+3과목 가설검정, 회귀분석

##가설검정-모의고사

```
# 모의고사 3유형
# 가설검정
# 문제 1
# 다음은 A그룹과 B그룹 인원의 키 데이터이다.
# 두 그룹의 키 평균이 다르다고 할 수 있는지
# 가설검정을 실시하고자 한다. 아래 물음에 답하시오(유의수준 1%)
#  In [1]:
#  A : A그룹 인원의 키 평균, B : B그룹 인원의 키 평균
# H0(귀무가설) : A = B
#  H1(대립가설) : A ≠ B
```

```
###############  실기환경 복사 영역  ###############
import pandas as pd
# 데이터 생성
data = {
    'Height': [172, 175, 173, 174, 177, 170, 169, 178, 171, 1
               168, 165, 171, 169, 170, 174, 171, 171, 168, 1
    'Group': ['A'] * 10 + ['B'] * 10
}
df = pd.DataFrame(data)
###############  실기환경 복사 영역  ###############
```

```
# 문제 1-1

# 독립표본 t검정을 실시하고 검정통계량, p-value 값을 구하시오.
# 정답은 반올림하여 소수점 셋째자리까지 구하시오
# (단, 두 그룹의 데이터는 정규성은 만족하지만, 등분산성은 만족하지 못한다고
# 문제 1-2
# 귀무가설 기각여부를 결정하시오(답은 채택 또는 기각으로 작성하시오)
```

```
group_a = df[df['Group'] == 'A']['Height']
group_b = df[df['Group'] == 'B']['Height']
```

```
import scipy.stats as stats

statistic, p_value, = stats.ttest_ind(group_a, group_b, equal

print(statistic, p_value)  #기각
```

```
2.4742269248601123 0.02354414015156895   # 귀무가설 기각(대립가설 채택)
```

```
# 문제 2

# 다음은 A, B, C 그룹 인원의 영어 성적 데이터이다.
# 세 그룹의 성적 평균이 같다고 할 수 있는지 ANOVA 분석을 실시하시오.
#  (유의수준 5%)
#  In [2]:
#  A, B, C : 각 그룹 인원의 성적
# H0(귀무가설) : A(평균) = B(평균) = C(평균)
#  H1(대립가설) : Not H0 (적어도 하나는 같지 않다)
```

```
###############  실기환경 복사 영역  ###############
import pandas as pd
# 데이터 생성
data = {
    'English_Score': [82, 84, 83, 85, 86, 87, 85, 84, 86, 88,
                      78, 77, 79, 76, 75, 77, 78, 79, 80, 76,
                      81, 83, 82, 84, 85, 83, 84, 82, 85, 86]
    'Group': ['A'] * 10 + ['B'] * 10 + ['C'] * 10
}
df = pd.DataFrame(data)
###############  실기환경 복사 영역  ###############
```

```
df.head(10)
```

```
# 문제 2-1

#  ANOVA 분석을 실시하고 검정통계량, p-value 값을 구하시오.
# 정답은 반올림하여 소수점 셋째자리까지 구하시오
# (단, 각 그룹의 데이터는 정규성, 등분산성을 만족한다고 가정한다)
# 문제 2-2
# 귀무가설 기각여부를 결정하시오(답은 채택 또는 기각으로 작성하시오)
```

```
a = df[df['Group'] == 'A']['English_Score']
b = df[df['Group'] == 'B']['English_Score']
c = df[df['Group'] == 'C']['English_Score']
```

```
statistic, p_value = stats.f_oneway(a,b,c)

print(statistic, p_value) #귀무가설 기각(대립가설 채택)
```

56.7 2.1575201522991445e-10

```
print(a.mean())
print(b.mean())
print(c.mean()) #귀무가설 기각(대립가설 채택)
```

85.0
77.5
83.5

```
# 문제 3
# 어느 그룹에서 성별에 따라 선택한 스포츠가 관련성이 있는지
# 검정해보고자 한다. 두 변수(Sport, Gender)의 독립성 검정을 실시하시오.
#  (유의수준 5%)
#  . Gender : Male, Female
#  . Sport : Soccer, Basketball, Swimming
```

```
#  H0(귀무가설) : 두 변수는 서로 독립이다
# H1(대립가설) : 두 변수는 서로 독립이 아니다
```

```
###############  실기환경 복사 영역  ###############
import pandas as pd
# 데이터 생성
data = {
    'Sport': ['Soccer'] * 35 + ['Basketball'] * 55 + ['Swimmi
    'Gender': ['Male'] * 20 + ['Female'] * 15 + ['Male'] * 30
}
df = pd.DataFrame(data)
###############  실기환경 복사 영역  ###############
```

```
table = pd.crosstab(df['Sport'],df['Gender'])
```

```
# 문제 3-1
#  Male
# 카이제곱검정(독립성검정)을 실시하고 검정통계량, p-value 값을 구하시오.
# 정답은 반올림하여 소수점 셋째자리까지 구하시오
# 문제 3-2
# 귀무가설 기각여부를 결정하시오(답은 채택 또는 기각으로 작성하시오)
```

```
statistic, p_value, ddof, expected = stats.chi2_contingency(t

print(statistic, p_value, ddof, expected)
```

2.8354978354978355 0.24225874791744745 2 [[27.5 27.5]
[17.5 17.5]
[30. 30.]]

##회귀분석-모의고사

```
# 모의고사 3유형
# 다중회귀분석
```

```
# 다음은 당뇨병 진척정도 데이터셋이다. 아래 물음에 답하시오.
```

```
###############  실기환경 복사 영역  ###############
import pandas as pd
import numpy as np
# 실기 시험 데이터셋으로 셋팅하기 (수정금지)
from sklearn.datasets import load_diabetes
# diabetes 데이터셋 로드
diabetes = load_diabetes()
x = pd.DataFrame(diabetes.data, columns=diabetes.feature_name
y = pd.DataFrame(diabetes.target)
y.columns = ['target']
df = pd.concat([y, x], axis=1)
###############  실기환경 복사 영역  ###############
```

```
# 문제3-1.
#  target 칼럼(종속변수)과 상관관계가 높은 독립변수 3개를 구하시오.
#  (단, 상관분석은 피어슨 상관분석으로 진행하시오)

# 문제 3-2.
#  3-1 에서 구한 3개의 독립변수를 가지고 다중회귀분석을 실시하고
# 아래 성능지표 5가지 값들을 구하시오.
# 성능지표 : Rsq(결정계수), Rsq-adj(수정결정계수), MSE, AIC값을 구하시
#  (단, 초기 200개 데이터를 사용하시오)
#  (단, 정답은 반올림하여 소수점 셋째자리까지 구하시오)

# 문제 3-3.
#  3-2 에서 구한 회귀 모델에 나머지 242개의 데이터를 적용하여
# MSE 값을 구하시오.
#  (단, 정답은 반올림하여 소수점 셋째자리까지 구하시오)
```

```
corr = df.corr(method='pearson')
#corr
```

```
abs(corr.loc['target']).sort_values(ascending=False)[1:4]
```

bmi 0.586450
s5 0.565883
bp 0.441482
Name: target, dtype: float64

```
print(df.head())
print(df.shape)
```

target age sex bmi bp s1 s2 \
0 151.0 0.038076 0.050680 0.061696 0.021872 -0.044223 -0.034821

1 75.0 -0.001882 -0.044642 -0.051474 -0.026328 -0.008449 -0.019163

2 141.0 0.085299 0.050680 0.044451 -0.005670 -0.045599 -0.034194

3 206.0 -0.089063 -0.044642 -0.011595 -0.036656 0.012191 0.024991

4 135.0 0.005383 -0.044642 -0.036385 0.021872 0.003935 0.015596

```
      s3         s4         s5         s6
```

0 -0.043401 -0.002592 0.019907 -0.017646

1 0.074412 -0.039493 -0.068332 -0.092204

2 -0.032356 -0.002592 0.002861 -0.025930

3 -0.036038 0.034309 0.022688 -0.009362

4 0.008142 -0.002592 -0.031988 -0.046641

(442, 11)

```
train = df.iloc[0:200]
test = df.iloc[200:442]

x_train = train[['bmi', 's5', 'bp']]
y_train = train['target']
x_test = test[['bmi', 's5', 'bp']]
y_test = test['target']
```

```
import statsmodels.api  as sm

x_train = sm.add_constant(x_train)
```

```
model = sm.OLS(y_train, x_train).fit()

print(model.summary())
```

```
print(round(model.rsquared,3))
print(round(model.rsquared_adj,3))
```

0.443
0.434

```
from sklearn.metrics import mean_squared_error
x_test = sm.add_constant(x_test)
pred = model.predict(x_test)
mse = mean_squared_error(y_test, pred)
print(mse)
```

3122.0611179053885

```
# 로지스틱 회귀분석
# 다음은 유방암 진단 데이터셋이다. 아래 물음에 답하시오.
```

```
###############  실기환경 복사 영역  ###############
import pandas as pd
import numpy as np

# 실기 시험 데이터셋으로 셋팅하기 (수정금지)
from sklearn.datasets import load_breast_cancer
# 유방암 데이터셋 load
cancer = load_breast_cancer()
X, y = load_breast_cancer(return_X_y= True, as_frame=True)
x = pd.DataFrame(X)
y = pd.DataFrame(y)
x = x.iloc[:,1:4]
x.columns = ['mean_texture', 'mean_perimeter','mean_area']
y.columns = ['class']
```

```
df = pd.concat([y, x], axis=1)
###############  실기환경 복사 영역  ###############
```

```
# 문제 3-4.
# 주어진 데이터셋에서 class 칼럼을 target 변수로 하여
# 로지스틱 회귀분석을 실시하고 residual deviance값을 구하시오.
#  (단, 초기 400개의 데이터를 사용하시오)
#  (단, 정답은 반올림하여 소수점 둘째자리까지 구하시오)

# 문제 3-5.
# 문제 3-4의 로지스틱 회귀모형에서 mean_perimeter 변수가 한단위 증가할
# 양성일 오즈가 몇 배 증가하는지 반올림하여 소수점 둘째 자리까지 구하시오.

# 문제 3-6.
#  3-4에서 학습한 모델에 나머지 169개 데이터를 적용하여
# 정확도(accuracy)를 구하시오.
#  (단, 정답은 반올림하여 소수점 둘째자리까지 구하시오)
```

```
df.head()
# df.shape
train = df[:400]
test = df[400:]
```

```
x_train = train.drop(columns='class')
y_train = train['class']
x_test = test.drop(columns='class')
y_test = test['class']

print(x_train.shape, y_train.shape)
print(x_test.shape, y_test.shape)
```

(400, 3) (400,)
(169, 3) (169,)

```
import statsmodels.api as sm
```

```
x_train = sm.add_constant(x_train)
model_glm = sm.GLM(y_train, x_train, family=sm.families.Binom
```

```
summary = model_glm.summary()
print(summary)
```

```
# 문제 3-4.
# 주어진 데이터셋에서 class 칼럼을 target 변수로 하여
# 로지스틱 회귀분석을 실시하고 residual deviance값을 구하시오.
#  (단, 초기 400개의 데이터를 사용하시오)
#  (단, 정답은 반올림하여 소수점 둘째자리까지 구하시오)

# residual deviance(잔차 이탈도) : 177.40
```

```
# 문제 3-5.
# 문제 3-4의 로지스틱 회귀모형에서 mean_perimeter 변수가 한단위 증가할
# 양성일 오즈가 몇 배 증가하는지 반올림하여 소수점 둘째 자리까지 구하시오.
coef = -0.4579
print(round(np.exp(coef),2))
```

```
from sklearn.metrics import accuracy_score
x_test = sm.add_constant(x_test)
pred = model_glm.predict(x_test)

# print(pred.head())
# print(y_test.head())

import numpy as np
pred = np.where( pred>0.5, 1, 0 )

acc = accuracy_score(y_test,pred)
print(round(acc,2))
```

#기출문제 7회 -3유형

```
# 기출문제 7회_제 3유형

# 문제 3-1
#  survived를 종속변수(Y), sex,sibsp,fare 변수를 독립변수(X)로 분석하
#  아래 질문에 답하시오.
#  (단, 총 891개의 데이터에서 초기 500개 데이터로 학습, 391개 데이터로 
#  (단, 모든 답은 반올림하여 소수점 둘째자리까지 구하시오)

# 1. 초기 500개 데이터로 분석시 sibsp 변수의 odds ratio는?
# 2. 초기 500개 데이터로 분석시 residual deviance는?
# 3. 나머지 391개 데이터 적용시 오분류율은?
```

```
######## 데이터 생성(수정금지) ########
# 데이터 불러오기
import pandas as pd

import numpy as np
# Seaborn의 내장 타이타닉 데이터셋을 불러옵니다.
import seaborn as sns
df = sns.load_dataset('titanic')
df = df[ ['survived','sex','sibsp','fare'] ]
# sex:성별, sibsp:탑승한 부모 및 자녀 수, fare:요금
# 성별을 map 함수를 활용해서 각각 1과 0에 할당한다.(여성을 1, 남성을 0)
# (실제 시험의 지시 조건에 따를 것)
df['sex'] = df['sex'].map({'female': 1,
                           'male': 0 })
 ########################################
```

`df.head()`

```
train = df.iloc[ :500, : ]
test = df.iloc[500: , ]

x_train = train.drop(columns='survived')
y_train = train['survived']
```

```
x_test = test.drop(columns='survived')
y_test = test['survived']

print(x_train.shape, y_train.shape)
print(x_test.shape, y_test.shape)
```

(500, 3) (500,)
(391, 3) (391,)

```
import statsmodels.api as sm

x_train = sm.add_constant(x_train)
model = sm.Logit(y_train, x_train).fit()
summary = model.summary()
print(summary)
```

```
x_test = sm.add_constant(x_test)
pred = model.predict(x_test)
```

```
# y_test
# pred

pred = np.where(pred > 0.5, 1,0)
```

```
# 1. 초기 500개 데이터로 분석시 sibsp 변수의 odds ratio는?

model.params
sibsp = model.params[2]
# print(sibsp)
odds_ratio = np.exp(sibsp)
print(round(odds_ratio,2))
```

```
# 2. 초기 500개 데이터로 분석시 residual deviance는?
ce = -238.54 * -1  # cross_entropy = (-)Log-Likelihood
```

```
rd = ce * 2   # residual_deviance = 2 * cross_entropy
print(rd)
```

```
# 3. 나머지 391개 데이터 적용시 오분류율은?
from sklearn.metrics import accuracy_score,confusion_matrix,c

acc = accuracy_score(y_test,pred)
matrix = confusion_matrix(y_test,pred)
report = classification_report(y_test,pred)


print(acc, matrix, report)

#  오분류율 =>  1 - 정확도
print(round(1-acc, 2))
```

```
0.782608695652174 [[210  32]
[ 53  96]]               precision    recall  f1-score   su
pport

       0       0.80      0.87      0.83       242
       1       0.75      0.64      0.69       149

accuracy                           0.78       391
macro avg       0.77      0.76      0.76       391
weighted avg       0.78      0.78      0.78       391

0.22
```

```
# 문제 3-2
# 다음은 당뇨병 환자의 질병 진행정도 데이터셋이다.
# target을 종속변수(Y), 나머지를 독립변수(X)로 분석했을 때 아래 질문에 답
# (단, 모든 답은 반올림하여 소수점 둘째자리까지 구하시오)
# 1. target 변수와 가장 큰 상관관계를 갖는 변수의 상관계수를 구하시오.
```

```
# 2. 다중선형회귀 모델링 후 결정계수(R2 score)를 구하시오.
# 3. 문제 2에서 구한 회귀모델에서 p-value가 가장 큰 변수의 p-value값을
```

```
###############  실기환경 복사 영역  ###############
# 데이터 불러오기
import pandas as pd
import numpy as np
# 실기 시험 데이터셋으로 셋팅하기 (수정금지)
from sklearn.datasets import load_diabetes
# diabetes 데이터셋 로드
diabetes = load_diabetes()
x = pd.DataFrame(diabetes.data, columns=diabetes.feature_name
y = pd.DataFrame(diabetes.target)
y.columns = ['target']
df = pd.concat([x,y], axis=1)
###############  실기환경 복사 영역  ###############
```

`df.head()`

```
# (단, 모든 답은 반올림하여 소수점 둘째자리까지 구하시오)
# 1. target 변수와 가장 큰 상관관계를 갖는 변수의 상관계수를 구하시오.
corr = df.corr()
corr
```

```
corr1 = abs(corr['target'].sort_values(ascending=False))[1]
print(round(corr1, 2))
```

0.59

```
# 2. 다중선형회귀 모델링 후 결정계수(R2 score)를 구하시오.
x = df.drop(columns='target')
y = df['target']

import statsmodels.api as sm

x= sm.add_constant(x)
model = sm.OLS(y,x).fit()
```

```
summary = model.summary()
print(summary)

r2_score = 0.518
print(round(r2_score, 2))
```

```
# 3. 문제 2에서 구한 회귀모델에서 p-value가 가장 큰 변수의 p-value값을

p_value = 0.867
print(round(p_value, 2))
```

0.87

#기출문제 8회- 3유형

```
# 제 3유형
# 예제문제 3유형-1번
# survived 를 종속변수(Y), 'pclass','age','parch','sibsp','fare
#  (단, 모든 답은 반올림하여 소수점 둘째자리까지 구하시오)
#  1-1. 전체 데이터를 모두 활용하여 로지스틱회귀분석 진행 후에 유의하지 않
#  1-2. 1번 문제에서 유의한 변수만 사용하여 로지스틱회귀분석을 진행했을 때
# 1-3. 만약 age변수가 5단위 증가하면 오즈비(Odds ratio)는 몇배로 변화하
```

```
######## 복사 영역 ########
# 데이터 생성(수정금지)
import pandas as pd
import numpy as np
 # Seaborn의 내장 타이타닉 데이터셋을 불러옵니다.
import seaborn as sns
df = sns.load_dataset('titanic')
df = df[ ['survived','pclass','age','parch','sibsp','fare'] ]
df = df.dropna()
######## 복사 영역 ########
print(df.head())
```

survived	pclass	age	parch	sibsp	fare
0	0	3	22.0	0	1 7.2500
1	1	1	38.0	0	1 71.2833
2	1	3	26.0	0	0 7.9250
3	1	1	35.0	0	1 53.1000
4	0	3	35.0	0	0 8.0500

```
x = df.drop(columns='survived')
y = df['survived']

x_train = df[ ['pclass','age','parch','sibsp','fare'] ]
y_train = df['survived']
```

```
import statsmodels.api as sm

x_train = sm.add_constant(x_train)
model_logit = sm.Logit(y_train, x_train).fit()
summary = model_logit.summary()
print(summary)
```

```
x_train = df.drop(columns=['survived', 'fare'])
y_train= df['survived']


print(x_train.shape)
print(y_train.shape)
```

(714, 4)
(714,)

```
import statsmodels.api as sm

x_train = sm.add_constant(x_train)
model_logit = sm.Logit(y_train, x_train).fit()
summary2 = model_logit.summary()
print(summary2)
```

```
#  1-2.
# 1번 문제에서 유의한 변수만 사용하여 로지스틱회귀분석을 진행했을 때 회귀계

print(round(model_logit.params.sum(),2))
# (정답) 1-2. 회귀계수의 합계 : 2.45
```

2.45

```
# 1-3. 만약 age변수가 5단위 증가하면 오즈비(Odds ratio)는 몇배로 변화하

result = np.exp(5*model_logit.params)
print(round(result,2))
print(round(result[2], 2))
```

const   1.404122e+08
pclass   0.000000e+00
age   8.000000e-01
parch   4.070000e+00
sibsp   2.600000e-01
dtype: float64
0.8

```
# 예제문제 3유형-2번
# 다음은 당뇨병 환자의 질병 진행정도 데이터셋이다.
#  target을 종속변수(Y), s1~s5를 독립변수(X)로 분석했을 때 아래 질문에
#  (단, 모든 답은 반올림하여 소수점 둘째자리까지 구하시오)
#  2-1. 다중선형회귀 모델링 후 p-value가 가장 작은 변수의 회귀계수 값을
#  2-2. 위에서 구한 모델의 결정계수(Rsq) 값을 구하시오.
#  2-3. 위에서 구한 모델에 s1=1, s2=2, s3=3, s4=4, s5=5 값을 대입하
```

```
 ###############  실기환경 복사 영역  ###############
 # 데이터 불러오기
import pandas as pd
 import numpy as np
 # 실기 시험 데이터셋으로 셋팅하기 (수정금지)
 from sklearn.datasets import load_diabetes
 # diabetes 데이터셋 로드
```

```
diabetes = load_diabetes()
 x = pd.DataFrame(diabetes.data, columns=diabetes.feature_nam
 y = pd.DataFrame(diabetes.target)
 y.columns = ['target']
df = pd.concat([x,y], axis=1)
 ###############  실기환경 복사 영역  ###############
 print(df.head())
```

```
    age       sex       bmi        bp        s1        s2
s3  \\
0  0.038076  0.050680  0.061696  0.021872 -0.044223 -0.0348
21 -0.043401

1 -0.001882 -0.044642 -0.051474 -0.026328 -0.008449 -0.0191
63  0.074412

2  0.085299  0.050680  0.044451 -0.005670 -0.045599 -0.0341
94 -0.032356

3 -0.089063 -0.044642 -0.011595 -0.036656  0.012191  0.0249
91 -0.036038

4  0.005383 -0.044642 -0.036385  0.021872  0.003935  0.0155
96  0.008142

     s4        s5        s6  target
0 -0.002592  0.019907 -0.017646   151.0

1 -0.039493 -0.068332 -0.092204    75.0

2 -0.002592  0.002861 -0.025930   141.0

3  0.034309  0.022688 -0.009362   206.0

4 -0.002592 -0.031988 -0.046641   135.0
```

```
# 독립변수와 종속변수 설정
x = df[ ['s1','s2','s3','s4','s5'] ]
y = df['target']
# print(x.head())
# print(y.head())
# 모델링
import statsmodels.api as sm
x = sm.add_constant(x)        # 주의 : 상수항 추가해줘야 함
model = sm.OLS(y, x).fit()    # 주의할 것 : y, x 순으로 입력해야 힘
# y_pred = model.predict(x)
summary = model.summary()
print(summary)
```

```
# p-value가 가장 작은 변수는 s5
print(model.params) # s5 의 회귀계수 확인
#(정답) 2-1 s5의 회귀계수 : 1163.74
```

```
const    152.133484
s1      -937.016237
s2       746.376765
s3        32.455999
s4        17.411725
s5      1163.736200
dtype: float64
```

```
# 모델의 rsq값 구하기
print(round(model.rsquared, 2))

#(정답) 2-2. Rsq : 0.36
```

```
# 새로운 데이터 적용
# 주의사항 : 모델에 상수항이 추가되었다면
# 새로운 데이터도 상수항을 추가해줘야 함(무조건 1 대입)
 # 리스트 형태
new_data = [1, 1, 2, 3, 4, 5] # 상수항, s1, s2, ... , s5
 # 데이터프레임 형태
```

```
new_data2 = pd.DataFrame({'const': [1],
                          's1': [1], 's2': [2], 's3': [3],
                          's4': [4], 's5': [5]})
print(new_data)
print(new_data2)
 # 예측값 계산
pred = model.predict(new_data)
pred2 = model.predict(new_data2)
 # 예측값 출력
print(pred)
print(pred2)
```

[1, 1, 2, 3, 4, 5]
const s1 s2 s3 s4 s5
0 1 1 2 3 4 5
[6693.56667609]
0 6693.566676
dtype: float64