개요

두 특정형변수, (X, Y)의 함수관계, 그 중 해석이 가장 용이한 직선 함수관계에 관심을 갖는다.

방향성이 없는 직선관계에 대한 상관분석과 달리 회귀분석은 입력 input에(영향을 주는) 해당되는 변수(독립변수)와 output(결과로 나타나는) 변수(목표변수)로 구성된다.

목표변수(Y)는 하나이고 정규분포를 따르는 측정형변수를 원칙으로 한다.

- 목표변수가 정규분포를 따르지 않는 경우 연결함수를 사용하여 목표변수를 정규분포화 한 후 회귀모형을 추정한다.

Distribution	Support of distribution	Typical uses	Link name	Link function, $\mathbf{X}\boldsymbol{\beta} = g(\mu)$
Normal	real: $(-\infty, +\infty)$	Linear-response data	Identity	$\mathbf{X}\boldsymbol{\beta} = \mu$
Exponential Gamma	real: $(0, +\infty)$	Exponential-response data, scale parameters	Negative inverse	$\mathbf{X}\boldsymbol{\beta} = -\mu^{-1}$
Inverse Gaussian	real: $(0, +\infty)$		Inverse squared	$\mathbf{X}\boldsymbol{\beta} = \mu^{-2}$
Poisson	integer: $0, 1, 2, \dots$	count of occurrences in fixed amount of time/space	Log	$\mathbf{X}\boldsymbol{\beta} = \ln(\mu)$
Bernoulli	integer: $\{0, 1\}$	outcome of single yes/no occurrence	Logit	$\mathbf{X}\boldsymbol{\beta} = \ln\left(\frac{\mu}{1-\mu}\right)$
Binomial	integer: $0, 1, \dots, N$	count of # of "yes" occurrences out of N yes/no occurrences		$\mathbf{X}\boldsymbol{\beta} = \ln\left(\frac{\mu}{n-\mu}\right)$
Categorical	integer: $[0, K)$ K-vector of integer: $[0, 1]$, where exactly one element in the vector has the value 1	outcome of single K-way occurrence		$\mathbf{X}\boldsymbol{\beta} = \ln\left(\frac{\mu}{1-\mu}\right)$
Multinomial	*K*-vector of integer: $[0, N]$	count of occurrences of different types (1 .. *K*) out of *N* total *K*-way occurrences		

예측변수($X's$)는 다수이고 역시 측정형변수를 원칙으로 한다.

- 변수가 하나인 경우는 단순회귀분석이라 하고 2개 이상인 경우에는 다중 회귀라 한다.
- 함수관계이므로 측정형이 아닌 범주형 변수는 직접 사용할 수 없다. 회귀분석에서 범주형 변수는 지시변수라 하며 범주형 변수의 수준(범주) 개수보다 1개 작은 이진형 binary dummy 더미 변수를 만들어 사용해야 한다. 이진형 더미변수는 $(0,1)$ 값만 갖는다.
- 이진형 더미변수는 절편에만 영향을 미친다. 만약 측정형 변수*더미 변수 항이 들어가면 해당 예측변수의 더미 변수 1인 경우 기울기가 달라지게 된다.

더미변수 모형 : 목표변수 Y : 수능성적, 측정형 예측변수 X : 일주일 공부시간, 더미변수 성별

$D = 0\ male,\ 1\ female$이라 하자.

$Y = a + b_1 D + b_2 X + b_3 (X * D) + e$

남학생 추정모형 : $Y = a + b_2 X$ - 절편 a, 기울기 b_2

여학생 추정 모형 : $Y = (a + b_1) + (b_2 + b_3) X$ - 절편 $a + b_1$, 기울기 $b_2 + b_3$

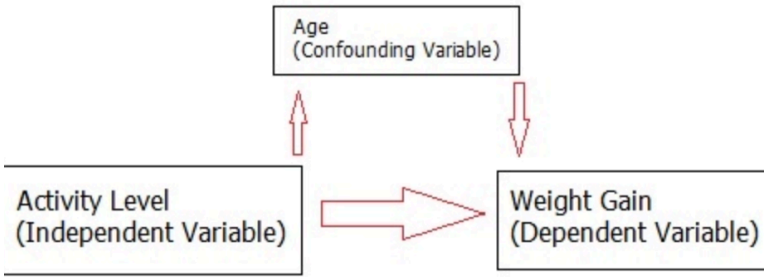
함수 형태 중 선형을 가장 선호 $y = a + b_1X_1 + b_2X_2 + \ldots + b_pX_p + e$

해석이 용이: 회귀계수는 기울기이므로 해당 예측변수가 한단위 증가(감소)할 때 목표변수의 단위 변화량

비선형모형도 로그변환을 통하여 선형으로 변환 가능

- Cobb-Douglas 생산함수 $Q = \alpha L^{\beta} L^{\lambda} u$ (Q=생산량, k=자본, L=투입노동, α, β, λ 모수, u오차항) - 양변을 로그 변환하면 $ln(Q) = ln(\alpha) + \beta ln(L) + \lambda ln(K) + ln(u)$
- 인구성장모형 $P = \alpha e^{\beta T} u$ (P= 총인구수, T=시간, α, β 모수) 모두 양변에 로그를 취하여 선형함수로 만들 수 있음 $ln(P) = ln(\alpha) + \beta T + ln(u)$

회귀분석은 함수관계에 대한 것이지 인과관계는 아니다. 단지 관심의 대상이 되는 목표변수에 직선적 함수를 갖는 예측변수를 찾는 것일 뿐이다. 인과관계는 실험설계를 통해서만 가능하다.
공부시간에 수능성적에 인과관계가 있는지 알아보기 위해서는 사회경제환경이 동일하고 공부능력이 동일한 고3학생을 산택하여 1년 동안 일주일 공부시간 평균과 수능성적 데이터를 수집하여 분석할 때만 인과관계 분석이 가능하다.
실험 데이터가 아닌 횡단 시점 데이터에서는 영향을 주는 요인을 통제한 후 예측변수와 목표변수의 관계를 정확하게 판단할 수 있다. 이런 변수를 통제 control 변수, 매개 mediator 변수, 교락 confounding 변수라 한다.



[운동능력과 체중 함수관계의 유의성을 정확하게 판단하기 위하여 나이를 통제변수로 활용하였다.]

회귀모형 순서

순서	내용
모형설정	관심 대상인 목표변수에 영향을 미치는 예측변수를 설정한다.
데이터 전처리	산점도를 그리고 필요 시 정규변환, 선형 변환 실시
모형추정	OLS 방법에 의해 모형 추정 $min(y - \hat{y})^2$
유의변수선택	목표변수에 유의한 영향을 미치는 변수 선택
다중공선성 진단	예측변수 매우 높은 상관관계로 인한 문제 진단 및 해결
모형진단 및 활용	잔차 활용 오차 가정 진단 이상치, 영향치 진단 및 해결 변수 간 영향력(중요도) 비교, 적합치, 예측구간, 신뢰구간 도출

회귀모형

모형

$$Y_i = a + b_1 X_{i1} + b_2 X_{i2} + \ldots + + b_p X_{ip} + e_i, \text{(가정)} e_i \sim iid\ N(0,\sigma^2)$$

$$\begin{bmatrix} y_1 \\ y_2 \\ \cdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & \cdots & x_{2p} \\ \cdots & & & \\ 1 & x_{n1} & \cdots & x_{np} \end{bmatrix} \begin{bmatrix} a \\ b_1 \\ \cdots \\ b_p \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ \cdots \\ e_n \end{bmatrix}, \underline{y} = X\underline{b} + \underline{e}, \text{(가정)} \underline{e} \sim MNV(\underline{0}, \sigma^2 I)$$

- 첨자 i는 표본 데이터를 구별을 위한 것임. $i = 1,2,...,n$ 총 표본의 크기
- Y : 종속 dependent 변수, 반응 response 변수, 내생 endogeous 변수, 목표 target 변수 - 목표변수는 확률변수이며 정규분포를 따른다고 가정한다.
- $X's$: 독립 independent 변수, 설명 exploratory 변수, 외생 exogenous 변수, 예측 predictor, feature 변수 - 결정변수 deterministic로 확률변수가 아니므로 분포함수를 갖지 않으며 모형 내에서 결정된다.
- p : 모형 내 고려되는 예측변수 개수
- α : 절편(intercept), 모든 예측변수 X 값들이 0 일때 목표변수 Y의 값
- β_k : 예측변수 X_k의 회귀계수 - 예측변수 X_k 가 한 단위 증가할 때마다 목표변수 Y의 증가량(편미분 계수)
- e_i : 오차항, noise

가정 assumption

선형성 linearity

목표변수와 예측변수들은 직선의 함수관계를 갖는다.

오차 가정 $\underline{e} \sim MNV(\underline{0}, \sigma^2 I)$

회귀모형에 설정된 예측변수($X's$)들에 의해 설명되지 못하는 부분, 오차항이 없으면 통계모형이 아니라 수학모형 (모형 내의 예측변수에 의해 종변속수 값을 정확하게 예측할 수 있음)

즉 목표변수 값의 패턴(변동)은 회귀모형이 설명하는 변동과 모형이 설명하지 못하는 변동(수학함수이면 없음) 나누어 변동의 비를 이용하여 모형의 유의성을 검정한다.

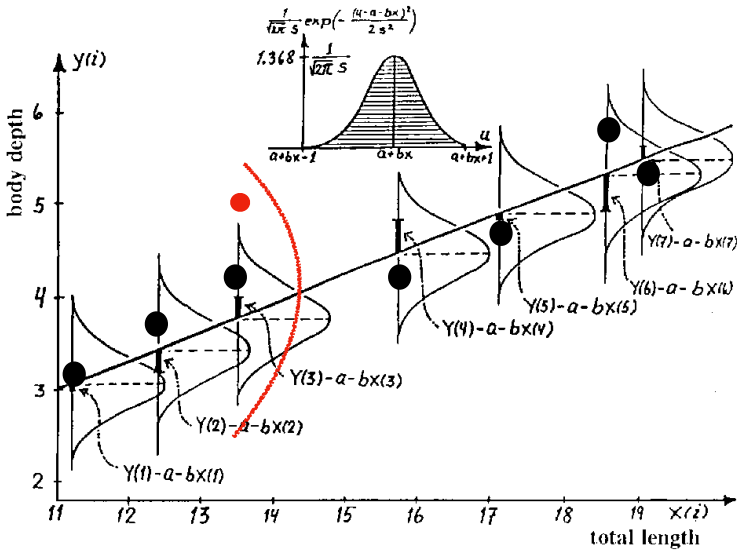
독립성(independent)

오차항은 서로 독립이다. 즉 각 오차는 서로 영향을 주지 않는다. 독립성 가정은 시계열 데이터(시간적 순서를 갖는 데이터) 경우에만 체크한다.

정규성(normality)

오차항은 정규 분포를 따른다. 이 가정은 분산분석 F-검정(회귀모형의 적합성), 회귀계수의 유의성 검정을 위한 $t-$검정을 사용하기 위하여 반드시 필요하다.

오차항이 정규분포를 따르므로 목표변수 $\underline{y} \sim MVN(X\underline{b}, \sigma^2 I)$도 정규분포를 따른다. 그림에서 볼 수 있듯이 데이터 관측값은 정규분포를 따르는 오차항으로 인하여 적합 회귀모형(회귀식 $X\underline{b}$) 선상에서 벗어나게 된다. 벗어나는 정도는 모든 산상에서 동일하며 정규분포 규칙을 따른다.



등분산성(Homoscedasticity)

오차항의 분산은 동일. 분산이 일정하다는 가정의 주어진 예측변수 값에서 관측되는 Y의 값의 분산이 일정하다는 의미와 같다. 분산이 다르면 설정된 회귀 모형이 적절함에도 불구하고 관측치가 직선에 모여 있지 않게 된다. 분산이 크므로 벗어나는 경향이 있다. 등분산 가정이 성립하지 않으면 많이 벗어난 관측치가 이상치인지 실제 발생 가능한 값인지 판단할 수 없다. 위의 그림에서 빨간 원 관측값은 등분산 가정이 만족하면 이상치에 해당되나 X(total length)=13.5에서 분산이 다른 예측변수 값에 비해 크 다면 빨간 원은 발생 가능(정상적인)한 관측값이다.

수리적 접근(상관계수 접근)

두 측정형 변수 (X, Y)는 평균으로 μ_x, μ_y 분산은 σ_x^2, σ_y^2를 따르고 상관계수를 ρ_{xy}라 하자.

$X = x$가 주어진 경우 $E(Y \mid x) = \mu_y + \rho \dfrac{\sigma_y}{\sigma_x}(x - \mu_x) = a + bx$

$$E(var(Y \mid x)) = \sigma_y^2(1 - \rho).$$

산점도 행렬 Scatter plot

예제 데이터

```
import pandas as pd
import numpy as np
#smsa=pd.read_csv('http://wolfpack.hnu.ac.kr/Stat_Notes/adv_stat/LinearModel/data/SMSA.csv')
smsa=pd.read_csv('http://www.users.miamioh.edu/baileraj/classes/sta363/DASL-SMSA-commasep.csv')
smsa.info()
```

```
1.city: City name : 60 U.S. Standard Metropolitan Statistical Areas
[기후요인] 4개
2.JanTemp: Mean January temperature (degrees Farenheit)
3.JulyTemp: Mean July temperature (degrees Farenheit)
4.RelHum: Relative Humidity
5.Rain: Annual rainfall (inches)
6.Mortality: Age adjusted mortality : 목표변수 Y
[사회경제요인] 6개
7.Education: Median education
8.PopDensity: Population density
9.%NonWhite: Percentage of non whites
10.%WC: Percentage of white collar workers
11.pop: Population *)인구밀도와 중복되어 예측변수에서 제외함
12.pop/house: Population per household
13.income: Median income
[환경요인] 4개
14.HCPot: HC pollution potential
15.NOxPot: Nitrous Oxide pollution potential
16.SO2Pot: Sulfur Dioxide pollution potential
17.NOx: Nitrous Oxide
```

데이터 전처리(1) preprocess : 이상치 진단

```
smsa.isnull().sum()  #check 결측값 : income,pop 1개 결측값 존재 확인
smsa.dropna(inplace=True) # 결측값 도시 1개 제외
smsa.shape  (59, 17) 59개 행, 17개 열 변수
```

City 변수를 ‘,’로 나눈 후 처음 행을 도시이름으로 저장하고 행 인덱스로 변환하였음

```
smsa['city_name']=smsa['city'].str.split(',',n=2,expand=True)[0]
smsa.set_index('city_name',inplace=True)
smsa.head(3)
```

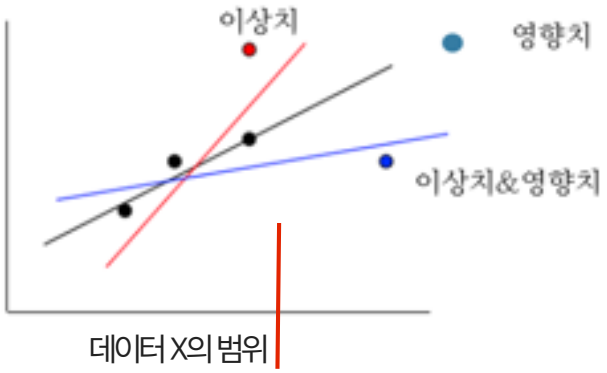
city_name	city	JanTemp	JulyTemp
Akron	Akron, OH	27	71

산점도 개념

2개의 측정형 변수 데이터를 2차원 공간에 표현하여 두 변수의 함수 관계를 예상한다.

- X- 축 : 결정의 요인, 예측변수
- Y-축 : Output, 목표변수

시각화 통계소프트웨어 발달로 각 변수의 히스토그램을 대각 셀(원소)에 출력할 수 있다.



진단내용

- 두 변수 간의 함수 관계를 본다. - 목표변수와 예측변수 선형성에 대한 시각적 진단 : 최종 진단은 해당 예측변수 회귀모형 회귀계수 유의성 검정으로 확인한다.
- 대각 셀에 나타난 히스트그램으로 치우침을 확인하고 필요 시 정규진단과 정규변환을 실행한다.
- 이상치와 영향치를 시각적으로 진단 : 통계량을 이용한 진단은 최종모형 도출 후 회귀진단에서 실시한다.

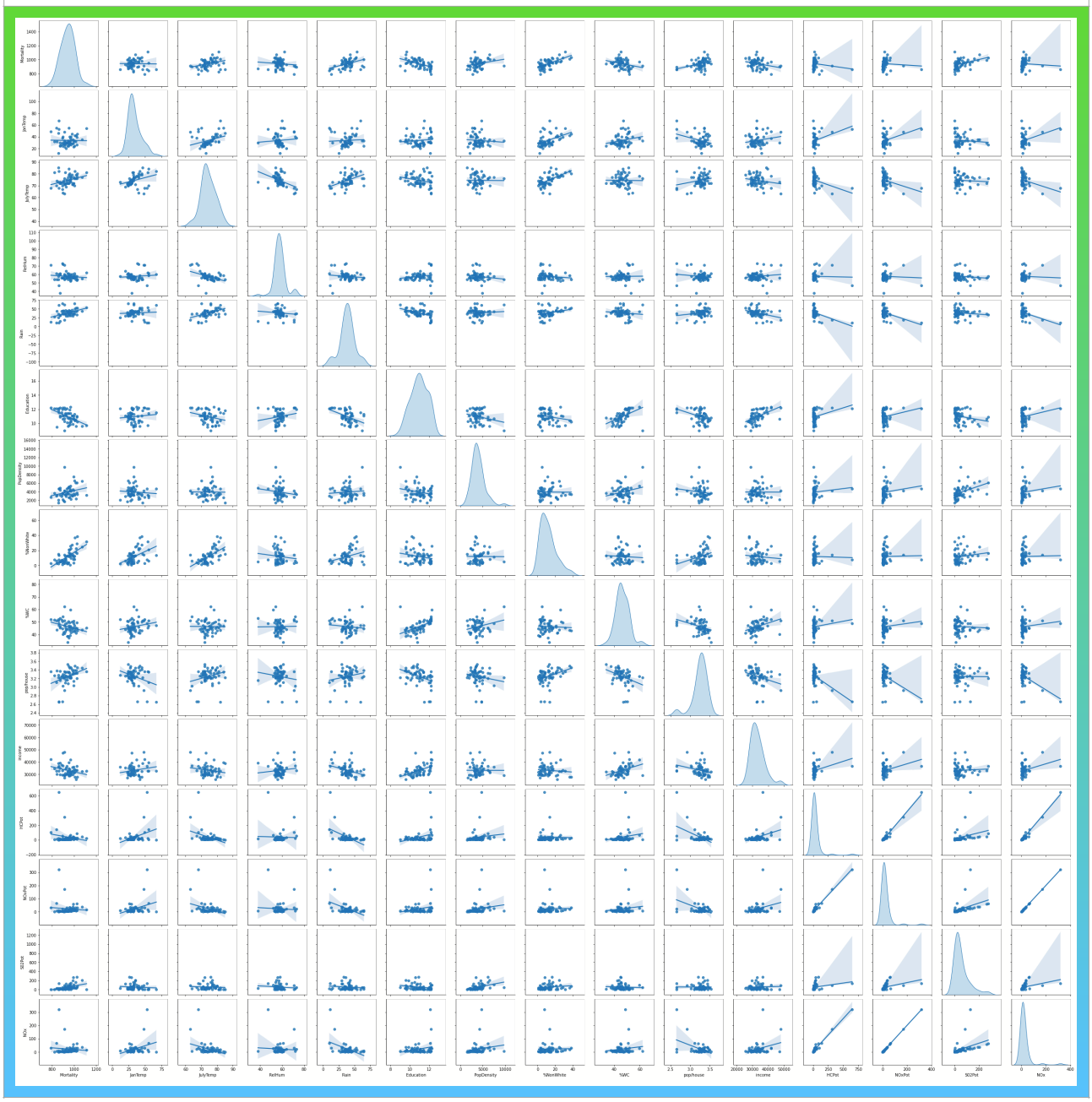
```
tmp=smsa.iloc[:,1:17].drop(columns=['pop','Mortality'])
df=pd.concat([smsa['Mortality'],tmp],axis=1)
df.shape,df.columns
```

불필요한 변수 제거하고 목표변수로 사용할 'Mortality' 변수를 가장 앞에 놓았음 – 산점도 행렬이나 상관계수 행렬 해석 시 편리함

```
((59, 15),
 Index(['Mortality', 'JanTemp', 'JulyTemp', 'RelHum', 'Rain', 'Education',
        'PopDensity', '%NonWhite', '%WC', 'pop/house', 'income', 'HCPot',
        'NOxPot', 'SO2Pot', 'NOx'],
       dtype='object'))
```

```
import seaborn as sns
import matplotlib.pyplot as plt
sns.pairplot(df,kind='reg',diag_kind='kde')
plt.show()
```

대각원소 : 데이터 히스트그램을 볼 때 환경 요인은 우로 치우친 형태를 갖는다. 제곱근 혹은 로그변환 필요함



전처리 (2) : 정규변환

```
from scipy.stats import shapiro
for k in range(0,df.shape[1]):
  if shapiro(df.iloc[:,k])[1]<0.05:
    print(smsa.columns[k],': Not Normal')
```

환경요인 외 다른 변수들이 정규분포를 따르지 않으나 좌로 치우친 경우나 낮은 봉우리로 인한 것이므로 환경요인 2개 HCPot, NOxPot 우로 치우침 문제 해결을 위한 정규변환, 로그 변환 혹은 제곱근 변환

```
JanTemp Not Normal
RelHum Not Normal
Rain Not Normal
Education Not Normal
PopDensity Not Normal
%WC Not Normal
pop Not Normal
pop/house Not Normal
income Not Normal
HCPot Not Normal
NOxPot Not Normal
```

```
import numpy as np
shapiro(np.log(df['HCPot']))[1], shapiro(np.log(df['NOxPot']))[1]
```

환경요인 2개 HCPot(유의확률이 4.6%로 완벽하지 않지만), NOxPot 로그정규변환으로 정규분포화 되었음

```
(0.046230245381593704, 0.45190250873565674)
```

원변수 로그변환하고 원변수는 제외하였음

```
df['log_HCPot']=np.log(df['HCPot'])
df['log_NOxPot']=np.log(df['NOxPot'])
df_clean=df.drop(columns=['HCPot','NOxPot'])
df_clean.columns
```

상관계수 행렬

```
#Using Pearson Correlation
pd.options.display.float_format = '{:.3f}'.format
df_cor=df_clean.corr(method ='pearson')
```

목표변수 Mortality와 예측변수들의 상관계수 첫 행(열)은 목표변수에 유의한 잠정 예측변수 사전 진단 가능함
나머지 예측변수 간 상관계수는 높은 상관계수가 존재하는 예측변수 군에 대한 사전 정보 얻을 수 있음 – 다중 공선성 문제 : 환경 요인들 간 상관관계 매우 높음

```
df_cor.style.background_gradient(cmap='coolwarm').set_precision(3)
```

	Mortality	JanTemp	JulyTemp	RelHum	Rain	Education	PopDensity	%NonWhite	%WC	pop/house	income	S02Pot	NOx	log_HCPot	log_NOxPot
Mortality	1.000	-0.016	0.322	-0.101	0.433	-0.508	0.252	0.647	-0.289	0.368	-0.283	0.419	-0.085	0.125	0.280
JanTemp	-0.016	1.000	0.322	0.086	0.059	0.108	-0.076	0.459	0.208	-0.325	0.198	-0.094	0.334	0.227	0.181
JulyTemp	0.322	0.322	1.000	-0.441	0.472	-0.269	-0.009	0.602	-0.013	0.257	-0.191	-0.071	-0.334	-0.413	-0.303
RelHum	-0.101	0.086	-0.441	1.000	-0.118	0.186	-0.149	-0.119	0.015	-0.144	0.128	-0.116	-0.053	0.185	0.097
Rain	0.433	0.059	0.472	-0.118	1.000	-0.473	0.084	0.303	-0.114	0.199	-0.362	-0.131	-0.460	-0.485	-0.389
Education	-0.508	0.108	-0.269	0.186	-0.473	1.000	-0.236	-0.209	0.486	-0.389	0.507	-0.229	0.229	0.176	0.032
PopDensity	0.252	-0.076	-0.009	-0.149	0.084	-0.236	1.000	-0.007	0.253	-0.167	-0.003	0.422	0.158	0.264	0.339
%NonWhite	0.647	0.459	0.602	-0.119	0.303	-0.209	-0.007	1.000	-0.057	0.353	-0.101	0.160	0.019	0.148	0.208
%WC	-0.289	0.208	-0.013	0.015	-0.114	0.486	0.253	-0.057	1.000	-0.347	0.366	-0.063	0.129	0.162	0.108
pop/house	0.368	-0.325	0.257	-0.144	0.199	-0.389	-0.167	0.353	-0.347	1.000	-0.295	-0.010	-0.449	-0.224	-0.123
income	-0.283	0.198	-0.191	0.128	-0.362	0.507	-0.003	-0.101	0.366	-0.295	1.000	0.068	0.312	0.292	0.245
S02Pot	0.419	-0.094	-0.071	-0.116	-0.131	-0.229	0.422	0.160	-0.063	-0.010	0.068	1.000	0.406	0.569	0.681
NOx	-0.085	0.334	-0.334	-0.053	-0.460	0.229	0.158	0.019	0.129	-0.449	0.312	0.406	1.000	0.738	0.705
log_HCPot	0.125	0.227	-0.413	0.185	-0.485	0.176	0.264	0.148	0.162	-0.224	0.292	0.569	0.738	1.000	0.939
log_NOxPot	0.280	0.181	-0.303	0.097	-0.389	0.032	0.339	0.208	0.108	-0.123	0.245	0.681	0.705	0.939	1.000

유의한 예측변수 사전 진단 가능

```
cor_target=abs(df_cor['Mortality'])
#Selecting highly correlated features
relevant_features=cor_target[cor_target>0.3]
relevant_features
```

목표변수와 상관관계가 높은 예측변수 – 유의 가능성 높음

```
Mortality     1.000
JulyTemp      0.322
Rain          0.433
Education     0.508
%NonWhite     0.647
pop/house     0.368
S02Pot        0.419
```

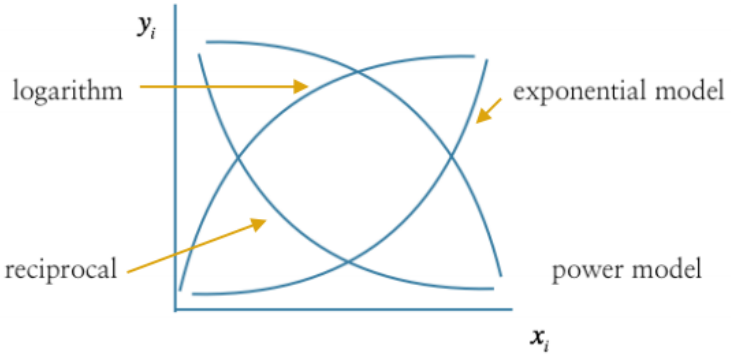
함수의 선형성 검토

개념

- 목표변수와 예측변수가 선형이 아닌 경우 - 선형회귀모형이므로 문제 해결 필요

진단도구

- 산점도 행렬 : 목표변수와 예측변수 간 개별 산점도 - 직선관계 발견



해결책 : 선형 변환 Linear Transformation

모형	식	Y변환	X변환
Power	$y = ax^b$	$ln(y)$	$ln(x)$
Exponential	$y = ae^{bx}$	$ln(y)$	x
logarithmic	$y = ln(ax^b)$	y	$ln(x)$
Reciprocal	$y = \frac{1}{a+bx}$	$\frac{1}{y}$	x
Square Root	$y = a + b\sqrt{x}$	y	$\sqrt{x}$
Square	$y = a + bx^2$	y	x^2

한계

단순회귀모형에서는 목표변수의 선형변환이 가능하나, 다중회귀모형에서는 불가능하다. 왜냐하면 한 예측변수와의 선형성을 수정하면 다른 예측변수와의 선형성 관계가 왜곡될 수 있기 때문이다.

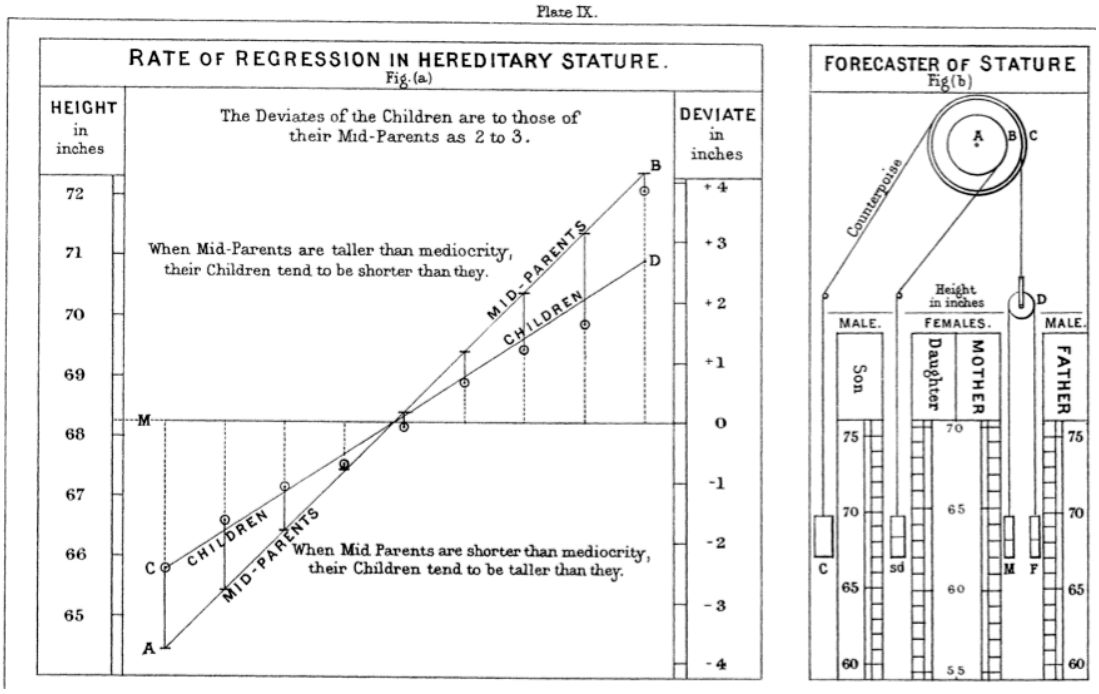
다중모형에서는 선형성 변환은 예측변수에 국한되며, 게다가 빅데이터, 예측변수가 많은 경우 선형성 진단은 필요하지 않다.

OLS 추정

개념

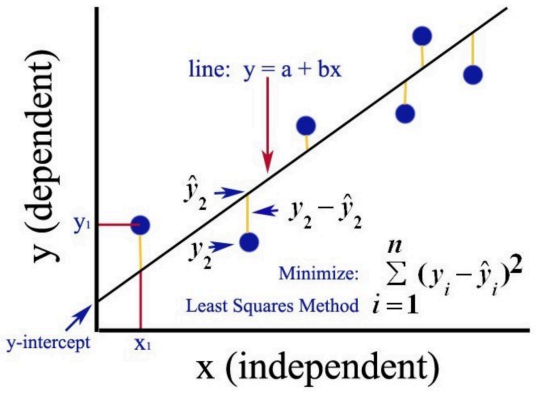
유전학자 Francis Galton은 처음에는 sweet peas 유전에 대해 연구하다가 자녀키, 부모키 평균(모 키에는 1.08배 하여 부 키와 평균을 내었음) 교차표를 만들어 자녀 키를 예측하려는 시도 하였다.

부모키가 평균에 가까운 경우에는 (양) 직선 관계가 존재하나, 부모의 키가 아주 크거나 작은 경우 자녀의 키는 직선적으로 커지거나 작아지지 않고 인간 키의 평균으로 회귀(regress)한다는 사실을 알게 되었다.



OLS 방법

Karl Pearson(피어슨 상관계수 제안자) 아버지, 성인아들 키를 이용하여 직선 관계식을 도출하는 방법을 제안하였다. 이를 최소자승법(Ordinary Least Squares) 추정방법이라 한다.



관측점들을 가장 대표하는 직선 (best fit)을 어떻게 구할 것인가?

데이터 $(x_i,\ y_i)$를 활용하여 점들에 가장 적합한 직선의 회귀계수 (a, b)를 추정하고 이를 이용하여 목표변수 추정값을 $\hat{y}_i = \hat{a} + \hat{b}x_i$(fitted value) 구한다. 오차항에 대한 추정값으로 $\hat{e}_i = y_i - \hat{y}_i$을 사용하여 직선의 유의성을 검증한다.

점들을 대표하는 (가장 적합하다고 판단되는) 추정식을 어떻게 구할 것인가?

$$\min_{a,b} \sum (e_i)^2 = \min_{a,b} \sum (y_i - \hat{y}_i)^2$$

수평, 수직 편차 중 왜 수직 편차인가?

- 수평편차는 예측변수의 변동에 대한 척도이고 수직편차는 목표변수의 변동을 척도이다.
- 선형모형은 목표변수의 변동(값)을 예측하는 것이 목표이므로 수직 편차를 최소화 하는 것이 점들을 대표하는 회귀식(적합식)이다.

왜 제곱인가?

- 수평 편차 $(y_i - \hat{y}_i)$의 합은 0이다. 그럼 왜 절대값을 사용하지 않나?
- 절대값은 수학적으로 다루기 쉽지 않으므로 1)회귀선에서 많이 벗어날수록 높은 페널티를 주고 2)수학적으로 다루기 쉬운 제곱을 사용한다.

수리적 해

$$\min_{\alpha,\beta} Q = \sum_i e_i^2 = \sum_i (y_i - \alpha - \beta x_i)^2$$

정규방정식

$$\frac{\partial Q}{\partial \alpha} = -2\sum_{i=1}^{n}(y_i - \hat{\alpha} - \hat{\beta}x_i) = 0$$

$$\frac{\partial Q}{\partial \beta} = -2\sum_{i=1}^{n} x_i(y_i - \hat{\alpha} - \hat{\beta}x_i) = 0$$

OLS 추정치

$$\hat{\beta} = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2} \quad \left(\hat{\beta} = \frac{S_{xy}}{S_{xx}} \right) \qquad \hat{\alpha} = \bar{y} - \hat{\beta}\bar{x}$$

상관계수와 관계 : $r = b\dfrac{\sqrt{\sum (x_i - \bar{x})^2}}{\sqrt{\sum (y_i - \bar{y})^2}}$

최대우도추정량 Maximum Likelihood Estimator

$y_i \sim iidN(\alpha + \beta x_i, \sigma^2)$ 이므로 우도함수는 다음과 같으므로 우도함수 최대화 = OLS 추정과 동일

$$L(\alpha, \beta; x_1, x_2, ..., x_n) = f(y_1, y_2, ..., y_n; \alpha, \beta)$$

$$= \Pi \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y_i - \alpha - \beta x_i)^2}{2\sigma^2}\right)$$

$$= \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^n \exp\left(-\frac{\sum (y_i - \alpha - \beta x_i)^2}{2\sigma^2}\right)$$

GAUSS-MARKOV Theorem : $OLS\ is\ BLUE$

회귀계수에 대한 OLS 추정치는 BLUE(Best Linear Unbiased Estimator)이다. 즉 모든 선형, 불편 추정량 중 최소 분산(minimum variance)를 갖는다. 분포함수가 지수족이므로 MLE는 CSS이고 불편추정량이므로 Rao-Blackwell 정리에 의해 MVUE이다.

잔차 = 오차의 추정량 $r_i = \hat{e}_i = y_i - \hat{y}_i$

- $\sum r_i = 0$: 잔차의 합은 0이다.
- $\sum x_i r_i = 0$: 관측치 x_i을 가중치로 계산한 가중평균은 0이다
- $\sum \hat{y}_i r_i = 0$: 예측치 $\hat{y}_i$을 가중치로 계산한 가중평균은 0이다
- 적합된 회귀직선은 $(\bar{x}, \bar{y})$을 지난다.

다중회귀 OLS 행렬 계산

행렬모형 : $\underline{y} = X\underline{b} + \underline{e}$, (가정) $\underline{e} \sim N(\underline{0}, \sigma^2 I_n)$

$$\underline{y} = \begin{pmatrix} y_1 \\ y_2 \\ \cdots \\ y_n \end{pmatrix}_{n\times 1}, \underline{e} = \begin{pmatrix} e_1 \\ e_2 \\ \cdots \\ e_n \end{pmatrix}_{n\times 1}, \underline{b} = \begin{pmatrix} \alpha \\ b_1 \\ b_2 \\ \cdots \\ b_p \end{pmatrix}_{(p+1)\times 1}, X = \begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p} \\ \cdots & & & & \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix}_{n\times p}$$

$$\mathbf{X}^{\mathsf{T}}\mathbf{X} = \begin{bmatrix} n & \sum x_{i1} & \sum x_{i2} & \cdots & \sum x_{ik} \\ \sum x_{i1} & \sum x_{i1}^2 & \sum x_{i1}x_{i2} & \cdots & \sum x_{i1}x_{ik} \\ \sum x_{i2} & \sum x_{i1}x_{i2} & \sum x_{i2}^2 & \cdots & \sum x_{i2}x_{ik} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sum x_{ik} & \sum x_{i1}x_{ik} & \sum x_{i2}x_{ik} & \cdots & \sum x_{ik}^2 \end{bmatrix} \quad \mathbf{X}^{\mathsf{T}}\mathbf{y} = \begin{bmatrix} \sum y_i \\ \sum x_{i1}y_i \\ \vdots \\ \sum x_{ik}y_i \end{bmatrix}$$

$$\min_{a, b_1, b_2, \ldots} \sum (e_i)^2 \rightarrowtail \min_{\underline{b}} \underline{e}'\underline{e} = \min_{\underline{\beta}} (\underline{y} - X\underline{b})'(\underline{y} - X\underline{b})$$

$$Q = (\underline{y} - X\underline{b})'(\underline{y} - X\underline{b}) = (\underline{y}' - \underline{b}'X')(\underline{y} - X\underline{b}) = \underline{y}'\underline{y} - \underline{y}'X\underline{b} - \underline{b}'X'\underline{y} + \underline{b}'X'X\underline{b}$$

Q를 최소화 하는 회귀계수(모수) $\underline{b}$를 찾기 위한 정규방정식 normal equation은 다음과 같다.

$\frac{\partial Q}{\partial \underline{b}} = -X'\underline{y} - X'\underline{y} + 2X'X\hat{\underline{b}} = 0$, 이를 정리하면 $\hat{\underline{b}} = (X'X)^{-1}(X'\underline{y})$ OLS이다.

OLS 추정치 성질 : GAUSS-MARKOV Theorem

회귀계수에 대한 OLS 추정치는 BLUE(Best Linear Unbiased Estimator)이다. 즉 모든 선형, 불편 추정량 중 최소 분산(minimum variance)를 갖는다. 분포함수가 지수족이므로 MLE는 CSS이고 불편추정량이므로 Rao-Blackwell 정리에 의해 MVUE이다.

[증명]

(1) OLS 추정치 $\hat{\underline{b}} = (X'X)^{-1}(X'\underline{y})$는 $\underline{y}$의 선형함수이다. **Linear**

또 다른 선형 추정치를 $\hat{\underline{b}^0} = (K+C)\underline{y}$라 하자. 단, $K = (X'X)^{-1}X'$

[궁극적으로 행렬 C은 0 행렬임을 보이면 된다.

(2) 불편성을 만족해야 하므로 $E(\hat{\underline{b}^0}) = E(K+C)\underline{y}) = E(K\underline{y}) + E(C\underline{y}) = \underline{b} + CX\underline{b}$

그러므로 $CX = 0$이어야 불편성을 만족한다.

(3) 최소분산성에 의하여

$$V(\underline{b}^0) = V[(K+C)\underline{y}] = (K+C)V(\underline{y})(K+C)' = \sigma^2(K+C)(K+C)'$$

이를 정리하면 $V(\underline{b}^0) = \sigma^2(KK' + CC') = V(\hat{\underline{b}}) + \sigma^2 CC'$이다.

$C'C$는 양반정치행렬(positive semi-definite matrix)이므로 음이 될 수 없으므로 OLS 추정치의 추정분산이 가장 적다. 그러므로 OLS는 최소분산 불편추정량(**MVUE-Best Unbiased Estimator**)이다. **(QED)**

OLS 평균과 (추정)분산

불편추정량 $E(\hat{\underline{b}}) = E((X'X)^{-1}X'\underline{y}) = (X'X)^{-1}X'E(\underline{y}) = (X'X)^{-1}X'X\underline{b} = \underline{b}$

추정분산 : $V(\hat{\underline{b}}) = V((X'X)^{-1}X'\underline{y}) = (X'X)^{-1}X')\sigma^2 I((X'X)^{-1}X')' = \sigma^2(X'X)^{-1}$

추정분산 추정치 : $\hat{V(\hat{\underline{b}})} = \hat{\sigma^2}(X'X)^{-1} = MSE(X'X)^{-1}$

예측변수 X^k 추정오차 $s(\hat{b}_k)$: $MSE(X'X)^{-1}$의 k번째 대각원소 양의 제곱근

MLE 추정

오차항의 가정 $\underline{e} \sim N(\underline{0}, \sigma^2 I_n)$이므로 $\underline{y} \sim N(\underline{X}\underline{b}, \sigma^2 I_n)$이다.

$\underline{y} \sim N(X\underline{b}, \sigma^2 I_n)$이므로 우도함수는 $L(\underline{y}; X, \underline{b}, \sigma^2) \propto e^{-\frac{1}{2\sigma^2}(\underline{y}-X\underline{b})'(\underline{y}-X\underline{b})}$이므로

로그 우도함수를 최대화 하는 MLE 추정치는 $\underset{\underline{\beta}}{min}(\underline{y} - X\underline{b})'(\underline{y} - X\underline{b})$을 최소화 하는 OLS 추정치와 동일하다. 그리고 MLE는 MVUE이므로 가장 좋은 추정치이다.

적합치 $\hat{y}_i$

회귀모형에 OLS 추정치를 대체한 모형식 : $\underline{\hat{y}} = X\underline{\hat{b}} = X(X'X)^{-1}X'\underline{y} = H\underline{y}$

[정의] Hat 행렬 $H = X(X'X)^{-1}X'$

- H 행렬은 대칭행렬($H' = H$)이며 멱등행렬($HH = H$)이다.
- 행렬 $(I - H)$도 대칭행렬이며 멱등행렬이다.

만약 $AA = A$가 성립하는 행렬 A를 멱등행렬(idempodent)이라 한다.
만약 A가 멱등행렬이면, $A^n = A$(n은 2보다 큰 정수)가 성립한다.

잔차 residual $r_i = y_i - \hat{y}_i$

관측치와 적합치의 차이로 오차항에 대한 추정치이다.

$$\underline{r} = \underline{y} - \underline{\hat{y}} = \underline{y} - H\underline{y} = (I - H)\underline{y}$$

- 잔차는 오차의 추정치이다. $\hat{e}_i = r_i$
- 잔차의 평균은 0이다.
- 잔차 분산 : $V(\underline{r}) = V((I - H)\underline{y}) = \sigma^2(I - H)$

회귀계수 추론방법 개념

분산분석적 접근

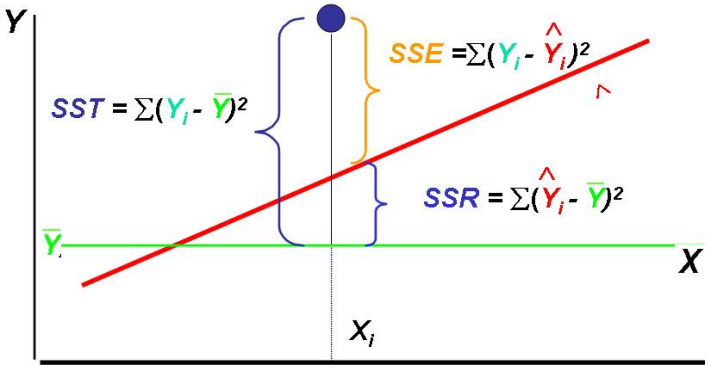
목표변수의 분산(총변동)을 모형이 설명변동과 설명하지 못하는 오차변동으로 이분화 하고 설명변동이 오차변동에 비해 충분히 크다면 회귀모형이 적절하다고(유의하다고) 판단한다.

총변동 : $SST = \sum (y_i - \bar{y})^2 = \sum y_i^2 - (\sum y_i)^2/n = \underline{y}'\underline{y} - (1/n)\underline{y}'J_n\underline{y}$

귀무가설 : 모든 회귀계수는 유의하지 않다.

$$\begin{pmatrix} b_1 \\ b_2 \\ \cdots \\ b_p \end{pmatrix}_{p\times 1} = \begin{pmatrix} 0 \\ 0 \\ \cdots \\ 0 \end{pmatrix}_{p\times 1}$$

대립가설 : 적어도 하나의 회귀계수는 유의하다. 적어도 $b_k \neq 0$이다.



변동분할 variation decomposition

총변동 Total Sum of Square

목표변수의 값들의 변동에 대한 척도로 예측변수들이 목표변수를 잘 설명한다는 것은 변화를 예측하는 것임 = 목표변수의 분산과 동일 개념

$$SST = \sum (y_i - \bar{y})^2 = \sum y_i^2 - (\sum y_i)^2/n = \underline{y}'\underline{y} - (1/n)\underline{y}'J_n\underline{y} = \underline{y}'(I - (1/n)J)\underline{y}$$

모형변동 Regression (Model) SS

목표변수 변동 중 설정된 모형에 의한 설명 부분이다.

총변동에 가까울수록 회귀모형 설정이 적절함을 의미한다.

$$SSR = \sum (\hat{y}_i - \bar{y})^2 = \underline{y}'(H - (1/n)J)\underline{y}$$

오차변동 Error SS

모형에 의해 설명되지 못하는 부분

$$SSE = \sum (y_i - \hat{y})^2 = (\underline{y} - H\underline{y})'(\underline{y} - H\underline{y}) = \underline{y}'(I - H)\underline{y}$$

이차형식

정의

확률변수 벡터 $\underline{y}$, 상수행렬 A에 대하여 $\underline{y}'A\underline{y}$을 이차형식이라 한다.

[이차형식 분포 정리]

만약 $\underline{y} \sim MVN(\underline{\mu} = \underline{0}, \Sigma = I)$(표준정규 다변량분포를 따른다면), 이차형식 $\underline{y}'A\underline{y}$은 자유도가 $rank(A)$ 인 카이제곱분포를 따른다.

그리고 평균 $E(Q) = \underline{y}'\mu\underline{y}$, 공분산 $COV(Q) = A\sigma^2A'$

변동분포

총변동 분포 $SST \sim \chi^2(n-1)$

- $SST = \underline{y}'(I-(1/n)J)\underline{y}$ 는 목표변수 $\underline{y}$의 이차형식이다.
- 오차항의 정규성 가정에 의하여 $\underline{y} \sim MVN(X\underline{b}, \sigma^2 I)$정규분포를 따르므로 이차형식인 SST는 자유도 $rank(I-J/n) = n-1$인 χ^2분포를 따른다.

오차변동 분포 $SSE \sim \chi^2(n-p-1)$

- $SSE = \underline{y}'(I-H)\underline{y}$는 목표변수 $\underline{y}$의 이차형식이다.
- 오차항의 정규성 가정에 의하여 $\underline{y} \sim MVN(X\underline{b}, \sigma^2 I)$정규분포를 따르므로 이차형식인 SSE는 자유도 $rank(I-H) = n-p-1$인 χ^2분포를 따른다.

자유도 분할: Cochran 정리

- 총변동의 자유도(관측치 중 자유로운 개수, 관측치 하나 하나는 독립적이고 정보를 갖고 있다)는 평균이 추정되었으므로 $(n-1)$이다. <=> 이차형식 결과와 동일
- SSE의 자유도는 $(n-p-1)$이다.
- $SST \sim \chi^2(n-1)$는 서로 독립인 $SSE \sim \chi^2(n-p-1)$와 SSR로 나뉘고 SSR도 $\underline{y}$의 이차형식이므로 카이제곱 분포를 따르고 Cochran 정리에 의해 자유도는 $(n-1)-(n-p-1) = p$(예측변수 개수)이다.
- $SSR \sim \chi^2(p)$

평균변동및 기대평균변동 Expected MSE

평균오차변동

$MSE = \frac{SSE}{n-p-1}$: 오차분산 σ^2의 추정값 - $\hat{\sigma^2} = MSE$

평균오차변동 기대값 : $E(\hat{\sigma^2}) = E(MSE) = \sigma^2$

평균회귀변동

$MSR = \frac{SSR}{p}$: 평균 회귀변동 : $E(MSR) = \sigma^2 + b^2 \sum (x_i - \bar{x})^2$ (단순회귀 경우)

회귀변동 값은 회귀계수가 0과 차이가 많거나 예측변수 값의 차이가 큰 경우 회귀변동 값은 커진다.

회귀변동이 커진다는 것은 회귀계수가 유의하든가 예측변수의 차이가 많다는 것을 의미한다.

분산분석표, F - 검정 : 설정 다중 회귀모형 유의성 검정 $H_0 : \underline{b} = \underline{0}$

- 귀무가설 : 설정한 회귀모형은 유의하지 않다. <=> 모든 예측변수의 회귀계수 값은 0이다.

  <=> $\underline{y} = X\underline{b}$ 적절하지 않다. <=> 목표변수는 설정된 모든 예측변수로 설명되지 않는다.

- 대립가설 : 설정한 회귀모형 $\underline{y} = X\underline{b}$은 유의하다.

  <=> 회귀계수 벡터 $\underline{b}$ 중 유의한 회귀계수가 적어도 하나가 있다.

- 검정통계량 : $TS = \frac{MSR}{MSE} \sim F(p, n-p-1)$

  그러므로 F-검정 결과 귀무가설이 기각되면 유의한 설명 변수가 하나 이상 있다는 것이므로 각 설명 변수에 대한 유의성을 t-검정을 이용하여 알아 보면 된다.

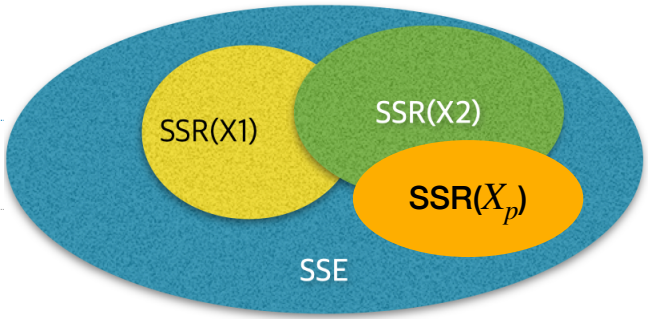
변동(source)	자승합 Sum of Sqaures	자유도 Df	평균자승합 Mean SS	F-통계량
회귀변동	$SSR = \underline{y}'(H - (1/n)J)\underline{y}$	p	$MSR = \frac{SSR}{p}$	$TS = \frac{MSR}{MSE} \sim F(p, n-p-1)$
오차변동	$SSE = \underline{y}'(I - H)\underline{y}$	$n-p-1$	$MSE = \frac{SSE}{n-p-1}$	
총변동	$SST = \underline{y}'(I - (1/n)J)\underline{y}$	$n-1$		

t - 검정 : 개별 예측변수 유의성 검정 $H_0 : b_k = 0$

- 귀무가설 : 예측변수 X_k는 유의하지 않다. $H_0 : b_k = 0$
- 대립가설 : 예측변수 X_k는 유의하다. $H_0 : b_k \neq 0$
- 검정통계량 : $TS = \dfrac{\hat{b}_k}{s(\hat{b}_k)} \sim t(n-p-1)$

예측변수 X^k 추정오차 $s(\hat{b}_k)$: $MSE(X'X)^{-1}$의 k번째 대각원소 양의 제곱근

두 개 이상 예측변수의 유의성 동시 검정



Full Model 예측변수 개수 p

$$Y_i = a + b_1X_{i1} + b_2X_{i2} + \ldots + b_pX_{ip} + e_i$$

Reduced Model

귀무가설 : 일부 예측변수가 유의하지 않다. 유의하지 않은 예측변수 개수 = k

만약 첫 2개 예측변수 $(X_1,\ X_2)$가 유의하지 않다면 귀무가설은 $H_0 : b_1 = b_2 = 0$이 된다. $k = 2$

Reduced model : $Y_i = a + b_3X_{i1} + b_4X_{i2} + \ldots + +b_pX_{ip} + e_i$

추가 자승합 Extra Sum of Sqaures 정의

- $SSR(X_1, X_2, \ldots, X_p)$: $Y_i = a + b_1X_{i1} + b_2X_{i2} + \ldots + b_pX_{ip} + e_i$ 완전모형의 회귀(모형)변동
- $SSE(X_1, X_2, \ldots, X_p)$: $Y_i = a + b_1X_{i1} + b_2X_{i2} + \ldots + b_pX_{ip} + e_i$ 완전모형의 오차변동
- $SSR(X_i \mid X_j) = SSR(X_i, X_j) - SSR(X_i)$

검정통계량 $H_0 : b_j = 0,\ for j = 1,2,...,k$

$$TS = \frac{(SSR_F - SSR_R)/k}{MSE_F} \sim F(k, n-p-1),\ k = \text{귀무가설에서 설정된 회귀계수 0인 예측변수 개수}$$

- Reduced Model : $Y_i = a + b_1X_{i1} + b_2X_{i2} + \ldots + b_pX_{ip} + e_i$ 모형에서 귀무가설 0 설정된 예측변수 제외됨

$H_0 : b_k = 0$ (예측변수 X_k 유의성 검정)

- Reduce Model : 완전모형에서 X_k 항이 제외됨
- $TS = \dfrac{SSR_F - SSR_R}{MSE_F} \sim F(1, n-p-1) = t^2(n-p-1)$ (t-검정과 동일)

Sequential 순차 자승합 SS, Type I

회귀모형 삽입 예측모형 순서 대로 자승합이 출력된다.

회귀모형 : $Y_i = a + b_1 X_{i1} + b_2 X_{i2} + \ldots + b_p X_{ip} + e_i$

- 삽입 순서에 따른 설명변동의 차이가 있음
- X_1 : $SSR(X_1 \mid \mu)$ - μ는 절편항을 의미한다.
- X_2 : $SSR(X_2 \mid X_1, \mu)$, …
- X_p : $SSR(X_p \mid X_1, X_2, \ldots, X_{p-1}, \mu)$
- 통제변수 모형에 주로 사용된다.

Type II: hierarchical or partially sequential 부분 자승합

다른 모든 예측변수가 고정인 경우 해당 변수의 변동

- $SSR(X_k \mid X_1, \ldots, X_{k-1}, X_{k+1}, \ldots, X_p, \mu)$
- 유의 변수 선택에 사용된다.
- 교차항이 있을 경우에는 Type III와 상이하다.

Type III: marginal or orthogonal , partial

타입 2 자승합과 동일하게 부분자숭합 개념이나 교차항이 있는 경우 상이하다.

실습 데이터 추론

```
df_clean.shape,df_clean.columns
```

최초 회귀모형 추정 데이터

```
((59, 15),
 Index(['Mortality', 'JanTemp', 'JulyTemp', 'RelHum', 'Rain', 'Education',
        'PopDensity', '%NonWhite', '%WC', 'pop/house', 'income', 'S02Pot',
        'NOx', 'log_HCPot', 'log_NOxPot'],
```

[방법1]

```
import statsmodels.api as sm
y=df_clean['Mortality']
X=sm.add_constant(df_clean.drop(columns=['Mortality'])) #add
intercept
fit=sm.OLS(y, X).fit()
fit.summary()
```

목표변수 : 사망율 지수 Mortality,예측변수 : 14개
유의한 예측변수 : 1월기온(-), 강우량(+), 비백인비율(+),사무직비율(-),로그_NOx

OLS Regression Results

Dep. Variable:	Mortality	**R-squared:**	0.779
Model:	OLS	**Adj. R-squared:**	0.708
Method:	Least Squares	**F-statistic:**	11.06
Date:	Sun, 25 Oct 2020	**Prob (F-statistic):**	3.52e-10
Time:	07:05:52	**Log-Likelihood:**	-282.62
No. Observations:	59	**AIC:**	595.2
Df Residuals:	44	**BIC:**	626.4
Df Model:	14		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	1411.8987	274.988	5.134	0.000	857.697	1966.100
JanTemp	-1.5872	0.747	-2.124	0.039	-3.093	-0.081
JulyTemp	-2.4543	2.029	-1.210	0.233	-6.543	1.634
RelHum	-0.1893	1.128	-0.168	0.867	-2.462	2.084
Rain	1.2786	0.568	2.250	0.029	0.134	2.424
Education	-6.5830	8.623	-0.763	0.449	-23.961	10.795
PopDensity	0.0037	0.004	0.897	0.374	-0.005	0.012
%NonWhite	5.1545	0.945	5.454	0.000	3.250	7.059
%WC	-2.0242	1.191	-1.699	0.096	-4.425	0.377
pop/house	-56.3598	39.773	-1.417	0.164	-136.518	23.798
income	-0.0004	0.001	-0.363	0.718	-0.003	0.002
S02Pot	0.0998	0.115	0.866	0.391	-0.132	0.332
NOx	-0.2581	0.191	-1.350	0.184	-0.643	0.127
log_HCPot	-21.8543	14.858	-1.471	0.148	-51.798	8.090
log_NOxPot	32.9805	13.847	2.382	0.022	5.073	60.888

분산분석 F-통계량=11.06

[방법2]

```
import statsmodels.api as sm
from statsmodels.formula.api import ols
fit_lm=ols('Mortality~JanTemp + Rain',data=df_clean).fit()
fit_lm.summary()
sm.stats.anova_lm(fit_lm,typ=2) # Type 2 ANOVA DataFrame
```

회귀모형을 적어야 하는 단점은 있으나 절편항 추가 필요 없고
범주형 변수의 경우 +C(범주형예측변수_이름) 적어주면 된다. [더미변수 만들 필요 없음]
그리고 분산분석표도 출력가능하다. *)변수가 10개 이하인 경우 사용하는 것이 편리하다.

```
순차적 SS : Type 1
sm.stats.anova_lm(fit_lm,typ=1)
# Type 1 ANOVA
```

	df	sum_sq	mean_sq	F	PR(>F)
JanTemp	1.000	57.505	57.505	0.018	0.895
Rain	1.000	42723.086	42723.086	13.059	0.001
Residual	56.000	183211.934	3271.642	nan	nan

마지막 예측변수(Rain)의 자승합은 $SSR(Rain|\mu(constant), JanTemp)$ 동일하다.
순차적 SS : $SSR(JanTemp|\mu)$
부분 SS : $SSR(JanTemp|\mu, Rain)$, 그러므로 항상 Type 1이 크다.

```
부분 SS : type 2, 3(절편,
intercept μ 까지 산출된다.)
2, 3의 차이는
sm.stats.anova_lm(fit_lm,typ=2)
```

	sum_sq	df	F	PR(>F)
JanTemp	387.127	1.000	0.118	0.732
Rain	42723.086	1.000	13.059	0.001
Residual	183211.934	56.000	nan	nan

Full Model vs Reduced Model 검정

Full 모형 : 사망률지수<-(1월기온,7월기온,강우량,습도)

REduced 모형 : 사망률지수<-(1월기온,7월기온) $H_0 : RelHum = Rain = 0$

```
import statsmodels.api as sm
y=df_clean['Mortality']
X_full=sm.add_constant(df_clean[['JanTemp','JulyTemp','RelHum','Rain']])
fit_full=sm.OLS(y,X_full).fit()   #Full model Fit
X_reduce=sm.add_constant(df_clean[['JanTemp','JulyTemp']])
fit_reduce=sm.OLS(y,X_reduce).fit() #Reduced model Fit
ts=(fit_reduce.ssr-fit_full.ssr)/fit_full.mse_resid   #test statistic
import scipy.stats as st
1-st.f.cdf(ts,1,fit_full.nobs-5,0,1) #p_value
```

유의확률이 0.05보다 작아 귀무가설 기각, (습도, 강우량) 동시에 유의하지 않다. 적어도 하나는 유의하다. $TS = \frac{(SSR_F - SSR_R)/k}{MSE_F} \sim F(k, n-p-1)$, $k = 2$(귀무가설 설정 예측변수 개수)

```
0.013239157499277288
```

변수선택 방법

개념

- 경험이나 이론에 의해 목표변수(Y)에 영향을 미칠 것 같은 예측변수를 선택하고 선형 회귀모형을 설정한다. => 데이터를 수집 후 회귀모형의 회귀계수를 추정하고(OLS) 분산분석의 F-검정에 의해 모형의 유의성을 검정한다.(H0 :β1 =β2 =...=βp =0,모든예측변수는유의하지않다) => F-검정(분산분석) 결과 귀무가설이 기각되면 t-검정을 이용하여 회귀계수(예측변수)의 유의성 검정을 한다.
- 오컴의 면도날(Occam's Razor) : 경제성의 원리(Principle of economy), 절약성의 원리(Principle of parsimony) - 모든 상황이 동일하다면 가장 간단한 방법이 최선이다. 즉, 회귀모형이 동일한 정보를 준다면(설정 모형이 예측변수의 변동을 설명하는 비율) 적은 예측변수 모형이 최선의 모형이다.
- 유의하지 않은 예측변수를 하나씩 제외하면서 모든 변수가 유의할 때까지 계속한다. 어떤 예측변수를 제외할 것인가? 가장 유의하지 않은 예측변수, 즉 t-검정통계량 값이 가 장 작은 것(p-value가 가장 큰 것)을 제외한다.

모형 유의성 검정 - F 검정

귀무가설 : 모든 예측변수가 유의하지 않음 - 선택되면 분석 종료하고 귀무가설이 기각되어 모형 내 설정한 예측변수 중 적어도 하나라도 유의하면 다음 단계로 넘어간다.

개별 예측변수 - t 검정

유의하지 않은 순서대로 하나씩 제외함 : 유의확률이 가장 크거나 t-통계량이 가장 작은 순서대로 모형 내 모든 예측변수가 유의할 때까지 반복 제거함

다중공선성 분석과 순서 문제

대부분의 교과서에서는 다중공선성 진단을 먼저 설명하고 있으나 어느 단계를 먼저해도 동일 결과를 얻는 경우가 대부분이므로 간편 작업(변수선택 과정을 먼저 거치면 다중공선성 진단을 하면 진단 변수의 수 줄어듬)을 위하여 변수선택을 먼저하는 것이 적절하다.

반드시 필요한가?

예측이 목적이라면 설명력이 전혀 없는(목표변수와 상관계수 0) 예측변수는 필드에서는 존재하지 않으므로 미미하지만 목표변수를 설명하므로 선택, 제외할 필요는 없다. 그러나 너무 많은 변수가 존재하면 과적합 문제가 발생하므로 데이터 개수에 비해 변수의 개수가 많아지는 경우 차원 축소 측면에서 변수선택은 필요하다.

통계량 이용

결정계수 Determination Coefficient $R_p^2 = \frac{SSR_p}{SST_p}$

결정계수는 모형내의 p개 예측변수들이 목표변수의 (총)변동(목표변수 관측값들의 변동)에 대한 설명력의 정도를 나타내는 수치이므로 변수선택의 지표가 된다.

결정계수의 크기가 70%이면 설정된 예측변수의 설명력이 충분함, 80% 이상이면 매우 충분하다고 판단함, 그리고 90% 이상이면 완전에 가까운 설명력을 가짐 - 현재 설정된 예측변수만으로 목표변수 변동을 매우 잘 예측할(설명) 수 있음

예측변수의 개수가 같은 경우 어떤 변수 그룹이 설명력이 높은가를 쉽게 알아보는 사용할 수 있으나 검정 통계량은 존재하지 않는 단점이 있다.

그러나 예측변수 개수가 증가할 때 마다 결정계수는 항상 증가하므로 변수의 개수가 다른 경우에는 다음의 수정결정 계수를 사용하는 것이 좋다.

수정결정계수 adjusted $R_{adj}^2 = 1 - \frac{(1-R^2)/(n-1)}{(n-p-1)}$

수정(adjusted) 결정계수는 결정계수의 문제점(유의하지 않은 예측변수가 삽입되어도 항상 증가)을 해결하기 위하여 계산되는 척도

그러나 예측변수의 증가로 줄어든 SSE의 값과 오차 자유도의 감소가 상쇄되므로 설명력의 지표로 좋은 것은 아니다. 그리고 유의성을 검정할 검정통계량이 존재하지 않는다.

수정결정계수는 예측변수 군과 개수가 서로 다른 회귀모형의 설명력을 비교할 때 사용된다.

목표변수의 예측 값을 높이는 모형을 선택하는 것이 분석 목적이라면 이 방법을 이용하여 최적 모형을 찾으면 됨

Mallow $C_p = \frac{SSE}{MSE} - n + 2(p+1)$, $C_p = \frac{(SSE - 2*p*MSE)}{n}$

Mallow C_p 값이 예측변수 개수(p)와 근사한 경우 좋은 회귀 모형으로 판단한다.

여전히 이 방법에서도 각 예측변수의 유의성에 대한 검정을 실시한 것은 아니다.

(이론적 근거) $MSE = \hat{\sigma}^2$이고 만약 모형이 적합하다면 $C_p \approx (p+1)$이다.

예측잔차자승합(Prediction REsidual Sum of Squares) $PRESS = \sum (y_i - \hat{y_{(i)}})^2$

$\hat{y_{(i)}}$: i-번째 관측치를 제외한 후 모형을 추정한 후 구한 목표변수 y_i의 적합값

i-번째 관측치 제외 잔차 = $(y_i - \hat{y}_{(i)})$

잔차자승합 PRESS 값이 적을수록 좋은 모형

AIC, SBC 모두 작을수록 적합도가 높다.

AIC(Akaike Information Criteria)

$AIC = 2p - 2ln(\hat{L})$, $\hat{L}$: 회귀모델의 최대우도함수

BIC(Bayesian information criterion)

$BIC = p * ln(n) - 2ln(\hat{L})$

Comment

(수정)결정계수, Mallow , PRESS 통계량 이용 방법은 어떤 변수 군들이 좋은지에 대한 지표만을 제공할 뿐 변수의 유의성은 검정되지 않았다.

일반적으로 이 방법들을 이용할 때는 수정 결정 계수와 Mallow C_p에 의해 좋다고 간주되는 변수 군들을 몇 개 선택하고 각 변수 군에 대해 PRESS 값을 계산하여 최적 변수 군을 선택하면 된다.

AIC,BIC, PRESS는 전혀 다른 변수 군을 비교할 때(변수들이 모두 유의한 경우) 사용되므로 널리 사용된다. 서로 다른 변수 군의 적합 정도 비교 시 사용된다.

AIC,BIC는 빅데이터 예측모형 비교 시 주로 사용된다.

부분 결정계수 Coefficient of Partial Determination

예를 들어 3개의 예측변수가 존재하는 X_1, X_2, X_3 모형을 가정하자. X_1에 의해 설명되고 남은 예측변수 변동을 두 예측변수(X_2, X_3)가 설명하는 변동을 다음과 같이 정의하고 이를 부분 결정계수라 한다.

$$R^2_{Y.2,3|1} = \frac{SSR(X_2, X_3 \mid X_1) = SSR(X_1, X_2, X_3) - SSR(X_1)}{SSE(X_1)} = \frac{SSE(X_1) - SSE(X_1, X_2, X_3)}{SSE(X_1)}$$

고전적 변수선택 방법

Backward 후진제거

모든 예측변수를 고려한 모형에서 유의하지 않은 예측변수를 하나씩 제거하는 방법이다.

- 고려된 예측변수를 모두 삽입한 후 예측변수 중 가장 유의하지 않은(유의확률이 가장 큰) 예측변수를 제외
- 가장 유의하지 않다는 것은 다음 검정 결과 유의하지 않고 F값이 가장 작은 예측변수를 의미한다.

$$\frac{SSE_R(no\ X_k) - SSE_F}{MSE_F} \sim F(1, n-p-1)$$

- 모든 예측변수가 유의할 때까지 위의 과정을 반복한다.

전진삽입 forward

1) 고려된 예측변수 중 설명력(SSR_k)이 가장 높고(유의확률이 가장 작음) 설명력이 유의하면 변수를 선택한다.

$$\cdot max \frac{SSR(X_k)}{MSE(X_k)} \sim F(1, n-2)$$

2) 이미 선택된 예측변수(X_k)의 설명 부분을 제외하고 남은 목표변수 변동 중 가장 많이 설명하고 ($SSR(X_m \mid X_k)$이 가장 큰) 그 설명력이 유의한 경우 X_m을 2번째로 선택한다.

$$\cdot max \frac{SSR(X_k, X_m) - SSR(X_k)}{MSE(X_k, X_m)} \sim F(1, n-2)$$

유의한 예측변수가 없을 때까지 2)의 작업을 반복한다.

stepwise 단계삽입

단계 삽입(stepwise)은 Forward 방법과 유사하지만 한 번 선택된 설명 변수에 대해서는 유의성 검정을 다시 실시한다는 점이 다르다.

- 고려된 예측변수 중 설명력($SSR(X_k)$)이 가장 높고(유의확률이 가장 작음) 설명력이 유의하면 변수를 선택한다. = 첫단계는 전진삽입 방법과 동일함
- 이미 선택된 예측변수(X_k)의 설명 부분을 제외하고 남은 목표변수 변동 중 가장 많이 설명하고 ($SSR(X_m \mid X_k)$이 가장 큰) 그 설명력이 유의한 경우 X_m을 2번째로 선택한다. = 2번째 단계까지도 전진삽입 방법과 동일하다.
- 새롭게 선택된 변수(X_m)를 이용하여 기존 선택된 예측변수(X_k)의 유의성을 검증하여(즉 X_m이 설명하고 남은 목표변수의 변동을 이미 삽입된 X_k 변동이 유의한지 검증) 만약 유의하지 않으면 을 제거한다.
- 유의한 예측변수가 존재하지 않을 때까지 위의을 반복한다.

실습 데이터 [후진제거방법]

```
df_clean.shape,df_clean.columns
```

```
((59, 15),
 Index(['Mortality', 'JanTemp', 'JulyTemp', 'RelHum', 'Rain', 'Education',
        'PopDensity', '%NonWhite', '%WC', 'pop/house', 'income', 'S02Pot',
        'NOx', 'log_HCPot', 'log_NOxPot'],
```

[후진제거]

```
y=df_clean['Mortality']
X=df_clean.drop(columns=['Mortality'])
#Backward Elimination
sig_level=0.2
cols = list(X.columns)
pmax = 1
while (len(cols)>0):
    p= []
    X_1 = X[cols]
    X_1 = sm.add_constant(X_1)
    model = sm.OLS(y,X_1).fit()
    p = pd.Series(model.pvalues.values[1:],index = cols)
    pmax = max(p)
    feature_with_p_max = p.idxmax()
    if(pmax>sig_level):
        cols.remove(feature_with_p_max)
    else:
        break
selected_features_BE = cols
print(selected_features_BE)
```

빅데이터 변수선택의 경우 고전적 변수선택방법(전진, 후진, 전진삽입)은 자주 사용하지 않아 파이썬은 고전적 방법에 대하여 모듈인 아닌 코딩 함수로 표현하다.
가장 사용 편리한 것이 후진제거이고 빅데이터방법 대부분이 전진삽입 방법이므로 전진삽입 방법은 빅데이터 모듈을 사용하면 된다.
유의수준 0.2로 하여 적어도 20%에서 유의한 변수만 선택한다. (0.15~0.3 권장, 가능하면 많은 예측변수 선택하자.)
최종 예측변수 유의성 분석은 1)다중공선성 문제 진단 2)회귀진단 후 결정된다.

```
['JanTemp', 'Rain', 'Education', '%NonWhite', '%WC', 'pop/house', 'NOx', 'log_NOxPot']
```

```
import statsmodels.api as sm
y=df_clean['Mortality']
X=sm.add_constant(df_clean[selected_features_BE])
fit=sm.OLS(y, X).fit()
fit.summary()
```

회귀분석 결과 예상대로 모든 설명변수는 20%에서 유의하다.

	coef	std err	t	P>|t|	[0.025	0.975]
const	1282.3987	168.975	7.589	0.000	943.002	1621.796
JanTemp	-2.2540	0.611	-3.690	0.001	-3.481	-1.027
Rain	1.5351	0.538	2.853	0.006	0.454	2.616
Education	-12.2127	7.147	-1.709	0.094	-26.569	2.143
%NonWhite	4.5873	0.749	6.125	0.000	3.083	6.092
%WC	-1.8660	1.046	-1.784	0.080	-3.967	0.235
pop/house	-63.9892	36.147	-1.770	0.083	-136.593	8.615
NOx	-0.2512	0.164	-1.530	0.132	-0.581	0.078
log_NOxPot	23.6629	5.931	3.989	0.000	11.749	35.576

결정계수 $R^2 = 75.8\,\%$

최종 회귀모형 선택 방법 : 모형비교

서로 예측변수가 고려된 모형 중 최적 모형을 선택하고자 할 때는 MAE(Mean Absolute Error 오차평균절대), MSE(Mean Sqaured Error 오차평균자승합), RMSE(Root Mean Sqaured Error 자승합 제곱근) 나 수정결정계수 값을 비교한다.

표본데이터 크기 : n이고 OLS 목표변수 적합치(추정치) : $\hat{y}_i$인 경우

$$\cdot MAE = \frac{1}{n}\sum |Y_i - \hat{Y}_i|$$

$$\cdot MSE = \frac{1}{n}\sum (Y_i - \hat{Y}_i)^2, RMSE = \sqrt{(MSE)}$$

새로운 자료 (표본크기 n^*) 수집하고 모형의 예측력을 다음 방법에 의해 계산하여 각 모형을 비교한다. 평균자승합예측오차 Mean Square Prediction Error $MSPE = \frac{1}{n^*}\sum^{n^*} (y_i - \hat{y}_i)^2$

새로운 자료로 회귀 모형을 추정하였을 때 이전 자료에서 추정된 회귀 모형과 유사하면 추정된 회귀 모형은 좋다고 판단할 수 있다. 새로운 자료 수집이 불가능한 경우에는 데이터를 splitting하여 모형추정 데이터와 예측 데이터로 나누어 분석한다.

수정결정계수나 PRESS, AIC, SBC에 의한 최적 회귀모형 선택은 예측변수가 서로 다른 그룹을 비교할 때 사용되는 통계량이다.

SSR 활용 최적모형 탐색 <=> 결정계수 최대와 동일함

```
def processSubset(feature_set):
    # Fit model on feature_set and calculate RSS
    model = sm.OLS(y,X[list(feature_set)])
    regr = model.fit()
    RSS = ((regr.predict(X[list(feature_set)]) - y) ** 2).sum()
    return {"model":regr, "RSS":RSS}
def getBest(k):
    tic = time.time()
    results = []
    for combo in itertools.combinations(X.columns, k):
        results.append(processSubset(combo))
    # Wrap everything up in a nice dataframe
    models = pd.DataFrame(results)
    # Choose the model with the highest RSS
    best_model = models.loc[models['RSS'].argmin()]
    toc = time.time()
    print("Processed", models.shape[0], "models on", k,
"predictors in", (toc-tic), "seconds.")
    # Return the best model, along with some other useful
information about the model
    return best_model
import time
import itertools
models_best = pd.DataFrame(columns=["RSS", "model"])
tic = time.time()
for i in range(1,X.shape[1]):
  models_best.loc[i] = getBest(i)
  # subet preocessing ...
```

SSR(설명변동) 기준 1개일 때 최적(SSR 가장 큰) 예측변수, 2개 일 때 가장 최적 변수군, …, 이런 식으로 예측변수 개수만큼 늘려가면서 최적 변수군을 선택하여 models_best 저장한다. 변수 개수가 7개인 최적 변수군을 출력하면 다음과 같다.

```
models_best.loc[8,"model"].model.exog_names #상수 constant 포함
```

```
print(models_best.loc[8,"model"].summary()) #회귀추정 결과 출력
```

```
==============================================================
                 coef    std err          t      P>|t|
--------------------------------------------------------------
const        955.5930     47.829     19.980      0.000
JanTemp       -1.8070      0.525     -3.441      0.001
Rain           1.7320      0.506      3.420      0.001
PopDensity     0.0061      0.004      1.680      0.099
%NonWhite      4.1307      0.619      6.672      0.000
%WC           -2.5796      0.950     -2.716      0.009
log_HCPot    -15.6012     12.664     -1.232      0.224
```

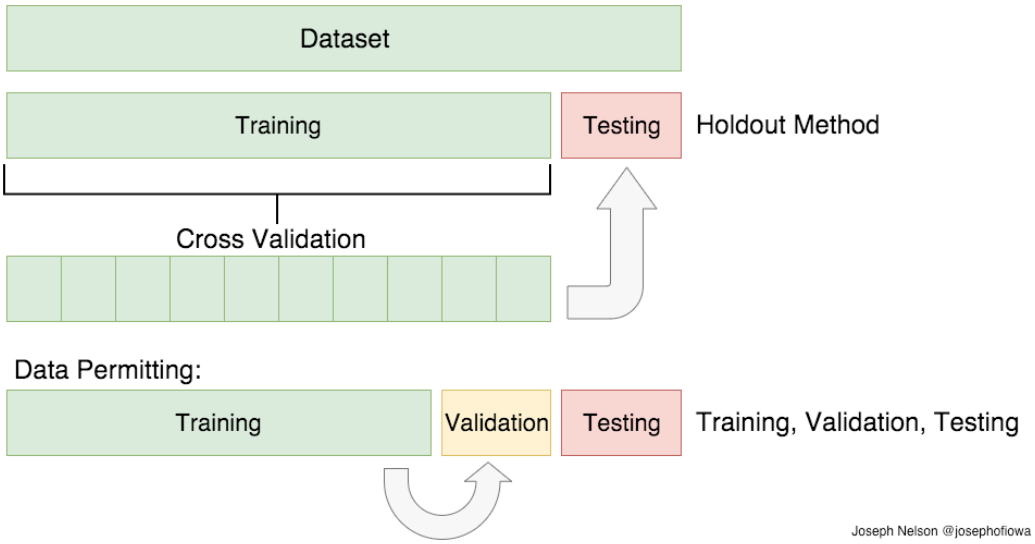
결정계수 $R^2 = 75\,\%$

빅데이터 접근 변수선택

Data Split

train and test data

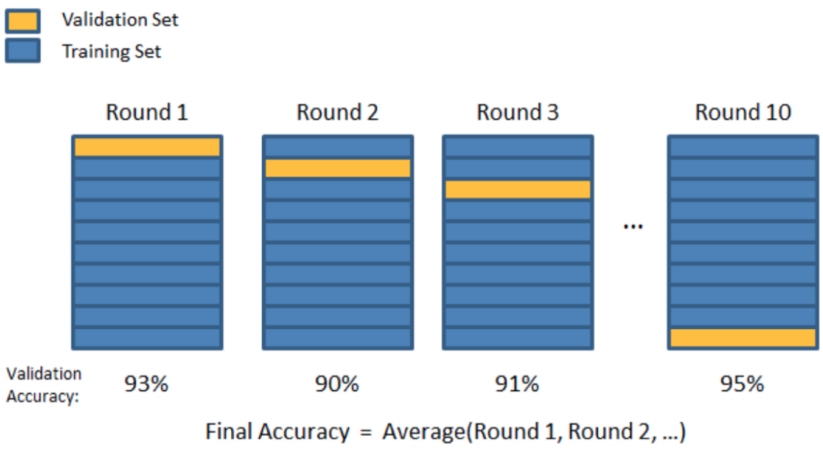
대용량 빅데이터의 경우에는 데이터 세트를 2분화 하여 모형을 추정하는 데이터(훈련 데이터 train dataset), 추정 모형을 이용하여 예측 정확도를 판단하는데(평가) 활용되는 데이터(검증 test dataset)로 활용한다. 일반적으로 8:2 혹은 7:3으로 나눈다. 이는 예측모형의 과적합과 underfitting 문제를 해결하기 위하여 머신러닝 기법에서 제안된 방법이다. 이방법은 전혀 새롭지 않은 개념이다. 이전 데이터마이닝에서 데이터를 3단계, train, validation, test 데이터로 나눈 것과 유사하다.



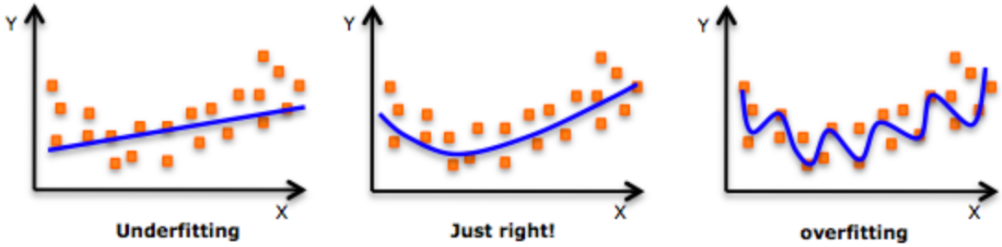
cross validation 필요 이유

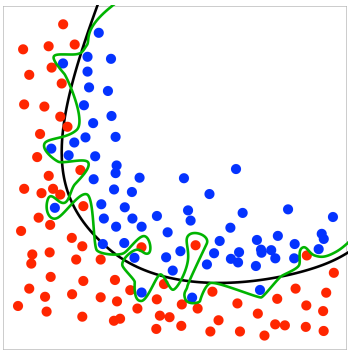
훈련 데이터와 검증 데이터의 스플릿이 무작위가 아닌 경우 문제가 발생한다. 훈련 데이터가 특정 변인에 의해 왜곡되어 있다면 과적합 문제가 발생하다. 예를 들어 우연히 소득수준이 높은 집단이 훈련 데이터에 많이 포함되어 있다면? 한 번의 데이터 스플릿에 의한 분석은 왜곡된 정보를 준다.

K-Folds Cross Validation에서 훈련 데이터를 k 개의 다른 부분 집합으로 나눈다. k-1 부분 집합을 사용하여 데이터를 훈련시키고 마지막 부분 집합을 테스트 데이터로 하여 검증하게 된다. 이런 과정을 k번 거쳐 계산된 정확도 평균을 훈련 데이터 정확도로 하게 되며 그런 다음 검증 데이터 세트에 대해 검증한다.



과적합과 과소적합





overfitting

과적합은 우리가 추정한 모형이"너무 잘 훈련"되었고 현재는 훈련 데이터 세트를 가장 잘 설명하고 있다는 것을 의미한다. 위의 오른쪽 두 추정 모형을 보면 과적합으로 인하여 데이터에 범용(향후 검증 데이터에 적용 가능한지? 혹은 일반 상황에도 적용 가능한지?) 적합한 모형인지 이 훈련 데이터에만 적합한지 알 수 없다는 사실이다.

관측치 수에 비해 너무 많은 특징(변수)가 존재하는 경우 발생하며 추정 모형은 훈련 데이터에서 매우 정확하지만 훈련되지 않은 또는 새로운 데이터에서는 정확하지 않을 수 있다.

이 추정모형의 결과를 일반화하고 다른 데이터, 즉 궁극적으로 수행하려는 작업에 대해 추론 할 수 없음을 의미한다. 기본적으로 과적합이 발생하면 모델은 데이터의 변수 간 실제 관계 대신 학습 데이터의 "노이즈"를 학습하거나 설명하게 되므로 이 노이즈가 포함되어 있지 않은 (검증)데이터 세트에는 적용할 수 없게 된다.

underfitting

과적합과 대조적으로, 모델이 과소적합되면 모델이 훈련 데이터에 적합하지 않으므로 데이터의 추세를 놓치게 되며 또한 추정 모형을 새 데이터로 일반화 할 수 없음을 의미한다.

이는 매우 간단한 모형 (충분한 예측 변수 / 독립 변수가 아님)의 결과로 선형이 아닌 데이터에 선형 모델 (예 : 선형 회귀)을 적용 할 때도 발생할 수 있다. 이 모델은 예측 능력이 좋지 않을 것이며 훈련 데이터에서 다른 데이터로 일반화 할 수 없음을 알 수 있다.

L1 and L2 Regularization

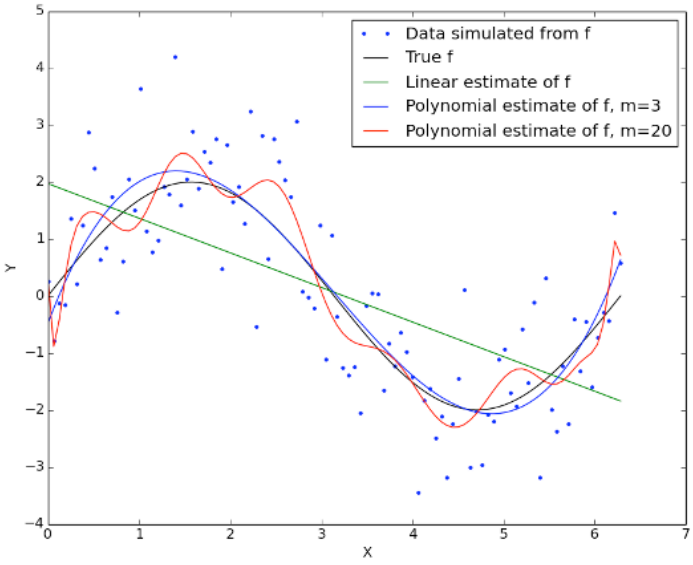
과적합 문제를 해결하기 위한 전처리 작업에 해당된다.

Norm $||X_p|| = (\sum |x_i|^p)^{1/p}$

벡터의 크기 혹은 두 벡터의 거리를 측정하는 함수이다. p=놈의 차수를 의미하며 p=1이면 L_1 놈, p=2이면 L_2놈이라 한다.

- L_1놈 : 맨하턴(Manhattan) 거리이다.
- L_2 놈 : 유클리디안(Eucleadian) 거리이다.

Regularization 정규화, 일반화 개념



모형을 추정한다는 것은 관측치에 가장 적합한 모형을 찾는 것인데 이는 원래 함수가 f임에도(검은 색) 불구하고 과적합 결과는 레드 함수로 추정되어 있다. 왜? 더 많은 정보를 사용하고도 원 함수와 다른 결과가 발생하나? 이는 (원 함수에서 관측된) 관측값(검은 점)에 가장 적합한 함수를 추정하였으므로 이런 현상이 발생한다.

함수를 추정하는 경우에는 관측값과 예측값에 의해 계산되는 손실함수(loss function, cost)를 최소화하는 추정함수를 최적 추정함수라 한다.

Loss function 손실함수 $L(y_i, \hat{y}_i)$

L1 손실함수 : $L(y_i, \hat{y}_i) = \sum |y_i - \hat{y}_i|$

- Least absolute deviations(LAD), Least absolute Errors(LAE)

L2 손실함수 : $L(y_i, \hat{y}_i) = \sum (y_i - \hat{y}_i)^2$

- Least squares error(LSE)

어떤 손실함수를 사용할 것인가? 이상치가 존재하는 경우 이상치에 덜 민감한 (resistant to outliers) L1 손실함수가 적절하나 미분이 가능하고(수학 연산이 용이함) 멀리 떨어진 관측값에 보다 큰 손실을 적용하는 L2 손실함수가 가장 널리 사용된다.

L1 Regularization

Lasso Regression (Least Absolute Shrinkage and Selection Operator)

손실함수 $OLS(L(y_i, \hat{y}_i)) + \lambda \sum |\beta_j|$

- λ=0이면 OLS 추정과 동일하고 λ가 커지면 회귀계수 영향이 작아져 과소적합이 된다.
- 목표변수에 영향도가 적은 예측변수의 회귀계수를 크기를 최소화 하고 영향도가 큰 예측변수의 회귀계수 크기를 상대적으로 크게 하여 주요 예측변수 'feature selection'에 사용된다.

L2 Regularization

Ridge Regression 능형 회귀분석

손실함수 $OLS(L(y_i, \hat{y}_i)) + \lambda \sum |\beta_j^2$

- λ=0이면 OLS 추정과 동일하고 λ가 커지면 불편성이 발생하고 MSE를 최소화 한다.
- 다중공선성 문제를 해결하는 정규화 방법이다.

Features Enginerring

Feature 선택

Feature 순위 또는 Feature 중요도 부여하는 과정으로 중요 변수(주요 신호)의 부분집합을 선택하거나, 불필요한(중요하지 않은) 변수들을 제거하여 변수의 차원(개수)을 줄이는데 그 목적이 있다. 분류모델 중 Decision Tree 같은 경우는 트리의 상단에 있을수록 중요도가 높으므로 이를 반영하여 특징 별로 중요도를 매길 수 있다. 회귀모델의 경우에는 유의 변수선책 알고리즘을 통해 변수 선택이 가능하다.

Feature 선택기법에는 상관분석과 모델기반 방법인 Lasso Regression, Recursive Feature Elimination, Tree-based Model, (Logistics) Discriminant Analysis 등이 있다

Feature 추출

원데이터 변수들의 구조적 관계를 활용하여 새로운 변수를 생성하거나(고차원의 공간을 저차원의 새로운 공간으로 차원축소) 변수들 간의 상관계수를 변수 유사성 척도로 하여 Feature를 탐색한다. PCA, SVD 차원축소 방법도 여기에 해당하며, 상관관계 유의한 변수군을 만드는 Canonical Correlation Analysis(CCA) 등이 Feature 추출 과정의 방법론이다.

LASSO 방법

데이터 값을 축소(shrinkage) - 중심점 평균으로 중심화 하여(L1 정규화 $OLS(L(y_i, \hat{y}_i)) + \lambda \sum |\beta_j|$) 목표변수에 영향을 많이 미치는(목표변수 변동을 보다 잘 설명하는) 예측변수를 선택하거나 다중공선성 문제 진단하는 데도 사용된다.

손실함수 $OLS(L(y_i, \hat{y}_i)) + \lambda \sum |\beta_j|$ 을 최소화 하는 것은 $\sum |\beta_j| \leq s$ 제약 조건 하에서 OLS의 SSE를 최소화 하는 추정치를 구하는 것과 동일하다.

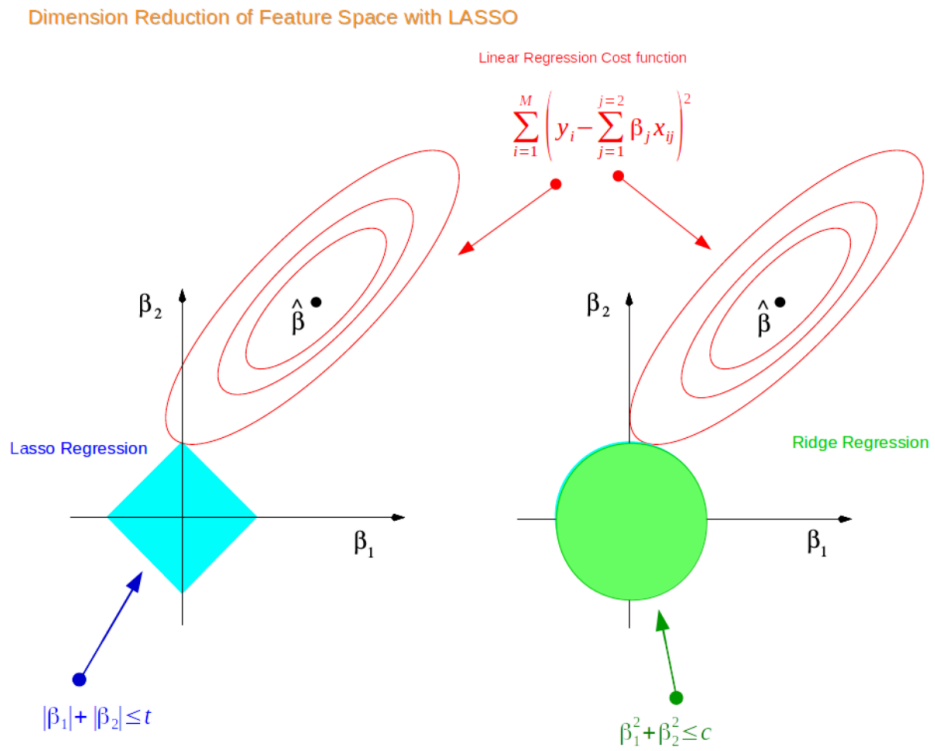
λ 결정

- $\lambda = 0$: OLS 추정과 동일, 모든 예측변수가 추정된다.
- $\lambda = \infty$: 모든 예측변수의 회귀계수가 0이 되어 모든 예측변수 제거된다.
- λ 가 증가하면 추정 편이는 증가하고 제거되는 예측변수의 개수는 많아진다.
- 최적 람다는 AIC, BIC를 최소화 하는 람다를 구한다.

Minimizing Information Criteria

$AIC_{ridge} = nln(\underline{e}'\underline{e}) + 2df_{ridge}$, $BIC_{ridge} = nln(\underline{e}'\underline{e}) + 2df_{ridge}ln(n)$

$df_{ridge} = tr(X'X + \lambda I)$의 자유도, OLS의 경우 $df_{OLS} = tr(X'X) = p$이다.



모형 개선 방안[파이썬 코드 활용]

Univariate Selection : 전진삽입방법과 동일

목표변수의 변동에 대한 예측변수의 설명력의 검정 통계량인 F-통계량의 크기로 설명력이 높은 변수를 선택

- 회귀모형의 변수 선택은 f_regression 이며 F-통계량에 의해 변수 선택된다. k 옵션은 최대 유의한 변수 개수 설정이며 디폴트는 10이다. 최대 개수와 유의한 변수 개수는 다르다.
- 결과를 보면 유의한 변수 10개까지 (pop_density_log) 이다. 유의확률 5% 이하
- 결과를 보면 상관계수 크기 순이고 상관계수 유의학률과 동일하다.
- 만약 범주형 판별분석의 유의한 예측변수를 선택하는 경우 chi2를 사용하며 y는 반드시 정수이어야 한다. y = np.array(y).astype(int)

```
df_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 59 entries, Akron to Youngstown-Warren
Data columns (total 15 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Mortality    59 non-null     float64
 1   JanTemp      59 non-null     int64
 2   JulyTemp     59 non-null     int64
 3   RelHum       59 non-null     int64
 4   Rain         59 non-null     int64
 5   Education    59 non-null     float64
 6   PopDensity   59 non-null     int64
 7   %NonWhite    59 non-null     float64
 8   %WC          59 non-null     float64
 9   pop/house    59 non-null     float64
 10  income       59 non-null     float64
 11  S02Pot       59 non-null     int64
 12  NOx          59 non-null     int64
 13  log_HCPot    59 non-null     float64
 14  log_NOxPot   59 non-null     float64
```

예제 데이터 n=59개 도시
목표변수 : Mortality
예측변수 : 나머지 변수

- 최적 변수 개수를 7로 하였음 – 하여, 랭킹에는 7개만 1, 나머지는 2, 3, 4 … 순으로 선택된다.

```
import numpy as np
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import f_regression
y=df_clean['Mortality']
X=df_clean.iloc[:,1:15]
bestfeatures = SelectKBest(score_func=f_regression, k=10)
#default k=10
fit=bestfeatures.fit(X,y)
dfpvalues = pd.DataFrame(fit.pvalues_)
dfscores = pd.DataFrame(fit.scores_)
dfcolumns = pd.DataFrame(X.columns)
#concat two dataframes for better visualization
featureScores = pd.concat([dfcolumns,dfscores,dfpvalues],axis=1)
featureScores.columns = ['Specs','Score','pvalues']  #naming the
dataframe columns
print(featureScores.nlargest(7,'Score'))  #print best features
```

```
          Specs  Score  pvalues
6    %NonWhite 40.944    0.000
4    Education 19.835    0.000
3         Rain 13.161    0.001
10     S02Pot 12.146    0.001
8    pop/house  8.929    0.004
1     JulyTemp  6.586    0.013
7          %WC  5.208    0.026
```

[SSR 방법Best Model]

```
['const',
 'JanTemp',
 'Rain',
 'PopDensity',
 '%NonWhite',
 '%WC',
 'log_HCPot',
 'log_NOxPot']
```

다름

```
select_vars=featureScores.loc[featureScores['pvalues']<0.1,:]['Specs']
select_vars
```

개별 예측변수의 유의확률(즉 목표변수와 상관계수 유의성과 동일) 10% 미만이나 예측변수가 동시에 고려되면 유의하지 않을 수 있음 [강우량, 인구밀도, 비백인비율, 가구원수, 가구소득, LOG_NOxPot] 등은 유의하지 않음

```
import statsmodels.api as sm
y=df_clean['Mortality']
X=sm.add_constant(df_clean[select_vars])
fit=sm.OLS(y, X).fit()
fit.summary()
```

```
1        JulyTemp
3            Rain
4       Education
5      PopDensity
6       %NonWhite
7              %WC
8       pop/house
9          income
10         S02Pot
13     log_NOxPot
```

[결정계수=71.5%]

	coef	std err	t	P>|t|	[0.025	0.975]
const	1004.8457	216.614	4.639	0.000	569.314	1440.378
JulyTemp	-2.0680	1.683	-1.229	0.225	-5.451	1.315
Rain	1.6587	0.588	2.820	0.007	0.476	2.841
Education	-3.8393	9.106	-0.422	0.675	-22.148	14.470
PopDensity	0.0067	0.004	1.544	0.129	-0.002	0.015
%NonWhite	3.6842	0.892	4.130	0.000	1.891	5.478
%WC	-2.1983	1.273	-1.727	0.091	-4.758	0.362
pop/house	32.9814	32.067	1.029	0.309	-31.493	97.456
income	-0.0009	0.001	-0.669	0.506	-0.004	0.002
S02Pot	0.2160	0.116	1.863	0.069	-0.017	0.449
log_NOxPot	4.6740	7.809	0.599	0.552	-11.027	20.375

Recursive Feature Elimination : 후진 제거방법과 동일

```
y=df_clean['Mortality']
X=df_clean.iloc[:,1:15]
from sklearn.feature_selection import RFE
from sklearn.svm import SVR
estimator = SVR(kernel="linear")
selector = RFE(estimator, 7, step=1) #최적 7개 변수만 선택
selector = selector.fit(X, y)
print(X.columns,'\n', selector.support_,'\n',selector.ranking_)
```

```
Index(['JanTemp', 'JulyTemp', 'RelHum', 'Rain', 'Education', 'PopDensity',
       '%NonWhite', '%WC', 'pop/house', 'income', 'S02Pot', 'NOx', 'log_HCPot',
       'log_NOxPot'],
      dtype='object')
 [ True  True False  True False False  True  True False False False False
  True  True]
 [1 1 4 1 6 8 1 1 2 7 3 5 1 1]
```

```
import numpy as np
var_select=pd.DataFrame(np.transpose([np.array(X.columns),selecto
r.ranking_]))
var_select.columns=['var_name','rank']
var_select.loc[var_select['rank']==1,"var_name"]
```

```
0        JanTemp
1       JulyTemp
3           Rain
6      %NonWhite
7            %WC
12     log_HCPot
13    log_NOxPot
```

Univariate

	Specs	Score	pvalues
6	%NonWhite	40.944	0.000
4	Education	19.835	0.000
3	Rain	13.161	0.001
10	S02Pot	12.146	0.001
8	pop/house	8.929	0.004
1	JulyTemp	6.586	0.013
7	%WC	5.208	0.026

많이 상이

(SSR Best Model)

```
['const',
 'JanTemp',
 'Rain',
 'PopDensity',
 '%NonWhite',
 '%WC',
 'log_HCPot',
 'log_NOxPot']
```

PopDensity(인구밀도) <-> 7월 기온만 상이함

```
import numpy as np
var_select=pd.DataFrame(np.transpose([np.array(X.columns),selecto
r.ranking_]))
var_select.columns=['var_name','rank']
var_select.loc[var_select['rank']==1,"var_name"]
```

```
0          JanTemp
1        JulyTemp
3            Rain
6       %NonWhite
7             %WC
12      log_HCPot
13     log_NOxPot
```

```
import statsmodels.api as sm
y=df_clean['Mortality']
X=sm.add_constant(df_clean[var_select.loc[var_select['rank']==1,"var_name"]])
fit=sm.OLS(y,X).fit()
fit.summary()
```

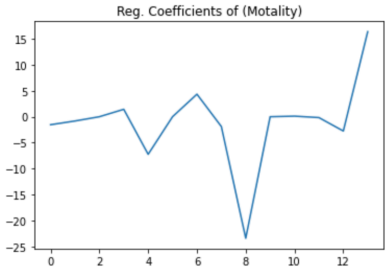
	coef	std err	t	P>|t|	[0.025	0.975]
const	1020.2165	130.341	7.827	0.000	758.545	1281.888
JanTemp	-1.8706	0.548	-3.416	0.001	-2.970	-0.771
JulyTemp	-1.0306	1.695	-0.608	0.546	-4.433	2.371
Rain	1.9783	0.496	3.987	0.000	0.982	2.974
%NonWhite	4.3129	0.803	5.370	0.000	2.701	5.925
%WC	-2.0109	0.938	-2.145	0.037	-3.893	-0.129
log_HCPot	-21.5781	14.598	-1.478	0.146	-50.886	7.729
log_NOxPot	37.2523	11.963	3.114	0.003	13.236	61.269

결정계수 73.8%

LASSO

```
from sklearn import linear_model
import matplotlib.pyplot as plt
lassoReg=linear_model.Lasso(alpha=0.1, fit_intercept=True,
normalize=True)
y=df_clean['Mortality']
X=df_clean.iloc[:,1:15]
fit=lassoReg.fit(X,y)
plt.title('Reg. Coefficients of (Motality)')
plt.plot(fit.coef_); plt.show()
```

회귀계수의 크기가 큰(단위를 표준화 하였으므로 회귀계수 크기가 크다는 것은 종속변수에 대한 영향도가 크다는 것을 의미한다. 가구원수 > LOG_NOxPot > Education > %NonWhite > log_HCPot 순으로 목표변수 Mortality에 영향을 미친다.



	Var	Coeff
0	JanTemp	-1.534
1	JulyTemp	-0.816
3	Rain	1.429
4	Education	-7.229
5	PopDensity	0.004
6	%NonWhite	4.337
7	%WC	-1.887
8	pop/house	-23.376
9	income	-0.000
10	S02Pot	0.110
11	NOx	-0.165
12	log_HCPot	-2.752
13	log_NOxPot	16.359

```
0        JanTemp
4      Education
6      %NonWhite
7            %WC
8      pop/house
12     log_HCPot
13    log_NOxPot
```

```
var_nm=pd.Series(X.columns,name='Var')
coeff=pd.Series(fit.coef_,name='Coeff')
LASSO=pd.concat([var_nm,coeff],axis=1)
LASSO[abs(LASSO['Coeff'])>0] #모든 예측변수 출력
```

```
LASSO_res=LASSO[abs(LASSO['Coeff'])>1.5]['Var'] #일부 선택 (7개)
import statsmodels.api as sm
y=df_clean['Mortality']
X=sm.add_constant(df_clean[LASSO_res])
fit=sm.OLS(y,X).fit()
fit.summary()
```

최종 임시 모형

다중 공선성 진단 및 최종 회귀진단 전에 잠정적으로 유의한 예측변수들을 선택할 필요가 있다. 유의한 혹은 주요한 변수를 선택하는 방법은 기준은 다음과 같다.

- 예측변수의 목표변수의 변동을 가장 잘 설명하는가? <=> 결정계수 (SSR Best Model)
- 예측변수는 유의한가? 개별 유의성 (상관계수, Univariate Selection), 공동 유의성(후진제거), REF 방법
- 영향력 높은 변수 선택

잠정 최종모형 선택

Univaiate selection $\alpha = 0.1$	SSR Best (7개 선택)	후진제거 $\alpha = 0.2$	RFE 방법 (7개 선택)
1 JulyTemp 3 Rain 4 Education 5 PopDensity 6 %NonWhite 7 %WC 8 pop/house 9 income 10 S02Pot 13 log_NOxPot	['const', 'JanTemp', 'Rain', 'PopDensity', '%NonWhite', '%WC', 'log_HCPot', 'log_NOxPot']	['JanTemp', 'Rain', 'Education', '%NonWhite', '%WC', 'pop/house', 'NOx', 'log_NOxPot']	JanTemp JulyTemp Rain %NonWhite %WC log_HCPot log_NOxPot

2 LASSO_res

0	JanTemp
4	Education
6	%NonWhite
7	%WC
8	pop/house
12	log_HCPot
13	log_NOxPot

	Mortality	JanTemp	JulyTemp
Mortality	1.000	-0.016	0.322
JanTemp	-0.016	1.000	0.322
JulyTemp	0.322	0.322	1.000
RelHum	-0.101	0.086	-0.441
Rain	0.433	0.059	0.472
Education	-0.508	0.108	-0.269
PopDensity	0.252	-0.076	-0.009
%NonWhite	0.647	0.459	0.602
%WC	-0.289	0.208	-0.013
pop/house	0.368	-0.325	0.257
income	-0.283	0.198	-0.191
S02Pot	0.419	-0.094	-0.071
NOx	-0.085	0.334	-0.334
log_HCPot	0.125	0.227	-0.413
log_NOxPot	0.280	0.181	-0.303

July 기온은 목표변수와 개별 유의성(상관계수=0.322) 측면에서는 JanTemp(상관계수=-0.016)보다 크지만 다른 예측변수와 함께 있다면 목표변수의 설명력(다른 예측변수들과 목표변수 변동 살명부분이 겹침)은 줄어든다. 상관계수 행렬에서 JulyTemp 열과 JanTemp 열의 타 예측변수(특히 유의한) 상관계수 값을 보면 알 수 있다.

최종적으로 후진제거(유의수준 0.2에서 유의) 방법의 8개 + RFE 2개 (Log_HCPot, JanTemp) - 겹치지 않는 = 10개 변수를 최종으로 하였음

다중공선성 Multicollinearity

개념

다중 회귀모형($y_i = a + \sum_k^p b_k X_{ki} + e_i$)은 예측변수와 목표변수간의 관계에 대한 분리하여(isolate) 유의성 검정(F-검정)과 각 예측변수의 유의성(t-검정)에 가장 큰 관심이 있다.

- 서로 다른 예측변수의 상대적 중요 정도는 무엇인가?(표준화 회귀계수)
- 목표변수에 대한 예측변수의 설명력의 크기는 얼마인가? (OLS 추정 회귀계수) - 예측변수(X_k) 한 단위 증감 시 목표변수는 b_k만큼 변동한다.
- 이런 해석은 다른 예측변수들이 고정되었다는 가정이 필요하다.

즉, 예측변수가 서로 독립(상관계수 0)이면 예측변수의 회귀계수(b_k, 상대적 효과, 설명력, marginal effect)에 의해 위의 질문에 답이 가능하다.

그러나 현실에서 예측변수가 완벽하게 독립인 경우는 없다. 예측변수들 간 상관계수가 낮으면(유의하지 않으면) 여전히 회귀계수 해석이 적절하다.

그러나 만약 예측변수간에 상관관계가 높으면(유의하면) OLS 회귀계수 추정과 검정(추정 분산이 변함)이 쓸모가 없게 되는데 이를 다중공선성(Multicollinearity) 문제라 한다.

- 예측변수 간의 상관관계가 높다는 것은 (변수의 유사성이 높다는 것이고) 목표변수를 설명하는 부분이 겹친다는 것이다.
- 예측변수가 목표변수를 잘 설명한다는 것은 목표변수 값의 변동 방향을 매우 잘 예측(변동 방향의 일치, 물론 회귀계수가 음이면 역방향)한다는 것을 의미한다.
- 그러므로 상관관계가 높은 예측변수들 간에는 목표변수 변동 설명 부분이 겹친다.

구조적 structural 다중공선성 : 설정된 회귀모형이 예측변수의 다항식 항을 포함한 경우 $x^2, x^3, \ldots$

데이터 다중공선성 : 예측변수 간의 높은 상관 관계로 인한 발생

다중공선성 문제는 무엇인가?

회귀모형 $\underline{y} = X\underline{b} + \underline{e}$의 OLS 추정치는 $\hat{\underline{b}} = (X'X)^{-1}X'\underline{y}$이다. 예측변수($X_k$) 간의 상관관계가 높으면(예: 두 설명변수의 상관관계가 높으면 (데이터 행렬의 한 열(변수)이 다른 열(변수와 상관계수가 큰 변수)로 표현되므로 $det|X'X| \approx 0$ 되므로 $X'X = \frac{adj(X'X)}{det|X'X|}$의 값이 매우 커진다.

- 이로 인하여, OLS 추정치 $\hat{\underline{b}} = (X'X)^{-1}X'\underline{y}$이므로 추정값이 불안해진다. - 다중공선성 문제를 일으키지 않는 예측변수의 회귀계수에 비해 변동이 커진다. - 회귀모형의 작은 변화에도 회귀계수의 추정치는 매우 민감하게 반응한다.
- 그리고 OLS 추정의 추정분산($s^2(\hat{\underline{b}}) = MSE(X'X)^{-1}$) 또한 매우 커져 회귀계수의 부호까지 바뀌는 문제가 발생한다.
- 회귀계수 추정 오차가 커지면 추정 정확도가 떨어지고 더 이상 회귀계수 유의성을 판단하는 유의확률(p-값)에 대한 신뢰성도 낮아진다. - 즉, 회귀계수 유의성 검정의 신뢰성 저하, -> 추정 회귀모형 신뢰성 낮아짐

과연 반드시 해결해야 하나?

다음의 경우가 연구목적이라면 굳이 해결할 필요는 없다.

예측(prediction)이 주목적인 경우 ***

다중공선성은 회귀계수 추정오차에 영향을 주지만 모형의 정확, 예측에는 영향을 주지 않는다. 오히려 예측력을 높이는 효과가 있음. 결정계수(Determination Coefficient)가 높아짐

결론적으로 빅데이터의 예측모형 개념에서는 다중공선성 문제를 진단할 필용 없음

그리고 모형 내 예측변수 회귀계수 부호가 (목표변수) 상관계수 부호와 동일하다면 진단하지 않아도 된다.

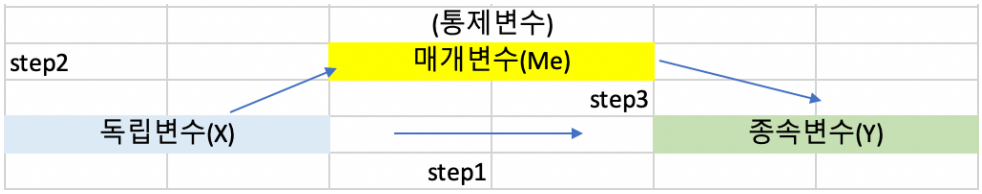
통제변수가 다중공선성 문제를 일으키는 경우

통제변수를 있는 모형의 경우 실험변수에만 다중공선성 문제가 없다면 해결할 필요 없음

The fact that some or all predictor variables are correlated among themselves does not, in general, inhibit **our ability to obtain a good fit** nor does it tend to affect **inferences about mean responses or predictions of new observations**. —Applied Linear Statistical Models, p289, 4th Edition.

Baron, R. M. and Kenny, D. A. (1986) "The Moderator(조절)-**Mediator(매개)** Variable"

매개효과 Meditator

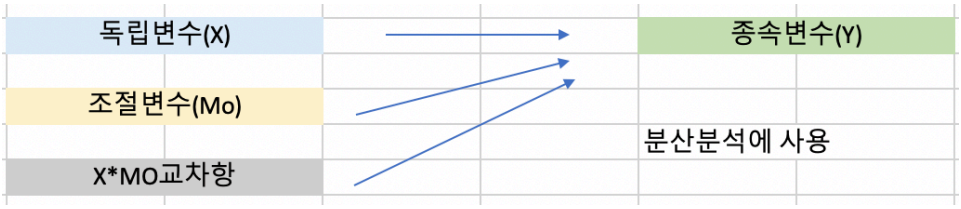


(순서1) $y = a + b_{11}X + e$: b_{11} 유의해야 한다.

(순서2) $y = a + b_{21}Me + e$: b_{21} 유의해야 한다.

(순서3) $y = a + b_{31}X + b_{32}Me + e$: b_{32} (매개효과) 유의하고 b_{31}은 b_{11}보다 절대값이 작아지거나 유의하지 않을 수 있음

조절효과 Moderate



$$y = a + b_1X + b_2MO + b_3X * MO + e$$

- b_3는 반드시 유의해야 한다.

맛보기

예제데이터 df_clean

	Mortality	JanTemp	JulyTemp	RelHum	Rain	Education	PopDensity	%NonWhite	%WC	pop/house	income	S02Pot	NOx	log_HCPot	log_NOxPot
Mortality	1.000	-0.016	0.322	-0.101	0.433	-0.508	0.252	0.647	-0.289	0.368	-0.283	0.419	-0.085	0.125	0.280
JanTemp	-0.016	1.000	0.322	0.086	0.059	0.108	-0.076	0.459	0.208	-0.325	0.198	-0.094	0.334	0.227	0.181
JulyTemp	0.322	0.322	1.000	-0.441	0.472	-0.269	-0.009	0.602	-0.013	0.257	-0.191	-0.071	-0.334	-0.413	-0.303
RelHum	-0.101	0.086	-0.441	1.000	-0.118	0.186	-0.149	-0.119	0.015	-0.144	0.128	-0.116	-0.053	0.185	0.097
Rain	0.433	0.059	0.472	-0.118	1.000	-0.473	0.084	0.303	-0.114	0.199	-0.362	-0.131	-0.460	-0.485	-0.389
Education	-0.508	0.108	-0.269	0.186	-0.473	1.000	-0.236	-0.209	0.486	-0.389	0.507	-0.229	0.229	0.176	0.032
PopDensity	0.252	-0.076	-0.009	-0.149	0.084	-0.236	1.000	-0.007	0.253	-0.167	-0.003	0.422	0.158	0.264	0.339
%NonWhite	0.647	0.459	0.602	-0.119	0.303	-0.209	-0.007	1.000	-0.057	0.353	-0.101	0.160	0.019	0.148	0.208
%WC	-0.289	0.208	-0.013	0.015	-0.114	0.486	0.253	-0.057	1.000	-0.347	0.366	-0.063	0.129	0.162	0.108
pop/house	0.368	-0.325	0.257	-0.144	0.199	-0.389	-0.167	0.353	-0.347	1.000	-0.295	-0.010	-0.449	-0.224	-0.123
income	-0.283	0.198	-0.191	0.128	-0.362	0.507	-0.003	-0.101	0.366	-0.295	1.000	0.068	0.312	0.292	0.245
S02Pot	0.419	-0.094	-0.071	-0.116	-0.131	-0.229	0.422	0.160	-0.063	-0.010	0.068	1.000	0.406	0.569	0.681
NOx	-0.085	0.334	-0.334	-0.053	-0.460	0.229	0.158	0.019	0.129	-0.449	0.312	0.406	1.000	0.738	0.705
log_HCPot	0.125	0.227	-0.413	0.185	-0.485	0.176	0.264	0.148	0.162	-0.224	0.292	0.569	0.738	1.000	0.939
log_NOxPot	0.280	0.181	-0.303	0.097	-0.389	0.032	0.339	0.208	0.108	-0.123	0.245	0.681	0.705	0.939	1.000

- 목표변수 사망율지수(Mortality)와 상관관계 가장 높은(가장 잘 설명하는) 비백인비율(%NonWhite)과 '7월기온'(0.58) 선택하였음
- 비백인비율과 상관관계가 매우 작으나 목표변수와는 상관관계 높은 변수 %WC(사무직 비율) 선택

비백인 비율 -> 사망률지수 단순 모형

```
import statsmodels.api as sm
y=df_clean['Mortality']
X=sm.add_constant(df_clean['%NonWhite'])
fit=sm.OLS(y,X).fit()
fit.summary()
```

비백인 비율이 1단위(1%) 증가하면 사망률지수 4.48증가함

	coef	std err	t	P>|t|	[0.025	0.975]
const	887.9018	10.412	85.274	0.000	867.051	908.752
%NonWhite	4.4855	0.701	6.399	0.000	3.082	5.889

결정계수=41.8%

(비백인 비율, 7월기온) -> 사망률지수 모형

```
import statsmodels.api as sm
y=df_clean['Mortality']
X=sm.add_constant(df_clean[['%NonWhite','JulyTemp']])
fit=sm.OLS(y,X).fit()
fit.summary()
```

- 7월 기온 유의하지도 않음 : 비백인비율 변수와 상관관계가 높아 사망률 지수를 설명하는 부분이 겹쳐 유의하지 않게 되었음
- 결정계수 증가도 매우 미미하고 비백인비율의 추청오차(std err)도 0.701->0.88으로 높아짐 - 다중공선성 문제로 인하여 추정이 왜곡됨
- 또한 비백인 비율 추정계수도 4.48-> 4.92 로 증가, 기울기(영향)이 변동이 심함
- 다중공성선 문제 발생 경고 창도 나타난다.

[2] The condition number is large, 2.63e+05. This might indicate that there are strong multicollinearity or other numerical problems.

	coef	std err	t	P>|t|	[0.025	0.975]
const	989.6236	122.238	8.096	0.000	744.752	1234.495
%NonWhite	4.9284	0.880	5.598	0.000	3.165	6.692
JulyTemp	-1.4378	1.721	-0.835	0.407	-4.886	2.011

결정계수=42.5%

비백인 비율, 사무직비율(상관관계 낮음) -> 사망률지수 모형

```
import statsmodels.api as sm
y=df_clean['Mortality']
X=sm.add_constant(df_clean[['%NonWhite','income']])
fit=sm.OLS(y,X).fit()
fit.summary()
```

- 목표변수, 사망률지수와 상관 관계 낮지만(설명력 부족) 비백인 비율이 설명하고 남은 부분에 대한 설명력은 충분하여 유의하다.
- 결정계수도 46.6%로 4.8% 증가하였고 비백인 비율의 추청오차(std err)도 0.701->0.668로 낮아짐
- 회귀계수(기울기)도 4.48->4.38 로 낮아졌음, 설명 영향도가 제대로 반영됨
- 다중공선성 경고 창도 없음

	coef	std err	t	P>|t|	[0.025	0.975]
const	1033.6913	56.358	18.342	0.000	920.794	1146.589
%NonWhite	4.3850	0.668	6.561	0.000	3.046	5.724
%WC	-3.1172	1.186	-2.628	0.011	-5.493	-0.741

결정계수=46.6%

진단도구

산점도_상관계수

산점도 행렬이나 상관계수를 계산하여 상관 관계가 높은 설명변수들을 판단하고 다중공선성 문제가 일어날 것이라는 예상을 한다. 회귀분석의 시작은 산점도 행렬과 상관계수 행렬이다. 상관계수 행렬은 종속변수에 영향을 주는 설명변수를 사전에 판단하고 설명변수 간 높은 상관관계로 인해 발생하는 다중공선성을 사전 진단할 수 있음

간단히 진단할 수 있으나 다중공선성 유의성을 검정할 수 없고 두 예측변수만의 쌍체(pairwise) 진단만 가능하다.

분산팽창지수 Variance Inflation Index $VIF_k = \frac{1}{1 - R_k^2}$

k-번째 예측변수를 목표변수(Y)로 하고 나머지 예측변수를 설명변수로 하여 회귀분석 한 후 결정계수를 구한다. 그 결정계수 값이 R_k^2이다.

- 결정계수 R_k^2 값이 크다는 것은 X_k가 다른 예측변수들에 의해 충분히 설명되고 있다는 것이고 이는 바로 다중공선성 문제가 발생할 수 있다는 증거이다.
- 다수의 예측변수들에 의해 설명되는 정도를 나타내지만 상관계수와 같이 두 개의(쌍체) 예측변수에 의한 진단에는 적절하지 못함

일반적으로 VIF 진단 기준값은 상이함

- 10 이상: (Hair, J. F. Jr., Anderson, R. E., Tatham, R. L. & Black, W. C. (1995). Multivariate Data Analysis (3rd ed). New York: Macmillan.)
- 5이상 : Ringle, Christian M., Wende, Sven, & Becker, Jan-Michael. (2015).
- 빅데이터에서는 3, 2.5를 사용하기도 함.

상태지수 Condition Index $CI = \sqrt{\frac{\lambda_{max}}{\lambda_k}}$

- 예측변수의 공분산(상관계수)행렬을 이용하여 고유값 eigen value(λ_k) 를 구한다.
- 이에 대응하는 Norm 1인 고유벡터가 주성분 부하 loading 이다.
- 고유값은 원변수(설명변수)의 선 형결합에 의해 만들어진 주성분 변수의 원변수 변동에 대한 설명력이다. 그러므로 고유값가 크다는 것(상태지수 값이 큰 값) 주성분의 원 변수 변동에 설명력이 크다는 것을 의미한다.
- 고유벡터 내의 부하크기가 상대적으로 큰 변수들 간에는 상관관계가 높다는 것을 의미한다.
- 상태지수가 10이상(일반적으로 30이상)인 부하벡터에서 부하계수가 상대적으로 큰 변수들에 의해 다중공선성 문제가 발생한다고 진단할 수 있다.

권장 진단 방법

- VIF 통계량 3 이상, 최종적으로 유의하다고 판단된 설명변수의 피어슨 상관계수 활용
- 만약 최종 모형에서 추정된 해당 설명변수 회귀계수 부호와 종속변수와 해당 설명변수의 피어슨 상관계수 부호가 동일하다면, 다중공선성 문제를 해결하지 않아도 됨.

해결방안

구조적 다중공선성 문제 해결 - 예측변수 centering (중심화) - 다항식 모형(예측변수의 제곱항, 세제곱항)에 적합

표준화는 데이터의 평균과 표준편차를 활용하여 단위를 없앤 개념임 : $z_i = \frac{x_i - \bar{x}_i}{s(x_i)}$

중심화는 데이터의 평균을 0으로 이동한 개념임, 산포는 동일하게 유지됨 : $c_i = x_i - \bar{x}_i$

문제변수 제거

- 다중공선성 문제가 되는 설명변수를 제외한다.
- 여러개가 존재하는 경우 목표변수와 상관관계가 낮은 예측변수를 제외한다.

주성분 분석 : http://203.247.53.31/Stat_Notes/adv_stat/MDA/MDA_PCA.pdf

예측변수를 (서로 독립인)주성분 변수로 변환하여 사용한다. $PC_{n\times p} = X_{n\times p}L_{p\times p}$ (L=부하행렬)

주성분 분석(Principal Component Analysis)은 다음 원칙에 의해 원변수의 선형 결합인 주성분을 얻는다.

- 주성분 변수 간에는 서로 상관 관계가 전혀 없다. (독립이다)
- 첫번째 주성분은 변수들의 변동을(분산, 이를 변수가 가진 정보로 표현) 가장 많이 설명하고 계속 구해지는 2, 3, …번째 주성분은 자료의 나머지 정보들을 설명하고 크기는 점점 줄어 든다.
- 주성분 변수(Principal Component)는 원변수(회귀모형의 예측변수 X's)의 선형결합이며 서로는 독립이다.
- 주성분 변수는 서로 독립이므로 주성분 변수를 설명변수로 사용한다면 다중공선성 문제가 발생하지 않는다.

능형 회귀분석 Ridge Regression

다중공선성은 회귀계수의 추정오차를 증가시키므로 불편성(OLS는 불편 추정량이다)을 포기하는 대신 MSE(Mean Square of Error 평균오차자승합)를 최소화 하는 편기(biased) 추정량을 구하는 추정 방법을 사용함으로써 다중공성선 문제를 해결하는데 이를 능형 회귀분석((Ridge Regression)이라 한다.

- 편기 추정량 $\hat{\underline{b}}_* = (X'X + cI)X'\underline{y}$의 $MSE(\hat{\underline{b}}_*) = E(\hat{\underline{b}}_* - \underline{b})^2$를 최소화 하는 $0 < c < 1$를 구하면 이를 능형 추정량이라 한다. $c = 0$이면 OLS 추정값이다.
- 상수 c와 VIF 산점도를 이용하여 각 예측변수의 추정 회귀계수 값이 안정화 되는 c값을 최적값으로 설정한다. 다소 주관적이다.

파이썬 코드 : 예제

데이터

```
y=df_clean['Mortality']
X=pd.concat([df_clean[selected_features_BE],df_clean[['JulyTemp',
'log_HCPot']]],axis=1)
```

- 유의수준 0.2 기준 후진제거 방법 변수선택 결과 selected_features_BE 8개 예측변수와 추가 2개 예측변수(7월기온, 로그_HCPot) 총 10개를 사용하였다.
- 예상대로 다중공선성 문제 발생 경고가 있음

[2] The condition number is large, 5.48e+03. This might indicate that there are strong multicollinearity or other numerical problems.

	coef	std err	t	P>|t|	[0.025	0.975]
const	1437.1726	219.686	6.542	0.000	995.465	1878.880
JanTemp	-2.0701	0.623	-3.320	0.002	-3.324	-0.817
Rain	1.3566	0.551	2.463	0.017	0.249	2.464
Education	-11.7192	7.335	-1.598	0.117	-26.467	3.029
%NonWhite	5.1329	0.910	5.640	0.000	3.303	6.963
%WC	-1.7022	1.060	-1.606	0.115	-3.833	0.429
pop/house	-71.4977	36.367	-1.966	0.055	-144.618	1.623
NOx	-0.2454	0.164	-1.493	0.142	-0.576	0.085
log_NOxPot	38.4468	12.409	3.098	0.003	13.498	63.396
JulyTemp	-1.6720	1.662	-1.006	0.320	-5.014	1.670
log_HCPot	-21.2653	14.465	-1.470	0.148	-50.349	7.819

결정계수=76.9%

VIF 진단

```
from statsmodels.stats.outliers_influence import variance_inflation_factor
def calculate_vif_(X, thresh=10):
    variables = list(range(X.shape[1]))
    dropped = True
    while dropped:
        dropped = False
        vif = [variance_inflation_factor(X.iloc[:, variables].values, ix)
               for ix in range(X.iloc[:, variables].shape[1])]
        maxloc = vif.index(max(vif))
        if max(vif) > thresh:
            print('dropping \'' + X.iloc[:, variables].columns[maxloc] +
                  '\' at index: ' + str(maxloc))
            del variables[maxloc]
            dropped = True
    print('Remaining variables:')
    print(X.columns[variables])
    return X.iloc[:, variables]

X0=calculate_vif_(X)
```

- 10을 기준으로 4개 변수만 선택되었음

```
dropping 'JulyTemp' at index: 8
dropping 'Education' at index: 2
dropping 'pop/house' at index: 4
dropping 'log_HCPot' at index: 6
dropping '%WC' at index: 3
dropping 'JanTemp' at index: 0
Remaining variables:
Index(['Rain', '%NonWhite', 'NOx', 'log_NOxPot'], dtype='object')
```

```
import statsmodels.api as sm
y=df_clean['Mortality']
X=sm.add_constant(X0)
fit=sm.OLS(y,X).fit()
fit.summary()
```

- 다중공선성 문제가 해결되어 더 이상 경고창이 출력되지 않음 : 모든 예측변수 유의함

	coef	std err	t	P>|t|	[0.025	0.975]
const	768.2807	26.569	28.917	0.000	715.014	821.548
Rain	2.0495	0.558	3.673	0.001	0.931	3.168
%NonWhite	2.9188	0.659	4.426	0.000	1.597	4.241
NOx	-0.4189	0.165	-2.533	0.014	-0.750	-0.087
log_NOxPot	29.4609	6.625	4.447	0.000	16.178	42.744

결정계수 = 62%

능형회귀 Ridge Regression

```
from sklearn.linear_model import Ridge
y=df_clean['Mortality']
X=pd.concat([df_clean[selected_features_BE],df_clean[['JulyTemp',
'log_HCPot']]],axis=1)
clf=Ridge(alpha=0.3) #디폴트 = 1, c와 동일
clf.fit(X, y)
```

• 다중공선성 문제가 해결되어 더 이상 경고창이 출력되지 않음 : 모든 예측변수 유의함

```
import numpy as np
np.set_printoptions(precision=4,suppress = True)
print(clf.intercept_,clf.coef_)   #절편, 회귀계수 추정값
```

```
1342.3689172915895 [ -1.9694   1.4436 -11.1503   4.8775  -1.6304 -51.571   -0.2116  35.395
  -1.4915 -17.7813]
```

```
1  X.columns
```

```
Index(['JanTemp', 'Rain', 'Education', '%NonWhite', '%WC', 'pop/house', 'NOx',
       'log_NOxPot', 'JulyTemp', 'log_HCPot'],
```

```
1  clf.score(X,y) #결정계수
```

```
0.7675370846431416
```

```
1  clf.predict(X) #적합치
```

```
array([ 947.9143,  932.4409,  933.5402,  974.9881, 1037.8143, 1092.2156,
        932.1136,  927.8502,  959.1228,  930.2303,  996.482 ,  998.8808,
```

주성분분석

```
from sklearn.preprocessing import StandardScaler
y=df_clean['Mortality']
X=pd.concat([df_clean[selected_features_BE],df_clean[['JulyTemp',
'log_HCPot']]],axis=1)
X_s=StandardScaler().fit_transform(X) #scale standardization
from sklearn.decomposition import PCA
pca=PCA(n_components=5) #80% rule 0.8
df_pca=pd.DataFrame(pca.fit_transform(X_s)) #PC variables
df_pca.columns=['PC1','PC2','PC3','PC4','PC5']
df_pca.set_index(df_clean.index,inplace=True)
df_pca.head(3)
```

- 주성분 변수 5개만 선택하였음 : n_components=5 대신 0.8을 사용하면 원변수 변동 80%까지 누적 설명하는 주성분 개수가 자동 선택된다.

	0	1	2	3	4
JanTemp	0.123	0.452	0.390	0.262	-0.299
Rain	-0.367	0.198	0.106	0.323	0.535
Education	0.265	-0.217	0.421	-0.423	-0.314
%NonWhite	-0.096	0.609	0.006	-0.280	-0.099
%WC	0.186	-0.022	0.522	-0.376	0.624
pop/house	-0.287	0.120	-0.412	-0.587	-0.039
NOx	0.441	0.176	-0.066	0.244	-0.098
log_NOxPot	0.401	0.285	-0.303	-0.078	0.260
JulyTemp	-0.308	0.400	0.257	-0.109	-0.140
log_HCPot	0.450	0.228	-0.235	-0.082	0.169

city_name	PC1	PC2	PC3	PC4	PC5
Akron	0.257	-0.782	-1.004	-0.390	-0.292
Albany-Schenectady-Troy	0.146	-1.631	0.294	-0.067	0.780
Allentown	-1.408	-1.008	-1.014	1.635	-0.129

주성분 변수 4만 제외하고 모두 유의함 – loadings(부하값)은 주성분 이름 부여에 사용함 : 주성분 1은 기후-환경 성분, 2는 비백인 거주 지역 성분

```
loadings=pd.DataFrame(pca.components_.T)
loadings.set_index(X.columns)
```

```
import statsmodels.api as sm
y=df_clean['Mortality']
X=sm.add_constant(df_pca)
fit=sm.OLS(y,X).fit()
fit.summary()
```

	coef	std err	t	P>|t|	[0.025	0.975]
const	941.1731	5.259	178.962	0.000	930.625	951.721
PC1	-8.3836	2.779	-3.017	0.004	-13.957	-2.810
PC2	24.7275	3.595	6.878	0.000	17.517	31.938
PC3	-18.8361	4.108	-4.585	0.000	-27.075	-10.597
PC4	-1.7539	5.583	-0.314	0.755	-12.952	9.444
PC5	18.2596	6.474	2.820	0.007	5.274	31.245

회귀진단 및 최종 예측모형

개념

회귀분석은 이론이나 경험으로 얻어진 사실을 기반한 객관 타당성에 의해 설정된 회귀모형을 데이터를 통하여 유의성을 판단하는 분석방법이다.

회귀분석 시작 시 오차항에 대한 "가정"과 함수의 선형성을 가정하였음

- 이 가정 하에 회귀계수에 대한 검정통계량의 샘플링분포도 구하였고 회귀계수에 대한 추론과 분석 결과를 해석하였다.
- 만약 이 가정이 성립되지 않으면 분석결과를 신뢰할 수 없게 된다.
- 하여, 회귀모형의 가정을 만족하는지 분석할 필요가 있음 - 이를 회귀진단(잔차분석)이라 함

선형성 가정은 회귀모형 유의성 검정 결과와 동일하므로 잔차분석이라 함은 오차항에 대한 3가지 가정(정규성, 등분산성, 독립성)을 진단하고 문제를 해결하는 방법이다.

- 독립성 진단은 시계열 자료에서만 하게 됨
- 추가적으로 회귀모형 추정 결과에 영향을 주는 (이상치, 영향치) 진단까지 잔차분석에 포함됨

최종 회귀모형 확정 전 회귀추론의 기본가정인 오차가정(정규성, 이분산성, 독립성), 회귀모형의 선형성, 그리고 회귀모형의 결정계수(모형 설명력)에 영향을 미치는 이상치(영향치) 진단이 필요하다.

잔차 residual 및 진단도구

회귀모형 및 추정

회귀모형 $y_i = \alpha + \sum_{k=1}^{p} \beta_k x_{ki} + e_i, e_i \sim N(0,\sigma^2)$ [행렬 표현] $\underline{y} = X\underline{\beta} + \underline{e}, \underline{e} \sim MN(\underline{0}, \sigma^2 I)$

- 추정 (OLS) : $\hat{\underline{\beta}} = (X'X)^{-1}X'\underline{y}$
- 적합치 : $\hat{\underline{y}} = H\underline{y}, H = X(X'X)^{-1}X'$는 멱등행렬

멱등 idempotent 행렬 $HH = H$가 성립하면 H는 멱등행렬이다.

잔차 정의

- $\underline{r} = (I - H)\underline{y}, r_i = y_i - \hat{y}_i$: 잔차는 종속변수 관측치와 모형 적합치의 차이
- 회귀모형 오차항(e_i)의 MVUE : $\hat{e}_i = r_i$

잔차성질

- 잔차의 평균은 $E(r_i) = 0$이고 분산은 $V(r_i) = \sigma^2$이다.
- 잔차는 서로 독립인가? 사실 오차는 독립을 가정하나 잔차는 독립이 아니다. 회귀계수 OLS추정에는 (x_i, y_i)모든 관측치가 포함되어 있기 때문이다.
- $\sum x_i r_i = 0$: 예측변수와 잔차의 곱의 합은 0이다 - 설명변수와 잔차는 독립이다. 예측변수에 의해 설명되고 남은 부분(잔차)은 서로 독립이다.
- $\sum \hat{y}_i r_i = 0$: 적합치와 잔차의 곱의 합은 0이다. 적합치는 예측변수에 의해 설명된 부분과 설명되지 않은 잔차 부분은 서로 독립이다.

잔차분석이란

오차의 추정치인 잔차를 이용하여 다음 작업 과정을 거치는 것이다.

- 설명변수와 종속변수의 함수 관계는 선형인가? <=> 회귀계수 유의성 검정과 동일 <=> 오차항의 패턴 없이 무작위 형태
- 오차의 분산은 설명 변수의 값에 따른 변화는 없는가? (등분산성)
- 오차항은 서로 독립인가? (독립성) : 시계열 데이터 회귀분석에서만 체크한다.
- 오차항은 정규분포를 따르는가? (정규성)
- 이상치나 영향치가 존재하는가? 등분산 가정에 의해 잔차의 값이(표준화 잔차 ± 2, 표준정규분포를 가정하므로) 큰 경우
- 고려된 설명 변수 이외 다른 주요한 설명 변수가 존재하지는 않는가? 잔차가 일정한 패턴을 갖는다. *)현실성 결여

잔차 종류

표준화 standardized 잔차

$$z_i = \frac{r_i - \bar{r}}{s(r_i) = \sqrt{MSE}}, MSE = \hat{\sigma}^2$$

- 표준화 잔차는 추정 회귀식으로부터 관측치가 얼마나 떨어져 있나를 나타내는 것으로 ± 2 (표준정규분포의 경우 ± 1.96 구간 안에는 95% 관측치가 있음) 보다 크면 이상치일 가능성이 높음

스튜던트 studentized 잔차

$$st_i = \frac{r_i}{\sqrt{MSE/(1 - h_{ii})}}, h_{ii} = \underline{x}_i'(X'X)^{-1}\underline{x}_i$$

- 잔차를 t-분포를 따르는 통계량으로 만든 것으로 ± 2이면 이상치(혹은 영향치)로 판단

• h_{ii} : $H = X(X'X)^{-1}X'$ 행렬의 대각 원소로 leverage 레버리지(지렛대)로 정의되며 영향치 판단에 사용

표준화/스튜던트 제외 standardized & deleted 잔차

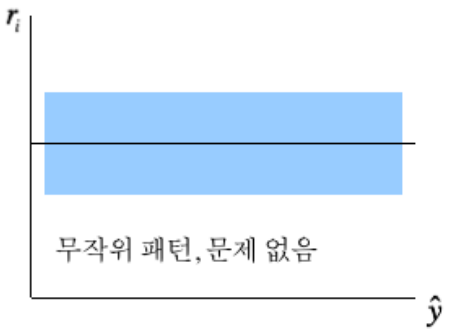
$$z_{(i)} = \frac{r_{(i)} - \bar{r}}{\sqrt{MSE_{(i)}}}, st_{(i)} = \frac{r_(i)}{\sqrt{MSE_{(i)}/(1-h_{ii})}}$$

• i – 번째 관측치를 제외하고 회귀모형을 추정한 후 얻은 적합치를 사용하여 얻은 잔차로 표준화/스튜던트 잔차에 비해 더 정확한 개념의 잔차이지만 현실에서는 자주 사용하지 않음

진단도구

① 잔차(Y-축)와 설명변수 산점도 Scatter plot of residual against independent variable

설명변수와 잔차의 산점도는 함수 형태를 가져서는 안된다. 왜냐하면 오차와 설명변수가 종속변수를 설명하지 못하는 부분에 해당되기 때문이다. 단순회귀에서는 설명변수가 하나이므로 설명변수와 종속변수가 동일하므로(동일한 형태) 단순회귀에서는 이 산점도를 사용하지 않는다. 다중회귀에서는 오차의 등분산성 진단을 위하여 가끔 사용하기도 하지만 유효성이 의심되어 이 산점도는 다중회귀 잔차분석에서도 거의 사용하지 않는다.



② 잔차와 종속변수 추정치 산점도 (*) 단순회귀분석에서는 이 도구만 주로 사용

잔차를 Y축, 종속변수의 예측치를 X-축으로 하여 산점도를 그린다. 잔차는 추정된 회귀 모형이 종속 변수의 변동을 설명하지 못하는 부분에 해당하므로 산점도에 일정한 패턴이 있으면 안되고 평균 0을 중심으로 무작위(random) 하게 흩어져 있어야 한다. 그리고 잔차가 크다는 것은 그 관측치가 이상치 가능성 있다. 또한 이 산점도에 의해 등분산성, 선형성도 진단한다.

③ 잔차(Y-축)와 시간(time, X-축)의 시간도표(time plot) : 시계열 데이터에만 국한된다.

④ 변수(관측치를 나누는 분류 변수) 수준별 잔차 그래프

관측치를 분류할만한 변수가 있을 때에만(예: 성별) 가능하다. 즉 설명변수 이외에 분류형 변수(이를 지시변수라 함)가 있을 때만 그린다.

⑤ 잔차에 대한 일변량 분석 : 전처리 작업에서 종속변수 정규변환 하였음

Stem and Leaf plot과 Shapiro-Wilks W-통계량(정규성), Box-Whisker plot(정규성, 이상치 혹은 영향치) 이상치나 영향치는 ②의 그래프에서 진단되므로 정규성만 검정하면 된다.

*) 잔차의 정규성 검정에 대한 찬반 : (a) 잔차는 종속변수의 평균의 개념으로 계산되어 대표본 large sample 이론에 의해 데이터가 충분히 크면 (>30) 정규분포에 근사하게 되어 표본의 크기가 충분히 큰 경우 정규성 진단은 하지 않아도 됨

잔차 구하기

변수선택 => 다중공선성 진단 통과한 4개 예측변수 활용

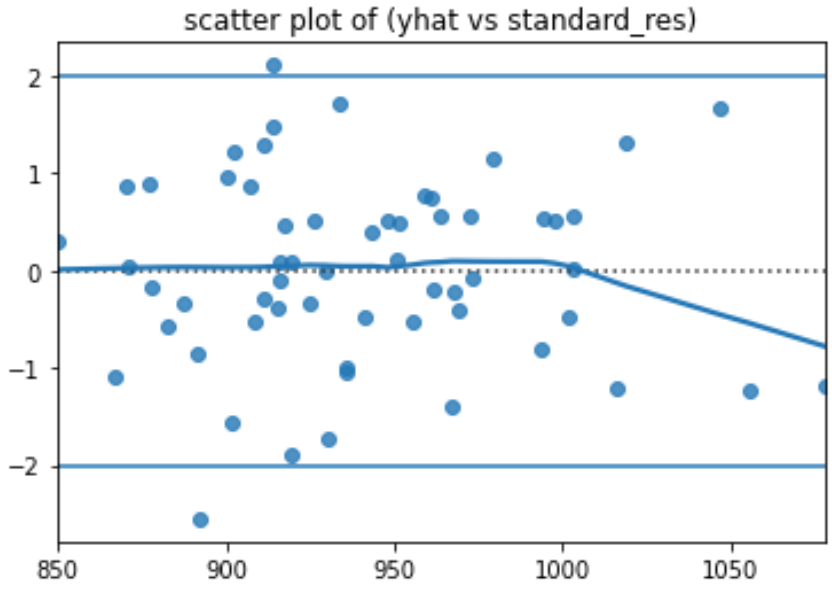
```
1  X0.columns
```

```
Index(['Rain', '%NonWhite', 'NOx', 'log_NOxPot'], dtype='object')
```

```
import statsmodels.api as sm
y=df_clean['Mortality']
X=sm.add_constant(X0)
fit=sm.OLS(y,X).fit()
```

- 회귀분석 추정 결과는 모두 fit 오브젝트에 저장되어 있다.
- 적합치는 *.fittedvalues, 표준화 잔차 *.resid_pearson

```
import seaborn as sns
import matplotlib.pyplot as plt
sns.residplot(fit.fittedvalues,fit.resid_pearson,lowess=True)
plt.title('scatter plot of (yhat vs standard_res)')
plt.axhline(2)
plt.axhline(-2)
plt.show()
```



이상치 2개 존제

선형성 Linearity

종속변수와 설명변수 간의 함수관계는 선형(직선)이다. 선형 회귀모형의 기본 가정이다.

회귀모형 전처리 작업(종속변수와 해당 설명변수의 산점도)에서 사전 진단 후 해결하였고 회귀모형(선형) 유의성 검정을 거쳐 선형성 검정은 필요 없으나 변수들간의 관계가 복잡하게 얽힌 경우 통합적 선형성 검정 분석방법이 필요하다.

Harvey A. and Collier P. (1977); Testing for Functional Misspecification in Regression Analysis, Journal of Econometrics 6, 103--119. Johnston, J. (1984); Econometric Methods, Third Edition, McGraw Hill Inc.

```
import statsmodels.stats.api as sms
sms.linear_harvey_collier(fit)
```

- 싱귤러 행렬이라는 경고문과 함께 출력되지 않음

```
/usr/local/lib/python3.6/dist-packages/numpy/linalg/linalg.py in _raise_linalgerror_singular(err, flag)
     95
     96 def _raise_linalgerror_singular(err, flag):
---> 97     raise LinAlgError("Singular matrix")
```

정규성 가정

오차의 가정 중 하나로 이를 기반으로 모형 전체의 유의성 검정인 분산분석 F-검정, 예측변수의 유의성 검정인 t-검정의 기본 가정이다.

오차는 잔차에 의해 추정되므로 잔차의 정규성 검정과 동일하다.

- 귀무가설 : 데이터는 정규분포를 따른다. / 대립가설 : 정규분포를 따르지 않는다.

```
from scipy.stats import shapiro
shapiro(fit.resid)[0:2]
```

- 앞의 값은 검정통계량이며 2번째 값은 유의확률(0.9737…)로 유의확률이 매우 크므로 잔차는 정규분포(귀무가설 만족)를 따른다.

```
(0.9924234747886658, 0.9737654328346252)
```

```
import statsmodels.stats.api as sms
sms.omni_normtest(fit.resid)[0:2]
```

- Omni Test : 앞의 값은 검정통계량이며 2번째 값은 유의확률(0.7667…)로 유의확률이 매우 크므로 잔차는 정규분포(귀무가설 만족)를 따른다.
- sm.OLS 출력결과에 자동출력된다.

```
(0.5312432426752998, 0.7667291865854793)
```

log_NOxPot	29.4609	6.625	4.447	0.000	16.178	42.744

Omnibus:	0.531	**Durbin-Watson:**	1.696
Prob(Omnibus):	0.767	**Jarque-Bera (JB):**	0.537

```
import statsmodels.stats.api as sms
sms.jarque_bera(fit.resid)[0:2]
```

- Jarque-Bera Test : 앞의 값은 검정통계량이며 2번째 값은 유의확률(0.537…)로 유의확률이 매우 크므로 잔차는 정규분포(귀무가설 만족)를 따른다.
- sm.OLS 출력결과에 자동출력된다.

```
(0.5374476144328763, 0.7643543356490612)
```

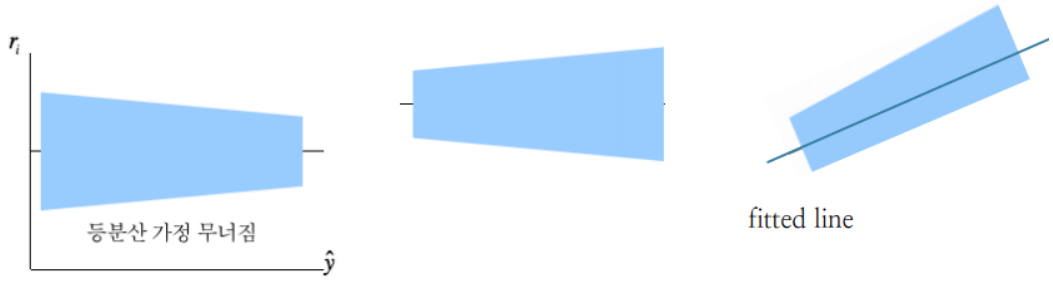
Jarque, Carlos M.; Bera, Anil K. (1980). "Efficient tests for normality, homoscedasticity and serial independence of regression residuals". *Economics Letters*. **6** (3): 255–259. doi:10.1016/0165-1765(80)90024-5.

등분산 가정

진단

잔차와 예측치 산점도 <=> (설명변수와 종속변수) 산점도 ➜ 나팔 fan 모양

종속변수의 값에 의존하여 분산이 커지거나 작아짐 - 등분산 가정이 무너지면 분산이 큰 부분에서 종속변수의 값이 적합선을 많이 벗어난 것이 적합 정도가 떨어진다고 결론 내릴 수 없음, 이는 분산이 다르므로 이상치가 발생할 수 있기 때문이다.



검정방법

- 귀무가설 : 데이터는 등분산성을 갖는다.
- 대립가설 : 데이터는 등분산성을 갖지 않는다.

두 방법 모두 잔차는 등분산성을 따르므로 오차의 등분산성 가정이 만족된다.

Breush-Pagan test

Gujarati, Damodar N.; Porter, Dawn C. (2009). Basic Econometrics (Fifth ed.). New York: McGraw-Hill Irwin. pp. 385-86.

```
import statsmodels.stats.api as sms
sms.het_breuschpagan(fit.resid,fit.model.exog)[0:2]
```

- 유의확률(0.6763…)로 유의확률이 크므로 등분산(귀무가설 만족)을 만족한다.

```
(2.3244229909598446, 0.6763260768386271)
```

Goldfeld-Quandt test

Goldfeld, Stephen M.; Quandt, R. E. (June 1965). "Some Tests for Homoscedasticity". Journal of the American Statistical Association. 60 (310): 539-547.

```
import statsmodels.stats.api as sms
sms.het_goldfeldquandt(fit.resid,fit.model.exog)[0:2]
```

- 유의확률(0.1299…)로 유의확률이 크므로 등분산(귀무가설 만족)을 만족한다.

```
(1.5841952349784663, 0.12998223318685576)
```

해결 방안

문제가 되는 예측변수로 회귀모형 나누기 *) 문제 변수가 1개일 때만 가능함

이분산 heterscedasticity 회귀모형 $y_i = \alpha + \sum_{k=1}^{p} \beta_k x_{ki} + e_i$

변환 : $\frac{y_i}{x_{ki}} = \frac{\alpha}{x_{ki}} + \beta_k + \sum_{k=1}^{p} \beta_k x_{ki}(no\ x_{ki}) + e_i$ (문제변수가 X_k인 경우)

등분산 회귀모형 : 종속변수는 $\frac{y_i}{x_{ki}}$, 절편은 b_k이다.

OLS대신 WLS 가중최소자승법 사용

$min_{\alpha,\beta_1,\ldots,\beta_p} \sum w_i(y_i - \alpha - \sum_{k=1}^{p} \beta_k x_{ki})^2$을 최소화 하는 추정치를 가중최소추정량이라 한다.

가중치 $w_i = \frac{1}{\hat{y}_i^2}$을 사용한다.

```
import statsmodels.api as sm
y=df_clean['Mortality']
X=sm.add_constant(X0)
fit=sm.OLS(y, X).fit()
fitw=sm.WLS(y, X,weights=1./(fit.fittedvalues**2)).fit()
fitw.summary()
```

- 본 예제는 등분산을 만족하나 만족하지 않는다는 가정 하에 실행한 예제이다.
- fitw 에 저장되는 내용은 일반 선형회귀 추정과 동일하다.

	coef	std err	t	P>|t|	[0.025	0.975]
const	765.2585	26.166	29.246	0.000	712.798	817.719
Rain	2.1369	0.554	3.856	0.000	1.026	3.248
%NonWhite	2.9559	0.685	4.314	0.000	1.582	4.330
NOx	-0.4007	0.155	-2.587	0.012	-0.711	-0.090
log_NOxPot	28.9534	6.498	4.456	0.000	15.926	41.980

결정계수 61.1%

독립성 가정

개념

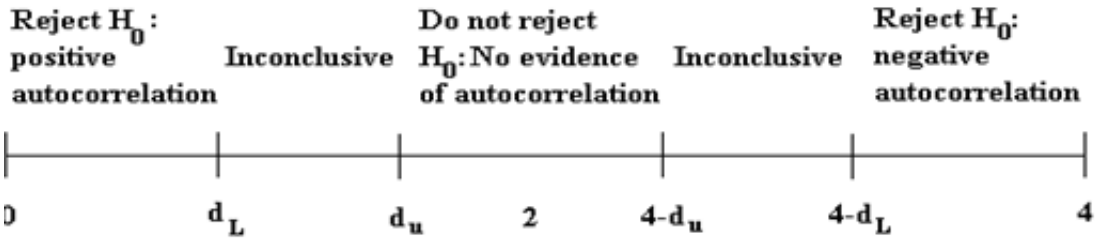
시계열 데이터의 경우 오차항이 전 차항의 오차들에 의해 영향을 받게 되면 오차의 독립성이 파괴된다.

오차항 독립이 아니면 종속변수에 설정된 설명변수가 설명하지 못하는 일정의 패턴이 존재하므로 회귀추정이 불완전함

진단도구

- Durbin Watson 통계량 $$d = \frac{\sum_{t=2}^{T}(e_t - e_{t-1})^2}{\sum_{t=1}^{T} e_t^2},$$
- DW 검정통계량의 값은 $2(1-r)$에 근사한다. 상관계수 r은 (e_t, e_{t-1})의 상관계수 (<=> 오차의 1차 자기상관계수)이다.
- 오차가 독립(자기상관이 존재하지 않음)이면 $r = 0$이고 $DW = 2$에 근사한다.

(DW 이용방법)



positive autocorrelation (양의 상관관계)

- If $DW < D_L$, 양의 상관관계가 존재한다.
- If $DW > D_U$, 자기상관이 존재하지 않는다. 독립이다.
- If $D_L < DW < D_U$, 결론 내릴 수 없음

negative autocorrelation (음의 상관관계) the test statistic (4 – d)

- If $DW > 4 - D_L$, 음의 상관관계가 존재한다.
- If $DW < 4 - D_U$, 자기상관이 존재하지 않는다. 독립이다.
- If $4 - d_u < DW < 4 - d_L$, 결론 내릴 수 없음

해결책

목표변수의 1차 전기 항(y_{t-1})을 예측변수로 사용하거나 종속변수의 차분항($\bigtriangledown Y_t = (Y_t - Y_{t-1})$)을 목표변수로 사용한다.

sm.OLS, sm.WLS 실행하면 DW(Durbin Watson) 자동 출력된다.

	coef	std err	t	P>|t|	[0.025	0.975]
const	768.2807	26.569	28.917	0.000	715.014	821.548
Rain	2.0495	0.558	3.673	0.001	0.931	3.168
%NonWhite	2.9188	0.659	4.426	0.000	1.597	4.241
NOx	-0.4189	0.165	-2.533	0.014	-0.750	-0.087
log_NOxPot	29.4609	6.625	4.447	0.000	16.178	42.744

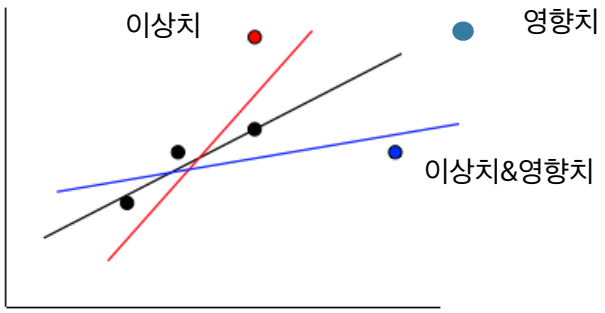
Omnibus: 0.531 **Durbin-Watson:** 1.696

이상치 & 영향치

개념

이상치 outlier : 설명변수 관측치 범위 내에 존재하며 적합 회귀선에서 벗어난 관측치 - (문제) 회귀모형 적합정도, 결정계수를 떨어뜨림 (해결) 삭제 후 회귀모형 적합

영향치 influential : 설명변수 관측값 범위를 벗어났으며 적합 회귀선 상에 있는 관측값 - (문제) 결정계수 값을 과하게 높이고 회귀계수 유의성 매우 높임 - 잘못된 결론 (해결) 설명변수 관측값 주변 관측치를 더 수집한 후 분석, 혹은 제외 후 모형 적합



진단도구

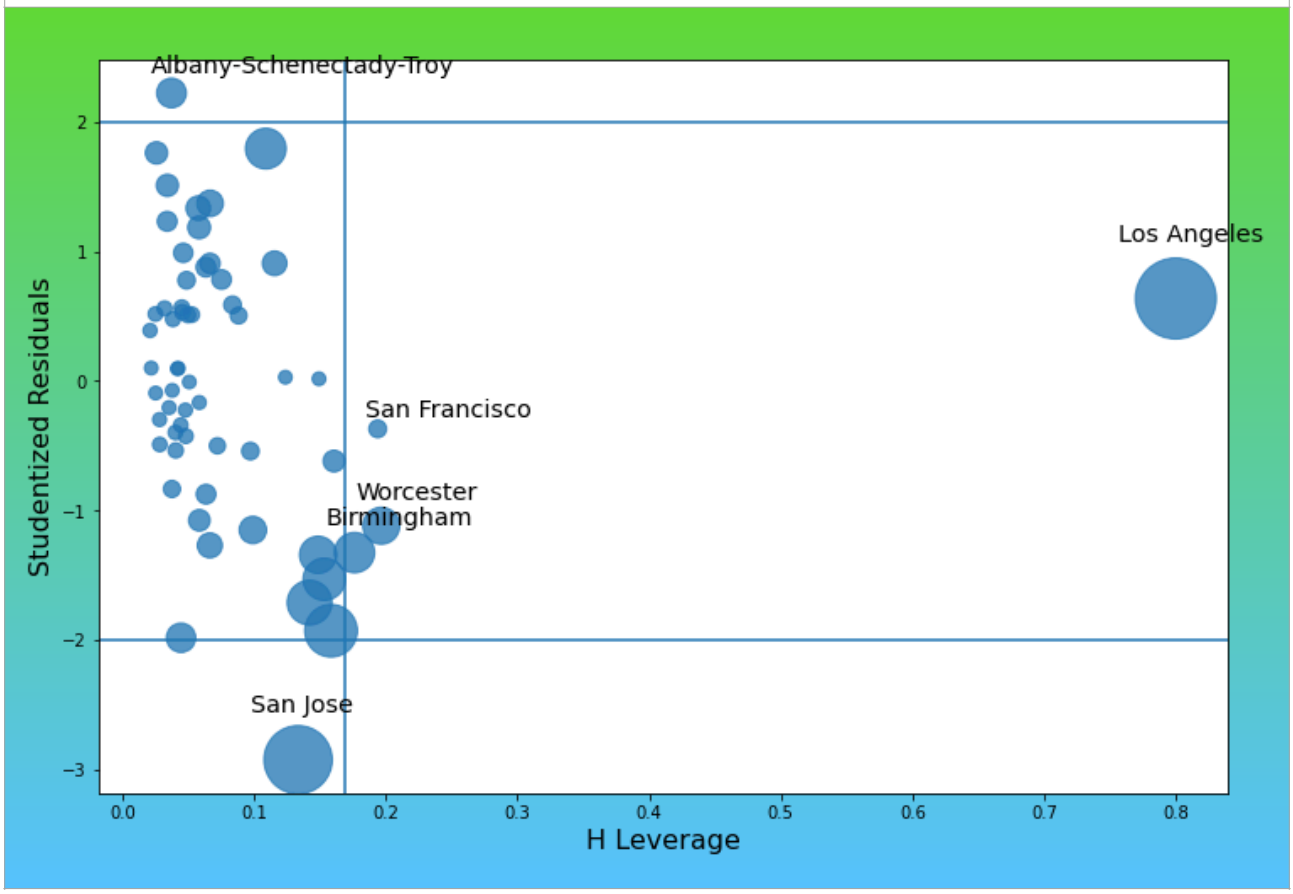
- 원 변수 산점도, 잔차-적합치 산점도
- 표준화 잔차나 스튜던트 잔차 (선호) : ± 2이상이면 이상치
- 레버리지 통계량 $h_{ii} = \underline{x}_i'(X'X)^{-1}\underline{x}_i$: $\bar{h} = \frac{p+1}{n}$(p=설명변수 개수, n =데이터 수) 2배보다 크면 영향치로 진단함

해결책

- 이상치 - 삭제 : 회귀모형의 적합성 높아짐 <=> 결정계수 높아짐
- 영향치 : 결정계수를 커지게 하는 경향이 있음 <=> 제외하고 추정 모형을 예측하는 것이 적절하다.

```
import statsmodels.api as sm
y=df_clean['Mortality']
X=sm.add_constant(X0)
fit=sm.OLS(y, X).fit()
import matplotlib.pyplot as plt
fig, ax=plt.subplots(figsize=(12,8))
fig=sm.graphics.influence_plot(fit,ax=ax, criterion="cooks")
plt.axvline(10/59)
plt.axhline(2)
plt.axhline(-2)
plt.title(' ')
plt.show()
```

- 이상치 진단 기준 ± 2, 영향치 진단 기준 $2*(4+1)/59$
- 이상 도시 : San Jose, Albany
- 영향 도시 : Los Angeles, San Francisco, Wprvester, Birmington



Wrapping Up

이상치 제거

```
import statsmodels.api as sm
y=df_clean['Mortality']
X=sm.add_constant(df_clean[['Rain', '%NonWhite', 'NOx', 'log_NOxPot']])
fit=sm.OLS(y, X).fit()
fit.summary()
```

	coef	std err	t	P>|t|	[0.025	0.975]
const	768.2807	26.569	28.917	0.000	715.014	821.548
Rain	2.0495	0.558	3.673	0.001	0.931	3.168
%NonWhite	2.9188	0.659	4.426	0.000	1.597	4.241
NOx	-0.4189	0.165	-2.533	0.014	-0.750	-0.087
log_NOxPot	29.4609	6.625	4.447	0.000	16.178	42.744

결정계수 62%

- 이상치 2개 제외 되었음

```
infl=fit.get_influence()
df_clean.stres=infl.resid_studentized
df_clean2=df_clean[(df_clean.stres<2) & (df_clean.stres>-2)]
df_clean2.shape  (57, 15)
```

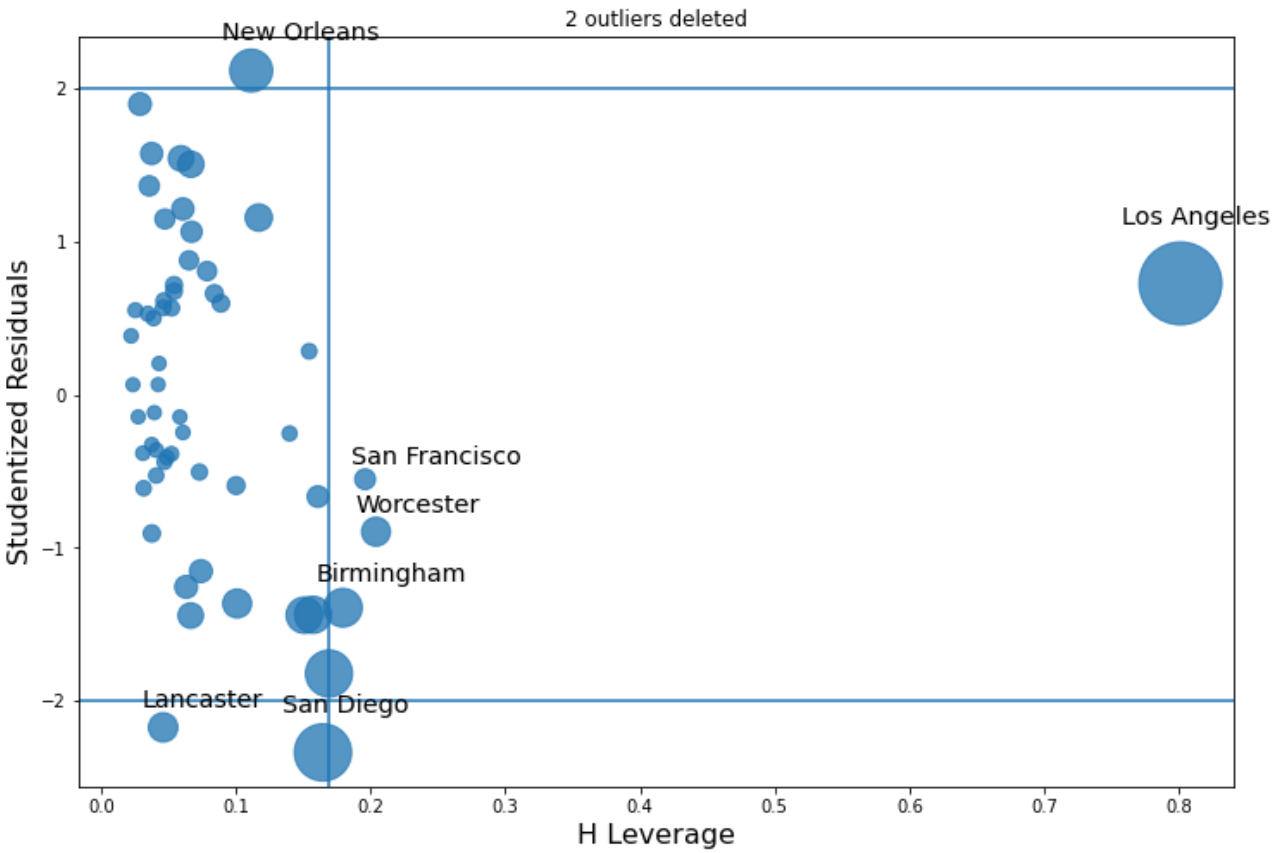
```
import statsmodels.api as sm
y=df_clean2['Mortality']
X=sm.add_constant(df_clean2[['Rain', '%NonWhite', 'NOx',
'log_NOxPot']])
fit=sm.OLS(y, X).fit()
fit.summary()
```

	coef	std err	t	P>|t|	[0.025	0.975]
const	780.5340	24.549	31.795	0.000	731.273	829.795
Rain	1.6660	0.522	3.192	0.002	0.619	2.713
%NonWhite	2.9610	0.602	4.918	0.000	1.753	4.169
NOx	-0.4758	0.153	-3.120	0.003	-0.782	-0.170
log_NOxPot	31.1068	6.063	5.130	0.000	18.940	43.274

결정계수 66.2%로 증가

```
import matplotlib.pyplot as plt
fig, ax=plt.subplots(figsize=(12,8))
fig=sm.graphics.influence_plot(fit2,ax=ax, criterion="cooks")
plt.axvline(10/59)
plt.axhline(2);plt.axhline(-2)
plt.title('2 outliers deleted')
plt.show()
```

- 다시 이상치 3개 진단되었음(New Orleans, Sandiago, Lancer) : 보다 정확한 예측모형을 얻으려면 이상치 제거를 계속 진행해야 한다.
- 영향도시는 동일 : Los Angeles, San Francisco, Wprvester, Birmington



최종회귀모형 : 결정계수=66.2%

$$Mortality = 780 + 1.67 * Rain + 2.96 * NonWhite - 0.47 * NOx + 31.107 * Log(NOxPot)$$

- 양의 영향 : 강우량, 비백인비율, Log(NOxPot) 높을수록 사망률 지수는 높아진다.
- 음의 영향 : NOx(Nitrogen oxides)가 높아질수록 사망율 지수는 낮아진다. 말이 되나? 개별 상관계수도 음이나 회귀계수 부호가 적절하다. 왜? 찾아야 한다. 이것이 통계학이 과학인 이유다. 가구당 가구원수(음), 환경요인과 높은 상관관계, - 대부분 사망율 지수를 낮추는 요인임

	Mortality	JanTemp	JulyTemp	RelHum	Rain	Education	PopDensity	%NonWhite	%WC	pop/house	income	S02Pot	NOx	l
Mortality	1.000	-0.016	0.322	-0.101	0.433	-0.508	0.252	0.647	-0.289	0.368	-0.283	0.419	-0.085	(
JanTemp	-0.016	1.000	0.322	0.086	0.059	0.108	-0.076	0.459	0.208	-0.325	0.198	-0.094	0.334	(
JulyTemp	0.322	0.322	1.000	-0.441	0.472	-0.269	-0.009	0.602	-0.013	0.257	-0.191	-0.071	-0.334	(
RelHum	-0.101	0.086	-0.441	1.000	-0.118	0.186	-0.149	-0.119	0.015	-0.144	0.128	-0.116	-0.053	(
Rain	0.433	0.059	0.472	-0.118	1.000	-0.473	0.084	0.303	-0.114	0.199	-0.362	-0.131	-0.460	(
Education	-0.508	0.108	-0.269	0.186	-0.473	1.000	-0.236	-0.209	0.486	-0.389	0.507	-0.229	0.229	(
PopDensity	0.252	-0.076	-0.009	-0.149	0.084	-0.236	1.000	-0.007	0.253	-0.167	-0.003	0.422	0.158	(
%NonWhite	0.647	0.459	0.602	-0.119	0.303	-0.209	-0.007	1.000	-0.057	0.353	-0.101	0.160	0.019	(
%WC	-0.289	0.208	-0.013	0.015	-0.114	0.486	0.253	-0.057	1.000	-0.347	0.366	-0.063	0.129	(
pop/house	0.368	-0.325	0.257	-0.144	0.199	-0.389	-0.167	0.353	-0.347	1.000	-0.295	-0.010	-0.449	-
income	-0.283	0.198	-0.191	0.128	-0.362	0.507	-0.003	-0.101	0.366	-0.295	1.000	0.068	0.312	(
S02Pot	0.419	-0.094	-0.071	-0.116	-0.131	-0.229	0.422	0.160	-0.063	-0.010	0.068	1.000	0.406	(
NOx	-0.085	0.334	-0.334	-0.053	-0.460	0.229	0.158	0.019	0.129	-0.449	0.312	0.406	1.000	(
log_HCPot	0.125	0.227	-0.413	0.185	-0.485	0.176	0.264	0.148	0.162	-0.224	0.292	0.569	0.738	1
log_NOxPot	0.280	0.181	-0.303	0.097	-0.389	0.032	0.339	0.208	0.108	-0.123	0.245	0.681	0.705	(

예측값, 신뢰구간, 예측구간

회귀모형 $\underline{y} = X\underline{\beta} + \underline{e}$

- OLS 추정치 : $\hat{\underline{\beta}} = (X'X)^{-1}X'\underline{y}$

주어진 예측변수 값 벡터 : $\underline{x}_0$

예측구간

주어진 예측변수 $\underline{x}_0$ 목표변수 적합치 : $\hat{\underline{y}}_0 | \underline{x}_0 = \underline{x}_0\hat{\underline{\beta}} + \underline{e}_0$

- $\underline{e}_0$는 오차이고 평균이 0이므로 추정량은 $\hat{\underline{y}}_0 | \underline{x}_0 = \underline{x}_0\hat{\underline{\beta}}$으로 기대값 추정량과 동일하다.
- 추정량 평균 : $E(\hat{\underline{y}}_0 | \underline{x}_0) = \underline{x}_0\hat{\underline{\beta}}$
- 추정량 분산 : $V(\hat{\underline{y}}_0 | \underline{x}_0) = \sigma^2(I + \underline{x}_0'(X'X)^{-1}\underline{x}_0)$
- 추정량은 목표변수의 선형결합이므로 추정량의 샘플링분포는 정규분포를 따른다.

신뢰구간

주어진 예측변수의 목표변수 기대값 추정량 : $E(\hat{\underline{y}}_0 | \underline{x}_0) = \underline{x}_0\hat{\underline{\beta}}$

- 추정량 평균 : $E(\hat{\underline{y}}_0 | \underline{x}_0) = \underline{x}_0\hat{\underline{\beta}}$
- 추정량 분산 : $V(\hat{\underline{y}}_0 | \underline{x}_0) = \sigma^2 I$
- 추정량은 목표변수의 선형결합이므로 추정량의 샘플링분포는 정규분포를 따른다.

예측구간, 신뢰구간 어느 것을 사용하나? 신뢰구간이 예측구간보다 작으나 예측변수의 개별 관측값이 주어진 경우 목표변수 관측값을 예측하는 것이므로 예측구간을 사용하는 것이 적절하다.

파이썬 코드

```
pred_int=fit2.get_prediction(X)
pred_int.summary_frame()
```

mean은 종속변수 적합치이므로 fit2.fittedvalues 와 동일하다.

city_name	mean	mean_se	mean_ci_lower	mean_ci_upper	obs_ci_lower	obs_ci_upper
Akron	943.667	6.428	930.768	956.567	870.214	1017.121
Allentown	909.086	8.792	891.444	926.729	834.653	983.520
Atlanta	999.955	9.752	980.387	1019.524	925.042	1074.868

새로운 도시[예측변수 순으로 1=절편항, 40=강우량, 15=비백인비율, 10=NOx, 2.5=log_NOxPot]의 사망율 지수를 추정해 보자.

```
pred_new=fit2.get_prediction([1,40,15,10,2.5])
pred_new.summary_frame()
```

	mean	mean_se	mean_ci_lower	mean_ci_upper	obs_ci_lower	obs_ci_upper
0	964.596	5.564	953.432	975.761	891.427	1037.765

표준화 회귀계수

회귀계수는 예측변수 한 단위당 목표변수의 변동 크기이다. 만약 예측변수의 단위가 유사하면 회귀계수의 크기만으로 목표변수의 영향력을 측정할 수 없다.

이를 위하여 예측변수를 표준화 한 후 회귀모형을 추정하면 회귀계수의 크기만으로 예측변수의 영향력을 측정할 수 있다. 예측변수를 표준화 했으므로 상수항을 삽입할 필요는 없다.

```
import statsmodels.api as sm
y=df_clean2['Mortality']
X=df_clean2[['Rain', '%NonWhite', 'NOx', 'log_NOxPot']]
from sklearn.preprocessing import scale
X_std=scale(X,with_mean=True,with_std=True)
fit3=sm.OLS(y, X_std).fit()
fit3.summary()
```

'log_NOxPot'=36.9 > '%NonWhite'=26.4 > 'Nox'=-22 > 'Rain'=18 순으로 크다. 'log_NOxPot'예측변수는 Rain에 비해 사망율 지수에 2배의 영향력을 갖는다.

	coef	std err	t	P>|t|	[0.025	0.975]
x1	18.5815	158.060	0.118	0.907	-298.447	335.610
x2	26.4295	145.913	0.181	0.857	-266.235	319.094
x3	-22.3754	194.712	-0.115	0.909	-412.917	368.166
x4	36.9863	195.739	0.189	0.851	-355.617	429.589

LASSO 방법

처음 2개는 일치하나 Rain이 NOx에 비해 사망율 지수에 대한 영향력 크기가 낮다.

```
from sklearn import linear_model
import matplotlib.pyplot as plt
lassoReg=linear_model.Lasso(alpha=0.1,
fit_intercept=True, normalize=True)
y=df_clean['Mortality']
X=df_clean[['Rain', '%NonWhite', 'NOx',
'log_NOxPot']]
fit=lassoReg.fit(X,y)
var_nm=pd.Series(X.columns,name='Var')
coeff=pd.Series(fit.coef_,name='Coeff')
LASSO=pd.concat([var_nm,coeff],axis=1)
```

	Var	Coeff
0	Rain	1.985
1	%NonWhite	2.915
2	NOx	-0.369
3	log_NOxPot	27.181