INFORME DE TB 1

CC216 – Fundamentos de Data Science
Carrera de Ciencias de la Computación

Sección: SC51
Docente: Richard Fernando Fernandez Vasquez

Integrantes:

Código	Apellidos	Nombres
u20191c439	Miranda Cardenas	Sofia Gabriel
U202220177	Medina Oropeza	Enzo Daniel
u202316342	Vásquez Trujillano	Edson Fabrizio

2024

ÍNDICE

Caso de análisis

Origen de los datos

Los datos fueron recuperados del artículo “Hotel booking demand datasets” perteneciente a Science Direct, plataforma de literatura académica revisada por Elsevier. El artículo fue desarrollado por Nuno Antonio, Ana de Almeida y Luis Nunes, y publicado en el mes de febrero de 2019. Los datos pertenecen a dos hoteles situados en Portugal. El primer hotel, denominado como H1, pertenece a la región turística de Algarve, mientras que el segundo hotel, denominado H2, pertenece a la región de Lisbon.

Los datos fueron extraídos por medio de consultas TSQL directas a la base de datos SQL del sistema de gestión de propiedades de ambos hoteles. Estas bases de datos contienen diversas tablas, de entre las cuales, la más importante es la tabla de cambios de reserva. La recopilación de datos de cada reserva se realizó un día antes de su respectiva fecha de llegada. Se tuvo en cuenta la tabla de cambios de reserva, puesto que, en comparación a la tabla de reservas, esta tabla considera los cambios posteriores a la realización de la reserva.

Casos de usos aplicables

Este análisis de datos de reservas hoteleras puede ser de interés para varios actores y puede responder a diversas necesidades o problemas:

Actores interesados en el análisis:

Gerentes y propietarios de hoteles: Pueden utilizar el análisis para optimizar la gestión de las reservas, identificar patrones de demanda, mejorar la eficiencia operativa y maximizar la ocupación.

Equipos de marketing: Con esta información, pueden ajustar sus campañas promocionales según la temporada, la región y los comportamientos de los clientes, mejorando el retorno sobre la inversión (ROI).

Desarrolladores de sistemas de gestión hotelera (PMS): Podrían utilizar los datos para identificar posibles mejoras en la funcionalidad del sistema y aumentar su eficacia.

Analistas de la industria turística: Este análisis puede ayudar a comprender las tendencias de viaje, cambios de comportamiento del consumidor, y la evolución de la demanda en regiones específicas como Algarve y Lisboa.

Investigadores académicos: Investigadores en turismo y gestión de datos pueden interesarse en los patrones y comportamientos de los consumidores a través del análisis de este dataset.

Inversionistas del sector hotelero: Los patrones de ocupación y demanda pueden ser clave para tomar decisiones sobre la inversión en nuevos proyectos o mejoras en los hoteles.

Problemas o necesidades que responde este análisis:

Optimización de la ocupación hotelera: El análisis puede ayudar a prever la demanda, optimizar tarifas y ajustar la disponibilidad para mejorar los ingresos y evitar habitaciones vacías.

Gestión de cancelaciones y cambios: El enfoque en la tabla de cambios de reserva es particularmente útil para entender el comportamiento de los clientes con respecto a cancelaciones y modificaciones, lo que ayuda a los hoteles a gestionar mejor las políticas de cancelación y sobreventa.

Previsión de demanda: Los hoteles pueden ajustar sus recursos operativos (personal, inventario, etc.) según las expectativas de ocupación basadas en datos históricos.

Segmentación de clientes: Al analizar los datos de reserva, los hoteles pueden identificar qué tipos de clientes (familias, viajeros de negocios, etc.) tienden a reservar en determinados períodos, permitiendo una segmentación más eficaz.

Análisis de la estacionalidad y regionalización: El análisis puede proporcionar información valiosa sobre la variación en la demanda en diferentes regiones (Algarve y Lisboa) y en diferentes momentos del año, lo cual es clave para la planificación a largo plazo.

Mejoras en la experiencia del cliente: Comprender los patrones de reserva y cambios puede permitir a los hoteles anticiparse a las necesidades de los clientes y ajustar la oferta de servicios para mejorar su satisfacción.

Preguntas clave para el análisis exploratorio

- ¿Cuántas reservas se realizan por tipo de hotel? ¿Qué tipo de hotel prefiere la gente?
- ¿Está aumentando la demanda con el tiempo?
- ¿Cuáles son las temporadas de reservas (alta, media, baja)?
- ¿Cuándo es menor la demanda de reservas?
- ¿Cuántas reservas incluyen niños y/o bebés?
- ¿Es importante contar con espacios de estacionamiento?
- ¿En qué meses del año se producen más cancelaciones de reservas?

Conjuntos de datos

Descripción del dataset

Variable	Tipo	Descripción
hotel	Categoría	Indica el hotel del que se extrajo los datos. Puede tomar el valor de: “Resort Hotel” o “City Hotel”
is_canceled	Categoría	Indica si la reserva fue cancelada (1) o no (0).
lead_time	Entero	Número de días transcurridos entre la fecha en que se realizó la reservación y la fecha de llegada.
arrival_date_year	Entero	Año de la fecha de llegada.
arrival_date_month	Categoría	Mes de la fecha de llegada. Representada por 12 categorías: “January”, “February”, “March”, “April”, “May”, “June”, “July”, “August”, “September”, “October”, “November”, “December”.
arrival_date_week_number	Entero	Número de semana de la fecha de llegada.
arrival_date_day_of_month	Entero	Día del mes de la fecha de llegada.
stays_in_weekend_nights	Entero	Número de noches de fin de semana que el huésped se alojó o reservó en el hotel.
stays_in_week_nights	Entero	Número de noches de entre semana (Lunes a Viernes) que el huésped se alojó o reservó en el hotel.
adults	Entero	Número de adultos.
children	Entero	Número de niños.
babies	Entero	Número de bebés.
meal	Categoría	Tipo de comida de la reserva. - Undefined/SC: Sin paquete de comida. - BB: Cama y desayuno

		- HB: Desayuno y una comida (normalmente cena) - FB: Desayuno, almuerzo y cena.
country	Categoría	País de Origen. Se representa por medio del formato ISO 3155–3:2013.
market_segment	Categoría	Designación de segmento de mercado. “TA” significa “Agentes de viajes” y “TO” significa “Operadores turísticos”.
distribution_channel	Categoría	Canal de distribución de reservas. “TA” significa “Agentes de viajes” y “TO” significa “Operadores turísticos”.
is_repeated_guest	Categoría	Indica si el propietario de la reserva ha realizado reservaciones con anterioridad (1) o no (0).
previous_cancellations	Entero	Número de reservas previas canceladas por el cliente antes de la reserva actual.
previous_bookings_not_canceled	Entero	Número de reservas previas no canceladas por el cliente antes de la reserva actual.
reserved_room_type	Categoría	Código del tipo de habitación reservada.
assigned_room_type	Categoría	Código del tipo de habitación asignado a la reserva. A veces, esta variable puede ser diferente a la de reserved_room_type. Es decir, el cuarto que ocupa el cliente puede llegar a ser distinto al que reservó.
booking_changes	Entero	Número de cambios realizados a la reserva entre la creación de la reserva en el sistema hasta el momento de check-in o cancelación.
deposit_type	Categoría	Indica si el cliente realizó un depósito para asegurar la reserva. - No Deposit: no realizó depósito. - Non Refund: se realizó

		depósito del valor total de la estancia. - Refundable: se realizó depósito de un valor menor al total de la estancia.
agent	Categoría	ID de la agencia de viajes que realizó la reservación.
company	Categoría	ID de la compañía o entidad que realizó o pagó la reserva.
days_in_waiting_list	Entero	Número de días que la reserva permaneció en la lista de espera antes de ser confirmada al cliente.
customer_type	Categoría	Tipo de reserva. - Contract: Cuando la reserva posee asociado un tipo de contrato. - Group: Cuando la reserva está asociada a un grupo. - Transient: La reserva no es parte de un grupo ni de un contrato. Tampoco está asociada a otra reserva transitoria. - Transient-party: Cuando la reserva es transitoria y está asociada, por lo menos, a otra reserva transitoria.
adr	Numérico	Tarifa diaria promedio calculada de dividir la suma de todas las transacciones de alojamiento entre el número de noches de alojamiento.
required_car_parking_spaces	Entero	Número de espacios de parqueo solicitados por el cliente.
total_of_special_requests	Entero	Número de solicitudes especiales hechas por el cliente (por ejemplo, doble cama o piso alto).
reservation_status	Categoría	Último estado de la reservación. - Canceled: Reserva cancelada por el cliente. - Check-Out: El cliente se registró y ya se retiró. - No-Show: El cliente no se registró e informó la razón del por qué.

reservation_status_date	Fecha	Fecha de la última actualización del estado de la reservación.

Análisis exploratorio de datos

Cargar datos

```
#Cargar data
setwd("D:/FABRIZIO/Documents/Downloads")
data <- read.csv ("hotel_bookings.csv", header = TRUE, sep = ",",
                  stringsAsFactors = FALSE)
data_Origen <- data # respaldo
```

Ilustración 1, demostración de carga de datos.

Cargamos los datos desde la carpeta de descargas del equipo. Agregamos los parámetros [header = TRUE], para indicar que la base contiene encabezados, [sep = “,”], para indicar que la coma es el separador entre datos del archivo .csv, y [stringsAsFactors = FALSE], para evitar cargar alguna columna del tipo String como un Factor, este cambio se realizará de forma manual.

Inspeccionar datos

```
View(data)  #Ver data
names(data) #Ver todas los nombres de columnas
ncol(data) #Dimension del conjunto (cant. de variables)
str(data)   #Estructura del conjunto y tipos de datos
head(data)  #Ver 6 primeras filas/registros
```

Ilustración 2, inspección de datos.

Por medio de las líneas escritas, podemos observar la data, ver los nombres de las variables que contiene, hallar la dimensión (tamaño) de la base, mostrar la estructura del conjunto y observar el tipo de dato de cada variable.

Pre-procesar datos

Identificar datos faltantes

```r
#----------------------------------------------------------------------
# Identificar datos faltantes (NA)
# Columnas con NA
unlist(lapply(data, function(x) any(is.na(x))))
# Cuantos NA por columna
sapply(data, function(x) sum(is.na(x)))
# Funcion para hallar solo columnas con NA
valor_NA <- function(x){
  sum = 0
  for(i in 1:ncol(x))
  {
    sumatoria = colSums(is.na(x[i]))
    if(sumatoria != 0){
      cat("En la columna",colnames(x[i]),"total de valores NA:",sumatoria,"\n")
    }
  }
}
valor_NA(data)

```

Ilustración 3, identificación de datos faltantes.

Se muestran las columnas que poseen NA:

```
> unlist(lapply(data, function(x) any(is.na(x))))
                         hotel                    is_canceled
                         FALSE                          FALSE
                     lead_time              arrival_date_year
                         FALSE                          FALSE
            arrival_date_month       arrival_date_week_number
                         FALSE                          FALSE
     arrival_date_day_of_month        stays_in_weekend_nights
                         FALSE                          FALSE
          stays_in_week_nights                         adults
                         FALSE                          FALSE
                      children                         babies
                          TRUE                          FALSE
                          meal                        country
                         FALSE                          FALSE
                market_segment           distribution_channel
                         FALSE                          FALSE
             is_repeated_guest         previous_cancellations
                         FALSE                          FALSE
previous_bookings_not_canceled             reserved_room_type
                         FALSE                          FALSE
             assigned_room_type                booking_changes
                         FALSE                          FALSE
                   deposit_type                          agent
                         FALSE                          FALSE
                       company           days_in_waiting_list
                         FALSE                          FALSE
                 customer_type                            adr
                         FALSE                          FALSE
   required_car_parking_spaces      total_of_special_requests
                         FALSE                          FALSE
            reservation_status        reservation_status_date
                         FALSE                          FALSE
>
```

Ilustración 4, demostración de datos faltantes.

Se muestran los NA en cada columna:

```
> sapply(data, function(x) sum(is.na(x)))
                          hotel                      is_canceled
                              0                                0
                      lead_time                arrival_date_year
                              0                                0
             arrival_date_month         arrival_date_week_number
                              0                                0
      arrival_date_day_of_month          stays_in_weekend_nights
                              0                                0
           stays_in_week_nights                           adults
                              0                                0
                       children                           babies
                              4                                0
                           meal                          country
                              0                                0
                 market_segment            distribution_channel
                              0                                0
              is_repeated_guest          previous_cancellations
                              0                                0
previous_bookings_not_canceled               reserved_room_type
                              0                                0
             assigned_room_type                   booking_changes
                              0                                0
                   deposit_type                            agent
                              0                                0
                        company            days_in_waiting_list
                              0                                0
                  customer_type                              adr
                              0                                0
    required_car_parking_spaces       total_of_special_requests
                              0                                0
             reservation_status          reservation_status_date
                              0                                0
>
```

Ilustración 5, demostración de datos faltantes.

Se muestran solo las columnas que poseen NA y la cantidad:

```
> valor_NA <- function(x){
+   sum = 0
+   for(i in 1:ncol(x))
+   {
+     sumatoria = colSums(is.na(x[i]))
+     if(sumatoria != 0){
+       cat("En la columna",colnames(x[i]),"total de valores NA:",sumatoria,"\n")
+     }
+   }
+ }
> valor_NA(data)
En la columna children total de valores NA: 4
>
```

Ilustración 6, demostración de datos faltantes.

Existen 4 valores NA en la columna “children”.

Tratamiento de datos faltantes

```r
#-------------------------------------------------------------
# Tratamiento de datos faltantes
# Reemplazar datos NA por la media
cambiar_NA <- function(x){
  col_numericas <- sapply(x, is.numeric)
  for(col in names(x)[col_numericas]){
    x[is.na(x[,col]), col] <- mean(x[,col], na.rm = TRUE)
  }
  return (x)
}
data <- cambiar_NA(data)
# Probar si existen valores NA
valor_NA(data)
```

Ilustración 6,tratamiento de datos faltantes.

Se intercambian los datos NA por la media de su respectiva columna. Estamos teniendo en cuenta que la única columna con datos NA es del tipo numérica (children).

```r
> cambiar_NA <- function(x){
+   col_numericas <- sapply(x, is.numeric)
+   for(col in names(x)[col_numericas]){
+     x[is.na(x[,col]), col] <- mean(x[,col], na.rm = TRUE)
+   }
+   return (x)
+ }
> data <- cambiar_NA(data)
> # Probar si existen valores NA
> valor_NA(data)
>
```

Ilustración 7, scrip donde se intercambian los datos.

Al ejecutar valor_NA(data) se muestra que ya no existen columnas donde se registren valores NA.

Identificación de datos atípicos

```r
#-------------------------------------------------------------------
# Identificar datos atípicos (Outliers)
# Verificar atipicos en bloxplot
# Ejemplo con la variable stays_in_week_nights
boxplot(data$stays_in_week_nights, main =
          "[stays_in_week_nights]\n antes del tratamiento", boxwex = 0.5)
# Mostrar valores atípicos
# Ejemplo con la variable stays_in_week_nights
sum(boxplot(data$stays_in_week_nights)$out)
# Cantidad de valores atipicos
# Ejemplo con la variable stays_in_week_nights
sum(!sapply(boxplot(data$stays_in_week_nights)$out, is.null))
# Imprimir variables numéricas y su cant. de datos atípicos
imp_outliers <- function(x){
  col_numericas <- sapply(x, is.numeric)
  for(col in names(x)[col_numericas]){
    total <- sum(!sapply(boxplot(x[col])$out, is.null))
    if(total != 0){
      cat("Para la columna ", col, ", su total de valores atípicos es: ", total, "\n")
    }
  }
}
imp_outliers(data)
```

Ilustración 7, identificación de datos atípicos.

En primer lugar, podemos verificar la existencia de datos atípicos en una columna (variable) por medio de boxplot.

```
> #-----------------------------------------------------------------
> # Identificar datos atípicos (Outliers)
> # Verificar atipicos en bloxplot
> # Ejemplo con la variable stays_in_week_nights
> boxplot(data$stays_in_week_nights, main =
+           "[stays_in_week_nights]\n antes del tratamiento", boxwex = 0.5)
>
```

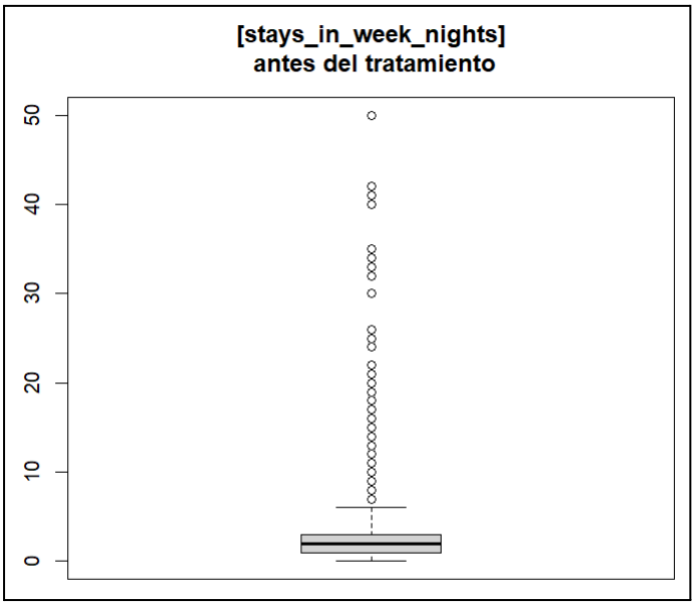

Ilustración 8 y 9, verificación de la existencia de los datos atípicos.

En este caso, se observa que la columna “stays_in_week_nights” posee varios datos atípicos, representados por puntos sin relleno.

En segundo lugar, podemos solicitar que se nos entregue directamente la cantidad de datos atípicos. Por ejemplo, para la columna “stays_in_week_nights” existen 31428 datos atípicos.

```
> # Mostrar valores atípicos
> # Ejemplo con la variable stays_in_week_nights
> sum(boxplot(data$stays_in_week_nights)$out)
[1] 31428
>
```

Ilustración 10, scrip donde se intercambian los datos.

En tercer lugar, implementamos una función para imprimir la cantidad de valores atípicos de cada columna.

```
> # Imprimir variables numéricas y su cant. de datos atípicos
> imp_outliers <- function(x){
+   col_numericas <- sapply(x, is.numeric)
+   for(col in names(x)[col_numericas]){
+     total <- sum(!sapply(boxplot(x[col])$out, is.null))
+     if(total != 0){
+       cat("Para la columna ", col, ", su total de valores atípicos es: ", total, "\n")
+     }
+   }
+ }
> imp_outliers(data)
Para la columna  lead_time , su total de valores atípicos es:  3005
Para la columna  stays_in_weekend_nights , su total de valores atípicos es:  265
Para la columna  stays_in_week_nights , su total de valores atípicos es:  3354
Para la columna  adults , su total de valores atípicos es:  29710
Para la columna  children , su total de valores atípicos es:  8594
Para la columna  babies , su total de valores atípicos es:  917
Para la columna  previous_cancellations , su total de valores atípicos es:  6484
Para la columna  previous_bookings_not_canceled , su total de valores atípicos es:  3620
Para la columna  booking_changes , su total de valores atípicos es:  18076
Para la columna  days_in_waiting_list , su total de valores atípicos es:  3698
Para la columna  adr , su total de valores atípicos es:  3793
Para la columna  required_car_parking_spaces , su total de valores atípicos es:  7416
Para la columna  total_of_special_requests , su total de valores atípicos es:  2877
>
```

Ilustración 11, impresión de datos atípicos.

Tratamiento de datos atípicos

```
#-----------------------------------------------------------------------
# Tratamiento de datos atípicos (Outliers)
# Metodo de los cuantiles
# 1) valor atipico por debajo del 5% -> media
# 2) valor atipico por encima del 95% -> mediana
tratar_outliers <- function(x, removeNA = TRUE){
  col_numericas <- sapply(x, is.numeric)
  for(col in names(x)[col_numericas]){
    cuantiles <- quantile(x[[col]], c(0.05, 0.95), na.rm = removeNA)
    x[[col]][x[[col]]<cuantiles[1]] <- mean(x[[col]], na.rm = removeNA)
    x[[col]][x[[col]]>cuantiles[2]] <- median(x[[col]], na.rm = removeNA)
  }
  return(x)
}
data_TA <- tratar_outliers(data)
```

Ilustración 12, tratamiento de datos atípicos.

Decidimos realizar la limpieza de los datos atípicos por medio de comparar los cuantiles de cada columna. Si un valor atípico está por debajo del 5% entonces será reemplazado por la media de la variable a la que pertenece. Caso contrario, si el valor atípico está por encima del 95%, será reemplazado por la mediana.

```
> tratar_outliers <- function(x, removeNA = TRUE){
+   col_numericas <- sapply(x, is.numeric)
+   for(col in names(x)[col_numericas]){
+     cuantiles <- quantile(x[[col]], c(0.05, 0.95), na.rm = removeNA)
+     x[[col]][x[[col]]<cuantiles[1]] <- mean(x[[col]], na.rm = removeNA)
+     x[[col]][x[[col]]>cuantiles[2]] <- median(x[[col]], na.rm = removeNA)
+   }
+   return(x)
+ }
> data_TA <- tratar_outliers(data)
```

Ilustración 13, ejecución del codigo.

Ejecutamos la función tratar_outliers(data) y guardamos la nueva data en data_TA. Finalmente, comparamos la variable “adults” en data y data_TA para demostrar el tratamiento de los datos atípicos.

```
> data_TA <- tratar_outliers(data)
> par(mfrow = c(1,2))
> boxplot(data$adults, main = "[Adulto] sin tratamiento", boxwex = 0.5)
> boxplot(data_TA$adults, main = "[Adulto] con tratamiento", boxwex = 0.5)
>
```

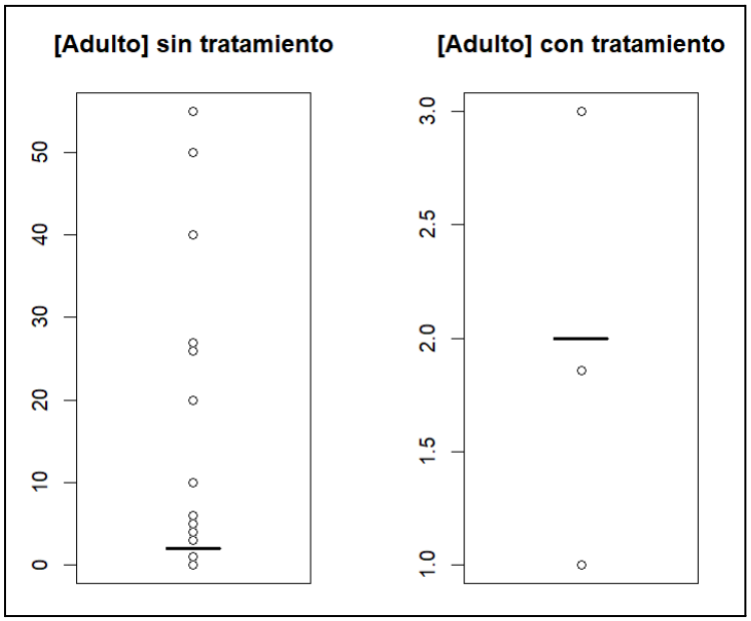

Ilustración 14 y 15, ejecución del script.

Como se observa en la imagen, el intervalo de valores de la variable “adulto” se redujo de [0; 55] a [1; 3], lo cual era el objetivo de realizar este tratamiento.

Visualización de datos:

Pregunta clave 1: ¿Cuántas reservas se realizan por tipo de hotel? ¿Qué tipo de hotel prefiere la gente?

Los resultados indican lo siguiente:

City Hotel: 46,228 reservas no canceladas.
Resort Hotel: 28,938 reservas no canceladas.
Interpretación:

Preferencia por el City Hotel:

El City Hotel registra significativamente más reservas no canceladas (46,228) que el Resort Hotel (28,938). Esto representa una diferencia de 17,290 reservas a favor del City Hotel.

El hecho de que las reservas en el City Hotel sean mayores podría deberse a varios factores: Mayor accesibilidad, dado que los City Hotels suelen estar ubicados en áreas urbanas con más demanda, como distritos comerciales o centros de negocios.
Los clientes prefieren estadías cortas o viajes de negocios, que suelen ser más comunes en áreas urbanas.

Menor demanda del Resort Hotel:

Los Resort Hotels, aunque todavía populares con casi 29,000 reservas no canceladas, tienen menos demanda comparado con los City Hotels.
Esto puede deberse a que los resorts tienden a atraer a turistas que buscan estadías más largas o de ocio, lo que podría ser menos frecuente o estacional en comparación con las estadías en hoteles urbanos.

```
  City Hotel Resort Hotel
       46228        28938
> # Crear el gráfico de barras
> barplot(tabla_frecuencia_hotel,
+         col = c("lightblue", "lightgreen"),  # Colores personalizados
+         main = "Total de Reservas No Canceladas por Tipo de Hotel",
+         xlab = "Tipo de Hotel",
+         ylab = "Total de Reservas No Canceladas",
+         ylim = c(0, max(tabla_frecuencia_hotel) * 1.1),  # Ajustar el límite del eje y
+         border = "black")  # Bordes de las barras
> # Conclusión
> if (tabla_frecuencia_hotel["City Hotel"] > tabla_frecuencia_hotel["Resort Hotel"]) {
+   cat("La preferencia de los huéspedes es por el City Hotel.\n")
+ } else {
+   cat("La preferencia de los huéspedes es por el Resort Hotel.\n")
+ }
La preferencia de los huéspedes es por el City Hotel.
> # Convertir la columna de fecha a tipo fecha
> data$reservation_status_date <- as.Date(data$reservation_status_date, format = "%Y-%m-%d")
> ####
```

Ilustración 16, ejecución del código.

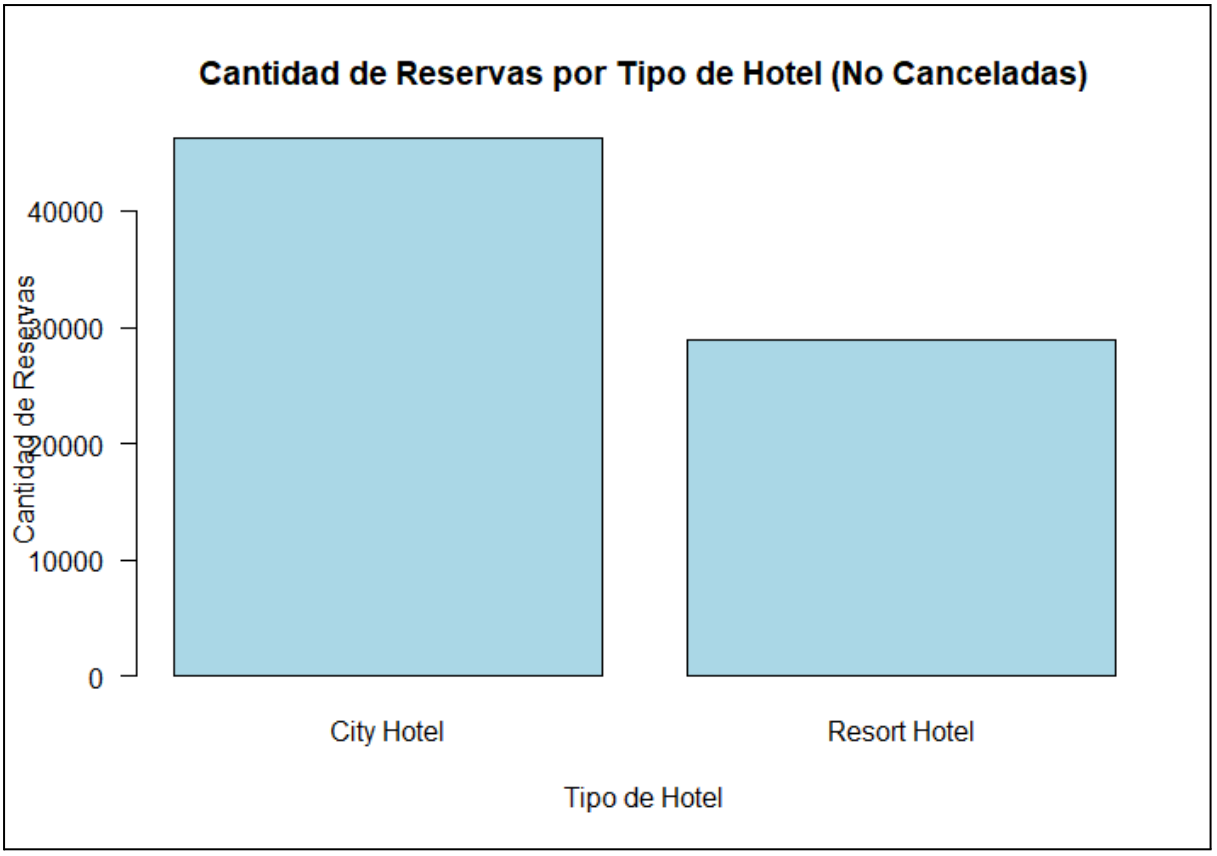

Ilustración 17,gráfico de barras de cancelaciones por tipo de hotel.

En conclusión, “City Hotel” es el hotel más preferido, pues es el hotel que posee más reservas dentro de la base de datos.

Pregunta clave 2: ¿Está aumentando la demanda con el tiempo?

Los resultados de la tabla de frecuencias de las reservas no canceladas muestran el siguiente total de reservas por año:

2015: 13,462 reservas no canceladas
2016: 36,369 reservas no canceladas
2017: 25,335 reservas no canceladas
Interpretación:

Aumento en 2016:

De 2015 a 2016, el número de reservas no canceladas casi se triplica, pasando de 13,462 en 2015 a 36,369 en 2016. Este aumento podría indicar una mayor demanda o mejoras en las operaciones del hotel que incentivaron una mayor retención de reservas.
El salto significativo sugiere que algún factor externo o interno impactó positivamente en la cantidad de reservas, como una buena estrategia de marketing, la apertura de nuevos servicios o un incremento del turismo en la región.

Disminución en 2017:

En 2017, el número de reservas no canceladas cae a 25,335, lo que representa una disminución del 30% con respecto a 2016.
Aunque sigue siendo más alto que en 2015, la caída podría señalar posibles desafíos durante ese año, como una menor afluencia de turistas, mayor competencia, o problemas operativos que podrían haber afectado las reservas o la tasa de cancelaciones.

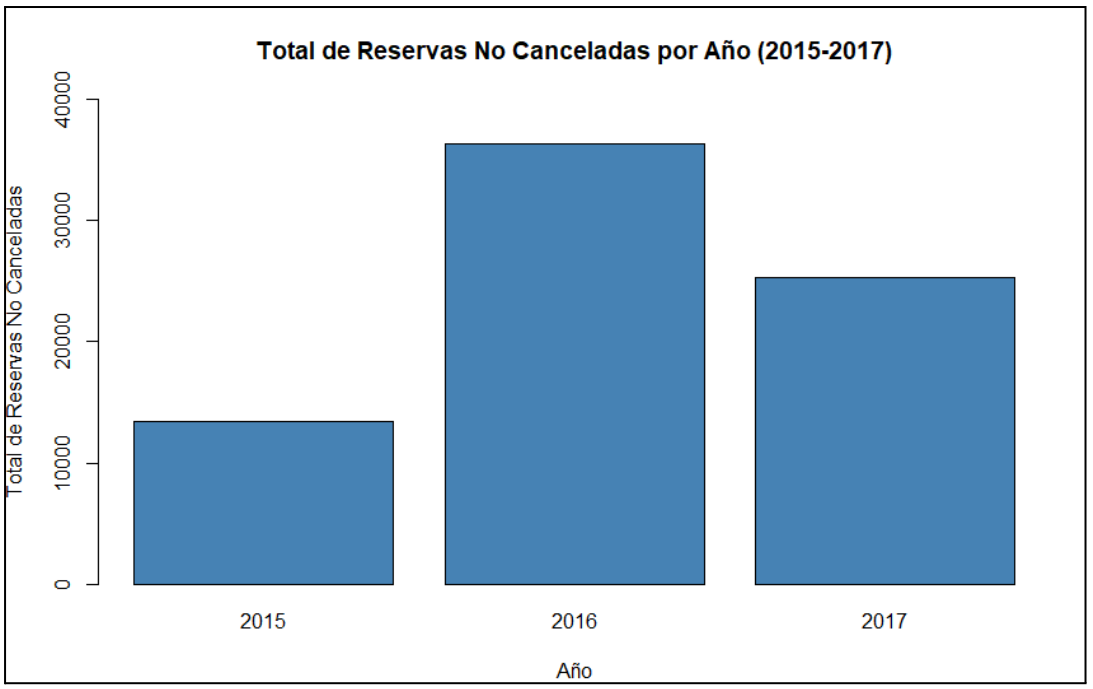

Ilustración 18, gráfico de barras.

```
> # Mostrar los resultados en la consola
> print(tabla_frecuencia_no_canceladas)

 2015  2016  2017
13462 36369 25335
>
```

Ilustración 19, ejecución del código.

Ilustración 20, gráfico de barras.

```
Análisis de Aumento de Demanda para City Hotel:
> print(df_city)
   Año Total_Reservas
1 2015           7465
2 2016          22727
3 2017          16036
```

Ilustración 21, ejecución del código.

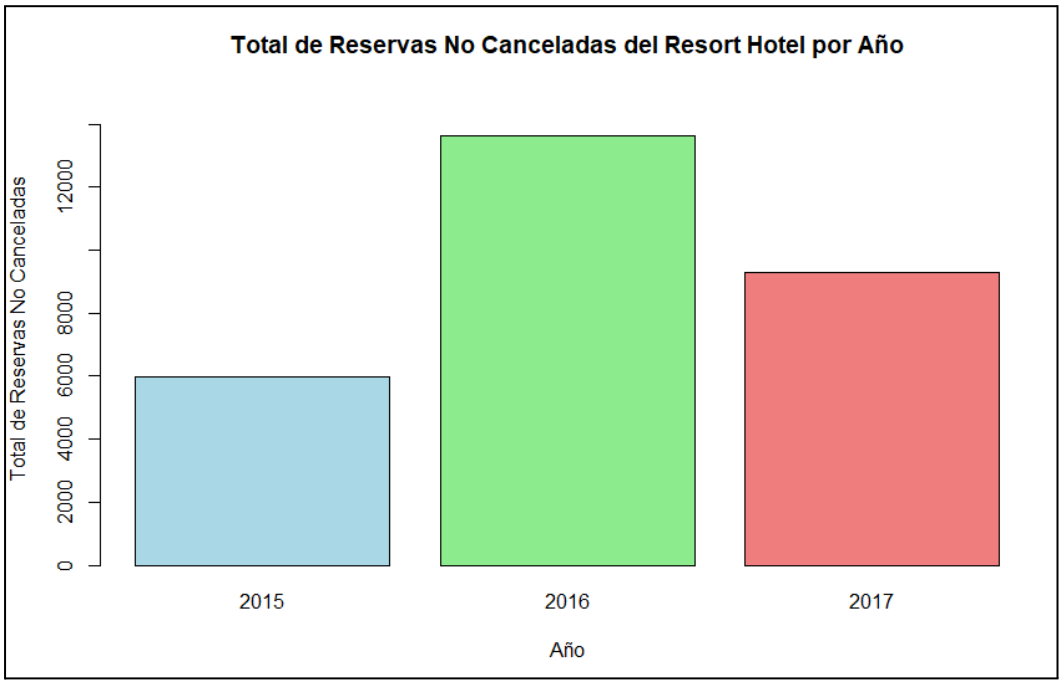

Ilustración 22, gráfico de barras.

```
Análisis de Aumento de Demanda para Resort Hotel:
> print(df_resort)
   Año Total_Reservas
1 2015           5997
2 2016          13642
3 2017           9299
> # Verificar si hubo aumento
```

Ilustración 23, ejecución del código.

En conclusión, a simple vista observamos que la demanda de reservaciones incrementa y decrece al pasar el tiempo, es decir, no permanece en un aumento constante.

Pregunta clave 3: ¿Cuáles son las temporadas de reservas (alta, media, baja)?

Con base en los datos de reservas no canceladas por mes, podemos dividir el año en tres temporadas según el volumen de reservas:

April	August	December	February	January	July	June	March	May	November	October	September
6565	8638	4409	5372	4122	7919	6404	6645	7114	4672	6914	6392

Ilustración 24, ejecución del código.

Temporada Alta (Mayor demanda de reservas):
Meses: August (8638 reservas), July (7919 reservas)
Estos meses corresponden a la temporada alta, con el mayor número de reservas. Esto puede deberse a factores como las vacaciones de verano en el hemisferio norte, cuando muchas personas viajan.

Temporada Media (Demanda moderada de reservas):
Meses: March (6645), May (7114), October (6914), June (6404), September (6392)
Estos meses tienen una demanda relativamente estable, aunque menor que en la temporada alta.

Temporada Baja (Menor demanda de reservas):
Meses: January (4122), February (5372), April (6565), November (4672), December (4409)
Enero y diciembre suelen tener la menor demanda, posiblemente debido a que la temporada festiva termina y las personas vuelven a sus rutinas habituales.

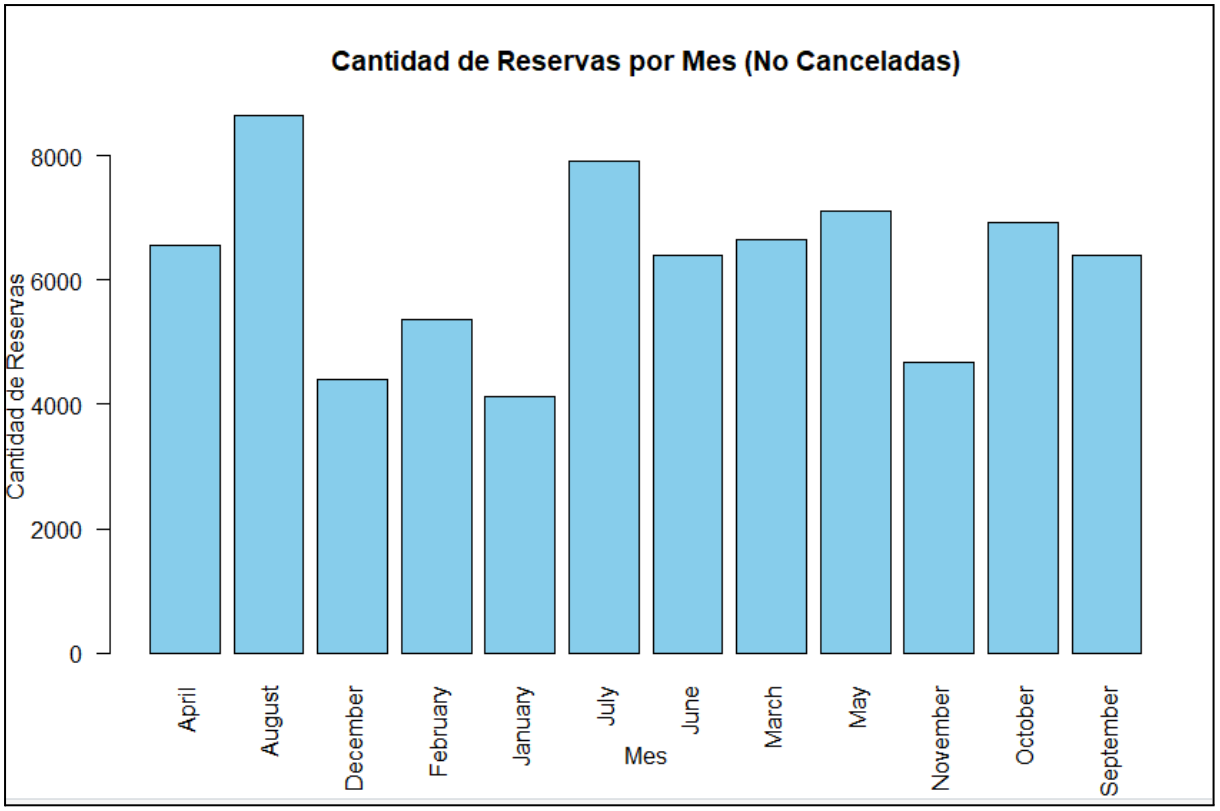

Ilustración 25, gráfico de barras.

Este análisis ayuda a identificar cuándo ajustar las tarifas, lanzar promociones o planificar recursos en función de la demanda esperada.

Pregunta clave 4: ¿Cuándo es menor la demanda de reservas?

El mes con menor demanda de reservas es enero (January). Esto indica que enero es el mes en el que se registran menos reservas en comparación con otros meses del año.

```
> cat("El mes con menor demanda de reservas es:", mes_
El mes con menor demanda de reservas es: January
> # Agrupar la tarifa promedio por mes
```

Ilustración 26, ejecución del código.

Este tipo de análisis es útil para implementar estrategias como descuentos o promociones durante meses de baja demanda, ayudando a incrementar las reservas durante estas épocas.

Pregunta clave 5: ¿Cuántas reservas incluyen niños y/o bebés?

Para responder esta pregunta, desarrollamos un código en R que analiza la cantidad de reservas que incluyen niños y/o bebés.

```r
kids_data <- data %>% mutate(
    reservation_type = case_when(
      children > 0 & babies > 0 ~ "Con niños y bebés",
      children > 0 & babies == 0 ~ "Solo niños",
      children == 0 & babies > 0 ~ "Solo bebés",
      TRUE ~ "Sin niños ni bebés"
    )
  )

# Asegurarse de que la nueva columna `reservation_type` sea de tipo factor o texto
data$reservation_type <- as.factor(data$reservation_type)

# Pregunta: ¿Cuántas reservas incluyen niños y/o bebés?
# Gráfico de barras para visualizar la cantidad de reservas según tipo (niños, bebés o ambos)
ggplot(kids_data, aes(x = reservation_type)) +
  geom_bar(fill = "lightblue") +
  labs(x = "Tipo de Reserva",
       y = "Cantidad de Reservas",
       title = "Reservas que Incluyen Niños y/o Bebés") +
  theme_minimal()

# Crear una tabla resumen para ver los números exactos de cada tipo de reserva
resumen_reservas <- kids_data %>%
  group_by(reservation_type) %>%
  summarise(count = n())

print(resumen_reservas)

# Calcular el total de reservas que incluyen niños y/o bebés
total_con_ninos_o_bebes <- resumen_reservas %>%
  filter(reservation_type != "Sin niños ni bebés") %>%
  summarise(total = sum(count))
```

Ilustración 27, propuesta de solución.

Resultado:

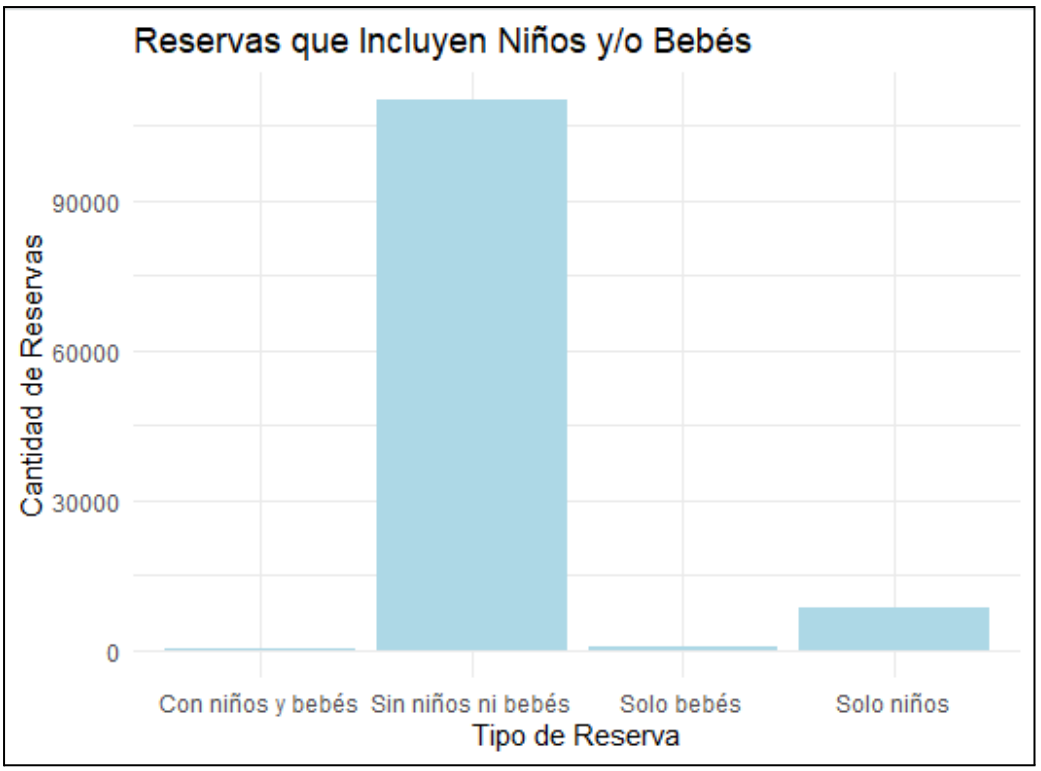

Ilustración 28, gráfico de barras.

```
> ggplot(kids_data, aes(x = reservation_type)) +
+   geom_bar(fill = "lightblue") +
+   labs(x = "Tipo de Reserva",
+        y = "Cantidad de Reservas",
+        title = "Reservas que Incluyen Niños y/o Bebés") +
+   theme_minimal()
> resumen_reservas <- kids_data %>%
+   group_by(reservation_type) %>%
+   summarise(count = n())
> print(resumen_reservas)
# A tibble: 4 x 2
  reservation_type    count
  <chr>               <int>
1 Con niños y bebés     175
2 Sin niños ni bebés 110058
3 Solo bebés            742
4 Solo niños           8415
> total_con_ninos_o_bebes <- resumen_reservas %>%
+   filter(reservation_type != "Sin niños ni bebés") %>%
+   summarise(total = sum(count))
> print(total_con_ninos_o_bebes)
# A tibble: 1 x 1
  total
  <int>
1  9332
```

Ilustración 29, ejecución del código.

Como se muestra en el código, se crea una nueva columna llamada “reservation_type” que clasificará cada reserva en cuatro categorías:

- “Con niños y bebés”
- “Solo niños”
- “Solo bebés”

- “Sin niños ni bebés”

Luego se crea un gráfico de barras que nos permitirá mostrar la cantidad de reservas en cada categoría. Después se crea un tabla resumen que cuenta la cantidad de reservas en cada categoría.
Por último se calcula el total de reservas que incluyen niños y/o bebés, excluyendo las reservas que no tienen ni niños ni bebés.

Tomando en cuenta los pasos y resultados, concluimos que la cantidad de reservas que incluyen niños y/o bebés es 9332.

Pregunta clave 6: ¿Es importante contar con espacios de estacionamiento?

Para responder esta pregunta, desarrollamos un código en R que analiza la demanda de estacionamiento, la tasa de cancelación por tipo de estacionamiento y solicitudes adicionales.

```r
# Pregunta clave 6
# ¿Es importante contar con espacios de estacionamiento?
# Crear una nueva columna indicando si hay estacionamiento o no
hotel_data <- data_TA %>%
mutate(estacionamiento = ifelse(required_car_parking_spaces > 0, "Con Estacionamiento", "Sin Estacionamiento"))

# Gráfico de barras de reservas con/sin estacionamiento
ggplot(hotel_data, aes(x = estacionamiento, fill = estacionamiento)) +
  geom_bar() +
  labs(title = "Distribución de Reservas con y sin Espacios de Estacionamiento",
       x = "Tipo de Reserva",
       y = "Cantidad de Reservas") +
  theme_minimal()

# Calcular la tasa de cancelación por tipo de estacionamiento
tasa_cancelacion <- hotel_data %>%
  group_by(estacionamiento) %>%
  summarise(tasa_cancelacion = mean(is_canceled))

# Gráfico de tasa de cancelación
ggplot(tasa_cancelacion, aes(x = estacionamiento, y = tasa_cancelacion, fill = estacionamiento)) +
  geom_col() +
  labs(title = "Tasa de Cancelación para Reservas con y sin Estacionamiento",
       x = "Tipo de Reserva",
       y = "Tasa de Cancelación") +
  scale_y_continuous(labels = scales::percent) +
  theme_minimal()

# Calcular el promedio de solicitudes especiales
solicitudes_especiales <- hotel_data %>%
  group_by(estacionamiento) %>%
  summarise(promedio_solicitudes = mean(total_of_special_requests))

# Gráfico de solicitudes especiales
```

Ilustración 30, propuesta de solución.

Resultado:

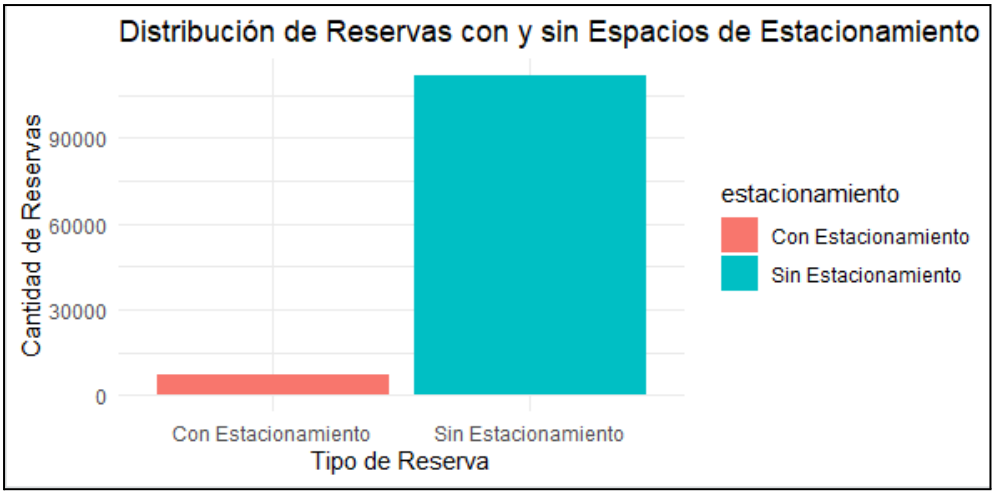

Ilustración 31, gráfico de barras.

Ilustración 32, gráfico de barras.

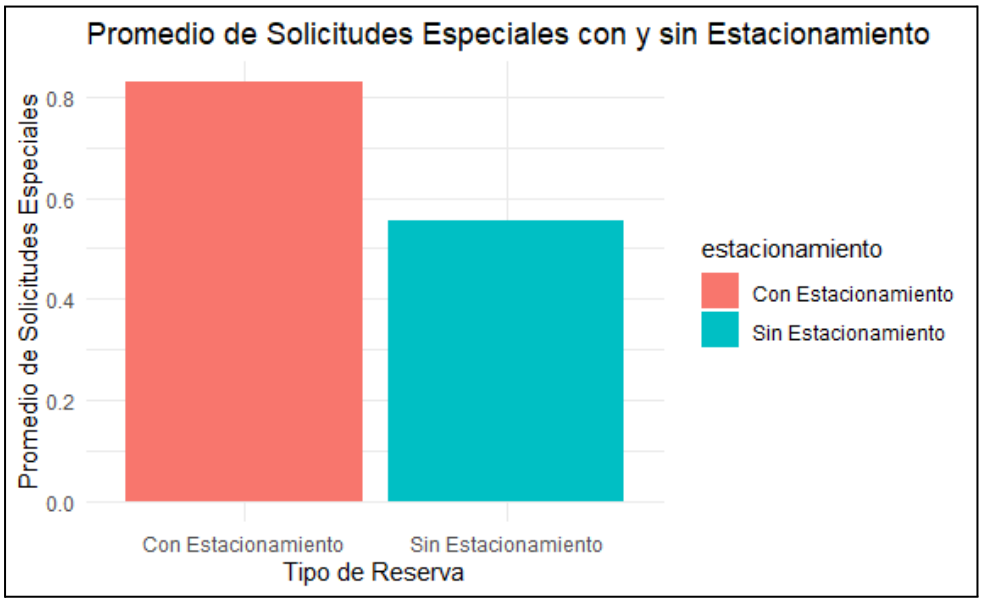

Ilustración 33, gráfico de barras.

Para esta pregunta, se utilizó un gráfico de barras para mostrar las reservas que requieren o no requieren estacionamiento, de igual forma otro gráfico para representar la tasa de cancelación por reservas. Por último, mostrar el promedio de solicitudes especiales con el propósito de determinar si los huéspedes que solicitan estacionamiento tienden a hacer más solicitudes especiales.
Con los resultados mostrados, se concluye que la cantidad de reservas que solicitaron espacios de estacionamiento es menor a las que no solicitaron, también la mayoría de reservas con solicitudes especiales han solicitado espacios de estacionamiento. Por ende, es importante contar con espacios de estacionamiento.

Pregunta clave 7: ¿En qué meses del año se producen más cancelaciones de reservas?

Para responder esta pregunta, desarrollamos un código en R que analiza las cancelaciones realizadas por mes y año.

```r
#---------------------------------------------------------------------
# Pregunta clave 7
# Cancelaciones por mes
par(mfrow = c(1, 1))
rm(canceladas)
canceladas <- data_TA[data_TA$is_canceled == 1, ] #Filtrar

# Reservaciones canceladas por mes diferenciando entre años
cuenta2 = table(canceladas$arrival_date_year, canceladas$arrival_date_month)
graf2 <- barplot(cuenta2, beside = TRUE,
        col = c("blue", "red", "green"), #Cada año posee un color
        main = "Cancelados por mes y año",
        ylab = "Cantidad de cancelaciones",
        xlab = "Mes de cancelacion",
        legend = rownames(cuenta2),
        args.legend = list(title = "Año", x = "topright"),
        las = 2,
        ylim = c(0, max(cuenta2) * 1.2))
text(graf2, cuenta2 + 2, labels = as.vector(cuenta2), cex = 0.8, pos = 3)
```

Ilustración 34, propuesta de solución.

Resultado:

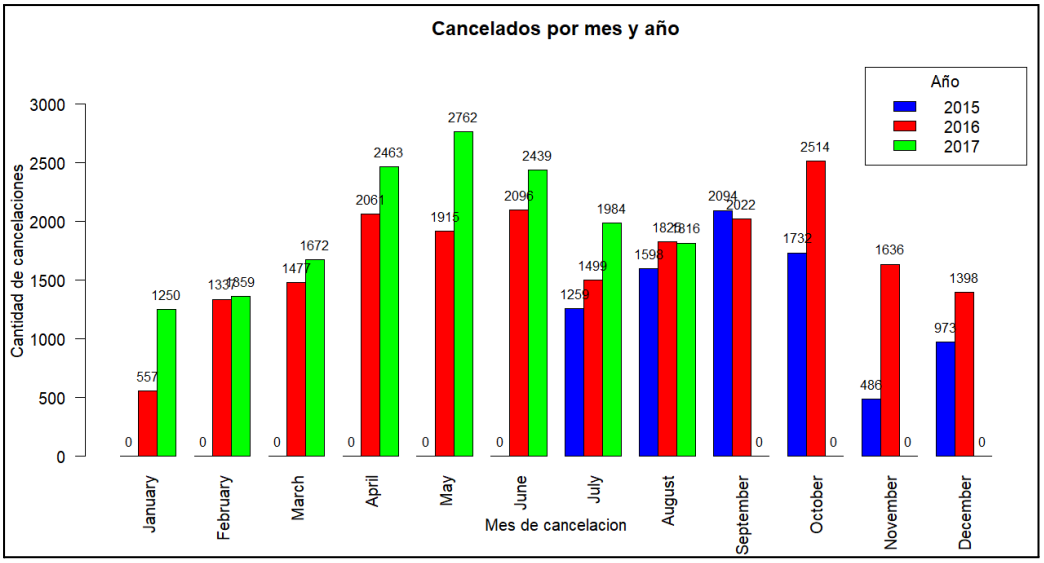

Ilustración 35,gráfico de barras por año.

Como se puede apreciar en la figura, existen barras con altura igual a 0. Esto se debe a que la base de datos solo posee datos desde julio del 2015 hasta agosto del 2017. Debido a esto, para continuar con el análisis, separamos tres intervalos de meses: primer intervalo desde enero hasta junio, segundo intervalo desde julio hasta agosto y tercer intervalo desde septiembre hasta diciembre.

El primer intervalo, contiene reservas canceladas en los meses descritos de 2017 y 2016. Si calculamos el promedio de cancelaciones por cada mes, obtenemos:

- Enero: (557 + 1250) / 2 = 903.5
- Febrero: (1337 + 1359) / 2 = 1348
- Marzo: (1477 + 1672) / 2 = 1574.5
- Abril: (2061 + 2463) / 2 = 2262
- Mayo: (1915 + 2762) / 2 = 2338.5
- Junio: (2096 + 2439) / 2 = 2267.5

El segundo intervalo, contiene las reservas canceladas de junio y agosto de los 3 años. A partir del mismo análisis obtenemos:

- Julio: (1259 + 1499 + 1984) / 3 = 1580.67
- Agosto: (1598 + 1825 + 1816) / 3 = 1746.3

Para el tercer intervalo, solo se toma en cuenta los años 2016 y 2015. El promedio de reservas canceladas por mes sería:

- Septiembre: (2094 + 2022) / 2 = 2058
- Octubre: (1732 + 2514) / 2 = 2123
- Noviembre: (486 + 1636) / 2 = 1061
- Diciembre: (973 + 1398) / 2 = 1185.5

Teniendo en cuenta los resultados, observamos que la temporada con mayor promedio de reservas canceladas es entre los meses de abril, mayo y junio.

Asimismo, se observan altas cantidades de reservas canceladas entre septiembre y octubre. Una información adicional rescatable es la temporada con menos cancelaciones de reserva, la cual ocurre entre los meses de noviembre, diciembre y enero.

Conclusiones preliminares

Patrones y Tendencias Observadas en las Reservas No Canceladas

Tendencias estacionales: Las reservas alcanzan su punto máximo en verano y disminuyen drásticamente en invierno. Esto muestra una fuerte correlación entre la estacionalidad y la demanda de reservas.

Los meses de julio (7,919 reservas) y agosto (8,638 reservas) muestran los picos más altos en reservas. Estos meses representan la temporada alta de turismo, donde las vacaciones de verano impulsan la demanda. Las familias y turistas suelen aprovechar el buen clima y las vacaciones escolares para viajar.

Consistencia en meses intermedios: Los meses de primavera y otoño presentan una demanda moderada y constante, lo que indica que hay oportunidades para mantener las ventas estables fuera de la temporada alta.

Meses como marzo, mayo, junio, septiembre, octubre muestran una demanda estable con entre 6,000 y 7,000 reservas. Estos meses no corresponden a vacaciones masivas, pero mantienen un flujo constante de viajeros, posiblemente relacionados con viajes de negocios o turismo menos estacional.

Para maximizar ingresos, se recomienda ajustar las tarifas al alza en julio y agosto, aprovechando el pico de demanda durante la temporada alta. En los meses de menor demanda, como enero y diciembre, es recomendable ofrecer promociones y descuentos para incentivar las reservas y aumentar la ocupación. Finalmente, en los meses intermedios como marzo, mayo y octubre, se sugiere mantener estrategias de precios competitivos para aprovechar la demanda estable y asegurar una ocupación constante a lo largo del año.

La preferencia por el City Hotel sobre el Resort Hotel

La preferencia por los City Hotels sobre los Resort Hotels indica que los viajeros valoran la conveniencia, la ubicación estratégica y la accesibilidad a atracciones urbanas.Además, los City Hotels pueden ofrecer tarifas más competitivas, especialmente en temporadas bajas, lo que contribuye a su popularidad.

Se recomienda que los City Hotels mejoren la experiencia del huésped invirtiendo en servicios personalizados y comodidades adecuadas para viajeros de negocios y turistas. Además, deben desarrollar estrategias de marketing que destaquen su ubicación y accesibilidad a atracciones locales, así como formar alianzas con empresas locales para ofrecer paquetes atractivos.

Reservas con niños y/o bebés

El análisis revela que, de las reservas realizadas, un total de 9,332 se clasifican en categorías que incluyen niños y/o bebés, lo que indica una significativa demanda por parte de familias que viajan con niños. Esta cifra destaca la importancia de adaptar los servicios y las instalaciones de los hoteles para satisfacer las necesidades de este segmento del mercado, ya que representan una parte considerable de la clientela.

Se sugiere que los hoteles implementen ofertas y paquetes específicos dirigidos a familias que viajan con niños y bebés. Esto podría incluir servicios como habitaciones familiares, actividades para niños, menús adaptados y facilidades como cunas o tronas. Además, es crucial realizar campañas de marketing que resalten estas características para atraer a este grupo demográfico y mejorar la experiencia del huésped.

Importancia de contar con estacionamientos

Los resultados indican que, aunque la cantidad de reservas que requieren estacionamiento es menor en comparación con las que no lo requieren, una proporción significativa de los huéspedes que efectúan solicitudes especiales también optan por espacios de estacionamiento. Esto sugiere que los huéspedes que valoran la comodidad y las facilidades adicionales, como el estacionamiento, son más propensos a realizar solicitudes especiales, lo que podría estar relacionado con su experiencia general en el hotel. Por lo tanto, es esencial contar con espacios de estacionamiento suficientes, ya que no solo satisface una necesidad básica, sino que también puede influir positivamente en la satisfacción del cliente y en la percepción de calidad del servicio del hotel.

Mes donde se produce más cancelaciones masivas

El análisis de las cancelaciones de reservas revela patrones significativos a lo largo del año. Se observa que los meses de abril, mayo y junio presentan el mayor promedio de cancelaciones, lo que sugiere que, durante la temporada de primavera y el inicio del verano, los huéspedes son más propensos a cancelar sus reservas. Además, se identifica un segundo pico de cancelaciones en septiembre y octubre, posiblemente relacionado con la transición de la temporada alta de verano a un periodo más tranquilo. En contraste, noviembre, diciembre y enero muestran la

menor cantidad de cancelaciones, lo que podría indicar que los viajeros prefieren mantener sus reservas durante la temporada festiva y el inicio del nuevo año. Estos hallazgos son cruciales para la gestión de reservas y estrategias de marketing, ya que permiten a los hoteles anticipar períodos de mayor riesgo de cancelaciones y ajustar sus políticas y promociones en consecuencia.

Para mitigar el impacto de las cancelaciones, se sugiere implementar políticas flexibles de reservas que ofrezcan incentivos para mantener las reservas durante los meses de alta cancelación, como abril a junio y septiembre a octubre. Estas políticas pueden incluir tarifas no reembolsables con descuentos atractivos o la opción de reprogramar las reservas sin penalización.