Assignment 2

Simple and Multiple Linear Regression

Author: Samuel Fredric Berg

Student ID: sb224sc

Date: 2026-01-25

Course: Machine Learning 4DT905

Imports

In [1]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import statsmodels.api as sm
```

Load data

In [2]:
```
df = pd.read_csv("../data/Boston.csv", index_col=0)
```

Number of predictors and names

In [3]:
```
df_names = df.columns.tolist()
print(f"Number of columns: {len(df_names)}")
print(f"Column names: {df_names}")
```

```
Number of columns: 14
Column names: ['crim', 'zn', 'indus', 'chas', 'nox', 'rm', 'age', 'dis', 'rad',
'tax', 'ptratio', 'black', 'lstat', 'medv']
```

Statistical summary of predictors

In [4]:
```
df.describe()
```

Out[4]:

	crim	zn	indus	chas	nox	rm
count	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000
mean	3.613524	11.363636	11.136779	0.069170	0.554695	6.284634
std	8.601545	23.322453	6.860353	0.253994	0.115878	0.702617
min	0.006320	0.000000	0.460000	0.000000	0.385000	3.561000
25%	0.082045	0.000000	5.190000	0.000000	0.449000	5.885500
50%	0.256510	0.000000	9.690000	0.000000	0.538000	6.208500
75%	3.677083	12.500000	18.100000	0.000000	0.624000	6.623500
max	88.976200	100.000000	27.740000	1.000000	0.871000	8.780000

Number of datapoints

In [5]:
```python
print(f"Number of datapoints: {len(df)}")
```

```
Number of datapoints: 506
```

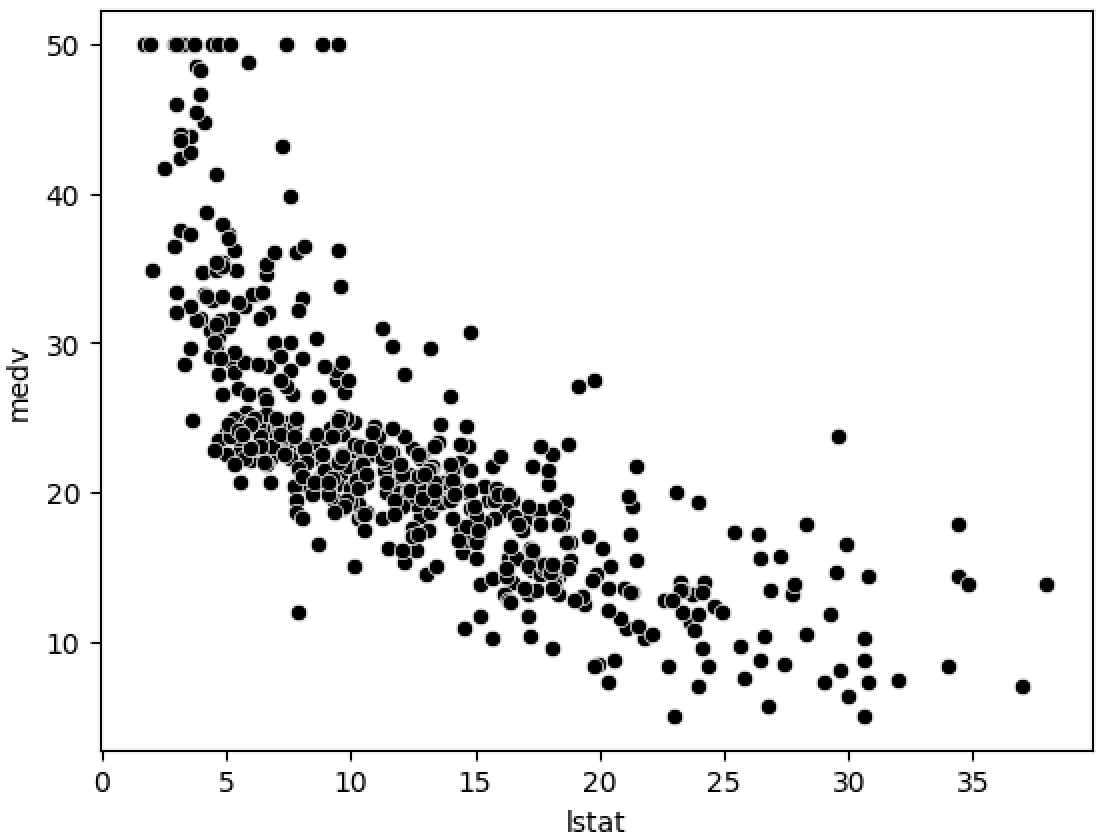
Display data in table format

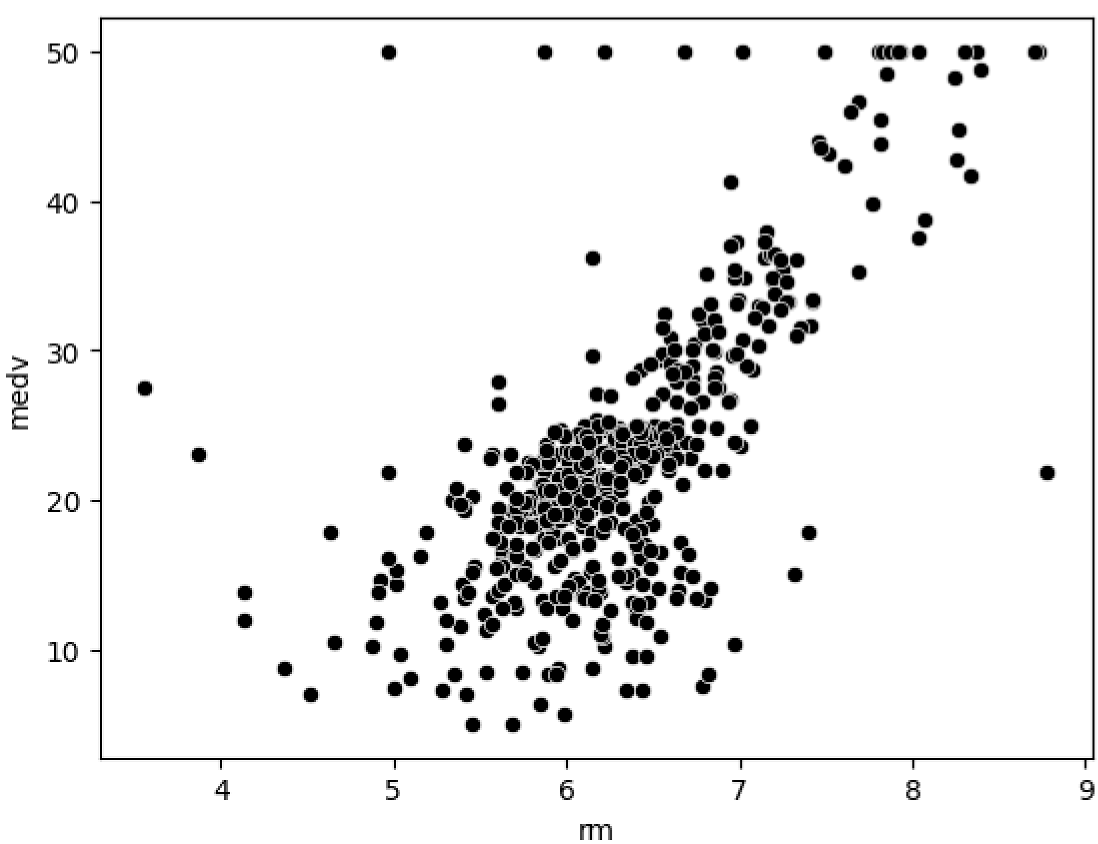
In [6]:
```python
print(df.head(5))
```

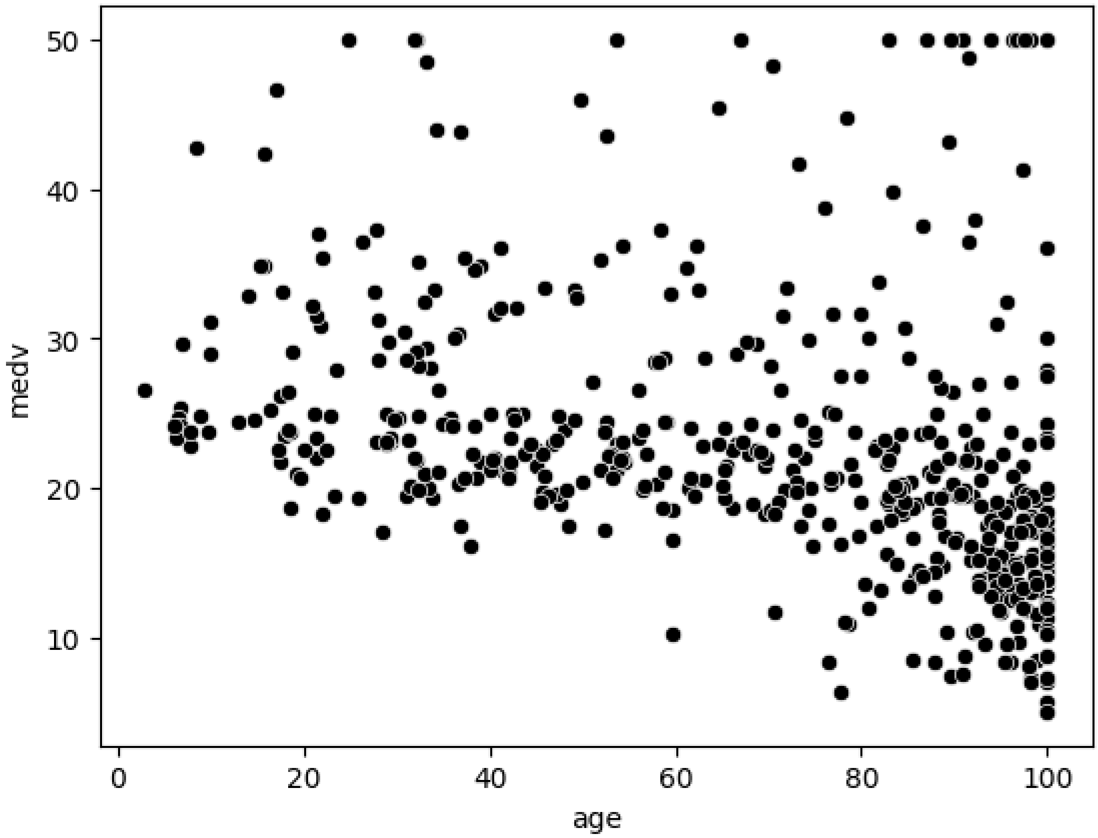
```
      crim    zn  indus  chas    nox     rm   age     dis  rad  tax  ptratio  \
1  0.00632  18.0   2.31     0  0.538  6.575  65.2  4.0900    1  296     15.3
2  0.02731   0.0   7.07     0  0.469  6.421  78.9  4.9671    2  242     17.8
3  0.02729   0.0   7.07     0  0.469  7.185  61.1  4.9671    2  242     17.8
4  0.03237   0.0   2.18     0  0.458  6.998  45.8  6.0622    3  222     18.7
5  0.06905   0.0   2.18     0  0.458  7.147  54.2  6.0622    3  222     18.7

    black  lstat  medv
1  396.90   4.98  24.0
2  396.90   9.14  21.6
3  392.83   4.03  34.7
4  394.63   2.94  33.4
5  396.90   5.33  36.2
```

Plot lstat, rm and age against medv

In [7]:
```python
sns.scatterplot(x="lstat", y="medv", data=df, color="black")
plt.show()

sns.scatterplot(x="rm", y="medv", data=df, color="black")
plt.show()

sns.scatterplot(x="age", y="medv", data=df, color="black")
plt.show()
```


medv
50
40
30
20
10
0
5
10
15
20
25
30
35
lstat


medv
50
40
30
20
10
4
5
6
7
8
9
rm



Linear regression

In [8]:
```
regression1 = sm.OLS(df["medv"], sm.add_constant(df["lstat"])).fit()
print(regression1.summary())

regression2 = sm.OLS(df["medv"], sm.add_constant(df["rm"])).fit()
print(regression2.summary())

regression3 = sm.OLS(df["medv"], sm.add_constant(df["age"])).fit()
print(regression3.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   medv   R-squared:                       0.544
Model:                            OLS   Adj. R-squared:                  0.543
Method:                 Least Squares   F-statistic:                     601.6
Date:                Sun, 25 Jan 2026   Prob (F-statistic):           5.08e-88
Time:                        14:43:14   Log-Likelihood:                -1641.5
No. Observations:                 506   AIC:                             3287.
Df Residuals:                     504   BIC:                             3295.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         34.5538      0.563     61.415      0.000      33.448      35.659
lstat         -0.9500      0.039    -24.528      0.000      -1.026      -0.874
==============================================================================
Omnibus:                      137.043   Durbin-Watson:                   0.892
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              291.373
Skew:                           1.453   Prob(JB):                     5.36e-64
Kurtosis:                       5.319   Cond. No.                         29.7
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctl
y specified.
                            OLS Regression Results
==============================================================================
Dep. Variable:                   medv   R-squared:                       0.484
Model:                            OLS   Adj. R-squared:                  0.483
Method:                 Least Squares   F-statistic:                     471.8
Date:                Sun, 25 Jan 2026   Prob (F-statistic):           2.49e-74
Time:                        14:43:14   Log-Likelihood:                -1673.1
No. Observations:                 506   AIC:                             3350.
Df Residuals:                     504   BIC:                             3359.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        -34.6706      2.650    -13.084      0.000     -39.877     -29.465
rm             9.1021      0.419     21.722      0.000       8.279       9.925
==============================================================================
Omnibus:                      102.585   Durbin-Watson:                   0.684
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              612.449
Skew:                           0.726   Prob(JB):                    1.02e-133
Kurtosis:                       8.190   Cond. No.                         58.4
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctl
y specified.
                            OLS Regression Results
==============================================================================
```

```
Dep. Variable:                   medv   R-squared:                       0.142
Model:                            OLS   Adj. R-squared:                  0.140
Method:                 Least Squares   F-statistic:                     83.48
Date:                Sun, 25 Jan 2026   Prob (F-statistic):           1.57e-18
Time:                        14:43:14   Log-Likelihood:                -1801.5
No. Observations:                 506   AIC:                             3607.
Df Residuals:                     504   BIC:                             3615.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         30.9787      0.999     31.006      0.000      29.016      32.942
age           -0.1232      0.013     -9.137      0.000      -0.150      -0.097
==============================================================================
Omnibus:                      170.034   Durbin-Watson:                   0.613
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              456.983
Skew:                           1.671   Prob(JB):                    5.85e-100
Kurtosis:                       6.240   Cond. No.                         195.
==============================================================================

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

Interpretation of regressions

lstat

For `lstat` gives R-squared value of (0.544), which means that approximately 54.4% of the variance in `medv` can be explained by the model. The F-statistic p-value (Prob(F-statistic) = 5.08e-88) is less than 0.05, leading us to reject the null hypothesis that all coefficients are zero. This indicates the model is statistically significant in predicting `medv`. The negative coefficient (-0.9500) indicates that as `lstat` increases, `medv` tends to decrease, suggesting an inverse relationship between these two variables.

rm

The same interpretation can be made for `rm`, which has an R-squared value of (0.484), F-statistic p-value of (2.49e-74), and a coefficient of (9.1021). The p-value < 0.05 indicates the model is statistically significant. The positive coefficient indicates that as `rm` increases, `medv` also tends to increase, suggesting a direct relationship between these two variables.

age

Same interpretation can be made for `age`, which has an R-squared value of (0.142), F-statistic p-value of (1.57e-18), and a coefficient of (-0.1232). The p-value < 0.05 indicates the model is statistically significant. The negative coefficient indicates that as `age` increases, `medv` tends to decrease, suggesting an inverse relationship between these two variables.

In [9]:
```
print(regression1.conf_int())
print(regression2.conf_int())
print(regression3.conf_int())
```

```
               0          1
const  33.448457  35.659225
lstat  -1.026148  -0.873951
               0          1
const -39.876641 -29.464601
rm      8.278855   9.925363
               0          1
const  29.015752  32.941604
age    -0.149647  -0.096679
```

Indicates the lower and upper bounds of the 95% confidence interval. First row (y intercept) and second row (slope of predictor). Smaller intervals in the slope indicates that the model is more precise.

The second model (`rm`) shows a positive relationship (positive coefficient) with a wider confidence interval. Meanwhile the first (`lstat`) and third (`age`) models show negative relationships (negative coefficients) with narrower confidence intervals.

Use model

In [10]:
```
use_lstat = pd.DataFrame({"lstat": [5, 10, 15]})
use_lstat = sm.add_constant(use_lstat)
predictor1 = regression1.get_prediction(use_lstat).summary_frame(alpha=0.05)
print(predictor1[["mean", "obs_ci_lower", "obs_ci_upper"]])

use_rm = pd.DataFrame({"rm": [5, 6.5, 8]})
use_rm = sm.add_constant(use_rm)
perdictor2 = regression2.get_prediction(use_rm).summary_frame(alpha=0.05)
print(perdictor2[["mean", "obs_ci_lower", "obs_ci_upper"]])

use_age = pd.DataFrame({"age": [25, 50, 75]})
use_age = sm.add_constant(use_age)
perdictor3 = regression3.get_prediction(use_age).summary_frame(alpha=0.05)
print(perdictor3[["mean", "obs_ci_lower", "obs_ci_upper"]])
```

```
        mean  obs_ci_lower  obs_ci_upper
0  29.803594     17.565675     42.041513
1  25.053347     12.827626     37.279068
2  20.303101      8.077742     32.528459
        mean  obs_ci_lower  obs_ci_upper
0  10.839924     -2.214474     23.894322
1  24.493088     11.480391     37.505784
2  38.146251     25.058353     51.234149
        mean  obs_ci_lower  obs_ci_upper
0  27.899610     11.090368     44.708852
1  24.820542      8.043748     41.597335
2  21.741474      4.971031     38.511917
```

Interpretation of results

lstat

Inserted values for `lstat` where 5, 10, 15 this means that with 95% confidence the predicted `medv` values will respectivly be between (17.56, 42.04), (12.82, 37.27) and (8.07, 32.52) approximatly.

rm & age

Same interpretation can be made for `rm` and `age` where inserted values are 5, 6.5, 8 for `rm` and 25, 50, 75 for `age`.

In [11]:
```
regression = sm.OLS(df["medv"], sm.add_constant(df[["lstat", "rm", "age"]])).f
print(regression.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   medv   R-squared:                       0.639
Model:                            OLS   Adj. R-squared:                  0.637
Method:                 Least Squares   F-statistic:                     296.2
Date:                Sun, 25 Jan 2026   Prob (F-statistic):         1.20e-110
Time:                        14:43:14   Log-Likelihood:                -1582.4
No. Observations:                 506   AIC:                             3173.
Df Residuals:                     502   BIC:                             3190.
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         -1.1753      3.182     -0.369      0.712      -7.427       5.076
lstat         -0.6685      0.054    -12.298      0.000      -0.775      -0.562
rm             5.0191      0.454     11.048      0.000       4.127       5.912
age            0.0091      0.011      0.811      0.418      -0.013       0.031
==============================================================================
Omnibus:                      138.819   Durbin-Watson:                   0.851
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              415.436
Skew:                           1.296   Prob(JB):                     6.15e-91
Kurtosis:                       6.603   Cond. No.                         985.
==============================================================================

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

Here the R-squared value (0.639) indicates that approximately 63.9% of the variance in `medv` can be explained by the model. The F-statistic p-value (Prob(F-statistic) = 1.20e-110) is less than 0.05, indicating we reject the null hypothesis that all coefficients are zero. This means the model is statistically significant in predicting `medv`.

In [12]:

```
regression = sm.OLS(df["medv"], sm.add_constant(df.drop(columns=["medv"]))).fi
print(regression.summary())
```

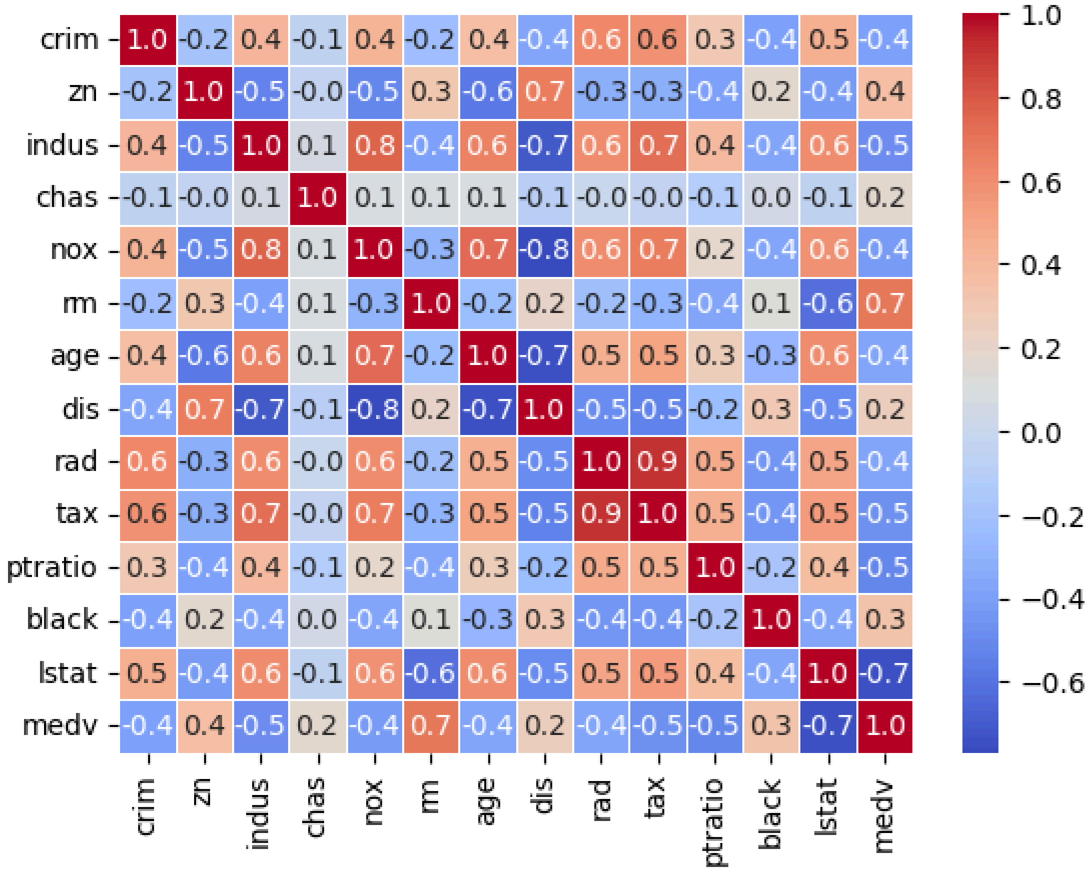
```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   medv   R-squared:                       0.741
Model:                            OLS   Adj. R-squared:                  0.734
Method:                 Least Squares   F-statistic:                     108.1
Date:                Sun, 25 Jan 2026   Prob (F-statistic):          6.72e-135
Time:                        14:43:14   Log-Likelihood:                -1498.8
No. Observations:                 506   AIC:                             3026.
Df Residuals:                     492   BIC:                             3085.
Df Model:                          13
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         36.4595      5.103      7.144      0.000      26.432      46.487
crim          -0.1080      0.033     -3.287      0.001      -0.173      -0.043
zn             0.0464      0.014      3.382      0.001       0.019       0.073
indus          0.0206      0.061      0.334      0.738      -0.100       0.141
chas           2.6867      0.862      3.118      0.002       0.994       4.380
nox          -17.7666      3.820     -4.651      0.000     -25.272     -10.262
rm             3.8099      0.418      9.116      0.000       2.989       4.631
age            0.0007      0.013      0.052      0.958      -0.025       0.027
dis           -1.4756      0.199     -7.398      0.000      -1.867      -1.084
rad            0.3060      0.066      4.613      0.000       0.176       0.436
tax           -0.0123      0.004     -3.280      0.001      -0.020      -0.005
ptratio       -0.9527      0.131     -7.283      0.000      -1.210      -0.696
black          0.0093      0.003      3.467      0.001       0.004       0.015
lstat         -0.5248      0.051    -10.347      0.000      -0.624      -0.425
==============================================================================
Omnibus:                      178.041   Durbin-Watson:                   1.078
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              783.126
Skew:                           1.521   Prob(JB):                    8.84e-171
Kurtosis:                       8.281   Cond. No.                     1.51e+04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctl
y specified.
[2] The condition number is large, 1.51e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Here the R-squared value (0.741) indicates that approximately 74.1% of the variance in `medv` can be explained by the model. The F-statistic p-value (Prob(F-statistic) = 6.72e-135) is less than 0.05, indicating we reject the null hypothesis that all coefficients are zero. This means the model is statistically significant in predicting `medv`.

Correlation matrix

In [13]:

```python
sns.heatmap(df.corr(), annot=True, cmap="coolwarm", fmt=".1f", linewidths=0.5)
plt.show()
```



This matrix shows the correlation coefficients between each pair of variables in the dataset. A correlation coefficient close to 1 indicates a strong positive correlation, while a coefficient close to -1 indicates a strong negative correlation. A coefficient around 0 suggests no correlation between the variables.

Example interpretations:

`crim` and `zn` have a correlation coefficient of -0.2, indicating a weak negative correlation. This suggests that as the value of `crim` increases, the value of `zn` tends to decrease slightly.

Use multiple linear regression model

In [14]:

```
selected_predictor_values = pd.DataFrame(
    pd.MultiIndex.from_product(
        [[5, 10, 15], [5, 6.5, 8]], names=["lstat", "rm"]
    ).to_frame(index=False)
)
print(selected_predictor_values)

regression = sm.OLS(df["medv"], sm.add_constant(df[["lstat", "rm"]])).fit()
selected_predictor_values = sm.add_constant(selected_predictor_values)
predictions = regression.get_prediction(selected_predictor_values)
```

```python
pred_summary = predictions.summary_frame(alpha=0.05)

print(pred_summary[["mean", "obs_ci_lower", "obs_ci_upper"]])
```

```
   lstat   rm
0      5  5.0
1      5  6.5
2      5  8.0
3     10  5.0
4     10  6.5
5     10  8.0
6     15  5.0
7     15  6.5
8     15  8.0
        mean  obs_ci_lower  obs_ci_upper
0  20.903875      9.889729     31.918021
1  28.546057     17.635923     39.456192
2  36.188239     25.225479     47.150999
3  17.692084      6.722152     28.662016
4  25.334266     14.437027     36.231505
5  32.976448     21.995024     43.957872
6  14.480292      3.537875     25.422709
7  22.122474     11.221204     33.023745
8  29.764656     18.747835     40.781477
```

Interpretation of results

Row one indicates that for value of `lstat` (5) and `rm` (5.0) will with 95% confidence result in a `medv` value between (9.88, 31.91), meanwhile row nine indicates that for value of `lstat` (15) and `rm` (8.0) will with 95% confidence result in a `medv` value between (18.74, 40.78). Same goes for all the subsequent rows with different values for `lstat` and `rm` which also provides a new boundary for all combinations of them.