Py 기본분석: 기술통계량

I. 기술통계량 계산

II. 평균의 계산

Statistics Methods

Method	Description
statistics.harmonic_mean()	Calculates the harmonic mean (central location) of the given data
statistics.mean()	Calculates the mean (average) of the given data
statistics.median()	Calculates the median (middle value) of the given data
statistics.median_grouped()	Calculates the median of grouped continuous data
statistics.median_high()	Calculates the high median of the given data
statistics.median_low()	Calculates the low median of the given data
statistics.mode()	Calculates the mode (central tendency) of the given numeric or nominal data
statistics.pstdev()	Calculates the standard deviation from an entire population
statistics.stdev()	Calculates the standard deviation from a sample of data
statistics.pvariance()	Calculates the variance of an entire population
statistics.variance()	Calculates the variance from a sample of data

1. 자료의 중심

- 평균(산술평균) : mean 함수를 이용하여 계산
- 중위수 : median 함수를 이용하여 계산
- 최빈값 : mode 함수를 이용하여 계산

b1-ch3-9.py

```
import numpy as np
import pandas as pd
ch2_1 = np.genfromtxt("http://kanggc.iptime.org/book/data/ch2_1.txt")
ch2_1
print("Data of ch2_1 is : ",f'\n{ch2_1}\n')

ch2_1_p = pd.Series(ch2_1.flatten())
ch2_1_p
print("Data of ch2_1_p is : ",f'\n{ch2_1_p}\n')

import statistics   # statistics package를 이용
m_ch2_1 = statistics.mean(ch2_1_p)
me_ch2_1 = statistics.median(ch2_1_p)
mo_ch2_1 = statistics.mode(ch2_1._p)
var_ch2_1 = statistics.variance(ch2_1_p)
sd_ch2_1 = statistics.stdev(ch2_1_p)
ch2_1_p.describe()

print("Mean of ch2_1 is :", m_ch2_1)
print("Median of ch2_1 is :", me_ch2_1)
print("Mode of ch2_1 is :", mo_ch2_1)
```

```
Mean of ch2_1 is : 64.76
Median of ch2_1 is : 65.0
Mode of ch2_1 is : 85.0
```

2. 자료의 변화량 및 요약통계량 및 히스토그램

- 분산 : var 함수를 이용하여 계산 또는 편차 및 변동을 먼저 구하고 변동을 자유도(=자료 -1)로 나눔
- 표준편차 : sd 함수를 이용하여 계산

b1-ch3-9.py

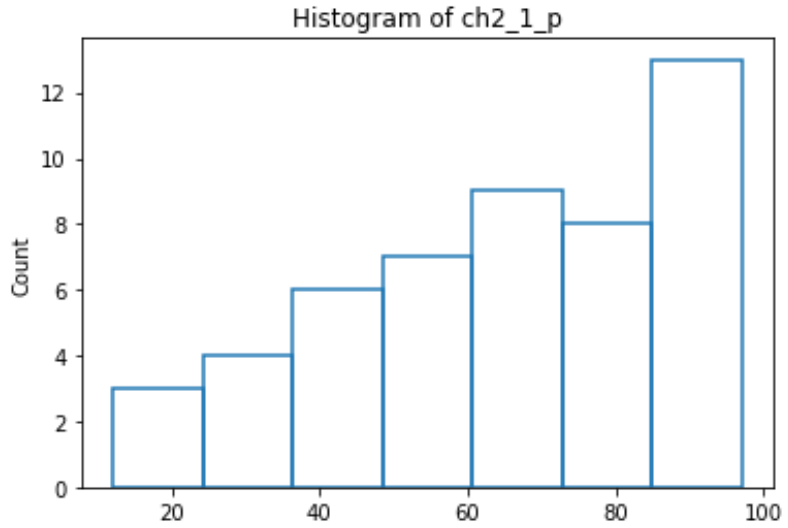
```
(앞에서 계속)
print("Variance of ch2_1 is : ", var_ch2_1)
print("Standard Deviation of ch2_1 is : ", sd_ch2_1)
print("Summary Statistics of ch2_1_p is : ", f'\n{ch2_1_p.describe()}\n')


import seaborn as sns # seaborn package를 이용
sns.histplot(ch2_1_p, fill=False).set(title='Histogram of ch2_1_p')
```

```
Variance of ch2_1 is :  490.88
Standard Deviation of ch2_1 is :  22.155811878601966
```

```
Summary Statistics of ch2_1_p is :
count    50.000000
mean     64.760000
std      22.155812
min      12.000000
25%      48.250000
50%      65.000000
75%      84.500000
max      97.000000
```

3. 히스토그램



II. 평균의 계산

1. 산술평균

- 산술평균 : mean 함수를 이용하여 계산

2. 조화평균(harmonic mean : H)

$$\frac{1}{H}=\frac{1}{n}\left(\frac{1}{X_1}+\frac{1}{X_2}+\cdots..+\frac{1}{X_n}\right)=\frac{\sum_{i=1}^{n}\frac{1}{X_i}}{n}$$

- 일반적으로 역수의 형태로 된 변수를 평균할 경우 사용
- n개의 양수인 변수의 조화평균은 그 변수의 역수를 산술평균한 것의 역수
- (예) 자동차가 처음 10km를 시속 30km로 달리고, 다음 10km를 시속 60km로 달렸을 경우 평균시속은?

- 산술평균 : $\frac{1}{2}(30+60)=45km/h$
- 조화평균 : $\frac{2}{\frac{1}{30}+\frac{1}{60}}=40km/h$
- 20km를 가는데 30분이 소요되는데 평균시속이 45km/h이면 20km를 가는데 30분이 소요되지 않고, 40km/h이면 30분이 소요됨

3. 기하평균(geometric mean : G)

$$G=\sqrt[n]{X_1X_2\cdots..X_n}$$

- 여러 개의 수를 연속으로 곱해 그 개수의 거듭제곱근으로 구한 수
- 비율을 평균하는 경우에 적합하며, 연평균 인구증가율이나 연평균 경제성장률을 구할 때 활용
- 최초년도와 최종년도의 데이터만 있으면 연평균 증가율 계산이 가능

연도	인구	전년대비비율(%)	$\overline{X}$ × 전년도인구	G × 전년도인구
2014	5000	–	–	–
2015	6000	120	6133	6128
2016	7800	130	7523	7510
2017	9204	118	9228	9204
–	–	–	$\overline{X}=122.67$	$G=122.56$

$$G=\left(\left(\frac{\text{최종년도의 값}}{\text{최초년도의 값}}\right)^{(1/n)}-1\right)\times 100,\ \text{n은 경과년도}$$

b1-ch3-10.py

```
statistics as s
import statistics
import math
x = [10, 2, 19, 24, 6, 23, 47, 24, 54, 77]
x
print("x list is : ", f'\n{x}\n')
xi = [1/10, 1/2, 1/19, 1/24, 1/6, 1/23, 1/47, 1/24
, 1/54, 1/77]
print("1/x list is : ", f'\n{xi}\n')
xg = math.prod(x)
print("Product of x is :", xg)

mean_x = sum(x)/len(x)
print("Mean is :", mean_x)
harmonic_x = (1/(sum(xi)/len(xi)))
print("Harmonic Mean is :", harmonic_x)
geometric_x = xg**(1/len(x))
print("Geometric Mean is :", geometric_x)

x_m = statistics.mean(x)
x_h = statistics.harmonic_mean(x)
x_g = statistics.geometric_mean(x)

print("Mean of x is :", x_m)
print("Harmonic Mean of x is :", x_h)
print("Geometric Mean of x is :", x_g)
```

```
Mean is : 28.6
Harmonic Mean is : 10.011092620317164
Geometric Mean is : 18.928088197760175
```

```
Mean of x is : 28.6
Harmonic Mean of x is : 10.011092620317163
Geometric Mean of x is : 18.92808819776017
```

b1-ch3-10.py

(안에서 계속)

```
b = [30, 60]
b_m = s.mean(b)
b_h = s.harmonic_mean(b)
print("Mean of b is :", b_m)
print("Harmonic Mean of b is :", b_h)

c = [120, 130, 118]
c_m = s.mean(c)
c_g = s.geometric_mean(c)
print("Mean of c is :", c_m)
print("Geometric Mean of c is :", c_g)

g = ((9204/5000)**(1/3)-1)*100
print("compound annual growth rate(CAGR) of c is :", g)
```

```
Mean of b is : 45
Harmonic Mean of b is : 40.0
```

```
Mean of c is : 122.66666666666667
Geometric Mean of c is : 122.55627013624778
```

```
compound annual growth rate(CAGR) of c is : 22.556270136247768
```