TRABAJO ACADÉMICO GRUPAL - TA1 #3

ADMINISTRACIÓN DE LA INFORMACIÓN

Docente: Reyes Silva, Patricia Daniela

Sección: CC51

Grupo: 2

Integrantes:

Galindo Alvarez, Franco - U202010807

Goyas Ayllon, Leonardo Andre - U202010206

Villafuerte Ramirez, Diego Tomas - U202010546

Índice

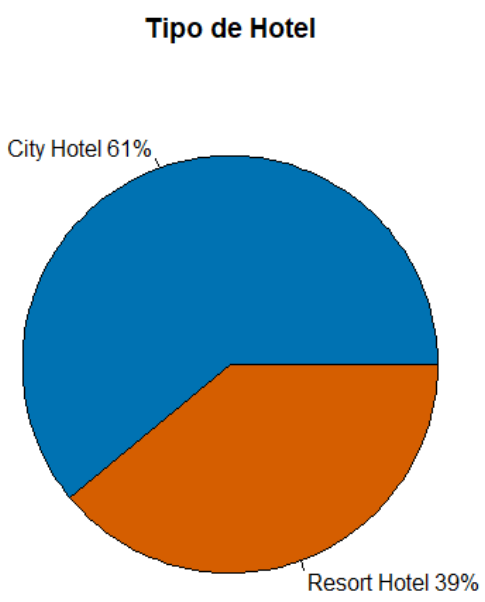
1. Caso de Análisis

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

Los datos de tipo “hotel” se dividen en City Hotel y Resort Hotel. Podemos observar en la siguiente gráfica que el porcentaje de la primera es mayor que el de la segunda. Hubo en total 52979 reservas, mientras que los de resort fueron 33848 reservas. El análisis muestra que la cantidad de reservas de City Hotel fueron 56.52% más que las de Resort Hotel, por lo que podemos concluir que la gente tiende a reservar más en los hoteles de ciudad.

```
> table(hotel_data_pre3$hotel)

  City Hotel Resort Hotel
       52979        33848
```



Data sin outliers

```
table(hotel_data_pre3$hotel)
num_city <- length(hotel_data_pre3$hotel[hotel_data_pre3$hotel ==
"City Hotel"])
num_city
num_resort <- length(hotel_data_pre3$hotel[hotel_data_pre3$hotel
== "Resort Hotel"])
num_resort
nums <- c(num_city, num_resort)
porcentajes <- round(nums/sum(nums)*100)
labels = levels(hotel_data_pre3$hotel)
labels <- paste(labels, porcentajes, sep = " ")
labels <- paste(labels, "%", sep = "")
pie(nums, labels = labels,
    main = "Tipo de Hotel",
    sub = "Data sin outliers",
    col = c("#0072B2", "#D55E00"))
num_city - num_resort
mayor <- num_city/num_resort * 100
mayor - 100
```

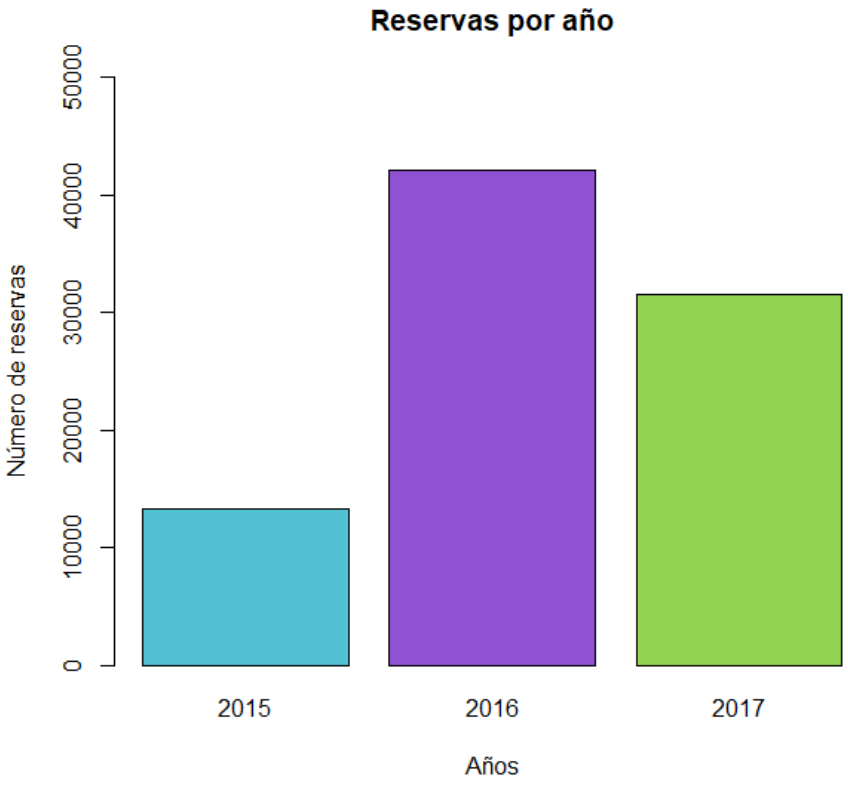
b. ¿Está aumentando la demanda con el tiempo?

El análisis fue realizado tomando en cuenta la frecuencia de las reservas según los años del data frame. El tiempo va del año 2015 al 2017, y podemos ver que aumenta del primer al segundo año y luego se reduce en el tercer año. Las frecuencias son de 13236, 42096 y 31495 respectivamente. El primer cambio es un aumento del 218.04% aproximado, mientras que el segundo cambio es de una reducción aproximada del 25.18% de las reservas anuales.

Podemos concluir que las reservas tuvieron un aumento notable respecto al año 2015, pero se encuentran en un estado de decrecimiento actualmente.

```
> table(hotel_data_pre3$arrival_date_year)

 2015  2016  2017
13235 42098 31494
```



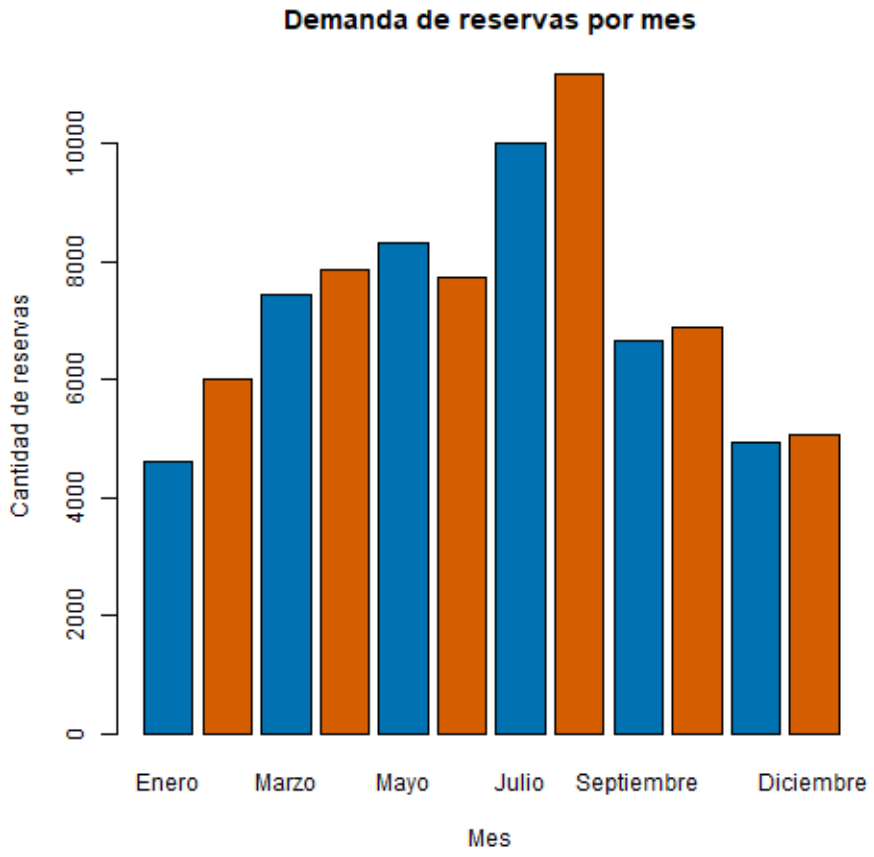

```
table(hotel_data_pre3$arrival_date_year)
labels = as.character(unique(hotel_data_pre3$arrival_date_year))
num15 <- sum(hotel_data_pre3$arrival_date_year == 2015)
num16 <- sum(hotel_data_pre3$arrival_date_year == 2016)
num17 <- sum(hotel_data_pre3$arrival_date_year == 2017)
nums <- c(num15, num16, num17)
barplot(names = labels, height = nums,
        main = "Reservas por año",
        col = c("#52C1D3", "#8F52D3", "#93D352"),
        xlab = "Años", ylab = "Número de reservas",
        ylim = c(0, 50000))
cambio1 <- num16/num15 * 100
cambio1 - 100

cambio2 <- num17/num16 * 100
cambio2 - 100
```

c. ¿Cuándo es menor la demanda de reservas?

La base de datos posee el mes en el cual las personas han hecho la reserva llamada `arrival_date_month`, separada en 12 niveles, cada uno representando un mes del año. Por lo que si obtenemos su tabla, podemos representar la cantidad de frecuencia de cada mes. Donde observamos que la menor cantidad de demandas se da entre finales e inicios de año.

```
barplot(
    table(hotel_data_pre3$arrival_date_month),
    col = c("#0072B2", "#D55E00"),
    main = "Demanda de reservas por mes",
    xlab = "Mes",
    ylab = "Cantidad de reservas",
    names = c("Enero", "Febrero", "Marzo",
     "Abril", "Mayo", "Junio", "Julio",
     "Agosto", "Septiembre", "Octubre",
     "Noviembre", "Diciembre")
)
```



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

El análisis se dio acabo hallando todas las entradas que poseen más de un niño o un bebé, por lo cual se emplearon las variables `children` y `babies`. Para a continuación comparar su cantidad con la del total de entradas.

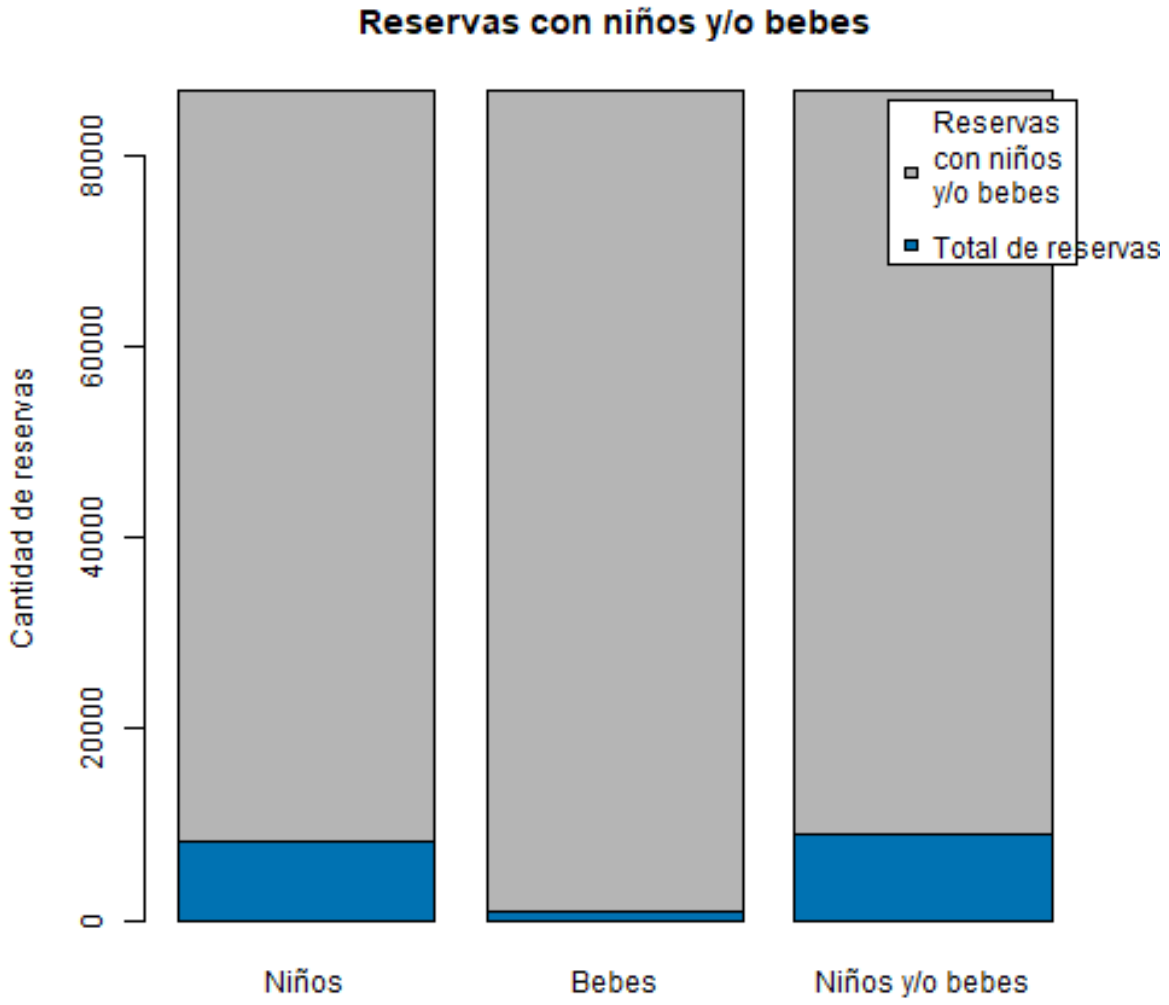
```
children_rows <- nrow(hotel_data_pre3[hotel_data_pre3$children > 0,])
babies_rows <- nrow(hotel_data_pre3[hotel_data_pre3$babies > 0,])
children_babies_rows <- nrow(hotel_data_pre3[hotel_data_pre3$children > 0 | hotel_data_pre3$babies > 0,])
total_rows <- nrow(hotel_data_pre3)

matrix_comp <- matrix(c(
    children_rows, total_rows - children_rows,
    babies_rows, total_rows - babies_rows,
    children_babies_rows, total_rows - children_babies_rows),
    ncol = 3, byrow = FALSE
)
```

```
      [,1]  [,2]  [,3]
[1,]  8135   910  8873
[2,] 78692 85917 77954
```

Como se puede observar en la matriz, siendo la primera fila la cantidad de reservas con niños y/o bebés, y la segunda, la cantidad restante de reservas. Concluimos que la cantidad de reservas que incluyen niños y/o bebes.

```
barplot(
    matrix_comp,
    col = c("#0072B2", "#b5b5b5"),
    legend = c("Total de reservas", "Reservas\ncon niños\ny/o
bebes\n"),
    main = "Reservas con niños y/o bebes",
    ylab = "Cantidad de reservas",
    names = c("Niños", "Bebes", "Niños y/o bebes"),
)
```



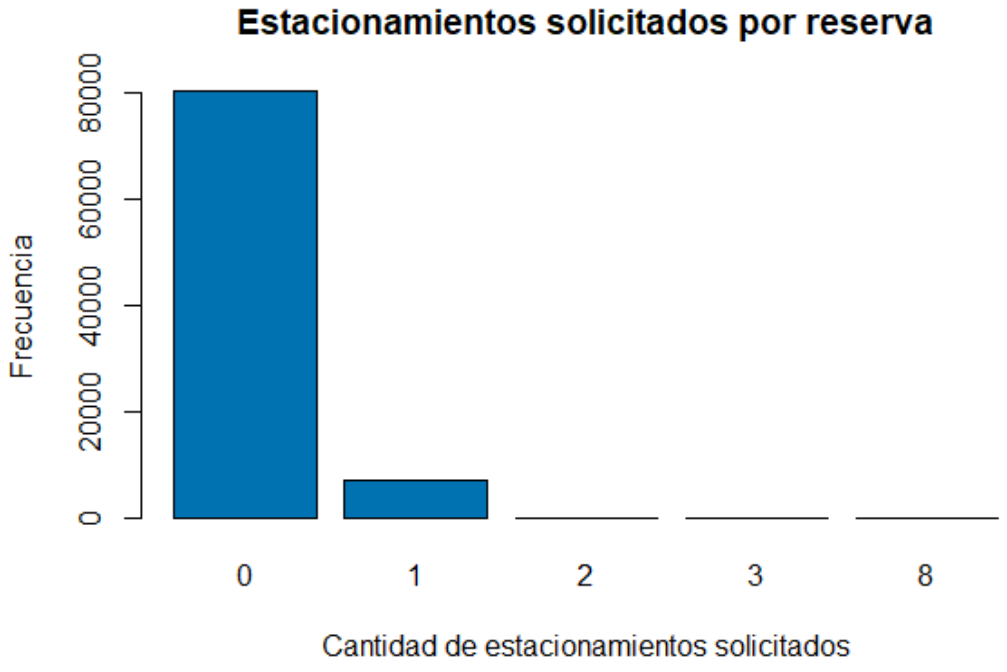
e. ¿Es importante contar con espacios de estacionamiento?

Para el análisis se realizó gracias a la variable `required_car_parking_spaces` para el conteo de los parqueos que se solicitan por reserva.

```
barplot(
  table(hotel_data_pre3$required_car_parking_spaces),
  col = c("#0072B2"),
  main = "Estacionamientos solicitados por reserva",
  xlab = "Cantidad de estacionamientos solicitados",
  ylab = "Frecuencia",
)
```

```
 [0]    [1]    [2]   [3]   [8]
79530  7266    26     3     2
```

Como se puede observar, la cantidad de reservas que piden estacionamientos es pequeña, abarcando solo el 8,4% del total de las reservas. Sin embargo, no deja de ser un factor importante para un grupo pequeño de personas.



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

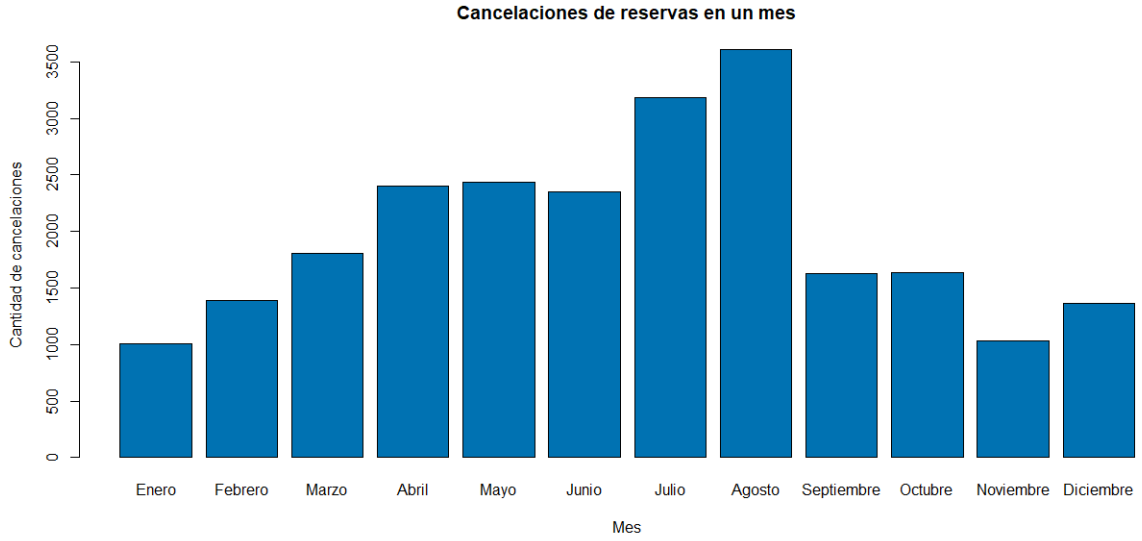
Para este caso primero se filtraron las entradas que se cancelaron, esto gracias a la variable `is_canceled` y se ordenaron los meses para la muestra de la tabla.

```
hotel_data_pre3_cancelados
<-c(hotel_data_pre3[hotel_data_pre3$is_canceled=="1",])
hotel_data_pre3_cancelados$arrival_date_month <-
ordered(hotel_data_pre3_cancelados$arrival_date_month, levels =
month.name)

table(hotel_data_pre3_cancelados$is_canceled,
hotel_data_pre3_cancelados$arrival_date_month)
```

	January	February	March	April	May	June	July	August	September	October	November	December
0	0	0	0	0	0	0	0	0	0	0	0	0
1	1006	1388	1809	2400	2437	2348	3182	3607	1629	1639	1035	1361

```
barplot(
  table(hotel_data_pre3_cancelados$is_canceled,
hotel_data_pre3_cancelados$arrival_date_month),
  col = c("#0072B2"),
  main = "Cancelaciones de reservas en un mes",
  xlab = "Mes",
  ylab = "Cantidad de cancelaciones",
  names = c("Enero","Febrero", "Marzo", "Abril", "Mayo", "Junio",
  "Julio", "Agosto", "Septiembre", "Octubre", "Noviembre",
"Diciembre")
)
```



Como se puede observar, el mes que tiene más reservas canceladas es Agosto, con una cantidad de 3607.

2. Conjunto de Datos (Data Set)

Para el presente trabajo, empleamos el conjunto de datos llamado **Hotel booking demand**, la cual se obtuvo de la página web Kaggle. Este dataset fue modificado para incorporar ruido en los datos, con valores faltantes (NA) y datos atípicos (outliers).

Estas son las variables que presentan:

Variable	Tipo	Descripción
ADR	Numérico	Tasa diaria promedio
Adults	Entero	Número de adultos
Agent	Categórico	ID de la agencia de viajes que realizó las reservas
ArrivalDateDayOfMonth	Entero	Día del mes de la fecha de llegada
ArrivalDateMonth	Categórico	Mes de fecha de llegada con 12 categorías: “Enero” a “Diciembre”
ArrivalDateWeekNumber	Entero	Número de semana de la fecha de llegada
ArrivalDateYear	Entero	Año de la fecha de llegada
AssignedRoomType	Categórico	Código del tipo de habitación asignado a la reserva. En ocasiones, el tipo de habitación asignado difiere del tipo de habitación reservado debido a razones de funcionamiento del hotel (por ejemplo, overbooking) o por solicitud del cliente. Se presenta el código en lugar de la designación por razones de anonimato
Babies	Entero	Número de bebés
BookingChanges	Entero	Número de cambios/modificaciones realizados en la reserva desde el momento en que se ingresa la reserva en el PMS hasta el momento del check-in o la cancelación
Children	Entero	Número de niños
Company	Categórico	ID de la empresa/entidad que realizó la reserva o responsable del pago de la reserva. Se presenta

		ID en lugar de designación por razones de anonimato
Country	Categórico	País de origen. Las categorías se representan en el formato ISO 3155–3:2013 [6]
CustomerType	Categórico	Tipo de reserva, asumiendo una de cuatro categorías: ● Contract – cuando la reserva tiene asociada una asignación u otro tipo de contrato ● Group – cuando la reserva está asociada a un grupo ● Transient – cuando la reserva no forma parte de un grupo o contrato, y no está asociada a otra reserva transitoria ● Transient-party – cuando la rserva es transitoria, pero está asociada al menos a otra reserva transitoria
DaysInWaitingList	Entero	Número de días que la reserva estuvo en lista de espera antes de ser confirmada al cliente
DepositType	Categórico	Indicación de si el cliente realizó un depósito para garantizar la reserva. Esta variable puede asumir tres categorías ● No Deposit – no se hizo ningún depósito ● Non Refund – se realizó un depósito por el valor del costo total de la estadía ● Refundable – se hizo un depósito con un valor por debajo del costo total de la estadía
DistributionChannel	Categórico	Canal de distribución de reservas. El término "TA" significa "Agentes de viajes" y "TO" significa "Operadores turísticos"
IsCanceled	Categórico	Valor que indica si la reserva fue cancelada (1) o no (0)
IsRepeatedGuest	Categórico	Valor que indica si el nombre de la reserva era de un huésped repetido (1) o no (0)
LeadTime	Entero	Número de días transcurridos entre la fecha de entrada de la reserva en el PMS y la fecha de llegada
MarketSegment	Categórico	Designación del segmento de mercado. En las categorías, el término "TA" significa "Agentes de viajes" y "TO" significa "Operadores turísticos"

Meal	Categórico	Tipo de comida reservada. Las categorías se presentan en paquetes estándar de comidas de hospitalidad: ● Undefined/SC – sin paquete de comidas ● BB – cama y Desayuno ● HB – media pensión (desayuno y otra comida, normalmente cena) ● FB – pensión completa (desayuno, comida y cena)
PreviousBookingsNotCanceled	Entero	Número de reservas anteriores no canceladas por el cliente antes de la reserva actual
PreviousCancellations	Entero	Número de reservas anteriores que fueron canceladas por el cliente antes de la reserva actual
RequiredCardParkingSpaces	Entero	Número de plazas de aparcamiento requeridas por el cliente
ReservationStatus	Categórico	Último estado de la reserva, asumiendo una de las tres categorías: ● Canceled – la reserva fue cancelada por el cliente ● Check-Out – el cliente se ha registrado pero ya se ha ido ● No-Show – el cliente no se registró e informó al hotel del motivo
ReservationStatusDate	Date	Fecha en la que se estableció el último estado. Esta variable se puede usar junto con ReservationStatus para comprender cuándo se canceló la reserva o cuándo se retiró el cliente del hotel.
ReservedRoomType	Categórico	Código del tipo de habitación reservado. Se presenta el código en lugar de la designación por razones de anonimato
StaysInWeekendNights	Entero	Número de noches de fin de semana (sábado o domingo) que el huésped se alojó o reservó para quedarse en el hotel
StaysInWeekNights	Entero	Número de noches de la semana (de lunes a viernes) que el huésped se hospedó o reservó para quedarse en el hotel
TotalOfSpecialRequests	Entero	Número de solicitudes especiales realizadas por el cliente (por ejemplo, cama doble o piso alto)

3. Análisis Exploratorio de Datos

Cargar Datos

Para cargar el dataset, empleamos la función `read.csv()`, considerando los parámetros, `header = TRUE`, `sep = ","`, `stringAsFactors = FALSE`.

```
#Cargando el dataset de reservas de hotel
hotel_data <- read.csv("data/hotel_bookings_miss.csv", header =
TRUE, sep = ",", stringsAsFactors = FALSE)
```

Inspeccionar Datos

El dataset presenta las siguientes características.

```
 #Identify variable names
    names(hotel_data)
 [1] "ï..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"
```

```
r$> #View the data types
    str(hotel_data)
'data.frame':   119390 obs. of  32 variables:
 $ ï..hotel                      : Factor w/ 2 levels "City
Hotel","Resort Hotel": 2 2 2 2 2 2 2 2 2 2 ...
 $ 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" ...

r$> #Summarize each attribute
    summary(hotel_data)
         ï..hotel      is_canceled       lead_time
arrival_date_year arrival_date_month
 City Hotel  :79330   Min.   :0.0000   Min.   :  0   Min.   :2015
Length:119390
 Resort Hotel:40060   1st Qu.:0.0000   1st Qu.: 18   1st Qu.:2016
Class :character
                      Median :0.0000   Median : 69   Median :2016
Mode  :character
                      Mean   :0.3704   Mean   :104   Mean   :2016
                      3rd Qu.:1.0000   3rd Qu.:160   3rd Qu.:2017
                      Max.   :1.0000   Max.   :737   Max.   :2017
                                       NA's   :21    NA's   :6
 arrival_date_week_number arrival_date_day_of_month
stays_in_weekend_nights stays_in_week_nights
 Min.   : 1.00            Min.   : 1.0              Min.   :
0.0000         Min.   : 0.0
 1st Qu.:16.00            1st Qu.: 8.0              1st Qu.:
0.0000         1st Qu.: 1.0
 Median :28.00            Median :16.0              Median :
1.0000         Median : 2.0
 Mean   :27.16            Mean   :15.8              Mean   :
0.9275         Mean   : 2.5
 3rd Qu.:38.00            3rd Qu.:23.0              3rd Qu.:
2.0000         3rd Qu.: 3.0
 Max.   :53.00            Max.   :31.0              Max.
:19.0000         Max.   :50.0
 NA's   :25               NA's   :7                 NA's   :25
NA's   :12
     adults          children          babies             meal
country
 Min.   : 0.000   Min.   : 0.0000   Min.   : 0.00000
```

```
Length:119390      Length:119390
 1st Qu.: 2.000   1st Qu.: 0.0000   1st Qu.: 0.00000   Class
:character   Class :character
 Median : 2.000   Median : 0.0000   Median : 0.00000   Mode
:character   Mode  :character
 Mean   : 1.856   Mean   : 0.1039   Mean   : 0.00795
 3rd Qu.: 2.000   3rd Qu.: 0.0000   3rd Qu.: 0.00000
 Max.   :55.000   Max.   :10.0000   Max.   :10.00000
 NA's   :12       NA's   :4         NA's   :32
 market_segment     distribution_channel is_repeated_guest
previous_cancellations
 Length:119390      Length:119390        Min.   :0.00000   Min.
: 0.00000
 Class :character   Class :character     1st Qu.:0.00000   1st
Qu.: 0.00000
 Mode  :character   Mode  :character     Median :0.00000   Median
: 0.00000
                                        Mean   :0.03191   Mean
: 0.08712
                                        3rd Qu.:0.00000   3rd
Qu.: 0.00000
                                        Max.   :1.00000   Max.
:26.00000

 previous_bookings_not_canceled reserved_room_type
assigned_room_type booking_changes
 Min.   : 0.0000                Length:119390      Length:119390
Min.   : 0.0000
 1st Qu.: 0.0000                Class :character   Class
:character   1st Qu.: 0.0000
 Median : 0.0000                Mode  :character   Mode
:character   Median : 0.0000
 Mean   : 0.1371
Mean   : 0.2211
 3rd Qu.: 0.0000
3rd Qu.: 0.0000
 Max.   :72.0000
Max.   :21.0000

 deposit_type          agent             company
days_in_waiting_list customer_type
 Length:119390      Length:119390      Length:119390      Min.   :
```

```
0.000      Length:119390
 Class :character   Class :character   Class :character   1st Qu.:
0.000      Class :character
 Mode  :character   Mode  :character   Mode  :character   Median :
0.000      Mode  :character
                                                          Mean   :
2.321
                                                          3rd Qu.:
0.000
                                                          Max.
:391.000
                                                          NA's
:7
      adr          required_car_parking_spaces
total_of_special_requests reservation_status
 Min.   :  -6.38   Min.   :0.00000             Min.   :0.0000
Length:119390
 1st Qu.:  69.29   1st Qu.:0.00000             1st Qu.:0.0000
Class :character
 Median :  94.58   Median :0.00000             Median :0.0000
Mode  :character
 Mean   : 101.83   Mean   :0.06252             Mean   :0.5714
 3rd Qu.: 126.00   3rd Qu.:0.00000             3rd Qu.:1.0000
 Max.   :5400.00   Max.   :8.00000             Max.   :5.0000

 reservation_status_date
 Length:119390
 Class :character
 Mode  :character
```

Como podemos observar, hemos comprobado que existen variables con valores nulos, atípicos, tales como stays_in_weekend_nights, agent, company, o también variables que deberían ser de otro tipo, como ï..hotel, que se presenta como string, pero debería ser de tipo Factor, con lo que presentaría dos niveles. Por lo tanto, nuestro objetivo será limpiar y preparar estos datos para poder realizar nuestro caso de análisis respectivo.

Pre-Procesar Datos

Primero se eliminaron los datos duplicados, para esto creamos un nuevo data frame, a fin de no modificar la data inicial.

```
hotel_data_pre <- unique(hotel_data)
```

Continuamos con la conversión de datos inutilizables a NA, para la correcta identificación en los comandos.

```
hotel_data_pre[hotel_data_pre == "NULL"] <- NA
hotel_data_pre[hotel_data_pre == ""] <- NA
```

Luego existe una redundancia en los datos. Según la documentación existen los tipos "*Undefined*" y "*SC*", los cuales son iguales. Así que modificamos el data frame para igualar ambos a "*SC*".

```
hotel_data_pre[hotel_data_pre == "Undefined"] <- "SC"
```

También se tiene que tener en cuenta la conversión del tipo de dato de la columna reservation_status_date.

```
hotel_data_pre$reservation_status_date <-
as.Date(hotel_data_pre$reservation_status_date,"%m/%d/%Y")
```

Además, la columna Company presenta más del 90% de datos como NA. Por esta razón decidimos eliminar la tabla.

```
hotel_data_pre2 <- subset(hotel_data_pre,select = -company)
```

Hecho esto, pasamos a la siguiente fase nuestro pre procesamiento, donde transformaremos las variables NA para poder utilizarlas en el análisis. Comenzamos por la columna lead_time, la cual tiene los datos más variados de todo el data frame. Además, tienen un rango muy amplio, por lo que decidimos que la transformación correcta sería asignarles el promedio de los datos normales.

```
hotel_data_pre2$lead_time <-
ifelse(is.na(hotel_data_pre2$lead_time),
```

```
mean(hotel_data_pre2$lead_time, na.rm = TRUE),
hotel_data_pre2$lead_time)
```

Los demás datos numéricos se encuentran en valores pequeños y de bajo rango de opciones, así que decidimos asignarles valores aleatorios. Asimismo, se asignaron valores aleatorios para los valores lógicos y de texto. Implementamos esto con las funciones `rand.valor(x)` y `random.df(df)`. La segunda recibe un data frame y agarra todas sus columnas, pasándolas por la primera función, la cual devuelve un valor aleatorio de los valores normales, y retorna un data frame con los valores ya asignados.

```
rand.valor <- function(x) {
  faltantes <- is.na(x)
  tot.faltantes <- sum(faltantes)
  x.obs <- x[!faltantes]
  valorado <- x
  valorado[faltantes] <- sample(x.obs, tot.faltantes, replace =
TRUE)
  return (valorado)
}
random.df <- function(df) {
  nombres <- names(df)
  for (nombre in nombres) {
    df[nombre] <- rand.valor(df[,nombre])
  }
  df
}
hotel_data_pre2 <- random.df(hotel_data_pre2)
```

Finalmente, para el preproceso de los datos se debe remover todos los outliers o “datos atípicos” de la base de datos. Por lo que se empleó la siguiente función para removerlos.

```
rm.outliers <- function(t, x) {
    i = grep(x,colnames(t))
    while(length(boxplot.stats(t[,i])$out) > 1) {
        t <- t[!t[,i] %in% boxplot.stats(t[,i])$out,]
    }
    return (t)
}
```

Ahora, copiando la base de datos, removemos los valores atípicos de las columnas que son relevantes para nuestro caso de análisis.

```r
#Removemos todos sus valores atípicos
boxplot(hotel_data_pre2$stays_in_weekend_nights)
hotel_data_pre3 <- rm.outliers(hotel_data_pre3,
"stays_in_weekend_nights")
boxplot(hotel_data_pre3$stays_in_weekend_nights)

#Estamos removiendo los outliers necesarios según nuestro criterio
boxplot(hotel_data_pre2$adr)
hotel_data_pre3 <- hotel_data_pre3[hotel_data_pre3$adr < 1000,]
boxplot(hotel_data_pre3$adr)

boxplot(hotel_data_pre2$lead_time)
hotel_data_pre3 <- hotel_data_pre3[hotel_data_pre3$lead_time <
700,]
boxplot(hotel_data_pre3$lead_time)

boxplot(hotel_data_pre2$adults)
hotel_data_pre3 <- hotel_data_pre3[hotel_data_pre3$adults < 10 &
hotel_data_pre3$adults > 0,]
boxplot(hotel_data_pre3$adults)

boxplot(hotel_data_pre2$children)
hotel_data_pre3 <- hotel_data_pre3[hotel_data_pre3$children < 9,]
boxplot(hotel_data_pre3$children)

boxplot(hotel_data_pre3$babies)
hotel_data_pre3 <- hotel_data_pre3[hotel_data_pre3$babies < 8,]
boxplot(hotel_data_pre3$babies)

nrow(hotel_data_pre2) - nrow(hotel_data_pre3)
#Finalmente se removieron el 0.71% de los datos
```

Visualizar Datos

Para la primera pregunta se necesita saber que tipo de hotel tiene más reservas. Para poder visualizar esta data necesitamos solamente solicitar las frecuencias que tienen cada tipo de datos. Los tipos de hoteles en el data frame son: “City Hotel” y “Resort Hotel”. Luego se hace un cálculo para saber cuan mayor es una variable respecto a la otra.

Para la segunda pregunta necesitamos analizar si hubo un incremento según el tiempo. Para hacer esto necesitamos el número de reservas por año. El data frame tiene datos correspondientes de los años 2015 al 2017. Después de obtener estos datos tenemos que realizar cálculos para hallar el incremento o reducción que hubo año por año.

Para la tercera pregunta, donde debemos conocer cuándo es la menor demanda de reservas, se necesita consultar el mes en el cual se realizó tal reserva. De esta manera se puede hallar fácilmente y comparar sus valores en una tabla.

Para la cuarta pregunta necesitamos saber cuántas reservas incluyen niños y/o bebés, por lo que al analizar la base de datos y quedarnos con las reservas con más de un bebé o más de un niño, se puede encontrar rápidamente la cantidad de reservas que los incluyen.

Para la quinta pregunta se necesita contabilizar la cantidad de espacios de estacionamiento se solicitan por reserva. De esta forma se facilita la visibilidad a la proporción de reservas que necesitan de este servicio para determinar si es un factor importante.

Para la sexta pregunta se necesita la información de reservas canceladas separadas por mes, con este fin, trabajamos en la base de datos filtrando las reservas que fueron canceladas. Con este proceso, se puede identificar fácilmente el mes en el cual se cancelan más reservas.

4. Conclusiones Preliminares

a. La gente tiende a hacer más reservas a hoteles de tipo “Hotel City”, siendo 56.52% más frecuentes que las de tipo “Resort Hotel”.
b. Las reservas tuvieron un aumento notable respecto al año 2015, pero se encuentran en un estado de decrecimiento actualmente. Del 2015 al 2016 hubo un aumento del 218.04% aproximado, mientras que del 2016 al 2017 una reducción aproximada del 25.18% de las reservas anuales.
c. La demanda de reservas a lo largo de los años que abarca la base de datos, su menor cantidad se encuentra a inicios y finales de año, incluyendo Enero, Febrero, Noviembre y Diciembre. Representa un 21.83% del total de reservas.
d. La cantidad de reservas que han incluido al menos un niño y/o bebé fueron de un 10.22%.
e. Los estacionamientos es una variable importante a tener en cuenta, aunque esto represente un 8.4% de las reservas.
f. En el mes de Agosto se cancelan la mayor cantidad de reservaciones, ascendiendo hasta 3607.