EDA HW 2 Technical Document

Group 2

Sun Country Exploratory Analysis

Current Problem:- The current issue at Sun Country is that customer beliefs and marketing strategies are based on anecdotal evidence, and there is relatively lesser information on customer profiles and their behaviors.

Task:- To provide recommendations for Sun Country's future digital strategies, data needs to be utilized and actionable insights need to be created to boost their performance. Hence there is a need to create customer profiles and understand UFLY membership patterns.

Before proceeding with the Analysis, we will be doing data cleaning and data transformation.

Data cleaning and Preparation

Normalizing Function - This is for Min-Max normalization of Data

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

Removing duplicate rows and rows where a different airline has been used and lastly where the gender and age of the customer are missing

```
SC = read.csv('SunCountry.csv')

#Deleting all rows that are duplicated in the dataset
SC <- SC[!duplicated(SC), ]

#remove rows with PNRLocatorID which books any non-SY trip
l<-unique(SC%>% filter(MarketingAirlineCode !="SY" )%>%select(PNRLocatorID))
SC<-SC%>% filter(!SC$PNRLocatorID %in% l$PNRLocatorID)

#if there's no inherent reasons for data to be missing
SC<-SC%>% filter(GenderCode != '')

#Final Null Check in columns
print('Final Null Check for the Columns')
```

```
## [1] "Final Null Check for the Columns"
```

```
print(colSums(is.na(SC)))
```

```
##         PNRLocatorID            TicketNum        CouponSeqNbr
##                    0                    0                   0
##     ServiceStartCity       ServiceEndCity       PNRCreateDate
##                    0                    0                   0
##     ServiceStartDate              PaxName       EncryptedName
##                    0                    0                   0
##           GenderCode          birthdateid                 Age
##                    0                    0                   0
##           PostalCode    BkdClassOfService  TrvldClassOfService
##                    0                    0                   0
##       BookingChannel          BaseFareAmt         TotalDocAmt
##                    0                    0                   0
##    UFlyRewardsNumber     UflyMemberStatus          CardHolder
##              2579112                    0                   0
##        BookedProduct           EnrollDate   MarketingFlightNbr
##                    0                    0                   0
## MarketingAirlineCode         StopoverCode
##                    0                    0
```

Removing columns not being used and transforming date columns to date format

```
#columns not used
SC<-SC %>%
  select(-StopoverCode, -TicketNum, -UFlyRewardsNumber, -MarketingAirlineCode)

#Transforming date columns
SC$PNRCreateDate<-as.Date(SC$PNRCreateDate)
SC$ServiceStartDate<-as.Date(SC$ServiceStartDate)
SC$EnrollDate<-as.Date(SC$EnrollDate)
```

Cleaning up UFLY membership columns for all customers

```
#Assume UflyRewardsNumber,UflyMemberStatus,EnrollDate,CardHolder=->NA if not a member

SC <- SC %>%
  mutate(UflyMemberStatus = ifelse(UflyMemberStatus != '',
                                   UflyMemberStatus, 'Not a Member'),
         CardHolder = ifelse(CardHolder != '', CardHolder, 'No Card'),
         BookedProduct = ifelse(BookedProduct != '',
                                BookedProduct, 'No Booked Product'),
         PostalCode = ifelse(PostalCode != '', PostalCode, -1)
  )

#Club all airports into other category
table(SC$BookingChannel)
```

```
##
##                 ANC                  BOS                  DCA
##                   9                    1                   24
##                 DFW                  FCM                  GJT
##                 503                 3513                    1
##                 HRL                  JFK                  LAN
##                  18                  254                  252
##                LAS                  LAX                  MCO
##                  177                  412                   17
##                MDW                  MIA                  MKE
##                  202                    1                  257
##                MSN                  MSP        Outside Booking
##                    1                 4788              1441750
##                PHX                  PSP   Reservations Booking
##                   21                   28               161320
##                RSW    SCA Website Booking                  SEA
##                   89              1426925                   42
##                SFO            SY Vacation Tour Operator Portal
##                  141                87278               126365
##                UFO                  XTM
##                  147                  475
```

```
SC<-SC %>%
  mutate(BookingChannel = ifelse(BookingChannel == "Outside Booking" |
                                   BookingChannel == "SCA Website Booking" |
                                   BookingChannel == "Tour Operator Portal" |
                                   BookingChannel == "Reservations Booking"|
                                   BookingChannel == "SY Vacation",
                                 BookingChannel,'Others'))
```

Analysis

We split up our analysis into three parts:-

1. Looking to develop customer segments by looking at natural customer patterns

2. Analyzing customers based on the group in which they traveled or if they traveled solo

3. We will be understanding the behavior of UFLY members vs the non UFLY members

Customer Segments

For our customer segmentation we need to convert the given dataset to a dataset where each row depicts a single customer. For this we will use a two step process. Currently as data is of each customer's every flight level, we need to first bring it down to a trip level and then from that bring it to a customer level. Along the way we add a few features and aggregate a few features.

Feature Creation

Creating a dataset where each row depicts an entire trip by a passenger Add a feature which indicates if trip was a round trip or a single way trip

```
SC_round_trips <- sqldf("
select PNRLocatorID, PaxName, CASE WHEN
       (first_value(ServiceStartCity) over (partition by PNRLocatorID,
       PaxName order by CouponSeqNbr asc) ==
        first_value(ServiceEndCity) over (partition by PNRLocatorID,
        PaxName order by CouponSeqNbr desc)
       ) THEN 1 else 0 end as round_trip
```

```
  from SC
");

SC_round_trips <- SC_round_trips[!duplicated(SC_round_trips), ]
```

Add a feature which indicates if the trip was a booked twice or for once

```
SC_repeated_trips <- sqldf("
select PNRLocatorID, PaxName, CASE WHEN
       ( COUNT(DISTINCT ServiceStartCity) == COUNT(*)
       ) THEN 0 else 1 end as repeated_trip
  from SC
  group by PNRLocatorID, PaxName
");

SC_new <- merge(x = SC_round_trips, y = SC_repeated_trips, by = c("PNRLocatorID",
                                                                  "PaxName"))
rm(SC_round_trips)
rm(SC_repeated_trips)
```

Check if the coach was upgraded during the travel and aggregating the traveled coach and booked coach columns

```
SC_upgraded_trips <- sqldf("
select PNRLocatorID, PaxName, max(upgraded_indi) as upgraded from (
  select PNRLocatorID, PaxName,
  case when BkdClassOfService == TrvldClassOfService then 0 else 1 end
  as upgraded_indi
    from SC
  )
  group by PNRLocatorID, PaxName
");


#Converting coach to an ordinal variable
SC<-SC %>%
  mutate(TravelledCoach_new =
           ifelse(TrvldClassOfService == "Coach", 0,
           ifelse(TrvldClassOfService == "Discount First Class", 1, 2)),
         BkdClassOfService_new =
           ifelse(BkdClassOfService == "Coach", 0,
           ifelse(BkdClassOfService == "Discount First Class", 1, 2)))

SC_upgrades <- sqldf("
  select PNRLocatorID, PaxName,
    max(TravelledCoach_new) as TravelledCoachAgg,
    max(BkdClassOfService_new) as BkdCoachAgg
  from SC
  group by PNRLocatorID, PaxName
")

SC_new <- merge(x = SC_new, y = SC_upgrades, by = c("PNRLocatorID", "PaxName"))
SC_new <- merge(x = SC_new, y = SC_upgraded_trips,
                by = c("PNRLocatorID", "PaxName"))

rm(SC_upgrades)
rm(SC_upgraded_trips)
```

Adding features which indicate number number of passengers travelling along with the customer and booking month, month of travel and the days between booking and flights and finally if a customer used coupons or miles, or not

```
SC <- SC %>% group_by(PNRLocatorID) %>% mutate(n_name = n_distinct(PaxName))%>%
  mutate(group_travel = ifelse(n_name>1,1,0))

#time to travel from the flight booking to the flight date
SC$time_to_travel <-
  as.integer(as.Date(as.character(SC$ServiceStartDate), format="%Y-%m-%d") -
              as.Date(as.character(SC$PNRCreateDate), format="%Y-%m-%d"))

SC <- SC %>% filter(CouponSeqNbr == 1)

#Getting the month of the year when the booking were made
SC = SC %>% mutate(booking_month = month(PNRCreateDate),
                   service_month = month(ServiceStartDate),
                   booked_product_custom =
                     ifelse(BookedProduct == 'No Booked Product', 0, 1))

SC_final = merge(x = SC, y = SC_new, by = c("PNRLocatorID", "PaxName"))
```

Further data cleaning and scaling

Outliers were removed, gender code was converted to 1 hot encoding to simplify evaluation, categorical variables were converted to factors.

```
SC <- SC_final
SC <- SC %>% filter(Age >= 0 & Age <= 130)
SC <- SC %>% filter(time_to_travel >= 0 & time_to_travel <= 365)
SC$BaseFareAmt <- log(SC$BaseFareAmt + 1)

SC_Clean_data = SC

SC_final <- SC %>% select(PNRLocatorID, PaxName, GenderCode, Age, group_travel,
                          time_to_travel, booking_month, service_month,
                          booked_product_custom,BookingChannel, round_trip,
                          repeated_trip, BkdCoachAgg, TravelledCoachAgg,
                          upgraded, UflyMemberStatus, BaseFareAmt, TotalDocAmt,
                          EncryptedName, birthdateid)

SC <- SC_final

SC[sapply(SC, is.character)] <- lapply(SC[sapply(SC, is.character)],
                                                   as.factor)
SC$group_travel <- as.factor(SC$group_travel)
SC$booked_product_custom <- as.factor(SC$booked_product_custom)
SC$round_trip <- as.factor(SC$round_trip)
```

```
SC$repeated_trip <- as.factor(SC$repeated_trip)
SC$BkdCoachAgg <- as.factor(SC$BkdCoachAgg)
SC$TravelledCoachAgg <- as.factor(SC$TravelledCoachAgg)
SC$upgraded <- as.factor(SC$upgraded)
SC <- SC %>%
  mutate(GenderCode = ifelse(GenderCode == 'F', 1, 0),
        UflyMemberStatus = ifelse(UflyMemberStatus == 'Not a Member', 0,
          ifelse(UflyMemberStatus == 'Standard', 1, 2))
  )
```

Bringing the trip level data to a customer level data

For categorical features we used mode as the aggregation and for numeric features we used mean value of the features We also added a feature that counts the total number of trips taken by a customer Grouping was done using name, Gender Code, Age to uniquely identify customers

```
getmode <- function(v) {
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v, uniqv)))]
}

SC_grpd = SC %>%
  group_by(PaxName, GenderCode, Age) %>%
  summarize(
    BkdCoachAgg = getmode(BkdCoachAgg),
    TravelledCoachAgg = getmode(TravelledCoachAgg),
    group_travel = getmode(group_travel),
    time_to_travel = mean(time_to_travel),
    booking_month = mean(booking_month),
    service_month = mean(service_month),
    booked_product_custom = getmode(booked_product_custom),
    round_trip = getmode(round_trip),
    repeated_trip = getmode(repeated_trip),
    upgraded = getmode(upgraded),
    UflyMemberStatus = getmode(UflyMemberStatus),
    BaseFareAmt = mean(BaseFareAmt),
    TotalDocAmt = mean(TotalDocAmt),
    trip_count = n()
  )
```

Further data cleaning and scaling

```
custLevel = SC_grpd

clust_final = custLevel %>%
  mutate(
         group_travel = as.numeric(group_travel),
         booked_product_custom = as.numeric(booked_product_custom ),
         round_trip = as.numeric(round_trip),
         repeated_trip  = as.numeric(repeated_trip),
         BkdCoachAgg = as.numeric(BkdCoachAgg),
         upgraded = as.numeric(upgraded))
```

Assumptions before clustering:

1. Coach, Discount First Class and First class are ordered classes.

Customer Level Clustering

Overview To understand Sun Country's customer behavior, we created different customer segments based on some features from the given dataset. Some important features which we will be focusing on are gender, age, group travel(if someone travels with 2 or more people), trip planning(number of days between booking a flight and the day of flying), travel class, UFly membership, and total amount spent.

Methodology

1. First PCA was done to reduce the number of dimenstions to 8.
2. Using the 8 dimensional data, clustering was done for clusters 2 to 10.
3. Based on SSE results with 7 clusters was chosen.

```
set.seed(42)
clust_final = clust_final[, c(1, 2, 3, 6, 7, 8, 9, 10, 11, 12, 4, 13, 14,
                              15, 16, 17)]

clust_final$group_travel = clust_final$group_travel - 1
clust_final$repeated_trip = clust_final$repeated_trip - 1
clust_final$round_trip   = clust_final$round_trip   - 1
clust_final$upgraded = clust_final$upgraded - 1
clust_final$BkdCoachAgg = clust_final$BkdCoachAgg - 1
clust_final$booked_product_custom = clust_final$booked_product_custom - 1
clust_final$UflyMemberStatus = ifelse(clust_final$UflyMemberStatus == 0, 0, 1)

clust_final_pc <- prcomp(clust_final[2:16],
                 center = TRUE,
                 scale. = TRUE)

eightColumns = clust_final_pc$x[,1:8]

eightColumnsDF = data.frame(eightColumns)

SSE_curve = c()
for(n in 2:10)
{
  kclusterp = kmeans(eightColumnsDF, n, iter.max = 50)
  sse = kclusterp$tot.withinss
  # print(n)
  SSE_curve = c(SSE_curve, sse)
}
set.seed(42)
kclusterp = kmeans(eightColumnsDF, 7)
clust_final$SevenClusters = kclusterp$cluster

clust_final_centers = clust_final[2:17] %>% group_by(SevenClusters) %>%
  summarise_all(mean)
```

```
clust_final_counts = clust_final[2:17] %>% group_by(SevenClusters) %>%
  summarise(n = n())
clust_final_centers = merge(clust_final_centers, clust_final_counts,
                            on = c("SevenClusters"))
print('Cluster centers are ')
```

```
## [1] "Cluster centers are "
```

```
print(clust_final_centers)
```

```
##   SevenClusters GenderCode      Age group_travel time_to_travel booking_month
## 1             1  0.4664537 39.81484    0.6255967       59.16285      6.656113
## 2             2  1.0000000 39.72195    0.8605607       80.39505      6.174796
## 3             3  0.0000000 38.50369    0.9049187       80.44613      6.275324
## 4             4  0.4756778 49.13280    0.5162477       53.92414      6.610347
## 5             5  0.5328632 39.41215    0.8281827       33.26565      5.700655
## 6             6  0.5145058 38.12898    0.3687849       33.69539      9.334810
## 7             7  0.4871869 39.33954    0.2666814       29.79958      4.466532
##   service_month booked_product_custom round_trip repeated_trip  BkdCoachAgg
## 1      6.766520            0.52724000  0.7463097  6.053888e-01 0.7999651588
## 2      5.811373            0.29936711  0.9407904  1.141419e-04 0.0002973696
## 3      5.856673            0.30312638  0.9275108  8.920839e-05 0.0003642676
## 4      6.567324            0.26267050  0.7994769  1.468911e-02 0.0071468759
## 5      5.660095            0.98380185  0.8319879  3.080790e-02 0.0286784022
## 6     10.235597            0.32944350  0.4693137  1.567811e-04 0.0002850566
## 7      4.687917            0.07563487  0.3183747  7.471217e-05 0.0001867804
##      upgraded UflyMemberStatus BaseFareAmt TotalDocAmt trip_count      n
## 1 0.030184078        0.2197782   5.8788753   481.75962   1.341223  86105
## 2 0.006911591        0.1984236   5.6719838   350.76880   1.223805 332919
## 3 0.005088595        0.1957641   5.6874730   357.52595   1.227500 269033
## 4 0.996335325        0.3347475   5.4794323   335.40094   1.476955  65763
## 5 0.047590787        0.1135790   0.3655498    13.33515   1.186671 114581
## 6 0.003855391        0.1373438   5.2828847   260.99749   1.266991 280646
## 7 0.002943660        0.1068832   5.2593677   260.71934   1.540569 267694
```

```
clust_final_mean = clust_final %>% summarize(across(where(is.numeric), mean))
print('Population mean is ')
```

```
## [1] "Population mean is "
```

```
print(clust_final_mean)
```

```
## # A tibble: 607,028 x 17
## # Groups:   PaxName [502,938]
##    PaxName GenderCode   Age group_travel time_to_travel booking_month
##    <fct>        <dbl> <dbl>        <dbl>          <dbl>         <dbl>
##  1 A ALZI           0  33              1              7             5
##  2 AABEAL           0  52              1              9             3
##  3 AABEAN           0  29              0              9             7
##  4 AABECA           0  50              1              0             2
##  5 AABECO           1  41.5            1             22.5           5
##  6 AABEJA           0  41            1             71            11
##  7 AABEKA           1  47.5          0.5          104.            8.5
##  8 AABEKE           0  29            0.5           61.5           2.5
##  9 AABEKI           1  15            1             82             3
## 10 AABELA           1  49            1              9             3
## # ... with 607,018 more rows, and 11 more variables: service_month <dbl>,
## #   booked_product_custom <dbl>, round_trip <dbl>, repeated_trip <dbl>,
## #   BkdCoachAgg <dbl>, upgraded <dbl>, UflyMemberStatus <dbl>,
## #   BaseFareAmt <dbl>, TotalDocAmt <dbl>, trip_count <dbl>, SevenClusters <dbl>
```

Population properties: In the given dataset, there are approximately 1.4 million customers with gender parity, and the average age is 39 years old. 62% of all customers travel in groups, and only 17% of all customers are UFly members. On average, customers usually travel in Coach class, plan their trips 55 days before flying and spend around $297.

Out of 7 clusters(customer segments) created, we will focus on 4 important customer segments.

Young Men: In this segment, 93% of passengers are Male and the average age is 36 years old. A majority of passengers in this segment travel in groups. 87% of passengers travel in groups and plan their trips well in advance. On average, these passengers book their tickets 80 days before departure.

Big Spenders: The average age of passengers in this segment is 50 years old. 37% of passengers are UFly members and usually travel in First Class. These passengers are big spenders; the average total amount spent by passengers in this segment is $616.

Impromptu Business Travelers: The average age of passengers in this segment is 38 years old. Only 26% of passengers travel in groups. These are the passengers who do not plan their trips well in advance. The average trip planning period for this group is 31 days.

Discount Travelers: In this segment, passengers are inclined to travel in groups, book close to the travel date. 84% of passengers travel in groups and book their tickets around one month before flying. They are more inclined to use their miles and other promotional discounts while booking their flights. Mostly, they do not spend even one dollar to book their tickets. This brings down the average amount spent in this segment to be $12.

Recommendations and Conclusions

1. Target elderly with more ads of Ufly Memberships. New Ufly Membership level can also be created specifically for the elderly to drive up the membership numbers and in turn the membership revenue.
2. Business Travelers also should be targeted with UFly Membership as they travel more frequently and would benefit with the membership. This would bring in more business travelers to Sun Country.
3. Discount travelers may need to make some sacrifice for low price, such as paying extra fees for luggage, boarding late etc. Sun Country can recommend combinations of trips with low price to these travelers.
4. Younger men can be targeted by directly advertising vacation packages for the vacation season months in advance with some discounts. This might increase the revenue.

Analysis by Type of Groups

Intro:

After clustering passengers by customer attributes, we also wondered if customer behavior may vary depending on what type of group they choose to travel in. To answer this question, we divided customers into three groups: 1. Families - any PNR (confirmation number) with more than one passenger, at least one of which is a child. Here, we're trying to narrow in on parents and children traveling together. 2. Solo Passengers - any PNR with one unique passenger. There could be more than one flight (ie. round trip), but there must be one unique passenger. 3. Non-family Groups - any PNR with multiple passengers, none of which are children

First, find all the PNR ID's that contain tickets for children, then flag all those PNR ID's as family travel. This information will be housed as a variable, family, in the dataframe.

```
kid = data[data$Age < 18, ]
ids = data[data$PNRLocatorID %in% kid$PNRLocatorID, ]$PNRLocatorID
data = data %>% mutate(family = ifelse(PNRLocatorID %in% ids, 1, 0))
```

Identify all the PNR ID's that belong to solo travelers and create a variable, solo, to store this info

```
# Create a unique passengerId
data = data %>% unite('passengerId', c(PaxName, GenderCode, birthdateid),
                      sep='_', remove=FALSE)

data = data %>% select(passengerId, everything())

# Find number of passengers per PNR and filter to solo flyers
numPass = data %>%
  filter(data$family == 0) %>%
  group_by(PNRLocatorID) %>%
  summarise(numPass = n_distinct(passengerId))
soloIds = numPass %>% filter(numPass == 1) %>% select(PNRLocatorID)
data = data %>% mutate(solo = ifelse(PNRLocatorID %in% soloIds$PNRLocatorID, 1, 0))
```

Lastly, create a variable, group, to indicate group travels (not a family or a solo passenger)

```
data = data %>% mutate(group = ifelse(family + solo == 0, 1, 0))
```

Create a table that summarizes various fields grouped by solo/family/group (for presentation).

```
calculate_mode <- function(x) {
  uniqx <- unique(na.omit(x))
  uniqx <- uniqx[uniqx != 'MSP']
  uniqx[which.max(tabulate(match(x, uniqx)))]
}

wkdays = c('Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday')
data$UflyMember = factor(ifelse(data$UflyMemberStatus == 'Not a Member',
                                'Not a Member', 'Member'),
                         levels=c('Member', 'Not a Member'))

data = data %>% mutate(group_type = ifelse(family == 1, 'Family',
                                           ifelse(solo == 1, 'Solo', 'Group')),
                       weekday = weekdays(as.Date(ServiceStartDate)),
                       type_of_day = ifelse(weekday %in% wkdays,
                                            'Weekday', 'Weekend'))

# Code for creating a mode function found at:
# https://exploratory.io/note/kanaugust/1701090969905358
group_summary = data %>% group_by(group_type) %>%
  summarise(age = mean(Age),
            destination = calculate_mode(as.factor(ServiceEndCity)),
            gender = calculate_mode(as.factor(GenderCode)),
            male_prop = sum(GenderCode=='M')/sum(GenderCode %in% c('M', 'F', 'U')),
            female_prop = sum(GenderCode=='F')/sum(GenderCode %in% c('M', 'F', 'U')),
            class = calculate_mode(as.factor(TrvldClassOfService)),
            Ufly_member_prop=sum(UflyMember=='Member')/sum(UflyMember
                                                           %in% c('Member',
                                                                  'Not a Member')),
            departure_day = calculate_mode(weekday),
            weekday_departure_prop = sum(type_of_day == 'Weekday')/sum(type_of_day
                                                                       %in%
                                                      c('Weekday', 'Weekend')))
```

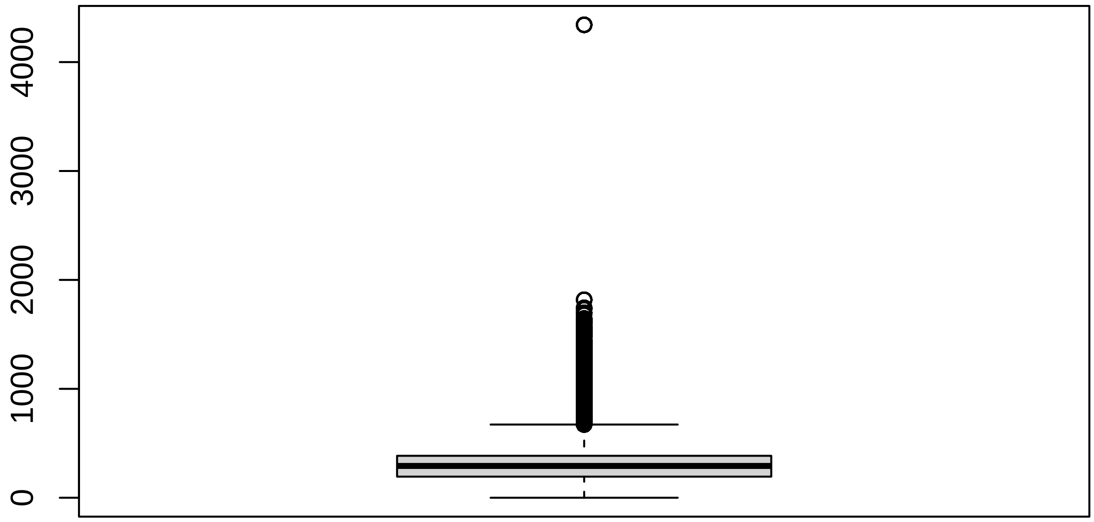
Calculate the same summary statistics for the total population and find the proportion of each group type in the population

```
pop_summary = data %>%
  summarise(age = mean(Age),
            destination = calculate_mode(as.factor(ServiceEndCity)),
            gender = calculate_mode(as.factor(GenderCode)),
            male_prop = sum(GenderCode=='M') / sum(GenderCode %in% c('M', 'F', 'U')),
            female_prop = sum(GenderCode=='F') / sum(GenderCode %in% c('M', 'F',
                                                                       'U')),
            class = calculate_mode(as.factor(TrvldClassOfService)),
            Ufly_member_prop = sum(UflyMember == 'Member')/sum(UflyMember
                                                               %in% c('Member',
                                                                'Not a Member')),
            departure_day = calculate_mode(weekday),
            weekday_departure_prop = sum(type_of_day == 'Weekday')/sum(type_of_day
                                                                       %in%
                                                        c('Weekday', 'Weekend')))

fam_solo_grp_prop = c(sum(data$family == 1) / nrow(data),
                      sum(data$solo == 1) / nrow(data),
                      sum(data$group == 1) / nrow(data))
```

Base Fare Analysis for Families

View the boxplot of Base Fare $ for families

```
boxplot(data %>% filter(family == 1) %>% select(BaseFareAmt))
```



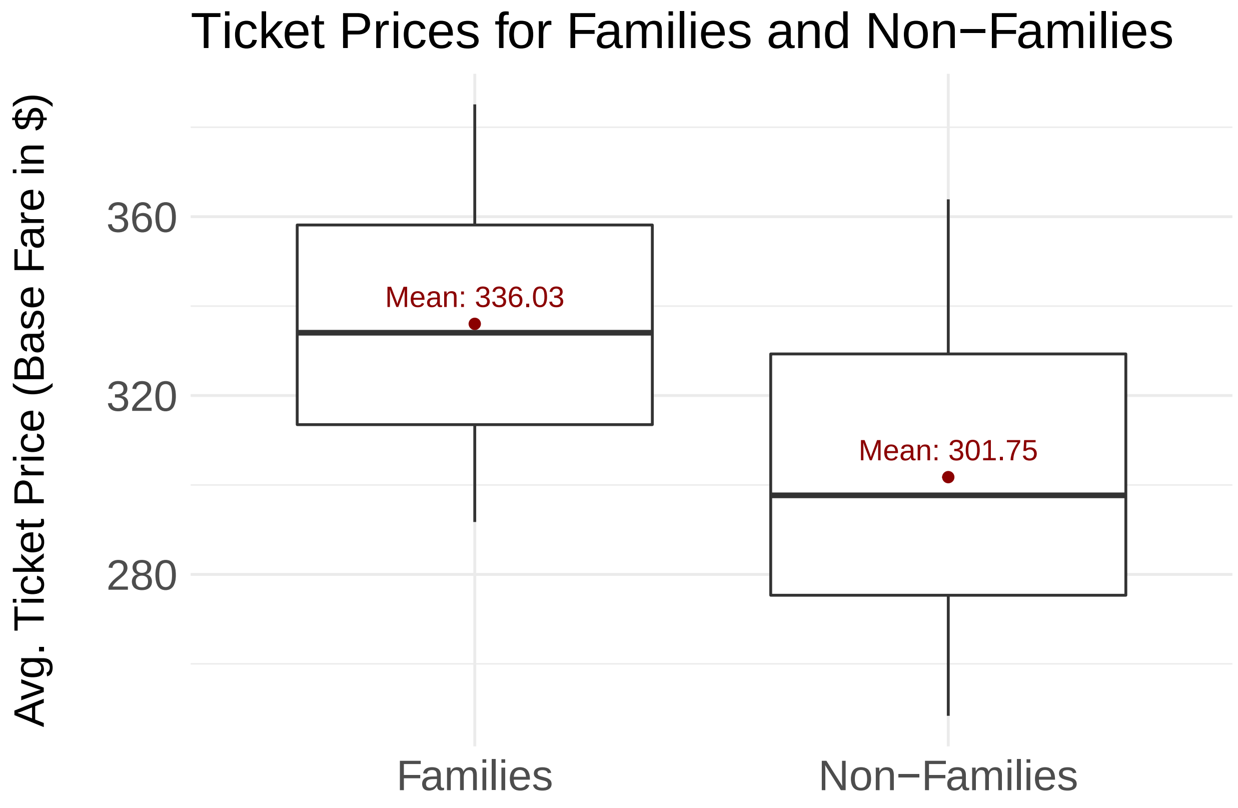
From this boxplot, we can see that there are a number of extremely high outliers that will skew further analysis. I will filter the data to 1.5x the IQR for analysis going forward.

To further clean the data, I create separate dataframes for families and non-families. I then verify that PNR ID's I'm designating as families contain more than 1 ticket. I also remove outliers for a more representative analysis.

```
# Create family and notfamily dataframes

families = data %>% filter(family == 1)
notfam = data %>% filter(family == 0)

# Make sure families have more than 1 ticket
check = families %>% select(c(PNRLocatorID, TicketNum, ServiceStartCity,
                              ServiceEndCity, GenderCode, birthdateid, Age))
tktCount = check %>% group_by(PNRLocatorID, ServiceStartCity) %>%
  summarise(tickets = n())
mult = tktCount %>% filter(tickets > 1) %>% select(PNRLocatorID)
mult = unique(mult)
families = families %>% filter(PNRLocatorID %in% mult$PNRLocatorID)

# Remove outliers
notfamAmtIQR = notfam[notfam$BaseFareAmt > 1.5 * quantile(notfam$BaseFareAmt)[2]
              & notfam$BaseFareAmt < quantile(notfam$BaseFareAmt)[4],]$BaseFareAmt
famAmtIQR = families[families$BaseFareAmt > 1.5 * quantile(families$BaseFareAmt)[2] &
            families$BaseFareAmt < quantile(families$BaseFareAmt)[4],]$BaseFareAmt
```

After further cleaning the data, we can compare boxplots for families and non-families

```
famplot = data.frame(group = 'Families', base_fare = famAmtIQR)
notfamplot = data.frame(group = 'Non-Families', base_fare = notfamAmtIQR)
plotdata = rbind(famplot, notfamplot)
# Code to combine DF's learned at
# https://stackoverflow.com/questions/26918358/ggplot2-multiple-boxplots-from-sources-of-different-leng

# ggplot text size, axis margin, and mean tips from:
# https://statisticsglobe.com/change-font-size-of-ggplot2-plot-in-r-axis-text-main-title-legend
# https://stackoverflow.com/questions/14487188/increase-distance-between-text-and-title-on-the-y-axis
# https://statisticsglobe.com/draw-boxplot-with-means-in-r
boxplot = ggplot(plotdata, aes(x = group, y = base_fare)) + geom_boxplot() +
  theme_minimal() +
  labs(x='', y='Avg. Ticket Price (Base Fare in $)',
       title='Ticket Prices for Families and Non-Families') +
  theme(text = element_text(size=16), axis.text=element_text(size=16),
        axis.title.y=element_text(margin=margin(t=0, r=20, b=0, l=0))) +
  stat_summary(fun = mean, geom = 'point', col = 'darkred') +
  stat_summary(fun = mean, geom = 'text', col = 'darkred',
               vjust = -.8, aes(label=paste('Mean:', round(..y.., digits = 2))))

plot(boxplot)
```



Here, we see that families spend significantly more per ticket than non-families on average. I will verify the significance of these results with a t-test.

```
t.test(notfamAmtIQR, famAmtIQR)
```

```
##
##  Welch Two Sample t-test
##
## data:  notfamAmtIQR and famAmtIQR
## t = -490.83, df = 372880, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -34.41820 -34.14441
## sample estimates:
## mean of x mean of y
##  301.7510  336.0324
```

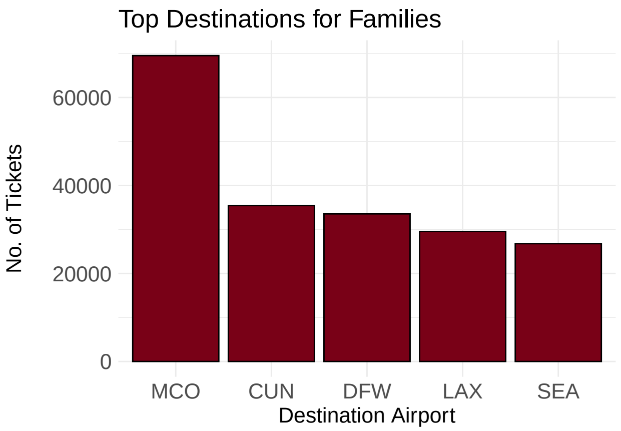
Conclusion: Based on these results, we can see that families spend ~ $35 more per ticket than non-families on average, with statistical significance.

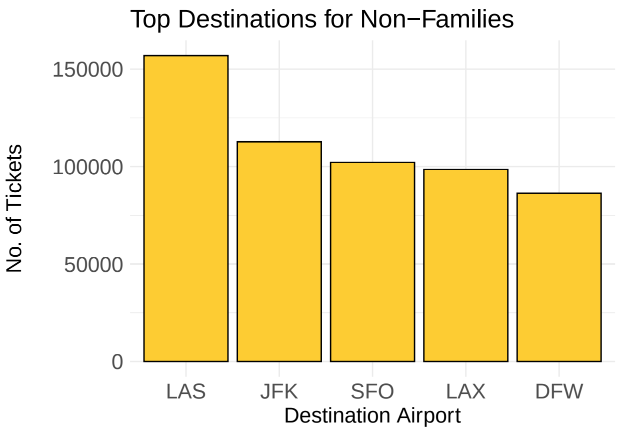
Frequent Destinations for Families

Next, I will generate plots of the most popular destinations for families and non-families to see where they are most similar and different.

```
# Find the top destinations for families
topFamCities = top_n(families %>% group_by(ServiceEndCity) %>%
                       summarise(tktCount = n()), 6)
topFamCities = topFamCities %>% filter(ServiceEndCity != 'MSP')
topFamCities = topFamCities[order(-topFamCities$tktCount), ]
famPlot = ggplot(topFamCities, aes(x = reorder(ServiceEndCity, -tktCount, sum),
                                   y = tktCount)) +
  geom_col(fill='#790117', colour='black') + theme_minimal() +
  labs(x='Destination Airport',
       y='No. of Tickets',
       title='Top Destinations for Families') +
  theme(text = element_text(size=16), axis.text=element_text(size=16),
        axis.title.y=element_text(margin=margin(t=0, r=20, b=0, l=0)))


# Find the top destinations for non-families
topNotFamCities = top_n(notfam %>% group_by(ServiceEndCity) %>%
                          summarise(tktCount = n()), 6)
topNotFamCities = topNotFamCities %>% filter(ServiceEndCity != 'MSP')
topNotFamCities = topNotFamCities[order(-topNotFamCities$tktCount), ]
notfamPlot = ggplot(topNotFamCities,
                    aes(x = reorder(ServiceEndCity, -tktCount, sum),
                        y = tktCount)) +
  geom_col(fill='#FDCC33', colour='black') + theme_minimal() +
  labs(x='Destination Airport',
       y='No. of Tickets',
       title='Top Destinations for Non-Families') +
  theme(text = element_text(size=16), axis.text=element_text(size=16),
        axis.title.y=element_text(margin=margin(t=0, r=20, b=0, l=0)))
```

```
# Show Plots
plot(famPlot)
plot(notfamPlot)
```





Conclusion: Based on these plots, we can see that families are traveling to warmer locations such as Orlando, Cancun, Dallas, and LA. These are typically thought of as vacation destinations. Alternatively, non-families are more likely to travel to large cities such as Las Vegas, New York, San Francisco, and LA. Knowing this can help to provide better experiences for passengers. For example, flights to warm vacation destinations could provide more kids activities/movies whereas flights to larger cities could provide different in-flight content more geared towards adults and business travelers.

Base Fare Analysis for Solo Passengers

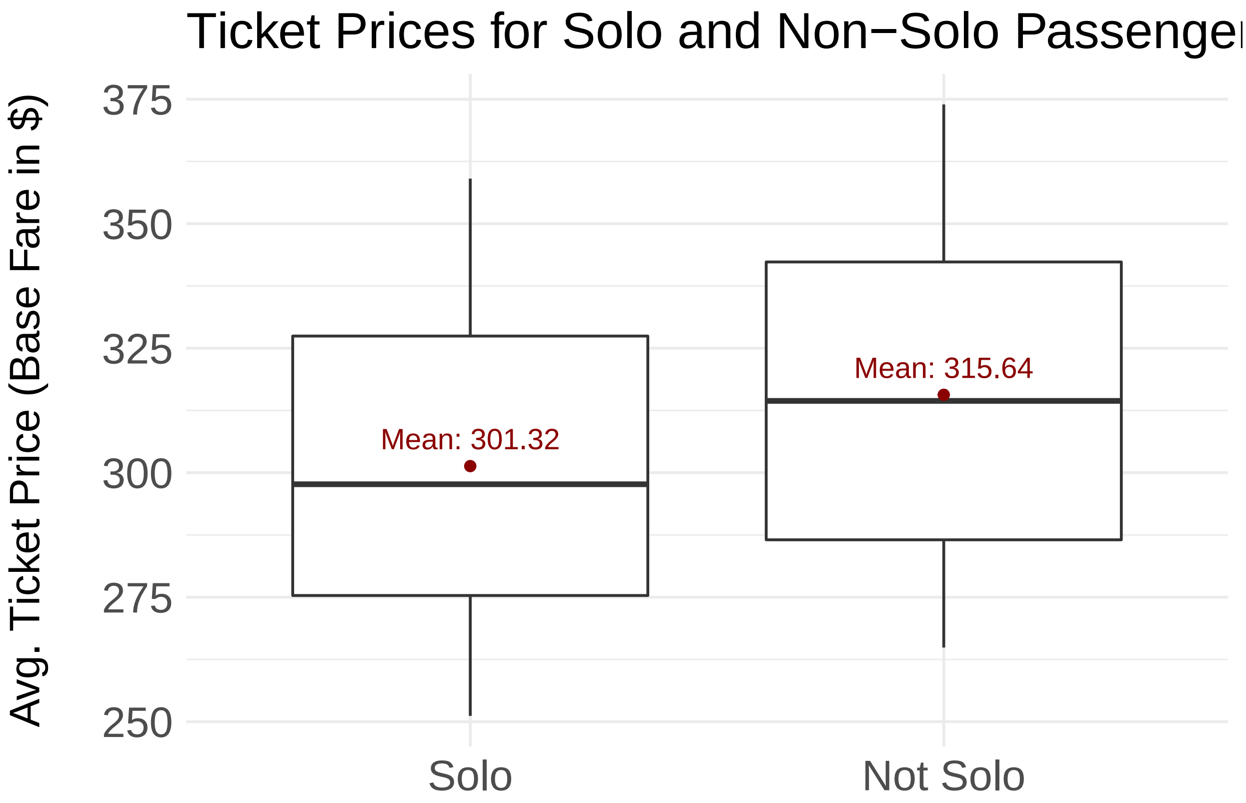
Create separate dataframes for solo passengers and non-solo passengers and remove outliers

```
solo = data %>% filter(solo == 1)
notsolo = data %>% filter(solo == 0)

# Remove outliers
notsoloAmtIQR = notsolo[notsolo$BaseFareAmt > 1.5 * quantile(notsolo$BaseFareAmt)[2] &
                notsolo$BaseFareAmt < quantile(notsolo$BaseFareAmt)[4],]$BaseFareAmt
soloAmtIQR = solo[solo$BaseFareAmt > 1.5 * quantile(solo$BaseFareAmt)[2] &
                  solo$BaseFareAmt < quantile(solo$BaseFareAmt)[4],]$BaseFareAmt
```

After further cleaning the data, we can compare boxplots for solo passengers and non-solo passengers

```
soloplot = data.frame(group = 'Solo', base_fare = soloAmtIQR)
notsoloplot = data.frame(group = 'Not Solo', base_fare = notsoloAmtIQR)
plotdata = rbind(soloplot, notsoloplot)
plotdata$group = factor(plotdata$group, levels=c('Solo', 'Not Solo'))
# Code to combine DF's learned at
# https://stackoverflow.com/questions/26918358/ggplot2-multiple-boxplots-from-sources-of-different-leng

# ggplot text size, axis margin, and mean tips from:
# https://statisticsglobe.com/change-font-size-of-ggplot2-plot-in-r-axis-text-main-title-legend
# https://stackoverflow.com/questions/14487188/increase-distance-between-text-and-title-on-the-y-axis
# https://statisticsglobe.com/draw-boxplot-with-means-in-r
```

```
boxplot = ggplot(plotdata, aes(x = group, y = base_fare)) + geom_boxplot() +
  theme_minimal() +
  labs(x='', y='Avg. Ticket Price (Base Fare in $)',
       title='Ticket Prices for Solo and Non-Solo Passengers') +
  theme(text = element_text(size=16), axis.text=element_text(size=16),
        axis.title.y=element_text(margin=margin(t=0, r=20, b=0, l=0))) +
  stat_summary(fun = mean, geom = 'point', col = 'darkred') +
  stat_summary(fun = mean, geom = 'text', col = 'darkred',
               vjust = -.8, aes(label=paste('Mean:', round(..y.., digits = 2))))

plot(boxplot)
```



Here, we see that solo passengers spend less per ticket than non-solo passengers on average. I will verify the significance of these results with a t-test.

```
t.test(notsoloAmtIQR, soloAmtIQR)
```

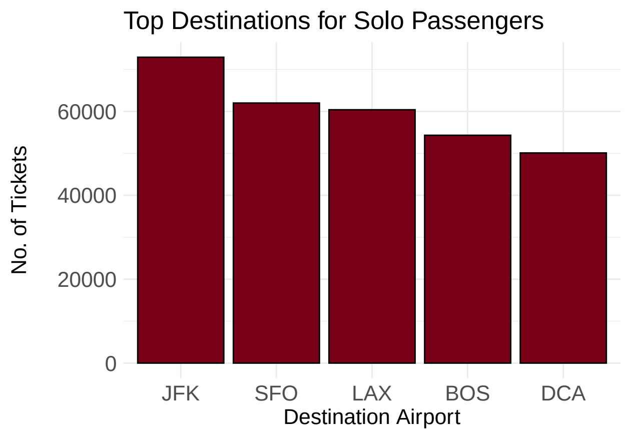
```
##
##  Welch Two Sample t-test
##
## data:  notsoloAmtIQR and soloAmtIQR
## t = 218.67, df = 733674, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  14.18870 14.44535
## sample estimates:
## mean of x mean of y
##  315.6408  301.3238
```

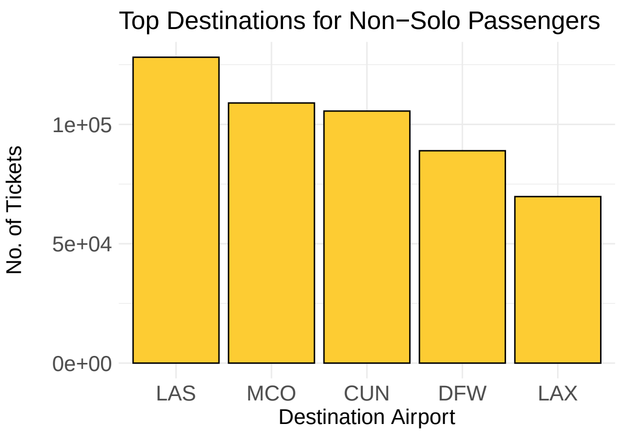
Conclusion: Based on these results, we can see that solo passengers spend ~ $15 less per ticket than passengers that travel with others, on average.

Frequent Destinations for Solo Passengers

Next, I will generate plots of the most popular destinations for solo and non-solo passengers to see where they are most similar and different.

```
# Find the top destinations for solo passengers
topSoloCities = top_n(solo %>% group_by(ServiceEndCity) %>%
                        summarise(tktCount = n()), 6)
topSoloCities = topSoloCities %>% filter(ServiceEndCity != 'MSP')
topSoloCities = topSoloCities[order(-topSoloCities$tktCount), ]
soloPlot = ggplot(topSoloCities, aes(x = reorder(ServiceEndCity, -tktCount, sum),
                                   y = tktCount)) +
  geom_col(fill='#790117', colour='black') + theme_minimal() +
  labs(x='Destination Airport',
       y='No. of Tickets',
       title='Top Destinations for Solo Passengers') +
  theme(text = element_text(size=16), axis.text=element_text(size=16),
        axis.title.y=element_text(margin=margin(t=0, r=20, b=0, l=0)))


# Find the top destinations for non-solo passengers
topNotSoloCities = top_n(notsolo %>% group_by(ServiceEndCity) %>%
                           summarise(tktCount = n()), 6)
topNotSoloCities = topNotSoloCities %>% filter(ServiceEndCity != 'MSP')
topNotSoloCities = topNotSoloCities[order(-topNotSoloCities$tktCount), ]
notsoloPlot = ggplot(topNotSoloCities,
                    aes(x = reorder(ServiceEndCity, -tktCount, sum),
                        y = tktCount)) +
  geom_col(fill='#FDCC33', colour='black') + theme_minimal() +
  labs(x='Destination Airport',
       y='No. of Tickets',
       title='Top Destinations for Non-Solo Passengers') +
  theme(text = element_text(size=16), axis.text=element_text(size=16),
        axis.title.y=element_text(margin=margin(t=0, r=20, b=0, l=0)))

# Show Plots
plot(soloPlot)
plot(notsoloPlot)
```





Conclusion: Based on these plots, we can see that solo passengers tend to travel to big cities with big business presences. These include New York, San Francisco, Los Angeles, Boston, and Washington D.C. We can infer from these results that solo travelers on Sun Country tend to be business passengers rather than vacationers.

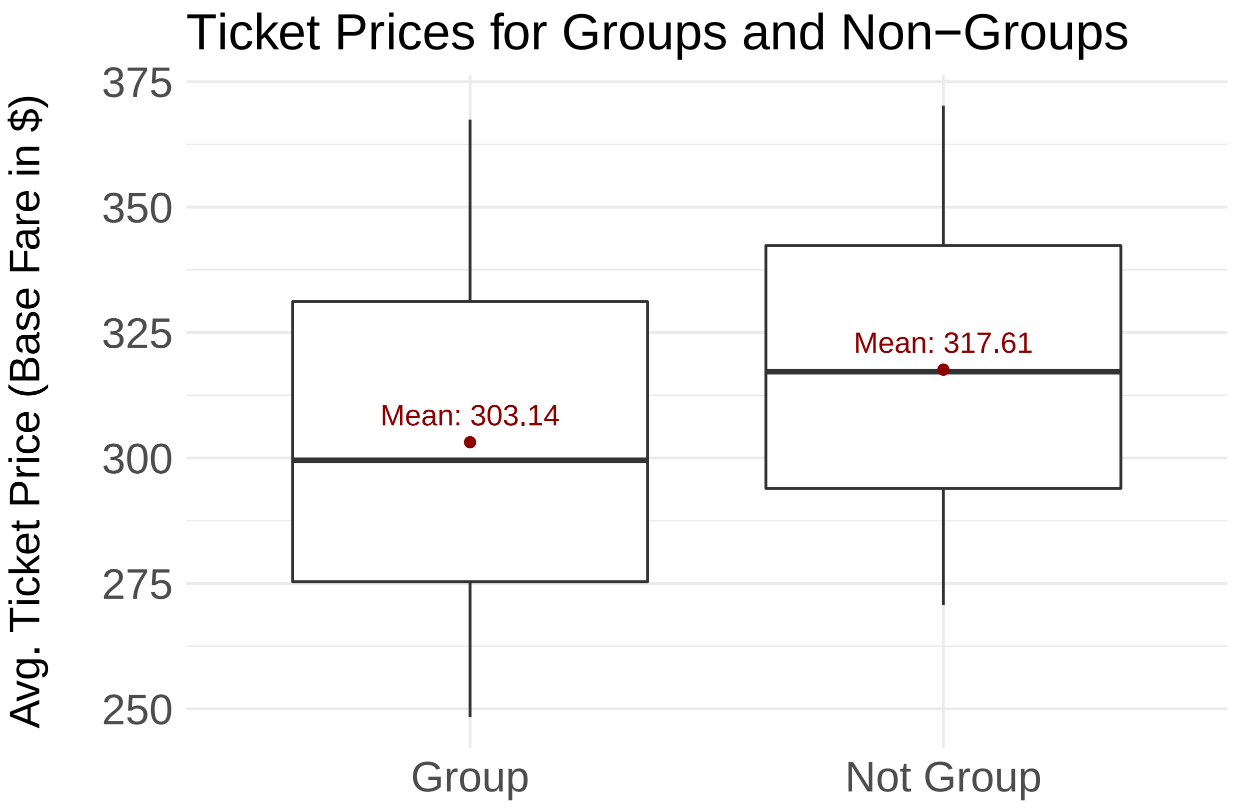
Base Fare Analysis for Groups

Create separate dataframes for groups and non-groups (solo and family) and remove outliers

```
grp = data %>% filter(group == 1)
notgrp = data %>% filter(group == 0)

# Remove outliers
notgrpAmtIQR = notgrp[notgrp$BaseFareAmt > 1.5 * quantile(notgrp$BaseFareAmt)[2] &
                notgrp$BaseFareAmt < quantile(notgrp$BaseFareAmt)[4],]$BaseFareAmt
grpAmtIQR = grp[grp$BaseFareAmt > 1.5 * quantile(grp$BaseFareAmt)[2] &
                  grp$BaseFareAmt < quantile(grp$BaseFareAmt)[4],]$BaseFareAmt
```

After further cleaning the data, we can compare boxplots for groups and non-groups

```
grpplot = data.frame(group = 'Group', base_fare = grpAmtIQR)
notgrpplot = data.frame(group = 'Not Group', base_fare = notgrpAmtIQR)
plotdata = rbind(grpplot, notgrpplot)
plotdata$group = factor(plotdata$group, levels=c('Group', 'Not Group'))
# Code to combine DF's learned at
# https://stackoverflow.com/questions/26918358/ggplot2-multiple-boxplots-from-sources-of-different-leng

# ggplot text size, axis margin, and mean tips from:
# https://statisticsglobe.com/change-font-size-of-ggplot2-plot-in-r-axis-text-main-title-legend
# https://stackoverflow.com/questions/14487188/increase-distance-between-text-and-title-on-the-y-axis
# https://statisticsglobe.com/draw-boxplot-with-means-in-r
boxplot = ggplot(plotdata, aes(x = group, y = base_fare)) + geom_boxplot() +
  theme_minimal() +
  labs(x='', y='Avg. Ticket Price (Base Fare in $)',
       title='Ticket Prices for Groups and Non-Groups') +
  theme(text = element_text(size=16), axis.text=element_text(size=16),
        axis.title.y=element_text(margin=margin(t=0, r=20, b=0, l=0))) +
  stat_summary(fun = mean, geom = 'point', col = 'darkred') +
  stat_summary(fun = mean, geom = 'text', col = 'darkred',
               vjust = -.8, aes(label=paste('Mean:', round(..y.., digits = 2))))

plot(boxplot)
```



Here, we see that groups spend less per ticket than non-groups on average. I will verify the significance of these results with a t-test.

```
t.test(notgrpAmtIQR, grpAmtIQR)
```

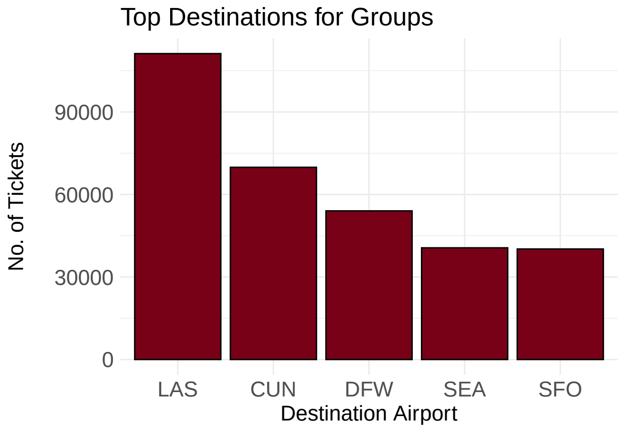
```
##
##  Welch Two Sample t-test
##
## data:  notgrpAmtIQR and grpAmtIQR
## t = 219.41, df = 783480, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  14.34028 14.59880
## sample estimates:
## mean of x mean of y
##  317.6087  303.1391
```

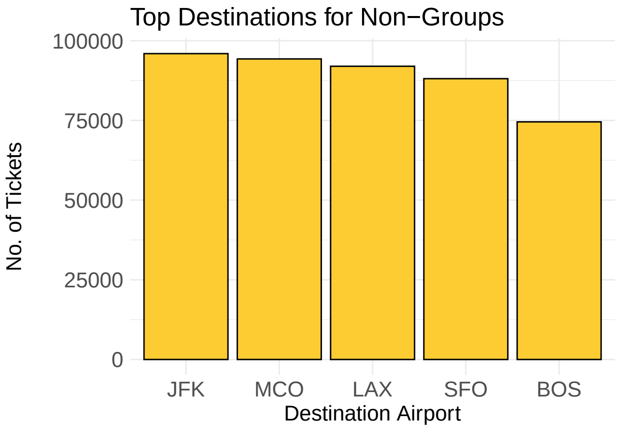
Conclusion: Based on these results, we can see that groups spend ~ $14 less per ticket than non-groups (combination of families and solo travelers), on average.

Frequent Destinations for Groups

Next, I will generate plots of the most popular destinations for groups and non-groups to see where they are most similar and different.

```r
# Find the top destinations for groups
topGrpCities = top_n(grp %>% group_by(ServiceEndCity) %>%
                       summarise(tktCount = n()), 6)
topGrpCities = topGrpCities %>% filter(ServiceEndCity != 'MSP')
topGrpCities = topGrpCities[order(-topGrpCities$tktCount), ]
grpPlot = ggplot(topGrpCities, aes(x = reorder(ServiceEndCity, -tktCount, sum),
                                   y = tktCount)) +
  geom_col(fill='#790117', colour='black') + theme_minimal() +
  labs(x='Destination Airport',
       y='No. of Tickets',
       title='Top Destinations for Groups') +
  theme(text = element_text(size=16), axis.text=element_text(size=16),
        axis.title.y=element_text(margin=margin(t=0, r=20, b=0, l=0)))


# Find the top destinations for non-groups
topNotGrpCities = top_n(notgrp %>% group_by(ServiceEndCity) %>%
                          summarise(tktCount = n()), 6)
topNotGrpCities = topNotGrpCities %>% filter(ServiceEndCity != 'MSP')
topNotGrpCities = topNotGrpCities[order(-topNotGrpCities$tktCount), ]
notgrpPlot = ggplot(topNotGrpCities,
                    aes(x = reorder(ServiceEndCity, -tktCount, sum),
                        y = tktCount)) +
  geom_col(fill='#FDCC33', colour='black') + theme_minimal() +
  labs(x='Destination Airport',
       y='No. of Tickets',
       title='Top Destinations for Non-Groups') +
  theme(text = element_text(size=16), axis.text=element_text(size=16),
        axis.title.y=element_text(margin=margin(t=0, r=20, b=0, l=0)))

# Show Plots
plot(grpPlot)
plot(notgrpPlot)
```

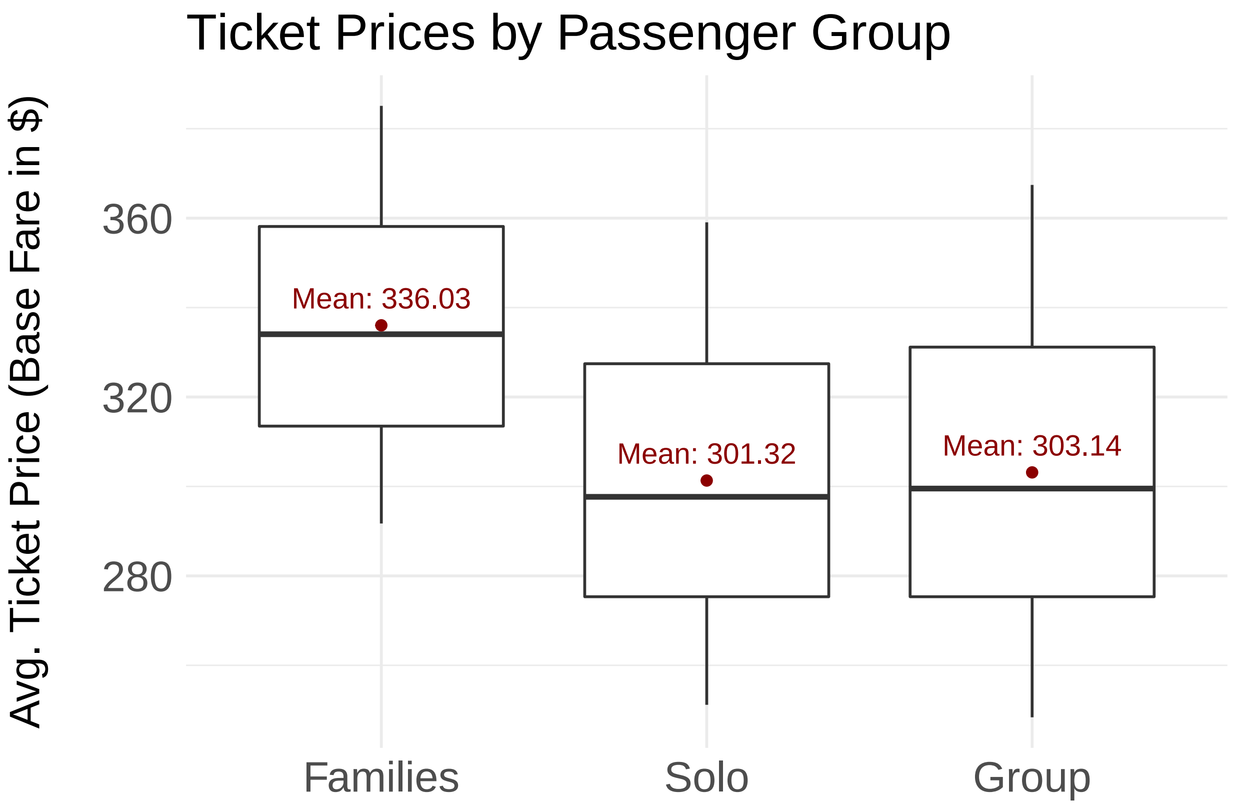




Conclusion: Based on these plots, we can see that the most popular destination for non-family groups is Las Vegas by far.

Base Fare Comparison of all Three Groups

```r
plotdata = rbind(famplot, soloplot, grpplot)
plotdata$group = factor(plotdata$group, levels=c('Families', 'Solo', 'Group'))
# Code to combine DF's learned at
# https://stackoverflow.com/questions/26918358/ggplot2-multiple-boxplots-from-\
#sources-of-different-lengths

# ggplot text size, axis margin, and mean tips from:
# https://statisticsglobe.com/change-font-size-of-ggplot2-plot-in-r-axis-text-\
#main-title-legend
# https://stackoverflow.com/questions/14487188/increase-distance-between-\
#text-and-title-on-the-y-axis
# https://statisticsglobe.com/draw-boxplot-with-means-in-r
boxplot = ggplot(plotdata, aes(x = group, y = base_fare)) + geom_boxplot() +
  theme_minimal() +
  labs(x='', y='Avg. Ticket Price (Base Fare in $)',
       title='Ticket Prices by Passenger Group') +
  theme(text = element_text(size=16), axis.text=element_text(size=16),
        axis.title.y=element_text(margin=margin(t=0, r=20, b=0, l=0))) +
  stat_summary(fun = mean, geom = 'point', col = 'darkred') +
  stat_summary(fun = mean, geom = 'text', col = 'darkred',
               vjust = -.8, aes(label=paste('Mean:', round(..y.., digits = 2))))

plot(boxplot)
```



Conclusion: From this boxplot, we can see that families spend ~$35 more per ticket than any other passenger group. Additionally, the rectangles represent 75% of the data for each group. The entire rectangle for families lies above the average for the total population, meaning that 75% of tickets sold to families cost more than the average for the population. This shows that families are a lucrative population and Sun Country should work to maximize their market share among this passenger group.

Group Analysis Conclusion:

After analyzing the differences between these three types of travel groups (families, solo passengers, and non-family groups), we recommend that Sun Country develop vacation packages to attract more family vacation travelers. Families tend to spend more per ticket flying to warm destinations such as Orlando and Cancun, so Sun Country could partner with organizations like Disney World to develop vacation packages. Additionally, they could add in-flight activities for children on these flights to help relieve stress for parents.

We also found that the solo traveler group aligns closely with the business traveler customer segment identified via clustering. This group flies mostly on weekdays to large cities, has higher Ufly membership, and spends less per ticket than other customer groups. Sun Country should continue to focus on Ufly membership among this group since they already have an affinity for the program (higher membership than any other group) and push business-oriented upgrades (ie. wifi) during the weekday flights to large cities to increase revenue.

Analysis 3: An Overview about Impact of Ufly Membership

Intro:

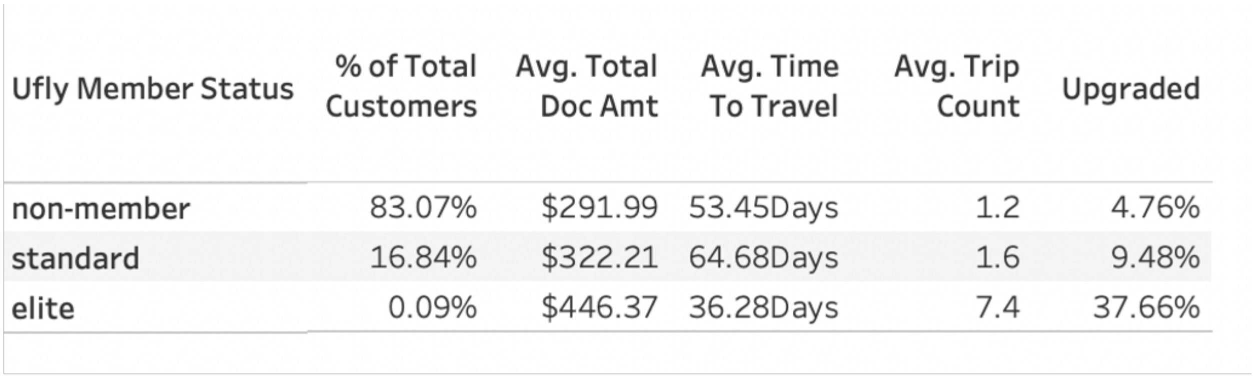
As stated in the case narrative, Sun Country wants to drive enrollment in Ufly Reward because this loyalty program will help build brand loyalty among travelers and win repeat business. This program has operated for some time, and we are curious if this program is effective and profitable. If not, Sun Country can turn to other methods. If yes, we want to know how we can convert non-members to members and choose what kind of people to convert. Previous clustering and group analysis show that ufly membership percentage within each group is an important attribute. And from which we can have a sense that we need to focus on which group and take tailored strategy to attract them into this program.

Ufly Members' Profile

Description and Rationale for the Chosen Analysis
We do this analysis because we want to get a basic understanding of customers with different membership statuses.

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

```r
knitr::include_graphics('Picture1.png')
```



Ufly Member Status	% of Total Customers	Avg. Total Doc Amt	Avg. Time To Travel	Avg. Trip Count	Upgraded
non-member	83.07%	$291.99	53.45Days	1.2	4.76%
standard	16.84%	$322.21	64.68Days	1.6	9.48%
elite	0.09%	$446.37	36.28Days	7.4	37.66%

Conclusion: We can see that members count for around 17% of total customers, but they spend more, their lead time is shorter, they fly more frequently, and they upgrade more.

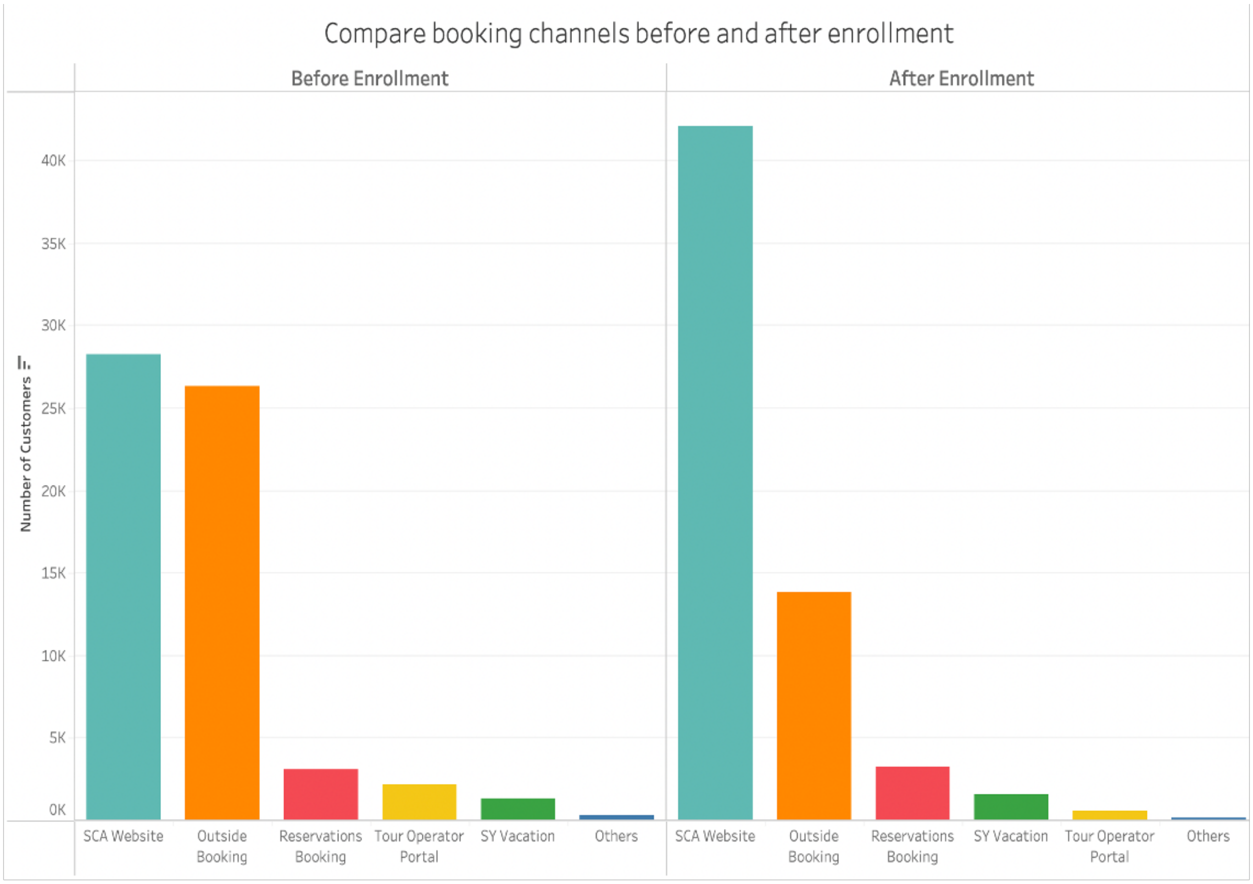
More specifically, elite members count for only 0.1%, but they spend the most, and their lead time is the shortest. This may be because they are less price-sensitive, so they do not need to book beforehand for the lower ticket price. In terms of average trip count, elite members fly significantly higher than the other two kinds of customers. Lastly, the upgraded percentage of elite members is also considerably higher. The 37% means that in the elite member subgroup, 37% of people get upgraded most times.

Compare UFly-Members before and after Enrollment

Description and Rationale for the Chosen Analysis
We do this analysis because we are curious about whether customers will change their behavior after becoming ufly-member?

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

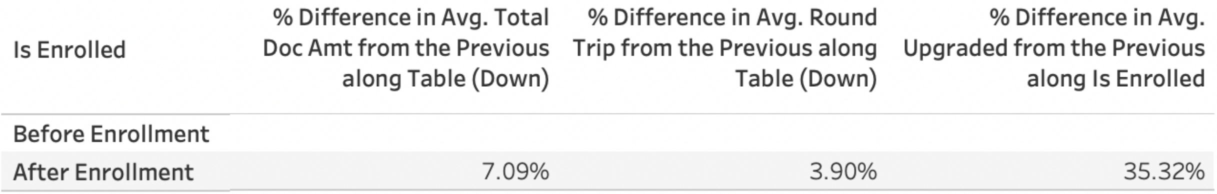
```r
knitr::include_graphics('Picture2.png')
```



Conclusion: Compared with before, they are more inclined to use official website booking and significantly reduce outside booking use. We know that the airline industry is competitive and price volatile. Outside booking apps can recommend all possible trip combinations provided by different airline companies to customers, and customers often compare and choose a flight with the lowest price from them. Under these circumstances, they still choose to use official website booking, showing that Ufly-program increases customer loyalty.

[Graphs generated from Tableau]

```r
knitr::include_graphics('pic3.png')
```



Is Enrolled	% Difference in Avg. Total Doc Amt from the Previous along Table (Down)	% Difference in Avg. Round Trip from the Previous along Table (Down)	% Difference in Avg. Upgraded from the Previous along Is Enrolled
Before Enrollment			
After Enrollment	7.09%	3.90%	35.32%

Conclusion: Compared with before, their pent amount increased 7%, round trip booking increased 3.9% (booking round trip increase average revenue per customer than one-way trip), upgrade increased 35.3%

(boosting their journey by paying extra fee increases average revenue per customer than not) Overall, we draw the conclusion that ufly-program indeed increases customer loyalty and is lucrative.

Converting Non-Members to UFly-Members

Description and Rationale for the Chosen Analysis
After we recognize that the ufly-program is a profit machine to the company, we wonder if there are some methods to drive non-member customers to enroll in this program?

Execution and Results

```
#Reading trip level data
tripdata_clean <- SC_Clean_data # read.csv("trip_data_clean.csv")
```

```
SC_t <- tripdata_clean %>% mutate(UflyMemberStatus = ifelse(UflyMemberStatus == "Not a Member", 0, 1))
```

```
#Computing the difference of date of UFly Membership enrollment and date of booking a flight
SC_t<-SC_t%>%
  mutate(gap=abs(as.integer(as.Date(as.character(EnrollDate), format="%Y-%m-%d") -
                                   as.Date(as.character(PNRCreateDate), format="%Y-%m-%d"))))
```

```
SC_t <- SC_t %>%
  group_by(PaxName, GenderCode, Age)%>%
  filter(sum(UflyMemberStatus)!=0)  ##filter out groups with all memberstatus are non-member
```

```
SC_t1 <- SC_t %>%
  group_by(PaxName, GenderCode, Age)%>%
  summarise(min_gap=min(gap))
```

```
s<-SC_t1%>%subset(min_gap==0)
dim(s)/dim(SC_t1)
```

```
## [1] 0.1012033 1.0000000
```

Conclusion: From the data, we noticed that 10% of Ufly Members enroll for the membership on the day when they are booking their flights. This is an unignorable figure. Why do they book on the same day? We conjecture that sun country has incorporated some product features. For example, when the customers are booking their flights. They may receive advertisement to introduce advantages about ufly-members, which successfully attracts them.

Recommendation: So Sun Country Airlines should target passengers with UFly-Membership offers while they are booking a flight.Thus, we can get some insights from our observations: to convert non-members, what does 'target' mean here is, if Sun Country Airlines is not doing it already, It could be a membership offer "pop-up" with lucrative offers like free upgrades or give an option to opt for membership when a customer is making the payment.

Ufly Membership Analysis Conclusion:

1. Compared with non-members, members count for only a tiny proportion of total customers, but they spend more, their lead time is shorter, they fly more frequently, and upgrade more.

2. Ufly-program increases customer loyalty and is lucrative.
3. Sun Country can convert non-members by showing Membership enrollment “pop-up” with lucrative offers.