jul-3"
```

Y se agregó el arreglo al df

```
# put it into the existing df
hotel$arrival_date = dates
# parse it
hotel %>% group_by(hotel) %>%
  mutate(arrival_date=as.Date(arrival_date, format = "%Y-%m-%d"))
```

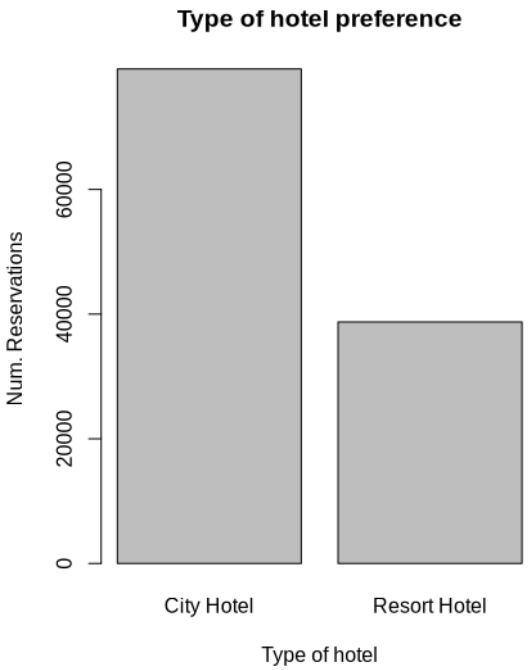
Guardamos el nuevo dataset

```
write.csv(hotel, "clean-hotel-bookings.csv")
```

Conclusiones preliminares

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

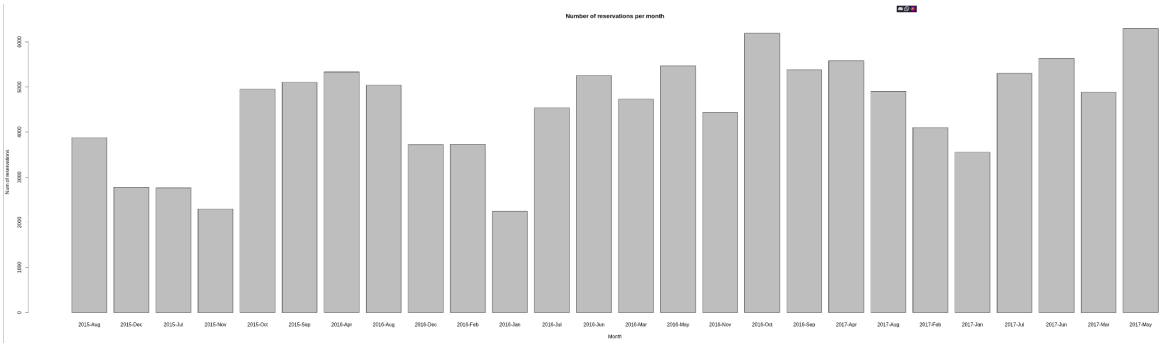
```
> table(hotel['hotel'])

  City Hotel Resort Hotel
       79326        38745
```



Se puede apreciar que las personas reservan con mucha más frecuencia un hotel regular de ciudad a resort. Esto puede ser debido a que, cuando la gente va a un resort, es durante vacaciones, que es un porcentaje pequeño del año. A eso agregarle que hay menos resorts que hoteles en la ciudad.

b. ¿Está aumentando la demanda con el tiempo?

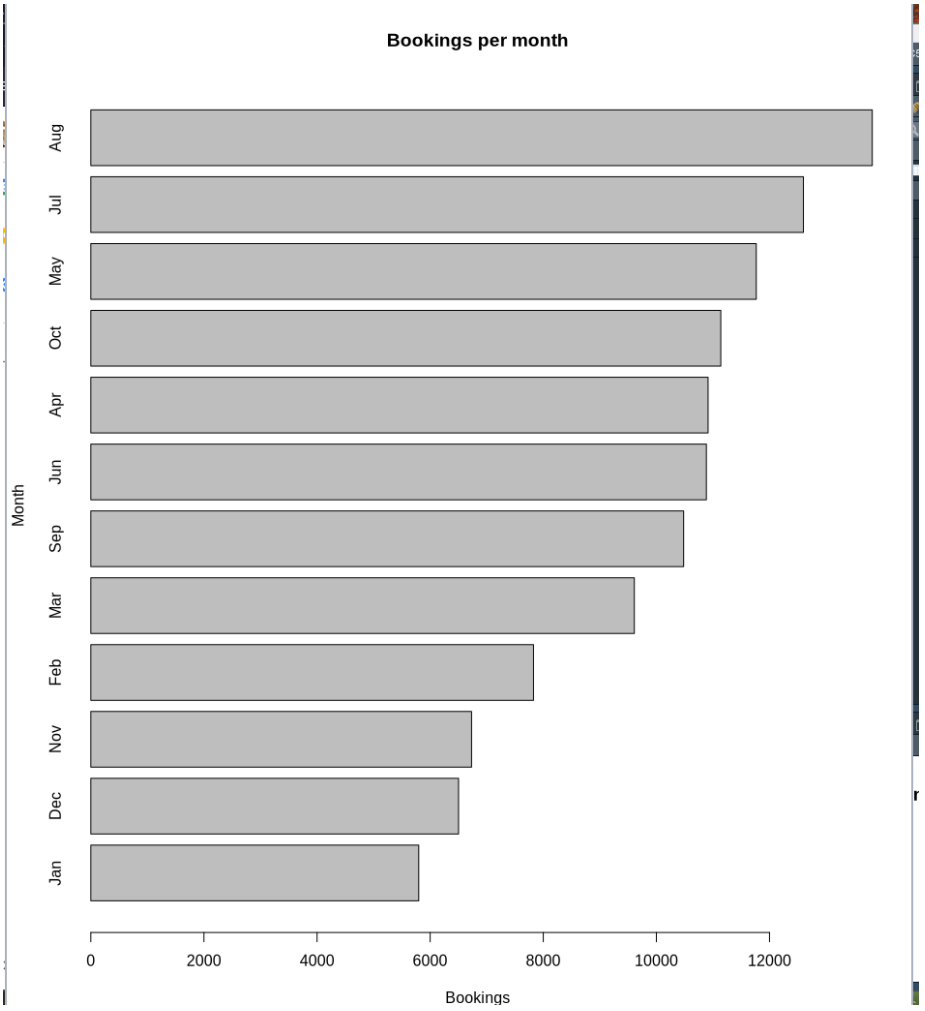
```
# array of %Y-%m of original dates
#and table it
dates <- table(sub('(^.{8}).*','\\1', hotel$arrival_date))
barplot(dates, main="Number of reservations per month",
        xlab="Month", ylab="Num of reservations")
```



Si comparamos la barra de cada mes con la del año siguiente, nos daremos cuenta que efectivamente **la demanda está subiendo**.

c. ¿Cuándo se producen las temporadas de reservas: alta, media y baja?

```
# c. ¿Cuándo se producen las temporadas de reservas: alta, media y baja?
mnts = substring(hotel$arrival_date,first=6,last=8)
barplot(table(mnts), horiz=TRUE, main="Bookings per month",
        xlab="Bookings", ylab="Month")
```



Según el gráfico se identifican 3 picos. **El alto** se encuentra entre los meses de Julio-Agosto (Verano en el hemisferio norte). **El intermedio** se encuentra entre los meses de media estación (Primavera y otoño). Mientras que las **temporada baja** sería en enero, diciembre y noviembre (Invierno en el hemisferio norte).

d. ¿Cuándo es menor la demanda de reservas?

Como se indica en el cuadro anterior, la menor demanda de reservas ocurre en los meses de **invierno** en el hemisferio norte.

e. ¿Cuántas reservas incluyen niños y/o bebes?

```
unblessed <- nrow(subset(hotel, children == 0 & babies == 0 ))
blessed <- nrow(subset(hotel, children > 0 | babies > 0 ))
```

```
> blessed
[1] 9261
> unblessed
[1] 108810
>
```

9261 Reservas se hicieron con niños o bebés, mientras que 108810 se hicieron sin niños ni bebés.

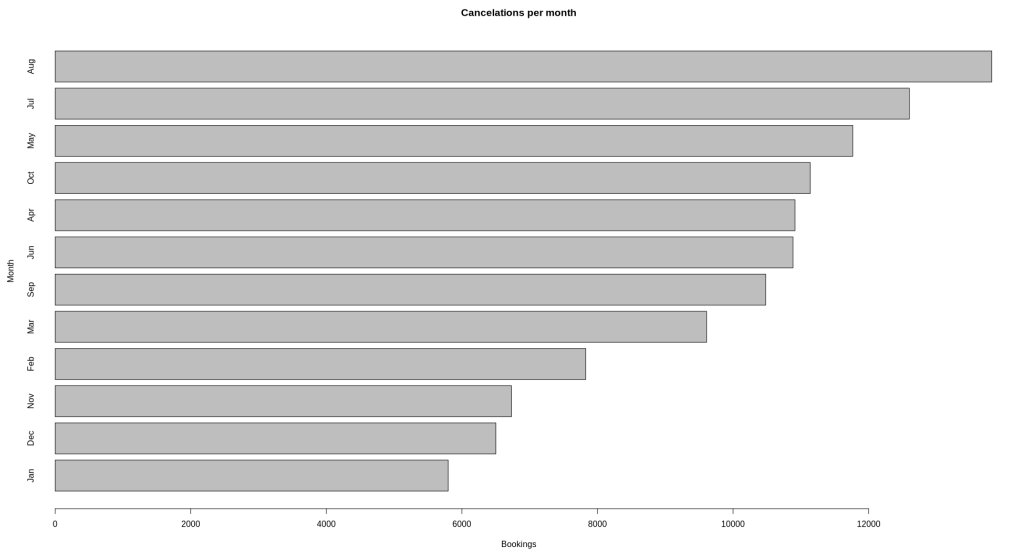
f. ¿Es importante contar con espacios de estacionamiento?

```
parking <- nrow(subset(hotel, required_car_parking_spaces > 0))
no_parking <- nrow(subset(hotel, required_car_parking_spaces == 0))
```

```
> no_parking <-
> parking
[1] 7366
> no_parking
[1] 110705
>
```

7366 reservas necesitaban estacionamiento mientras que 110705 no necesitaban. Se podría decir que no es indispensable, ya que solo el 6% de reservas requerían de estacionamiento.

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



En el gráfico se puede apreciar que el mes de **Agosto-Julio** es en el que más se cancelan las reservas. También es el mes en el que ocurren más reservas.