In [88]:
```
import plotly.io as pio
pio.renderers.default = 'notebook'
```

Multiple Linear Regression with Dataset mtcars

We will be using a multiple linear regression model to try and find relationships between variables in mtcars, specifically variables
mpg (miles per gallon), **qsec** (1/4 mile time), and **am** (transmission type).

Multiple Linear Regression assumptions

1. The model has a linear relationship to the parameters (the regression coefficients).
2. Residuals have a covariance of zero and are uncorrelated for time series data.
3. The residuals are normally distributed and their anticipated mean value is zero.
4. Homogeneity of variance: the size of the error in our prediction doesn't change significantly across the values of the independent variable.
5. Independence of observations: the observations in the dataset were collected using statistically valid sampling methods, and there are no hidden relationships among observations.
6. There is no multicollinearity (no two independent variables are highly correlated).

In [89]:
```
#imports
import os
import numpy as np
import pandas as pandas
import matplotlib
import matplotlib.pyplot as plt
from sklearn import linear_model
import seaborn as sns
import scipy.stats as stats
matplotlib.style.use('ggplot')
from sklearn import metrics
from sklearn import datasets, linear_model
from sklearn.linear_model import LinearRegression
import statsmodels.api as sm
```

In [90]:
```
#load the dataset mtcars
mtcars = pandas.read_csv(r"C:\Users\Rbrig\Downloads\mtcars\mtcars\mtcars.csv")
```

In [91]:
```
#get a visual of the statistics of mtcars
mtcars.describe()
```

Out[91]:

	mpg	cyl	disp	hp	drat	wt	qsec	vs	
count	32.000000	32.000000	32.000000	32.000000	32.000000	32.000000	32.000000	32.000000	32.0
mean	20.090625	6.187500	230.721875	146.687500	3.596563	3.217250	17.848750	0.437500	0.4
std	6.026948	1.785922	123.938694	68.562868	0.534679	0.978457	1.786943	0.504016	0.4
min	10.400000	4.000000	71.100000	52.000000	2.760000	1.513000	14.500000	0.000000	0.0
25%	15.425000	4.000000	120.825000	96.500000	3.080000	2.581250	16.892500	0.000000	0.0
50%	19.200000	6.000000	196.300000	123.000000	3.695000	3.325000	17.710000	0.000000	0.0
75%	22.800000	8.000000	326.000000	180.000000	3.920000	3.610000	18.900000	1.000000	1.0
max	33.900000	8.000000	472.000000	335.000000	4.930000	5.424000	22.900000	1.000000	1.0

First Model

In [92]:
```
#creating a model with all the variables of mtcars

X = mtcars.iloc[:,2:]
Y = mtcars.mpg

X2 = sm.add_constant(X)
est = sm.OLS(Y, X2).fit()
print(est.summary2())
```

```
                 Results: Ordinary least squares
=================================================================
Model:              OLS              Adj. R-squared:     0.807
Dependent Variable: mpg              AIC:                161.7098
Date:               2024-02-27 13:40 BIC:                177.8329
No. Observations:   32               Log-Likelihood:     -69.855
Df Model:           10               F-statistic:        13.93
Df Residuals:       21               Prob (F-statistic): 3.79e-07
R-squared:          0.869            Scale:              7.0235
-------------------------------------------------------------------
          Coef.    Std.Err.     t      P>|t|    [0.025    0.975]
-------------------------------------------------------------------
const    12.3034   18.7179   0.6573   0.5181  -26.6226   51.2293
cyl      -0.1114    1.0450  -0.1066   0.9161   -2.2847    2.0618
disp      0.0133    0.0179   0.7468   0.4635   -0.0238    0.0505
hp       -0.0215    0.0218  -0.9868   0.3350   -0.0668    0.0238
drat      0.7871    1.6354   0.4813   0.6353   -2.6138    4.1881
wt       -3.7153    1.8944  -1.9612   0.0633   -7.6550    0.2243
qsec      0.8210    0.7308   1.1234   0.2739   -0.6988    2.3409
vs        0.3178    2.1045   0.1510   0.8814   -4.0588    4.6943
am        2.5202    2.0567   1.2254   0.2340   -1.7568    6.7973
gear      0.6554    1.4933   0.4389   0.6652   -2.4500    3.7608
carb     -0.1994    0.8288  -0.2406   0.8122   -1.9229    1.5241
-----------------------------------------------------------------
Omnibus:              1.907        Durbin-Watson:           1.861
Prob(Omnibus):        0.385        Jarque-Bera (JB):        1.747
Skew:                 0.521        Prob(JB):                0.418
Kurtosis:             2.526        Condition No.:           12213
=================================================================
Notes:
[1] Standard Errors assume that the covariance matrix of the
errors is correctly specified.
[2] The condition number is large, 1.22e+04. This might indicate
that there are strong multicollinearity or other numerical
problems.
```

Diagnosing a Multiple Linear Regression Model

1. Adjusted R^2

- R^2 measures how well the model fits the data. It interprets well in simple linear regression, but not in multiple linear regression.
- This is because as the number of independent variables increases, R^2 tends to increase even if there is no significant relation between the target and the explanatory variable.
- To fix this issue, R^2 is adjusted, and then defined as

$$R^2 = 1 - (1 - R^2)\frac{n-1}{n-p-1}$$

  where p is number of explanatory variables in the model.

The adjusted R^2 value of our multiple linear regression model is 0.807, which means about 80% of the variation in the dependent variable is explained by the model.

2. Hypothesis test, T-test, and P Value

- The t-test is used to check whether the relationship between the dependent and independent variable is statistically significant or not at a given significance level (is there a relationship between x and y).
- If the t-statistic value is higher than the t-critical value at a given significance level, we reject the null hypothesis.
  Otherwise, we fail to reject null hypothesis (that there is no relationship between x and y).
- We want to know the p value from our t-test.
- P-value is defined as the probability under the assumption of no effect or no difference (null hypothesis), of obtaining a result equal to or more extreme than what was actually observed.
- We want our p value to be less than 0.05, this means our data is significant.
- In the summary table of our multiple linear regression model, we see many variables have p-value > 0.05. This indicates multicollinearity in our model.

3. Multicollinearity and Variance Inflation Factor (VIF)

- Multicollinearity occurs when two or more independent variables in a regression model are strongly correlated with each other.
- Multicollinearity undermines the statistical significance of an independent variable.
- Variance Inflation Factor measures the magnitude of multicollinearity.
- VIF is given by $\frac{1}{1-R^2}$
- VIF is usually interpreted on a scale:
  VIF < 5: The variable has low multicollinearity (no significant issue).
  VIF between 5 and 10: The variable has moderate multicollinearity (requires further investigation).
  VIF > 10: The variable has high multicollinearity (serious issue, consider mitigation techniques).

Variance Inflation Factor Table

In [93]:
```
#create and view a table of varibles and their corresponding VIF
from statsmodels.stats.outliers_influence import variance_inflation_factor

vif=[variance_inflation_factor(X.values, j) for j in range(X.shape[1])]
vif_factor=pandas.DataFrame({'VIF': vif},index=X.columns)
vif_factor
```

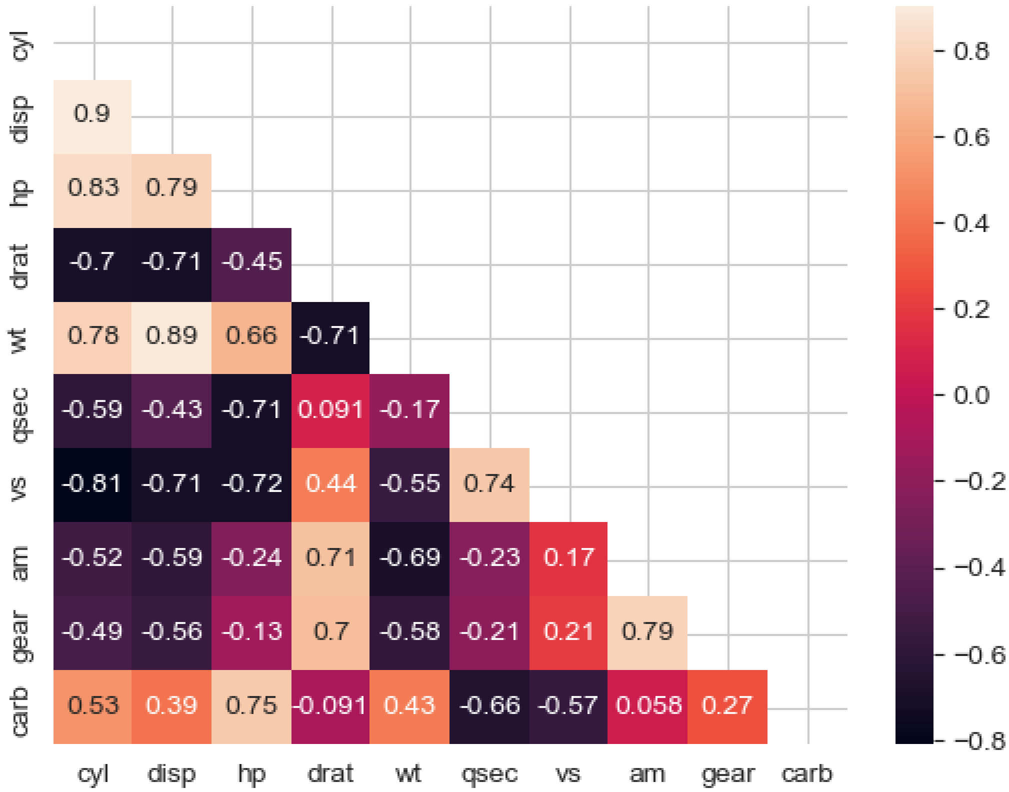
Out[93]:

	VIF
cyl	112.629828
disp	98.930791
hp	56.047781
drat	132.214353
wt	182.948049
qsec	317.534376
vs	8.752581
am	7.412020
gear	119.804879
carb	32.213836

Since the VIF of all the variables we are interested are > 4, we will check the correlation between features using a correlation matrix.

Correlation Matrix

In [94]:
```
#creating and displaying correlation matrix
sns.set_style("whitegrid")
mask = np.triu(np.ones_like(X.corr()))
corr = sns.heatmap(X.corr(), annot=True, mask=mask)
```



4. Selecting Variables

Our correlation matrix shows that there is a strong correlation between certain variables. Correlation coefficients whose magnitude are between 0.7 and 0.9 indicate variables which can be considered strongly correlated.

- cyl:
  vs, wt, drat, hp, disp
- hp:
  carb, vs, qsec
- drat:
  gear, am, wt

We will select one variable from each group to prevent multicollinearity.
We select the variables we are interested in (wt, qsec, am).

Variance Inflation Factor and P values Table

In [95]:
```
#create and display table with VIF and p values
vif_factor['p-values']=est.pvalues[1:]
vif_factor
```

Out[95]:

	VIF	p-values
cyl	112.629828	0.916087
disp	98.930791	0.463489
hp	56.047781	0.334955
drat	132.214353	0.635278
wt	182.948049	0.063252
qsec	317.534376	0.273941
vs	8.752581	0.881423
am	7.412020	0.233990
gear	119.804879	0.665206
carb	32.213836	0.812179

Second Model

In [96]:
```
#create multiple linear regression model with variables wt, qsec, am
X3 = X2[['const','wt','qsec','am']]
est2 = sm.OLS(Y, X3).fit()
print(est2.summary2())
```

```
                  Results: Ordinary least squares
=================================================================
Model:              OLS              Adj. R-squared:     0.834
Dependent Variable: mpg              AIC:                152.1194
Date:               2024-02-27 13:40 BIC:                157.9823
No. Observations:   32               Log-Likelihood:     -72.060
Df Model:           3                F-statistic:        52.75
Df Residuals:       28               Prob (F-statistic): 1.21e-11
R-squared:          0.850            Scale:              6.0459
------------------------------------------------------------------
           Coef.    Std.Err.     t      P>|t|     [0.025    0.975]
------------------------------------------------------------------
const      9.6178     6.9596   1.3819   0.1779   -4.6383   23.8739
wt        -3.9165     0.7112  -5.5069   0.0000   -5.3733   -2.4597
qsec       1.2259     0.2887   4.2467   0.0002    0.6346    1.8172
am         2.9358     1.4109   2.0808   0.0467    0.0457    5.8259
-----------------------------------------------------------------
Omnibus:              2.574        Durbin-Watson:           1.714
Prob(Omnibus):        0.276        Jarque-Bera (JB):        2.213
Skew:                 0.540        Prob(JB):                0.331
Kurtosis:             2.297        Condition No.:           296
=================================================================
Notes:
[1] Standard Errors assume that the covariance matrix of the
errors is correctly specified.
```

Adjusted R^2 of our new model is higher than the old one at 83.4%.

Multicollinearity is taken care of, and all the variables have p-values < 0.05, so they are significant.

5. Residual Analysis

Now the rest of our assumptions must be met.

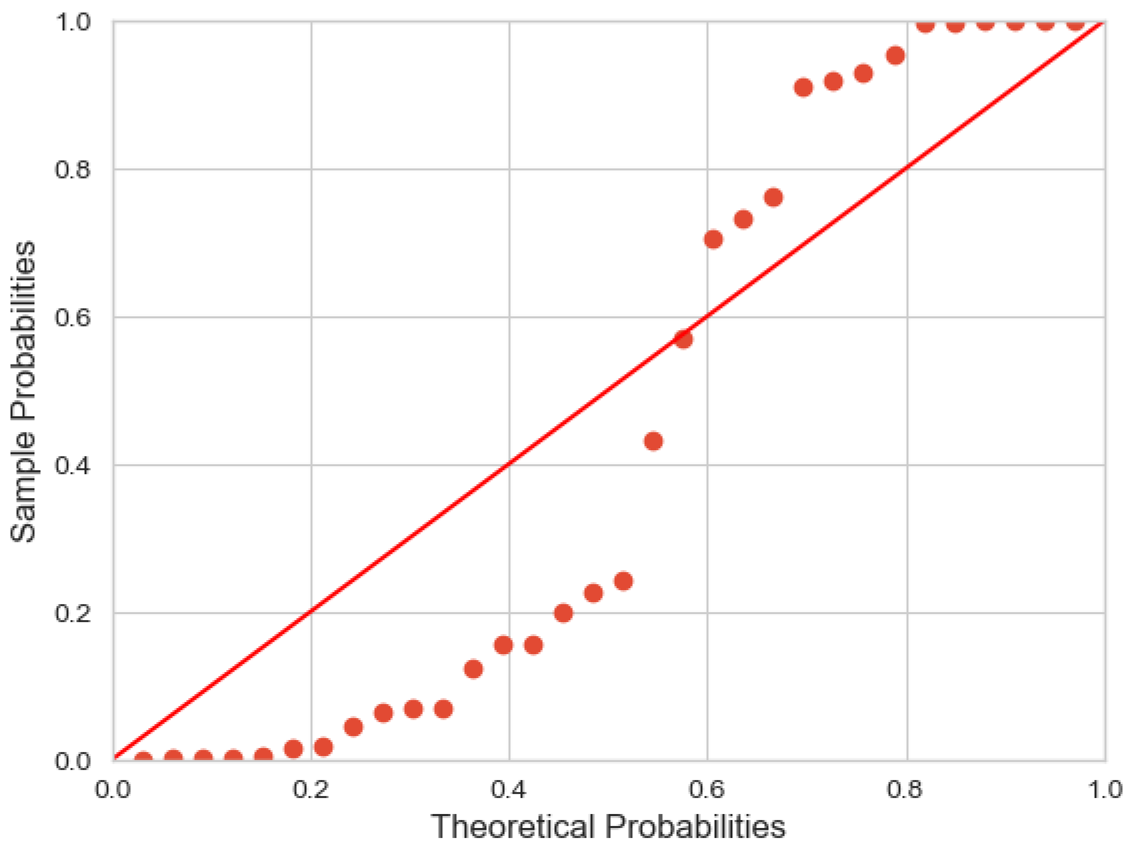
First, we make sure residuals are normally distributed

Plot of Probabilities vs Residuals

In [97]:

```
#test for normality of residuals
probplot = sm.ProbPlot(est2.resid)
plt.figure()
probplot.ppplot(line='45')
plt.show()
```

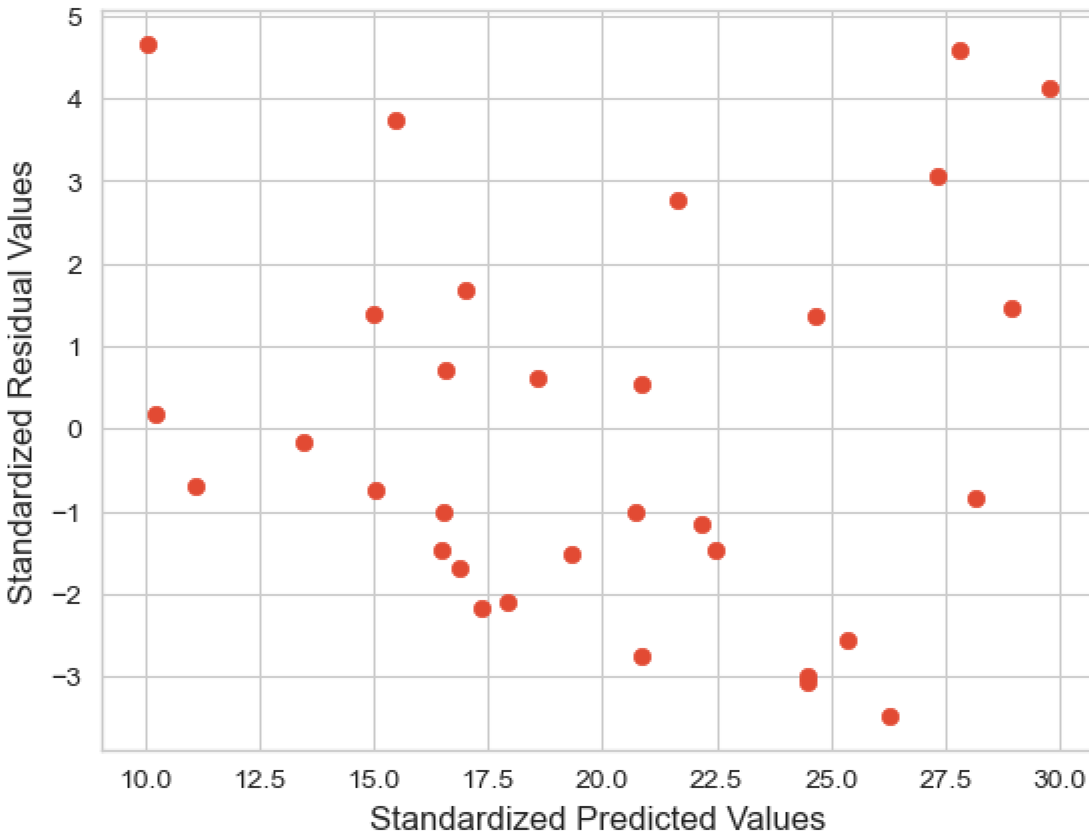
```
<Figure size 640x480 with 0 Axes>
```



Next, we test for homogeneity of variance.

Plot of Predicted Values vs Residual VaLues

In [98]:
```
def standardized_values(vals):
    return (vals - vals.mean())/vals.std()
plt.scatter(est2.fittedvalues, est2.resid)
plt.xlabel("Standardized Predicted Values")
plt.ylabel("Standardized Residual Values")
plt.show()
```



The residuals are random and do not follow any pattern. This means the residuals have constant variance (homogeneity of variance).

Conclusions

Now our multiple linear regression model is statistically significant, meets all our assumptions, and can be used to make predictions.

We get the equation
$(mpg) = 9.618 - 3.917(wt) + 1.226(qsec) + 2.936(am)$

Miles per gallon (**mpg**) is positively correlated with 1/4 mile time (**qsec**), and transmission type (**am**), and negatively correlated with weight (**wt**) of the car.

Sources

- mtcars download https://gist.github.com/seankross/a412dfbd88b3db70b74b