```
# Travel time for deliveries is dependent on distance travelled in miles, number of deliverie
#Following is a model for analysis of the collected data:
```

```
import pandas
import math
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
import statsmodels.api as sm
from statsmodels.stats.outliers_influence import variance_inflation_factor
```

```
df = pandas.read_csv('D:\DS\Data\TravelTime.csv')
```

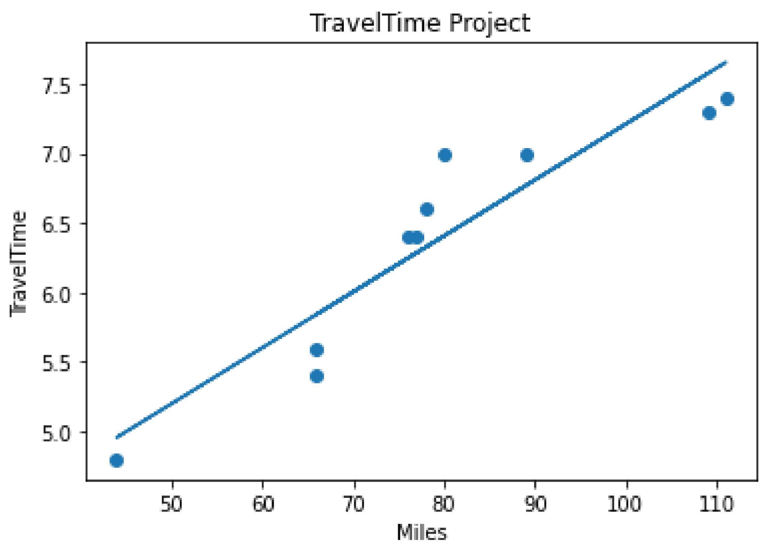
```
df
```

| | miles | deliveries | gasPrice | traveltime |
|---|---|---|---|---|
| **0** | 89 | 4 | 3.84 | 7.0 |
| **1** | 66 | 1 | 3.19 | 5.4 |
| **2** | 78 | 3 | 3.78 | 6.6 |
| **3** | 111 | 6 | 3.89 | 7.4 |
| **4** | 44 | 1 | 3.57 | 4.8 |
| **5** | 77 | 3 | 3.57 | 6.4 |
| **6** | 80 | 3 | 3.03 | 7.0 |
| **7** | 66 | 2 | 3.51 | 5.6 |
| **8** | 109 | 5 | 3.54 | 7.3 |
| **9** | 76 | 3 | 3.25 | 6.4 |

```
'''
Dependent variable = TravelTime
Independent variables:
    miles
    deliveries
    gasPrice
```

```
The following independent variables are compared with the dependent variable:

TravelTime vs miles
TravelTime vs deliveries
TravelTime vs gasPrice

For multicolinearity the following independent variables are considered:
miles vs deliveries
miles vs gasPrice
deliveries vs gasPrice

Dependent variable vs multiple independent variables
TravelTime vs (deliveries, gasPrice)
TravelTime vs (miles, gasPrice)
TravelTime vs (miles, deliveries)
TravelTime vs (miles, deliveries, gasPrice)

'''
```

```
# TravelTime vs miles

X = df[['miles']]
y = df['traveltime']

model = LinearRegression().fit(X, y)

plt.scatter(X, y)
plt.plot(X, model.predict(X))
plt.title('TravelTime Project')
plt.xlabel('Miles')
plt.ylabel('TravelTime')
plt.show()

print('co-efficient', model.coef_)
print ('constant', model.intercept_)
```




```
co-efficient [0.04025678]
constant 3.185560248999555
```

```
# TravelTime vs miles

X = sm.add_constant(X)
res = sm.OLS(y,X).fit()
print(res.summary())
```

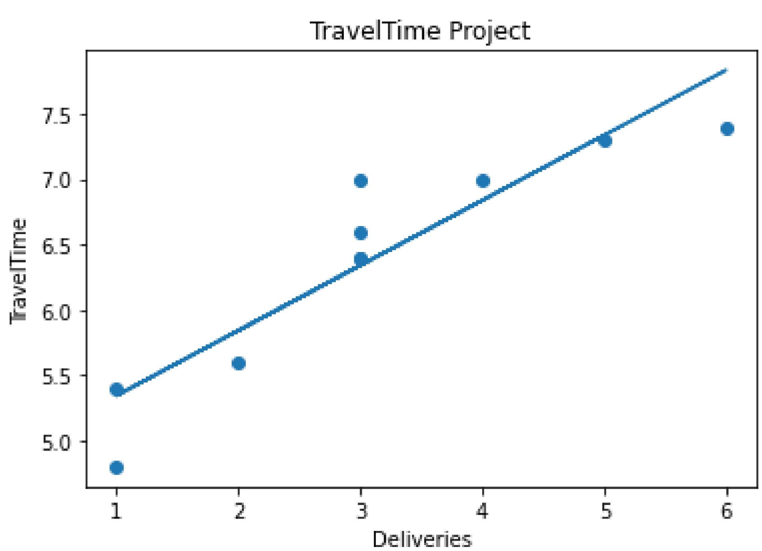
```
                            OLS Regression Results
==============================================================================
Dep. Variable:             traveltime   R-squared:                       0.862
Model:                            OLS   Adj. R-squared:                  0.844
Method:                 Least Squares   F-statistic:                     49.77
Date:                Sat, 16 Apr 2022   Prob (F-statistic):           0.000107
Time:                        12:56:04   Log-Likelihood:                -2.3532
No. Observations:                  10   AIC:                             8.706
Df Residuals:                       8   BIC:                             9.312
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          3.1856      0.467      6.822      0.000       2.109       4.262
miles          0.0403      0.006      7.055      0.000       0.027       0.053
==============================================================================
Omnibus:                        0.542   Durbin-Watson:                   2.608
Prob(Omnibus):                  0.763   Jarque-Bera (JB):                0.554
Skew:                           0.370   Prob(JB):                        0.758
Kurtosis:                       2.115   Cond. No.                         353.
==============================================================================
```

```
Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specifi
C:\Users\fakhi\AppData\Local\Programs\Python\Python310\lib\site-packages\scipy\stats\_st
  warnings.warn("kurtosistest only valid for n>=20 ... continuing "
```

```
print('standard error of regression', math.sqrt(1-(res.rsquared_adj))*(df.std()['traveltime']
# print(variance_inflation_factor(res.resid,1))
```

```
standard error of regression 0.3423088398195204
```

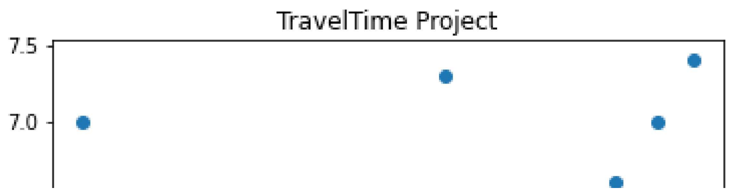
```
# TravelTime vs deliveries

X = df[['deliveries']]
y = df['traveltime']

model = LinearRegression().fit(X, y)

plt.scatter(X, y)
plt.plot(X, model.predict(X))
plt.title('TravelTime Project')
plt.xlabel('Deliveries')
plt.ylabel('TravelTime')
plt.show()
```




```
# TravelTime vs Deliveries
X = sm.add_constant(X)
res = sm.OLS(y,X).fit()
print(res.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:             traveltime   R-squared:                       0.840
Model:                            OLS   Adj. R-squared:                  0.820
Method:                 Least Squares   F-statistic:                     41.96
Date:                Sat, 16 Apr 2022   Prob (F-statistic):           0.000193
Time:                        12:56:05   Log-Likelihood:                -3.0794
No. Observations:                  10   AIC:                             10.16
Df Residuals:                       8   BIC:                             10.76
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          4.8454      0.265     18.261      0.000       4.234       5.457
deliveries     0.4983      0.077      6.478      0.000       0.321       0.676
==============================================================================
Omnibus:                        0.391   Durbin-Watson:                   1.970
Prob(Omnibus):                  0.822   Jarque-Bera (JB):                0.065
Skew:                           0.147   Prob(JB):                        0.968
Kurtosis:                       2.736   Cond. No.                         8.41
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specifi
C:\Users\fakhi\AppData\Local\Programs\Python\Python310\lib\site-packages\scipy\stats\_st
  warnings.warn("kurtosistest only valid for n>=20 ... continuing "
```

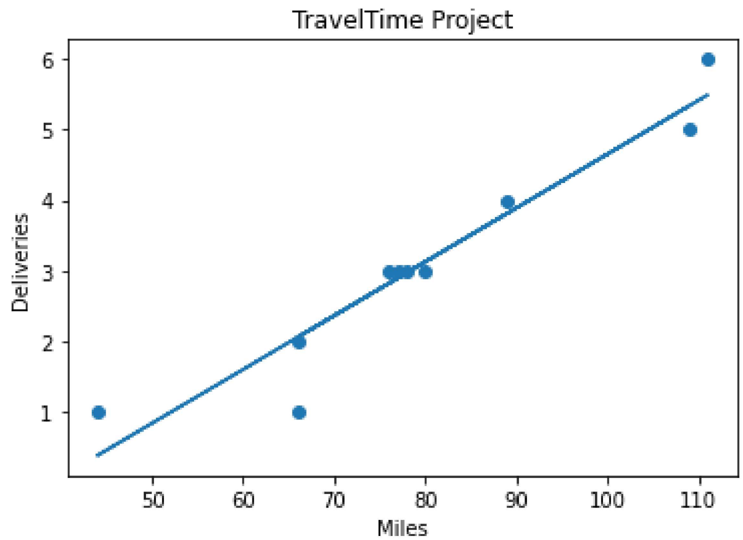
```
# TravelTime vs Gas Price

X = df[['gasPrice']]
y = df['traveltime']

model = LinearRegression().fit(X, y)

plt.scatter(X, y)
#plt.plot(X, model.predict(X))
plt.title('TravelTime Project')
plt.xlabel('Gas Price')
plt.ylabel('TravelTime')
plt.show()
```



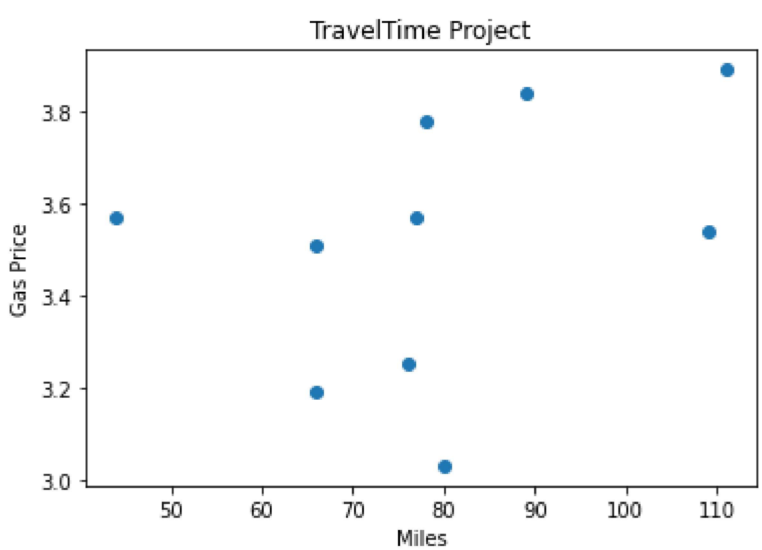

```
# TravelTime vs Gas Price
X = sm.add_constant(X)
res = sm.OLS(y,X).fit()
print(res.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:             traveltime   R-squared:                       0.071
Model:                            OLS   Adj. R-squared:                 -0.045
Method:                 Least Squares   F-statistic:                    0.6151
Date:                Sat, 16 Apr 2022   Prob (F-statistic):              0.455
Time:                        12:56:05   Log-Likelihood:                -11.868
No. Observations:                  10   AIC:                             27.74
Df Residuals:                       8   BIC:                             28.34
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          3.5365      3.649      0.969      0.361      -4.878      11.951
gasPrice       0.8113      1.034      0.784      0.455      -1.574       3.197
==============================================================================
Omnibus:                        1.232   Durbin-Watson:                   2.823
Prob(Omnibus):                  0.540   Jarque-Bera (JB):                0.765
Skew:                          -0.619   Prob(JB):                        0.682
Kurtosis:                       2.451   Cond. No.                         49.6
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specifi
C:\Users\fakhi\AppData\Local\Programs\Python\Python310\lib\site-packages\scipy\stats\_st
  warnings.warn("kurtosistest only valid for n>=20 ... continuing "
```

```
# miles vs deliveries

X = df[['miles']]
y = df['deliveries']

model = LinearRegression().fit(X, y)

plt.scatter(X, y)
plt.plot(X, model.predict(X))
plt.title('TravelTime Project')
plt.xlabel('Miles')
plt.ylabel('Deliveries')
plt.show()
```

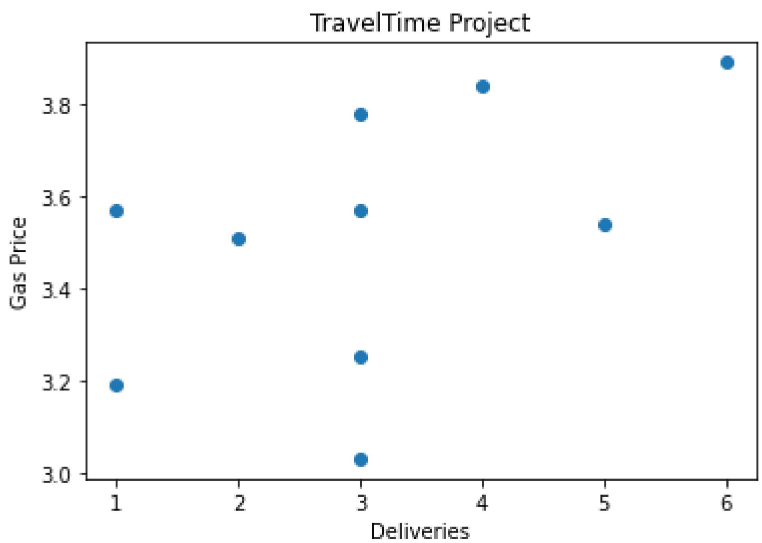



```
# miles vs Gas Price

X = df[['miles']]
y = df['gasPrice']

model = LinearRegression().fit(X, y)

plt.scatter(X, y)
#plt.plot(X, model.predict(X))
plt.title('TravelTime Project')
plt.xlabel('Miles')
plt.ylabel('Gas Price')
plt.show()
```




```
# deliveries vs Gas Price

X = df[['deliveries']]
y = df['gasPrice']
```

```
model = LinearRegression().fit(X, y)

plt.scatter(X, y)
#plt.plot(X, model.predict(X))
plt.title('TravelTime Project')
plt.xlabel('Deliveries')
plt.ylabel('Gas Price')
plt.show()
```




```
# TravelTime vs (deliveries, gasPrice)

X = df[['deliveries', 'gasPrice']]
y = df['traveltime']


X = sm.add_constant(X)
res = sm.OLS(y,X).fit()
print(res.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:             traveltime   R-squared:                       0.888
Model:                            OLS   Adj. R-squared:                  0.855
Method:                 Least Squares   F-statistic:                     27.63
Date:                Sat, 16 Apr 2022   Prob (F-statistic):           0.000476
Time:                        12:56:06   Log-Likelihood:                -1.3104
No. Observations:                  10   AIC:                             8.621
Df Residuals:                       7   BIC:                             9.529
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          7.3243      1.458      5.025      0.002       3.878      10.771
deliveries     0.5665      0.079      7.129      0.000       0.379       0.754
```

```
gasPrice      -0.7650      0.444     -1.724      0.128      -1.814       0.284
==============================================================================
Omnibus:                        1.321   Durbin-Watson:                   2.054
Prob(Omnibus):                  0.517   Jarque-Bera (JB):                0.811
Skew:                           0.339   Prob(JB):                        0.667
Kurtosis:                       1.781   Cond. No.                         71.9
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specifi
C:\Users\fakhi\AppData\Local\Programs\Python\Python310\lib\site-packages\scipy\stats\_st
  warnings.warn("kurtosistest only valid for n>=20 ... continuing "
```

```
# TravelTime vs (miles, gasPrice)
X = df[['miles', 'gasPrice']]
y = df['traveltime']


X = sm.add_constant(X)
res = sm.OLS(y,X).fit()
print(res.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:             traveltime   R-squared:                       0.866
Model:                            OLS   Adj. R-squared:                  0.828
Method:                 Least Squares   F-statistic:                     22.63
Date:                Sat, 16 Apr 2022   Prob (F-statistic):           0.000879
Time:                        12:56:06   Log-Likelihood:                -2.1863
No. Observations:                  10   AIC:                             10.37
Df Residuals:                       7   BIC:                             11.28
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          3.8676      1.482      2.609      0.035       0.362       7.373
miles          0.0414      0.006      6.445      0.000       0.026       0.057
gasPrice      -0.2191      0.449     -0.488      0.641      -1.282       0.844
==============================================================================
Omnibus:                        0.731   Durbin-Watson:                   2.740
Prob(Omnibus):                  0.694   Jarque-Bera (JB):                0.563
Skew:                           0.025   Prob(JB):                        0.755
Kurtosis:                       1.839   Cond. No.                     1.11e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specifi
[2] The condition number is large, 1.11e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
C:\Users\fakhi\AppData\Local\Programs\Python\Python310\lib\site-packages\scipy\stats\_st
  warnings.warn("kurtosistest only valid for n>=20 ... continuing "
```

```
# TravelTime vs (miles, deliveries)
X = df[['miles', 'deliveries']]
y = df['traveltime']


X = sm.add_constant(X)
res = sm.OLS(y,X).fit()
print(res.summary())
```

```
C:\Users\fakhi\AppData\Local\Programs\Python\Python310\lib\site-packages\scipy\stats\_st
  warnings.warn("kurtosistest only valid for n>=20 ... continuing "
                            OLS Regression Results
==============================================================================
Dep. Variable:             traveltime   R-squared:                       0.871
Model:                            OLS   Adj. R-squared:                  0.835
Method:                 Least Squares   F-statistic:                     23.72
Date:                Sat, 16 Apr 2022   Prob (F-statistic):           0.000763
Time:                        12:56:06   Log-Likelihood:                -1.9830
No. Observations:                  10   AIC:                             9.966
Df Residuals:                       7   BIC:                             10.87
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          3.7322      0.887      4.208      0.004       1.635       5.830
miles          0.0262      0.020      1.310      0.232      -0.021       0.074
deliveries     0.1840      0.251      0.733      0.487      -0.409       0.777
==============================================================================
Omnibus:                        1.340   Durbin-Watson:                   2.402
Prob(Omnibus):                  0.512   Jarque-Bera (JB):                0.867
Skew:                           0.654   Prob(JB):                        0.648
Kurtosis:                       2.393   Cond. No.                         670.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specifi
```

```
# VIF dataframe
vif_data = pandas.DataFrame()
vif_data["feature"] = X.columns

# calculating VIF for each feature
vif_data["VIF"] = [variance_inflation_factor(X.values, i)
                          for i in range(len(X.columns))]

print(vif_data)
```

```
      feature        VIF
0       const  63.263365
```

```
1       miles  11.593044
```

```
# TravelTime vs (miles, deliveries, gasPrice)
X = df[['miles', 'deliveries', 'gasPrice']]
y = df['traveltime']


X = sm.add_constant(X)
res = sm.OLS(y,X).fit()
print(res.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:             traveltime   R-squared:                       0.895
Model:                            OLS   Adj. R-squared:                  0.842
Method:                 Least Squares   F-statistic:                     16.99
Date:                Sat, 16 Apr 2022   Prob (F-statistic):            0.00245
Time:                        12:56:07   Log-Likelihood:               -0.98426
No. Observations:                  10   AIC:                             9.969
Df Residuals:                       6   BIC:                             11.18
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          6.2114      2.321      2.677      0.037       0.533      11.890
miles          0.0141      0.022      0.636      0.548      -0.040       0.068
deliveries     0.3832      0.300      1.277      0.249      -0.351       1.117
gasPrice      -0.6066      0.527     -1.152      0.293      -1.895       0.682
==============================================================================
Omnibus:                        2.874   Durbin-Watson:                   2.406
Prob(Omnibus):                  0.238   Jarque-Bera (JB):                1.204
Skew:                           0.457   Prob(JB):                        0.548
Kurtosis:                       1.567   Cond. No.                     1.79e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specifi
[2] The condition number is large, 1.79e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
C:\Users\fakhi\AppData\Local\Programs\Python\Python310\lib\site-packages\scipy\stats\_st
  warnings.warn("kurtosistest only valid for n>=20 ... continuing "
```

```
# VIF dataframe
vif_data = pandas.DataFrame()
vif_data["feature"] = X.columns

# calculating VIF for each feature
vif_data["VIF"] = [variance_inflation_factor(X.values, i)
                          for i in range(len(X.columns))]

print(vif_data)
```

```
      feature         VIF
0       const  453.235497
1       miles   14.936013
2  deliveries   17.353065
3    gasPrice    1.713803
```

```
df.corr()
```

| | miles | deliveries | gasPrice | traveltime |
|---|---|---|---|---|
| **miles** | 1.000000 | 0.955898 | 0.355796 | 0.928179 |
| **deliveries** | 0.955898 | 1.000000 | 0.498242 | 0.916443 |
| **gasPrice** | 0.355796 | 0.498242 | 1.000000 | 0.267212 |
| **traveltime** | 0.928179 | 0.916443 | 0.267212 | 1.000000 |

```
df2 = pandas.read_csv("D:\DS\Data\Summary.csv")
df2
```

| | F | P-value | std error | Rsq(Adj) | RSq(Pred) | miles | deliveries | gasPrice | VIF |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 49.77 | < 0.001 | 0.34230 | 0.8442 | 0.7907 | X | NaN | NaN | 1.00 |
| **1** | 41.96 | < 0.001 | 0.36809 | 0.8199 | 0.7027 | NaN | X | NaN | 1.00 |
| **2** | 0.62 | 0.455 | 0.88640 | 0.0000 | 0.0000 | NaN | NaN | X | 1.00 |
| **3** | 23.72 | 0.001 | 0.35264 | 0.8347 | 0.5995 | X | X | NaN | 11.59 |
| **4** | 22.63 | 0.001 | 0.35988 | 0.8278 | 0.6811 | X | NaN | X | 1.14 |
| **5** | 27.63 | < 0.001 | 0.32970 | 0.8555 | 0.7176 | NaN | X | X | 1.33 |
| **6** | 16.99 | 0.002 | 0.34469 | 0.8420 | 0.5749 | X | X | X | below |
| **7** | NaN | NaN | NaN | NaN | NaN | 14.94 | 17.35 | 1.71 | NaN |

```
'''
Conclusion

In terms of dependent variable TimeTravel, the following have a high correlation:

TravelTime vs Miles
TravelTime vs Number of Deliveries

Miles vs Number of Deliveries- Miles and Deliveries have multicollinearity and thus one of th
variance inflation factor VIF is 11.59 and low predicted R Squared
therefore one of miles or deliveries independent variables has to be dropped. Also two variab
```

```
miles and deliveries can be dropped.

Gas Price has a low correlation with Time Travel and a high standard error and can be taken
out from the list of independent variables. All models with gas price can be droppped.

Miles travelled has the highest Adjusted RSquared, RSquared predicted and a low srandard erro
This is the best model and should be used.

'''
```