20_practice

Randy

12/10/2021

Linear Smoothing

Nadaraya-Waston Estimator

$$\hat{r}_n(x) = \sum_{i=1}^{n} l_i(x) Y_i$$

weight function

$$l_i(x) = \frac{K\left(\frac{x-x_i}{h}\right)}{\sum_{j=1}^{n} K\left(\frac{x-x_j}{h}\right)}$$

risk of Nadaraya-Waston estimator

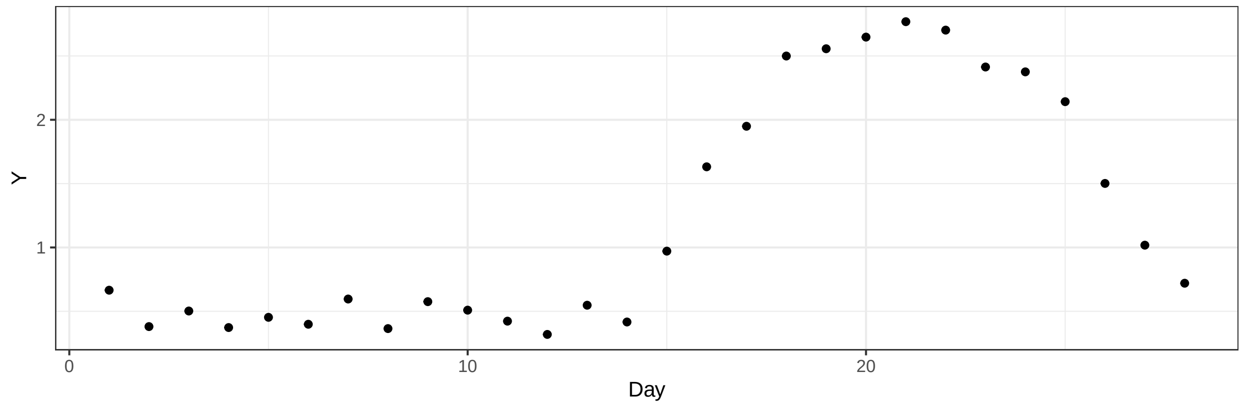
$$R(\hat{r}_n,\, r) = \frac{h_n^4}{4}\left(\int x^2 K(x)dx\right)^2 \int \left(r''(x) + 2r'(x)\frac{f'(x)}{f(x)}K(x)dx\right)^2 dx + \frac{\sigma^2 \int K^2(x)dx}{nh_n}\int \frac{1}{f(x)}dx + o(nh_n^{-1}) + o(h_n^4)$$

as $h_n \to 0$ and $nh_n \to \infty$

Import the dataset

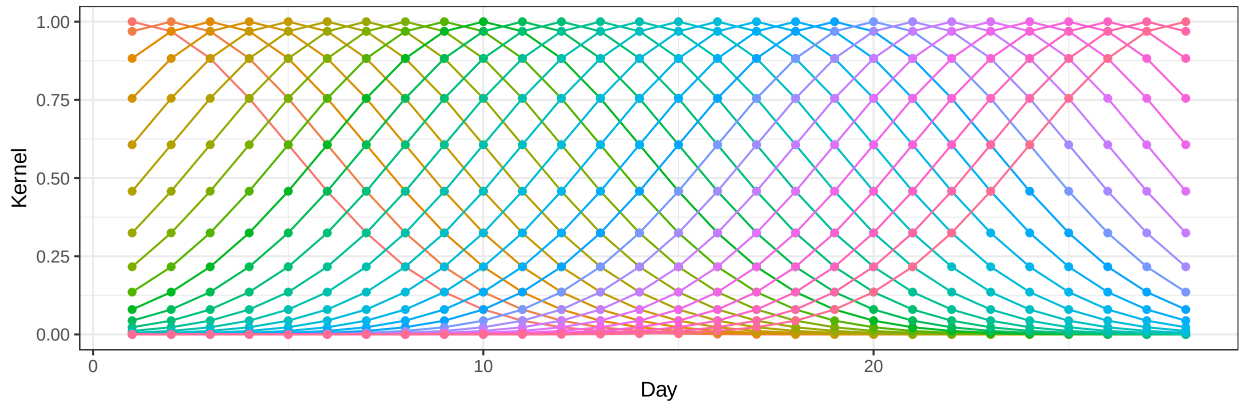
Outcomes

```
day <- progsim_one$daystacked
y <- progsim_one$prog_y
ggplot() +
  geom_point(aes(day, y)) +
  theme_bw() +
  xlab("Day") +
  ylab("Y")
```

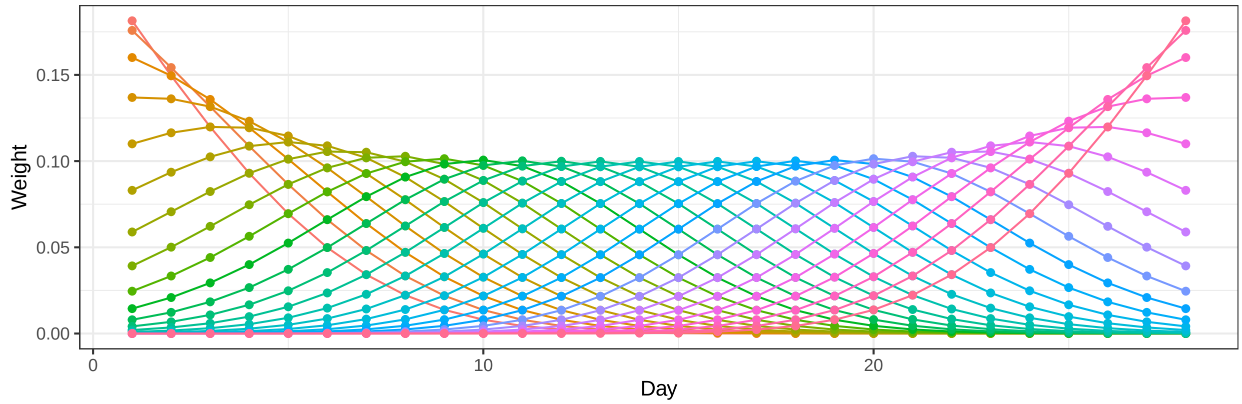


Window size

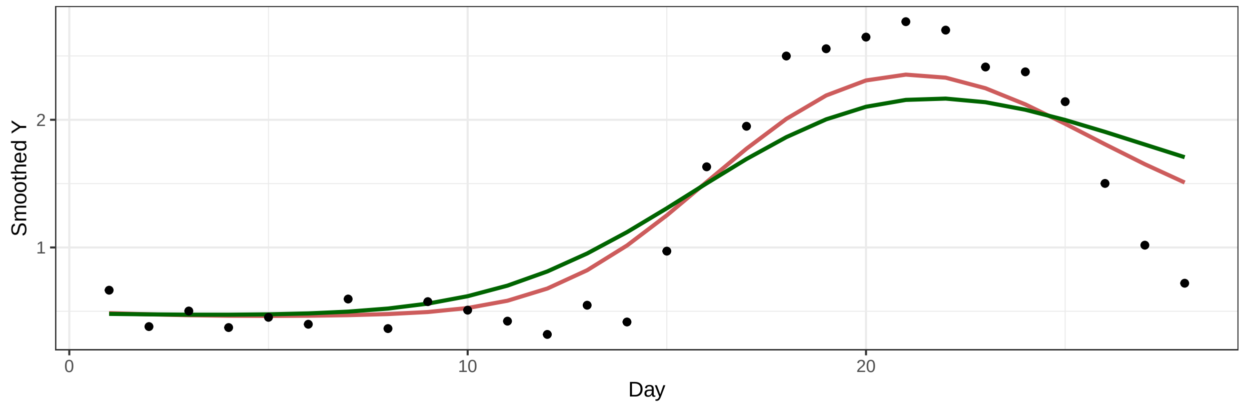
```
## window size of 3 ---------------------------------------------
l1 <- 3
kernel_se <- matrix(data = NA, nrow = length(day), ncol = length(day))
for (i in 1:length(day)) {
  for (j in 1:length(day)) {
    kernel_se[i, j] <- exp(-(day[i] - day[j])^2 / (2 * l1^2))
  }
}
weight1 <- t(kernel_se) / rowSums(kernel_se, na.rm = T)
y_hat1 = weight1 %*% y

## window size of 4 ---------------------------------------------
l1 <- 4
kernel_se <- matrix(data = NA, nrow = length(day), ncol = length(day))
for (i in 1:length(day)) {
  for (j in 1:length(day)) {
    kernel_se[i, j] <- exp(-(day[i] - day[j])^2 / (2 * l1^2))
  }
}

weight2 <- t(kernel_se) / rowSums(kernel_se, na.rm = T)
y_hat2 = weight2 %*% y

# matplot(day, weight2, "l")

## the kernel
autoplot(zoo::zoo(kernel_se), facet = NULL) +
  geom_point() +
  theme_bw() +
  theme(legend.position = "none") +
  xlab("Day") +
  ylab("Kernel")
```



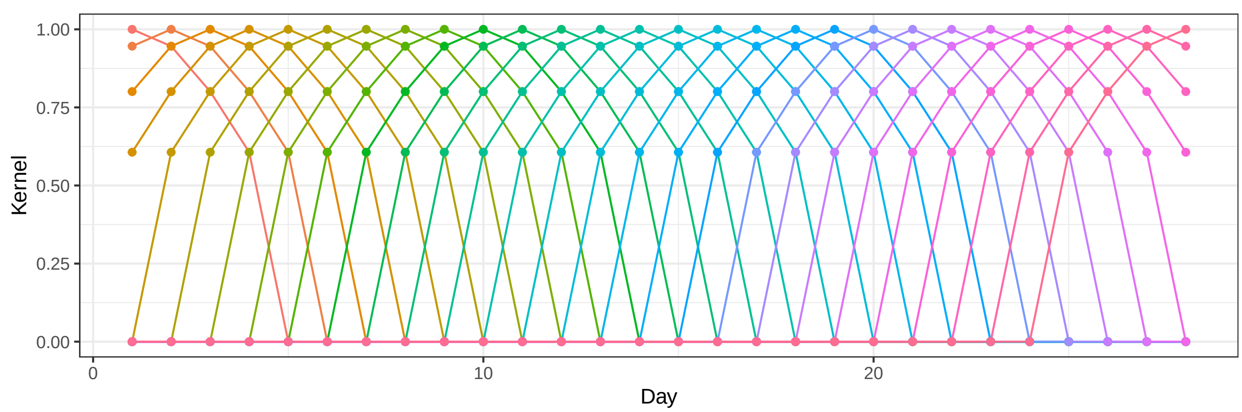

```r
## weight function sum up to 1 for each day
autoplot(zoo::zoo(weight2), facet = NULL) +
  geom_point() +
  theme_bw() +
  theme(legend.position = "none") +
  xlab("Day") +
  ylab("Weight")
```



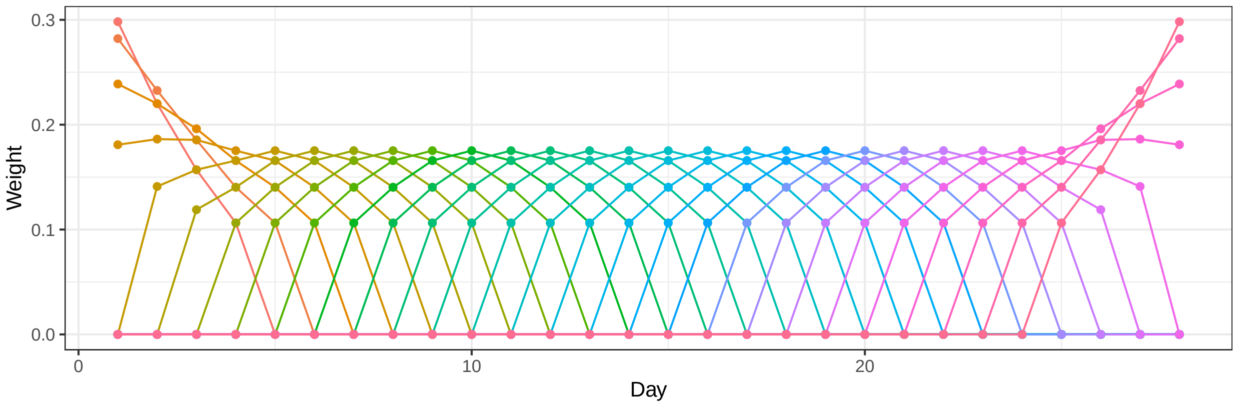

```r
## smoothing curve
ggplot() +
  geom_line(aes(day, y_hat1), color = "indianred", size = 1) +
  geom_line(aes(day, y_hat2), color = "darkgreen", size = 1) +
  geom_point(aes(day, y)) +
  theme_bw() +
  xlab("Day") +
  ylab("Smoothed Y")
```



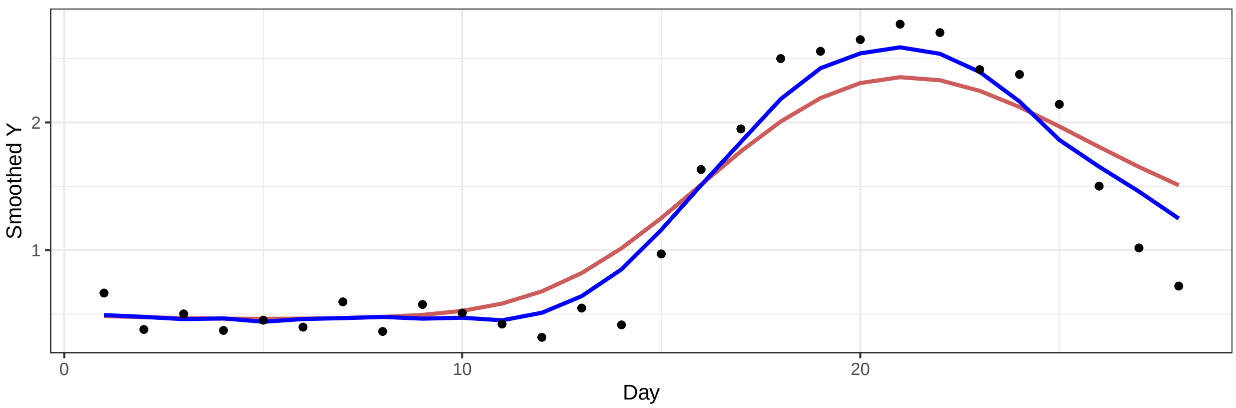

```
## beyond window size set as 0
l1 <- 3
kernel_seif <- matrix(data = NA, nrow = length(day), ncol = length(day))
for (i in 1:length(day)) {
  for (j in 1:length(day)) {
    kernel_seif[i, j] <- exp(-(day[i] - day[j])^2 / (2 * l1^2)) * ifelse(abs(day[i] - day[j]) > l1, 0,
  }
}
# View(kernel_seif)
weight3 <- t(kernel_seif) / rowSums(kernel_seif, na.rm = T)
y_hat3 = weight3 %*% y

## kernel function for each day
autoplot(zoo::zoo(kernel_seif), facet = NULL) +
  geom_point() +
  theme_bw() +
  theme(legend.position = "none") +
  xlab("Day") +
  ylab("Kernel")
```




```
## weight funtion for each day
autoplot(zoo::zoo(weight3), facet = NULL) +
  geom_point() +
  theme_bw() +
  theme(legend.position = "none") +
  xlab("Day") +
  ylab("Weight")
```




```r
## the estimation
ggplot() +
  geom_line(aes(day, y_hat1), color = "indianred", size = 1) +
  geom_line(aes(day, y_hat3), color = "blue", size = 1) +
  geom_point(aes(day, y)) +
  theme_bw() +
  xlab("Day") +
  ylab("Smoothed Y")
```



Summary

The graphs are consistent with the equation: there are the boundary bias and design bias $2r'(x)\frac{f'(x)}{f(x)}$ for kernel estimator.