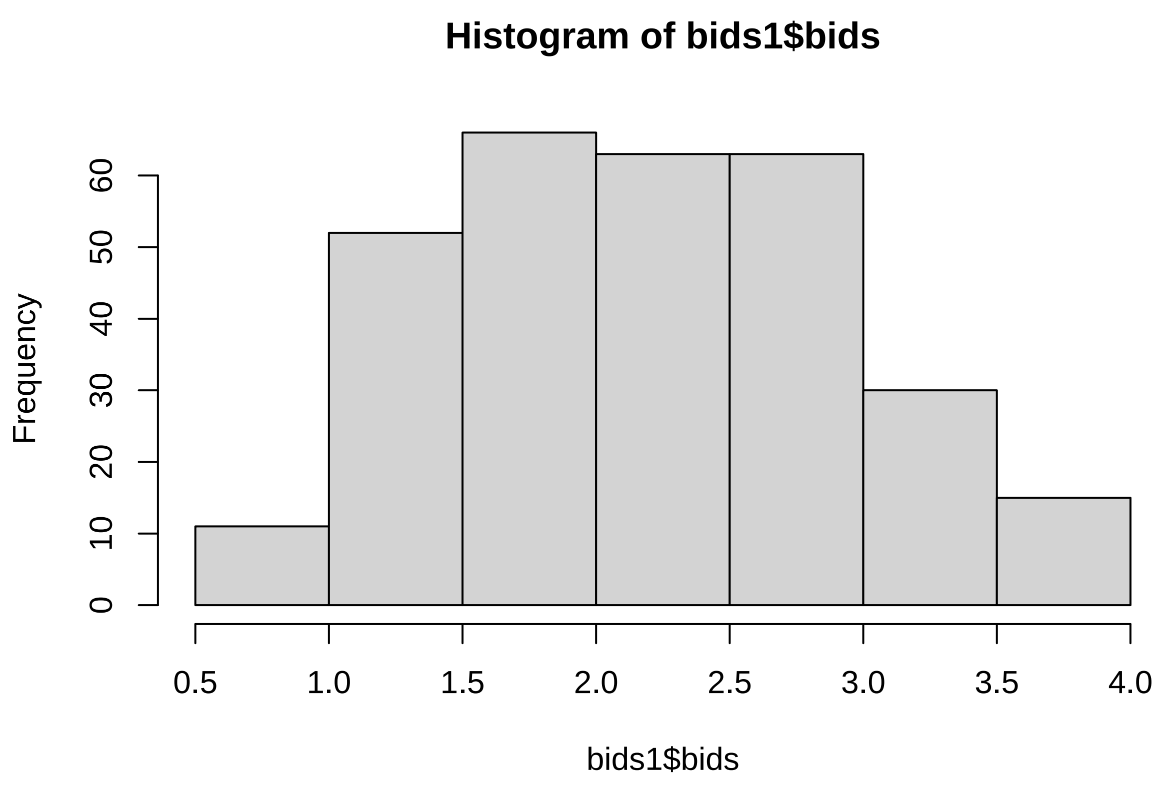
EC8855_PS1_Auction

Ying Wang

Load the data of simulated bids from a First Price Sealed Bid (FPSB) auction with three bidders.

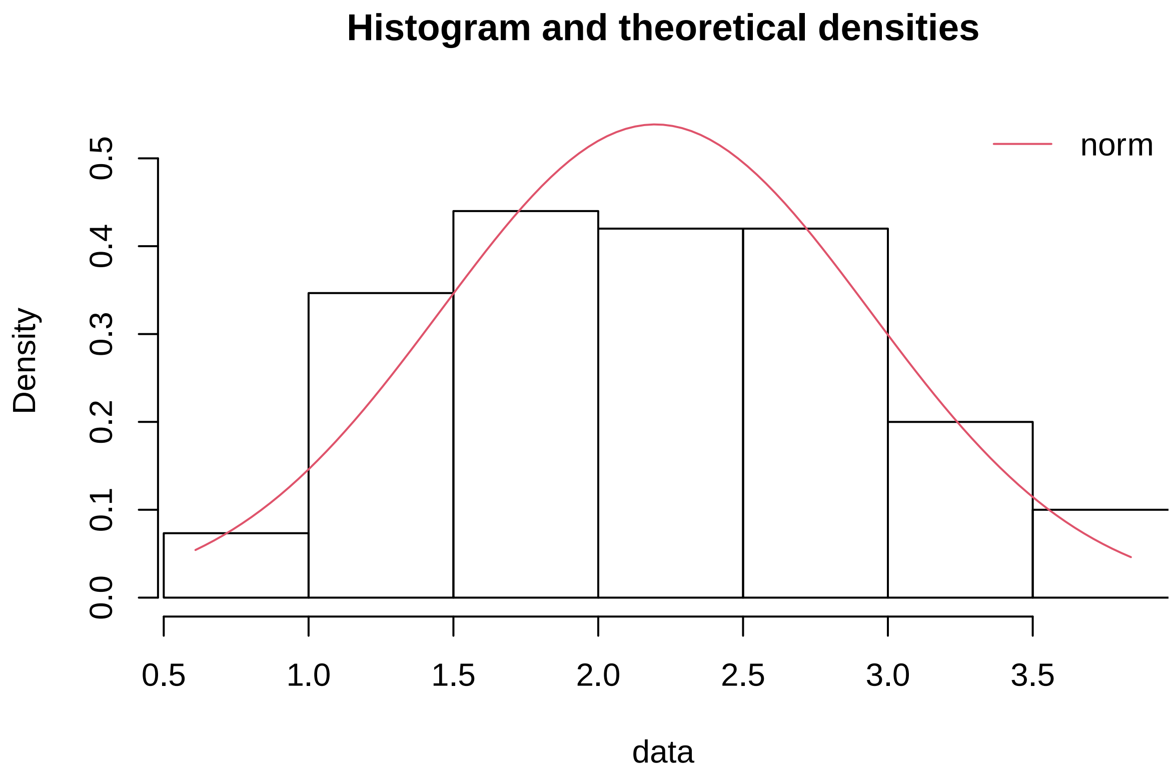
```
library(tidyverse)
library(fitdistrplus)
library(spatstat)

bids1 <- read_csv("bids1.csv", col_names = c("bids"))
hist(bids1$bids)
```



1. Estimate the density of bids.

a. Using an assumed normal distribution

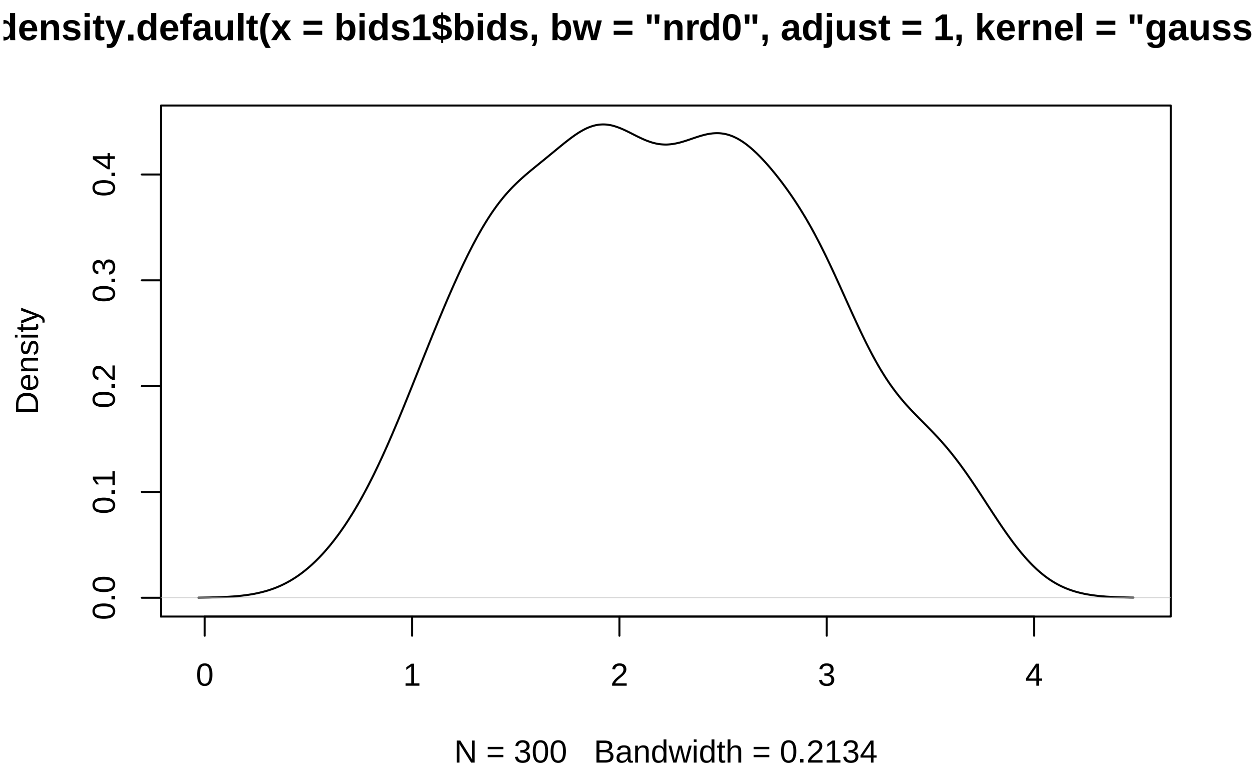
```
fhat.1a <- fitdist(bids1$bids, "norm")
denscomp(fhat.1a)
```



b. Using a Gaussian kernel

For a Gaussian kernel, the Silverman's plug-in estimate for the optimal bandwidth is $h_n^* = 0.9 * \min(s, IQ/1.34) * n^{-1/5}$, where we replace s is the sample standard deviation. The stats package will give this directly by via `bw.nrd0`.

```
fhat.1b <- density(bids1$bids, bw="nrd0", adjust = 1, kernel = "gaussian")
plot(fhat.1b)
```



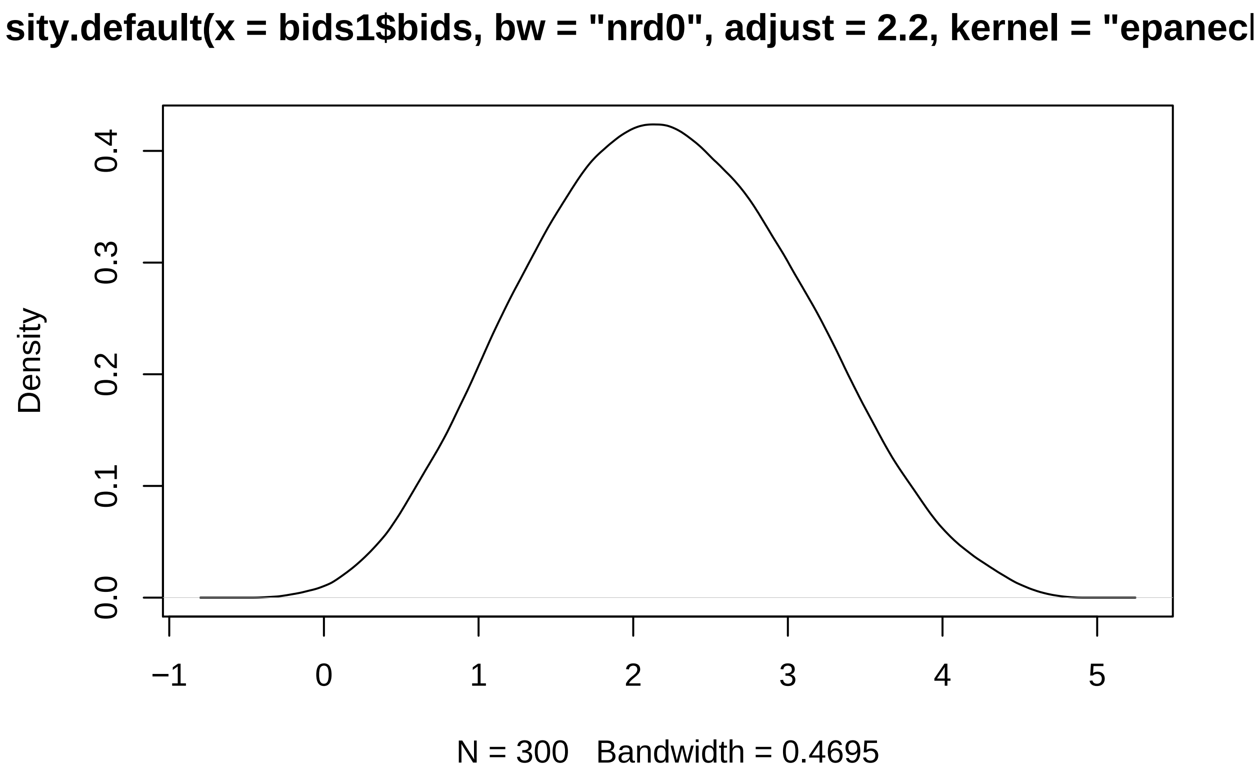
c. Using an Epanechnikov kernel

For a Gaussian kernel, the Silverman's plug-in estimate for the optimal bandwidth is $h_n^* = 0.9 * \min(s, IQ/1.34) * n^{-1/5}$.

For an Epanechnikov kernel, the Silverman's plug-in estimate for the optimal bandwidth is $h_n^* = 1.99 * \min(s, IQ/1.34) * n^{-1/5}$.

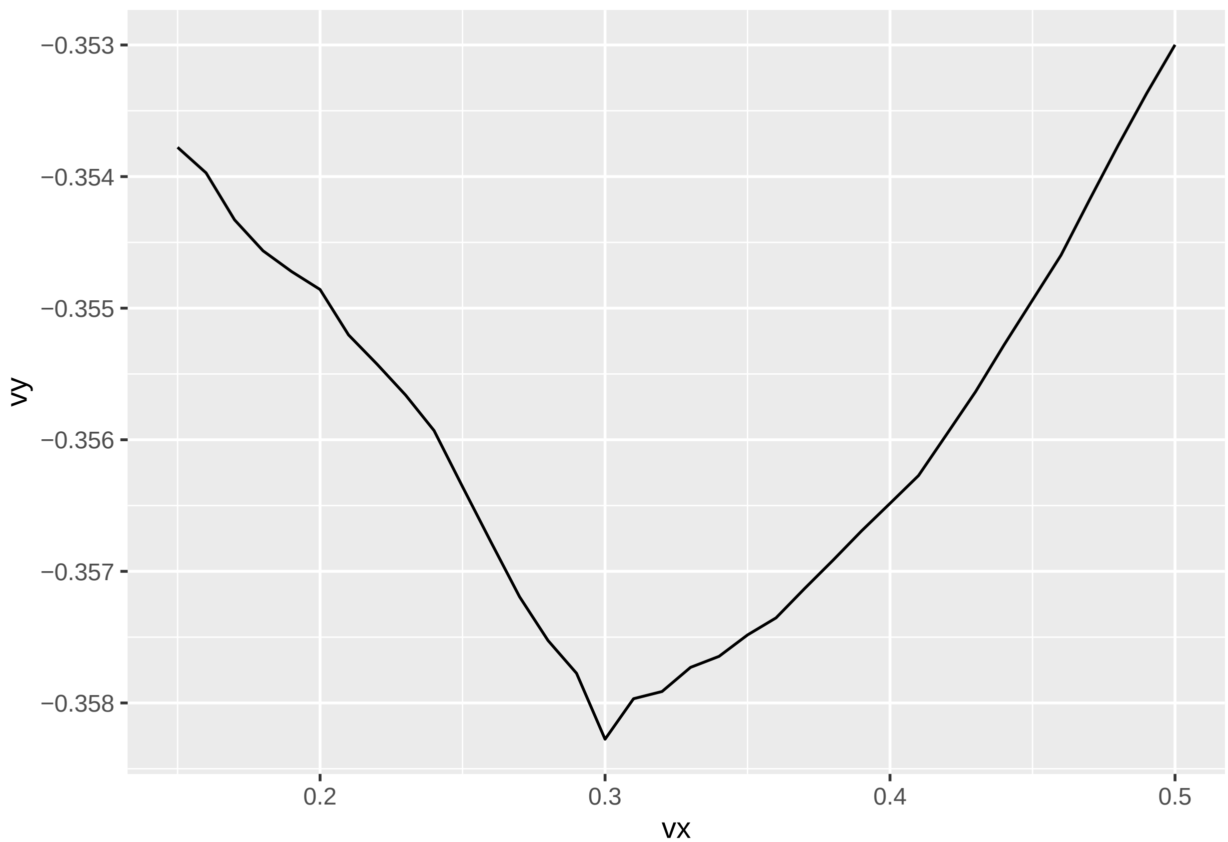
Therefore, the Silverman's plug-in estimate for the optimal bandwidth for a Gaussian kernel needs to be transformed to Epanechnikov kernel by setting $adjust = 1.99/0.9 = 2.2$.

```
fhat.1c <- density(bids1$bids, bw="nrd0", adjust = 2.2, kernel = "epanechnikov")
plot(fhat.1c)
```



2. Use a least-squares cross-validation to pick the bandwidth for the Epanechnikov kernel.

```
bids <- bids1$bids
X = bids
J <- function(h){
  fhat=Vectorize(function(x) density(X,from=x,to=x,n=1,bw=h,kernel = "epanechnikov")$y)
  fhati=Vectorize(function(i) density(X[-i],from=X[i],to=X[i],n=1,bw=h,kernel = "epanechnikov")$y)
  F=fhati(1:length(X))
  return(integrate(function(x) fhat(x)^2,-Inf,Inf)$value-2*mean(F))
}
vx=seq(.15,.5,by=.01)
vy=Vectorize(J)(vx)
df=data.frame(vx,vy)

qplot(vx,vy,geom="line",data=df)
```

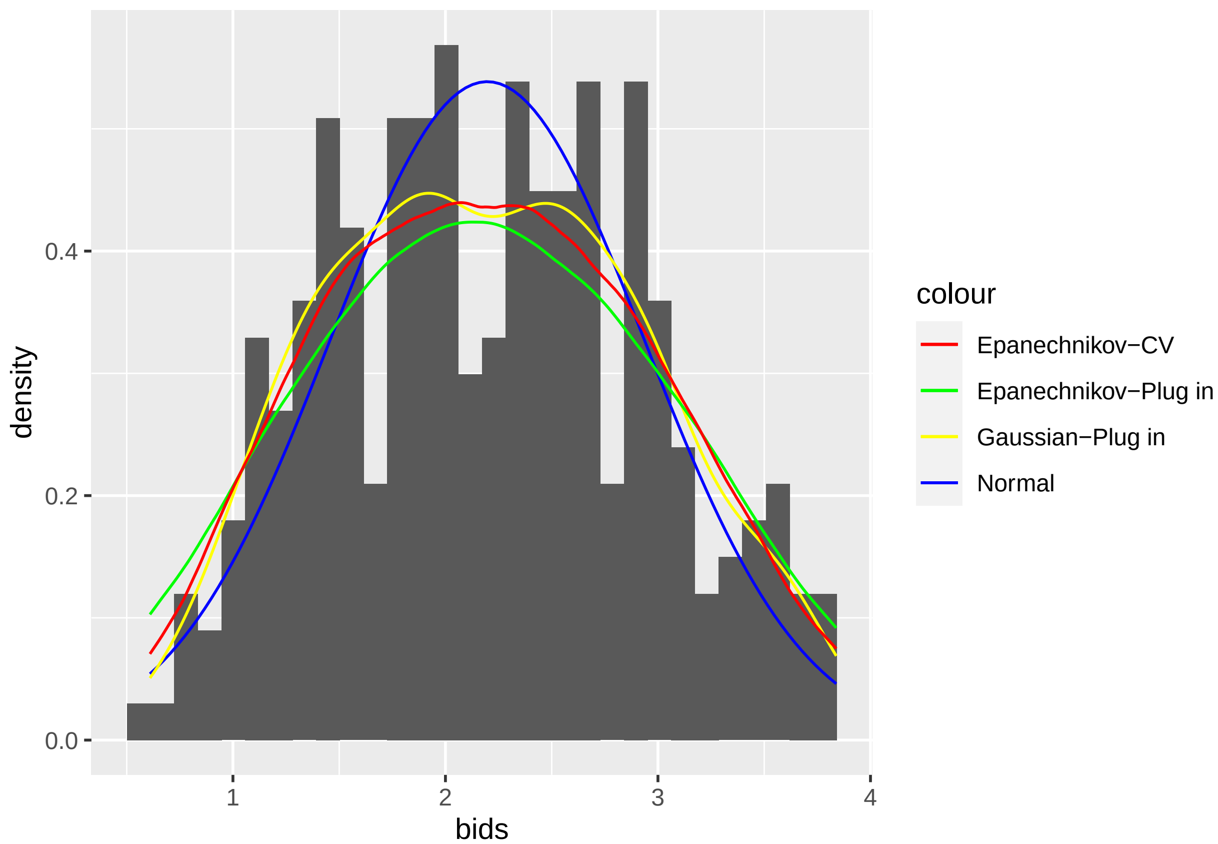



```
myopt<- optimize(J,interval=c(.1,.8))

bw_cv <- myopt$minimum

bw.ucv(bids)
```

```
## [1] 0.2701727
```

3. Compare four estimated density functions.

```
data <- as.data.frame(bids)
ggplot(data,aes(bids))  + geom_histogram(aes(y = stat(density))) +
  geom_line(stat = 'function', fun = dnorm, args = as.list(fhat.1a$estimate), aes(col = 'Normal')) +
  geom_line(stat = "density",bw = "nrd0", aes(col = 'Gaussian-Plug in')) +
  geom_line(stat = "density", bw = "nrd0", adjust = 2.2, kernel = "epanechnikov", aes(col = 'Epanechnik
  geom_line(stat = "density", bw = bw_cv, kernel = "epanechnikov", aes(col = 'Epanechnikov-CV')) +
  scale_color_manual(values = c('red', 'green', 'yellow','blue'))
```



The estimated density funtion using Gaussian kernel with Silverman's plug-in bandwidth appears to fit the data best.

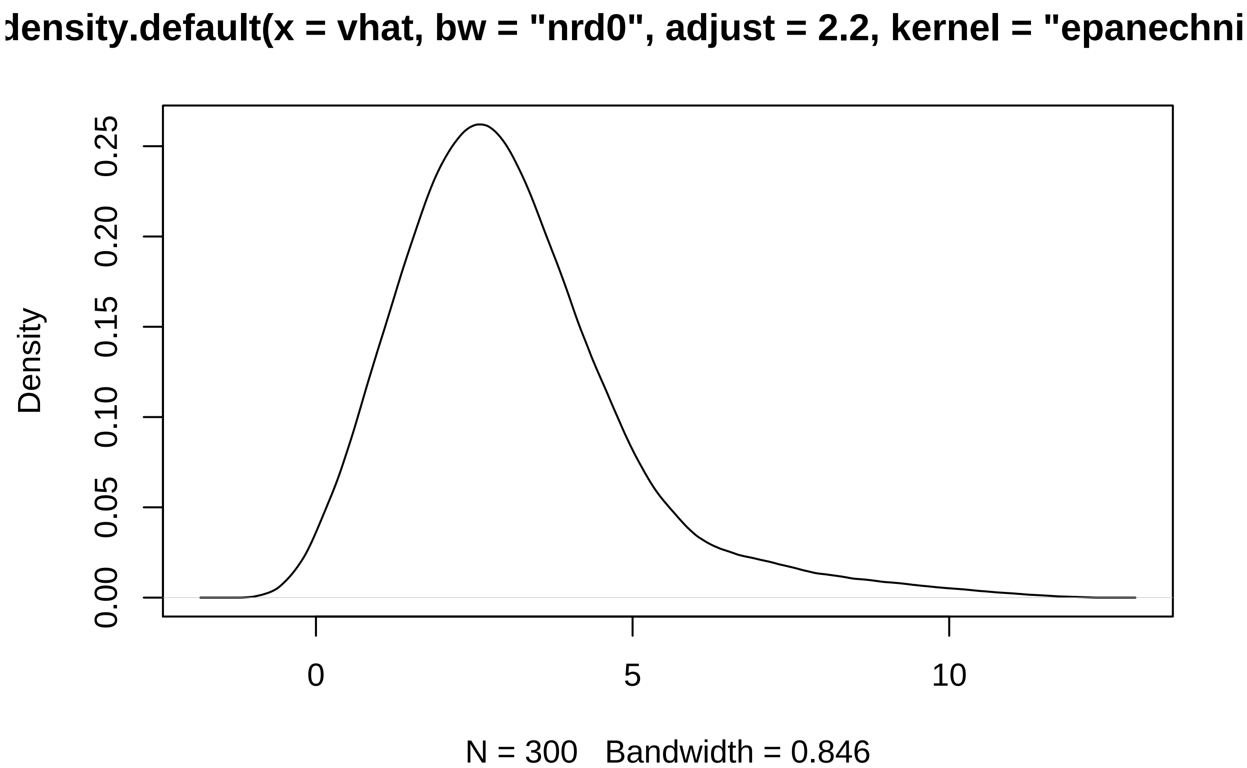
4. Use GPV and the cross-validated Epanechnikov kernel to recover the valuation implied for each bid.

$\hat{v} = b + \frac{\hat{G}_B(b)}{(n-1)\hat{g}_B(b)}$, where $n = 3$ is the number of bidders, $\hat{G}_B(b)$ is the estimated CDF of the bids, and $\hat{g}_B(b)$ is the estimated density of the bids.

```
n = 3
fhat.4 <- function(x) density(bids1$bids, from=x, to=x, n=1, bw=bw_cv, adjust = 1, kernel = "epanechnik
fhat.4.pdf <- density(bids1$bids, bw=bw_cv, adjust = 1, kernel = "epanechnikov")
Fhat.4.cdf <- CDF(fhat.4.pdf)
vhat <- bids + Fhat.4.cdf(bids) / ((n-1)*sapply(bids,fhat.4))
```

5. Estimate the distribution of v using another Epanechnikov kernel with plug-in bandwidth.

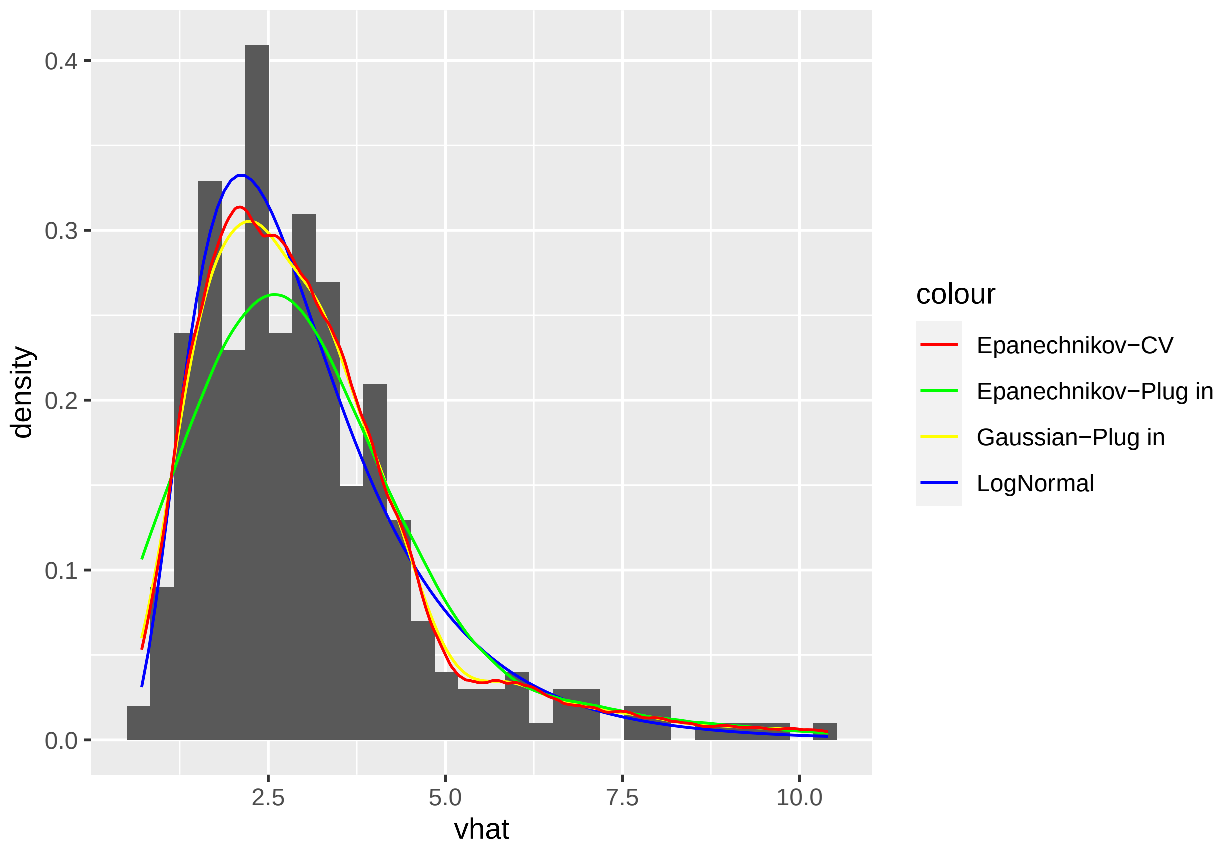
```
vhat_pdf <- density(vhat, bw="nrd0", adjust = 2.2, kernel = "epanechnikov")
plot(vhat_pdf)
```



6. Guess what distribution the valuations were generated with.

From question 5, the estimated density function of v is bell shaped and skewed to the right. It is reasonable to guess that it is generated with lognormal distribution. Let $\mu = 1, \sigma = 0.5$, check its fitness down below.

```
data.v <- as.data.frame(vhat)
ggplot(data.v,aes(vhat))  + geom_histogram(aes(y = stat(density))) +
  geom_line(stat = 'function', fun = dlnorm, args = list(meanlog = 1, sdlog = 0.5), aes(col = 'LogNormal
  geom_line(stat = "density", bw = "nrd0", kernel = "gaussian", aes(col = 'Gaussian-Plug in')) +
  geom_line(stat = "density", bw = "nrd0", adjust = 2.2, kernel = "epanechnikov", aes(col = 'Epanechnik
  geom_line(stat = "density", bw = bw_cv, kernel = "epanechnikov", aes(col = 'Epanechnikov-CV')) +
  scale_color_manual(values = c('red', 'green', 'yellow','blue'))
```


density
0.4
0.3
0.2
0.1
0.0
2.5
5.0
7.5
10.0
vhat
colour
Epanechnikov−CV
Epanechnikov−Plug in
Gaussian−Plug in
LogNormal