In [1]:
```python
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from scipy.stats import chi2_contingency, ttest_ind
import numpy as np
import statsmodels.formula.api as smf
import statsmodels.api as sm
from statsmodels.formula.api import ols
```

In [2]:
```python
# 1. Data preparation functions

def download_data():
    import kaggle
    kaggle.api.dataset_download_files('shashwatwork/dementia-prediction-dataset', path='./', unzip=True)

def load_and_preprocess_data():
    df = pd.read_csv('dementia_dataset.csv')
    df = df.drop_duplicates()
    df['SES'] = df['SES'].fillna(df['SES'].median())
    df['MMSE'] = df['MMSE'].fillna(df['MMSE'].median())
    df['M/F'] = df['M/F'].str.upper()
    df['Group'] = df['CDR'].apply(lambda x: 'Nondemented' if x == 0 else 'Demented')
    df['Dementia'] = (df['Group'] == 'Demented').astype(int)
    return df
```

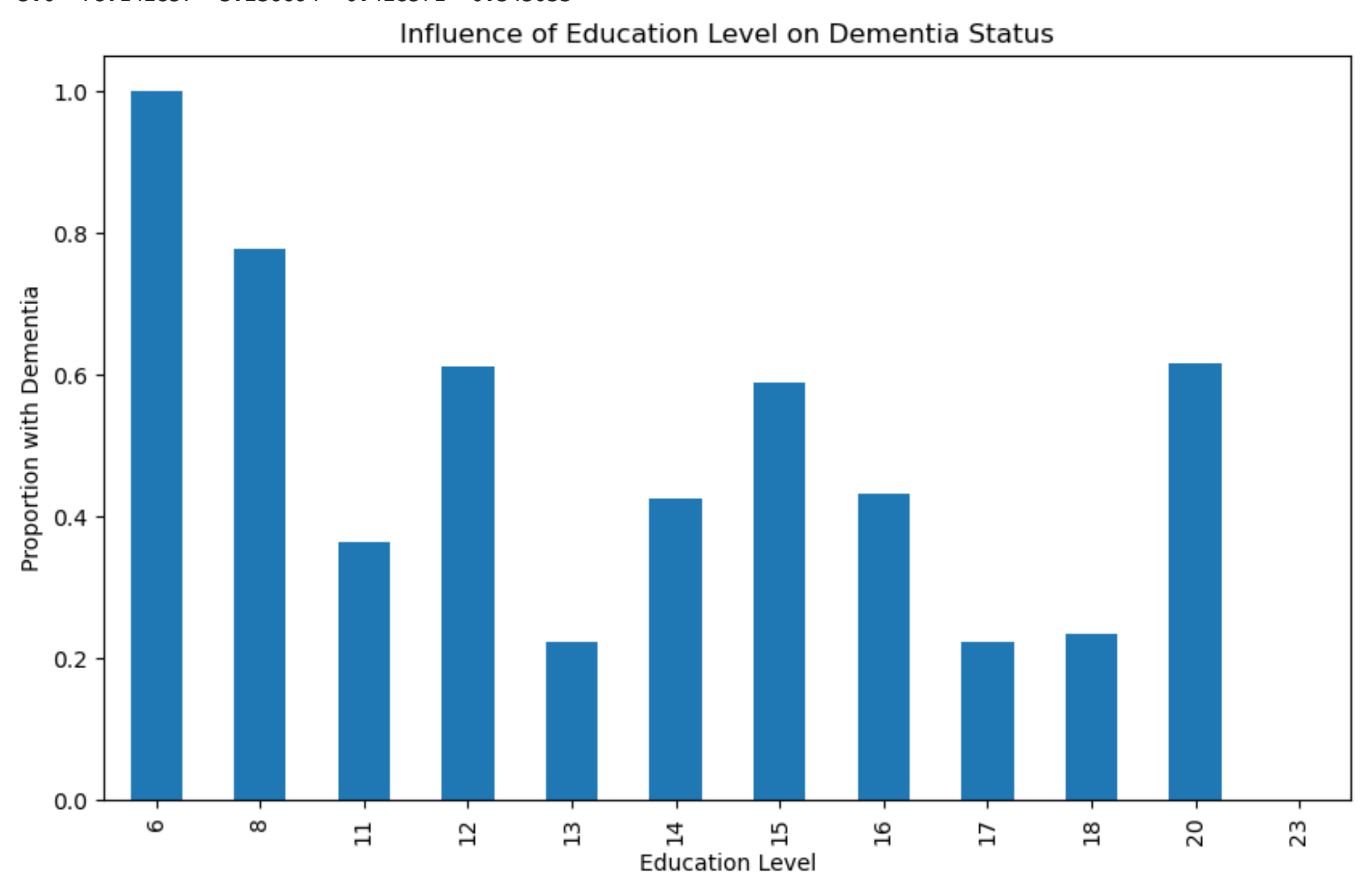
In [3]:
```python
# 2. Summary stats function

def compute_summary_stats(df):
    summary_stats = df.groupby('Age')[['SES', 'Dementia']].agg(['mean', 'std'])
    summary_stats_ses = df.groupby('SES')[['Age', 'CDR']].agg(['mean', 'std'])
    return summary_stats, summary_stats_ses
```

In [4]:
```python
# 3. Visualization functions

def plot_dementia_education(df):
    dementia_education = df.groupby('EDUC')['Dementia'].mean()
    plt.figure(figsize=(10,6))
    dementia_education.plot(kind='bar')
    plt.title("Influence of Education Level on Dementia Status")
    plt.xlabel('Education Level')
    plt.ylabel('Proportion with Dementia')
    plt.show()

def plot_dementia_education_regression(df):
    dementia_education = df.groupby('EDUC')['Dementia'].mean()
    plt.figure(figsize=(10,6))
    sns.regplot(x=dementia_education.index, y=dementia_education.values)
    plt.title("Influence of Education Level on Dementia Status with Regression")
    plt.xlabel('Education Level')
    plt.ylabel('Proportion with Dementia')
    plt.show()

def plot_age_ses_distribution(df):
    plt.figure(figsize=(10,6))
    sns.scatterplot(data=df, x='Age', y='SES', hue='Dementia')
    plt.title("Distribution of Age and SES based on Dementia Status")
    plt.xlabel('Age')
    plt.ylabel('SES')
    plt.show()

def plot_MMSE_distribution(df):
    plt.figure(figsize=(10,6))
    sns.boxplot(data=df, x='Dementia', y='MMSE', hue='M/F')
    plt.title("MMSE Score Distribution based on Gender and Dementia Status")
    plt.show()
```

In [5]:
```python
# 4. Statistical testing functions

def perform_ols_regression(df):
    model = smf.ols(formula='Dementia ~ EDUC', data=df)
    results = model.fit()
    return results.summary()

def logistic_regression(df):
    df['intercept'] = 1
    X = df[['intercept', 'EDUC']]
    y = df['Dementia']
    logit_model = sm.Logit(y, X)
    result = logit_model.fit()
    return result.summary()

def chi_squared_test(df):
    crosstab = pd.crosstab(df['EDUC'], df['Dementia'])
    chi2, p, dof, expected = chi2_contingency(crosstab)
    return chi2, p

def correlation_analysis(df):
    correlation_matrix = df[['EDUC', 'Dementia']].corr()
    correlation = correlation_matrix.loc['EDUC', 'Dementia']
    return correlation

def multivariate_regression(df):
    model = smf.ols(formula='Dementia ~ Age + SES', data=df)
    results = model.fit()
    return results.summary()

def two_way_ANOVA(df):
    df = df.rename(columns={'M/F': 'Gender'})
    model = ols('MMSE ~ C(Gender) + C(Dementia) + C(Gender):C(Dementia)', data=df).fit()
    anova_table = sm.stats.anova_lm(model, typ=2)
    return anova_table

def t_test(df):
    male_MMSE = df[df['M/F'] == 'M']['MMSE']
    female_MMSE = df[df['M/F'] == 'F']['MMSE']
    t_stat, p_val = ttest_ind(male_MMSE, female_MMSE)
    return t_stat, p_val
```

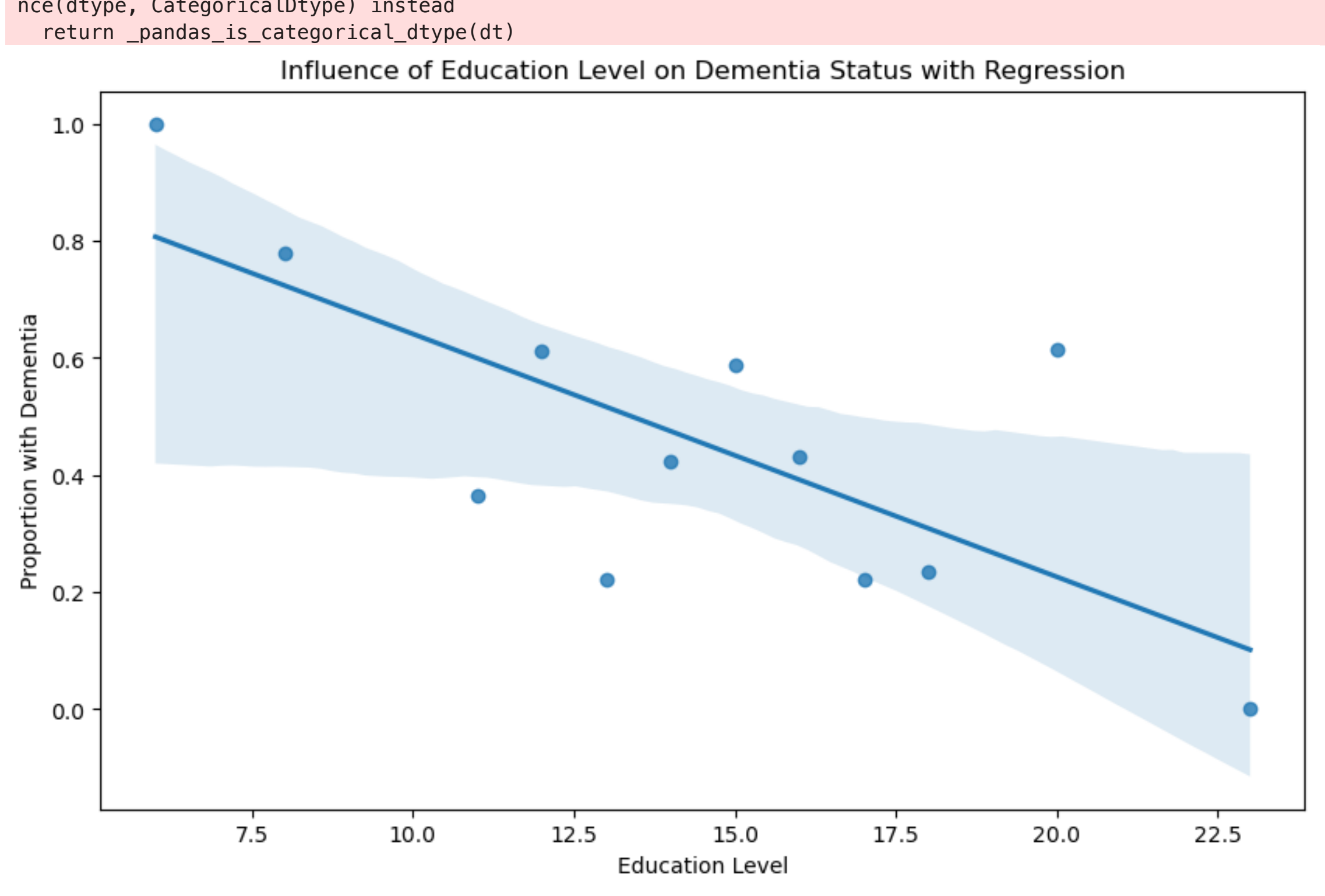
In [6]:
```python
# Now let's run all the functions:

download_data()
df = load_and_preprocess_data()
summary1, summary2 = compute_summary_stats(df)
print(summary1)
print(summary2)

plot_dementia_education(df)
print(perform_ols_regression(df))
plot_dementia_education_regression(df)
print(logistic_regression(df))
chi2, p = chi_squared_test(df)
print(f"Chi-Squared: {chi2}\nP-value: {p}")
correlation = correlation_analysis(df)
print(f"Correlation: {correlation}")
print(multivariate_regression(df))
print(two_way_ANOVA(df))
t_stat, p_val = t_test(df)
print(f"T-statistic: {t_stat}\nP-value: {p_val}")
plot_age_ses_distribution(df)
plot_MMSE_distribution(df)
```

```
          SES            Dementia
         mean       std      mean       std
Age
60   2.500000  2.121320  0.000000  0.000000
61   1.750000  0.957427  0.250000  0.500000
62   2.500000  1.732051  0.500000  0.577350
63   1.666667  0.577350  0.000000  0.000000
64   3.000000  1.000000  0.333333  0.577350
65   2.500000  0.836660  0.333333  0.516398
66   2.100000  0.875595  0.600000  0.516398
67   3.000000  1.264911  0.333333  0.516398
68   2.214286  1.050902  0.500000  0.518875
69   2.538462  1.050031  0.384615  0.506370
70   2.916667  1.083625  0.500000  0.522233
71   2.666667  1.188177  0.444444  0.511310
72   2.636364  1.286291  0.636364  0.504525
73   2.846154  1.046606  0.500000  0.509902
74   2.166667  0.834847  0.250000  0.452267
75   2.363636  1.135801  0.636364  0.492366
76   2.437500  0.813941  0.562500  0.512348
77   2.125000  1.087811  0.250000  0.447214
78   2.761905  1.135991  0.476190  0.511766
79   1.272727  0.646670  0.545455  0.522233
80   2.400000  1.273206  0.400000  0.502625
81   2.388889  1.144752  0.500000  0.514496
82   2.529412  1.328422  0.529412  0.514496
83   2.333333  0.778499  0.416667  0.514929
84   2.230769  1.091928  0.692308  0.480384
85   2.555556  1.236033  0.222222  0.440959
86   2.700000  1.251666  0.500000  0.527046
87   2.333333  1.366260  0.333333  0.516398
88   3.100000  0.994429  0.300000  0.483046
89   2.571429  1.133893  0.285714  0.487950
90   3.000000  0.707107  0.400000  0.547723
91   2.250000  0.957427  0.250000  0.500000
92   1.500000  1.000000  0.500000  0.577350
93   2.000000  0.000000  0.000000  0.000000
94   1.000000       NaN  0.000000       NaN
95   2.000000       NaN  0.000000       NaN
96   1.000000       NaN  1.000000       NaN
97   1.000000       NaN  0.000000       NaN
98   1.000000       NaN  1.000000       NaN
           Age                 CDR
          mean       std      mean       std
SES
1.0  77.829545  8.472025  0.301136  0.425750
2.0  76.672131  7.418363  0.254098  0.353529
3.0  77.243902  7.264935  0.280488  0.369330
4.0  76.243243  7.717283  0.337838  0.352176
5.0  78.142857  3.236694  0.428571  0.345033
```



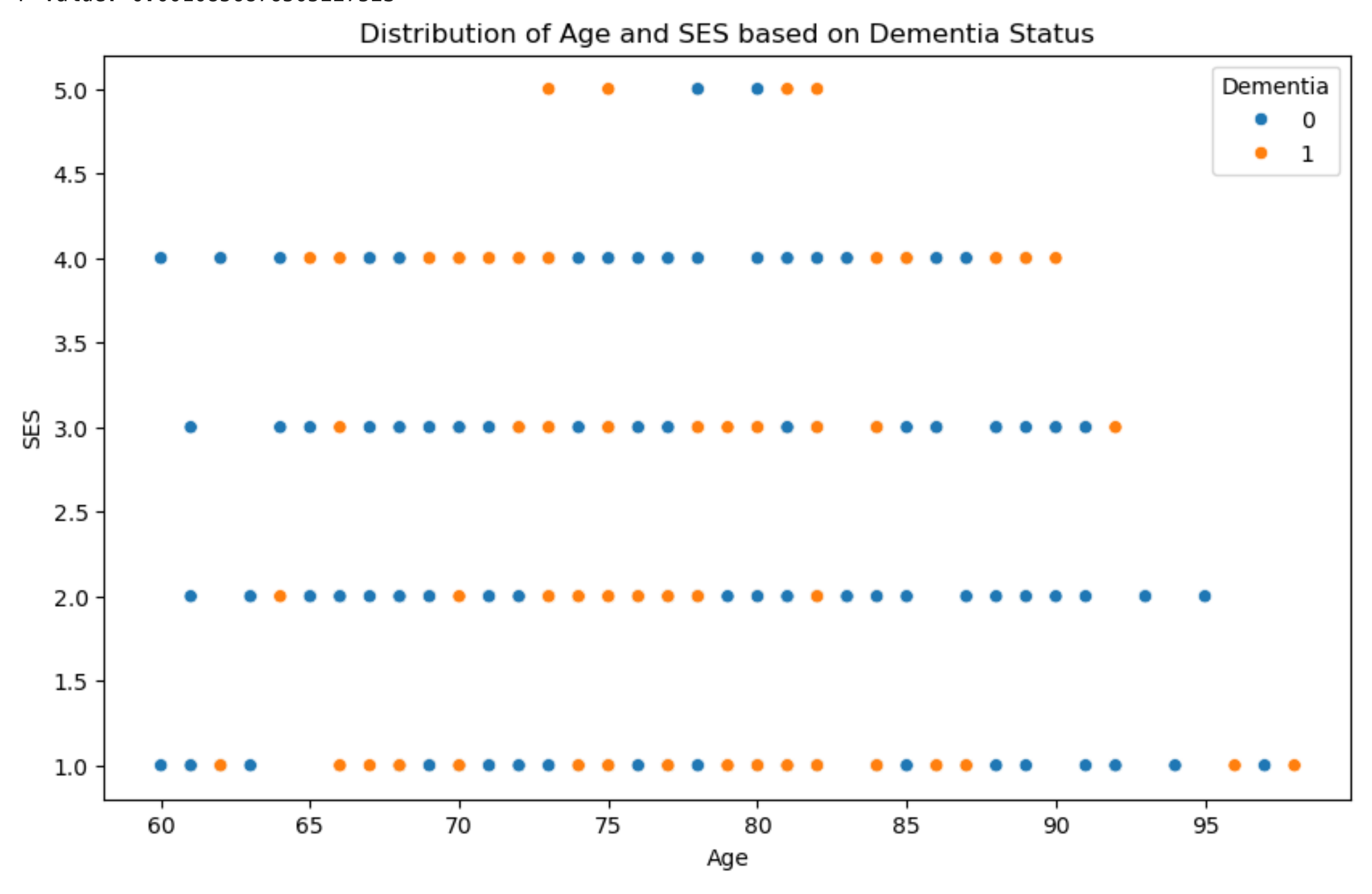

```
                            OLS Regression Results
==============================================================================
Dep. Variable:               Dementia   R-squared:                       0.047
Model:                            OLS   Adj. R-squared:                  0.045
Method:                 Least Squares   F-statistic:                     18.41
Date:                Wed, 08 Nov 2023   Prob (F-statistic):           2.28e-05
Time:                        11:42:25   Log-Likelihood:                -259.64
No. Observations:                 373   AIC:                             523.3
Df Residuals:                     371   BIC:                             531.1
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      0.9972      0.131      7.640      0.000       0.741       1.254
EDUC          -0.0376      0.009     -4.291      0.000      -0.055      -0.020
==============================================================================
Omnibus:                     1997.964   Durbin-Watson:                   1.081
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               51.583
Skew:                           0.207   Prob(JB):                     6.29e-12
Kurtosis:                       1.226   Cond. No.                         77.4
==============================================================================

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

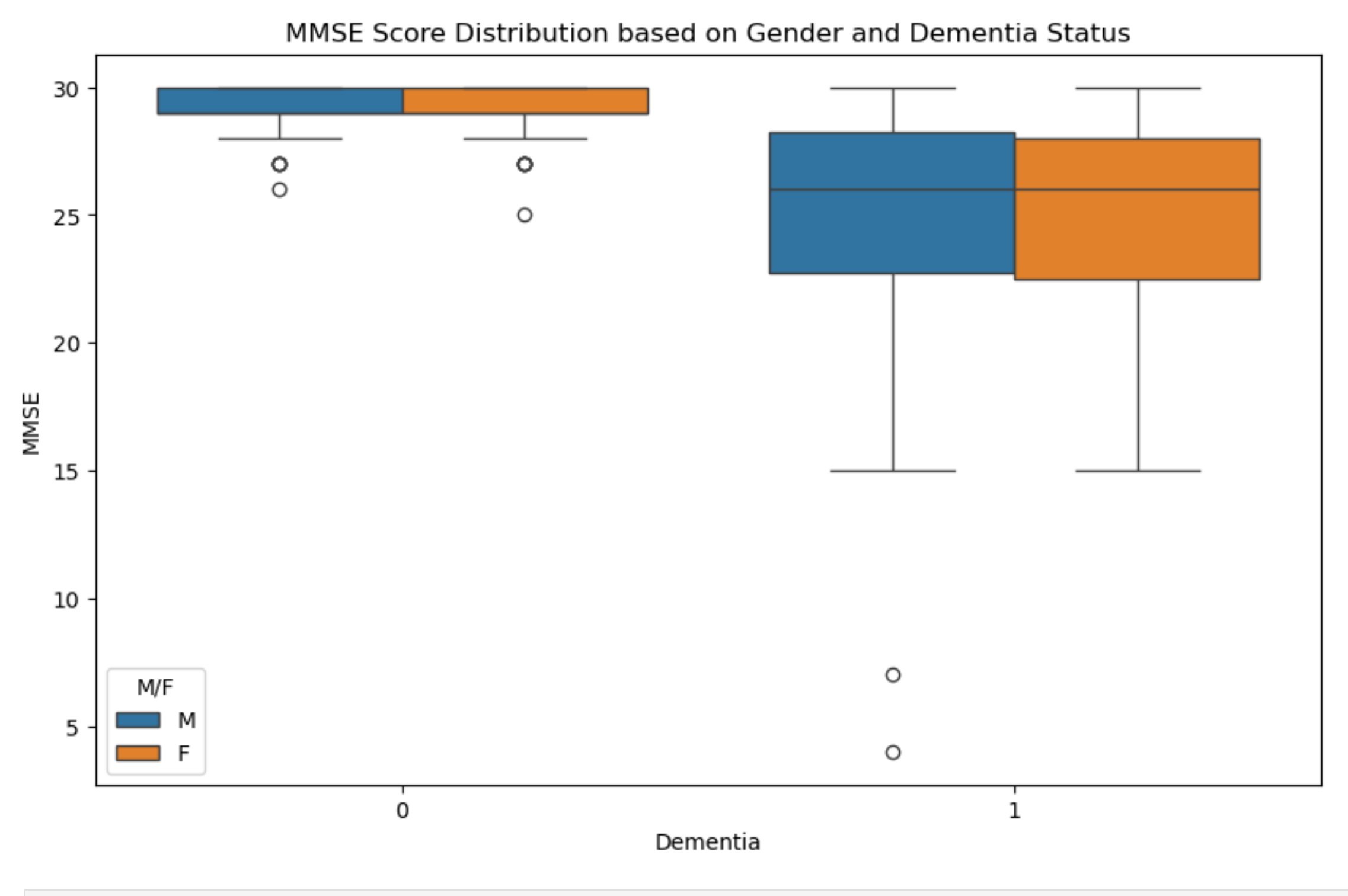
```
/Users/benjheindl/anaconda3/lib/python3.10/site-packages/patsy/util.py:672: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  return _pandas_is_categorical_dtype(dt)
/Users/benjheindl/anaconda3/lib/python3.10/site-packages/patsy/util.py:672: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  return _pandas_is_categorical_dtype(dt)
/Users/benjheindl/anaconda3/lib/python3.10/site-packages/patsy/util.py:672: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  return _pandas_is_categorical_dtype(dt)
```




```
/Users/benjheindl/anaconda3/lib/python3.10/site-packages/patsy/util.py:672: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  return _pandas_is_categorical_dtype(dt)
/Users/benjheindl/anaconda3/lib/python3.10/site-packages/patsy/util.py:672: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  return _pandas_is_categorical_dtype(dt)
/Users/benjheindl/anaconda3/lib/python3.10/site-packages/patsy/util.py:672: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  return _pandas_is_categorical_dtype(dt)
/Users/benjheindl/anaconda3/lib/python3.10/site-packages/patsy/util.py:672: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  return _pandas_is_categorical_dtype(dt)
/Users/benjheindl/anaconda3/lib/python3.10/site-packages/patsy/util.py:672: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  return _pandas_is_categorical_dtype(dt)
/Users/benjheindl/anaconda3/lib/python3.10/site-packages/patsy/util.py:672: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  return _pandas_is_categorical_dtype(dt)
```

```
Optimization terminated successfully.
         Current function value: 0.663472
         Iterations 5
                           Logit Regression Results
==============================================================================
Dep. Variable:               Dementia   No. Observations:                  373
Model:                          Logit   Df Residuals:                      371
Method:                           MLE   Df Model:                            1
Date:                Wed, 08 Nov 2023   Pseudo R-squ.:                 0.03519
Time:                        11:42:25   Log-Likelihood:                -247.48
converged:                       True   LL-Null:                       -256.50
Covariance Type:            nonrobust   LLR p-value:                 2.149e-05
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
intercept      2.1093      0.573      3.684      0.000       0.987       3.232
EDUC          -0.1596      0.039     -4.105      0.000      -0.236      -0.083
==============================================================================
Chi-Squared: 43.77947781881186
P-value: 7.94175383289061e-06
Correlation: -0.21742843686620875
                            OLS Regression Results
==============================================================================
Dep. Variable:               Dementia   R-squared:                       0.011
Model:                            OLS   Adj. R-squared:                  0.005
Method:                 Least Squares   F-statistic:                     2.005
Date:                Wed, 08 Nov 2023   Prob (F-statistic):              0.136
Time:                        11:42:25   Log-Likelihood:                -266.66
No. Observations:                 373   AIC:                             539.3
Df Residuals:                     370   BIC:                             551.1
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      0.4171      0.270      1.547      0.123      -0.113       0.947
Age           -0.0010      0.003     -0.310      0.757      -0.008       0.006
SES            0.0456      0.023      1.962      0.050   -9.24e-05       0.091
==============================================================================
Omnibus:                     1752.814   Durbin-Watson:                   1.075
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               59.588
Skew:                           0.209   Prob(JB):                     1.15e-13
Kurtosis:                       1.087   Cond. No.                         812.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
                            sum_sq     df           F        PR(>F)
C(Gender)                 1.785740    1.0    0.193212  6.605139e-01
C(Dementia)            1471.621773    1.0  159.225337  1.355442e-30
C(Gender):C(Dementia)     0.209154    1.0    0.022630  8.805057e-01
Residual               3410.439847  369.0         NaN           NaN
T-statistic: -3.293216186741949
P-value: 0.0010856876505227523
```





In [ ]: