빅분기 실기_개념정리

<1유형>데이터 다루기

 # Pandas 함수#

- **groupby 함수**

`변수명 = df.groupby( '분류기준 1' )[ '분류기준2' ].함수( )`

- **select_dtypes 함수**

`df.select_dtypes(include='데이터타입_종류')`

- **str.contains 함수**
-
- 빅분기 시험환경에서는 df.head()함수도 print 적용해야 보임

  `ex) print(df.head())`

==========

<2유형>데이터 모델링

**** < base line code> ****

1. 데이터 탐색(EDA)

   `df.head()`

   `df.shape()`

   `df.info()`

   `df.describe()`

2. 이상치, 결측치, 피처변수 처리
3. ##결측치처리 : 범주형(최빈값-mode), 연속형(중앙값-median) ##

3. 데이터 분할(train_test_split) ⇒ 분류 문제에서 target변수(y) 층화추출 꼭 해주기

   ###(stratify = y)

4. 모델링(랜덤포레스트 분류 or회귀)

   `from sklearn.ensemble import RandomForestClassifier / RandomForestRegressor`

5. (선택) # 하이퍼 파라미터 최적화(그리드 서치)

   `from sklearn.model_selection import GridSearchCV`

6. 답안 제출 (filiename.to_csv('xxxxx.csv', index=False)로 제출, read_csv()로 확인)

==========

<3유형>가설검정과 회귀분석

1. 가설검정

<#1>모집단 1개 - 단일표본 t검정(1 sample t-test) (집단의 평균, 특정 값)

Q1. mtcars 데이터셋의 mpg열의 데이터 평균이 20과 같다고 할수있는지 검정하시오. (유의수준 5%)

<가설설정>

H0 : mpg열의 평균이 20과 같다.

H1 : mpg열의 평균이 20과 같지 않다.

#t-test전에 정규성 검정(shapiro)검정 ⇒ 정규성 만족시 1samp_ttest ,

⇒ 정규성 만족x시 wilcoxon 부호순위검정 실시

##정규성 검정##

H0 : 정규분포를 따른다.

H1 : 정규분포를 따르지 않는다.

```
import scipy.stats as stats
from scipy.stats as shapiro
```

```
statistic, pvalue = stats.shapiro(df['mpg'])
print(round(statistic, 4), round(pvalue, 4))
```

`#statistic=0.9476, p_value=0.1229` 이므로

귀무가설 채택(정규분포를 따른다.)

(**정규성** 만족) 단일표본 t 검정 실시 ⇒ (**ttest_1samp**)

```
statistic, pvalue = stats,ttest_1samp(df['mpg'], popmean=20, alternative ='two-sided'  )
print(round(statistc,4) , round(pvalue, 4))
```

`#statistic=0.0851, pvalue=0.7891` 이므로

귀무가설 채택 (mpg열의 평균이 20과 같다고 할 수 있다.)

(**정규성** 만족x) 윌콕슨 부호순위검정 실시 ⇒ (**wilcoxon**)

```
statistic, pvalue = stats.wilcoxon(df['mpg'] - 17, alternative = 'two-sided')
print(round(statistic, 4), round(pvalue,4))
```

`#statistic=249.0, pvalue=0.7891` 이므로

귀무가설 채택 (mpg열의 평균이 20과 같다고 할 수 있다.)

<#2>모집단 2개

1. (정규성 O) 대응표본(쌍체) t-검정 ⇒ paired t-test (stats.ttest_rel)

(정규성 X)윌콕슨 부호순위 검정 ⇒ wilcoxon

2. (정규성 O) 독립표본 t검정 ⇒ 2sample t-test (stats.ttest_ind)

(정규성 X)윌콕슨 순위합 검정(ranksums)

1. **대응 표본(쌍체) t검정** 시 **주의점**

- **정규성 검정** (**차이값**에 대한 정규성 검정) ⇒ `stats.shapiro(df['after'] - df['before'])`

  `H0 : after - before = 0`

  `H1 : after - before ≠ 0`

2. **독립표본 t검정** 시 **주의점**

- **정규성 검정**(**두 집단 모두 정규성을 따르는지**) ⇒ `stats.shapiro(['A']) , stats.shapiro(['B'])`

  등분산 여부 확인 필수 ⇒ `stats.bartlett(df['A'], df['B'])`

  (등분산 만족O) `stats.ttest_ind(df['A'], df['B'], equal_var=True , alternative='two-sided')`

  (등분산 만족X) `stats.ttest_ind(df['A'], df['B'], equal_var=False , alternative='two-sided')`

<#3>모집단 3개 이상- 분산분석(ANOVA) : A집단 vs B집단 vs C집단.....

(정규성 만족o) - **ANOVA 분석 (stats.f_oneway) # 데이터가 각각 들어가야 함**

(정규성 만족x) - 크루스칼-왈리스 검정(kruskal - wallis test)

****등분산 여부 확인**

#예시

다음 A,B,C 그룹의 성적 평균이 같다고 할 수 있는지 ANOVA분석을 실시하시오.

H0 : A=B=C(세 그룹의 평균이 모두 같다.)

H1 : Not H0 (적어도 하나는 같지 않다.)

<#3> 카이제곱 검정

1. 적합도 검정 - 각 범주에 속할 확률이 같은지 - **chisquare()**
2. 독립성 검정 - 두 개의 범주형 변수가 서로 독립인지 - **chi2_contingency()**

1-1. 적합도 검정 예시 문제 - example_1

랜덤 박스에 상품이 들어있다. 다음은 랜덤박스 에서 100번 상품을 꺼냈을 때의 상품 데이터라고 할 때,

상품이 동일한 비율로 들어있다고 할 수 있는지 검정해보시오. (유의수준 5%)

```
import pandas as pd
import numpy as np
```

```
#데이터 생성
rowl = [30, 20, 15, 35]
df = pd.DataFrame([rowl], columns=['A','B','C','D'])
```

#가설설정

H0 : 랜덤박스에 상품 A,B,C,D가 동일한 비율로 들어있다.

H1 : 랜덤박스에 상품 A,B,C,D가 동일한 비율로 들어있지 않다.

```
from scipy.stats as import chisquare
```

```
f_obs = [30, 20, 15, 35]
f_exp = [25, 25, 25, 25]    #관측빈도와 기대빈도 구하기
```

```
statistic, pvalue = chisquare(f_obs=f_obs, f_exp=f_exp)
print(statistic, pvalue)
#statistic = 10.0, p_value= 0.0185661354….로 귀무가설 기각(대립가설 채택)
```

1-2. 적합도 검정 예시 문제 - example_2

랜덤 박스에 상품이 들어있다. 다음은 랜덤박스 에서 150번 상품을 꺼냈을 때의 상품 데이터라고 할 때,

상품별로 A: 30%, B:15%, C:55% 비율로 들어있다고 할 수 있는지 검정해보시오. (유의수준 5%)

```
import pandas as pd
import numpy as np

row1 = [50,25,75]
df = pd.DataFrame([row1], columns=['A','B','C'])
```

#가설설정

H0 : 랜덤박스에 상품 A, B, C가 30%, 15%, 55% 의 비율로 들어있다.

H1 : 랜덤박스에 상품 A, B, C가 30%, 15%, 55% 의 비율로 들어있지 않다.

```
#검정실시
from scipy.stats import chisquare

f_obs = [50, 25, 75]
a = 150*0.3
b = 150*0.15
c = 150*0.55

f_exp = [a, b, c]

statistic, pvalue = chisquare(f_obs=f_obs, f_exp=f_exp)
print(statistic, pvalue)
# statistic = 1.5151515151…. , p_value = 0.4688015…..
#귀무가설 채택(대립가설 기각)
```

2-1. 독립성 검정 예시 문제 - example_1

연령대에 따라 먹는 아이스크림의 차이가 있는지 독립성 검정을 실시하시오.

```
import pandas as pd
import numpy as np
```

```
row1, row2 = [200, 190, 250],  [220, 250, 300]
df = pd.DaraFrame([row1, row2], columns=['딸기', '초코', '바닐라'], index = ['10대', '20대'])
```

#가설설정

H0 : 연령대와 먹는 아이스크림의 종류는 서로 관련이 없다(두 변수는 서로 독립이다)

H1 : 연령대와 먹는 아이스크림의 종류는 서로 관련이 있다(두 변수는 서로 독립이 아니다)

```
#검정실시
from scipy.stats import chi2_contingency
statistic, pvalue, dof, expected = chi2_contingency(df)
```

```
print(statistic, pvalue, dof, np.round(expected, 2))
#1.708360….
0.4256320……
2
[[190.64  199.72  249.65]
[229.36  240.28  300.35]]
```

cf) 만약 데이터 형태가 다른 경우 - `pd.crosstab()` 사용

```
df = pd.DataFrame({
'아이스크림' : ['딸기', '초코', '바닐라', '딸기', '초코', '바닐라'],
'연령' : ['10대', '10대', '10대', '20대', '20대', '20대'],
'인원' : [200,190,250,220,250,300]
})
```

```
# pd.crosstab(index=   , columns=    , values=      , aggfunc=sum)
```

```
table = pd.crosstab(index=df['연령'], columns=df['아이스크림], values=df['인원'], aggfunc=sum)
```

```
from scipy.stats import chi2_contingency
```

```
statistic, pvalue, dof, expected = chi2_contingency(table)
print(statistic, pvalue, dof, expected)
```

```
#1.708360….
0.4256320……
2
[[190.64  199.72  249.65]
[229.36  240.28  300.35]]
```

2-2. 독립성 검정 예시 문제 - example_2

타이타닉 데이터에서 성별(sex)과 생존여부(survived) 변수간 독립성 검정을 실시하시오.

```
import seaborn as sns
df = sns.load_dataset('titanic')
```

```
table = pd.crosstab(df['sex], df['survived])
print(table)
```

#가설설정

H0 : 성별과 생존여부는 서로 관련이 없다(두 변수는 서로 독립이다)

H1 : 성별과 생존여부는 서로 관련이 있다(두 변수는 서로 독립이 아니다)

```
# 검정실시(통계량, p-value, 기대빈도 확인)
from scipy.stats import chi2_contingency
statistic, pvalue, dof, expected = chi2_contingency(table)
print(statistic, pvalue, dof, np.round(expected))
```

```
#260.71702016732104
#1.1973570627755645e-58
#1
```

```
#[[193.47  120.53]
  [355.53  221.47]]
```