hw4nk_ipynb

September 22, 2024

[4]:
```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import statsmodels.api as sm
import statsmodels.formula.api as smf
```

[5]:
```
#Question 5
kbbdata=pd.read_csv('../data/KelleyBlueBookData.csv')
kbbdata.sample(10)
```

[5]:
```
            Price  Mileage       Make        Model           Trim   Type  \
488  21403.756420    27168    Pontiac  Bonneville    GXP Sedan 4D  Sedan
193  11615.021020    19014  Chevrolet        AVEO     LT Sedan 4D  Sedan
718  25845.206110    36557       SAAB         9_5 Linear Wagon 4D  Wagon
80   51154.047220     2202   Cadillac      CST-V        Sedan 4D  Sedan
566  18701.222620    24992    Pontiac  Grand Prix    GTP Sedan 4D  Sedan
479  20221.808810    26223  Chevrolet Monte Carlo     SS Coupe 2D  Coupe
535  15595.884130    18315    Pontiac    Grand Am     GT Coupe 2D  Coupe
53   21575.456830    20137      Buick     Lesabre Limited Sedan 4D  Sedan
222  12045.920700    19136  Chevrolet    Cavalier        Coupe 2D  Coupe
199   9919.048185    34621  Chevrolet        AVEO     LT Sedan 4D  Sedan

     Cylinder  Liter  Doors  Cruise  Sound  Leather
488         8    4.6      4       1      0        1
193         4    1.6      4       0      1        1
718         4    2.3      4       1      1        1
80          8    5.7      4       1      1        1
566         6    3.8      4       1      0        0
479         6    3.8      2       1      1        1
535         6    3.4      2       0      1        1
53          6    3.8      4       1      1        0
222         4    2.2      2       0      1        1
199         4    1.6      4       0      1        0
```

[6]:
```
kbb =kbbdata[['Price','Mileage','Liter','Cylinder']].copy()
kbb.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 804 entries, 0 to 803
Data columns (total 4 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   Price     804 non-null    float64
 1   Mileage   804 non-null    int64  
 2   Liter     804 non-null    float64
 3   Cylinder  804 non-null    int64  
dtypes: float64(2), int64(2)
memory usage: 25.3 KB
```

[8]:
```
model = smf.ols('Price~Mileage+Liter+Cylinder',data=kbb).fit()
model.summary()
```

[8]:

Dep. Variable:	Price	**R-squared:**	0.342
Model:	OLS	**Adj. R-squared:**	0.340
Method:	Least Squares	**F-statistic:**	138.8
Date:	Sun, 22 Sep 2024	**Prob (F-statistic):**	2.18e-72
Time:	15:25:09	**Log-Likelihood:**	-8367.7
No. Observations:	804	**AIC:**	1.674e+04
Df Residuals:	800	**BIC:**	1.676e+04
Df Model:	3		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	4707.6150	1602.866	2.937	0.003	1561.296	7853.934
Mileage	-0.1544	0.035	-4.461	0.000	-0.222	-0.086
Liter	1545.2522	893.411	1.730	0.084	-208.454	3298.958
Cylinder	2847.9345	712.040	4.000	0.000	1450.247	4245.622

Omnibus:	214.158	**Durbin-Watson:**	0.074
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	444.825
Skew:	1.499	**Prob(JB):**	2.56e-97
Kurtosis:	5.071	**Cond. No.**	1.37e+05

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.37e+05. This might indicate that there are
strong multicollinearity or other numerical problems.

[9]:
```
sm.stats.anova_lm(model,typ=1)
```

[9]:

	df	sum_sq	mean_sq	F	PR(>F)
Mileage	1.0	1.605590e+09	1.605590e+09	24.890171	7.448191e-07
Liter	1.0	2.421824e+10	2.421824e+10	375.435819	7.134623e-69
Cylinder	1.0	1.031948e+09	1.031948e+09	15.997457	6.930502e-05
Residual	800.0	5.160560e+10	6.450701e+07	NaN	NaN

[10]:
```
null_model = smf.ols('Price ~ Mileage + Liter', data=kbb).fit()

SSE_H0 = sum(null_model.resid ** 2)
df_SSE_H0 = null_model.df_resid

SSE_H1 = sum(model.resid ** 2)
df_SSE_H1 = model.df_resid

print(f"SSE_H0:{SSE_H0}")
print(f"df_SSE_H0: {df_SSE_H0}")
print(f"SSE_H1: {SSE_H1}")
print(f"df_SSE_H1: {df_SSE_H1}")
```

```
SSE_H0:52637552166.235855
df_SSE_H0: 801.0
SSE_H1: 51605604120.22623
df_SSE_H1: 800.0
```

[12]:
```
F_alt = ((SSE_H0 - SSE_H1) / (df_SSE_H0 - df_SSE_H1)) / (SSE_H1 / df_SSE_H1)
print(f"F_alt: {F_alt}")
```

```
F_alt: 15.997457076258398
```

[13]:
```
sm.stats.anova_lm(model,typ=2)
```

[13]:

	sum_sq	df	F	PR(>F)
Mileage	1.283997e+09	1.0	19.904763	0.000009
Liter	1.929760e+08	1.0	2.991552	0.084086
Cylinder	1.031948e+09	1.0	15.997457	0.000069
Residual	5.160560e+10	800.0	NaN	NaN

[17]:
```
null_model = smf.ols('Price ~ Liter + Cylinder', data=kbb).fit()

SSE_H0 = sum(null_model.resid ** 2)
df_SSE_H0 = null_model.df_resid

SSE_H1 = sum(model.resid ** 2)
df_SSE_H1 = model.df_resid

print(f"SSE_H0:{SSE_H0}")
print(f"df_SSE_H0: {df_SSE_H0}")
print(f"SSE_H1: {SSE_H1}")
print(f"df_SSE_H1: {df_SSE_H1}")
```

```
SSE_H0:52889600780.65453
df_SSE_H0: 801.0
SSE_H1: 51605604120.22623
df_SSE_H1: 800.0
```

[18]:
```
F_alt = ((SSE_H0 - SSE_H1) / (df_SSE_H0 - df_SSE_H1)) / (SSE_H1 / df_SSE_H1)
print(f"F_alt: {F_alt}")
```

```
F_alt: 19.904763171642575
```

[19]:
```
#Question 6
model_1 = smf.ols('Price ~ Mileage + Cylinder', data=kbb).fit()
model_1.summary()
```

[19]:

Dep. Variable:	Price	**R-squared:**	0.340
Model:	OLS	**Adj. R-squared:**	0.338
Method:	Least Squares	**F-statistic:**	206.2
Date:	Sun, 22 Sep 2024	**Prob (F-statistic):**	5.95e-73
Time:	15:48:45	**Log-Likelihood:**	-8369.2
No. Observations:	804	**AIC:**	1.674e+04
Df Residuals:	801	**BIC:**	1.676e+04
Df Model:	2		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	3145.7503	1325.934	2.372	0.018	543.034	5748.467
Mileage	-0.1524	0.035	-4.401	0.000	-0.220	-0.084
Cylinder	4027.6746	204.612	19.684	0.000	3626.036	4429.313

Omnibus:	198.944	**Durbin-Watson:**	0.077
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	385.493
Skew:	1.439	**Prob(JB):**	1.96e-84
Kurtosis:	4.797	**Cond. No.**	1.01e+05

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.01e+05. This might indicate that there are
strong multicollinearity or other numerical problems.

[21]:
```
SSE = sum(model_1.resid ** 2)
SST = sum((kbb['Price'] - kbb['Price'].mean()) ** 2)
n = len(kbb)
p = len(model_1.params)

R2 = 1 - (SSE / SST)
R2_adj = 1 - ((SSE / (n - p)) / (SST / (n - 1)))

print(f"SSE: {SSE}")
print(f"SST: {SST}")
print(f"n: {n}")
print(f"p: {p}")
print(f"R^2: {R2}")
print(f"R^2_adj: {R2_adj}")
```

```
SSE: 51798580167.895294
```

```
SST: 78461382864.00787
n: 804
p: 3
R^2: 0.33982070826263056
R^2_adj: 0.338172320517968
```

[23]:
```python
model.summary()
```

[23]:

Dep. Variable:	Price	**R-squared:**	0.342
Model:	OLS	**Adj. R-squared:**	0.340
Method:	Least Squares	**F-statistic:**	138.8
Date:	Sun, 22 Sep 2024	**Prob (F-statistic):**	2.18e-72
Time:	15:57:54	**Log-Likelihood:**	-8367.7
No. Observations:	804	**AIC:**	1.674e+04
Df Residuals:	800	**BIC:**	1.676e+04
Df Model:	3		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	4707.6150	1602.866	2.937	0.003	1561.296	7853.934
Mileage	-0.1544	0.035	-4.461	0.000	-0.222	-0.086
Liter	1545.2522	893.411	1.730	0.084	-208.454	3298.958
Cylinder	2847.9345	712.040	4.000	0.000	1450.247	4245.622

Omnibus:	214.158	**Durbin-Watson:**	0.074
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	444.825
Skew:	1.499	**Prob(JB):**	2.56e-97
Kurtosis:	5.071	**Cond. No.**	1.37e+05

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.37e+05. This might indicate that there are
strong multicollinearity or other numerical problems.