예측애널리틱스 Homework #2

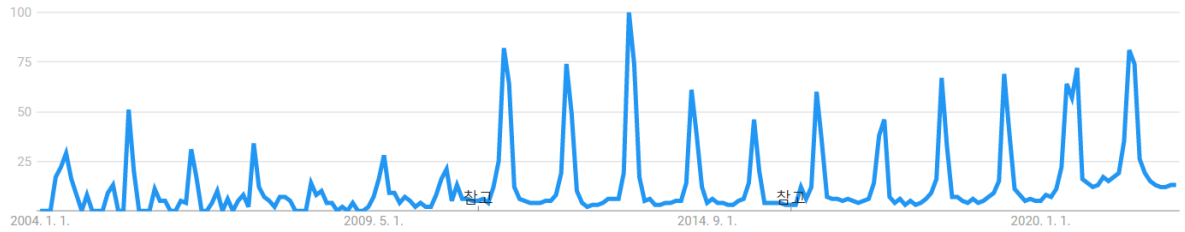
고려대학교
KOREA UNIVERSITY

대학	고려대학교 공과대학
학과	산업경영공학부
학번	2017170819
이름	박상민

1.첫 번째 데이터 '장마'

데이터에 대한 간단한 설명 및 기초통계량 분석

Google trend를 통해 seasonal variation이 있는 데이터 2가지를 선택했다. 첫 번째 데이터는 여름에 관심도가 집중되는 '장마' 검색어에 대한 데이터이다.



2010~2011 년 경부터 전체적인 검색량이 증가하고 있는 것을 알 수 있다. 이는 스마트폰의 보급 시기와 맞물리는데, 한국 사람들이 스마트폰을 사용하기 시작하면서 구글 사용량이 증가했다고 해석했다. 매년 초여름인 6~7 월에 검색량이 급증하고, 다른 달에는 검색량이 매우 적어진다.

```
import pandas as pd
data = pd.read_csv('rainfall.csv')
```

따라서 비교적 비슷한 패턴이 지속되는 2011 년 1 월부터 2022 년 3 월까지의 '장마' 검색어에 대한 관심도를 엑셀파일로 저장하였고, 이를 불러왔다. column 들은 time, rain, month 로 책정했고, time 은 말 그대로 2011 년 1 월을 1 로 설정해서 달마다 1 씩 증가하도록 하는 t 값이다. Rain 은 달마다 수집된 '장마' 검색어의 관심도이다. Month 는 12 개의 달을 할당한 column 이다.

```
data.describe()
```

데이터의 기초통계량을 구해보았다.

	time	rain
count	135.000000	135.000000
mean	68.000000	16.866667
std	39.115214	21.130299
min	1.000000	2.000000
25%	34.500000	5.000000
50%	68.000000	7.000000
75%	101.500000	16.500000
max	135.000000	100.000000

종속변수인 rain 의 평균값은 16.87, 표준편차는 21.13 이고, 최솟값은 2, 최댓값은 100 이다. 이때 흥미로운 점은 중간값이 7 이기 때문에 평균과 차이가 9 정도 난다는 점이다. 이는 장마라는 검색어의 관심도가 초여름에 매우 높게 나타났으며, 다른 달에는 비교적 낮게 나타났음을 의미한다.

Binary 변수 방법을 이용하여 모델링 하고 일부 데이터를 testing data로 사용하여 예측 성능을 평가

```
data=pd.get_dummies(data, columns=['month'])
data
```

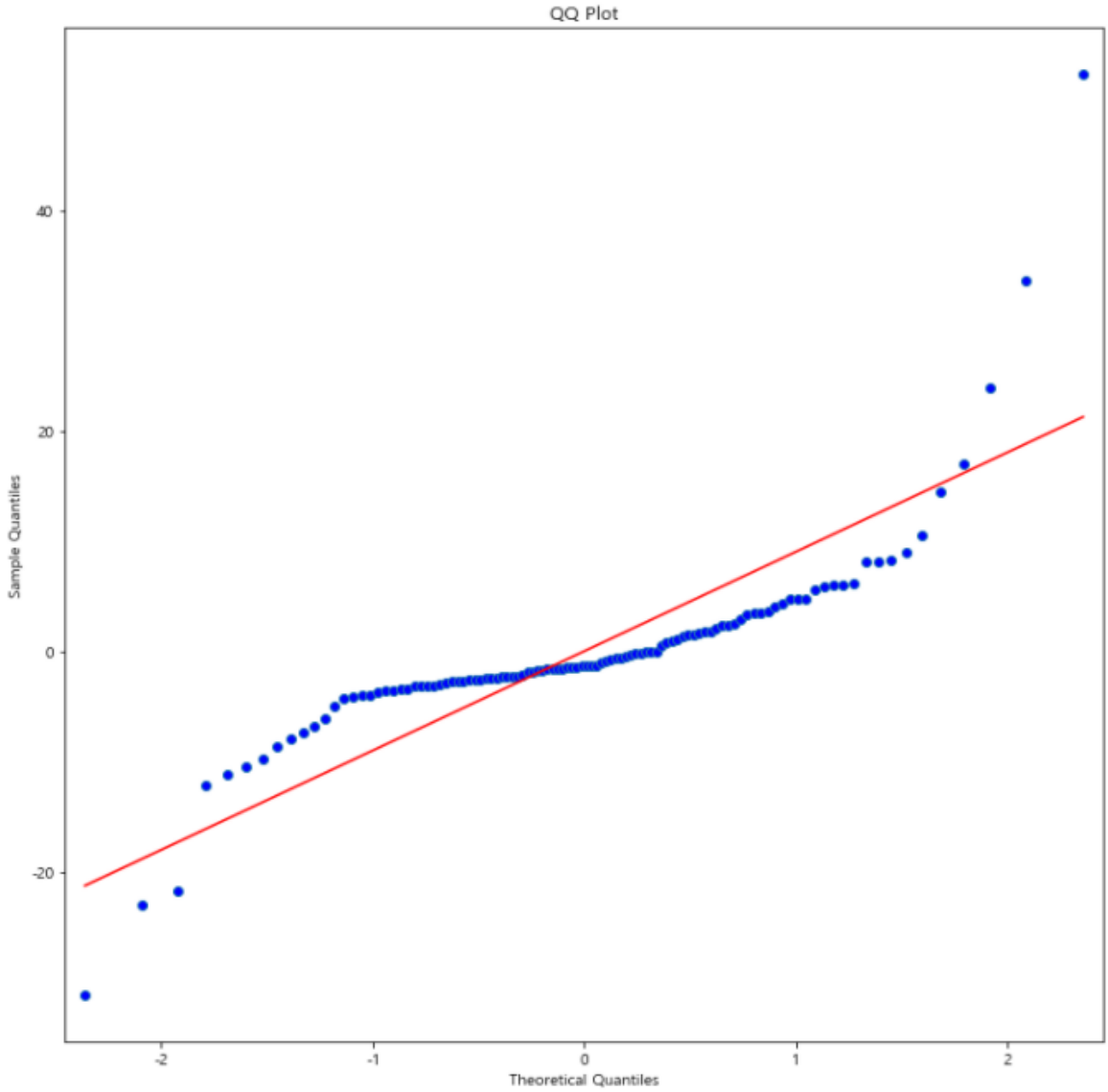
Binary 변수를 생성해서 seasonal variation을 모델링하고자 했다. 1년을 주기로 생각하면 12달이기 때문에 11개의 binary 변수가 필요하다. Get_dummies를 이용해 범주형 변수인 month를 binary 변수로 만들면 총 12개의 binary 변수가 만들어진다. 변수는 11개만 필요하기 때문에 12월을 기준으로 하고, 만약 data instance가 12월이면 모든 binary 변수가 0이 된다. 따라서 12월에 해당하는 binary 변수를 삭제했다.

	time	rain	month_Apr	month_Aug	month_Feb	month_Jan	month_Jul	month_Jun	month_Mar	month_May	month_Nov	month_Oct	month_Sep
0	1	5	0	0	0	1	0	0	0	0	0	0	0
1	2	6	0	0	1	0	0	0	0	0	0	0	0
2	3	4	0	0	0	0	0	0	1	0	0	0	0
3	4	12	1	0	0	0	0	0	0	0	0	0	0
4	5	25	0	0	0	0	0	0	0	1	0	0	0
5	6	82	0	0	0	0	0	1	0	0	0	0	0
6	7	64	0	0	0	0	1	0	0	0	0	0	0
7	8	12	0	1	0	0	0	0	0	0	0	0	0
8	9	6	0	0	0	0	0	0	0	0	0	0	1
9	10	5	0	0	0	0	0	0	0	0	0	1	0
10	11	4	0	0	0	0	0	0	0	0	1	0	0

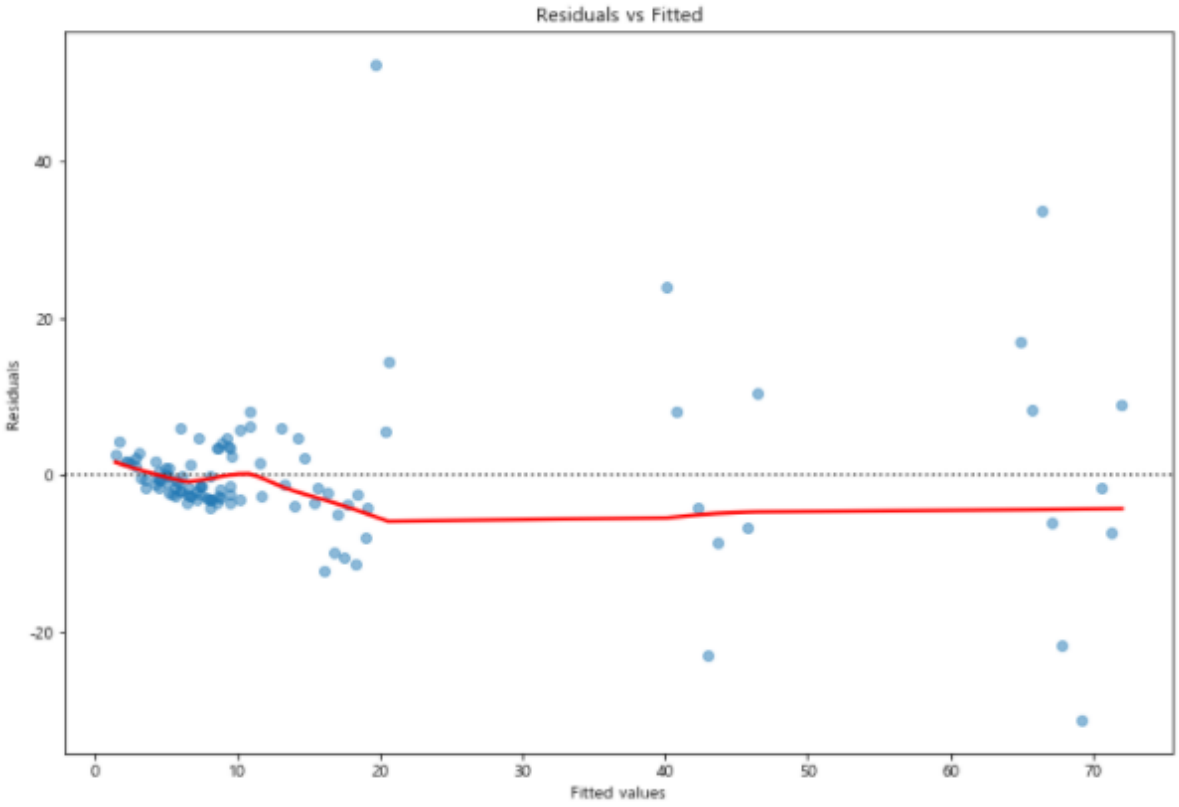
위에서 전처리한 데이터를 기반으로 종속변수 'rain'을 target variable로 하는 다중회귀모델을 구축하였다. 이 때 패키지는 statsmodels의 OLS(Ordinary Least Square)를 사용하였다. 또한 차후 회귀모델의 예측 정확도를 알아보기 위해 sklearn의 train_test_split을 사용하여 테스트 데이터의 크기를 0.2로 설정하였다.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=2021)
```

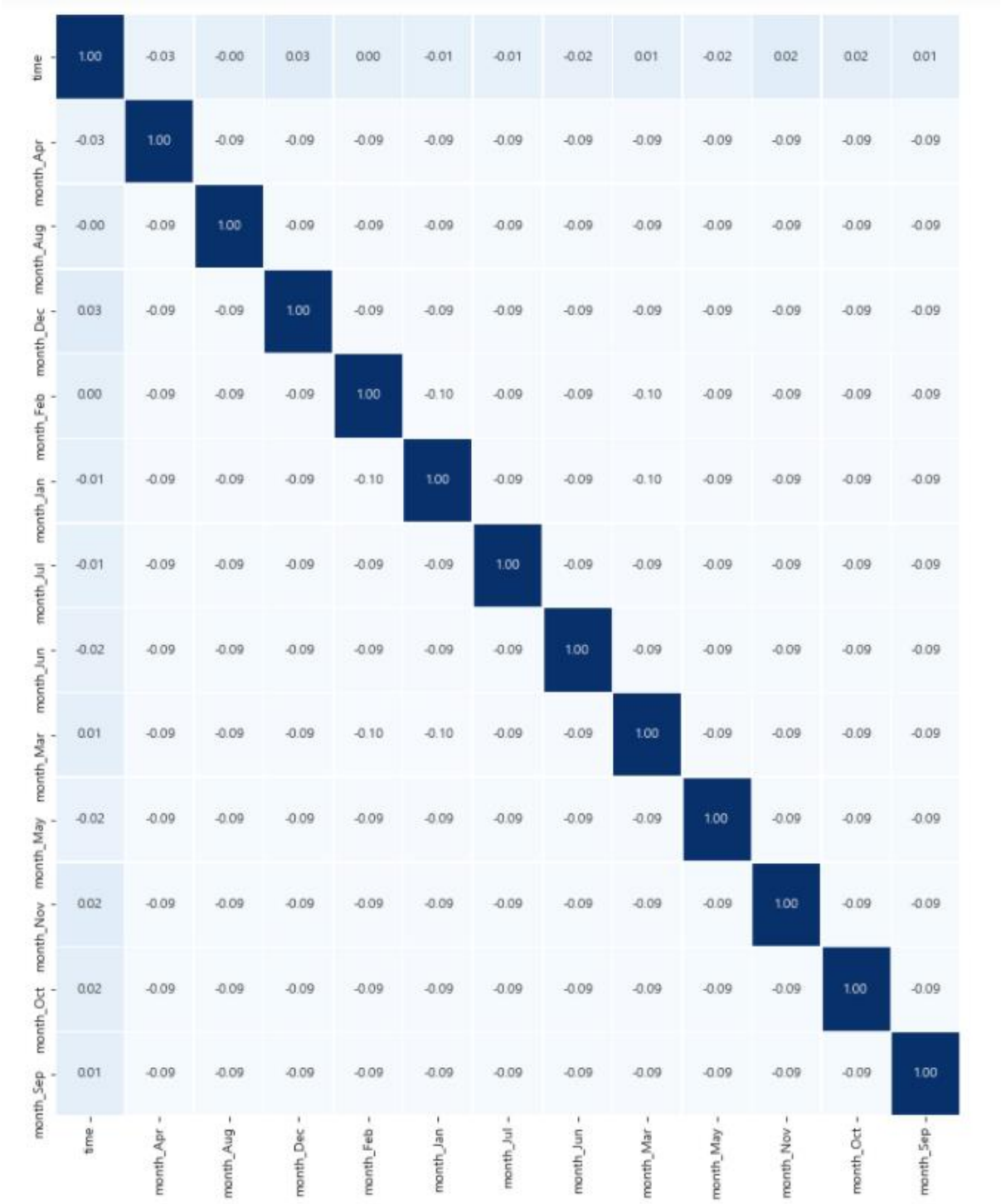
```
X_train = sm.add_constant(X_train)
model = sm.OLS(y_train, X_train, axis=1)
model_trained = model.fit()
```



가로축은 이론적인 quantile, 세로축은 관측치의 quantile이다. 정규성을 만족하기 위해서는 이론적인 quantile 값 기준으로 -2에서 2 사이에서 일직선에 따라 실제 관측치가 위치해야 한다. 해당 데이터는 정규 분포에 근사하다고 정성적으로 판단할 수 있다.



가로축은 추정 y 값, 세로축은 잔차이다. 등분산성을 띤다는 것은 y 추정 값의 변동성이 특정 변수의 변화에 영향을 받지 않아야 한다는 것을 의미한다. 그래프에서 y 추정 값에 상관없이 잔차가 일정하게 분포하면 등분산성을 충족한다고 볼 수 있다. 해당 그래프에서는 평평하게 일직선을 따르는 경향을 보이기 때문에 등분산성을 만족한다고 할 수 있다.



독립변수들 간 다중공산성을 확인해보면, 모든 변수들의 상관관계가 매우 낮음을 알 수 있고, 이는 다중공산성이 없다고 해석할 수 있다.

OLS Regression Results

Dep. Variable:	rain	R-squared:	0.808
Model:	OLS	Adj. R-squared:	0.783
Method:	Least Squares	F-statistic:	33.21
Date:	Wed, 30 Mar 2022	Prob (F-statistic):	9.84e-29
Time:	01:14:09	Log-Likelihood:	-390.75
No. Observations:	108	AIC:	807.5
Df Residuals:	95	BIC:	842.4
Df Model:	12		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	11.6002	1.771	6.549	0.000	8.084	15.116
time	0.0590	0.025	2.393	0.019	0.010	0.108
month_Apr	-5.8798	3.251	-1.808	0.074	-12.335	0.575
month_Aug	1.2084	2.805	0.431	0.668	-4.361	6.777
month_Dec	-9.7806	3.085	-3.170	0.002	-15.905	-3.656
month_Feb	-10.0285	3.078	-3.258	0.002	-16.140	-3.917
month_Jan	-10.9215	2.929	-3.728	0.000	-16.737	-5.106
month_Jul	28.0975	3.454	8.136	0.000	21.241	34.954
month_Jun	52.9988	3.071	17.256	0.000	46.901	59.096
month_Mar	-8.0225	2.818	-2.847	0.005	-13.618	-2.427
month_May	1.5884	3.250	0.489	0.626	-4.863	8.040
month_Nov	-10.0154	3.262	-3.070	0.003	-16.492	-3.539
month_Oct	-9.3220	3.244	-2.874	0.005	-15.761	-2.883
month_Sep	-8.3225	2.942	-2.829	0.006	-14.163	-2.482

Omnibus:	70.538	Durbin-Watson:	1.987
Prob(Omnibus):	0.000	Jarque-Bera (JB):	726.260
Skew:	1.860	Prob(JB):	1.97e-158
Kurtosis:	15.147	Cond. No.	1.05e+18

변수 선택 방법 중 하나인 Backward selection을 통해 모델의 파라미터 중 P-value가 가장 높은 변수를 하나씩 제거하면서 성능을 비교하였다. 회귀분석 결과를 살펴보면 대부분의 변수가 유효하지만, month_Aug와 month_May는 t-test의 p-value가 매우 높기 때문에 유의한 변수가 아니다. 따라서 이들을 제거해주었다.

OLS Regression Results

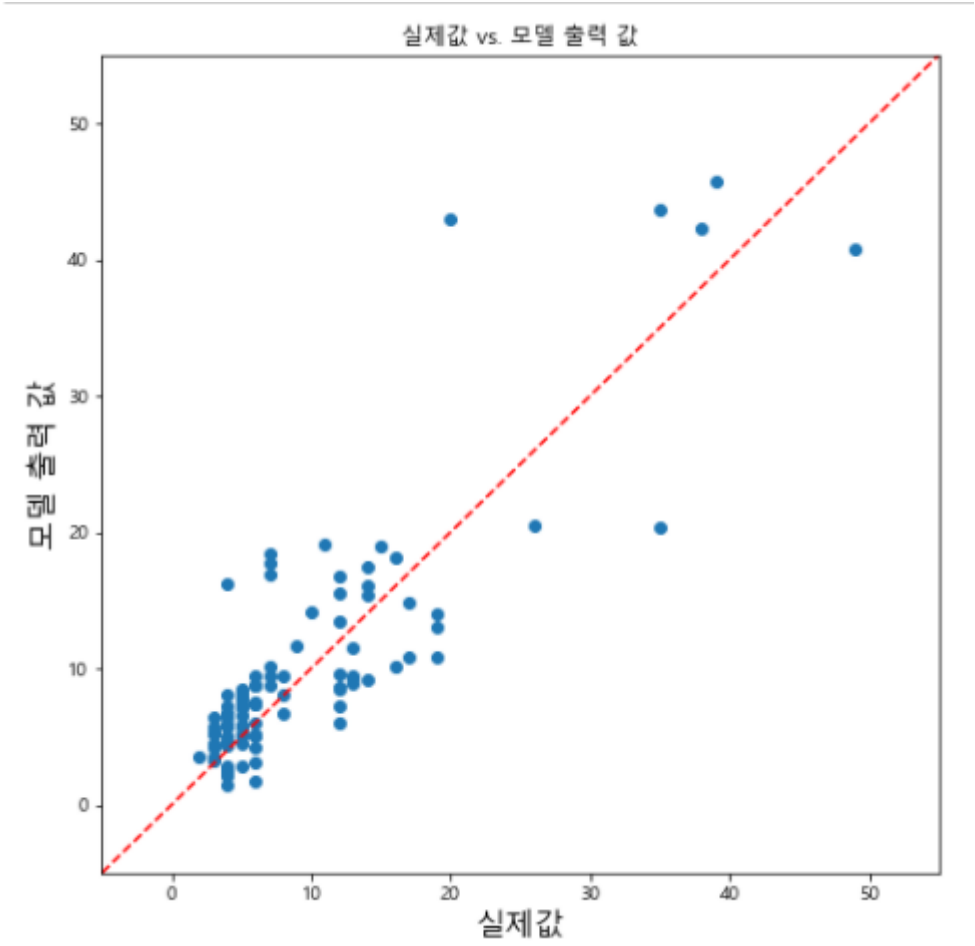
Dep. Variable:	rain	R-squared:	0.807
Model:	OLS	Adj. R-squared:	0.785
Method:	Least Squares	F-statistic:	36.61
Date:	Wed, 30 Mar 2022	Prob (F-statistic):	1.59e-29
Time:	01:15:27	Log-Likelihood:	-390.75
No. Observations:	108	AIC:	805.5
Df Residuals:	96	BIC:	837.7
Df Model:	11		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	12.9662	2.774	4.674	0.000	7.460	18.473
time	0.0590	0.025	2.407	0.018	0.010	0.108
month_Apr	-7.2476	4.049	-1.790	0.077	-15.286	0.790
month_Dec	-11.1492	3.872	-2.879	0.005	-18.836	-3.462
month_Feb	-11.3970	3.871	-2.945	0.004	-19.080	-3.714
month_Jan	-12.2899	3.737	-3.289	0.001	-19.707	-4.873
month_Jul	26.7295	4.237	6.308	0.000	18.319	35.140
month_Jun	51.6306	3.873	13.331	0.000	43.943	59.318
month_Mar	-9.3911	3.626	-2.590	0.011	-16.589	-2.193
month_Nov	-11.3841	4.035	-2.821	0.006	-19.394	-3.374
month_Oct	-10.6901	4.036	-2.649	0.009	-18.702	-2.679
month_Sep	-9.6911	3.739	-2.592	0.011	-17.113	-2.269

Omnibus:	69.926	Durbin-Watson:	1.990
Prob(Omnibus):	0.000	Jarque-Bera (JB):	711.755
Skew:	1.841	Prob(JB):	2.78e-155
Kurtosis:	15.025	Cond. No.	676.

month_Aug와 month_May를 제거한 후 다시 다중회귀분석을 해보면, 각 coefficient들의 t-test p-value는 0.1보다 다 작기 때문에 β=0이라는 귀무가설을 기각한다. 따라서 모든 변수는 유의하다고 해석할 수 있다. 이때 R square 값은 변하지 않았고, Adjusted R square값은 오히려 증가했기 때문에 적절한 변수제거라고 할 수 있다.

앞서 training set로 학습을 마쳤기 때문에 처음 보는 미래의 반응변수, 즉 test set의 반응변수 예측함으로써 일반화 성능을 평가하였다.



가로 축은 실제 값, 세로 축은 예측 값을 나타낸다. 점들이 일직선을 따라야 정확히 예측했다고 할 수 있는데, 어느 정도 정답과 유사한 패턴으로 예측이 되는 것을 정성적으로 판단하였다.

또한, 모델의 성능을 정량적으로 측정하기 위해 MSE(평균 제곱 오차), RMSE(제곱근 평균 제곱 오차), MAE(평균 절대 오차) 값을 아래와 같이 구했다. 정량적인 지표는 여러 모델을 비교하기 위한 기준으로 삼을 때 필요하다.

Mean Squared Error (평균 제곱 오차)

$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$

```
print(mean_squared_error(y_test, y_test_pred))
```

87.98324215409102

Root Mean Squared Error (제곱근 평균 제곱 오차)

$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$

```
print(np.sqrt(mean_squared_error(y_test, y_test_pred)))
```

9.379938280931864

Mean Absolute Error (평균 절대 오차)

$\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$

```
print(mean_absolute_error(y_test, y_test_pred))
```

5.4470042522169875

Mean Absolute Percentage Error (평균 절대 백분율 오차)

$\frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\%$

```
def mean_absolute_percentage_error(y_true, y_pred):
    y_true, y_pred = np.array(y_true), np.array(y_pred)
    return np.mean(np.abs((y_true-y_pred)/y_true))*100

print(mean_absolute_percentage_error(y_test, y_test_pred))
```

29.037836239440583

R squared (결정계수)

```
print(r2_score(y_test, y_test_pred))
```

0.8279586081865253

테스트 셋에 대한 $R2$ 값은 0.8280이며 새로운 데이터에 대해 82.80% 정도의 정확도를 보인다는 뜻이며, 매우 높은 수치이다.

최종결과정리

```
print('Training MSE: {:.3f}'.format(mean_squared_error(y_train, y_train_pred)))
print('Training RMSE: {:.3f}'.format(np.sqrt(mean_squared_error(y_train, y_train_pred))))
print('Training MAE: {:.3f}'.format(mean_absolute_error(y_train, y_train_pred)))
print('Training MAPE: {:.3f}'.format(mean_absolute_percentage_error(y_train, y_train_pred)))
print('Training R2: {:.3f}'.format(r2_score(y_train, y_train_pred)))
```

```
Training MSE: 81.311
Training RMSE: 9.017
Training MAE: 5.135
Training MAPE: 42.497
Training R2: 0.807
```

```
print('Testing MSE: {:.3f}'.format(mean_squared_error(y_test, y_test_pred)))
print('Testing RMSE: {:.3f}'.format(np.sqrt(mean_squared_error(y_test, y_test_pred))))
print('Testing MAE: {:.3f}'.format(mean_absolute_error(y_test, y_test_pred)))
print('Testing MAPE: {:.3f}'.format(mean_absolute_percentage_error(y_test, y_test_pred)))
print('Testing R2: {:.3f}'.format(r2_score(y_test, y_test_pred)))
```

```
Testing MSE: 87.983
Testing RMSE: 9.380
Testing MAE: 5.447
Testing MAPE: 29.038
Testing R2: 0.828
```

최종적으로 Training set과 test set에 대한 예측정확도 값을 비교해보았다. MSE, RMSE, MAE, MAPE 값이 작을수록, R square 값이 클수록 예측 성능이 좋다고 할 수 있다. training set에 비해 test set의 MSE, RMSE, MAE 값은 증가하였고, $R2$ 값은 감소하였다. training set으로 학습을 진행하였고, test set은 처음 보는 데이터이므로 모델의 예측 성능이 떨어지는 것은 당연한 결과이다.

Trigonometric 방법을 이용하여 모델링 하고 일부 데이터를 testing data로 사용하여 예측 성능을 평가

```
import pandas as pd
data = pd.read_csv('rainfall.csv')
```

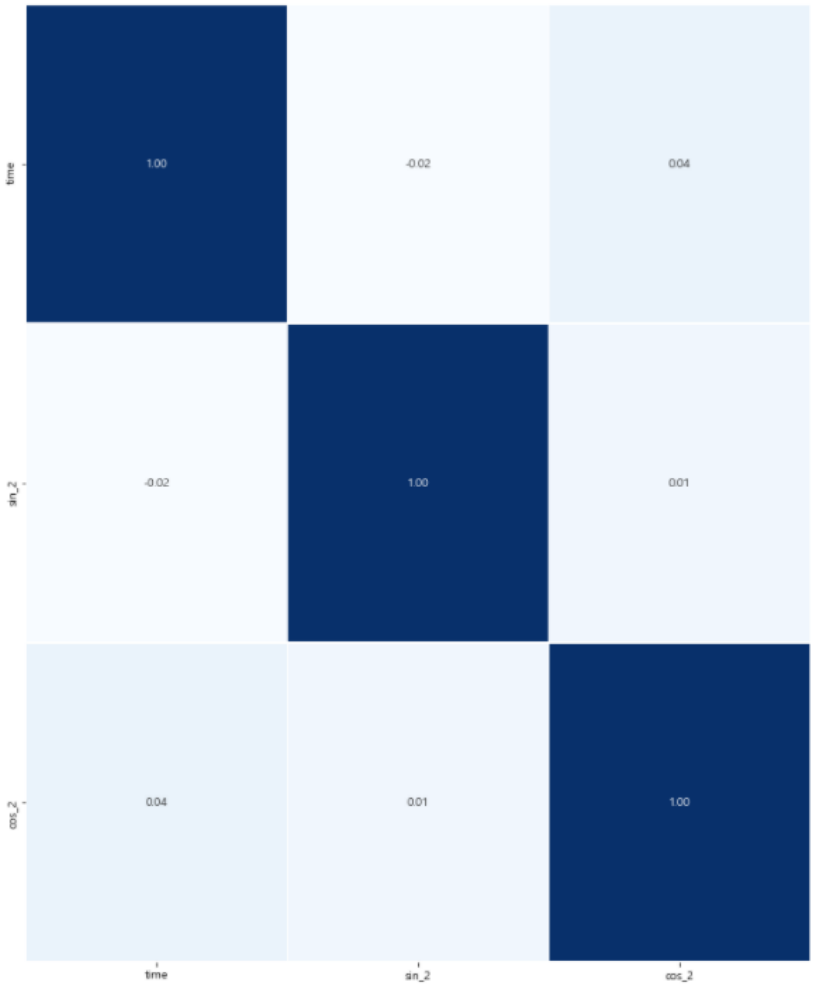
```
X = data.drop(['rain','month'],axis = 1)
y = data['rain']
```

```
X['sin_2'] = 0
X['cos_2'] = 0
```

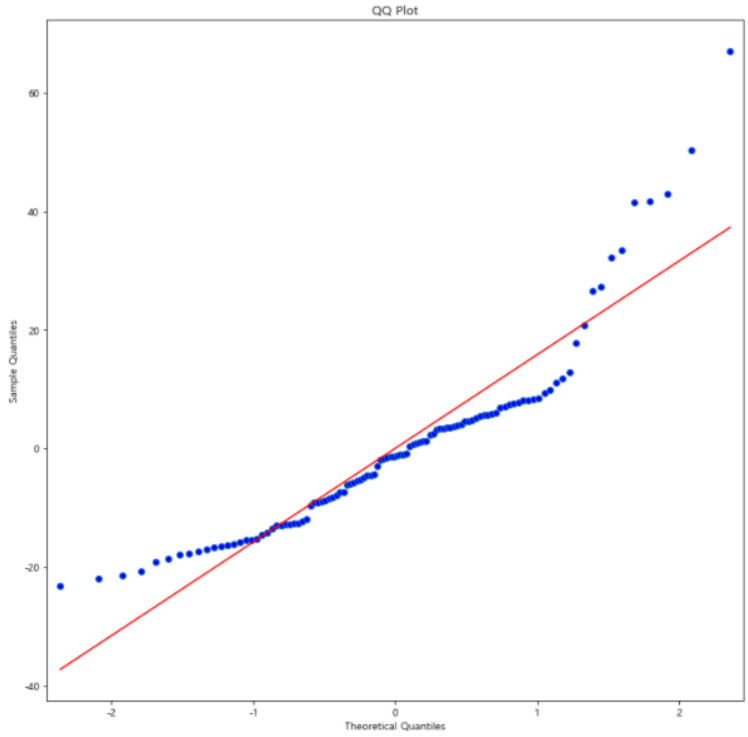
Trigonometric 방법을 이용하기 위해 다시 데이터를 불러온 후, X 변수에는 time column 만 남긴 후 $\sin(\frac{2\pi t}{L})$ 와 $\cos(\frac{2\pi t}{L})$ 를 x 변수로 추가해주었다. $\sin(\frac{2\pi t}{L})$ 와 $\cos(\frac{2\pi t}{L})$ 의 t에 time column의 값을 할당해주면 다음과 같이 데이터가 채워진다.

	time	sin_2	cos_2
0	1	0.500000	0.866025
1	2	0.866025	0.500000
2	3	1.000000	0.000000
3	4	0.866025	-0.500000
4	5	0.500000	-0.866025
5	6	0.000000	-1.000000
6	7	-0.500000	-0.866025
7	8	-0.866025	-0.500000
8	9	-1.000000	-0.000000
9	10	-0.866025	0.500000
10	11	-0.500000	0.866025
11	12	-0.000000	1.000000
12	13	0.500000	0.866025

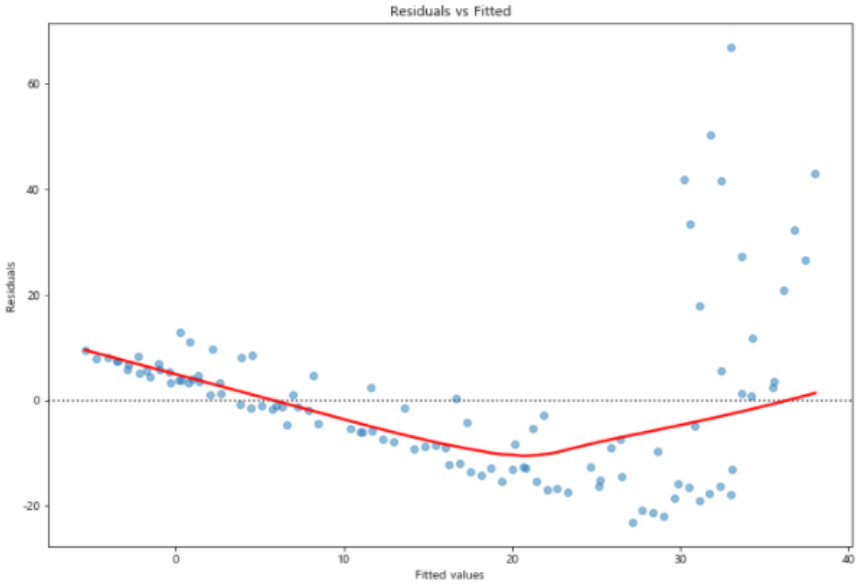
따라서 독립변수는 3개가 되고, 이를 통해 회귀분석을 진행할 수 있다.



x변수들 간 상관관계가 거의 없기 때문에 다중공산성은 없다.



가로축은 이론적인 quantile, 세로축은 관측치의 quantile이다. 정규성을 만족하기 위해서는 이론적인 quantile 값 기준으로 -2에서 2 사이에서 일직선에 따라 실제 관측치가 위치해야 한다. 해당 데이터는 정규 분포에 근사하다고 정성적으로 판단할 수 있다.



가로축은 추정 y 값, 세로축은 잔차이다. 등분산성을 띤다는 것은 y 추정 값의 변동성이 특정 변수의 변화에 영향을 받지 않아야 한다는 것을 의미한다. 그래프에서 y 추정 값에 상관없이 잔차가 일정하게 분포하면 등분산성을 충족한다고 볼 수 있다. 해당 그래프에서는 평평하게 일직선을 따르는 경향을 보이지 않기 때문에 등분산성을 만족한다고 보기 어렵다.

OLS Regression Results

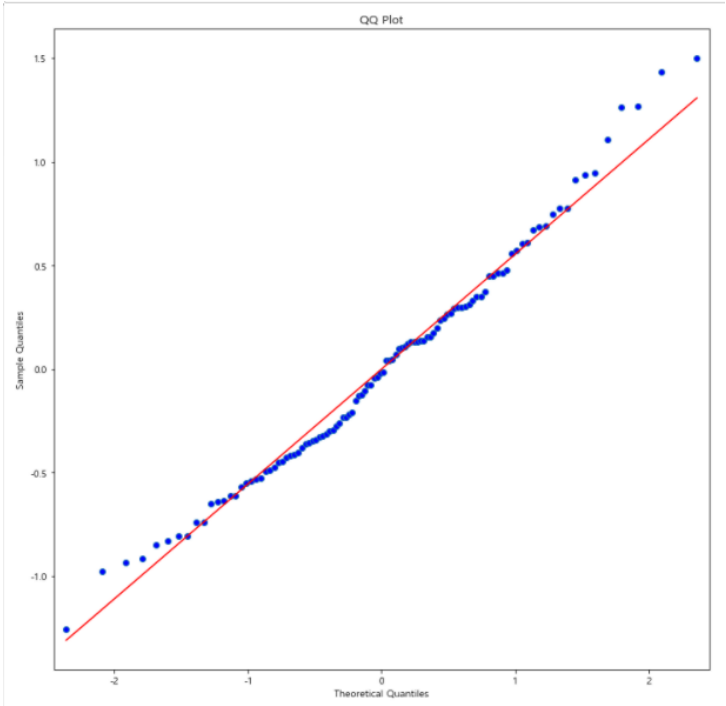
Dep. Variable:	rain	R-squared:	0.409
Model:	OLS	Adj. R-squared:	0.392
Method:	Least Squares	F-statistic:	23.97
Date:	Wed, 30 Mar 2022	Prob (F-statistic):	7.17e-12
Time:	12:55:23	Log-Likelihood:	-451.34
No. Observations:	108	AIC:	910.7
Df Residuals:	104	BIC:	921.4
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	12.7340	3.190	3.992	0.000	6.408	19.060
time	0.0520	0.041	1.276	0.205	-0.029	0.133
sin_2	-2.4422	2.149	-1.136	0.258	-6.704	1.819
cos_2	-18.7420	2.249	-8.332	0.000	-23.202	-14.282

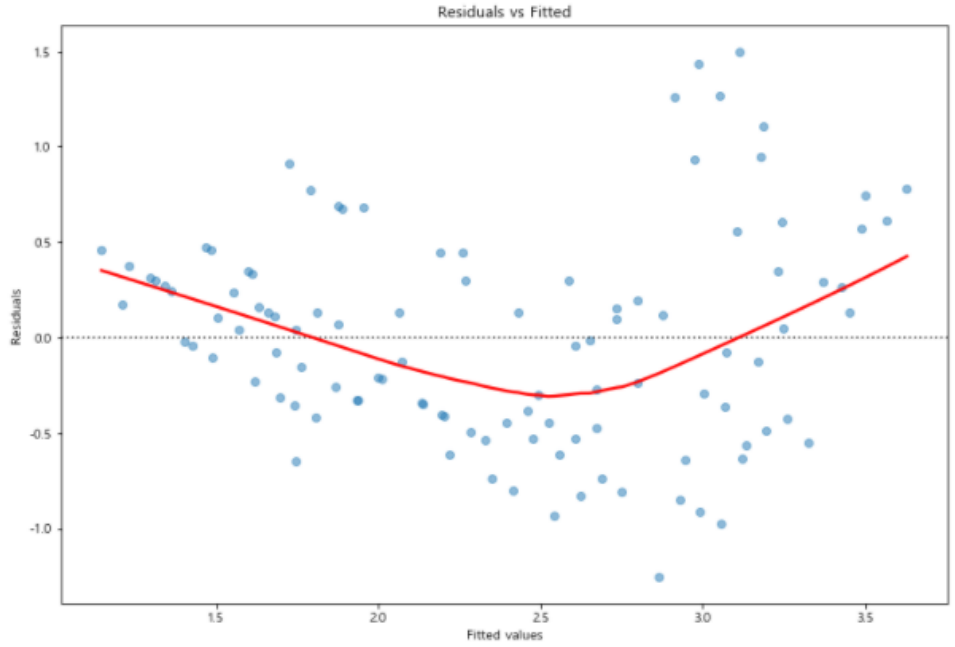
Omnibus:	41.151	Durbin-Watson:	1.937
Prob(Omnibus):	0.000	Jarque-Bera (JB):	89.325
Skew:	1.518	Prob(JB):	4.01e-20
Kurtosis:	6.260	Cond. No.	162.

회귀분석 결과를 살펴보아도 R square 값이 0.409로 낮은 편임을 알 수 있다.

따라서 등분산성을 만족시키기 위해 log transformation을 진행해봤다.



Log transformation을 하기 전보다 추세선에 더욱 가까운 모습으로, 정규성을 충분히 만족한다.



잔차가 일정하게 분포하고 평평하게 일직선을 따르는 경향을 보이기 때문에 등분산성을 만족한다고 할 수 있다.

OLS Regression Results

Dep. Variable:	rain	R-squared:	0.591
Model:	OLS	Adj. R-squared:	0.580
Method:	Least Squares	F-statistic:	50.16
Date:	Wed, 30 Mar 2022	Prob (F-statistic):	3.92e-20
Time:	12:56:40	Log-Likelihood:	-89.620
No. Observations:	108	AIC:	187.2
Df Residuals:	104	BIC:	198.0
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	2.0189	0.112	18.026	0.000	1.797	2.241
time	0.0054	0.001	3.742	0.000	0.003	0.008
sin_2	-0.0894	0.075	-1.185	0.239	-0.239	0.060
cos_2	-0.9353	0.079	-11.844	0.000	-1.092	-0.779

Omnibus:	3.002	Durbin-Watson:	1.846
Prob(Omnibus):	0.223	Jarque-Bera (JB):	2.874
Skew:	0.397	Prob(JB):	0.238
Kurtosis:	2.903	Cond. No.	162.

회귀분석 결과를 살펴보면 R square값도 상당히 증가한 것을 볼 수 있다. Sin_2 변수의 t-test p-value가 크기 때문에 제거하고 다시 분석해보면,

OLS Regression Results

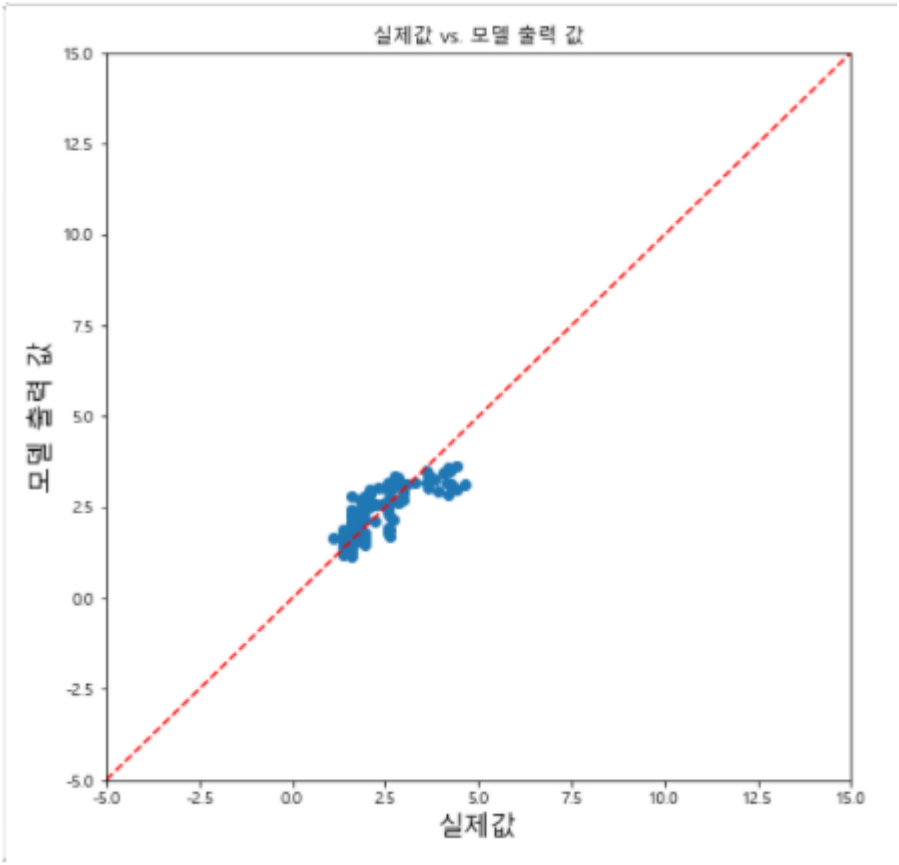
Dep. Variable:	rain	R-squared:	0.586
Model:	OLS	Adj. R-squared:	0.578
Method:	Least Squares	F-statistic:	74.25
Date:	Wed, 30 Mar 2022	Prob (F-statistic):	8.00e-21
Time:	13:05:42	Log-Likelihood:	-90.344
No. Observations:	108	AIC:	186.7
Df Residuals:	105	BIC:	194.7
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	2.0176	0.112	17.980	0.000	1.795	2.240
time	0.0054	0.001	3.743	0.000	0.003	0.008
cos_2	-0.9391	0.079	-11.878	0.000	-1.096	-0.782

Omnibus:	3.849	Durbin-Watson:	1.822
Prob(Omnibus):	0.146	Jarque-Bera (JB):	3.811
Skew:	0.453	Prob(JB):	0.149
Kurtosis:	2.838	Cond. No.	162.

R square와 adjusted R square에 큰 차이가 없기 때문에 좋은 변수 제거였다고 할 수 있다.

앞서 training set로 학습을 마쳤기 때문에 처음 보는 미래의 반응변수, 즉 test set의 반응변수 예측함으로써 일반화 성능을 평가하였다.



가로 축은 실제 값, 세로 축은 예측 값을 나타낸다. 점들이 일직선을 따라야 정확히 예측했다고 할 수 있는데, 어느 정도 정답과 유사한 패턴으로 예측이 되는 것을 정성적으로 판단하였다.

또한, 모델의 성능을 정량적으로 측정하기 위해 MSE(평균 제곱 오차), RMSE(제곱근 평균 제곱 오차), MAE(평균 절대 오차) 값을 아래와 같이 구했다. 정량적인 지표는 여러 모델을 비교하기 위한 기준으로 삼을 때 필요하다.

```
print(mean_squared_error(y_test, y_test_pred))
```
```
0.30966898815321914
```
```
print(np.sqrt(mean_squared_error(y_test, y_test_pred)))
```
```
0.5564790994756399
```
```
print(mean_absolute_error(y_test, y_test_pred))
```
```
0.43237084284289556
```
```
def mean_absolute_percentage_error(y_true, y_pred):
    y_true, y_pred = np.array(y_true), np.array(y_pred)
    return np.mean(np.abs((y_true-y_pred)/y_true))*100

print(mean_absolute_percentage_error(y_test, y_test_pred))
```
```
15.945488497142488
```
```
print(r2_score(y_test, y_test_pred))
```
```
0.6629576006703184
```

테스트 셋에 대한 $R2$ 값은 0.6630이며 새로운 데이터에 대해 66.30% 정도의 정확도를 보인다는 뜻이며, 양호한 수치이다.

```
print('Training MSE: {:.3f}'.format(mean_squared_error(y_train, y_train_pred)))
print('Training RMSE: {:.3f}'.format(np.sqrt(mean_squared_error(y_train, y_train_pred))))
print('Training MAE: {:.3f}'.format(mean_absolute_error(y_train, y_train_pred)))
print('Training MAPE: {:.3f}'.format(mean_absolute_percentage_error(y_train, y_train_pred)))
print('Training R2: {:.3f}'.format(r2_score(y_train, y_train_pred)))
```
```
Training MSE: 0.312
Training RMSE: 0.559
Training MAE: 0.457
Training MAPE: 19.976
Training R2: 0.586
```
```
print('Testing MSE: {:.3f}'.format(mean_squared_error(y_test, y_test_pred)))
print('Testing RMSE: {:.3f}'.format(np.sqrt(mean_squared_error(y_test, y_test_pred))))
print('Testing MAE: {:.3f}'.format(mean_absolute_error(y_test, y_test_pred)))
print('Testing MAPE: {:.3f}'.format(mean_absolute_percentage_error(y_test, y_test_pred)))
print('Testing R2: {:.3f}'.format(r2_score(y_test, y_test_pred)))
```
```
Testing MSE: 0.310
Testing RMSE: 0.556
Testing MAE: 0.432
Testing MAPE: 15.945
Testing R2: 0.663
```

최종적으로 Training set과 test set에 대한 예측정확도 값을 비교해보았다. MSE, RMSE, MAE, MAPE 값이 작을수록, R square 값이 클수록 예측 성능이 좋다고 할 수 있다. training set에 비해 test set의 MSE, RMSE, MAE, MAPE, R2 값이 증가하였으므로, 예측을 잘 하는 모델이라고 할 수 있다.

2.두 번째 데이터 '수영복'

데이터에 대한 간단한 설명 및 기초통계량 분석

Google trend를 통해 seasonal variation이 있는 데이터를 찾았는데, 2번째 데이터는 바로 '수영복' 검색어에 대한 관심도 데이터이다. 마찬가지로 2011년 1월부터 2022년 3월까지의 데이터를 사용했다. 7~8월 성수기철에 가장 검색량이 많은 것을 확인할 수 있었다.


```
import pandas as pd
data = pd.read_csv('swimsuit.csv')
```

column 들은 time, rain, month 로 책정했고, time 은 말 그대로 2011 년 1 월을 1 로 설정해서 달마다 1 씩 증가하도록 하는 t 값이다. Swimsuit 는 달마다 수집된 '장마' 검색어의 관심도이다. Month 는 12 개의 달을 할당한 column 이다.

```
data.describe()
```

	time	swimsuit
count	135.000000	135.000000
mean	68.000000	40.088889
std	39.115214	13.808102
min	1.000000	20.000000
25%	34.500000	31.000000
50%	68.000000	37.000000
75%	101.500000	46.500000
max	135.000000	100.000000

데이터의 기초통계량을 구해보았다. 종속변수인 swimsuit 의 평균값은 40.09, 표준편차는 13.81 이고, 최솟값은 20, 최댓값은 100 이다. 평균과 중간값이 비슷한 것으로 보아, 특정 달에 검색량이 편중된 정도가 작다고 해석할 수 있다.

Binary 변수 방법을 이용하여 모델링 하고 일부 데이터를 testing data로 사용하여 예측 성능을 평가

```
data=pd.get_dummies(data, columns=['month'])
data
```

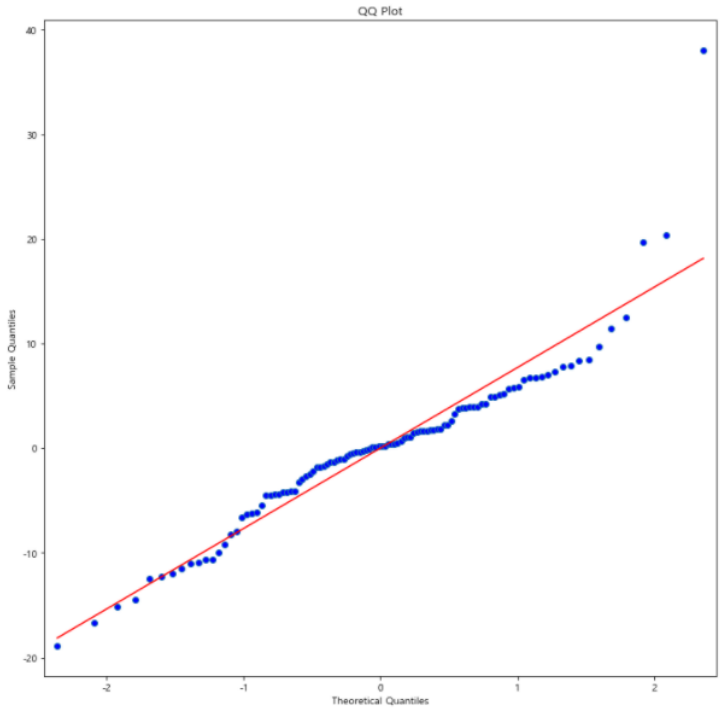
Binary 변수를 생성해서 seasonal variation을 모델링하고자 했다. 1년을 주기로 생각하면 12달이기 때문에 11개의 binary 변수가 필요하다. Get_dummies를 이용해 범주형 변수인 month를 binary 변수로 만들면 총 12개의 binary 변수가 만들어진다. 변수는 11개만 필요하기 때문에 12월을 기준으로 하고, 만약 data instance가 12월이면 모든 binary 변수가 0이 된다. 따라서 12월에 해당하는 binary 변수를 삭제했다.

	time	swimsuit	month_Apr	month_Aug	month_Feb	month_Jan	month_Jul	month_Jun	month_Mar	month_May	month_Nov	month_Oct	month_Sep
0	1	38	0	0	0	1	0	0	0	0	0	0	0
1	2	29	0	0	1	0	0	0	0	0	0	0	0
2	3	27	0	0	0	0	0	0	1	0	0	0	0
3	4	27	1	0	0	0	0	0	0	0	0	0	0
4	5	34	0	0	0	0	0	0	0	1	0	0	0
5	6	46	0	0	0	0	0	1	0	0	0	0	0
6	7	58	0	0	0	0	1	0	0	0	0	0	0
7	8	55	0	1	0	0	0	0	0	0	0	0	0
8	9	38	0	0	0	0	0	0	0	0	0	0	1
9	10	30	0	0	0	0	0	0	0	0	0	1	0
10	11	32	0	0	0	0	0	0	0	0	1	0	0
11	12	32	0	0	0	0	0	0	0	0	0	0	0

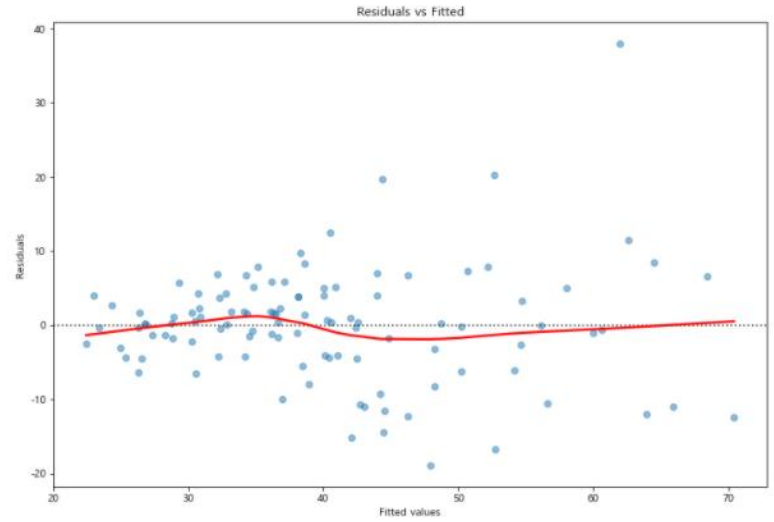
위에서 전처리한 데이터를 기반으로 종속변수 'swimsuit'을 target variable로 하는 다중회귀모델을 구축하였다. 이 때 패키지는 statsmodels의 OLS(Ordinary Least Square)를 사용하였다. 또한 차후 회귀모델의 예측 정확도를 알아보기 위해 sklearn의 train_test_split을 사용하여 테스트 데이터의 크기를 0.2로 설정하였다.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=2021)
```

```
X_train = sm.add_constant(X_train)
model = sm.OLS(y_train, X_train, axis=1)
model_trained = model.fit()
```

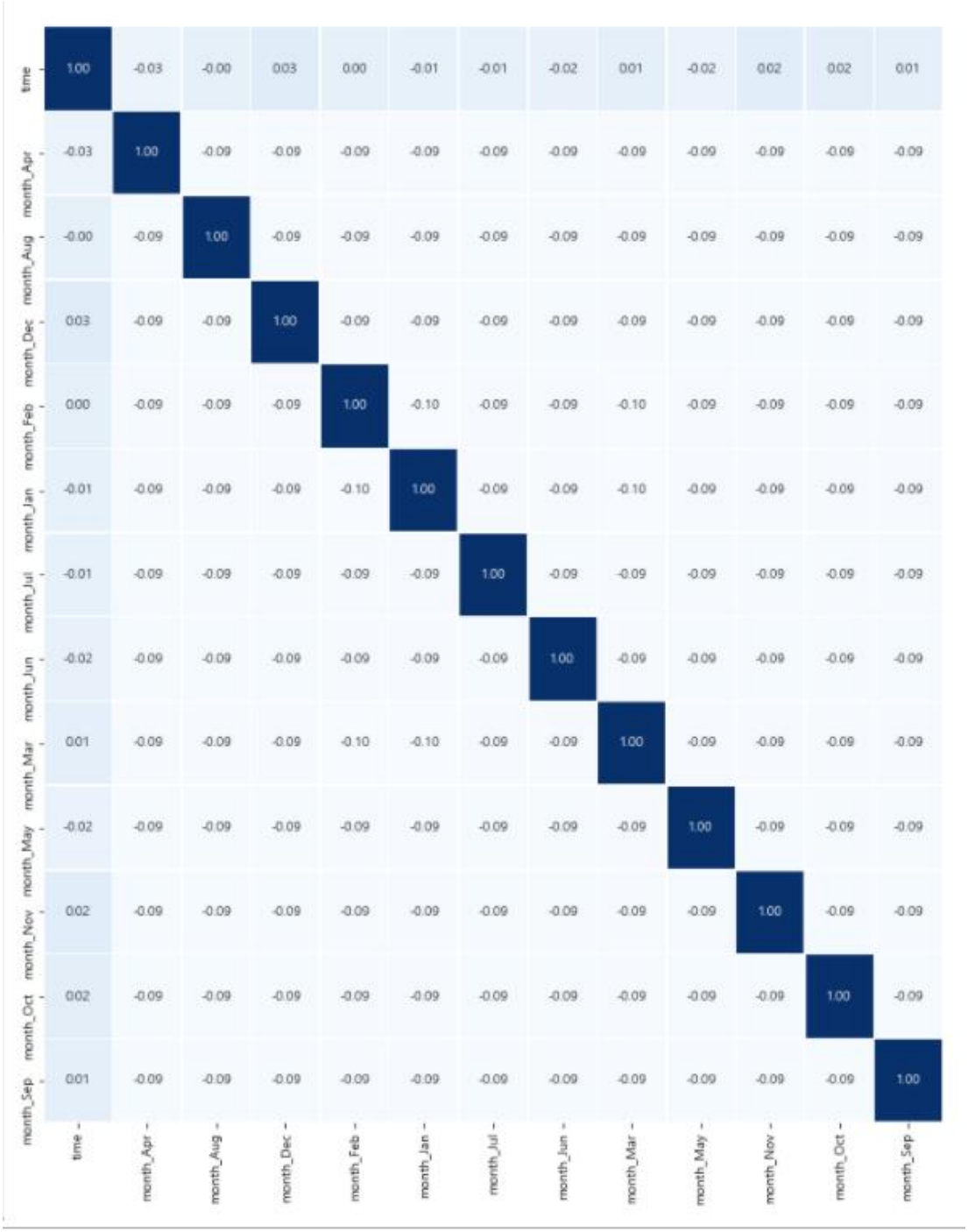


가로축은 이론적인 quantile, 세로축은 관측치의 quantile이다. 정규성을 만족하기 위해서는 이론적인 quantile 값 기준으로 -2에서 2 사이에서 일직선에 따라 실제 관측치가 위치해야 한다. 해당 데이터는 정규 분포에 근사하다고 정성적으로 판단할 수 있다.



가로축은 추정 y 값, 세로축은 잔차이다. 등분산성을 띤다는 것은 y 추정 값의 변동성이 특정 변수의 변화에 영향을 받지 않아야 한다는 것을 의미한다. 그래프에서 y 추정 값에 상관없이 잔차가 일정하게 분포하면 등분산성을 충족한다고 볼 수 있다. 해당 그래프에

서는 fitted value가 커질수록 분산이 커지는 보습을 보이기 때문에 등분산성을 만족한다고 보기 어렵다.



독립변수들 간 다중공산성을 확인해보면, 모든 x변수들 간 상관관계가 매우 낮음을 알 수 있고, 이는 다중공산성이 없다고 해석할 수 있다.

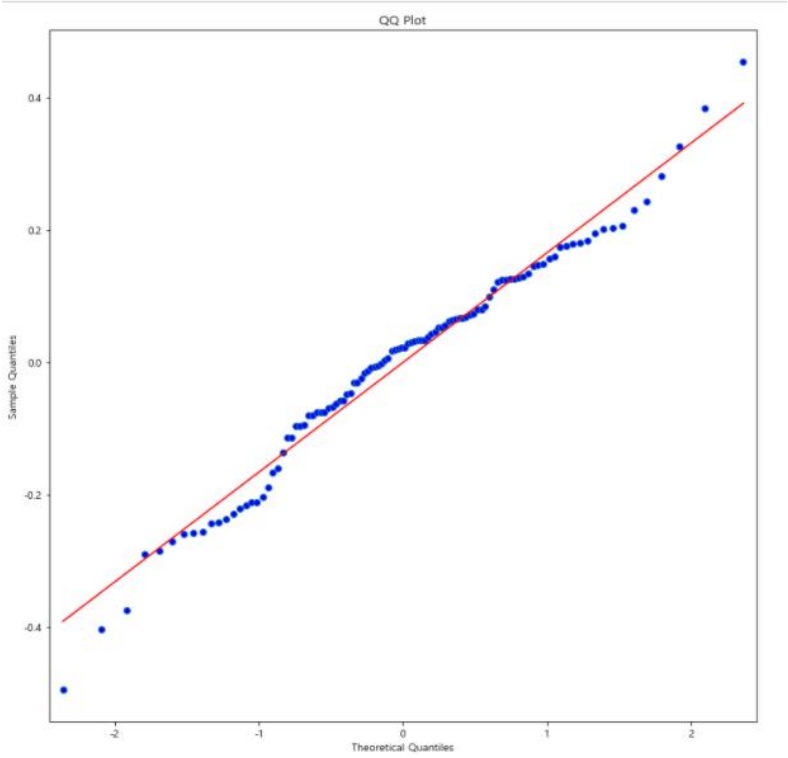
OLS Regression Results

Dep. Variable:	swimsuit	R-squared:	0.671
Model:	OLS	Adj. R-squared:	0.630
Method:	Least Squares	F-statistic:	16.15
Date:	Wed, 30 Mar 2022	Prob (F-statistic):	4.53e-18
Time:	13:55:22	Log-Likelihood:	-373.60
No. Observations:	108	AIC:	773.2
Df Residuals:	95	BIC:	808.1
Df Model:	12		
Covariance Type:	nonrobust		

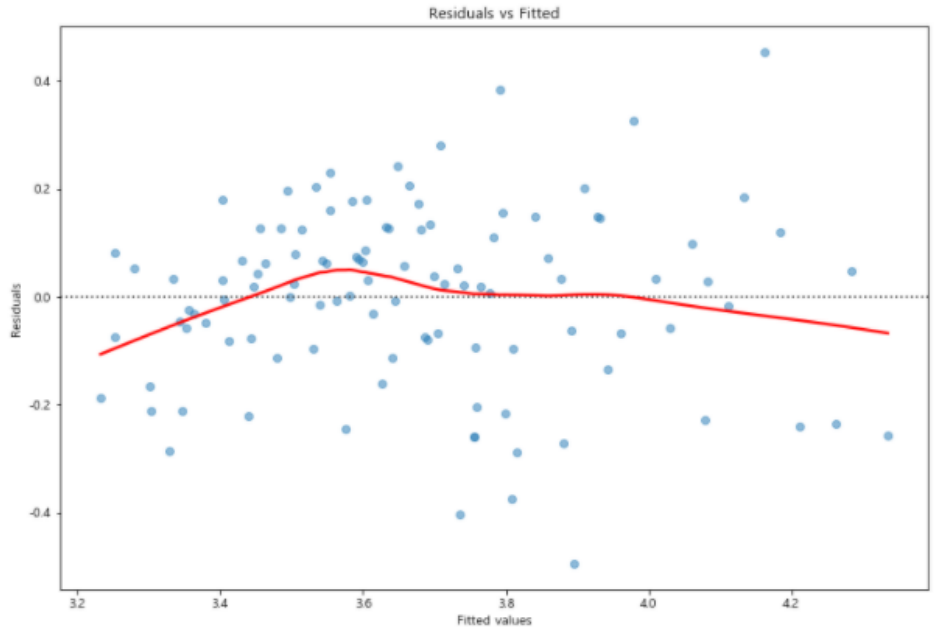
	coef	std err	t	P>|t|	[0.025	0.975]
const	47.4471	1.511	31.398	0.000	44.447	50.447
time	-0.1638	0.021	-7.791	0.000	-0.205	-0.122
month_Apr	-4.6856	2.774	-1.689	0.094	-10.193	0.821
month_Aug	19.7799	2.393	8.265	0.000	15.029	24.531
month_Dec	-2.4414	2.632	-0.928	0.356	-7.667	2.784
month_Feb	0.7905	2.626	0.301	0.764	-4.424	6.005
month_Jan	4.9218	2.499	1.969	0.052	-0.040	9.883
month_Jul	24.1215	2.946	8.187	0.000	18.272	29.971
month_Jun	10.1470	2.620	3.872	0.000	4.945	15.349
month_Mar	-0.9828	2.405	-0.409	0.684	-5.756	3.791
month_May	-0.4447	2.772	-0.160	0.873	-5.949	5.059
month_Nov	-2.9621	2.783	-1.064	0.290	-8.487	2.563
month_Oct	-1.3324	2.767	-0.481	0.631	-6.826	4.162
month_Sep	0.5354	2.510	0.213	0.832	-4.447	5.518

Omnibus:	32.916	Durbin-Watson:	2.143
Prob(Omnibus):	0.000	Jarque-Bera (JB):	122.402
Skew:	0.941	Prob(JB):	2.63e-27
Kurtosis:	7.864	Cond. No.	1.05e+18

회귀분석 결과를 살펴보아도 R square 값은 0.671로 양호하지만, t-test의 p-value가 너무 큰 변수들이 많다. 따라서 등분산성을 만족시키기 위해 log transformation을 진행해봤다.



Log transformation을 하기 전보다 추세선에 더욱 가까운 모습으로, 정규성을 충분히 만족한다.



잔차가 일정하게 분포하고 평평하게 일직선을 따르는 경향을 보이기 때문에 등분산성을 만족한다고 할 수 있다.

OLS Regression Results

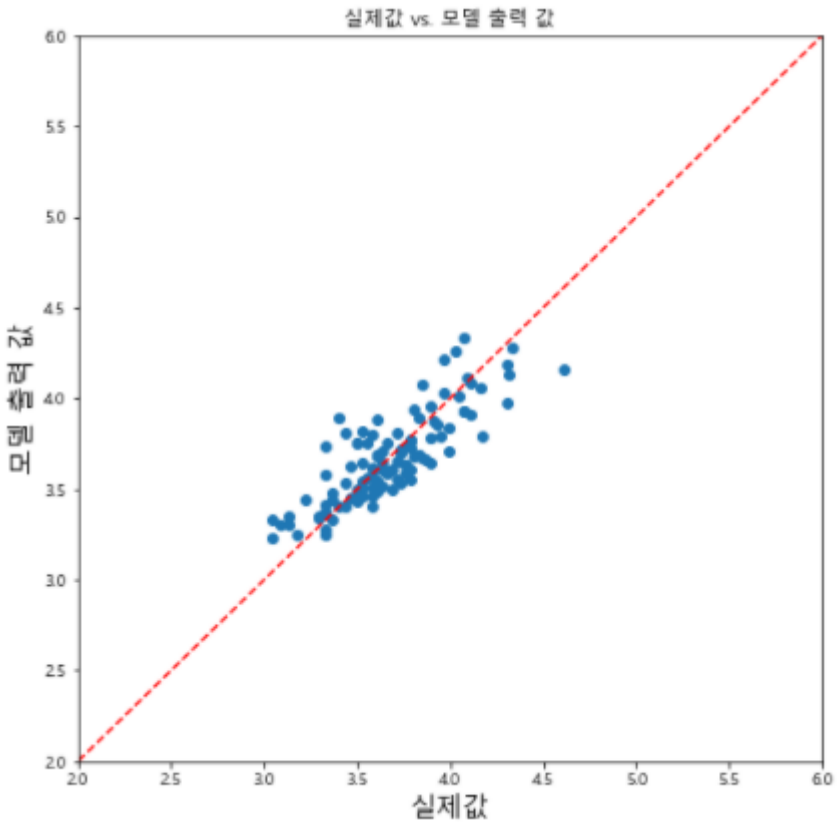
Dep. Variable:	swimsuit	R-squared:	0.696
Model:	OLS	Adj. R-squared:	0.658
Method:	Least Squares	F-statistic:	18.12
Date:	Wed, 30 Mar 2022	Prob (F-statistic):	1.28e-19
Time:	13:56:09	Log-Likelihood:	40.768
No. Observations:	108	AIC:	-55.54
Df Residuals:	95	BIC:	-20.67
Df Model:	12		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	3.6530	0.033	112.103	0.000	3.588	3.718
time	-0.0042	0.000	-9.245	0.000	-0.005	-0.003
month_Apr	0.0994	0.060	1.662	0.100	-0.019	0.218
month_Aug	0.6420	0.052	12.439	0.000	0.540	0.744
month_Dec	0.1526	0.057	2.688	0.008	0.040	0.265
month_Feb	0.2509	0.057	4.430	0.000	0.138	0.363
month_Jan	0.3434	0.054	6.371	0.000	0.236	0.450
month_Jul	0.7103	0.064	11.179	0.000	0.584	0.836
month_Jun	0.4509	0.057	7.979	0.000	0.339	0.563
month_Mar	0.1916	0.052	3.696	0.000	0.089	0.295
month_May	0.2177	0.060	3.641	0.000	0.099	0.336
month_Nov	0.1476	0.060	2.459	0.016	0.028	0.267
month_Oct	0.1969	0.060	3.300	0.001	0.078	0.315
month_Sep	0.2499	0.054	4.618	0.000	0.142	0.357

Omnibus:	2.641	Durbin-Watson:	2.176
Prob(Omnibus):	0.267	Jarque-Bera (JB):	2.058
Skew:	-0.303	Prob(JB):	0.357
Kurtosis:	3.300	Cond. No.	1.05e+18

회귀분석 결과를 살펴보면 R square값과 adjusted R square 값이 소폭 증가했음을 알 수 있고, 변수들도 대부분 t-test p-value가 작기 때문에 coefficient β=0이라는 귀무가설을 기각한다. 따라서 모든 변수는 유의하다고 해석할 수 있다. Month_Apr의 p-value가 0.05를 초과하지만, 0.1 이하이기 때문에 일단 제거하지 않기로 했다.

앞서 training set로 학습을 마쳤기 때문에 처음 보는 미래의 반응변수, 즉 test set의 반응변수 예측함으로써 일반화 성능을 평가하였다.



가로 축은 실제 값, 세로 축은 예측 값을 나타낸다. 점들이 일직선을 따라야 정확히 예측했다고 할 수 있는데, 어느 정도 정답과 유사한 패턴으로 예측이 되는 것을 정성적으로 판단하였다.

또한, 모델의 성능을 정량적으로 측정하기 위해 MSE(평균 제곱 오차), RMSE(제곱근 평균 제곱 오차), MAE(평균 절대 오차) 값을 아래와 같이 구했다. 정량적인 지표는 여러 모델을 비교하기 위한 기준으로 삼을 때 필요하다.

```
print(mean_squared_error(y_test, y_test_pred))
```

0.0465248215893496

```
print(np.sqrt(mean_squared_error(y_test, y_test_pred)))
```

0.2156961325322028

```
print(mean_absolute_error(y_test, y_test_pred))
```

0.17481121515930984

```
def mean_absolute_percentage_error(y_true, y_pred):
    y_true, y_pred = np.array(y_true), np.array(y_pred)
    return np.mean(np.abs((y_true-y_pred)/y_true))*100

print(mean_absolute_percentage_error(y_test, y_test_pred))
```

4.867163898817717

```
print(r2_score(y_test, y_test_pred))
```

0.607410267022586

테스트 셋에 대한 $R2$ 값은 0.6630이며 새로운 데이터에 대해 66.30% 정도의 정확도를 보인다는 뜻이며, 양호한 수치이다.

```
print('Training MSE: {:.3f}'.format(mean_squared_error(y_train, y_train_pred)))
print('Training RMSE: {:.3f}'.format(np.sqrt(mean_squared_error(y_train, y_train_pred))))
print('Training MAE: {:.3f}'.format(mean_absolute_error(y_train, y_train_pred)))
print('Training MAPE: {:.3f}'.format(mean_absolute_percentage_error(y_train, y_train_pred)))
print('Training R2: {:.3f}'.format(r2_score(y_train, y_train_pred)))
```

```
Training MSE: 0.028
Training RMSE: 0.166
Training MAE: 0.129
Training MAPE: 3.528
Training R2: 0.696
```

```
print('Testing MSE: {:.3f}'.format(mean_squared_error(y_test, y_test_pred)))
print('Testing RMSE: {:.3f}'.format(np.sqrt(mean_squared_error(y_test, y_test_pred))))
print('Testing MAE: {:.3f}'.format(mean_absolute_error(y_test, y_test_pred)))
print('Testing MAPE: {:.3f}'.format(mean_absolute_percentage_error(y_test, y_test_pred)))
print('Testing R2: {:.3f}'.format(r2_score(y_test, y_test_pred)))
```

```
Testing MSE: 0.047
Testing RMSE: 0.216
Testing MAE: 0.175
Testing MAPE: 4.867
Testing R2: 0.607
```

최종적으로 Training set과 test set에 대한 예측정확도 값을 비교해보았다. MSE, RMSE, MAE, MAPE 값이 작을수록, R square 값이 클수록 예측 성능이 좋다고 할 수 있다. training set에 비해 test set의 MSE, RMSE, MAE, MAPE값이 증가하였고, R2 값은 감소하였다. training set으로 학습을 진행하였고, test set은 처음 보는 데이터이므로 모델의 예측 성능이 떨어지는 것은 당연한 결과이다.

Trigonometric 방법을 이용하여 모델링 하고 일부 데이터를 testing data로 사용하여 예측 성능을 평가

```
import pandas as pd
data = pd.read_csv('swimsuit.csv')
```

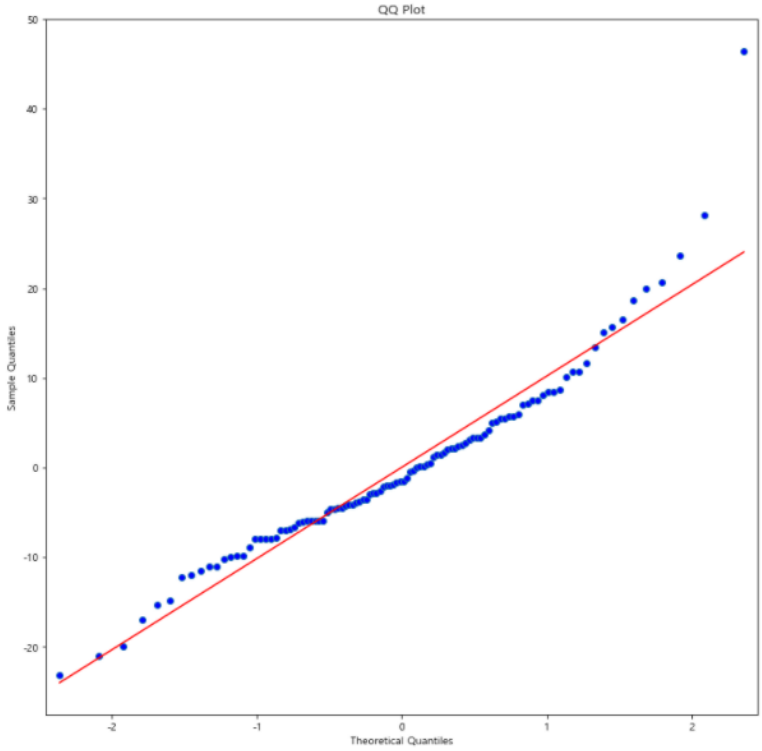
```
X = data.drop(['swimsuit','month'],axis = 1)
y = data['swimsuit']
```

```
X['sin_2'] = 0
X['cos_2'] = 0
```

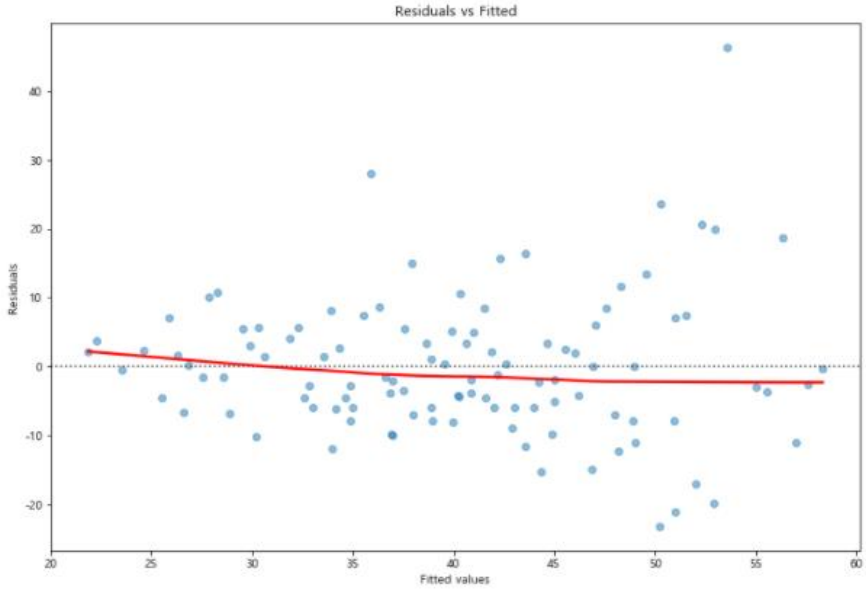
Trigonometric 방법을 이용하기 위해 다시 데이터를 불러온 후, X 변수에는 time column 만 남긴 후 $\sin(\frac{2\pi t}{12})$ 와 $\cos(\frac{2\pi t}{12})$ 를 x 변수로 추가해주었다. $\sin(\frac{2\pi t}{12})$ 와 $\cos(\frac{2\pi t}{12})$ 의 t에 time column의 값을 할당해주면 다음과 같이 데이터가 채워진다.

	time	sin_2	cos_2
0	1	0.500000	0.866025
1	2	0.866025	0.500000
2	3	1.000000	0.000000
3	4	0.866025	-0.500000
4	5	0.500000	-0.866025
5	6	0.000000	-1.000000
6	7	-0.500000	-0.866025
7	8	-0.866025	-0.500000
8	9	-1.000000	-0.000000
9	10	-0.866025	0.500000
10	11	-0.500000	0.866025
11	12	-0.000000	1.000000

따라서 독립변수는 3개가 되고, 이를 통해 회귀분석을 진행할 수 있다.

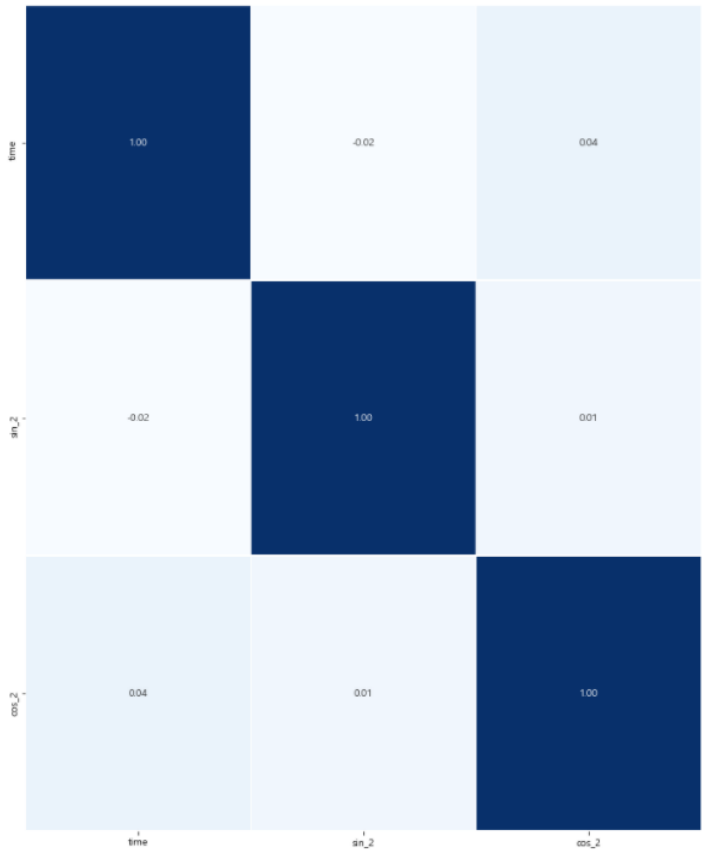


가로축은 이론적인 quantile, 세로축은 관측치의 quantile이다. 정규성을 만족하기 위해서는 이론적인 quantile 값 기준으로 -2에서 2 사이에서 일직선에 따라 실제 관측치가 위치해야 한다. 해당 데이터는 정규 분포에 근사하다고 정성적으로 판단할 수 있다.



가로축은 추정 y 값, 세로축은 잔차이다. 등분산성을 띤다는 것은 y 추정 값의 변동성이 특정 변수의 변화에 영향을 받지 않아야 한다는 것을 의미한다. 그래프에서 y 추정 값에

상관없이 잔차가 일정하게 분포하면 등분산성을 충족한다고 볼 수 있다. 해당 그래프에서는 나름 평평하게 일직선을 따르는 경향을 보이고 있다. 하지만 fitted value가 커질수록 분산이 커지는 양상을 보이고 있다. 따라서 등분산성을 가진다고 확신하기는 어렵다.



독립변수들 간 다중공산성을 확인해보면, 모든 변수들의 상관관계가 매우 낮음을 알 수 있고, 이는 다중공산성이 없다고 해석할 수 있다.

OLS Regression Results

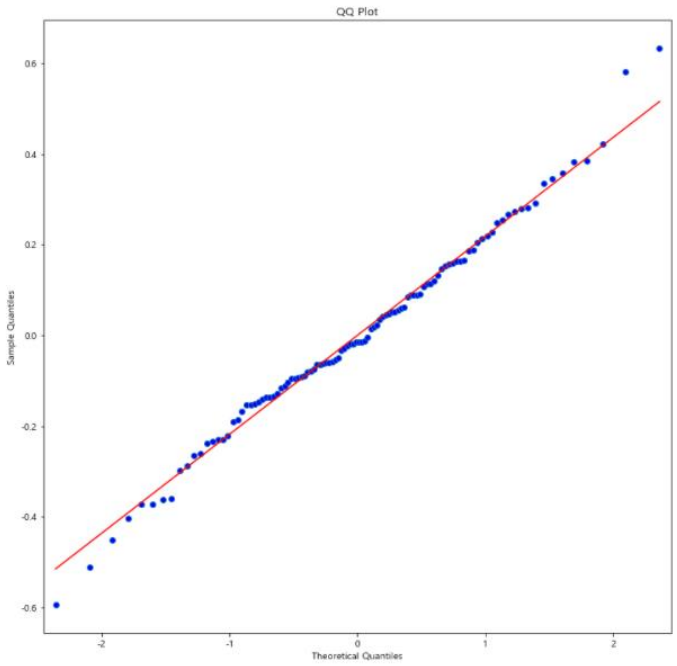
Dep. Variable:	swimsuit	R-squared:	0.424
Model:	OLS	Adj. R-squared:	0.407
Method:	Least Squares	F-statistic:	25.47
Date:	Wed, 30 Mar 2022	Prob (F-statistic):	1.95e-12
Time:	17:02:30	Log-Likelihood:	-403.89
No. Observations:	108	AIC:	815.8
Df Residuals:	104	BIC:	826.5
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	51.7901	2.056	25.192	0.000	47.713	55.867
time	-0.1670	0.026	-6.362	0.000	-0.219	-0.115
sin_2	-4.6424	1.385	-3.352	0.001	-7.389	-1.896
cos_2	-6.2199	1.450	-4.291	0.000	-9.094	-3.345

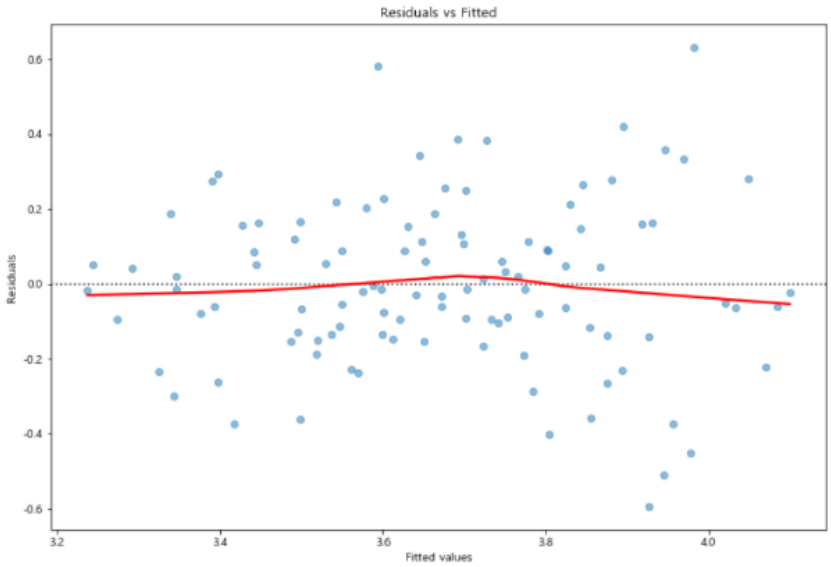
Omnibus:	30.962	Durbin-Watson:	1.926
Prob(Omnibus):	0.000	Jarque-Bera (JB):	71.422
Skew:	1.081	Prob(JB):	3.10e-16
Kurtosis:	6.347	Cond. No.	162.

회귀분석 결과를 살펴보아도 R square 값이 0.424로 매우 낮다. 따라서 다른 모델링 방법을 찾는 것이 중요하다.

등분산성 만족 여부가 확실하지 않기 때문에 log transformation을 해보았다.



Log transformation을 하기 전보다 추세선에 더욱 가까운 모습으로, 정규성을 충분히 만족한다.



잔차가 일정하게 분포하고 평평하게 일직선을 따르는 경향을 보이기 때문에 등분산성을 만족한다고 할 수 있다.

OLS Regression Results

Dep. Variable:	swimsuit	R-squared:	0.473
Model:	OLS	Adj. R-squared:	0.457
Method:	Least Squares	F-statistic:	31.06
Date:	Wed, 30 Mar 2022	Prob (F-statistic):	2.04e-14
Time:	17:03:09	Log-Likelihood:	11.020
No. Observations:	108	AIC:	-14.04
Df Residuals:	104	BIC:	-3.312
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	3.9655	0.044	89.903	0.000	3.878	4.053
time	-0.0042	0.001	-7.536	0.000	-0.005	-0.003
sin_2	-0.1008	0.030	-3.392	0.001	-0.160	-0.042
cos_2	-0.1309	0.031	-4.210	0.000	-0.193	-0.069

Omnibus:	1.202	Durbin-Watson:	1.976
Prob(Omnibus):	0.548	Jarque-Bera (JB):	0.712
Skew:	0.084	Prob(JB):	0.700
Kurtosis:	3.361	Cond. No.	162.

회귀 분석 결과를 보면 R square 값이 0.473이기 때문에 높은 편은 아니다. 따라서 더 세부적인 분석이 필요하다고 판단하고, Trigonometric model 2, 즉 $\sin(\frac{4\pi t}{L})$ 와 $\cos(\frac{4\pi t}{L})$ 를 변수로 추가해서 더욱 정교한 분석을 할 수 있다.

```
import pandas as pd
data = pd.read_csv('swimsuit.csv')
```

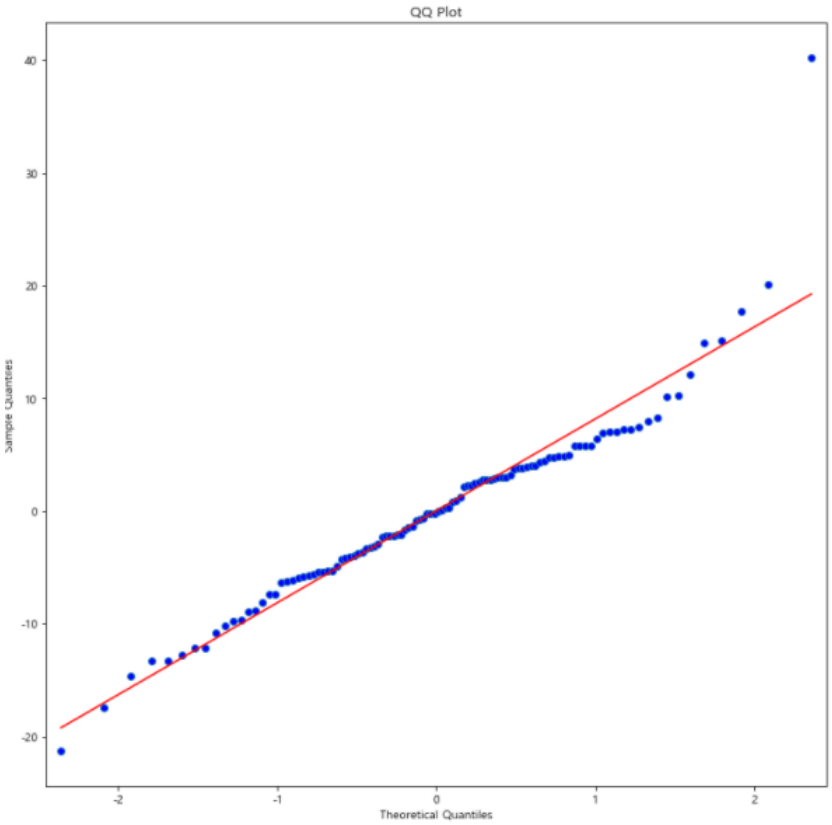
```
X = data.drop(['swimsuit','month'],axis = 1)
y = data['swimsuit']
```

```
X['sin_2'] = 0
X['cos_2'] = 0
X['sin_4'] = 0
X['cos_4'] = 0
```

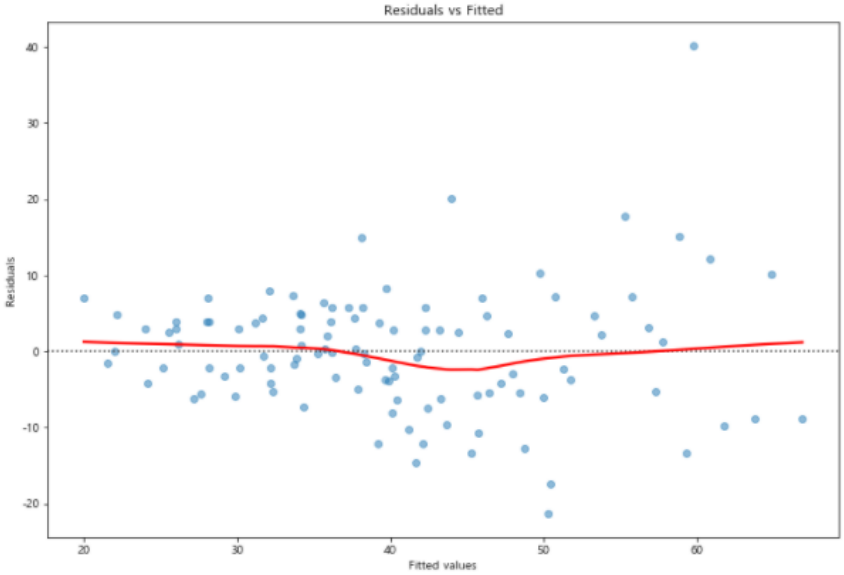
X 변수에는 time column만 남긴 후 $\sin(\frac{2\pi t}{L})$, $\cos(\frac{2\pi t}{L})$, $\sin(\frac{4\pi t}{L})$, $\cos(\frac{4\pi t}{L})$ 를 x 변수로 추가해주었다. $\sin(\frac{2\pi t}{L})$, $\cos(\frac{2\pi t}{L})$, $\sin(\frac{4\pi t}{L})$, $\cos(\frac{4\pi t}{L})$ 의 t에 time column의 값을 할당해주면 다음과 같이 데이터가 채워진다.

	time	sin_2	cos_2	sin_4	cos_4
0	1	0.500000	0.866025	0.866025	0.5
1	2	0.866025	0.500000	0.866025	-0.5
2	3	1.000000	0.000000	0.000000	-1.0
3	4	0.866025	-0.500000	-0.866025	-0.5
4	5	0.500000	-0.866025	-0.866025	0.5
5	6	0.000000	-1.000000	-0.000000	1.0
6	7	-0.500000	-0.866025	0.866025	0.5
7	8	-0.866025	-0.500000	0.866025	-0.5
8	9	-1.000000	-0.000000	0.000000	-1.0
9	10	-0.866025	0.500000	-0.866025	-0.5
10	11	-0.500000	0.866025	-0.866025	0.5
11	12	-0.000000	1.000000	-0.000000	1.0

따라서 독립변수는 5개가 되고, 이를 통해 회귀분석을 진행할 수 있다.



가로축은 이론적인 quantile, 세로축은 관측치의 quantile이다. 정규성을 만족하기 위해서는 이론적인 quantile 값 기준으로 -2에서 2 사이에서 일직선에 따라 실제 관측치가 위치해야 한다. 해당 데이터는 정규 분포에 근사하다고 정성적으로 판단할 수 있다.



가로축은 추정 y 값, 세로축은 잔차이다. 등분산성을 띤다는 것은 y 추정 값의 변동성이 특정 변수의 변화에 영향을 받지 않아야 한다는 것을 의미한다. 그래프에서 y 추정 값에 상관없이 잔차가 일정하게 분포하면 등분산성을 충족한다고 볼 수 있다. 해당 그래프에서는 평평하게 일직선을 따르는 경향을 보이고 있기 때문에 등분산성을 만족한다고 할 수 있다.

OLS Regression Results

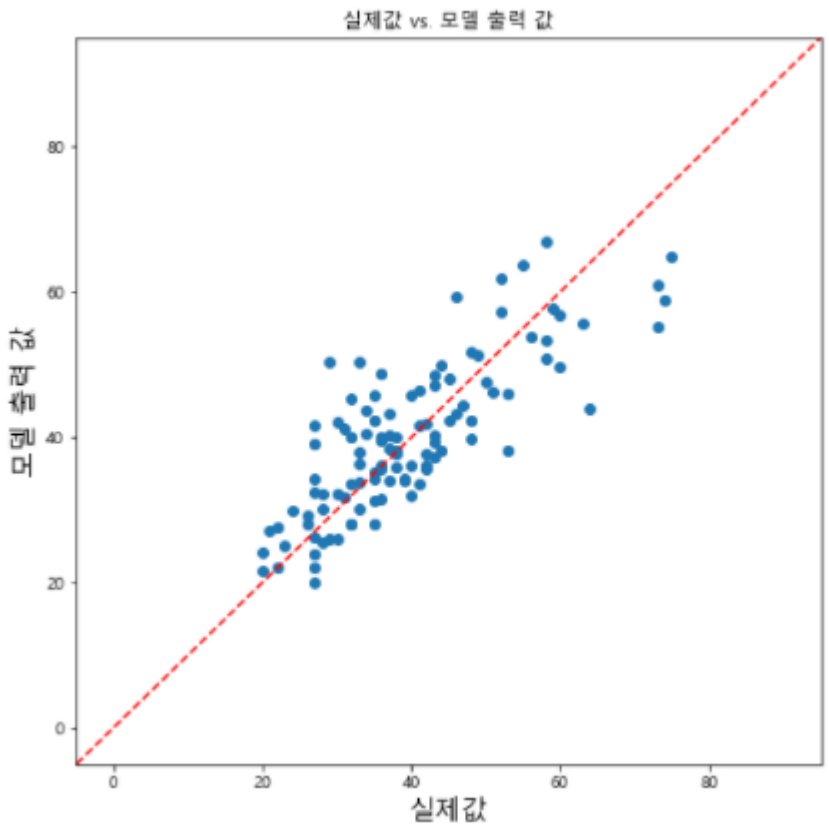
Dep. Variable:	swimsuit	R-squared:	0.630
Model:	OLS	Adj. R-squared:	0.612
Method:	Least Squares	F-statistic:	34.73
Date:	Wed, 30 Mar 2022	Prob (F-statistic):	1.38e-20
Time:	17:06:38	Log-Likelihood:	-379.96
No. Observations:	108	AIC:	771.9
Df Residuals:	102	BIC:	788.0
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	51.5969	1.664	31.013	0.000	48.297	54.897
time	-0.1678	0.021	-7.899	0.000	-0.210	-0.126
sin_2	-4.6215	1.121	-4.125	0.000	-6.844	-2.399
cos_2	-6.5156	1.174	-5.550	0.000	-8.844	-4.187
sin_4	8.5620	1.170	7.319	0.000	6.242	10.882
cos_4	2.2203	1.122	1.978	0.051	-0.006	4.447

Omnibus:	32.480	Durbin-Watson:	1.973
Prob(Omnibus):	0.000	Jarque-Bera (JB):	109.919
Skew:	0.966	Prob(JB):	1.35e-24
Kurtosis:	7.549	Cond. No.	162.

회귀분석 결과 모든 x변수들의 t-test p-value가 0.05를 크게 초과하지 않고, R square 값도 0.630으로 많이 높아졌다. Adjusted R square도 0.612로 높은 편이고, R square와 크게 차이 나지 않는다. 따라서 유의미한 회귀모델이라고 할 수 있다.

앞서 training set로 학습을 마쳤기 때문에 처음 보는 미래의 반응변수, 즉 test set의 반응변수 예측함으로써 일반화 성능을 평가하였다.



가로 축은 실제 값, 세로 축은 예측 값을 나타낸다. 점들이 일직선을 따라야 정확히 예측했다고 할 수 있는데, 어느 정도 정답과 유사한 패턴으로 예측이 되는 것을 정성적으로 판단하였다.

또한, 모델의 성능을 정량적으로 측정하기 위해 MSE(평균 제곱 오차), RMSE(제곱근 평균

제곱 오차), MAE(평균 절대 오차) 값을 아래와 같이 구했다. 정량적인 지표는 여러 모델을 비교하기 위한 기준으로 삼을 때 필요하다.

```
print(mean_squared_error(y_test, y_test_pred))
```

```
94.41969519955683
```

```
print(np.sqrt(mean_squared_error(y_test, y_test_pred)))
```

```
9.716979736500269
```

```
print(mean_absolute_error(y_test, y_test_pred))
```

```
7.588351611250624
```

```
def mean_absolute_percentage_error(y_true, y_pred):
    y_true, y_pred = np.array(y_true), np.array(y_pred)
    return np.mean(np.abs((y_true-y_pred)/y_true))*100

print(mean_absolute_percentage_error(y_test, y_test_pred))
```

```
19.688391732905778
```

```
print(r2_score(y_test, y_test_pred))
```

```
0.5826696872659552
```

테스트 셋에 대한 $R2$ 값은 0.583이며 새로운 데이터에 대해 58.3% 정도의 정확도를 보인다는 뜻이며, 양호한 수치이다.

```
print('Training MSE: {:.3f}'.format(mean_squared_error(y_train, y_train_pred)))
print('Training RMSE: {:.3f}'.format(np.sqrt(mean_squared_error(y_train, y_train_pred))))
print('Training MAE: {:.3f}'.format(mean_absolute_error(y_train, y_train_pred)))
print('Training MAPE: {:.3f}'.format(mean_absolute_percentage_error(y_train, y_train_pred)))
print('Training R2: {:.3f}'.format(r2_score(y_train, y_train_pred)))
```

```
Training MSE: 66.576
Training RMSE: 8.159
Training MAE: 5.997
Training MAPE: 15.090
Training R2: 0.630
```

```
print('Testing MSE: {:.3f}'.format(mean_squared_error(y_test, y_test_pred)))
print('Testing RMSE: {:.3f}'.format(np.sqrt(mean_squared_error(y_test, y_test_pred))))
print('Testing MAE: {:.3f}'.format(mean_absolute_error(y_test, y_test_pred)))
print('Testing MAPE: {:.3f}'.format(mean_absolute_percentage_error(y_test, y_test_pred)))
print('Testing R2: {:.3f}'.format(r2_score(y_test, y_test_pred)))
```

```
Testing MSE: 94.420
Testing RMSE: 9.717
Testing MAE: 7.588
Testing MAPE: 19.688
Testing R2: 0.583
```

최종적으로 Training set과 test set에 대한 예측정확도 값을 비교해보았다. MSE, RMSE, MAE, MAPE 값이 작을수록, R square 값이 클수록 예측 성능이 좋다고 할 수 있다. training set에 비해 test set의 MSE, RMSE, MAE, MAPE값이 증가하였고, R2 값은 감소하였다. training set으로 학습을 진행하였고, test set은 처음 보는 데이터이므로 모델의 예측 성능이 떨어지는 것은 당연한 결과이다.