Advanced Regression
Problem Set 7: Smoothing Spline

Rahul Roy and Suryasis Jana

1. Use the following code to simulate data on a predictor X and a response y.

```
# Define the true regression function
def reg(x):
    return 5 * np.sin(x) + 23 * (np.cos(x))**2

np.random.seed(1234) # Set seed for reproducibility
# Generate data
X = np.random.uniform(5, 15, 500)
y = reg(X) + np.random.normal(0, 5, 500)
```

   The function "reg" represents the truth (i.e., the regression function) of the simulated dataset.

2. Make a random train-test split of the data with test data size 20%.

3. Generate a scatter plot of the data and plot the regression function in the same graph.

4. Apply the smoothing spline technique to the training dataset using the B-spline basis with the following values of the degrees of freedom (df): 4, 5, 10, 20, 30, 40, 50.

   For each case, plot the fitted regression line and comment on your findings (i.e., how do the estimates change with changing df?)

5. For each case, compute the training and test errors.

6. Repeat the procedure 50 times. Each time, generate a fresh dataset according to step 1 and make a train-test split according to step 2. Find the average of the training errors and the testing errors thus obtained for each case and comment.

7. Consider the "Boston" dataset and let the variable "medv" be the response (Y) and the variable "lstat" be the predictor (X). Use a k-fold cross-validation method to find the best choice of degrees of freedom for applying the smoothing spline technique.