In [1]:
```
import pandas as pd
import numpy as np
import os
from matplotlib import pyplot as plt
import seaborn as sns
import matplotlib.cm as cm
from scipy.stats import chi2_contingency
```

In [2]:
```
pwd = os.getcwd()
file_path=pwd+'/data.xls'
```

In [3]:
```
df= pd.read_excel(io=file_path,sheet_name='Data')
df_triptype=pd.read_excel(file_path,sheet_name='Trip Type')
df_triptype.index+=1
```

In [4]:
```
df
```

Out[4]:

	ID_USER	USER_STATE	USER_TIMEZONE	ID_HOTEL	HOTEL_CITY	HOTEL_STATE	HOTEL_TIMEZONE
0	45	GA	Eastern	105170	Memphis	TN	Centra
1	45	GA	Eastern	223229	SanAntonio	TX	Centra
2	45	GA	Eastern	258688	Albuquerque	NM	Mountair
3	45	GA	Eastern	98827	ELPaso	TX	Centra
4	45	GA	Eastern	99518	SanAntonio	TX	Centra
...	...	...	...	...	...	...	..
4664	65440	MI	Eastern	95715	Minneapolis	MN	Centra
4665	65457	AZ	Mountain	1027019	FortWorth	TX	Centra
4666	65457	AZ	Mountain	224458	Milwaukee	WI	Centra
4667	65457	AZ	Mountain	223749	Columbus	OH	Easterr
4668	65457	AZ	Mountain	92744	Albuquerque	NM	Mountair

4669 rows × 9 columns

In [5]:
```
df.isnull().sum()
```

Out[5]:
```
ID_USER           0
USER_STATE        0
USER_TIMEZONE     0
ID_HOTEL          0
HOTEL_CITY        0
HOTEL_STATE       0
HOTEL_TIMEZONE    0
Trip Type         0
Rating            0
dtype: int64
```

데이터의 결측치를 먼저 확인하는게 우선이라고 판단하여 결측치 확인을 먼저 진행하였습니다.

In [6]:
```
df_triptype
```

Out[6]:

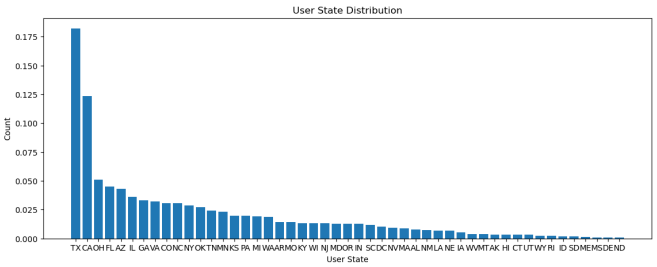
	Trip Type
1	Family

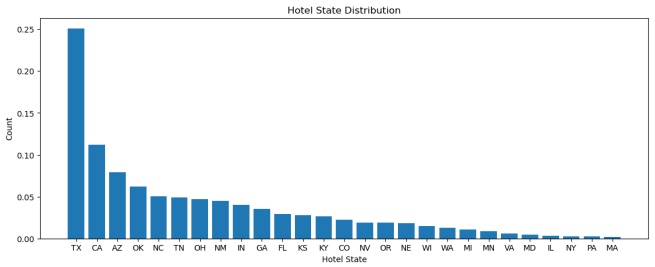
2	Couples
3	Business
4	Solo travel
5	Friends

데이터 셋 전체에 대한 전반적인 분석

1. User , Hotel State barplot

In [7]:
```
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(30, 5))

user_state_counts = df['USER_STATE'].value_counts(normalize=True)
ax1.bar(user_state_counts.index, user_state_counts.values)
ax1.set_xlabel('User State')
ax1.set_ylabel('Count')
ax1.set_title('User State Distribution')

hotel_state_counts = df['HOTEL_STATE'].value_counts(normalize=True)
ax2.bar(hotel_state_counts.index, hotel_state_counts.values)
ax2.set_xlabel('Hotel State')
ax2.set_ylabel('Count')
ax2.set_title('Hotel State Distribution')

plt.show()
```

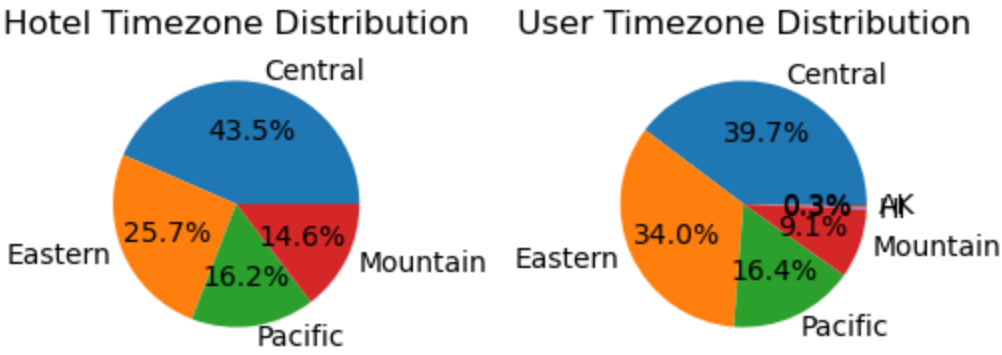




USER_STATE, HOTEL_STATE 각각의 데이터에 대한 분석을 진행하려고 하였으나, 각 데이터들의 범주가 너무 많아서 이를 clustering 할 수 있는 방법에 대해 고민을 하였습니다. 각각의 주 들은 같은 주 내에서도 다른 시간대를 가질 수 있지만, 주어진 데이터를 활용하여 clustering 할 수 있는 방법을 고민하였고, USER_TIMEZONE으로 묶는 방법을 선택하게 되었고, 오차를 감안하고 분석을 진행해야겠다고 생각했습니다. 시각화를 위해 barplot 을 선택한 이유는 pie chart로 그리면 범주가 너무 많아 가독성이 떨어진다고 판단하였습니다.

2. User , Hotel Timezone Pie plot

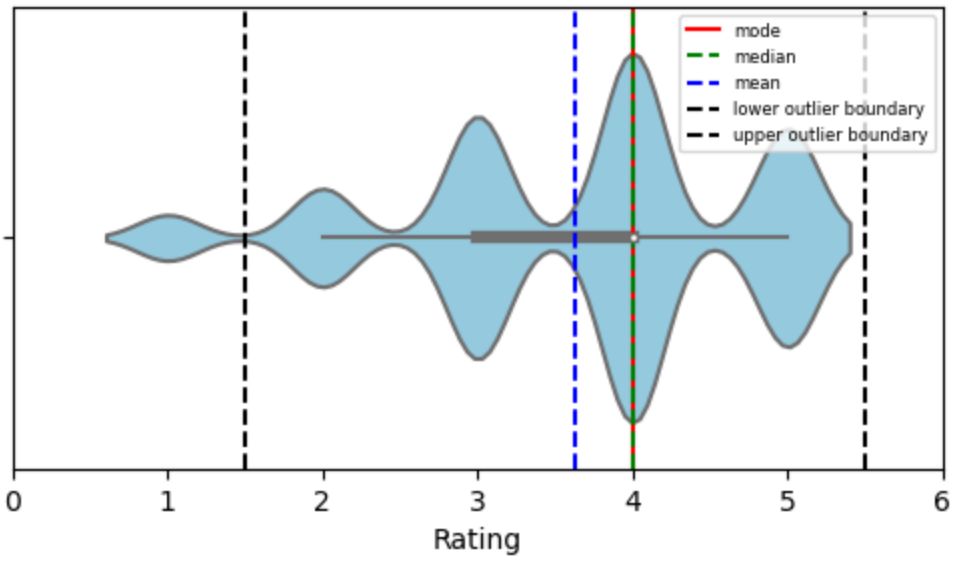
In [8]:
```
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(6, 2))

timezone_counts = df['HOTEL_TIMEZONE'].value_counts()
ax1.pie(timezone_counts, labels=timezone_counts.index, autopct='%1.1f%%')
ax1.set_title('Hotel Timezone Distribution')

timezone_counts = df['USER_TIMEZONE'].value_counts()
ax2.pie(timezone_counts, labels=timezone_counts.index, autopct='%1.1f%%')
ax2.set_title('User Timezone Distribution')

plt.show()
```



각 TIMEZONE 마다의 비율의 차이는 있었지만 각각의 분포들의 순위를 매겨봤을 때 똑같은 결과가 나온다는 점을 알 수 있었습니다.

3. Rating Violin Plot

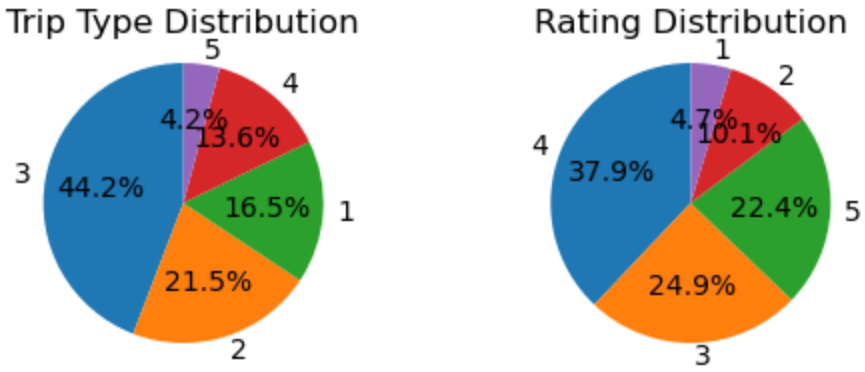
In [9]:
```
all_rating = df['Rating']

minimum = np.min(all_rating)
maximum= np.max(all_rating)
q1 = np.quantile(all_rating, 0.25)
q3 = np.quantile(all_rating, 0.75)
iqr = q3 - q1
upper = q3 + 1.5*iqr
lower = q1 - 1.5*iqr

fig, ax = plt.subplots(figsize=(6, 3))
sns.violinplot(x=all_rating, ax=ax, color='skyblue')
plt.axvline(all_rating.mode()[0], color='r', linestyle='-', label='mode')
plt.axvline(all_rating.median(), color='g', linestyle='--', label='median')
plt.axvline(all_rating.mean(), color='b', linestyle='--', label='mean')
plt.axvline(lower, color='k', linestyle='--', label='lower outlier boundary')
plt.axvline(upper, color='k', linestyle='--', label='upper outlier boundary')
plt.xlim(minimum-1, maximum+1)
plt.legend(fontsize=6)
plt.show()
```



먼저 Rating에 대해 어떠한 형태의 데이터로 생각을 해야할 지에 대한 고민을 하였습니다. 각각의 점수에 대해 좋고 나쁨의 정도를 판단할 수 있는 척도가 없었고, 명세가 없었기 때문에 고민을 했는데, 순서형 데이터라고 판단하고 분석을 진행하였습니다. 위 바이올린 플롯으로 negatively skewed Data 의 패턴을 가진다는 점을 확인 할 수 있었습니다.

4. Trip Type , Rating Pie Chart

In [10]:
```
triptype_df = df['Trip Type']
rating_df = df['Rating']
triptype_counts = triptype_df.value_counts(normalize=True)
rating_counts= rating_df.value_counts(normalize=True)
fig, ax = plt.subplots(1,2,figsize=(6,2))
ax[0].pie(triptype_counts, labels=triptype_counts.index, autopct='%1.1f%%', startangle=9
ax[0].axis('equal')
ax[0].set_title('Trip Type Distribution')
ax[1].pie(rating_counts, labels=rating_counts.index, autopct='%1.1f%%', startangle=90)
ax[1].axis('equal')
ax[1].set_title('Rating Distribution')
plt.show()
```



전체 데이터에 대한 Trip Type 의 비율은 Business > Couples > Family > Solo travel > Friends 순으로 나타내어 진다는 것을 확인할 수 있었습니다.

Correlation

USER_TIMEZONE 과 HOTEL_TIMEZONE의 상관관계

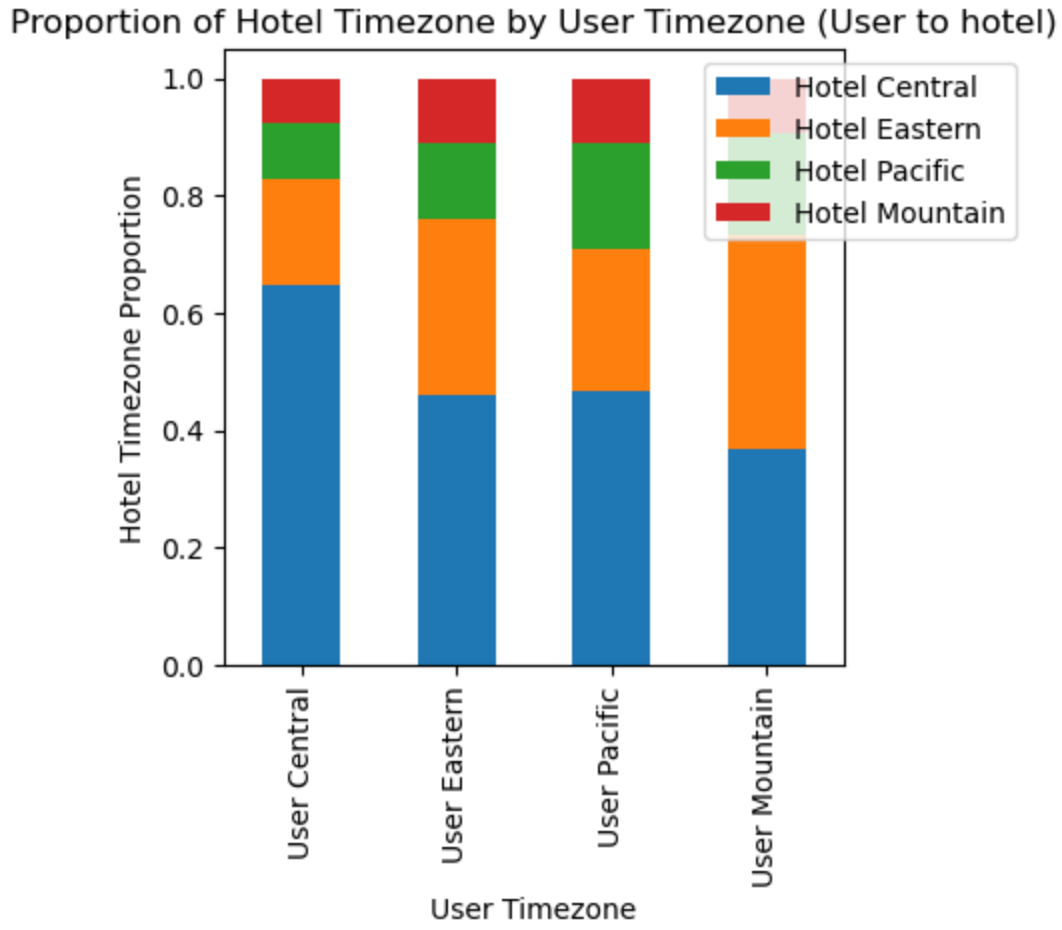
In [11]:
```
ctr_df=df[(df['USER_TIMEZONE']=='Central')]
ctr_value_counts = ctr_df['HOTEL_TIMEZONE'].value_counts(normalize=True)
est_df = df[df['USER_TIMEZONE']=='Eastern']
est_value_counts = est_df['HOTEL_TIMEZONE'].value_counts(normalize=True)
pcf_df=df[(df['USER_TIMEZONE']=='Pacific')]
pcf_value_counts = pcf_df['HOTEL_TIMEZONE'].value_counts(normalize=True)
mnt_df = df[df['USER_TIMEZONE']=='Mountain']
mnt_value_counts = mnt_df['HOTEL_TIMEZONE'].value_counts(normalize=True)

#{'Central': 0, 'Eastern': 1, 'Pacific': 2, 'Mountain': 3, 'HI': 4, 'AK': 5}

# 각 지역의 HOTEL_TIMEZONE 비율을 리스트로 저장
ctr_list = ctr_value_counts.tolist()
est_list = est_value_counts.tolist()
pcf_list = pcf_value_counts.tolist()
mnt_list = mnt_value_counts.tolist()

# 리스트를 데이터프레임으로 결합
timezone_df_plot = pd.DataFrame({'User Central': ctr_list,
                        'User Eastern': est_list,
                        'User Pacific': pcf_list,
                        'User Mountain': mnt_list})

timezone_df_plot = timezone_df_plot.T.rename(columns={0: 'Hotel Central', 1: 'Hotel East
```

In [12]:
```
timezone_df_plot
```

Out[12]:

	Hotel Central	Hotel Eastern	Hotel Pacific	Hotel Mountain
User Central	0.646900	0.182210	0.094340	0.076550
User Eastern	0.459068	0.301637	0.130353	0.108942
User Pacific	0.468057	0.242503	0.178618	0.110821
User Mountain	0.367681	0.365340	0.173302	0.093677

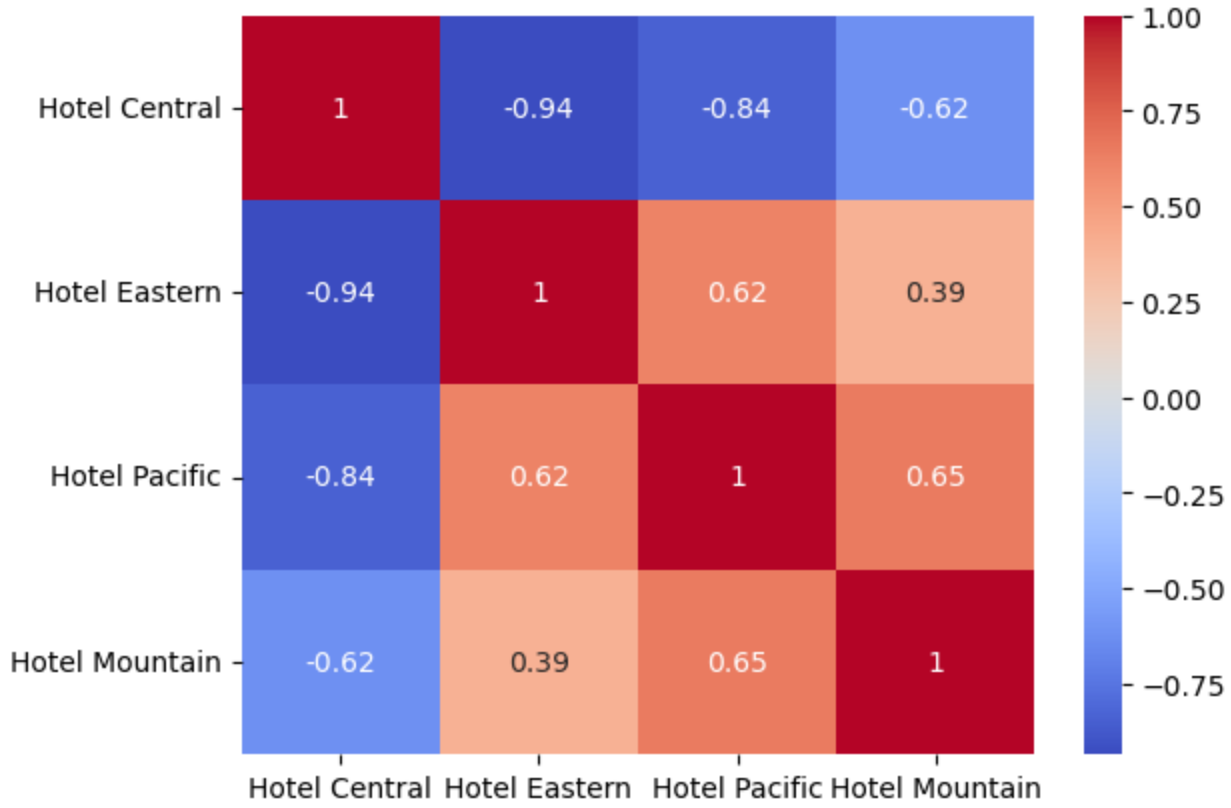
In [13]:
```
# 막대 그래프 그리기
ax = timezone_df_plot.plot(kind='bar', stacked=True, figsize=(4, 4))

# 축 레이블과 제목 설정
ax.set_xlabel('User Timezone')
ax.set_ylabel('Hotel Timezone Proportion')
ax.set_title('Proportion of Hotel Timezone by User Timezone (User to hotel)')

# 범례 위치 및 크기 조정
ax.legend(loc='upper right', bbox_to_anchor=(1.3, 1))

# 그래프 보여주기
plt.show()
```



Correlation Coefficient

In [14]:
```
timezone_df_corr = timezone_df_plot.corr()
sns.heatmap(timezone_df_corr, annot=True, cmap='coolwarm')
```

Out[14]:
```
<Axes: >
```



카이제곱검정

In [15]:
```
obs = pd.crosstab(df['USER_TIMEZONE'], df['HOTEL_TIMEZONE'])
chi2, p, dof, expected = chi2_contingency(obs)
print(f"chi2 = {chi2}, p-value = {p}, degrees of freedom = {dof}\n")
```

```
chi2 = 1476.4587971967267, p-value = 5.834347019266235e-306, degrees of freedom = 15
```

p-value 의 값을 통해 두 변수는 유의미한 상관관계가 있다고 보았습니다, 또한 chi2 값이 매우 크기 때문에 강한 상관관계가 있다고 판단하였습니다. USER_TIMEZONE 은 실제로 {'Central': 0, 'Eastern': 1, 'Pacific': 2, 'Mountain': 3, 'HI': 4, 'AK': 5} 총 6개의 범주가 존재합니다. 저는 그 중 수가 적은 HI,AK 를 제외한 {'Central': 0, 'Eastern': 1, 'Pacific': 2, 'Mountain': 3} 에 대해서만 분석을 진행 하였습니다. USER_TIMEZONE 의 각 케이스당 방문한 호텔들의 HOTEL_TIMEZONE 에 해당하는 값을 추출하였고, 각각의 표본에 대해 표본의 크기가 다르기 때문에 normalize를 진행하고, corr() 메서드와 heatmap 을 이용한 시각화를 통해 둘의 상관관계를 살펴보았습니다. p-value 의 값을 통해 두 변수는 유의미한 상관관계가 있다고 보았습니다, 또한 chi2 값이 매우 크기 때문에 강한 상관관계가 있다고 판단하였습니다. 흥미로웠던 점은 특히나 Central과 Eastern이 강한 상관관계를 가지며, Mountain 의 경우에만 비교적 다른 attributes 과 약한 상관관계를 가진다는 점이었습니다.

Trip type 과 Rating 의 상관관계

In [16]:
```
family_df=df[(df['Trip Type']==1)]
family_rating_counts = family_df['Rating'].value_counts(normalize=True)
couples_df = df[df['Trip Type']==2]
couples_rating_counts = couples_df['Rating'].value_counts(normalize=True)
business_df=df[(df['Trip Type']==3)]
business_rating_counts = business_df['Rating'].value_counts(normalize=True)
solo_df = df[df['Trip Type']==4]
solo_rating_counts = solo_df['Rating'].value_counts(normalize=True)
friends_df = df[df['Trip Type']==5]
friends_rating_counts = friends_df['Rating'].value_counts(normalize=True)
#{'Family': 1, 'Couples': 2, 'Business': 3, 'Solo travel': 4, 'Friends': 5}
```

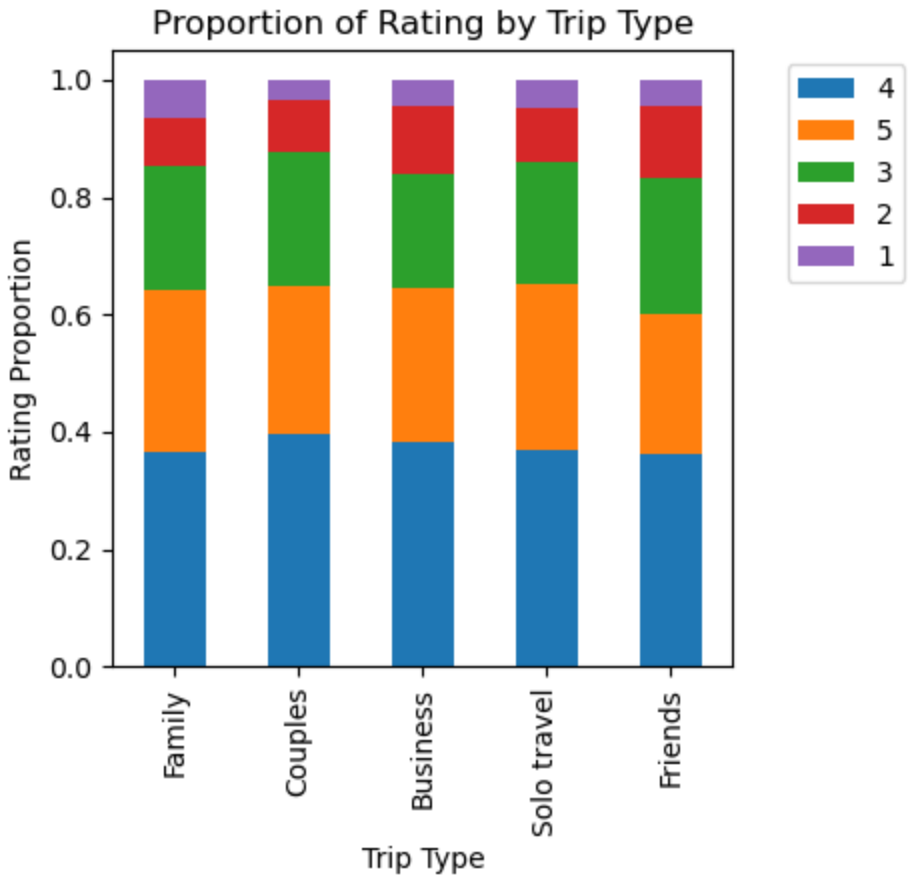
```
# 각 지역의 HOTEL_TIMEZONE 비율을 리스트로 저장
family_list = family_rating_counts.tolist()
couples_list = couples_rating_counts.tolist()
business_list = business_rating_counts.tolist()
solo_list = solo_rating_counts.tolist()
friends_list=friends_rating_counts.tolist()

# 리스트를 데이터프레임으로 결합
triptype_df_plot = pd.DataFrame({'Family': family_list,
                         'Couples': couples_list,
                         'Business': business_list,
                         'Solo travel': solo_list,
                         'Friends':friends_list
                        })

triptype_df_plot = triptype_df_plot.T.rename(columns={0: '4', 1: '5', 2: '3', 3: '2' ,4:
```

In [17]:
```
triptype_df_plot
```

Out[17]:

	4	5	3	2	1
Family	0.367056	0.276265	0.210117	0.081712	0.064851
Couples	0.395025	0.253731	0.226866	0.089552	0.034826
Business	0.381183	0.264791	0.194956	0.113482	0.045587
Solo travel	0.368504	0.283465	0.206299	0.092913	0.048819
Friends	0.362245	0.239796	0.229592	0.122449	0.045918

In [18]:
```
# 막대 그래프 그리기
ax = triptype_df_plot.plot(kind='bar', stacked=True, figsize=(4, 4))

# 축 레이블과 제목 설정
ax.set_xlabel('Trip Type')
ax.set_ylabel('Rating Proportion')
ax.set_title('Proportion of Rating by Trip Type')

# 범례 위치 및 크기 조정
ax.legend(loc='upper right', bbox_to_anchor=(1.3, 1))

# 그래프 보여주기
plt.show()
```



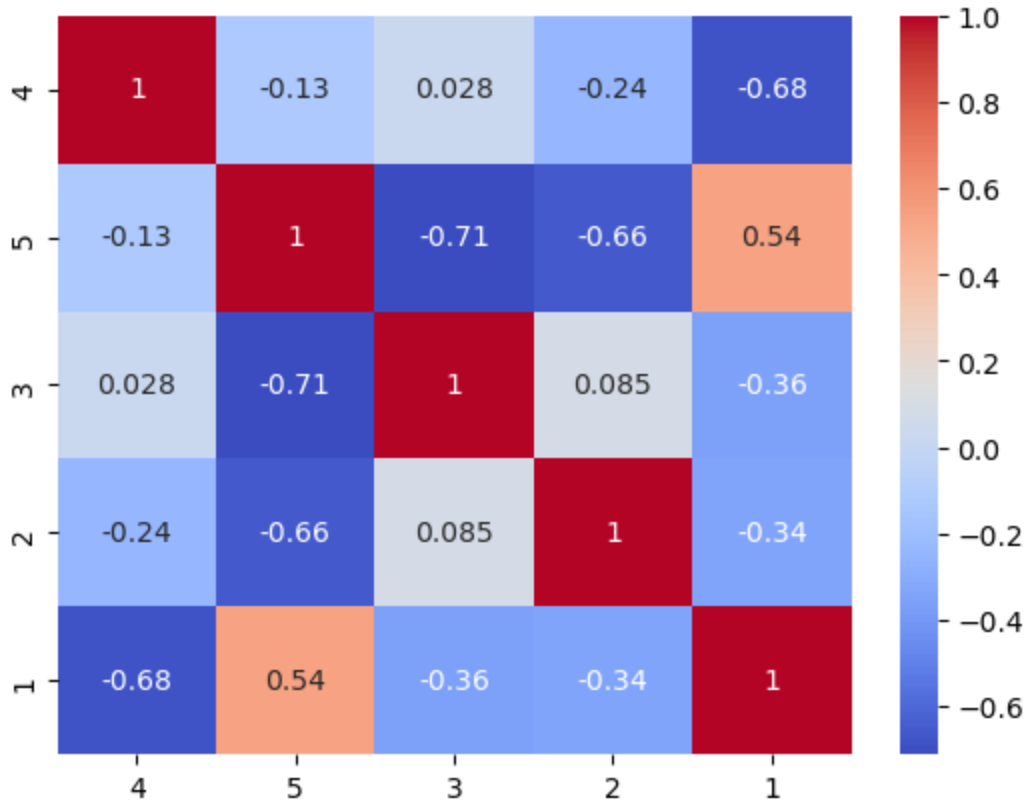
Correlation Coefficiency

In [19]:
```
triptype_df_corr=triptype_df_plot.corr()
```

In [20]:
```
sns.heatmap(triptype_df_corr, annot=True, cmap='coolwarm')
triptype_df_plot
```

Out[20]:

	4	5	3	2	1
Family	0.367056	0.276265	0.210117	0.081712	0.064851
Couples	0.395025	0.253731	0.226866	0.089552	0.034826
Business	0.381183	0.264791	0.194956	0.113482	0.045587
Solo travel	0.368504	0.283465	0.206299	0.092913	0.048819
Friends	0.362245	0.239796	0.229592	0.122449	0.045918



카이제곱검정

In [21]:
```
obs = pd.crosstab(df['Trip Type'], df['Rating'])
chi2, p, dof, expected = chi2_contingency(obs)
print(f"chi2 = {chi2}, p-value = {p}, degrees of freedom = {dof}\n")
```

```
chi2 = 52.38769499780465, p-value = 9.482428562239382e-06, degrees of freedom = 16
```

Rating과 Trip Type 관의 상관관계가 보여지는데 이 경우에는 특정 케이스들에 대해서 상관관계가 존재한다고 보았는데, 중간값에 해당하는 3을 기준으로 보았을 때, 상대적으로 중간값에 해당하는 1,2와의 상관관계는 비교적 약하게 나타났고 양끝에 있는 1과 5에 대해서 상관관계가 비교적 높게 나타났습니다. 그리고 흥미로웠던 점은, 양끝값을 두고 살펴 보았을때, 평점 1점과 5점을 보았을 때 평점들과의 상관관계가 비교적 다 높게 나왔다는 점을 확인해볼 수 있었습니다. 분석을 진행하기 전에 생각했었던 부분은 여행의 특정 유형에 따라 평점을 특히 높게 준다던지, 특히 낮게 준다던지의 편향성이 두드려져 강한 상관관계를 보이지 않을까 하는 생각을 했었는데, 특정 유형에 따라는 아니고 전반적인 데이터에서 평점을 높게 준다던지, 낮게 준다던지와 같은 패턴이 있다는 점에서 흥미로웠습니다.

In [22]:
```
!jupyter nbconvert --to webpdf --allow-chromium-download 20180374.ipynb
```

```
[NbConvertApp] Converting notebook 20180374.ipynb to webpdf
[NbConvertApp] Building PDF
[NbConvertApp] PDF successfully created
[NbConvertApp] Writing 648256 bytes to 20180374.pdf
```

In [ ]:
```
```