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 projects 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 ‘mps’ 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		

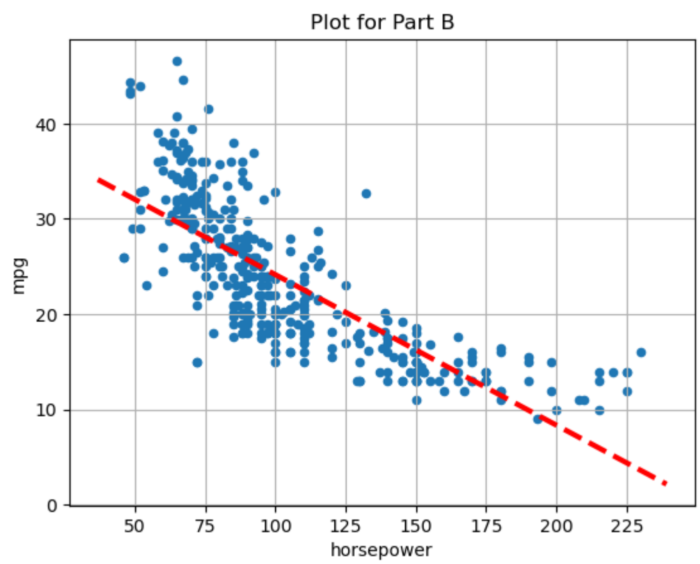
Fig. 2: Result of ‘summary()’ function

The relationship between the predictor and the response is negative relation since ‘mps’ 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 _.

	0	1
intercept	38.525212	41.346510
horsepower	-0.170517	-0.145172

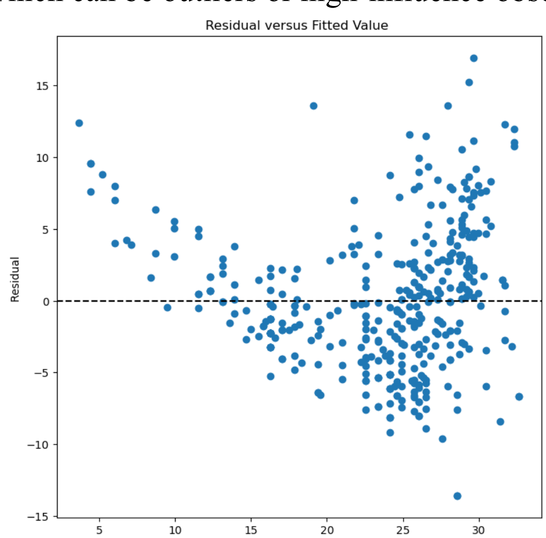
Fig. 3: Confidence Interval Points

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



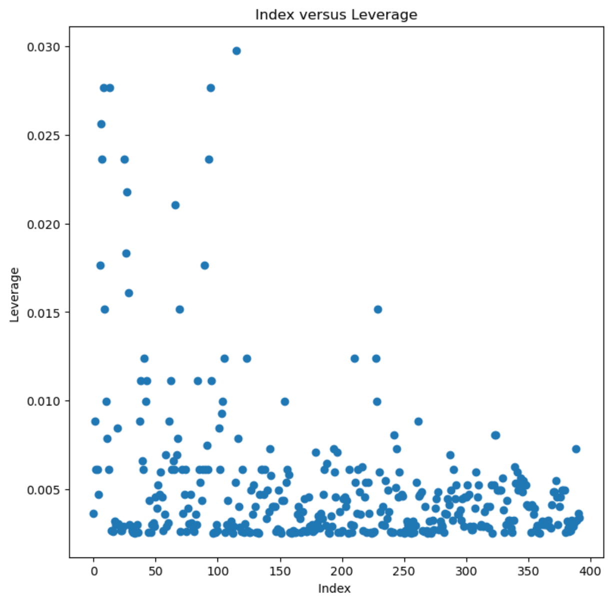
Graph 1: Plot for Part B

c) The diagnostic plots of the least squares regression fit are produced according to given description in the lab. In graph 2, the points are scattered around the zero, which is typically good. Also, there is a vertical spread of the residuals appears to increase somewhat as the fitted values increases. There are a few points appear to have large positive or negative residuals compared to most of points, which can be outliers or high-influence observation.



Graph 2: Diagnostic Plot 1

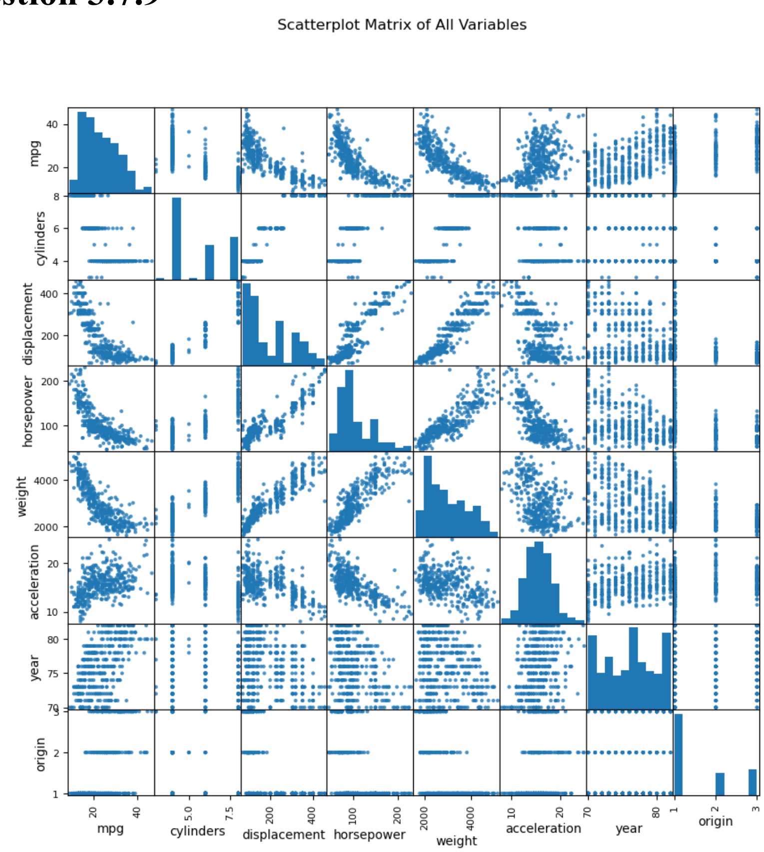
Leverage shows how much influence a data point can have on the fitted regression. Most of the points seem to have fairly low leverage values around or below 0.005. A few points near low index rise up to 0.02 or even 0.003. This shows that those observations have higher-than-average leverage.



Graph 3: Diagnostic Plot 2

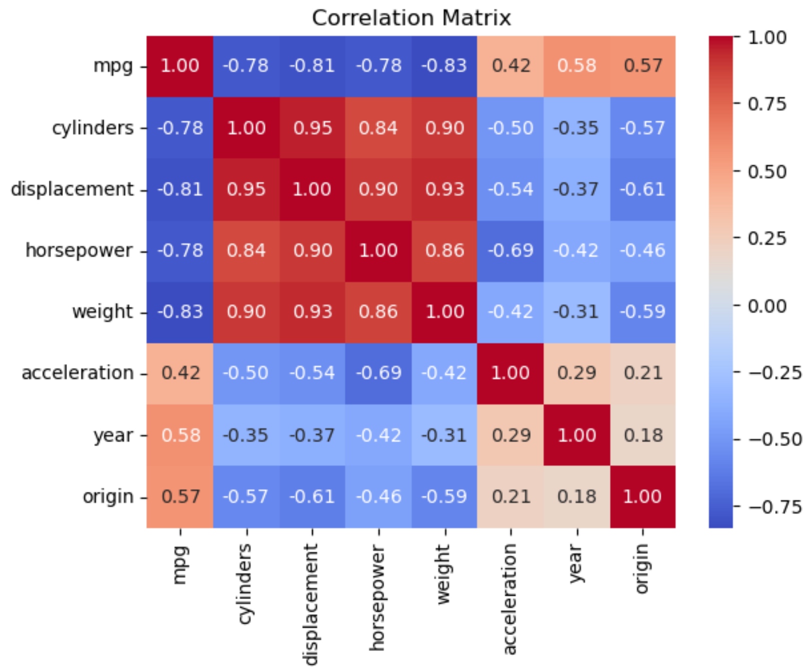
Solution of Question 3.7.9

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

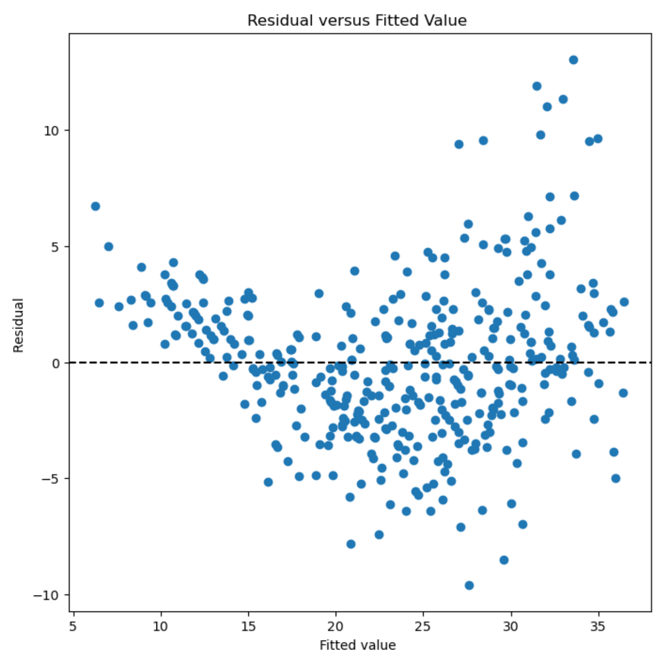
Fig. 4: 'anova_lm()' function result

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

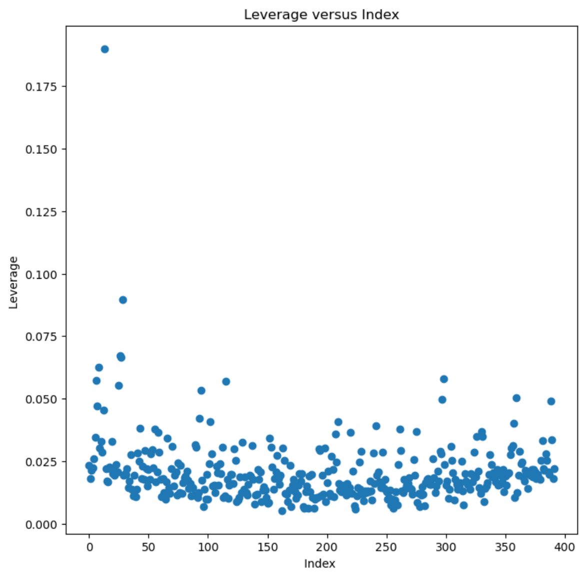
Fig. 5: ‘summarize()’ function result

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.

	coef	std err	t	P>|t|
intercept	1.350000e+01	9.638000e+00	1.401	0.162
cylinders	-6.027000e-01	3.260000e-01	-1.851	0.065
displacement	2.030000e-02	8.000000e-03	2.493	0.013
horsepower	-3.860000e-02	2.100000e-02	-1.800	0.073
weight	-7.600000e-03	1.000000e-03	-7.161	0.000
acceleration	-5.320000e-02	1.380000e-01	-0.385	0.700
year	4.238000e-01	1.120000e-01	3.771	0.000
origin	-1.380250e+01	4.694000e+00	-2.940	0.003
horsepower:weight:acceleration	5.706000e-07	3.840000e-07	1.486	0.138
year:origin	1.959000e-01	6.000000e-02	3.252	0.001

	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

	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

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 figure _, _ and _.

	coef	std err	t	P>|t|
intercept	1.591000e-11	6.660000e-13	23.876	0.000
cylinders	3.705600e+00	3.885000e+00	0.954	0.341
displacement	-6.432000e-01	3.572000e+00	-0.180	0.857
horsepower	-1.848690e+01	3.588000e+00	-5.153	0.000
weight	-3.310450e+01	5.092000e+00	-6.501	0.000
acceleration	-1.310380e+01	3.741000e+00	-3.503	0.001
year	1.002667e+02	4.438000e+00	22.593	0.000
origin	3.304100e+00	1.187000e+00	2.783	0.006

	coef	std err	t	P>|t|
intercept	-49.7981	9.178	-5.426	0.000
cylinders	-0.2370	1.538	-0.154	0.878
displacement	0.2258	0.229	0.984	0.326
horsepower	-0.7798	0.308	-2.533	0.012
weight	-0.6217	0.079	-7.872	0.000
acceleration	-0.8253	0.834	-0.989	0.323
year	12.7903	0.859	14.891	0.000
origin	3.2604	0.768	4.247	0.000

	coef	std err	t	P>|t|
intercept	1.208000e+00	2.356000e+00	0.513	0.608
cylinders	-8.830000e-02	2.500000e-02	-3.502	0.001
displacement	5.680000e-05	1.380000e-05	4.109	0.000
horsepower	-3.621000e-05	4.980000e-05	-0.728	0.467
weight	-9.351000e-07	8.980000e-08	-10.416	0.000
acceleration	6.300000e-03	3.000000e-03	2.334	0.020
year	5.000000e-03	0.000000e+00	14.160	0.000
origin	4.129000e-01	6.900000e-02	5.971	0.000

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')
Auto.columns
X = pd.DataFrame({
    'intercept' : np.ones(Auto.shape[0]),
    'horsepower': Auto['horsepower']
})

y = Auto['mpg']
model = sm.OLS(y,X)
results = model.fit()
summarize(results)
# Looking
results.conf_int(alpha =0.05)
def predicted(b,m,x):
    y = m * x + b
    print(f'{y:.3f}')
    return y

calculated_result = predicted(results.params [0],results.params [1],98)
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='--')
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")
terms = Auto.columns.drop('mpg')
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()
summarize(result_mul)
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 ')
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')
allvars = Auto.columns.drop('mpg')

X_alternative_1 = MS(['cylinders', 'displacement', 'horsepower', 'weight',
'acceleration',
       'year', 'origin', ('horsepower', 'weight', 'acceleration'), ('year',
'origin')]).fit_transform(Auto)

model_alternative = sm.OLS(y, X_alternative_1)
results_alternative = model_alternative.fit()
summarize(results_alternative)
X_alternative_2 = MS(['cylinders', 'displacement', 'horsepower', 'weight',
'acceleration',
       'year', 'origin', ('displacement', 'acceleration'), ('weight',
'origin')]).fit_transform(Auto)

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)
# log result
allvars = Auto.columns.drop('mpg')

X = MS(allvars).fit_transform(Auto)
log_X = np.log10(X +1e-12)
model_log = sm.OLS(y, log_X)
results_log = model_log.fit()
summarize (results_log)
# square root result

sqrt_X = np.sqrt(X)
model_sqrt = sm.OLS(y, sqrt_X)
results_sqrt = model_sqrt.fit()
summarize(results_sqrt)
#results_sqrt.summary()
# square result
square_X = np.square(X)
model_square = sm.OLS(y, square_X)
results_square = model_square.fit()
summarize(results_square)
```