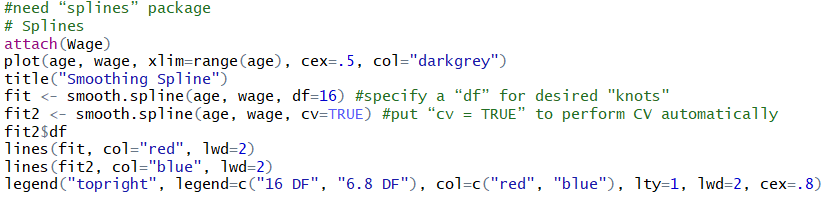
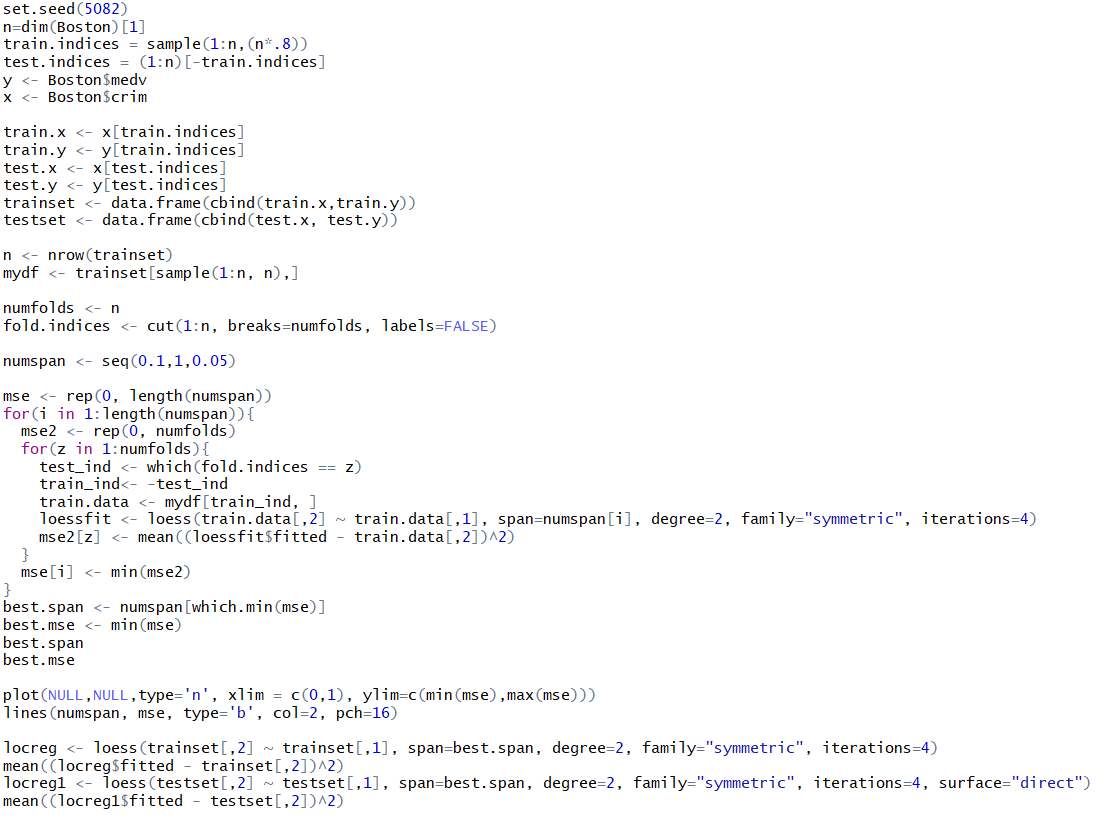
|  |  |
| --- | --- |
| * Spline – a series of knots where different cubic functions meet to account for the non-linearity in our data * A spline for every unique observation * Left function is the loss formula * Right function is the penalty formula * Lambda is our tuning parameter that encourages smoothness. * Lambda also controls the bias-variance tradeoff, similar with shrinkage methods (ridge and lasso regression) * Higher lambda = more smoothness * Lambda = 0 means no penalty, minimize RSS 🡪 0, overfit the model * Lambda = infinity means a perfectly smooth model, RSS is the same as the linear Least Squares line * When we select lambda, we are influencing the effective degrees of freedom * It’s possible to show that as lambda 🡪 infinity, the effective degrees of freedom decreases from n to 2 * Use CV to pick lambda   100   * Little difference in specifying DF and using CV to find lambda. We prefer the lower DF, simpler the better | * Doing a weighted least squares regression at target points using only nearby observations * Each new target point, requires more weights * Memory-based procedure, like nearest-neighbors, we need all the training data each time to computer a prediction   A close up of a map  Description automatically generated   * Example of selecting target points and performing local regression. Yellow line is from local regression. Blue line is the function that generated the points   Steps to perform a local regression   1. Choose how to define the weighting function K 2. Fit a linear, constant, or quadratic regression 3. Fit span S, like the tuning parameter lambda, which controls flexibility of the non-linear fit   Span   * Smaller = more local and wigglier our fit * Larger = global fit, essentially performing regular LS regression, because we are using all the training observations * Use CV to choose the best S   1. Gather the fraction s = kin of training points whose are closest  to xo.  2. Assign a weight Kio = K (:ri, to each point in this neighborhood,  so that the point furthest from To has weight zero, and the closest  has the highest weight. All but these k nearest neighbors get weight  3. Fit a weighted least squares regression of the yi on the using the  aforementioned weights, by finding and that minimize  E Ki0(Yi — '30 —  4. The fitted value at is given by = +  (7.14)   * Implement to p-dimensional neighbors, but performs poor with p>4, due to few training observations |

**Smoothing Spline Local Regression**

**Smoothing Spline**



**Local Regression**



Degree of polynomial: 0 = weighted moving average, 1 = locally linear, 2 (default) = locally quadratic, not recommended to go over 3 (overfitting)

Surface (control parameter): how the fitted surface should be computed, direct or interpolate

Iteration: # of iterations for robust fitting, only when using family = symmetric

Span: smoothing parameter, default = 0.75