

DemoDay_M2

Proyecto: “Demanda reservaciones de hotel”

Perla Peña

Experto: Erick

14 marzo 2022

[Descripción del dataset](#)

[Descripción de las variables](#)

[Procesamiento del data](#)

[EDA](#)

[Percnotación por tipo de mercado](#)

[Cancelación de la reservación por tipo de mercado](#)

[Costo promedio por tipo de habitación](#)

[Frecuencia por tipo de mercado](#)

[Estado de las reservaciones por mes y tipo de hotel](#)

[Pruebas estadísticas de hipótesis](#)

[Prueba de normalidad de los datos](#)

[Pruebas de correlación](#)

[Comprobar si existe correlación entre el tipo de deposito y la cancelación](#)

[Métodos de clasificación](#)

[Random Forest](#)

[Modelo](#)

[Importancia](#)

[Matriz de confusión y predicción](#)

[Importancia de las variables](#)

[Separación árboles](#)

[Grafica de las variables resultantes de Random Forest](#)

[Tiempo de espera](#)

Tipo de depósito
Precio promedio por noche
Tipo de segmento de mercado
Requisitos especiales
Regresión logística
Coeficientes de la regresión logística
SVM
Summary
Matriz de confusión
Weighted k-Nearest Neighbor Classifier
Conclusiones

Descripción del dataset

Tomado del artículo “Hotel booking demand datasets” de Nuno Antonio publicado en la Science direct en 2019.

“This data article describes two datasets with hotel demand data. One of the hotels (H1) is a resort hotel and the other is a city hotel (H2). Both datasets share the same structure, with 31 variables describing the 40,060 observations of H1 and 79,330 observations of H2. Each observation represents a hotel booking. Both datasets comprehend bookings due to arrive between the 1st of July of 2015 and the 31st of August 2017, including bookings that effectively arrived and bookings that were canceled. Since this is hotel real data, all data elements pertaining hotel or costumer identification were deleted. Due to the scarcity of real business data for scientific and educational purposes, these datasets can have an important role for research and education in revenue management, machine learning, or data mining, as well as in other fields.”

<https://www.sciencedirect.com/science/article/pii/S2352340918315191>

Descripción de las variables

Table 1 (continued)

Variable	Type	Description	Source/Engineering
<i>DistributionChannel</i>	Categorical	Booking distribution channel. The term "TA" means "Travel Agents" and "TO" means "Tour Operators"	BO, BL and DC
<i>IsCanceled</i>	Categorical	Value indicating if the booking was canceled (1) or not (0)	BO
<i>IsRepeatedGuest</i>	Categorical	Value indicating if the booking name was from a repeated guest (1) or not (0)	BO, BL and C/ Variable created by verifying if a profile was associated with the booking customer. If so, and if the customer profile creation date was prior to the creation date for the booking on the PMS database it was assumed the booking was from a repeated guest
<i>LeadTime</i>	Integer	Number of days that elapsed between the entering date of the booking into the PMS and the arrival date	BO and BL/ Subtraction of the entering date from the arrival date
<i>MarketSegment</i>	Categorical	Market segment designation. In categories, the term "TA" means "Travel Agents" and "TO" means "Tour Operators"	BO, BL and MS
<i>Meal</i>	Categorical	Type of meal booked. Categories are presented in standard hospitality meal packages: Undefined/SC – no meal package; BB – Bed & Breakfast; HB – Half board (breakfast and one other meal – usually dinner); FB – Full board (breakfast, lunch and dinner)	BO, BL and ML
<i>PreviousBookingsNotCanceled</i>	Integer	Number of previous bookings not cancelled by the customer prior to the current booking	BO and BL / In case there was no customer profile associated with the booking, the value is set to 0. Otherwise, the value is the number of bookings with the same customer profile created before the current booking and not canceled.
<i>PreviousCancellations</i>	Integer	Number of previous bookings that were cancelled by the customer prior to the current booking	BO and BL/ In case there was no customer profile associated with the booking, the value is set to 0. Otherwise, the value is the number of bookings with the same customer profile created before the current booking and canceled.
<i>RequiredCarParkingSpaces</i>	Integer	Number of car parking spaces required by the customer	BO and BL
<i>ReservationStatus</i>	Categorical	Reservation last status, assuming one of three categories: Canceled – booking was canceled by the customer; Check-Out – customer has checked in but already departed; No-Show – customer did not check-in and did inform the hotel of the reason why	BO

<i>ReservationStatusDate</i>	Date	Date at which the last status was set. This variable can be used in conjunction with the <i>ReservationStatus</i> to understand when was the booking canceled or when did the customer checked-out of the hotel	BO
<i>ReservedRoomType</i>	Categorical	Code of room type reserved. Code is presented instead of designation for anonymity reasons	BO and BL
<i>StaysInWeekendNights</i>	Integer	Number of weekend nights (Saturday or Sunday) the guest stayed or booked to stay at the hotel	BO and BL/ Calculated by counting the number of weekend nights from the total number of nights
<i>StaysInWeekNights</i>	Integer	Number of week nights (Monday to Friday) the guest stayed or booked to stay at the hotel	BO and BL/Calculated by counting the number of week nights from the total number of nights
<i>TotalOfSpecialRequests</i>	Integer	Number of special requests made by the customer (e.g. twin bed or high floor)	BO and BL/Sum of all special requests

Data in Brief

Hotel booking demand datasets

This data article describes two datasets with hotel demand data. One of the hotels (H1) is a resort hotel and the other is a city hotel (H2). Both datasets share the same structure, with 31 variables describing the 40,060 observations of H1

 <https://www.sciencedirect.com/science/article/pii/S2352340918315191>



Procesamiento del data

```
#Descripción del dataset
summary(booking)
dim(booking)
head(booking)
str(booking)

> dim(booking)
[1] 119390      32
> str(booking)
'data.frame': 119390 obs. of 32 variables:
 $ hotel                  : chr "Resort Hotel" "Resort Hotel" "Resort Hotel" "Resort Hotel" ...
 $ is_canceled             : int 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 ...
 $ 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                  : int   0 0 0 304 240 240 0 303 240 15 ...
$ company                : int   0 0 0 0 0 0 0 0 0 0 ...
$ 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 1 1 0 1 1 0 ...
$ reservation_status     : chr   "Check-Out" "Check-Out" "Check-Out" "Check-Out" ...
$ reservation_status_date: chr   "01/07/2015" "01/07/2015" "02/07/2015" "02/07/2015" ...

```

Revisar si existen NA

```

#Después de la limpieza de datos verificamos que no existan NA
(colSums(is.na(booking)))

      hotel          is_canceled
0
      lead_time        arrival_date_year
0
      arrival_date_month    arrival_date_week_number
0
      arrival_date_day_of_month stays_in_weekend_nights
0
      stays_in_week_nights       adults
0
      children            babies
0
      meal                  country
0
      market_segment        distribution_channel
0
      is_repeated_guest    previous_cancellations
0
      previous_bookings_not_canceled reserved_room_type
0
      assigned_room_type    booking_changes
0
      deposit_type            agent
0
      company                days_in_waiting_list
0
      customer_type          adr
0
      required_car_parking_spaces total_of_special_requests
0
      reservation_status    reservation_status_date
0

```

Modificación de la base de datos

```

#cambiar a formato fecha reservation_status_date
bforest <- mutate(booking, reservation_status_date = as.Date(reservation_status_date, "%d/%m/%Y"),
                  is_canceled = factor(is_canceled))

b1 <- mutate(booking, reservation_status_date = as.Date(reservation_status_date, "%d/%m/%Y"),
             hotel = factor(hotel),
             is_canceled = factor(is_canceled),
             market_segment = factor(market_segment))

>str(bforest)
'data.frame': 119390 obs. of  32 variables:
 $ hotel                      : chr "Resort Hotel" "Resort Hotel" "Resort Hotel" "Resort Hotel" ...
 $ is_canceled                 : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 2 2 ...
 $ lead_time                   : int 342 737 7 13 14 14 0 9 85 75 ...

> str(b1)
'data.frame': 119390 obs. of  32 variables:
 $ hotel                      : Factor w/ 2 levels "City Hotel","Resort Hotel": 2 2 2 2 2 2 2 2 2 ...
 $ is_canceled                 : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 2 2 ...
 $ lead_time                   : int 342 737 7 13 14 14 0 9 85 75 ...

#Estandarización de la base de datos (usada para SVM)
## Preparar los datos
hotel_df <- hotel_stays%>%
  select(is_canceled, hotel, arrival_date_month, meal,
         adr, deposit_type, lead_time, adults, required_car_parking_spaces,
         total_of_special_requests, market_segment,
         stays_in_week_nights, stays_in_weekend_nights)%>%
  mutate_if(is.character,factor)

#Instalar biblioteca
install.packages("tidymodels")
library(tidymodels)

#Normalización base de datos
hotel_rec <- recipe(is_canceled ~., data = hotel_train) %>%
  step_dummy(all_nominal(), -all_outcomes()) %>%
  step_zv(all_numeric())%>%
  step_normalize(all_numeric()) %>%
  prep()

```

EDA

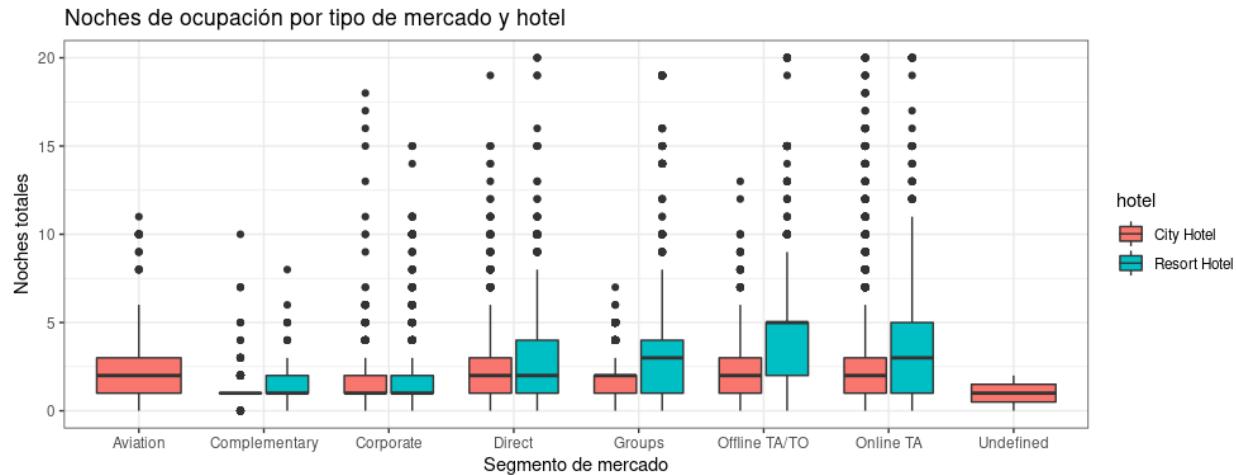
Percnotación por tipo de mercado

```

b1 %>%
  ggplot(aes(x= market_segment, y = stays_in_week_nights, fill= hotel))+
  geom_boxplot()+
  ggtitle("Noches de ocupación por tipo de mercado y hotel") +
  xlab("Segmento de mercado") +
  ylab("Noches totales")+

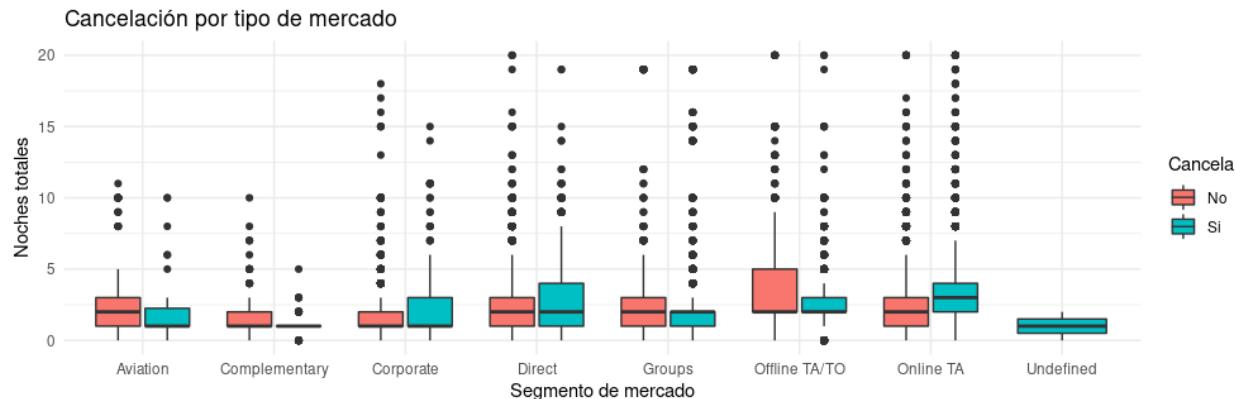
```

```
theme(plot.title = element_text(size=12))+  
theme_bw() +  
ylim(0,20)
```



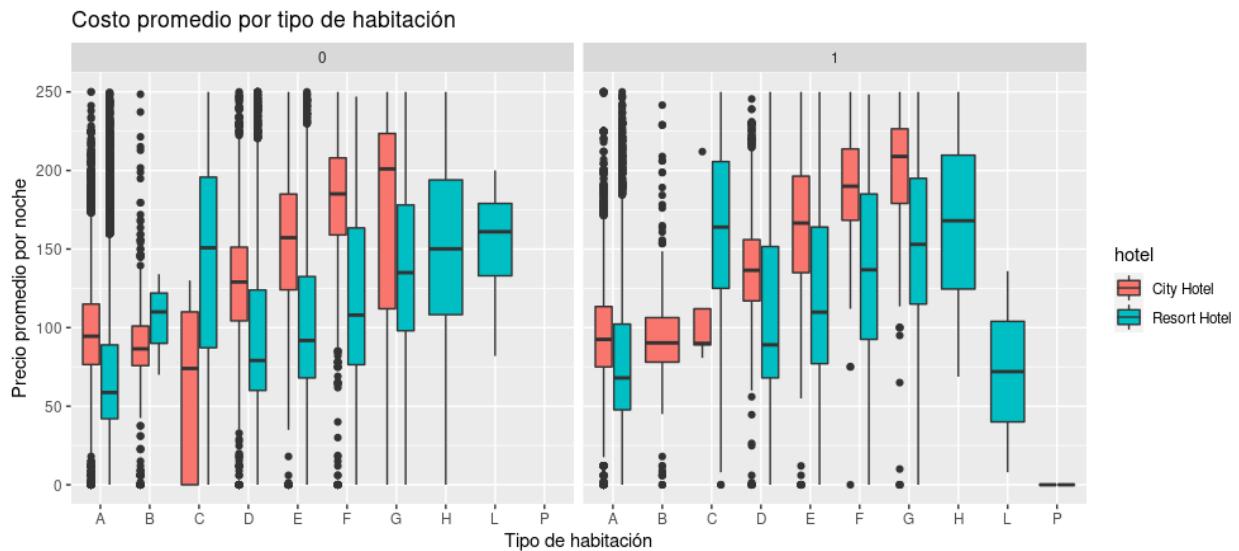
Cancelación de la reserva por tipo de mercado

```
b1 %>%  
  ggplot(aes(x=market_segment, y = stays_in_week_nights, fill = is_canceled))+  
  geom_boxplot()  
  ggtitle("Cancelación por tipo de mercado") +  
  xlab("Segmento de mercado") +  
  ylab("Noches totales") +  
  theme(plot.title = element_text(size=12)) +  
  theme_minimal() +  
  ylim(0,20) +  
  scale_fill_discrete(name = "Cancela", labels = c("No", "Si"))
```



Costo promedio por tipo de habitación

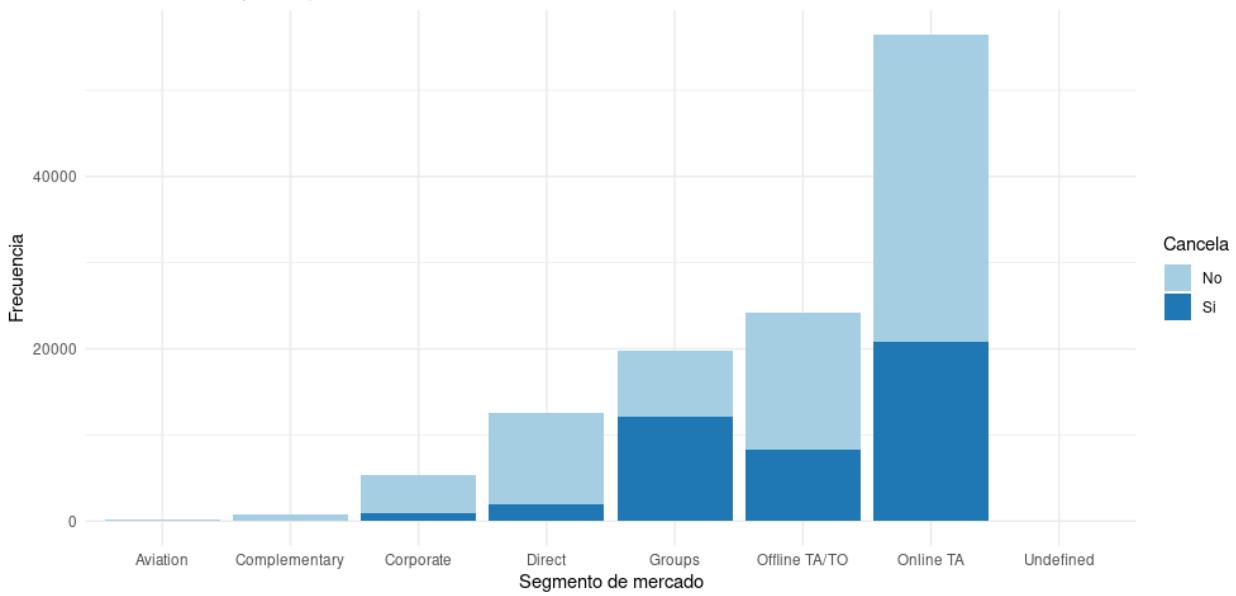
```
b1 %>%
  ggplot(aes(x=reserved_room_type, y = adr, fill = hotel))+
  geom_boxplot()+
  facet_wrap("is_canceled") +
  ylim(0,250)+
  ggtitle("Costo promedio por tipo de habitación") +
  xlab("Tipo de habitación") +
  ylab("Precio promedio por noche")+
  theme(plot.title = element_text(size=12))+
  theme_gray()
```



Frecuencia por tipo de mercado

```
b1 %>%
  ggplot(aes(x=market_segment, y=frequency(market_segment), fill=is_canceled)) +
  geom_bar(stat="identity")+
  scale_fill_brewer(palette="Paired", name = "Cancela", labels = c("No", "Si"))+
  theme_minimal()+
  ggtitle("Reservaciones por segmento de mercado") +
  xlab("Segmento de mercado") +
  ylab("Frecuencia")
```

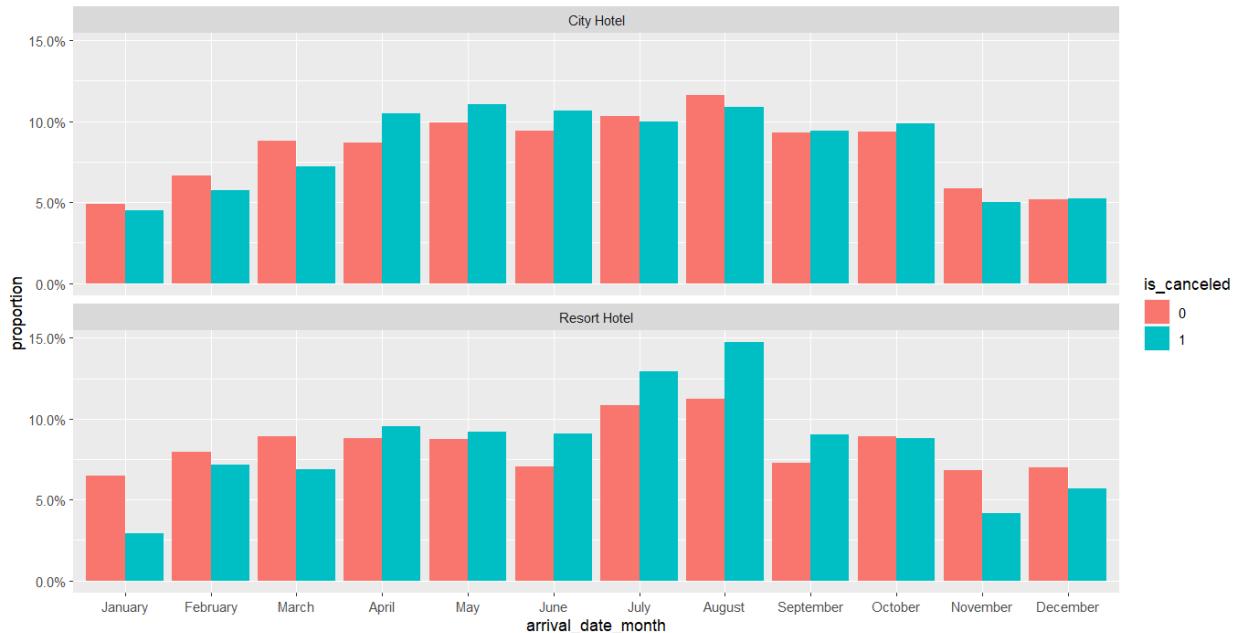
Reservaciones por segmento de mercado



Estado de las reservaciones por mes y tipo de hotel

```
#Seleccionar datos
hotel_stays <- bforest %>%
  mutate(hotel = factor(hotel), children = case_when(children + babies > 0 ~ "children",
                                                       TRUE ~ "none"),
         required_car_parking_spaces = case_when(required_car_parking_spaces > 0 ~ "parking",
                                                       TRUE ~ "none")) %>%
  select(-reservation_status, -babies)

#Hacer el plot
hotel_stays%>%
  mutate(arrival_date_month = factor(arrival_date_month,
                                       levels = month.name)) %>%
  count(hotel, arrival_date_month, is_canceled)%>%
  group_by(hotel, is_canceled)%>%
  mutate(proportion = n / sum(n))%>%
  ggplot(aes(arrival_date_month, proportion, fill = is_canceled))+
  geom_col(position = "dodge")+
  scale_y_continuous(labels = scales::percent_format())+
  facet_wrap(~hotel, nrow = 2)
```



Pruebas estadísticas de hipótesis

Prueba de normalidad de los datos

```
##Verificar la normalidad de los datos con un histograma
#opción 1
hist(b1$lead_time)

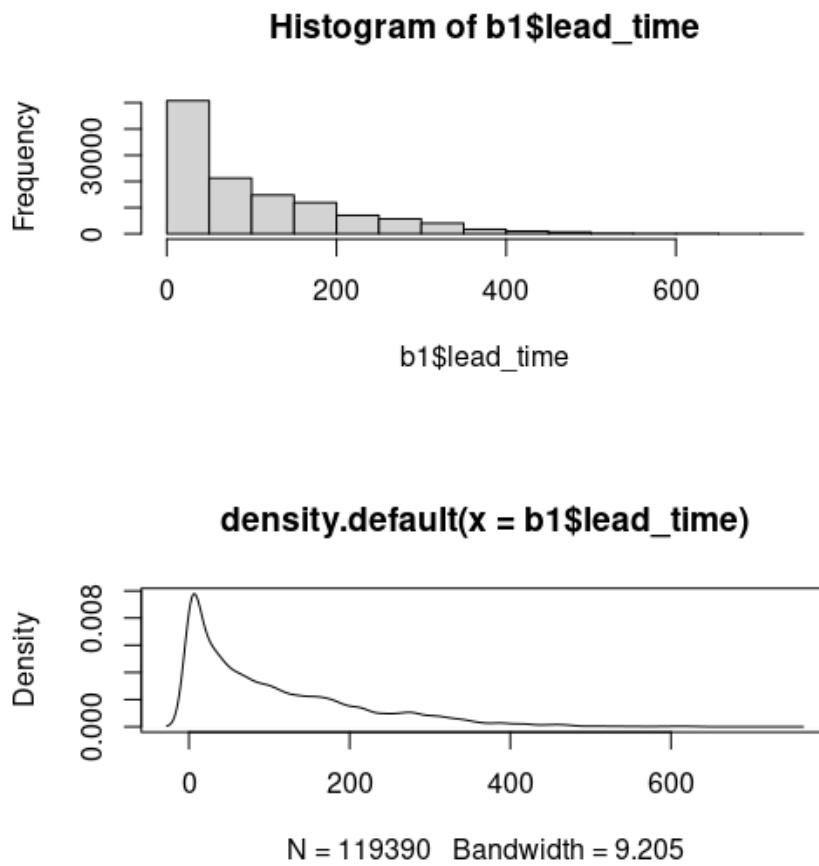
#opción 2
install.packages("skimr")
library(skimr)
skim(hotel_stays)

#Densidad
d <- density(b1$lead_time)
plot(d)

#Prueba de shapiro test
lt.test <- shapiro.test(b1$lead_time[0:5000])
lt.test

Shapiro-Wilk normality test

data: b1$lead_time[0:5000]
W = 0.88581, p-value < 2.2e-16
```



Valor de $p < 0.05$ por lo que rechazamos H_0 . Los valores no son gaussianos a una significancia del 5%

```
skim(hotel_stays)
-- Data Summary -----
Name          hotel_stays
Number of rows 119390
Number of columns 30

Column type frequency:
 character      11
 Date           1
 factor         2
 numeric        16

Group variables     None

-- Variable type: character -----
# A tibble: 11 x 8
  skim_variable n_missing
  * <chr>          <int>
  1 arrival_date_month      0
  2 children            0
```

```

3 meal 0
4 country 0
5 market_segment 0
6 distribution_channel 0
7 reserved_room_type 0
8 assigned_room_type 0
9 deposit_type 0
10 customer_type 0
11 required_car_parking_spaces 0
  complete_rate min max empty n_unique
* <dbl> <int> <int> <int> <int>
1 1 3 9 0 12
2 1 4 8 0 2
3 1 2 9 0 5
4 1 2 4 0 178
5 1 6 13 0 8
6 1 3 9 0 5
7 1 1 1 0 10
8 1 1 1 0 12
9 1 10 10 0 3
10 1 5 15 0 4
11 1 4 7 0 2
  whitespace
* <int>
1 0
2 0
3 0
4 0
5 0
6 0
7 0
8 0
9 0
10 0
11 0

-- Variable type: Date -----
# A tibble: 1 x 7
  skim_variable n_missing complete_rate
* <chr> <int> <dbl>
1 reservation_status_date 0 1
  min max median n_unique
* <date> <date> <date> <int>
1 2014-10-17 2017-09-14 2016-08-07 926

-- Variable type: factor -----
# A tibble: 2 x 6
  skim_variable n_missing complete_rate ordered
* <chr> <int> <dbl> <lgl>
1 hotel 0 1 FALSE
2 is_canceled 0 1 FALSE
  n_unique top_counts
* <int> <chr>
1 2 Cit: 79330, Res: 40060
2 0: 75166, 1: 44224

-- Variable type: numeric -----
# A tibble: 16 x 11
  skim_variable n_missing
* <chr> <int>
1 lead_time 0

```

```

2 arrival_date_year          0
3 arrival_date_week_number   0
4 arrival_date_day_of_month  0
5 stays_in_weekend_nights   0
6 stays_in_week_nights      0
7 adults                     0
8 is_repeated_guest         0
9 previous_cancellations    0
10 previous_bookings_not_canceled 0
11 booking_changes           0
12 agent                      0
13 company                    0
14 days_in_waiting_list     0
15 adr                        0
16 total_of_special_requests 0

  complete_rate      mean       sd      p0      p25
*   <dbl>        <dbl>    <dbl>    <dbl>    <dbl>
  1       104.     107.      0       18
  2       1 2016.    0.707  2015    2016
  3       1 27.2     13.6     1       16
  4       1 15.8     8.78     1       8
  5       1 0.928    0.999    0       0
  6       1 2.50     1.91     0       1
  7       1 1.86     0.579    0       2
  8       1 0.0319   0.176    0       0
  9       1 0.0871   0.844    0       0
 10      1 0.137    1.50     0       0
 11      1 0.221    0.652    0       0
 12      1 74.8     107.     0       7
 13      1 10.8     53.9     0       0
 14      1 2.32     17.6     0       0
 15      1 102.     50.5    -6.38   69.3
 16      1 0.571    0.793    0       0

  p50      p75      p100 hist
* <dbl>    <dbl>    <dbl> <chr>
  1 69      160     737  █
  2 2016    2017    2017  ████
  3 28      38      53   ████
  4 16      23      31   ████
  5 1       2       19   █
  6 2       3       50   █
  7 2       2       55   █
  8 0       0       1    █
  9 0       0       26   █
 10 0       0       72   █
 11 0       0       21   █
 12 9       152     535  ████
 13 0       0       543  ████
 14 0       0       391  ████
 15 94.6   126    5400 ████
 16 0       1       5    █

```

Pruebas de correlación

De acuerdo a las pruebas de normalidad se detectó que los datos no cumplen con esta condición por lo que se optó por utilizar prueba no paramétrica.

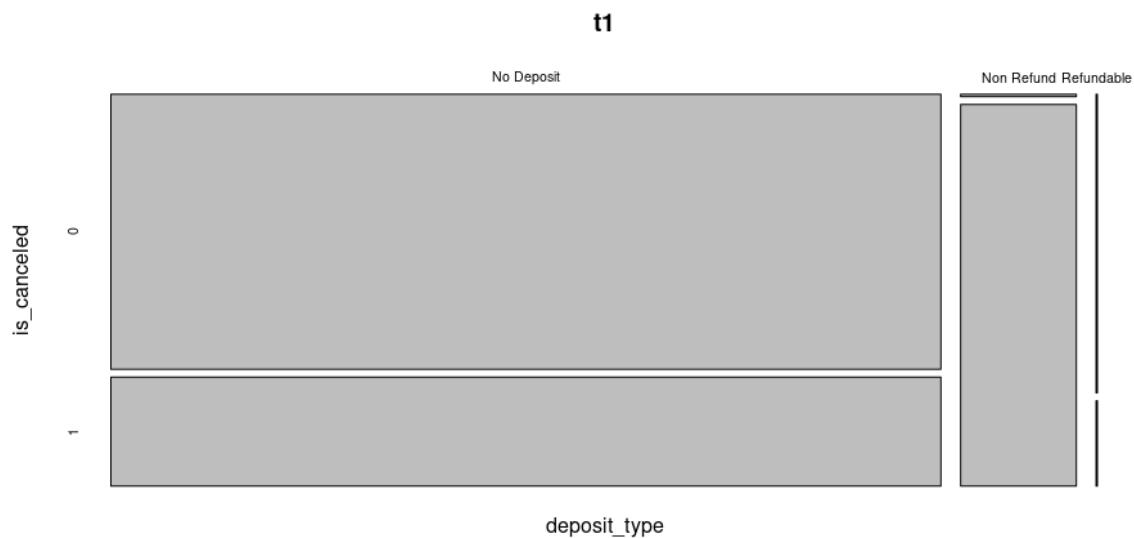
Comprobar si existe correlación entre el tipo de deposito y la cancelación

```
#verificar otras variables
attach(booking)
t1 <- table(deposit_type, is_canceled)
plot(t1)
t1.cor <- round(cor(t1),1)
t1.cor
install.packages("corrplot")
install.packages("PerformanceAnalytics")
library(corrplot)
library(PerformanceAnalytics)

#correlación
corrplot(t1.cor, method="number", type="upper")
chart.Correlation(t1, histogram = F, pch = 19)

> t1
is_canceled
deposit_type      0      1
  No Deposit 74947 29694
  Non Refund    93 14494
  Refundable    126    36

#no se agregan resultados ya que no tienen lógica los plot
```



Métodos de clasificación

Random Forest

```

#Instalar paquetes
urlPackage <- "https://cran.r-project.org/src/contrib/Archive/randomForest/randomForest_4.6-12.tar.gz"
install.packages(urlPackage, repos=NULL, type="source")
library(randomForest)

#Crear semilla, datos test y datos train
set.seed(101)
tamano.total <- nrow(bforest)
tamano.entreno <- round(tamano.total*0.7)
datos.indices <- sample(1:tamano.total , size=tamano.entreno)
datos.entreno <- bforest[datos.indices,]
datos.test <- bforest[-datos.indices,]

#Modelo Random Forest
modelo <- randomForest(is_canceled~, data=datos.test)
modelo

#gráficos y resultados
varImpPlot(modelo)
plot(modelo)
legend("right", colnames(modelo$err.rate), lty = 1:5, col = 1:6)
importance(modelo2)

# Separar árboles
> split_var_1 <- sapply(seq_len(modelo$ntree),
+                         function(i) getTree(modelo, i, labelVar=TRUE)[1, "split var"])
> table(split_var_1)
split_var_1

```

Modelo

```

Call:
randomForest(formula = is_canceled ~ ., data = datos.test
              Type of random forest: classification
              Number of trees: 500
              No. of variables tried at each split: 5

              OOB estimate of error rate: 0.01%
Confusion matrix:
      0     1 class.error
0 22522     0 0.0000000000
1    2 13293 0.0001504325
> |

```

Importancia

```

importance(modelo)
MeanDecreaseGini

```

```

hotel                      64.451176
lead_time                  1060.414075
arrival_date_year           145.024817
arrival_date_month           78.903162
arrival_date_week_number     156.995166
arrival_date_day_of_month    92.282868
stays_in_weekend_nights      58.899494
stays_in_week_nights          97.893647
adults                      40.672072
children                    16.697532
babies                       1.422965
meal                          43.000536
country                     1390.876412
market_segment                662.677977
distribution_channel          148.391380
is_repeated_guest              18.708229
previous_cancellations        597.263148
previous_bookings_not_canceled 49.835286
reserved_room_type             56.435486
assigned_room_type              197.988419
booking_changes                 232.813508
deposit_type                  3112.127408
agent                         331.114216
company                        27.670444
days_in_waiting_list            9.898190
customer_type                  225.226143
adr                            259.516468
required_car_parking_spaces      315.916987
total_of_special_requests       720.081914
reservation_status               27901.715443
reservation_status_date         782.833377
> importance(modelo2)
                                         MeanDecreaseGini
hotel                           0.000000e+00
lead_time                      1.574528e-02
arrival_date_year                0.000000e+00
arrival_date_month                0.000000e+00
arrival_date_week_number          0.000000e+00
arrival_date_day_of_month         0.000000e+00
stays_in_weekend_nights           0.000000e+00
stays_in_week_nights                0.000000e+00
adults                           0.000000e+00
children                          0.000000e+00
babies                            0.000000e+00
meal                             0.000000e+00
country                          6.459493e-01
market_segment                   0.000000e+00
distribution_channel                0.000000e+00
is_repeated_guest                  0.000000e+00
previous_cancellations              0.000000e+00
previous_bookings_not_canceled     0.000000e+00
reserved_room_type                  0.000000e+00
assigned_room_type                  0.000000e+00
booking_changes                     0.000000e+00
deposit_type                      2.480106e+02
agent                            0.000000e+00
company                           0.000000e+00
days_in_waiting_list                  0.000000e+00
customer_type                      1.772218e-01
adr                               0.000000e+00
required_car_parking_spaces         0.000000e+00

```

```
total_of_special_requests      0.000000e+00
reservation_status            3.871516e+04
reservation_status_date       0.000000e+00
```

Matriz de confusión y predicción

```
> pred <- predict(modelo, newdata = datos.entreno)
> caret::confusionMatrix(pred, datos.entreno$is_canceled)
Confusion Matrix and Statistics

Reference
Prediction      0      1
      0 22522      0
      1      0 13295

Accuracy : 1
95% CI : (0.9999, 1)
No Information Rate : 0.6288
P-Value [Acc > NIR] : < 2.2e-16

Kappa : 1

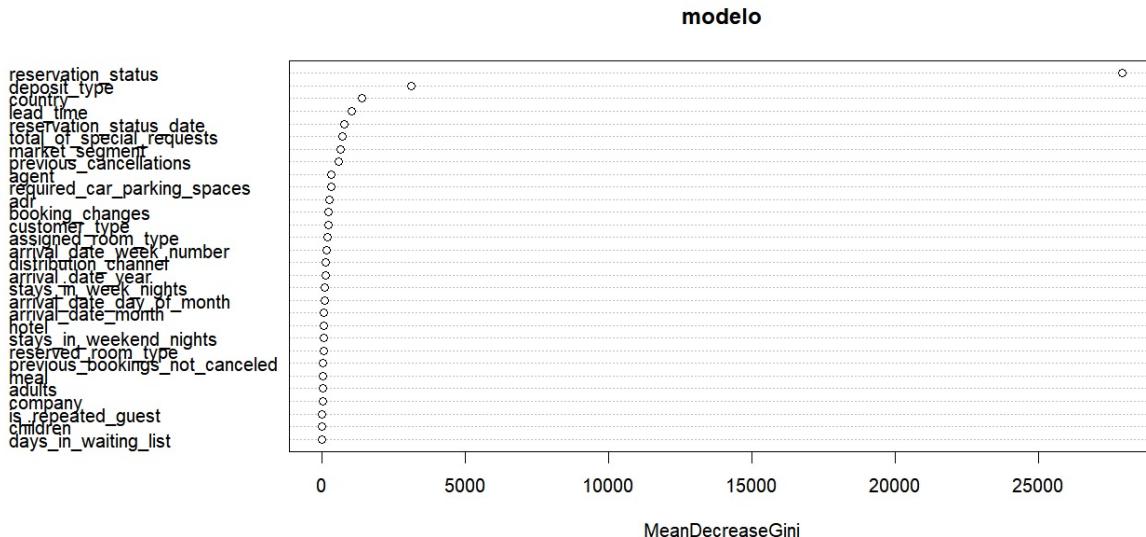
McNemar's Test P-Value : NA

Sensitivity : 1.0000
Specificity : 1.0000
Pos Pred Value : 1.0000
Neg Pred Value : 1.0000
Prevalence : 0.6288
Detection Rate : 0.6288
Detection Prevalence : 0.6288
Balanced Accuracy : 1.0000

'Positive' Class : 0
```

Se destaca de la matriz de confusión que el modelo analizó todos los datos.

Importancia de las variables



- Se excluye `reservation_status` ya que indica si fue cancelada o no
- Se rescatan del modelo:
 - `deposit_type`
 - `country`
 - `lead_time`
 - `total_of_special_requests`

Separación árboles

```

split_var_1
    adr
    17
    adults
    0
    agent
    0
    arrival_date_day_of_month
    0
    arrival_date_month
    0
    arrival_date_week_number
    0
    arrival_date_year
    0
    assigned_room_type
    23
    babies
    0
    booking_changes
  
```

```

      22
children
      0
company
      2
country
      63
customer_type
      8
days_in_waiting_list
      2
deposit_type
      86
distribution_channel
      7
hotel
      12
is_repeated_guest
      1
lead_time
      37
market_segment
      17
meal
      0
previous_bookings_not_canceled
      1
previous_cancellations
      40
required_car_parking_spaces
      20
reservation_status
      91
reservation_status_date
      21
reserved_room_type
      0
stays_in_week_nights
      0
stays_in_weekend_nights
      0
total_of_special_requests
      30

```

Grafica de las variables resultantes de Random Forest

```

#agrupar por lead_time (Tiempo de espera)
grouplead_time <- aggregate(booking["is_canceled"], by=booking["lead_time"], mean)
grouplead_time

grouplead_time %>%
  ggplot(aes(x=lead_time, y = (is_canceled), color = is_canceled))+
  geom_point()+
  geom_smooth(method = "lm", se = FALSE)+
  ggtitle("Tiempo de espera y cancelación de la reservación") +
  xlab("Tiempo de espera") +
  ylab("Cancelación")

```

```

#agrupar por deposit_type (Tipo de depósito)
groupdeposit_type <- aggregate(booking["is_canceled"], by=booking["deposit_type"], mean)
groupdeposit_type

plot(factor(groupdeposit_type$deposit_type), groupdeposit_type$is_canceled)

groupdeposit_type %>%
  ggplot(aes(x= factor(deposit_type), y = is_canceled, fill= factor(deposit_type)))+
  geom_bar(stat = "identity")+
  theme_minimal()+
  ggtitle("Tipo de depósito") +
  xlab("Tipo de depósito") +
  ylab("Cancelación")+
  scale_fill_brewer(palette="Paired", name = "Tipo de depósito")

#agrupar por adr (Tarifa diaria promedio)
groupadr <- aggregate(booking["is_canceled"], by=booking["adr"], mean)
groupadr

groupadr %>%
  ggplot(aes(x=adr, y = (is_canceled)))+
  geom_point(shape = 18, fill="blue", color="darkred", size=1)+
  xlim(0,500)+
  geom_smooth(method = "lm", se = FALSE)+
  ggtitle("Precio promedio por noche") +
  xlab("Precio promedio") +
  ylab("Cancelación")

#agrupar por market_segment (Tipo de segmento de mercado)
groupmarket_segment <- aggregate(booking["is_canceled"], by=booking["market_segment"], mean)
groupmarket_segment

plot(factor(groupmarket_segment$market_segment), groupmarket_segment$is_canceled)

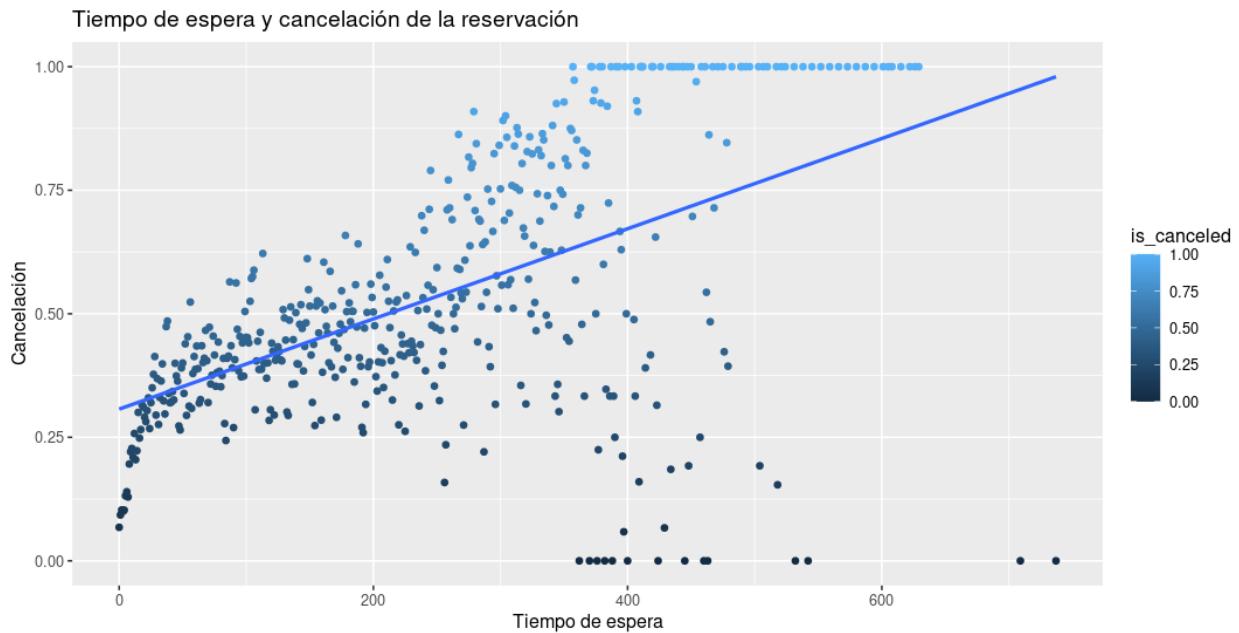
groupmarket_segment %>%
  ggplot(aes(x= factor(market_segment), y = is_canceled, fill=factor(market_segment)))+
  geom_bar(stat = "identity")+
  theme_minimal()+
  ggtitle("Tipo de segmento de mercado") +
  xlab("Tipo de segmento de mercado") +
  ylab("Cancelación")+
  scale_fill_brewer(palette="Paired")

#agrupar por total_special_request (Total de requerimientos especiales)
grouprequest <- aggregate(booking["is_canceled"], by=booking["total_of_special_requests"], mean)
grouprequest

grouprequest %>%
  ggplot(aes(x= factor(total_of_special_requests), y = is_canceled))+
  geom_point()

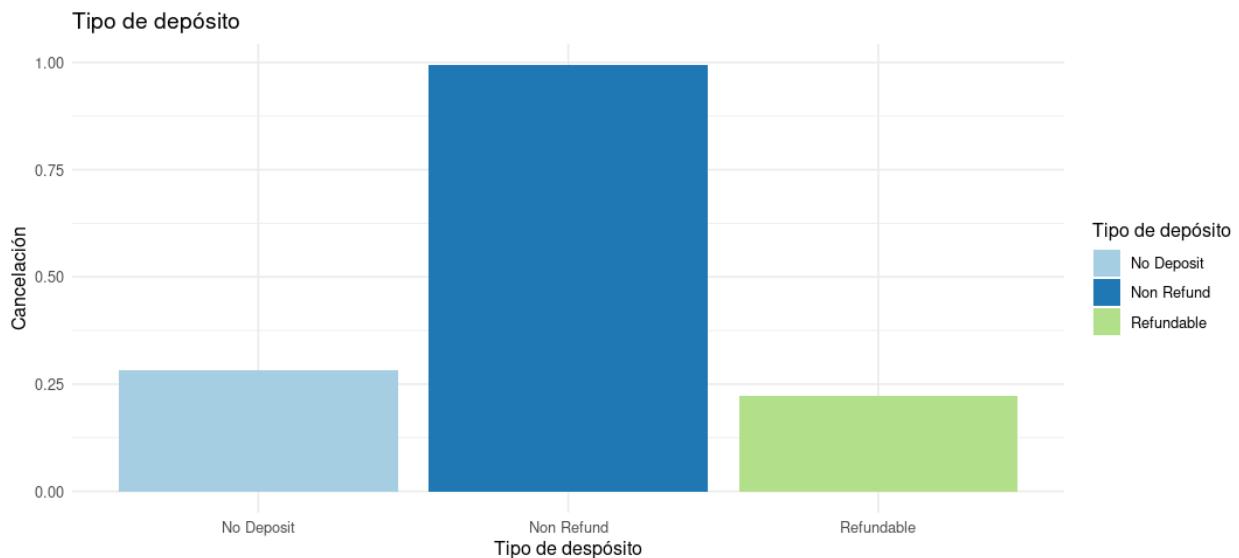
```

Tiempo de espera



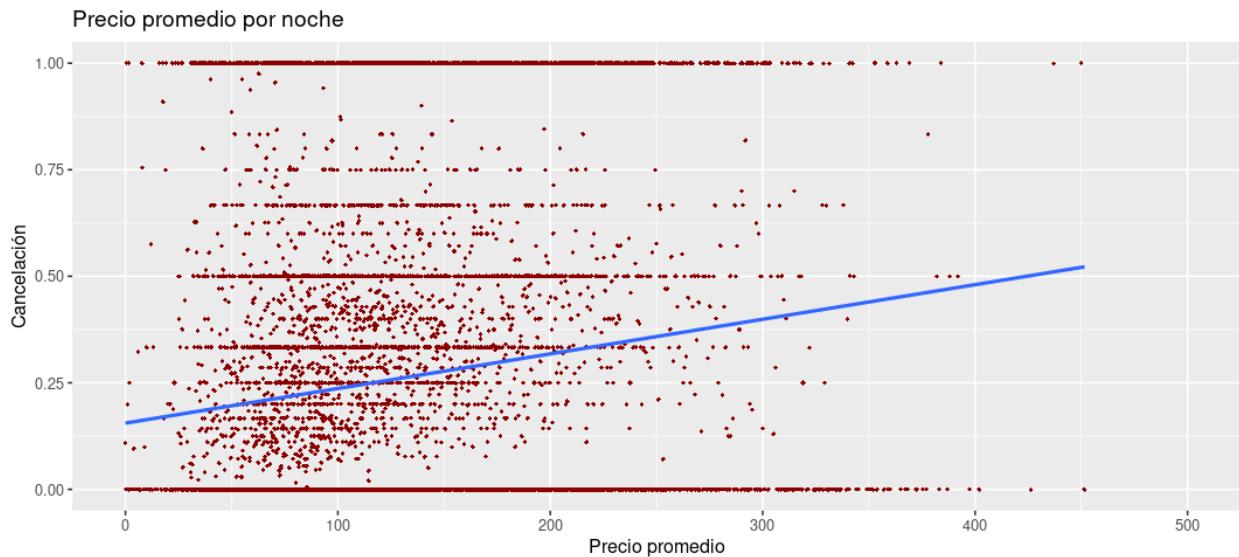
Entre mayor es el tiempo de espera aumenta la cancelación de las reservación.

Tipo de depósito

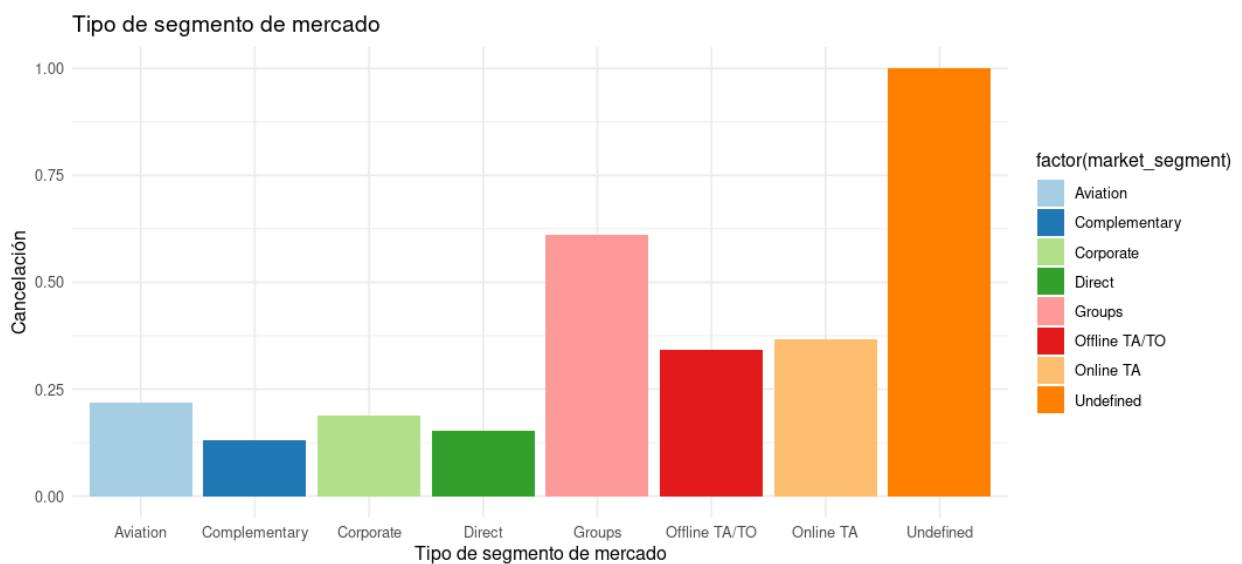


El 99% de los que pagaron la cifra completa cancelaron, es necesario revisar la base completa para tomar una decisión.

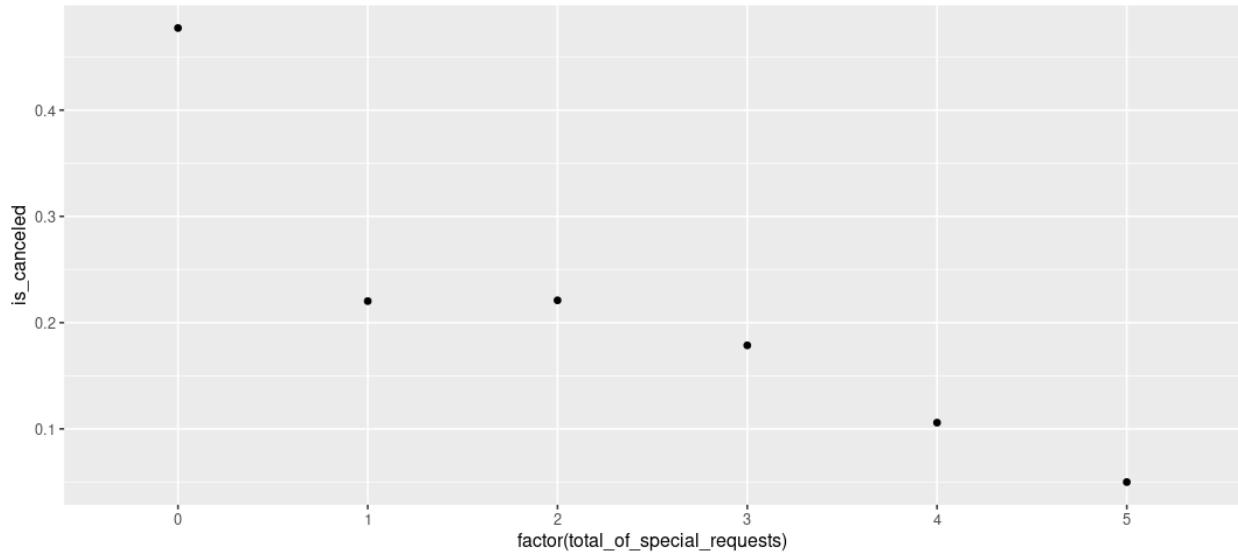
Precio promedio por noche



Tipo de segmento de mercado



Requisitos especiales



Regresión logística

```

#set.seed(1)
df <- bforest
nobs <- nrow(bforest)
itrain <- sample(nobs, 0.8 * nobs)
train <- df[itrain, ]
test <- df[-itrain, ]

#Regresión logística con todas las variables
rl <- glm(is_canceled ~., data = train)
rl
summary(rl)
#Regresión logística con las variables de random forest
rl2 <- glm(is_canceled ~ deposit_type + country + lead_time +
            market_segment + adr, family = binomial, data = train)
rl2
summary(rl2)
#Ajuste de la regresión logística sin la variable que no aporta
rl3 <- update(rl2, ~. -country)
summary(rl3)
#Ajuste de la regresión logística con la variable faltante
rl4 <- update(rl3, ~. +total_of_special_requests)
summary(rl4)
#coeficientes
rl4$coeff
plot(rl4)

```

Coeficientes de la regresión logística

```

Call:
glm(formula = is_canceled ~ deposit_type + country + lead_time +

```

```

market_segment + adr, family = binomial, data = train)

Deviance Residuals:
    Min      1Q  Median      3Q     Max 
-3.2576 -0.7322 -0.4590  0.1997  2.7651 

Coefficients:
                                         Estimate Std. Error z value
(Intercept)                         -1.500e+01  5.354e+02 -0.028
deposit_typeNon Refund               5.475e+00  1.191e-01 45.978
deposit_typeRefundable             -3.617e-02  2.260e-01 -0.160
countryAGO                          1.482e+01  5.354e+02  0.028
countryAIA                          -1.305e-01  7.572e+02  0.000
countryALB                          1.246e+01  5.354e+02  0.023
countryAND                          1.507e+01  5.354e+02  0.028
countryARE                          1.536e+01  5.354e+02  0.029
countryARG                          1.257e+01  5.354e+02  0.023
countryARM                          1.377e+01  5.354e+02  0.026
countryASM                          5.838e-02  7.572e+02  0.000
countryATA                          9.178e-01  6.486e+02  0.001
countryATF                          -2.872e-01  7.572e+02  0.000
countryAUS                          1.251e+01  5.354e+02  0.023
countryAUT                          1.217e+01  5.354e+02  0.023
countryAZE                          1.360e+01  5.354e+02  0.025
countryBDI                          1.585e+00  7.572e+02  0.002
countryBEL                          1.222e+01  5.354e+02  0.023
countryBEN                          2.613e+01  6.112e+02  0.043
countryBFA                          4.973e-01  7.572e+02  0.001
countryBGD                          1.524e+01  5.354e+02  0.028
countryBGR                          1.193e+01  5.354e+02  0.022
countryBHR                          1.416e+01  5.354e+02  0.026
countryBHS                          1.158e+00  7.572e+02  0.002
countryBIH                          1.278e+01  5.354e+02  0.024
countryBLR                          1.348e+01  5.354e+02  0.025
countryBOL                          3.287e-01  5.721e+02  0.001
countryBRA                          1.324e+01  5.354e+02  0.025
countryBRB                          8.131e-01  6.182e+02  0.001
countryBWA                          -1.192e-01  7.572e+02  0.000
countryCAF                          4.464e-01  6.176e+02  0.001
countryCHE                          1.249e+01  5.354e+02  0.023
countryCHL                          1.265e+01  5.354e+02  0.024
countryCHN                          1.372e+01  5.354e+02  0.026
countryCIV                          1.369e+01  5.354e+02  0.026
countryCMR                          2.177e-01  5.664e+02  0.000
countryCN                           1.229e+01  5.354e+02  0.023
countryCOL                          1.285e+01  5.354e+02  0.024
countryCOM                          4.049e-02  6.510e+02  0.000
countryCPV                          1.426e+01  5.354e+02  0.027
countryCRI                          1.113e+01  5.354e+02  0.021
countryCUB                          1.134e+00  5.696e+02  0.002
countryCYM                          1.545e-01  7.572e+02  0.000
countryCYP                          1.264e+01  5.354e+02  0.024
countryCZE                          1.247e+01  5.354e+02  0.023
countryDEU                          1.198e+01  5.354e+02  0.022
countryDJI                          -1.592e-01  7.572e+02  0.000
countryDNK                          1.238e+01  5.354e+02  0.023
countryDOM                          1.344e+01  5.354e+02  0.025
countryDZA                          1.292e+01  5.354e+02  0.024
countryECU                          1.339e+01  5.354e+02  0.025
countryEGY                          1.356e+01  5.354e+02  0.025
countryESP                          1.283e+01  5.354e+02  0.024

```

countryEST	1.233e+01	5.354e+02	0.023
countryETH	1.361e+01	5.354e+02	0.025
countryFIN	1.206e+01	5.354e+02	0.023
countryFRA	1.234e+01	5.354e+02	0.023
countryFRO	1.337e+01	5.354e+02	0.025
countryGAB	1.397e+01	5.354e+02	0.026
countryGBR	1.231e+01	5.354e+02	0.023
countryGEO	1.426e+01	5.354e+02	0.027
countryGGY	2.748e+01	6.552e+02	0.042
countryGHA	1.406e+01	5.354e+02	0.026
countryGIB	1.408e+01	5.354e+02	0.026
countryGLP	2.687e+01	6.557e+02	0.041
countryGNB	1.315e+01	5.354e+02	0.025
countryGRC	1.282e+01	5.354e+02	0.024
countryGTM	5.159e-01	5.984e+02	0.001
countryGUY	7.913e-01	7.572e+02	0.001
countryHKG	1.555e+01	5.354e+02	0.029
countryHND	2.760e+01	7.572e+02	0.036
countryHRV	1.271e+01	5.354e+02	0.024
countryHUN	1.285e+01	5.354e+02	0.024
countryIDN	1.497e+01	5.354e+02	0.028
countryIMN	2.692e+01	7.572e+02	0.036
countryIND	1.246e+01	5.354e+02	0.023
countryIRL	1.260e+01	5.354e+02	0.024
countryIRN	1.298e+01	5.354e+02	0.024
countryIRQ	1.047e+00	5.587e+02	0.002
countryISL	1.140e+01	5.354e+02	0.021
countryISR	1.276e+01	5.354e+02	0.024
countryITA	1.319e+01	5.354e+02	0.025
countryJAM	4.666e-01	5.830e+02	0.001
countryJEY	2.665e+01	5.939e+02	0.045
countryJOR	1.182e+01	5.354e+02	0.022
countryJPN	1.188e+01	5.354e+02	0.022
countryKAZ	1.302e+01	5.354e+02	0.024
countryKEN	5.989e-01	6.097e+02	0.001
countryKHM	2.743e+01	6.557e+02	0.042
countryKNA	1.092e+00	7.572e+02	0.001
countryKOR	1.334e+01	5.354e+02	0.025
countryKWT	1.290e+01	5.354e+02	0.024
countryLAO	1.539e-01	6.557e+02	0.000
countryLBN	1.312e+01	5.354e+02	0.024
countryLBY	1.111e+00	5.699e+02	0.002
countryLCA	4.353e-01	7.572e+02	0.001
countryLIE	4.864e-01	7.572e+02	0.001
countryLKA	8.119e-01	5.756e+02	0.001
countryLTU	1.189e+01	5.354e+02	0.022
countryLUX	1.310e+01	5.354e+02	0.024
countryLVA	1.209e+01	5.354e+02	0.023
countryMAC	1.612e+01	5.354e+02	0.030
countryMAR	1.360e+01	5.354e+02	0.025
countryMCO	1.276e+01	5.354e+02	0.024
countryMDG	2.203e+00	7.572e+02	0.003
countryMDV	1.511e+01	5.354e+02	0.028
countryMEX	1.148e+01	5.354e+02	0.021
countryMKD	1.225e+01	5.354e+02	0.023
countryMLI	5.000e-01	7.572e+02	0.001
countryMLT	1.323e+01	5.354e+02	0.025
countryMMR	2.881e-01	7.572e+02	0.000
countryMNE	1.299e+01	5.354e+02	0.024
countryMOZ	1.258e+01	5.354e+02	0.024
countryMRT	-3.662e-01	7.572e+02	0.000

countryMUS	1.267e+01	5.354e+02	0.024
countryMWI	5.158e-01	6.557e+02	0.001
countryMYS	1.210e+01	5.354e+02	0.023
countryMYT	2.617e+01	6.532e+02	0.040
countryNGA	1.460e+01	5.354e+02	0.027
countryNLD	1.221e+01	5.354e+02	0.023
countryNOR	1.267e+01	5.354e+02	0.024
countryNPL	4.974e-01	7.572e+02	0.001
countryNULL	1.287e+01	5.354e+02	0.024
countryNZL	1.100e+01	5.354e+02	0.021
countryOMN	1.283e+01	5.354e+02	0.024
countryPAK	1.426e+01	5.354e+02	0.027
countryPAN	2.652e-01	5.711e+02	0.000
countryPER	1.317e+01	5.354e+02	0.025
countryPHL	1.443e+01	5.354e+02	0.027
countryPOL	1.234e+01	5.354e+02	0.023
countryPRI	1.265e+01	5.354e+02	0.024
countryPRT	1.402e+01	5.354e+02	0.026
countryPRY	2.089e-01	5.984e+02	0.000
countryPYF	8.238e-01	7.572e+02	0.001
countryQAT	1.496e+01	5.354e+02	0.028
countryROU	1.263e+01	5.354e+02	0.024
countryRUS	1.329e+01	5.354e+02	0.025
countryRWA	6.659e-01	6.501e+02	0.001
countrySAU	1.453e+01	5.354e+02	0.027
countrySDN	1.848e+00	7.572e+02	0.002
countrySEN	1.461e+01	5.354e+02	0.027
countrySGP	1.388e+01	5.354e+02	0.026
countrySLE	-9.023e-02	7.572e+02	0.000
countrySLV	-5.289e-04	6.551e+02	0.000
countrySMR	1.268e+00	7.572e+02	0.002
countrySRB	1.079e+01	5.354e+02	0.020
countrySTP	1.967e+00	7.572e+02	0.003
countrySUR	3.121e-01	5.828e+02	0.001
countrySVK	1.294e+01	5.354e+02	0.024
countrySVN	1.271e+01	5.354e+02	0.024
countrySWE	1.232e+01	5.354e+02	0.023
countrySYC	1.420e+01	5.354e+02	0.027
countrySYR	7.974e-01	6.098e+02	0.001
countryTGO	4.698e-01	6.338e+02	0.001
countryTHA	1.336e+01	5.354e+02	0.025
countryTJK	2.729e+01	5.783e+02	0.047
countryTMP	1.366e+01	5.354e+02	0.026
countryTUN	1.400e+01	5.354e+02	0.026
countryTUR	1.354e+01	5.354e+02	0.025
countryTWN	1.285e+01	5.354e+02	0.024
countryTZA	1.326e+01	5.354e+02	0.025
countryUGA	3.980e-01	6.546e+02	0.001
countryUKR	1.324e+01	5.354e+02	0.025
countryUMI	2.717e+01	7.572e+02	0.036
countryURY	1.283e+01	5.354e+02	0.024
countryUSA	1.264e+01	5.354e+02	0.024
countryUZB	1.412e+01	5.354e+02	0.026
countryVEN	1.401e+01	5.354e+02	0.026
countryVNM	1.310e+01	5.354e+02	0.024
countryZAF	1.327e+01	5.354e+02	0.025
countryZMB	1.363e+01	5.354e+02	0.025
countryZWE	1.473e+01	5.354e+02	0.028
lead_time	5.643e-03	9.897e-05	57.018
market_segmentComplementary	-9.322e-01	2.187e-01	-4.263
market_segmentCorporate	-1.017e+00	1.859e-01	-5.469

market_segmentDirect	-9.241e-01	1.821e-01	-5.075
market_segmentGroups	-4.593e-01	1.822e-01	-2.521
market_segmentOffline TA/T0	-8.000e-01	1.815e-01	-4.408
market_segmentOnline TA	6.181e-01	1.802e-01	3.431
market_segmentUndefined	1.449e+01	3.786e+02	0.038
adr	3.582e-03	1.801e-04	19.887
	Pr(> z)		
(Intercept)	0.977645		
deposit_typeNon Refund	< 2e-16 ***		
deposit_typeRefundable	0.872881		
countryAGO	0.977911		
countryAIA	0.999862		
countryALB	0.981432		
countryAND	0.977540		
countryARE	0.977118		
countryARG	0.981266		
countryARM	0.979486		
countryASM	0.999938		
countryATA	0.998871		
countryATF	0.999697		
countryAUS	0.981354		
countryAUT	0.981859		
countryAZE	0.979733		
countryBDI	0.998330		
countryBEL	0.981786		
countryBEN	0.965897		
countryBFA	0.999476		
countryBGD	0.977286		
countryBGR	0.982226		
countryBHR	0.978898		
countryBHS	0.998780		
countryBIH	0.980950		
countryBLR	0.979907		
countryBOL	0.999542		
countryBRA	0.980271		
countryBRB	0.998951		
countryBWA	0.999874		
countryCAF	0.999423		
countryCHE	0.981385		
countryCHL	0.981152		
countryCHN	0.979553		
countryCIV	0.979608		
countryCMR	0.999693		
countryCN	0.981685		
countryCOL	0.980849		
countryCOM	0.999950		
countryCPV	0.978756		
countryCRI	0.983418		
countryCUB	0.998411		
countryCYM	0.999837		
countryCYP	0.981165		
countryCZE	0.981413		
countryDEU	0.982147		
countryDJI	0.999832		
countryDNK	0.981551		
countryDOM	0.979978		
countryDZA	0.980744		
countryECU	0.980051		
countryEGY	0.979793		
countryESP	0.980877		
countryEST	0.981621		

countryETH	0.979724
countryFIN	0.982036
countryFRA	0.981608
countryFRO	0.980076
countryGAB	0.979188
countryGBR	0.981657
countryGEO	0.978755
countryGGY	0.966549
countryGHA	0.979051
countryGIB	0.979026
countryGLP	0.967317
countryGNB	0.980412
countryGRC	0.980891
countryGTM	0.999312
countryGUY	0.999166
countryHKG	0.976836
countryHND	0.970923
countryHRV	0.981065
countryHUN	0.980847
countryIDN	0.977688
countryIMN	0.971640
countryIND	0.981438
countryIRL	0.981224
countryIRN	0.980665
countryIRQ	0.998505
countryISL	0.983017
countryISR	0.980988
countryITA	0.980350
countryJAM	0.999361
countryJEY	0.964201
countryJOR	0.982382
countryJPN	0.982303
countryKAZ	0.980599
countryKEN	0.999216
countryKHM	0.966638
countryKNA	0.998849
countryKOR	0.980120
countryKWT	0.980775
countryLAO	0.999813
countryLBN	0.980454
countryLBY	0.998445
countryLCA	0.999541
countryLIE	0.999487
countryLKA	0.998875
countryLTU	0.982278
countryLUX	0.980481
countryLVA	0.981983
countryMAC	0.975979
countryMAR	0.979734
countryMCO	0.980983
countryMDG	0.997678
countryMDV	0.977493
countryMEX	0.982892
countryMKD	0.981748
countryMLI	0.999473
countryMLT	0.980294
countryMMR	0.999696
countryMNE	0.980645
countryMOZ	0.981251
countryMRT	0.999614
countryMUS	0.981115

countryMWI	0.999372
countryMYS	0.981964
countryMYT	0.968044
countryNGA	0.978250
countryNLD	0.981806
countryNOR	0.981125
countryNPL	0.999476
countryNULL	0.980816
countryNZL	0.983610
countryOMN	0.980881
countryPAK	0.978751
countryPAN	0.999630
countryPER	0.980372
countryPHL	0.978492
countryPOL	0.981611
countryPRI	0.981146
countryPRT	0.979111
countryPRY	0.999721
countryPYF	0.999132
countryQAT	0.977708
countryROU	0.981183
countryRUS	0.980202
countryRWA	0.999183
countrySAU	0.978345
countrySDN	0.998052
countrySEN	0.978234
countrySGP	0.979321
countrySLE	0.999905
countrySLV	0.999999
countrySMR	0.998663
countrySRB	0.983921
countrySTP	0.997927
countrySUR	0.999573
countrySVK	0.980716
country SVN	0.981057
countrySWE	0.981648
countrySYC	0.978837
countrySYR	0.998957
countryTGO	0.999409
countryTHA	0.980092
countryTJK	0.962358
countryTMP	0.979643
countryTUN	0.979133
countryTUR	0.979825
countryTWN	0.980853
countryTZA	0.980248
countryUGA	0.999515
countryUKR	0.980271
countryUMI	0.971373
countryURY	0.980880
countryUSA	0.981159
countryUZB	0.978960
countryVEN	0.979126
countryVNM	0.980479
countryZAF	0.980226
countryZMB	0.979686
countryZWE	0.978049
lead_time	< 2e-16 ***
market_segmentComplementary	2.02e-05 ***
market_segmentCorporate	4.52e-08 ***
market_segmentDirect	3.88e-07 ***

```

market_segmentGroups      0.011708 *
market_segmentOffline TA/T0 1.04e-05 ***
market_segmentOnline TA     0.000602 ***
market_segmentUndefined    0.969474
adr                      < 2e-16 ***
---
Signif. codes:  0 '****' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 125869  on 95511  degrees of freedom
Residual deviance: 86109  on 95331  degrees of freedom
AIC: 86471

Number of Fisher Scoring iterations: 12

> rl3 <- update(rl2, ~. -country)
> summary(rl3)

Call:
glm(formula = is_canceled ~ deposit_type + lead_time + market_segment +
    adr, family = binomial, data = train)

Deviance Residuals:
    Min      1Q      Median      3Q      Max
-3.3501 -0.8429 -0.5397  0.1737  2.2036

Coefficients:
              Estimate Std. Error z value
(Intercept) -1.540e+00 1.724e-01 -8.932
deposit_typeNon Refund 6.123e+00 1.186e-01 51.631
deposit_typeRefundable -2.658e-01 2.067e-01 -1.286
lead_time      3.925e-03 8.776e-05 44.724
market_segmentComplementary -4.413e-01 2.117e-01 -2.084
market_segmentCorporate -6.177e-01 1.779e-01 -3.472
market_segmentDirect -7.840e-01 1.740e-01 -4.507
market_segmentGroups -2.633e-01 1.739e-01 -1.514
market_segmentOffline TA/T0 -7.958e-01 1.734e-01 -4.591
market_segmentOnline TA     3.001e-01 1.720e-01  1.745
market_segmentUndefined  1.106e+01 5.124e+01  0.216
adr                  2.958e-03 1.710e-04 17.295
Pr(>|z|)
(Intercept) < 2e-16 ***
deposit_typeNon Refund < 2e-16 ***
deposit_typeRefundable 0.198390
lead_time < 2e-16 ***
market_segmentComplementary 0.037156 *
market_segmentCorporate 0.000517 ***
market_segmentDirect 6.59e-06 ***
market_segmentGroups 0.130061
market_segmentOffline TA/T0 4.41e-06 ***
market_segmentOnline TA 0.081027 .
market_segmentUndefined 0.829173
adr < 2e-16 ***
---
Signif. codes:  0 '****' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 125869  on 95511  degrees of freedom
Residual deviance: 94305  on 95500  degrees of freedom

```

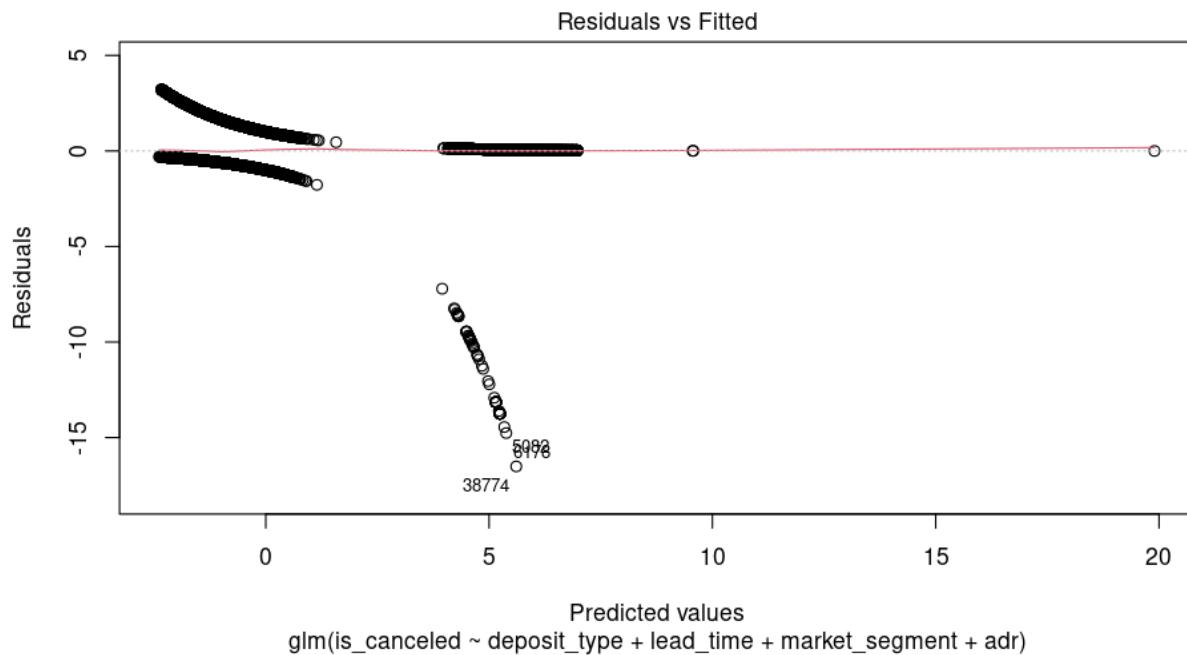
```
AIC: 94329
```

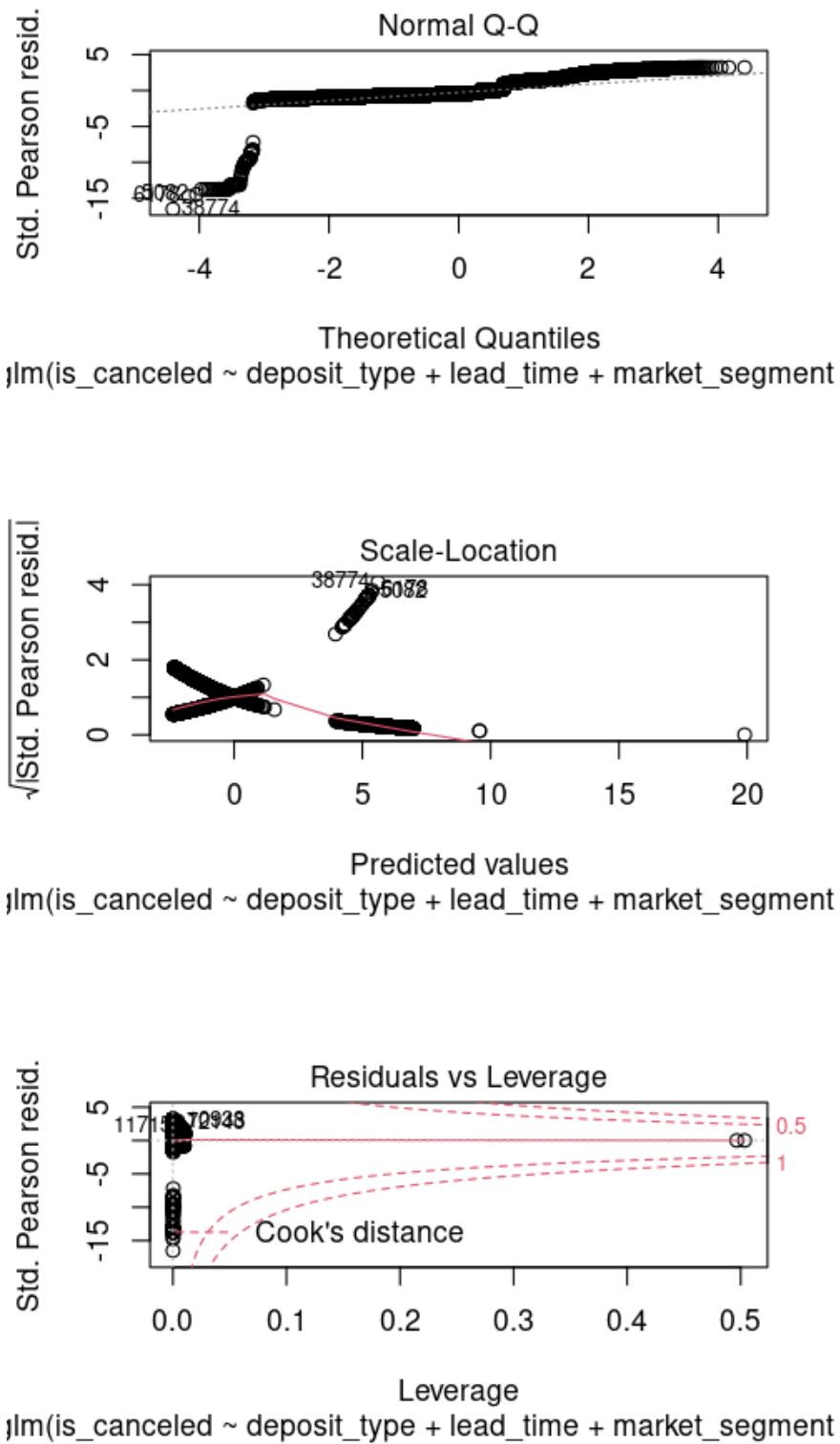
```
Number of Fisher Scoring iterations: 8
```

De acuerdo al modelo de regresión logística las variables que tienen mayor significancia de acuerdo al valor de P son:

```
deposit_typeNon Refund      < 2e-16 ***
deposit_typeRefundable     0.198390
lead_time                  < 2e-16 ***
market_segmentComplementary 0.037156 *
market_segmentCorporate    0.000517 ***
market_segmentDirect        6.59e-06 ***
market_segmentOffline TA/TO 4.41e-06 ***
adr                         < 2e-16 **
```

Se usará deposit_type, lead_time, market_segment y adr para ex





SVM

```

## Preparar los datos
hotel_df <- hotel_stays%>%
  select(is_canceled, hotel, arrival_date_month, meal,
         adr, deposit_type, lead_time, adults, required_car_parking_spaces,
         total_of_special_requests, market_segment,
         stays_in_week_nights, stays_in_weekend_nights)%>%
  mutate_if(is.character,factor)

#Instalar biblioteca
install.packages("tidymodels")
library(tidymodels)

#Crear semilla, data train - test y normalizar los datos
hotel_rec <- recipe(is_canceled ~., data = hotel_train) %>%
  step_dummy(all_nominal(), -all_outcomes()) %>%
  step_zv(all_numeric())%>%
  step_normalize(all_numeric()) %>%
  prep()

#data para crear modelo
test_proc <-bake(hotel_rec, new_data = hotel_test)
hotel_rec_juice <- juice(hotel_rec)

#svm
best6 <- svm(is_canceled ~ ., data = juice(hotel_rec),
              kernel = "radial")
best6
summary(best6)

#matriz de confusión train
table(juice(hotel_rec)$is_canceled, fitted(best6), dnn = c("Actual", "Predicho"))
pred <- predict(best6, juice(hotel_rec))

#matriz de confusión test
mac <- table(true = test_proc$is_canceled,
             pred = predict(best6,
                            newdata = test_proc))
mac

#accuracy
round(sum(diag(mac))/sum(colSums(mac)), 5)

```

Summary

```

> best6

Call:
svm(formula = is_canceled ~ ., data = juice(hotel_rec),
     kernel = "radial")

Parameters:
  SVM-Type: C-classification
  SVM-Kernel: radial
  cost: 1

```

```

Number of Support Vectors: 40492

> str(best6)
List of 30
 $ call      : language svm(formula = is_canceled ~ ., data = juice(hotel_rec), kernel = "radial")
 $ type      : num 0
 $ kernel    : num 2
 $ cost      : num 1
 $ degree    : num 3
 $ gamma     : num 0.0312
 $ coef0     : num 0
 $ nu        : num 0.5
 $ epsilon   : num 0.1
 $ sparse    : logi FALSE
 $ scaled    : logi [1:32] TRUE TRUE TRUE TRUE TRUE TRUE ...
 $ x.scale   :List of 2

```

Matriz de confusión

```

#train
> table(juice(hotel_rec)$is_canceled, fitted(best6), dnn = c("Actual", "Predicho"))
      Predicho
Actual      0      1
  0 52817 3555
  1 13773 19397

#Matriz de confusión con la base test
> mac <- table(true = test_proc$is_canceled,
+                 pred = predict(best6,
+                               newdata = test_proc))
> mac
      pred
true      0      1
  0 17552 1242
  1  4622 6432

> #accuracy
> round(sum(diag(mac))/sum(colSums(mac)), 5)
[1] 0.80354

#Ajustar el modelo
>fit.hotel <- svm(is_canceled ~ ., data = juice(hotel_rec),
+                     kernel = "radial",
+                     decision.values = TRUE)

>summary(fit.hotel)
>str(fit.hotel)

fitted <- attributes(predict(fit.hotel, test_proc,
+                             decision.values = TRUE))$decision.values

#Resultado del modelo ajustado
>mc <- table(true = test_proc,
+             pred = predict(fit.hotel,
+                           newdata = test_proc))

```

```

>mc
pred
true   No  Yes
  No  4163  657
  Yes    82   98

#accuracy
> round(sum(diag(mc))/sum(colSums(mc)), 5)
[1] 0.8522

# Verificar las proporciones
> rs <- apply(mc, 1, sum)
> r1 <- round(mc[1,]/rs[1], 5)
> r2 <- round(mc[2,]/rs[2], 5)
> rbind(No=r1, Yes=r2)
      No     Yes
No  0.86369 0.13631
Yes 0.45556 0.54444

```

El modelo tiene un accuracy del 80.35%, solo el 19.64% no quedo correctamente clasificado. Con el ajuste el accuracy llega a un 85%.

Weighted k-Nearest Neighbor Classifier

```

#Instalar bibliotecas
install.packages("kknn")
library(kknn)

#Ejecutar modelo
knn_spec <- nearest_neighbor() %>%
  set_engine("kknn") %>%
  set_mode("classification")

knn_fit <- knn_spec %>%
  fit(is_canceled~.,
       data=juice(hotel_rec))

#summary
> summary(knn_fit)
      Length Class           Mode
lvl        2   -none-         character
spec       5   nearest_neighbor  list
fit       10   train.kknn      list
preproc    1   -none-         list
elapsed    1   -none-         list
> knn_fit
parsnip model object

Call:
kknn::train.kknn(formula = is_canceled ~ ., data = data, ks = min_rows(5,      data, 5))

Type of response variable: nominal
Minimal misclassification: 0.1929821
Best kernel: optimal

```

```
Best k: 5

#evaluar knn
set.seed(1234)
validation_splits <- mc_cv(juice(hotel_rec), prop =0.9,
                           strata = is_canceled)
validation_splits

knn_res <- fit_resamples(
  knn_spec,
  is_canceled ~.,
  validation_splits,
  control = control_resamples(save_pred = TRUE)
)
```

Conclusiones

- En este proyecto la finalidad es predecir si se cancelará o no la reservación para ello se utilizaron los métodos random forest, regresión logística y SVM.
- Del random forest se obtuvieron las variables que tienen mayor significancia para el objetivo,
- Una vez detectadas, con la regresión logística se midió su nivel de significancia.
- Con el modelo SVM se realizó la clasificación de las variables.
- Se sugiere hacer tuning del SVM para modelos futuros, así como ajuste en el random forest.
- Se sugiere estudiar el data con weighted k-nearest neighbor classifier.