In [1]:

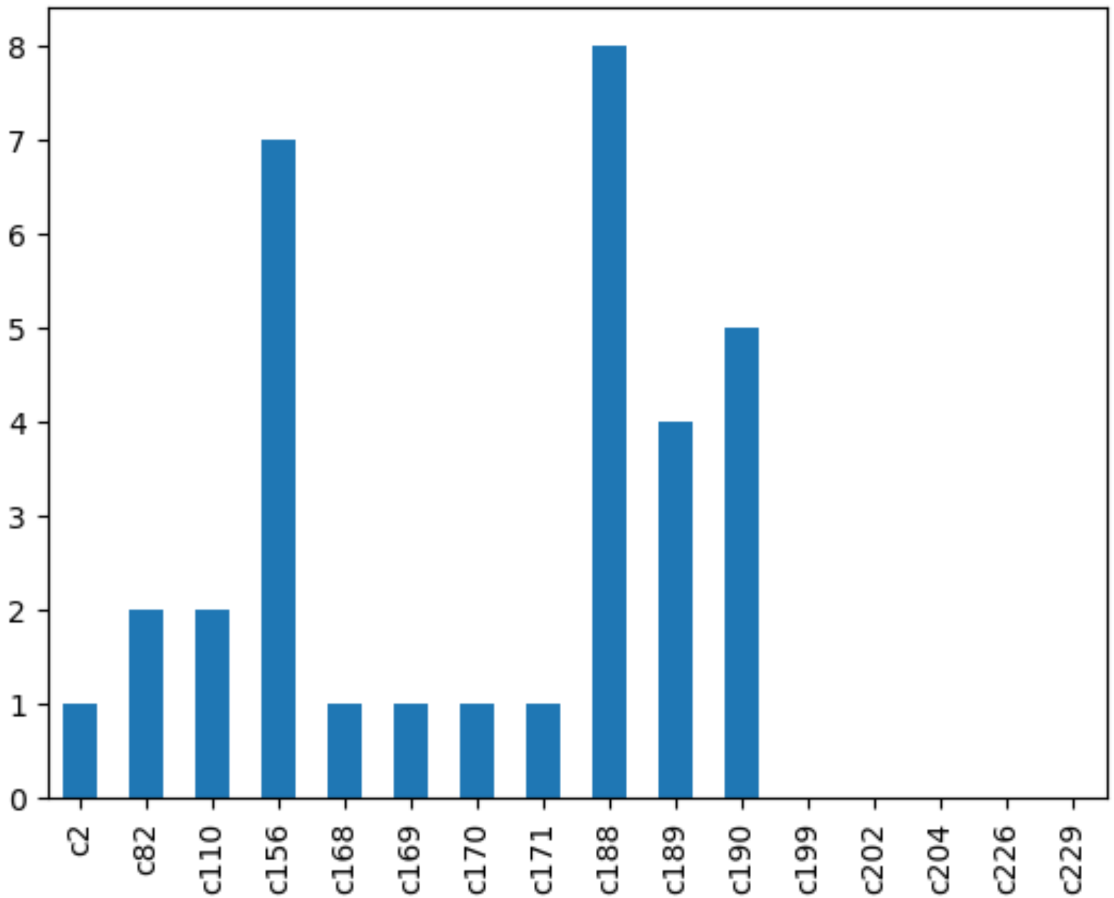
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api as sm
from sklearn.model_selection import train_test_split
```

In [2]:

```
data=pd.read_csv('E11.csv')
pd.set_option('display.max_rows',None)
pd.set_option('display.max_columns',None)
data.describe()
y=data.nunique()
w= data.apply(lambda col: col.astype(str).str.count('#REF!')).sum().drop(columns=['c82','c110'])
x=data.isnull().sum().drop(columns=['c82','c110'])
z=x+w
z1=z[z>0]
y[y<10].plot(kind='bar')
```

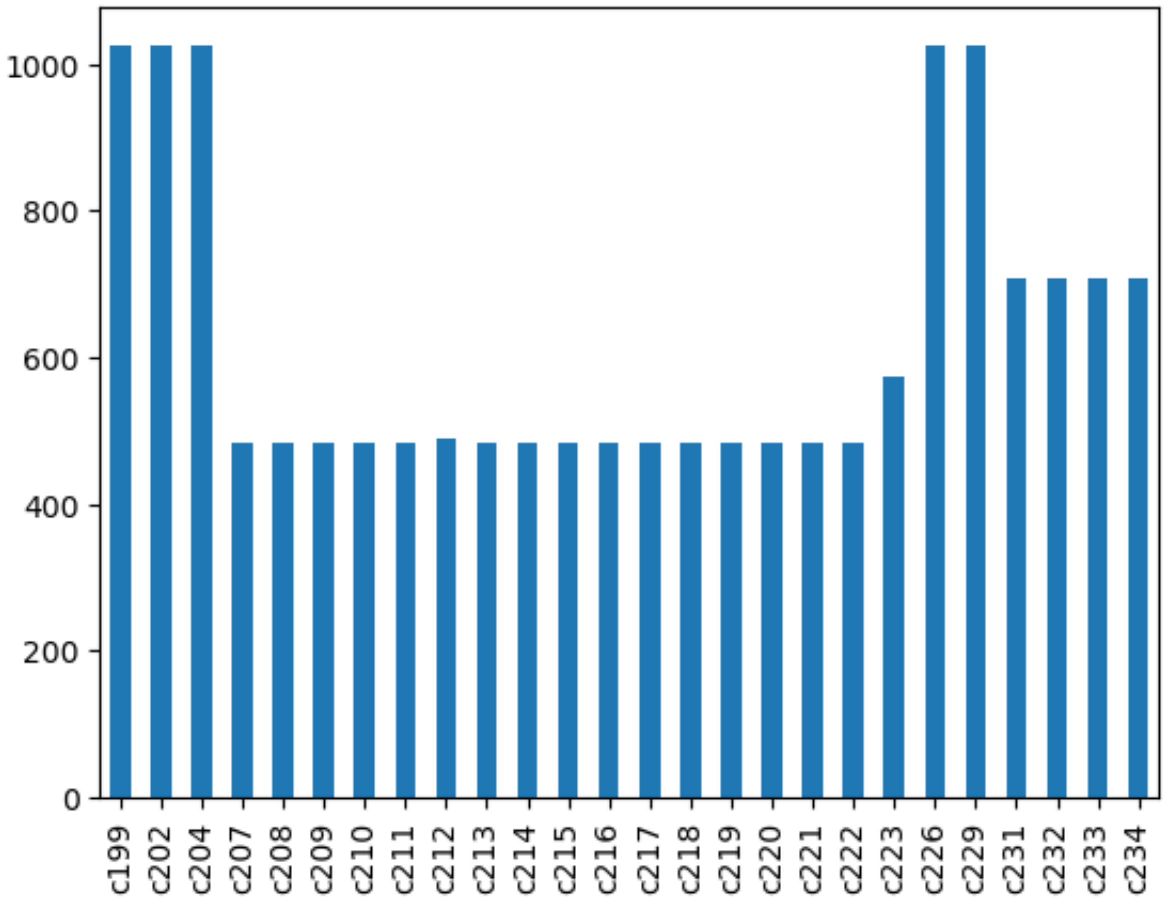
Out[2]:

```
<AxesSubplot:>
```


0
1
2
3
4
5
6
7
8
c2
c82
c110
c156
c168
c169
c170
c171
c188
c189
c190
c199
c202
c204
c226
c229

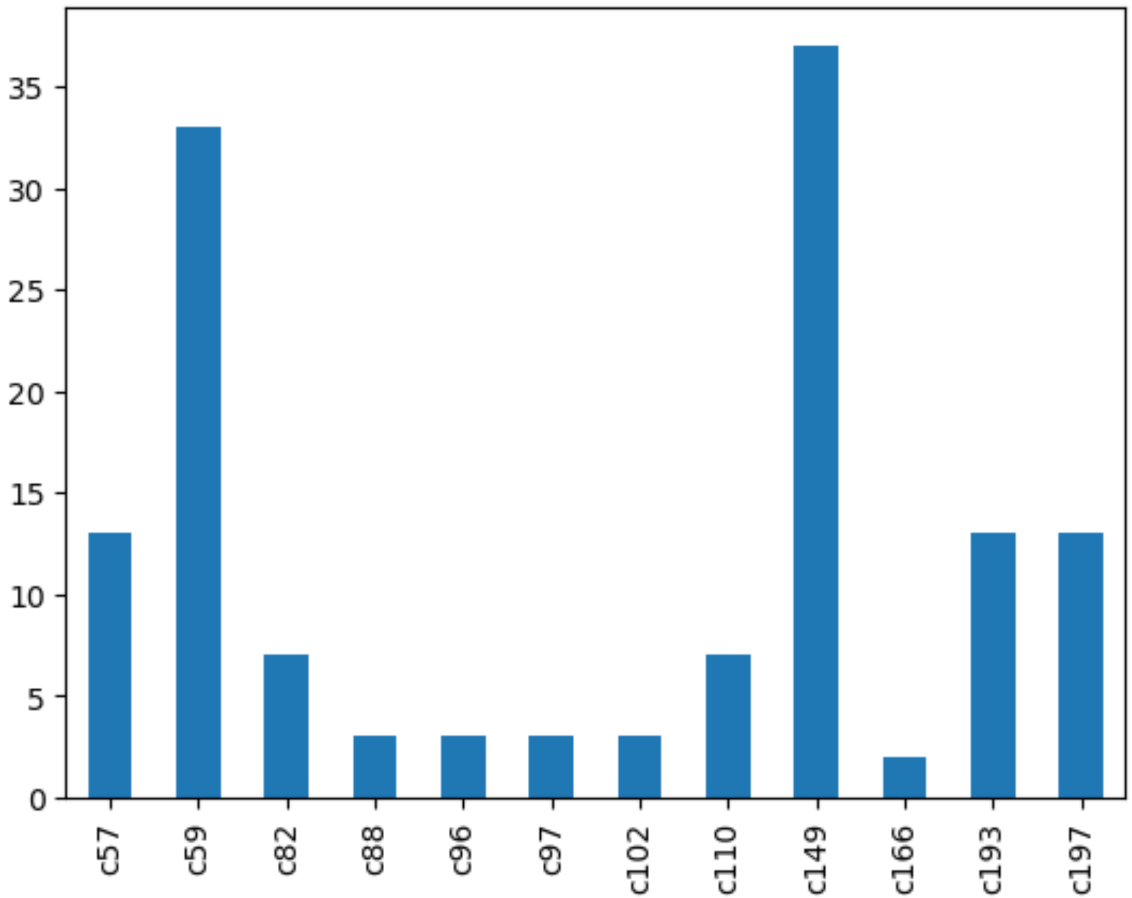
In [3]:
```
data=pd.read_csv('E11.csv')
pd.set_option('display.max_rows',None)
pd.set_option('display.max_columns',None)
data.describe()
y=data.nunique()
w= data.apply(lambda col: col.astype(str).str.count('#REF!')).sum().drop(columns=['c82','c110'])
x=data.isnull().sum().drop(columns=['c82','c110'])
z=x+w
z1=z[z>0]
x[x>50].plot(kind='bar')
```

Out[3]:
```
<AxesSubplot:>
```


1000
800
600
400
200
0
c199
c202
c204
c207
c208
c209
c210
c211
c212
c213
c214
c215
c216
c217
c218
c219
c220
c221
c222
c223
c226
c229
c231
c232
c233
c234

In [4]:
```
data=pd.read_csv('E11.csv')
pd.set_option('display.max_rows',None)
pd.set_option('display.max_columns',None)
data.describe()
y=data.nunique()
w= data.apply(lambda col: col.astype(str).str.count('#REF!')).sum().drop(columns=['c82','c110'])
x=data.isnull().sum().drop(columns=['c82','c110'])
z=x+w
z1=z[z>0]
z1[z<50].plot(kind='bar')
```

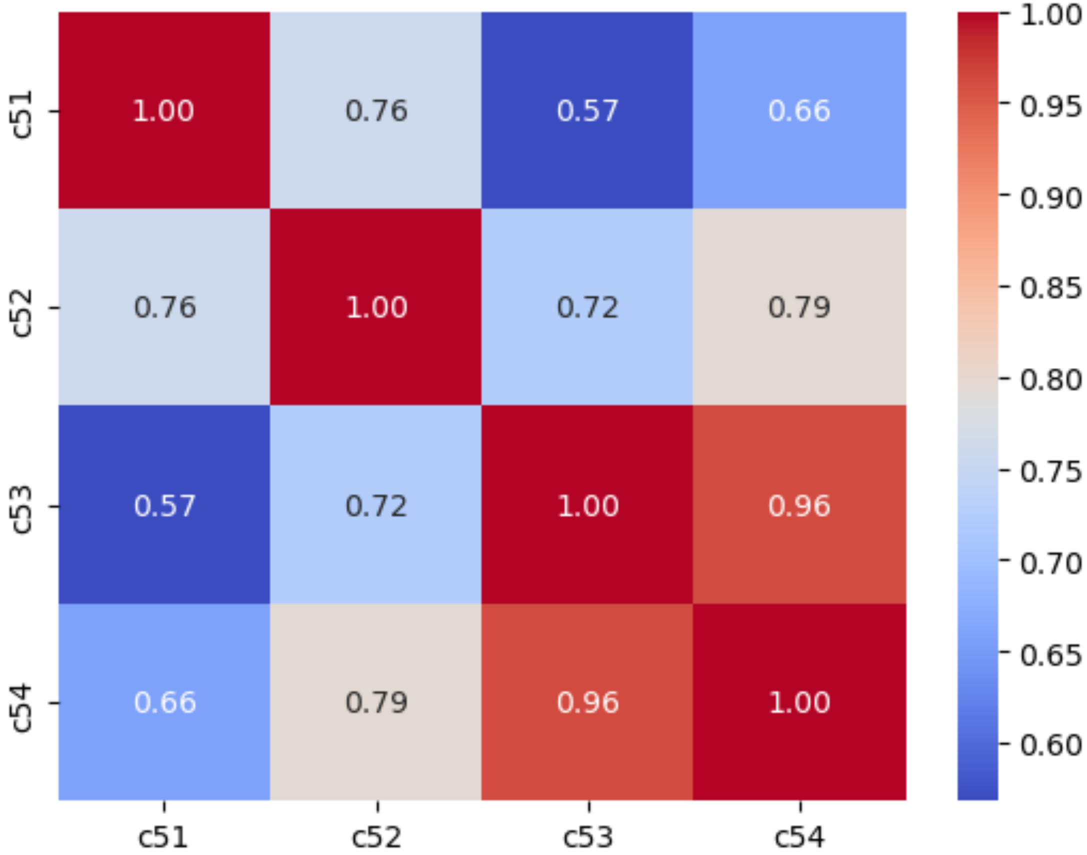
Out[4]:
```
<AxesSubplot:>
```


35
30
25
20
15
10
5
0
c57
c59
c82
c88
c96
c97
c102
c110
c149
c166
c193
c197

In [6]:
```
first_nan0=[563,548,580,580,580,580,563,563,563,578]
numberofnan0=[13,33,3,3,3,3,13,13,13,2]
coltomanage=['c57','c59','c88','c96','c97','c102','c118','c193','c197','c166']
for k in range(len(first_nan0)):
    manage=[]
    first_nan=first_nan0[k]
    numberofnan=numberofnan0[k]
    col=coltomanage[k]
    for i in range(numberofnan):
        data[col][first_nan+i]=0
    for i in range(numberofnan+2):
        manage.append(float(data[col][first_nan-1+i]))
    for i in range(numberofnan):
        data[col][first_nan+i]=np.round(manage[0]-(manage[0]-manage[numberofnan+1])/numberofnan*i,9)
```

```
/var/folders/_n/200kljtd2dg7jx4fc971742w0000gn/T/ipykernel_16395/3156396506.py:10: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#retu
rning-a-view-versus-a-copy
  data[col][first_nan+i]=0
/var/folders/_n/200kljtd2dg7jx4fc971742w0000gn/T/ipykernel_16395/3156396506.py:14: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#retu
rning-a-view-versus-a-copy
  data[col][first_nan+i]=np.round(manage[0]-(manage[0]-manage[numberofnan+1])/numberofnan*i,9)
```

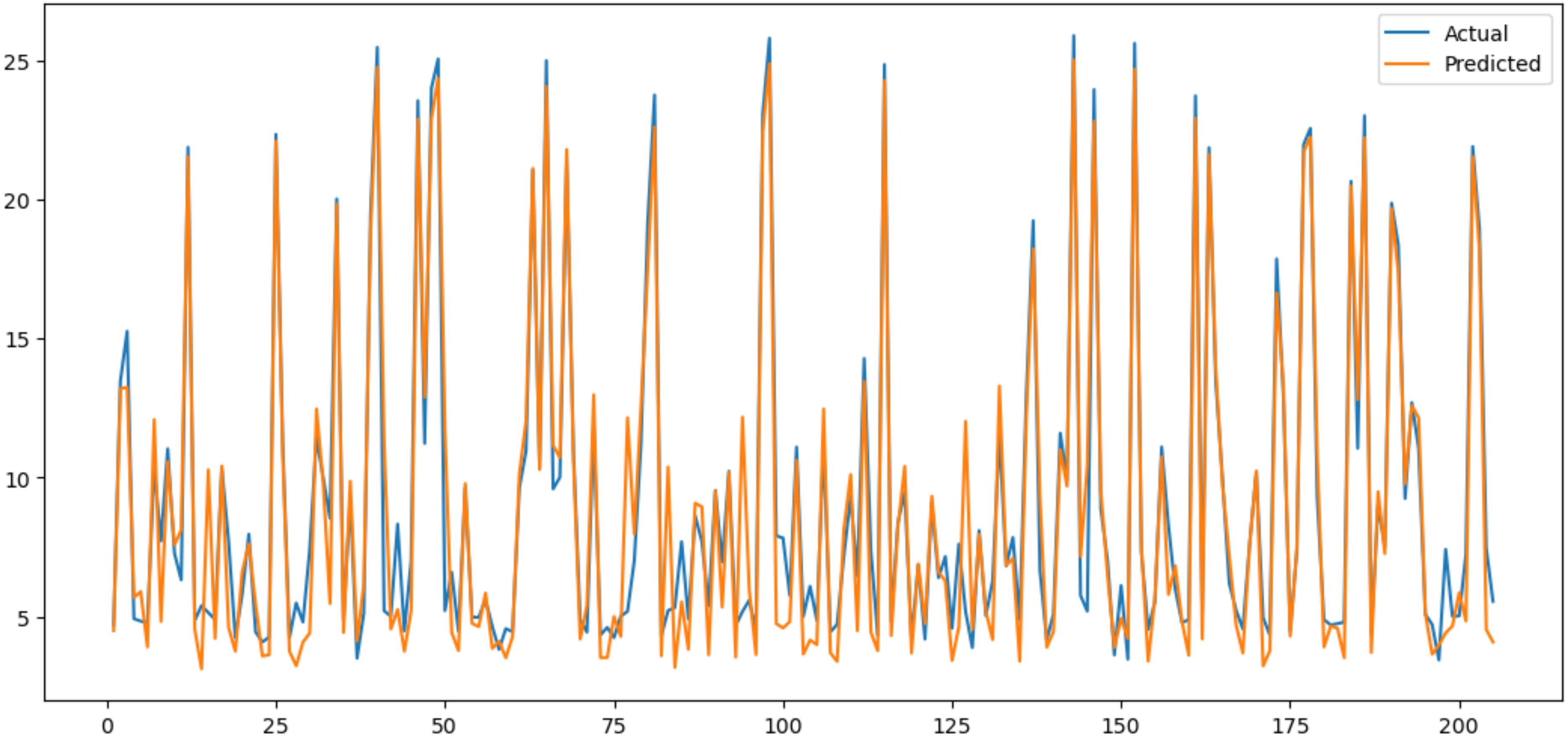
In [7]:
```
correlation_matrix=data[['c51','c52','c53','c54']].corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.show()
```



In [8]:
```
X=data['c53']
y=data['c54']
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=69)
model=sm.OLS(y_train,X_train).fit()
print(model.summary())
yp=model.predict(X_test)
l=np.arange(1,len(X_test)+1)
plt.figure(figsize=(13,6))
plt.plot(l,y_test,l,yp)
plt.legend(['Actual', 'Predicted'])
```

```
                                 OLS Regression Results
=======================================================================================
Dep. Variable:                    c54   R-squared (uncentered):                   0.978
Model:                            OLS   Adj. R-squared (uncentered):              0.978
Method:                 Least Squares   F-statistic:                          3.684e+04
Date:                Mon, 13 Nov 2023   Prob (F-statistic):                        0.00
Time:                        15:45:27   Log-Likelihood:                         -1546.6
No. Observations:                 820   AIC:                                      3095.
Df Residuals:                     819   BIC:                                      3100.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
c53            0.9166      0.005    191.941      0.000       0.907       0.926
==============================================================================
Omnibus:                      346.080   Durbin-Watson:                   1.956
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1930.676
Skew:                          -1.849   Prob(JB):                         0.00
Kurtosis:                       9.544   Cond. No.                         1.00
==============================================================================

Notes:
[1] R² is computed without centering (uncentered) since the model does not contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

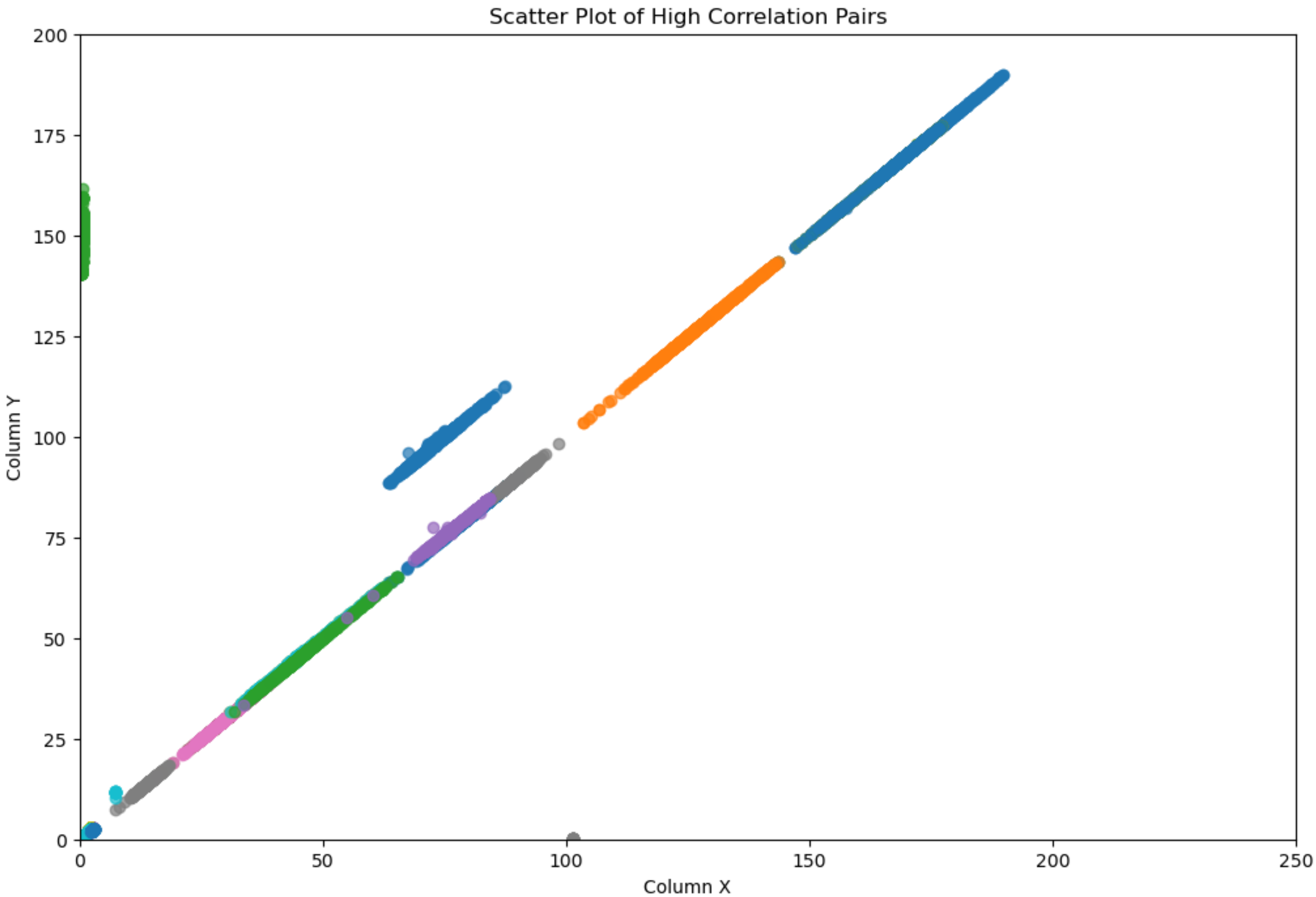
Out[8]:
```
<matplotlib.legend.Legend at 0x7fe1f8a367f0>
```



In [9]:
```
X=data.drop(columns=['c1','c2','c51','c52','c53','c54','c82','c110','c149','c156','c168','c169','c170','c171','c1
y=data['c53'] #Change as desired
```

In [10]:
```
correlation_matrix = X.corr()
corr_threshold = 0.99
high_corr_pairs = []
for i in range(len(correlation_matrix.columns)):
    for j in range(i+1, len(correlation_matrix.columns)):
        correlation = correlation_matrix.iloc[i, j]
        if (correlation) > corr_threshold:
            pair = (correlation_matrix.columns[i], correlation_matrix.columns[j], correlation)
            high_corr_pairs.append(pair)
for pair in high_corr_pairs[:10]: # first 10 pairs are displayed
    print(f"Columns {pair[0]} and {pair[1]} have correlation: {pair[2]}")
high_corr_columns = set()
for pair in high_corr_pairs:
    high_corr_columns.add(pair[0])
    high_corr_columns.add(pair[1])
plt.figure(figsize=(12, 8))
for pair in high_corr_pairs:
    plt.scatter(X[pair[0]], X[pair[1]], label=f'{pair[0]} vs {pair[1]}', alpha=0.7)
plt.title("Scatter Plot of High Correlation Pairs")
plt.xlabel("Column X")
plt.ylabel("Column Y")
plt.ylim(0,200)
plt.xlim(0,250)
plt.show()
```

```
Columns c3 and c148 have correlation: 0.9999930694447685
Columns c4 and c69 have correlation: 0.9999966664696086
Columns c4 and c128 have correlation: 0.9999966664696086
Columns c5 and c24 have correlation: 0.9999999999999203
Columns c6 and c25 have correlation: 0.999999999999514
Columns c17 and c136 have correlation: 0.9999764852436428
Columns c18 and c138 have correlation: 0.9999990558879929
Columns c19 and c139 have correlation: 0.999999956535309
Columns c28 and c77 have correlation: 0.9991488123567125
Columns c28 and c144 have correlation: 0.9991488133443961
```


Scatter Plot of High Correlation Pairs
200
175
150
125
100
75
50
25
0
Column Y
0
50
100
150
200
250
Column X

In [11]:
```
correlated_columns=['c148', 'c69', 'c128', 'c24', 'c25', 'c136', 'c138', 'c139', 'c77', 'c144', 'c145', 'c104', '
X=X.drop(columns=correlated_columns)
```

In [12]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=5)
dropped_vars = []
R2_values = []
dropped_vars = []
R2_values = []
X=sm.add_constant(X)
X_train = X_train.apply(pd.to_numeric, errors='coerce')
while len(X_train.columns) > 1:
    model = sm.OLS(y_train, X_train).fit()
    max_p_value = model.pvalues[1:].max()

    if max_p_value > 0.05:
        dropped_var = model.pvalues[1:].idxmax()
        X_train = X_train.drop(columns=[dropped_var])
        model = sm.OLS(y_train, X_train).fit()
        dropped_vars.append(dropped_var)
        R2_values.append(model.rsquared)
    else:
        break

summary_table = pd.DataFrame({
    "Dropped Variable": dropped_vars,
    "R2": R2_values
})
print(summary_table.head(10))
print(model.summary())
```

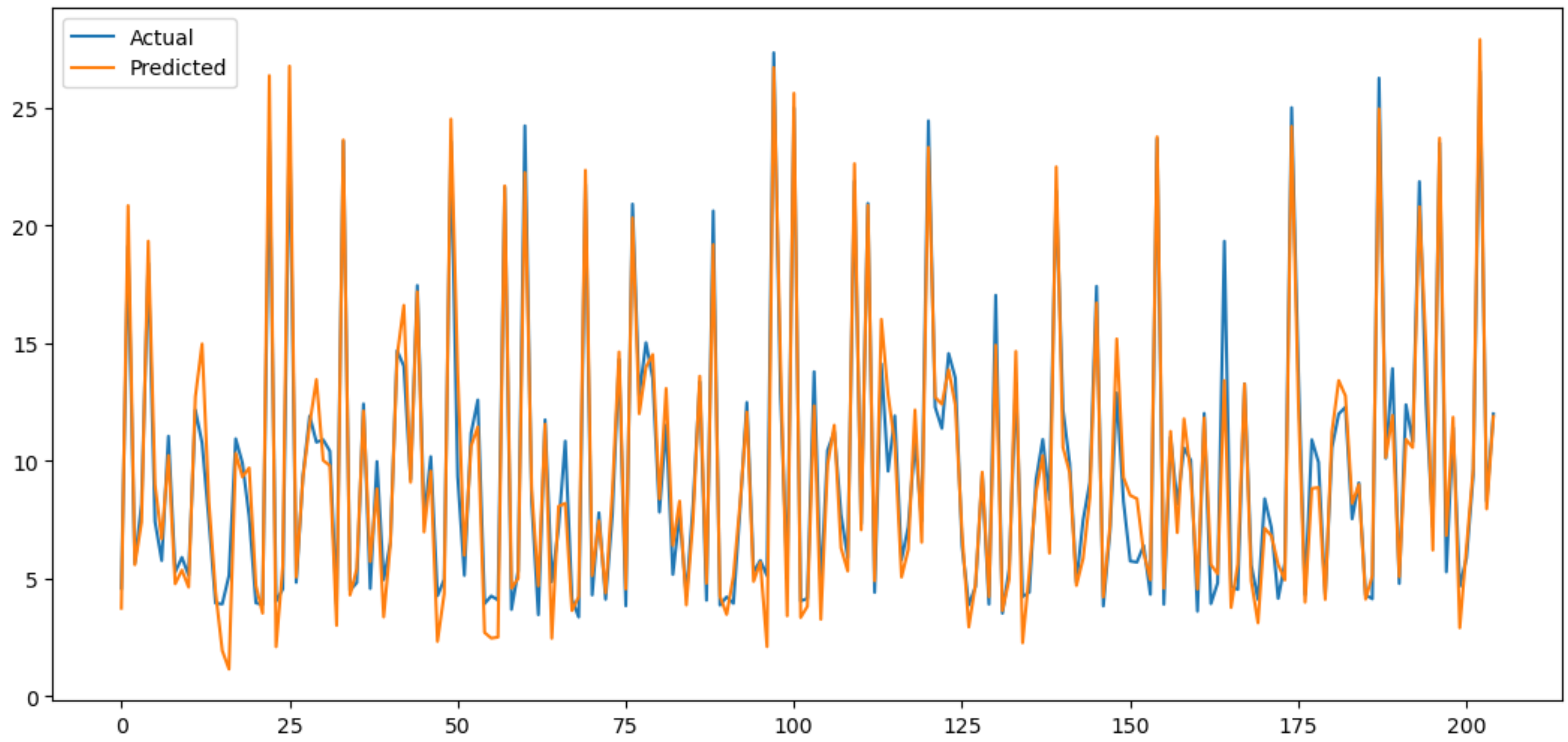
```
X_test = X_test.apply(pd.to_numeric, errors='coerce')
X_test = X_test[X_train.columns]

y_pred_test = model.predict(X_test)
df = pd.DataFrame({'Predicted': y_pred_test.tolist(), 'Actual': y_test.tolist()})
df['error'] = (df['Predicted'] - df['Actual'])

x = np.arange(len(df['Actual']))
plt.figure(figsize=(13, 6))
plt.plot(x, df['Actual'], label='Actual')
plt.plot(x, df['Predicted'], label='Predicted')
plt.legend()

plt.show()
```

```
  Dropped Variable        R2
0              c58  0.983364
1              c30  0.983364
2              c23  0.983364
3             c235  0.983364
4               c4  0.983364
5              c32  0.983363
6             c166  0.983362
7              c61  0.983361
8             c196  0.983359
9             c239  0.983357
                            OLS Regression Results
==============================================================================
Dep. Variable:                    c53   R-squared:                       0.982
Model:                            OLS   Adj. R-squared:                  0.979
Method:                 Least Squares   F-statistic:                     419.4
Date:                Mon, 13 Nov 2023   Prob (F-statistic):               0.00
Time:                        15:45:35   Log-Likelihood:                -1059.6
No. Observations:                 820   AIC:                             2307.
Df Residuals:                     726   BIC:                             2750.
Df Model:                          93
Covariance Type:            nonrobust
```

	coef	std err	t	P>|t|	[0.025	0.975]
c3	-0.2290	0.038	-5.986	0.000	-0.304	-0.154
c6	-0.6585	0.109	-6.047	0.000	-0.872	-0.445
c9	-0.2184	0.085	-2.554	0.011	-0.386	-0.051
c12	0.7557	0.120	6.276	0.000	0.519	0.992
c13	0.4730	0.086	5.475	0.000	0.303	0.643
c15	0.4657	0.109	4.292	0.000	0.253	0.679
c16	0.4760	0.115	4.141	0.000	0.250	0.702
c18	-0.1307	0.023	-5.643	0.000	-0.176	-0.085
c21	-0.1509	0.056	-2.679	0.008	-0.261	-0.040
c22	-0.1139	0.047	-2.437	0.015	-0.206	-0.022
c26	-0.3050	0.065	-4.712	0.000	-0.432	-0.178
c28	-4.3331	1.793	-2.416	0.016	-7.854	-0.812
c29	0.2771	0.060	4.600	0.000	0.159	0.395
c31	-4.8853	1.801	-2.712	0.007	-8.421	-1.349
c33	0.4211	0.158	2.669	0.008	0.111	0.731
c34	4.0666	1.481	2.745	0.006	1.158	6.975
c37	-0.6809	0.294	-2.317	0.021	-1.258	-0.104
c38	-156.6189	11.636	-13.460	0.000	-179.464	-133.774
c40	13.8799	2.830	4.905	0.000	8.324	19.435
c41	204.6175	44.192	4.630	0.000	117.859	291.376
c42	0.7313	0.221	3.307	0.001	0.297	1.165
c44	-0.2605	0.104	-2.515	0.012	-0.464	-0.057
c45	0.3348	0.070	4.760	0.000	0.197	0.473
c46	10.9214	2.435	4.486	0.000	6.141	15.701
c48	-252.2337	41.360	-6.098	0.000	-333.433	-171.034
c49	2.2566	0.263	8.573	0.000	1.740	2.773
c50	-2.452e+04	3432.686	-7.144	0.000	-3.13e+04	-1.78e+04
c56	-1.4531	0.252	-5.755	0.000	-1.949	-0.957
c57	39.5539	4.365	9.062	0.000	30.985	48.123
c59	2.2190	0.863	2.573	0.010	0.526	3.912
c60	-0.1952	0.053	-3.677	0.000	-0.299	-0.091
c62	-0.5789	0.255	-2.268	0.024	-1.080	-0.078
c63	1.1440	0.327	3.502	0.000	0.503	1.785
c64	-80.3896	14.503	-5.543	0.000	-108.862	-51.917
c65	-1.9518	0.283	-6.894	0.000	-2.508	-1.396

c66	3.4583	1.485	2.328	0.020	0.542	6.374
c67	0.2452	0.085	2.889	0.004	0.079	0.412
c72	-0.6442	0.189	-3.408	0.001	-1.015	-0.273
c73	0.5848	0.223	2.617	0.009	0.146	1.023
c74	-112.9854	25.590	-4.415	0.000	-163.225	-62.746
c76	4.8697	1.778	2.739	0.006	1.379	8.361
c78	-3765.2923	583.407	-6.454	0.000	-4910.659	-2619.925
c79	1858.6809	247.805	7.501	0.000	1372.180	2345.181
c80	-86.9993	41.425	-2.100	0.036	-168.327	-5.672
c81	1.059e+04	1420.276	7.458	0.000	7803.415	1.34e+04
c84	-3.696e+04	2994.564	-12.342	0.000	-4.28e+04	-3.11e+04
c85	-559.6237	60.139	-9.305	0.000	-677.691	-441.556
c86	-1343.2570	262.442	-5.118	0.000	-1858.492	-828.022
c88	-234.7921	48.017	-4.890	0.000	-329.061	-140.523
c89	39.3104	7.682	5.117	0.000	24.229	54.392
c90	3615.7699	718.541	5.032	0.000	2205.104	5026.436
c91	-441.1915	89.802	-4.913	0.000	-617.495	-264.888
c92	2324.1207	353.704	6.571	0.000	1629.717	3018.525
c93	-70.3289	27.193	-2.586	0.010	-123.715	-16.942
c94	778.0640	151.339	5.141	0.000	480.949	1075.179
c95	6.3653	2.602	2.446	0.015	1.257	11.474
c96	120.5638	24.378	4.946	0.000	72.705	168.423
c97	18.8536	7.899	2.387	0.017	3.345	34.362
c98	-2758.7118	408.379	-6.755	0.000	-3560.456	-1956.968
c99	780.2489	113.477	6.876	0.000	557.467	1003.031
c100	8.2205	1.800	4.568	0.000	4.687	11.754
c102	-1.1339	0.421	-2.692	0.007	-1.961	-0.307
c103	-39.7705	5.130	-7.752	0.000	-49.842	-29.699
c108	-4263.6659	861.228	-4.951	0.000	-5954.460	-2572.872
c116	250.0350	35.001	7.144	0.000	181.320	318.750
c117	-0.1063	0.015	-6.968	0.000	-0.136	-0.076
c118	-49.3430	11.327	-4.356	0.000	-71.581	-27.105
c124	-0.7585	0.202	-3.760	0.000	-1.155	-0.362
c133	44.8430	9.843	4.556	0.000	25.520	64.166
c140	2.9280	0.389	7.536	0.000	2.165	3.691
c142	-1.7176	0.252	-6.807	0.000	-2.213	-1.222
c143	-0.1092	0.036	-3.072	0.002	-0.179	-0.039
c126	2.454e+04	3432.931	7.149	0.000	1.78e+04	3.13e+04

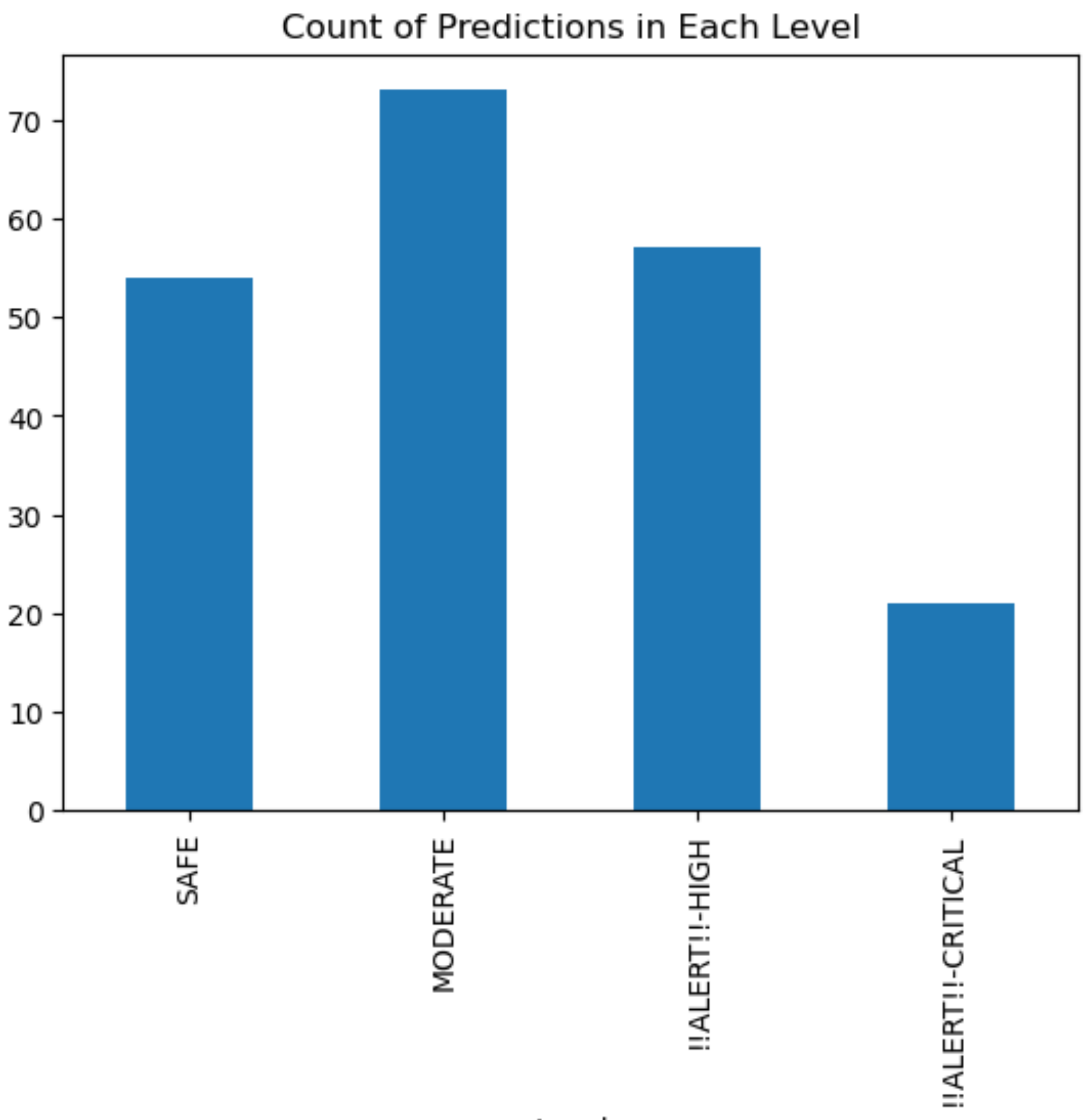
```
c131         18.9837      2.914      6.515      0.000      13.263      24.704
c132      -6060.5068    941.898     -6.434      0.000   -7909.676   -4211.337
c150          0.4259      0.104      4.115      0.000       0.223       0.629
c152        127.8454     15.896      8.043      0.000      96.639     159.052
c153          0.1374      0.029      4.748      0.000       0.081       0.194
c154         -1.7962      0.382     -4.697      0.000      -2.547      -1.045
c155          0.1355      0.054      2.521      0.012       0.030       0.241
c164        115.2759      9.346     12.335      0.000      96.928     133.624
c175       2.037e+07   9.56e+06      2.131      0.033     1.6e+06    3.91e+07
c176          1.8925      0.646      2.930      0.003       0.624       3.161
c179      -2.037e+07   9.56e+06     -2.131      0.033   -3.91e+07    -1.6e+06
c180         -1.4198      0.463     -3.065      0.002      -2.329      -0.510
c182         -0.0746      0.027     -2.780      0.006      -0.127      -0.022
c183       2.037e+07   9.56e+06      2.131      0.033     1.6e+06    3.91e+07
c191         13.4818      2.252      5.986      0.000       9.061      17.903
c193         -3.8429      0.662     -5.801      0.000      -5.143      -2.542
c198          0.9769      0.182      5.361      0.000       0.619       1.335
c201        226.3692     63.294      3.576      0.000     102.108     350.630
c205         -0.8236      0.223     -3.699      0.000      -1.261      -0.387
c230         -0.7044      0.223     -3.163      0.002      -1.142      -0.267
c241         -2.9924      0.660     -4.532      0.000      -4.289      -1.696
==============================================================================
Omnibus:                        8.148   Durbin-Watson:                   1.969
Prob(Omnibus):                  0.017   Jarque-Bera (JB):               10.276
Skew:                           0.115   Prob(JB):                      0.00587
Kurtosis:                       3.498   Cond. No.                     1.08e+12
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 3.2e-15. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```



In [13]:

```
bins = [-np.inf, 5, 10, 20, np.inf]
labels = ['SAFE', 'MODERATE', '!!ALERT!!-HIGH', '!!ALERT!!-CRITICAL']
y_pred_levels = pd.cut(y_pred_test, bins=bins, labels=labels, include_lowest=True)
df_results = pd.DataFrame({'Predicted': y_pred_test, 'Level': y_pred_levels})
print(df_results.head(10))
df_results.groupby('Level').size().plot(kind='bar', title='Count of Predictions in Each Level')
plt.show()
```

```
     Predicted             Level
293   3.740007              SAFE
697  20.847745 !!ALERT!!-CRITICAL
353   5.586053          MODERATE
481   7.405671          MODERATE
823  19.337790     !!ALERT!!-HIGH
462   8.985489          MODERATE
536   6.683364          MODERATE
438  10.232335     !!ALERT!!-HIGH
347   4.774104              SAFE
148   5.356615          MODERATE
```

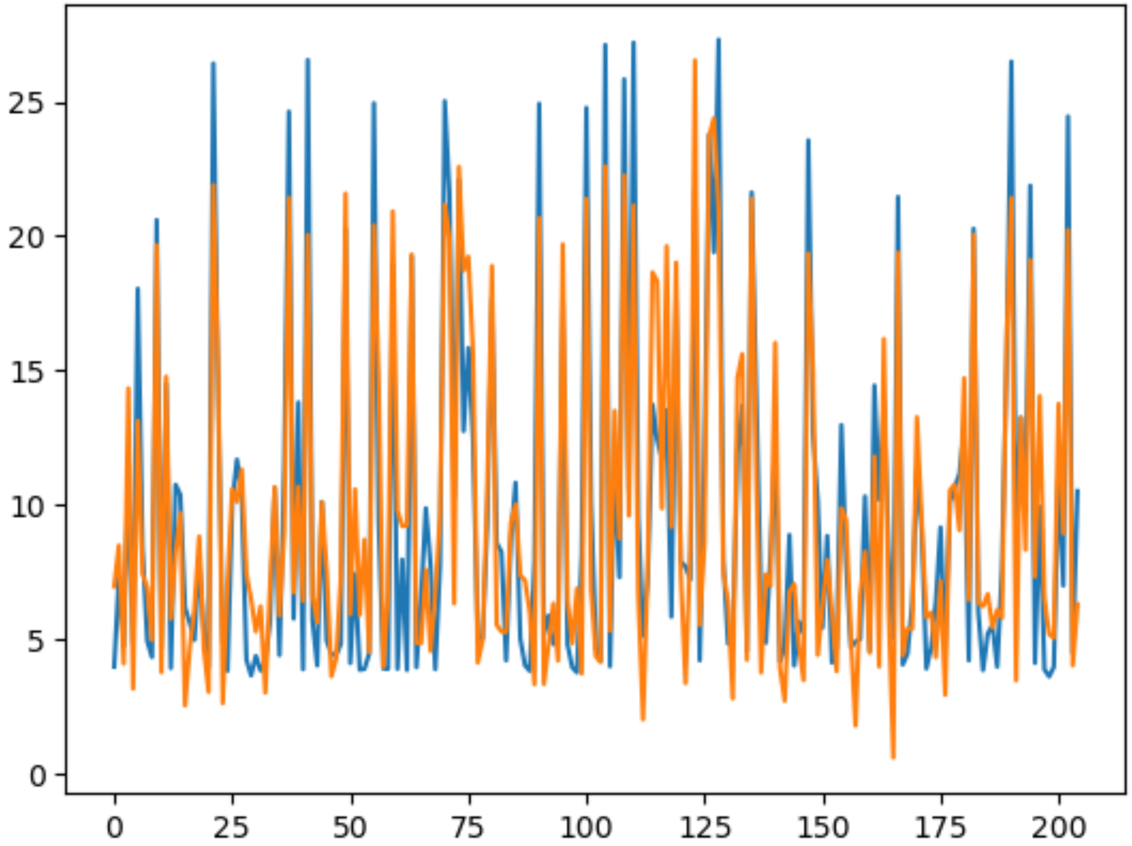

Count of Predictions in Each Level
70
60
50
40
30
20
10
0
SAFE
MODERATE
!!ALERT!!-HIGH
!!ALERT!!-CRITICAL
Level

In [14]:
```
y1=data['c53'] #change as desired
X1=data[['c26', 'c27', 'c28', 'c29', 'c30', 'c31', 'c32',
'c33', 'c39', 'c139', 'c142', 'c143', 'c155', 'c156', 'c157', 'c158', 'c160', 'c161', 'c162', 'c163']]
X=sm.add_constant(X)
X_train, X_test, y_train, y_test = train_test_split(X1, y1, test_size=0.2, random_state=1)
X_test = X_test.reset_index(drop=True)
y_test = y_test.reset_index(drop=True)
model = sm.OLS(y_train, X_train).fit()
dropped_vars = []
R2_values = []
while len(X.columns) > 1:
    model = sm.OLS(y_train, X_train).fit()
    max_p_value = model.pvalues[0:].max()
    if max_p_value > 0.05:
        dropped_var = model.pvalues[0:].idxmax()
        X_train = X_train.drop(columns=[dropped_var])
        model = sm.OLS(y_train, X_train).fit()
        dropped_vars.append(dropped_var)
        R2_values.append(model.rsquared)
    else:
        break
summary_table = pd.DataFrame({
 "Dropped Variable": dropped_vars,
 "R2": R2_values
})
print(summary_table.head(10))
print(model.summary())
X_test=X_test.drop(columns = dropped_vars)
y_pred_test=model.predict(X_test)
df = pd.DataFrame({'Predicted': y_pred_test.tolist(), 'Actual': y_test.tolist()})
df['error'] = (df['Predicted']-df['Actual'])
x=[]


for i in range(len(df['Actual'])):
    x.append(i)
```

```
plt.plot(x,df['Actual'])
plt.plot(x,df['Predicted'])
plt.show()
sorted_coefficients = model.params.sort_values(ascending=False)
print(sorted_coefficients)
```

```
  Dropped Variable        R2
0             c161  0.948143
1              c32  0.948141
2             c158  0.948107
3             c162  0.947981
                                 OLS Regression Results
=======================================================================================
Dep. Variable:                    c53   R-squared (uncentered):                   0.948
Model:                            OLS   Adj. R-squared (uncentered):              0.947
Method:                 Least Squares   F-statistic:                              915.7
Date:                Mon, 13 Nov 2023   Prob (F-statistic):                        0.00
Time:                        15:45:35   Log-Likelihood:                         -1973.4
No. Observations:                 820   AIC:                                      3979.
Df Residuals:                     804   BIC:                                      4054.
Df Model:                          16
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
c26            0.3656      0.114      3.197      0.001       0.141       0.590
c27           -1.0321      0.363     -2.840      0.005      -1.745      -0.319
c28            0.3156      0.106      2.977      0.003       0.108       0.524
c29           -0.2846      0.119     -2.397      0.017      -0.518      -0.052
c30            3.6710      1.021      3.595      0.000       1.667       5.675
c31            0.6169      0.092      6.688      0.000       0.436       0.798
c33            0.5778      0.181      3.196      0.001       0.223       0.933
c39           -8.8058      2.616     -3.366      0.001     -13.941      -3.671
c139          -0.3633      0.077     -4.713      0.000      -0.515      -0.212
c142          -1.3709      0.215     -6.370      0.000      -1.793      -0.948
c143           0.4975      0.060      8.311      0.000       0.380       0.615
c155           0.6634      0.023     28.972      0.000       0.618       0.708
```

```
c156          -0.9715      0.199     -4.873      0.000      -1.363      -0.580
c157          -0.1277      0.015     -8.517      0.000      -0.157      -0.098
c160          -0.0029      0.001     -2.486      0.013      -0.005      -0.001
c163           0.0504      0.005      9.468      0.000       0.040       0.061
==============================================================================
Omnibus:                        7.888   Durbin-Watson:                   2.003
Prob(Omnibus):                  0.019   Jarque-Bera (JB):                7.530
Skew:                           0.196   Prob(JB):                       0.0232
Kurtosis:                       2.743   Cond. No.                     2.72e+04
==============================================================================

Notes:
[1] R² is computed without centering (uncentered) since the model does not contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[3] The condition number is large, 2.72e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```



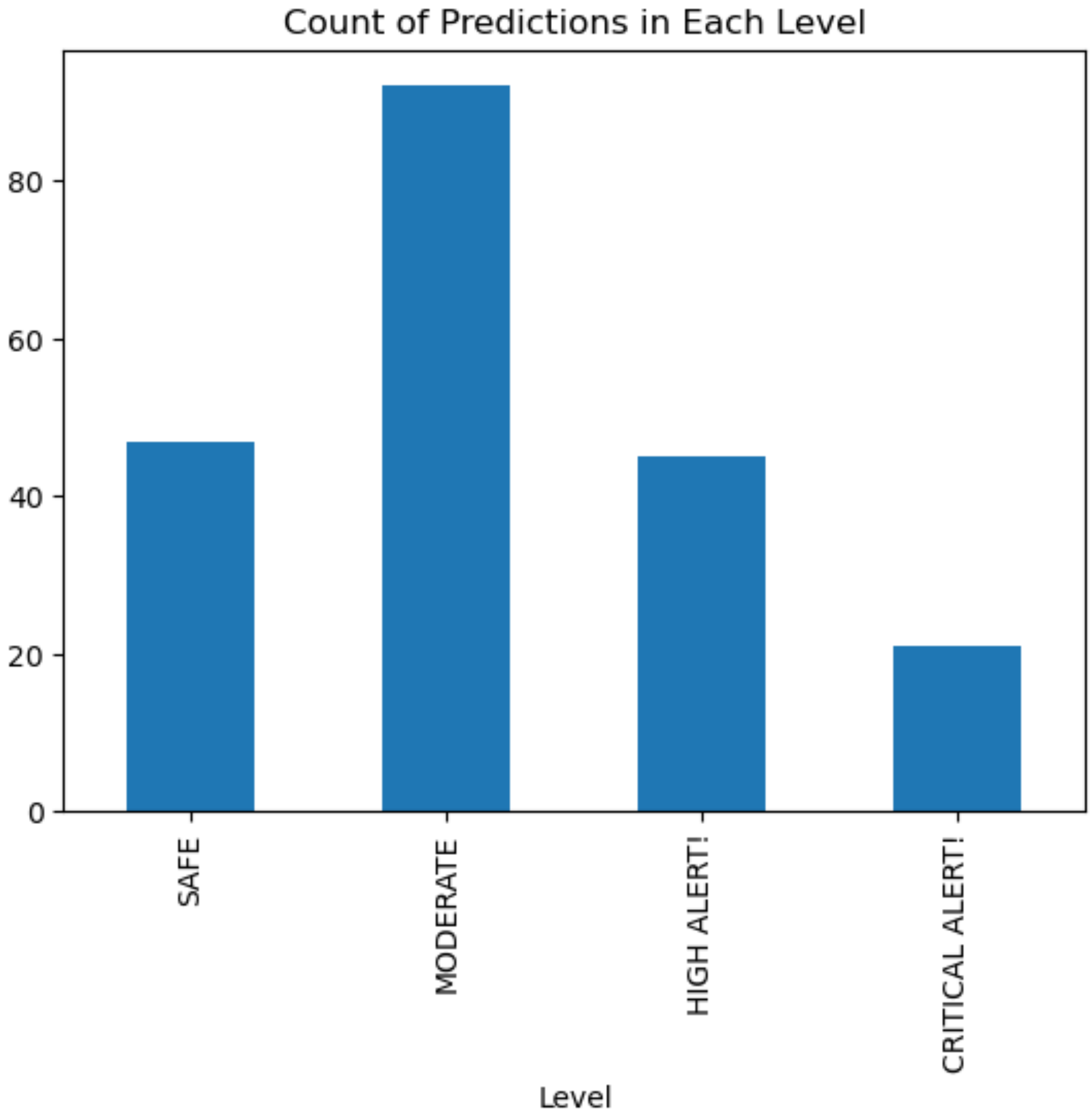
25
20
15
10
5
0
0
25
50
75
100
125
150
175
200

```
c30      3.670999
c155     0.663373
c31      0.616908
c33      0.577768
c143     0.497452
c26      0.365595
c28      0.315583
c163     0.050352
c160    -0.002877
c157    -0.127678
c29     -0.284599
c139    -0.363329
c156    -0.971483
c27     -1.032121
c142    -1.370942
c39     -8.805813
dtype: float64
```

In [15]:

```
bins = [-np.inf, 5, 10, 20, np.inf]
labels = ['SAFE', 'MODERATE', 'HIGH ALERT!', 'CRITICAL ALERT!']
y_pred_levels = pd.cut(y_pred_test, bins=bins, labels=labels, include_lowest=True)
df_results = pd.DataFrame({'Predicted': y_pred_test, 'Level': y_pred_levels})
high_alert_indices = df_results[df_results['Level'] == 'HIGH ALERT!'].index.tolist()
critical_alert_indices = df_results[df_results['Level'] == 'CRITICAL ALERT!'].index.tolist()
print(df_results.head(10))
df_results.groupby('Level').size().plot(kind='bar', title='Count of Predictions in Each Level')
plt.show()
```

	Predicted	Level
0	6.984742	MODERATE
1	8.504620	MODERATE
2	4.114058	SAFE
3	14.342545	HIGH ALERT!
4	3.185929	SAFE
5	13.140587	HIGH ALERT!
6	7.466948	MODERATE
7	6.920808	MODERATE
8	4.988905	SAFE
9	19.671613	HIGH ALERT!

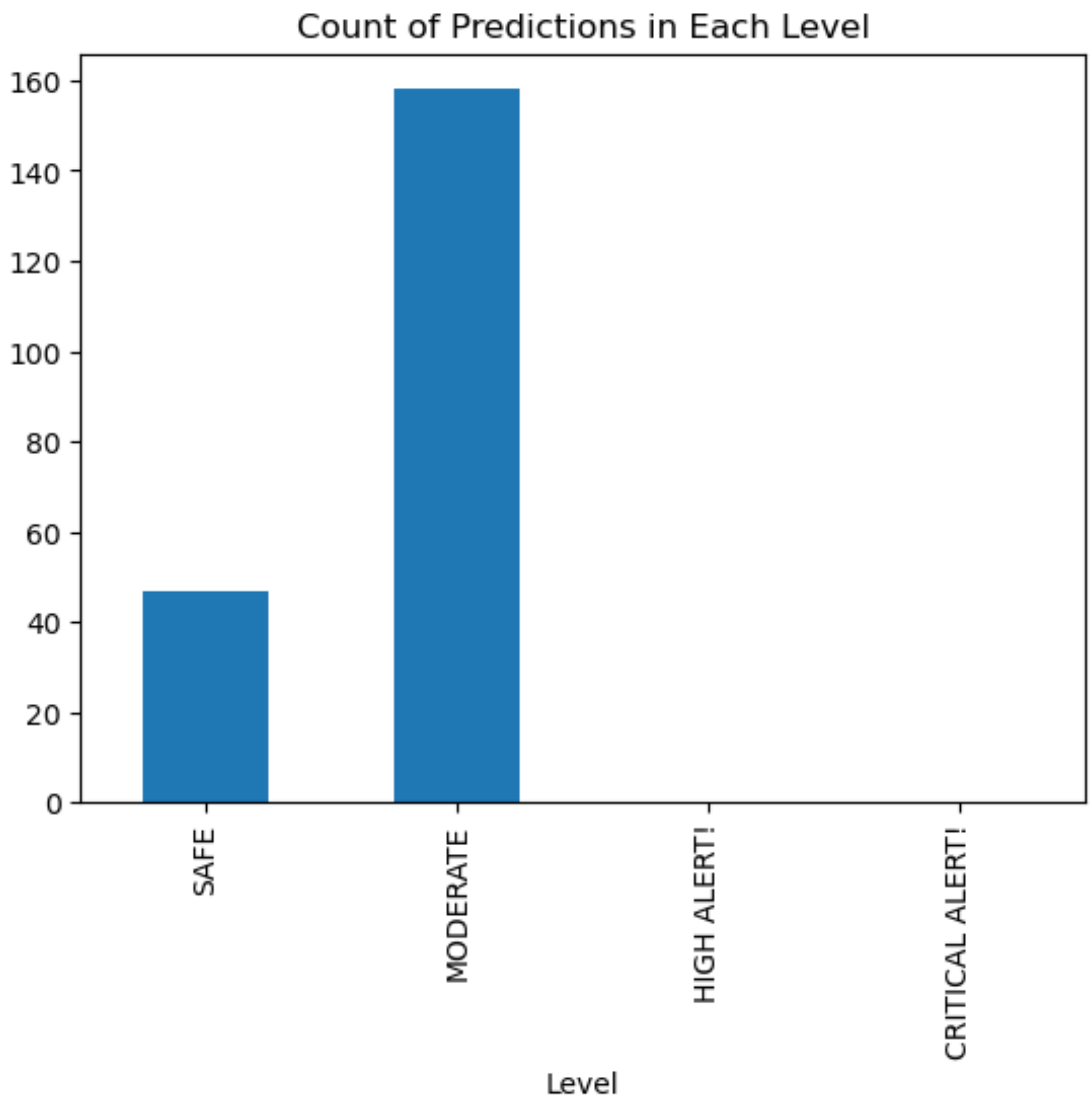

Count of Predictions in Each Level
80
60
40
20
0
SAFE
MODERATE
HIGH ALERT!
CRITICAL ALERT!
Level

In [16]:
```
epsilon=0.01
threshold=10 # Change as desired
for i in high_alert_indices:
    while (model.predict(X_test.iloc[i]) > threshold).any():
        X_test.loc[i, 'c30'] -= epsilon
for i in critical_alert_indices:
    while (model.predict(X_test.iloc[i]) > threshold).any():
        X_test.loc[i, 'c30'] -= epsilon
```

In [17]:
```
y_pred_test=model.predict(X_test)
```

In [18]:
```
bins = [-np.inf, 5, 10, 20, np.inf]
labels = ['SAFE', 'MODERATE', 'HIGH ALERT!', 'CRITICAL ALERT!']
y_pred_levels = pd.cut(y_pred_test, bins=bins, labels=labels, include_lowest=True)
df_results = pd.DataFrame({'Predicted': y_pred_test, 'Level': y_pred_levels})
high_alert_indices = df_results[df_results['Level'] == 'HIGH ALERT!'].index.tolist()
critical_alert_indices = df_results[df_results['Level'] == 'CRITICAL ALERT!'].index.tolist()
print(df_results.head(10))
df_results.groupby('Level').size().plot(kind='bar', title='Count of Predictions in Each Level')
plt.show()
```

```
   Predicted     Level
0   6.984742  MODERATE
1   8.504620  MODERATE
2   4.114058      SAFE
3   9.974056  MODERATE
4   3.185929      SAFE
5   9.983527  MODERATE
6   7.466948  MODERATE
7   6.920808  MODERATE
8   4.988905      SAFE
9   9.980175  MODERATE
```



In [19]:
```
X=data.drop(columns=['c1','c2','c82','c110','c149','c156','c168','c169','c170','c171','c188','c189','c190','c199'
y=data['c241']
```

```
correlated_columns=['c148', 'c69', 'c128', 'c24', 'c25', 'c136', 'c138', 'c139', 'c77', 'c144', 'c145', 'c104', '
X=X.drop(columns=correlated_columns)
X=sm.add_constant(X)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=9)
dropped_vars = []
R2_values = []
dropped_vars = []
R2_values = []
X=sm.add_constant(X)
X_train = X_train.apply(pd.to_numeric, errors='coerce')
while len(X_train.columns) > 1:
    model = sm.OLS(y_train, X_train).fit()
    max_p_value = model.pvalues[1:].max()

    if max_p_value > 0.05:
        dropped_var = model.pvalues[1:].idxmax()
        X_train = X_train.drop(columns=[dropped_var])
        model = sm.OLS(y_train, X_train).fit()
        dropped_vars.append(dropped_var)
        R2_values.append(model.rsquared)
    else:
        break

summary_table = pd.DataFrame({
    "Dropped Variable": dropped_vars,
    "R2": R2_values
})
print(summary_table.head(10))
print(model.summary())
X_test = X_test.apply(pd.to_numeric, errors='coerce')
X_test = X_test[X_train.columns]

y_pred_test = model.predict(X_test)
df = pd.DataFrame({'Predicted': y_pred_test.tolist(), 'Actual': y_test.tolist()})
df['error'] = (df['Predicted'] - df['Actual'])

x = np.arange(len(df['Actual']))
plt.figure(figsize=(13, 6))
```

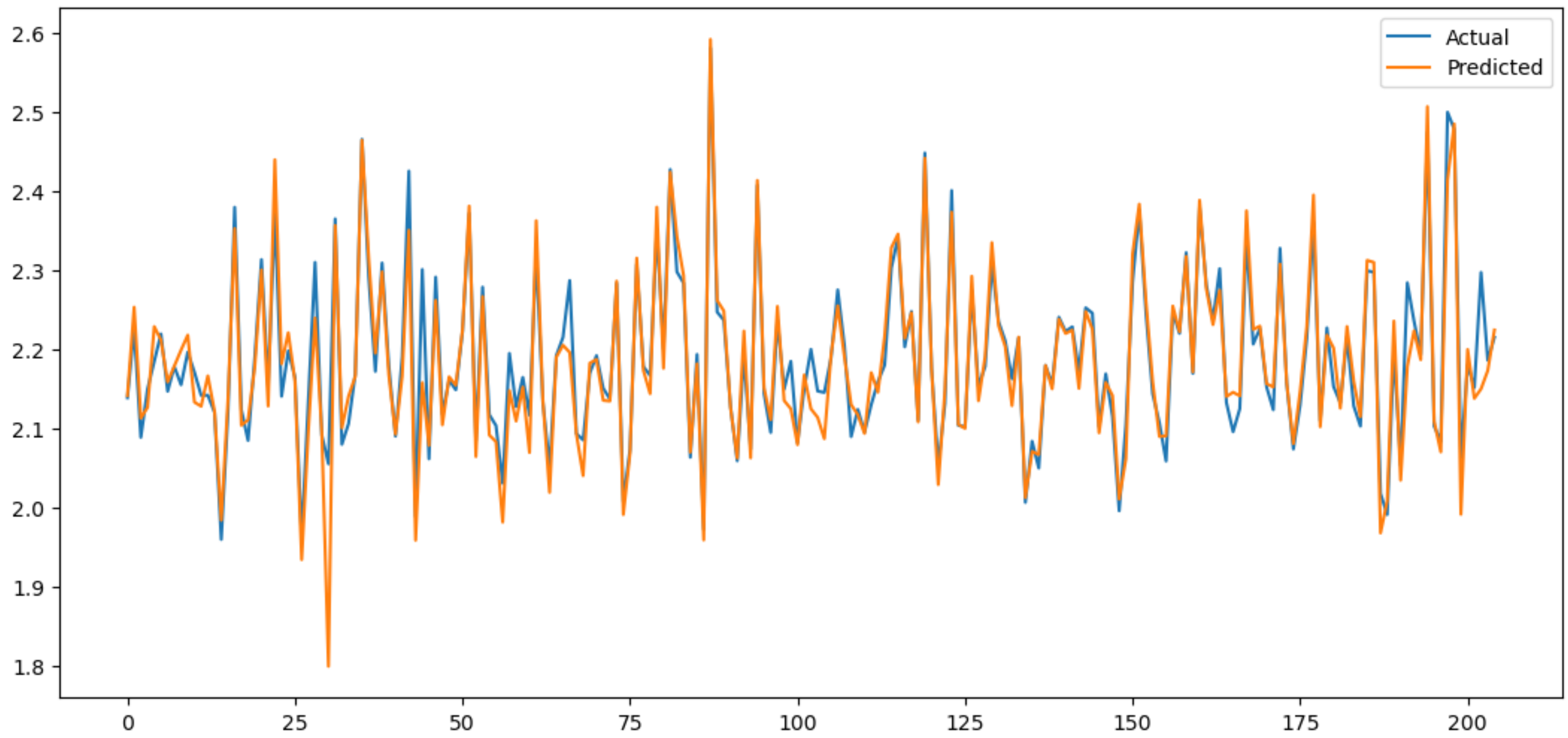
```
plt.plot(x, df['Actual'], label='Actual')
plt.plot(x, df['Predicted'], label='Predicted')
plt.legend()
plt.show()
sorted_coefficients = model.params.abs().sort_values(ascending=False)
print(sorted_coefficients.head(10))
sorted_coefficients[:10].plot(kind='bar')
```

```
  Dropped Variable        R2
0              c10  0.999259
1              c64  0.999259
2              c60  0.999259
3              c58  0.999259
4              c59  0.999259
5              c12  0.999259
6             c155  0.999259
7             c160  0.999259
8              c31  0.999259
9             c129  0.999259
                            OLS Regression Results
==============================================================================
Dep. Variable:                   c241   R-squared:                       0.999
Model:                            OLS   Adj. R-squared:                  0.999
Method:                 Least Squares   F-statistic:                     9216.
Date:                Mon, 13 Nov 2023   Prob (F-statistic):               0.00
Time:                        15:45:41   Log-Likelihood:                 1913.3
No. Observations:                 820   AIC:                            -3635.
Df Residuals:                     724   BIC:                            -3183.
Df Model:                          95
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        272.1890     56.409      4.825      0.000     161.445     382.933
c3             0.0017      0.001      2.510      0.012       0.000       0.003
c4            -0.1261      0.010    -12.825      0.000      -0.145      -0.107
c5             0.0196      0.006      3.404      0.001       0.008       0.031
c6             0.0158      0.003      5.330      0.000       0.010       0.022
```

c7	0.1250	0.027	4.607	0.000	0.072	0.178
c8	-0.0386	0.006	-6.829	0.000	-0.050	-0.028
c9	-0.0246	0.006	-4.022	0.000	-0.037	-0.013
c11	-0.0922	0.002	-37.979	0.000	-0.097	-0.087
c13	-0.0076	0.002	-3.818	0.000	-0.011	-0.004
c14	0.0174	0.008	2.228	0.026	0.002	0.033
c16	-0.0169	0.004	-4.336	0.000	-0.025	-0.009
c17	-0.0079	0.001	-5.830	0.000	-0.011	-0.005
c18	0.0049	0.001	4.354	0.000	0.003	0.007
c19	-0.0111	0.003	-3.209	0.001	-0.018	-0.004
c23	0.0018	0.001	1.979	0.048	1.43e-05	0.004
c26	-0.0047	0.001	-5.370	0.000	-0.006	-0.003
c27	-0.0237	0.009	-2.687	0.007	-0.041	-0.006
c28	0.0042	0.002	2.451	0.014	0.001	0.008
c37	-0.0446	0.007	-6.763	0.000	-0.058	-0.032
c38	-0.2965	0.051	-5.791	0.000	-0.397	-0.196
c39	-18.4896	2.030	-9.109	0.000	-22.475	-14.504
c40	0.3212	0.040	8.087	0.000	0.243	0.399
c42	-0.0229	0.006	-3.969	0.000	-0.034	-0.012
c43	0.0126	0.004	3.443	0.001	0.005	0.020
c47	0.4690	0.105	4.449	0.000	0.262	0.676
c49	-0.4112	0.093	-4.437	0.000	-0.593	-0.229
c51	0.0059	0.001	5.664	0.000	0.004	0.008
c52	-0.0077	0.002	-4.308	0.000	-0.011	-0.004
c57	-0.5114	0.079	-6.481	0.000	-0.666	-0.356
c62	0.0365	0.007	5.300	0.000	0.023	0.050
c63	-0.0316	0.009	-3.680	0.000	-0.049	-0.015
c65	0.4847	0.059	8.166	0.000	0.368	0.601
c67	-0.3876	0.052	-7.461	0.000	-0.490	-0.286
c71	-0.0289	0.006	-4.490	0.000	-0.042	-0.016
c74	1.7062	0.776	2.198	0.028	0.183	3.230
c75	0.0049	0.001	9.384	0.000	0.004	0.006
c78	-178.1708	15.107	-11.794	0.000	-207.829	-148.512
c79	-98.6597	14.982	-6.585	0.000	-128.072	-69.247
c81	-148.6193	39.758	-3.738	0.000	-226.675	-70.564
c86	112.8265	29.833	3.782	0.000	54.257	171.396
c87	-8.6546	0.994	-8.703	0.000	-10.607	-6.702
c88	3.9355	0.773	5.094	0.000	2.419	5.452

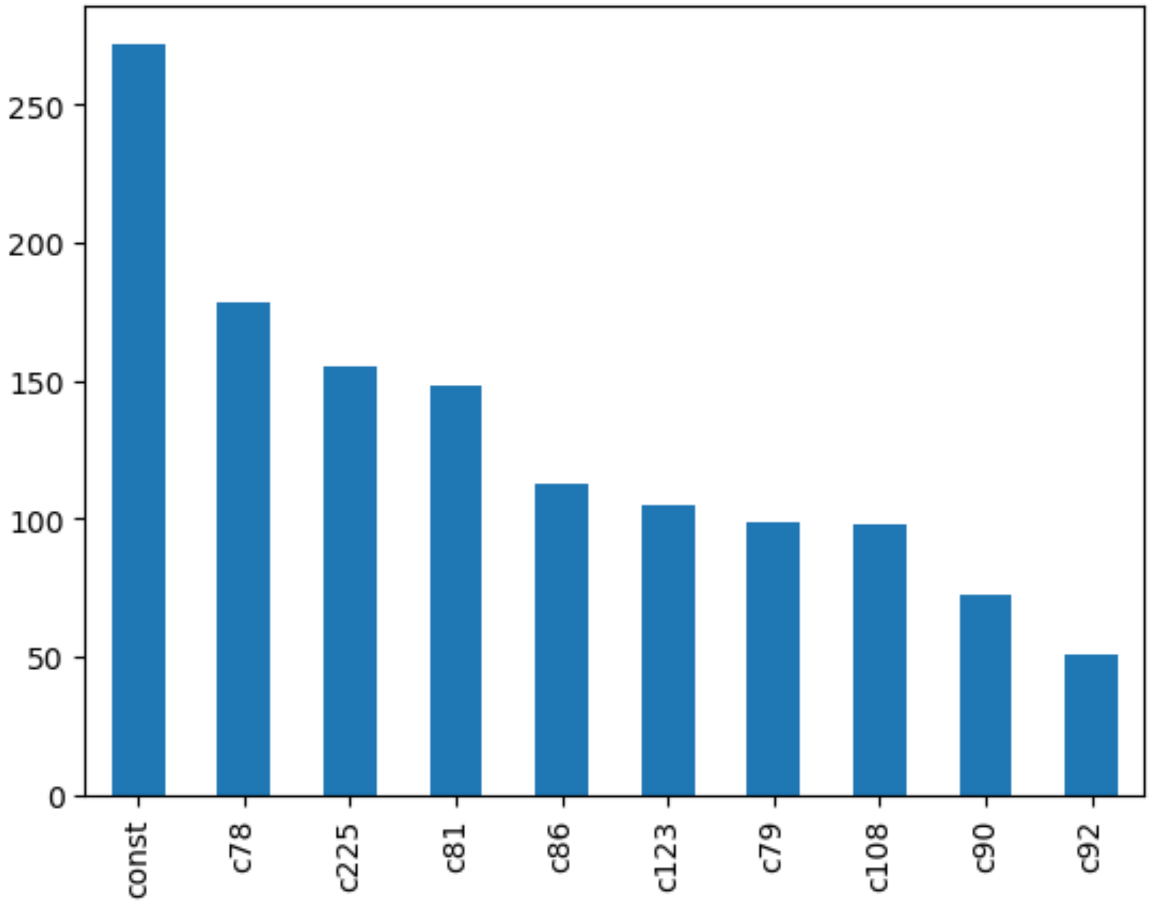
c90	72.3891	12.664	5.716	0.000	47.526	97.252
c91	4.1834	1.953	2.142	0.033	0.349	8.018
c92	50.6877	3.297	15.373	0.000	44.214	57.161
c93	-6.2011	0.723	-8.575	0.000	-7.621	-4.781
c94	20.8593	4.654	4.482	0.000	11.723	29.996
c95	-2.1456	0.475	-4.520	0.000	-3.077	-1.214
c96	-2.2092	0.392	-5.630	0.000	-2.980	-1.439
c99	-16.3195	3.142	-5.194	0.000	-22.488	-10.151
c100	0.5662	0.048	11.771	0.000	0.472	0.661
c101	0.1366	0.028	4.897	0.000	0.082	0.191
c102	0.0029	0.001	5.102	0.000	0.002	0.004
c103	0.9114	0.141	6.446	0.000	0.634	1.189
c108	-97.7174	12.185	-8.020	0.000	-121.639	-73.796
c116	0.0323	0.004	9.106	0.000	0.025	0.039
c117	-0.0005	0.000	-2.328	0.020	-0.001	-8.45e-05
c118	1.1157	0.559	1.995	0.046	0.018	2.213
c123	105.3647	14.430	7.302	0.000	77.035	133.695
c124	-0.5985	0.076	-7.831	0.000	-0.749	-0.448
c134	-0.4793	0.103	-4.674	0.000	-0.681	-0.278
c135	0.0267	0.005	5.491	0.000	0.017	0.036
c142	0.0292	0.006	4.848	0.000	0.017	0.041
c146	-0.0003	0.000	-3.214	0.001	-0.001	-0.000
c125	-6.9839	0.894	-7.811	0.000	-8.739	-5.228
c126	2.8135	0.336	8.362	0.000	2.153	3.474
c131	0.1322	0.066	2.003	0.046	0.003	0.262
c132	-44.7670	21.898	-2.044	0.041	-87.758	-1.776
c150	-0.0058	0.002	-2.383	0.017	-0.011	-0.001
c152	0.9457	0.463	2.041	0.042	0.036	1.855
c153	-0.0016	0.000	-4.883	0.000	-0.002	-0.001
c157	0.0013	0.001	2.224	0.026	0.000	0.002
c163	-0.0003	6.41e-05	-5.121	0.000	-0.000	-0.000
c175	0.3504	0.026	13.368	0.000	0.299	0.402
c177	0.1541	0.025	6.259	0.000	0.106	0.202
c178	0.1511	0.019	7.884	0.000	0.113	0.189
c180	-0.1503	0.012	-12.159	0.000	-0.175	-0.126
c181	-0.1511	0.020	-7.437	0.000	-0.191	-0.111
c182	-0.1471	0.019	-7.678	0.000	-0.185	-0.110
c183	0.2017	0.013	15.765	0.000	0.177	0.227

```
c191          0.6236      0.153      4.076      0.000       0.323       0.924
c192          0.2277      0.028      8.169      0.000       0.173       0.282
c193          0.0436      0.014      3.113      0.002       0.016       0.071
c194         -0.1784      0.019     -9.274      0.000      -0.216      -0.141
c195         -0.1127      0.010    -11.304      0.000      -0.132      -0.093
c196         -0.0304      0.005     -6.513      0.000      -0.040      -0.021
c198          0.0746      0.012      6.275      0.000       0.051       0.098
c201        -22.2219      2.235     -9.942      0.000     -26.610     -17.834
c205          0.9187      0.061     15.138      0.000       0.800       1.038
c225       -155.2364     10.298    -15.075      0.000    -175.453    -135.020
c230          0.0165      0.005      3.296      0.001       0.007       0.026
c235         -0.8172      0.415     -1.971      0.049      -1.631      -0.003
c237         -0.0082      0.001     -5.532      0.000      -0.011      -0.005
c238          0.0017      0.000      3.550      0.000       0.001       0.003
c239         -0.0029      0.001     -4.170      0.000      -0.004      -0.002
==============================================================================
Omnibus:                      337.616   Durbin-Watson:                   1.972
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             8512.547
Skew:                           1.287   Prob(JB):                         0.00
Kurtosis:                      18.573   Cond. No.                     4.35e+08
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 4.35e+08. This might indicate that there are
strong multicollinearity or other numerical problems.
```




```
const    272.189047
c78      178.170826
c225     155.236374
c81      148.619289
c86      112.826513
c123     105.364709
c79       98.659659
c108      97.717412
c90       72.389054
c92       50.687678
dtype: float64
```

Out[19]: <AxesSubplot:>



In [20]:

```
X=data.drop(columns=['c1','c2','c82','c110','c149','c156','c168','c169','c170','c171','c188','c189','c190','c199'
'c33', 'c39', 'c139', 'c142', 'c143', 'c155', 'c156', 'c157', 'c158', 'c160', 'c161', 'c162', 'c163'])
y=data['c241']
correlated_columns=['c148', 'c69', 'c128', 'c24', 'c25', 'c136', 'c138', 'c77', 'c144', 'c145', 'c104', 'c106', '
X=X.drop(columns=correlated_columns)
X=sm.add_constant(X)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=9)
```

```
dropped_vars = []
R2_values = []
dropped_vars = []
R2_values = []
X=sm.add_constant(X)
X_train = X_train.apply(pd.to_numeric, errors='coerce')
while len(X_train.columns) > 1:
    model = sm.OLS(y_train, X_train).fit()
    max_p_value = model.pvalues[1:].max()

    if max_p_value > 0.05:
        dropped_var = model.pvalues[1:].idxmax()
        X_train = X_train.drop(columns=[dropped_var])
        model = sm.OLS(y_train, X_train).fit()
        dropped_vars.append(dropped_var)
        R2_values.append(model.rsquared)
    else:
        break

summary_table = pd.DataFrame({
    "Dropped Variable": dropped_vars,
    "R2": R2_values
})
print(summary_table.head(10))
print(model.summary())
X_test = X_test.apply(pd.to_numeric, errors='coerce')
X_test = X_test[X_train.columns]

y_pred_test = model.predict(X_test)
df = pd.DataFrame({'Predicted': y_pred_test.tolist(), 'Actual': y_test.tolist()})
df['error'] = (df['Predicted'] - df['Actual'])

x = np.arange(len(df['Actual']))
plt.figure(figsize=(13, 6))
plt.plot(x, df['Actual'], label='Actual')
plt.plot(x, df['Predicted'], label='Predicted')
plt.legend()
plt.show()
```

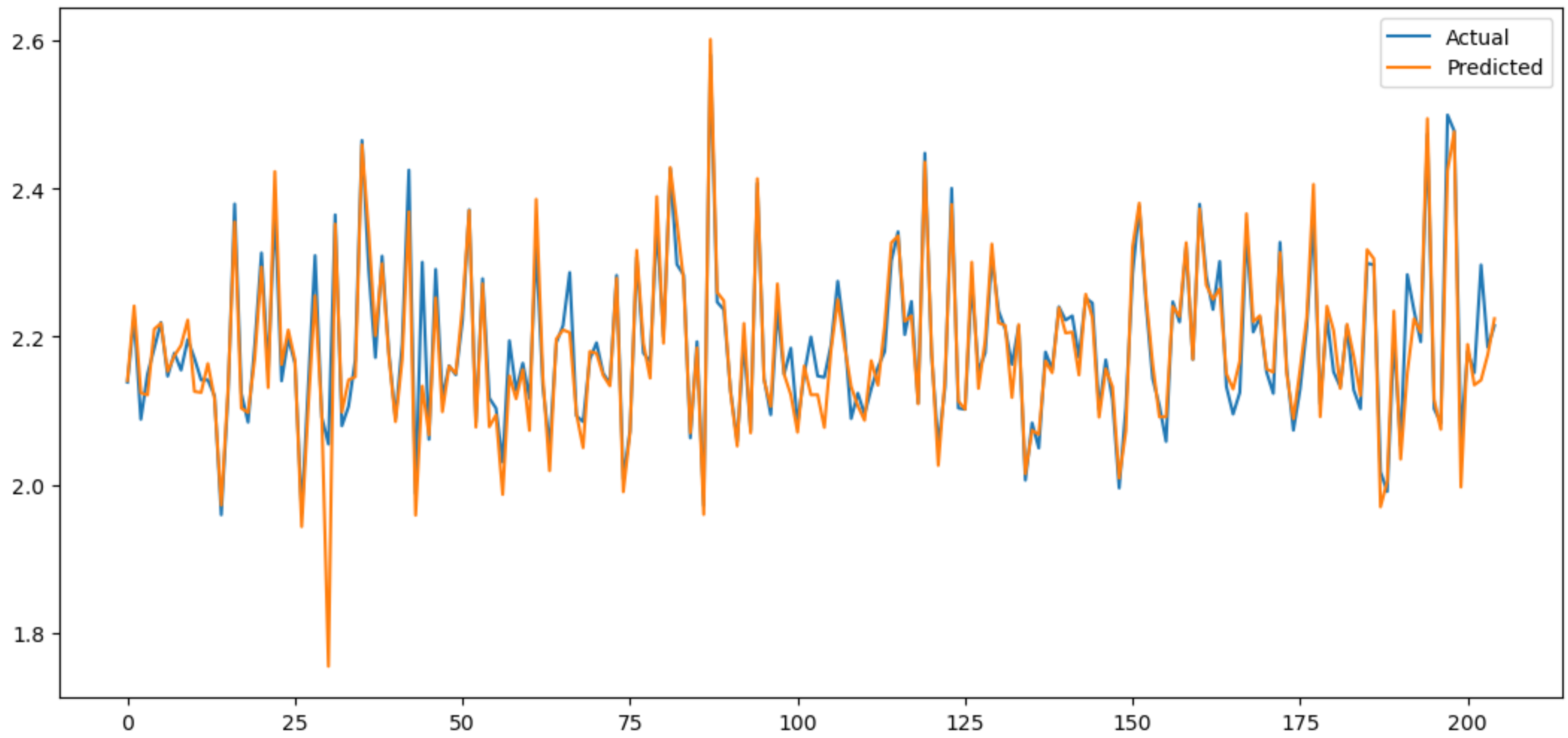
```
sorted_coefficients = model.params.abs().sort_values(ascending=False)
print(sorted_coefficients.head(10))
sorted_coefficients[:20].plot(kind='bar')
```

```
  Dropped Variable        R2
0             c187  0.999198
1              c35  0.999198
2              c58  0.999198
3              c53  0.999198
4              c22  0.999198
5              c20  0.999198
6              c41  0.999198
7              c85  0.999198
8              c74  0.999198
9              c80  0.999198
                            OLS Regression Results
==============================================================================
Dep. Variable:                   c241   R-squared:                       0.999
Model:                            OLS   Adj. R-squared:                  0.999
Method:                 Least Squares   F-statistic:                     9162.
Date:                Mon, 13 Nov 2023   Prob (F-statistic):               0.00
Time:                        15:45:44   Log-Likelihood:                 1891.0
No. Observations:                 820   AIC:                            -3598.
Df Residuals:                     728   BIC:                            -3165.
Df Model:                          91
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         76.1844    120.967      0.630      0.529    -161.301     313.670
c3             0.0018      0.001      2.861      0.004       0.001       0.003
c4            -0.1367      0.011    -13.011      0.000      -0.157      -0.116
c5             0.0231      0.006      3.919      0.000       0.012       0.035
c6             0.0155      0.003      4.799      0.000       0.009       0.022
c7             0.0803      0.026      3.076      0.002       0.029       0.132
c8            -0.0530      0.006     -9.004      0.000      -0.065      -0.041
c9            -0.0381      0.006     -6.290      0.000      -0.050      -0.026
c11           -0.0910      0.002    -37.429      0.000      -0.096      -0.086
```

c13	-0.0053	0.002	-2.433	0.015	-0.010	-0.001
c16	-0.0190	0.004	-5.165	0.000	-0.026	-0.012
c17	-0.0055	0.001	-4.156	0.000	-0.008	-0.003
c18	0.0061	0.001	4.317	0.000	0.003	0.009
c19	-0.0069	0.004	-1.975	0.049	-0.014	-4.27e-05
c37	-0.0393	0.007	-5.840	0.000	-0.053	-0.026
c38	-0.2087	0.048	-4.375	0.000	-0.302	-0.115
c42	-0.0221	0.006	-3.904	0.000	-0.033	-0.011
c43	0.0077	0.003	2.249	0.025	0.001	0.014
c47	0.3508	0.129	2.729	0.007	0.098	0.603
c48	-7.4657	0.906	-8.237	0.000	-9.245	-5.686
c49	-0.3071	0.112	-2.731	0.006	-0.528	-0.086
c51	0.0046	0.001	4.650	0.000	0.003	0.007
c52	-0.0080	0.002	-4.312	0.000	-0.012	-0.004
c56	0.0354	0.013	2.639	0.009	0.009	0.062
c57	-0.4149	0.068	-6.083	0.000	-0.549	-0.281
c62	0.0382	0.007	5.604	0.000	0.025	0.052
c63	-0.0339	0.008	-4.171	0.000	-0.050	-0.018
c65	1.4143	0.380	3.724	0.000	0.669	2.160
c67	-0.2074	0.056	-3.733	0.000	-0.316	-0.098
c71	-0.0294	0.007	-4.307	0.000	-0.043	-0.016
c73	0.0173	0.006	2.979	0.003	0.006	0.029
c75	0.0045	0.001	8.812	0.000	0.003	0.005
c78	-206.4913	15.722	-13.134	0.000	-237.357	-175.625
c79	-82.2045	16.457	-4.995	0.000	-114.513	-49.895
c81	-178.9241	46.587	-3.841	0.000	-270.386	-87.463
c86	83.3030	36.517	2.281	0.023	11.612	154.994
c87	-7.6355	1.022	-7.469	0.000	-9.642	-5.629
c88	3.7124	0.639	5.814	0.000	2.459	4.966
c90	307.4082	102.953	2.986	0.003	105.289	509.528
c91	7.4693	2.545	2.935	0.003	2.474	12.465
c92	12.0362	3.948	3.049	0.002	4.285	19.787
c93	-6.9950	0.735	-9.516	0.000	-8.438	-5.552
c94	19.7743	4.714	4.195	0.000	10.521	29.028
c95	-1.8085	0.535	-3.381	0.001	-2.858	-0.758
c96	-1.9887	0.308	-6.451	0.000	-2.594	-1.383
c98	36.2877	10.505	3.454	0.001	15.665	56.911
c99	-18.5899	3.622	-5.133	0.000	-25.700	-11.480

c100	0.6443	0.050	12.779	0.000	0.545	0.743
c101	0.1202	0.031	3.818	0.000	0.058	0.182
c102	0.0027	0.000	5.870	0.000	0.002	0.004
c103	1.0112	0.167	6.072	0.000	0.684	1.338
c116	0.0286	0.004	7.956	0.000	0.022	0.036
c123	55.9257	15.503	3.607	0.000	25.489	86.362
c124	-0.5896	0.078	-7.516	0.000	-0.744	-0.436
c134	-0.2847	0.089	-3.197	0.001	-0.460	-0.110
c135	0.0137	0.004	3.239	0.001	0.005	0.022
c146	-0.0007	0.000	-2.449	0.015	-0.001	-0.000
c147	-0.0003	0.000	-2.062	0.040	-0.001	-1.56e-05
c125	-5.1375	0.887	-5.792	0.000	-6.879	-3.396
c126	2.4783	0.350	7.083	0.000	1.791	3.165
c129	-0.0132	0.005	-2.540	0.011	-0.023	-0.003
c131	0.0191	0.007	2.712	0.007	0.005	0.033
c132	-40.9655	16.267	-2.518	0.012	-72.901	-9.030
c150	-0.0171	0.003	-5.738	0.000	-0.023	-0.011
c151	2.9536	0.117	25.242	0.000	2.724	3.183
c153	-0.0017	0.000	-5.338	0.000	-0.002	-0.001
c164	0.2440	0.036	6.723	0.000	0.173	0.315
c167	-0.9807	0.322	-3.046	0.002	-1.613	-0.349
c175	0.3951	0.027	14.748	0.000	0.343	0.448
c177	0.1328	0.025	5.334	0.000	0.084	0.182
c178	0.1465	0.020	7.467	0.000	0.108	0.185
c180	-0.1484	0.013	-11.637	0.000	-0.173	-0.123
c181	-0.1453	0.021	-6.979	0.000	-0.186	-0.104
c182	-0.1420	0.020	-7.246	0.000	-0.181	-0.104
c183	0.2501	0.012	20.589	0.000	0.226	0.274
c191	0.6099	0.158	3.858	0.000	0.300	0.920
c192	0.2106	0.030	7.052	0.000	0.152	0.269
c193	0.0961	0.031	3.147	0.002	0.036	0.156
c194	-0.2008	0.022	-9.202	0.000	-0.244	-0.158
c195	-0.1189	0.010	-11.369	0.000	-0.139	-0.098
c196	-0.0262	0.005	-4.925	0.000	-0.037	-0.016
c197	-0.0329	0.015	-2.157	0.031	-0.063	-0.003
c198	0.1024	0.018	5.761	0.000	0.068	0.137
c201	-7.7934	1.371	-5.683	0.000	-10.486	-5.101
c203	0.0733	0.030	2.466	0.014	0.015	0.132

```
c205           1.0337      0.063     16.405      0.000       0.910       1.157
c224         -12.2813      5.012     -2.450      0.015     -22.122      -2.441
c225        -174.8283     10.689    -16.355      0.000    -195.814    -153.843
c235          -1.0151      0.411     -2.471      0.014      -1.822      -0.209
c237          -0.0035      0.001     -2.352      0.019      -0.006      -0.001
c238           0.0014      0.000      2.864      0.004       0.000       0.002
c239          -0.0020      0.001     -2.711      0.007      -0.003      -0.001
==============================================================================
Omnibus:                      323.266   Durbin-Watson:                   2.021
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             9755.766
Skew:                           1.150   Prob(JB):                         0.00
Kurtosis:                      19.740   Cond. No.                     9.14e+08
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 9.14e+08. This might indicate that there are
strong multicollinearity or other numerical problems.
```




```
c90      307.408224
c78      206.491346
c81      178.924128
c225     174.828343
c86       83.303028
c79       82.204471
const     76.184432
c123      55.925735
c132      40.965496
c98       36.287727
dtype: float64
```

Out[20]: <AxesSubplot:>