In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [3]:
```
dt = pd.read_excel('./과제 데이터.xls')
```

In [4]:
```
# sample을 통한 컬럼 확인
dt.sample()
```

Out[4]:

	ID_USER	USER_STATE	USER_TIMEZONE	ID_HOTEL	HOTEL_CITY	HOTEL_STATE	HOTEL_TIMEZONE	Trip Type	R
3506	49680	TX	Central	89547	Atlanta	GA	Eastern	3	

In [5]:
```
# 누락 데이터 확인 및 Dtype 확인
dt.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4669 entries, 0 to 4668
Data columns (total 9 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   ID_USER          4669 non-null   int64 
 1   USER_STATE       4669 non-null   object
 2   USER_TIMEZONE    4669 non-null   object
 3   ID_HOTEL         4669 non-null   int64 
 4   HOTEL_CITY       4669 non-null   object
 5   HOTEL_STATE      4669 non-null   object
 6   HOTEL_TIMEZONE   4669 non-null   object
 7   Trip Type        4669 non-null   int64 
 8   Rating           4669 non-null   int64 
dtypes: int64(4), object(5)
memory usage: 328.4+ KB
```

In [6]:
```
# 중복 데이터 확인
dt.duplicated()
```

Out[6]:
```
0       False
1       False
2       False
3       False
4       False
        ...
4664    False
4665    False
4666    False
4667    False
4668    False
Length: 4669, dtype: bool
```

In [7]:
```
# 범주가 너무 넓고 경계가 애매하여 사용하지 않을 timezone 열을 제거
dt.drop(['USER_TIMEZONE', 'HOTEL_TIMEZONE'], axis=1, inplace=True)
```

In [8]:
```
# 데이터 표준화
dt['Trip Type'].replace({1:'Family', 2:'Couples', 3:'Business',\
                         4:'Solo travel', 5:'Friends'}, inplace=True)
```

In [9]:
```
# ID는 숫자형보다 문자열로의 쓰임이 더 맞다고 생각
dt['ID_USER'] = dt['ID_USER'].astype(object)
dt['ID_HOTEL'] = dt['ID_HOTEL'].astype(object)
print(dt.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4669 entries, 0 to 4668
Data columns (total 7 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   ID_USER      4669 non-null   object
 1   USER_STATE   4669 non-null   object
 2   ID_HOTEL     4669 non-null   object
 3   HOTEL_CITY   4669 non-null   object
 4   HOTEL_STATE  4669 non-null   object
 5   Trip Type    4669 non-null   object
 6   Rating       4669 non-null   int64 
dtypes: int64(1), object(6)
memory usage: 255.5+ KB
None
```

In [10]:
```
# 기본적인 전처리 결과 확인
print(dt)
```

```
     ID_USER USER_STATE ID_HOTEL   HOTEL_CITY HOTEL_STATE    Trip Type  Rating
0         45         GA   105170      Memphis          TN     Business       5
1         45         GA   223229   SanAntonio          TX     Business       5
2         45         GA   258688  Albuquerque          NM     Business       5
3         45         GA    98827       ELPaso          TX     Business       5
4         45         GA    99518   SanAntonio          TX     Business       3
...      ...        ...      ...          ...         ...          ...     ...
4664   65440         MI    95715  Minneapolis          MN  Solo travel       5
4665   65457         AZ  1027019    FortWorth          TX     Business       3
4666   65457         AZ   224458    Milwaukee          WI     Business       3
4667   65457         AZ   223749     Columbus          OH     Business       2
4668   65457         AZ    92744  Albuquerque          NM  Solo travel       3

[4669 rows x 7 columns]
```

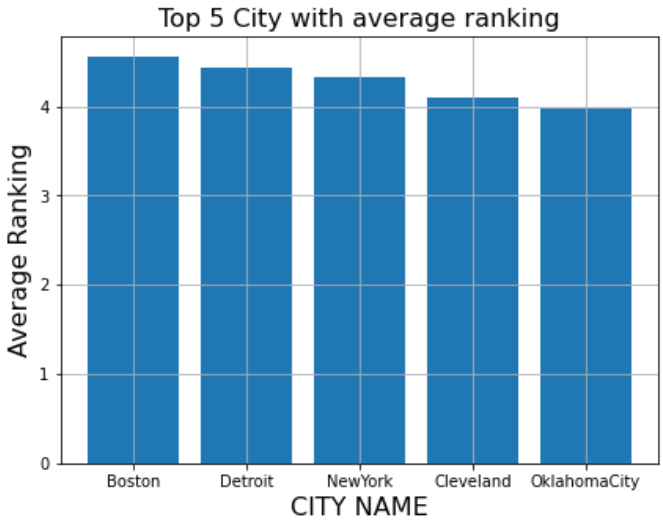
In [11]:
```
# <분석 내용>
# 1. 도시별 호텔들의 평균 평점 비교
# 2. trip type별 자주 가는 지역 상위 3개
# 3. state를 넘어가는 여행자 수의 비율 + trip_type 비율
# 4. state별 사람들의 평점
```

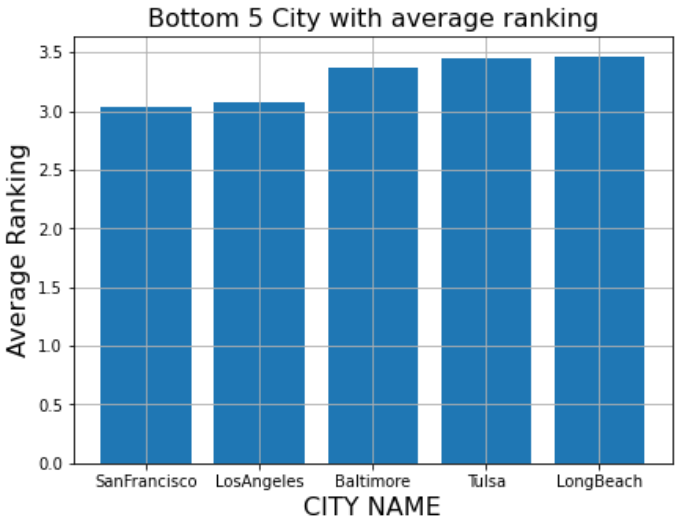
In [12]:
```
# 1. 도시별 호텔들의 평균 평점 비교
g_city = dt.groupby(['HOTEL_CITY'])
top_I = g_city['Rating'].mean().nlargest(5, keep='all').index.tolist()
top_D = g_city['Rating'].mean().nlargest(5, keep='all').tolist()
bottom_I = g_city['Rating'].mean().nsmallest(5, keep='all').index.tolist()
bottom_D = g_city['Rating'].mean().nsmallest(5, keep='all').tolist()

print(top_I, top_D, bottom_I, bottom_D)
```

```
['Boston', 'Detroit', 'NewYork', 'Cleveland', 'OklahomaCity'] [4.555555555555555, 4.4285
71428571429, 4.333333333333333, 4.1, 3.971264367816092] ['SanFrancisco', 'LosAngeles',
'Baltimore', 'Tulsa', 'LongBeach'] [3.0357142857142856, 3.0789473684210527, 3.3636363636
363638, 3.4444444444444446, 3.4642857142857144]
```

In [13]:
```
# 그래프 그리기
# subplot을 설정합니다
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(15, 5))

# 첫번째 subplot
ax1.bar(top_I, top_D)
ax1.set_title('Top 5 City with average ranking', fontsize=16)
ax1.set_xlabel('CITY NAME', fontsize=16)
ax1.set_ylabel('Average Ranking', fontsize=16)
ax1.grid(1)

# 두번째 subplot
```

```
ax2.bar(bottom_I, bottom_D)
ax2.set_title('Bottom 5 City with average ranking', fontsize=16)
ax2.set_xlabel('CITY NAME', fontsize=16)
ax2.set_ylabel('Average Ranking', fontsize=16)
ax2.grid(1)

plt.show()
```





1번 해석

- 호텔 이외의 주변 환경을 영향을 받아서 호텔의 평점이 높을 수 있다고 판단했다.
- 그래서 도시 별로 평점을 계산해서 상위 5개랑 하위 5개를 출력했다.
- 눈에 띄게 다르지는 않았다는 점에서 큰 영향을 미치지는 않는다고 판단했다.

In [14]:
```
# 2. trip type별 자주 가는 지역 상위 3개
# 가장 많이 방문한 지역 확인
g_STATE = dt.groupby(['HOTEL_STATE'])
print(g_STATE['ID_USER'].count())
```

```
HOTEL_STATE
AZ     369
CA     522
CO     103
FL     137
GA     164
IL      14
IN     187
KS     129
KY     123
MA       9
MD      22
MI      51
MN      42
NC     235
NE      85
NM     211
NV      89
NY      12
OH     220
OK     291
OR      87
PA      11
TN     229
TX    1171
VA      27
WA      60
```

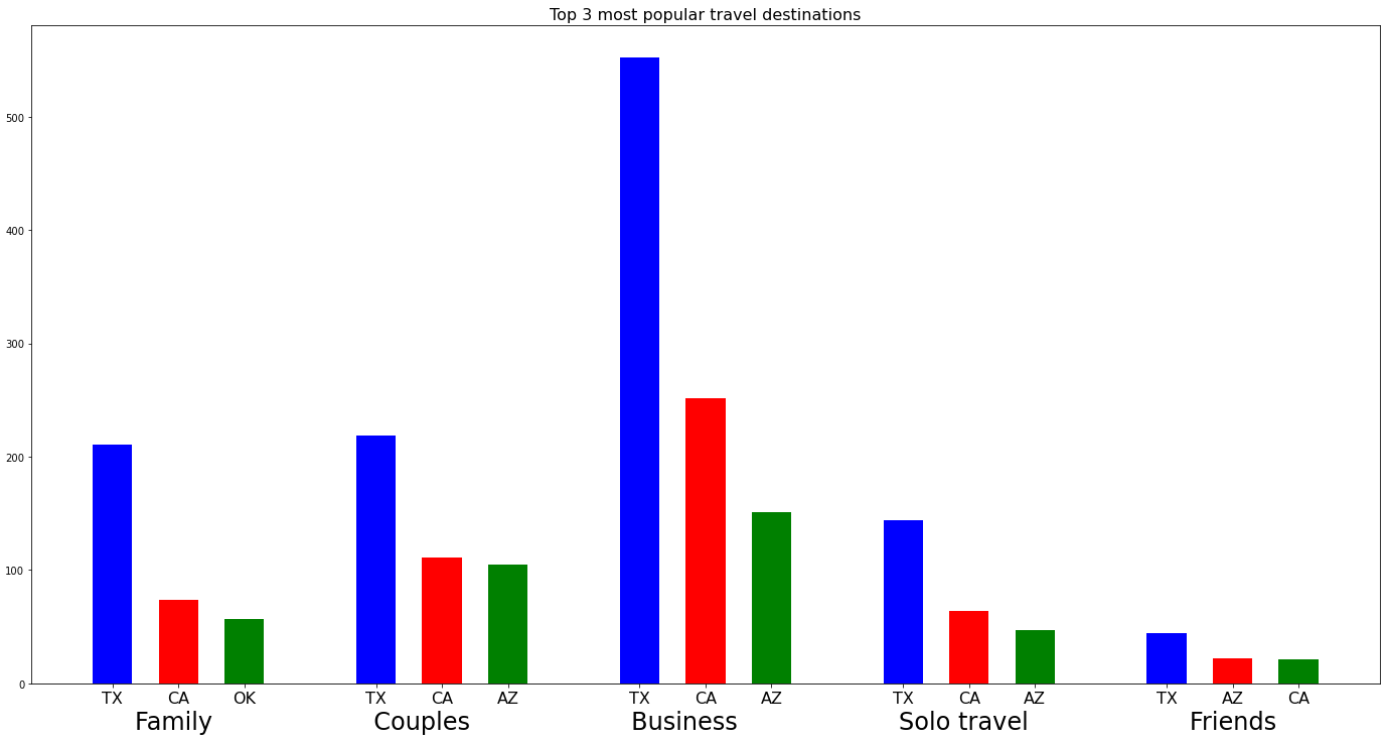
```
WI      69
Name: ID_USER, dtype: int64
```

In [15]:
```python
# trip type 별 자주가는 상위 3개 지역 선택
g_trip = dt.groupby(['Trip Type'])
g_trip_1 = g_trip.get_group('Family')
top3_1 = g_trip_1['HOTEL_STATE'].value_counts().nlargest(3).index.tolist()
top3_1D = g_trip_1['HOTEL_STATE'].value_counts().nlargest(3).tolist()

g_trip_2 = g_trip.get_group('Couples')
top3_2 = g_trip_2['HOTEL_STATE'].value_counts().nlargest(3).index.tolist()
top3_2D = g_trip_2['HOTEL_STATE'].value_counts().nlargest(3).tolist()

g_trip_3 = g_trip.get_group('Business')
top3_3 = g_trip_3['HOTEL_STATE'].value_counts().nlargest(3).index.tolist()
top3_3D = g_trip_3['HOTEL_STATE'].value_counts().nlargest(3).tolist()

g_trip_4 = g_trip.get_group('Solo travel')
top3_4 = g_trip_4['HOTEL_STATE'].value_counts().nlargest(3).index.tolist()
top3_4D = g_trip_4['HOTEL_STATE'].value_counts().nlargest(3).tolist()

g_trip_5 = g_trip.get_group('Friends')
top3_5 = g_trip_5['HOTEL_STATE'].value_counts().nlargest(3).index.tolist()
top3_5D = g_trip_5['HOTEL_STATE'].value_counts().nlargest(3).tolist()

top3 = [top3_1, top3_2, top3_3, top3_4, top3_5]
top3D = [top3_1D, top3_2D, top3_3D, top3_4D, top3_5D]
print(top3, top3D)
```

```
[['TX', 'CA', 'OK'], ['TX', 'CA', 'AZ'], ['TX', 'CA', 'AZ'], ['TX', 'CA', 'AZ'], ['TX',
'AZ', 'CA']] [[211, 74, 57], [219, 111, 105], [553, 252, 151], [144, 64, 47], [44, 22, 2
1]]
```

In [16]:
```python
# subplot을 설정합니다
fig, ax = plt.subplots(figsize=(24, 12))
arr = [0, 0.5, 1, 2, 2.5, 3, 4, 4.5, 5, 6, 6.5, 7, 8, 8.5, 9]

# Family 그래프
ax.bar(arr[0], top3_1D[0], color='b', width=0.3, tick_label=top3_1[0])
ax.bar(arr[1], top3_1D[1], color='r', width=0.3, tick_label=top3_1[1])
ax.bar(arr[2], top3_1D[2], color='g', width=0.3, tick_label=top3_1[2])
#plt.legend(loc='center left', fontsize=12)

# Couples 그래프
ax.bar(arr[3], top3_2D[0], color='b', width=0.3, label=top3_2[0])
ax.bar(arr[4], top3_2D[1], color='r', width=0.3, label=top3_2[1])
ax.bar(arr[5], top3_2D[2], color='g', width=0.3, label=top3_2[2])

# Business 그래프
ax.bar(arr[6], top3_3D[0], color='b', width=0.3, label=top3_3[0])
ax.bar(arr[7], top3_3D[1], color='r', width=0.3, label=top3_3[1])
ax.bar(arr[8], top3_3D[2], color='g', width=0.3, label=top3_3[2])

# Solo travle 그래프
ax.bar(arr[9], top3_4D[0], color='b', width=0.3, label=top3_4[0])
ax.bar(arr[10], top3_4D[1], color='r', width=0.3, label=top3_4[1])
ax.bar(arr[11], top3_4D[2], color='g', width=0.3, label=top3_4[2])

# Friends 그래프
ax.bar(arr[12], top3_5D[0], color='b', width=0.3, label=top3_5[0])
ax.bar(arr[13], top3_5D[1], color='r', width=0.3, label=top3_5[1])
ax.bar(arr[14], top3_5D[2], color='g', width=0.3, label=top3_5[2])

# x축 레이블과 제목을 설정합니다
ax.set_xticks(arr)
ax.set_xticklabels(top3_1+top3_2+top3_3+top3_4+top3_5, fontsize=16)
```

```
ax.set_xlabel('Family                    Couples                    \
Business                    Solo travel                    Friends', fontsize=24)
ax.set_title('Top 3 most popular travel destinations', fontsize=16)

plt.show()
```



2번 해석

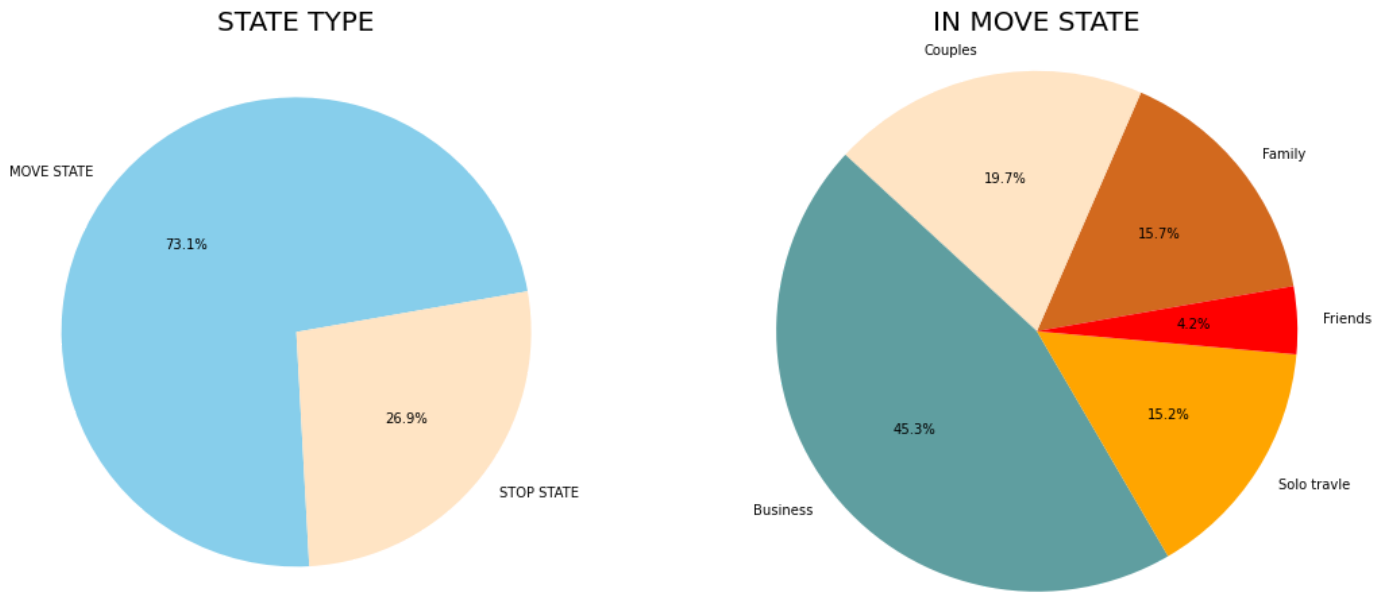
- trip type별로 자주 가는 상위 3개 지역을 찾아내려고 했다.
- 여행의 유형과 잘 어울리는 지역이 있는지 알아내고 싶었다.
- 단순히 여행을 많이 가는 지역을 찾아보기도 했다.
- trip type별로 상위 3개를 선택했을 때, 단순히 여행을 많이 가는 지역에서 높은 순위를 차지했던 TX,CA,AZ,OK가 높은 순위를 차지했다.
- 따라서 여행의 유형과 장소의 관계는 크게 있어보이지 않았다.

In [17]:
```
# 3. state를 넘어가는 여행자 수의 비율  +  trip type 비율
# 여행자들의 여행 스타일을 알기
dt_STATE = dt[dt['USER_STATE'] != dt['HOTEL_STATE']]
print('STATE를 이동한 여행자 비율 :', round(len(dt_STATE)/len(dt)*100,1),'%')
print('Family 목적인 STATE를 이동한 여행자 비율 :', \
      round(len(dt_STATE[dt_STATE['Trip Type']=='Family'])/len(dt_STATE)*100,1),'%')
print('Couples 목적인 STATE를 이동한 여행자 비율 :', \
      round(len(dt_STATE[dt_STATE['Trip Type']=='Couples'])/len(dt_STATE)*100,1),'%')
print('Business 목적인 STATE를 이동한 여행자 비율 :', \
      round(len(dt_STATE[dt_STATE['Trip Type']=='Business'])/len(dt_STATE)*100,1),'%')
print('Solo travel 목적인 STATE를 이동한 여행자 비율 :', \
      round(len(dt_STATE[dt_STATE['Trip Type']=='Solo travel'])/len(dt_STATE)*100,1),'%'
print('Friends 목적인 STATE를 이동한 여행자 비율 :', \
      round(len(dt_STATE[dt_STATE['Trip Type']=='Friends'])/len(dt_STATE)*100,1),'%')
```

```
STATE를 이동한 여행자 비율 : 73.1 %
Family 목적인 STATE를 이동한 여행자 비율 : 15.7 %
Couples 목적인 STATE를 이동한 여행자 비율 : 19.7 %
Business 목적인 STATE를 이동한 여행자 비율 : 45.3 %
Solo travel 목적인 STATE를 이동한 여행자 비율 : 15.2 %
Friends 목적인 STATE를 이동한 여행자 비율 : 4.2 %
```

In [18]:
```
# 그래프 그리기
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(18, 8))
```

```
# STATE TYPE subplot 그리기
labels_1 = ['MOVE STATE', 'STOP STATE']
sizes_1 = [round(len(dt_STATE)/len(dt)*100,1), (100-round(len(dt_STATE)/len(dt)*100,1))]
colors_1 = ['skyblue', 'bisque']

ax1.set_title('STATE TYPE', fontsize=20)
ax1.pie(sizes_1, labels=labels_1, colors=colors_1, autopct='%1.1f%%', startangle=10)

# IN MOVE STATE subplot 그리기
labels_2 = ['Family', 'Couples', 'Business', 'Solo travle', 'Friends']
sizes_2 = [round(len(dt_STATE[dt_STATE['Trip Type']=='Family'])/len(dt_STATE)*100,1),
        round(len(dt_STATE[dt_STATE['Trip Type']=='Couples'])/len(dt_STATE)*100,1),
        round(len(dt_STATE[dt_STATE['Trip Type']=='Business'])/len(dt_STATE)*100,1),
        round(len(dt_STATE[dt_STATE['Trip Type']=='Solo travel'])/len(dt_STATE)*100,1),
        round(len(dt_STATE[dt_STATE['Trip Type']=='Friends'])/len(dt_STATE)*100,1)]
colors_2 = ['chocolate', 'bisque', 'cadetblue', 'orange', 'red']

ax2.set_title('IN MOVE STATE', fontsize=20)
ax2.pie(sizes_2, labels=labels_2, colors=colors_2, autopct='%1.1f%%', startangle=10)


plt.axis('equal')
plt.show()
```

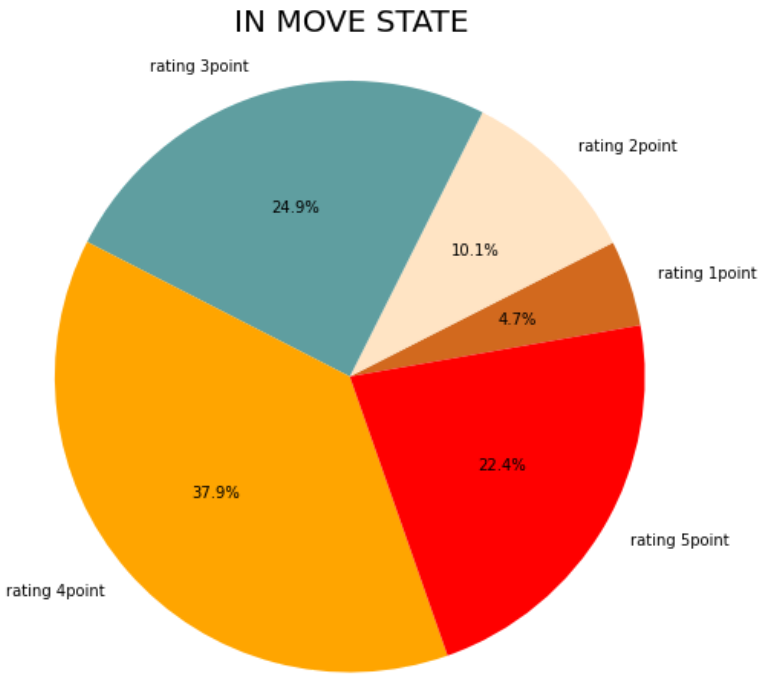


3번 해석

- 사람들의 여행 스타일을 알아보기 위해서 데이터의 비율을 살펴봤다.
- 우선 STATE를 기준으로 STATE를 이동하는 비율을 살펴봤고, STATE를 이동하는 비율 중에 어떤 목적이 제일 많은지 확인했다.
- 결과적으로 사람들은 STATE를 이동을 많이 하고, Business 혹은 Couples들인 경우가 많았다.

In [19]:
```
# 4. state별 사람들의 평점
labels_1 = ['rating 1point', 'rating 2point', 'rating 3point', \
            'rating 4point', 'rating 5point']
sizes_1 = [round(len(dt[dt['Rating']==1])/len(dt)*100,1),
        round(len(dt[dt['Rating']==2])/len(dt)*100,1),
        round(len(dt[dt['Rating']==3])/len(dt)*100,1),
        round(len(dt[dt['Rating']==4])/len(dt)*100,1),
        round(len(dt[dt['Rating']==5])/len(dt)*100,1)]
colors_1 = ['chocolate', 'bisque', 'cadetblue', 'orange', 'red']

plt.figure(figsize=(16,8))
```

```
plt.title('IN MOVE STATE', fontsize=20)
plt.pie(sizes_1, labels=labels_1, colors=colors_1, autopct='%1.1f%%', startangle=10)

plt.axis('equal')
plt.show()
```



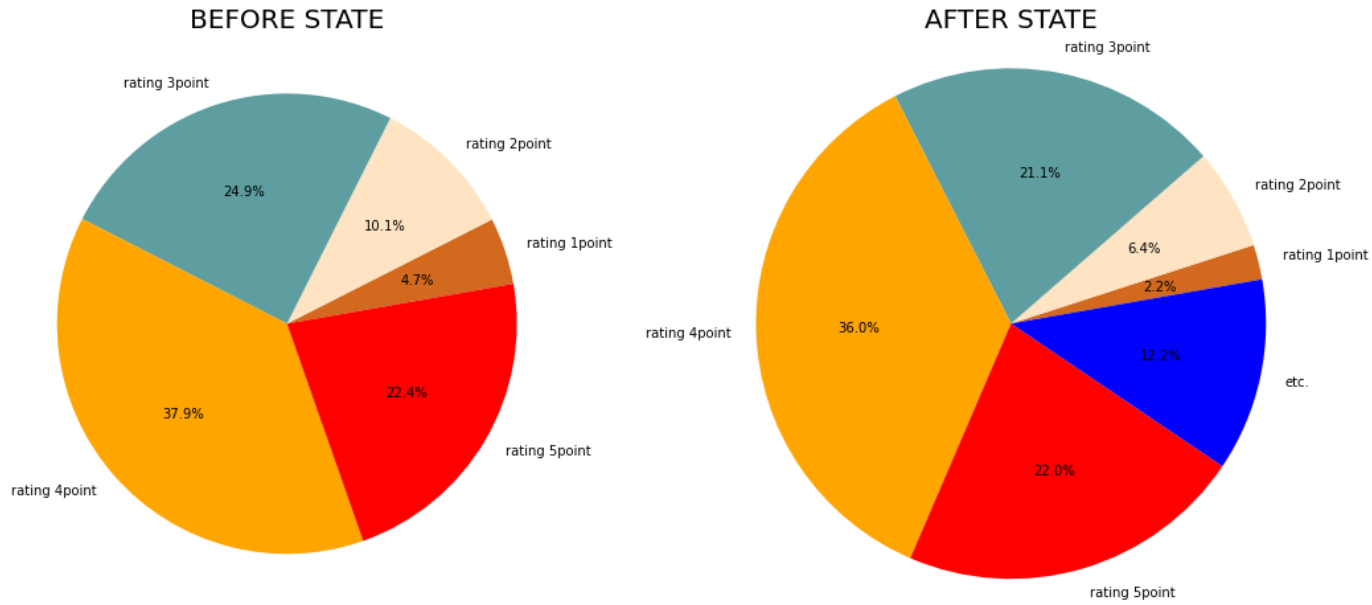
In [20]:
```
print(dt['Rating'].describe())
```

```
count    4669.000000
mean        3.632898
std         1.078614
min         1.000000
25%         3.000000
50%         4.000000
75%         4.000000
max         5.000000
Name: Rating, dtype: float64
```

In [21]:
```
# 하위 25%보다 낮은 평균 평점을 가진 사람은 평점을 낮게 줄 확률이 높은 사람으로 판단!
# 하위 25%의 데이터를 지우고 점수의 비율이 어떻게 변하는지 확인
g_STATE = dt.groupby(['ID_USER'])
g_STATE_mean = g_STATE['Rating'].mean()
low_list = g_STATE_mean[g_STATE_mean<3].index.tolist()
dt_drop = dt.drop(dt[dt['ID_USER'].isin(low_list)].index )
```

In [22]:
```
# 그래프 그리기
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(18, 8))

# STATE TYPE subplot 그리기
labels_1 = ['rating 1point', 'rating 2point', 'rating 3point', \
            'rating 4point', 'rating 5point']
sizes_1 = [round(len(dt[dt['Rating']==1])/len(dt)*100,1),
        round(len(dt[dt['Rating']==2])/len(dt)*100,1),
        round(len(dt[dt['Rating']==3])/len(dt)*100,1),
        round(len(dt[dt['Rating']==4])/len(dt)*100,1),
        round(len(dt[dt['Rating']==5])/len(dt)*100,1)]
colors_1 = ['chocolate', 'bisque', 'cadetblue', 'orange', 'red']


ax1.set_title('BEFORE STATE', fontsize=20)
ax1.pie(sizes_1, labels=labels_1, colors=colors_1, autopct='%1.1f%%', startangle=10)
```

```
# IN MOVE STATE subplot 그리기
labels_2 = ['rating 1point', 'rating 2point', 'rating 3point', \
            'rating 4point', 'rating 5point', 'etc.']
sizes_2 = [round(len(dt_drop[dt_drop['Rating']==1])/len(dt)*100,1),
        round(len(dt_drop[dt_drop['Rating']==2])/len(dt)*100,1),
        round(len(dt_drop[dt_drop['Rating']==3])/len(dt)*100,1),
        round(len(dt_drop[dt_drop['Rating']==4])/len(dt)*100,1),
        round(len(dt_drop[dt_drop['Rating']==5])/len(dt)*100,1),
        round((len(dt)-len(dt_drop))/len(dt)*100,1)]
colors_2 = ['chocolate', 'bisque', 'cadetblue', 'orange', 'red', 'blue']


ax2.set_title('AFTER STATE', fontsize=20)
ax2.pie(sizes_2, labels=labels_2, colors=colors_2, autopct='%1.1f%%', startangle=10)


plt.axis('equal')
plt.show()
```



4번 해석

- 평점을 주는 특징을 찾기 위해서 state별 평균 평점을 살펴보았지만, 눈에 띄는 특징을 찾지는 못했다.
- 다음으로 평점을 낮게 준 사람들은 전반적으로 낮게 주려는 경향이 있는지 살펴봤다.
- describe()와 점수별 데이터 비율을 확인 후, 평균 평점이 하위 25%인 데이터를 대상으로 이들을 제거했을 때, 다른 데이터의 비율의 변화를 확인해봤다.
- 12.2%가 감소했는데 상대적으로 3,2,1점에서 많이 감소했다는 점에서 낮은 점수를 주는 사람은 낮은 점수를 주는 경향이 있음을 알 수 있었다.