# 제3유형_다중회귀분석 및 상관분석

## ✅ 다중회귀분석

In [1]:
```
import pandas as pd
import numpy as np
```

## 당뇨병 환자의 질병 진행정도 데이터셋

In [2]:
```
###############  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_names)
y = pd.DataFrame(diabetes.target)
y.columns = ['target']
###############  실기환경 복사 영역  ###############
```

In [3]:
```
# 데이터 설명
print(diabetes.DESCR)
```

```
.. _diabetes_dataset:

Diabetes dataset
----------------

Ten baseline variables, age, sex, body mass index, average blood
pressure, and six blood serum measurements were obtained for each of n =
442 diabetes patients, as well as the response of interest, a
quantitative measure of disease progression one year after baseline.

**Data Set Characteristics:**

  :Number of Instances: 442

  :Number of Attributes: First 10 columns are numeric predictive values

  :Target: Column 11 is a quantitative measure of disease progression one year after baseline

  :Attribute Information:
      - age     age in years
      - sex
      - bmi     body mass index
      - bp      average blood pressure
      - s1      tc, total serum cholesterol
      - s2      ldl, low-density lipoproteins
      - s3      hdl, high-density lipoproteins
      - s4      tch, total cholesterol / HDL
      - s5      ltg, possibly log of serum triglycerides level
      - s6      glu, blood sugar level

Note: Each of these 10 feature variables have been mean centered and scaled by the standard deviation times the
square root of `n_samples` (i.e. the sum of squares of each column totals 1).

Source URL:
https://www4.stat.ncsu.edu/~boos/var.select/diabetes.html

For more information see:
Bradley Efron, Trevor Hastie, Iain Johnstone and Robert Tibshirani (2004) "Least Angle Regression," Annals of S
tatistics (with discussion), 407-499.
(https://web.stanford.edu/~hastie/Papers/LARS/LeastAngle_2002.pdf)
```

## 1. sklearn 라이브러리 활용

In [4]:
```
# sklearn 라이브러리 활용
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression
```

In [5]:
```
# 독립변수와 종속변수 설정
x = x[ ['age','sex','bmi'] ]
print(x.head())
print(y.head())
```

```
        age       sex       bmi
0  0.038076  0.050680  0.061696
1 -0.001882 -0.044642 -0.051474
2  0.085299  0.050680  0.044451
3 -0.089063 -0.044642 -0.011595
4  0.005383 -0.044642 -0.036385
   target
0   151.0
1    75.0
2   141.0
3   206.0
4   135.0
```

- 회귀식 : y = b0 + b1x1 + b2x2 + b3x3
  (x1=age, x2=sex, x3=bmi)

In [6]:

```
# 모델링
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(x, y)
```

Out[6]:

```
▾ LinearRegression
LinearRegression()
```

In [7]:

```
# 회귀분석 관련 지표 출력

# 1. Rsq(결정계수) : model.score(x, y)
model.score(x, y)
print(round(model.score(x, y), 2) )
```

```
0.35
```

In [8]:

```
# 2. 회귀계수 출력 : model.coef_
print(np.round(model.coef_, 2) )        # 전체 회귀계수
print(np.round(model.coef_[0,0], 2) )  # x1 의 회귀계수
print(np.round(model.coef_[0,1], 2) )  # x2 의 회귀계수
print(np.round(model.coef_[0,2], 2) )  # x3 의 회귀계수
```

```
[[138.9   -36.14 926.91]]
138.9
-36.14
926.91
```

In [9]:

```
# 3. 회귀계수(절편) : model.intercept_
print(np.round(model.intercept_, 2) )
```

```
[152.13]
```

- 회귀식 : y = b0 + b1x1 + b2x2 + b3x3
  (x1=age, x2=sex, x3=bmi) ### 결과 : y = 152.13 + 138.9age - 36.14sex + 926.91bmi

## 2. statsmodels 라이브러리 사용

In [10]:

```
###############  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_names)
y = pd.DataFrame(diabetes.target)
y.columns = ['target']
###############  실기환경 복사 영역  ###############
```

In [11]:

```
# statsmodel.formula 활용
import statsmodels.api as sm
# 독립변수와 종속변수 설정
x = x[['age','sex','bmi']]
y = y['target']
print(x.head())
print(y.head())
```

```
        age       sex       bmi
0  0.038076  0.050680  0.061696
1 -0.001882 -0.044642 -0.051474
2  0.085299  0.050680  0.044451
3 -0.089063 -0.044642 -0.011595
4  0.005383 -0.044642 -0.036385
0    151.0
1     75.0
2    141.0
3    206.0
4    135.0
Name: target, dtype: float64
```

In [12]:

```
# 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 함
# y_pred = model.predict(x)
summary = model.summary()
print(summary)
```

```
                            OLS Regression Results                            
==============================================================================
Dep. Variable:                 target   R-squared:                       0.351
Model:                            OLS   Adj. R-squared:                  0.346
Method:                 Least Squares   F-statistic:                     78.94
Date:                Fri, 10 Nov 2023   Prob (F-statistic):           7.77e-41
Time:                        23:16:26   Log-Likelihood:                -2451.6
No. Observations:                 442   AIC:                             4911.
Df Residuals:                     438   BIC:                             4928.
Df Model:                           3                                         
Covariance Type:            nonrobust                                         
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        152.1335      2.964     51.321      0.000     146.307     157.960
age          138.9039     64.254      2.162      0.031      12.618     265.189
sex          -36.1353     63.391     -0.570      0.569    -160.724      88.453
bmi          926.9120     63.525     14.591      0.000     802.061    1051.763
==============================================================================
Omnibus:                       14.687   Durbin-Watson:                   1.851
Prob(Omnibus):                  0.001   Jarque-Bera (JB):                8.290
Skew:                           0.150   Prob(JB):                       0.0158
Kurtosis:                       2.400   Cond. No.                         23.7
==============================================================================

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

In [13]:

```
# 1. Rsq(결정계수)
# r2 = 0.351

# 2. 회귀계수
# age = 138.9039
# sex = -36.1353
# bmi = 926.9120

# 3. 회귀계수(절편)
# const = 152.1335

# 4. 회귀식 p-value
# pvalue = 7.77e-41
```

### (결과 비교해보기) 두 라이브러리 모두 같은 결과값을 출력

- 회귀식 : y = b0 + b1x1 + b2x2 + b3x3 (x1=age, x2=sex, x3=bmi) #### 1. sklearn : y = 152.13 + 138.9age - 36.14sex + 926.91bmi #### 2. statsmodel : y = 152.13 + 138.9age - 36.14sex + 926.91bmi

## ✅ 상관분석

In [14]:

```
###############  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_names)
y = pd.DataFrame(diabetes.target)
y.columns = ['target']
###############  실기환경 복사 영역  ###############
```

```
In [15]: # 상관분석을 할 2가지 변수 설정
         x = x['bmi']
         y = y['target']
         print(x.head())
         print(y.head())
```

```
0    0.061696
1   -0.051474
2    0.044451
3   -0.011595
4   -0.036385
Name: bmi, dtype: float64
0    151.0
1     75.0
2    141.0
3    206.0
4    135.0
Name: target, dtype: float64
```

```
In [16]: # 라이브러리 불러오기
         from scipy.stats import pearsonr

         # 상관계수에 대한 검정실시
         r, pvalue = pearsonr(x, y)

         # 가설설정
         # H0 : 두 변수간 선형관계가 존재하지 않는다 (ρ = 0)
         # H1 : 두 변수간 선형관계가 존재한다 (ρ ≠ 0)


         # 1. 상관계수
         print(round(r, 2) )

         # 2. p-value
         print(round(pvalue, 2))

         # 3. 검정통계량
         # 통계량은 별도로 구해야 함 (T = r * root(n-2) / root(1-r2) )
         # r = 상관계수
         # n = 데이터의 개수

         n = len(x)   # 데이터 수
         r2 = r**2    # 상관계수의 제곱
         statistic = r * ((n-2)**0.5) / ((1-r2)**0.5)

         print(round(statistic, 2))

         # 4. 귀무가설 기각여부 결정(채택/기각)
         # p-value 값이 0.05보다 작기 때문에 귀무가설을 기각한다.(대립가설채택)
         # 즉, 두 변수간 선형관계가 존재한다고 할 수 있다.(상관계수가 0이 아니다)

         # 답 : 기각
```

```
0.59
0.0
15.19
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js