In [2]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import datetime
import warnings
warnings.filterwarnings('ignore')
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
```

In [3]:
```python
sun_df = pd.read_csv('SunCountry.csv')
```

Filter 1: Filter for sun country airlines data

In [4]:
```python
sun_air = sun_df[sun_df['MarketingAirlineCode']=='SY']
```

In [5]:
```python
# For temporary calculations
sun_lite = sun_air.head(100)
sun_lite.head()
```

Out[5]:

	PNRLocatorID	TicketNum	CouponSeqNbr	ServiceStartCity	ServiceEndCity	PNRCreateDate	ServiceStartDate	PaxName	EncryptedName	Ger
0	AAABJK	3377365159634	2	JFK	MSP	2013-11-23	2013-12-13	BRUMSA	4252554D4241434B44696420493F7C2067657420746869...	
1	AAABJK	3377365159634	1	MSP	JFK	2013-11-23	2013-12-08	BRUMSA	4252554D4241434B44696420493F7C2067657420746869...	
2	AAABMK	3372107381942	2	MSP	SFO	2014-02-04	2014-02-23	EILDRY	45494C4445525344696420493F7C206765742074686973...	
3	AAABMK	3372107381942	1	SFO	MSP	2014-02-04	2014-02-20	EILDRY	45494C4445525344696420493F7C206765742074686973...	
4	AAABTP	3372107470782	1	MCO	MSP	2014-03-13	2014-04-23	SKELMA	534B454C544F4E44696420493F7C206765742074686973...	

Filter 2: Drop records with null birthdateid

Only ~1% null records were there and no way to standardize them

In [5]:
```python
sun_air = sun_air.dropna(subset=['birthdateid'], axis=0)
```

In [6]:
```python
sun_air['birthdateid'].isna().sum()
```

Out[6]:
```
0
```

Defining Primary Key: Combination of encrypted name, birthdateid and gendercode

In [9]:
```python
sun_air['Cus_ID'] = sun_air['EncryptedName']+sun_air['birthdateid'].astype(str)+sun_air['GenderCode'].astype(str)
```

Data Transformation

Add columns

In [10]:
```python
# Smoothening of Age
sun_air['Age'] = sun_air['Age'].where((sun_air['Age'] >= 0) & (sun_air['Age'] <= 100),np.nan)
sun_air['Age'] = sun_air['Age'].fillna(sun_air['Age'].mean()).astype('int64')

# Age groups
sun_air.loc[sun_air['Age'] <= 17,'Age_group'] = 'Children'
sun_air.loc[(sun_air['Age'] >= 18) & (sun_air['Age'] <= 25),'Age_group'] = 'Youth'
sun_air.loc[(sun_air['Age'] >= 26) & (sun_air['Age'] <= 40), 'Age_group'] = 'Young Adults'
sun_air.loc[(sun_air['Age'] >= 41) & (sun_air['Age'] <= 54), 'Age_group'] = 'Middle Aged'
sun_air.loc[(sun_air['Age'] >= 55) & (sun_air['Age'] <= 100),'Age_group'] = 'Senior'
sun_air['Age_group'].fillna('Other', inplace=True)
```

In [11]:
```python
# missing value %
sun_air.isna().sum()* 100 / len(sun_air)
```

Out[11]:
```
PNRLocatorID             0.000000
TicketNum                0.000000
CouponSeqNbr             0.000000
ServiceStartCity         0.000000
ServiceEndCity           0.000000
PNRCreateDate            0.000000
ServiceStartDate         0.000000
PaxName                  0.000000
EncryptedName            0.000000
GenderCode               0.000000
birthdateid              0.000000
Age                      0.000000
PostalCode              79.696253
BkdClassOfService        0.000000
TrvldClassOfService      0.000000
BookingChannel           0.000000
BaseFareAmt              0.000000
TotalDocAmt              0.000000
UFlyRewardsNumber       79.576243
UflyMemberStatus        79.576243
CardHolder              79.576243
BookedProduct           64.920483
EnrollDate              79.576243
MarketingFlightNbr       0.000000
MarketingAirlineCode     0.000000
StopoverCode            50.109896
Cus_ID                   0.000000
Age_group                0.000000
dtype: float64
```

Updating datatypes

In [12]:
```python
sun_air.dtypes
```

Out[12]:
```
PNRLocatorID             object
TicketNum                 int64
CouponSeqNbr              int64
ServiceStartCity         object
ServiceEndCity           object
PNRCreateDate            object
ServiceStartDate         object
PaxName                  object
EncryptedName            object
GenderCode               object
birthdateid             float64
Age                       int64
PostalCode               object
BkdClassOfService        object
TrvldClassOfService      object
BookingChannel           object
BaseFareAmt             float64
TotalDocAmt             float64
UFlyRewardsNumber       float64
UflyMemberStatus         object
CardHolder               object
BookedProduct            object
EnrollDate               object
MarketingFlightNbr       object
MarketingAirlineCode     object
StopoverCode             object
Cus_ID                   object
Age_group                object
dtype: object
```

MarketingFlightNbr - object to int

In [13]:
```python
sun_air['MarketingFlightNbr'] = sun_air['MarketingFlightNbr'].where(sun_air['MarketingFlightNbr'] != 'OPEN', 0)
```

In [14]:
```python
sun_air['MarketingFlightNbr'] = sun_air['MarketingFlightNbr'].astype('int64')
```

In [15]:
```python
sun_air['MarketingFlightNbr'].head()
```

Out[15]:
```
0    244
1    243
2    397
3    392
4    342
Name: MarketingFlightNbr, dtype: int64
```

PNRCreateDate - object to date

In [16]:
```python
sun_air['PNRCreateDate'] = pd.to_datetime(sun_air['PNRCreateDate'])
```

In [17]:
```python
sun_air['PNRCreateDate'].head()
```

Out[17]:
```
0   2013-11-23
1   2013-11-23
2   2014-02-04
3   2014-02-04
4   2014-03-13
Name: PNRCreateDate, dtype: datetime64[ns]
```

ServiceStartDate - object to date

In [18]:
```python
sun_air['ServiceStartDate'] = pd.to_datetime(sun_air['PNRCreateDate'])
```

In [19]:
```python
sun_air['ServiceStartDate'].head()
```

Out[19]:
```
0   2013-11-23
1   2013-11-23
2   2014-02-04
3   2014-02-04
4   2014-03-13
Name: ServiceStartDate, dtype: datetime64[ns]
```

EnrollDate - object to date

In [20]:
```python
sun_air['EnrollDate'] = pd.to_datetime(sun_air['PNRCreateDate'])
```

In [21]:
```python
sun_air['EnrollDate'].head()
```

Out[21]:
```
0   2013-11-23
1   2013-11-23
2   2014-02-04
3   2014-02-04
4   2014-03-13
Name: EnrollDate, dtype: datetime64[ns]
```

Drop duplicates

In [22]:
```python
sun_air.drop_duplicates(inplace=True)
```

In [6]:
```python
sun_air.head()
```

Out[6]:

	PNRLocatorID	TicketNum	CouponSeqNbr	ServiceStartCity	ServiceEndCity	PNRCreateDate	ServiceStartDate	PaxName	EncryptedName	Ger
0	AAABJK	3377365159634	2	JFK	MSP	2013-11-23	2013-12-13	BRUMSA	4252554D4241434B44696420493F7C2067657420746869...	
1	AAABJK	3377365159634	1	MSP	JFK	2013-11-23	2013-12-08	BRUMSA	4252554D4241434B44696420493F7C2067657420746869...	
2	AAABMK	3372107381942	2	MSP	SFO	2014-02-04	2014-02-23	EILDRY	45494C4445525344696420493F7C206765742074686973...	
3	AAABMK	3372107381942	1	SFO	MSP	2014-02-04	2014-02-20	EILDRY	45494C4445525344696420493F7C206765742074686973...	
4	AAABTP	3372107470782	1	MCO	MSP	2014-03-13	2014-04-23	SKELMA	534B454C544F4E44696420493F7C206765742074686973...	

In [24]:
```python
sun_air.describe()
```

Out[24]:

	TicketNum	CouponSeqNbr	birthdateid	Age	BaseFareAmt	TotalDocAmt	UFlyRewardsNumber	MarketingFlightNbr
count	3.258027e+06	3.258027e+06	3.258027e+06	3.258027e+06	3.258027e+06	3.258027e+06	6.761000e+05	3.258027e+06
mean	3.374388e+12	1.460199e+00	4.491284e+04	4.020675e+01	2.845755e+02	3.115526e+02	2.042188e+08	3.634179e+02
std	2.587984e+09	5.731262e-01	7.040379e+03	1.886540e+01	1.800219e+02	2.112936e+02	1.485196e+07	1.939676e+02
min	3.372052e+12	1.000000e+00	-6.752900e+05	0.000000e+00	0.000000e+00	0.000000e+00	1.000002e+08	0.000000e+00
25%	3.372107e+12	1.000000e+00	3.957900e+04	2.600000e+01	1.711600e+02	1.879000e+02	2.008613e+08	2.510000e+02
50%	3.372108e+12	1.000000e+00	4.499900e+04	4.000000e+01	2.697600e+02	2.980000e+02	2.029677e+08	3.460000e+02
75%	3.377303e+12	2.000000e+00	5.013200e+04	5.500000e+01	3.665200e+02	4.098000e+02	2.103816e+08	4.300000e+02
max	3.379578e+12	8.000000e+00	1.112840e+06	1.000000e+02	4.342000e+03	1.757200e+04	2.410863e+08	8.877000e+03

Feature Creation

UFly Membership Status, Age, Gender

In [28]:
```python
#1
part1 = sun_air[['Cus_ID', 'GenderCode', 'Age_group', 'UflyMemberStatus']]
```

In [29]:
```python
# Most of the customers are not members
part1.groupby('UflyMemberStatus')['Cus_ID'].nunique()
```

Out[29]:
```
UflyMemberStatus
Elite         1293
Standard    266788
Name: Cus_ID, dtype: int64
```

In [30]:
```python
#2
part1['UflyMemberStatus'] = part1['UflyMemberStatus'].fillna('Not_member')
```

In [35]:
```python
#3
# Conversion of membership status column to categories: 1-not member, 2-standard, 3-elite
#part1['UflyMemberStatus'] = part1['UflyMemberStatus'].where(part1['UflyMemberStatus'] != 'Not_member', 1)
#part1['UflyMemberStatus'] = part1['UflyMemberStatus'].where((part1['UflyMemberStatus'] == 'Elite') | (part1['UflyMemberStatus'] == 1), 2)
#part1['UflyMemberStatus'] = part1['UflyMemberStatus'].where((part1['UflyMemberStatus'] == 2) | (part1['UflyMemberStatus'] == 1), 3)
#part1['UflyMemberStatus'] = part1['UflyMemberStatus'].astype(int)
```

In [36]:
```python
part1['Cus_ID'].nunique()
```

Out[36]:
```
1528104
```

In [37]:
```python
#4
part1 = part1.groupby(['Cus_ID', 'GenderCode', 'Age_group'], as_index=False)['UflyMemberStatus'].max()
```

In [39]:
```python
# 83% customers are not members, 17% are standard, ~0% are not members
part1.groupby('UflyMemberStatus')['Cus_ID'].count()/15281.04
```

Out[39]:
```
UflyMemberStatus
1    82.917066
2    17.619547
3     0.087232
Name: Cus_ID, dtype: float64
```

In [45]:
```python
#5
#part1['UflyMemberStatus'] = part1['UflyMemberStatus'].where((part1['UflyMemberStatus']==1) | (part1['UflyMemberStatus']==2),'Elite')
#part1['UflyMemberStatus'] = part1['UflyMemberStatus'].where((part1['UflyMemberStatus']==1) | (part1['UflyMemberStatus']=='Elite'),'Standard')
#part1['UflyMemberStatus'] = part1['UflyMemberStatus'].where((part1['UflyMemberStatus']=='Standard') | (part1['UflyMemberStatus']=='Elite'),'Not mem
```

In [44]:
```python
part1.head()
```

Out[44]:

	Cus_ID	GenderCode	Age_group	UflyMemberStatus
0	4120414C52484D414E44696420493F7C20676574207468...	M	Young Adults	Not member
1	414142454C44696420493F7C2067657420746869732072...	M	Youth	Not member
2	4141424552472042524F4F4B5344696420493F7C206765...	F	Middle Aged	Not member
3	41414245524744696420493F7C20676574207468697320...	M	Middle Aged	Not member
4	41414245524744696420493F7C20676574207468697320...	M	Young Adults	Standard

Card holders & Number of trips

In [46]:
```python
sun_air.groupby('CardHolder')['Cus_ID'].count()
```

Out[46]:
```
CardHolder
False    641473
True      34627
Name: Cus_ID, dtype: int64
```

In [47]:
```python
#1
sun = pd.DataFrame(sun_air.groupby('Cus_ID')['CardHolder'].sum()>0).astype(int).reset_index()
```

In [48]:
```python
#2
sun2 = pd.DataFrame(sun_air.groupby('Cus_ID')['TicketNum'].count()).reset_index()
```

In [49]:
```python
#3
sun = sun.merge(sun2, on="Cus_ID")
```

In [50]:
```python
#4
sun.columns=['Cus_ID','CardHolder','NumTrips']
```

In [51]:
```python
part2 = sun
```

In [54]:
```python
#5
#part2['CardHolder'] = part2['CardHolder'].where(part2['CardHolder']==1,'No')
#part2['CardHolder'] = part2['CardHolder'].where(part2['CardHolder']=='No','Yes')
```

In [55]:
```python
part2.head()
```

Out[55]:

	Cus_ID	CardHolder	NumTrips
0	4120414C52484D414E44696420493F7C20676574207468...	No	1
1	414142454C44696420493F7C2067657420746869732072...	No	1
2	4141424552472042524F4F4B5344696420493F7C206765...	No	1
3	41414245524744696420493F7C20676574207468697320...	No	2
4	41414245524744696420493F7C20676574207468697320...	No	2

Total amount spent & number of discounts

In [56]:
```python
#1
feature1 = sun_air.groupby('Cus_ID', as_index=False)['TotalDocAmt'].sum()
```

In [57]:
```python
#2
feature2 = sun_air.groupby('Cus_ID', as_index=False)['BookedProduct'].count()
```

In [58]:
```python
#3
part3 = feature1.merge(feature2, on="Cus_ID")
```

In [59]:
```python
part3.head()
```

Out[59]:

	Cus_ID	TotalDocAmt	BookedProduct
0	4120414C52484D414E44696420493F7C20676574207468...	174.0	0
1	414142454C44696420493F7C2067657420746869732072...	231.9	0
2	4141424552472042524F4F4B5344696420493F7C206765...	294.9	0
3	41414245524744696420493F7C20676574207468697320...	0.0	2
4	41414245524744696420493F7C20676574207468697320...	973.6	0

Number of trips by class

In [60]:
```python
#1
part4 = sun_air[['Cus_ID', 'BkdClassOfService', 'ServiceStartCity']]
```

In [61]:
```python
part4.groupby('BkdClassOfService')['Cus_ID'].count()
```

Out[61]:
```
BkdClassOfService
Coach                   3168492
Discount First Class        759
First Class               88776
Name: Cus_ID, dtype: int64
```

In [62]:
```python
#2
part4['BkdClassOfService'] = part4['BkdClassOfService'].where(part4['BkdClassOfService']=='Coach', 'First Class')
```

In [63]:
```python
#3
part4 = part4.groupby(['Cus_ID', 'BkdClassOfService'], as_index=False)['ServiceStartCity'].count()
```

In [66]:
```python
#4
part4 = pd.pivot_table(part4, index=['Cus_ID'],
              columns=['BkdClassOfService'], aggfunc=np.sum).reset_index()
```

In [67]:
```python
part4.columns = ['Cus_ID', 'Coach', 'First Class']
```

In [68]:
```python
#5
part4['Coach'] = part4['Coach'].fillna(0)
part4['First Class'] = part4['First Class'].fillna(0)
```

Out[68]:
```
0
```

In [69]:
```python
#5
part4['total trips'] = part4['Coach'] + part4['First Class']
part4['Coach'] = part4['Coach']/part4['total trips']*100
part4['First Class'] = part4['First Class']/part4['total trips']*100
part4['Coach'] = part4['Coach'].where(part4['Coach'] == 100, 'First Class')
part4['Coach'] = part4['Coach'].where(part4['Coach'] == 'First Class', 'Coach')
part4['Preferred class of travel'] = part4['Coach']
part4 = part4[['Cus_ID', 'Preferred class of travel']]
```

In [70]:
```python
part4.groupby('Preferred class of travel')['Cus_ID'].count()
```

Out[70]:
```
Preferred class of travel
Coach          1482534
First Class      45570
Name: Cus_ID, dtype: int64
```

Upgrade & downgrade

In [71]:
```python
def upgrade(row):
    if (row['BkdClassOfService'] == 'Coach' and row['TrvldClassOfService'] == 'First Class') or \
    (row['BkdClassOfService'] == 'Coach' and row['TrvldClassOfService'] == 'Discount First Class'):
        return  'Upgrade'
    elif (row['BkdClassOfService'] == 'First Class' and row['TrvldClassOfService'] == 'Coach') or \
    (row['BkdClassOfService'] == 'Discount First Class' and row['TrvldClassOfService'] == 'Coach'):
        return 'Downgrade'
    else:
        return 'No upgrade'
```

In [72]:
```python
sun_air['upgrade'] = sun_air.apply(lambda row: upgrade(row), axis='columns')
```

In [73]:
```python
part5 = sun_air[['Cus_ID', 'upgrade', 'ServiceStartCity']]
```

In [74]:
```python
part5 = part5.groupby(['Cus_ID', 'upgrade'], as_index=False)['ServiceStartCity'].count()
```

In [76]:
```python
part5 = pd.pivot_table(part5, index=['Cus_ID'],
              columns=['upgrade'], aggfunc=np.sum).reset_index()
```

In [77]:
```python
part5.columns = ['Cus_ID', 'Downgrade', 'No upgrade', 'Upgrade']
```

In [78]:
```python
part5['Downgrade'] = part5['Downgrade'].fillna(0)
```

In [79]:
```python
part5['No upgrade'] = part5['No upgrade'].fillna(0)
```

In [80]:
```python
part5['Upgrade'] = part5['Upgrade'].fillna(0)
```

In [82]:
```python
part5 = part5[['Cus_ID', 'Downgrade', 'Upgrade']]
```

In [83]:
```python
part5['Upgrades'] = part5['Upgrade'] - part5['Downgrade']
```

In [84]:
```python
part5 = part5[['Cus_ID', 'Upgrades']]
```

In [88]:
```python
part5.head()
```

Out[88]:

	Cus_ID	Upgrades
0	4120414C52484D414E44696420493F7C20676574207468...	0.0
1	414142454C44696420493F7C2067657420746869732072...	1.0
2	4141424552472042524F4F4B5344696420493F7C206765...	0.0
3	41414245524744696420493F7C20676574207468697320...	0.0
4	41414245524744696420493F7C20676574207468697320...	2.0

BookingChannel

In [89]:
```python
part6 = sun_air[['Cus_ID', 'BookingChannel', 'ServiceStartCity']]
```

In [90]:
```python
part6['BookingChannel'] = part6['BookingChannel'].where((part6['BookingChannel']== 'Outside Booking') | \
                                                        (part6['BookingChannel']== 'SCA Website Booking') | \
                                                        (part6['BookingChannel']== 'Reservations Booking') | \
                                                        (part6['BookingChannel']== 'SY Vacation') | \
                                                        (part6['BookingChannel']== 'Tour Operator Portal'), 'Airport')
```

In [92]:
```python
part6.head()
```

Out[92]:

	Cus_ID	BookingChannel	ServiceStartCity
0	4252554D4241434B44696420493F7C2067657420746869...	Outside Booking	JFK
1	4252554D4241434B44696420493F7C2067657420746869...	Outside Booking	MSP
2	45494C4445525344696420493F7C206765742074686973...	SCA Website Booking	MSP
3	45494C4445525344696420493F7C206765742074686973...	SCA Website Booking	SFO
4	534B454C544F4E44696420493F7C206765742074686973...	Reservations Booking	MCO

In [93]:
```python
part6.groupby('BookingChannel')['Cus_ID'].count()
```

Out[93]:
```
BookingChannel
Airport                   11373
Outside Booking         1444753
Reservations Booking     161321
SCA Website Booking     1426937
SY Vacation               87278
Tour Operator Portal     126365
Name: Cus_ID, dtype: int64
```

In [94]:
```python
part6 = part6.groupby(['Cus_ID', 'BookingChannel'], as_index=False)['ServiceStartCity'].count()
```

In [95]:
```python
part6['Cus_ID'].nunique()
```

Out[95]:
```
1528104
```

In [96]:
```python
part6 = pd.pivot_table(part6, index=['Cus_ID'],
              columns=['BookingChannel'], aggfunc=np.sum).reset_index()
```

In [97]:
```python
part6.columns = ['Cus_ID', 'Airport', 'Outside Booking', 'Reservations Booking', 'SCA Website Booking', 'SY Vacation', 'Tour Operator Portal']
```

In [98]:
```python
part6['Airport'] = part6['Airport'].fillna(0)
part6['Outside Booking'] = part6['Outside Booking'].fillna(0)
part6['Reservations Booking'] = part6['Reservations Booking'].fillna(0)
part6['SCA Website Booking'] = part6['SCA Website Booking'].fillna(0)
part6['SY Vacation'] = part6['SY Vacation'].fillna(0)
part6['Tour Operator Portal'] = part6['Tour Operator Portal'].fillna(0)
```

In [124…
```python
part6['Cus_ID'].nunique()
```

Out[124]:
```
1528104
```

In [99]:
```python
part6.head()
```

Out[99]:

	Cus_ID	Airport	Outside Booking	Reservations Booking	SCA Website Booking	SY Vacation	Tour Operator Portal
0	4120414C52484D414E44696420493F7C20676574207468...	0.0	0.0	0.0	1.0	0.0	0.0
1	414142454C44696420493F7C2067657420746869732072...	0.0	1.0	0.0	0.0	0.0	0.0
2	4141424552472042524F4F4B5344696420493F7C206765...	1.0	0.0	0.0	0.0	0.0	0.0
3	41414245524744696420493F7C20676574207468697320...	0.0	0.0	0.0	0.0	0.0	2.0
4	41414245524744696420493F7C20676574207468697320...	0.0	2.0	0.0	0.0	0.0	0.0

In [101…
```python
def transform(a,e,i,o,u,g):
    if a > e and a > i and a > o and a > u and a > g:
        return 'Airport'
    elif e > a and e > i and e > o and e > u and e > g:
        return 'Outside Booking'
    elif i > a and i > e and i > o and i > u and i > g:
        return 'Reservations Booking'
    elif o > a and o > e and o > i and o > u and o > g:
        return 'SCA Website Booking'
    elif u > a and u > e and u > i and u > o and u > g:
        return 'SY Vacation'
    elif g > a and g > e and g > i and g > o and g > u:
        return 'Tour Operator Portal'
    else:
        return 'No Preference'
```

In [102…
```python
part6['Preferred source'] = part6.apply(lambda x: transform(x['Airport'], x['Outside Booking'], \
                                                            x['Reservations Booking'], x['SCA Website Booking'], \
                                                            x['SY Vacation'], x['Tour Operator Portal']), axis=1)
```

In [103…
```python
part6.head()
```

Out[103]:

	Cus_ID	Airport	Outside Booking	Reservations Booking	SCA Website Booking	SY Vacation	Tour Operator Portal	Preferred source
0	4120414C52484D414E44696420493F7C20676574207468...	0.0	0.0	0.0	1.0	0.0	0.0	SCA Website Booking
1	414142454C44696420493F7C2067657420746869732072...	0.0	1.0	0.0	0.0	0.0	0.0	Outside Booking
2	4141424552472042524F4F4B5344696420493F7C206765...	1.0	0.0	0.0	0.0	0.0	0.0	Airport
3	41414245524744696420493F7C20676574207468697320...	0.0	0.0	0.0	0.0	0.0	2.0	Tour Operator Portal
4	41414245524744696420493F7C20676574207468697320...	0.0	2.0	0.0	0.0	0.0	0.0	Outside Booking

In [104…
```python
part6 = part6[['Cus_ID', 'Preferred source']]
```

In [105…
```python
part6.groupby('Preferred source')['Cus_ID'].nunique()
```

Out[105]:
```
Preferred source
Airport                   5117
No Preference            30483
Outside Booking         686252
Reservations Booking     70865
SCA Website Booking     637999
SY Vacation              37006
Tour Operator Portal     60382
Name: Cus_ID, dtype: int64
```

In [106…
```python
part6.head()
```

Out[106]:

	Cus_ID	Preferred source
0	4120414C52484D414E44696420493F7C20676574207468...	SCA Website Booking
1	414142454C44696420493F7C2067657420746869732072...	Outside Booking
2	4141424552472042524F4F4B5344696420493F7C206765...	Airport
3	41414245524744696420493F7C20676574207468697320...	Tour Operator Portal
4	41414245524744696420493F7C20676574207468697320...	Outside Booking

Merge

In [107…
```python
p = part1.merge(part2, on="Cus_ID")
```

In [108…
```python
q = p.merge(part3, on="Cus_ID")
```

In [109…
```python
r = q.merge(part4, on="Cus_ID")
```

In [110…
```python
s = r.merge(part5, on="Cus_ID")
```

In [111…
```python
t = s.merge(part6, on="Cus_ID")
```

In [113…
```python
t.columns = ['CustomerID', 'Gender', 'Age_group', 'Ufly_membership_status', 'Card_holder?', 'Total_trips', 'Total_amount_spent', '#Discounts', 'Pref
```

In [114…
```python
t.drop_duplicates(inplace=True)
```

In [115…
```python
t.head()
```

Out[115]:

	CustomerID	Gender	Age_group	Ufly_membership_status	Card_holder?	Total_trips	Total_amount_spent	#Discounts	Preferred_travel_class
0	4120414C52484D414E44696420493F7C20676574207468...	M	Young Adults	Not member	No	1	174.0	0	Coach
1	414142454C44696420493F7C2067657420746869732072...	M	Youth	Not member	No	1	231.9	0	Coach
2	4141424552472042524F4F4B5344696420493F7C206765...	F	Middle Aged	Not member	No	1	294.9	0	Coach
3	41414245524744696420493F7C20676574207468697320...	M	Middle Aged	Not member	No	2	0.0	2	Coach
4	41414245524744696420493F7C20676574207468697320...	M	Young Adults	Standard	No	2	973.6	0	Coach

In [117…
```python
t.to_csv('sun.csv')
```

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import datetime
import warnings
warnings.filterwarnings('ignore')
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
```

In [2]:
```
sun_df = pd.read_csv('sun.csv')
```

In [34]:
```
sun_df.head()
```

Out[34]:

	Unnamed: 0	CustomerID	Gender	Age_group	Uffy_membership_status	Card_holder?	Total_trips	Total_amount_spent	#Discounts	Preferred_tr
0	0	4120414C52484D414E4469642493F7C20676574207468...	M	Young Adults	Not member	No	1	174.0	0	
1	1	414142454C4469642493F7C20676574207468973207 2...	M	Youth	Not member	No	1	231.9	0	
2	2	414142455247204254F4F4B534469642493F7C206765...	F	Middle Aged	Not member	No	1	294.9	0	
3	3	414142455247446964204937C20676574207468973207...	M	Middle Aged	Not member	No	2	0.0	2	
4	4	414142455247446964204937C20676574207468973207...	M	Young Adults	Standard	No	2	973.6	0	

Random Sampling

We have used block randomization technique to take sample of data. This helped us wholistically understand every type of Sun Country Airlines customer.

In [43]:
```
new1 = sun_df[sun_df['Uffy_membership_status']=='Elite'].sample(n=1300,random_state=2)
new2 = sun_df[sun_df['Uffy_membership_status']=='Standard'].sample(n=1000,random_state=2)
new3 = sun_df[sun_df['Card_holder?']=='Yes'].sample(n=2400,random_state=2)
new4 = sun_df[sun_df['Card_holder?']=='No'].sample(n=1200,random_state=2)
new5 = sun_df[sun_df['Preferred_travel_class']=='Coach'].sample(n=800,random_state=2)
new6 = sun_df[sun_df['Preferred_travel_class']=='First Class'].sample(n=1400,random_state=2)
new7 = sun_df[sun_df['Age_group']=='Young Adults'].sample(n=800,random_state=2)
new8 = sun_df[sun_df['Age_group']=='Children'].sample(n=1100,random_state=2)
new9 = sun_df[sun_df['Age_group']=='Youth'].sample(n=1100,random_state=2)
new10 = sun_df[sun_df['Age_group']=='Middle Aged'].sample(n=800,random_state=2)
new11 = sun_df[sun_df['Age_group']=='Senior'].sample(n=800,random_state=2)
new12 = sun_df[sun_df['Preferred_source-booking']=='SCA Website Booking'].sample(n=400,random_state=2)
new13 = sun_df[sun_df['Preferred_source-booking']=='Outside Booking'].sample(n=400,random_state=2)
new14 = sun_df[sun_df['Preferred_source-booking']=='Airport'].sample(n=800,random_state=2)
new15 = sun_df[sun_df['Preferred_source-booking']=='Tour Operator Portal'].sample(n=800,random_state=2)
new16 = sun_df[sun_df['Preferred_source-booking']=='Reservations Booking'].sample(n=800,random_state=2)
new17 = sun_df[sun_df['Preferred_source-booking']=='No Preference'].sample(n=800,random_state=2)
new18 = sun_df[sun_df['Preferred_source-booking']=='SY Vacation'].sample(n=800,random_state=2)
```

In [44]:
```
new_df = pd.concat([new1, new2, new3, new4, new5, new6, new7, new8, new9, new10, new11, new12, new13, new14, new15, \
                    new16, new17, new18])
```

In [45]:
```
new_df.drop_duplicates(inplace=True)
```

In [46]:
```
new_df.count()
```

Out[46]:
```
Unnamed: 0                 17345
CustomerID                 17345
Gender                     17345
Age_group                  17345
Uffy_membership_status     17345
Card_holder?               17345
Total_trips                17345
Total_amount_spent         17345
#Discounts                 17345
Preferred_travel_class     17345
#Upgrades                  17345
Preferred_source-booking   17345
dtype: int64
```

In [47]:
```
# We did not add the feature - "total amount spent" in X because it is correlated with "total trips".
# This helps in better analysis and faster execution.
X = new_df[['Gender', 'Age_group', 'Uffy_membership_status', 'Card_holder?', 'Total_trips', \
            '#Discounts', '#Upgrades', 'Preferred_travel_class', 'Preferred_source-booking']]
```

In [48]:
```
X.head()
```

Out[48]:

	Gender	Age_group	Uffy_membership_status	Card_holder?	Total_trips	#Discounts	#Upgrades	Preferred_travel_class	Preferred_source-booking
748819	F	Middle Aged	Elite	No	2	2	0.0	First Class	Reservations Booking
1453483	M	Senior	Elite	No	6	0	0.0	Coach	Outside Booking
403441	M	Young Adults	Elite	No	5	5	0.0	First Class	Airport
780044	M	Middle Aged	Elite	No	2	2	0.0	First Class	SCA Website Booking
22251	F	Young Adults	Elite	No	1	1	0.0	Coach	SCA Website Booking

In [49]:
```
import gower
from sklearn_extra.cluster import KMedoids
```

In [50]:
```
# We have used gower distance as distance metric
gower_dist = gower.gower_matrix(X)
```

Cluster quality analysis

In [56]:
```
from sklearn.metrics import silhouette_samples, silhouette_score
```

In [68]:
```
print(silhouette_score(gower_dist, X['cluster']))
```

```
0.22189361
```

Cluster quality was measured using silhouette coefficient which was highest for 5 cluster solution.

Silhouette score = 0.22

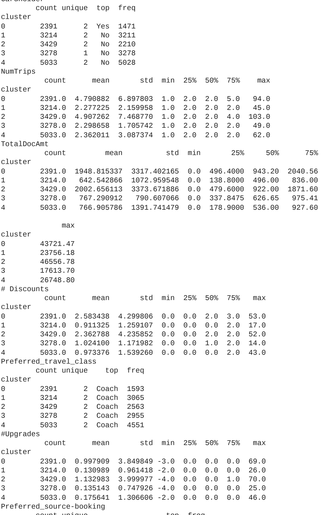
Cluster creation

In [52]:
```
clusterer = KMedoids(n_clusters = 5, random_state = 10, method = 'pam')
X['cluster'] = clusterer.fit_predict(gower_dist)
```

Visualizing Clusters

In [53]:
```
import seaborn as sns
sns.pairplot(X, hue='cluster')
```

Out[53]:
```
<seaborn.axisgrid.PairGrid at 0x28d9a1a9460>
```



Understanding CLusters

In [54]:
```
new_df['cluster'] = X['cluster']
```

In [55]:
```
# Summary statistics by cluster

print('gender')
print(new_df.groupby('cluster')['Gender'].describe())
print('Age Group')
print(new_df.groupby('cluster')['Age_group'].describe())
print('UffyMemberStatus')
print(new_df.groupby('cluster')['Uffy_membership_status'].describe())
print('CardHolder')
print(new_df.groupby('cluster')['Card_holder?'].describe())
print('NumTrips')
print(new_df.groupby('cluster')['Total_trips'].describe())
print('TotalDocAmt')
print(new_df.groupby('cluster')['Total_amount_spent'].describe())
print('# Discounts')
print(new_df.groupby('cluster')['#Discounts'].describe())
print('Preferred_travel_class')
print(new_df.groupby('cluster')['Preferred_travel_class'].describe())
print('#Upgrades')
print(new_df.groupby('cluster')['#Upgrades'].describe())
print('Preferred_source-booking')
print(new_df.groupby('cluster')['Preferred_source-booking'].describe())
```

```
gender
        count unique top  freq
cluster
0        2391      2   F  2214
1        3214      1   F  3214
2        3429      1   M  3429
3        3278      1   F  3278
4        5033      2   M  5032
Age Group
        count unique          top  freq
cluster
0        2391      5       Senior  1366
1        3214      5  Young Adults   973
2        3429      5       Senior  1473
3        3278      5     Children   927
4        5033      5  Young Adults  1306
UflyMemberStatus
        count unique         top  freq
cluster
0        2391      3    Standard  2144
1        3214      3  Not member  2937
2        3429      3    Standard  2577
3        3278      3  Not member  2375
4        5033      3  Not member  4838
CardHolder
        count unique  top  freq
cluster
0        2391      2  Yes  1471
1        3214      2   No  3211
2        3429      2   No  2210
3        3278      1   No  3278
4        5033      2   No  5028
NumTrips
         count      mean       std  min  25%  50%  75%    max
cluster
0       2391.0  4.790882  6.897803  1.0  2.0  2.0  5.0   94.0
1       3214.0  2.277225  2.159958  1.0  2.0  2.0  2.0   45.0
2       3429.0  4.907262  7.468770  1.0  2.0  2.0  4.0  103.0
3       3278.0  2.298658  1.705742  1.0  2.0  2.0  2.0   49.0
4       5033.0  2.362011  3.087374  1.0  2.0  2.0  2.0   62.0
TotalDocAmt
         count         mean          std  min       25%     50%      75%  \
cluster
0       2391.0  1948.815337  3317.402165  0.0  496.4000  943.20  2040.56
1       3214.0   642.542866  1072.959548  0.0  138.8000  496.00   836.00
2       3429.0  2002.656113  3373.671886  0.0  479.6000  922.00  1871.60
3       3278.0   767.290912   790.607066  0.0  337.8475  626.65   975.41
4       5033.0   766.905786  1391.741479  0.0  178.9000  536.00   927.60

             max
cluster
0       43721.47
1       23756.18
2       46556.78
3       17613.70
4       26748.80
# Discounts
         count      mean       std  min  25%  50%  75%   max
cluster
0       2391.0  2.583438  4.299806  0.0  0.0  2.0  3.0  53.0
1       3214.0  0.911325  1.259107  0.0  0.0  0.0  2.0  17.0
2       3429.0  2.362788  4.235852  0.0  0.0  2.0  2.0  52.0
3       3278.0  1.024100  1.171982  0.0  0.0  1.0  2.0  14.0
4       5033.0  0.973376  1.539260  0.0  0.0  0.0  2.0  43.0
Preferred_travel_class
        count unique    top  freq
cluster
0        2391      2  Coach  1593
1        3214      2  Coach  3065
2        3429      2  Coach  2563
3        3278      2  Coach  2955
4        5033      2  Coach  4551
#Upgrades
         count      mean       std  min  25%  50%  75%   max
cluster
0       2391.0  0.997909  3.849849 -3.0  0.0  0.0  0.0  69.0
1       3214.0  0.130989  0.961418 -2.0  0.0  0.0  0.0  26.0
2       3429.0  1.132983  3.999977 -4.0  0.0  0.0  1.0  70.0
3       3278.0  0.135143  0.747926 -4.0  0.0  0.0  0.0  25.0
4       5033.0  0.175641  1.306606 -2.0  0.0  0.0  0.0  46.0
Preferred_source-booking
        count unique                 top  freq
cluster
0        2391      7  SCA Website Booking  1683
1        3214      6     Outside Booking  1911
2        3429      7  SCA Website Booking  2291
3        3278      7  SCA Website Booking  2053
4        5033      7     Outside Booking  1810
```

Visualizations

Other visualizations in documents were made in tableau.

In [2]:
```
df = pd.read_csv('5_cluster_soln.csv')
```

In [3]:
```
df.head()
```

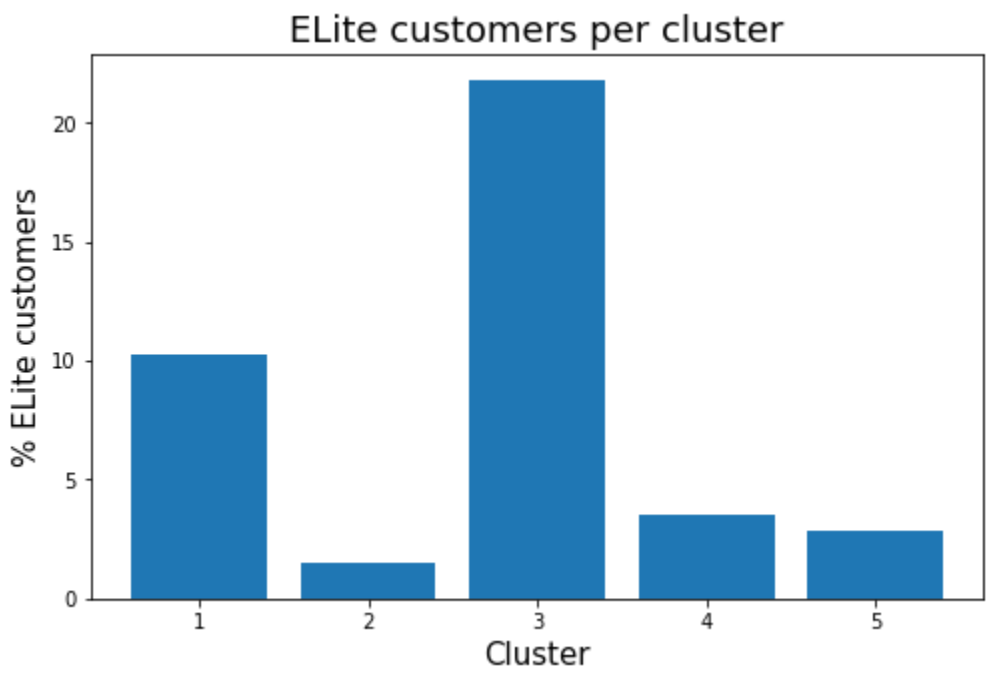
Out[3]:

	Unnamed: 0	Gender	Age_group	Ufly_membership_status	Card_holder?	Total_trips	#Discounts	#Upgrades	Preferred_travel_class	Preferred_source-booking	cluster
0	748819	F	Middle Aged	Elite	No	2	2	0.0	First Class	Reservations Booking	0
1	1453483	M	Senior	Elite	No	6	0	0.0	Coach	Outside Booking	2
2	403441	M	Young Adults	Elite	No	5	5	0.0	First Class	Airport	4
3	780044	M	Middle Aged	Elite	No	2	2	0.0	First Class	SCA Website Booking	2
4	22251	F	Young Adults	Elite	No	1	1	0.0	Coach	SCA Website Booking	3

In [13]:
```
total = df.groupby('cluster')['Gender'].count()
```

In [14]:
```
elite = df[df['Ufly_membership_status']=='Elite'].groupby('cluster')['Ufly_membership_status'].count()
```

In [16]:
```
plt.subplots(figsize=(8,5))
plt.bar([1,2,3,4,5], elite/total*100)
plt.title('ELite customers per cluster', fontsize=18)
plt.xlabel('Cluster', fontsize=15)
plt.ylabel('% ELite customers', fontsize=15)
plt.show()
```



In [19]:
```
first_class = df[df['Preferred_travel_class']=='First Class'].groupby('cluster')['Preferred_travel_class'].count()
```

In [21]:
```
plt.subplots(figsize=(8,5))
plt.bar([1,2,3,4,5], first_class/total*100)
plt.title('First class travelers per cluster', fontsize=18)
plt.xlabel('Cluster', fontsize=15)
plt.ylabel('% FC travelers', fontsize=15)
plt.show()
```

In [ ]:
```
```