ComS 573: Homework #4

Due on April 4, 2014

Professor De Brabanter at 10am

Josh Davis

Problem 1

From ISLR: Chapter 7, Problem 1.

A cubic regression spline with one knot ξ can be obtained using a basis of the form $1, x, x^2, x^3, (x-\xi)^3_+$ where $(x-\xi)^3$ if $x > \xi$ and equals 0 otherwise. Show that a function of the form

$$f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 (x-\xi)^3_+$$

is indeed a cubic regression spline, regardless of the values of $\beta_0, \beta_1, \beta_2, \beta_3, \beta_4$.

Solution

To solve this, we need to do four things.

(a) The first is that we need to find a cubic polynomial $f_1(x) = a_1 + b_1 x + c_1 x^2 + d_1 x^3$ such that $f(x) = f_1(x)$ for all $x \leq \xi$. While expressing a_1, b_1, c_1, d_1 in terms of $\beta_0, \beta_1, \beta_2, \beta_3, \beta_4$.

We also need to find a cubic polynomial $f_2(x) = a_2 + b_2 x + c_2 x^2 + d_2 x^3$ such that $f(x) = f_2(x)$ for all $x > \xi$. While expressing a_2, b_2, c_2, d_2 in terms of $\beta_0, \beta_1, \beta_2, \beta_3, \beta_4$.

These two functions tell us that $f(x)$ is a piecewise polynomial.

(b) The second thing that we need to do is to show that $f_1(\xi) = f_2(\xi)$. That is, $f(x)$ is continuous at ξ.

(c) The third thing that we need to do is to show that $f_1'(\xi) = f_2'(\xi)$. That is, $f'(x)$ is continuous at ξ.

(d) Lastly we need to show that $f_1''(\xi) = f_2''(\xi)$. That is, $f''(x)$ is continuous at ξ.

Part A

We need to find a new cubic polynomial $f_1(x)$ such that $f(x) = f_1(x)$ for all $x \leq \xi$. Given the positive constraint on the 4th basis of $(x-\xi)^3_+$, we know that if $x \geq \xi$, then the basis is equal to 0.

Thus for $f_1(x) = f(x)$, if we let $a_1 = \beta_0, b_1 = \beta_1, c_1 = \beta_2, d_1 = \beta_3$, then it is easy to see that $f_1(x) = f(x)$ because:

$$\begin{aligned}
f(x) &= \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 (x-\xi)^3_+ \\
&= \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + 0 \text{ because } x \leq \xi \\
&= a_1 + b_1 x + c_1 x^2 + d_1 x^3
\end{aligned}$$

Therefore $f_1(x) = f(x)$ for all $x \leq \xi$.

Now we must do the same but instead find a cubic polynomial $f_2(x) = a_2 + b_2 x + c_2 x^2 + d_2 x^3$ such that $f(x) = f_2(x)$ for all $x \leq \xi$.

This gives us:

$$\begin{aligned}
f(x) &= \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 (x-\xi)^3_+ \\
&= \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 (x-\xi)^3 \quad \text{because } x > \xi \\
&= \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 (x^3 - 3x^2\xi + 3x\xi^2 - \xi^3) \\
&= \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 x^3 - 3\beta_4 x^2 \xi + 3\beta_4 x \xi^2 - \beta_4 \xi^3 \\
&= \beta_0 - \beta_4 \xi^3 + \beta_1 x + 3\beta_4 x \xi^2 + \beta_2 x^2 - 3\beta_4 x^2 \xi + \beta_3 x^3 + \beta_4 x^3 \\
&= (\beta_0 - \beta_4 \xi^3) + (\beta_1 + 3\beta_4 \xi^2) x + (\beta_2 - 3\beta_4 \xi) x^2 + (\beta_3 + \beta_4) x^3
\end{aligned}$$

where $a_2 = \beta_0 - \beta_4\xi^3$, $b_2 = \beta_1 + 3\beta_4\xi^2$, $c_2 = \beta_2 - 3\beta_4\xi$, and $d_2 = \beta_3 + \beta_4$.

Thus the first part is finished and our functions, f_1 and f_2 are:

$$\begin{aligned}
f_1(x) &= \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 \\
f_2(x) &= (\beta_0 - \beta_4\xi^3) + (\beta_1 + 3\beta_4\xi^2)x + (\beta_2 - 3\beta_4\xi)x^2 + (\beta_3 + \beta_4)x^3
\end{aligned}$$

Part B

Now we need to show that our cubic functions will still be continuous at the knots, thus we need to show that $f_1(\xi) = f_2(\xi)$.

$$\begin{aligned}
f_2(\xi) &= (\beta_0 - \beta_4\xi^3) + (\beta_1 + 3\beta_4\xi^2)\xi + (\beta_2 - 3\beta_4\xi)\xi^2 + (\beta_3 + \beta_4)\xi^3 \\
&= \beta_0 - \beta_4\xi^3 + \beta_1\xi + 3\beta_4\xi^3 + \beta_2\xi^2 - 3\beta_4\xi^3 + \beta_3\xi^3 + \beta_4\xi^3 \\
&= \beta_0 + (\beta_4\xi^3 - \beta_4\xi^3) + \beta_1\xi + (3\beta_4\xi^3 - 3\beta_4\xi^3) + \beta_2\xi^2 + \beta_3\xi^3 \\
&= \beta_0 + \beta_1\xi + \beta_2\xi^2 + \beta_3\xi^3 \\
&= f_1(\xi)
\end{aligned}$$

Thus the two functions are continous at the knots, ξ.

Part C

Now we need to show that the first derivatives of the cubic functions will still be continuous at the knots, thus we need to show that $f_1'(\xi) = f_2'(\xi)$.

The derivative of $f_1(x)$ is:

$$\frac{\mathrm{d}}{\mathrm{d}x}(f_1(x)) = \beta_1 + 2\beta_2 x + 3\beta_3 x^2$$

The derivative of $f_2(x)$ is:

$$\frac{\mathrm{d}}{\mathrm{d}x}(f_2(x)) = (\beta_1 + 3\beta_4\xi^2) + 2(\beta_2 - 3\beta_4\xi)x + 3(\beta_3 + \beta_4)x^2$$

This gives $f_2'(\xi)$:

$$\begin{aligned}
f_2'(\xi) &= (\beta_1 + 3\beta_4\xi^2) + 2(\beta_2 - 3\beta_4\xi)\xi + 3(\beta_3 + \beta_4)\xi^2 \\
&= \beta_1 + 3\beta_4\xi^2 + 2\beta_2\xi - 6\beta_4\xi^2 + 3\beta_3\xi^2 + 3\beta_4\xi^2 \\
&= \beta_1 + 2\beta_2\xi + (3\beta_4\xi^2 + 3\beta_4\xi^2 - 6\beta_4\xi^2) + 3\beta_3\xi^2 \\
&= \beta_1 + 2\beta_2\xi + 3\beta_3\xi^2 \\
&= f_1'(\xi)
\end{aligned}$$

Thus first derivatives are still continous at the knots, ξ.

Part D

Now we need to show that the second derivatives of the cubic functions will still be continuous at the knots, thus we need to show that $f_1''(\xi) = f_2''(\xi)$.

The second derivative of $f_1(x)$ is:

$$\frac{\mathrm{d}}{\mathrm{d}x}(f_1'(x)) = 2\beta_2 + 6\beta_3 x$$

Problem 1 continued on next page...

The second derivative of $f_2(x)$ is:

$$\frac{\mathrm{d}}{\mathrm{d}x}(f_2'(x)) = 2(\beta_2 - 3\beta_4\xi) + 6(\beta_3 + \beta_4)x$$

This gives $f_2''(\xi)$:

$$\begin{aligned} f_2''(\xi) &= 2(\beta_2 - 3\beta_4\xi) + 6(\beta_3 + \beta_4)\xi \\ &= 2\beta_2 - 6\beta_4\xi + 6\beta_3\xi + 6\beta_4\xi \\ &= 2\beta_2 + (6\beta_4\xi - 6\beta_4\xi) + 6\beta_3\xi \\ &= 2\beta_2 + 6\beta_3\xi \\ &= f_1''(\xi) \end{aligned}$$

Thus the second derivatives are still continous at the knots, ξ.

Conclusion

Thus since we meet all the criteria from the beginning of this solution, we know that $f(x)$ is indeed a cubic spline.

Problem 2

Similar to Problem 11 from ISLR: Chapter 7.

Do an iterative approach to GAMs by repeatedly holding all but one coefficient estimate fixed at its current value, and update only the coefficient estimate using a simple linear regression. Continue the process until convergence – that is, until the coefficient estimates stop changing. The process flow is sketched below:

1. Download the *adv.dat* data set ($n = 200$) with response Y and two predictors, X_1, X_2 on BlackBoard.
2. Initialize $\hat{\beta}_1$ (estimated coefficient of X_1) to take on a value of your choice, say 0.
3. Keeping $\hat{\beta}_1$ fixed, fit the model:
$$Y - \hat{\beta}_1 X_1 = \beta_0 + \beta_2 X_2 + e$$
4. Keeping $\hat{\beta}_2$ fixed, fit the model:
$$Y - \hat{\beta}_2 X_2 = \beta_0 + \beta_1 X_1 + e$$

```
# Read in the data
data <- read.csv("./adv.dat")
```

Part A

Write a for loop to repeat (3) and (4) 1,000 times. Report the estimates of $\hat{\beta}_0, \hat{\beta}_1$ and $\hat{\beta}_2$ at each iteration of the for loop. Create a plot in which each of these values is displayed with $\hat{\beta}_0, \hat{\beta}_1$ and $\hat{\beta}_2$ each shown in a different color.

Solution

First let's write some helper functions:

```
estimate.beta <- function(fixed, Y, X1, X2) {
    a <- Y - fixed * X1
    fit <- lm(a ~ X2)

    # Return coefficient we want
    fit$coef[2]
}
```

Now let's try the backfitting

```
N <- 1000
R <- 1:N
beta0 <- rep(NA, N)
beta1 <- rep(NA, N)
beta2 <- rep(NA, N)

# Start off with our estimate of 0
beta1[1] <- 0

for (i in R) {
    beta2[i] <- estimate.beta(beta1[i], data$Y, data$X1, data$X2)
```

Problem 2 continued on next page...

```
    # Keep the lists the same size...
    if (i != 1000) {
        beta1[i + 1] <- estimate.beta(beta2[i], data$Y, data$X2, data$X1)
    }

    # Assign beta0 manually
    fit <- lm(data$Y ~ I(beta1[i] * data$X1) + I(beta2[i] * data$X2))
    beta0[i] <- fit$coef[1]
}
```

Plotting these values gives us the following:

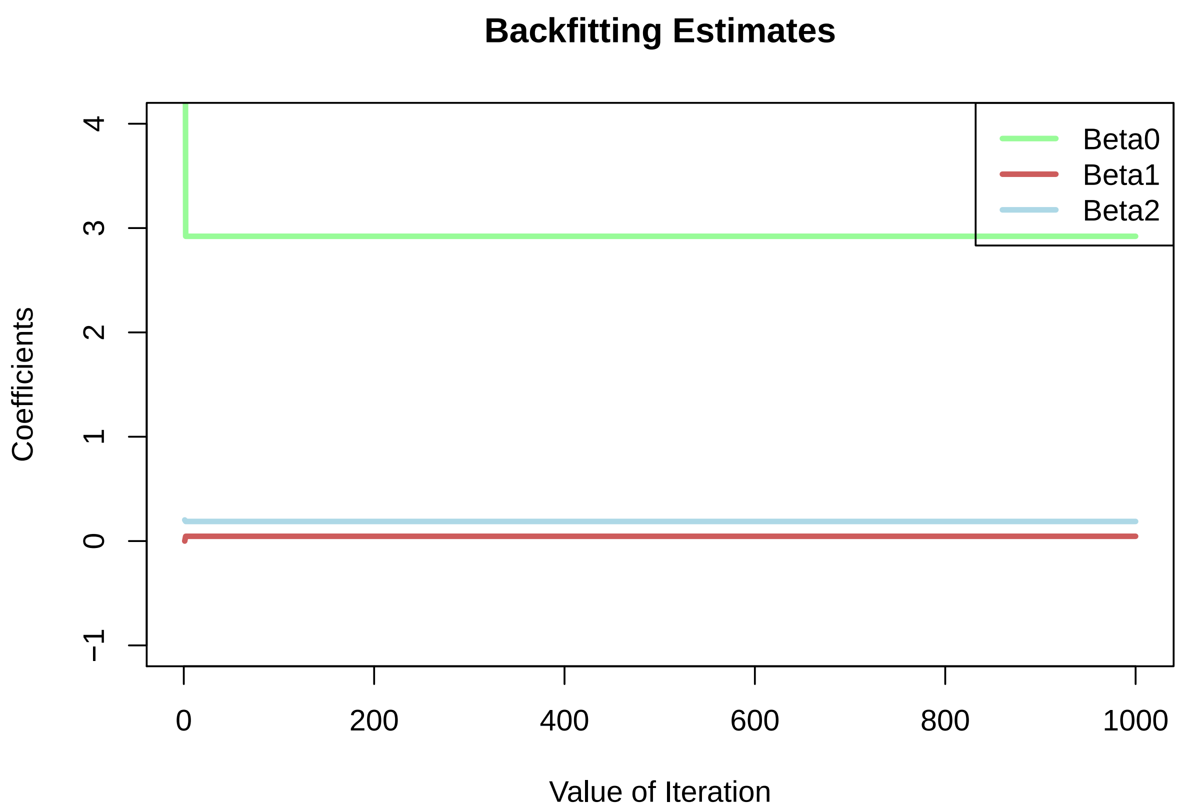

Figure 1: Comparison of estimates over 1000 iterations

Part B

Compare your answers in (a) to the results of simply performing multiple linear regression to predict Y using X_1 and X_2. Use the `abline()` function to overlay those multiple linear regression coefficient esitmates on the plot obtained in (a).

Solution

Plotting the previous plot with the added lines as black ones.

```
fit <- lm(data$Y ~ data$X1 + data$X2)

plotPartA()

# Plot beta0
abline(h = fit$coef[1], lty = 2, lwd = 3, col = "black")

# Plot beta1
abline(h = fit$coef[2], lty = 2, lwd = 3, col = "black")

# Plot beta1
abline(h = fit$coef[3], lty = 2, lwd = 3, col = "black")
```

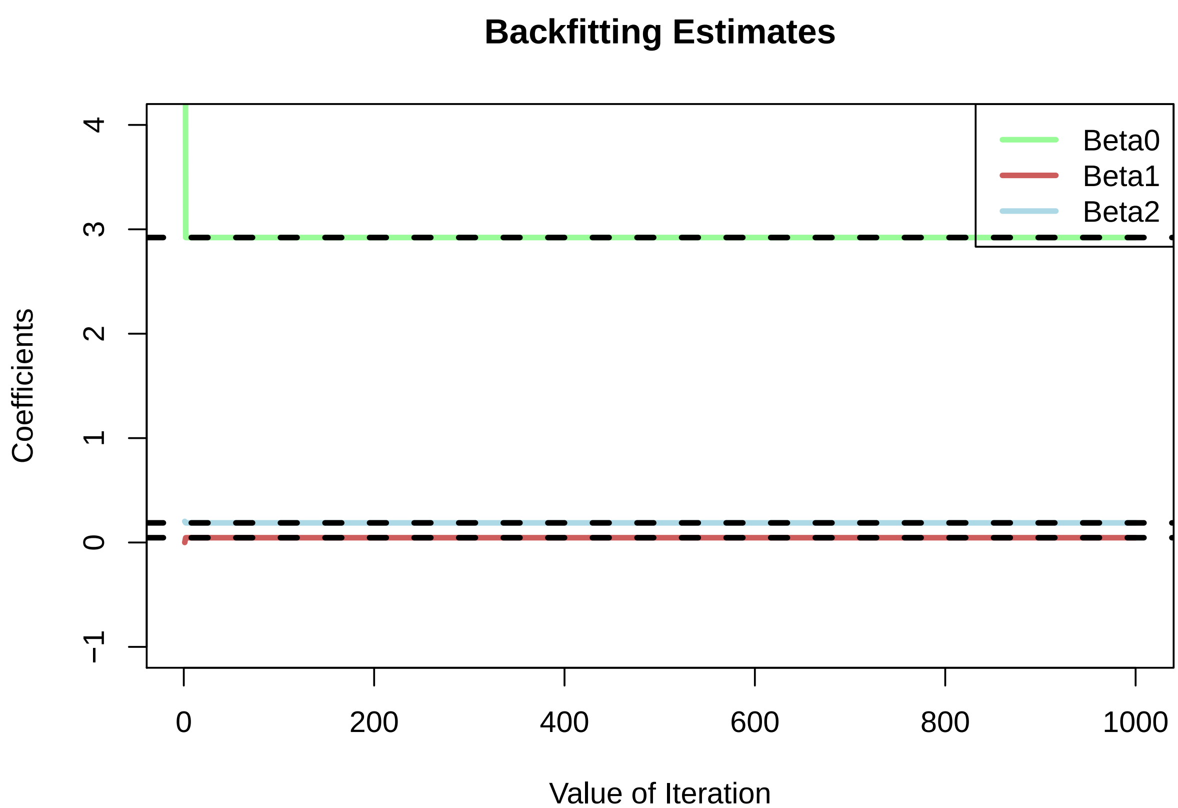

Figure 2: Comparison of estimates over 1000 iterations

Part C

On this data set, how many backfitting iterations were required in order to obtain a "good" approximation to the multiple regression coefficient estimates? What would be a good stopping criterion?

Solution

It happens quite early. Below we'll determine how many iterations it took.

One way to create a stopping criterion is to look at the difference between the current and previous value. Once we reach a point where it doesn't change for some tolerance h, then we could stop. The tolerance could be changed depending on the situation but in our case, let's see what ours looked like (starting at 2):

```r
beta0.diff <- rep(NA, N)
beta1.diff <- rep(NA, N)
beta2.diff <- rep(NA, N)

# Start at 2, we can't have a difference at index 1
for (i in 2:N) {
    beta0.diff[i] <- abs(beta0[i] - beta0[i - 1])
    beta1.diff[i] <- abs(beta1[i] - beta1[i - 1])
    beta2.diff[i] <- abs(beta2[i] - beta2[i - 1])
}
```

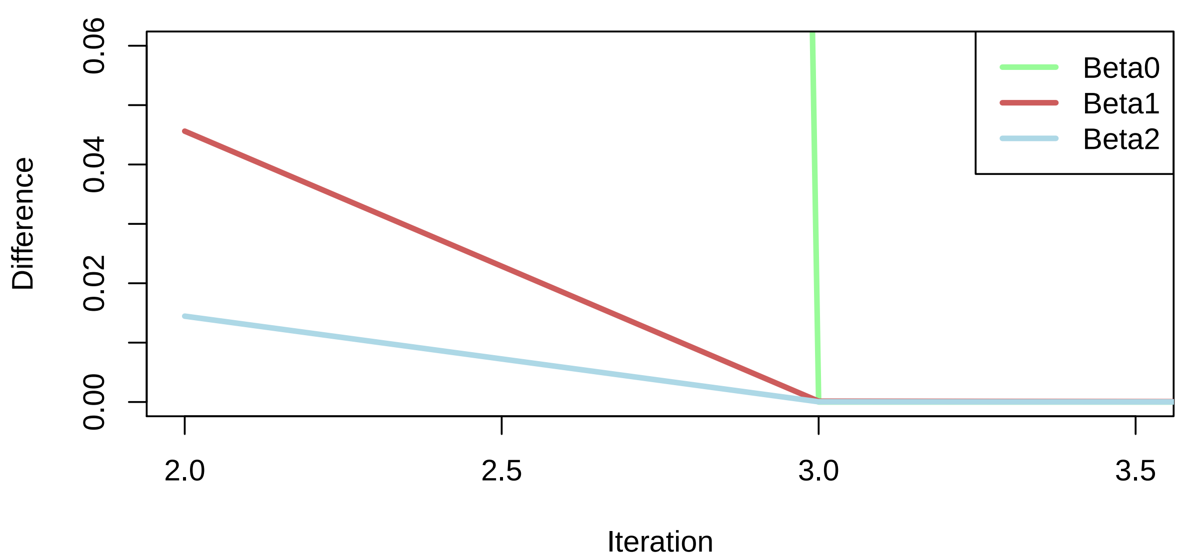

Figure 3: Differences between current and previous estimate

Thus we can see ours converged around just 3 iterations. And in our case, it they converged very quickly and thus a tolerance of 0.005 would have ended it properly. It might be hard to determine what will make a good tolerance though.

Problem 3

Show that the Nadaraya-Watson estimator is equal to **local constant** fitting. *Hint:* Use the local polynomial cost function to start and adapt where necessary.

Solution

We want to show that the Nadaraya-Watson estimator is equal to the local constant fitting. To do this, let's look at the normal local polynomial cost function:

$$\min_{\beta_j, j=0,\ldots p} \sum_{i=1}^{N} K_h(x - X_i) \left[Y_i - \sum_{j=0}^{p} \beta_j X_i^j \right]^2$$

Let's take a look at what happens when the number of our dimensions is 0, or $p = 0$. We get:

$$\begin{aligned}
&\min_{\beta_j, j=0,\ldots 0} \sum_{i=1}^{N} K_h(x - X_i) \left[Y_i - \sum_{j=0}^{0} \beta_j X_i^j \right]^2 \\
&= \min_{\beta_0} \sum_{i=1}^{N} K_h(x - X_i) \left[Y_i - \beta_0 \right]^2 \\
&= \sum_{i=1}^{N} K_h(x - X_i)
\end{aligned}$$

Remembering that for two random variables, say X and Y and the joint pdf of $f(x, y)$, if we want the conditional expectation, $\mathrm{E}[Y \mid X = x]$, then we have:

$$\mathrm{E}[Y \mid X = x] = \frac{\int y f(x, y)}{\int f(x, y)} = m(x)$$

Using this fact and plugging in our minimized local polynomial cost function, we get:

$$\begin{aligned}
m(x) &= \frac{\int y f(x, y)}{\int f(x, y)} \\
&= \frac{\sum_{i=1}^{N} K_h(x - X_i) Y_i}{\sum_{i=1}^{N} K_h(x - X_i)} \\
&= \frac{\sum_{i=1}^{N} K_h(x - X_i) Y_i}{\sum_{i=1}^{N} K_h(x - X_i)}
\end{aligned}$$

which is equal to the Nadaraya-Watson estimator. Thus we can see that when using the local polynomial cost function and reducing the dimensions to 0, we end up with the NW estimator. Basically it is a special case of the local polynomial function.

Problem 4

Show that the kernel density estimate

$$f(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right)$$

with kernel K and bandwidth $h > 0$, is a bonafide density. Did you really need any condition(s) on K? If so, which one(s)?

Solution

According to what a bona fide kernel density estimator is, it can be anything as long as our density, $f(x)$ satisfies the following:

(a) $\int_{-\infty}^{+\infty} f(x)\mathrm{d}x = 1$

(b) And non-negative for all values (because it is a probability).

These are the basic conditions for a probability density function. Let's address each one of these conditions and see what happens.

Part A

First let's take the integral of $f(x)$:

$$\begin{aligned}
\int_{-\infty}^{+\infty} f(x)\mathrm{d}x &= \int_{-\infty}^{+\infty} \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right)\mathrm{d}x \\
&= \frac{1}{nh}\int_{-\infty}^{+\infty} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right)\mathrm{d}x \\
&= \frac{1}{nh}\int_{-\infty}^{+\infty} \left[K\left(\frac{x - X_1}{h}\right) + K\left(\frac{x - X_2}{h}\right) + \cdots + K\left(\frac{x - X_n}{h}\right)\right]\mathrm{d}x \\
&= \frac{1}{nh}\int_{-\infty}^{+\infty} K\left(\frac{x - X_1}{h}\right)\mathrm{d}x + \int_{-\infty}^{+\infty} K\left(\frac{x - X_2}{h}\right) + \cdots + \int_{-\infty}^{+\infty} K\left(\frac{x - X_n}{h}\right)\mathrm{d}x
\end{aligned}$$

If we use our scaled kernel instead of having h as a parameter, we get:

$$\begin{aligned}
\int_{-\infty}^{+\infty} f_h(x)\mathrm{d}x &= \frac{1}{n}\left[\int_{-\infty}^{+\infty} K_h\left(x - X_1\right)\mathrm{d}x + \int_{-\infty}^{+\infty} K_h\left(x - X_2\right) + \cdots + \int_{-\infty}^{+\infty} K_h\left(x - X_n\right)\mathrm{d}x\right] \\
&= 1
\end{aligned}$$

Thus we can easily see that the first condition that must hold is that the addition of all the integrals of our kernel function must equal n, or that each integral of the kernel function, K_h, must equal 1:

$$\int_{-\infty}^{+\infty} K\left(x\right)\mathrm{d}x = 1$$

Part B

By allowing a kernel function to be negative, that would be saying that we give values around it a negative influence. Thus this condition should pass onto our K function as well. Therefore:

$$K(x) \text{ is nonnegative as well}$$

Part C

Lastly, there is another condition that is in Kris' notes. It says that the kernel, K, should also be symmetric and centered on the point. This is to ensure that the average of the distribution is the same as the sample.