

IST687- Final Project

LAB INSTRUCTOR:AYSE DALGALI

Jeffrey Stephens, Mark Gopani, Warren Fernandes

TABLE OF CONTENTS

Chapter No.	Section No.	Title
1.	1.1 1.2 1.3 1.4	OVERVIEW Problem Statement Dataset and Descriptive Statistics Variable Descriptions Business Questions
2.	2.1	CLEANING THE DATASET
3.	3.1 3.2 3.3 3.4 3.5	EXPLORATORY ANALYSIS Filtering people who have bookings longer than 10 days Comparing Lead Time Grouping Customer type and Reservation status Categorising customers and Identifying dominant categories Categorising and mapping Revenue generated
4.	4.1 4.2 4.3	BUSINESS ANALYSIS MODELS Association rules mining Linear Modelling Support Vector Machine
5.		CONCLUSION

1. Overview

1.1 Problem Statement

The key goal of the project is to analyse the two datasets given to us which are representing hotel bookings data from two different locations. We will try to analyse the Customer segmentation, Revenue analysis and Cancellation analysis. We will provide a comparative analysis between the two locations and understand how each property is performing over time. The analysis will help the hotel operator ascertain key trends or emerging challenges to determine if any change is needed in the ongoing strategies.

1.2 Dataset and Descriptive Statistics

The two datasets containing the customer data are taken into consideration. One of the hotels (H1) is a resort hotel and the other is a city hotel (H2). Both datasets have the same structure, with 31 variables describing the 40,060 observations of H1 and 79,330 observations of H2. Each observation represents a hotel reservation by a customer. The bookings in the dataset were made between 1st of July of 2015 and the 31st of August 2017. There is data describing whether a Customer arrived or didn't show up or the booking was cancelled.

We know that both hotels are located in Portugal (southern Europe) ("H1 at the resort region of Algarve and H2 at the city of Lisbon"). The distance between these two locations is ca. 280 km by car and both locations border on the North Atlantic.

We can look at a descriptive summary of the datasets using the "hmisc" package. This package provides the count of variables and observations. It works well with both categorical and numerical data, giving appropriate summaries in each case, even adapting its output to take into account for instance how many categories exist in a given field. It shows how much data is missing, if any.

```

> Hmisc::describe(City_hotel)
City_hotel

28 Variables      79330 Observations
-----
IsCanceled
  n missing distinct   Info    Sum    Mean    Gmd
79330      0        2   0.729  33102  0.4173  0.4863
-----
LeadTime
  n missing distinct   Info    Mean    Gmd   .05   .10   .25   .50   .75   .90   .95
79330      0      453     1   109.7   116.3   1.0   4.0  23.0  74.0  163.0  277.0  334.6
lowest :    0    1    2    3    4, highest: 608 615 622 626 629
-----
Arrival Date
  n missing distinct   Info    Mean    Gmd   .05   .10   .25   .50
40060    39270    792     1 2016-07-09 23963000 2015-08-09 2015-09-03 2015-10-22 2016-07-02
.75      .90      .95
2017-02-20 2017-06-05 2017-07-11
lowest : 2015-07-01 2015-07-02 2015-07-03 2015-07-04 2015-07-05, highest: 2017-08-27 2017-08-28 2017-08-29 2017-08-30 2017-08-31
-----
ReservationStatusDate
  n missing distinct   Info    Mean    Gmd   .05   .10   .25   .50
79330      0      864     1 2016-07-31 22730541 2015-08-04 2015-09-23 2016-02-05 2016-08-10
.75      .90      .95
2017-02-06 2017-05-30 2017-07-13
lowest : 2014-10-17 2015-01-01 2015-01-20 2015-01-30 2015-02-17, highest: 2017-09-03 2017-09-04 2017-09-05 2017-09-06 2017-09-07
-----
ReservationStatus
  n missing distinct
79330      0        3

Value      Canceled Check-Out   No-Show
Frequency    32186    46228     916
Proportion   0.406     0.583     0.012

```

For numeric data, instead of viewing the range as such, we can see the highest and lowest 5 entries. It helps to show at a glance whether there is any weird outlier or whether there are several values many standard deviations away from the mean.

```
> Hmisc::describe(Resort_hotel)
```

```
Resort_hotel
```

```
28 Variables      40060 Observations
```

```
IsCanceled
```

n	missing	distinct	Info	Sum	Mean	Gmd
40060	0	2	0.602	11122	0.2776	0.4011

```
LeadTime
```

n	missing	distinct	Info	Mean	Gmd	.05	.10	.25	.50	.75	.90	.95
40060	0	412	0.999	92.68	103.9	0	1	10	57	155	238	287

```
lowest : 0 1 2 3 4, highest: 471 532 542 709 737
```

```
Arrival Date
```

n	missing	distinct	Info	Mean	Gmd	.05	.10	.25	.50	.75
40060	0	793	1	2016-08-16	22784249	2015-08-14	2015-09-23	2016-02-14	2016-08-19	2017-03-05
.90	.95									

```
2017-06-22 2017-07-28
```

```
lowest : 2015-07-01 2015-07-02 2015-07-03 2015-07-04 2015-07-05, highest: 2017-08-27 2017-08-28 2017-08-29 2017-08-30 2017-08-31
```

```
ReservationStatusDate
```

n	missing	distinct	Info	Mean	Gmd	.05	.10	.25	.50	.75
40060	0	913	1	2016-07-29	22828733	2015-07-30	2015-09-15	2016-01-26	2016-07-31	2017-02-11
.90	.95									

```
2017-06-02 2017-07-17
```

```
lowest : 2014-11-18 2015-01-01 2015-01-02 2015-01-18 2015-01-21, highest: 2017-09-08 2017-09-09 2017-09-10 2017-09-12 2017-09-14
```

```
ReservationStatus
```

n	missing	distinct
40060	0	4

Value	Canceled	Check-Out	No-Show	
Frequency	1	10831	28937	291
Proportion	0.000	0.270	0.722	0.007

1.3 Variable Description

<i>Variable</i>	<i>Description</i>
<i>ADR</i>	Average Daily Rate
<i>Adults</i>	Number of adults
<i>Agent</i>	ID of the travel agency that made the booking
<i>ArrivalDate</i>	Date customer was scheduled to arrive
<i>AssignedRoomType</i>	Code for the type of room assigned to the booking. Sometimes the assigned room type differs from the reserved room type due to hotel operation reasons (e.g., overbooking) or by customer request. Code is presented instead of designation for anonymity reasons
<i>Babies</i>	Number of babies
<i>BookingChanges</i>	Number of changes/amendments made to the booking from the moment the booking was entered on the PMS until the moment of check-in or cancellation
<i>Children</i>	Number of children
<i>Company</i>	ID of the company/entity that made the booking or responsible for paying the booking. ID is presented instead of designation for anonymity reasons
<i>Country</i>	Country of origin. Categories are represented in the ISO 3155–3:2013 format
<i>CustomerType</i>	Type of booking, assuming one of four categories: Contract - when the booking has an allotment or other type of contract associated to it Group – when the booking is associated to a group Transient – when the booking is not part of a group or contract, and is not associated to other transient booking Transient-party – when the booking is transient, but is associated to at least other transient booking

<i>DaysInWaitingList</i>	Number of days the booking was in the waiting list before it was confirmed to the customer
-	-
<i>DepositType</i>	Indication on if the customer made a deposit to guarantee the booking. This variable can assume three categories:
-	No Deposit – no deposit was made.
	Non-Refund – a deposit was made in the value of the total stay cost
	Refundable – a deposit was made with a value under the total cost of stay.
	-
<i>DistributionChannel</i>	Booking distribution channel. The term ‘TA’ means Travel Agents and ‘TO’ means Tour Operators
-	-
<i>IsCanceled</i>	Value indicating if the booking was canceled (1) or not (0)
-	-
<i>IsRepeatedGuest</i>	Value indicating if the booking name was from a repeated guest (1) or not (0)
-	-
<i>LeadTime</i>	Number of days that elapsed between the entering date of the booking into the PMS and the arrival date
-	-
<i>MarketSegment</i>	Market segment designation. In categories, the term ‘TA’ means Travel Agents and ‘TO’ means Tour Operators
-	-
<i>Meal</i>	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)
	-
<i>PreviousBookingsNotCancelled</i>	Number of previous bookings not cancelled by the customer prior to the current booking
-	-
<i>PreviousCancellations</i>	Number of previous bookings that were cancelled by the customer prior to the current booking
-	-
<i>RequiredCardParkingSpaces</i>	Number of car parking spaces required by the customer
-	-
<i>ReservationStatus</i>	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
	-

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

1.4 Business Questions

- For customers staying for longer periods is there a difference in the customer base for the two locations?
- What type of customer bookings are the most?
- Which areas generate the highest revenue?
- What is the distribution of bookings type with respect to the reservation status?
- What is the meal preference of the customers?
- Which features are most important to predict cancelations?
- What are the best rules for ARS?

2. Cleaning the Dataset

When we look at the missing values in the data, we can see that the City Hotel dataset has quite a few NA values.

```
> sum(is.na(H1))
[1] 0
> sum(is.na(H2))
[1] 39270
```


We shall therefore fill in the NA values. The best way is to use Interpolation which is an estimation of a value within two known values in a sequence of values.

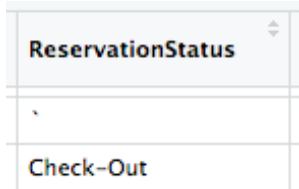
When graphical data contains a gap, but data is available on either side of the gap or at a few specific points within the gap, interpolation allows us to estimate the values within the gap.

In R we can do interpolation with the help of the 'imputeTS' package.

```
> H2<-na_interpolation(H2)
> H1<-na_interpolation(H1)
> sum(is.na(H1))
[1] 0
> sum(is.na(H2))
[1] 0
```

Now we can see that there are no NA values in either datasets. Now we can proceed with our analysis.

We also observe an anomaly in the reservation status column of Resort dataset.



ReservationStatus
Check-Out

We fix the data in the following way:

```
H1$ReservationStatus[13] <- 'Check-Out'
```

We are assuming the data to be 'check-out' based on the values around which have a similar ReservationStatusDate.

We also need to consider the variable types to conduct our analysis therefore we convert some of the variables to factors while analysing the association rules or other types as required.

3. Exploratory Analysis

3.1 Filtering people who have bookings longer than 10 days

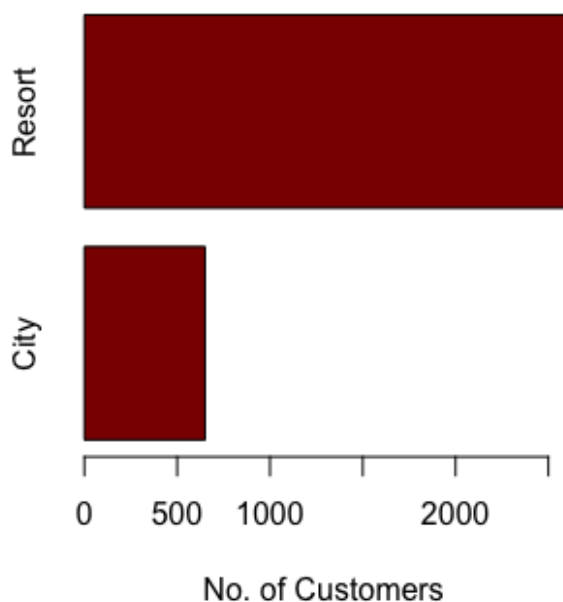
We consider the StaysInWeekendNights and StaysInWeekNights and make a new variable which holds the total Number of nights the customer will be staying. We then filter out people who stay for 10 or more nights. We create a new dataset to store the observations of customers staying for 10 or more days.

```
City_hotel$TotDaysCity <-City_hotel$StaysInWeekendNights+City_hotel$StaysInWeekNights
LongStayCity<-City_hotel[which(City_hotel$TotDaysCity>=10),]

Resort_hotel$TotDaysResort <-Resort_hotel$StaysInWeekendNights+Resort_hotel$StaysInWeekNights
LongStayResort<-Resort_hotel[which(Resort_hotel$TotDaysResort>=10),]
```

We can hope to observe through our analysis that there will be more people staying at the Resort Hotel for vacation than the City hotel.

Customers Staying 10 or more Days

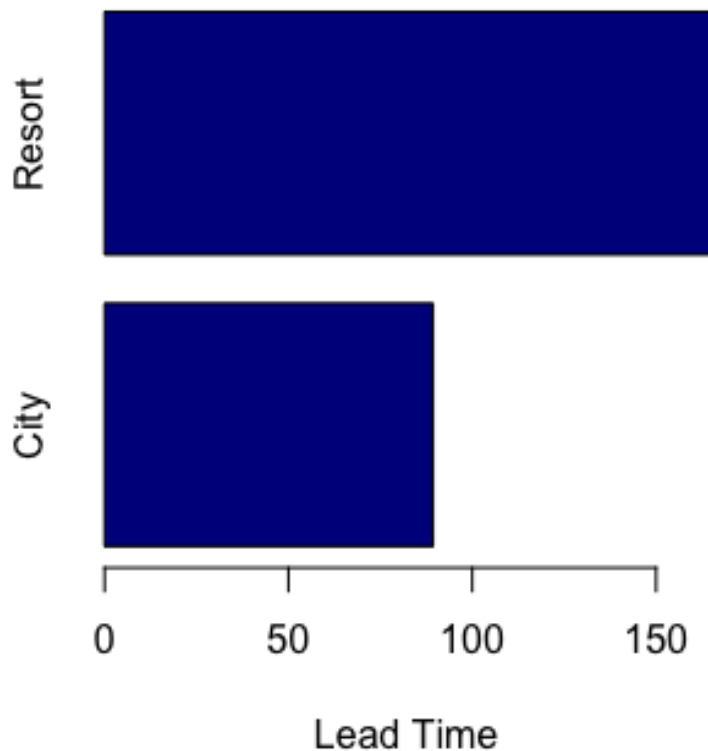


```
Totalobs<-c(nrow(LongStayCity),nrow(LongStayResort))
barplot(Totalobs, main="Customers Staying 10 or more Days",xlab="No. of Customers",
        names.arg = c("City", "Resort"),
        col = "darkred",
        horiz = TRUE)
```

3.2 Comparing Lead Time

Now if we consider the LeadTime variable which shows us the number of days that elapsed between the entering date of the booking into the PMS and the arrival date. We will aim to compare the lead time of the two datasets for people who stay for 10 or more nights.

Mean Lead Time comparision



```
means<-c(mean(LongStayCity$LeadTime),mean(LongStayResort$LeadTime))
barplot(means, main="Mean Lead Time comparision",xlab="Lead Time",
        names.arg = c("City", "Resort"),
        col = "darkBlue",
        horiz = TRUE)
```

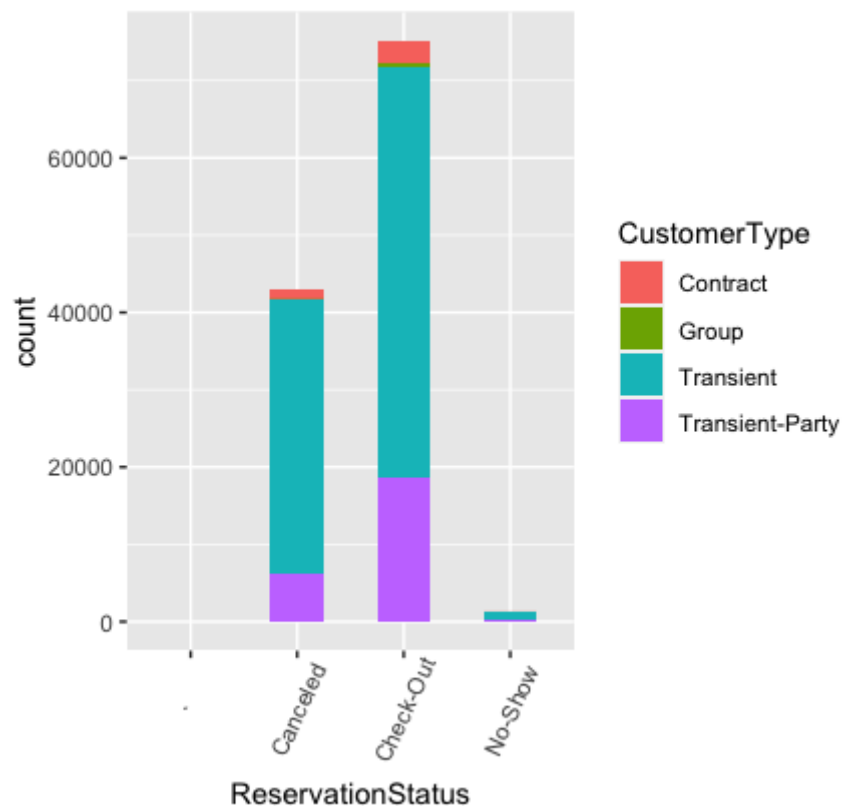
Seeing a higher mean lead time for the Resort Hotel represents that the bookings were made well in advance on an average compared to City Hotel. This is a general trend showing people tend to plan ahead to go to a Resort Hotel, while for a City Hotel reservation, not much prior planning is put in.

3.3 Grouping Customer type and Reservation status

We then considered the variables ReservationStatus and CustomerType for both the datasets combined. ReservationStatus gives us one of the three values 'Canceled', 'Check-out', 'No-show' and CustomerType gives us 'Contract', 'Group', 'Transient', 'Transient-Party'.

We found out that CustomerType 'Transient' has the highest weightage of 75% followed by 'Transient- Party' with 21% and 'Contract' with 3.4% and 'Group' with 0.4%.

Therefore we can conclude that customers of type 'Transient' contribute the most to the revenue even though they are also significant in the ReservationStatus 'Canceled' category.



```
g <- ggplot(Hotels, aes(ReservationStatus)) +  
  geom_bar(aes(fill=CustomerType), width = 0.5) +  
  theme(axis.text.x = element_text(angle=65, vjust=0.6))
```

3.4 Categorising customers and Identifying dominant categories

We can further explore this data by considering the Adults, Children and Babies variables to categorise the customers into Groups of "Single", "Couple" and "Family"

We create a new variable to store the categories of customers staying for 10 or more days.

```
LongStayCity$Cuscategory<-"Single"
LongStayCity$Cuscategory[LongStayCity$Children > 0 | LongStayCity$Babies >0] <- 'Family'
LongStayCity$Cuscategory[LongStayCity$Adults == 2 & LongStayCity$Children == 0
                          & LongStayCity$Babies == 0] <- "Couple"
LongStayCity$Cuscategory <- factor(LongStayCity$Cuscategory, levels = c("Family", "Single", "Couple"))
table(LongStayCity$Cuscategory)
#
LongStayResort$Cuscategory<-"Single"
LongStayResort$Cuscategory[LongStayResort$Children > 0 | LongStayResort$Babies >0] <- 'Family'
LongStayResort$Cuscategory[LongStayResort$Adults == 2 & LongStayResort$Children == 0
                          & LongStayResort$Babies == 0] <- "Couple"
LongStayResort$Cuscategory <- factor(LongStayResort$Cuscategory, levels = c("Family", "Single", "Couple"))
table(LongStayResort$Cuscategory)

> table(LongStayCity$Cuscategory)
```

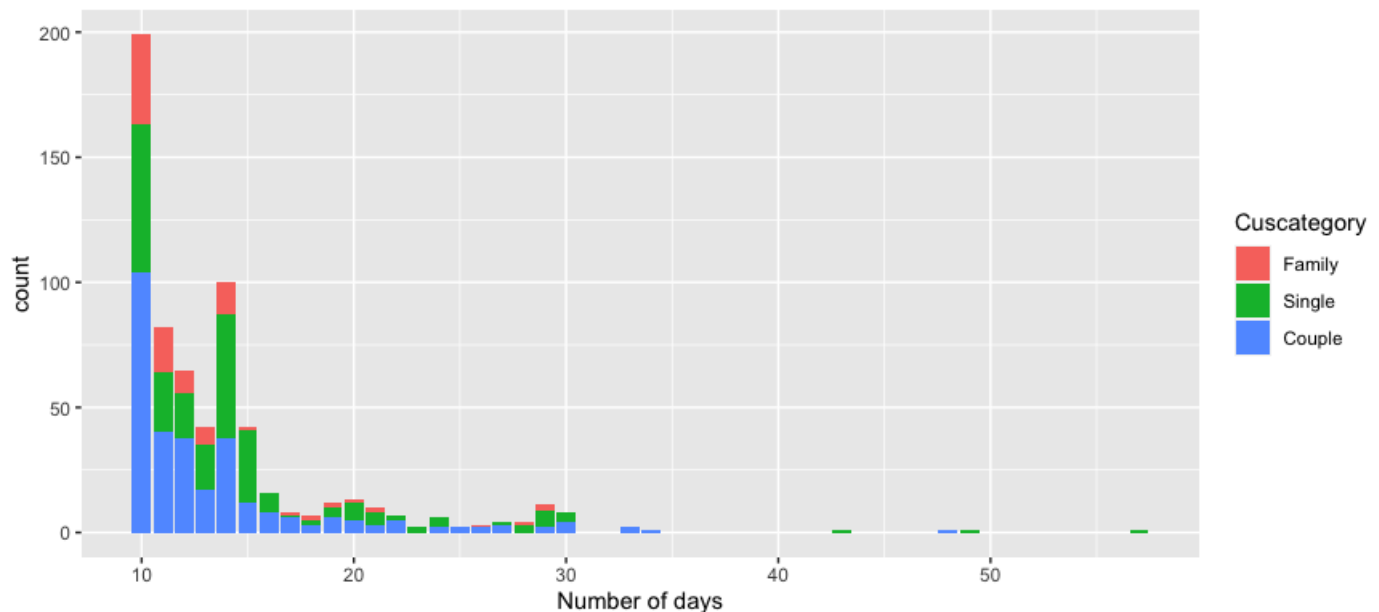
```
Family Single Couple
   96   250   304
```

```
> table(LongStayResort$Cuscategory)
```

```
Family Single Couple
  210   306  2089
```

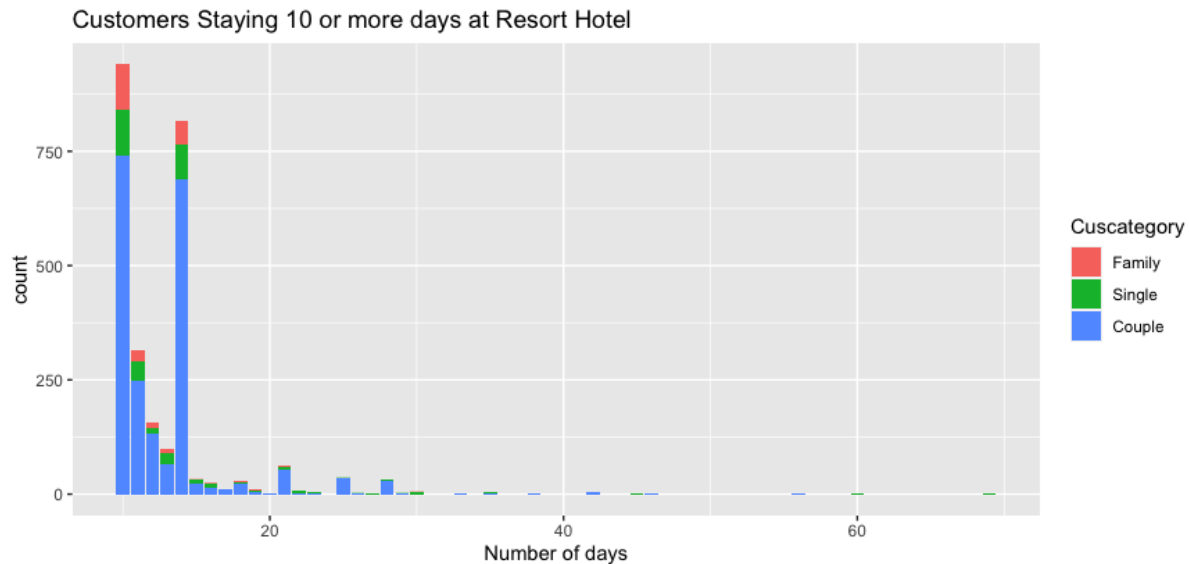
We now have an idea of the category of customers that generates maximum revenue in a unit transaction.

Customers Staying 10 or more days at City Hotel



Such Customers play a vital role in boosting the total revenue generated.

We find that Customer categories 'Single' and 'Couple' show a higher weightage w.r.t generating revenue in a unit transaction with 38.46% and 46.76% weightage respectively. We see that the category 'Family' contributes about 14.7%

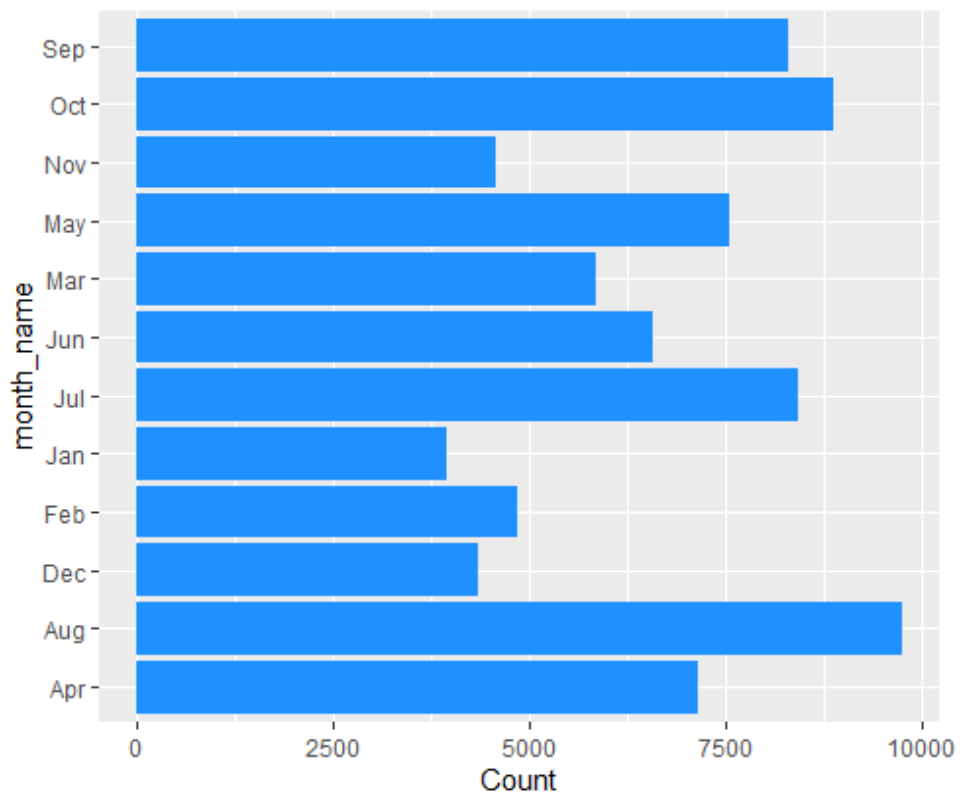
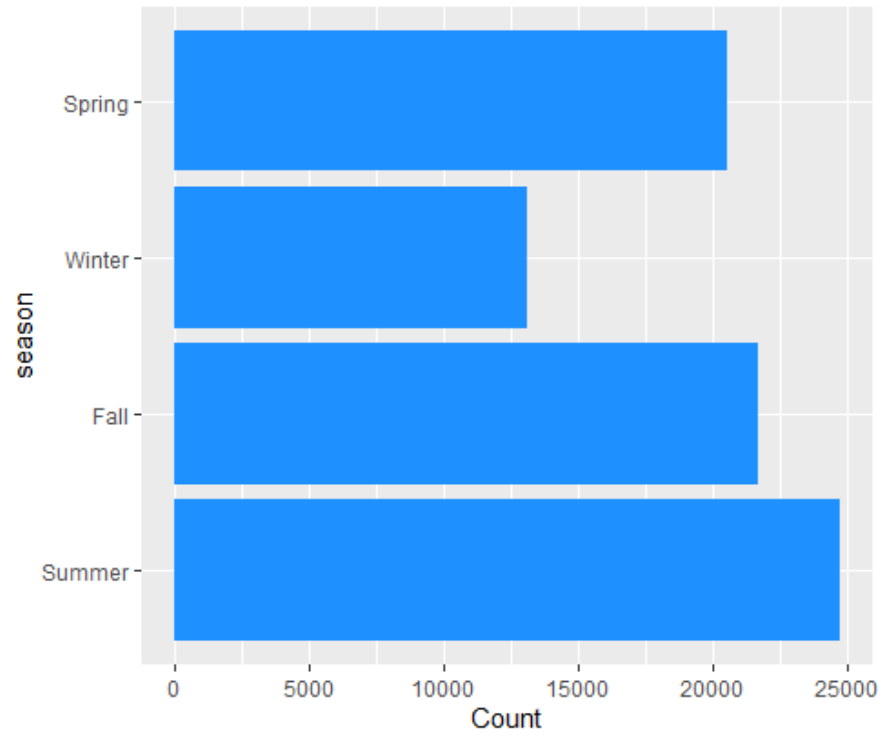


We can see from our analysis and plots that maximum revenue is generated by the Customer category 'Couples' at the Resort hotel staying for 10 or more days.

Couples constitute to 80.19% of guests who stay at Resort Hotel for 10 or more days.

Singles constitute to 11.74% of guests and Families constitute 8.07% of guests.

```
Hotels%>%group_by(month_name)%>%summarise(Count = n())%>%arrange(-Count)%>%ggplot(aes(x = month_name, y = Count)) +
  geom_bar(stat = 'identity',fill = "dodgerblue") + coord_flip()
Hotels%>%group_by(season)%>%summarise(Count = n())%>%arrange(-Count)%>%ggplot(aes(x = season, y = Count)) +
  geom_bar(stat = 'identity',fill = "dodgerblue") + coord_flip()
```

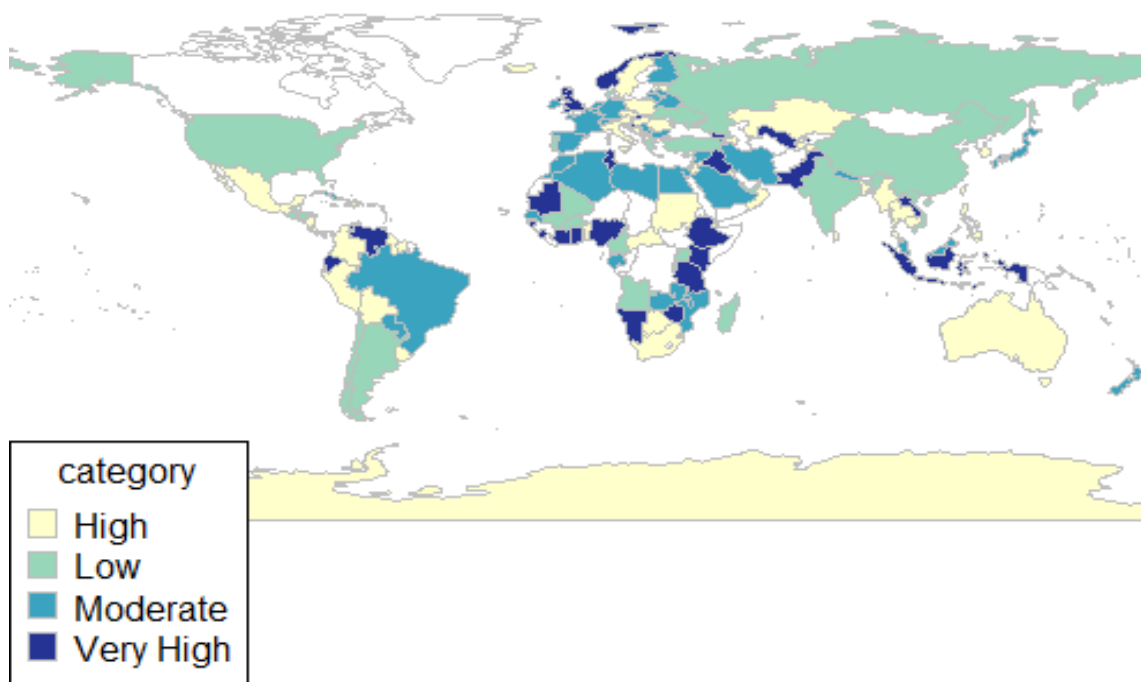


3.5 Categorising and mapping Revenue generated

We make an ARS variable which is the Average Revenue per Stay and we categorise it storing it into a new variable 'RevenueCategory' which has four levels 'Very High', 'High', 'Moderate' and 'Low'.

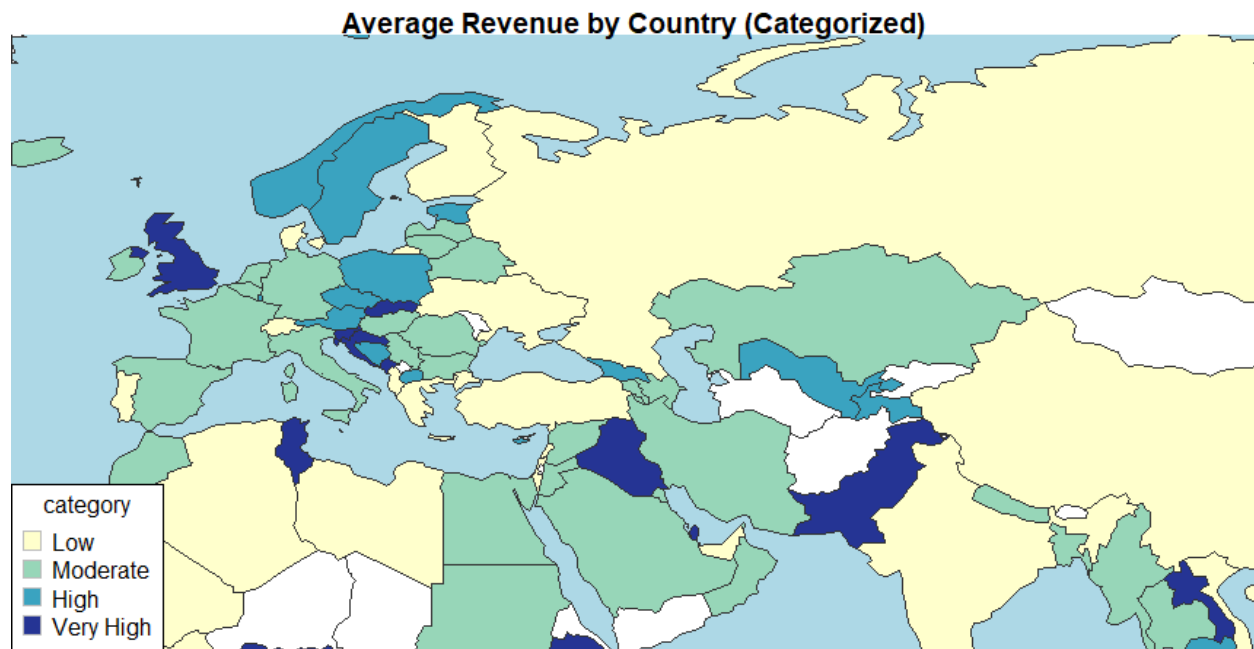
```
Hotels$ARS <- (Hotels$StaysInWeekendNights + Hotels$StaysInWeekNights) * Hotels$ADR
cuts = quantile(Hotels$ARS, c(0, 0.25, 0.5, 0.75, 1))
length(cuts)
Hotels$RevenueCategory <- "Very High"
Hotels$RevenueCategory[Hotels$ARS >= cuts[1] & Hotels$ARS < cuts[2]] = "Low"
Hotels$RevenueCategory[Hotels$ARS >= cuts[2] & Hotels$ARS < cuts[3]] = "Moderate"
Hotels$RevenueCategory[Hotels$ARS >= cuts[3] & Hotels$ARS < cuts[4]] = "High"
```

RevenueCategory



```
nPDF <- joinCountryData2Map(Hotels, joinCode = "IS03", nameJoinColumn = "Country")
colourPalette <- brewer.pal(6, 'YlGnBu')
mapCountryData( nPDF, nameColumnToPlot="RevenueCategory", colourPalette = colourPalette, numCats = 4)
```

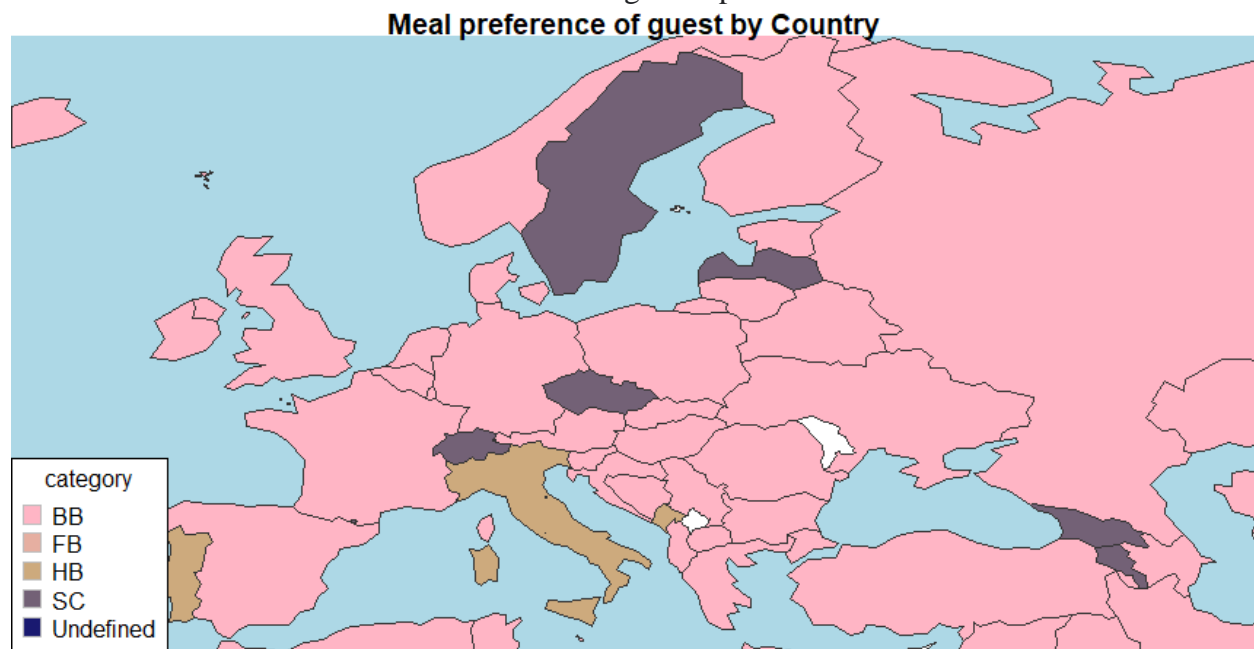
Having an idea about which countries are the source of high revenue can aid advertising strategies to focus on yielding areas.



```
mapCountryData(nPDF, nameColumnToPlot="RevenueCategory", colourPalette = colourPalette, mapRegion = "eurasia")
```

3.6 Meal category map for Eurasia

We can also look at the meal map of Eurasia to identify the meal categories country wise. This way the business can have a reference for formulating meal plan offers or discounts.



```
mapCountryData(nPDF, nameColumnToPlot="Meal", colourPalette = colourPalette, mapRegion = "eurasia")
```

4. BUSINESS ANALYSIS MODELS

4.1 Association Rules Mining

Association rule mining is a procedure which is meant to find frequent patterns, correlations, associations, or causal structures from data sets found in various kinds of databases such as relational databases, transactional databases, and other forms of data repositories.

Given a set of transactions, association rule mining aims to find the rules which enable us to predict the occurrence of a specific item based on the occurrences of the other items in the transaction.

The main applications of association rule mining are:

- Basket data analysis - is to analyze the association of purchased items in a single basket or single purchase as per the examples given above.
- Cross marketing - is to work with other businesses that complement your own, not competitors. For example, vehicle dealerships and manufacturers have cross marketing campaigns with oil and gas companies for obvious reasons.
- Catalog design - the selection of items in a business' catalog are often designed to complement each other so that buying one item will lead to buying of another. So these items are often complements or very related.

Association rules are created by thoroughly analyzing data and looking for frequent if/then patterns. Then, depending on the following two parameters, the important relationships are observed:

- Support: Support indicates how frequently the if/then relationship appears in the database.
- Confidence: Confidence tells about the number of times these relationships have been found to be true.

So, in a given transaction with multiple items, Association Rule Mining primarily tries to find the rules that govern how or why such products/items are often bought together.

In our case, we first look at linear models which are derived from the clean data of the hotels. The same variables are considered for association rules mining. Most machine learning algorithms work with numeric datasets and hence tend to be mathematical. However, association rule mining is suitable for non-numeric, categorical data, therefore we picked certain variables which we found suitable for the association rules and saved them as factors. The variables we chose are DepositType, IsCanceled, IsRepeatedGuest, Adults, PreviousCancellations, from both the datasets. We also created a new column in both datasets which considers the total visitors for a booking and then categorises the visitors into three categories: 'Group', 'Single' and 'Couple'.

```

depor<- as.factor(H1$DepositType)
canr<-as.factor(ifelse(H1$IsCanceled==0,
                      "Not Canceled","Canceled"))
repr<-as.factor(ifelse(H1$IsRepeatedGuest==0,
                      "New Guest","Repeated Guest"))
cuscatgr<- 'Single'
H1$cuscatgrtot<-as.numeric(H1$Adults) + as.numeric(H1$Children) + as.numeric(H1$Babies)
cuscatgr<-as.factor(ifelse(H1$cuscatgrtot>2,
                          "Family","Couple"))
prevcatr<-as.factor(ifelse(H1$PreviousCancellations>0,
                          "has cancelled before","Never cancelled before"))
newdfResort<-data.frame(depor,cuscatgr,canr,repr,prevcatr)

```

```

depo <-as.factor(H2$DepositType)
can<-as.factor(ifelse(H2$IsCanceled==0,
                     "Not Canceled","Canceled"))
rep<-as.factor(ifelse(H2$IsRepeatedGuest==0,
                     "New Guest","Repeated Guest"))
cuscatg<- 'Single'
H2$Children <- na_interpolation(H2$Children)
H2$cuscatgrtot<-as.numeric(H2$Adults) + as.numeric(H2$Children) + as.numeric(H2$Babies)
cuscatg<-as.factor(ifelse(H2$cuscatgrtot>2,
                         "Family","Couple"))
prevcat<-as.factor(ifelse(H2$PreviousCancellations>0,
                         "has cancelled before","Never cancelled before"))
newdfCity<-data.frame(depo,cuscatg,can,rep,prevcat)

```

The IsCanceled variable which is our independent variable has two ranges = 'Canceled', 'Not Canceled'. For IsRepeated guest we have 'New Guest' and 'Repeated Guest', Adults is indexed as 'Group' for number of adults greater than 2 and 'Private' otherwise. PreviousCanceled is indexed as 'has cancelled before', 'Never cancelled before'. Our next step is to create a transaction set.

```

Citytrans <- as(newdfCity,"transactions") |
Resorttrans <-as(newdfResort,"transactions")

```

Now Applying the association mining rules, for the city_hotel dataset we opt for a minimum support of 0.05 which provides us with 12 rules while for the resort_hotel dataset we opt for a minimum support of 0.005 as we do not get any rules with support=0.05. We have 18 rules for the Resort_hotel dataset.

We have our independent variable IsCanceled on rhs and the other important variables affecting it on the lhs.

```
rulesCity <- apriori(Citytrans,
                     parameter=list(supp=0.05, conf=0.55),
                     control=list(verbose=F),
                     appearance=list(default="lhs",rhs = "can=Canceled"))
summary(rulesCity)
```

```
> summary(rulesCity)
set of 12 rules
```

```
rulesResort <- apriori(Resorttrans,
                      parameter=list(supp=0.005, conf=0.55),
                      control=list(verbose=F),
                      appearance=list(default="lhs",rhs = "canr=Canceled"))
inspect(rulesResort)
```

We then filter the rules to find good rules with high lift after reviewing the ruleset. We opt for lift >1 for the City Hotel data set and lift >3 for the Resort Hotel data set.

```
goodrulesCity<-rulesCity[quality(rulesCity)$lift>1]
inspect(goodrulesCity[1:5])
#summary(goodrulesCity)

goodrulesResort<-rulesResort[quality(rulesResort)$lift>3]
inspect(goodrulesResort[1:5])
#summary(goodrulesResort)
```

```
> inspect(goodrulesCity[1:5])
```

	lhs	rhs	support	confidence	coverage	lift	count
[1]	{prevcat=has cancelled before}	=> {can=Canceled}	0.06325476	0.9311561	0.06793143	2.231545	5018
[2]	{depo=Non Refund}	=> {can=Canceled}	0.16190596	0.9981349	0.16220850	2.392062	12844
[3]	{cuscateg=Couple,prevcat=has cancelled before}	=> {can=Canceled}	0.06264969	0.9310603	0.06728854	2.231316	4970
[4]	{rep=New Guest,prevcat=has cancelled before}	=> {can=Canceled}	0.05856549	0.9997848	0.05857809	2.396016	4646
[5]	{depo=Non Refund,cuscateg=Couple}	=> {can=Canceled}	0.16172948	0.9983659	0.16199420	2.392616	12830

```
> summary(goodrulesCity)
set of 12 rules
```

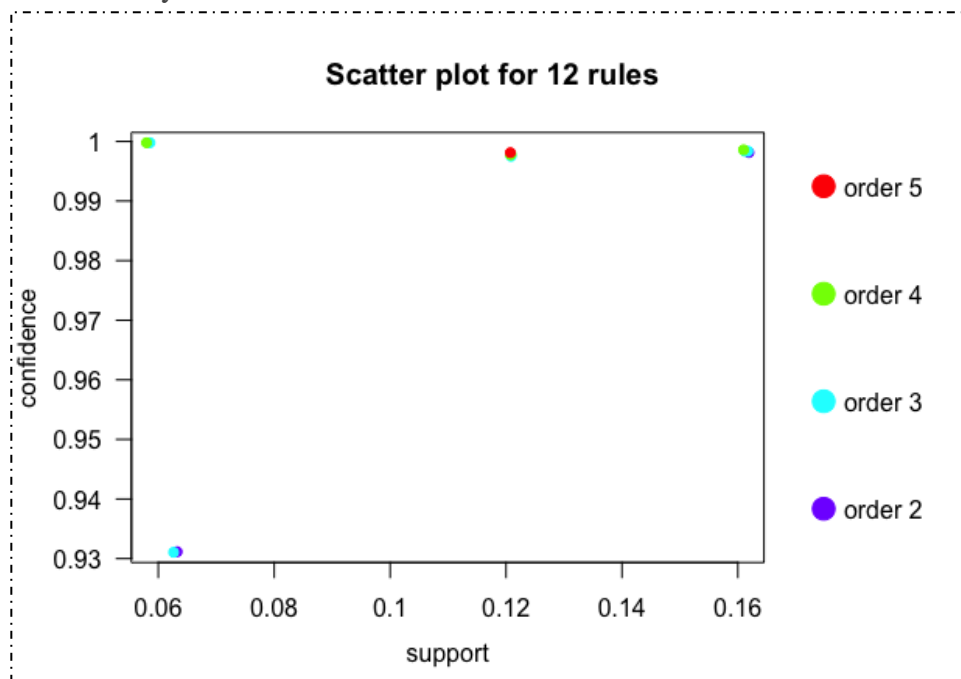
```
> inspect(goodrulesResort[1:5])
```

	lhs	rhs	support	confidence	coverage	lift	count
[1]	{prevcatr=has cancelled before}	=> {canr=Canceled}	0.023065402	0.8438356	0.027333999	3.039386	924
[2]	{depor=Non Refund}	=> {canr=Canceled}	0.041188218	0.9598604	0.042910634	3.457292	1650
[3]	{depor=Non Refund,prevcatr=has cancelled before}	=> {canr=Canceled}	0.009735397	1.0000000	0.009735397	3.601870	390
[4]	{cuscategr=Couple,prevcatr=has cancelled before}	=> {canr=Canceled}	0.022091862	0.8404558	0.026285572	3.027213	885
[5]	{repr=New Guest,prevcatr=has cancelled before}	=> {canr=Canceled}	0.020818772	0.9164835	0.022715926	3.301055	834

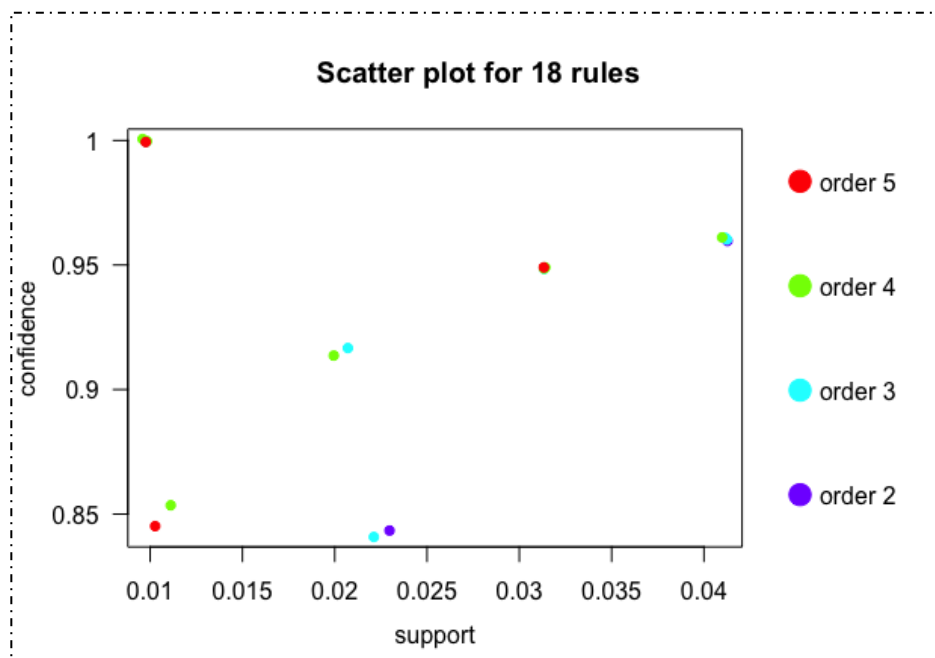
```
> summary(goodrulesResort)
set of 18 rules
```

We now have 12 rules for the City data set and 18 Rules for the Resort Dataset

Rules of City dataset:



Rules of Resort Dataset:



These rules were inspected using the inspect function. The insights we derived from the association rules for users who cancel bookings are:

A guest who has cancelled a reservation before is likely to cancel again.

Couples reservations which were cancelled before are likely to be cancelled again.

A guest with a non-refundable deposit is also likely to cancel.

A New guest who has cancelled a reservation before is likely to cancel.

We then create a new variable called Revenue Category. The revenue Category column is based on the ADR indirectly.

We First calculate Average Daily rate of stay(ARS) which is a product of the total number of nights of stay and the ADR. Based on the ARS values, we categorise the RevenueCategory column into four groups: “Very High”, “High”, “Moderate” and “Low”.

The Revenue category is defined for the combined dataset of the two hotels.

```
Hotels = rbind(H2, H1)

Hotels$ARS <- (Hotels$StaysInWeekendNights + Hotels$StaysInWeekNights) * Hotels$ADR
cuts = quantile(Hotels$ARS, c(0, 0.25, 0.5, 0.75, 1))
length(cuts)
Hotels$RevenueCategory <-("Very High")
Hotels$RevenueCategory[Hotels$ARS >= cuts[1] & Hotels$ARS < cuts[2]] = "Low"
Hotels$RevenueCategory[Hotels$ARS >= cuts[2] & Hotels$ARS < cuts[3]] = "Moderate"
Hotels$RevenueCategory[Hotels$ARS >= cuts[3] & Hotels$ARS < cuts[4]] = "High"
#table(Hotels$RevenueCategory)
Hotels$recat<-as.factor(Hotels$RevenueCategory)
```

We save the new column as factors and then create a new data frame including the variables: market segment, ReservedRoomType, Meals and customer category and Revenue Category. Now we will convert this data frame to a transactions set and then create a ruleset with support=0.05, confidence=0.55 and then we filter out the good rules with lift>2.7 after analysing the ruleset.

```
rulesHotel <- apriori(newdfHotelstrans,
                      parameter=list(supp=0.005, conf=0.55),
                      control=list(verbose=F),
                      appearance=list(default="lhs",rhs = "Hotels.recat=Very High"))
#summary(rulesHotel)
goodrulesHotel<-rulesHotel[quality(rulesHotel)$lift>2.5]
inspect(goodrulesHotel[1:5])
summary(goodrulesResort)
```

```

> goodrulesHotel<-rulesHotel[quality(rulesHotel)$lift>2.5]
> inspect(goodrulesHotel[1:5])

```

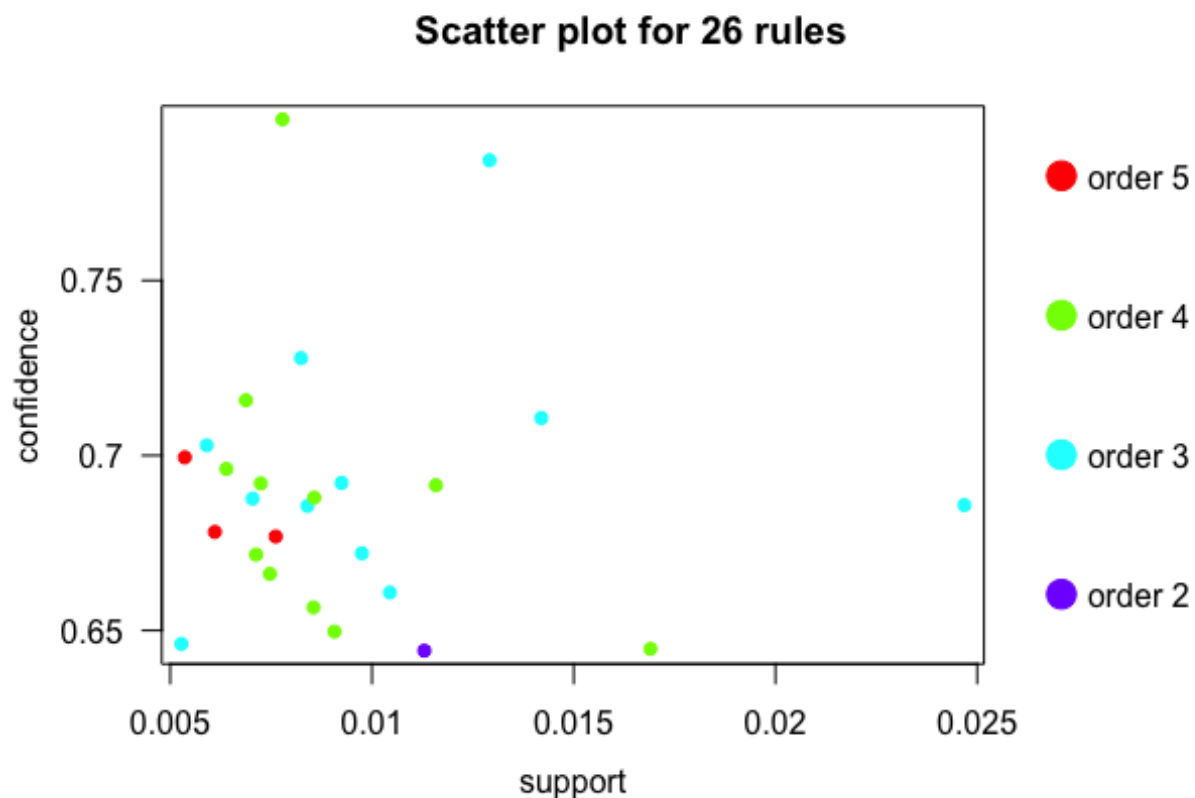
	lhs	rhs	support	confidence	coverage	lift	count
[1]	{as.factor.Hotels.ReservedRoomType.=G}	=> {Hotels.recat=Very High}	0.011299104	0.6442216	0.01753916	2.574514	1349
[2]	{cuscatgeg=Family, as.factor.Hotels.ReservedRoomType.=G}	=> {Hotels.recat=Very High}	0.009247006	0.6921630	0.01335958	2.766103	1104
[3]	{as.factor.Hotels.MarketSegment.=Online TA, as.factor.Hotels.ReservedRoomType.=G}	=> {Hotels.recat=Very High}	0.008401039	0.6855776	0.01225396	2.739786	1003
[4]	{cuscatgeg=Family, as.factor.Hotels.ReservedRoomType.=F}	=> {Hotels.recat=Very High}	0.009749560	0.6720554	0.01450708	2.685747	1164
[5]	{as.factor.Hotels.MarketSegment.=Online TA, as.factor.Hotels.ReservedRoomType.=F}	=> {Hotels.recat=Very High}	0.010444761	0.6608373	0.01580534	2.640916	1247

```

> summary(goodrulesHotel)
set of 26 rules

```

We can plot the rules to get a better idea of the values.



These rules were inspected using the inspect function. The insights we derived from the association rules for ASR are:

Room type 'G' indirectly generates the highest revenue.

Families reserving room type 'G' are focus for high revenue.

Online TA prove to be an area of focus as well.

Room type 'F' indirectly generates high revenue after room type 'F'.

4.2 Linear Modelling

Linear regression attempts to model the relationship between two variables by fitting a linear equation to observed data whereas, multiple regression tries to model the relationship between many variables. For instance, the variables in this dataset LeadTime, Adults, Reservation status and others are the explanatory variables, and the dependent variables are ISCanceled, and IsRepeatedGuest. Figures 1-3 compare ISCanceled and IsRepeatedGuest between hotels. Figure 1 shows more cancelations in the city hotel. However, the Resort has slightly more repeated guests and Figure 3 shows ADR is higher in the City.

```
ggplot(data = Hotels,aes(IsCanceled))+ geom_histogram(binwidth = 0.5, col='black', fill='blue', alpha = 0.4) + facet_wrap(~Hotel)
```

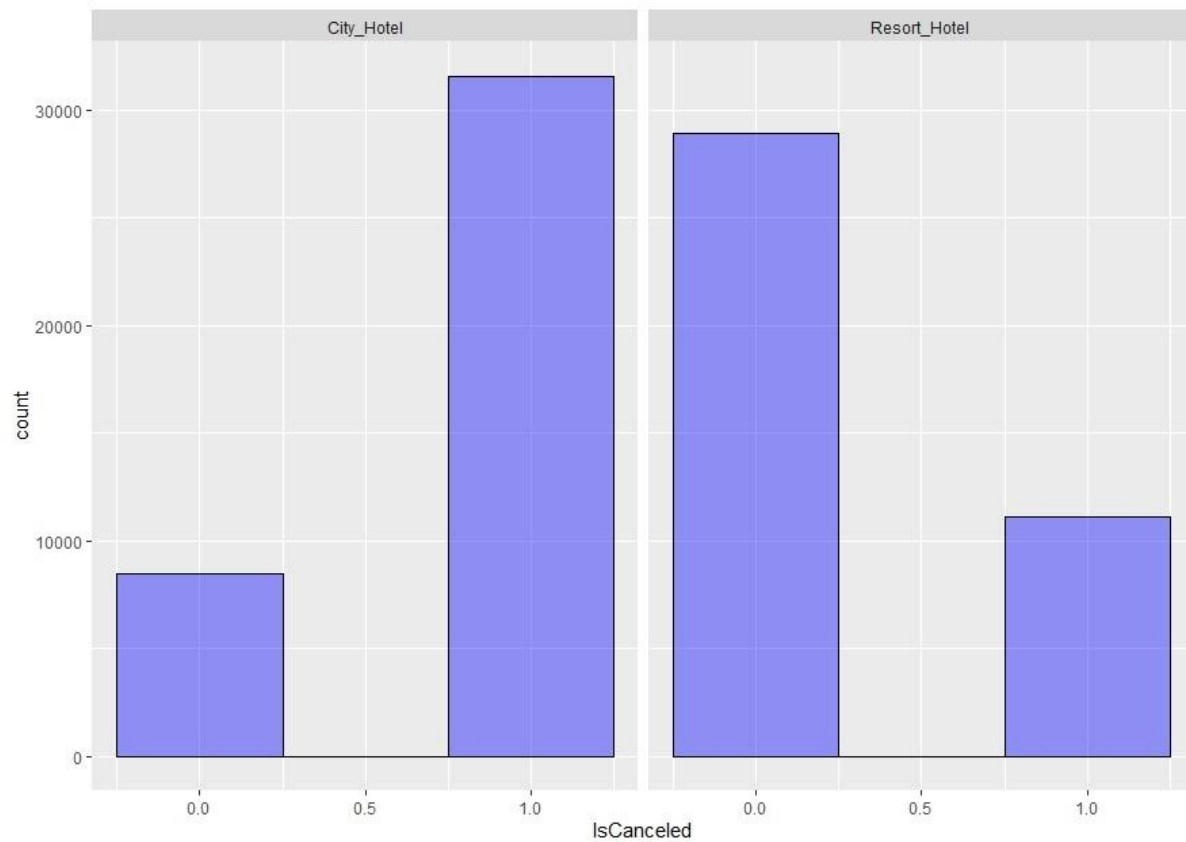


Figure 1. Shows the number of canceled hotel reservations as 0 (not canceled) and 1 (canceled)


```
ggplot(data = Hotels,aes(IsRepeatedGuest))+ geom_histogram(binwidth = 0.5, col='black', fill='blue',  
alpha = 0.4) + facet_wrap(~Hotel)
```

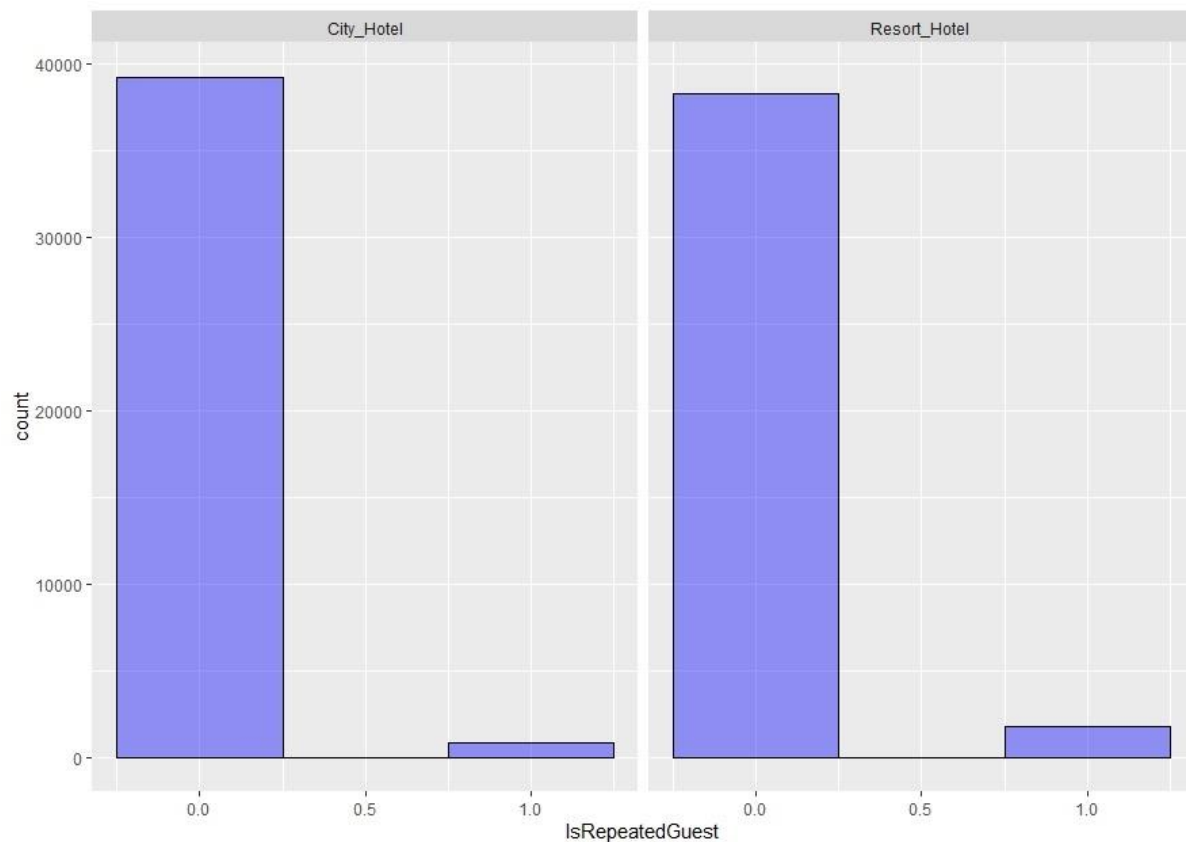


Figure 2. Shows the number of repeated guests at each hotel as 0 (not repeated) or 1 (repeated)

In order to fit a linear model we first determine whether or not there is a relationship between the variables in the Resort Hotel (H1) and the City Hotel (H2). To do this we explore the relationships using graphs and correlation coefficients (figures 4 and 5). Please note this does not imply that one variable causes the other, but that there is a significant association between the two variables. Figures 4 and 5 below shows the relationship between variables for the Resort and City Hotels. However this graph is difficult to read, so we plotted only the correlation coefficients (figures 6 and 7) for both the City and Resort Hotels. Figures 6 and 7 illustrate the important relationships between variables clearer and highlights the possible variables to input into the linear model for each hotel.

```

#subset data to only numeric columns in H1 and H2
subset1 = H1[,c(1:2,6:10,15:17,20,24,26:28)]
subset2 = H2[,c(1:2,6:10,15:17,20,24,26:28)]
#Plot Correlations
chart.Correlation(subset1, histogram = TRUE, pch = 19)
chart.Correlation(subset2, histogram = TRUE, pch = 19)

```

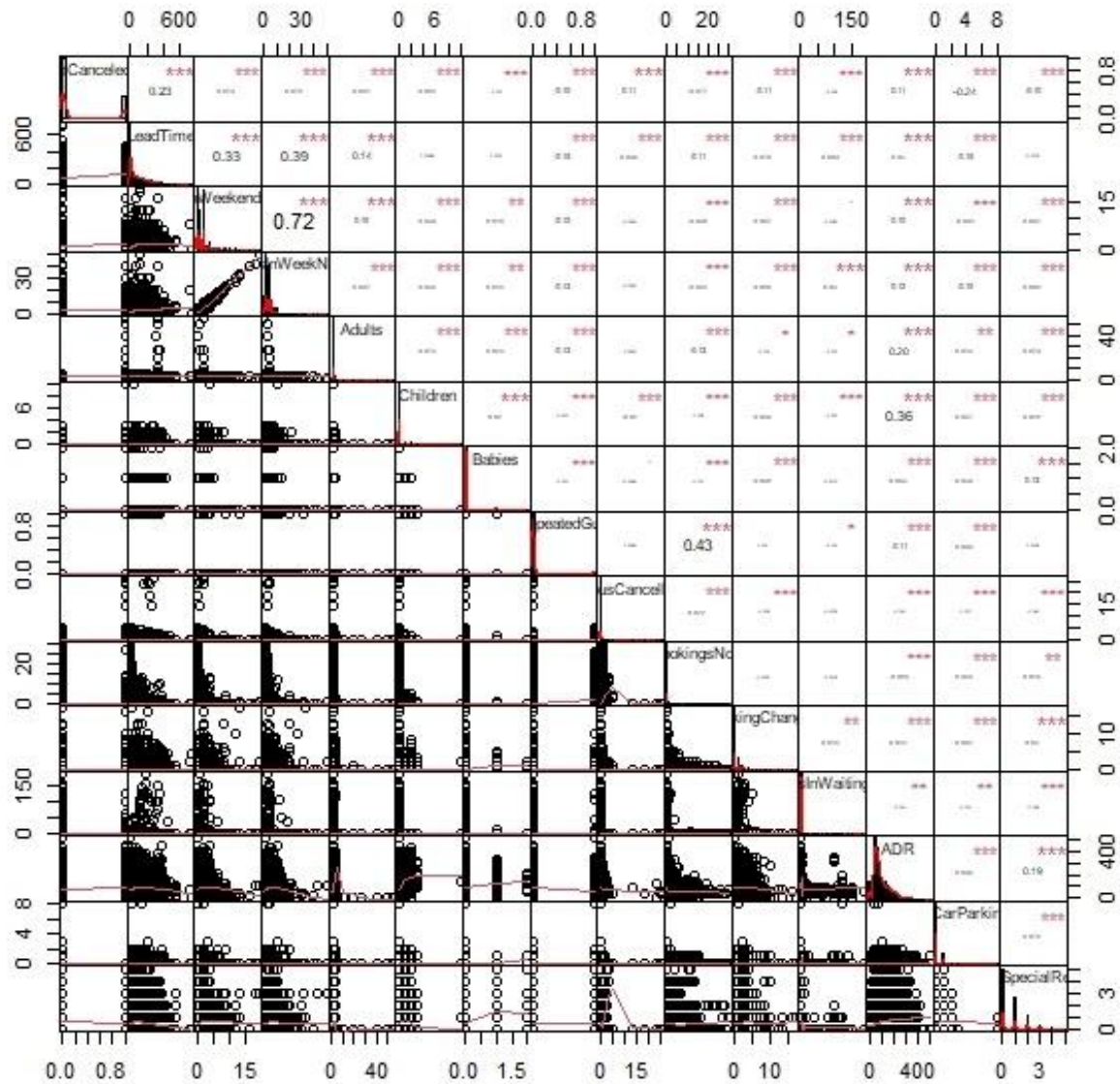


Figure 4. Correlation graphs and correlation coefficients for all numeric variables in the Resort Hotel

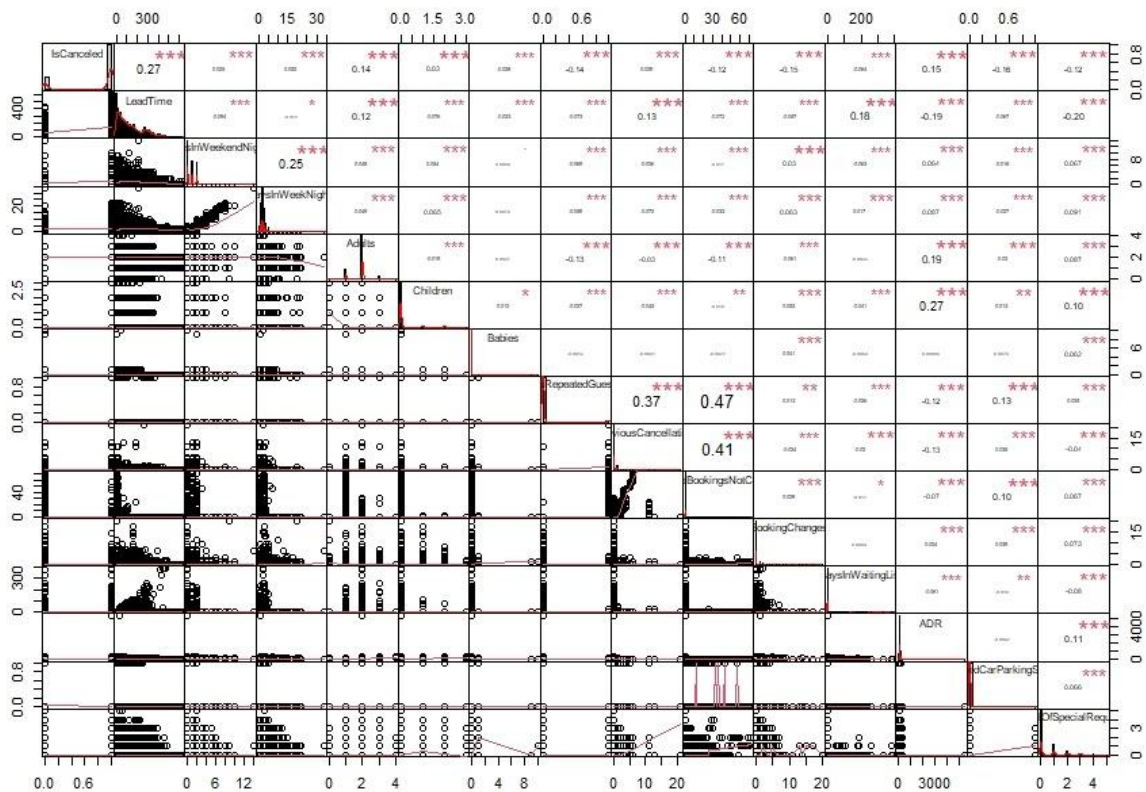


Figure 5. Correlation graphs and correlation coefficients for all numeric variables in the City hotel

When there appears to be no association between the proposed explanatory and dependent variables (i.e., the graphs do not indicate any increasing or decreasing trend), then fitting a linear regression model to the data probably will not provide a useful model. Furthermore, the correlation coefficient is an important numerical measure of association between two variables, which is a value between -1 and 1 indicating the strength of the association of the observed data for the two variables.

By looking at figures 6 and 7 for each hotel we can see right away that the Repeated Guest is positively correlated with Previous Booking Not Cancelled and that if the guest stays weekends they also stay weeknights. In figures 6 and 7 the shades of grey indicate a positive correlation, and the number indicates the strength of the relationship. Whereas the shades of red indicate a negative correlation and the values measures the strength of the relationship. The values range from -1 to 1 with negative numbers representing negative correlations and positive values representing positive correlations.

```
ggpairs(subset1)
ggcorr(subset1, nbreaks=8, palette='RdGy', label=TRUE, hjust = 1, label_size=5, label_color='white',
layout.exp = 1)
ggpairs(subset2)
ggcorr(subset2, nbreaks=8, palette='RdGy', label=TRUE, hjust = 1, label_size=5, label_color='white',
layout.exp = 1)
```

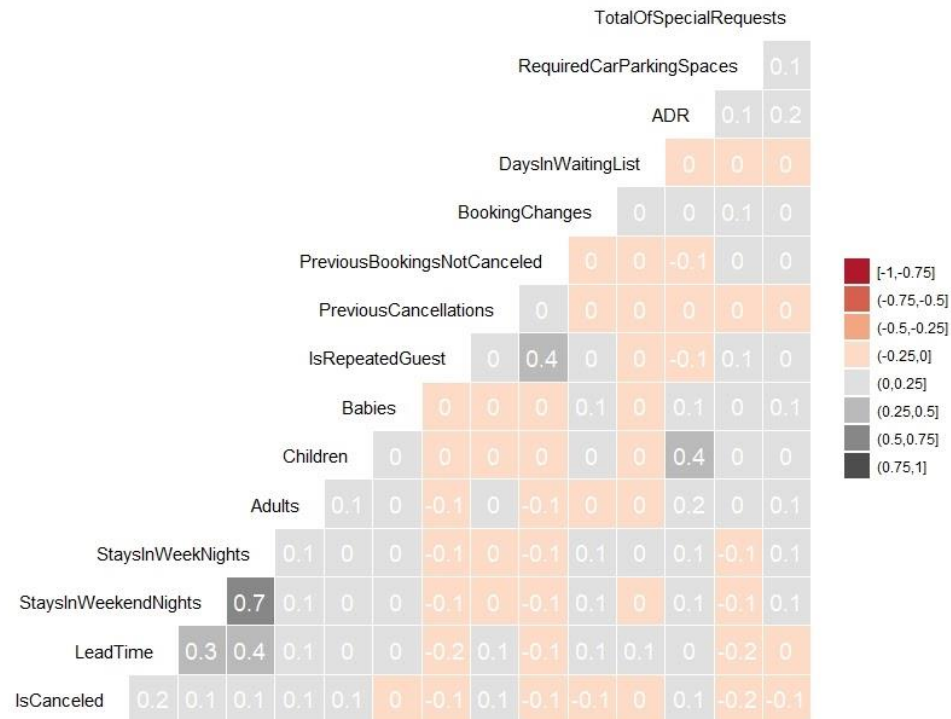


Figure 6. Shows the correlation coefficients values for all numerical variables in Resort Hotel

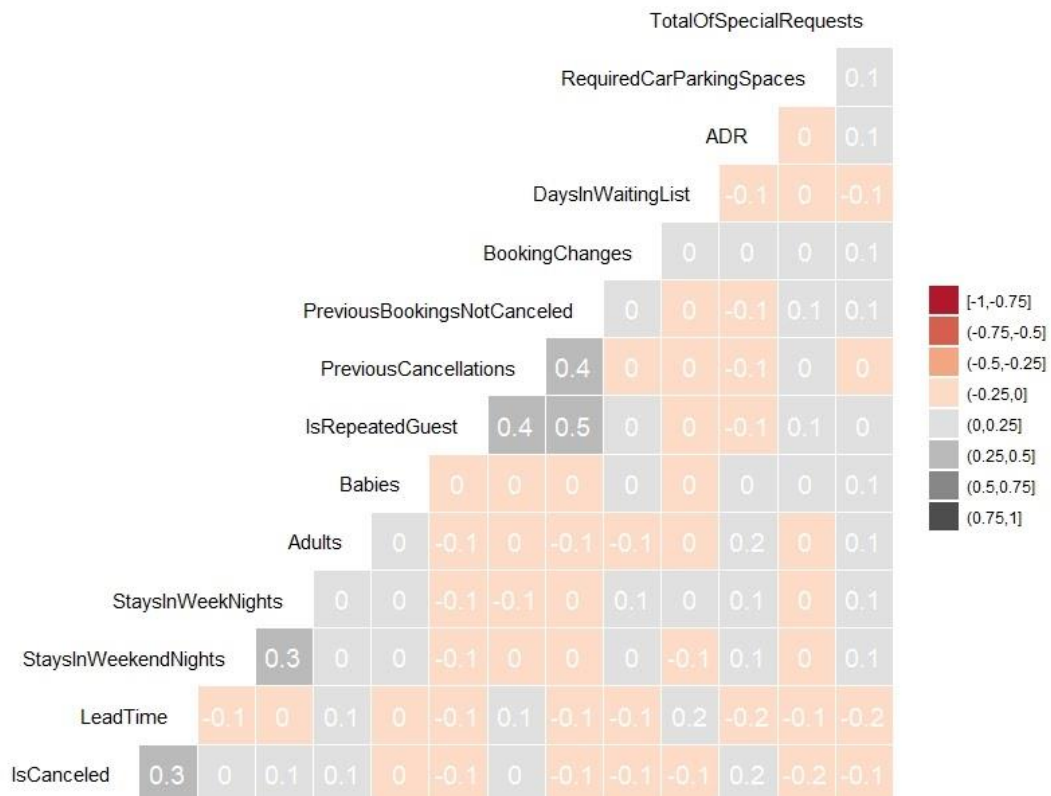


Figure 7. Shows the correlation coefficients values for all numerical variables in City Hotel.

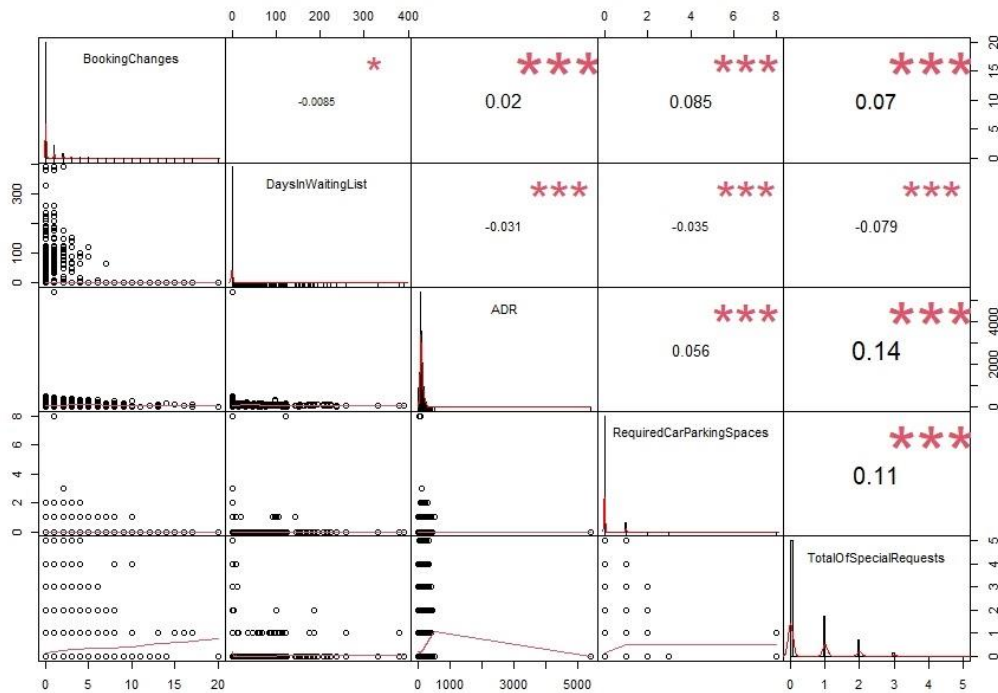
To improve linear models both hotels were combined into one data frame. Below figure 8 a and b are graphs and correlations of numeric variables for the Hotels data frame as one dataset. This resulted in more significant correlations. Moreover, this was computationally expensive, so we had to subset the data into 2 parts due to RAM issues. However, figure 9 shows the correlation coefficients for all numeric variables as one graph. We see that required parking is negatively correlated with Iscanceled and includes the other correlations when comparing hotels individually.

```
subset1_17 = Hotels[,c(1:2,6:10,15:17)]
```

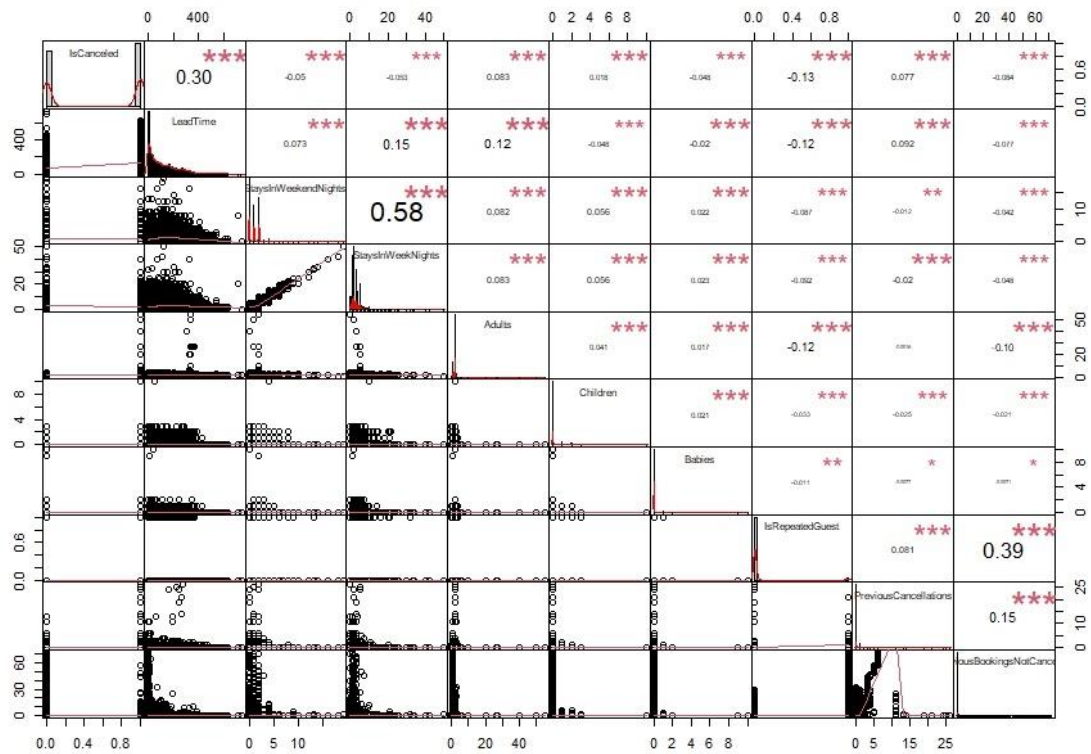
```
subset1_28 = Hotels[,c(20,24,26:28)]
```

```
chart.Correlation(subset1, histogram = TRUE, pch = 19)
```

```
chart.Correlation(subset1_28, histogram = TRUE, pch = 19)
```



a.



b.

Figure 8 a and b. Correlation graphs for all numeric variables in both Hotels

`ggpairs(subset1)`

`ggcorr(subset1, nbreaks=8, palette='RdGy', label=TRUE, hjust = 1, label_size=5, label_color='white', layout.exp = .9)`

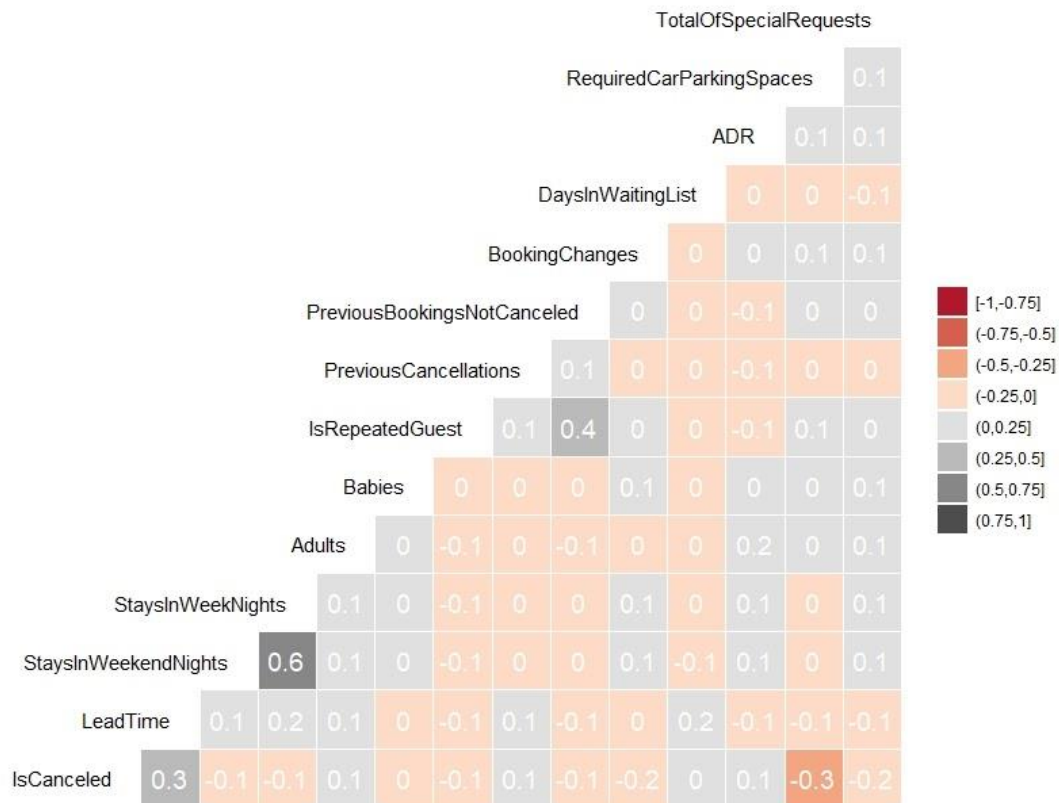


Figure 9. Correlation Coefficients for both Hotels

Call:

```
lm(formula = IsCanceled ~ LeadTime + CustomerType + Hotel + DepositType +
  ADR + RequiredCarParkingSpaces + TotalOfSpecialRequests,
  data = Hotels)
```

Residuals:

Min	1Q	Median	3Q	Max
-4.8175	-0.3042	-0.1725	0.2801	2.1811

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	1.407e-01	7.052e-03	19.945	< 2e-16 ***
LeadTime	6.890e-04	1.221e-05	56.441	< 2e-16 ***
CustomerTypeGroup	-6.297e-02	1.794e-02	-3.509	0.000449 ***
CustomerTypeTransient	1.088e-01	6.503e-03	16.723	< 2e-16 ***
CustomerTypeTransient-Party	-4.052e-02	6.858e-03	-5.909	3.46e-09 ***
HotelResort_Hotel	-1.796e-02	2.590e-03	-6.933	4.13e-12 ***
DepositTypeNon Refund	5.320e-01	4.173e-03	127.490	< 2e-16 ***

DepositTypeRefundable	-8.502e-03	3.175e-02	-0.268	0.788880
ADR	9.281e-04	2.399e-05	38.696	< 2e-16 ***
RequiredCarParkingSpaces	-2.787e-01	4.919e-03	-56.656	< 2e-16 ***
TotalOfSpecialRequests	-8.882e-02	1.574e-03	-56.440	< 2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4026 on 119379 degrees of freedom

Multiple R-squared: 0.3049, Adjusted R-squared: 0.3048

F-statistic: 5235 on 10 and 119379 DF, p-value: < 2.2e-16

```
> lmCanceled$predict <- predict(lmCanceled)
```

```
> p = mean(lmCanceled$predict)
```

```
> p
```

```
[1] 0.3704163
```

Residuals

Residuals are the difference between the actual observed response values and the response values that the model predicted. The Residuals section of the model output breaks it down into 5 summary points. When assessing how well the model fit the data, we look for a symmetrical distribution across these points on the mean value zero (0).

Coefficients

The coefficients of the model, in simple linear regression, are two unknown constants that represent the intercept and slope terms in the linear model.

Coefficient - Estimate

The coefficient Estimate contains two rows; the first one is the intercept. The second row in the Coefficients is the slope.

Coefficient - Standard Error

The coefficient Standard Error measures the average amount that the coefficient estimates vary from the actual average value of our response variable. Ideally want a lower number relative to its coefficients.

Coefficient - t value

The coefficient t-value is a measure of how many standard deviations our coefficient estimate is far away from 0. The farther away from zero indicates to reject the null hypothesis.

Coefficient - Pr(>t)

The Pr(>t) acronym found in the model output relates to the probability of observing any value equal or larger than t. A small p-value indicates that it is unlikely there is a relationship between the predictor and response variables due to chance. Typically, a p-value of 5% or less is a good cut-off point.

Residual Standard Error

Residual Standard Error is a measure of the quality of a linear regression fit. Theoretically, every linear model is assumed to contain an error term E. Due to the presence of this error term, we are not capable of perfectly predicting our response variable (ISCanceled and IsRepeatGuest) from the predictor variables. The Residual Standard Error is the average amount that the response will deviate from the true regression line. For IsCanceled the Residual Standard Error is 0.4 and for IsRepeatGuest was 0.15.

Adjusted R-squared

The R-squared statistic provides a measure of how well the model is fitting the actual data. It takes the form of a proportion of variance. This is a measure of the linear relationship between our predictor variable and our response / target variable. It always lies between 0 and 1 (i.e.: a number near 0 represents a regression that does not explain the variance in the response variable well and a number close to 1 does explain the observed variance in the response variable). For IsCanceled the adjusted R-squared is 0.3. Whereas, the adjusted R-squared for IsRepeatGuest is 0.21.

Lastly, to predict if the guest will cancel their reservation a probability was calculated for each observation. Then a mean was calculated to provide a percent that the variables will predict a guest canceling. For IsCanceled there is a 37% chance the variables LeadTime, CustomerType, Hotel, DepositType, ADR, RequiredCarParkingSpaces, and TotalOfSpecialRequests will predict if guest will cancel their reservation. IsRepeatGuest was also calculated the same way as IsCanceled and the probability that a guest will repeat their stay was only predicted at a probability of 0.3 or 3% using the variables LeadTime, `Arrival Date`, ReservationStatus, StaysInWeekendNights, StaysInWeekNights, Adults, Children, Babies, Country, DistributionChannel, PreviousBookingsNotCanceled, and PreviousCancellations

4.3 Support Vector Machine

In machine learning, support vector machines are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. However, they are mostly used in classification problems.

Advantages and Disadvantages of SVM:

Advantages

- **High Dimensionality:** SVM is an effective tool in high-dimensional spaces, which is particularly applicable to document classification and sentiment analysis where the dimensionality can be extremely large.
- **Memory Efficiency:** Since only a subset of the training points are used in the actual decision process of assigning new members, just these points need to be stored in memory (and calculated upon) when making decisions.
- **Versatility:** Class separation is often highly non-linear. The ability to apply new kernels allows substantial flexibility for the decision boundaries, leading to greater classification performance.

Disadvantages

- **Kernel Parameters Selection:** SVMs are very sensitive to the choice of the kernel parameters. In situations where the number of features for each object exceeds the number of training data samples, SVMs can perform poorly. This can be seen intuitively as if the high-dimensional feature space is much larger than the samples. Then there are less effective support vectors on which to support the optimal linear hyperplanes, leading to poorer classification performance as new unseen samples are added.
- **Non-Probabilistic:** Since the classifier works by placing objects above and below a classifying hyperplane, there is no direct probabilistic interpretation for group membership. However, one potential metric to determine the "effectiveness" of the classification is how far from the decision boundary the new point is.

First, we select the variables required for our SVM model from the original dataset.

```
hotel_resort <- data.frame(agent = H1$Agent,  
                           customer_type = H1$CustomerType,  
                           deposit = H1$DepositType,  
                           cancellations = as.factor(H1$PreviousCancellations),
```

```
reserved_room = H1$ReservedRoomType,
assigned_room = H1$AssignedRoomType,
is_cancelled = as.factor(H1$IsCanceled))
```

```
str(hotel_resort)
```

We create the training and testing datasets by subsetting the created dataset. The variable trainList1 created using the createDataPartition contains the indexes of the rows partitioned in the 60:40 ratio. We use the createDataPartition function because it balances the outcome variable IsCanceled (0:1 values for IsCanceled column is the same) for the training and testing datasets.

```
trainList1 <- createDataPartition(y=hotel_resort$is_cancelled,p=.60,list=FALSE)
```

```
trainSet1 <- hotel_resort[trainList1,]
```

```
testSet1 <- hotel_resort[-trainList1,]
```

Creating the SVM Model

First, we create an SVM model using all the variables in the subset created.

This model has a Variables used to predict IsCanceled are CustomerType and PreviousCancellations.

The Cross-Validation error of 0.227 implies that the model is wrong at predicting the IsCanceled factor 22.7% of times.

Cost of prediction is 5 and Number of Cross Validation is 3.

```
svm_City <- ksvm(IsCanceled ~., data=trainSet2, C=5, cross=3, prob.model=TRUE)
#Creating a svm model using ksvm() function.
svm_City
# The cross validation error is 0.227 which means that about 22.7% of the
#instances that the model was learning on was mistaken.
# Larger cross validation error indicates that the model is not good.
pred_City <- predict(svm_City, newdata=testSet2, type="response")
#Predicting the class of data in testSet to validate the model generated using predict() function
#and passing the model and data as parameters. Storing these prediction in predOut variable.
pred_City #Printing the preOut variable to the console.
confusionMatrix(pred_City, testSet1$IsCanceled)

svm_combined <- ksvm(IsCanceled ~ PreviousCancellations + CustomerType + MarketSegment,
                    data=trainSet1, C=5, cross=3, prob.model=TRUE) #Creating a svm model using ksvm() function.
svm_combined
pred_combined <- predict(svm_combined, newdata=testSet1, type="response")
#Predicting the class of data in testSet to validate the model generated using predict() function
#and passing the model and data as parameters. Storing these prediction in predOut variable.
table(pred_combined, testSet1$IsCanceled)
confusionMatrix(pred_combined, testSet1$IsCanceled)
```

```

svm_Resort <- ksvm(IsCanceled ~., data=trainSet1,
                  C=5, cross=3, prob.model=TRUE)
#Creating a svm model using ksvm() function.
svm_Resort
pred_resort <- predict(svmResort, testSet1)
confusionMatrix(svmResort, testSet1$IsCanceled)

svm_combined2 <- ksvm(IsCanceled ~ PreviousCancellations + season + MarketSegment, data=trainSet2,
                    C=5, cross=3, prob.model=TRUE) #Creating a svm model using ksvm() function.
svm_combined2
pred_combined2 <- predict(svm_combined2, newdata=testSet2, type="response")
#Predicting the class of data in testSet to validate the model generated using predict() function
#and passing the model and data as parameters. Storing these prediction in predOut variable.
table(pred_combined2, testSet2$IsCanceled)
confusionMatrix(pred_combined2, testSet2$IsCanceled)

```

Confusion Matrix and Statistics Reference

Prediction 0 1

0 11507 4088

1 68 360

Accuracy : 0.7406

95% CI : (0.7338, 0.7474)

No Information Rate : 0.7224

P-Value [Acc > NIR] : 1.127e-07

Kappa : 0.104

Mcnemar's Test P-Value

: < 2.2e-16

Sensitivity : 0.99413

Specificity : 0.08094

Pos Pred Value : 0.73786

Neg Pred Value : 0.84112

Prevalence : 0.72240

Detection Rate : 0.71816

Detection Prevalence : 0.97329

Balanced Accuracy : 0.53753

'Positive' Class : 0

Conclusions:

1. More than 60% of the population booked the City hotel.
2. Most bookings were made from July to October which is in the summer to fall season of the year..
3. 80% guests come from Portugal, United Kingdom, France, Germany and Spain.
4. Around 35% of bookings are cancelled.
5. More couples stay at the Resort Hotel for 10 or more nights. While at the City hotel the weight for customers staying 10 or more nights is evenly distributed among families and couples while groups show negligible weightage in both.
6. Couples is the most common accommodation type. So resort and city locations can make arrangement plans accordingly.
7. The variables that are significant for predicting cancellation are previousCancellations, MarketSegment, Season, CustomerType, IsRepeatedGuest.