Universidad Peruana de Ciencias Aplicadas

INFORME DEL TRABAJO PARCIAL

CURSO FUNDAMENTOS DATA SCIENCE

Carrera de Ciencias de la Computación.

Sección: cc52

Alumnos:	
Código	Nombres y apellidos
u202019493	Javier Enrique Silva Barrientos
u201916314	Tomas Alonso Pastor Salazar
u202114233	Tarrillo Ayllon Anthony Hans

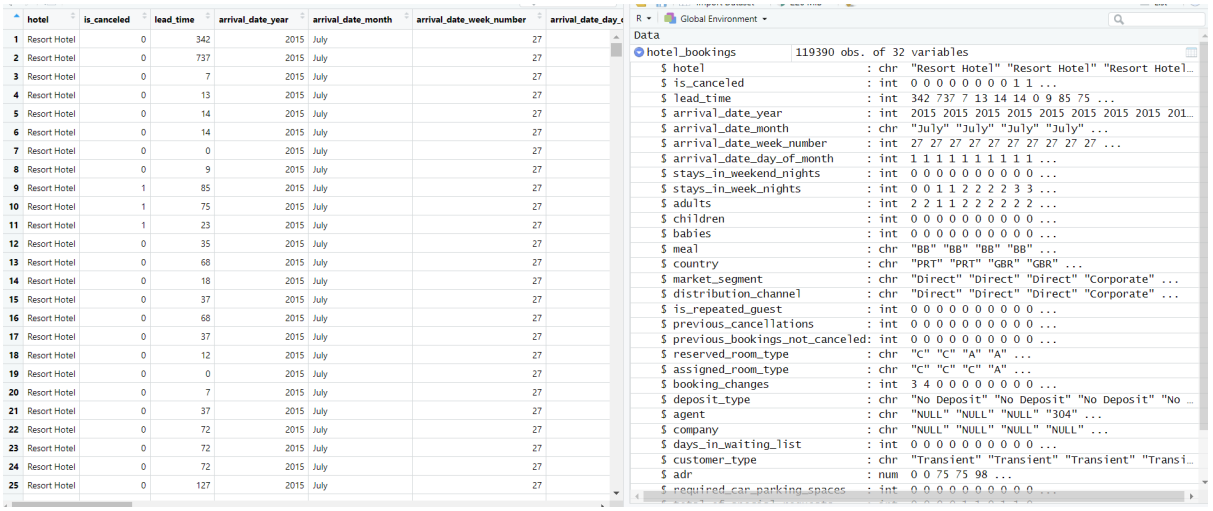
Septiembre - 2023

Descripción del dataset

El dataset que estamos analizando se llama hotel_booking.csv. Es un dataset de hoteles recomendado por el enunciado de la TP. El cual es importante para una amplia gama de personas y organizaciones que tienen interés en la industria hotelera, ya sea para la toma de decisiones personales, operaciones de un negocio, llevar a cabo investigaciones o cumplir con funciones regulatorias. Estos datos son importantes para informar y mejorar la toma de decisiones relacionadas con los hoteles y el alojamiento en general.

Para poder abrir el dataset descargado hemos utilizado el siguiente código en R:

```
## I.CARGA DE DATOS
setwd("C:/Users/Anthony/Downloads/R")
hotel_bookings<-read.csv('hotel_bookings.csv', header=TRUE,stringsAsFactors = FALSE, sep=',',dec='.')
```



	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_
1	Resort Hotel	0	342	2015	July	27	
2	Resort Hotel	0	737	2015	July	27	
3	Resort Hotel	0	7	2015	July	27	
4	Resort Hotel	0	13	2015	July	27	
5	Resort Hotel	0	14	2015	July	27	
6	Resort Hotel	0	14	2015	July	27	
7	Resort Hotel	0	0	2015	July	27	
8	Resort Hotel	0	9	2015	July	27	
9	Resort Hotel	1	85	2015	July	27	
10	Resort Hotel	1	75	2015	July	27	
11	Resort Hotel	1	23	2015	July	27	
12	Resort Hotel	0	35	2015	July	27	
13	Resort Hotel	0	68	2015	July	27	
14	Resort Hotel	0	18	2015	July	27	
15	Resort Hotel	0	37	2015	July	27	
16	Resort Hotel	0	68	2015	July	27	
17	Resort Hotel	0	37	2015	July	27	
18	Resort Hotel	0	12	2015	July	27	
19	Resort Hotel	0	0	2015	July	27	
20	Resort Hotel	0	7	2015	July	27	
21	Resort Hotel	0	37	2015	July	27	
22	Resort Hotel	0	72	2015	July	27	
23	Resort Hotel	0	72	2015	July	27	
24	Resort Hotel	0	72	2015	July	27	
25	Resort Hotel	0	127	2015	July	27	

Este conjunto de datos, denominado 'hotel_booking.csv', desempeña un papel fundamental en nuestra investigación actual. Su relevancia es significativa tanto para una audiencia diversa como para organizaciones interesadas en la industria hotelera. Estos datos son esenciales para informar y mejorar la toma de decisiones relacionadas con la gestión de hoteles y alojamientos en general. Además, su utilidad no se limita a este propósito, ya que también desempeña un papel importante en el campo del aprendizaje automático, mediante la implementación de lenguajes de programación como Python o R.

En primer lugar, convertimos las variables correspondientes a factores para poder visualizar mejor estos datos y procedemos a examinar las características del conjunto de datos:

```r
# Convertimos a factores las variables correspondientes
hotel_bookings$is_canceled<-as.numeric(hotel_bookings$is_canceled)
hotel_bookings$is_canceled  <- as.factor(hotel_bookings$is_canceled)
hotel_bookings$arrival_date_year  <- as.factor(hotel_bookings$arrival_date_year)
hotel_bookings$arrival_date_month  <- as.factor(hotel_bookings$arrival_date_month)
hotel_bookings$meal  <- as.factor(hotel_bookings$meal)
hotel_bookings$country  <- as.factor(hotel_bookings$country)
hotel_bookings$market_segment  <- as.factor(hotel_bookings$market_segment)
hotel_bookings$distribution_channel  <- as.factor(hotel_bookings$distribution_channel)
hotel_bookings$reserved_room_type  <- as.factor(hotel_bookings$reserved_room_type)
hotel_bookings$assigned_room_type  <- as.factor(hotel_bookings$assigned_room_type)
hotel_bookings$deposit_type  <- as.factor(hotel_bookings$deposit_type)
hotel_bookings$customer_type  <- as.factor(hotel_bookings$customer_type)
hotel_bookings$reservation_status  <- as.factor(hotel_bookings$reservation_status)
hotel_bookings$agent  <- as.factor(hotel_bookings$agent)
hotel_bookings$is_repeated_guest  <- as.factor(hotel_bookings$is_repeated_guest)
```

```r
45  # Estructura del conjunto de datos
46  str(hotel_bookings)
```

```
> str(hotel_bookings)
'data.frame':   119390 obs. of  32 variables:
 $ hotel                         : chr  "Resort Hotel" "Resort Hotel" "Resort Hotel" "Resort Hotel" ...
 $ is_canceled                   : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 2 2 ...
 $ lead_time                     : int  342 737 7 13 14 14 0 9 85 75 ...
 $ arrival_date_year             : Factor w/ 3 levels "2015","2016",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ arrival_date_month            : Factor w/ 12 levels "April","August",..: 6 6 6 6 6 6 6 6 6 6 ...
 $ 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  0 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                          : Factor w/ 5 levels "BB","FB","HB",..: 1 1 1 1 1 1 1 1 2 1 3 ...
 $ country                       : Factor w/ 178 levels "ABW","AGO","AIA",..: 137 137 60 60 60 60 137 137 137 137 ...
 $ market_segment                : Factor w/ 8 levels "Aviation","Complementary",..: 4 4 4 3 7 7 4 4 7 6 ...
 $ distribution_channel          : Factor w/ 5 levels "Corporate","Direct",..: 2 2 2 1 4 4 2 2 4 4 ...
 $ is_repeated_guest             : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ 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            : Factor w/ 10 levels "A","B","C","D",..: 3 3 1 1 1 1 3 3 1 4 ...
 $ assigned_room_type            : Factor w/ 12 levels "A","B","C","D",..: 3 3 3 1 1 1 3 3 1 4 ...
 $ booking_changes               : int  3 4 0 0 0 0 0 0 0 0 ...
 $ deposit_type                  : Factor w/ 3 levels "No Deposit","Non Refund",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ agent                         : Factor w/ 334 levels "1","10","103",..: 334 334 334 157 103 103 334 156 103 40 ...
 $ company                       : chr  "NULL" "NULL" "NULL" "NULL" ...
 $ days_in_waiting_list          : int  0 0 0 0 0 0 0 0 0 0 ...
 $ customer_type                 : Factor w/ 4 levels "Contract","Group",..: 3 3 3 3 3 3 3 3 3 3 ...
 $ 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            : Factor w/ 3 levels "Canceled","Check-Out",..: 2 2 2 2 2 2 2 2 1 1 ...
 $ reservation_status_date       : chr  "2015-07-01" "2015-07-01" "2015-07-02" "2015-07-02" ...
```

```r
# Resumen del conjunto de datos
summary(hotel_bookings)
```

```
> summary(hotel_bookings)
    hotel          is_canceled   lead_time      arrival_date_year arrival_date_month arrival_date_week_number arrival_date_day_of_month stays_in_weekend_nights
 Length:119390      0:75166     Min.   :  0   2015:21996        August :13877      Min.   : 1.00            Min.   : 1.0              Min.   : 0.0000
 Class :character   1:44224     1st Qu.: 18   2016:56707        July   :12661      1st Qu.:16.00            1st Qu.: 8.0              1st Qu.: 0.0000
 Mode  :character               Median : 69   2017:40687        May    :11791      Median :28.00            Median :16.0              Median : 1.0000
                                Mean   :104                     October:11160      Mean   :27.17            Mean   :15.8              Mean   : 0.9276
                                3rd Qu.:160                     April  :11089      3rd Qu.:38.00            3rd Qu.:23.0              3rd Qu.: 2.0000
                                Max.   :737                     June   :10939      Max.   :53.00            Max.   :31.0              Max.   :19.0000
                                                                (Other):47873
 stays_in_week_nights     adults          children         babies                meal          country           market_segment  distribution_channel
 Min.   : 0.0         Min.   : 0.000   Min.   : 0.0000   Min.   : 0.000000   BB       :92310   PRT    :48590   Online TA    :56477   Corporate: 6677
 1st Qu.: 1.0         1st Qu.: 2.000   1st Qu.: 0.0000   1st Qu.: 0.000000   FB       :  798   GBR    :12129   Offline TA/TO:24219   Direct   :14645
 Median : 2.0         Median : 2.000   Median : 0.0000   Median : 0.000000   HB       :14463   FRA    :10415   Groups       :19811   GDS      :  193
 Mean   : 2.5         Mean   : 1.856   Mean   : 0.1039   Mean   : 0.007949   SC       :10650   ESP    : 8568   Direct       :12606   TA/TO    :97870
 3rd Qu.: 3.0         3rd Qu.: 2.000   3rd Qu.: 0.0000   3rd Qu.: 0.000000   Undefined: 1169   DEU    : 7287   Corporate    : 5295   Undefined:    5
 Max.   :50.0         Max.   :55.000   Max.   :10.0000   Max.   :10.000000                     ITA    : 3766   Complementary:  743
                                       NA's   :4                                               (Other):28635   (Other)      :  239
 is_repeated_guest previous_cancellations previous_bookings_not_canceled reserved_room_type assigned_room_type booking_changes       deposit_type
 0:115580          Min.   : 0.00000       Min.   : 0.0000                A      :85994       A      :74053       Min.   : 0.0000   No Deposit:104641
 1:  3810          1st Qu.: 0.00000       1st Qu.: 0.0000                D      :19201       D      :25322       1st Qu.: 0.0000   Non Refund: 14587
                   Median : 0.00000       Median : 0.0000                E      : 6535       E      : 7806       Median : 0.0000   Refundable:   162
                   Mean   : 0.08712       Mean   : 0.1371                F      : 2897       F      : 3751       Mean   : 0.2211
                   3rd Qu.: 0.00000       3rd Qu.: 0.0000                G      : 2094       G      : 2553       3rd Qu.: 0.0000
                   Max.   :26.00000       Max.   :72.0000                B      : 1118       C      : 2375       Max.   :21.0000
                                                                         (Other): 1551       (Other): 3530
     agent          company          days_in_waiting_list        customer_type        adr          required_car_parking_spaces total_of_special_requests
 9      :31961   Length:119390      Min.   :  0.000      Contract       : 4076   Min.   :  -6.38   Min.   :0.00000             Min.   :0.0000
 NULL   :16340   Class :character   1st Qu.:  0.000      Group          :  577   1st Qu.:  69.29   1st Qu.:0.00000             1st Qu.:0.0000
 240    :13922   Mode  :character   Median :  0.000      Transient      :89613   Median :  94.58   Median :0.00000             Median :0.0000
 1      : 7191                      Mean   :  2.321      Transient-Party:25124   Mean   : 101.83   Mean   :0.06252             Mean   :0.5714
 14     : 3640                      3rd Qu.:  0.000                             3rd Qu.: 126.00   3rd Qu.:0.00000             3rd Qu.:1.0000
 7      : 3539                      Max.   :391.000                             Max.   :5400.00   Max.   :8.00000             Max.   :5.0000
 (Other):42797
 reservation_status reservation_status_date
 Canceled :43017    Length:119390
 Check-Out:75166    Class :character
 No-Show  : 1207    Mode  :character
```

Acerca de las variables mencionaremos sobre las más importantes que la columna hotel representa si es un Resort hotel o un City hotel, la columna is_cancelled representa si el pago se canceló o no, lead_time representa el tiempo de espera, las columnas arrival_date_(year, month, week_number, day_of_the_month) representan la fecha de llegada de la estadia en el hotel, las columnas adults,children,babies representan la cantidad de adultos, niños y bebes registrados, respectivamente, en los hoteles, la columna required_car_parking_spaces representa la cantidad de espacios requeridos para estacionamiento por registro.

2. Limpieza de datos

Antes de esto, creamos un conjunto de datos igual a este que será el que editaremos para que sea el final.Será hotel_bookings_final:

```
hotel_bookings_final<-hotel_bookings
```

2.1 Elementos N.A.

Para poder hacer una limpieza de datos vacíos, debemos saber en donde se encuentran y eso se puede visualizar con el siguiente código de R.

```
## convertir los NULLS en NA
nullos<-which(hotel_bookings_final=="NULL",arr.ind=TRUE)
hotel_bookings_final[nullos]<-NA
## convertir los UNDEFINED en NA
indef<-which(hotel_bookings_final=="Undefined",arr.ind=TRUE)
hotel_bookings_final[indef]<-NA
```

```r
## Funcion para hallar la cantidad de datos faltantes en cada variable o columna

nroNA <- function(x){
  sum = 0
  for(i in 1:ncol(x))
  {
    cat("En la columna",colnames(x[i]),"total de valores NA:",colSums(is.na(x[i])),"\n")
  }
}
nroNA(hotel_bookings_final)
```

```
> nroNA(hotel_bookings_final)
En la columna hotel total de valores NA: 0
En la columna is_canceled total de valores NA: 0
En la columna lead_time total de valores NA: 0
En la columna arrival_date_year total de valores NA: 0
En la columna arrival_date_month total de valores NA: 0
En la columna arrival_date_week_number total de valores NA: 0
En la columna arrival_date_day_of_month total de valores NA: 0
En la columna stays_in_weekend_nights total de valores NA: 0
En la columna stays_in_week_nights total de valores NA: 0
En la columna adults total de valores NA: 0
En la columna children total de valores NA: 4
En la columna babies total de valores NA: 0
En la columna meal total de valores NA: 1169
En la columna country total de valores NA: 488
En la columna market_segment total de valores NA: 2
En la columna distribution_channel total de valores NA: 5
En la columna is_repeated_guest total de valores NA: 0
En la columna previous_cancellations total de valores NA: 0
En la columna previous_bookings_not_canceled total de valores NA: 0
En la columna reserved_room_type total de valores NA: 0
En la columna assigned_room_type total de valores NA: 0
En la columna booking_changes total de valores NA: 0
En la columna deposit_type total de valores NA: 0
En la columna agent total de valores NA: 16340
En la columna company total de valores NA: 112593
En la columna days_in_waiting_list total de valores NA: 0
En la columna customer_type total de valores NA: 0
En la columna adr total de valores NA: 0
En la columna required_car_parking_spaces total de valores NA: 0
En la columna total_of_special_requests total de valores NA: 0
En la columna reservation_status total de valores NA: 0
En la columna reservation_status_date total de valores NA: 0
```

Con esto podemos visualizar que la columna “children”, “meal”, “country”,”market_segment”,”distribution_channel”,”agent” y “company” tienen valores NA en sus conjuntos de datos.

Para poder sacar la moda de una variable, esto para poder reemplazar por ciertos valores NA:

```r
##hallo la moda

Mode <- function(x) {
  uniq_x <- unique(x)
  freq_x <- tabulate(match(x, uniq_x))
  return(uniq_x[which.max(freq_x)])
}
```

Para poder eliminar los NA del dataset si es que la cantidad de estos no presenta un alto porcentaje en su total o presenta un porcentaje demasiado alto se hace de la siguiente manera:

```
#Comenzamos a reemplazar los datos faltantes
#children
#Presenta solo 4 valores NA de 119390 por lo que podemos eliminarlos.
hotel_bookings_final<-hotel_bookings_final[!is.na(hotel_bookings_final$children),]
#Al eliminar estas dos filas tambien eliminamos aquellas cuales su market_segment era NA.
```

```
#agent
#podriamos reemplazar los valores vacios por la moda pero esta variable no afecta en la estadística
#por esto lo eliminaremos
hotel_bookings_final$agent <- NULL
#company
#El 94.30% presentan valores NA por lo que lo mejor sera eliminar esta variable ya que, además,
#no presenta ningún dato estadistico más
hotel_bookings_final$company <- NULL
```

Para los demás valores NA del dataset vamos a reemplazarlos por la moda ya que estos representan un porcentaje para nada despreciable en su total:

```
#meal
#Remplazamos los valores vacios por la moda
moda<-Mode(hotel_bookings_final$meal)
hotel_bookings_final$meal[is.na(hotel_bookings_final$meal)]<-moda

#country
#Remplazamos los valores vacios por la moda
moda<-Mode(hotel_bookings_final$country)
hotel_bookings_final$country[is.na(hotel_bookings_final$country)]<-moda
```

Volvemos a llamar a la función que nos muestra la cantidad de valores NA de cada columna:

```
#Volvemos a ver si hay valores NA:
nroNA(hotel_bookings_final)
```

```
> nroNA(hotel_bookings_final)
En la columna hotel total de valores NA: 0
En la columna is_canceled total de valores NA: 0
En la columna lead_time total de valores NA: 0
En la columna arrival_date_year total de valores NA: 0
En la columna arrival_date_month total de valores NA: 0
En la columna arrival_date_week_number total de valores NA: 0
En la columna arrival_date_day_of_month total de valores NA: 0
En la columna stays_in_weekend_nights total de valores NA: 0
En la columna stays_in_week_nights total de valores NA: 0
En la columna adults total de valores NA: 0
En la columna children total de valores NA: 0
En la columna babies total de valores NA: 0
En la columna meal total de valores NA: 0
En la columna country total de valores NA: 0
En la columna market_segment total de valores NA: 0
En la columna distribution_channel total de valores NA: 0
En la columna is_repeated_guest total de valores NA: 0
En la columna previous_cancellations total de valores NA: 0
En la columna previous_bookings_not_canceled total de valores NA: 0
En la columna reserved_room_type total de valores NA: 0
En la columna assigned_room_type total de valores NA: 0
En la columna booking_changes total de valores NA: 0
En la columna deposit_type total de valores NA: 0
En la columna days_in_waiting_list total de valores NA: 0
En la columna customer_type total de valores NA: 0
En la columna adr total de valores NA: 0
En la columna required_car_parking_spaces total de valores NA: 0
En la columna total_of_special_requests total de valores NA: 0
En la columna reservation_status total de valores NA: 0
En la columna reservation_status_date total de valores NA: 0
```

Y aca podemos ver los valores eliminados “NA” que observamos en el anterior código:

Data	
hotel_bookings	119390 obs. of 32 variables
hotel_bookings_final	119385 obs. of 30 variables

3. Identificación de outliers

Una vez que ya tenemos el dataset limpio, podemos visualizar algunos outliers importantes. Haremos el boxplot para las edades del dataset que tenemos, primero veremos como detectar los outliers de cada columna:

```r
#funcion para determinar la cantidad de valores atipicos de una variable
is_outlier <- function(vector) {
  q1 <- quantile(vector, 0.25)
  q3 <- quantile(vector, 0.75)
  iqr <- q3 - q1
  limite_superior <- q3 + 1.5 * iqr
  limite_inferior <- q1 - 1.5 * iqr
  valores_atipicos <- vector > limite_superior | vector < limite_inferior
  return(valores_atipicos)
}
```

Usamos esta función para detectar los valores atípicos de cada variable:

```r
#Funcion que revisa cuantos valores atipicos tiene cada variable.
for(i in 1:ncol(hotel_bookings_final)){
  if(is.numeric(hotel_bookings_final[,i])){
    sum(is_outlier(hotel_bookings_final[,i]))
    if (sum(is_outlier(hotel_bookings_final[,i]))){cat("En la columna",colnames(hotel_bookings_final[i]),"total de valores atipicos:",sum(is_outlier(hotel_bookings_final
  }else(cat("no tiene\n")) }else(cat("no tiene\n"))}
```

```
no tiene
no tiene
En la columna lead_time total de valores atipicos: 3005
no tiene
no tiene
no tiene
no tiene
En la columna stays_in_weekend_nights total de valores atipicos: 265
En la columna stays_in_week_nights total de valores atipicos: 3354
En la columna adults total de valores atipicos: 29709
En la columna children total de valores atipicos: 8589
En la columna babies total de valores atipicos: 917
no tiene
no tiene
no tiene
no tiene
En la columna is_repeated_guest total de valores atipicos: 3810
En la columna previous_cancellations total de valores atipicos: 6484
En la columna previous_bookings_not_canceled total de valores atipicos: 3620
no tiene
no tiene
En la columna booking_changes total de valores atipicos: 18076
no tiene
En la columna days_in_waiting_list total de valores atipicos: 3698
no tiene
En la columna adr total de valores atipicos: 3793
En la columna required_car_parking_spaces total de valores atipicos: 7415
En la columna total_of_special_requests total de valores atipicos: 2877
no tiene
no tiene
```

Podemos observar que varias columnas tienen valores atípicos sin embargo, como grupo, decidimos no modificar la gran mayoría puesto que nos iban a ayudar a dar respuestas mas completas a ciertas preguntas.

Aún así hemos hecho una demostración de reemplazo de estos valores por la moda o el límite superior:

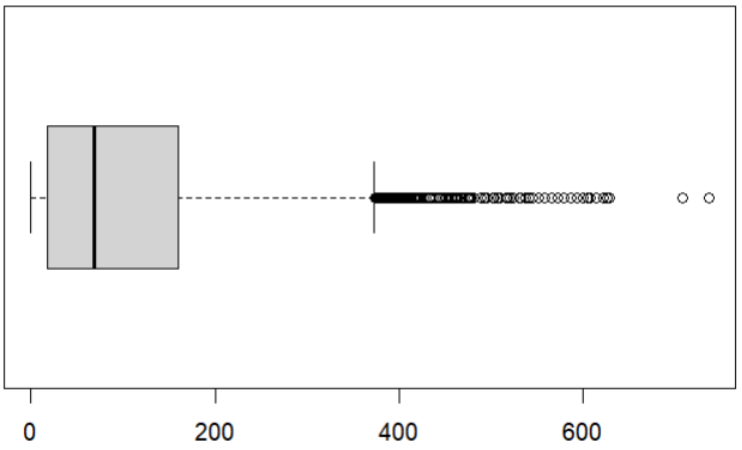
```r
#Reemplazamos lo valores atipicos de cada variable
#FUNCION PARA REEMPLAZAR VALORES ATIPICOS POR LA MODA:
convert <- function(x){
  b<-boxplot.stats(x)
  while(length(b$out)>0){
    b<-boxplot.stats(x)
    for(i in 1:nrow(hotel_bookings_final)){
      if(x[i] %in% b$out){
        x[i]<-b$stats[3]
      } }
  }
  return(x)
}
#FUNCION PARA REEMPLAZAR VALORES ATIPICOS POR EL LIMITE SUPERIOR:
convertmax <- function(x){
  b<-boxplot.stats(x)
  while(length(b$out)>0){
    b<-boxplot.stats(x)
    for(i in 1:nrow(hotel_bookings_final)){
      if(x[i] %in% b$out){
        x[i]<-b$stats[5]
      } }
  }
  return(x)
}
```

Usando estas funciones reemplazamos los valores atípicos de LEAD_TIME y adr:

```r
#LEAD_TIME
#en este caso convertimos los valores atipicos a valores del limite superior
#ANTES DEL REEMPLAZO:
boxplot.stats(hotel_bookings_final$lead_time)
boxplot(hotel_bookings_final$lead_time, horizontal = TRUE)
sum(is_outlier(hotel_bookings_final$lead_time))
```

```
$stats
[1]   0  18  69 160 373

$n
[1] 119385

$conf
[1] 68.35066 69.64934
```

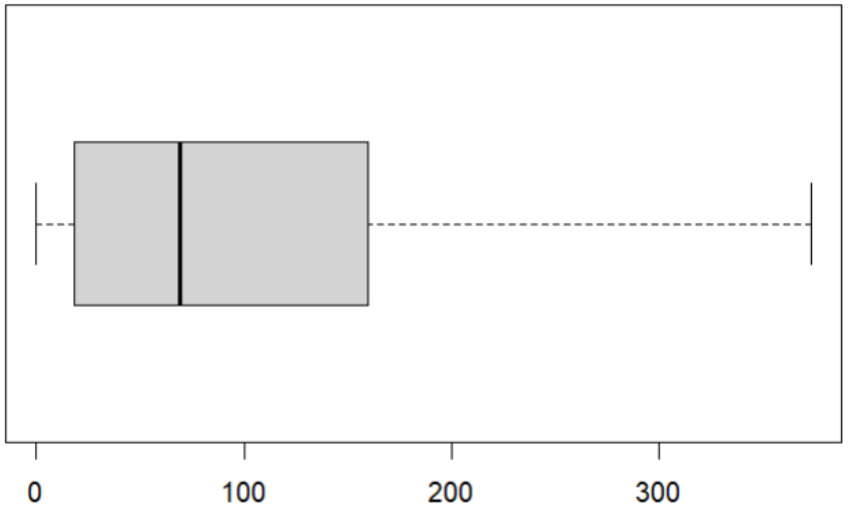
```
$out
  [1] 737 394 460 381 382 709 468 468 468 468 468 468 468 468 468 468 398 424 434 374 406 406 406
 [24] 406 406 400 379 399 385 422 390 390 394 376 376 376 375 385 385 385 397 397 385 385 397
 [47] 397 397 385 385 397 397 397 397 397 385 385 385 385 385 397 397 397 385 385 397 397 385 385
 [70] 397 397 385 385 385 542 542 542 542 542 542 542 542 542 542 542 403 403 383 383 383 383 383 383
 [93] 383 383 383 383 383 383 383 383 383 383 383 383 383 383 383 383 383 383 383 383 383 383 383
[116] 383 383 383 383 383 383 383 383 383 383 383 383 383 383 383 383 383 383 383 383 383 383 384
[139] 385 393 393 393 393 393 393 393 393 393 393 393 393 393 393 393 393 393 393 393 393 393 393
[162] 393 393 393 393 393 393 393 435 375 382 386 386 385 385 385 385 385 386 386 386 386 386 386
[185] 386 386 386 386 386 386 386 386 386 378 378 378 378 378 378 378 378 378 378 378 378 378 378
[208] 378 378 378 378 471 471 471 471 471 471 462 462 462 462 462 462 462 462 462 462 462 462 462
[231] 462 462 462 462 462 462 462 462 411 411 411 411 411 411 411 411 411 411 411 411 411 411 411
[254] 411 411 411 411 411 450 411 411 411 411 390 381 378 454 399 468 468 468 468 468 468 468 468
[277] 468 468 468 468 468 468 468 468 468 468 468 468 460 460 532 468 468 468 468 468 468 468 468
[300] 468 468 468 468 468 468 468 468 383 386 383 406 422 445 542 542 542 542 445 445 542 542 542
[323] 542 542 542 542 542 542 383 383 383 383 383 383 383 383 383 383 383 383 383 383 383 384 383 383
[346] 383 383 383 386 386 386 386 386 386 386 386 386 386 386 386 386 390 386 386 389 389 386 386 386
[369] 386 386 386 386 386 386 386 386 386 386 386 386 386 386 386 386 386 386 386 386 385 386 386
[392] 386 386 386 386 386 386 386 386 386 386 386 386 388 388 388 388 388 388 388 388 388 388 388
[415] 388 388 388 388 388 388 388 388 388 379 379 407 407 379 379 393 393 393 393 393 393 393 393
[438] 393 393 393 393 393 393 393 393 393 393 393 393 393 393 393 393 393 393 393 393 393 393 443
[461] 443 443 443 443 443 443 443 443 443 443 443 443 443 443 443 443 443 443 443 443 443 443 437 437
[484] 437 437 437 437 437 437 437 437 437 437 437 437 437 437 437 437 437 437 437 437 437 437 437
[507] 437 437 437 437 437 451 451 451 451 451 451 451 451 451 451 451 451 451 451 451 451 451 451 451
[530] 451 451 451 451 451 451 451 451 451 451 451 451 384 384 384 384 384 384 384 384 384 384 384
[553] 384 384 384 384 384 384 384 384 384 384 384 384 379 379 379 379 379 379 379 379 379 379 379
[576] 379 379 379 379 379 379 379 379 379 379 379 379 379 379 379 379 379 379 379 379 379 391 391
[599] 391 391 391 391 391 391 391 391 391 391 391 391 391 391 391 391 391 391 391 391 391 386 386
[622] 386 386 386 386 386 386 386 386 386 386 386 386 386 386 386 386 386 386 386 386 386 386 386
[645] 386 386 386 386 386 386 386 391 391 391 391 391 391 391 391 391 391 391 391 391 391 391 391 391
[668] 391 391 391 391 391 391 391 391 391 391 391 391 391 391 391 405 405 405 405 405 405 405 405
[691] 405 405 405 405 405 405 405 405 405 405 405 405 405 405 405 405 398 398 398 398 398 398 398
[714] 398 398 398 398 398 398 398 398 398 398 398 398 398 398 398 398 398 398 398 398 398 398 398
[737] 398 398 398 405 405 405 405 405 405 405 405 405 405 405 405 405 405 405 405 405 405 405 405
[760] 405 405 405 405 405 405 405 405 405 405 405 405 412 412 412 412 412 412 412 412 412 412 412
[783] 412 412 412 412 412 412 412 412 412 412 412 412 412 412 412 412 412 412 412 419 419 419 419
[806] 419 419 419 419 419 419 419 419 419 419 419 419 419 419 419 419 419 419 419 419 419 419 419
[829] 419 419 419 419 419 419 420 420 420 420 420 420 420 420 420 420 420 420 420 420 420 420 420 420
[852] 420 420 420 420 420 420 420 420 420 420 420 420 426 426 426 426 426 426 426 426 426 426 426
[875] 426 426 426 426 426 426 426 426 426 426 426 426 426 426 426 426 426 426 426 426 426 426 433
```




```
> sum(is_outlier(hotel_bookings_final$lead_time))
[1] 3005
```

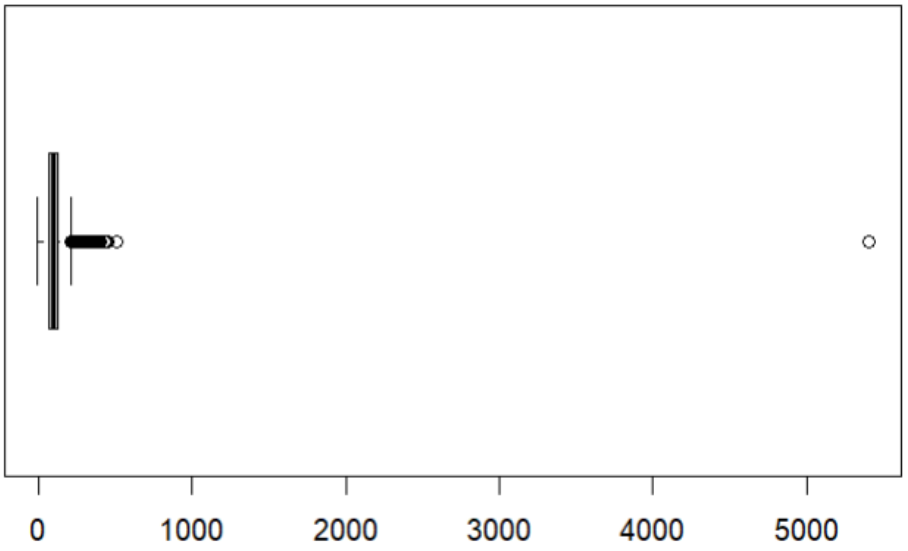
```
174  #DESPUES DEL REEMPLAZO:
175  hotel_bookings_final[,3]<-convertmax(hotel_bookings_final[,3])
176  boxplot.stats(hotel_bookings_final$lead_time)
177  boxplot(hotel_bookings_final$lead_time, horizontal = TRUE)
178  sum(is_outlier(hotel_bookings_final$lead_time))
```

```
$stats
[1]   0  18  69 160 373

$n
[1] 119385

$conf
[1] 68.35066 69.64934

$out
numeric(0)
```



```r
#adr
#Reemplazamos los valores por la moda
#ANTES DEL REEMPLAZO:
boxplot.stats(hotel_bookings_final$adr)
boxplot(hotel_bookings_final$adr, horizontal = TRUE)
sum(is_outlier(hotel_bookings_final$adr))
```

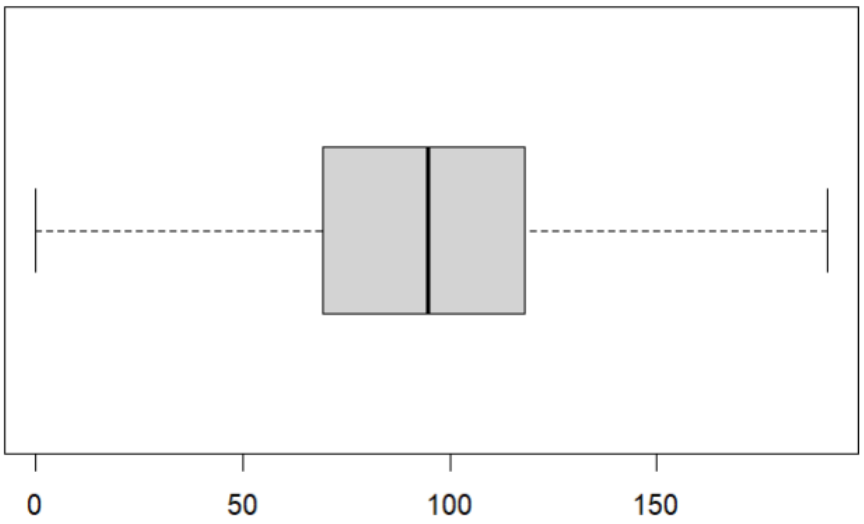
```
$stats
[1]  -6.38  69.29  94.59 126.00 211.03

$n
[1] 119385

$conf
[1] 94.33068 94.84932

$out
  [1] 225.00 213.75 230.67 216.13 249.00 241.50 214.00 214.00 240.64 217.05 233.00 222.67 240.00
 [14] 233.05 219.43 240.00 250.33 280.74 219.50 214.00 252.00 233.00 237.00 222.14 220.55 221.00
 [27] 230.50 230.50 241.00 242.60 268.00 217.20 239.30 267.00 226.00 277.50 221.00 250.00 211.75
 [40] 246.00 252.00 276.43 228.00 211.50 277.00 214.00 254.00 221.00 233.00 214.00 241.00 274.93
 [53] 252.00 258.33 255.00 243.00 222.20 243.00 266.40 236.00 271.00 232.00 223.00 229.00 266.00
 [66] 262.00 234.00 242.50 248.00 299.33 218.00 248.00 223.99 214.00 225.90 213.00 213.00 236.00
 [79] 229.67 239.50 220.40 241.00 241.00 241.00 222.07 229.00 213.50 236.00 248.00 260.71 222.00
 [92] 221.43 218.50 259.00 229.00 233.00 231.60 261.40 219.33 212.83 221.20 332.00 270.00 276.60
[105] 232.00 214.00 220.00 225.00 220.00 220.00 272.00 222.00 219.00 219.00 224.00 260.00 238.63
[118] 225.80 235.00 231.43 237.33 280.00 215.33 212.14 236.67 212.00 240.00 240.00 224.00 224.00
[131] 212.00 239.00 227.00 242.00 250.00 237.00 233.00 252.00 287.00 259.00 212.00 227.00 247.00
[144] 252.00 240.00 240.00 240.00 216.00 212.00 226.00 259.00 224.43 240.00 223.00 212.00 227.00
[157] 288.00 222.33 226.00 226.00 226.00 226.00 262.00 222.00 226.00 243.32 259.00 219.80 241.00
[170] 222.00 292.00 259.00 266.50 253.57 241.00 240.00 232.00 259.00 212.14 256.50 212.00 252.00
[183] 244.50 282.00 250.00 240.00 221.00 219.00 283.32 231.00 231.00 272.70 221.00 230.00 240.00
[196] 231.00 246.00 241.00 236.00 259.00 233.00 221.00 299.00 245.67 248.75 248.89 298.00 289.00
[209] 262.00 251.00 213.00 224.00 274.00 230.00 299.00 241.00 273.00 269.00 269.00 254.00 259.00
[222] 236.71 259.00 243.63 231.00 219.00 243.63 369.00 262.00 278.60 246.50 271.00 216.00 234.00
[235] 222.50 218.00 240.00 240.00 216.00 254.31 240.00 261.50 240.00 240.00 214.00 246.00 259.00
[248] 224.00 224.00 225.67 219.00 231.50 226.50 231.50 251.00 241.00 256.00 216.00 259.00 291.00
[261] 241.00 249.00 219.00 225.67 221.00 219.00 225.67 214.00 251.50 234.60 234.00 279.00 241.00
[274] 259.00 241.00 254.00 277.67 299.00 227.92 258.00 247.67 269.00 263.00 235.57 309.00 289.90
[287] 241.00 230.00 216.00 248.16 261.00 214.99 217.14 236.67 236.50 259.00 216.50 219.00 219.00
[300] 256.00 219.00 246.00 231.00 221.00 246.00 231.00 221.00 229.00 219.00 217.33 230.00 219.00
[313] 314.50 266.50 258.00 212.00 216.50 286.79 219.00 281.00 239.00 219.00 239.00 231.00 219.00
[326] 274.00 219.00 227.92 214.00 214.00 275.00 237.00 222.00 222.00 222.00 237.00 247.33 237.00
[339] 256.75 222.00 288.00 222.00 251.86 238.16 226.14 212.00 259.00 234.00 274.00 227.00 304.00
[352] 286.00 329.00 231.00 231.00 231.00 235.67 281.00 214.00 226.73 251.73 274.00 219.00 231.00
[365] 249.00 229.00 271.00 322.00 241.00 287.00 224.00 239.00 239.10 219.00 248.00 220.49 269.00
[378] 229.00 214.60 214.00 214.00 226.50 265.67 249.50 240.00 249.50 262.00 253.00 232.25 322.00
[391] 269.00 234.00 221.00 240.00 243.80 213.80 241.75 247.57 221.00 246.67 246.67 231.80 231.80
[404] 223.29 216.00 234.00 227.10 221.00 240.60 252.00 219.00 264.00 216.00 219.00 254.00 246.00
[417] 221.07 292.40 212.86 233.00 217.00 246.02 340.00 384.00 250.00 302.11 250.00 382.00 275.00
[430] 229.40 275.00 243.00 223.00 213.00 213.00 225.50 228.00 228.00 262.50 223.00 212.00 213.33
[443] 215.00 223.00 228.00 228.00 233.00 260.00 212.00 238.00 273.00 261.00 248.00 248.00 311.00
[456] 212.00 248.00 258.00 228.00 288.00 228.00 222.00 222.00 238.00 216.00 248.00 265.00 260.00
[469] 220.00 250.00 243.00 217.50 238.00 300.86 292.00 238.71 243.71 232.00 259.86 242.00 223.57
[482] 219.14 241.00 264.00 247.00 241.00 265.00 274.67 270.00 239.00 303.00 242.75 212.00 212.00
[495] 293.00 222.00 222.00 232.00 275.00 260.00 230.86 218.00 270.00 260.00 219.50 272.00 230.30
[508] 215.00 219.00 293.33 253.25 311.00 217.71 240.00 233.10 289.60 214.00 240.00 252.00 252.00
```



```
> sum(is_outlier(hotel_bookings_final$adr))
[1] 3793
```

```
186 #DESPUES DEL REEMPLAZO:
187 hotel_bookings_final$adr<-convert(hotel_bookings_final$adr)
188 boxplot.stats(hotel_bookings_final$adr)
189 boxplot(hotel_bookings_final$adr, horizontal = TRUE)
190 sum(is_outlier(hotel_bookings_final$adr))
```

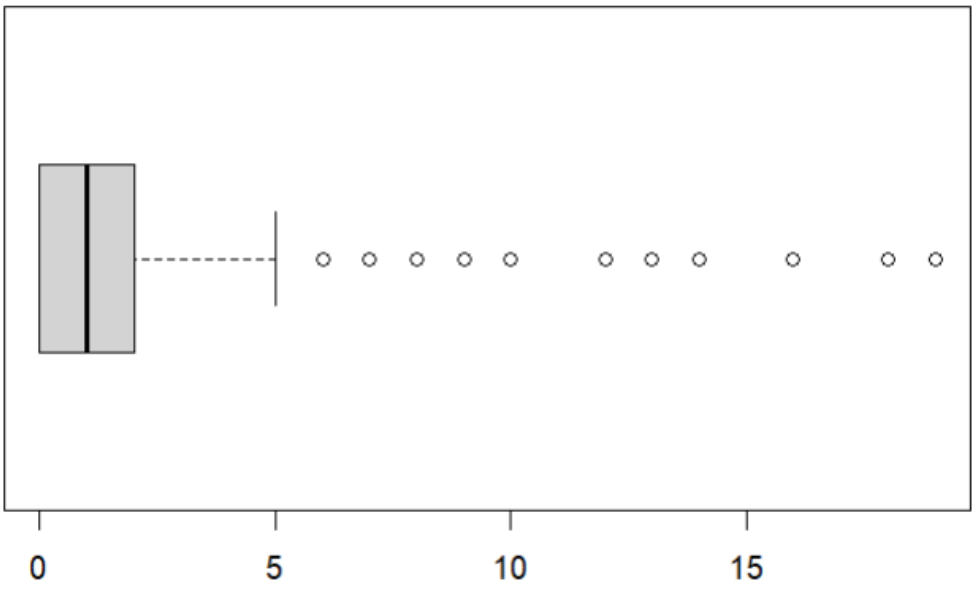
```
$stats
[1]   0.00  69.29  94.59 118.00 191.00

$n
[1] 119385

$conf
[1] 94.36726 94.81274

$out
numeric(0)
```



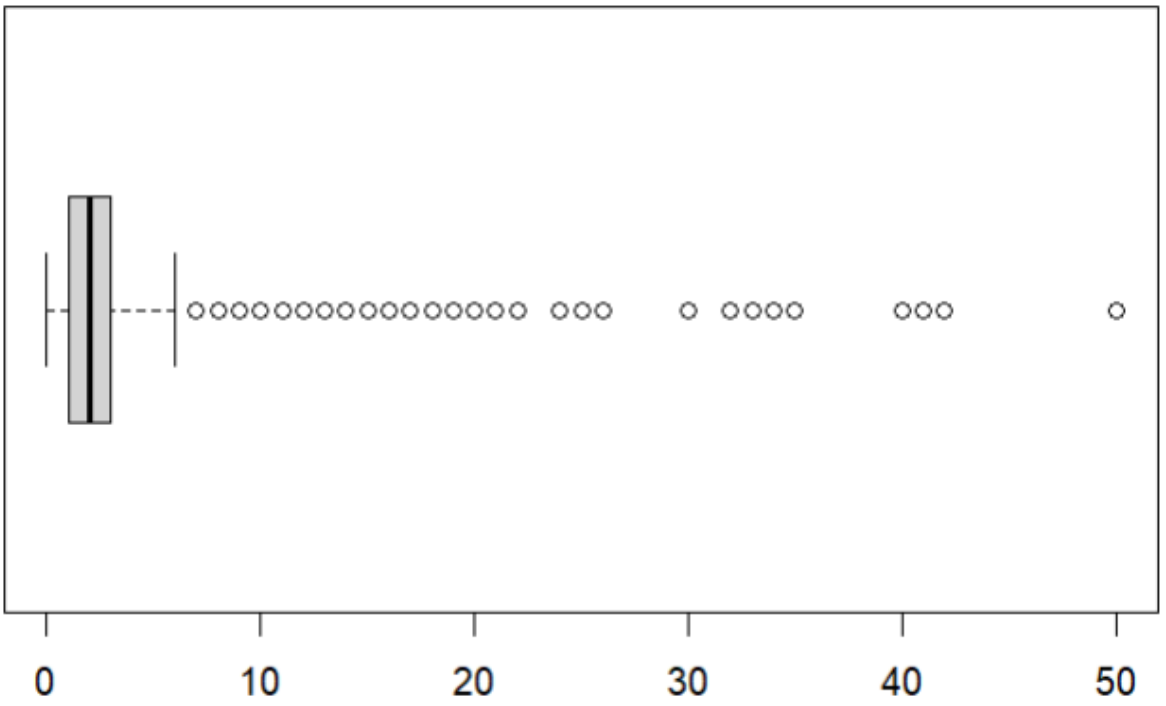

```
> sum(is_outlier(hotel_bookings_final$adr))
[1] 0
```

Además podemos ver también los diagramas de cajas de las demas variables:

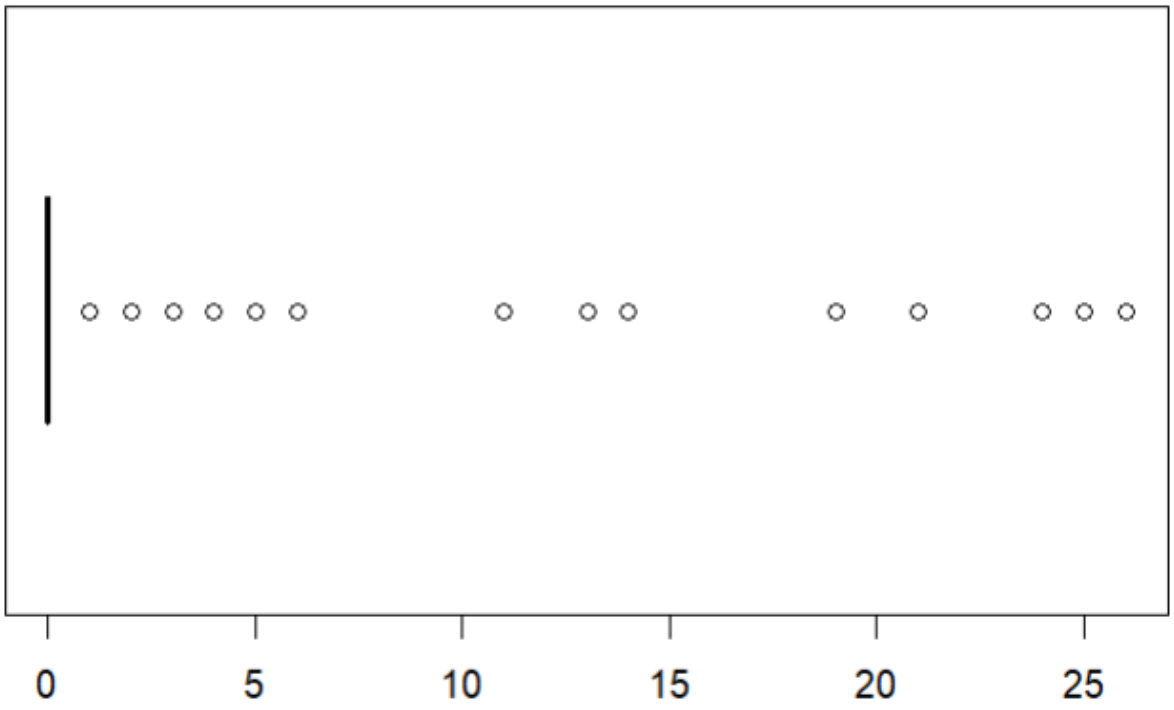
```
184 boxplot(hotel_bookings_final$stays_in_weekend_nights, horizontal = TRUE)
```




```
184 boxplot(hotel_bookings_final$stays_in_week_nights, horizontal = TRUE)
```




```
184 boxplot(hotel_bookings_final$previous_cancellations, horizontal = TRUE)
```



En las primeras 2 situaciones podemos ver que somos capaces de actualizar estos valores atípicos con facilidad, esto porque primero creamos una función equivalente a is.na() pero que detecte si algún valor es atípico: is_outlier, función hecha con la fórmula de tukey,
Valor atípico (outlier) = Valor < Q1 - 1.5 * (Q3 - Q1) o Valor > Q3 + 1.5 * (Q3 - Q1)
,lo cual nos permite recorrer cada columna y contar la cantidad de valores atípicos en estas. Además de la funcion is_outlier también creamos y usamos la función convert() y convertmax() que reemplazan los valores atípicos por la moda y por el valor en el límite superior, respectivamente.

A pesar de la gran cantidad de valores atípicos, se trata de una característica en los datos de los registros de hoteles.

Concluimos, pues, que no necesitamos actuar sobre los demás valores atípicos de nuestro dataset.

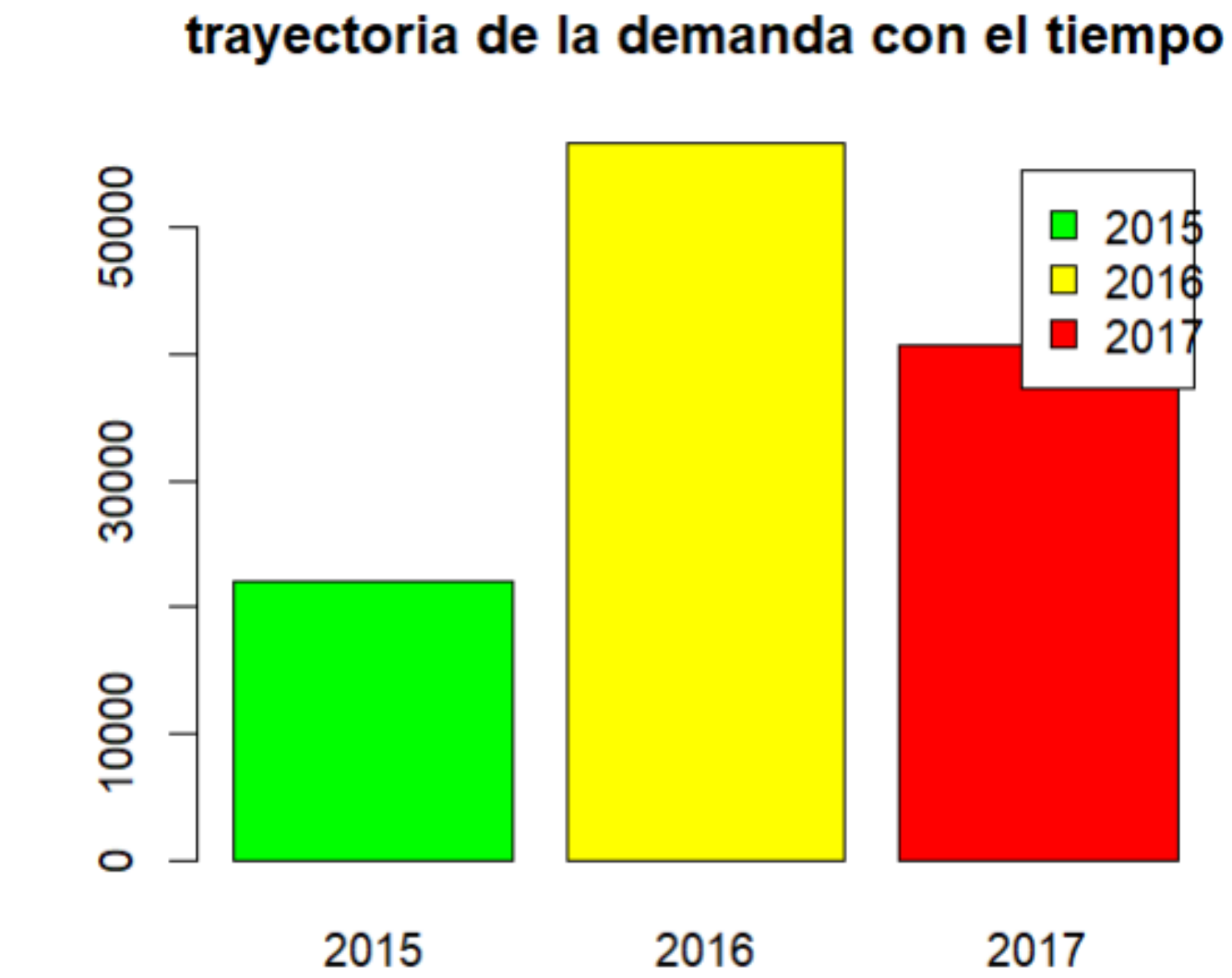
4. Visualizar datos

```
#¿Cuantas reservas se realizan por tipo de hotel?
hotel_bookings_final$hotel<-as.factor(hotel_bookings_final$hotel)
resumen_hotel<-summary(hotel_bookings_final$hotel)
barplot(resumen_hotel, col=c("green","yellow"), legend = c("City Hotel", "Resort Hotel"),
        main = "reservas por tipo de hotel", names= c("City Hotel", "Resort Hotel") )
summary(hotel_bookings_final$hotel)
```


```
> summary(hotel_bookings_final$hotel)
  City Hotel Resort Hotel
       79326        40059
```

Podemos ver que se realizaron 79326 reservas a city hotel y 40059 a resort hotel, teniendo la delantera city hotel.

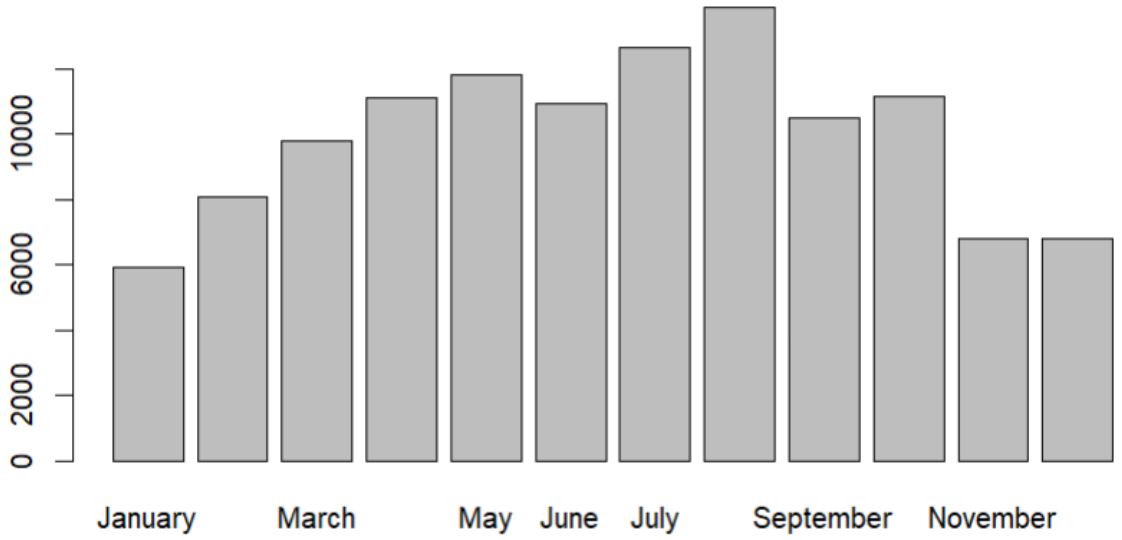
```r
#Esta aumentando la demanda con el tiempo?
hotel_bookings_final$arrival_date_year<-as.factor(hotel_bookings_final$arrival_date_year)
resumen_anios<-summary(hotel_bookings_final$arrival_date_year)
barplot(resumen_anios, col=c("green","yellow","red"), legend = c("2015", "2016","2017"),
        main = "trayectoria de la demanda con el tiempo", names= c("2015", "2016","2017") )
resumen_anios
```




```
> resumen_anios
 2015  2016  2017
21991 56707 40687
```

Podemos ver que en comparación con 2015 hubo un incremento significativo de reservas, en 2016 tuvo su máximo de registros y en 2017 estuvo bastante bien igual. Podemos decir que en general si aumentó la demanda con el tiempo.

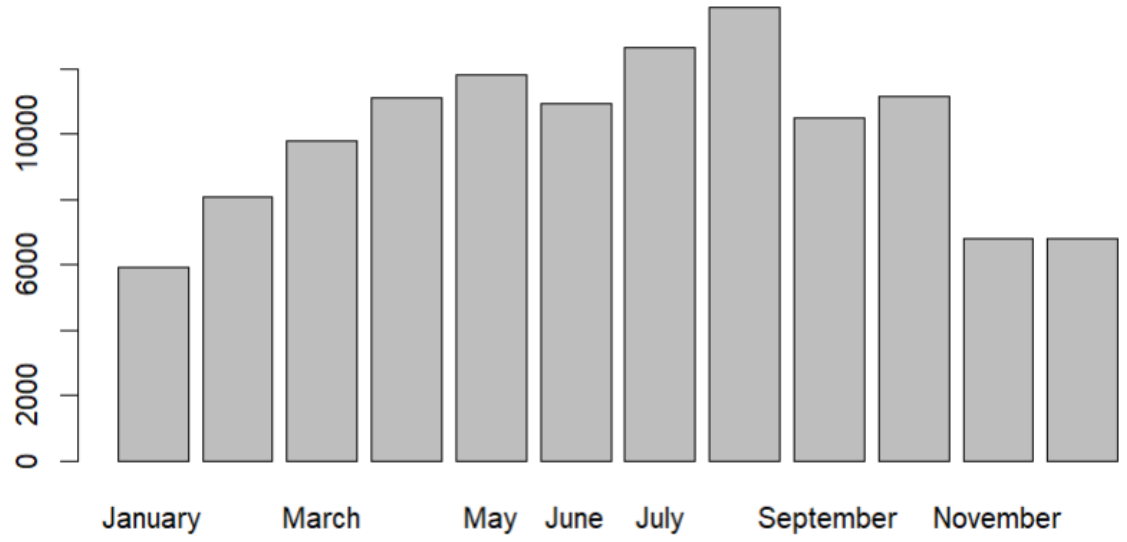
```r
#¿Cuándo se producen las temporadas de reservas: alta, media y baja?
orden_meses <- c("January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December")
hotel_bookings_final$arrival_date_month <- factor(hotel_bookings_final$arrival_date_month, levels = orden_meses)
resumen_mes<-summary(hotel_bookings_final$arrival_date_month)
barplot(resumen_mes)
summary(hotel_bookings_final$arrival_date_month)
```




```
> summary(hotel_bookings_final$arrival_date_month)
  January  February     March     April       May      June      July    August September   October  November
     5929      8068      9794     11089     11791     10939     12660     13873     10508     11160      6794
 December
     6780
```

Podemos decir que Primavera e invierno representan las temporadas de reservas bajas, verano representa una temporada de reservas media y Otoño una temporada de reservas alta.

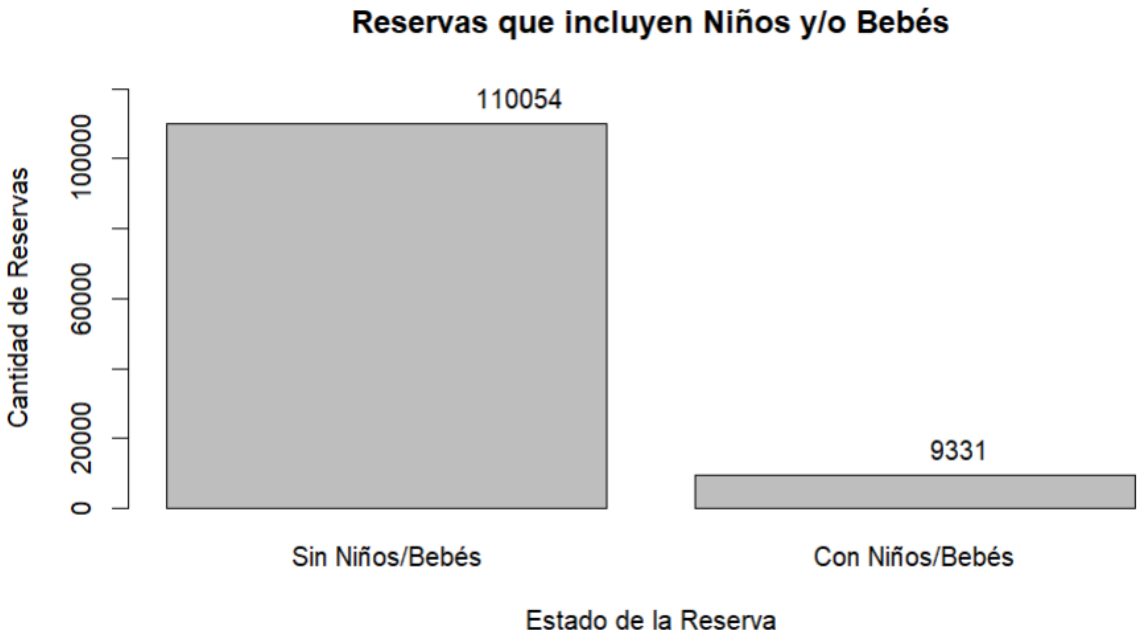
```r
#¿Cuándo es menor la demanda de reservas?
orden_meses <- c("January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December")
hotel_bookings_final$arrival_date_month <- factor(hotel_bookings_final$arrival_date_month, levels = orden_meses)
resumen_mes<-summary(hotel_bookings_final$arrival_date_month)
barplot(resumen_mes)
summary(hotel_bookings_final$arrival_date_month)
```




```
> summary(hotel_bookings_final$arrival_date_month)
  January  February     March     April       May      June      July    August September   October  November
     5929      8068      9794     11089     11791     10939     12660     13873     10508     11160      6794
 December
     6780
```

Enero es el mes donde se realizan menos reservas de todo el año.

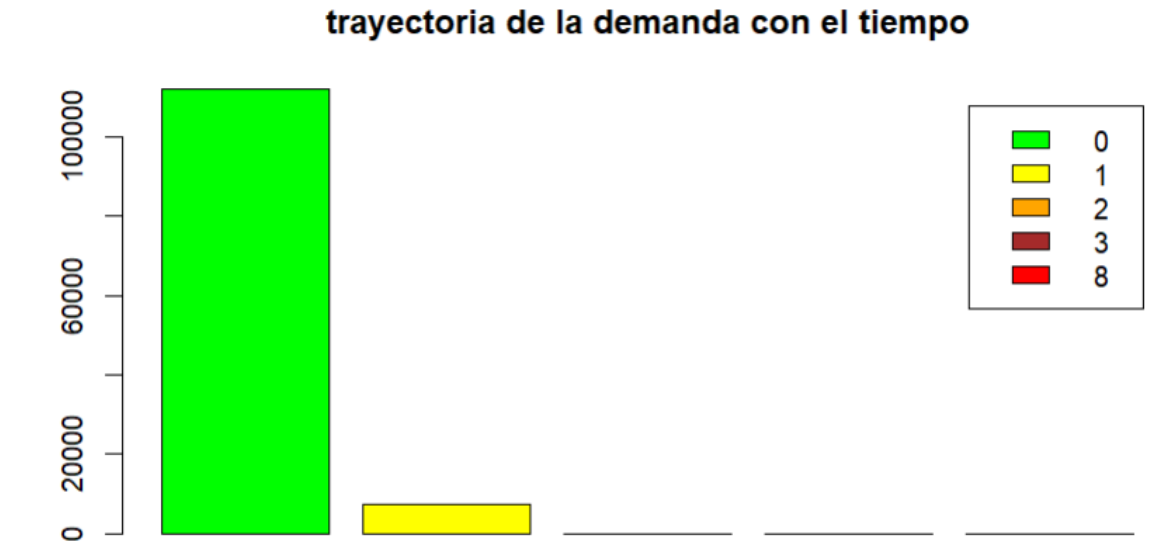
```r
#¿Cuántas reservas incluyen niños y/o bebes?
# Calcular la frecuencia de reservas con niños y/o bebés
reservas_con_ninos <- table(hotel_bookings_final$babies > 0 | hotel_bookings_final$children > 0)
names(reservas_con_ninos)<-c("Sin Niños/Bebés", "Con Niños/Bebés")
# Crear el gráfico de barras
barplot(reservas_con_ninos,
        names.arg = c("Sin Niños/Bebés", "Con Niños/Bebés"),
        main = "Reservas que incluyen Niños y/o Bebés",
        xlab = "Estado de la Reserva",
        ylab = "Cantidad de Reservas",
        ylim = c(0, max(reservas_con_ninos) * 1.1))  # Establecer un rango de ejes y

text(x = 1:2, y = reservas_con_ninos + 50, labels = reservas_con_ninos, pos = 3)
names(reservas_con_ninos)<-c("Sin Niños/Bebés", "Con Niños/Bebés")
reservas_con_ninos
```




```
Sin Niños/Bebés Con Niños/Bebés
         110054           9331
```

Podemos ver que solo 9331 de 119385 registros son con niños o bebes.

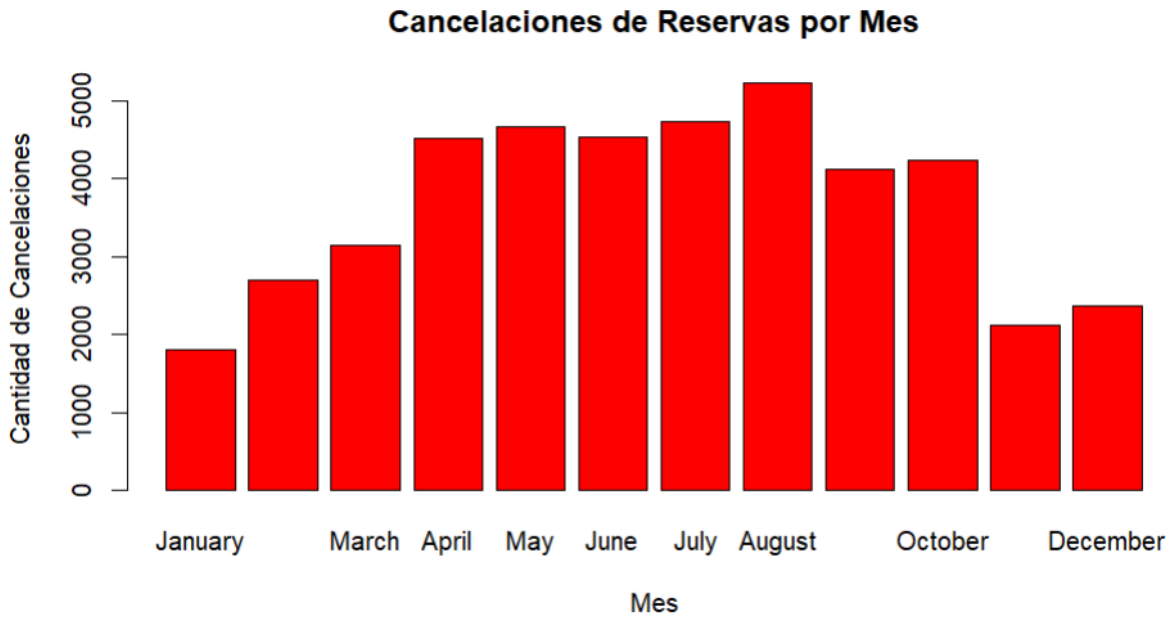
```r
#¿Es importante contar con espacios de estacionamiento?
options(scipen = 999)
hotel_bookings_final$required_car_parking_spaces<-as.factor(hotel_bookings_final$required_car_parking_spaces)
resumen_estac<-summary(hotel_bookings_final$required_car_parking_spaces)
barplot(resumen_estac, col=c("green","yellow","orange","brown","red"), legend = c("0", "1","2","3","8"),
        main = "trayectoria de la demanda con el tiempo", names= c("0", "1","2","3","8"))
resumen_estac
```




```
     0      1      2      3      8
111970   7382     28      3      2
```

Segun los datos no es importante porque la gran mayoria no usa coches para realizar los registros ya sea por ser online o diferentes motivos, de los registros totales solo 7382 tuvieron 1 coche representando solo el 6.2% del total, 28 personas tuvieron 2 coches, 3 personas usaron 3 vehiculos y solo 2 usaron 8.

```r
#¿En qué meses del año se producen más cancelaciones de reservas?
# Calcular la cantidad de cancelaciones por mes
cancelaciones_por_mes <- aggregate(is_canceled ~ arrival_date_month, data = hotel_bookings_final, FUN = sum)

# Ordenar los meses cronológicamente
meses_ordenados <- c("January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December")
cancelaciones_por_mes$arrival_date_month <- factor(cancelaciones_por_mes$arrival_date_month, levels = meses_ordenados)

# Crear el gráfico de barras
barplot(cancelaciones_por_mes$is_canceled,
        names.arg = cancelaciones_por_mes$arrival_date_month,
        col = "red",
        main = "Cancelaciones de Reservas por Mes",
        xlab = "Mes",
        ylab = "Cantidad de Cancelaciones")
cancelaciones_por_mes
```




```
> cancelaciones_por_mes
   arrival_date_month is_canceled
1             January        1807
2            February        2696
3               March        3149
4               April        4524
5                 May        4677
6                June        4535
7                July        4742
8              August        5235
9           September        4116
10            October        4246
11           November        2122
12           December        2371
```

Entre verano y otoño se presentan mas cancelaciones, desde Agosto a Octubre se mantiene a cierto nivel constante.

5. Conclusiones preliminares

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

Observamos que se realizaron un total de 79,326 reservas en hoteles de ciudad (City Hotel) y 40,059 en hoteles resort (Resort Hotel). La mayoría de las reservas se hicieron en hoteles de ciudad, lo que sugiere una preferencia general por este tipo de alojamiento.
b. ¿Está aumentando la demanda con el tiempo?

A través del análisis temporal, notamos que la demanda de reservas experimentó un incremento significativo en comparación con el año 2015. El año 2016 registró el mayor número de reservas, y aunque hubo ciertas variaciones, en general, se puede concluir que la demanda ha aumentado con el tiempo.
c. ¿Cuándo se producen las temporadas de reservas: alta, media y baja?

Identificamos que la temporada de reservas baja se encuentra en primavera e invierno, mientras que la temporada de reservas media es durante el verano. La temporada de reservas alta se da en otoño. Esta información es esencial para la planificación de la ocupación y la gestión hotelera.
d. ¿Cuándo es menor la demanda de reservas?

En enero observamos que se realizan menos reservas en comparación con otros meses del año. Esto puede estar relacionado con las festividades de fin de año y las vacaciones de invierno.
e. ¿Cuántas reservas incluyen niños y/o bebés?

Constatamos que de un total de 119,385 registros, solo 9,331 incluyen niños o bebés. Esto indica que la mayoría de las reservas se realizan para adultos sin niños o bebés, lo que puede ser valioso para la planificación de servicios y alojamiento.
f. ¿Es importante contar con espacios de estacionamiento?

Basándonos en los datos, podemos afirmar que no es esencial contar con espacios de estacionamiento en los hoteles. Solo el 6.2% del total de registros incluyó un vehículo, y la mayoría de los huéspedes no optaron por esta opción.
g. ¿En qué meses del año se producen más cancelaciones de reservas?

En los meses de verano y otoño se observan más cancelaciones de reservas, especialmente entre agosto y octubre. Esta información es valiosa para anticipar fluctuaciones en la demanda y adaptar las estrategias de gestión hotelera en consecuencia.
Estas conclusiones proporcionan una visión general de las tendencias y patrones identificados en los datos del conjunto 'hotel_booking.csv'. Son valiosas para la toma de decisiones y la planificación en la industria hotelera, así como para comprender mejor el comportamiento de los huéspedes y las temporadas de demanda.

enlace del GITHUB:
https://github.com/AnthonyConH/-CC216-TP-2023-2-CC51/tree/main