# MA 751: Data Assignment 2

*Benjamin Draves*

*4/2/2019*

## Exercise 1: Ozone Regression Splines

In this exercise, we look to explain the behavior of Ozone concentration as a function of three covariates; radiation, temperature, and wind. We first consider multi-dimensional regression splines with knots $\{\xi_i^{(j)}\}_{i=}^5$ for covariate $j$. To make this development rigourous, let the knots be located at the quantiles at each covariate variable. Moreover, define the vector

$$h(X_j) = \begin{bmatrix} 1 \\ X_j \\ N_1(X_j) \\ N_2(X_j) \\ N_3(X_j) \end{bmatrix}$$

where $N_k(\cdot)$ are given in (5.4). Next, define the parameter vector $\theta_j = (\theta_{j0} \quad \theta_{j1} \dots \theta_{j4}) \in \mathbb{R}^{5\times 1}$. Lastly, for notational convience, for $j = 1, 2, 3$ define

$$h(X_j) = \begin{bmatrix} 1 \\ h_j^*(X_j) \end{bmatrix} \quad \theta_j = \begin{bmatrix} \theta_{0j} \\ \theta_j^* \end{bmatrix}$$

With this, we can rearrange our modeling equation as follows

$$\begin{aligned} Y &= \theta_0 + h_1(X_1)^T\theta_1 + h_2(X_2)^T\theta_2 + h_3(X_3)^T\theta_3 \\ &= (\theta_0 + \theta_{01} + \theta_{02} + \theta_{03}) + (h_1(X_1)^*)^T\theta_1^* + (h_2(X_2)^*)^T\theta_2^* + (h_3(X_3)^*)^T\theta_3^* \\ &= \left( \begin{bmatrix} 1 \\ h_1^*(X_1) \\ h_2^*(X_2) \\ h_3^*(X_3) \end{bmatrix} \right)^T \begin{bmatrix} \theta_0 + \theta_{01} + \theta_{02} + \theta_{03} \\ \theta_1^* \\ \theta_2^* \\ \theta_3^* \end{bmatrix} \end{aligned}$$

therefore, the rows of the design vector are now linearly independent. Incorperating the full training set define the design matrix

$$H = \begin{bmatrix} 1 & h_1^*(X_{11})^T & h_2^*(X_{12})^T & h_3^*(X_{13})^T \\ 1 & h_1^*(X_{21})^T & h_2^*(X_{22})^T & h_3^*(X_{23})^T \\ \vdots & \vdots & \vdots & \vdots \\ 1 & h_1^*(X_{n1})^T & h_2^*(X_{n2})^T & h_3^*(X_{n3})^T \end{bmatrix}$$

Therefore we see by design our design matrix has linearly independent columns and our modeling equation can now be written as $\mathbf{Y} = \mathbf{H}\theta$ where $\theta = (\theta_0 + \theta_{01} + \theta_{02} + \theta_{03} \quad \theta_1^* \quad \theta_2^* \quad \theta_3^*)^T$ Therefore, to minimizie the squared error loss, we arrive at the OLS problem which has closed form solution

$$\hat{\theta} = \underset{\theta \in \mathbb{R}^{13}}{\arg\min} (\mathbf{Y} - \mathbf{H}\theta)^T (\mathbf{Y} - \mathbf{H}\theta) = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{Y}$$

Therefore, our estimate of the Ozone Concentration is given by $\hat{\mathbf{Y}} = \mathbf{H}\hat{\theta} = \mathbf{H}(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{Y}$. Now assuming that $\mathbf{Y} \sim N(\mathbf{H}\theta, \sigma^2\mathbf{I})$ we see this estimate has the property that

$$\begin{aligned} \mathbb{E}(\hat{\theta}|\mathbf{X}) &= \theta \qquad \text{Var}(\theta|\mathbf{X}) = \sigma^2(\mathbf{H}^T\mathbf{H})^{-1} \\ \mathbb{E}(\hat{\mathbf{Y}}|\mathbf{X}) &= \mathbf{H}\theta \qquad \text{Var}(\hat{\mathbf{Y}}|\mathbf{X}) = \sigma^2\mathbf{H}(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T \end{aligned}$$

Having show that $\hat{\mathbf{Y}}$ has these properties, notice that we can write the following

$$\hat{\mathbf{Y}} = \hat{\theta_0}^* \mathbf{1}_n + \mathbf{H}_1(X_1)^*\hat{\theta}_1^* + \mathbf{H}_2(X_2)^*\hat{\theta}_2^* + \mathbf{H}_3(X_3)^*\hat{\theta}_3^*$$

where

$$\mathbf{H}_j(X_j)^* = \begin{bmatrix} (h_j[(X_1)_j]^*)^T \\ (h_j[(X_2)_j]^*)^T \\ \vdots \\ (h_j[(X_n)_j]^*)^T \end{bmatrix} \in \mathbf{R}^{n \times 4}$$

With this expression, we see we can decompose the variance of $\hat{\mathbf{Y}}$ into the variance attributiable to feature vector $X_j$. To be precise, we see that

$$\begin{aligned} \text{Var}_{X_j}(\hat{\mathbf{Y}}) &= \text{Var}(\mathbf{H}_j(X_j)^*\hat{\theta}_j^*) = \mathbf{H}_j(X_j)^*\text{Var}(\mathbf{H}_j(X_j)^*\hat{\theta}_j^*)(\mathbf{H}_j(X_j)^*)^T \\ &= \mathbf{H}_j(X_j)^*\hat{\Sigma}_{jj}(\mathbf{H}_j(X_j)^*)^T \end{aligned}$$

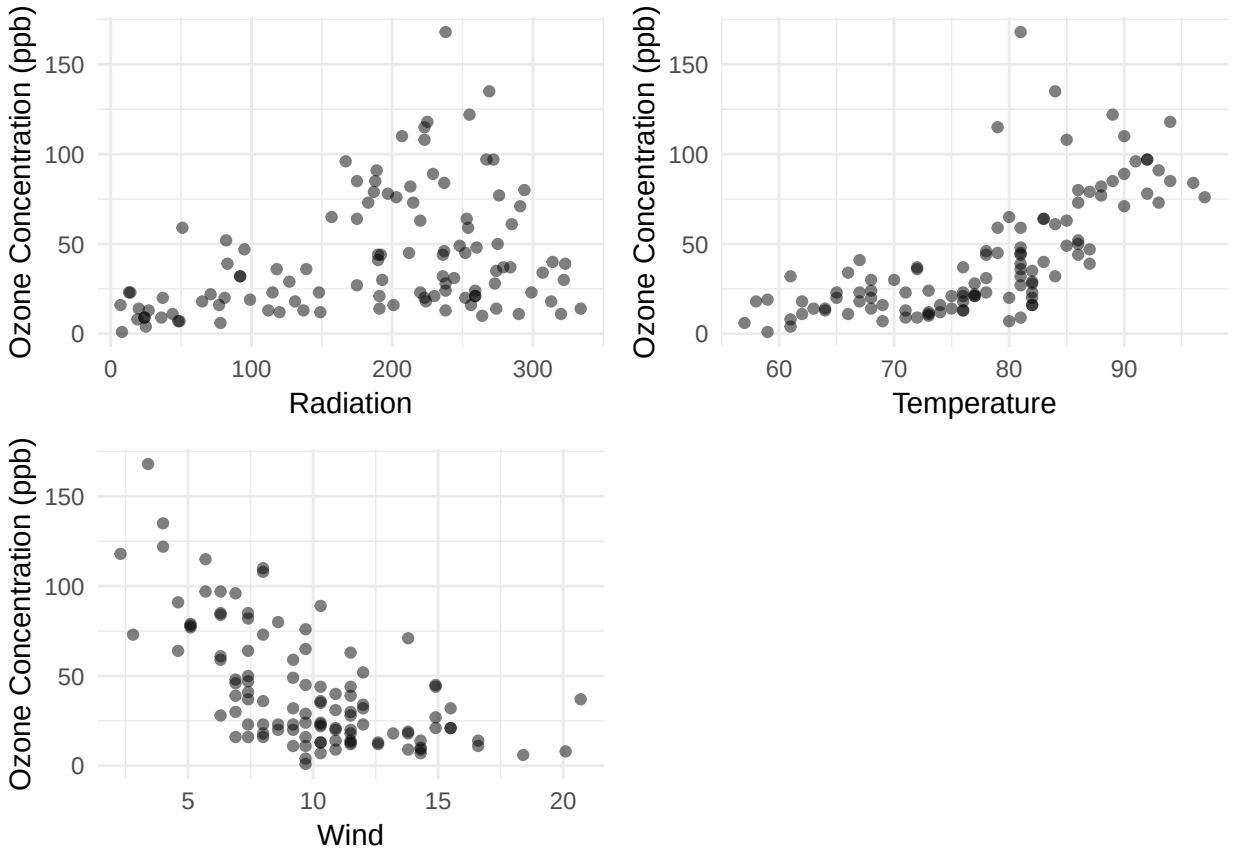
where $\hat{\Sigma}_{jj} \in \mathbb{R}^{4\times4}$ corresponding to the diagnol block of the full variance-covariance matrix $\hat{\sigma}^2\mathbf{H}(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T$ where $\hat{\sigma}^2$ is the estimated variance of the response variable. With these calculations, we see that our regression spline estimator has approximate $1-\alpha$ confidence interval, with respect to feacture $X_j$ given by

$$(\mathbf{h}_j(X_j)^*)^T\hat{\theta}_j^* \pm z_{\alpha/2}\sqrt{(\mathbf{h}_j(X_j)^*)^T\hat{\Sigma}_{jj}\mathbf{h}_j(X_j)^*}$$

In this exercise, we look to implement these natural cubic splies and visualize their fit to the data along with the approximate confidence intervals for each feature; Radiation, Temperature, and Wind. We first begin by visualizing the three marginal relationships.

```
# read in data
dat <- read.table("~/Documents/Work/github/Courses/MA 751/DA2/data/ozone_data.txt",
    header = T)

# visualize each relationship
library(ggplot2)
```

```
p1 <- ggplot(dat, aes(x = radiation, y = ozone)) + geom_point(alpha = 0.5) +
    labs(x = "Radiation", y = "Ozone Concentration (ppb)") +
    theme_minimal()

p2 <- ggplot(dat, aes(x = temperature, y = ozone)) + geom_point(alpha = 0.5) +
    labs(x = "Temperature", y = "Ozone Concentration (ppb)") +
    theme_minimal()

p3 <- ggplot(dat, aes(x = wind, y = ozone)) + geom_point(alpha = 0.5) +
    labs(x = "Wind", y = "Ozone Concentration (ppb)") + theme_minimal()

library(gridExtra)
grid.arrange(p1, p2, p3, ncol = 2, nrow = 2)
```

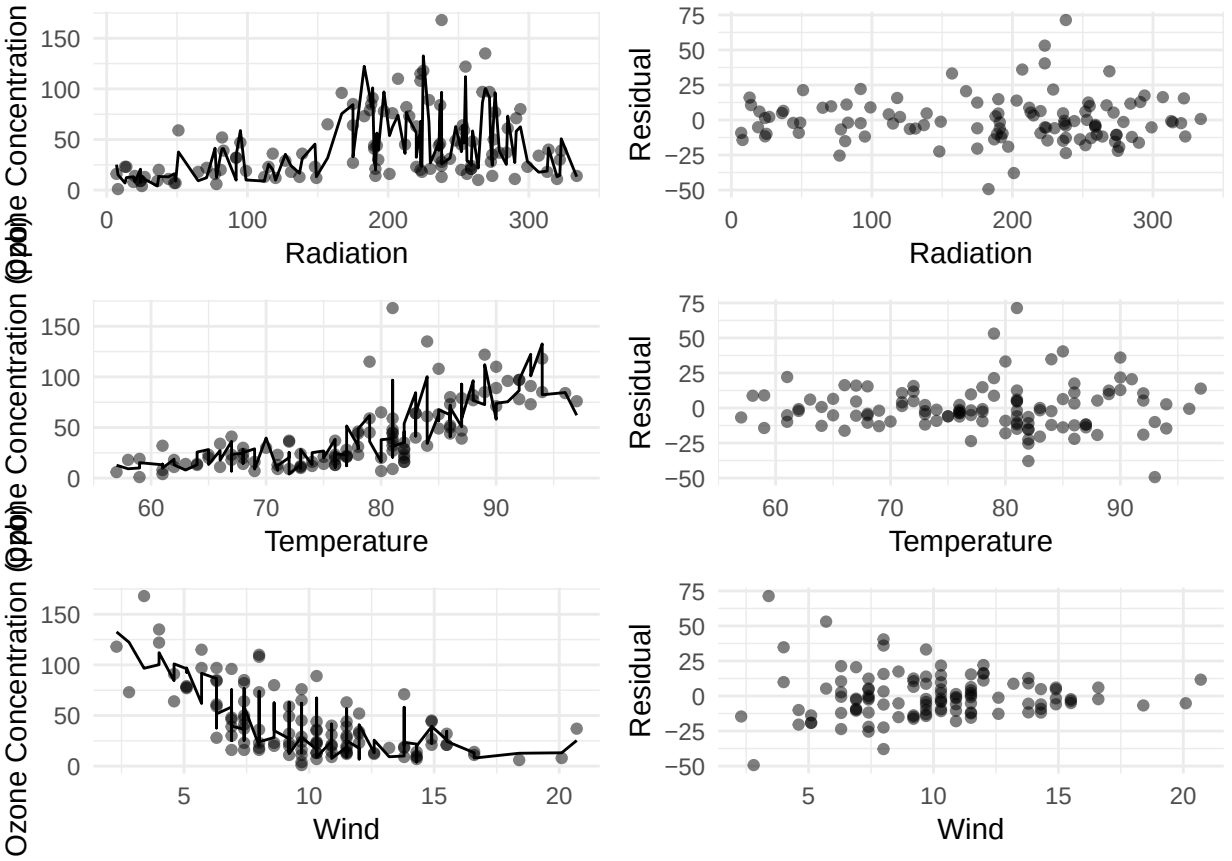



Next, we develop some of the base functionality for the spline regression functions. Once this functionality is built, we then fist the multidimensional natural cubic splines.

```
#-----------------------------------------
# Make feeature vectors given knots functions
#-----------------------------------------
N <- function(x, num, knots) {
    if (num > length(knots)) {
        print("num must be less than # of Knots")
        return(-1)
    }
    if (num == 1)
        return(x)
```

```
    k <- num - 1
    K <- length(knots)
    dk <- (ifelse(x - knots[k] > 0, (x - knots[k])^3, 0) - ifelse(x -
        knots[K] > 0, (x - knots[K])^3, 0))/(knots[K] - knots[k])
    dKm1 <- (ifelse(x - knots[K - 1] > 0, (x - knots[K - 1])^3,
        0) - ifelse(x - knots[K] > 0, (x - knots[K])^3, 0))/(knots[K] -
        knots[K - 1])

    return(dk - dKm1)
}
h <- function(x, knots) {
    vec <- numeric(length(knots) - 1)
    for (i in 1:length(vec)) vec[i] <- N(x, i, knots)
    return(vec)
}

# define knots at quantiles of features
knot_radiation <- unname(quantile(dat$radiation))
knot_temperature <- unname(quantile(dat$temperature))
knot_wind <- unname(quantile(dat$wind))

# build dataframe for each feature
H_radiation <- t(sapply(dat$radiation, FUN = function(x) h(x,
    knot_radiation)))
H_temperature <- t(sapply(dat$temperature, FUN = function(x) h(x,
    knot_temperature)))
H_wind <- t(sapply(dat$wind, FUN = function(x) h(x, knot_wind)))

# join dataframes
H <- cbind(rep(1, nrow(dat)), H_radiation, H_temperature, H_wind)


#-------------------------------------------
# Calculate Regression Spline Estimate from OLS Solution
#-------------------------------------------


# calculate regression spline estimate theta_hat <- solve(H)
# %*% crosspod(H, dat$ozone) stable matrix inversion
v <- crossprod(H, dat$ozone)
C <- chol(crossprod(H))
theta_hat <- backsolve(C, backsolve(C, v, trans = TRUE))


#-------------------------------------------
# Visualize Fit + Residuals
#-------------------------------------------

# Visualize results
library(reshape2)
df_radiation <- data.frame(Radiation = dat$radiation, Observed = dat$ozone,
    Predicted = H %*% theta_hat)
```

```r
p4 <- ggplot(df_radiation) + geom_point(alpha = 0.5, aes(x = Radiation,
    y = Observed)) + geom_line(aes(x = Radiation, y = Predicted)) +
    labs(x = "Radiation", y = "Ozone Concentration (ppb)") +
    theme_minimal()
p4.5 <- ggplot(df_radiation) + geom_point(alpha = 0.5, aes(x = Radiation,
    y = Observed - Predicted)) + labs(x = "Radiation", y = "Residual") +
    theme_minimal()


df_temperature <- data.frame(Temperature = dat$temperature, Observed = dat$ozone,
    Predicted = H %*% theta_hat)
p5 <- ggplot(df_temperature) + geom_point(alpha = 0.5, aes(x = Temperature,
    y = Observed)) + geom_line(aes(x = Temperature, y = Predicted)) +
    labs(x = "Temperature", y = "Ozone Concentration (ppb)") +
    theme_minimal()
p5.5 <- ggplot(df_temperature) + geom_point(alpha = 0.5, aes(x = Temperature,
    y = Observed - Predicted)) + labs(x = "Temperature", y = "Residual") +
    theme_minimal()


df_wind <- data.frame(Wind = dat$wind, Observed = dat$ozone,
    Predicted = H %*% theta_hat)
p6 <- ggplot(df_wind) + geom_point(alpha = 0.5, aes(x = Wind,
    y = Observed)) + geom_line(aes(x = Wind, y = Predicted)) +
    labs(x = "Wind", y = "Ozone Concentration (ppb)") + theme_minimal()

p6.5 <- ggplot(df_wind) + geom_point(alpha = 0.5, aes(x = Wind,
    y = (Observed - Predicted))) + labs(x = "Wind", y = "Residual") +
    theme_minimal()

grid.arrange(p4, p4.5, p5, p5.5, p6, p6.5, ncol = 2, nrow = 3)
```

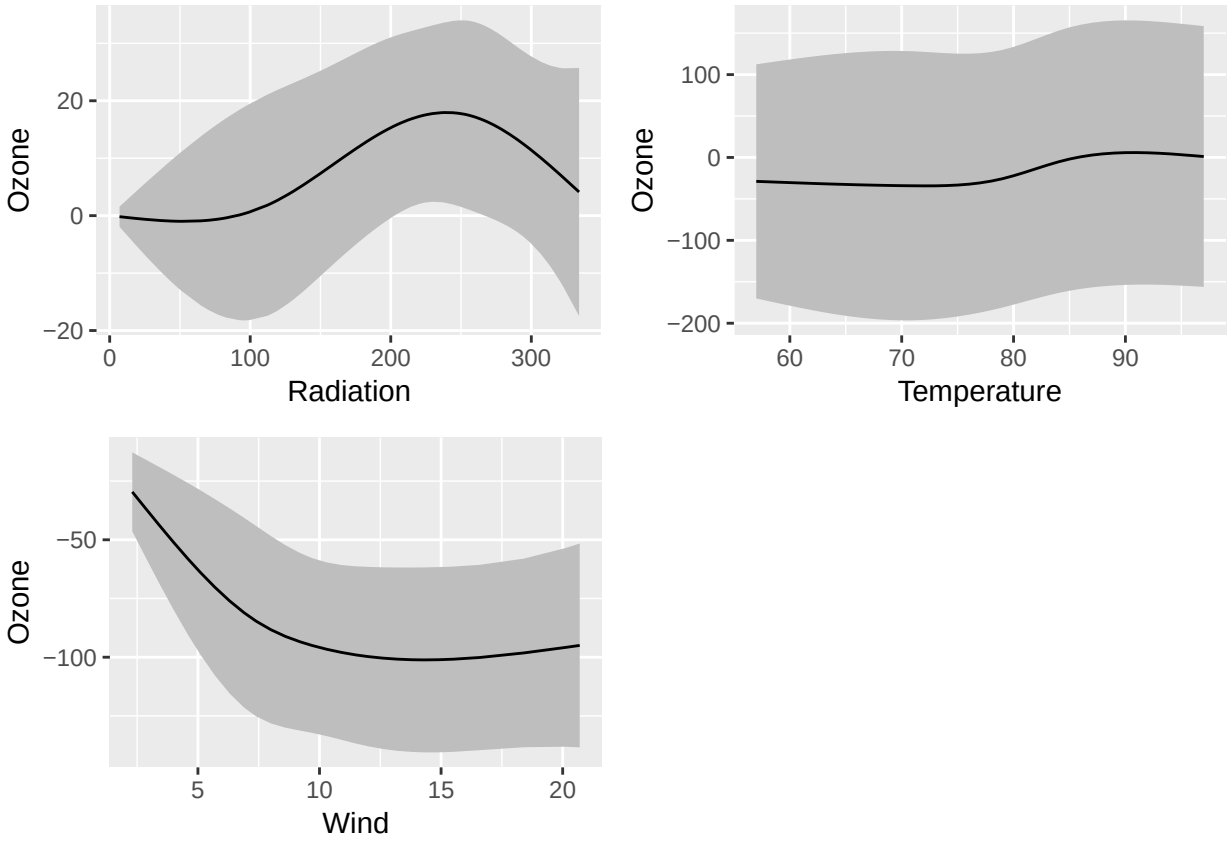



It appears that our multidimensiona spline function is doing quite well at explaining the general trend of the data. It appears that this function is overfitting these trends but it is difficult to decipher as this multidimensional spline is only visualized in one dimension at a time. In addition to these plots, we include a residual plot for each feature.

Next, we look to include the variance of this estimate in each of these figures by plotting the pointwise 95% confidence interval using the confidence interval developed above

```
library(dplyr)
#---------------------------------------
# Plot estimates with Approximate CIs
#---------------------------------------

# Get variance estimate
Sigma.inv <- as.numeric(crossprod(dat$ozone - H %*% theta_hat))/(nrow(dat) -
    ncol(H)) * solve(crossprod(H))

# calculate interval widths
rad.inter <- sqrt(diag(H_radiation %*% Sigma.inv[2:5, 2:5] %*%
    t(H_radiation)))
temp.inter <- sqrt(diag(H_temperature %*% Sigma.inv[6:9, 6:9] %*%
    t(H_temperature)))
wind.inter <- sqrt(diag(H_wind %*% Sigma.inv[10:13, 10:13] %*%
    t(H_wind)))

# define plotting dataframes
df_rad <- data.frame(Radiation = dat$radiation, Ozone = H_radiation %*%
```

```
    theta_hat[2:5], lower = H_radiation %*% theta_hat[2:5] -
    2 * rad.inter, upper = H_radiation %*% theta_hat[2:5] + 2 *
    rad.inter)
df_temp <- data.frame(Temperature = dat$temperature, Ozone = H_temperature %*%
    theta_hat[6:9], lower = H_temperature %*% theta_hat[6:9] -
    2 * temp.inter, upper = H_temperature %*% theta_hat[6:9] +
    2 * temp.inter)
df_wind <- data.frame(Wind = dat$wind, Ozone = H_wind %*% theta_hat[10:13],
    lower = H_wind %*% theta_hat[10:13] - 2 * wind.inter, upper = H_wind %*%
        theta_hat[10:13] + 2 * wind.inter)

p7 <- ggplot(df_rad, aes(x = Radiation, y = Ozone)) + geom_ribbon(aes(ymin = lower,
    ymax = upper), fill = "grey") + geom_line(aes(y = Ozone))

p8 <- ggplot(df_temp, aes(x = Temperature, y = Ozone)) + geom_ribbon(aes(ymin = lower,
    ymax = upper), fill = "grey") + geom_line(aes(y = Ozone))

p9 <- ggplot(df_wind, aes(x = Wind, y = Ozone)) + geom_ribbon(aes(ymin = lower,
    ymax = upper), fill = "grey") + geom_line(aes(y = Ozone))

grid.arrange(p7, p8, p9, ncol = 2, nrow = 2)
```




Here, we have plotted the estimated function along with the approximate pointwise 95% confidence interval. There are two main features of these figures. The first is the fact that the intervals are quite wide. As comapred to those given in ESL, this should not be surprising as they do not include an estimate of $\sigma^2$, the conditional variance of $\mathbf{Y}$. For completeness, we include this estimate which causes our intervals to be considerablly larger. Moreover, as it appears we are overfitting this data, the bias in our estimator should be

quite low. As a result, however, the variance of this estimator is also inflated. The width of the confidence intervals given here reflect this Mean Squared Error tradeoff.

In addition, notice that the width of 3 intervals vary a function of $x$. This is attributitable to the fact that regression splines are locally adaptive. As we fix the knots aprior, some regions of the feature space will be more densly covered and as a result we expect a decrese in with of the confidence interval widths. This is also reflected in the visualizations above.

## Exercise 2: Ozone Smoothing Splines

To address the overfitting of the regression splines we turn to smoothing splines to produce a more regular solution to the problem. Indeed, we fit three different univariate smoothing splines; Ozone as a function of Radiation, Temperature, and Wind individually. The general problem we look to solve is given by

$$\hat{f} = \underset{f \in \mathcal{F}}{\arg\min} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \int f''(x) dx$$

where $\mathcal{F}$ is a predetermined Hilbert space. Recall from the previous homework, we showed that the solution of this problem is a natural cubic spline with knots at the datapoints and our optimization problem reduces to the following

$$\begin{aligned} \hat{\theta} &= \underset{\theta \in \mathbb{R}^p}{\arg\min} (\mathbf{Y} - \mathbf{H}\theta)^T (\mathbf{Y} - \mathbf{H}\theta) + \lambda \theta^T \Omega_N \theta \\ &= (\mathbf{H}^T \mathbf{H} + \lambda \Omega_n)^{-1} \mathbf{H}^T \mathbf{Y} \end{aligned}$$

where $[\Omega_n]_{ij} = \int h_j''(x) h_k''(x) dx$, $p$ is the number of unique $X$ values, and

$$\mathbf{H} = \begin{bmatrix} 1 & x_1 & N_2(x_1) & \dots & N_{p-2}(x_1) \\ 1 & x_2 & N_2(x_2) & \dots & N_{p-2}(x_2) \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & x_n & N_2(x_n) & \dots & N_{p-2}(n_2) \end{bmatrix} \in \mathbb{R}^{n \times p}$$

Therefore, we see that $\hat{\theta}$ is the generalized Ridge Regression estimator with regularization matrix $\Omega_n$. Therefore, it suffices to derive the form of $\Omega_n$. Without loss of generality, assume that $k \leq j$. First notice as $N_1''(x) = N_2''(x) = 0$, we only need to consider $k \geq 3$. With this, first notice that

$$N_k''(x) = 6 \left( \frac{(x - \xi_k)_+ - (x - \xi_K)_+}{\xi_K - \xi_k} - \frac{(x - \xi_{K-1})_+ - (x - \xi_K)_+}{\xi_K - \xi_{K-1}} \right)$$

With this expression, we can now find $[\Omega_n]_{jk}$ by breaking the integral over the $(-\infty, \xi_j) \cup [\xi_k, \xi_j) \cup [\xi_j, \xi_{K-1}) \cup [\xi_{K-1}, \xi_K) \cup [\xi_K, \infty)$. Well notice that $x \in (-\infty, \xi_j) \cup [\xi_k, \xi_j)$, $N_j''(x) = 0$. Moreover for $x \in [\xi_K, \infty)$, $N_j''(x) = 0$. Therefore we need only consider the regions $[\xi_j, \xi_{K-1}) \cup [\xi_{K-1}, \xi_K)$.

$$
\begin{aligned}
[\Omega_n]_{jk} &= \int_{[\xi_j,\xi_{K-1})} N_j''(x)N_k''(x)dx + \int_{[\xi_{K-1},\xi_K)} N_j''(x)N_k''(x)dx \\
&= 36\int_{[\xi_j,\xi_{K-1})} \left(\frac{x-\xi_k}{\xi_K-\xi_k}\right)\left(\frac{x-\xi_j}{\xi_K-\xi_j}\right)dx \\
&+ 36\int_{[\xi_{K-1},\xi_K)} \left(\frac{x-\xi_k}{\xi_K-\xi_k}\right)\left(\frac{x-\xi_j}{\xi_K-\xi_j}\right)dx \\
&- 36\int_{[\xi_{K-1},\xi_K)} \left(\frac{x-\xi_k}{\xi_K-\xi_k}\right)\left(\frac{x-\xi_{K-1}}{\xi_K-\xi_{K-1}}\right)dx \\
&- 36\int_{[\xi_{K-1},\xi_K)} \left(\frac{x-\xi_{K-1}}{\xi_K-\xi_{K-1}}\right)\left(\frac{x-\xi_j}{\xi_K-\xi_j}\right)dx \\
&+ 36\int_{[\xi_{K-1},\xi_K)} \left(\frac{x-\xi_{K-1}}{\xi_K-\xi_{K-1}}\right)^2 dx
\end{aligned}
$$

As each of these integrals have the same form, we simply integrate a general integal of this form. In implementation, we will construct $\Omega_n$ by evaluating each of these integrals and summing them in an appropriate fashion. The general integral form is given by

$$
\begin{aligned}
&\int_{[l,u]} \left(\frac{x-\text{knot1}}{\xi_K-\text{knot1}}\right)\left(\frac{x-\text{knot2}}{\xi_K-\text{knot2}}\right)dx \\
&= \frac{1}{(\xi_K-\text{knot1})(\xi_K-\text{knot2})}\left[\frac{1}{3}(u^3-l^3) - \frac{(\text{knot1}+\text{knot2})}{2})(u^2-l^2) + \text{knot1*knot2}(u-l)\right]
\end{aligned}
$$

Having constructed $\Omega_n$, we now look to derived some basic properties of the smoothing estimate $\hat{\theta}$. As in exercise one if we assume that $\mathbf{Y} \sim N(\mathbf{H}\theta, \sigma^2\mathbf{I})$ then

$$
\begin{aligned}
\mathbb{E}[\hat{\theta}] &= (\mathbf{H}^T\mathbf{H} + \lambda\Omega_n)^{-1}\mathbf{H}^T\mathbf{H}\theta \\
\text{Var}[\hat{\theta}] &= \sigma^2(\mathbf{H}^T\mathbf{H} + \lambda\Omega_n)^{-1}\mathbf{H}^T\mathbf{H}(\mathbf{H}^T\mathbf{H} + \lambda\Omega_n)^{-1}
\end{aligned}
$$

Notice that $\hat{\theta}$ is a biased estimator for $\theta$. As a result, by defining $\hat{\mathbf{Y}} = \mathbf{H}\hat{\theta} = \mathbf{H}(\mathbf{H}^T\mathbf{H} + \lambda\Omega_n)^{-1}\mathbf{H}^T\mathbf{Y}$ we have

$$
\begin{aligned}
\mathbb{E}[\hat{\mathbf{Y}}] &= \mathbf{H}(\mathbf{H}^T\mathbf{H} + \lambda\Omega_n)^{-1}\mathbf{H}^T\mathbf{H}\theta \\
\text{Var}[\hat{\mathbf{Y}}] &= \sigma^2\mathbf{H}(\mathbf{H}^T\mathbf{H} + \lambda\Omega_n)^{-1}\mathbf{H}^T\mathbf{H}(\mathbf{H}^T\mathbf{H} + \lambda\Omega_n)^{-1}\mathbf{H}^T
\end{aligned}
$$

Therefore, we see that $\hat{\mathbf{Y}}$ is a biased estimator of $\mathbf{H}\theta$. Therefore, while our function should be a biased estimate, we hope to see a favorable bias-variance trade off for this estimator. From here, we build an approximate $1-\alpha$ confidence interval of the form

$$
(\mathbf{h}(X_j)^*)^T\hat{\theta}_j^* \pm z_{\alpha/2}\hat{\sigma}\sqrt{(\mathbf{h}(X_j)^*)^T(\mathbf{H}^T\mathbf{H} + \lambda\Omega_n)^{-1}\mathbf{H}^T\mathbf{H}(\mathbf{H}^T\mathbf{H} + \lambda\Omega_n)^{-1}\mathbf{h}(X_j)^*}
$$

Notice here that we did not specify which feature, Radiation, Temperature, or Wind was given in $\mathbf{H}$. Therefore, we can apply every estimate, variance calculation, and confidence interval developed here to each design matrix $\mathbf{H}$ by simply updating the $\mathbf{H}$ matrix given above for the appropriate feature. Now that we have these properties, we are ready to implement the smoothing spline. We begin by defininining functions to calculate $\Omega_n$ and $\hat{\theta}$.

```
#---------------------------------
# Set up knots
#---------------------------------

# build dataframe for each feature
knot_r <- sort(unique(dat$radiation))
H_radiation <- cbind(rep(1, nrow(dat)), t(sapply(dat$radiation,
    FUN = function(x) h(x, knot_r))))

knot_t <- sort(unique(dat$temperature))
H_temperature <- cbind(rep(1, nrow(dat)), t(sapply(dat$temperature,
    FUN = function(x) h(x, knot_t))))

knot_w <- sort(unique(dat$wind))
H_wind <- cbind(rep(1, nrow(dat)), t(sapply(dat$wind, FUN = function(x) h(x,
    knot_w))))

#---------------------------------
# Get estimate
#---------------------------------

# build Omegan function
f <- function(k1, k2, u, l, knotK) {
    1/((knotK - k1) * (knotK - k2)) * (u^3 - l^3)/3 - (k1 + k2)/2 *
        (u^2 - l^2) + k1 * k2 * (u - l)
}

Omegan <- function(knots) {
    K <- length(knots)
    mat <- matrix(NA, nrow = K, ncol = K)
    for (k in 1:(K - 2)) {
        for (j in k:(K - 2)) {
            mat[k + 2, j + 2] <- 36 * (f(knots[k], knots[j],
                knots[K - 1], knots[j], knots[K]) + f(knots[k],
                knots[j], knots[K], knots[K - 1], knots[K]) -
                f(knots[k], knots[K - 1], knots[K], knots[K -
                  1], knots[K]) - f(knots[K - 1], knots[j], knots[K],
                knots[K - 1], knots[K]) + f(knots[K - 1], knots[K -
                1], knots[K], knots[K - 1], knots[K]))
        }
    }

    mat[lower.tri(mat)] <- t(mat)[lower.tri(mat)]
    mat[1, ] <- mat[2, ] <- mat[, 1] <- mat[, 2] <- 0
    return(mat)

}
```

```r
# ridge regression function
Slam <- function(H, y, lambda, knots) {
    solve(crossprod(H) + lambda * Omegan(knots)) %*% crossprod(H,
        y)
}
```

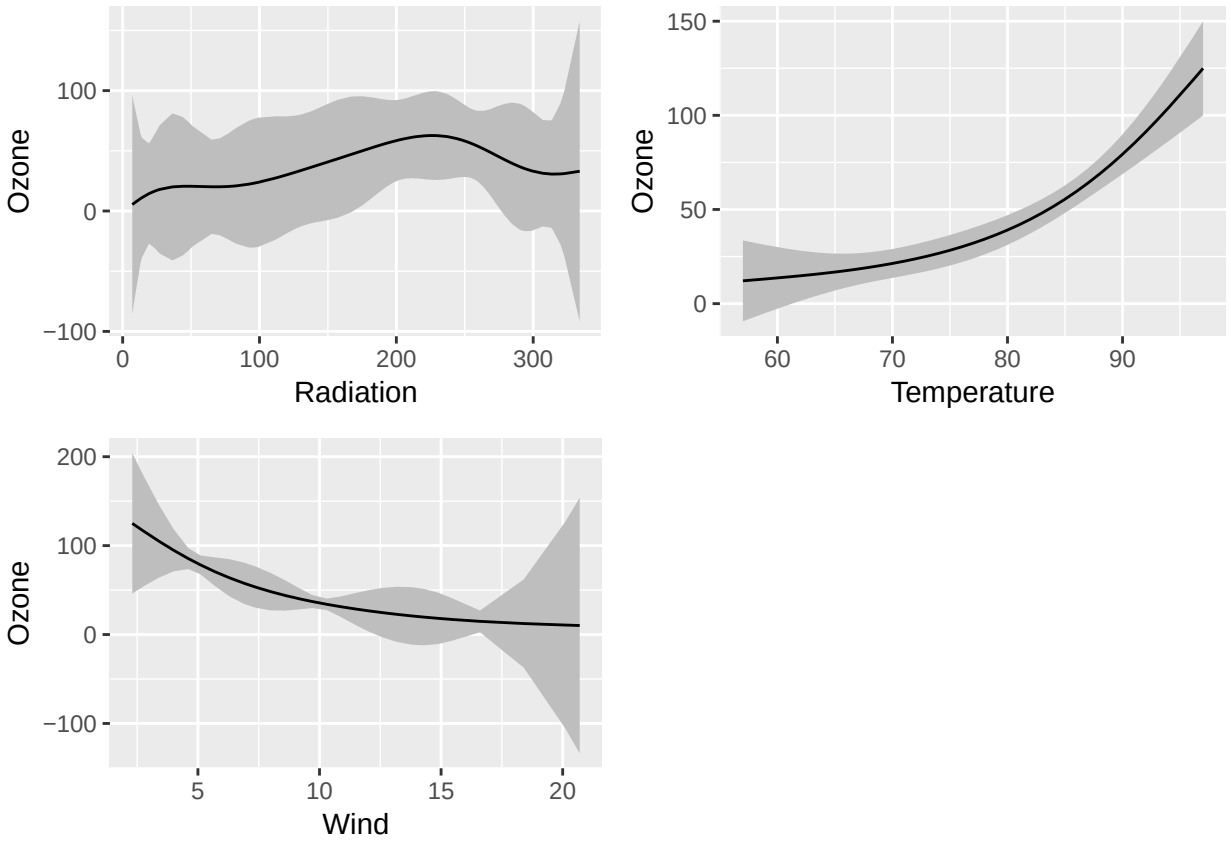
Having implemented our smoothing sline, we now turn to Leave-one-out cross validation for choice of smoothing parameter $\lambda$. We consider 100 candidate $\lambda \in \Lambda$ values in $(.001, 5)$.

```r
#-------------------------------------
# Leave-one-Out Cross Validation
#-------------------------------------

# Define lambda values + error vectors
L <- 100
n <- nrow(dat)
Lambda <- seq(0.001, 5, length.out = L)
Error_rad <- numeric(L)
Error_temp <- numeric(L)
Error_wind <- numeric(L)

# define Omegas
OmR <- Omegan(knot_r)
OmT <- Omegan(knot_t)
OmW <- Omegan(knot_w)

for (l in 1:L) {
    for (i in 1:nrow(dat)) {
        Error_rad[l] <- Error_rad[l] + 1/n * (crossprod(H_radiation[i,
            ], solve(crossprod(H_radiation[-i, ]) + Lambda[l] *
            OmR) %*% crossprod(H_radiation[-i, ], dat$ozone[-i])) -
            dat$ozone[i])^2

        Error_temp[l] <- Error_temp[l] + 1/n * (crossprod(H_temperature[i,
            ], solve(crossprod(H_temperature[-i, ]) + Lambda[l] *
            OmT) %*% crossprod(H_temperature[-i, ], dat$ozone[-i])) -
            dat$ozone[i])^2

        Error_wind[l] <- Error_wind[l] + 1/n * (crossprod(H_wind[i,
            ], solve(crossprod(H_wind[-i, ]) + Lambda[l] * OmW) %*%
            crossprod(H_wind[-i, ], dat$ozone[-i])) - dat$ozone[i])^2
    }
}


#-------------------------------------
# Get optimal smoothing parameter
#-------------------------------------
lam_opt_r <- Lambda[which.min(Error_rad)]
lam_opt_t <- Lambda[which.min(Error_temp)]
lam_opt_w <- Lambda[which.min(Error_wind)]

#-------------------------------------
# Get estimates
```

```
#-------------------------------------
theta_hat_r <- solve(crossprod(H_radiation) + lam_opt_r * Omegan(knot_r)) %*%
    crossprod(H_radiation, dat$ozone)

theta_hat_t <- solve(crossprod(H_temperature) + lam_opt_t * Omegan(knot_t)) %*%
    crossprod(H_temperature, dat$ozone)

theta_hat_w <- solve(crossprod(H_wind) + lam_opt_w * Omegan(knot_w)) %*%
    crossprod(H_wind, dat$ozone)
```

Having now obtained our smoothing spline estimates, we proceed as we did in the first exercise to visualize the smoothing spline fits.

```
library(dplyr)
#-------------------------------------
# Get Variance Estimates
#-------------------------------------
rad.inter <- sqrt(as.numeric(crossprod(dat$ozone - H_radiation %*%
    theta_hat_r))/(nrow(dat) - ncol(H_radiation))) * sqrt(diag(crossprod(H_radiation %*%
    solve(crossprod(H_radiation) + lam_opt_r * OmR) %*% t(H_radiation))))

temp.inter <- sqrt(as.numeric(crossprod(dat$ozone - H_temperature %*%
    theta_hat_t))/(nrow(dat) - ncol(H_temperature))) * sqrt(diag(crossprod(H_temperature %*%
    solve(crossprod(H_temperature) + lam_opt_t * OmT) %*% t(H_temperature))))

wind.inter <- sqrt(as.numeric(crossprod(dat$ozone - H_wind %*%
    theta_hat_w))/(nrow(dat) - ncol(H_wind))) * sqrt(diag(crossprod(H_wind %*%
    solve(crossprod(H_wind) + lam_opt_w * OmW) %*% t(H_wind))))

#-------------------------------------
# Visualize fits
#-------------------------------------
df_rad <- data.frame(Radiation = dat$radiation, Ozone = H_radiation %*%
    theta_hat_r, lower = H_radiation %*% theta_hat_r - 2 * rad.inter,
    upper = H_radiation %*% theta_hat_r + 2 * rad.inter)
df_temp <- data.frame(Temperature = dat$temperature, Ozone = H_temperature %*%
    theta_hat_t, lower = H_temperature %*% theta_hat_t - 2 *
    temp.inter, upper = H_temperature %*% theta_hat_t + 2 * temp.inter)
df_wind <- data.frame(Wind = dat$wind, Ozone = H_wind %*% theta_hat_w,
    lower = H_wind %*% theta_hat_w - 2 * wind.inter, upper = H_wind %*%
        theta_hat_w + 2 * wind.inter)

p10 <- ggplot(df_rad, aes(x = Radiation, y = Ozone)) + geom_ribbon(aes(ymin = lower,
    ymax = upper), fill = "grey") + geom_line(aes(y = Ozone))

p11 <- ggplot(df_temp, aes(x = Temperature, y = Ozone)) + geom_ribbon(aes(ymin = lower,
    ymax = upper), fill = "grey") + geom_line(aes(y = Ozone))

p12 <- ggplot(df_wind, aes(x = Wind, y = Ozone)) + geom_ribbon(aes(ymin = lower,
    ymax = upper), fill = "grey") + geom_line(aes(y = Ozone))

grid.arrange(p10, p11, p12, ncol = 2, nrow = 2)
```




In these figures, we see that through regularization, we have achieved slight smoothing of the functions from Exercise 1. In addition, we see that the confidence bands have reduced significantly. This can be attributed to the same MSE tradeoff as discussed in exercise 1. That is, as we were overfitting in the multidimensional regression spline, we had very high variance as the bias was minimal. In this exercise, we have introduced a small bias term and as a result see that the variance has reduced significantly.