회귀분석의 결과 보고

#01. 작업준비

패키지 참조

```
from pandas import read_excel, DataFrame
from statsmodels.formula.api import ols
from statsmodels.stats.outliers_influence import variance_inflation_fact
```

데이터 가져오기

```
df = read_excel("https://data.hossam.kr/E04/cars.xlsx")
df.head()
```

	speed	dist
0	4	2
1	4	10
2	7	4
3	7	22

	speed	dist
4	8	16

회귀분석 수행

```
model = ols("dist ~ speed", data=df)
fit = model.fit()
tbl = fit.summary()
tbl
```

OLS Regression Results

Dep. Variable:	dist	**R-squared:**	0.651
Model:	OLS	**Adj. R-squared:**	0.644
Method:	Least Squares	**F-statistic:**	89.57
Date:	Tue, 25 Jul 2023	**Prob (F-statistic):**	1.49e-12
Time:	09:51:41	**Log-Likelihood:**	-206.58
No. Observations:	50	**AIC:**	417.2
Df Residuals:	48	**BIC:**	421.0
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]

Intercept	-17.5791	6.758	-2.601	0.012	-31.168	-3.990
speed	3.9324	0.416	9.464	0.000	3.097	4.768

Omnibus:	8.975	**Durbin-Watson:**	1.676
Prob(Omnibus):	0.011	**Jarque-Bera (JB):**	8.189
Skew:	0.885	**Prob(JB):**	0.0167
Kurtosis:	3.893	**Cond. No.**	50.7

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

```
model.exog_names
```

```
['Intercept', 'speed']
```

회귀분석 결과 활용

모델을 활용한 결과값 얻기

```
speed = [10, 15, 20, 25, 30, 35, 40]
pred = fit.predict({"speed": speed})
pred
```

```
0     21.744993
1     41.407036
2     61.069080
3     80.731124
4    100.393168
5    120.055212
6    139.717255
dtype: float64
```

베타값 얻기

독립변수의 영향력을 나타내는 값. 계수를 표준화한 값으로 `0~1` 사이의 값을 갖는다. (1에 가까울수록 영향력이 큼)

#02. 회귀분석 결과 다루기

결과표의 크기

```
len(tbl.tables)
```

```
3
```

결과표 확인

```
tbl.tables[0]
```

회귀분석의 결과 보고

#01. 작업준비

패키지 참조

데이터 가져오기

회귀분석 수행

회귀분석 결과 활용

모델을 활용한 결과값 얻기

베타값 얻기

#02. 회귀분석 결과 다루기

결과표의 크기

결과표 확인

첫 번째 표의 내용

VIF값 생성

OLS Regression Results

Dep. Variable:	dist	**R-squared:**	0.651
Model:	OLS	**Adj. R-squared:**	0.644
Method:	Least Squares	**F-statistic:**	89.57
Date:	Tue, 25 Jul 2023	**Prob (F-statistic):**	1.49e-12
Time:	09:51:41	**Log-Likelihood:**	-206.58
No. Observations:	50	**AIC:**	417.2
Df Residuals:	48	**BIC:**	421.0
Df Model:	1		
Covariance Type:	nonrobust		

```
tbl.tables[1]
```

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	-17.5791	6.758	-2.601	0.012	-31.168	-3.990
speed	3.9324	0.416	9.464	0.000	3.097	4.768

```
tbl.tables[2]
```

Omnibus:	8.975	**Durbin-Watson:**	1.676
Prob(Omnibus):	0.011	**Jarque-Bera (JB):**	8.189

Skew:	0.885	**Prob(JB):**	0.0167
Kurtosis:	3.893	**Cond. No.**	50.7

첫 번째 표의 내용

```
import re

my = {}

for k in range(0, 3, 2):
    items = tbl.tables[k].data
    #print(items)

    for item in items:
        #print(item)
        n = len(item)

        for i in range(0, n, 2):
            key = item[i].strip()[:-1]
            value = item[i+1].strip()

            if key and value:
                my[key] = value

my
```

```
{'Dep. Variable': 'dist',
 'R-squared': '0.651',
 'Model': 'OLS',
 'Adj. R-squared': '0.644',
 'Method': 'Least Squares',
 'F-statistic': '89.57',
 'Date': 'Tue, 25 Jul 2023',
 'Prob (F-statistic)': '1.49e-12',
 'Time': '09:51:41',
 'Log-Likelihood': '-206.58',
 'No. Observations': '50',
 'AIC': '417.2',
 'Df Residuals': '48',
 'BIC': '421.0',
 'Df Model': '1',
 'Covariance Type': 'nonrobust',
 'Omnibus': '8.975',
 'Durbin-Watson': '1.676',
 'Prob(Omnibus)': '0.011',
 'Jarque-Bera (JB)': '8.189',
 'Skew': '0.885',
 'Prob(JB)': '0.0167',
 'Kurtosis': '3.893',
 'Cond. No': '50.7'}
```

VIF값 생성

```
for i in range(1, len(model.exog_names)):
    vif = variance_inflation_factor(model.exog, i)

    if vif < 10:
        print("%s의 VIF: %f (good)" % (model.exog_names[i], vif))
    else:
        print("%s의 VIF: %f (bad)" % (model.exog_names[i], vif))
```

```
speed의 VIF: 1.000000 (good)
```

```
my['variables'] = []

for i, v in enumerate(tbl.tables[1].data):
    if i == 0:
        continue

    # 변수의 이름
    name = v[0].strip()
    # 변수의 이름 목록
    name_list = list(model.exog_names)
    # 변수의 이름 목록에서 현재 변수가 몇 번째 항목인지 찾기
    j = name_list.index(name)

    vif = 0

    # 0번째인 Intercept는 제외
    if j > 0:
```

```
        vif = variance_inflation_factor(model.exog, j)

    my['variables'].append({
        "name": name,
        "coef": v[1].strip(),
        "std err": v[2].strip(),
        "t": v[3].strip(),
        "P-value": v[4].strip(),
        "Beta": 0,
        "VIF": vif,
    })

my
```

```
{'Dep. Variable': 'dist',
 'R-squared': '0.651',
 'Model': 'OLS',
 'Adj. R-squared': '0.644',
 'Method': 'Least Squares',
 'F-statistic': '89.57',
 'Date': 'Tue, 25 Jul 2023',
 'Prob (F-statistic)': '1.49e-12',
 'Time': '09:51:41',
 'Log-Likelihood': '-206.58',
 'No. Observations': '50',
 'AIC': '417.2',
 'Df Residuals': '48',
 'BIC': '421.0',
 'Df Model': '1',
 'Covariance Type': 'nonrobust',
```

```
 'Omnibus': '8.975',
 'Durbin-Watson': '1.676',
 'Prob(Omnibus)': '0.011',
 'Jarque-Bera (JB)': '8.189',
 'Skew': '0.885',
 'Prob(JB)': '0.0167',
 'Kurtosis': '3.893',
 'Cond. No': '50.7',
 'variables': [{'name': 'Intercept',
   'coef': '-17.5791',
   'std err': '6.758',
   't': '-2.601',
   'P-value': '0.012',
   'Beta': 0,
   'VIF': 0},
  {'name': 'speed',
   'coef': '3.9324',
   'std err': '0.416',
   't': '9.464',
   'P-value': '0.000',
   'Beta': 0,
   'VIF': 1.0}]}
```

```
mylist = []

for i in my['variables']:
    if i['name'] == 'Intercept':
        continue

    item = {
```

```
        "독립변수": i['name'],
        "B": i['coef'],
        "표준오차": i['std err'],
        "β": i['Beta'],
        "t": "%s*" % i['t'],
        "유의확률" : i['P-value'],
        "VIF": i["VIF"]
    }

    mylist.append(item)

df = DataFrame(mylist, index=['dist'])
df.index.name = '종속변수'
df
```

	독립변수	B	표준오차	β	t	유의확률	VIF
종속변수							
dist	speed	3.9324	0.416	0	9.464*	0.000	1.0

별표 개수	p값 범위
*	$p < 0.05$
**	$p < 0.01$
***	$p < 0.001$

```
"R(%s), R^2(%s), F(%s), 유의확률(%s), Durbin-Watson(%s)" % (my['R-squared
```

```
'R(0.651), R^2(0.644), F(89.57), 유의확률(1.49e-12), Durbin-Watson(1.676)
```