HW_1

statisticalpuzzle.txt 데이터를 이용해서 다중회귀분석을 수행하고 잔차에 대한 분석하자.

1. 데이터 준비

HW_13_statisticalpuzzle.txt 파일을 공백 구분 형식으로 읽어온다.

종속변수는 **y**, 설명변수는 **x1~x6**이다.

절편 항은 **statsmodels.api.add_constant()**로 추가한다.

python

```
df  = pd.read_csv('HW_13_statisticalpuzzle.txt',
delim_whitespace=True)
y   = df['y']
X   = sm.add_constant(df.drop(columns=['y']))
```

---

2. 회귀모형 적합

sm.OLS(y, X).fit()으로 다중회귀모형을 적합한다.

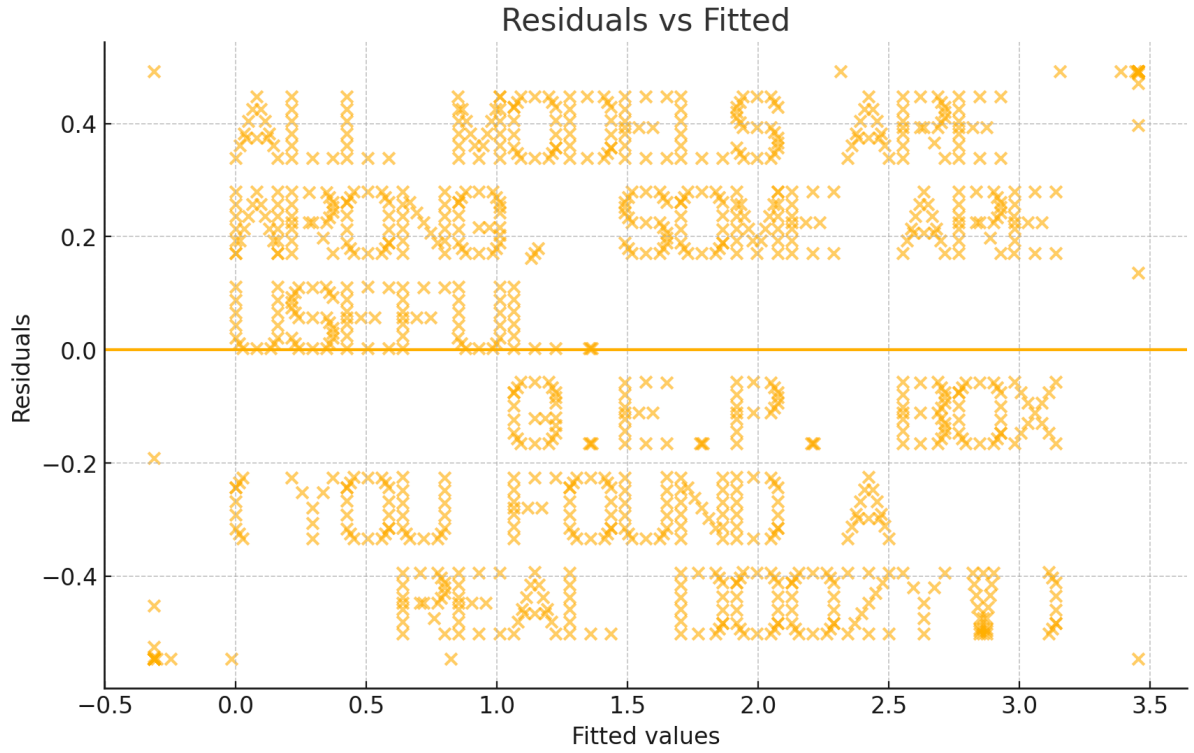
지표	결과	해석
R^2	**0.900**	설명변수들이 **yyy** 변동의 **90 %**를 설명한다.
Adj. R^2	**0.899**	변수 수를 보정해도 설명력 손실이 거의 없다.
F-stat	**1 424 (p < 0.001)**	모형 전체가 통계적으로 유의하다.
계수	절편 **0.64**, 각 **xkx_kxk** 계수 **≈ 1 (p < 0.001)**	데이터 생성 규칙이 **y≈0.64+∑xky ≈ 0.64 + \sum x_ky≈0.64+∑xk** 형태임을 시사한다.

---

3. 잔차 시각화 및 해석

3-1. Residuals vs Fitted

적합값이 커질수록 잔차 범위가 넓어지는 '깔때기' 패턴이 나타난다. 이는 오차 분산이 일정하지 않다는 이분산성을 시사한다.

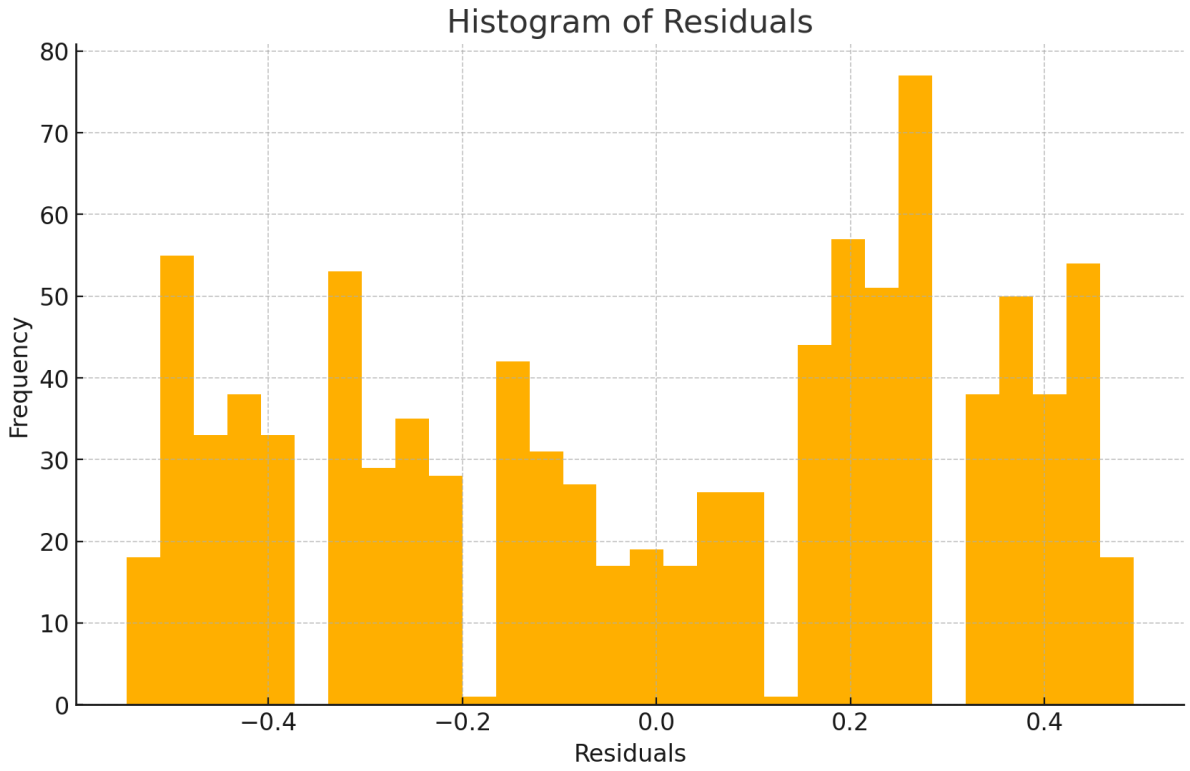


3-2. QQ Plot

중앙부는 직선에 가깝지만 양쪽 꼬리가 크게 벗어난다. 잔차의 분포가 정규분포 가정에서 벗어나 있음을 확인한다.

3-3. Residual Histogram

분포가 좌·우 꼬리에 뚜렷한 뾰족함 없이 편평하다. 정규분포 대비 꼬리 가벼움이 드러난다.



4. 잔차 검정

검정	통계량	**p-value**	결론
Breusch-Pagan	**LM ≈ 11.3**	**0.009**	**p < 0.05** → 등분산 가설 기각**(**이분산 존재**)**
Jarque-Bera	**JB ≈ 79.2**	**< 0.001**	**p < 0.05** → 정규성 가설 기각
Durbin-Watson	**0.067**	**–**	**0**과 **2** 사이에서 크게 벗어나 양의 자기상관 의심

---

5. 개선 방안

1. 헤테로스케다스틱티 보정
    - **White·HC3 robust** 표준오차 사용하거나
    - 잔차 분산을 적합값 함수로 두고 가중 최소제곱**(WLS)** 재적합한다**.**
2. 자기상관 보정
    - 데이터가 시계열 구조라면 **GLS**나 **AR(1)** 오차 모형을 고려한다**.**
3. 정규성·등분산 동시 개선
    - **Box-Cox** 등 응답 변환을 적용하거나
    - 분위수 회귀처럼 분포 가정을 완화한 방법을 쓴다**.**

---

6. 결론

모형은 잔차 가정**(**등분산·정규성·독립성**)**이 모두 위배된다**.**

따라서 계수 추정치는 믿을 수 있으나**,** 신뢰구간·가설검정·예측 정확도는 저하될 위험이 있다**.**

```python
import pandas as pd
import statsmodels.api as sm
import matplotlib.pyplot as plt
from statsmodels.stats.diagnostic import het_breuschpagan
from scipy import stats

# Load dataset
file_path = 'HW_13_statisticalpuzzle.txt'
df = pd.read_csv(file_path, delim_whitespace=True)

# Separate predictors and response
Y = df['y']
X = df.drop(columns=['y'])
X = sm.add_constant(X)  # add intercept

# Fit multiple regression model
model = sm.OLS(Y, X).fit()

# Display model summary
print(model.summary())

# Obtain fitted values and residuals
fitted = model.fittedvalues
residuals = model.resid

# Residuals vs Fitted plot
plt.figure()
plt.scatter(fitted, residuals)
plt.axhline(0)
plt.xlabel("Fitted values")
plt.ylabel("Residuals")
plt.title("Residuals vs Fitted")
plt.show()

# QQ plot for residuals
sm.qqplot(residuals, line='45')
plt.title("QQ Plot of Residuals")
plt.show()

# Histogram of residuals
plt.figure()
plt.hist(residuals, bins=30)
plt.title("Histogram of Residuals")
plt.xlabel("Residuals")
plt.ylabel("Frequency")
```

```
plt.show()

# Breusch-Pagan test for heteroscedasticity
bp_test = het_breuschpagan(residuals, X)
labels = ['Lagrange multiplier statistic', 'p-value', 'f-value', 'f p-value']
print("Breusch-Pagan test results:", dict(zip(labels, bp_test)))

# Jarque-Bera test for normality
jb_stat, jb_pvalue = stats.jarque_bera(residuals)
print(f"Jarque-Bera test: stat = {jb_stat:.4f}, p-value = {jb_pvalue:.4f}")
```