



**UNIVERSIDAD PERUANA DE CIENCIAS APLICADAS**

Administración de la Información

Sección: CC51

**INTEGRANTES:**

Chávez Arias, Bill Brandon - U20171C042

Peralta Ireijo, Sebastian Fernando - U201816030

Costa Morales, Juan Josemaria - U201822001

**PROFESORA:**

Reyes Silva, Patricia Daniela

Mayo, 2021-1

# Índice

1. Caso análisis
2. Conjunto de datos
  - 2.1. Descripción de la estructura de datos
3. Análisis exploratorio de datos
  - 3.1. Cargar Datos
  - 3.2. Inspeccionar Datos
  - 3.3. Pre-Procesar Datos
    - 3.3.1 Detección y soluciones para valores NA
    - 3.3.2 Detección y soluciones para valores outlier
  - 3.4. Visualización Gráfica - TA
    - 3.4.1. Estado de la reserva por mes
    - 3.4.2. Reservaciones de familia por mes
    - 3.4.3. Comidas ordenadas por tipo
    - 3.4.4. Cuartos reservados por tipo
  - 3.5 Visualización Gráfica - EA
    - 3.5.1 ¿Cuántas reservas se realizan por tipo de hotel? o ¿Qué tipo de hotel prefiere la gente?
    - 3.5.2 ¿Cuándo se producen las temporadas de reservas: alta, media y baja?
    - 3.5.3 ¿Cuántas reservas incluyen niños y/o bebés?
    - 3.5.4 ¿Es importante contar con espacios de estacionamiento?
    - 3.5.5 ¿En qué meses del año se producen más cancelaciones de reservas?
4. Conclusiones preliminares
5. Repositorio de Github

## 6. Referencias

## 1. Caso Análisis

Los datos con los cuales se trabajará consisten en la información sobre dos hoteles, uno de ciudad y otro estilo resort. Contiene información relacionada a la demanda de dichos hoteles. Por ejemplo: cuando se hizo una reservación, la duración de la estadía; la cantidad de adultos, niños o bebés, el tipo de cuarto, entre otras cosas. Según Kaggle, la data en cuestión fue recopilada/escrita por Nuno Antonio, Ana Almeida y Luis Nunes para “*Data in Brief, Volume 22*”, Febrero 2019. Los datos fueron posteriormente descargados y limpiados por Thomas Mock y Antoine Bichat (11 de Febrero, 2020).

El estudio y análisis de estos datos es importante, ya que permite entender mejor el comportamiento, gustos y tendencias de los clientes de hoteles. Dicha información debe ser de interés para otras cadenas hoteleras, para adaptarse a las tendencias de sus potenciales clientes. Otro interesado serían las agencias de viajes, debido a que los datos podrían ayudar a sugerir u ofrecer experiencias hoteleras de acorde al gusto mayoritario o la fecha de viaje.

## 2. Conjunto de Datos (Data Set)

#	Nombre	Tipo	Descripción	Valores
1	Hotel	Factor	Permite visualizar a cual de los dos hoteles incluidos en el dataset pertenece la data.	Resort Hotel - City Hotel
2	is_canceled	Numeric	Indica si la reserva fue cancelada (1) o no (0).	0 - 1
3	lead_time	Numeric	Representa el tiempo, en días, que ha transcurrido entre la realización de la reserva y la llegada del cliente.	0 - 737
4	arrival_date_year	Numeric	Muestra el año de llegada del cliente.	2015 - 2017
5	arrival_date_week_number	Factor	Muestra el mes de llegada del cliente.	January, February, March, April, May, June, July, August, September, October, November, December.
6	arrival_date_week_number	Numeric	Representa el número de semana del año en el que llega el cliente.	1-53
7	arrival_date_day_of_month	Numeric	Representa el número del día del mes en el que.	1-31

8	stays_in_weekend_nights	Numeric	Muestra la cantidad de fines de semana en los que el cliente se quedó o reservó en el hotel.	0-19
9	stays_in_week_nights	Numeric	Muestra la cantidad de días de semana (Lunes a Viernes) en los que el cliente se quedó o reservó en el hotel.	0-50
10	adults	Numeric	Muestra la cantidad de adultos que se registraron	0-55
11	children	Numeric	Muestra la cantidad de niños que se registraron	0-10
12	babies	Numeric	Permite visualizar el número de bebés que el cliente ha registrado.	0-10.
13	meal	Factor	Permite visualizar el tipo de comida que se ha reservado, se utiliza el estándar para paquetes de alimentos.	BB = “Bed and Breakfast”, HB = “Half Board(Breakfast and dinner normally)”, SC = “Self Catering(No meals included)”, FB = “Full Board(Breakfast, Lunch and Dinner)” y Undefined
14	country	Factor	Permite visualizar el país de origen el cual está representado en el formato ISO 3155-3:2013.	En su mayoría PRT = “Portugal” y GBR = “Gran Bretaña” entre otros
15	market_segment	Factor	Permite visualizar el segmento de mercado designado para cada cliente separados por categorías como TA(Travel Agent) o TO(Tour Operators)	Online TA, Offline TA/TO, groups, direct y corporate
16	distribution_channel	Factor	Permite visualizar el número de bebés que el cliente ha registrado	Valores numéricos mayores o iguales a cero.
17	is_repeated_guest	Numeric	Permite visualizar si es un cliente recurrente (1) o no (0).	0-1

18	previous_cancellations	Numeric	Permite visualizar el número de reservas canceladas que ha realizado el cliente.	0-26
19	previous_bookings_not_canceled	Numeric	Permite visualizar el número de reservas no canceladas del cliente.	0-72
20	reserved_room_type	Factor	Permite visualizar el código del tipo de cuarto reservado, utilizando códigos para mantener el anonimato.	A-P
21	assigned_room_type	Factor	Permite visualizar el código del tipo de cuarto asignado, este puede ser diferente del cuarto reservado debido a diferentes motivos	A-P
22	booking_changes	Numeric	Permite visualizar el número de cambios realizados en la reserva del momento que se realizó hasta el check-in o cancelación.	0-21
23	Deposit_type	Factor	Permite visualizar si el cliente realizó un depósito para garantizar la reserva.	No deposit - Non Refund - Refundable
24	agent	Factor	Permite visualizar si el cliente realizó un depósito para garantizar la reserva.	Numéricos (Existen valores nulos: NULL)
25	company	Factor	Permite visualizar el DNI de la empresa / entidad que realizó la reserva o responsable del pago de ella.	Numéricos (Existen valores nulos: NULL)
26	days_in_waiting_list	Numeric	Permite visualizar el número de días que la reserva estuvo en lista de espera antes de ser confirmada.	0-391
27	customer_type	Factor	Permite visualizar si el cliente	Transient,

			realizó un depósito para garantizar la reserva.	Transient-Party, Contract.
28	adr	Numeric	Permite visualizar la tarifa diaria promedio según se define dividiendo la suma de todas las transacciones de alojamiento por el número total de noches de estadía.	-6.38-5.4k
29	required_car_parking_spaces	Numeric	Permite visualizar el número de plazas de aparcamiento requeridas por el cliente.	0-8
30	total_of_special_requests	Numeric	Permite visualizar el número de solicitudes especiales realizadas por el cliente.	0-5
31	reservation_status	Factor	Permite visualizar el último estado de la reserva, asumiendo una de las tres categorías.	Check-Out, Canceled, No-Show
32	reservation_status_date	Factor	Permite visualizar la fecha en la que se estableció el último estado	Fechas

### 3. Análisis exploratorio de datos

#### 3.1. Cargar Datos


Para cargar los datos creamos una variable (data) a la cual le cargamos la información del .csv con el comando `data<-read.csv("../Escritorio/hotel_bookings_miss.csv",header = TRUE, stringsAsFactors = FALSE)`, la dirección depende de donde se guardó el .csv.

#### 3.2. Inspeccionar Datos

Una vez cargados los datos podemos inspeccionarlos utilizando el método `summary` para visualizar un resumen de cada columna, esto nos da una idea de que tipo de datos tiene y donde deberíamos realizar un procesamiento de los datos. Adicionalmente, tenemos el método `str` para obtener la estructura más general del dataset. Nos permite visualizar la cantidad de columnas y filas, los nombres de las columnas, el tipo de dato de cada atributo y un poco de detalle sobre el contenido de las columnas. Por último, la información relacionada a

la cantidad de observaciones y variables que tiene el dataset se puede observar a la derecha, en el ambiente de trabajo.

Al cargar los datos se puede observar que tiene 119390 observaciones y 32 variables:

Data	
 data	119390 obs. of 32 variables

Ejemplo `summary(data$children)` columna tipo Int:

Evidencia que la columna tiene valores NA.

```
> summary(data$children)
```

```
      0      1     10      2      3 NA
110796 4861      1 3652    76      4
```

Ejemplo `summary(data$arrival_month_date)` columna tipo Factor:

```
> summary(data$arrival_date_month)
```

```
April August December February January July June March May November October
September 11089 13877      6780      8068 5929 12661 10939 9794 11791 6794
11160 10508
```

Ejemplo `str(data)`:

```
> str(dataset)
'data.frame': 119390 obs. of 32 variables:
 $ 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 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 : 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       : Factor w/ 6 levels "0","1","10","2",...: 1 1 1 1 1 1 1 1 1 1 ...
 $ 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 : 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 : 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        : Factor w/ 353 levels "10","100","101",...: 353 353 353 353 353 353 353 353 353 353 ...
 $ 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 : Factor w/ 926 levels "2014-10-17","2014-11-18",...: 122 122 123 123 124 124 124 73 62 ...
```

### 3.3. Pre-Procesar Datos

#### 3.3.1 Identificación de datos faltantes (NA)

Para detectar si la columna que queremos utilizar tiene NA utilizamos el comando `summary` de la siguiente forma, especificando también la columna:



```
> summary(data$lead_time)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
0	18	69	104	160	737	21

### 3.3.2 Explicación y aplicación de la técnica utilizada para eliminar o completar los datos faltantes

Ahora podemos limpiar los datos NA de dos formas, una es eliminando el dato NA, y la segunda es crear diversas funciones para asignar un valor al dato conforme a su columna. En este primer caso eliminaremos los datos NA.

```
> data <- data[!is.na(data$lead_time),]
```

En caso se desee reemplazar el valor, podemos aplicarlas de 2 formas diferentes, una de ellas es utilizando la media de la columna para asignarle algún valor al dato NA, y la segunda opción es crear una función que retorne un random de algún valor de la columna para asignar el valor al dato NA.

En el primer caso para reemplazar los NA utilizaremos la media, primero verificamos que existen datos NA con summary, y luego los reemplazamos.

```
> data.limpio=na.omit(data)
```

```
> data.income.limpio=data[!is.na(data$arrival_date_week_number)]
```

En el segundo caso creamos una función que retorne un valor random de la columna y otra función que cree una columna nueva para tener los datos NA reemplazados.

Función que retorna valor para el NA:

```
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)  
}
```

Función que retorna una nueva columna:

```
random.df <- function(df, cols){  
  nombres <- names(df)  
  for (col in cols) {  
    nombre <- paste(nombres[col], "valorado", sep = ".")  
    df[nombre] <- rand.valor(df[,col])  
  }  
  df  
}
```

Llamada de función y visualización:

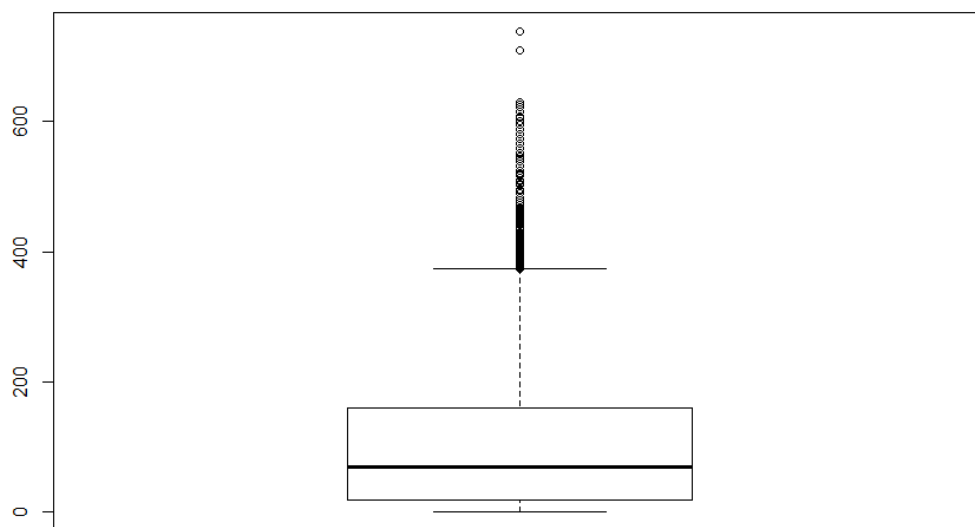
```
data.limpio=random.df(data,c(8))  
View(data.limpio)
```

### 3.3.3 Identificación de datos atípicos (Outliers).

Para la detección de valores atípicos, también conocidos como outliers, se suele recurrir a hacer un boxplot. Ya que este nos permite tener una fácil visualización de los valores que difieren mucho de lo usual, por lo que son atípicos.

Ejemplo: Boxplot para la variable *lead\_time*

```
> boxplot(data$lead_time)
```



En la gráfica se puede observar múltiples valores (bolitas) que van más allá del límite superior o bigote superior. Por lo tanto, dichos valores son atípicos.

### 3.3.4 Explicación y aplicación de la(s) técnica(s) utilizada(s) para transformar los datos atípicos.

Para corregir dichos valores atípicos, cambiaremos dichos valores por la media o la mediana dependiendo de su posición en el boxplot. Mediante una función, aquellos valores mayores al percentil 95 serán reemplazados por la mediana, mientras que, aquellos menores al percentil 5 serán cambiados por la media.

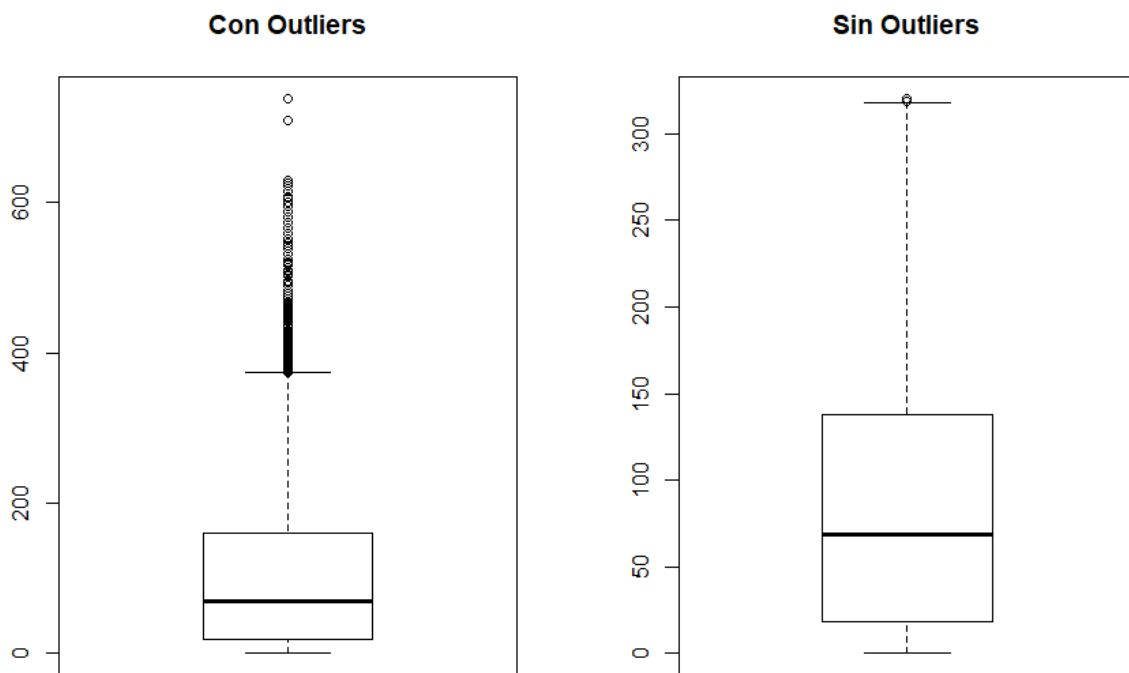
Para conseguir lo mencionado se utiliza una función aprendida en los laboratorios que es la siguiente:

```
> fix_outliers <- function(x, removeNA = TRUE){  
+   #Calculamos los cuantiles 1) por arriba del 5% y por debajo del 95%  
+   quantiles <- quantile(x, c(0.05, 0.95), na.rm = removeNA)  
+   x[x < quantiles[1]] <- mean(x, na.rm = removeNA)  
+   x[x > quantiles[2]] <- median(x, na.rm = removeNA)  
+   x  
+ }
```

Una vez generada la función *fix\_outliers()*, se pueden usar los siguientes comandos para visualizar el resultado del uso de dicha función de la siguiente manera:

```
>sinOutliers <- fix_outliers(data$lead_time)
>boxplot(sinOutliers, main="Sin Outliers")
```

Ahora comparamos el boxplot de la variable *lead\_time* con outliers, con un nuevo boxplot con la data ya procesada y libre de outliers para la misma variable:



Como se puede ver, ya no hay valores atípicos. Es decir, los valores mayores al límite superior o el bigote ya no existen y han sido reemplazados por valores más coherentes.

### 3.4. Visualización Gráfica - TA

#### 3.4.1. Estados de reserva por mes

La siguiente gráfica muestra la cantidad de reservaciones que se hicieron por mes, y en estas, cuantas fueron “Canceladas” con el color Rojo, “Verificadas” con el color Verde y “No presentadas” con el color Azul. Además que se puede observar cuales fueron los meses donde más reservas se hicieron, y en los que menos hay también. La gráfica está construida con el siguiente comando.

```
ggplot(data,aes(x=arrival_date_month))+geom_bar(aes(fill=reservation_status))
```

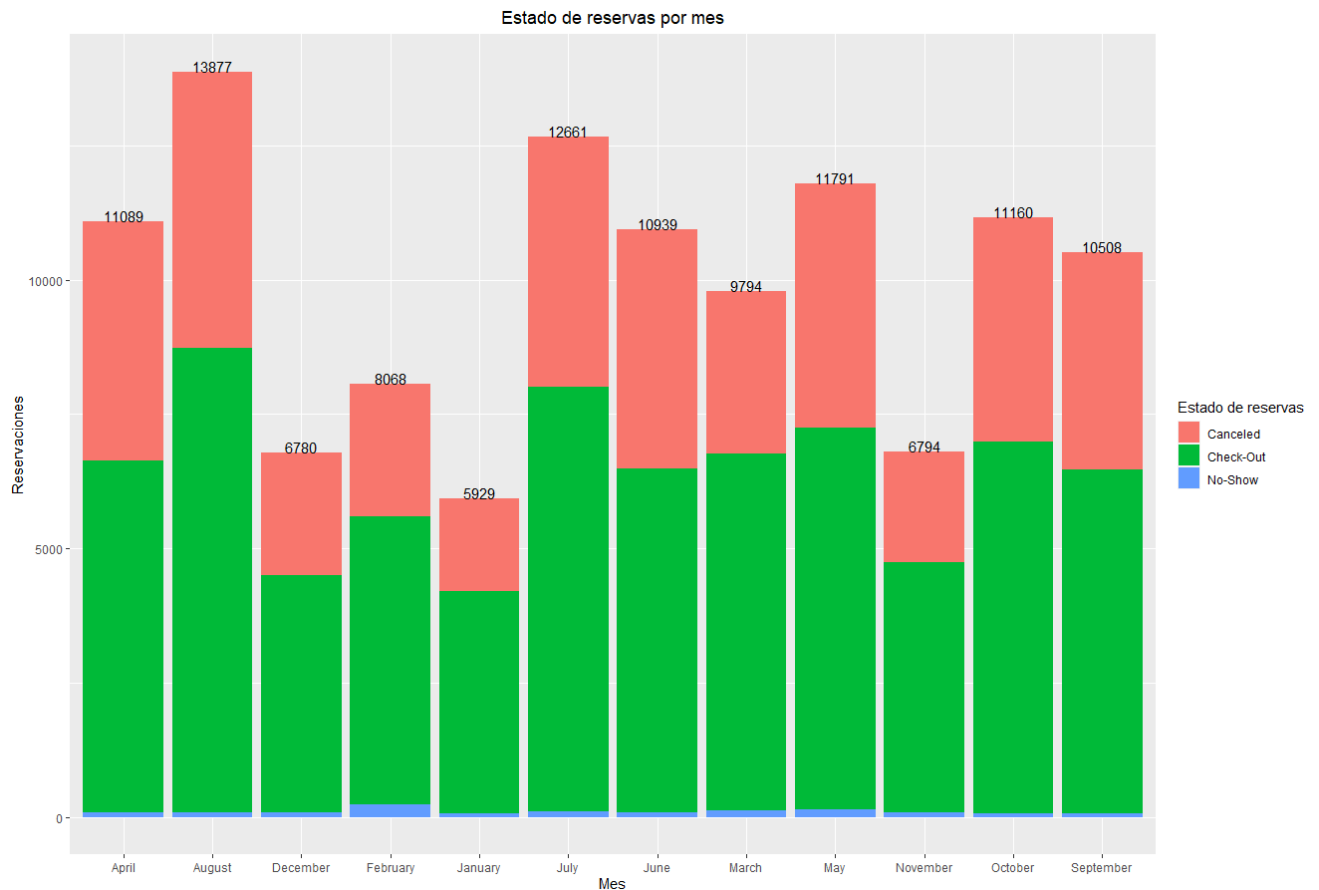


Figura 1

### 3.4.2. Reservaciones de familia por mes

La siguiente gráfica representa la proporción de reservas en familia en cada mes del año. Considerando que la presencia de un niño o bebé (Children/Babies), en los detalles de la reserva, significa que fue un viaje familiar. Se calculó la cantidad de reservas con un niño o bebe por mes y se dividió por la cantidad de reservas totales para el mismo mes. Para realizar dicho gráfico se utilizaron los siguientes comandos en un script:

```
Meses <- c('January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'October',
'September', 'November', 'December')
```

```
library(dplyr)
freq <- count(data, arrival_date_month)
freq$QTY_Families <- 0
freq$Prop <- 0
row.names(freq) <- Meses
```

```
#Limpieza de Columna Children que contiene valores NA
data$children <- ifelse(is.na(data$children), mean(data$children, na.rm = TRUE), data$children)
data$babies <- ifelse(is.na(data$babies), mean(data$babies, na.rm = TRUE), data$babies)
```

```
for (i in 1:119390) {
```

```

if (data$children[i] > 0 | data$babies[i] > 0) {
  month <- as.character(data$arrival_date_month[i])
  freq[freq$arrival_date_month == month, "QTY_Families"] <- freq[freq$arrival_date_month ==
month, "QTY_Families"] + 1
}
}
for (i in 1:12) {
  freq$Prop[i] <- freq$QTY_Families[i]/freq$n[i]
}

```

```
freq <- freq[match(Meses, freq$arrival_date_month),] #ordenar freq por mes
```

```

myGraph <- barplot(freq$QTY_Families,xlab="Meses",ylab="Cantidad",main="Reservaciones de
Familias por Mes",
  names.arg=Meses, col="orange", las=2)
text(myGraph, freq2+0.4 , paste("",freq$QTY_Families, sep="") ,cex=0.9)

```

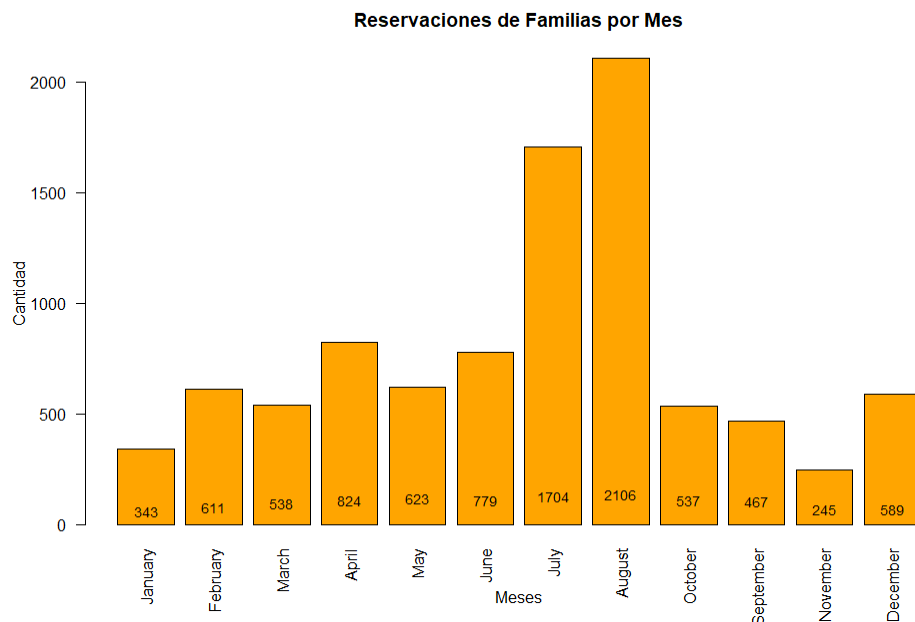


Figura 2

### 3.4.3. Comidas ordenadas por tipo

La siguiente gráfica circular permite observar los tipos de comidas que se ordenaron utilizando el estándar para paquetes de alimentos. El estándar se define en base al tipo de plan de comida que incluye la estadía. Los planes están escritos con sus siglas en inglés y son los siguientes:

- BB = “Bed and Breakfast”
- HB = “Half Board(Breakfast and dinner normally)”
- SC = “Self Catering(No meals included)”
- FB = “Full Board(Breakfast, Lunch and Dinner)”

Para realizar este gráfico se utilizaron los siguientes comandos en un script:

```
tablaComidas <- table(data$meal)
```

```
mealLabel <- paste(names(tablaComidas), "\n", tablaComidas, sep = "")
```

```
pie(tablaComidas, labels = mealLabel, main = "Comidas Ordenadas por Tipo",  
    col = rainbow(length(tablaComidas)))
```

### Comidas Ordenadas por Tipo

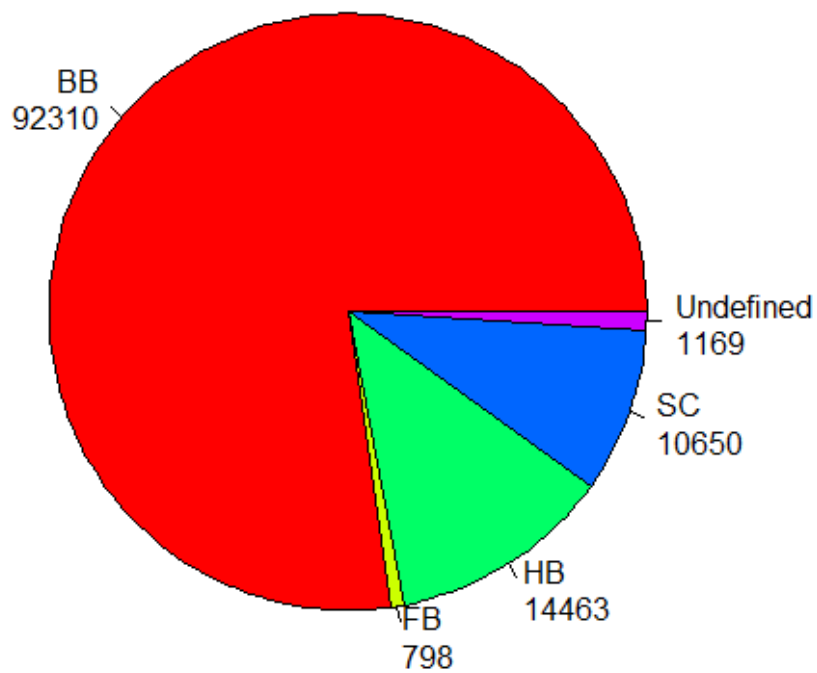


Figura 3

#### 3.4.4. Cuartos reservados por tipo

En las siguientes tablas podemos observar el número de cuartos reservados y asignados por tipo. Las categorías se definen con letras para mantener el anonimato.

```
roomTable <- table(dataset$reserved_room_type)  
barplot(roomTable, main = "Cuartos reservados por tipo")
```

```
assignedRoomTable <- table(dataset$assigned_room_type)  
barplot(assignedRoomTable, main = "Cuartos asignados por tipo", col = "green")
```

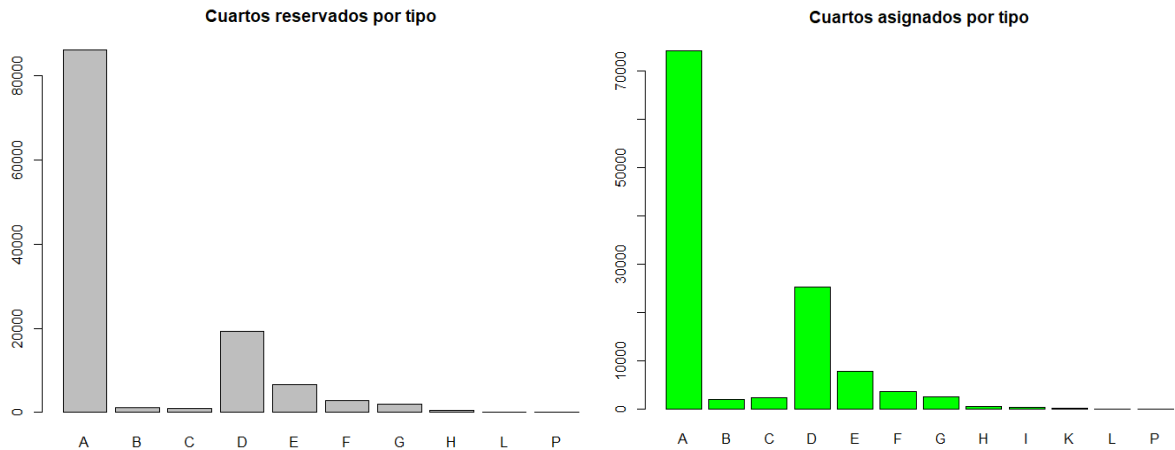


Figura 4

## 3.5 Visualización Gráfica - EA

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

Se realizó la representación gráfica de cuantas reservas se realizan por tipo de hotel, las cuales se pueden apreciar en la figura 5. De forma vertical tenemos a City Hotel y Resort Hotel, y se puede observar que cada uno tiene 3 tipos de Reservas, las cuales son “Canceled”, “Check-Out” y “No-Show”. Para representar la gráfica de barras se utilizaron los siguientes códigos.

```
install.packages("ggplot2")
library(ggplot2)

ggplot(data,aes(x=hotel))+
  geom_bar(aes(fill=reservation_status))+
  xlab("Hoteles")+
  ylab("Reservas")+
  theme(plot.title=element_text(hjust=0.5))+
  ggtitle("Numero de reservas por hotel")+
  labs(fill="Estado de reservas")+
  geom_text(stat='count',aes(label=..count..),vjust=0)
```

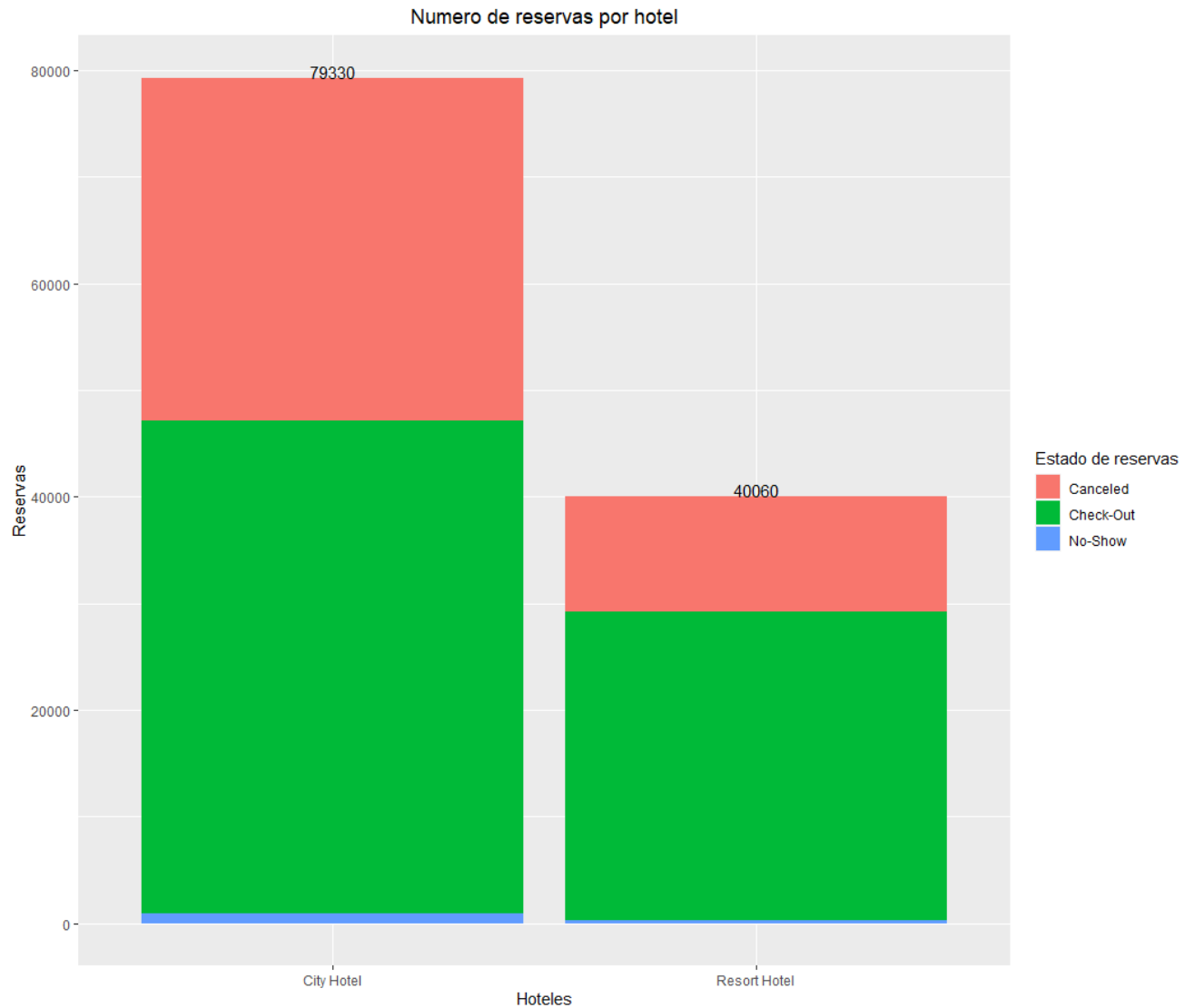


Figura 5

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

Para identificar las temporadas de reservas se utilizó un gráfico de barras. Se contó cuántas reservaciones fueron registradas para cada mes, de tal manera que, se pueda identificar la variación en la cantidad de reservas durante los meses del año. Y por lo tanto, reconocer las temporadas alta, media y baja. Se elaboró el gráfico, usando los comandos a continuación:

```
Meses <- c('January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'October',
           'September', 'November', 'December')

freq <- count(data, arrival_date_month)
row.names(freq) <- Meses
freq <- freq[match(Meses, freq$arrival_date_month),] #ordenar freq por mes

freq2 <- as.integer(freq$n/100)
```



```
myGraph <- barplot(freq2,xlab="Meses",ylab="Cantidad(10^2)",main="Cantidad de Reservas por mes del año",
  names.arg=Meses, col="orange", las=2)
```

```
text(myGraph, freq2+0.4 , paste("",freq2, sep="") ,cex=1, pos=1)
```

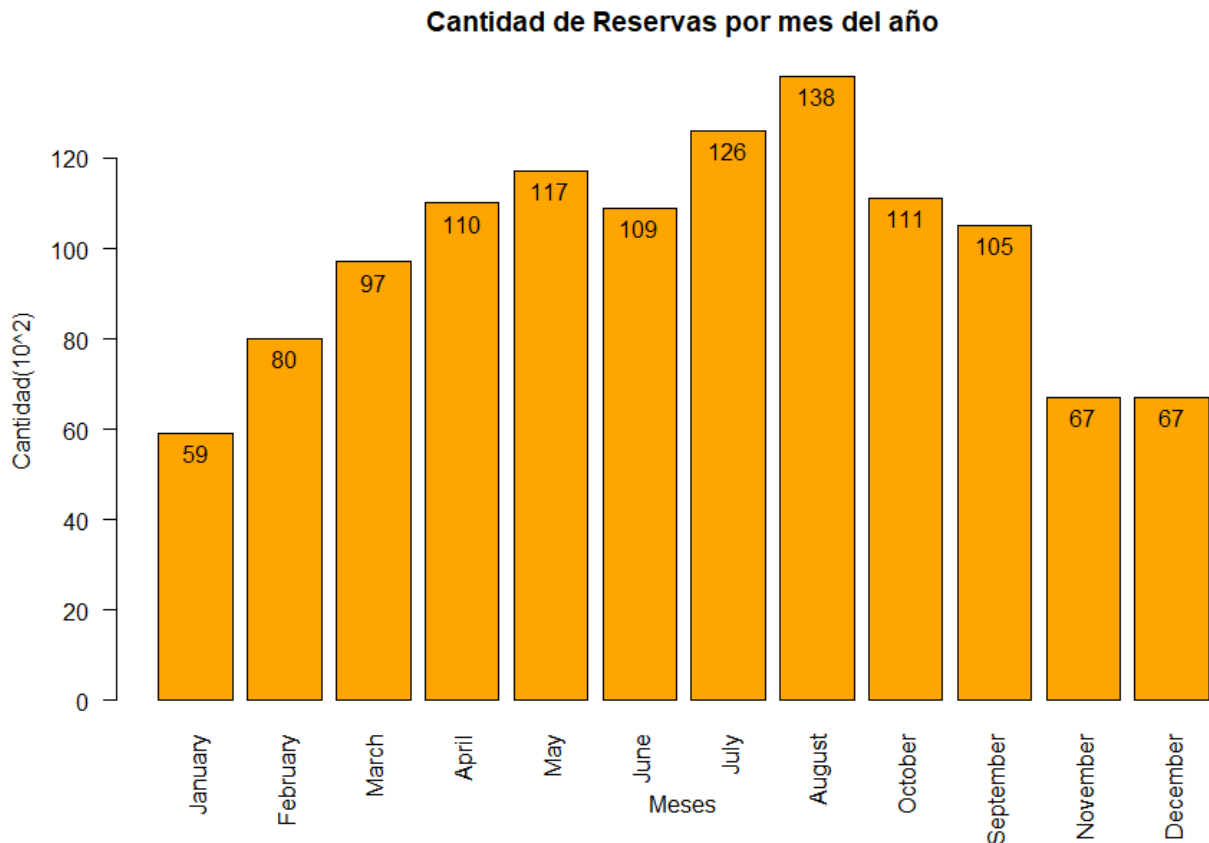


Figura 6

### 3.5.3. ¿Cuántas reservas incluyen niños y/o bebés?

Para hallar la cantidad de reservas que incluyen niños y/o bebés primero se hizo la limpieza de la variable `children` que contiene valores NA. Luego se halló la cantidad de reservas con bebés o niños y se almacenó en el valor `contador`, dicho valor es igual a 9336. Para la visualización se utilizó un pie chart que refleja, el porcentaje que 9336 reservas representa en relación al total de observaciones, es decir 119390. Por lo tanto,

$$\frac{9336}{119390} \times 100 = 0,07819... = 8\% \text{ de las reservas incluyen niños y/o bebés.}$$

```
contador <- 0
```

```
summary(data$children)
summary(data$babies)
```

```
#Limpieza de Columna Children y Babies que contiene valores NA
```

```
data$children <- ifelse(is.na(data$children), mean(data$children, na.rm = TRUE), data$children)
data$babies <- ifelse(is.na(data$babies), mean(data$babies, na.rm = TRUE), data$babies)
```

```

for (i in 1:119390) {
  if (data$children[i] > 0 | data$babies[i] > 0) {
    contador <- contador + 1
  }
}

Status <- c("Sin niño/bebé", "Incluye niño/bebé")
Cantidad <- c(119390 - contador, contador)

ResBebesNino <- data.frame(Status, Cantidad)

pct <- round(Cantidad/sum(Cantidad)*100)
Status <- paste(Status, pct)
Status <- paste(Status,"%",sep="")
pie(Cantidad, labels=Status, col=c("Yellow","Gray"), main="Porcentaje de Reservas con bebés
y/o niños")

```

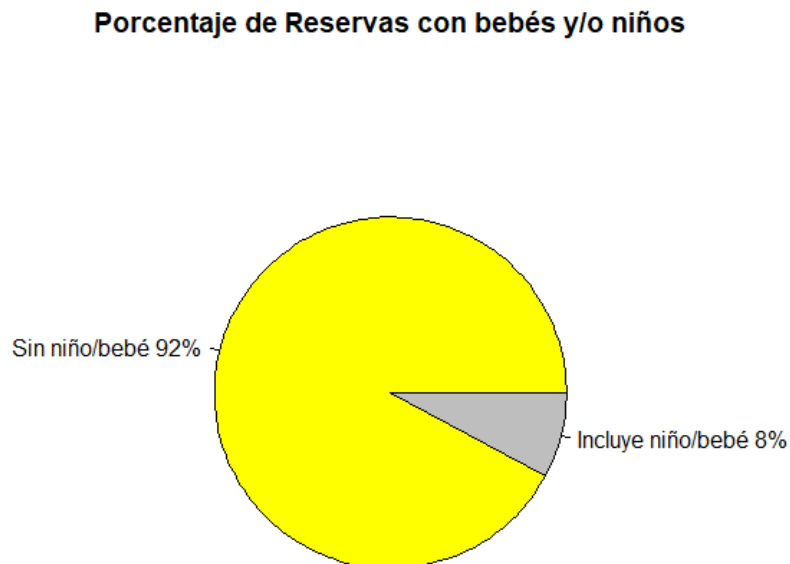


Figura 7: Se visualiza el porcentaje de reservas con bebés y niños. Se encontraron 9336 observaciones que cumplen la condición, alrededor del 8% de las observaciones totales.

### 3.5.4. ¿Es importante contar con espacios de estacionamiento?

Para determinar si es necesario tener espacios de estacionamiento utilizamos la variable `required_car_parking_Spaces`, sin embargo, para utilizarla debemos eliminar los outliers que existen debido a que la columna solo debería contar con valores 0 y 1. Luego de la limpieza tomamos los valores y calculamos los porcentajes los cuales son redondeados para facilitar su lectura. Finalmente, para visualizar los resultados creamos un pie chart y asignamos los valores necesarios.

```

tablaEstacionamiento <- table(fix_outliers(data$required_car_parking_spaces))

```

```

percent <- round(tablaEstacionamiento/sum(tablaEstacionamiento)*100)
carLabel <- c("No requiere","requiere")
carLabel <- paste(carLabel,percent)
carLabel <- paste(carLabel,"%",sep = "")
pie(tablaEstacionamiento,labels = carLabel,main = "Reservas con estacionamiento
requerido",col = terrain.colors(3))

```

### Reservas con estacionamiento requerido

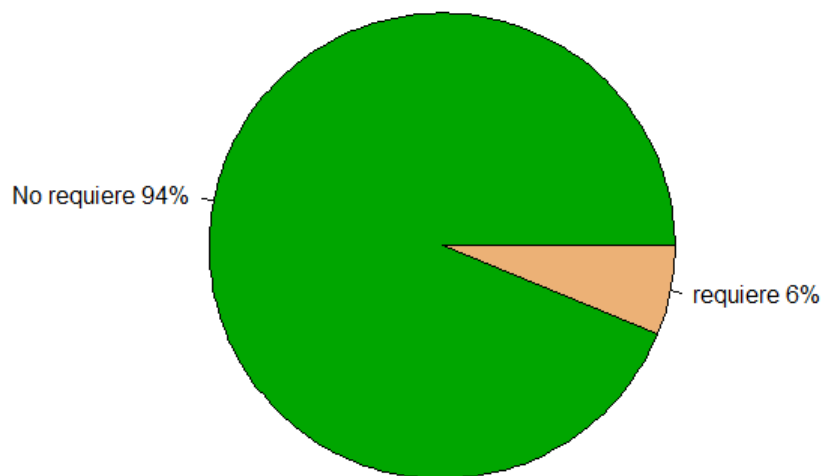


Figura 8

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

Para responder a la pregunta podemos utilizar la variable `is_canceled` en adición con `arrival_date_month`. Primero, creamos una tabla que contendrá los meses y el número de cancelaciones que será calculado luego de recorrer los valores dentro del dataset. No hace falta hacer limpieza en esta variable debido a que no hay valores NA ni outliers. Para finalizar, visualizamos los resultados en un gráfico de barras ordenado por mes.

```

library(dplyr)
meses <- c('January', 'February', 'March', 'April', 'May', 'June', 'July','August',
'October','September', 'November', 'December')

freq <- count(data, arrival_date_month)
freq <- freq[match(meses, freq$arrival_date_month),]
freq$Canceled <- 0
row.names(freq) <- meses

for (i in 1:119390) {

```

```

if (data$is_canceled[i] > 0) {
  contador <- contador + 1
  mes <- as.character(data$arrival_date_month[i])
  freq[freq$arrival_date_month == mes,"Canceled"] <- freq[freq$arrival_date_month ==
mes,"Canceled"]+1
}
}

grafico <- barplot(freq$Canceled,xlab="meses",ylab="Cantidad",main="Reservaciones
Canceladas por Mes",names.arg=meses, col="red",las =2)
text(x = grafico,y = grafico, label = freq$Canceled, pos = 3, cex = 0.8)

```

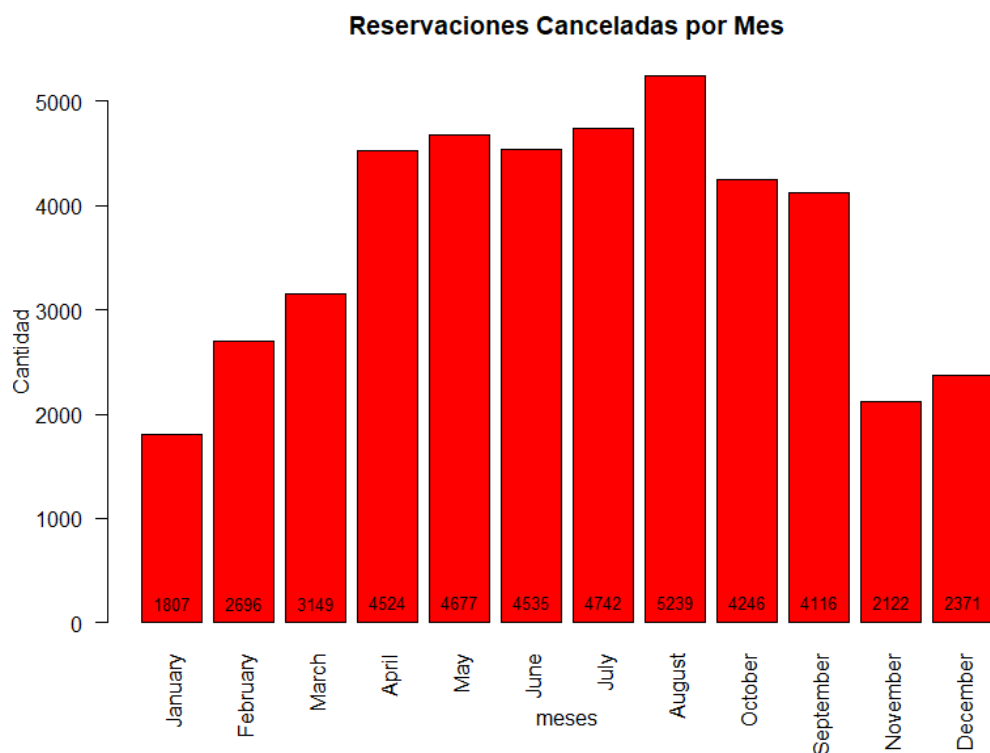


Figura 9

## 4. Conclusiones Preliminares

- Conclusiones TA:
  - En el gráfico se puede visualizar la cantidad de reservas que se realizaron en cada mes según el dataset, cada mes también está dividido en si la reservación fue “Cancelada”, “Valida” o “No presentada”, esta información es útil para las empresas al conocer la cantidad de reservas canceladas, de esta manera se estima un porcentaje de reservas que serán canceladas en el futuro y así no cerrar las

reservas a una fecha temprana, también contaremos con la información de en qué mes se realizan más reservas, de esta manera destinar los mayores atractivos y fomentar clientela fiel. (Figura 1, 3.4.1)

- Se puede observar que en Julio y Agosto es cuando hubo una mayor proporción de familias que hicieron reservas. Estos dos meses contienen 3810 reservaciones (1704 y 2106 para Julio y Agosto, respectivamente), por lo cual estos meses representan el 40% de las reservaciones con niños o bebés del dataset. Por lo tanto, se puede concluir que estos hoteles tienen un mayor atractivo para clientes con niños o bebés durante los meses mencionados. Cabe mencionar que, Julio y Agosto coinciden con las épocas de vacaciones en Estados Unidos. (Figura 2, 3.4.2)
- Entre los planes de comida reservadas se puede identificar que el tipo de comida BB(Bed & Breakfast) es el más pedido siendo HB(Breakfast & Dinner) el segundo. Con esto podemos determinar qué planes son mejores para promocionar. Adicionalmente, podemos inferir que los planes BB y HB son los más populares debido a la naturaleza de los viajes y estadías en hoteles ya que la gente que reserva hoteles suele salir en la tarde o noche para vacacionar en el lugar donde se hospedan. (Figura 3, 3.4.3)
- Luego de analizar el número de cuartos reservados y asignados se puede observar que el tipo de cuarto A es el más popular. Esto permite analizar el tipo y cantidad de gente que hace reservas. Adicionalmente, podemos observar que si bien la mayoría de personas terminan siendo asignadas al cuarto de tipo A un gran número de ellas cambia su tipo de cuarto aumentando los valores de los tipos D y E. (Figura 4, 3.4.4)

- Conclusiones EA:

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

- Según los datos expresados en la gráfico se observa que City Hotel tiene 79,330 reservas, un número superior al de Resort Hotel que cuenta con 40,060 reservas, esto es una diferencia de 39,270 reservas, es una gran diferencia. No obstante es importante resaltar el estado de estas, se puede observar en el gráfico tres colores, la parte roja representa las reservas que fueron canceladas, la parte verde las que fueron verificadas y la azul en las que no se presentaron. La cantidad de reservas no tienen la misma relevancia que la cantidad de reservas que fueron verificadas y las que fueron canceladas, según el gráfico, del total de reservas de City Hotel poco menos del 40% de las reservas fueron canceladas, sin embargo Resort Hotel poco más del 25% fueron canceladas, esto hace que la diferencia de reservas verificadas sea de aproximadamente 16000. (Figura 5 3.5.1)

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

- Se puede observar que en Mayo, Julio y Agosto se registraron 11791, 12661 y 13877 reservas. Estos son los meses que la mayor cantidad de reservas, por lo cual alrededor de Mayo hasta Agosto vendría a ser la temporada alta. Por otro lado, la temporada baja incluye los meses de Enero, Noviembre y Diciembre con 5929, 6794 y 6780 reservaciones. Los meses restantes como Marzo, Abril, Octubre y Septiembre son la temporada media con cantidades de reservas que varían alrededor de 9700 reservaciones y 11100 reservaciones. Los meses de la temporada media se caracterizan por tener tendencia ascendentes o descendentes en comparación con el mes previo y siguiente. (Figura 6, 3.5.2)

¿Cuántas reservas incluyen niños y/o bebés?

- En el gráfico se observa que el 8% de reservaciones incluyen a un niño y/o bebé. Dicho porcentaje es equivalente a 9366 reservaciones. Con estos resultados se puede concluir que, alrededor de 1 de cada 10 reservaciones o el 10% de reservaciones incluyen a un menor de edad. Por lo tanto, los tipos de hoteles utilizados para el dataset no causan mucho interés a aquellos que llevan consigo niños y/o bebés. (Figura 7, 3.5.3)

¿Es importante contar con espacios de estacionamiento?

- En los datos analizados en la figura 8 se calculó que el 93.81% de las reservaciones no necesitan espacio para estacionar sus carros. Este resultado debe estar vinculado a la ubicación del hotel y el tipo de huéspedes que tienen. Por parte de la ubicación, en caso de estar en una zona metropolitana donde el acceso al transporte público facilita el movimiento por la ciudad en comparación con una zona más rural o con menor transporte público como una playa afectará el resultado. De la misma manera que si la mayoría de huéspedes son turistas los cuales pueden no tener acceso a un carro los datos pueden variar. Finalmente, podemos concluir que es importante contar con estacionamientos, sin embargo el número de estacionamientos en relación con la cantidad de huéspedes debería seguir el mismo patrón que el gráfico lo cual ya sucede en muchos hoteles que cuentan con estacionamientos pequeños en comparación al tamaño del edificio. (Figura 8, 3.5.4)

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

- Utilizando el gráfico podemos observar que la mayoría de cancelaciones suceden a mitad del año entre los meses de Abril y Agosto siendo el último el mes con mayor cantidad de cancelaciones. Ese rango dependiendo del hemisferio en que se encuentren los hoteles puede significar la época de verano o invierno y podemos llegar a dos conclusiones. En primer lugar, los valores son más altos a mitad de año debido al incremento de las reservas por vacaciones de verano y con el incremento de reservas también aumentan las cancelaciones. Finalmente, en caso se encuentre en el hemisferio sur las vacaciones de invierno al ser más cortas es más probable que las reservas sean canceladas por inconvenientes con el trabajo o un cambio en el clima. (Figura 9, 3.5.5)

## 5. Repositorio de Github

La evidencia del desarrollo del trabajo de análisis de datos fue documentado en un repositorio de github. Además, el detalle de los scripts utilizados para las visualizaciones TA se encuentra en dicho repositorio bajo la carpeta code. El enlace a dicho repositorio será el siguiente: <https://github.com/SebsPER/ea-2021-1-cc51>.

## 6. Referencias

Mostipak, J. (Feb 12 2020). *Hotel booking demand from the paper: hotel booking demand datasets*. Recuperado de: <https://www.kaggle.com/jessemostipak/hotel-booking-demand>.