Module 3 Homework

ISE-529 Predictive Analytics

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 [96]:
```
import pandas;

df = pandas.read_csv(filepath_or_buffer = "HW Problem 1 Dataset.csv");
bd_df = pandas.get_dummies(df);
print(bd_df.head(3));
```

```
   X1  X2  X3  X4  X5           Y  X6_Blue  X6_Red  X6_Yellow
0  11  47  18   3  56  153.157223        0       0          1
1  19  91  11  93   1  809.384179        0       1          0
2  48  33  36  31  22  395.466944        0       1          0
```

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

In [97]:
```
import statsmodels.api as sm;

X = bd_df[["X1", "X2", "X3", "X4", "X5", "X6_Blue", "X6_Red", "X6_Yellow"]];
Y = bd_df["Y"];

X = sm.add_constant(X);

model = sm.OLS(Y, X).fit();
print(model.summary());
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      Y   R-squared:                       0.892
Model:                            OLS   Adj. R-squared:                  0.891
Method:                 Least Squares   F-statistic:                     1170.
Date:                Mon, 18 Jul 2022   Prob (F-statistic):               0.00
Time:                        06:27:30   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       -126.5616     16.115     -7.854      0.000    -158.185     -94.939
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
X6_Blue     -327.6865      8.734    -37.519      0.000    -344.826    -310.548
X6_Red        55.1448     10.647      5.180      0.000      34.252      76.037
X6_Yellow    145.9802      8.780     16.626      0.000     128.750     163.210
==============================================================================
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.                     1.11e+17
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 1.09e-27. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

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

X2 and X5 don't have statistically significant relationship to the response variable, because their p-values are higher than 0.05, which is considered an insignificant p-value.

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 [98]:
```
dp_bd_df = bd_df.drop(["X2", "X5"], axis = 1);
print(dp_bd_df.head(10));
```

```
   X1  X3  X4            Y  X6_Blue  X6_Red  X6_Yellow
0  11  18   3   153.157223        0       0          1
1  19  11  93   809.384179        0       1          0
2  48  36  31   395.466944        0       1          0
3   4  43  68   892.610788        0       0          1
4  82  37  65   476.573108        1       0          0
5  41   6  88   797.891711        0       0          1
6  29  83  12   871.984975        1       0          0
7  22  44  89   952.367041        0       0          1
8  12  39  67   343.993916        1       0          0
9   2  68  96  1297.651894        0       0          1
```

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

In [99]:

```python
X = dp_bd_df[["X1", "X3", "X4", "X6_Blue", "X6_Red", "X6_Yellow"]];
Y = dp_bd_df["Y"];

X = sm.add_constant(X);

sm_model = sm.OLS(Y, X).fit();
print(sm_model.summary());
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      Y   R-squared:                       0.892
Model:                            OLS   Adj. R-squared:                  0.891
Method:                 Least Squares   F-statistic:                     1641.
Date:                Mon, 18 Jul 2022   Prob (F-statistic):               0.00
Time:                        06:27:30   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       -124.9434     12.853     -9.721      0.000    -150.165     -99.722
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
X6_Blue     -327.1534      8.165    -40.068      0.000    -343.176    -311.131
X6_Red        55.6757     10.113      5.505      0.000      35.830      75.521
X6_Yellow    146.5344      8.108     18.073      0.000     130.624     162.445
==============================================================================
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.                     1.11e+18
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 6.82e-30. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

1f) Write the full regression equation

Y = 4.0403 *X1* + *13.0704* X3 + 4.8084 *X4* + *-327.1534* X6_Blue + 55.6757 *X6_Red* + *146.5344* X6_Yellow - 124.9434

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

Y = 4.0403 *X1* + *13.0704* X3 + 4.8084 *X4* + *146.5344* 1 - 124.9434

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

Y = 4.0403 *X1* + *13.0704* X3 + 4.8084 *X4* + *-327.1534* 1 - 124.9434

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

Y = 4.0403 *X1* + *13.0704* X3 + 4.8084 *X4* + *55.6757* 1 - 124.9434

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

In [100…

```python
from sklearn.linear_model import LinearRegression;

X = dp_bd_df[["X1", "X3", "X4", "X6_Blue", "X6_Red", "X6_Yellow"]];
Y = dp_bd_df["Y"];

sk_model = LinearRegression().fit(X, Y);
print(sk_model.coef_);
```

```
[   4.04032489   13.07044352    4.8083891  -285.50560589   97.32346179
  188.1821441 ]
```

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

In [101…

```python
from sklearn import metrics;
```

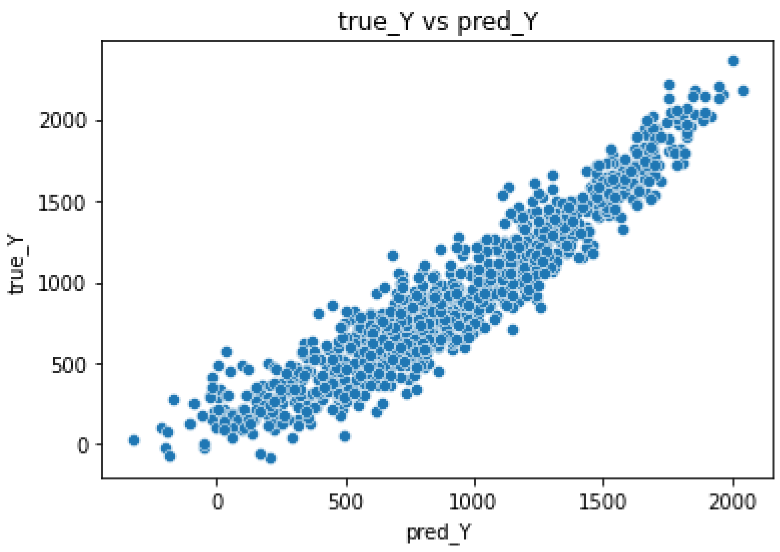
```
Y_pred_list = sk_model.predict(dp_bd_df[["X1", "X3", "X4", "X6_Blue", "X6_Red", "X6_Yellow"]]);

print("R^2: ", metrics.r2_score(dp_bd_df["Y"], Y_pred_list));
print("Mean Squared Error: ", metrics.mean_squared_error(dp_bd_df["Y"], Y_pred_list));
print("Mean Absolute Error: ", metrics.mean_absolute_error(dp_bd_df["Y"], Y_pred_list));
print("Max Error: ", metrics.max_error(dp_bd_df["Y"], Y_pred_list));
```

```
R^2:  0.8919740759220801
Mean Squared Error:  26738.19374639029
Mean Absolute Error:  130.8626856230258
Max Error:  540.8391996665016
```

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

In [102…

```
import seaborn;

temp_df = pandas.DataFrame(
    {
        "true_Y": dp_bd_df["Y"],
        "pred_Y": Y_pred_list,
        "X1": dp_bd_df["X1"],
        "X3": dp_bd_df["X3"],
        "X4": dp_bd_df["X4"]
    }
);
seaborn.scatterplot(data = temp_df, x = "pred_Y", y = "true_Y").set(title = "true_Y vs pred_Y");
```

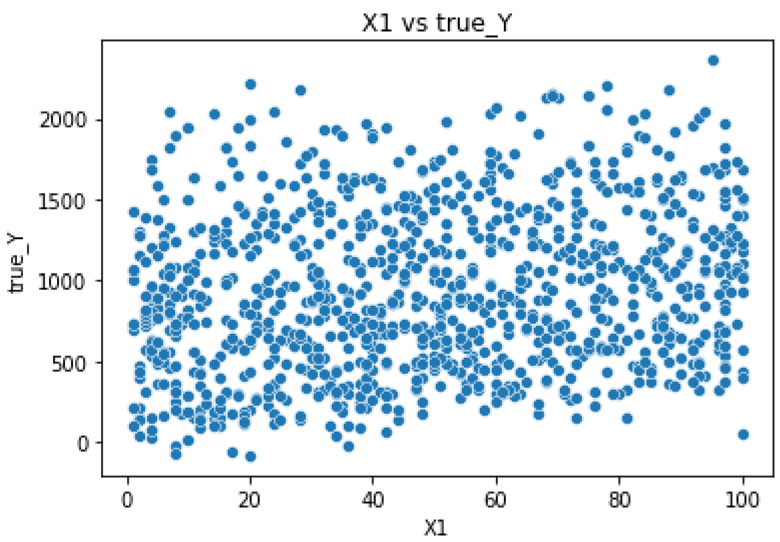


Investigate adding nonlinear terms

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

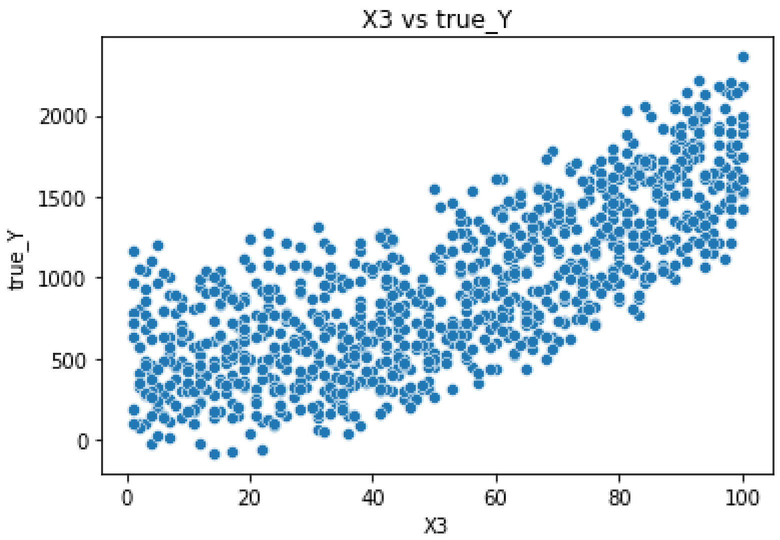
In [103…

```
seaborn.scatterplot(data = temp_df, x = "X1", y = "true_Y").set(title = "X1 vs true_Y");
```



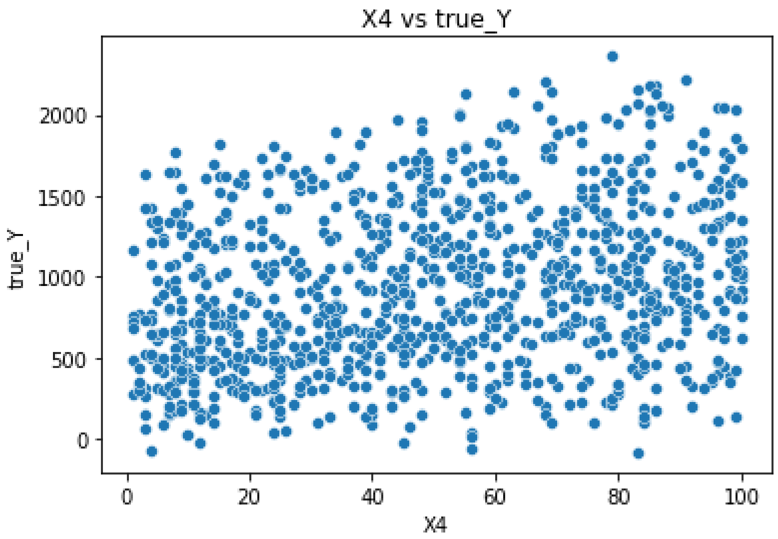
In [104…

```
seaborn.scatterplot(data = temp_df, x = "X3", y = "true_Y").set(title = "X3 vs true_Y");
```



In [105…

```
seaborn.scatterplot(data = temp_df, x = "X4", y = "true_Y").set(title = "X4 vs true_Y");
```



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

X1 and X4 have nonlinear relationships with the response variable.

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)

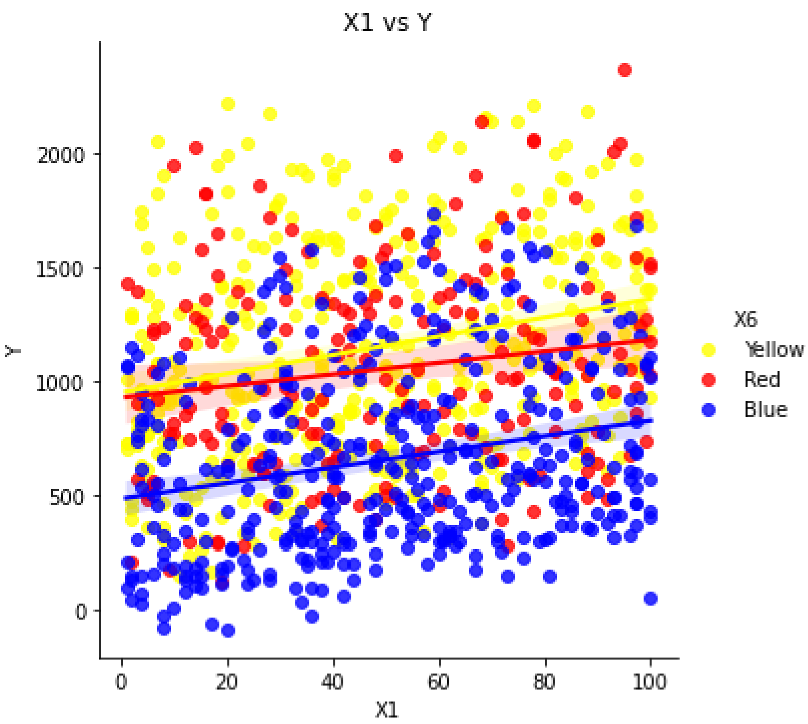
In [106…
```
sqrt_bd_df = dp_bd_df;
sqrt_bd_df["X1^2"] = sqrt_bd_df["X1"] ** 2;

X = sqrt_bd_df[["X1", "X3", "X4", "X6_Blue", "X6_Red", "X6_Yellow", "X1^2"]];
Y = sqrt_bd_df["Y"];

sk_model_sqrt = LinearRegression().fit(X, Y);
print(sk_model_sqrt.coef_);

Y_pred_list = sk_model_sqrt.predict(dp_bd_df[["X1", "X3", "X4", "X6_Blue", "X6_Red", "X6_Yellow", "X1^2"]]);

print("R^2: ", metrics.r2_score(dp_bd_df["Y"], Y_pred_list));
print("Mean Squared Error: ", metrics.mean_squared_error(dp_bd_df["Y"], Y_pred_list));
print("Mean Absolute Error: ", metrics.mean_absolute_error(dp_bd_df["Y"], Y_pred_list));
print("Max Error: ", metrics.max_error(dp_bd_df["Y"], Y_pred_list));
```

```
[ 4.77561843e+00  1.30667064e+01  4.80507629e+00 -2.85852064e+02
  9.74807249e+01  1.88371339e+02 -7.29697800e-03]
R^2:  0.8921007192743416
Mean Squared Error:  26706.847432823768
Mean Absolute Error:  130.5816887588017
Max Error:  552.7524568701187
```

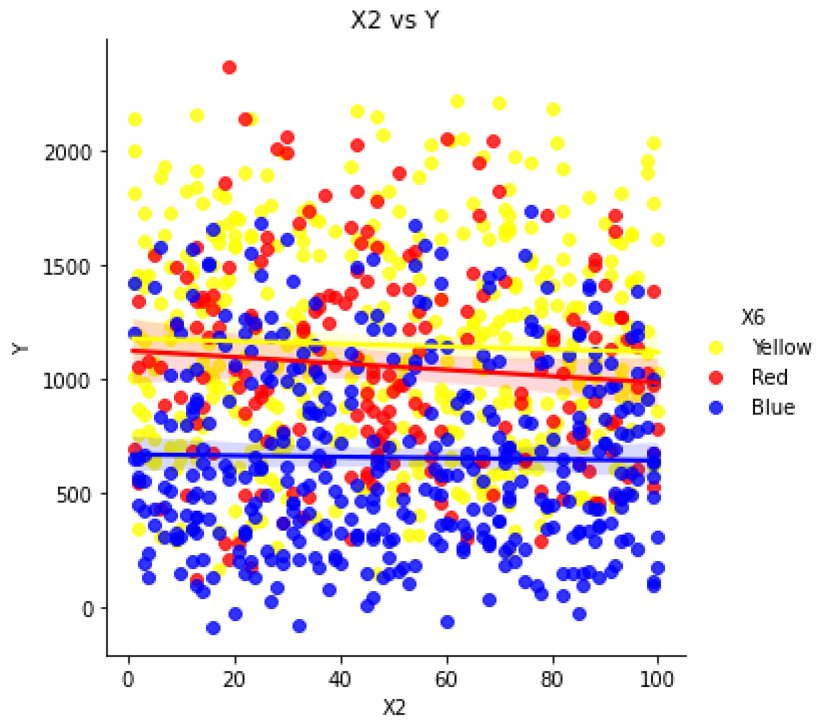
Investigate adding interaction effects

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

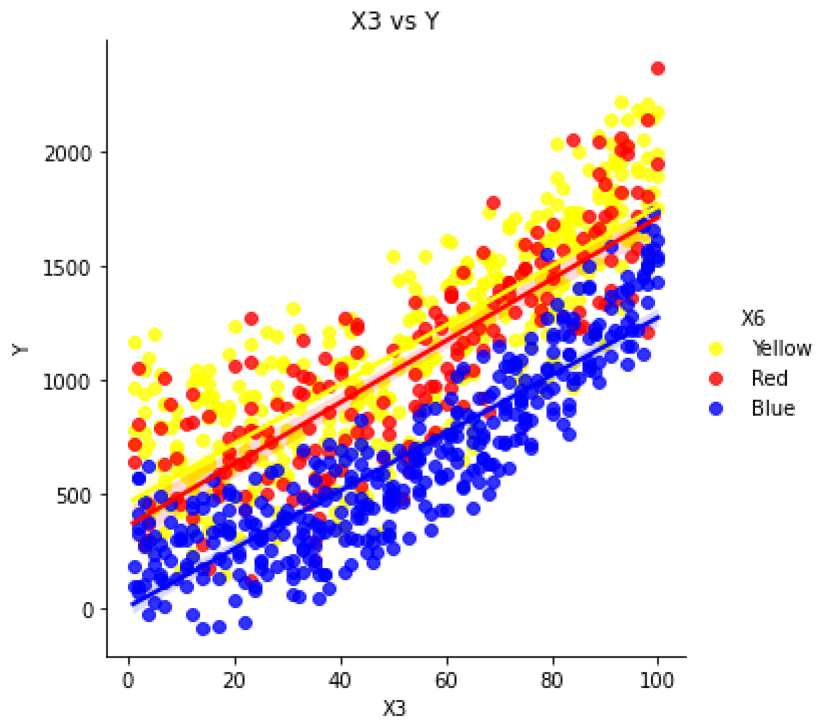
In [107…
```
color_dict = dict(
    {
        "Yellow": "yellow",
        "Blue": "blue",
        "Red": "red"
    }
);

seaborn.lmplot(data = df, x = "X1", y = "Y", hue = "X6", palette = color_dict).set(title = "X1 vs Y");
```



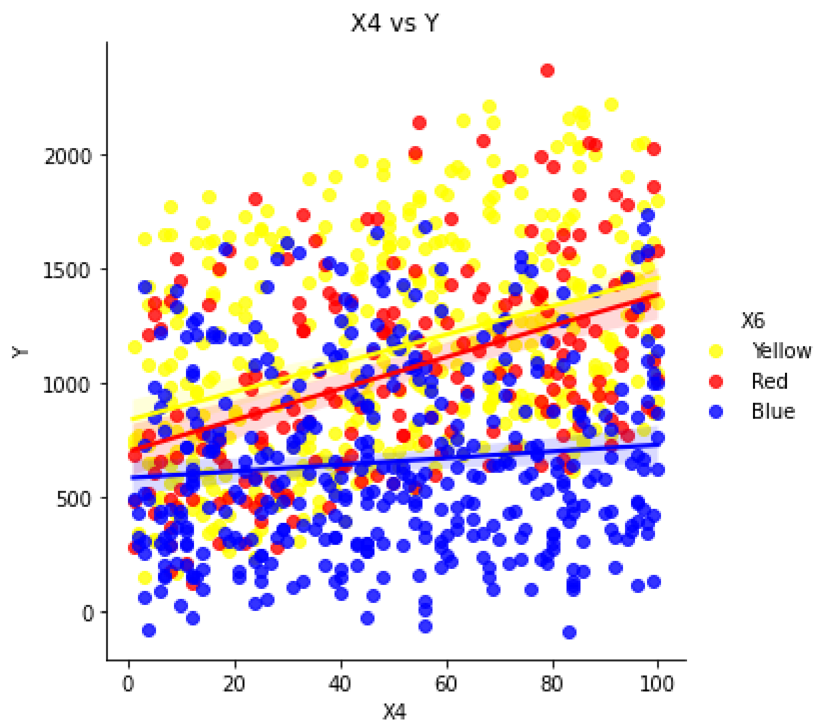
In [108…
```
seaborn.lmplot(data = df, x = "X2", y = "Y", hue = "X6", palette = color_dict).set(title = "X2 vs Y");
```



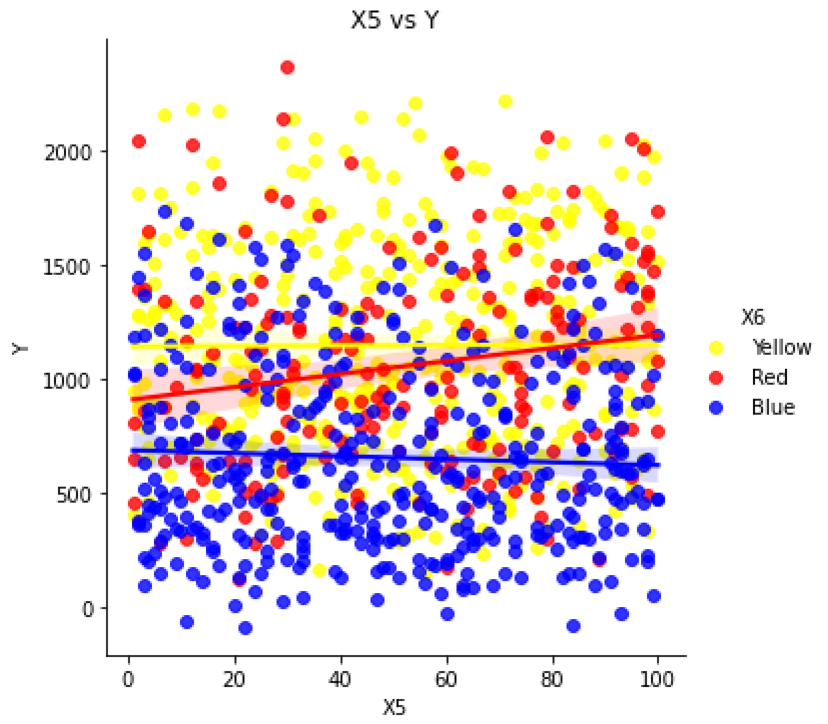
In [109…
```
seaborn.lmplot(data = df, x = "X3", y = "Y", hue = "X6", palette = color_dict).set(title = "X3 vs Y");
```



In [110…
```
seaborn.lmplot(data = df, x = "X4", y = "Y", hue = "X6", palette = color_dict).set(title = "X4 vs Y");
```



In [111…
```
seaborn.lmplot(data = df, x = "X5", y = "Y", hue = "X6", palette = color_dict).set(title = "X5 vs Y");
```



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

For X1, X2 and X5, the red line appears to have interactions with the color category. For X4, the blue line appears to interactions with the color category.

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

In [112…

```
ie_df = df;
x1_red_list = [];
x2_red_list = [];
x5_red_list = [];
x4_blue_list = [];

for idx, row in df.iterrows():
    if row["X6"] == "Blue":
        x4_blue_list.append(row["X4"] * 1);
    else:
        x4_blue_list.append(0);

    if row["X6"] == "Red":
        x1_red_list.append(row["X1"] * 1);
        x2_red_list.append(row["X2"] * 1);
        x5_red_list.append(row["X5"] * 1);
    else:
        x1_red_list.append(0);
        x2_red_list.append(0);
        x5_red_list.append(0);

ie_df["X1*isRed"] = x1_red_list;
ie_df["X2*isRed"] = x2_red_list;
ie_df["X5*isRed"] = x5_red_list;
ie_df["X4*isBlue"] = x4_blue_list;

X = ie_df[["X1", "X2", "X3", "X4", "X5", "X1*isRed", "X2*isRed", "X5*isRed", "X4*isBlue"]];
Y = ie_df["Y"];

ie_model = LinearRegression().fit(X, Y);
print(ie_model.coef_);

Y_pred_list = ie_model.predict(ie_df[["X1", "X2", "X3", "X4", "X5", "X1*isRed", "X2*isRed", "X5*isRed", "X4*isBlue"]]);

print("R^2: ", metrics.r2_score(ie_df["Y"], Y_pred_list));
print("Mean Squared Error: ", metrics.mean_squared_error(ie_df["Y"], Y_pred_list));
print("Mean Absolute Error: ", metrics.mean_absolute_error(ie_df["Y"], Y_pred_list));
print("Max Error: ", metrics.max_error(ie_df["Y"], Y_pred_list));
```

```
[ 4.24611575e+00 -3.17356201e-04  1.30611533e+01  8.02432317e+00
  9.61370293e-02 -2.95725489e-01 -5.62013614e-01 -3.19169834e-01
 -8.22879397e+00]
R^2:  0.9011482271049731
Mean Squared Error:  24467.4403704698
Mean Absolute Error:  125.54166583486094
Max Error:  605.6174064118887
```

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

In [113…

```
X = ie_df[["X1", "X2", "X3", "X4", "X5", "X1*isRed", "X2*isRed", "X5*isRed", "X4*isBlue"]];
Y = ie_df["Y"];

X = sm.add_constant(X);

model = sm.OLS(Y, X).fit();
print(model.summary());
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      Y   R-squared:                       0.901
Model:                            OLS   Adj. R-squared:                  0.900
Method:                 Least Squares   F-statistic:                     1003.
Date:                Mon, 18 Jul 2022   Prob (F-statistic):               0.00
Time:                        06:27:35   Log-Likelihood:                -6471.5
No. Observations:                1000   AIC:                         1.296e+04
Df Residuals:                     990   BIC:                         1.301e+04
Df Model:                           9
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       -186.8344     20.514     -9.108      0.000    -227.090    -146.579
X1             4.2461      0.190     22.400      0.000       3.874       4.618
X2            -0.0003      0.184     -0.002      0.999      -0.361       0.360
X3            13.0612      0.174     75.025      0.000      12.720      13.403
X4             8.0243      0.187     42.869      0.000       7.657       8.392
X5             0.0961      0.190      0.507      0.613      -0.276       0.469
X1*isRed      -0.2957      0.376     -0.786      0.432      -1.034       0.443
X2*isRed      -0.5620      0.354     -1.586      0.113      -1.257       0.133
X5*isRed      -0.3192      0.363     -0.879      0.379      -1.031       0.393
X4*isBlue     -8.2288      0.188    -43.759      0.000      -8.598      -7.860
==============================================================================
Omnibus:                        0.296   Durbin-Watson:                   1.949
Prob(Omnibus):                  0.862   Jarque-Bera (JB):                0.325
Skew:                           0.041   Prob(JB):                        0.850
Kurtosis:                       2.968   Cond. No.                         492.
==============================================================================

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

X4*isBlue appears to be statistically significant