

```
In [1]: %matplotlib inline
```

탐색

기본 정보

```
In [2]: # -*- coding : utf-8 -*-

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

drinks = pd.read_csv('C:/Users/sungwonKim/Desktop/SAM/이것이 데이터분석이다 with 파이·
```

```
In [3]: 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 [4]: # 다른 변수와 continent 개수 보면 23개의 차이가 있음.
```

```
In [5]: drinks.head(10)
```

```
Out[5]:
```

	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

나라 / 맥주 소비량 / 스피릿 소비량 / 와인 소비량 / 알콜 섭취량 / 대륙

In [6]:

```
# 숫자정보
drinks.describe()
```

Out[6]:

	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

탐색 및 시각화

피처(열) 간의 상관관계 탐색

In [7]:

```
# 'beer_servings(소비량)', 'wine_servings' 두 피처간의 상관계수를 계산.
# pearson은 상관계수를 구하는 계산 방법 중 하나를 의미
# 가장 널리 쓰이는 방법입니다.
corr = drinks[['beer_servings', 'wine_servings']].corr(method = 'pearson')
print(corr)
```

```

           beer_servings  wine_servings
beer_servings      1.000000      0.527172
wine_servings      0.527172      1.000000
```

In [8]:

```
# 피처간의 상관계수 행렬을 구하기.
cols = ['beer_servings', 'spirit_servings', 'wine_servings', 'total_litres_of_pure_alcohol']
corr = drinks[cols].corr(method = 'pearson')
print(corr)
```

```

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

In [9]:

```
type(corr)
```

Out[9]: pandas.core.frame.DataFrame

In [10]:

```
!pip install seaborn
```

Requirement already satisfied: seaborn in c:\Users\Wsungwonkim\Anaconda3\lib\site-packages (0.11.1)

Requirement already satisfied: scipy>=1.0 in c:\Users\Wsungwonkim\Wanaconda3\lib\site-packages (from seaborn) (1.6.2)
 Requirement already satisfied: matplotlib>=2.2 in c:\Users\Wsungwonkim\Wanaconda3\lib\site-packages (from seaborn) (3.3.4)
 Requirement already satisfied: numpy>=1.15 in c:\Users\Wsungwonkim\Wanaconda3\lib\site-packages (from seaborn) (1.20.1)
 Requirement already satisfied: pandas>=0.23 in c:\Users\Wsungwonkim\Wanaconda3\lib\site-packages (from seaborn) (1.2.4)
 Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.3 in c:\Users\Wsungwonkim\Wanaconda3\lib\site-packages (from matplotlib>=2.2->seaborn) (2.4.7)
 Requirement already satisfied: kiwisolver>=1.0.1 in c:\Users\Wsungwonkim\Wanaconda3\lib\site-packages (from matplotlib>=2.2->seaborn) (1.3.1)
 Requirement already satisfied: pillow>=6.2.0 in c:\Users\Wsungwonkim\Wanaconda3\lib\site-packages (from matplotlib>=2.2->seaborn) (8.2.0)
 Requirement already satisfied: python-dateutil>=2.1 in c:\Users\Wsungwonkim\Wanaconda3\lib\site-packages (from matplotlib>=2.2->seaborn) (2.8.1)
 Requirement already satisfied: cycler>=0.10 in c:\Users\Wsungwonkim\Wanaconda3\lib\site-packages (from matplotlib>=2.2->seaborn) (0.10.0)
 Requirement already satisfied: six in c:\Users\Wsungwonkim\Wanaconda3\lib\site-packages (from cycler>=0.10->matplotlib>=2.2->seaborn) (1.15.0)
 Requirement already satisfied: pytz>=2017.3 in c:\Users\Wsungwonkim\Wanaconda3\lib\site-packages (from pandas>=0.23->seaborn) (2021.1)

In [11]:

```
import seaborn as sns

cols_view = ['beer', 'spirit', 'wine', 'alcohol'] # 그래프 출력을 위한 cols 이름을 축소
sns.set(font_scale=1.2)
hm = sns.heatmap(corr.values,
                  cbar=True, # 맨 오른쪽 게이지(?) 나타내는 옵션
                  annot=True, # 상관 계수 수치 나타내는 옵션
                  square=True, # 그림을 정사각형으로 만드는 옵션
                  fmt='.2f', # 소수점 두 자리 실수로 만들기
                  annot_kws={'size': 15}, # 상관 계수 수치 나타내는 글자 크기
                  yticklabels=cols_view,
                  xticklabels=cols_view)

plt.tight_layout() # plot 간 적절한 간격 유지. 현재 그림이 하나라서 티가 안 남.
plt.show()
```

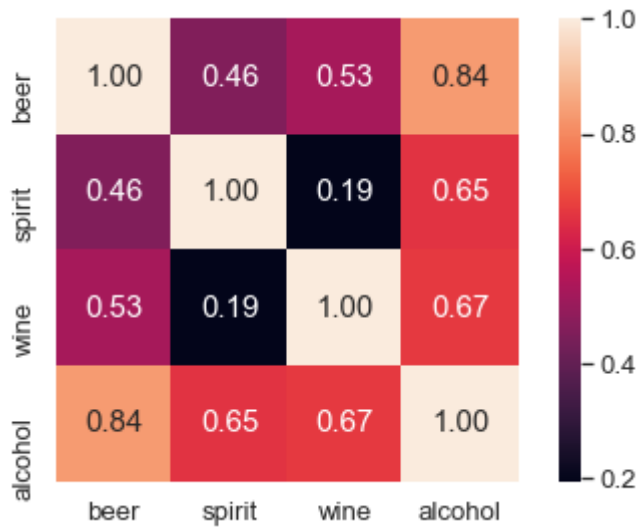


In [12]:

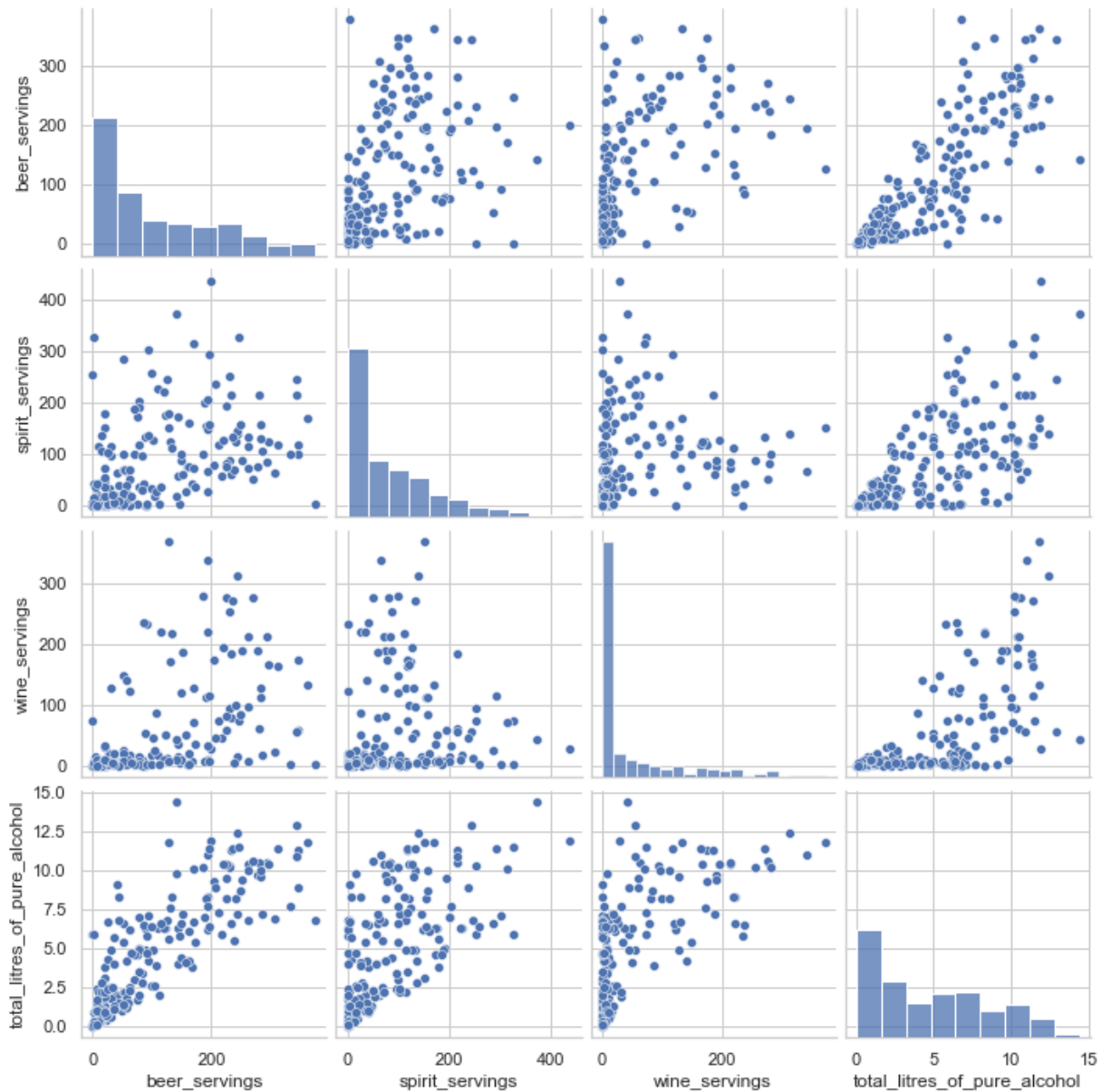
```
# 상관 행렬 히트맵으로 시각화.
cols_view = ['beer', 'spirit', 'wine', 'alcohol'] # 그래프 출력을 위한 cols 이름을 축소
sns.set(font_scale=1.2)
hm = sns.heatmap(corr.values,
                  cbar=True,
                  annot=True,
                  square=True,
                  fmt='.2f',
```

```
annot_kws={'size': 15},
yticklabels=cols_view,
xticklabels=cols_view)
```

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



```
In [13]: # 시각화 라이브러리를 이용한 피쳐간의 scatter plot.
sns.set(style='whitegrid', context='notebook')
sns.pairplot(drinks[['beer_servings', 'spirit_servings',
                    'wine_servings', 'total_litres_of_pure_alcohol']], height=2.5)
plt.show()
```



탐색적 분석

결측 데이터

In [14]:

```
print(drinks.isnull().sum())
print("-----")
print(drinks.dtypes)
```

```
country          0
beer_servings    0
spirit_servings  0
wine_servings    0
total_litres_of_pure_alcohol  0
continent        23
dtype: int64
```

```
-----
country          object
beer_servings    int64
spirit_servings  int64
wine_servings    int64
total_litres_of_pure_alcohol  float64
continent        object
dtype: object
```

```
In [15]: # 결측데이터를 처리 : 기타 대륙으로 통합 -> 'OT'
drinks['continent'] = drinks['continent'].fillna('OT')
# continent 열에 결측값이 있다면 OT로 채워주기.
```

```
In [16]: drinks['continent'].isnull().sum()
```

```
Out[16]: 0
```

```
In [17]: drinks.head(10)
```

```
Out[17]:
```

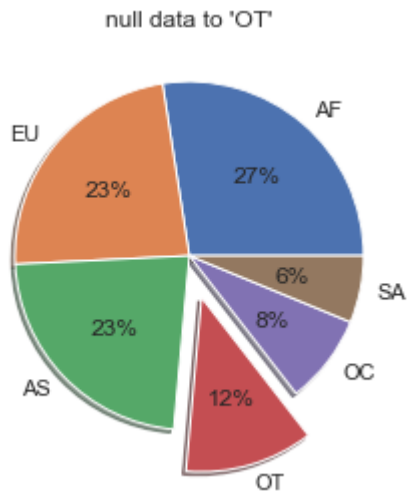
	country	beer_servings	spirit_servings	wine_servings	total_litres_of_pure_alcohol	continent
0	Afghanistan	0	0	0	0.0	AS
1	Albania	89	132	54	4.9	EU
2	Algeria	25	0	14	0.7	AF
3	Andorra	245	138	312	12.4	EU
4	Angola	217	57	45	5.9	AF
5	Antigua & Barbuda	102	128	45	4.9	OT
6	Argentina	193	25	221	8.3	SA
7	Armenia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

```
In [18]: drinks['continent'].value_counts()
```

```
Out[18]: AF    53
EU    45
AS    44
OT    23
OC    16
SA    12
Name: continent, dtype: int64
```

```
In [19]: labels = drinks['continent'].value_counts().index.tolist()
fracs1 = drinks['continent'].value_counts().values.tolist()
explode = (0, 0, 0, 0.25, 0, 0) # 빠져나오는 간격

# pie 차트
plt.pie(fracs1, explode=explode, labels=labels, autopct='%.0f%%', shadow=True)
# fracs1, labels 자리는 꼭 입력해줘야하는 parameter
plt.title('null data to W'OTW'')
plt.show()
```



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

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

어떤 값을 도출할까?

```
In [20]: drinks['continent'].value_counts().index.tolist()
```

```
Out[20]: ['AF', 'EU', 'AS', 'OT', 'OC', 'SA']
```

```
In [21]: drinks['continent'].value_counts().values.tolist()
```

```
Out[21]: [53, 45, 44, 23, 16, 12]
```

그룹 단위

```
In [22]: # 대륙별 spirit_servings의 평균, 최소, 최대, 합계를 계산.
result = drinks.groupby('continent').spirit_servings.agg(['mean', 'min', 'max', 'sum'])
# 같은 문법
# result = drinks.groupby('continent')['spirit_servings'].agg(['mean', 'min', 'max', 'sum'])
# 함수 agg : 그룹별로 list로 정의해놓은 연산을 각각 적용해서 리턴해줌.
result.head()
```

```
Out[22]:
```

	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

```
In [23]: drinks.total_litres_of_pure_alcohol.mean()
```

```
Out[23]: 4.717098445595855
```

```
In [24]: drinks.groupby('continent')['total_litres_of_pure_alcohol'].mean()
```

```
Out[24]: continent
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 [25]: # 전체 평균보다 많은 알코올을 섭취하는 대륙 출력.
total_mean = drinks.total_litres_of_pure_alcohol.mean() # 데이터 전체 평균
continent_mean = drinks.groupby('continent')['total_litres_of_pure_alcohol'].mean() #
continent_over_mean = continent_mean[continent_mean >= total_mean]
print(continent_over_mean)
```

```
continent
EU      8.617778
OT      5.995652
SA      6.308333
Name: total_litres_of_pure_alcohol, dtype: float64
```

```
In [26]: # 평균 beer_servings이 가장 높은 대륙 출력.
beer_continent = drinks.groupby('continent').beer_servings.mean().idxmax()
# groupby 의 index는 대륙이다.
# idxmax : 가장 큰 값의 index를 뽑으면 대륙이 나오겠다.
print(beer_continent)
```

EU

시각화

```
In [27]: n_groups = len(result.index)
means = result['mean'].tolist() # 값들을 list로
mins = result['min'].tolist() # 값들을 list로
maxs = result['max'].tolist() # 값들을 list로
sums = result['sum'].tolist() # 값들을 list로

index = np.arange(n_groups) # : 0, 1, 2, 3
bar_width = 0.1

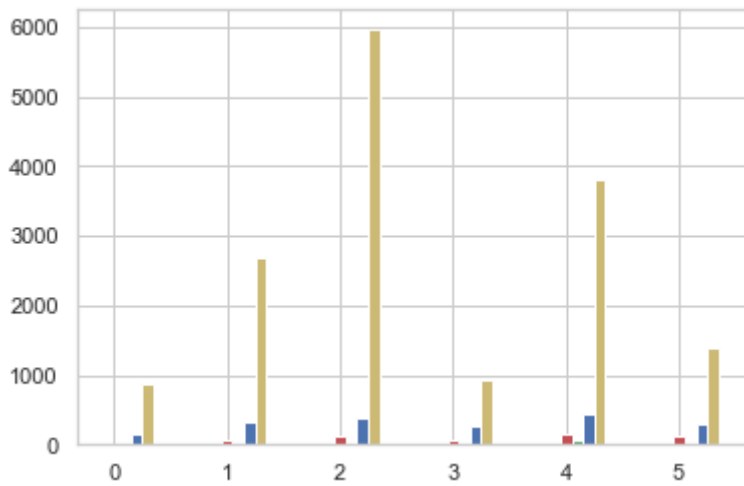
rects1 = plt.bar(index, means, bar_width, # 막대 두께
                  color='r', # red
                  label='Mean') # mean 막대 그래프

rects2 = plt.bar(index + bar_width, mins, bar_width,
                  color='g',
                  label='Min')

rects3 = plt.bar(index + bar_width * 2, maxs, bar_width,
                  color='b',
                  label='Max')

rects4 = plt.bar(index + bar_width * 3, sums, bar_width,
                  color='y',
                  label='Sum')

# plt.xticks(index, result.index.tolist())
# plt.legend()
plt.show()
```

```
In [28]: # 대륙별 spirit_servings의 평균, 최소, 최대, 합계를 시각화.
# result = 대륙별 spirit_servings의 평균, 최소, 최대, 합계를 계산한 list
n_groups = len(result.index)
means = result['mean'].tolist() # 값들을 list로
mins = result['min'].tolist() # 값들을 list로
maxs = result['max'].tolist() # 값들을 list로
sums = result['sum'].tolist() # 값들을 list로

index = np.arange(n_groups) # : 0, 1, 2, 3
bar_width = 0.1

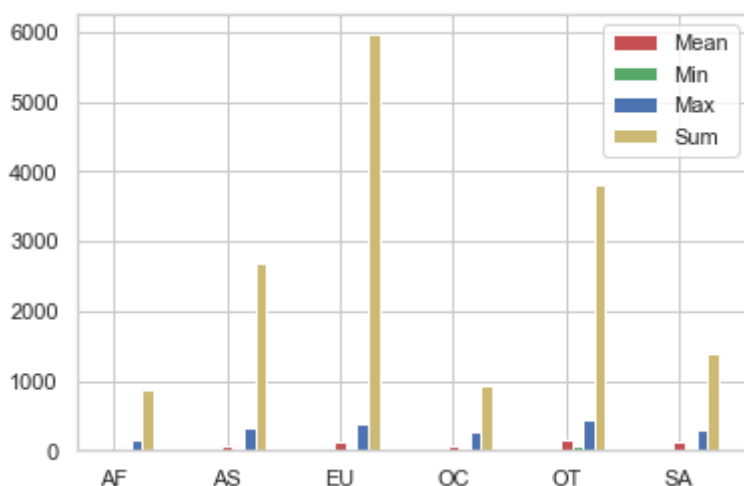
rects1 = plt.bar(index, means, bar_width, # 막대 두께
                  color='r', # red
                  label='Mean') # mean 막대 그래프

rects2 = plt.bar(index + bar_width, mins, bar_width,
                  color='g',
                  label='Min')

rects3 = plt.bar(index + bar_width * 2, maxs, bar_width,
                  color='b',
                  label='Max')

rects4 = plt.bar(index + bar_width * 3, sums, bar_width,
                  color='y',
                  label='Sum')

plt.xticks(index, result.index.tolist()) # x 축에 대륙 이름 넣기
plt.legend() # 우측 상단 legend
plt.show()
```



```

In [29]: # 대륙별 beer_servings의 평균, 최소, 최대, 합계를 시각화.
# result = 대륙별 beer_servings의 평균, 최소, 최대, 합계를 계산한 list
n_groups = len(result.index)

result2 = drinks.groupby('continent').beer_servings.agg(['mean', 'min', 'max', 'sum'])

means = result2['mean'].tolist() # 값들을 list로
mins = result2['min'].tolist() # 값들을 list로
maxs = result2['max'].tolist() # 값들을 list로
sums = result2['sum'].tolist() # 값들을 list로

index = np.arange(n_groups) # : 0, 1, 2, 3
bar_width = 0.1

rects1 = plt.bar(index, means, bar_width, # 막대 두께
                  color='r', # red
                  label='Mean') # mean 막대 그래프

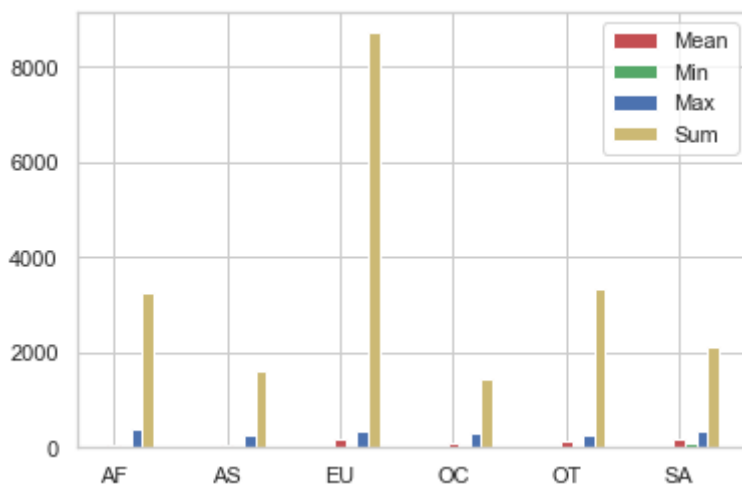
rects2 = plt.bar(index + bar_width, mins, bar_width,
                  color='g',
                  label='Min')

rects3 = plt.bar(index + bar_width * 2, maxs, bar_width,
                  color='b',
                  label='Max')

rects4 = plt.bar(index + bar_width * 3, sums, bar_width,
                  color='y',
                  label='Sum')

plt.xticks(index, result.index.tolist()) # x 축에 대륙 이름 넣기
plt.legend() # 우측 상단 legend
plt.show()

```



```

In [30]: # 대륙별 wine_servings의 평균, 최소, 최대, 합계를 시각화.
# result = 대륙별 wine_servings의 평균, 최소, 최대, 합계를 계산한 list
n_groups = len(result.index)

result3 = drinks.groupby('continent').wine_servings.agg(['mean', 'min', 'max', 'sum'])

means = result3['mean'].tolist() # 값들을 list로
mins = result3['min'].tolist() # 값들을 list로
maxs = result3['max'].tolist() # 값들을 list로
sums = result3['sum'].tolist() # 값들을 list로

index = np.arange(n_groups) # : 0, 1, 2, 3
bar_width = 0.1

```

```

rects1 = plt.bar(index, means, bar_width, # 막대 두께
                  color='r', # red
                  label='Mean') # mean 막대 그래프

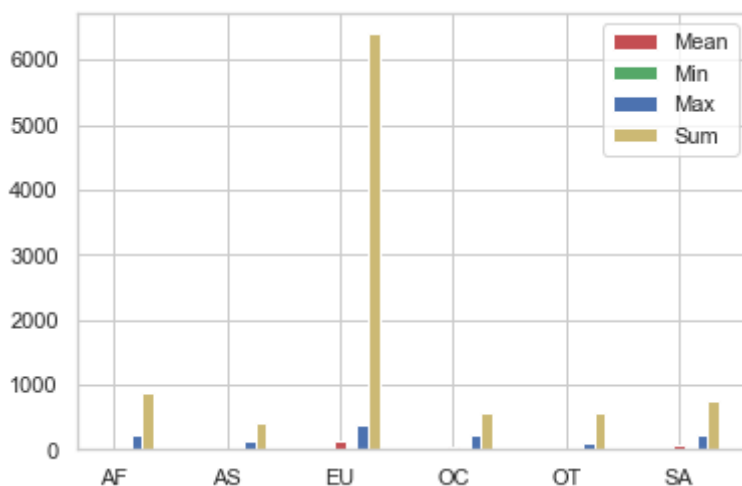
rects2 = plt.bar(index + bar_width, mins, bar_width,
                  color='g',
                  label='Min')

rects3 = plt.bar(index + bar_width * 2, maxs, bar_width,
                  color='b',
                  label='Max')

rects4 = plt.bar(index + bar_width * 3, sums, bar_width,
                  color='y',
                  label='Sum')

plt.xticks(index, result.index.tolist()) # x 축에 대륙 이름 넣기
plt.legend() # 우측 상단 legend
plt.show()

```



In [31]:

```

# 대륙별 total_servings의 평균, 최소, 최대, 합계를 시각화.
# result = 대륙별 total_servings의 평균, 최소, 최대, 합계를 계산한 list
n_groups = len(result.index)

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

result4 = drinks.groupby('continent').total_servings.agg(['mean', 'min', 'max', 'sum'])

means = result4['mean'].tolist() # 값들을 list로
mins = result4['min'].tolist() # 값들을 list로
maxs = result4['max'].tolist() # 값들을 list로
sums = result4['sum'].tolist() # 값들을 list로

index = np.arange(n_groups) # : 0, 1, 2, 3
bar_width = 0.1

rects1 = plt.bar(index, means, bar_width, # 막대 두께
                  color='r', # red
                  label='Mean') # mean 막대 그래프

rects2 = plt.bar(index + bar_width, mins, bar_width,
                  color='g',
                  label='Min')

rects3 = plt.bar(index + bar_width * 2, maxs, bar_width,
                  color='b',

```

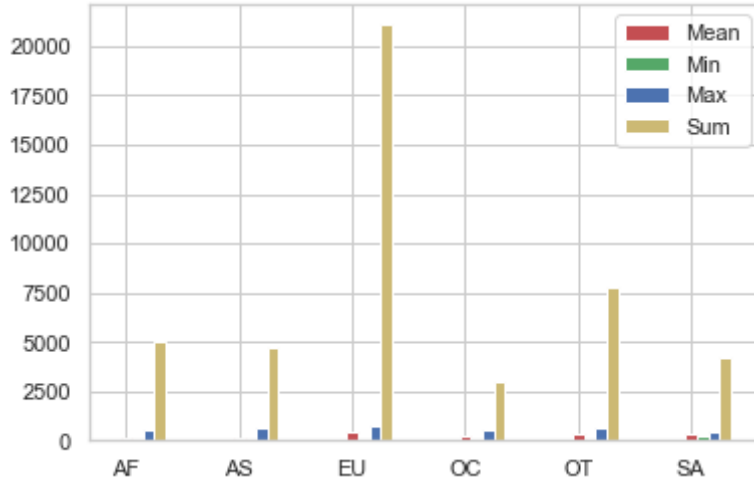
```

label='Max')

rects4 = plt.bar(index + bar_width * 3, sums, bar_width,
                  color='y',
                  label='Sum')

plt.xticks(index, result.index.tolist()) # x 축에 대륙 이름 넣기
plt.legend() # 우측 상단 legend
plt.show()

```



```

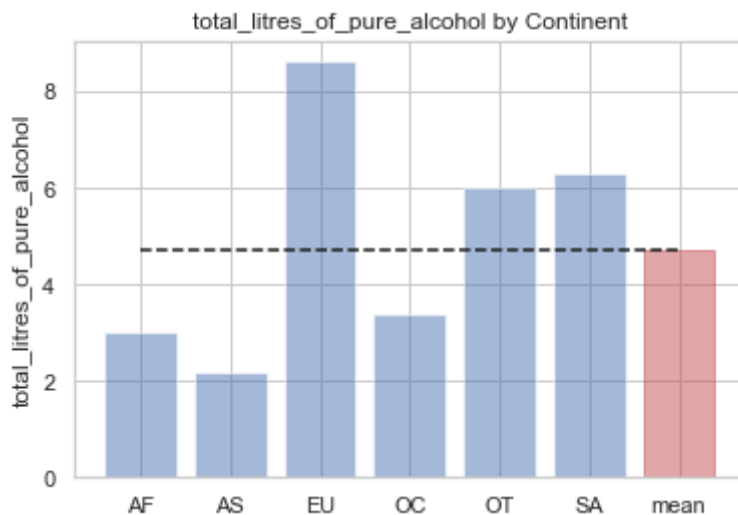
In [32]: # 대륙별 total_litres_of_pure_alcohol을 시각화.
continents = continent_mean.index.tolist() # index = 대륙 이름
continents.append('mean')
x_pos = np.arange(len(continents))
alcohol = continent_mean.tolist()
alcohol.append(total_mean)

bar_list = plt.bar(x_pos, alcohol, align='center', alpha=0.5)
bar_list[len(continents) - 1].set_color('r') # 마지막 거 빨간색으로 만들기
plt.plot([0., 6], [total_mean, total_mean], "k--") # 점선
plt.xticks(x_pos, continents)

plt.ylabel('total_litres_of_pure_alcohol')
plt.title('total_litres_of_pure_alcohol by Continent')

plt.show()

```



```

In [33]: # 대륙별 beer_servings을 시각화.
beer_group = drinks.groupby('continent')['beer_servings'].sum()

```

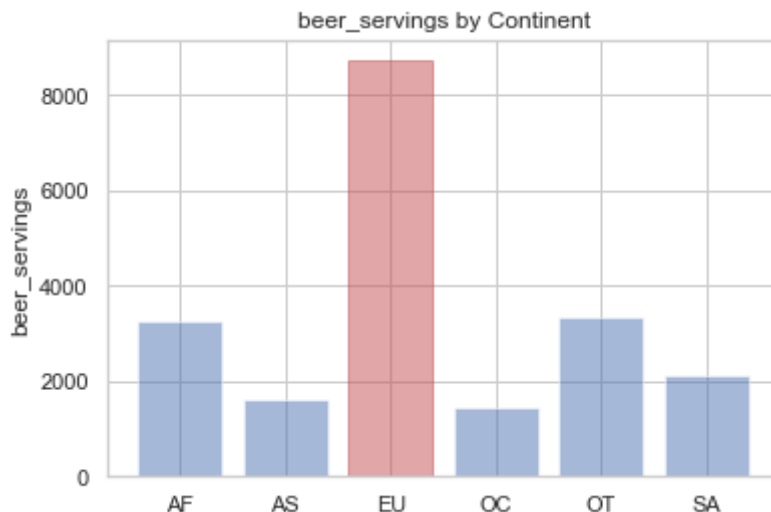
```

continents = beer_group.index.tolist()
y_pos = np.arange(len(continents))
alcohol = beer_group.tolist()

bar_list = plt.bar(y_pos, alcohol, align='center', alpha=0.5)
bar_list[continents.index("EU")].set_color('r')
plt.xticks(y_pos, continents)
plt.ylabel('beer_servings')
plt.title('beer_servings by Continent')

plt.show()

```



In [34]:

```

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

```

```

[16.339622641509433, 60.84090909090909, 132.55555555555554, 58.4375, 165.7391304347826, 114.75]
[0, 0, 0, 0, 68, 25]
[152, 326, 373, 254, 438, 302]
[866, 2677, 5965, 935, 3812, 1377]

```

In [35]:

```

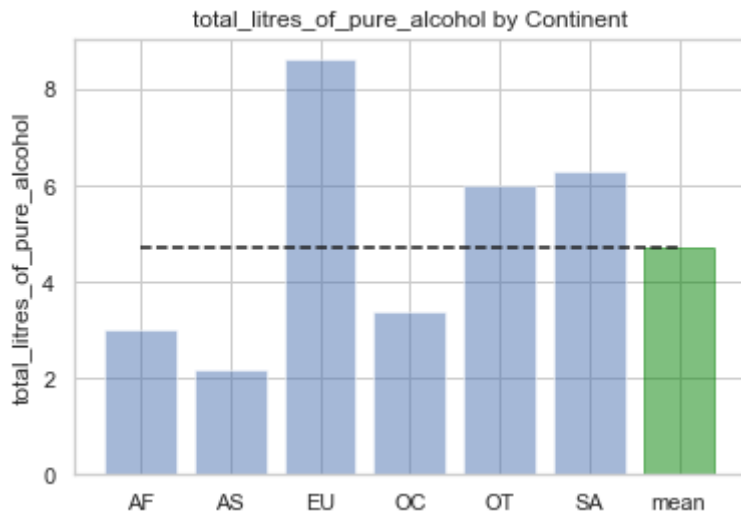
# 대륙별 total_litres_of_pure_alcohol 시각화 - 그래프 색상을 변경.
continents = continent_mean.index.tolist()
continents.append('mean')
x_pos = np.arange(len(continents))
alcohol = continent_mean.tolist()
alcohol.append(total_mean)

bar_list = plt.bar(x_pos, alcohol, align='center', alpha=0.5)
bar_list[len(continents) - 1].set_color('green')
plt.plot([0., 6], [total_mean, total_mean], "k--")
plt.xticks(x_pos, continents)

plt.ylabel('total_litres_of_pure_alcohol')
plt.title('total_litres_of_pure_alcohol by Continent')

plt.show()

```

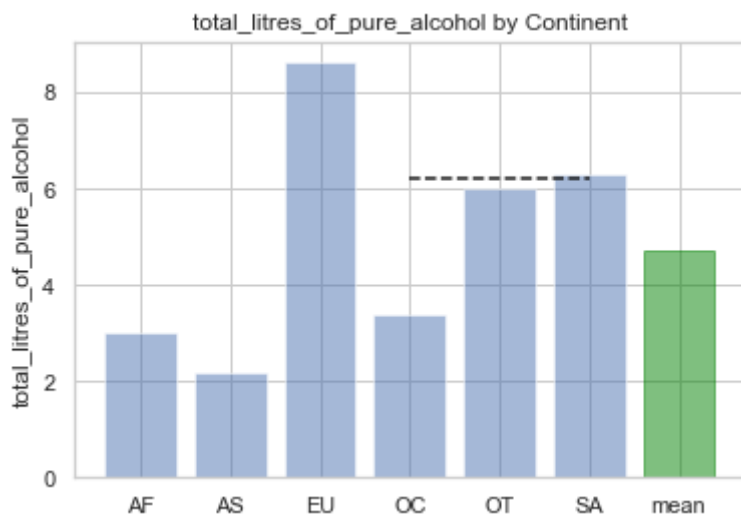


```
In [36]: # 대륙별 total_litres_of_pure_alcohol 시각화 - 점선을 변경.
continents = continent_mean.index.tolist()
continents.append('mean')
x_pos = np.arange(len(continents))
alcohol = continent_mean.tolist()
alcohol.append(total_mean)

bar_list = plt.bar(x_pos, alcohol, align='center', alpha=0.5)
bar_list[len(continents) - 1].set_color('green')
plt.plot([3., 5], [total_mean + 1.5, total_mean + 1.5], "k--")
plt.xticks(x_pos, continents)

plt.ylabel('total_litres_of_pure_alcohol')
plt.title('total_litres_of_pure_alcohol by Continent')

plt.show()
```



분석

```
In [37]: !pip install scipy
```

Requirement already satisfied: scipy in c:\Users\Wsungwonkim\Wanaconda3\Wlib\site-packages (1.6.2)
 Requirement already satisfied: numpy<1.23.0,>=1.16.5 in c:\Users\Wsungwonkim\Wanaconda3\Wlib\site-packages (from scipy) (1.20.1)

T - test

```
In [38]: # 아프리카와 유럽 간의 맥주 소비량 차이를 검정합니다.
africa = drinks.loc[drinks['continent']=='AF']
europe = drinks.loc[drinks['continent']=='EU']

# 두 집단의 차이 검정 => t-test

# library : scipy, 모듈 : stats
from scipy import stats
# 함수 ttest_ind 사용해서 t-test 실행
tTestResult = stats.ttest_ind(africa['beer_servings'], europe['beer_servings'])
# 이분산성 parameter 추가 가능!!
tTestResultDiffVar = stats.ttest_ind(africa['beer_servings'], europe['beer_servings'],
                                     equal_var=False)

print("The t-statistic and p-value assuming equal variances is %.3f and %.3f." % tTestResult)
print("The t-statistic and p-value not assuming equal variances is %.3f and %.3f" % tTestResultDiffVar)
```

The t-statistic and p-value assuming equal variances is -7.268 and 0.000.
The t-statistic and p-value not assuming equal variances is -7.144 and 0.000

어느 나라가 술을 독하게 마실까?

```
In [39]: # total_servings 피처를 생성.
drinks['total_servings'] = drinks['beer_servings'] + drinks['wine_servings'] + drinks['spirit_servings']

# 술 소비량 대비 알콜 비율 피처를 생성.
drinks['alcohol_rate'] = drinks['total_litres_of_pure_alcohol'] / drinks['total_servings']
drinks['alcohol_rate'] = drinks['alcohol_rate'].fillna(0)

# 순위 정보를 생성.
country_with_rank = drinks[['country', 'alcohol_rate']]
country_with_rank = country_with_rank.sort_values(by=['alcohol_rate'], ascending=0)
country_with_rank.head(50)
```

```
Out[39]:
```

	country	alcohol_rate
63	Gambia	0.266667
153	Sierra Leone	0.223333
124	Nigeria	0.185714
179	Uganda	0.153704
142	Rwanda	0.151111
183	Tanzania	0.132558
26	Burkina Faso	0.110256
33	Central African Republic	0.090000
28	Cote d'Ivoire	0.088889
104	Mali	0.085714
162	Sudan	0.080952
101	Malawi	0.075000
27	Burundi	0.071591
47	DR Congo	0.063889
138	South Korea	0.059394
192	Zimbabwe	0.054651

	country	alcohol_rate
164	Swaziland	0.050000
191	Zambia	0.045455
66	Ghana	0.040909
31	Cameroon	0.038158
92	Laos	0.033514
71	Guinea-Bissau	0.031250
58	Ethiopia	0.030435
56	Eritrea	0.027778
15	Belarus	0.025853
95	Lesotho	0.025225
85	Japan	0.023729
34	Chad	0.023529
172	Togo	0.022807
22	Botswana	0.022222
88	Kenya	0.021951
18	Benin	0.021569
159	South Africa	0.021466
121	New Zealand	0.020350
60	Finland	0.020284
98	Lithuania	0.020062
38	Comoros	0.020000
149	Saudi Arabia	0.020000
57	Estonia	0.019916
39	Congo	0.019767
182	United Kingdom	0.019259
81	Ireland	0.019128
8	Australia	0.019083
6	Argentina	0.018907
155	Slovakia	0.018843
93	Latvia	0.018784
115	Mozambique	0.018571
29	Cabo Verde	0.018519
4	Angola	0.018495
161	Sri Lanka	0.018333

In [40]:

위와 다른 방법


```
drinks['alcohol_rate_rank'] = drinks['alcohol_rate'].rank(ascending=False)
drinks['alcohol_rate_rank'] = drinks['alcohol_rate_rank'].apply(np.floor)
drinks.loc[drinks['country'] == 'South Korea', 'alcohol_rate_rank']
```

```
Out[40]: 138    15.0
         Name: alcohol_rate_rank, dtype: float64
```

```
In [41]: country_with_rank[country_with_rank['country'] == 'South Korea']['alcohol_rate'].value
```

```
Out[41]: 0.0593939393939394
```

```
In [42]: # 국가별 순위 정보를 그래프로 시각화.
country_list = country_with_rank.country.tolist()
x_pos = np.arange(len(country_list))
rank = country_with_rank.alcohol_rate.tolist()

bar_list = plt.bar(x_pos, rank)
bar_list[country_list.index("South Korea")].set_color('r')
plt.ylabel('alcohol rate')
plt.title('liquor drink rank by contry')
plt.axis([0, 200, 0, 0.3])

korea_rank = country_list.index("South Korea") # 14 출력 (0부터 시작이어서.)
korea_alc_rate = country_with_rank[country_with_rank['country'] == 'South Korea']['alcohol_rate']
plt.annotate('South Korea : ' + str(korea_rank + 1),
            xy=(korea_rank, korea_alc_rate),
            xytext=(korea_rank + 10, korea_alc_rate + 0.05), # 글자 위치
            arrowprops=dict(facecolor='red', shrink=0.05)) # 화살표 색, 길이

plt.show()
```



```
In [ ]:
```

https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogId=this_africa&logNo=220342032113

```
In [43]: drinks[['country', 'beer_servings', 'wine_servings', 'spirit_servings', 'total_servings']]
```

```
Out[43]:
```

	country	beer_servings	wine_servings	spirit_servings	total_servings
107	Mauritania	0	0	0	0
97	Libya	0	0	0	0

	country	beer_servings	wine_servings	spirit_servings	total_servings
80	Iraq	9	0	3	12
17	Belize	263	8	114	385
192	Zimbabwe	64	4	18	86

In []: