Clustering and Segmentation of Ticketing Data

Management Science Senior Design Project

Hunter Ross
Mary Liz Tuttle
Ramon Trespalacios

May 8th, 2014

I. EXECUTIVE SUMMARY

Sabre Airline Solutions offers data solutions and software to aid airlines sell products, market themselves, and operate efficiently. One of their services is to provide airline reservation systems and help them make smarter operational decisions that personalize products to travelers.

The company would like to provide traveler segmentation services for their customer reservation system to support various marketing programs. The Senior Design group, Team Sabre, was charged with creating segmentation rules that classify clusters of data provided by the client, Sabre Airline Solutions. The data provided represents tickets purchased through the company and shows specific characteristics of each purchase.

Team Sabre replicated the data to create pre-booking and post-booking results. Pre-booking segmentation will show clusters that do not include variable such as fare and travel time, because these can't be known until after booking. On the other hand, post-booking data will provide segments that include purchases made. Pre-booking clusters could be used to make promotions for customers while booking, and post-booking clusters could be used to make promotions after booking.

The team used k-means clustering in R to find the optimal amount of clusters in the data provided. The analysis of the output followed the segmentation rules created by Team Sabre and provided the client with clustering classifications.

Ross Darrow, Senior Principal at Sabre, and Dick Barr, Associate Professor and Chair of the Engineering, Management, Information, and Systems department, served as advisors to the team. With their help, the team was able to identify the best clustering method and classify each segment within the deadline. Team Sabre hopes Sabre Airline Solutions will utilize these segment classification methodology for future projects.

II. BACKGROUND & DESCRIPTION OF SITUATION

Sabre Airline Solutions would like for SMU's Team Sabre to provide traveler segmentation services for their customer reservation system in order to provide solutions or strategies for various marketing programs. The data provided will include rows representing each ticket, along with columns describing variable fields.

These variables include:

- Booking date, departure date, return date (and day-of-week)
- Fare or fare ratio (ratio to lowest fare)
- Travel time or ratio (ratio to non-stop time)
- Market
- Travel agency type (big global, small, online agency)

In order to properly segment and analyze the customer data, we will use k-means clustering in R – programming language to discover the optimal number of clusters for our data set. Once we find the optimal number of clusters, we will then interpret the output and provide subjective analysis to determine any conclusions.

III. ANALYSIS OF SITUATION

The problem that we are addressing is to break down a large set of data into clusters so that we can create specific sets of rules for the ticketing segmentation. Our team is looking at both pre-booking and post-booking ticketing information for 14 specific days. The accumulation of data added up to 2,535,955 tickets. We chose k-means clustering, which aims to partition 'n' observations into 'k' clusters in which each observation belongs to the cluster with the nearest mean. After deciding the optimal number of clusters, we must classify each cluster through the variables that we believe give us the best insights and targets into the type of traveler we are dealing with. To classify the clusters, we will look at variables that show the highest standard deviations away from the mean. Along with the standardized variables, there are certain fields that will give us percentages of each cluster who departed or returned home on a certain day. The variables that we will use in both our pre-booking and post-booking segmentation are listed below. Please note that a large standard deviation above the mean gives one result, while a large standard deviation below the mean gives the opposite.

Our segmentation will provide insights into the design of good fare products. For example, if an airline has created a ticket fare product for a specific market like business-travelers, our segmentation can confirm if the product is well-defined or well-targeted. Aside from creating a set of rules to better define and target ticket fares, our goals are:

- Establish strong and trusted relationship with Sabre client
- Provide solutions or proposals for Sabre's various marketing programs

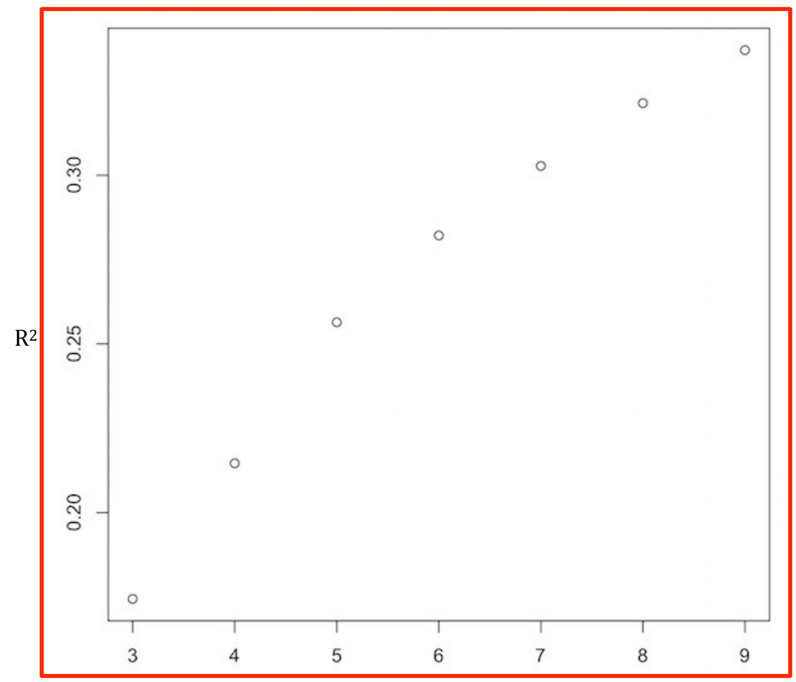
IV. TECHNICAL DESCRIPTION OF THE MODEL

The project required the use of two software programs: Microsoft Excel and R. Microsoft Excel allowed the team to see the data and determine variable types for each column. Variable type ranged from integers to characters to time and dates. In R the team was able to read the data provided in fourteen different files all at once and organize it in order to create the clusters.

After the R-code (Appendix I) read through the data, it selected only markets that had at least 100 bookings. After identifying the big markets, the program narrowed the data to only flights within the United States. Once, the data was trimmed to what was considered meaningful data for this project. The program standardized each variable in order to make them comparable between tickets. Once the variables were standardized, we created a matrix to be used for k-means clustering.

Once the data was ready for clustering, we ran k-means for different amount of clusters to find the optimal number of clusters for this particular data. The team decided to create a graph that showed the number of clusters versus R2. The purpose of this was to see when the graph started to plateau. If the increment of R^2 is insignificant from a certain amount of clusters to the next, then the results will provide very similar clusters and the classification will not be significant.

Number of clusters vs R^2 graph



In the graph it can be seen that the R^2 increment from five to six clusters is approximately .05. The increment from six to seven is .025. And the increment from seven to eight is less than .01. Therefore, Team Sabre decided that six clusters would be the optimal amount because it is when the graph starts to plateau.

Once the optimal number of clusters was decided, the team created tables for pre-booking and post-booking data. The rows of the tables showed each cluster while the columns described each variable. Some variables displayed if they were above or below the mean (advanced purchased, length of stay, fare, outbound and inbound travel time). Others showed percentages of how many tickets in that cluster belonged to that particular variable (departure day of week, return day of week, agency type, number of people). There was also a dummy variable that showed the percentage of tickets that stayed overnight on Saturday. See Appendix I and Appendix II to see the tables for post-booking and pre-booking clusters.

V. ANALYSIS AND MANAGERIAL INTERPRETATION

After analyzing the R code output and determining the optimal number of clusters, team Sabre was able to interpret and segment clusters into traveler categories. These traveler categories are identified in tables labeled below that include our reasoning.

Post-booking

Set of Rules to classify clusters:

1. Determine if the value displayed in the output table (Appendix II) is above or below the mean for the following variables: Advanced Purchase, Length of Stay, Fare, Inbound Travel Time and Outbound Travel Time.
2. Determine the most popular of the following variables: Departure Day of Week, Return Day of Week, Agency Type, and Number of Passengers.
3. Check percentage of tickets in the cluster that stayed Saturday night and determine if its high or low. If the percentage is high, we concluded that most customers stayed the weekend.
4. Use subjective analysis to name the cluster.

The post booking clusters include the following classifications: last-minute single business traveler, extended stay leisure traveler, cheap business traveler, planned vacation traveler, recurrent business traveler, and recurrent business traveler. The summary of the data gathered from the output can be seen in the following tables.

Clusters:

Cluster 1 – *last-minute single business traveler*
• Last-minute AP
• Short stay
• Expensive fare
• Monday/Tuesday departures
• Thursday/Friday returns
• G corp purchase avenue
• Single-traveler

Cluster 2 – *extended stay leisure traveler*
• High advanced purchase
• Long length of stay
• Any day travel
• G corp & Gonline
• 75% saturdays
• 81% single

Cluster 3 – *cheap business traveler*
• Long length of stay
• Long in & outbound travel times - layovers
• Purchased well in advance
• 35% stay Saturday
• Gcorp & Gonline
• 85% single

Cluster 4 – *planned vacation traveler*
• High advanced purchase
• Short L.O.S.
• Cheaper fare
• Gcorp & Gonline
• 25% return Sunday
• 64% single/10% family

Cluster 5 – *recurrent business traveler*
• Long L.O.S.
• Depart Monday/Tuesday
• Return Friday
• Gcorp & Gonline
• 89% single
• 30% Saturday nights

Cluster 6 – *quick-trip business travelers*
• Short L.O.S.
• Departs Monday/Tuesday
• Returns Wednesday/Thursday
• Gcorp
• 93%

Pre-booking

The pre-booking cluster classification Rules are the same rules as for Post-booking but with the removal of the Fare, Outbound Travel Time and Inbound Travel Time variables.

These clusters include leisure weekend traveler, leisure traveler, vacation traveler, week long business traveler, quick trip business traveler, and g-corp quick trip business traveler.

Clusters:

Cluster 1 – leisure weekend
• Increased Advanced Purchase and LOS
• Departs Thursday/Friday
• Return Sunday/Monday
• Gonline
• 96% Stay Saturday
• 70% Single

Cluster 2 – leisure travelers
• Increased Advanced Purchase and LOS
• Departs Any Day
• Return Any Day
• Gonline
• 85% Stay Saturday
• 80% Single

Cluster 3 – holiday/vacation travelers
• Increased Advanced Purchase and LOS
• Departs Any Day
• Return Sunday
• Gonline & Gcorp
• 50% Stay Saturday
• 67% Single, 22% Couple, 11% Family

Cluster 4 – week-long business traveler
• Avg. Advanced Purchase and increased LOS
• Departs Sunday/Monday
• Return Thursday/Friday
• Gcorp
• 0.03% Stay Saturday
• 95% Single

Cluster 5 – quick-trip business travelers
• Avg. Advanced Purchase and decreased LOS
• Departs Monday/Tuesday/Wednesday
• Return Thursday/Friday
• FSC and Unclass
• No Stay Saturday
• 93% Single

Cluster 6 – gcorp business travelers
• Avg. Advanced Purchase and decreased LOS
• Departs Monday/Tuesday/Wednesday
• Return Thursday/Friday
• 100% Gcorp
• No Stay Saturday
• 99% Single

VI. CONCLUSION AND CRITIQUE

We propose to Sabre to use post-booking clusters to offer discounts or premiums in post-booking services such as hotels and other reservations. For example, in Cluster 5: Recurrent Business Travelers could receive a discount in airport lounges or partnered hotels.

We propose to Sabre to use pre-booking clusters to choose a more appropriate fare targeted to each cluster. For example, in Cluster 3: Vacation travelers would prefer to pay a cheaper fare because they book in advance.

We would recommend that Sabre create a segment specific set of rules to classify each cluster appropriately. Our analysis was based on our personal interpretation of the data by comparison with other variables and clusters whereas Sabre has additional access to post-booking data. From their extended resources, Sabre can take a more quantitative approach to calculate appropriate ranges for the variables to then match each individual cluster.

From this study, team Sabre learned how to utilize R code to import and cluster data. We then analyzed the output data produced from clustering to then segment different airline customer markets. To be able to segment the data we first had to conduct analysis on the given output while understanding the measurements of the output. Although the R code was challenging to work with at the beginning, our team became “R experts” with the help of our client, Ross Darrow. Darrow guided us with sample R code and we would not have been able to complete this project without his guidance. Darrow is meeting with Sabre employees this week to discuss possible implementation strategies for segmentation purposes based on our efforts.

APPENDIX I

R-code

```
# Ross Darrow, Sabre Research, Feb 12, 2014
# For Spring 2014 SMU Mgmt Sci Capstone project
# Segmentation
# Using TDW data pulled by Jeff Johnson
# Sample rows of data

Files <- list.files (path=getwd())
fullFiles <- paste (getwd(),
                    Files,
                    sep = "/")

setClass ("myDate")
setAs ("character",
       "myDate",
       function (from) as.Date (from, format="%Y-%m-%d"))

varNames  <- c ("bookDate",        # BookingDate
                "bookID",          # PassengerID
                "numPax",          # NumberofPassengers
                "org",             # Origin
                "dst",             # Destination
                "org_cnty",        # OriginCountry
                "dst_cnty",        # DestinationCountry
                "AP",              # AdvancedPurchaseDays
                "SatNight",        # Y/N SaturdayNightStay
                "LOS",             # LengthofStay
                "AL",              # Airline
                "Fare",            # TotalAirfare
                "outb_seg",        # OutboundSegments = count
                "inb_seg",         # InboundSegments = count
                "outb_trav_time",  # OutboundElapsedTime = travel time e.g. 2:53
                "inb_trav_time",   # InboundElapsedTime
                "outb_deptDate",   # OutboundDepartureDate
                "outb_deptTime",   # OutboundDepartureTime
                "inb_deptDate",    # InboundDepartureDate
                "inb_deptTime",    # InboundDepartureTime
                "agency_type")     # travel agency type

varFormats <- c ("myDate" ,             # bookDate
                 "NULL" ,               # bookID - not needed
                 "numeric",             # numPax
                 rep ("character", 4),  # org, dst, org_cnty, dst_cnty
                 "numeric",             # AP
                 "character",           # SatNight = Y/N
                 "numeric",             # LOS
                 "character",           # AL
                 "numeric",             # Fare
                 rep ("numeric", 2),    # outb_seg, inb_seg
                rep ("character" , 2), # outb_trav_time, inb_trav_time
                 "myDate",              # outb_deptDate
                 "character",           # outb_deptTime
                 "myDate",              # inb_deptDate
                 rep ("character",2))   # inb_deptTime, agency_type


sampleData <- read.csv (fullFiles [1],
                        stringsAsFactors = FALSE,
                        colClasses       = varFormats,
                        col.names        = varNames,
                        nrows            = -1)

for (i in 2:length (fullFiles))   {
   nextDay  <- read.csv (fullFiles [i],
```

```
                         stringsAsFactors = FALSE,
                         colClasses       = varFormats,
                         col.names        = varNames,
                         nrows            = -1)

   sampleData <- rbind (sampleData, nextDay)
}

sampleData$mkt <- paste (sampleData$org,
                         sampleData$dst,
                         sep = "-")

mktList <- table (sampleData$mkt)
bigMkt  <- names (which (mktList > 100))

sampleData <- subset (sampleData,
                      mkt %in% bigMkt  &
                      org_cnty == "US" &
                      dst_cnty == "US",
                      select = -c (org_cnty,   # using only US to US
                                   dst_cnty,
                                   outb_seg,   # using travel times instead of number of segments
                                   inb_seg,
                                   org,        # market is org & dst pasted
                                   dst))

decTime <- function (mm_dd_list) {
   x <- as.numeric (unlist (mm_dd_list))
   x[1] + x[2]/60
}

timeConv <- function (mmdd)   {
   sapply (strsplit (mmdd, ":"),
           decTime)
}

sampleData$outb_trav_time <- timeConv (sampleData$outb_trav_time)
sampleData$inb_trav_time  <- timeConv (sampleData$inb_trav_time)
sampleData$outb_deptTime  <- timeConv (sampleData$outb_deptTime)
sampleData$inb_deptTime   <- timeConv (sampleData$inb_deptTime)

stdVar <- function (var, df)   {
   formula <- paste (var, "mkt", sep = "~")

   avg     <- aggregate (as.formula (formula),
                         data = sampleData,
                         median,
                         na.rm = TRUE)
   names (avg) [2] <- "median"

   IQR     <- aggregate (as.formula (formula),
                         data = sampleData,
                         IQR,
                         na.rm = TRUE)
   names (IQR) [2] <- "IQR"

   IQR$IQR <- ifelse (is.na (IQR$IQR) |
                      IQR$IQR < 0.01,
                      1, IQR$IQR)

   summary <- merge (avg,
                     IQR,
                     by = "mkt")

   df <- merge (df,
                summary,
                by = "mkt")

   col <- which (names (df) == var)
```

```
   new_col <- length (names (df)) + 1

   df [, new_col] <- (df [, col] - df$median) / df$IQR

   df [, new_col] <- pmin (df [, new_col],  3)
   df [, new_col] <- pmax (df [, new_col], -3)

   names (df) [new_col] <- paste (var, "_std", sep = "")

   drops <- c ("median", "IQR")
   df <- df [, !(names (df) %in% drops)]

   return (invisible (df))
}

sampleData <- stdVar ("AP"            , sampleData)
sampleData <- stdVar ("LOS"           , sampleData)
sampleData <- stdVar ("Fare"          , sampleData)
sampleData <- stdVar ("outb_trav_time", sampleData)
sampleData <- stdVar ("inb_trav_time" , sampleData)
sampleData <- stdVar ("outb_deptTime" , sampleData)
sampleData <- stdVar ("inb_deptTime"  , sampleData)

deptDOW <- weekdays (sampleData$outb_deptDate, abbreviate = TRUE)
retDOW  <- weekdays (sampleData$inb_deptDate , abbreviate = TRUE) # return DOW

agent   <-  ifelse (sampleData$agency_type == "FSC-Corporate",        "FSC",
            ifelse (sampleData$agency_type == "Global Corporate",     "GCorp",
            ifelse (sampleData$agency_type == "Global Online",        "GOnline",
            ifelse (sampleData$agency_type == "Leisure Agency",       "LAgent",
            ifelse (sampleData$agency_type == "Leisure Consolidator", "LConsol",
            ifelse (sampleData$agency_type == "Leisure Generalist",   "LGeneral",
            ifelse (sampleData$agency_type == "Leisure Supplier",     "LSupply",
            ifelse (sampleData$agency_type == "Other",                "Other",
            ifelse (sampleData$agency_type == "Regional Online",      "RegOnline",
            ifelse (sampleData$agency_type == "Unclassified",         "Unclass",
                                                                      "Unknown"))))))))))
sampleData <- cbind (sampleData,
                     model.matrix ( ~ deptDOW - 1),
                     model.matrix ( ~ retDOW  - 1),
                     model.matrix ( ~ agent   - 1))

sampleData$SatNightdummy <- as.integer (sampleData$SatNight == "Y")

# also set up numPax as a categorical variable

numPax_cat <- cut (sampleData$numPax,
                   breaks = c (0, 1, 2, 5, 1000),
                   labels = c ("single", "couple", "family", "group"),
                   include.lowest = TRUE)

sampleData <- cbind (sampleData,
                     model.matrix ( ~ numPax_cat - 1))

sampleData <- na.omit (sampleData)

dataMatrix <- sampleData [, 16:ncol (sampleData)]   #Selects variables/columns to do Matrix

Clus9 <- kmeans (dataMatrix, 9)
Sortof_R2_9 <- Clus9$betweenss / Clus9$totss


Clus6 <- kmeans (dataMatrix, 6)
Sortof_R2_6 <- Clus6$betweenss / Clus6$totss

library ("rpart")

prebook_data <- dataMatrix [, c (1, 2, 8:ncol (dataMatrix))] #Do the Clustering for this variable as
```

well

APPENDIX II

Post-booking cluster output: The columns are labeled by the different variable names and the rows represent each cluster.

	AP_std	LOS_std	Fare_std	outb_trav_time_std	inb_trav_time_std	outb_deptTime_std	inb_deptTime_std	deptDOWFri	deptDOWMon	deptDOWSat	deptDOWSun	deptDOWThu	deptDOWTue	deptDOWWed
1	-0.31591419	-0.3981356	1.47667280	0.03108718	0.066405942	0.17397864	0.07436622	0.06107130	0.2725231	0.02448072	0.1205448	0.1257178	0.2093082	0.1863541
2	0.31820472	2.2264045	-0.09411384	0.04218029	0.047315772	0.28862257	-0.17592798	0.14488539	0.1476081	0.13764397	0.2038431	0.1245153	0.1172597	0.1242445
3	0.06583190	0.2480178	0.11644778	2.06094014	2.220614234	0.02842091	-0.26963096	0.11704364	0.2386814	0.07222243	0.1336581	0.1272766	0.1685861	0.1425317
4	2.08050997	0.5062332	-0.31467783	-0.07452635	0.007743745	0.14925422	-0.12325974	0.18857937	0.1376824	0.11312125	0.1123404	0.1935382	0.1131821	0.1415562
5	0.07022718	0.1925625	-0.03752386	-0.02682594	-0.041353387	0.20669228	-0.96233342	0.11228334	0.2433333	0.05448641	0.1278870	0.1425700	0.1719640	0.1474760
6	0.08984135	-0.2139060	-0.05430233	-0.03196644	-0.073036888	0.06956802	0.33101953	0.07140106	0.3196165	0.03456259	0.1191236	0.0988373	0.2105083	0.1459506

	retDOWFri	retDOWMon	retDOWSat	retDOWSun	retDOWThu	retDOWTue	retDOWWed	agentFSC	agentGCorp	agentGOnline	agentLAgent	agentLConsol	agentLGeneral	agentLSupply	agentOther	agentRegOnline
1	0.2151298	0.07326816	0.03670131	0.05823960	0.2727920	0.1366491	0.2072200	0.20621549	0.5401737	0.06825337	0.006090519	0.01831902	0.03825163	0.001985351	0.0038678753	0.003227184
2	0.2088037	0.12765139	0.14631087	0.17490592	0.1198968	0.1168748	0.1055565	0.09978333	0.3020584	0.39328886	0.015067282	0.01820333	0.03652070	0.005530847	0.0014397309	0.029393317
3	0.1973090	0.11428715	0.08614331	0.13341892	0.1910282	0.1166283	0.1611852	0.11963650	0.3470025	0.35916122	0.007338041	0.01163010	0.02488389	0.003373233	0.0009817619	0.027703307
4	0.1397917	0.16059060	0.10634716	0.25199014	0.1286469	0.1036801	0.1089534	0.10366998	0.2093884	0.45304276	0.012534099	0.02135664	0.03838314	0.024226506	0.0009938039	0.016326779
5	0.3207332	0.10402463	0.08106892	0.13552722	0.1702349	0.0743987	0.1140124	0.14867087	0.4451200	0.22608488	0.006403138	0.01473917	0.03150962	0.003275370	0.0018134023	0.011948775
6	0.1820069	0.06846483	0.04407880	0.06690434	0.3117237	0.1224686	0.2043529	0.15270353	0.5489647	0.13472454	0.005679853	0.01613550	0.02918540	0.003120971	0.0016508915	0.007833873

	agentUnclass	SatNightdummy	numPax_catsingle	numPax_catcouple	numPax_catfamily	numPax_catgroup
1	0.11361587	0.1055321	0.9534669	0.03526173	0.01070982	0.0005615933
2	0.09871422	0.7721519	0.8115093	0.13427985	0.05081822	0.0033926331
3	0.09828947	0.3560020	0.8557062	0.10528767	0.03681607	0.0021900842
4	0.12007788	0.5708694	0.6406486	0.24483070	0.10638773	0.0081329669
5	0.11043479	0.3082011	0.8927492	0.07934689	0.02625919	0.0016447137
6	0.10000079	0.1400310	0.9348370	0.04707006	0.01642637	0.0016666143

APPENDIX III

Pre-booking cluster output: The columns are labeled by the different variable names and the rows represent each cluster.

	AP_std	LOS_std	deptDOWFri	deptDOWMon	deptDOWSat	deptDOWSun	deptDOWThu	deptDOWTue	deptDOWWed	retDOWFri	retDOWMon	retDOWSat	retDOWSun	retDOWThu	retDOWTue	retDOWWed
1	0.14196885	0.1973270	0.34522091	0.01488015	0.162779332	0.01099281	0.278707630	0.05086619	0.13655298	0.01327859	0.28477548	0.02937682	0.441411772	0.02967758	0.13060543	0.07087431
2	0.46310243	2.4621882	0.15126192	0.14288994	0.145384698	0.16020019	0.131931371	0.13438021	0.13395167	0.17001087	0.13379861	0.15132314	0.176944151	0.12189112	0.12657453	0.11945758
3	2.23761512	0.4411452	0.15445497	0.16072022	0.110701318	0.12503007	0.169700024	0.13206272	0.14733068	0.15541709	0.13652972	0.12306001	0.208929410	0.15120208	0.10146953	0.12339217
4	0.07379563	0.7082457	0.01190875	0.50501489	0.024726948	0.34012853	0.009617942	0.08490558	0.02369737	0.53557609	0.01107651	0.08705911	0.017708682	0.27809666	0.01327293	0.05721002
5	-0.07069748	-0.4278471	0.02925808	0.27241207	0.006000562	0.11868659	0.109383584	0.26343259	0.20082653	0.21144807	0.04328584	0.08198733	0.002551612	0.28846036	0.13576652	0.23650026
6	-0.07742389	-0.4365890	0.02171361	0.31390733	0.007359638	0.08673268	0.096011378	0.27830246	0.19597290	0.19355216	0.04107954	0.03344715	0.001923929	0.33854824	0.13842550	0.25302349

	agentFSC	agentGCorp	agentGOnline	agentLAgent	agentLConsol	agentLGeneral	agentLSupply	agentOther	agentRegOnline	agentUnclass	SatNightdummy	numPax_catsingle	numPax_catcouple
1	0.07540076	0.1662155	0.54391110	0.011286054	0.02061716	0.03124154	0.007225781	0.001218082	0.025730097	0.11715390	0.963450029	0.7053445	0.218081747
2	0.09625480	0.2628832	0.41950197	0.017065369	0.01864181	0.03896720	0.007958737	0.001438695	0.036579580	0.10070863	0.849151323	0.7982460	0.143165435
3	0.11460708	0.2460742	0.39950978	0.012278511	0.02267859	0.04058094	0.024797553	0.001248468	0.016814230	0.12141065	0.477773833	0.6722485	0.216752380
4	0.12146405	0.6401723	0.09067978	0.005233671	0.01186585	0.02858785	0.004169777	0.001381346	0.007756128	0.08868927	0.034636603	0.9558227	0.029119800
5	0.44146553	0.0000000	0.18724438	0.010401787	0.03358605	0.07015102	0.002588238	0.005603780	0.010926760	0.23803245	0.003296341	0.9366553	0.047540563
6	0.00000000	1.0000000	0.00000000	0.000000000	0.00000000	0.00000000	0.000000000	0.000000000	0.000000000	0.00000000	0.008065891	0.9935122	0.005776659

	numPax_catfamily	numPax_catgroup
1	0.0721751030	4.398629e-03
2	0.0547775380	3.811011e-03
3	0.1027867181	8.212400e-03
4	0.0141051710	9.523564e-04
5	0.0141315362	1.672588e-03
6	0.0006770283	3.409495e-05