EXERCISE 8

WEIYU LI

1. Let $X_1, \ldots, X_n$ *i.i.d.* with density $f(x)$ and denote $\hat{f}_h^{(r)}$ as the r-th derivative of the KDE estimator. Solve its asymptotic distribution.

Solve. Let $h = cn^{-1/(2r+5)}$. From Page 5 in the slides,

$$\begin{aligned} E\hat{f}_h^{(r)}(x) &= f^{(r)}(x) + \frac{1}{2}f^{(r+2)}(x)\kappa_{21}h^2 + o(h^2) \\ &= f^{(r)}(x) + \frac{1}{2}f^{(r+2)}(x)\kappa_{21}c^2 n^{-2/(2r+5)} + o(n^{-2/(2r+5)}). \end{aligned}$$

And from Page 7,

$$\begin{aligned} Var(\hat{f}_h^{(r)}(x)) &= \frac{f(x)}{nh^{2r+1}} \int [K^{(r)}(u)]^2 du + o\left(\frac{1}{nh^{2r+1}}\right) \\ &= \frac{f(x)}{n^{4/(2r+5)}c^{2r+1}} \int [K^{(r)}(u)]^2 du + o\left(\frac{1}{n^{4/(2r+5)}}\right). \end{aligned}$$

Therefore, by central limit theorem,

$$n^{2/(2r+5)}\left(\hat{f}_n^{(r)}(x) - f^{(r)}(x)\right) \to N\left(\frac{1}{2}f^{(r+2)}(x)\kappa_{21}c^2, \frac{f(x)\int [K^{(r)}(u)]^2 du}{c^{2r+1}}\right).$$

□

2. Generate 100 data points from the distribution

$$0.3N(0,1) + 0.7N(1, 0.3^2),$$

or rigorously speaking, from the pdf $f(x) = 0.3\phi(x;0,1) + 0.7\phi(x;1,0.3^2)$. Use the bandwidth choices from R package *kedd* to draw the curves of first derivative density estimators with different bandwidth in one plot. And draw the curve of the kernel CDF estimator.

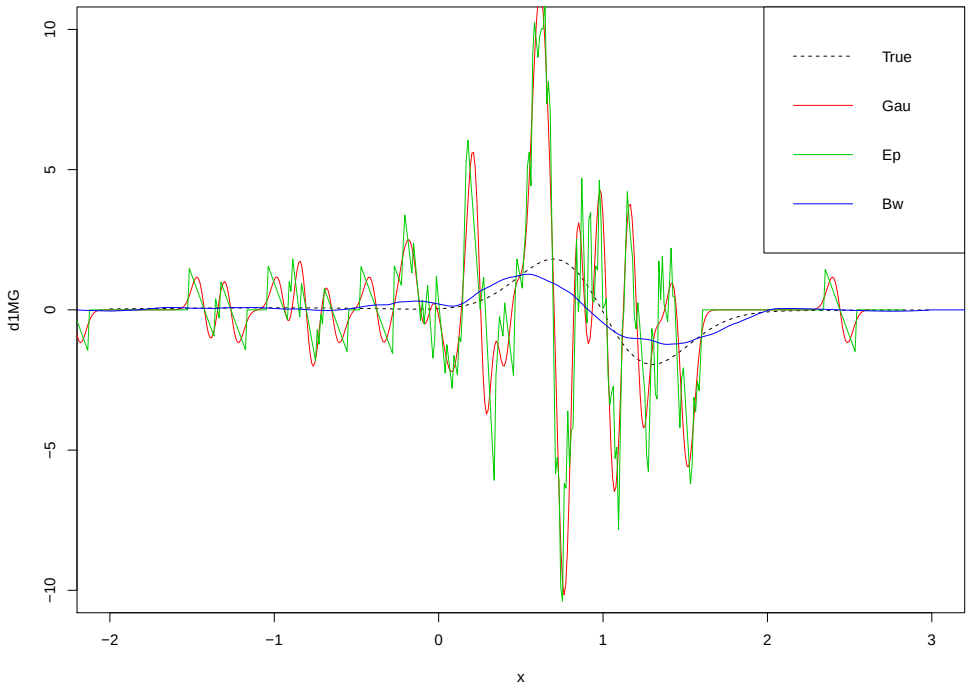
Hint. The generation step is similar to Exercise 7, and we use another function *kCDF* from R package *sROC* to obtain kernel CDF estimator in an easy way.

```
library(kedd)
set.seed(0)
n <- 100
pMG  <- function(x) {0.3 * pnorm(x, 0, 1) + 0.7 * pnorm(x, 1, 0.3)}
d1MG  <- function(x) {
  0.3 * dnorm(x, 0, 1) * (- x) + 0.7 * dnorm(x, 1, 0.3) * (1 - x) / 0.3^2
}
# the above are cdf and first derivative of the Mixed Gaussian distribution
rMG <- function(n){
```

Date: 2019/10/28.

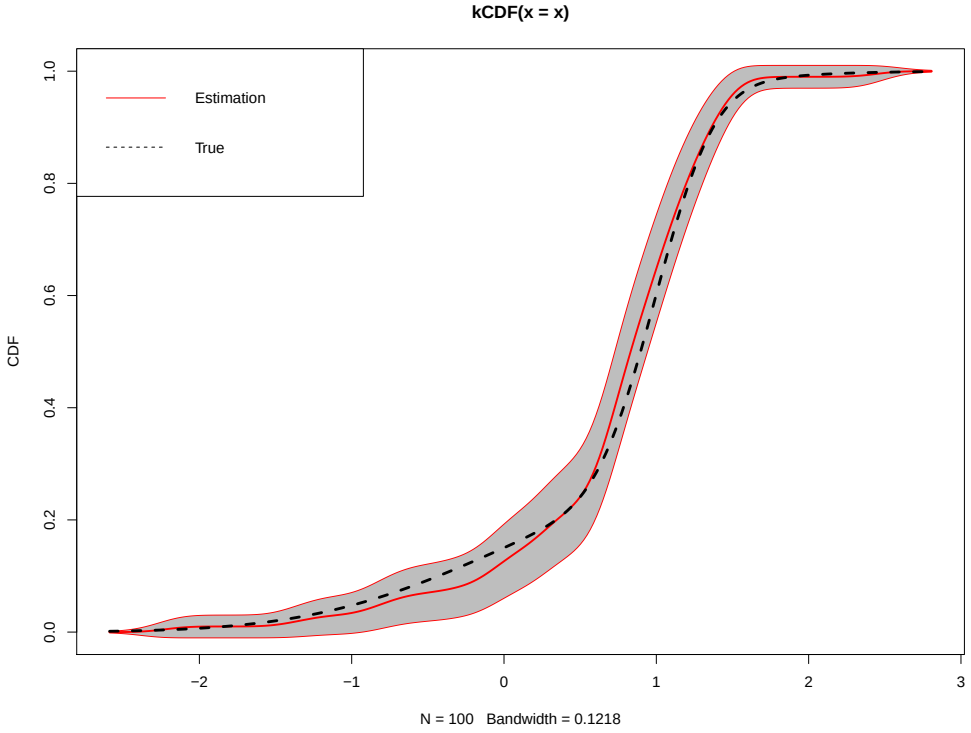
liweiyu@mail.ustc.edu.cn.

```
  # randomly generate n points from the Mixed Gaussian distribution
  r <- runif(n, 0, 1)
  x <- r
  ind <- which(r < 0.3) # index for those generated from N(0,1)
  x[ind] <- rnorm(length(ind), 0, 1)
  x[-ind] <- rnorm(n-length(ind), 1, 0.3)
  return(x)
}
x <- rMG(n)
fhat_gau=dkde(x,deriv.order = 1) # default kernel : gaussian
fhat_ep=dkde(x,deriv.order = 1, kernel = 'epanechnikov') # kernel : ep
fhat_bw=dkde(x,deriv.order = 1, kernel = 'biweight') # kernel : biweight
plot(d1MG, xlim = c(-2,3), ylim = c(-10,10), lty = 8)
lines(fhat_gau, col = 2)
lines(fhat_ep, col = 3)
lines(fhat_bw, col = 4)
legend('topright',legend = c('True', 'Gau', 'Ep', 'Bw'),
  col = 1:4, lty = c(8,1,1,1))
```




```
#estimate cdf
library(sROC)
Fhat <- kCDF(x)
plot(Fhat, col = 2)
```

```
lines(Fhat$x, pMG(Fhat$x), lty = 8, lwd = 3)
legend('topleft',legend = c('Estimation', 'True'), col = c(2,1), lty = c(1,8))
```



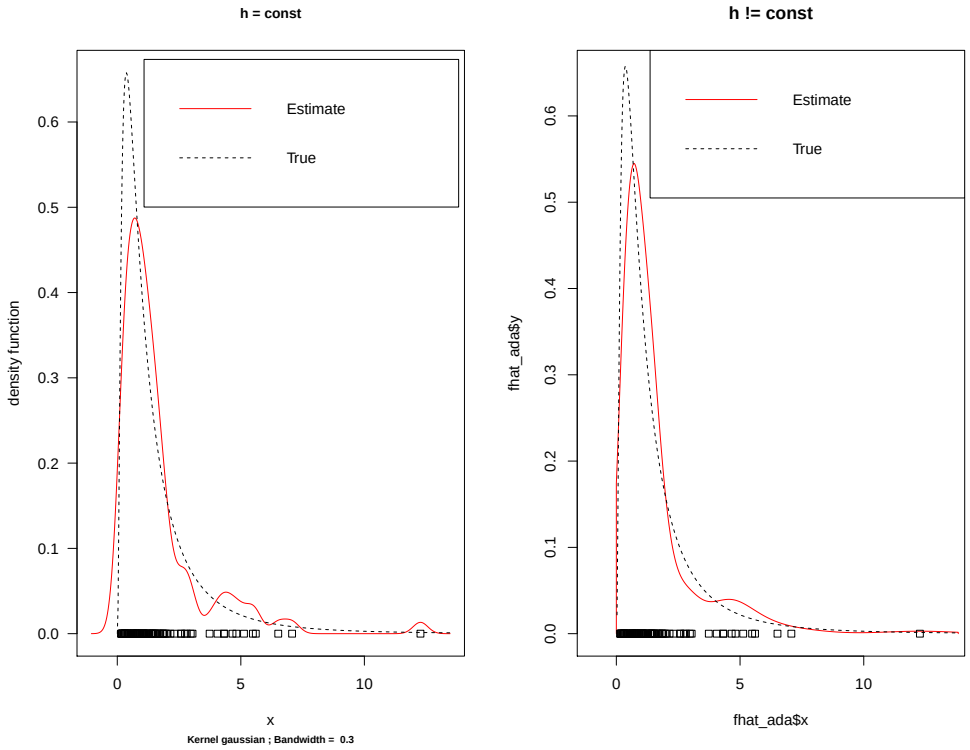
□

3. For adaptive KDE, reproduce the plot in Page 21 in the slides. (Refer to Fig. 3.15 in the reading material *Adaptive KDE.pdf***).**
Hint. To simplify the codes, we use another function *kde* from R package *IsoplotR* to obtain adaptive KDE.

```
library(kedd)
library(IsoplotR)
set.seed(0)
n <- 100
h <- 0.3
dlognorm  <- function(x) {dnorm(log(x), 0, 1) / x}
rlognorm <- function(n){
  # randomly generate n points from the log normal distribution
  r <- rnorm(n, 0, 1)
  x <- exp(r)
  return(x)
}
x <- rlognorm(n)
fhat <- dkde(x, deriv.order = 0, h = h)
```

```
# h = const
par(mfrow = c(1, 2))
plot(fhat, dlognorm, main = 'h = const')
points(x,rep(0,n), pch = 0)

# h != const
fhat_ada <- kde(x, adaptive = T, plot = F)
plot(fhat_ada$x, fhat_ada$y, 'l', col = 2, ylim = c(0, 0.7), main = 'h != const')
lines(fhat_ada$x, dlognorm(fhat_ada$x), 'l', lty = 2)
legend('topright',legend = c('Estimate', 'True'), col = c(2,1), lty = c(1,8))
points(x,rep(0,n), pch = 0)
```



We can observe from the plots that adaptive KDE better recovers the spike signal, while it's less sensitive to the outliers. □

4. For boundary correction KDE, read the reading material. Reproduce the plot in Page 39 in the slides (or Fig. 3 in *Jones1993_Article_SimpleBoundaryCorrectionForKer.pdf*).

(Just a checking exercise, and I'm lazy to show how to code...)