Estimation of Regression Model - AMS 578 Final Project (Spring 2023)

[1]Revanth Kumar Seeragaswamy (SBUID : 114779414),
[1]Department of Applied Mathematics and Statistics,
Stony Brook University, Stony Brook, NY.

(1) INTRODUCTION

The goal of this report is to estimate a regression model used by the TA to generate data and to analyse the relationship between the dependent and independent variables, with a particular focus on the interactions between the G and E variables, in order to investigate the natural development of depression in a representative sample of humans, as examined in the study conducted by Caspi et al. and other researchers [1].

(2) METHODS SECTION

In the methods section, the approach taken for statistical analysis is discussed, including the independent variables considered and methodological considerations, such as model validation settings and procedures. The regression model was obtained using a Jupyter notebook in Python, where the data files consisted of the dependent variable Y, eight environment independent variables (E_1 to E_8), and thirty genetic independent indicator variables (G_1 to G_{30}) for 1015 observations. As there is no missing data in the dataset, there is no need for data cleaning. The following steps were taken to obtain the final model:

1. The dependent variable underwent a box cox transformation to find the appropriate nonlinear transformation. The obtained fitted lambda is 0.59.
2. The correlations between the independent variables and dependent variable were examined, and those with a p-value less than 0.01 were selected for further analysis. The selected variables for only environment model were E_1, E_2, E_6 and the selected variables for gene-environment model were E_1, E_2, E_6, G_4, G_{15}, and G_{26}.
3. The Combinations of 2, 3 and 4 factors were created from the selected variables for gene-environment, resulting in a total of 209 combinations.
4. Backward stepwise regression was used to select the important independent variables. The selected 24 variables for gene-environment model were G_{26}, $E_2 . G_{26}$, $E_6 . G_4$, $G_{26} . G_{26}$, $E_1^2 . E_2$, $E_1 . E_6 . G_{15}$, $E_2^2 . E_6$, $E_2^2 . G_{26}$, $E_2 . E_6^2$, $E2 . G_{26}^2$, $E_6^2 . G_4$, $E_6 . G_4^2$, G_{26}^3, $E_1^2 . E_2 . E_6$, $E_1 . E_2^2 . G_4$, $E_1 . E_2 . E_6^2$, $E_1 . E_6 . G_{15}^2$, $E_2^3 . E_6$, $E_2^3 . G_{26}$, $E_2^2 . G_{26}^2$, $E_2 . G_{26}^3$, $E_6^2 . G_4^2$, $E_6 . G_4^3$, G_{26}^4.
5. To avoid multicollinearity problems, highly correlated independent variables were dropped based on a correlation heat map generated using the seaborn library. The final selected variables for gene-environment model were G_{26}, $E_6 . G_4$, $E_1^2 . E_2$ and $E_2^3 . E_6$.
6. The multiple linear regression model was then applied using the selected independent variables, and the model was validated using R-Squared, F-Static, MSE. The T-score, and p-value were used for coefficients of independent variable validation.

(3) RESULTS

The Fitted model using environment variable is as follows

$$Y_i = (3.4982 + 0.0818\,E_1 + 0.1161\,E_2 + 0.0527\,E_6)^{1.695}$$

The analysis showed that a model using the variables E_1,E_2 and E_6 was statistically not significant, with an R-squared value of 0.128 and an F-score of 49.54. These results accepts the null hypothesis that environment variable alone not associated with 'y' value depression. But for gene, gene-environment and environment-environment interactions analysis showed that the model using the variables G_{26}, $E_6.G_4$, $E_1^2.E_2$, and $E_2^3E_6$ was statistically significant, with an R-squared value of 0.770 and an F-score of 844.1. These results reject the null hypothesis. The t-scores for all coefficients exceeded the critical value of $t_{\alpha/2}$ = 2.576 at a confidence level of 99%. Additionally, the p-values were almost zero, further supporting the rejection of the null hypothesis that the coefficients are not significant. Please see the table below for more details. The Mean Squared Error for the model is 0.62.

Table1

OLS Regression Results

Dep. Variable:	y	R-squared:	0.770
Model:	OLS	Adj. R-squared:	0.769
Method:	Least Squares	F-statistic:	844.1
Date:	Mon, 17 Apr 2023	Prob (F-statistic):	3.18e-320
Time:	18:55:46	Log-Likelihood:	-211.84
No. Observations:	1015	AIC:	433.7
Df Residuals:	1010	BIC:	458.3
Df Model:	4		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	4.4513	0.027	167.776	0.000	4.399	4.503
G26	0.1310	0.020	6.663	0.000	0.092	0.170
E6_G4	0.1305	0.002	53.206	0.000	0.126	0.135
E1_E1_E2	0.0007	5.42e-05	12.697	0.000	0.001	0.001
E2_E2_E2_E6	5.377e-05	5.02e-06	10.705	0.000	4.39e-05	6.36e-05

Omnibus:	12.233	Durbin-Watson:	2.152
Prob(Omnibus):	0.002	Jarque-Bera (JB):	13.406
Skew:	-0.208	Prob(JB):	0.00123
Kurtosis:	3.378	Cond. No.	1.22e+04

Table 2

Analysis Of Variance Table

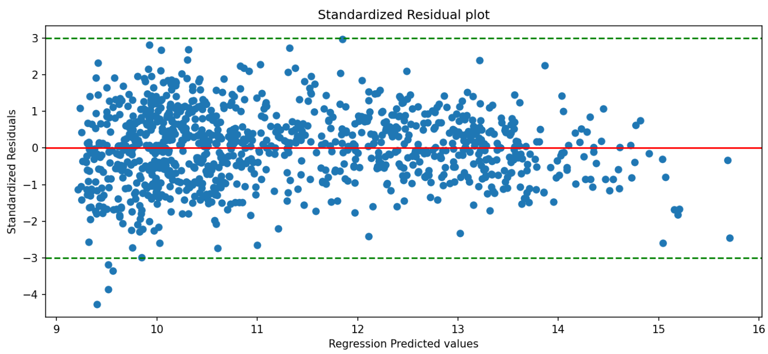
	df	sum_sq	mean_sq	F	PR(>F)
G26	1.0	4.341118	4.341118	48.601661	5.656611e-12
E6_G4	1.0	254.106063	254.106063	2844.883868	5.153245e-296
E1_E1_E2	1.0	32.908657	32.908657	368.433972	3.001901e-70
E2_E2_E2_E6	1.0	10.236362	10.236362	114.602781	2.087334e-25
Residual	1010.0	90.213568	0.089320	NaN	NaN

The Fitted model is as follows

$$Y_i = (4.4513 + 0.1310\, G_{26i} + 0.1305\, E_{6i}\, G_{4i} + 0.0007\, E_{1i}\, E_{1i}\, E_{2i} + 0.000054\, E_{2i}\, E_{2i}\, E_{6i})^{1.695}$$

(4) CONCLUSION

Based on the results, It can be concluded that gene, gene-environment and environment-environment interactions shows a positive association in predicting the natural development of depression (dependent variable 'Y'). Furthermore, the gene and gene-environment relationship exhibits a steeper positive slope, indicating that this factor has a greater influence on the dependent variable 'Y'. The model's performance is visualized in Figure 1 and Figure 2, which show the standardized residual plot and fitted vs predicted plot, respectively. The standardized residual plot reveals that all observations are within 3 standard deviations, indicating that the model is significant. However, four observations below the -3 standard deviation appear to be outliers or influential points, which requires further analysis to determine whether they should be kept or removed.



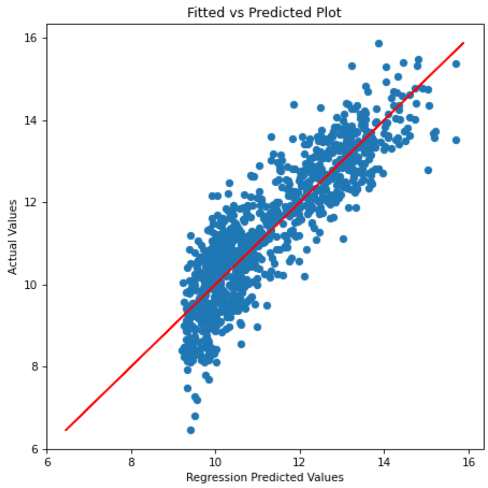


Figure 1 : Standardized Residual plot

Figure 2 : Fitted vs Predicted Plot

(5) REFERENCE

[1] Influence of Life Stress on Depression: Moderation by a Polymorphism in the 5-HTT Gene. Avshalom Caspi, *et al. Science* 301, 386 (2003); DOI: 10.1126/science.1083968.

[2] Interaction Between the Serotonin Transporter Gene (5-HTTLPR), Stressful Life Events, and Risk of Depression: A Meta-analysis. Neil Risch; Richard Herrell; Thomas Lehner; et al. *JAMA*. 2009;301(23):2462-2471 (doi:10.1001/jama.2009.878).

[3] Python code : https://www.geeksforgeeks.org

(6) APPENDIX

Python Code : Attached Jupyter Notebook in pdf format for your reference.

AMS 578 - Final Project - Spring 2023

Revanth Kumar Seerangaswamy (SBU ID : 114779414)

In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import statistics
import scipy
from scipy import stats
from scipy.stats import norm
from statsmodels.formula.api import ols
import statsmodels.api as sm
import matplotlib.pyplot as plt
from stepwise_regression import step_reg
from sklearn.metrics import mean_squared_error
import warnings
warnings.filterwarnings("ignore")
```

In [2]:
```
data = pd.read_csv('779414_project.csv')
data.head()
```

Out[2]:

	Y	E1	E2	E3	E4	E5	E6	E7	E8	G1	...	G21	G22	G23	G24	G25	G26	G27	G28	G29	G30
0	10.525167	7.854389	8.913405	6.565393	6.993341	9.292653	5.353894	5.140077	8.536922	0	...	1	1	1	1	1	0	1	0	0	0
1	13.415526	9.154240	9.843604	7.765238	7.403832	7.948206	7.361937	9.287343	7.048212	1	...	1	0	1	0	1	1	1	0	1	1
2	13.617077	6.482948	7.711028	7.804107	7.297177	5.665062	7.295711	9.542363	5.714542	0	...	1	1	1	1	0	1	0	0	0	0
3	9.776806	6.712727	9.601666	8.123204	7.883681	6.676015	6.949926	6.767072	8.551386	1	...	0	1	0	1	0	0	1	1	0	0
4	11.129903	7.065883	8.935642	5.711895	8.510285	6.712400	6.688294	6.821933	7.218909	1	...	0	0	1	0	1	0	0	0	0	0

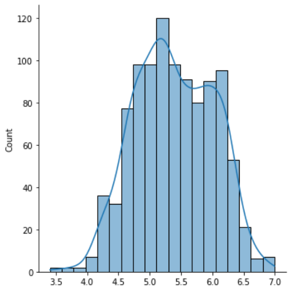
5 rows × 39 columns

In [3]:
```
#Box Cox trasformation for Normality
fitted_data, fitted_lambda = stats.boxcox(data['Y'])
print('Lambda Value : ', round(fitted_lambda,2))
```

```
Lambda Value :  0.59
```

In [4]:
```
#Ploting Trasformed Y data
import seaborn as sns
sns.displot(fitted_data, kde=True)
```

Out[4]:
```
<seaborn.axisgrid.FacetGrid at 0x7fe659b86a30>
```



In [5]:
```
# Finding correlation between Y variable and all X variable

corre = []
pval = []
col = []
for i in data:
    corr, pvalue = scipy.stats.pearsonr (data['Y'], data[i])
    col.append(i)
    corre.append(corr)
    pval.append(round(pvalue,4))
corr_pval = pd.DataFrame(
    {'Variables': col,
     'Correlation': corre,
     'Pvalue': pval
    })
```

In [6]:
```
#Selecting Variables with correlation P-vlaue less than 0.01
selected_Value =corr_pval[corr_pval.Pvalue < 0.01].Variables[1:]
selected_Value
```

Out[6]:
```
1      E1
2      E2
6      E6
12     G4
23    G15
34    G26
Name: Variables, dtype: object
```

In [10]:
```
#Y and X Variables
x = data[selected_Value]
y = fitted_data
print('Dimensions of Y Variable :',y.shape)
print('Dimensions of X Variables :',x.shape)
```

```
Dimensions of Y Variable : (1015,)
Dimensions of X Variables : (1015, 6)
```

In [17]:
```
lm=ols('y~E1+E2+E6', data=x).fit()
print(lm.summary())
table=sm.stats.anova_lm(lm)
print(table)
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.128
Model:                            OLS   Adj. R-squared:                  0.126
Method:                 Least Squares   F-statistic:                     49.54
```

In [17]:
```python
lm=ols('y~E1+E2+E6', data=x).fit()
print(lm.summary())
table=sm.stats.anova_lm(lm)
print(table)
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.128
Model:                            OLS   Adj. R-squared:                  0.126
Method:                 Least Squares   F-statistic:                     49.54
Date:                Tue, 02 May 2023   Prob (F-statistic):           7.14e-30
Time:                        14:08:53   Log-Likelihood:                -887.54
No. Observations:                1015   AIC:                             1783.
Df Residuals:                    1011   BIC:                             1803.
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      3.4982      0.164     21.347      0.000       3.177       3.820
E1             0.0818      0.012      6.570      0.000       0.057       0.106
E2             0.1161      0.013      9.128      0.000       0.091       0.141
E6             0.0527      0.013      4.068      0.000       0.027       0.078
==============================================================================
Omnibus:                      178.225   Durbin-Watson:                   2.081
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               40.137
Skew:                           0.083   Prob(JB):                     1.92e-09
Kurtosis:                       2.040   Cond. No.                         119.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
              df      sum_sq    mean_sq          F        PR(>F)
E1           1.0   16.075440  16.075440  47.577690  9.306924e-12
E2           1.0   28.545279  28.545279  84.484060  2.146495e-19
E6           1.0    5.590702   5.590702  16.546525  5.116427e-05
Residual  1011.0  341.594347   0.337878        NaN           NaN
```

In [11]:
```python
#Function to create combinations
from itertools import combinations_with_replacement

#Function to create 2 combinations
def f2(df):
    #with loop
    new_df = pd.DataFrame()
    for p in combinations_with_replacement(df.columns,2):
            title = p
            new_df[title] = df[p[0]]*df[p[1]]
    return new_df

#Function to create 3 combinations
def f3(df):
    #with loop
    new_df = pd.DataFrame()
    for p in combinations_with_replacement(df.columns,3):
            title = p
            new_df[title] = df[p[0]]*df[p[1]]*df[p[2]]
    return new_df

#Function to create 4 combinations
def f4(df):
    #with loop
    new_df = pd.DataFrame()
    for p in combinations_with_replacement(df.columns,4):
            new_df[p] = df[p[0]]*df[p[1]]*df[p[2]]*df[p[3]]
    return new_df
```

In [12]:
```python
# 2 combinations with replacement
x_2 = f2(x)
print('Dimensions of 2 combinations for GE        :',x_2.shape)

# 3 combinations with replacement
x_3 = f3(x)
print('Dimensions of 3 combinations for GE        :',x_3.shape)

# 4 combinations with replacement
x_4 = f4(x)
print('Dimensions of 4 combinations foe GE        :',x_4.shape)

x_all = pd.concat([x,x_2,x_3,x_4], axis=1)
print('Dimensions of all combinations added for GE:',x_all.shape)
```

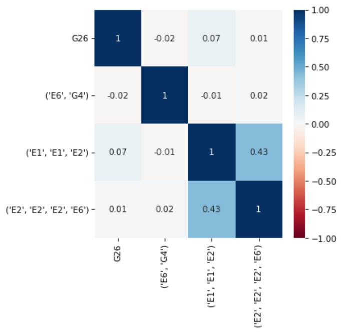
```
Dimensions of 2 combinations for GE        : (1015, 21)
Dimensions of 3 combinations for GE        : (1015, 56)
Dimensions of 4 combinations foe GE        : (1015, 126)
Dimensions of all combinations added for GE: (1015, 209)
```

In [13]:
```python
# Selecting the important independent variables using Stepwise Backward regression
backselect = step_reg.backward_regression(x_all, y, 0.01,verbose=False)
backselect
```

Out[13]:
```
['G26',
 ('E2', 'G26'),
 ('E6', 'G4'),
 ('G26', 'G26'),
 ('E1', 'E1', 'E2'),
 ('E1', 'E6', 'G15'),
 ('E2', 'E2', 'E6'),
 ('E2', 'E2', 'G26'),
 ('E2', 'E6', 'E6'),
 ('E2', 'G26', 'G26'),
 ('E6', 'E6', 'G4'),
 ('E6', 'G4', 'G4'),
 ('G26', 'G26', 'G26'),
 ('E1', 'E1', 'E2', 'E6'),
 ('E1', 'E2', 'E2', 'G4'),
 ('E1', 'E2', 'E6', 'E6'),
 ('E1', 'E6', 'G15', 'G15'),
 ('E2', 'E2', 'E2', 'E6'),
 ('E2', 'E2', 'E2', 'G26'),
 ('E2', 'E2', 'G26', 'G26'),
 ('E2', 'G26', 'G26', 'G26'),
 ('E6', 'E6', 'G4', 'G4'),
 ('E6', 'G4', 'G4', 'G4'),
 ('G26', 'G26', 'G26', 'G26')]
```

In [14]:
```python
#Droping most correlated Variables
x_selected = x_all[backselect]
x_selected.drop(x_selected.columns[[1,3,5,6,7,8,9,10,11,12,13,14,15,16,18,19,20,21,22,23]], axis=1, inplace=True)
fig, ax = plt.subplots(figsize=(5, 5),dpi=75)
corr = round(x_selected.corr(),2)
sns.heatmap(corr, cmap='RdBu', vmin=-1, vmax=1, annot=True)
```

Out[14]: <AxesSubplot:>



In [15]:
```python
# Frist 5 rows of selected independent variables
```

In [15]:
```python
# Frist 5 rows of selected independent variables
x_selected.head()
```

Out[15]:

	G26	(E6, G4)	(E1, E1, E2)	(E2, E2, E2, E6)
0	0	0.000000	549.880720	3791.409942
1	1	7.361937	824.895055	7021.898765
2	1	7.295711	324.083860	3345.064599
3	0	0.000000	432.657807	6152.052244
4	0	0.000000	446.127128	4771.913824

In [16]:
```python
X = pd.DataFrame({'G26': x_selected.iloc[:,0], 'E6_G4': x_selected.iloc[:,1],
                  'E1_E1_E2': x_selected.iloc[:,2],'E2_E2_E2_E6': x_selected.iloc[:,3]})
```

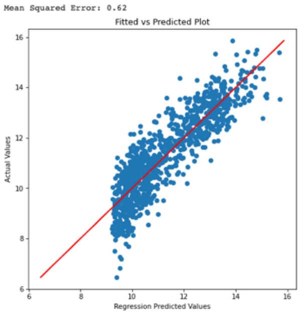
In [14]:
```python
from statsmodels.formula.api import ols
lm=ols('y~G26+E6_G4+E1_E1_E2+E2_E2_E2_E6', data=X).fit()
print(lm.summary())
table=sm.stats.anova_lm(lm)
print(table)
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.770
Model:                            OLS   Adj. R-squared:                  0.769
Method:                 Least Squares   F-statistic:                     844.1
Date:                Tue, 25 Apr 2023   Prob (F-statistic):          3.18e-320
Time:                        14:13:15   Log-Likelihood:                -211.84
No. Observations:                1015   AIC:                             433.7
Df Residuals:                    1010   BIC:                             458.3
Df Model:                           4
Covariance Type:            nonrobust
===============================================================================
                  coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------
Intercept       4.4513      0.027    167.776      0.000       4.399       4.503
G26             0.1310      0.020      6.663      0.000       0.092       0.170
E6_G4           0.1305      0.002     53.206      0.000       0.126       0.135
E1_E1_E2        0.0007   5.42e-05     12.697      0.000       0.001       0.001
E2_E2_E2_E6  5.377e-05   5.02e-06     10.705      0.000    4.39e-05    6.36e-05
==============================================================================
Omnibus:                       12.233   Durbin-Watson:                   2.152
Prob(Omnibus):                  0.002   Jarque-Bera (JB):               13.406
Skew:                          -0.208   Prob(JB):                      0.00123
Kurtosis:                       3.378   Cond. No.                     1.22e+04
==============================================================================

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.
                 df      sum_sq     mean_sq            F         PR(>F)
G26             1.0    4.341118    4.341118    48.601661   5.656611e-12
E6_G4           1.0  254.106063  254.106063  2844.883868  5.153245e-296
E1_E1_E2        1.0   32.908657   32.908657   368.433972   3.001901e-70
E2_E2_E2_E6     1.0   10.236362   10.236362   114.602781   2.087334e-25
Residual     1010.0   90.213568    0.089320          NaN            NaN
```

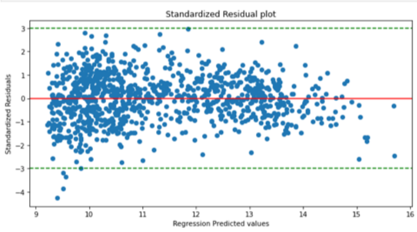
In [15]:
```python
#Predicting Dependent Variable Y
y_pred = lm.predict(X)

#Transforming predicted dependent Variable Y
if fitted_lambda == 0:
    y_pred_trans = np.exp(y_pred)
else:
    y_pred_trans = (y_pred * fitted_lambda + 1) ** (1 / fitted_lambda)
```

In [16]:
```python
#Plot fitted vs Actual values
print('Mean Squared Error:', round(mean_squared_error(data['Y'], y_pred_trans),3))
fig, ax = plt.subplots(figsize=(7, 7),dpi=75)
plt.scatter(y_pred_trans, data['Y'])
plt.plot(data['Y'], data['Y'], color='red')
plt.xlabel('Regression Predicted Values')
plt.ylabel('Actual Values')
plt.title('Fitted vs Predicted Plot')
plt.savefig('plot_1.png')
plt.show()
```

```
Mean Squared Error: 0.62
```



In [17]:
```python
#Cal Standardized Residuals
std_resid = lm.get_influence().resid_studentized_internal

fig, ax = plt.subplots(figsize=(10, 5),dpi=75)

plt.scatter(y_pred_trans, std_resid)
plt.axhline(y=0, color='r', linestyle='-')
plt.axhline(y=3, color='g', linestyle='--')
plt.axhline(y=-3, color='g', linestyle='--')
plt.xlabel('Regression Predicted values')
plt.ylabel('Standardized Residuals')
plt.title('Standardized Residual plot')
plt.savefig('plot_2.png')
plt.show()
```



In [18]:
```python
residuals = lm.resid
# create the probability plot
fig, ax = plt.subplots(figsize=(10, 5),dpi=75)
stats.probplot(residuals, plot=ax, fit=True)

# customize the plot
ax.set_title('Probability Plot of Residuals')
ax.set_xlabel('Theoretical Quantiles')
ax.set_ylabel('Sample Quantiles')
plt.show()
```