HW 5

Kevin Lin

2/13/2026

1

We are given $X_i \in \{0, 1\}$ (0 means control, 1 means treatment) and Y_i be the outcome for observation i. Let $n_0 = \sum_{i=1}^{n} \mathbf{1}\{X_i = 0\}$ and $n_1 = \sum_{i=1}^{n} \mathbf{1}\{X_i = 1\}$ be the number of control and treatment observations, respectively. Let $\bar{Y}_0$ and $\bar{Y}_1$ be the sample means of the control and treatment groups, respectively, and $\hat{\sigma}_0^2$ and $\hat{\sigma}_1^2$ the sample variance, defined as

$$\hat{\sigma}_g^2 = \frac{1}{n_g - 1} \sum_{i:X_i=g} (Y_i - \bar{Y}_g)^2, \quad g \in \{0, 1\}$$

The pooled variance two sample t-statistic is given by:

$$t = \frac{\bar{Y}_1 - \bar{Y}_0}{s_p \sqrt{\frac{1}{n_0} + \frac{1}{n_1}}}$$

where the pooled standard deviation is:

$$s_p = \sqrt{\frac{(n_0 - 1)\hat{\sigma}_0^2 + (n_1 - 1)\hat{\sigma}_1^2}{n_0 + n_1 - 2}}$$

Lastly, we are given the linear regression model $Y = \beta_0 + \beta_1 X + \epsilon$.

(a) We can show that the OLS slope estimate equals the difference between the sample mean of the two groups as follows:
OLS satisfies the normal equations:

$$\sum_{i=1}^{n} (Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_i) = 0, \quad \sum_{i=1}^{n} X_i (Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_i) = 0$$

We know the group means should be:

$$\bar{Y}_0 = \frac{1}{n_0} \sum_{i:X_i=0} Y_i, \quad \bar{Y}_1 = \frac{1}{n_1} \sum_{i:X_i=1} Y_i$$

Using the second normal equation and that X_i is binary ($X_i^2 = X_i$), we have:

$$\sum_{i=1}^{n} X_i Y_i - \hat{\beta}_0 \sum_{i=1}^{n} X_i - \hat{\beta}_1 \sum_{i=1}^{n} X_i = 0$$

Again, because X_i is binary, any summation terms with $X_i = 0$ will vanish, so we can simplify the above to:

$$\begin{aligned}
\sum_{i:X_i=1} Y_i - \hat{\beta}_0 \sum_{i:X_i=1} 1 - \hat{\beta}_1 \sum_{i:X_i=1} 1 = 0 \\
\sum_{i:X_i=1} Y_i - \hat{\beta}_0 n_1 - \hat{\beta}_1 n_1 = 0 \\
\frac{1}{n_1} \sum_{i:X_i=1} Y_i = \hat{\beta}_0 + \hat{\beta}_1 \\
\bar{Y}_1 = \hat{\beta}_0 + \hat{\beta}_1
\end{aligned}$$

Now we go back to the first equation and simplify by splitting into their respective groups:

$$\begin{aligned}
\sum_{i:X_i=0} (Y_i - \hat{\beta}_0) + \sum_{i:X_i=1} (Y_i - \hat{\beta}_0 - \hat{\beta}_1) = 0 \\
\sum_{i:X_i=0} Y_i - n_0 \hat{\beta}_0 + \sum_{i:X_i=1} Y_i - n_1 \hat{\beta}_0 - n_1 \hat{\beta}_1 = 0 \\
\sum_{i:X_i=0} Y_i + \sum_{i:X_i=1} Y_i = (n_0 + n_1)\hat{\beta}_0 + n_1 \hat{\beta}_1
\end{aligned}$$

Substitute in the group means and the equation for $\bar{Y}_1$:

$$\begin{aligned}
n_0 \bar{Y}_0 + n_1 \bar{Y}_1 = (n_0 + n_1)\hat{\beta}_0 + n_1 \hat{\beta}_1 \\
n_0 \bar{Y}_0 + n_1 (\hat{\beta}_0 + \hat{\beta}_1) = (n_0 + n_1)\hat{\beta}_0 + n_1 \hat{\beta}_1 \\
n_0 \bar{Y}_0 = n_0 \hat{\beta}_0 \\
\bar{Y}_0 = \hat{\beta}_0
\end{aligned}$$

Finally, we can substitute $\hat{\beta}_0$ back into the equation for $\bar{Y}_1$ to get:

$$\begin{aligned}
\bar{Y}_1 = \bar{Y}_0 + \hat{\beta}_1 \\
\hat{\beta}_1 = \bar{Y}_1 - \bar{Y}_0
\end{aligned}$$

which proves that the OLS slope estimate is the difference between the sample means of the two groups.

(b) We can show that the OLS intercept estimate equals the sample mean of the control group in the same steps we took above with the first normal equation:

$$\sum_{i:X_i=0}(Y_i - \hat{\beta}_0) + \sum_{i:X_i=1}(Y_i - \hat{\beta}_0 - \hat{\beta}_1) = 0$$
$$\sum_{i:X_i=0} Y_i - n_0\hat{\beta}_0 + \sum_{i:X_i=1} Y_i - n_1\hat{\beta}_0 - n_1\hat{\beta}_1 = 0$$
$$\sum_{i:X_i=0} Y_i + \sum_{i:X_i=1} Y_i = (n_0 + n_1)\hat{\beta}_0 + n_1\hat{\beta}_1$$

Substitute in the group means and the equation for $\bar{Y}_1$:

$$n_0\bar{Y}_0 + n_1\bar{Y}_1 = (n_0 + n_1)\hat{\beta}_0 + n_1\hat{\beta}_1$$
$$n_0\bar{Y}_0 + n_1(\hat{\beta}_0 + \hat{\beta}_1) = (n_0 + n_1)\hat{\beta}_0 + n_1\hat{\beta}_1$$
$$n_0\bar{Y}_0 = n_0\hat{\beta}_0$$
$$\bar{Y}_0 = \hat{\beta}_0$$

(c) We can show that the homoscedastic OLS estimator of the error standard deviation equals the pooled standard deviation as follows: We know $\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_i$, so:

$$\hat{\sigma}^2 = \frac{1}{n-2}\sum_{i=1}^{n}(Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_i)^2$$

We can split the above summation into the control and treatment groups:

$$\hat{\sigma}^2 = \frac{1}{n-2}\left(\sum_{i:X_i=0}(Y_i - \hat{\beta}_0)^2 + \sum_{i:X_i=1}(Y_i - \hat{\beta}_0 - \hat{\beta}_1)^2\right)$$

Substitute in the equations for $\hat{\beta}_0$ and $\hat{\beta}_1$:

$$\hat{\sigma}^2 = \frac{1}{n-2}\left(\sum_{i:X_i=0}(Y_i - \bar{Y}_0)^2 + \sum_{i:X_i=1}(Y_i - \bar{Y}_1)^2\right)$$

Recall that the corresponding within-group sample variances are:

$$\hat{\sigma}_0^2 = \frac{1}{n_0 - 1}\sum_{i:X_i=0}(Y_i - \bar{Y}_0)^2, \quad \hat{\sigma}_1^2 = \frac{1}{n_1 - 1}\sum_{i:X_i=1}(Y_i - \bar{Y}_1)^2$$

We can rearrange the above to get:

$$\sum_{i:X_i=0}(Y_i - \bar{Y}_0)^2 = (n_0 - 1)\hat{\sigma}_0^2, \quad \sum_{i:X_i=1}(Y_i - \bar{Y}_1)^2 = (n_1 - 1)\hat{\sigma}_1^2$$

Substitute the above into the equation for $\hat{\sigma}^2$:

$$\hat{\sigma}^2 = \frac{1}{n-2}\left((n_0-1)\hat{\sigma}_0^2 + (n_1-1)\hat{\sigma}_1^2\right)$$
$$\hat{\sigma} = \sqrt{\frac{(n_0-1)\hat{\sigma}_0^2 + (n_1-1)\hat{\sigma}_1^2}{n-2}} = s_p$$

which proves that the homoscedastic OLS estimator of the error standard deviation equals the pooled standard deviation.

(d) We can show that the OLS t-statistic for the slope estimate equals the two-sample t-statistic as follows:
We know our null hypothesis $H_0 : \beta_1 = 0$ and the OLS t-statistic is given by:

$$t_0 = \frac{\hat{\beta}_1}{\hat{se}(\hat{\beta}_1)}$$

We need to prove that:

$$\frac{\hat{\beta}_1}{\hat{se}(\hat{\beta}_1)} = \frac{\bar{Y}_1 - \bar{Y}_0}{s_p\sqrt{\frac{1}{n_0}+\frac{1}{n_1}}}$$

From the previous parts, we've already proved that $\hat{\beta}_1 = \bar{Y}_1 - \bar{Y}_0$ and $\hat{\sigma} = s_p$. We just need to show that $\hat{se}(\hat{\beta}_1) = s_p\sqrt{\frac{1}{n_0}+\frac{1}{n_1}}$. We know that the OLS standard error of the slope estimate is given by:

$$\hat{se}(\hat{\beta}_1) = \frac{\hat{\sigma}}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2}}$$

Here, we know that $\bar{X} = \frac{n_1}{n_0+n_1}$ since X_i is binary. Then we have:

$$X_i = 0: \quad X_i - \bar{X} = 0 - \frac{n_1}{n_0+n_1} = -\frac{n_1}{n_0+n_1}$$
$$X_i = 1: \quad X_i - \bar{X} = 1 - \frac{n_1}{n_0+n_1} = \frac{n_0}{n_0+n_1}$$

Thus,

$$\begin{aligned}
\sum_{i=1}^{n}(X_i - \bar{X})^2 &= \sum_{i:X_i=0}\left(-\frac{n_1}{n_0+n_1}\right)^2 + \sum_{i:X_i=1}\left(\frac{n_0}{n_0+n_1}\right)^2 \\
&= n_0\left(\frac{n_1}{n_0+n_1}\right)^2 + n_1\left(\frac{n_0}{n_0+n_1}\right)^2 \\
&= \frac{n_0 n_1^2 + n_1 n_0^2}{(n_0+n_1)^2} \\
&= \frac{n_0 n_1 (n_0+n_1)}{(n_0+n_1)^2} \\
&= \frac{n_0 n_1}{n_0+n_1}
\end{aligned}$$

Plugging this back into our equation for $\hat{se}(\hat{\beta}_1)$, and applying our previous proof that $\hat{\sigma} = s_p$ we have:

$$\hat{se}(\hat{\beta}_1) = \frac{\hat{\sigma}}{\sqrt{\frac{n_0 n_1}{n_0+n_1}}} = \hat{\sigma}\sqrt{\frac{n_0+n_1}{n_0 n_1}} = s_p\sqrt{\frac{1}{n_0}+\frac{1}{n_1}}$$

(e) In any finite sample, the OLS slope and intercept will always equal the difference in group means and the control group mean, respectively, regardless of the distribution of the error term ϵ. This comes algebraicly from the linear model, the fact that X is binary, and the OLS normal equations. However, the OLS estimator of the error standard deviation will only equal the pooled standard deviation if the error term ϵ is homoscedastic. This is because the pooled variance formula assumes a common population variance across the two groups, which the homoscedastic OLS estimator of the error standard deviation also assumes. If the error term (grouped variances) differ, then the pooled t-test is invalid and the homoscedastic OLS SE is wrong, thus breaking the equality. Likewise, the t-statistic for the slope estimate will only equal the two-sample t-statistic if the error term is homoscedastic, since the OLS t-statistic relies on the pooled SE. It also requires that the correct degrees of freedom (in this case, $n-2$) is used to calculate the OLS SE, which is the same as the degrees of freedom used to calculate the pooled SE (two degrees of freedom for β_0, β_1, and two degrees of freedom considering X_0, X_1 where $(n_0-1)+(n_1-1) = n-2$). If these assumptions are violated, then the OLS t-statistic will not equal the two-sample t-statistic.

2

See attached Jupyter notebook for code and plots.

In [2]:
```
import pandas as pd
import numpy as np

raw = pd.read_csv("boston.txt", sep=r"\s+", skiprows=22, header=None)
X = np.hstack([raw.values[::2, :], raw.values[1::2, :2]])
y = raw.values[1::2, 2]
columns = [ "crim", "zn", "indus", "chas", "nox", "rm", "age",
"dis", "rad", "tax", "ptratio", "b", "lstat"]
df = pd.DataFrame(X, columns=columns)
df["medv"] = y
```

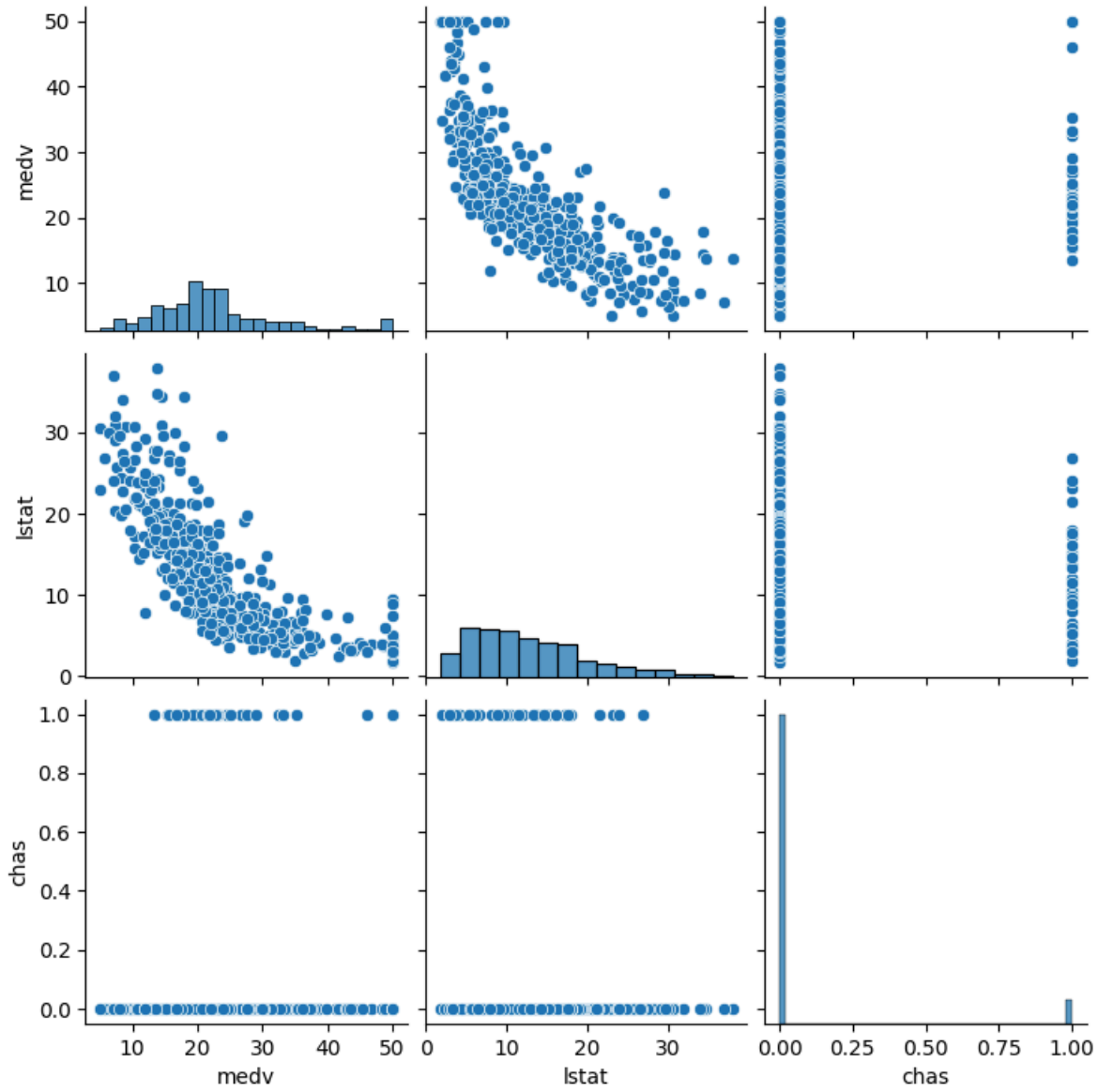
a

In [5]:
```
import matplotlib.pyplot as plt
from seaborn import pairplot
import statsmodels.api as sm

pairplot(df[["medv", "lstat", "chas"]])
X = sm.add_constant(df[["lstat", "chas"]])
model = sm.OLS(df["medv"], X).fit()
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   medv   R-squared:                       0.563
Model:                            OLS   Adj. R-squared:                  0.561
Method:                 Least Squares   F-statistic:                     323.4
Date:                Tue, 10 Feb 2026   Prob (F-statistic):           4.93e-91
Time:                        17:06:14   Log-Likelihood:                -1631.1
No. Observations:                 506   AIC:                             3268.
Df Residuals:                     503   BIC:                             3281.
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         34.0941      0.561     60.809      0.000      32.993      35.196
lstat         -0.9406      0.038    -24.729      0.000      -1.015      -0.866
chas           4.9200      1.069      4.601      0.000       2.819       7.021
==============================================================================
Omnibus:                      131.896   Durbin-Watson:                   0.975
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              275.510
Skew:                           1.406   Prob(JB):                     1.49e-60
Kurtosis:                       5.272   Cond. No.                         57.8
==============================================================================

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



The pair plots indicate a non linear relationship between `lstat` and `medv`, `medv` drops quickly for low-mid `lstat`, and then flattens out. Usually taking the log, square root, or inverse of `lstat` can help fix this. For this problem, we will use a log transformation.

Intercept: 34.0941, SE 0.561, p < 0.0001
This is the predicted `medv` for an `lstat` and `chas` value of 0. This isn't meaningful realistically as `lstat` 0 isn't possible, but this anchors the regression line. The value is statistically significant and well estimated from the low SE and p value.

`lstat` **coefficient**: -0.9406, SE 0.038, p < 0.0001
Holding `chas` constant, a 1 percentage point increase in `lstat` is attributed to about a $940.6 decrease in `medv`. The value is statistically significant and well estimated from the low SE and p value.

`chas` **coefficient**: 4.92, SE 1.069, p < 0.0001 Holding `lstat` constant, properties that border the Charles River are attributed to a $4,920 increase in `medv`. The value is

statistically significant and well estimated from the low SE and p value.

b

In [6]:
```
# add interaction term to model

df["lstat_chas"] = df["lstat"] * df["chas"]
X = sm.add_constant(df[["lstat", "chas", "lstat_chas"]])
model_interaction = sm.OLS(df["medv"], X).fit()
print(model_interaction.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   medv   R-squared:                       0.569
Model:                            OLS   Adj. R-squared:                  0.566
Method:                 Least Squares   F-statistic:                     220.8
Date:                Tue, 10 Feb 2026   Prob (F-statistic):           2.68e-91
Time:                        17:16:58   Log-Likelihood:                -1627.4
No. Observations:                 506   AIC:                             3263.
Df Residuals:                     502   BIC:                             3280.
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         33.7672      0.570     59.222      0.000      32.647      34.887
lstat         -0.9150      0.039    -23.478      0.000      -0.992      -0.838
chas           9.8251      2.103      4.672      0.000       5.693      13.957
lstat_chas    -0.4329      0.160     -2.703      0.007      -0.748      -0.118
==============================================================================
Omnibus:                      130.570   Durbin-Watson:                   1.007
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              275.834
Skew:                           1.383   Prob(JB):                     1.27e-60
Kurtosis:                       5.330   Cond. No.                         114.
==============================================================================

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

Intercept: 33.7672, SE 0.570, p < 0.0001
This is the predicted `medv` for an `lstat` and `chas` value of 0. This isn't meaningful realistically as `lstat` 0 isn't possible, but this anchors the regression line. The value is statistically significant and well estimated from the low SE and p value.

`lstat` **coefficient**: -0.9150, SE 0.039, p < 0.0001
Holding `chas` constant, a 1 percentage point increase in `lstat` is attributed to about a $915 decrease in `medv`. The value is statistically significant and well estimated from the low SE and p value. The coefficient is slightly smaller than the model without the interaction term.

`chas` **coefficient**: 9.8251, SE 2.103, p < 0.0001 Holding `lstat` constant, properties that border the Charles River are attributed to a $9,825.10 increase in `medv`. The value is statistically significant and well estimated from the low SE and p value. The coefficient is significantly larger than the model without the interaction term.

`lstat:chas` **coefficient**: -0.4329, SE 0.160, p 0.007
The negative effect of `lstat` is stronger for properties bordering the Charles River. The value is statistically significant and well estimated from the low SE and p value. Specifically, for properties that border the Charles River, a 1 percentage point increase in `lstat` decreased `medv` by $1,347.90 compared to the $915 elsewhere.

Given that the new R squared value of this model (0.569, 0.566) is barely an improvement from the previous model (0.563, 0.561), interactions do not improve the model, but the term is real and statistically meaningful.

C

In [7]:
```
# fit two separate models of medv as a function of lstat
# one model for chas = 0 and one model for chas = 1
# compare to joint interaction model to determine which is better

df0 = df[df["chas"] == 0]
df1 = df[df["chas"] == 1]

X0 = sm.add_constant(df0["lstat"])
model0 = sm.OLS(df0["medv"], X0).fit()
print(model0.summary())

X1 = sm.add_constant(df1["lstat"])
model1 = sm.OLS(df1["medv"], X1).fit()
print(model1.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   medv   R-squared:                       0.552
Model:                            OLS   Adj. R-squared:                  0.551
Method:                 Least Squares   F-statistic:                     577.2
Date:                Tue, 10 Feb 2026   Prob (F-statistic):           9.59e-84
Time:                        17:25:11   Log-Likelihood:                -1504.9
No. Observations:                 471   AIC:                             3014.
Df Residuals:                     469   BIC:                             3022.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         33.7672      0.557     60.603      0.000      32.672      34.862
lstat         -0.9150      0.038    -24.026      0.000      -0.990      -0.840
==============================================================================
Omnibus:                      139.989   Durbin-Watson:                   1.006
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              330.529
Skew:                           1.523   Prob(JB):                     1.68e-72
Kurtosis:                       5.751   Cond. No.                         30.0
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spe
cified.
                            OLS Regression Results
==============================================================================
Dep. Variable:                   medv   R-squared:                       0.582
Model:                            OLS   Adj. R-squared:                  0.569
Method:                 Least Squares   F-statistic:                     45.90
Date:                Tue, 10 Feb 2026   Prob (F-statistic):           1.01e-07
Time:                        17:25:11   Log-Likelihood:                -120.33
No. Observations:                  35   AIC:                             244.7
Df Residuals:                      33   BIC:                             247.8
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         43.5923      2.593     16.814      0.000      38.318      48.867
lstat         -1.3479      0.199     -6.775      0.000      -1.753      -0.943
==============================================================================
Omnibus:                        1.398   Durbin-Watson:                   1.269
Prob(Omnibus):                  0.497   Jarque-Bera (JB):                1.349
Skew:                           0.395   Prob(JB):                        0.509
Kurtosis:                       2.453   Cond. No.                         25.9
==============================================================================

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

Model	n	R^2	Adj R^2	AIC	BIC
Interaction	506	0.569	0.566	3263	3281

Model	n	R^2	Adj R^2	AIC	BIC
`chas` = 0	471	0.552	0.551	3014	3022
`chas` = 1	35	0.582	0.569	244.7	247.8

Note that the slopes for each `chas` specific model are the same as the indicated slopes in the interaction model when accounting for the interaction term. Additionally, the `chas` = 1 model only has 35 samples. Thus, we prefer the interaction model due to it using all 506 observations, and provides a direct statistical test for slope differences in the `chas` specific models with better estimates. The comparison given by the model is straighforward and captures the majority of the information given by both groups.

d

In [ ]:
```
df_restrict = df[df["rad"] <= 8]

X_num = sm.add_constant(df_restrict[["lstat", "rad"]].astype(float))
model_num = sm.OLS(df_restrict["medv"].astype(float), X_num).fit()
print(model_num.summary())

rad_dummies = pd.get_dummies(df_restrict["rad"], prefix="rad")
X_cat = sm.add_constant(pd.concat([df_restrict["lstat"].astype(float), rad_dummies.
model_cat = sm.OLS(df_restrict["medv"].astype(float), X_cat).fit()
print(model_cat.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   medv   R-squared:                       0.458
Model:                            OLS   Adj. R-squared:                  0.455
Method:                 Least Squares   F-statistic:                     156.7
Date:                Tue, 10 Feb 2026   Prob (F-statistic):           4.77e-50
Time:                        17:37:08   Log-Likelihood:                -1212.6
No. Observations:                 374   AIC:                             2431.
Df Residuals:                     371   BIC:                             2443.
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         33.0491      1.096     30.164      0.000      30.895      35.204
lstat         -0.9524      0.054    -17.592      0.000      -1.059      -0.846
rad            0.3819      0.197      1.937      0.054      -0.006       0.770
==============================================================================
Omnibus:                       85.140   Durbin-Watson:                   0.926
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              147.053
Skew:                           1.313   Prob(JB):                     1.17e-32
Kurtosis:                       4.593   Cond. No.                         44.0
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spe
cified.
                            OLS Regression Results
==============================================================================
Dep. Variable:                   medv   R-squared:                       0.500
Model:                            OLS   Adj. R-squared:                  0.489
Method:                 Least Squares   F-statistic:                     45.67
Date:                Tue, 10 Feb 2026   Prob (F-statistic):           1.40e-50
Time:                        17:37:08   Log-Likelihood:                -1197.4
No. Observations:                 374   AIC:                             2413.
Df Residuals:                     365   BIC:                             2448.
Df Model:                           8
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         30.5985      0.585     52.271      0.000      29.447      31.750
lstat         -0.9116      0.054    -16.761      0.000      -1.019      -0.805
rad_1.0        0.4850      1.239      0.391      0.696      -1.952       2.922
rad_2.0        5.3733      1.141      4.708      0.000       3.129       7.618
rad_3.0        5.6042      0.930      6.026      0.000       3.775       7.433
rad_4.0        1.9095      0.646      2.958      0.003       0.640       3.179
rad_5.0        4.8236      0.617      7.824      0.000       3.611       6.036
rad_6.0        1.5964      1.116      1.430      0.154      -0.599       3.792
rad_7.0        3.7895      1.335      2.839      0.005       1.165       6.414
rad_8.0        7.0170      1.139      6.159      0.000       4.777       9.257
==============================================================================
Omnibus:                       93.551   Durbin-Watson:                   1.000
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              181.753
Skew:                           1.350   Prob(JB):                     3.41e-40
Kurtosis:                       5.093   Cond. No.                     1.10e+17
```

```
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spe
cified.
[2] The smallest eigenvalue is 4.55e-30. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

Model	n	R^2	Adj R^2	AIC	BIC
Numerical `rad`	374	0.458	0.455	2431	2443
Categorical `rad`	374	0.500	0.489	2413	2448

Using `rad` as categorical significantly improves R^2 by 0.4. The AIC strongly favors the categorical specification, as the delta of 18 shows that allowing non-linear, level-specific effects of rad improves the fit. BIC slightly favors numerical specification however, as rad is heavily penalized with extra parameters as a categorical variable. Looking at the coefficients, the numerical rad value is barely statistically significant with a p-value of 0.054. However for the categorical model, different rads have varying statistical significance levels. This matches how rad is constructed realistically as it is an index, not a continuous economic quantity. Thus overall, we prefer the categorical specification due to higher adjusted R^2 and better representation of the predictor variables.

e

In [13]:
```
b3 = model_cat.params["rad_3.0"]
b4 = model_cat.params["rad_4.0"]
diff = b3 - b4
print(f"Estimated difference in expected value of medv between houses w/ rad = 3 an

cov = model_cat.cov_params()
var_diff = (
    cov.loc["rad_3.0", "rad_3.0"]
    + cov.loc["rad_4.0", "rad_4.0"]
    - 2 * cov.loc["rad_3.0", "rad_4.0"]
)
se_diff = np.sqrt(var_diff)
print(f"Estimated standard error of the difference: {se_diff}")
```

```
Estimated difference in expected value of medv between houses w/ rad = 3 and rad =
4: 3.6947186903988825
Estimated standard error of the difference: 1.1451240401723881
```

LLM Usage: All work was done by myself in VSCode with GitHub Copilot integration. The integration "provides code suggestions, explanations, and automated implementations based on natural language prompts and existing code context," and also offers autonomous coding and an in-IDE chat interface that is able to interact with the current codebase. Only the Copilot provided automatic inline suggestions for both LaTex and Python in `.tex` and `.ipynb` Jupyter notebook files respectively were taken into account / used.