기상청

4. 데이터 탐색과 시각화

Python을 활용한 분석
기초 가이드 북

기상청

Ⅳ. 데이터 탐색과 시각화

1. 개요
2. 히스토그램(Histogram)
3. 기술통계 분석
4. 상자도표(Box plot)
5. 상관관계 분석
6. 산점도(Scatter plot)
7. 기타

1. 개요

데이터를 파악하기 위해 히스토그램, 기술통계, 상자도표, 상관분석, 산점도 등을 적용하여 정보를 확인할 수 있습니다. 이러한 데이터 탐색 과정은 데이터의 신뢰성, 필요한 데이터 정제의 정도, 모형 구축 시 활용에 적합한 변수 탐색 등을 알 수 있게 하고 전처리 과정과 함께 매우 중요한 부분입니다.

● 히스토그램(Histogram)

• 표로 되어 있는 도수 분포를 시각적으로 나타낸 것입니다. 일반적으로 히스토그램의 가로축은 속성값(계급, 범주 등)을 뜻하고, 세로축은 그 빈도를 뜻합니다. 때때로 축을 반대로 하여 나타내기도 합니다.

● 기술통계 분석

• 기술통계란 측정이나 실험에서 수집한 자료의 정리, 표현, 요약, 해석 등을 통해 자료의 특성을 규명하는 통계적 방법입니다. 자료의 특성을 표현하는 지표로 대푯값(평균값, 중앙값, 최빈값), 산포도(분산, 표준편차, 범위, 사분위수, 평균편차, 표준오차, 변이계수), 왜도 및 첨도가 있습니다.

● 상자 도표(Box plot)

• 상자도표는 5개의 통계(최소값, 첫 번째 사분위수, 중앙값, 두 번째 사분위수, 최대값)를 시각적으로 나타냅니다. 변수가 가진 값의 분포와 함께 이상치가 표시되어 있어 자료의 중심과 흩어진 정도를 살펴 볼 수 있습니다.

● 상관관계 분석

• 두 개의 변수 간 어떤 선형적 관계를 가지고 있고, 그 관계의 강도는 어떠한지를 분석하는 방법입니다. 상관분석으로 도출되는 상관계수는 -1 부터 1 사이의 값을 가지며 + 부호인 경우는 정적 상관관계(positive relationship, x가 커질수록 y도 커짐)를 나타내고,
- 부호가 있으면 부적 상관관계(negative relationship, x가 커질수록 y는 작아짐)를 나타냅니다. 상관계수의 절대값이 1에 가까울 수록 관련성이 깊은 것을 나타냅니다.

● 산점도(scatter plot)

• 산점도는 주어진 데이터를 점으로 표현하여 시각적으로 나타냅니다. 데이터의 실제 값들의 분포를 파악하는데 유용한 방법입니다.

2. 히스토그램(Histogram)

히스토그램을 출력하기 위해 보통 matplotlib 패키지를 이용합니다. 최근 matplotlib을 기반으로 하는 파이썬 시각화 라이브러리인 seaborn을 이용하여 가독성이 좋은 디자인의 표/그래프를 출력할 수 있습니다.

- **matplotlib 를 이용한 히스토그램**

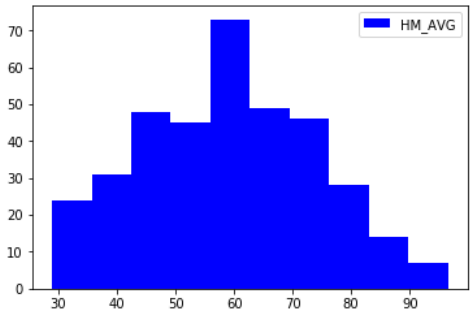
[구조]

```
import matplotlib.pyplot as plt


plt.hist(데이터명.변수명, 옵션)
plt.legend() # 변수명을 그래프에 표기
```

[예시]

```
import pandas as pd
import matplotlib.pyplot as plt
mydata = pd.read_csv('mydata.csv')

plt.hist(mydata.HM_AVG, color='b',
label='HM_AVG')
plt.legend()
```

[예시의 결과: 평균 상대습도(HM_AVG)]



- **seaborn 를 이용한 히스토그램**

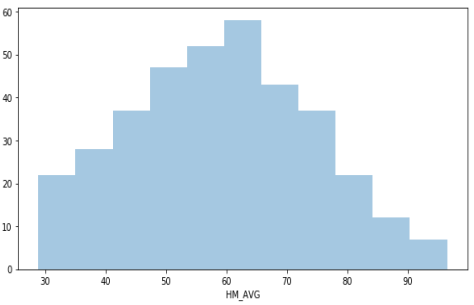
[구조]

```
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(가로 길이,세로 길이))
sns.distplot(데이터명.변수명)
```

[예시]

```
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(10,5))
sns.distplot(mydata.HM_AVG, kde =
False) # kde: 가우시안 확률밀도 여부
```

[예시의 결과: 평균 상대습도(HM_AVG)]



3. 기술통계 분석

데이터의 기본적인 통계량은 numpy, pandas, scipy 패키지를 이용해 출력할 수 있습니다.

- **numpy 를 이용한 기술통계 분석**

[구조] * x는 array

```
import numpy as np

np.mean(x) # 평균
np.var(x) # 분산
np.str(x) # 표준편차
np.max(x) # 최대값
np.min(x) # 최소값
np.median(x) # 중앙값
np.percentile(x, 25) # 1사분위수
np.percentile(x, 50) # 2사분위수
np.percentile(x, 75) # 3사분위수

# x 에 데이터명 또는 변수명 입력
```

[예시]

```
import numpy as np

# 데이터에 있는 모든 변수의 결과
np.var(mydata)

# 데이터 특정 변수(x1)의 평균
np.mean(mydata['x1']) # 평균
```

- **pandas 를 이용한 기술통계 분석**

[구조]

```
import pandas as pd

데이터명.describe()
```

[예시]

```
import pandas as pd

s = pd.DataFrame(mydata)
s.describe()
```

- **SciPy 를 이용한 기술통계 분석**

[구조]

```
import scipy as sp

sp.stats.describe(데이터명)
```

[예시]

```
import scipy as sp

sp.stats.describe(mydata)
```

4. Box plot

박스플롯을 출력하기 위해 matplotlib 패키지 중 pyplot 이라는 모듈을 이용할 수 있습니다.

- **matplotlib 를 이용한 Box plot**

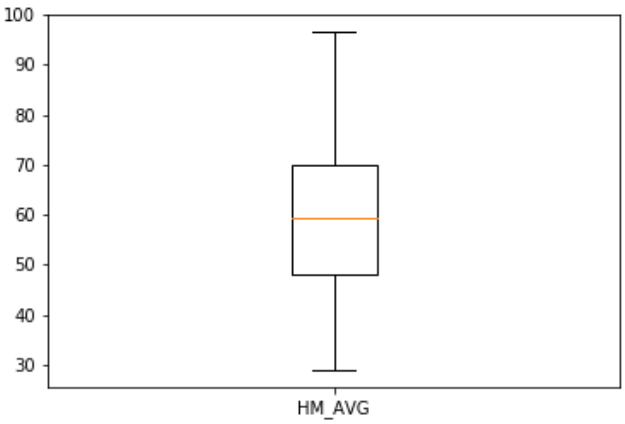
[구조]

```
import matplotlib.pyplot as plt


데이터명[['변수명']].plot(kind='box)
plt.xticks([1], ['변수명'])
```

[예시]

```
import pandas as pd
import matplotlib.pyplot as plt

mydata=pd.read_csv('mydata.csv')
#데이터 로드
plt.boxplot(mydata.HM_AVG)
plt.xticks([1], ['HM_AVG'])
```

[예시의 결과: 평균 상대습도]



5. 상관관계 분석

상관분석 결과를 출력하기 위해 numpy 패키지와 pandas 패키지를 이용합니다.

- **numpy 를 이용한 상관분석**

[구조]

```
import numpy as np

np.corrcoef(변수명1,변수명2)
```

[예시]

```
import numpy as np

a=np.array([1,2,3,4,5])
b=np.array([3,2,3,1,2])

np.corrcoef(a,b)
```

[예시의 결과]

```
array([[ 1.        , -0.56694671],
       [-0.56694671,  1.        ]])
```

- **pandas 를 이용한 상관분석**

[구조]

```
import pandas as pd


데이터명.corr(method='pearson')
```

[예시]

```
import pandas as pd
mydata=pd.read_csv('mydata.csv')

X = mydata.iloc[:,1:4] # 두번째~세번째 컬럼 선택

X.corr(method='pearson')
```

[예시의 결과]

	CA_TOT	HM_AVG	RN_DAY
CA_TOT	1.000000	0.665570	0.595266
HM_AVG	0.665570	1.000000	0.614174
RN_DAY	0.595266	0.614174	1.000000

6. Scatter plot

산점도를 출력하기 위해 matplotlib 패키지의 pyplot 모듈을 이용합니다.

- **matplotlib 를 이용한 Scatter plot**

[구조]

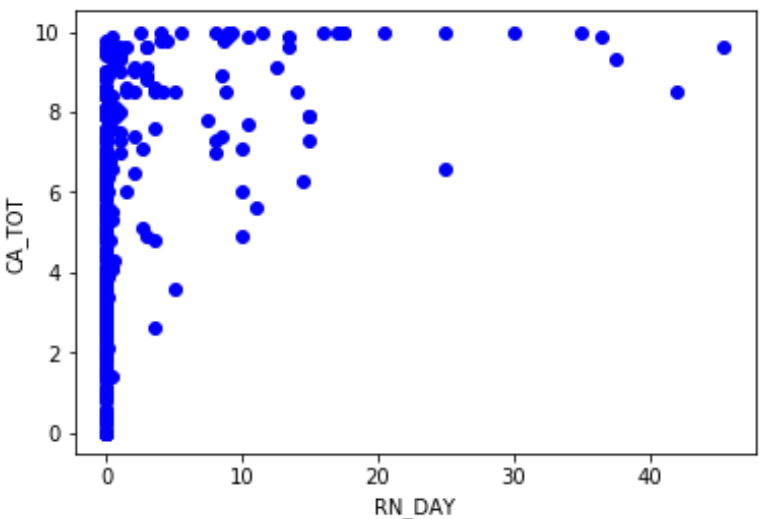
```
import matplotlib.pyplot as plt


plt.scatter('x축 변수명', 'y축 변수명', 옵션)
```

[예시]

```
import matplotlib.pyplot as plt

mydata=pd.read_csv('mydata.csv')
plt.scatter(mydata['RN_DAY'],
mydata['CA_TOT'],color='b', marker='o')
plt.xlabel('RN_DAY')
plt.ylabel('CA_TOT')
```

산점도 옵션	설명							
label	그래프 레이블을 입력합니다.							
size	도형의 크기를 입력합니다.							
alpha	도형 색상의 투명도를 입력합니다(0이면 투명, 1이면 불투명).							
color	그래프에 이용할 색상을 입력합니다.							
marker	표시할 도형 종류를 입력합니다.							
	'.'	'o'	'^'	'x'	'D'	'P'	'*'	'+'
	점	원	삼각형	엑스형	다이아몬드	오각형	별	덧셈모양

[예시의 결과: 강수량과 전운량]



7. 기타 (1/2)

데이터 탐색 및 전처리 시 유용한 기타 파이썬 코드입니다.

- **상위 또는 하위 행 선택해 출력하기**

[구조]

```
import pandas as pd

mydata.head()     # 상위 행 보기
mydata.tail()     # 하위 행 보기
```

- **결측값 처리**

[구조]

```
import pandas as pd

mydata.fillna(999)   # 데이터에서 결측값은 999로 치환
mydata.dropna()      # 데이터의 결측값 제외
mydata.isnull()      # 데이터에서 결측인지
mydata.notnull()     # 데이터에서 결측이 아닌지
```

- **데이터 타입 변환**

[구조]

```
# 데이터 타입 변환
데이터명[ " 변수명 " ]=데이터명[ " 변수명 " ].astype('변환할 타입')
```

[예시]

```
# target 변수를 string 타입으로 변환
mydata["target"]=mydata["target"].astype('category')

# 데이터 타입 보기
mydata.dtypes
```

- **선형, 누적, 바 그래프**

[구조]

```
데이터명.변수명.plot.line() # 선형

데이터명.변수명.plot.bar() # 바

데이터명.변수명.plot.density() # 누적
```

[예시]

```
mydata.x1.plot.line()

mydata.x1.plot.bar()

mydata.x1.plot.density()
```

7. 기타 (2/2)

데이터 탐색 및 전처리 시 유용한 기타 파이썬 코드입니다.

- **행 병합**

[구조]

```
import pandas as pd

pd.concat([데이터명1, 데이터명2],axis=0)
```

[예시]

```
import pandas as pd

pd.concat([mydata1, mydata2],axis=0)
```

- **열 병합**

[구조]

```
import pandas as pd

pd.concat([데이터명1, 데이터명2],axis=1)
```

[예시]

```
import pandas as pd

pd.concat([mydata1,mydata2],axis=1)
```

- **데이터 병합**

[구조]

```
import pandas as pd

pd.merge(왼쪽 위치 데이터명, 오른쪽
위치 데이터명, on='고유키',how="left")
#how 옵션 종류: left, right, outer, inner
```

[예시]

```
import pandas as pd

pd.merge(mydata1, mydata2,
on='key',how="left")
```

- **특정 조건에 따른 데이터 변환**

[구조]

```
import numpy as np
import pandas as pd

데이터명["생성할 컬럼명"] =
np.where(데이터명[ ' 참조
컬럼명']==조건 값, 'True일 경우 값',
'False일 경우 값')
```

[예시]

```
import numpy as np

mydata.target.unique()
mydata["new"] =
np.where(mydata['target']==0.0, 'up',
'down')
mydata.head()
```

본 문서의 내용은 기상청의 날씨마루(http://big.kma.go.kr) 내
Python 기초 교육 자료 입니다.