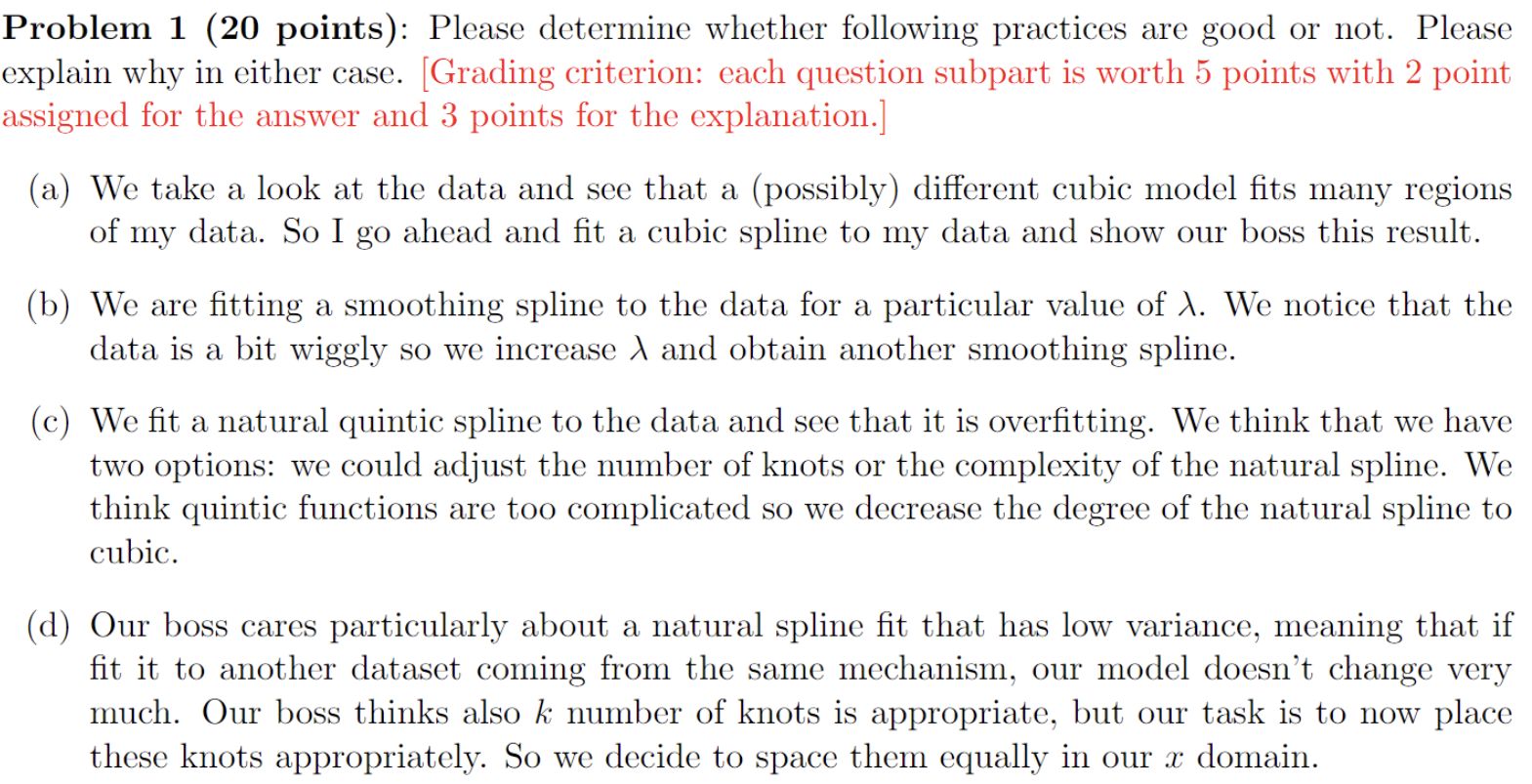
**Data 558: Statistical Machine Learning**

**Spring 2023**

**Homework – 5**

**Arjun Sharma**

**I. Conceptual Questions**



**Solution:**

(a) It may be good practise depending on the situation:

If the boss is technically sound and understands splines, interpretability would not be too much of a concern, otherwise interpretability is an issue.

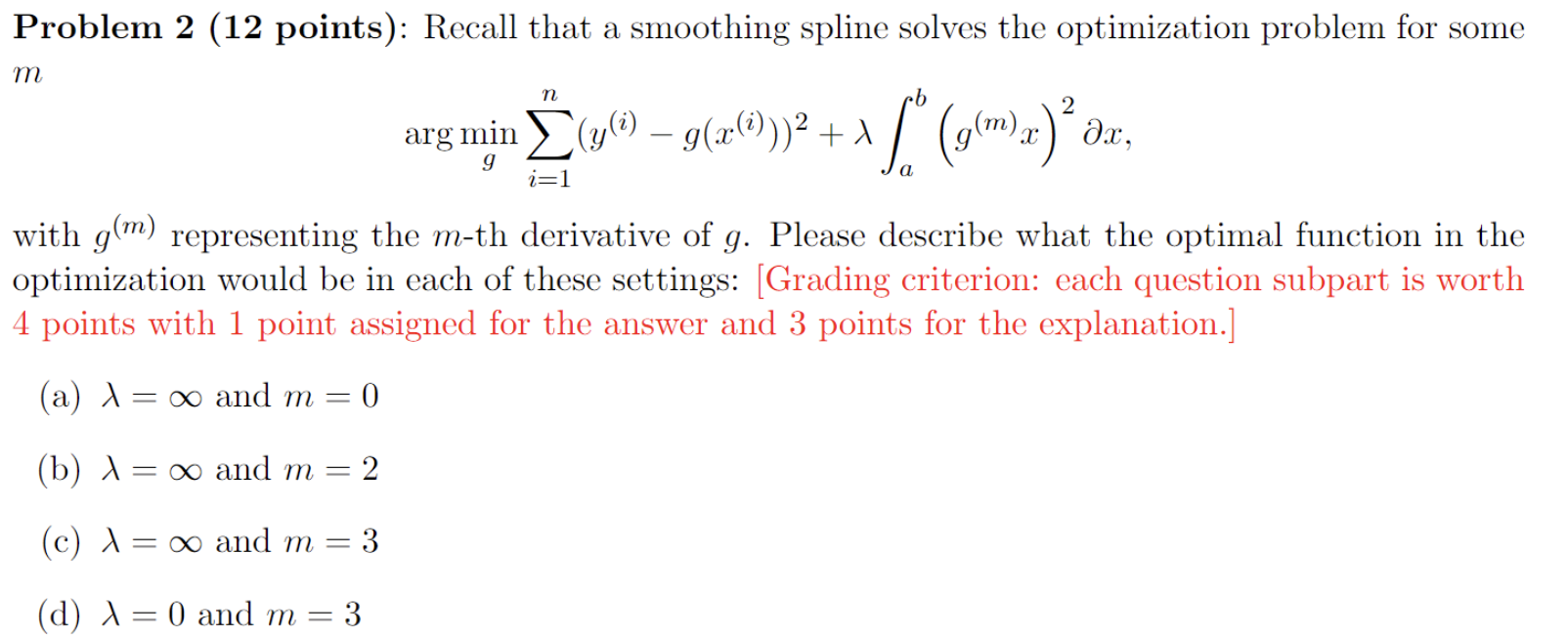
If the underlying relationship between the data is complex, it may be considered good practise to do so, otherwise we risk overfitting, especially with a higher number of knots

(b) This is bad practise. We are changing lambda simply based on the ‘wiggliness’ of the data, and not based on any other criterion. We are penalizing the spline harshly just to alter the shape of the spline. We should use other methods to determine the optimal value for lambda and not simply increase it. An alternative could be k-fold cross validation.

(c) This may be bad practise depending on the kind of data we are working with.

While the measure taken is fair, to reduce overfitting by reducing complexity is the right thing to do via the bias-variance tradeoff. However, we must also account for the regions where our quintic model overfits the data and where it generalizes well. In order to determine the optimal complexity for different regions in the data, we can use alternatives like changing the number of knots and where they lie. This can help us assign a suitable model for each region of data.

(d) This is bad practise. By equally spacing our knots, we have not optimized knot placement for the complexity in the data by the region. Although I would argue that a predetermined number of knots is not good practise, It would be good practise to follow the boss’ instructions but also evaluate the regions for which the placement of a knot would be justified. This can be done through methods like K-fold cross validation as an example. By equally spacing our knots, we are optimizing for the complexity in the data across the region.

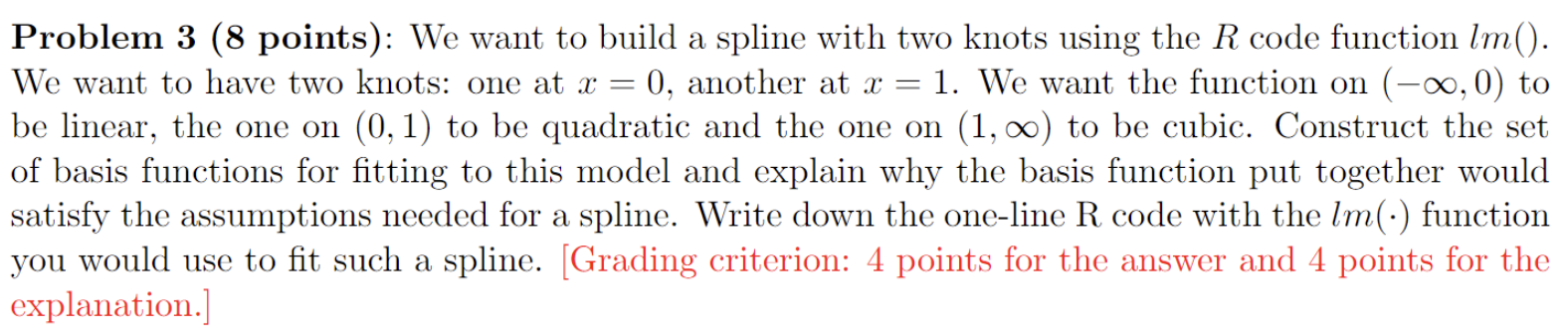


a. λ is infinite, and hence leads to extremely high model complexity irrespective of the value of the g term. This leads to a piecewise constant model that overfits the data in a very extreme manner.

b. λ is infinite but m is 2, hence we have the 2nd derivative of the g term. This means that our optimal function is a model that is piecewise cubic, where a third-degree curve connecting each datapoint, so each datapoint is also a knot.

c. λ is infinite but m is 3, hence we have the 3rd derivative of the g term. Our optimal function is once again a model which is piecewise cubic like in b, where once again, each datapoint is a knot.

d. The λ is 0, hence the penalty does not exist. In this case, the optimal function is just the RSS from the data.



Given that we have 2 knots, we require 3 basis functions. The 3 basis functions are described as:

Function 1: Linear when < 1 - f1(x) = x for x ∈ (−∞, 0), 0 otherwise

Function 2: quadratic when >= 0 and <= 1 - f2(x) = x2 for x ∈ (0, 1), 0 otherwise

Function 3: cubic when > 1 - f3(x) = (x − 1)3 for x ∈ (1, ∞), 0 otherwise

The three functions are continuous for the specified conditions, they also incorporate the knots as in the question. Also, these functions can be described as a piece-wise equation. Hence, we can verify that the assumptions for a spline are satisfied.

The spline can be written in R as follows (y is our target): Spline\_fit <- lm( y ~ f1 + f2 + f3, data = data)



Number of parameters by complexity:

1. Linear: 2 parameters (slope, intercept)

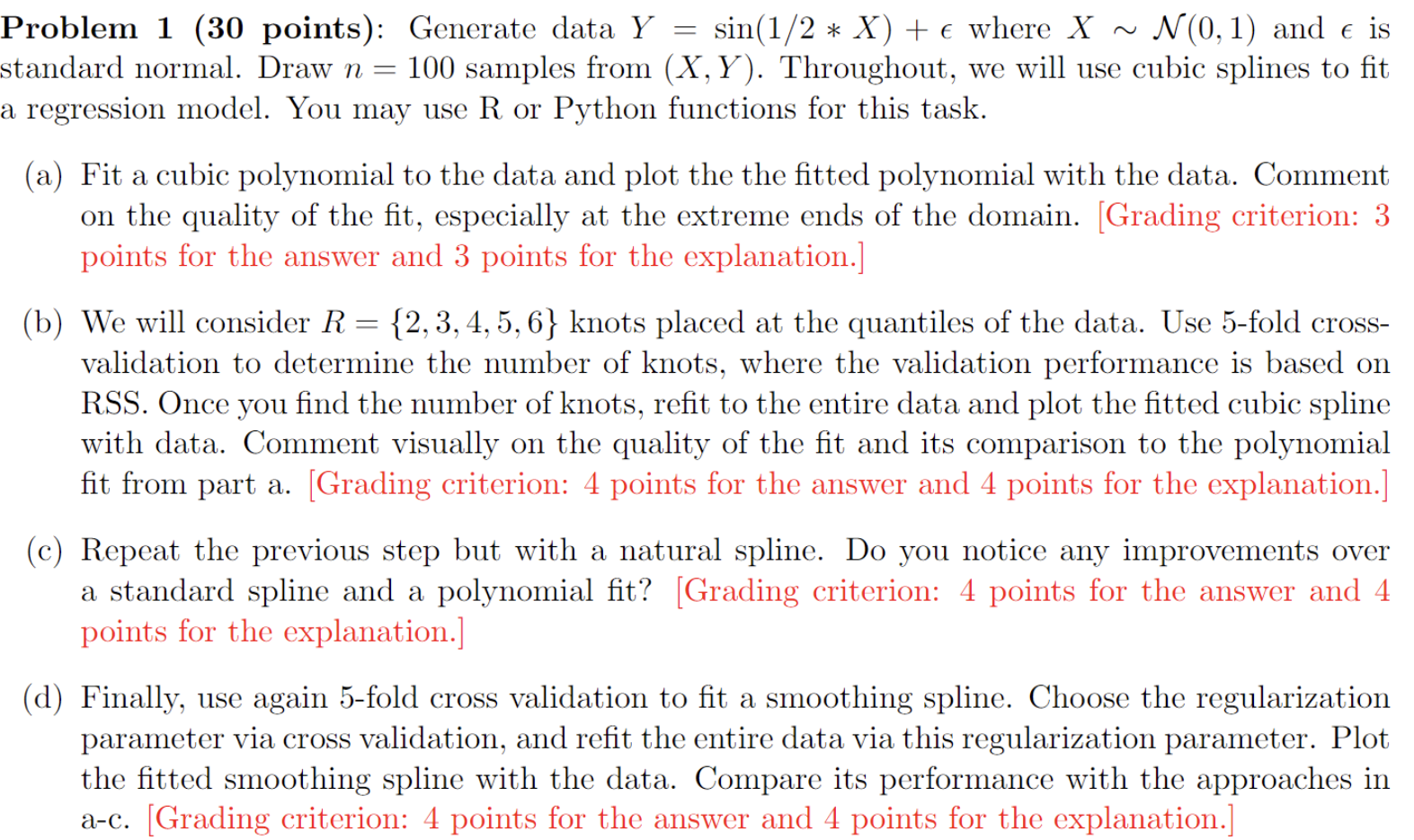
2. Quadratic: 3 parameters (intercept, linear coefficient, quadratic coefficient)

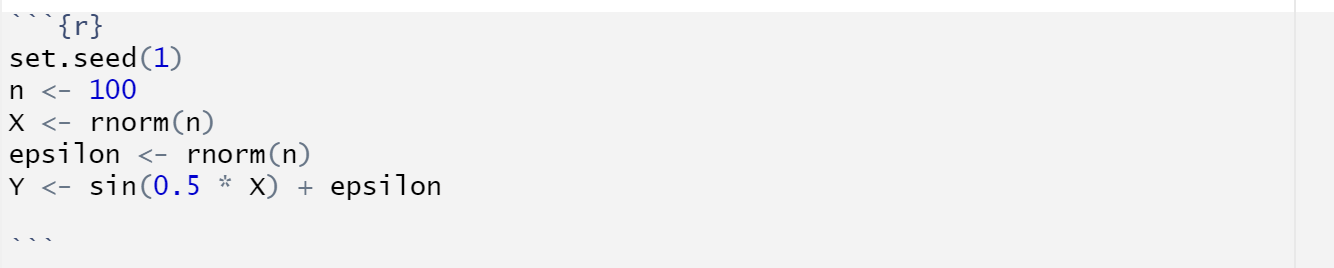
3. Cubic: 4 parameters (intercept, linear, quadratic and cubic coefficients)

Now, we have 4 functions, of which 2 are quadratic, and 1 each is linear and cubic.

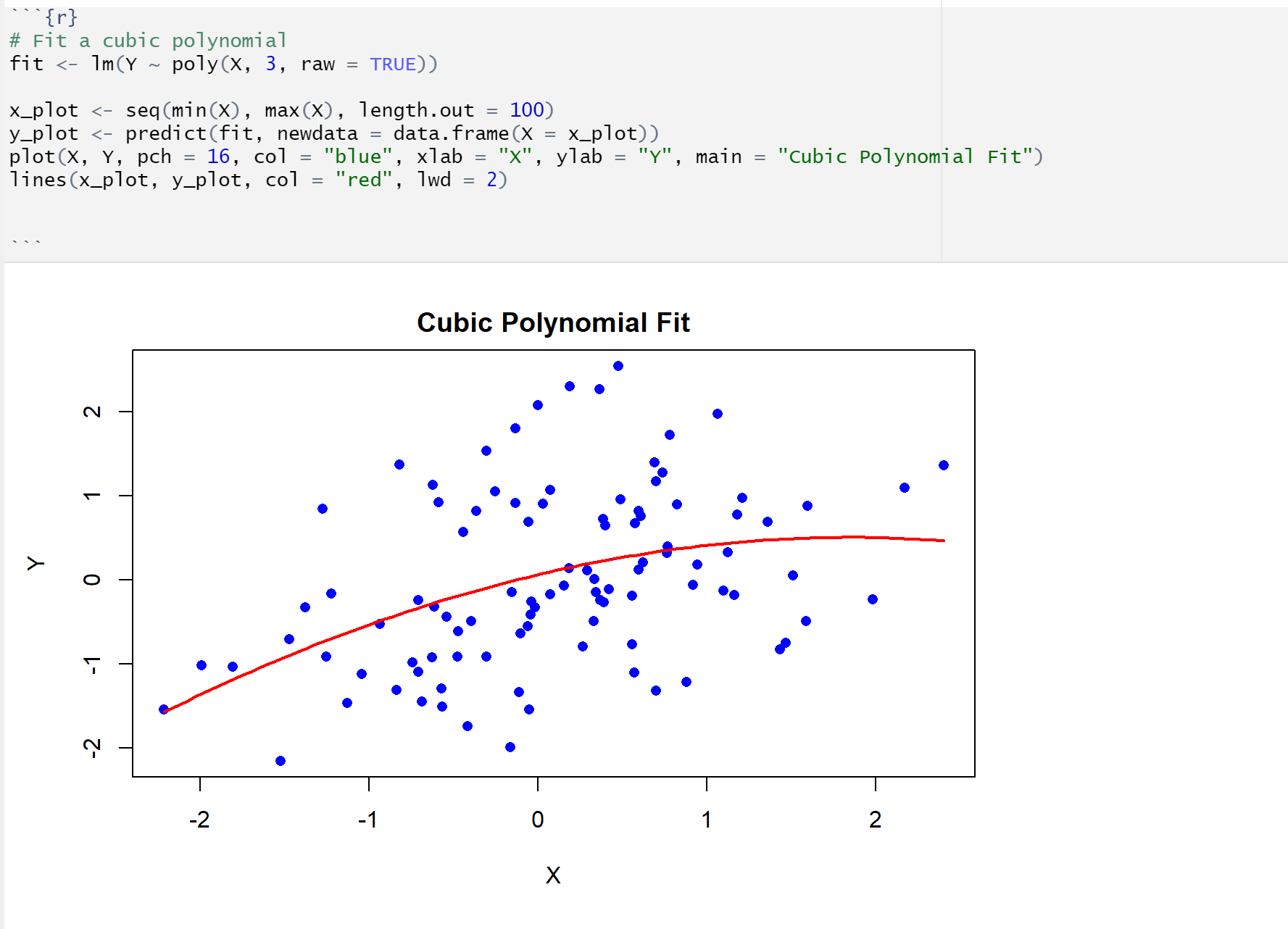
Hence, we have 2+3+3+4 parameters = 12 parameters.

**II. Applied Questions**



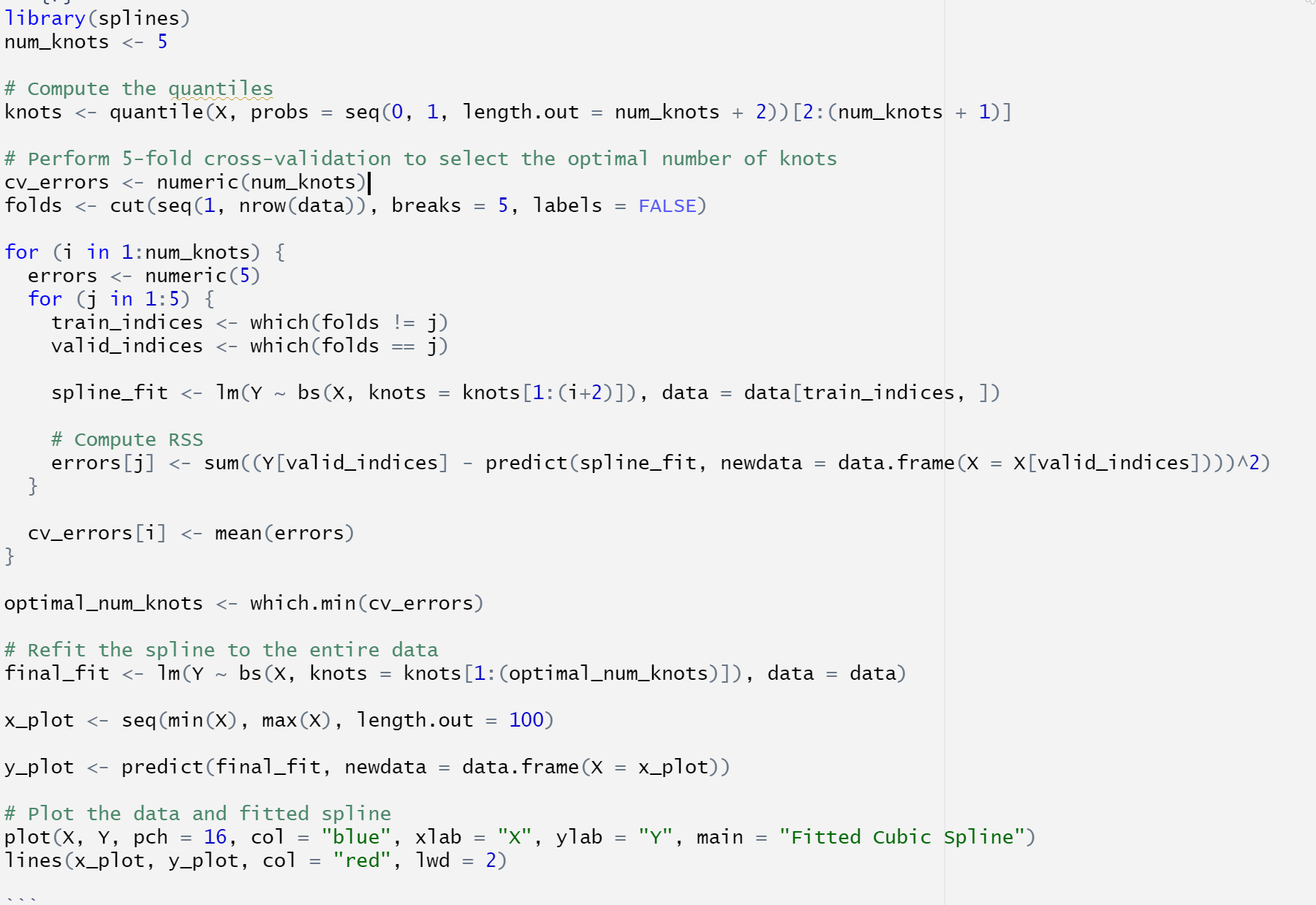


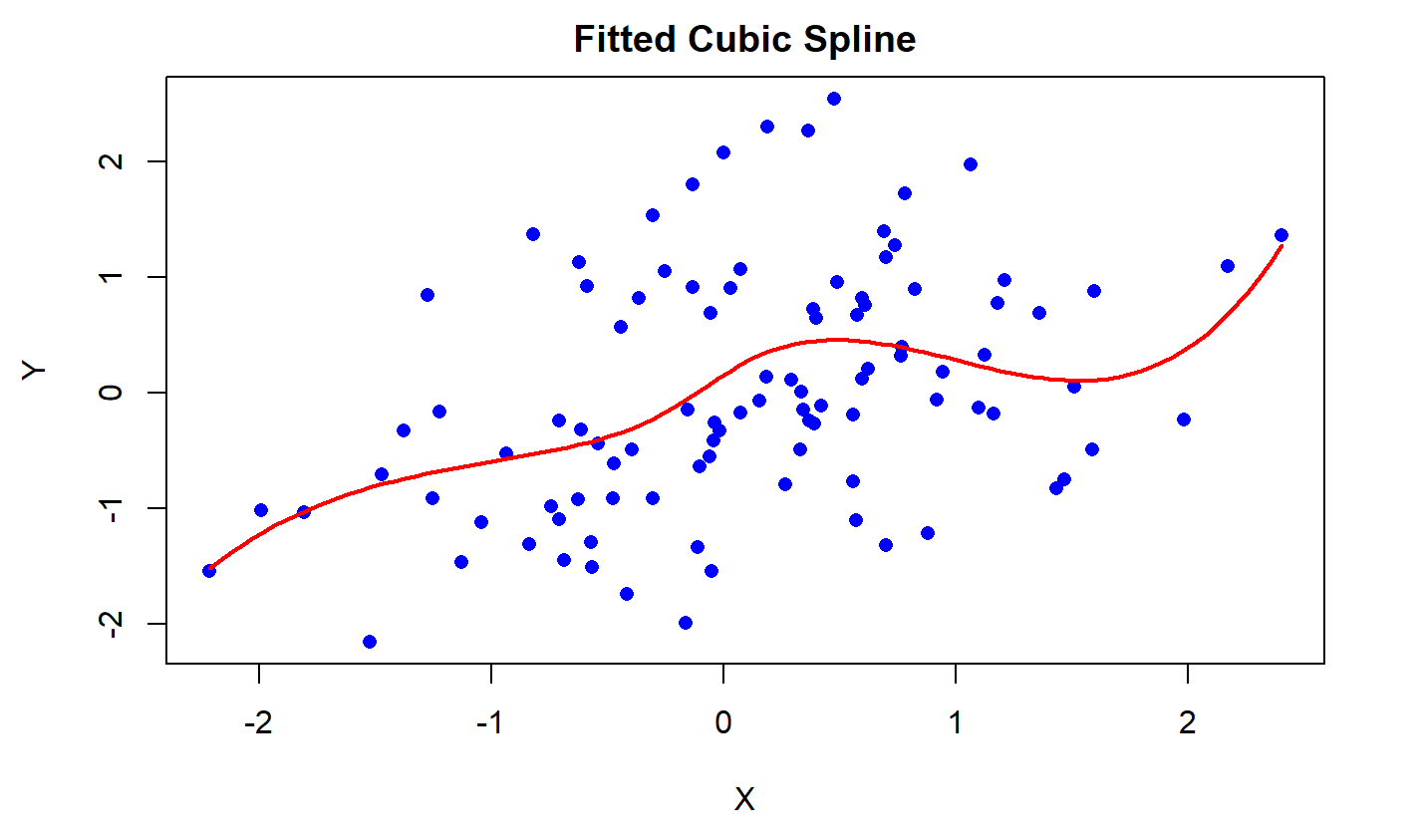
(a)



Even though the function is in an approximate region with the distribution, it is a poor fit because the function foes not fit most of the points.

(b)

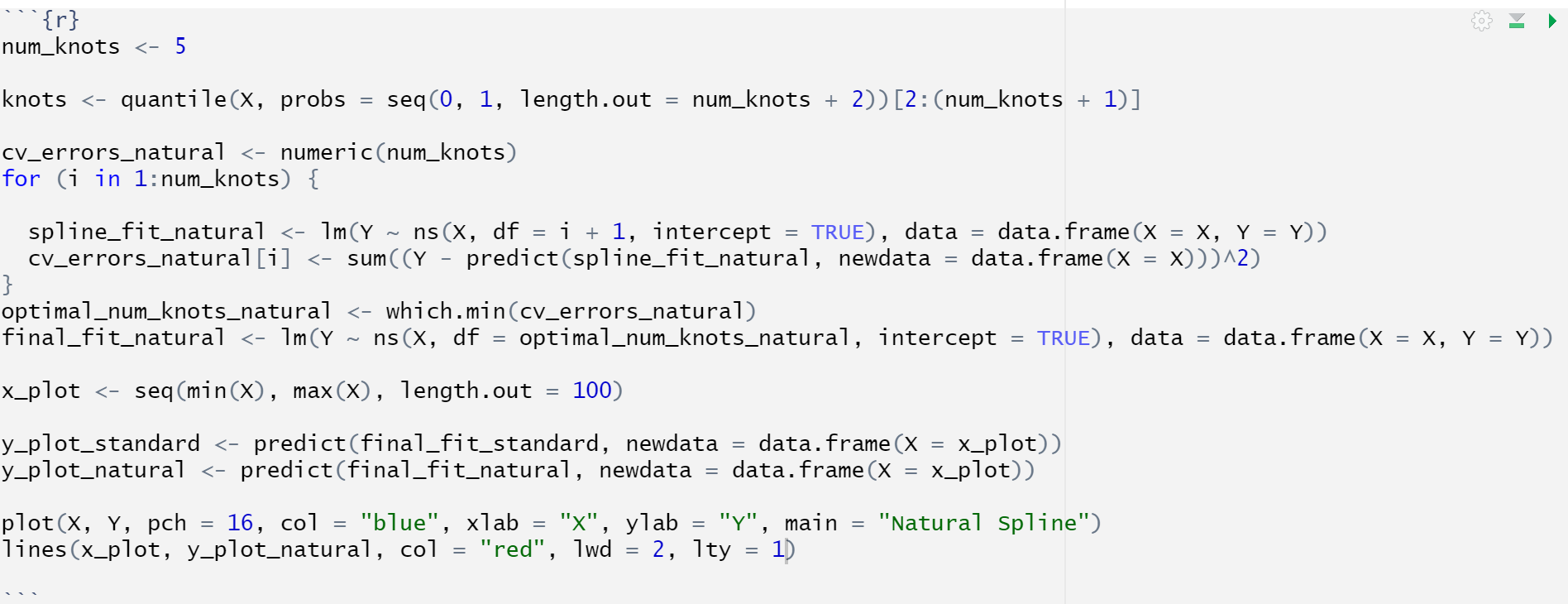


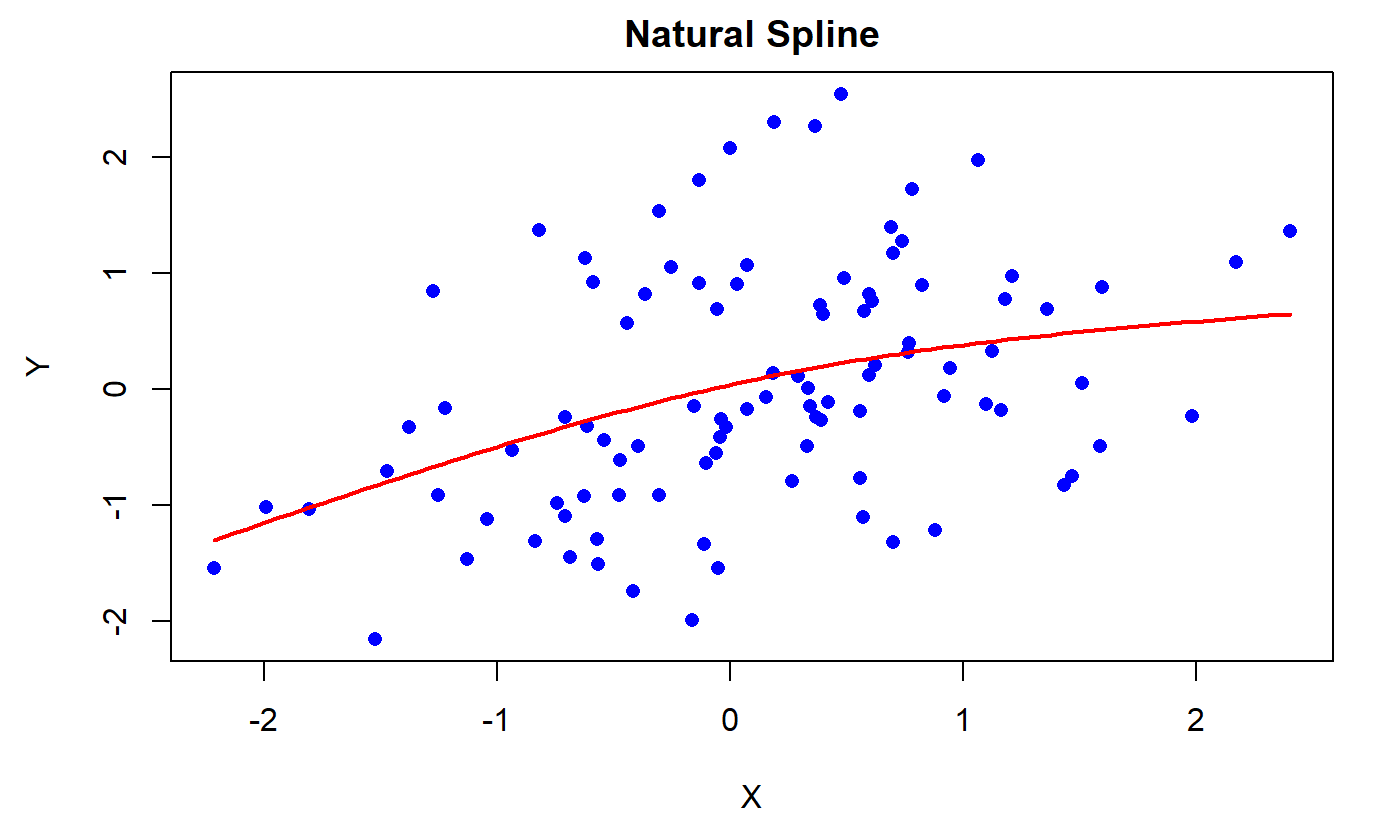




This function still does not generalize well. Although the addition of knots allows for the function to better fit some regions of the data, it still does not fit most of the points.

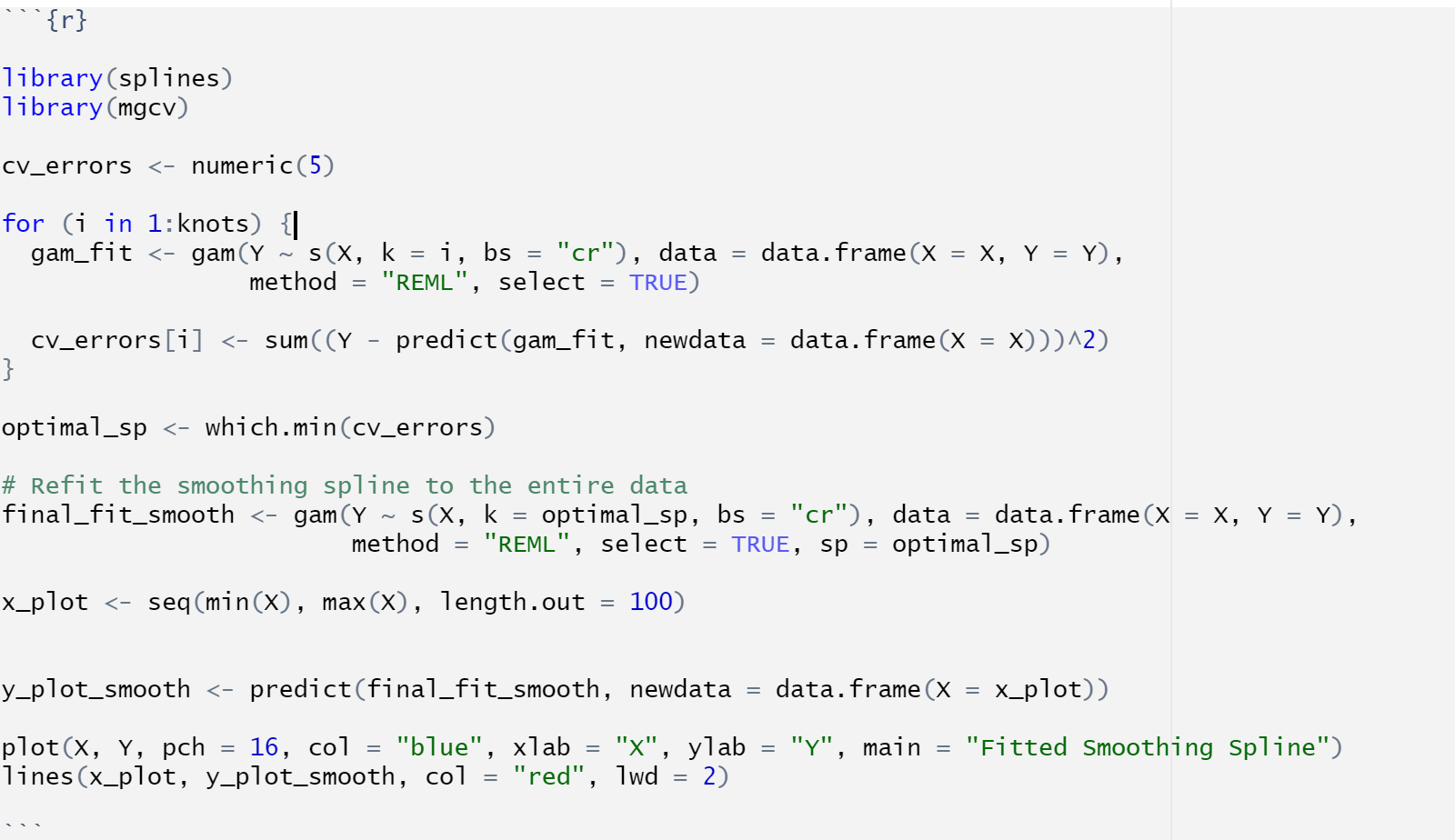
c)





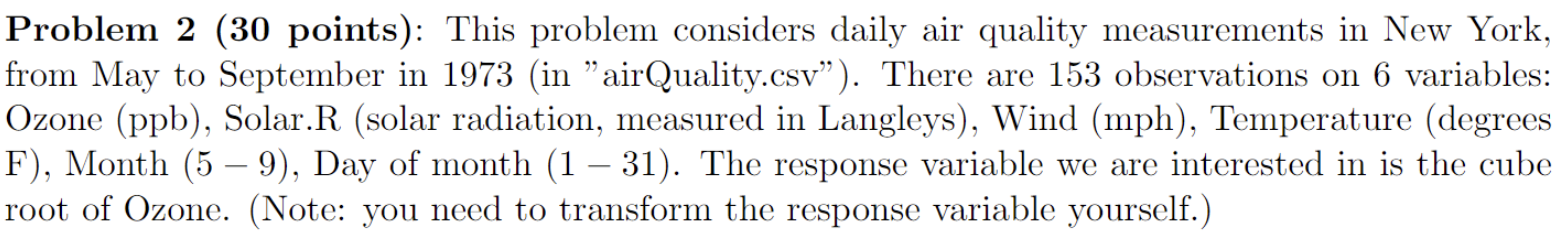
The natural spline does not fit the data better than the previous fits, but it does appear smoother

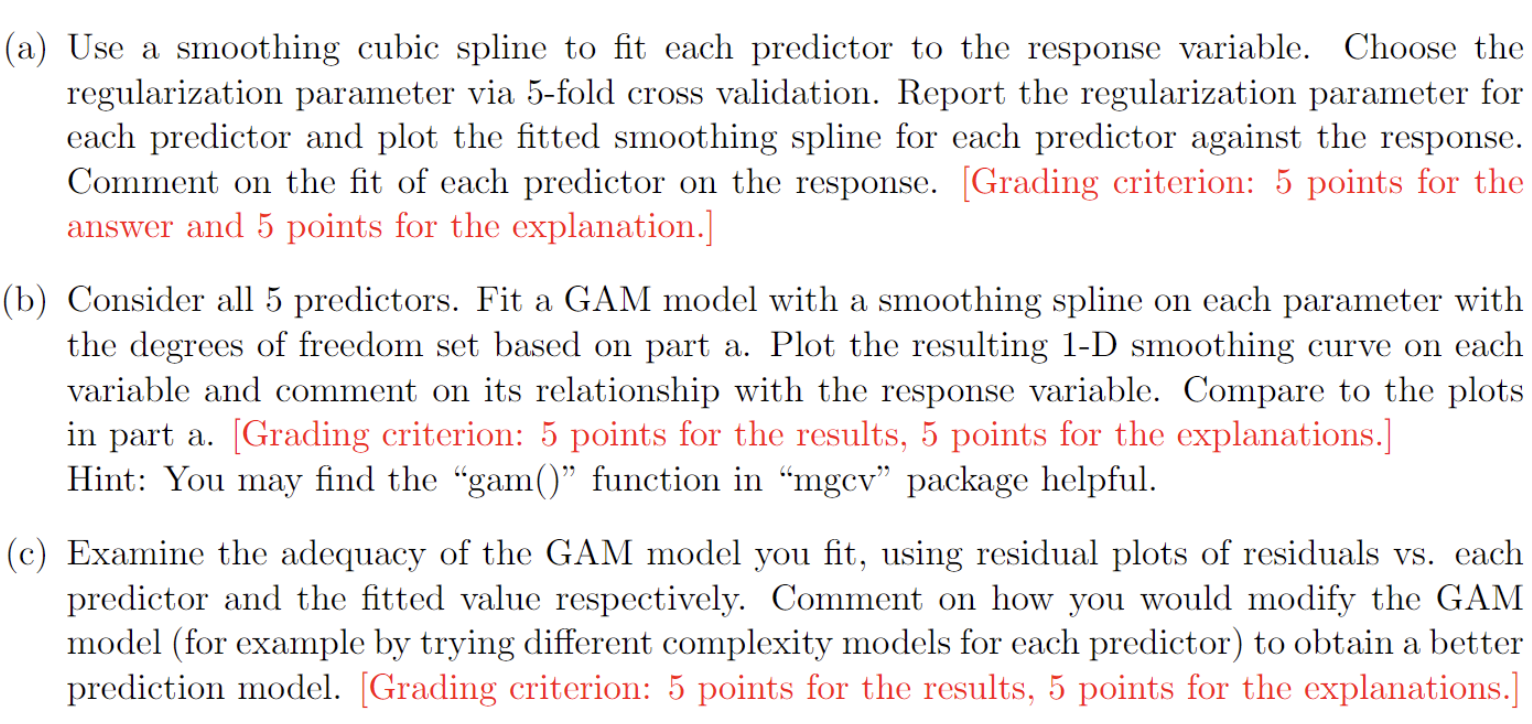
(d)





We see that the smoothened spline tends to fit the data better than the other fits

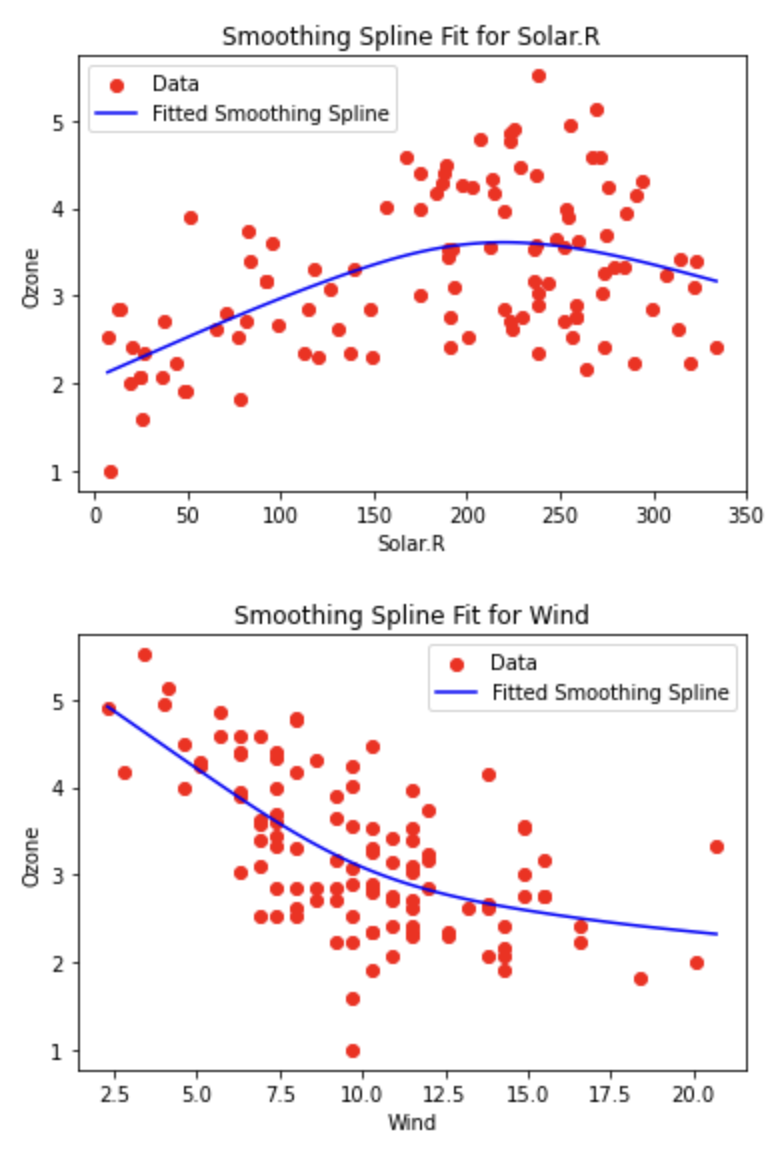


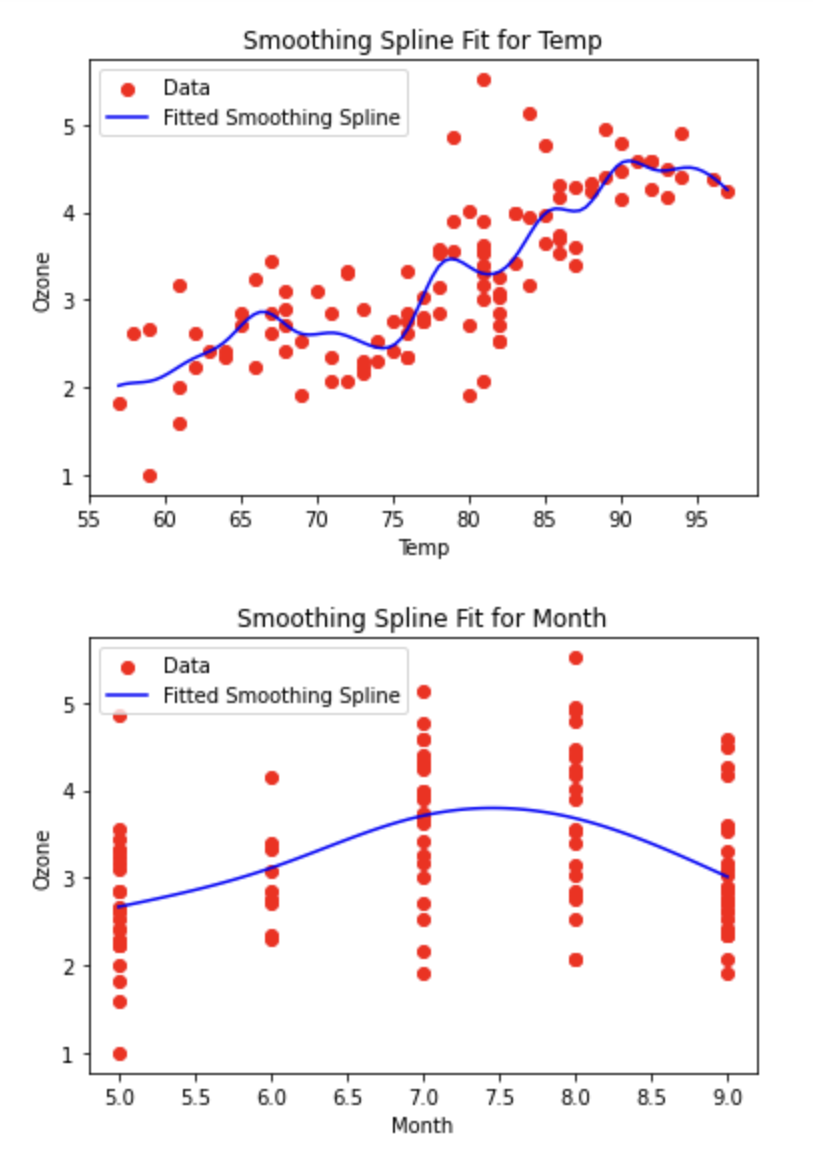


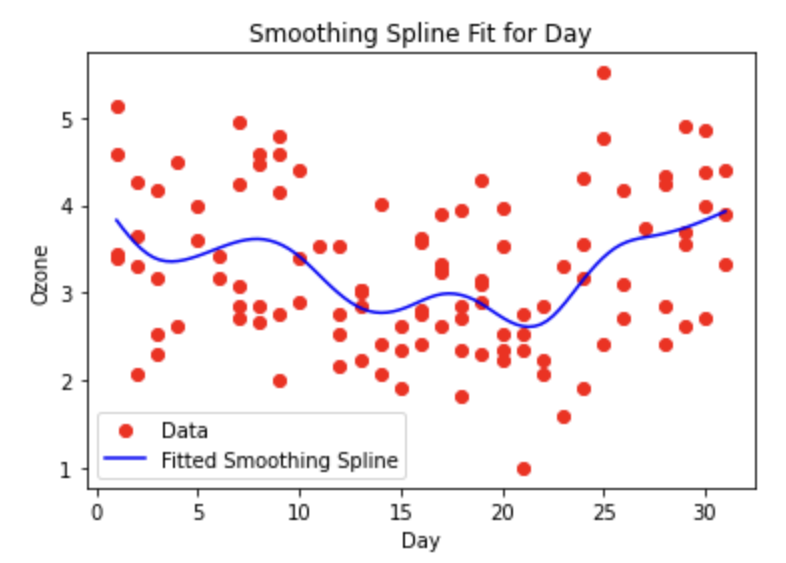
a)

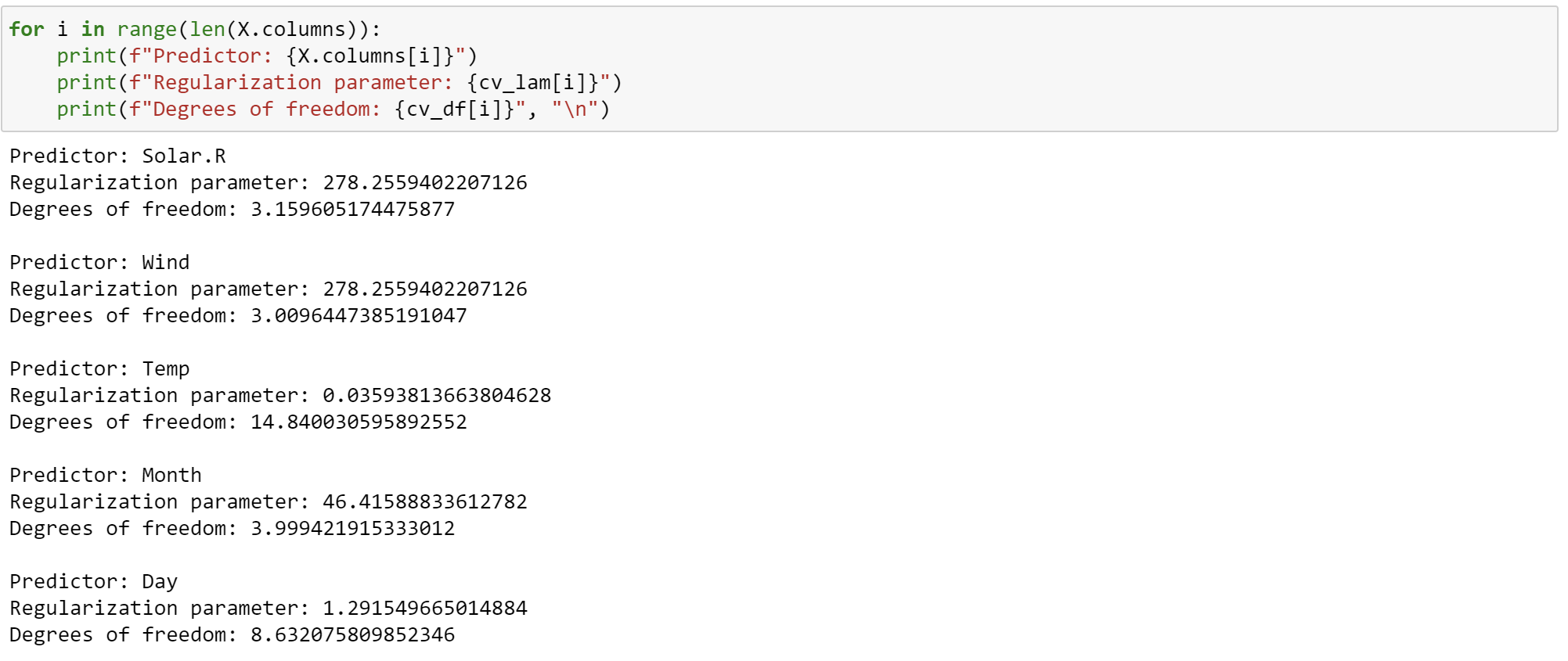










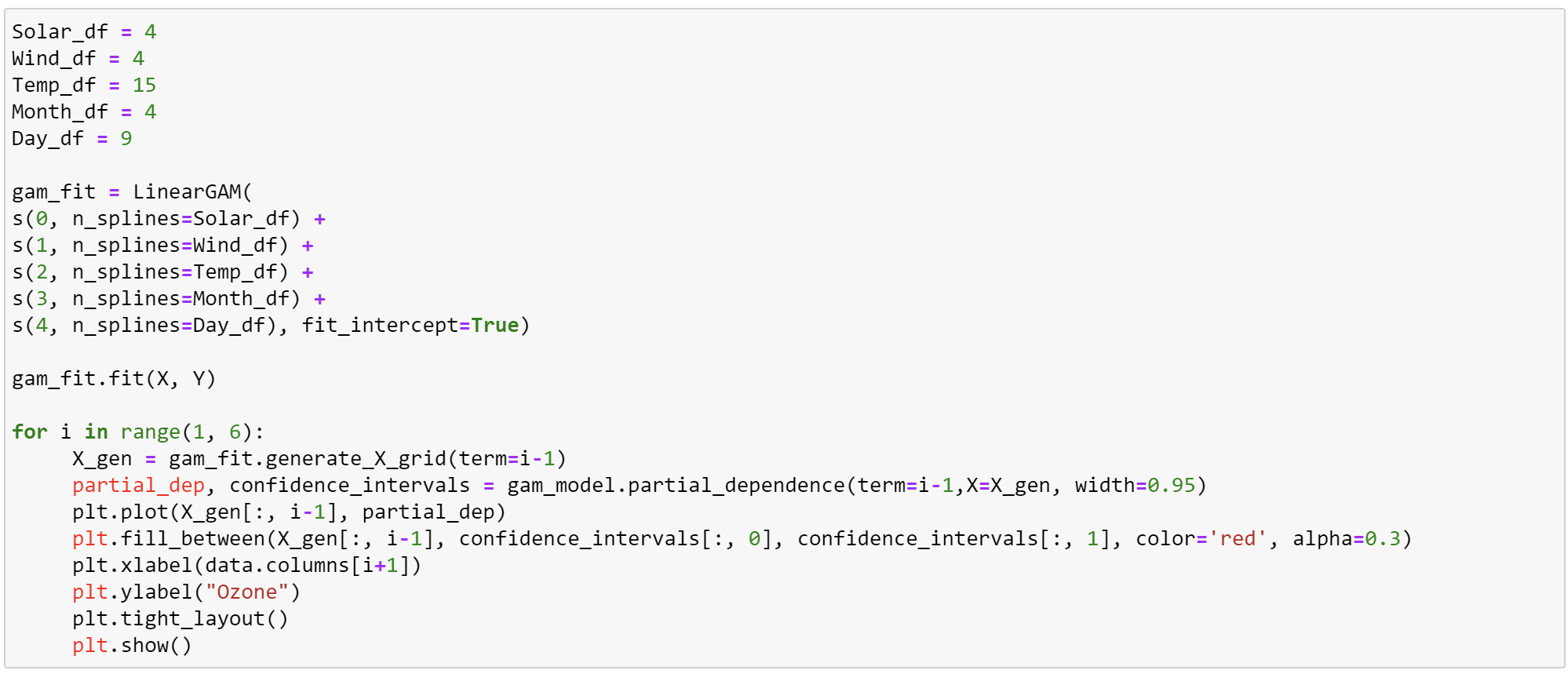


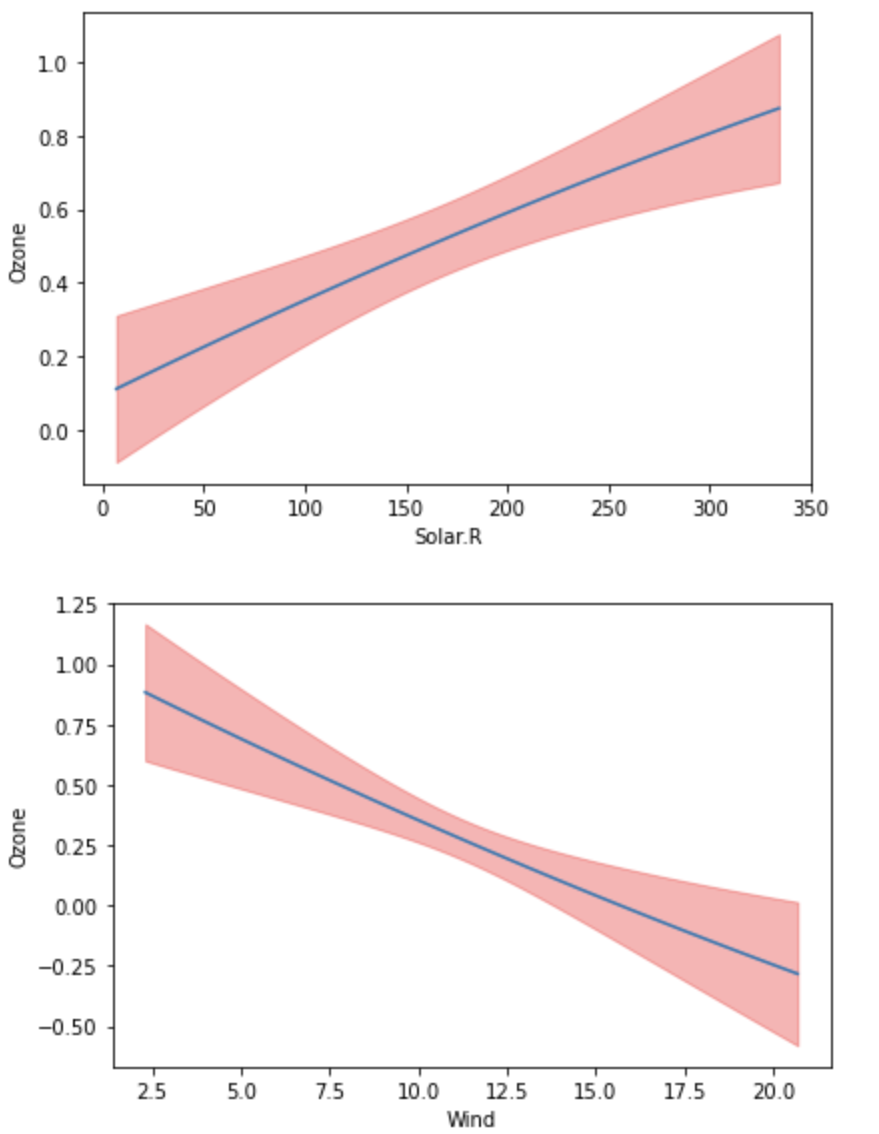
The functions for all predictors represent the trend to an appreciable extent in the data.

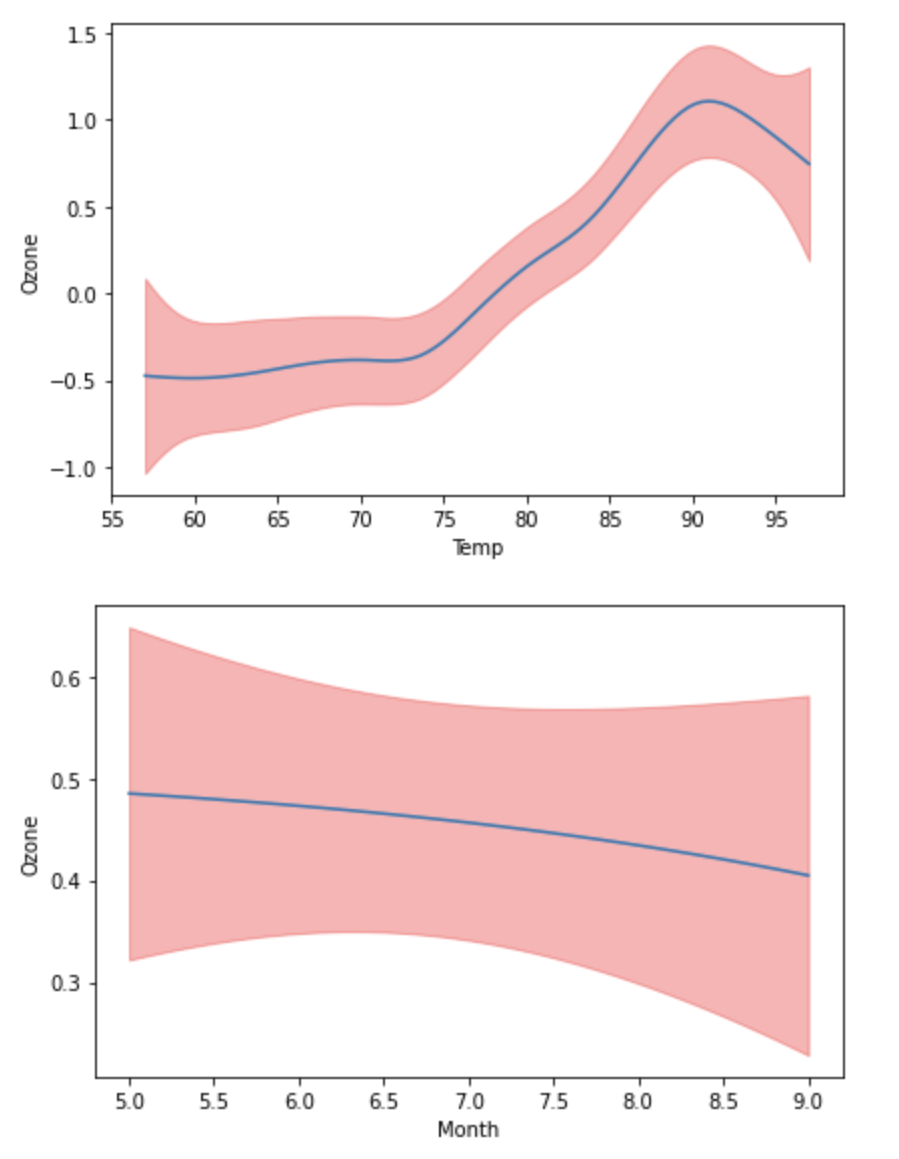
The fit for Temp seems to fit the data well. The same can be said for Wind.

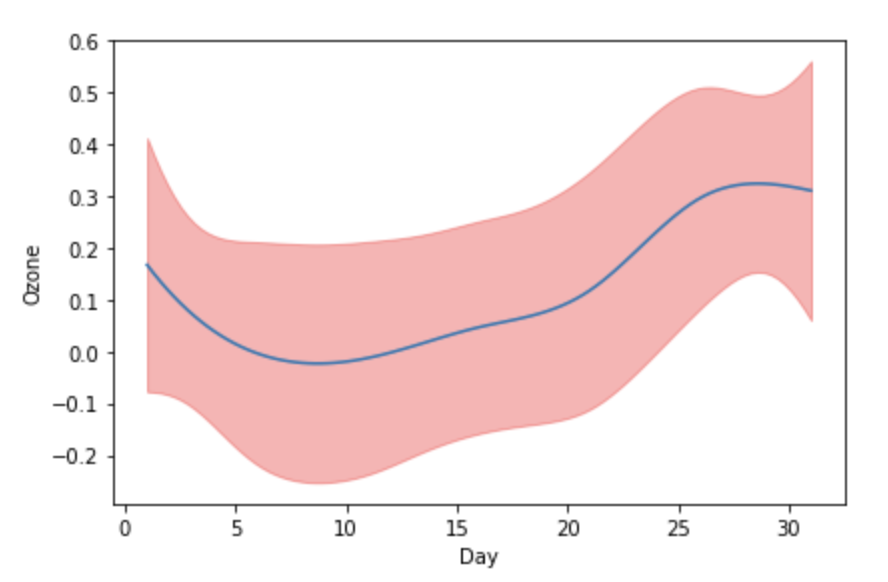
The fit for month, day and Solar.R although representative of the trend in the data, it does not fit the data appropriately.

b)







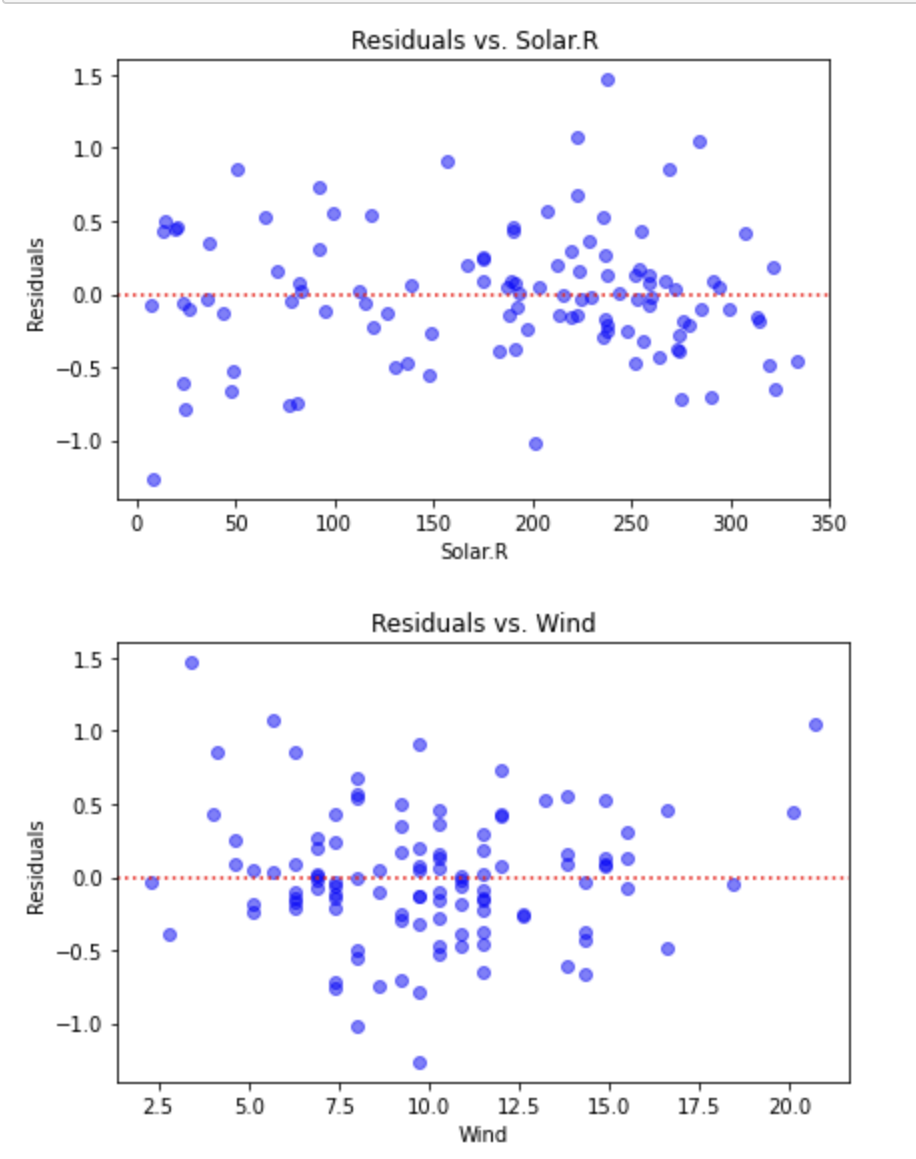


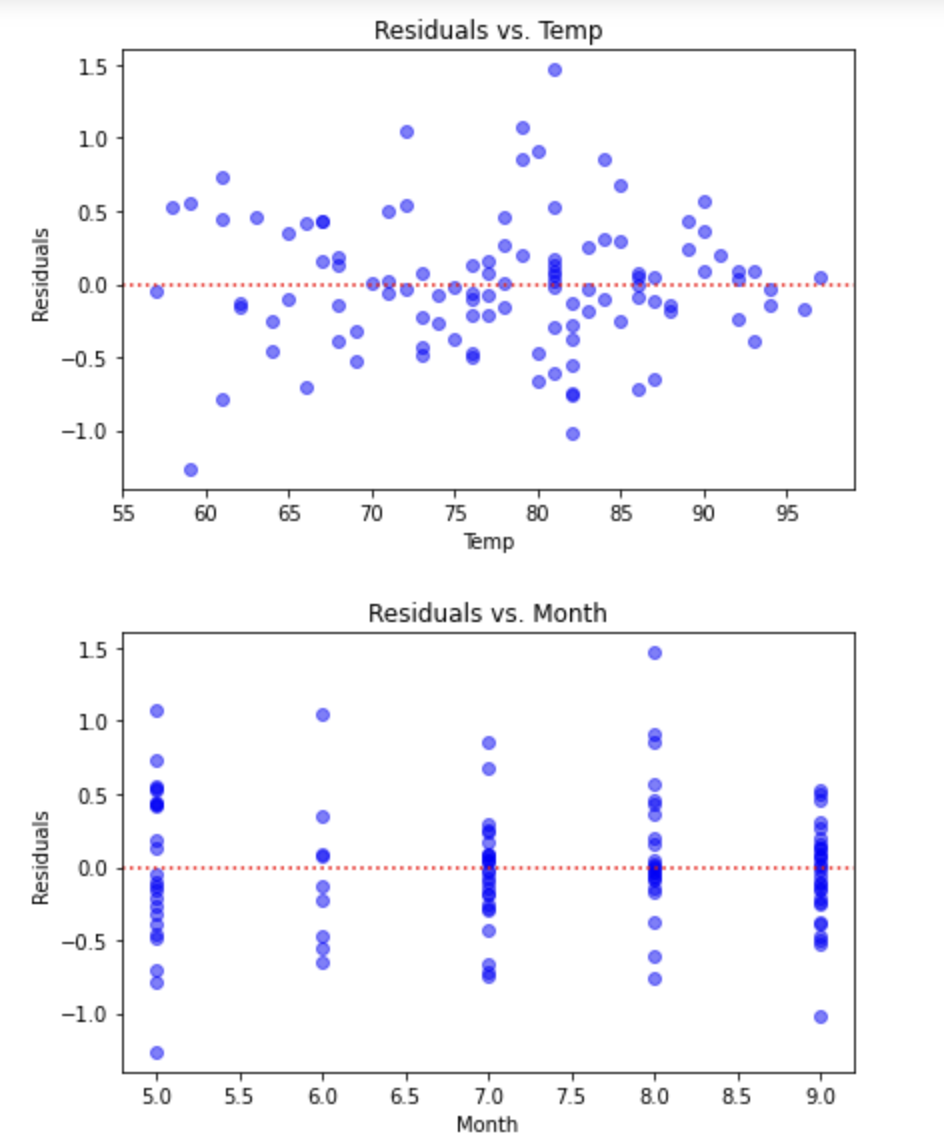
Except for Day, all predictors have wider residuals at the edges and narrower residuals in the middle values of the predictor. Month and Day have much wider ranges as compared to the other predictors.

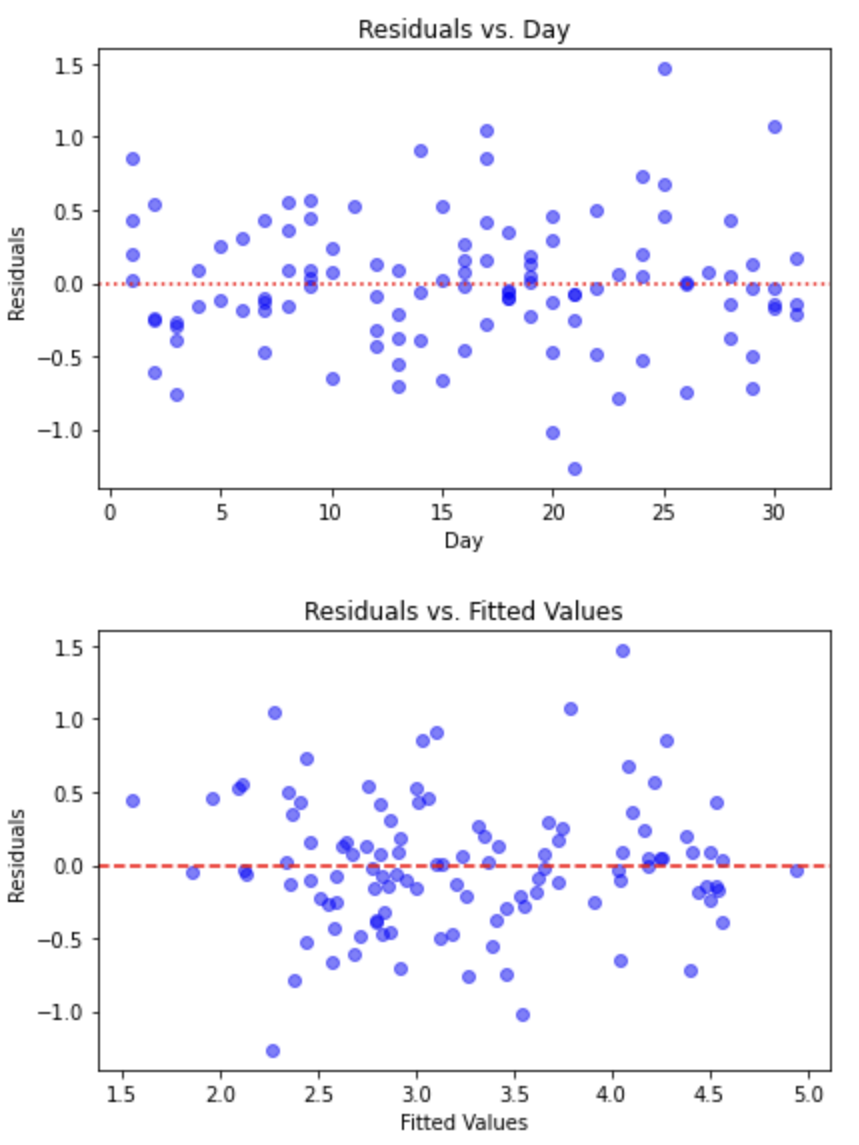
The quality of fit is consistent as in a. However, I would say that Solar.R’s smoothing curve also denotes good fit.

c)









In order to improve model performance, we would require different complexities for each predictor at different regions as well. I notice the following trends in residuals across the predictors which necessitate different complexities in the said conditions:

1. Solar.R has higher residuals at the extremes, lower residuals in the middle.

2. Day has higher residuals for higher values

3. I don’t see any specific trend in the residuals for month

4. Residuals are higher for smaller and moderate values in Temp and Wind.

The residuals vs the fitted values are also relatively higher at the lower and higher values, this suggests the need for varying levels of complexity.