Module 3 Homework

ISE-529 Predictive Analytics

In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
```

1a) Read the file "HW Problem 1 Dataset.csv" into a dataframe and convert the category column X6 into binary dummary variables. Display the first three rows of the resulting dataset.

In [2]:
```
prob_1_dataset = pd.read_csv('HW Problem 1 Dataset.csv')
prob_1_dataset['Blue'] = pd.get_dummies(prob_1_dataset['X6'])['Blue']
prob_1_dataset['Red'] = pd.get_dummies(prob_1_dataset['X6'])['Red']
prob_1_dataset = prob_1_dataset.drop('X6', axis = 1)
prob_1_dataset = prob_1_dataset[['X1', 'X2', 'X3', 'X4', 'X5', 'Blue', 'Red', 'Y']]
prob_1_dataset.head(10)
```

Out[2]:

	X1	X2	X3	X4	X5	Blue	Red	Y
0	11	47	18	3	56	0	0	153.157223
1	19	91	11	93	1	0	1	809.384179
2	48	33	36	31	22	0	1	395.466944
3	4	86	43	68	98	0	0	892.610788
4	82	52	37	65	100	1	0	476.573108
5	41	11	6	88	37	0	0	797.891711
6	29	96	83	12	4	1	0	871.984975
7	22	71	44	89	44	0	0	952.367041
8	12	67	39	67	12	1	0	343.993916
9	2	45	68	96	5	0	0	1297.651894

1b) Using statsodels, perform a regression for Y using X1 through X5 and your dummy variables display the fit summary below.

In [3]:
```
import statsmodels.api as sm
model_1 = sm.OLS(prob_1_dataset['Y'], sm.add_constant(prob_1_dataset.drop('Y',1)))
model_1.fit().summary()
```

Out[3]:

OLS Regression Results

Dep. Variable:	Y	**R-squared:**	0.892
Model:	OLS	**Adj. R-squared:**	0.891
Method:	Least Squares	**F-statistic:**	1170.
Date:	Tue, 26 Jul 2022	**Prob (F-statistic):**	0.00
Time:	10:53:36	**Log-Likelihood:**	-6515.8
No. Observations:	1000	**AIC:**	1.305e+04
Df Residuals:	992	**BIC:**	1.309e+04
Df Model:	7		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	19.4186	22.324	0.870	0.385	-24.389	63.226
X1	4.0399	0.181	22.331	0.000	3.685	4.395
X2	0.0312	0.179	0.174	0.862	-0.321	0.383
X3	13.0721	0.181	72.131	0.000	12.717	13.428
X4	4.8075	0.180	26.651	0.000	4.454	5.162
X5	0.0114	0.182	0.063	0.950	-0.346	0.369
Blue	-473.6667	11.629	-40.732	0.000	-496.487	-450.847
Red	-90.8354	14.185	-6.404	0.000	-118.671	-63.000

Omnibus:	3.577	**Durbin-Watson:**	1.920
Prob(Omnibus):	0.167	**Jarque-Bera (JB):**	3.649
Skew:	0.139	**Prob(JB):**	0.161
Kurtosis:	2.899	**Cond. No.**	514.

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

1c) Investigating the resulting coefficient p-values, Which predictors appear to not have a statistically significant relationship to the response variable?

Variables X2 and X5 do not appear to have a relationship to the response variable.

1d) Drop any predictors that you found not to have a relationship with the response and display the first 10 rows of the resulting dataframe.

In [4]:

```
prob_1_dataset_2 = prob_1_dataset.drop(['X2', 'X5'], 1)
prob_1_dataset_2.head(10)
```

Out[4]:

	X1	X3	X4	Blue	Red	Y
0	11	18	3	0	0	153.157223
1	19	11	93	0	1	809.384179
2	48	36	31	0	1	395.466944
3	4	43	68	0	0	892.610788
4	82	37	65	1	0	476.573108
5	41	6	88	0	0	797.891711
6	29	83	12	1	0	871.984975
7	22	44	89	0	0	952.367041
8	12	39	67	1	0	343.993916
9	2	68	96	0	0	1297.651894

1e) Re-run the regression without the irrelevant variables and display the fit summary

In [5]:

```
X = prob_1_dataset_2.drop(["Y"], axis = 1)
y = prob_1_dataset_2['Y']
model_2 = sm.OLS(y, sm.add_constant(X))
model_2.fit().summary()
```

Out[5]:

OLS Regression Results

Dep. Variable:	Y	**R-squared:**	0.892
Model:	OLS	**Adj. R-squared:**	0.891
Method:	Least Squares	**F-statistic:**	1641.
Date:	Tue, 26 Jul 2022	**Prob (F-statistic):**	0.00
Time:	10:53:36	**Log-Likelihood:**	-6515.9
No. Observations:	1000	**AIC:**	1.304e+04
Df Residuals:	994	**BIC:**	1.307e+04
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	21.5910	18.091	1.193	0.233	-13.911	57.093
X1	4.0403	0.181	22.375	0.000	3.686	4.395
X3	13.0704	0.181	72.310	0.000	12.716	13.425
X4	4.8084	0.180	26.693	0.000	4.455	5.162
Blue	-473.6877	11.608	-40.806	0.000	-496.468	-450.908
Red	-90.8587	14.169	-6.412	0.000	-118.664	-63.053

Omnibus:	3.549	**Durbin-Watson:**	1.919
Prob(Omnibus):	0.170	**Jarque-Bera (JB):**	3.621
Skew:	0.138	**Prob(JB):**	0.164
Kurtosis:	2.898	**Cond. No.**	346.

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

1f) Write the full regression equation

$$Y = 21.59 + 4.04X1 + 13.07X2 + 4.81X4 - 473.69 * Blue - 90.86 * Red$$

1g) Write the equation for the observations where the "color" category is yellow:

$Y = 21.59 + 4.04X1 + 13.07X2 + 4.81X4$

1h) Write the equation for the observations where the "color" category is blue:

$Y = -452.1 + 4.04X1 + 13.07X2 + 4.81X4$

Write the equation for the observations where the "color" category is red:

$Y = -69.3 + 4.04X1 + 13.07X2 + 4.81X4$

1i) Now, use the sklearn library to run the same regression and display the resulting model coefficients

In [6]:
```
from sklearn.linear_model import LinearRegression
model_3 = LinearRegression(fit_intercept = True)
```

In [7]:
```
X = prob_1_dataset_2.drop(["Y"], axis = 1)
y = prob_1_dataset_2['Y']
model_3.fit(X,y)
print('Intercept:', model_3.intercept_)
print('Coefficients:', model_3.coef_)
```

```
Intercept: 21.590975945275886
Coefficients: [   4.04032489   13.07044352    4.8083891  -473.68774998  -90.85868231]
```

1j) Calculate and display the following fit assessment statistics: R^2, Mean Squared Error, Mean Absolute Error, and Max Error

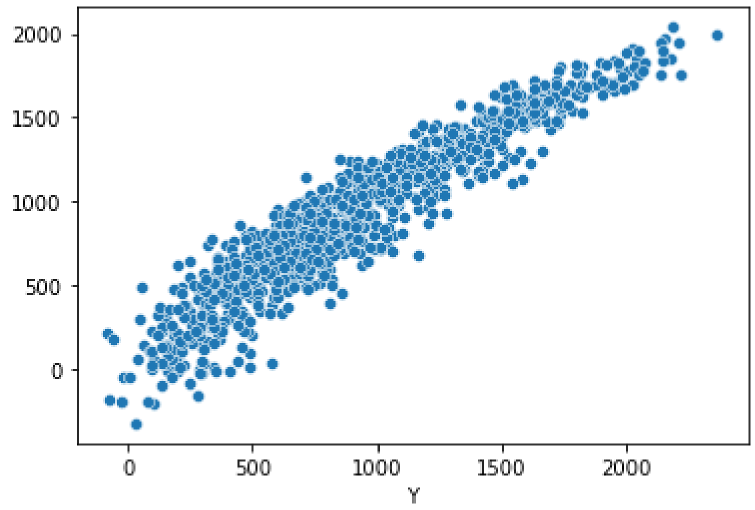
In [8]:
```
y_hat = model_3.predict(X)
from sklearn import metrics
print("R2: ", metrics.r2_score(y,y_hat))
print("MSE: ", metrics.mean_squared_error(y,y_hat))
print("MAE: ", metrics.mean_absolute_error(y,y_hat))
print("Max error: ", metrics.max_error(y,y_hat))
```

```
R2:  0.8919740759220801
MSE:  26738.19374639029
MAE:  130.86268562302584
Max error:  540.8391996665018
```

1k) Using Seaborn, create a scatterplot of the actual values of Y vs the predicted values of Y

In [9]:
```
sns.scatterplot(x = y, y = y_hat)
```

Out[9]: <AxesSubplot:xlabel='Y'>

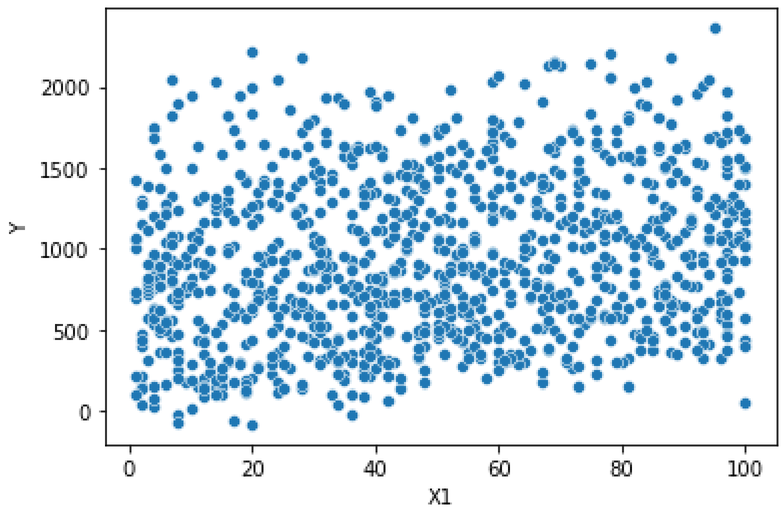


Investigate adding nonlinear terms

1L) Now, create one scatterplot for each numeric predictor (not including dummy variables) against the response variables:

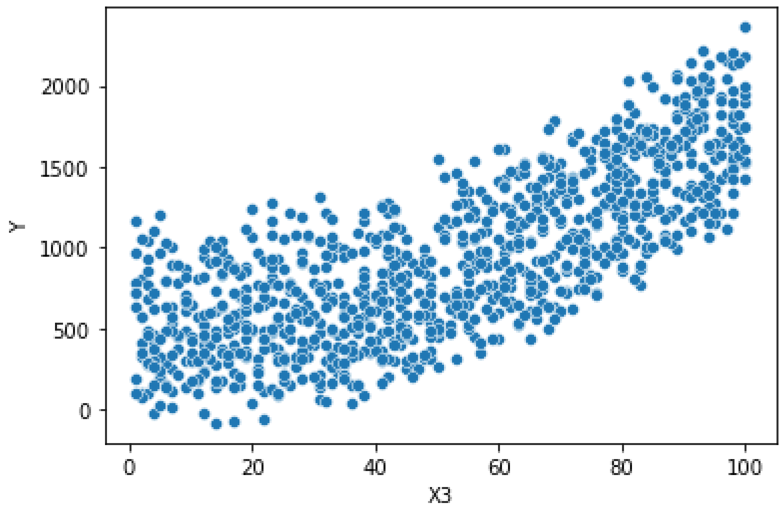
In [10]:
```
sns.scatterplot(x = X['X1'] , y=y)
```

Out[10]: <AxesSubplot:xlabel='X1', ylabel='Y'>



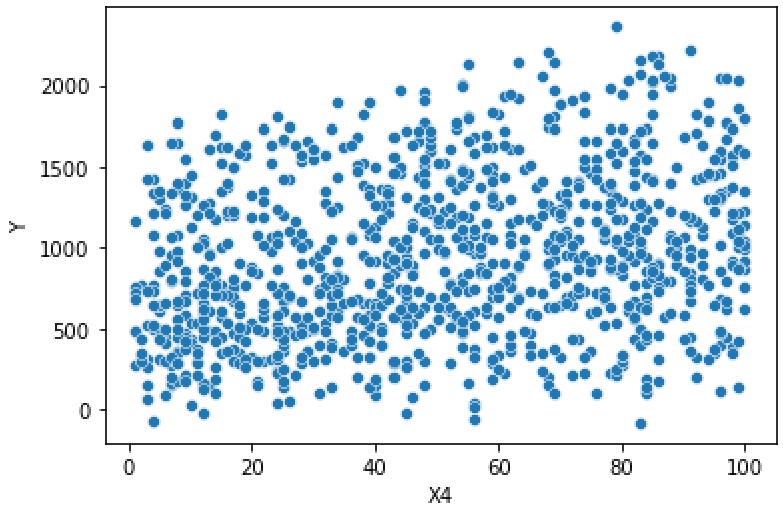
In [11]:
```
sns.scatterplot(x = X['X3'] , y=y)
```

Out[11]: <AxesSubplot:xlabel='X3', ylabel='Y'>



In [12]:
```
sns.scatterplot(x = X['X4'] , y=y)
```

Out[12]: <AxesSubplot:xlabel='X4', ylabel='Y'>



1M) Which predictor or predictors appear to have a nonlinear relationship with the response variable?

X3

1n) Try adding a squared term of any predictors that appear to have a nonlinear relationship. Re-run the regression and display the resulting coefficients and assessment statistics (R^2, Mean Squared Error, Mean Absolute Error, and Max Error)

In [13]:
```
prob_1_dataset_3 = prob_1_dataset_2.copy()
prob_1_dataset_3['X3_2'] = prob_1_dataset_3['X3']**2
X = prob_1_dataset_3.drop('Y',1)
```

In [14]:
```
X
```

Out[14]:

	X1	X3	X4	Blue	Red	X3_2
0	11	18	3	0	0	324
1	19	11	93	0	1	121
2	48	36	31	0	1	1296
3	4	43	68	0	0	1849
4	82	37	65	1	0	1369
...	...	...	...	...	...	...
995	54	86	23	0	0	7396
996	15	77	100	0	1	5929
997	17	27	91	0	1	729
998	57	12	2	0	0	144
999	61	25	86	1	0	625

1000 rows × 6 columns

In [15]:
```
model_4 = LinearRegression(fit_intercept = True)
model_4.fit(X,y)
print("Intercept:", model_4.intercept_)
print("Coefficients:", model_4.coef_)
```

```
Intercept: 268.14577682358595
Coefficients: [ 4.00826126e+00 -1.85296119e+00  4.88222728e+00 -4.60386174e+02
 -7.25791090e+01  1.46948111e-01]
```

In [16]:
```
y_hat = model_4.predict(X)
print("R2: ", metrics.r2_score(y,y_hat))
print("MSE: ", metrics.mean_squared_error(y,y_hat))
```

```
print("MAE: ", metrics.mean_absolute_error(y,y_hat))
print("Max error: ", metrics.max_error(y,y_hat))
```

```
R2:  0.9393009387730067
MSE:  15024.016440171197
MAE:  97.87452904328543
Max error:  377.7320018040741
```

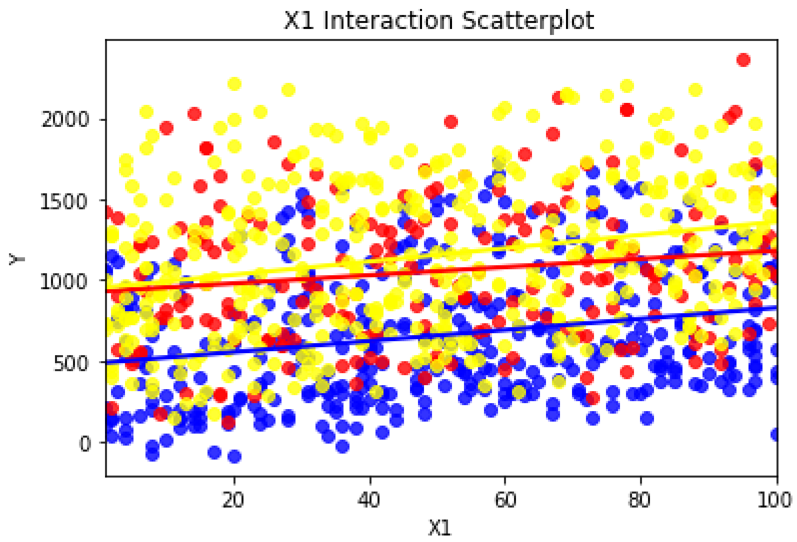
Investigate adding interaction effects

1o) For each numeric predictor, plot a scatterplot against the response variable color coding and the points according to their category values and include regresion lines

In [17]:
```
blue_observations = prob_1_dataset_3.loc[prob_1_dataset_3['Blue'] == 1]
red_observations = prob_1_dataset_3.loc[prob_1_dataset_3['Red'] == 1]
yellow_observations = prob_1_dataset_3.loc[np.logical_and(prob_1_dataset_3['Blue'] == 0, prob_1_dataset_3['Red'] == 0)]
```

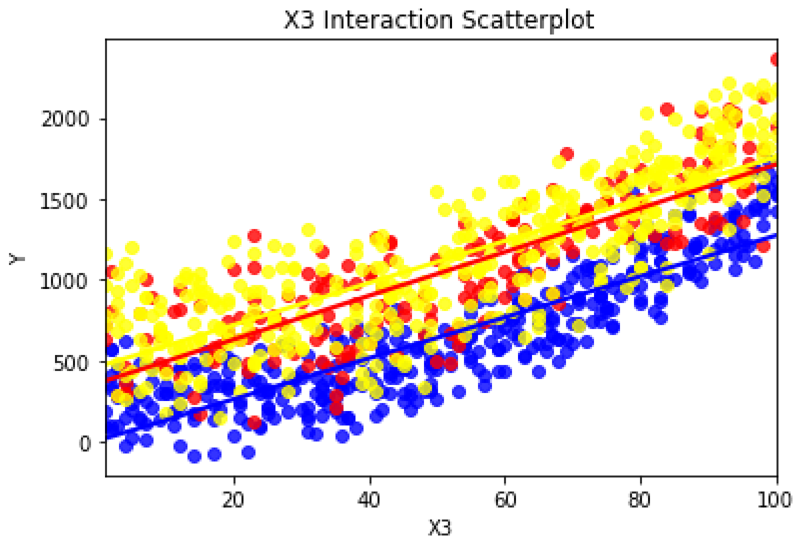
In [18]:
```
sns.regplot(x = "X1", y = "Y", ci = None, data = blue_observations, color='blue')
sns.regplot(x = "X1", y = "Y", ci = None, data = red_observations, color='red')
sns.regplot(x = "X1", y = "Y", ci = None, data = yellow_observations, color = 'yellow').set(title = 'X1 Interaction Scatterplot')
```

Out[18]:
```
[Text(0.5, 1.0, 'X1 Interaction Scatterplot')]
```



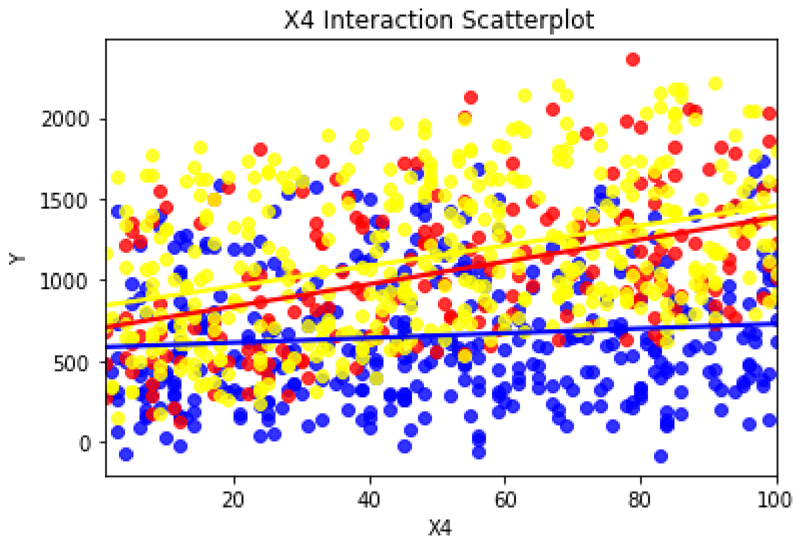
In [19]:
```
sns.regplot(x = "X3", y = "Y", ci = None, data = blue_observations, color='blue')
sns.regplot(x = "X3", y = "Y", ci = None, data = red_observations, color='red')
sns.regplot(x = "X3", y = "Y", ci = None, data = yellow_observations,color='yellow').set(title = 'X3 Interaction Scatterplot')
```

Out[19]:
```
[Text(0.5, 1.0, 'X3 Interaction Scatterplot')]
```



In [20]:
```
sns.regplot(x = "X4", y = "Y", ci = None, data = blue_observations, color = 'blue')
sns.regplot(x = "X4", y = "Y", ci = None, data = red_observations, color = 'red')
sns.regplot(x = "X4", y = "Y", ci = None, data = yellow_observations, color = 'yellow').set(title = 'X4 Interaction Scatterplot')
```

Out[20]:
```
[Text(0.5, 1.0, 'X4 Interaction Scatterplot')]
```



1p) Which predictor appears to have interactions with the color category?

X4

1q) Add an interaction effect to the model for this predictor, run the new regression, and display the coefficients and fit statistics

In [21]:
```
prob_1_dataset_4 = prob_1_dataset_3.copy()
prob_1_dataset_4['X4_Red_interaction'] = prob_1_dataset_4['X4']*prob_1_dataset_4['Red']
prob_1_dataset_4['X4_Blue_interaction'] = prob_1_dataset_4['X4']*prob_1_dataset_4['Blue']
X = prob_1_dataset_4.drop('Y',1)
X
```

Out[21]:

	X1	X3	X4	Blue	Red	X3_2	X4_Red_interaction	X4_Blue_interaction
0	11	18	3	0	0	324	0	0
1	19	11	93	0	1	121	93	0
2	48	36	31	0	1	1296	31	0
3	4	43	68	0	0	1849	0	0
4	82	37	65	1	0	1369	0	65
...	...	...	...	...	...	...	...	...
995	54	86	23	0	0	7396	0	0
996	15	77	100	0	1	5929	100	0
997	17	27	91	0	1	729	91	0
998	57	12	2	0	0	144	0	0
999	61	25	86	1	0	625	0	86

1000 rows × 8 columns

In [22]:
```
model_5 = LinearRegression(fit_intercept = True)
model_5.fit(X,y)
print("Intercept:", model_5.intercept_)
print("Coefficients:", model_5.coef_)
```

```
Intercept: 159.66263491435905
Coefficients: [ 4.11560664e+00 -1.87609237e+00  6.96730244e+00 -2.02419518e+02
 -6.30062776e+01  1.47004575e-01 -2.65455180e-01 -5.19612312e+00]
```

In [23]:
```
y_hat = model_5.predict(X)
print("R2: ", metrics.r2_score(y,y_hat))
print("MSE: ", metrics.mean_squared_error(y,y_hat))
print("MAE: ", metrics.mean_absolute_error(y,y_hat))
print("Max error: ", metrics.max_error(y,y_hat))
```

```
R2:  0.960104039499943
MSE:  9874.906700908248
MAE:  79.49538411126156
Max error:  451.06023439614233
```

1r) Using statsmodels, run the same regression and assess the p-values of the coefficients. Which interaction affects appear to be statistically significa

In [24]:
```
model_6 = sm.OLS(y, sm.add_constant(X))
model_6.fit().summary()
```

Out[24]:

OLS Regression Results

Dep. Variable:	Y	**R-squared:**	0.960
Model:	OLS	**Adj. R-squared:**	0.960
Method:	Least Squares	**F-statistic:**	2981.
Date:	Tue, 26 Jul 2022	**Prob (F-statistic):**	0.00
Time:	10:53:37	**Log-Likelihood:**	-6017.8
No. Observations:	1000	**AIC:**	1.205e+04
Df Residuals:	991	**BIC:**	1.210e+04
Df Model:	8		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	159.6626	14.894	10.720	0.000	130.435	188.891
X1	4.1156	0.110	37.400	0.000	3.900	4.332
X3	-1.8761	0.449	-4.178	0.000	-2.757	-0.995
X4	6.9673	0.173	40.188	0.000	6.627	7.308
Blue	-202.4195	14.156	-14.299	0.000	-230.199	-174.640
Red	-63.0063	17.426	-3.616	0.000	-97.202	-28.810

X3_2	0.1470	0.004	34.295	0.000	0.139	0.155
X4_Red_interaction	-0.2655	0.295	-0.900	0.368	-0.844	0.313
X4_Blue_interaction	-5.1961	0.247	-21.069	0.000	-5.680	-4.712

Omnibus:	2.115	**Durbin-Watson:**	1.985
Prob(Omnibus):	0.347	**Jarque-Bera (JB):**	2.130
Skew:	-0.030	**Prob(JB):**	0.345
Kurtosis:	3.218	**Cond. No.**	3.11e+04

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 3.11e+04. This might indicate that there are
strong multicollinearity or other numerical problems.

X4_Blue_Interacton appears to be statistically significant

```
In [ ]:
```