Hotel Cancellation Prediction in Europe

> By Weilin Zhang, Leah Fowlkes, Jiaqi Wang, Mingwei Li, Man Shi

Project Description:
Our goal is to explore the hotel cancellation status in Europe. From a dataset which collected the information of more than 100,000 customers' booking information for a city hotel and a resort hotel in countries of Europe. By creating predictions for whether a customer will cancel a hotel room or not, we will harness 70% of hotel data (training set) to build several prediction models that assigns a probability of cancellation for each hotel room in the last 30% (test set). We will study the correlations between hotel cancellation and other features, such as deposity type, daily cost rate, and customer type to construct our prediction models. Finally, we would also compare the predictive performance on each supervised learning model and find which method get highest accuracy.

Data:
Link to original dataset.

Initial Setup

In [31]:
```
# Install Packages for Analysis
install.packages(c("MASS","caret"))
install.packages("tidyverse")
install.packages("ggcorrplot")
install.packages("corrplot")
install.packages('pROC')
install.packages('e1071')
install.packages(c("ggthemes", "glmnet","pROC"))
```

```
Installing packages into ‘/home/jupyter/.R/library’
(as ‘lib’ is unspecified)

Installing package into ‘/home/jupyter/.R/library’
(as ‘lib’ is unspecified)

Installing package into ‘/home/jupyter/.R/library’
(as ‘lib’ is unspecified)

Installing package into ‘/home/jupyter/.R/library’
(as ‘lib’ is unspecified)

Installing package into ‘/home/jupyter/.R/library’
(as ‘lib’ is unspecified)

Installing package into ‘/home/jupyter/.R/library’
(as ‘lib’ is unspecified)

Installing packages into ‘/home/jupyter/.R/library’
(as ‘lib’ is unspecified)
```

In [2]:
```
# Load Packages
library(tidyverse)
```

```
library(MASS)
library(caret)
library(magrittr)
library(tidyr)
library(zoo)
library(data.table)
library(ggplot2)
library(scales)
library(randomForest)
library(e1071)
library(corrplot)
library(ggcorrplot)
library(caTools)
library(caret)
library(pROC)
library(glmnet)
library(mlbench)
theme_set(theme_bw())
```

```
── Attaching packages ─────────────────────────────────────── tidyverse 1.3.0 ──

✔ ggplot2 3.3.3     ✔ purrr   0.3.4
✔ tibble  3.0.6     ✔ dplyr   1.0.4
✔ tidyr   1.1.2     ✔ stringr 1.4.0
✔ readr   1.4.0     ✔ forcats 0.5.1

── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()


Attaching package: ‘MASS’


The following object is masked from ‘package:dplyr’:

    select


Loading required package: lattice


Attaching package: ‘caret’


The following object is masked from ‘package:purrr’:

    lift


Attaching package: ‘magrittr’


The following object is masked from ‘package:purrr’:

    set_names


The following object is masked from ‘package:tidyr’:

    extract
```

```
Attaching package: ‘zoo’

The following objects are masked from ‘package:base’:

    as.Date, as.Date.numeric


Attaching package: ‘data.table’

The following objects are masked from ‘package:dplyr’:

    between, first, last

The following object is masked from ‘package:purrr’:

    transpose


Attaching package: ‘scales’

The following object is masked from ‘package:purrr’:

    discard

The following object is masked from ‘package:readr’:

    col_factor

randomForest 4.6-14
Type rfNews() to see new features/changes/bug fixes.

Attaching package: ‘randomForest’

The following object is masked from ‘package:dplyr’:

    combine

The following object is masked from ‘package:ggplot2’:

    margin

corrplot 0.84 loaded
Type 'citation("pROC")' for a citation.

Attaching package: ‘pROC’

The following objects are masked from ‘package:stats’:
```

```
    cov, smooth, var

Loading required package: Matrix

Attaching package: 'Matrix'

The following objects are masked from 'package:tidyr':

    expand, pack, unpack

Loaded glmnet 4.1-1
```

Loading the Hotel Data

In [2]:
```
# European Hotel Data
dat_eu <- read.csv("eu_housing_data.csv")
head(dat_eu)
```

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number
	<fct>	<int>	<int>	<int>	<fct>	<int>
1	Resort Hotel	0	342	2015	July	27
2	Resort Hotel	0	737	2015	July	27
3	Resort Hotel	0	7	2015	July	27
4	Resort Hotel	0	13	2015	July	27
5	Resort Hotel	0	14	2015	July	27
6	Resort Hotel	0	14	2015	July	27

Data Description

- hotel (Resort Hotel or City Hotel)
- is_canceled: Value indicating if the booking was canceled (1) or not (0)
- lead_time: Number of days that elapsed between the entering date of the booking into the PMS and the arrival date
- arrival_date_year: Year of arrival date
- arrival_date_month: Month of arrival date
- arrival_date_week_number: Week number of year for arrival date
- arrival_date_day_of_month: Day of arrival date

- stays_in_weekend_nights: Number of weekend nights (Saturday or Sunday) the guest stayed or booked to stay at the hotel
- stays_in_week_nights: Number of week nights (Monday to Friday) the guest stayed or booked to stay at the hotel
- adults: Number of adults
- children: Number of children
- babies: Number of babies
- meal: 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)
- country: Country of origin. Categories are represented in the ISO 3155–3:2013 format
- market_segment: Market segment designation. In categories, the term “TA” means “Travel Agents” and “TO” means “Tour Operators”
- distribution_channel: Booking distribution channel. The term “TA” means “Travel Agents” and “TO” means “Tour Operators”
- is_repeated_guest: Value indicating if the booking name was from a repeated guest (1) or not (0)
- previous_cancellations: Number of previous bookings that were cancelled by the customer prior to the current booking
- previous_bookings_not_canceled: Number of previous bookings not cancelled by the customer prior to the current booking
- reserved_room_type: Code of room type reserved. Code is presented instead of designation for anonymity reasons
- assigned_room_type: 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
- booking_changes: 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
- deposit_type: 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.
- agent: ID of the travel agency that made the booking
- company: ID of the company/entity that made the booking or responsible for paying the booking. ID is presented instead of designation for anonymity reasons
- days_in_waiting_list: Number of days the booking was in the waiting list before it was confirmed to the customer
- customer_type: 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

- adr: Average Daily Rate as defined by dividing the sum of all lodging transactions by the total number of staying nights
- required_car_parking_spaces: Number of car parking spaces required by the customer
- total_of_special_requests: Number of special requests made by the customer (e.g. twin bed or high floor)
- reservation_status: 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
- reservation_status_date: Date at which the last status was set. This variable can be used in conjunction with the ReservationStatus to understand when was the booking canceled or when did the customer checked-out of the hotel

Reviewing Summary Statistics

In [3]:
```
# Summary stats
summary(dat_eu)
```

```
         hotel        is_canceled       lead_time     arrival_date_year
City Hotel  :68198   Min.   :0.0000   Min.   :  0.0   Min.   :2015
Resort Hotel:36367   1st Qu.:0.0000   1st Qu.: 19.0   1st Qu.:2016
                     Median :0.0000   Median : 72.0   Median :2016
                     Mean   :0.3796   Mean   :107.5   Mean   :2016
                     3rd Qu.:1.0000   3rd Qu.:166.0   3rd Qu.:2017
                     Max.   :1.0000   Max.   :737.0   Max.   :2017

arrival_date_month arrival_date_week_number arrival_date_day_of_month
August :12303      Min.   : 1.0             Min.   : 1.00
July   :10812      1st Qu.:16.0             1st Qu.: 8.00
May    :10164      Median :28.0             Median :16.00
October:10023      Mean   :27.3             Mean   :15.79
April  : 9734      3rd Qu.:38.0             3rd Qu.:23.00
June   : 9424      Max.   :53.0             Max.   :31.00
(Other):42105
stays_in_weekend_nights stays_in_week_nights     adults
Min.   : 0.0000         Min.   : 0.000       Min.   : 0.000
1st Qu.: 0.0000         1st Qu.: 1.000       1st Qu.: 2.000
Median : 1.0000         Median : 2.000       Median : 2.000
Mean   : 0.9189         Mean   : 2.506       Mean   : 1.852
3rd Qu.: 2.0000         3rd Qu.: 3.000       3rd Qu.: 2.000
Max.   :16.0000         Max.   :41.000       Max.   :55.000

   children            babies                  meal          country
Min.   : 0.00000   Min.   : 0.000000   BB       :80863   PRT    :48590
1st Qu.: 0.00000   1st Qu.: 0.000000   FB       :  791   GBR    :12129
Median : 0.00000   Median : 0.000000   HB       :13777   FRA    :10415
Mean   : 0.09667   Mean   : 0.008205   SC       : 7985   ESP    : 8568
3rd Qu.: 0.00000   3rd Qu.: 0.000000   Undefined: 1149   DEU    : 7287
Max.   :10.00000   Max.   :10.000000                     ITA    : 3766
NA's   :4                                                (Other):13810
      market_segment  distribution_channel is_repeated_guest
Online TA    :46162   Corporate: 6218      Min.   :0.00000
Offline TA/TO:22795   Direct   :12897      1st Qu.:0.00000
Groups       :18863   GDS      :  152      Median :0.00000
Direct       :10893   TA/TO    :85293      Mean   :0.03536
Corporate    : 4902   Undefined:    5      3rd Qu.:0.00000
Complementary:  719                       Max.   :1.00000
(Other)      :  231
previous_cancellations previous_bookings_not_canceled reserved_room_type
```

```
 Min.   : 0.00000       Min.   : 0.0000               A      :76424
 1st Qu.: 0.00000       1st Qu.: 0.0000               D      :16214
 Median : 0.00000       Median : 0.0000               E      : 5634
 Mean   : 0.09861       Mean   : 0.1483               F      : 2347
 3rd Qu.: 0.00000       3rd Qu.: 0.0000               G      : 1702
 Max.   :26.00000       Max.   :72.0000               B      :  901
                                                      (Other): 1343
 assigned_room_type booking_changes       deposit_type       agent
 A      :65676      Min.   : 0.0000   No Deposit:89856   9      :25138
 D      :21814      1st Qu.: 0.0000   Non Refund:14554   NULL   :15083
 E      : 6758      Median : 0.0000   Refundable:  155   240    :12331
 F      : 3084      Mean   : 0.2143                      1      : 6995
 C      : 2122      3rd Qu.: 0.0000                      6      : 3220
 G      : 2090      Max.   :21.0000                      14     : 2898
 (Other): 3021                                           (Other):38900
    company      days_in_waiting_list        customer_type        adr
 NULL   :98201   Min.   :  0.000      Contract       : 3881   Min.   :  -6.38
 40     :  924   1st Qu.:  0.000      Group          :  496   1st Qu.:  67.00
 223    :  781   Median :  0.000      Transient      :77491   Median :  91.94
 67     :  266   Mean   :  2.524      Transient-Party:22697   Mean   : 100.03
 45     :  249   3rd Qu.:  0.000                             3rd Qu.: 123.30
 153    :  206   Max.   :391.000                             Max.   :5400.00
 (Other): 3938
 required_car_parking_spaces total_of_special_requests reservation_status
 Min.   :0.00000             Min.   :0.0000            Canceled :38752
 1st Qu.:0.00000             1st Qu.:0.0000            Check-Out:64870
 Median :0.00000             Median :0.0000            No-Show  :  943
 Mean   :0.06292             Mean   :0.5493
 3rd Qu.:0.00000             3rd Qu.:1.0000
 Max.   :8.00000             Max.   :5.0000

 reservation_status_date
 2015-10-21: 1453
 2015-07-06:  804
 2016-11-25:  777
 2015-01-01:  763
 2016-01-18:  604
 2015-07-02:  467
 (Other)   :99697
```

In [4]:

```
# Review Structure
str(dat_eu)
```

```
'data.frame':   104565 obs. of  32 variables:
 $ hotel                         : Factor w/ 2 levels "City Hotel","Resort Hote
l": 2 2 2 2 2 2 2 2 2 2 ...
 $ is_canceled                   : int  0 0 0 0 0 0 0 0 1 1 ...
 $ lead_time                     : int  342 737 7 13 14 14 0 9 85 75 ...
 $ arrival_date_year             : int  2015 2015 2015 2015 2015 2015 2015 2015
2015 2015 ...
 $ arrival_date_month            : Factor w/ 12 levels "April","August",..: 6 6
6 6 6 6 6 6 6 6 ...
 $ arrival_date_week_number      : int  27 27 27 27 27 27 27 27 27 27 ...
 $ arrival_date_day_of_month     : int  1 1 1 1 1 1 1 1 1 1 ...
 $ stays_in_weekend_nights       : int  0 0 0 0 0 0 0 0 0 0 ...
 $ stays_in_week_nights          : int  0 0 1 1 2 2 2 2 3 3 ...
 $ adults                        : int  2 2 1 1 2 2 2 2 2 2 ...
 $ children                      : int  0 0 0 0 0 0 0 0 0 0 ...
 $ babies                        : int  0 0 0 0 0 0 0 0 0 0 ...
 $ meal                          : Factor w/ 5 levels "BB","FB","HB",..: 1 1 1 1
1 1 1 2 1 3 ...
 $ country                       : Factor w/ 28 levels "AUT","BEL","BGR",..: 24
24 12 12 12 12 24 24 24 24 ...
 $ market_segment                : Factor w/ 8 levels "Aviation","Complementar
```

```
y",..: 4 4 4 3 7 7 4 4 7 6 ...
 $ distribution_channel          : Factor w/ 5 levels "Corporate","Direct",..: 2
2 2 1 4 4 2 2 4 4 ...
 $ is_repeated_guest             : int  0 0 0 0 0 0 0 0 0 0 ...
 $ previous_cancellations        : int  0 0 0 0 0 0 0 0 0 0 ...
 $ previous_bookings_not_canceled: int  0 0 0 0 0 0 0 0 0 0 ...
 $ reserved_room_type            : Factor w/ 10 levels "A","B","C","D",..: 3 3 1
1 1 1 3 3 1 4 ...
 $ assigned_room_type            : Factor w/ 12 levels "A","B","C","D",..: 3 3 3
1 1 1 3 3 1 4 ...
 $ booking_changes               : int  3 4 0 0 0 0 0 0 0 0 ...
 $ deposit_type                  : Factor w/ 3 levels "No Deposit","Non Refun
d",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ agent                         : Factor w/ 322 levels "1","10","103",..: 322 3
22 322 151 101 101 322 150 101 40 ...
 $ company                       : Factor w/ 343 levels "10","100","101",..: 343
343 343 343 343 343 343 343 343 343 ...
 $ days_in_waiting_list          : int  0 0 0 0 0 0 0 0 0 0 ...
 $ customer_type                 : Factor w/ 4 levels "Contract","Group",..: 3 3
3 3 3 3 3 3 3 3 ...
 $ adr                           : num  0 0 75 75 98 ...
 $ required_car_parking_spaces   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ total_of_special_requests     : int  0 0 0 0 1 1 0 1 1 0 ...
 $ reservation_status            : Factor w/ 3 levels "Canceled","Check-Out",..:
2 2 2 2 2 2 2 2 1 1 ...
 $ reservation_status_date       : Factor w/ 926 levels "2014-10-17","2014-11-1
8",..: 122 122 123 123 124 124 124 124 73 62 ...
```

Checking Missing Value

In [15]:

```
is.na(dat_eu) <- dat_eu == "NULL"
colSums(is.na(dat_eu)) # checking na values
```

hotel: 0 **is_canceled:** 0 **lead_time:** 0 **arrival_date_year:** 0 **arrival_date_month:** 0 **arrival_date_week_number:** 0 **arrival_date_day_of_month:** 0 **stays_in_weekend_nights:** 0 **stays_in_week_nights:** 0 **adults:** 0 **children:** 4 **babies:** 0 **meal:** 0 **country:** 0 **market_segment:** 0 **distribution_channel:** 0 **is_repeated_guest:** 0 **previous_cancellations:** 0 **previous_bookings_not_canceled:** 0 **reserved_room_type:** 0 **assigned_room_type:** 0 **booking_changes:** 0 **deposit_type:** 0 **agent:** 15083 **company:** 98201 **days_in_waiting_list:** 0 **customer_type:** 0 **adr:** 0 **required_car_parking_spaces:** 0 **total_of_special_requests:** 0 **reservation_status:** 0 **reservation_status_date:** 0

In [16]:

```
# childen, agent and company have missing values
dat_eu <- na.locf(na.locf(dat_eu), fromLast = TRUE) # backward fill NA in other
colSums(is.na(dat_eu)) # NO NAs right now
```

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

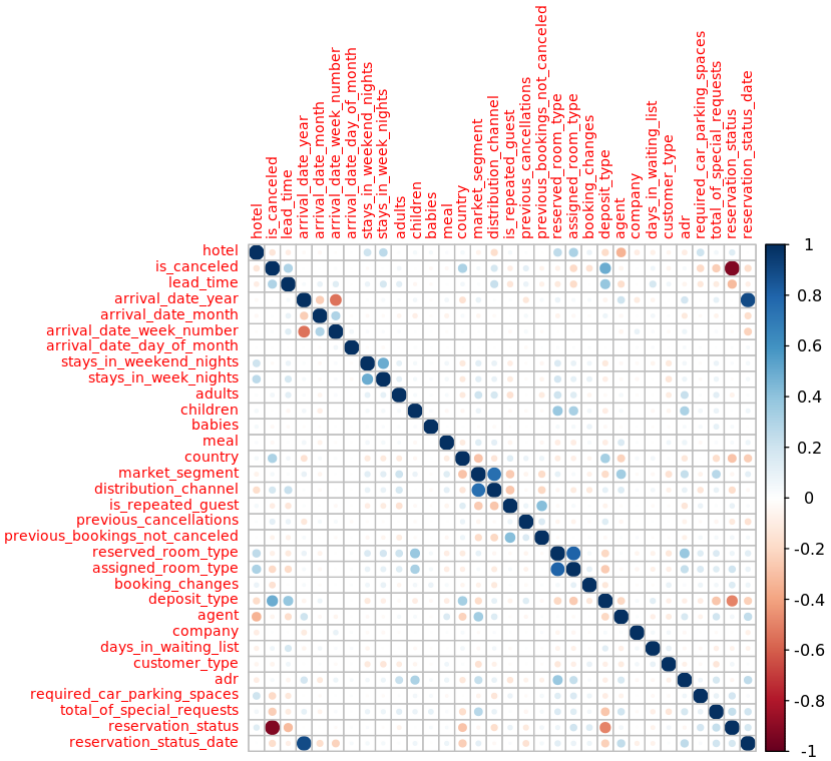
In [ ]:

EDA

Correlation Matrix

In [19]:
```
# convert all factors to numeric
df <- dat_eu %>% mutate_if(is.factor, as.numeric)
```

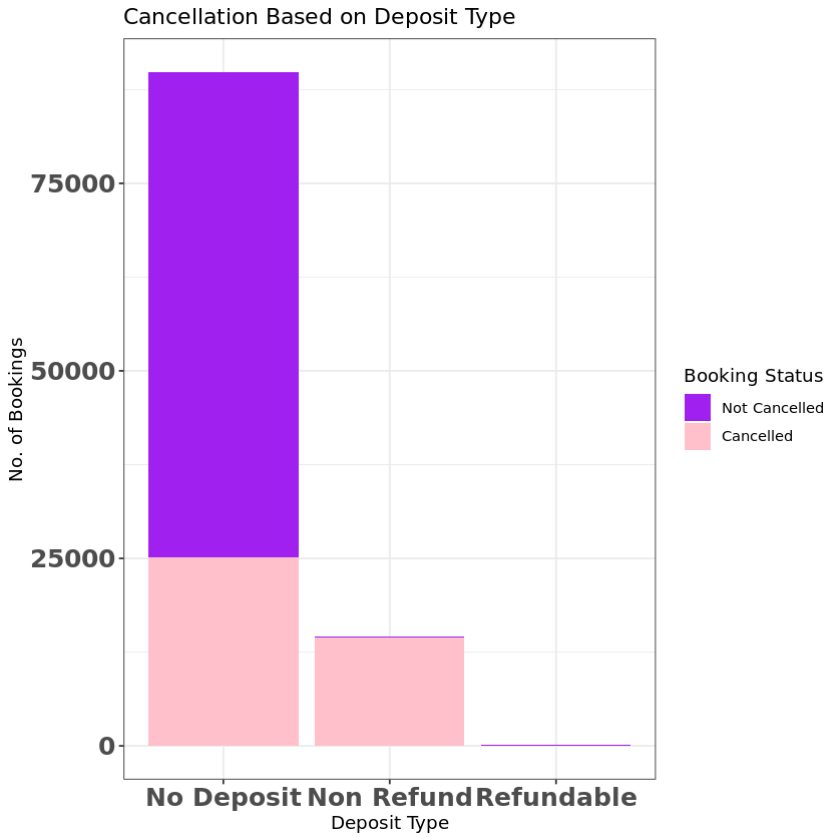
In [20]:
```
# heatmap of correlation among variables
corrplot(cor(df),tl.cex = 0.7)
```



Positively Correlated Variables

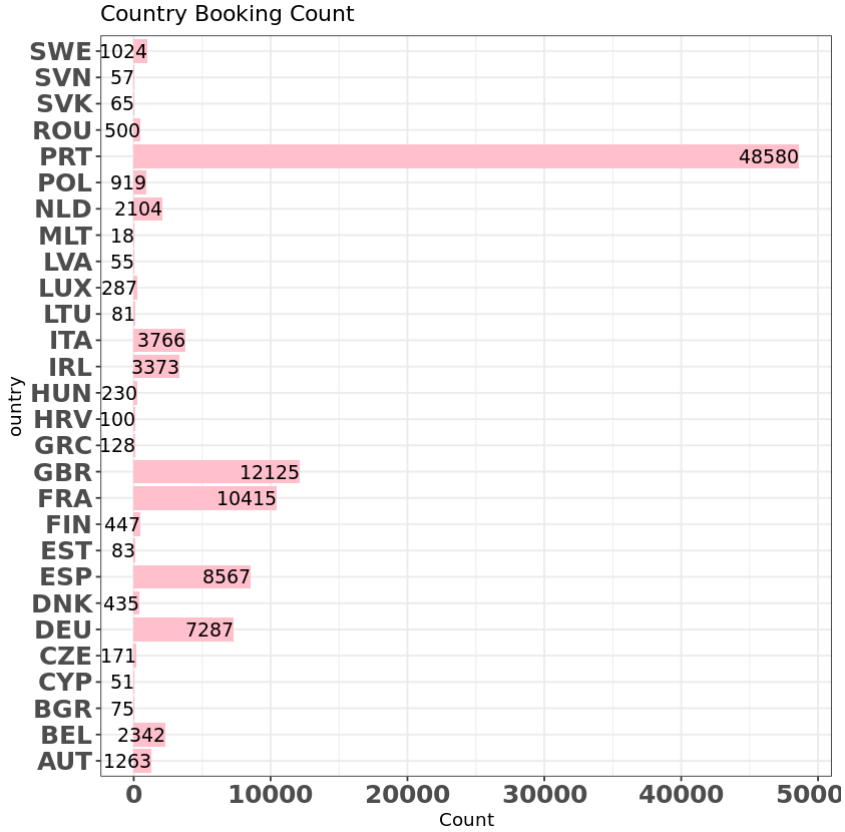
In [21]:
```
# Cancellations Based on Deposit Type; Strongest Positive Correlation

ggplot(dat_eu,aes(x=factor(deposit_type),fill=factor(is_canceled)))+

geom_bar()+theme(axis.text.x = element_text(face="bold", size=15),axis.text.y =

labs(
    title = "Cancellation Based on Deposit Type",
    x = "Deposit Type",
    y = "No. of Bookings",size=15) +

scale_fill_manual(
    name = "Booking Status",
    breaks = c("0", "1"),
```

```
    labels = c("Not Cancelled", "Cancelled"),
    values = c("0" = "purple", "1"="pink")
  )
```



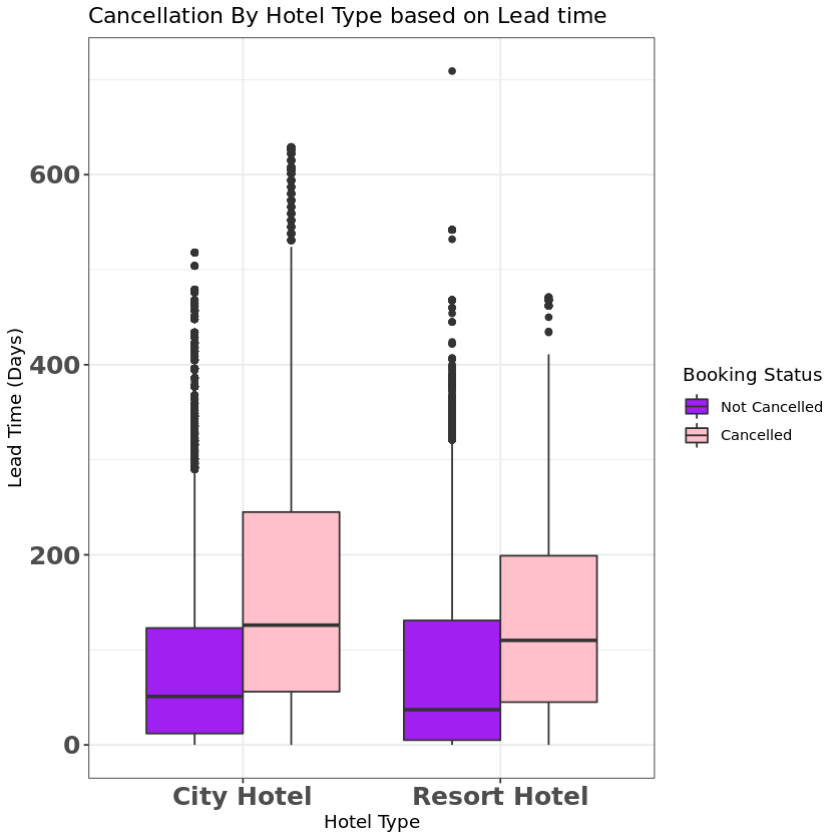
In [22]:

```
# Country

ggplot(dat_eu, aes(x = country)) +

geom_bar(fill = "pink") +

geom_text(stat = "count", aes(label = ..count..), hjust = 1,size=4) +

coord_flip() + labs(title = "Country Booking Count",
                    x = "ountry",
                    y = "Count") +

theme(axis.text.x = element_text(face="bold", size=15),axis.text.y = element_tex
```



In [23]:

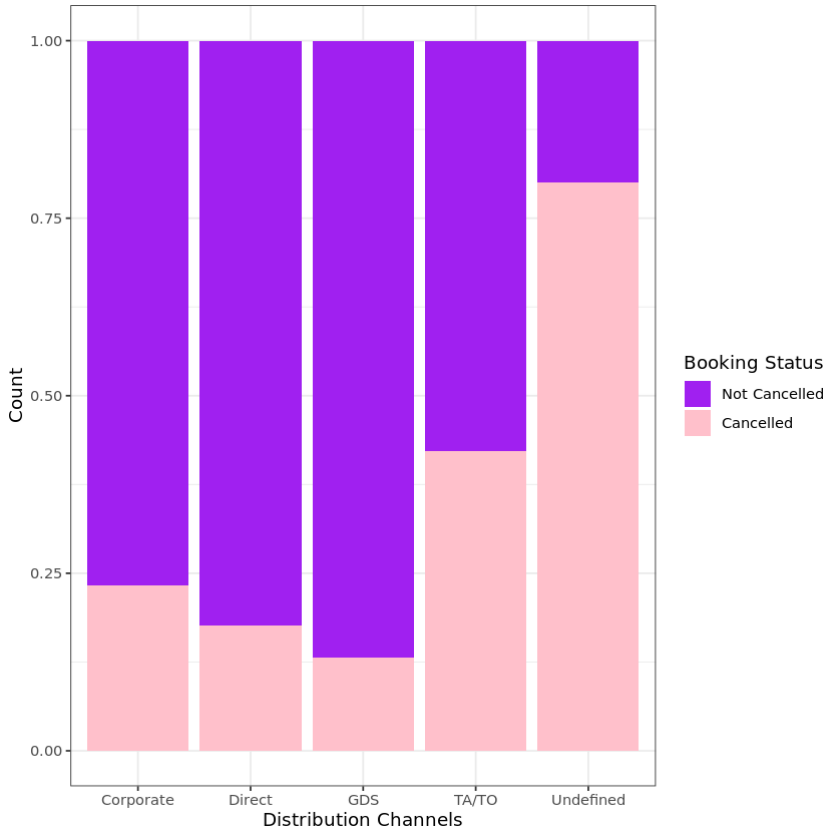
```
# Lead Time and Hotel Type
ggplot(data = dat_eu, aes(x = hotel,y = lead_time,fill = factor(is_canceled))) +

geom_boxplot(position = position_dodge()) +

labs(
    title = "Cancellation By Hotel Type based on Lead time",
    x = "Hotel Type",
    y = "Lead Time (Days)") +

theme(axis.text.x = element_text(face="bold", size=15),axis.text.y = element_tex

scale_fill_manual(
    name = "Booking Status",
    breaks = c("0", "1"),
    labels = c("Not Cancelled", "Cancelled"),
    values = c("0" = "purple", "1"="pink")
  )
```



In [24]:

```
# Distribution Channels
ggplot(dat_eu, aes(distribution_channel, fill = factor(is_canceled))) +
  # Add a bar layer with position "fill"
  geom_bar(position = "fill") +
  # Add a brewer fill scale with default palette
  scale_fill_brewer() +
  scale_x_discrete("Distribution Channels") +
  scale_y_continuous("Count")+
  scale_fill_manual(
    name = "Booking Status",
    breaks = c("0", "1"),
    labels = c("Not Cancelled", "Cancelled"),
    values = c("0" = "purple", "1"="pink")
  )
```

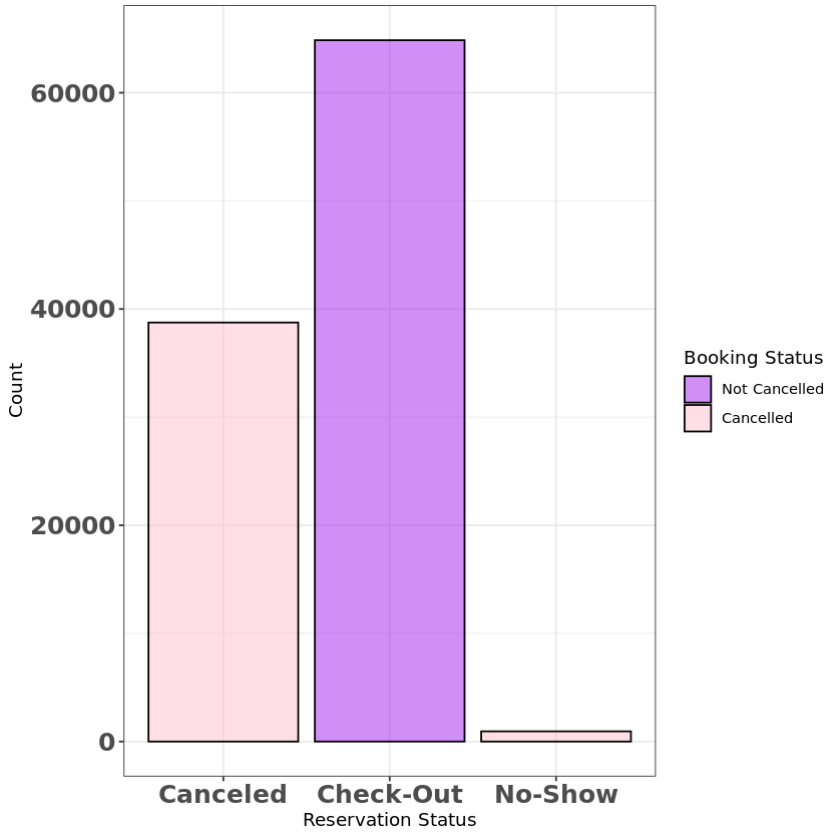
```
Scale for 'fill' is already present. Adding another scale for 'fill', which
will replace the existing scale.
```



Negatively Correlated Variables

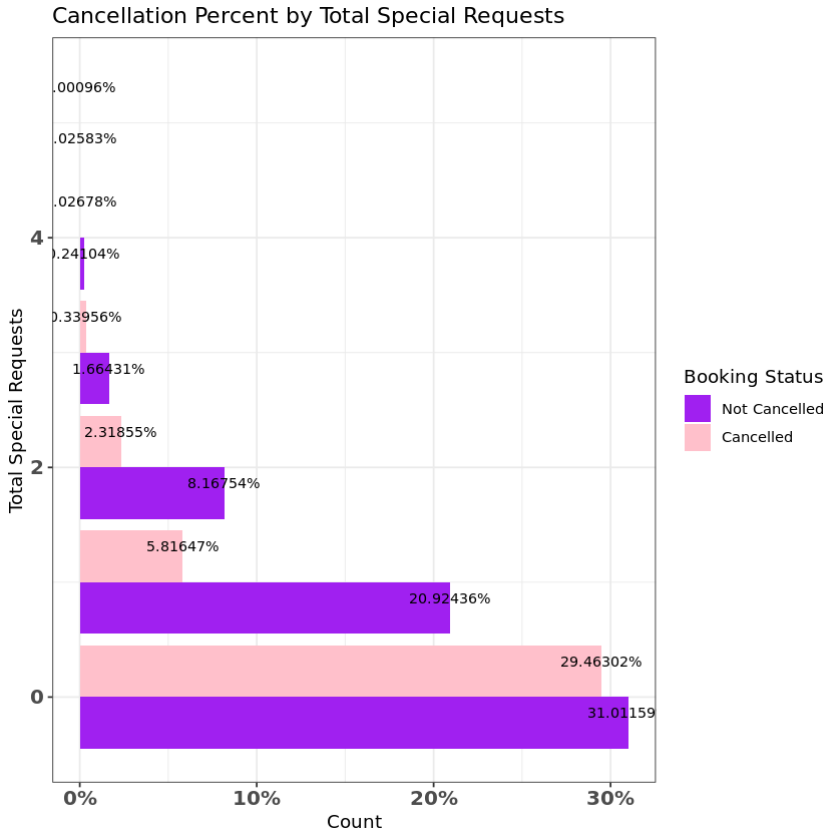
In [25]:

```
#Cancellations By Reservation Status; strongest negative correlation
ggplot(dat_eu,aes(x=factor(reservation_status),fill=factor(is_canceled))) +
geom_bar(col ="black",alpha=0.5) +
theme(axis.text.x = element_text(face="bold", size=15),axis.text.y = element_tex
scale_x_discrete("Reservation Status") +
scale_y_continuous("Count")+
scale_fill_manual(
    name = "Booking Status",
    breaks = c("0", "1"),
    labels = c("Not Cancelled", "Cancelled"),
    values = c("0" = "purple", "1"="pink")
  )
```



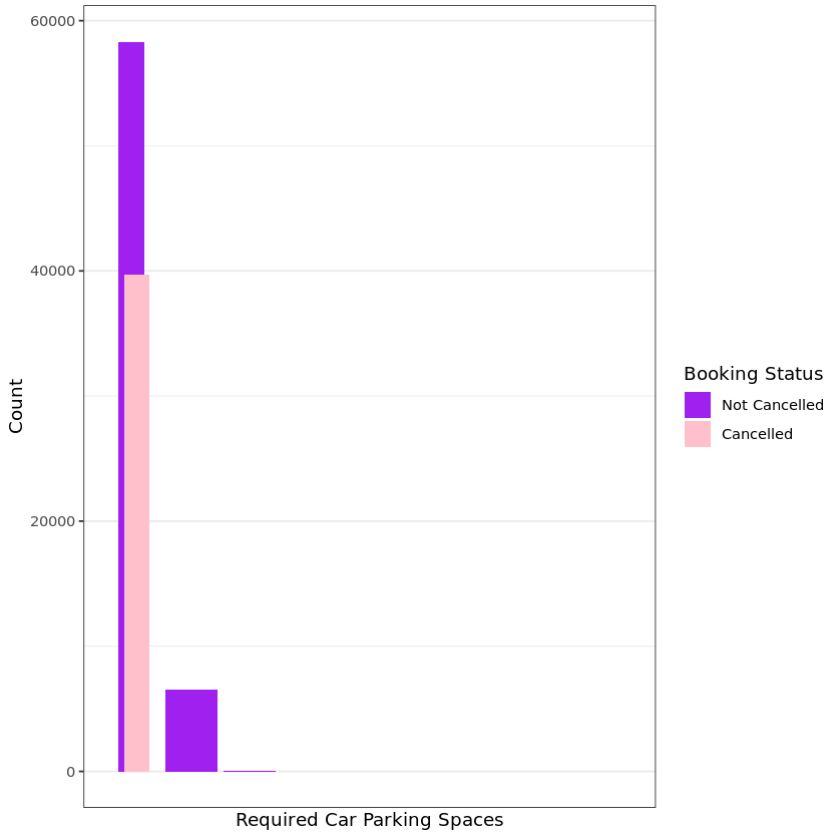
In [26]:

```
# Total special request
ggplot(data = dat_eu,aes(x = total_of_special_requests,y = prop.table(stat(count
         label = scales::percent(prop.table(stat(count))))) +

geom_bar(position = position_dodge()) +

geom_text(stat = "count",position = position_dodge(.9),vjust = -0.5,size = 3) +

theme(axis.text.x = element_text(face="bold", size=12),axis.text.y = element_tex

scale_y_continuous(labels = scales::percent) + coord_flip() +

labs(title = "Cancellation Percent by Total Special Requests",x = "Total Special

scale_fill_manual(
    name = "Booking Status",
    breaks = c("0", "1"),
    labels = c("Not Cancelled", "Cancelled"),
    values = c("0" = "purple", "1"="pink")
  )
```



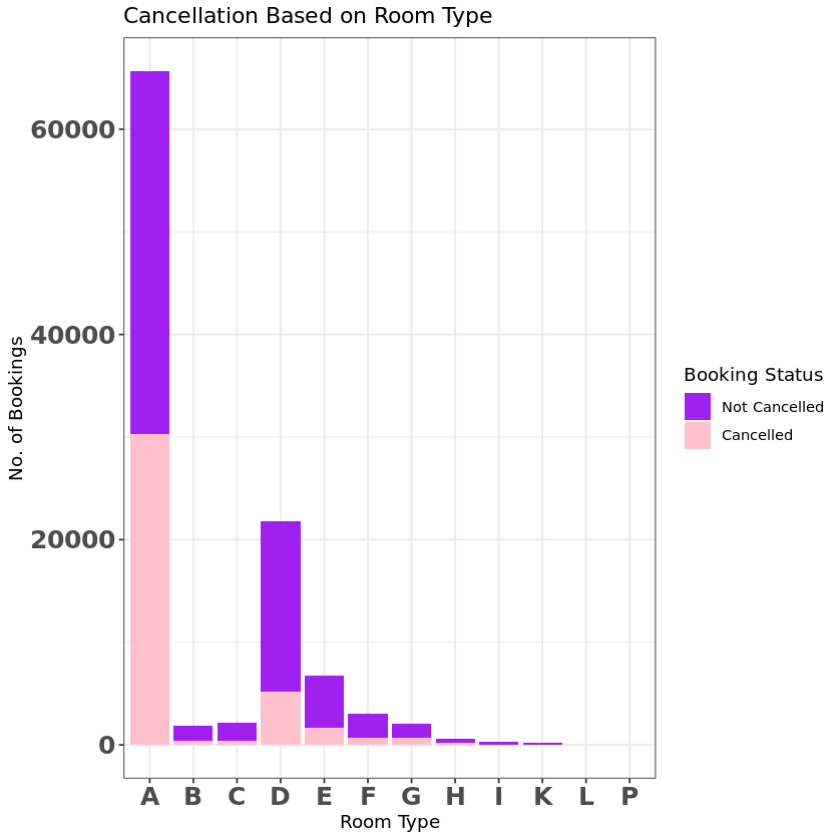
In [27]:

```
#Required Car Parking Spaces
ggplot(dat_eu, aes(required_car_parking_spaces, fill = factor(is_canceled))) +
  # Change position to use the functional form, with width 0.2
  geom_bar(position = position_dodge(width = 0.2)) +
  scale_x_discrete("Required Car Parking Spaces") +
  scale_y_continuous("Count")+
  scale_fill_manual(
    name = "Booking Status",
    breaks = c("0", "1"),
    labels = c("Not Cancelled", "Cancelled"),
    values = c("0" = "purple", "1"="pink")
  )
```



In [28]:

```
# Assigned room type
ggplot(dat_eu,aes(x=factor(assigned_room_type),fill=factor(is_canceled)))+

geom_bar()+

theme(axis.text.x = element_text(face="bold", size=15),axis.text.y = element_tex

labs(
    title = "Cancellation Based on Room Type",
    x = "Room Type",
    y = "No. of Bookings",size=15
  ) +

scale_fill_manual(
    name = "Booking Status",
    breaks = c("0", "1"),
    labels = c("Not Cancelled", "Cancelled"),
    values = c("0" = "purple", "1"="pink")
  )
```

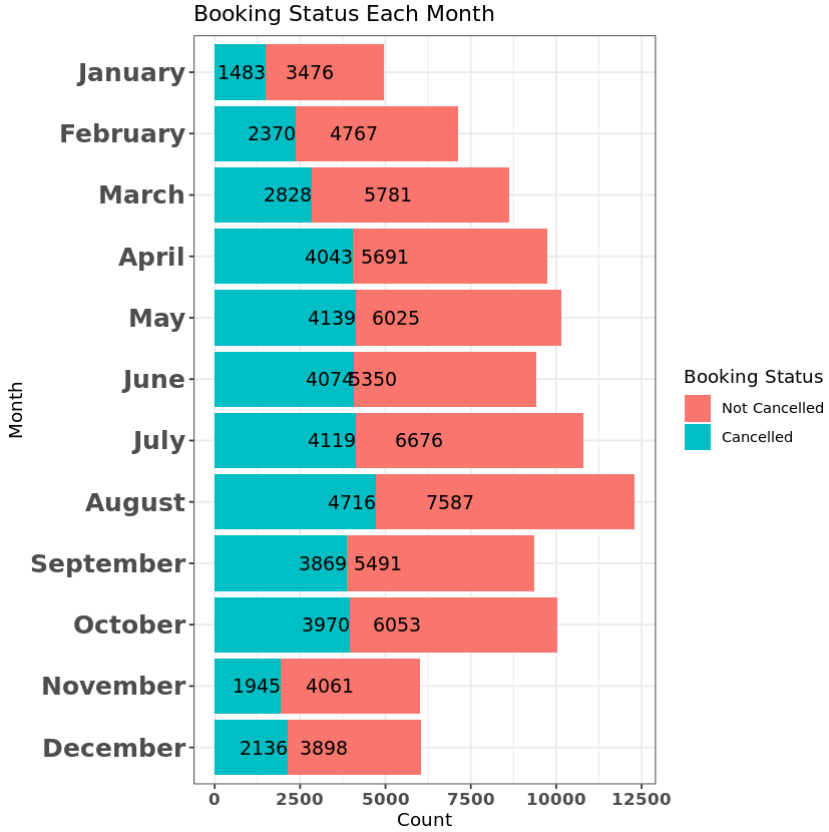


In [ ]:

Interested Categories

In [29]:

```
# Cancellations Per Month

# Ordering y axis
dat_eu$arrival_date_month  = factor(dat_eu$arrival_date_month, levels=c("Decembe

ggplot(dat_eu, aes(arrival_date_month, fill = factor(is_canceled))) +
geom_bar() + geom_text(stat = "count", aes(label = ..count..), hjust = 1,size=4)
coord_flip() + scale_fill_discrete(
    name = "Booking Status",
    breaks = c("0", "1"),
    label = c("Not Cancelled", "Cancelled")) +
labs(title = "Booking Status Each Month",
       x = "Month",
       y = "Count") +
theme(axis.text.x = element_text(face="bold", size=10),axis.text.y = element_tex
```



In [ ]:

Setup for Train and Test Sets

In [ ]:
```
cols <- c("agent", "company", "reservation_status_date", "country", "arrival_dat
          "arrival_date_month", "arrival_date_week_number", "arrival_date_day_of
          "assigned_room_type", "reservation_status")
dat.clean <- dat[, setdiff(colnames(dat), cols)]
dat.clean$is_canceled = as.factor(dat.clean$is_canceled)

training.samples <- dat.clean$is_canceled %>% createDataPartition(p = 0.7, list

train  <- dat.clean[training.samples, ]
test <- dat.clean[-training.samples, ]
```

In [ ]:

Additional Information

In [ ]:
```
# How we Filtered the data from Original Dataset
# Original Dataset
dat <- read.csv("hotel_bookings.csv", stringsAsFactors = T)

#Filter out all EU countries
eu_country <- c('AUT', 'BEL', 'BGR', 'HRV', 'CYP', 'CZE', 'DNK', 'EST',
                'FIN', 'FRA', 'DEU', 'GRC', 'HUN', 'IRL', 'ITA', 'LVA',
                'LTU', 'LUX', 'MLT', 'NLD', 'POL', 'PRT', 'ROU', 'SVK',
```

```
                'SVN', 'ESP', 'SWE', 'GBR')

for (i in dat['country']){
  check <- i %in% eu_country
  check_country <- c(check)
}

dat['check_country'] <- check_country

dat_eu <- subset(dat, check_country == TRUE)

dat_eu <- dat_eu[,1:32]

# Original Cleaning of Data
# childen, agent and company have missing values
dat_eu <- na.locf(na.locf(dat_eu), fromLast = TRUE) # backward fill NA in other
colSums(is.na(dat_eu)) # NO NAs right now
```

In [ ]:

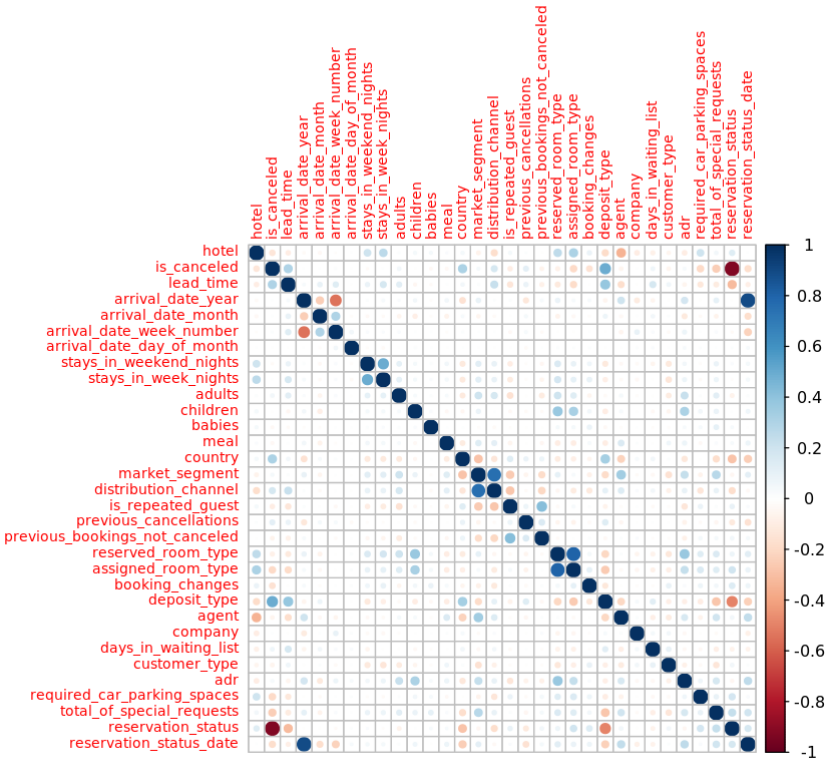
In [ ]:

Linear, Lasso, Ridge Regression

Linear regression

In [18]:
```
cols <- c("agent", "company", "reservation_status_date", "country", "arrival_dat
          "arrival_date_month", "arrival_date_week_number", "arrival_date_day_of
          "assigned_room_type", "reservation_status")
dat.clean <- dat[, setdiff(colnames(dat), cols)]
```

In [86]:
```
# heatmap of correlation among variables
corrplot(cor(df),tl.cex = 0.7)
```



In [19]:
```
head(dat.clean)
```

	hotel	is_canceled	lead_time	stays_in_weekend_nights	stays_in_week_nights	adults	children
	<fct>	<int>	<int>	<int>	<int>	<int>	<int>
1	Resort Hotel	0	342	0	0	2	0
2	Resort Hotel	0	737	0	0	2	0
3	Resort Hotel	0	7	0	1	1	0
4	Resort Hotel	0	13	0	1	1	0
5	Resort Hotel	0	14	0	2	2	0
6	Resort Hotel	0	14	0	2	2	0

In [24]:
```
## The dataset is split into train and test set in 70:30 ratio with the is_cance
set.seed(100)
training.samples <- dat.clean$is_canceled %>%
  createDataPartition(p = 0.7, list = FALSE)
train  <- dat.clean[training.samples, ]
test <- dat.clean[-training.samples, ]
```

```
y.train <- train$is_canceled
y.test <- test$is_canceled
```

In [14]:

```
f1 <- as.formula(is_canceled ~ lead_time + total_of_special_requests + deposit_t
fit.lm1 <- lm(f1, train)
yhat.train.lm1 <- predict(fit.lm1)
mse.train.lm1 <- mean((y.train - yhat.train.lm1)^2)
mse.train.lm1
```

0.1651200862462

In [15]:

```
yhat.test.lm1 <- predict(fit.lm1, test)
mse.test.lm1 <- mean((y.test - yhat.test.lm1)^2)
mse.test.lm1
```

0.162776445251584

In [18]:

```
summary(fit.lm1)
```

```
Call:
lm(formula = f1, data = train)

Residuals:
     Min       1Q   Median       3Q      Max
-1.00024 -0.29891 -0.21432  0.09996  1.75181

Coefficients:
                               Estimate Std. Error t value Pr(>|t|)
(Intercept)                   2.927e-01  2.505e-03 116.840  < 2e-16 ***
lead_time                     5.165e-04  1.492e-05  34.622  < 2e-16 ***
total_of_special_requests    -6.133e-02  2.001e-03 -30.647  < 2e-16 ***
deposit_typeNon Refund        5.908e-01  4.900e-03 120.578  < 2e-16 ***
deposit_typeRefundable       -1.141e-01  3.971e-02  -2.872  0.00408 **
required_car_parking_spaces -2.645e-01  6.211e-03 -42.582  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4064 on 73190 degrees of freedom
Multiple R-squared:  0.2987,    Adjusted R-squared:  0.2987
F-statistic:  6235 on 5 and 73190 DF,  p-value: < 2.2e-16
```

In [20]:

```
predicted<-predict(fit.lm1,newdata=test)
TAB<- table(test$is_canceled,predicted > 0.5)
mcrate <- sum(diag(TAB))/sum(TAB)
mcrate
```

0.760145366444579

In [25]:

```
# Fit the linear regression on the training data
lm_reg<-lm(is_canceled~.,data=train)
summary(lm_reg)
```

```
Call:
lm(formula = is_canceled ~ ., data = train)

Residuals:
    Min      1Q  Median      3Q     Max
```

```
-2.7533 -0.2827 -0.1423  0.1795  1.7938

Coefficients:
                                  Estimate Std. Error t value Pr(>|t|)
(Intercept)                      0.0875019  0.0322688   2.712 0.006696 **
hotelResort Hotel                0.0031035  0.0037942   0.818 0.413381
lead_time                        0.0005811  0.0000160  36.306  < 2e-16 ***
stays_in_weekend_nights          0.0068137  0.0017011   4.006 6.19e-05 ***
stays_in_week_nights             0.0062422  0.0009437   6.615 3.75e-11 ***
adults                           0.0242695  0.0025762   9.421  < 2e-16 ***
children                         0.0279937  0.0049654   5.638 1.73e-08 ***
babies                           0.0621775  0.0138640   4.485 7.31e-06 ***
mealFB                           0.0497322  0.0171374   2.902 0.003709 **
mealHB                          -0.0255700  0.0046711  -5.474 4.41e-08 ***
mealSC                           0.0043094  0.0059922   0.719 0.472043
mealUndefined                   -0.0887132  0.0144774  -6.128 8.96e-10 ***
market_segmentComplementary      0.0963021  0.0373923   2.575 0.010013 *
market_segmentCorporate         -0.0055924  0.0316963  -0.176 0.859952
market_segmentDirect             0.0050558  0.0339825   0.149 0.881731
market_segmentGroups             0.0006141  0.0328510   0.019 0.985085
market_segmentOffline TA/TO     -0.0768459  0.0328420  -2.340 0.019293 *
market_segmentOnline TA          0.1340365  0.0328276   4.083 4.45e-05 ***
distribution_channelDirect      -0.0360443  0.0136955  -2.632 0.008494 **
distribution_channelGDS         -0.2030654  0.0395696  -5.132 2.88e-07 ***
distribution_channelTA/TO        0.0237480  0.0108562   2.188 0.028709 *
distribution_channelUndefined    0.1194060  0.3886331   0.307 0.758657
is_repeated_guest                0.0030979  0.0091784   0.338 0.735724
previous_cancellations           0.0252228  0.0016572  15.220  < 2e-16 ***
previous_bookings_not_canceled  -0.0036471  0.0010826  -3.369 0.000756 ***
reserved_room_typeB              0.0101038  0.0161989   0.624 0.532804
reserved_room_typeC              0.0356077  0.0178723   1.992 0.046338 *
reserved_room_typeD             -0.0091992  0.0044155  -2.083 0.037218 *
reserved_room_typeE              0.0234119  0.0068559   3.415 0.000638 ***
reserved_room_typeF             -0.0401938  0.0111479  -3.606 0.000312 ***
reserved_room_typeG              0.0142842  0.0133112   1.073 0.283232
reserved_room_typeH              0.0177806  0.0212979   0.835 0.403803
reserved_room_typeL              0.1321693  0.1737495   0.761 0.446846
reserved_room_typeP              0.6862218  0.2746533   2.499 0.012474 *
booking_changes                 -0.0544374  0.0023627 -23.040  < 2e-16 ***
deposit_typeNon Refund           0.6099344  0.0061574  99.058  < 2e-16 ***
deposit_typeRefundable          -0.0325312  0.0375725  -0.866 0.386589
days_in_waiting_list            -0.0001033  0.0000815  -1.268 0.204947
customer_typeGroup              -0.0269867  0.0227077  -1.188 0.234665
customer_typeTransient           0.0641824  0.0079368   8.087 6.22e-16 ***
customer_typeTransient-Party     0.0271360  0.0085004   3.192 0.001412 **
adr                              0.0005600  0.0000344  16.279  < 2e-16 ***
required_car_parking_spaces     -0.2553529  0.0061192 -41.730  < 2e-16 ***
total_of_special_requests       -0.1150879  0.0021027 -54.734  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3883 on 73150 degrees of freedom
  (2 observations deleted due to missingness)
Multiple R-squared:   0.36,     Adjusted R-squared:  0.3596
F-statistic: 956.8 on 43 and 73150 DF,  p-value: < 2.2e-16
```

In [26]:

```
predicted<-predict(lm_reg,newdata=test)
```

In [27]:

```
TAB<- table(test$is_canceled,predicted > 0.5)
TAB
```

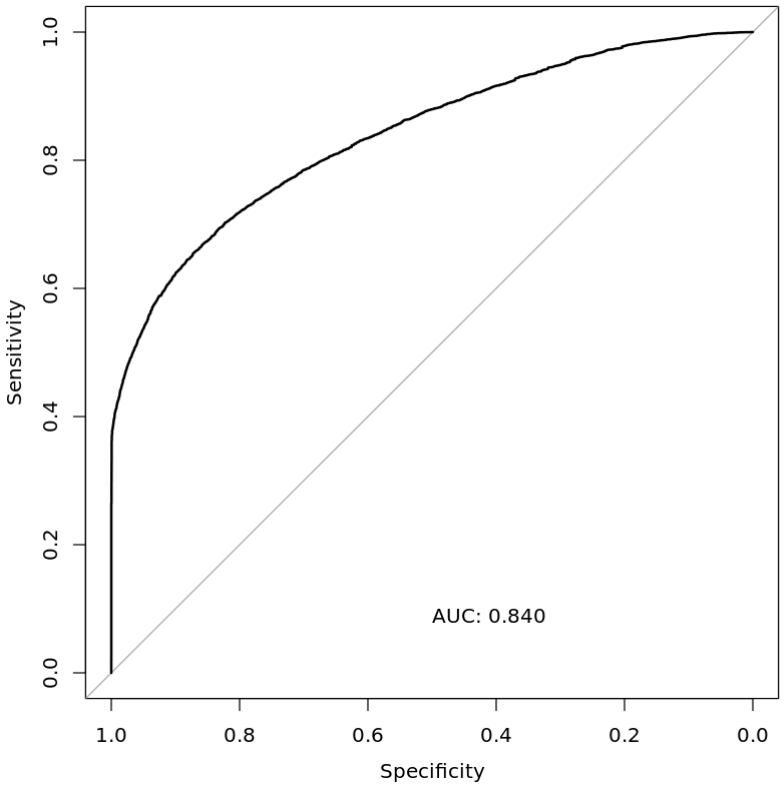
```
  FALSE  TRUE
0 18731   693
1  5923  6020
```

In [28]:
```
mcrate <- sum(diag(TAB))/sum(TAB)
mcrate
```

0.78907769311697

In [31]:
```
# Plot the ROC curve to check the performance of final model
lm_roc<-roc(y.test,predicted,auc=TRUE)
plot(lm_roc,print.auc=TRUE,print.auc.y=.1)
```

```
Setting levels: control = 0, case = 1

Setting direction: controls < cases
```



Ridge Regression

In [141…
```
# Convert the type of response from integer to categorical (dummy variable)
dat.clean$is_canceled <- as.factor(dat.clean$is_canceled)
```

In [142…
```
# The dataset is split into train and test set in 70:30 ratio with the is_cancel
set.seed(100)
training.samples <- dat.clean$is_canceled %>%
  createDataPartition(p = 0.7, list = FALSE)
train  <- dat.clean[training.samples, ]
test <- dat.clean[-training.samples, ]
```

In [143…
```
f1 <- as.formula(is_canceled ~.)
x1.train <- model.matrix(f1, train)[, -1]
y.train <- train$is_canceled
x1.test <- model.matrix(f1, test)[, -1]
y.test <- test$is_canceled
```

In [42]:
```
# Fit ridge regression model with 10 folds cross validation on the training set
fit.ridge <- cv.glmnet(x1.train, y.train, alpha = 0, nfolds = 10, family = "bino
ridge_pred_train <- predict(fit.ridge, x1.train, s = fit.ridge$lambda.1se, type=
confusionMatrix(as.factor(ridge_pred_train),y.train)
```

```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 43375 12651
         1  2034 15136

               Accuracy : 0.7994
                 95% CI : (0.7965, 0.8023)
    No Information Rate : 0.6204
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.54

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

            Sensitivity : 0.9552
            Specificity : 0.5447
         Pos Pred Value : 0.7742
         Neg Pred Value : 0.8815
             Prevalence : 0.6204
         Detection Rate : 0.5926
   Detection Prevalence : 0.7654
      Balanced Accuracy : 0.7500

       'Positive' Class : 0
```

In [22]:
```
# Prediction on the test set
ridge_pred_class <- predict(fit.ridge, x1.test, s = fit.ridge$lambda.1se, type='
confusionMatrix(as.factor(ridge_pred_class),y.test)
```

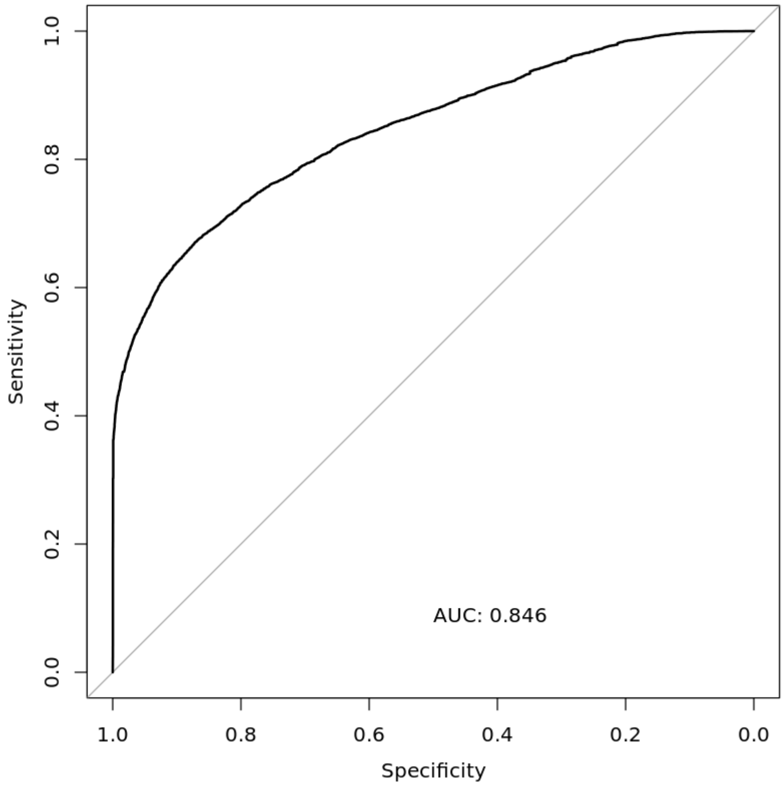
```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 18537  5404
         1   924  6504

               Accuracy : 0.7983
                 95% CI : (0.7938, 0.8027)
    No Information Rate : 0.6204
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.538

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

            Sensitivity : 0.9525
            Specificity : 0.5462
```

```
            Pos Pred Value : 0.7743
            Neg Pred Value : 0.8756
                Prevalence : 0.6204
            Detection Rate : 0.5909
      Detection Prevalence : 0.7632
         Balanced Accuracy : 0.7494

          'Positive' Class : 0
```

In [87]:

```
# Plot the ROC curve to check the performance of final model
ridge_pred_prob<-predict(fit.ridge, x1.test, s = fit.ridge$lambda.1se,type="resp
ridge_roc<-roc(y.test,ridge_pred_prob,auc=TRUE)
plot(ridge_roc,print.auc=TRUE,print.auc.y=.1)
```

```
Installing package into ‘/home/jupyter/.R/library’
(as ‘lib’ is unspecified)

Type 'citation("pROC")' for a citation.


Attaching package: ‘pROC’


The following objects are masked from ‘package:stats’:

    cov, smooth, var


Setting levels: control = 0, case = 1

Warning message in roc.default(y.test, ridge_pred_prob, auc = TRUE):
“Deprecated use a matrix as predictor. Unexpected results may be produced, pleas
e pass a numeric vector.”
Setting direction: controls < cases
```



Lasso Regression

Create a random subsample to fit the lasso regression model first since our dataset is too large

In [145…

```
train.sample.size <- 15000
train.sample <- train[sample(nrow(train), train.sample.size),]

x1.train.sample <- model.matrix(f1, train.sample)[, -1]
y.train.sample <- train.sample$is_canceled
```

Fit lasso regression model on training sample set

In [92]:

```
fit.lasso <- cv.glmnet(x1.train.sample, y.train.sample, alpha = 1,nfolds = 10, f
```

```
Warning message:
"from glmnet Fortran code (error code -76); Convergence for 76th lambda value no
t reached after maxit=100000 iterations; solutions for larger lambdas returned"
Warning message:
"from glmnet Fortran code (error code -75); Convergence for 75th lambda value no
t reached after maxit=100000 iterations; solutions for larger lambdas returned"
Warning message:
"from glmnet Fortran code (error code -75); Convergence for 75th lambda value no
t reached after maxit=100000 iterations; solutions for larger lambdas returned"
Warning message:
"from glmnet Fortran code (error code -75); Convergence for 75th lambda value no
t reached after maxit=100000 iterations; solutions for larger lambdas returned"
Warning message:
"from glmnet Fortran code (error code -75); Convergence for 75th lambda value no
t reached after maxit=100000 iterations; solutions for larger lambdas returned"
Warning message:
"from glmnet Fortran code (error code -75); Convergence for 75th lambda value no
t reached after maxit=100000 iterations; solutions for larger lambdas returned"
```

```
Warning message:
“from glmnet Fortran code (error code -75); Convergence for 75th lambda value no
t reached after maxit=100000 iterations; solutions for larger lambdas returned”
```

Get the coefficients of variables

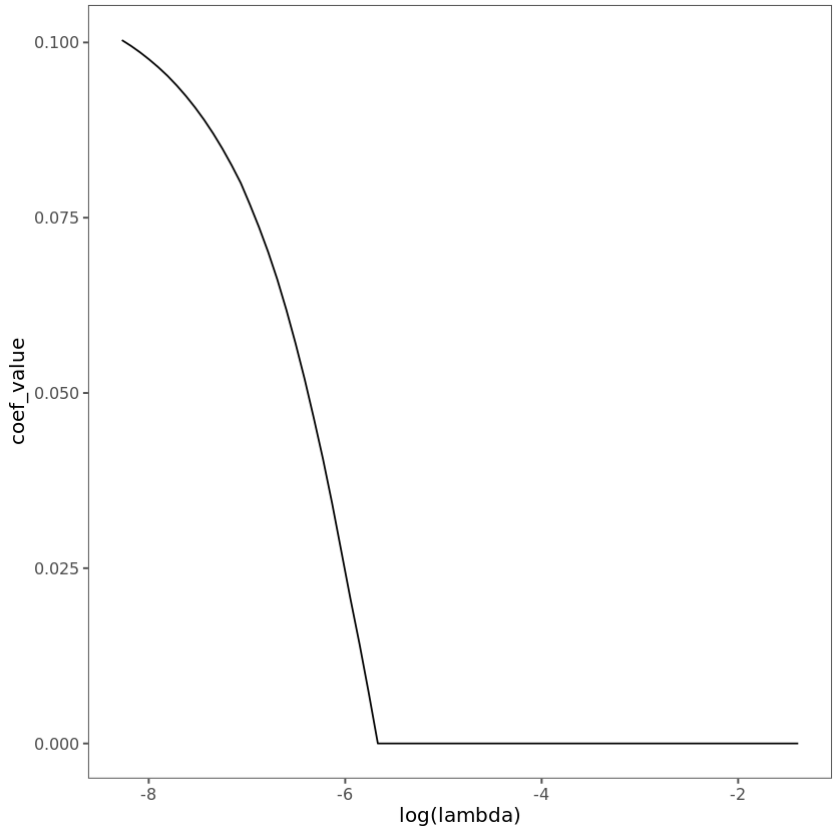
In [112…

```
lasso.coef <- predict(fit.lasso,
                      type = "coefficients",
                      s = fit.lasso$lambda)
```

Plot the coefficients versus the log(lambda)

In [113…

```
to_plot <- data.table(
  lambda = fit.lasso$lambda,
  coef_value = lasso.coef[2, ]
)
ggplot(to_plot, aes(log(lambda), coef_value)) +
  geom_line() +
  theme_few()
```



Prediction on the training set

For the result of the training model, we can observe that our model perfectly preticted 12204 out of 15000, giving the accuracy of 81.36%. There were 2796 cases that our model got wrong. These 2796 cases were divided between False negatives and false positives as 425 and 2371 respectively.

In [157…

```
lasso_pred_train <- predict(fit.lasso, x1.train.sample, s = fit.lasso$lambda.1se
confusionMatrix(as.factor(lasso_pred_train),y.train.sample)
```

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 8865 2371
         1  425 3339

               Accuracy : 0.8136
                 95% CI : (0.8073, 0.8198)
    No Information Rate : 0.6193
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5769

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

            Sensitivity : 0.9543
            Specificity : 0.5848
         Pos Pred Value : 0.7890
         Neg Pred Value : 0.8871
             Prevalence : 0.6193
         Detection Rate : 0.5910
   Detection Prevalence : 0.7491
      Balanced Accuracy : 0.7695

       'Positive' Class : 0
```

Prediction on the test set

For the result of the testing model, we can observe that our model perfectly preticted 25080 out of 31368, giving the accuracy of 79.95%. There were 6288 cases that our model got wrong. These 6288 cases were divided between False negatives and false positives as 1143 and 5145 respectively.

In [160…

```
lasso_pred_class <- predict(fit.lasso, x1.test, s = fit.lasso$lambda.1se, type='
confusionMatrix(as.factor(lasso_pred_class),y.test)
```

```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 18317  5145
         1  1143  6763

               Accuracy : 0.7995
                 95% CI : (0.7951, 0.804)
    No Information Rate : 0.6204
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5447

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

            Sensitivity : 0.9413
            Specificity : 0.5679
         Pos Pred Value : 0.7807
         Neg Pred Value : 0.8554
             Prevalence : 0.6204
         Detection Rate : 0.5839
   Detection Prevalence : 0.7480
      Balanced Accuracy : 0.7546
```

```
'Positive' Class : 0
```

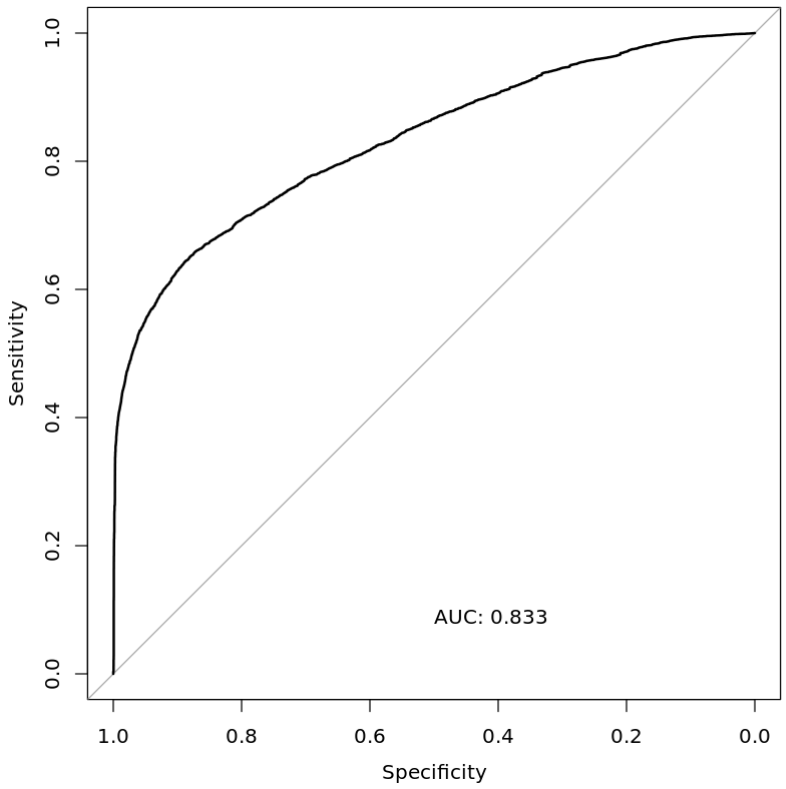
Plot the ROC curve to check the performance of final model

As evident, our model has an AUC of 0.833.

In [166…

```
lasso_pred_prob <- predict(fit.lasso, x1.test, s = fit.lasso$lambda.1se, type='r
lasso_roc<-roc(y.test,lasso_pred_prob,auc=TRUE)
plot(lasso_roc,print.auc=TRUE,print.auc.y=.1)
```

```
Setting levels: control = 0, case = 1

Warning message in roc.default(y.test[-1], lasso_pred_prob, auc = TRUE):
“Deprecated use a matrix as predictor. Unexpected results may be produced, pleas
e pass a numeric vector.”
Setting direction: controls < cases
```



Logistic Regression

In [8]:

```
# Data Selection
cols <- c("agent", "company", "reservation_status_date", "country", "arrival_dat
          "arrival_date_month", "arrival_date_week_number", "arrival_date_day_of
          "assigned_room_type")
dat.clean <- dat[, setdiff(colnames(dat), cols)]
library(MASS)
library(caret)
training.samples <- dat.clean$is_canceled %>%
  createDataPartition(p = 0.7, list = FALSE)
train  <- dat.clean[training.samples, ]
test <- dat.clean[-training.samples, ]
```

In [9]:

```
# Fit the model
model <- glm(is_canceled~., data = train, family = binomial) %>%
  stepAIC(trace = T)
summary(model)
summary(model)
```

```
Warning message:
“glm.fit: fitted probabilities numerically 0 or 1 occurred”
Start:  AIC=61876.59
is_canceled ~ hotel + lead_time + stays_in_weekend_nights + stays_in_week_nights
+
    adults + children + babies + meal + market_segment + distribution_channel +
    is_repeated_guest + previous_cancellations + previous_bookings_not_canceled
+
    reserved_room_type + booking_changes + deposit_type + days_in_waiting_list +
    customer_type + adr + required_car_parking_spaces + total_of_special_request
s



Warning message:
“glm.fit: fitted probabilities numerically 0 or 1 occurred”
Warning message:
“glm.fit: algorithm did not converge”
Warning message:
“glm.fit: fitted probabilities numerically 0 or 1 occurred”
Warning message:
“glm.fit: fitted probabilities numerically 0 or 1 occurred”
Warning message:
“glm.fit: fitted probabilities numerically 0 or 1 occurred”
Warning message:
“glm.fit: fitted probabilities numerically 0 or 1 occurred”
Warning message:
“glm.fit: fitted probabilities numerically 0 or 1 occurred”
Warning message:
“glm.fit: fitted probabilities numerically 0 or 1 occurred”
Warning message:
“glm.fit: fitted probabilities numerically 0 or 1 occurred”
Warning message:
“glm.fit: fitted probabilities numerically 0 or 1 occurred”
Warning message:
“glm.fit: fitted probabilities numerically 0 or 1 occurred”
Warning message:
“glm.fit: fitted probabilities numerically 0 or 1 occurred”
Warning message:
“glm.fit: fitted probabilities numerically 0 or 1 occurred”
Warning message:
“glm.fit: fitted probabilities numerically 0 or 1 occurred”
Warning message:
“glm.fit: fitted probabilities numerically 0 or 1 occurred”
Warning message:
“glm.fit: fitted probabilities numerically 0 or 1 occurred”
Warning message:
“glm.fit: algorithm did not converge”
Warning message:
“glm.fit: fitted probabilities numerically 0 or 1 occurred”
Warning message:
“glm.fit: fitted probabilities numerically 0 or 1 occurred”
Warning message:
“glm.fit: fitted probabilities numerically 0 or 1 occurred”
Warning message:
“glm.fit: fitted probabilities numerically 0 or 1 occurred”
Warning message:
```

```
"glm.fit: fitted probabilities numerically 0 or 1 occurred"
Warning message:
"glm.fit: fitted probabilities numerically 0 or 1 occurred"
                                 Df Deviance   AIC
- days_in_waiting_list            1    61787 61875
<none>                                 61787 61877
- stays_in_weekend_nights         1    61791 61879
- babies                          1    61793 61881
- stays_in_week_nights            1    61816 61904
- hotel                           1    61819 61907
- children                        1    61824 61912
- adults                          1    61825 61913
- reserved_room_type              9    61846 61918
- is_repeated_guest               1    61832 61920
- distribution_channel            4    61877 61959
- meal                            4    61904 61986
- customer_type                   3    62012 62096
- adr                             1    62098 62186
- booking_changes                 1    62301 62389
- previous_bookings_not_canceled  1    62655 62743
- lead_time                       1    62867 62955
- market_segment                  7    63310 63386
- total_of_special_requests       1    64175 64263
- previous_cancellations          1    64247 64335
- required_car_parking_spaces     1    64526 64614
- deposit_type                    2    69061 69147

Warning message:
"glm.fit: fitted probabilities numerically 0 or 1 occurred"
Step:  AIC=61874.92
is_canceled ~ hotel + lead_time + stays_in_weekend_nights + stays_in_week_nights
+
    adults + children + babies + meal + market_segment + distribution_channel +
    is_repeated_guest + previous_cancellations + previous_bookings_not_canceled
+
    reserved_room_type + booking_changes + deposit_type + customer_type +
    adr + required_car_parking_spaces + total_of_special_requests


Warning message:
"glm.fit: fitted probabilities numerically 0 or 1 occurred"
Warning message:
"glm.fit: algorithm did not converge"
Warning message:
"glm.fit: fitted probabilities numerically 0 or 1 occurred"
Warning message:
"glm.fit: fitted probabilities numerically 0 or 1 occurred"
Warning message:
"glm.fit: fitted probabilities numerically 0 or 1 occurred"
Warning message:
"glm.fit: fitted probabilities numerically 0 or 1 occurred"
Warning message:
"glm.fit: fitted probabilities numerically 0 or 1 occurred"
Warning message:
"glm.fit: fitted probabilities numerically 0 or 1 occurred"
Warning message:
"glm.fit: fitted probabilities numerically 0 or 1 occurred"
Warning message:
"glm.fit: fitted probabilities numerically 0 or 1 occurred"
Warning message:
"glm.fit: fitted probabilities numerically 0 or 1 occurred"
Warning message:
"glm.fit: fitted probabilities numerically 0 or 1 occurred"
Warning message:
"glm.fit: fitted probabilities numerically 0 or 1 occurred"
Warning message:
```

```
“glm.fit: fitted probabilities numerically 0 or 1 occurred”
Warning message:
“glm.fit: fitted probabilities numerically 0 or 1 occurred”
Warning message:
“glm.fit: fitted probabilities numerically 0 or 1 occurred”
Warning message:
“glm.fit: algorithm did not converge”
Warning message:
“glm.fit: fitted probabilities numerically 0 or 1 occurred”
Warning message:
“glm.fit: fitted probabilities numerically 0 or 1 occurred”
Warning message:
“glm.fit: fitted probabilities numerically 0 or 1 occurred”
Warning message:
“glm.fit: fitted probabilities numerically 0 or 1 occurred”
Warning message:
“glm.fit: fitted probabilities numerically 0 or 1 occurred”
                                Df Deviance   AIC
<none>                               61787 61875
- stays_in_weekend_nights         1    61792 61878
- babies                          1    61793 61879
- stays_in_week_nights            1    61817 61903
- hotel                           1    61820 61906
- children                        1    61824 61910
- adults                          1    61826 61912
- reserved_room_type              9    61846 61916
- is_repeated_guest               1    61832 61918
- distribution_channel            4    61878 61958
- meal                            4    61905 61985
- customer_type                   3    62013 62095
- adr                             1    62098 62184
- booking_changes                 1    62301 62387
- previous_bookings_not_canceled  1    62658 62744
- lead_time                       1    62874 62960
- market_segment                  7    63314 63388
- total_of_special_requests       1    64175 64261
- previous_cancellations          1    64258 64344
- required_car_parking_spaces     1    64527 64613
- deposit_type                    2    69065 69149
Call:
glm(formula = is_canceled ~ hotel + lead_time + stays_in_weekend_nights +
    stays_in_week_nights + adults + children + babies + meal +
    market_segment + distribution_channel + is_repeated_guest +
    previous_cancellations + previous_bookings_not_canceled +
    reserved_room_type + booking_changes + deposit_type + customer_type +
    adr + required_car_parking_spaces + total_of_special_requests,
    family = binomial, data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-8.4904  -0.7393  -0.4172   0.1745   6.0232

Coefficients:
                                 Estimate Std. Error z value Pr(>|z|)
(Intercept)                    -2.495e+00  2.200e-01 -11.337  < 2e-16 ***
hotelResort Hotel               1.456e-01  2.552e-02   5.708 1.14e-08 ***
lead_time                       3.761e-03  1.166e-04  32.238  < 2e-16 ***
stays_in_weekend_nights         2.590e-02  1.159e-02   2.235  0.02540 *
stays_in_week_nights            3.416e-02  6.307e-03   5.416 6.08e-08 ***
adults                          1.215e-01  2.163e-02   5.615 1.97e-08 ***
children                        1.989e-01  3.254e-02   6.114 9.72e-10 ***
babies                          3.261e-01  1.245e-01   2.619  0.00883 **
mealFB                          5.774e-01  1.263e-01   4.571 4.85e-06 ***
mealHB                         -2.340e-01  3.353e-02  -6.980 2.95e-12 ***
mealSC                          7.880e-02  3.524e-02   2.236  0.02533 *
```

```
mealUndefined                   -7.440e-01  1.192e-01  -6.242 4.31e-10 ***
market_segmentComplementary      5.059e-01  2.792e-01   1.812  0.06992 .
market_segmentCorporate         -1.191e-01  2.153e-01  -0.553  0.58022
market_segmentDirect             1.723e-01  2.387e-01   0.722  0.47048
market_segmentGroups             5.479e-02  2.251e-01   0.243  0.80772
market_segmentOffline TA/TO     -5.697e-01  2.256e-01  -2.525  0.01156 *
market_segmentOnline TA          6.720e-01  2.250e-01   2.986  0.00282 **
market_segmentUndefined         -1.353e+02  3.875e+07   0.000  1.00000
distribution_channelDirect      -6.567e-01  1.140e-01  -5.759 8.47e-09 ***
distribution_channelGDS         -1.467e+00  3.137e-01  -4.678 2.90e-06 ***
distribution_channelTA/TO       -1.956e-02  8.429e-02  -0.232  0.81653
distribution_channelUndefined    1.517e+02  3.875e+07   0.000  1.00000
is_repeated_guest               -6.605e-01  1.019e-01  -6.484 8.92e-11 ***
previous_cancellations           2.913e+00  7.533e-02  38.676  < 2e-16 ***
previous_bookings_not_canceled -5.661e-01  3.299e-02 -17.158  < 2e-16 ***
reserved_room_typeB              1.026e-01  1.016e-01   1.010  0.31240
reserved_room_typeC              5.356e-02  1.167e-01   0.459  0.64629
reserved_room_typeD             -6.535e-02  2.839e-02  -2.302  0.02134 *
reserved_room_typeE              6.129e-02  4.535e-02   1.351  0.17654
reserved_room_typeF             -4.774e-01  7.756e-02  -6.156 7.47e-10 ***
reserved_room_typeG             -9.935e-02  8.947e-02  -1.110  0.26685
reserved_room_typeH             -1.179e-01  1.470e-01  -0.802  0.42242
reserved_room_typeL             -6.237e-01  1.144e+00  -0.545  0.58561
reserved_room_typeP              1.457e+01  3.786e+02   0.038  0.96930
booking_changes                 -4.161e-01  2.046e-02 -20.340  < 2e-16 ***
deposit_typeNon Refund           5.461e+00  1.319e-01  41.413  < 2e-16 ***
deposit_typeRefundable          -1.134e-01  2.645e-01  -0.429  0.66823
customer_typeGroup               9.611e-02  1.987e-01   0.484  0.62855
customer_typeTransient           7.583e-01  6.493e-02  11.678  < 2e-16 ***
customer_typeTransient-Party     4.262e-01  6.927e-02   6.153 7.61e-10 ***
adr                              4.714e-03  2.668e-04  17.666  < 2e-16 ***
required_car_parking_spaces     -3.187e+02  9.718e+05   0.000  0.99974
total_of_special_requests       -6.886e-01  1.500e-02 -45.910  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 97132  on 73195  degrees of freedom
Residual deviance: 61787  on 73152  degrees of freedom
AIC: 61875

Number of Fisher Scoring iterations: 12
Call:
glm(formula = is_canceled ~ hotel + lead_time + stays_in_weekend_nights +
    stays_in_week_nights + adults + children + babies + meal +
    market_segment + distribution_channel + is_repeated_guest +
    previous_cancellations + previous_bookings_not_canceled +
    reserved_room_type + booking_changes + deposit_type + customer_type +
    adr + required_car_parking_spaces + total_of_special_requests,
    family = binomial, data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-8.4904  -0.7393  -0.4172   0.1745   6.0232

Coefficients:
                                 Estimate Std. Error z value Pr(>|z|)
(Intercept)                     -2.495e+00  2.200e-01 -11.337  < 2e-16 ***
hotelResort Hotel                1.456e-01  2.552e-02   5.708 1.14e-08 ***
lead_time                        3.761e-03  1.166e-04  32.238  < 2e-16 ***
stays_in_weekend_nights          2.590e-02  1.159e-02   2.235  0.02540 *
stays_in_week_nights             3.416e-02  6.307e-03   5.416 6.08e-08 ***
adults                           1.215e-01  2.163e-02   5.615 1.97e-08 ***
children                         1.989e-01  3.254e-02   6.114 9.72e-10 ***
```

```
babies                            3.261e-01  1.245e-01   2.619  0.00883 **
mealFB                            5.774e-01  1.263e-01   4.571 4.85e-06 ***
mealHB                           -2.340e-01  3.353e-02  -6.980 2.95e-12 ***
mealSC                            7.880e-02  3.524e-02   2.236  0.02533 *
mealUndefined                    -7.440e-01  1.192e-01  -6.242 4.31e-10 ***
market_segmentComplementary       5.059e-01  2.792e-01   1.812  0.06992 .
market_segmentCorporate          -1.191e-01  2.153e-01  -0.553  0.58022
market_segmentDirect              1.723e-01  2.387e-01   0.722  0.47048
market_segmentGroups              5.479e-02  2.251e-01   0.243  0.80772
market_segmentOffline TA/TO      -5.697e-01  2.256e-01  -2.525  0.01156 *
market_segmentOnline TA           6.720e-01  2.250e-01   2.986  0.00282 **
market_segmentUndefined          -1.353e+02  3.875e+07   0.000  1.00000
distribution_channelDirect       -6.567e-01  1.140e-01  -5.759 8.47e-09 ***
distribution_channelGDS          -1.467e+00  3.137e-01  -4.678 2.90e-06 ***
distribution_channelTA/TO        -1.956e-02  8.429e-02  -0.232  0.81653
distribution_channelUndefined     1.517e+02  3.875e+07   0.000  1.00000
is_repeated_guest                -6.605e-01  1.019e-01  -6.484 8.92e-11 ***
previous_cancellations            2.913e+00  7.533e-02  38.676  < 2e-16 ***
previous_bookings_not_canceled   -5.661e-01  3.299e-02 -17.158  < 2e-16 ***
reserved_room_typeB               1.026e-01  1.016e-01   1.010  0.31240
reserved_room_typeC               5.356e-02  1.167e-01   0.459  0.64629
reserved_room_typeD              -6.535e-02  2.839e-02  -2.302  0.02134 *
reserved_room_typeE               6.129e-02  4.535e-02   1.351  0.17654
reserved_room_typeF              -4.774e-01  7.756e-02  -6.156 7.47e-10 ***
reserved_room_typeG              -9.935e-02  8.947e-02  -1.110  0.26685
reserved_room_typeH              -1.179e-01  1.470e-01  -0.802  0.42242
reserved_room_typeL              -6.237e-01  1.144e+00  -0.545  0.58561
reserved_room_typeP               1.457e+01  3.786e+02   0.038  0.96930
booking_changes                  -4.161e-01  2.046e-02 -20.340  < 2e-16 ***
deposit_typeNon Refund            5.461e+00  1.319e-01  41.413  < 2e-16 ***
deposit_typeRefundable           -1.134e-01  2.645e-01  -0.429  0.66823
customer_typeGroup                9.611e-02  1.987e-01   0.484  0.62855
customer_typeTransient            7.583e-01  6.493e-02  11.678  < 2e-16 ***
customer_typeTransient-Party      4.262e-01  6.927e-02   6.153 7.61e-10 ***
adr                               4.714e-03  2.668e-04  17.666  < 2e-16 ***
required_car_parking_spaces      -3.187e+02  9.718e+05   0.000  0.99974
total_of_special_requests        -6.886e-01  1.500e-02 -45.910  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 97132  on 73195  degrees of freedom
Residual deviance: 61787  on 73152  degrees of freedom
AIC: 61875

Number of Fisher Scoring iterations: 12
```

In [10]:

```
preds <-ifelse(predict(model, test)>0.5, 1, 0)
```

In [11]:

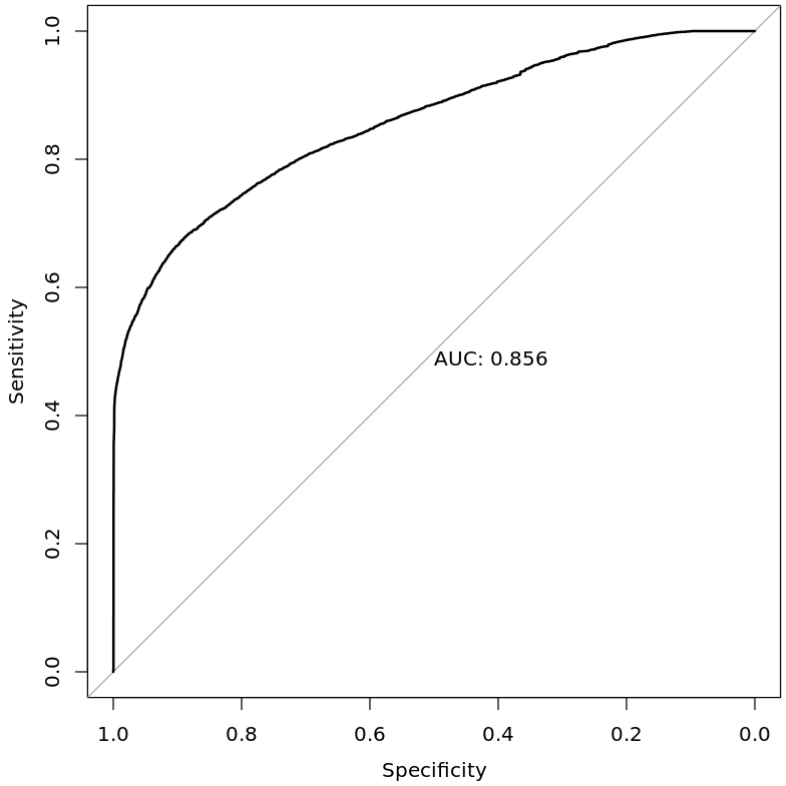
```
# accuray
mean(preds==test$is_canceled)
```

0.796040677101597

In [12]:

```
# ROC
library(pROC)
test_prob = predict(model, newdata = test, type = "response")
test_roc = roc(test$is_canceled ~ test_prob, plot = TRUE, print.auc = TRUE)
```

```
Type 'citation("pROC")' for a citation.

Attaching package: ‘pROC’

The following objects are masked from ‘package:stats’:

    cov, smooth, var

Setting levels: control = 0, case = 1

Setting direction: controls < cases
```



In [ ]:

Decision Tree, Boosting Tree, & Random Forest

Decision Tree

In [25]:

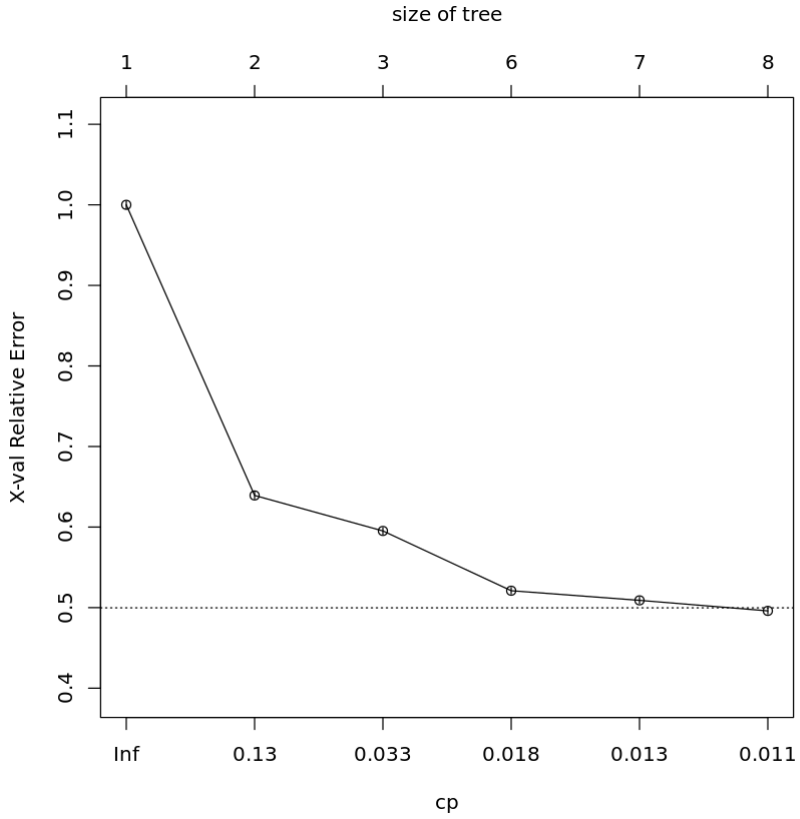
```
# Spliting the data into 70% training and 30% test set
library(MASS)
library(caret)
training.samples <- dat.clean$is_canceled %>%
  createDataPartition(p = 0.7, list = FALSE)
train  <- dat.clean[training.samples, ]
test <- dat.clean[-training.samples, ]
```

```
y.train <- train$is_canceled
y.test <- train$is_canceled
```

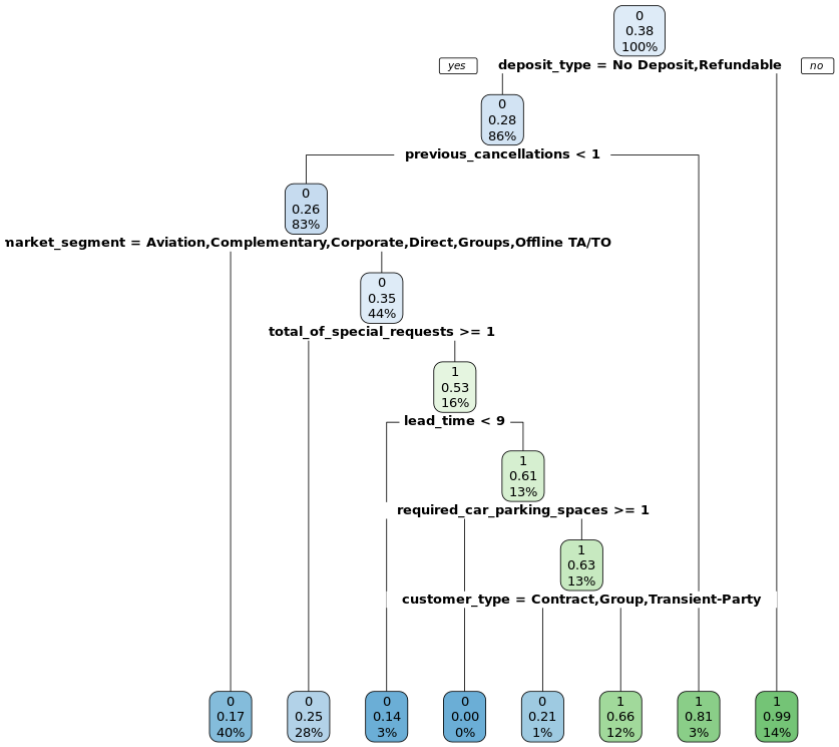
In [6]:
```
library(rpart)
library(rpart.plot)
```

In [7]:
```
# Fit the model
mod <- rpart(is_canceled~., data=train, method="class")
```

In [8]:
```
plotcp(mod)
```



In [9]:
```
rpart.plot(mod)
```



In [10]:
```
preds <- predict(mod, test, type="class")
```

In [11]:
```
table_mat  <- table(preds, test$is_canceled)
table_mat
```

```
preds     0     1
    0 17995  4484
    1  1466  7424
```

In [12]:
```
accuracy_Test <- sum(diag(table_mat)) / sum(table_mat)
accuracy_Test
```

0.810322292709363

In [13]:
```
print(paste('Accuracy for test', accuracy_Test))
```

```
[1] "Accuracy for test 0.810322292709363"
```

In [14]:
```
library(pROC)
```

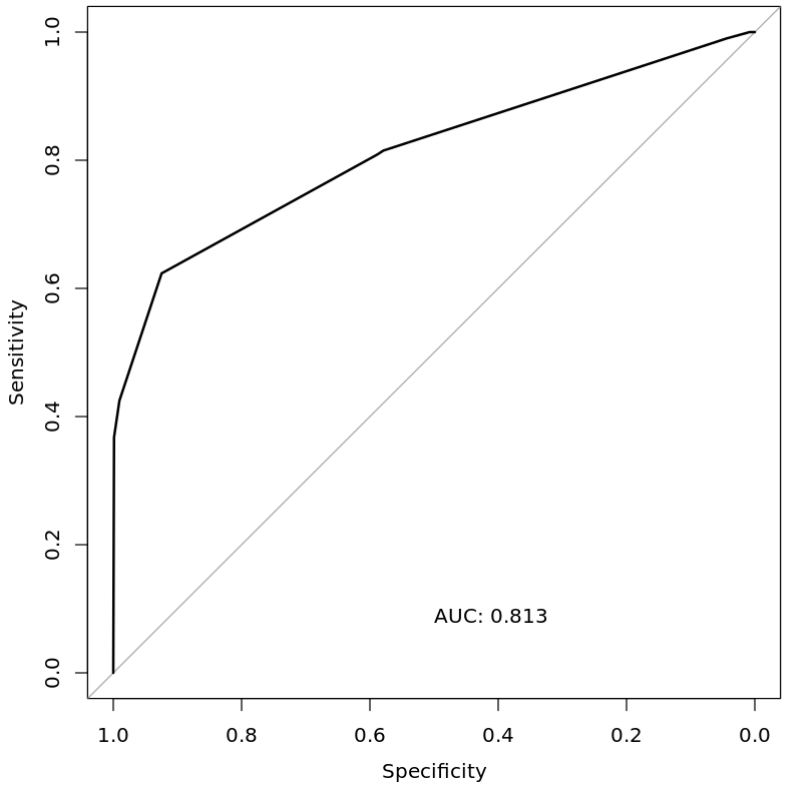
```
Type 'citation("pROC")' for a citation.

Attaching package: ‘pROC’

The following objects are masked from ‘package:stats’:
```

```
    cov, smooth, var
```

In [15]:

```
dc_pred_prob<-predict(mod,test,type="prob")[,2]
dc_roc<-roc(test$is_canceled,dc_pred_prob,auc=TRUE)
plot(dc_roc,print.auc=TRUE,print.auc.y=.1)
```

```
Setting levels: control = 0, case = 1

Setting direction: controls < cases
```



In [ ]:

```
#Construct function to return accuracy
#Tune the maximum depth
#Tune the minimum number of sample a node must have before it can split
#Tune the minimum number of sample a leaf node must have
```

In [16]:

```
accuracy_tune <- function(fit) {
  preds <- predict(fit, test, type = 'class')
  table_mat <- table(test$is_canceled, preds)
  accuracy_Test <- sum(diag(table_mat)) / sum(table_mat)
  accuracy_Test
}

control <- rpart.control(minsplit = 4,
                         minbucket = round(5 / 3),
                         maxdepth = 3,
                         cp = 0)
tune_fit <- rpart(is_canceled~., data = train, method = 'class', control = contr
accuracy_tune(tune_fit)
```

0.779495680448851

The classification arruracy for decision tree is 0.779.

Boosting Tree

In [26]:
```
library(gbm)
```

Fitting the boosting tree model. In case that we are predicting the binary result, we use distribution as "bernoulli". Moreover, we select to use 1000 iteration trees, 2 depth of tree interaction, and shrinkage of 0.01.

In [27]:
```
fit.btree <- gbm(is_canceled~.,
                 data = train,
                 distribution = "bernoulli",
                 n.trees = 1000,
                 interaction.depth = 2,
                 cv.folds = 2,
                 shrinkage = 0.01)
```

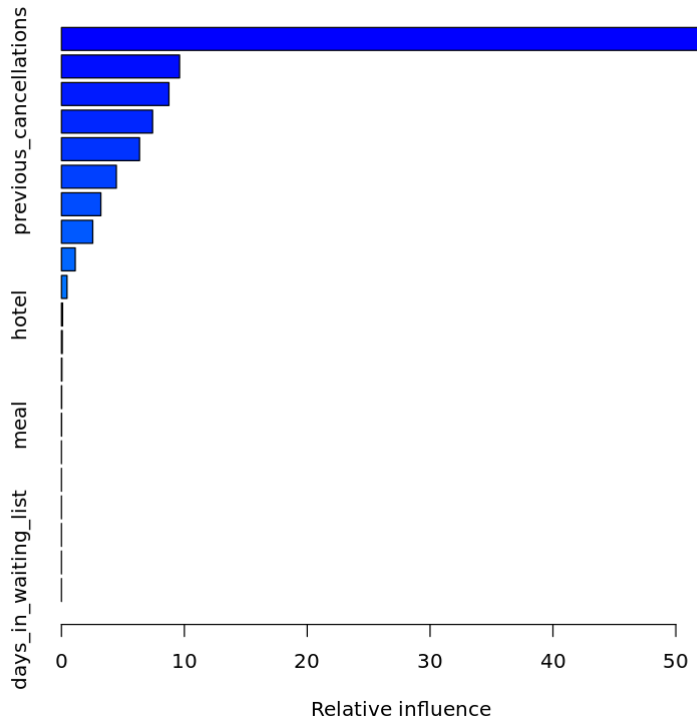
In the summary, "Deposit_type" has the largest influences in our model function. The rest factors that also remarcable are "lead_time", "market_segment", "previous_cancellations", and "total_of_special_requests"

In [8]:
```
summary(fit.btree)
```

A data.frame: 21 × 2

	var	rel.inf
	<fct>	<dbl>
deposit_type	deposit_type	55.83367666
lead_time	lead_time	9.61372202
market_segment	market_segment	8.74687385
previous_cancellations	previous_cancellations	7.41858626
total_of_special_requests	total_of_special_requests	6.35660001
required_car_parking_spaces	required_car_parking_spaces	4.46861011
booking_changes	booking_changes	3.20584056
customer_type	customer_type	2.54648510
adr	adr	1.12379761
previous_bookings_not_canceled	previous_bookings_not_canceled	0.45428106
hotel	hotel	0.09181246
reserved_room_type	reserved_room_type	0.06793349
stays_in_week_nights	stays_in_week_nights	0.03728539
adults	adults	0.02012096

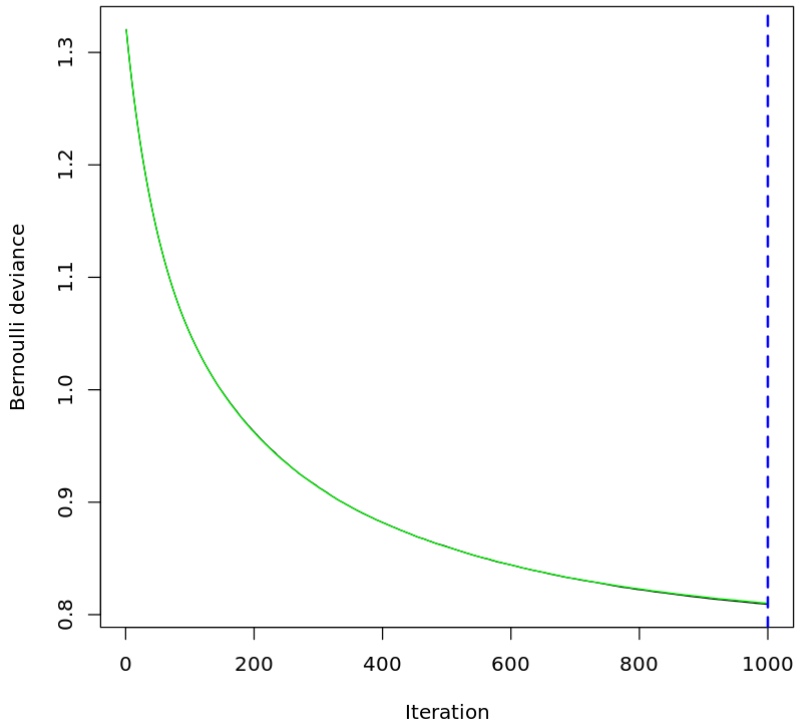
	var	rel.inf
	<fct>	<dbl>
meal	meal	0.01437445
stays_in_weekend_nights	stays_in_weekend_nights	0.00000000
children	children	0.00000000
babies	babies	0.00000000
distribution_channel	distribution_channel	0.00000000
is_repeated_guest	is_repeated_guest	0.00000000
days_in_waiting_list	days_in_waiting_list	0.00000000



We have find out the best iteration of the trees is 1000. It probally can do better with more iterations, but it will cause the model run too slow.

In [29]:
```
best.iter <- gbm.perf(fit.btree,method="cv")
print(best.iter)
```

```
[1] 1000
```



Predict the model

We use the boosting model we create, test dataset, and best number of iteration to predict the model. For the result of the boosting model, we can observe that our model perfectly preticted 25240 out of 31369, giving the accuracy of 80.46%. There were 6129 cases that our model got wrong. These 6129 cases were divided between False negatives and false positives as 335 and 5794 respectively.

In [11]:
```
canceled.predict <- predict(fit.btree,test,n.trees = best.iter)
```

In [13]:
```
prediction_binary <- as.factor(ifelse(canceled.predict>0.5, 1,0))
test$is_canceled <- as.factor(test$is_canceled)
confusionMatrix(prediction_binary, test$is_canceled)
```

```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 19301  5794
         1   335  5939

               Accuracy : 0.8046
                 95% CI : (0.8002, 0.809)
    No Information Rate : 0.626
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5396

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

            Sensitivity : 0.9829
```

```
            Specificity : 0.5062
         Pos Pred Value : 0.7691
         Neg Pred Value : 0.9466
             Prevalence : 0.6260
         Detection Rate : 0.6153
   Detection Prevalence : 0.8000
      Balanced Accuracy : 0.7446

       'Positive' Class : 0
```

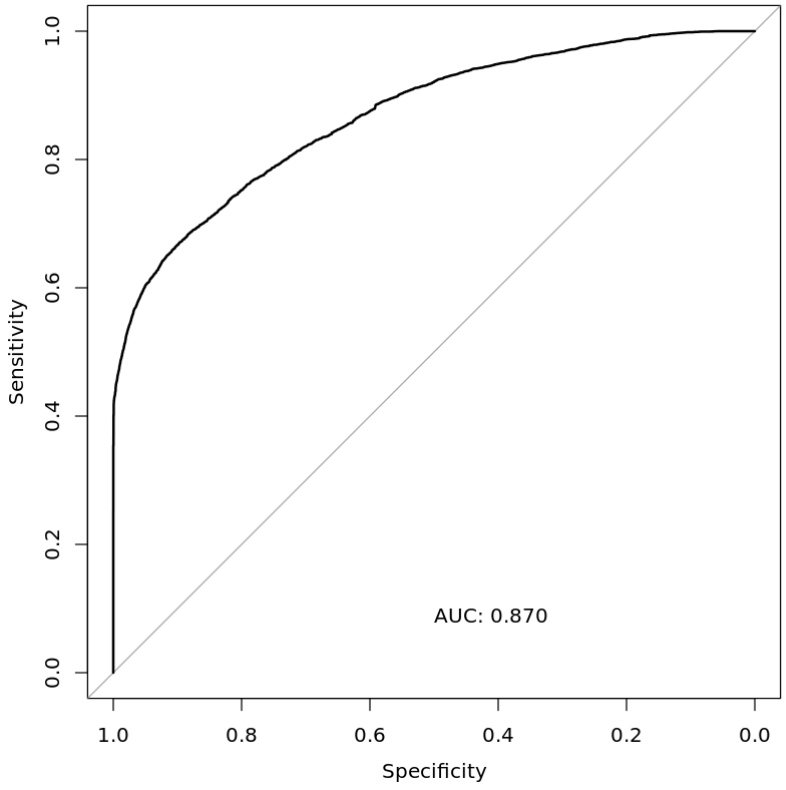
Plot the ROC curve to check the performance of final model

As evident, our model has an AUC of 0.870.

In [14]:

```
par(mfrow = c(1, 1))

library(stats)
library(pROC)
canceled.roc = roc(test$is_canceled,canceled.predict, auc=TRUE)
plot(canceled.roc, print.auc=TRUE,print.auc.y=.1)
```

```
Setting levels: control = 0, case = 1

Setting direction: controls < cases
```



In [ ]:

Random Forest

We drop the company and agent columns since there are too many NAs, and backward fill the NAs in children column.

Columns like reservation date and country have been removed as they have many unique values and cannot be used for building Decision Trees.

Feature Engineering

Adding a new column to denote number of nights stayed in total and the total cost which is calculated as:

In [14]:
```
dat_eu <- dat_eu %>%
  mutate(stay_nights_total = stays_in_weekend_nights + stays_in_week_nights)

head(dat_eu)
```

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number
	<fct>	<int>	<int>	<int>	<fct>	<int>
18	Resort Hotel	0	0	2015	July	27
19	Resort Hotel	0	7	2015	July	27
20	Resort Hotel	0	37	2015	July	27
21	Resort Hotel	0	72	2015	July	27
22	Resort Hotel	0	72	2015	July	27
23	Resort Hotel	0	72	2015	July	27

We drop the company and agent columns since there are too many NAs, the backward filled values are not reilable for analyzing the model result.

Columns like reservation date,country has been removed as it has many unique values and cannot be used for building Decision Trees.

Columns arrival_date_week_number, stays_in_weekend_nights, stays_in_week_nights have been removed as they are redundant once we have added the columns stay_nights_total, stay_cost_total.

Column market_segment is similar as distribution_channel, so we also drop it.

In [15]:
```
dat_rf <- dat_eu[c('hotel','is_canceled','lead_time','adults','children','babies
                                'distribution_channel','is_repeated_guest', 'adr
                                'previous_cancellations','previous_bookings_not_
                                'deposit_type','days_in_waiting_list','customer_
                                'arrival_date_month', 'arrival_date_day_of_month
                                'required_car_parking_spaces','stay_nights_total

dat_rf$is_canceled <- as.factor(dat_rf$is_canceled)
```

```
dat_rf$is_repeated_guest <- as.factor(dat_rf$is_repeated_guest)
str(dat_rf)
```

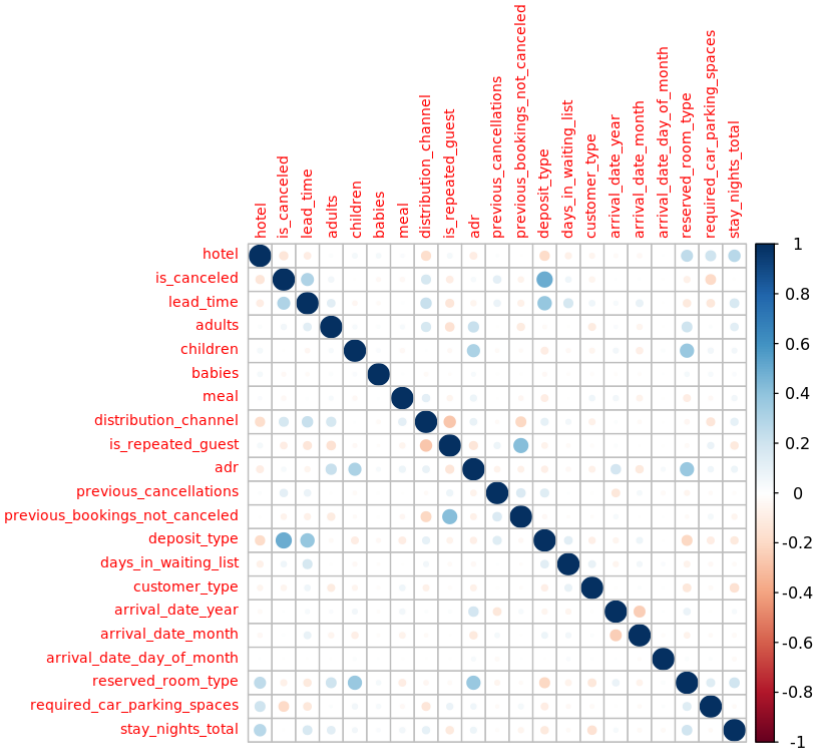
```
'data.frame':   104548 obs. of  21 variables:
 $ hotel                         : Factor w/ 2 levels "City Hotel","Resort Hote
l": 2 2 2 2 2 2 2 2 2 2 ...
 $ is_canceled                   : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1
2 ...
 $ lead_time                     : int  0 7 37 72 72 72 127 78 48 60 ...
 $ adults                        : int  2 2 1 2 2 2 2 2 2 2 ...
 $ children                      : int  0 0 0 0 0 0 0 0 0 0 ...
 $ babies                        : int  0 0 0 0 0 0 0 0 0 0 ...
 $ meal                          : Factor w/ 5 levels "BB","FB","HB",..: 1 1 1 1
1 1 3 1 1 1 ...
 $ distribution_channel          : Factor w/ 5 levels "Corporate","Direct",..: 1
2 4 2 2 2 4 4 4 4 ...
 $ is_repeated_guest             : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1
1 ...
 $ adr                           : num  107.4 153 97.3 84.7 84.7 ...
 $ previous_cancellations        : int  0 0 0 0 0 0 0 0 0 0 ...
 $ previous_bookings_not_canceled: int  0 0 0 0 0 0 0 0 0 0 ...
 $ deposit_type                  : Factor w/ 3 levels "No Deposit","Non Refun
d",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ days_in_waiting_list          : int  0 0 0 0 0 0 0 0 0 0 ...
 $ customer_type                 : Factor w/ 4 levels "Contract","Group",..: 3 3
3 3 3 3 1 3 1 3 ...
 $ arrival_date_year             : int  2015 2015 2015 2015 2015 2015 2015 2015
2015 2015 ...
 $ arrival_date_month            : Factor w/ 12 levels "April","August",..: 6 6
6 6 6 6 6 6 6 6 ...
 $ arrival_date_day_of_month     : int  1 1 1 1 1 1 1 1 1 1 ...
 $ reserved_room_type            : Factor w/ 10 levels "A","B","C","D",..: 1 7 6
1 1 4 4 4 4 5 ...
 $ required_car_parking_spaces   : int  0 0 0 0 0 0 0 1 0 0 ...
 $ stay_nights_total             : int  1 4 5 6 6 6 7 7 7 7 ...
```

In [16]:

```
# convert all factors to numeric
df <- dat_rf %>% mutate_if(is.factor, as.numeric)
```

In [17]:

```
# check multicollinearity
corrplot(cor(df),tl.cex = 0.7)
```



There is no obvious multicollinearity among the independent variables.

In [18]:
```
# The dataset is split into train and test set in 70:30 ratio with the is_cancel
library(caTools)
set.seed(100)
split = sample.split(dat_rf$is_canceled, SplitRatio = 0.7)
train_rf = subset(dat_rf, split == TRUE)
test_rf = subset(dat_rf, split == FALSE)
dim(train_rf)
dim(test_rf)
```

73183 · 21

31365 · 21

In [57]:
```
# Fit a random forest model with mtry=2 on the training set
set.seed(100)
rf_model1<-randomForest(is_canceled~.,
                        data=train_rf,
                        ntree=500,
                        cutoff=c(0.5,0.5),
                        mtry=2,
                        importance=TRUE)
rf_model1
```

```
Call:
 randomForest(formula = is_canceled ~ ., data = train_rf, ntree = 500,      cuto
ff = c(0.5, 0.5), mtry = 2, importance = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2
```

```
        OOB estimate of  error rate: 20.95%
Confusion matrix:
      0     1 class.error
0 45136   263 0.005793079
1 15068 12716 0.542326519
```

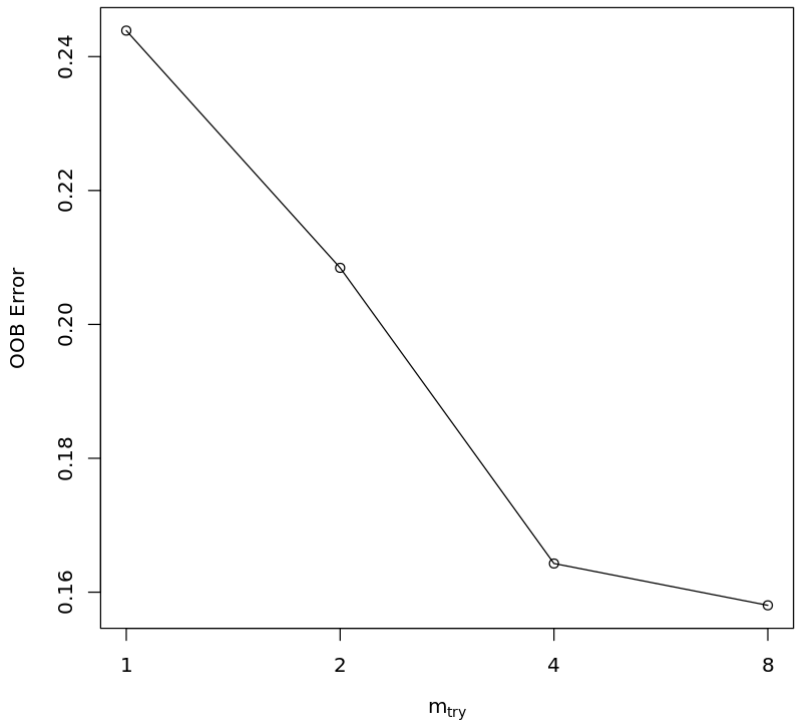
OOB error is the mean prediction error on each training variable.

The above model gave an error rate of 20.95% on OOB dataset.

We will try to find the best hyper parameter value for mtry after tuning.

In [58]:
```
# Tuning parameter
set.seed(100)
rf_model2 <- tuneRF(x = train_rf%>%select(-is_canceled),
                    y = train_rf$is_canceled,mtryStart=2,
                    ntreeTry = 500)
```

```
mtry = 2  OOB error = 20.85%
Searching left ...
mtry = 1        OOB error = 24.39%
-0.1699771 0.05
Searching right ...
mtry = 4        OOB error = 16.43%
0.211865 0.05
mtry = 8        OOB error = 15.8%
0.03801048 0.05
```



The model is showing least Out of Bound(OOB) error for mtry=8. Now train the model based on the new mtry value.

In [19]:
```
# Fit a new model with mtry=8
rf_model3<-randomForest(is_canceled~.,
                        data=train_rf,
```

```
                         ntree=500,
                         cutoff=c(0.5,0.5),
                         mtry=8,
                         importance=TRUE)
rf_model3
```

```
Call:
 randomForest(formula = is_canceled ~ ., data = train_rf, ntree = 500,      cuto
ff = c(0.5, 0.5), mtry = 8, importance = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 8

        OOB estimate of  error rate: 15.87%
Confusion matrix:
      0     1 class.error
0 41536  3863  0.08508998
1  7749 20035  0.27890153
```

In [60]:

```
# Prediction on the test set
rf_pred_class<-predict(rf_model3,test_rf,type="class")
confusionMatrix(as.factor(rf_pred_class),test_rf$is_canceled)
```

```
Installing package into '/home/jupyter/.R/library'
(as 'lib' is unspecified)

Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 17794  3261
         1  1663  8647

               Accuracy : 0.843
                 95% CI : (0.8389, 0.847)
    No Information Rate : 0.6203
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.6578

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

            Sensitivity : 0.9145
            Specificity : 0.7262
         Pos Pred Value : 0.8451
         Neg Pred Value : 0.8387
             Prevalence : 0.6203
         Detection Rate : 0.5673
   Detection Prevalence : 0.6713
      Balanced Accuracy : 0.8203

       'Positive' Class : 0
```

After Evaluating the probabilities and the class, the best random forest model gave an accuracy of 0.843.

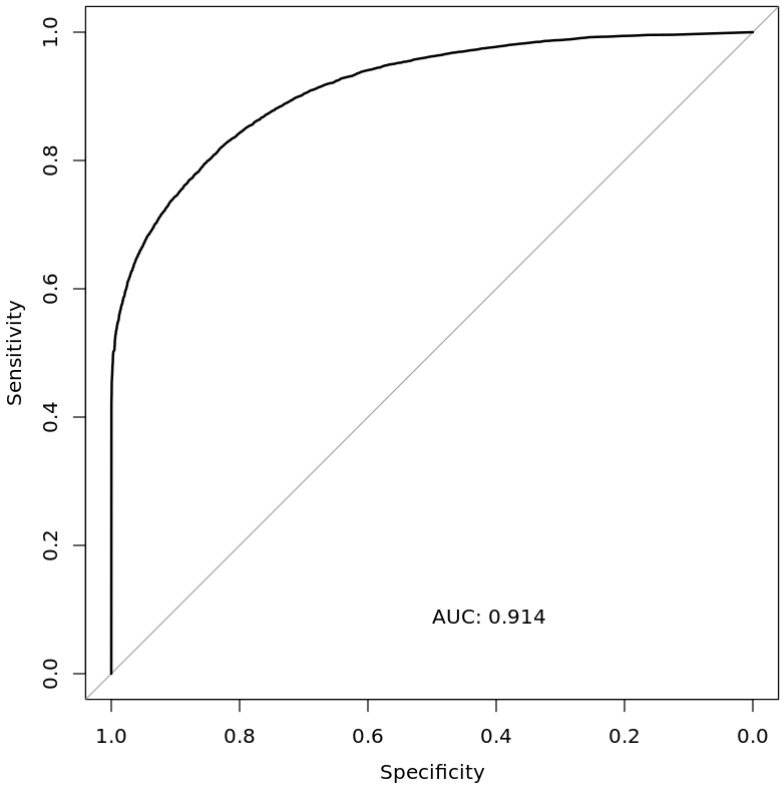
Then we plot the ROC curve. The AUC is also good.

In [61]:

```
# Plot the ROC curve to check the performance of final model
rf_pred_prob<-predict(rf_model3,test_rf,type="prob")[,2]
```

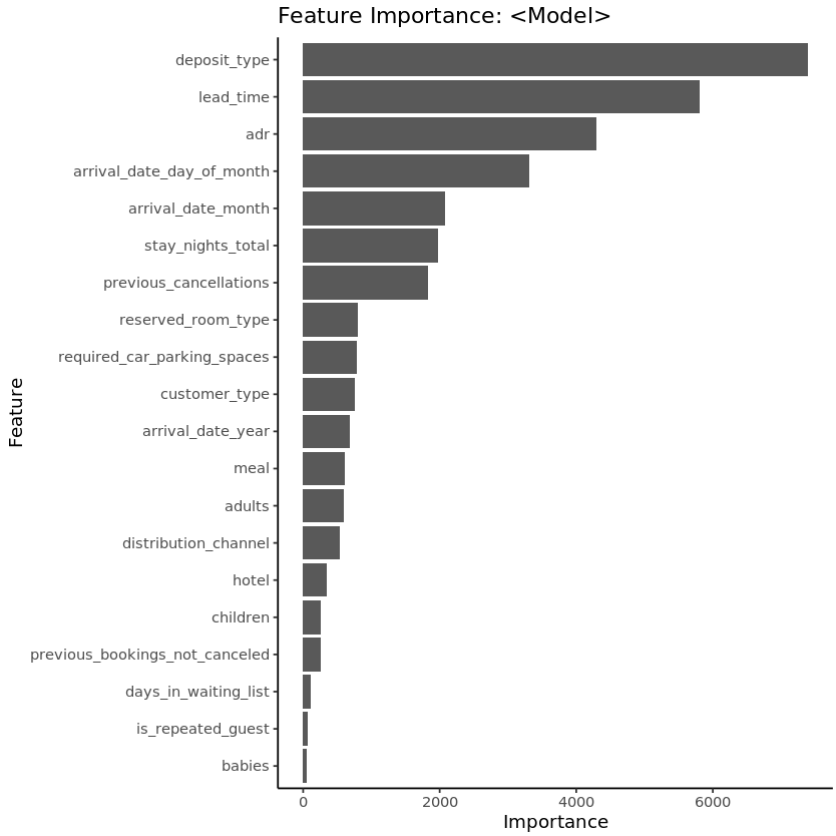
```
rf_roc<-roc(test_rf$is_canceled,rf_pred_prob,auc=TRUE)
plot(rf_roc,print.auc=TRUE,print.auc.y=.1)
```

```
Setting levels: control = 0, case = 1

Setting direction: controls < cases
```



In [26]:

```
#check variables that are most predictive using variable importance plot
# make dataframe from importance() output
  feat_imp_df <- importance(rf_model3) %>%
    data.frame() %>%
    mutate(feature = row.names(.))

  # plot dataframe
  ggplot(feat_imp_df, aes(x = reorder(feature, MeanDecreaseGini),
                          y = MeanDecreaseGini)) +
    geom_bar(stat='identity') +
    coord_flip() +
    theme_classic() +
    labs(
      x     = "Feature",
      y     = "Importance",
      title = "Feature Importance: <Model>"
    )
```



From the plot of feature importance we notice that the deposit_type (No Deposit, Non Refund and Refundable), lead_time (Number of days that elapsed between the entering date of the booking and the arrival date), adr (Average Daily Rate) and arrival_date (Day of arrival date) are the most important features in the model. They are also the features that affected the cancelation the most.

Conclusion

- From the result of several prediction models, we notice that the deposit_type (if the customer made a deposit to guarantee the booking), lead_time (Number of days between the booking and the arrival date), adr (Average Daily Rate),Previous_cancellations (number of previous bookings that were cancelled by the customer) and Total number of special request and Market segment are the most important features which contributed a lot to the predictions.
- In the future, the hotels manager could pay more attention to these kinds of factors to get more information for why customers cancel and make some changes to avoid high cancellation rate.