2018

Devising targeted marketing recommendations for a major airline using consumer insights

Table of contents:

Analysis overview

The key theme behind our approach is to design effective strategies to improve customer experience towards improving customer loyalty and profitability for Sun Country. The key avenues of improvement we that we are focusing on are as follows:

1. **Design customized marketing strategy**: As the starting point before getting into any specific problem, we have tried to identify natural clusters among customers to device relevant targeted marketing strategies. An unbiased evaluation of heterogeneous clusters is done to determine recommendations for improving profitability and customer experience.

2. **Improving customer loyalty program**: Providing UFly membership to potential loyalists will help us secure more long-term customers. The assumption is that there will be significant characteristic and behavior difference among UFly members which will be identified through clustering.

3. **Identification of high churn groups**: Identification of customer clusters and analyzing them to observe any specific characteristics leading to churn will help Sun Country in better retention and improving loyalty towards the brand.

4. **Invest in high return customer**: Historical data suggests that there has been under utilization of flight capacity and in such cases, customers were provided free upgrades. In this approach we are trying to identify the customers who can be better targeted as potential buyers for high revenue class bookings.

As a baseline, for each analysis and the business problem we are trying to solve, we assume that there are homogeneous clusters of customers with significant differences as compared other clusters that will lead us to relevant insights. We have identified a few variables which we want to focus on to extract information from these clusters (mentioned in each approach)

Data cleaning and transformation

Load dependencies

```
library(data.table)

library(dplyr)

library(lubridate)

library(ggplot2)

library(clustMixType)
```

Overview of the data

```
summary(sun)
```

```
##  PNRLocatorID         TicketNum              CouponSeqNbr
##  Length:3435388     Min.   :3372052115142   Min.   :1.000
```

```
##  Class :character   1st Qu.:3372107128282   1st Qu.:1.000
##  Mode  :character   Median :3372107783924   Median :1.000
##                     Mean   :3374303059408   Mean   :1.464
##                     3rd Qu.:3377293084751   3rd Qu.:2.000
##                     Max.   :3379578145804   Max.   :8.000
##
##  ServiceStartCity   ServiceEndCity     PNRCreateDate
##  Length:3435388     Length:3435388     Length:3435388
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  ServiceStartDate     PaxName          EncryptedName
##  Length:3435388     Length:3435388     Length:3435388
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##   GenderCode         birthdateid           Age
##  Length:3435388     Min.   :-675290   Min.   :-2883.00
##  Class :character   1st Qu.:  39620   1st Qu.:   26.00
##  Mode  :character   Median :  45088   Median :   40.00
##                     Mean   :  44982   Mean   :   40.05
##                     3rd Qu.:  50251   3rd Qu.:   55.00
##                     Max.   :1112840   Max.   : 2012.00
##                     NA's   :43999     NA's   :43999
##   PostalCode        BkdClassOfService  TrvldClassOfService
##  Length:3435388     Length:3435388     Length:3435388
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  BookingChannel      BaseFareAmt      TotalDocAmt      UFlyRewardsNumber
##  Length:3435388     Min.   :   0.0   Min.   :    0.0   Min.   :100000191
##  Class :character   1st Qu.: 174.9   1st Qu.:  189.8   1st Qu.:200860074
##  Mode  :character   Median : 273.2   Median :  302.2   Median :202968124
##                     Mean   : 287.7   Mean   :  314.9   Mean   :204242059
##                     3rd Qu.: 370.2   3rd Qu.:  414.6   3rd Qu.:210381861
##                     Max.   :4342.0   Max.   :17572.0   Max.   :241086274
##                                                        NA's   :2740908
##  UflyMemberStatus   CardHolder      BookedProduct       EnrollDate
##  Length:3435388     Mode :logical   Length:3435388     Length:3435388
##  Class :character   FALSE:659060    Class :character   Class :character
##  Mode  :character   TRUE :35420     Mode  :character   Mode  :character
```

```
##                     NA's :2740908
##
##
##
##  MarketingFlightNbr MarketingAirlineCode StopoverCode
##  Length:3435388     Length:3435388       Length:3435388
##  Class :character   Class :character     Class :character
##  Mode  :character   Mode  :character     Mode  :character
##
##
##
##
```

There are a few things that immediately jump out as potential issues from this summary. The PNRCreateDate and ServiceStartDate fields are character type, so they will need to be changed to Date to perform any date manipulations. The birthdateid field has a number of NAs, as well as some potential outlier values. The age column has negative values, impossibly large values, and the amount of NAs as birthdateid. There are a lot of NAs for the uFlyRewardsNumber field as well.

Change Date fields to Date Types

```
sun <- sun %>%
  mutate(PNRCreateDate = as.Date(PNRCreateDate)) %>%
  mutate(ServiceStartDate = as.Date(ServiceStartDate))
```

Handling NA values in the Age and birthdateid columns

```
nrow(sun %>% filter(is.na(Age) & is.na(birthdateid)))

## [1] 43999

sun <- sun %>%
  filter(!is.na(Age))
```

The rows in which Age and birthdateid are NA are all the same. There are 43,999 NA rows, which represents about 1% of the total dataset. There is no good way to impute this missing data, so it will be removed, and it will be assumed that this 1% will not affect the analysis.

Handling Impossible Ages

```
nrow(sun %>% filter(Age > 100 | Age<0))

## [1] 1650

sun <- sun %>%
  mutate(Age = replace(Age, Age < 0 | Age > 100, median(Age)))
```

As noted before, there are ages less than 0 and well above 100. Most of these are not possible, and will be replaced with the median age. Replacing with the median age could affect analysis as it reduces the variance, but there are only 1650 rows so the affect should be minimal.

Handling Missing UflyMemberStatus

```
sun <- sun %>%
  mutate(UflyMemberStatus = replace(UflyMemberStatus, UflyMemberStatus == "",
"Non_Member"))

sun %>%
  distinct(UflyMemberStatus) %>%
  select(UflyMemberStatus)

## # A tibble: 3 x 1
##   UflyMemberStatus
##   <chr>
## 1 Non_Member
## 2 Standard
## 3 Elite
```

We are assuming that a missing UflyMemberStatus signifies that they are not a member. This missing field will be coded as "Non_Member" in order to use that status as a factor in the analysis.

Filtering MarketingAirlineCode

```
table(sun$MarketingAirlineCode)

##
##      DE      F9      FI      HA      SY
##       1    3295      13    1705 3386375

sun <- sun %>%
  filter(MarketingAirlineCode == "SY") %>%
  filter(GenderCode != "") %>%
  select(-MarketingAirlineCode)
```

As seen in the above table, there is data for non Sun Country flights. This data is not useful for analysis, so it will be removed.

Combining Booking Channels

```
nrow(sun %>%
  distinct(BookingChannel) %>%
  select(BookingChannel))

## [1] 29

table(sun$BookingChannel)

##
##                 ANC                 BOS                 DCA
##                   9                   1                  24
##                 DFW                 FCM                 GJT
##                 521                3798                   1
##                 HRL                 JFK                 LAN
##                  18                 256                 269
```

```
##                 LAS                 LAX                 MCO
##                 179                 414                  17
##                 MDW                 MIA                 MKE
##                 204                   1                 262
##                 MSN                 MSP     Outside Booking
##                   1                4942             1467237
##                 PHX                 PSP Reservations Booking
##                  21                  28              226321
##                 RSW  SCA Website Booking                 SEA
##                  91             1454454                  42
##                 SFO          SY Vacation Tour Operator Portal
##                 141               93609              132889
##                 UFO                 XTM
##                 149                 476
```

```
sun <- sun %>%
  mutate(BookingChannel = replace(BookingChannel,
                                  !(BookingChannel %in% c("Reservations Booki
ng", "Outside Booking",
                                                          "SY Vaction", "Tour
Operator Portal",
                                                          "SCA Website Bookin
g")),
                                  "Other Booking"))
sun %>%
  distinct(BookingChannel) %>%
  select(BookingChannel)
```

```
## # A tibble: 5 x 1
##   BookingChannel
##   <chr>
## 1 Outside Booking
## 2 SCA Website Booking
## 3 Reservations Booking
## 4 Other Booking
## 5 Tour Operator Portal
```

There are 29 distinct booking channels in the data, many of which have a relatively small number of occurences, as seen in the table above. These channels will be combined into an "Other Booking" group to simplify this variable.

Replace Missing CardHolder Values

```
sun <- sun %>%
  mutate(CardHolder = replace(CardHolder, is.na(CardHolder), FALSE))
```

We will assume that missing values for the CardHolder variable mean that the customer does not have a card.

Problems with PNRLocatorID and CouponSeqNbr

```
# Example PNR ID that does not have a CouponSeqNbr of 1
sun %>%
  select(PNRLocatorID, CouponSeqNbr) %>%
  filter(PNRLocatorID == "AADOFC")

## # A tibble: 1 x 2
##   PNRLocatorID CouponSeqNbr
##   <chr>               <int>
## 1 AADOFC                  2

bad_pnr <- sun %>%
  select(PNRLocatorID, CouponSeqNbr) %>%
  group_by(PNRLocatorID) %>%
  summarise(min_coup = min(CouponSeqNbr)) %>%
  filter(min_coup != 1)

sun_good_pnr <- sun %>%
  filter(!(PNRLocatorID %in% bad_pnr$PNRLocatorID))
```

As seen in the output above, there are flights that do not have a CouponSeqNbr of 1. This means that the data was entered incorrectly, or the data for that flight sequence is incomplete. These rows will be removed from the data.

Removing Duplicate PNRs

```
sun_distinct_pnr <- sun_good_pnr %>%
  distinct(PNRLocatorID, CouponSeqNbr, PaxName, ServiceStartCity, ServiceEndC
ity, ServiceStartDate,
           .keep_all = TRUE)
```

There are some duplicate rows in the data. Only the distinct rows will be kept.

Time Booked In Advance

```
sun <- sun %>%
  mutate(days_booked_advance = as.numeric(ServiceStartDate - PNRCreateDate))
```

This variable measures how many days a customer booked in advance. This behavior may differentiate customers into different groups based on purchasing behavior. Knowing what types of customers are more likely to book in advance compared to last minute could help Sun Country tailor their advertising and marketing.

Seasonality

```
sun <- sun %>%
  mutate(quarter = if_else(month(ServiceStartDate) %in% c(1,2,3), "Q1",
                           if_else(month(ServiceStartDate) %in% c(4,5,6), "Q2
",
                                   if_else(month(ServiceStartDate) %in% c(7,8
,9), "Q3",
                                           if_else(month(ServiceStartDate) %i
n% c(10,11,12), "Q4", NA_character_)))))
```

Another variable that may differentiate customer behavior is when they are flying. We have decided to group the data into quarters. Since the data is over the course of two years, this variable will not capture the differences between quarters in different years.

Group Size

```
sun <- sun %>%
  group_by(PNRLocatorID) %>%
  summarise(group_size = n_distinct(PaxName)) %>%
  right_join(sun, by="PNRLocatorID")
```

Group size is a third variable that may differentiate customers. For example, a group size of 4 may indicate a family, while a group size of 2 may indicate a couple. This variable can be used in clustering to identify customer segments.

Customer Upgrades

```
sun <- sun %>%
   mutate(upgraded = if_else(TrvldClassOfService != BkdClassOfService, "Yes",
"No"))
```

Sun Country allows customers to upgrade from Coach to First Class after they have booked the flight. Finding the characteristics of customers who are more likely to upgrade can aid in upselling and sending offers.

What Groups of Customers Does Sun Country Have?

What characteristics Does Each Group Have?

Marketing to customers can be significantly improved by knowing who your customers are, if there are different types of customers, and what makes your customers different. To accomplish this, the data will be aggregated to the customer level and k prototypes clustering will be ran to determine the different customer groups. Once these groups are known, we can determine the key differences between the groups and develop strategies to market towards each group effectively.

Removing Irrelevant Variables and Rows

This analysis will not take into account where customers flying, so the dataset will be filtered down to just the first leg of a flight sequence. This is done so that customers who have connecting flights are not counted more than customers who take direct flights. Then, the variables not used for aggregating will be removed.

```
sun <- sun %>%
  filter(CouponSeqNbr == 1) %>%
  select(-c(CouponSeqNbr, TicketNum, BkdClassOfService, TrvldClassOfService,
            EnrollDate, BookingChannel, BookedProduct, StopoverCode, Marketin
gFlightNbr,
            PostalCode, EncryptedName, ServiceStartCity, ServiceEndCity, PNRC
reateDate,
            ServiceStartDate, UFlyRewardsNumber))
```

Aggregate the Flight Data to the Customer Level

Currently the sun data is at the flight level. This means the same customer can be seen in multiple rows. The data will be aggregated so that one row equals one customer. Various summary statistics for each customer will be computed.

```
sun_cust <- sun %>%
  group_by(PaxName, birthdateid) %>%
  summarize(trips_taken = n_distinct(PNRLocatorID),
            avg_group_size = mean(group_size),
            gender = min(GenderCode),
            age = mean(Age),
            times_upgraded = sum(upgraded == "Yes"),
            ufly_member_status = min(UflyMemberStatus),
            is_card_holder = sum(CardHolder),
            avg_days_booked_advance = mean(days_booked_advance),
            avg_base_fare = mean(BaseFareAmt),
            q1_count = sum(quarter == "Q1"),
            q2_count = sum(quarter == "Q2"),
            q3_count = sum(quarter == "Q3"),
            q4_count = sum(quarter == "Q4")) %>%
  mutate(is_card_holder = if_else(is_card_holder > 0, "Yes", "No")) %>%
  ungroup() %>%
  select(trips_taken, avg_group_size, age, times_upgraded, avg_days_booked_ad
vance, avg_base_fare,
         q1_count, q2_count, q3_count, q4_count, gender, ufly_member_status,
is_card_holder)
glimpse(sun_cust)
```

```
## Observations: 1,321,690
## Variables: 13
## $ trips_taken             <int> 1, 1, 1, 1, 2, 1, 2, 1, 1, 1, 1, 1, 1,...
## $ avg_group_size          <dbl> 2.0, 6.0, 1.0, 2.0, 2.5, 2.0, 1.5, 4.0...
## $ age                     <dbl> 33.0, 52.0, 29.0, 50.0, 41.5, 41.0, 47...
## $ times_upgraded          <int> 0, 0, 1, 0, 0, 1, 2, 0, 1, 0, 0, 0, 0,...
## $ avg_days_booked_advance <dbl> 7.0, 9.0, 9.0, 0.0, 22.5, 71.0, 103.5,...
## $ avg_base_fare           <dbl> 151.630, 0.000, 432.560, 264.190, 44.6...
## $ q1_count                <int> 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0,...
## $ q2_count                <int> 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0,...
## $ q3_count                <int> 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1,...
## $ q4_count                <int> 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,...
## $ gender                  <chr> "M", "M", "M", "M", "F", "M", "F", "M"...
## $ ufly_member_status      <chr> "Non_Member", "Non_Member", "Standard"...
## $ is_card_holder          <chr> "No", "No", "No", "No", "No", "No", "N...
```

Prepare Features for Clustering

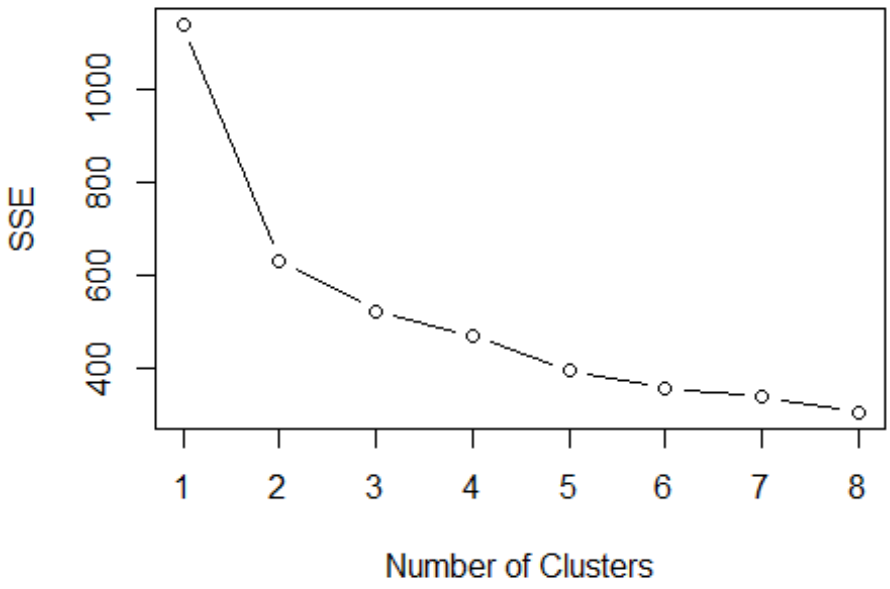
To conduct K Prototypes clustering, the numeric features should be normalized in order to avoid a single variable with a larger scale dominating the analysis. We will use min-max scaling here. In addition, the categorical variables will be converted to factors.

```
sun_cust <- as_tibble(fread("SunCountry_cust.csv"))
 sun_cust_normalized <- sun_cust

 sun_cust_normalized$gender <- as.factor(sun_cust_normalized$gender)
 sun_cust_normalized$ufly_member_status <- as.factor(sun_cust_normalized$ufly
_member_status)
 sun_cust_normalized$is_card_holder <- as.factor(sun_cust_normalized$is_card_
holder)

# min-max scaling function
normalize <- function(x){
   return ((x - min(x))/(max(x) - min(x)))
}
sun_cust_normalized[1:10] <- lapply(sun_cust_normalized[1:10], normalize)
```

Determine Number of Clusters

To determine the number of clusters, an elbow plot will be constructed. Since the dataset is large, a sample has been taken. After determining the number of clusters on the sample, the final clustering can be done on the full dataset.

```
set.seed(6)
sun_cust_sample <- as.data.frame(sample_n(sun_cust_normalized, 20000))

# will try 1 to 8 clusters and take the average SSE of 5 different solutions
for each number of clusters
# this is done as there will be a random starting point in the k prototypes a
lgorithm
SSE_curve <- c()
for (k in 1:8){
  temp_sse <- c()
  for (i in 1:5){
    kpro <- kproto(as.data.frame(sun_cust_sample), k)
    sse <- sum(kpro$withinss)
    temp_sse[i] <- sse
  }
  SSE_curve[k] <- mean(temp_sse)
}
plot(1:8, SSE_curve, type="b", xlab="Number of Clusters", ylab="SSE")
```



Based on the elbow plot, anywhere from 3 to 5 clusters would be reasonable. This analysis will proceed with five clusters.

Run K Prototypes

The K prototypes algorithm will be run on the normalized data with k equal to 5. After clustering, the original data will be given a cluster membership column in order to visualize the different clusters.

```
num_clusters <- 5
kpro <- kproto(as.data.frame(sun_cust_normalized), num_clusters)

sun_cust$cluster <- kpro$cluster

sun_cust <- as_tibble(fread("SunCountry_cust_cluster.csv"))
sun_cust$cluster <- as.factor(sun_cust$cluster)
```

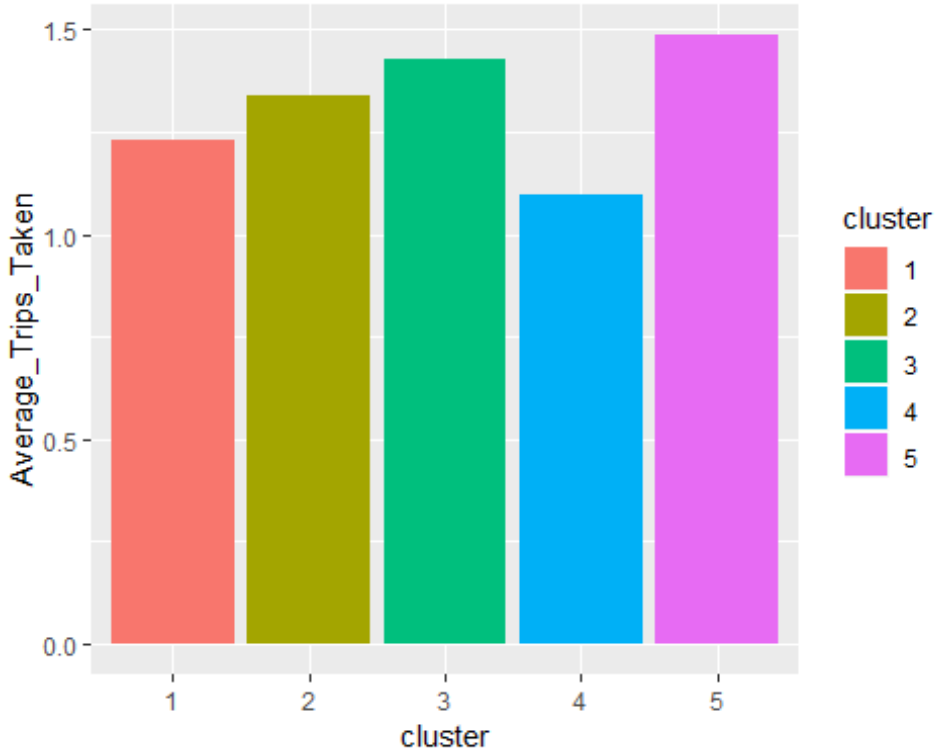
How Big Are the Clusters?

```
round(prop.table(table(sun_cust$cluster)) * 100, 2)

##
##     1     2     3     4     5
## 16.32 30.97 26.38  0.62 25.71
```

We can see that clusters 1, 2, 3 and 5 make up the majority of our customers. Cluster 4 is very small, but may represent a significant type customer that can be targeted differently.

Which Cluster Takes More Trips?

```
sun_cust %>%
  group_by(cluster) %>%
  summarize(Average_Trips_Taken = mean(trips_taken)) %>%
  ggplot(aes(x=cluster, y=Average_Trips_Taken, fill = cluster)) +
  geom_bar(stat = "identity")
```

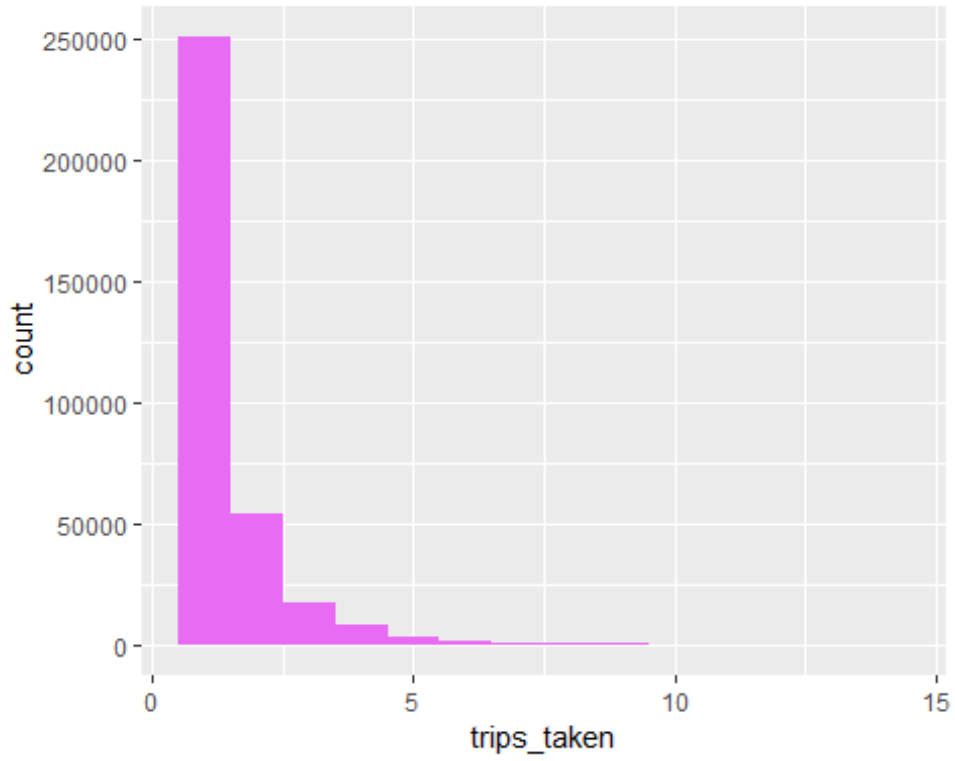


We can see that cluster 4 takes the least number of trips, while cluster 5 takes the most.

How are Cluster 5's Trips Distributed?

Let's look at the histogram for cluster 5. Number of trips is restricted as there are very few data points past 20.

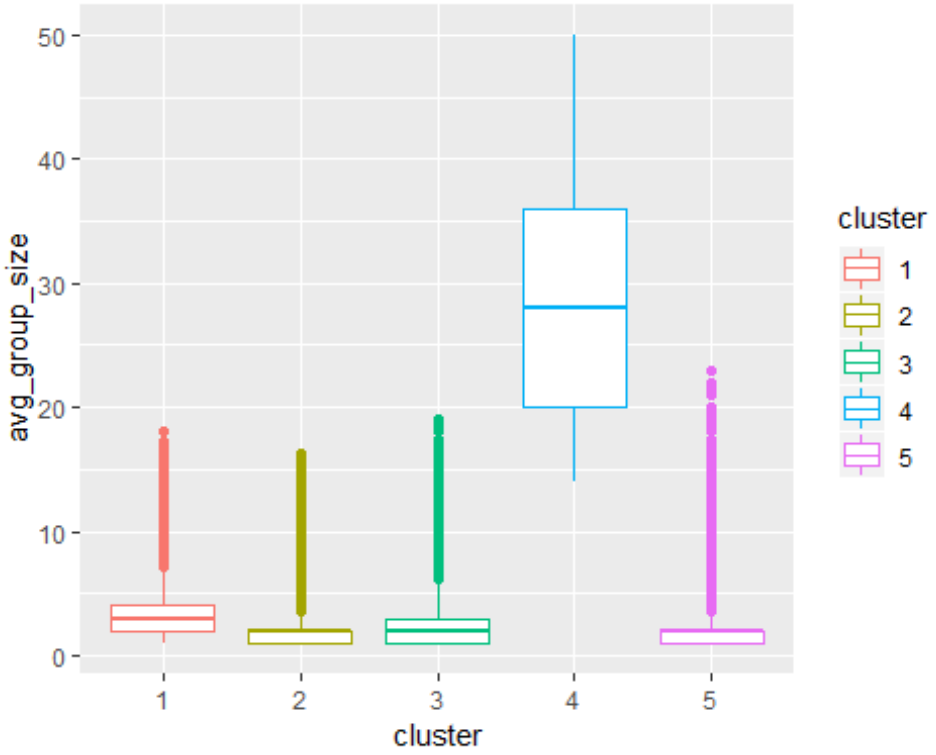
```
sun_cust %>%
  filter(cluster == "5", trips_taken < 15) %>%
  ggplot(aes(x=trips_taken)) + geom_histogram(binwidth = 1, fill="#E76BF3")
```



Based on the graph, it appears that the group with largest number of trips, on average, is still mostly distributed at 1 trip.

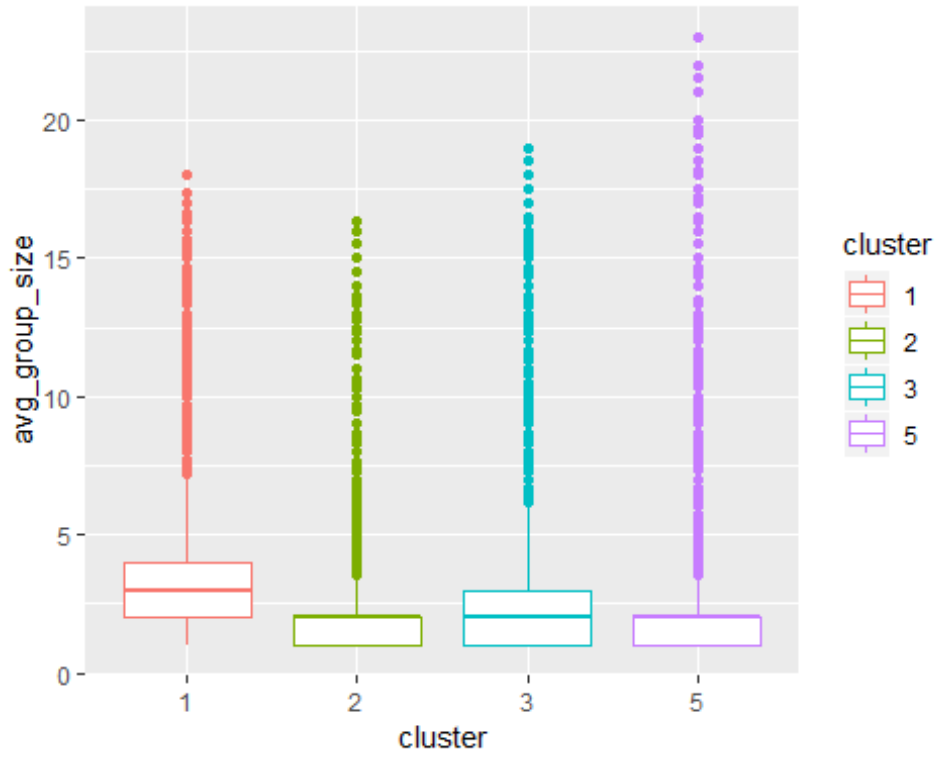
What are the Cluster's Group Sizes?

```
sun_cust %>%
  ggplot(aes(x=cluster, y=avg_group_size, col=cluster)) + geom_boxplot()
```



It looks like cluster 4 is composed of large groups. Let's look at the clusters separately.

```
sun_cust %>%
  filter(cluster != "4") %>%
  ggplot(aes(x=cluster, y=avg_group_size, col=cluster)) + geom_boxplot()
```



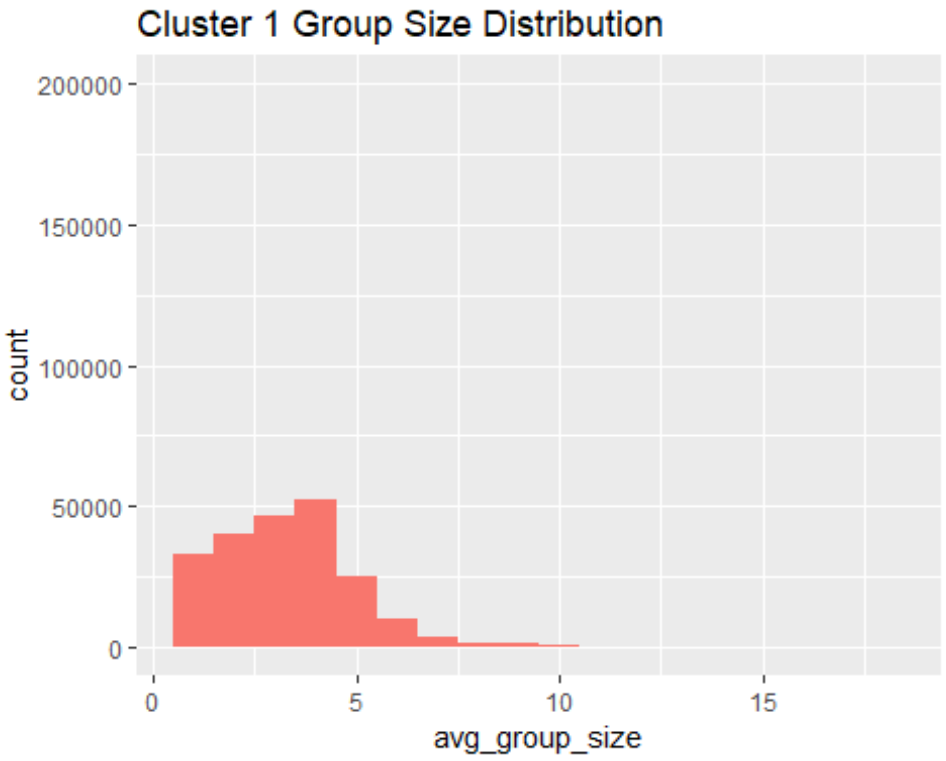

```
sun_cust %>%
  filter(cluster != "4") %>%
  group_by(cluster) %>%
  summarise(median_group_size = median(avg_group_size))
```

```
## # A tibble: 4 x 2
##   cluster median_group_size
##   <fct>               <dbl>
## 1 1                       3
## 2 2                       2
## 3 3                       2
## 4 5                       2
```

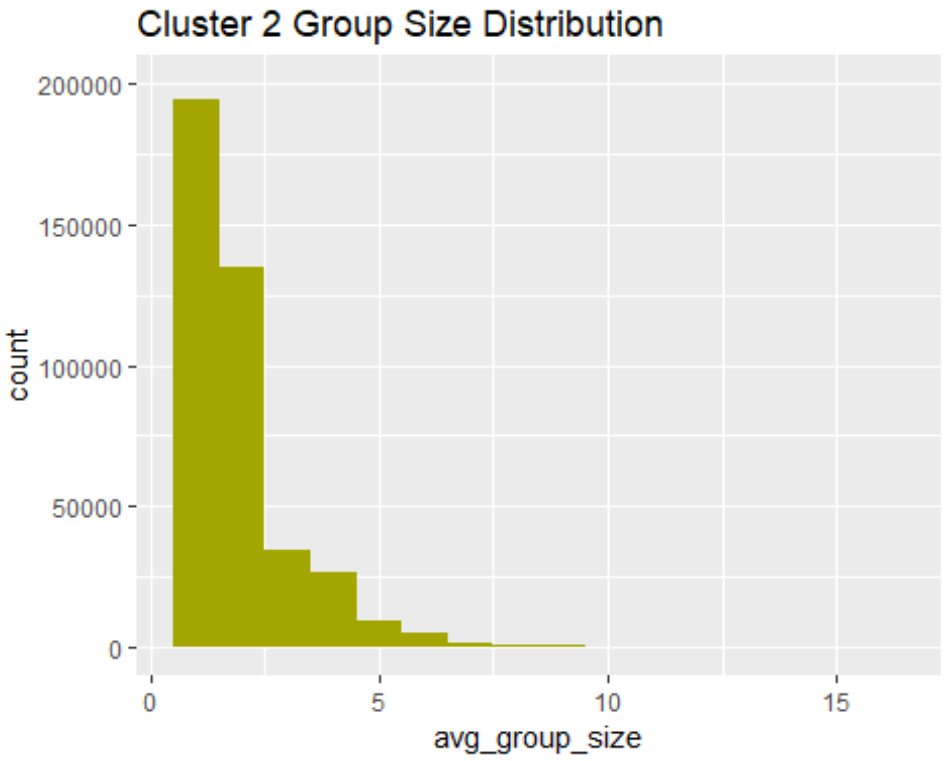
Cluster 1 looks to be composed of smaller groups and could potentially be a family travel segment. Clusters 2 and 5 seem to be composed of mostly of groups of size 1 and 2, while cluster 3 incorporates some groups. Let's Look at the distributions of the clusters.

Distribution of Group Size

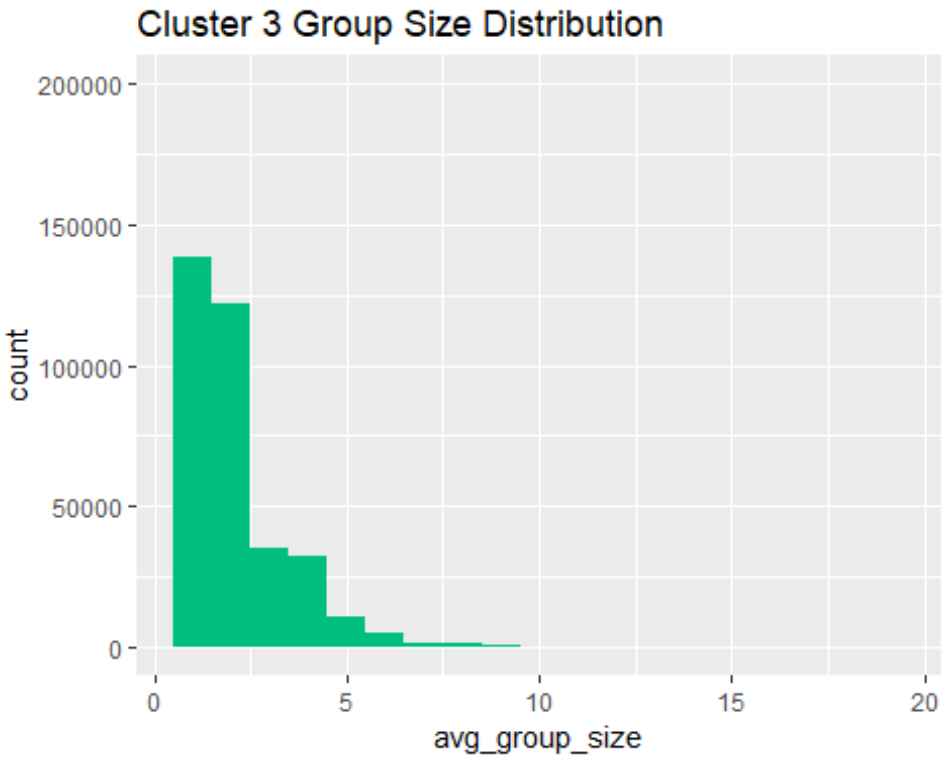
```
sun_cust %>%
  filter(cluster == "1") %>%
  ggplot(aes(x=avg_group_size)) +
  geom_histogram(binwidth = 1, fill="#F8766D") +
  labs(title="Cluster 1 Group Size Distribution") +
  ylim(0, 200000)
```



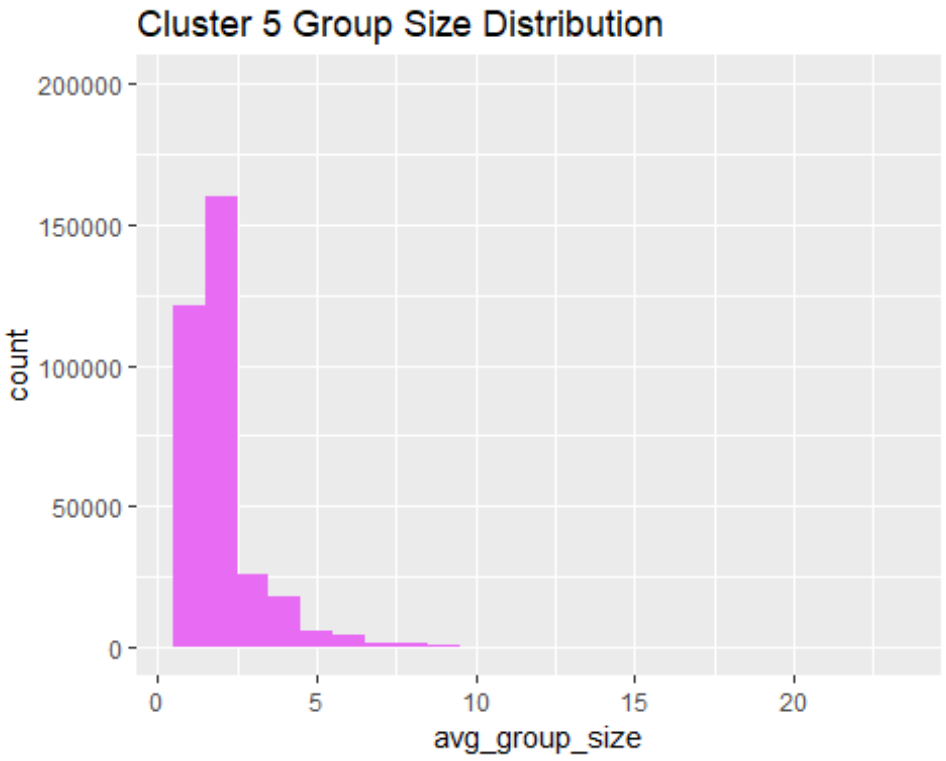

```
sun_cust %>%
  filter(cluster == "2") %>%
  ggplot(aes(x=avg_group_size)) +
  geom_histogram(binwidth = 1,fill="#A3A500") +
  labs(title="Cluster 2 Group Size Distribution") +
  ylim(0, 200000)
```




```
sun_cust %>%
  filter(cluster == "3") %>%
  ggplot(aes(x=avg_group_size)) +
  geom_histogram(binwidth = 1, fill="#00BF7D") +
  labs(title="Cluster 3 Group Size Distribution") +
  ylim(0, 200000)
```



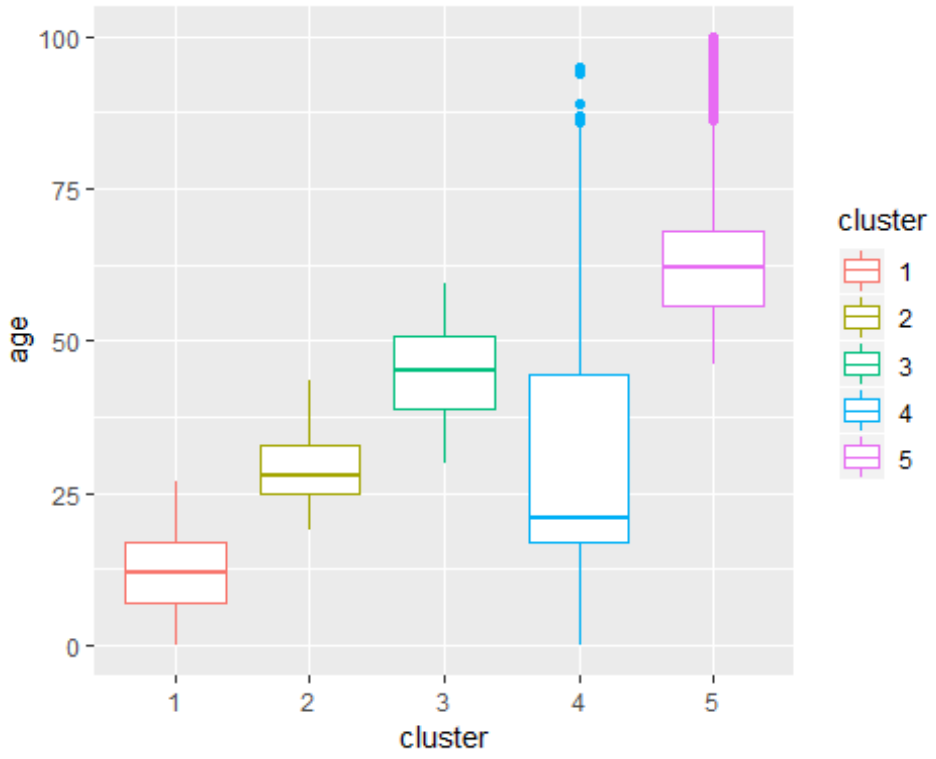

```
sun_cust %>%
  filter(cluster == "5") %>%
  ggplot(aes(x=avg_group_size)) +
  geom_histogram(binwidth = 1, fill="#E76BF3") +
  labs(title="Cluster 5 Group Size Distribution") +
  ylim(0, 200000)
```



As suspected, cluster 1 has a lot of groups of size 3 and 4. Cluster 2 has more single travelers than couples. Cluster 3 has about equal number of single and couple travelers, and cluster 5 appears to be mostly couples, with some single travelers.

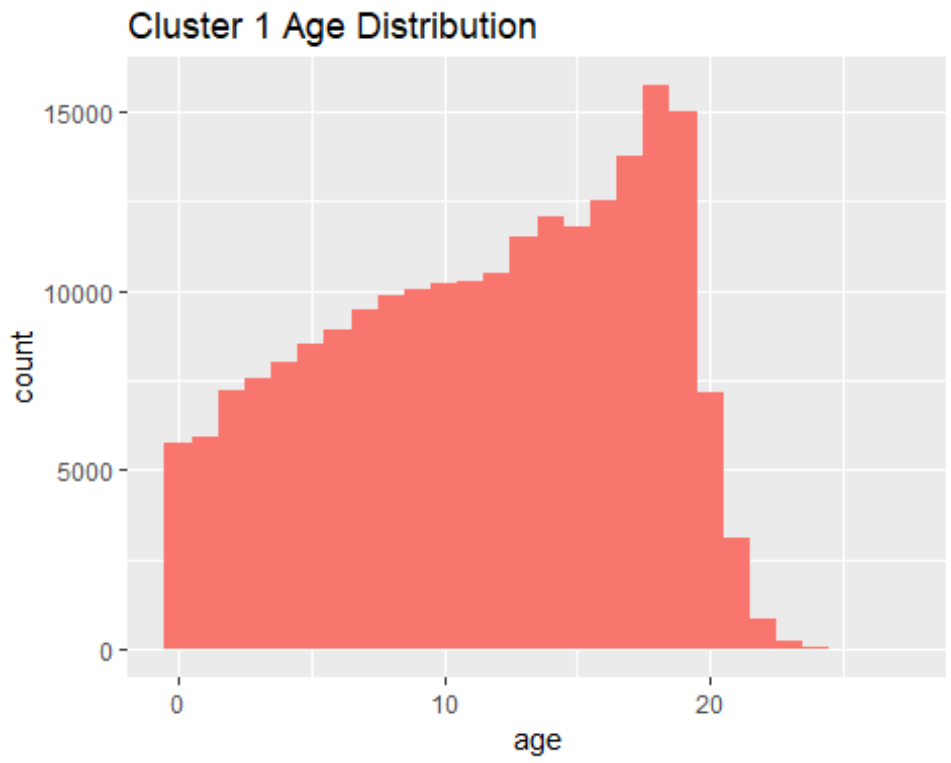
How Do the Ages of the Clusters Differ?

```
sun_cust %>%
  ggplot(aes(x=cluster, y=age, col=cluster)) + geom_boxplot()
```

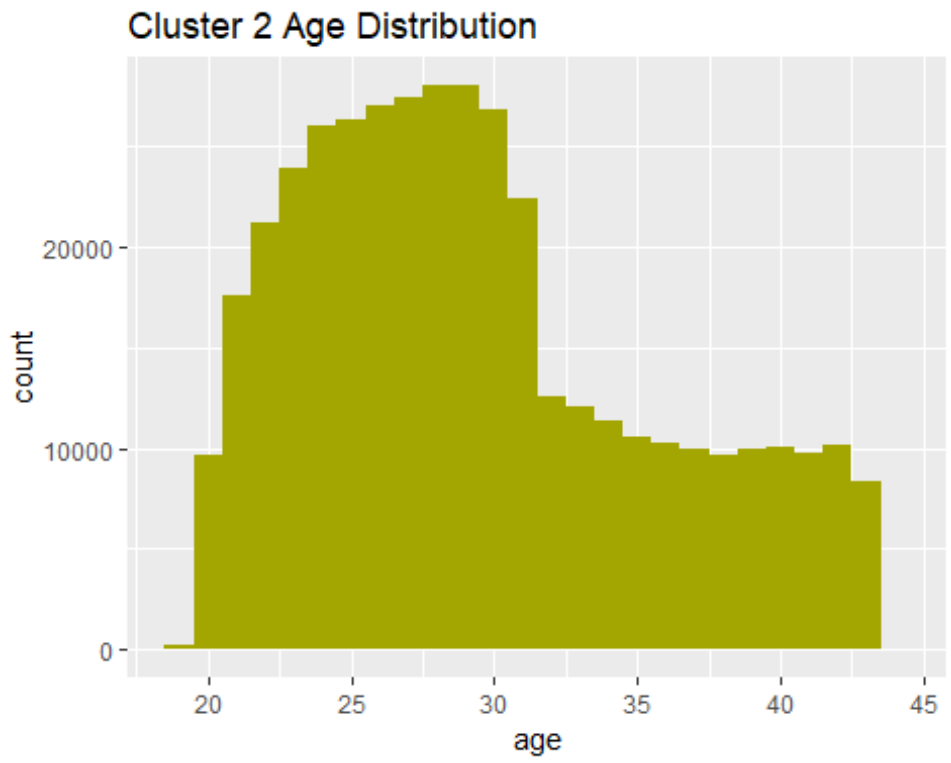


There is a clear distinction in groups by age. Cluster 1, the suspected family cluster, has a median age of 12. This would make sense as there would be a lot of children in a family cluster. The distribution should show this. Cluster 2 looks to be young adults. Cluster 5 appears to be older couples, while cluster 3 could be composed of single middle aged business travelers as well as middle aged couples.

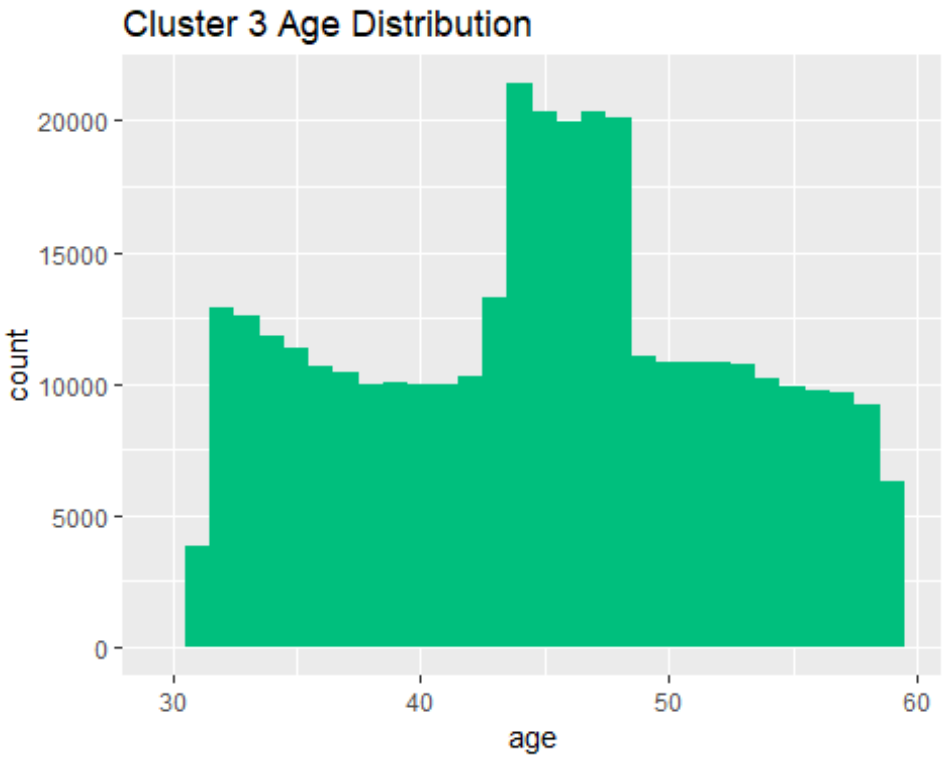
```
sun_cust %>%
  filter(cluster == "1") %>%
  ggplot(aes(x=age)) + geom_histogram(binwidth = 1, fill="#F8766D") + labs(ti
tle="Cluster 1 Age Distribution")
```



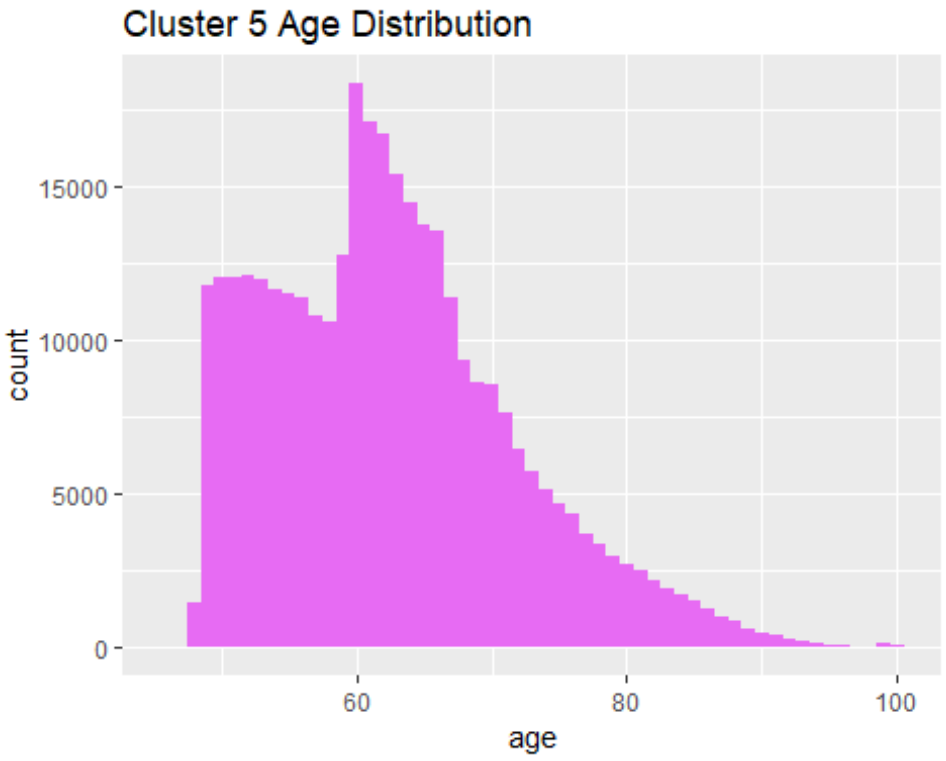

```
sun_cust %>%
  filter(cluster == "2") %>%
  ggplot(aes(x=age)) + geom_histogram(binwidth = 1, fill="#A3A500") + labs(ti
tle="Cluster 2 Age Distribution")
```




```
sun_cust %>%
  filter(cluster == "3") %>%
  ggplot(aes(x=age)) + geom_histogram(binwidth = 1, fill="#00BF7D") + labs(ti
tle="Cluster 3 Age Distribution")
```



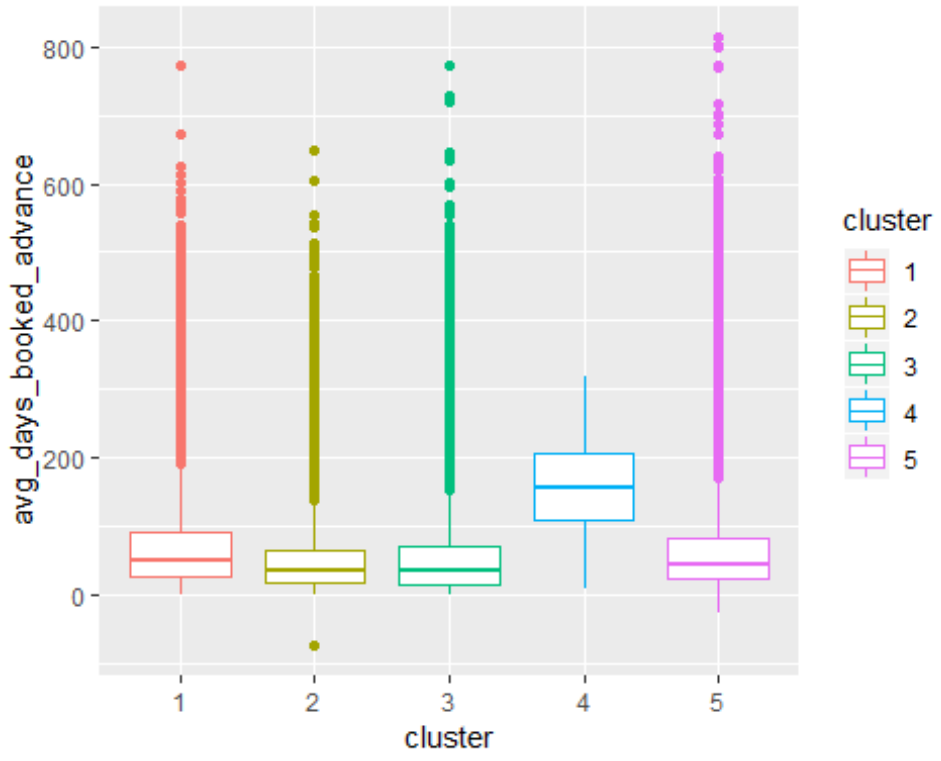

```
sun_cust %>%
  filter(cluster == "5") %>%
  ggplot(aes(x=age)) + geom_histogram(binwidth = 1, fill="#E76BF3") + labs(ti
tle="Cluster 5 Age Distribution")
```



It appears that cluster 1 has picked up the child travelers, but the parents are in a different segment. Cluster 2 is composed primarily of young adults. The age cuts off quickly after 31. Cluster 3 appears to fill in the gaps between cluster 2 and cluster 5. Cluster 5 is composed of primarily people 60 and up. Since clusters 2 and 3 have about equal number of groups of size 3 and 4, it appears that the parents are distributed across these two clusters.

How Far in Advance do Customers Book Flights?

```
sun_cust %>%
  ggplot(aes(x=cluster, y=avg_days_booked_advance, col=cluster)) + geom_boxpl
ot()
```



Not suprisingly, cluster 4, the large group cluster, books well in advance. Let's separate the other clusters to see how they differ.

```
sun_cust %>%
  group_by(cluster) %>%
  summarise(median_days_booked_in_advance = median(avg_days_booked_advance))
```

```
## # A tibble: 5 x 2
##   cluster median_days_booked_in_advance
##   <fct>                           <dbl>
## 1 1                                51
## 2 2                                36
## 3 3                                35.5
## 4 4                               155
## 5 5                                45
```

Cluster 1 books more in advance than the other clusters. Since this is the child cluster, with the parents scattered throughout the others, this may be a way to filter out the parents from clusters 2 and 3 as they generally book trips 2 weeks after cluster 1. It also appears that older couples book in advance.

How Much Does Each Cluster Pay?

```
sun_cust %>%
  group_by(cluster) %>%
  summarise(sum_fare = sum(avg_base_fare),
            num_group = n()) %>%
  mutate(pct_fare = round(sum_fare / sum(sum_fare),3)*100,
```

```
         pct_group = round(num_group / sum(num_group),3)*100,
         difference = pct_fare - pct_group) %>%
  select(cluster, pct_fare, pct_group, difference)
```

```
## # A tibble: 5 x 4
##   cluster pct_fare pct_group difference
##   <fct>      <dbl>     <dbl>      <dbl>
## 1 1           17        16.3      0.700
## 2 2           28.6      31       -2.4
## 3 3           27.5      26.4      1.1
## 4 4            0.8       0.6      0.2
## 5 5           26        25.7      0.3
```

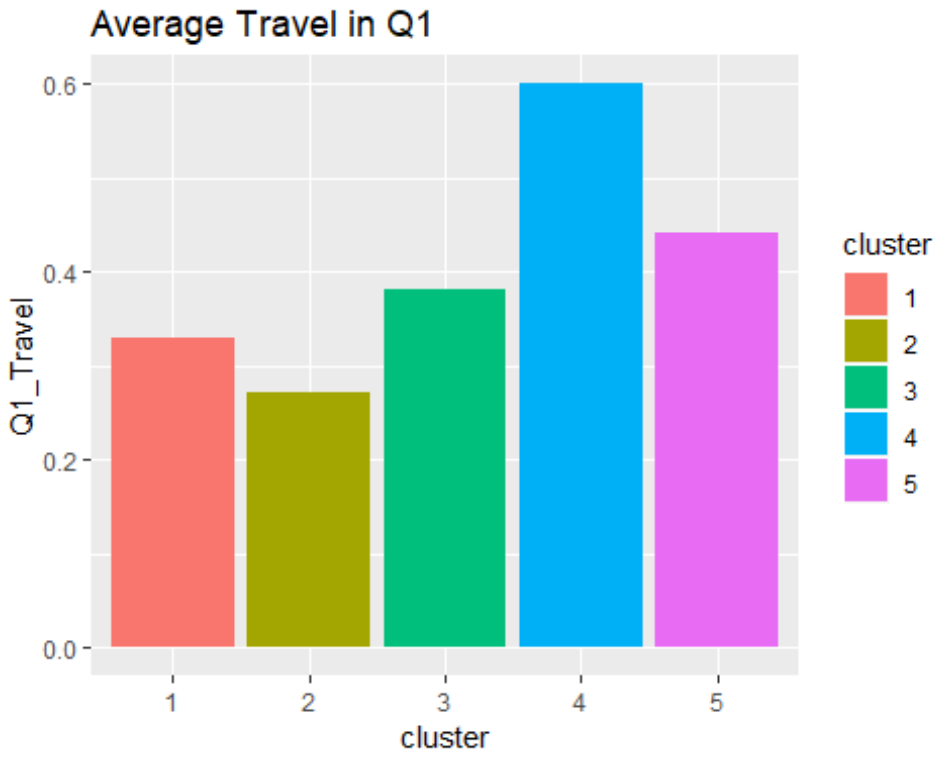
The above table compares what percent of the total fares each cluster makes up as well as what percent of the customer base they make up. Cluster 2 makes up 28.6% of the fares, however they are 31.0% of customers. This could mean that cluster 2 is more price sensitive and waits for sales and deals before purchasing. Conversely, cluster 3 appears to be a bit less price sensitive.

When Does Each Cluster Travel?

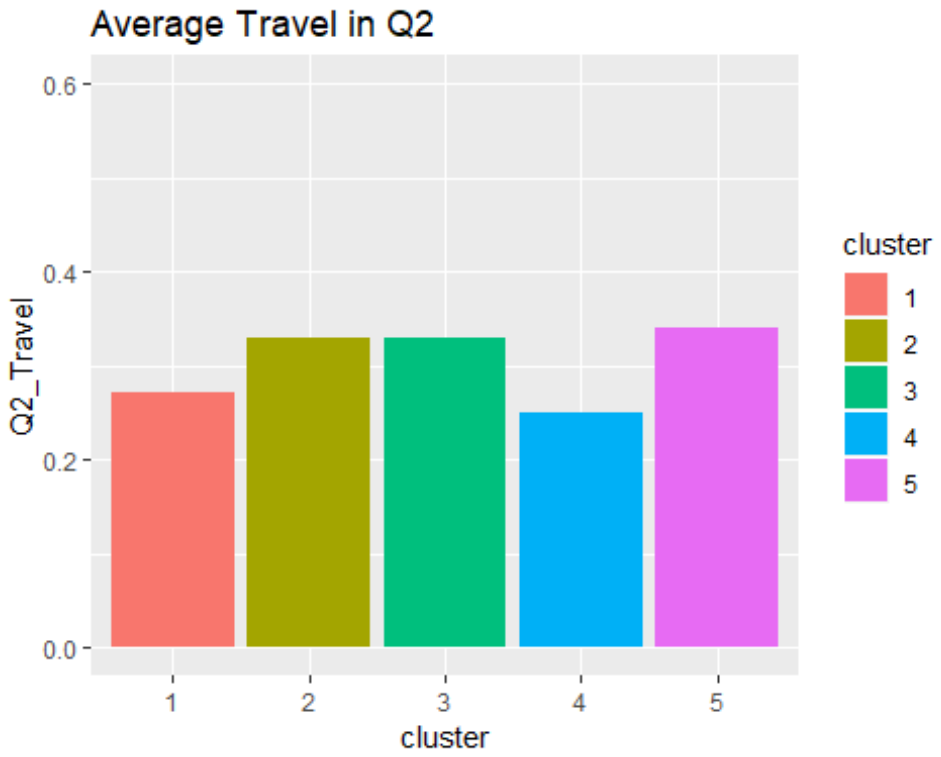
```
sun_cust %>%
  group_by(cluster) %>%
  summarise(Q1_Travel = round(mean(q1_count),2),
            Q2_Travel = round(mean(q2_count),2),
            Q3_Travel = round(mean(q3_count),2),
            Q4_Travel = round(mean(q4_count),2))
```

```
## # A tibble: 5 x 5
##   cluster Q1_Travel Q2_Travel Q3_Travel Q4_Travel
##   <fct>       <dbl>     <dbl>     <dbl>     <dbl>
## 1 1            0.33      0.27      0.34     0.290
## 2 2            0.27      0.33      0.37     0.37
## 3 3            0.38      0.33      0.36     0.37
## 4 4            0.6       0.25      0.15     0.1
## 5 5            0.44      0.34      0.32     0.4
```

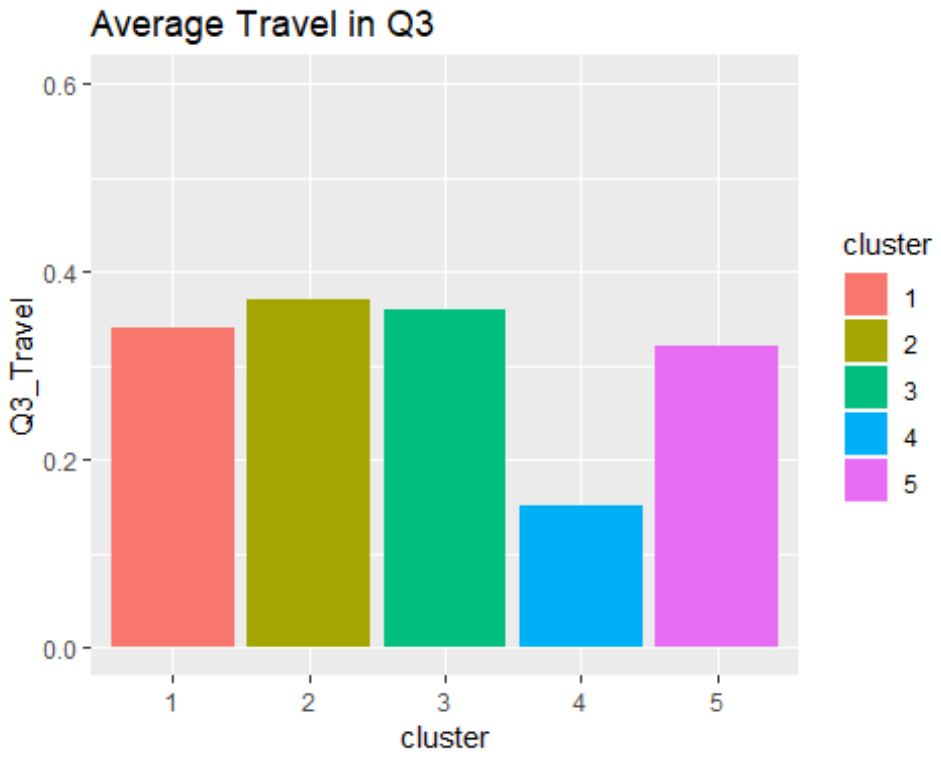
```
sun_cust %>%
  group_by(cluster) %>%
  summarise(Q1_Travel = round(mean(q1_count),2)) %>%
  ggplot(aes(x = cluster, y = Q1_Travel, fill = cluster)) +
  geom_bar(stat = "identity") +
  labs(title = "Average Travel in Q1") +
  ylim(0, 0.6)
```



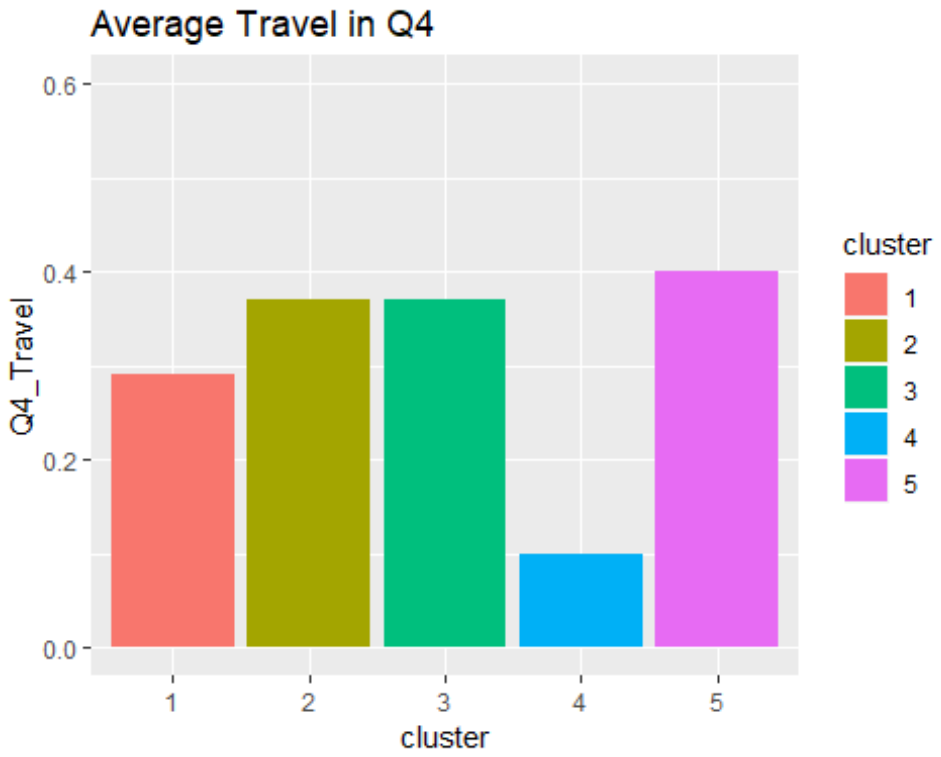

```
sun_cust %>%
  group_by(cluster) %>%
  summarise(Q2_Travel = round(mean(q2_count),2)) %>%
  ggplot(aes(x = cluster, y = Q2_Travel, fill = cluster)) +
  geom_bar(stat = "identity") +
  labs(title = "Average Travel in Q2") +
  ylim(0, 0.6)
```




```r
sun_cust %>%
  group_by(cluster) %>%
  summarise(Q3_Travel = round(mean(q3_count),2)) %>%
  ggplot(aes(x = cluster, y = Q3_Travel, fill = cluster)) +
  geom_bar(stat = "identity") +
  labs(title = "Average Travel in Q3") +
  ylim(0, 0.6)
```



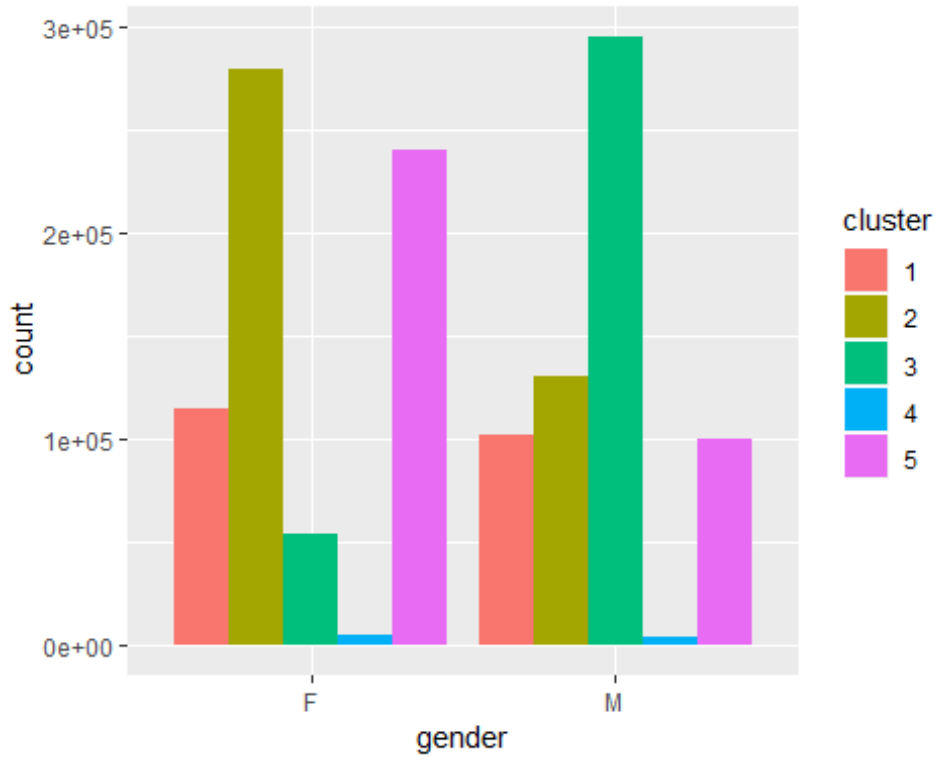

```
sun_cust %>%
  group_by(cluster) %>%
  summarise(Q4_Travel = round(mean(q4_count),2)) %>%
  ggplot(aes(x = cluster, y = Q4_Travel, fill = cluster)) +
  geom_bar(stat = "identity") +
  labs(title = "Average Travel in Q4") +
  ylim(0, 0.6)
```



The above table and graphs compare when each cluster travels by quarter. Cluster 1 travels more heavily during Q1 and Q3. This makes sense as Q1 encompasses spring break and Q3 encompasses the summer. Cluster 2, the young adults, travels much less in Q1 as compared to the other three. Cluster 3 travels less in Q2 but has overall higher rates across all the clusters. This cluster could be composed of business travelers who consistently travel year-round. Cluster 4, the large group cluster, primarily travels in Q1 and rarely ever travels in Q3 and Q4. Cluster 5, the older couples cluster, travels most frequently in Q1 and Q4, but like cluster 3 has consistently higher rates across all quarters. Older people who are retired have more free time to travel, so the consistently higher rates make sense.

Which Gender is Each Cluster Composed Of?

```
sun_cust %>%
  filter(gender %in% c("M","F")) %>%
  ggplot(aes(x=gender)) +
  geom_bar(aes(fill = cluster), position = "dodge")
```

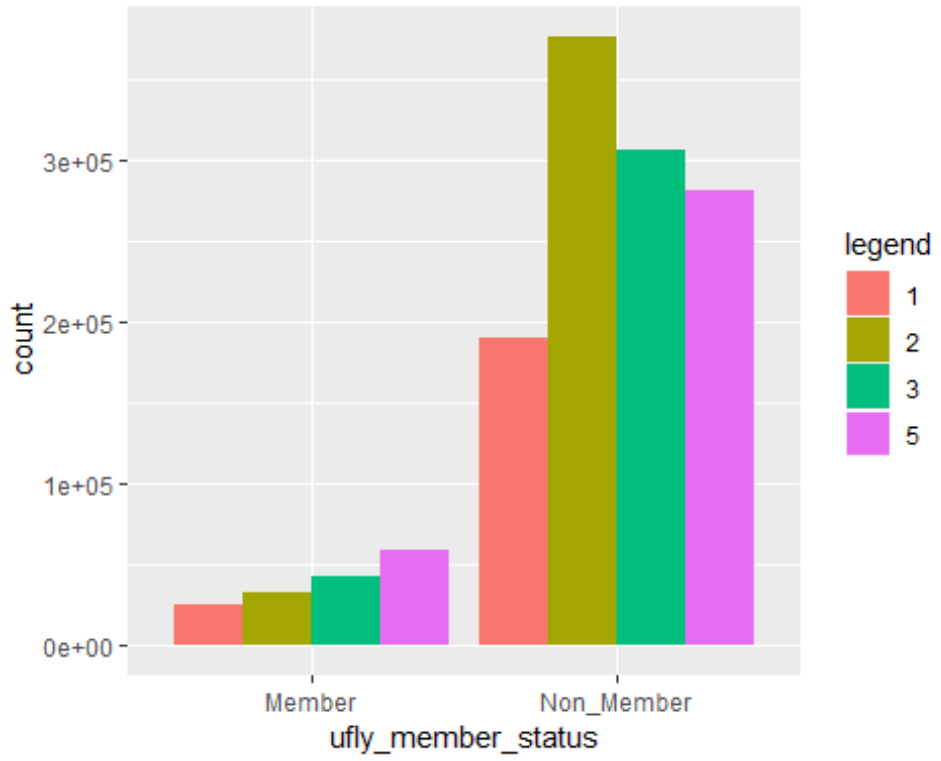


The above plot shows the number of women and men in each cluster. Cluster 1 is evenly distributed. Clusters 2 and 5 are female dominated, while cluster 3 is male dominated.

Which Clusters Contain Ufly Members?

Cluster 4 will be ignored since it does not make sense to track ufly member status with group travel.

```
sun_cust %>%
  filter(cluster != "4") %>%
  mutate(ufly_member_status = if_else(ufly_member_status == "Non_Member", "No
n_Member", "Member")) %>%
  ggplot(aes(x=ufly_member_status)) +
  geom_bar(aes(fill=cluster), position = "dodge") +
  scale_fill_manual("legend", values = c("1"= "#F8766D", "2" = "#A3A500", "3"
= "#00BF7D","5" = "#E76BF3"))
```




```
sun_cust %>%
  filter(cluster != "4") %>%
  mutate(ufly_member_status = if_else(ufly_member_status == "Non_Member", "No
n_Member", "Member")) %>%
  group_by(cluster) %>%
  summarise(member_count = sum(ufly_member_status == "Member"),
            group_count = n()) %>%
  mutate(member_percent = round((member_count/group_count)*100,1)) %>%
  select(cluster, member_percent)
```

```
## # A tibble: 4 x 2
##   cluster member_percent
##   <fct>            <dbl>
## 1 1                 11.8
## 2 2                  8
## 3 3                 12.3
## 4 5                 17.2
```

The bar graph above shows the total count of members and non-members in each cluster. Cluster 5 has the most Ufly members, while cluster 1 has the least. These numbers are absolute numbers, so the table following the graph represents what percent of each cluster are members. Cluster 5 is still the largest by percentage, but Cluster 2 is the lowest. Cluster 2 is made up of young adults who are also more price sensitive than the other groups. This represents an opportunity for adding these customers to the rewards program.

Conclusions

Cluster 1

Cluster 1 represents about 16 percent of the data and is primarily composed of children, with the parents distributed throughout clusters 2, 3 and 5. While targeting children themselves through marketing and advertising does not make sense, this cluster can be used to target families in general. The median time that cluster 1 books in advance is 51 days. In addition, this cluster travels the most in quarter 1 and quarter 3 which line up with spring break and summer vacation.

Sun Country should increase their marketing towards families starting in January and ending in June.

Cluster 2

Cluster 2 is the largest cluster and represents nearly 31 percent of the data. It is made up of mostly single travelers with many couples as well and encompasses young adults from ages 21 to 30. This group generally books more last minute; however, they are also more price sensitive compared to the other clusters. The group has consistent travel year-round, but also has the lowest percentage of Ufly Members.

Sun Country should aim to increase the Ufly membership of young adults. Since the group appears to be more price sensitive compared to the other groups, Sun Country's message should explain how a Ufly membership can save money. Sun Country should also implement "member only" sales to further entice people to join the program.

Cluster 3

Cluster 3 makes up about 26 percent of the data and is composed equally of singles and couples. In addition, they are primarily aged from 40 to 50 and are male. Like cluster 2, cluster 3 does not book far in advance and travels year-round. Unlike cluster 2, however, cluster 3 appeara to be less price sensitive.

Cluster 4

Cluster 4 makes up less than one percent of the data, however it is the most unique cluster with a median group size of 28. Unsurprisingly, this group books well in advance with a median of 155 days. This group primarily travels in Q1 with almost no travel in Q3 or Q4.

Sun Country should advertise to groups looking to travel in Q1. They will have to begin these campaigns 150 to 200 days before the quarter.

Cluster 5

Cluster 5 makes up about one quarter of the data. It is composed of primarily couples with many single travelers too. This group is primarily over the age of 60 and travels year-round, but with increased travel in Q1 and Q4. This group also has the highest percentage of Ufly members at 17.2 percent.

Sun Country should continue to convert older members to the Ufly rewards program, as they are likely to join. In addition, the marketing towards this group should focus on destinations that older couples and friends would likely travel to, especially during Q1 and Q4.

Approach 2: Identifying loyalty characteristics for UFly program

Explanation:

We expect that there will be a significant difference in different types of UFly members, with respect to their travel patterns. By identifying the features of UFly members, we can create offers that best suit the demand for varied groups. All non-members who follow similar patterns can also be sent similar offers

Key Goal:

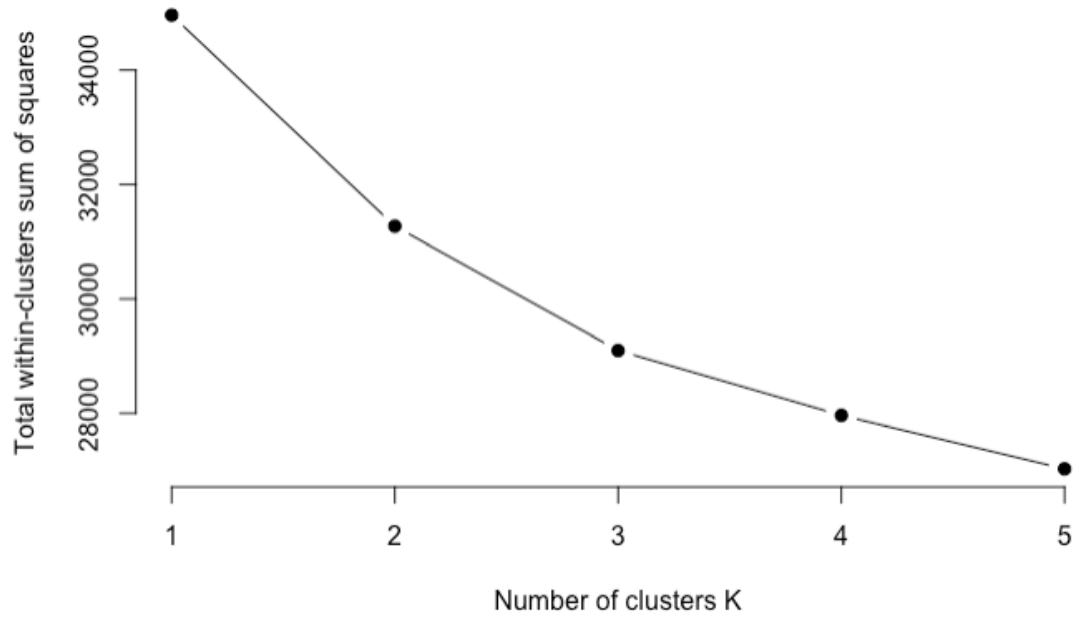
Identify the needs and demands of UFly Members to keep them satisfied.

```
## Finding no of tickets booked under each PNR and filtering for members only
sun2 <- sun %>%
  group_by(PNRLocatorID) %>%
  summarise(n_pnr = n()) %>%
  right_join(sun, by="PNRLocatorID") %>%
  filter(UflyMemberStatus != "Non_Member") %>%
  mutate(n_trips = n_pnr/group_size)

## Warning: package 'bindrcpp' was built under R version 3.4.4

## Arranging the data by PNR and CouponSeqNumber
## Finding overall no of trips in one pnr (say MSP -> JFK, JFK -> MSP are two
trips)

# If grouped by ticket number -> find no of round trips & Single trips
sun31 <- sun2 %>%
        arrange(PNRLocatorID,CouponSeqNbr,TicketNum) %>%
        group_by(TicketNum) %>%
        summarise(round_trip = n())  %>% # 2 = round trip, 1 = one way
        right_join(sun2, by="TicketNum")

# Adding zipcode
#install.packages("zipcode")
library(zipcode)
data(zipcode)
sun32 <- sun31 %>%
         left_join(zipcode,by = c("PostalCode" = "zip")) %>%
         select(-latitude,-longitude)
```

```r
## numerical grouping
sun33 <- sun32 %>%
         group_by(UFlyRewardsNumber) %>%
         arrange(UFlyRewardsNumber,quarter) %>%
         mutate(membership_days = Sys.Date() - as.Date(EnrollDate)  ) %>%
         select(UFlyRewardsNumber,UflyMemberStatus,Age,membership_days,quarte
r,GenderCode,CardHolder,state,PNRLocatorID,CouponSeqNbr,TicketNum,BookingChan
nel,BookedProduct,BaseFareAmt,TotalDocAmt,n_trips,group_size,round_trip,days_
booked_advance,ServiceStartCity,ServiceEndCity,n_trips) %>% summarise(UflyMem
berStatus = max(UflyMemberStatus), GenderCode = max(GenderCode),Age = max(Age
),CardHolder = max(CardHolder), state = max(state), membership_days= mean(mem
bership_days),BaseFareAmt = mean(BaseFareAmt),TotalDocAmt = mean(TotalDocAmt)
,days_booked_advance = mean(days_booked_advance), group_size =mean(group_size
),n_trips = mean(n_trips))

#categorical grouping
#1 group on Qtrs
#install.packages("tidyverse")
library(tidyverse)

sun3_qtr <- sun32 %>%
         group_by(UFlyRewardsNumber,quarter) %>%
         arrange(UFlyRewardsNumber,quarter) %>%
         mutate(n_qtr = n()  ) %>%
         select(UFlyRewardsNumber, quarter,n_qtr)%>%
         summarize(n_qtr = max(n_qtr))%>%
         spread(quarter,n_qtr) %>%
         replace_na(list(Q1 = 0,Q2 = 0,Q3 = 0,Q4 = 0))
#is.na(sun3_qtr)

#2 Group on BkdClass ofService And Travelled Class of Service
sun3_bkd <- sun32 %>%
         group_by(UFlyRewardsNumber,BkdClassOfService) %>%
         mutate(n_bkd = n()  ) %>%
         select(UFlyRewardsNumber, BkdClassOfService,n_bkd)%>%
         summarize(n_bkd = max(n_bkd))%>%
         spread(BkdClassOfService,n_bkd)
sun3_bkd <- sun3_bkd %>%
  rename(
    Coach_bkd = Coach,
    `Discount First Class Bkd ` = `Discount First Class`,
    `First Class Bkd` = `First Class`
    )

sun3_trl <- sun32 %>%
         group_by(UFlyRewardsNumber,TrvldClassOfService) %>%
         mutate(n_bkd = n()  ) %>%
         select(UFlyRewardsNumber, TrvldClassOfService,n_bkd)%>%
         summarize(n_bkd = max(n_bkd))%>%
         spread(TrvldClassOfService,n_bkd)
```

```r
sun3_bkd[is.na(sun3_bkd)] <- 0
sun3_trl[is.na(sun3_trl)] <- 0

# Booking Channel
sun3_channel <- sun32 %>%
         group_by(UFlyRewardsNumber,BookingChannel) %>%
         mutate(n_bkd = n()  ) %>%
         select(UFlyRewardsNumber, BookingChannel,n_bkd)%>%
         summarize(n_bkd = max(n_bkd))%>%
         spread(BookingChannel,n_bkd)

sun3_channel[is.na(sun3_channel)] <- 0

##Booked Product
# sun3_bp <- sun32 %>%
#          group_by(UFlyRewardsNumber,BookedProduct) %>%
#          mutate(n_bkd = n()  ) %>%
#          select(UFlyRewardsNumber, BookedProduct,n_bkd)%>%
#          summarize(n_bkd = max(n_bkd))%>%
#          spread(BookedProduct,n_bkd)
# sun3_bp[is.na(sun3_bp)] <- 0
#table(sun32$BookedProduct)


sun32 <- sun32 %>%
         mutate(ServiceStartCity = replace(ServiceStartCity,
                                    !(ServiceStartCity %in% c("BOS","MSP","JFK"
,"SFO","RSW","LAS","LAX","MCO")),"Other"),
                  ServiceEndCity = replace(ServiceEndCity,!(ServiceEndCity %in%
c("BOS","MSP","JFK","SFO","RSW","LAS","LAX","MCO")),
                                    "Other"))
##Service StartCity
sun3_start <- sun32 %>%
         group_by(UFlyRewardsNumber,ServiceStartCity) %>%
         mutate(n_bkd = n()  ) %>%
         select(UFlyRewardsNumber, ServiceStartCity,n_bkd)%>%
         summarize(n_bkd = max(n_bkd))%>%
         spread(ServiceStartCity,n_bkd)
sun3_start[is.na(sun3_start)] <- 0

# ##Service EndCity
# sun3_end <- sun32 %>%
#          group_by(UFlyRewardsNumber,ServiceEndCity) %>%
#          mutate(n_bkd = n()  ) %>%
#          select(UFlyRewardsNumber, ServiceEndCity,n_bkd)%>%
#          summarize(n_bkd = max(n_bkd))%>%
#          spread(ServiceEndCity,n_bkd)
# sun3_end[is.na(sun3_start)] <- 0
```

```r
sun_members <- sun33 %>%
         right_join(sun3_qtr, by="UFlyRewardsNumber") %>%
         right_join(sun3_bkd, by="UFlyRewardsNumber") %>%
         right_join(sun3_trl, by="UFlyRewardsNumber") %>%
         right_join(sun3_start, by="UFlyRewardsNumber") %>%
         right_join(sun3_channel, by="UFlyRewardsNumber")


 sun_members <- sun_members %>%
                mutate(UflyMemberStatus = as.factor(UflyMemberStatus), Gender
Code= as.factor(GenderCode), CardHolder = as.factor(CardHolder), state = as.f
actor(state))

#library(clustMixType)

#Elbow Method for finding the optimal number of clusters

#set.seed(123)

# Compute and plot wss for k = 2 to k = 5.

# k.max <- 5

# wss <- sapply(1:k.max, function(k){kproto(sun_final[,2:20], k)$tot.withinss
})

# wss

# plot(1:k.max, wss,

#      type="b", pch = 19, frame = FALSE,

#      xlab="Number of clusters K",

#      ylab="Total within-clusters sum of squares")

## no of optimal clusters -> 4
```




```
##chosing 4 clusters
kpres <- readRDS("kpres")
#kpres <- kproto(sun_final[,2:20], 4)
#saveRDS(kpres)
#kpres$cluster
#predicted.clusters <- predict(kpres, x)
sun_f <- na.omit(sun_members)
sun_clusters <- cbind(sun_f[,1], kpres$cluster)

sun_clusters <- sun_members %>% right_join(sun_clusters, by = "UFlyRewardsNum
ber")
fwrite(sun_clusters, file = "SunCountry_clusters.csv")
# Gender_clus <- sun_clusters %>%
#                 group_by(`kpres$cluster`,GenderCode) %>%
#                 mutate(n_gender = n()  ) %>%
#                 select(`kpres$cluster`,GenderCode,n_gender) %>%
#                 distinct(`kpres$cluster`,GenderCode,n_gender)%>%
#                 filter(GenderCode !='U')
Age_clus <- sun_clusters %>%
                select(`kpres$cluster`,Age) %>%
                group_by(`kpres$cluster`) %>%
                mutate(Age = mean(Age)  ) %>%
                select(`kpres$cluster`,Age) %>%
                distinct(`kpres$cluster`,Age)
Status_clus <- sun_clusters %>%
                select(`kpres$cluster`,UflyMemberStatus) %>%
                group_by(`kpres$cluster`,UflyMemberStatus) %>%
                mutate(n_a = n()  ) %>%
                select(`kpres$cluster`,UflyMemberStatus,n_a) %>%
```

```
                distinct(`kpres$cluster`,UflyMemberStatus,n_a) %>%
                spread(UflyMemberStatus,n_a)
Rev_clus <- sun_clusters %>%
                select(`kpres$cluster`,TotalDocAmt) %>%
                group_by(`kpres$cluster`) %>%
                mutate(TotalDocAmt = mean(TotalDocAmt)  ) %>%
                distinct(`kpres$cluster`,TotalDocAmt)
days_Booked_clus <- sun_clusters %>%
                select(`kpres$cluster`,days_booked_advance) %>%
                group_by(`kpres$cluster`) %>%
                mutate(days_booked_advance = mean(days_booked_advance)  ) %>%
                distinct(`kpres$cluster`,days_booked_advance)
Website_Booked_clus <- sun_clusters %>%
                select(`kpres$cluster`,`SCA Website Booking`) %>%
                group_by(`kpres$cluster`) %>%
                mutate(`SCA Website Booking` = mean(`SCA Website Booking`)  )
%>%
                distinct(`kpres$cluster`,`SCA Website Booking`)
membership_clus <- sun_clusters %>%
                select(`kpres$cluster`,membership_days) %>%
                group_by(`kpres$cluster`) %>%
                summarise(membership_days = mean(membership_days)  ) %>%
                distinct(`kpres$cluster`,membership_days)
n_trips_clus <- sun_clusters %>%
                select(`kpres$cluster`,n_trips) %>%
                group_by(`kpres$cluster`) %>%
                summarise(n_trips = mean(n_trips)  ) %>%
                distinct(`kpres$cluster`,n_trips)
class_clus <-  sun_clusters %>%
                select(`kpres$cluster`,`First Class`,`Discount First Class`,C
oach) %>%
                group_by(`kpres$cluster`) %>%
                summarise(`First Class` = mean(`First Class`), `Discount Firs
t Class` = mean(`Discount First Class`), Coach =mean(Coach) ) %>%
                distinct(`kpres$cluster`,`First Class`,`Discount First Class`
,Coach)

city_clus <- sun_clusters %>%
                select(`kpres$cluster`,JFK, MSP, LAS, LAX, BOS) %>%
                group_by(`kpres$cluster`) %>%
                summarise(JFK = mean(JFK),MSP = mean(MSP), LAS = mean(LAS), L
AS = mean(LAX), BOS = mean(BOS)  ) %>%
                distinct(`kpres$cluster`,JFK, MSP, LAS, LAX, BOS)

## Warning: Trying to compute distinct() for variables not found in the data:
## - `LAX`
## This is an error, but only a warning is raised for compatibility reasons.
## The following variables will be used:
## - kpres$cluster
## - JFK
```

```
## - MSP
## - LAS
## - BOS

clus_table  <- Status_clus%>% right_join(Age_clus , by = "kpres$cluster") %>%
                              right_join(Rev_clus , by = "kpres$cluster") %>%
                              right_join(membership_clus , by = "kpres$cluster"
)%>%
                              right_join(days_Booked_clus , by = "kpres$cluster
") %>%
                              right_join(n_trips_clus , by = "kpres$cluster") %
>%
                              right_join(class_clus , by = "kpres$cluster") %>%
                              right_join(city_clus , by = "kpres$cluster") %>%
                              right_join(Website_Booked_clus , by = "kpres$clus
ter") %>%
                              arrange(`kpres$cluster`)

clus_table[,c(1:5)]

## # A tibble: 4 x 5
## # Groups:   kpres$cluster [4]
##   `kpres$cluster` Elite Standard   Age TotalDocAmt
##             <int> <int>    <int> <dbl>       <dbl>
## 1               1    20    84368  45.1         298
## 2               2   420    11458  53.2         345
## 3               3    33    70165  45.4         299
## 4               4    16    26148  41.9         493

clus_table[,c(1,6:8,16)]

## # A tibble: 4 x 5
## # Groups:   kpres$cluster [4]
##   `kpres$cluster` membership_days  days_booked_advance n_trips
##             <int> <time>                         <dbl>   <dbl>
## 1               1 2547.30230601507                59.6    1.80
## 2               2 3058.47381714093                59.9    1.86
## 3               3 2474.83986723269                47.8    1.75
## 4               4 2711.50149059777               141      1.97
## # ... with 1 more variable: `SCA Website Booking` <dbl>

clus_table[,c(1,9:11)]

## # A tibble: 4 x 4
## # Groups:   kpres$cluster [4]
##   `kpres$cluster` `First Class` `Discount First Class` Coach
##             <int>         <dbl>                  <dbl> <dbl>
## 1               1         0.175                  0.133  2.43
## 2               2         1.99                   1.45   9.39
## 3               3         0.202                  0.147  2.45
## 4               4         0.345                  0.105  2.62
```

```
clus_table[,c(1,12:15)]
```

```
## # A tibble: 4 x 5
## # Groups:   kpres$cluster [4]
##   `kpres$cluster`    JFK   MSP    LAS    BOS
##             <int>  <dbl> <dbl>  <dbl>  <dbl>
## 1               1 0.0964  1.33 0.108  0.0932
## 2               2 0.551   6.19 0.490  0.467
## 3               3 0.113   1.36 0.122  0.106
## 4               4 0.0253  1.53 0.0376 0.0315
```

Interpretation of this approach:

We found 4 different clusters of UFlyMembers on the basis of their Age, Membership status, the set of cities they book most flights from, the travelled class, how often they book from the website, their membership days and the number of days they book in advance.

Conclusions from this approach:

We found that our oldest UFly members preferred online SCA booking and also traveled First Class more than other clusters. Since they are also churning, we should target them with better offers online. Our highest spenders did not use the SCA Website booking often. We should target these group of people by incentivizing booking through our website so that we can track their online behavior.

Potential drawbacks:

1. We haven't considered the impact of other online factors such as third-party offers on bookings.
2. The difference between Elite and Standard travelers are not highlighted in these groups. So, we have assumed that they are behaving similarly.

Based on these conclusions, explanation, and goals for the next approach:

Now we have identified an opportunity to keep loyal customers engaged with us through UFly program. However, it is important to reduce potential customer churn. Identifying customer characteristics and/or behavior will help us in targeting customers who we are likely to lose and improve their experience. Hence as a next step, we will try to uncover patterns (if they exist) occurring with higher churn rate.

Approach 3: Reduction in customer churn

Explanation:

We will start with trying to identify the reasons behind customer churn. The key assumption is that there are segments of customers exhibiting similar behavior than other customers which tend to show a high churn rate. Identifying such latent customer clusters will help us in understanding characteristics that lead to churn. The churned customers have been identified based on a threshold for days for which they have not traveled with Sun Country. It has been fixed based on comparing an average number of days for customers between two journeys. Various customer characteristics such as age, gender etc. have been used. Moreover, our assumption is that due to the extremely competitive market and low switching costs, a single bad experience could lead to customer churn. Hence we are looking at variables such as source and destination for the last journey. The only numerical variables considered are age and amount for the last journey. However their distributions do not have extreme scale difference, for ease of interpretability, they haven't been normalized.

Key Goal:

Identify consumer `characteristics and/or behavioral patterns` that might be the cause of `customer churn`.

```
sun$PNRCreateDate <- as.Date(sun$PNRCreateDate, '%Y-%m-%d')
sun$ServiceStartDate <- as.Date(sun$ServiceStartDate, '%Y-%m-%d')
last_travel <- sun %>% filter(CouponSeqNbr == 1) %>% group_by(EncryptedName,
birthdateid, GenderCode) %>% summarise(first = min(PNRCreateDate), last = max
(PNRCreateDate), travels = n())
last_travel$dwsc <- last_travel$last - last_travel$first
last_travel$avg_days <- as.numeric(last_travel$dwsc / last_travel$travels)
```

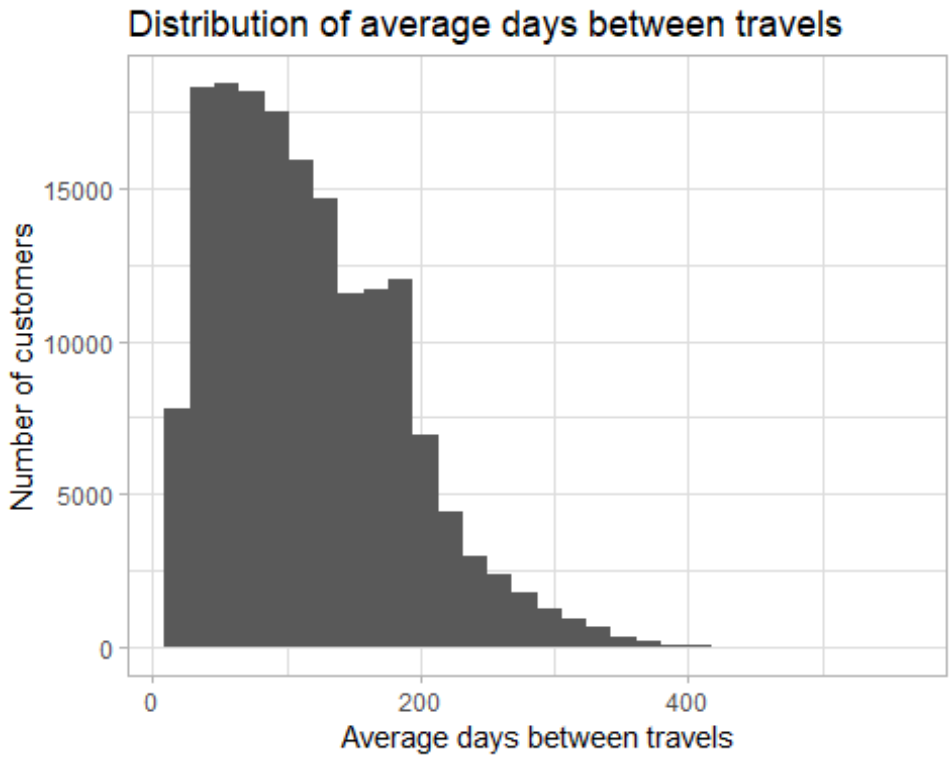
To determine threshold for number of days travelled, we find out average number of days between to consicutive flights by a customer based on number of days between first and last flight divided by total number of flights taken. The connecting flights are excluded to determine a particular `journey`.

```
summary(last_travel$avg_days)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    0.00    0.00   13.01    0.00  557.50
```

```
ggplot(last_travel[last_travel$avg_days >= 20,], aes(x = avg_days)) +
  geom_histogram() +
  theme_light() +
  ggtitle('Distribution of average days between travels') +
  xlab('Average days between travels') +
  ylab('Number of customers')
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
last_travel$churned <- ifelse(last_travel$avg_days > 200, 1, 0)
```

Hence based on the above distribution, the people who have not traveled with sun country for more than 200 days have been identified as the customers who have churned.

Hence preparing relevant data for clustering:

```
sun_req <- sun %>% filter(CouponSeqNbr == 1) %>% select(EncryptedName, Servic
eStartCity, birthdateid
                          ,ServiceEndCity
                          ,PNRCreateDate
                          ,ServiceStartDate
                          ,GenderCode
                          ,Age
                          ,BkdClassOfService
                          ,TrvldClassOfService
                          ,BookingChannel
                          ,BaseFareAmt
                          ,TotalDocAmt
                          ,UflyMemberStatus
                          ,CardHolder
                          ,BookedProduct
                          ,EnrollDate)

clust_data <- merge(last_travel, sun_req,
                    by.x = c('EncryptedName', 'GenderCode', 'birthdateid', 'l
ast'),
```

```
                    by.y = c('EncryptedName', 'GenderCode', 'birthdateid', 'P
NRCreateDate'))
clust_data$id <- seq(1, nrow(clust_data))
eval_flag <- clust_data %>% select(id, churned)
clust_data <- clust_data %>% select(id, ServiceStartCity, ServiceEndCity, Ser
viceStartDate, Age, GenderCode, UflyMemberStatus,
                                    CardHolder, TotalDocAmt, churned)

clust_data[which(is.na(clust_data$UflyMemberStatus) | clust_data$UflyMemberSt
atus == ''),'UflyMemberStatus'] <- 'Not Member'
clust_data[which(is.na(clust_data$CardHolder)),'CardHolder'] <- 'Not Member'


clust_data <- clust_data %>% mutate(ServiceStartCity = as.factor(ServiceStart
City),
                                    ServiceEndCity = as.factor(ServiceEndCity
),
                                    GenderCode = as.factor(GenderCode),
                                    UflyMemberStatus = as.factor(UflyMemberSt
atus),
                                    CardHolder = as.factor(CardHolder),
                                    churned = as.factor(churned))
```

Since the original data size at customer level is extremely large, to identify clusters effectively, we have used stratified sampling. Based on our goal to identify churn, the sampling is done based on churn flag.

```
# Stratified sampling based on churn variable
set.seed(1994)
clust_data_fin <- stratified(clust_data, 'churned', 0.05)
clust_data_fin <- na.omit(clust_data_fin)
```

Due to mixed nature of data, we have selected `k-prototypes` as method for clustering.

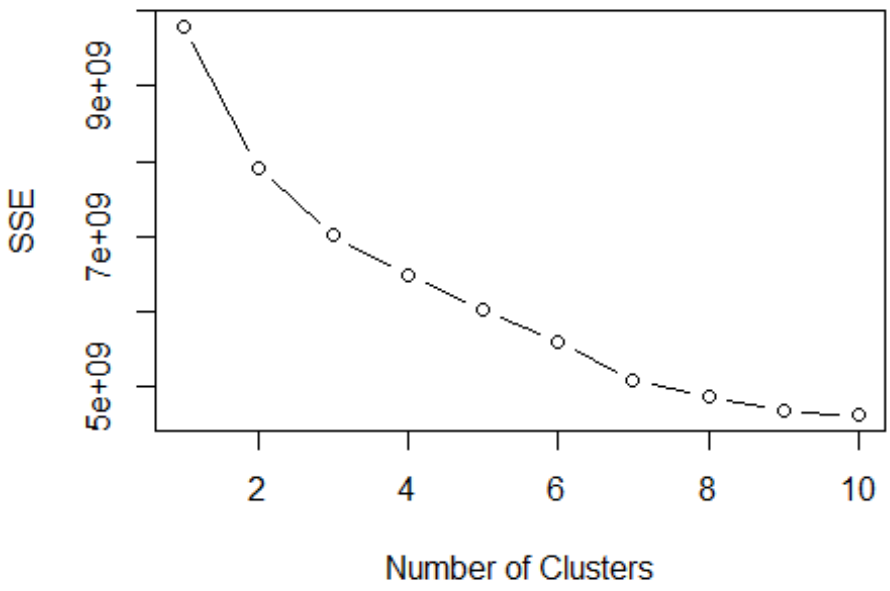
```
SSE_curve <- c()
for (k in 1:10){
  kpro <- kproto(clust_data_fin %>% select(-id, -churned), k)
  sse <- sum(kpro$withinss)
  SSE_curve[k] <- sse
}
```

```
## # NAs in variables:
## ServiceStartCity   ServiceEndCity ServiceStartDate              Age
##                0                0                0                0
##       GenderCode UflyMemberStatus       CardHolder      TotalDocAmt
##                0                0                0                0
## 0 observation(s) with NAs.
##
## Estimated lambda: 42079.07
##
```

```
## # NAs in variables:
## ServiceStartCity   ServiceEndCity ServiceStartDate              Age
##                0                0                0                0
##       GenderCode UflyMemberStatus       CardHolder      TotalDocAmt
##                0                0                0                0
## 0 observation(s) with NAs.
##
## Estimated lambda: 42079.07
##
## # NAs in variables:
## ServiceStartCity   ServiceEndCity ServiceStartDate              Age
##                0                0                0                0
##       GenderCode UflyMemberStatus       CardHolder      TotalDocAmt
##                0                0                0                0
## 0 observation(s) with NAs.
##
## Estimated lambda: 42079.07
##
## # NAs in variables:
## ServiceStartCity   ServiceEndCity ServiceStartDate              Age
##                0                0                0                0
##       GenderCode UflyMemberStatus       CardHolder      TotalDocAmt
##                0                0                0                0
## 0 observation(s) with NAs.
##
## Estimated lambda: 42079.07
##
## # NAs in variables:
## ServiceStartCity   ServiceEndCity ServiceStartDate              Age
##                0                0                0                0
##       GenderCode UflyMemberStatus       CardHolder      TotalDocAmt
##                0                0                0                0
## 0 observation(s) with NAs.
##
## Estimated lambda: 42079.07
##
## # NAs in variables:
## ServiceStartCity   ServiceEndCity ServiceStartDate              Age
##                0                0                0                0
##       GenderCode UflyMemberStatus       CardHolder      TotalDocAmt
##                0                0                0                0
## 0 observation(s) with NAs.
##
## Estimated lambda: 42079.07
##
## # NAs in variables:
## ServiceStartCity   ServiceEndCity ServiceStartDate              Age
##                0                0                0                0
##       GenderCode UflyMemberStatus       CardHolder      TotalDocAmt
##                0                0                0                0
```

```
## 0 observation(s) with NAs.
##
## Estimated lambda: 42079.07
##
## # NAs in variables:
## ServiceStartCity   ServiceEndCity ServiceStartDate              Age
##                0                0                0                0
##       GenderCode UflyMemberStatus       CardHolder      TotalDocAmt
##                0                0                0                0
## 0 observation(s) with NAs.
##
## Estimated lambda: 42079.07
##
## # NAs in variables:
## ServiceStartCity   ServiceEndCity ServiceStartDate              Age
##                0                0                0                0
##       GenderCode UflyMemberStatus       CardHolder      TotalDocAmt
##                0                0                0                0
## 0 observation(s) with NAs.
##
## Estimated lambda: 42079.07
##
## # NAs in variables:
## ServiceStartCity   ServiceEndCity ServiceStartDate              Age
##                0                0                0                0
##       GenderCode UflyMemberStatus       CardHolder      TotalDocAmt
##                0                0                0                0
## 0 observation(s) with NAs.
##
## Estimated lambda: 42079.07
```

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



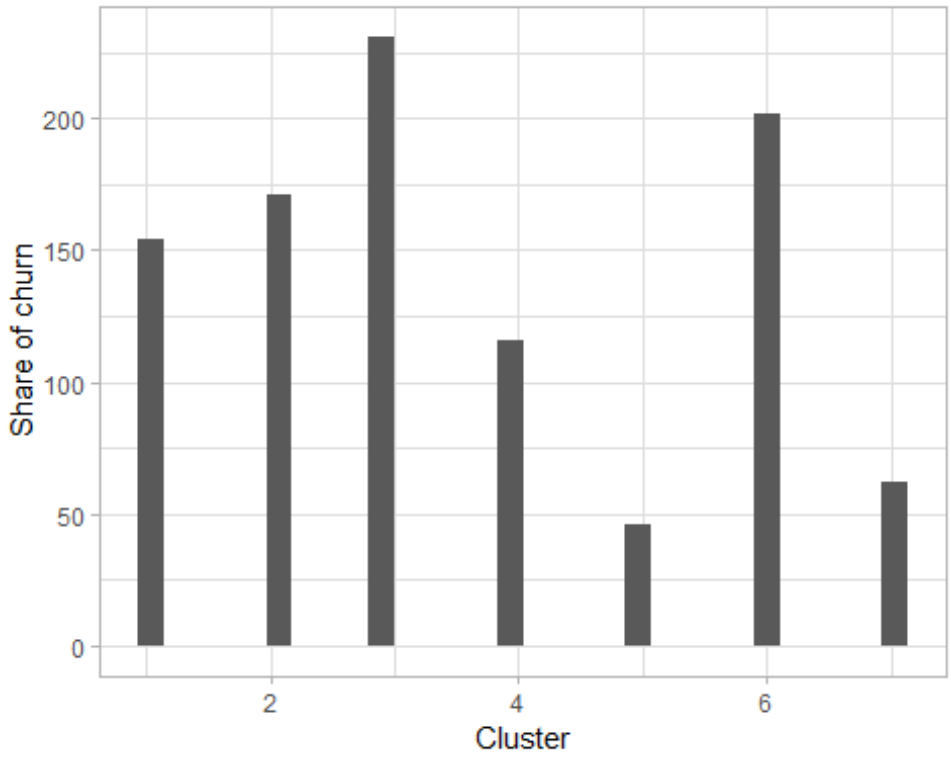

```
save.image('EDA.RData')

# Final
churnclust <- kproto(clust_data_fin, 7)

## # NAs in variables:
##               id ServiceStartCity   ServiceEndCity ServiceStartDate
##                0                0                0                0
##              Age       GenderCode UflyMemberStatus       CardHolder
##                0                0                0                0
##      TotalDocAmt          churned
##                0                0
## 0 observation(s) with NAs.
##
## Estimated lambda: 1.73074e+11

clust_data_fin$clusts <- churnclust$cluster
clust_data_fin$churned <- as.numeric(clust_data_fin$churned)
```

Based on the graph below, we observe that the cluster 3 ha highest share of churn while cluster 5 has lowest share

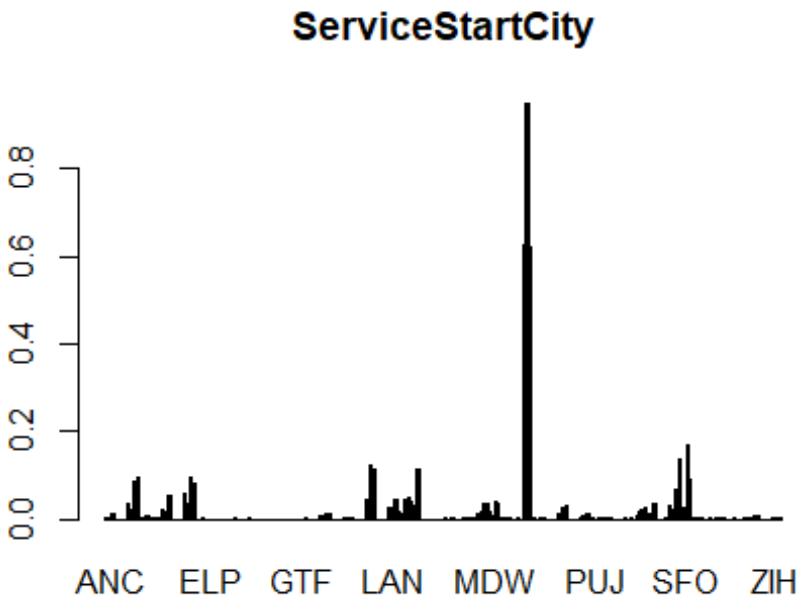
```
ggplot(clust_data_fin[which(clust_data_fin$churned == 2),], aes(x = clusts))
+
  geom_histogram() +
  theme_light() +
```

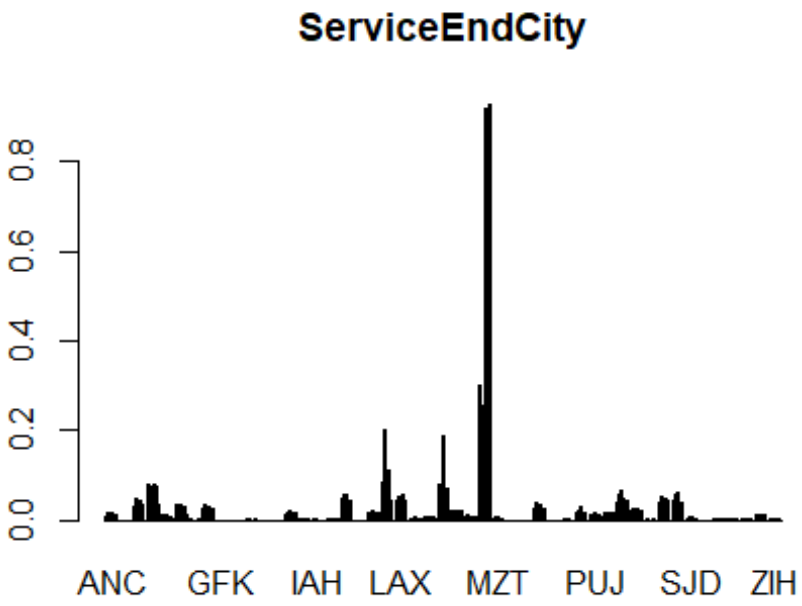
```
  xlab('Cluster') +
  ylab('Share of churn')
```

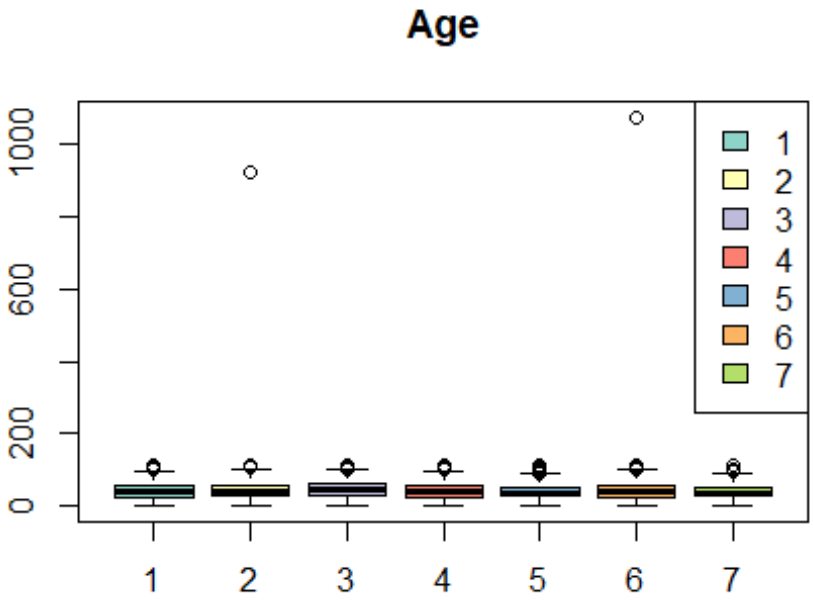
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

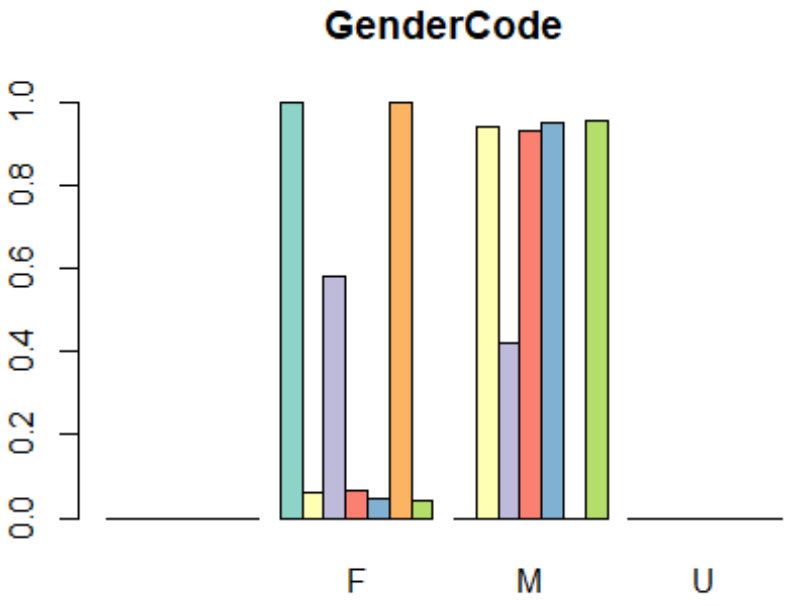


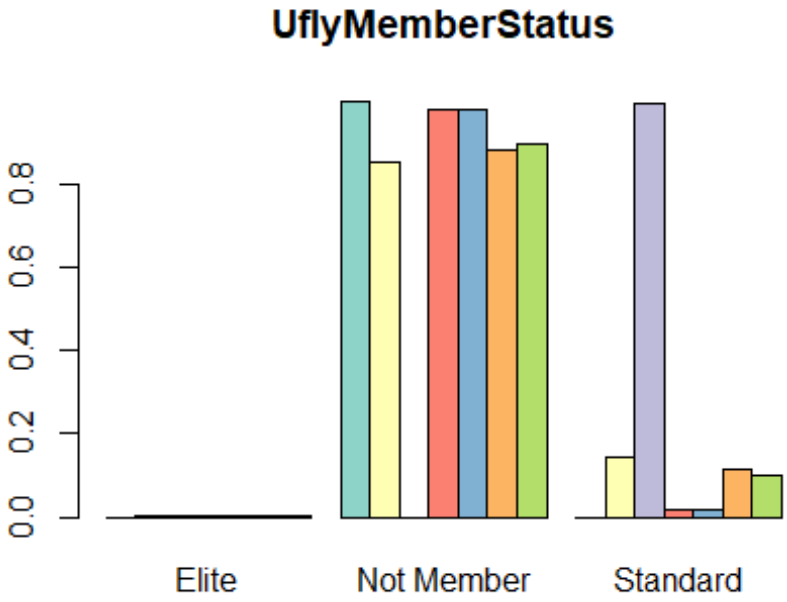

```
clprofiles(churnclust, clust_data_fin)
```

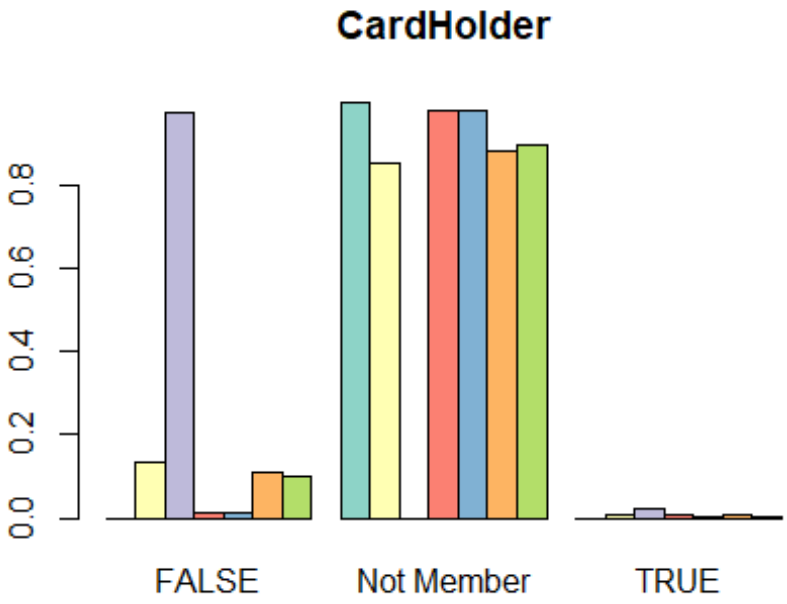

ServiceStartCity
0.0
0.2
0.4
0.6
0.8
ANC
ELP
GTF
LAN
MDW
PUJ
SFO
ZIH

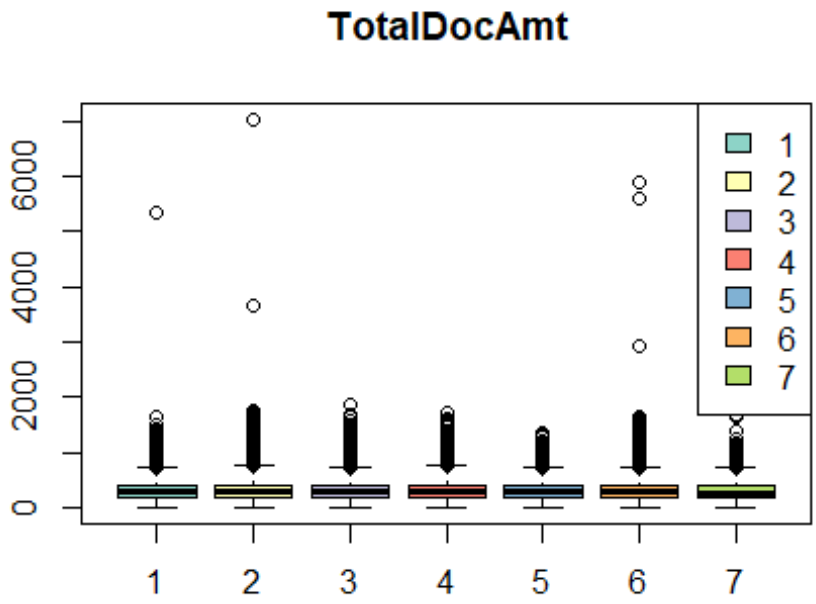

ServiceEndCity
0.0
0.2
0.4
0.6
0.8
ANC
GFK
IAH
LAX
MZT
PUJ
SJD
ZIH


Age
0
200
600
1000
1
2
3
4
5
6
7


GenderCode
0.0
0.2
0.4
0.6
0.8
1.0
F
M
U


UflyMemberStatus
0.0
0.2
0.4
0.6
0.8
Elite
Not Member
Standard


CardHolder
0.0
0.2
0.4
0.6
0.8
FALSE
Not Member
TRUE




```
churnclust$centers
```

```
##           id ServiceStartCity ServiceEndCity ServiceStartDate      Age
## 1: 1192161.4              MSP            MSP       2014-03-06 38.50139
## 2:  404689.6              MSP            LAS       2013-12-09 40.31919
## 3: 1067266.7              MSP            MSP       2014-02-26 44.18173
## 4: 1232860.9              MSP            MCO       2014-03-26 38.25014
## 5: 1256657.7              SFO            MSP       2014-01-28 37.28083
## 6:  359490.1              MSP            MSP       2014-01-15 39.40488
## 7:  418151.3              SEA            MSP       2013-09-05 37.59553
##    GenderCode UflyMemberStatus CardHolder TotalDocAmt churned
## 1:          F       Not Member Not Member    288.4781       0
## 2:          M       Not Member Not Member    303.1008       0
## 3:          F         Standard      FALSE    321.7616       0
## 4:          M       Not Member Not Member    301.6022       0
## 5:          M       Not Member Not Member    288.6982       0
## 6:          F       Not Member Not Member    294.7801       0
## 7:          M       Not Member Not Member    291.9139       0
```

Interpretation of this approach:

1. The cluster having the highest churn rate have following characteristics:
2. Highest cost for flight and higher for comparable return journeys as well
3. They are the customers with highest age
4. Consists of females in majority
5. `UFly members` are predominant in these clusters

6. On the other hand, the cluster having lowest churn have lowest airfares and are the youngest customers.

Conclusions from this approach:

1. It is highly interesting to note that the highest share of churned customers is in the cluster which belongs to a UFly membership status - Hence we recommend surveying female UFly members to improve the customer experience
2. We need to promote younger customers to maintain their loyalty

Potential drawbacks:

1. Clustering is done on sampled data and even if it is done in a repetitive manner, it might not always result in a true representation of data
2. The customers who are not flying with us for more than 200 days could be seasonal annual flyers hence might not have churned in reality
3. There might be other factors resulting in churn which are not visible due to limited data

Based on these conclusions, explanation, and goals for the next approach:

As now we have identified how to keep the current revenue, we would like to now identify how to generate more revenue. Hence, next, we identify patterns among customers which can lead to newer revenue generation opportunities. One way to do that is to maximize capacity utilization in 'First Class' segment.

Approach 4: Identify customer segments most likely to buy first class tickets

Explanation:

Looking at the past 2 years of data it appears there have often been cases where customers were offered a free upgrade from 'Coach' to 'First Class'. Indicating that there is some underutilization of capacity in first class. This presents an opportunity to identify and upgrade targeted customer groups for a free upgrade to first class based on how likely they are to book a first class ticket in the future.

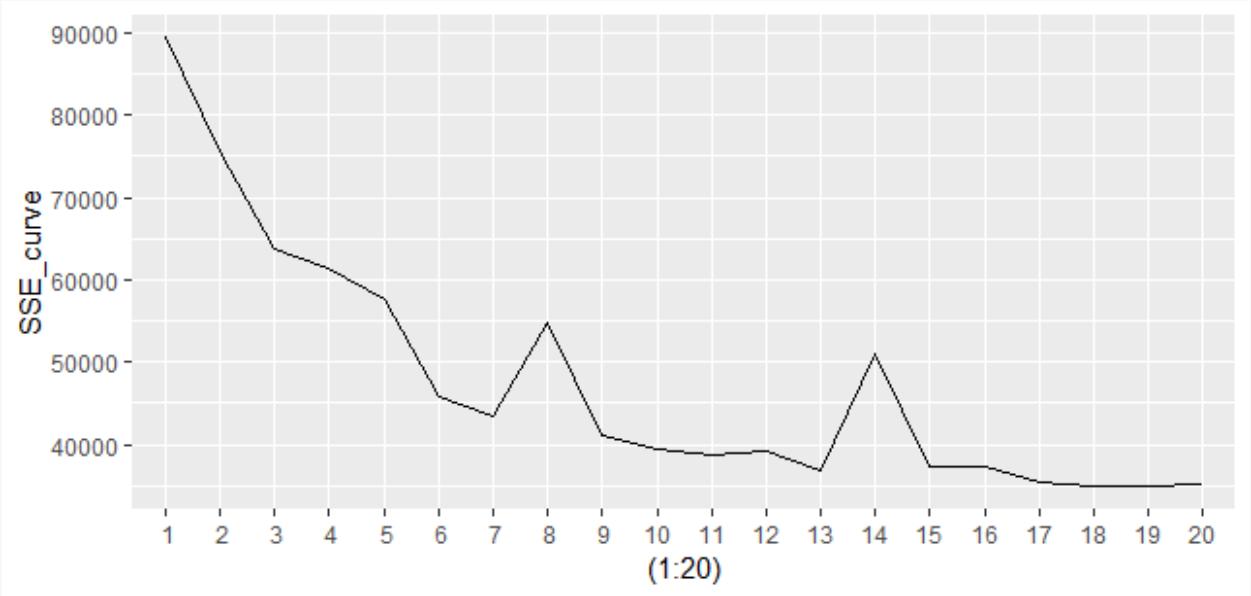
Key Goal:

Identify customer groups most likely to book 'First Class' tickets in future if given a free 'First Class' upgrade in past.

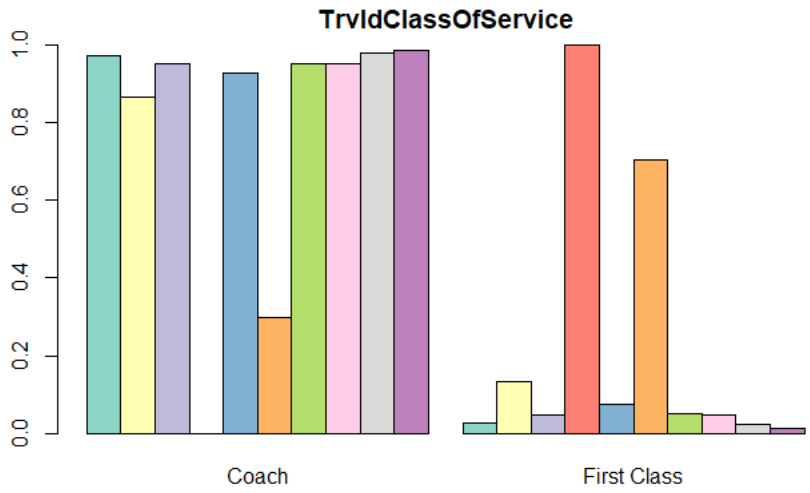
```
# Loading data and correcting column data types
sun <- fread("SunCountry_cleaned_2.csv")

normalize <- function(x){
  x <- as.numeric(x)
  return ((x - min(x))/(max(x) - min(x)))}

sun <- sun %>%
```

```
  mutate(TotalDocAmt = normalize(TotalDocAmt),
         BaseFareAmt = normalize(BaseFareAmt),
         days_booked_advance = normalize(days_booked_advance),
         Age = normalize(Age))

sun <- mutate(sun,ServiceStartDate = as.Date(ServiceStartDate))

#Names and start date of people who were upgradaded
sun_upgrade_names <-  sun %>% filter(BkdClassOfService!=TrvldClassOfService)
%>% select(EncryptedName,ServiceStartDate) %>% group_by(EncryptedName) %>% su
mmarise(FirstStartDate = max(ServiceStartDate))

#Joined Data frame of all trips in which customer where once upgraded from co
ach to first class
sun_upgrade <- sun %>% inner_join(sun_upgrade_names, by = 'EncryptedName') %>
% filter(ServiceStartDate > FirstStartDate)

#Convert 'Discount First Class' to 'First Class' as we are only looking at Fi
rst Class travellers.
sun_upgrade$BkdClassOfService[sun_upgrade$BkdClassOfService=='Discount First
Class']  <- "First Class"
sun_upgrade$TrvldClassOfService[sun_upgrade$TrvldClassOfService=='Discount Fi
rst Class']  <- "First Class"

sun_cluster <- sun_upgrade %>%
  dplyr::select(-PNRLocatorID,
                -TicketNum,
                -PNRCreateDate,
                -CouponSeqNbr,
                -ServiceStartDate,
                -PaxName,
                -EncryptedName,
                -PostalCode,
                -BookedProduct,
                -EnrollDate,
                -MarketingFlightNbr,
                -BkdClassOfService,
                -birthdateid,
                -UFlyRewardsNumber,
                -StopoverCode,
                -FirstStartDate)


SSE Curve for K Prototype
SSE_curve <- c()
for (k in 2:10){
   kpro <- kproto(sun_cluster, k)
   sse <- sum(kpro$withinss)
   SSE_curve[k] <- sse
```

```
 }
 ggplot()+ aes(x=(1:10), y=SSE_curve) + geom_line() + scale_x_discrete(limits
=(1:10))
```

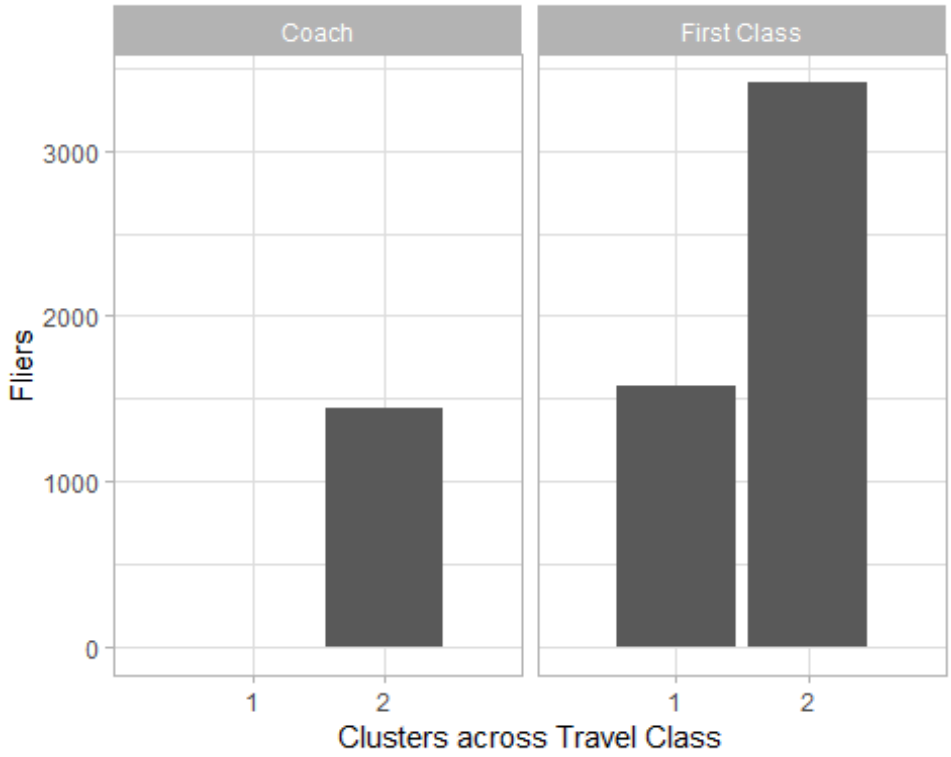



```
kpro10 <- kproto(sun_cluster, 10)

clprofiles(kpro10, sun_cluster, vars=c('TrvldClassOfService'))
```



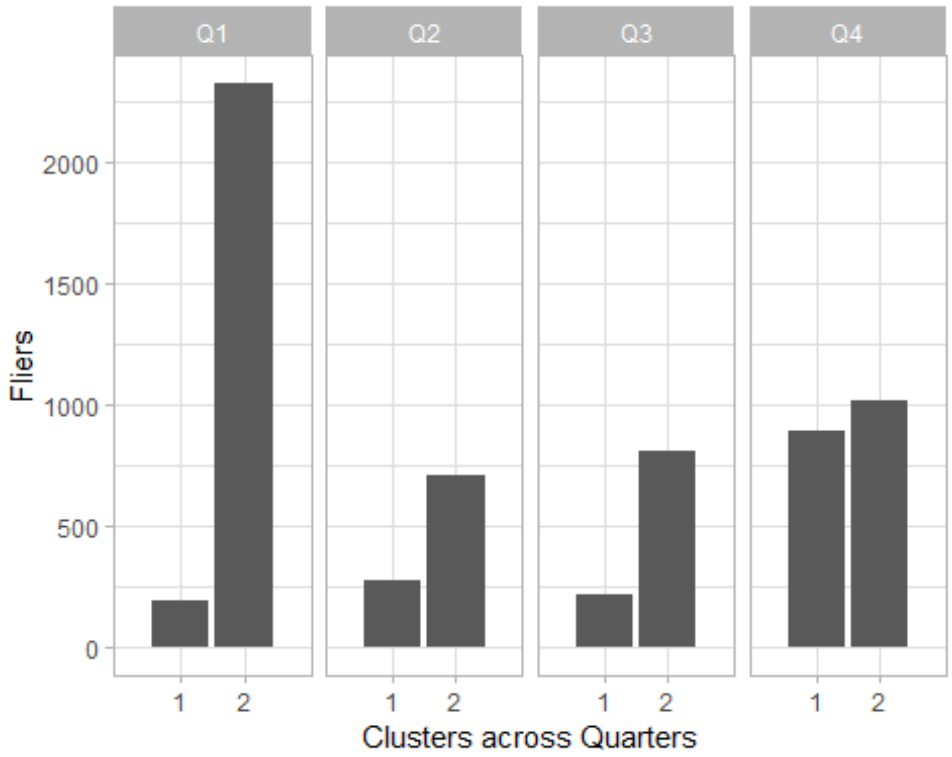
Plotting the 10 clusters across travel class, indicates that there are two clusters of interest for us, orange and red. Cluster red only consists of first class fliers and the orange cluster has higher proportion of First class fliers.

```
#Filtering the data set to only look at the clusters identified as clusters o
f interest from above
sun_cluster$cluster <-  kpro10$cluster
sun_cluster <- sun_cluster %>% mutate(cluster = case_when((cluster != 6 & clu
ster!=4)~ 3, cluster==4 ~ 1, cluster==6 ~2, TRUE ~ as.numeric(.$cluster)))

sun_cluster_2 <- sun_cluster[which(sun_cluster$cluster != 3),]
ggplot(sun_cluster_2, aes(x=cluster))+ facet_grid(. ~ TrvldClassOfService) +
geom_bar() + theme_light() +scale_x_discrete( limits=1:2) + xlab("Clusters ac
ross Travel Class") + ylab("Fliers")
```



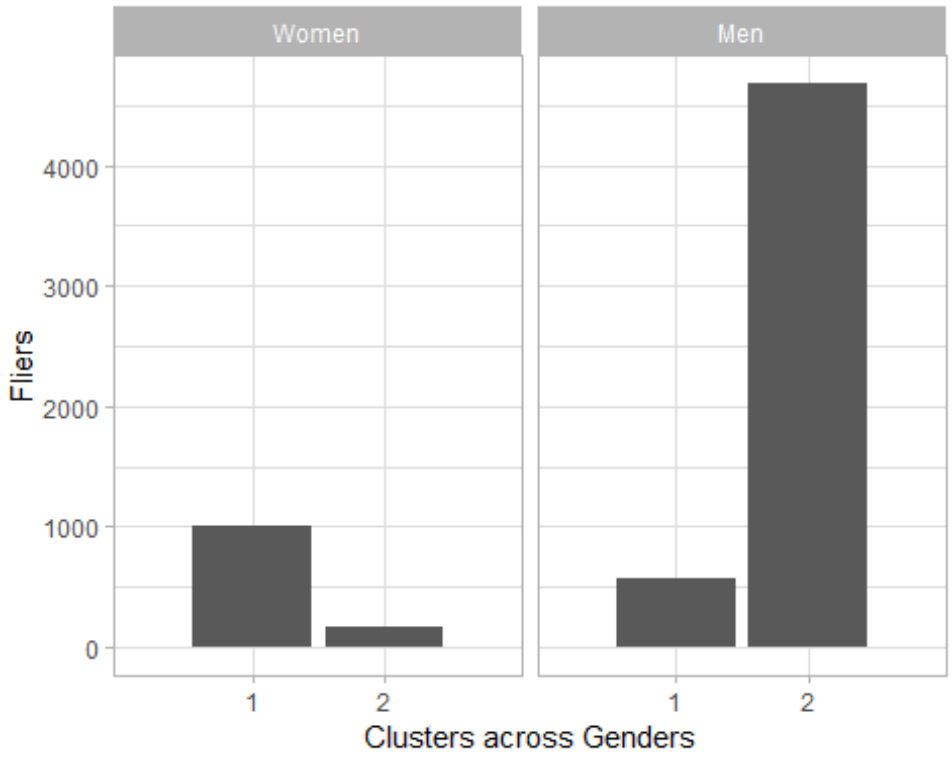
Filtering out the clusters of interest, it can be observed that Cluster 1 consists wholly of people flying first class and Cluster 2 has a mix of both

```
ggplot(sun_cluster_2, aes(x=cluster))+ facet_grid(. ~ quarter) + geom_bar()+
theme_light()+scale_x_discrete( limits=1:2)+ xlab("Clusters across Quarters")
+ ylab("Fliers")
```



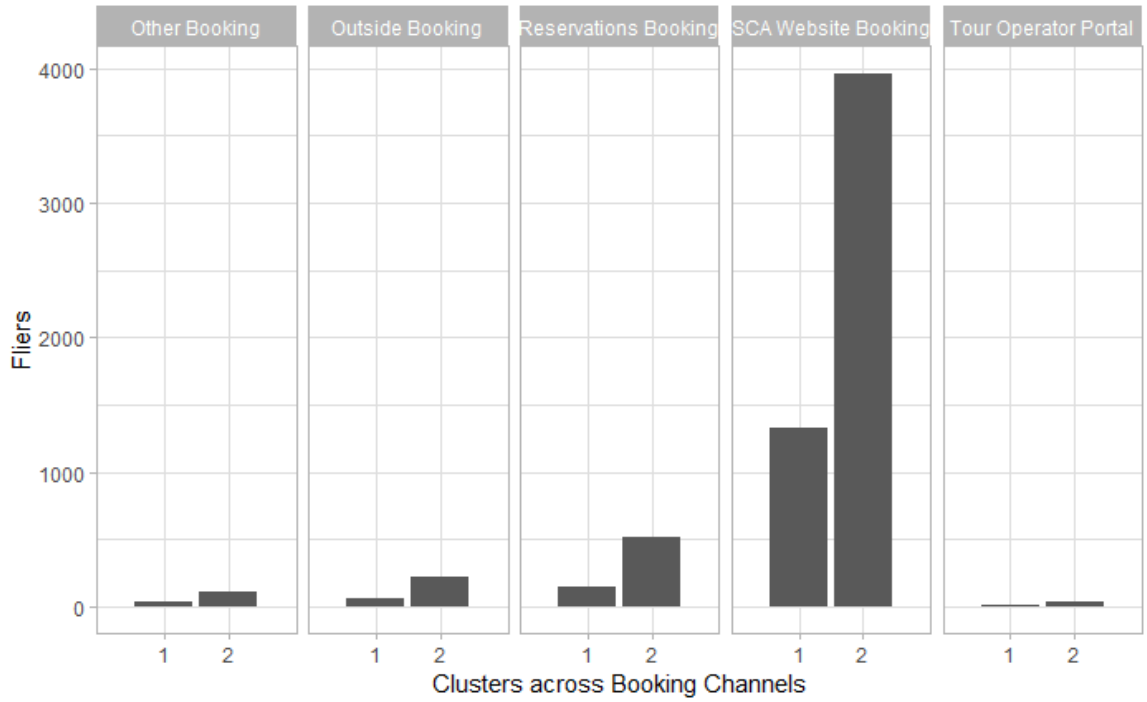
When clustering across quarters it can be observed that cluster 1 fliers fly the most in Q1. However, the gap reduces significantly over the year and is almost the same in Q4.

```
ggplot(sun_cluster_2, aes(x=cluster))+ facet_grid(. ~ GenderCode,labeller = a
s_labeller(c(`F`='Women',`M`='Men'))) + geom_bar()+ theme_light()+scale_x_dis
crete( limits=1:2)+ xlab("Clusters across Genders") + ylab("Fliers")
```



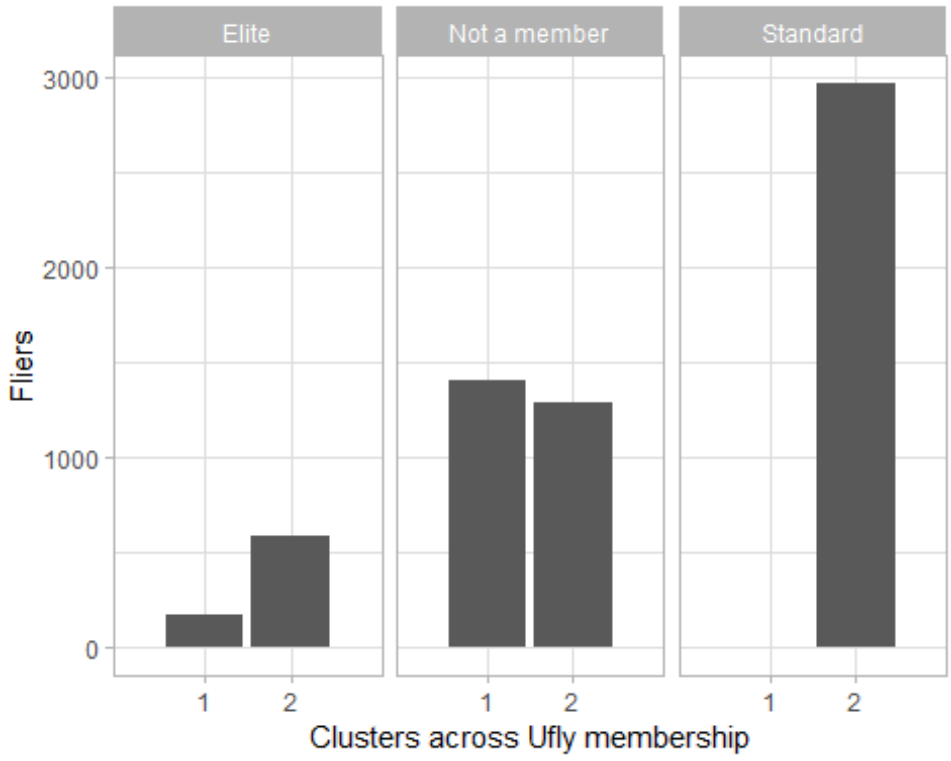
When clustering across genders, women are in higher proportion in cluster 1 while men are in higher proportion in cluster 2.

```
ggplot(sun_cluster_2, aes(x=cluster))+ facet_grid(. ~ BookingChannel) + geom_
bar() +scale_x_discrete( limits=1:2)+ theme_light()+xlab("Clusters across Boo
king Channels") + ylab("Fliers")
```



When clustered across booking channels, a significant number of fliers booked their tickets from Sun Country airline's website.

```
ggplot(sun_cluster_2, aes(x=cluster))+ facet_grid(. ~ UflyMemberStatus,labell
er = as_labeller(c(`Elite`='Elite',`Standard`='Standard', `Non_Member`='Not a
member'))) + geom_bar()+ theme_light() +scale_x_discrete( limits=1:2)+ theme_
light()+xlab("Clusters across Ufly membership") + ylab("Fliers")
```



When clustered across Ufly membership status, cluster 2 fliers are usually Standard or Elite members. While cluster 1 fliers are either Elite members or not members at all.

Interpretation of this approach:

Of the 10 clusters, we analyzed 2 clusters of interest which have the highest proportion of first-class fliers. From our analysis, customers who are most likely to buy first class tickets in the future or those which fall into the following customer segments. For both customer segments, customers should have booked their tickets from Sun Country's website

- Customer Group 1: Women who are either Elite Ufly members or not members.
- Customer Group 2: Men with Standard or Elite Ufly membership

Customer group 2 should be targetted most in the first quarter of the year and Customer group 1 in the fourth quarter.

Conclusions from this approach:

The two customer segments identified along with the timelines to best target them respectively can increase capacity utilization of First Class seats. First Class seats have high-profit margin compared to coach and are also less susceptible to competitor pricing. Hence, such targeted upgrade of customers from coach to business class can be a significant revenue generator.

Potential drawbacks:

1. An assumption is that the airline has some free capacity in the first class segment to try these strategies. Looking at the past 2 years of data, it appears the airline does have this capacity.

2. Fliers are usually priced sensitive and a competitive pricing by a competitor can significantly affect an airline's performance. We assumed that Sun Country airlines prices it's tickets competitively.

3. Instances, where the booking class and travel class are different in the data, are assumed to be instances when the customer was upgraded for free to first class.

Conclusion:

Sun Country should leverage the customer characteristics and behavior patterns identified to tackle various business problems in order to improve overall customer satisfaction, retention and loyalty towards higher profitability. The key patterns to be targeted are as follows:

1. Customers to target for UFly promotion:
    1. Oldest UFly members preferred online SCA booking and also traveled First Class more than other customers. Hence, we should target them with better offers online.
    2. Our highest spenders did not use the SCA Website booking often. We should target these group of people by incentivizing booking through our website so that we can track their online behavior.
2. Customers to target for churn reduction:
    1. Surveying female UFly members in higher age group (44+) to improve the customer experience is important as they constitute to the clusters with highest churn rate
    2. We need to promote younger customers to maintain their loyalty towards our brand
    3. We need to revisit our current pricing strategy as it could be clearly observed that the cluster with highest churn had highest price while that with lowest churn had lowest price
3. Further targeting strategy for improved marketing: First Class seats have high-profit margin compared to coach and are also less susceptible to competitor pricing. Hence, such targeted upgrade of customers from coach to business class can be significant revenue generator. Hence the segments identified can be best targeted for capacity utilization

    1. Women who are either Elite Ufly members or non-members need to be targeted for first class seats (Best targeted in fourth quarter)
    2. Men with Standard or Elite Ufly membership (Best targeted in first quarter)

4. For exhaustive marketing strategy recommendations based on natural consumer clusters, refer to 'conclusion' section of approach 1.