Mustafa Cankan BALCI
22101761
7 March 2025

GE461 – Introduction to Data Science Project 1

Introduction

A project focuses on the interpolation of different linear regression. The project constructs on 2 question 3.7.8 and 3.7.9 from *An Introduction to Statistical Learning with Application in Python*. A dataset of project is called Auto, which is provided by the course book python library (ISLP). The columns of dataset are ‘name’ 'mpg', 'cylinders', 'displacement', 'horsepower', 'weight’, ‘acceleration', 'year', 'origin'. A simple linear regression and multilinear regression are applied in the provided questions respectively. In both question the estimated feature is ‘mpg’.

Solution of Question 3.7.8

a) In this part, I looked the relationship between ‘mpg’ as a response and ‘horsepower’ as the predictor. A simple linear formula is provided below.

$$y = \beta_0 + \beta_1 x$$

First of all, ordinary least square function (OLS) function is applied depending on the question. ‘summarize()’ function result is figure 1. According to obtained result, a relationship among the predictor and response. The t-statistic for the horsepower coefficient is large in the magnitude, and the corresponding p-value is effectively zero. As a result, horsepower is highly significant in predicting the response. The coefficient of determination, or R^2, tells us how much of the variation in the response variable is explained by our predictor. In this case, an R^2 of 0.606 means that about 60.6% of the variation in the response can be attributed to horsepower alone. This suggests that horsepower is a significant predictor, though other factors likely play a role as well.

	coef	std err	t	P>|t|
intercept	39.9359	0.717	55.660	0.0
horsepower	-0.1578	0.006	-24.489	0.0

Fig. 1: Result of ‘summarize()’ function

OLS Regression Results

Dep. Variable:	mpg	R-squared:	0.606
Model:	OLS	Adj. R-squared:	0.605
Method:	Least Squares	F-statistic:	599.7
Date:	Mon, 03 Mar 2025	Prob (F-statistic):	7.03e-81
Time:	19:35:37	Log-Likelihood:	-1178.7
No. Observations:	392	AIC:	2361.
Df Residuals:	390	BIC:	2369.
Df Model:	1		
Covariance Type:	nonrobust		

Fig. 2: Result of ‘summary()’ function

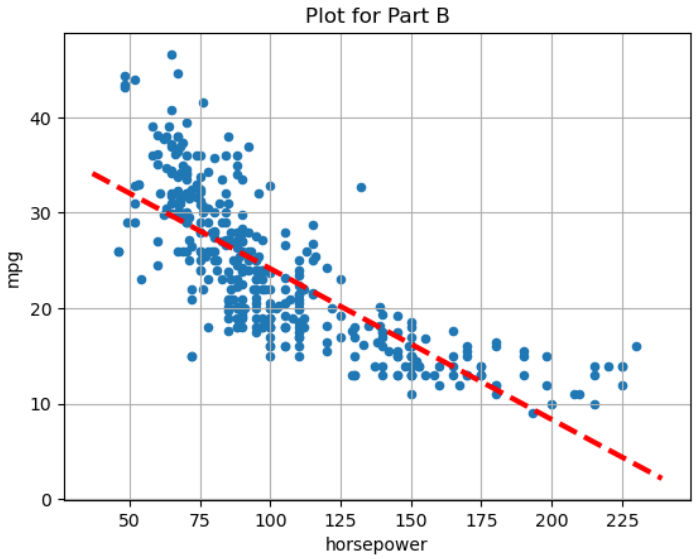
The relationship between the predictor and the response is negative relation since ‘mpg’ decreases if ‘horsepower’ increases. The negative linear relationship occurred owing to the negative slope. The prediction function is created for predicting result. The predicted mpg

associated with a horsepower of 98 is 24.467. Besides, the associated 95% confidence and prediction interval is provided in figure 3.

```
Result: 24.46707715251242
Confidence Interval: [[23.97307896 24.96107534]
Prediction Interval: [[14.80939607 34.12475823]
```

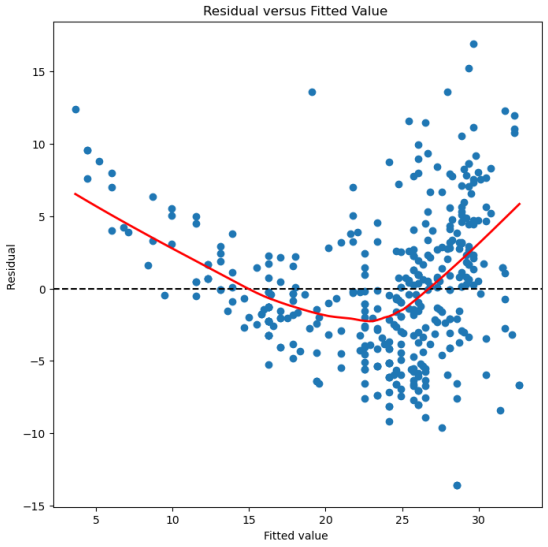
Fig. 3: 95% confidence and prediction interval

b) The plot was drawn for showing the least squares regression line and data points.



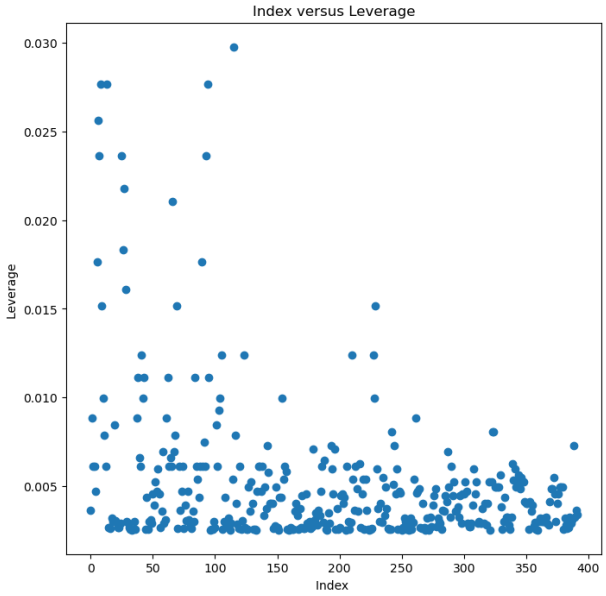
Graph 1: Plot for Part B

c) The diagnostic plots of the least squares regression fit are produced according to given description in the lab. Graph 1 shows that the residuals' spread widens at higher predicted values. The points diverge farther from the zero line as fitted values increase, suggesting that the error variance of the model is not constant. There might be mild non-linearity in the data.



Graph 2: Diagnostic Plot 1

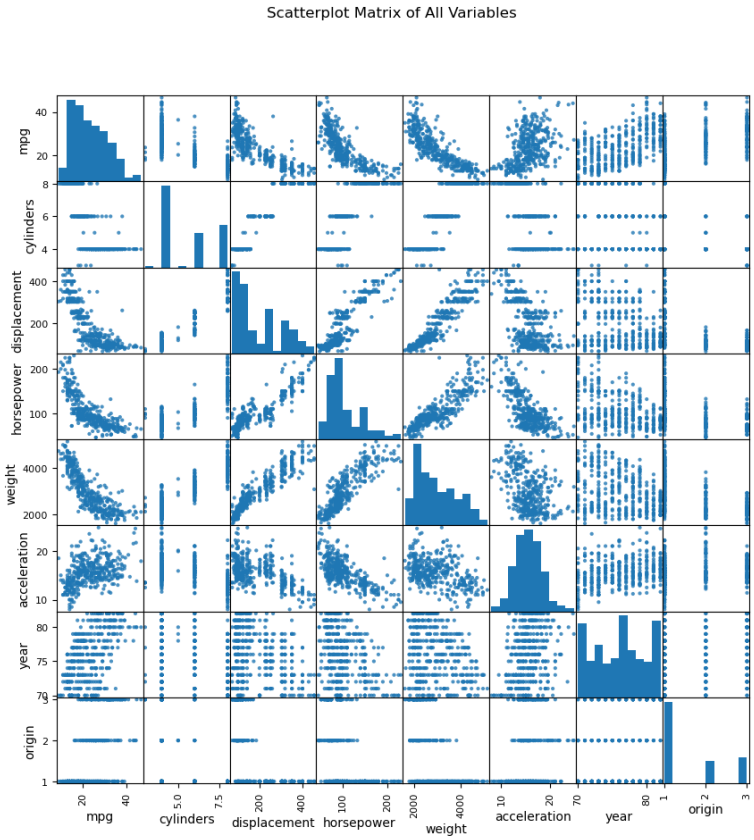
Leverage shows how much influence a data point can have on the fitted regression. Most of the points seem to have fairly low leverage values around or below 0.005. A few points near low index rise up to 0.02 or even 0.003. This shows that those observations have higher-than-average leverage.



Graph 3: Diagnostic Plot 2

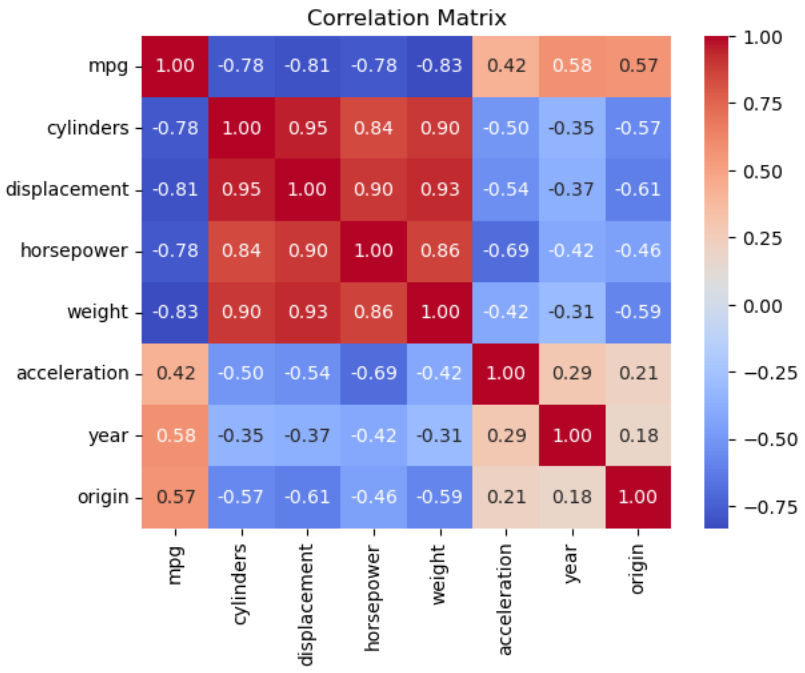
Solution of Question 3.7.9

a)



Graph 4: Scatterplot Matrix of All Variables

b)



Graph 5: Correlation Matrix

c) In the part c, multilinear regression is applied. All other features expect name 'name' used as parameters in this case. Based on the ANOVA table, most predictors significantly influence the response variable. In particular, the extremely small p-values for cylinders, displacement, horsepower, weight, year, and origin suggest that these factors have a statistically significant effect. The only predictor that does not appear to contribute significantly is acceleration, with a p-value of approximately 0.77, which is well above the conventional significance threshold. Overall, these results imply a strong relationship between nearly all predictors and the response variable.

	df	sum_sq	mean_sq	F	PR(>F)
cylinders	1.0	14403.083079	14403.083079	1300.683788	2.319511e-125
displacement	1.0	1073.344025	1073.344025	96.929329	1.530906e-20
horsepower	1.0	403.408069	403.408069	36.430140	3.731128e-09
weight	1.0	975.724953	975.724953	88.113748	5.544461e-19
acceleration	1.0	0.966071	0.966071	0.087242	7.678728e-01
year	1.0	2419.120249	2419.120249	218.460900	1.875281e-39
origin	1.0	291.134494	291.134494	26.291171	4.665681e-07
Residual	384.0	4252.212530	11.073470	NaN	NaN

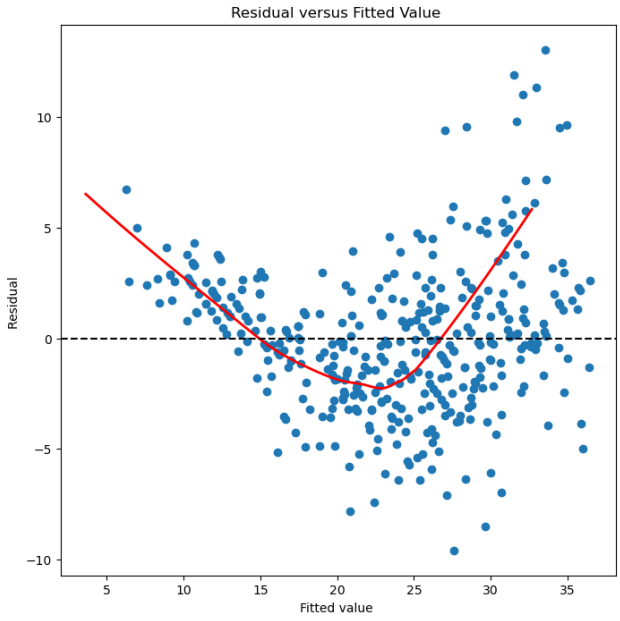
Fig. 4: 'anova_lm()' function result

In figure 5, most of predictors show a statistically significant relationship with the response. The intercept is also significant, but cylinders, horsepower, and acceleration do not appear to be significant at the 5% level.The coefficient for the year variable suggests 0.7508.

	coef	std err	t	P>|t|
Intercept	-17.2184	4.644	-3.707	0.000
cylinders	-0.4934	0.323	-1.526	0.128
displacement	0.0199	0.008	2.647	0.008
horsepower	-0.0170	0.014	-1.230	0.220
weight	-0.0065	0.001	-9.929	0.000
acceleration	0.0806	0.099	0.815	0.415
year	0.7508	0.051	14.729	0.000
origin	1.4261	0.278	5.127	0.000

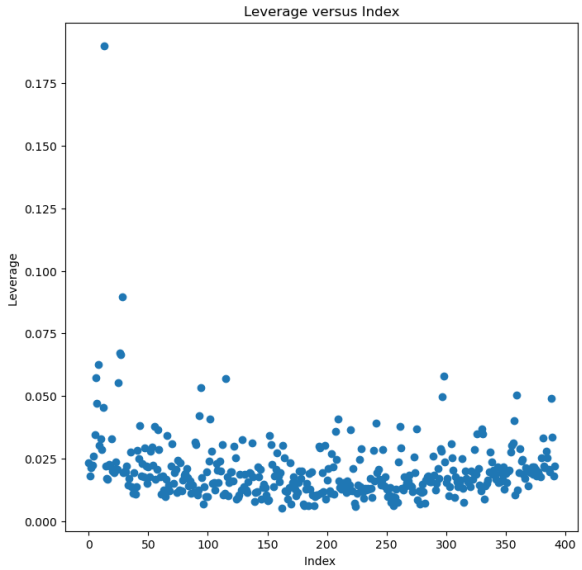
Fig. 5: 'summarize()' function result

d) Some of diagnostic plots of the linear regression fit as described in the lab is plotted. The slight curve or bow shape shows that the relationship between predictors and the response can have a nonlinear relationship. The residual plots suggest unusually large outliers. When the fitted values increase, the residuals become more spread out. This might be a sign of non-constant variance.



Graph 6: Diagnostic Plot 1

In graph 7, there is a single point near the top around 0.18, which is higher than all other points. This particular observation might have a strong influence on the fitted model. It can be an outlier or a potential data entry error. Data points expect single high-leverage point have relatively low leverage below 0.05.



Graph 7: Diagnostic Plot 2

e) Also, some models with different interactions as described in the lab is tried. Figure 6 shows that both the horsepower x weight x acceleration and the year x origin interactions show statistically significant effects, with p-values of approximately 0.003 and 0.001 in orderly. In figure 7, the interaction of cylinders x acceleration is also significant with a p-value near 0. Consequently, there are clearly some interactions between variables that are statistically based on their small p-values.

	coef	std err	t	P>|t|
intercept	-36.4267	4.911	-7.417	0.000
cylinders	-0.0936	0.307	-0.304	0.761
displacement	0.0585	0.010	5.685	0.000
horsepower	-0.0481	0.014	-3.460	0.001
weight	-0.0010	0.001	-0.918	0.359
acceleration	0.6902	0.132	5.245	0.000
year	0.7826	0.047	16.560	0.000
origin	5.7569	1.201	4.795	0.000
displacement:acceleration	-0.0047	0.001	-6.922	0.000
weight:origin	-0.0021	0.001	-4.011	0.000

Fig. 6: Interaction model 1

	coef	std err	t	P>|t|
intercept	-34.2416	5.741	-5.964	0.000
cylinders	2.9398	0.781	3.764	0.000
displacement	0.0086	0.008	1.122	0.263
horsepower	-0.0380	0.014	-2.697	0.007
weight	-0.0052	0.001	-7.653	0.000
acceleration	1.0967	0.232	4.719	0.000
year	0.7654	0.050	15.412	0.000
origin	1.2687	0.272	4.657	0.000
cylinders:acceleration	-0.2141	0.045	-4.802	0.000

Fig. 7: Interaction model 2

f) Finally, the different transformations of the variables such as log(X), $\sqrt{X}$, X^2 are observed for finding how does these transformations effect the linear model. The result obtained these trials are showed below.

OLS Regression Results

Dep. Variable:	mpg	R-squared:	0.864
Model:	OLS	Adj. R-squared:	0.862
Method:	Least Squares	F-statistic:	305.0
Date:	Thu, 06 Mar 2025	Prob (F-statistic):	5.69e-161
Time:	19:45:18	Log-Likelihood:	-969.66
No. Observations:	392	AIC:	1957.
Df Residuals:	383	BIC:	1993.
Df Model:	8		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	154.0938	11.112	13.868	0.000	132.246	175.941
np.log10(cylinders)	0.7252	3.645	0.199	0.842	-6.441	7.892
np.log10(displacement)	-3.0400	3.353	-0.907	0.365	-9.633	3.553
horsepower	0.1278	0.020	6.347	0.000	0.088	0.167
np.log(horsepower)	-20.5485	2.597	-7.912	0.000	-25.655	-15.442
np.log(weight)	-11.3050	2.112	-5.352	0.000	-15.458	-7.152
np.log10(acceleration)	-10.1528	3.510	-2.893	0.004	-17.054	-3.252
year	0.7648	0.045	17.139	0.000	0.677	0.853
np.log10(origin)	2.8479	1.115	2.555	0.011	0.656	5.040

Omnibus:	29.059	Durbin-Watson:	1.585
Prob(Omnibus):	0.000	Jarque-Bera (JB):	58.458
Skew:	0.421	Prob(JB):	2.02e-13
Kurtosis:	4.694	Cond. No.	1.02e+04

Fig. 8: Log(X) Transformation

	df_resid	ssr	df_diff	ss_diff	F	Pr(>F)
0	384.0	4252.212530	0.0	NaN	NaN	NaN
1	383.0	3231.194158	1.0	1021.018373	121.02338	1.201179e-24

Fig. 9: Anova_lm() result comparison with normal multilinear and log transform linear model

Figure 8 shows the summary result obtained in the log(X) transformation. The log transformation of cylinder and displacement are not important due to high p-value, 0.842 and 0.365 respectively. R^2 value suggests that the model explains about 84.6% of the variation in response. In figure 9, F score is 121 and p-value is near to 0. According to result obtain in the Anova test, horsepower can have a log term in the linear model.

OLS Regression Results

Dep. Variable:	mpg	R-squared:	0.855
Model:	OLS	Adj. R-squared:	0.852
Method:	Least Squares	F-statistic:	281.8
Date:	Thu, 06 Mar 2025	Prob (F-statistic):	2.50e-155
Time:	19:50:14	Log-Likelihood:	-982.99
No. Observations:	392	AIC:	1984.
Df Residuals:	383	BIC:	2020.
Df Model:	8		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-0.8743	4.301	-0.203	0.839	-9.330	7.582
np.power(cylinders, 2)	0.0457	0.025	1.803	0.072	-0.004	0.096
displacement	-0.0090	0.007	-1.261	0.208	-0.023	0.005
horsepower	-0.3143	0.035	-9.076	0.000	-0.382	-0.246
weight	-0.0036	0.001	-5.354	0.000	-0.005	-0.002
np.power(acceleration, 2)	-0.0082	0.003	-2.808	0.005	-0.014	-0.002
np.power(horsepower, 2)	0.0010	0.000	9.385	0.000	0.001	0.001
year	0.7368	0.046	15.992	0.000	0.646	0.827
origin	0.9881	0.256	3.859	0.000	0.485	1.491

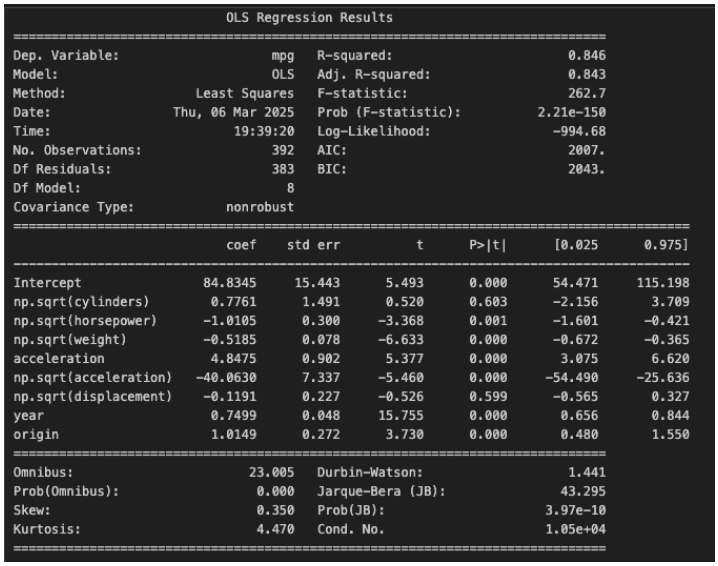
Omnibus:	27.789	Durbin-Watson:	1.540
Prob(Omnibus):	0.000	Jarque-Bera (JB):	45.896
Skew:	0.471	Prob(JB):	1.08e-10
Kurtosis:	4.386	Cond. No.	4.55e+05

Fig. 10: Square(X) Transformation

	df_resid	ssr	df_diff	ss_diff	F	Pr(>F)
0	384.0	4252.212530	0.0	NaN	NaN	NaN
1	383.0	3458.652357	1.0	793.560174	87.876293	6.175863e-19

Fig. 11: Anova_lm() result comparison with normal multilinear and square transform linear model

Figure 10 shows the summary result obtained in the square(X) transformation. R^2 value suggests that the model explains about 85.5% of the variation in response. In figure 11, F = 87.88 and p-value is near to 0. So that, the high R-squared value and lower SSR reinforce that sqrt transformations are helping.



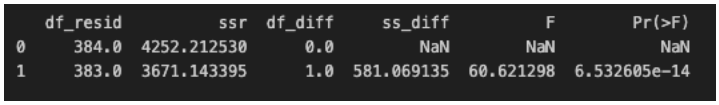
OLS Regression Results

Dep. Variable:	mpg	R-squared:	0.846
Model:	OLS	Adj. R-squared:	0.843
Method:	Least Squares	F-statistic:	262.7
Date:	Thu, 06 Mar 2025	Prob (F-statistic):	2.21e-150
Time:	19:39:20	Log-Likelihood:	-994.68
No. Observations:	392	AIC:	2007.
Df Residuals:	383	BIC:	2043.
Df Model:	8		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	84.8345	15.443	5.493	0.000	54.471	115.198
np.sqrt(cylinders)	0.7761	1.491	0.520	0.603	-2.156	3.709
np.sqrt(horsepower)	-1.0105	0.300	-3.368	0.001	-1.601	-0.421
np.sqrt(weight)	-0.5185	0.078	-6.633	0.000	-0.672	-0.365
acceleration	4.8475	0.902	5.377	0.000	3.075	6.620
np.sqrt(acceleration)	-40.0630	7.337	-5.460	0.000	-54.490	-25.636
np.sqrt(displacement)	-0.1191	0.227	-0.526	0.599	-0.565	0.327
year	0.7499	0.048	15.755	0.000	0.656	0.844
origin	1.0149	0.272	3.730	0.000	0.480	1.550

Omnibus:	23.005	Durbin-Watson:	1.441
Prob(Omnibus):	0.000	Jarque-Bera (JB):	43.295
Skew:	0.350	Prob(JB):	3.97e-10
Kurtosis:	4.470	Cond. No.	1.05e+04

Fig. 12: Sqrt(X) Transformation



	df_resid	ssr	df_diff	ss_diff	F	Pr(>F)
0	384.0	4252.212530	0.0	NaN	NaN	NaN
1	383.0	3671.143395	1.0	581.069135	60.621298	6.532605e-14

Fig. 13: Anova_lm() result comparison with normal multilinear and sqrt transform linear model

Figure 12 shows the summary result obtained in the sqrt(X) transformation. R^2 value suggests that the model explains about 84.6% of the variation in response. In figure 13, F = 60.62 and p-value is near to 0, showing that the including square-root transformations are a statistically important reduction in the unexplained variance. As a result, the high R-squared value and lower p-value reinforce that sqrt transformations are helping the multilinear mode.

Finally, All transformations are affected the multilinear model significantly since they have high F-test score and p-value near to 0. Among 3 different transformation log(X) transformation has F value in Anova test between non-transformed linear model.

Appendix

```
import numpy as np
import pandas as pd
from matplotlib .pyplot import subplots
import statsmodels.api as sm
from statsmodels.stats. outliers_influence \
import variance_inflation_factor as VIF
from statsmodels.stats.anova import anova_lm
from ISLP import load_data
from ISLP.models import ( ModelSpec as MS ,
summarize ,poly)
import seaborn as sns
Auto = load_data('Auto')
print(Auto.columns)
print(Auto)
X = pd.DataFrame({
    'intercept' : np.ones(Auto.shape[0]),
    'horsepower': Auto['horsepower']
})
y = Auto['mpg']
model = sm.OLS(y,X)
results = model.fit()
print(summarize(results))
new_df = pd.DataFrame({'intercept' :1,'horsepower': [98]})
prediction = results.get_prediction(new_df)
mean = prediction.predicted_mean
conf_interval = prediction.conf_int(alpha=0.05)
pred_interval = prediction.conf_int(obs= True, alpha=0.05)
print("Result: " + str(mean[0]))
print("Confidence Interval:",  conf_interval)
print("Prediction Interval:", pred_interval)
print(results.summary())
def abline(ax, b,m,*args,**kwargs):
    "Add a line with slope m and intercept b to ax"
    xlim = ax.get_xlim ()
    ylim = [m * xlim [0] + b, m * xlim [1] + b]
    ax.plot(xlim , ylim , *args , ** kwargs)

ax = Auto.plot.scatter('horsepower', 'mpg')
abline(ax ,
results.params [0],
results.params [1],
'r--',
linewidth =3)
ax.set_title("Plot for Part B")
ax.grid()
ax = subplots (figsize =(8 ,8))[1]
```

```
ax.scatter(results.fittedvalues , results.resid)
ax.set_xlabel ('Fitted value ')
ax.set_ylabel ('Residual ')
ax.axhline (0, c='k', ls='--')
# Compute a LOWESS smoothed curve of the residuals
lowess_smoothed = sm.nonparametric.lowess(results.resid, results.fittedvalues,
frac=0.8)
# Add the smooth curve to the plot
ax.plot(lowess_smoothed[:, 0], lowess_smoothed[:, 1], color='red', linewidth=2)
ax.set_title('Residual versus Fitted Value ')
infl = results. get_influence ()
ax = subplots (figsize =(8 ,8))[1]
ax.scatter(np.arange(X.shape [0]) , infl. hat_matrix_diag )
ax.set_xlabel ('Index ')
ax.set_ylabel ('Leverage ')
np.argmax(infl. hat_matrix_diag )
ax.set_title('Index versus Leverage')
# A part
from pandas.plotting import scatter_matrix
import matplotlib.pyplot as plt
scatter_matrix(Auto, alpha=0.8, figsize=(10, 10), diagonal='hist')
plt.suptitle("Scatterplot Matrix of All Variables")
# B part
result_corr = Auto.corr()
print(result_corr)
sns.heatmap(result_corr, annot=True, cmap="coolwarm", fmt=".2f")
plt.title("Correlation Matrix")
from statsmodels.stats.anova import anova_lm
import statsmodels.formula.api as smf
# Fit the model using the formula interface (excluding 'name')
formula = 'mpg ~ cylinders + displacement + horsepower + weight + acceleration + year
+ origin'
model2 = smf.ols(formula, data=Auto)
result_mul= model2.fit()
print(summarize(result_mul))
print(anova_lm(result_mul))
ax = subplots (figsize =(8 ,8))[1]
ax.scatter(result_mul.fittedvalues , result_mul.resid)
ax.set_xlabel ('Fitted value ')
ax.set_ylabel ('Residual ')
# Compute a LOWESS smoothed curve of the residuals
lowess_smoothed = sm.nonparametric.lowess(results.resid, results.fittedvalues,
frac=0.8)
# Add the smooth curve to the plot
ax.plot(lowess_smoothed[:, 0], lowess_smoothed[:, 1], color='red', linewidth=2)
ax.axhline (0, c='k', ls='--')
ax.set_title('Residual versus Fitted Value ')
infl = result_mul.get_influence ()
```

```
ax = subplots (figsize =(8 ,8))[1]
ax.scatter(np.arange(X.shape [0]) , infl. hat_matrix_diag )
ax. set_xlabel ('Index ')
ax. set_ylabel ('Leverage ')
np.argmax(infl.hat_matrix_diag )
ax.set_title('Leverage versus Index')
# Part E
X_alternative_2 = MS(['cylinders', 'displacement', 'horsepower', 'weight',
'acceleration',
       'year', 'origin', ('displacement', 'acceleration'), ('weight',
'origin')]).fit_transform(Auto)
y = Auto['mpg']
model_alternative2 = sm.OLS(y, X_alternative_2)
results_alternative2 = model_alternative2.fit()
summarize(results_alternative2)
X_alternative_3 = MS(['cylinders', 'displacement', 'horsepower', 'weight',
'acceleration',
       'year', 'origin', ('cylinders', 'acceleration')]).fit_transform(Auto)

model_alternative3 = sm.OLS(y, X_alternative_3)
results_alternative3 = model_alternative3.fit()
summarize(results_alternative3)
#F
# log result
formula_log = 'mpg ~  np.log10(cylinders)+ np.log10(displacement) + horsepower+
np.log(horsepower) + np.log(weight) + np.log10(acceleration) + year +
np.log10(origin)'
model_log = smf.ols(formula_log, data=Auto)
results_log = model_log.fit()
results_log.summary()
print(results_log.summary())
print(anova_lm(result_mul, results_log ))
# square result
formula_square = 'mpg ~ np.power(cylinders,2)+ displacement + horsepower + weight +
np.power(acceleration,2) + np.power(horsepower,2)  + year + origin'
model_square = smf.ols(formula_square, data=Auto)
results_square = model_square.fit()
print(results_square.summary())
print(anova_lm(result_mul, results_square ))
formula_square_root = 'mpg ~ np.sqrt(cylinders)+ np.sqrt(horsepower)+ np.sqrt(weight)
+ acceleration+ np.sqrt(acceleration) + np.sqrt(displacement)  + year + origin '
model_square_root = smf.ols(formula_square_root, data=Auto)
results_square_root = model_square_root.fit()
print(results_square_root.summary())
print(anova_lm(result_mul, results_square_root ))
```