5

Diagnosis

Sometimes when you innovate, you make mistakes. It is best to admit them quickly, and get on with improving your other innovations.

–Steve Jobs

introduction

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.

잔차와 진단도구

잔차 residual

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) : $\hat{\alpha}, \hat{\beta}_1, \hat{\beta}_2, \ldots, \hat{\beta}_p$, [행렬] $\underline{\hat{\beta}} = (X'X)^{-1}X'\underline{y}$
- 적합치 : $\hat{y}_i = \hat{\alpha} + \sum_{k=1}^{p} \hat{\beta}_k x_{ki}$, [행렬] $\underline{\hat{y}} = H\underline{y}$, $H = X(X'X)^{-1}X'$, H는 멱등행렬

잔차 정의

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

잔차성질

- 잔차의 평균의 0이고 분산 σ^2이다.
- 잔차는 서로 독립인가? 아니다. 회귀계수 OLS추정($\underline{\hat{\beta}} = (X'X)^{-1}X'\underline{y}$)에는 (x_i, y_i) 모든 관측치가 포함되어 있기 때문임
- $\sum x_i r_i = 0$: 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 과정

- 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) 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 student 잔차

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

- 잔차를 t-분포를 따르는 통계량으로 만든 것으로 ± 2이면 이상치(혹은 영향치)로 판단
- h_{ii} : Hat 행렬의 대각 원소로 leverage 레버리지(지렛대)로 정의되먀 영향치 판단에 사용된다.

표준화/스튜던트 제외 standardized & deleted 잔차

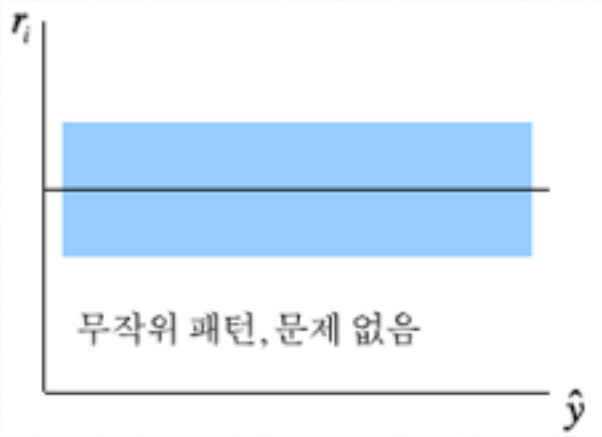
- $z_i = \frac{r_{(i)} - \bar{r}}{\sqrt{MSE_{(i)}}}, \; s_i = \frac{r_(i)}{\sqrt{MSE_{(i)}/(1-h_{ii})}}$
- i번째 관측치를 제외하고 회귀모형을 추정한 후 얻은 적합치를 사용하여 얻은 잔차로 표준화/스튜던트 잔차에 비해 더 정확한 개념의 잔차이지만 현실에서는 자주 사용하지 않음

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

파이썬 사례연구

사례 데이터

원데이터

```
import pandas as pd
smsa=pd.read_csv('http://203.247.53.31/Stat_Notes/example_data/SMSA_USA.csv')
smsa[['city_nm','state_nm','tmp_nm']]=smsa['city_name'].str.split(',',expand=True)
smsa.loc[smsa.tmp_nm.isna()==False, 'state_nm']=smsa.tmp_nm
smsa.drop(columns=['city_name','tmp_nm'],inplace=True)
smsa.columns
```

```
Index(['jan_temp', 'july_temp', 'humidity', 'rainfall', 'mortality_index',
       'education', 'pop_density', 'non_white_ratio', 'white_color_ratio',
       'population', 'person_houshold', 'household_income', 'HCPot', 'NOxPot',
       'SO2Pot', 'southern', 'city_nm', 'state_nm'],
```

정규변환 후

```
import numpy as np
def rskewed(k):
  if (shapiro(np.sqrt(smsa.iloc[:,k]))[1]<0.05) & (shapiro(np.log(smsa.iloc[:,k]))[1]<0.05):
        print('No Appropriate Normal Transformation(right skewed)')
  else:
    if(shapiro(np.sqrt(smsa.iloc[:,k]))[1]<shapiro(np.log(smsa.iloc[:,k]))[1]):
      print(k,':',smsa.columns[k],'로그변환 : 유의확률',shapiro(np.log(smsa.iloc[:,k]))[1])
      smsa[smsa.columns[k]+str('_log')]=np.log(smsa.iloc[:,k])
    else:
      print(k,':',smsa.columns[k],'제곱근변환 : 유의확률',shapiro(np.sqrt(smsa.iloc[:,k]))[1])
      smsa[smsa.columns[k]+str('_sqrt')]=np.sqrt(smsa.iloc[:,k])
for k in [0,6,7,9,11,12,13,14]:
  rskewed(k)
```

```
No Appropriate Normal Transformation(right skewed)
6 : pop_density 로그변환 : 유의확률 0.6019314527511597
7 : non_white_ratio 제곱근변환 : 유의확률 0.24234969913959503
9 : population 로그변환 : 유의확률 0.388794809579849 24
No Appropriate Normal Transformation(right skewed)
No Appropriate Normal Transformation(right skewed)
13 : NOxPot 로그변환 : 유의확률 0.451902508735656 74
No Appropriate Normal Transformation(right skewed)
```

```
smsa_clean=smsa.drop(['pop_density','non_white_ratio','population','NOxPot'],axis=1)
smsa_clean.set_index(['state_nm','city_nm'],inplace=True)
smsa_clean.columns
```

```
Index(['jan_temp', 'july_temp', 'humidity', 'rainfall', 'mortality_index',
       'education', 'white_color_ratio', 'person_houshold', 'household_income',
       'HCPot', 'S02Pot', 'southern', 'pop_density_log',
       'non_white_ratio_sqrt', 'population_log', 'NOxPot_log'],
```

변수선택 결과

1월기온, 강우량, 교육기간, 비백인비율_제곱근, NOxPot_로그, 가구원수, 인구밀도_로그

SO2Pot, 가구원수, 7월소득, 사무직비율, 가구소득

다중공선성 VIF 진단 결과

- Index(['rainfall', 'non_white_ratio_sqrt', 'NOxPot_log', 'S02Pot'] - 모든 변수 고려하여 다중공선성 진단결과와 동일

그러므로 변수선택, 다중공선성 진단 어느 것이든 먼저해도 되므로 변수 숫자를 줄인 후 다중공선성 진단하는 것이 적절함

잔차 산점도 그리기

```
import statsmodels.api as sm
X=sm.add_constant(smsa_clean[['rainfall', 'non_white_ratio_sqrt', 'NOxPot_log', 'S02Pot']])
y=smsa_clean['mortality_index']
fit=sm.OLS(y, X).fit(); print(fit.summary())
```

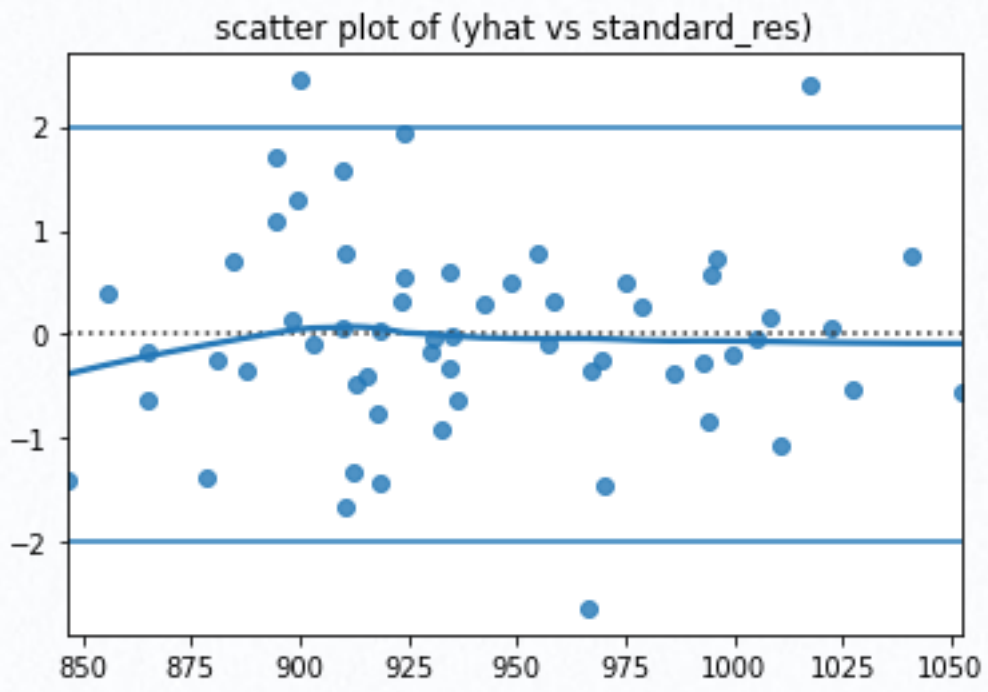
	coef	std err	t	P>|t|
const	749.9201	27.390	27.380	0.000
rainfall	2.3070	0.536	4.300	0.000
non_white_ratio_sqrt	21.1766	4.500	4.706	0.000
NOxPot_log	8.3901	6.853	1.224	0.226
S02Pot	0.2734	0.115	2.381	0.021

결정계수 62%

```
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()
```

fit.resid_pearson : 표준화 잔차임, 3개 관측치가 이상치로 판단됨

잔차 산점도에서 패턴이 발견되지 않았고 등분산성도 만족해 보이므로 선형모형 추정은 적절해 보인다.



선형성

선형성 Linearity

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

회귀모형 전처리 작업(종속변수와 해당 설명변수의 산점도)에서 사전 진단 후 해결하므로 큰 문제는 없으나 변수들간의 관계가

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.

예측변수 4개인 경우 : ['rainfall','S02Pot','non_white_ratio_sqrt','NOxPot_log']

- 검정통계량 행렬이 singular이어서 계산 불가능

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

```
/usr/local/lib/python3.6/dist-packages/statsmodels/sandbox/stats/diagnostic.
  rresid_scaled = rresid/np.sqrt(rvarraw)   #this is N(0,sigma2) distributed
Ttest_1sampResult(statistic=nan, pvalue=nan)
```

예측변수 4개인 경우 : ['non_white_ratio_sqrt','NOxPot_log'] 결정계수 41.6%

	coef	std err	t	P>|t|
const	830.0761	19.470	42.633	0.000
non_white_ratio_sqrt	29.0494	5.104	5.692	0.000
NOxPot_log	7.7094	5.504	1.401	0.167

선형성을 만족한다.

```
Ttest_1sampResult(statistic=-1.3128649183029175, pvalue=0.1946801876580764)
```

정규성

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.

앞의 값은 검정통계량이며 2번째 값은 유의확률로 모든 정규성 검정 방법 결과는 동일하게 유의확률이 매우 크므로 잔차는 정규분포를 따른다.

Shapiro Wilks Test

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

```
(0.9764071106910706, 0.30617254972457886)
```

Omni Test

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

```
(2.4099034575110974, 0.2997064663681783)
```

Jarque-Bera Test

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

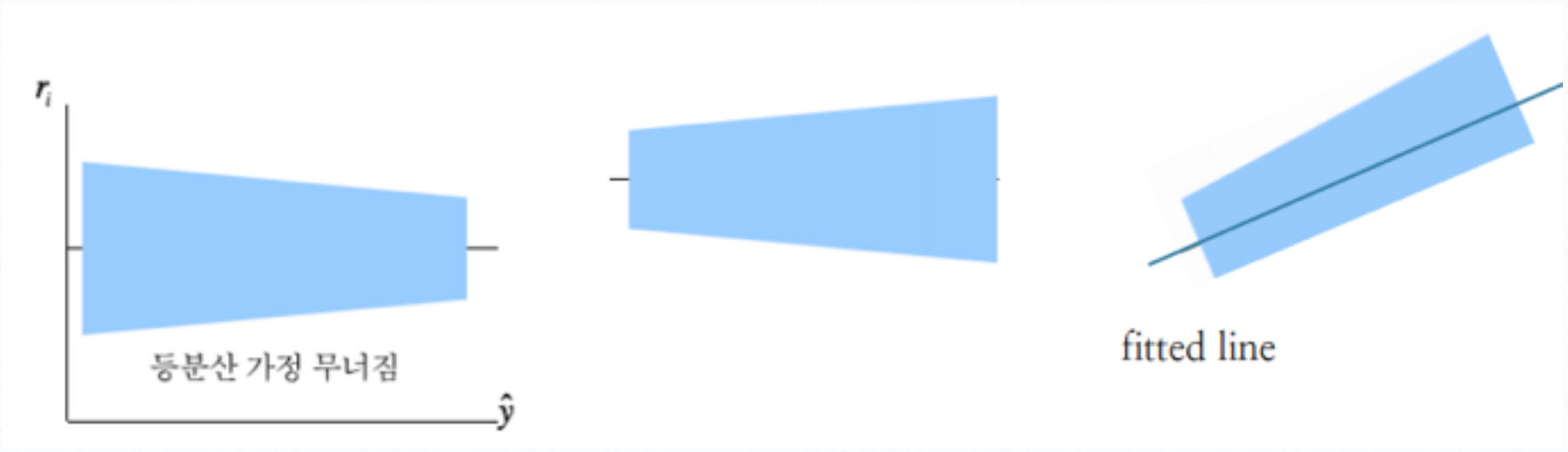
```
(1.6269737093137193, 0.4433096119893797)
```

등분산성

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 크므로 $\sigma_i^2 \propto x_i$



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.het_breuschpagan(fit.resid,fit.model.exog)[0:2]
```

```
(4.603492413842828, 0.3304517157923367)
```

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]
```

```
(1.2360606749433112, 0.30093136098479994)
```

해결 방안

문제가 되는 예측변수로 회귀모형 나누기 *)단순 회귀모형일 때만 가능함

- 이분산 heterscedasticity 회귀모형 $y_i = \alpha + \sum_{k=1}^{p} \beta_k x_{ki} + e_i$
- 변환 : $\frac{y_i}{x_{ki}} = \frac{\alpha}{x_{ki}} + \beta_k + \sum_{k=1}^{p} \beta_k x_{ki}(no\ x_{ki}) + e_i$
- 등분산 회귀모형 : 종속변수는 $\frac{y_i}{x_{ki}}$, 절편은 β_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.

[등분산성을 만족하였지만 예제임 - 가중치는 적합치의 역수임]

```
import statsmodels.api as sm
y=smsa_clean['mortality_index']
X=sm.add_constant(smsa_clean[['rainfall','non_white_ratio_sqrt', 'NOxPot_log','SO2Pot']])
fit=sm.OLS(y, X).fit()
fitw=sm.WLS(y, X,weights=1./(fit.fittedvalues**2)).fit()
print(fitw.summary())
```

결정계수 61.2%

	coef	std err	t	P>|t|
const	749.4647	27.456	27.297	0.000
rainfall	2.3758	0.537	4.423	0.000
non_white_ratio_sqrt	20.7074	4.582	4.519	0.000
NOxPot_log	7.7735	6.831	1.138	0.260
S02Pot	0.2872	0.120	2.391	0.020

독립성

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

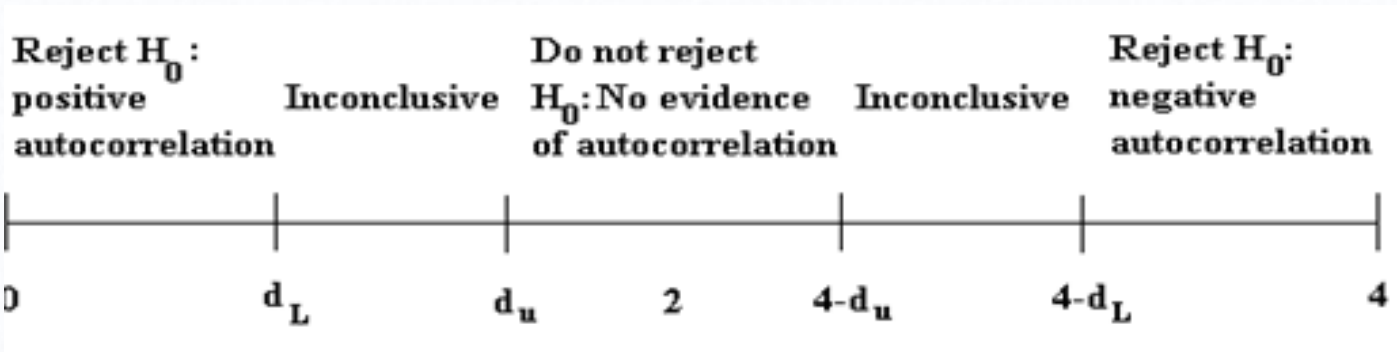
진단도구

시계열 데이터 모형에만 진단

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 독립이면 $r = 0$ 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
const	749.9201	27.390	27.380	0.000	695.007
rainfall	2.3070	0.536	4.300	0.000	1.231
non_white_ratio_sqrt	21.1766	4.500	4.706	0.000	12.154
NOxPot_log	8.3901	6.853	1.224	0.226	-5.350
S02Pot	0.2734	0.115	2.381	0.021	0.043

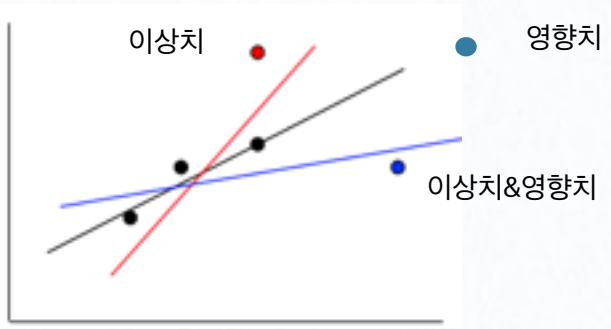
Omnibus:	2.410	Durbin-Watson:	1.575

영향치_이상치

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) : +2 이상이면 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

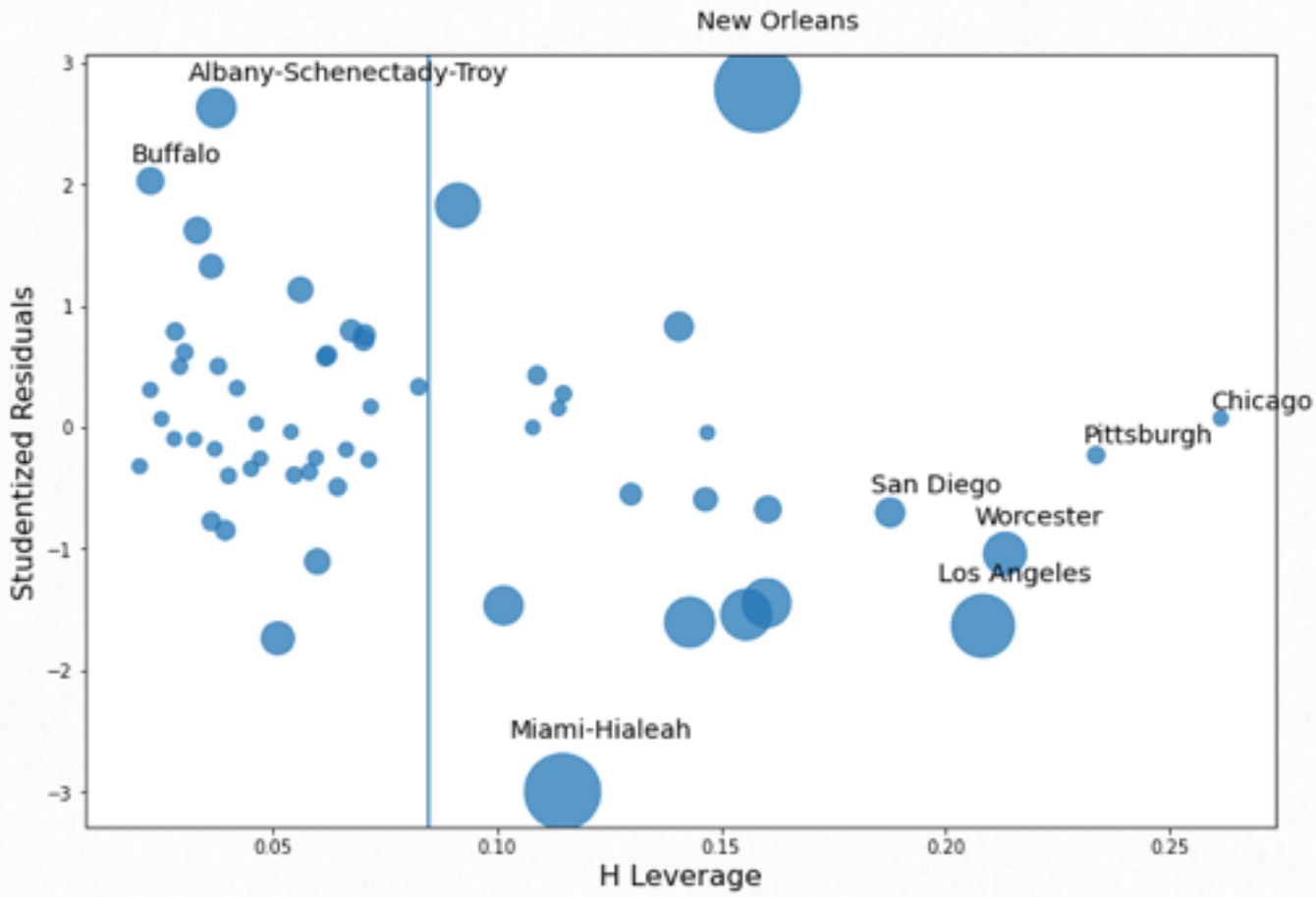
해결책

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

```
df=smsa_clean.reset_index('state_nm') #주도 이름 행인덱스에서 제외
import statsmodels.api as sm
X=sm.add_constant(df[['rainfall','non_white_ratio_sqrt', 'NOxPot_log','SO2Pot']])
y=df['mortality_index']
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(5/59);plt.title(' '); plt.show()
```

New Orions, Alabany : 사망율 높은 이상 도시, Maimi : 사망율 낮은 이상 도시

영향 도시 : 시카고, 피치버그



이상치 3개 제외(스튜던트 잔차 ±2) 결정계수 71.2% (이상치 제거 전 62%)

```
smsa_clean.stres=infl.resid_studentized
smsa_clean2=smsa_clean[(smsa_clean.stres<2) & (smsa_clean.stres>-2)]
smsa_clean2.shape
```

```
(56, 16)
```

영향치 2개 제외(cook 0.2 이상) 결정계수 66.7% (영향치 제거 전 62%)

```
smsa_clean['cook']=infl.cooks_distance[0]
smsa_clean3=smsa_clean[smsa_clean['cook']<0.2]
smsa_clean3.shape
```

```
(57, 17)
```

Wrapping Up

최종 모형

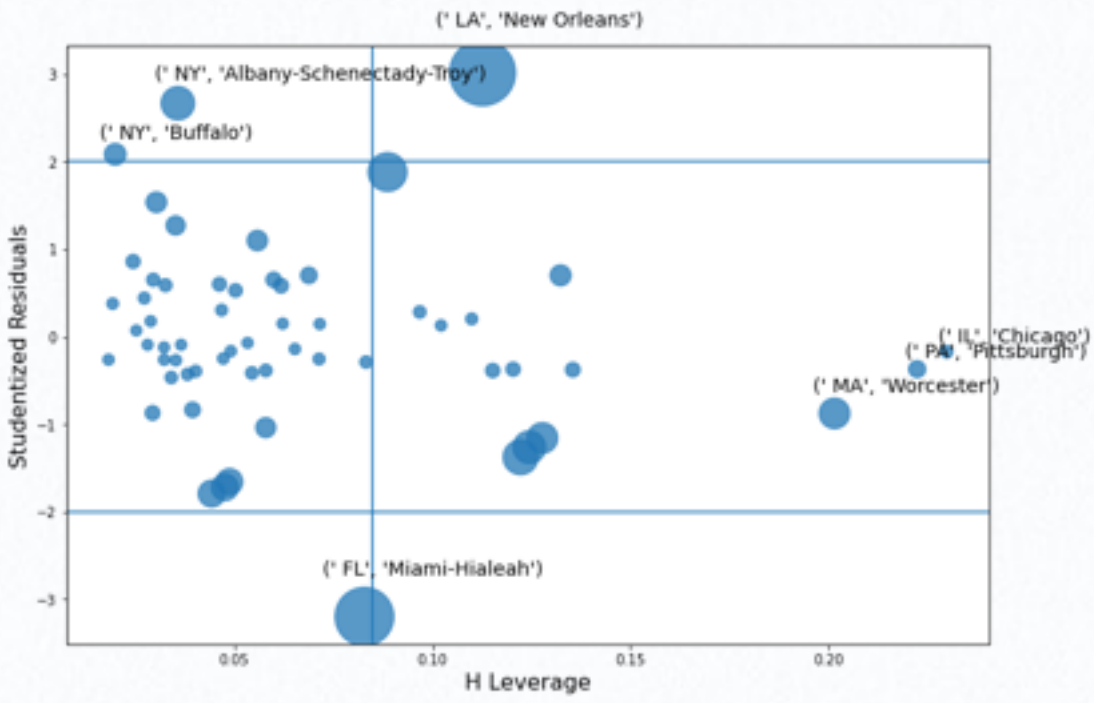
- 변수선택 결과 최종 얻은 NOxPot_log : 유의하지 않아 제외하였음

```
import statsmodels.api as sm
X=sm.add_constant(smsa_clean[['rainfall','non_white_ratio_sqrt','SO2Pot']])
y=smsa_clean['mortality_index']
fit=sm.OLS(y, X).fit(); print(fit.summary())
```

```
==============================================================================
                 coef    std err          t      P>|t|
------------------------------------------------------------------------------
const        771.5640     21.014     36.716      0.000
rainfall       2.0005      0.477      4.197      0.000
non_white_ratio_sqrt    22.6356      4.359      5.193      0.000
SO2Pot         0.3669      0.086      4.263      0.000
==============================================================================
```

결정계수 : 60.9%

이상치 제외




```
infl=fit.get_influence()
smsa_clean.stres=infl.resid_studentized
smsa_clean2=smsa_clean[(smsa_clean.stres<2) & (smsa_clean.stres>-2)]
```

결정계수 : 74.4% (n=55, 이상치 4개 제외)

- 이상치 제거 영향 : 결정계수 높아짐,

	coef	std err	t	P>|t|	[0.025	0.975]
const	762.6993	16.539	46.117	0.000	729.497	795.902
rainfall	2.2147	0.372	5.955	0.000	1.468	2.961
non_white_ratio_sqrt	21.4821	3.427	6.268	0.000	14.601	28.363
S02Pot	0.3892	0.066	5.886	0.000	0.256	0.522

최종회귀모형 : $MO = 763 + 2.21Rain + 21.5NW_{sqrt} + 0.38SO2Pot$

- 양의 영향 : 강우량, 비백인비율, SO2Pot 높을수록 사망률 지수는 높아진다.

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 : $\underline{x}_k$

예측구간

주어진 예측변수의 목표변수 추정량 : $\hat{\underline{y}}_k | \underline{x}_k = \underline{x}_k\hat{\underline{\beta}} + \underline{e}_k$

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

신뢰구간

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

- 추정량 평균 : $E(E(\hat{\underline{y}}_k | \underline{x}_k)) = \underline{x}_k\hat{\underline{\beta}}$

- 추정량 분산 $V(E(\underline{\hat{y}}_k \mid \underline{x}_k)) = = \sigma^2(\underline{x}_k'(X'X)^{-1}\underline{x}_k)$
- 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.

```
p=fit.get_prediction(X)
p.summary_frame()
```

		mean	mean_se	mean_ci_lower	mean_ci_upper	obs_ci_lower	obs_ci_upper
state_nm	city_nm						
OH	Akron	929.114084	4.185916	920.710506	937.517661	867.772335	990.455833
PA-NJ	Allentown	892.201415	9.268784	873.593553	910.809276	828.652673	955.750156
GA	Atlanta	987.959832	8.744795	970.403924	1005.515740	924.711117	1051.208547
MD	Baltimore	1044.209981	11.037579	1022.051117	1066.368845	979.532278	1108.887684
AL	Birmingham	1041.389435	11.246191	1018.811763	1063.967106	976.567052	1106.211817

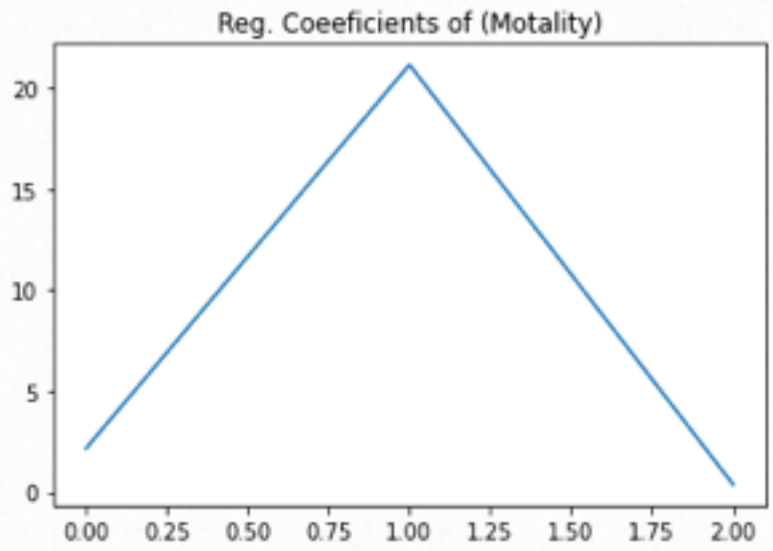
만약 새로운 데이터에 대한 예측값과 신뢰구간을 구한다면 X에 새로운 데이터프레임을 입력하면 된다.

영향력 비교 [LASSO]

전통적인 방법에서는 표준화 회귀계수(예측변수를 표준화한 계수)를 이용하였지만 빅데이터에서는 L1 정규화 방법인 LASSO 이용하는 것이 적절하다.

비백인비율_제곱근 예측변수가 사망율_지수(목표변수)에 가장 큰 영향을 미친다.

```
from sklearn import linear_model
import matplotlib.pyplot as plt
lassoReg=linear_model.Lasso(alpha=0.1, fit_intercept=True, normalize=True)
y=smsa_clean2['mortality_index']
X=smsa_clean2[['rainfall','non_white_ratio_sqrt','SO2Pot']]
fit_lasso=lassoReg.fit(X,y)
plt.title('Reg. Coeeficients of (Motality)')
plt.plot(fit_lasso.coef_); plt.show()
```



	Var	Coeff
0	rainfall	2.150825
1	non_white_ratio_sqrt	21.140727
2	SO2Pot	0.378238

```
var_nm=pd.Series(X.columns,name='Var')
coeff=pd.Series(fit_lasso.coef_,name='Coeff')
pd.concat([var_nm,coeff],axis=1)
```