Assignment 3 submission

Zhuo Diao

October 3rd, 2025

1 Theoretical exercise: Local linear regression as a linear smoother

1.1 Prove that $\hat{m}(x) = \sum_{i=1}^{n} w_{ni}(x) Y_i$

Let the design matrix X be

$$X = \begin{bmatrix} 1 & X_1 - x \\ \vdots & \vdots \\ 1 & X_n - x \end{bmatrix}$$

Let W be the diagonal matrix of kernel weights:

$$W = \operatorname{diag}\left(K\left(\frac{X_1 - x}{h}\right), \dots, K\left(\frac{X_n - x}{h}\right)\right)$$

Since the local linear regression estimator at a point x is defined by

$$(\hat{\beta}_0(x), \hat{\beta}_1(x)) = \arg\min_{\beta_0, \beta_1 \in \mathbb{R}} \sum_{i=1}^{n} \left(Y_i - \beta_0 - \beta_1 (X_i - x)\right)^2 K\left(\frac{X_i - x}{h}\right),$$

then the weighted least squares estimator has the closed-form:

$$\hat{\beta}(x) = (X^\top W X)^{-1} X^\top W Y$$

where $Y = (Y_1, \dots, Y_n)^\top$.

Thus,

$$\hat{m}(x) = \hat{\beta}_0(x) = e_1^\top \hat{\beta}(x) = e_1^\top (X^\top W X)^{-1} X^\top W Y,$$

where $e_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$.

Let

$$w(x) := W X (X^\top W X)^{-1} e_1 \in \mathbb{R}^n.$$

Then

$$\hat{m}(x) = \sum_{i=1}^{n} w_{ni}(x) Y_i, \quad \text{where } w_{ni}(x) = [w(x)]_i .$$

Therefore, $\hat{m}(x)$ can be expressed as a weighted average of the observations:

$$\hat{m}(x) = \sum_{i=1}^{n} w_{ni}(x) Y_i, \quad \text{where } w_{ni}(x) = [w(x)]_i ,$$

where the weights $w_{ni}(x)$ depend only on $x, X_i, K,$ and h, but not on Y_i.

1.2 Derive an explicit expression for $w_{ni}(x)$ in terms of $S_{n,0}(x)$, $S_{n,1}(x)$, $S_{n,2}(x)$, and the kernel

To find

$$(\hat{\beta}_0, \hat{\beta}_1) = \arg\min_{\beta_0,\beta_1} \sum_{i=1}^{n} \left(Y_i - \beta_0 - \beta_1(X_i - x)\right)^2 K\left(\frac{X_i - x}{h}\right),$$

we take partial derivatives and set them to 0:

$$-2\sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right)\left(Y_i - \beta_0 - \beta_1(X_i - x)\right) = 0$$

$$-2\sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right)(X_i - x)\left(Y_i - \beta_0 - \beta_1(X_i - x)\right) = 0$$

Let

$$\begin{aligned}
S_0 &= \sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right), \\
S_1 &= \sum_{i=1}^{n} (X_i - x) K\left(\frac{X_i - x}{h}\right), \\
S_2 &= \sum_{i=1}^{n} (X_i - x)^2 K\left(\frac{X_i - x}{h}\right), \\
T_0 &= \sum_{i=1}^{n} Y_i K\left(\frac{X_i - x}{h}\right), \\
T_1 &= \sum_{i=1}^{n} Y_i (X_i - x) K\left(\frac{X_i - x}{h}\right).
\end{aligned}$$

Then we have

$$\begin{cases} S_0 \cdot \hat{\beta}_0 + S_1 \cdot \hat{\beta}_1 = T_0 \\ S_1 \cdot \hat{\beta}_0 + S_2 \cdot \hat{\beta}_1 = T_1 \end{cases}$$

Thus,

$$\hat{\beta}_0 = \frac{T_0 \cdot S_2 - T_1 \cdot S_1}{S_0 \cdot S_2 - S_1^2}$$

Since $\hat{m}(x) = \hat{\beta}_0(x)$, we substitute the expressions for T_0 and T_1 to get

$$\begin{aligned}
\hat{m}(x) &= \frac{\sum_{i=1}^{n} Y_i K\left(\frac{X_i - x}{h}\right) S_2 - \sum_{i=1}^{n} Y_i (X_i - x) K\left(\frac{X_i - x}{h}\right) S_1}{S_0 S_2 - S_1^2} \\
&= \sum_{i=1}^{n} \left[\frac{K\left(\frac{X_i - x}{h}\right)\left(S_2 - (X_i - x) S_1\right)}{S_0 S_2 - S_1^2} \right] Y_i
\end{aligned}$$

Thus,

$$w_{ni}(x) = \frac{K\left(\frac{X_i - x}{h}\right)\left(S_2 - (X_i - x) S_1\right)}{S_0 S_2 - S_1^2}$$

Let

$$S_{n,k}(x) = \frac{1}{nh}\sum_{i=1}^{n}(X_i - x)^k K\left(\frac{X_i - x}{h}\right), \quad k = 0, 1, 2.$$

Then

$$\begin{aligned} S_0 &= nh \cdot S_{n,0}(x), \\ S_1 &= nh \cdot S_{n,1}(x), \\ S_2 &= nh \cdot S_{n,2}(x). \end{aligned}$$

Substituting these into the formula of $w_{ni}(x)$, we get

$$\begin{aligned} w_{ni}(x) &= \frac{K\left(\frac{X_i - x}{h}\right)\left(nh \cdot S_{n,2}(x) - (X_i - x) \cdot nh \cdot S_{n,1}(x)\right)}{(nh)^2 \cdot \left(S_{n,0}(x)S_{n,2}(x) - S_{n,1}^2(x)\right)} \\ &= \frac{1}{nh} \cdot \frac{K\left(\frac{X_i - x}{h}\right)\left(S_{n,2}(x) - (X_i - x)S_{n,1}(x)\right)}{S_{n,0}(x)S_{n,2}(x) - S_{n,1}^2(x)} \end{aligned}$$

Therefore the explicit expression for $w_{ni}(x)$ in terms of $S_{n,0}(x)$, $S_{n,1}(x)$, $S_{n,2}(x)$, and the kernel is

$$w_{ni}(x) = \frac{1}{nh} \cdot \frac{K\left(\frac{X_i - x}{h}\right)\left[S_{n,2}(x) - (X_i - x)S_{n,1}(x)\right]}{S_{n,0}(x)S_{n,2}(x) - S_{n,1}^2(x)}$$

where

$$S_{n,k}(x) = \frac{1}{nh}\sum_{i=1}^{n}(X_i - x)^k K\left(\frac{X_i - x}{h}\right), \quad k = 0, 1, 2.$$

1.3 Prove that $\sum_{i=1}^{n} w_{ni}(x) = 1$

$$\begin{aligned} \sum_{i=1}^{n} w_{ni}(x) &= \sum_{i=1}^{n} \frac{1}{nh} \frac{K\left(\frac{X_i - x}{h}\right)\left[S_{n,2}(x) - (X_i - x)\, S_{n,1}(x)\right]}{S_{n,0}(x)S_{n,2}(x) - \left(S_{n,1}(x)\right)^2} \\ &= \frac{1}{nh} \frac{S_{n,2}(x)\sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right) - S_{n,1}(x)\sum_{i=1}^{n}(X_i - x)\, K\left(\frac{X_i - x}{h}\right)}{S_{n,0}(x)S_{n,2}(x) - \left(S_{n,1}(x)\right)^2} \end{aligned}$$

Since

$$\sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right) = nh\, S_{n,0}(x), \qquad \sum_{i=1}^{n}(X_i - x)\, K\left(\frac{X_i - x}{h}\right) = nh\, S_{n,1}(x),$$

we then have

$$\begin{aligned} \sum_{i=1}^{n} w_{ni}(x) &= \frac{1}{nh} \frac{S_{n,2}(x)\,(nh\, S_{n,0}(x)) - S_{n,1}(x)\,(nh\, S_{n,1}(x))}{S_{n,0}(x)S_{n,2}(x) - \left(S_{n,1}(x)\right)^2} \\ &= \frac{S_{n,2}(x)S_{n,0}(x) - \left(S_{n,1}(x)\right)^2}{S_{n,0}(x)S_{n,2}(x) - \left(S_{n,1}(x)\right)^2} \\ &= 1 \end{aligned}$$

Q.E.D.

2 Practical exercise: Global bandwidth selection

```
> # models and helper functions
>
> library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.1     v tibble    3.2.1
v lubridate 1.9.4     v tidyr     1.3.1
v purrr     1.0.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
> # Generate one dataset
> gen_data <- function(n, alpha, beta, sigma2=1) {
+   x <- rbeta(n, alpha, beta)
+   y <- sin((x/3 + 0.1)^(-1)) + rnorm(n, sd = sqrt(sigma2))
+   tibble(x=x, y=y)
+ }
>
> # Cut into N blocks
> block_ids_by_quantile <- function(x, N) {
+   N <- max(1, as.integer(N)) # ensure at least 1 block
+   qs <- unique(quantile(x, probs = seq(0, 1, length.out = N+1), type = 1)) # avoid empty blocks by un
+   cut(x, breaks = qs, include.lowest = TRUE, labels = FALSE)
+ }
>
> # Fit in each block; return fitted values, second derivatives, and df used
> fit_by_block <- function(df, N) {
+   id <- block_ids_by_quantile(df$x, N)
+   df$block <- id
+   n <- nrow(df)
+   mhat <- numeric(n)
+   m2hat <- numeric(n)
+   df_used <- 0L
+
+   for (j in sort(unique(id))) {
+     idx <- which(id == j)
+     xj <- df$x[idx]
+     yj <- df$y[idx]
+     fit <- lm(yj ~ poly(xj, 4, raw = TRUE)) # ensure not fit with orthogonal polynomials
+     cf <- coef(fit)
+     # Fitted values mhat_j(x)
+     mhat[idx] <- cf[1] + cf[2]*xj + cf[3]*xj^2 + cf[4]*xj^3 + cf[5]*xj^4
+     # Second derivative m''(x) = 2*cf3 + 6*cf4*x + 12*cf5*x^2
+     m2hat[idx] <- 2*cf[3] + 6*cf[4]*xj + 12*cf[5]*xj^2
+     df_used <- df_used + length(cf)   # degree of freedom is 5 because of quartic poly
+   }
```

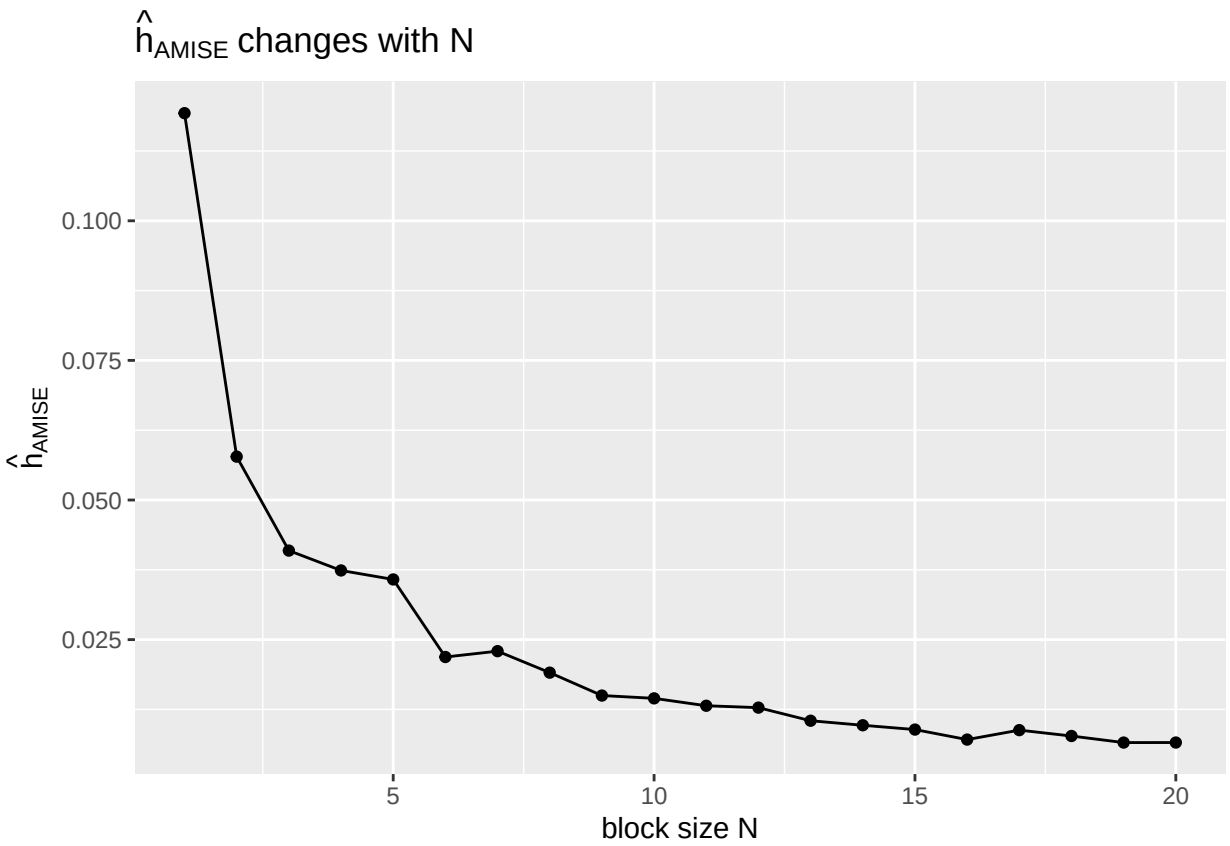
```
+   list(mhat = mhat, m2hat = m2hat, df_used = df_used)
+ }
>
> # Estimates 22(N) and ^2(N)
> theta22_hat <- function(m2hat) {
+   mean(m2hat^2)
+ }
> sigma2_hat <- function(y, mhat, df_used, n) {
+   sum((y - mhat)^2)/max(1, n - df_used) #prevent division by zero or negative values
+ }
>
> # Mallow's Cp
> Cp_of_N <- function(N, y, x) {
+   n <- length(y)
+   Nmax <- max(min(floor(n/20), 5), 1)
+   # Fit at N and at Nmax
+   fitN <- fit_by_block(tibble(x=x, y=y), N)
+   fitNmx <- fit_by_block(tibble(x=x, y=y), Nmax)
+   rssN <- sum((y - fitN$mhat)^2)
+   rssNmx <- sum((y - fitNmx$mhat)^2)
+   cp <- rssN / (rssNmx / (n - 5*Nmax)) - (n - 10*N)
+   tibble(N=N, Cp=cp, rssN=rssN, rssNmax=rssNmx, Nmax=Nmax)
+ }
>
> # h_AMISE (support length is 1 for Beta distribution)
> h_AMISE_hat <- function(n, sigma2_hat, theta22_hat, support_len = 1) {
+   n^(-1/5) * ((35 * sigma2_hat * support_len) / theta22_hat)^(1/5)
+ }
```

```
> #test(one run)
> set.seed(517)
> n <- 500
> df <- gen_data(n, alpha=2, beta=5, sigma2=1)
>
> # Choose N by Cp
> candidateN <- 1:max(min(floor(n/20), 5), 1)
> cp_tbl <- bind_rows(lapply(candidateN, Cp_of_N, y=df$y, x=df$x))
> N_opt <- cp_tbl$N[ which.min(cp_tbl$Cp) ]
> # Fit with optimal N
> fit_opt <- fit_by_block(df, N_opt)
> theta22_hat_val <- theta22_hat(fit_opt$m2hat)
> sigma2_hat_val <- sigma2_hat(df$y, fit_opt$mhat, df_used = 5*N_opt, n = n)
> hhat <- h_AMISE_hat(n, sigma2_hat_val, theta22_hat_val, support_len = 1)
>
> list(N_opt = N_opt,
+      theta22_hat = theta22_hat_val,
+      sigma2_hat = sigma2_hat_val,
+      h_AMISE_hat = hhat)
```

```
$N_opt
[1] 2

$theta22_hat
```

```
[1] 114538.5

$sigma2_hat
[1] 1.05229

$h_AMISE_hat
[1] 0.05776388
```

```
> # how does h_AMISE change with N
> candidateN <- 1:20
> differentN_tbl <- lapply(candidateN, function(N) {
+   fitN <- fit_by_block(df, N)
+   tibble(
+     N = N,
+     theta22 = theta22_hat(fitN$m2hat),
+     sigma2 = sigma2_hat(df$y, fitN$mhat, df_used = 5*N, n = n),
+     hhat = h_AMISE_hat(n, sigma2 = sigma2, theta22 = theta22, support_len = 1)
+   )
+ }) %>% bind_rows()
>
> ggplot(differentN_tbl, aes(N, hhat)) +
+   geom_line() + geom_point() +
+   labs(title = hat(h)[AMISE] ~ "changes with N",
+        x = "block size N", y = expression(hat(h)[AMISE]))
```



(1) As shown in above plot, $\hat{h}_{\text{AMISE}}$ tends to decrease as block size N increases.

(The theoretical h_{AMISE} is independent of the block size N.)

Explanation:

According to the formula of $\hat{h}_{\text{AMISE}}$, with fixed n and |supp(X)|, $\hat{h}_{\text{AMISE}}$ depends on $\hat{\sigma}^2$ and $\hat{\theta}_{22}$.

$$\hat{h}_{\text{AMISE}} \propto \left(\frac{\hat{\sigma}^2}{\hat{\theta}_{22}} \right)^{1/5}$$

As N increases, there are fewer data in each block, fits are more flexible and $\hat{m}$ tends to overfit, thus residuals are lower, leading to smaller $\hat{\sigma}^2$; while the curvature $\hat{m}''$ is wiggly, leading to bigger $\hat{\theta}_{22}$. Thus, as N increases, $\hat{\sigma}^2$ tends to decrease while $\hat{\theta}_{22}$ tends to increase, so their ratio tends to decreases. Therefore, $\hat{h}_{\text{AMISE}}$ tends to decrease as block size N increases.

However, as seen in the plot, the trend is not perfectly monotonic. Although each block has approximately the same number of observations due to quantile-based blocking, the geometry of blocks can vary significantly depending on the shape of the distribution of X. In regions where X has low density (like in skewed Beta distributions), blocks span wider intervals, which can lead to unstable polynomial fits and erratic estimates of the second derivative $\hat{m}''$. This makes $\hat{\theta}_{22}$ unstable, especially for large N, contributing to fluctuations in $\hat{h}_{\text{AMISE}}$.

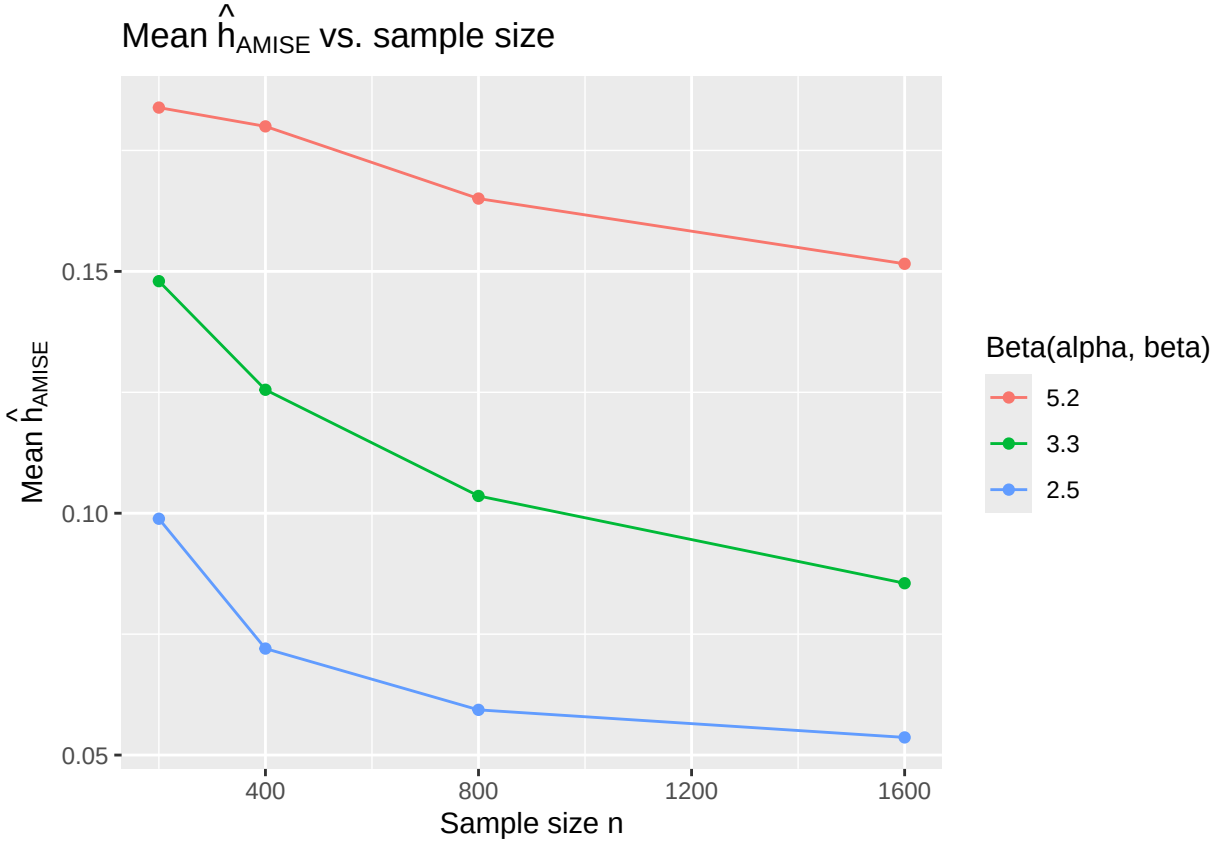
(2) N should depend on n. Because with more data, we can have larger N (finer blocking) without overfitting as each block can still have enough observations.

```
> # different n and different Beta parameters
> grid_n <- c(200, 400, 800, 1600)
> grid_ab <- tibble(alpha = c(2,3,5), beta = c(5,3,2))  #right-skew, symmetric, left-skew
> rep_each <- 200  # repetitions for stability
>
> run_cell <- function(n, alpha, beta, reps = rep_each) {
+   Nmax <- max(min(floor(n/20), 5), 1)
+   candidateN <- 1:Nmax
+
+   out_tbl <- purrr::map_dfr(1:reps, function(.r) {
+     df <- gen_data(n, alpha, beta, sigma2 = 1)
+     cp_tbl <- bind_rows(lapply(candidateN, Cp_of_N, y = df$y, x = df$x))
+     N_opt <- cp_tbl$N[which.min(cp_tbl$Cp)]
+
+     fit <- fit_by_block(df, N_opt)
+     th22 <- theta22_hat(fit$m2hat)
+     sg2 <- sigma2_hat(df$y, fit$mhat, df_used = 5*N_opt, n = n)
+     hhat <- h_AMISE_hat(n, sg2, th22, 1)
+
+     tibble(N_opt = N_opt, theta22 = th22, sigma2 = sg2, hhat = hhat)
+   })
+
+   out_tbl %>% mutate(n = n, alpha = alpha, beta = beta, .before = 1)
+ }
>
> set.seed(517)
> big_tbl <- bind_rows(
+   lapply(grid_n, function(nn)
+     bind_rows(lapply(1:nrow(grid_ab), function(i)
+       run_cell(nn, grid_ab$alpha[i], grid_ab$beta[i], reps=rep_each)
```

```
+     ))
+   )
+ )
>
> summ_tbl <- big_tbl %>%
+   group_by(n, alpha, beta) %>%
+   summarise(hhat_mean = mean(hhat), hhat_sd = sd(hhat),
+             Nopt_mean = mean(N_opt), .groups="drop")
>
> summ_tbl
```

```
# A tibble: 12 x 6
       n alpha  beta hhat_mean hhat_sd Nopt_mean
   <dbl> <dbl> <dbl>     <dbl>   <dbl>     <dbl>
 1   200     2     5    0.0989 0.0424       1.66
 2   200     3     3    0.148  0.0412       1.27
 3   200     5     2    0.184  0.0642       1.18
 4   400     2     5    0.0720 0.0255       2.00
 5   400     3     3    0.126  0.0355       1.4
 6   400     5     2    0.180  0.0629       1.17
 7   800     2     5    0.0593 0.0104       2.13
 8   800     3     3    0.104  0.0286       1.66
 9   800     5     2    0.165  0.0434       1.14
10  1600     2     5    0.0537 0.00658      2.12
11  1600     3     3    0.0855 0.0221       2.00
12  1600     5     2    0.152  0.0399       1.14
```

```
> # How does hhat_mean behave for different sample size n
> ggplot(summ_tbl, aes(x = n, y = hhat_mean, color = interaction(alpha, beta))) +
+   geom_line() + geom_point() +
+   labs(title = expression("Mean" ~ hat(h)[AMISE] ~ "vs. sample size"),
+        x = "Sample size n", y = expression("Mean" ~ hat(h)[AMISE]),
+        color = "Beta(alpha, beta)")
```



(3) How does $\hat{h}_{\text{AMISE}}$ behave with different sample size n? (The impact of the amount of available information)

As the sample size n increases, the estimated bandwidth $\hat{h}_{\text{AMISE}}$ tends to decrease across all Beta(α, β) distributions. This is consistent with the formula:

$$h_{\text{AMISE}} \propto n^{1/5}$$

With a larger sample size n, we can estimate local behavior more precisely (with less variance), so using narrower bandwidths becomes optimal. In other words, having more data provides more information, which lets us capture more local features.

(4) How does $\hat{h}_{\text{AMISE}}$ behave under different Beta(α, β) parameters? (The impact of the shape of the distribution of the covariate) ("What happens when the number of observations varies a lot between different regions in the support of X ? How is this linked to the parameters of the Beta distribution?")

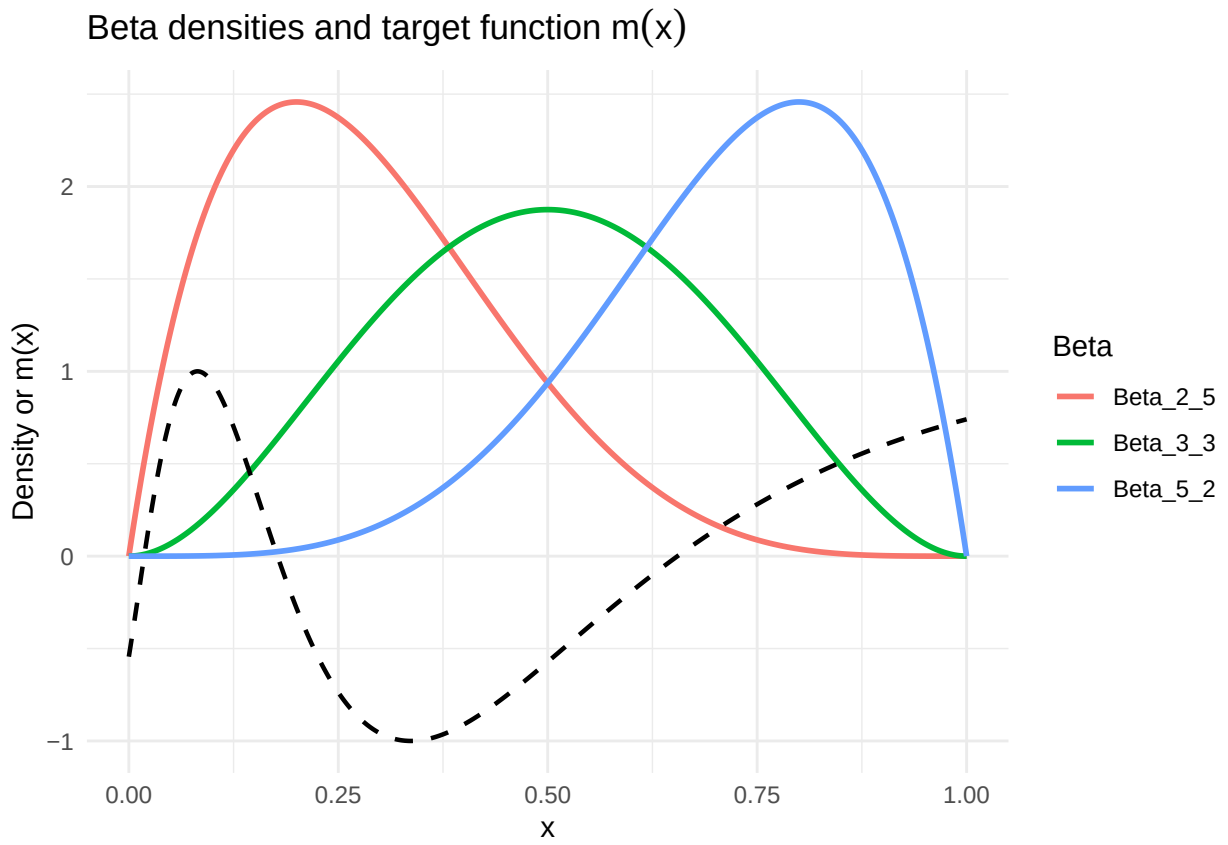
Intuitively, we might think the symmetric case Beta(3,3) would yield the smallest $\hat{h}_{\text{AMISE}}$ values, because the data is more evenly distributed and offers more balanced coverage across the support $[0, 1]$.

However, interestingly, from the plot above we see $\hat{h}_{\text{AMISE}}$ values for Beta(2,5) are consistently lower than those for Beta(3,3).

A possible explanation is that the regression function $m(x) = \sin\left(\frac{1}{x/3+0.1}\right)$ has higher curvature near $x = 0$ (as plot below shows), where Beta(2,5) concentrates more observations. This denser sampling in regions of high curvature may lead to more accurate estimation of $m''(x)$ and thus smaller $\hat{h}_{\text{AMISE}}$ values.

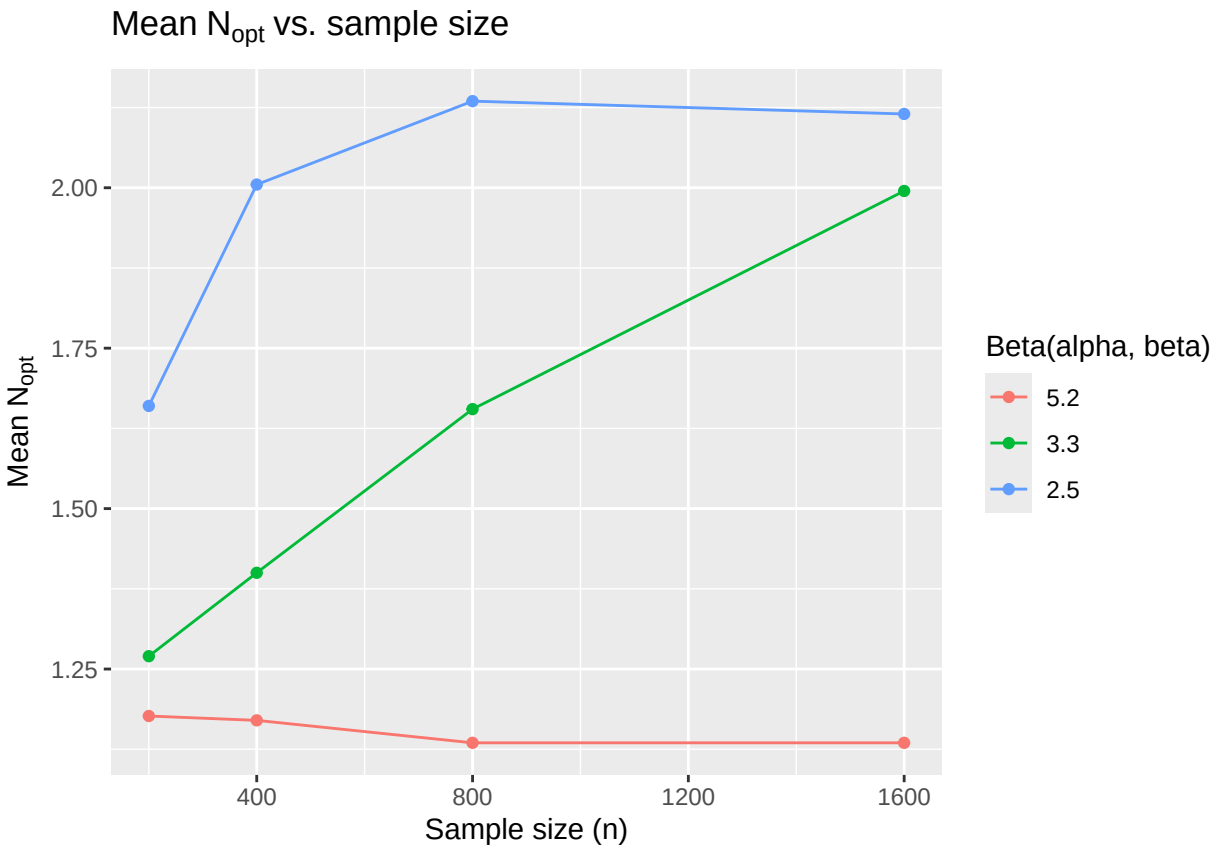
This shows that not just the overall balance of X matters, but also how the covariate distribution aligns with the complexity of the regression function.

```
> library(tidyverse)
>
> # target function
> m <- function(x) sin(1 / (x / 3 + 0.1))
>
> # Grid of x values
> x_vals <- seq(0, 1, length.out = 1000)
>
> # Densities of X under different Beta distributions
> dens_df <- tibble(
+   x = x_vals,
+   Beta_3_3 = dbeta(x, 3, 3),
+   Beta_2_5 = dbeta(x, 2, 5),
+   Beta_5_2 = dbeta(x, 5, 2),
+   m_x = m(x)
+ )
>
>
> dens_long <- dens_df %>%
+   pivot_longer(cols = starts_with("Beta"), names_to = "Beta", values_to = "density")
>
> # Plot density vs. function
> ggplot(dens_long, aes(x = x, y = density, color = Beta)) +
+   geom_line(size = 1) +
+   geom_line(aes(y = m_x), color = "black", linetype = "dashed", size = 0.8, data = distinct(dens_df,
+   labs(title = expression("Beta densities and target function " * m(x)),
+        y = "Density or m(x)",
+        x = "x") +
+   theme_minimal()
```



The plot below below again supports the idea that the block size N should depend on n. (However, the effect may be less pronounced or even plateau for highly skewed covariate distributions like Beta(2,5), where some regions remain sparsely populated regardless of n)

```
> # How does Nopt behave for different sample size n
> ggplot(summ_tbl, aes(x = n, y = Nopt_mean, color = interaction(alpha, beta))) +
+   geom_line() + geom_point() +
+   labs(title = expression("Mean" ~ N[opt] ~ "vs. sample size"),
+        x = "Sample size (n)", y = expression("Mean" ~ N[opt]),
+        color = "Beta(alpha, beta)")
```


Mean N_{opt} vs. sample size
2.00
1.75
1.50
1.25
Mean N_{opt}
400
800
1200
1600
Sample size (n)
Beta(alpha, beta)
5.2
3.3
2.5