기초분석

5

요약 통계량(평균, 분산, 백분위) 계산 및 활용, 일변량분석(모평균, 모비율 분석)

요약통계량

예제 데이터

```
import pandas as pd
url='http://203.247.53.31/Stat_Notes/example_data/US_crime.csv'
crime=pd.read_csv(url,encoding='ms949')
```

```
crime.columns
```

```
Index(['주이름', '도시이름', '인구수', '폭력사건', '살인', '강간', '강도', '폭행
', '재산범죄', '주거침입',
       '절도', '차량절도'],
      dtype='object')
```

```
crime.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9292 entries, 0 to 9291
Data columns (total 12 columns):
주이름     9292 non-null object
도시이름    9292 non-null object
인구수     9289 non-null float64
폭력사건    9288 non-null float64
살인      9292 non-null int64
강간      5431 non-null float64
강도      9292 non-null int64
폭행      9289 non-null float64
재산범죄    9288 non-null float64
주거침입    9290 non-null float64
절도      9290 non-null float64
차량절도    9292 non-null int64
```

- non-null 결측치 없음
- object : 문자열 형식
- int : 정수형식
- float : 실수 형식

```
crime.head(3)
```

	주이름	도시이름	인구수	폭력사건	살인	강간	강도	폭행	재산범죄
0	ALABAMA	Abbeville	2645.0	11.0	1	1.0	2	7.0	63.0
1	ALABAMA	Adamsville	4481.0	19.0	1	0.0	7	11.0	321.0
2	ALABAMA	Addison	744.0	1.0	0	1.0	0	0.0	25.0

기초통계량 구하기

전체 데이터 : DF.dropna().describe()[['측정 변수1', '측정 변수2', ...]]

- dropna() : 결측치 제외하고 통계량 계산, 결측치 없다면 제외해도 됨
- describe() 만 사용하면 데이터 내 모든 측정형 변수 결과 모두 출력된다.

```
pd.options.display.float_format = '{:.2f}'.format
crime.dropna().describe()[['강도','절도']]
```

	강도	절도
count	5425.00	5425.00
mean	20.13	369.23
std	182.00	1399.42

그룹변수 이용 DF.groupby(['그룹변수1', '그룹변수2']).describe()[['변수1', …']]

```
crime.groupby(['주이름']).describe()[['강도', '절도']]
```

	강도				
	count	mean	std	min	25%
주이름					
ALABAMA	196.00	18.24	83.37	0.00	0.00
ALASKA	27.00	22.19	100.17	0.00	0.00
ARIZONA	63.00	97.33	427.08	0.00	1.00
ARKANSAS	154.00	12.29	77.44	0.00	0.00

- 향후 분석에 사용하기 위하여 결과를 데이터프레임 'crime_des' 이름으로 저장

```
crime_des=crime.groupby(['주이름']).describe()[['강도', '절도']]
crime_des.head(3)
```

	강도								절도	
	count	mean	std	min	25%	50%	75%	max	count	mean
주이름										
ALABAMA	196.00	18.24	83.37	0.00	0.00	1.00	8.00	969.00	196.00	393.20
ALASKA	27.00	22.19	100.17	0.00	0.00	1.00	2.00	522.00	27.00	539.89
ARIZONA	63.00	97.33	427.08	0.00	1.00	5.00	19.50	3233.00	63.00	2063.16

열 첫번째 인덱스 사용하기

일반 데이터프레임에서는 열 인덱스 = 변수이름과 일치한다.

crime_des[['변수명1', '변수명2', …]]

```
crime_des['강도']
```

	count	mean	std	min	25%	50%	75%
주이름							
ALABAMA	196.00	18.24	83.37	0.00	0.00	1.00	8.00
ALASKA	27.00	22.19	100.17	0.00	0.00	1.00	2.00
ARIZONA	63.00	97.33	427.08	0.00	1.00	5.00	19.50

열 두번째 인덱스 사용하기

DF.loc[:, (slice(None), ['이름1','이름2', …])]

```
crime_des.loc[:, (slice(None), ['mean','std'])]
```

	강도		절도	
	mean	std	mean	std
주이름				
ALABAMA	18.24	83.37	393.20	1076.53
ALASKA	22.19	100.17	539.89	1877.36
ARIZONA	97.33	427.08	2063.16	5917.16

GROUPBY 기초통계량 구하기

DF.groupby('그룹명').통계량()[['측정변수1', '측정변수2', ...]]

- 통계량() 뒤의 측정변수 생략하면 데이터 내 모든 측정형 변수 통계량 출력
- 통계량 : count, mean, median, min, max, mode, mad(mean absolute value), abs, prod, std, avr, quantile(0.9)
- 그룹변수는 행 인덱스에 사용된다.

```
crime_m=crime.groupby('주이름').mean()[['강도','절도']]
crime_m.head(3)
```

주이름	강도	절도
ALABAMA	18.244898	393.198980
ALASKA	22.185185	539.888889
ARIZONA	97.333333	2063.158730

그룹변수가 2개인 경우
DF.groupby('그룹명1', '그룹명2', ...).통계량()[['측정변수1', '측정변수2', ...]]

그룹 변수 하나더 만들기 (인구수 많은 지역과 적은 지역, 평균기준)

```
import numpy as np
crime['인구수그룹']=np.where(crime['인구수']>crime['인구수'].mean(),'이상','미만')
```

```
crime['인구수그룹'].value_counts()
```

```
미만    7463
이상    1829
Name: 인구수그룹, dtype: int64
```

```
crime_m2=crime.groupby(['주이름','인구수그룹']).mean()[['강도','절도']]
crime_m2.head(3)
```

주이름	인구수그룹	강도	절도
ALABAMA	미만	3.78	156.24
	이상	121.92	2091.42
ALASKA	미만	1.04	114.83

요약통계량 활용방법

1) 평균, 표준편차 요약통계량을 하나의 데이터프레임을 만든다.

		강도	절도	강도	절도
주이름	인구수그룹				
ALABAMA	미만	3.78	156.24	7.45	200.54
	이상	121.92	2091.42	213.86	2469.29
ALASKA	미만	1.04	114.83	1.49	115.30

2) 주이름, 인구수그룹 자체 활용방법 (첫번째 행 인덱스)

DF.loc[[‘첫 행 범주1’,’범주2’, ...],:]

```
crime_des=pd.concat([crime_m,crime_sd],axis=1)
```

```
crime_des.loc[['ALASKA','ALABAMA'],:]
```

		강도	절도	강도	절도
주이름	인구수그룹				
ALABAMA	미만	3.78	156.24	7.45	200.54
	이상	121.92	2091.42	213.86	2469.29
ALASKA	미만	1.04	114.83	1.49	115.30
	이상	191.33	3940.33	286.58	5113.97

3) 주이름, 인구수그룹 자체 활용방법 (첫번째, 두번째 행 인덱스)

DF.loc[(slice(None), slice('두번째 행인덱스 범주')), :]

```
crime_des.loc[(slice(None), slice('미만')), :]
```

		강도	절도	강도	절도
주이름	인구수그룹				
ALABAMA	미만	3.78	156.24	7.45	200.54
ALASKA	미만	1.04	114.83	1.49	115.30
ARIZONA	미만	3.49	190.08	7.02	173.22

4) 행 인덱스를 열 변수(인덱스로 만들기)

DF.reset_index(level=['행인덱스명1','행인덱스명2'])

```
crime_des.reset_index(level=['주이름','인구수그룹'])
```

	주이름	인구수그룹	강도	절도	강도	절도
0	ALABAMA	미만	3.78	156.24	7.45	200.54
1	ALABAMA	이상	121.92	2091.42	213.86	2469.29
2	ALASKA	미만	1.04	114.83	1.49	115.30

그래프요약

그래프 요약 (일집단) : 히스토그램 histogram

- 결측치 제거하고 인구 1000명당 절도건수 계산 - Series 저장
- 100 미만이 데이터만 저장

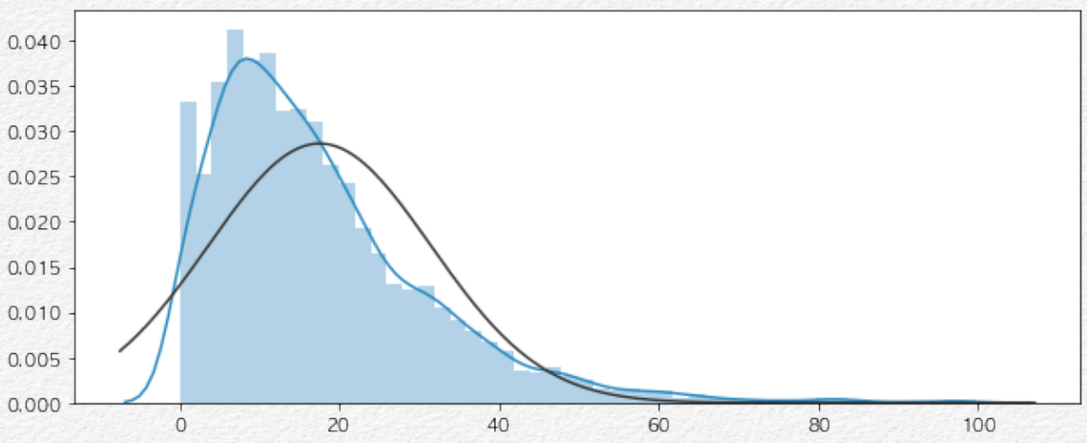
```
crime0=crime.dropna()
x=crime0['절도']/crime0['인구수']*1000
x=x[x<100]
x.describe()
```

```
count    5400.00
mean       17.52
std        13.94
min         0.00
25%         7.56
50%        14.38
75%        23.80
max        99.66
dtype: float64
```

import seaborn as sns; import matplotlib.pyplot as plt
from scipy import stats; sns.distplot(x, fit=stats.norm); plt.show()

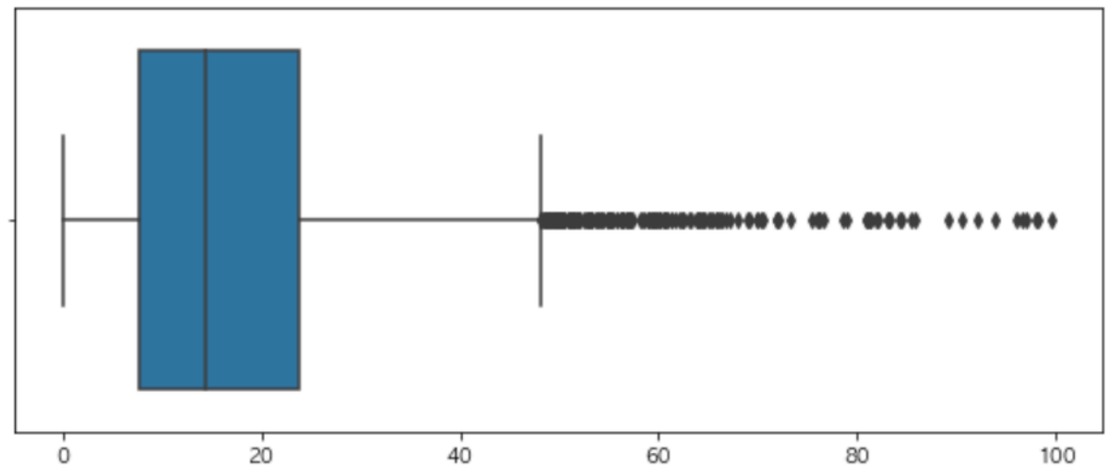
검은색 - fit 정규분포, 파랑색 - 히스트그램 kernel 분포함수

우로 치우침



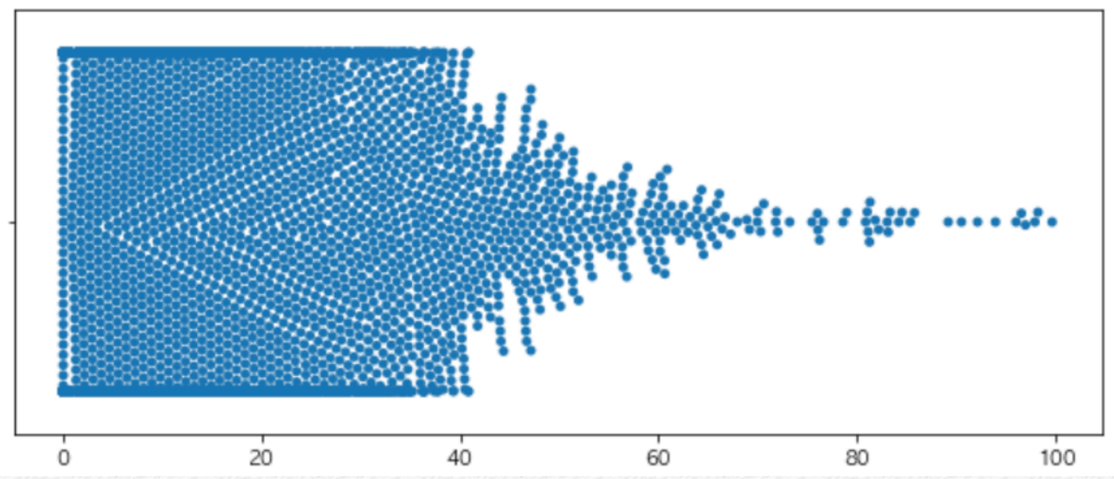
나무상자 그림 box whisker plot

```
sns.boxplot(x)
plt.show()
```



데이터 관측값 모두 표시

```
sns.swarmplot(x)
plt.show()
```



그래프 요약 : 2집단 이상 (나무상자 그림)

예제 데이터 2018년 4분기 미세먼지 데이터

http://203.247.53.31/Stat_Notes/example_data/mise/2018Q4.csv

```
import pandas as pd
url='http://203.247.53.31/Stat_Notes/example_data/mise/2018Q4.csv'
미세=pd.read_csv(url)
```

```
미세.head(3)
```

	지역	망	측정소코드	측정소명	측정일시	SO2	CO	O3	N
0	서울 중구	도시대기	111121	중구	2018100101	0.00	0.30	0.04	0.
1	서울 중구	도시대기	111121	중구	2018100102	0.00	0.30	0.04	0.

지역 변수 데이터 공백으로 2개로 나눔, 첫 번째는 [0], 두번째는 [1]

```
미세['지역'].str.split(' ', n=1, expand=True)
```

	0	1
0	서울	중구
1	서울	중구

```
미세['시도']=미세['지역'].str.split(' ', n=1, expand=True)[0]
```

측정일시 10자리 숫자 int 형식 -> date_time형식 만들기

- astype(str) : 숫자형 - 문자형으로 변환
- str.slice(start=, stop=, step=) : 문자열 데이터 일부 짜르기

```
(미세['측정일시'].astype(str)).str.slice(0,8)
```

```
0       20181001
1       20181001
```

- pd.to_datetime() : 8자리 문자를 날짜 형식으로 변환

```
pd.to_datetime((미세['측정일시'].astype(str)).str.slice(0,8))
```

```
0      2018-10-01
1      2018-10-01
2      2018-10-01
```

```
미세['측정일']=pd.to_datetime((미세['측정일시'].astype(str)).str.slice(0,8))
미세.info()
```

```
시도      849878 non-null object
측정일     849878 non-null datetime64[ns]
```

5대 광역시 데이터 subset

```
1 미세=미세[(미세['시도']=='서울') | (미세['시도']=='대전') |
2 (미세['시도']=='광주') | (미세['시도']=='대구') | (미세['시도']=='부산')]
```

```
미세['시도'].value_counts()
```

```
서울    87576
부산    52992
대구    33120
대전    24080
광주    19872
```

광역시 다수 지점, 일별 대표값(max) 지정

미세먼지는 지역, 시간별 최대값이 대표값으로 적절함

```
미세subset=미세.groupby(['시도','측정일'])['PM10','PM25'].max()
미세subset.head(3)
```

		PM10	PM25
시도	측정일		
광주	2018-10-01	62.00	18.00
	2018-10-02	67.00	61.00
	2018-10-03	49.00	41.00

행 인덱스 열 변수로 변환하기

```
미세subset.reset_index(inplace=True)
미세subset.head(3)
```

	시도	측정일	PM10	PM25
0	광주	2018-10-01	62.00	18.00
1	광주	2018-10-02	67.00	61.00
2	광주	2018-10-03	49.00	41.00

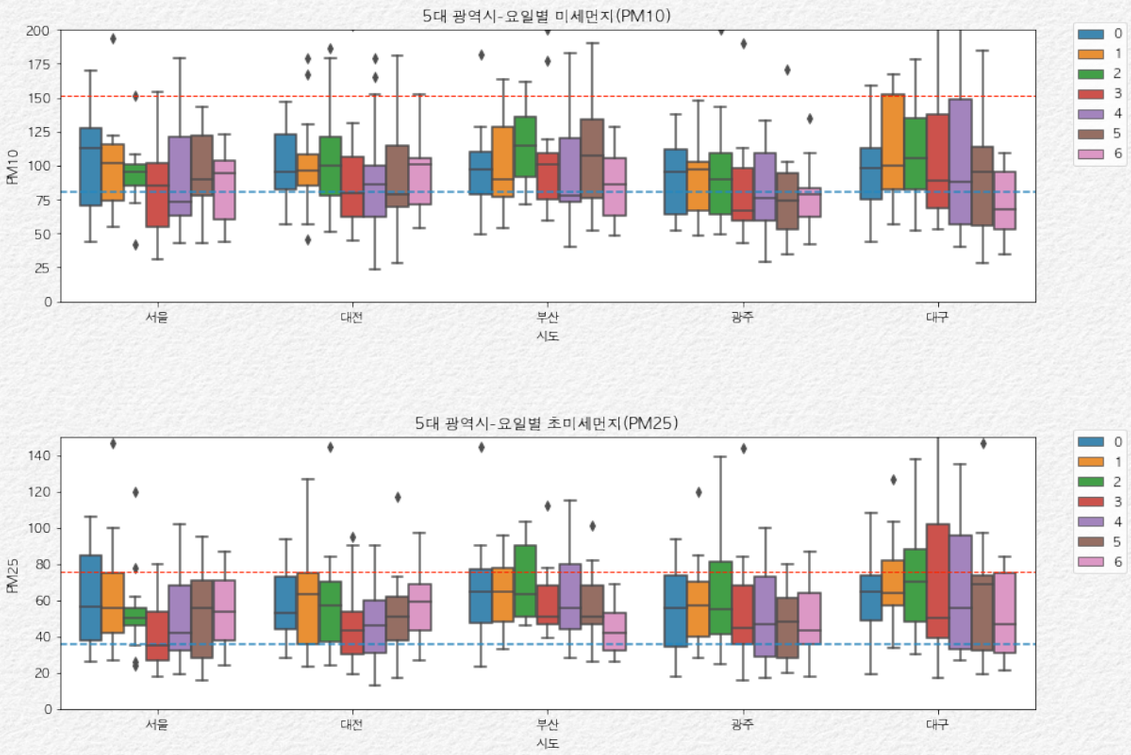
행 인덱스 열 변수명으로 변환

5대 광역시 PM10, PM25 박스플롯

측정일의 요일(dt.weekday 0=월요일, 2=화요일, ... 6=일요일) 데이터 만듬

```
미세subset['요일']=미세subset['측정일'].dt.weekday
```

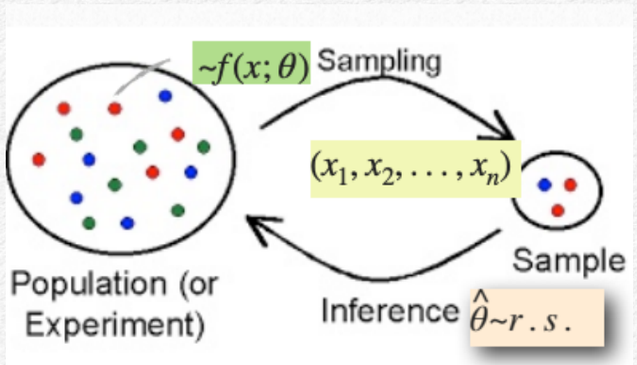
```
plt.rcParams["figure.figsize"] = (14,10) #set size of Graph
plt.subplots_adjust(hspace = 0.5, wspace = 0.3)
plt.subplot(2,1,1)
ax = sns.boxplot(x="시도", y="PM10", hue="요일",
                 data=미세subset, order=['서울', '대전', '부산', '광주', '대구'])
plt.axhline(151, ls='--', c='Red', lw=1); plt.axhline(81, ls='--')
plt.ylim(0,200)
plt.title('5대 광역시-요일별 미세먼지(PM10)')
ax.legend(bbox_to_anchor=(1.1,1.05))
plt.subplot(2,1,2)
ax = sns.boxplot(x="시도", y="PM25", hue="요일",
                 data=미세subset, order=['서울', '대전', '부산', '광주', '대구'])
plt.axhline(76, ls='--', c='Red', lw=1); plt.axhline(36, ls='--')
plt.ylim(0,150)
plt.title('5대 광역시-요일별 초미세먼지(PM25)')
ax.legend(bbox_to_anchor=(1.1,1.05))
plt.show()
```




```
plt.rcParams["figure.figsize"] = (14,10) #set size of Graph
plt.subplots_adjust(hspace = 0.5, wspace = 0.3)
plt.subplot(2,1,1)
ax = sns.boxplot(x="시도", y="PM10", hue="요일",
                 data=미세subset, order=['서울', '대전', '부산', '광주', '대구'])
plt.axhline(151, ls='--', c='Red', lw=1); plt.axhline(81, ls='--')
plt.ylim(0,200)
plt.title('5대 광역시-요일별 미세먼지(PM10)')
ax.legend(bbox_to_anchor=(1.1,1.05))
plt.subplot(2,1,2)
ax = sns.boxplot(x="시도", y="PM25", hue="요일",
                 data=미세subset, order=['서울', '대전', '부산', '광주', '대구'])
plt.axhline(76, ls='--', c='Red', lw=1); plt.axhline(36, ls='--')
plt.ylim(0,150)
plt.title('5대 광역시-요일별 초미세먼지(PM25)')
ax.legend(bbox_to_anchor=(1.1,1.05))
plt.show()
```

1집단_모평균

개념



Population Mean	Sample Mean
$\mu = \frac{\sum_{i=1}^{N} x_i}{N}$	$\overline{X} = \frac{\sum_{i=1}^{n} x_i}{n}$

(수리적 증명) $E(\bar{X}) = \mu, V(\bar{X}) = \frac{\sigma^2}{n}$

중심극한정리 Central Limit Theorem

표본크기가 충분히 크면($n \geq 20 \sim 30$) 모집단 분포함수와 관계없이 표본합, 표본평균은 정규분포에 근사한다.

모집단 분산 σ^2의 MVUE : $\hat{\sigma^2} = s^2$ (표본분산) => $\frac{\bar{X} - \mu}{s/\sqrt{n}} \sim t(n-1)$

예제 데이터

```
import pandas as pd
url='http://203.247.53.31/Stat_Notes/example_data/US_crime.csv'
crime=pd.read_csv(url,encoding='ms949')
```

```
crime.columns
```

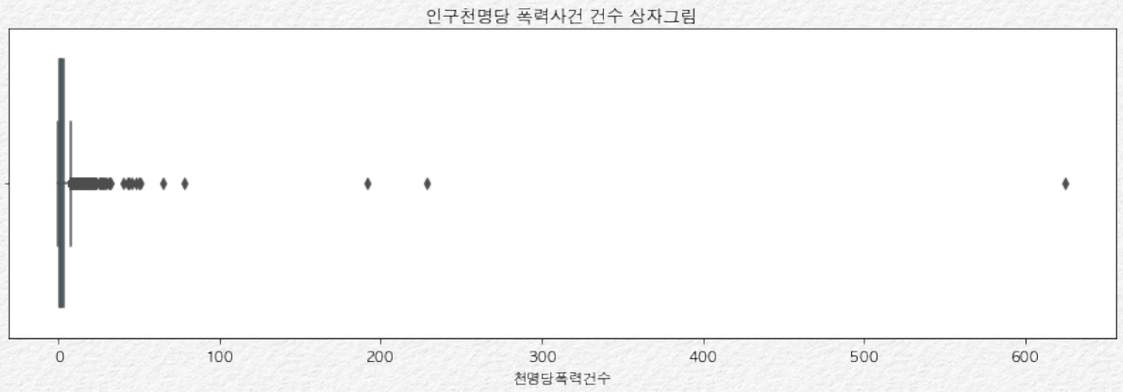
```
Index(['주이름', '도시이름', '인구수', '폭력사건', '살인', '강간', '강도', '폭행
', '재산범죄', '주거침입',
       '절도', '차량절도'],
      dtype='object')
```

(분석 변수) 인구 1,000명당 폭력사건 건수

```
crime['천명당폭력건수']=crime['폭력사건']/crime['인구수']*1000
```

순서 1) 상자수염 그림 - 치우침, 이상치 진단

```
import matplotlib.pyplot as plt
import seaborn as sns
sns.boxplot(crime['천명당폭력건수'])
plt.title('인구천명당 폭력사건 건수 상자그림')
plt.show()
```



우로 치우침

우로 치우침 존재 -> 정규성 검정 후 정규변환 필요

이상치 존재

이상치는 치우침 문제 해결 후(정규변환은 특성 상 이상치 문제를 해결) 이상치가 존재하면 그 때 해결해도 되나, 본 예제에서는 극단에 가까운 이상치가 존재, 이를 제거하고 치우침을 해결하는 것이 적절하다.

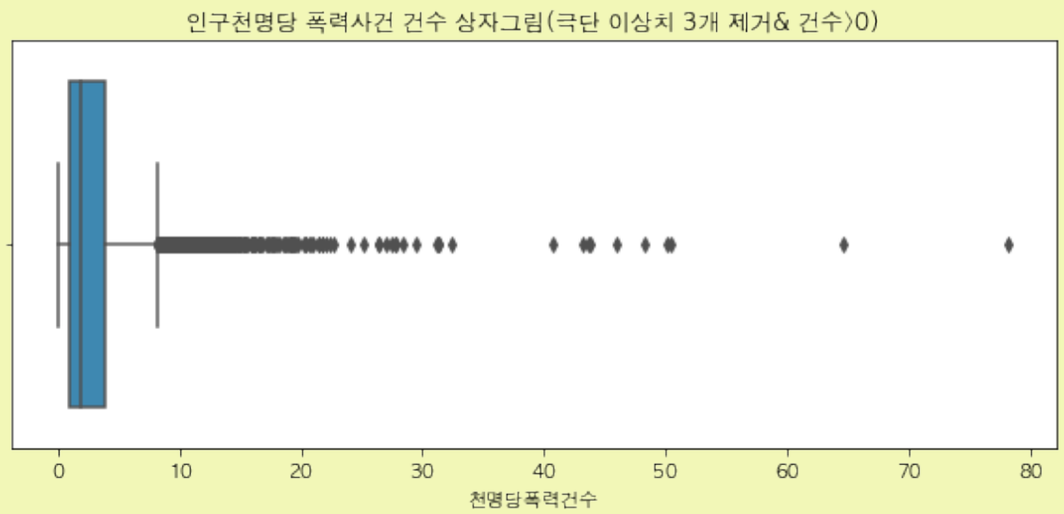
이 부분은 필요하지 않을 수 있음

- 극단 이상치 3개 도시 : 폭력 사건 극심 도시, 향후 분석에서는 제외
- 또한 폭력건수가 없는 도시는 제외(분석 대상에서)

```
crime2=crime[(crime['천명당폭력건수']<100) & (crime['천명당폭력건수']>0)]
crime2.loc[:,['주이름','도시이름','천명당폭력건수']].head(3)
```

	주이름	도시이름	천명당폭력건수
0	ALABAMA	Abbeville	4.158790
1	ALABAMA	Adamsville	4.240125
2	ALABAMA	Addison	1.344086

```
sns.boxplot(crime2['천명당폭력건수'])
plt.title('인구천명당 폭력사건 건수 상자그림(극단 이상치 3개 제거& 건수>0)')
plt.show()
```



순서 2) 치우침 해결 - 정규성 검정 및 정규변환 (필요시)
http://203.247.53.31/2019spring/LM2019/%EC%84%A0%ED%98%95%EB%AA%A8%ED%98%95_LM%20%EC%A0%84%EC%B2%98%EB%A6%AC.pdf

정규성검정 : 3 방법 모두 유의수준 1%에서 정규성 귀무가설 기각

1) Shapiro Wilk W-통계량

$W = \frac{\left(\sum_{i=1}^{n} a_i x_{(i)}\right)^2}{\sum_{i=1}^{n}(x_i - \overline{x})^2}$, 상수 a_i 는 분산행렬을 이용하여 구함

```
import scipy.stats as stats  #정규성 검정 : Shapiro-Wilks
st, p = stats.shapiro(crime2['천명당폭력건수'])
print('통계량=',st,'유의확률=',p)
```

```
통계량= 0.6386462450027466 유의확률= 0.0
```

2) Anderson-Darling AD 통계량

$$A^2 = n \int_{-\infty}^{+\infty} (F_n(x) - F(x))^2 [F(x)(1 - F(x))]^{-1} dF(x)$$

```
import scipy.stats as stats  #정규성 검정 : Anderson Darling
stats.anderson(crime2['천명당폭력건수'])
```

```
AndersonResult(statistic=627.6875489169561, critical_values=array
significance_level=array([15. , 10. ,  5. ,  2.5,  1. ]))
```

3) D'Agostino's K-squared test

https://en.wikipedia.org/wiki/D%27Agostino%27s_K-squared_test

```
import scipy.stats as stats  #D'Agostino, R. and Pearson, E. S. (1973)
st, p = stats.normaltest(crime2['천명당폭력건수'])
print('통계량=',st,'유의확률=',p)
```

```
통계량= 8877.249819208031 유의확률= 0.0
```

치우침 해결 - 정규성 가정이 기각되었고 상자그림을 보면 우로 치우침 - 정규변환 필요

Box-Cox tranformation $x' = \frac{(x^\lambda - 1)}{\lambda}$ -> 최적 $\lambda = 0.07$

Tukey 변환가 동일하지만 최적의 λ 값은 MLE 방법에 의해 찾음

```
xt, lmd = stats.boxcox(crime2['천명당폭력건수'])  #box cox transformation
```

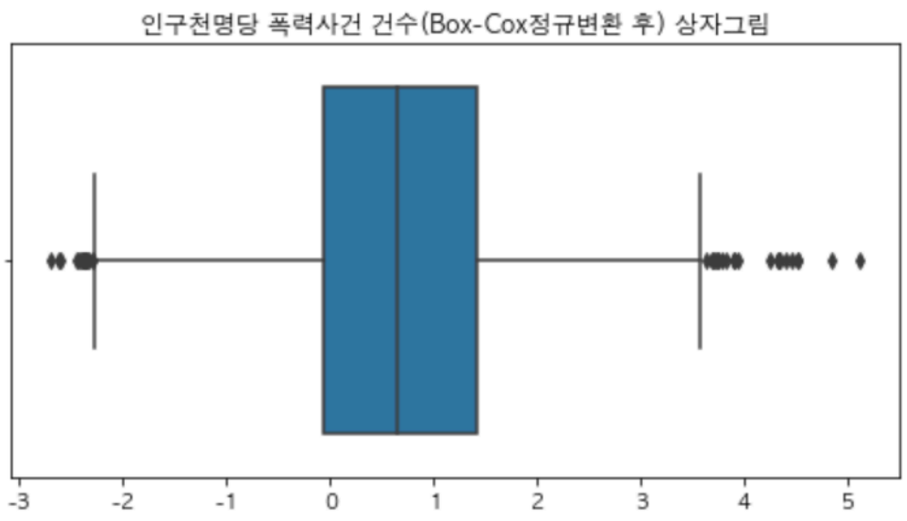
```
lmd  #optimal lambda
```

```
0.07049399339914822
```

Power Transformation

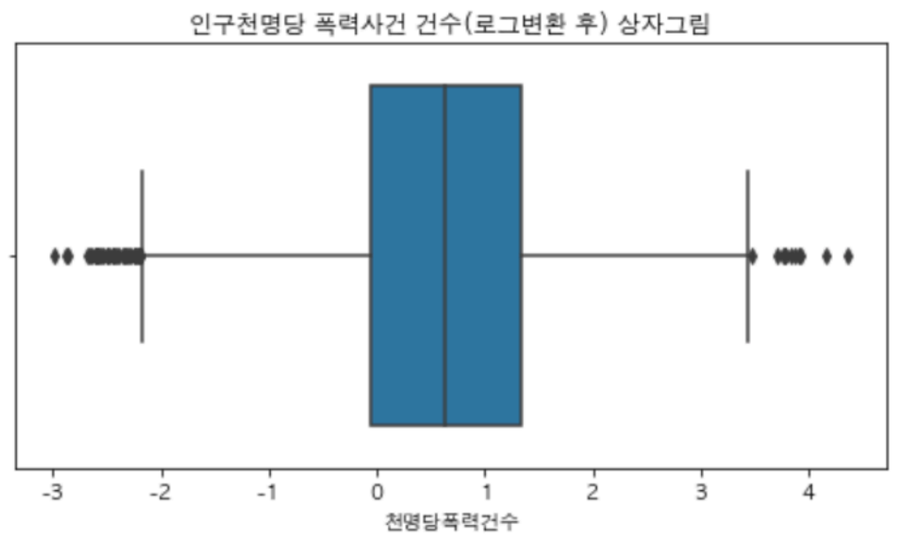
$$Y^* = \begin{cases} Y^3, \textit{ left} \\ Y^2, \textit{ mild left} \\ \sqrt{Y}, \textit{ mild right} \\ \ln(Y), \textit{ right} \\ 1/Y, \textit{ severe right} \end{cases}$$

```
sns.boxplot(xt)  #box-cox normal transformation
plt.title('인구천명당 폭력사건 건수(Box-Cox정규변환 후) 상자그림')
plt.show()
```




```
import numpy as np
xt1=np.log(crime2['천명당폭력건수'])
```

```
sns.boxplot(xt1)  #box-cox normal transformation
plt.title('인구천명당 폭력사건 건수(로그변환 후) 상자그림')
plt.show()
```



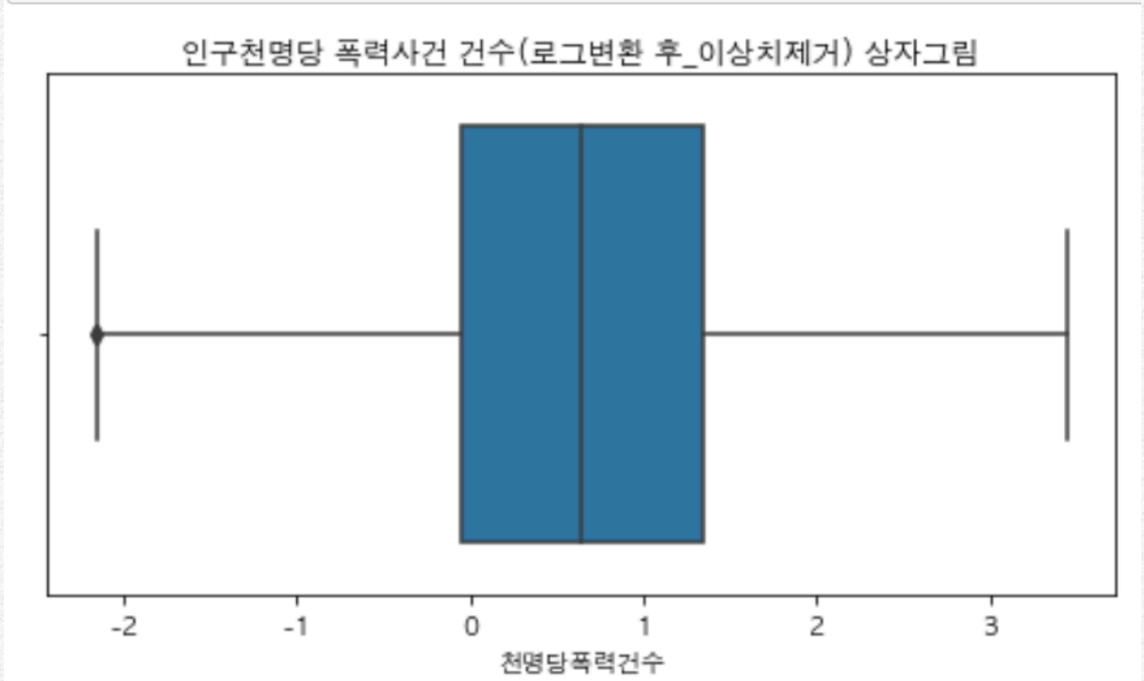
[결론] 로그변환, Cox-Box $\lambda = 0.07$ 정규변환 유사해 보인다. 앞에서 언급하였듯이 $\lambda = 0$인 경우 로그변환이므로 유사한 결과를 얻는다. 결과가 유사하다면 활용 해석이 간편한 로그변환이 용이하다.

최종 데이터 - (자연)로그변환 후 이상치 제거

```
q1=xt1.quantile(0.25); q3=xt1.quantile(0.75)
lb=q1-1.5*(q3-q1); ub=q3+1.5*(q3-q1)   #이상치 진단 imaginary line
```

```
xt1=xt1[(xt1>lb) & (xt1<ub)]   #이상치 제거
```

```
sns.boxplot(xt1)  #box-cox normal transformation
plt.title('인구천명당 폭력사건 건수(로그변환 후_이상치제거) 상자그림')
plt.show()
```



순서 3) 1집단 모평균 추론

모수=모집단평균 ($\theta = \mu$) MVUE($\hat{\theta} = \bar{x}$) 샘플링분포

$$\frac{\bar{x} - \mu}{s/\sqrt{n}} \sim t(n-1)$$

```
xt1.describe()
```

```
count    8174.000000
mean        0.621430
std         0.998987
min        -2.166536
25%        -0.054463
50%         0.642692
75%         1.346651
max         3.445314
Name: 천명당폭력건수, dtype: float64
```

신뢰구간 $(\bar{x} - t_{(\frac{\alpha}{2},n-1)}\frac{s}{\sqrt{n}},\ \bar{x} + t_{(\frac{\alpha}{2},n-1)}\frac{s}{\sqrt{n}})$

95% 신뢰구간 (0.263, 13.193) <- 자연로그 변환하였으므로 exp 변환 필요함

```
stats.t.interval(0.95, xt1.count()-1, loc=xt1.mean(), scale=xt1.std())  #95% C.I
```

```
(-1.3368378407219295, 2.5796986812931704)
```

```
import numpy as np
print('하한=',np.exp(-1.3368378407219295),'상한=',np.exp(2.5796986812931704))
```

```
하한= 0.26267497680195523 상한= 13.193162214097708
```

가설검정 : $H_0 : \mu = \mu_0 : Ts = \frac{\bar{x} - \mu_0}{s/\sqrt{n}} \sim t(n-1)$

작년 인구 천명당 폭력건수는 평균 1.9명이었다. (자연로그 변환이므로 $ln(1.9) = 0.6418538861723947$) 올해 작년보다 폭력건수가 줄어다고 할 수 있나?

귀무가설 : $\mu = 0.6418538861723947$, 대립가설 : $\mu < 0.6418538861723947$

```
stats.ttest_1samp(xt1,np.log(1.9)) #two-sided
```

```
Ttest_1sampResult(statistic=-1.8483618134258697, pvalue=0.0645861181323626)
```

양측검정을 위한 유의확률이 0.065이므로 단측검정은 0.033 -> 귀무가설 기각, 그러므로 올해 인구천명당 폭력건수는

```
np.exp(xt1.mean())
```

```
1.8615889929123852
```

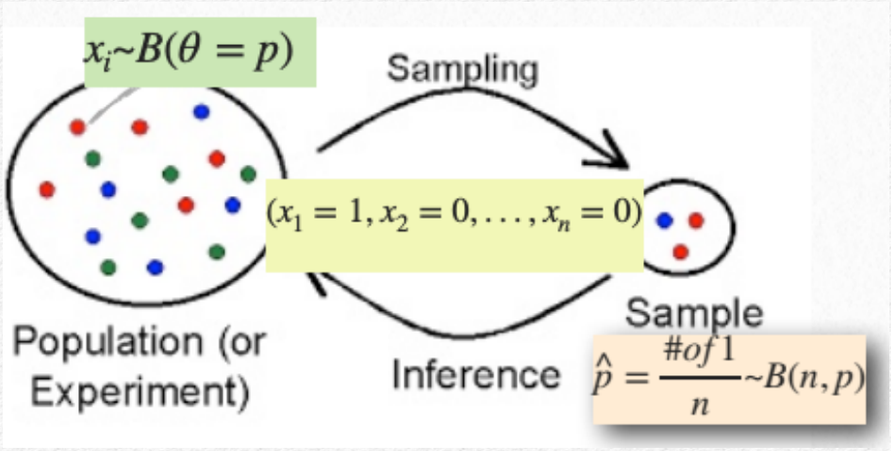
으로 작년 1.9명보다 줄어다고 할 수 있다. (유의수준 5%)

```
import pandas as pd
url='http://wolfpack.hnu.ac.kr/Stat_Notes/example_data/US_crime.csv'
crime=pd.read_csv(url,encoding='ms949')

crime.columns

crime['천명당폭력건수']=crime['폭력사건']/crime['인구수']*1000
#인구 1000명당 폭력건수

import matplotlib.pyplot as plt
import seaborn as sns
sns.boxplot(crime['천명당폭력건수'])
plt.title('인구천명당 폭력사건 건수 상자그림')
plt.show()

crime2=crime[(crime['천명당폭력건수']<100) & (crime['천명당폭력건수']>0)]
crime2.loc[:,['주이름','천명당폭력건수']].head(3)

crime2['천명당폭력건수'].describe()

sns.boxplot(crime2['천명당폭력건수'])
plt.title('인구천명당 폭력사건 건수 상자그림(극단 이상치 3개 제거& 건수>0)')
plt.show()

import scipy.stats as stats  #정규성 검정 : Shapiro-Wilks
st, p = stats.shapiro(crime2['천명당폭력건수'])
print('통계량=',st,'유의확률=',p)

import scipy.stats as stats  #정규성 검정 : Anderson Darling
stats.anderson(crime2['천명당폭력건수'])

import scipy.stats as stats  #D'Agostino, R. and Pearson, E. S. (1973)
st, p = stats.normaltest(crime2['천명당폭력건수'])
print('통계량=',st,'유의확률=',p)
```

```
xt, lmd = stats.boxcox(crime2['천명당폭력건수'])   #box cox transformation

lmd  #optimal lambda

sns.boxplot(xt)  #box-cox normal transformation
plt.title('인구천명당 폭력사건 건수(Box-Cox정규변환 후) 상자그림')
plt.show()

import numpy as np
xt1=np.log(crime2['천명당폭력건수'])

sns.boxplot(xt1)  #box-cox normal transformation
plt.title('인구천명당 폭력사건 건수(로그변환 후) 상자그림')
plt.show()

q1=xt1.quantile(0.25); q3=xt1.quantile(0.75)
lb=q1-1.5*(q3-q1); ub=q3+1.5*(q3-q1)   #이상치 진단 imaginary line

xt1=xt1[(xt1>lb) & (xt1<ub)]   #이상치 제거

sns.boxplot(xt1)  #box-cox normal transformation
plt.title('인구천명당 폭력사건 건수(로그변환 후_이상치제거) 상자그림')
plt.show()

xt1.describe()

stats.t.interval(0.95, xt1.count()-1, loc=xt1.mean(), scale=xt1.std())
#95% C.I

import numpy as np
print('하한=',np.exp(-1.3368378407219295),'상한
=',np.exp(2.5796986812931704))

stats.ttest_1samp(xt1,np.log(1.9)) #two-sided

np.exp(xt1.mean())
```

1집단_모비율

개념



- MVUE for p : 표본비율 $\hat{p} = \frac{\#\_of\_success}{n}$
- 표본비율 평균 : $E(\hat{p}) = p$: 불편추정량
- 표본비율 분산 : $V(\hat{p}) = \frac{p(1-p)}{n}$

$$ts = \frac{\hat{p} - p}{\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}} \sim N(0,1)$$

- 100($1-\alpha$)% 신뢰구간

$$\hat{p} - z_{(1-\alpha/2)}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \le p \le \hat{p} + z_{(1-\alpha/2)}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$$

예제 데이터

아침으로 시리얼을 구입하는 고객 1,250명을 대상으로 시리얼 구입 할 때 고려하는 이 유와 1)건강음식을 위하여 2)다이어트 효과 3)식이요 법 4)가격고려, 그리고 시리얼 구입 비용을 조사하였다.

```
import pandas as pd
url='http://wolfpack.hnu.ac.kr/2015_Fall/D4BE/cereal.csv'
cereal=pd.read_csv(url)
cereal.head(3)
```

	Group	Spend
0	1	14.77

모비율 추론

소표본 대표본 구분

기준 : $min(np_0, n(1-p_0)) > 5$ - 대표본

소표본인 경우 : $\sum X_i = (num)of\,1's \sim B(n, p_0)$ 이용하여 가설 검정

연구가설 설정

작년 다이어트 효과(2번)를 우선 고려하여 아침 시리얼을 선택한다는 고객 비율이 35%이었다. 올해 수집한 데이터로부터 다이어트 효과 우선 고려하여 시리얼 선택한 비율에 대한 95% 신뢰구간을 구하시오. 그리고 작년에 비해 증가하였는지 유의수준 5%에서 검정하시오.

```
count=cereal[cereal['Group']==2].shape[0]
nobs=cereal.shape[0]
print(count,':',nobs)
h0=0.35
```

```
484 : 1250
```

n=1,250 / x=#of (2)=484

모비율 추정

- 올해 시리얼 선택 고려사항이 다이어트 비율 38.7%

```
phat=cereal[cereal['Group']==2].shape[0]/cereal.shape[0]
phat
```

```
0.3872
```

95% 신뢰구간 계산 : 모비율 p

- (36%, 41.4%) : 95% 다이어트 고려 시리얼 선택비율 신뢰구간

```
import numpy as np
from statsmodels.stats.proportion import proportion_confint
lb,ub=proportion_confint(count, nobs, alpha=0.05, method='normal')
```

```
print("하한 : {:.2%}, 상한 : {:.2%}".format(lb,ub))
```

```
하한 : 36.02%, 상한 : 41.42%
```

가설검정 $H_0 : p = 0.35, H_a : p > 0.35$

유의확률 0.003/2로 매우 작음 - 귀무가설이 기각되어 올해 다이어트 고려 시리얼 비율 38.7%는 작년에 비해 유의적으로 증가하였다고 결론 내림.

```
from statsmodels.stats.proportion import proportions_ztest
stat, pval = proportions_ztest(count,nobs,h0)
```

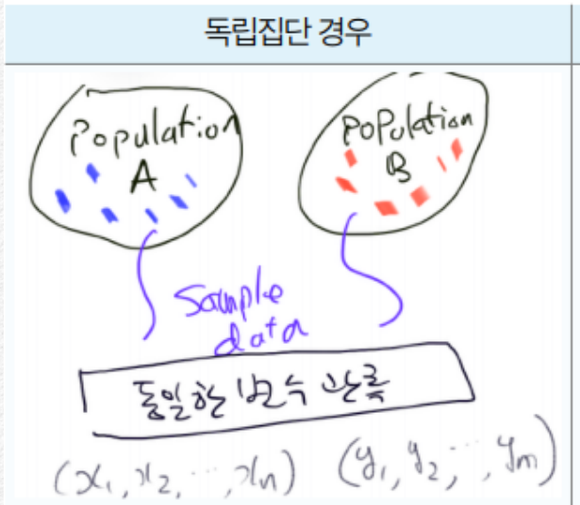
```
print("검정통계량 : % .2f, 유의확률 : % .3f" %(stat, pval/2))
```

```
검정통계량 :  2.70, 유의확률 :  0.003
```

```
import pandas as pd
url='http://wolfpack.hnu.ac.kr/2015_Fall/D4BE/cereal.csv'
cereal=pd.read_csv(url)

cereal['Group'].value_counts()/cereal.shape[0]

phat=cereal[cereal['Group']==2].shape[0]/cereal.shape[0]

count=cereal[cereal['Group']==2].shape[0]
nobs=cereal.shape[0]
print(count,':',nobs,'phat=',phat); h0=0.35

import numpy as np
from statsmodels.stats.proportion import proportion_confint
lb,ub=proportion_confint(count, nobs, alpha=0.05, method='normal')

print("하한 : {:.2%}, 상한 : {:.2%}".format(lb,ub))

from statsmodels.stats.proportion import proportions_ztest
stat, pval = proportions_ztest(count,nobs,h0)

print("검정통계량 : % .2f, 유의확률 : % .3f" %(stat, pval/2))
```

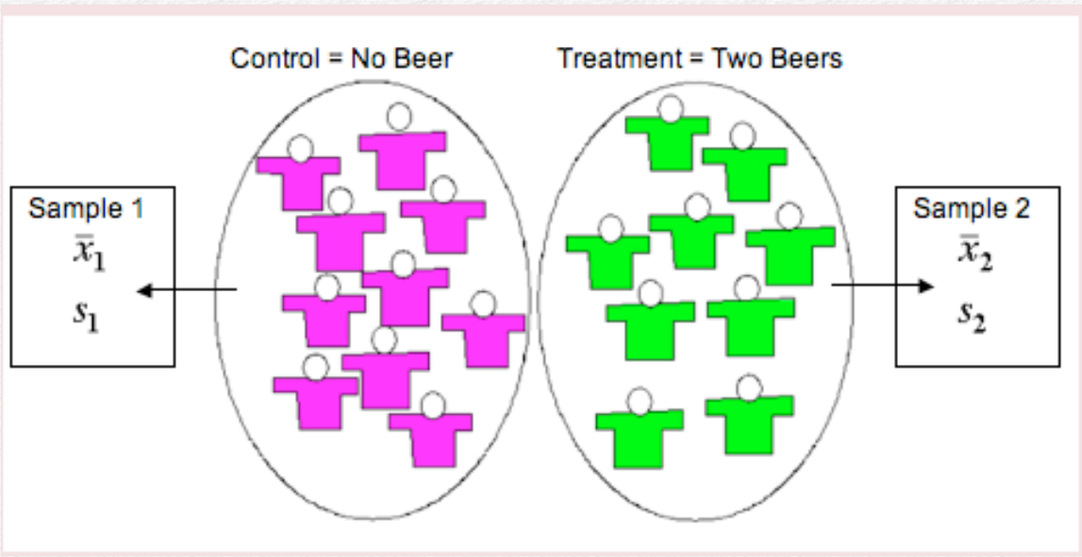
2독립집단 모평균차이

개념

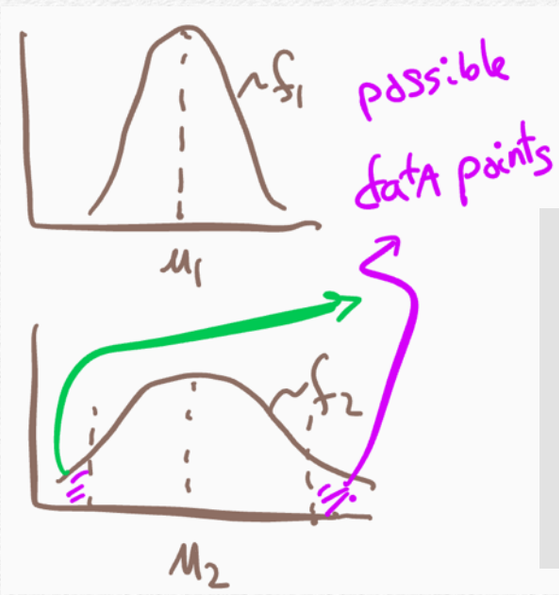


독립집단 경우	짝진집단 경우
관측변수 X=Y 동일함	
서로 다른 개체로 부터 관측한다.	동일개체로부터 관측된다.
관측치 개수 n, m 크기는 같을 필요 없음	관측치 개수는 n개 항상 동일하다.
경영학 전공자와 통계학 전공자의 일주일 공부 시간 차이를 알고자 한다. 경영학 전공자 30명, 통계학 전공자 20명을 임의 추출(확률표본)하여 공부시간을 측정하였다.	경영학 전공자와 통계학 전공자의 일주일 공부 시간 차이를 알고자 한다. 그런데. 학점 군에 따른 공부시간 차이를 고려하여 (4.5~4.3), (4.3~4.1), (4.1~3.9), ... 각 구간에서 한 명씩 임의 추출하였다.

독립인 두 집단 평균 추론 개념



등분산 검정 필요 이유



1, 2집단 평균($\mu_1 = \mu_2$) 크기는 동일하나 1집단에 비해 2집단의 분산이 커, 2집단의 경우 꼬리 부분의 데이터 표본에 포함되는 경우 검정통계량 크기가 왜곡될 수 있음

(Welch – Satterthwaite's) : 이분산 t-검정

$$t = \frac{(\bar{X}_1 - \bar{X}_2) - (\mu_1 - \mu_2)H_0}{\sqrt{\frac{\hat{\sigma}_1^2}{n_1} + \frac{\hat{\sigma}_2^2}{n_2}}} \sim t\left(\text{d.f.} = \frac{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)^2}{\frac{1}{n_1 - 1}\left(\frac{s_1^2}{n_1}\right)^2 + \frac{1}{n_2 - 1}\left(\frac{s_2^2}{n_2}\right)^2} \right)$$

등분산 $t(n_1 + n_2 - 2)$

$$t = \frac{(\bar{X}_1 - \bar{X}_2) - (\mu_1 - \mu_2)}{\sqrt{\left(\frac{s_p^2}{n_1} + \frac{s_p^2}{n_2}\right)}}$$

$$s_p^2 = \frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}$$

예제 데이터

```
import pandas as pd
url='http://wolfpack.hnu.ac.kr/Stat_Notes/example_data/US_crime.csv'
crime=pd.read_csv(url,encoding='ms949')
```

```
crime.columns
```

```
Index(['주이름', '도시이름', '인구수', '폭력사건', '살인', '강간', '강도', '폭행', '재산범죄', '주거칩입',
       '절도', '차량절도'],
      dtype='object')
```

```
crime['천명당폭력건수']=crime['폭력사건']/crime['인구수']*1000   #인구 1000명당 폭력건수
```

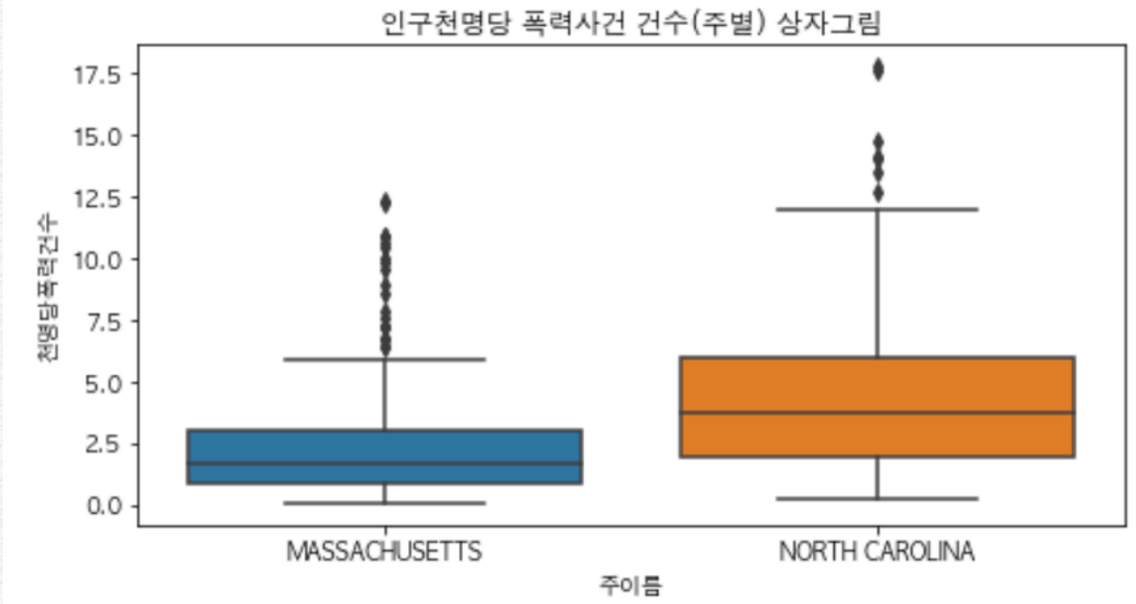
```
crime2=crime[(crime['천명당폭력건수']<100) & (crime['천명당폭력건수']>0)]
```

```
crime3=crime2[(crime2['주이름']=='MASSACHUSETTS') |
              (crime2['주이름']=='NORTH CAROLINA')]
crime3s=crime3[['주이름','천명당폭력건수']]
crime3s.head(3)
```

	주이름	천명당폭력건수
3136	MASSACHUSETTS	2.977483
3137	MASSACHUSETTS	0.349788
3138	MASSACHUSETTS	1.833623

순서 1 : 상자 수염 그림

```
import matplotlib.pyplot as plt
import seaborn as sns
sns.boxplot(y=crime3['천명당폭력건수'],x=crime3['주이름'])
plt.title('인구천명당 폭력사건 건수(주별) 상자그림')
plt.show()
```



치우침 해결 필요 - 없음

동일한 척도에 대한 것이므로 두 집단의 분포는 동일하므로 굳이 치우침을 해결하여 비교할 필요는 없음

이상치 제거 - 필요함

ma, nc - Pandas Series 데이터 만들고 각각 이상치 제거

```
ma=crime3[crime3['주이름']=='MASSACHUSETTS']['천명당폭력건수']
nc=crime3[crime3['주이름']=='NORTH CAROLINA']['천명당폭력건수']
```

```
ma_q1=ma.quantile(0.25);ma_q3=ma.quantile(0.75)
ma0=ma[(ma>ma_q1-1.5*(ma_q3-ma_q1)) & (ma<ma_q3+1.5*(ma_q3-ma_q1))]
nc_q1=nc.quantile(0.25);nc_q3=nc.quantile(0.75)
nc0=nc[(nc>nc_q1-1.5*(nc_q3-nc_q1)) & (nc<nc_q3+1.5*(nc_q3-nc_q1))]
```

순서 2 : 두 모집단 등분산 검정 : $H_0 : \sigma_1^2 = \sigma_2^2$: Levene 방법

Levene, H. (1960). In Contributions to Probability and Statistics: Essays in Honor of Harold Hotelling, I. Olkin et al. eds., Stanford University Press, pp. 278-292.

- 유의확률이 매우 작아 귀무가설(등분산) 기각 -> 등분산 가정 무너짐

```
stats.levene(ma,nc)
```

```
LeveneResult(statistic=22.77357015899783, pvalue=2.4399490866987887e-06)
```

순서 3 : 두 모집단 평균 차이 추론

- 이분산의 검정통계량 7.18, 유의확률 0에 가까우므로 귀무가설이 기각되고 NC=4.29건, MA=2.36건보다 유의적으로 크다.

```
stats.ttest_ind(ma,nc, equal_var = False)
```

```
Ttest_indResult(statistic=-7.184715248021864, pvalue=4.31592392630537e-12)
```

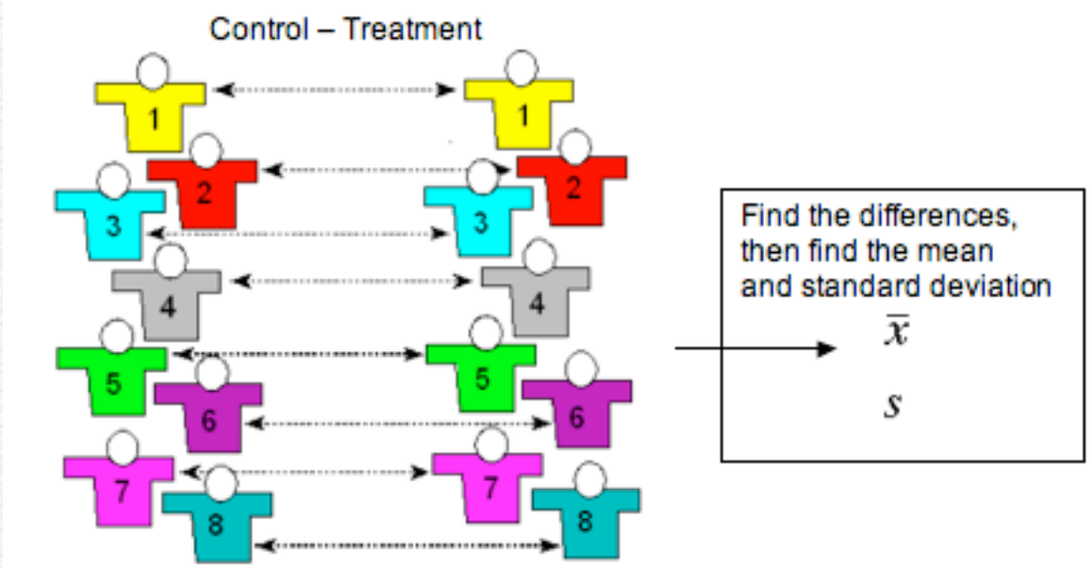
```
print('MA(m,sd)=',ma.mean(), ma.std(),'\n',
      'NC(m,sd)=',nc.mean(), nc.std())
```

```
MA(m,sd)= 2.3607692188506437 2.3015769368996595
 NC(m,sd)= 4.287963581180141 3.229275380557057
```

```
import pandas as pd
url='http://wolfpack.hnu.ac.kr/Stat_Notes/example_data/US_crime.csv'
crime=pd.read_csv(url,encoding='ms949')

import scipy.stats as stats
crime['천명당폭력건수']=crime['폭력사건']/crime['인구수']*1000
ma=crime[crime['주이름']=='MASSACHUSETTS']['천명당폭력건수']
nc=crime[crime['주이름']=='NORTH CAROLINA']['천명당폭력건수']

ma_q1=ma.quantile(0.25);ma_q3=ma.quantile(0.75)
ma0=ma[(ma>ma_q1-1.5*(ma_q3-ma_q1)) &
(ma<ma_q3+1.5*(ma_q3-ma_q1))]
nc_q1=nc.quantile(0.25);nc_q3=nc.quantile(0.75)
nc0=nc[(nc>nc_q1-1.5*(nc_q3-nc_q1)) & (nc<nc_q3+1.5*(nc_q3-nc_q1))]

stats.levene(ma,nc)

stats.ttest_ind(ma,nc, equal_var = False)

print('MA(m,sd)=',ma.mean(), ma.std(),'\n',
      'NC(m,sd)=',nc.mean(), nc.std())
```

2짝진집단 모평균차이

짝진 두 집단 평균 추론



차이에 대한 일변량 추론과 동일함. 점추정치 (MVUE) : $\hat{\mu}_d = \bar{d} = \frac{\sum d_i}{n}$

예제 데이터

http://wolfpack.hnu.ac.kr/Stat_Notes/elem_stat/Stat_methods/mba2.csv

MBA 전공 재무, 마케팅 연봉을 조사한 자료이다. 성적(GPA)에 따른 차이가 있을 가능성을 고려 하여 (4, 3.92)=1그룹, (3.92, 3,84)=2그룹2, ..., 총 25개 그룹에서 한 명씩 임의 추출하여 연봉을 조사하였다. 유의수준 5%에서 재무전공, 마케팅 전공의 연봉 차이가 있는지 검정하시오.

stats.ttest_rel(mba['Finance'],mba['Marketing'])

```
#two paired two population means
import pandas as pd
url='http://wolfpack.hnu.ac.kr/Stat_Notes/elem_stat/Stat_methods/mba2.csv'
mba=pd.read_csv(url)
mba.head(3)
```

	Group	Finance	Marketing
0	1	95171	89329
1	2	88009	92705

```
from scipy import stats
stats.ttest_rel(mba['Finance'],mba['Marketing'])
```

```
Ttest_relResult(statistic=3.80968841351656, pvalue=0.0008510
```

```
mba[['Finance','Marketing']].describe()
```

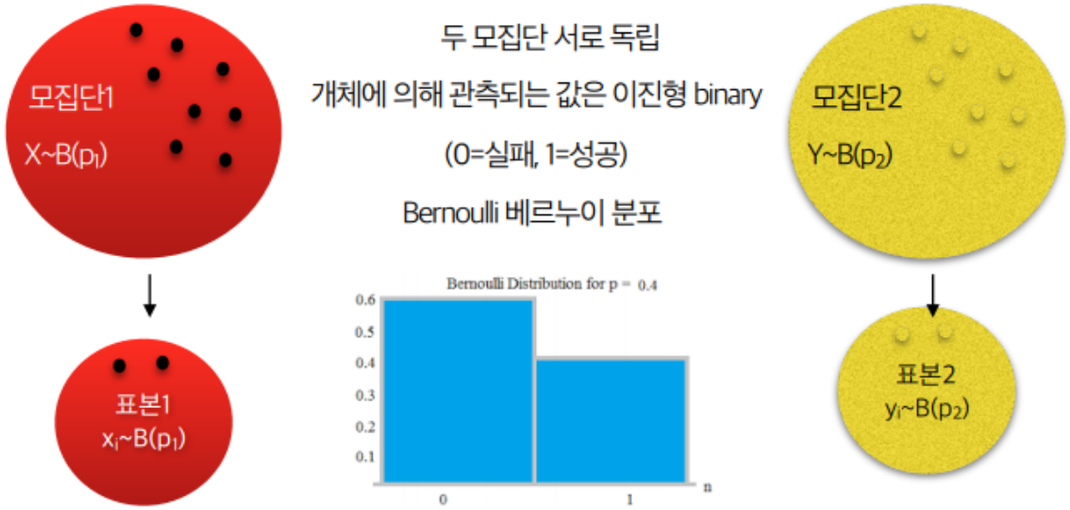
	Finance	Marketing
count	25.000000	25.000000
mean	65438.200000	60373.680000
std	21094.591949	21666.605287

재무 6.54만불 > 마케팅 6.03만불 : 유의확률 0.001보다 작으므로 귀무가설이 기각되어 재무 전공자가 마케팅 전공자에 비해 초봉이 높다.

```
#two paired two population means
import pandas as pd
url='http://wolfpack.hnu.ac.kr/Stat_Notes/elem_stat/Stat_methods/mba2.csv'
mba=pd.read_csv(url)
mba.head(3)

from scipy import stats
stats.ttest_rel(mba['Finance'],mba['Marketing'])

mba[['Finance','Marketing']].describe()
```

2독립집단 모비율차이

개념



확률표본 (iid) = 서로 독립이고 동일분포에서 추출

$$x_i \sim B(p_1) \qquad y_i \sim B(p_2)$$

$x_1 = 0, x_1 = 1, x_3 = 1, \ldots, x_n = 0$ 데이터 $y_1 = 1, y_2 = 1, y_3 = 0, \ldots, y_m = 0$

점추정치 (MVUE) : $\hat{p}_1 = \frac{\#of1's\_sample1}{n}, \hat{p}_2 = \frac{\#of1's\_sample2}{m}$

신뢰구간

$$(\hat{p}_1 - \hat{p}_2) \pm z_{1-\alpha/2}\sqrt{\frac{\hat{p}_1(1-\hat{p}_1)}{n} + \frac{\hat{p}_2(1-\hat{p}_2)}{m}}$$

가설검정 $H_0 : p_1 = p_2$

$$TS = \frac{\hat{\theta} - \theta}{s(\hat{\theta})} = \frac{\hat{p}_1 - \hat{p}_2 - 0}{\sqrt{(\hat{p}_1(1-\hat{p}_2))/(1/n + 1/m)}} \sim N(0,1), \quad \hat{p} = \frac{\#of1's\_S1 + \#of1's\_S2}{n+m}$$

예제 데이터

세미소사와 맥과이어 홈런 경쟁으로 인하여 여성 팬이 증가하였다고 주장한다. 이를 알아보기 위하여 CNN/ USA이 다음 조사를 하였다. 1995년 1008명 여성 중 413이 팬이라고 대답했고, 홈런 경쟁이 있은 1998년에는 1082 여성 중 681명이 팬이라고 답하였다. 이를 이용하여 주장에 대해 답하시오.

```
import numpy as np
from statsmodels.stats.proportion import proportions_ztest
count = np.array([413,681]); nobs = np.array([1008,1082])
proportions_ztest(count, nobs)
```

```
(-10.047201272460136, 9.451353157591624e-24)
```

```
phat=sum(count)/sum(nobs)
phat1=count[0]/nobs[0]; phat2=count[1]/nobs[1]
print('통합비율= %.2f, 1995년 비율=  %.2f, 1998년 비율=  %.2f' %(phat,phat1,phat2))
```

```
통합비율= 0.52, 1995년 비율=  0.41, 1998년 비율=  0.63
```

유의확률 <0.001 -> 귀무가설 기각, 1998년 여성팬 비율이 63%으로 1995년 41%보다 유의적 증가, 홈련경쟁이 여성 팬 유입 효과가 있음

```
#two independent two population means
import numpy as np
from statsmodels.stats.proportion import proportions_ztest
count = np.array([413,681]); nobs = np.array([1008,1082])
proportions_ztest(count, nobs)

phat=sum(count)/sum(nobs)
phat1=count[0]/nobs[0]; phat2=count[1]/nobs[1]
print('통합비율= %.2f, 1995년 비율=  %.2f, 1998년 비율=  %.2f'
%(phat,phat1,phat2))
```

2짝진집단 모비율차이

개념 McNemar 검정

동일한 개체로부터 이진형(성공, 실패) 변수를 서로 다른 기간(before - after)에 측정하여 프로그램 효과가 있는지 알아보는 방법이다. Bland (2000) 1319명 어린이, 12살에 독감에 걸릴 가능성은 나이가 14살이 되면 높아지는 지 낮아지는지 알아보기 위하여 조사한 결과 이다.

Severe colds at age 12	Severe colds at age 14				Total
	Yes		No		
Yes	212	A	144	B	356
No	256	C	707	D	963
Total	468		851		1319

점추정치 (MVUE) : $\hat{p}_A = (356/1319) = 0.354, \hat{p}_D = 144/1319 = 0.27$

검정통계량 : $TS = \frac{(B-C)^2}{B+C} \sim asymtotic\chi^2(1)$

귀무가설 : $H_0 : p_1 = p_2$ (12살 독감 걸릴 확률은 14살 독감 걸릴 확률 동등)

유의확률<0.001이므로 귀무가설 기각, 12살 독감 걸릴 확률 35.4%, 14살 27%로 나아짐 - 12살에 독감이 걸리면 14살에 다시 걸릴 확률이 낮아짐

```
# Example of calculating the mcnemar test
from statsmodels.stats.contingency_tables import mcnemar
table = [[212, 144],[256, 707]]
# calculate mcnemar test
result = mcnemar(table, exact=True)
# summarize the finding
print('statistic=%.3f, p-value=%.3f' % (result.statistic, result.pvalue))
```

```
statistic=144.000, p-value=0.000
```

# Example of calculating the mcnemar test
from statsmodels.stats.contingency_tables import mcnemar
table = [[212, 144],[256, 707]]
# calculate mcnemar test
result = mcnemar(table, exact=True)
# summarize the finding
print('statistic=%.3f, p-value=%.3f' % (result.statistic, result.pvalue))