SIMPLE LINEAR REGRESSION ANALYSIS OF WAGES AND EDUCATION OF YOUNG MALES DATASET

The dataset offers detailed information about people, including characteristics relating to employment and demographics. Year, years of education (school), job experience (exper), and union membership status. Additionally, it contains categorical information such as health status, marital status, and ethnicity (ethn). The wage column, which shows income levels, provides wage-related information. The industry and occupation sections offer information pertaining to a given industry and career. The residence column covers geographic dispersion, identifying places like "north_east."

The dataset investigate the connections between salary levels and education (school year), It examine how incomes are affected by school variable. However, the data also makes it possible to analyze sectors by occupation and industry while looking at regional trends. All things considered, it provides insightful information for socioeconomic research, pay inequality, and labor market trends.

In [161…
```
# packages
import pandas as pd
import numpy as np
import seaborn as sns
import datetime
from matplotlib import pyplot as plt
from statsmodels.formula.api import ols
import statsmodels.api as sm
import statsmodels.formula.api as smf
```

In [153…
```
df = pd.read_csv('Males.csv')
```

In [59]:
```
df.head(6)
```

Out[59]:

	rownames	nr	year	school	exper	union	ethn	maried	health	wage	
0	1	13	1980	14	1	no	other	no	no	1.197540	Busines
1	2	13	1981	14	2	yes	other	no	no	1.853060	
2	3	13	1982	14	3	no	other	no	no	1.344462	Busines
3	4	13	1983	14	4	no	other	no	no	1.433213	Busines
4	5	13	1984	14	5	no	other	no	no	1.568125	
5	6	13	1985	14	6	no	other	no	no	1.699891	Busines

In [61]:
```
df.describe()
```

Out[61]:

	rownames	nr	year	school	exper	wage
count	4360.000000	4360.000000	4360.000000	4360.000000	4360.000000	4360.000000
mean	2180.500000	5262.058716	1983.500000	11.766972	6.514679	1.649147
std	1258.767916	3496.149815	2.291551	1.746181	2.825873	0.532609
min	1.000000	13.000000	1980.000000	3.000000	0.000000	-3.579079
25%	1090.750000	2329.000000	1981.750000	11.000000	4.000000	1.350717
50%	2180.500000	4569.000000	1983.500000	12.000000	6.000000	1.671143
75%	3270.250000	8406.000000	1985.250000	12.000000	9.000000	1.991086
max	4360.000000	12548.000000	1987.000000	16.000000	18.000000	4.051860

In [63]:
```
df["school"].describe()
```

Out[63]:
```
count    4360.000000
mean       11.766972
std         1.746181
min         3.000000
25%        11.000000
50%        12.000000
75%        12.000000
max        16.000000
Name: school, dtype: float64
```

In [65]:
```
df['wage'].mean()
```

Out[65]:
```
1.6491471906705277
```

In [67]:
```
df['wage'].std()
```

Out[67]:
```
0.5326094063484761
```

In [69]:
```
df['wage'].describe()
```

Out[69]:
```
count    4360.000000
mean        1.649147
std         0.532609
min        -3.579079
25%         1.350717
50%         1.671143
75%         1.991086
max         4.051860
Name: wage, dtype: float64
```

In [71]:
```
df['school'].max()
```

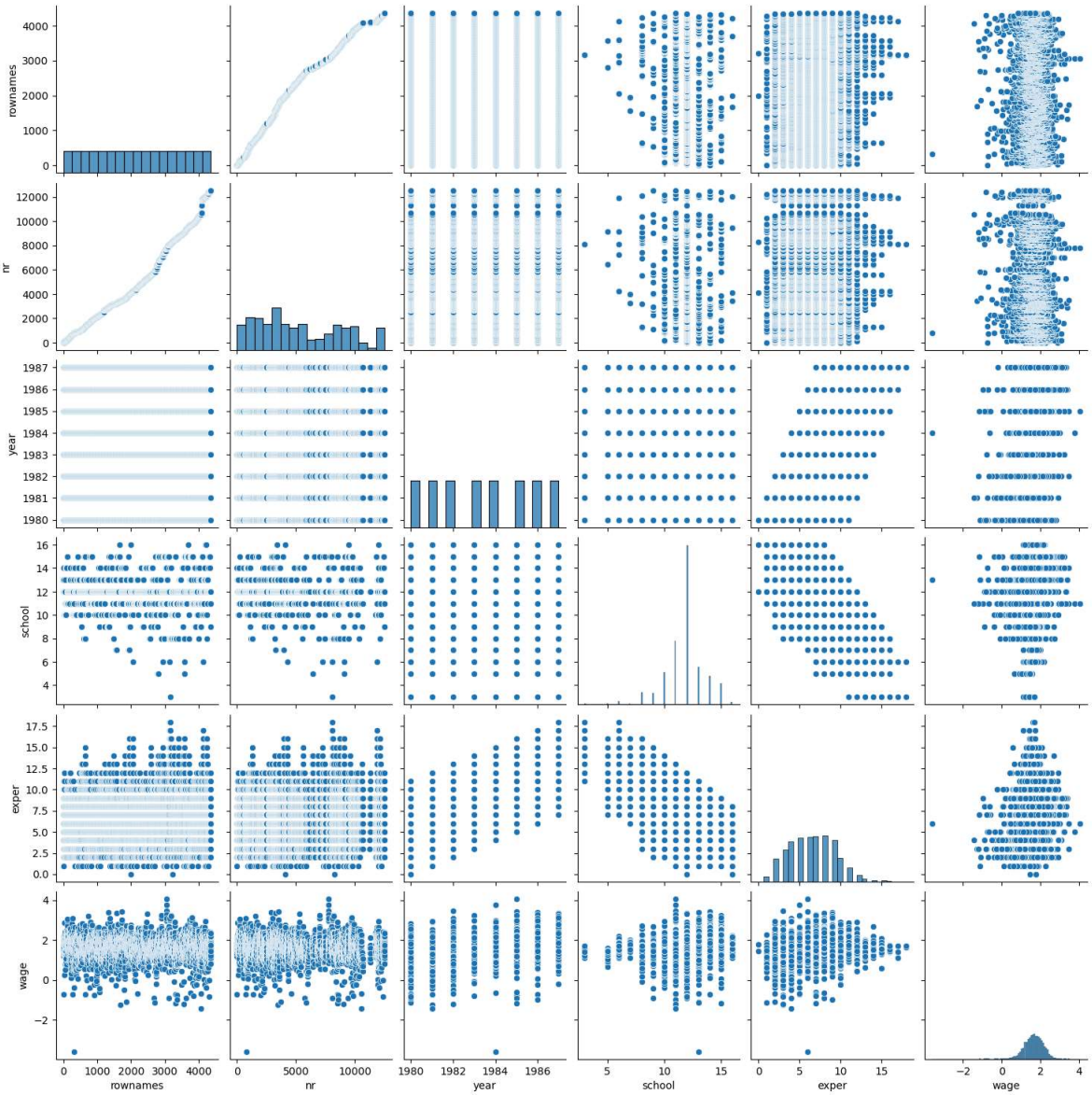
Out[71]:
```
16
```

In [47]:
```
#TOTAL COUNT OF MISSING VALUES
print('Total count of missing values:', df.isna().sum())

#checkout for duplicate
print('shape of data frame:',df.shape)
```

```
Total count of missing values: rownames         0
nr                0
year              0
school            0
exper             0
union             0
ethn              0
maried            0
health            0
wage              0
industry          0
occupation        0
residence      1245
dtype: int64
shape of data frame: (4360, 13)
```

In [50]:
```
sns.pairplot(df)
```

Out[50]: <seaborn.axisgrid.PairGrid at 0x182562dfef0>



In [109…
```
#COLUMS FOR ANALYSIS
ols_data = df[['school','wage']]
print(ols_data)
```

```
      school      wage
0         14  1.197540
1         14  1.853060
2         14  1.344462
3         14  1.433213
4         14  1.568125
...      ...       ...
4355       9  1.591879
4356       9  1.212543
4357       9  1.765962
4358       9  1.745894
4359       9  1.466543

[4360 rows x 2 columns]
```

In [157…

```
# RESULTS AND EVALUATION
# summary of the results from the model
model = smf.ols("wage ~ school", data=ols_data)
result = model.fit()
print(result.summary())
```

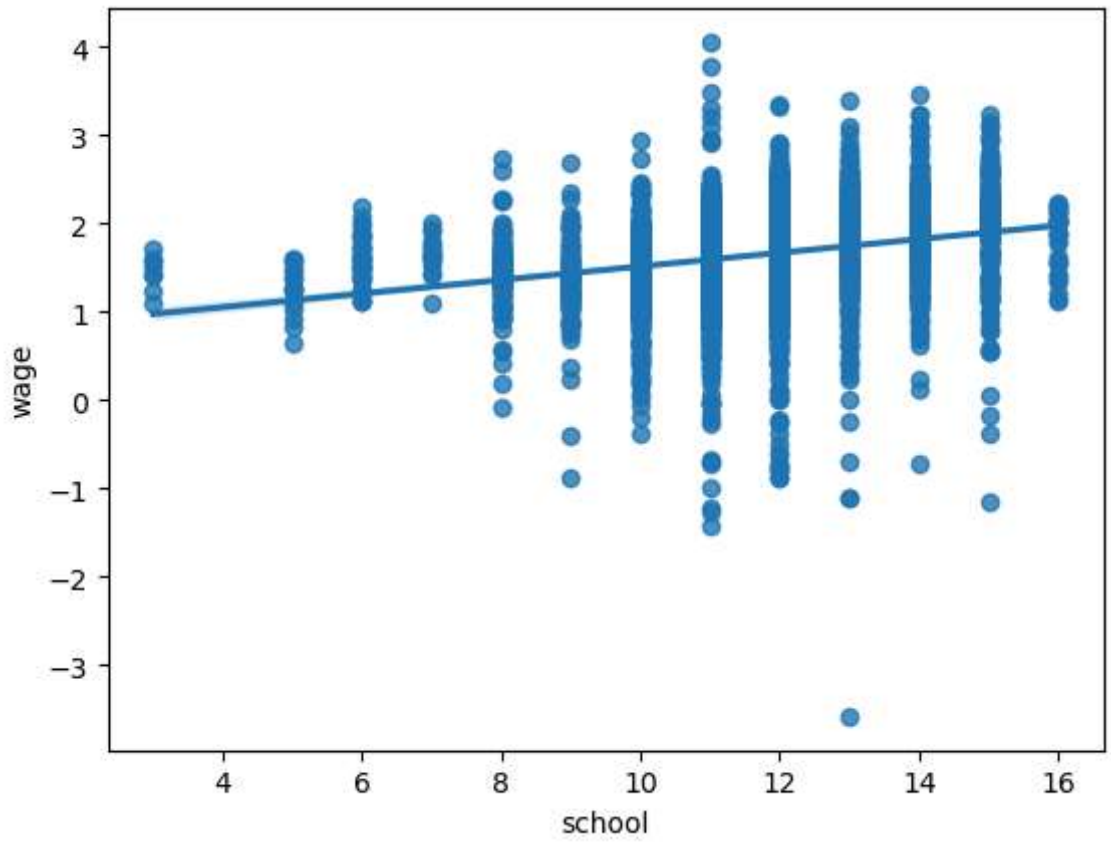
```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   wage   R-squared:                       0.064
Model:                            OLS   Adj. R-squared:                  0.063
Method:                 Least Squares   F-statistic:                     295.8
Date:                Fri, 15 Nov 2024   Prob (F-statistic):           3.34e-64
Time:                        01:20:31   Log-Likelihood:                -3296.2
No. Observations:                4360   AIC:                             6596.
Df Residuals:                    4358   BIC:                             6609.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      0.7442      0.053     13.993      0.000       0.640       0.848
school         0.0769      0.004     17.200      0.000       0.068       0.086
==============================================================================
Omnibus:                     1165.271   Durbin-Watson:                   0.993
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             7453.017
Skew:                          -1.117   Prob(JB):                         0.00
Kurtosis:                       9.003   Cond. No.                         81.6
==============================================================================

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

In [110…

```
# OLS DATA WITH BEST FIT REGRESSION LINE
sns.regplot(x= "school", y="wage", data=ols_data)
```

Out[110…

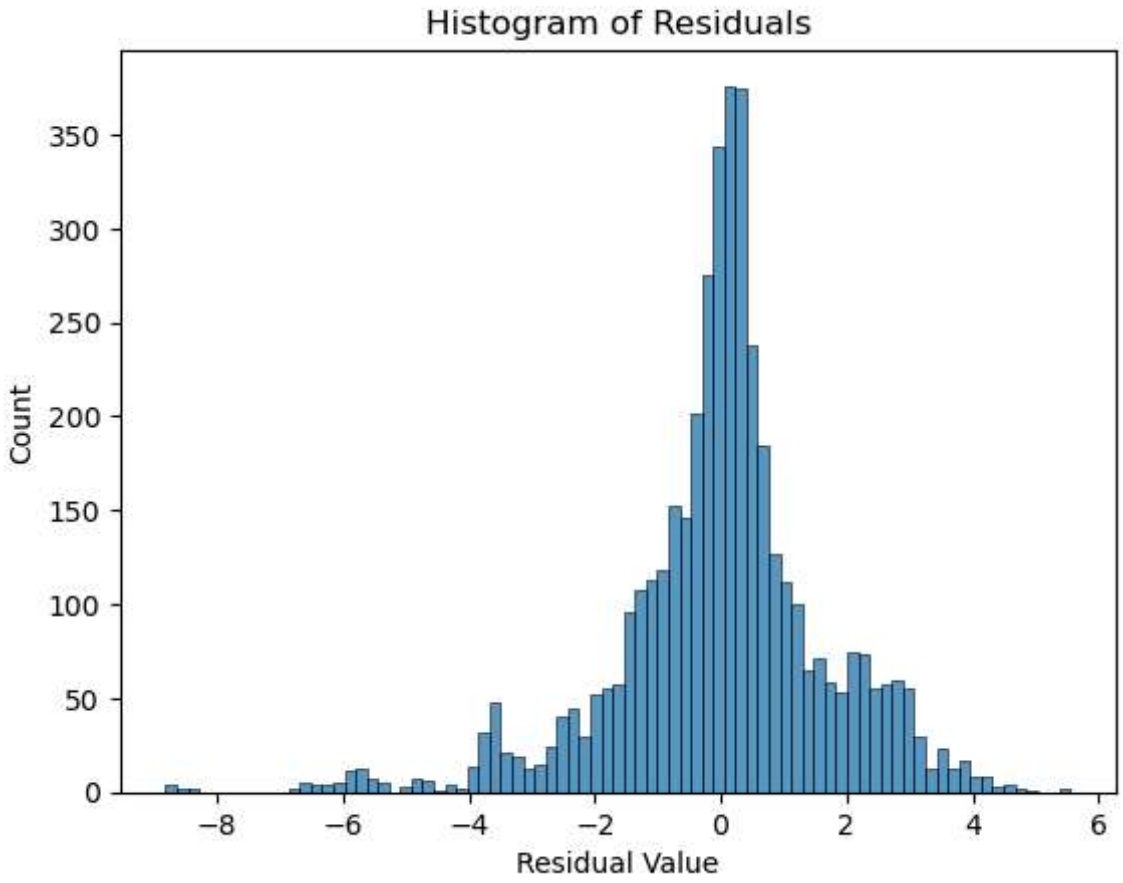
```
<Axes: xlabel='school', ylabel='wage'>
```



In [123…

```
#VISUALISATION OF THE DISTRIBUTIOMN OF RESIDUALS
# Normality Assumption
model = smf.ols("school ~ wage", data=ols_data).fit()
residuals = model.resid
print(residuals)

# Visualization of residuals
fig = sns.histplot(residuals)  # Added KDE for a smoother curve
fig.set_xlabel("Residual Value")
fig.set_title("Histogram of Residuals")
plt.show()
```

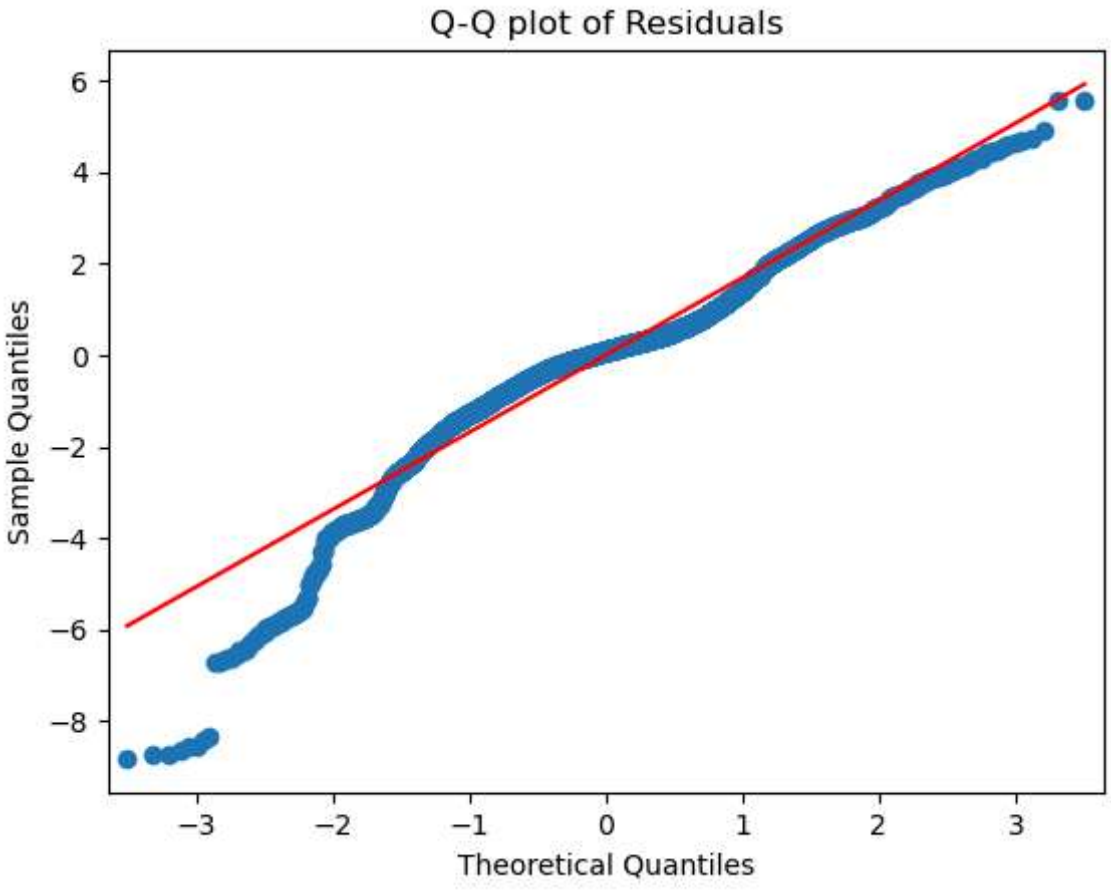
```
0       2.606338
1       2.064468
2       2.484888
3       2.411524
4       2.300003
          ...
4355   -2.719633
4356   -2.406064
4357   -2.863534
4358   -2.846946
4359   -2.616027
Length: 4360, dtype: float64
```



In [212…
```
import statsmodels.api as sm

model = sm.OLS(df['school'], df['wage']).fit()

# the residuals code
residuals = model.resid

# Print the residuals
print(residuals)
```

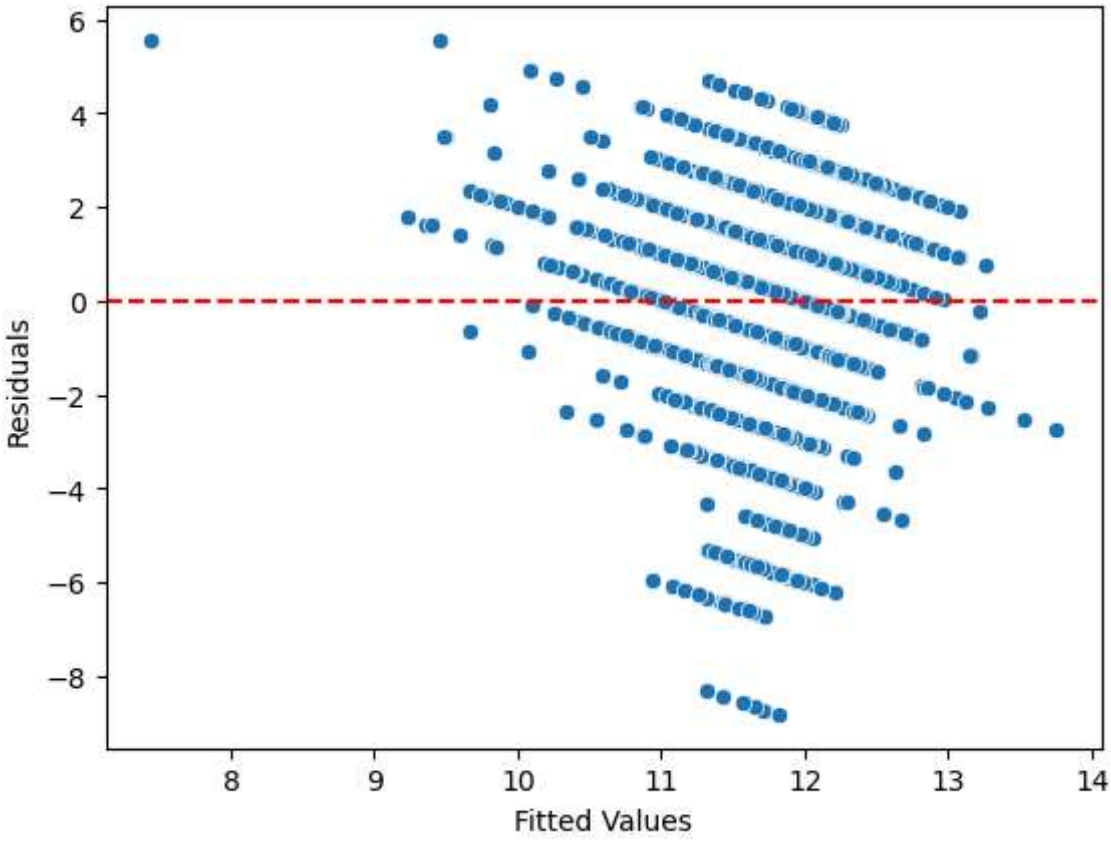
```
0       6.168740
1       1.881998
2       5.207953
3       4.627566
4       3.745317
          ...
4355   -1.410019
4356    1.070631
4357   -2.548428
4358   -2.417197
4359   -0.590391
Length: 4360, dtype: float64
```

In [159…
```
# OBSERVATION AND VISUALISATION OF RESIDUAL DISTRIBUTION CODE
sm.qqplot(residuals, line ='s')
plt.title("Q-Q plot of Residuals")
plt.show()
```



In [145…

```
# Fit the model using statsmodels' formula API (ols function)
model = smf.ols("school ~ wage", data=ols_data).fit()

# the residuals code
residuals = model.resid

# the fitted (predicted) values code
fitted_values = model.fittedvalues

# scatter plot of fitted values vs residuals code
fig = sns.scatterplot(x=fitted_values, y=residuals)

# code of the horizontal line at 0 to assess residuals around zero
fig.axhline(0, color='red', linestyle='--')

# Labeling the axes code
fig.set_xlabel("Fitted Values")
fig.set_ylabel("Residuals")

# Show the plot
plt.show()
```



KEY TAKE AWAY FROM THE DATA The results of the OLS regression show the correlation between earnings and years of education. With an R-squared value of 0.064, the model only partially explains the salary variance, suggesting that education by itself has little predictive potential. With p-values less than 0.001, the intercept (0.7442) and the school coefficient (0.0769) are both statistically significant, indicating a consistent positive correlation between education and income. In particular, incomes rise by an average of 0.0769 units for every extra year of education. The model's overall significance is supported by the F-statistic (295.8) and the p-value that goes with it. Notwithstanding the importance, the low R-squared suggests that factors other than education probably affect salaries. The diagnostics reveal a high kurtosis (9.003) and some skewness (-1.117), suggesting that the residuals may not be normal. The explanatory power of the model may be improved by further investigation including extra predictors or interaction factors. CONCLUSIVELY P-values: Both the school coefficient and the intercept have extremely low p-values (less than 0.001), suggesting that they are both statistically significant. This indicates that we can be sure that there is a reason why wages and education are related.

Data source: https://www.kaggle.com/datasets/jacopoferretti/wages-and-education-of-young-males-dataset