```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.formula.api as smf
from sklearn.linear_model import LinearRegression
```

Q1

```
# 구글드라이브 마운트

from google.colab import drive
drive.mount('/content/drive')
```

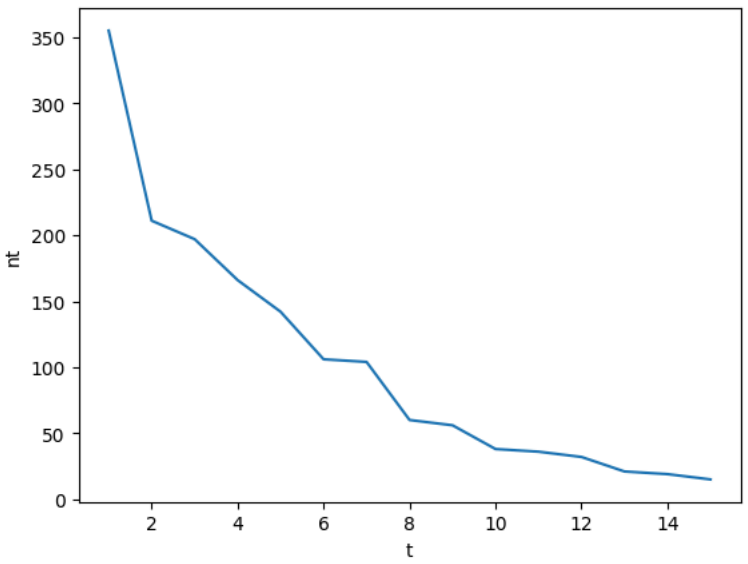
```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
data = pd.read_csv("/content/drive/MyDrive/tobigs/정규세션/1주차/bacteria.csv")
```

```
data.head()
```

	t	nt
0	1	355
1	2	211
2	3	197
3	4	166
4	5	142

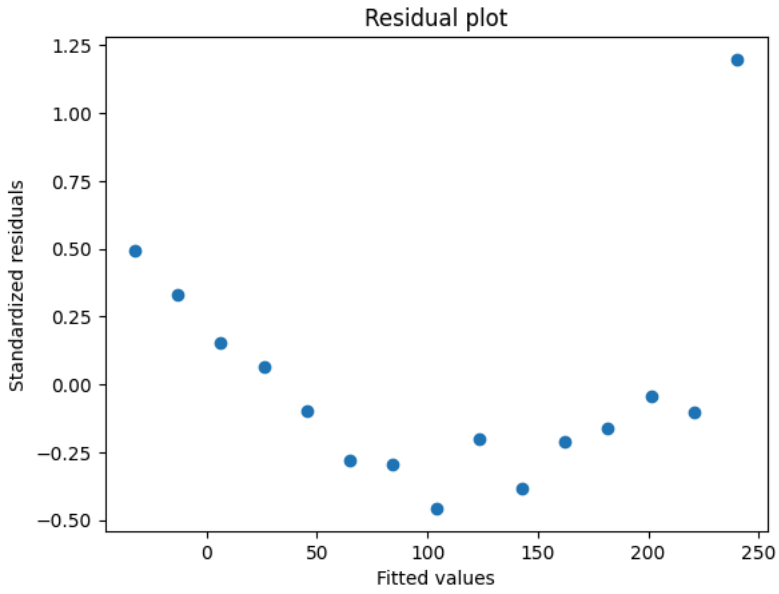
```
# 데이터 추세 파악
plt.plot(data['t'], data['nt'])
plt.xlabel('t')
plt.ylabel('nt')
plt.show()
```




```
model = LinearRegression()
model.fit(data[['t']], data['nt'])

# 예측값과 표준화 잔차 계산
fitted_values = model.predict(data[['t']])
standardized_residuals = (data['nt'] - fitted_values) / data['nt'].std()
```

```
# residual plot
plt.scatter(fitted_values, standardized_residuals)
plt.xlabel('Fitted values')
plt.ylabel('Standardized residuals')
plt.title('Residual plot')
plt.show()
```

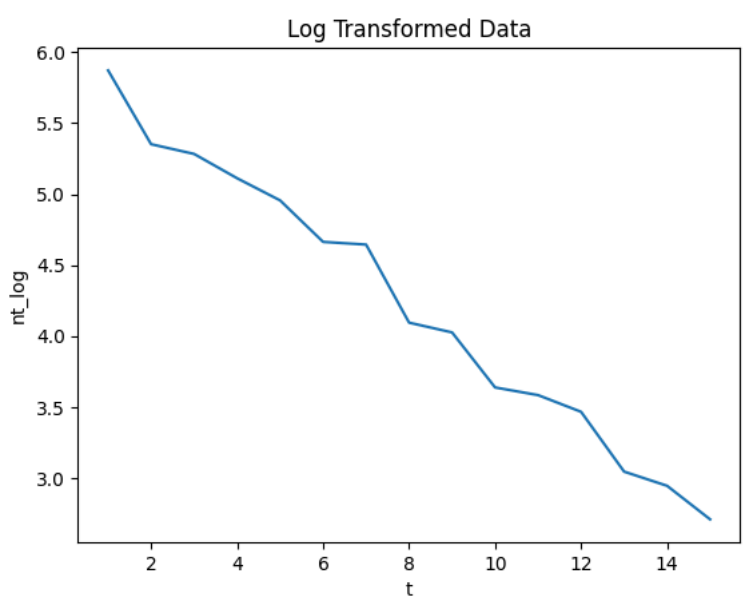


bacteria 데이터에 단순 회귀 모형을 적용하였더니 등분산성을 위배하는 잔차 양상이 관찰되었습니다. 이를 개선하기 위해 데이터 변환 기법, 가중 최소제곱법 등 적절한 통계적 기법을 동원하여 등분산성이 관측되도록 해주세요. (아래에 코드 작성해주세요)

```
#ans :

# 로그 변환
data['nt_log'] = np.log(data['nt'])

# 로그 변환된 데이터 시각화
plt.plot(data['t'], data['nt_log'])
plt.xlabel('t')
plt.ylabel('nt_log')
plt.title('Log Transformed Data')
plt.show()
```



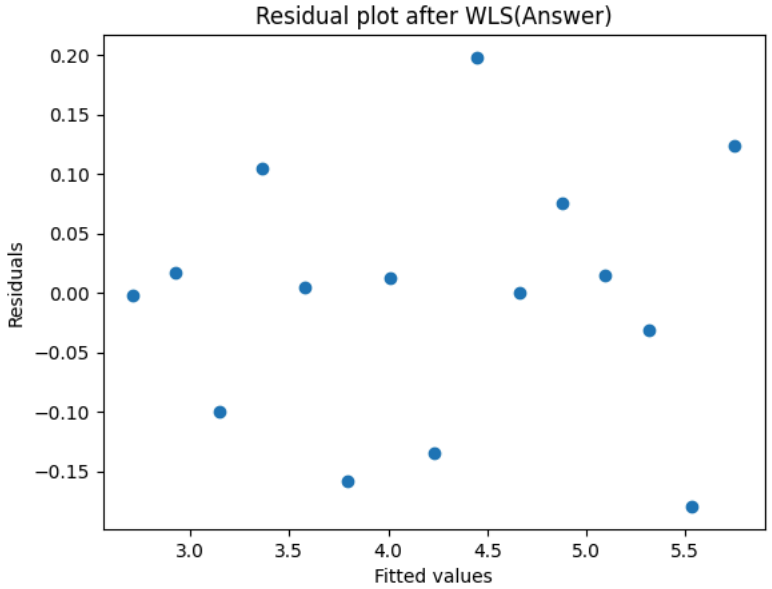

```
# 단순선형회귀 모델 생성
model_log = LinearRegression()
model_log.fit(data[['t']], data['nt_log'])

# 로그 변환 데이터 - 예측값과 표준화 잔차 계산
fitted_values_log = model_log.predict(data[['t']])
standardized_residuals_log = (data['nt_log'] - fitted_values_log) / data['nt_log'].std()
```

```
# 로그 변환된 데이터에 가중 최소제곱법(WLS) 적용
import statsmodels.api as sm

weights = 1 /standardized_residuals_log**2
X = sm.add_constant(data[['t']])
model_wls = sm.WLS(data['nt_log'], X, weights=weights).fit()

# 잔차 그래프
plt.scatter(model_wls.fittedvalues, model_wls.resid)
plt.xlabel('Fitted values')
plt.ylabel('Residuals')
plt.title('Residual plot after WLS(Answer)')
plt.show()
```



Q2

```
# 데이터 로드
data2 = pd.read_csv("/content/drive/MyDrive/tobigs/정규세션/1주차/artificial1.csv")
```

```
data2.head()
```

	X	Y
0	11.0142	24.8831
1	2.7066	14.7374
2	11.5839	64.0250
3	8.9989	16.1965
4	2.1201	7.3907

```
# 선형 회귀 모형 적합
result1 = smf.ols('Y ~ X', data=data2).fit()

# 잔차 표준화 및 산점도 그리기
fitted_values = result1.predict()
standardized_residuals = result1.get_influence().resid_studentized_internal
```

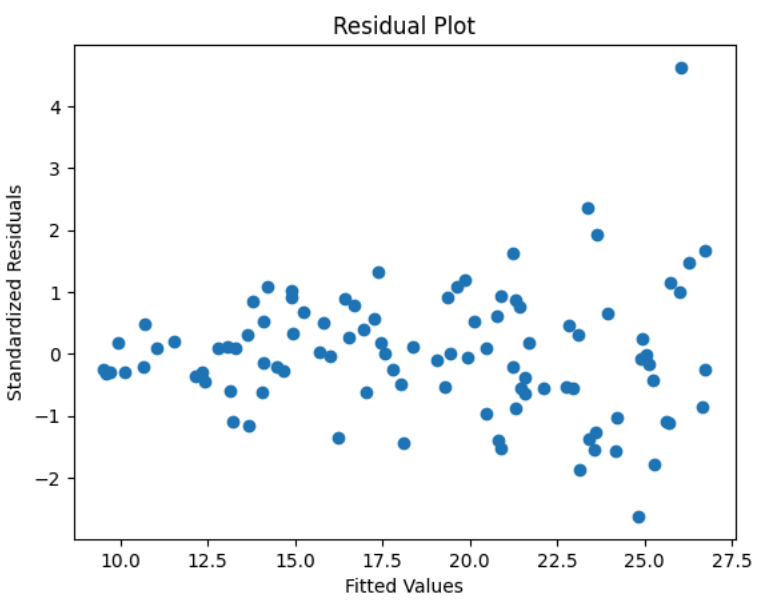
```
# 요약 정보 출력
print(result1.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      Y   R-squared:                       0.261
Model:                            OLS   Adj. R-squared:                  0.254
Method:                 Least Squares   F-statistic:                     34.69
Date:                Wed, 24 Jul 2024   Prob (F-statistic):           5.44e-08
Time:                        01:13:04   Log-Likelihood:                -352.93
No. Observations:                 100   AIC:                             709.9
Df Residuals:                      98   BIC:                             715.1
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      6.0482      2.324      2.602      0.011       1.435      10.661
X              1.7254      0.293      5.890      0.000       1.144       2.307
==============================================================================
Omnibus:                       23.982   Durbin-Watson:                   2.279
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               61.074
```

```
Skew:                          0.817   Prob(JB):                     5.47e-14
Kurtosis:                      6.463   Cond. No.                         22.4
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

```
plt.scatter(fitted_values, standardized_residuals)
plt.xlabel('Fitted Values')
plt.ylabel('Standardized Residuals')
plt.title('Residual Plot')
plt.show()
```

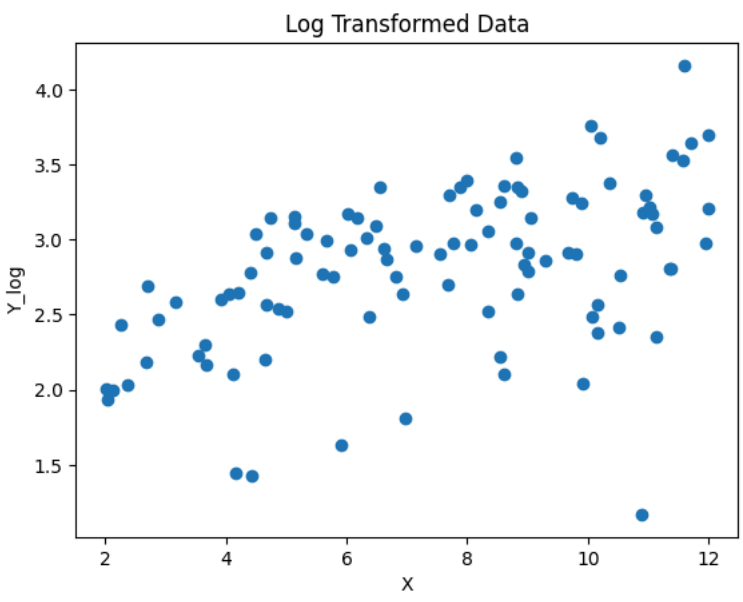


artificial1데이터에 단순 회귀 모형을 피팅하였더니 등분산성을 위배하는 잔차 양상이 관찰되었습니다. 이를 개선하기 위해 데이터 변환 기법, 가중 최소제곱법 등 적절한 통계적 기법을 동원하여 등분산성이 관측되도록 해주세요. (아래에 코드 작성해주세요)

```
# ans :

# 로그 변환
data2['Y_log'] = np.log(data2['Y'])

# 로그 변환된 데이터 시각화
plt.scatter(data2['X'], data2['Y_log'])
plt.xlabel('X')
plt.ylabel('Y_log')
plt.title('Log Transformed Data')
plt.show()
```



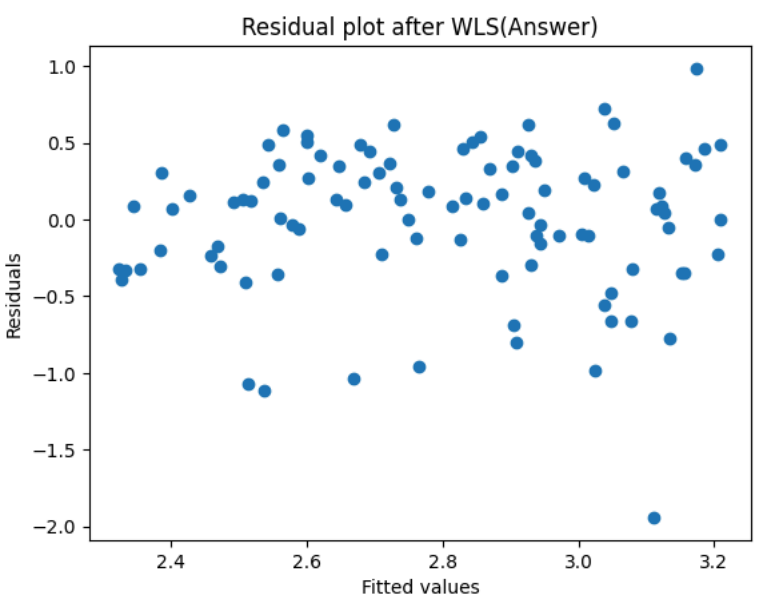

```
# 단순선형회귀 모델 생성
model_log2 = LinearRegression()
model_log2.fit(data2[['X']], data2['Y_log'])

# 로그 변환 데이터 - 예측값과 표준화 잔차 계산
fitted_values_log2 = model_log2.predict(data2[['X']])
standardized_residuals_log2 = (data2['Y_log'] - fitted_values_log2) / data2['Y_log'].std()
```

```
# 로그 변환된 데이터에 가중 최소제곱법(WLS) 적용
import statsmodels.api as sm

weights2 = 1 /standardized_residuals_log2**2
X2 = sm.add_constant(data2[['X']])
model_wls2 = sm.WLS(data2['Y_log'], X2, weights=weights2).fit()

# 잔차 그래프
plt.scatter(model_wls2.fittedvalues, model_wls2.resid)
plt.xlabel('Fitted values')
plt.ylabel('Residuals')
plt.title('Residual plot after WLS(Answer)')
plt.show()
```



Q3. 단순회귀모델에서 등분산성이 위배되는 것이 문제가 되는 이유가 무엇인지에 대해서 강의 내용을 바탕으로 서술하여주세요.

ans : 등분산성을 만족한다는 것은 회귀 모델이 다양한 X값에 대해 일관된 예측 정확도를 유지한다는 것이다. 등분산성이 위배되면 회귀 계수의 표준 오차 추정이 부정확해지기 때문에, 회귀 계수의 신뢰 구간의 부정확해지고, 회귀 계수의 p-value를 신뢰할 수 없게 된다.