Atlas-PS 11

David Atlas

11/7/2018

Problem 1

We begin by reading in the data.

```
data <- log(scan("./F12.txt"))
```

a)

Silverman's Rule of Thumb

We find h_0 via Silverman's Rule of Thumb, in which $h = \left(\frac{4}{3n}\right)^{\frac{1}{5}} \hat{\sigma}$, where $\hat{\sigma}$ is the sample standard deviation.

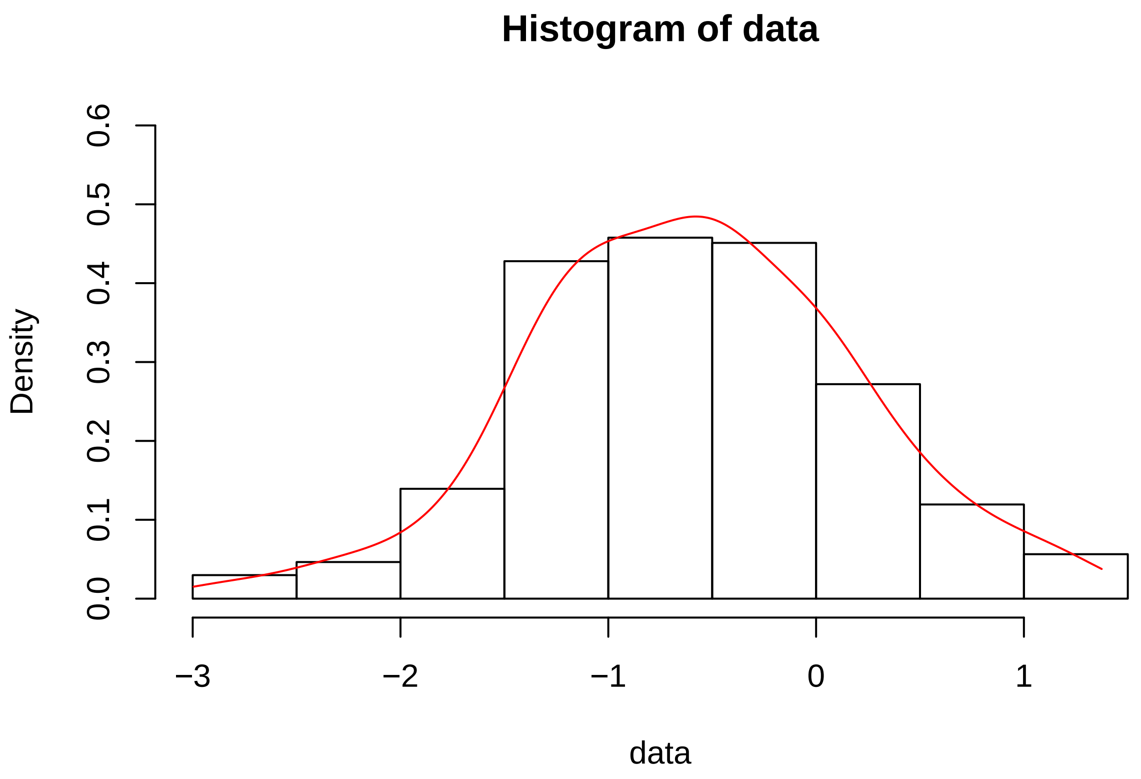
```
silvermans_rule_of_thumb <- function(data){
  h <- (4 / (3 * length(data)))^(1/5) * sd(data)
  return(h)
}

silvermans_rule_of_thumb(data)
```

```
## [1] 0.2337528
```

Silverman's Rule of Thumb yields $h_0 = .233$.

We plot the resulting density estimation over a histogram of the data.

```
get_kernel_density <- function(x, data, K, h){
  n <- length(data)
  return(sum(sapply(data, function(X_i){
    return(K((x - X_i) / h))
  })) / (n  * h))
}

X <- seq(min(data), max(data), .01)
f_hat <- sapply(X, function(x) get_kernel_density(x, data, dnorm, h=silvermans_rule_of_thumb(data)))
hist(data, freq=F, ylim=c(0, .6))
lines(X, f_hat, col='red')
```



The density estimate here is pretty good.

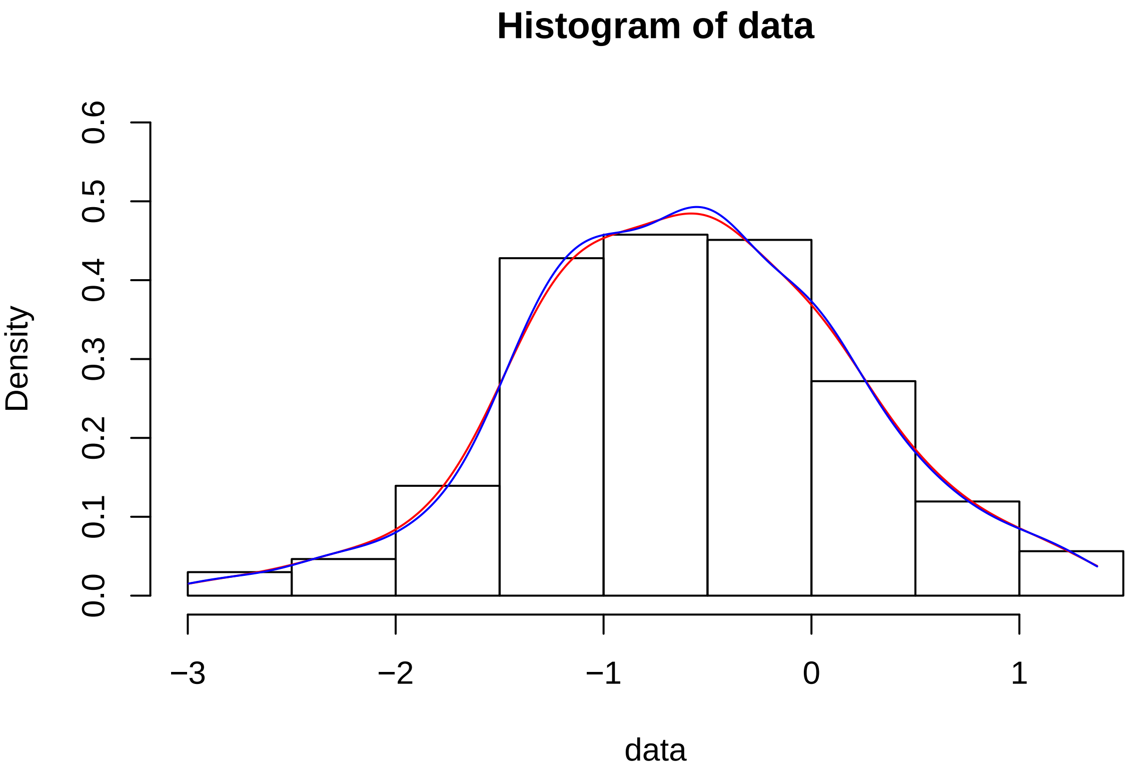
Sheather-Jones Method

Next, we find an estimate of h using the Sheather-Jones method. We take the $\hat{f}$ values that we found above, and take the ratio of f second differences to those of x to give an estimate of the second derivative. We find the roughness of that function.

```
derivative <- function(f_x, x){
  return(diff(f_x) / diff(x))
}

fprime <- derivative(f_hat, X)
f2prime <- derivative(fprime, X[2:length(X)])
R_k <- sum(dnorm(seq(min(data), max(data), .01))^2)
R_f2prime <- sum(f2prime^2)
n <- length(data)
h <- (R_k / (n * R_f2prime))^.2
```

Next, we find $\hat{f}$ using our value of $h = .202$.

```
f_hat_sj <- sapply(X, function(x) get_kernel_density(x, data, dnorm, h=.202))
hist(data, freq=F, ylim=c(0, .6))
lines(X, f_hat, col='red')
lines(X, f_hat_sj, col='blue')
```



Here, we see the data with the density estimate yielded by Silverman's Rule of Thumb in red, with the density estimated by the Sheather-Jones method in blue. There is minimal difference between the two, as h didn't vary much between the estimates.

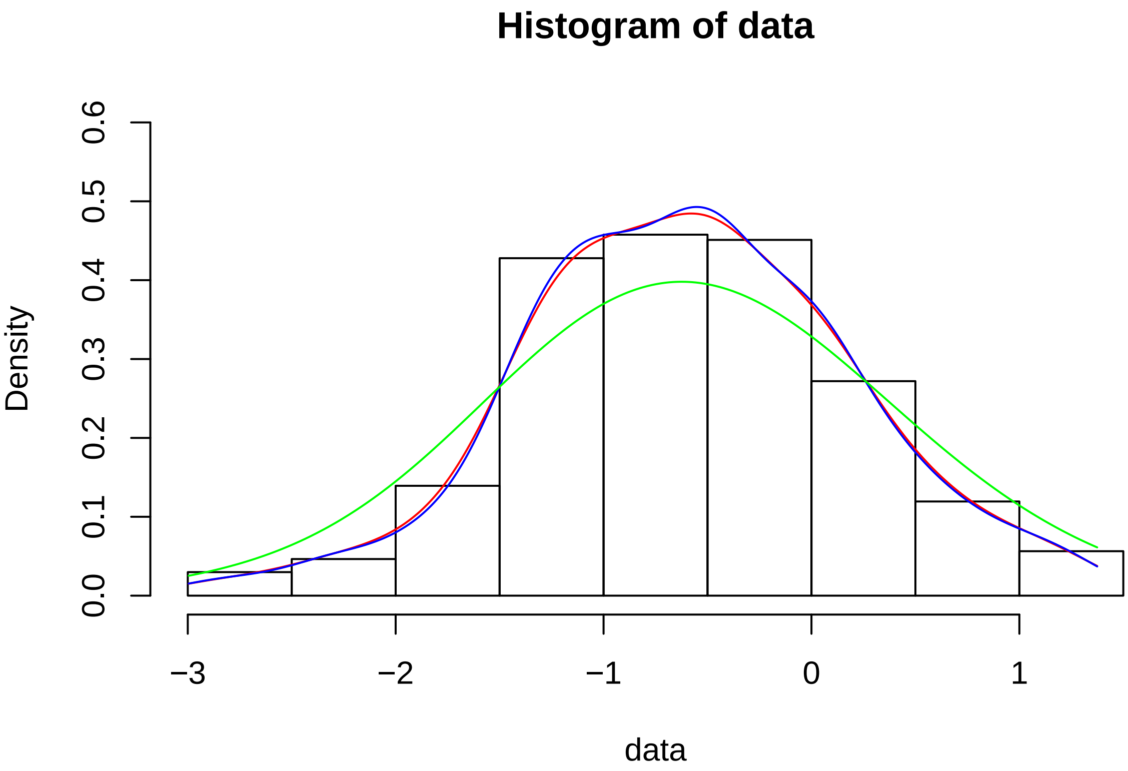
Terrell's Maximum Smoothing Principle

To find Terrell's Maximum Smoothing value of h, we use the formula:

$$h = 3\left(\frac{R(K)}{35n}\right)^{\frac{1}{5}}\hat{\sigma}.$$

We already found $R(K)$ for the normal density, so we simply plug it in to find $\hat{f}$.

```r
n <- length(data)
h <- 3 * ((R_k / (35 * n))^(1/5)) * sd(data)
f_hat_tmsp <- sapply(X, function(x) get_kernel_density(x, data, dnorm, h=h))
hist(data, freq=F, ylim=c(0, .6))
lines(X, f_hat, col='red')
lines(X, f_hat_sj, col='blue')
lines(X, f_hat_tmsp, col='green')
```



Here, we add in the density given via the maximum smoothing principle in green. It acts as expects; very smooth, but not as close of a fit. There is only one mode, and it is clearly the primary mode of the data.

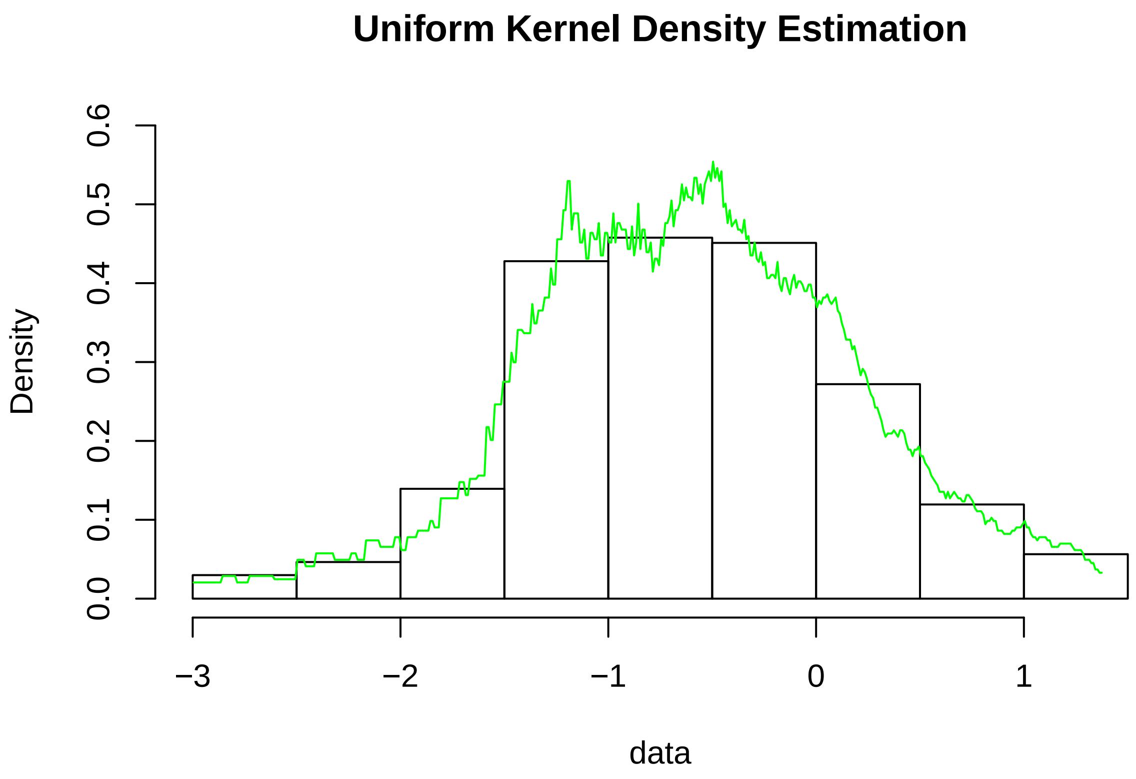
After fitting all 3, it looks like Sheather-Jones and Silverman's Rule of Thumb are basically equivalent. The maximum smoothing principle is a worse fit, but certainly does not overfit the data.

b)

Uniform Kernel

We find $\hat{f}$ using a uniform kernel:

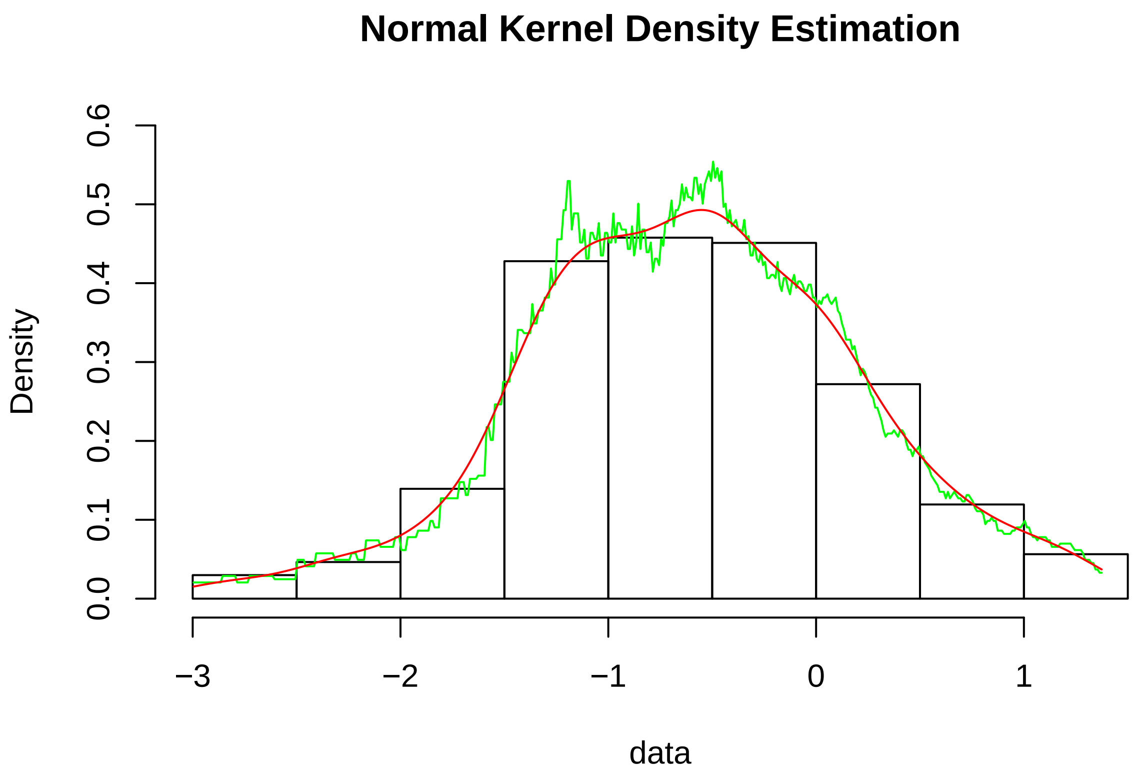
```r
h <- .202

# Uniform density with zero indictor.
k_uniform <- function(x){
  return(ifelse(abs(x) < 1, dunif(x, -1, 1), 0))
}

f_hat_unif <- sapply(X, function(x) get_kernel_density(x, data, k_uniform, h=h))
hist(data, freq=F, ylim=c(0, .6), main="Uniform Kernel Density Estimation")
lines(X, f_hat_unif, col="green")
```



Normal Kernel

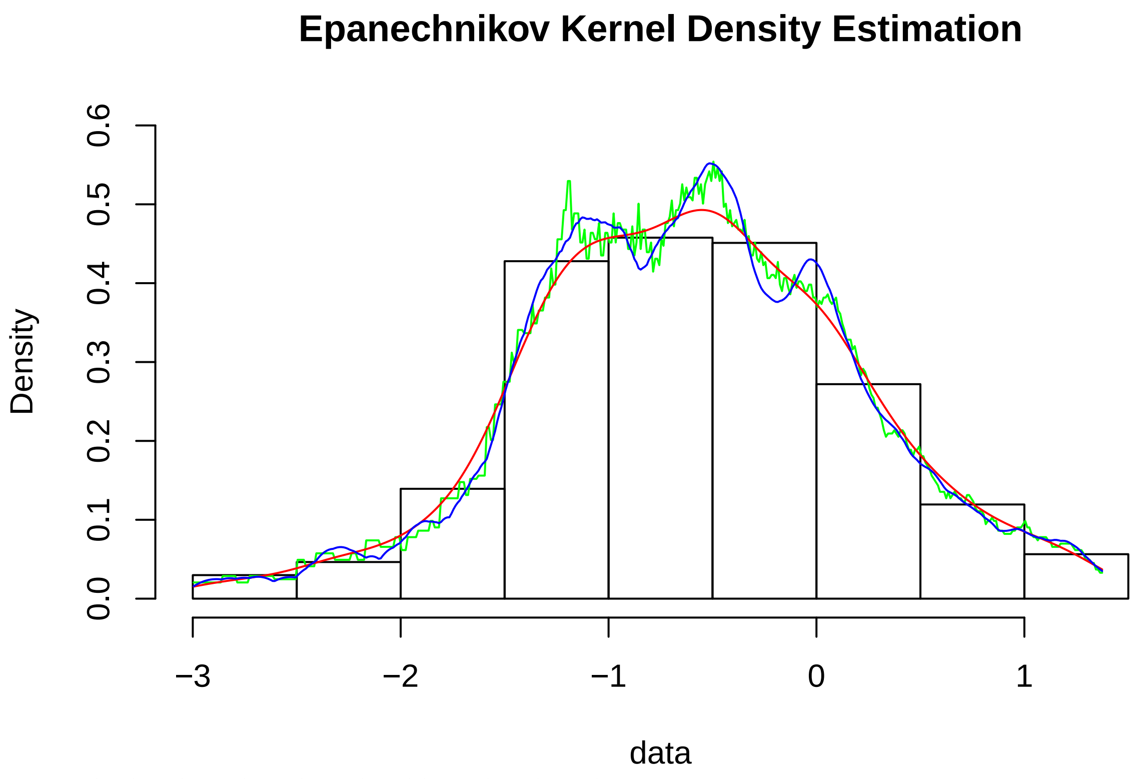
We already found this above, but we plot it in red against the data and the uniform kernel.

```
f_hat_norm <- sapply(X, function(x) get_kernel_density(x, data, dnorm, h=h))
hist(data, freq=F, ylim=c(0, .6), main="Normal Kernel Density Estimation")
lines(X, f_hat_unif, col="green")
lines(X, f_hat_norm, col="red")
```



Epanechnikov Kernel

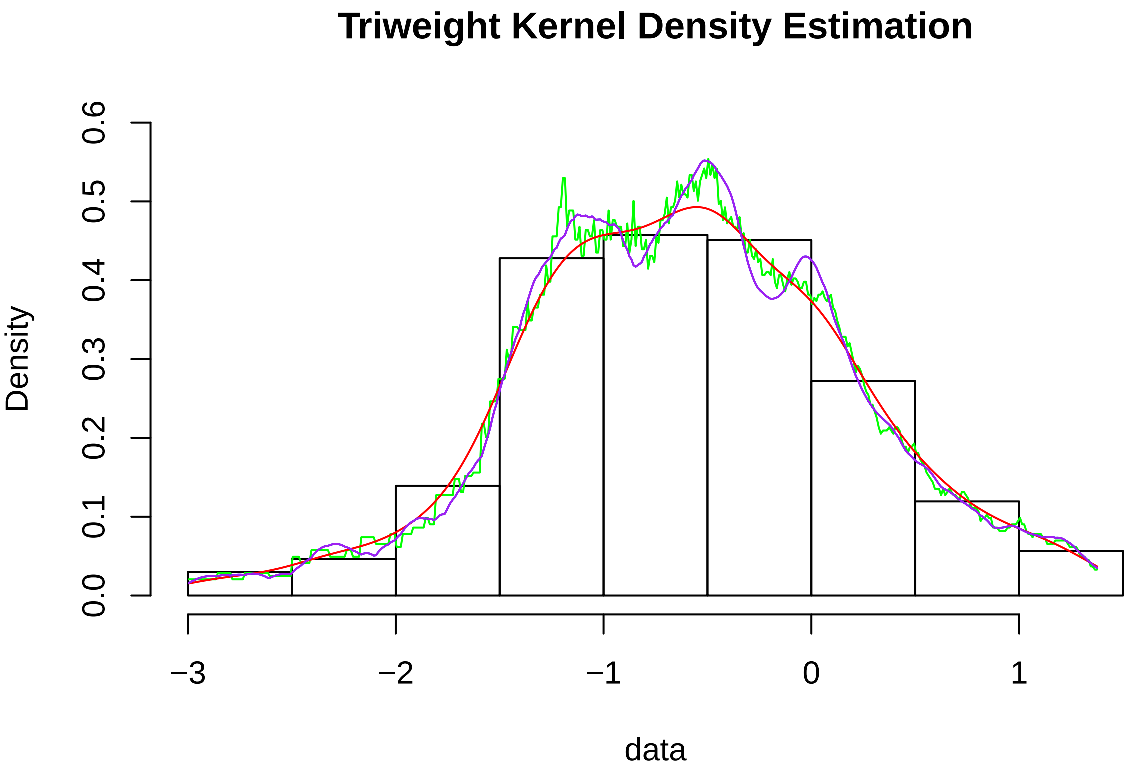
```
epach <- function(z) ifelse(abs(z) < 1, .75 * (1 - z^2), 0)
f_hat_epach <- sapply(X, function(x) get_kernel_density(x, data, epach, h=h))
hist(data, freq=F, ylim=c(0, .6), main="Epanechnikov Kernel Density Estimation")
lines(X, f_hat_unif, col="green")
lines(X, f_hat_norm, col="red")
lines(X, f_hat_epach, col='blue')
```



Here, the Epanechnikov kernel is plotted in blue, alot the other two.

Triweight Kernel

```
triweight <- function(z) ifelse(abs(z) < 1, (35/32) * (1 - z^2)^3, 0)
f_hat_triweight <- sapply(X, function(x) get_kernel_density(x, data, epach, h=h))
hist(data, freq=F, ylim=c(0, .6), main="Triweight Kernel Density Estimation")
lines(X, f_hat_unif, col="green")
lines(X, f_hat_norm, col="red")
lines(X, f_hat_epach, col='blue')
lines(X, f_hat_triweight, col='purple')
```

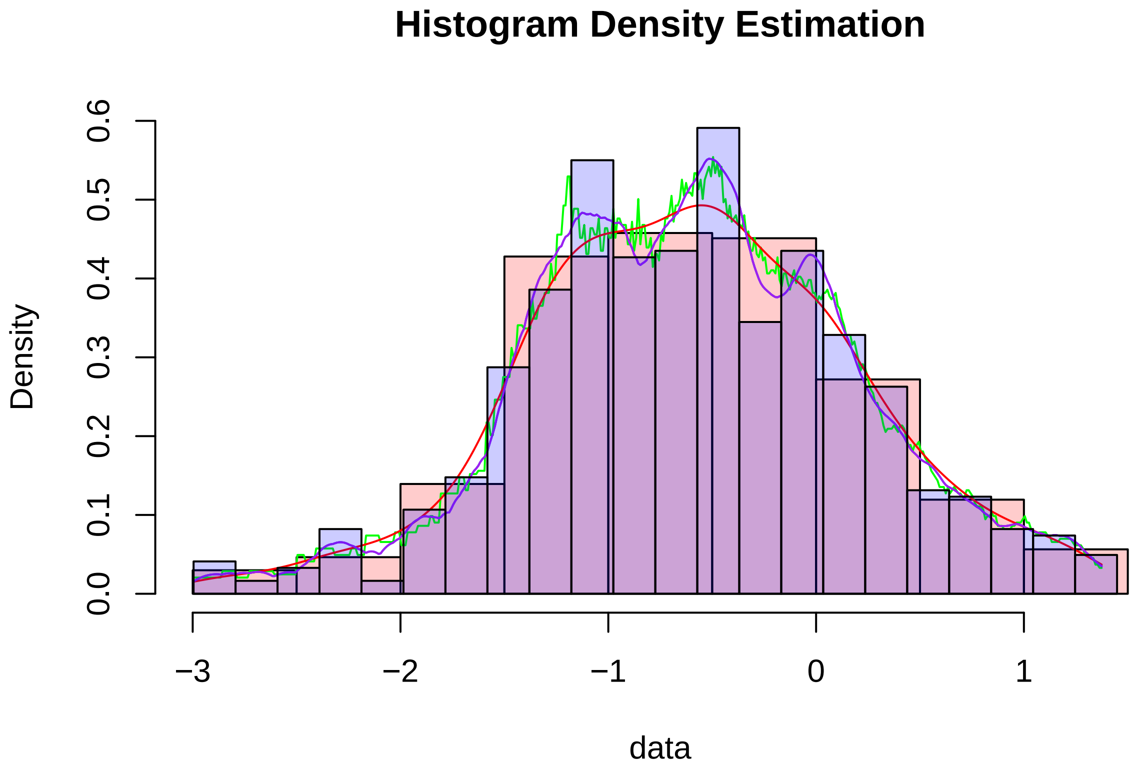


Using the triweight kernel, the data is fit tightly, with many of the same peaks, but smoother over each region.

Histogram Estimation

Next, we add in a histogram estimation using the Sheather-Jones Bandwidth found above:

```
bins <- seq(min(data), max(data) + h, h)
hist(data, freq=F, ylim=c(0, .6), main="Histogram Density Estimation", col=rgb(1, 0, 0 ,0.2))
lines(X, f_hat_unif, col="green")
lines(X, f_hat_norm, col="red")
lines(X, f_hat_epach, col='blue')
lines(X, f_hat_triweight, col='purple')
hist(data, breaks=bins, add=T, freq=F, col=rgb(0, 0, 1, 0.2))
```



Here are all the densities overlaid, where the original histogram is shown in red, and the histogram with Sheather-Jones bandwidth is shown in blue. It appears to perhaps overfit the data, showing several modes that may not be so significant.