

Develop a program to implement Simple Linear Regression model and evaluate the model by verifying the performance

```
"""
import pandas as pd
import numpy as np
mba_salary_df = pd.read_excel( 'mba_salary.xlsx' )
mba_salary_df.head( 10 )

mba_salary_df.info()

import statsmodels.api as sm
X = sm.add_constant( mba_salary_df['Percentage in grade 10'] )
X.head(5)

Y = mba_salary_df['Salary']

from sklearn.model_selection import train_test_split
train_X, test_X, train_y, test_y = train_test_split( X,Y,train_size = 0.8,random_state = 100 )

mba_lm = sm.OLS( train_y, train_X ).fit()

print( mba_lm.params )

# Commented out IPython magic to ensure Python compatibility.
#import matplotlib.pyplot as plt
# import seaborn as sn
# %matplotlib inline

mba_salary_resid = mba_lm.resid
probplot = sm.ProbPlot(mba_salary_resid)
plt.figure( figsize = (8, 6))
probplot.ppplot( line='45' )
plt.title("Normal P-P Plot of Regression Standardized Residuals" )
plt.show()

"""OUTLIER ANALYSIS"""

from scipy.stats import zscore

mba_salary_df['z_score_salary'] = zscore( mba_salary_df.Salary )

mba_salary_df[ (mba_salary_df.z_score_salary > 3.0) | (mba_salary_df.z_score_salary < -3.0) ]

pred_y = mba_lm.predict( test_X )

from sklearn.metrics import r2_score, mean_squared_error
np.abs(r2_score(test_y, pred_y))

import numpy
np.sqrt(mean_squared_error(test_y, pred_y))
```