중선형회귀분석(Multiple Linear Regression Analysis)

체중(x1), 나이(x2)와 혈압(y)과의 관계조사

공정온도(x1), 공정압력(x2)과 제품의 강도(y)와의 관계조사

x : 독립변수(independent variable) or 설명변수(explanatory variable)가 2개 이상

y : 종속변수(dependent variable) or 반응변수(response variable)

중선형회귀모형(Multiple Linear Regression Model)

x1, x2 와 y의 관찰값이 관계가 선형 such that

$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + e_i \quad i = 1,2,,\cdots,n \quad e_i \sim iid\ N(0, \sigma^2)$

단순선형회귀분석(Simple Linear Regression Analysis)

eg) 약의 복용량(x)과 약효지속시간(y)의 관계조사

공정온도(x)와 제품의 강도(y)관계조사

TOEFL점수(x)와 학과 영어과목성적(y)과의 관계

x : 독립변수(independent variable), 설명변수(explanatory variable)

입력변수(input variable)

y : 종속변수(dependent variable), 반응변수(response variable)

출력변수(output variable), target 변수

1.단순선형회귀모형(Simple Linear Regression Model)

2변수의 x와 y의 관찰값이 관계가 다음과 같은 경우:

$y_i = \beta_0 + \beta_1 x_i + e_i \quad i = 1,2,,\cdots,n \quad e_i \sim iid\ N(0, \sigma^2)$

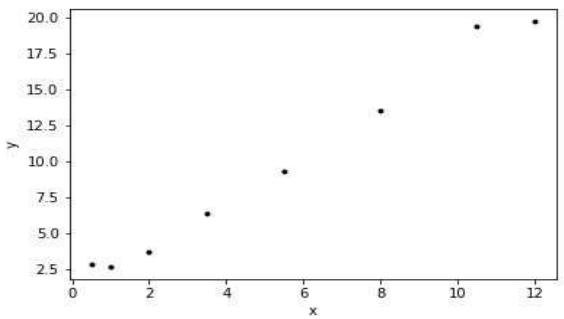
β_0 : y와 x의 직선의 절편(intercept)을 나타내는 모수(parameter)

β_1 : y와 x의 직선의 기울기(slope)를 나타내는 모수(parameter)

~ x가 1증가하면 y는 β_1만큼 증가

y_i : i번째 실험의 y의 값 , x_i : i번째 실험의 x의 값(입력변수)

eg1) 약의 복용량(x)과 약효지속시간(y)의 관계조사(x_y.csv)



i	x_i (mg)	y_i (hr)
1	0.5	2.86
2	1.0	2.66
3	2.0	3.69
4	3.5	6.40
5	5.5	9.27
6	8.0	13.53
7	10.5	19.38
8	12.0	19.71

2. 최소제곱법(Least Square Method)

주어진 데이터 $(x_1,y_1),(x_2,y_2),\cdots,(x_n,y_n)$를 이용하여

β_0, β_1 의 추정값인 $\hat{\beta}_0, \hat{\beta}_1$ 를 구하기

$\hat{\beta}_0, \hat{\beta}_1$은 $Q = \sum_{i=1}^{n} (y_i - b_0 - b_1 x_i)^2$ 를 최소화시키는 b_0, b_1의 값

→ $\hat{\beta}_1$: $\frac{\partial Q}{\partial b_1} = 0$ 를 만족시키는 b_1의 값 , $\hat{\beta}_0$: $\frac{\partial Q}{\partial b_0} = 0$ 를 만족시키는 b_0의 값

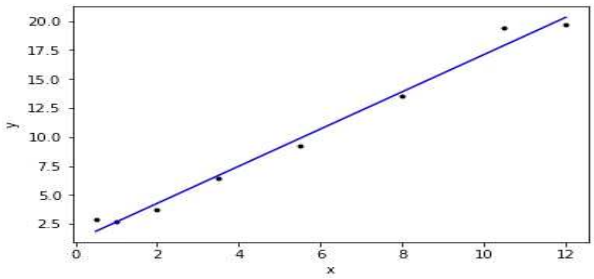
≡ $X = (\mathbf{1}, \boldsymbol{x})$, $\mathbf{1} \in R^{n\times 1}$, $\boldsymbol{x} \in R^{n\times d}, d=1$, $\boldsymbol{y} \in R^{n\times 1}$

$\hat{\boldsymbol{\beta}} = \begin{pmatrix}\hat{\beta}_0 \\ \hat{\beta}_1\end{pmatrix} = (X'X)^{-1}X'\boldsymbol{y}$

* 적합된 회귀직선(Fitted regression line) : $\hat{y} = X\hat{\boldsymbol{\beta}} = \hat{\beta}_0 + \hat{\beta}_1 x$

eg) $\hat{\boldsymbol{\beta}} = \begin{pmatrix}\hat{\beta}_0 \\ \hat{\beta}_1\end{pmatrix} = \begin{pmatrix}1.0605 \\ 1.6046\end{pmatrix}$

적합된 회귀직선(Fitted regression line) : $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x = -1.0605 + 1.6046x$ =
~약복용량이 1mg 증가하면 약효지속시간은 1.6046시간 증가.



eg) 시험자료가 주어지면, $\boldsymbol{x}_t = (1.5, 3, 6, 9, 13)' \sim \boldsymbol{X}_t = (\mathbf{1}, \boldsymbol{x}_t), \mathbf{1} \in R^{5\times 1}$
약효지속시간의 예측값은 $\hat{y}(\boldsymbol{x}_t) = X_t\hat{\boldsymbol{\beta}} = \hat{\beta}_0 + \hat{\beta}_1 x_t = -1.0605 + 1.6046x_t$

3. 결정계수

SSTO(Total Sum of Squares, 전체제곱합) = $\sum_{i=1}^{n}(y_i - \bar{y})^2$

SSE(Error Sum of Squares, 오차제곱합) = $\sum_{i=1}^{n}\hat{e}_i = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$

MSE(Mean Squared errors) = $\frac{SSE}{n-2}$: σ^2 의 추정값 = $\hat{\sigma}^2$

SSR(Refression Sum of Squares, 회귀제곱합) = $\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2$ = SSTO-SSE

결정계수(R^2 coefficient of determination): x변수를 사용함으로써 줄어드는 전체의 변동의 비율 - x변수가 설명할수 있는 y변수의 변동의 비율 - 주어진 모델이 전체변동(y)를 얼마나 잘 설명하는지 나타내는 값 [0,1]

$R^2 = \frac{SSR}{SSTO}$

상관계수(r, Correlation coefficient) = $\pm\sqrt{R^2}$ (부호= $\hat{\beta}_1$의 부호) [- or +]

<Python>
```
>import numpy as np
>from numpy import linalg as LA
>import matplotlib.pyplot as plt
>n=8; nt=5
>x=np.array([ 0.5,  1. ,  2. ,  3.5,  5.5,  8. , 10.5, 12. ]).reshape(n,1)
>y=np.array([ 2.8654883 ,  2.66422646,  3.68812138,  6.39507074,  9.26655548,
13.52528373, 19.38166085, 19.6975463 ]).reshape(n,1)
>xt=np.array([ 1.5 ,  3,  6,  9, 13]).reshape(nt,1)

>X=np.concatenate((np.ones((n,1)),x),axis=1)
>beta=LA.inv(X.T.dot(X)).dot(X.T).dot(y)
>print('beta=',beta)

>yh=X.dot(beta)
>e=y-yh # 잔차(residual)
>SSTO=np.sum((y-np.mean(y))**2)
>SSE=np.sum(e**2); SSR=SSTO-SSE
>R2=SSR/SSTO
>corr=np.sign(beta[1])*np.sqrt(R2)
>corr1=np.corrcoef(x.T,y.T) # corr=corr1[0,1]

>print('R square=',R2,'cor(x,y)=', corr)
>print('cor(x,y) matrix=',corr1)

>plt.figure(1)
>plt.plot(x,y,'k.');plt.plot(x,yh,'b-');plt.xlabel('x'); plt.ylabel('y')

>xt=np.array([ 1.5,  2.5,  6. ,  9.5, 11. ]).reshape(nt,1)
>Xt=np.concatenate((np.ones((nt,1)),xt),axis=1)
>yth=Xt.dot(beta)
>print('yth=',yth)

##########################################
>from sklearn.linear_model import LinearRegression
>model = LinearRegression().fit(x, y)
>beta0=model.intercept_
>beta1=model.coef_
>R2= model.score(x,y)
>yh=model.predict(x)
>yth=model.predict(xt)
```