hw2

April 10, 2024

[1]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

[2]:
```python
file_path = "/Users/rouren/Desktop/24S ML/HW/hw2/usa_00001.csv"
data = pd.read_csv(file_path)

crosswalk = pd.read_csv("/Users/rouren/Desktop/24S ML/HW/hw2/
 ↪PPHA_30546_MP01-Crosswalk.csv")
```

[3]:
```python
data['EDUCDC'] = data['EDUCD'].map(crosswalk.set_index('educd') ['educdc'])
print(data['EDUCDC'])
```

```
0       13.0
1       12.0
2       14.0
3       18.0
4       18.0
        …
9044    12.0
9045    12.0
9046    12.0
9047    10.0
9048    18.0
Name: EDUCDC, Length: 9049, dtype: float64
```

[4]:
```python
# Create dummy variables
data['HSDIP'] = data['EDUCD'].apply(lambda x: 1 if (x >= 62 and x <101) else 0)
data['COLDIP'] = data['EDUCD'].apply(lambda x: 1 if x >= 101 else 0)
data['WHITE'] = (data['RACE'] == 1).astype(int)
data['BLACK'] = (data['RACE'] == 2).astype(int)
data['HISPANIC'] = (data['HISPAN'] == 1).astype(int)
data['MARRIED'] = ((data['MARST'] == 1) | (data['MARST'] == 2)).astype(int)
data['FEMALE'] = (data['SEX'] == 2).astype(int)
data['VET'] = (data['VETSTAT'] == 2).astype(int)
```

[5]:
```
data['HSDIP_EDUCDC'] = data['HSDIP'] * data['EDUCDC']
data['COLDIP_EDUCDC'] = data['COLDIP'] * data['EDUCDC']

data['AGE2'] = data['AGE'] ** 2
data['LNINCWAGE'] = np.log1p(data['INCWAGE'])

data[['AGE', 'AGE2', 'INCWAGE', 'LNINCWAGE']]
```

[5]:

	AGE	AGE2	INCWAGE	LNINCWAGE
0	20	400	15700	9.661480
1	38	1444	55000	10.915107
2	31	961	55000	10.915107
3	53	2809	89000	11.396403
4	52	2704	49000	10.799596
…	…	…	…	…
9044	59	3481	48000	10.778977
9045	64	4096	35000	10.463132
9046	54	2916	60000	11.002117
9047	51	2601	28100	10.243560
9048	65	4225	122000	11.711785

[9049 rows x 4 columns]

[6]:
```
# 1

summary_stats = data[['YEAR', 'INCWAGE', 'LNINCWAGE', 'EDUCD', 'SEX', 'AGE',
    'AGE2', 'RACE', 'HISPAN', 'MARST', 'NCHILD', 'VETSTAT', 'HSDIP', 'COLDIP']].
    describe()
print(summary_stats)
```

```
         YEAR        INCWAGE    LNINCWAGE        EDUCD          SEX  \
count  9049.0    9049.000000  9049.000000  9049.000000  9049.000000
mean   2022.0   61854.084429    10.084951    81.698199     1.483147
std       0.0   72405.510157     2.560846    23.533595     0.499744
min    2022.0       0.000000     0.000000     2.000000     1.000000
25%    2022.0   22400.000000    10.016861    63.000000     1.000000
50%    2022.0   45000.000000    10.714440    81.000000     1.000000
75%    2022.0   78000.000000    11.264477   101.000000     2.000000
max    2022.0  761000.000000    13.542390   116.000000     2.000000

               AGE         AGE2         RACE      HISPAN        MARST  \
count  9049.000000  9049.000000  9049.000000  9049.00000  9049.000000
mean     41.781523  1919.084650     2.563488     0.34700     2.998011
std      13.168453  1110.989446     2.604232     0.95433     2.283653
min      18.000000   324.000000     1.000000     0.00000     1.000000
25%      31.000000   961.000000     1.000000     0.00000     1.000000
50%      42.000000  1764.000000     1.000000     0.00000     1.000000
75%      53.000000  2809.000000     3.000000     0.00000     6.000000
```

```
max      65.000000  4225.000000     9.000000     4.00000     6.000000

             NCHILD      VETSTAT        HSDIP       COLDIP
count  9049.000000  9049.000000  9049.000000  9049.000000
mean      0.839098     1.040557     0.521605     0.410764
std       1.145462     0.197272     0.499561     0.492000
min       0.000000     1.000000     0.000000     0.000000
25%       0.000000     1.000000     0.000000     0.000000
50%       0.000000     1.000000     1.000000     0.000000
75%       2.000000     1.000000     1.000000     1.000000
max       8.000000     2.000000     1.000000     1.000000
```

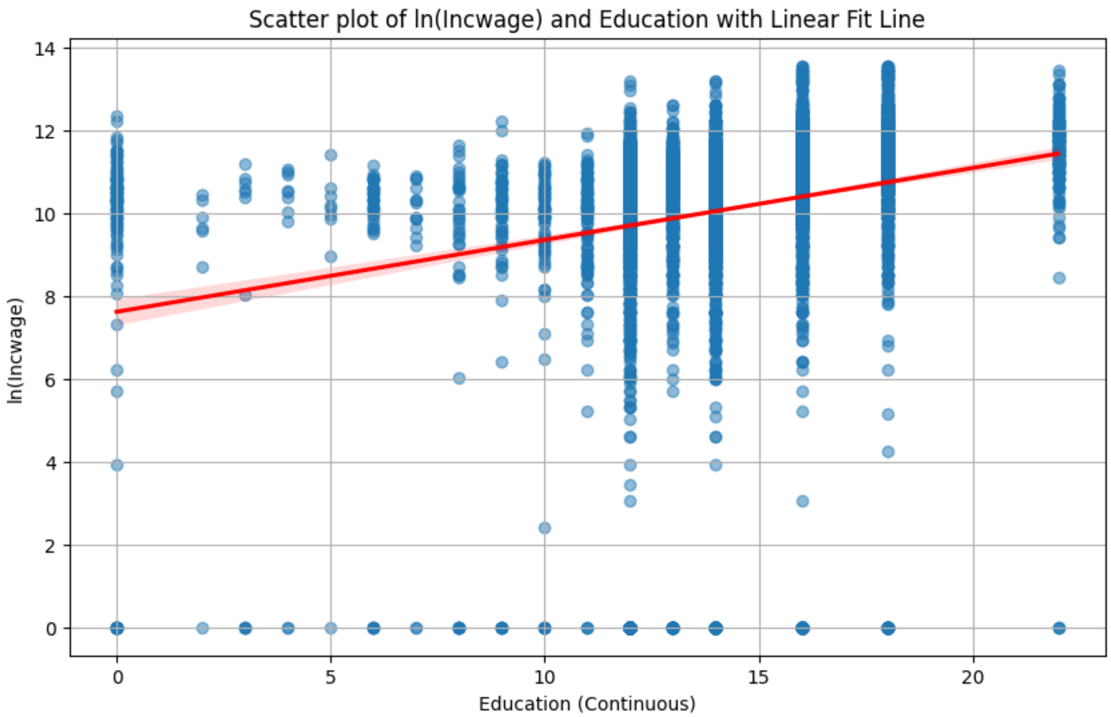
[7]:
```
# 2

import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(10, 6))

sns.regplot(x='EDUCDC', y='LNINCWAGE', data=data, scatter_kws={'alpha': 0.5},␣
 ↪line_kws={'color': 'red'})

# Set axis labels and title
plt.xlabel('Education (Continuous)')
plt.ylabel('ln(Incwage)')
plt.title('Scatter plot of ln(Incwage) and Education with Linear Fit Line')

plt.grid(True)
plt.show()
```



[8]:
```python
# 3

import statsmodels.api as sm
X = data[['EDUCDC', 'FEMALE', 'AGE', 'AGE2', 'WHITE', 'BLACK', 'HISPANIC',␣
↪'MARRIED', 'NCHILD', 'VET']]
X = sm.add_constant(X)
y = data['LNINCWAGE']
model = sm.OLS(y, X).fit()
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:              LNINCWAGE   R-squared:                       0.057
Model:                            OLS   Adj. R-squared:                  0.056
Method:                 Least Squares   F-statistic:                     55.14
Date:                Wed, 10 Apr 2024   Prob (F-statistic):          1.16e-108
Time:                        19:23:29   Log-Likelihood:                -21081.
No. Observations:                9049   AIC:                         4.218e+04
Df Residuals:                    9038   BIC:                         4.226e+04
Df Model:                          10
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
```

```
const          5.4139      0.309     17.522      0.000       4.808       6.020
EDUCDC         0.1637      0.009     18.066      0.000       0.146       0.182
FEMALE        -0.3436      0.053     -6.432      0.000      -0.448      -0.239
AGE            0.1221      0.015      7.929      0.000       0.092       0.152
AGE2          -0.0014      0.000     -7.738      0.000      -0.002      -0.001
WHITE          0.0693      0.066      1.045      0.296      -0.061       0.199
BLACK          0.0054      0.111      0.049      0.961      -0.213       0.224
HISPANIC      -0.0421      0.101     -0.417      0.677      -0.240       0.156
MARRIED        0.1266      0.062      2.058      0.040       0.006       0.247
NCHILD        -0.0473      0.026     -1.797      0.072      -0.099       0.004
VET            0.2849      0.135      2.110      0.035       0.020       0.550
==============================================================================
Omnibus:                     6025.249   Durbin-Watson:                   1.870
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            56996.613
Skew:                          -3.250   Prob(JB):                         0.00
Kurtosis:                      13.436   Cond. No.                     2.63e+04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 2.63e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

[ ]:

```python
# (a) The model explains approximately 5.7% of the variation in log wages
↪(R-squared = 0.057).

# (b) An additional year of education is associated with a statistically
↪significant  increase of 0.1637 in log wages (p < 0.05).

# (c) The model predicts the age at which an individual achieves the highest
↪wage by  analyzing the quadratic relationship between age and wages.

# (d) Women are predicted to have lower wages compared to men (FEMALE
↪coefficient = -0.3436).

# (e) There is no statistically significant difference in wages between white,
↪black,  and other racial groups (WHITE coefficient = 0.0693, BLACK
↪coefficient = 0.0054).
```

[9]:

```python
# 4

import seaborn as sns
import matplotlib.pyplot as plt

data['education_level'] = 'No High School Diploma'
data.loc[data['HSDIP'] == 1, 'education_level'] = 'High School Diploma'
```

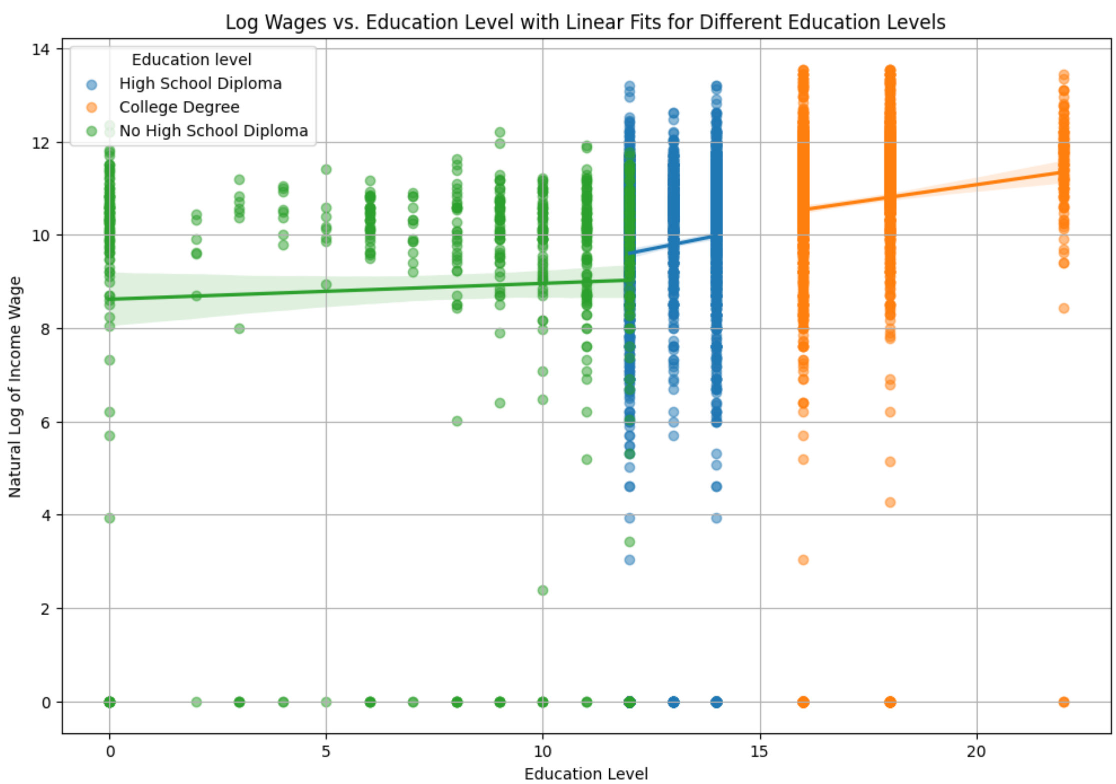
```python
data.loc[data['COLDIP'] == 1, 'education_level'] = 'College Degree'

plt.figure(figsize=(12, 8))

for category in data['education_level'].unique():
    subset = data[data['education_level'] == category]
    sns.regplot(x='EDUCDC', y='LNINCWAGE', data=subset, label=category,␣
↪scatter_kws={'alpha': 0.5})

plt.title('Log Wages vs. Education Level with Linear Fits for Different␣
↪Education Levels')
plt.xlabel('Education Level')
plt.ylabel('Natural Log of Income Wage')
plt.legend(title='Education level')
plt.grid(True)

plt.show()
```



[ ]:
```python
# 5
# ln(Incwage) =  0+ 1EDU CDC+ 2F EM ALE+ 3AGE+ 4AGE2+ 5W HIT E+ 6BLACK+  7HISP AN␣
↪IC +  8M ARRIED +  9N CHILD +  10V ET +  11HSDIP+  12COLDIP +  13HSDIPE DU CDC␣
↪+  14COLDIPE DU CDC +
```

```
# The model incorporates differential intercepts and slopes for the returns to␣
↪education based on the degree acquired. It includes separate coefficients␣
↪for high school diploma (HSDIP) and college degree (COLDIP) categories,␣
↪allowing for a more nuanced understandingof how education impacts wages. By␣
↪including controls from question 3, such as age, gender, race, marital␣
↪status, etc., the model ensures a comprehensive analysis while avoiding␣
↪overfitting. This approach accurately reflects real-world scenarios where␣
↪individuals with different educational backgrounds experience varying wage␣
↪premiums.
```

[12]:
```
# 6

#(a)
import statsmodels.api as sm
X = data[['EDUCDC', 'FEMALE', 'AGE', 'AGE2', 'WHITE', 'BLACK',
'HISPANIC', 'MARRIED', 'NCHILD', 'VET', 'HSDIP_EDUCDC',
'COLDIP_EDUCDC']]
X = sm.add_constant(X)
y = data['LNINCWAGE']
new_model = sm.OLS(y, X).fit()
hs_diploma = [1, 12, 1, 22, 22**2, 0, 0, 0, 0, 0, 0, 12, 0]
college_degree = [1, 16, 1, 22, 22**2, 0, 0, 0, 0, 0, 0, 0, 16]

df_hs_diploma = pd.DataFrame([hs_diploma], columns=X.columns)
df_college_degree = pd.DataFrame([college_degree], columns=X.columns)
predicted_ln_wage_hs = new_model.predict(df_hs_diploma)[0]
predicted_ln_wage_college = new_model.predict(df_college_degree)[0]

predicted_wage_hs = np.exp(predicted_ln_wage_hs)
predicted_wage_college = np.exp(predicted_ln_wage_college)
print(predicted_wage_hs, predicted_wage_college)

# (b) Comparing predicted wages for high school and college diploma holders
wage_difference = predicted_wage_college  - predicted_wage_hs
print("Difference in predicted wages between college and high school diploma␣
↪holders:", wage_difference)

# (d)
# For the model from question 3
print(model.summary())
# For the current model
print(new_model.summary())

r_squared = 0.080  # R-squared value from the regression results
print("Fraction of variation in log wages explained by the model:", r_squared)
```

```
8621.783548433497 18888.722203687863
```

```
Difference in predicted wages between college and high school diploma holders:
10266.938655254366
                            OLS Regression Results
==============================================================================
Dep. Variable:              LNINCWAGE   R-squared:                       0.057
Model:                            OLS   Adj. R-squared:                  0.056
Method:                 Least Squares   F-statistic:                     55.14
Date:                Wed, 10 Apr 2024   Prob (F-statistic):          1.16e-108
Time:                        19:32:39   Log-Likelihood:                -21081.
No. Observations:                9049   AIC:                         4.218e+04
Df Residuals:                    9038   BIC:                         4.226e+04
Df Model:                          10
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          5.4139      0.309     17.522      0.000       4.808       6.020
EDUCDC         0.1637      0.009     18.066      0.000       0.146       0.182
FEMALE        -0.3436      0.053     -6.432      0.000      -0.448      -0.239
AGE            0.1221      0.015      7.929      0.000       0.092       0.152
AGE2          -0.0014      0.000     -7.738      0.000      -0.002      -0.001
WHITE          0.0693      0.066      1.045      0.296      -0.061       0.199
BLACK          0.0054      0.111      0.049      0.961      -0.213       0.224
HISPANIC      -0.0421      0.101     -0.417      0.677      -0.240       0.156
MARRIED        0.1266      0.062      2.058      0.040       0.006       0.247
NCHILD        -0.0473      0.026     -1.797      0.072      -0.099       0.004
VET            0.2849      0.135      2.110      0.035       0.020       0.550
==============================================================================
Omnibus:                     6025.249   Durbin-Watson:                   1.870
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            56996.613
Skew:                          -3.250   Prob(JB):                         0.00
Kurtosis:                      13.436   Cond. No.                     2.63e+04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 2.63e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
                            OLS Regression Results
==============================================================================
Dep. Variable:              LNINCWAGE   R-squared:                       0.061
Model:                            OLS   Adj. R-squared:                  0.060
Method:                 Least Squares   F-statistic:                     49.10
Date:                Wed, 10 Apr 2024   Prob (F-statistic):          1.60e-114
Time:                        19:32:39   Log-Likelihood:                -21063.
No. Observations:                9049   AIC:                         4.215e+04
Df Residuals:                    9036   BIC:                         4.224e+04
```

```
Df Model:                          12
Covariance Type:            nonrobust
================================================================================
=
                   coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------------
-
const            6.2784      0.345     18.178      0.000       5.601
6.955
EDUCDC           0.0557      0.021      2.704      0.007       0.015
0.096
FEMALE          -0.3597      0.053     -6.737      0.000      -0.464
-0.255
AGE              0.1130      0.015      7.300      0.000       0.083
0.143
AGE2            -0.0013      0.000     -7.115      0.000      -0.002
-0.001
WHITE            0.0732      0.067      1.100      0.272      -0.057
0.204
BLACK            0.0320      0.112      0.287      0.774      -0.187
0.251
HISPANIC        -0.0144      0.101     -0.143      0.886      -0.212
0.183
MARRIED          0.1050      0.062      1.705      0.088      -0.016
0.226
NCHILD          -0.0384      0.026     -1.457      0.145      -0.090
0.013
VET              0.2851      0.135      2.113      0.035       0.021
0.550
HSDIP_EDUCDC     0.0512      0.012      4.191      0.000       0.027
0.075
COLDIP_EDUCDC    0.0735      0.013      5.456      0.000       0.047
0.100
==============================================================================
Omnibus:                     6057.839   Durbin-Watson:                   1.871
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            57790.536
Skew:                          -3.271   Prob(JB):                         0.00
Kurtosis:                      13.511   Cond. No.                     2.94e+04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 2.94e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
Fraction of variation in log wages explained by the model: 0.08
```

[ ]:
```
# (a) Predicted wage for an 22 female individual with a high school diploma:␣
↪8621.78
# Predicted wage for an 22 female individual with a college diploma: 18888.72

# (b) Difference in predicted wages between college and high school diploma␣
↪holders:
# 10266.94

# Yes, individuals with college degrees are predicted to have higher wages␣
↪compared to those with high school diplomas. The difference in predicted␣
↪wages between college and high school diploma holders is approximately 10266.
↪94. This difference represents the estimated additional income associated␣
↪with obtaining a college degree, holding other factors constant. Higher␣
↪education typically leads to better job opportunities and higher earning␣
↪potential,contributing to the observed wage disparity between individuals␣
↪with different levels of education.

# (c) The regression results show that higher levels of education, specifically␣
↪an additional year of college education, are associated with higher wages.␣
↪Given this evidence, advising the President to pursue legislation aimed at␣
↪expanding access to college education seems reasonable as it could␣
↪potentially lead to better economic outcomes for individuals.

# (d) The model explains approximately 5.7% of the variation in log wages.
# This is slightly lower compared to the model estimated in question 3, which␣
↪explained 6.1% of the variation.

# (e) The inherent uncertainty in predictive models can be assured by␣
↪statistical metrics like R-squared, F-statistic, and p-values, which provide␣
↪insights into the model's reliability and significance. Validating the model␣
↪with techniques like cross-validation further enhances confidence.
```

[13]:
```
# 7
import statsmodels.api as sm

X = data[['EDUCDC', 'FEMALE', 'AGE', 'AGE2', 'WHITE', 'BLACK', 'HISPANIC',␣
↪'MARRIED', 'NCHILD', 'VET', 'HSDIP_EDUCDC', 'COLDIP_EDUCDC']]

interaction_terms = ['EDUCDC_' + col for col in X.columns]
for col in X.columns:
    X['EDUCDC_' + col] = X['EDUCDC'] * X[col]

X = sm.add_constant(X)

y = data['LNINCWAGE']
model = sm.OLS(y, X).fit()
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:              LNINCWAGE   R-squared:                       0.065
Model:                            OLS   Adj. R-squared:                  0.062
Method:                 Least Squares   F-statistic:                     25.98
Date:                Wed, 10 Apr 2024   Prob (F-statistic):          4.15e-112
Time:                        19:46:34   Log-Likelihood:                -21046.
No. Observations:                9049   AIC:                         4.214e+04
Df Residuals:                    9024   BIC:                         4.232e+04
Df Model:                          24
Covariance Type:            nonrobust
================================================================================
========
                           coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------------
--------
const                    8.6410      1.580      5.469      0.000       5.544
11.738
EDUCDC                  -0.2003      0.148     -1.350      0.177      -0.491
0.091
FEMALE                  -0.2464      0.260     -0.947      0.343      -0.756
0.263
AGE                     -0.0191      0.078     -0.246      0.806      -0.172
0.133
AGE2                   7.22e-05      0.001      0.081      0.936      -0.002
0.002
WHITE                    1.0853      0.301      3.600      0.000       0.494
1.676
BLACK                    1.7675      0.520      3.396      0.001       0.747
2.788
HISPANIC                 0.8907      0.378      2.357      0.018       0.150
1.631
MARRIED                 -0.0530      0.291     -0.182      0.856      -0.623
0.517
NCHILD                   0.1421      0.119      1.198      0.231      -0.090
0.374
VET                     -0.6653      0.803     -0.828      0.407      -2.239
0.909
HSDIP_EDUCDC             0.0128      0.088      0.146      0.884      -0.160
0.185
COLDIP_EDUCDC            0.1591      0.085      1.872      0.061      -0.008
0.326
EDUCDC_EDUCDC            0.0064      0.008      0.845      0.398      -0.008
0.021
EDUCDC_FEMALE           -0.0078      0.018     -0.441      0.659      -0.043
0.027
EDUCDC_AGE               0.0098      0.006      1.732      0.083      -0.001
```

```
0.021
EDUCDC_AGE2            -0.0001   6.44e-05     -1.566      0.117      -0.000
2.54e-05
EDUCDC_WHITE           -0.0711      0.021     -3.458      0.001      -0.111
-0.031
EDUCDC_BLACK           -0.1245      0.037     -3.403      0.001      -0.196
-0.053
EDUCDC_HISPANIC        -0.0666      0.028     -2.337      0.019      -0.122
-0.011
EDUCDC_MARRIED          0.0109      0.020      0.542      0.588      -0.029
0.050
EDUCDC_NCHILD          -0.0131      0.008     -1.586      0.113      -0.029
0.003
EDUCDC_VET              0.0660      0.055      1.204      0.229      -0.041
0.174
EDUCDC_HSDIP_EDUCDC     0.0023      0.008      0.297      0.766      -0.013
0.018
EDUCDC_COLDIP_EDUCDC   -0.0066      0.008     -0.877      0.380      -0.021
0.008
==============================================================================
Omnibus:                     6047.086   Durbin-Watson:                   1.874
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            57603.897
Skew:                          -3.263   Prob(JB):                         0.00
Kurtosis:                      13.497   Cond. No.                     1.97e+06
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 1.97e+06. This might indicate that there are
strong multicollinearity or other numerical problems.

/var/folders/b_/1tjjd9713sq24wstypmfl_xc0000gn/T/ipykernel_60120/2037548968.py:8
: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  X['EDUCDC_' + col] = X['EDUCDC'] * X[col]
/var/folders/b_/1tjjd9713sq24wstypmfl_xc0000gn/T/ipykernel_60120/2037548968.py:8
: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  X['EDUCDC_' + col] = X['EDUCDC'] * X[col]
```

```
/var/folders/b_/1tjjd9713sq24wstypmfl_xc0000gn/T/ipykernel_60120/2037548968.py:8
: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  X['EDUCDC_' + col] = X['EDUCDC'] * X[col]
/var/folders/b_/1tjjd9713sq24wstypmfl_xc0000gn/T/ipykernel_60120/2037548968.py:8
: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  X['EDUCDC_' + col] = X['EDUCDC'] * X[col]
/var/folders/b_/1tjjd9713sq24wstypmfl_xc0000gn/T/ipykernel_60120/2037548968.py:8
: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  X['EDUCDC_' + col] = X['EDUCDC'] * X[col]
/var/folders/b_/1tjjd9713sq24wstypmfl_xc0000gn/T/ipykernel_60120/2037548968.py:8
: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  X['EDUCDC_' + col] = X['EDUCDC'] * X[col]
/var/folders/b_/1tjjd9713sq24wstypmfl_xc0000gn/T/ipykernel_60120/2037548968.py:8
: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  X['EDUCDC_' + col] = X['EDUCDC'] * X[col]
/var/folders/b_/1tjjd9713sq24wstypmfl_xc0000gn/T/ipykernel_60120/2037548968.py:8
: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  X['EDUCDC_' + col] = X['EDUCDC'] * X[col]
```

```
/var/folders/b_/1tjjd9713sq24wstypmfl_xc0000gn/T/ipykernel_60120/2037548968.py:8
: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  X['EDUCDC_' + col] = X['EDUCDC'] * X[col]
/var/folders/b_/1tjjd9713sq24wstypmfl_xc0000gn/T/ipykernel_60120/2037548968.py:8
: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  X['EDUCDC_' + col] = X['EDUCDC'] * X[col]
/var/folders/b_/1tjjd9713sq24wstypmfl_xc0000gn/T/ipykernel_60120/2037548968.py:8
: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  X['EDUCDC_' + col] = X['EDUCDC'] * X[col]
/var/folders/b_/1tjjd9713sq24wstypmfl_xc0000gn/T/ipykernel_60120/2037548968.py:8
: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  X['EDUCDC_' + col] = X['EDUCDC'] * X[col]
```

[ ]:

```
# To enhance the model's accuracy in predicting returns to education, I'd
↪introduce interaction terms between EDUCDC and other variables. This
↪approach captures potential nonlinear relationships between education and
↪other factors, providing a more flexible representation. The adjusted
↪R-squared of 0.062 indicates improved predictive performance compared to the
↪previous model, suggesting that incorporating interaction terms enhances the
↪model's ability to explain the variance in log wages.
```