Project_stat

I.Caria, C.De Luca, M.Pernini

13/06/2022

Libraries

```
library(ggplot2)
library(tidyverse)
library(leaps)
library(ggcorrplot)
library(regclass)
library(boot)
#library(caret)
library(MASS)
library(knitr)
library(corrplot)
library(glmnet)
library(plotly)
```

Hotel booking demand dataset

We decided to analyze the *Hotel booking demand dataset* that we load from Kaggle. This dataset contains information about two different kinds of hotel: City Hotel and Resort Hotel. Each observation represents an ho.tel booking. Both hotels are located in Portugal: the resort hotel at the resort region of Algarve and the city hotel at the city of Lisbon.

```
# Load the dataset

hotel_bookings <- read.csv("hotel_bookings.csv", na.strings="NULL")
View(hotel_bookings)
```

Dataset Pre-Processing

The dataset contains 32 variables describing 119390 observations.

```
# First look to the dataset

glimpse(hotel_bookings)
```

```
## Rows: 119,390
## Columns: 32
```

```
## $ hotel                           <chr> "Resort Hotel", "Resort Hotel", "Resort~
## $ is_canceled                     <int> 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, ~
## $ lead_time                       <int> 342, 737, 7, 13, 14, 14, 0, 9, 85, 75, ~
## $ arrival_date_year               <int> 2015, 2015, 2015, 2015, 2015, 2015, 201~
## $ arrival_date_month              <chr> "July", "July", "July", "July", "July",~
## $ arrival_date_week_number        <int> 27, 27, 27, 27, 27, 27, 27, 27, 27, 27,~
## $ arrival_date_day_of_month       <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ stays_in_weekend_nights         <int> 0, 0, 0, 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, 4, 4, 4, ~
## $ adults                          <int> 2, 2, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, ~
## $ children                        <chr> "0", "0", "0", "0", "0", "0", "0", "0",~
## $ babies                          <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ meal                            <chr> "BB", "BB", "BB", "BB", "BB", "BB", "BB~
## $ country                         <chr> "PRT", "PRT", "GBR", "GBR", "GBR", "GBR~
## $ market_segment                  <chr> "Direct", "Direct", "Direct", "Corporat~
## $ distribution_channel            <chr> "Direct", "Direct", "Direct", "Corporat~
## $ is_repeated_guest               <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ previous_cancellations          <int> 0, 0, 0, 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, 0, 0, 0, ~
## $ reserved_room_type              <chr> "C", "C", "A", "A", "A", "A", "C", "C",~
## $ assigned_room_type              <chr> "C", "C", "C", "A", "A", "A", "C", "C",~
## $ booking_changes                 <int> 3, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ deposit_type                    <chr> "No Deposit", "No Deposit", "No Deposit~
## $ agent                           <int> NA, NA, NA, 304, 240, 240, NA, 303, 240~
## $ company                         <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ days_in_waiting_list            <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ customer_type                   <chr> "Transient", "Transient", "Transient", ~
## $ adr                             <dbl> 0.00, 0.00, 75.00, 75.00, 98.00, 98.00,~
## $ required_car_parking_spaces     <int> 0, 0, 0, 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, 0, 0, 3, ~
## $ reservation_status              <chr> "Check-Out", "Check-Out", "Check-Out", ~
## $ reservation_status_date         <chr> "2015-07-01", "2015-07-01", "2015-07-02~
```

As we can see from the code above, there are many character variables that we converted into factors. Furthermore, we noticed that some categorical variables like *children* were numeric, so we converted them.

```
# Convert character columns into factors

hotel_bookings_new <- as.data.frame(unclass(hotel_bookings),
                       stringsAsFactors = TRUE)

# Convert binary columns "is_canceled" and "is_repeated_guest" into factor

hotel_bookings_new$is_canceled <- as.factor(hotel_bookings_new$is_canceled)
levels(hotel_bookings_new$is_canceled) <- c(0, 1)

hotel_bookings_new$is_repeated_guest <- as.factor(hotel_bookings_new$is_repeated_guest)
levels(hotel_bookings_new$is_repeated_guest) <- c(0, 1)

# Convert column "arrival_date_year" into factor

hotel_bookings_new$arrival_date_year <- as.factor(hotel_bookings_new$arrival_date_year)
levels(hotel_bookings_new$arrival_date_year) <- c("2015", "2016", "2017")
```

```
# Convert column "children" into numeric

hotel_bookings_new$children <- as.numeric(as.character(hotel_bookings_new$children))

# Convert column "reservation status date" into date

hotel_bookings_new$reservation_status_date <- as.Date(
  hotel_bookings_new$reservation_status_date, format = "%Y-%m-%d")
```

The dataset provides two different variables for the stay: *stays_in_weekend_nights* and *stays_in_week_nights*. We decided to add the sum of these two variables as a new variable *total_stays* for ease of analyses.

```
# New column for total stays

hotel_bookings_new=hotel_bookings_new%>%
  mutate(total_stays=(stays_in_week_nights + stays_in_weekend_nights) )
```

Missing values

```
colSums(is.na(hotel_bookings_new))[colSums(is.na(hotel_bookings_new))>0]
```

```
## children  country    agent  company
##        4      488    16340   112593
```

Since there are only 4 Nan values for the variable *children*, we decided to replace them with the value 0. The variables *agent* and *company* have too many Nan values, therefore we removed them. We left untouched the variable *country* because we did not use it in our models.

```
# Replacing missing values in children column from the corresponding babies column

n_children <- length(hotel_bookings_new$children)
for (i in 1:n_children) {
  if (is.na(hotel_bookings_new$children[i]))
    hotel_bookings_new$children[i] <- 0
}

# Remove columns "agent" and "company"

index_agent <- which(colnames(hotel_bookings_new)=="agent")
index_company <- which(colnames(hotel_bookings_new)=="company")
hotel_bookings_new = hotel_bookings_new[-c(index_agent, index_company)]
```

At the end of the pre-processing, we obtained the following dataset:

```
##           hotel       is_canceled   lead_time   arrival_date_year
##  City Hotel  :79330   0:75166     Min.   :  0   2015:21996
##  Resort Hotel:40060   1:44224     1st Qu.: 18   2016:56707
##                                   Median : 69   2017:40687
##                                   Mean   :104
##                                   3rd Qu.:160
```

```
##                                 Max.   :737
##
##  arrival_date_month arrival_date_week_number arrival_date_day_of_month
##  August :13877      Min.   : 1.00            Min.   : 1.0
##  July   :12661      1st Qu.:16.00            1st Qu.: 8.0
##  May    :11791      Median :28.00            Median :16.0
##  October:11160      Mean   :27.17            Mean   :15.8
##  April  :11089      3rd Qu.:38.00            3rd Qu.:23.0
##  June   :10939      Max.   :53.00            Max.   :31.0
##  (Other):47873
##  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   BB       :92310   PRT    :48590
##  1st Qu.: 0.0000   1st Qu.: 0.000000   FB       :  798   GBR    :12129
##  Median : 0.0000   Median : 0.000000   HB       :14463   FRA    :10415
##  Mean   : 0.1039   Mean   : 0.007949   SC       :10650   ESP    : 8568
##  3rd Qu.: 0.0000   3rd Qu.: 0.000000   Undefined: 1169   DEU    : 7287
##  Max.   :10.0000   Max.   :10.000000                     (Other):31913
##                                                          NA's   :  488
##        market_segment  distribution_channel is_repeated_guest
##  Online TA    :56477   Corporate: 6677      0:115580
##  Offline TA/TO:24219   Direct   :14645      1:  3810
##  Groups       :19811   GDS      :  193
##  Direct       :12606   TA/TO    :97870
##  Corporate    : 5295   Undefined:    5
##  Complementary:  743
##  (Other)      :  239
##  previous_cancellations previous_bookings_not_canceled reserved_room_type
##  Min.   : 0.00000       Min.   : 0.0000                A      :85994
##  1st Qu.: 0.00000       1st Qu.: 0.0000                D      :19201
##  Median : 0.00000       Median : 0.0000                E      : 6535
##  Mean   : 0.08712       Mean   : 0.1371                F      : 2897
##  3rd Qu.: 0.00000       3rd Qu.: 0.0000                G      : 2094
##  Max.   :26.00000       Max.   :72.0000                B      : 1118
##                                                        (Other): 1551
##  assigned_room_type booking_changes       deposit_type    days_in_waiting_list
##  A      :74053      Min.   : 0.0000   No Deposit:104641   Min.   :  0.000
##  D      :25322      1st Qu.: 0.0000   Non Refund: 14587   1st Qu.:  0.000
##  E      : 7806      Median : 0.0000   Refundable:   162   Median :  0.000
##  F      : 3751      Mean   : 0.2211                       Mean   :  2.321
##  G      : 2553      3rd Qu.: 0.0000                       3rd Qu.:  0.000
##  C      : 2375      Max.   :21.0000                       Max.   :391.000
##  (Other): 3530
##          customer_type        adr          required_car_parking_spaces
##  Contract       : 4076   Min.   :  -6.38   Min.   :0.00000
##  Group          :  577   1st Qu.:  69.29   1st Qu.:0.00000
##  Transient      :89613   Median :  94.58   Median :0.00000
```

```
##  Transient-Party:25124   Mean   : 101.83   Mean   :0.06252
##                          3rd Qu.: 126.00   3rd Qu.:0.00000
##                          Max.   :5400.00   Max.   :8.00000
##
##  total_of_special_requests reservation_status reservation_status_date
##  Min.   :0.0000            Canceled :43017    Min.   :2014-10-17
##  1st Qu.:0.0000            Check-Out:75166    1st Qu.:2016-02-01
##  Median :0.0000            No-Show  : 1207    Median :2016-08-07
##  Mean   :0.5714                               Mean   :2016-07-30
##  3rd Qu.:1.0000                               3rd Qu.:2017-02-08
##  Max.   :5.0000                               Max.   :2017-09-14
##
##   total_stays
##  Min.   : 0.000
##  1st Qu.: 2.000
##  Median : 3.000
##  Mean   : 3.428
##  3rd Qu.: 4.000
##  Max.   :69.000
##
```

EDA

As stated above,the dataset contains information about two different kinds of hotel: City Hotel and Resort Hotel. There are 79330 observation for the former and 40060 for the latter.

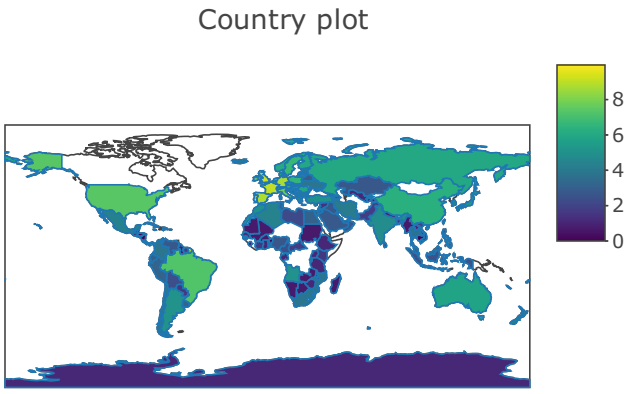
```
# Hotel donut plot

df_hotel <- as.data.frame(hotel_bookings_new[, c("hotel")])
df_hotel <- as.data.frame(lapply(df_hotel, function(x) as.data.frame(table(x))))

colnames(df_hotel) <- c("hotel", "frequency")

df_hotel
```

```
##          hotel frequency
## 1   City Hotel     79330
## 2 Resort Hotel     40060
```

```
colors_donut <- c('rgb((119,51,68))','rgb((227,181,164))')

fig_hotel <- df_hotel %>% plot_ly(labels = ~hotel, values = ~frequency,
                                  marker = list(colors = colors_donut,
                                  line = list(color = '#FFFFFF', width = 1)))
fig_hotel <- fig_hotel %>% add_pie(hole = 0.6)
fig_hotel <- fig_hotel %>% layout(title = "Total number of booking for each hotel",
                      showlegend = T,
                      xaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE),
                      yaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE))

fig_hotel
```

Both hotels are located in Portugal; this is the reason why most of the guests come from Portugal, as we can see from the map plot below:

```
# Country plot

df_country <- as.data.frame(hotel_bookings_new[hotel_bookings_new$is_canceled==0, c("country")])
df_country <- as.data.frame(lapply(df_country, function(x) as.data.frame(table(x))))
colnames(df_country) <- c('country', 'frequency')

# we used log-scale to see better the different number of guests for each country
fig_country <- plot_ly(df_country, type='choropleth', locations=df_country$country,
                       z=log(df_country$frequency+1), colorscale = 'Viridis' )
fig_country <- fig_country %>% layout(title = "Country plot")
fig_country
```



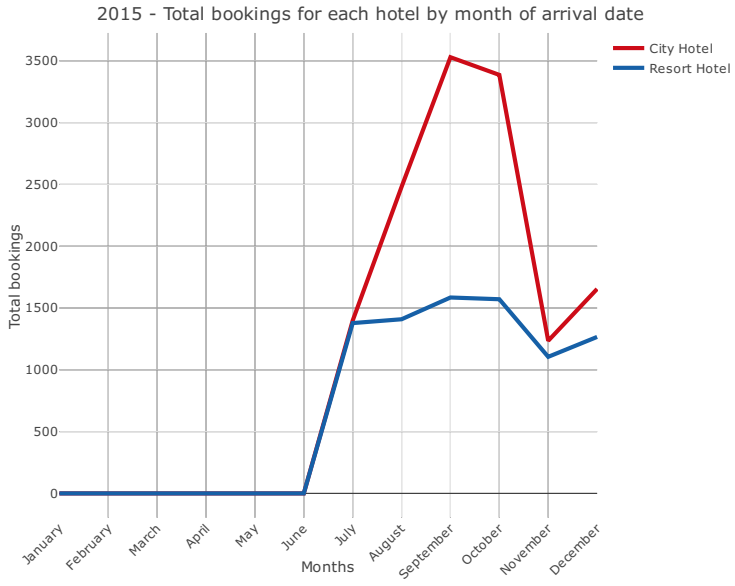
Total bookings considering both canceled and not canceled Total guests considering only is_canceled==0

```
# Total bookings for each hotel by month of arrival date

df_months_City <- as.data.frame(hotel_bookings_new[hotel_bookings_new$hotel=='City Hotel',
                                                   c( "arrival_date_month")])

df_months_Resort <- as.data.frame(hotel_bookings_new[hotel_bookings_new$hotel=='Resort Hotel',
                                                     c( "arrival_date_month")])

df_months_City <- as.data.frame(lapply(df_months_City, function(x) as.data.frame(table(x))))

df_months_Resort <- as.data.frame(lapply(df_months_Resort, function(x) as.data.frame(table(x))))

colnames(df_months_City) <- c("arrival_date_month_City", "frequency_City" )

colnames(df_months_Resort) <- c("arrival_date_month_Resort", "frequency_Resort")

df_months_City$arrival_date_month_City <- factor(df_months_City$arrival_date_month_City,
          levels = c("January", "February", "March","April", "May", "June", "July",
                     "August", "September", "October", "November", "December"))

df_months_Resort$arrival_date_month_Resort <- factor(df_months_Resort$arrival_date_month_Resort,
          levels = c("January", "February", "March","April", "May", "June", "July",
                     "August", "September", "October", "November", "December"))
```

```
fig_months <- plot_ly()

fig_months <- fig_months %>% add_lines(data=df_months_City,
                                       x = ~arrival_date_month_City, y = ~frequency_City,
                                       name = 'City Hotel', type = 'scatter', mode = 'lines',
                                       line = list(color = 'rgb(205, 12, 24)', width = 4))

fig_months <- fig_months %>% add_lines(data=df_months_Resort, x = ~arrival_date_month_Resort,
                                       y = ~frequency_Resort, name = 'Resort Hotel',
                                       type = 'scatter', mode = 'lines',
                                       line = list(color = 'rgb(22, 96, 167)', width = 4))

fig_months <- fig_months %>%
layout(title = "Total bookings for each hotel by month of arrival date",
         xaxis = list(title = "Months"),
         yaxis = list (title = "Total bookings"))

fig_months
```

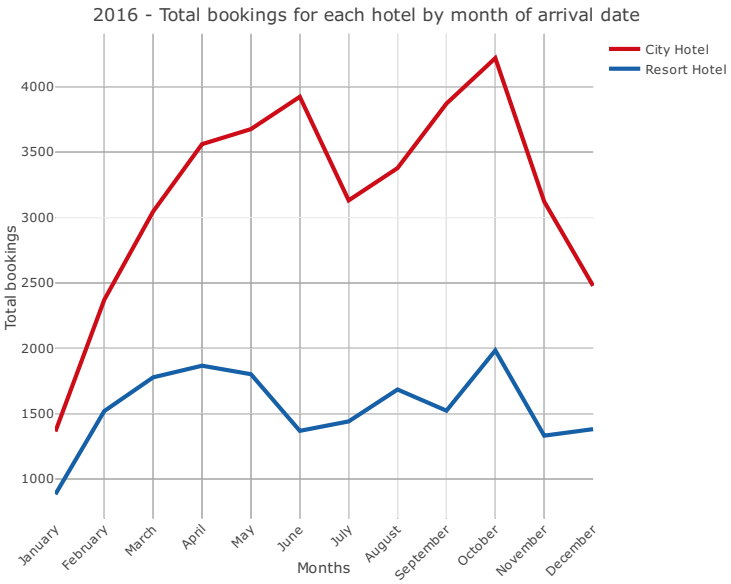
For each year

```r
# Total bookings for each hotel by month (year: 2015)

df_months_City_2015 <- as.data.frame(hotel_bookings_new[hotel_bookings_new$hotel=='City Hotel'
             & hotel_bookings_new$arrival_date_year==2015, c( "arrival_date_month")])

df_months_Resort_2015 <- as.data.frame(hotel_bookings_new[hotel_bookings_new$hotel=='Resort Hotel'
             & hotel_bookings_new$arrival_date_year==2015, c( "arrival_date_month")])

df_months_City_2015 <- as.data.frame(lapply(df_months_City_2015, function(x)
  as.data.frame(table(x))))

df_months_Resort_2015 <- as.data.frame(lapply(df_months_Resort_2015, function(x)
  as.data.frame(table(x))))

colnames(df_months_City_2015) <- c("arrival_date_month_City", "frequency_City" )

colnames(df_months_Resort_2015) <- c("arrival_date_month_Resort", "frequency_Resort")

df_months_City_2015$arrival_date_month_City <- factor(df_months_City_2015$arrival_date_month_City,
           levels = c("January", "February", "March","April", "May", "June", "July",
                      "August", "September", "October", "November", "December"))

df_months_Resort_2015$arrival_date_month_Resort <- factor(df_months_Resort_2015$arrival_date_month_Reso


fig_months_2015 <- plot_ly()

fig_months_2015 <- fig_months_2015 %>% add_lines(data=df_months_City_2015, x = ~arrival_date_month_City

fig_months_2015 <- fig_months_2015 %>% add_lines(data=df_months_Resort_2015, x = ~arrival_date_month_Res

fig_months_2015 <- fig_months_2015 %>% layout(title = "2015 - Total bookings for each hotel by month of
         xaxis = list(title = "Months",tickangle = -45),
         yaxis = list (title = "Total bookings"))
fig_months_2015
```



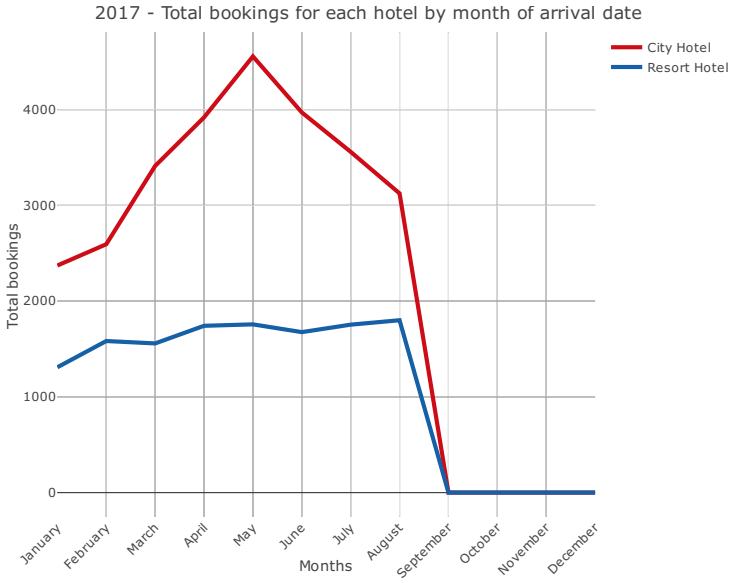

```
# Total bookings for each hotel by month (year: 2016)

df_months_City_2016 <- as.data.frame(hotel_bookings_new[hotel_bookings_new$hotel=='City Hotel' & hotel_

df_months_Resort_2016 <- as.data.frame(hotel_bookings_new[hotel_bookings_new$hotel=='Resort Hotel'& hote

df_months_City_2016 <- as.data.frame(lapply(df_months_City_2016, function(x) as.data.frame(table(x))))

df_months_Resort_2016 <- as.data.frame(lapply(df_months_Resort_2016, function(x) as.data.frame(table(x)

colnames(df_months_City_2016) <- c("arrival_date_month_City", "frequency_City" )

colnames(df_months_Resort_2016) <- c("arrival_date_month_Resort", "frequency_Resort")

df_months_City_2016$arrival_date_month_City <- factor(df_months_City_2016$arrival_date_month_City, level

df_months_Resort_2016$arrival_date_month_Resort <- factor(df_months_Resort_2016$arrival_date_month_Reso


fig_months_2016 <- plot_ly()
```

```
fig_months_2016 <- fig_months_2016 %>% add_lines(data=df_months_City_2016, x = ~arrival_date_month_City

fig_months_2016 <- fig_months_2016 %>% add_lines(data=df_months_Resort_2016, x = ~arrival_date_month_Res

fig_months_2016 <- fig_months_2016 %>% layout(title = "2016 - Total bookings for each hotel by month of
         xaxis = list(title = "Months",tickangle = -45),
         yaxis = list (title = "Total bookings"))

fig_months_2016
```



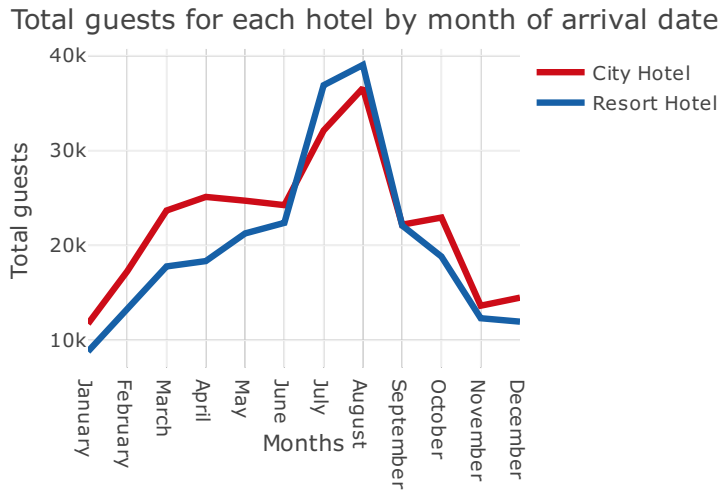

```
# Total bookings for each hotel by month (year: 2017)

df_months_City_2017 <- as.data.frame(hotel_bookings_new[hotel_bookings_new$hotel=='City Hotel' & hotel_

df_months_Resort_2017 <- as.data.frame(hotel_bookings_new[hotel_bookings_new$hotel=='Resort Hotel'& hote

df_months_City_2017 <- as.data.frame(lapply(df_months_City_2017, function(x) as.data.frame(table(x))))

df_months_Resort_2017 <- as.data.frame(lapply(df_months_Resort_2017, function(x) as.data.frame(table(x)

colnames(df_months_City_2017) <- c("arrival_date_month_City", "frequency_City" )
```

```r
colnames(df_months_Resort_2017) <- c("arrival_date_month_Resort", "frequency_Resort")

df_months_City_2017$arrival_date_month_City <- factor(df_months_City_2017$arrival_date_month_City, level

df_months_Resort_2017$arrival_date_month_Resort <- factor(df_months_Resort_2017$arrival_date_month_Reso


fig_months_2017 <- plot_ly()

fig_months_2017 <- fig_months_2017 %>% add_lines(data=df_months_City_2017, x = ~arrival_date_month_City

fig_months_2017 <- fig_months_2017 %>% add_lines(data=df_months_Resort_2017, x = ~arrival_date_month_Res

fig_months_2017 <- fig_months_2017 %>% layout(title = "2017 - Total bookings for each hotel by month of
         xaxis = list(title = "Months",tickangle = -45),
         yaxis = list (title = "Total bookings"))

fig_months_2017
```



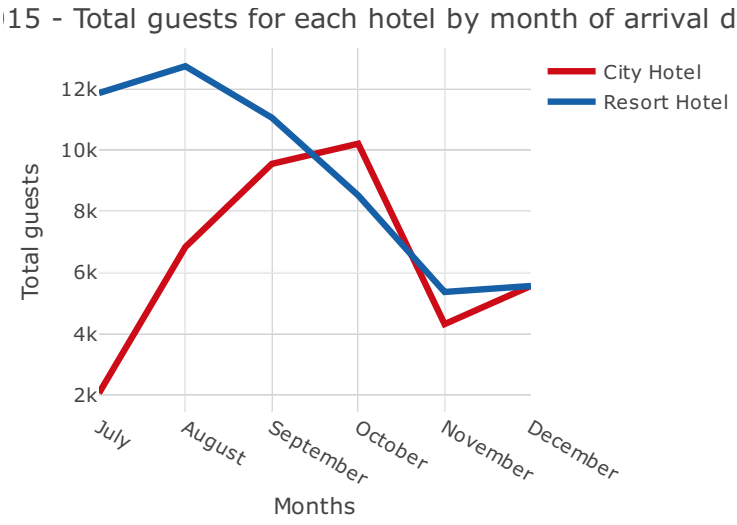
Total guests

```r
# Total guests for each hotel by month

df_City <- as.data.frame(hotel_bookings_new[hotel_bookings_new$is_canceled==0 & hotel_bookings_new$hotel

df_guests_City <- df_City %>%
  group_by(arrival_date_month) %>%
  summarise(guests = sum(total_stays*(adults+children+babies))) %>%
  ungroup()

df_Resort <- as.data.frame(hotel_bookings_new[hotel_bookings_new$is_canceled==0 & hotel_bookings_new$ho

df_guests_Resort <- df_Resort %>%
  group_by(arrival_date_month) %>%
  summarise(guests = sum(total_stays*(adults+children+babies))) %>%
  ungroup()


df_guests_City$arrival_date_month <- factor(df_guests_City$arrival_date_month, levels = c("January", "F

df_guests_Resort$arrival_date_month <- factor(df_guests_Resort$arrival_date_month, levels = c("January"


fig_guests <- plot_ly()

fig_guests <- fig_guests %>% add_lines(data=df_guests_City, x = ~arrival_date_month, y = ~guests, name =

fig_guests <- fig_guests %>% add_lines(data=df_guests_Resort, x = ~arrival_date_month, y = ~guests, nam

fig_guests <- fig_guests %>% layout(title = "Total guests for each hotel by month of arrival date",
         xaxis = list(title = "Months"),
         yaxis = list (title = "Total guests"))

fig_guests
```



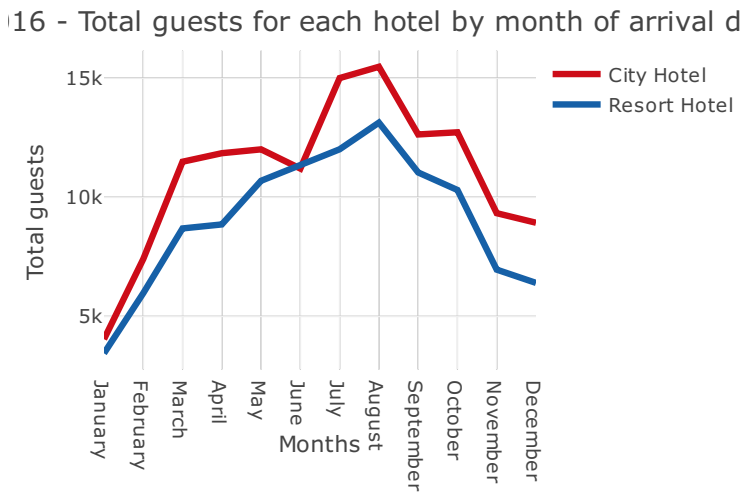

```
# Total guests for each hotel by month (year: 2015)

df_2015_City <- as.data.frame(hotel_bookings_new[hotel_bookings_new$is_canceled==0 & hotel_bookings_new$

df_guests_2015_City <- df_2015_City %>%
  group_by(arrival_date_month) %>%
  summarise(guests = sum(total_stays*(adults+children+babies))) %>%
  ungroup()

df_2015_Resort <- as.data.frame(hotel_bookings_new[hotel_bookings_new$is_canceled==0 & hotel_bookings_ne

df_guests_2015_Resort <- df_2015_Resort %>%
  group_by(arrival_date_month) %>%
  summarise(guests = sum(total_stays*(adults+children+babies))) %>%
  ungroup()


df_guests_2015_City$arrival_date_month <- factor(df_guests_2015_City$arrival_date_month, levels = c("Jul

df_guests_2015_Resort$arrival_date_month <- factor(df_guests_2015_Resort$arrival_date_month, levels = c


fig_guests_2015 <- plot_ly()

fig_guests_2015 <- fig_guests_2015 %>% add_lines(data=df_guests_2015_City, x = ~arrival_date_month, y =
```

```
fig_guests_2015 <- fig_guests_2015 %>% add_lines(data=df_guests_2015_Resort, x = ~arrival_date_month, y

fig_guests_2015 <- fig_guests_2015 %>% layout(title = "2015 - Total guests for each hotel by month of a
         xaxis = list(title = "Months"),
         yaxis = list (title = "Total guests"))

fig_guests_2015
```




```
# Total guests for each hotel by month (year: 2016)

df_2016_City <- as.data.frame(hotel_bookings_new[hotel_bookings_new$is_canceled==0 & hotel_bookings_new

df_guests_2016_City <- df_2016_City %>%
  group_by(arrival_date_month) %>%
  summarise(guests = sum(total_stays*(adults+children+babies))) %>%
  ungroup()

df_2016_Resort <- as.data.frame(hotel_bookings_new[hotel_bookings_new$is_canceled==0 & hotel_bookings_ne

df_guests_2016_Resort <- df_2016_Resort %>%
  group_by(arrival_date_month) %>%
  summarise(guests = sum(total_stays*(adults+children+babies))) %>%
  ungroup()
```

```
df_guests_2016_City$arrival_date_month <- factor(df_guests_2016_City$arrival_date_month, levels = c("Ja

df_guests_2016_Resort$arrival_date_month <- factor(df_guests_2016_Resort$arrival_date_month, levels = c


fig_guests_2016 <- plot_ly()

fig_guests_2016 <- fig_guests_2016 %>% add_lines(data=df_guests_2016_City, x = ~arrival_date_month, y =

fig_guests_2016 <- fig_guests_2016 %>% add_lines(data=df_guests_2016_Resort, x = ~arrival_date_month, y

fig_guests_2016 <- fig_guests_2016 %>% layout(title = "2016 - Total guests for each hotel by month of a
         xaxis = list(title = "Months"),
         yaxis = list (title = "Total guests"))

fig_guests_2016
```




```
# Total guests for each hotel by month (year: 2017)

df_2017_City <- as.data.frame(hotel_bookings_new[hotel_bookings_new$is_canceled==0 & hotel_bookings_new

df_guests_2017_City <- df_2017_City %>%
  group_by(arrival_date_month) %>%
  summarise(guests = sum(total_stays*(adults+children+babies))) %>%
  ungroup()
```

```
df_2017_Resort <- as.data.frame(hotel_bookings_new[hotel_bookings_new$is_canceled==0 & hotel_bookings_ne

df_guests_2017_Resort <- df_2017_Resort %>%
  group_by(arrival_date_month) %>%
  summarise(guests = sum(total_stays*(adults+children+babies))) %>%
  ungroup()


df_guests_2017_City$arrival_date_month <- factor(df_guests_2017_City$arrival_date_month, levels = c("Jan

df_guests_2017_Resort$arrival_date_month <- factor(df_guests_2017_Resort$arrival_date_month, levels = c


fig_guests_2017 <- plot_ly()

fig_guests_2017 <- fig_guests_2017 %>% add_lines(data=df_guests_2017_City, x = ~arrival_date_month, y =

fig_guests_2017 <- fig_guests_2017 %>% add_lines(data=df_guests_2017_Resort, x = ~arrival_date_month, y

fig_guests_2017 <- fig_guests_2017 %>% layout(title = "2016 - Total guests for each hotel by month of ar
         xaxis = list(title = "Months"),
         yaxis = list (title = "Total guests"))

fig_guests_2017
```