1.3) 국가별 음주 데이터 분석하기

In [8]:

```
# 주피터 노트북을 실행한 브라우저 내부(inline)에서 matplotlib으로 바로 그림을 볼 수 있

#%matplotlib inline

# 라이브러리 임포트
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

file_path = 'drinks.csv'
drinks = pd.read_csv(file_path)
```

<Step1. 탐색> 데이터의 기초 정보 살펴보기

In [9]:

```
print(drinks.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 193 entries, 0 to 192
Data columns (total 6 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   country                       193 non-null    object 
 1   beer_servings                 193 non-null    int64  
 2   spirit_servings               193 non-null    int64  
 3   wine_servings                 193 non-null    int64  
 4   total_litres_of_pure_alcohol  193 non-null    float64
 5   continent                     170 non-null    object 
dtypes: float64(1), int64(3), object(2)
memory usage: 9.2+ KB
None
```

In [10]:

```
drinks
```

Out[10]:

	country	beer_servings	spirit_servings	wine_servings	total_litres_of_pure_alcohol	continent
0	Afghanistan	0	0	0	0.0	AS
1	Albania	89	132	54	4.9	EU
2	Algeria	25	0	14	0.7	AF
3	Andorra	245	138	312	12.4	EU
4	Angola	217	57	45	5.9	AF
...	...	...	...	...	...	...
188	Venezuela	333	100	3	7.7	SA
189	Vietnam	111	2	1	2.0	AS
190	Yemen	6	0	0	0.1	AS
191	Zambia	32	19	4	2.5	AF
192	Zimbabwe	64	18	4	4.7	AF

193 rows × 6 columns

In [11]:

```
# 상위 10개 데이터 출력
drinks.head(10)
```

Out[11]:

	country	beer_servings	spirit_servings	wine_servings	total_litres_of_pure_alcohol	continent
0	Afghanistan	0	0	0	0.0	AS
1	Albania	89	132	54	4.9	EU
2	Algeria	25	0	14	0.7	AF
3	Andorra	245	138	312	12.4	EU
4	Angola	217	57	45	5.9	AF
5	Antigua & Barbuda	102	128	45	4.9	NaN
6	Argentina	193	25	221	8.3	SA
7	Armenia	21	179	11	3.8	EU
8	Australia	261	72	212	10.4	OC
9	Austria	279	75	191	9.7	EU

In [12]:

```
# 수치형 피처의 각 컬럼별 요약 통계(갯수, 평균, 표준편차, 최솟값, 4분위수, 최댓값) 지
drinks.describe()
```

Out[12]:

	beer_servings	spirit_servings	wine_servings	total_litres_of_pure_alcohol
count	193.000000	193.000000	193.000000	193.000000
mean	106.160622	80.994819	49.450777	4.717098
std	101.143103	88.284312	79.697598	3.773298
min	0.000000	0.000000	0.000000	0.000000
25%	20.000000	4.000000	1.000000	1.300000
50%	76.000000	56.000000	8.000000	4.200000
75%	188.000000	128.000000	59.000000	7.200000
max	376.000000	438.000000	370.000000	14.400000

<Step2. 인사이트 도출> 탐색과 시각화

[피처간의 상관관계 탐색]

[두 피처간의 상관계수 구하기] 여러 개의 피처들의 서로간의 상관계수를 구하기 앞서, 두 피처간의 상관계수를 구하는 실습

- 상관 분석 : 두 변수 간의 선형적 관계를 상관 계수로 표현하는 것
- 상관 계수 : 두 변수 사이의 통계적 관계를 표현하기 위해 특정한 상관 관계의 정도를 수치적으로 나타낸 계수
- 상관 계수를 구하는 것은 공분산의 개념을 포함 여러 유형의 상관계수가 존재하지만 제각기 자신들만의 정의와 특징이 있음

- 값의 범위는 -1에서 +1 사이에 속하며 여기서 ±1은 정도가 가장 센 잠재적 일치를 나타내고 0은 정도가 가장 센 불일치를 나타냄
- 1에 가까울 수록 서로 강한 양의 상관 관계가 있다는 것

- 피어슨 상관 계수(Pearson Correlation Coefficient ,PCC): 두 변수 X 와 Y 간의 선형 상관 관계를 계량화한 수치 스

- 스피어먼 상관 계수(Spearman Correlation Coefficient ,SCC):두 변수의 순위 사이의 통계적 의존성을 측정하는 비모수적인 척도

<피처간의 상관 관계를 통계적으로 탐색하는 방법>

- 단순 상관 분석 방법: 피처가 2 개일 때 상관 계수를 계산하는 방법
- 다중 상관 분석 방법: 피처가 여러 개일 때 상호간의 연관성을 분석하는 방법

[여러 피처의 상관관계 분석]

In [13]:

```
# 맥주와 와인 소비량의 상관 관계 알아보기
# 'beer_servings', 'wine_servings' 두 피처간의 상관계수를 계산합니다.
# 피어슨 상관 계수(Pearson Correlation Coefficient ,PCC): 두 변수 X 와 Y 간의 선형 상
# corr() 함수로 피처 간의 상관 계수를 매트릭스(matrix, 행렬, 숫자·기호 등을 가로, 세
# 단순 상관 분석 방법
# corr = drinks[['beer_servings', 'wine_servings']].corr(method = 'pearson')
# print(corr)
# 다중 상관 분석 방법
#corr = drinks.corr(method = 'pearson')
corr = drinks[['beer_servings', 'spirit_servings', 'wine_servings', 'total_litres_of_p
print(corr)
```

```
                              beer_servings  spirit_servings  wine_servings  ₩
beer_servings                      1.000000         0.458819       0.527172
spirit_servings                    0.458819         1.000000       0.194797
wine_servings                      0.527172         0.194797       1.000000
total_litres_of_pure_alcohol       0.835839         0.654968       0.667598

                              total_litres_of_pure_alcohol
beer_servings                                     0.835839
spirit_servings                                   0.654968
wine_servings                                     0.667598
total_litres_of_pure_alcohol                      1.000000
```

[seaborn 시각화 라이브러리 활용]

- 히트맵(heatmap), 페어플롯(pairplot) 기법 사용하기

- 히트맵(heatmap) : 히트(heat)와 지도를 뜻하는 맵(map)을 결합시킨 단어로 색상으로 표현할 수 있는 다양한 정보를 일정한 이미지 위에 열분포 형태의 그래픽으로 출력

In [14]:

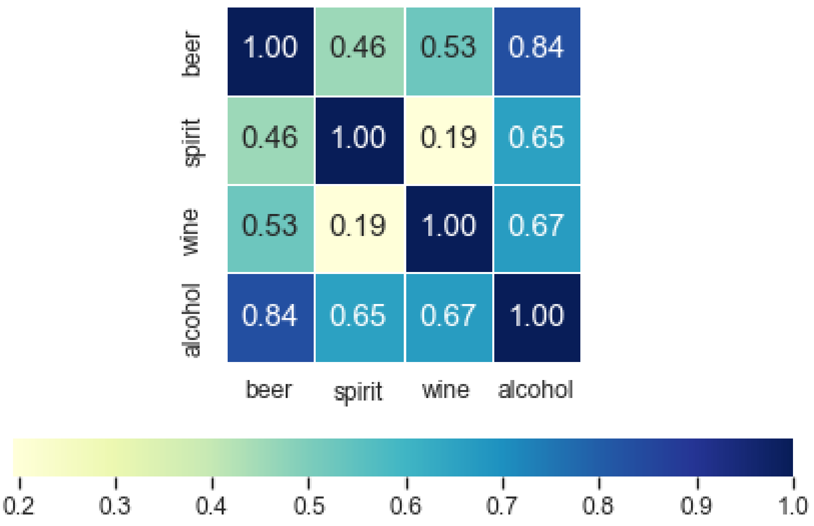
```
!pip install seaborn
```

```
Requirement already satisfied: seaborn in c:₩users₩yj₩anaconda3₩lib₩site-packages (0.1
1.2)
Requirement already satisfied: numpy>=1.15 in c:₩users₩yj₩anaconda3₩lib₩site-packages
(from seaborn) (1.20.3)
Requirement already satisfied: matplotlib>=2.2 in c:₩users₩yj₩anaconda3₩lib₩site-packa
ges (from seaborn) (3.4.3)
Requirement already satisfied: scipy>=1.0 in c:₩users₩yj₩anaconda3₩lib₩site-packages
(from seaborn) (1.7.1)
```

```
Requirement already satisfied: pandas>=0.23 in c:\users\yj\anaconda3\lib\site-packages
(from seaborn) (1.3.4)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\yj\anaconda3\lib\site-
packages (from matplotlib>=2.2->seaborn) (2.8.2)
Requirement already satisfied: pillow>=6.2.0 in c:\users\yj\anaconda3\lib\site-package
s (from matplotlib>=2.2->seaborn) (8.4.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\yj\anaconda3\lib\site-pac
kages (from matplotlib>=2.2->seaborn) (1.3.1)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\yj\anaconda3\lib\site-pack
ages (from matplotlib>=2.2->seaborn) (3.0.4)
Requirement already satisfied: cycler>=0.10 in c:\users\yj\anaconda3\lib\site-packages
(from matplotlib>=2.2->seaborn) (0.10.0)
Requirement already satisfied: six in c:\users\yj\anaconda3\lib\site-packages (from cy
cler>=0.10->matplotlib>=2.2->seaborn) (1.16.0)
Requirement already satisfied: pytz>=2017.3 in c:\users\yj\anaconda3\lib\site-packages
(from pandas>=0.23->seaborn) (2021.3)
```

In [15]:

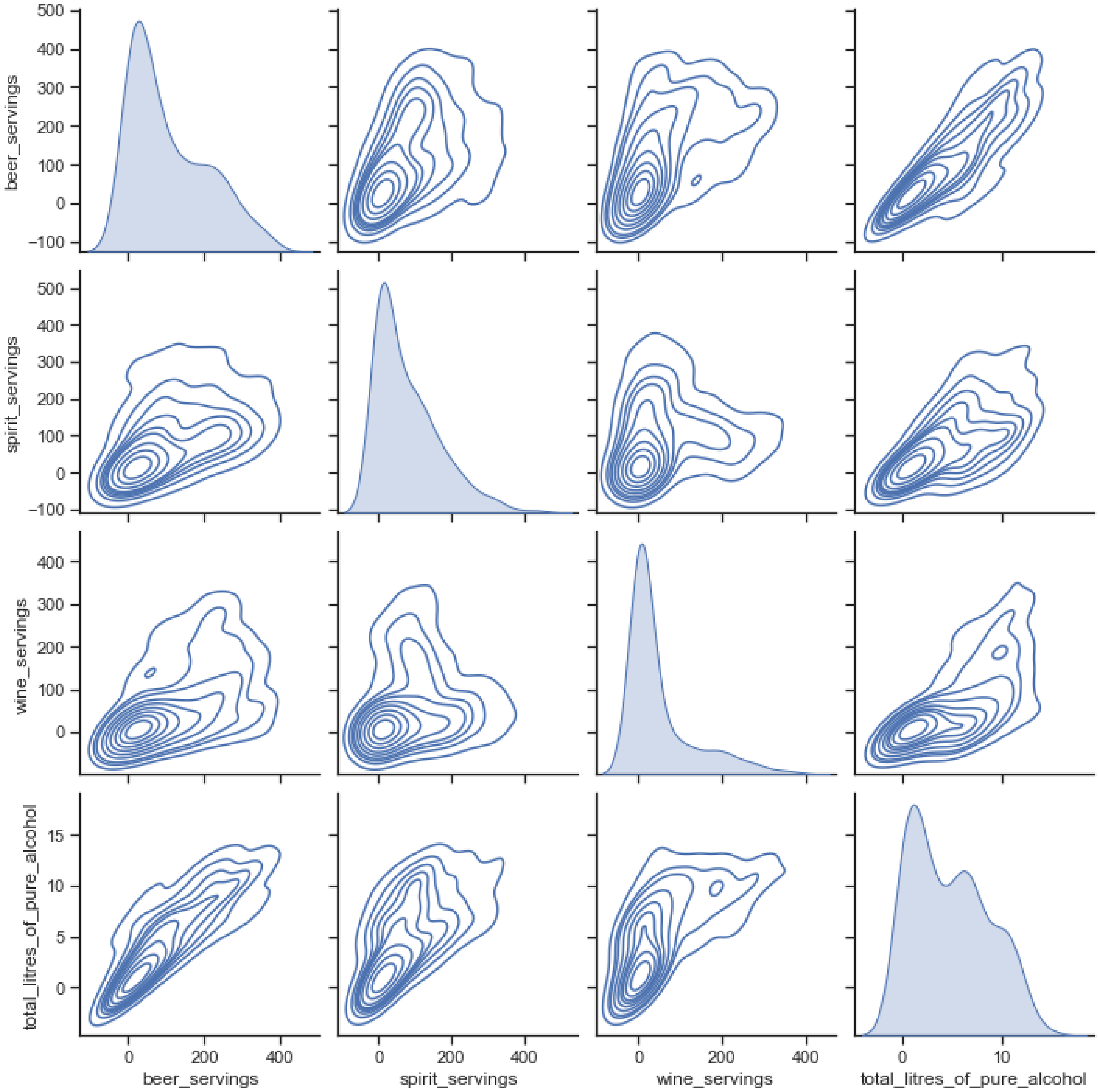
```python
import seaborn as sns

# corr 행렬 히트맵(heatmap)을 시각화합니다.
# 히트맵(heatmap)을 통해 corr.values 를 파라미터로 넣어줌
cols_view = ['beer', 'spirit', 'wine', 'alcohol'] # 그래프 출력을 위한 cols 이름을 축
sns.set(font_scale=1.1)
#cbar 히트맵 바 출력 여부, annot 상관 계수 출력 여부, square 사각형의 형태, fmt 소수점
# annot_kws 상관 계수 폰트 크기, yticklabels y 레이블명, xticklabels x 레이블명
hm = sns.heatmap(corr.values,
            cbar=True,
            annot=True,
            square=True,
            fmt='.2f',
            annot_kws={'size': 15},
            yticklabels=cols_view,
            xticklabels=cols_view,
            linewidths=.5,
            cmap="YlGnBu",
            cbar_kws={"orientation": "horizontal"})

plt.tight_layout()
plt.show()
```



In [16]:

```python
# 시각화 라이브러리를 이용한 피처간의 scatter plot을 출력합니다.
# 페어플롯(pairplot)은 데이터 프레임을 파타미터로 넣어 줌
# style : darkgrid, whitegrid, dark, white, ticks, 이미지의 전반적인 모양을 스타일링,
# context : notebook, paper, talk, poster,  어떤 상황에서 보여줄 것인가에 따라 4종류의
sns.set(style='ticks', context='notebook')
```

```
# sns.pairplot(drinks[['beer_servings', 'spirit_servings',
#                      'wine_servings', 'total_litres_of_pure_alcohol']], height=2.5)
# kind="hist", kind="kde"
# corner=True, Set corner=True to plot only the lower triangle:
sns.pairplot(drinks[['beer_servings', 'spirit_servings',
                     'wine_servings', 'total_litres_of_pure_alcohol']], height=2.5, k
plt.show()
```



In [17]:

```
# 1. 인사이트 도출1 : 대륙별 평균 wine_servings 탐색
drinks.groupby('continent').mean()
```

Out[17]:

	beer_servings	spirit_servings	wine_servings	total_litres_of_pure_alcohol
continent				
AF	61.471698	16.339623	16.264151	3.007547
AS	37.045455	60.840909	9.068182	2.170455
EU	193.777778	132.555556	142.222222	8.617778
OC	89.687500	58.437500	35.625000	3.381250
SA	175.083333	114.750000	62.416667	6.308333

In [18]:

```
# 대륙별  wine_servings 평균 1
drinks.groupby('continent').wine_servings.mean()
```

Out[18]:

```
continent
AF     16.264151
AS      9.068182
EU    142.222222
OC     35.625000
SA     62.416667
Name: wine_servings, dtype: float64
```

In [19]:

```
# 대륙별  wine_servings 평균2
drinks.groupby('continent')['wine_servings'].mean()
```

Out[19]:

```
continent
AF     16.264151
AS      9.068182
EU    142.222222
OC     35.625000
SA     62.416667
Name: wine_servings, dtype: float64
```

In [20]:

```
# 대륙별  wine_servings 평균3
drinks.groupby('continent').mean()['wine_servings']
```

Out[20]:

```
continent
AF     16.264151
AS      9.068182
EU    142.222222
OC     35.625000
SA     62.416667
Name: wine_servings, dtype: float64
```

In [21]:

```
# 2. 인사이트 도출2 : 전체 평균보다 적은 알코올을 섭취하는 대륙
total_mean = drinks.total_litres_of_pure_alcohol.mean()
total_mean
```

Out[21]:

```
4.717098445595855
```

In [22]:

```
# 전체 평균보다 적은 알코올을 섭취하는 대륙의 인덱스와 값 추출
continent_mean = drinks.groupby('continent').total_litres_of_pure_alcohol.mean()
continent_under_mean = continent_mean[continent_mean <= total_mean]
print(continent_under_mean)
```

```
continent
AF    3.007547
AS    2.170455
OC    3.381250
Name: total_litres_of_pure_alcohol, dtype: float64
```

In [23]:

```
# 전체 평균보다 적은 알코올을 섭취하는 대륙의 인덱스만 추출
continent_under_mean = continent_mean[continent_mean <= total_mean].index.tolist()
print(continent_under_mean)
```

```
['AF', 'AS', 'OC']
```

In [24]:

```
# 전체 평균보다 적은 알코올을 섭취하는 대륙의 인덱스만 추출
continent_under_mean = continent_mean[continent_mean <= total_mean].values.tolist()
print(continent_under_mean)
```

```
[3.0075471698113208, 2.1704545454545454, 3.38125]
```

In [25]:

```
# 대륙별 알코올 섭취 평균1
drinks.groupby('continent').total_litres_of_pure_alcohol.mean()
```

Out[25]:

```
continent
AF    3.007547
AS    2.170455
EU    8.617778
OC    3.381250
SA    6.308333
Name: total_litres_of_pure_alcohol, dtype: float64
```

In [26]:

```
# 대륙별 알코올 섭취 평균2
drinks.groupby('continent')['total_litres_of_pure_alcohol'].mean()
```

Out[26]:

```
continent
AF    3.007547
AS    2.170455
EU    8.617778
OC    3.381250
SA    6.308333
Name: total_litres_of_pure_alcohol, dtype: float64
```

In [27]:

```
# 대륙별 알코올 섭취 평균3
drinks.groupby('continent').mean()['total_litres_of_pure_alcohol']
```

Out[27]:

```
continent
AF    3.007547
AS    2.170455
EU    8.617778
OC    3.381250
SA    6.308333
Name: total_litres_of_pure_alcohol, dtype: float64
```

<Step3. 탐색적 분석>

[결측 데이터 전처리]

In [28]:

```
print(drinks.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 193 entries, 0 to 192
Data columns (total 6 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   country                       193 non-null    object 
 1   beer_servings                 193 non-null    int64  
 2   spirit_servings               193 non-null    int64  
 3   wine_servings                 193 non-null    int64  
 4   total_litres_of_pure_alcohol  193 non-null    float64
 5   continent                     170 non-null    object 
dtypes: float64(1), int64(3), object(2)
memory usage: 9.2+ KB
None
```

In [29]:

```
# 결측데이터 처리: 기타 대륙으로 통합 -> 'OT' (Others)
# fillna() 함수는 피처의 결측값을 특정 값으로 채워주는 함수
drinks['continent'] = drinks['continent'].fillna('OT')
```

In [30]:

```
print(drinks.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 193 entries, 0 to 192
Data columns (total 6 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   country                       193 non-null    object 
 1   beer_servings                 193 non-null    int64  
 2   spirit_servings               193 non-null    int64  
 3   wine_servings                 193 non-null    int64  
 4   total_litres_of_pure_alcohol  193 non-null    float64
 5   continent                     193 non-null    object 
dtypes: float64(1), int64(3), object(2)
memory usage: 9.2+ KB
None
```

In [31]:

```
drinks.head(10)
```

Out[31]:

	country	beer_servings	spirit_servings	wine_servings	total_litres_of_pure_alcohol	continent
0	Afghanistan	0	0	0	0.0	AS
1	Albania	89	132	54	4.9	EU
2	Algeria	25	0	14	0.7	AF
3	Andorra	245	138	312	12.4	EU
4	Angola	217	57	45	5.9	AF
5	Antigua & Barbuda	102	128	45	4.9	OT
6	Argentina	193	25	221	8.3	SA
7	Armenia	21	179	11	3.8	EU
8	Australia	261	72	212	10.4	OC
9	Austria	279	75	191	9.7	EU

In [32]:

```
drinks['continent'].value_counts()
```

Out[32]:

```
AF    53
EU    45
AS    44
OT    23
OC    16
SA    12
Name: continent, dtype: int64
```

In [33]:

```
drinks['continent'].value_counts().index.tolist()
```

Out[33]:

```
['AF', 'EU', 'AS', 'OT', 'OC', 'SA']
```

In [34]:

```
drinks['continent'].value_counts().values.tolist()
```

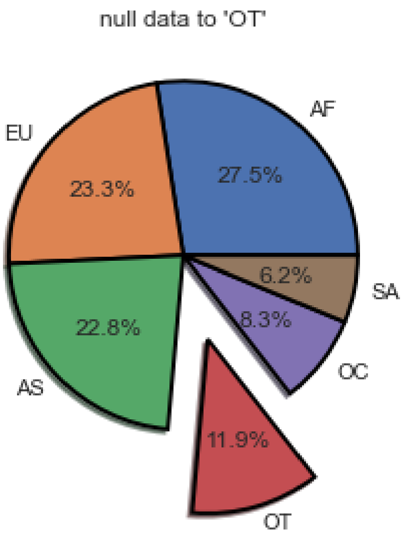
Out[34]:

```
[53, 45, 44, 23, 16, 12]
```

전체 대륙 중에서 OT 가 차지하는 비율을 파이차트로 확인

In [35]:

```
# 그래프를 표시하기 위한 인덱스와 값을 추출해서 리스트로 생성
labels = drinks['continent'].value_counts().index.tolist()
fracs1 = drinks['continent'].value_counts().values.tolist()
explode = (0, 0, 0, 0.5, 0, 0)

plt.pie(fracs1, labels=labels, autopct='%.1f%%', shadow=True, explode=explode, radiu
plt.title('null data to ₩'OT₩'')
plt.show()
```



[그룹 단위의 데이터 분석 : 대륙별 분석]

apply, agg 함수를 이용한 대륙별 분석

1. 대륙별 spirit_servings의 평균, 최소, 최대, 합계를 계산

In [36]:

```
drinks.groupby('continent').spirit_servings.describe()
```

Out[36]:

continent	count	mean	std	min	25%	50%	75%	max
AF	53.0	16.339623	28.102794	0.0	1.00	3.0	19.00	152.0
AS	44.0	60.840909	84.362160	0.0	1.00	16.0	98.00	326.0
EU	45.0	132.555556	77.589115	0.0	81.00	122.0	173.00	373.0
OC	16.0	58.437500	70.504817	0.0	18.00	37.0	65.25	254.0
OT	23.0	165.739130	94.993884	68.0	101.00	137.0	190.50	438.0
SA	12.0	114.750000	77.077440	25.0	65.75	108.5	148.75	302.0

- agg() 함수는 apply() 함수와 거의 동일하게 함수 파라미터를 받음
- agg() 함수는 함수 파라미터를 병렬로 설정하여 그룹에 대한 여러가지 연산 결과를 동시에 얻을 수 있는 함수
- agg() 함수를 이용해 'mean', 'min', 'max', 'sum' 함수 파라미터를 간단히 탐색

In [37]:

```
drinks.groupby('continent').spirit_servings.agg(['mean', 'min', 'max', 'sum'])
```

Out[37]:

	mean	min	max	sum

continent	mean	min	max	sum
continent				
AF	16.339623	0	152	866
AS	60.840909	0	326	2677
EU	132.555556	0	373	5965
OC	58.437500	0	254	935
OT	165.739130	68	438	3812
SA	114.750000	25	302	1377

1. 전체 평균보다 많은 알코올을 섭취하는 대륙

2-1: 전체 알코올 섭취 평균 2-2: 대륙별 알코올 섭취 평균 2-3: 대륙별 알코올 섭취 평균이 전체 알코올 섭취 평균보다 낮은 대륙을 구하기

In [38]:

```
# 2-1: 전체 알코올 섭취 평균
total_mean = drinks.total_litres_of_pure_alcohol.mean()
# 2-2: 대륙별 알코올 섭취 평균
#drinks.groupby('continent')['total_litres_of_pure_alcohol'].mean()
continent_mean = drinks.groupby('continent').total_litres_of_pure_alcohol.mean()
# 데이터프레임[조건]
continent_mean[continent_mean < total_mean]
```

Out[38]:

```
continent
AF    3.007547
AS    2.170455
OC    3.381250
Name: total_litres_of_pure_alcohol, dtype: float64
```

1. 평균 beer_servings이 가장 높은 대륙

- idxmin, idxmax 는 전체 인덱스 중 최소값, 최대값을 반환
- 3-1: 대륙별 평균 beer_servings 계산
- 3-2: 결과 중 값이 가장 높은 인덱스만 추출

In [39]:

```
# 평균 beer_servings이 가장 높은 대륙의 인덱스와 값
drinks.groupby('continent').beer_servings.mean()
```

Out[39]:

```
continent
AF     61.471698
AS     37.045455
EU    193.777778
OC     89.687500
OT    145.434783
SA    175.083333
Name: beer_servings, dtype: float64
```

In [40]:

```
# 평균 beer_servings이 가장 높은 대륙의 인덱스
drinks.groupby('continent').beer_servings.mean().idxmax()
```

Out[40]:

```
'EU'
```

In [41]:

```
# 평균 beer_servings이 가장 높은 대륙의 값
drinks.groupby('continent').beer_servings.mean().max()
```

Out[41]:

```
193.77777777777777
```

In [42]:

```
# 평균 beer_servings이 가장 낮은 대륙의 인덱스
drinks.groupby('continent').beer_servings.mean().idxmin()
```

Out[42]:

```
'AS'
```

In [43]:

```
# 평균 beer_servings이 가장 낮은 대륙의 값
drinks.groupby('continent').beer_servings.mean().min()
```

Out[43]:

```
37.04545454545455
```

1. 대륙별 spirit_servings의 평균, 최소, 최대, 합계를 시각화합니다.

In [44]:

```
result = drinks.groupby('continent').spirit_servings.agg(['mean', 'min', 'max', 'sum'
result
```

Out[44]:

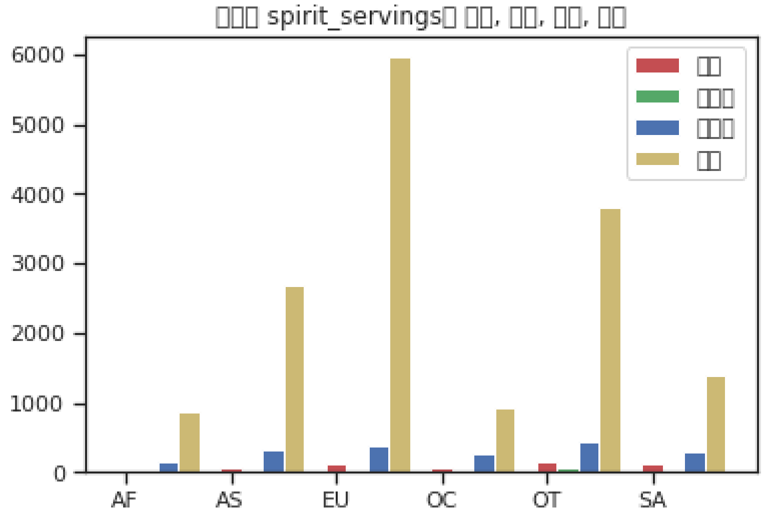
	mean	min	max	sum
continent				
AF	16.339623	0	152	866
AS	60.840909	0	326	2677
EU	132.555556	0	373	5965
OC	58.437500	0	254	935
OT	165.739130	68	438	3812
SA	114.750000	25	302	1377

In [45]:

```
# 그래프에 한글 표시
plt.rc('font', family='NanumGothic')

n_groups = len(result.index)
means = result['mean'].tolist()
mins = result['min'].tolist()
maxs = result['max'].tolist()
sums = result['sum'].tolist()

index = np.arange(n_groups) # n_groups의 크기 6만큼 배열 생성하여 index에 저장
bar_width = 0.2

# 평균 값에 대한 그래프 생성
rects1 = plt.bar(index, means, bar_width, color='r',label='평균')
rects2 = plt.bar(index + bar_width, mins, bar_width, color='g',label='최소값')
rects3 = plt.bar(index + bar_width * 2, maxs, bar_width, color='b',label='최대값')
rects4 = plt.bar(index + bar_width * 3, sums, bar_width, color='y',label='합계')

plt.title('대륙별 spirit_servings의 평균, 최소, 최대, 합계')
plt.xticks(index, result.index.tolist())
plt.legend() # 그래프에 범례를 달고 싶을 때 쓰는 함수
plt.show()
```

```
findfont: Font family ['NanumGothic'] not found. Falling back to DejaVu Sans.
C:₩Users₩YJ₩anaconda3₩lib₩site-packages₩matplotlib₩backends₩backend_agg.py:240: Runtim
eWarning: Glyph 45824 missing from current font.
  font.set_text(s, 0.0, flags=flags)
C:₩Users₩YJ₩anaconda3₩lib₩site-packages₩matplotlib₩backends₩backend_agg.py:240: Runtim
eWarning: Glyph 47449 missing from current font.
  font.set_text(s, 0.0, flags=flags)
C:₩Users₩YJ₩anaconda3₩lib₩site-packages₩matplotlib₩backends₩backend_agg.py:240: Runtim
eWarning: Glyph 48324 missing from current font.
  font.set_text(s, 0.0, flags=flags)
C:₩Users₩YJ₩anaconda3₩lib₩site-packages₩matplotlib₩backends₩backend_agg.py:240: Runtim
eWarning: Glyph 51032 missing from current font.
  font.set_text(s, 0.0, flags=flags)
C:₩Users₩YJ₩anaconda3₩lib₩site-packages₩matplotlib₩backends₩backend_agg.py:240: Runtim
eWarning: Glyph 54217 missing from current font.
  font.set_text(s, 0.0, flags=flags)
C:₩Users₩YJ₩anaconda3₩lib₩site-packages₩matplotlib₩backends₩backend_agg.py:240: Runtim
eWarning: Glyph 44512 missing from current font.
  font.set_text(s, 0.0, flags=flags)
C:₩Users₩YJ₩anaconda3₩lib₩site-packages₩matplotlib₩backends₩backend_agg.py:240: Runtim
eWarning: Glyph 52572 missing from current font.
  font.set_text(s, 0.0, flags=flags)
C:₩Users₩YJ₩anaconda3₩lib₩site-packages₩matplotlib₩backends₩backend_agg.py:240: Runtim
eWarning: Glyph 49548 missing from current font.
  font.set_text(s, 0.0, flags=flags)
C:₩Users₩YJ₩anaconda3₩lib₩site-packages₩matplotlib₩backends₩backend_agg.py:240: Runtim
eWarning: Glyph 54633 missing from current font.
  font.set_text(s, 0.0, flags=flags)
C:₩Users₩YJ₩anaconda3₩lib₩site-packages₩matplotlib₩backends₩backend_agg.py:240: Runtim
eWarning: Glyph 44228 missing from current font.
  font.set_text(s, 0.0, flags=flags)
findfont: Font family ['NanumGothic'] not found. Falling back to DejaVu Sans.
C:₩Users₩YJ₩anaconda3₩lib₩site-packages₩matplotlib₩backends₩backend_agg.py:240: Runtim
eWarning: Glyph 44050 missing from current font.
  font.set_text(s, 0.0, flags=flags)
C:₩Users₩YJ₩anaconda3₩lib₩site-packages₩matplotlib₩backends₩backend_agg.py:203: Runtim
eWarning: Glyph 45824 missing from current font.
  font.set_text(s, 0, flags=flags)
C:₩Users₩YJ₩anaconda3₩lib₩site-packages₩matplotlib₩backends₩backend_agg.py:203: Runtim
eWarning: Glyph 47449 missing from current font.
  font.set_text(s, 0, flags=flags)
C:₩Users₩YJ₩anaconda3₩lib₩site-packages₩matplotlib₩backends₩backend_agg.py:203: Runtim
eWarning: Glyph 48324 missing from current font.
  font.set_text(s, 0, flags=flags)
C:₩Users₩YJ₩anaconda3₩lib₩site-packages₩matplotlib₩backends₩backend_agg.py:203: Runtim
eWarning: Glyph 51032 missing from current font.
  font.set_text(s, 0, flags=flags)
C:₩Users₩YJ₩anaconda3₩lib₩site-packages₩matplotlib₩backends₩backend_agg.py:203: Runtim
eWarning: Glyph 54217 missing from current font.
  font.set_text(s, 0, flags=flags)
C:₩Users₩YJ₩anaconda3₩lib₩site-packages₩matplotlib₩backends₩backend_agg.py:203: Runtim
eWarning: Glyph 44512 missing from current font.
  font.set_text(s, 0, flags=flags)
C:₩Users₩YJ₩anaconda3₩lib₩site-packages₩matplotlib₩backends₩backend_agg.py:203: Runtim
eWarning: Glyph 52572 missing from current font.
  font.set_text(s, 0, flags=flags)
C:₩Users₩YJ₩anaconda3₩lib₩site-packages₩matplotlib₩backends₩backend_agg.py:203: Runtim
eWarning: Glyph 49548 missing from current font.
  font.set_text(s, 0, flags=flags)
C:₩Users₩YJ₩anaconda3₩lib₩site-packages₩matplotlib₩backends₩backend_agg.py:203: Runtim
eWarning: Glyph 54633 missing from current font.
  font.set_text(s, 0, flags=flags)
C:₩Users₩YJ₩anaconda3₩lib₩site-packages₩matplotlib₩backends₩backend_agg.py:203: Runtim
eWarning: Glyph 44228 missing from current font.
```

```
  font.set_text(s, 0, flags=flags)
C:\Users\YJ\anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:203: Runtim
eWarning: Glyph 44050 missing from current font.
  font.set_text(s, 0, flags=flags)
```



In [46]:

```
drinks
```

Out[46]:

	country	beer_servings	spirit_servings	wine_servings	total_litres_of_pure_alcohol	continent
0	Afghanistan	0	0	0	0.0	AS
1	Albania	89	132	54	4.9	EU
2	Algeria	25	0	14	0.7	AF
3	Andorra	245	138	312	12.4	EU
4	Angola	217	57	45	5.9	AF
...	...	...	...	...	...	...
188	Venezuela	333	100	3	7.7	SA
189	Vietnam	111	2	1	2.0	AS
190	Yemen	6	0	0	0.1	AS
191	Zambia	32	19	4	2.5	AF
192	Zimbabwe	64	18	4	4.7	AF

193 rows × 6 columns

In [47]:

```
drinks.to_csv("drinks_bigdata.csv")
```

1. 대륙별 total_litres_of_pure_alcohol을 시각화

In [48]:

```
# 대륙별 total_litres_of_pure_alcohol의 평균
continent_mean = drinks.groupby('continent').total_litres_of_pure_alcohol.mean()
```

In [49]:

```
continent_mean
```

```
continent
```

Out[49]:
```
AF    3.007547
AS    2.170455
EU    8.617778
OC    3.381250
OT    5.995652
SA    6.308333
Name: total_litres_of_pure_alcohol, dtype: float64
```

In [50]:
```
continent_mean.index
```

Out[50]:
```
Index(['AF', 'AS', 'EU', 'OC', 'OT', 'SA'], dtype='object', name='continent')
```

In [51]:
```
continent_mean.values
```

Out[51]:
```
array([3.00754717, 2.17045455, 8.61777778, 3.38125   , 5.99565217,
       6.30833333])
```

In [52]:
```
continents = continent_mean.index.tolist()
```

In [53]:
```
continents.append('mean')
```

In [54]:
```
continents
```

Out[54]:
```
['AF', 'AS', 'EU', 'OC', 'OT', 'SA', 'mean']
```

In [55]:
```
total_mean= drinks.total_litres_of_pure_alcohol.mean()
total_mean
```

Out[55]:
```
4.717098445595855
```

In [56]:
```
#continent_mean.tolist()
alcohol=continent_mean.values.tolist()
```

In [57]:
```
# 대륙별 알코올 평균 리스트 마지막에 전체 평균을 추가
alcohol.append(total_mean)
alcohol
```

Out[57]:
```
[3.0075471698113208,
 2.1704545454545454,
 8.617777777777778,
 3.38125,
 5.995652173913044,
 6.308333333333334,
 4.717098445595855]
```

In [58]:
```
len(continents)
```

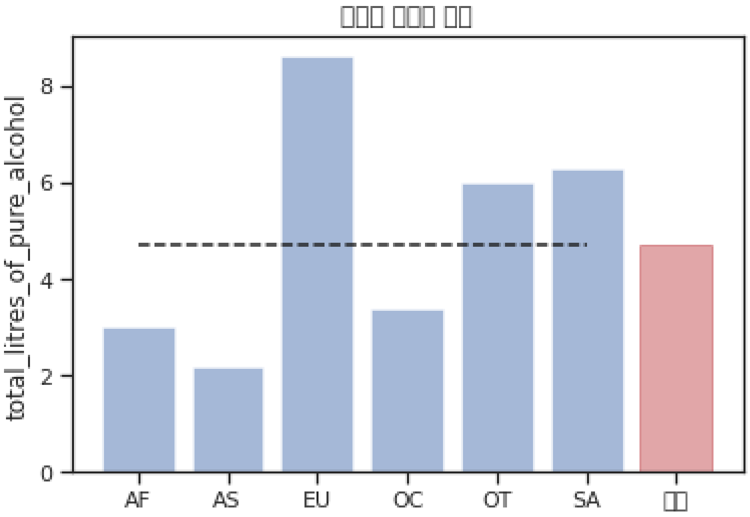
Out[58]:
```
7
```

In [59]:
```
# 대륙별 평균의 인덱스만 추출해서 continents 리스트 생성
continent_mean = drinks.groupby('continent').total_litres_of_pure_alcohol.mean()
continents = continent_mean.index.tolist()
```

```
continents.append('평균') # 전체 평균을 표시하기 위한 'mean' 인덱스 추가

# 대륙별 평균의 값만 추출해서 alcohol 리스트 생성
total_mean= drinks.total_litres_of_pure_alcohol.mean()
alcohol = continent_mean.tolist()
alcohol.append(total_mean) # 전체 평균을 표시하기 위한 total_mean의 값 추가

# 막대 그래프 출력
x_pos = np.arange(len(continents))

bar_list = plt.bar(x_pos, alcohol, align='center', alpha=0.5)
bar_list[len(continents) - 1].set_color('r')  # 마지막 바의 색상을 red로 지정

plt.plot([0, 5], [total_mean, total_mean], "k--")  # 평균 수치에 선 생성
plt.xticks(x_pos, continents)

plt.ylabel('total_litres_of_pure_alcohol')
plt.title('대륙별 알코올 평균')


plt.show()
```

```
C:₩Users₩YJ₩anaconda3₩lib₩site-packages₩matplotlib₩backends₩backend_agg.py:240: Runtim
eWarning: Glyph 45824 missing from current font.
  font.set_text(s, 0.0, flags=flags)
C:₩Users₩YJ₩anaconda3₩lib₩site-packages₩matplotlib₩backends₩backend_agg.py:240: Runtim
eWarning: Glyph 47449 missing from current font.
  font.set_text(s, 0.0, flags=flags)
C:₩Users₩YJ₩anaconda3₩lib₩site-packages₩matplotlib₩backends₩backend_agg.py:240: Runtim
eWarning: Glyph 48324 missing from current font.
  font.set_text(s, 0.0, flags=flags)
C:₩Users₩YJ₩anaconda3₩lib₩site-packages₩matplotlib₩backends₩backend_agg.py:240: Runtim
eWarning: Glyph 50508 missing from current font.
  font.set_text(s, 0.0, flags=flags)
C:₩Users₩YJ₩anaconda3₩lib₩site-packages₩matplotlib₩backends₩backend_agg.py:240: Runtim
eWarning: Glyph 53076 missing from current font.
  font.set_text(s, 0.0, flags=flags)
C:₩Users₩YJ₩anaconda3₩lib₩site-packages₩matplotlib₩backends₩backend_agg.py:240: Runtim
eWarning: Glyph 50732 missing from current font.
  font.set_text(s, 0.0, flags=flags)
C:₩Users₩YJ₩anaconda3₩lib₩site-packages₩matplotlib₩backends₩backend_agg.py:240: Runtim
eWarning: Glyph 54217 missing from current font.
  font.set_text(s, 0.0, flags=flags)
C:₩Users₩YJ₩anaconda3₩lib₩site-packages₩matplotlib₩backends₩backend_agg.py:240: Runtim
eWarning: Glyph 44512 missing from current font.
  font.set_text(s, 0.0, flags=flags)
C:₩Users₩YJ₩anaconda3₩lib₩site-packages₩matplotlib₩backends₩backend_agg.py:203: Runtim
eWarning: Glyph 54217 missing from current font.
  font.set_text(s, 0, flags=flags)
C:₩Users₩YJ₩anaconda3₩lib₩site-packages₩matplotlib₩backends₩backend_agg.py:203: Runtim
eWarning: Glyph 44512 missing from current font.
  font.set_text(s, 0, flags=flags)
C:₩Users₩YJ₩anaconda3₩lib₩site-packages₩matplotlib₩backends₩backend_agg.py:203: Runtim
eWarning: Glyph 45824 missing from current font.
  font.set_text(s, 0, flags=flags)
C:₩Users₩YJ₩anaconda3₩lib₩site-packages₩matplotlib₩backends₩backend_agg.py:203: Runtim
eWarning: Glyph 47449 missing from current font.
  font.set_text(s, 0, flags=flags)
C:₩Users₩YJ₩anaconda3₩lib₩site-packages₩matplotlib₩backends₩backend_agg.py:203: Runtim
eWarning: Glyph 48324 missing from current font.
  font.set_text(s, 0, flags=flags)
C:₩Users₩YJ₩anaconda3₩lib₩site-packages₩matplotlib₩backends₩backend_agg.py:203: Runtim
eWarning: Glyph 50508 missing from current font.
```

```
  font.set_text(s, 0, flags=flags)
C:\Users\YJ\anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:203: Runtim
eWarning: Glyph 53076 missing from current font.
  font.set_text(s, 0, flags=flags)
C:\Users\YJ\anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:203: Runtim
eWarning: Glyph 50732 missing from current font.
  font.set_text(s, 0, flags=flags)
```

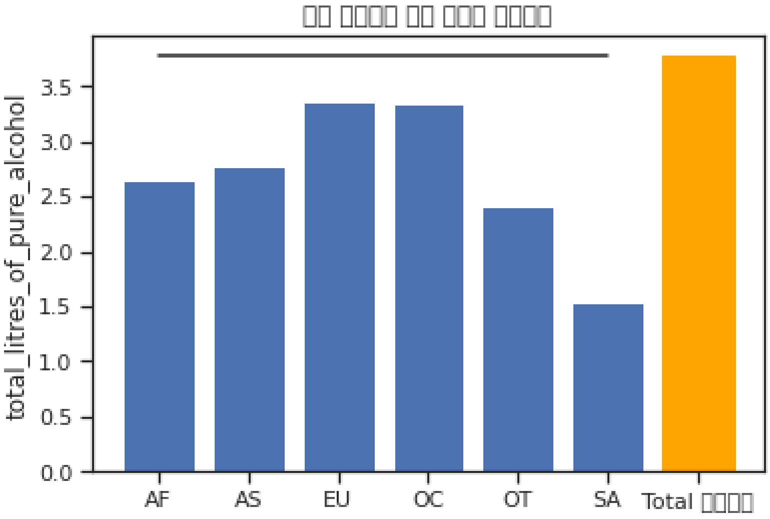


In [60]:

```
# 대륙별 total_litres_of_pure_alcohol 시각화 - 점선을 변경합니다.
# 점선의 위치도 변경해 봅니다.
# 전체 표준편차 대비 대륙별 표준편차 그래프 구하기
total_std = drinks.total_litres_of_pure_alcohol.std()
continent_std = drinks.groupby('continent')['total_litres_of_pure_alcohol'].std()
continents = continent_std.index.tolist()
continents.append('Total 표준편차')
x_pos = np.arange(len(continents))
alcohol = continent_std.tolist()
alcohol.append(total_std)

bar_list = plt.bar(x_pos, alcohol, align='center', alpha=1)
bar_list[len(continents) - 1].set_color('orange')
#plt.plot([3, 5], [total_max, total_max], "k--") # 점선 표기 부분
plt.plot([0, 5], [total_std, total_std], "k-") # 실선 표기
plt.xticks(x_pos, continents)

plt.ylabel('total_litres_of_pure_alcohol')
plt.title('전체 표준편차 대비 대륙별 표준편차')

plt.show()
```

```
C:\Users\YJ\anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:240: Runtim
eWarning: Glyph 51204 missing from current font.
  font.set_text(s, 0.0, flags=flags)
C:\Users\YJ\anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:240: Runtim
eWarning: Glyph 52404 missing from current font.
  font.set_text(s, 0.0, flags=flags)
C:\Users\YJ\anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:240: Runtim
eWarning: Glyph 54364 missing from current font.
  font.set_text(s, 0.0, flags=flags)
C:\Users\YJ\anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:240: Runtim
eWarning: Glyph 51456 missing from current font.
  font.set_text(s, 0.0, flags=flags)
C:\Users\YJ\anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:240: Runtim
eWarning: Glyph 54200 missing from current font.
  font.set_text(s, 0.0, flags=flags)
C:\Users\YJ\anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:240: Runtim
```

```
eWarning: Glyph 52264 missing from current font.
  font.set_text(s, 0.0, flags=flags)
C:\Users\YJ\anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:240: Runtim
eWarning: Glyph 48708 missing from current font.
  font.set_text(s, 0.0, flags=flags)
C:\Users\YJ\anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:203: Runtim
eWarning: Glyph 54364 missing from current font.
  font.set_text(s, 0, flags=flags)
C:\Users\YJ\anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:203: Runtim
eWarning: Glyph 51456 missing from current font.
  font.set_text(s, 0, flags=flags)
C:\Users\YJ\anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:203: Runtim
eWarning: Glyph 54200 missing from current font.
  font.set_text(s, 0, flags=flags)
C:\Users\YJ\anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:203: Runtim
eWarning: Glyph 52264 missing from current font.
  font.set_text(s, 0, flags=flags)
C:\Users\YJ\anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:203: Runtim
eWarning: Glyph 51204 missing from current font.
  font.set_text(s, 0, flags=flags)
C:\Users\YJ\anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:203: Runtim
eWarning: Glyph 52404 missing from current font.
  font.set_text(s, 0, flags=flags)
C:\Users\YJ\anaconda3\lib\site-packages\matplotlib\backends\backend_agg.py:203: Runtim
eWarning: Glyph 48708 missing from current font.
  font.set_text(s, 0, flags=flags)
```



In [65]:

```
!pip install scipy
```

```
Requirement already satisfied: scipy in c:\users\yj\anaconda3\lib\site-packages (1.7.
1)
Requirement already satisfied: numpy<1.23.0,>=1.16.5 in c:\users\yj\anaconda3\lib\site
-packages (from scipy) (1.20.3)
```

통계적 분석 분석 대상간의 통계적 차이 검정하기

In [64]:

```
africa = drinks.loc[drinks['continent']=='AF']
europe = drinks.loc[drinks['continent']=='EU']

from scipy import stats
tTestResult = stats.ttest_ind(africa['beer_servings'],europe['beer_servings'])
tTestResultDiffVar = stats.ttest_ind(africa['beer_servings'],europe['beer_servings'],
                                     equal_var = False)
```

```
print("%.3f and %.3f." % tTestResult)
print("%.3f and %.3f" % tTestResultDiffVar)
```

```
-7.268 and 0.000.
-7.144 and 0.000
```

대한민국은 술을 얼마나 독하게 마시는 나라인가 분석

In [77]:

```
drinks['total_servings']= drinks['beer_servings']+ drinks['wine_servings']+drinks['sp
drinks['total_servings']
```

Out[77]:

```
0        0
1      275
2       39
3      695
4      319
      ... 
188    436
189    114
190      6
191     55
192     86
Name: total_servings, Length: 193, dtype: int64
```

In [78]:

```
drinks['alcohol_rate'] = drinks['total_litres_of_pure_alcohol']/drinks['total_serving
drinks['alcohol_rate'] = drinks['alcohol_rate'].fillna(0)
drinks['alcohol_rate']
```

Out[78]:

```
0      0.000000
1      0.017818
2      0.017949
3      0.017842
4      0.018495
         ...   
188    0.017661
189    0.017544
190    0.016667
191    0.045455
192    0.054651
Name: alcohol_rate, Length: 193, dtype: float64
```

In [79]:

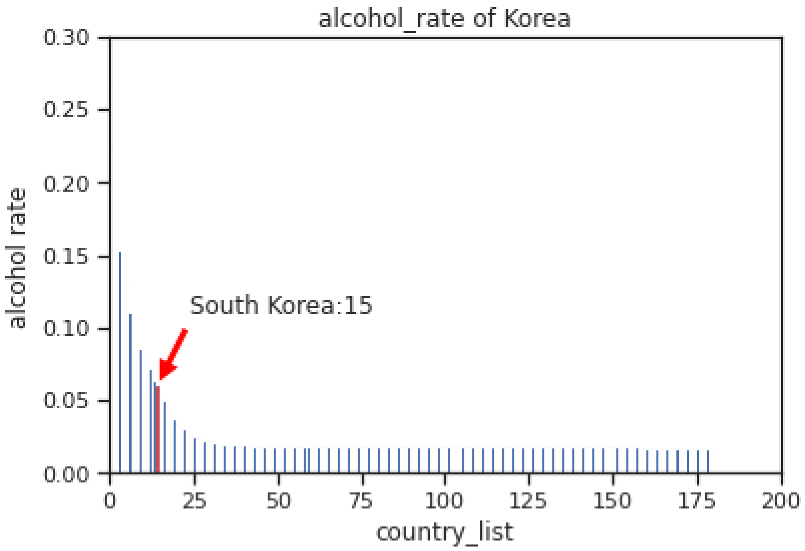
```
country_with_rank = drinks[['country','alcohol_rate']]
country_with_rank = country_with_rank.sort_values(by=['alcohol_rate'],ascending =0)
country_with_rank.head(5)
```

Out[79]:

	country	alcohol_rate
63	Gambia	0.266667
153	Sierra Leone	0.223333
124	Nigeria	0.185714
179	Uganda	0.153704
142	Rwanda	0.151111

In [116…

```
# 순위 정보 country_with_rank 중  country 피처로 리스트 생성
country_list = country_with_rank.country.tolist()
x_pos = np.arange(len(country_list)) # x축의 값 나라별 이름이 길어 숫자로 대처함
# 그래프 값, 순위 정보 중 alcohol_rate 피처로 리스트 생성
rank = country_with_rank.alcohol_rate.tolist()

bar_list =plt.bar(x_pos,rank) # x축 값, 그래프 값으로 막대 그래프 생성
bar_list[country_list.index("South Korea")].set_color("r") #한국 그래프 red 색상 적용
plt.ylabel('alcohol rate')
plt.xlabel('country_list')
plt.title('alcohol_rate of Korea')
plt.axis([0,200,0,0.3]) # x,y 축에 대한 옵션을 다루는 메소드 x축 0~200 y축 0~3 까지
korea_rank = country_list.index("South Korea")
# 한국의 알콜 소비량 추출, 나라별 순위 데이터 프레임에서 나라가 한국인 것을 알콜 레이
# 시리즈에서 값을 추출 하면 시리즈가 되기 때문에 값 하나만 추출 values[0]
# 화살표와 화살표 텍스트 스타일 지정
korea_alc_rate = country_with_rank[country_with_rank['country'] == 'South Korea']['al
plt.annotate('South Korea:' +str(korea_rank +1 ), # 텍스트 값
            xy=(korea_rank,korea_alc_rate),  # 화살표 머리 표시할 x,y 위치
             xytext = (korea_rank + 10, korea_alc_rate + 0.05), # 화살표 텍스트를 표
            arrowprops = dict(facecolor='red', shrink = 0.05)) # 화살표 색상

plt.show()
```



연습문제 1

In [94]:

```
result = drinks.groupby('continent').wine_servings.agg('mean')
result.head()
```

Out[94]:

```
continent
AF     16.264151
AS      9.068182
EU    142.222222
OC     35.625000
OT     24.521739
Name: wine_servings, dtype: float64
```

연습문제 2

In [99]:

```
total_servings = drinks[beer_servings.sum()+drinks.spirit_servings.sum()+drinks.wine_
total_servings
```

Out[99]:

```
45665
```

In [117…

```
total_servings=drinks.groupby('continent')['beer_servings'].sum() +drinks.groupby('co
total_servings
```

Out[117…

```
continent
AF     4986
AS     4706
EU    21085
OC     2940
OT     7721
SA     4227
dtype: int64
```

연습문제 3

In [128…

```
total_mean = total_servings.mean()
total_mean
```

Out[128…

```
7610.833333333333
```

In [149…

```
drinks.groupby('country')['spirit_servings'].sum().idxmax()
```

Out[149…

```
'Grenada'
```

In [ ]:

In [202…

```
continent_mean = total_servings
continent_down_mean = continent_mean[continent_mean<= total_men]
continent_list = continent_down_mean.index
continent_list
```

Out[202…

```
Index(['AF', 'AS', 'OC', 'SA'], dtype='object', name='continent')
```

In [217…

```
drinks['continent'].values
```

Out[217…

```
array(['AS', 'EU', 'AF', 'EU', 'AF', 'OT', 'SA', 'EU', 'OC', 'EU', 'EU',
       'OT', 'AS', 'AS', 'OT', 'EU', 'EU', 'OT', 'AF', 'AS', 'SA', 'EU',
       'AF', 'SA', 'AS', 'EU', 'AF', 'AF', 'AF', 'AF', 'AS', 'AF', 'OT',
       'AF', 'AF', 'SA', 'AS', 'SA', 'AF', 'AF', 'OC', 'OT', 'EU', 'OT',
       'EU', 'EU', 'AS', 'AF', 'EU', 'AF', 'OT', 'OT', 'SA', 'AF', 'OT',
       'AF', 'AF', 'EU', 'AF', 'OC', 'EU', 'EU', 'AF', 'AF', 'EU', 'EU',
       'AF', 'EU', 'OT', 'OT', 'AF', 'AF', 'SA', 'OT', 'OT', 'EU', 'EU',
       'AS', 'AS', 'AS', 'AS', 'EU', 'AS', 'EU', 'OT', 'AS', 'AS', 'AS',
       'AF', 'OC', 'AS', 'AS', 'AS', 'EU', 'AS', 'AF', 'AF', 'AF', 'EU',
       'EU', 'AF', 'AF', 'AS', 'AS', 'AF', 'EU', 'OC', 'AF', 'AF', 'OT',
       'OC', 'EU', 'AS', 'EU', 'AF', 'AF', 'AS', 'AF', 'OC', 'AS', 'EU',
       'OC', 'OT', 'AF', 'AF', 'OC', 'EU', 'AS', 'AS', 'OC', 'OT', 'OC',
       'SA', 'SA', 'AS', 'EU', 'EU', 'AS', 'AS', 'EU', 'EU', 'AS', 'AF',
       'OT', 'OT', 'OT', 'OC', 'EU', 'AF', 'AS', 'AF', 'EU', 'AF', 'AF',
       'AS', 'EU', 'EU', 'OC', 'AF', 'AF', 'EU', 'AS', 'AF', 'SA', 'AF',
       'EU', 'EU', 'AS', 'AS', 'AS', 'EU', 'AS', 'AF', 'OC', 'OT', 'AF',
       'AS', 'AS', 'OC', 'AF', 'EU', 'AS', 'EU', 'AF', 'OT', 'SA', 'AS',
       'OC', 'SA', 'AS', 'AS', 'AF', 'AF'], dtype=object)
```

In [ ]:

In [224…

```
drinks['continent_list'] = drinks['continent']

drinks['continent_list']

#drinks['continent'== continent_list]
```

Out[224…

```
0      AS
1      EU
2      AF
3      EU
4      AF
       ..
188    SA
189    AS
190    AS
191    AF
192    AF
Name: continent_list, Length: 193, dtype: object
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [196…

```
continent_mean
```

Out[196…

```
continent
AF     4986
AS     4706
EU    21085
OC     2940
OT     7721
SA     4227
dtype: int64
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [150…

```
continent_down_mean.groupby('country')['spirit_servings'].sum().idxmax()
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
~₩AppData₩Local₩Temp/ipykernel_13096/3107428466.py in <module>
----> 1 continent_down_mean.groupby('country')['spirit_servings'].sum().idxmax()

~₩anaconda3₩lib₩site-packages₩pandas₩core₩series.py in groupby(self, by, axi
s, level, as_index, sort, group_keys, squeeze, observed, dropna)
   1882         # error: Argument "squeeze" to "SeriesGroupBy" has incompatib
le type
   1883         # "Union[bool, NoDefault]"; expected "bool"
-> 1884         return SeriesGroupBy(
   1885             obj=self,
   1886             keys=by,

~₩anaconda3₩lib₩site-packages₩pandas₩core₩groupby₩groupby.py in __init__(sel
f, obj, keys, axis, level, grouper, exclusions, selection, as_index, sort,
 group_keys, squeeze, observed, mutated, dropna)
    887             from pandas.core.groupby.grouper import get_grouper
    888 
--> 889             grouper, exclusions, obj = get_grouper(
    890                 obj,
    891                 keys,

~₩anaconda3₩lib₩site-packages₩pandas₩core₩groupby₩grouper.py in get_grouper(o
bj, key, axis, level, sort, observed, mutated, validate, dropna)
    860                 in_axis, level, gpr = False, gpr, None
    861             else:
--> 862                 raise KeyError(gpr)
    863         elif isinstance(gpr, Grouper) and gpr.key is not None:
    864             # Add key to exclusions

KeyError: 'country'
```

In [ ]:

```

```

In [ ]:

```

```

연습문제 4

In [118…

```
drinks['alcohol_rate'] = drinks['total_litres_of_pure_alcohol']/drinks['total_servings
drinks['alcohol_rate'] = drinks['alcohol_rate'].fillna(0)
drinks['alcohol_rate']
```

Out[118…

```
0      0.000000
1      0.017818
2      0.017949
3      0.017842
4      0.018495
         ...   
188    0.017661
189    0.017544
190    0.016667
191    0.045455
192    0.054651
Name: alcohol_rate, Length: 193, dtype: float64
```

연습문제 5

In [132…

```
country_with_rank = drinks[['continent','alcohol_rate']]
country_with_rank = country_with_rank.sort_values(by=['alcohol_rate'],ascending =0)
country_with_rank.groupby('continent')['alcohol_rate'].mean()
```

Out[132…

```
continent
AF    0.049741
AS    0.016359
EU    0.017527
OC    0.016895
OT    0.017854
SA    0.017868
Name: alcohol_rate, dtype: float64
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: