2. Let $X_1, \ldots, X_n \sim f$ and let $\widehat{f}_n$ be the kernel density estimator using the boxcar kernel:

$$K(x) = \begin{cases} 1 & -\frac{1}{2} < x < \frac{1}{2} \\ 0 & \text{otherwise.} \end{cases}$$

(a) Show that

$$\mathbb{E}(\widehat{f}(x)) = \frac{1}{h} \int_{x-(h/2)}^{x+(h/2)} f(y)dy$$

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

Definition 8.2. The kernel density estimator with bandwidth h and kernel K:

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

$$\begin{aligned}
\mathbb{E}\hat{f}(x) &= \frac{1}{n} \sum_{i=1}^{n} \frac{1}{h} \mathbb{E} K\left(\frac{x-X_i}{h}\right) \\
&= \frac{1}{n} \sum_{i=1}^{n} \frac{1}{h} \int_{\left|\frac{x-y_i}{h}\right| \le \frac{1}{2}} f(y_i)\, dy_i \\
&= \frac{1}{n} \sum_{i=1}^{n} \frac{1}{h} \int_{x-\frac{1}{2}h}^{x+\frac{1}{2}h} f(y_i)\, dy_i = \frac{1}{h} \int_{x-\frac{1}{2}h}^{x+\frac{1}{2}h} f(y)\, dy
\end{aligned}$$

and

$$\mathbb{V}(\widehat{f}(x)) = \frac{1}{nh^2} \left[\int_{x-(h/2)}^{x+(h/2)} f(y)dy - \left(\int_{x-(h/2)}^{x+(h/2)} f(y)dy \right)^2 \right].$$

$$\begin{aligned}
\mathbb{E}\hat{f}^2(x) &= \frac{1}{n^2} \sum_{i,j} \frac{1}{h^2} K\left(\frac{x-X_i}{h}\right) K\left(\frac{x-X_j}{h}\right) \\
&= \frac{1}{n^2 h^2} \left[\sum_{i=j=1}^{n} \mathbb{E} K\left(\frac{x-X_i}{h}\right) + \sum_{i \neq j} \mathbb{E} K\left(\frac{x-X_i}{h}\right) K\left(\frac{x-X_j}{h}\right) \right] \\
&= \frac{1}{nh^2} \int_{x-\frac{1}{2}h}^{x+\frac{1}{2}h} f(y)dy - \frac{n-1}{nh^2} \left(\int_{x-\frac{1}{2}h}^{x+\frac{1}{2}h} f(y)dy \right)^2
\end{aligned}$$

$$\begin{aligned}
\Rightarrow Var(\hat{f}(x)) &= \mathbb{E}\hat{f}^2(x) - (\mathbb{E}\hat{f}(x))^2 \\
&= \frac{1}{nh^2} \left[\int_{x-\frac{1}{2}h}^{x+\frac{1}{2}h} f(y)\,dy - \left(\int_{x-\frac{1}{2}h}^{x+\frac{1}{2}h} f(y)\,dy \right)^2 \right]
\end{aligned}$$

(b) Show that if $h \to 0$ and $nh \to \infty$ as $n \to \infty$ then $\widehat{f}_n(x) \xrightarrow{P} f(x)$.

Fix any $\varepsilon > 0$

$P(|X - \mathbb{E}X| > \epsilon) \leq \frac{\mathrm{Var}(X)}{\epsilon^2}$

$$\mathbb{P}\{|\hat{f}_n(x) - f(x)| > \varepsilon\}$$

$$= \mathbb{P}\{|\hat{f}_n(x) - \mathbb{E}\hat{f}_n(x) + \mathbb{E}\hat{f}_n(x) - f(x)| > \varepsilon\}$$

$$\leq \mathbb{P}\{|\hat{f}_n(x) - \mathbb{E}\hat{f}_n(x)| + |\mathbb{E}\hat{f}_n(x) - f(x)| > \varepsilon\}$$

$$\mathbb{P}\{|\hat{f}_n(x) - \mathbb{E}\hat{f}_n(x)| > \varepsilon\} \leq \frac{1}{\varepsilon^2} Var(\hat{f}_n(x)) \to \frac{1}{\varepsilon^2}\left(\frac{1}{nh} f(x) - \frac{1}{n} f^2(x)\right)$$

$\to 0$ as $n \to \infty$, $nh \to \infty$

$$\mathbb{E}\hat{f}_n(x) = \frac{1}{h}\int_{x-\frac{1}{2}h}^{x+\frac{1}{2}h} f(y)\,dy \xrightarrow{h \to 0} \frac{d}{dx}\int_{x-\frac{1}{2}h}^{x+\frac{1}{2}h} f(y)\,dy = f(x)$$

$$\mathbb{P}\{|\hat{f}_n(x) - f(x)| < \varepsilon\} \to 0 \Rightarrow \hat{f}_n(x) \xrightarrow{P} f(x)$$

4. Prove equation 6.35.

$$\widehat{J}(h) = \frac{1}{hn^2}\sum_i \sum_j K^*\left(\frac{X_i - X_j}{h}\right) + \frac{2}{nh}K(0) + O\left(\frac{1}{n^2}\right) \qquad (6.35)$$

where $K^*(x) = K^{(2)}(x) - 2K(x)$ *and* $K^{(2)}(z) = \int K(z-y)K(y)dy$.

$$\hat{J}(h) = \int \hat{f}^2(x)\,dx - \frac{2}{n}\sum_{i=1}^{n} \hat{f}_{-i}(X_i)$$

$$= \frac{1}{n^2h^2}\int \left[\sum_{i=j=1}^{n} K^2\left(\frac{x-X_i}{h}\right) + \sum_{i\neq j}^{n} K\left(\frac{x-X_i}{h}\right)K\left(\frac{x-X_j}{h}\right)\right] dx$$

$$- \frac{2}{n}\sum_{i=1}^{n}\frac{1}{(n-1)h}\sum_{j\neq i} K\left(\frac{x-X_j}{h}\right)$$

$$= \frac{1}{n^2h}\sum_i\sum_j \int K\left(\frac{x-X_i}{h}\right)K\left(\frac{x-X_j}{h}\right)dx - \frac{2}{n}\sum_{i=1}^{n}\frac{1}{(n-1)h}\sum_{j\neq i}K\left(\frac{x-X_j}{h}\right)$$

By Taylor expansion,

$$= \frac{1}{n^2 h} \sum_i \sum_j K^{(2)}\left(\frac{X_i - X_j}{h}\right) - 2K\left(\frac{X_i - X_j}{h}\right) + \frac{2}{nh} K(0) + O\left(\frac{1}{n^2}\right)$$

STA3007_hw10_codes

Yuzhou Peng

2025-04-20

```
library(np)
```

```
## Nonparametric Kernel Methods for Mixed Datatypes (version 0.60-18)
## [vignette("np_faq",package="np") provides answers to frequently asked questions]
## [vignette("np",package="np") an overview]
## [vignette("entropy_np",package="np") an overview of entropy-based methods]
```

```
library(ggplot2)
```

```
ceodat <- read.table("C:\\Users\\Penguin\\Desktop\\STA3007\\ceodat.txt")
ceodat <- ceodat[-31,]

ceodat <- ceodat[-1,]

colnames(ceodat) <- c("AGE", "SAL")
rownames(ceodat) <- 1:nrow(ceodat)

Y <- ceodat$SAL
X <- ceodat$AGE

Y <- as.numeric(Y)
X <- as.numeric(X)
```

#Histogram

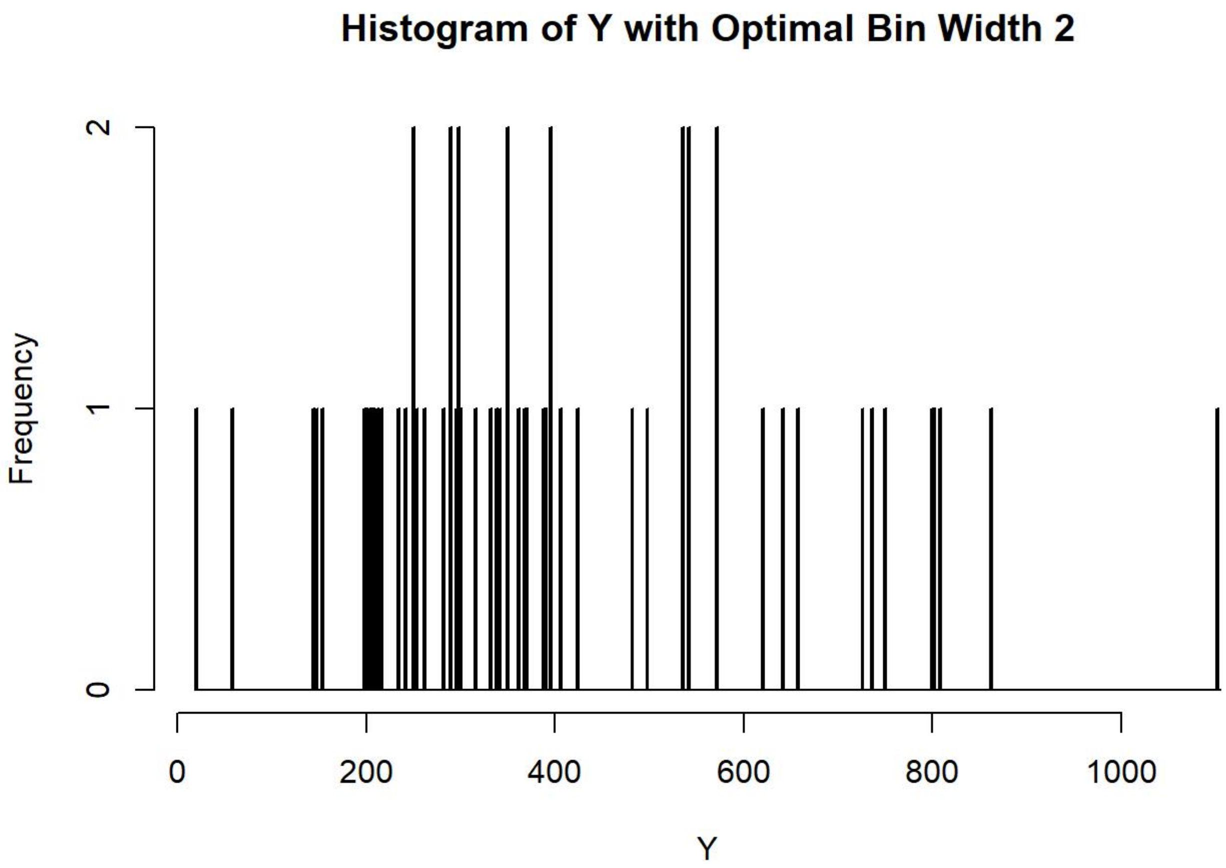
```
LSCV_hist <- function(Y, h) {
  n <- length(Y)
  if (n < 2) stop("Need at least two data points")
  range_Y <- range(Y)
  # Create breaks covering the data range with bin width h
  breaks <- seq(from = floor(range_Y[1]/h)*h - h,
               to = ceiling(range_Y[2]/h)*h + h,
               by = h)
  hist_counts <- hist(Y, breaks = breaks, plot = FALSE)$counts
  term1 <- sum(hist_counts^2) / (n^2 * h)
  sum_term2 <- sum(hist_counts * (hist_counts - 1))
  term2 <- 2 * sum_term2 / (n * h * (n - 1))
  return(term1 - term2)
}
```

```
h_values <- seq(0.1, 2, by = 0.1)  # Adjust based on data spread
```

```
lscores <- sapply(h_values, function(h) LSCV_hist(Y, h))
```

```
optimal_h <- h_values[which.min(lscores)]
```

```
hist(Y, breaks = seq(min(Y) - optimal_h, max(Y) + optimal_h, by = optimal_h),
     main = paste("Histogram of Y with Optimal Bin Width", round(optimal_h, 2)),
     xlab = "Y", col = "lightblue", border = "black")
```



Kernel Method

```
data <- data.frame(X = X, Y = Y)
n <- nrow(ceodat)

h_normal <- 1.06 * sd(X) * n^(-1/5)

# Fit kernel regression model with the computed bandwidth
bw <- npregbw(
  formula = Y ~ X,
  data = data,
  bws = h_normal,     # Use the precomputed bandwidth
  bwtype = "fixed",   # Fix the bandwidth (no cross-validation)
  ckertype = "gaussian" # Gaussian kernel
)
```

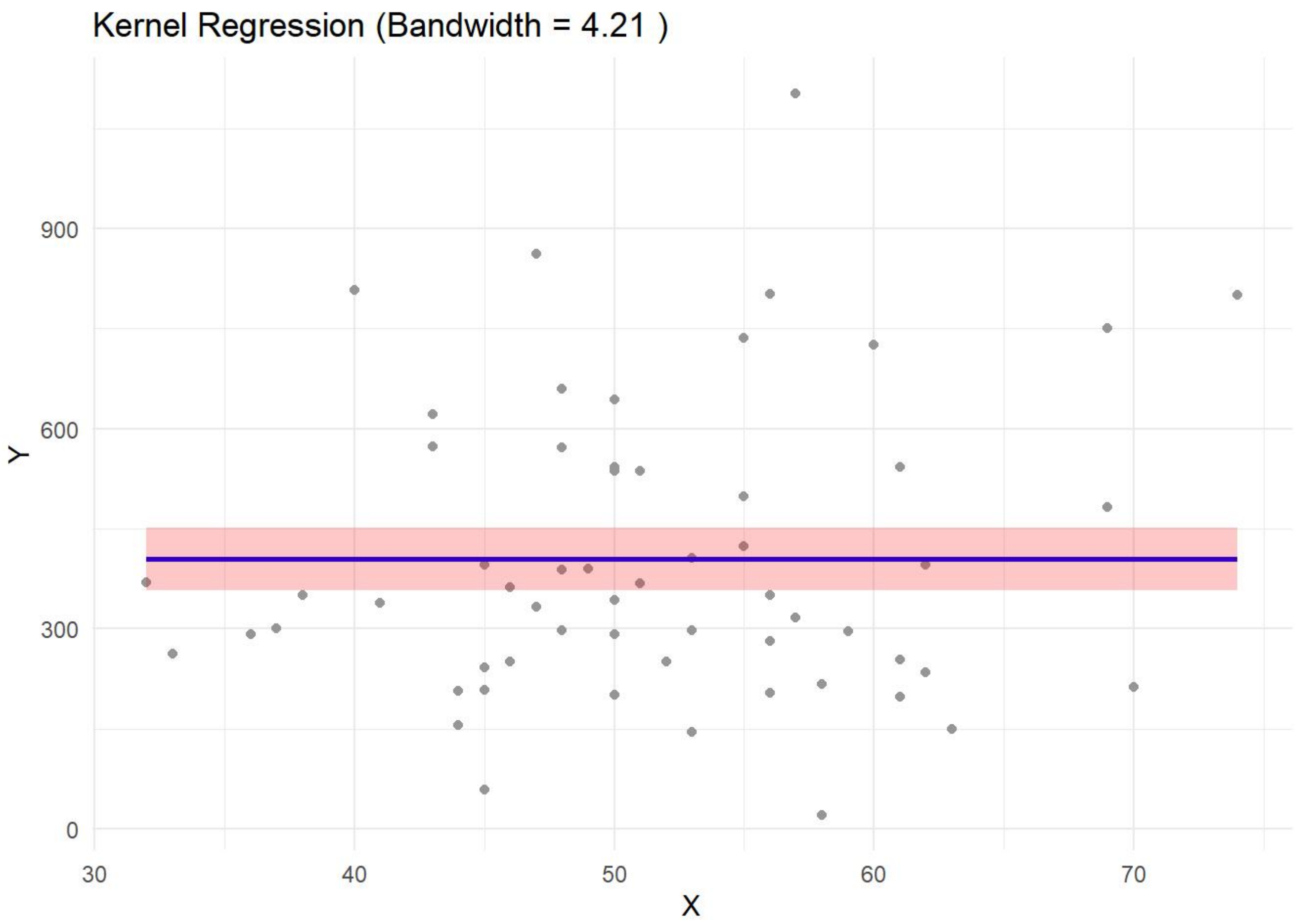
```
##
Multistart 1 of 1 |
Multistart 1 of 1 |
Multistart 1 of 1 |
Multistart 1 of 1 /
Multistart 1 of 1 |
Multistart 1 of 1 |
```

```
# Fit the model
model <- npreg(bw)

#  Generate predictions and confidence intervals
x_grid <- seq(min(X), max(X), length.out = 100) # Grid of X values
pred <- predict(model, newdata = data.frame(X = x_grid), se.fit = TRUE) # Predictions with SEs

# Compute 95% confidence bands
conf_lower <- pred$fit - 1.96 * pred$se.fit
conf_upper <- pred$fit + 1.96 * pred$se.fit

# Plot results with confidence bands
ggplot() +
  geom_point(data = data, aes(x = X, y = Y), color = "gray60") +
  geom_line(aes(x = x_grid, y = pred$fit), color = "blue", linewidth = 1) +
  geom_ribbon(
    aes(x = x_grid, ymin = conf_lower, ymax = conf_upper),
    fill = "red", alpha = 0.2
  ) +
  labs(
    title = paste("Kernel Regression (Bandwidth =", round(h_normal, 2), ")"),
    x = "X", y = "Y"
  ) +
  theme_minimal()
```


Kernel Regression (Bandwidth = 4.21)
Y
X
0
300
600
900
30
40
50
60
70

```
data <- data.frame(X = X, Y = Y)
n <- nrow(ceodat)

h_normal <- 1.06 * sd(X) * n^(-1/5)

# Fit kernel regression model with the computed bandwidth
bw <- npregbw(
  formula = Y ~ X,
  data = data,
  bws = h_normal,     # Use the precomputed bandwidth
  bwtype = "fixed",   # Fix the bandwidth (no cross-validation)
  ckertype = "epanechnikov"
)
```

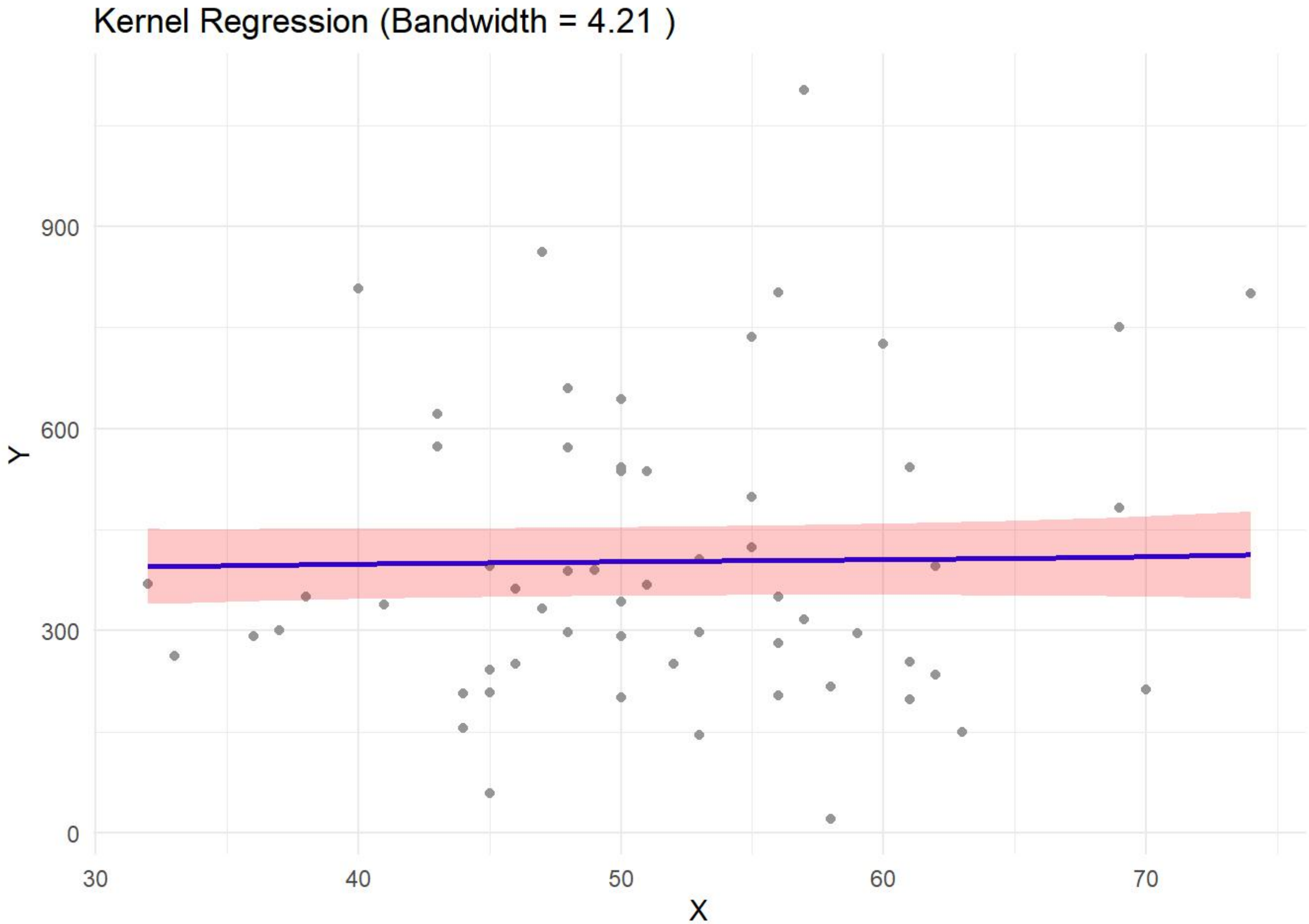
```
##
Multistart 1 of 1 |
Multistart 1 of 1 |
Multistart 1 of 1 |
Multistart 1 of 1 /
Multistart 1 of 1 |
Multistart 1 of 1 |
```

```
# Fit the model
model <- npreg(bw)

#  Generate predictions and confidence intervals
x_grid <- seq(min(X), max(X), length.out = 100) # Grid of X values
pred <- predict(model, newdata = data.frame(X = x_grid), se.fit = TRUE) # Predictions with SEs

# Compute 95% confidence bands
conf_lower <- pred$fit - 1.96 * pred$se.fit
conf_upper <- pred$fit + 1.96 * pred$se.fit

# Plot results with confidence bands
ggplot() +
  geom_point(data = data, aes(x = X, y = Y), color = "gray60") +
  geom_line(aes(x = x_grid, y = pred$fit), color = "blue", linewidth = 1) +
  geom_ribbon(
    aes(x = x_grid, ymin = conf_lower, ymax = conf_upper),
    fill = "red", alpha = 0.2
  ) +
  labs(
    title = paste("Kernel Regression (Bandwidth =", round(h_normal, 2), ")"),
    x = "X", y = "Y"
  ) +
  theme_minimal()
```


Kernel Regression (Bandwidth = 4.21)
900
600
300
0
Y
30
40
50
60
70
X