Homework 2
Customer Segmentation for Sun Country Airlines

Vijay R Dhulipala
Qian Fu
Yidan Gao
Arivarasu Perumal
Yi-Li Yu
Chuchen Xiong

20th October 2019

Contents

Background and The Business Problem.

The trend towards consolidation has emerged in the U.S airline industry since December 2013, and the three big players dominating the U.S market right now are American Airlines, Delta and United Airlines. Our client company is a unique player in this industry, Sun Country Airlines.

It is a homegrown airline company based in Minnesota, which offers flight services to multiple locations. It has endured lots of challenges over the past decades, like the multiple ownership changes, two bankruptcies, and fierce competition from the national brand, which heavily threaten the existence of Sun Country Airlines. However, it turned out to be profitable and stable ultimately. Now it has entered into a new period of operation, where it positioned itself as an independent airline with a small fleet, serving a select roster of destinations from its Minnesota headquarters.

To make this small regional airline set itself apart from others in a unique way, Sun Country Airlines aims to put a priority on drawing actionable customer insights and segmenting customers effectively with historical data, to improve customer experience.

Our taskHired by the analytics team of Sun Country Airlines, our motivation is to better understand their customers and address the challenges they face using historical data, to elevate customer experience and generate higher revenue. Specifically, we are going to develop a clear and robust picture of different segments of Sun Country's customers.

Our Task

Our Approach

To better drive the analysis, we narrowed down our focus into two specific questions. Firstly, we are interested in understanding if our Rewards programs are beneficially and how we can better exploit its benefits as a customer and airlines.

To address the first question, we do cluster analysis on the Non-Ufly members' data, which contains lots of useful information like total amount one person contribute, flight frequency, the number of average days one person book their flights in advance, booking channel, etc. In this way, we could furthermore explore which segment of non-Ufly members is more suitable for converting into U-Fly members.

In terms of our second question, we tried to examine those customers who do not return back to Minneapolis with Sun Country Airlines. With clustering analysis on the flights-level data, we could better know about the characteristics of non-returning passengers and thereby offer actionable insights.

Understanding Data

Data Filtering and Cleaning

Load dependencies

```
# Importing reqired libraries
library(ggplot2)
theme_set(theme_classic())

library(reshape2) ; library(ggrepel) ; library(RSQLite)
library(XML) ; library(knitr) ; library(dplyr)
library(naniar) ; library(gridExtra) ; library(magrittr)
library(lubridate) ; library(RSQLite) ; library(tidyr)
library(data.table) ; library(RColorBrewer) ; library(stringr);
library(clustMixType) ;library(cluster); library(cowplot)
library(fpc);library(dbscan);library(factoextra)
# My theme
my_theme <- theme_classic() +
    theme(text=element_text(size=9,  family="serif"))
```

Reading Data

```
suncountry <- read.csv('~/Desktop/SunCountry.csv')
```

Data Glimpse

```
# Look at the data types and head values in every column
glimpse(suncountry)
```

```
## Observations: 3,435,388
## Variables: 26
## $ PNRLocatorID         <fct> AAABJK, AAABJK, AAABMK, AAABMK, AAABTP, A...
## $ TicketNum            <dbl> 3.377365e+12, 3.377365e+12, 3.372107e+12,...
## $ CouponSeqNbr         <int> 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1,...
## $ ServiceStartCity     <fct> JFK, MSP, MSP, SFO, MCO, PSP, JFK, JFK, M...
## $ ServiceEndCity       <fct> MSP, JFK, SFO, MSP, MSP, MSP, MSP, MSP, M...
## $ PNRCreateDate        <fct> 2013-11-23, 2013-11-23, 2014-02-04, 2014-...
## $ ServiceStartDate     <fct> 2013-12-13, 2013-12-08, 2014-02-23, 2014-...
## $ PaxName              <fct> BRUMSA, BRUMSA, EILDRY, EILDRY, SKELMA, H...
## $ EncryptedName        <fct> 4252554D4241434B44696420493F7C20676574207...
## $ GenderCode           <fct> F, F, M, M, F, M, F, M, M, M, F, M, F, F,...
## $ birthdateid          <int> 35331, 35331, 46161, 46161, 34377, 39505,...
## $ Age                  <int> 66, 66, 37, 37, 69, 54, 25, 69, 49, 58, 2...
## $ PostalCode           <fct> , , , , , , , , , 55116, , , 55126, 55126...
## $ BkdClassOfService    <fct> Coach, Coach, Coach, Coach, Coach, Coach,...
## $ TrvldClassOfService  <fct> Coach, First Class, Discount First Class,...
## $ BookingChannel       <fct> Outside Booking, Outside Booking, SCA Web...
## $ BaseFareAmt          <dbl> 234.20, 234.20, 293.96, 293.96, 112.56, 1...
## $ TotalDocAmt          <dbl> 0.0, 0.0, 338.0, 338.0, 132.0, 194.8, 191...
## $ UFlyRewardsNumber    <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, 20236...
## $ UflyMemberStatus     <fct> , , , , , , , , , Standard, , , Standard,...
## $ CardHolder           <fct> , , , , , , , , , false, , , false, false...
## $ BookedProduct        <fct> CHEOPQ, CHEOPQ, , , , , SSWMIR, SSWMIR, S...
## $ EnrollDate           <fct> , , , , , , , , , 2010-03-04 10:50:43.000...
## $ MarketingFlightNbr   <fct> 244, 243, 397, 392, 342, 610, 244, 244, 3...
## $ MarketingAirlineCode <fct> SY, SY, SY, SY, SY, SY, SY, SY, SY, SY, S...
## $ StopoverCode         <fct> O, , O, , , , , , , , , , O, , O, , , O, ...
```

What does the data look like? • 3M+ records containing within a span of 2 years of historical flight bookings • 1.1M unique booking PNRs • Demographic details include EncryptedName, Gender, Age, PostalCode • Details about Ufly Member status, TravelClass, BookingChannel

Filtering the data for Sun Country

```
table(suncountry$MarketingAirlineCode)
```

```
##
##      DE      F9      FI      HA      SY
##       1    3319      13    1705 3430350
```

```
Sun <- suncountry %>% filter(MarketingAirlineCode == "SY")
```

We observe that all the non-members have blank values in the status column. We are considering flyers to be non-members if they have member status as blank or unidentified values.

```
table(Sun$UflyMemberStatus)
```

```
##
##              Elite Standard
##   2735876    14361   680113
```

```
Sun <- Sun %>%
  mutate(UflyMemberStatus = ifelse(UflyMemberStatus == "", "Non Ufly Member",
                                   levels(UflyMemberStatus)[UflyMemberStatus]))
```

Since passengers might book tickets from various such as airports, SCA Website Booking, Outside Booking, etc…We observe that the five prominent channels are Outside Booking, SCA Website Booking, Tour Operator Portal, Reservations Booking and SY Vacation. Hence, we marked all the other channels are 'Others'.

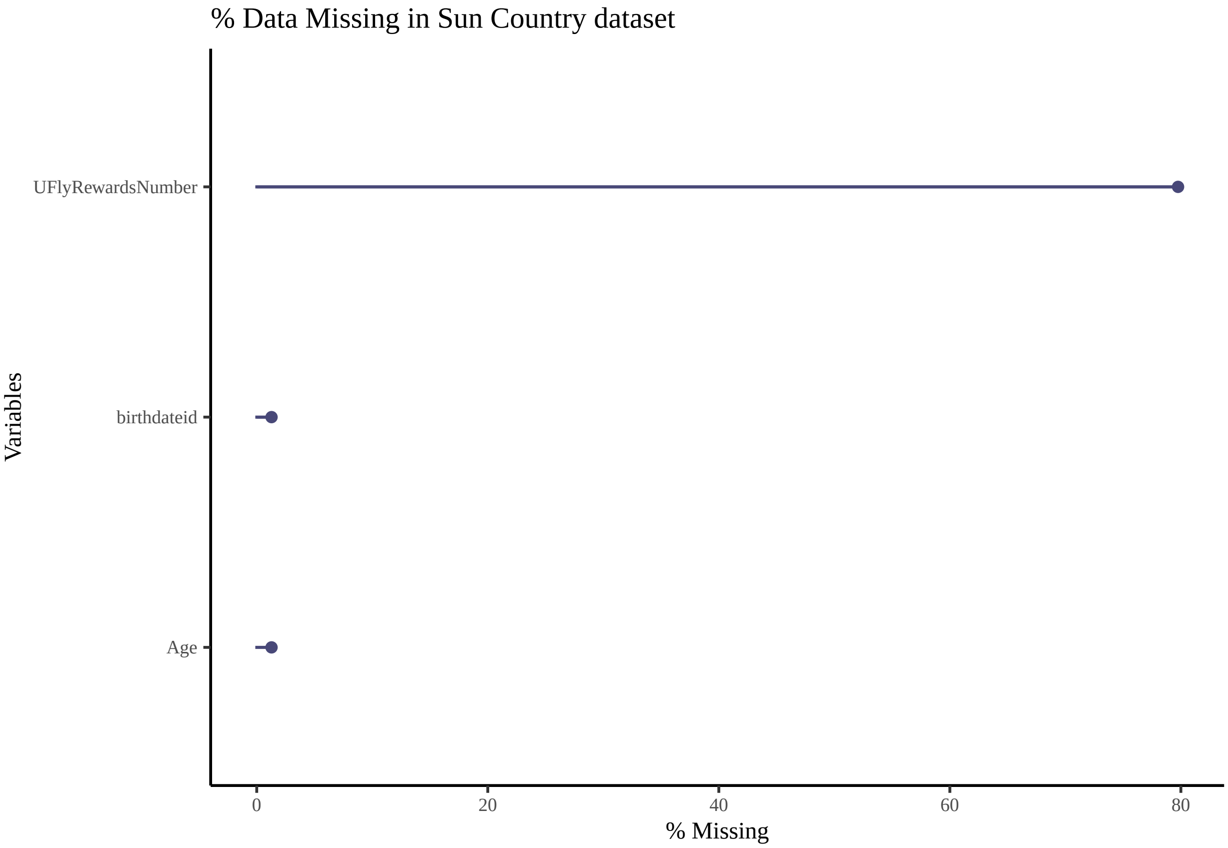
```
# Converting all the other booking channel to 'Other'
Sun <- Sun %>%
  mutate(BookingChannel = ifelse(BookingChannel == "Outside Booking" |
                                   BookingChannel == "SCA Website Booking" |
                                   BookingChannel == "Tour Operator Portal" |
                                   BookingChannel == "Reservations Booking"|
                                   BookingChannel == "SY Vacation",
                                 levels(BookingChannel)[BookingChannel],'Others'))

table(Sun$BookingChannel,useNA = "always")
```

```
##
##                Others      Outside Booking Reservations Booking
##                 11915              1471179               262327
##   SCA Website Booking          SY Vacation Tour Operator Portal
##               1457127                94601               133201
##                  <NA>
##                     0
```

```r
# Converting the column to factor for analysis
Sun$BookingChannel<-as.factor(Sun$BookingChannel)
```

Exploring data for erroneous and missing values

```r
gg_miss_var((Sun)[colSums(is.na(Sun)) > 0], show_pct = TRUE) +
  labs(title = "% Data Missing in Sun Country dataset") + my_theme
```



We observed that most of the UFly Reward Numbers are missing. This is due to the fact that these travellers are Non-UFly members. Hence, we are making an assumption that these missing data won't affect our analysis is valid.

```r
table(Sun$GenderCode)
```

```
##
##                F       M       U
##   43975 1767909 1618426      40
```

```r
# Identifying blank values for genders and removing them.
Sun <- Sun %>% filter(GenderCode %in% c("F", "M", "U"))
```

There are negative values in the id, but we are not going to remove it as they are just id's.

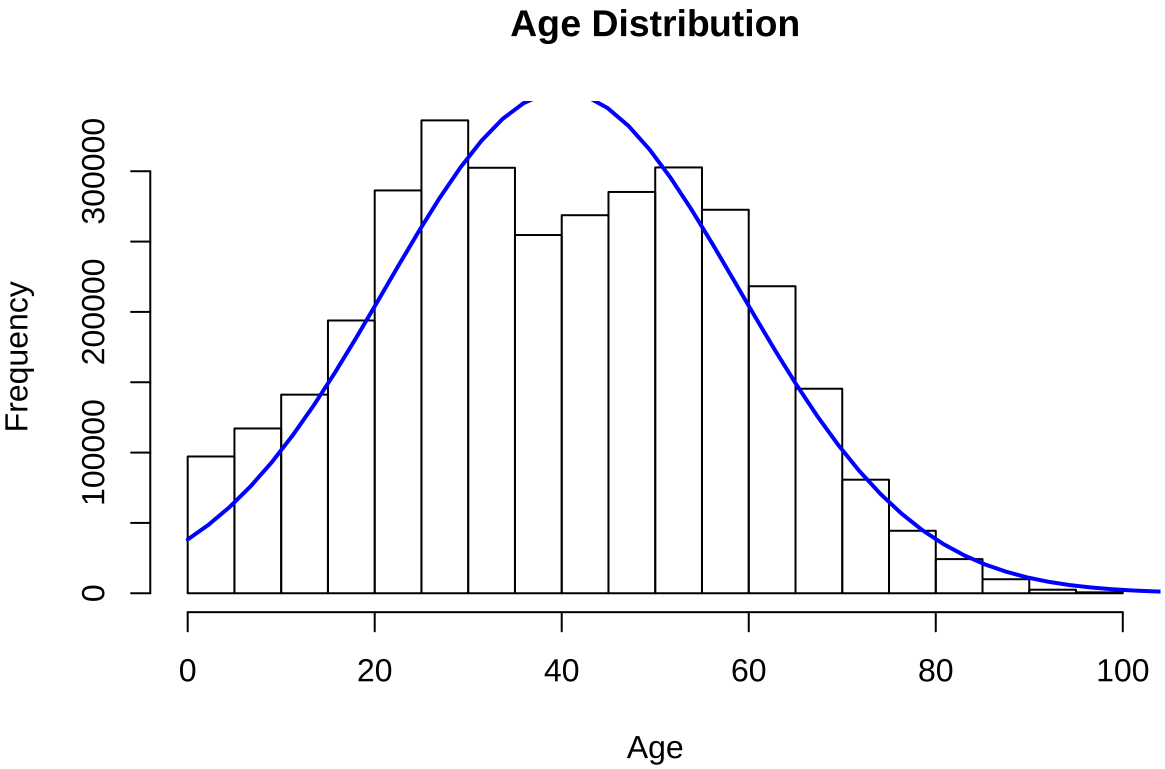
```
# birthdateid column
summary(Sun$birthdateid)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -675290   39618   45085   44981   50250 1112840
```

```
# Age Column
summary(Sun$Age)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -2883.00    26.00    40.00    40.05    55.00  2012.00
```

```
# We are removing rows with error values and select only age between [0,100]
Sun <- Sun[!(Sun$Age < 0 | Sun$Age > 100),]
# Distribution after cleaning the Age column
x <- Sun$Age
# Plotting a normal distribution curve for age
h <-hist(x, breaks=20, xlab="Age", main="Age Distribution")
xfit<-seq(0,110,length = 50)
yfit<-dnorm(xfit,mean=mean(x, na.rm = T),sd=sd(x, na.rm = T))
yfit <- yfit*diff(h$mids[1:2])*length(x)
lines(xfit, yfit, col="blue", lwd=2) + my_theme
```



```
## NULL
```

From our assumption, that we can consider ages between 0-100 was validated by plotting the histogram. The distribution of the ages still remains the same. Further, the data loss from this is insignificant. This assumption also removes all the missing values from age and birth-id column. Finally, from this article (https://nciph.sph.unc.edu/focus/vol3/issue5/3-5DataBasics_issue.pdf) is further corroborated that the assumption is valid/

We want to understand if there is any pattern with the CouponSeqNbr and PNR. We observed that as the CouponSeqNbr increases the count decreases. We wanted to see if all PNR's are not starting with a couponSeqNo of 1.

```
# Coupon Sequence Number column
table(Sun$CouponSeqNbr)
```

```
##
##       1       2       3       4       5       6       7       8
## 1926359 1385857   42593   29199     592     120       6       1
```

```
Sun_temp<- Sun %>%
  select(PNRLocatorID, CouponSeqNbr) %>%
  group_by(PNRLocatorID) %>%
  summarise(min_CouponSeqNbr = min(CouponSeqNbr)) %>%
  filter(min_CouponSeqNbr > 1)

head(Sun_temp)
```

```
## # A tibble: 6 x 2
##   PNRLocatorID min_CouponSeqNbr
##   <fct>                   <int>
## 1 AAAWGV                      3
## 2 AADOFC                      2
## 3 AAJIBZ                      2
## 4 AAJUAM                      2
## 5 AAKOYN                      2
## 6 AALHSE                      2
```

```
# We can observe that the min_CouponSeqNbr is greater than 1. Hence, we removed journeys that has to st

Sun <- Sun %>% group_by(PNRLocatorID) %>%
  mutate(flag = ifelse(min(CouponSeqNbr) != 1, 1, 0)) %>% filter(flag == 0)
```

Data Transformation

After successfully cleaning and capturing the errors in our data, we are interested in creating two data set. First, the existing one which will help us understand data from flight or path view. Second, a customer level data set where each transaction is a customer. This is challenging, as, from our observation, we have multiple rows in a single PNR for each leg of the trip for each customer. Aggregating the data to bring it to a level where we have one row for each customer without losing much of the other information by generating new columns and using aggregation.

The size of the dataset is very large. So we wanted to take a sample and run all the data transformation steps on a smaller sample consisting of data related to 20,000 data points. Further, we used the Kolmogorov-Smirnov test and chi-sq test to understand if all our columns are similar to the parent dataset. We observed a P-Value of 0.78 which implies that we can reject the null hypothesis that both data sets are independent. Hence, we can observe a close relation or replication of our true dataset. (commenting the code due to tedious calculation)

```r
sample_size = 20000
set.seed(1)
idxs = sample(1:nrow(Sun),sample_size,replace=F)
subsample = Sun[idxs,]
# pvalues = list()
#
# for (col in names(Sun)) {
#   if (class(Sun[,col]) %in% c("numeric","integer")) {
#     # Numeric variable. Using Kolmogorov-Smirnov test
#     pvalues[[col]] = ks.test(subsample[[col]],Sun[[col]])$p.value
#
#   } else {
#     # Categorical variable. Using Pearson's Chi-square test
#     probs = table(Sun[[col]])/nrow(Sun)
#     pvalues[[col]] = chisq.test(table(subsample[[col]]),p=probs)$p.value
#
#   }
# }
```

Transformation to populate the First City for booking
In order to aggregate multiple rows for each PNR, we need a unique id or a primary key. We Assume that a combination of four columns “PaxName”, “EncryptedName”, “GenderCode”, “birthdateid” will be able to generate a unique id needed for aggregation. Based on the unique id we group the data and pick the ServiceStartCity in the first row of the group as the First_City of the journey.

```r
# As we can observe that there are identicial row.
# We generated a UID as a primary key
subsample <- subsample %>%
  mutate(uid=paste(PaxName,EncryptedName,GenderCode,birthdateid,sep=""))

StartCity <- subsample %>%
  arrange(PNRLocatorID, CouponSeqNbr) %>%
  group_by(PNRLocatorID, PaxName) %>%
  do(data.frame(First_City = first(.$ServiceStartCity)))

# Merged the selection with out oruginal table.
subsample <-merge(subsample,StartCity,
by.x=c("PNRLocatorID","PaxName"),
by.y = c("PNRLocatorID","PaxName"))
```

Transformation to populate the Last City for booking
Based on the unique id we group the data and pick the ServiceEndCity in the last row of the group as the Last_City of the journey.

```r
EndCity <-subsample %>%
  arrange(PNRLocatorID,CouponSeqNbr)%>%
  group_by(PNRLocatorID,PaxName)%>%
  do(data.frame(Last_City=last(.$ServiceEndCity)))

# Merged the selection with out oruginal table.
subsample <-merge(subsample,EndCity,
by.x=c("PNRLocatorID","PaxName"),
by.y = c("PNRLocatorID","PaxName"))
```

Transformation to find if customers stay at a destination in a round trip
However, this approach will miss out on the stay and the final destination. In order to capture the final destination and stay informed, we need to consider both the legs of the journey and combine them. Further, we are looking at the time difference between all the legs of the journey and assume that the one with the maximum difference and pick the ServiceEndCity of that leg as the Final_Destination.

```
subsample$ServiceStartDate <- as.Date(subsample$ServiceStartDate)

# The place of maximum stay during the trip.
Max_Stay <- subsample%>%
  arrange(PNRLocatorID, CouponSeqNbr) %>%
  group_by(PNRLocatorID, PaxName) %>%
  mutate(stay = lead(ServiceStartDate)-ServiceStartDate, default=0) %>%
  select(PNRLocatorID, PaxName, ServiceStartCity, ServiceEndCity, ServiceStartDate, stay)

# Filling zero for direct flights
Max_Stay$stay[is.na(Max_Stay$stay)] <- 0
Max_Stay$stay <- as.numeric(Max_Stay$stay)

# Merge the new column with the data
Final_Destination <- Max_Stay %>%
  group_by(PNRLocatorID, PaxName) %>%
  do(data.frame(Final_Destination =
                first(as.character(.$ServiceEndCity)[.$stay==max(.$stay)])))

subsample <- merge(subsample, Final_Destination,
                   by.x=c("PNRLocatorID","PaxName"),
                   by.y = c("PNRLocatorID","PaxName"))
```

Transformation for round trip and group size
To capture the group size and round trip travel. We created a boolean flag and matched if the start and end city are the same. Further, we calculated the group size using the unique id and PNR we calculated the count of the group.

```
subsample <- subsample%>%
  mutate(round_trip = ifelse(as.character(First_City)==as.character(Last_City), 1, 0))

# We look at the group size, number of people who traveled together in each trip.
subsample <- subsample %>%
group_by(PNRLocatorID) %>%
mutate(group_size= length(unique(uid)))

# Create a flag for group
subsample <- subsample %>%
  group_by(PNRLocatorID)%>%
  mutate(group = ifelse(group_size > 1, 1, 0))
```

Transformation to calculate the number of days in advance the booking was made and also look for seasonality
In order to differentiate travellers based on the pre-book dates, we calculated the using service start date and PNR generated to date.

```
subsample$ServiceStartDate<-as.Date(subsample$ServiceStartDate)
```

```r
# Convert ServiceStartDate from factor to Date format
subsample<- subsample %>%
  group_by(PNRLocatorID, PaxName) %>% mutate(month_no = month(ServiceStartDate))

# We look at the number of days the ticket was booked in advance.
subsample$PNRCreateDate <- as.Date(subsample$PNRCreateDate)
subsample$ServiceStartDate <- as.Date(subsample$ServiceStartDate)
subsample <- subsample %>%
  mutate(days_pre_booked = as.numeric(floor(difftime(ServiceStartDate,
                                                     PNRCreateDate,units=c("days")))))
```

Selecting Columns of Interest for clustering

```r
# We transformed the data such that each row represents a unique customer-PNR combination.
subsample <- subsample %>%
  select(PNRLocatorID, uid, PaxName, ServiceStartDate, BookingChannel, Age,
       UFlyRewardsNumber,UflyMemberStatus,First_City, Last_City,Final_Destination,
       round_trip,group_size,group, month_no ,days_pre_booked)

data_transformed <- subsample %>%
  group_by(PNRLocatorID, uid, PaxName) %>%
  summarise(ServiceStartDate = first(ServiceStartDate),
            BookingChannel = first(BookingChannel),
            UFlyRewards = first(UFlyRewardsNumber),
            UflyMemberStatus = first(UflyMemberStatus),
            Age = max(Age),
            First_City = first(First_City),
            Last_City = last(Last_City),
            Final_Destination = first(Final_Destination),
            round_trip = first(round_trip),
            group_size = first(group_size),
            group = first(group),
            month_no = last(month_no),
            days_pre_booked = max(days_pre_booked))

# Retaining only those attributes that are meaningful for clustering
data_transformed <- data_transformed %>%
  select(-PNRLocatorID, -uid, -PaxName, -ServiceStartDate, -UFlyRewards)
```

```r
normalize <- function(x){return ((x - min(x))/(max(x) - min(x)))}

temp <- ungroup(data_transformed)
customer_data_clust = mutate(temp,
                             Age = normalize(Age),
                             days_pre_booked = normalize(days_pre_booked),
                             group_size=normalize(group_size))
```

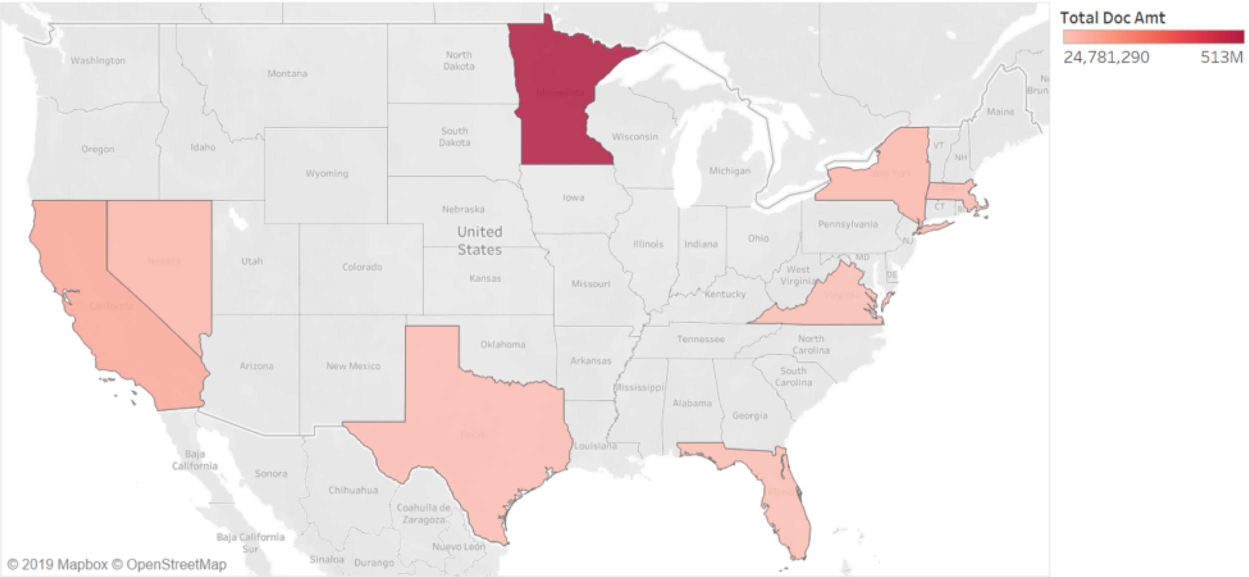
An Overview Sun Country

Where do our flyers tarvel?

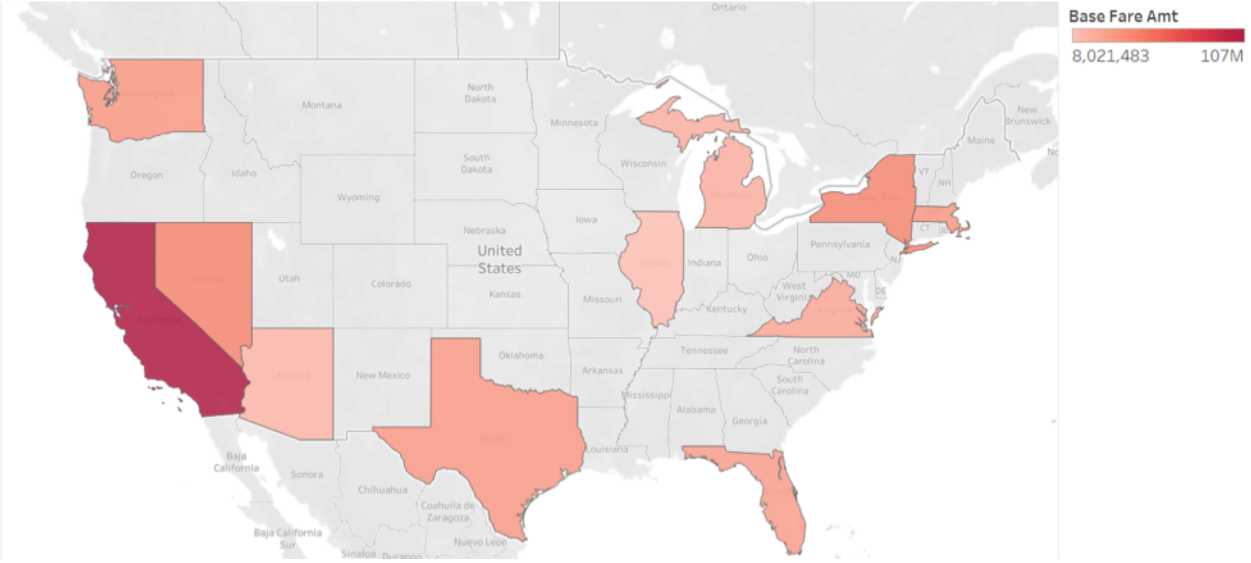
Description and Rationale for the Chosen Analysis
As a homegrown airline company based in Minnesota, we wanted to do an extensive study on the air routes where we operate and to find the routes in which our customers frequently travel. Hence we started with analysing, from where most of our customers depart and what are the most frequent destinations.

Execution and Results (without code)
[Graphs generated from Tableau]

```r
knitr::include_graphics('./figures/Most Frequeny Start.png')
```



```r
knitr::include_graphics('./figures/Most Frequeny End.png')
```



Interpretation

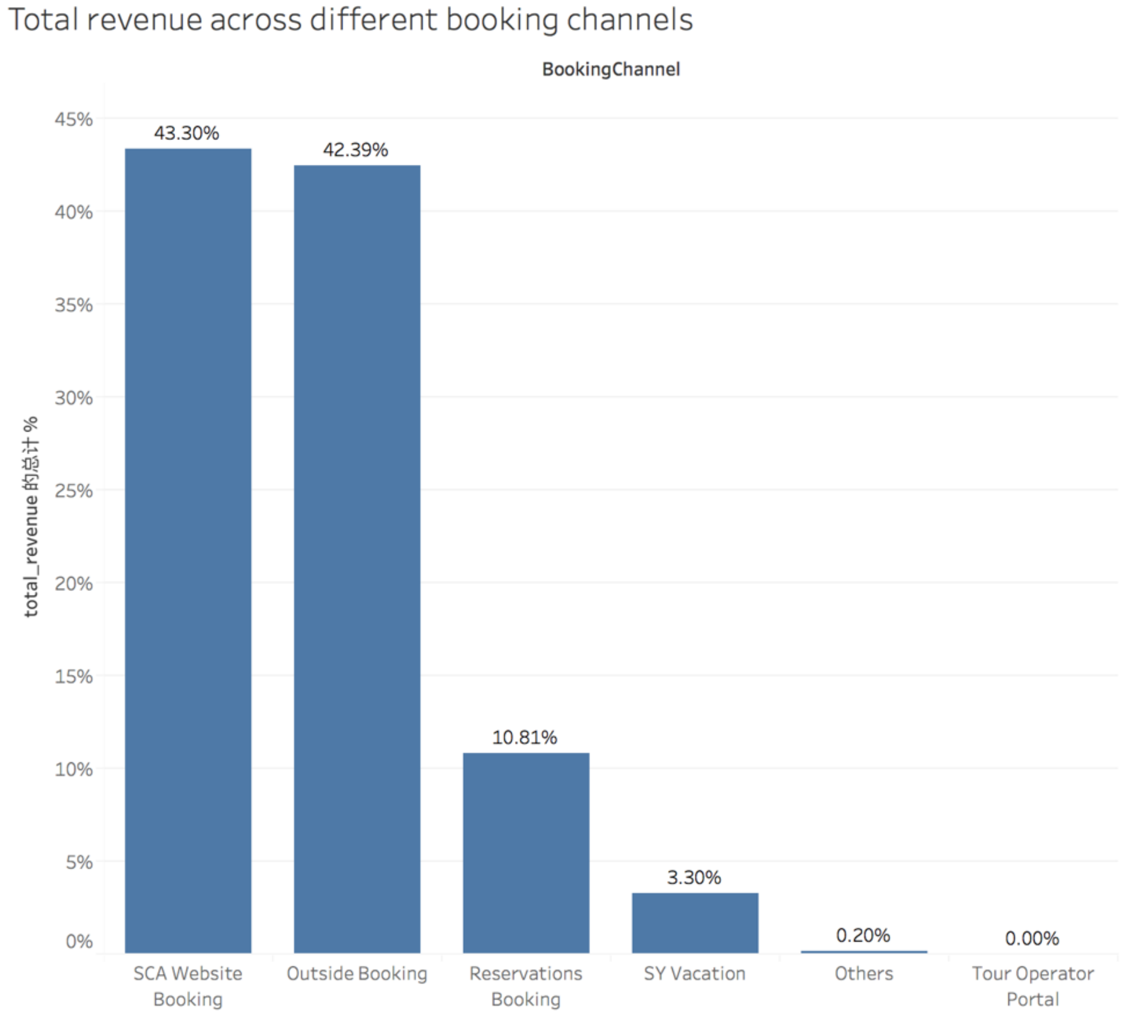
Above graphs clearly shows that most of our customers depart from Minnesota which matches the fact that Sun Country is headquartered at Minnesota and from the second graph, it is clearly visible that the frequent destination where the customers travel is to the state of California.

Where do our flyers book tickets?

Description and Rationale for the Chosen Analysis
In general, the flight tickets are booked through multiple platforms which gives the customers a varied option to book the tickets. It is the prime duty of the company to provide customer satisfaction on this aspect. So we wanted to study how our customers are booking their tickets and come up with the business strategies to improve our options for the customers to book the flight tickets.

Execution and Results (without code)

```r
knitr::include_graphics('./figures/BookingCha.png')
```



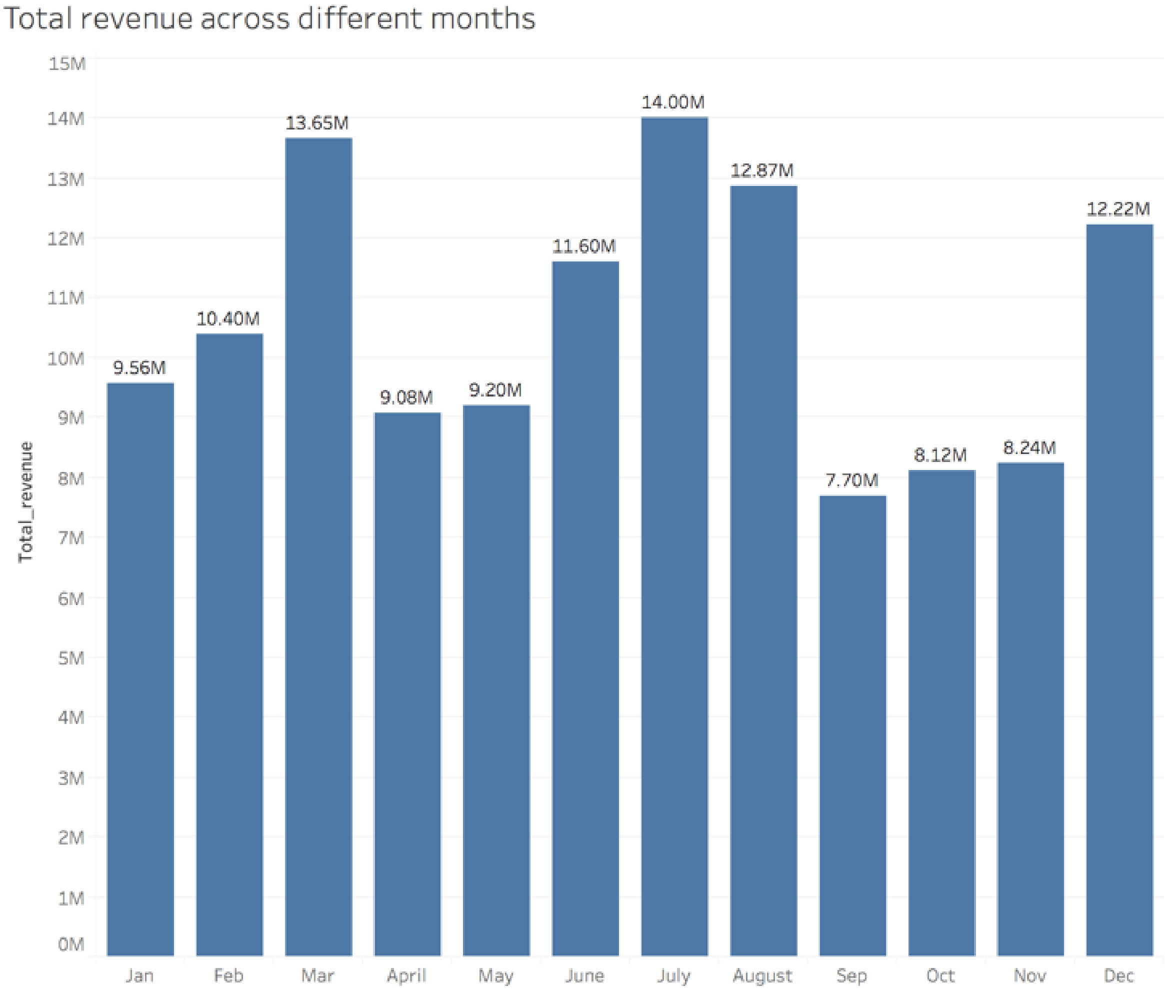
Interpretation
SCA website booking contributes to 43.30 % of the total revenue and following that, the next highest contributor is 'outside booking' with a contribution of 42.39 % of the total, both constitute to a total of 85.7% of total revenue. There is a very less percentage of people who books the ticket in person. We also can see that 'SY Vacation' contributes to only 3.30 % to the total revenue

When do our flyers travel?

Description and Rationale for the Chosen Analysis
It is important to understand the pattern of ticket sales with time. It will give us when our customers travelling and where are they travelling. This analysis can help us to sort out the reason behind why our airlines are not performing well at certain periods of the year and can explain the reason behind the pattern observed in other analysis like where are most of our customers travelling.

Execution and Results (without code)

```
knitr::include_graphics('./figures/RevenueMonths.png')
```



Interpretation
With the graph, we can see that revenue for the company during the months of December to March and June to August is very high when compared to the low revenue gained during the months of September to November of the year.

Summary
We saw earlier that most of our customers travel from Minnesota to other places of the country and 'Revenue vs Month of Year' graph reassures the same by getting most of the revenue in December to march (Winter) and in June to August (Summer). As winter is very harsh and summer is very pleasant in Minnesota, most customers are travelling out of Minnesota in winters and many customers travelling towards Minnesota during summers. This can be interpreted as most of the people who are travelling to and from Minnesota

are for vacation purposes. Hence we can use this analysis as a basis and try to concentrate these customers for the vacation packages.

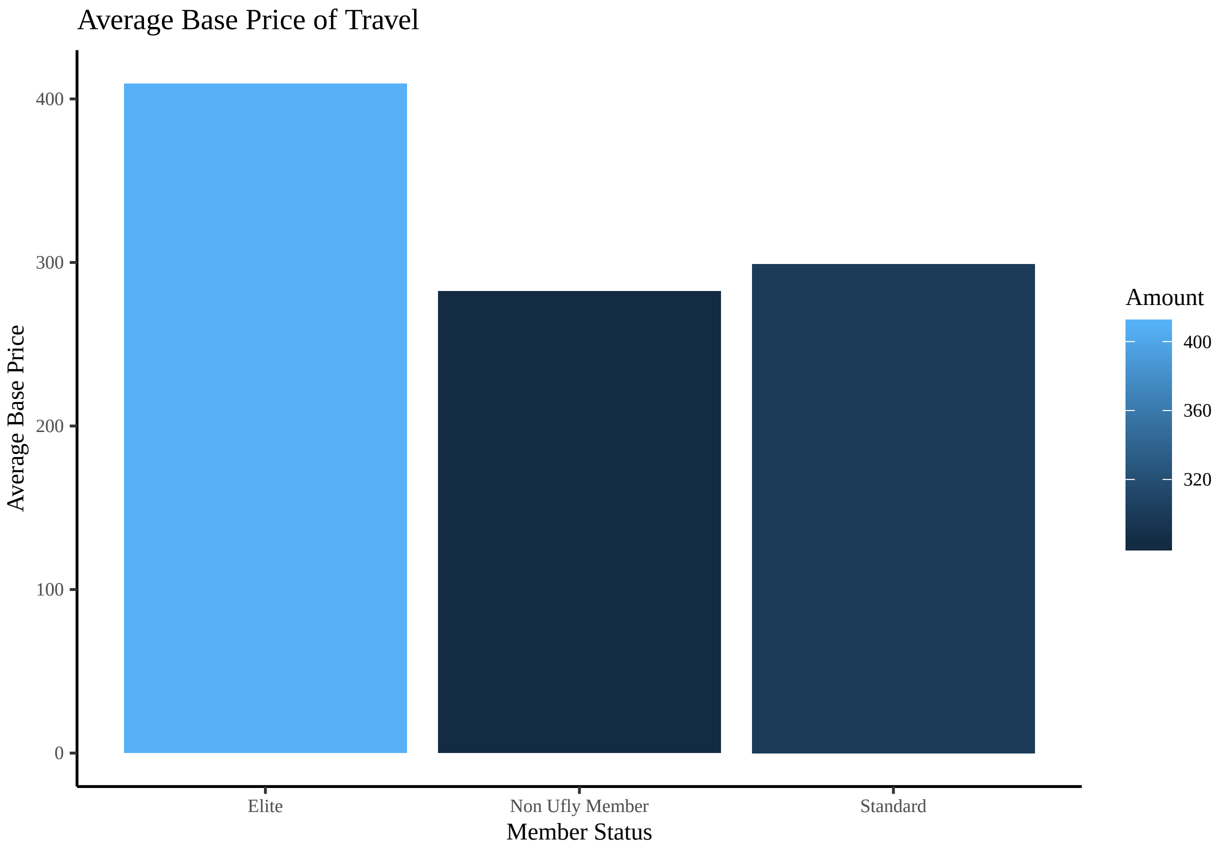
An Overview of UFly and Non UFly Members

Are UFly Mmebers more profitable?

Description and Rationale for the Chosen Analysis
Sun County's UFly Rewards is a points-based program where members earn 10 points for a Sun Country Airlines round trip coach flight; 14 points for a roundtrip first-class flight. Ufly members can redeem 100 points for a roundtrip coach ticket for travel within the 48 contiguous states. This offer in which Sun Country helps travels to maintain an alliance with the airlines and establish a bond that will elevate the customer experience. Hence, it is necessary to understand if these travelers are generating more revenue when compared to non-travelers

Execution and Results (with code) In order to understand the difference, we calculated the mean base price of different members.

```
Mem_vs_non <- Sun %>%
  group_by(UflyMemberStatus) %>%
  summarise(Amount = mean(BaseFareAmt),
            count = n())

ggplot(Mem_vs_non,aes(x = UflyMemberStatus,y = Amount)) +
    geom_bar(aes(fill = Amount),stat = "identity",position = "dodge")+
  labs(title = "Average Base Price of Travel", x = 'Member Status', y = 'Average Base Price') + my_theme
```



Interpretation
From this analysis, we came to an understanding that both our Elite and Standard members spend more money when compared to our Non-UFly members. As from our client's information, UFly Rewards members get a free checked bag, priority check-in line, and other additional benefits. Elite members get two free bags for themselves and anyone else on their reservation.

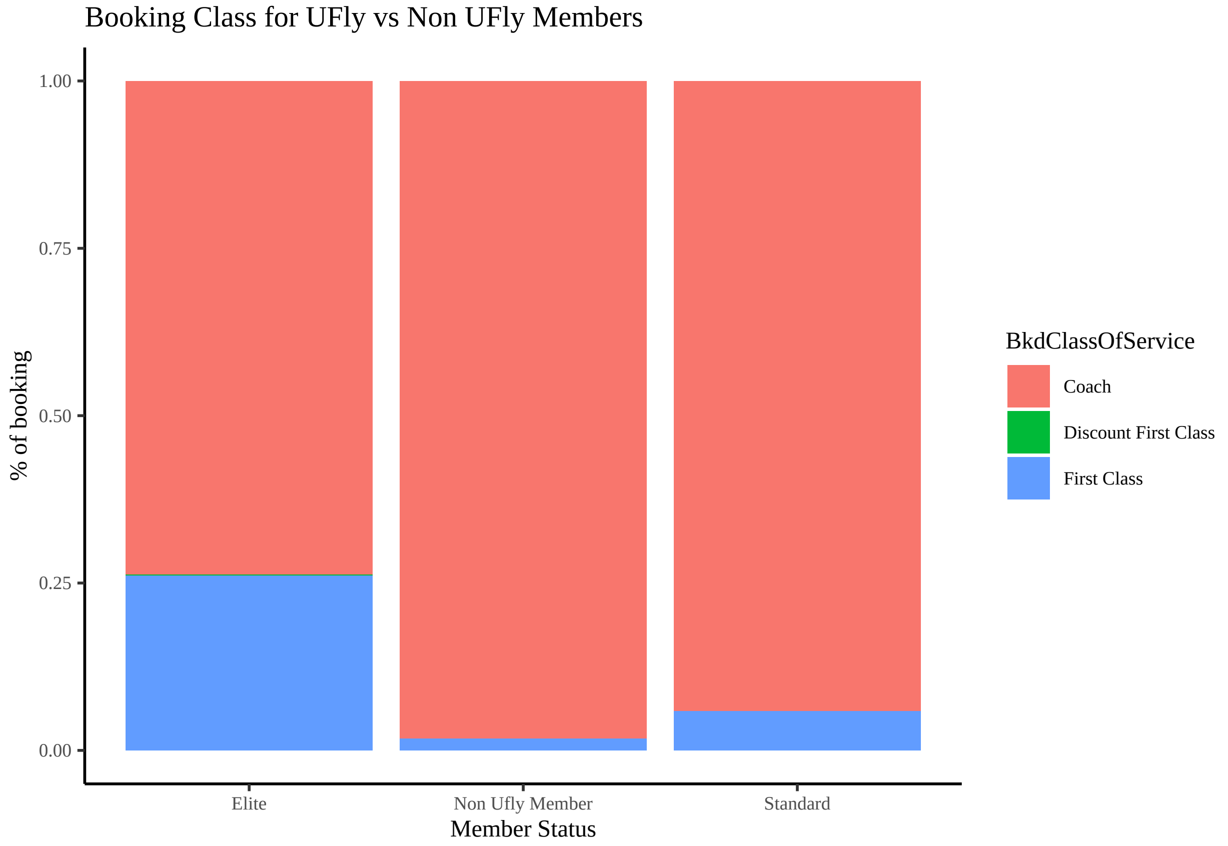
With these additional benefits, we can confidently say that our members try to travel in better luxury.

Do UFly members book same class as Non UFly Members?

Description and Rationale for the Chosen Analysis
Sun Country has two types of seating which are provided for travelers, First Class, and Coach. One way to understand if our rewards programs are working is if our members are traveling in better classes. This might be due to the fact that Sun Country provides miles points for every travel which in return can be used to upgrade seating. Hence, if more travelers are traveling through First Class, it means they have high points (due to many travels) or might have spent more money. Either way is a win-win situation.

Execution and Results (with code)

```
Mem_vs_non <- Sun %>%
  group_by(UflyMemberStatus,BkdClassOfService) %>%
  summarise(count = n())

ggplot(Mem_vs_non,aes(x = UflyMemberStatus,y = count)) +
    geom_bar(aes(fill = BkdClassOfService),stat = "identity",position = "fill")+
  labs(title = "Booking Class for UFly vs Non UFly Members", x = 'Member Status', y = '% of booking') +
```



Interpretation
From the results we can see, most of our First Class travelers are from our member program. This implies that our rewards program members are frequent luxury travelers when compared to Non-UFly members.

Do our members book our ticket before than our Non-Members?

Description and Rationale for the Chosen Analysis
To understand the loyalty of our members, we can analyze if they book Sun Country tickets beforehand. As we know that the Airline industry is a highly competitive and price volatile industry. It is difficult to generate loyal customers. Hence with this measure, we have a slight understanding if our travellers are preferring Sun Country.

Execution and Results (with code)

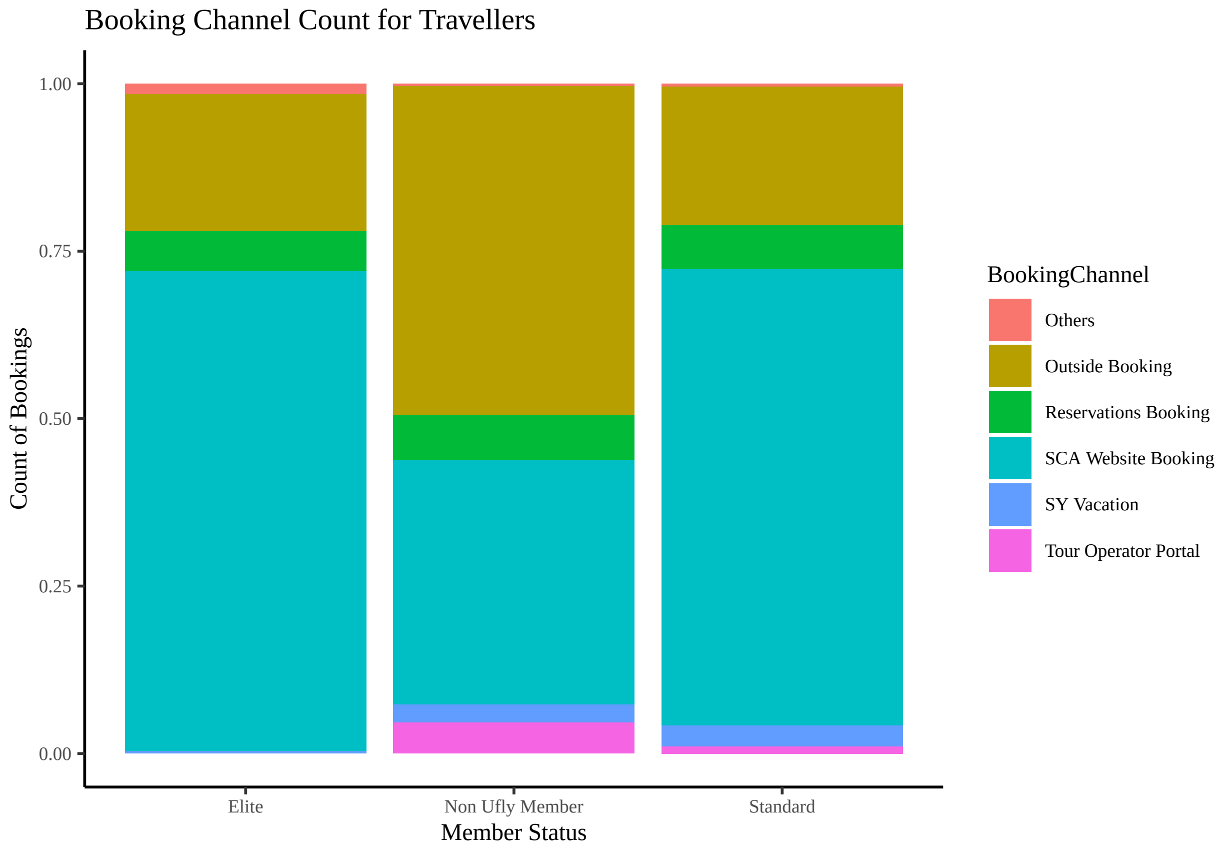
```
# Mem_vs_non <- Sun %>%
#   mutate(diff = difftime(ServiceStartDate, PNRCreateDate, units = "days")) %>%
#   group_by(UflyMemberStatus) %>%
#   summarise(Booking = mean(diff))
#
#
# ggplot(Mem_vs_non,aes(x = UflyMemberStatus,y = Booking)) +
#     geom_bar(aes(fill = Booking),stat = "identity",position = "dodge")+
#   labs(title = "Avg Days Pre Booked for Members and Non Members", x = 'Member Status', y = 'Avg. Days
```

Interpretation
We can observe that our UFly Members trends to book Sun Country tickets much later than Non-UFly members. Non-members are unstable with their bookings due to price wars between airlines. However, our members tend to book much later with Sun Country.

Where do flyers frequently book tickets?

Description and Rationale for the Chosen Analysis
To understand the our members, we can analyze if they book though Sun Country website or not. As we know that the Airline industry is a highly competitive and price volatile industry. It is difficult to generate loyal customers. Hence with this measure, we have a slight understanding if our travellers are preferring Sun Country.

Execution and Results (with code)

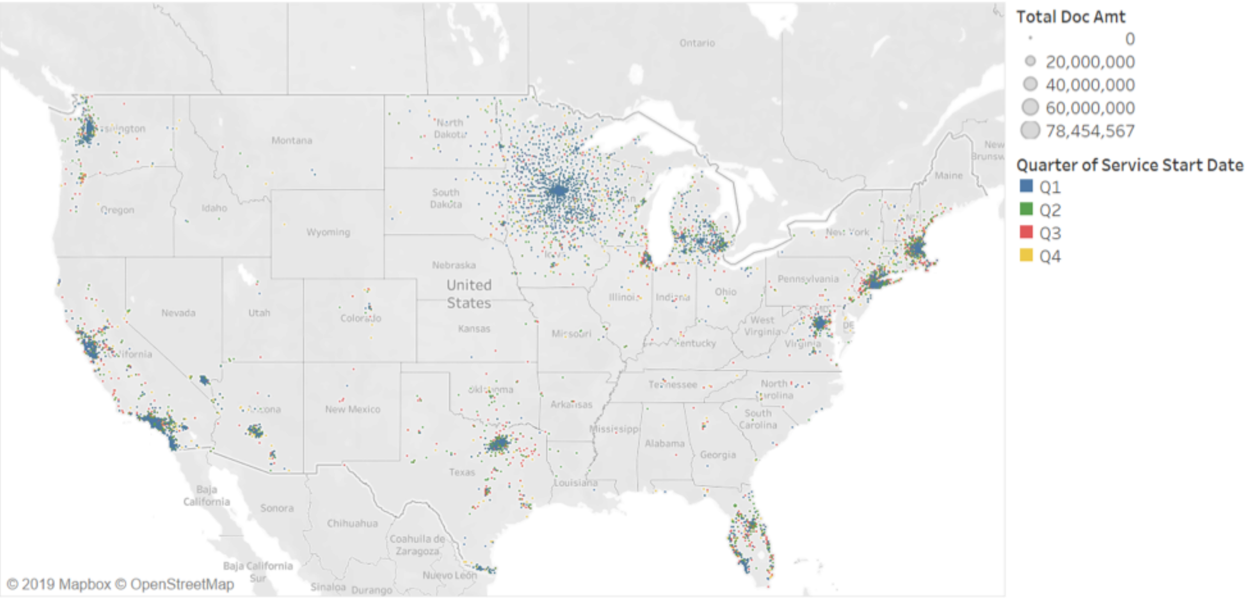


Where do our members tarvel and when do they travel?

Description and Rationale for the Chosen Analysis
Finally, To have a granular look about our members, we are interested in understanding if our members when and when our members travel. As we have an understand that most of our origination flight are from MSP and we have various destitution especially to warmer climates. (See an overview of Sun Country Analysis)

Execution and Results (withouth code)
[Graphs generated from Tableau]

```
knitr::include_graphics('./figures/UFly Travel.png')
```



Interpretation
From the map, we can understand that most of our members are from MSP and traveling to the east and west coast during the first quarter i.e. in the month of January to March.

Summary

From this analysis, we can clearly establish the fact that our UFly Members are more engaged and yield higher revenue when compared to Non-UFly members. • They generate higher revenue per person • They are slightly less insensitive to price as they book much later than Non-Members • Our First Class seats are mostly booked or upgraded to member travelers • They have faith in Sun County and tend to book tickets using our website when compared to other websites. • They show a similar pattern like overall travelers i.e. they highly originate from MSP.

Clustering to segment UFly and Non-UFly Members

Hierarchical clustering and Dendogram

Description and Rationale for the Chosen Analysis
To understand and observe any patterns in our data we are interested to find any patterns in the data. To cluster our data we first need to understand what is the optimal number of clusters. As a first step, we are using Hierarchical to understand this.

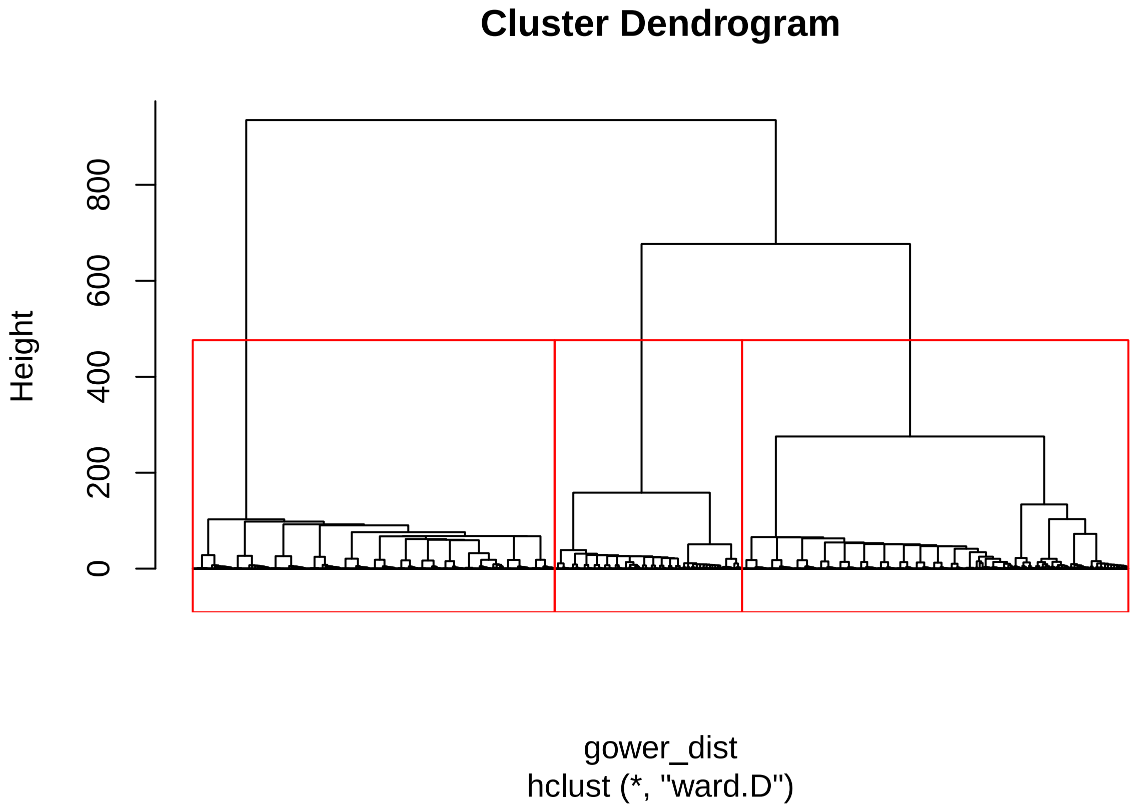
Execution and Results (with code)

```r
# Calculate distance - daizy
customer_d <- customer_data_clust[,c(3:5,8:13)]
customer_d$BookingChannel<-as.factor(customer_d$BookingChannel)
customer_d$UflyMemberStatus<-as.factor(customer_d$UflyMemberStatus)
customer_d$month_no<-as.factor(customer_d$month_no)
customer_d$Final_Destination <- as.factor(customer_d$Final_Destination)

gower_dist <- daisy(customer_d,
                    metric = "gower",
                    type = list(logratio = 3))
```

```
h_cluster <- hclust(gower_dist, method = "ward.D")

# Plotting the h_cluster
plot(h_cluster, hang = 0, label = F, main = "Cluster Dendrogram") + my_theme
```

```
## NULL
```

```
groups<-cutree(h_cluster,k=3)
rect.hclust(h_cluster, k = 3, border = "red")
```



Interpretation
From the above cluster analysis we can figure out that an optimum number of cluster would be three.

K-Prototypes Clustering

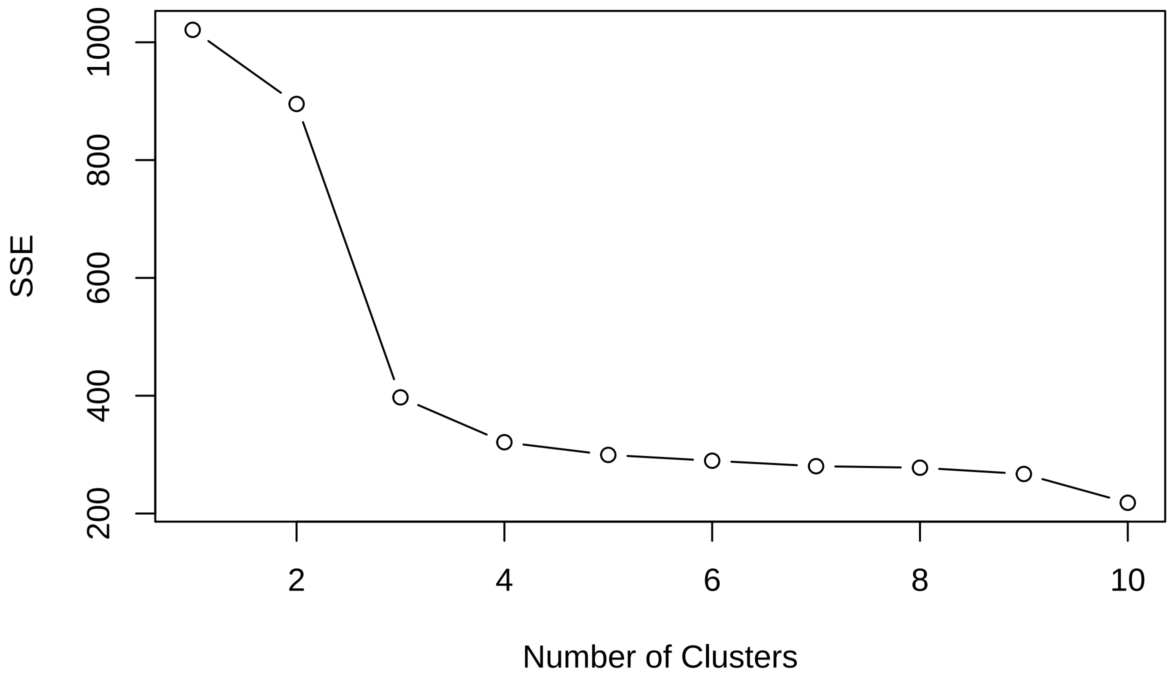
Description and Rationale for the Chosen Analysis
As our data have mixed data type we want to use k-prototypes to find an optimum number of cluster. In order to do this, we will plot the sum of squared errors curve to try and find the optimal number of clusters. Then, we will loop through the clustering several times to try and get a sense of generalized performance, as k-prototypes chooses random starting points for the clustering.

```
prototype_data <- customer_data_clust %>%
  select(First_City, Final_Destination, round_trip,
         group_size, group, days_pre_booked,
         BookingChannel, UflyMemberStatus) %>%
  as.data.frame()
```

```r
SSE_curve <- c()

for (i in 1:10){
  sse_avg <- c()
  k = i

for (t in 1:5){
  kpro <- kproto(prototype_data, k)
  sse <- sum(kpro$withinss)
  sse_avg[k] <- sse}
  SSE_curve[i] <- mean(sse_avg, na.rm = T)}
```

```
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size
##                 0                 0                 0                 0
##             group   days_pre_booked    BookingChannel  UflyMemberStatus
##                 0                 0                 0                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.01325227
##
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size
##                 0                 0                 0                 0
##             group   days_pre_booked    BookingChannel  UflyMemberStatus
##                 0                 0                 0                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.01325227
##
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size
##                 0                 0                 0                 0
##             group   days_pre_booked    BookingChannel  UflyMemberStatus
##                 0                 0                 0                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.01325227
##
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size
##                 0                 0                 0                 0
##             group   days_pre_booked    BookingChannel  UflyMemberStatus
##                 0                 0                 0                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.01325227
##
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size
##                 0                 0                 0                 0
##             group   days_pre_booked    BookingChannel  UflyMemberStatus
##                 0                 0                 0                 0
## 0 observation(s) with NAs.
## 
## Estimated lambda: 0.01325227 
## 
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size 
##                 0                 0                 0                 0 
##             group   days_pre_booked    BookingChannel  UflyMemberStatus 
##                 0                 0                 0                 0 
## 0 observation(s) with NAs.
## 
## Estimated lambda: 0.01325227 
## 
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size 
##                 0                 0                 0                 0 
##             group   days_pre_booked    BookingChannel  UflyMemberStatus 
##                 0                 0                 0                 0 
## 0 observation(s) with NAs.
## 
## Estimated lambda: 0.01325227 
## 
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size 
##                 0                 0                 0                 0 
##             group   days_pre_booked    BookingChannel  UflyMemberStatus 
##                 0                 0                 0                 0 
## 0 observation(s) with NAs.
## 
## Estimated lambda: 0.01325227 
## 
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size 
##                 0                 0                 0                 0 
##             group   days_pre_booked    BookingChannel  UflyMemberStatus 
##                 0                 0                 0                 0 
## 0 observation(s) with NAs.
## 
## Estimated lambda: 0.01325227 
## 
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size 
##                 0                 0                 0                 0 
##             group   days_pre_booked    BookingChannel  UflyMemberStatus 
##                 0                 0                 0                 0 
## 0 observation(s) with NAs.
## 
## Estimated lambda: 0.01325227 
## 
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size 
##                 0                 0                 0                 0 
##             group   days_pre_booked    BookingChannel  UflyMemberStatus 
##                 0                 0                 0                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.01325227
##
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size
##                 0                 0                 0                 0
##             group   days_pre_booked    BookingChannel  UflyMemberStatus
##                 0                 0                 0                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.01325227
##
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size
##                 0                 0                 0                 0
##             group   days_pre_booked    BookingChannel  UflyMemberStatus
##                 0                 0                 0                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.01325227
##
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size
##                 0                 0                 0                 0
##             group   days_pre_booked    BookingChannel  UflyMemberStatus
##                 0                 0                 0                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.01325227
##
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size
##                 0                 0                 0                 0
##             group   days_pre_booked    BookingChannel  UflyMemberStatus
##                 0                 0                 0                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.01325227
##
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size
##                 0                 0                 0                 0
##             group   days_pre_booked    BookingChannel  UflyMemberStatus
##                 0                 0                 0                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.01325227
##
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size
##                 0                 0                 0                 0
##             group   days_pre_booked    BookingChannel  UflyMemberStatus
##                 0                 0                 0                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.01325227
##
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size
##                 0                 0                 0                 0
##             group   days_pre_booked    BookingChannel  UflyMemberStatus
##                 0                 0                 0                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.01325227
##
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size
##                 0                 0                 0                 0
##             group   days_pre_booked    BookingChannel  UflyMemberStatus
##                 0                 0                 0                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.01325227
##
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size
##                 0                 0                 0                 0
##             group   days_pre_booked    BookingChannel  UflyMemberStatus
##                 0                 0                 0                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.01325227
##
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size
##                 0                 0                 0                 0
##             group   days_pre_booked    BookingChannel  UflyMemberStatus
##                 0                 0                 0                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.01325227
##
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size
##                 0                 0                 0                 0
##             group   days_pre_booked    BookingChannel  UflyMemberStatus
##                 0                 0                 0                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.01325227
##
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size
##                 0                 0                 0                 0
##             group   days_pre_booked    BookingChannel  UflyMemberStatus
##                 0                 0                 0                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.01325227
##
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size
##                 0                 0                 0                 0
##             group   days_pre_booked    BookingChannel  UflyMemberStatus
##                 0                 0                 0                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.01325227
##
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size
##                 0                 0                 0                 0
##             group   days_pre_booked    BookingChannel  UflyMemberStatus
##                 0                 0                 0                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.01325227
##
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size
##                 0                 0                 0                 0
##             group   days_pre_booked    BookingChannel  UflyMemberStatus
##                 0                 0                 0                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.01325227
##
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size
##                 0                 0                 0                 0
##             group   days_pre_booked    BookingChannel  UflyMemberStatus
##                 0                 0                 0                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.01325227
##
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size
##                 0                 0                 0                 0
##             group   days_pre_booked    BookingChannel  UflyMemberStatus
##                 0                 0                 0                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.01325227
##
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size
##                 0                 0                 0                 0
##             group   days_pre_booked    BookingChannel  UflyMemberStatus
##                 0                 0                 0                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.01325227
##
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size
##                 0                 0                 0                 0
##             group   days_pre_booked    BookingChannel  UflyMemberStatus
##                 0                 0                 0                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.01325227
##
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size
##                 0                 0                 0                 0
##             group   days_pre_booked    BookingChannel  UflyMemberStatus
##                 0                 0                 0                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.01325227
##
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size
##                 0                 0                 0                 0
##             group   days_pre_booked    BookingChannel  UflyMemberStatus
##                 0                 0                 0                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.01325227
##
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size
##                 0                 0                 0                 0
##             group   days_pre_booked    BookingChannel  UflyMemberStatus
##                 0                 0                 0                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.01325227
##
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size
##                 0                 0                 0                 0
##             group   days_pre_booked    BookingChannel  UflyMemberStatus
##                 0                 0                 0                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.01325227
##
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size
##                 0                 0                 0                 0
##             group   days_pre_booked    BookingChannel  UflyMemberStatus
##                 0                 0                 0                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.01325227
##
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size
##                 0                 0                 0                 0
##             group   days_pre_booked    BookingChannel  UflyMemberStatus
##                 0                 0                 0                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.01325227
##
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size
##                 0                 0                 0                 0
##             group   days_pre_booked    BookingChannel  UflyMemberStatus
##                 0                 0                 0                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.01325227
##
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size
##                 0                 0                 0                 0
##             group   days_pre_booked    BookingChannel  UflyMemberStatus
##                 0                 0                 0                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.01325227
##
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size
##                 0                 0                 0                 0
##             group   days_pre_booked    BookingChannel  UflyMemberStatus
##                 0                 0                 0                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.01325227
##
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size
##                 0                 0                 0                 0
##             group   days_pre_booked    BookingChannel  UflyMemberStatus
##                 0                 0                 0                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.01325227
##
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size
##                 0                 0                 0                 0
##             group   days_pre_booked    BookingChannel  UflyMemberStatus
##                 0                 0                 0                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.01325227
##
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size
##                 0                 0                 0                 0
##             group   days_pre_booked    BookingChannel  UflyMemberStatus
##                 0                 0                 0                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.01325227
##
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size
##                 0                 0                 0                 0
##             group   days_pre_booked    BookingChannel  UflyMemberStatus
##                 0                 0                 0                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.01325227
##
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size
##                 0                 0                 0                 0
##             group   days_pre_booked    BookingChannel  UflyMemberStatus
##                 0                 0                 0                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.01325227
##
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size
##                 0                 0                 0                 0
##             group   days_pre_booked    BookingChannel  UflyMemberStatus
##                 0                 0                 0                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.01325227
##
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size
##                 0                 0                 0                 0
##             group   days_pre_booked    BookingChannel  UflyMemberStatus
##                 0                 0                 0                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.01325227
##
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size
##                 0                 0                 0                 0
##             group   days_pre_booked    BookingChannel  UflyMemberStatus
##                 0                 0                 0                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.01325227
##
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size
##                 0                 0                 0                 0
##             group   days_pre_booked    BookingChannel  UflyMemberStatus
##                 0                 0                 0                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.01325227
##
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size
##                 0                 0                 0                 0
##             group   days_pre_booked    BookingChannel  UflyMemberStatus
##                 0                 0                 0                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.01325227
##
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size
##                 0                 0                 0                 0
##             group   days_pre_booked    BookingChannel  UflyMemberStatus
##                 0                 0                 0                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.01325227
```

```
plot(1:10, SSE_curve, type="b", xlab="Number of Clusters", ylab="SSE")
```



```r
prototype_data <- drop_na(prototype_data)
```

```r
num_proto_clusters = 3
kpro <- kproto(as.data.frame(prototype_data), num_proto_clusters)
```

```
## # NAs in variables:
##        First_City Final_Destination        round_trip        group_size
##                 0                 0                 0                 0
##             group   days_pre_booked    BookingChannel  UflyMemberStatus
##                 0                 0                 0                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.01325227
```

Interpretation
Based on running our analysis on multiple samples, there seems to be a general cutoff point around 3 clusters. However, we can use 4 cluster too. But from the Hierarchical Cluster and Business standpoint, we believe that 3 cluster would be an optimum approach for the analysis.

DBScan

Description and Rationale for the Chosen Analysis
We Finally want to understand if the number density-based cluster are also the same as the K Prototype. Hence, we are interested to determine the cluster amount from an optimal distance. We established an optimal distance from eps from various clusters (excluded the code due to tedious clustering analysis).

```r
db <- dbscan(gower_dist, eps = 0.13, minPts = 5)
db
```

```
## DBSCAN clustering for 19935 objects.
## Parameters: eps = 0.13, minPts = 5
## The clustering contains 2 cluster(s) and 79 noise points.
##
##     0     1     2
##    79 19453   403
##
## Available fields: cluster, eps, minPts
```

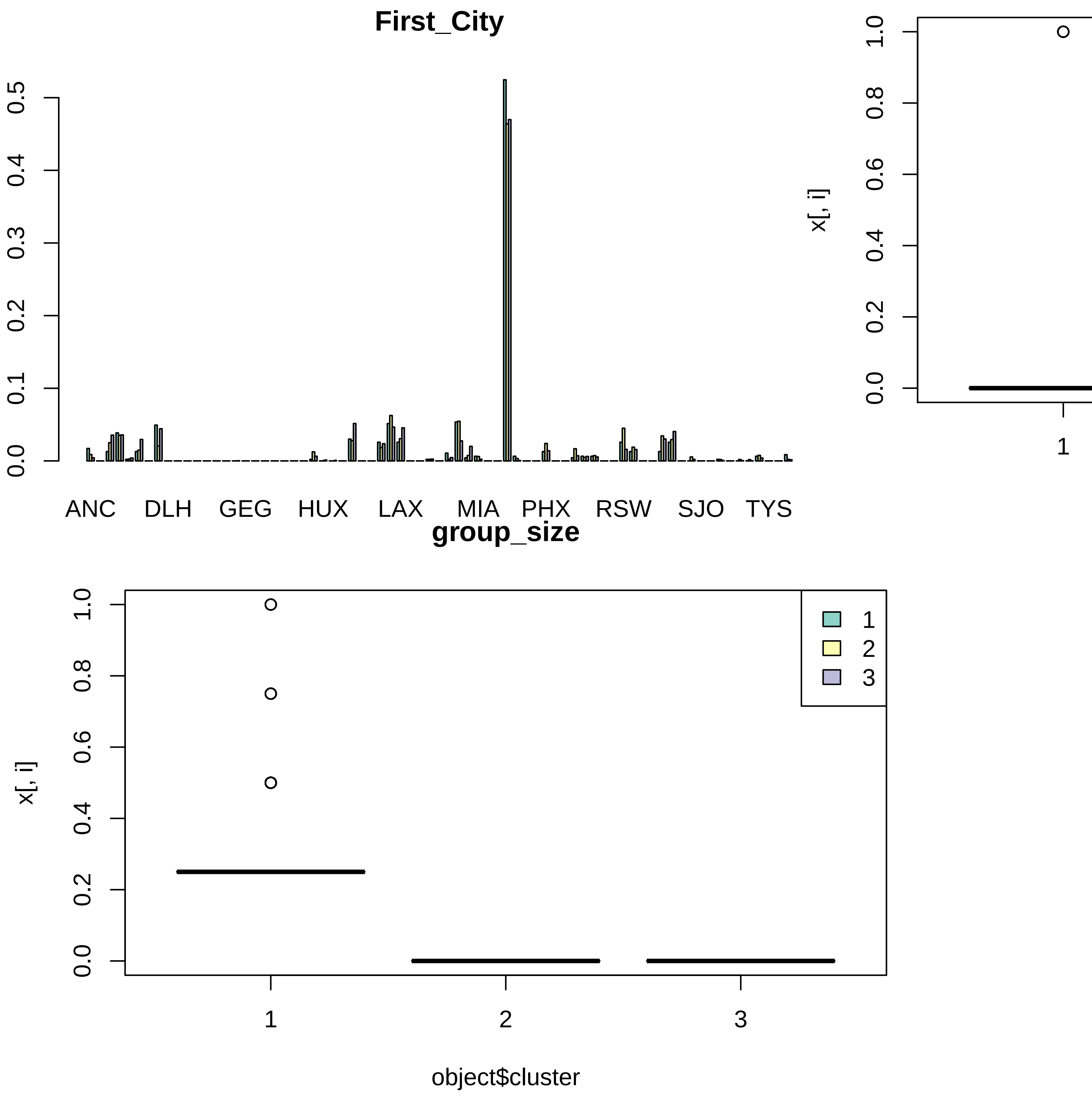
Interpretation and Conclusion
From the density-based cluster, we found that we observe 3 cluster which is the same about we considered from our hierarchical clustering.

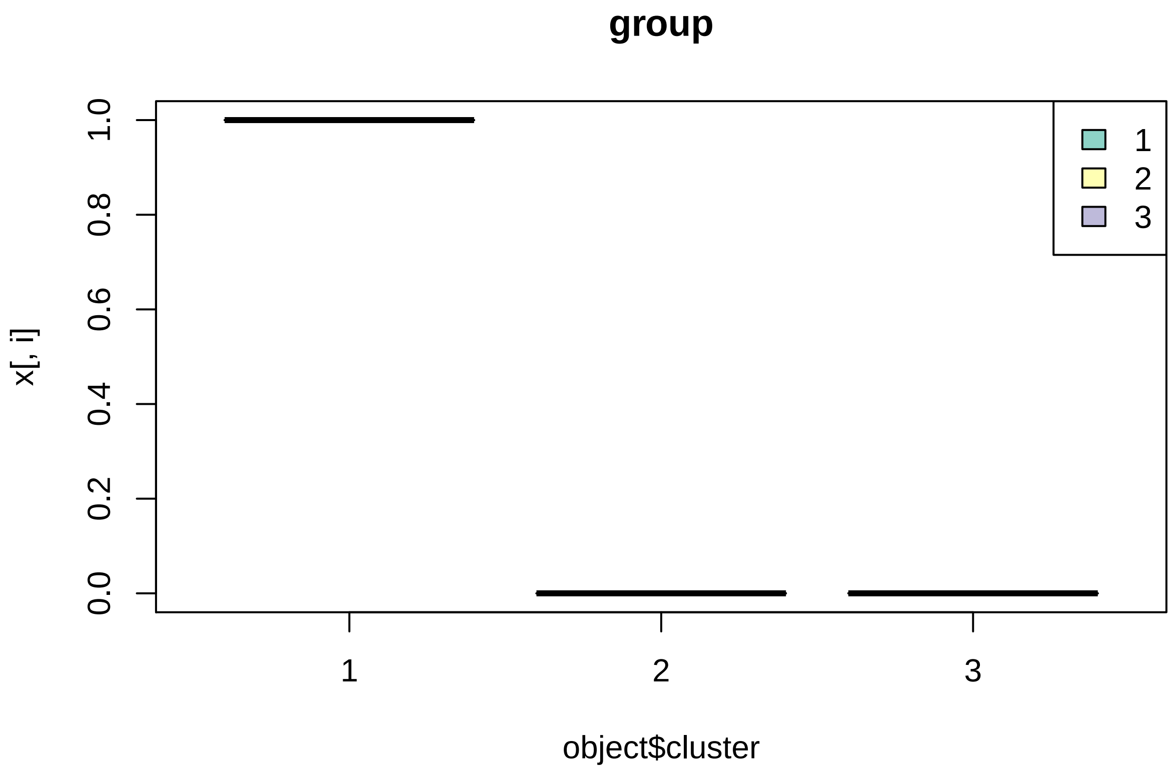
Now that we have the number of cluster we are interested to observe if we find patterns and relationships in our clusters.

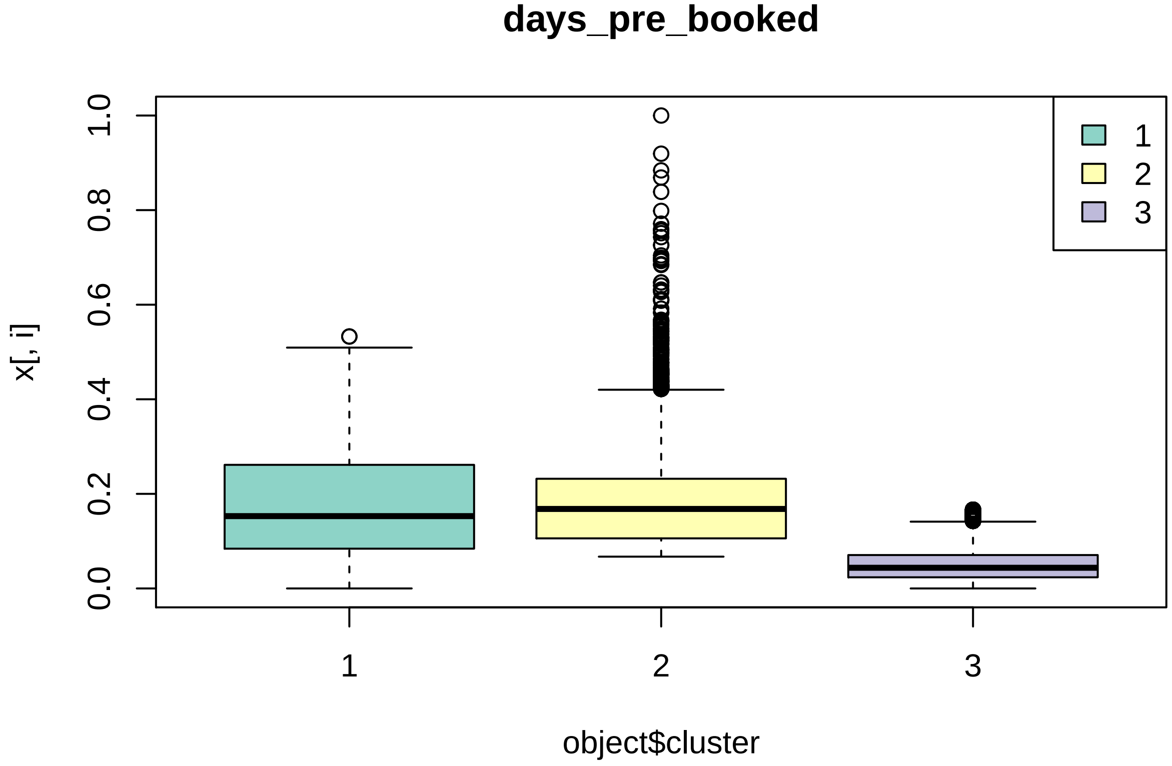
Final Clustering and Mergeing

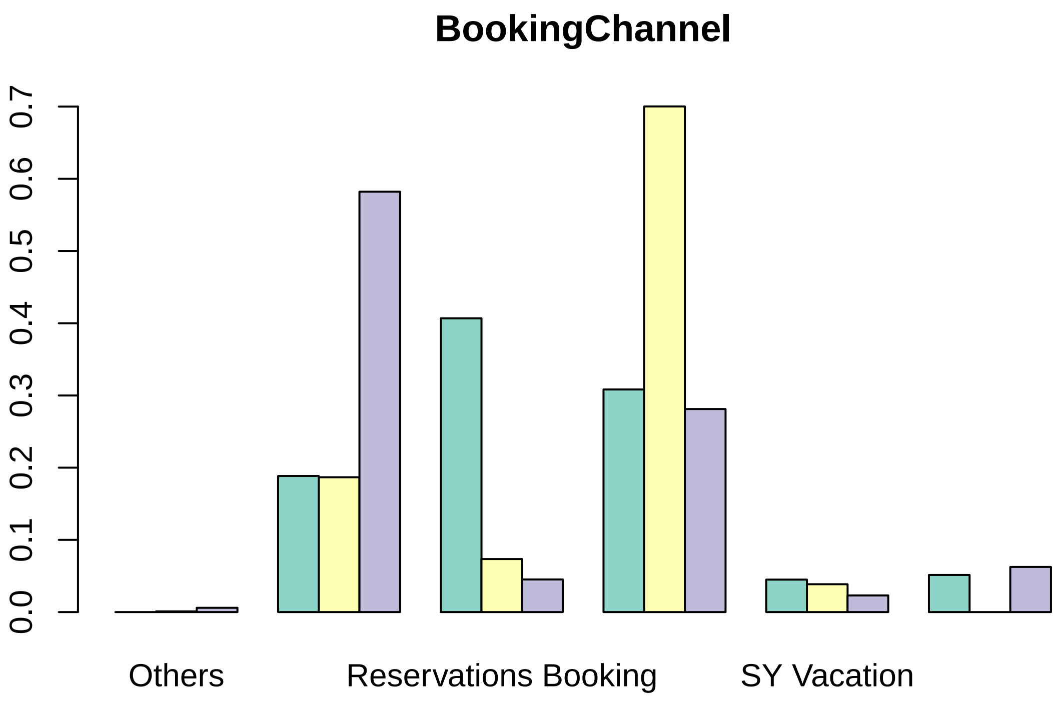
Description and Rationale for the Chosen Analysis
Finally, we use K Prototype to run our analysis. We divide the cluster and merge them back to the data frame to understand the whole data and derive better conclusions.

```r
clprofiles(kpro, as.data.frame(prototype_data))
```




group
1.0
0.8
0.6
0.4
0.2
0.0
x[, i]
1
2
3
object$cluster
1
2
3


days_pre_booked
1.0
0.8
0.6
0.4
0.2
0.0
x[, i]
1
2
3
object$cluster
1
2
3



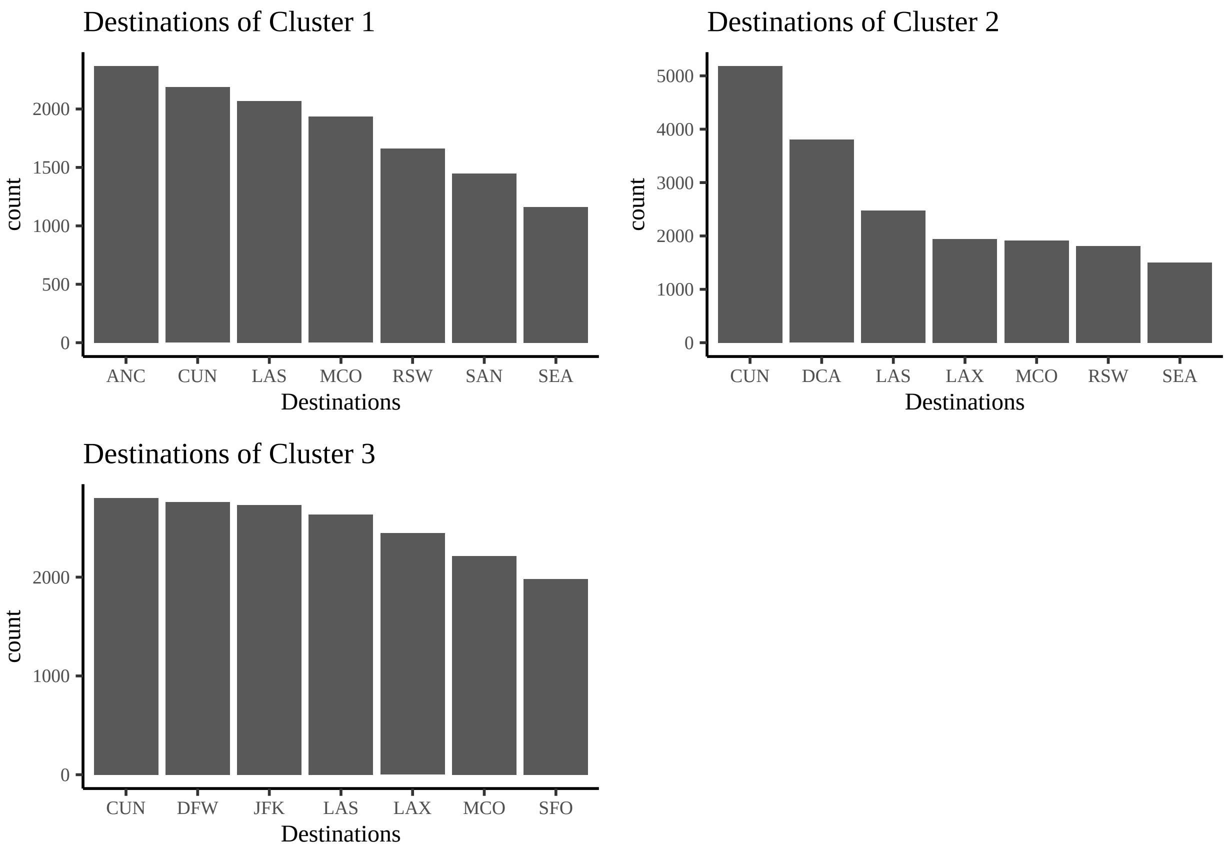
```
data_transformed$cluster <- kpro$cluster
final_segments <- merge(suncountry, data_transformed, by = 'PNRLocatorID')
```

Cluster Analysis

Top destination of segments

Description and Rationale for the Chosen Analysis
After getting the clusters we want to identify the gap and look for oppourtunites. We are intrsted where our customers are travelling.

```
c1 <- final_segments %>% filter(cluster == 1)

cl1_cities <- final_segments %>%
  filter(cluster == 1 & Final_Destination != 'MSP') %>%
  select(Final_Destination) %>%
  group_by(Final_Destination) %>%
  summarise(count = n()) %>%
  arrange(count) %>%
  top_n(7) %>%
  ggplot() + geom_bar(aes(sort(Final_Destination, decreasing = T), count),
                      stat = 'identity') + labs(x = 'Destinations',
                                                title = 'Destinations of Cluster 1')+
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) + my_theme

cl2_cities <- final_segments %>%
  filter(cluster == 2 & Final_Destination != 'MSP') %>%
  select(Final_Destination) %>%
  group_by(Final_Destination) %>%
  summarise(count = n()) %>%
  arrange(count) %>%
  top_n(7) %>%
  ggplot() + geom_bar(aes(sort(Final_Destination, decreasing = T), count),
                      stat = 'identity') + labs(x = 'Destinations',
                                                title = 'Destinations of Cluster 2')+
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) + my_theme

cl3_cities <- final_segments %>%
  filter(cluster == 3 & Final_Destination != 'MSP') %>%
  select(Final_Destination) %>%
  group_by(Final_Destination) %>%
  summarise(count = n()) %>%
  arrange(count) %>%
  top_n(7) %>%
  ggplot() + geom_bar(aes(sort(Final_Destination, decreasing = T), count),
                      stat = 'identity') + labs(x = 'Destinations',
                                                title = 'Destinations of Cluster 3')+
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) + my_theme

plot_grid(cl1_cities, cl2_cities, cl3_cities)
```

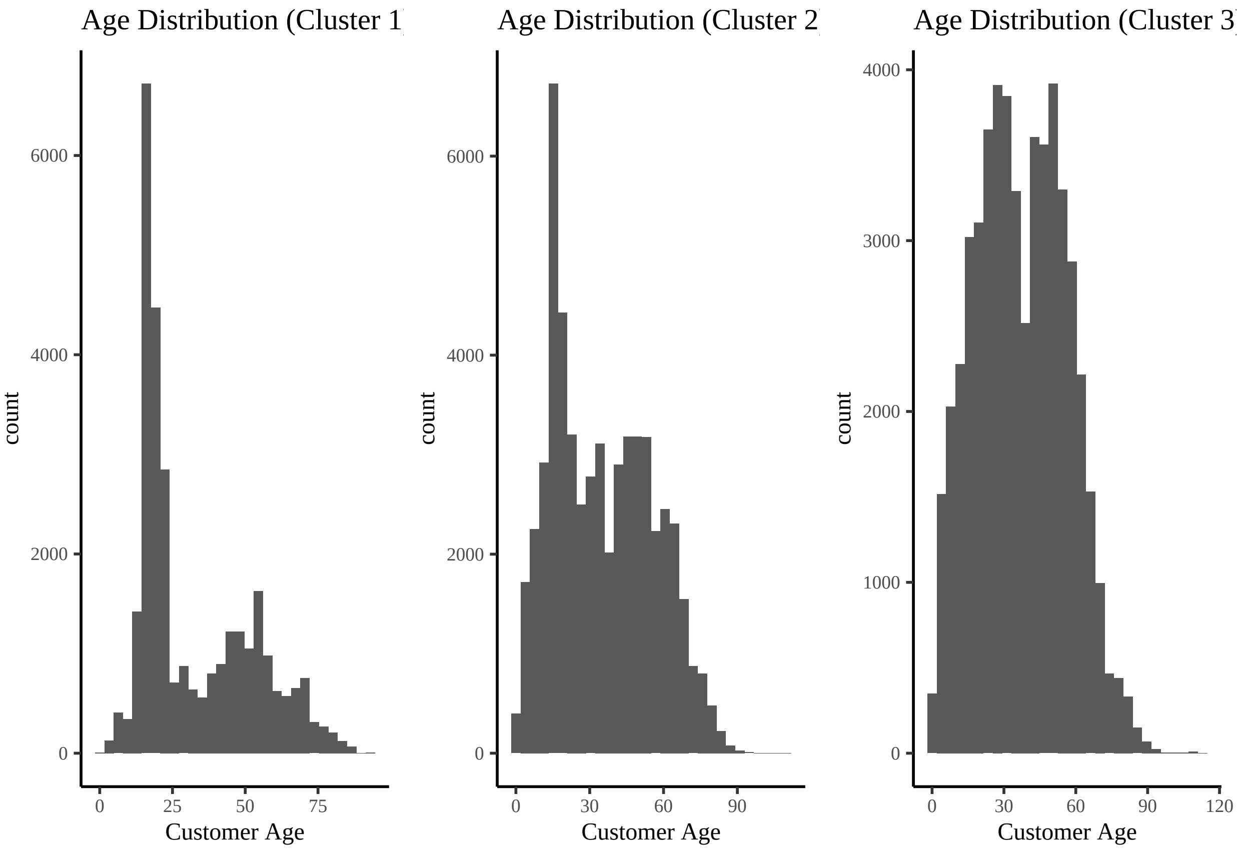


Interpretation and Conclusion
Cluster 2 is flying to locations like Cancun, Las Vegas, LA, and Orlando which suggests these might be vacationers. Cluster 3 flies most often to cities like Dallas, New York. Considering that these are solo travelers it is possible they are flying for business or other non-vacation reason. Cluster 1 and 3 both have the top 7 destinations as the same cities. It is less clear what this could mean.

Age distubution of clusters

Description and Rationale for the Chosen Analysis
After understanding the destinations we want to corroborate our assumption if these travellers are having families or solo travellers. This can be done by understanding the age distribution of our travellers.

```
age1 <- final_segments %>%
  filter(cluster == 1) %>%
  select(Age.x) %>%
  ggplot() + geom_histogram(aes(Age.x)) + labs(title = 'Age Distribution (Cluster 1)', x = 'Customer Ag

age2 <- final_segments %>%
  filter(cluster == 2) %>%
  select(Age.x) %>%
  ggplot() +
  geom_histogram(aes(Age.x)) +
  labs(title = 'Age Distribution (Cluster 2)', x = 'Customer Age') + my_theme

age3 <- final_segments %>%
  filter(cluster == 3) %>%
  select(Age.x) %>%
  ggplot() +
  geom_histogram(aes(Age.x)) +
  labs(title = 'Age Distribution (Cluster 3)', x = 'Customer Age') + my_theme

grid.arrange(age1,age2,age3, nrow = 1)
```



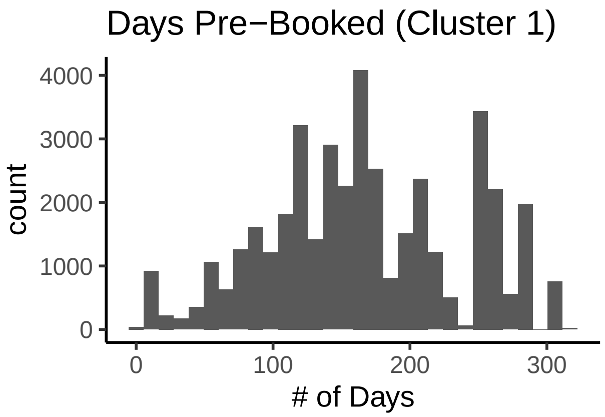
Interpretation and Conclusion

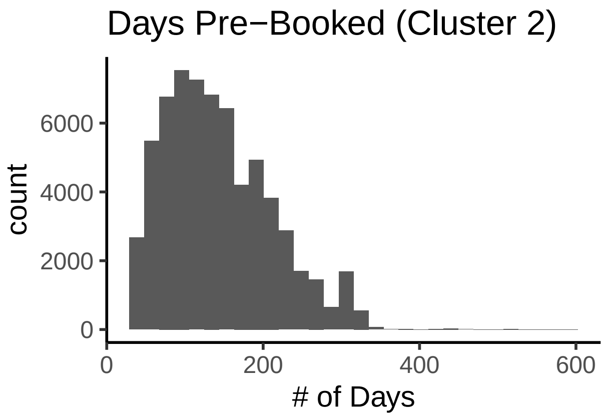
From comparing the results from age distribution and destination, we can conclude that we have segmented our data into 3 different classes of customers i.e. Business or Working class travelers who travel to big cities (Cluster 3), second solo and families travelers who travel to vacation destinations (Cluster 2), and finally, young travelers who travel to both business and vacations (Cluster 1).

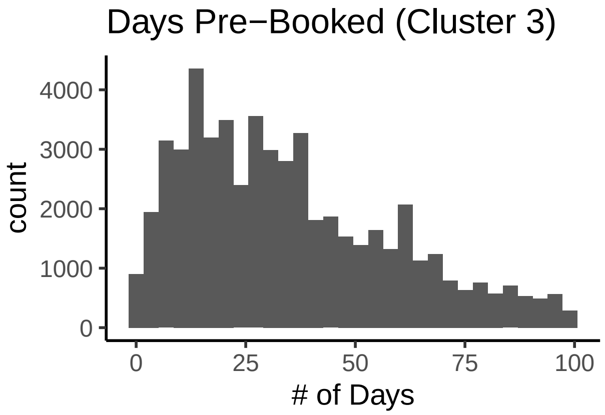
Day pre-booked based on clusters

Description and Rationale for the Chosen Analysis
Now that we have narrowed our customer segments we want to further observe if there is any difference in the days pre-booked. This might further help.

```
prebook1 <- final_segments %>%
  filter(cluster == 1) %>%
  select(days_pre_booked) %>%
  ggplot() +
  geom_histogram(aes(days_pre_booked)) +
  labs(title= 'Days Pre-Booked (Cluster 1)', x = '# of Days')

prebook2 <- final_segments %>%
  filter(cluster == 2) %>%
  select(days_pre_booked) %>%
  ggplot() + geom_histogram(aes(days_pre_booked)) +
  labs(title= 'Days Pre-Booked (Cluster 2)', x = '# of Days')

prebook3 <- final_segments %>%
  filter(cluster == 3) %>%
  select(days_pre_booked) %>%
  ggplot() + geom_histogram(aes(days_pre_booked)) +
  labs(title= 'Days Pre-Booked (Cluster 3)', x = '# of Days')

grid.arrange(prebook1,prebook2,prebook3, nrow = 2)
```







Interpretation and Conclusion
From all the above observation, as we segmented our travelers into 1.Business or Working class travelers(Cluster 3) 2.Solo and families travelers who travel to vacation destinations (Cluster 2) 3.Young travelers who travel to both business and vacations (Cluster 1)

We can observe that business people tend to book tickets much later when compared to vacationers people. Further, solo travelers book tickets from a wide range i.e. this might be a pattern from a long vacation to immediate vacation or small travel.

Summary of Customer SegmentationFrom our analysis, we observed that we have 3 segments of consumers

1.Business or Working class travelers who travel to cities and book ticket late. 2.Solo and family vacation traveler who pre-book tickets quite early. 3. Young travelers who travel to both business and vacations and book tickets sparsely.

Hub and Spoke Analysis.

The words “hub” and “spoke” create a pretty vivid image of how this system works. A hub is a central airport that flights are routed through, and spokes are the routes that planes take out of the hub airport. Most major airlines have multiple hubs.

Our Client, Sun country uses Hub and Spoke model with it Hub based on MSP. The purpose of the hub-and-spoke system is to save airlines money and give passengers better routes to destinations. Airplanes are an airline’s most valuable commodity, and every flight has certain set costs. Each seat on the plane represents a portion of the total flight cost. For each seat that is filled by a passenger, an airline lowers its break-even price, which is the seat price at which an airline stops losing money and begins to show a profit on the flight.

We have noticed that there is a huge gap between the number of flights flying out from MSP and the number of flights coming back. This generates an opportunity for Sun Country. The company can generate revenue from this kind of flight pattern and target customers at a more personal level.

Assumption
We assume that customers didn't book both the go and return ticket at the same time would not fly back through Sun Country airline, and since the huge difference between leaving MSP and flying back, this assumption seems reasonable. We then select those data and do the clustering and try finding the patterns of the flights that are not returning.

```
#  Normalizing numeric columns
normalize <- function(x){
  return ((x - min(x))/(max(x) - min(x)))
  }

sun_single_norm$Age <- normalize(sun_single_norm$Age)
sun_single_norm$total_amount <- normalize(sun_single_norm$total_amount)
sun_single_norm$date_diff <- normalize(sun_single_norm$date_diff)
```

Clustering for routes pattern

Clustering using k-prototypes

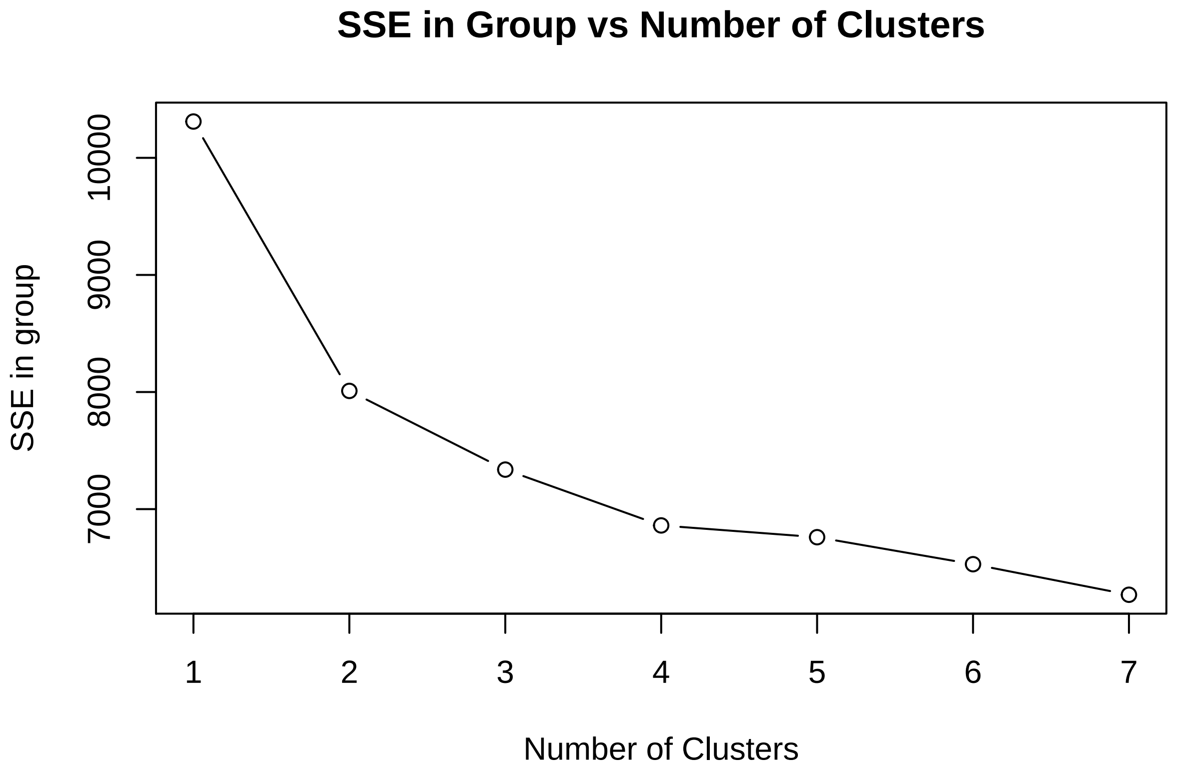
Description and Rationale for the Chosen Analysis
As our data have mixed data type we want to use k-prototypes to find an optimal number of clusters. In order to do this, we will plot the sum of squared error curve to try and find the optimal number of clusters. Then, we will loop through the clustering several times to try and get a sense of generalized performance, as k-prototypes chooses random starting points for the clustering.

```
SSE_curve <- c()
for (n in 1:7) {
  kcluster <- kproto(as.data.frame(sun_single_norm), n)
  print(kcluster$withinss)
  sse <- sum(kcluster$withinss)
  SSE_curve[n] <- sse
}
```

```
## # NAs in variables:
##    ServiceEndCity        GenderCode                Age    BookingChannel
##                 0                 0                  0                 0
##      total_amount         date_diff    service_weekday ServiceStartMonth
##                 0                 0                  0                 0
##      class_change
##                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.02349921
##
## [1] 10310.11
## # NAs in variables:
```

```
##    ServiceEndCity        GenderCode               Age    BookingChannel
##                 0                 0                 0                 0
##      total_amount         date_diff   service_weekday ServiceStartMonth
##                 0                 0                 0                 0
##      class_change
##                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.02349921
##
## [1] 4088.286 3921.403
## # NAs in variables:
##    ServiceEndCity        GenderCode               Age    BookingChannel
##                 0                 0                 0                 0
##      total_amount         date_diff   service_weekday ServiceStartMonth
##                 0                 0                 0                 0
##      class_change
##                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.02349921
##
## [1] 2280.251 2802.365 2254.920
## # NAs in variables:
##    ServiceEndCity        GenderCode               Age    BookingChannel
##                 0                 0                 0                 0
##      total_amount         date_diff   service_weekday ServiceStartMonth
##                 0                 0                 0                 0
##      class_change
##                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.02349921
##
## [1] 1730.327 1728.725 1789.771 1611.904
## # NAs in variables:
##    ServiceEndCity        GenderCode               Age    BookingChannel
##                 0                 0                 0                 0
##      total_amount         date_diff   service_weekday ServiceStartMonth
##                 0                 0                 0                 0
##      class_change
##                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.02349921
##
## [1] 1274.073 1227.957 1561.283 1538.796 1158.099
## # NAs in variables:
##    ServiceEndCity        GenderCode               Age    BookingChannel
##                 0                 0                 0                 0
##      total_amount         date_diff   service_weekday ServiceStartMonth
##                 0                 0                 0                 0
##      class_change
##                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.02349921
##
## [1] 1042.2908  946.4718  968.6732 1274.9403 1389.8335  907.4748
## # NAs in variables:
##    ServiceEndCity        GenderCode               Age    BookingChannel
##                 0                 0                 0                 0
##      total_amount         date_diff   service_weekday ServiceStartMonth
##                 0                 0                 0                 0
##      class_change
##                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.02349921
##
## [1] 1143.5654  697.1945  866.4193  755.2567 1095.9486  821.9754  888.4637
```

```
plot(1:7, SSE_curve, type="b", xlab="Number of Clusters", ylab="SSE in group",
     main = 'SSE in Group vs Number of Clusters')
```



Interpretation
Based on running our analysis on multiple samples, there seems to be a general cutoff point around 3 clusters. However, we can use 4 clusters too. But from the Hierarchical Cluster and Business standpoint, we believe that 3 clusters would be an optimum approach for the analysis.

```
# Clustering
set.seed(42)
kproto_single_norm <- kproto(as.data.frame(sun_single_norm), 3)
```

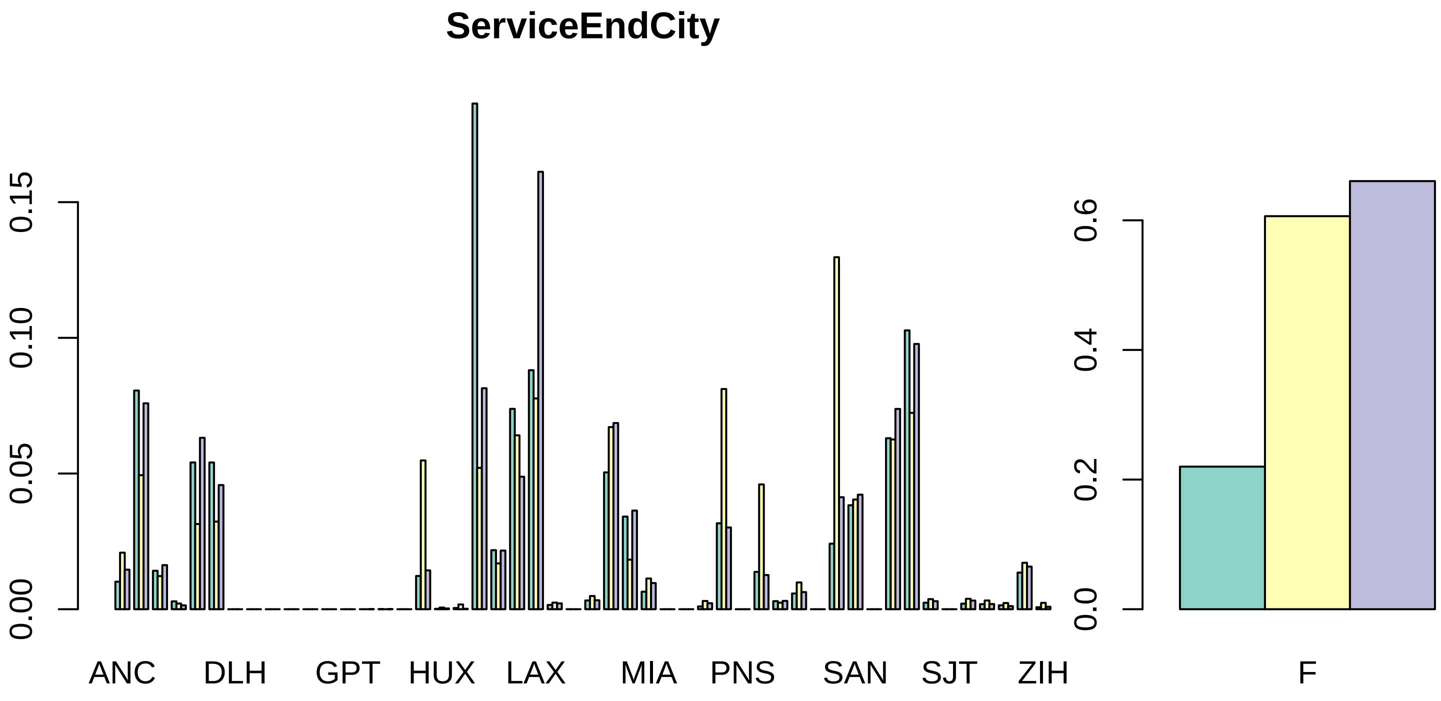
```
## # NAs in variables:
##    ServiceEndCity        GenderCode               Age    BookingChannel
##                 0                 0                 0                 0
##      total_amount         date_diff   service_weekday ServiceStartMonth
##                 0                 0                 0                 0
##      class_change
##                 0
## 0 observation(s) with NAs.
##
## Estimated lambda: 0.02349921
```

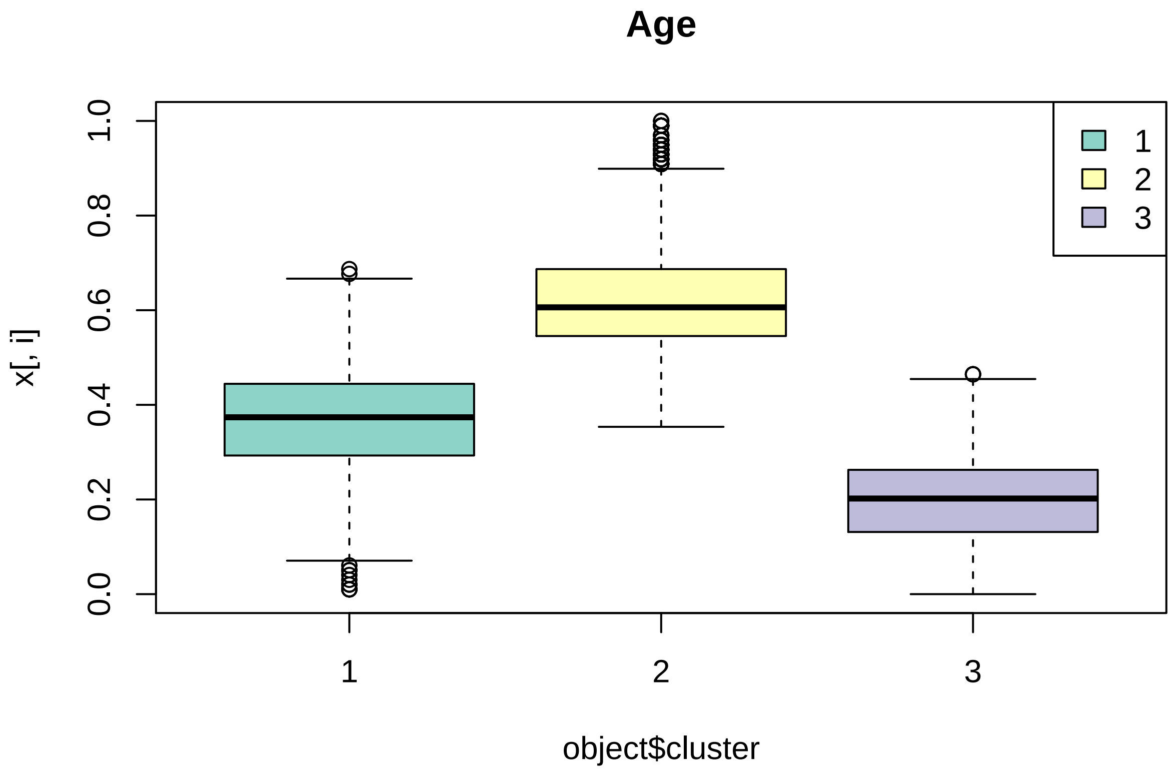
```
# Centers of the cluster (with mean and mode)
kproto_single_norm$centers
```

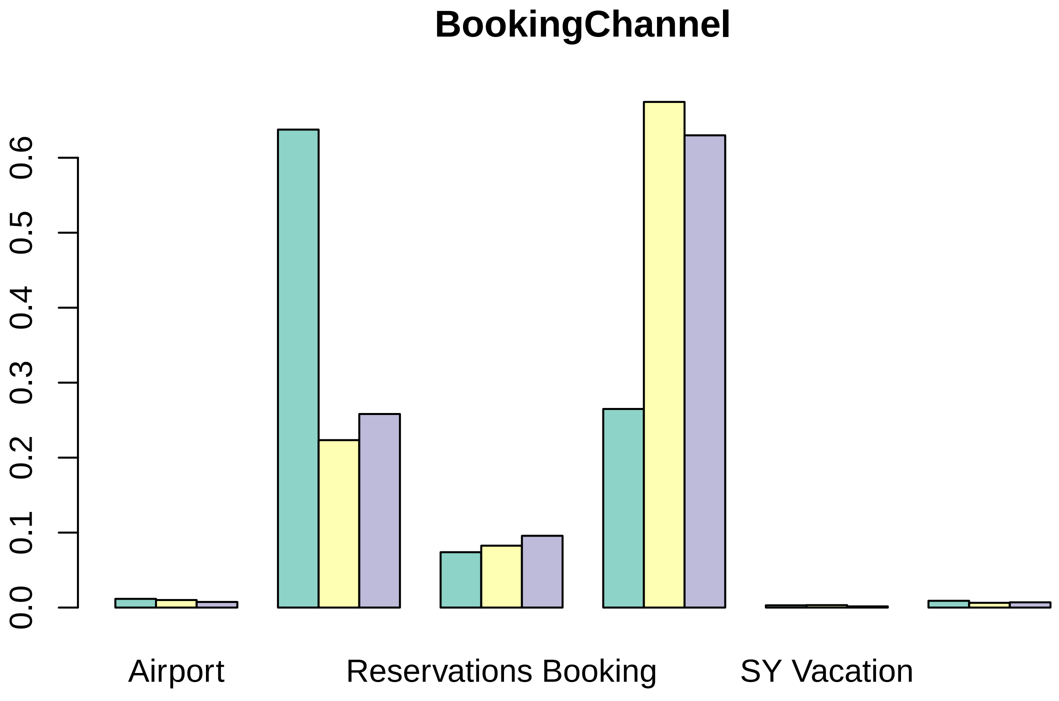
```
##   ServiceEndCity GenderCode       Age      BookingChannel total_amount
## 1            JFK          M 0.3703848     Outside Booking   0.03476508
## 2            RSW          F 0.6192174 SCA Website Booking   0.03255796
## 3            LAX          F 0.1985980 SCA Website Booking   0.03044997
##    date_diff service_weekday ServiceStartMonth class_change
## 1 0.04593714          Monday               Dec  non-updated
## 2 0.07913890          Sunday               Dec  non-updated
## 3 0.07462658          Sunday               Jul  non-updated
```

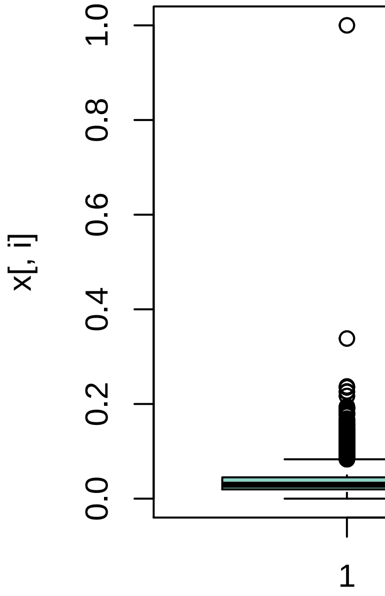
```
sun_single_origin$cluster <- kproto_single_norm$cluster
```

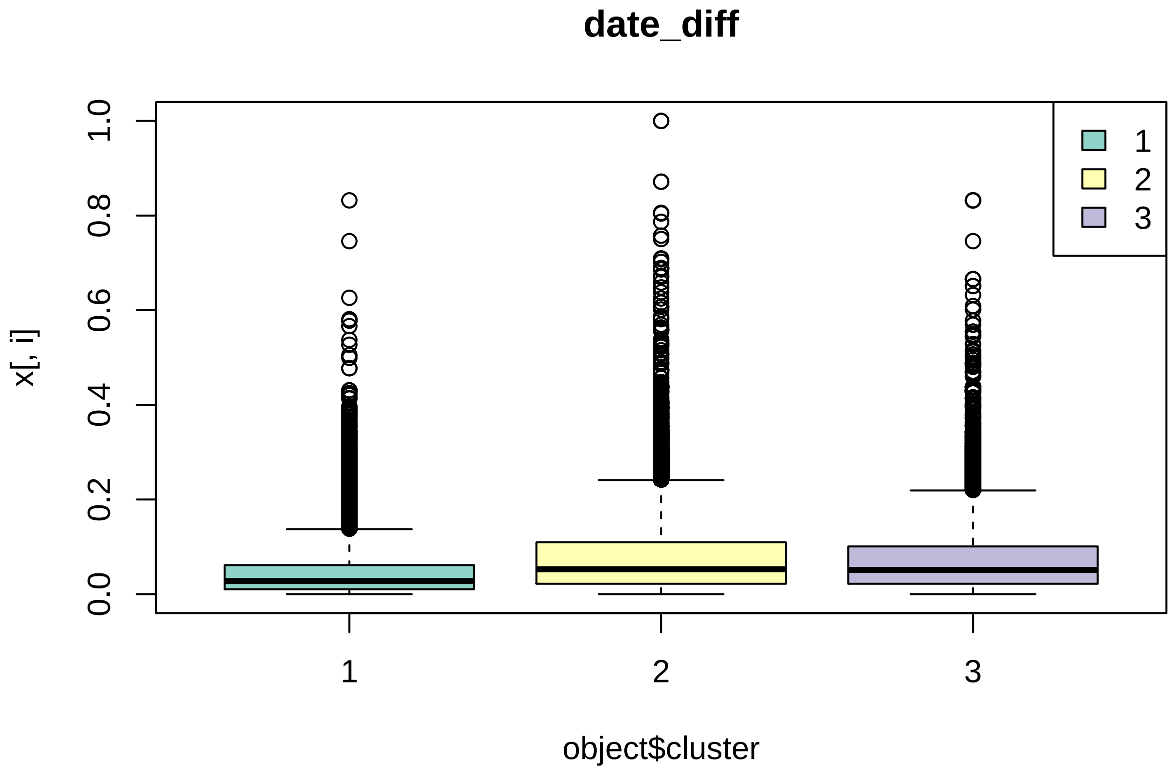
```
clprofiles(kproto_single_norm, as.data.frame(sun_single_norm))
```

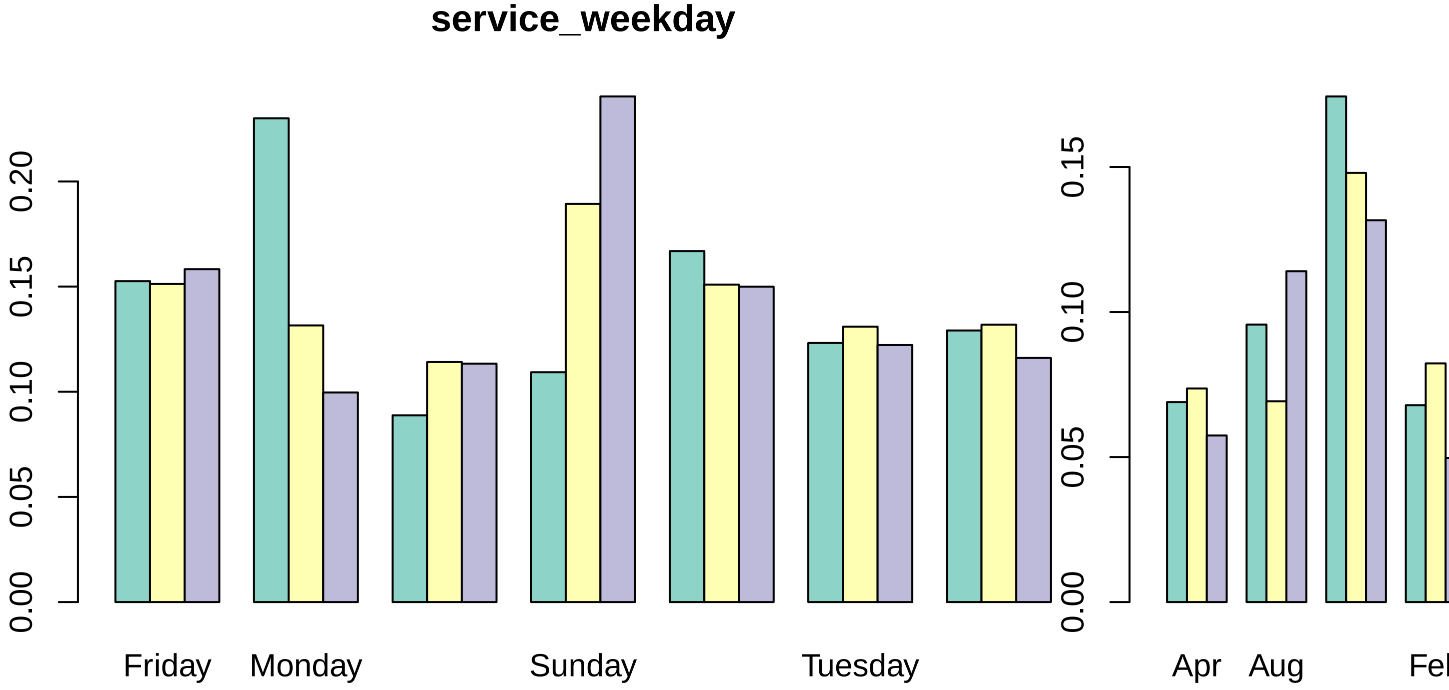


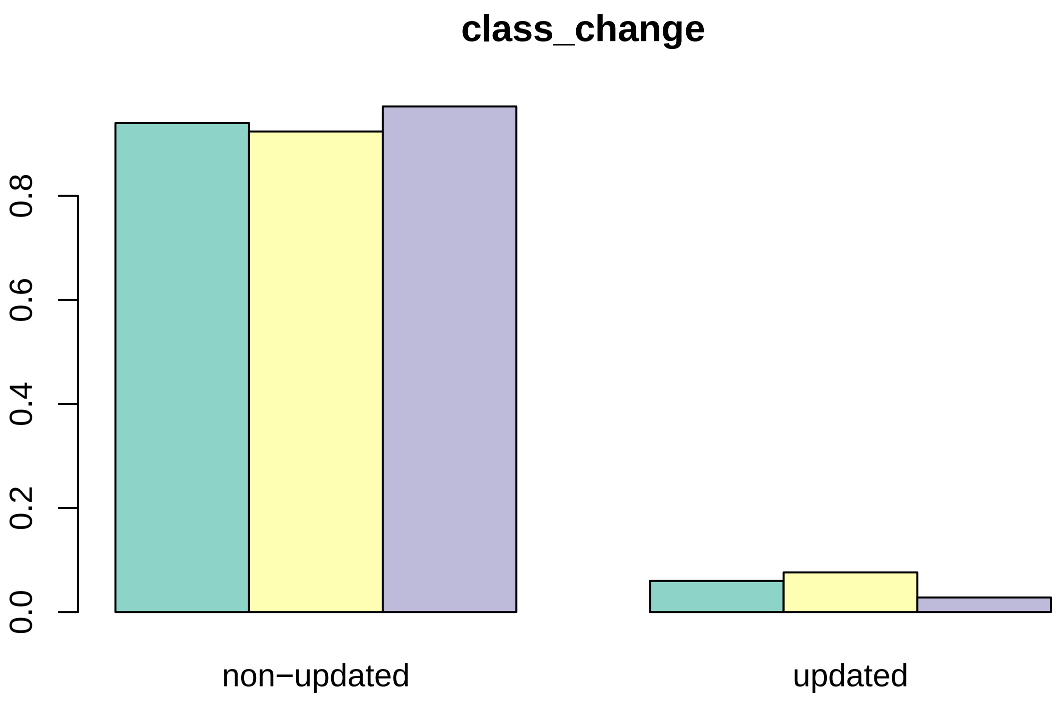

Age
1.0
0.8
0.6
0.4
0.2
0.0
x[, i]
1
2
3
1
2
3
object$cluster


BookingChannel
0.6
0.5
0.4
0.3
0.2
0.1
0.0
Airport
Reservations Booking
SY Vacation


1.0
0.8
0.6
0.4
0.2
0.0
x[, i]
1


date_diff
x[, i]
1.0
0.8
0.6
0.4
0.2
0.0
1
2
3
object$cluster


service_weekday
0.20
0.15
0.10
0.05
0.00
Friday
Monday
Sunday
Tuesday
0.15
0.10
0.05
0.00
Apr
Aug
Fel



Interpretation We get three clusters from the results with apparent different End city and age distributions. People flying to LAX are having a much younger age than others, and people flying to RSW are having higher ages. Also, the major booking channels in every group are also different from others. We will be able to dictate a distinct marketing strategy by looking into their patterns deeply and comparing them to similar ones.

From this, we can finally identify that most of our travelers are moving to JFK, RSW, and LAX but are not returning though Sun Country Airlines. Hence, we have identified an gap where our client can utilize. Thus, we want to understand how these cities differ and identify patterns.

Route patterns for Non-return tickets - JFK

We examine the cluster in which service end city is mostly JFK and focus on people visiting JFK in this group.

```
# All tickets to JFK
sun_to_jfk <- sun %>%
  filter(ServiceEndCity == 'JFK' & ServiceStartCity == 'MSP' )

# Return tickets to JFK
sun_return_jfk <- sun %>%
  group_by(PNRLocatorID, PaxName) %>%
  mutate(max_coup = max(CouponSeqNbr)) %>%
  filter(max_coup > 1) %>%
  ungroup() %>%
  filter(ServiceEndCity == 'JFK' & ServiceStartCity == 'MSP')

sun_return_jfk$ServiceStartMonth <- factor(sun_return_jfk$ServiceStartMonth,
                                           levels = c('Jan','Feb','Mar','Apr',
                                                      'May','Jun','Jul','Aug','Sep',
                                                      'Oct','Nov','Dec'))
```

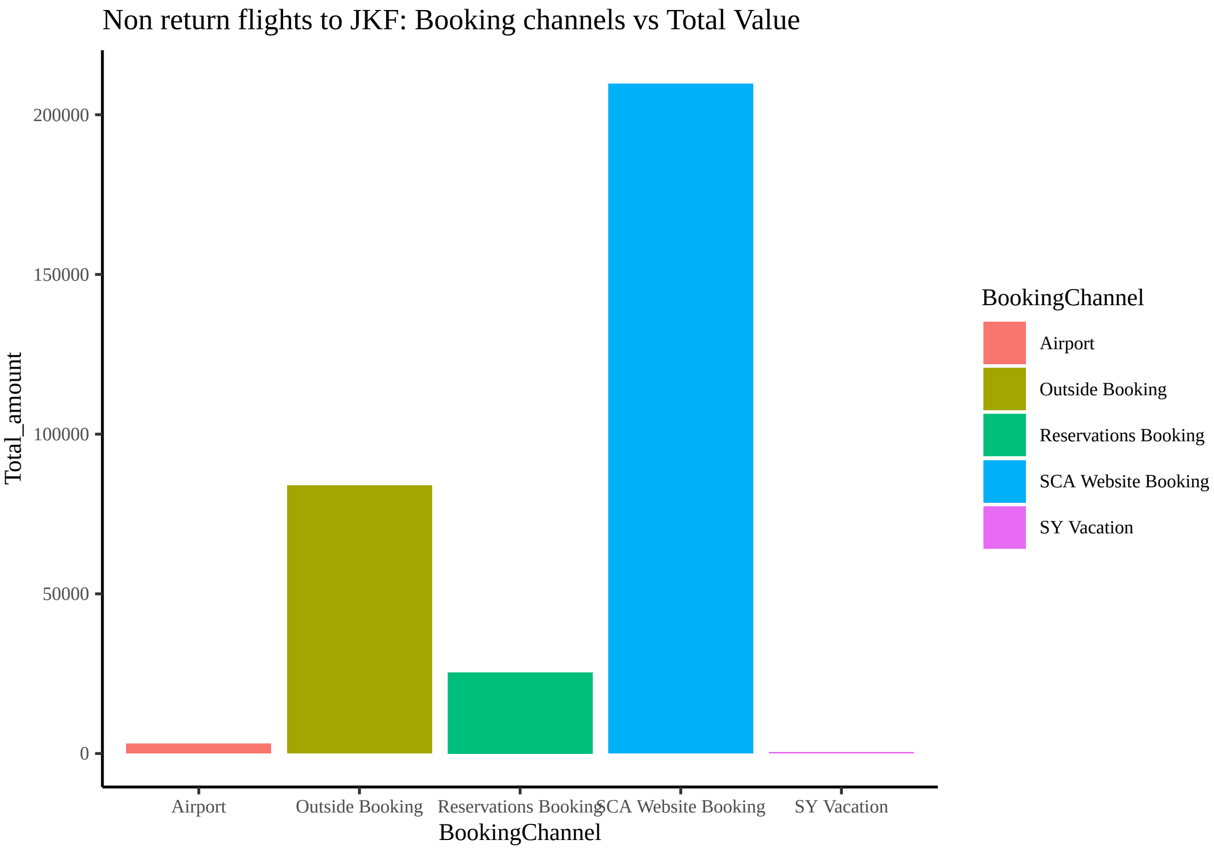
```
# Single tickets to JFK
sun_single_jfk <- sun_single_origin %>% filter(ServiceEndCity == 'JFK' & cluster == 2)
sun_single_jfk$ServiceStartMonth <- factor(sun_single_jfk$ServiceStartMonth,
                                           levels = c('Jan','Feb','Mar','Apr',
                                                      'May','Jun','Jul','Aug',
                                                      'Sep','Oct','Nov','Dec'))
```

JFK - Booking Channel Patterns

```
# single tickets summary
jfk_sum <- sun_single_jfk %>%
  group_by(BookingChannel) %>%
  summarize(Count = n(),
            Avg_amount = mean(total_amount),
            Total_amount = sum(total_amount),
            Days_before = mean(date_diff),
            Age = mean(Age))

jfk_sum[c(1,2,5,6)]
```

```
## # A tibble: 5 x 4
##   BookingChannel       Count Days_before   Age
##   <fct>                <int>       <dbl> <dbl>
## 1 Airport                 16        2.81  60.4
## 2 Outside Booking        333       38.7   65.6
## 3 Reservations Booking   102       53.8   65.4
## 4 SCA Website Booking    921       34.8   60.2
## 5 SY Vacation              1       13     51
```

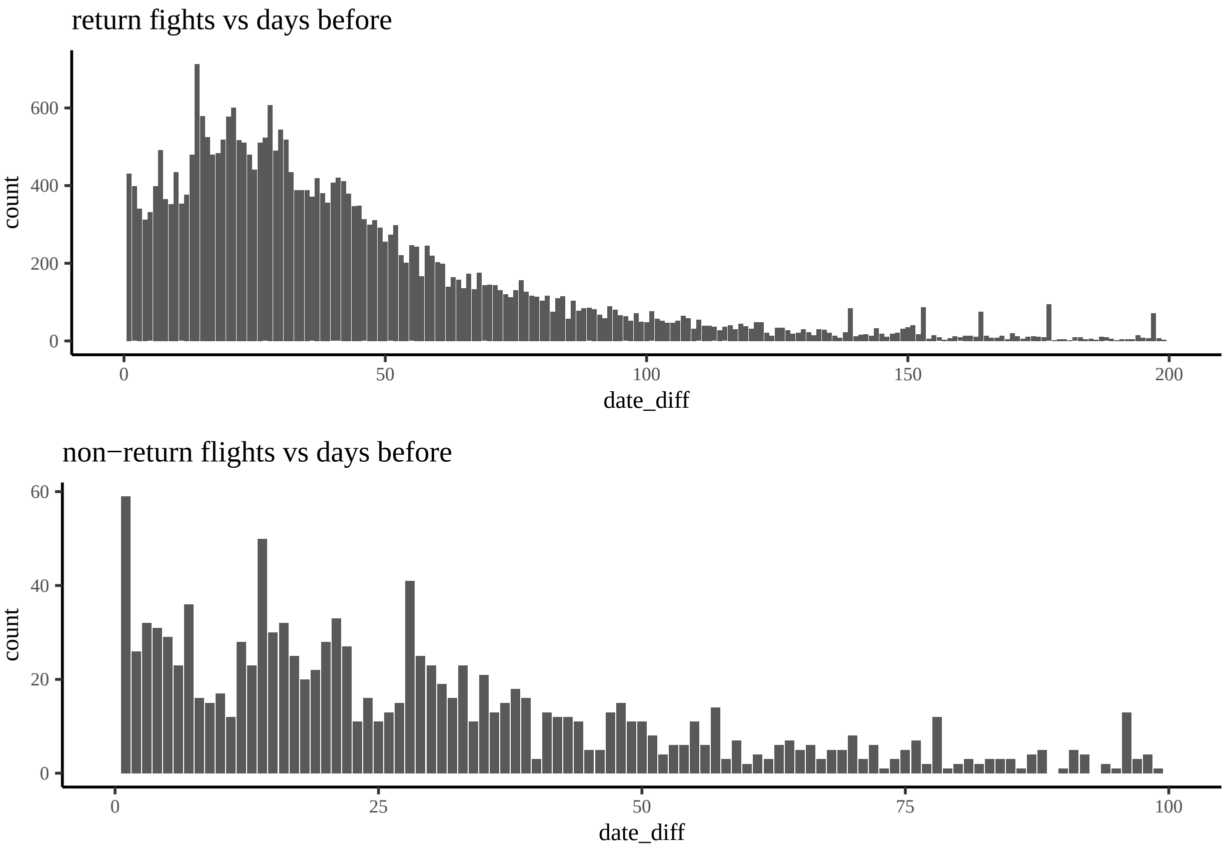
```
ggplot(jfk_sum, aes(x=BookingChannel, y=Total_amount)) +
  geom_histogram(aes(fill = BookingChannel),stat = 'identity') +
  ggtitle('Non return flights to JKF: Booking channels vs Total Value') +
  theme(legend.position = "none") + my_theme
```



Interpretation As shown in the table, the average age of the customers in this group is around 38. Also, the booking channels from outside booking and SCA website booking are the top two channels for having the highest value. The airport has a little contribution to the ticket booking source. The customers are booking the tickets on average 27 days before the flight.

JFK - Pre-Booking Patterns

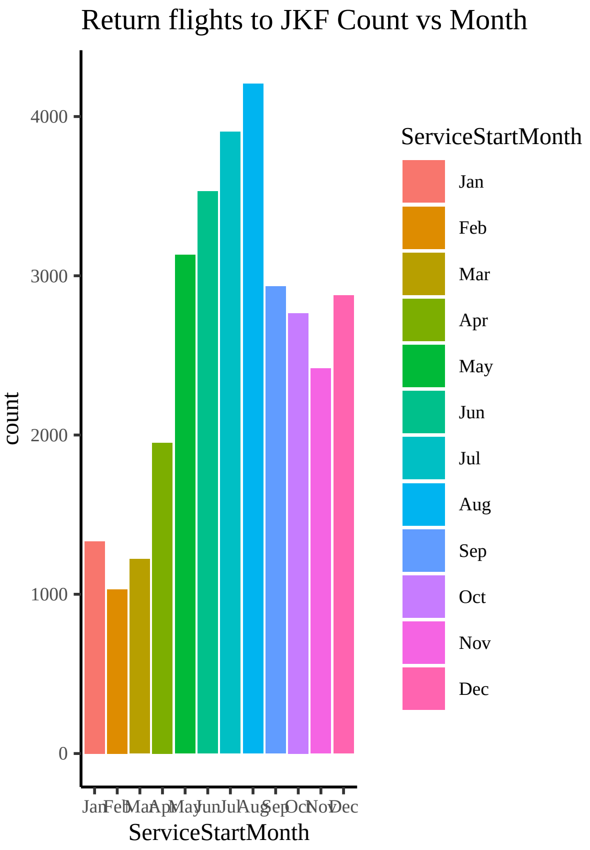
```
# Return tickets dates diff
jd1<-ggplot(sun_return_jfk, aes(x=date_diff)) +
  geom_histogram(stat = 'count')+
  scale_x_continuous(limits = c(0,200)) +
  ggtitle('return fights vs days before') + my_theme

# non return dates diff
jd2<- ggplot(sun_single_jfk, aes(x=date_diff)) +
  geom_histogram(stat = 'count')+
  scale_x_continuous(limits = c(0,100)) +
  ggtitle('non-return flights vs days before') + my_theme

grid.arrange(jd1, jd2)
```

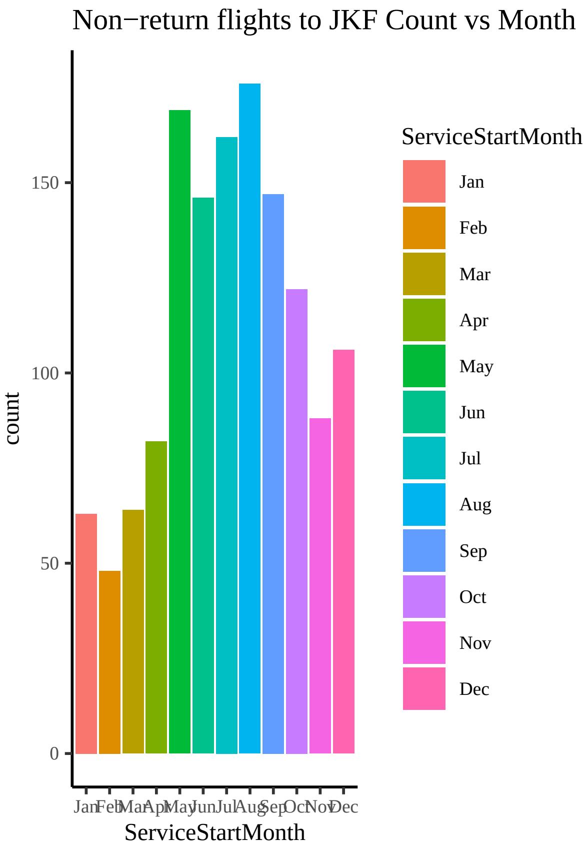


Interpretation By comparing to the flights with return tickets, we can see the influence of how many days buying the tickets in advance will cause the result. For those flights with return tickets, they are majorly brought within 50 days, on the other hand the one-way tickets are mostly booked in 25 days. This indicates that people are more likely to buy return tickets if they have more time before the flights, and that is something the company can take advantage of.

JFK - Seasonity of Travel

```r
# Return tickets vs service month
jm1 <- ggplot(sun_return_jfk, aes(x=ServiceStartMonth, fill = ServiceStartMonth)) +
  geom_histogram(stat = 'count') +
  ggtitle('Return flights to JKF Count vs Month') +
  theme(legend.position = "none") + my_theme

# single ticket to jkf
jm2 <- ggplot(sun_single_jfk, aes(x = ServiceStartMonth, fill = ServiceStartMonth)) +
  geom_histogram(stat = 'count') +
  ggtitle('Non-return flights to JKF Count vs Month') +
  theme(legend.position = "none") + my_theme

grid.arrange(jm1,jm2, nrow=1)
```





Interpretation For the seasonality distribution, we can recognize the identical pattern from the flights with or without return tickets. There are much more flights in the summer than in another season. Also, the number of flights in December is also very considerable, the time that Sun Country can focus on marketing their airlines.

Conclusion Noticed that in the gender graph, we can see that male customers constitute a larger proportion than females in this group, which destination is majorly JFK airport in New York City. That somewhat indicates that if the company market our airlines packages to male customers that are between 30 and 40, the number of buying return tickets can be increased. Also, for customers booking their tickets to New York in 25 days in advanced, Sun Country can provide some benefits, such as collaborating with New York companies for discount coupon on their products, to attract customers buying return tickets.

We now look deeper into the cluster in which service end city is mostly RSW. We focus on the flights heading to RSW for further research.

Route atterns for Non-return tickets - RSW

We now look deeper into the cluster which service end city is mostly RSW. We focus on the flights heading to RSW for further research.

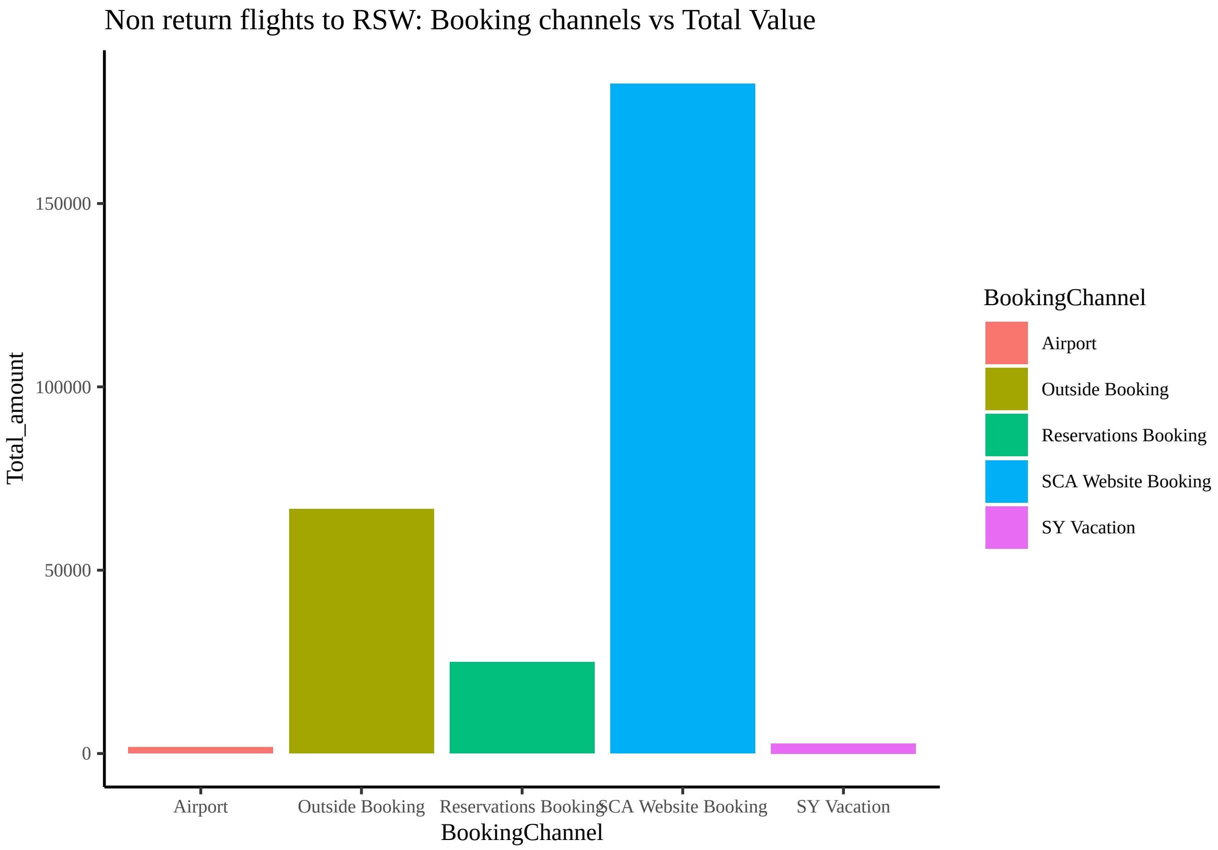
```r
# All tickets to RSW
sun_to_rsw <- sun %>%
  filter(ServiceEndCity == 'RSW' & ServiceStartCity == 'MSP')

# Return tickets to RSW
sun_return_rsw <- sun %>% group_by(PNRLocatorID, PaxName) %>%
  mutate(max_coup = max(CouponSeqNbr)) %>%
  filter(max_coup > 1) %>%
  ungroup() %>%
  filter(ServiceEndCity == 'RSW' & ServiceStartCity == 'MSP')

sun_return_rsw$ServiceStartMonth <- factor(sun_return_rsw$ServiceStartMonth,
                                       levels = c('Jan','Feb','Mar','Apr',
                                                  'May','Jun','Jul','Aug',
                                                  'Sep','Oct','Nov','Dec'))
# Single tickets to RSW
sun_single_rsw <- sun_single_origin %>%
  filter(ServiceEndCity == 'RSW' & cluster == 3)

sun_single_rsw$ServiceStartMonth <- factor(sun_single_rsw$ServiceStartMonth,
                                       levels = c('Jan','Feb','Mar','Apr',
                                                  'May','Jun','Jul','Aug',
                                                  'Sep','Oct','Nov','Dec'))
```

JFK - Booking Channel Patterns

```
# single tickets summary
rsw_sum <- sun_single_rsw %>% group_by(BookingChannel) %>%
  summarize(Count = n(),
            Avg_amount = mean(total_amount),
            Total_amount = sum(total_amount),
            Days_before = mean(date_diff),
            Age = mean(Age))

rsw_sum[c(1,2,5,6)]
```

```
## # A tibble: 5 x 4
##   BookingChannel       Count Days_before   Age
##   <fct>                <int>       <dbl> <dbl>
## 1 Airport                  7        17.7  18.9
## 2 Outside Booking        227        73.0  14.8
## 3 Reservations Booking    93        71.8  12.8
## 4 SCA Website Booking    731        68.2  17.8
## 5 SY Vacation              4       107    16
```

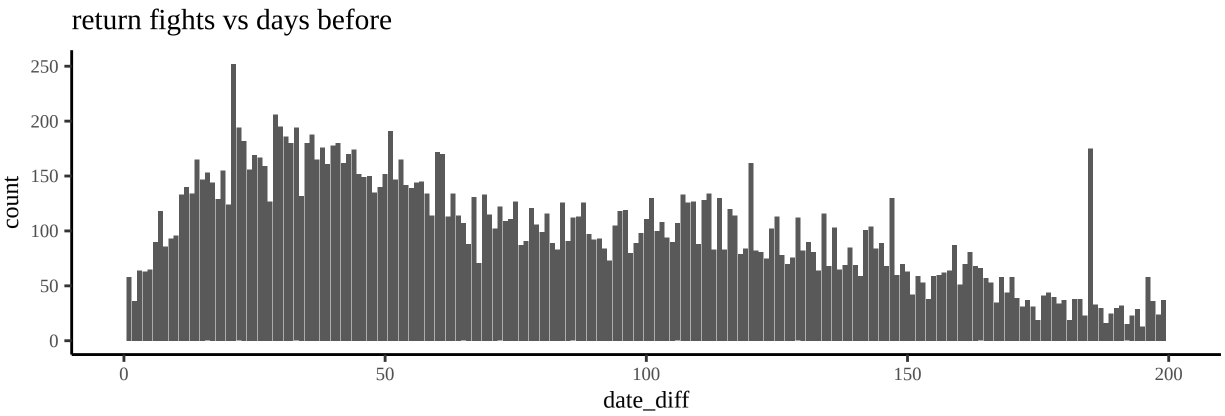
```
ggplot(rsw_sum, aes(x=BookingChannel, y=Total_amount)) +
  geom_histogram(aes(fill = BookingChannel),stat = 'identity') +
  ggtitle('Non return flights to RSW: Booking channels vs Total Value') +
  theme(legend.position = "none") + my_theme
```

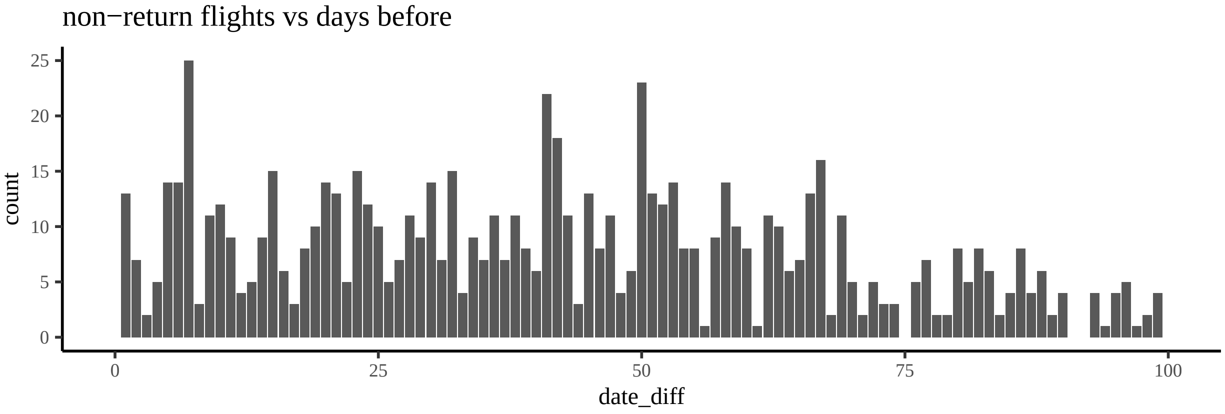


Interpretation The age of customers flying to RSW is much higher than others, around 65 on average. The customers not having return tickets are the ones that are almost retired, old people. Also, the non-return tickets in this route are majority from the SCA website, having a much heavier weight than the other channels. People flying in this route book their tickets earlier than those in other groups, with an average of 60 days in advance.

RSW - Pre-Booking Patterns

```
# Return tickets dates diff
rd1<-ggplot(sun_return_rsw, aes(x=date_diff)) +
  geom_histogram(stat = 'count')+
  scale_x_continuous(limits = c(0,200)) +
  ggtitle('return fights vs days before') + my_theme

# Non-Return dates diff
rd2<- ggplot(sun_single_rsw, aes(x=date_diff)) +
  geom_histogram(stat = 'count')+
  scale_x_continuous(limits = c(0,100)) +
  ggtitle('non-return flights vs days before') + my_theme

grid.arrange(rd1, rd2)
```

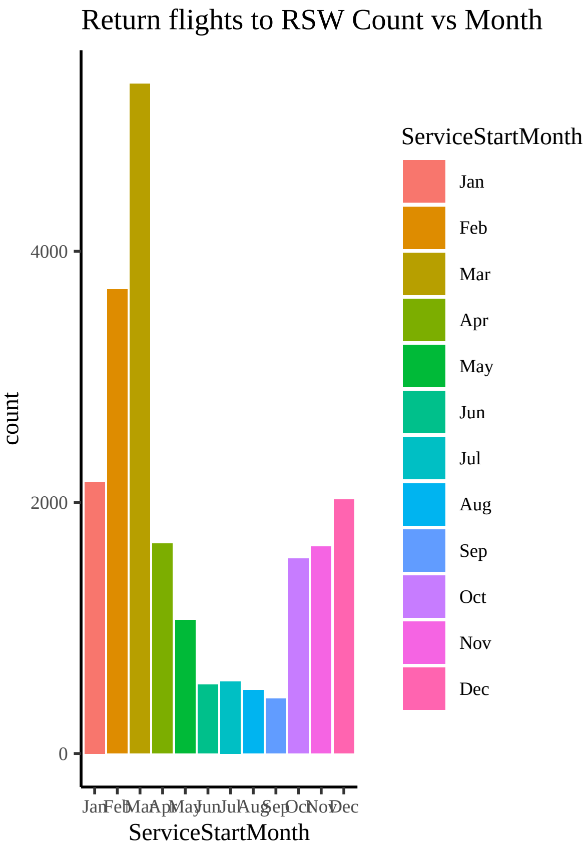


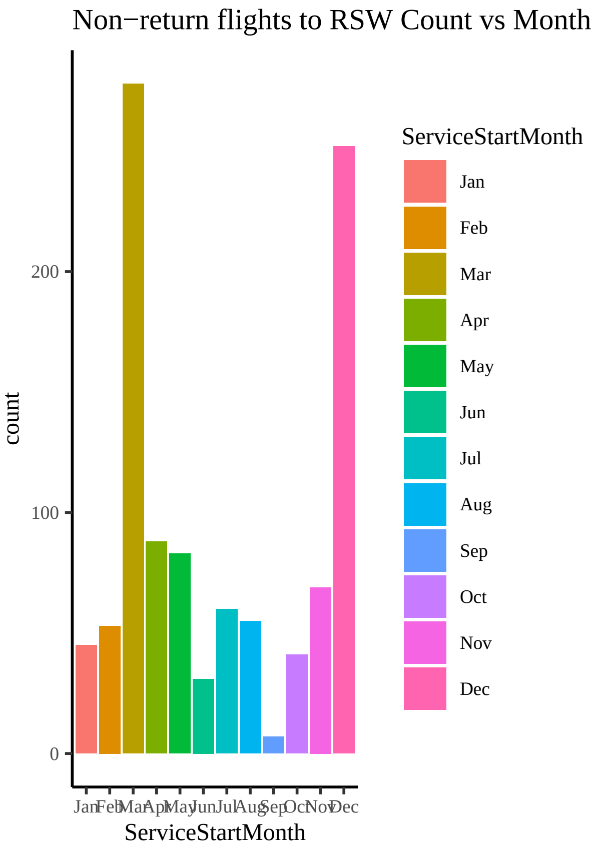


Interpretation From the plot of how many days in advance of buying tickets, there is no significant different part between the return tickets and the non-return ones. Customers flying in this route mostly book their flights in 2 months on average. We can guess that since RSW is in Florida, some good place to go for a vacation, customers schedule much earlier than in other groups.

RSW - Seasonilioty of Travel

```
# Return ticket to rsw
rm1 <- ggplot(sun_return_rsw, aes(x=ServiceStartMonth, fill = ServiceStartMonth)) +
  geom_histogram(stat = 'count') +
  ggtitle('Return flights to RSW Count vs Month') +
  theme(legend.position = "none") + my_theme

# Single ticket to rsw
rm2 <- ggplot(sun_single_rsw, aes(x = ServiceStartMonth, fill = ServiceStartMonth)) +
  geom_histogram(stat = 'count') +
  ggtitle('Non-return flights to RSW Count vs Month') +
  theme(legend.position = "none") + my_theme

grid.arrange(rm1,rm2, nrow=1)
```





Interpretation Clearly shown in the graph, there is an extremely large ratio of non-return ticket flights from October to December compared to the return tickets graph. This situation should be solved for the Sun Country if they want to have more values from the return tickets.

Conclution Sun Country should realize that there are many elders flying to RSW without having a return ticket, hence the marketing direction should focus on those people above 60, with an image of elderly-friendly on this route. In this route, people book their tickets earlier than others in advanced, having much more time to plan for the flight. This indicates that probably the route is for vacation, and Sun Country can make some collaboration with a local travel agent to boost their customers to have return tickets in their travel package.

Also, in October till December, the number of the one-way ticket to RSW grows abnormally and SunCountry has to fix this problem. This phenomenon implies that people are tending to go to a warmer city in Florida in December more casually. sun country should have more flexible winter packages for return tickets on this route or marketing their airlines more heavily in this period of time to hold the values.

At last, we studied the flights to LAX in the third cluster.

Route atterns for Non-return tickets - LAX

At last, we studied the flights to LAX in the thirs cluster.

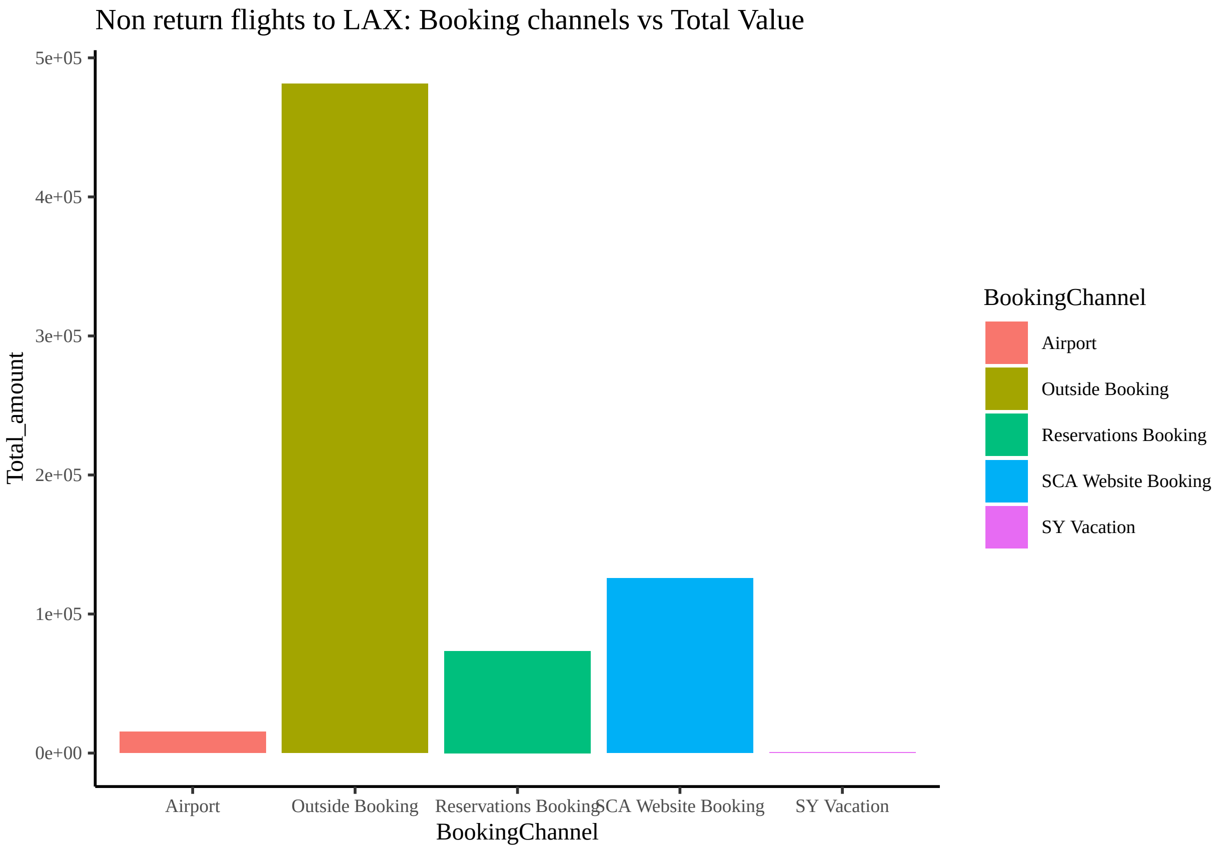
```r
# All tickets
sun_to_lax <- sun %>% filter(ServiceEndCity == 'LAX' & ServiceStartCity == 'MSP')

# Return tickets
sun_return_lax <- sun %>%
  group_by(PNRLocatorID, PaxName) %>%
  mutate(max_coup = max(CouponSeqNbr)) %>%
  filter(max_coup > 1) %>% ungroup() %>%
  filter(ServiceEndCity == 'LAX' & ServiceStartCity == 'MSP')

sun_return_lax$ServiceStartMonth <- factor(sun_return_lax$ServiceStartMonth,
                                           levels = c('Jan','Feb','Mar','Apr',
                                                      'May','Jun','Jul','Aug',
                                                      'Sep','Oct','Nov','Dec'))
# Single tickets
sun_single_lax <- sun_single_origin %>%
  filter(ServiceEndCity == 'LAX' & cluster == 1)
sun_single_lax$ServiceStartMonth <- factor(sun_single_lax$ServiceStartMonth,
                                               levels = c('Jan','Feb','Mar','Apr',
                                                          'May','Jun','Jul','Aug',
                                                          'Sep','Oct','Nov','Dec'))
```

LAX - Booking Channel Patterns

```
# Single tickets summary
lax_sum <- sun_single_lax %>% group_by(BookingChannel) %>%
  summarize(Count = n(),
            Avg_amount = mean(total_amount),
            Total_amount = sum(total_amount),
            Days_before = mean(date_diff),
            Age = mean(Age))

lax_sum[c(1,2,5,6)]
```

```
## # A tibble: 5 x 4
##   BookingChannel       Count Days_before   Age
##   <fct>                <int>       <dbl> <dbl>
## 1 Airport                 43        1.47  40.0
## 2 Outside Booking       1665       27.0   38.2
## 3 Reservations Booking   221       18.7   41.6
## 4 SCA Website Booking    469       30.3   41.5
## 5 SY Vacation              2      146     40
```

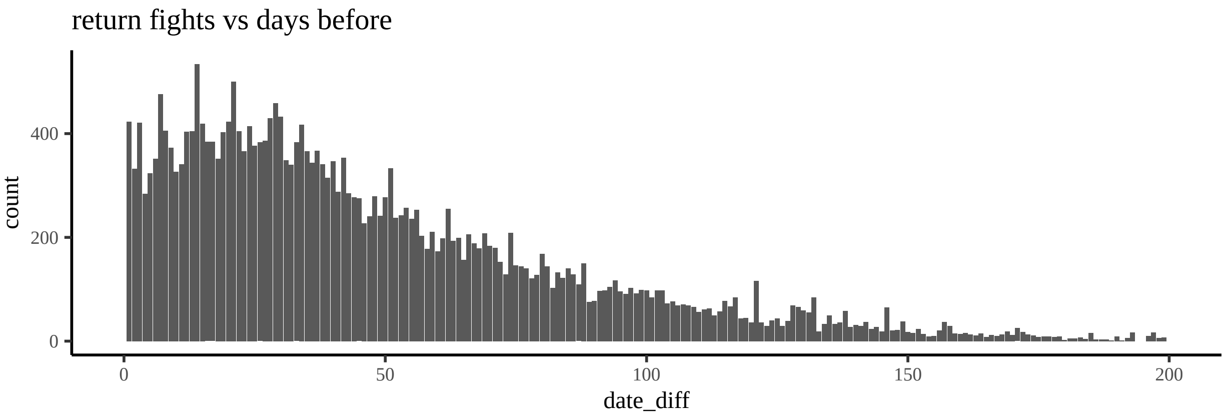
```
ggplot(lax_sum, aes(x=BookingChannel, y=Total_amount)) +
  geom_histogram(aes(fill = BookingChannel),stat = 'identity') +
  ggtitle('Non return flights to LAX: Booking channels vs Total Value') +
  theme(legend.position = "none") + my_theme
```

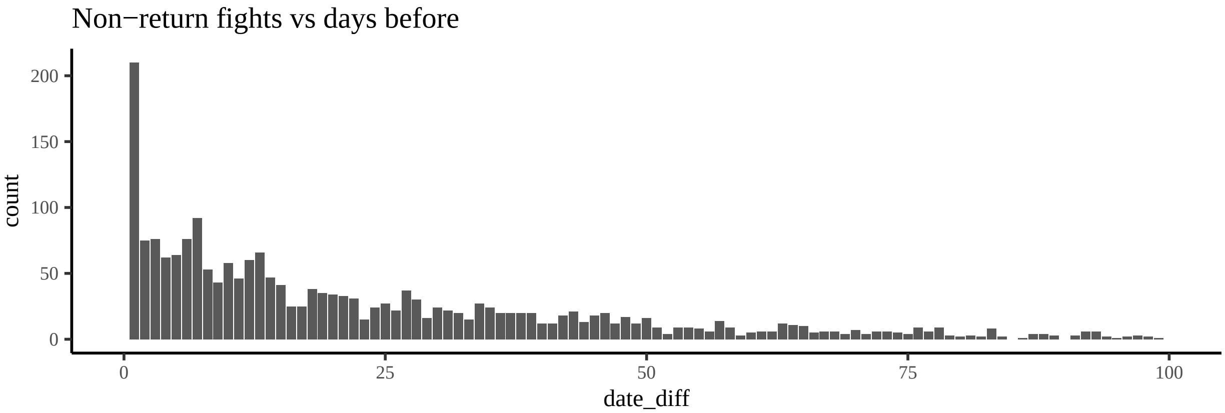


Interpretation Customers in this group are mostly heading to LAX, having an average age of 25, with all the young people. The booking sources from outside booking and SCA website booking are having the same weight. Also, people in this group flying to LAX are booking the tickets one and a half months before.

LAX - Pre-Booking Patterns

```r
# Return tickets dates diff
ld1 <- ggplot(sun_return_lax, aes(x=date_diff)) +
  geom_histogram(stat = 'count')+
  scale_x_continuous(limits = c(0,200)) +
  ggtitle('return fights vs days before') + my_theme

# non return dates diff
ld2 <- ggplot(sun_single_lax, aes(x=date_diff)) +
  geom_histogram(stat = 'count')+
  scale_x_continuous(limits = c(0,100)) +
  ggtitle('Non-return fights vs days before') + my_theme

grid.arrange(ld1,ld2)
```

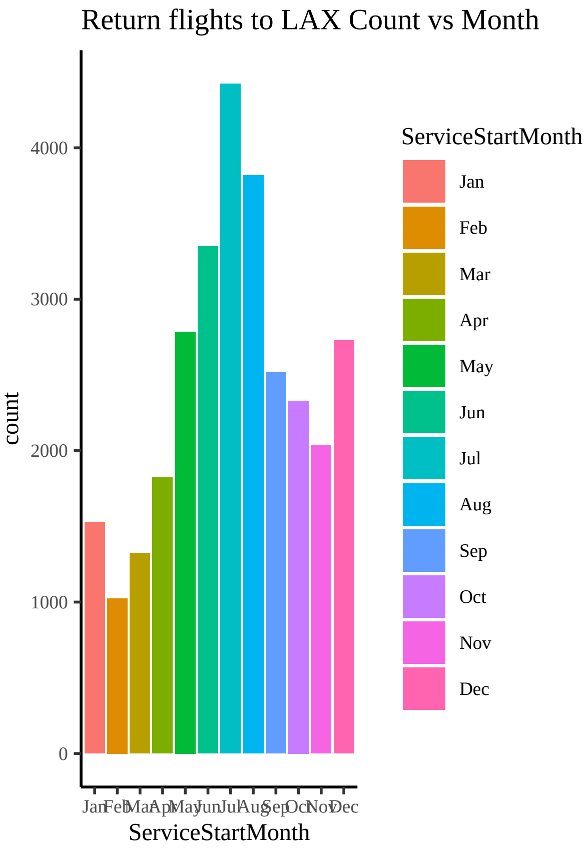


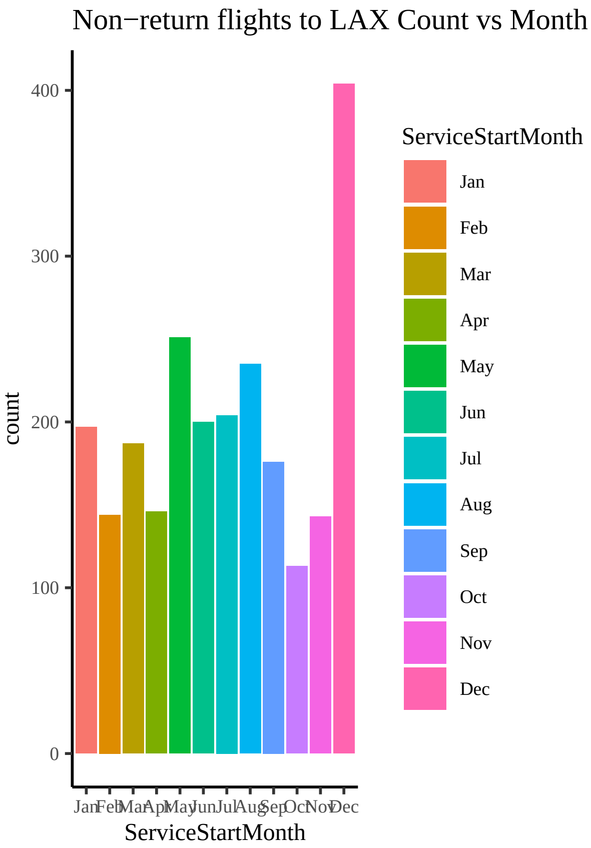


Interpretation There is an extremely high number of customers buying one-way tickets to LAX just before the flight compare to others. Sun Country can make some proper response to this situation to decrease the number of non-return flights in this period of time.

LAX - Seasonilioty of Travel

```
# Return ticket to lax

lm1 <- ggplot(sun_return_lax, aes(x=ServiceStartMonth, fill = ServiceStartMonth)) +
  geom_histogram(stat = 'count') +
  ggtitle('Return flights to LAX Count vs Month') +
  theme(legend.position = "none") + my_theme

# Single tickets to LAX
lm2 <- ggplot(sun_single_lax, aes(x = ServiceStartMonth, fill = ServiceStartMonth)) +
  geom_histogram(stat = 'count') +
  ggtitle('Non-return flights to LAX Count vs Month') +
  theme(legend.position = "none") + my_theme

grid.arrange(lm1,lm2,nrow=1)
```





Interpretation In the graph, we can see that in this group, customers flying to LAX with non-return tickets happen more in the summer time than another season. Notice that in December, there is also an extremely large amount of non-return flights happen. Sun Country should try to understand this abnormally and give out proper solutions.

Conclution To focus on this type of customers, Sun Country better target those young age customers. In this age, people are likely to have lower economic ability to afford. If the company can come up with some return packages with some favourable lower-cost service, young customers will be more likely to buy return tickets. Also, there are a considerable amount of people buying the tickets to LAX just before the flights. The reason the one-way ticket ratio is so high is probably because of the uncertainty of the trip, causing the lack of the motivation to but return tickets. Sun Country can offer more refundable return tickets with flexible return dates on this route to attract those customers.

Final Takeaways

What have we found?

Overview of Sun Country
We saw earlier that most of our customers travel from Minnesota to other places in the country. As winter is very harsh and summer is very pleasant in Minnesota, most customers are travelling out of Minnesota in winters and many customers traveling towards Minnesota during summers. This can be interpreted as most of the people who are travelling to and from Minnesota are for vacation purposes. *Summary of Customer Segmentation*
From our analysis, we observed that we have 3 segments of consumers

1. Business or Working class travelers who travel to cities and book ticket late.

2. Solo and family vacation travelers who pre-book tickets quite early.
3. Young travelers who travel to both business and vacations and book tickets sparsely.

Summary of Major Routes
We identified that business travelers who travel to major cities like JFK, LAX and RSW tend to purchase only a one-way ticket.

Final Recommendation

To consolidate our analysis, we know that UFly members are beneficial when compared to Non-UFly. Further we have identified 3 segments of customer and major cities of travel. Our Recommendation are:

1. For business travelers, we need to market with additional rewards like fast check-in with UFly and seat upgrade using the miles generated from the rewards program.
2. We need to market for round trip travel as most of the business travelers are not using Sun Country to return. This can be done by educating our sales reps in JFK, LAX, RSW and other major cities. Further, we can have negotiated with firms and be their travel partner which will boost our reputation and revenue.
3. For vacation travelers (families) we need to market our UFly program as a whole vacation package. This will encourage families to take membership and earn rewards. Further, we need to promote our miles project where travelers can use miles to book tickets or upgrade tickets.
4. Finally for solo travelers, we need to market based on ticket discount using UFly membership.

With these measures, it would help Sun Country to convert Non-Member to Members which will eventually yield better customer experience and revenue.