Project 7

Jose Alfaro

Due: May 3, 2019

Question: 1

For this question, the predictors were standardized and the data was split into training and test sets as described in previous projects.

Part: A

To begin our analysis, we fit a support vector classifier to the training data in which the cost parameter was optimally chosen through the use of 10-fold cross-validation. We saw that the optimal cost parameter was chosen to be 0.001 with an associated gamma of 0.012 and had a total of 929 support vectors. This allowed us to create the confusion matrix shown below in which the support vector classifier predicted all of the test data to be a "No".

Predicted/Actual	No	Yes
No	941	59
Yes	0	0

Table 1: Confusion Matrix for SVC with Optimally Chosen Cost Parameter (Question: 1A)

From here we see:

$$Sensitivity: \frac{0}{59} = 0\%$$
$$Specificity: \frac{941}{941} = 100\%$$
$$Misclassification\ Rate: \frac{59}{1000} = 5.9\%$$

Part: B

In hopes of producing better results, a support vector machine was fitted with a polynomial kernel of degree two. The cost parameter was also chosen optimally through the use of 10-fold cross-validation. We saw that the optimal cost parameter was chosen to be 10 with an associated gamma of 0.5 and had a total of 51 support vectors. This allowed us to create the confusion matrix shown below in which the support vector machine classifier predicted all of the test data to be a "No" as seen in part A.

Predicted/Actual	No	Yes
No	941	59
Yes	0	0

Table 2: Confusion Matrix for Support Vector Machine with Polynomial Kernel of Degree 2 (Question: 1B)

From here we see:

$$Sensitivity: \frac{0}{59} = 0\%$$
$$Specificity: \frac{941}{941} = 100\%$$
$$Misclassification\ Rate: \frac{59}{1000} = 5.9\%$$

Part: C

Lastly, we fit a support vector machine with a radial kernel of degree 2. Both the cost and the gamma parameters were chosen optimally through the use of 10-fold cross-validation. We saw that the optimal cost parameter was chosen to be 1 and the optimal gamma parameter was chosen to be 0.5. We note that this method used a total of 34 support vectors. This allowed us to create the confusion matrix shown below in which the support vector machine classifier predicted all of the test data to be a "No" as seen in parts A and B.

Predicted/Actual	No	Yes
No	941	59
Yes	0	0

Table 3: Confusion Matrix for Support Vector Machine with Radial Kernel with γ and Cost Parameter Chosen Optimally (Question: 1C)

From here we see:

$$Sensitivity: \frac{0}{59} = 0\%$$

$$Specificity: \frac{941}{941} = 100\%$$

$$Misclassification\ Rate: \frac{59}{1000} = 5.9\%$$

Part: D

From these methods, we note that they all produced similar results with the only differences being the number of support vectors used and the differing kernels/parameters. Thus, we conclude that among these methods the best model is the most parsimonious with the least amount of estimated parameters. Therefore, the support vector classifier with only one estimated parameter from *part A* is chosen among these methods. Note, this method produced a sensitivity of 0% and a misclassification rate of 5.9%

In project 6, we looked at the same data, but took on a different approach. Here, we concluded that the best method that would maximize sensitivity while providing a relatively low misclassification rate was through a bagging approach with $B = 1,000$ and $m \approx \sqrt{p}$. As a reminder, this method produced a sensitivity of 18.75% and a misclassification rate of 6.62%.

Although, the support vector machine from *part A* has a lower misclassification rate than the method bagging method chosen in project 6, it has a very low sensitivity of 0%. Thus, if our goal is to correctly classify observations into the "Yes" and "No" categories, then the support vector machine in *part A* is the best method to use due to its low error rate. On the other hand, if our goal is to correctly classify individuals that will buy the insurance policy, then the bagging approach from project 6 would be better since it has a much higher sensitivity.

Question: 2

The business school admission data was split such that the first five observations in each category was treated as test data and the remaining observations as training data.

Part: A

A support vector classifier was fit to the training data such that the cost parameter was chosen optimally through the use of 10-fold cross-validation. From here, we saw that the optimal cost value was 1 with an associated gamma value of 0.5. We also noted that there were 23 support vectors. In order to evaluate its performance, we constructed a confusion matrix and calculated the misclassification rate (shown below). Here we saw that:

Predicted/Actual	Admit	Reject	Borderline
Admit	4	0	0
Reject	0	3	0
Borderline	1	2	5

Table 4: Confusion Matrix for SVC with Optimally Chosen Cost Parameter (Question: 2A)

$$Misclassification\ Rate: \frac{3}{15} = 20\%$$

Part: B

We repeated Part A, however this time we fit a support vector machine with a polynomial kernel of degree 2 and found the optimal value for coin using 10-fold cross-validation. We saw that the optimal cost value was 10 with an associated gamma of 0.5. Moreover, we also noted that there were 51 support vectors. To evaluate its performance, we constructed a confusion matrix and calculated the misclassification rate (shown below). Here we saw that:

Predicted/Actual	Admit	Reject	Borderline
Admit	2	5	0
Reject	2	0	0
Borderline	1	0	5

Table 5: Confusion Matrix for Support Vector Machine with Polynomial Kernel of Degree 2 (Question: 2B)

$$Misclassification\ Rate: \frac{8}{15} = 53.33\%$$

Part: C

Once again we repeated *part A*, however this time we fit a support vector machine with a radial kernel and found the optimal values for coin and gamma using 10-fold cross-validation. We saw that the optimal cost value was 10 with an associated gamma of 0.5. Moreover, we also noted that there were 23 support vectors. To evaluate its performance, we constructed a confusion matrix and calculated the misclassification rate (shown below). Here we saw that:

Predicted/Actual	Admit	Reject	Borderline
Admit	5	0	0
Reject	0	3	0
Borderline	0	2	5

Table 6: Confusion Matrix for Support Vector Machine with Radial Kernel with γ and Cost Parameter Chosen Optimally (Question: 2C)

$$Misclassification\ Rate: \frac{2}{15} = 13.33\%$$

Part: D

Out of the three methods explored here, we saw that the method using the radial kernel had the lowest misclassification rate of 13.33%. Therefore, we choose this method to be better than the others studied here.

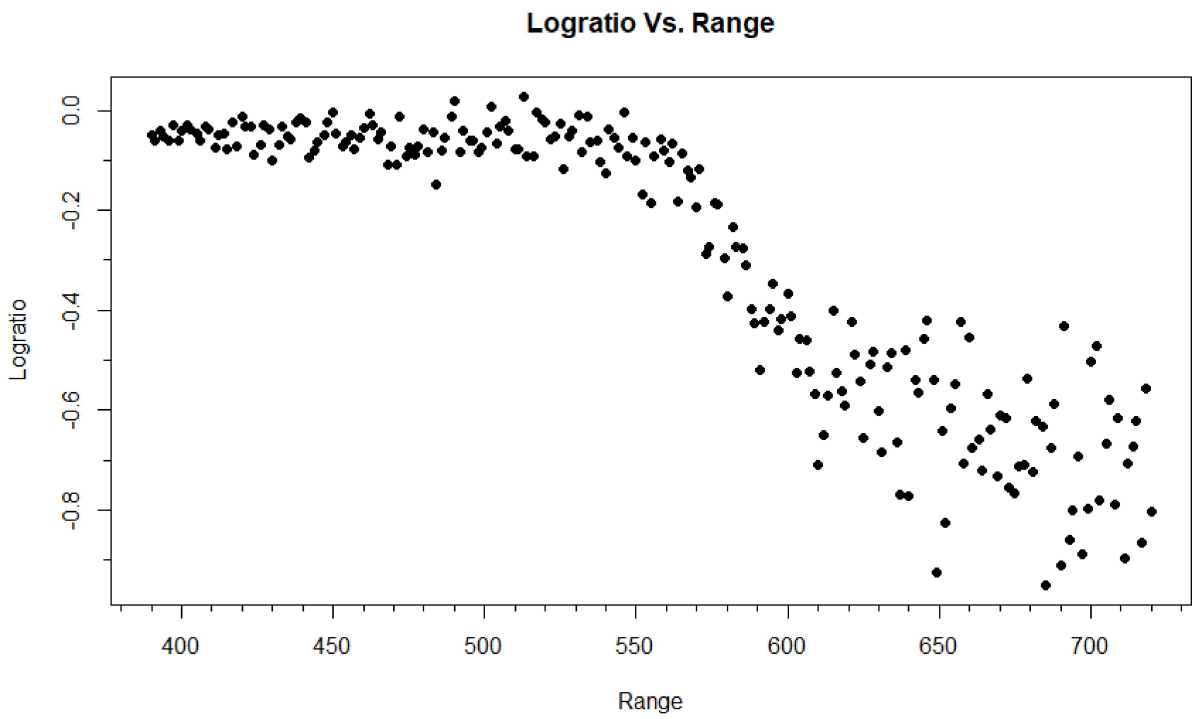
In project 3, we selected QDA as the optimal classifier since it had the lowest test error rate of 13.33% among the other methods.

It is difficult to compare these two methods since they both produced the same misclassification rate of 13.33%. Therefore, we look at the confusion matrices and see that both methods produced the exact same results. With this in mind, we chose to keep the QDA method since it makes use of the entire dataset where as the support vector machine only uses a subset of data.

Question: 3

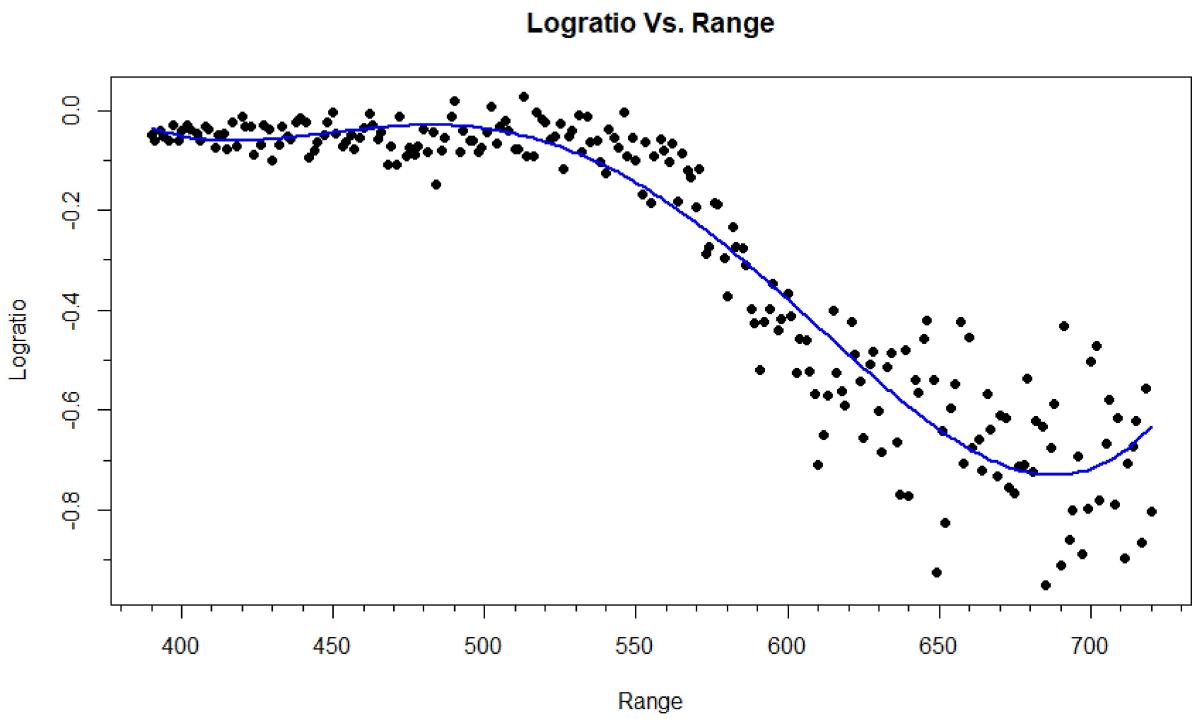
Part: A

A scatterplot of the data can be seen below. Here, we see that the data does not follow a linear trend as there seems the be an s-shaped curve. We also see that the general trend in the data is that as range increases, the logratio decreases. Moreover, the variability in the data seems to increase as range increases.



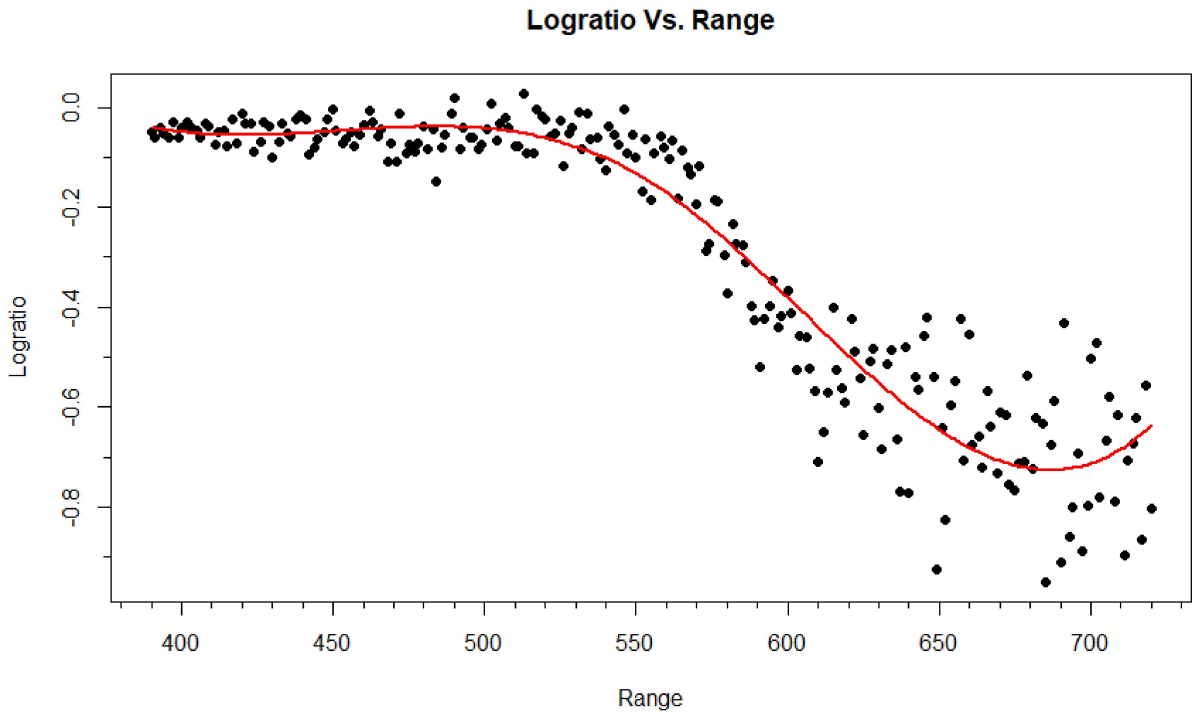
Part: B

To model the data, we fit a polynomial of degree 4 to the data (shown below). Here we saw that the polynomial of degree four does a really good job of fitting the data towards the beginning, but then struggles a little towards the end. To better explain this fit, we calculated the MSE via LOOCV. For this method we calculated the MSE to be 0.0079.



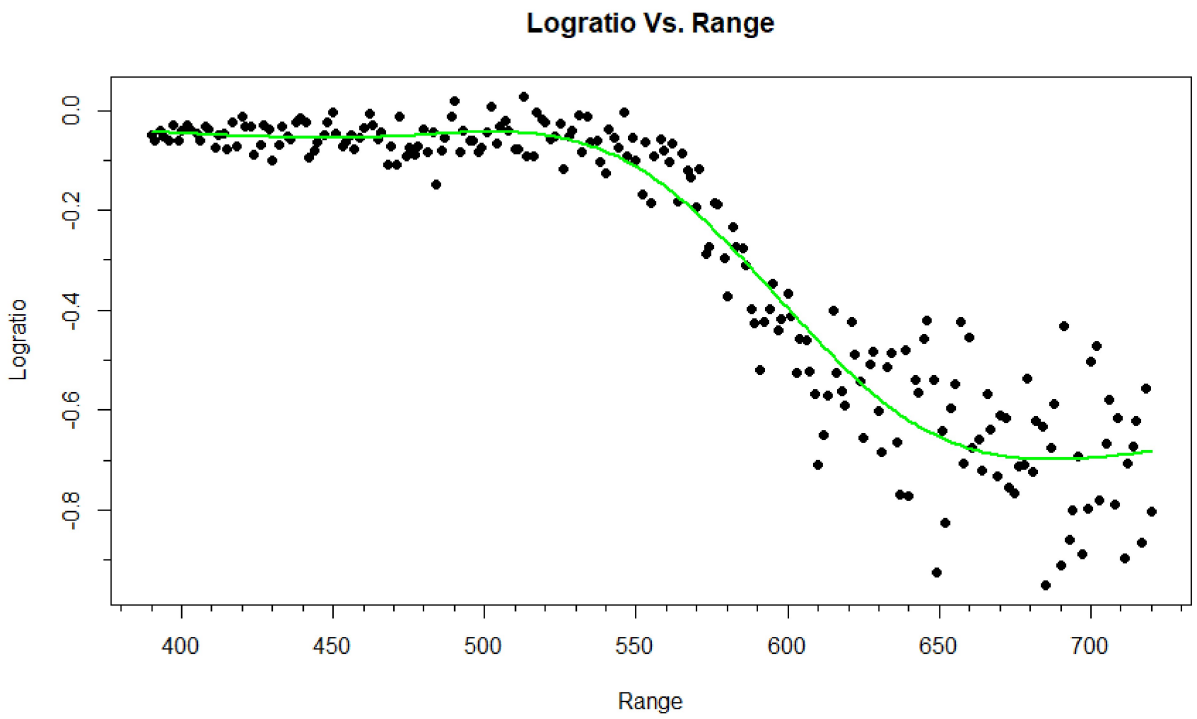
Part: C

We continued our analysis by fitting a cubic regression spline with 4 degrees of freedom (shown below). We saw that the cubic regression spline with 4 degrees of freedom fits the data well and appears to have a better overall fit than the polynomial of degree 4. However, it is noticeable that this method seems to underestimate the data trend where there is intense curvature. For this method we calculated the MSE via LOOCV to be 0.0076 which is lower than that of the polynomial of degree 4.



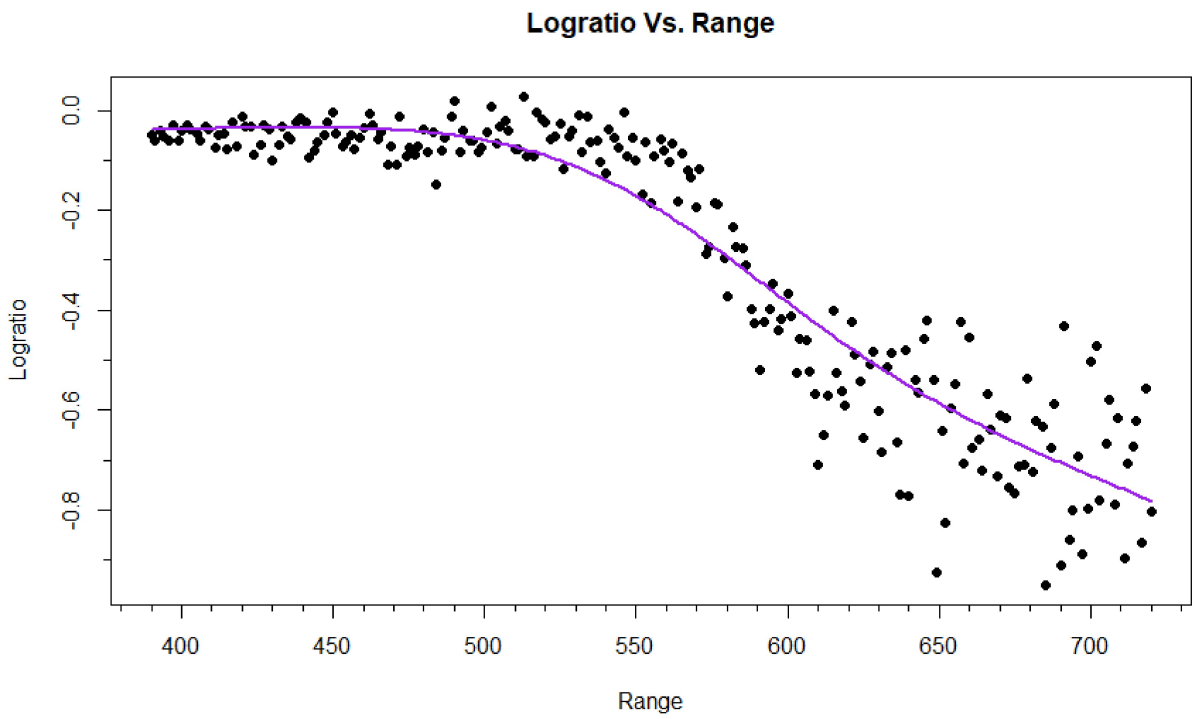
Part: D

We then fitted a natural cubic regression spline with 4 degrees of freedom (shown below). We saw that the natural cubic regression spline fit the data very well and even managed to capture the effects of the curvature towards the middle of the scatterplot. For this method we calculated the MSE via LOOCV to be 0.0070 which is lower than that of the polynomial of degree 4 and the cubic regression spline with 4 degrees of freedom.



Part: E

Lastly, we fitted a smoothing spline with 4 degrees of freedom (shown below). We saw that the smoothing spline fit did a terrible job in fitting the data since it tended to overestimate where data was well concentrated and underestimate when the data was less concentrated. Moreover, this method seemed to struggle the most out of all the methods studied in this question in capturing the curvature towards the middle of the plot



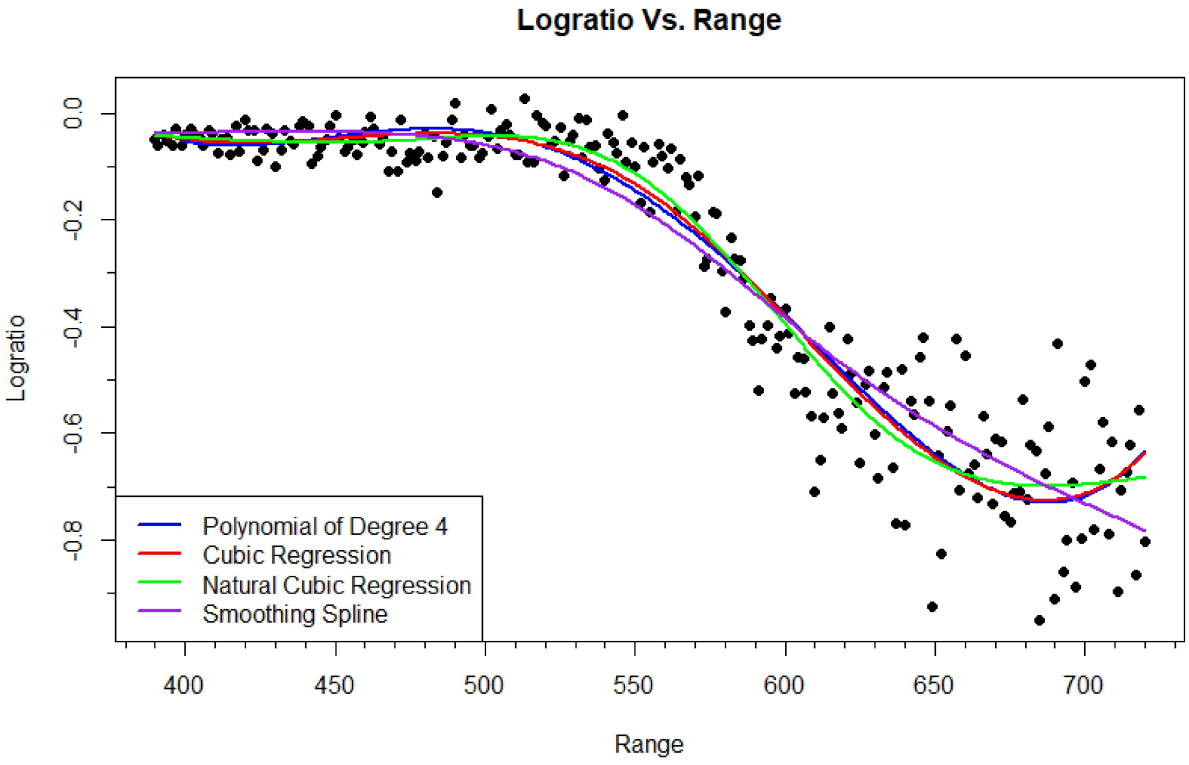
Part: F

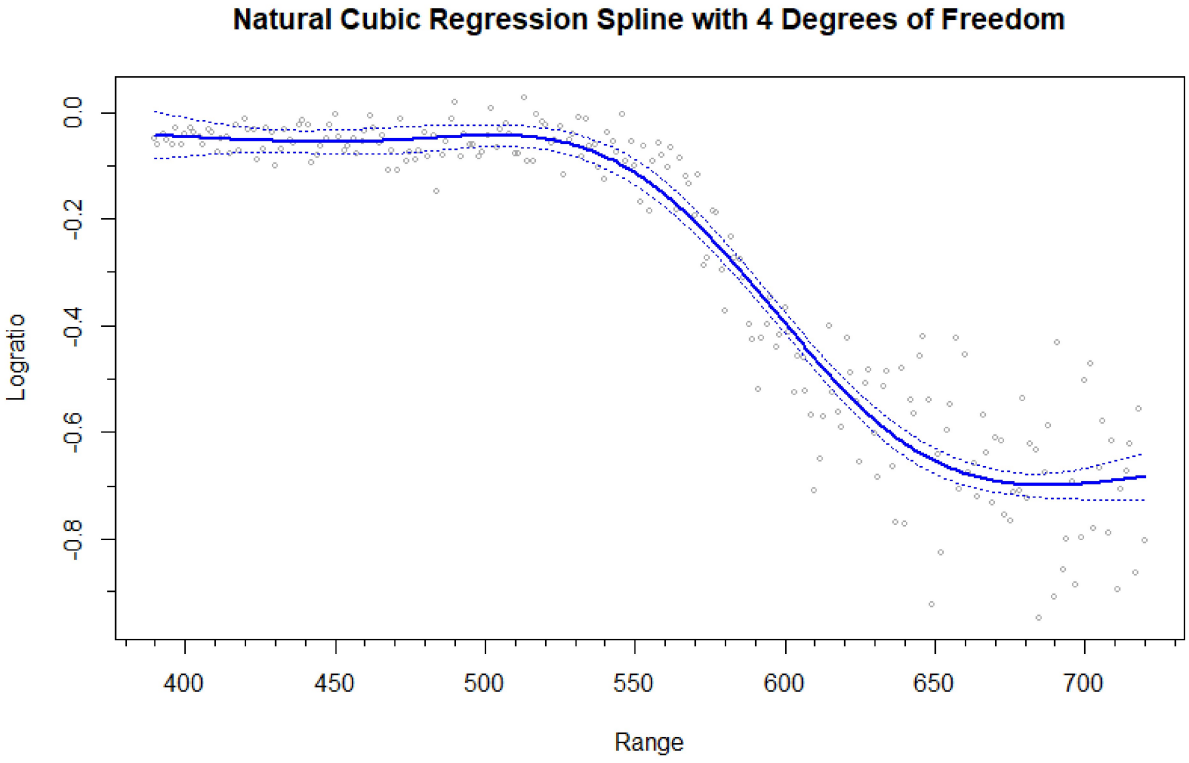
Out of the four fits, the natural cubic spline with 4 degrees of freedom seemed to have the better fit. This was determined both visually and via the MSE calculations of all the methods. In the table below we see that the natural cubic spline with 4 degrees of freedom had the smallest MSE. Also, the scatterplot below compares all of the methods studied in this question and makes it clear that visually the natural cubic spline fits the data more closely than all of the other methods.

Laslty, we plotted the scatterplot of the data with the fit provided by the natural cubic spline superimposed. We also provided the approximate 95% confidence interval in which we note that the margins are the most wide towards the extreme ends of the data and is the most narrow towards the middle of the interval near the intense curvature.

Method	MSE
Polynomial of Degree 4	0.0079
Cubic Regression Spline (4 DF)	0.0076
Natural Cubic Regression (4DF)	0.0070

Table 7: MSE of Each Method Computed via LOOCV


Logratio Vs. Range
Logratio
0.0
-0.2
-0.4
-0.6
-0.8
Polynomial of Degree 4
Cubic Regression
Natural Cubic Regression
Smoothing Spline
400
450
500
550
600
650
700
Range


Natural Cubic Regression Spline with 4 Degrees of Freedom
Logratio
0.0
-0.2
-0.4
-0.6
-0.8
400
450
500
550
600
650
700
Range