

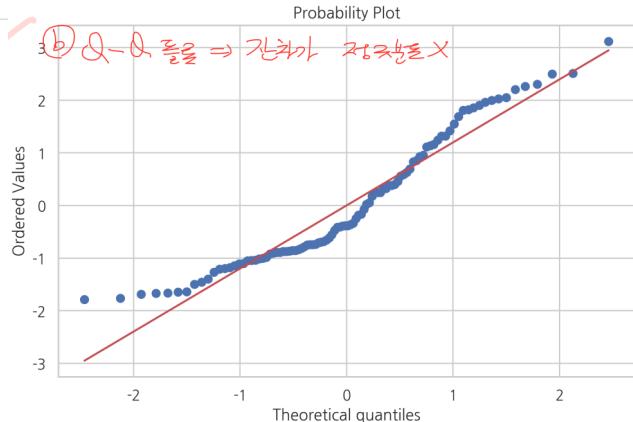
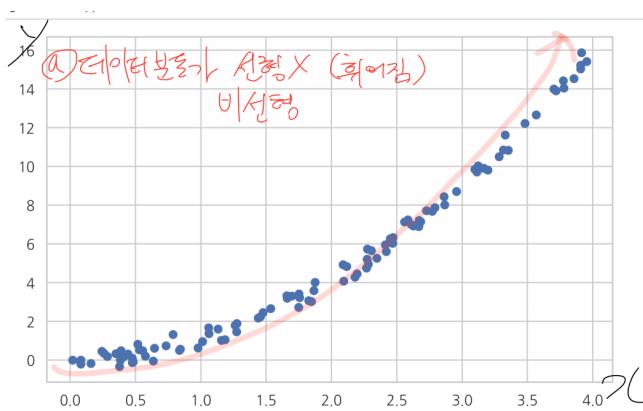
06.01 모형진단과 수정

- 제대로 된 모델인지, 아니라면 모델을 어떻게 수정할 것인지

모형 진단 : 사용된 데이터가 사용된 모형의 가정을 제대로 만족하고 있는지 확인 (모델의 가정을 진짜 가져가도 되는건지..)

1) 잔차 정규성 => 잔차가 정규성이 없다면, 기본 가정을 위배

- 잔차가 정규분포가 아니다. (데이터 분포가 비선형일 경우, 발생 가능한 현상)
(1-2page, a,b)

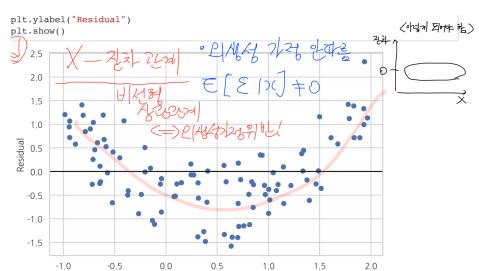
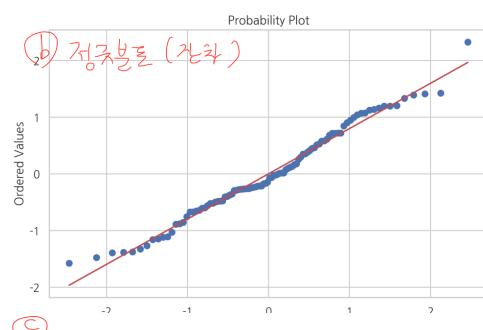
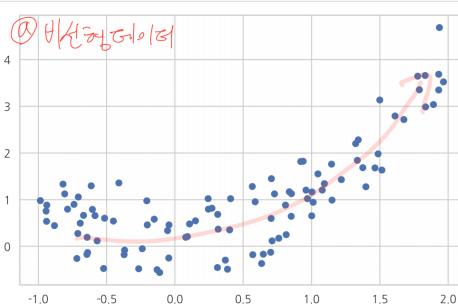


2) 잔차와 독립 변수의 관계 => 관계가 있다면, 기본 가정을 위배

- 잔차가 정규분포이더라도(데이터가 비선형임에도), 잔차 - 독립변수 간 관계 다시 살펴봐야 함
(3-5 page, a-d)

```
def make_regression3(n_sample=100, bias=0, noise=0.5, random_state=0):
    np.random.seed(random_state)
    x = np.random.rand(n_sample) * 3 - 1
    epsilon = noise * np.random.randn(n_sample)
    y = x ** 2 + bias + epsilon
    return x, y

x3, y3 = make_regression3()
plt.scatter(x3, y3)
plt.show()
```



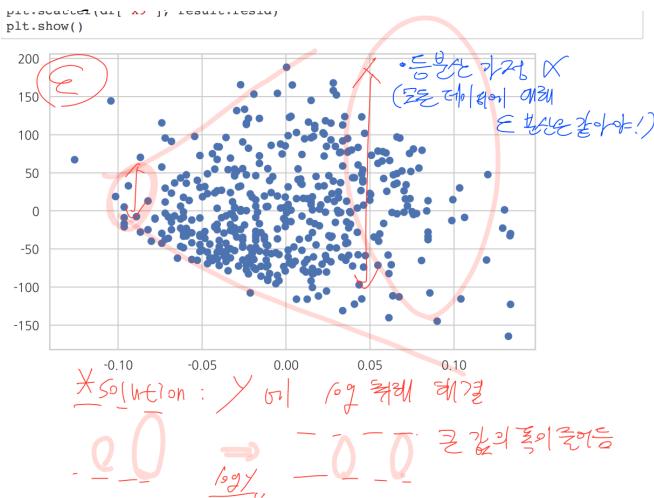
```
In [8]:
test = sm.stats.omni_normtest(result3.resid)
for xi in zip(['Chi^2', 'P-value'], test):
    print("%-12s: %6.3f" % xi)
```

Chi^2 : 1.202
P-value : 0.548 => 잔차는 정규분포

데이터가 모형 가정을 따르지 않지만 잔차는 정규 분포를 따르는 것을 알 수 있다.

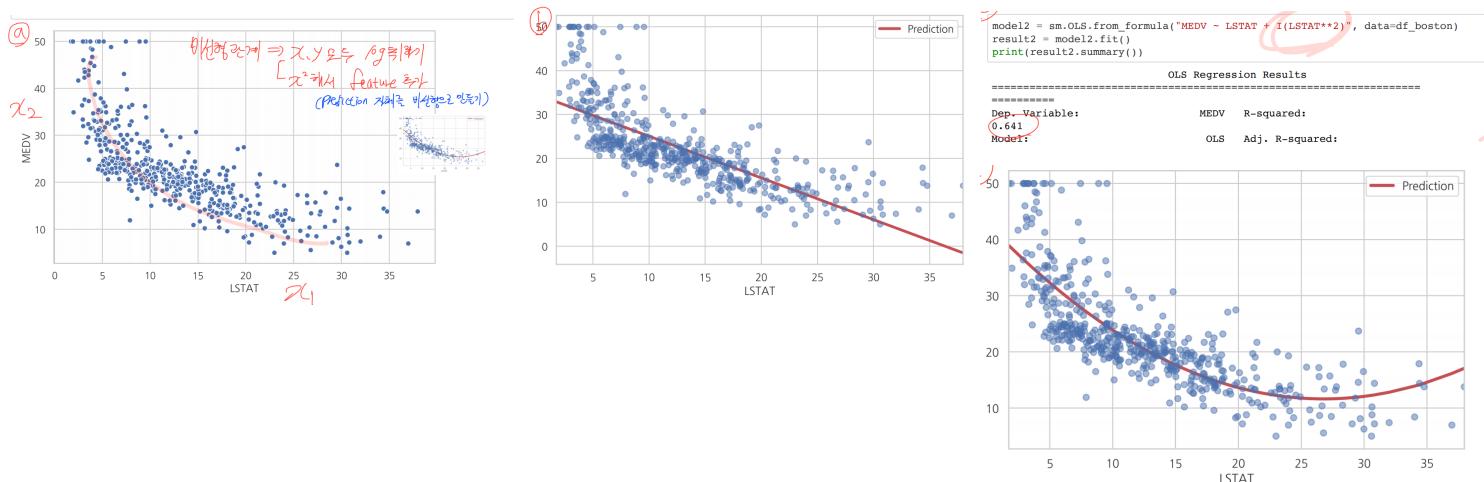
이러한 경우에는 잔차와 독립 변수간의 관계를 살펴보는 것이 도움이 될 수 있다. 데이터가 올바른 모형으로 분석되었다면 잔차는 더이상 독립 변수와 상관관계를 가지지 않아야 한다. 만약 잔차와 독립 변수간에 어떤 비선형 상관관계를 찾을 수 있다면 올바른 모형이 아니다.

3) 이분산성 => ϵ 의 분산이 데이터에 따라 다르다면, 기본 가정을 위배



4) 자기 상관 계수 => y_1, y_2, y_3 간 상관관계 시, 시계열 모형을 활용해야 함

5) 비선형 변형 => 종속-독립 간 비선형이면, 관계를 선형으로 바꿀 수 있도록, "독립 자체를 비선형으로" 변환! => 비선형 변환 방법 : 1) x,y log취하기 2) 제곱, 세제곱으로 feature 추가해 비선형으로 변환



7) 시간 독립변수의 변형 => 1) epoch 기준 지나온 시간(실수형)으로 변형 (시간값 => 반드시 스케일링!) => 2) 연/월/일/요일 등을 개별 feature column으로 분리! (상관관계 발견 가능) (16-17p, a-b)

	Demand	Date
360	173.727990	2014-12-27
361	188.512817	2014-12-28
362	191.273009	2014-12-29
363	186.240144	2014-12-30
364	186.370181	2014-12-31

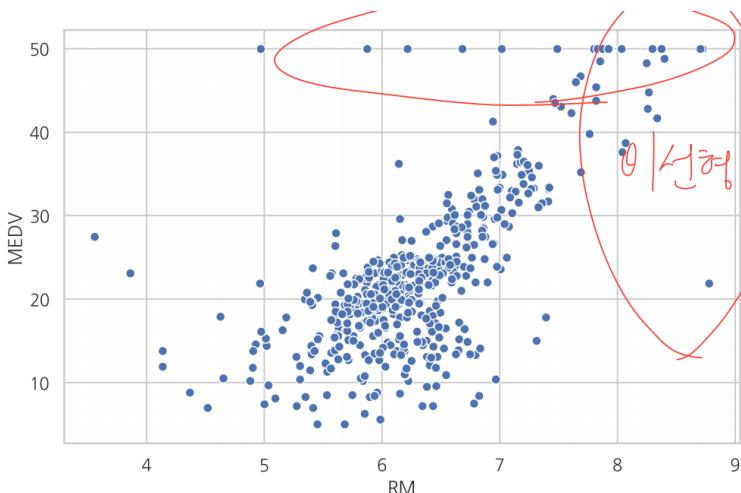
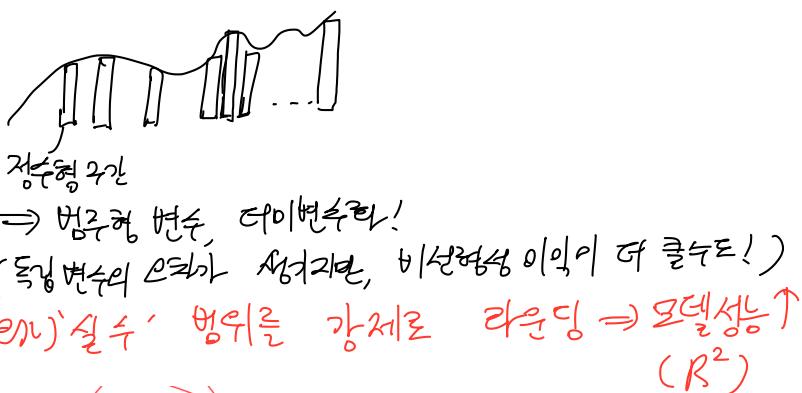
파이썬 datetime 자료형은 toordinal 명령으로 특정 시점으로부터 경과한 시간의 일 단위 값을 구하거나 timestamp 메서드로 초 단위 값을 구할 수 있다. 날짜 - 0(-0)

In [11]:	import datetime as dt	model5 = sm.OLS.from_formula("Demand ~ scale(Ordinal)", data=df_elec)
	df_elec["Ordinal"] = df_elec.Date.map(dt.datetime.toordinal)	Dep. Variable: Demand R-squared:
	df_elec["Timestamp"] = df_elec.Date.map(dt.datetime.timestamp)	0.031 Model: OLS
Out[11]:		Adj. R-squared:
	Y X 12/24 3:00 PM	0.028 Method: Least Squares
		F-statistic: 11.58 Date: Mon, 17 Jun 2019
		Time: 17:28:32 Prob (F-statistic): 0.000739
		No. Observations: 365 AIC: 3423.0
		Df Residuals: 363 BIC: 3431.0
		Df Model: 1 Covariance Type: nonrobust
		=====
		coef std err t P> t
	25 0.975]	[0.0
	Intercept 221.2775 1.374 160.997 0.000 218.5	
	scale(Ordinal) 223.980 -4.6779 1.374 -3.404 0.001 -7.3	
	81 -1.975]	

여기에서는 일 단위 시간 값을 사용하여 회귀분석을 한다. 시간 값의 경우 크기가 크므로 반드시 스케일링을 해 주어야 한다.

6) 범주형을 사용한 비선형성 => 독립변수를 강제로 범주형으로 끊어서 종속의 '비선형 변화'를 '비선형 상수항'으로 모형화해 모델 성능 향상 (11-15p, a-e)

너무 비선형 심할 때



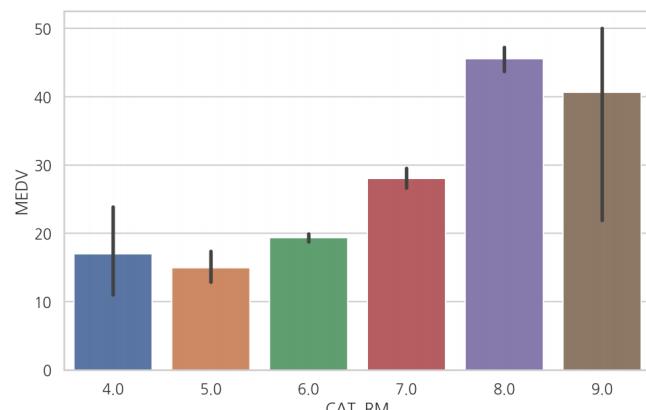
OLS Regression Results					
Dep. Variable:	MEDV	R-squared:			
0.484					
Model:	OLS	Adj. R-squared:			
0.483					
Method:	Least Squares	F-statistic:			
471.8					
Date:	Mon, 17 Jun 2019	Prob (F-statistic):			
2.49e-74					
Time:	17:28:29	Log-Likelihood:			
-1673.1					
No. Observations:	506	AIC:			
3350.					
Df Residuals:	504	BIC:			
3359.					
Df Model:	1	Covariance Type:	nonrobust		
coef	std err	t	P> t	[0.025]	
0.975					

Intercept	-34.6706	2.650	-13.084	0.000	-39.877
-29.465					
RM	9.1021	0.419	21.722	0.000	8.279
9.925					
=====					
Omnibus:	102.585	Durbin-Watson:			
0.684					
Prob(Omnibus):	0.000	Jarque-Bera (JB):			
612.449					
Skew:	0.726	Prob(JB):			
1.02e-133					
Kurtosis:	8.190	Cond. No.			
58.4					
=====					
Warnings:					
[1] Standard Errors assume that the covariance matrix of the errors					
is correctly specified.					

RM 변수값을 정수로 라운딩(rounding)하면 RM 변수가 가지는 비선형성을 잡을 수 있다. 다음 플롯은 카테고리값으로 범위 RM 변수와 종속변수의 관계를 시각화한 것이다.

```
rooms = np.arange(3, 10)
labels = [str(r) for r in rooms[:-1]]
df_boston["CAT_RM"] = np.round(df_boston.RM)

sns.barplot(x="CAT_RM", y="MEDV", data=df_boston)
plt.show()
```



이렇게 하면 RM 변수으로 인한 종속변수의 변화를 비선형 상수항으로 모형화 할 수 있다. 선형모형보다 성능이 향상된 것을 볼 수 있다.

OLS Regression Results					
Dep. Variable:	MEDV	R-squared:			
0.537					
Model:	OLS	Adj. R-squared:			
0.532					
Method:	Least Squares	F-statistic:			
115.8					
Date:	Mon, 17 Jun 2019	Prob (F-statistic):			
3.57e-81					
Time:	17:28:29	Log-Likelihood:			
-1645.6					
No. Observations:	506	AIC:			
3303.		BIC:			
3329.					
Df Model:	5	Covariance Type:	nonrobust		
coef	std err	t	P> t	[0.025]	
[0.025 0.975]					

Intercept	11.492	22.548	0.509	0.000	
11.492					
C(np.round(RM))(T.5.0)	-2.0741	2.998	-0.692	0.489	
-7.964					
C(np.round(RM))(T.6.0)	2.3460	2.836	0.827	0.409	
-3.226					
C(np.round(RM))(T.7.0)	11.0272	2.869	3.843	0.000	
5.389					
C(np.round(RM))(T.8.0)	28.5425	3.093	9.228	0.000	
22.466					
C(np.round(RM))(T.9.0)	23.6133	4.595	5.139	0.000	
14.586					
C(np.round(RM))(T.0.0)	32.641				
=====					
Omnibus:	81.744	Durbin-Watson:			
0.799					
Prob(Omnibus):	0.000	Jarque-Bera (JB):			
467.887					
Skew:	0.542	Prob(JB):			
2.51e-102					
Kurtosis:	7.584	Cond. No.			
31.1					

8) 주기성을 갖는 드롭변수

⇒ 각도레이터는 무조건 \sin , \cos 로 변경! ($0^\circ = 360^\circ$)

9) 종속변수 변형

⇒ 비선형 y 를 직선형으로 변형!

```
plt.scatter(boston.target, y_hat1)
plt.xlabel("실제 집값")
plt.ylabel("집값 예측치")
plt.title("집값 예측치와 실제 집값의 관계")
plt.show()
```

