

Universidad Peruana de Ciencias Aplicadas

UPC



TA1

Administración De La Información

Profesora: Patricia Daniela Reyes Silva

Ciclo: 2021 - 2

Integrantes:

- ❖ Renzo Paredes Villagra - u20191b335
- ❖ Sergio Andres Flores Ñahuis - u201915474

1. Caso de análisis

Para este trabajo, se va analizar el conjunto de datos llamado Hotel booking demand. Este conjunto de datos proviene del artículo publicado en el diario Data in Brief en febrero de 2019 y realizado por los autores Nuno Antonio, Ana de Almeida y Luis Nunes. Este artículo contiene información perteneciente a dos hoteles: un hotel resort y un hotel urbano. Asimismo, el conjunto presenta 40,060 observaciones del hotel tipo resort y 79,330 observaciones del hotel urbano. Todas estas están descritas a través de 31 variables, las cuales incluyen fecha de llegada, cantidad de adultos, cantidad de niños, tipo de comida, canal de distribución, país, tipo de depósito, cambios en la reserva, tipo de cliente y entre otros.

El propósito de este artículo es que estos conjuntos de datos recopilados pueden ser usados en investigaciones dentro de los ámbitos científico y educacional, así como en los campos de machine learning y data mining.

2. Conjunto de datos

Item	Variable	Descripción
1	hotel	nombre del hotel
2	is_canceled	reserva hecha fue cancelada
3	lead_time	días de diferencia entre la fecha de reserva y la de entrada
4	arrival_date_year	año de la fecha de entrada
5	arrival_date_month	mes de la fecha de entrada
6	arrival_date_week_number	número de la semana de la fecha de entrada
7	arrival_date_day_of_month	día del mes de la fecha de entrada

8	stays_in_weekend_nights	número de noches de fin de semana que se reservaron
9	stays_in_week_nights	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	tipo de comida
14	country	país de procedencia
15	market_segment	segmento del mercado para la reserva
16	distribution_channel	canal de distribución de la reserva
17	is_repeated_guest	determina si el huésped ya se alojó en el hotel antes
18	previous_cancellations	cantidad de cancelaciones previas
19	previous_bookings_not_cancelled	cantidad de cancelaciones previas que no han sido canceladas
20	reserved_room_type	tipo de habitación reservada
21	assigned_room_type	tipo de habitación asignada
22	booking_changes	cantidad de cambios en la reserva

23	deposit_type	tipo de depósito en la reserva
24	agent	agente de viajes que realizó la reserva
25	company	compañía que realizó la reserva
26	days_in_waiting_list	cantidad de días en la lista de espera
27	customer_type	tipo de cliente
28	adr	tarifa diaria promedio de una reserva
29	required_car_parking_spaces	cantidad de espacios requeridos de aparcamiento de autos
30	total_of_special_requests	cantidad total de solicitudes especiales
31	reservation_status	estado en que se encuentra la reserva
32	reservation_status_date	fecha de la actualización de estado

3. Análisis exploratorio de datos

i. CARGAR DATOS

Hemos adquirido nuestros datos usando esta instrucción:

```
> booking_data <- read.csv("../hotel_bookings_miss.csv", header = TRUE, stringsAsFactors = FALSE, sep = ",")
```

Donde en la función `read.csv` le mandamos primero la dirección del archivo en formato `.csv`, luego al `header` lo igualamos a `TRUE` lo que indicará que el archivo tiene encabezados, también al `stringsAsFactors` lo igualamos a `FALSE` lo que hará que los valores `NA` los cargue como objeto `NA` y no como `string`, y por último la

sep lo igualamos al string coma lo que indicará que los datos los va a separar por coma.

ii. INSPECCIONAR DATOS

Ya que se quiere conocer de forma más detallada las variables que se encuentran en nuestro dataset, emplearemos las siguientes instrucciones:

> colnames(booking_data)

```
[1] "hotel" "is_canceled" "lead_time"
[4] "arrival_date_year" "arrival_date_month" "arrival_date_week_number"
[7] "arrival_date_day_of_month" "stays_in_weekend_nights" "stays_in_week_nights"
[10] "adults" "children" "babies"
[13] "meal" "country" "market_segment"
[16] "distribution_channel" "is_repeated_guest" "previous_cancellations"
[19] "previous_bookings_not_canceled" "reserved_room_type" "assigned_room_type"
[22] "booking_changes" "deposit_type" "agent"
[25] "company" "days_in_waiting_list" "customer_type"
[28] "adr" "required_car_parking_spaces" "total_of_special_requests"
[31] "reservation_status" "reservation_status_date"
```

Aquí se puede apreciar el número total de variables y sus respectivos nombres, que van a determinar las columnas en nuestro conjunto de datos.

>str(booking_data)

```
'data.frame': 119390 obs. of 32 variables:
 $ hotel: chr "Resort Hotel" "Resort Hotel" "Resort Hotel" "Resort Hotel" ...
 $ is_canceled: int 0 0 0 0 0 0 0 0 1 1 ...
 $ lead_time: int 342 737 7 13 14 14 0 9 85 75 ...
 $ arrival_date_year: int 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 ...
 $ arrival_date_month: chr "July" "July" "July" "July" ...
 $ arrival_date_week_number: int 27 27 27 27 27 27 27 27 27 27 ...
 $ arrival_date_day_of_month: int 1 1 1 1 1 1 1 1 1 1 ...
 $ stays_in_weekend_nights: int NA 0 0 0 0 0 0 0 0 0 ...
 $ stays_in_week_nights: int 0 0 1 1 2 2 2 2 3 3 ...
 $ adults: int 2 2 1 1 2 2 2 2 2 2 ...
 $ children: int 0 0 0 0 0 0 0 0 0 0 ...
 $ babies: int 0 0 0 0 0 0 0 0 0 0 ...
 $ meal: chr "BB" "BB" "BB" "BB" ...
 $ country: chr "PRT" "PRT" "GBR" "GBR" ...
 $ market_segment: chr "Direct" "Direct" "Direct" "Corporate" ...
 $ distribution_channel: chr "Direct" "Direct" "Direct" "Corporate" ...
 $ is_repeated_guest: int 0 0 0 0 0 0 0 0 0 0 ...
 $ previous_cancellations: int 0 0 0 0 0 0 0 0 0 0 ...
 $ previous_bookings_not_canceled: int 0 0 0 0 0 0 0 0 0 0 ...
 $ reserved_room_type: chr "C" "C" "A" "A" ...
 $ assigned_room_type: chr "C" "C" "C" "A" ...
 $ booking_changes: int 3 4 0 0 0 0 0 0 0 0 ...
 $ deposit_type: chr "No Deposit" "No Deposit" "No Deposit" "No Deposit" ...
 $ agent: chr "NULL" "NULL" "NULL" "304" ...
 $ company: chr "NULL" "NULL" "NULL" "NULL" ...
 $ days_in_waiting_list: int 0 0 0 0 0 0 0 0 0 0 ...
 $ customer_type: chr "Transient" "Transient" "Transient" "Transient" ...
 $ adr: num 0 0 75 75 98 ...
 $ required_car_parking_spaces: int 0 0 0 0 0 0 0 0 0 0 ...
 $ total_of_special_requests: int 0 0 0 0 1 1 0 1 1 0 ...
 $ reservation_status: chr "Check-Out" "Check-Out" "Check-Out" "Check-Out" ...
 $ reservation_status_date: chr "7/1/2015" "7/1/2015" "7/2/2015" "7/2/2015" ...
```

Con esto, se observan el número total de observaciones en el dataset y los tipos de dato de cada una de las variables, así como algunos de los valores que contiene cada columna.

> summary(booking_data)

```

hotel          is_canceled    lead_time    arrival_date_year arrival_date_month arrival_date_week_number
Length:119390   Min.   :0.0000   Min.   : 0   Min.   :2015   Length:119390   Min.   : 1.00
Class :character 1st Qu.:0.0000 1st Qu.: 18 1st Qu.:2016   Class :character 1st Qu.:16.00
Mode  :character Median :0.0000 Median : 69 Median :2016   Mode  :character Median :28.00
                  Mean  :0.3704 Mean  :104 Mean  :2016   Mean  :27.16
                  3rd Qu.:1.0000 3rd Qu.:160 3rd Qu.:2017   3rd Qu.:38.00
                  Max.   :1.0000 Max.   :737 Max.   :2017   Max.   :53.00
                  NA's   :21   NA's   :6   NA's   :25
arrival_date_day_of_month stays_in_weekend_nights stays_in_week_nights adults children
Min.   : 1.0           Min.   : 0.0000   Min.   : 0.0   Min.   : 0.000   Min.   : 0.0000
1st Qu.: 8.0           1st Qu.: 0.0000   1st Qu.: 1.0   1st Qu.: 2.000   1st Qu.: 0.0000
Median :16.0           Median : 1.0000   Median : 2.0   Median : 2.000   Median : 0.0000
Mean   :15.8           Mean   : 0.9275   Mean   : 2.5   Mean   : 1.856   Mean   : 0.1039
3rd Qu.:23.0           3rd Qu.: 2.0000   3rd Qu.: 3.0   3rd Qu.: 2.000   3rd Qu.: 0.0000
Max.   :31.0           Max.   :19.0000   Max.   :50.0   Max.   :55.000   Max.   :10.0000
NA's   :7             NA's   :25   NA's   :12   NA's   :12   NA's   :4
babies          meal          country    market_segment distribution_channel is_repeated_guest
Min.   : 0.00000   Length:119390   Length:119390   Length:119390   Length:119390   Min.   :0.00000
1st Qu.: 0.00000   Class :character Class :character Class :character Class :character 1st Qu.:0.00000
Median : 0.00000   Mode  :character Mode  :character Mode  :character Mode  :character Median :0.00000
Mean   : 0.00795                                     Mean :0.03191
3rd Qu.: 0.00000                                     3rd Qu.:0.00000
Max.   :10.00000                                     Max.   :1.00000
NA's   :32
previous_cancellations previous_bookings_not_canceled reserved_room_type assigned_room_type booking_changes
Min.   : 0.00000   Min.   : 0.0000   Length:119390   Length:119390   Min.   : 0.0000
1st Qu.: 0.00000   1st Qu.: 0.0000   Class :character Class :character 1st Qu.: 0.0000
Median : 0.00000   Median : 0.0000   Mode  :character Mode  :character Median : 0.0000
Mean   : 0.08712   Mean   : 0.1371                                     Mean  : 0.2211
3rd Qu.: 0.00000   3rd Qu.: 0.0000                                     3rd Qu.: 0.0000
Max.   :26.00000   Max.   :72.0000                                     Max.   :21.0000

```

Esta función otorga algunos de los datos más relevantes para cada variable en el dataset. Estos valores pueden variar dependiendo del tipo de dato. En el caso de “hotel”, se obtienen la cantidad de observaciones, tipo de clase y la moda. Sin embargo, en el caso de “adults”, conseguimos el valor mínimo, la media, mediana, cantidad de valores NA, entre otros.

iii. PRE-PROCESAR DATOS

Para que se nos complique el trabajo y reducir los errores, primero identificamos si hay valores NA en la Data usando la siguiente instrucción:

```

> valorNA <- function(x){
>   sum=0
>   for(i in 1:ncol(x))
>   {
>     cat(colnames(x[i]), ", Valores NA:", colSums(is.na(x[i])), "\n")
>   }
> }
> valorNA(booking_data)

```

Donde le creamos una función llamada valorNA que, al recibir el valor de la Data, muestra cuántos valores NA tiene cada columna del Data.

```
hotel , valores NA: 0
is_canceled , valores NA: 0
lead_time , valores NA: 21
arrival_date_year , valores NA: 6
arrival_date_month , valores NA: 0
arrival_date_week_number , valores NA: 25
arrival_date_day_of_month , valores NA: 7
stays_in_weekend_nights , valores NA: 25
stays_in_week_nights , valores NA: 12
adults , valores NA: 12
children , valores NA: 4
babies , valores NA: 32
meal , valores NA: 0
country , valores NA: 0
market_segment , valores NA: 0
distribution_channel , valores NA: 0
is_repeated_guest , valores NA: 0
previous_cancellations , valores NA: 0
previous_bookings_not_canceled , valores NA: 0
reserved_room_type , valores NA: 0
assigned_room_type , valores NA: 0
booking_changes , valores NA: 0
deposit_type , valores NA: 0
agent , valores NA: 0
company , valores NA: 0
days_in_waiting_list , valores NA: 7
customer_type , valores NA: 0
adr , valores NA: 0
required_car_parking_spaces , valores NA: 0
total_of_special_requests , valores NA: 0
reservation_status , valores NA: 0
reservation_status_date , valores NA: 0
```

Luego de identificar los valores NA, creamos una función llamada moda, luego creamos otro data pero quitando todos los valores NA para que no halla tanto errores y de más exacto el valor de la moda, después creamos una función llamada replace_na donde recibe el Data original y la Data sin valores NA y por último retornará una Data donde todos sus valores NA fueron reemplazados por la moda y donde este resultado lo recibe un nuevo Data llamado booking_data_na.

```
> moda <- function(v){
>   uniqv <- unique(v)
>   uniqv[which.max(tabulate(match(v,uniqv)))]
> }
>
> booking_limpio= na.omit(booking_data)
>
> replace_na <- function(x, x_limpio){
>   for(i in 1:ncol(x))
```

```

> {
>   if (colSums(is.na(x[i])) > 0)
>   {
>     moda_col = moda(x_limpio[,i])
>     x[i][is.na(x[i])] = moda_col
>   }
> }
> return(x)
> }
>
> booking_data_na = replace_na(booking_data, booking_limpio)

```

```

> valorNA(booking_data_na)
hotel , Valores NA: 0
is_canceled , Valores NA: 0
lead_time , Valores NA: 0
arrival_date_year , Valores NA: 0
arrival_date_month , Valores NA: 0
arrival_date_week_number , Valores NA: 0
arrival_date_day_of_month , Valores NA: 0
stays_in_weekend_nights , Valores NA: 0
stays_in_week_nights , Valores NA: 0
adults , Valores NA: 0
children , Valores NA: 0
babies , Valores NA: 0
meal , Valores NA: 0
country , Valores NA: 0
market_segment , Valores NA: 0
distribution_channel , Valores NA: 0
is_repeated_guest , Valores NA: 0
previous_cancellations , Valores NA: 0
previous_bookings_not_canceled , Valores NA: 0
reserved_room_type , Valores NA: 0
assigned_room_type , Valores NA: 0
booking_changes , Valores NA: 0
deposit_type , Valores NA: 0
agent , Valores NA: 0
company , Valores NA: 0
days_in_waiting_list , Valores NA: 0
customer_type , Valores NA: 0
adr , Valores NA: 0
required_car_parking_spaces , Valores NA: 0
total_of_special_requests , Valores NA: 0
reservation_status , Valores NA: 0
reservation_status_date , Valores NA: 0

```

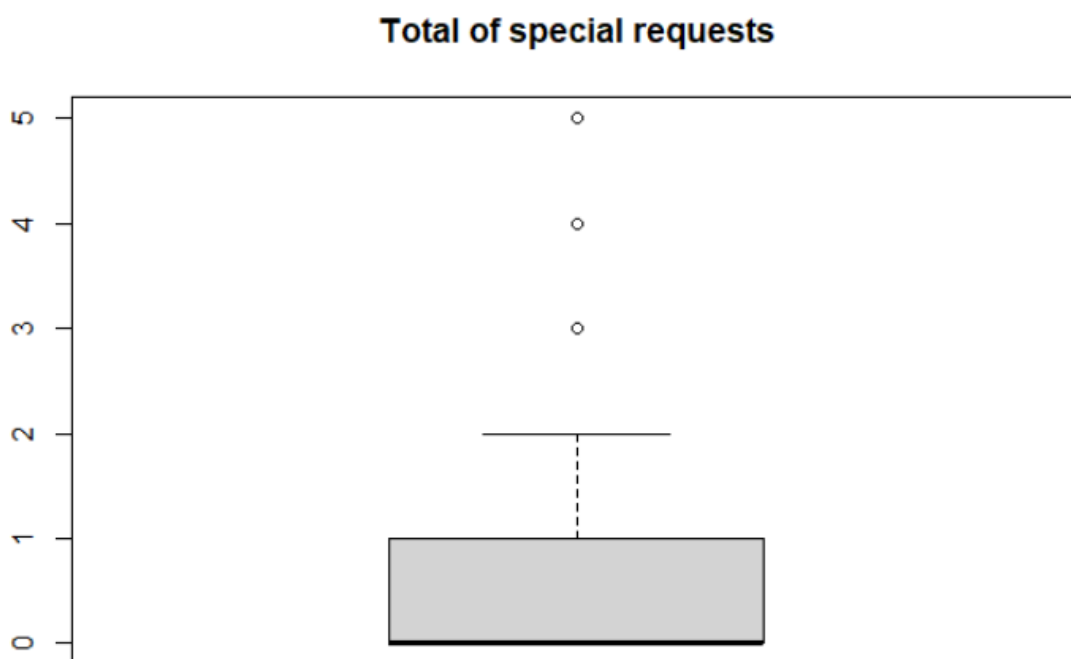
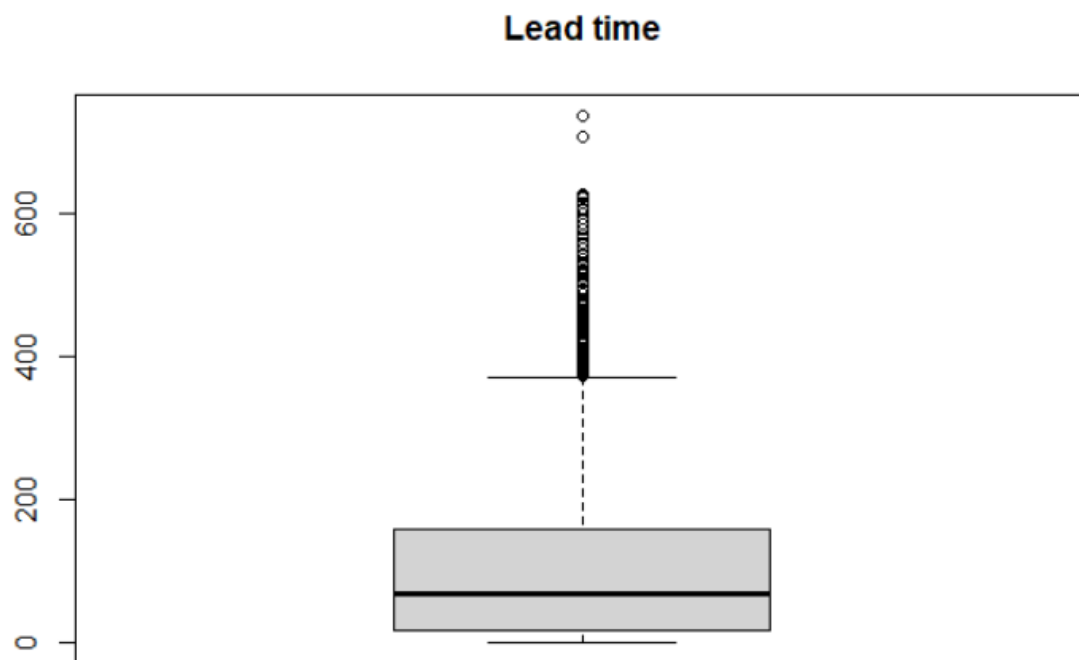
Para detectar los valores atípicos o outliers en cada una de las variables del dataset, utilizaremos una función boxplot que nos permitirá visualizar si existen en una columna específica a través del diagrama de caja.

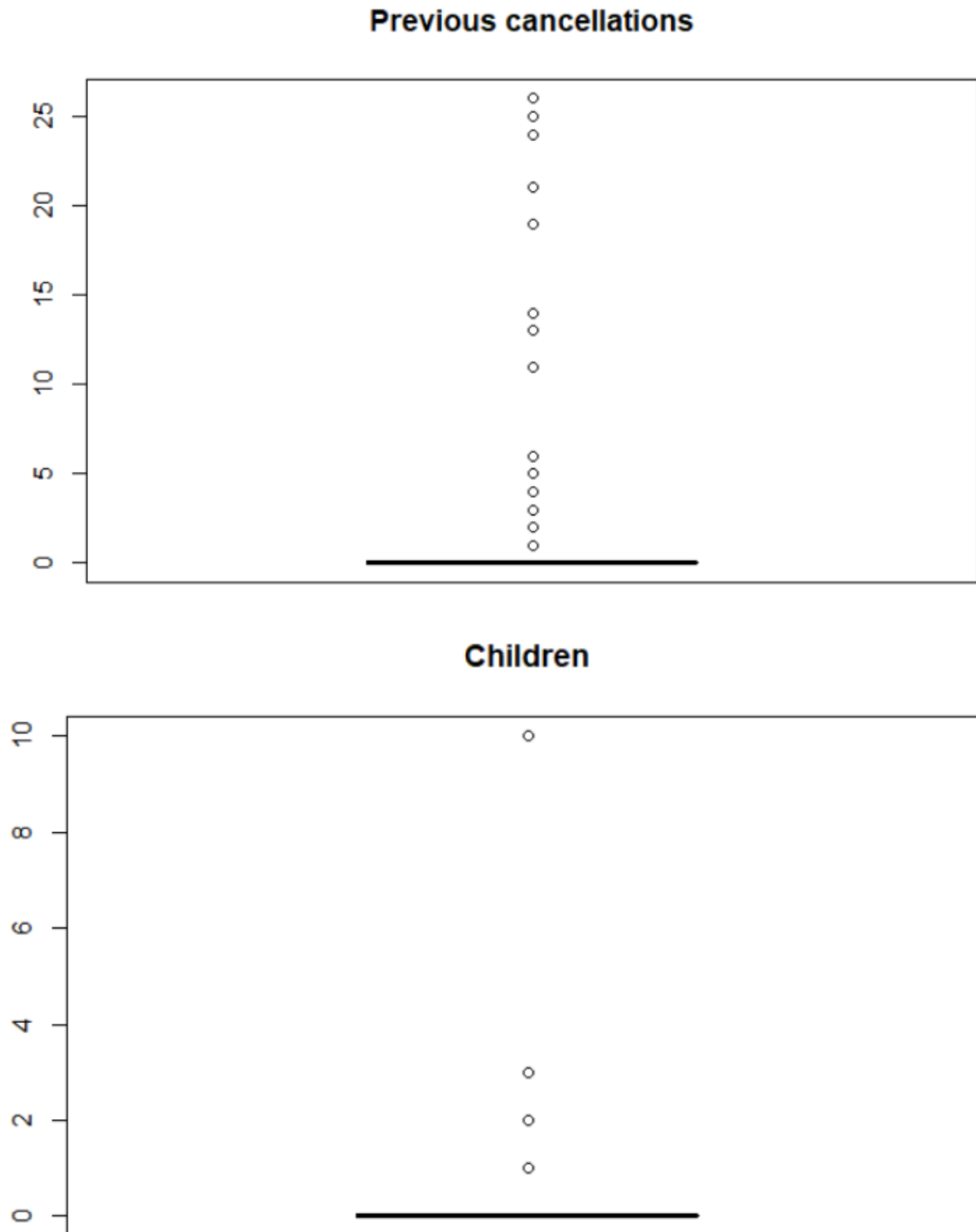

```
>boxplot(booking_data_na$lead_time, main="Lead time")
```

```
>boxplot(booking_data_na$total_of_special_requests, main="Total of special requests")
```

```
>boxplot(booking_data_na$previous_cancellations, main="Previous cancellations")
```

```
>boxplot(booking_data_na$children, main="Children")
```

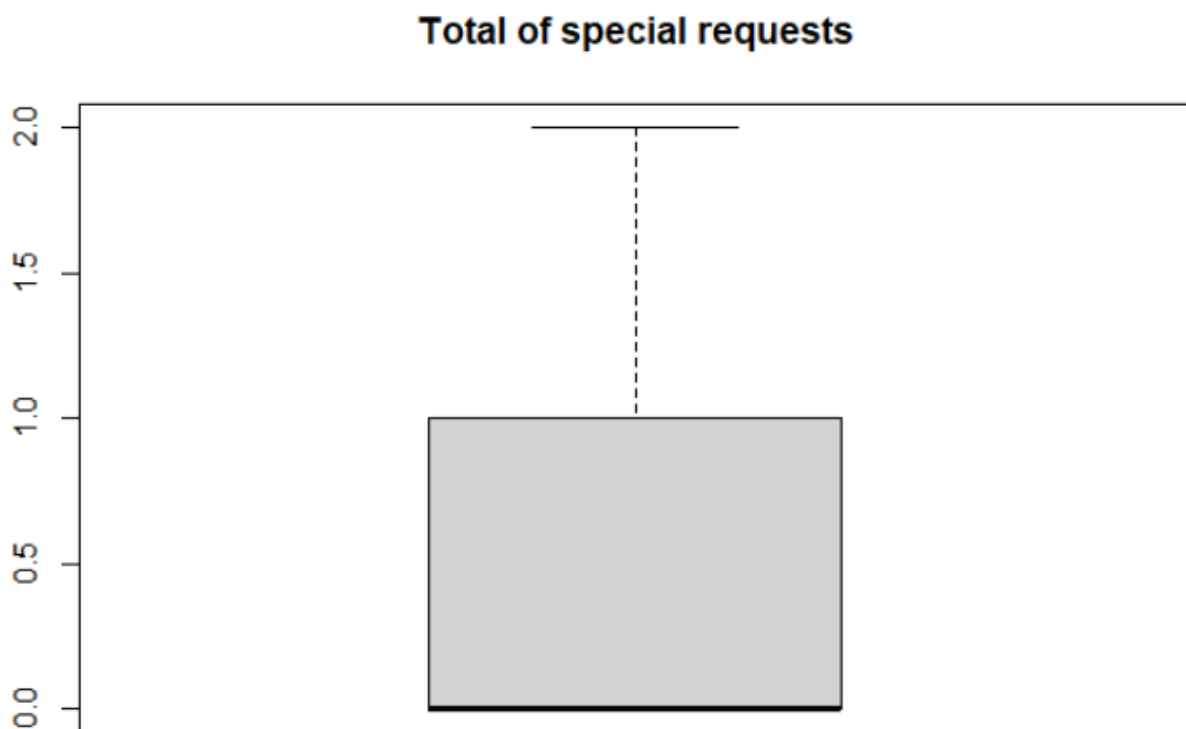


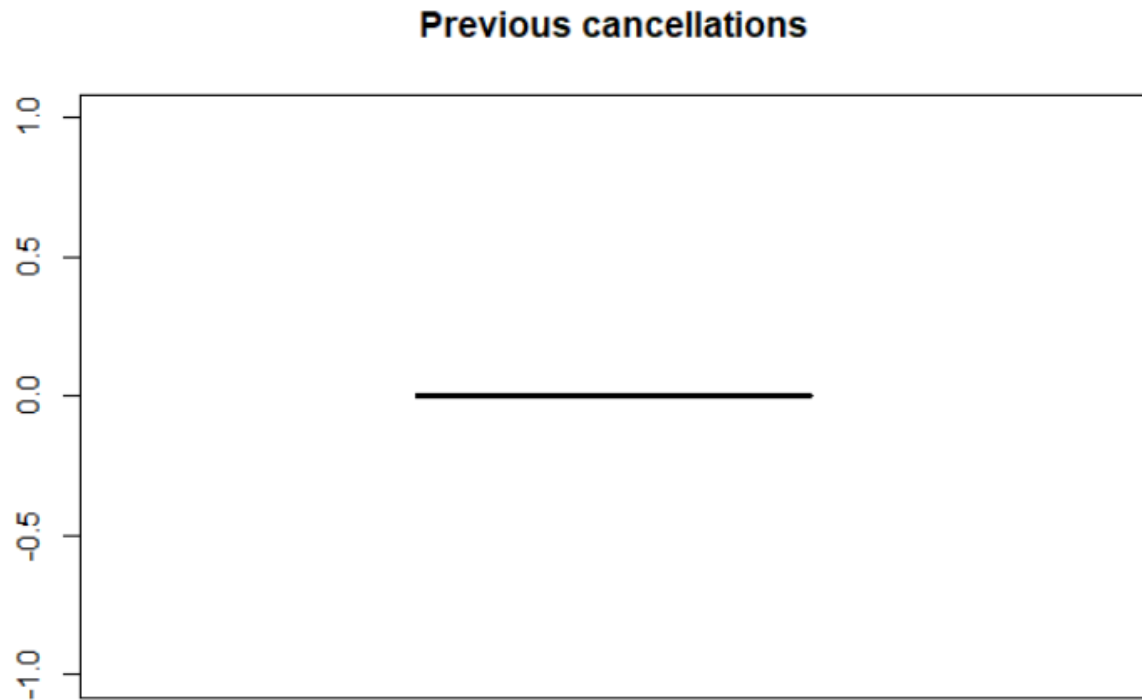


Al examinar las variables “total _of_ special_requests”, “lead_time”, “previous_cancellations” y “children”, se puede visualizar claramente que hay valores atípicos en varias variables del dataset, por lo que será necesario corregir las observaciones.

Para ello, hemos decidido emplear el método de reemplazo de los outliers con la media de cada una de las variables que solo contenga este tipo de valores.

```
>fix_outliers <- function(df){  
> for(i in 1:ncol(df))  
> {  
>   if(typeof(df[,i]) != "character")  
>   {  
>     Q <- quantile(df[,i], probs=c(0.25, 0.75))  
>     iqr <- IQR(df[,i])  
>     up <- Q[2]+1.5*iqr  
>     low<- Q[1]-1.5*iqr  
>     df[,i][df[,i] < (Q[1] - 1.5*iqr)] = round(mean(df[,i]),0)  
>     df[,i][df[,i] > (Q[2] + 1.5*iqr)] = round(mean(df[,i]),0)  
>   }  
> }  
> return(df)  
> }  
  
> booking_data_na = fix_outliers(booking_data_na)
```





Volvemos a graficar todos los datos de la variables “total_of_special_requests” y “previous_cancellations” para comparar con los respectivos diagramas de caja anteriormente presentadas. Con ello, observamos que los valores atípicos han desaparecido, debido a que estos han tomado el valor de la media generada en cada variable.

iv. VISUALIZAR DATOS

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

> count(booking_data_na, "hotel")

Al aplicar esta función se puede visualizar cuantas reservas hubo para cada hotel al recibir la Data y la columna hotel.

```

      hotel  freq
1  City Hotel 79330
2 Resort Hotel 40060

```

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

> library(ggplot2)

>

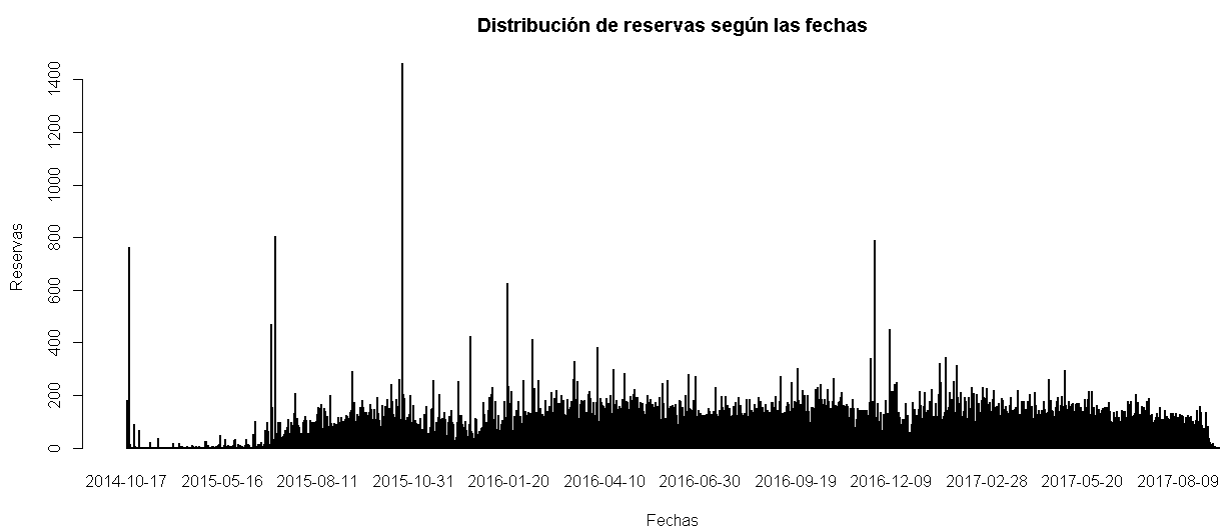
> n_bookings <- booking_data_na

```

>
> n_bookings$reservation_status_date <-
as.Date(n_bookings$reservation_status_date,"%m/%d/%Y")
> n_bookings <- n_bookings[order(n_bookings$reservation_status_date),]
>
> table_n <- table(n_bookings$reservation_status_date)
>
> barplot(table_n,main="Distribución de reservas según las
fechas",xlab="Fechas",ylab="Reservas")

```

Primero usamos la librería ggplot2 para dibujar y creamos un nuevo Data llamado n_booking , luego a este nuevo Data le cambiamos el orden la fecha para poder ordenarlo de forma sencilla, luego creamos una tabla usando la columna de “reservation_status_date”, y por último creamos la gráfica usando barplot.



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

Para conocer las temporadas de reservas, necesitamos calcular la cantidad de reservas que realizaron en cada temporada y categorizar cada una.

```

>n_bookings$reservation_year <- format(
as.Date(n_bookings$reservation_status_date,
format="%m/%d/%Y"),"%Y")

>n_bookings_per_year <- subset(n_bookings, n_bookings$reservation_year
!= "2014")

```

```
>table_year <- table(n_bookings_per_year$reservation_year)
```

```
>barplot(table_year,main="Temporadas de reservas",
```

```
      xlab="Año",
```

```
      ylab="# de reservas",
```

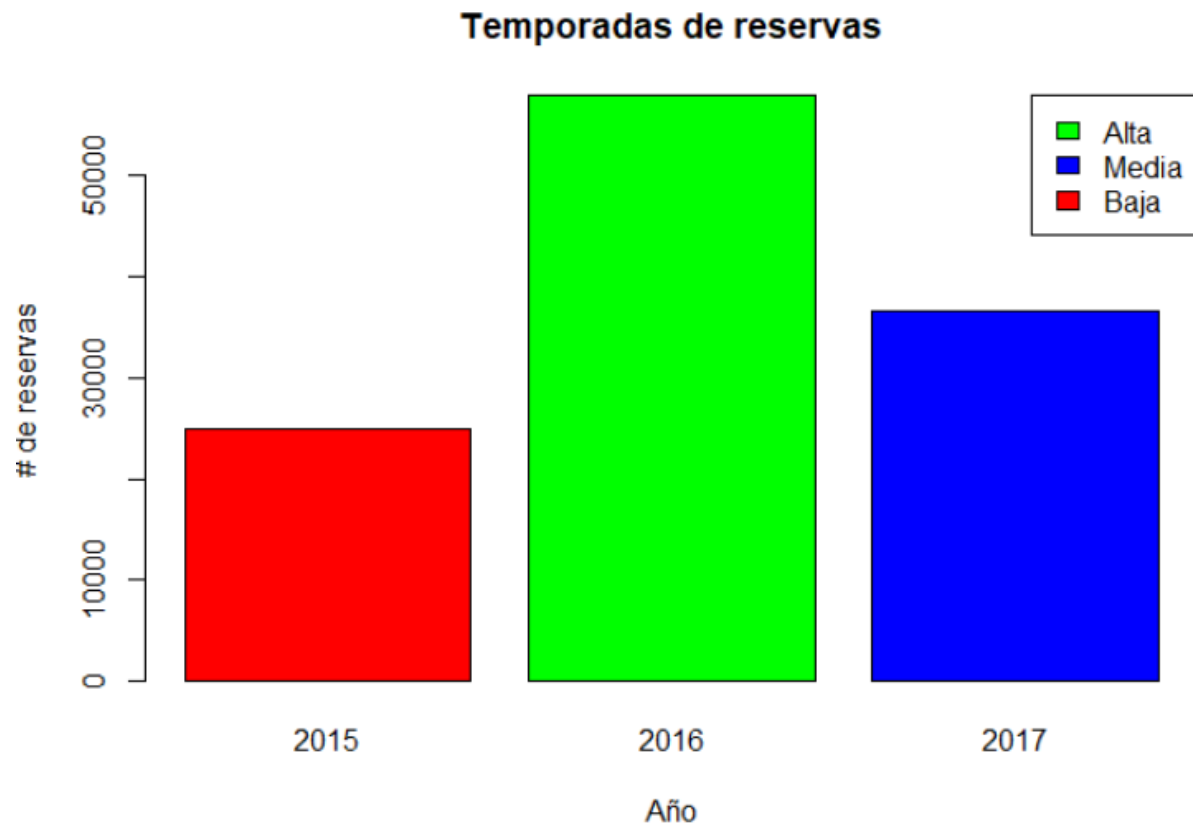
```
      col= c("red", "green", "blue"))
```

```
>legend("topright",
```

```
      c("Alta","Media","Baja"),
```

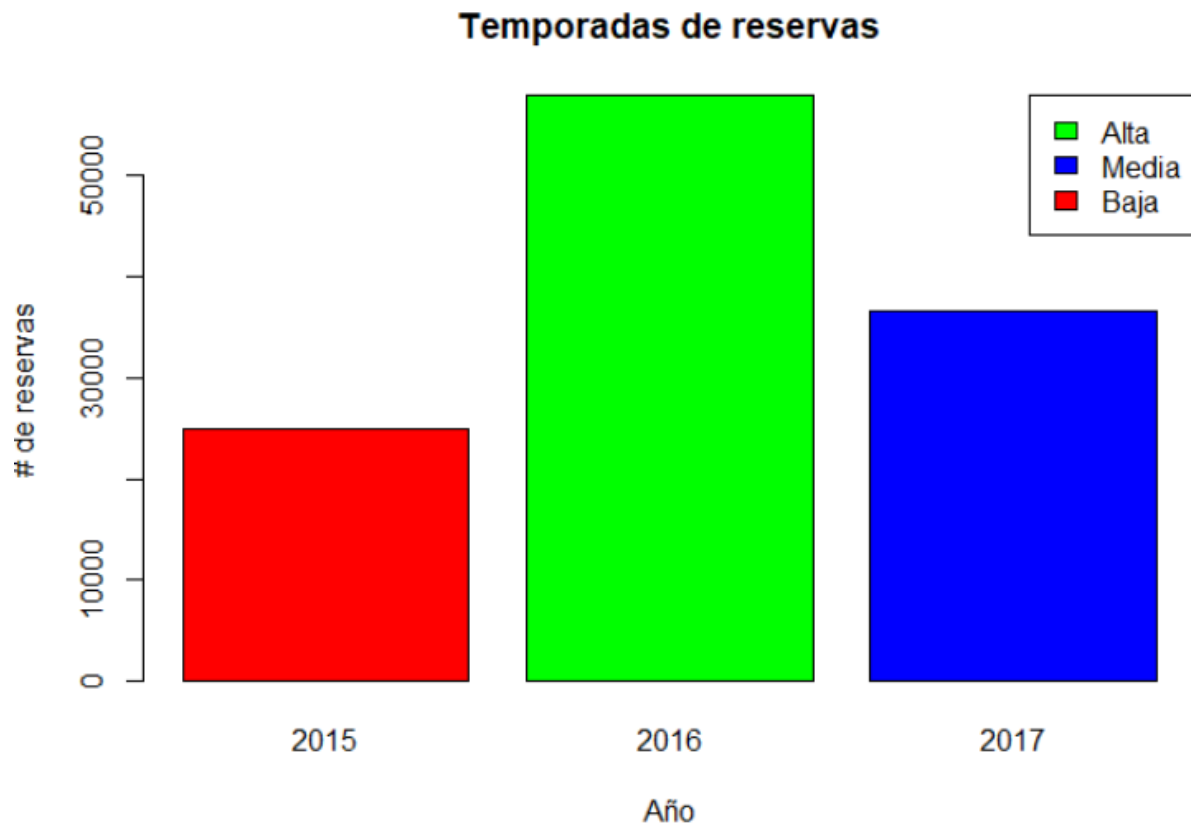
```
      fill= c("green","blue","red"))
```

Primero, creamos una nueva columna en el dataset `n_bookings` donde se almacenará solamente el año extraído de la variable “`reservation_status_date`”, ya que consideraremos los años del registro de la reserva como las temporadas. Luego, quitaremos todas las observaciones que contenga el año de registro de reserva 2014, pues presenta suficientes datos sobre las cantidad de reservas de como para considerarlo como una temporada. Por último, crearemos una tabla con la columna que almacena los años de registro de reserva de cada observación y la usaremos para generar un gráfico de barras que muestre las temporadas de reservas.



d. ¿Cuándo es menor la demanda de reservas?

Para descubrir el momento en que existe una menor demanda de reserva utilizaremos una gráfica que nos muestre las temporadas de reservas en el dataset.



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

Para conocer las reservas que incluyen niños y/o bebés realizaremos la siguiente instrucción:

```
> countBC <- function(df){
>   sumb=0
>   sumc=0
>   sumbyc=0
>   sumboc=0
>   for(i in 1:nrow(df))
>   {
>     if (df$Children[i] > 0 & df$babies[i] > 0)
>     {
>       sumbyc = sumbyc + 1
>     }
>     if (df$Children[i] > 0 & df$babies[i] == 0)
```



```

> {
>   sumc = sumc + 1
> }
> if (df$children[i] == 0 & df$babies[i] > 0)
> {
>   sumb = sumb + 1
> }
> if (df$children[i] > 0 | df$babies[i] > 0)
> {
>   sumboc = sumboc + 1
> }
> }
> cat("Cantidad de reservas solo con bebés:",sumb,"\n")
> cat("Cantidad de reservas solo con niños:",sumc,"\n")
> cat("Cantidad de reservas con bebés y niños:",sumbyc,"\n")
> cat("Cantidad de reservas con bebés o niños:",sumboc,"\n")
> }
> countBC(booking_data_na)

```

Donde primero creamos una función llamada countBC que recibe la data y donde dentro de esta creamos una sumatoria para cada tipo de casos como solo con bebés, sólo con niños, con bebés y niños, con bebés o niños. Donde cada sumatoria corresponde a la cantidad de reservas que incluye su tipo de caso.

```

> countBC(booking_data_na)
Cantidad de reservas solo con bebés: 0
Cantidad de reservas solo con niños: 0
Cantidad de reservas con bebés y niños: 0
Cantidad de reservas con bebés o niños: 0

```

Sale este resultado ya que cuando aplicamos la técnica para transformar los datos atípicos, ya que al haber muy pocas reservas que incluyen niños y/o bebés se transformaron a 0.

f. ¿Es importante contar con espacios de estacionamiento?

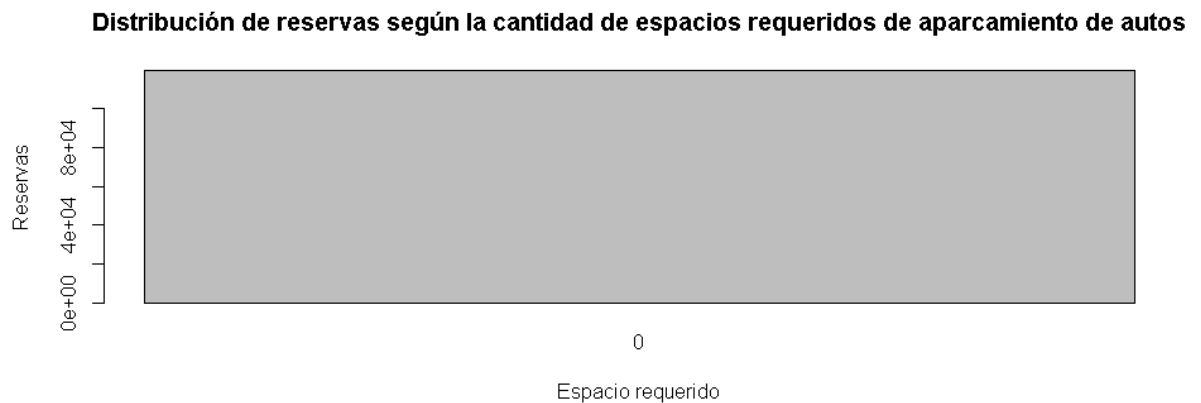
```

> table_vehiculos <- table(n_bookings$required_car_parking_spaces)

```

```
> barplot(table_vehiculos,main="Distribución de reservas según la
cantidad de espacios requeridos de aparcamiento de autos",xlab="Espacio
requerido",ylab="Reservas")
```

Primero creamos una tabla usando la columna `required_car_parking_spaces` y por último mostramos una gráfica de barras usando esta tabla.



Donde aquí también sucede lo mismo que el anterior que al haber muy pocos datos mayor de 0, estos valores se consideraron como valores atípicos y se reemplazaron por 0.

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

```
> library(ggplot2)
```

```
> mts<-c("January","February","March","April","May","June","July","
August","September","October","November","December")
```

```
> b2 <- booking_data_na
```

```
> b2$reservation_status_month<-format(as.Date(b2$reservation_status_date,
format="%m/%d/%Y"),"%m")
```

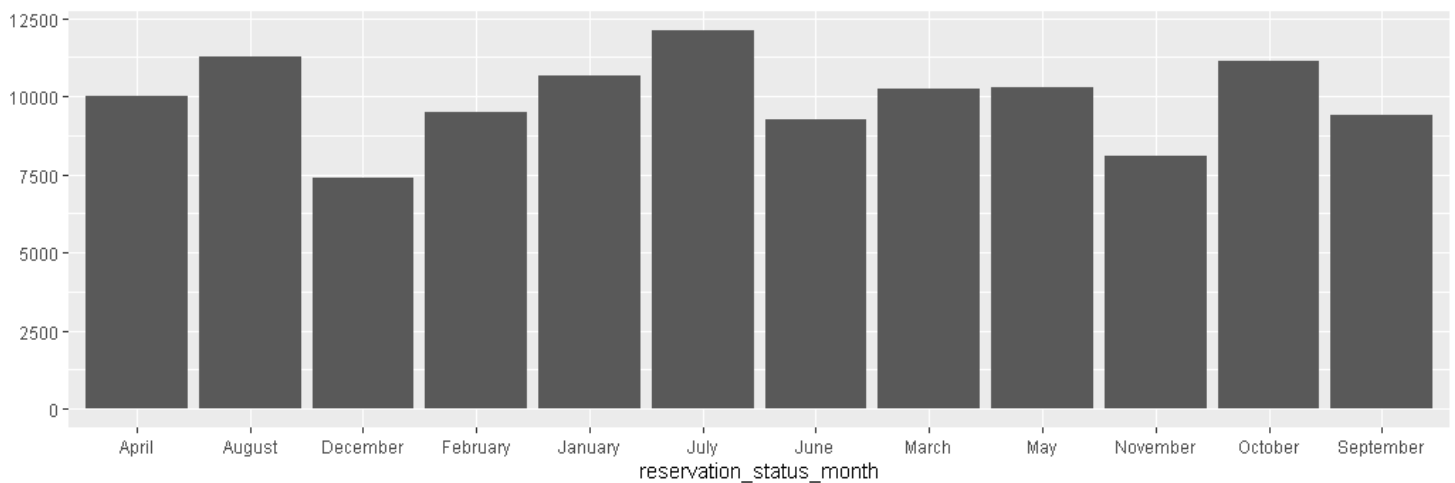
```
> b2$reservation_status_month <- as.integer(b2$reservation_status_month)
```

```
> b2$reservation_status_month <- mts[b2$reservation_status_month]
```

```
> cancelByMonth <- subset(b2, b2$is_canceled == 1, select = c(is_canceled,
reservation_status_month))
```

```
> qplot(data = b2, reservation_status_month, fill = is_canceled, bins = 30,
main="")
```

Ya que usaremos un histograma para visibilizar el número de cancelaciones de reserva por mes, invocamos la librería `ggplot2`. Luego, agregaremos una columna llamada `reservation_status_month` que contendrá los meses en que los estados de las reservas fueron actualizadas. Crearemos un subset del dataset modificado que solo incluirá todas las reservas que han sido canceladas, y dos columnas de las variables `is canceled` y `reservation_status_month`. Para ello, escribimos la condición de que la variable `is canceled` tiene que ser igual a 1. Finalmente, graficamos el subset creado para observar el número de reservas canceladas en cada mes.



4. Conclusiones preliminares

- a. En el City Hotel se realizaron 79330 reservas mientras que en el Resort Hotel se realizaron 40060 reservas, por lo que se concluye que la gente prefiere el City Hotel para hospedarse.
- b. Con respecto a los últimos días la demanda no está aumentando , aunque es bien difícil especificar si está aumentando o disminuyendo ya que hay una gran variación de cantidad de reservas hechas cada día.
- c. Con respecto a las tres temporadas reservas anuales se dividen en tres categorías donde el año 2015 pertenece a la categoría baja, el año 2016 pertenece a la categoría alta, y el año 2017 pertenece a la categoría media.
- d. Hubo la menor demanda de reservas en el año 2015 en el cual este pertenece a la categoría baja con respecto a las temporadas de reservas anuales.
- e. La cantidad de reservas que incluyen solo bebés es 0, solo niños es 0, con bebés y niños es 0, y con bebés o niños es 0.

f. No es importante contar con espacios de estacionamientos.

g. Los tres primeros meses en que se producen mayor cancelaciones son los meses de Julio, Agosto y Octubre.

5. Referencias

Link del github: <https://github.com/sergio185678/ea-2021-1-cc51>

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