Analysis of Auto Data: Modeling Growth and Evaluating Multiple Linear Regression Assumptions

Issues

The data is obtained as a subset of the Auto data mentioned in “An Introduction to Statistical Learning with Applications in R”, Chapter 3, page 123.
For this data set there are 4 predictor variables: *displacement*, *horsepower*, *weight*, and *acceleration*, and one predicted variable, *mpg*. Here we answer the following issues,
(a) Produce a scatterplot matrix which includes all of the variables in the data set.

(b) Compute the matrix of correlations between the variables using the function cor() . You will need to exclude the name variable, cor() which is qualitative.

(c) Use the lm() function to perform a multiple linear regression with mpg as the response and all other variables except name as the predictors. Use the summary() function to print the results. Comment on the output.

(d) Use the plot() function to produce diagnostic plots of the linear regression fit. Comment on any problems you see with the fit.Do the residual plots suggest any unusually large outliers? Does the leverage plot identify any observations with unusually high leverage?

(e) Use the * and : symbols to fit linear regression models with interaction effects. Do any interactions appear to be statistically significant?

(f) Try a few different transformations of the variables, such as log(X), √X, X^2 . Comment on your findings.

Findings

The data set consists of auto data (displacement, horsepower, weight, acceleration, mpg) as an Excel (xls) file.

We can find out that there are around 389 values. All the values show a correlation between each of the other variables.

- Displacement shows positive correlation with horsepower and weight and negative correlation with acceleration and mpg.
- Horsepower shows a positive correlation with displacement and weight and a negative correlation with acceleration and mpg.
- Weight shows positive correlation with displacement and horsepower and a negative correlation with acceleration and mpg.
- Acceleration shows positive correlation with mpg only and negative correlations with the rest.
- Mpg shows a positive correlation with acceleration only and negative correlation with the rest.

After creating a regression line, the model is summarised. The condition number is large, 3.33e+04. This might indicate that there are strong multicollinearity or other numerical problems.

Discussions

Multiple linear regression is a powerful tool for analysing relationships between multiple variables. In this case, auto dataset with variables such as displacement, horsepower, weight, acceleration, and mpg, multiple linear regression can be used to identify the factors that affect fuel efficiency and to develop models that can be used to predict the fuel efficiency of new automobiles.

Once the model has been developed, it can be used to predict the fuel efficiency of new automobiles based on the four independent variables. The predicted values can be compared to the actual values to assess the accuracy of the model and to identify any areas where improvements can be made.

From the multi linear regression model, we can now find out the r squared value of 0.712 and a skewness of 1.053 and a kurtosis value of 5.102.

From the findings, we can understand that the data is not normally distributed. And the regression line passes through the scattered data points.

Appendix A: Method

The data is imported using the read_excel command from the pandas package as it is an excel file. Then we find out if there are any null values. If not, we create a scatter plot showing the entire figure of all the variables to find the correlations.

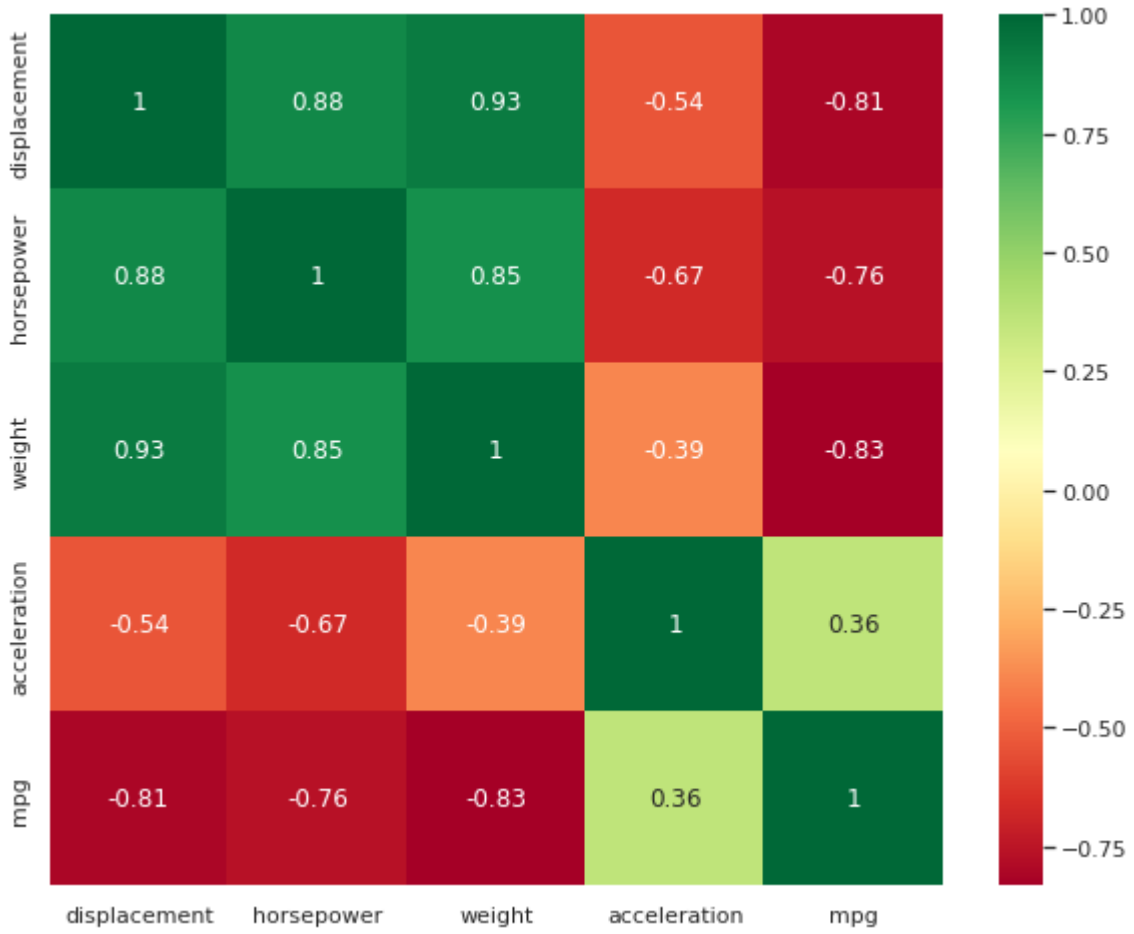
Then we find the regression model and use it to find the residuals. Residuals are the difference between the actual value and predicted value. And we create a histogram to find out it is not a normal distribution. Correlation matrix is also found to know the relation between all the variables.

We then plot out the diagnostic plots of the residuals which are of 4 different types namely- Residual vs fitted, Normal Q-Q, Scale location and Residual vs leverage. All these models will help us decide if the model is a correct fit or not.

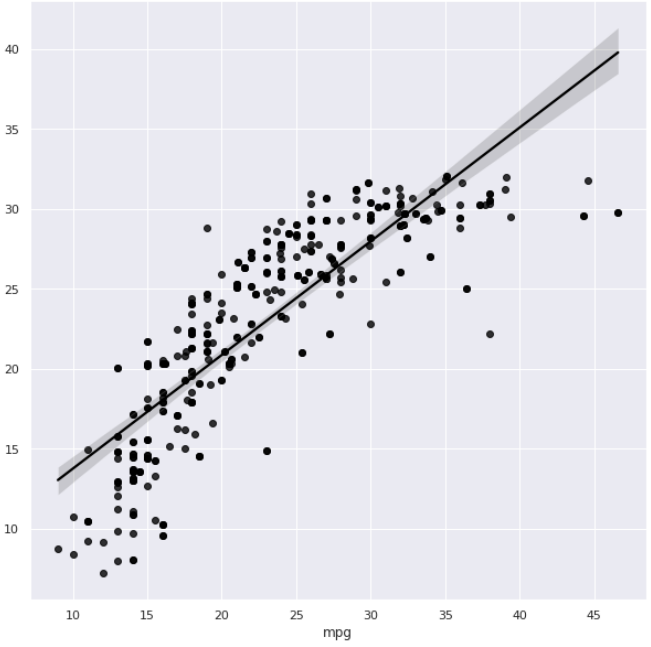
This is done to find out and predict the best fit for the mpg variable.

Appendix B: Results

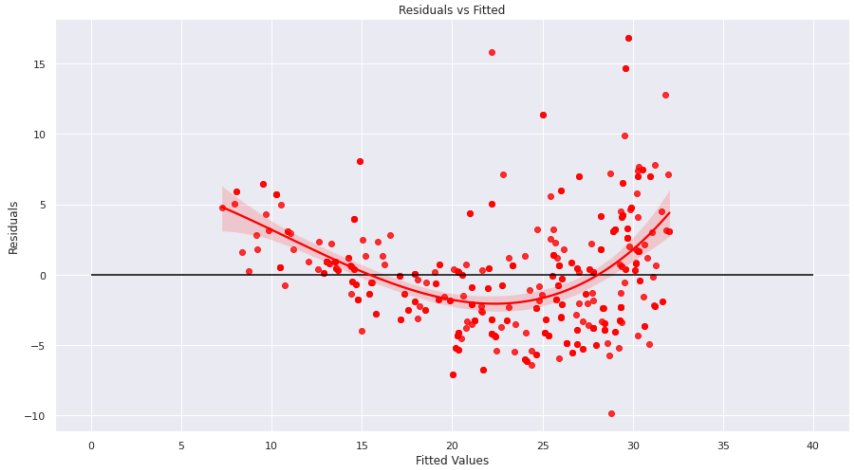
The scatter plot figure helps to find out the correlation between all the variables. Over here we are using the heatmap to understand it in a numerical way. Greener the box, it is positively correlated. More red the box, it is negatively correlated.



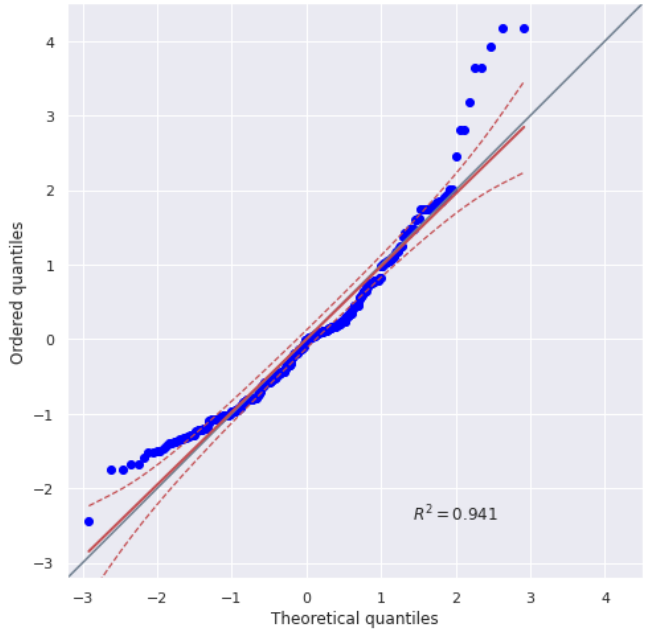
Now, we can find out that the regression line is passing through the data points. It is between the actual y and the predicted y.



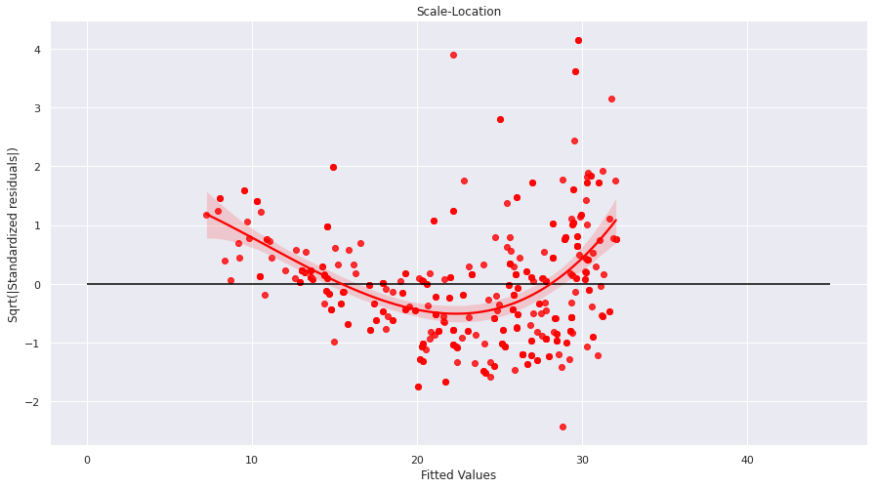
Now, as we plot the diagnostic plots.
Residual vs fitted plot makes us understand that there isn't any clear pattern for the model to fit the best.



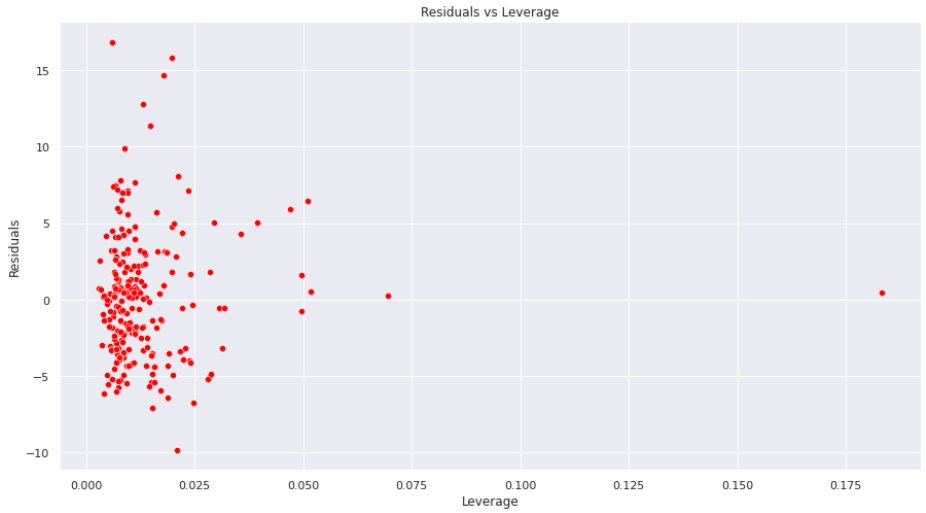
Normal q-q plot shows most of the data points are outside the dotted lines so we can understand it is not normally distributed.



Scale-Location figure helps us to understand that the data points are uniformly distributed at a region and then they are scattered after 20 on the X axis.



In the residuals vs leverage plot, we can find out that one is way off the plot and the rest of the data points are closely scattered. Those points influence if the model can be good or not.



Then we create new interactions and variables to create the models again to see if they would be a good fit.

1. Is atleast one of the predictors useful in predicting the response?

   Yes, there is a relationship between the predicted variable(mpg) and the predictors. As we saw in the heatmap, mpg shows a correlation. P value is close to 0 so we can reject the null hypothesis. We can also say that on the basis of the Correlation Matrix and also model's R-squared value is 0.712

which means Model can predict 71.2 % of mpg values on the basis of other variables.

2.Do all the predictors help to explain the response, or is only a subset of the are predictors useful?

If the p value is less than 0.05, then the predictors are significant. Otherwise they are insignificant. So, all the variables help to explain the response.

3.How well does the model fit the data?

The model is a good fit for the multiple linear regression as the data points are distributed and their contribution helps in accessing the overall plots.

4.Given a set of predictor values, what response value should we predict, and how accurate is our prediction?

The response value that should be predicted will depend on the specific problem and the nature of the predictor values.The accuracy of the prediction will depend on the specific modelling approach and the quality of the data.In general, the accuracy of the prediction can be assessed by comparing the predicted response values to the actual response values using metrics such as mean squared error, root mean squared error, or R-squared. It is also important to validate the model using a separate test set to ensure that it generalises well to new data.

Appendix C: Code

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api as sm
```

```
df = pd.read_excel("auto_data_porandla_rithik.xls")
df.head()

df.info()

df.isnull().sum()

df.describe()

"""Scatter plot for the variables"""

sns.pairplot(df)

df.corr()

sns.set(rc={'figure.figsize':(10,8)})
sns.heatmap(df.corr(),cmap="RdYlGn",annot=True);
plt.show()

X = df[['displacement','horsepower','weight',
              'acceleration']]
y = df['mpg']

X.head()

y.head()

X = sm.add_constant(X)

model = sm.OLS(y,X).fit()

model.params

round(model.params, 3)

model.summary()

y_pred = model.predict(X)

print("The predicted values are: ")
print(y_pred)

y_residual = pd.DataFrame({'Actual Value' : y, 'Predicted Value' :
y_pred, 'Residual' : y - y_pred})

y_residual

sns.distplot(y_residual['Residual'],color='blue',bins=50)
plt.title('Distribution of Residuals')
plt.show()
```

```
"""So it is clearly not a normal dist."""

plt.scatter(y, y_pred, c='black')
plt.xlabel('Actual')
plt.ylabel('Predicted')
plt.title("Actual y VS Predicted y")
plt.show();

sns.set(rc={'figure.figsize':(10,10)})
sns.regplot(y,y_pred,color='black');

from sklearn import metrics

mean_absolute_error = metrics.mean_absolute_error(y,y_pred)
mean_squared_error = metrics.mean_squared_error(y,y_pred)
root_mean_squared_error = np.sqrt(metrics.mean_squared_error(y,y_pred))

print("Mean Absolute Error : ",round(mean_absolute_error,3))
print("Mean Squared Error : ",round(mean_squared_error,3))
print("Root Mean Squared Error : ",round(root_mean_squared_error,3))

res = df.copy()

res['resid'] = model.resid

res['fitted_values'] = model.fittedvalues

res['resid_std'] = model.resid_pearson

res['leverage'] = model.get_influence().hat_matrix_diag

res

sns.set(rc={'figure.figsize':(15,8)})
sns.regplot(x=res['fitted_values'],y=res['resid'],color='red',order=3)
plt.hlines(y=0,xmin=0,xmax=40,color='black')
plt.xlabel("Fitted Values")
plt.ylabel("Residuals")
plt.title("Residuals vs Fitted")
plt.show();

!pip install pingouin

import pingouin as pg

pg.qqplot(res['resid'],dist='norm');


sns.regplot(x=res['fitted_values'],y=res['resid_std'],color='red',order
=3)
```

```
plt.hlines(y=0,xmin=0,xmax=45,color='black')
plt.xlabel("Fitted Values")
plt.ylabel("Sqrt(|Standardized residuals|)")
plt.title("Scale-Location")
plt.show()

sns.scatterplot(x=res['leverage'],y=res['resid'],color='red')
plt.title("Residuals vs Leverage")
plt.xlabel("Leverage")
plt.ylabel("Residuals")
plt.show()

df['acceleration_horsepower'] = df['acceleration'] * df['horsepower']

df.corr()

"""9 f"""

X

X_new = X.loc[:,X.columns != 'const']

X_new.head()

"""log of all varibles"""

X_log =
np.log2(X_new[['displacement','horsepower','weight','acceleration']])

X_log.head()

X_log = sm.add_constant(X_log)

X_log.head()

model_log = sm.OLS(y,X_log).fit()

model_log.params

round(model_log.params,3)

model_log.summary()

y_pred_log = model_log.predict(X_log)

mean_absolute_error = metrics.mean_absolute_error(y,y_pred_log)
mean_squared_error = metrics.mean_squared_error(y,y_pred_log)
root_mean_squared_error =
np.sqrt(metrics.mean_squared_error(y,y_pred_log))

print("Mean Absolute Error : ",round(mean_absolute_error,3))
```

```
print("Mean Squared Error : ",round(mean_squared_error,3))
print("Root Mean Squared Error : ",round(root_mean_squared_error,3))

X_square =
np.power(X_new[['displacement','horsepower','weight','acceleration']],
2)

X_square.head()

X_square = sm.add_constant(X_square)

X_square.head()

model_square = sm.OLS(y,X_square).fit()

model_square.params

round(model_square.params,6)

model_square.summary()

y_pred_square = model_square.predict(X_square)

mean_absolute_error = metrics.mean_absolute_error(y,y_pred_square)
mean_squared_error = metrics.mean_squared_error(y,y_pred_square)
root_mean_squared_error =
np.sqrt(metrics.mean_squared_error(y,y_pred_square))

print("Mean Absolute Error : ",round(mean_absolute_error,3))
print("Mean Squared Error : ",round(mean_squared_error,3))
print("Root Mean Squared Error : ",round(root_mean_squared_error,3))
```