SFERS DATA Seminar with Python: 2

Yitae Kwon

SFERS of SNU

2024-1

Contents

Visualization and Representation

실증연구에서 가장 중요한 것?

- 연구 주제? 연구 방법론? 연구 결과? X
- ~~연구 결과가 정책 결정권자(policy maker)의 믿음과 부합하는가? 즉 그 사람이 맘대로 할 수 있는 실증적 근거를 제공해주는가? O~~
- 연구 결과를 얼마나 잘 정리해서 시각화하고 표현하는가? △ O!!!!

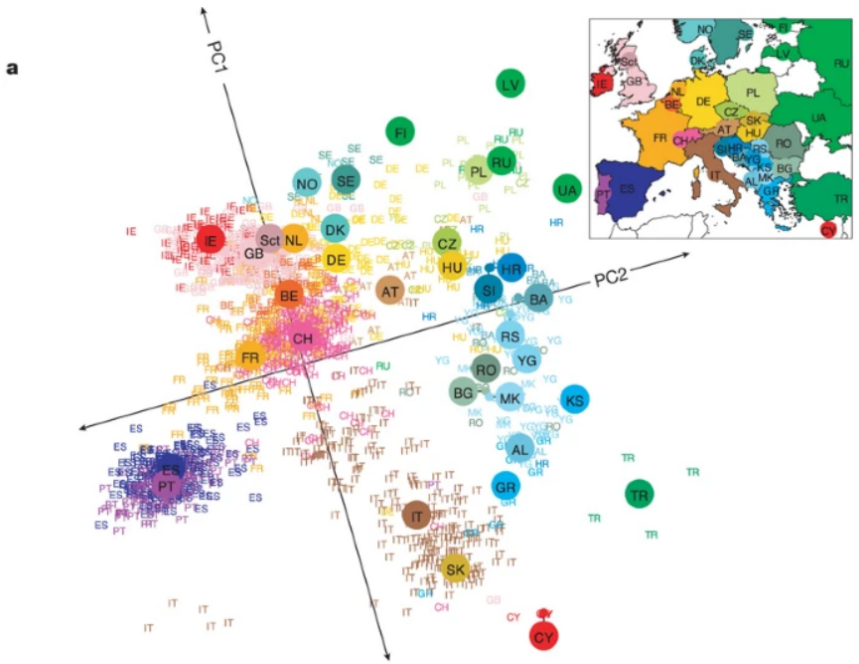


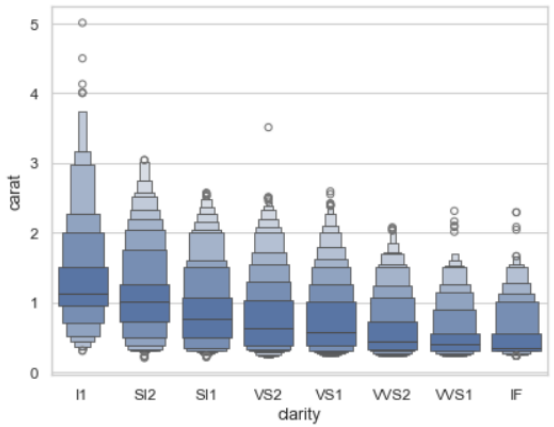
Figure: 유럽 지역 사람들의 유전데이터가 가진 정보와 유럽의 지도

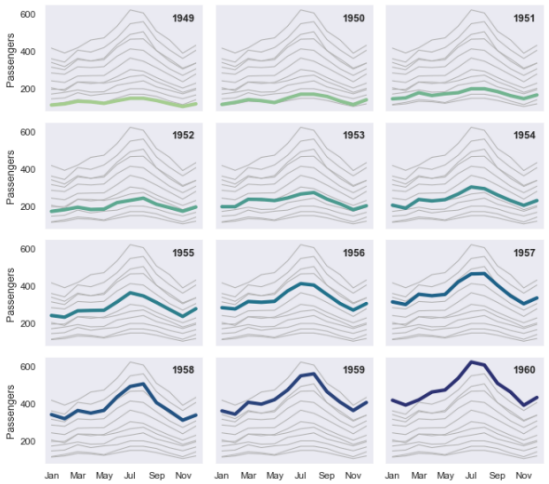
Visualization in Python

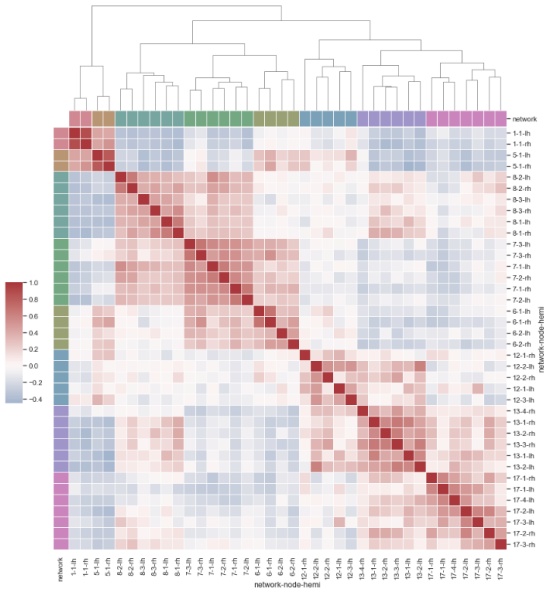
- **Matplotlib** is a comprehensive library for creating static, animated, and **interactive** visualizations in **Python**. Matplotlib makes easy things easy and hard things possible.

- **Seaborn** is a **Python** data visualization library based on **matplotlib**. It provides a high-level interface for drawing attractive and informative **statistical** graphics.

Fancy Examples







우리는 간단한 확인 과정에서는 `matplotlib`을, 최종 프레젠테이션 과정에서는 `seaborn`을 사용하는 경우가 많습니다.

도움이 되는 링크들

- Matplotlib cheet sheet: `https://matplotlib.org/cheatsheets/`
- Matplotlib tutorials: `https://matplotlib.org/stable/tutorials/index.html`
- Seaborn cheet sheet: `https://www.kaggle.com/code/themlphdstudent/cheat-sheet-seaborn-charts`
- Seaborn tutorial: `https://seaborn.pydata.org/tutorial.html`
- Seaborn으로 만들 수 있는 플롯들: `https://seaborn.pydata.org/examples/index.html`

시각화한 자료를 정리하기

최근 자연과학계와 경제학계와 같이 수식이 많이 포함되는 학문을 주로 하는 학계에서는 수식 조판에 최적화된 언어 LaTeX을 자주 사용하고 있습니다.
(LaTeX is a high-quality typesetting system; it includes features designed for the production of technical and scientific documentation. LaTeX is the de facto standard for the communication and publication of scientific documents.)
한편 무료로 웹에서도 LaTeX을 조판하는 툴이 있는데, 바로 **Overleaf**입니다.

회원가입 후 무료로 LaTeX문서를 자유롭게 이용할 수 있습니다. 가장 큰 장점은 여러 저널들의 스타일이 템플릿으로 저장되어 있어, 템플릿을 불러와 본문만 채우면 됩니다. 또한 공유와 협동 작업 역시도 간편합니다.

시각화한 자료를 정리하기

단순히 세로로 된 A4 용지만 채울 수 있는 게 아니라, 수식이 많이 포함된 발표자료를 만들 때에도 LaTeX을 사용할 수 있습니다. beamer라는 패키지를 이용하면, 다양한 스타일의 슬라이드를 만들 수 있습니다. 파워포인트와 달리 학술발표에 최적화된 정적이고 꾸밀 필요가 별로 없는 pdf 기반 슬라이드를 제공하기에, 강의자료 등에서 정말 많이 사용됩니다.

마지막으로, LaTeX양식은 워드, 한글을 포함한 다양한 문서 조판 프로그램에서 수식을 작성하는 데 사용할 수 있습니다. 특히, Jupyter Notebook에서 글씨를 쓰기 위해 사용되는 markdown cell에서도 LaTeX양식의 사용이 가능합니다. 다만 이 세미나의 궁극적인 목적은 데이터 분석법을 익히는 것임과 동시에, LaTeX문법은 배운다고 해서 알 수 있는 게 아닙니다. 따라서 이 세미나에서는 다루지 않고, 과제(HW2)에서 연습해 볼 기회를 드리겠습니다.

재현성 위기와 p-해킹

재현성 위기(replication(reproducibility) crisis)
: 자연과학 및 사회과학 연구에서 대부분의 연구가 재현이 불가능한 것으로 밝혀졌습니다. 수많은 경우 데이터를 제멋대로 가공하거나, 연구가설에 부합하도록 조작합니다. 혹은 결과가 맘대로 나오지 않을 경우 출판하지 않는 **출판 편의**(publication bias)가 있기도 합니다. 재현할 경우 유명 저널에 실린 논문들조차도 10건 중 6건 정도가 원본과는 상이한 결과가 나온다는 사실이 밝혀지기도 했습니다.
결과가 독자에게 신뢰 가능한 것으로 여겨지려면, 데이터 수집부터 분석 결과가 투명하게 공개되어야만 합니다. 허나 대부분의 연구는 2011년 재현성 위기 이후에도 아직도 raw data를 공개하지 않고 있습니다.(...) 데이터를 이용한 연구의 경우, 법적/윤리적 문제가 있지 않는 한 그 데이터를 GitHub 등에 공개하여 다른 연구자들이 이를 재현하고, 그 신뢰성을 확인할 수 있도록 해야 합니다.

재현성 위기와 p-해킹

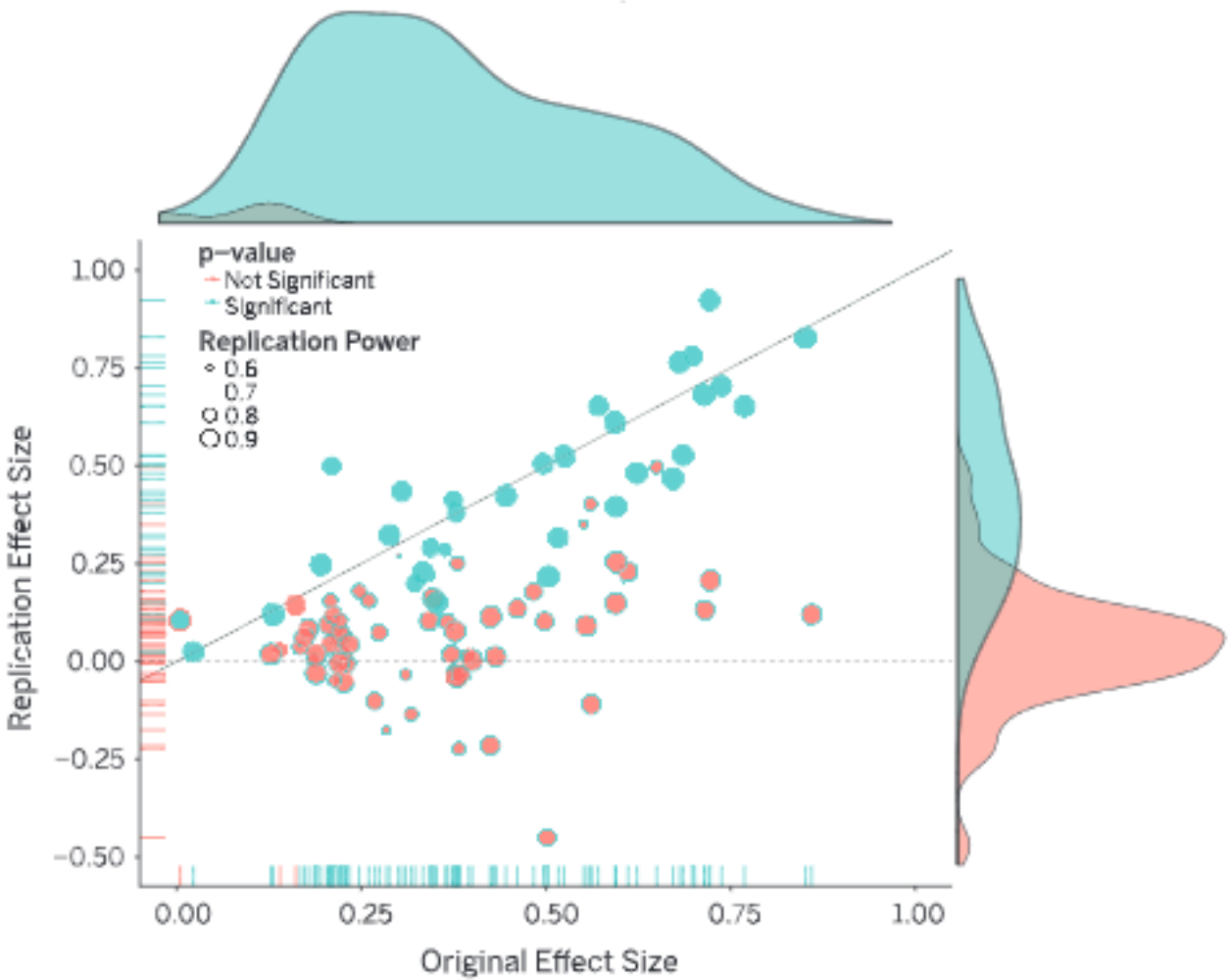


Figure: effect size and p value

재현성 위기와 p-해킹

p-해킹(p-hacking)
: 일반적으로 사용되는 유의수준인 0.01, 0.05, 0.1에 맞추기 위하여 p값을 조금씩 낮추는 학계의 관행

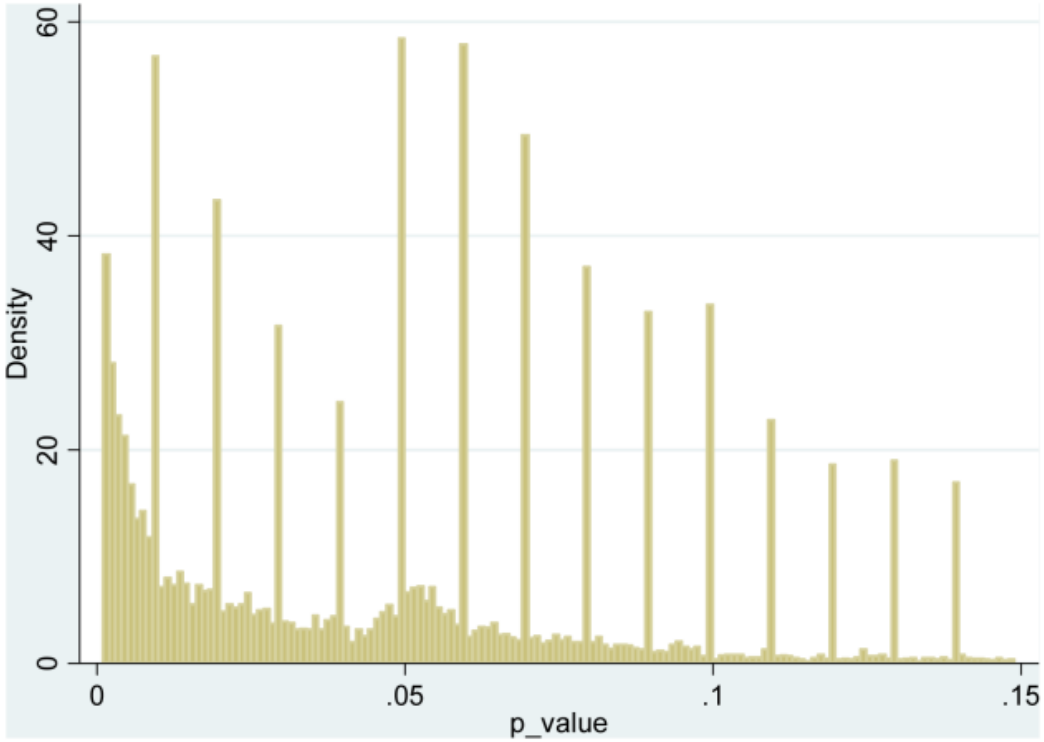


Figure: distribution of reported p values

도움이 되는 링크들

- LaTeX/Overleaf 튜토리얼(서울대 물리천문학부): https://glittery-vise-fbb.notion.site/LaTeX-How-to-38b7bb9106834228962c4c56a4b64030
- KTUG 한글 TeX사용자 그룹: http://www.ktug.org/xe/
- TeX기호를 그리면 명령어를 알려주는 사이트: https://detexify.kirelabs.org/classify.html
- TeX표를 그리면 코드를 알려주는 사이트: https://www.tablesgenerator.com/
- 사진 혹은 손글씨로 수식을 쓰면 코드로 변환하는 사이트: https://mathpix.com/image-to-latex

Multiple Linear Regression

Multiple Linear Regression

다중회귀분석(Multiple Linear Regression)은 아래의 형태를 가지고 있습니다.

$$Y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_k x_{ik} + \epsilon_i$$

이때 ϵ_i는 $\mathcal{N}(0, \sigma^2)$를 따르는 I.I.D. 확률변수이며, $\mathbb{E}[\epsilon_i | x_{i1}, \cdots, x_{ik}] = 0$이라 가정합니다. 이제 설명변수 x의 개수가 이제 k개가 되었습니다. 이제 우리가 n개의 observation $(Y_i, x_{i1}, \cdots, x_{ik})_{i=1,2,\cdots,n}$을 관찰했다고 생각해 봅시다. 그러면 n개의 $y_i = Y_i$의 결정 과정은 아래처럼 쓸 수 있습니다.

$$\begin{aligned}
y_1 &= \beta_0 + \beta_1 x_{11} + \cdots + \beta_k x_{1k} + \epsilon_1 \\
y_2 &= \beta_0 + \beta_1 x_{21} + \cdots + \beta_k x_{2k} + \epsilon_2 \\
&\vdots \\
y_n &= \beta_0 + \beta_1 x_{n1} + \cdots + \beta_k x_{nk} + \epsilon_n
\end{aligned}$$

이를 행렬 형태로 쓴다면,

$$y = X\beta + \epsilon$$

이고, 이들은 각각

$$y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, X = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1k} \\ 1 & x_{21} & \cdots & x_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{nk} \end{pmatrix}, \beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_k \end{pmatrix}, \epsilon = \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{pmatrix}$$

입니다. 이때 우리는 y와 ϵ은 n차원 확률벡터로 바라보며, X와 β는 이미 외생적으로 관측된 $n \times (k+1)$ 행렬과 정해진 계수의 $k+1$차원 벡터로 취급합니다. 차원을 계산해보면, 좌변과 우변이 모두 길이 n인 벡터가 됨을 알 수 있습니다.

Assumptions of Multiple Linear Regression

모형

$$y = X\beta + \epsilon$$

에서, 우리는 원활한 추정을 위하여 아래를 가정합니다.

1. **Linearity**: 선형모형은 참이다. 즉 $\mathbb{E}[y] = X\beta$이다.
2. $\epsilon \sim \mathcal{N}(0, \sigma^2 I_n)$
   - **Independence**: $\epsilon_1, \cdots, \epsilon_n$은 상호 독립이다.
   - **Homoskedasticity**: ϵ_i는 동일한 분산 σ^2을 가진다.
   - **Normality**: ϵ_i는 정규분포를 따른다.
3. **No Endogeneity**: 모형은 참이며 누락된 변수가 없어서, $\mathrm{E}[\epsilon|X] = 0$.
4. **Identifiability**: 행렬 X는 full rank이다. 즉 $(X^T X)^{-1}$가 존재한다.

Least Squares Estimator

최소제곱법을 통하여 β의 추정량

$$\hat{\beta} = \begin{pmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \vdots \\ \hat{\beta}_k \end{pmatrix}$$

를 찾고자 하는 경우, 푸는 최소화 문제는

$$\hat{\beta} = \underset{\beta}{\text{argmin}} (y - X\beta)^T (y - X\beta)$$

으로 주어집니다. 이를 **normal equation**이라 부릅니다. 또한, 이러한 방식을 **OLS**(ordinary least squres)라고 부릅니다.

Least Squares Estimator

최적화 문제를 푼다면, **최소제곱추정량**(Least Squares Estimator; LSE)는 아래와 같이 주어집니다.

$$\hat{\beta} = (X^T X)^{-1} X^T y$$

이때, 이 추정량이 유일하게 **식별(identify)**될 조건(identifiability condition)은 $(X^T X)^{-1}$가 존재하는 것입니다.
한편 이를 이용하면 **hat matrix** $H = X(X^T X)^{-1} X^T$를 정의할 때

$$\hat{y} = X\hat{\beta} = Hy$$
$$e = y - \hat{y} = (I_n - H)y$$

이므로 예측값 $\hat{y}$와 추정된 잔차 e도 쉽게 구할 수 있습니다.

Unbiasedness of Estimators

앞서 구한 최소제곱추정량들은 모두 **불편추정량**(unbiased estimator; UE)입니다. 따라서,

$$\mathbb{E}[\hat{\beta}] = \beta$$
$$\mathbb{E}[\hat{y}] = X\beta$$
$$\mathbb{E}[e] = 0_n$$

임이 성립합니다. 일반적으로, 추정량을 판단하는 과정에서 우리는 해당 추정량에 편의가 없을수록 좋아합니다.

Note: 모수 θ에 대한 추정량 $\hat{\theta}$의 **편의**(bias)는

$$\text{bias}(\hat{\theta}) = \mathbb{E}[\hat{\theta}] - \theta$$

로 정의합니다. $\text{bias}(\hat{\theta}) \neq 0$이면, $\hat{\theta}$는 **편의추정량**(biased estimator)입니다.

Gauss-Markov Theorem

Gauss-Markov Theorem

최소제곱추정량 $\hat{\beta}^{\mathsf{LSE}}$는 BLUE(Best Linear Unbiased Estimator)이다. 즉 $C \in \mathbb{R}^{(k+1)\times n}$에 의해 만들어지는 모든 선형 비편향 추정량 $\hat{\beta} = Cy$에 대하여(즉 $\mathbb{E}[\hat{\beta}] = CX\beta = \beta$이므로 $CX = I_{k+1}$일 때),

$$\mathrm{Var}(\hat{\beta}^{\mathsf{LSE}}) \preceq \mathrm{Var}(\hat{\beta})$$

가 성립한다. 이때 $A \preceq B$는 $A - B$가 negative definite matrix임을 의미한다.

분산이 작다는 것은 곧 추정량을 안정적으로 얻을 수 있음을 의미합니다. 만약 두 불편추정량이 있을 때 한 쪽의 분산이 작을 경우, 분산이 작은 추정량을 우리는 **효율적**(efficient)이라고 부릅니다. 즉 선형 연산을 통해 만들 수 있는 불편추정량 중 가장 좋은 것이 최소제곱추정량 $\hat{\beta}^{\mathsf{LSE}}$이므로, 우리는 이를 가장 많이 사용합니다.

다중회귀분석을 사용하는 이유

"All models are wrong, some are useful." (George Box)

다중회귀분석의 가정을 완벽히 만족하는 실제 데이터는 존재하지 않습니다. 그럼에도...

- 많은 요소가 상호작용하는 현상을 간단히 모형화하기 좋고
- **실험** 상황에서 실험자가 임의로 X를 세팅함으로써 그것이 y에 미치는 **인과**적인 영향을 확인할 수 있으며(내생성이 확실히 부재하므로)
- 두 변수 x와 y 사이의 **상관**관계를 확인하고, 통계적으로 얼마나 유의한지 확인할 수 있기도 하고
- 최소제곱추정량은 BLUE이기에 효율적인 데다가
- 많은 사람들이 알고 있기 때문에 다른 사람들에게 설명하기 용이합니다.

Multiple Linear Regression in Python

```
1      import pandas as pd
2      import numpy as np
3      import
4      import statsmodels.api as sm statsmodels.formula.api as smf
5      import matplotlib.pyplot as plt
6      import seaborn as sns
7
8      data = pd.read_csv('./data/week1.csv', index_col= 0)
9
```

	경상수지_lag	상품수지_lag	서비스수지_lag	본원소득수지_lag	이전소득수지_lag	자본수지_lag	자본이전_lag	비생산비금융자산_lag	금융계정_lag	직접투자_lag	증권투자_lag	파생금융상품_lag	기타투자_lag	준비자산_lag	외국인보유비율변화
2018/01	4486.5	7915.5	-3707.4	1123.7	-845.3	-31.1	-14.9	-16.2	-6524.9	82.8	-12072.0	1451.9	4776.6	-764.2	0.357013
2018/02	2556.9	7338.0	-4654.4	1543.5	-1670.2	-6.1	2.3	-8.4	-4640.5	-885.9	-3516.5	1310.9	-104.9	-1444.1	0.060606
2018/03	3226.7	4793.3	-2723.5	1553.1	-396.2	-13.3	-6.6	-6.7	-3635.3	-253.1	-9867.3	888.1	5756.7	-159.7	-3.690714
2018/04	5210.4	9216.5	-2314.0	-1008.4	-683.7	1.7	5.7	-4.0	-5341.4	-1923.4	-399.2	1497.1	-3328.1	-1187.8	-0.243810
2018/05	1490.4	9402.2	-1970.1	-5269.2	-672.5	47.7	-0.5	48.2	-37.2	-2189.0	-4070.4	749.3	8591.9	-3119.0	0.329690
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2023/08	4113.9	4427.5	-2572.1	3356.3	-1097.8	-32.1	-0.2	-31.9	-3677.0	-1036.9	-4274.2	293.6	-125.7	1466.2	0.638420
2023/09	5412.7	5201.4	-1549.5	1879.0	-118.2	-26.5	3.0	-29.5	-6286.7	-2112.9	-4246.9	328.8	-1847.5	1591.8	0.251555
2023/10	6072.7	7486.3	-3209.9	2180.4	-384.1	23.7	25.6	-1.9	-4370.2	-2071.5	-5131.1	-665.3	2262.2	1235.5	-0.162105
2023/11	7437.8	5433.3	-1279.8	3358.5	-74.2	44.0	45.9	-1.9	-8773.7	-171.3	-4400.9	-430.0	-3812.2	40.7	-0.730813
2023/12	3890.7	6878.2	-2210.9	-116.6	-660.0	-11.5	-7.2	-4.3	-1128.1	-2888.1	2203.0	229.6	-54.9	-617.7	-2.642344

Multiple Linear Regression in Python

단순회귀분석에서와 동일하게, `statsmodels.formula.api` 모듈을 `smf`로 불러와 `sm.ols().fit()`을 이용하면 모형과 적합을 수행해줄 수 있습니다. 한편, 해당 결과 `ols_result`에서 다양한 값에 접근하는 방법들은 Week 1 Slide를 참고해 주세요.

```
1    ols_result = smf.ols(formula = '외국인보유비율변화 ~ 상품수지_lag +
     서비스수지_lag + 증권투자_lag', data = data).fit()
2    print(ols_result.summary())
3
```

이때 우리가 사용한 모형은 반응변수를 y_i =외국인보유비율변화, 설명변수 세 개를 각각 x_{i1} =전월상품수지, x_{i2} =전월서비스수지, x_{i3} =전월증권투자로 하는 다중선형회귀모형

$$\mathbb{E}[y_i|X_i] = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3}$$

입니다. ($i = 1, 2, \cdots, n$이며, 우리의 경우 $n = 72$)

Multiple Linear Regression in Python

```
                            OLS Regression Results
==============================================================================
Dep. Variable:              외국인보유비율변화   R-squared:                       0.009
Model:                            OLS   Adj. R-squared:                 -0.034
Method:                 Least Squares   F-statistic:                    0.2116
Date:                Thu, 21 Mar 2024   Prob (F-statistic):              0.888
Time:                        18:09:49   Log-Likelihood:                -70.471
No. Observations:                  72   AIC:                             148.9
Df Residuals:                      68   BIC:                             158.0
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      0.0279      0.168      0.166      0.869      -0.308       0.364
상품수지_lag   -1.561e-05   2.12e-05     -0.736      0.464   -5.79e-05    2.67e-05
서비스수지_lag   1.594e-05   7.31e-05      0.218      0.828      -0.000       0.000
증권투자_lag   -1.912e-06   1.79e-05     -0.107      0.915   -3.76e-05    3.38e-05
==============================================================================
Omnibus:                       76.989   Durbin-Watson:                   1.411
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              759.961
Skew:                          -3.136   Prob(JB):                    9.47e-166
Kurtosis:                      17.628   Cond. No.                     1.66e+04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.66e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Interpretation of Multiple Linear Regression

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik} + \epsilon_i$$

β_1의 의미는 무엇일까?
$\Rightarrow$ 다른 변수들 $x_{i2}, \cdots, x_{ik}$가 **고정되는** 경우, x_{i1}가 1단위 증가할 때 상승하는 y_i의 기댓값
$\Rightarrow$ 만약 다른 변수들 $x_{i2}, \cdots, x_{ik}$이 **동일한** 두 집단이 있을 때, x_{i1}이 1단위 만큼 더 큰 집단의 y_i는 다른 집단의 y_i에 비하여 평균적으로 β_1만큼 크다.

Note: 이를 'x_{i1}를 1만큼 상승시키면, y_i도 β_1만큼 상승한다' 인과적으로 해석해서는 안 됩니다. 왜냐하면 다른 설명변수들을 통제하지 않았기에, x_{i1}의 상승이 x_{i2}를 비롯한 다른 변수들에 영향을 줄 수 있기 때문입니다.

Interpretation of Multiple Linear Regression

간단한 예시를 확인하여 봅시다. 예를 들어, 어떠한 배구선수 i의 공격성공률 y_i는 아래의 과정을 거쳐 결정됩니다.

$$y_i = 0.1 + 0.05 \times (\text{키})_i + 0.6 \times (\text{점프력})_i + 0.05 \times (\text{힘})_i + \epsilon_i$$

이때 사실 $(\text{힘})_i = 999.9 + 0.1i - 0.2(\text{점프력})_i^2$이라고 합니다. 즉 근육이 많은 선수는 힘이 좋지만 이 때문에 점프력은 낮아져 둘 사이에 음의 상관관계가 존재합니다. 그렇다면 우리가 키, 점프력, 힘을 수집해서 회귀분석을 돌릴 경우, 아마 회귀계수는

$$(\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2, \hat{\beta}_3)^T \approx (0.1, 0.05, 0.6, 0.05)^T$$

처럼 나올 것이라 기대할 수 있습니다. 그럼 우리가 웨이트트레이닝을 통해 힘을 1만큼 신장시키면 공격성공률이 **인과적으로** 0.05만큼 증가할까요?

Interpretation of Multiple Linear Regression

아닙니다. 예를 들어 배구선수 S씨($i = 1$)는 키가 200cm, 점프력이 50, 힘이 500이었습니다. 이때 y_i는

$$\mathbb{E}[y_i|X_i] = 0.1 + 0.05 \times 200 + 0.6 \times 50 + 0.05 \times 500 = 65.1$$

퍼센트입니다. 이때 배구선수 S씨가 웨이트를 통해 힘을 501로 늘렸습니다. 이에 따라 점프력은 49.95로 감소합니다. 따라서

$$\mathbb{E}[y_i|X_i'] = 0.1 + 0.05 \times 200 + 0.6 \times 49.95 + 0.05 \times 501 = 65.12$$

로 증가량은 사실 0.02밖에 되지 않습니다. 이는 힘을 늘려서 증가하는 공격성공률 중 일부를 점프력의 하락에 의한 공격성공률 감소가 깎아먹었기 때문입니다.

Interpretation of Multiple Linear Regression

이에 따라 회귀계수는 인과적인 효과를 의미하는 것이 아니라, 상관적인 효과를 의미하게 됩니다. 적절한 비교는 배구선수 S씨($i = 1$)

$$\mathbb{E}[y_i|X_i] = 0.1 + 0.05 \times 200 + 0.6 \times 50 + 0.05 \times 500 = 65.1$$

와 키 200cm, 점프력이 50, 힘이 501인 배구선수 K씨($i = 11$)

$$\mathbb{E}[y_i|X_i] = 0.1 + 0.05 \times 200 + 0.6 \times 50 + 0.05 \times 501 = 65.15$$

를 비교하는 것입니다. 이 경우 힘의 증가에 따른 다른 변수들의 효과까지 모두 상관성에 녹아들어 있기 때문에, 올바른 비교입니다. 몇몇 조건들이 만족된다면 다중회귀분석이 인과성을 밝히는 데에도 사용될 수 있지만, 대부분의 경우에는 아닙니다.

Interpretation of Multiple Linear Regression

$$Y = X\beta + \epsilon$$

에서, 설명변수 $k+1$개 중 앞의 l개를 모은 $n \times l$ 행렬을 X_1, 뒤의 $k+1-l$개를 모은 $n \times (k+1-l)$ 행렬을 X_2라 하고, 회귀계수 $k+1$개도 동일한 방식으로 나눈 두 벡터를 β_1, β_2라고 할 때

$$y = X\beta + \epsilon = \begin{pmatrix} X_1 & X_2 \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} = X_1\beta_1 + X_2\beta_2 + \epsilon$$

처럼 쓸 수 있다. 그렇다면,

Frisch-Waugh-Lovell theorem(FWL theorem)

$M_{X_1} = I - X_1(X_1^T X_1)^{-1} X_1^T$에 대하여,

$$M_{X_1} y = M_{X_1} X_2 \beta_2 + M_{X_1} \epsilon$$

을 회귀분석하여 얻은 추정량 $\hat{\beta}_2$와 OLS 추정량 $\hat{\beta}_2^{\mathsf{LSE}}$는 동일하다.

Interpretation of Multiple Linear Regression

단순회귀분석에서 알아보았듯, 축소모형

$$y = X_1\beta_1 + \epsilon'$$

의 잔차 y'는 y 중 설명변수 X_1에 의하여 설명되지 않는 부분을 의미하며, 앞서 우리가 확인하였듯 $y' = y - \hat{y} = (I - H_1)y = M_{X_1}y$입니다.
동일한 이유로, X_2 중에서 X_1에 의해 설명되지 않는 부분은 $M_{X_1}X_2$처럼 쓸 수 있습니다. 그렇다면 모형

$$M_{X_1}y = M_{X_1}X_2\beta_2 + M_{X_1}\epsilon$$

을 적합하겠다는 것은 곧 **반응변수 중 X_1가 설명하지 못하는 부분**이 X_2 **중 X_1가 설명하지 못하는 부분**과 어떠한 관계를 가지는지 보겠다는 것과 동일합니다. 즉 X_2의 계수 부분인 β_2는 y 중 X_1**의 영향을 전부 배제한 채** X_2가 오롯이 설명할 수 있는 정도를 의미하게 됩니다. 이에 따라 X_2가 변화하며 X_1에도 변화가 생기고, 이에 따라 간접적으로 y가 바뀌는 효과의 전파를 파악할 수 없는 것입니다.

Interpretation of Multiple Linear Regression

이제 이를 염두에 두고 인과추론을 위해서는 다른 모형이 필요함을 인식한 채, 회귀모형을 적합한 결과를 해석하여 봅시다. 인과모형들도 회귀분석에 기초를 두는 만큼, 회귀분석의 해석법을 이해하는 것은 매우 중요합니다.

일반적으로, 회귀분석 결과를 리포트하라면 이 표를 이용합니다.

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	0.0279	0.168	0.166	0.869	-0.308	0.364
상품수지_lag	-1.561e-05	2.12e-05	-0.736	0.464	-5.79e-05	2.67e-05
서비스수지_lag	1.594e-05	7.31e-05	0.218	0.828	-0.000	0.000
증권투자_lag	-1.912e-06	1.79e-05	-0.107	0.915	-3.76e-05	3.38e-05

여기에서 `coef`가 상응하는 추정량 $\hat{\beta}$입니다.

Interpretation of Multiple Linear Regression

오차가 정규분포를 따른다면,

$$\begin{aligned}
\mathrm{Var}(\hat{\beta}) &= \sigma^2 (X^T X)^{-1} \\
\hat{\beta} &\sim \mathcal{N}(\beta, \sigma^2 (X^T X)^{-1}) \\
\hat{\beta}_j &\sim \mathcal{N}(\beta_j, \sigma^2 (X^T X)^{-1}_{j+1,j+1}) \\
\hat{\sigma}^2 &= \frac{(y - X\beta)^T (y - X\beta)}{n - k - 1} \\
\frac{(n-k-1)\hat{\sigma}^2}{\sigma^2} &\sim \chi^2(n-k-1)
\end{aligned}$$

임이 알려져 있습니다. 따라서 $\hat{\mathrm{se}}(\hat{\beta}_j) = \sqrt{\hat{\sigma}^2 (X^T X)^{-1}_{j+1,j+1}}$가 `std err`로 리턴되며, `t`는 $H_0 : \beta_j = 0$의 검정을 위한 검정통계량

$$T_j = \frac{\hat{\beta}_j - \beta_j}{\hat{\mathrm{se}}(\hat{\beta}_j)} \sim t(n-k-1)$$

의 관측값 T_j^{obs}을 의미하게 됩니다.

Interpretation of Multiple Linear Regression

따라서 일반적으로

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	0.0279	0.168	0.166	0.869	-0.308	0.364
상품수지_lag	-1.561e-05	2.12e-05	-0.736	0.464	-5.79e-05	2.67e-05
서비스수지_lag	1.594e-05	7.31e-05	0.218	0.828	-0.000	0.000
증권투자_lag	-1.912e-06	1.79e-05	-0.107	0.915	-3.76e-05	3.38e-05

위를 리턴하되, P>|t|가 미리 정한 유의수준(주로 0.05)보다 작으면 해당 계수에 상응하는 설명변수가 반응변수와 갖는 상관관계가 유의하다고 해석합니다. 반면 그 값이 유의수준보다 크거나, 신뢰구간 [0.025 0.975]가 0을 포함한다면 그렇지 않다고 해석합니다. 신뢰구간은 아래와 같습니다.

$$\left(\hat{\beta}_j - t_{\alpha/2}(n-k-1) \times \hat{\text{se}}(\hat{\beta}_j), \hat{\beta}_j + t_{\alpha/2}(n-k-1) \times \hat{\text{se}}(\hat{\beta}_j)\right)$$

Test for Significance of Regression

한편 각각의 계수만이 아니라, 모형 자체의 유의성에 대해 확인해볼 수도 있습니다. 이 경우의 귀무가설은

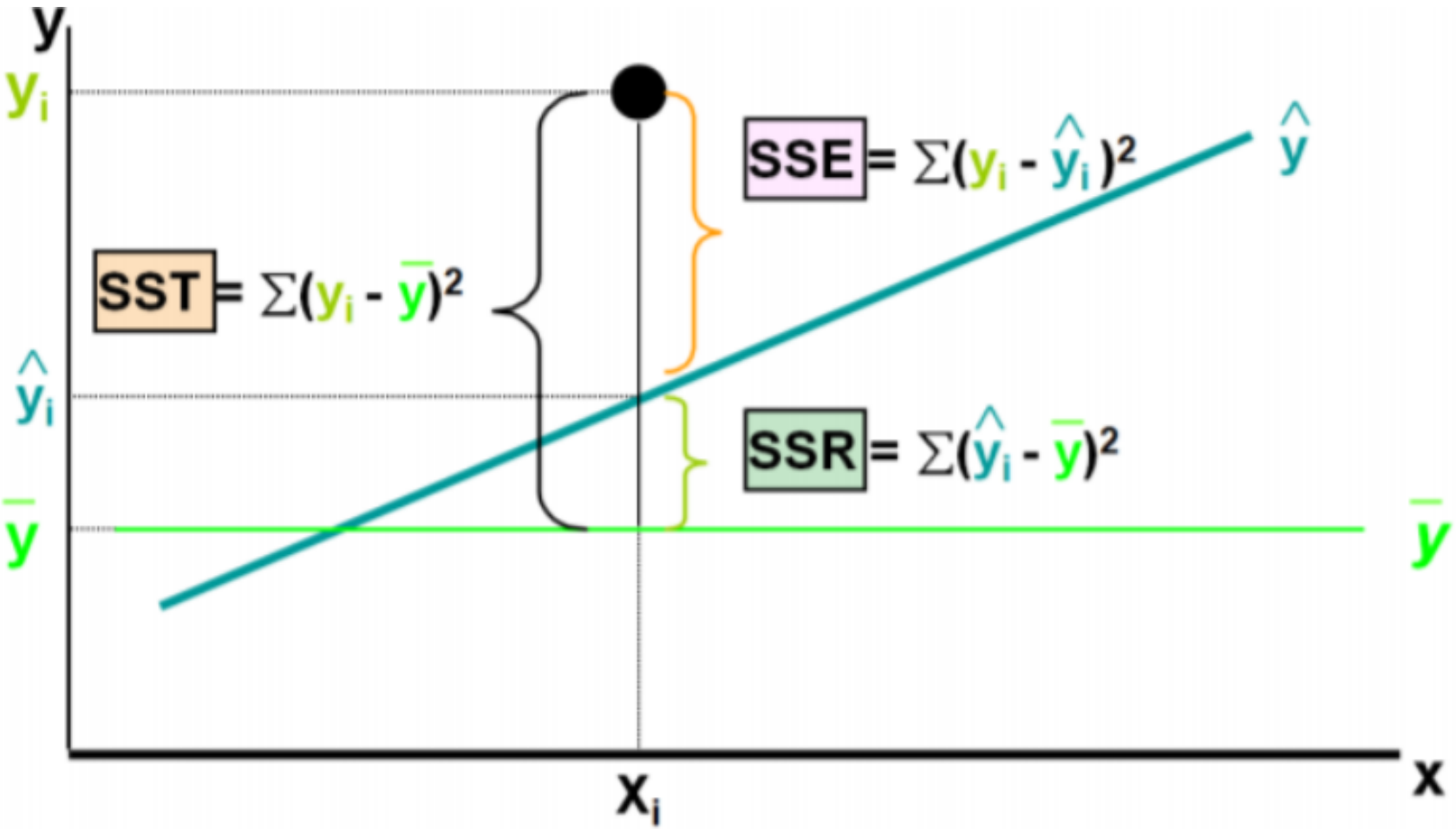
$$H_0 : \beta_1 = \cdots = \beta_k$$

입니다. 반면 대립가설은

$$H_1 : \beta_j \neq 0 \quad \text{for at least one } j$$

으로 주어집니다. 우리는 이 검정에 단순회귀분석에서의 SST, SSR, SSE를 이용하려 합니다.

Test for Significance of Regression



Test for Significance of Regression

- SST: 반응변수 y_i의 총 변동

$$SST = \sum_{i=1}^{n}(y_i - \bar{y})^2 = y^T \left(I_n - \frac{1}{n} 1 1^T \right) y \sim \sigma^2 \cdot \chi^2(n-1)$$

- SSR: y_i의 변동 중 **R**egression으로 설명 가능한 변동

$$SSR = \sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2 = y^T \left(H - \frac{1}{n} 1 1^T \right) y \sim \sigma^2 \cdot \chi^2(k)$$

- SSE: y_i의 변동 중 **E**rror로 설명 가능한 변동

$$SSE = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = y^T (I_n - H) y \sim \sigma^2 \cdot \chi^2(n-k-1)$$

Test for Significance of Regression

Some Facts

- $SST = SSR + SSE$
- $SSR \perp\!\!\!\perp SSE$
- $\hat{\sigma}^2 = MSE = \dfrac{SSE}{n-k-1}$
- $F = \dfrac{SSR/k}{SSE/(n-k-1)} \sim F(k, n-k-1)$

만약 F가 크다면, 이는 SSR이 SSE에 비하여 유의미하게 큼을 의미합니다. 따라서 전체 변동 중 회귀분석에 의해 설명할 수 있는 변동이 많으므로, 회귀모형이 유의하다고 판단합니다. 반면 F가 작다면 모형의 설명력이 부족하다고 판단합니다. 따라서 모형 유의성에 대한 판단은

$$F > F_\alpha(k, n-k-1)$$

일 경우 H_0을 기각함으로써 수행합니다.

Test for Significance of Regression

OLS Regression Results

Dep. Variable:	외국인보유비율변화	R-squared:	0.009
Model:	OLS	Adj. R-squared:	-0.034
Method:	Least Squares	F-statistic:	0.2116
Date:	Thu, 21 Mar 2024	Prob (F-statistic):	0.888
Time:	18:09:49	Log-Likelihood:	-70.471
No. Observations:	72	AIC:	148.9
Df Residuals:	68	BIC:	158.0
Df Model:	3		
Covariance Type:	nonrobust		

여기에서 `F-statistic`과 `Prob (F-statistic)`이 F통계량과 그에 상응하는 p값입니다. 또한, `No. Observations`와 `Df Residuals`, `Df Model` 역시 확인할 수 있습니다. 현재 변수가 $k = 3$개이므로, SSE의 자유도는 $72 - 3 - 1 = 68$입니다. SSR의 자유도는 $k = 3$이 됩니다.

R^2 and adjusted R^2

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

으로, 전체 변동 중 회귀분석으로 설명되는 변동의 비율을 의미합니다. 한편

$$R^2 = \frac{SSR}{SSR + SSE} = \frac{\frac{SSR}{SSE}}{1 + \frac{SSR}{SSE}} = \frac{\frac{n-k-1}{k}F}{1 + \frac{n-k-1}{k}F} \sim \text{Beta}\left(\frac{k}{2}, \frac{n-k-1}{2}\right)$$

이므로,

$$\mathbb{E}[R^2] = \frac{\frac{k}{2}}{\frac{k}{2} + \frac{n-k-1}{2}} = \frac{k}{n-1}$$

입니다. 따라서 변수를 많이 넣으면, 귀무가설이 맞는 경우에도 R^2가 매우 높게 나오는 상황이 연출될 수 있습니다.

R^2 and adjusted R^2

이러한 상황을 피하기 위하여,

$$R^2_{\text{adj}} = 1 - (1 - R^2)\frac{n-1}{n-k-1}$$

으로 정의하여 $\mathbb{E}[R^2_{\text{adj}}] = 0$이 되도록 **조정된 결정계수**(adjusted R sqaured)를 정의합니다. 이는 변수의 개수가 증가함에 따라 설명력이 증가하는 것처럼 보이는 현상을 방지하고, 모형 자체의 적합성과 선형성을 강조하는 데 유용합니다.

OLS Regression Results

Dep. Variable:	외국인보유비율변화	R-squared:	0.009
Model:	OLS	Adj. R-squared:	-0.034
Method:	Least Squares	F-statistic:	0.2116
Date:	Thu, 21 Mar 2024	Prob (F-statistic):	0.888
Time:	18:09:49	Log-Likelihood:	-70.471
No. Observations:	72	AIC:	148.9
Df Residuals:	68	BIC:	158.0
Df Model:	3		
Covariance Type:	nonrobust		

Testing Complicated Hypothesis

Case 1. 어떠한 계수가 다른 계수에 비해 크다
예를 들어, 어떠한 변수 x_1의 영향이 다른 변수 x_2의 영향보다 크다고 이야기하려면

$$H_0 : \beta_1 \leq \beta_2, \quad H_1 : \beta_1 > \beta_2$$

를 검정해야 합니다. 이때 우리는 H_0을 $H_0 : \beta_1 - \beta_2 \leq 0$으로 바꾸어 검정할 수 있습니다. 그러면 적당한 행벡터 C에 대하여

$$\beta_1 - \beta_2 = (0, 1, -1, 0, \cdots, 0) \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_k \end{pmatrix} = C\beta$$

으로 쓸 수 있습니다.

Testing Complicated Hypothesis

이때 우리가

$$\hat{\beta} \sim \mathcal{N}(\beta, \sigma^2(X^T X)^{-1})$$

임을 알고 있으므로,

$$\hat{\beta}_1 - \hat{\beta}_2 = C\hat{\beta} \sim \mathcal{N}(C\beta, \sigma^2 C(X^T X)^{-1} C^T)$$

입니다. 따라서 잘 정리하면

$$T_C = \frac{(\hat{\beta}_1 - \hat{\beta}_2) - (\beta_1 - \beta_2)}{\sqrt{\hat{\sigma}^2 C(X^T X)^{-1} C^T}} \sim t(n-k-1)$$

임을 알 수 있습니다. 그러므로 만약

$$T_C^{\text{obs}} > t_\alpha(n-k-1)$$

이라면 귀무가설을 기각합니다.

Testing Complicated Hypothesis

대부분의 패키지에서 이 정도의 복잡한 가설을 검정하려면 직접 코딩을 하셔야 합니다. 그 과정은 아래와 같습니다.

```
    from scipy import stats

    covmat = ols_result.cov_params() # covariance matrix
    C = np.zeros(4); C[1] = 1; C[2] = -1 # make C
    Tcse = np.matmul(np.matmul(C, covmat), C.transpose()) #
    variance
    betadiff = np.matmul(C, ols_result.params) # \hat{\beta_1} - \
    hat{\beta_2}
    TC = betadiff / np.sqrt(Tcse) # t statistic
    1 - stats.t.cdf(df = 68, x = TC) # p-value

```

우리의 경우, p값으로 0.6635가 나와 귀무가설을 기각할 수 없습니다.

Testing Complicated Hypothesis

Case 2. 복잡한 형태의 선형 가설 검정
Case 1을 확장하면, 특정한 행렬 $C \in \mathbb{R}^{q \times (k+1)}$에 대하여

$$H_0 : C\beta = 0$$

을 검정하고 싶을 수 있습니다. 이때 주의할 것은 $C\beta$는 벡터일 수 있습니다. 우리는 이러한 C를 **contrast matrix**라고 주로 부릅니다. 그렇다면

$$F_C = \frac{(C\hat{\beta})^T [\hat{\sigma}^2 C(X^T X)^{-1} C^T]^{-1} (C\hat{\beta})}{q} \sim F(q, n-k-1)$$

임이 알려져 있으며, 만약 이 값이 크다면 귀무가설을 기각할 수 있습니다.

Testing Complicated Hypothesis

contrast matrix C가 결부된 F검정은 `.f_test(C)` 함수를 이용할 수 있습니다.

```
1       C = np.array([[0, 1, -1, 0], [0, 0, 0, 1]])
2       print(ols_result.f_test(C))
3
```

여기에서 F통계량은 0.09가 되며, 분모의 자유도는 68, 분자의 자유도는 2이고, 상응하는 p값은 0.91이게 됩니다.

```
<F test: F=0.0903177281595697, p=0.9137502630832091, df_denom=68, df_num=2>
```

Testing Complicated Hypothesis

Case 3. 특정 계수만 검정
Case 2의 조금 특정한 케이스이긴 하지만,

$$H_0 : \beta_1 = \beta_2 = 0$$

처럼 몇몇 계수가 0인지를 검정해야 할 때가 있습니다. 이 경우 귀무가설을 적용한 모형을 **Reduced Model**(RM), 모든 변수를 넣은 모형을 **Full Model**(FM)으로 하여 둘 모두를 적합합니다.

$$\text{RM} : y_i = \beta_3 x_{i3} + \epsilon_i$$
$$\text{FM} : y_i = \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \epsilon_i$$

그렇다면 귀무가설에 의하여 생성되는 오차는

$$SSH = SSE(RM) - SSE(FM)$$

이며 그 자유도는 H_0에서 가하는 **제약**(restriction)의 개수 $r = 2$와 같습니다.

Testing Complicated Hypothesis

그렇다면 full model에서 귀무가설을 가함에 따라 생기는 제약의 상대적 크기를

$$F_R = \frac{SSH/r}{SSE(FM)/(n-k-1)} \sim F(r, n-k-1)$$

을 통해 검정할 수 있습니다. 이는 사실 Case 2에서 C를 적절히 설정하여 Case 3의 가설을 검정할 수 있도록 설계하였을 때의 F통계량 F_C와도 동일합니다.

Testing Complicated Hypothesis

.f_test()는 행렬만이 아니라 문자열 형태로 hypothesis를 늘어놓아도 검정의 수행이 가능합니다.

```
1       print(ols_result.f_test('(상품수지_lag = 0), (
        서비스수지_lag = 0)'))
2
```

```
<F test: F=0.31025501064939465, p=0.734292372322748, df_denom=68, df_num=2>
```

여기에서 $F_R = 0.31$이며, 이에 상응하는 자유도와 p값 역시 구할 수 있습니다.

도움되는 링크들

- OLS 결과에서 접근 가능한 밸류들: `https://www.statsmodels.org/dev/generated/statsmodels.regression.linear_model.RegressionResults.html#statsmodels.regression.linear_model.RegressionResults`
- prediction과 visualization: `https://www.statsmodels.org/stable/examples/notebooks/generated/predict.html`
- 일반적으로, 이들 값 중 어떤 밸류를 리턴할지는 선행연구에서 준 테이블에 기반하여 결정합니다. 다만 통계학자들은 그냥 저 테이블 자체를 전부 리포트하는 것을 선호하시는 분들도 있습니다.

Regression Diagnostics

Residual Analysis

만약 다중선형회귀의 적합이 잘 되었다면, 오차 ϵ_i의 추정량 e_i은 ϵ_i의 성질을 일부 따라야 합니다. 대표적인 성질에는

$$\bar{e} = \frac{1}{n} 1^T e = 0$$

이 있습니다. 단, ϵ_i들과는 다르게 e_i는 X들에 의존하기에, 서로 독립적이지는 않습니다. 한편 e_i는 σ^2에도 의존하므로, 우리는 아래처럼 **잔차**(reisidual) e_i를 스케일링할 수 있습니다.

$$d_i = \frac{e_i}{\sqrt{MSE}}, \quad r_i = \frac{e_i}{\sqrt{MSE(1 - H_{ii})}}$$

이때 d_i는 **standardized residual**, r_i는 **studentized residual**이라 불립니다.

Residual Analysis

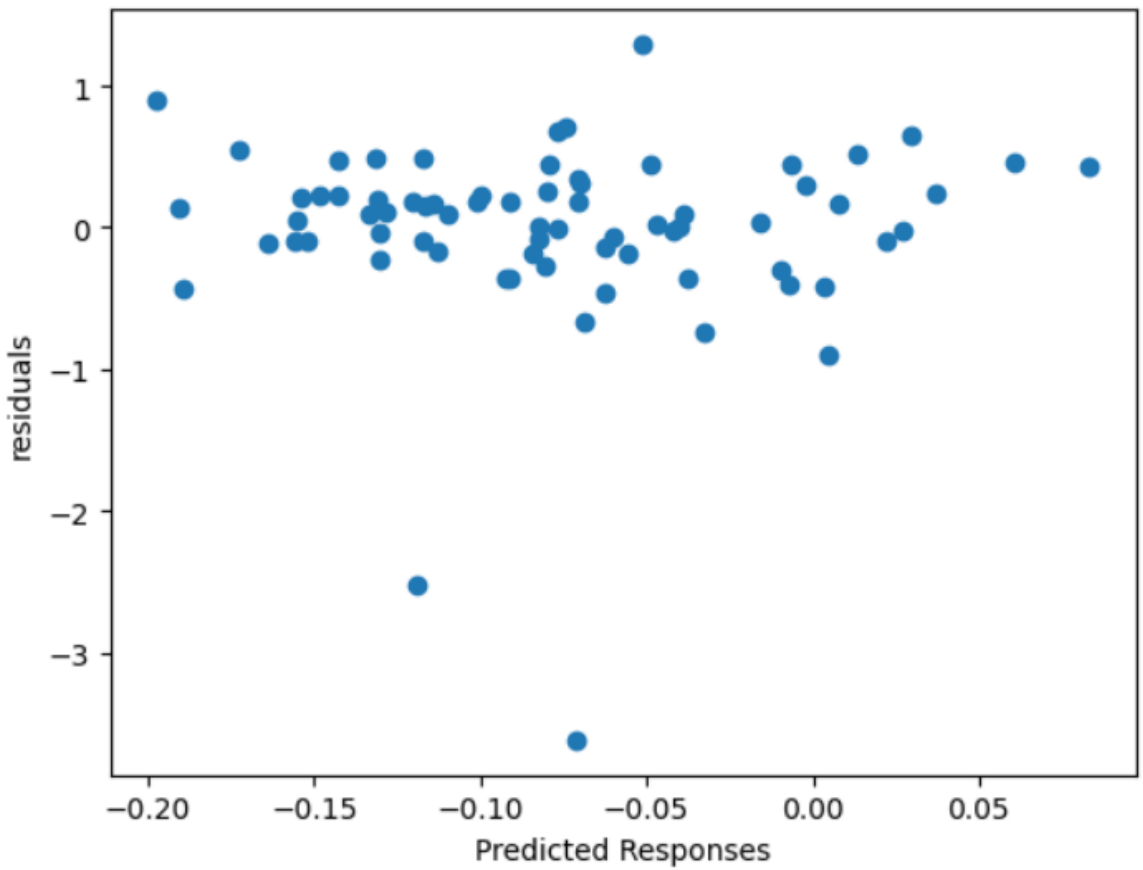
선형성(linearity) 가정의 확인을 위해 가장 좋은 방법 중 하나는 $\hat{y}$ vs. e의 그래프를 그리는 것입니다. 만약 선형성이 있다면, 둘은 무관해야 합니다. 즉 0을 주변으로 특정한 패턴 없이 균등하게 분포할 것을 기대합니다.

```
    residuals = ols_result.resid
    fittedy = ols_result.predict()
    plt.scatter(fittedy, residuals)
    plt.xlabel('Predicted Responses')
    plt.ylabel('residuals')
    plt.show()

```

Note: 이 경우 표준화나 스튜던트화는 수행하지 않아도 괜찮습니다.

Residual Analysis: Linearity



전반적으로 잔차와 예측값 사이에는 문제가 없는 것을 볼 수 있습니다.

Residual Analysis: Linearity

선형성을 시각적으로 확인하는 방법이 맘에 드시지 않을 수 있습니다. 이 경우 **Lack of fit**(Lof) 검정을 이용할 수 있습니다. 이 경우 가설은

$$H_0 : \mathbb{E}[y_i|X_i] = X_i\beta$$

가 됩니다. 다만 이는 선형회귀의 다른 조건들을 필요로 하며, 한 X_i에 대하여 반복 측정을 하여 y_i가 여러 개 있다고 가정해야 합니다. 따라서 X_i가 연속형 변수이면서 관찰연구인 경우에는 사용하기 어렵다는 치명적 단점이 있긴 합니다. 반대로 순서가 있는 이산형 변수 혹은 실험연구인 경우에는 유용하게 사용할 수 있습니다. 다만 우리는 잘 사용하지 않을 예정이라, 링크만 첨부합니다.

Link: `https://online.stat.psu.edu/stat462/node/113/`

Residual Analysis: Linearity

이외에도 다양한 사람들이 제작한 linearity 판단 검정들이 있습니다. 단, Lof 검정 정도는 수리적으로 이해하기 쉬운 반면, 이런 검정들은 조금 고수준의 테크닉이 필요해서 간단하게 언급만 하고 지나갑니다.

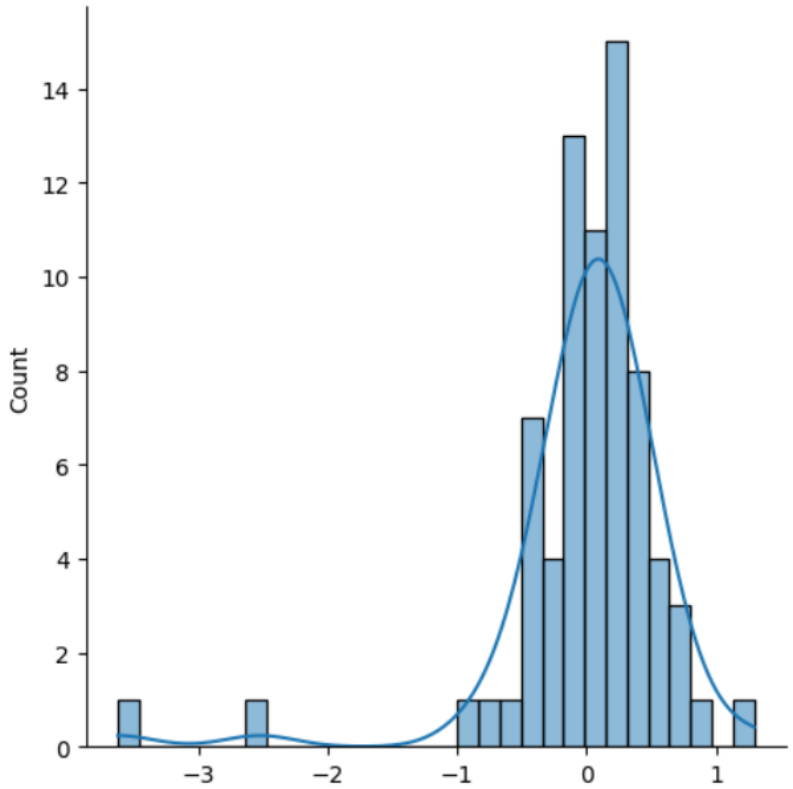
- **Harvey-Collier test**: `linear_harvey_collier()`를 통해 진행하며, p값이 0.05보다 작은 경우 선형성 가정을 기각합니다.
- **Rainbow test**: `linear_rainbow()`를 통해 진행하며, 마찬가지로 p 값이 0.05보다 작은 경우 기각합니다.
- **Ransey's RESET test**: `linear_reset()` 함수를 이용합니다.
- **Lagrange Multiplier test**: `linear_lm()` 함수를 이용합니다.

Residual Analysis

한편 앞서 그린 plot에서 우리는 등분산 가정(homoskedasticity) 역시 확인해 줄 수 있습니다. 몇몇 특이한 점들을 제외하면 분산은 거의 일정하게 유지되는 듯한 모습을 보입니다. 이에 대해서는 추후 조금 더 자세하게 확인합니다. 정규성 가정(normality) 역시 대략적으로 확인 가능합니다. 정규성 가정이 만족된다면, e_i들의 분포는 각 X_i들에서 정규분포를 따라야 합니다. 만약 특정한 패턴이 보이지 않는다면, 정규성 가정에 큰 위배 사항이 없다고 볼 수 있습니다. 한편 이를 시각적으로 조금 더 상세히 확인하는 방법 중 하나는 **Q-Q plot**을 그려보는 것입니다. 이는 정규분포의 누적분포함수와 e_i의 누적분포함수를 시각적으로 비교합니다.

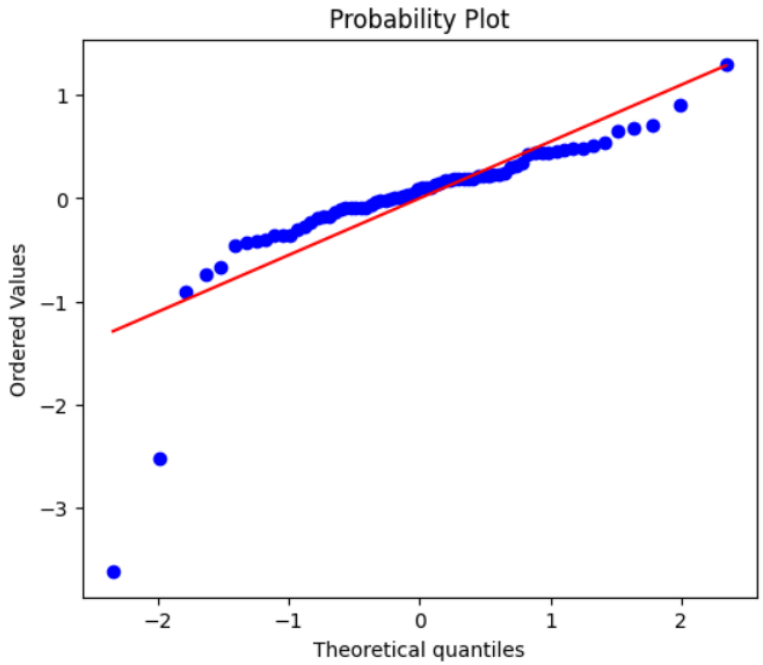
Residual Analysis: Normality

```
1     sns.displot(x = ols_result.resid, bins = 30, kde = True)
2
```



Residual Analysis: Normality

```
1       plt.figure(figsize = (6, 5))
2       stats.probplot(ols_result.resid, dist = stats.norm, plot = plt)
3       plt.show()
4
```



Residual Analysis: Normality

분포함수가 정규분포와 유사한지, 그리고 Q-Q plot에서 붉은 선과 푸른 점들의 그래프와 얼마나 유사한지 보는 것 역시 좋으나, 이를 수리적으로 엄밀하게 검정하고자 할 수 있습니다.
그 결과는 아래의 표에 있는 Omnibus와 Jarque-Bera (JB)에 나와 있습니다.

Omnibus:	76.989	Durbin-Watson:	1.411
Prob(Omnibus):	0.000	Jarque-Bera (JB):	759.961
Skew:	-3.136	Prob(JB):	9.47e-166
Kurtosis:	17.628	Cond. No.	1.66e+04

Residual Analysis: Normality

Omnibus test는 분포함수를 이용하여 그 분포가 정규분포와 얼마나 다른지 검정하는 방법입니다. Q-Q plot을 수식화하여 검정한다고 생각하면 됩니다. 이는 D'Agostino's K^2 test를 사용한다고 알려져 있습니다.
한편 Jarque-Bera test는 분포의 첨도와 왜도를 통하여 정규분포와 얼마나 다른지 검정합니다.

```
1     sm.stats.omni_normtest(ols_result.resid)
2     sm.stats.jarque_bera(ols_result.resid) # (value, pvalue, skew,
      kurtosis)
3
```

이들의 결과는 `summary()`에도 제공되며, 각각 다른 함수로 접근할 수도 있습니다.

Residual Analysis: Nomarlity

이외에도 Kolmogorov-Smirnov test, Lilliefors test, Anderson-Darling test 등을 이용할 수 있습니다. 이들 다섯 결과를 모두 리포트하는 것도 추천드리지만, 일반적으로는 자크-베라 검정을 많이 이용합니다.

```
1     sm.stats.diagnostic.kstest_normal(ols_result.resid) # (value,
      pvalue)
2     sm.stats.diagnostic.lilliefors(ols_result.resid) # (value,
      pvalue)
3     sm.stats.normal_ad(ols_result.resid) # (value, pvalue)
4
```

Jarque-Bera Test

$$JB = \frac{n}{6}\left(S^2 + \frac{1}{4}(K-3)^2\right)$$

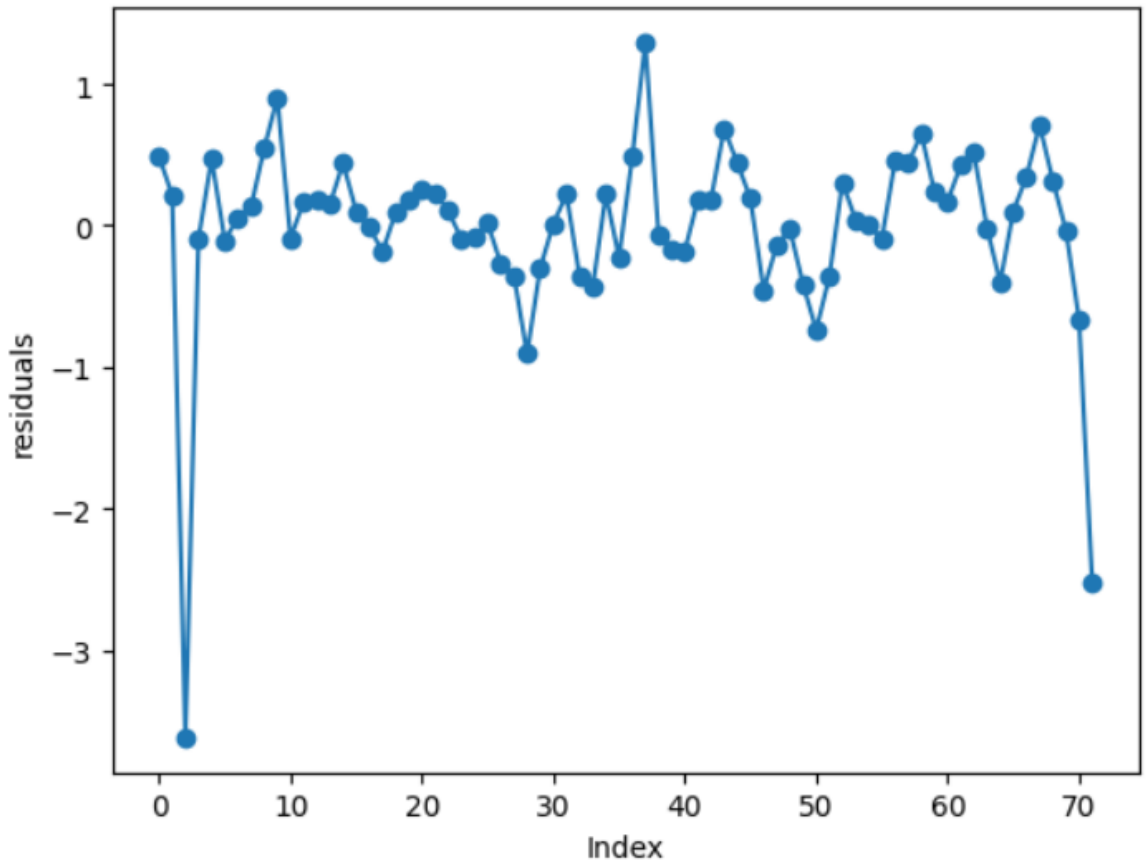
이때 n은 관측값의 개수, S, K는 표본왜도와 표본첨도

$$S = \frac{\frac{1}{n}\sum_{i=1}^{n} e_i^3}{\left(\frac{1}{n}\sum_{i=1}^{n} e_i^2\right)^{3/2}}$$
$$T = \frac{\frac{1}{n}\sum_{i=1}^{n} e_i^4}{\left(\frac{1}{n}\sum_{i=1}^{n} e_i^2\right)^{2}}$$

을 의미합니다 정규분포는 왜도를 0, 첨도를 3으로 가짐이 알려져 있으므로, 이 값이 크면 클수록 잔차의 분포가 정규분포와 상이한 왜도와 첨도를 가진다는 의미입니다. 이는 점근적으로 $\chi^2(2)$를 따름이 알려져 있습니다.

Residual Analysis: Independency

Independency를 보는 방법 중 하나는 시계열적 상관을 고려하는 것입니다. 저희 데이터처럼 패널 자료를 이용하는 경우, 연속한 두 시간 자료에 상관성이 있을 수 있습니다. 이때는 index에 따른 residual의 변화를 볼 수 있습니다.



Residual Analysis: Independency

```
1       sm.stats.durbin_watson(residuals)
2       # 1.4109115534738934
3
```

Durbin-Watson test의 검정통계량은

$$d = \frac{\sum_{t=2}^{n}(e_t - e_{t-1})^2}{\sum_{t=1}^{n} e_t^2}$$

이며, 이는 `summary()`에서도 리포트됩니다. 이 값이 2에 가깝다면 **자기상관**(autocorrelation)이 없다고 보고, 0에 가깝다면 양의 상관이, 4에 가깝다면 음의 상관이 있다고 봅니다. 이를 검정에 이용하려면, `https://real-statistics.com/statistics-tables/durbin-watson-table/`과 같은 table을 통하여 기각역을 확인하고 사용할 수 있습니다. 다만 일반적으로 그냥 d를 리포트해서 극단적인 값인지를 주로 보지, p값을 잘 리포트하지는 않습니다.

Residual Analysis: Independency

이외에도 시계열분석에서 주로 다루겠지만 residual가 어떠한 확률과정에 의해 생성되는 것이 아니라 IID 확률변수열임을 보이기 위하여

- Ljung-Box test: lag가 1만이 아니라 다른 lag에 대하여 시계열이 있는지에 대해 검정할 수 있습니다. `.acorr_ljungbox()`를 이용합니다.
- Breusch-Godfrey test: `.acorr_breusch_godfrey()`를 이용합니다.
- lm test: `.acorr_lm()`을 이용합니다.

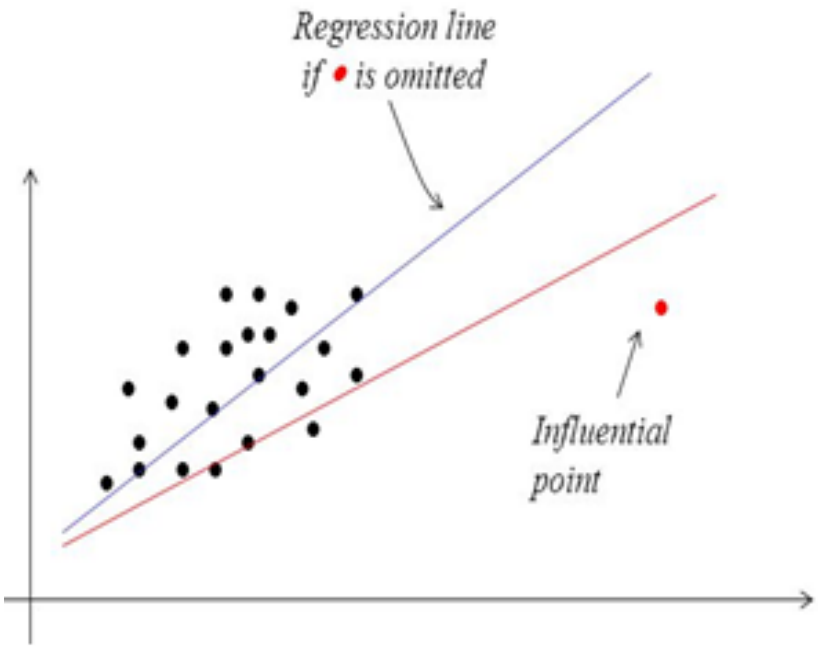
Note: 이들의 결과는 매우 상이할 수 있습니다. 전체를 다 리포트하는 게 좋으나, 모두에서 자기상관이 없다는 결과를 내는 데이터는 많지 않으므로 보통 Durbin-Watson test의 값만 리포트합니다.

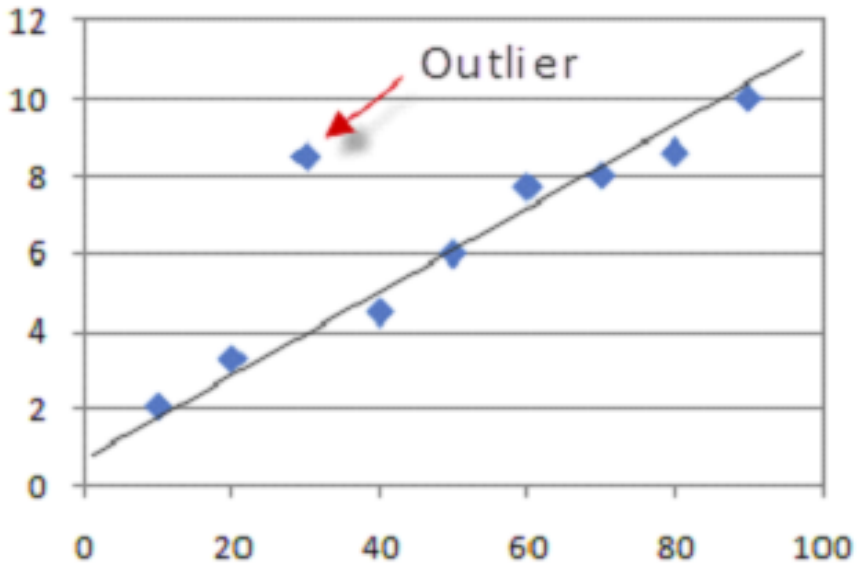
도움이 되는 링크들

- `statsmodels`의 다양한 diagnostics 함수들:
  `https://www.statsmodels.org/dev/diagnostic.html`
- 한편, 이러한 진단에서 어떠한 부분에 문제가 있을 수 있습니다. 그런 경우 모형을 수정하거나, 변환을 하든가 하여 이러한 문제들을 없앨 수 있도록 노력해야 합니다. linearity와 normality를 위해서는, 적절히 이론적 근거나 잔차그림 등을 계속 그려보면서 적절한 변환(로그변환, Box-Cox 변환 등)이나 변수를 찾고, 이것이 만족될 수 있도록 해야 합니다. autocorrelation이 있을 경우 시계열 모형들을 고려해볼 수 있습니다. 다만 완벽한 해결법은 없어서, 적절한 수준에서 타협하고 사용하거나, 충분한 숙고와 논의를 통하여 변환을 만들어내야 합니다.

Extreme Data Points

- **Outlier**(이상점): standardized residual d_i의 절대값이 2보다 커, 반응변수(y) 측면에서 이상점으로 취급가능한 점
- **Influential observation**(영향점): 해당 점의 유무에 따라 모형의 적합 결과가 매우 달라지는 관측값



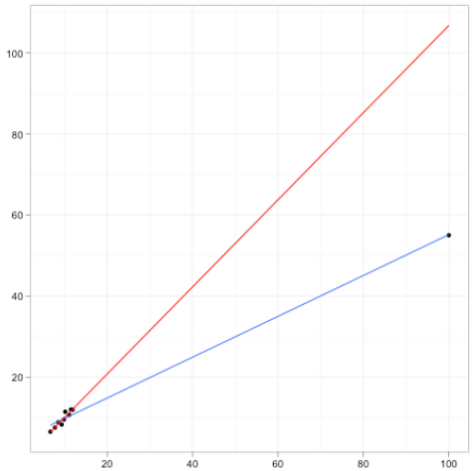


Leverage

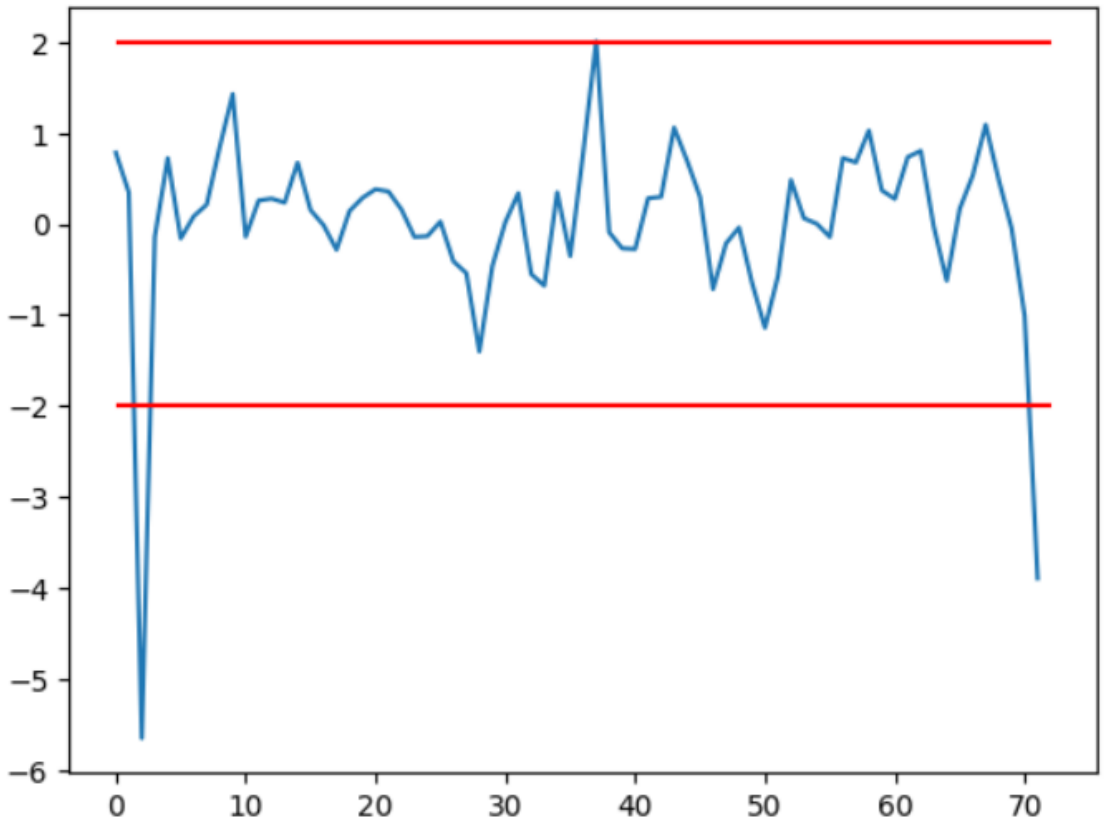
$H = X(X^T X)^{-1} X^T$의 i행 i열 원소를 H_{ii}라고 할 때, i번째 data의 **leverage**를 H_{ii}으로 정의합니다. 만약

$$H_{ii} > \frac{2(k+1)}{n}$$

이라면, 이를 **high leverage point**라고 부릅니다. 이들은 설명변수 x 측면에서 이상점이 됩니다. 일반적으로 회귀선은 이러한 high leverage point를 지나도록 만들어지며, 이에 따라 많은 경우 influential observation으로 작용합니다.



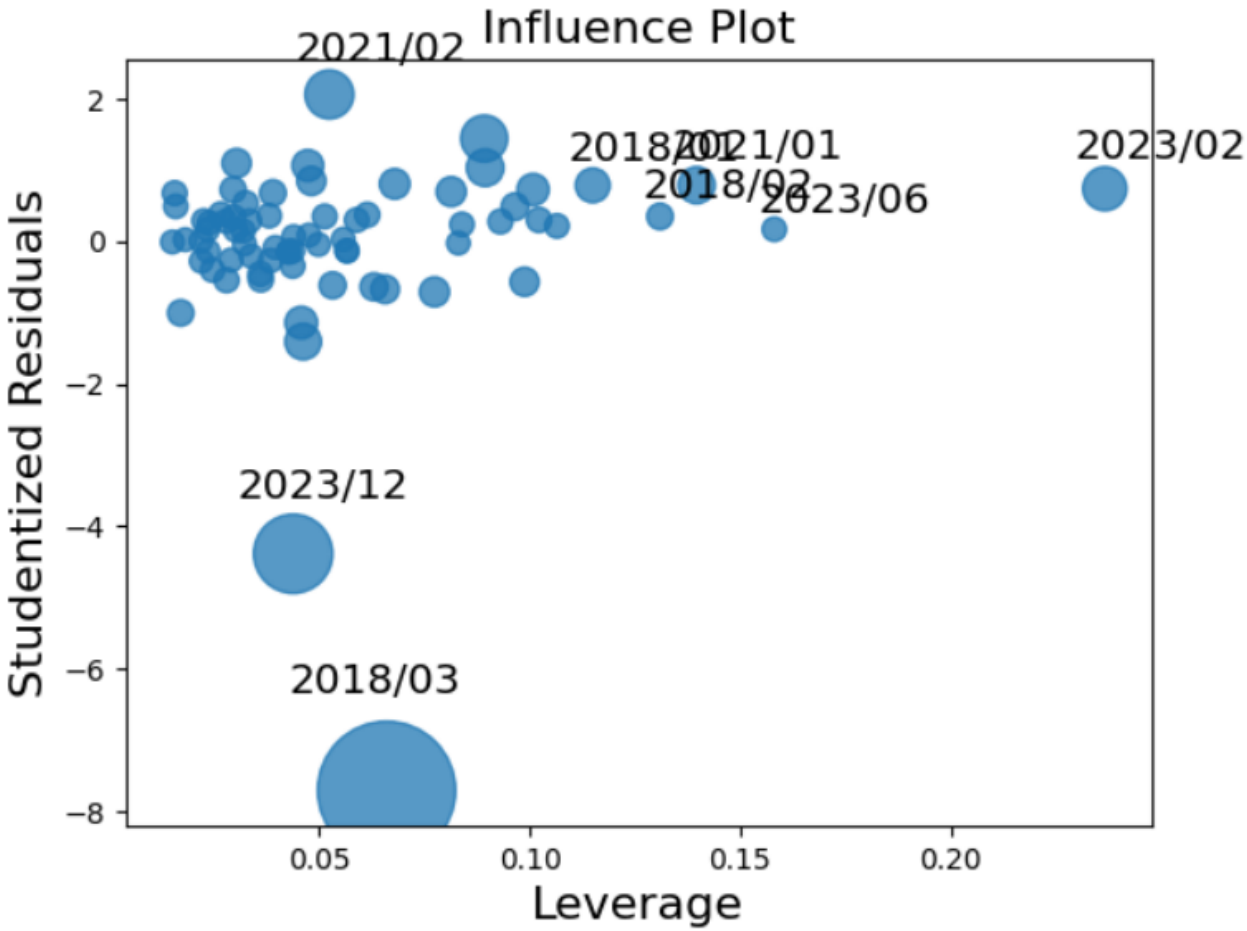
Measure of Influence



먼저, 잔차도를 통하여 outlier의 개수와 위치를 대략적으로 파악할 수 있습니다.

Measure of Influence

`influence_plot()`을 통해 각 점들의 influence, outlier, leverage를 시각화 할 수 있습니다. 이때 studentized residual은 저희가 사용한 것과 정의가 살짝 다른데, 이를 설명하기에는 시간이 부족해 생략합니다.



Measure of Influence

앞 페이지 그림에서 점의 크기는 influence의 크기를 의미합니다. influence를 계측하는 수단에는 여러 가지가 있습니다.

Cook's distance

$$C_i := \left(\frac{H_{ii}}{1 - H_{ii}} \right) \frac{d_i^2}{k+1}$$

$C_i > 1$ 혹은 $C_i > F_{0.5}(k+1, n-k-1)$이라면 influential observation으로 취급합니다. 한편 i번째 관측값을 빼고 회귀분석을 돌릴 때의 계수를 $\hat{\beta}_{-(i)}$라고 한다면,

$$C_i = \frac{(\hat{\beta} - \hat{\beta}_{-(i)})^T X^T X (\hat{\beta} - \hat{\beta}_{-(i)})}{(k+1)\hat{\sigma}^2}$$

이기도 합니다.

DFFITS(**DiF**erence in **FITS**)

$$DFFITS_i := \frac{\hat{y}_i - \hat{y}_{i-(i)}}{s_{-(i)}\sqrt{H_{ii}}}$$

이때 $\hat{y}_{i-(i)}$는 i번째 관측값을 빼고 적합한 회귀모형에 x_i를 넣었을 때의 에측값입니다. $s_{-(i)}$는 i번째 관측값은 빼고 적합한 회귀모형에서 얻은 $\hat{\sigma}$입니다. 만약

$$|DFFITS_i| > 2\sqrt{\frac{k+1}{n-k-1}}$$

이라면 이 점을 influence point로 취급할 수 있습니다.

Measure of Influence

```
1    influence.summary_frame()
2
```

	dfb_Intercept	dfb_상품수지_lag	dfb_서비스수지_lag	dfb_증권투자_lag	cooks_d	standard_resid	hat_diag	dffits_internal	student_resid	dffits
2018/01	-0.114468	-0.004906	-0.170910	-0.171540	0.019954	0.783917	0.114953	0.282519	0.781672	0.281709
2018/02	-0.065820	0.009170	-0.124968	0.011683	0.004551	0.347567	0.130952	0.134919	0.345309	0.134042
2018/03	0.131194	0.652290	0.893240	1.466649	0.564037	-5.651877	0.065969	-1.502048	-7.704413	-2.047531
2018/04	0.008544	-0.018308	0.011978	-0.014665	0.000223	-0.141534	0.042669	-0.029880	-0.140510	-0.029664
2018/05	-0.033074	0.081771	-0.024847	0.000192	0.003974	0.723307	0.029486	0.126075	0.720747	0.125629
...	...	...	...	...	...	...	...	...	...	...
2023/08	0.006809	-0.060309	-0.125557	-0.041895	0.009319	1.091915	0.030316	0.193067	1.093485	0.193345
2023/09	0.027986	-0.009778	0.000868	-0.021226	0.000963	0.488969	0.015857	0.062067	0.486216	0.061718
2023/10	0.004542	-0.001599	0.008665	0.001432	0.000032	-0.049266	0.049772	-0.011275	-0.048904	-0.011192
2023/11	-0.071005	0.011270	-0.033151	0.049714	0.004398	-1.007624	0.017034	-0.132642	-1.007739	-0.132657
2023/12	0.034915	-0.319440	0.386007	-0.683495	0.173531	-3.894610	0.043760	-0.833140	-4.385840	-0.938225

Measure of Influence

- `dfb_regressorname`: j번째 회귀계수 β_j의 스튜던트화된 변동

$$DFBETAS_{j,i} = \frac{\hat{\beta}_j - \hat{\beta}_{j-(i)}}{s_{-(i)}\sqrt{(X^T X)^{-1}_{j+1,j+1}}}$$

- `hat_diag`: leverage
- `dffits_internal`: DFFITS 계산에서 $s_{-(i)}$ 대신 $\hat{\sigma}$를 이용

Handling Outliers

Outlier, 혹은 influential point를 확인하였다면,

- 만약 이들 관측값이 잘못된 측정이나, 해당 시기 발생한 과도한 충격에 의한 것임이 알려져 있을 경우에는 그냥 삭제해도 됩니다.
- 반면 그렇지 않은 경우,
  1. 가장 중요한 정보를 담고 있는 자료로 보고 중요하게 리포트하거나,
  2. 새로운 모형을 적합하여 이상점을 없애거나

  등을 고려할 수 있습니다.

한편 아래에서는 `statsmodels`가 제공하는 다양한 이상점 관련 통계량들을 계산하는 함수들을 제시합니다. `https://www.statsmodels.org/stable/generated/statsmodels.stats.outliers_influence.OLSInfluence.htm`

Multicollinearity

우리는 앞서 다중선형회귀에서의 식별가능성이 $(X^TX)^{-1}$의 존재성에 의해 보장된다고 배웠습니다.

Some Linear Algebra

- $(X^TX)^{-1}$가 존재하는 것은 $(k+1)\times(k+1)$ 행렬 X^TX의 행렬식 $\det(X^TX)$이 0이 아닌 것과 동치입니다.
- $\det(X^TX)\neq 0$은 $\text{rank}(X^TX)=k+1$과 동치입니다.
- $\text{rank}(X^TX)=k+1$은 X가 full column rank $k+1$을 가지는 것과 동치입니다. 즉 X의 range는 차원이 $k+1$입니다.
- X의 range는 X의 column space의 차원과 동일합니다.
- X에 존재하는 $k+1$개의 column이 만드는 column space의 차원이 $k+1$이라는 것은, X의 모든 column이 **선형독립**이어야 함을 의미합니다.

Multicollinearity

만약 설명변수들에 대하여 어떠한 상수 $\gamma_0, \gamma_1, \cdots, \gamma_k$가 존재하여 모든 i에 대해서

$$\gamma_0 x_{i0} + \gamma_1 x_{i1} + \cdots + \gamma_k x_{ik} = 0$$

가 성립한다면, 설명변수들 사이에 **정확한 다중공선성**(exact multicollinearity)가 있다고 말합니다. 만약 정확한 다중공선성이 있다면, 우리는 유일한 최소제곱추정량을 얻을 수 없습니다.

한편 정확한 다중공선성이 없더라도, 만약 한 변수가 다른 변수들의 선형결합으로 거의 표현이 가능하다면 **다중공선성**(multicollinearity)이 있다고 말합니다.

Multicollinearity

다중공선성이 존재한다는 것은, 다르게 말하면 어떠한 변수 x_{ij}에 대하여, x_{ij}를 제외한 설명변수들의 쌍 $x_{i0}, x_{i1}, \cdots, x_{i(j-1)}, x_{i(j+1)}, \cdots, x_{ik}$이 있다면 x_{ij}를 잘 예측할 수 있다는 의미입니다. 그렇다면 우리가

$$x_{ij} = \gamma_0 x_{i0} + \gamma_1 x_{i1} + \cdots + \gamma_{j-1} x_{i(j-1)} + \gamma_{j+1} x_{i(j+1)} + \cdots + \gamma_k x_{ik} + \epsilon_i'$$

을 적합하여 얻은 결정계수를 R_j^2라 한다면, 어떤 j에 대해서는 R_j^2이 1에 가깝게 나온다는 것입니다.

Multicollinearity

한편 우리가 회귀계수의 추정량이 가지는 분산이

$$\mathrm{Var}(\hat{\beta}) = \sigma^2 (X^T X)^{-1}$$

임을 알고 있습니다. 또한 $\hat{\beta}_j$의 분산은

$$\mathrm{Var}(\hat{\beta}_j) = \sigma^2 (X^T X)^{-1}_{j+1,j+1}$$

으로 주어집니다. 이때 간단한 계산을 통해

$$(X^T X)^{-1}_{j+1,j+1} \propto \frac{1}{1 - R_j^2}$$

임을 밝힐 수 있습니다.

Multicollinearity

우리는 비편향추정량 중 가장 효율이 좋은 추정량을 분산이 작은 추정량으로 정의했습니다. 그렇기에 BLUE인 최소제곱추정량을 써왔던 것입니다. 그러나 다중공선성이 존재한다면, R_j^2가 1에 가깝기에, $(X^TX)^{-1}_{j+1,j+1}$이 커지며, 이에 따라 각 계수 $\hat{\beta}_j$의 분산이 커지는 결과를 낳게 됩니다. 큰 분산은 곧 추정량이 수렴하는 속도가 느리다는 것을 의미하며, 다르게 말하면 이 추정량이 실제와 멀 가능성이 굉장히 높다는 의미입니다.

VIF(Variance Inflation Factor)

$$VIF_j = \frac{1}{1 - R_j^2}$$

Multicollinearity

다중공선성을 파악하는 방법에는 크게 세 가지가 있습니다.

- (X^TX)를 잘 들여다보기: 만약 (X^TX)의 대각원소가 아닌 곳이 크다면, 해당하는 두 설명변수의 쌍이 선형적 관계를 가짐을 의미합니다. 다만 쌍별로만 관찰 가능합니다.
- *VIF*를 이용하기: 관행적으로, *VIF*가 5, 혹은 10보다 크다면 해당 j 번째 변수에 대해 다중공선성이 있는 것으로 파악합니다. 이는 해당 회귀계수가 잘 추정되지 않고 있음을 암시합니다.
- **Condition number**를 이용하기:

$$\kappa = \frac{\lambda_{\max}(X^TX)}{\lambda_{\min}(X^TX)}$$

가 100보다 크다면, 다중공선성이 있는 것으로 판단합니다.

```
Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.66e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Multicollinearity

```
    # condition number
    ols_result.condition_number
    # VIF
    from statsmodels.stats.outliers_influence import
    variance_inflation_factor
    pd.DataFrame({'variable': column, 'VIF':
    variance_inflation_factor(X, i)}
    for i, column in enumerate(X.columns)
    if column != 'Intercept')

```

	variable	VIF
0	상품수지_lag	1.081357
1	서비스수지_lag	1.026166
2	증권투자_lag	1.087203

Multicollinearity

다중공선성을 해결하는 방법에는 여러가지가 있습니다.

- VIF_j가 높게 나온 j번째 변수가 다른 변수들에 의해 잘 설명된다는 것은 곧 해당 변수가 없어도 회귀분석 자체는 잘 돌아간다는 의미입니다. 따라서 이 변수가 매우 중요하지 않은 경우, 제거합니다.
- n을 늘려서, 표본에 의해 우연히 관찰된 다중공선성을 희석시킬 수 있습니다. 다만 실제로 정확한 다중공선성이 존재할 경우에는 불가능합니다.
- condition number가 높게 나오는 경우, 변수들을 z점수화하여 사용할 수 있습니다. 이 경우 condition number가 꽤 많이 내려갑니다. 단, VIF는 고정됩니다.
- 변수에 로그변환 등을 취하여 선형 관계를 없앨 수 있습니다.

Heteroskedasticity

앞서 **이분산성**(heteroskedasticity)는 잔차도를 통해 간단히 확인해볼 수 있다고 하였습니다. 한편 이분산성이 있는 경우에는, 최소제곱추정량이 더이상 BLUE가 아니게 되며, 이에 따라 다른 더 좋은 추정량을 찾아야 할 수 있습니다. 따라서 이분산성을 통계적으로 검정할 수 있는 방법에 대해 알아보도록 하겠습니다.

- Breusch-Pagan test: `het_breuschpagan()` 함수를 이용합니다.
- White's test: `het_white()` 함수를 이용합니다.
- Goldfeld-Quandt test: `het_goldfeldquandt()` 함수를 이용합니다.
- Engle's test: 자기상관까지 고려하는 이분산성 검정으로, `het_arch()` 함수를 이용합니다.

이들을 통해 저희 데이터에는 이분산성이 없음을 확인할 수 있습니다.

Heteroskedasticity

한편 이분산성이 있다면,

$$\text{Var}(\epsilon) = \Omega \neq \sigma^2 I_n$$

으로 쓸 수 있고 이때 $\hat{\beta}$의 공분산 행렬은

$$\text{Var}(\hat{\beta}) = (X^T X)^{-1} X^T \Omega X (X^T X)^{-1}$$

으로 주어집니다. 그럼에도 우리는 이를 $\sigma^2 (X^T X)^{-1}$로 취급하고 추론을 하기 때문에, 검정과 신뢰구간의 계산에 오류가 생기게 되는 것입니다.

Heteroskedasticity

만약 ϵ_i들의 독립성은 유지된다면,

$$\text{Var}(\epsilon) = \Omega = \text{diag}(\sigma_1^2, \cdots, \sigma_n^2)$$

처럼 쓸 수 있고, 이때

$$\hat{\sigma}_i^2 = e_i^2$$

을 사용할 수 있습니다. 따라서 **heteroskedasticity-consistent(HC), 혹은 heteroskedasticity-robust 공분산행렬 추정량**은

$$\widehat{\text{Var}}_{HC0}(\hat{\beta}) = (X^T X)^{-1}(X^T \text{diag}(e_i^2) X)(X^T X)^{-1}$$

처럼 주어집니다. 이는 이분산이 존재하는 상황에서도 강건하게 OLS 추정량 $\hat{\beta}$의 분산을 구해줄 수 있다는 점에서 의미가 있습니다.

Heteroskedasticity

- HC0

$$\widehat{\text{Var}}_{HC0}(\hat{\beta}) = (X^T X)^{-1}(X^T \text{diag}(e_i^2) X)(X^T X)^{-1}$$

- HC1

$$\widehat{\text{Var}}_{HC1}(\hat{\beta}) = \frac{n}{n-k-1}\widehat{\text{Var}}_{HC0}(\hat{\beta})$$

- HC2(unbiased under homoskedasticity and BF problem)

$$\widehat{\text{Var}}_{HC2}(\hat{\beta}) = (X^T X)^{-1}\left(X^T \text{diag}\left(\frac{e_i^2}{1-H_{ii}}\right) X\right)(X^T X)^{-1}$$

- HC3

$$\widehat{\text{Var}}_{HC2}(\hat{\beta}) = (X^T X)^{-1}\left(X^T \text{diag}\left(\frac{e_i^2}{(1-H_{ii})^2}\right) X\right)(X^T X)^{-1}$$

Heteroskedasticity

```
ols_hc1 = ols_result.get_robustcov_results(cov_type = "HC1")
ols_hc1.cov_type # 'HC1'
ols_hc2 = ols_result.get_robustcov_results(cov_type = "HC2")
ols_hc2.cov_type # 'HC2'
print(ols_hc2.summary())

```

```
Covariance Type:                 HC2
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      0.0279      0.116      0.240      0.811      -0.204       0.260
상품수지_lag   -1.561e-05   1.78e-05     -0.875      0.385   -5.12e-05       2e-05
서비스수지_lag   1.594e-05   6.78e-05      0.235      0.815      -0.000       0.000
증권투자_lag   -1.912e-06   2.42e-05     -0.079      0.937   -5.02e-05    4.64e-05
==============================================================================
```

Covariance Type 등이 변하였습니다.

Extensions

Variance Stabilizing Transformation

1. 분산안정화변환
   만약 $\mathbb{E}[y] = \mu, \mathrm{Var}(y) = g(\mu)$임을 대략적으로 알고 있다면, $h' = \frac{1}{\sqrt{g}}$인 함수 h를 이용할 경우

$$\mathrm{Var}(h(y)) \approx h'(\mu)\mathrm{Var}(y) = h'(\mu)^2 g(\mu) = 1$$

   으로 안정화가 됨을 알 수 있습니다.
2. 로그변환
   만약 정확히 g를 모르더라도, g가 증가함수라면 h를 로그함수로 할 때

$$\mathrm{Var}(\ln y) \approx \frac{1}{\mu^2}\mathrm{Var}(y) = \frac{g(\mu)}{\mu^2}$$

   으로 $g(\mu)$가 μ^2보다 빠르게 증가하는 지수함수 등일 때 분산을 비슷하게 만드는 데 기여할 수 있습니다.

WLS and GLS

만약 분산행렬이

$$\text{Var}(\epsilon) = \Sigma = \text{diag}(\sigma_1^2, \cdots, \sigma_n^2)$$

이라면, 이때의 BLUE는 **WLS**(Weighted Least Squares) 추정량을 통해 계산됩니다. 이는

$$\hat{\beta}^{WLS} = \underset{\beta}{\text{argmin}} \sum_{i=1}^{n} w_i (y_i - X_i\beta)^2, \quad w_i \propto \frac{1}{\sigma_i^2}$$

와 같습니다. 이를 조금 더 일반화하면, Σ가 대각행렬이 아니라 일반적인 양정치행렬이라고 할 수 있습니다 .이 경우 BLUE는 **GLS**(Generalized Least Squares) 추정량을 통해 계산됩니다. 식으로 표현하면,

$$\hat{\beta}^{GLS} = \underset{\beta}{\text{argmin}} (y - X\beta)^T \Sigma^{-1} (y - X\beta)$$

WLS and GLS

WLS는 GLS의 일반적인 형태이므로, 우리는 GLS에 대해서만 알아보겠습니다. GLS 추정량은

$$\hat{\beta}^{\text{GLS}} = (X^T \Sigma X)^{-1} X^T \Sigma^{-1} y$$

로 쓸 수 있으며, BLUE이고,

$$\mathbb{E}[\hat{\beta}^{\text{GLS}}] = \hat{\beta}, \quad \text{Var}(\hat{\beta}^{\text{GLS}}) = \sigma^2 (X^T \Sigma^{-1} X)^{-1}$$

이라는 성질을 가집니다. 하지만 이 방법의 가장 큰 문제점은 우리가 선행적으로 Σ의 구조를 어느 정도 알아야 한다는 점입니다(**Infeasible**).

WLS and GLS

covariance structure를 모르는 경우, **feasible GLS**를 대용적으로 이용할 수 있습니다. 그 과정은 아래와 같습니다.

1. 먼저 모형을 OLS와 같은 간단한 모형으로 적합한 다음, 잔차와 적절한 가정을 이용하여 covariance matrix $\hat{\Sigma}$를 만든다.
2. $\hat{\Sigma}$를 이용해 GLS를 적합한다.

우리처럼 패널 데이터를 이용하는 경우, 많이 사용하는 가정 중 하나는 error가 자기상관성을 가진다는 것입니다. 즉

$$\epsilon_i = \rho\epsilon_{i-1} + \eta_i$$

이며 $\eta \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2)$이라 두면 잔차로써 추정한 ρ를 이용해 Var(ϵ)을 추정할 수 있습니다.

WLS and GLS

AR 구조를 이용한 feasible GLS인 GLSAR은 `smf.glsar()`을 통해 쉽게 수행해줄 수 있습니다.

```
glsar_result = smf.glsar(formula = '외국인보유비율변화 ~ 상품수지_lag + 서비스수지_lag + 증권투자_lag', data = data).fit()
print(glsar_result.summary())
```

GLSAR Regression Results

Dep. Variable:	외국인보유비율변화	R-squared:	0.012
Model:	GLSAR	Adj. R-squared:	-0.032
Method:	Least Squares	F-statistic:	0.2698
Date:	Sat, 23 Mar 2024	Prob (F-statistic):	0.847
Time:	21:44:55	Log-Likelihood:	-69.667
No. Observations:	71	AIC:	147.3
Df Residuals:	67	BIC:	156.4
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	0.0472	0.171	0.277	0.783	-0.293	0.388
상품수지_lag	-1.55e-05	2.13e-05	-0.729	0.468	-5.79e-05	2.69e-05
서비스수지_lag	2.847e-05	7.51e-05	0.379	0.706	-0.000	0.000
증권투자_lag	1.166e-06	1.84e-05	0.063	0.950	-3.55e-05	3.78e-05

Omnibus:	74.555	Durbin-Watson:	1.424
Prob(Omnibus):	0.000	Jarque-Bera (JB):	694.009
Skew:	-3.059	Prob(JB):	1.99e-151
Kurtosis:	17.042	Cond. No.	1.65e+04

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.65e+04. This might indicate that there are strong multicollinearity or other numerical problems.

Polynomial Regression

만약 잔차도에서 선형성이 발견되지 않고, 이차함수 그래프와 유사한 형태의 잔차가 나온다면 해당 변수에 대한 제곱항을 새로 넣는 것을 고려해볼 수 있습니다.

```
1   ols_result = smf.ols(formula = '외국인보유비율변화 ~ 상품수지_lag +
      서비스수지_lag + I(상품수지_lag ** 2)', data = data).fit()
2   print(ols_result.summary())
3
```

이 경우 `I()`를 이용할 수 있습니다.

Polynomial Regression

고차항까지 고려하는 경우, `np.vander()`을 이용합니다.

```
1    ols_result_2 = smf.ols(formula = '외국인보유비율변화 ~ np.vander(
     상품수지_lag, 3, increasing = True)', data = data).fit()
2    print(ols_result_2.summary())
3
```

단, 고차항이 많아질수록

- multicollinearity $\Rightarrow$ 적절히 centering, 직교다항식 등을 이용하여 해결
- 해석이 어려워짐
- 일반적인 설명변수를 넘어가는 값들에 대해 예측력이 감소 (extrapolation 문제)

Without Intercept

절편 부분을 없애고 싶은 경우, formula를 쓸 때 뒤에 -1을 붙여 줍니다.

```
1     ols_result_4 = smf.ols(formula = '외국인보유비율변화 ~
      상품수지_lag + 서비스수지_lag + 증권투자_lag - 1', data = data).fit()
2     print(ols_result_4.summary())
3
```

Interaction Term

상호작용항을 넣고자 하는 경우, 두 변수 사이에 *을 넣어 주면 됩니다. 오직 상호작용항만 넣고자 하는 경우에는 :을 이용할 수 있습니다.

```
1  ols_result_5 = smf.ols(formula = '외국인보유비율변화 ~ 상품수지_lag +
     서비스수지_lag * 증권투자_lag', data = data).fit()
2  print(ols_result_5.summary())
3
4  ols_result_6 = smf.ols(formula = '외국인보유비율변화 ~ 상품수지_lag +
     서비스수지_lag :증권투자_lag', data = data).fit()
5  print(ols_result_6.summary())
6
```

Indicator Variable

남자/여자, 경제학부/통계학과/경영대학과 같이 수치형으로 표현하기 애매한 자료의 경우, **지시변수**(indicator variable)을 추가하여 회귀분석을 수행할 수 있습니다. 수치형변수 역시도 `I()`를 이용하여 지시변수로 변환한 뒤 회귀분석에 이용할 수 있습니다.

```
1    ols_result_7 = smf.ols(formula = '외국인보유비율변화 ~
     상품수지_lag + I(서비스수지_lag > 0)', data = data).fit()
2    print(ols_result_7.summary())
3
```

Factors

변수가 경제학부/통계학과/경영대학과 같이 세 개 이상의 항목으로 이루어진 경우, 하나를 베이스라인으로 하고 나머지들을 모두 지시변수로 만들어 이용합니다. 단순히 포뮬러에 넣기만 하여도, 자동으로 계산해 줍니다.

```
1      from functools import reduce
2      from operator import add
3      data['
       분기'] = reduce(add, [[1] * 3, [2] * 3, [3] * 3, [4] * 3]* 6)
4      data['분기'] = data['분기'].apply(str)
5      ols_result_8 = smf.ols(formula = '외국인보유비율변화 ~
       상품수지_lag + 분기', data = data).fit()
6      print(ols_result_8.summary())
7
```

Interpretation of Complex Model

여기에서는 코드 없이 이론적으로만 한 번 해석하는 방법을 알아 보도록 하겠습니다. 반응변수는 계량경제학 중간고사 점수이고, 설명변수는 TA세션 참여 횟수와 경제통계학 학점(A/B/C)라고 해 봅시다. 우리는 그렇다면 아래처럼 모형을 세울 수 있습니다.

$$y = \beta_0 + \beta_1(TA) + \beta_2(A) + \beta_3(B) + \beta_4(TA : A) + \beta_5(TA : B) + \epsilon$$

만약 이 모형을 적합하려면, `TA`세션*경통학점만 formula에 넣어주면 되겠습니다. 이 모형을 어떻게 해석해야 할까요?
Note: C학점이 변수에 없는 이유는, C학점을 받은 학생을 기저에 두고 A와 B학점으로 그 성적이 바뀌었을 때 영향만 고려해도 되기 때문입니다.

Interpretation of Complex Model

만약 학생이 경통 A학점이라면, 해당 학생의 모형은

$$y = \beta_0 + \beta_1(TA) + \beta_2(A) + \beta_4(TA : A) + \epsilon = (\beta_0 + \beta_2) + (\beta_1 + \beta_4)(TA) + \epsilon$$

가 됩니다. 따라서 경통 A인 학생이 TA세션에 1회 더 참여함에 따른 성적 상승 효과는 $\beta_1 + \beta_4$가 되며, 이것이 유의한지는 앞에서 다룬 것처럼 복잡한 F통계량에 기반하여 새로 검정해야 합니다. 경통 B학점인 학생의 모형은

$$y = \beta_0 + \beta_1(TA) + \beta_3(B) + \beta_5(TA : B) + \epsilon = (\beta_0 + \beta_3) + (\beta_1 + \beta_5)(TA) + \epsilon$$

이며, A와 B학점을 받은 학생의 TA세션 참여에 따른 효과 차이는 $\beta_4 - \beta_5$가 될 것입니다. 또다시 이것이 유의하냐에 대한 가설 검정을 통해 두 학생의 TA세션 효과 차이를 확인해줄 수 있겠습니다.

Generalized Linear Model

일반화선형모형(Generalized Linear Model; GLM)은 선형회귀분석을 확장하여 선형성, 등분산성, 비정규성 하에서도 변수들의 선형관계로써 반응변수를 추정할 수 있게 하는 확장된 모형입니다.

대표적으로, y가 어떠한 수치형 변수가 아니라 이산형 변수인 상황을 고려할 수 있습니다. 적절한 설명변수들로써 각 개인이 어떤 프로젝트에 참여할 것인지(1), 참여하지 않을 것인지(0) 예측하고자 합니다. 그렇다면 ϵ은 항상 y의 값이 1과 0이 되는 방향으로 결정되기에, 다중선형회귀에서의 ϵ이 갖추어야 할 여러 조건를 만족하지 못합니다. 일반화선형 모형 중 **로지스틱 회귀분석**이 이러한 문제를 해결할 수 있습니다.

Generalized Linear Model

GLM은 아래와 같은 형태를 가집니다.

$$g(\mathbb{E}[y_i|X_i]) = X_i\beta$$

이때 g를 **link function**이라 부르며, g는 y_i가 어떠한 분포를 따르냐에 따라 결정됩니다. 만약 다중회귀분석에서처럼 y_i가 평균이 $X_i\beta$인 정규분포를 따른다면, g는 항등함수가 되겠습니다(identity link).

- y가 푸아송 분포를 따르는 경우, log link ln()을 사용합니다.
- y가 감마분포를 따르는 경우, reciprocal link $\frac{1}{\cdot}$을 이용합니다.
- y가 inverse normal 분포를 따르는 경우, $\frac{1}{\cdot^2}$을 이용합니다.
- y가 이항분포를 따르면, complementary log-log link $\log(-\log(1-\cdot))$를 사용하기도 합니다.

Logistic Regression

만약 y_i가 이항분포를 따르는 경우, $\mathbb{E}[y_i|X_i]$는 설명변수로 X_i를 가지는 개인이 $y_i = 1$일 확률로 이해할 수 있으며, 이는 곧 그 확률이 X_i들의 선형결합으로 표현될 수 있음을 의미합니다. **로지스틱 회귀분석**(logistic regression)에서는 link function으로 **logit transformation**

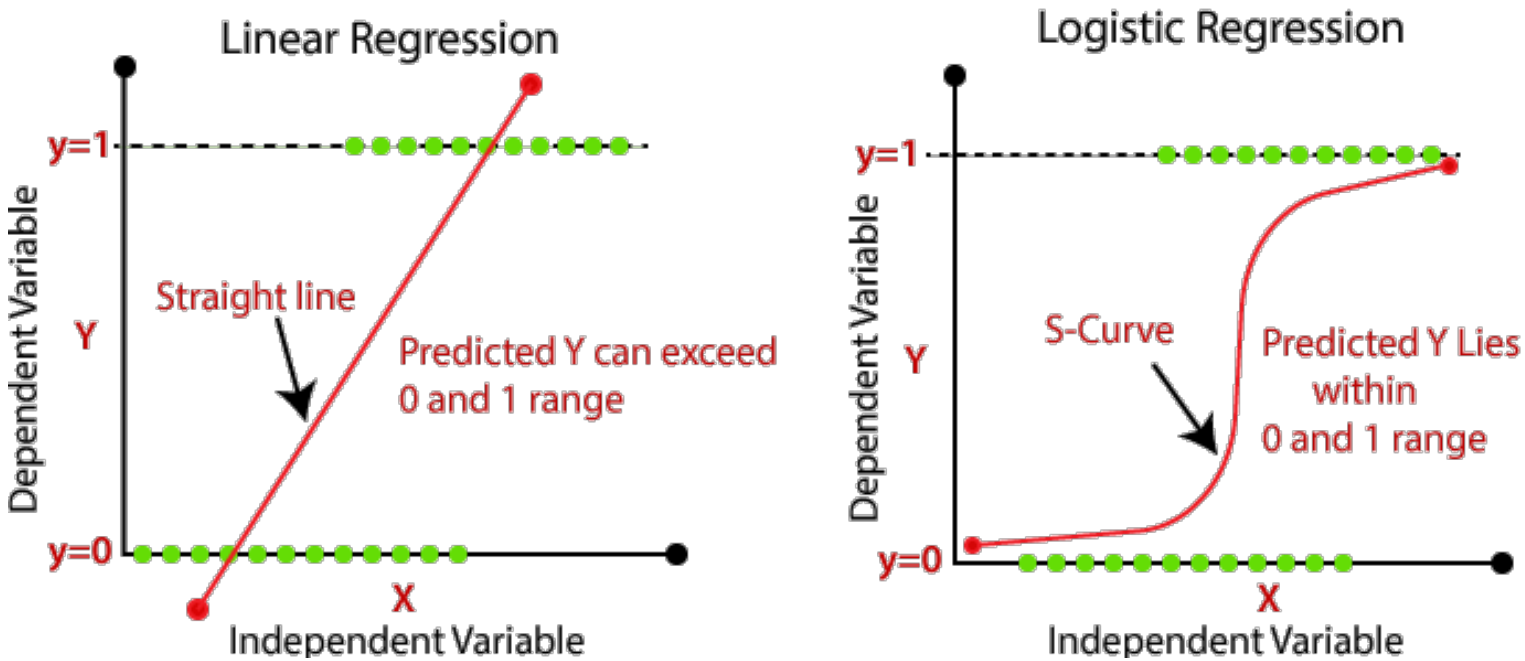
$$g(\mu) = \ln \frac{\mu}{1-\mu}$$

을 이용합니다. 따라서 정리하면

$$\pi_i = \mathbb{E}[y_i|X_i] = \frac{e^{X\beta}}{1+e^{X\beta}} = \frac{1}{1+e^{-X\beta}}$$

가 되며, $X\beta$가 증가할수록 분모가 감소하여 $y_i = 1$일 확률이 높아지고, 반대로 감소할수록 $y_i = 0$일 확률이 높아짐을 볼 수 있습니다. 또한 이 값은 항상 $[0, 1]$의 범위에 있게 됩니다.

Logistic Regression



이때, $\frac{\pi_i}{1-\pi_i}$를 **odds**(오즈)라고 부릅니다.

Logistic Regression

```
1    # logistic regression model
2    data['외국인증가'] = (data['외국인보유비율변화'] > 0) * 1
3    logistic_result = smf.logit('외국인증가 ~
     상품수지_lag', data= data).fit()
4    print(logistic_result.summary())
5
```

외국인보유비율 증감 여부에 따라 '외국인증가' 열을 새로 만들고, 이를 반응변수로 한 다음 `smf.logit()` 함수를 이용하여 로지스틱 회귀를 적합한 결과는 다음 페이지와 같습니다.

한편, 최소제곱법을 이용했던 선형회귀와는 다르게 GLM은 가능도 기반으로 적합합니다. 다만 여기에서 진단 등을 말씀드리기에는 너무 많은 시간이 필요한 관계로... 계수의 해석에 관련해서만 논의하겠습니다.

Logistic Regression

```
Optimization terminated successfully.
         Current function value: 0.679533
         Iterations 4
                           Logit Regression Results
==============================================================================
Dep. Variable:                  외국인증가   No. Observations:                   72
Model:                          Logit   Df Residuals:                       70
Method:                           MLE   Df Model:                            1
Date:                Sat, 23 Mar 2024   Pseudo R-squ.:                 0.01909
Time:                        23:05:00   Log-Likelihood:                -48.926
converged:                       True   LL-Null:                       -49.879
Covariance Type:            nonrobust   LLR p-value:                    0.1675
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      0.4229      0.429      0.985      0.325      -0.419       1.264
상품수지_lag    -8.69e-05   6.44e-05     -1.349      0.177      -0.000    3.94e-05
==============================================================================
```

유의성 판단 등은 다중선형회귀와 동일합니다.

'상품수지_lag'의 계수 -8.69e-05를 $\hat{\beta}_1$이라 해 봅시다. 그렇다면 전월 상품수지가 1만큼 차이나는 두 경우를 고려해 보면,

$$\ln \frac{\hat{\pi}_i(x_i)}{1-\hat{\pi}_i(x_i)} = \hat{\beta}_0 + \hat{\beta}_1 x_i$$
$$\ln \frac{\hat{\pi}_i(x_i+1)}{1-\hat{\pi}_i(x_i+1)} = \hat{\beta}_0 + \hat{\beta}_1 (x_i+1)$$

이고, 양변을 빼고 지수로 취하면

$$e^{\hat{\beta}_i} = \exp\left(\ln \frac{\hat{\pi}_i(x_i+1)}{1-\hat{\pi}_i(x_i+1)} - \ln \frac{\hat{\pi}_i(x_i)}{1-\hat{\pi}_i(x_i)}\right) = \frac{\widehat{\text{odds}}_{x_i+1}}{\widehat{\text{odds}}_{x_i}}$$

입니다. 즉 회귀계수는 변수가 1단위 차이나는 두 집단의 **오즈비**(odds ratio)의 로그로 취급합니다.

Probit Model

Probit Model은 로지스틱 회귀와 거의 비슷한데, link function만 다른 형태입니다. 여기에서는 **probit link** $\Phi^{-1}(\cdot)$을 이용합니다. Φ는 표준정규분포의 누적분포함수입니다. 따라서 probit model은

$$P(y_i = 1|X_i) = \Phi(X^T\beta)$$

처럼 써집니다.

```
1  probit_result = smf.probit('외국인증가 ~
     상품수지_lag', data= data).fit()
2  print(probit_result.summary())
3
```

도움이 되는 링크들

- 일반적인 GLM을 statsmodels에서 수행하기: https://www.statsmodels.org/stable/glm.html
- count data에 많이 사용하는 Poisson regression 소개: https://www.pymc.io/projects/examples/en/latest/generalized_linear_models/GLM-poisson-regression.html
- Probit Model과 Logit Model 중에서 일반적으로 Logit model이 해석이 더 쉽고 연구도 많이 되어 있어서 많이 사용되지만, 각각은 서로 다른 가정을 가집니다. 만약 응용미시처럼 특정 모형이 있고 이를 적합하는 상황이라면, 해당 모형의 가정에 맞는 link를 찾고 이를 사용하셔야 합니다. (discrete choice model 등에서 각종 link를 사용함) (https://stats.stackexchange.com/questions/20523/difference-between-logit-and-probit-models)

Take-Home Messages

- 오늘 템포를 느리게 하겠다고 마음을 먹었는데, 어째 지난주보다 더 빨라진 것 같네요ㅠㅠ 학부 계량 반 정도를 2시간에 하다 보니, 어쩔 수 없는 것 같습니다...
- 계산이나 유도 디테일은 다 버려도, 키워드들은 포함하려 노력했습니다. 관련해서 궁금하신 내용은 키워드들 바탕으로 구글링해보시길 추천드립니다.
- endogeneity의 확인과 해결법에 대해서는 중간고사 이후 인과추론 세션에서 다루겠습니다. model selection과 nonlinear regression에 대한 이야기도 빼 두었는데, 이는 차원축소/비모수모형 세션에서 다룹니다.
- 종합하면, 실제로 진단해 보면 완벽히 가정에 들어 맞는 데이터는 없기 마련이고, 적절한 선에서 변형한 뒤 타협하여 좋은 결과들만(...) 리포트하는 게 대부분의 응용통계에서 '국룰'로 취급되는 것 같습니다. 다만 앞서 말씀드렸듯, 그러면 솔직하게 어떻게 전처리를 했다고 꼭 리포트를 해 주셔야 하고, 노가다가 귀찮다고 값 조작하시는 건 절대 금물입니다.

HW2

1. HW2는 다양한 형태의 회귀분석을 직접 수행해보는 것을 목표로 합니다.
2. HW2의 수행 과정에서는 1주차에서 사용한 `pandas` 테크닉이 많이 필요할 수 있습니다.
3. HW2의 최종 목표는 제가 어떠한 데이터 생성 과정을 통해 만든 인공 데이터에서 실제 모형을 찾고 그에 대한 진단과 추론을 해보는 것입니다.
4. 또한 해당 결과는 적절한 시각화를 동반하여, LaTeX을 통해 조판되어야만 합니다.