Yichen Dong HW 11

Yichen Dong

November 19, 2018

Problem 1A

Initial Setup

```
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

F12 = scan('F12.txt')
F12 = as.data.frame(F12)
F12 = F12 %>%
  mutate(X = log(F12))
n = length(F12$X)
sd = sd(F12$X)
F12 = F12%>%
  arrange(X)
```

I sorted by X so that I can plot the density using lines later.

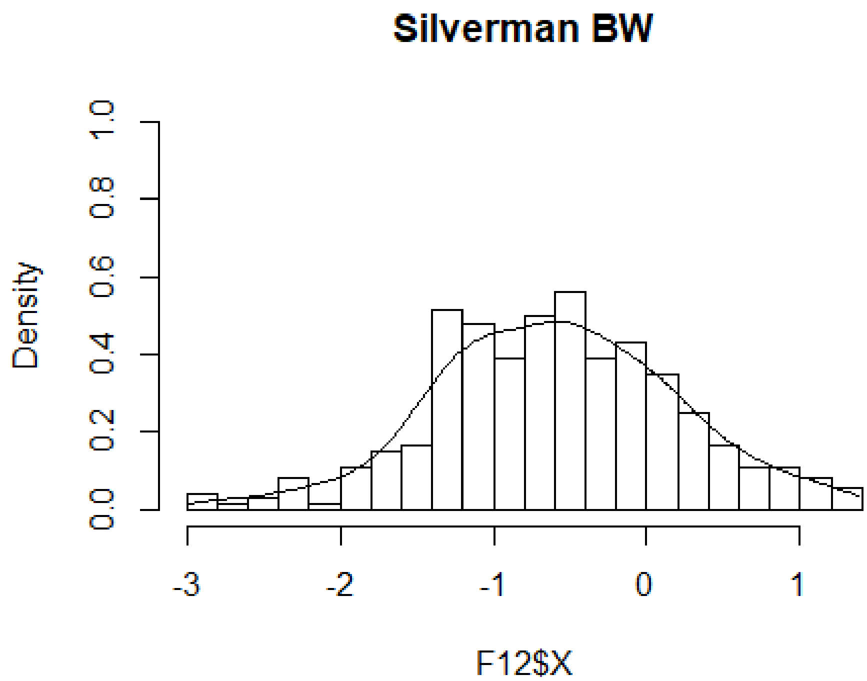
Silverman_bw

```
Silverman_bw = (4/(3*n))^(1/5)*sd
Silverman_density = vector(mode = 'double',length = n)
for(i in 1:length(F12$X)){
  for(j in 1:length(F12$X)){
    Silverman_density[i] = max(Silverman_density[i],0) + 1/(n*Silverman_bw)*(
exp(-(F12$X[i] - F12$X[j])^2/(2*Silverman_bw^2))/sqrt(2*pi))
  }
}
F12 = F12 %>%
  cbind(Silverman_density)

Silverman_bw

## [1] 0.2337528
```

```
hist(F12$X,freq = FALSE,ylim=c(0,1),breaks = 20,main='Silverman BW')
lines(F12$X,F12$Silverman_density)
```



SJ BW

Note: I tried to find the SJ BW by hand, but I got really stuck on the integration for R(f''). So I used the code in the book to generate a SJ BW so I could do the rest of the problem set.

Normal $K(z) = \exp(-z^2/2)/\sqrt{2\pi}$

x_i is constant

$$= \frac{e^{-\left(\frac{(x-x_i)^2}{2h_0^2}\right)}}{\sqrt{2\pi}} \to \frac{d}{dx}\,\frac{e^{-\left(\frac{x-x_i}{2h_0^2}\right)^2}}{\sqrt{2\pi}} = \frac{d}{dx}\, -\left(\frac{x-x_i}{\sqrt{2}h_0}\right)^2 \cdot \frac{e^{-\left(\frac{x-x_i^2}{2h_0^2}\right)}}{\sqrt{2\pi}}$$

$$= -\frac{x+x_i}{2h_0^2} \cdot \frac{e^{-\left(\frac{x-x_i}{2h_0}\right)}}{\sqrt{2\pi}} \to \frac{d}{dx}\, -\frac{e^{-\left(\frac{x-x_i}{2h_0}\right)}(x-x_i)}{h_0^2\sqrt{2\pi}}$$

$$= \frac{(x-x_i)^2 e^{-\frac{(x-x_i)^2}{2h_0^2}}}{\sqrt{2\pi}\cdot h_0^4} - \frac{e^{-\frac{(x-x_i)^2}{2h_0^2}}}{\sqrt{2\pi}\,h_0^2}$$

$$\hat{f}''(x) = \frac{1}{n h_0^3} \sum_{i=1}^{n}$$

(or is it just $\hat{f}''(x) = \frac{1}{n h_0^3} \sum_{i=1}^{n} \frac{e^{-\frac{(x-x_i)^2}{2h_0^2}}\left(\left(\frac{x-x_i}{h_0}\right)^2 - 1\right)}{\sqrt{2\pi}}$)

$$\hat{h} = \left(\frac{R(K)}{n\sigma_K^4 R(f'')}\right)^{1/5}$$

$\to h^5(n\sigma_K^4 R(f'')) = R(K)$

$R(f'') = \frac{R(K)}{h^5 n \sigma_K^4}$

R(K) for normal is $\frac{1}{2\sqrt{\pi}}$

$$R(f'') = \int (f''(z))^2\,dx$$

$$= \int\left(\frac{1}{4n h_0^5\sqrt{2\pi}} \cdot \sum_{i=1}^{n} \frac{(x-x_i)^2 e^{-\frac{(x-x_i)^2}{2h_0^2}}}{\sqrt{2\pi}\cdot h_0^4} - \frac{e^{-\frac{(x-x_i)^2}{2h_0^2}}}{\sqrt{2\pi}\,h_0^2}\right)^2 dx$$

```r
h_0 = Silverman_bw

R_f_d2 = 0
for (j in 1:length(F12$X)){
  for (i in 1:length(F12$X)){
    R_f_d2= -exp(-(F12$X[j] - F12$X[i])/h_0)/(32*n^2*h_0^9*pi)
  }
}
R_k = 1/(2*sqrt(pi))
sd = sd(F12$X)
SJ_bw = (R_k/(-n*sd^4*R_f_d2))^(1/5)
R_k/(h_0^5*n*sd^4) #This is what R_f_d2 should be in order for it to match th
e method they used in the book
```

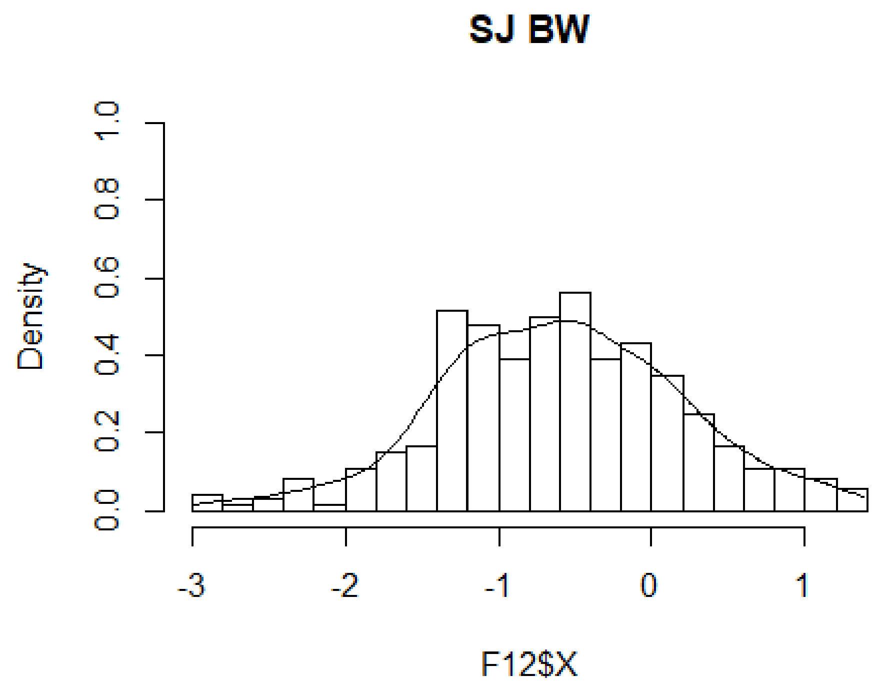
```
## [1] 1.686412
```

```r
#I couldn't figure out how to get R(f''(x)), so I used the method they had in
the book code
SJ_bw = bw.SJ(F12$X)
SJ_density = vector(mode = 'double',length = n)

for(i in 1:length(F12$X)){
  for(j in 1:length(F12$X)){
    SJ_density[i] = max(SJ_density[i],0) + 1/(n*SJ_bw)*(exp(-(F12$X[i] - F12$
X[j])^2/(2*SJ_bw^2))/sqrt(2*pi))
  }
}
F12 = F12 %>%
  cbind(SJ_density)
SJ_bw
```

```
## [1] 0.208654
```

```r
hist(F12$X,freq = FALSE,ylim=c(0,1),breaks = 20,main='SJ BW')
lines(F12$X,F12$SJ_density)
```

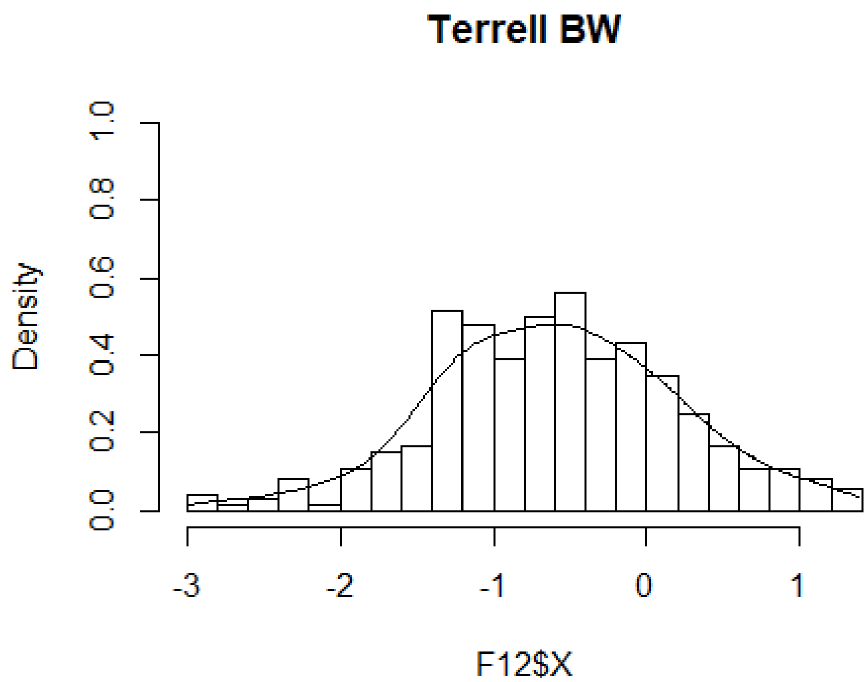


Terrell BW

```
Terrell_bw = 3*(R_k/(35*n))^(1/5)*sd
Terrell_density = vector(mode = 'double',length = n)
for(i in 1:length(F12$X)){
  for(j in 1:length(F12$X)){
    Terrell_density[i] = Terrell_density[i] + 1/(n*Terrell_bw)*(exp(-(F12$X[i
] - F12$X[j])^2/(2*Terrell_bw^2))/sqrt(2*pi))
  }
}
F12 = F12 %>%
  cbind(Terrell_density)
Terrell_bw

## [1] 0.2524386

hist(F12$X,freq = FALSE,ylim=c(0,1),breaks = 20,main='Terrell BW')
lines(F12$X,F12$Terrell_density)
```

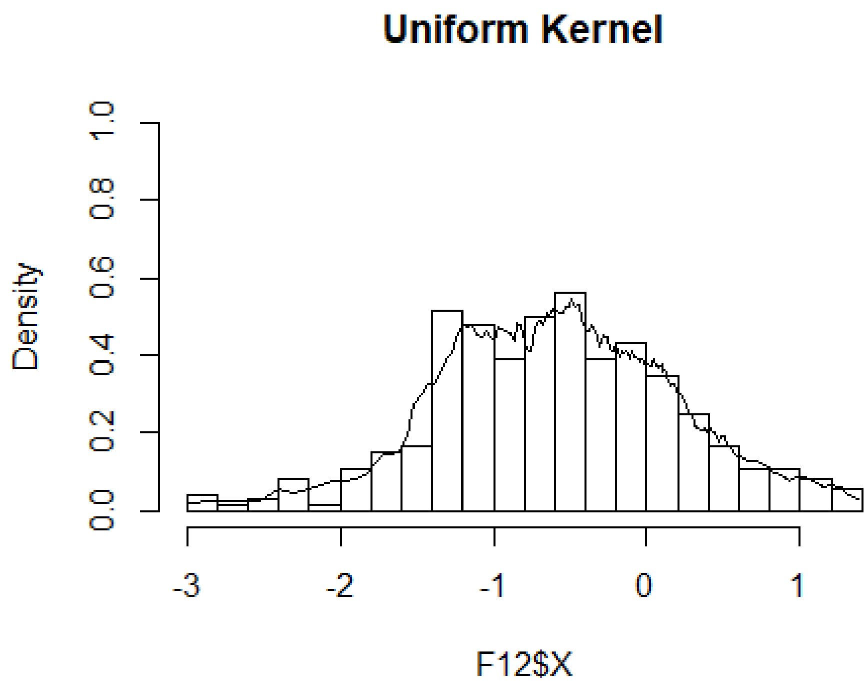


I feel like these are all so similar, which resulted in the density graph looking almost the same.

Part B

Uniform

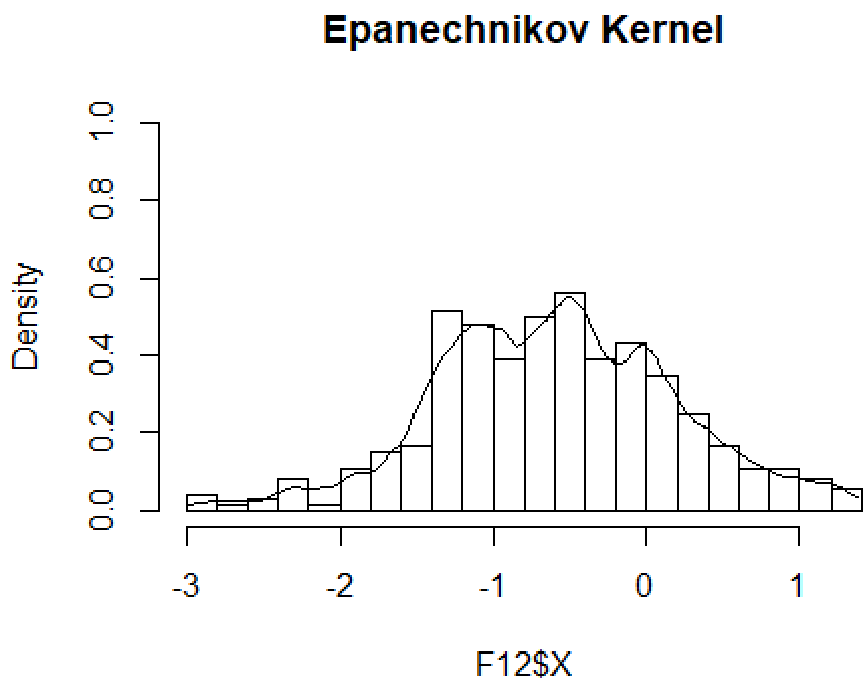
```
SJ_density_unif = vector(mode = 'double',length = n)
for(i in 1:length(F12$X)){
  for(j in 1:length(F12$X)){
    SJ_density_unif[i] = SJ_density_unif[i] + 1/(n*SJ_bw)*(1/2)*(ifelse(abs((
F12$X[i] - F12$X[j])/SJ_bw)<1,1,0))
  }
}
F12 = F12 %>%
  cbind(SJ_density_unif)
hist(F12$X,freq = FALSE,ylim=c(0,1),breaks = 20,main='Uniform Kernel')
lines(F12$X,F12$SJ_density_unif)
```



The uniform seems way too wiggly and seems to be overfitting the density on the underlying data.

Epanechnikov

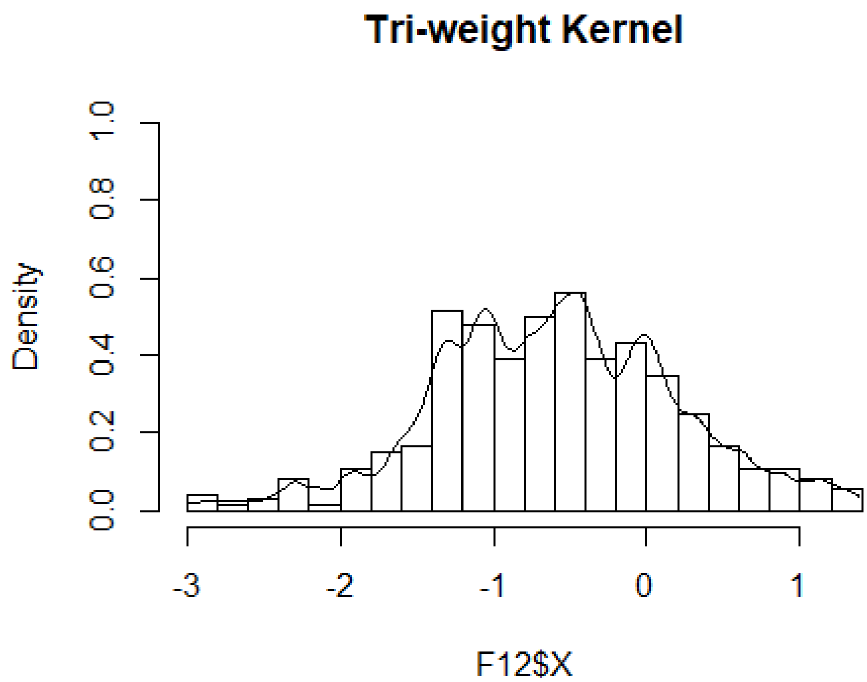
```
SJ_density_Ep = vector(mode = 'double',length = n)
for(i in 1:length(F12$X)){
  for(j in 1:length(F12$X)){
    SJ_density_Ep[i] = SJ_density_Ep[i] + 1/(n*SJ_bw)*(3/4*(1-((F12$X[i]-F12$
X[j])/SJ_bw)^2))*(ifelse(abs((F12$X[i] - F12$X[j])/SJ_bw)<1,1,0))
  }
}
F12 = F12 %>%
  cbind(SJ_density_Ep)
hist(F12$X,freq = FALSE,ylim=c(0,1),breaks = 20,main='Epanechnikov Kernel')
lines(F12$X,F12$SJ_density_Ep)
```



The density in this case seems smooth and doesn't wiggle too much, while still capturing some of the variations in the distribution.

Tri-weight

```
SJ_density_Tri = vector(mode = 'double',length = n)
for(i in 1:length(F12$X)){
  for(j in 1:length(F12$X)){
    SJ_density_Tri[i] = SJ_density_Tri[i] + 1/(n*SJ_bw)*(35/32*(1-((F12$X[i]-F12$X[j])/SJ_bw)^2)^3)*(ifelse(abs((F12$X[i] - F12$X[j])/SJ_bw)<1,1,0))
  }
}
F12 = F12 %>%
  cbind(SJ_density_Tri)
hist(F12$X,freq = FALSE,ylim=c(0,1),breaks = 20,main='Tri-weight Kernel')
lines(F12$X,F12$SJ_density_Tri)
```

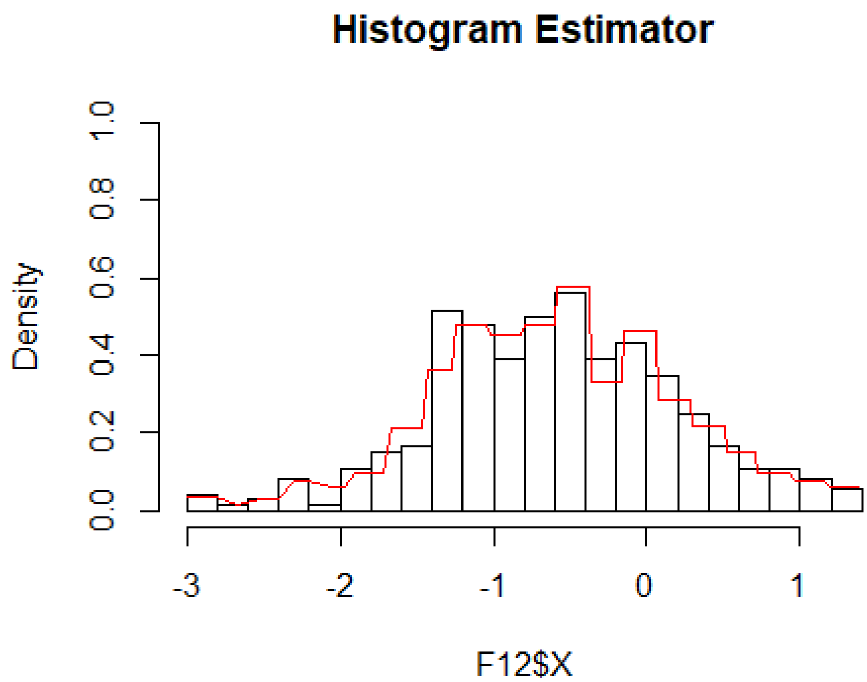


This looks similar to the Epanechnikov Kernel, except it tracks the movement of the distribution more closely. It might be slightly overfitting, or we could say the Epanechnikov is slightly underfitting.

Histogram

```
h_bins = seq(from = min(F12$X),to = max(F12$X),by=(max(F12$X)-min(F12$X))/20)
v_k = h_bins[2] - h_bins[1]
n_k = NULL
hist_estimator = vector(mode = 'double',length = n)
for (i in 1:(length(h_bins)-1)){
  n_k[i] = sum(ifelse((F12$X >= h_bins[i] & F12$X <= h_bins[i+1]),1,0))
}

for(i in 1:length(F12$X)){
  for(j in 1:length(n_k)){
    hist_estimator[i] = hist_estimator[i] + n_k[j]/(n*v_k)*ifelse((F12$X[i] >
= h_bins[j] & F12$X[i] <= h_bins[j+1]),1,0)
  }
}
F12 = F12 %>%
  cbind(hist_estimator)
hist(F12$X,freq = FALSE,ylim=c(0,1),breaks = 20,main='Histogram Estimator')
lines(F12$X,F12$hist_estimator,col='red')
```



Not too sure what I expected to be honest. This basically looks like a histogram, with slightly different densities.