# Midterm

## Problem 1

### This problem will involve linear regression on the dataset midterm data 1.csv. The response column is response and all other columns are features.

### (a) (5 points) Load the dataset. Remove any unnecessary columns. Remove any rows that have an NA value. Format the columns for feat.c and feat.g as categorical variables. Make pairwise plots showing the relations between all columns. Compute the pairwise correlations between all numerical columns. Split the dataset into a training set (75% of observations) and validation set (25% of observations).

In [7]:
```
#%% Step 1 - Import and Clean Data
import pandas as pd;
#import data
file_path = 'C:/Users/danma/Downloads/midterm_data_1.csv'
colnames=['response', 'a', 'b', 'c', 'd', 'e', 'f', 'g','h', 'i']
df = pd.read_table(file_path, sep=",",names=colnames)
df = df.iloc[1: , :]
df = df.astype(float)
df = df.reset_index(drop=True)
#remove row column as predictor
df = df.loc[:, df.columns != 'row']
#drops na values
df = df.dropna()
print("Original Data Frame (after cleaning):\n",df.head())
del file_path, colnames
#%% Step 2 - Format feat.c and feat.g as Categorical
from sklearn.preprocessing import OneHotEncoder;
oe = OneHotEncoder()
#encode C
encoded_C = oe.fit_transform(df[["c"]])
encoded_C = pd.DataFrame(encoded_C.toarray(),columns=["c_1","c_2","c_3","c_4"])
df = df.join(encoded_C,how='left')
#encode G
encoded_G = oe.fit_transform(df[["g"]])
encoded_G = pd.DataFrame(encoded_G.toarray(),columns=["g_1","g_2","g_3"])
df = df.join(encoded_G,how='left')
#drop original categorical columns
df = df.loc[:, df.columns != "c"]
df = df.loc[:, df.columns != "g"]
#drops na values
df = df.dropna()
print("\nFinal Data Frame (after encoding categorical columns):\n",df.head())
del encoded_C, encoded_G, oe
#%% Step 3 - Pairwise Plots and Numerical Correlations
import seaborn as sns
sns.set_theme(style="ticks")
```

```
sns.pairplot(df)

print("\nCorrelation Matrix (Numerical):\n",df.iloc[: , :8].corr())
#%% Step 4 - Split Data into Training and Testing
from sklearn.model_selection import train_test_split as TTS;

x = df.loc[:, df.columns != 'response']
y = df['response']
#turns y into a 1-d array instead of a dataframe column
y = y.to_numpy()
y = y.ravel()
train, test = TTS(df, test_size=0.25)
x_train = train.loc[:, df.columns != 'response']
y_train = train['response']
x_test = test.loc[:, df.columns != 'response']
y_test = test['response']
del x, y
```

```
Original Data Frame (after cleaning):
     response         a         b    c         d         e          f    g  \
0   1.658814 -0.879361 -2.297552  2.0 -2.052926 -1.458801   6.463630  2.0
1  10.691572  1.550930 -2.332102  4.0  1.110204 -1.744876 -11.834376  1.0
2  32.508862 -1.506886 -5.306166  1.0 -1.814569 -3.747318 -18.031607  1.0
3  37.747154  5.785842 -3.683903  2.0  6.453664 -3.645221 -10.534296  3.0
4  18.072661  1.988523 -3.895907  2.0  1.600221 -1.874998 -22.835629  1.0

          h         i
0 -0.438952 -1.307249
1  3.157077 -1.750373
2  0.065876 -3.795091
3 -0.293807 -3.495105
4  2.482003 -1.888469

Final Data Frame (after encoding categorical columns):
     response         a         b         d         e          f         h  \
0   1.658814 -0.879361 -2.297552 -2.052926 -1.458801   6.463630 -0.438952
1  10.691572  1.550930 -2.332102  1.110204 -1.744876 -11.834376  3.157077
2  32.508862 -1.506886 -5.306166 -1.814569 -3.747318 -18.031607  0.065876
3  37.747154  5.785842 -3.683903  6.453664 -3.645221 -10.534296 -0.293807
4  18.072661  1.988523 -3.895907  1.600221 -1.874998 -22.835629  2.482003

          i  c_1  c_2  c_3  c_4  g_1  g_2  g_3
0 -1.307249  0.0  1.0  0.0  0.0  0.0  1.0  0.0
1 -1.750373  0.0  0.0  0.0  1.0  1.0  0.0  0.0
2 -3.795091  1.0  0.0  0.0  0.0  1.0  0.0  0.0
3 -3.495105  0.0  1.0  0.0  0.0  0.0  0.0  1.0
4 -1.888469  0.0  1.0  0.0  0.0  1.0  0.0  0.0

Correlation Matrix (Numerical):
           response         a         b         d         e         f  \
response  1.000000  0.392181  0.027444  0.371863 -0.727407  0.046261
a         0.392181  1.000000  0.008759  0.976953 -0.026792 -0.028526
b         0.027444  0.008759  1.000000  0.018483 -0.014510 -0.031152
d         0.371863  0.976953  0.018483  1.000000 -0.014533 -0.055249
e        -0.727407 -0.026792 -0.014510 -0.014533  1.000000 -0.050704
f         0.046261 -0.028526 -0.031152 -0.055249 -0.050704  1.000000
h         0.210419  0.000467 -0.018514  0.001035  0.042234  0.003176
i        -0.726199 -0.026090 -0.013379 -0.013986  0.998763 -0.051840

                h         i
response  0.210419 -0.726199
a         0.000467 -0.026090
b        -0.018514 -0.013379
d         0.001035 -0.013986
e         0.042234  0.998763
f         0.003176 -0.051840
h         1.000000  0.044195
i         0.044195  1.000000
```

## (b) (5 points) Make a linear model using all features. How can you interpret the coefficients of feat.c? What does R2 signify? How can you interpret the value of the residual standard error? What does the F-statistic say about your model? Make a residual plot of the residuals vs. fitted values and comment on what this says about validity of the linearity and constant-variance error assumptions of the model.

In [8]:
```
#%% Step 5 - Create Linear Model and Show Summary Screen
```

```
import numpy as np;
import statsmodels.api as sm;

model = sm.OLS(y_train,x_train).fit()
print(model.summary())
print("\nResidual Standard Error:")
print(np.sqrt(model.mse_resid))

import matplotlib.pyplot as plt;
plt.style.use('seaborn') # pretty matplotlib plots
plt.rc('font', size=14)
plt.rc('figure', titlesize=18)
plt.rc('axes', labelsize=15)
plt.rc('axes', titlesize=18)
#%% Step 6 - Plot Residuals vs Fitted
#Residuals vs Fitted Plot
plt.figure()
sns.residplot(x=model.fittedvalues, y=y_train,
                          lowess=True,
                          scatter_kws={'alpha': 0.5},
                          line_kws={'color': 'red', 'lw': 1, 'alpha': 0.8})

plt.title('Residuals vs Fitted')
plt.xlabel('Fitted values')
plt.ylabel('Residuals')
plt.show()
```

OLS Regression Results

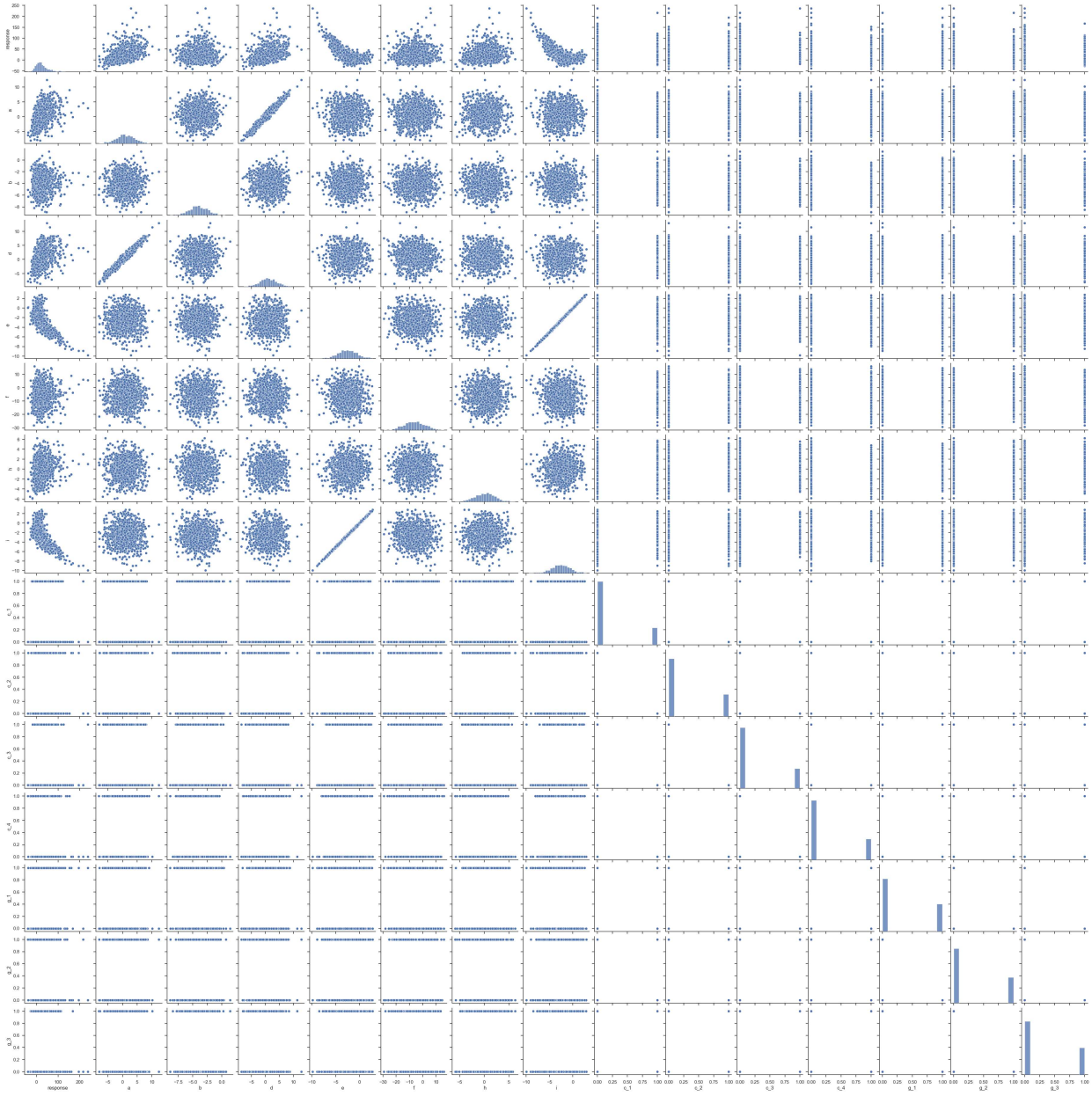
| | | | |
|---|---|---|---|
| Dep. Variable: | response | R-squared: | 0.726 |
| Model: | OLS | Adj. R-squared: | 0.721 |
| Method: | Least Squares | F-statistic: | 161.3 |
| Date: | Wed, 26 Oct 2022 | Prob (F-statistic): | 4.65e-196 |
| Time: | 20:11:04 | Log-Likelihood: | -3136.4 |
| No. Observations: | 744 | AIC: | 6299. |
| Df Residuals: | 731 | BIC: | 6359. |
| Df Model: | 12 | | |
| Covariance Type: | nonrobust | | |

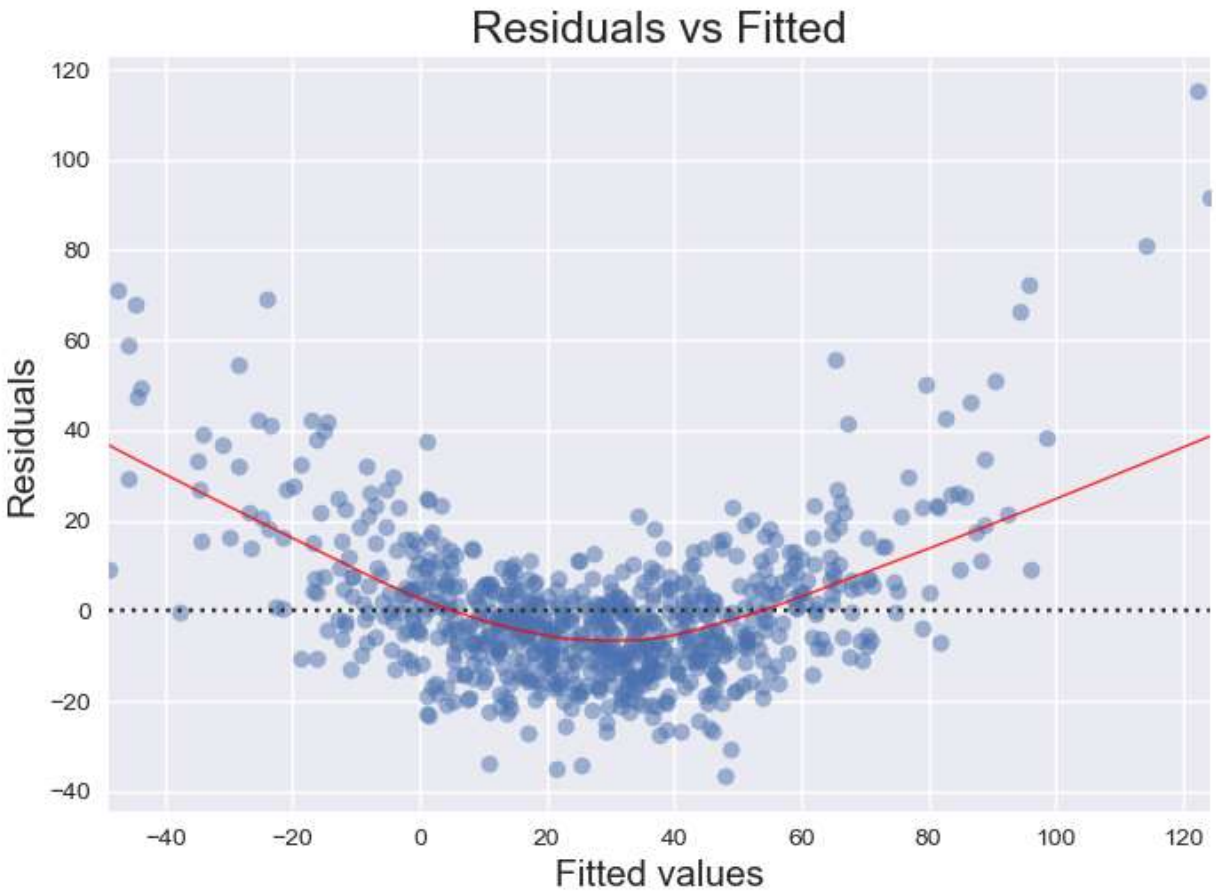
| | coef | std err | t | P>\|t\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| a | 4.6743 | 0.912 | 5.123 | 0.000 | 2.883 | 6.466 |
| b | 0.8840 | 0.398 | 2.218 | 0.027 | 0.102 | 1.666 |
| d | -0.8569 | 0.893 | -0.960 | 0.338 | -2.610 | 0.896 |
| e | -9.8762 | 6.223 | -1.587 | 0.113 | -22.093 | 2.340 |
| f | 0.0495 | 0.078 | 0.636 | 0.525 | -0.103 | 0.202 |
| h | 3.8463 | 0.302 | 12.728 | 0.000 | 3.253 | 4.440 |
| i | -1.3471 | 6.205 | -0.217 | 0.828 | -13.530 | 10.835 |
| c_1 | -0.6752 | 1.423 | -0.474 | 0.635 | -3.470 | 2.119 |
| c_2 | -0.6567 | 1.355 | -0.485 | 0.628 | -3.316 | 2.003 |
| c_3 | -1.0602 | 1.368 | -0.775 | 0.439 | -3.746 | 1.626 |
| c_4 | -2.2343 | 1.336 | -1.673 | 0.095 | -4.857 | 0.388 |
| g_1 | -0.2116 | 1.436 | -0.147 | 0.883 | -3.030 | 2.607 |
| g_2 | -1.6412 | 1.451 | -1.131 | 0.259 | -4.491 | 1.208 |
| g_3 | -2.7735 | 1.437 | -1.930 | 0.054 | -5.594 | 0.047 |

| | | | |
|---|---|---|---|
| Omnibus: | 326.485 | Durbin-Watson: | 2.060 |
| Prob(Omnibus): | 0.000 | Jarque-Bera (JB): | 2008.402 |
| Skew: | 1.885 | Prob(JB): | 0.00 |
| Kurtosis: | 10.112 | Cond. No. | 1.23e+17 |

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 5.66e-30. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.

Residual Standard Error:
16.533985910561245


response
a
b
d
e
f
h
i
c_1
c_2
c_3
c_4
g_1
g_2
g_3




## Coefficients of feat.c

We can interpret the coefficients of feat.c by using the p-values to determine importance. We can observe that feat.c 1-3 are large p-values resulting in low importance in predicting the depdendent variable. But, we can also see that the 4th value for feat.c is potentially important depending on the alpha chosen.

## R2

With the R-squared value equaling 0.716 we can say that 71.6% of the variance can be explained by the regression model.

## Residual Standard Error

The residual standard error shows us how well the model is fit to the data, but unless we are able to compare it to another RSE we do not have many assumptions to be made on the accuracy of this model.

## F-Statistic

Our F-Statistic and it's p-value shows us if the model is explaining anything. By using it in conjunction with a hypothesis test we can see that with our small F-Statistic p-value at least one slope from the betas, dependent variables, does explain for some of the response, Y.

# Residual Plot

We can see from the residual plot that the model is not linear in nature and more reflects that of a quadratic line meaning we would benefit from a quadratic term in th model. As for constant-variance seeing as the spread of the residuals remains relatively constant we can say there is constant variance.

### (c) (5 points) Make a linear model that includes all interactions between features and all quadratic terms for numerical features. From this model, identify a reduced set of coefficients that are the most relevant predictors. Look at the residual plot of your reduced model and comment on any observed differences between this plot and the residual plot from part b).

In [9]:
```
#%% Step 7 - Make Linear Model with Interaction
import statsmodels.formula.api as smf
#preliminary terms for model
inter = 'response ~ a + b + c_1 + c_2 + c_3 + c_4 + d + e + f + g_1 + g_2 + g_3 + h +
#interaction terms with a
a_inter = '+ a:b + a:c_1 + a:c_2 + a:c_3 + a:c_4 + a:d + a:e + a:f + a:g_1 + a:g_2 + a
#interaction terms with b
b_inter = '+ b:c_1 + b:c_2 + b:c_3 + b:c_4 + b:d + b:e + b:f + b:g_1 + b:g_2 + b:g_3 +
#interaction terms with c
c_1_inter = '+ c_1:c_2 + c_1:c_3 + c_1:c_4 + c_1:d + c_1:e + c_1:f + c_1:g_1 + c_1:g_2
c_2_inter = '+ c_2:c_3 + c_2:c_4 + c_2:d + c_2:e + c_2:f + c_2:g_1 + c_2:g_2 + c_2:g_3
c_3_inter = '+ c_3:c_4 + c_3:d + c_3:e + c_3:f + c_3:g_1 + c_3:g_2 + c_3:g_3 + c_3:h +
c_4_inter = '+ c_4:d + c_4:e + c_4:f + c_4:g_1 + c_4:g_2 + c_4:g_3 + c_4:h + c_4:i'
#interaction terms with d
d_inter = '+ d:e + d:f + d:g_1 + d:g_2 + d:g_3 + d:h + d:i'
#interaction terms with e
e_inter = '+ e:f + e:g_1 + e:g_2 + e:g_3 + e:h + e:i'
#interaction terms with f
f_inter = '+ f:g_1 + f:g_2 + f:g_3 + f:h + f:i'
#interaction terms with g
g_1_inter = '+ g_1:g_2 + g_1:g_3 + g_1:h + g_1:i'
g_2_inter = '+ g_2:g_3 + g_2:h + g_2:i'
g_3_inter = '+ g_3:h + g_3:i'
#interaction terms with h
h_inter = '+ h:i'
#create quadratic terms for numerical (non-categorical)
quadterms = '+ np.square(a) + np.square(b) + np.square(d) + np.square(e) + np.square(f
fullterms  = inter+a_inter+b_inter+c_1_inter+c_2_inter+c_3_inter+c_4_inter+d_inter+e_i
model2 =  smf.ols(fullterms, data=train).fit()
print(model2.summary())
print("\nResidual Standard Error:")
print(np.sqrt(model2.mse_resid))

#%% Step 8 - create reduced model
print("\n\nReduced Model:\n")
selectedterms = 'response ~ b + c_1 + c_2 + c_3 + c_4 + d + e + f + g_1 + g_2 + g_3 +
reducedterms = selectedterms+b_inter+c_2_inter+c_3_inter+c_4_inter+d_inter+e_inter+f_i
```

```
reducedmodel = smf.ols(reducedterms, data=train).fit()
print(reducedmodel.summary())
print("\nResidual Standard Error:")
print(np.sqrt(reducedmodel.mse_resid))

# Step 9 - create residual plot from reduced model
#Residuals vs Fitted Plot
plt.figure()
sns.residplot(x=reducedmodel.fittedvalues, y=y_train,
                          lowess=True,
                          scatter_kws={'alpha': 0.5},
                          line_kws={'color': 'red', 'lw': 1, 'alpha': 0.8})

plt.title('Residuals vs Fitted')
plt.xlabel('Fitted values')
plt.ylabel('Residuals')
plt.show()

del fullterms, inter, a_inter, b_inter, c_1_inter, c_2_inter, c_3_inter, c_4_inter, d_
```

OLS Regression Results

| | | | |
|---|---|---|---|
| Dep. Variable: | response | R-squared: | 0.934 |
| Model: | OLS | Adj. R-squared: | 0.926 |
| Method: | Least Squares | F-statistic: | 115.3 |
| Date: | Wed, 26 Oct 2022 | Prob (F-statistic): | 0.00 |
| Time: | 20:11:12 | Log-Likelihood: | -2607.7 |
| No. Observations: | 744 | AIC: | 5379. |
| Df Residuals: | 662 | BIC: | 5758. |
| Df Model: | 81 | | |
| Covariance Type: | nonrobust | | |

| | coef | std err | t | P>\|t\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| Intercept | 2.6066 | 1.433 | 1.819 | 0.069 | -0.207 | 5.420 |
| a | 3.2134 | 1.053 | 3.053 | 0.002 | 1.146 | 5.280 |
| b | 0.4553 | 0.588 | 0.774 | 0.439 | -0.700 | 1.611 |
| c_1 | 2.1490 | 1.561 | 1.377 | 0.169 | -0.916 | 5.214 |
| c_2 | 0.5614 | 1.498 | 0.375 | 0.708 | -2.381 | 3.504 |
| c_3 | 1.0809 | 1.491 | 0.725 | 0.469 | -1.847 | 4.009 |
| c_4 | -1.1847 | 1.401 | -0.846 | 0.398 | -3.935 | 1.566 |
| d | -1.3096 | 1.038 | -1.261 | 0.208 | -3.348 | 0.729 |
| e | -10.1503 | 7.368 | -1.378 | 0.169 | -24.618 | 4.318 |
| f | 0.0316 | 0.090 | 0.351 | 0.725 | -0.145 | 0.209 |
| g_1 | 2.6627 | 1.262 | 2.111 | 0.035 | 0.186 | 5.140 |
| g_2 | -0.1142 | 1.411 | -0.081 | 0.936 | -2.885 | 2.657 |
| g_3 | 0.0581 | 1.354 | 0.043 | 0.966 | -2.601 | 2.717 |
| h | 0.2094 | 0.358 | 0.585 | 0.558 | -0.493 | 0.912 |
| i | 10.7418 | 7.282 | 1.475 | 0.141 | -3.556 | 25.040 |
| a:b | 0.0924 | 0.332 | 0.278 | 0.781 | -0.559 | 0.744 |
| a:c_1 | 1.8721 | 1.003 | 1.867 | 0.062 | -0.097 | 3.841 |
| a:c_2 | 0.5644 | 0.895 | 0.631 | 0.528 | -1.193 | 2.321 |
| a:c_3 | 0.8660 | 0.873 | 0.992 | 0.321 | -0.847 | 2.580 |
| a:c_4 | -0.0891 | 0.856 | -0.104 | 0.917 | -1.770 | 1.592 |
| a:d | -0.0401 | 1.045 | -0.038 | 0.969 | -2.092 | 2.012 |
| a:e | 2.5390 | 5.447 | 0.466 | 0.641 | -8.157 | 13.235 |
| a:f | -0.0763 | 0.063 | -1.212 | 0.226 | -0.200 | 0.047 |
| a:g_1 | 1.1159 | 0.820 | 1.361 | 0.174 | -0.494 | 2.726 |
| a:g_2 | 0.4649 | 0.778 | 0.598 | 0.550 | -1.062 | 1.992 |
| a:g_3 | 1.6326 | 0.787 | 2.073 | 0.039 | 0.087 | 3.179 |
| a:h | 0.1844 | 0.254 | 0.725 | 0.469 | -0.315 | 0.684 |
| a:i | -2.1158 | 5.448 | -0.388 | 0.698 | -12.813 | 8.581 |
| b:c_1 | -0.1978 | 0.414 | -0.478 | 0.633 | -1.011 | 0.615 |
| b:c_2 | 0.5006 | 0.418 | 1.197 | 0.232 | -0.320 | 1.321 |
| b:c_3 | 0.2549 | 0.409 | 0.623 | 0.534 | -0.549 | 1.059 |
| b:c_4 | -0.1024 | 0.399 | -0.257 | 0.798 | -0.886 | 0.682 |
| b:d | -0.1398 | 0.327 | -0.427 | 0.669 | -0.782 | 0.503 |
| b:e | -3.9086 | 2.352 | -1.662 | 0.097 | -8.527 | 0.710 |
| b:f | -0.0023 | 0.028 | -0.080 | 0.936 | -0.058 | 0.053 |
| b:g_1 | 0.5195 | 0.345 | 1.506 | 0.133 | -0.158 | 1.197 |
| b:g_2 | -0.0067 | 0.392 | -0.017 | 0.986 | -0.776 | 0.763 |
| b:g_3 | -0.0574 | 0.363 | -0.158 | 0.874 | -0.770 | 0.655 |
| b:h | -0.1569 | 0.109 | -1.443 | 0.149 | -0.370 | 0.057 |
| b:i | 3.8354 | 2.339 | 1.640 | 0.101 | -0.757 | 8.427 |
| c_1:c_2 | -1.384e-13 | 1.59e-13 | -0.871 | 0.384 | -4.51e-13 | 1.74e-13 |
| c_1:c_3 | 7.918e-14 | 2.74e-13 | 0.289 | 0.773 | -4.6e-13 | 6.18e-13 |
| c_1:c_4 | -8.011e-14 | 5.35e-14 | -1.497 | 0.135 | -1.85e-13 | 2.5e-14 |
| c_1:d | -1.8555 | 0.965 | -1.923 | 0.055 | -3.750 | 0.039 |
| c_1:e | -9.1479 | 6.513 | -1.405 | 0.161 | -21.936 | 3.640 |
| c_1:f | 0.0582 | 0.081 | 0.716 | 0.474 | -0.101 | 0.218 |

| | | | | | | |
|---|---|---|---|---|---|---|
| c_1:g_1 | 1.9782 | 1.076 | 1.838 | 0.066 | -0.135 | 4.091 |
| c_1:g_2 | -0.2775 | 1.045 | -0.266 | 0.791 | -2.329 | 1.774 |
| c_1:g_3 | 0.4483 | 1.041 | 0.431 | 0.667 | -1.595 | 2.492 |
| c_1:h | 0.6861 | 0.297 | 2.310 | 0.021 | 0.103 | 1.269 |
| c_1:i | 9.7658 | 6.451 | 1.514 | 0.131 | -2.900 | 22.432 |
| c_2:c_3 | -6.142e-14 | 7.4e-14 | -0.830 | 0.407 | -2.07e-13 | 8.39e-14 |
| c_2:c_4 | -8.76e-14 | 1.82e-13 | -0.480 | 0.631 | -4.46e-13 | 2.71e-13 |
| c_2:d | 0.2188 | 0.880 | 0.249 | 0.804 | -1.508 | 1.946 |
| c_2:e | -5.4019 | 6.345 | -0.851 | 0.395 | -17.860 | 7.056 |
| c_2:f | 0.0445 | 0.076 | 0.586 | 0.558 | -0.105 | 0.194 |
| c_2:g_1 | 0.4724 | 0.958 | 0.493 | 0.622 | -1.409 | 2.354 |
| c_2:g_2 | -0.8143 | 1.003 | -0.812 | 0.417 | -2.784 | 1.156 |
| c_2:g_3 | 0.9033 | 1.045 | 0.864 | 0.388 | -1.149 | 2.956 |
| c_2:h | -0.0863 | 0.301 | -0.287 | 0.774 | -0.678 | 0.505 |
| c_2:i | 5.1486 | 6.309 | 0.816 | 0.415 | -7.240 | 17.537 |
| c_3:c_4 | -4.833e-14 | 1.01e-13 | -0.477 | 0.633 | -2.47e-13 | 1.51e-13 |
| c_3:d | -0.2362 | 0.881 | -0.268 | 0.789 | -1.966 | 1.493 |
| c_3:e | 8.2513 | 6.324 | 1.305 | 0.192 | -4.167 | 20.670 |
| c_3:f | 0.0071 | 0.078 | 0.091 | 0.928 | -0.147 | 0.161 |
| c_3:g_1 | -0.3898 | 1.026 | -0.380 | 0.704 | -2.405 | 1.626 |
| c_3:g_2 | 1.3798 | 0.989 | 1.396 | 0.163 | -0.561 | 3.321 |
| c_3:g_3 | 0.0908 | 1.027 | 0.088 | 0.930 | -1.926 | 2.108 |
| c_3:h | -0.2989 | 0.317 | -0.944 | 0.346 | -0.921 | 0.323 |
| c_3:i | -8.1106 | 6.317 | -1.284 | 0.200 | -20.514 | 4.292 |
| c_4:d | 0.5633 | 0.841 | 0.670 | 0.503 | -1.088 | 2.215 |
| c_4:e | -3.8518 | 5.911 | -0.652 | 0.515 | -15.458 | 7.755 |
| c_4:f | -0.0782 | 0.078 | -0.998 | 0.319 | -0.232 | 0.076 |
| c_4:g_1 | 0.6019 | 0.924 | 0.651 | 0.515 | -1.213 | 2.417 |
| c_4:g_2 | -0.4023 | 1.029 | -0.391 | 0.696 | -2.423 | 1.618 |
| c_4:g_3 | -1.3843 | 0.924 | -1.498 | 0.135 | -3.199 | 0.431 |
| c_4:h | -0.0915 | 0.302 | -0.303 | 0.762 | -0.685 | 0.502 |
| c_4:i | 3.9380 | 5.904 | 0.667 | 0.505 | -7.654 | 15.530 |
| d:e | -1.8108 | 5.392 | -0.336 | 0.737 | -12.399 | 8.777 |
| d:f | 0.0468 | 0.062 | 0.761 | 0.447 | -0.074 | 0.168 |
| d:g_1 | -0.5308 | 0.799 | -0.664 | 0.507 | -2.100 | 1.038 |
| d:g_2 | -0.0392 | 0.767 | -0.051 | 0.959 | -1.546 | 1.467 |
| d:g_3 | -0.7397 | 0.774 | -0.955 | 0.340 | -2.260 | 0.781 |
| d:h | -0.1881 | 0.249 | -0.755 | 0.450 | -0.677 | 0.301 |
| d:i | 1.2996 | 5.392 | 0.241 | 0.810 | -9.287 | 11.887 |
| e:f | -0.0908 | 0.445 | -0.204 | 0.838 | -0.964 | 0.783 |
| e:g_1 | -3.3979 | 5.290 | -0.642 | 0.521 | -13.784 | 6.988 |
| e:g_2 | -5.9672 | 5.225 | -1.142 | 0.254 | -16.227 | 4.292 |
| e:g_3 | -0.7852 | 5.886 | -0.133 | 0.894 | -12.342 | 10.772 |
| e:h | -1.9747 | 1.748 | -1.129 | 0.259 | -5.408 | 1.459 |
| e:i | 26.4133 | 53.437 | 0.494 | 0.621 | -78.514 | 131.340 |
| f:g_1 | 0.0615 | 0.069 | 0.895 | 0.371 | -0.073 | 0.196 |
| f:g_2 | -0.0166 | 0.071 | -0.234 | 0.815 | -0.156 | 0.123 |
| f:g_3 | -0.0132 | 0.066 | -0.199 | 0.842 | -0.143 | 0.117 |
| f:h | -0.0358 | 0.022 | -1.646 | 0.100 | -0.079 | 0.007 |
| f:i | 0.1023 | 0.444 | 0.230 | 0.818 | -0.770 | 0.974 |
| g_1:g_2 | 0 | 0 | nan | nan | 0 | 0 |
| g_1:g_3 | 0 | 0 | nan | nan | 0 | 0 |
| g_1:h | 0.2967 | 0.259 | 1.146 | 0.252 | -0.212 | 0.805 |
| g_1:i | 3.7641 | 5.263 | 0.715 | 0.475 | -6.571 | 14.099 |
| g_2:g_3 | 0 | 0 | nan | nan | 0 | 0 |
| g_2:h | 0.0229 | 0.281 | 0.082 | 0.935 | -0.529 | 0.575 |
| g_2:i | 5.9841 | 5.190 | 1.153 | 0.249 | -4.207 | 16.175 |
| g_3:h | -0.1102 | 0.261 | -0.422 | 0.673 | -0.623 | 0.403 |
| g_3:i | 0.9936 | 5.872 | 0.169 | 0.866 | -10.536 | 12.523 |
| h:i | 1.0189 | 1.743 | 0.585 | 0.559 | -2.404 | 4.442 |

| | | | | | | |
|---|---|---|---|---|---|---|
| np.square(a) | 0.0783 | 0.539 | 0.145 | 0.884 | -0.979 | 1.136 |
| np.square(b) | 0.0556 | 0.103 | 0.542 | 0.588 | -0.146 | 0.257 |
| np.square(d) | -0.0176 | 0.521 | -0.034 | 0.973 | -1.042 | 1.006 |
| np.square(e) | -10.4375 | 26.813 | -0.389 | 0.697 | -63.087 | 42.212 |
| np.square(f) | -0.0023 | 0.004 | -0.555 | 0.579 | -0.011 | 0.006 |
| np.square(h) | -0.0684 | 0.060 | -1.147 | 0.252 | -0.186 | 0.049 |
| np.square(i) | -13.7019 | 26.652 | -0.514 | 0.607 | -66.035 | 38.631 |

| | | | |
|---|---|---|---|
| Omnibus: | 0.780 | Durbin-Watson: | 1.997 |
| Prob(Omnibus): | 0.677 | Jarque-Bera (JB): | 0.628 |
| Skew: | -0.003 | Prob(JB): | 0.730 |
| Kurtosis: | 3.142 | Cond. No. | 2.14e+16 |

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 4.71e-26. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.

Residual Standard Error:
8.537622184301524

Reduced Model:

OLS Regression Results

| | | | |
|---|---|---|---|
| Dep. Variable: | response | R-squared: | 0.927 |
| Model: | OLS | Adj. R-squared: | 0.919 |
| Method: | Least Squares | F-statistic: | 123.3 |
| Date: | Wed, 26 Oct 2022 | Prob (F-statistic): | 0.00 |
| Time: | 20:11:12 | Log-Likelihood: | -2646.2 |
| No. Observations: | 744 | AIC: | 5432. |
| Df Residuals: | 674 | BIC: | 5755. |
| Df Model: | 69 | | |
| Covariance Type: | nonrobust | | |

| | coef | std err | t | P>\|t\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| Intercept | 2.1228 | 1.467 | 1.447 | 0.148 | -0.758 | 5.004 |
| b | 0.2561 | 0.608 | 0.421 | 0.674 | -0.937 | 1.449 |
| c_1 | 1.4865 | 1.621 | 0.917 | 0.360 | -1.697 | 4.670 |
| c_2 | 0.4331 | 1.551 | 0.279 | 0.780 | -2.613 | 3.479 |
| c_3 | 2.0046 | 1.533 | 1.307 | 0.192 | -1.006 | 5.015 |
| c_4 | -1.8013 | 1.431 | -1.259 | 0.209 | -4.611 | 1.009 |
| d | 1.7332 | 0.314 | 5.513 | 0.000 | 1.116 | 2.350 |
| e | -12.5539 | 7.628 | -1.646 | 0.100 | -27.532 | 2.424 |
| f | 0.0343 | 0.094 | 0.367 | 0.714 | -0.149 | 0.218 |
| g_1 | 2.1959 | 1.307 | 1.680 | 0.093 | -0.371 | 4.762 |
| g_2 | -0.2749 | 1.461 | -0.188 | 0.851 | -3.144 | 2.594 |
| g_3 | 0.2018 | 1.392 | 0.145 | 0.885 | -2.532 | 2.935 |
| h | 0.6291 | 0.489 | 1.287 | 0.199 | -0.331 | 1.589 |
| i | 13.0747 | 7.540 | 1.734 | 0.083 | -1.731 | 27.880 |
| c_1:c_2 | 3.876e-13 | 8.46e-13 | 0.458 | 0.647 | -1.27e-12 | 2.05e-12 |
| c_1:c_3 | -3.259e-13 | 8.56e-13 | -0.381 | 0.704 | -2.01e-12 | 1.36e-12 |
| c_1:c_4 | 7.406e-14 | 8.55e-13 | 0.087 | 0.931 | -1.61e-12 | 1.75e-12 |
| c_1:e | -8.4886 | 6.754 | -1.257 | 0.209 | -21.750 | 4.773 |
| c_1:f | 0.0745 | 0.084 | 0.885 | 0.376 | -0.091 | 0.240 |
| c_1:g_1 | 1.7509 | 1.111 | 1.576 | 0.116 | -0.431 | 3.933 |

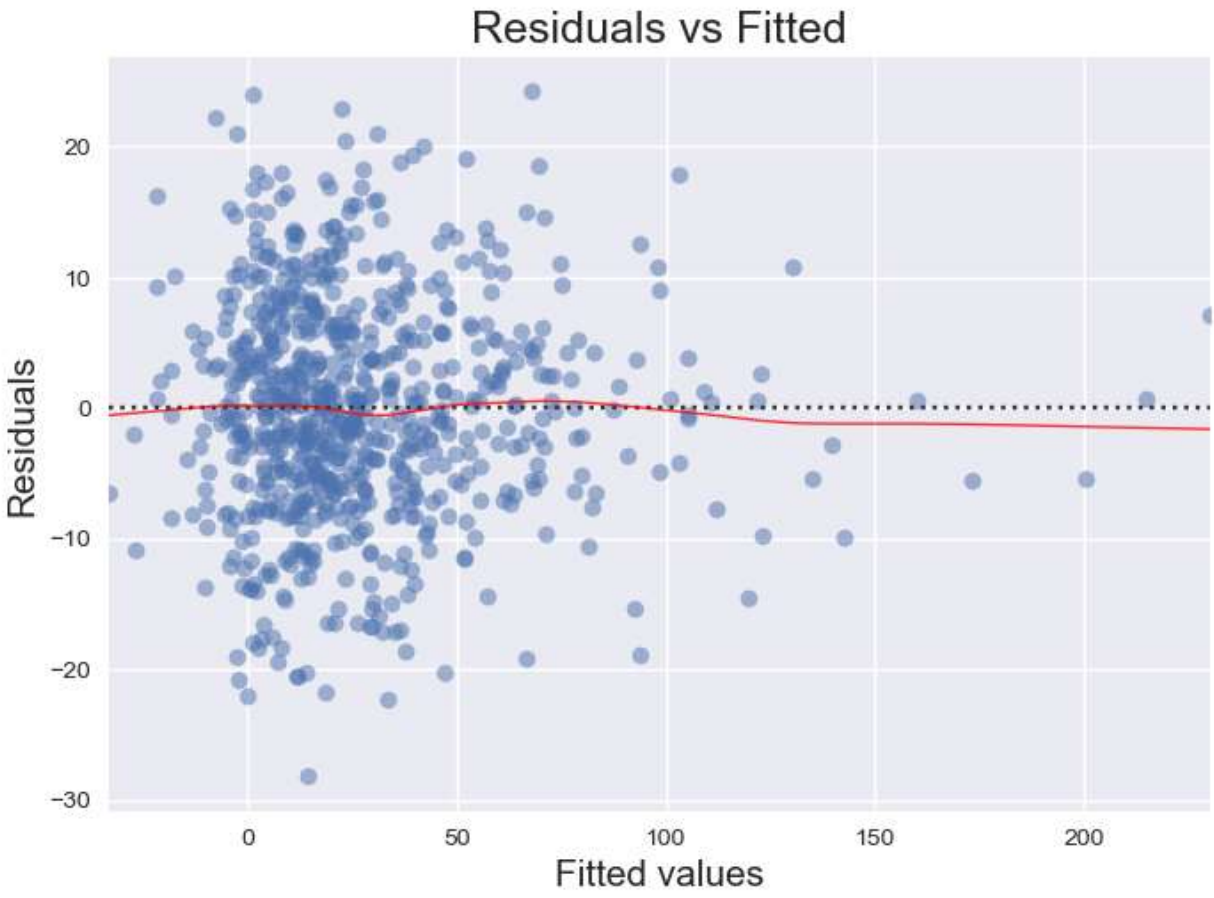
| | | | | | | |
|---|---|---|---|---|---|---|
| c_1:g_2 | -0.1827 | 1.087 | -0.168 | 0.867 | -2.317 | 1.951 |
| c_1:g_3 | -0.0818 | 1.078 | -0.076 | 0.940 | -2.198 | 2.035 |
| c_1:i | 8.9924 | 6.689 | 1.344 | 0.179 | -4.142 | 22.126 |
| g_1:h | 0.3187 | 0.282 | 1.131 | 0.258 | -0.235 | 0.872 |
| g_1:i | 4.2721 | 5.402 | 0.791 | 0.429 | -6.336 | 14.880 |
| g_2:h | 0.3180 | 0.313 | 1.016 | 0.310 | -0.296 | 0.932 |
| g_2:i | 4.7287 | 5.395 | 0.877 | 0.381 | -5.864 | 15.321 |
| b:c_1 | -0.3731 | 0.430 | -0.868 | 0.385 | -1.217 | 0.470 |
| b:c_2 | 0.5036 | 0.431 | 1.167 | 0.243 | -0.343 | 1.351 |
| b:c_3 | 0.3361 | 0.422 | 0.797 | 0.426 | -0.492 | 1.164 |
| b:c_4 | -0.2106 | 0.411 | -0.513 | 0.608 | -1.017 | 0.596 |
| b:d | -0.0215 | 0.071 | -0.302 | 0.763 | -0.162 | 0.118 |
| b:e | -4.5683 | 2.435 | -1.876 | 0.061 | -9.349 | 0.212 |
| b:f | -0.0082 | 0.029 | -0.283 | 0.777 | -0.065 | 0.049 |
| b:g_1 | 0.3832 | 0.356 | 1.075 | 0.283 | -0.317 | 1.083 |
| b:g_2 | -0.0229 | 0.406 | -0.056 | 0.955 | -0.821 | 0.775 |
| b:g_3 | -0.1043 | 0.375 | -0.278 | 0.781 | -0.841 | 0.633 |
| b:h | -0.1886 | 0.112 | -1.678 | 0.094 | -0.409 | 0.032 |
| b:i | 4.4577 | 2.421 | 1.841 | 0.066 | -0.297 | 9.212 |
| c_2:c_3 | -6.716e-14 | 9.68e-14 | -0.694 | 0.488 | -2.57e-13 | 1.23e-13 |
| c_2:c_4 | -3.357e-14 | 3.18e-13 | -0.106 | 0.916 | -6.58e-13 | 5.9e-13 |
| c_2:d | 0.9078 | 0.326 | 2.788 | 0.005 | 0.268 | 1.547 |
| c_2:e | -8.2952 | 6.562 | -1.264 | 0.207 | -21.179 | 4.588 |
| c_2:f | 0.0493 | 0.078 | 0.629 | 0.529 | -0.105 | 0.203 |
| c_2:g_1 | 0.2381 | 0.985 | 0.242 | 0.809 | -1.696 | 2.172 |
| c_2:g_2 | -1.0006 | 1.045 | -0.958 | 0.339 | -3.052 | 1.051 |
| c_2:g_3 | 1.1956 | 1.075 | 1.112 | 0.266 | -0.915 | 3.306 |
| c_2:h | -0.6962 | 0.473 | -1.473 | 0.141 | -1.624 | 0.232 |
| c_2:i | 7.9420 | 6.527 | 1.217 | 0.224 | -4.873 | 20.757 |
| c_3:c_4 | 1.255e-14 | 1.38e-13 | 0.091 | 0.927 | -2.58e-13 | 2.83e-13 |
| c_3:d | 0.7600 | 0.356 | 2.136 | 0.033 | 0.061 | 1.459 |
| c_3:e | 8.8212 | 6.554 | 1.346 | 0.179 | -4.048 | 21.691 |
| c_3:f | 0.0128 | 0.081 | 0.157 | 0.875 | -0.147 | 0.172 |
| c_3:g_1 | -0.2984 | 1.068 | -0.279 | 0.780 | -2.396 | 1.799 |
| c_3:g_2 | 1.4945 | 1.025 | 1.458 | 0.145 | -0.519 | 3.508 |
| c_3:g_3 | 0.8085 | 1.054 | 0.767 | 0.443 | -1.261 | 2.878 |
| c_3:h | -0.9597 | 0.502 | -1.911 | 0.056 | -1.946 | 0.026 |
| c_3:i | -8.4328 | 6.546 | -1.288 | 0.198 | -21.286 | 4.420 |
| c_4:d | 0.5830 | 0.314 | 1.857 | 0.064 | -0.033 | 1.199 |
| c_4:e | -4.5912 | 6.121 | -0.750 | 0.453 | -16.609 | 7.427 |
| c_4:f | -0.1023 | 0.081 | -1.264 | 0.207 | -0.261 | 0.057 |
| c_4:g_1 | 0.5053 | 0.960 | 0.526 | 0.599 | -1.380 | 2.390 |
| c_4:g_2 | -0.5861 | 1.065 | -0.550 | 0.582 | -2.677 | 1.505 |
| c_4:g_3 | -1.7205 | 0.949 | -1.813 | 0.070 | -3.584 | 0.143 |
| c_4:h | -0.7289 | 0.484 | -1.507 | 0.132 | -1.679 | 0.221 |
| c_4:i | 4.5732 | 6.113 | 0.748 | 0.455 | -7.430 | 16.576 |
| d:e | 0.5021 | 1.093 | 0.460 | 0.646 | -1.643 | 2.648 |
| d:f | -0.0295 | 0.014 | -2.052 | 0.041 | -0.058 | -0.001 |
| d:g_1 | 0.5127 | 0.185 | 2.770 | 0.006 | 0.149 | 0.876 |
| d:g_2 | 0.4040 | 0.192 | 2.101 | 0.036 | 0.026 | 0.782 |
| d:g_3 | 0.8165 | 0.197 | 4.154 | 0.000 | 0.431 | 1.202 |
| d:h | -0.0008 | 0.052 | -0.015 | 0.988 | -0.104 | 0.102 |
| d:i | -0.6116 | 1.092 | -0.560 | 0.575 | -2.755 | 1.532 |
| e:f | -0.2157 | 0.459 | -0.470 | 0.638 | -1.116 | 0.685 |
| e:g_1 | -4.0005 | 5.428 | -0.737 | 0.461 | -14.658 | 6.657 |
| e:g_2 | -4.7661 | 5.432 | -0.877 | 0.381 | -15.432 | 5.899 |
| e:g_3 | -3.7872 | 6.008 | -0.630 | 0.529 | -15.583 | 8.009 |
| e:h | -1.5171 | 1.815 | -0.836 | 0.404 | -5.081 | 2.046 |
| e:i | 32.9774 | 55.229 | 0.597 | 0.551 | -75.464 | 141.419 |
| f:g_1 | 0.1070 | 0.070 | 1.526 | 0.128 | -0.031 | 0.245 |

| | | | | | | |
|---|---|---|---|---|---|---|
| f:g_2 | -0.0423 | 0.073 | -0.577 | 0.564 | -0.186 | 0.102 |
| f:g_3 | -0.0305 | 0.069 | -0.444 | 0.658 | -0.165 | 0.104 |
| f:h | -0.0346 | 0.023 | -1.532 | 0.126 | -0.079 | 0.010 |
| f:i | 0.2264 | 0.458 | 0.495 | 0.621 | -0.672 | 1.125 |
| g_3:h | -0.0075 | 0.289 | -0.026 | 0.979 | -0.575 | 0.560 |
| g_3:i | 4.0740 | 5.993 | 0.680 | 0.497 | -7.693 | 15.841 |
| h:i | 0.5782 | 1.810 | 0.320 | 0.749 | -2.975 | 4.131 |
| np.square(a) | 0.2585 | 0.083 | 3.110 | 0.002 | 0.095 | 0.422 |
| np.square(b) | 0.0442 | 0.106 | 0.415 | 0.678 | -0.165 | 0.253 |
| np.square(d) | -0.2207 | 0.080 | -2.750 | 0.006 | -0.378 | -0.063 |
| np.square(e) | -13.7452 | 27.721 | -0.496 | 0.620 | -68.176 | 40.686 |
| np.square(f) | -0.0022 | 0.004 | -0.510 | 0.610 | -0.011 | 0.006 |
| np.square(h) | -0.0745 | 0.062 | -1.204 | 0.229 | -0.196 | 0.047 |
| np.square(i) | -16.9582 | 27.537 | -0.616 | 0.538 | -71.027 | 37.110 |

| | | | |
|---|---|---|---|
| Omnibus: | 0.134 | Durbin-Watson: | 2.007 |
| Prob(Omnibus): | 0.935 | Jarque-Bera (JB): | 0.056 |
| Skew: | -0.002 | Prob(JB): | 0.972 |
| Kurtosis: | 3.042 | Cond. No. | 1.73e+16 |

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 7.16e-26. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.

Residual Standard Error:
8.909811415840341



# Reduced Decision

For the reduced model we will be selecting values with a p-value greater than 0.05, to be at a 95% CI.

# Residual Plot Observation

Compared to the original model with a non-linear plot this models lowess line seems to almost follow a more linear trajectory, significantly more than the previous one. Although the plot of residuals appears to be coneshaped showing heteroscedasticity, potentially concluding that this model is not a good fit for regression.

### (d) (5 points) Calculate the MSE value on the validation set using your full quadratic model and your reduced model. Comment on the degree of overfitting compared to the model performance on training data and the adequacy of your reduced model compared to your full model.

In [10]:
```
#%% Step 9 - MSE Comparison
print("MSE for full quadratic model: %.3f" %model2.mse_model)
print("MSE for reduced model: %.3f" %reducedmodel.mse_model)
```

```
MSE for full quadratic model: 8402.044
MSE for reduced model: 9787.161
```

# MSE Comparison

Comparing the MSE's we can see that the reduced model has a greater MSE, although by 1000, while having 12 less terms in the model. This larger MSE indicates that the reduced model could be considered underfitted to the data. The adequacy can be commmented on by viewing the the R^2 values for each model, noting that more variables leads to large R-squared typically. Keeping that in mind the difference between the two is .008 making the reduced model appear to be the better model since it has 12 variables less.

### (e) (5 points) Using your reduced model, calculate a 95% confidence interval for each validation set prediction (you can do this using the predict function in R). Calculate the percentage of true observations from your validation set that fall within your prediction interval. (For this problem, you don't need to print all of the confidence intervals. Please only print the final value of the number of true observations that fall within your confidence interval).

In [11]:
```
#%% Step 10 - Predict Confidence Interval
dt = reducedmodel.get_prediction(x_test).summary_frame(alpha = 0.05)
ym_ci_lower = dt['mean_ci_lower']
ym_ci_upper = dt['mean_ci_upper']
comparison = pd.concat([ym_ci_lower,y_test.reset_index(),ym_ci_upper], axis=1)
in_conf_int = len(comparison.query('mean_ci_lower<=response & mean_ci_upper>=response'
perc_conf = in_conf_int / len(y_test) * 100
print("A total of ", in_conf_int,"or %.2f" %perc_conf, "% of true observations fall wi
del dt, in_conf_int, model, model2, reducedmodel, reducedterms, comparison
```

```
A total of  101 or 40.73 % of true observations fall within the confidence interval.
```