IST 687 M002

GROUP 1

PROJECT REPORT

HOTEL CANCELATION INSIGHTS, ANALYSIS AND RECOMMENDATIONS

Team Members: Chaitanya Kunapareddi, Chaithra Kopparam Cheluvaiah, Ruchak Nilesh Vira, Shourya Goel, Vinit Shambhu Horakeri

In this project we are going to analyze the hotel cancellation dataset and understand the patterns and derive actionable insights via visualization and model the data to predict the cancellations.

Let's start this approach in steps.

Data Reading:

The data received is a url
["https://intro-datascience.s3.us-east-2.amazonaws.com/Resort01.csv"]
Reading this csv file via url to access the data and saving in a dataframe to keep the data in structured format.

```
```{r}
hotelData <- read_csv("https://intro-datascience.s3.us-east-2.amazonaws.com/Resort01.csv")
```
```

Here we read the data using read_csv function.

	IsCanceled	LeadTime	StaysInWeekendNights	StaysInWeekNights	Adults	Children	Babies	Meal	Count
1	0	342	0	0	2	0	0	BB	PRT
2	0	737	0	0	2	0	0	BB	PRT
3	0	7	0	1	1	0	0	BB	GBR
4	0	13	0	1	1	0	0	BB	GBR
5	0	14	0	2	2	0	0	BB	GBR
6	0	14	0	2	2	0	0	BB	GBR
7	0	0	0	2	2	0	0	BB	PRT
8	0	9	0	2	2	0	0	FB	PRT
9	1	85	0	3	2	0	0	BB	PRT
10	1	75	0	3	2	0	0	HB	PRT
11	1	23	0	4	2	0	0	BB	PRT
12	0	35	0	4	2	0	0	HB	PRT
13	0	68	0	4	2	0	0	BB	USA
14	0	18	0	4	2	1	0	HB	ESP
15	0	37	0	4	2	0	0	BB	PRT
16	0	68	0	4	2	0	0	BB	IRL
17	0	37	0	4	2	0	0	BB	PRT
18	0	12	0	1	2	0	0	BB	IRL
19	0	0	0	1	2	0	0	BB	FRA
20	0	7	0	4	2	0	0	BB	GBR
21	0	37	1	4	1	0	0	BB	GBR
22	0	72	2	4	2	0	0	BB	PRT
23	0	72	2	4	2	0	0	BB	PRT
24	0	72	2	4	2	0	0	BB	PRT
25	0	127	2	5	2	0	0	HB	GBR
26	0	78	2	5	2	0	0	BB	PRT

Showing 1 to 25 of 40,060 entries, 20 total columns

This is how the data looks like when viewed.
It has 40,060 instances recorded and 20 attributes for each instance.

The *IsCanceled* column that tells us if there is any cancellation. This is our independent variable.
LeadTime is an int which describes the number of days that elapsed between the entering date of the booking into and the arrival date.

StaysInWeekendNights: Number of weekend nights (Saturday or Sunday) the guest booked the hotel for.
StaysInWeekNights: Number of week nights (Monday to Friday) the guests booked the hotel for.
Adults: Number of adult guests
Children: Number of children guests
Babies: Number of babies guests
Meal: Type of meal booked.
Country: Country of origin.
MarketSegment: Market segment designation.
IsRepeatedGuest: categorical Value indicating if the booking name was from a repeated guest.
PreviousCancellations: Number of previous bookings that were cancelled by the customer prior to the current booking.
PreviousBookingsNotCanceled: Number of previous bookings not cancelled by the customer prior to the current booking
ReservedRoomType: Code of room type reserved.
AssignedRoomType: Code for the type of room assigned to the booking.
BookingChanges: Number of changes/amendments made to the booking
DepositType: Deposit made by customer to confirm the booking.
CustomerType: customer types that booked the hotel.
RequiredCardParkingSpaces: Number of car parking spaces required by the customer
TotalOfSpecialRequests: Number of special requests made by the customer

Statistical Data Analysis:

Here we are going to derive statistical inferences from the dataset to understand what the data types of the 20 columns are and have a glance at some of the sample values.

````
```{r}
str(hotelData)
```
````

```
spec_tbl_df [40,060 × 20] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
 $ IsCanceled                 : num [1:40060] 0 0 0 0 0 0 0 0 1 1 ...
 $ LeadTime                   : num [1:40060] 342 737 7 13 14 14 0 9 85 75 ...
 $ StaysInWeekendNights       : num [1:40060] 0 0 0 0 0 0 0 0 0 0 ...
 $ StaysInWeekNights          : num [1:40060] 0 0 1 1 2 2 2 2 3 3 ...
 $ Adults                     : num [1:40060] 2 2 1 1 2 2 2 2 2 2 ...
 $ Children                   : num [1:40060] 0 0 0 0 0 0 0 0 0 0 ...
 $ Babies                     : num [1:40060] 0 0 0 0 0 0 0 0 0 0 ...
 $ Meal                       : chr [1:40060] "BB" "BB" "BB" "BB" ...
 $ Country                    : chr [1:40060] "PRT" "PRT" "GBR" "GBR" ...
 $ MarketSegment              : chr [1:40060] "Direct" "Direct" "Direct" "Corporate" ...
 $ IsRepeatedGuest            : num [1:40060] 0 0 0 0 0 0 0 0 0 0 ...
 $ PreviousCancellations      : num [1:40060] 0 0 0 0 0 0 0 0 0 0 ...
 $ PreviousBookingsNotCanceled: num [1:40060] 0 0 0 0 0 0 0 0 0 0 ...
 $ ReservedRoomType           : chr [1:40060] "C" "C" "A" "A" ...
 $ AssignedRoomType           : chr [1:40060] "C" "C" "C" "A" ...
 $ BookingChanges             : num [1:40060] 3 4 0 0 0 0 0 0 0 0 ...
 $ DepositType                : chr [1:40060] "No Deposit" "No Deposit" "No Deposit" "No Deposit" ...
 $ CustomerType               : chr [1:40060] "Transient" "Transient" "Transient" "Transient" ...
 $ RequiredCarParkingSpaces   : num [1:40060] 0 0 0 0 0 0 0 0 0 0 ...
 $ TotalOfSpecialRequests     : num [1:40060] 0 0 0 0 1 1 0 1 1 0 ...

 - attr(*, "spec")=
  .. cols(
  ..   IsCanceled = col_double(),
  ..   LeadTime = col_double(),
  ..   StaysInWeekendNights = col_double(),
  ..   StaysInWeekNights = col_double(),
  ..   Adults = col_double(),
  ..   Children = col_double(),
  ..   Babies = col_double(),
  ..   Meal = col_character(),
  ..   Country = col_character(),
  ..   MarketSegment = col_character(),
  ..   IsRepeatedGuest = col_double(),
  ..   PreviousCancellations = col_double(),
  ..   PreviousBookingsNotCanceled = col_double(),
  ..   ReservedRoomType = col_character(),
  ..   AssignedRoomType = col_character(),
  ..   BookingChanges = col_double(),
  ..   DepositType = col_character(),
  ..   CustomerType = col_character(),
  ..   RequiredCarParkingSpaces = col_double(),
  ..   TotalOfSpecialRequests = col_double()
  .. )
```

As we see, there are 13 numerical columns (consisting of numbers) of which 12 are categorical (the col_double indicates it's a factor) , so the numerics placed in these columns are factors for values.

There are 7-character columns which are strings.

```{r}
summary(hotelData)
```

```
  IsCanceled       LeadTime      StaysInWeekendNights StaysInWeekNights     Adults         Children      
 Min.   :0.0000   Min.   :  0.00   Min.   : 0.00        Min.   : 0.000    Min.   : 0.000   Min.   : 0.0000  
 1st Qu.:0.0000   1st Qu.: 10.00   1st Qu.: 0.00        1st Qu.: 1.000    1st Qu.: 2.000   1st Qu.: 0.0000  
 Median :0.0000   Median : 57.00   Median : 1.00        Median : 3.000    Median : 2.000   Median : 0.0000  
 Mean   :0.2776   Mean   : 92.68   Mean   : 1.19        Mean   : 3.129    Mean   : 1.867   Mean   : 0.1287  
 3rd Qu.:1.0000   3rd Qu.:155.00   3rd Qu.: 2.00        3rd Qu.: 5.000    3rd Qu.: 2.000   3rd Qu.: 0.0000  
 Max.   :1.0000   Max.   :737.00   Max.   :19.00        Max.   :50.000    Max.   :55.000   Max.   :10.0000  
     Babies            Meal              Country          MarketSegment       IsRepeatedGuest   PreviousCancellations
 Min.   :0.0000   Length:40060       Length:40060       Length:40060       Min.   :0.00000   Min.   : 0.0000      
 1st Qu.:0.0000   Class :character   Class :character   Class :character   1st Qu.:0.00000   1st Qu.: 0.0000      
 Median :0.0000   Mode  :character   Mode  :character   Mode  :character   Median :0.00000   Median : 0.0000      
 Mean   :0.0139                                                            Mean   :0.04438   Mean   : 0.1017      
 3rd Qu.:0.0000                                                            3rd Qu.:0.00000   3rd Qu.: 0.0000      
 Max.   :2.0000                                                            Max.   :1.00000   Max.   :26.0000      
 PreviousBookingsNotCanceled ReservedRoomType   AssignedRoomType   BookingChanges    DepositType       
 Min.   : 0.0000             Length:40060       Length:40060       Min.   : 0.000   Length:40060      
 1st Qu.: 0.0000             Class :character   Class :character   1st Qu.: 0.000   Class :character  
 Median : 0.0000             Mode  :character   Mode  :character   Median : 0.000   Mode  :character  
 Mean   : 0.1465                                                   Mean   : 0.288                     
 3rd Qu.: 0.0000                                                   3rd Qu.: 0.000                     
 Max.   :30.0000                                                   Max.   :17.000                     
 CustomerType       RequiredCarParkingSpaces TotalOfSpecialRequests
 Length:40060       Min.   :0.0000           Min.   :0.0000        
 Class :character   1st Qu.:0.0000           1st Qu.:0.0000        
 Mode  :character   Median :0.0000           Median :0.0000        
                    Mean   :0.1381           Mean   :0.6198        
                    3rd Qu.:0.0000           3rd Qu.:1.0000        
                    Max.   :8.0000           Max.   :5.0000        
```

On exploring the statistical summary (generated for numeric columns) we generate the following inferences for the columns:

1. IsCanceled – ranges either 0 or 1 and most values are dominated by 0.
2. Leadtime – averages to 92.68 i.e., customers book at least 3 months ahead of arrival.
3. StaysInWeekNights – usually the customers stay for a couple of days but there are rare cases where some have stayed for up to more than 2 weeks.

Analyzing columns:

Although we have understood how the values are spread numerically, we must apply some exploratory data analysis to visually see the spread. In this analysis we will detect the null values, we will check for any strange patterns or outliers in the data and see if the data is too biased to mis lead our model.

Firstly, we shall explore all records of the column and check if there are any null or flagged null values. This way we can come to conclusion as to what preprocessing shall be done.

Exploring the columns data

Producing tables of categorical variables

```{r}
print('count for cancelled data')
table(hotelData$IsCanceled)
print('count for Mean data')
table(hotelData$Meal)
print('count for country data')
table(hotelData$Country)
print('count for market segment data')
table(hotelData$MarketSegment)
print('count for repeated guest data')
table(hotelData$IsRepeatedGuest)
print('count for reserved room type data')
table(hotelData$ReservedRoomType)
print('count for assigned room data')
table(hotelData$AssignedRoomType)
print('count for deposit data')
table(hotelData$DepositType)
```

```
[1] "count for market segment data"

Complementary     Corporate        Direct        Groups Offline TA/TO     Online TA
          201          2309          6513          5836          7472         17729
[1] "count for repeated guest data"

    0     1
38282  1778
[1] "count for reserved room type data"

    A     B     C     D     E     F     G     H     L     P
23399     3   918  7433  4982  1106  1610   601     6     2
[1] "count for assigned room data"

    A     B     C     D     E     F     G     H     I     L     P
17046   159  2214 10339  5638  1733  1853   712   363     1     2
[1] "count for deposit data"

No Deposit Non Refund Refundable
     38199       1719        142
```

```
[1] "count for cancelled data"

    0     1
28938 11122
[1] "count for Mean data"

       BB        FB        HB        SC Undefined
    30005       754      8046        86      1169
[1] "count for country data"

  AGO   ALB   AND   ARE   ARG   ARM   AUS   AUT   AZE   BDI   BEL   BGR   BHR   BHS   BIH   BLR   BRA   BWA   CAF
   24     3     5    11    57     2    87   210     3     1   448     5     1     1     1     7   430     1     3
  CHE   CHL   CHN   CIV   CMR    CN   COL   COM   CPV   CRI   CUB   CYM   CYP   CZE   DEU   DJI   DNK   DOM   DZA
  435    17   134     2     2   710    16     1     5     2     4     1     8    27  1203     1    65     3    12
  ECU   EGY   ESP   EST   FIN   FJI   FRA   GBR   GEO   GGY   GIB   GRC   HKG   HRV   HUN   IDN   IND   IRL   IRN
    2     1  3957    33   151     1  1611  6814    11     1    13    10     4    11    47     5    37  2166     5
  ISL   ISR   ITA   JAM   JEY   JOR   JPN   KAZ   KOR   KWT   LBN   LKA   LTU   LUX   LVA   MAC   MAR   MDG   MDV
    6    28   459     5     3     2     9     5     9     3     6     1    46    80    33     1    75     1     6
  MEX   MKD   MLT   MOZ   MUS   MWI   MYS   NGA   NLD   NOR   NPL   NZL   OMN   PAK   PER   PHL   PLW   POL   PRI
    6     1     2     6     1     2    10    10   514   123     1    14    11     4     1    16     1   333     9
  PRT   QAT   ROU   RUS   SAU   SEN   SGP   SMR   SRB   SUR   SVK   SVN   SWE   SYC   SYR   TGO   THA   TUN   TUR
17630     1   177   189     1     1     4     1     7     4    12    11   304     1     1     1     6     1    23
  TWN   UGA   UKR   URY   USA   UZB   VEN   VNM   ZAF   ZMB   ZWE
   12     1    23     8   479     1     3     2    18     1     2
```

As we investigate this result, we see that in the country column the value NULL does not represent any country it is a flagged null value. So, we are unaware as to which country it is representing.

```{r}
sum(is.na(hotelData$IsCanceled))
sum(is.na(hotelData$LeadTime))
sum(is.na(hotelData$StaysInWeekendNights))
sum(is.na(hotelData$StaysInWeekNights))
sum(is.na(hotelData$Adults))
sum(is.na(hotelData$Children))
sum(is.na(hotelData$Babies))
sum(is.na(hotelData$Meal))
sum(is.na(hotelData$Country))
sum(is.na(hotelData$MarketSegment))
sum(is.na(hotelData$IsRepeatedGuest))
sum(is.na(hotelData$PreviousCancellations))
sum(is.na(hotelData$PreviousBookingsNotCanceled))
sum(is.na(hotelData$ReservedRoomType))
sum(is.na(hotelData$AssignedRoomType))
sum(is.na(hotelData$BookingChanges))
sum(is.na(hotelData$DepositType))
sum(is.na(hotelData$CustomerType))
sum(is.na(hotelData$RequiredCarParkingSpaces))
sum(is.na(hotelData$TotalOfSpecialRequests))
```

```
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
[1] 0
```

Also, the value CN must be changed to CAN as it represents the country wrongfully, hence misleading the data.

Let's, explore the results after cleaning the data,

```{r}
table(is.na(hotelData$Country))
```

```
FALSE  TRUE
39596   464
```

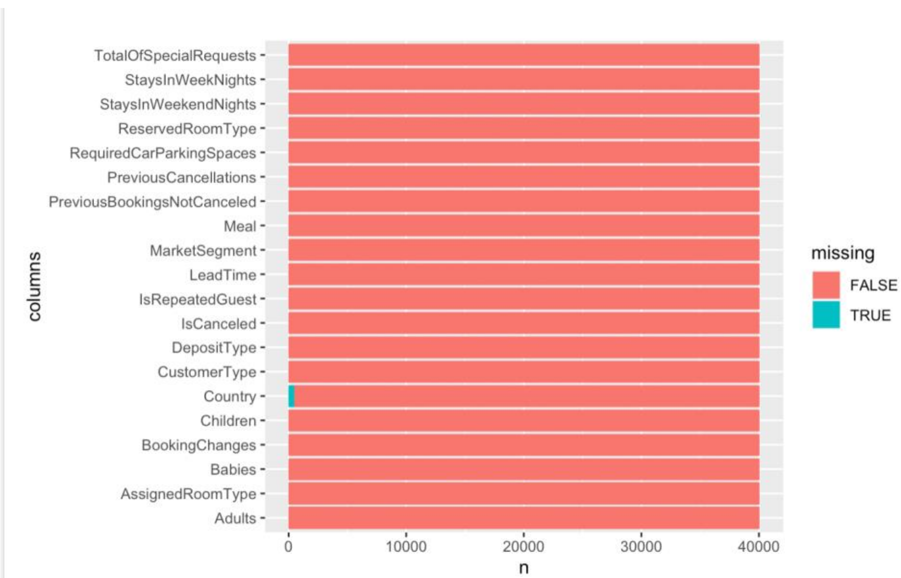
```{r}
hotelData$Country[hotelData$Country=="NULL"] <- NA
hotelData$Country[hotelData$Country=="CN"] <- "CAN"
table(hotelData$Country)
```

```
  AGO   ALB   AND   ARE   ARG   ARM   AUS   AUT   AZE   BDI   BEL   BGR   BHR   BHS   BIH   BLR   BRA 
   24     3     5    11    57     2    87   210     3     1   448     5     1     1     1     7   430 
  BWA   CAF   CAN   CHE   CHL   CHN   CIV   CMR   COL   COM   CPV   CRI   CUB   CYM   CYP   CZE   DEU 
    1     3   710   435    17   134     2     2    16     1     5     2     4     1     8    27  1203 
  DJI   DNK   DOM   DZA   ECU   EGY   ESP   EST   FIN   FJI   FRA   GBR   GEO   GGY   GIB   GRC   HKG 
    1    65     3    12     2     1  3957    33   151     1  1611  6814    11     1    13    10     4 
  HRV   HUN   IDN   IND   IRL   IRN   ISL   ISR   ITA   JAM   JEY   JOR   JPN   KAZ   KOR   KWT   LBN 
   11    47     5    37  2166     5     6    28   459     5     3     2     9     5     9     3     6 
  LKA   LTU   LUX   LVA   MAC   MAR   MDG   MDV   MEX   MKD   MLT   MOZ   MUS   MWI   MYS   NGA   NLD 
    1    46    80    33     1    75     1     6     6     1     2     6     1     2    10    10   514 
  NOR   NPL   NZL   OMN   PAK   PER   PHL   PLW   POL   PRI   PRT   QAT   ROU   RUS   SAU   SEN   SGP 
  123     1    14    11     4     1    16     1   333     9 17630     1   177   189     1     1     4 
  SMR   SRB   SUR   SVK   SVN   SWE   SYC   SYR   TGO   THA   TUN   TUR   TWN   UGA   UKR   URY   USA 
    1     7     4    12    11   304     1     1     1     6     1    23    12     1    23     8   479 
  UZB   VEN   VNM   ZAF   ZMB   ZWE 
    1     3     2    18     1     2 
```

Visualizing the columns we get,

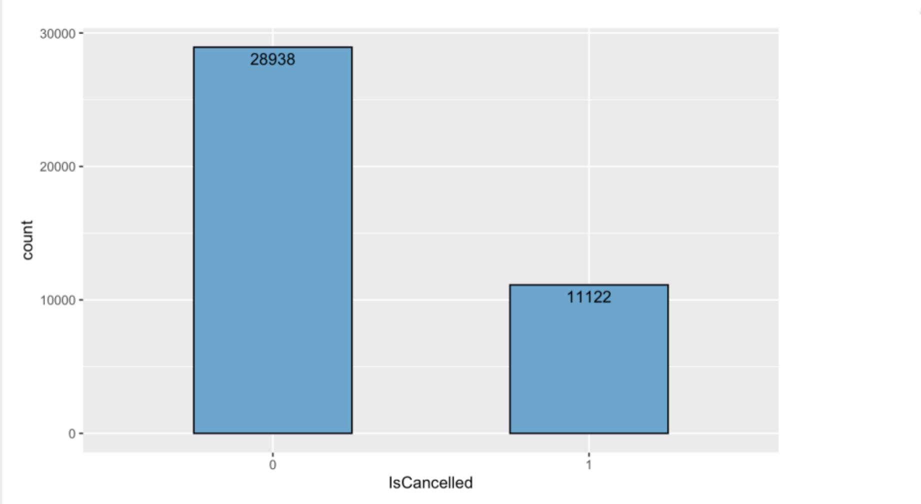
```
hotelData$Country[hotelData$Country=="NULL"] <- NA
#Converting "NULL" in Country to NA

hotelData  %>% summarise_all(list(~is.na(.)))%>% pivot_longer(everything(),names_to = "columns", values_to="missing") %>%
  count(columns, missing) %>%ggplot(aes(y=columns,x=n,fill=missing))+
  geom_col()
```



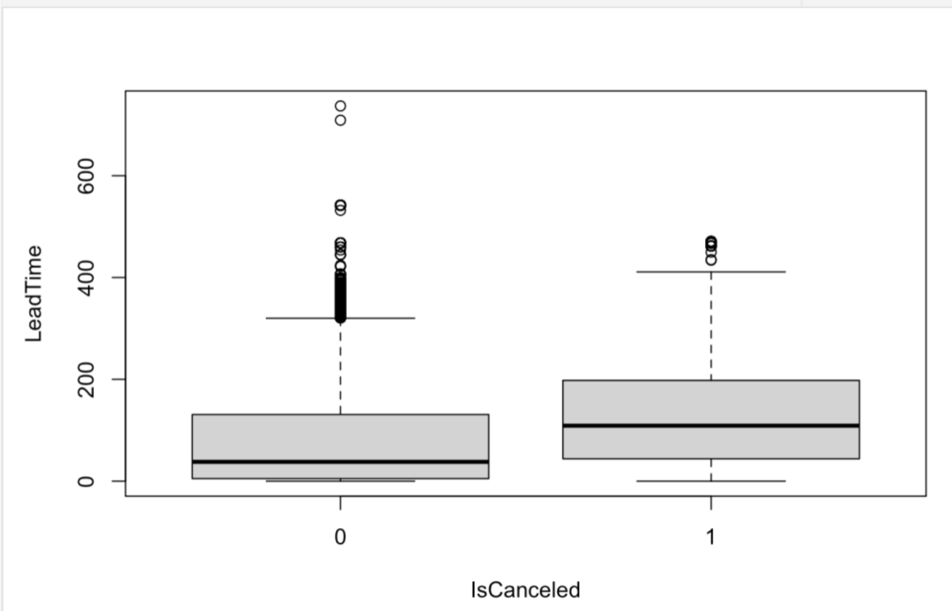
Now when we see the proportion of cancellations for our hotel, there are 27.7% cancellations. with the knowledge of exploratory data analysis let us find out the relation between columns and we might end up having the reasons for the 27%.

```
ggplot(hotelData, aes(x=factor(IsCanceled)))+
geom_bar(fill="skyblue3",color="black", width=0.5)+scale_x_discrete("IsCancelled")+ geom_text(aes(label=..count..),
stat="count", vjust=1.5)
```



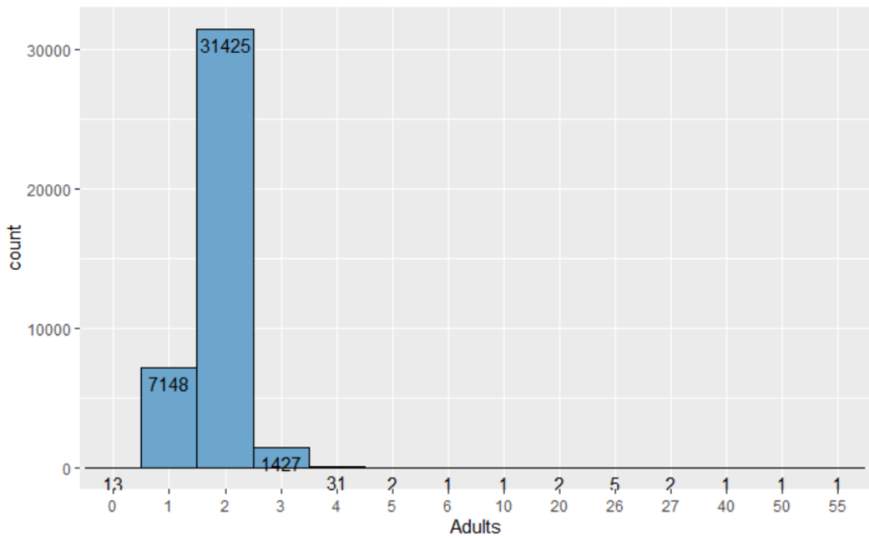
Let's have a look at the lead time column. We have created a box plot to see spread of this data. Most of the people who haven't cancelled their booking have booked it many days ago, I.e., more than a year ago. This exceeds the usual time of booking of 90 days. (Although it's not known in the data, its implicitly being used to measure against.)

```{r}
boxplot(LeadTime~IsCanceled,data=hotelData)
```



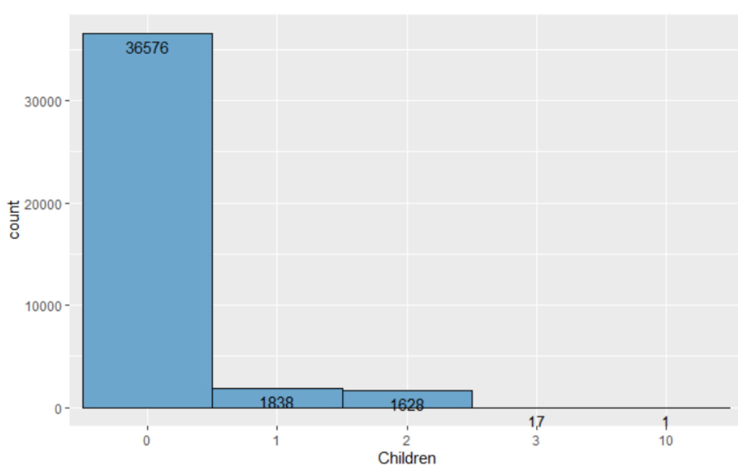
The plot also supports the assumption that keeping the bookings open way too early is bringing out changes in booking like cancellation.

```{r}
ggplot(hotelData, aes(x=factor(Adults)))+geom_bar(fill="skyblue3", color="black", width =
1.0)+scale_x_discrete("Adults")+geom_text(aes(label=..count..),stat="count", vjust=1.5)
```

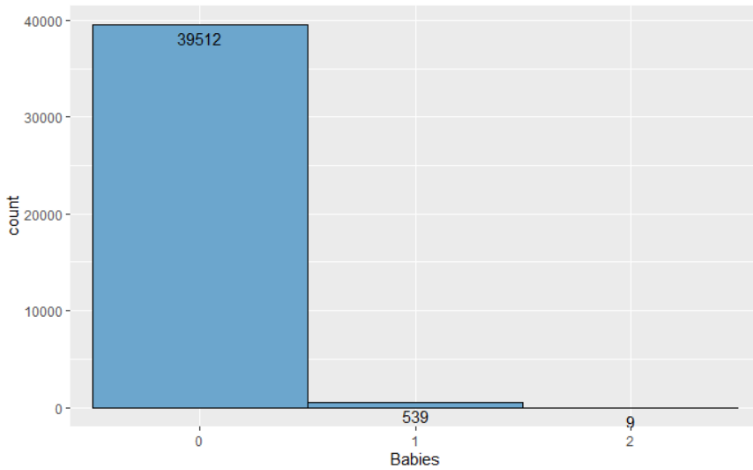


The chart above shows that most of the adults value ranges from 1 to 3 and those above 4 are outliers in the data. Similarly, exploring children and babies column below.

```{r}
ggplot(hotelData, aes(x=factor(Children)))+geom_bar(fill="skyblue3", color="black", width =
1.0)+scale_x_discrete("Children")+geom_text(aes(label=..count..),stat="count", vjust=1.5)
```

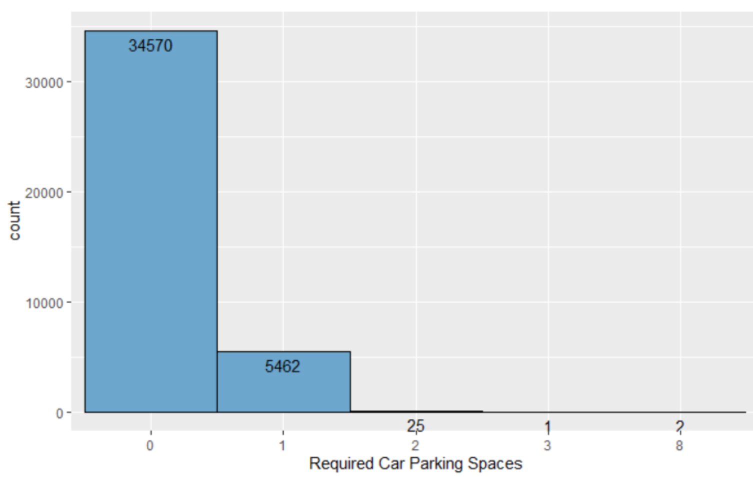


```{r}
ggplot(hotelData, aes(x=factor(Babies)))+geom_bar(fill="skyblue3", color="black", width =
1.0)+scale_x_discrete("Babies")+geom_text(aes(label=..count..),stat="count", vjust=1.5)
```



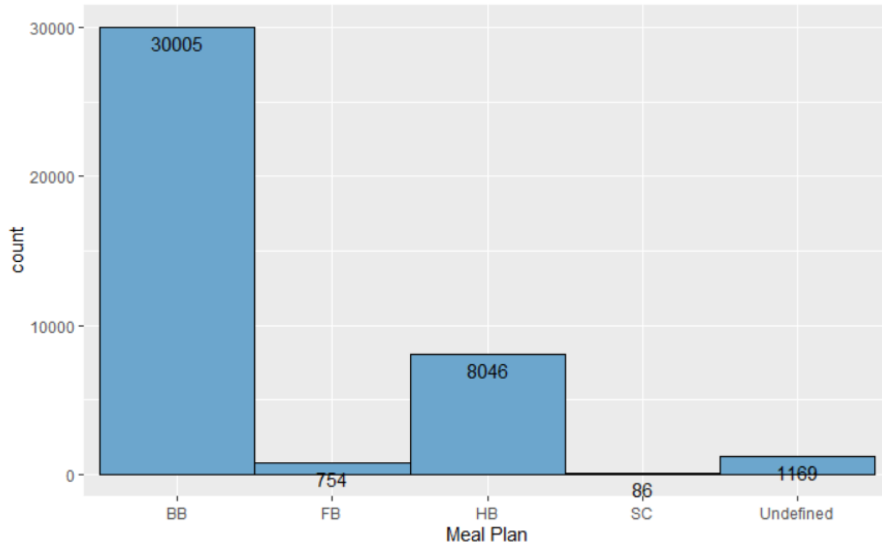
Most of the guests have no babies or children with them and there are hardly a few with 1 –2 of them.

```{r}
ggplot(hotelData, aes(x=factor(RequiredCarParkingSpaces)))+geom_bar(fill="skyblue3", color="black", width =
1.0)+scale_x_discrete("Required Car Parking Spaces")+geom_text(aes(label=..count..),stat="count", vjust=1.5)
```



As we see in the above graph the required car parking spaces column, lot of guests do not actually desire to have one.

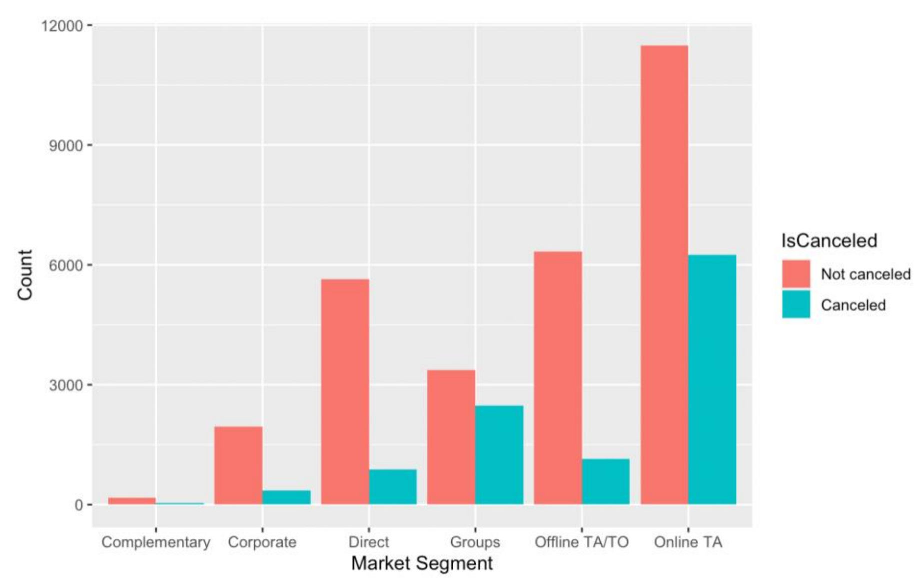
```{r}
ggplot(hotelData, aes(x=factor(Meal)))+geom_bar(fill="skyblue3", color="black", width =
1.0)+scale_x_discrete("Meal Plan")+geom_text(aes(label=..count..),stat="count", vjust=1.5)
```



A lot guests opt for bed and breakfast and Half meal plan so an increase in this plan can yield a good result. Although these are speculation from charts above let's increase the exploration stakes further.

To understand the relationship better we shall use another variable to see the segmentation more clearly.

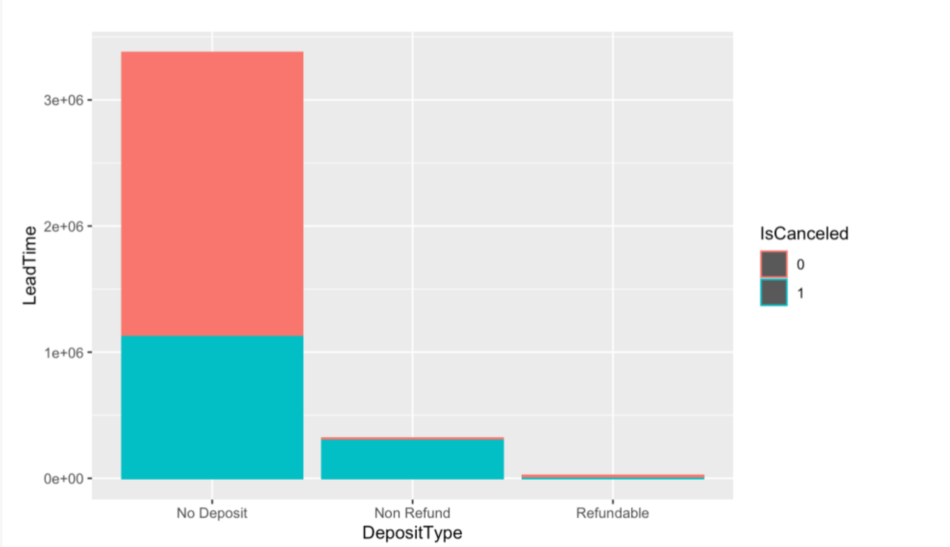
```
ggplot(hotelData, aes(x=factor(MarketSegment), fill=factor(IsCanceled))) +
geom_bar(position="dodge") +labs(x="Market Segment",y="Count")+
scale_fill_discrete(name="IsCanceled", labels=c("Not canceled", "Canceled"))
```



In this chart we have plotted the market segments that are being used by the hotels and the cancellations that are happening in each. We can explicitly state the fact that online TA has more cancellations, it has more non cancellations too. But we must have a look at the proportions of cancellations as it will give us more accurate measure as to which segment needs more attention to.

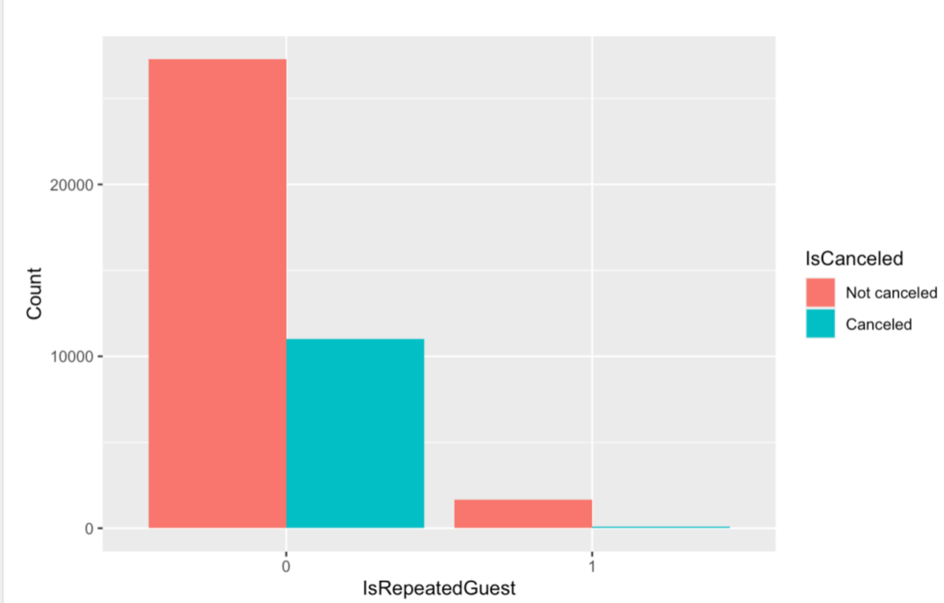
As we notice in the graph the complementary, corporate, direct, and offline TA have very lower rate of cancellations as compared to groups and online TA.

```
hotelData$IsCanceled<-as.factor(hotelData$IsCanceled)
ggplot(data = hotelData, aes(x = DepositType, y = LeadTime, color=IsCanceled)) +geom_bar(stat = "identity")
```



We now increase the analysis parameters to 3. On checking the deposit type against the cancellations and lead time we see that when people are being offered no deposits for their booking there is an increase in bookings where in the other cases there is a higher rate of cancellations. With this we can infer that if there is no booking time window and no deposits being taken during the booking confirmation there is a higher rate of success.

```
ggplot(hotelData, aes(x=factor(IsRepeatedGuest), fill=factor(IsCanceled))) +
geom_bar(position="dodge") +labs(x="IsRepeatedGuest",y="Count")+
scale_fill_discrete(name="IsCanceled", labels=c("Not canceled", "Canceled"))
```



Now let us look at the data in more optimistic manner. Lot of guests who have come to the hotel before have come again, this indicates that the factors that we have discussed earlier might help in validating the results.

Looking at the global performance of the hotel we pick the top 10 countries, and plot their parameters.

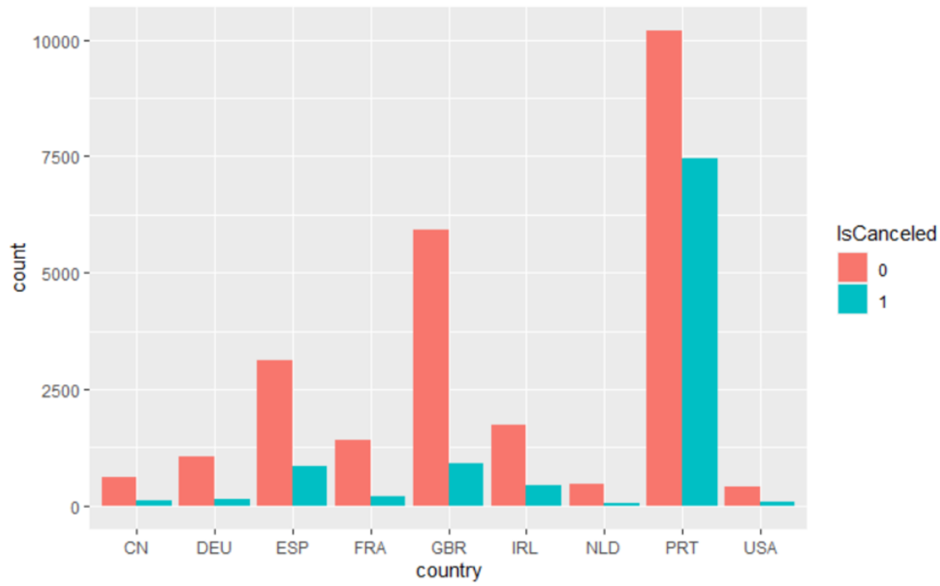
```
ggplot(data = hotelData_top10countries, aes(x = country, fill = IsCanceled)) +
    geom_bar(position = "dodge")
```

```
ggplot(data = hotelData_top10countries, aes(x = country, fill = RequiredCarParkingSpaces)) +
    geom_bar(position = "dodge")
```

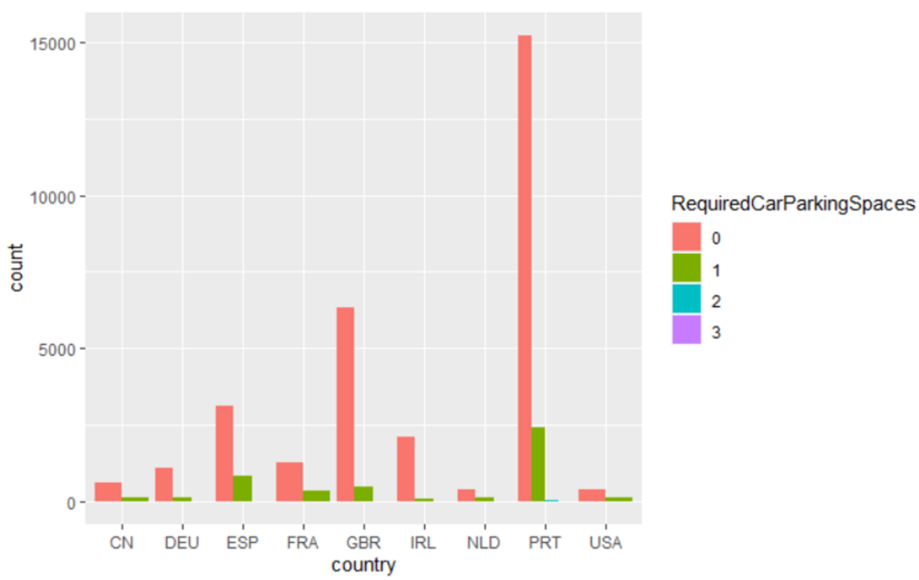
```
ggplot(data = hotelData_top10countries, aes(x = country, fill = MarketSegment)) +
    geom_bar(position = "dodge")
```

```
ggplot(data = hotelData_top10countries, aes(x = country, fill = DepositType)) +
    geom_bar(position = "dodge")
```

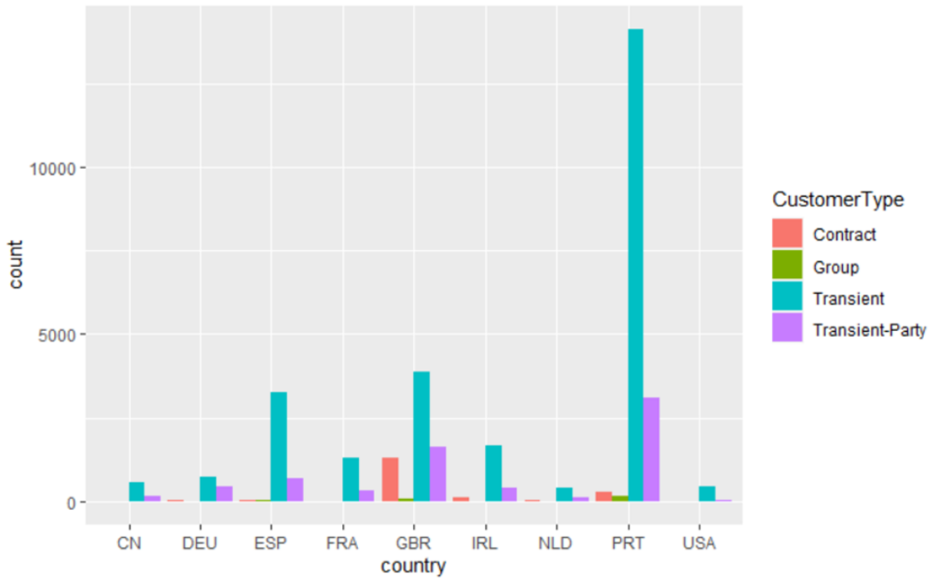
This will yield in the following graphs,



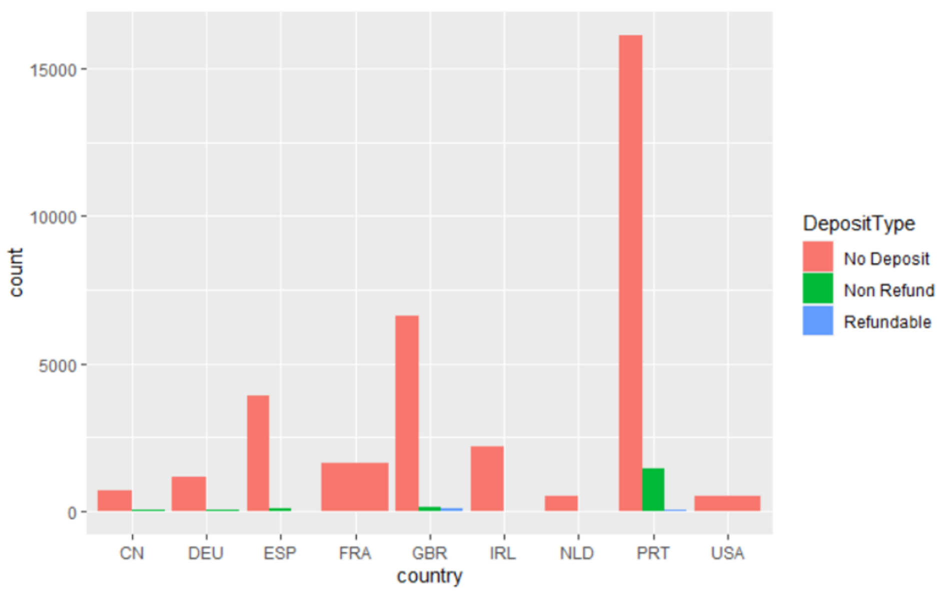
As we see the cancelation rate is higher in Portugal than any other country.



There is no car parking required spaces as well in Portugal.

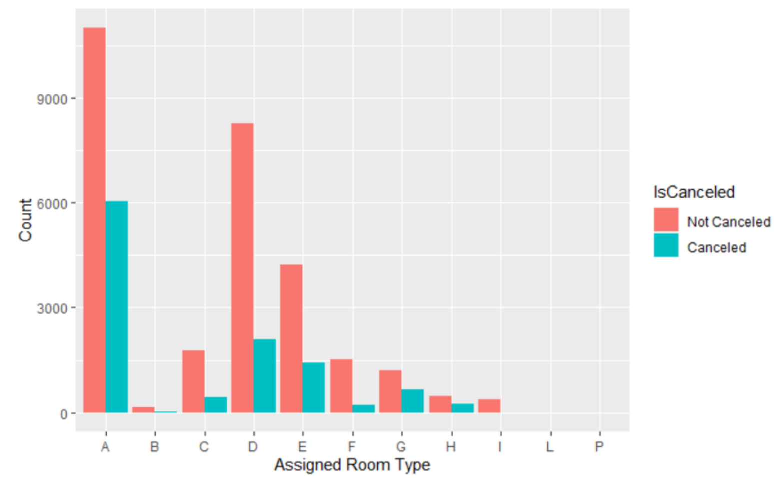


Most of the bookings are transient, that is booked for a brief period or booked just before arrival. This may be the reason there are no car parking spaces.



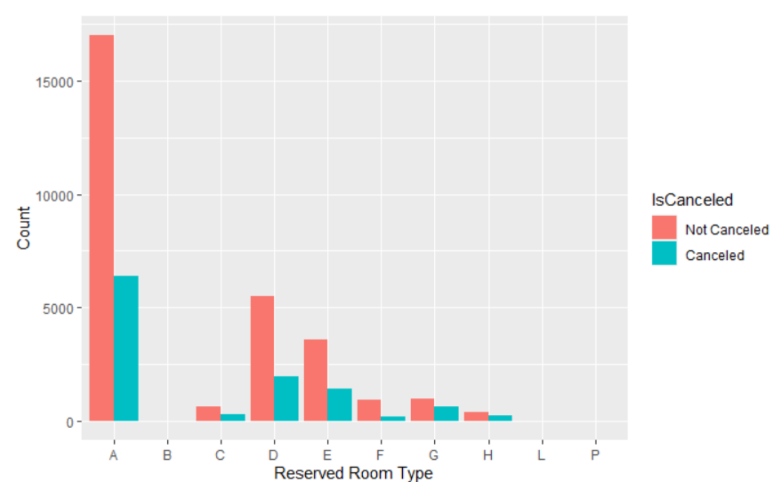
There are no deposits being collected in Portugal, this could lead to random booking and cancelling as there is no loss for customers. If the hotel takes some amount which could be refundable or nonrefundable there could be lower cancellation rates.

```{r}
ggplot(hotelData, aes(x=factor(AssignedRoomType), fill=factor(IsCanceled)))+
  geom_bar(position="dodge")+labs(x="Assigned Room Type", y="Count")+
  scale_fill_discrete(name="IsCanceled",labels=c("Not Canceled","Canceled"))
```



This plot shows the cases where the customers were assigned a room versus their cancellation. Here we try to infer it as a reason for cancellation because some customer would want to get their desired room. we shall have a look at that chart as well.

```{r}
ggplot(hotelData, aes(x=factor(ReservedRoomType), fill=factor(IsCanceled)))+
  geom_bar(position="dodge")+labs(x="Reserved Room Type", y="Count")+
  scale_fill_discrete(name="IsCanceled",labels=c("Not Canceled","Canceled"))
```



As we see, the cancellation rate is much lower, as guests are happy with the desired room.

The assumption may be affected by spurious reasons but the inferences made are helpfully towards data modelling.

Tuning and Mining:

In this section, we are going to apply supervised and unsupervised learning techniques to create a model that can best predict outcomes to bring the best out of the dataset and improve the hotel bookings.

Breaking this process into several steps;

Cleaning the data

The data set contains 40,060 records and should not contain any nulls. In the above chart we noticed that there is a flagged null value for the country which does not add value to the dataset and so it is replaced by na.
Since we can still not move ahead with nulls in the data, we use na.interpolation function to add aggregated results to the column but we must have numericals to do so.

Association Rules mining

As we discussed that the relationship between the column is significantly valuable, we apply association rules to see the best fit of values when appearing together. To do so we convert the attributes to factors first and then create a transaction dataset.

```
#converting data frame to factors
hotelData$IsCanceled <- as.factor(hotelData$IsCanceled)
hotelData$Meal <- as.factor(hotelData$Meal)
hotelData$Country <- as.factor(hotelData$Country)
hotelData$MarketSegment <- as.factor(hotelData$MarketSegment)
hotelData$IsRepeatedGuest <- as.factor(hotelData$IsRepeatedGuest)
hotelData$ReservedRoomType <- as.factor(hotelData$ReservedRoomType)
hotelData$AssignedRoomType <- as.factor(hotelData$AssignedRoomType)
hotelData$DepositType <- as.factor(hotelData$DepositType)
hotelData$CustomerType <- as.factor(hotelData$CustomerType)
```

```
#converting to transactions
hotelDataTransactions <- as(hotelData, 'transactions')
hotelDataTransactions
```

```
transactions in sparse format with
 40060 transactions (rows) and
 186 items (columns)
```

The transactions are a sparse matrix dataset the creates a term document frequency explaining the occurrence of words in every document. This is the reason we have 186 columns.

Now we are good to proceed for creating rules and to predict whether the guest is willing to cancel or not.

```
```{r}
rules1 <- apriori(hotelDataTransactions,
                  parameter=list(supp=0.1, conf=0.55),
                  control=list(verbose=F),
                  appearance=list(default="lhs",rhs=("IsCanceled=1")))
```

Support is the proportion of times that a particular set of items occurs relative to the whole dataset. We set this parameter to 0.1.
```

Confidence is proportion of times that the consequent occurs when the antecedent is present and we set this parameter to 0.55.
The RHS is checked for both cancelation types to give best lift values.

Now the rules are ready, so let us inspect them.

```
inspect(rules1)
```

This yields in the below mentioned result.

```
```{r}
inspect(top10Lift)
```

```
     lhs                                        rhs              support confidence  coverage     lift count
[1]  {Adults=[2,55],
      Country=PRT,
      IsRepeatedGuest=0,
      TotalOfSpecialRequests=[0,1)}          => {IsCanceled=1} 0.1075387  0.5966759 0.1802297 2.149149  4308
[2]  {Adults=[2,55],
      Children=[0,10],
      Country=PRT,
      IsRepeatedGuest=0,
      TotalOfSpecialRequests=[0,1)}          => {IsCanceled=1} 0.1075387  0.5966759 0.1802297 2.149149  4308
[3]  {Adults=[2,55],
      Babies=[0,2],
      Country=PRT,
      IsRepeatedGuest=0,
      TotalOfSpecialRequests=[0,1)}          => {IsCanceled=1} 0.1075387  0.5966759 0.1802297 2.149149  4308
[4]  {Adults=[2,55],
      Country=PRT,
      IsRepeatedGuest=0,
      PreviousCancellations=[0,26],
      TotalOfSpecialRequests=[0,1)}          => {IsCanceled=1} 0.1075387  0.5966759 0.1802297 2.149149  4308
[5]  {Adults=[2,55],
      Country=PRT,
      IsRepeatedGuest=0,
      PreviousBookingsNotCanceled=[0,30],
```

Ex: We can state that if adults are greater than 2, the guest is repeated and hotel is in Portugal the higher lift and confidence state that there is more success rate in cancelling than booking.

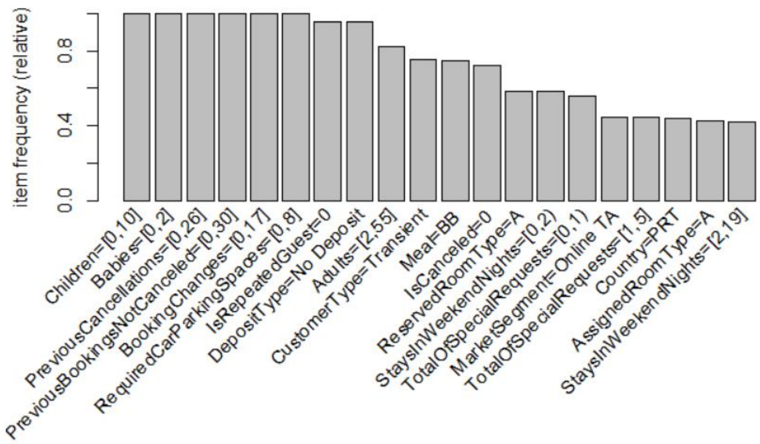
The LHS represents the combination of columns and RHS shows our condition.

```
```{r}
top10Lift <- sort(rules1, decreasing = TRUE, na.last = NA, by = "lift")
top10Lift
```

```
set of 240 rules
```

There is a total of 240 such rules for our dataset.

```{r}
itemFrequencyPlot(hotelDataTransactions, topN=20)
```



This plot shows us the columns that create a significant impact towards our predictions. (Top 20 impactful dependent columns)

Are these all the rules? We shall now change the confidence and support values to create more rules to find the impactful insights for the model.

```{r}
rules2 <- apriori(hotelDataTransactions,
                  parameter=list(supp=0.09, conf=0.50),
                  control=list(verbose=F),
                  appearance=list(default="lhs",rhs=("IsCanceled=1")))
rules2
```

```
set of 997 rules
```

Support is set to 0.09 and confidence to 0.5. this yields in 997rules.

```
top10Lift2 <- sort(rules2, decreasing = TRUE, na.last = NA, by = "lift")
inspect(top10Lift2)
```

```
     lhs                                  rhs            support confidence  coverage     lift count
[1]  {Adults=[2,55],
      Country=PRT,
      IsRepeatedGuest=0,
      ReservedRoomType=A,
      AssignedRoomType=A}              => {IsCanceled=1} 0.09375936  0.6358558 0.1474538 2.290270  3756
[2]  {Adults=[2,55],
      Country=PRT,
      IsRepeatedGuest=0,
      PreviousBookingsNotCanceled=[0,30],
      ReservedRoomType=A,
      AssignedRoomType=A}              => {IsCanceled=1} 0.09375936  0.6358558 0.1474538 2.290270  3756
[3]  {Adults=[2,55],
      Children=[0,10],
      Country=PRT,
      IsRepeatedGuest=0,
      ReservedRoomType=A,
      AssignedRoomType=A}              => {IsCanceled=1} 0.09375936  0.6358558 0.1474538 2.290270  3756
[4]  {Adults=[2,55],
      Babies=[0,2],
      Country=PRT,
      IsRepeatedGuest=0,
      ReservedRoomType=A,
      AssignedRoomType=A}              => {IsCanceled=1} 0.09375936  0.6358558 0.1474538 2.290270  3756
[5]  {Adults=[2,55],
      Country=PRT,
      IsRepeatedGuest=0,
      PreviousCancellations=[0,26],
      ReservedRoomType=A,
      AssignedRoomType=A}              => {IsCanceled=1} 0.09375936  0.6358558 0.1474538 2.290270  3756
```

Ex: the rule states that adult in country Portugal who are new to the hotel & have reserved the room type A and were assigned the same room have cancelled the room. These rules have higher lift and count values which show the likelihood of happening.

Modeling:

Towards the first set of data modelling, we split the data into a training set and testing set. There is no best proportion of splitting the data so we start with 66% as training set an 34% as testing.

```{r}
set.seed(1)
trainList <- createDataPartition(y=hotelData$IsCanceled, p=0.66, list=FALSE)
#creating a trainList of only the IsCanceled column with a partition at the 66%
#mark of the dataset
#str(trainList)
trainSet <- hotelData[trainList,]
#creating a training dataset of xyz records
testSet <- hotelData[-trainList,]
#creating a testing dataset of  xyzrecords
```

```{r}
dim(trainSet)
#verifying that the training dataset has 26441 records
```

```
[1] 26441    20
```

```{r}
dim(testSet)
#verifying that the testing dataset has 13619 records
```

```
[1] 13619    20
```

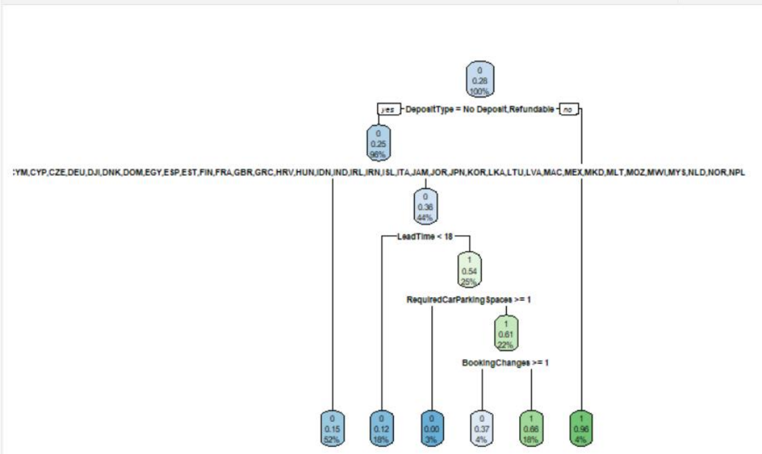
So now we have 26441 records for our training set and 13619 records for our testing set.

Post data partitioning we create our first model, using rpart i.e., decision trees. This algorithm works in such a way that it shows all the decisions like a flow chart and based on the features it decides the best fit way for each instance.

```
```{r}
cartTree1 <- rpart(IsCanceled ~ ., data = trainSet)
```
```

We take all the columns as independent columns and use them to predict them against the isCanceled column. The model is stored in a var. [cartTree1]

```
```{r}
library(rpart.plot)
rpart.plot(cartTree1)
```
```



This plot helps us understand the decision making of the algorithm for the given columns.

```{r}
rpartPred1 <- predict(cartTree1, newdata=testSet, type="class")
#rpartPred stores the prediction of cartTree by using predict() on the testing
#dataset
confusionMatrix(rpartPred1, testSet$IsCanceled)
```

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 8988 1583
         1  850 2198

               Accuracy : 0.8214
                 95% CI : (0.8148, 0.8278)
    No Information Rate : 0.7224
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5263

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9136
            Specificity : 0.5813
         Pos Pred Value : 0.8503
         Neg Pred Value : 0.7211
             Prevalence : 0.7224
         Detection Rate : 0.6600
   Detection Prevalence : 0.7762
      Balanced Accuracy : 0.7475

       'Positive' Class : 0
```

To check the efficiency of this model, we predict it against the test set and create a confusion matrix.
The diagonal values show the true positive values (the one that have been predicted correctly). With **82.144%** accuracy of the model, we can say this is an accurate model given a lot of columns being considered.

There were 1583 values that have been predicted wrongly as not cancelled when they were not and 850 that have been predicted cancelled when they were not. Technically this explains the sensitivity (0.913) and specificity (0.58).

The p value is lower than 0.05 hence shows the predictors have significantly contributed towards the modelling.

```{r}
varImp <- varImp(cartTree1)
arrange(varImp, desc(Overall))
```

Description: df [19 x 1]

	Overall <dbl>
LeadTime	2501.70553
RequiredCarParkingSpaces	2181.73038
MarketSegment	2007.04026
Country	1482.71160
DepositType	1082.85924
StaysInWeekendNights	394.21041
StaysInWeekNights	367.38978
BookingChanges	345.72524
PreviousCancellations	243.89002
AssignedRoomType	154.26788
Meal	90.41333
Adults	0.00000
Children	0.00000
Babies	0.00000
IsRepeatedGuest	0.00000
PreviousBookingsNotCanceled	0.00000
ReservedRoomType	0.00000
CustomerType	0.00000
TotalOfSpecialRequests	0.00000

1-10 of 19 rows

11-19 of 19 rows

But have all columns been so important? No. As we see there are 5 columns (leadtime, requiredcarparkingspaces, marketsegment, country and deposittype) that have contributed highly and 6 columns (staysinweeknights, staysinweekendnights, bookingchanges, previouscancellation, assignedroomtype, meal)

The 9 columns (adults, children, babies, isrepeatedguests, previousbookingnotcancelled, reservedroomtype, customertype, totalofspecialrequests) are the least contributors.

So, let's create another model with top 5 columns.

```{r}
cartTree2 <- rpart(IsCanceled ~ LeadTime+RequiredCarParkingSpaces+Country+
                   DepositType+MarketSegment, data = trainSet)
```

```{r}
rpartPred2 <- predict(cartTree2, newdata=testSet, type="class")
#rpartPred stores the prediction of cartTree by using predict() on the testing
#dataset
confusionMatrix(rpartPred2, testSet$IsCanceled)
```

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 9179 1754
         1  659 2027

               Accuracy : 0.8228
                 95% CI : (0.8163, 0.8292)
    No Information Rate : 0.7224
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.515

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9330
            Specificity : 0.5361
         Pos Pred Value : 0.8396
         Neg Pred Value : 0.7547
             Prevalence : 0.7224
         Detection Rate : 0.6740
   Detection Prevalence : 0.8028
      Balanced Accuracy : 0.7346

       'Positive' Class : 0
```

The model has performed better now, (minor difference but still counts) the accuracy has increased and is now **82.28%**, but we shall also investigate other parameters such as sensitivity and specificity which are equivalent to the previous model. So, the least contributors can be removed to create a robust model.

```
```{r}
cartTree3 <- rpart(IsCanceled ~ LeadTime+RequiredCarParkingSpaces+Country+DepositType+
                     +StaysInWeekendNights+StaysInWeekNights+
                     BookingChanges+PreviousCancellations+AssignedRoomType+Meal,
                   data = trainSet)
```
```

```
```{r}
rpartPred3 <- predict(cartTree3, newdata=testSet, type="class")
#rpartPred stores the prediction of cartTree by using predict() on the testing
#dataset
confusionMatrix(rpartPred3, testSet$IsCanceled)
```
```

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 8995 1585
         1  843 2196

               Accuracy : 0.8217
                 95% CI : (0.8152, 0.8281)
    No Information Rate : 0.7224
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5269

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9143
            Specificity : 0.5808
         Pos Pred Value : 0.8502
         Neg Pred Value : 0.7226
             Prevalence : 0.7224
         Detection Rate : 0.6605
   Detection Prevalence : 0.7769
      Balanced Accuracy : 0.7476

       'Positive' Class : 0
```

Comparing our model with another model with 11 columns (all the columns that have contributed).

```
```{r}
cartTree3 <- rpart(IsCanceled ~ LeadTime+RequiredCarParkingSpaces+Country+DepositType+
                     +StaysInWeekendNights+StaysInWeekNights+
                     BookingChanges+PreviousCancellations+AssignedRoomType+Meal,
                   data = trainSet)
```
```

```
```{r}
rpartPred3 <- predict(cartTree3, newdata=testSet, type="class")
#rpartPred stores the prediction of cartTree by using predict() on the testing
#dataset
confusionMatrix(rpartPred3, testSet$IsCanceled)
```
```

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 8995 1585
         1  843 2196

               Accuracy : 0.8217
                 95% CI : (0.8152, 0.8281)
    No Information Rate : 0.7224
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5269

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9143
            Specificity : 0.5808
         Pos Pred Value : 0.8502
         Neg Pred Value : 0.7226
             Prevalence : 0.7224
         Detection Rate : 0.6605
   Detection Prevalence : 0.7769
      Balanced Accuracy : 0.7476

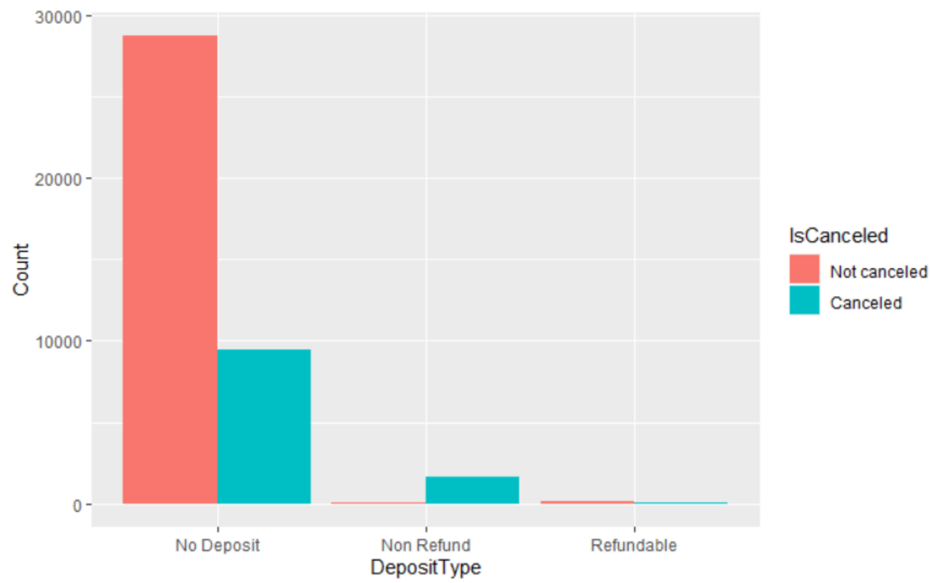
       'Positive' Class : 0
```

The model's performance has lowered. (minor)

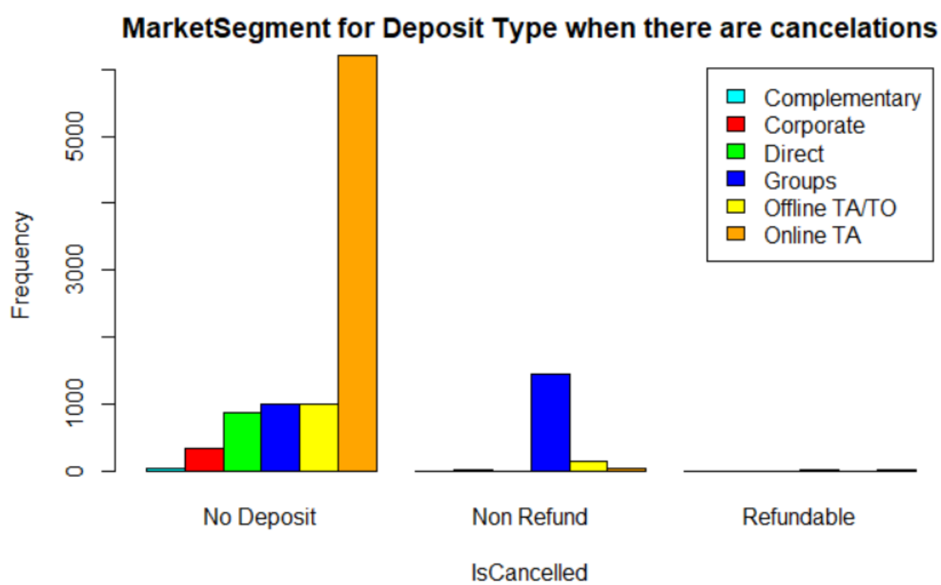
All the models have performed equivalently.
the accuracy is **82.17%** but we shall also investigate other parameters such as sensitivity and specificity which are equivalent to the previous model. So, the 6 minor contributions can also be removed.

<u>Recommendations to the CEO:</u>

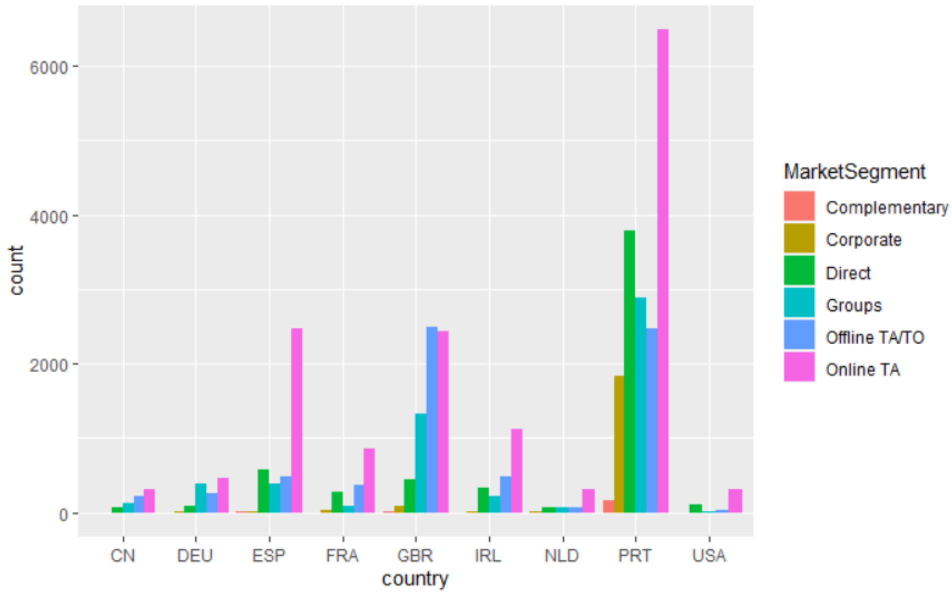
1)



- When we take a close look at this graph, we can see that about 95% of the non-refund DepositType are canceling their hotel room reservations, which is very odd as they have already paid money up front and they are still canceling.
- We take a closer look at this along with different Market Segments



- Now we see that for the non-refund type of deposit type, the market segment “groups” is considerably higher than the others.
- We dig even more deep and see top 10 country market segments

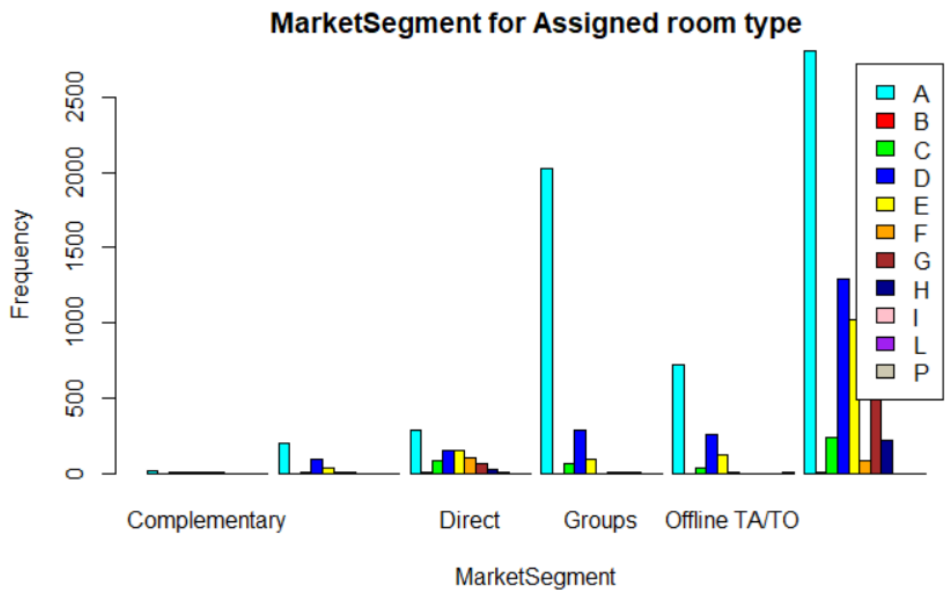


- The recommendation is that for the countries Portugal and Great Britain, which have high count for the groups market segment, should ask the groups who come to book to either opt for a no deposit type or a refundable type so that their cancelations would decrease.

2) This is based on the deposit type

- The recommendation is that the CEO should increase the amount of the deposit value so that when people who really want to come would pay up front and not end up canceling their reservation as the amount to be paid would be substantially higher than the previous amount.

3) The last recommendation is on the assigned room type



By seeing this graph, we can say that for the groups market segment, the assigned room type A is the highest and they are getting canceled as well. So we recommend that you allow flexibility in changing the room types for groups of customers so that they don't cancel their room reservation.