Universidad Peruana de Ciencias Aplicadas

FACULTAD DE INGENIERÍA

Ciclo: Quinto
Curso: Fundamento de Data Science
Sección: 258
Docente: Nérida Isabel Manrique Tunque

INFORME DEL TB1
"Análisis EDA: Hotel booking demand"

Integrantes:

Cahuana López, Leicy Cristell (U20231E777)
Huamán Cortez, Anabella Karina (U202216171)
Mercado De La Rosa, Luis Marcelo (U20211B656)
Montenegro López, Valentina Étoile (U202312021)

2025 - 01

ÍNDICE

1. CASO DE ANÁLISIS

1.1. Origen de Datos

El conjunto de datos utilizado se titula "Hotel Booking Demand" y fue originalmente publicado por Nuno Antonio, Ana de Almeida y Luis Nunes en 2019, a través del portal ScienceDirect. La versión base está disponible en la plataforma Kaggle y contiene información real recopilada de reservas de dos hoteles ubicados en Portugal, uno de tipo City Hotel y otro Resort Hotel (Duong, 2023).

Para efectos del presente trabajo, el dataset ha sido modificado intencionalmente por el equipo docente del curso, añadiendo valores faltantes (NA) y datos atípicos (outliers) con el propósito de evaluar competencias de preprocesamiento y análisis en ciencia de datos. Los datos provienen de registros administrativos, lo que les otorga un grado alto de fiabilidad.

1.2 Casos de Usos Aplicables

¿Quién podría estar interesado en este análisis?

- Gerentes de hoteles y cadenas hoteleras.
- Agencias de viajes y operadores turísticos.
- Departamentos de marketing y ventas.
- Desarrolladores de software de gestión hotelera.

¿Qué problemas o necesidades resuelve?

- Optimización de la ocupación hotelera, ajustando estrategias según la demanda histórica.

- Predicción de cancelaciones y mejora en la política de reservas.
- Identificación de temporadas altas, medias y bajas.
- Determinación del impacto de servicios adicionales (ej. estacionamiento, comidas).
- Segmentación del cliente para campañas publicitarias personalizadas.

2. CONJUNTO DE DATOS (DATA SET)

El conjunto de datos hotel_bookings.csv contiene 119,390 registros y 32 variables. Cada registro representa una reserva en uno de los dos hoteles analizados. Las variables incluyen características del huésped, detalles de la reserva, fechas, duración de la estancia, información sobre cancelaciones, entre otros.

Variable	Tipo	Descripción
hotel	Categórico	Tipo de hotel: "City Hotel" o "Resort Hotel".
is_canceled	Binaria (Factor)	1 si fue cancelada, 0 si no.
lead_time	Numérico	Días entre la reserva y la llegada.
arrival_date_year	Numérico	Año de llegada.
arrival_date_month	Categórico	Mes de llegada (texto).
arrival_date_week_number	Numérico	Semana del año en que se llega.
arrival_date_day_of_month	Numérico	Día del mes de llegada.
stays_in_weekend_nights	Numérico	Noches de fin de semana.

stays_in_week_nights	Numérico	Noches entre semana.
adults	Numérico	Número de adultos.
children	Numérico	Número de niños.
babies	Numérico	Número de bebés.
meal	Categórico	Tipo de comida incluida (BB, HB, FB, SC, etc.).
country	Categórico	País de origen del huésped.
market_segment	Categórico	Canal de comercialización.
distribution_channel	Categórico	Medio por el cual se realizó la reserva.
is_repeated_guest	Binaria (Factor)	1 si es huésped recurrente, 0 si es la primera vez.
previous_cancellations	Numérico	Cancelaciones previas del mismo cliente.
previous_bookings_not_canceled	Numérico	Reservas anteriores no canceladas del mismo cliente.
reserved_room_type	Categórico	Tipo de habitación solicitada.
assigned_room_type	Categórico	Tipo de habitación asignada.
booking_changes	Numérico	Cambios realizados después de la reserva inicial.
deposit_type	Categórico	Tipo de depósito: "No Deposit", "Non

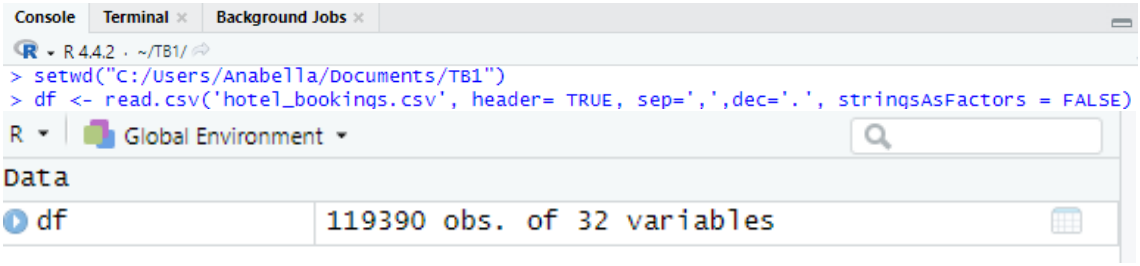
		Refund", "Refundable".
agent	Categórico	ID del agente de reserva.
company	Categórico	ID de la empresa (si aplica).
days_in_waiting_list	Numérico	Días en lista de espera.
customer_type	Categórico	Tipo de cliente: Transient, Contract, etc.
adr	Numérico	Tarifa promedio diaria por habitación reservada.
required_car_parking_spaces	Numérico	Cantidad de espacios de estacionamiento solicitados.
total_of_special_requests	Numérico	Cantidad de solicitudes especiales.
reservation_status	Categórico	Estado final de la reserva (Check-Out, Canceled, No-Show).
reservation_status_date	Fecha	Fecha del último estado asignado a la reserva.

3. ANÁLISIS EXPLORATORIO DE DATOS (EDA)

CARGAR DATOS

```
#Limpieza de datos

rm(list=ls(all=TRUE))
graphics.off()
```

```
cat("\014"))

#Configuración del directorio de trabajo
setwd("C:/Users/Anabella/Documents/TB1")

#Carga del dataset con los parámetros necesarios
df <- read.csv('hotel_bookings.csv', header= TRUE,
sep=',',dec='.', stringsAsFactors = FALSE)
```




```
#Visualización de las primeras filas del dataset
Head(df)
```

```
> head(df)
         hotel is_canceled lead_time arrival_date_year arrival_date_month
1 Resort Hotel           0       342              2015               July
2 Resort Hotel           0       737              2015               July
3 Resort Hotel           0         7              2015               July
4 Resort Hotel           0        13              2015               July
5 Resort Hotel           0        14              2015               July
7 Resort Hotel           0         0              2015               July
  arrival_date_week_number arrival_date_day_of_month stays_in_weekend_nights
1                       27                         1                       0
2                       27                         1                       0
3                       27                         1                       0
4                       27                         1                       0
5                       27                         1                       0
7                       27                         1                       0
  stays_in_week_nights adults children babies meal country market_segment
1                    0      2        0      0   BB     PRT         Direct
2                    0      2        0      0   BB     PRT         Direct
3                    1      1        0      0   BB     GBR         Direct
4                    1      1        0      0   BB     GBR      Corporate
5                    2      2        0      0   BB     GBR      Online TA
7                    2      2        0      0   BB     PRT         Direct
  distribution_channel is_repeated_guest previous_cancellations
1               Direct                 0                       0
2               Direct                 0                       0
3               Direct                 0                       0
4            Corporate                 0                       0
5              TA/TO                   0                       0
7               Direct                 0                       0
  previous_bookings_not_canceled reserved_room_type assigned_room_type booking_changes
1                              0                  C                  C               3
2                              0                  C                  C               4
3                              0                  A                  C               0
4                              0                  A                  A               0
5                              0                  A                  A               0
7                              0                  C                  C               0
  deposit_type agent company days_in_waiting_list customer_type adr
1   No Deposit  NULL    NULL                    0     Transient   0
2   No Deposit  NULL    NULL                    0     Transient   0
3   No Deposit  NULL    NULL                    0     Transient  75
4   No Deposit   304    NULL                    0     Transient  75
5   No Deposit   240    NULL                    0     Transient  98
7   No Deposit  NULL    NULL                    0     Transient 107
  required_car_parking_spaces total_of_special_requests reservation_status
1                           0                         0          Check-Out
2                           0                         0          Check-Out
3                           0                         0          Check-Out
4                           0                         0          Check-Out
5                           0                         1          Check-Out
7                           0                         0          Check-Out
  reservation_status_date
1              2015-07-01
2              2015-07-01
3              2015-07-02
4              2015-07-02
5              2015-07-03
7              2015-07-03
```

INSPECCIONAR DATOS

```
#Estructura de las variables

str(df)
```

```
> str(df)         # Ver estructura de las columnas y tipos de datos
'data.frame':   119390 obs. of  32 variables:
 $ hotel                         : chr  "Resort Hotel" "Resort Hotel" "Resort Hotel" "Resort Hot
el" ...
 $ 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  0 0 0 0 0 0 0 0 0 0 ...
 $ stays_in_week_nights          : int  0 0 1 1 2 2 2 2 3 3 ...
 $ adults                        : int  2 2 1 1 2 2 2 2 2 2 ...
 $ children                      : int  0 0 0 0 0 0 0 0 0 0 ...
 $ babies                        : int  0 0 0 0 0 0 0 0 0 0 ...
 $ meal                          : 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  "2015-07-01" "2015-07-01" "2015-07-02" "2015-07-02" ...
```

#Dimensiones del dataframe

dim(df)

```
> dim(df)         # Ver cuántas filas y columnas tiene
[1] 119390     32
```

#Nombres de las columnas

Names(df)

```
> names(df)       # Ver nombre de las variables
 [1] "hotel"                         "is_canceled"
 [3] "lead_time"                     "arrival_date_year"
 [5] "arrival_date_month"            "arrival_date_week_number"
 [7] "arrival_date_day_of_month"     "stays_in_weekend_nights"
 [9] "stays_in_week_nights"          "adults"
[11] "children"                      "babies"
[13] "meal"                          "country"
[15] "market_segment"                "distribution_channel"
[17] "is_repeated_guest"             "previous_cancellations"
[19] "previous_bookings_not_canceled" "reserved_room_type"
[21] "assigned_room_type"            "booking_changes"
[23] "deposit_type"                  "agent"
[25] "company"                       "days_in_waiting_list"
[27] "customer_type"                 "adr"
[29] "required_car_parking_spaces"   "total_of_special_requests"
[31] "reservation_status"            "reservation_status_date"
```

#Estadísticas descriptivas

summary(df)

```
> summary(df)     # Ver resumen estadístico de las variables
    hotel           is_canceled       lead_time       arrival_date_year arrival_date_month
 Length:119390      Min.   :0.0000   Min.   :  0.0   Min.   :2015      Length:119390
 Class :character   1st Qu.:0.0000   1st Qu.: 18.0   1st Qu.:2016      Class :character
 Mode  :character   Median :0.0000   Median : 69.0   Median :2016      Mode  :character
                    Mean   :0.3704   Mean   :104.0   Mean   :2016
                    3rd Qu.:1.0000   3rd Qu.:160.0   3rd Qu.:2017
                    Max.   :1.0000   Max.   :737.0   Max.   :2017

 arrival_date_week_number arrival_date_day_of_month stays_in_weekend_nights
 Min.   : 1.00            Min.   : 1.0              Min.   : 0.0000
 1st Qu.:16.00            1st Qu.: 8.0              1st Qu.: 0.0000
 Median :28.00            Median :16.0              Median : 1.0000
 Mean   :27.17            Mean   :15.8              Mean   : 0.9276
 3rd Qu.:38.00            3rd Qu.:23.0              3rd Qu.: 2.0000
 Max.   :53.00            Max.   :31.0              Max.   :19.0000

 stays_in_week_nights     adults          children          babies
 Min.   : 0.0         Min.   : 0.000   Min.   : 0.0000   Min.   : 0.000000
 1st Qu.: 1.0         1st Qu.: 2.000   1st Qu.: 0.0000   1st Qu.: 0.000000
 Median : 2.0         Median : 2.000   Median : 0.0000   Median : 0.000000
 Mean   : 2.5         Mean   : 1.856   Mean   : 0.1039   Mean   : 0.007949
 3rd Qu.: 3.0         3rd Qu.: 2.000   3rd Qu.: 0.0000   3rd Qu.: 0.000000
 Max.   :50.0         Max.   :55.000   Max.   :10.0000   Max.   :10.000000
                                       NA's   :4
     meal             country          market_segment     distribution_channel
 Length:119390      Length:119390      Length:119390      Length:119390
 Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character

 is_repeated_guest previous_cancellations previous_bookings_not_canceled
 Min.   :0.00000   Min.   : 0.00000       Min.   : 0.0000
 1st Qu.:0.00000   1st Qu.: 0.00000       1st Qu.: 0.0000
 Median :0.00000   Median : 0.00000       Median : 0.0000
 Mean   :0.03191   Mean   : 0.08712       Mean   : 0.1371
 3rd Qu.:0.00000   3rd Qu.: 0.00000       3rd Qu.: 0.0000
 Max.   :1.00000   Max.   :26.00000       Max.   :72.0000

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

    agent             company          days_in_waiting_list customer_type
 Length:119390      Length:119390      Min.   :  0.000      Length:119390
 Class :character   Class :character   1st Qu.:  0.000      Class :character
 Mode  :character   Mode  :character   Median :  0.000      Mode  :character
                                       Mean   :  2.321
                                       3rd Qu.:  0.000
                                       Max.   :391.000

      adr          required_car_parking_spaces total_of_special_requests
 Min.   :  -6.38   Min.   :0.00000             Min.   :0.0000
 1st Qu.:  69.29   1st Qu.:0.00000             1st Qu.:0.0000
 Median :  94.58   Median :0.00000             Median :0.0000
 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 reservation_status_date
 Length:119390      Length:119390
 Class :character   Class :character
 Mode  :character   Mode  :character
```

Análisis de tipos de variables:

Según la estructura del DataFrame, el conjunto de datos tiene 119,390 registros (filas) y 32 variables (columnas).

1. Variables de tipo chr (texto)

Estas variables se presentan inicialmente como texto (chr), pero muchas de ellas corresponden a datos categóricos, por lo que deben ser convertidas a tipo factor para un análisis adecuado. Además, una de ellas representa fechas y debe ser transformada a tipo Date.

- `$hotel` (convertir a factor)
- `$arrival_date_month`
- `$meal` (convertir a factor)
- `$country`
- `$market_segment` (convertir a factor)
- `$distribution_channel` (convertir a factor)
- `$reserved_room_type` (convertir a factor)
- `$assigned_room_type` (convertir a factor)
- `$deposit_type` (convertir a factor)
- `$agent` (convertir a factor)
- `$company` (convertir a factor)
- `$customer_type` (convertir a factor)
- `$reservation_status` (convertir a factor)
- $reservation_status_date *(*esta como tipo texto, pero debe transformarse a tipo Date ya que representa una fecha*)*

2. Variables de tipo int o num (numéricas)

Estas variables se interpretan como numéricas. Algunas representan cantidades, y otras son binarias (0 o 1), por lo que estas últimas deben convertirse a factor para un análisis adecuado.

- `$is_canceled` (binaria, convertir a factor)
- `$is_repeated_guest` (binaria, convertir a factor)
- `$lead_time`
- `$arrival_date_year`
- `$arrival_date_week_number`

- $arrival_date_day_of_month
- $stays_in_weekend_nights
- $stays_in_week_nights
- $adults
- $children
- $babies
- $previous_cancellations
- $previous_bookings_not_canceled
- $booking_changes
- $days_in_waiting_list
- $adr
- $required_car_parking_spaces
- $total_of_special_requests

```
#Detectar registros duplicados y eliminarlos

sum(duplicated(df))
```

```
> sum(duplicated(df)) #detectar registros duplicados
[1] 31994
```

```
df <- df[!duplicated(df), ]
```

```
> df <- df[!duplicated(df), ] #Eliminamos los registros duplicados
```

Se identificaron 31,994 registros duplicados, los cuales fueron eliminados para garantizar la calidad y eficiencia del análisis.

```
#Convertir de texto a factor

df$hotel<- as.factor(df$hotel)
df$arrival_date_month <- as.factor(df$arrival_date_month)
df$agent <- as.factor(df$agent)
df$company <- as.factor(df$company)
df$reservation_status <- as.factor(df$reservation_status)
df$is_canceled <- as.factor(df$is_canceled)
df$meal <- as.factor(df$meal)
```

```
df$is_repeated_guest <- as.factor(df$is_repeated_guest)

df$reserved_room_type <- as.factor(df$reserved_room_type)

df$assigned_room_type <- as.factor(df$assigned_room_type)

df$deposit_type <- as.factor(df$deposit_type)

df$customer_type <- as.factor(df$customer_type)

df$market_segment <- as.factor(df$market_segment)

df$distribution_channel <-
as.factor(df$distribution_channel)
```

```
> # Convertir texto a factor
> df$hotel               <- as.factor(df$hotel)
> df$arrival_date_month  <- as.factor(df$arrival_date_month)
> df$agent               <- as.factor(df$agent)
> df$company             <- as.factor(df$company)
> df$reservation_status  <- as.factor(df$reservation_status)
> df$is_canceled         <- as.factor(df$is_canceled)
> df$meal                <- as.factor(df$meal)
> df$is_repeated_guest   <- as.factor(df$is_repeated_guest)
> df$reserved_room_type  <- as.factor(df$reserved_room_type)
> df$assigned_room_type  <- as.factor(df$assigned_room_type)
> df$deposit_type        <- as.factor(df$deposit_type)
> df$customer_type       <- as.factor(df$customer_type)
> df$market_segment      <- as.factor(df$market_segment)
> df$distribution_channel <- as.factor(df$distribution_channel)
```

```
#Comprobando los cambios de estas variables

lapply(df[c("hotel","arrival_date_month","agent",
"company","reservation_status", "is_canceled", "meal",
"is_repeated_guest", "reserved_room_type",
"assigned_room_type", "deposit_type", "customer_type",
"market_segment", "distribution_channel")], summary)
```

```
> lapply(df[c("hotel","arrival_date_month","agent", "company","reservation_status", "is_canceled",
"meal", "is_repeated_guest",
+             "reserved_room_type", "assigned_room_type", "deposit_type",
+             "customer_type", "market_segment", "distribution_channel")], summary)
$hotel
  City Hotel Resort Hotel
       53428        33968

$arrival_date_month
    April    August  December  February   January      July      June     March       May
     7908     11257      5131      6098      4693     10057      7765      7513      8355
 November   October September
     4995      6934      6690

$agent
      9     240    NULL      14       7     250     241      28       8       1       6      40
  28759   13028   12193    3349    3300    2779    1644    1502    1383    1232    1117     986
    314     242      83      85     243     171      27       3      22      11      15     196
    844     722     614     524     477     402     395     363     345     318     303     281
    177      96     138      37      16     229       5      10      21      42     115     156
    277     272     257     225     220     218     216     214     199     194     192     187
     26     175     195      86     273     251     134     143     298     152     168      19
    187     182     181     180     179     175     168     163     160     156     153     151
    315      12       2     147      95     410      20     146     142      30      94     330
    151     133     129     128     125     121     117     112     111     106     103      99
     89      29     191      52      69     159      13      36      17     464      39      75
     90      84      82      82      81      80      77      77      76      75      74      70
    132     339      38      98     185     118     253      34     220     531     234     208
     68      68      67      66      65      64      63      62      61      60      59      54
    157     181     184      56      91     104     155      58      71     436     468      79
     52      52      52      49      49      48      48      48      48      47      43      43
    127     248      87 (Other)
     41      40      39    1991

$company
   NULL      40     223      45     153     154     219     174     281     233      51     405
  82137     851     503     238     206     133     131     121     119      95      80      77
     94      47     331     169     135     110      91      62      67     270     113     148
     76      62      60      53      52      48      46      44      44      43      36      36
    280       9     195     498     204     269      86      20     218     238      72     221
     36      36      34      34      33      33      32      31      31      30      29      27
     68     178     307     179     418      46      38      81     216     144     227     150
     26      25      25      24      24      24      22      22      21      20      20      18
    286     342      88     163     251     290     292     337     103     183     365     242
     18      18      18      17      17      17      17      17      16      16      16      15
    308     408      78     209     263     343     477     485      82     112      12     120
     15      15      15      14      14      14      14      14      14      13      13      13
    197      31     390     428      43      92     274     291     338     380     396     465
     13      13      13      13      13      13      12      12      12      12      12      12
     99     143     186     268     329     399     525     203     323     355     356     371
     12      11      11      11      11      11      11      10      10      10      10      10
    435     450     108 (Other)
     10      10       9     771

$reservation_status
 Canceled Check-Out   No-Show
    23011     63371      1014

$is_canceled
    0     1
63371 24025
```

```
$meal
       BB        FB        HB        SC Undefined
    67978       360      9085      9481       492

$is_repeated_guest
    0     1
83981  3415

$reserved_room_type
    A     B     C     D     E     F     G     H     L     P
56552   999   915 17398  6049  2823  2052   596     6     6

$assigned_room_type
    A     B     C     D     E     F     G     H     I     K     L     P
46313  1820  2165 22432  7195  3627  2498   706   357   276     1     6

$deposit_type
No Deposit Non Refund Refundable
     86251       1038        107

$customer_type
       Contract           Group       Transient Transient-Party
           3139             544           71986           11727

$market_segment
     Aviation Complementary     Corporate        Direct        Groups Offline TA/TO
          227           702          4212         11804          4942         13889
    Online TA     Undefined
        51618             2

$distribution_channel
Corporate    Direct       GDS     TA/TO Undefined
     5081     12988       181     69141         5
```

```
#Conversion de texto a date

df$reservation_status_date <-
as.Date(df$reservation_status_date)
```

```
> # Convertir fecha
> df$reservation_status_date <- as.Date(df$reservation_status_date)
```

```
#Comprobando el cambio de esta variable

str(df$reservation_status_date)
```

```
> #comprobando el cambio de esta variable
> str(df$reservation_status_date)
 Date[1:87396], format: "2015-07-01" "2015-07-01" "2015-07-02" "2015-07-02" "2015-07-03" "2015-07-0
3" ...
```

PRE-PROCESAR DATOS

Resumir Estadísticas Básicas:

Este resumen inicial nos dará una idea general del comportamiento de las variables.

Comprobamos un resumen breve con “`summary(df)`”

```
> summary(df)
         hotel       is_canceled   lead_time      arrival_date_year arrival_date_month
 City Hotel  :53428   0:63371     Min.    :  0.00   Min.   :2015      August :11257
 Resort Hotel:33968   1:24025     1st Qu.: 11.00   1st Qu.:2016      July   :10057
                                  Median : 49.00   Median :2016      May    : 8355
                                  Mean   : 79.89   Mean   :2016      April  : 7908
                                  3rd Qu.:125.00   3rd Qu.:2017      June   : 7765
                                  Max.   :737.00   Max.   :2017      March  : 7513
                                                                     (Other):34541
 arrival_date_week_number arrival_date_day_of_month stays_in_weekend_nights
 Min.    : 1.00           Min.    : 1.00            Min.    : 0.000
 1st Qu.:16.00            1st Qu.: 8.00            1st Qu.: 0.000
 Median :27.00            Median :16.00            Median : 1.000
 Mean   :26.84            Mean   :15.82            Mean   : 1.005
 3rd Qu.:37.00            3rd Qu.:23.00            3rd Qu.: 2.000
 Max.   :53.00            Max.   :31.00            Max.   :19.000

 stays_in_week_nights     adults          children           babies                meal
 Min.    : 0.000       Min.    : 0.000   Min.    : 0.0000   Min.    : 0.00000   BB       :67978
 1st Qu.: 1.000        1st Qu.: 2.000   1st Qu.: 0.0000   1st Qu.: 0.00000   FB       :  360
 Median : 2.000        Median : 2.000   Median : 0.0000   Median : 0.00000   HB       : 9085
 Mean   : 2.625        Mean   : 1.876   Mean   : 0.1386   Mean   : 0.01082   SC       : 9481
 3rd Qu.: 4.000        3rd Qu.: 2.000   3rd Qu.: 0.0000   3rd Qu.: 0.00000   Undefined:  492
 Max.   :50.000        Max.   :55.000   Max.   :10.0000   Max.   :10.00000
                                        NA's   :4
    country            market_segment  distribution_channel is_repeated_guest
 PRT     :27453   Online TA     :51618   Corporate: 5081      0:83981
 GBR     :10433   Offline TA/TO:13889   Direct   :12988      1: 3415
 FRA     : 8837   Direct        :11804   GDS      :  181
 ESP     : 7252   Groups        : 4942   TA/TO    :69141
 DEU     : 5387   Corporate     : 4212   Undefined:    5
 ITA     : 3066   Complementary:  702
 (Other):24968   (Other)       :  229

 previous_cancellations previous_bookings_not_canceled reserved_room_type assigned_room_type
 Min.    : 0.00000      Min.    : 0.000                A      :56552      A      :46313
 1st Qu.: 0.00000      1st Qu.: 0.000                D      :17398      D      :22432
 Median : 0.00000      Median : 0.000                E      : 6049      E      : 7195
 Mean   : 0.03041      Mean   : 0.184                F      : 2823      F      : 3627
 3rd Qu.: 0.00000      3rd Qu.: 0.000                G      : 2052      G      : 2498
 Max.   :26.00000      Max.   :72.000                B      :  999      C      : 2165
                                                     (Other): 1523      (Other): 3166
 booking_changes       deposit_type       agent          company      days_in_waiting_list
 Min.    : 0.0000   No Deposit:86251   9      :28759   NULL   :82137   Min.    :  0.0000
 1st Qu.: 0.0000   Non Refund: 1038   240    :13028   40     :  851   1st Qu.:  0.0000
 Median : 0.0000   Refundable:  107   NULL   :12193   223    :  503   Median :  0.0000
 Mean   : 0.2716                      14     : 3349   45     :  238   Mean   :  0.7496
 3rd Qu.: 0.0000                      7      : 3300   153    :  206   3rd Qu.:  0.0000
 Max.   :21.0000                      250    : 2779   154    :  133   Max.   :391.0000
                                      (Other):23988   (Other): 3328
        customer_type        adr          required_car_parking_spaces
 Contract       : 3139   Min.    :  -6.38   Min.    :0.00000
 Group          :  544   1st Qu.:  72.00   1st Qu.:0.00000
 Transient      :71986   Median :  98.10   Median :0.00000
 Transient-Party:11727   Mean   : 106.34   Mean   :0.08423
                         3rd Qu.: 134.00   3rd Qu.:0.00000
                         Max.   :5400.00   Max.   :8.00000

 total_of_special_requests reservation_status reservation_status_date
 Min.    :0.0000           Canceled :23011    Min.    :2014-10-17
 1st Qu.:0.0000           Check-Out:63371    1st Qu.:2016-03-18
 Median :0.0000           No-Show  : 1014    Median :2016-09-08
 Mean   :0.6986                              Mean   :2016-08-31
 3rd Qu.:1.0000                              3rd Qu.:2017-03-05
 Max.   :5.0000                              Max.   :2017-09-14
```

Notamos que podemos juntar “arrival_date_day_of_month”, “arrival_date_year” y “arrival_date_month”

```
#Unimos las fechas
#Aseguremonos que la fecha sea un numero para poder unirlo
df$arrival_date_day_of_month <-
as.numeric(df$arrival_date_day_of_month)
```

```
df$arrival_date_year <- as.numeric(df$arrival_date_year)
# Convertir el nombre del mes a número
df$arrival_date_month <- match(df$arrival_date_month,
month.name)

#Combinar las columnas de día, mes y año para formar una
fecha completa
df$arrival_date <- paste(df$arrival_date_year,
df$arrival_date_month, df$arrival_date_day_of_month, sep = "-
")

# Convertir el texto a una fecha (tipo Date)
df$arrival_date <- as.Date(df$arrival_date, format = "%Y-%m-
%d")
```

```
> summary(df$arrival_date)
        Min.      1st Qu.       Median         Mean      3rd Qu.         Max.
"2015-07-01" "2016-03-13" "2016-09-06" "2016-08-28" "2017-03-18" "2017-08-31"
>
```

Ahora podemos eliminar las 3 columnas para reducir datos y haremos lo mismo con stays_in_weekend_nights y stays_in_week_night

```
> summary(df$total_nights)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   2.000   3.000   3.631   5.000  69.000
>
```

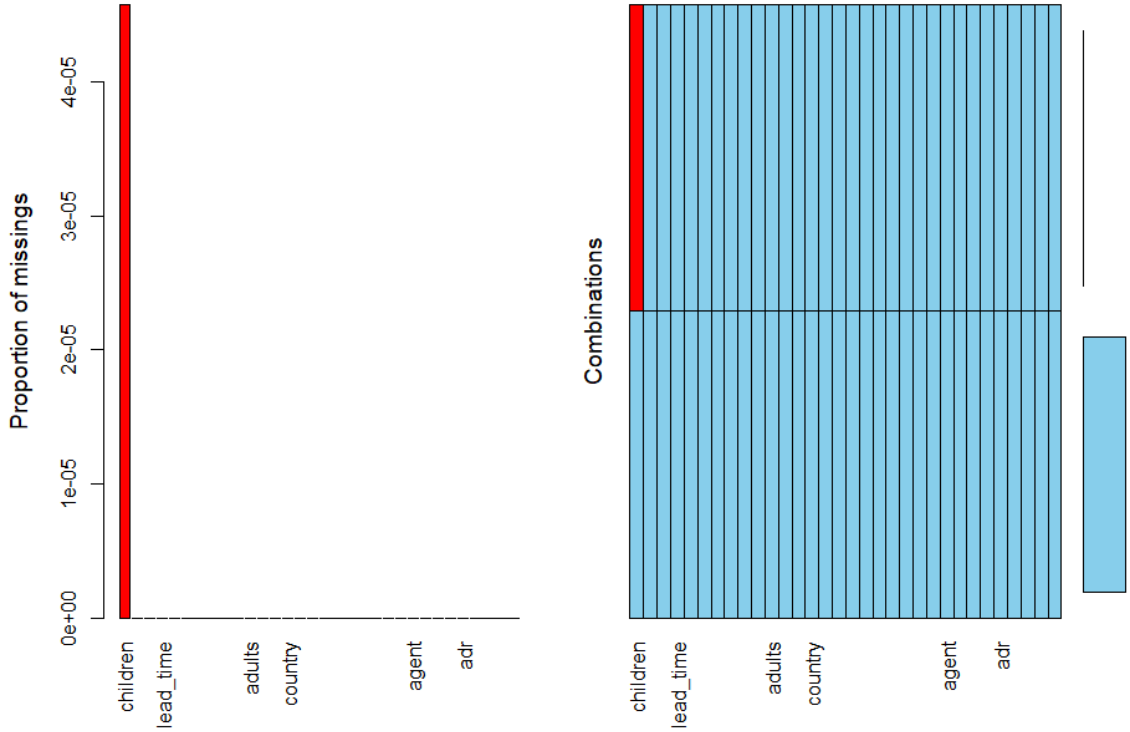
Podemos notar que hay valores "NA" en "children", además habrá que transformar en "NA" los datos que están como "Undefined" en la columna "meal", los datos "Undefined" que están en la columna "distribution_channel" y los valores "NULL" de las columnas "company" y "agent".

```
> str(df)
'data.frame':   87396 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                   : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 2 2 2 ...
 $ lead_time                     : int  342 737 7 13 14 0 9 85 75 23 ...
 $ 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 3 3 4 ...
 $ adults                        : int  2 2 1 1 2 2 2 2 2 2 ...
 $ children                      : int  0 0 0 0 0 0 0 0 0 0 ...
 $ babies                        : int  0 0 0 0 0 0 0 0 0 0 ...
 $ meal                          : Factor w/ 5 levels "BB","FB","HB",..: 1 1 1 1 1 1 2 1 3 1 ...
 $ country                       : Factor w/ 178 levels "ABW","AGO","AIA",..: 137 137 60 60 60 13
7 137 137 137 137 ...
 $ market_segment                : Factor w/ 8 levels "Aviation","Complementary",..: 4 4 4 3 7 4
4 7 6 7 ...
 $ distribution_channel          : Factor w/ 5 levels "Corporate","Direct",..: 2 2 2 1 4 2 2 4 4
4 ...
 $ is_repeated_guest             : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ previous_cancellations        : int  0 0 0 0 0 0 0 0 0 0 ...
 $ previous_bookings_not_canceled: int  0 0 0 0 0 0 0 0 0 0 ...
 $ reserved_room_type            : Factor w/ 10 levels "A","B","C","D",..: 3 3 1 1 1 3 3 1 4 5
...
 $ assigned_room_type            : Factor w/ 12 levels "A","B","C","D",..: 3 3 3 1 1 3 3 1 4 5
...
 $ 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 33
4 156 103 40 103 ...
 $ 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 0 1 1 0 0 ...
 $ reservation_status            : Factor w/ 3 levels "Canceled","Check-Out",..: 2 2 2 2 2 2 2 1
1 1 ...
 $ reservation_status_date       : Date, format: "2015-07-01" "2015-07-01" "2015-07-02" ...
```

Identificación de Datos Faltantes

Para identificar los valores “NA” utilizaremos lo siguiente.

```
#Instalación y carga del paquete VIM para visualizar valores
faltantes
install.packages("VIM", dependencies = TRUE)
library(VIM)
aggr(df, numbers = TRUE, sortVar = TRUE)
```



Solo por ahora nos da que la variable “`Children`” tiene datos faltantes que está en un 4e-03% en sus registros del dataset.

También con esto:

```
# Comprobar valores faltantes en todo el conjunto de datos
colSums(is.na(df))
```

Podemos visualizar con una estadística básica los datos faltantes.

```
> # Comprobar valores faltantes en todo el conjunto de datos
> colSums(is.na(df))
                          hotel                    is_canceled                      lead_time
                              0                              0                              0
             arrival_date_year             arrival_date_month        arrival_date_week_number
                              0                              0                              0
     arrival_date_day_of_month         stays_in_weekend_nights           stays_in_week_nights
                              0                              0                              0
                         adults                       children                         babies
                              0                              4                              0
                           meal                        country                 market_segment
                              0                              0                              0
           distribution_channel              is_repeated_guest          previous_cancellations
                              0                              0                              0
previous_bookings_not_canceled           reserved_room_type             assigned_room_type
                              0                              0                              0
                booking_changes                   deposit_type                          agent
                              0                              0                              0
                        company            days_in_waiting_list                  customer_type
                              0                              0                              0
                            adr      required_car_parking_spaces     total_of_special_requests
                              0                              0                              0
             reservation_status             reservation_status_date
                              0                              0
>
```

Pero también sabemos que hay datos que son “NULL” o datos negativos que tenemos que pasar a “NA”. Por lo que al analizar el dataset nos damos cuenta de que hay datos donde hay niños o bebes, pero no hay adultos por lo que es incoherente, así por ahora a todos los casos donde no hay adultos lo pondremos como “NA”.

```
#Reemplazar por “NA” a "adultos" 0 porque es muy raro que
vayan niños por varios dias sin adultos o bebes sin adultos.
df$adults[df$adults == "0"] <- NA
```

```
summary(df$adults)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
 1.000   2.000   2.000   1.884   2.000  55.000     385
```

También hay valores “Undefined” en “meal” por lo que lo pasaremos como “NA” para después darle un valor.

```
> summary(df$meal)
       BB         FB         HB         SC Undefined       NA's
    67978        360       9085       9481          0        492
>
```

Además, en “company” y “agent” hay datos “NULL” por lo que también lo pasaremos como NA para después darle un valor o eliminarlo.

```
#Reemplazar "NULL" (como texto) por NA en las columnas
'company' y 'agent'
df$company[df$company == "NULL"] <- NA
df$agent[df$agent == "NULL"] <- NA
summary(df$company)
summary(df$agent)
```

```
> # Reemplazar "NULL" (como texto) por NA en las columnas 'company' y 'agent'
> df$company[df$company == "NULL"] <- NA
> df$agent[df$agent == "NULL"] <- NA
> summary(df$company)
     40      223       45      153      154      219      174      281      233       51      405 
    851      503      238      206      133      131      121      119       95       80       77 
     94       47      331      169      135      110       91       62       67      270      113 
     76       62       60       53       52       48       46       44       44       43       36 
    148      280        9      195      498      204      269       86       20      218      238 
     36       36       36       34       34       33       33       32       31       31       30 
     72      221       68      178      307      179      418       46       38       81      216 
     29       27       26       25       25       24       24       24       22       22       21 
    144      227      150      286      342       88      163      251      290      292      337 
     20       20       18       18       18       18       17       17       17       17       17 
    103      183      365      242      308      408       78      209      263      343      477 
     16       16       16       15       15       15       15       14       14       14       14 
    485       82      112       12      120      197       31      390      428       43       92 
     14       14       13       13       13       13       13       13       13       13       13 
    274      291      338      380      396      465       99      143      186      268      329 
     12       12       12       12       12       12       12       11       11       11       11 
    399      525      203      323      355      356      371      435      450      108 (Other) 
     11       11       10       10       10       10       10       10       10        9      771 
   NA's 
  82137 
> summary(df$agent)
     9    240     14      7    250    241     28      8      1      6     40 
 28759  13028   3349   3300   2779   1644   1502   1383   1232   1117    986 
   314    242     83     85    243    171     27      3     22     11     15 
   844    722    614    524    477    402    395    363    345    318    303 
   196    177     96    138     37     16    229      5     10     21     42 
   281    277    272    257    225    220    218    216    214    199    194 
   115    156     26    175    195     86    273    251    134    143    298 
   192    187    187    182    181    180    179    175    168    163    160 
   152    168     19    315     12      2    147     95    410     20    146 
   156    153    151    151    133    129    128    125    121    117    112 
   142     30     94    330     89     29    191     52     69    159     13 
   111    106    103     99     90     84     82     82     81     80     77 
    36     17    464     39     75    132    339     38     98    185    118 
    77     76     75     74     70     68     68     67     66     65     64 
   253     34    220    531    234    208    157    181    184     56     91 
    63     62     61     60     59     54     52     52     52     49     49 
   104    155     58     71    436    468     79    127    248     87 (Other) 
    48     48     48     48     47     43     43     41     40     39   1991 
 NA's 
12193 
```

Asimismo, en "market segment" y "distribution_channel" hay valores "Undefined" por lo que lo volveremos "NA".

```
#Reemplazar "Undefined" por NA en la columna 'distribution_channel'
df$distribution_channel[df$distribution_channel == "Undefined"] <- NA

#Reemplazar "Undefined" por NA en el market segment
df$market_segment[df$market_segment == "Undefined"] <- NA
```

```
> # Reemplazar "Undefined" por NA en la columna 'distribution_channel'
> df$distribution_channel[df$distribution_channel == "Undefined"] <- NA
> # Reemplazar "Undefined" por NA en el market segment
> df$market_segment[df$market_segment == "Undefined"] <- NA
> summary(df$distribution_channel)
Corporate    Direct       GDS     TA/TO Undefined      NA's
     5081     12988       181     69141         0         5
> summary(df$market_segment)
     Aviation Complementary     Corporate        Direct        Groups Offline TA/TO
          227           702          4212         11804          4942         13889
    Online TA     Undefined          NA's
        51618             0             2
>
```

Por último, hay valores "Undefined" y "NULL" en "country" por lo que lo pasaremos a "NA".

```
> #Reemplazar los NULL o "Undefined" por NA en los paises
> df$country[df$country == "Undefined"] <- NA
> df$country[df$country == "NULL"] <- NA
> summary(df$country)
    PRT     GBR     FRA     ESP     DEU     ITA     IRL     BEL     BRA     NLD     USA
  27453   10433    8837    7252    5387    3066    3016    2081    1995    1911    1875
    CHE      CN     AUT     SWE     CHN     POL     RUS     NOR     ROU     FIN     ISR
   1570    1093     947     837     816     765     561     515     458     422     403
    DNK     AUS     AGO     LUX     MAR     TUR     ARG     HUN     JPN     IND     CZE
    384     378     342     262     232     213     203     202     183     143     136
    KOR     GRC     HRV     DZA     IRN     EST     ZAF     MEX     LTU     COL     BGR
    119     117      91      82      80      79      78      74      73      69      68
    CHL     NZL     UKR     MOZ     SRB     LVA     ARE     SVK     CYP     SAU     SVN
     65      63      62      56      54      51      48      48      45      45      44
    TWN     THA     TUN     SGP     PHL     EGY     NGA     URY     LBN     ISL     PER
     43      40      37      34      32      31      30      30      29      26      26
    ECU     IDN     BLR     MYS     CPV     HKG     VEN     GEO     JOR     KAZ     CRI
     25      25      24      24      23      23      21      19      19      19      18
    OMN     MLT     AZE     KWT     GIB     QAT     PAK     BIH     DOM     MDV     ALB
     18      17      16      16      15      14      13      12      12      12      11
    IRQ     PRI     SEN     BGD     CMR     MAC     GNB     MKD     ARM     CUB (Other)
     11      11      11      10      10      10       9       9       8       8     211
   NA's
    452
```

Ahora veremos otra vez la estadística para ver si hay algún dato incoherente que debemos arreglar.

```
                NA's
      adr
 Min.   :  -6.38
 1st Qu.:  72.00
 Median :  98.10
 Mean   : 106.34
 3rd Qu.: 134.00
 Max.   :5400.00
```

Notamos que el mínimo valor en "adr" es negativo y eso es incoherente por lo que le pondremos "NULL" para luego darle otro valor, los datos atípicos vamos a modificarlo luego:

```
df$adr[df$adr <0] <- NA
```

Vamos a verificar de nuevo

`summary(df$adr)`

```
>    #le pondremos NULL para luego darle otro valor, los datos atipicos vamos a modificarlo
>    #luego
> df$adr[df$adr <0] <- NA
> #Vamos a verificar de nuevo
> summary(df$adr)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
    0.0    72.0    98.1   106.3   134.0  5400.0       1
```

Vamos a verificar cuantos valores “NA” hay para comprobar valores faltantes en todo el conjunto de datos

`colSums(is.na(df))`

```
> #Vamos a verificar cuantos valores NA hay
> # Comprobar valores faltantes en todo el conjunto de datos
> colSums(is.na(df))
                        hotel                   is_canceled                     lead_time
                            0                             0                             0
      arrival_date_week_number                        adults                      children
                            0                           385                             4
                       babies                          meal                       country
                            0                           492                           452
                market_segment            distribution_channel              is_repeated_guest
                            2                             5                             0
        previous_cancellations previous_bookings_not_canceled             reserved_room_type
                            0                             0                             0
             assigned_room_type                booking_changes                  deposit_type
                            0                             0                             0
                        agent                         company            days_in_waiting_list
                        12193                         82137                             0
                customer_type                           adr    required_car_parking_spaces
                            0                             1                             0
     total_of_special_requests              reservation_status         reservation_status_date
                            0                             0                             0
                  arrival_date                   total_nights
                            0                             0
```

Para el porcentaje de valor “NA” con todos los registros de la columna

`colSums(is.na(df)) / nrow(df) * 100`

```
> #Porcentaje de valor NA con todos los registros de la columna
> colSums(is.na(df)) / nrow(df) * 100
                        hotel                   is_canceled                     lead_time
                  0.000000000                  0.000000000                  0.000000000
      arrival_date_week_number                        adults                      children
                  0.000000000                  0.440523594                  0.004576869
                       babies                         meal                       country
                  0.000000000                  0.562954826                  0.517186141
               market_segment           distribution_channel               is_repeated_guest
                  0.002288434                  0.005721086                  0.000000000
        previous_cancellations previous_bookings_not_canceled             reserved_room_type
                  0.000000000                  0.000000000                  0.000000000
            assigned_room_type               booking_changes                 deposit_type
                  0.000000000                  0.000000000                  0.000000000
                         agent                      company            days_in_waiting_list
                 13.951439425                 93.982562131                  0.000000000
                 customer_type                          adr    required_car_parking_spaces
                  0.000000000                  0.001144217                  0.000000000
     total_of_special_requests              reservation_status       reservation_status_date
                  0.000000000                  0.000000000                  0.000000000
                  arrival_date                 total_nights
                  0.000000000                  0.000000000
```

Tratamiento de Datos Faltantes:

Para los datos faltantes de "Children" vamos a utilizar imputación para no perder registros, la lógica es que según el promedio de adultos que vienen con niños pondremos esos valores en los valores "NA" con X adultos, un ejemplo es si el promedio de 2 adultos es 2 niños. Además, que vemos que los 4 datos faltantes tienen 2 o 3 adultos.

```
# 1. Calcular el promedio de niños cuando hay exactamente 2 o
3 adultos (ignorando NA)
# 2. Reemplazar valores NA en 'children' cuando hay 2 o 3
adultos con el promedio calculado
mean_children_2_adults <- mean(df$children[df$adults == 2],
na.rm = TRUE)
df$children[is.na(df$children) & df$adults == 2] <-
mean_children_2_adults
mean_children_3_adults <- mean(df$children[df$adults == 3],
na.rm = TRUE)
df$children[is.na(df$children) & df$adults == 3] <-
mean_children_3_adults

# Verificar los cambios
```

```
summary(df$children)
```

```
> # 1. Calcular el promedio de niños cuando hay exactamente 2 o 3 adultos (ignorando NA)
> # 2. Reemplazar valores NA en 'children' cuando hay 2 o 3 adultos con el promedio calculado
> mean_children_2_adults <- mean(df$children[df$adults == 2], na.rm = TRUE)
> df$children[is.na(df$children) & df$adults == 2] <- mean_children_2_adults
> mean_children_3_adults <- mean(df$children[df$adults == 3], na.rm = TRUE)
> df$children[is.na(df$children) & df$adults == 3] <- mean_children_3_adults
> # Verificar los cambios
> summary(df$children)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.0000  0.0000  0.0000  0.1386  0.0000 10.0000
```

Porcentaje de valor “NA” con todos los registros de la columna

```
colSums(is.na(df)) / nrow(df) * 100
```

```
> colSums(is.na(df)) / nrow(df) * 100
                         hotel                    is_canceled                      lead_time
                   0.000000000                   0.000000000                   0.000000000
      arrival_date_week_number                        adults                      children
                   0.000000000                   0.440523594                   0.000000000
                        babies                          meal                       country
                   0.000000000                   0.562954826                   0.517186141
                market_segment           distribution_channel             is_repeated_guest
                   0.002288434                   0.005721086                   0.000000000
        previous_cancellations previous_bookings_not_canceled             reserved_room_type
                   0.000000000                   0.000000000                   0.000000000
            assigned_room_type                booking_changes                  deposit_type
                   0.000000000                   0.000000000                   0.000000000
                         agent                       company          days_in_waiting_list
                  13.951439425                  93.982562131                   0.000000000
                customer_type                           adr    required_car_parking_spaces
                   0.000000000                   0.001144217                   0.000000000
     total_of_special_requests             reservation_status       reservation_status_date
                   0.000000000                   0.000000000                   0.000000000
                  arrival_date                  total_nights
                   0.000000000                   0.000000000
>
```

En la estadística podemos ver que “meal“, “adults”, “country”, “market_segment”, “adr” y “distribution_channel“ tienen un porcentaje menor a 1% por lo que vamos a utilizar imputación para rellenar los datos faltantes. Además, en las columnas “company” y “agent” vemos que faltan datos en un 94% y 14% en unos registros de 87396 filas por lo que lo mejor es eliminar la columna ya que tampoco será necesario esos datos.
Primero, para reemplaza los “NA” de “distribution_channel” y “market_segment” vamos a ver si comparten similitudes por lo que utilizaremos los porcentajes para ver la distribución de los valores.

```
# Ver distribución de valores
table(df$meal)
prop.table(table(df$meal)) * 100

#Porcentajes
```

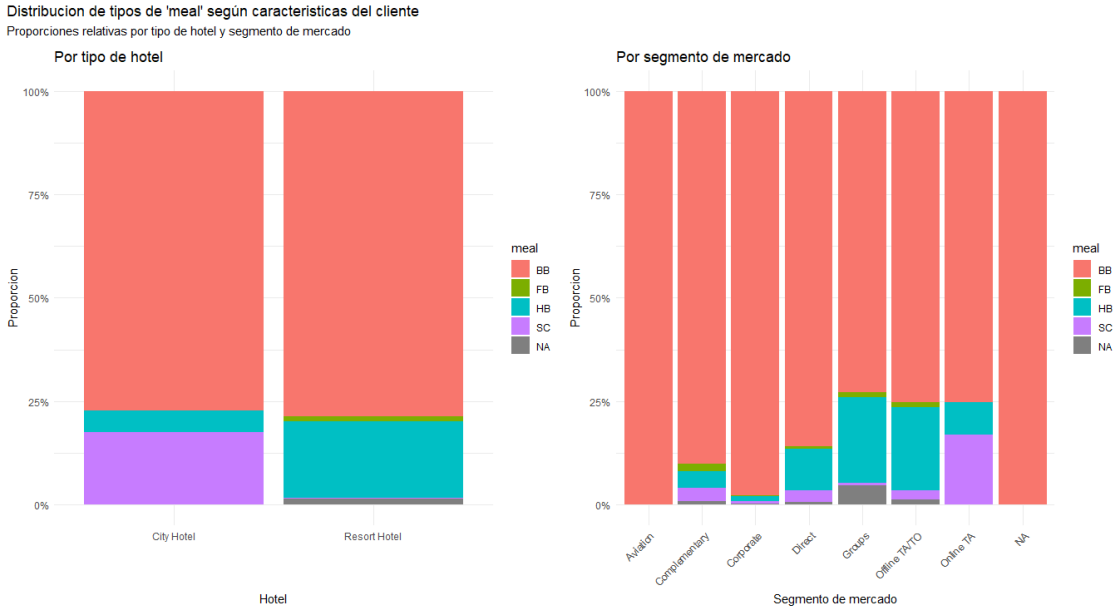
```
table(df$distribution_channel)
prop.table(table(df$distribution_channel)) * 100
> # Ver distribución de valores
> table(df$meal)

       BB         FB         HB         SC Undefined
    67978        360       9085       9481         0
> prop.table(table(df$meal)) * 100  # Porcentajes

        BB          FB          HB          SC  Undefined
78.2219461  0.4142502 10.4540643 10.9097395  0.0000000
> table(df$distribution_channel)

Corporate    Direct       GDS     TA/TO Undefined
     5081     12988       181     69141         0
> prop.table(table(df$distribution_channel)) * 100

 Corporate     Direct        GDS      TA/TO  Undefined
 5.8140999 14.8619423  0.2071151 79.1168427  0.0000000
> |
```

Podemos notar que en “meal” domina con un 78% el BB, además el valor “NA” tiene un porcentaje de 0.56%.

Para visualizarlo mejor vamos a utilizar graficas.

```
# Gráfico 1: Proporción de meals por hotel

p1 <- ggplot(df, aes(x = hotel, fill = meal)) +
  geom_bar(position = "fill") +
  labs(title = "Por tipo de hotel",
       y = "Proporción",
       x = "Hotel") +
  scale_y_continuous(labels = scales::percent) +
  theme_minimal()


# Gráfico 2: Proporción de meals por segmento de mercado

p2 <- ggplot(df, aes(x = market_segment, fill = meal)) +
  geom_bar(position = "fill") +
  labs(title = "Por segmento de mercado",
       y = "Proporción",
       x = "Segmento de mercado") +
  scale_y_continuous(labels = scales::percent) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))


# Combinar ambos con título general

p1 + p2 +
  plot_annotation(
    title = "Distribución de tipos de 'meal' según
características del cliente",
```

```
    subtitle = "Proporciones relativas por tipo de hotel y
segmento de mercado"
  )
```



Y viendo los gráficos sabemos que por moda debemos rellenar los datos vacíos con "BB".

Asimismo, para confirmar con esto:

```
# Encontramos la moda de 'meal' (valor más frecuente)
moda_meal <- names(sort(table(df$meal), decreasing =
TRUE))[1]
print(moda_meal)
# Esto te va a dar el valor que más aparece
```

```
> # Encontramos la moda de 'meal' (valor más frecuente)
> moda_meal <- names(sort(table(df$meal), decreasing = TRUE))[1]
> print(moda_meal)  # Esto te va a dar el valor que más aparece
[1] "BB"
```

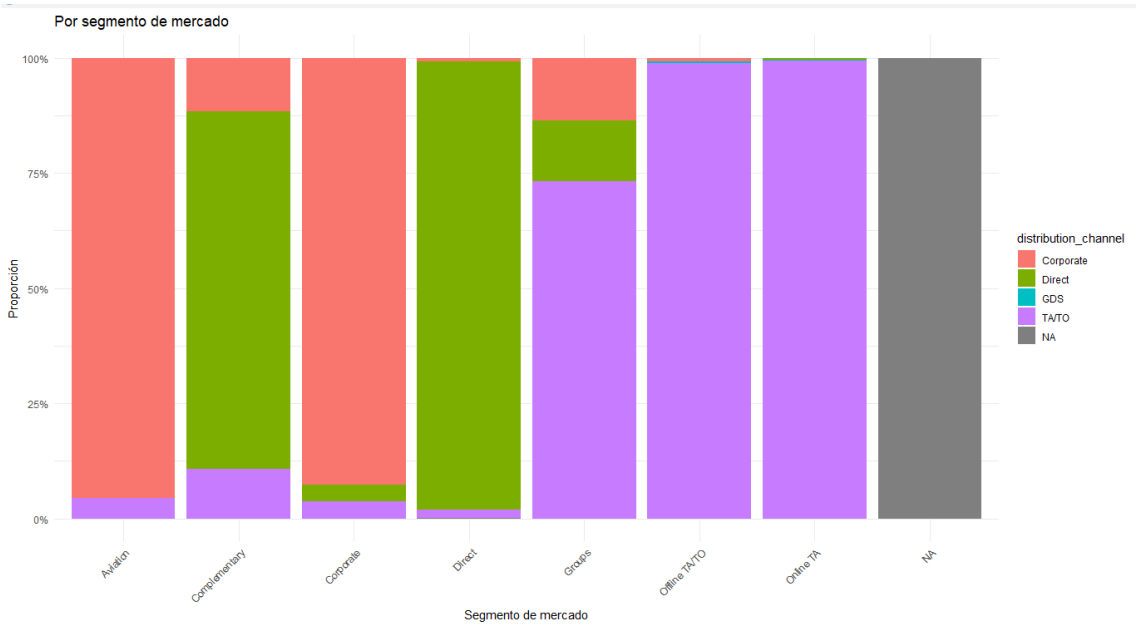
```
# Reemplazar NAs en 'meal' con la moda (BB)
df$meal[is.na(df$meal)] <- moda_meal
summary(df$meal)
```

```
> # Reemplazar NAs en 'meal' con la moda (BB)
> df$meal[is.na(df$meal)] <- moda_meal
> summary(df$meal)
      BB        FB        HB        SC Undefined
   68470       360      9085      9481         0
```

Notamos que ya no hay valores “NA”.

Ahora para los datos “NA” de “distribution_channel” vamos a hacer una imputación y viendo las columnas de la data podemos notar que está casi relacionado con “market_segment” por lo que haremos casi lo mismo.

```
#p4: Proporción de distribution_channel por segmento de
mercado
p4 <- ggplot(df, aes(x = market_segment, fill =
distribution_channel)) +
  geom_bar(position = "fill") +
  labs(title = "Por segmento de mercado",
       y = "Proporción",
       x = "Segmento de mercado") +
  scale_y_continuous(labels = scales::percent) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Podemos notar que en “Aviation” y “Corporate” la mayoría es “Corporate”, en “complementary” y “Direct” la mayoría es “Direct”, en “Groups” y “Online TA/TO”, “Online TA” la mayoría es “TA/TO”. Por lo que vamos a reemplazar en “distribution_channel” con la mayor moda de cada categoría.

```
# Paso 1: Convertimos a character para trabajar sin errores
```

```
df$distribution_channel <-
as.character(df$distribution_channel)

# Paso 3: Calcular la moda de 'distribution_channel' cuando
tanto 'distribution_channel' como 'market_segment' son NA
# Calculamos la moda general para distribution_channel
mode_dist_channel <-
names(sort(table(df$distribution_channel), decreasing =
TRUE))[1]

# Paso 4: Rellenar NA en distribution_channel según el
market_segment (o con la moda si ambos son NA)
df$distribution_channel <-
ifelse(is.na(df$distribution_channel) &
is.na(df$market_segment), mode_dist_channel,

ifelse(is.na(df$distribution_channel) & df$market_segment
%in% c("Corporate", "Aviation"), "Corporate",

ifelse(is.na(df$distribution_channel) & df$market_segment
%in% c("Direct", "Complementary"), "Direct",

ifelse(is.na(df$distribution_channel) & df$market_segment
%in% c("Groups", "Online TA"), "TA/TO",

df$distribution_channel))))

# Paso 5: Convertimos de vuelta a factor
df$distribution_channel <- as.factor(df$distribution_channel)
summary(df$ distribution_channel)
```

```
> summary(df$distribution_channel)
Corporate    Direct       GDS     TA/TO
     5081     12990       181     69144
```

Ahora utilizamos: “`summary(df$market_segment)`” para poder visualizar si hay datos “NA”.

```
> summary(df$market_segment)
     Aviation Complementary     Corporate        Direct        Groups Offline TA/TO
          227           702          4212         11804          4942         13889
    Online TA     Undefined          NA's
        51618             0             2
```

Y notamos que sí hay datos “NA” por lo que vamos a aplicar casi la misma lógica que el anterior.

```
# Paso 1: Convertimos a character para trabajar sin errores
df$market_segment <- as.character(df$market_segment)

# Paso 3: Calcular la moda de 'market_segment' según la
combinación de valores de 'distribution_channel'
# Calculamos la moda general para market_segment
mode_market_segment_corporate <- "Corporate"
#Como sabemos que "Corporate" predomina en "Corporate" y
"Aviation"
mode_market_segment_direct <- "Direct"        # "Direct"
predomina en "Direct" y "Complementary"
mode_market_segment_ta_to <- "TA/TO"          # "TA/TO"
predomina en "Groups" y "Online TA"

# Paso 4: Rellenar los NA en market_segment según
distribution_channel
df$market_segment <- ifelse(is.na(df$market_segment) &
df$distribution_channel == "Corporate",
mode_market_segment_corporate,
                            ifelse(is.na(df$market_segment) &
df$distribution_channel == "Direct",
mode_market_segment_direct,

ifelse(is.na(df$market_segment) & df$distribution_channel ==
"TA/TO", mode_market_segment_ta_to,

df$market_segment)))

# Paso 5: Convertimos de vuelta a factor
df$market_segment <- as.factor(df$market_segment)
summary(df$market_segment)
```

```
summary(df$market_segment)
   Aviation Complementary     Corporate        Direct        Groups Offline TA/TO
        227           702          4212         11804          4942         13889
  Online TA         TA/TO
      51618             2
```

Ahora vamos a ver una estadistica para ver lo que nos falta

```
summary(df)
```

```
         hotel       is_canceled   lead_time      arrival_date_week_number     adults
City Hotel  :53428   0:63371     Min.   :  0.00   Min.   : 1.00            Min.   : 1.000
Resort Hotel:33968   1:24025     1st Qu.: 11.00   1st Qu.:16.00            1st Qu.: 2.000
                                 Median : 49.00   Median :27.00            Median : 2.000
                                 Mean   : 79.89   Mean   :26.84            Mean   : 1.884
                                 3rd Qu.:125.00   3rd Qu.:37.00            3rd Qu.: 2.000
                                 Max.   :737.00   Max.   :53.00            Max.   :55.000
                                                                           NA's   :385
   children           babies                meal          country           market_segment
Min.   : 0.0000   Min.   : 0.00000   BB       :68470   PRT    :27453   Online TA     :51618
1st Qu.: 0.0000   1st Qu.: 0.00000   FB       :  360   GBR    :10433   Offline TA/TO:13889
Median : 0.0000   Median : 0.00000   HB       : 9085   FRA    : 8837   Direct        :11804
Mean   : 0.1386   Mean   : 0.01082   SC       : 9481   ESP    : 7252   Groups        : 4942
3rd Qu.: 0.0000   3rd Qu.: 0.00000   Undefined:    0   DEU    : 5387   Corporate     : 4212
Max.   :10.0000   Max.   :10.00000                      (Other):27582   Complementary:  702
                                                        NA's   :  452   (Other)       :  229
distribution_channel is_repeated_guest previous_cancellations previous_bookings_not_cancele
Corporate: 5081      0:83981           Min.   : 0.00000       Min.   : 0.000
Direct   :12990      1: 3415           1st Qu.: 0.00000       1st Qu.: 0.000
GDS      :  181                        Median : 0.00000       Median : 0.000
TA/TO    :69144                        Mean   : 0.03041       Mean   : 0.184
                                       3rd Qu.: 0.00000       3rd Qu.: 0.000
                                       Max.   :26.00000       Max.   :72.000

reserved_room_type assigned_room_type booking_changes       deposit_type       agent
A      :56552      A      :46313      Min.   : 0.0000   No Deposit:86251   9      :28759
D      :17398      D      :22432      1st Qu.: 0.0000   Non Refund: 1038   240    :13028
E      : 6049      E      : 7195      Median : 0.0000   Refundable:  107   14     : 3349
F      : 2823      F      : 3627      Mean   : 0.2716                      7      : 3300
G      : 2052      G      : 2498      3rd Qu.: 0.0000                      250    : 2779
B      :  999      C      : 2165      Max.   :21.0000                      (Other):23988
(Other): 1523      (Other): 3166                                           NA's   :12193

   company      days_in_waiting_list         customer_type        adr
40     :  851   Min.   :  0.0000     Contract       : 3139   Min.   :   0.0
223    :  503   1st Qu.:  0.0000     Group          :  544   1st Qu.:  72.0
45     :  238   Median :  0.0000     Transient      :71986   Median :  98.1
153    :  206   Mean   :  0.7496     Transient-Party:11727   Mean   : 106.3
154    :  133   3rd Qu.:  0.0000                             3rd Qu.: 134.0
(Other): 3328   Max.   :391.0000                             Max.   :5400.0
NA's   :82137                                                NA's   :1
required_car_parking_spaces total_of_special_requests reservation_status
Min.   :0.00000             Min.   :0.0000            Canceled :23011
1st Qu.:0.00000             1st Qu.:0.0000            Check-Out:63371
Median :0.00000             Median :0.0000            No-Show  : 1014
Mean   :0.08423             Mean   :0.6986
3rd Qu.:0.00000             3rd Qu.:1.0000
Max.   :8.00000             Max.   :5.0000

reservation_status_date  arrival_date         total_nights
Min.   :2014-10-17       Min.   :2015-07-01   Min.   : 0.000
1st Qu.:2016-03-18       1st Qu.:2016-04-01   1st Qu.: 2.000
Median :2016-09-08       Median :2016-09-20   Median : 3.000
Mean   :2016-08-31       Mean   :2016-09-15   Mean   : 3.631
3rd Qu.:2017-03-05       3rd Qu.:2017-04-01   3rd Qu.: 5.000
Max.   :2017-09-14       Max.   :2017-08-31   Max.   :69.000
```

Para lo de country no podemos encontrar un patrón por ahora por lo que como es un factor el valor "NA" será reemplazado por la mayor cantidad de ciudadanos que van al hotel según el data set, voy a hacer una estadística de "country".

```
summary(df$country)
```

```
#El mayor valor es PRT, por lo que reemplazaremos con ese
df$country[is.na(df$country)] <- "PRT"
```

```
> #Para lo de country no podemos encontrar un patron por ahora por lo que
>    #como es un factor el valor NA será reemplazado por la mayor cantidad de ciudadanos
>     #que van al hotel segun el data set, voy a hacer una estadistica de country
> summary(df$country)  #El mayor valor es PRT, por lo que reemplazaremos con ese
   PRT     GBR     FRA     ESP     DEU     ITA     IRL     BEL     BRA     NLD     USA
 27453   10433    8837    7252    5387    3066    3016    2081    1995    1911    1875
   CHE      CN     AUT     SWE     CHN     POL     RUS     NOR     ROU     FIN     ISR
  1570    1093     947     837     816     765     561     515     458     422     403
   DNK     AUS     AGO     LUX     MAR     TUR     ARG     HUN     JPN     IND     CZE
   384     378     342     262     232     213     203     202     183     143     136
   KOR     GRC     HRV     DZA     IRN     EST     ZAF     MEX     LTU     COL     BGR
   119     117      91      82      80      79      78      74      73      69      68
   CHL     NZL     UKR     MOZ     SRB     LVA     ARE     SVK     CYP     SAU     SVN
    65      63      62      56      54      51      48      48      45      45      44
   TWN     THA     TUN     SGP     PHL     EGY     NGA     URY     LBN     ISL     PER
    43      40      37      34      32      31      30      30      29      26      26
   ECU     IDN     BLR     MYS     CPV     HKG     VEN     GEO     JOR     KAZ     CRI
    25      25      24      24      23      23      21      19      19      19      18
   OMN     MLT     AZE     KWT     GIB     QAT     PAK     BIH     DOM     MDV     ALB
    18      17      16      16      15      14      13      12      12      12      11
   IRQ     PRI     SEN     BGD     CMR     MAC     GNB     MKD     ARM     CUB (Other)
    11      11      11      10      10      10       9       9       8       8     211
  NA's
   452
> df$country[is.na(df$country)] <- "PRT"
```

Ahora vamos a reemplazar los valores “NA” de “aduts”.

```
#Calcular la media redondeada de adults, excluyendo NAs
mean_adults <- round(mean(df$adults, na.rm = TRUE))

#Imputar valores NA en adults con la media
df$adults[is.na(df$adults)] <- mean_adults

#Aplicamos la condición: si no hay niños ni bebés y la
reserva fue cancelada, poner adults en 0
df <- df %>%
  mutate(adults = ifelse(children == 0 & babies == 0 &
is_canceled == 1, 0, adults))
```

```
> mean_adults <- round(mean(df$adults, na.rm = TRUE))
> # Imputar valores NA en adults con la media
> df$adults[is.na(df$adults)] <- mean_adults
> # Aplicamos la condición: si no hay niños ni bebés y la reserva fue cancelada, poner adults en
0
> df <- df %>%
+   mutate(adults = ifelse(children == 0 & babies == 0 & is_canceled == 1, 0, adults))
> # Imputar valores NA en adults con la media
> df$adults[is.na(df$adults)] <- mean_adults
> # Aplicamos la condición: si no hay niños ni bebés y la reserva fue cancelada, poner adults en
0
> df <- df %>%
+   mutate(adults = ifelse(children == 0 & babies == 0 & is_canceled == 1, 0, adults))
> summary(df$adults)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   1.000   2.000   1.417   2.000   4.000
```

Después, le daremos un valor medio al “adr” “NA” para que no afecte a la dataset

```
df$adr[is.na(df$adr)] <- round(mean(df$adr, na.rm = TRUE))
```

```
> #Por ultimo le darenos un valor medio al adr NA para que no afecte a la dataset
> df$adr[is.na(df$adr)] <- round(mean(df$adr, na.rm = TRUE))
> summary(df$adr)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    0.0    72.0    98.1   106.3   134.0  5400.0
```

Vamos a suponer que las fechas son correctas porque es dificil averiguar si alguno es correcto igual que el lead time y arreglamos los errores de reservas como incoherencies.

```
df <- df %>%
  mutate(
    expected_nights = as.numeric(reservation_status_date -
arrival_date),
    status_check = case_when(
      reservation_status == "Check-Out" ~ expected_nights ==
total_nights,
      reservation_status == "No-Show" ~
reservation_status_date == arrival_date,
      reservation_status == "Canceled" ~ expected_nights <=
0,
      TRUE ~ NA
    )
  ) %>%

  #Arreglar casos inconsistentes
  mutate(
    reservation_status = case_when(
      # Si es No-Show, pero la fecha no coincide y arrival es
mayor, y sí se canceló → cambiar a "Canceled"
      reservation_status == "No-Show" &
        status_check == FALSE &
        arrival_date > reservation_status_date &
        is_canceled == 1 ~ "Canceled",

      TRUE ~ reservation_status
    ),

    #Ajustar total_nights si es Check-Out inconsistente y no
fue cancelado
```

```
    total_nights = case_when(
      reservation_status == "Check-Out" &
        status_check == FALSE &
        is_canceled == 0 ~ expected_nights,

      TRUE ~ total_nights
    )
  )
```

Eliminamos las columnas que habiamos creado

```
df <- subset(df, select = -c(expected_nights))
df <- subset(df, select = -c(status_check))
```

```
> #Vamos a suponer que las fechas son correctas porque es dificil averiguar si alguno es correcto
igual que el lead time
> #Arreglamos los errores de reservas como incoherencias
> df <- df %>%
+   mutate(
+     expected_nights = as.numeric(reservation_status_date - arrival_date),
+     status_check = case_when(
+       reservation_status == "Check-Out" ~ expected_nights == total_nights,
+       reservation_status == "No-Show" ~ reservation_status_date == arrival_date,
+       reservation_status == "Canceled" ~ expected_nights <= 0,
+       TRUE ~ NA
+     )
+   ) %>%
+   # Arreglar casos inconsistentes
+   mutate(
+     reservation_status = case_when(
+       # Si es No-Show, pero la fecha no coincide y arrival es mayor, y sí se canceló → cambiar
a "Canceled"
+       reservation_status == "No-Show" &
+         status_check == FALSE &
+         arrival_date > reservation_status_date &
+         is_canceled == 1 ~ "Canceled",
+
+       TRUE ~ reservation_status
+     ),
+     # Ajustar total_nights si es Check-Out inconsistente y no fue cancelado
+     total_nights = case_when(
+       reservation_status == "Check-Out" &
+         status_check == FALSE &
+         is_canceled == 0 ~ expected_nights,
+
+       TRUE ~ total_nights
+     )
+   )
```

Vemos el porcentaje de valores faltantes por columna

```
colSums(is.na(df)) / nrow(df) * 100
```

```
> # Ver el porcentaje de valores faltantes por columna
> colSums(is.na(df)) / nrow(df) * 100
                         hotel                    is_canceled                      lead_time
                       0.00000                        0.00000                        0.00000
       arrival_date_week_number                         adults                       children
                       0.00000                        0.00000                        0.00000
                         babies                           meal                        country
                       0.00000                        0.00000                        0.00000
                 market_segment           distribution_channel              is_repeated_guest
                       0.00000                        0.00000                        0.00000
        previous_cancellations previous_bookings_not_canceled             reserved_room_type
                       0.00000                        0.00000                        0.00000
            assigned_room_type                booking_changes                   deposit_type
                       0.00000                        0.00000                        0.00000
                         agent                        company               days_in_waiting_list
                      13.95144                       93.98256                        0.00000
                 customer_type                            adr     required_car_parking_spaces
                       0.00000                        0.00000                        0.00000
      total_of_special_requests             reservation_status          reservation_status_date
                       0.00000                        0.00000                        0.00000
                  arrival_date                    total_nights                 expected_nights
                       0.00000                        0.00000                        0.00000
                  status_check
                       0.00000
```

Y notamos que ya no hay datos “NA” (excepto por “company” y “agent”) por lo que ya está completo, por último, eliminamos las columnas que tienen más del 10% con “NA” porque son más de 87 mil registros y un valor del 10% es alto.

```
#Eliminamos la columna agent y company porque faltan muchos datos
df <- subset(df, select = -c(agent))
df <- subset(df, select = -c(company))
```

```
> #Eliminamos la columna agent y company porque faltan muchos datos
> df <- subset(df, select = -c(agent))
> df <- subset(df, select = -c(company))
> # Combinar las columnas de día, mes y año para formar una fecha completa
> df$arrival_date <- paste(df$arrival_date_year, df$arrival_date_month, df$arrival_date_day_of_mo
nth, sep = "-")
> # Convertir el texto a una fecha (tipo Date)
> df$arrival_date <- as.Date(df$arrival_date, format = "%Y-%m-%d")
> View(df)
> # Combinar las columnas de día, mes y año para formar una fecha completa
> df$arrival_date <- paste(df$arrival_date_year, df$arrival_date_month, df$arrival_date_day_of_mo
nth, sep = "-")
```

Ahora podemos decir que los datos ya están limpios por lo que haremos un resumen estadístico para comprobarlo.

```
summary(df)
```

```
previous_cancellations previous_bookings_not_canceled reserved_room_type assigned_room_type
Min.   : 0.00000       Min.   : 0.000                 A      :56552       A      :46313
1st Qu.: 0.00000       1st Qu.: 0.000                 D      :17398       D      :22432
Median : 0.00000       Median : 0.000                 E      : 6049       E      : 7195
Mean   : 0.03041       Mean   : 0.184                 F      : 2823       F      : 3627
3rd Qu.: 0.00000       3rd Qu.: 0.000                 G      : 2052       G      : 2498
Max.   :26.00000       Max.   :72.000                 B      :  999       C      : 2165
                                                      (Other): 1523       (Other): 3166
booking_changes       deposit_type   days_in_waiting_list         customer_type
Min.   : 0.0000   No Deposit:86251   Min.   :  0.0000     Contract       : 3139
1st Qu.: 0.0000   Non Refund: 1038   1st Qu.:  0.0000     Group          :  544
Median : 0.0000   Refundable:  107   Median :  0.0000     Transient      :71986
Mean   : 0.2716                      Mean   :  0.7496     Transient-Party:11727
3rd Qu.: 0.0000                      3rd Qu.:  0.0000
Max.   :21.0000                      Max.   :391.0000

     adr          required_car_parking_spaces total_of_special_requests reservation_status
Min.   :  -6.38   Min.   :0.00000             Min.   :0.0000            Canceled :23011
1st Qu.:  72.00   1st Qu.:0.00000             1st Qu.:0.0000            Check-Out:63371
Median :  98.10   Median :0.00000             Median :0.0000            No-Show  : 1014
Mean   : 106.34   Mean   :0.08423             Mean   :0.6986
3rd Qu.: 134.00   3rd Qu.:0.00000             3rd Qu.:1.0000
Max.   :5400.00   Max.   :8.00000             Max.   :5.0000

reservation_status_date arrival_date
Min.   :2014-10-17      Length:87396
1st Qu.:2016-03-18      Class :character
Median :2016-09-08      Mode  :character
Mean   :2016-08-31
3rd Qu.:2017-03-05
Max.   :2017-09-14
```

Detectar Outliers:

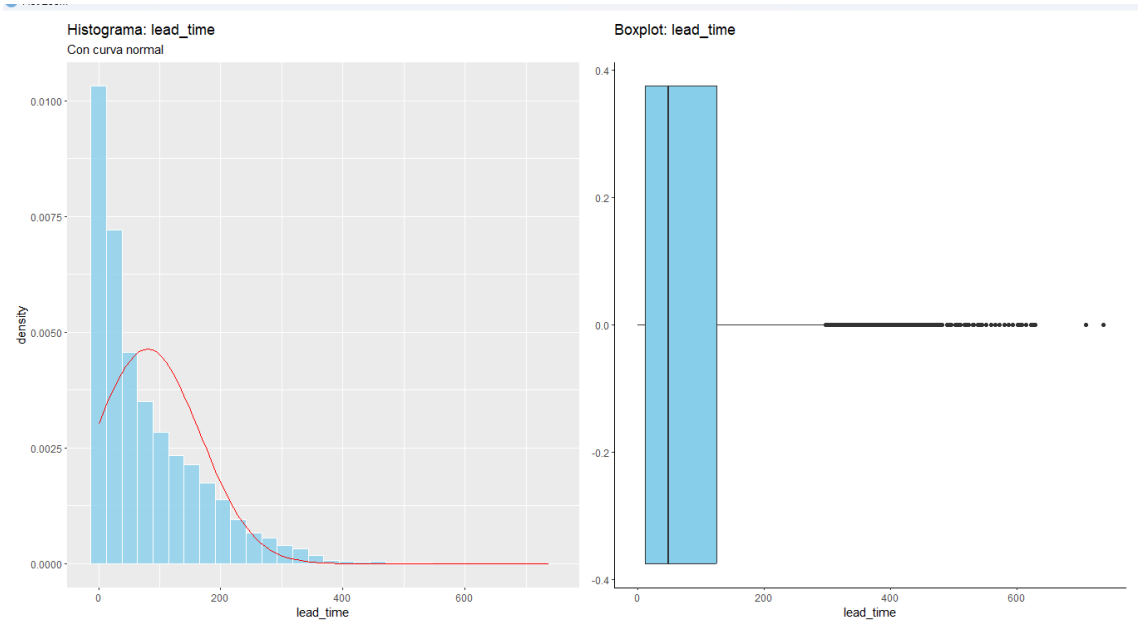
Para poder hallar los valores atípicos utilizaremos en cada variable numérica un gráfico de histograma con curva simple y junto con su "`boxplots`".

Primero comenzamos con "`lead_time`"

```
#Vamos a crear primero para los valores numericos graficas de
histograma y boxplots para
#poder identificar los valores atipicos
summary(df)
#Visualizamos las estadisticas para ver que todo este bien
p5 <- ggplot(df, aes(x = lead_time)) +
  geom_histogram(aes(y = ..density..), bins = 30, fill =
"skyblue", color = "white", alpha = 0.8) +
  stat_function(fun = dnorm, args = list(mean =
mean(df$lead_time, na.rm = TRUE),
                                         sd =
sd(df$lead_time, na.rm = TRUE)), color = "red") +
  labs(title = "Histograma: lead_time", subtitle = "Con curva
normal")

b5 <- ggplot(df, aes(x = lead_time)) +
```

```
  geom_boxplot(fill = "skyblue") +
  labs(title = "Boxplot: lead_time") +
  theme_classic()

d1 <- (p5 | b5)
  plot_annotation(
    title = "Análisis de outliers en del lead time",
    subtitle = "Distribución y valores atípicos de
lead_time",
    caption = "Estado: con sueñoo, quiero dormir"
  )
```

Para el lead time si vamos a reemplazar los valores atípicos porque esos días de reserva son demasiado largos y lo máximo debería ser hasta 180 días o un aproximado.



También utilizamos los “outliers” para visualizar los datos atípicos

```
#Visualización de datos atipicos
#lead_time
outliers<-boxplot(df$lead_time,plot=FALSE)$out
outliers
```

```
> #Visualización de datos atipicos
> #lead_time
> outliers<-boxplot(df$lead_time,plot=FALSE)$out
> outliers
   [1] 342 737 368 364 324 394 366 304 321 317 315 312 299 298 327 460 346 333 381 304 297 297
  [23] 327 333 314 323 340 356 328 336 302 302 344 382 338 310 340 305 354 347 349 352 354 361
  [45] 338 328 328 328 330 350 368 334 334 334 312 709 354 468 468 468 304 307 311 312 314 319
  [67] 319 322 322 323 328 330 300 304 301 306 304 309 322 322 322 322 322 322 310 314 301 301
  [89] 304 305 314 349 317 317 308 398 317 324 323 309 327 303 356 424 309 298 297 331 311 333
 [111] 305 305 305 306 339 301 322 324 307 434 349 349 357 325 329 346 329 308 317 314 332 338
 [133] 304 343 345 360 348 348 360 328 305 367 367 306 353 299 373 374 333 328 333 333 314 406
 [155] 406 406 406 400 326 340 315 327 346 346 346 345 303 379 303 399 332 316 315 315 346 315
 [177] 322 315 361 309 341 304 330 347 373 361 347 320 304 364 338 301 301 354 305 348 314 385
 [199] 355 363 360 358 311 333 311 338 305 311 333 332 422 390 390 334 356 327 327 327 327 327
 [221] 335 335 335 335 335 335 327 336 336 336 336 336 336 336 336 336 336 336 336 336 336 336
 [243] 336 336 336 336 336 336 336 336 370 363 350 338 394 347 376 376 342 375 297 328 328 343
 [265] 343 385 397 397 397 397 343 343 329 397 397 343 397 338 344 344 310 542 542 347 542 542
 [287] 339 333 354 542 542 542 542 333 542 342 342 319 403 403 341 383 383 383 383 383 383 383
 [309] 383 383 383 383 383 383 383 383 383 383 383 383 383 384 359 385 393 315 360 360 360 360
 [331] 368 337 337 337 337 337 337 337 337 337 316 362 363 339 339 363 339 364 364 364 364 364
 [353] 364 364 364 364 364 364 364 364 364 364 364 364 364 364 364 364 364 364 364 364 364 364
 [375] 364 364 364 364 364 364 365 365 365 365 342 342 342 342 342 342 342 342 435 342 375 382
 [397] 386 386 385 385 385 385 385 385 386 386 386 319 314 378 378 378 350 308 309 335 317 317
 [419] 315 319 319 322 307 298 348 342 320 316 312 312 312 312 312 321 339 332 339 309 320 340
 [441] 350 364 313 351 302 308 330 319 302 302 303 337 338 313 305 305 302 322 336 310 333 299
 [463] 300 471 317 462 312 305 351 352 352 305 305 297 304 322 350 355 305 311 312 337 313 313
 [485] 353 313 313 313 353 352 338 321 342 312 314 322 411 450 350 309 344 321 344 320 320 310
 [507] 323 319 320 310 319 319 320 320 320 314 355 326 318 319 319 321 303 329 340 328 304 304
 [529] 304 309 304 309 304 307 330 332 314 306 306 319 298 345 343 308 310 368 331 307 321 326
 [551] 312 340 298 330 330 313 321 326 320 329 324 336 332 336 326 326 331 317 317 325 305 311
 [573] 342 311 330 342 311 311 336 312 313 309 319 319 336 309 310 310 339 333 319 317 353 303
 [595] 305 305 305 334 334 303 325 306 320 300 317 347 347 320 324 315 324 335 345 334 352 338
 [617] 326 339 335 313 349 315 346 354 333 340 323 322 390 332 299 299 327 328 347 372 312 352
 [639] 347 335 381 340 300 318 335 358 306 358 342 343 359 309 359 357 329 343 298 298 334 334
 [661] 328 358 352 378 300 310 304 304 304 304 353 304 312 312 319 302 341 333 333 328 328 343
 [683] 314 310 311 316 298 328 314 339 338 311 311 306 306 311 305 301 306 327 307 341 309 300
 [705] 342 358 371 358 358 306 349 353 337 315 360 338 338 353 301 344 327 305 350 305 303 335
 [727] 454 297 327 348 399 321 355 299 321 325 468 468 311 328 310 460 329 318 305 352 334 333
 [749] 301 355 301 305 343 343 309 315 532 305 318 297 304 303 342 468 468 468 468 468 468 468
 [771] 468 383 386 383 366 297 297 297 332 301 304 335 311 319 308 317 351 322 345 319 328 338
 [793] 342 315 333 342 305 357 406 301 329 339 302 336 336 336 336 336 336 336 336 336 334 327
 [815] 327 336 336 336 336 336 336 336 336 336 336 336 348 297 297 422 298 336 326 297 297 445 305
 [837] 542 542 542 445 445 542 542 542 542 542 342 342 342 342 342 342 315 383 383 383 383 384
 [859] 383 383 383 364 341 365 365 363 364 364 364 364 364 364 364 364 363 364 364 364 364 364
 [881] 364 365 363 364 365 365 364 365 297 363 339 339 338 318 364 364 364 301 301 359 386 386
 [903] 386 386 386 386 386 390 386 389 389 386 386 386 386 386 386 386 386 386 386 386 386 386
 [925] 386 386 386 386 386 323 362 315 336 350 325 309 353 316 307 312 312 312 312 312 355 339
 [947] 339 339 339 339 339 313 337 326 313 338 351 351 351 351 351 351 317 317 317 300 314 302
 [969] 302 302 302 302 313 319 351 326 297 361 361 329 303 302 358 352 334 352 353 388 388 388
 [991] 388 388 388 388 302 307 307 309 388 319
 [ reached 'max' / getOption("max.print") -- omitted 1396 entries ]
```

Intentamos con el metodo IQR

```
#Calcular el IQR para la variable 'lead_time'

Q1 <- quantile(df$lead_time, 0.25)

Q3 <- quantile(df$lead_time, 0.75)

IQR <- Q3 - Q1


#Limites inferior y superior

lower_bound <- Q1 - 1.5 * IQR
```

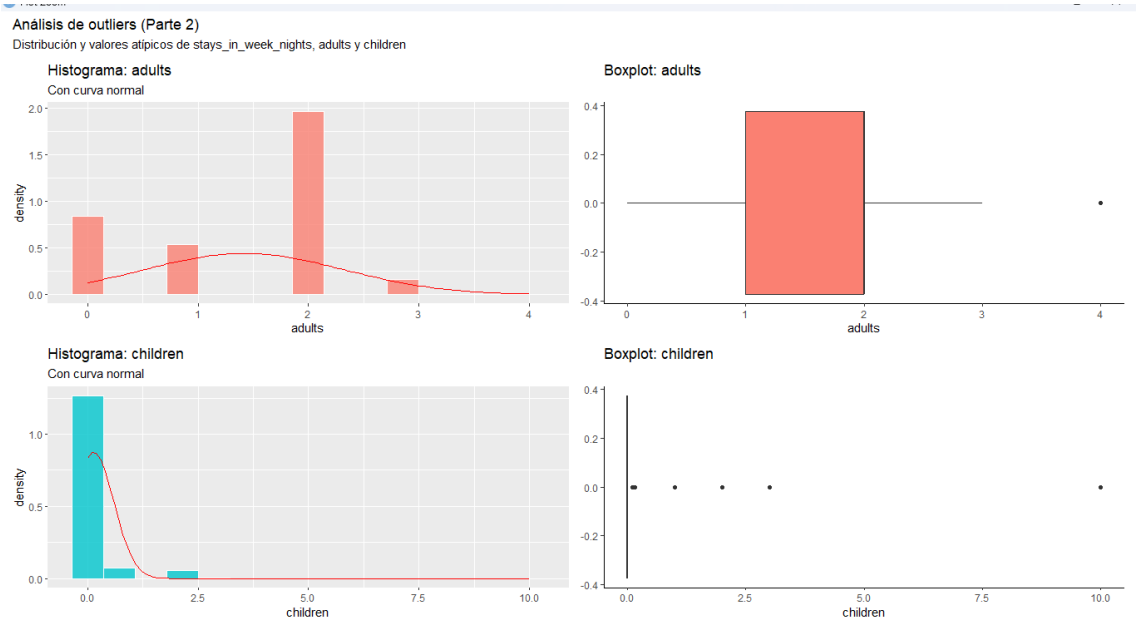
```
upper_bound <- Q3 + 1.5 * IQR


#Identificar datos atípicos

outliers <- df$lead_time[df$lead_time < lower_bound |
df$lead_time > upper_bound

]

outliers
```

Nos saldrá un resultado similar.

```
  #Segunda parte de adults, childre
  p9 <- ggplot(df, aes(x = adults)) +
    geom_histogram(aes(y = ..density..), bins = 15, fill =
  "salmon", color = "white", alpha = 0.8) +
    stat_function(fun = dnorm, args = list(mean =
  mean(df$adults, na.rm = TRUE),
                                           sd = sd(df$adults,
  na.rm = TRUE)), color = "red") +
    labs(title = "Histograma: adults", subtitle = "Con curva
  normal")


  p10 <- ggplot(df, aes(x = children)) +
    geom_histogram(aes(y = ..density..), bins = 15, fill =
  "turquoise3", color = "white", alpha = 0.8) +
    stat_function(fun = dnorm, args = list(mean =
  mean(df$children, na.rm = TRUE),
                                           sd = sd(df$children,
  na.rm = TRUE)), color = "red") +
    labs(title = "Histograma: children", subtitle = "Con curva
  normal")


  b9 <- ggplot(df, aes(x = adults)) +
    geom_boxplot(fill = "salmon") +
    labs(title = "Boxplot: adults") +
    theme_classic()


  b10 <- ggplot(df, aes(x = children)) +
    geom_boxplot(fill = "turquoise3") +
    labs(title = "Boxplot: children") +
    theme_classic()
  d2 <-
    (p9 | b9) /
```

```
  (p10 | b10) +
  plot_annotation(
    title = "Análisis de outliers (Parte 2)",
    subtitle = "Distribución y valores atípicos de
stays_in_week_nights, adults y children",

  )
```

Notamos que hay muchos adultos en `"0"` pero muchos se deben porque cancelaron por lo que adultos lo dejamos como está, pero los niños vemos un valor de casi 10 niños por lo que sí es un valor atípico y lo vamos a reemplazar por lo que no afectará para después analizar porque ese casi 10 sí podría afectarnos si buscamos máximo




```
#Visualización de datos atipicos

#adults

outliers<-boxplot(df$adults,plot=FALSE)$out
outliers
```

```
> outliers
   [1]  1  1  1  1  3  3  3  3  3  3  1  3  3  1  3  3  1  1  4  1  1  1  3  3  1  3  3  1  1
  [30]  1  1  3  3  3  1  3  3  3  1  1  3  3  1  3  4  1  3  1  1  1  3  1  1  3  1  1  1  1
  [59]  1  1  1  1  3  3  3  1  1  1  1  1  1  3  3  3  1  3  3  3  1  3  1  3  3  1  3  3  1
  [88]  1  3  1  1  1  1  3  1  3  1  1  3  3  1  3  3  1  1  1  4  1  1  3  1  1  3  1  1  3
 [117]  3  3  3  3  3  3  3  1  1  3  3  1  3  3  3  3  1  1  3  3  1  3  3  3  3  3  1  1  1
 [146]  1  3  3  1  1  1  3  1  1  3  1  3  1  3  1  1  1  1  3  1 40  3  1  1  3  1 26  1  1
 [175]  3  3  1 50  1  1  1  1  1  1  1  1  3  1  1  1  3  1  1  1 26  1  1  1  1  1  1  1
 [204]  1  3 26  3  1  1 27  1  1  3 27  3  1 26  1  1  1  1  1  1  1  1  1  1  1  3  1  1  1
 [233]  1  1  1  1  1  1  1  1  1  1  1 26  1 55  1  1  0 20  6  5  1  1  1  3  1  1  1  1  1
 [262]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0  1  1 10 20  5  1  1  1  1  1  1  1  1  1
 [291]  1  1  1  1  1  1  3  3  3  1  1  1  1  1  1  1  1  1  1  1  1  1  1  3  1  1  1  1  1
 [320]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  3
 [349]  1  1  1  1  1  1  1  1  1  1  1  1  3  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 [378]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  3  1  1  1  1  1
 [407]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 [436]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  0  1  3  1  1  3  1  1  3  1  1  1  1  1
 [465]  1  3  1  3  1  1  1  1  1  1  3  3  3  1  1  1  1  1  1  1  1  3  3  1  3  1  1  1  1
 [494]  1  1  3  1  1  1  1  1  1  1  3  0  1  1  1  3  0  3  3  3  1  1  1  1  1  1  1  1  1
 [523]  1  1  1  1  1  1  1  1  1  1  1  3  1  1  1  1  1  1  3  3  3  1  1  1  3  0  3  1  1
 [552]  3  3  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 [581]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  3  1  1  1  1  1  3  3  3  3  3  1
 [610]  1  3  3  1  3  3  3  1  1  1  1  3  1  1  1  3  1  1  1  3  1  1  1  3  1  1  1  1  1
 [639]  3  1  1  1  1  3  1  1  1  1  1  1  1  3  3  1  1  1  1  3  1  1  1  1  1  1  1  1  1
 [668]  1  1  1  3  1  1  1  1  1  1  1  1  1  1  1  1  3  1  1  1  1  1  1  1  3  1  3  1  1
 [697]  1  1  1  1  1  3  1  1  1  1  1  3  1  1  1  1  1  3  1  1  1  1  1  1  1  1  3  3  1
 [726]  1  1  1  3  1  1  1  1  1  1  1  1  1  1  1  3  1  1  1  1  1  1  1  1  3  1  1  1  1
 [755]  3  1  1  1  1  1  1  1  3  1  1  4  1  1  1  3  1  1  1  1  1  1  1  1  1  1  1  1  1
 [784]  1  1  1  1  1  1  3  1  1  1  1  1  3  1  1  3  1  1  3  1  3  1  1  1  1  3  3  1  3
 [813]  1  3  3  3  1  1  3  1  1  1  3  1  1  3  3  3  1  1  3  1  1  1  1  1  1  1  1  1  3
 [842]  3  1  3  1  1  1  1  1  3  1  1  3  3  1  3  1  3  3  1  1  1  1  3  1  1  3  1  1  3
 [871]  3  1  3  3  1  1  3  3  1  1  1  3  3  1  1  1  1  3  1  3  3  3  3  1  1  1  1  3  3
 [900]  1  1  1  1  3  3  3  1  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  1  3  3  1  3  3
 [929]  1  1  3  1  3  3  3  3  3  1  3  3  3  1  3  3  3  3  3  1  3  1  3  1  1  1  1  3  3
 [958]  3  3  3  3  1  1  3  1  1  1  1  1  3  1  3  3  1  3  3  1  4  3  1  1  1  1  1  1  3
 [987]  3  4  1  1  3  3  1  1  1  1  1  3  1  1
 [ reached 'max' / getOption("max.print") -- omitted 21899 entries ]
```

```
#children

outliers<-boxplot(df$children,plot=FALSE)$out
outliers
```

```
> outliers
   [1]  1  2  2  2  1  1  2  2  1  2  1  2  2  2  2  2  1  2  1  2  1  2  2  2  2  1  2  1  1
  [30]  1  1  1  2  2  1  1 10  2  2  2  1  1  2  2  2  2  2  2  1  1  1  2  2  1  1  1  1  2
  [59]  2  1  2  2  1  2  1  2  2  2  2  2  1  2  1  1  1  2  1  1  2  2  2  2  2  1  1  2  2
  [88]  1  1  2  2  2  1  1  1  1  2  2  1  2  1  2  2  2  2  2  2  2  1  2  1  2  1  2  2  2
 [117]  1  2  1  2  2  1  2  2  2  2  1  1  2  1  1  2  1  1  1  2  1  1  2  2  1  2  2  1  1
 [146]  1  1  2  2  1  1  2  1  2  2  1  2  1  1  1  1  1  1  1  2  1  2  1  1  2  2  1  1  2
 [175]  2  2  1  1  1  2  2  1  1  2  2  2  1  2  1  1  2  2  2  2  1  1  1  2  2  2  1  2  2
 [204]  2  2  2  1  2  2  2  2  2  2  2  2  1  1  1  1  1  2  2  2  2  2  2  1  1  2  1  1  1
 [233]  1  1  1  1  1  2  2  1  2  2  2  1  2  1  2  2  1  1  2  1  2  2  1  1  1  1  1  1  2
 [262]  1  1  1  1  1  1  1  2  1  1  1  2  1  2  2  1  1  2  2  1  1  2  2  2  2  1  2  2  1
 [291]  2  2  1  2  1  1  1  1  1  1  1  2  2  1  2  1  2  1  1  2  1  2  1  1  2  1  2  1  2
 [320]  1  1  1  1  1  2  2  1  2  2  1  2  2  2  2  2  2  1  2  2  1  1  2  1  1  1  1  1  2
 [349]  2  2  2  2  2  2  2  2  1  1  2  1  2  1  2  1  2  1  2  1  2  1  2  1  2  2  2  2  2
 [378]  2  1  2  1  2  2  2  1  2  2  1  2  2  2  2  1  2  2  2  1  2  2  2  1  2  1  1  2  1
 [407]  1  2  2  2  1  1  2  2  2  1  1  1  2  1  2  2  2  1  2  2  2  1  2  1  2  1  2  1  1
 [436]  2  2  1  1  1  1  2  1  2  2  1  2  2  1  2  2  1  2  1  1  1  2  2  1  2  2  1  2  1
 [465]  2  2  2  1  1  1  2  1  1  3  2  1  1  2  2  1  2  1  1  2  1  1  2  1  2  1  1  2  1
 [494]  2  2  2  2  2  2  2  1  1  2  1  2  1  1  2  2  2  2  2  2  2  2  2  2  2  1  2  1  2  2
 [523]  2  1  2  2  2  2  2  2  1  2  1  1  2  2  2  1  1  1  2  2  2  1  1  2  1  2  1  1  2
 [552]  2  2  2  1  2  2  1  1  1  1  2  1  2  2  1  1  2  2  1  2  2  1  2  2  1  2  1  1  1
 [581]  1  1  1  2  2  1  2  2  1  2  1  2  2  1  1  2  1  2  1  1  2  2  1  1  1  1  2  2  2
 [610]  2  2  2  1  1  2  1  1  2  1  1  1  1  1  1  1  1  1  1  2  1  2  1  1  2  1  3  1  1
 [639]  1  1  1  2  1  1  1  1  2  2  2  2  2  1  2  2  1  2  1  1  1  2  1  2  1  1  1  1  2
 [668]  2  2  2  2  2  2  2  1  1  2  2  2  2  1  2  1  1  2  1  1  2  2  1  1  1  1  1  1  1
 [697]  1  1  1  2  1  1  2  2  2  1  2  2  1  1  2  1  2  2  2  2  2  2  1  2  2  2  2  1  2
 [726]  2  2  1  1  1  2  2  1  2  2  1  2  2  2  2  2  1  2  1  1  2  1  1  2  2  1  1  1  2
 [755]  1  2  1  2  1  2  2  2  1  1  1  1  2  1  1  1  1  2  1  1  2  1  2  1  2  2  1  2  2
 [784]  1  1  2  2  1  2  1  1  1  2  1  1  1  2  2  2  1  2  2  2  2  1  1  1  1  1  1  1  2
 [813]  1  1  1  2  1  1  2  1  2  2  1  2  1  2  1  2  2  1  1  2  1  1  1  2  1  1  1  2  2
 [842]  2  2  2  2  1  1  1  2  2  2  2  1  2  2  2  2  2  2  2  2  2  2  2  1  2  2  2  2  2  2
 [871]  1  2  2  2  1  2  2  1  2  1  1  2  1  2  1  1  2  1  2  2  2  2  1  2  2  2  2  1  2
 [900]  2  1  1  2  1  1  2  2  1  1  2  1  1  2  2  2  2  1  1  1  1  1  2  1  1  1  1  2  1
 [929]  1  1  2  2  2  2  2  1  2  2  1  2  2  2  2  2  2  2  2  2  1  2  2  2  2  2  1  2
 [958]  2  2  1  2  2  2  2  2  2  2  1  2  2  2  2  2  2  1  2  2  2  2  2  2  2  1  2  2  2
 [987]  2  2  2  2  2  2  2  2  2  2  2  2  1  1
 [ reached 'max' / getOption("max.print") -- omitted 7368 entries ]
```

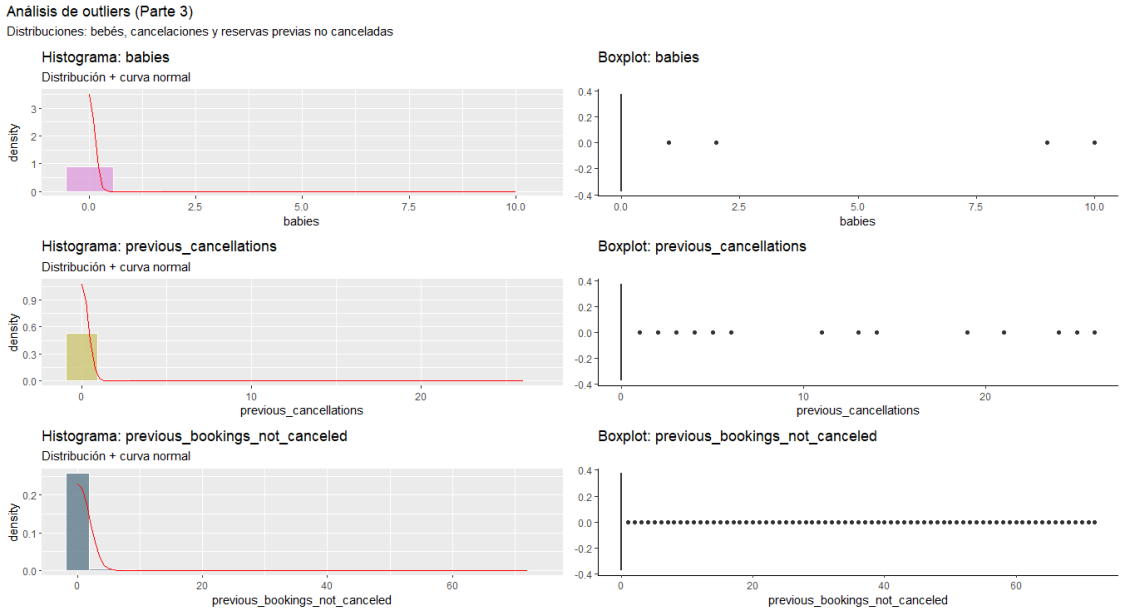
Ahora avanzamos con la parte 3 que son “babies”, “previous_cancellations” y “previous_bookings_not_canceled”.

```
p11 <- ggplot(df, aes(x = babies)) +
  geom_histogram(aes(y = ..density..), bins = 10, fill =
"plum", color = "white", alpha = 0.8) +
  stat_function(fun = dnorm, args = list(mean =
mean(df$babies, na.rm = TRUE),
                                         sd = sd(df$babies,
na.rm = TRUE)), color = "red") +
  labs(title = "Histograma: babies", subtitle = "Distribución
+ curva normal")
```

```
p12 <- ggplot(df, aes(x = previous_cancellations)) +
  geom_histogram(aes(y = ..density..), bins = 15, fill =
"khaki3", color = "white", alpha = 0.8) +
  stat_function(fun = dnorm, args = list(mean =
mean(df$previous_cancellations, na.rm = TRUE),
                                         sd =
sd(df$previous_cancellations, na.rm = TRUE)), color = "red")
+
  labs(title = "Histograma: previous_cancellations", subtitle
= "Distribución + curva normal")
```

```
p13 <- ggplot(df, aes(x = previous_bookings_not_canceled)) +
  geom_histogram(aes(y = ..density..), bins = 20, fill =
"lightskyblue4", color = "white", alpha = 0.8) +
  stat_function(fun = dnorm, args = list(mean =
mean(df$previous_bookings_not_canceled, na.rm = TRUE),
                                         sd =
sd(df$previous_bookings_not_canceled, na.rm = TRUE)), color =
"red") +
  labs(title = "Histograma: previous_bookings_not_canceled",
subtitle = "Distribución + curva normal")
b11 <- ggplot(df, aes(x = babies)) +
  geom_boxplot(fill = "plum") +
  labs(title = "Boxplot: babies") +
  theme_classic()


b12 <- ggplot(df, aes(x = previous_cancellations)) +
  geom_boxplot(fill = "khaki3") +
  labs(title = "Boxplot: previous_cancellations") +
  theme_classic()


b13 <- ggplot(df, aes(x = previous_bookings_not_canceled)) +
  geom_boxplot(fill = "lightskyblue4") +
  labs(title = "Boxplot: previous_bookings_not_canceled") +
  theme_classic()
d3 <- (p11 | b11) /
      (p12 | b12) /
      (p13 | b13) +
  plot_annotation(
    title = "Análisis de outliers (Parte 3)",
    subtitle = "Distribuciones: bebés, cancelaciones y
reservas previas no canceladas",

  )
```

Viendo el grafico vamos a modificar algunos valores atípicos en “babies” porque es muy raro que alguien vaya con casi 10 bebes, lo de “previous_cancelllations” y “previous_bookings_not_canceled” son números posibles y comunes por lo que lo dejaremos tal cual.



#Visualización de datos atipicos

#babies

```
outliers<-boxplot(df$babies,plot=FALSE)$out
outliers
```

```
> outliers<-boxplot(df$babies,plot=FALSE)$out
> outliers
  [1]  1  1  1  1  2  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 [30]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 [59]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 [88]  1  2  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
[117]  1  1  1  1  1  1  2  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
[146]  1  1  1  1  1  1  1  1  1  1  1  1  2  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
[175]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  2  1  1  1  1  1  1  1  1
[204]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
[233]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
[262]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
[291]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
[320]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
[349]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
[378]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  2  1  1  1  1  1  1  1  1  1  1  1  1
[407]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  2  1  1  1
[436]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
[465]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  2  1  1
[494]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
[523]  1  1  1  1  1  1  1  1  1  1  1  1  2  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
[552]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
[581]  1 10  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
[610]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
[639]  1  1  1  1  1  1  1  9  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
[668]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  2  1  1  1  2  1  1  1  1
[697]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
[726]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  2  1  1  1  1  1  1  1  1  1  1  1  1
[755]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
[784]  1  2  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
[813]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  2  1  1  1  1  1  1  1  1  1  1  1  1  1
[842]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  2  1  1  1  1  1  1  1  1  1  1  1  1  1
[871]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
[900]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
```

#previous_cancellations

```
outliers<-boxplot(df$previous_cancellations,plot=FALSE)$out
```

outliers

```
> outliers
   [1]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  [30]  1  2  2  2  2  2  2  2  2  2  2  2  2  1  1  1  1  1  1  1  1  1  1  1  1  1  1  2
  [59]  3  1  1  1  1  1  1  1  1  1  1  1  1  2  2  1  1  1  1  1  1  2  2  2  1  1  2  1  1
  [88]  1  1  1  2  2  2  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  2  1  1  1 26  1
 [117]  1  1  1  1  1  1  1  1  2  2  1  1  1  1  1  1  1  1  1  1  1  1 25 25  1  1  1  1  1
 [146]  1  1  1  1  1  1  1  1  1  1  1  1  1 14  1  1  3  3  1  1  1  1  1  1  1  1  1  1  1
 [175]  2  3  4  4  4  4  2  1  1  1  1  2  2  1  1  1  1  1  1  1  1  1  1  1  1 24  1 19  1
 [204]  1  1  1  1  1  1  3  1  1  1  1  1  1  1  1  2  1  1  1  2  1  1  1  1  1  1  1  1  1
 [233]  1  1  1  1  1  1  1  1  1  2  2  2  1  1  2  2 24  1  1  1  1  1  1  1  1  1  1  1  1
 [262]  1  1  1  1  1  1  1  3  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 [291]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 [320]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 [349]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 [378]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  2  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 [407]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  2  2  1  1  1  1
 [436]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 [465]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  3  3  3  4  4  5
 [494]  5  5  2  3  1  1  1  1  1  1  1  1  1  1 21  1  1  1  1  1  1  1  1  2  2  1  1  1  1
 [523]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 [552]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 [581]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 [610]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 [639]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 [668]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 [697]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 [726]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 [755]  1  1  1  1  1  1  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2  2
 [784]  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  4  4  4  4  5  5  5  5  5  5  5
 [813]  5  5  5  5  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  6  1  1  1  1  1  1  1  2  2
 [842]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  2  2  2  2  3  3  3  3  3  3  4  4  4
 [871]  4  4  4  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 [900]  1  1  1  1  1  1  1  1  1  1  1  1  1  1 13 13 13 13  1  1  1  1  1  1  1  1  1  1  1
 [929]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 [958]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 [987]  1  1  1  1  1  1  1  1  1  1  1  1  1  1
 [ reached 'max' / getOption("max.print") -- omitted 685 entries ]
```

#previous_bookings_not_canceled

```
outliers<-
boxplot(df$previous_bookings_not_canceled,plot=FALSE)$out
outliers
```

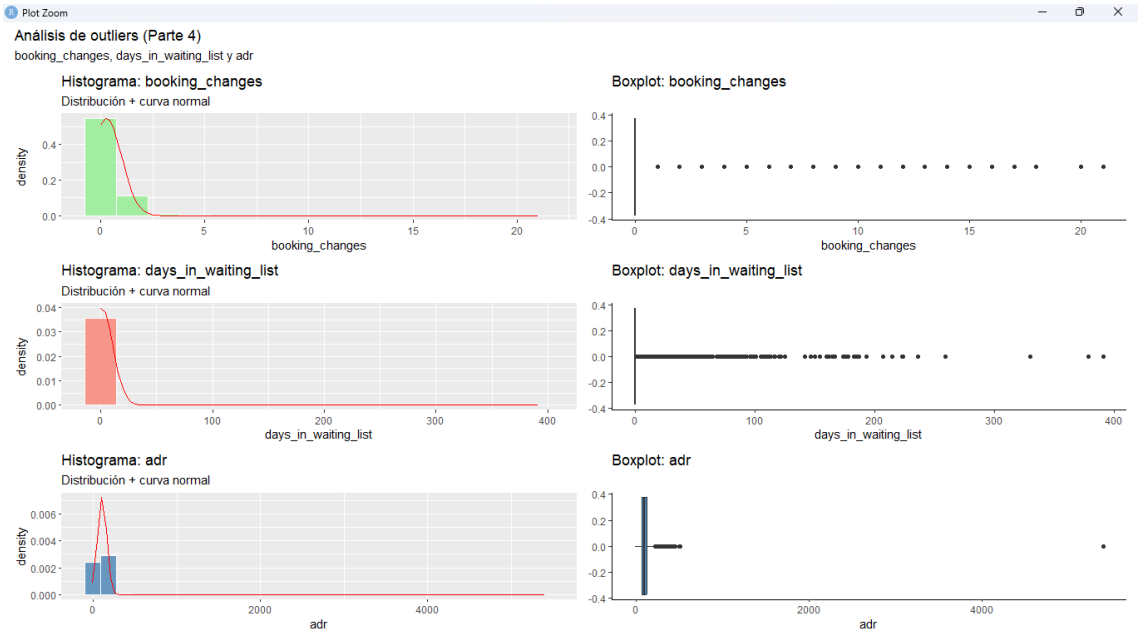
```
> outliers
   [1]  1  2  3  1  1  1  2  3  1  2  1  2  3  1  1  2  3  1  1  2  3  1  2  1  1  1  2  1  2
  [30]  1  1  3  4  5  1  1  2  3  4  1  2  1  1  2  3  1  2  1  2  3  3  4  5  6  7  8  1  1
  [59]  2  3  4  5  6  2  3  4  5  1  1  2  3  4  5  6  7  8  9 10 11 12  1  2  3  4  1  2  1
  [88]  2  3  4  5  6  7  8  1  1  2  3  4  1  1  1  1  2  3  1  2  2  2  3  4  5  6  7  8  9
 [117] 10 11 12 13  1  1  2  1  2  3  1  2  1  2  3  4  5  1  1  1  2  3  4  5  1  1  2  3  4
 [146]  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 18 20 21 22 23 24 25 25 27 27 28 29 30  1
 [175]  1  2  3  4  5  1  2  1  2  3  4  5  1  2  1  1  1  2  1  2  1  2  1  2  1  1  1  1  2
 [204]  1  1  1  1  1  1  2  3  4  5  6  1  2  3  4  1  1  2  1  2  1  1  1  2  3  4  2  1  1
 [233]  1  1  1  2  3  4  5  1  2  3  4  5  6  1  2  3  1  2  3  4  1  2  1  2  3  4  5  6  7
 [262]  8  9  1  1  1  1  2  1  2  3  4  1  2  1  2  3  4  5  1  2  3  1  2  3  1  2  2  3  4
 [291]  5  6  1  1  1  2  1  2  2  1  2  2  1  2  3  4  5  6  7  1  2  3  4  5  6  1  1  2  1
 [320]  2  2  3  4  5  6  7  8  9 10 11 12 13 14 14  1  2  3  4  5  6  3  3  4  5  6  7  8  8
 [349]  9  1  1  2  3  4  5  6  1  3  4  5  6  7  8  9 10 11  1  1  1  1  1  1  1  2  1  1  1
 [378]  1  2  1  1  3  4  5  5  1  2  3  2  2  2  2  2  3  4  5  6  1  1  1  2  1  2  3  4  5
 [407]  1  1  1  2  1  1  2  1  1  2  3  1  2  3  4  1  2  3  1  1  2  2  1  1  2  3  2  3  4
 [436]  5  6  7  8  9 10  1  2  3  4  5  6  1  1  2  3  1  2  3  4  1  1  2  3  4  2  3  1  2
 [465]  3  4  1  2  3  1  1  2  3  1  2  2  3  1  2  3  1  2  1  1  1  2  3  4  1  1  1  1  1
 [494]  2  1  1  2  1  2  1  1  3  1  2  1  2  3  4  5  6  1  1  1  1  2  3  1  1  1  1  2  3
 [523]  2  3  1  1  1  4  1  2  3  3  3  3  4  5  6  7  8  9 10 10 11 11 12 13  1  1  2  3  3
 [552]  5  1  1  2  3  1  2  3  1  4  5  6  7  8  9  1  1  2  3  1  2  1  2  1  2  3  1  1  1
 [581]  2  3  4  1  2  1  1  2  1  2  1  2  3  3  4  1  1  1  1  2  3  4  5  6  7  8  8  9 10
 [610]  1  1  2  1  1  1  1  2  3  4  5  6  7  8  9 10  1  1  1  2  2  3  4  1  1  2  2  4  5
 [639]  6  7  7  8  9 10 11 12 13  1  1  1  2  1  2  3  1  1  1  1  2  1  1  1  1  2  3  4  5
 [668]  6  7  8  9 10  1  1  2  1  2  3  4  1  2  3  4  5  5  6  7  1  2  3  1  2  3  4  5  6
 [697]  7  8  9 10 11 12  1  1  2  3  3  5  6  1  1  1  2  1  1  2  1  2  3  4  1  2  1  1  1
 [726]  2  1  2  1  1  1  1  2  3  4  5  1  1  1  1  1  1  1  1  1  1  1  2  3  4  5  6  1
 [755]  1  1  1  1  1  1  1  1  2  1  1  1  2  3  1  1  1  1  2  3  4  5  6  1  2  1  2  3  4
 [784]  4  5  6  7  8  9  1  1  1  1  1  2  3  3  1  1  1  1  2  3  4  5  1  1  1  2  3  1  2
 [813]  3  4  5  5  1  1  1  1  1  1  1  1  1  1  1  2  1  1  2  1  2  1  2  3  4  5  6  7  8
 [842]  9 10 10  1  2  3  4  1  2  1  2  3  1  2  3  4  5  1  2  3  4  5  6  7  8  1  1  1  2
 [871]  3  4  5  6  1  2  1  1  1  2  3  4  5  1  1  1  2  3  4  5  1  2  2  3  4  5  6  7  8
 [900]  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26  1  1  1  1  1  1  2  3  4  5  1
 [929]  2  1  1  1  1  1  1  1  1  2  2  4  5  6  2  1  1  1  1  2  1  1  2  3  1  1  1  2  1
 [958]  2  1  2  3  4  5  6  7  1  2  2  4  5  1  1  1  2  3  1  2  1  1  1  2  2  1  1  1  2
 [987]  3  1  2  1  2  3  1  2  3  4  1  1  2  3
 [ reached 'max' / getOption("max.print") -- omitted 2545 entries ]
```

Parte 4, “booking_changes”, “days_in_waiting_list” y “adr”

```
p14 <- ggplot(df, aes(x = booking_changes)) +
  geom_histogram(aes(y = ..density..), bins = 15, fill =
"lightgreen", color = "white", alpha = 0.8) +
  stat_function(fun = dnorm, args = list(mean =
mean(df$booking_changes, na.rm = TRUE),
                                         sd =
sd(df$booking_changes, na.rm = TRUE)), color = "red") +
  labs(title = "Histograma: booking_changes", subtitle =
"Distribución + curva normal")


p15 <- ggplot(df, aes(x = days_in_waiting_list)) +
  geom_histogram(aes(y = ..density..), bins = 15, fill =
"salmon", color = "white", alpha = 0.8) +
  stat_function(fun = dnorm, args = list(mean =
mean(df$days_in_waiting_list, na.rm = TRUE),
                                         sd =
sd(df$days_in_waiting_list, na.rm = TRUE)), color = "red") +
  labs(title = "Histograma: days_in_waiting_list", subtitle =
"Distribución + curva normal")


p16 <- ggplot(df, aes(x = adr)) +
```

```
  geom_histogram(aes(y = ..density..), bins = 30, fill =
"steelblue", color = "white", alpha = 0.8) +
  stat_function(fun = dnorm, args = list(mean = mean(df$adr,
na.rm = TRUE),
                                         sd = sd(df$adr,
na.rm = TRUE)), color = "red") +
  labs(title = "Histograma: adr", subtitle = "Distribución +
curva normal")
b14 <- ggplot(df, aes(x = booking_changes)) +
  geom_boxplot(fill = "lightgreen") +
  labs(title = "Boxplot: booking_changes") +
  theme_classic()


b15 <- ggplot(df, aes(x = days_in_waiting_list)) +
  geom_boxplot(fill = "salmon") +
  labs(title = "Boxplot: days_in_waiting_list") +
  theme_classic()


b16 <- ggplot(df, aes(x = adr)) +
  geom_boxplot(fill = "steelblue") +
  labs(title = "Boxplot: adr") +
  theme_classic()
d4 <- (p14 | b14) /
  (p15 | b15) /
  (p16 | b16) +
  plot_annotation(
    title = "Análisis de outliers (Parte 4)",
    subtitle = "booking_changes, days_in_waiting_list y adr",

  )
```

Aquí notamos que sí puede haber como 30 cambios en la reserva, pero es un valor muy raro y eso afectaría en nuestras estadísticas por lo que podemos modificar los valores atípicos, en "days_in_waiting_list" lo normal es hasta 180 días, pero después ya es muy raro, por lo que también podemos modificar los valores atípicos. Y en "adr" notamos en el grafico un punto muy al extremo derecho por lo que eso sí es probablemente un error por lo que lo reemplazaremos.



#Visualización de datos atipicos

#booking_changes

```
outliers<-boxplot(df$booking_changes,plot=FALSE)$out
outliers
```

```
> outliers
   [1]  3  4  1  1  1  2  1  1  2  1  1  1  1  2  1  3  1  1  3  1  1  1  1  1  5  1  2  2  1
  [30]  1  2  2  1  1  1  1  1  1  2  1  1  1  1  1  2  1  2  1  2  2  1  2  1  1  1  1  1
  [59]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  3  3  1  4  1  1  1  1  2  3  1  1  1  1  1
  [88]  1  1  1  1  2  1  1  1  1  1  1  1  1  1  5  1  1  1  1  1  1  1  2  1  1  1  1  1  1  1
 [117]  1  1  2  1  4  3  1  1  1  1  1  1  1  1  1  1  2  1  1  1  2  1  1  1  1  1  1  1  1
 [146]  1  1  1  1  1  1  1  1  1  1  2  1  1  1  1  1  1  1  2  1  2  1  1  1  1  3  1  1  1
 [175]  1  4  1  2  1  1  1  1  1  1  2  1  1  1  1  1  2  1  1  1  2  1  1  1  1  1  1  1  1
 [204]  1  1  1  1  1  1  5  1  3  1  2  1  1  1  2  2  2  2  1  2  1  1  1  1  1  2  1  1  1
 [233]  1  1  1  2  2  1  1  1  1  1  1  3  2 17  1  1  1  1  1  1  1  1  1  1  2  2  1  2  1
 [262]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  2  3  1  1  4  1  1  1  1  3  1  1  1
 [291]  1  1  1  3  1  1  1  1  1  1  1  1  1  1  1  1  1  1  2  3  1  1  2  2  1  2  2  1  1  4
 [320]  1  1  1  2  1  1  1  1  2  1  1  3  1  1  2  1  2  1  1  1  1  1  1  1  1  1  1  1  2
 [349]  1  1  1  1  1  1  1  1  1  2  2  1  2  2  4  1  1  1  2  1  1  2  1  1  1  1  2  1  1
 [378]  1  2  1  1  1  1  1  1  1  1  1  1  1  2  2  1  2  6  2  2  1  1  3  1  1  1  2  1  1
 [407]  1  1  2  2  3  2  1  1  1  2  2  1  2  2  1  1  1  1  2  1  1  2  1  1  1  1  2  1  3
 [436]  4  2  1  4  1  1  1  3  2  1  4  1  3  3  1  2  2  1  3  1  4  1  1  1  2  2  1  1  1
 [465]  1  4  1  1  1  1  1  1  1  1  1  1  1  1  1  1  2  1  3  3  1  1  1  3  2  2  1  1  1  1
 [494]  1  1  1  1  1  1  3  2  1  1  1  2  4  1  1  1  2  1  1  1  1  1  1  1  1  1  1  1  1
 [523]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  2  1  1  1  3
 [552]  5  1  1  1  3  1  1  1  1  2  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  2
 [581]  1  1  1  1  2  1  1  4  1  1  1  2  1  2  1  1  1  1  1  1  1  1  2  1  1  1  1  1  1
 [610]  1  1  1  1  1  1  1  1  1  1  1  1  1  2  1  1  1  1  1  1  1  2  2  1  1  4  1  2  2
 [639]  2  1  2  1  1  1  2  2  1  1  1  1  1  2  1  1  2  2  1  1  1  1  1  2  2  1  2  2  1
 [668]  2  1  1  3  2  2  2  1  3  1  1  1  1  2  1  1  2  1  1  4  1  4  6  1  1  2  2  1  1
 [697]  1  1  1  1  1  2  1  1  1  1  1  1  1  2  1  1  1  1  1  1  1  1  2  1  1  2  2  2  1
 [726]  1  2  1  1  1  1  1  1  4  3  1  1  1  1  1  1  1  1  3  1  1  2  2  2  1  2  1  1  2
 [755]  1  1  1  1  1  1  3  1  1  2  1  1  1  2  2  1  1  1  1  2  1  1  1  1  1  1  1  1  1
 [784]  1  1  1  2  2  1  1  2  2  1  2  1  1  2  2  1  1  1  1  1  1  1  3  2  1  2  1  1  2
 [813]  1  3  3  4  1  2  4  4  5  2  2  2  1  1  2  2  2  2  2  2  2  2  2  2  2  2  2  3  2
 [842]  3  2  2  1  2  3  1  4  1  1  1  1  3  3  5  1  1  2  2  1  3  2  2  1  1  1  1  1  1
 [871]  2  2  2  1  1  2  3  1  2  1  1  3  1  1  1  1  1  1  1  1  1  2  1  1  1  1  1  1  1
 [900]  1  1  2  1  1  1  4  2  1  1  1  1  1  1  2  1  1  1  1  1  1  1  1  1  2  3  1  1  1
 [929]  1  2  1  1  2  1  1  1  2  1  1  2  1  4  1  2  2  1  2  1  2  1  1  1  1  2  1  1  1
 [958]  1  1  1  2  2  2  1  1  1  3  1  1  1  1  1  1  1  2  2  1  2  2  1  1  1  1  1  1  4
 [987]  1  1  1  1  1  2  1  2  1  1  2  1  1  1
 [ reached 'max' / getOption("max.print") -- omitted 14902 entries ]
```

#days_in_waiting_list

```
outliers<-boxplot(df$days_in_waiting_list,plot=FALSE)$out
outliers
```

```
> outliers
  [1]  50  47  47  65  65  65 122 122 122 122 122 122 122 122 122 122 122 122 122 122 122 122
 [23] 122 122 122 122  75  75  75  75  75  75 101 101 101 101 101 101 101 101 101 150 150 150
 [45] 125 125 125 125 125 125 125 125 125  14  60  34  50  50 100  22 121  61  39   5   5   1
 [67]   8  47  65   1 107 125  43  52   2  11 150 150 150  22 122 142   2 116  13  44  97   1
 [89]  97   1  97  83   4  14   4   4 113 113 113 113 113  18  20 185 185 107  93 109  34   6
[111]   6  13   8  60  60  37  60  60  43   6   6 105 100  34   1 154  14  14 154  14   5  14
[133]  64  64  64 121   8  61  61  20  99  99  38  38  38  38  38  48  48  48  48  65  65  33
[155]  33  33  33  33  33  77  77  77  77  77  77  77  77  21  80  80  80  59  59  59  59  59
[177]  59  59  40  40  77  77  77  77  40  77  77  40  77  77  40  58  58  58  89  53  53  53
[199]  53  50  50  50  49  50  50  49  49  50  58  58  58  58  58  58  58  58  58  58  58  58
[221]  58  58  58  58  58  58  60  60  69  69  87  87  87  87  87  87  87  87  87  87  87  87
[243]  87  87  87  87  87  87  87  87  87  87  87  87  87  91  69  57  57  69  69  69  69  69
[265]  99  99  99  99  99  99 111 111 111 111 111 111  79  98  91  85  63  63  63  63  63  63
[287]  63  63  63  63  63  63  63  63  63  63  63  63  63  63  63  63  63  63  63  63  63  63
[309]  63  63  63  63  63  63  63  63  63  63  63  63  63  63  63  63  63  63  63  63  63  63
[331]  15   3  20  93  93 101 101 101 101   4  41  41  34  34 224 224 224 224 224  31  39  39
[353]  56 187 187 187 176 176 187 187 187 176 187 187 176 176 176 187 187  77  48  48  48  50
[375]  71  55  96 236 236 259  60  96 236 259 259 259 259 259 259 236 259 259 207 207 207 207
[397] 207 176 176 176 176 176 215 215 215 215 160 236 236 236  75 120  98  98  98  98  98 111
[419]  30  31   3  32  27  69  62  48  48  48  48  48  48  24  39  80  39  80 108 108 108 108
[441] 108 108 108  39  39  15  40  40   1   1  96 147   5  41  32  14 107 107 379 379 379  70
[463] 107  38   8   8   8  35  35 178 178 178 330 330 330 330 223 223 223 223 223 223 223 223
[485] 223  69 174  24 162 162 391  68  68  11  11 391 391  20  20 391 391 193  10  50   8  76
[507]  76  16  20  16  22  22  28  21  16   9   9 165   2  44  17  53  16   5  38  43  28  50
[529] 100 111  25  17  46  46  21   4   7 113  46  18  47  28  27  49  84  30 175 183  39  18
[551]   5  20  56  56  44  27  31  17  10  46  60  60  14   7  17  15  23  13  63  33  32  34
[573]  93  44  44   7  11  35  99 117  57  34  12  11  23  28  80  34  71  54  59  59  59  26
[595]  26  59  73  59  71  26  26  71  71  71  71  31  31  31  17  17  17  17  39  45   6   6
[617]   6  17  31  19  19  15  38  27  27  27  27  27  27  27  27  27  27  27  27  35  35  21
[639]  28  31  31  40  42  25  10  50  44  44  44  44  46  46  46  46  42  41  22  17  56  41
[661]   9  28   1  43  43  43  43  43  43  43  31  31  31  15  15  15  15  15  15  15  15  15
[683]  15  15  15  15  15  15  15  15   4   4   4   4   4   4   4   4   4  22  56  34  34  34
[705]  34  34  34  34  72  56  56  56  56  56  56  48  48  48  48  48  48  48  48   1  12  98
[727]  98  98  98  98  98   1  25  25   2  12  61  61  61  61  61  61  27  96  96  96  96  96
[749]  27 147 147 147 147 147 147 147 147 147 147 147 147 147 147  49  49  44  44  44  44  44  44
[771]  44  44  44  44  44  44  44  44  38  38  38  38  38  38  25   5   6 178 178 178 178 178
[793] 178 178 178 178  38  38  38  38  38  11  11 174 174 174 174 162  81 162 162 162 162  68
[815]  68  68  68  68  99  92  74   6  21  10   9 167  35  35  15  25  25  25  25  13  28  14
[837]  83   2  49   4   1 223   5 105  50  65   7  63  63  49  22  46  84  25   1  36   9  98
[859]  59  23
```

#adr

```
outliers<-boxplot(df$adr,plot=FALSE)$out
outliers
```

```
> #adr
> outliers<-boxplot(df$adr,plot=FALSE)$out
> outliers
  [1] 230.67 249.00 241.50 240.64 233.00 240.00 233.05 240.00 250.33 280.74 252.00 233.00
 [13] 237.00 230.50 230.50 241.00 242.60 268.00 239.30 267.00 277.50 250.00 246.00 252.00
 [25] 276.43 228.00 277.00 254.00 233.00 241.00 274.93 252.00 258.33 255.00 243.00 243.00
 [37] 266.40 236.00 271.00 232.00 229.00 266.00 262.00 234.00 242.50 248.00 299.33 248.00
 [49] 236.00 229.67 239.50 241.00 241.00 229.00 236.00 248.00 260.71 259.00 229.00 233.00
 [61] 231.60 261.40 332.00 270.00 276.60 232.00 272.00 260.00 238.63 235.00 231.43 237.33
 [73] 280.00 236.67 240.00 240.00 239.00 242.00 250.00 237.00 233.00 252.00 287.00 259.00
 [85] 247.00 252.00 240.00 240.00 240.00 259.00 240.00 288.00 262.00 243.32 259.00 241.00
 [97] 292.00 259.00 266.50 253.57 241.00 240.00 232.00 259.00 256.50 252.00 244.50 282.00
[109] 250.00 240.00 283.32 231.00 231.00 272.70 230.00 240.00 231.00 246.00 241.00 236.00
[121] 259.00 233.00 299.00 245.67 248.75 248.89 298.00 289.00 262.00 251.00 274.00 230.00
[133] 299.00 241.00 273.00 269.00 269.00 254.00 259.00 236.71 259.00 243.63 231.00 243.63
[145] 369.00 262.00 278.60 246.50 271.00 234.00 240.00 254.31 240.00 261.50 246.00 259.00
[157] 231.50 251.00 241.00 256.00 259.00 291.00 241.00 249.00 251.50 234.60 234.00 279.00
[169] 241.00 259.00 254.00 277.67 299.00 227.92 258.00 247.67 269.00 263.00 235.57 309.00
[181] 289.90 241.00 230.00 248.16 261.00 236.67 236.50 259.00 256.00 246.00 231.00 246.00
[193] 231.00 229.00 230.00 314.50 266.50 258.00 286.79 281.00 239.00 239.00 231.00 274.00
[205] 227.92 275.00 237.00 247.33 256.75 288.00 251.86 238.16 259.00 234.00 274.00 304.00
[217] 286.00 329.00 231.00 235.67 281.00 251.73 274.00 231.00 249.00 229.00 271.00 322.00
[229] 241.00 287.00 239.00 239.10 248.00 269.00 229.00 265.67 249.50 240.00 249.50 262.00
[241] 253.00 232.25 322.00 269.00 234.00 240.00 243.80 241.75 247.57 246.67 246.67 231.80
[253] 231.80 234.00 227.10 240.60 252.00 264.00 254.00 246.00 292.40 233.00 246.02 340.00
[265] 384.00 250.00 302.11 382.00 275.00 229.40 243.00 228.00 228.00 262.50 228.00 233.00
[277] 260.00 238.00 273.00 261.00 248.00 248.00 311.00 248.00 258.00 228.00 288.00 228.00
[289] 238.00 248.00 265.00 260.00 250.00 243.00 238.00 300.86 292.00 238.71 243.71 232.00
[301] 259.86 242.00 241.00 264.00 247.00 241.00 265.00 274.67 270.00 239.00 303.00 242.75
[313] 293.00 232.00 275.00 260.00 230.86 270.00 260.00 272.00 230.30 293.33 253.25 311.00
[325] 240.00 233.10 289.60 240.00 252.00 252.00 232.00 229.00 234.67 242.00 338.00 230.00
[337] 229.00 302.00 243.16 342.29 274.45 244.00 234.56 290.67 290.67 260.00 255.00 255.00
[349] 306.00 232.91 303.00 249.50 250.00 237.34 245.00 237.00 289.00 229.00 229.00 278.00
[361] 289.50 317.00 353.00 257.00 315.00 244.00 265.00 244.00 230.00 248.00 278.57 315.00
[373] 234.00 245.00 277.00 270.71 277.00 244.00 228.00 292.00 237.00 305.00 237.00 299.00
[385] 232.00 232.00 245.00 262.00 292.00 244.00 255.00 245.00 255.00 245.00 267.86 245.00
[397] 266.50 292.00 292.00 257.00 249.00 229.50 279.50 310.00 245.00 257.00 315.00 252.00
[409] 278.00 315.00 287.00 237.50 244.00 363.00 363.00 248.00 246.00 305.00 244.00 242.10
[421] 229.33 266.60 266.60 257.00 252.00 311.70 294.86 250.00 230.00 229.00 248.00 299.00
[433] 309.10 257.00 229.17 266.30 228.57 227.80 230.00 230.00 230.00 231.00 301.43 258.27
[445] 242.25 315.71 244.72 450.00 264.00 300.40 253.33 241.60 257.34 229.00 266.00 282.29
[457] 230.00 250.00 270.00 269.50 230.00 283.60 378.00 244.00 358.75 328.00 230.00 252.00
[469] 330.00 259.33 230.00 292.00 250.00 269.00 298.00 378.00 230.00 310.00 262.00 230.00
[481] 230.00 231.00 239.00 323.00 252.00 297.00 269.00 239.00 239.00 239.00 259.00 303.70
[493] 230.00 250.00 294.50 275.00 240.00 378.00 240.00 317.00 262.00 298.00 248.33 252.00
[505] 230.00 262.00 292.00 292.00 231.84 297.00 230.00 259.00 258.00 392.00 294.50 280.00
[517] 273.25 230.00 230.00 230.00 270.00 264.29 250.00 231.60 230.00 252.00 250.00 259.43
[529] 262.00 229.52 297.38 230.00 300.00 232.33 303.20 340.00 230.00 256.57 262.00 228.00
[541] 232.00 230.00 261.00 230.00 437.00 249.00 230.00 230.00 264.00 230.00 388.00 249.00
[553] 274.50 262.00 254.00 310.00 240.00 262.00 297.00 308.00 246.80 259.00 244.00 230.00
[565] 230.00 273.00 245.00 231.43 270.00 230.00 310.00 249.00 230.00 229.00 230.00 252.00
[577] 230.00 230.00 264.00 308.00 262.00 241.00 289.80 378.00 338.00 311.50 318.00 240.00
[589] 247.00 290.00 260.00 238.57 270.00 290.00 270.00 237.50 230.00 230.00 282.00 262.00
[601] 270.00 249.00 330.00 250.00 265.00 257.60 257.60 250.00 231.60 227.22 234.00 238.00
[613] 250.00 340.00 248.00 230.00 235.71 230.00 247.20 227.10 244.00 244.00 270.00 250.00
[625] 268.00 236.00 343.00 254.00 230.00 289.60 255.45 234.62 260.00 270.00 340.00 270.00
[637] 244.00 297.00 378.00 295.50 242.00 232.00 230.00 240.91 278.14 260.00 280.00 230.00
[649] 237.50 319.00 268.00 270.00 238.57 230.00 252.00 240.00 255.00 239.14 229.43 250.00
[661] 260.00 270.00 249.00 229.00 331.33 251.43 259.00 249.00 235.00 240.00 243.33 303.33
[673] 295.67 295.67 242.00 342.17 262.38 284.86 251.43 340.86 251.00 299.43 294.29 293.86
[685] 245.30 290.00 290.00 242.00 229.00 244.16 295.00 289.80 253.80 230.00 236.67 230.00
[697] 230.00 236.00 230.00 286.25 230.00 341.00 230.00 312.00 232.50 250.00 297.00 265.00
[709] 293.60 328.00 228.57 233.33 230.00 284.00 290.00 230.00 248.10 228.00 253.00 254.00
[721] 236.50 305.00 241.00 233.92 335.00 230.84 245.50 288.10 275.25 367.00 239.00 229.50
[733] 251.00 232.00 244.00 237.00 227.86 237.00 508.00 236.00 317.00 318.82 232.00 252.00
[745] 248.00 252.00 241.00 256.00 244.00 227.40 247.00 244.00 236.00 241.00 252.00 297.50
[757] 243.00 253.00 232.00 242.67 237.45 241.00 241.00 236.93 233.00 229.00 266.00 252.00
[769] 235.71 234.29 279.00 229.00 271.00 244.00 274.00 241.00 239.00 258.43 235.07 234.67
[781] 229.60 239.65 264.50 256.50 244.00 259.00 284.00 300.00 305.00 237.33 305.00 245.33
[793] 245.20 237.00 232.40 249.00 274.00 254.00 252.17 236.00 230.00 246.00 282.00 311.33
[805] 249.50 244.00 240.86 263.57 262.50 239.00 287.50 271.00 231.00 271.00 296.00 271.00
[817] 249.00 230.00 232.00 231.88 295.00 243.00 257.00 276.60 252.60 252.60 271.00 259.00
```

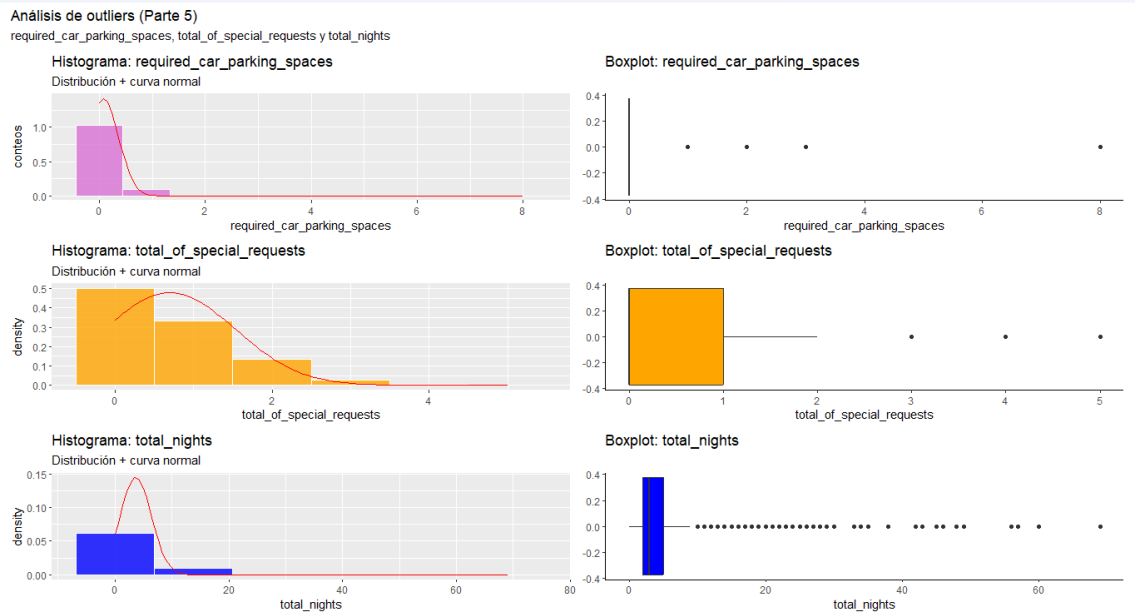
```
[829] 253.00 241.00 278.00 281.00 235.00 231.00 241.00 240.00 249.00 239.00 318.00 276.00
[841] 259.00 229.00 242.00 231.00 308.40 232.00 237.75 274.00 301.00 229.00 251.00 260.33
[853] 248.00 241.00 281.00 237.67 302.86 288.10 236.00 231.00 257.00 307.00 231.00 264.00
[865] 233.00 274.00 282.00 229.00 240.50 241.00 241.00 265.83 251.00 251.00 241.00 263.00
[877] 340.71 240.57 229.50 283.00 252.00 283.20 273.50 253.00 263.00 336.50 259.00 253.00
[889] 251.00 251.83 246.00 318.71 234.17 232.33 268.33 231.00 352.00 335.00 249.00 251.00
[901] 359.00 236.00 253.50 310.00 239.08 353.67 269.00 276.00 230.00 245.00 313.71 271.00
[913] 240.00 236.00 262.60 288.00 234.70 229.00 268.00 229.00 244.00 231.17 241.50 243.50
[925] 269.00 229.71 246.20 242.33 276.00 256.00 268.00 231.00 229.00 357.00 231.00 246.00
[937] 261.00 227.10 243.00 261.00 239.00 244.00 286.00 327.40 233.00 296.00 251.07 327.60
[949] 235.29 227.10 329.00 268.50 244.67 229.00 256.00 281.00 243.00 272.67 253.00 249.67
[961] 263.00 254.00 271.00 241.00 279.00 243.17 239.67 234.00 239.00 263.00 279.00 241.00
[973] 270.00 251.00 227.67 359.00 322.00 315.38 236.00 306.00 286.00 278.50 253.00 235.00
[985] 235.00 235.00 253.16 299.00 231.76 250.00 230.00 236.00 270.50 293.00 261.00 243.00
[997] 253.00 253.00 261.00 257.67
[ reached 'max' / getOption("max.print") -- omitted 1490 entries ]
```

Parte 5, “required_car_parking_spaces” y “total_of_special_requests”

```
p17 <- ggplot(df, aes(x = required_car_parking_spaces)) +
  geom_histogram(aes(y = ..density..), bins = 10, fill =
"orchid", color = "white", alpha = 0.8) +
  stat_function(fun = dnorm, args = list(mean =
mean(df$required_car_parking_spaces, na.rm = TRUE),
                                         sd =
sd(df$required_car_parking_spaces, na.rm = TRUE)), color =
"red") +
  labs(title = "Histograma: required_car_parking_spaces", y=
'conteos', subtitle = "Distribución + curva normal")


p18 <- ggplot(df, aes(x = total_of_special_requests)) +
  geom_histogram(aes(y = ..density..), bins = 6, fill =
"orange", color = "white", alpha = 0.8) +
  stat_function(fun = dnorm, args = list(mean =
mean(df$total_of_special_requests, na.rm = TRUE),
                                         sd =
sd(df$total_of_special_requests, na.rm = TRUE)), color =
"red") +
  labs(title = "Histograma: total_of_special_requests",
subtitle = "Distribución + curva normal")


p8 <- ggplot(df, aes(x = total_nights)) +
  geom_histogram(aes(y = ..density..), bins = 6, fill =
"blue", color = "white", alpha = 0.8) +
  stat_function(fun = dnorm, args = list(mean =
mean(df$total_nights, na.rm = TRUE),
                                         sd =
sd(df$total_nights, na.rm = TRUE)), color = "red") +
  labs(title = "Histograma: total_nights", subtitle =
"Distribución + curva normal")

b17 <- ggplot(df, aes(x = required_car_parking_spaces)) +
  geom_boxplot(fill = "orchid") +
  labs(title = "Boxplot: required_car_parking_spaces") +
```

```
  theme_classic()

b18 <- ggplot(df, aes(x = total_of_special_requests)) +
  geom_boxplot(fill = "orange") +
  labs(title = "Boxplot: total_of_special_requests") +
  theme_classic()

b8 <- ggplot(df, aes(x = total_nights)) +
  geom_boxplot(fill = "blue") +
  labs(title = "Boxplot: total_nights") +
  theme_classic()


d5 <- (p17 | b17) /
  (p18 | b18) /
  (p8| b8) +
  plot_annotation(
    title = "Análisis de outliers (Parte 5)",
    subtitle = "required_car_parking_spaces,
total_of_special_requests y total_nights"

  )
```

Notamos según el grafico que hay algunos puntos muy atípicos en “required_car_parking_spaces” por lo que lo vamos a reemplazar y en “total_of_special_requests” no porque la diferencia no es mucha en “total_nights” sí notamos algunos puntos muy a la derecha por lo que vamos a reemplazar sus valores atípicos.



```
#Visualización de datos atipicos

#required_car_parking_spaces

outliers<-
boxplot(df$required_car_parking_spaces,plot=FALSE)$out
outliers
```

```
> #Visualización de datos atipicos
> #required_car_parking_spaces
> outliers<-boxplot(df$required_car_parking_spaces,plot=FALSE)$out
> outliers
   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
  [45] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
  [89] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [133] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [177] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [221] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [265] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [309] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [353] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [397] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [441] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [485] 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [529] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [573] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [617] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1
 [661] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [705] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [749] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [793] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [837] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [881] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [925] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [969] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1
 [ reached 'max' / getOption("max.print") -- omitted 6313 entries ]
> #total_of_special_requests
```

```
#total_of_special_requests

outliers<-
boxplot(df$total_of_special_requests,plot=FALSE)$out
outliers
```

```
> #total_of_special_requests
> outliers<-boxplot(df$total_of_special_requests,plot=FALSE)$out
> outliers
   [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 3 4 3 4 3 3 4 3 3 3 3 3 3 3 3 3 3
  [45] 3 3 3 4 3 3 3 3 3 3 3 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
  [89] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 3 3 3 3 3 3 3 4 4 3 3 3 3
 [133] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 5 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 [177] 3 3 3 3 3 3 3 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 5 3 3 3 3 3 4 4 3 3 4 3 4 3 3
 [221] 3 3 3 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 3 3 3 4 3 3 3 4 3 3 3 3 3 3 3 3 3 4 4 3 3 3 4
 [265] 3 3 3 3 3 3 3 3 3 3 5 3 3 4 4 3 3 3 3 3 3 4 4 4 4 4 4 4 4 5 5 4 4 4 3 3 3 3 4 4 4 4 4 4 4
 [309] 4 4 4 4 3 3 3 3 3 3 3 3 3 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 3 3 3 3 3 4 3 3 4 3 3 3 3 3
 [353] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 3 3 3 3 3 3 3 3
 [397] 3 3 3 3 3 3 3 3 3 3 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 4 3 3 3 3 3 3 3 3 3 4 3 3 3 4 3
 [441] 3 3 3 3 3 3 3 3 3 3 3 3 3 4 3 4 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 5 3 3
 [485] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 3 3 3 3 3 3 3 3 3 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 [529] 3 3 3 3 3 3 3 3 4 3 5 3 3 3 3 3 3 3 5 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 4 3 3 3 3 3 4 3 3
 [573] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 3 4 3 3 3 3 3 4 3 3 5 3 4 4 4 4 4 5 3 3 3 3 3 4 3 3
 [617] 3 3 3 3 3 3 3 3 3 3 3 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 4 4 4 3 3 3
 [661] 3 3 3 3 3 3 4 3 3 3 3 3 3 3 3 3 5 3 3 3 3 3 3 4 3 3 3 4 3 3 3 3 3 3 3 3 4 3 3 3 3 3 3 3 3
 [705] 3 3 3 3 3 3 4 3 3 3 3 3 3 3 3 3 3 3 4 3 3 3 3 4 3 3 3 4 3 3 3 3 3 4 3 4 3 3 3 3 4 3 3 3 3 3 3 3
 [749] 3 3 3 3 3 3 3 3 4 4 4 3 3 3 3 3 4 3 3 4 4 3 3 3 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 3 3 3 3 3 3 3 3
 [793] 3 3 3 3 3 4 4 4 3 3 3 3 3 3 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 [837] 3 3 3 3 3 3 3 3 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 3 3 3 3 4 3 4 3 3 4 5 4 3 3 3 3 4 3 3 4
 [881] 5 4 3 3 3 3 3 3 4 3 3 3 3 4 3 3 3 3 4 3 3 4 3 3 3 3 3 3 3 4 4 4 3 4 4 4 4 4 3 3 3 3 3 3 3
 [925] 4 3 3 4 4 4 4 3 3 4 4 3 4 3 3 3 5 3 3 4 3 3 3 3 3 3 4 3 4 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 [969] 3 3 3 4 3 3 3 3 3 3 3 3 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 [ reached 'max' / getOption("max.print") -- omitted 1673 entries ]
```

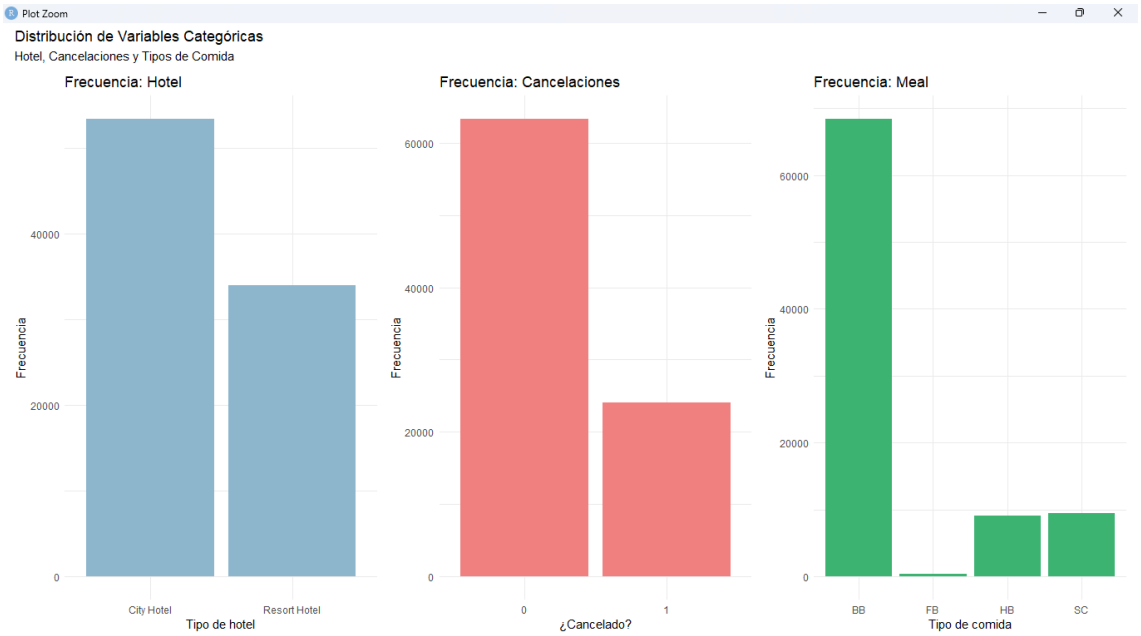
#total_nights

```
outliers<-boxplot(df$total_nights,plot=FALSE)$out
outliers
```

```
> outliers
   [1] 14 15 10 11 14 10 10 10 10 14 14 21 10 10 10 10 10 11 14 14 10 10 10 10 10 11 14 14 14
  [30] 10 11 13 10 11 11 14 11 14 14 15 14 10 14 10 14 10 10 11 10 10 10 14 10 12 14 10 10 10
  [59] 10 14 14 14 10 10 13 14 14 11 10 10 12 12 10 10 14 14 18 10 14 14 10 10 10 10 10 10 10
  [88] 14 10 10 11 12 10 11 10 12 14 10 10 10 14 10 10 10 10 14 14 10 10 11 14 46 10 14 14 10
 [117] 10 14 10 18 11 12 12 14 14 14 28 14 15 14 14 14 14 10 10 10 14 10 11 11 11 10 10 11 10
 [146] 10 14 10 12 14 12 12 14 10 11 12 14 10 12 12 21 10 11 11 17 10 11 11 15 10 11 19 22 23
 [175] 23 11 16 12 13 13 13 13 13 13 11 11 11 10 10 10 11 11 11 10 12 14 10 21 14 10 29 14 28
 [204] 19 10 10 19 19 13 13 13 13 13 13 10 14 10 11 14 10 28 28 14 12 12 28 28 42 42 20 26 26
 [233] 12 10 14 14 21 14 10 10 28 10 10 10 14 28 10 21 14 14 11 13 14 11 11 13 10 10 11 17 14
 [262] 17 10 14 14 10 10 14 10 14 14 33 14 14 14 14 12 14 10 10 14 10 10 11 10 10 14 10 10 11
 [291] 10 10 10 10 10 10 11 11 11 11 10 13 10 11 11 10 11 14 10 10 10 10 10 11 14 10 14 14 10
 [320] 13 14 14 10 11 10 14 14 10 11 14 14 14 10 10 10 10 10 10 10 10 11 14 14 10 14 10 10 14
 [349] 10 10 11 10 11 14 14 10 10 11 14 14 14 14 14 11 14 14 14 14 14 14 12 10 10 12 10 10 10
 [378] 10 10 14 15 10 14 14 14 14 10 10 10 14 15 11 13 10 10 10 11 14 10 12 12 12 12 12 14 14
 [407] 10 10 12 14 10 14 14 12 10 10 14 14 18 11 12 12 12 14 10 10 12 10 11 11 13 12 12 12 13
 [436] 10 14 19 11 14 16 14 10 14 14 10 12 10 10 14 14 14 14 10 10 10 10 14 14 14 14 10 10 14
 [465] 14 10 12 13 10 12 14 10 10 11 11 14 10 14 14 11 11 10 10 10 14 11 12 12 12 14 10 10 10
 [494] 14 14 11 11 12 14 14 14 14 14 14 10 10 10 10 10 10 10 11 14 14 10 12 14 14 10 14 14 10
 [523] 10 10 10 10 10 14 15 15 10 14 11 10 14 14 14 13 14 14 10 14 14 14 10 10 10 10 10 12 14
 [552] 14 14 21 21 10 11 10 10 14 10 10 10 14 10 10 15 30 30 10 10 14 18 28 18 18 11 19 15 12
 [581] 13 10 10 12 12 14 10 10 14 56 14 16 14 12 13 12 29 28 14 11 28 11 14 21 28 14 28 10 26
 [610] 25 25 25 14 25 25 30 21 10 10 14 14 11 18 18 18 18 18 18 21 14 14 10 10 10 10 15 10
 [639] 10 10 14 11 11 11 14 12 12 10 14 18 10 12 14 10 10 10 10 11 11 14 10 10 10 14 14 10 10
 [668] 10 10 10 14 10 11 14 14 14 14 10 14 14 10 14 10 10 14 12 21 21 11 14 14 14 14 11 10 14
 [697] 14 10 13 14 14 27 14 10 10 10 10 14 10 14 21 10 11 14 10 10 10 10 14 10 10 10 10 11 14
 [726] 10 10 10 14 10 10 14 10 10 14 10 10 10 11 11 14 14 14 11 12 12 12 14 10 10 11 14 10 10
 [755] 10 10 14 10 10 10 14 14 12 10 10 11 12 14 14 10 11 10 10 11 10 10 13 11 14 16 16 13 12
 [784] 12 11 14 10 10 10 15 14 14 12 10 11 12 10 11 10 10 10 12 10 12 12 11 14 14 14 14 10 10
 [813] 10 10 10 14 10 11 11 10 10 11 11 10 14 10 10 14 10 10 10 10 10 11 10 13 10 11 14 14 14
 [842] 10 10 10 13 10 13 10 10 12 10 10 10 11 13 13 19 10 10 14 10 10 10 11 10 11 10 11 10 10
 [871] 14 13 12 10 14 12 15 16 13 10 11 10 10 60 69 30 10 11 15 14 13 14 14 13 13 13 10 11 14
 [900] 13 15 10 14 14 14 14 21 10 13 12 14 14 10 10 14 14 22 21 28 28 10 13 13 10 14 14 10 16
 [929] 10 10 10 10 11 10 12 10 11 13 14 10 14 10 10 14 14 15 14 30 11 11 11 11 10 11 14 14 11
 [958] 11 11 15 14 14 10 10 14 10 14 10 10 10 10 10 13 10 14 10 14 10 14 12 10 11 10 10 14 10
 [987] 14 10 10 10 18 10 10 10 10 10 14 14 12 12
 [ reached 'max' / getOption("max.print") -- omitted 2008 entries ]
```

Ahora continuaremos con los categóricos y para eso haremos grafico de barras simples para ver cómo se distribuyen

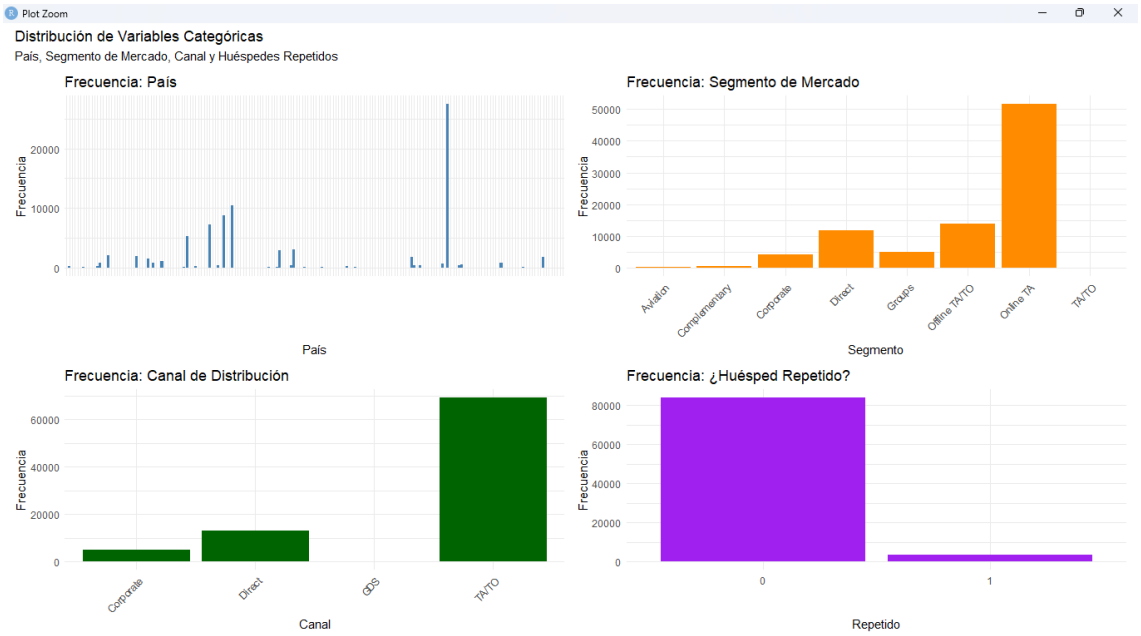
```r
#Gráfico para 'hotel'

p_hotel <- ggplot(df, aes(x = hotel)) +
  geom_bar(fill = "lightskyblue3") +
  labs(title = "Frecuencia: Hotel", x = "Tipo de hotel", y =
"Frecuencia") +
  theme_minimal()


#Gráfico para 'is_canceled'

p_cancel <- ggplot(df, aes(x = factor(is_canceled))) +
  geom_bar(fill = "lightcoral") +
  labs(title = "Frecuencia: Cancelaciones", x =
"¿Cancelado?", y = "Frecuencia") +
  theme_minimal()


#Gráfico para 'meal'

p_meal <- ggplot(df, aes(x = meal)) +
  geom_bar(fill = "mediumseagreen") +
  labs(title = "Frecuencia: Meal", x = "Tipo de comida", y =
"Frecuencia") +
  theme_minimal()


#Combinar los 3 en d6

d6 <- (p_hotel | p_cancel | p_meal) +
  plot_annotation(
    title = "Distribución de Variables Categóricas",
    subtitle = "Hotel, Cancelaciones y Tipos de Comida",

  )


#Mostrar d6

d6
```



Notamos que en “Meal” hay una categoría que es muy mínima, pero en “Meal” solo hay 4 categorías y no hace mucho ruido por lo que no será necesario eliminarlo. Los demás no tienen valores atípicos.

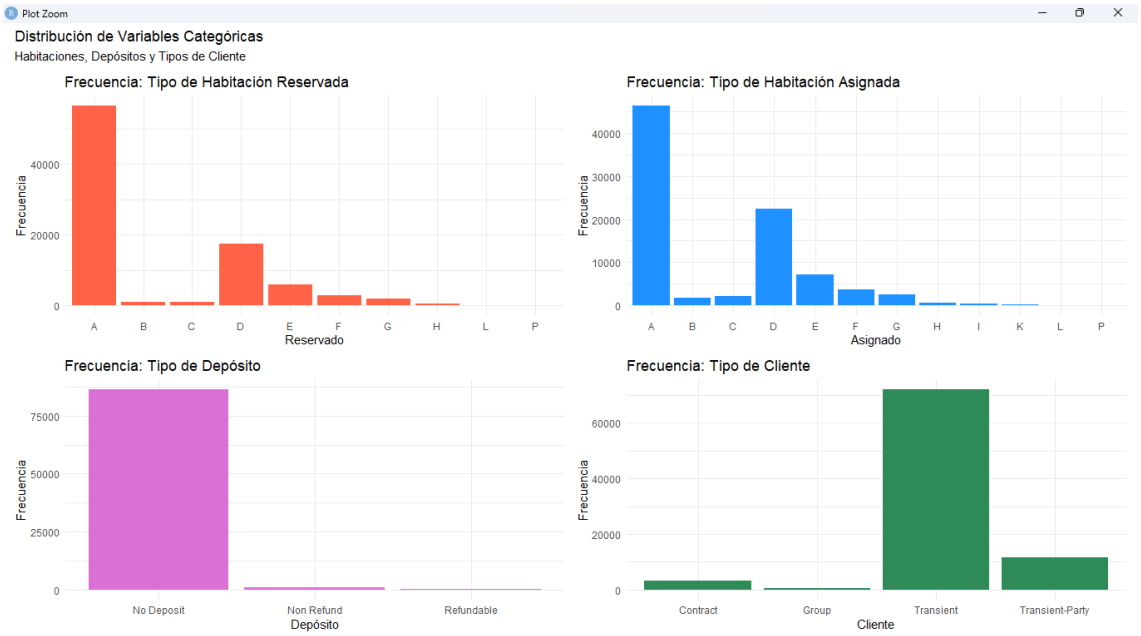
```
#Gráfico para 'country'
p_country <- ggplot(df, aes(x = country)) +
  geom_bar(fill = "steelblue") +
  labs(title = "Frecuencia: País", x = "País", y =
"Frecuencia") +
  theme_minimal() +
  theme(axis.text.x = element_blank())  # Oculta las
etiquetas si hay muchas

#Gráfico para 'market_segment'
p_market <- ggplot(df, aes(x = market_segment)) +
  geom_bar(fill = "darkorange") +
  labs(title = "Frecuencia: Segmento de Mercado", x =
"Segmento", y = "Frecuencia") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

#Gráfico para 'distribution_channel'
p_channel <- ggplot(df, aes(x = distribution_channel)) +
  geom_bar(fill = "darkgreen") +
  labs(title = "Frecuencia: Canal de Distribución", x =
"Canal", y = "Frecuencia") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```
#Gráfico para 'is_repeated_guest'
p_repeat <- ggplot(df, aes(x = factor(is_repeated_guest))) +
  geom_bar(fill = "purple") +
  labs(title = "Frecuencia: ¿Huésped Repetido?", x =
"Repetido", y = "Frecuencia") +
  theme_minimal()

#Combinar los 4 en d7
d7 <- (p_country | p_market) / (p_channel | p_repeat) +
  plot_annotation(
    title = "Distribución de Variables Categóricas",
    subtitle = "País, Segmento de Mercado, Canal y Huéspedes
Repetidos"
  )

#Mostrar d7
d7
```



Según el grafico notamos que en “country” hay demasiado ruido por lo que vamos a tener que agrupar los datos menores a un 2% o un valor similar en otra categoría “other” para que se vea mejor el grafico y se entienda, lo mismo como “market segment”. En Canal de distribución no será necesario porque solo son 4 datos y no hay mucho ruido y lo mismo en frecuencia de un huésped.

```r
#Gráfico para 'reserved_room_type'

p_reserved <- ggplot(df, aes(x = reserved_room_type)) +
  geom_bar(fill = "tomato") +
  labs(title = "Frecuencia: Tipo de Habitación Reservada", x
= "Reservado", y = "Frecuencia") +
  theme_minimal()


#Gráfico para 'assigned_room_type'

p_assigned <- ggplot(df, aes(x = assigned_room_type)) +
  geom_bar(fill = "dodgerblue") +
  labs(title = "Frecuencia: Tipo de Habitación Asignada", x =
"Asignado", y = "Frecuencia") +
  theme_minimal()


#Gráfico para 'deposit_type'

p_deposit <- ggplot(df, aes(x = deposit_type)) +
  geom_bar(fill = "orchid") +
  labs(title = "Frecuencia: Tipo de Depósito", x =
"Depósito", y = "Frecuencia") +
  theme_minimal()


#Gráfico para 'customer_type'

p_customer <- ggplot(df, aes(x = customer_type)) +
  geom_bar(fill = "seagreen") +
  labs(title = "Frecuencia: Tipo de Cliente", x = "Cliente",
y = "Frecuencia") +
  theme_minimal()


#Combinar los 4 en d8

d8 <- (p_reserved | p_assigned) / (p_deposit | p_customer) +
  plot_annotation(
    title = "Distribución de Variables Categóricas",
    subtitle = "Habitaciones, Depósitos y Tipos de Cliente"
  )


#Mostrar d8

d8
```

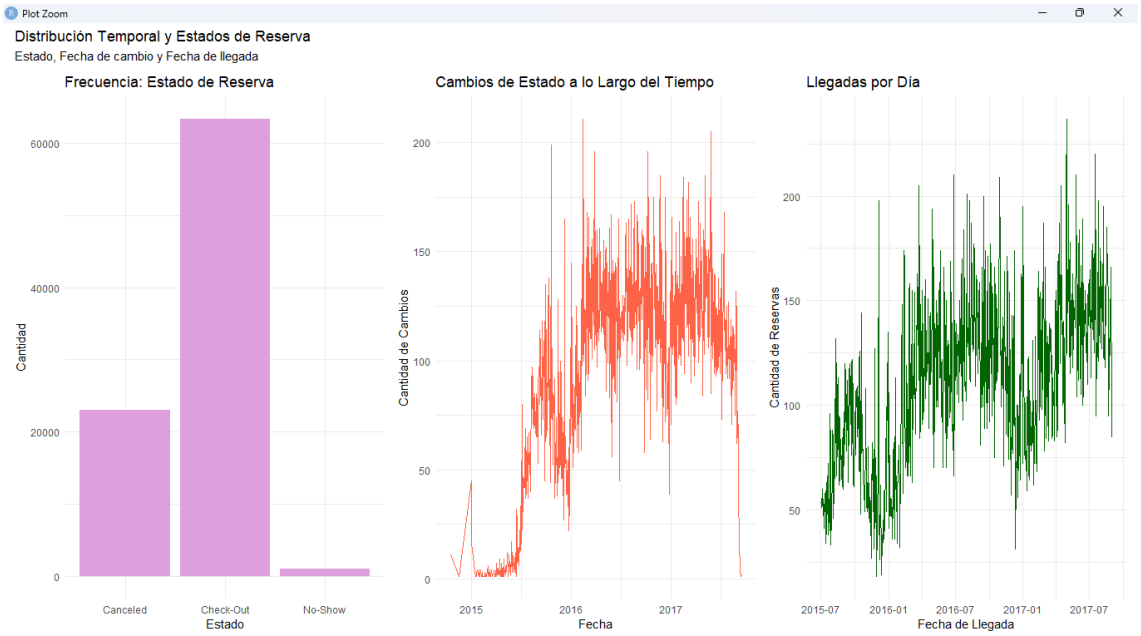


Notamos gracias a los gráficos que en tipo de habitación reservada hay muchos datos menores a un 2% o similares en comparación con los otros por lo que también para eliminar ese ruido, esos valores atípicos vamos a ponerlo en “other”, lo mismo con tipo de habitación asignada.

Pero para la Frecuencia de tipo de depósito no es necesario porque solo hay 3 categorías y lo mismo con frecuencia del tipo de cliente donde son 4 categorías.

```
#Gráfico 1: Estado de reserva
p_status <- ggplot(df, aes(x = reservation_status)) +
  geom_bar(fill = "plum") +
  labs(title = "Frecuencia: Estado de Reserva", x = "Estado",
y = "Cantidad") +
  theme_minimal()

#Gráfico 2: Fechas de cambio de estado
p_status_date <- df %>%
  count(reservation_status_date) %>%
  ggplot(aes(x = reservation_status_date, y = n)) +
  geom_line(color = "tomato") +
  labs(title = "Cambios de Estado a lo Largo del Tiempo", x =
"Fecha", y = "Cantidad de Cambios") +
  theme_minimal()

#Gráfico 3: Fechas de llegada
p_arrival <- df %>%
  count(arrival_date) %>%
```

```
  ggplot(aes(x = arrival_date, y = n)) +
  geom_line(color = "darkgreen") +
  labs(title = "Llegadas por Día", x = "Fecha de Llegada", y
= "Cantidad de Reservas") +
  theme_minimal()

#Combinar los tres gráficos
d9 <- (p_status | p_status_date | p_arrival) +
  plot_annotation(
    title = "Distribución Temporal y Estados de Reserva",
    subtitle = "Estado, Fecha de cambio y Fecha de llegada"
  )

#Mostrar todo junto
d9
```



Estos gráficos son para ver cómo se comportan las fechas y la frecuencia del Estado de Reserva. Como notamos en la frecuencia del estado de reserva no será necesario reemplazar algo porque solo hay 3 categorías por lo que no hay mucho ruido y ese dato puede ser importante.

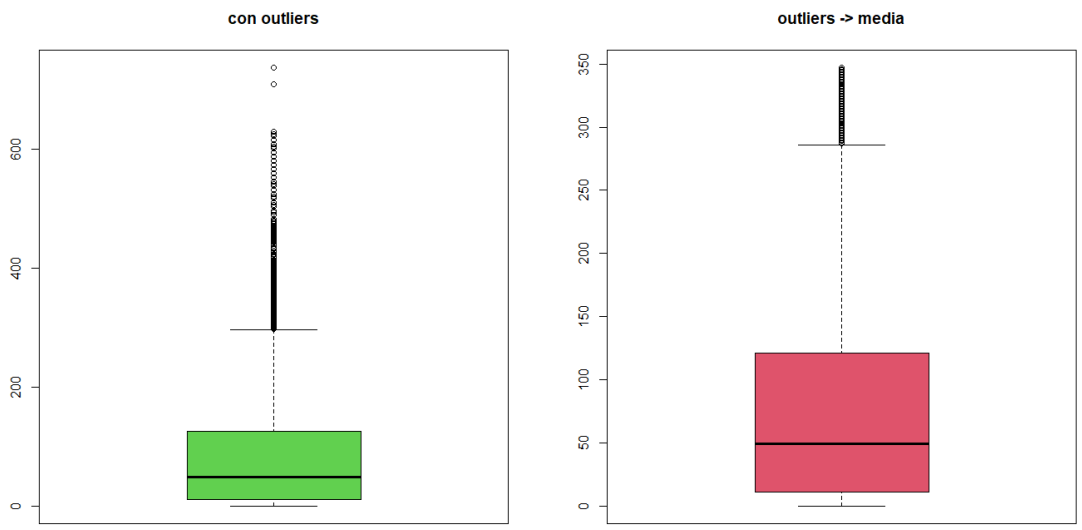
Tratamiento de "Outliers":

```
#Eliminación de atipicos
df_clean<-df
```

```
num_cols <- sapply(df_clean, is.numeric)
```

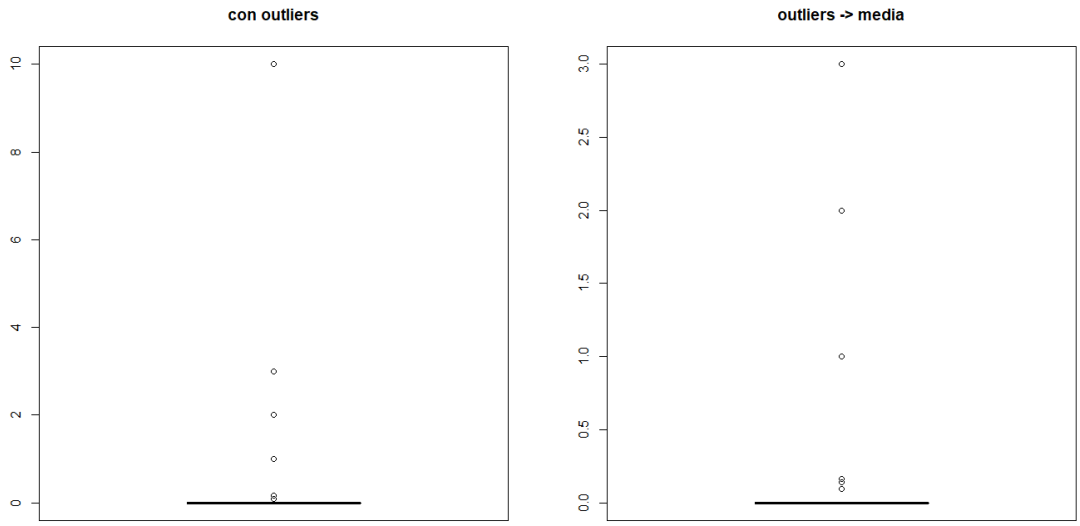
Según los análisis pasados vamos a modificar los valores atípicos.

```
#Lista de variables numéricas
#Vamos a comenzar con lead_time
D1

#Definir límites del 1% y 99%
lower_bound <- quantile(df$lead_time, 0.00)
upper_bound <- quantile(df$lead_time, 0.99)

#Calcular media del lead_time (solo dentro del rango
"normal")
media <- mean(df$lead_time[df$lead_time >= lower_bound &
df$lead_time <= upper_bound], na.rm = TRUE)

#Reemplazar outliers por la media
df_clean$lead_time <- ifelse(df$lead_time < lower_bound |
df$lead_time > upper_bound,
                             media,
                             df$lead_time)
#Visualizar antes y después
par(mfrow = c(1, 2))
boxplot(df$lead_time, main = "con outliers", col = 3)
boxplot(df_clean$lead_time, main = "outliers -> media", col =
2)
```



Ahora vamos a modificar los valores atípicos de “children”.

```
d2
#Definir límites del 1% y 99%
upper_bound <- quantile(df$children, 0.9999, na.rm = TRUE)

#Calcular la media solo de los valores dentro del rango
aceptable
media <- mean(df$children[df$children <= upper_bound], na.rm
= TRUE)

#Reemplazar solo los valores mayores al límite por la media
df_clean$children <- ifelse(df$children > upper_bound, media,
df$children)

#Visualizar antes y después
par(mfrow = c(1, 2))
boxplot(df$children, main = "con outliers", col = 3)
boxplot(df_clean$children, main = "outliers -> media", col =
2)
```



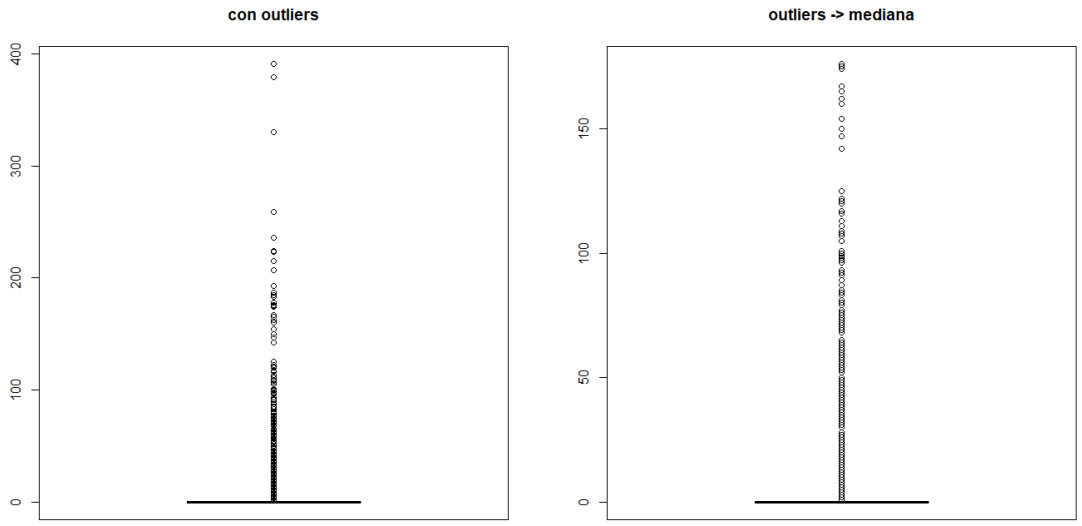

```
#Ver resumen
summary(df$children)
summary(df_clean$children)
```

```
> summary(df$children)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.0000  0.0000  0.0000  0.1386  0.0000 10.0000
> summary(df_clean$children)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.0000  0.0000  0.0000  0.1385  0.0000  3.0000
>
```

Ahora vamos con “days_in_waiting_list”

```
#Definir límites del 1% y 99%
#Definir el límite superior (percentil 99)
upper_bound <- quantile(df$days_in_waiting_list, 0.999, na.rm
= TRUE)

#Calcular la mediana de los valores dentro del rango
aceptable
mediana <-
median(df$days_in_waiting_list[df$days_in_waiting_list <=
upper_bound], na.rm = TRUE)
```

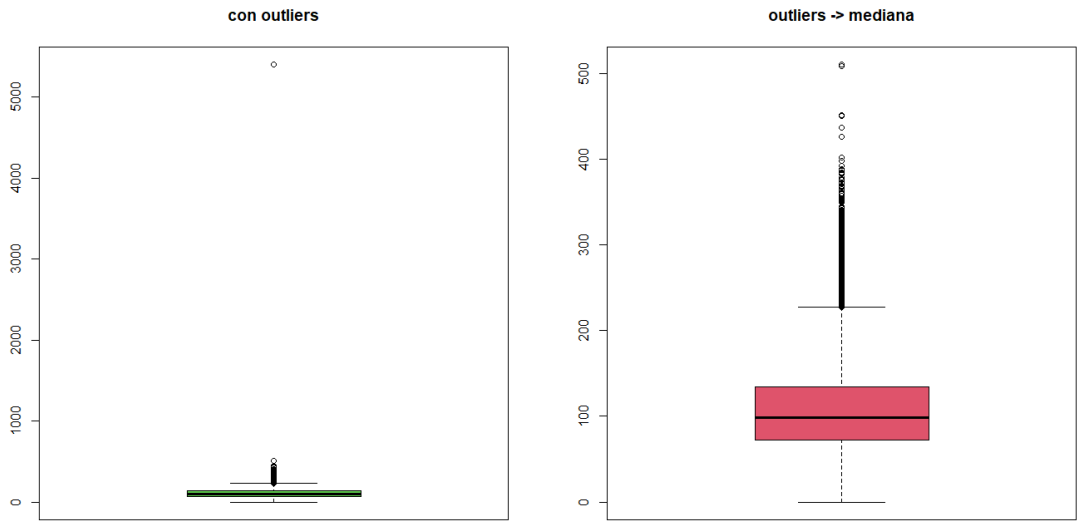
```r
#Reemplazar valores mayores al límite por la mediana
df_clean$days_in_waiting_list <-
ifelse(df$days_in_waiting_list > upper_bound, mediana,
df$days_in_waiting_list)

#Visualizar antes y después
par(mfrow = c(1, 2))
boxplot(df$days_in_waiting_list, main = "con outliers", col =
3)
boxplot(df_clean$days_in_waiting_list, main = "outliers ->
mediana", col = 2)
```




```r
#Ver resumen
summary(df$days_in_waiting_list)
summary(df_clean$days_in_waiting_list)
```

```
> summary(df$days_in_waiting_list)
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
  0.0000   0.0000   0.0000   0.7496   0.0000 391.0000
> summary(df_clean$days_in_waiting_list)
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
  0.0000   0.0000   0.0000   0.5391   0.0000 176.0000
>
```

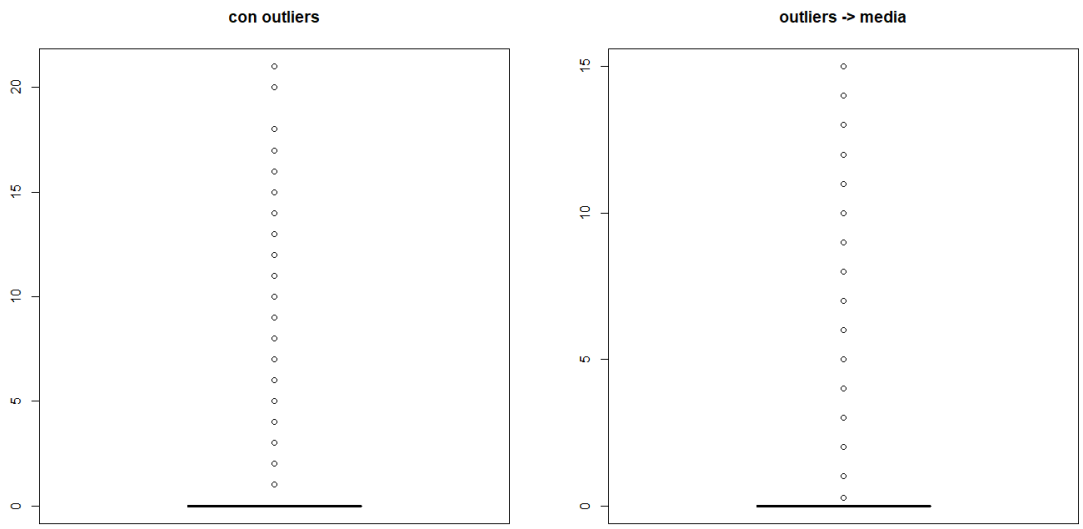
```r
#Ahora vamos con ADR
#Definir el límite superior (percentil 99)
upper_bound <- quantile(df$adr, 0.99999, na.rm = TRUE)

#Calcular la mediana de los valores dentro del rango
aceptable
mediana <- median(df$adr[df$adr <= upper_bound], na.rm =
TRUE)

#Reemplazar valores mayores al límite por la mediana
df_clean$adr <- ifelse(df$adr > upper_bound, mediana, df$adr)

#Visualizar antes y después
par(mfrow = c(1, 2))
boxplot(df$adr, main = "con outliers", col = 3)
boxplot(df_clean$adr, main = "outliers -> mediana", col = 2)
```




```r
#Ver resumen
summary(df$adr)
summary(df_clean$adr)
```

```
> summary(df$adr)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    0.0    72.0    98.1   106.3   134.0  5400.0
> summary(df_clean$adr)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    0.0    72.0    98.1   106.3   134.0   510.0
>
```

```
#Ahora vamos por booking changes
summary(df$booking_changes)

#Definir el límite superior (percentil 99)
upper_bound <- quantile(df$booking_changes, 0.9999, na.rm =
TRUE)

#Calcular la media de los valores dentro del rango aceptable
media <- mean(df$booking_changes[df$booking_changes <=
upper_bound], na.rm = TRUE)

#Reemplazar los valores mayores al límite por la media
df_clean$booking_changes <- ifelse(df$booking_changes >
upper_bound, media, df$booking_changes)

#Visualizar antes y después
par(mfrow = c(1, 2))
boxplot(df$booking_changes, main = "con outliers", col = 3)
boxplot(df_clean$booking_changes, main = "outliers -> media",
col = 2)
```

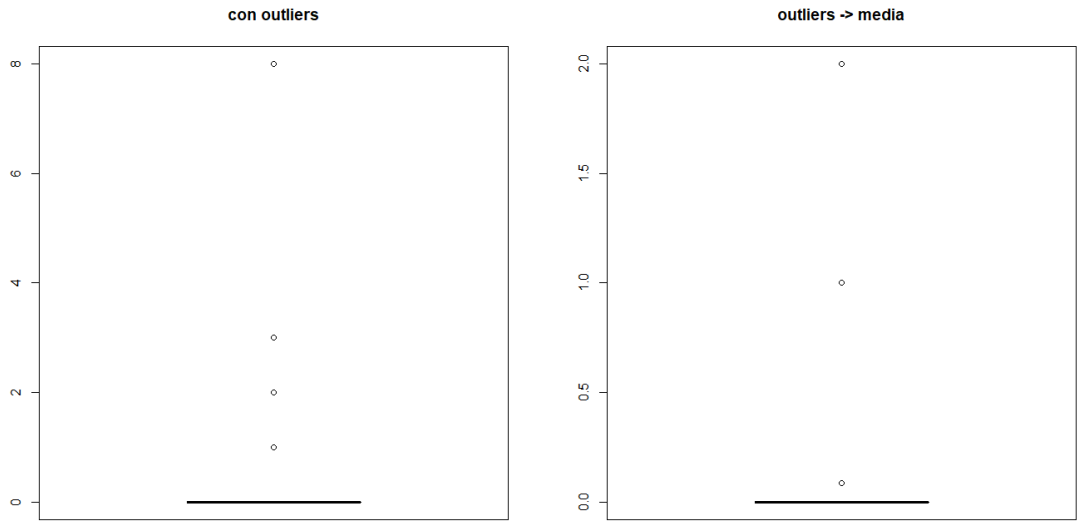



```
#Ver resumen
summary(df$booking_changes)
summary(df_clean$booking_changes)
```

```
> summary(df$booking_changes)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.0000  0.0000  0.0000  0.2716  0.0000 21.0000
> summary(df_clean$booking_changes)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.0000  0.0000  0.0000  0.2702  0.0000 15.0000
```

```
#Ahora con required_car_parking_spaces
summary(df$booking_changes)

#Definir el límite superior (percentil 99)
upper_bound <- quantile(df$required_car_parking_spaces,
0.9999, na.rm = TRUE)

#Calcular la media de los valores dentro del rango aceptable
media <-
mean(df$required_car_parking_spaces[df$required_car_parking_s
paces <= upper_bound], na.rm = TRUE)

#Reemplazar los valores mayores al límite por la media
```

```
df_clean$required_car_parking_spaces <-
ifelse(df$required_car_parking_spaces > upper_bound,
                                                media,

df$required_car_parking_spaces)

#Visualizar antes y después
par(mfrow = c(1, 2))
boxplot(df$required_car_parking_spaces, main = "con
outliers", col = 3)
boxplot(df_clean$required_car_parking_spaces, main =
"outliers -> media", col = 2)
```

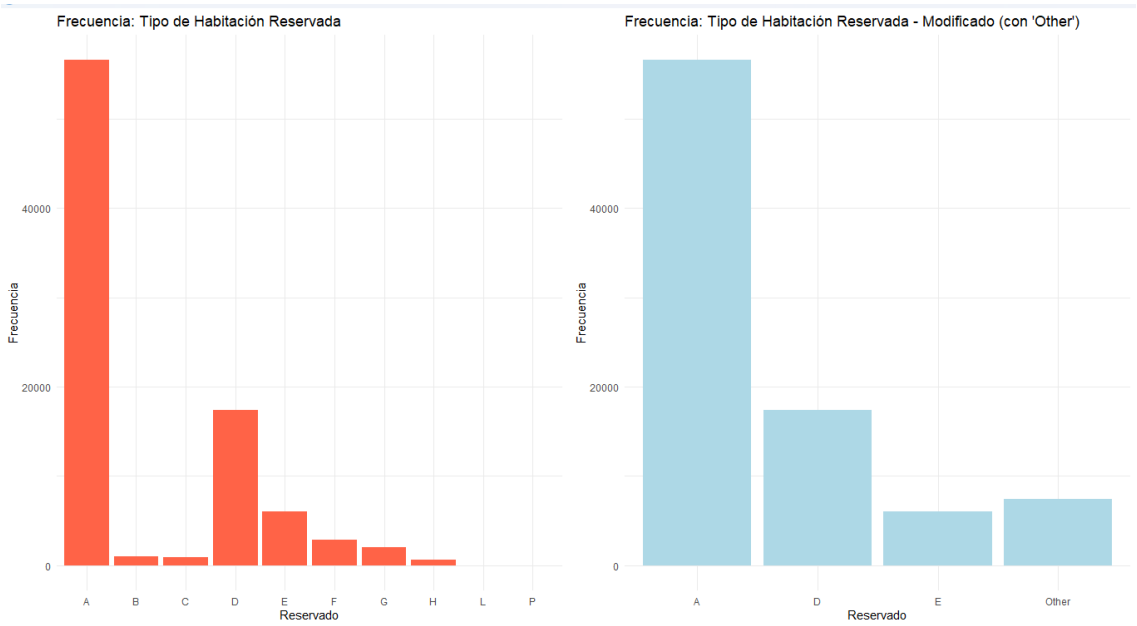



```
#Ver resumen
summary(df$required_car_parking_spaces)
summary(df_clean$required_car_parking_spaces)
```

```
> summary(df$required_car_parking_spaces)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.00000 0.00000 0.00000 0.08423 0.00000 8.00000
> summary(df_clean$required_car_parking_spaces)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.00000 0.00000 0.00000 0.08394 0.00000 2.00000
```

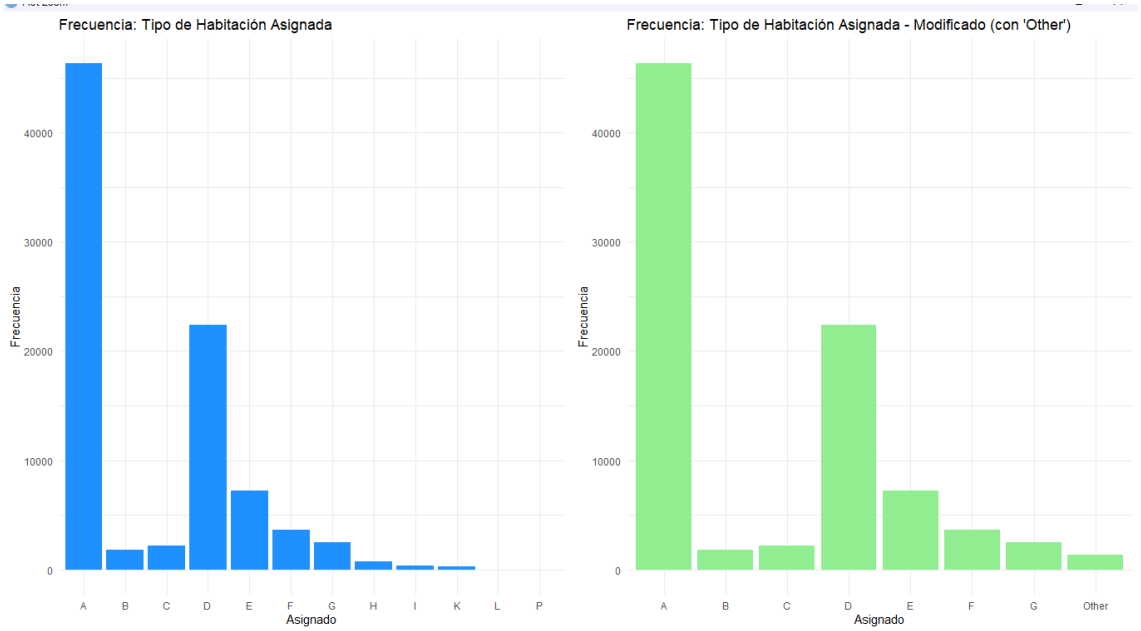
```r
#Ahora con los categoricos
# Definir el umbral (porcentaje)
umbral <- 0.04  # Esto significa el 1%

#Calcular la frecuencia de cada categoría
categoria_count <- table(df$reserved_room_type)

#Calcular el porcentaje de cada categoría
categoria_percent <- prop.table(categoria_count)

#Crear un vector lógico donde las categorías que tengan un
porcentaje menor al umbral se agruparán en "Other"
df_clean$reserved_room_type <- df$reserved_room_type

#Añadir el nivel "Other" a los factores de la columna
levels(df_clean$reserved_room_type) <-
c(levels(df_clean$reserved_room_type), "Other")

#Reemplazar las categorías menos frecuentes por "Other"
df_clean$reserved_room_type[df_clean$reserved_room_type %in%
names(categoria_percent[categoria_percent < umbral])] <-
"Other"

#Ver el resumen de la columna original y la nueva
summary(df$reserved_room_type)
summary(df_clean$reserved_room_type)

#Ver la distribución de categorías
table(df_clean$reserved_room_type)

#Gráfico para la columna modificada (con "Other")
p_reserved_clean <- ggplot(df_clean, aes(x =
reserved_room_type)) +
```

```
  geom_bar(fill = "lightblue") +
  labs(title = "Frecuencia: Tipo de Habitación Reservada -
Modificado (con 'Other')",
       x = "Reservado", y = "Frecuencia") +
  theme_minimal()

#Usamos gridExtra para mostrar ambos gráficos en una sola
fila
grid.arrange(p_reserved, p_reserved_clean, ncol = 2)
```



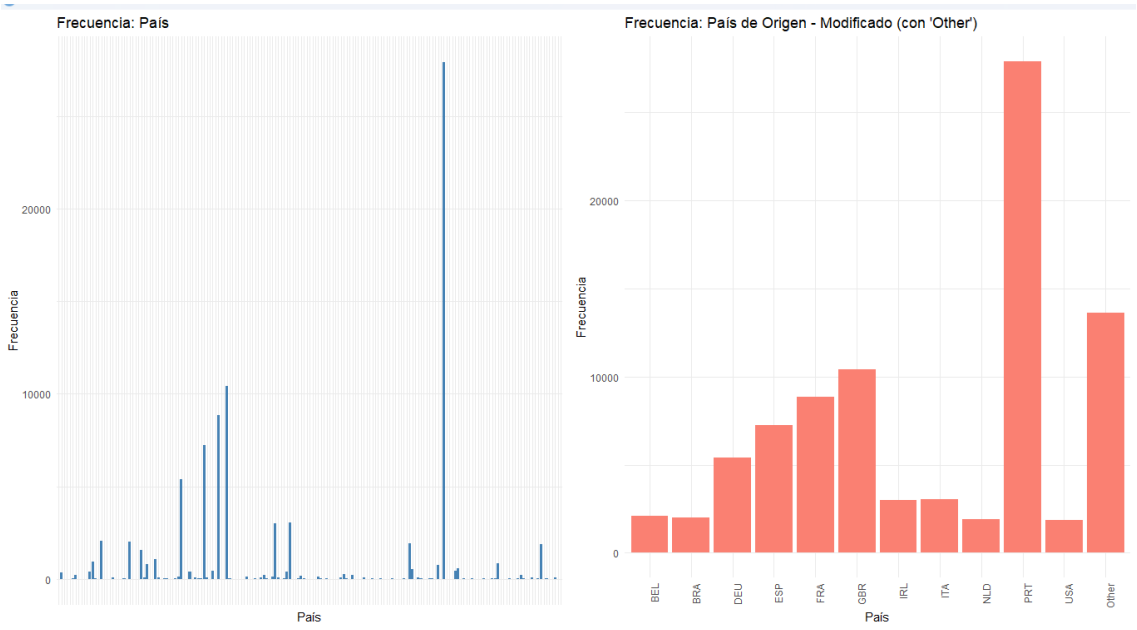

```
#Ahora con los categóricos: assigned_room_type

#Definir el umbral (porcentaje)
umbral <- 0.02  # Esto significa el 4%

#Calcular la frecuencia de cada categoría
categoria_count_assigned <- table(df$assigned_room_type)

#Calcular el porcentaje de cada categoría
categoria_percent_assigned <-
prop.table(categoria_count_assigned)

#Copiar columna original a df_clean
```

```
df_clean$assigned_room_type <- df$assigned_room_type

#Añadir el nivel "Other" a los factores de la columna
levels(df_clean$assigned_room_type) <-
c(levels(df_clean$assigned_room_type), "Other")

#Reemplazar las categorías menos frecuentes por "Other"
df_clean$assigned_room_type[df_clean$assigned_room_type %in%
names(categoria_percent_assigned[categoria_percent_assigned <
umbral])] <- "Other"

#Ver resumen y distribución
summary(df$assigned_room_type)
summary(df_clean$assigned_room_type)
table(df_clean$assigned_room_type)

#Gráfico para la columna modificada (con "Other")
p_assigned_clean <- ggplot(df_clean, aes(x =
assigned_room_type)) +
  geom_bar(fill = "lightgreen") +
  labs(title = "Frecuencia: Tipo de Habitación Asignada -
Modificado (con 'Other')",
       x = "Asignado", y = "Frecuencia") +
  theme_minimal()

#Mostrar el gráfico original y el modificado (si tienes
p_assigned)
 grid.arrange(p_assigned, p_assigned_clean, ncol = 2)
```




```
#Ahora con los categóricos: country

#Definir el umbral (porcentaje)
 umbral <- 0.02  # Esto significa el 4%

#Calcular la frecuencia de cada país
 country_count <- table(df$country)

#Calcular el porcentaje de cada país
 country_percent <- prop.table(country_count)

#Copiar columna original a df_clean
 df_clean$country <- df$country

 #Añadir el nivel "Other" a los factores de la columna
 levels(df_clean$country) <- c(levels(df_clean$country),
"Other")

 #Reemplazar los países con porcentaje menor al umbral por
"Other"
 df_clean$country[df_clean$country %in%
names(country_percent[country_percent < umbral])] <- "Other"
```

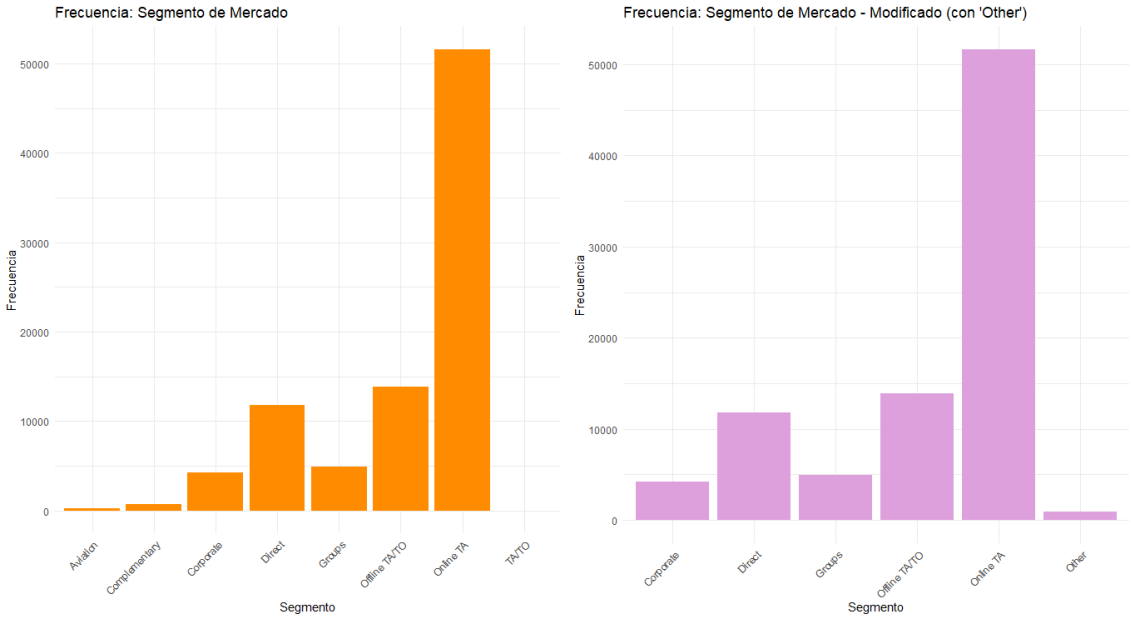
```
#Ver resumen y distribución
summary(df$country)
summary(df_clean$country)
table(df_clean$country)

#Gráfico para la columna modificada (con "Other")
p_country_clean <- ggplot(df_clean, aes(x = country)) +
  geom_bar(fill = "salmon") +
  labs(title = "Frecuencia: País de Origen - Modificado (con
'Other')",
       x = "País", y = "Frecuencia") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))

#Mostrar el gráfico limpio
grid.arrange(p_country, p_country_clean, ncol = 2)
```




```
#Ahora con los categóricos: market_segment

#Definir el umbral (porcentaje)
umbral <- 0.02 # Esto significa el 4%
```

```
 #Calcular la frecuencia de cada segmento
 segment_count <- table(df$market_segment)

 #Calcular el porcentaje de cada segmento
 segment_percent <- prop.table(segment_count)

 #Copiar columna original a df_clean
 df_clean$market_segment <- df$market_segment

 #Añadir el nivel "Other" a los factores de la columna
 levels(df_clean$market_segment) <-
c(levels(df_clean$market_segment), "Other")

 #Reemplazar los segmentos con porcentaje menor al umbral por
"Other"
 df_clean$market_segment[df_clean$market_segment %in%
names(segment_percent[segment_percent < umbral])] <- "Other"

 #Ver resumen y distribución
 summary(df$market_segment)
 summary(df_clean$market_segment)
 table(df_clean$market_segment)

 #Gráfico para la columna modificada (con "Other")
 p_market_clean <- ggplot(df_clean, aes(x = market_segment))
+
   geom_bar(fill = "plum") +
   labs(title = "Frecuencia: Segmento de Mercado - Modificado
(con 'Other')",
        x = "Segmento", y = "Frecuencia") +
   theme_minimal() +
   theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```
#Mostrar el gráfico limpio
grid.arrange(p_market, p_market_clean, ncol = 2)
```



Ahora tenemos dos dataset, uno con los datos limpios, pero sin modificar los valores atípicos que es "df" y otro donde los valores atípicos si están modificados y es "df_clean".

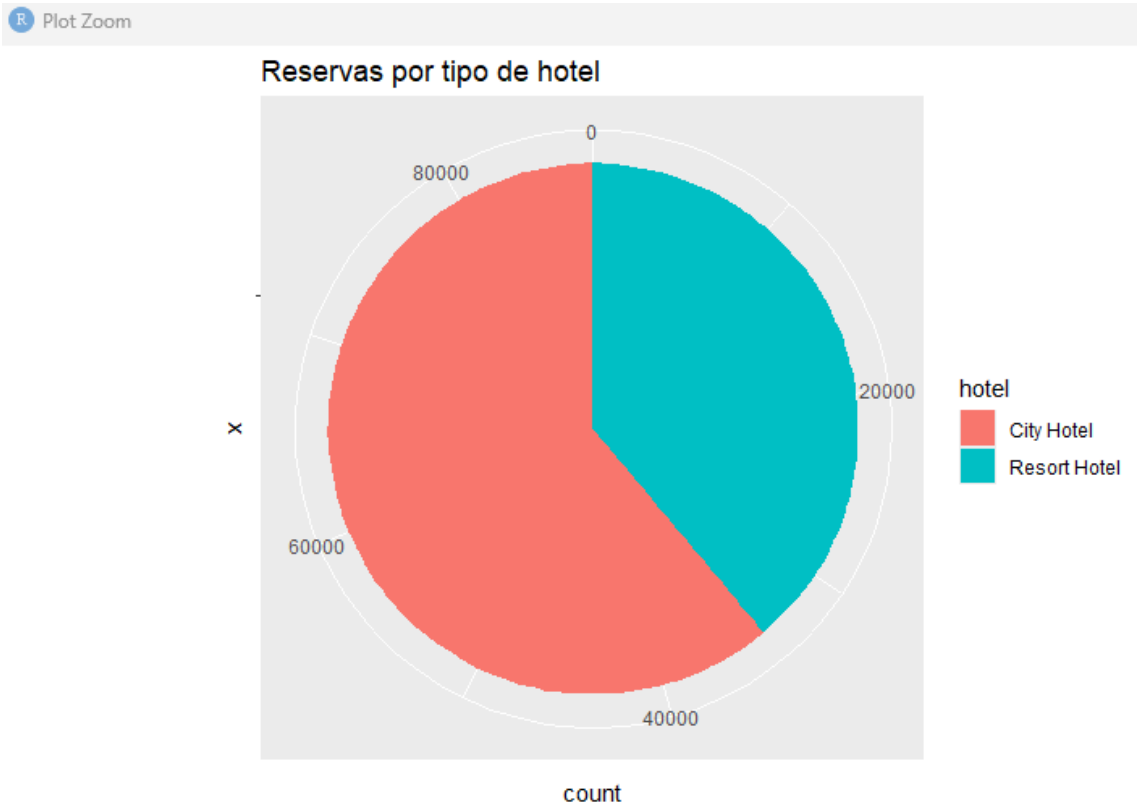
VISUALIZACIÓN DE DATOS

A continuación, se responderán las siguientes preguntas a través de visualizaciones basadas en los datos pre-procesados:

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

```
#Gráfico circular para el conteo de reservas por tipo de hotel
e1<-ggplot(df,aes(x="",fill=hotel))+
  geom_bar()+
  labs(title = "Reservas por tipo de hotel",
       )+
  coord_polar(theta="y")
```

e1



Se usó un gráfico circular para comparar que hotel tiene más y menos reservas. Además, se puede observar que las personas prefieren el Resort Hotel.

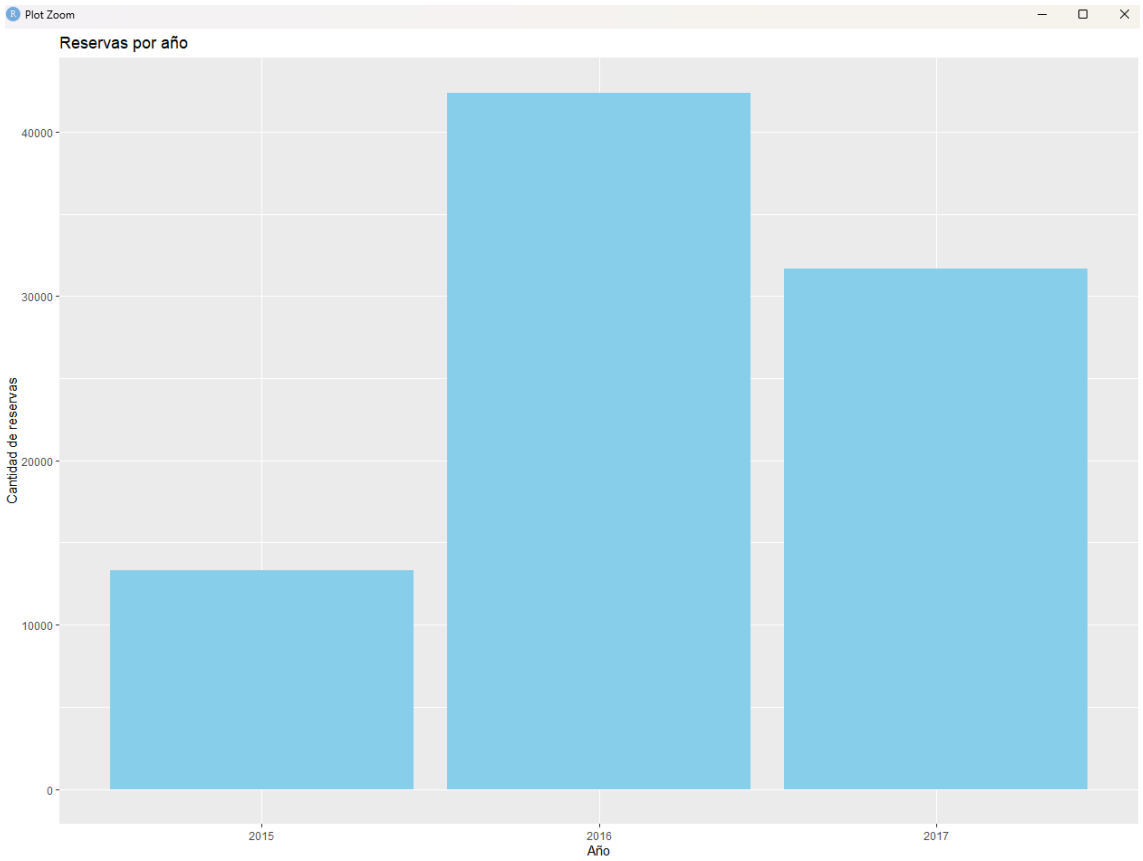
❖ ¿Está aumentando la demanda con el tiempo?

Para este caso y los siguientes separaremos los meses y años para mostrar un mejor gráfico

```
#Separación de meses y años
df$arrival_year <- format(df$arrival_date, "%Y")
df$arrival_month <- format(df$arrival_date, "%B")
```

Ahora sí, procederemos a responder las preguntas

```
#Gráfico de barras para el conteo de demandas por año
e2<-ggplot(df,aes(x=arrival_year))+
  geom_bar(fill="skyblue")+
  labs(title = "Reservas por año",x="Año",
       y="Cantidad de reservas") +
  theme_classic2() +
e2
```

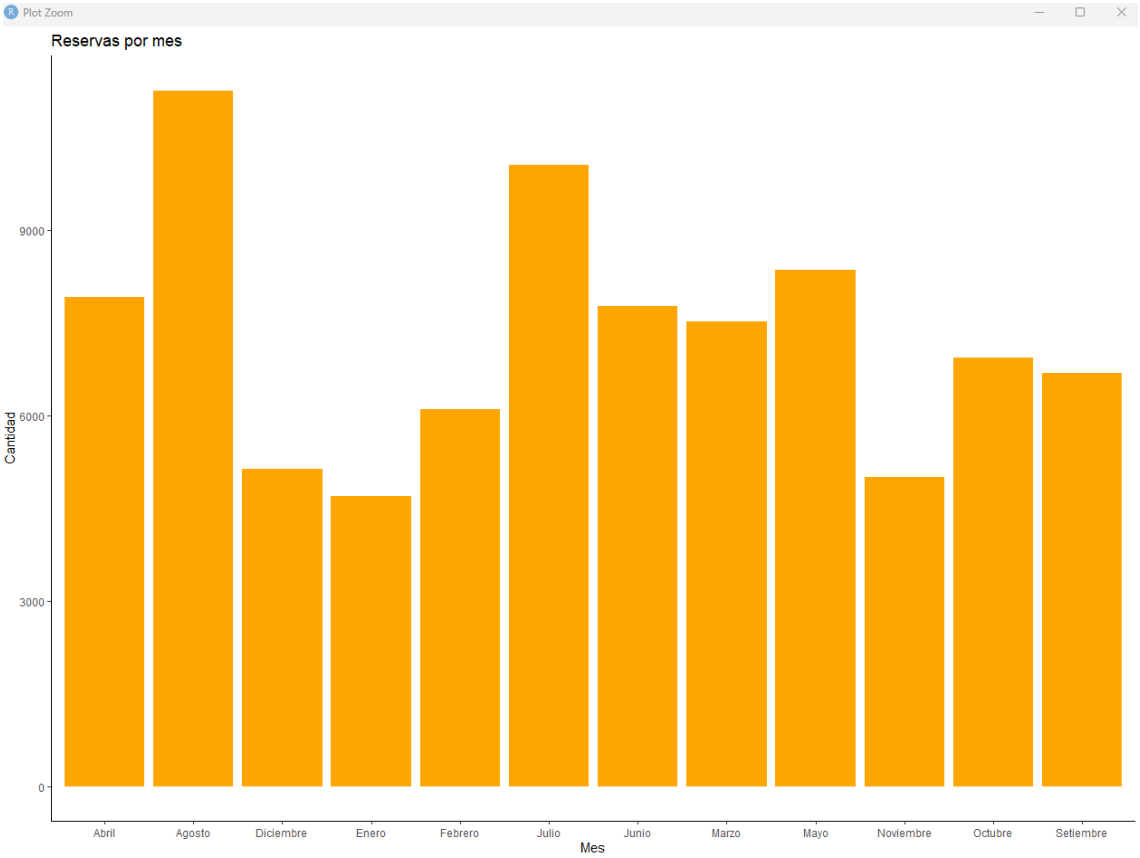


Podemos observar que hubo una caída de demandas del 2016 al 2017.

❖ ¿Cuáles son las temporadas de reservas (alta, media, baja)?

```
#Gráfico de barras para verificar cuales son las
temporadas de reservas
e3<-ggplot(df,aes(x =arrival_month)) +
  geom_bar(fill = "orange") +
  labs(title = "Reservas por mes", x = "Mes",
```

```
        y = "Cantidad") +
  theme_classic()
e3
```

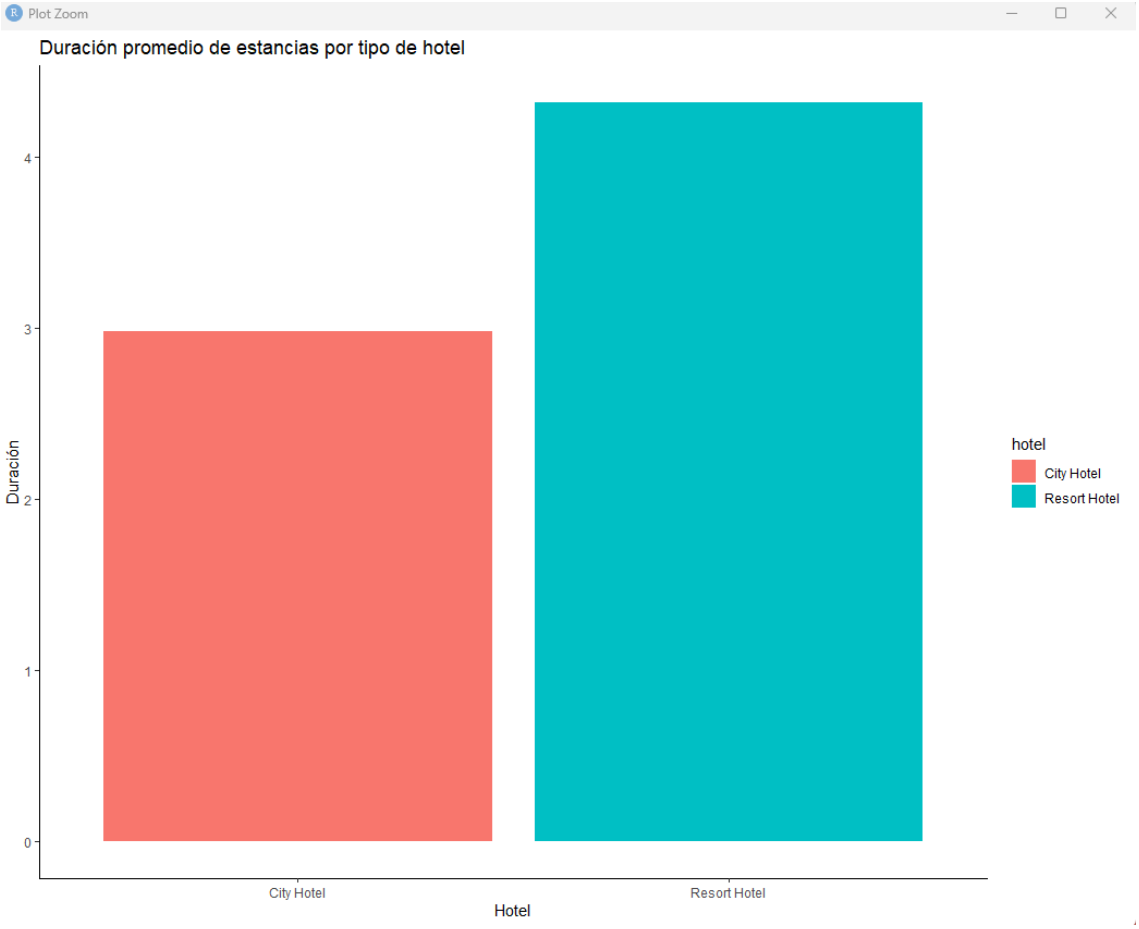


En el gráfico de barras realizado se puede visualizar que el mes que de agosto tiene más reservas, enero es el que tiene menos reservas y febrero es el que ni menos ni más reservas.

❖ ¿Cuál es la duración promedio de las estancias por tipo de hotel?

```
#Grafico de barras para calcular el promedio de las
estancias por tipo de hotel.
#1. Calcular duración de la estancia
df$duracion <- df$stays_in_week_nights +
df$stays_in_weekend_nights


#2. Calcular el promedio por tipo de hotel
promedio<- aggregate(duracion ~ hotel, data = df, FUN =
mean)
```

```
#3. Grafico de barras
e4 <- ggplot(promedio, aes(x = hotel, y = duracion, fill =
hotel)) +
  geom_bar(stat = "identity") + #usar los valores ya
calculados
  labs(title = "Duración promedio de estancias por tipo de
hotel", x = "Hotel", y = "Duración") +
  theme_classic()
e4
```

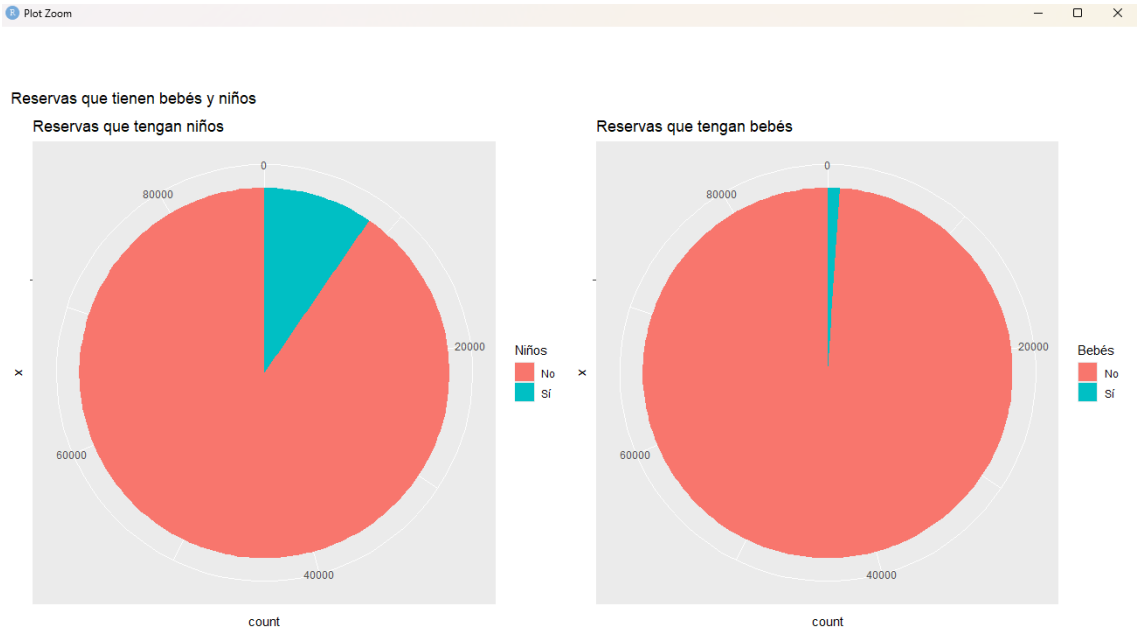


Se puede observar que por promedio las personas prefieren quedarse en el "Resort Hotel".

- ¿Cuántas reservas incluyen niños y/o bebés?

```
# Gráfico circular para saber cuántas reservas incluyen
niños
#1. Se crea una nueva columna para saber si las reservas
tienen niños o no
df$has_children<-ifelse(df$children > 0, "Sí", "No")

#2. Contar las reservas
conteo<-count(df, has_children)

#3. Realizar el gráfico
e5.1<-ggplot(df, aes(x ="",fill=has_children)) +
  geom_bar() +
  labs(title = "Reservas que tengan niños",
       fill="Niños")+
  coord_polar(theta = "y")
e5.1


# Gráfico circular para saber cuántas reservas incluyen
bebés
#1. Se crea una nueva columna para saber si las reservas
tienen bebés o no
df$has_babies<-ifelse(df$babies > 0, "Sí", "No")

#2. Contar las reservas
conteo<-count(df, has_babies)

#3. Realizar el gráfico
e5.2<-ggplot(df, aes(x ="",fill=has_babies)) +
  geom_bar() +
  labs(title = "Reservas que tengan bebés",
       fill="Bebés")+
  coord_polar(theta = "y")
```

```
e5.2


#Combinacion de ambos graficos
e5.3<-(e5.1 | e5.2) +
  plot_annotation(title = 'Reservas que tienen bebés y
niños',)
e5.3
```

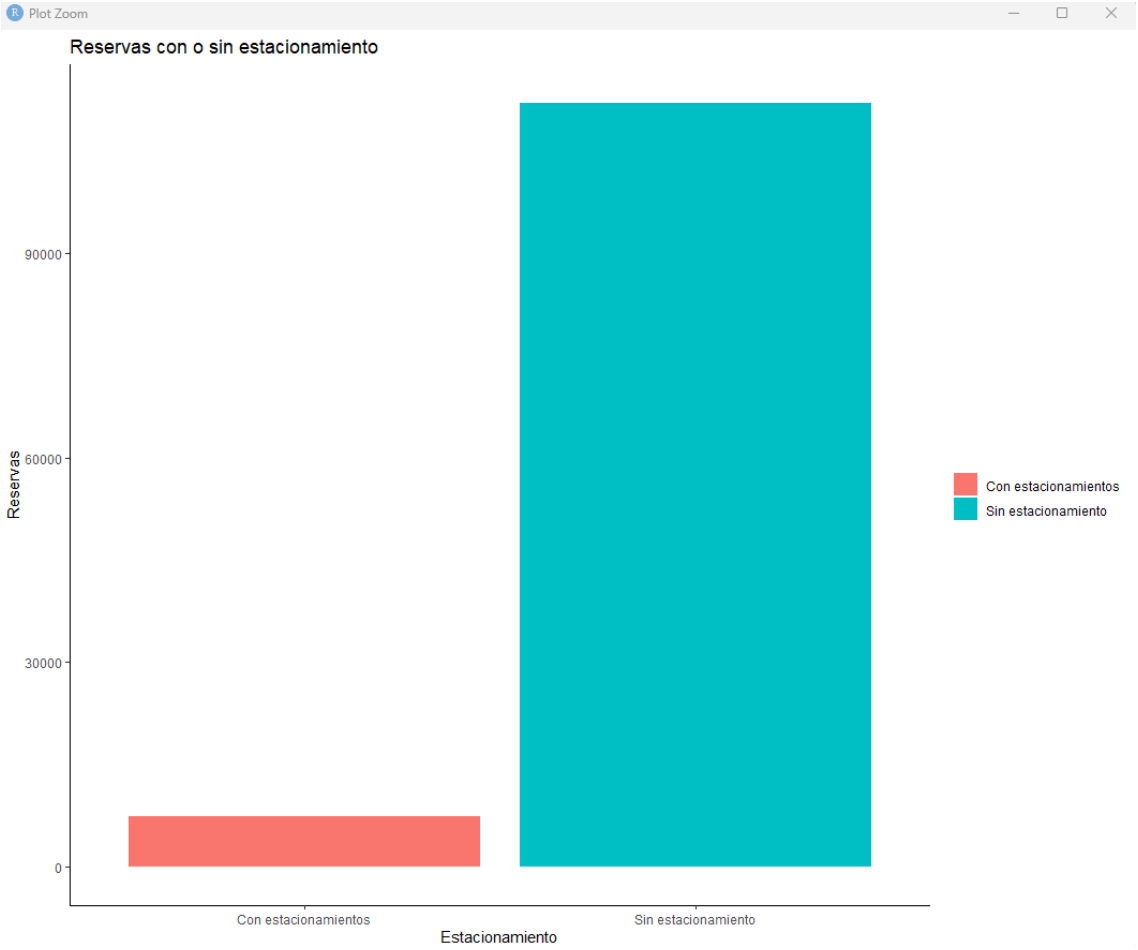


Según los gráficos la mayoría de las reservas no incluyen niños ni bebés.

❖ ¿Es importante contar con espacios de estacionamiento?

```
#Grafico de barras para saber cuántas reservas incluyen
estacionamiento
#1. Se crea una nueva columna
df$required_parking<-
ifelse(df$required_car_parking_spaces>0,"Con
estacionamientos","Sin estacionamiento")

#2. Contar los estacionamientos
```

```
conteo_estacionamiento<-table(df$required_parking)


#3. Graficar
 e6<-ggplot(df, aes(x = required_parking, fill =
required_parking)) +
 geom_bar() +
labs(title = "Reservas con o sin estacionamiento",
x = "Estacionamiento",
y = "Reservas", fill="") +
e6
```

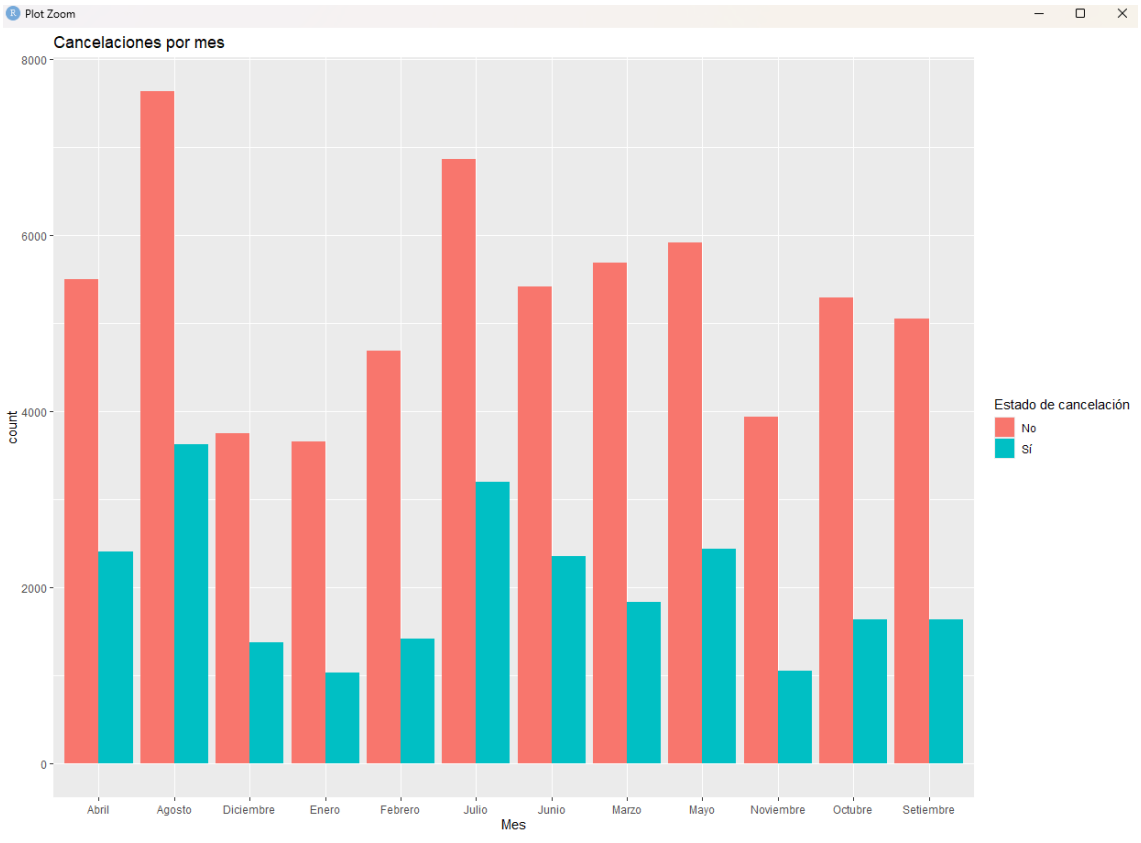


Según el gráfico, la mayoría de las reservas no necesitan un estacionamiento, así que no los estacionamientos no son tan necesarios.

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

Para que el gráfico se entienda mejor, hemos cambiado los 0 y 1 que se observan en la columna is_canceled, por sí y no:

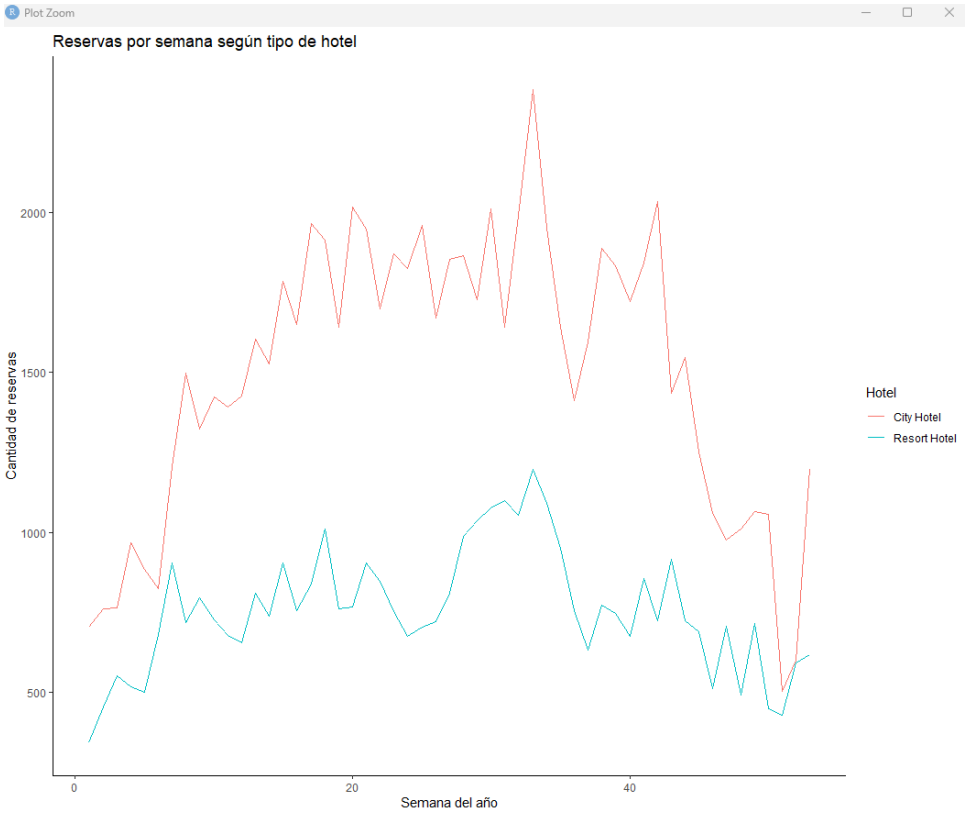
```
#Cambio de 0 y 1
df$is_canceled <- ifelse(df$is_canceled == 1, "Sí", "No")

#Gráfico de barras para verificar cuales son los meses en que hay mas cancelaciones
e7 <-ggplot(df,aes(x = arrival_month,fill= is_canceled
)) +
  geom_histogram() +
  labs(title = "Cancelaciones por mes", x=" Mes ",
fill="Estado de cancelación") +
  theme_get()
e7
```



En el gráfico de barras se puede visualizar que el mes en el que se producen más cancelaciones es el mes de agosto.

- Cantidad de reservas por semana por hotel

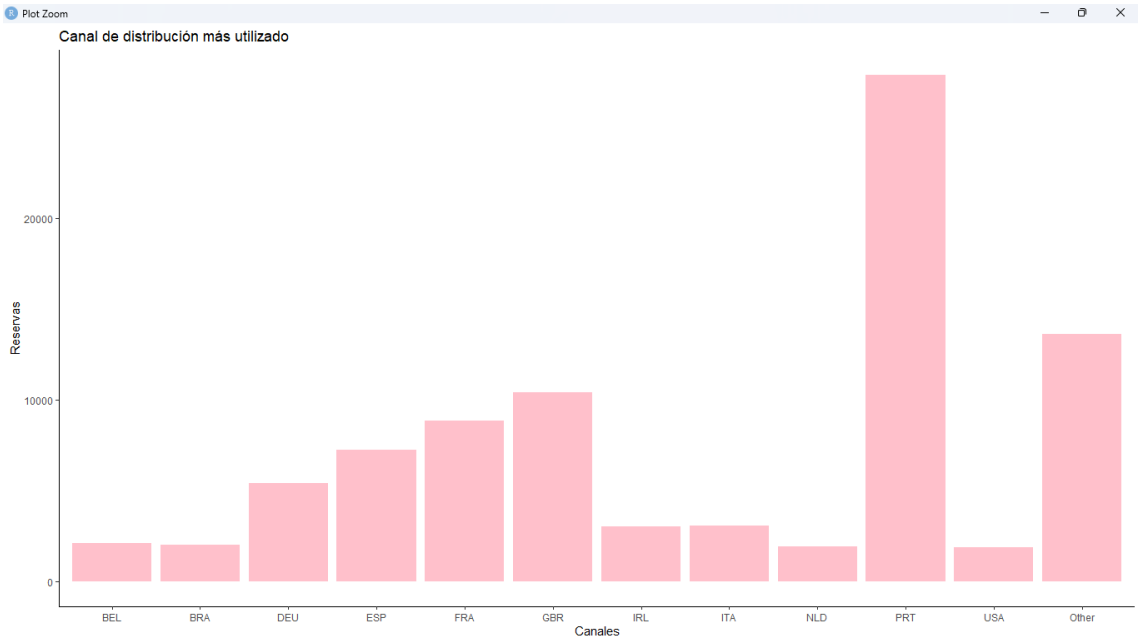
```
##Gráfico de líneas para la cantidad de reservas por semana por hotel

reservas_semanales <- count(df, arrival_date_week_number, hotel)

e10<-ggplot(reservas_semanales, aes(x = arrival_date_week_number, y = n, color = hotel)) +
  geom_line() +
  labs(title = "Reservas por semana según tipo de hotel",
       x = "Semana del año",
       y = "Cantidad de reservas",
       color = "Hotel") +
  theme_classic()

e10
```



En el grafico observamos que hay una mayor reserva en city hotel

❖ La mayor cantidad de personas que van al hotel

```
#Grafico de barras para hallar la cantidad mayor de personas que vienen de un pais

e9<-ggplot(df_clean, aes(x = country)) +

 geom_bar(fill = "pink") +

 labs(title = "Canal de distribución más utilizado", x = "Canales", y = "Reservas")+

 theme_classic()

e9
```



Para este caso hemos utilizado el df_clean, en otras palabras el dataset modificando los valores atípicos, ya que gracias a eso nos damos cuenta la mayor cantidad de personas de un país es PTR.

CONCLUSIONES

- El análisis muestra que la mayoría de las personas prefieren el “City Hotel”, el cual concentra un número mucho mayor de reservas en comparación con el “Resort Hotel. Esto podría deberse a factores como su ubicación, accesibilidad o servicios ofrecidos. Esta tendencia revela una clara preferencia por un tipo específico de hotel, lo que

sugiere la importancia de identificar y aprovechar sus características más valoradas.

- Respecto a la evolución de la demanda, entre los años 2015 y 2017 se observó un crecimiento sostenido, con un pico en 2016. Aunque en 2017 hubo una leve disminución, la cantidad de reservas se mantuvo por encima de los niveles de 2015, lo que confirma una tendencia general positiva. Esto indica que los hoteles deben prepararse para atender un flujo creciente de huéspedes año tras año, ajustando su capacidad operativa y comercial.
- Respecto a las temporadas de mayor demanda, se identificó que el mes de agosto presenta el pico más alto de reservas, consolidándose como temporada alta. En contraste, enero y diciembre son los meses con menor actividad, lo que sugiere una temporada baja durante estos periodos.
- En promedio, los huéspedes del **Resort Hotel** tienen estancias más largas que los del **City Hotel**. Esto sugiere que quienes se alojan en el Resort tienden a realizar viajes de descanso o vacaciones, mientras que los del City Hotel probablemente realizan visitas más breves, posiblemente por motivos laborales o de negocios.
- Otro aspecto que se analizó fue la presencia de niños o bebés en las reservas. En este caso, se identificó que la mayoría de las reservas no incluyen menores, por lo que el perfil principal de los huéspedes serían adultos. Aun así, podría ser útil mantener algunos servicios dirigidos a familias, sin enfocarse exclusivamente en este segmento.
- El análisis de la variable relacionada al estacionamiento reveló que la gran mayoría de reservas no requieren espacios para autos, lo que podría relacionarse con el tipo de transporte utilizado por los clientes o con la ubicación céntrica de los hoteles.
- Asimismo, el mes con mayor número de cancelaciones también fue agosto, lo que podría estar vinculado a sobreofertas, reprogramaciones o alta estacionalidad.

- Además, las reservas en el “City Hotel” se mantienen constantes y en volúmenes superiores a lo largo de las semanas del año, reafirmando su preferencia en comparación al “Resort Hotel”.
- La mayor cantidad de personas que van a ambos hoteles son de PTR según el grafico de barras.

Bibliografía

Duong, E. (2023, abril 7). *Hotel Booking Project - Exploratory Data Analysis*. Medium. https://medium.com/@ethan.duong1120/hotel-booking-project-exploratory-data-analysis-48bcfb7ae7cd