DW Final PROj

Tianhe Wang

4/26/2021

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
library(tidyverse)
```

```
## Warning: package ’tidyverse’ was built under R version 4.0.3

## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.0     v dplyr   1.0.5
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

## Warning: package ’ggplot2’ was built under R version 4.0.4

## Warning: package ’tibble’ was built under R version 4.0.4

## Warning: package ’tidyr’ was built under R version 4.0.4

## Warning: package ’readr’ was built under R version 4.0.3

## Warning: package ’dplyr’ was built under R version 4.0.4

## Warning: package ’forcats’ was built under R version 4.0.4

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
## Warning: package ’lubridate’ was built under R version 4.0.4
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(scales)
```

```
##
## Attaching package: 'scales'

## The following object is masked from 'package:purrr':
##
##     discard

## The following object is masked from 'package:readr':
##
##     col_factor
```

```
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(zoo)
```

```
##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(cowplot)
```

```
## Warning: package 'cowplot' was built under R version 4.0.5

##
## Attaching package: 'cowplot'

## The following object is masked from 'package:lubridate':
##
##     stamp
```

```
library(countrycode)
```

```
## Warning: package 'countrycode' was built under R version 4.0.5
```

```
library(choroplethrMaps)
```

```
## Warning: package 'choroplethrMaps' was built under R version 4.0.5
```

```
library(choroplethr)
```

```
## Warning: package 'choroplethr' was built under R version 4.0.5

## Loading required package: acs

## Warning: package 'acs' was built under R version 4.0.5

## Loading required package: XML

## Warning: package 'XML' was built under R version 4.0.4

##
## Attaching package: 'acs'

## The following object is masked from 'package:dplyr':
##
##     combine

## The following object is masked from 'package:base':
##
##     apply
```

Background

As a tour guide, my father is responsible for designing, scheduling trips for customers and taking care of them well. Thanks to his job, I have also got many chances to visit a lot of places since I remember things. In my memories, hotel is always one of the most interesting part of the trip. No matter if it is a hotel, hostel or a guest hotel, all of them gave me unique experience and memories. Therefore, I decided to collect some data about hotel for this project.

Data Collection

After deciding the big direction, I went to Kaggle and UCI Machine Learning repository to find a good dataset. There are actually many interesting datasets like Airbnb's open data. Also some datasets including hotels' review which allow people to do the language sentiments analysis. However, I chose the Hotel Booking Demand dataset in the end because there are 32 columns and 11930 rows which is a perfect dataset for wrangling purpose in my mind. The dataset includes many useful information so that I can do whatever analysis I want. After downloading the csv. file from Kaggle (https://www.kaggle.com/jessemostipak/hotel-booking-demand), I start thinking of the ultimate goal of this project. Then I decided to focus on cancels of hotel bookings and try to predict if the customer cancel or not order based on the information. I think this is valuable because, given a customer's information, hotels can better anticipate if the customer would like to stay or not. Also, hotels can improve their services based on the analysis to stay more customers and better segment customer groups for advertising and more business purposes. Basically, these are reasons why I do this project.

here is a brief look at data

```
summary(original_data)
```

```
##     hotel            is_canceled       lead_time   arrival_date_year
##  Length:119390      Min.   :0.0000   Min.   :  0   Min.   :2015     
##  Class :character   1st Qu.:0.0000   1st Qu.: 18   1st Qu.:2016     
##  Mode  :character   Median :0.0000   Median : 69   Median :2016     
##                     Mean   :0.3704   Mean   :104   Mean   :2016     
##                     3rd Qu.:1.0000   3rd Qu.:160   3rd Qu.:2017     
##                     Max.   :1.0000   Max.   :737   Max.   :2017     
##                                                                     
##  arrival_date_month arrival_date_week_number arrival_date_day_of_month
##  Length:119390      Min.   : 1.00            Min.   : 1.0             
##  Class :character   1st Qu.:16.00            1st Qu.: 8.0             
##  Mode  :character   Median :28.00            Median :16.0             
##                     Mean   :27.17            Mean   :15.8             
##                     3rd Qu.:38.00            3rd Qu.:23.0             
##                     Max.   :53.00            Max.   :31.0             
##                                                                       
##  stays_in_weekend_nights stays_in_week_nights     adults      
##  Min.   : 0.0000         Min.   : 0.0         Min.   : 0.000  
##  1st Qu.: 0.0000         1st Qu.: 1.0         1st Qu.: 2.000  
##  Median : 1.0000         Median : 2.0         Median : 2.000  
##  Mean   : 0.9276         Mean   : 2.5         Mean   : 1.856  
##  3rd Qu.: 2.0000         3rd Qu.: 3.0         3rd Qu.: 2.000  
##  Max.   :19.0000         Max.   :50.0         Max.   :55.000  
##                                                               
##     children           babies              meal             country         
##  Min.   : 0.0000   Min.   : 0.000000   Length:119390      Length:119390     
##  1st Qu.: 0.0000   1st Qu.: 0.000000   Class :character   Class :character  
##  Median : 0.0000   Median : 0.000000   Mode  :character   Mode  :character  
##  Mean   : 0.1039   Mean   : 0.007949                                        
##  3rd Qu.: 0.0000   3rd Qu.: 0.000000                                        
##  Max.   :10.0000   Max.   :10.000000                                        
##  NA’s   :4                                                                  
##  market_segment     distribution_channel is_repeated_guest
##  Length:119390      Length:119390        Min.   :0.00000  
##  Class :character   Class :character     1st Qu.:0.00000  
##  Mode  :character   Mode  :character     Median :0.00000  
##                                          Mean   :0.03191  
##                                          3rd Qu.:0.00000  
##                                          Max.   :1.00000  
##                                                           
##  previous_cancellations previous_bookings_not_canceled reserved_room_type
##  Min.   : 0.00000       Min.   : 0.0000                Length:119390     
##  1st Qu.: 0.00000       1st Qu.: 0.0000                Class :character  
##  Median : 0.00000       Median : 0.0000                Mode  :character  
##  Mean   : 0.08712       Mean   : 0.1371                                  
##  3rd Qu.: 0.00000       3rd Qu.: 0.0000                                  
##  Max.   :26.00000       Max.   :72.0000                                  
##                                                                          
##  assigned_room_type booking_changes   deposit_type           agent          
##  Length:119390      Min.   : 0.0000   Length:119390      Length:119390
##  Class :character   1st Qu.: 0.0000   Class :character   Class :character
##  Mode  :character   Median : 0.0000   Mode  :character   Mode  :character
##                     Mean   : 0.2211
##                     3rd Qu.: 0.0000
##                     Max.   :21.0000
##
##    company          days_in_waiting_list customer_type           adr
##  Length:119390      Min.   :  0.000      Length:119390      Min.   :  -6.38
##  Class :character   1st Qu.:  0.000      Class :character   1st Qu.:  69.29
##  Mode  :character   Median :  0.000      Mode  :character   Median :  94.58
##                     Mean   :  2.321                         Mean   : 101.83
##                     3rd Qu.:  0.000                         3rd Qu.: 126.00
##                     Max.   :391.000                         Max.   :5400.00
##
##  required_car_parking_spaces total_of_special_requests reservation_status
##  Min.   :0.00000             Min.   :0.0000            Length:119390
##  1st Qu.:0.00000             1st Qu.:0.0000            Class :character
##  Median :0.00000             Median :0.0000            Mode  :character
##  Mean   :0.06252             Mean   :0.5714
##  3rd Qu.:0.00000             3rd Qu.:1.0000
##  Max.   :8.00000             Max.   :5.0000
##
##  reservation_status_date
##  Length:119390
##  Class :character
##  Mode  :character
##
##
##
##
```

```
head(original_data)
```

```
##          hotel is_canceled lead_time arrival_date_year arrival_date_month
## 1 Resort Hotel           0       342              2015               July
## 2 Resort Hotel           0       737              2015               July
## 3 Resort Hotel           0         7              2015               July
## 4 Resort Hotel           0        13              2015               July
## 5 Resort Hotel           0        14              2015               July
## 6 Resort Hotel           0        14              2015               July
##   arrival_date_week_number arrival_date_day_of_month stays_in_weekend_nights
## 1                       27                         1                       0
## 2                       27                         1                       0
## 3                       27                         1                       0
## 4                       27                         1                       0
## 5                       27                         1                       0
## 6                       27                         1                       0
##   stays_in_week_nights adults children babies meal country market_segment
## 1                    0      2        0      0   BB     PRT         Direct
## 2                    0      2        0      0   BB     PRT         Direct
## 3                    1      1        0      0   BB     GBR         Direct
## 4                    1      1        0      0   BB     GBR      Corporate
## 5                    2      2        0      0   BB     GBR      Online TA
## 6                    2      2        0      0   BB     GBR      Online TA
##   distribution_channel is_repeated_guest previous_cancellations
## 1               Direct                 0                      0
## 2               Direct                 0                      0
## 3               Direct                 0                      0
## 4            Corporate                 0                      0
## 5                TA/TO                 0                      0
## 6                TA/TO                 0                      0
##   previous_bookings_not_canceled reserved_room_type assigned_room_type
## 1                              0                  C                  C
## 2                              0                  C                  C
## 3                              0                  A                  C
## 4                              0                  A                  A
## 5                              0                  A                  A
## 6                              0                  A                  A
##   booking_changes deposit_type agent company days_in_waiting_list customer_type
## 1               3   No Deposit  NULL    NULL                    0     Transient
## 2               4   No Deposit  NULL    NULL                    0     Transient
## 3               0   No Deposit  NULL    NULL                    0     Transient
## 4               0   No Deposit   304    NULL                    0     Transient
## 5               0   No Deposit   240    NULL                    0     Transient
## 6               0   No Deposit   240    NULL                    0     Transient
##   adr required_car_parking_spaces total_of_special_requests reservation_status
## 1   0                           0                         0          Check-Out
## 2   0                           0                         0          Check-Out
## 3  75                           0                         0          Check-Out
## 4  75                           0                         0          Check-Out
## 5  98                           0                         1          Check-Out
## 6  98                           0                         1          Check-Out
##   reservation_status_date
## 1              2015-07-01
## 2              2015-07-01
## 3              2015-07-02
## 4              2015-07-02
## 5              2015-07-03
## 6              2015-07-03
```

```
dim(original_data)
```

```
## [1] 119390     32
```

Data Preprocessing

In this section, I first clean the NA values in the dataset. Because some NAs are strings, I have to change them manully instead of using na.omit function only. Besides, for the columns Agent and Company, the reason they have a lot of NAs is the way Data Collector tidying the data. For example, data collector uses different number to represent what agents each visitor use. For those who travel without using agents, their value in column Agent is NA. Therefore, we can not simply remove this kind of NAs from dataset. I then decided to set those values to 0.

data cleaning

```
#Clean NA values
cleaned_data = original_data
#See how many NAs in dataset
map(cleaned_data, ~sum(is.na(.)))
```

```
## $hotel
## [1] 0
##
## $is_canceled
## [1] 0
##
## $lead_time
## [1] 0
##
## $arrival_date_year
## [1] 0
##
## $arrival_date_month
## [1] 0
##
## $arrival_date_week_number
## [1] 0
##
## $arrival_date_day_of_month
## [1] 0
##
## $stays_in_weekend_nights
## [1] 0
##
## $stays_in_week_nights
## [1] 0
##
## $adults
## [1] 0
##
## $children
## [1] 4
##
## $babies
## [1] 0
##
## $meal
## [1] 0
##
## $country
## [1] 0
##
## $market_segment
## [1] 0
##
## $distribution_channel
## [1] 0
##
## $is_repeated_guest
## [1] 0
##
## $previous_cancellations
## [1] 0
##
## $previous_bookings_not_canceled
## [1] 0
##
## $reserved_room_type
## [1] 0
##
## $assigned_room_type
## [1] 0
##
## $booking_changes
## [1] 0
##
## $deposit_type
## [1] 0
##
## $agent
## [1] 0
##
## $company
## [1] 0
##
## $days_in_waiting_list
## [1] 0
##
## $customer_type
## [1] 0
##
## $adr
## [1] 0
##
## $required_car_parking_spaces
## [1] 0
##
## $total_of_special_requests
## [1] 0
##
## $reservation_status
## [1] 0
##
## $reservation_status_date
## [1] 0
```

```r
cleaned_data <- na.omit(cleaned_data)
#See how many Char NULL in dataset
map(cleaned_data, ~sum(.== "NULL"))
```

```
## $hotel
## [1] 0
##
## $is_canceled
## [1] 0
##
## $lead_time
## [1] 0
##
## $arrival_date_year
## [1] 0
##
## $arrival_date_month
## [1] 0
##
## $arrival_date_week_number
## [1] 0
##
## $arrival_date_day_of_month
## [1] 0
##
## $stays_in_weekend_nights
## [1] 0
##
## $stays_in_week_nights
## [1] 0
##
## $adults
## [1] 0
##
## $children
## [1] 0
##
## $babies
## [1] 0
##
## $meal
## [1] 0
##
## $country
## [1] 488
##
## $market_segment
## [1] 0
##
## $distribution_channel
## [1] 0
##
## $is_repeated_guest
## [1] 0
##
## $previous_cancellations
## [1] 0
##
```

```
## $previous_bookings_not_canceled
## [1] 0
##
## $reserved_room_type
## [1] 0
##
## $assigned_room_type
## [1] 0
##
## $booking_changes
## [1] 0
##
## $deposit_type
## [1] 0
##
## $agent
## [1] 16338
##
## $company
## [1] 112589
##
## $days_in_waiting_list
## [1] 0
##
## $customer_type
## [1] 0
##
## $adr
## [1] 0
##
## $required_car_parking_spaces
## [1] 0
##
## $total_of_special_requests
## [1] 0
##
## $reservation_status
## [1] 0
##
## $reservation_status_date
## [1] 0
```

```
cleaned_data <- cleaned_data[!cleaned_data$country=="NULL", ]
cleaned_data$company[cleaned_data$company == "NULL"] <- as.character(0)
cleaned_data$agent[cleaned_data$agent == "NULL"] <- as.character(0)
dim(cleaned_data)
```

```
## [1] 118898     32
```

```
cleaned_data <- as_tibble(cleaned_data) %>%
  mutate(is_canceled = as.factor(is_canceled))
```

```
#uncomment code below to get a copy of cleaned and tidy csv data.
#write.csv(cleaned_data, here("tidy_data.csv"), row.names = F)
```

Data Visualization

In this section, I start visually analyze data. Firstly, I plotted a Choropleth map here to give basic idea where those visitors come from and a barplot to show more clearly. As you can see underneath, most visitors come from Europe, United States and China. Surprisingly, almost half comes from Portugal and this is what you can see only from the barplot.

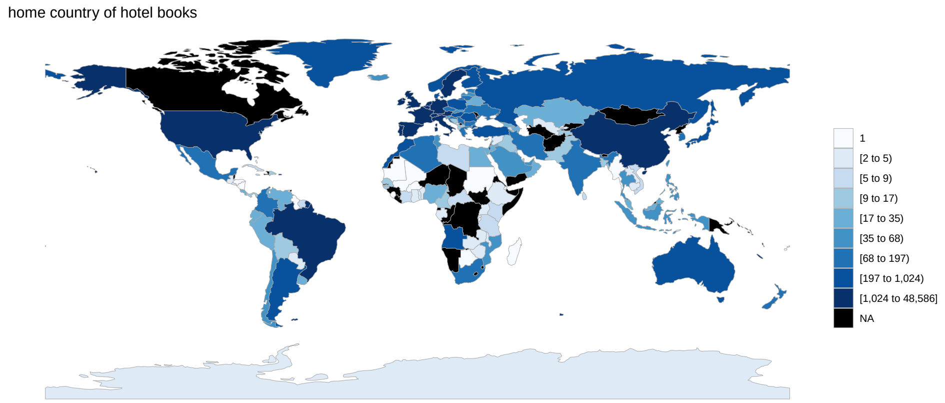
visitor home country analysis

```
#A choropleth map to demonstrate where most visitors come from
data(country.regions)
cleaned_data$country[cleaned_data$country=="CN"] <- "CHN"
country_data <- cleaned_data %>%
  select(iso2c = country) %>%
  group_by(iso2c) %>%
  summarize(value = n()) %>%
  arrange(iso2c)
code = country_data$iso2c
code = countrycode(code, 'iso3c', 'iso2c')
```

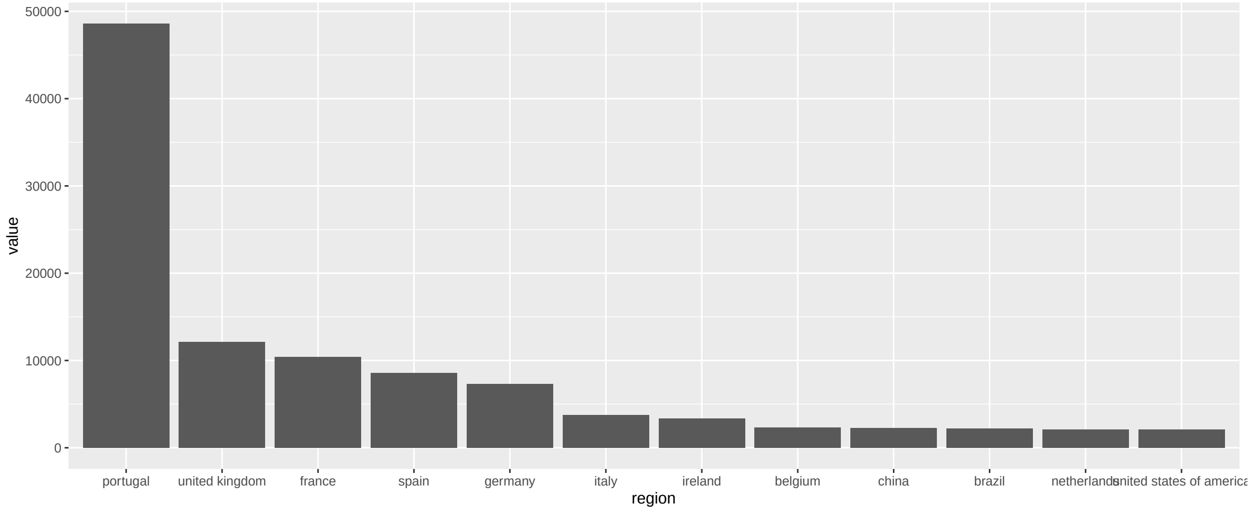
```
## Warning in countrycode(code, "iso3c", "iso2c"): Some values were not matched unambiguously: TMP
```

```
code[is.na(code)] <- "TL"
country_data$iso2c = code
country_data <- country_data %>%
  left_join(country.regions, by = "iso2c") %>%
  select(region, value)

country_data <- na.omit(country_data)
country_choropleth(country_data, title = "home country of hotel books", num_colors=9)
```

```
## Warning in self$bind(): The following regions were missing and are being set
## to NA: afghanistan, moldova, mongolia, namibia, niger, papua new guinea, north
## korea, western sahara, south sudan, solomon islands, somaliland, somalia,
## swaziland, chad, turkmenistan, trinidad and tobago, vanuatu, yemen, belize,
## brunei, bhutan, canada, democratic republic of the congo, republic of congo,
## eritrea, guinea, gambia, equatorial guinea, haiti, kyrgyzstan, kosovo, liberia,
## lesotho
```



```
#Barplot to show more clearly the countries with most visitors
country_data %>%
  arrange(desc(value)) %>%
  head(12) %>%
  ggplot(aes(x=reorder(region, -value), y=value))+
  geom_bar(stat="identity")+
  xlab("region")
```



## time and season analysis

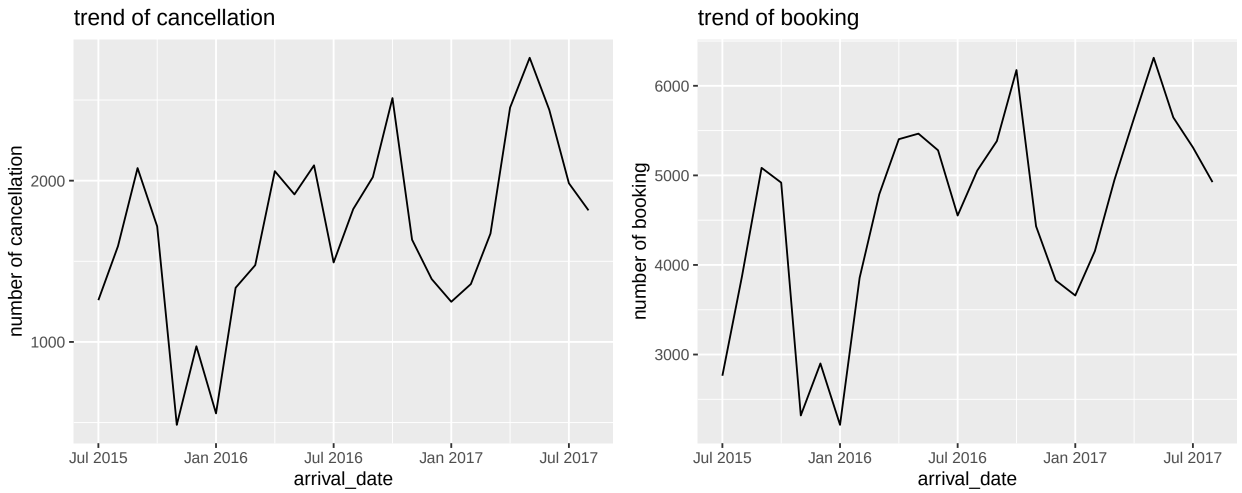
Then I want to analyze how number of cancels change according to season change. I first combine 3 arrival_date columns into one as yearmon format(xxxx-xx) to be able to plot through time. Then I count the number of cancels and number of bookings in each month separately and set them as y-axis.The plot underneath gives insight that trend of number of cancels is almost as same as trend of number of bookings.

```
#combine arrival date info into one column as date format
data <- cleaned_data
data$arrival_date_month <- str_sub(data$arrival_date_month,1,3)
data$arrival_date_month = match(data$arrival_date_month, month.abb)
data$arrival_date <- paste(data$arrival_date_year, data$arrival_date_month, data$arrival_date_day_of_mon
data$arrival_date <- ymd(data$arrival_date)
data$arrival_date <- as.Date(data$arrival_date, "%y/%m/%d")
data$arrival_date <- as.yearmon(data$arrival_date)
```

```
time_data <- data
head(data[, c(2, 4, 5, 7, 33)])
```

```
## # A tibble: 6 x 5
##   is_canceled arrival_date_year arrival_date_mo~ arrival_date_day_~ arrival_date
##   <fct>                   <int>            <int>              <int> <yearmon>
## 1 0                        2015                7                  1 Jul 2015
## 2 0                        2015                7                  1 Jul 2015
## 3 0                        2015                7                  1 Jul 2015
## 4 0                        2015                7                  1 Jul 2015
## 5 0                        2015                7                  1 Jul 2015
## 6 0                        2015                7                  1 Jul 2015
```

```
#compare the trend of cancellation and the trend of bookings
date <- data %>%
  select(is_canceled, arrival_date)
date_total <- date %>%
  group_by(arrival_date) %>%
  summarise(n = n())

date_canceled <- date %>%
  filter(is_canceled == 1) %>%
  group_by(arrival_date) %>%
  summarise(n = n())

total <- ggplot(date_canceled)+
  geom_line(aes(arrival_date, n))+
  ggtitle("trend of cancellation")+
  ylab("number of cancellation")

canceled <- ggplot(date_total)+
  geom_line(aes(arrival_date, n))+
  ggtitle("trend of booking")+
  ylab("number of booking")


plot_grid(total, canceled)
```



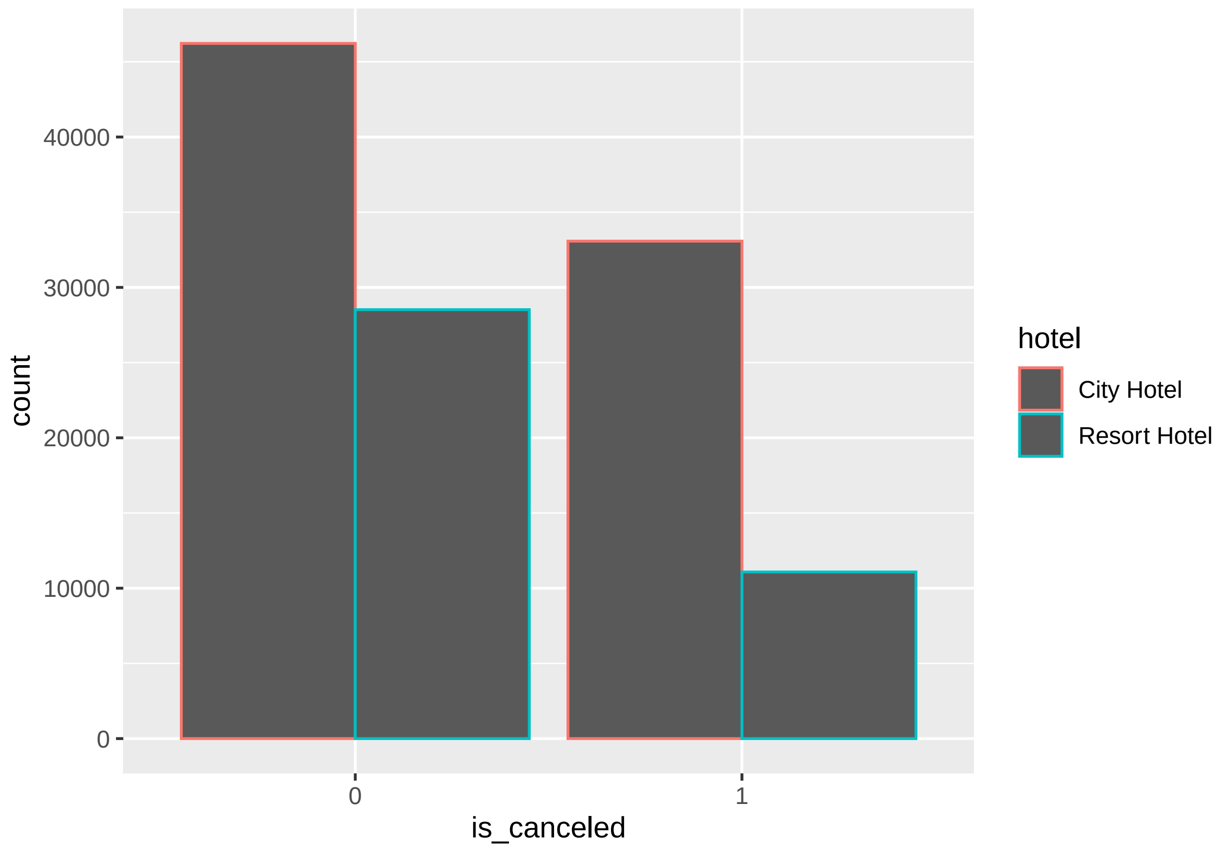
## hotel type analysis **Then I tried to find the relationship between number of cancels and price but first, let's take a look how different types may vary in number of cancels. The plot below shows that the proportion of cancels for city hotel is much higher than for resort city. Therefore, I think it is worth spliting those two types of hotel in later analysis**

```
require(tidyverse)
hotel_data <- cleaned_data %>%
  select(hotel, is_canceled)




hotel_data$is_canceled <- as.factor(hotel_data$is_canceled)

hotel_data %>%
  group_by(hotel, is_canceled) %>%
  summarise(n())
```

```
## `summarise()` has grouped output by 'hotel'. You can override using the `.groups` argument.
```

```
## # A tibble: 4 x 3
## # Groups:   hotel [2]
##   hotel        is_canceled `n()`
##   <chr>        <fct>       <int>
## 1 City Hotel   0           46226
## 2 City Hotel   1           33076
## 3 Resort Hotel 0           28519
## 4 Resort Hotel 1           11077
```

```
ggplot(hotel_data)+
  geom_bar(aes(is_canceled, color=hotel), position="dodge")
```



```r
city_hotel_cancelrate <- 33076/(46226+33076)
resort_hotel_cancelrate <- 11077/(28519+11077)

paste(c("city hotel cancel rate is:", city_hotel_cancelrate), collapse = " ")
```

```
## [1] "city hotel cancel rate is: 0.417089102418602"
```

```r
paste(c("resort hotel cancel rate is:", resort_hotel_cancelrate), collapse = " ")
```

```
## [1] "resort hotel cancel rate is: 0.279750479846449"
```

```r
#we can see city hotel has larger cancel rate. I think it is worth analyzing those two types seperately
```

hotel price analysis

adr here menas Average Daily Rate as defined by dividing the sum of all lodging transactions by the total number of staying nights

Again, I compare price and number of cancels as time goes. For the Resort hotel, the number of cancels is really stable compared to price change. I would say price does not influence cancels in Resort hotel a lot. However, for the city hotel, I think both price and number of cancels have a similar pattern though it is not really obvious. If we take a closer look, we can see that the direction change of price and cancel lines for City hotel are really similar. Therefore, I would conclude price and cancels have certain relationship for City hotel at least.

```
#price trend over the whole time
price_data <- time_data %>%
  select(hotel, arrival_date, adr)
dim(price_data[price_data$adr==0,])
```

```
## [1] 1938    3
```

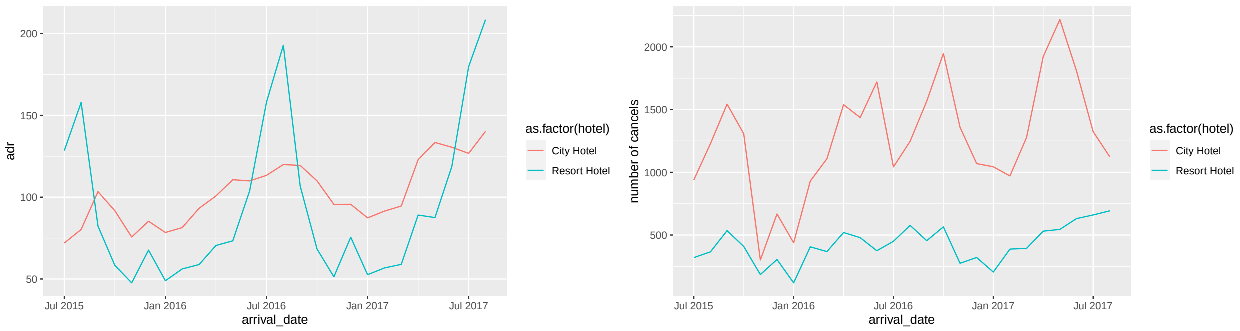
```
price_data <- price_data[price_data$adr!=0,]

price_data <- price_data %>%
  group_by(hotel, arrival_date) %>%
  summarise(adr=mean(adr))
```

```
## `summarise()` has grouped output by 'hotel'. You can override using the `.groups` argument.
```

```
price_plot <- ggplot(price_data, aes(arrival_date, adr))+
  geom_line(aes(color = as.factor(hotel)))+
  xlab("arrival_date")

#cancel_trend over the whole time
cancel_data <- time_data %>%
  select(hotel, arrival_date, is_canceled) %>%
  filter(is_canceled == 1) %>%
  group_by(hotel, arrival_date) %>%
  summarise(n=n())
```

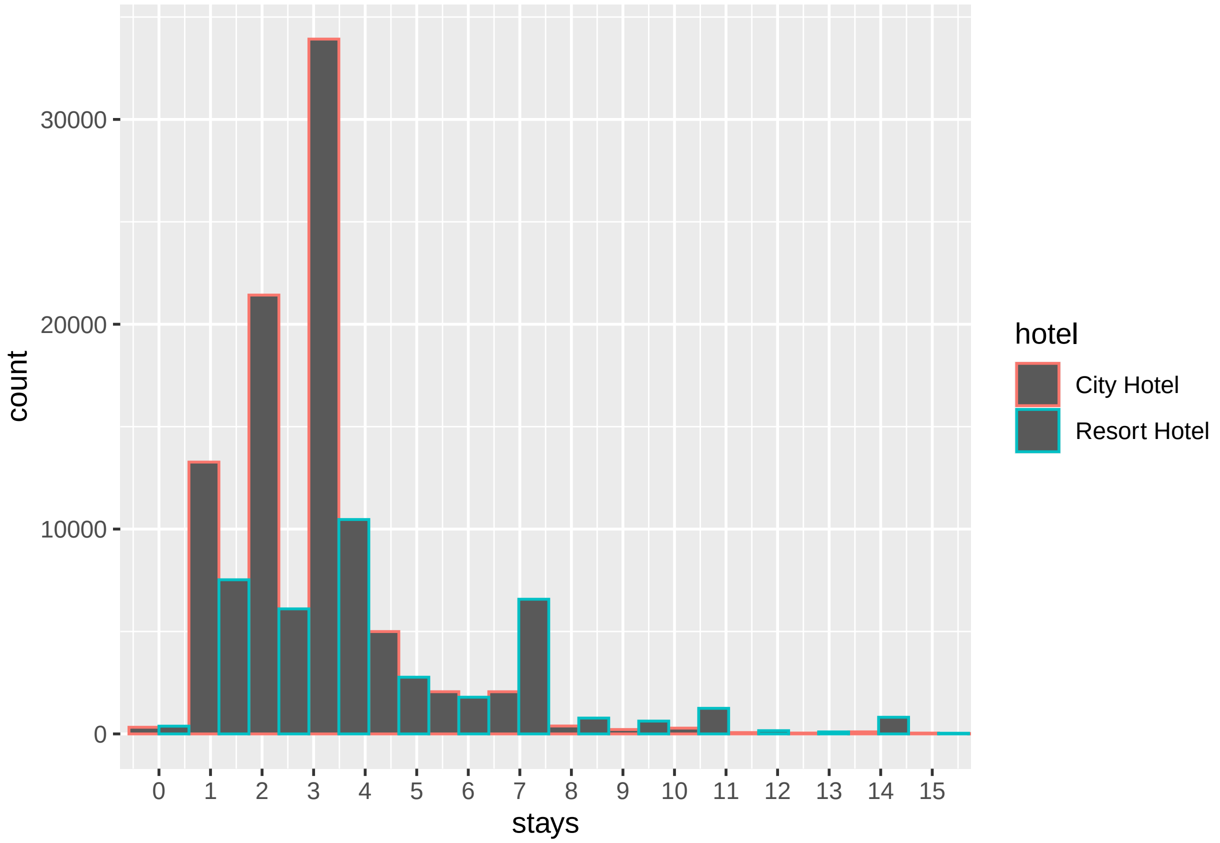
```
## `summarise()` has grouped output by 'hotel'. You can override using the `.groups` argument.
```

```
cancel_plot <- ggplot(cancel_data, aes(arrival_date, n))+
  geom_line(aes(color= as.factor(hotel)))+
  ylab("number of cancels")


plot_grid(price_plot, cancel_plot)
```



## stayed night analysis **stays_night is another numerical variable which could influence the prediction of cancellation so I would like to see its distribution here. The plot below shows that most people choose to stay 1-3 nights for both two types of hotel. However, it is interesting that a considerable number of people choose to live in Resort hotel for 7 nights(1-week). This might create some outliers in the following model fitting part.**

```
library(scales)
stay_data <- cleaned_data %>%
  mutate(stays=stays_in_weekend_nights+stays_in_week_nights) %>%
  select(hotel, stays)


#get all stayed nights value for x labels
x_axis_labels <- min(stay_data[,2]):max(stay_data[,2])

stay_data %>%
  ggplot(aes(stays, color = hotel))+
  geom_histogram(bins = 50, position = "dodge")+
  coord_cartesian(xlim=c(0, 15))+
  scale_x_continuous(labels = x_axis_labels, breaks = x_axis_labels)
```

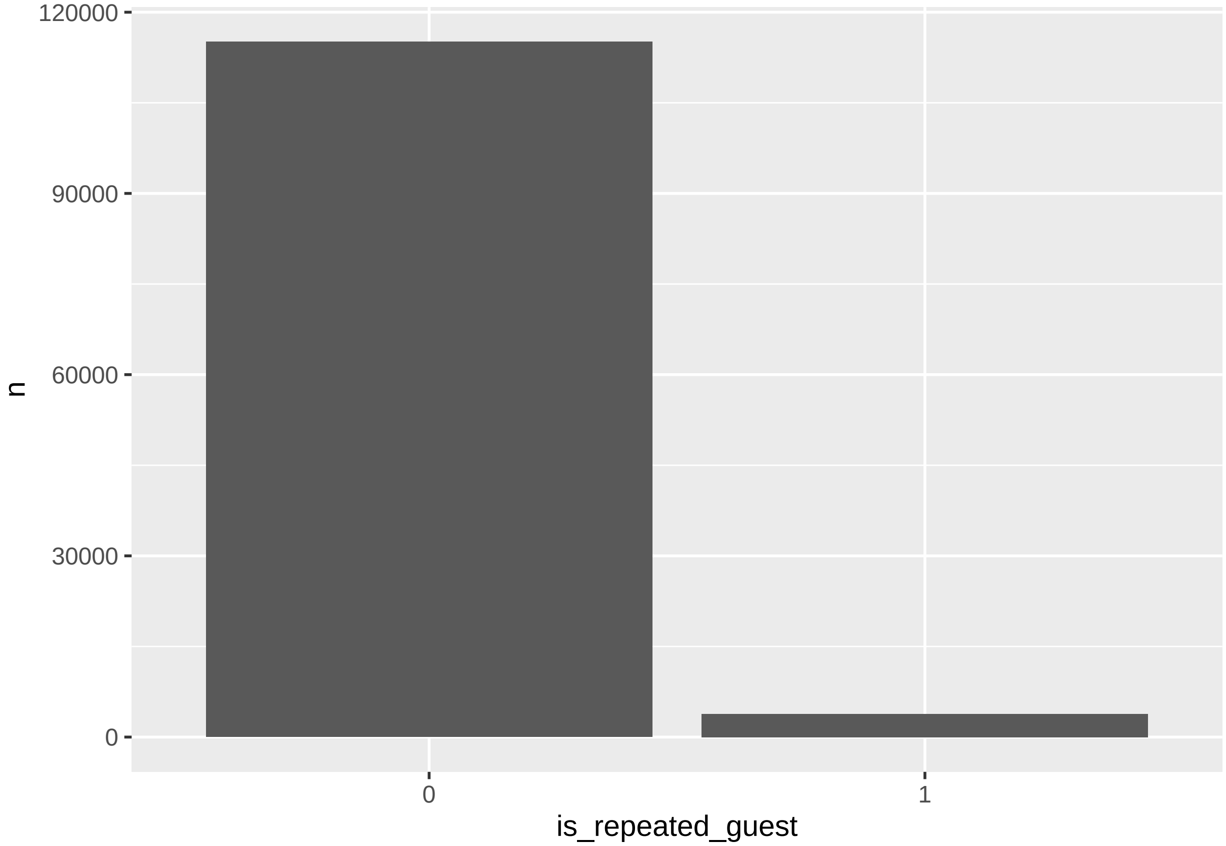


## repeated guests analysis **Finally, I want to look at the proportion of repeated guests in the full data. Unfortunately, there is only 5% repeated guests in this data collection. The sample would be too small if I continue split them by hotel type. Therefore, I stop the analysis here.**

```
rg_data <- cleaned_data %>%
  select(hotel,is_repeated_guest) %>%
  #mutate(is_canceled = as.factor(is_canceled)) %>%
  mutate(is_repeated_guest = as.factor(is_repeated_guest)) %>%
  group_by(hotel, is_repeated_guest) %>%
  summarise(n=n())
```

```
## `summarise()` has grouped output by 'hotel'. You can override using the `.groups` argument.
```

```
rg_data %>%
  ggplot(aes(is_repeated_guest, n))+
  geom_bar(stat = "identity")
```



```
repeats_rate <-length(which(rg_data$is_repeated_guest == 0))/length(rg_data$is_repeated_guest)
paste(c("The proportion of repeated guest is", repeats_rate), collapse = " ")
```

```
## [1] "The proportion of repeated guest is 0.5"
```

```
#only 5 percent are repeated guest so I do not think it is worth grouping by hotel and work on the canc
```

Model Fitting

apply logistic regression model

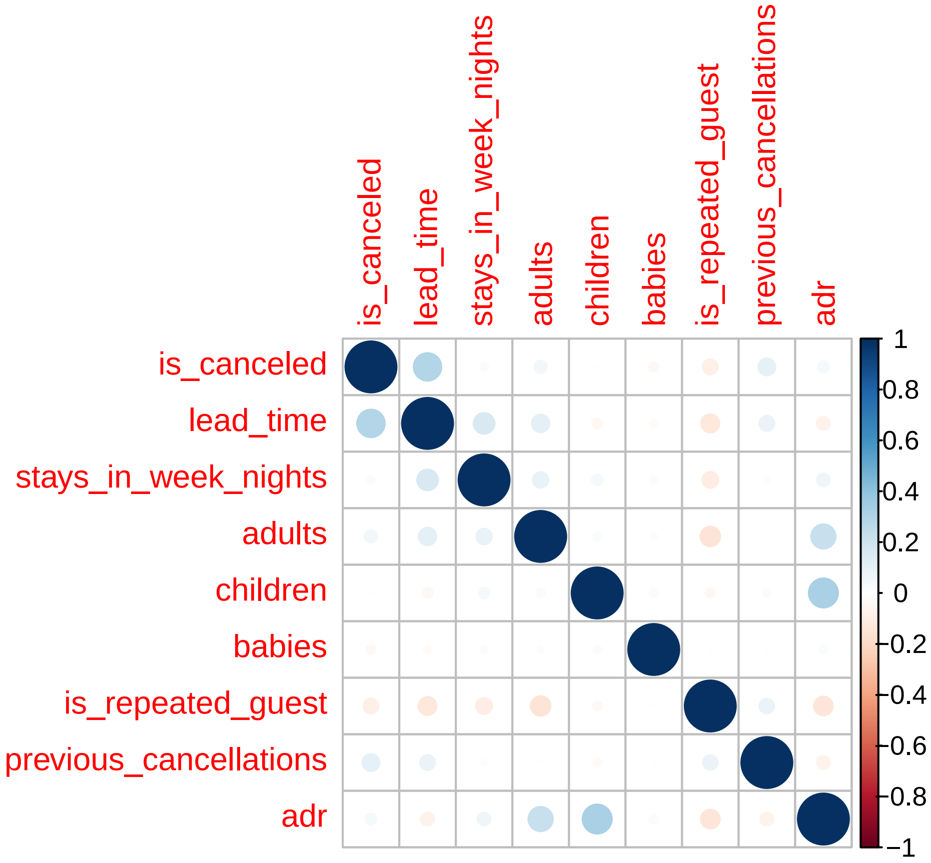
I fit a logistic regression model here to predict if guests cancel or not. I first plot a corr plot to show the correlations among some important features. Without using cross validation, I got a accuracy of 0.68. It is not a good enough result but I just stop here.

```
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.0.3
```

```
## corrplot 0.84 loaded
```

```
corr_data <- cleaned_data[,c(2:3, 9:12, 17:18, 28) ]
corr_data <- apply(corr_data, 2, as.numeric)
corr_data <- as.data.frame(corr_data)
M <- cor(corr_data)
corrplot(M, method = "circle")
```



```
train <- corr_data[1:50000, ]
test <- corr_data[50001:118898,]

model <- glm(is_canceled ~ .,
             data = train,
             family = binomial((link = 'logit')))
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(model)
```

```
##
## Call:
## glm(formula = is_canceled ~ ., family = binomial((link = "logit")),
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.3101  -0.8774  -0.7489   1.3037   2.7898
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)            -1.4636175  0.0411926 -35.531  < 2e-16 ***
## lead_time               0.0039795  0.0001147  34.689  < 2e-16 ***
## stays_in_week_nights   -0.0511543  0.0048271 -10.597  < 2e-16 ***
## adults                  0.0916350  0.0208225   4.401 1.08e-05 ***
## children                0.1225615  0.0234970   5.216 1.83e-07 ***
## babies                 -0.7924877  0.1044142  -7.590 3.20e-14 ***
## is_repeated_guest      -2.5529861  0.1260533 -20.253  < 2e-16 ***
## previous_cancellations  2.6155759  0.1067349  24.505  < 2e-16 ***
## adr                     0.0035112  0.0001904  18.437  < 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: 63274  on 49999  degrees of freedom
## Residual deviance: 59093  on 49991  degrees of freedom
## AIC: 59111
##
## Number of Fisher Scoring iterations: 7
```

```
fitted.results <- predict(model,newdata=test,type='response')
fitted.results <- ifelse(fitted.results > 0.5,1,0)
misClasificError <- mean(fitted.results != test$is_canceled)
print(paste('Accuracy',1-misClasificError))
```

```
## [1] "Accuracy 0.686551133559755"
```

Conclusion

Based on made analysis, I would say number of cancels is positively proportional to the number of bookings. Guests in different types of hotel tend to have different cancel behavior and staying time. Price has certain level of impact on number of cancels for City hotel but not obvious. As for Resort hotel, price does not have any apparent influence.