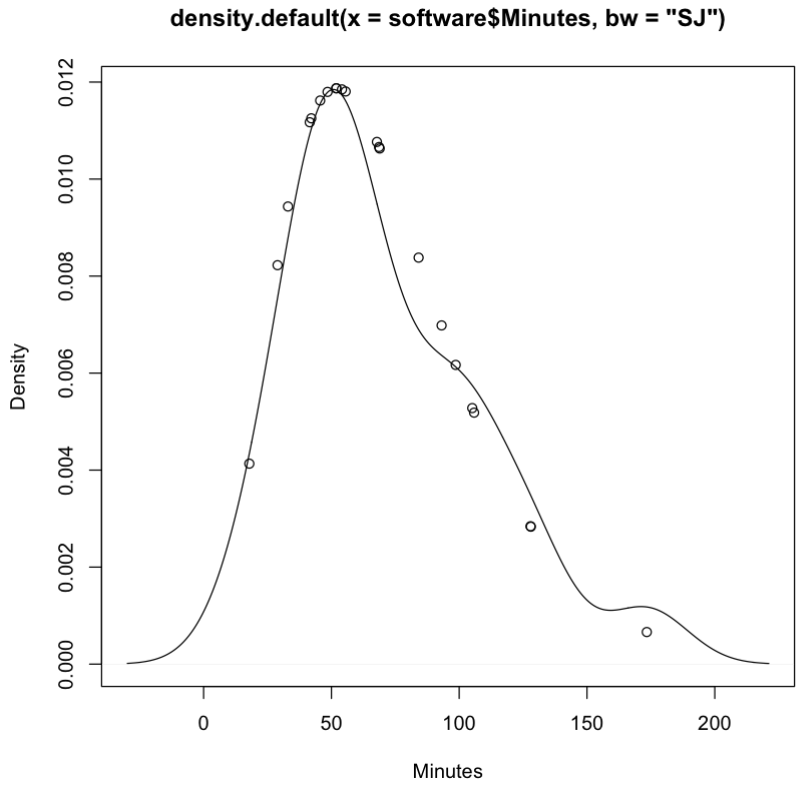
5.1 Consider the software data. Plot the kernel density estimate based on the smoothing parameter chosen by the Sheather-Jones method. Add a curve to the plot representing the density function of a gamma distribution, which can be calculated using the R function dgamma.

Recall that the gamma distribution has two parameters, corresponding to the arguments shape and scale to the function dgamma. The mean of the distribution is $(shape) \times (scale)$ and the variance of the distribution is $(shape) \times (scale)^2$; see the help file for rgamma for further details.Use these relationships, along with the sample mean and sample variance of the failure data, to choose the values of the arguments (that is, we are using the method of moments estimators of the parameters).

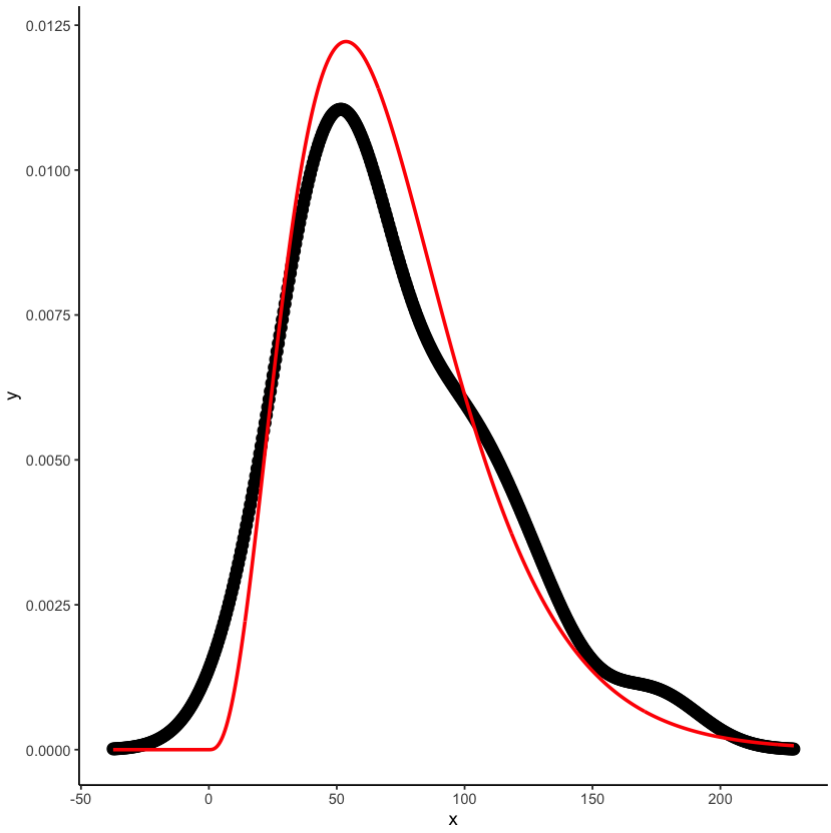
Based on this plot, does it appear that the failure data follow a gamma distribution?

The curve representing the density function would suggest that the data follows a gamma distribution

In [6]:
```
#data_loc<-'/Users/Alexis/Documents/Spring2020/nonparametrics/data/software.csv'
#trying again with the right dataset
data_loc<-'/Users/aporter1350/Documents/Courses/Spring2020/nonparametrics/data/failure.csv'
software<-read.csv(data_loc, header=FALSE)
colnames(software) <- c("Minutes")
#solve for a and s
variance=var(software$Minutes)
mu=mean(software$Minutes)
s=variance/mu
a=mu^2/variance
rate=1/s
gamma_curve=dgamma(software$Minutes, shape=a, scale=s)
plot(density(software$Minutes, bw="SJ"), xlab='Minutes')
points(software$Minutes, dgamma(software$Minutes, shape=a, scale=s))
```



In [7]:
```
library(ggplot2)
library(MASS)
den=density(software$Minutes)
df=data.frame(x=den$x, y=den$y)
fit.params <- fitdistr(software$Minutes, "gamma", lower = c(0, 0))
ggplot(data=df, aes(x=x, y=y))+
geom_point(size = 3) +
geom_line(aes(x=x, y=dgamma(x,fit.params$estimate["shape"], fit.params$e
stimate["rate"])), color="red", size = 1) +
theme_classic()
```

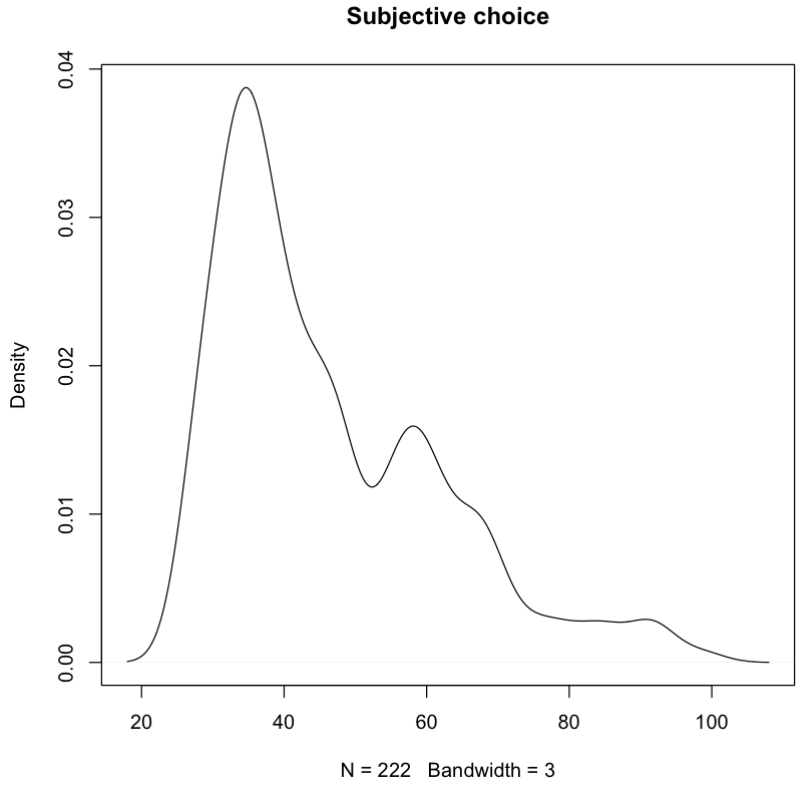


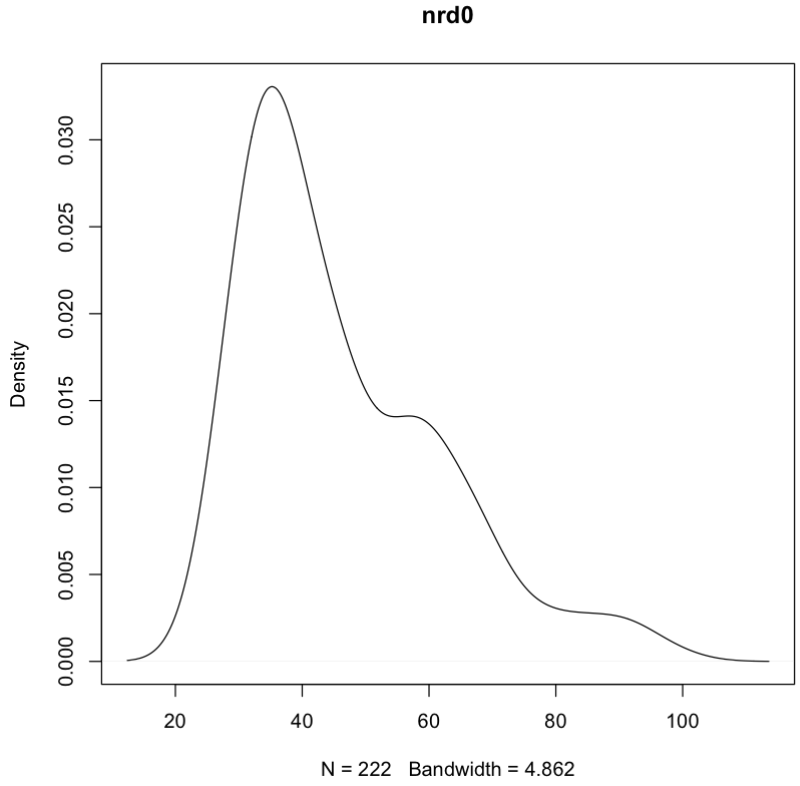
5.2 Consider the scores on the peabody data. Plot the kernel density estimates for these data based on the nrd0, Sheather-Jones, and cross validation methods, together with the estimate based on the value of h you chose in Exercise 4.6.

In your opinion do any of the three methods produce a density estimate that is preferable to the one you chose subjectively?

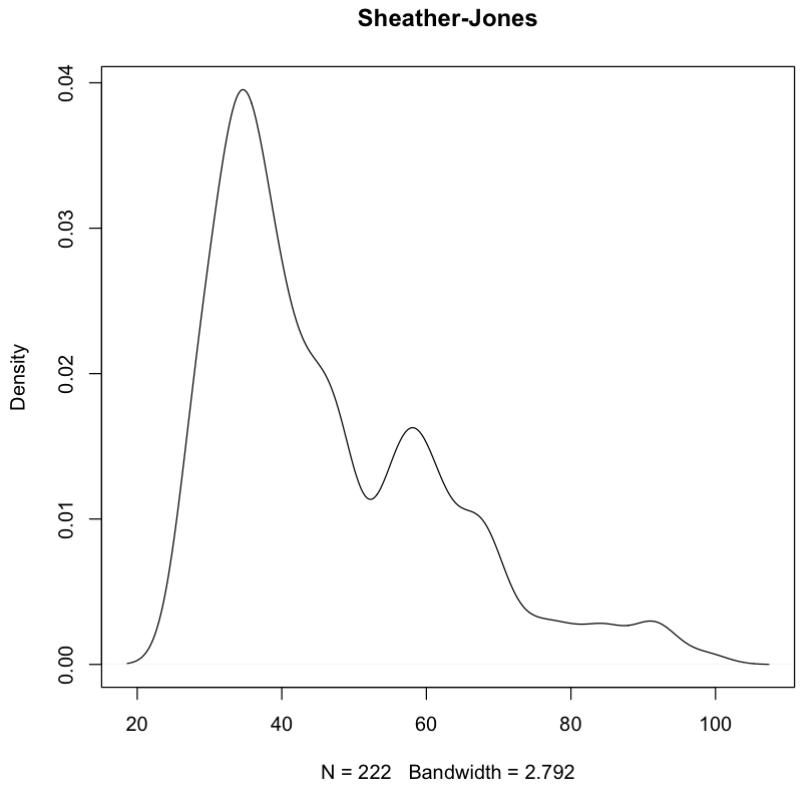
The Sheather-Jones produced a density estimate that is the most similar to te one I chose subjectively

In [43]:
```
#bw=3 change depending on answers
data_loc<-'/Users/Alexis/Documents/Spring2020/nonparametrics/data/peabody.csv'
#data_loc<-'/Users/aporter1350/Documents/Courses/Spring2020/nonparametrics/data/peabody.csv'
pea<-read.csv(data_loc)
plot(density(pea$peabody, bw=3), main='Subjective choice')
plot(density(pea$peabody, bw='nrd0'), main='nrd0')
plot(density(pea$peabody, bw="SJ"), main='Sheather-Jones')
plot(density(pea$peabody, bw='ucv'), main='Unbiased cross-validation')
```

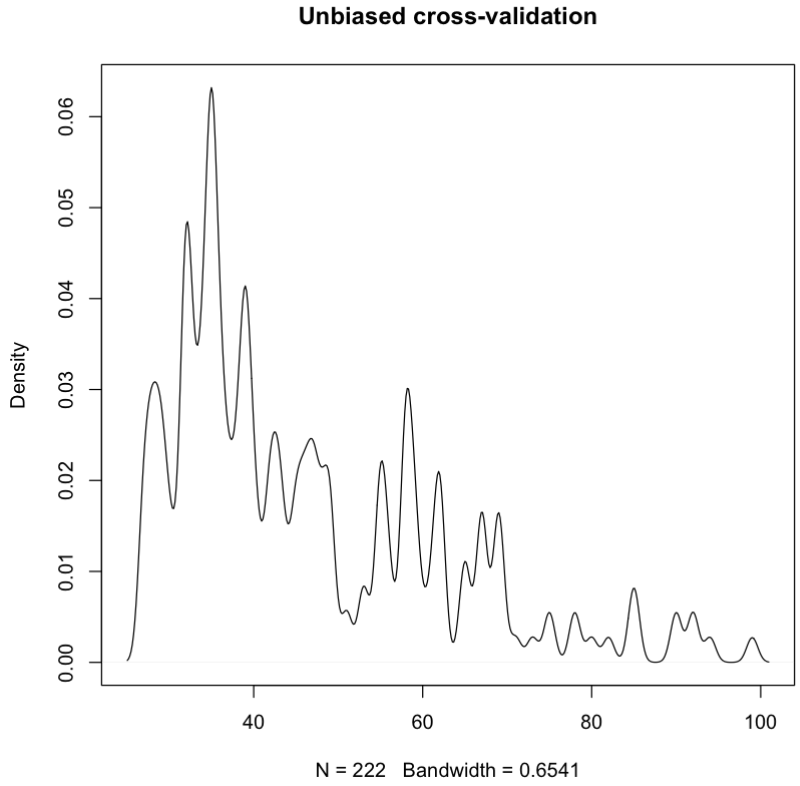





```
Warning message in bw.ucv(x):
“minimum occurred at one end of the range”
```



4.8618269137599



5.3 The dataset geyser contains data on the time between eruptions of the old faithful geyser at Yellowstone. Over the period August 1 - Augest 15 1985. Estimate the density of the time between eruptions using a kernel estimate. Consider four methods of choosing the smoothing parameter, nrd, nrd0, SJ, and ucv. Give the value of the smoothing parameter found by each method and plot the density estimate that you consider to be the best choice (from among the four possibilities). Comment on the general feature of the density.

nrd h=4.71

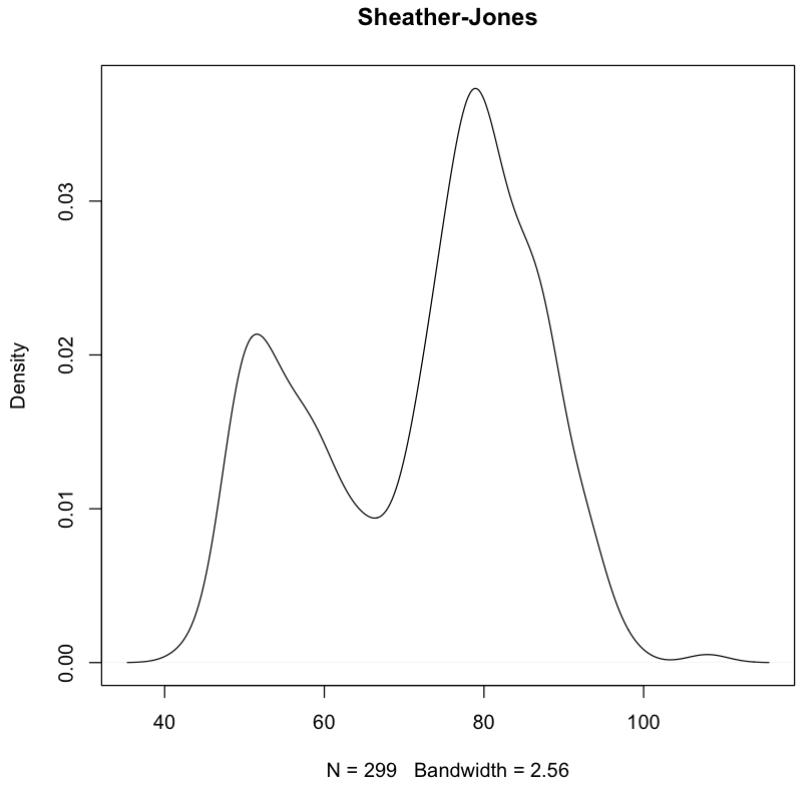
nrd0 h=4.0

SJ h=2.56

ucv h=2.2

Based on the results of h and the reading I would suggest the Sheather-Jones estimate to be the best choice as it generally fits the data without over smoothing it.

In [48]:
```
#using the column waiting
data_loc<-'/Users/Alexis/Documents/Spring2020/nonparametrics/data/geyser.csv'
#data_loc<-'/Users/aporter1350/Documents/Courses/Spring2020/nonparametrics/data/peabody.csv'
geyser<-read.csv(data_loc)
#plot(density(geyser$waiting, bw='nrd'), main='nrd')
#plot(density(geyser$waiting, bw='nrd0'), main='nrd0')
plot(density(geyser$waiting, bw="SJ"), main='Sheather-Jones')
#plot(density(geyser$waiting, bw='ucv'), main='Unbiased cross-validation')
```



5.4 Consider the peabody data

(a) Let $\hat{p}(.)$ denote the kernel density estimate with the smoothing parameter taken to be h=2. Use numerical integration to find $\hat{\mu}$ and $\hat{\sigma}$. When calculating the integral numerically use the method described in Section 5.3 to find the limits of integration

μ=46.46

$\hat{\sigma}^2$=5291.01

$\hat{\sigma}$=72.74

(b) Repeat part (a) for h=3,4, and 5

h=3

μ=46.46

$\hat{\sigma}^2$=5669.116

$\hat{\sigma}$=75.29

h=4

μ=46.46

$\hat{\sigma}^2$=6047.18

$\hat{\sigma}$=77.76

h=5

μ=46.46

$\hat{\sigma}^2$=6425.25

$\hat{\sigma}$=80.16

(c) Summarize how the values of $\hat{\mu}$ and $\hat{\sigma}$ change as h increases

The mean stays the same as h increases. The values of $\hat{\sigma}$ increases as h increases. The variance increases when you increase h which is in line with our last chapter discussing the bias variance trade off.

In [62]:
```
denout=density(pea$peabody, bw=2)
#determine upper and lower bounds by looking at min and max
summary(denout$x)
f_mu=function(y){y*approxfun(denout)(y)}
integrate(f=f_mu, lower=21, upper=105)

f_sig=function(y){y-(y*approxfun(denout)(y))^2*approxfun(denout)(y)}
integrate(f=f_sig, lower=21, upper=105)
sqrt(5291.01)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
     21      42      63      63      84     105 
```

```
46.45819 with absolute error < 0.00065
```

```
5291.01 with absolute error < 0.47
```

72.7393291143106

In [ ]:
```
#repeat with h=3
denout=density(pea$peabody, bw=3)
#determine upper and lower bounds by looking at min and max
#summary(denout$x)
f_mu=function(y){y*approxfun(denout)(y)}
integrate(f=f_mu, lower=18, upper=108)

f_sig=function(y){y-(y*approxfun(denout)(y))^2*approxfun(denout)(y)}
integrate(f=f_sig, lower=18, upper=108)
#sqrt(5669.116)
```

In [69]:
```
#repeat with h=4
denout=density(pea$peabody, bw=4)
#determine upper and lower bounds by looking at min and max
summary(denout$x)
f_mu=function(y){y*approxfun(denout)(y)}
integrate(f=f_mu, lower=15, upper=111)

f_sig=function(y){y-(y*approxfun(denout)(y))^2*approxfun(denout)(y)}
integrate(f=f_sig, lower=15, upper=111)
sqrt(6047.182)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
     15      39      63      63      87     111 
```

```
46.45845 with absolute error < 0.0046
```

```
6047.182 with absolute error < 0.058
```

77.7636290305436

In [71]:
```
#repeat with h=5
denout=density(pea$peabody, bw=5)
#determine upper and lower bounds by looking at min and max
summary(denout$x)
f_mu=function(y){y*approxfun(denout)(y)}
integrate(f=f_mu, lower=12, upper=114)

f_sig=function(y){y-(y*approxfun(denout)(y))^2*approxfun(denout)(y)}
integrate(f=f_sig, lower=12, upper=114)
sqrt(6425.249)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   12.0    37.5    63.0    63.0    88.5   114.0 
```

```
46.45822 with absolute error < 0.0021
```

```
6425.249 with absolute error < 0.64
```

80.1576509136838

5.5 For the failure data use fan.test to test the hypothesis that the data follow a gamma distribution, choosing the parameters of the distribution using the same approach used in 5.1. Choose the smoothing parameter using the Sheather-Jones method and compare the densities over the interval (15,185). Based on the results would the assumption of a gamma distribution for the failure data seem reasonable?

The results follow the null hypothesis is not reject so the estimated density of the data is consisten with the gamma distribution.

In [9]:
```
library(GoFKernel)
variance=var(software$Minutes)
mu=mean(software$Minutes)
s=variance/mu
a=mu^2/variance
fan.test(software$Minutes, fun.den=dgamma, par=list(shape=a, scale=s),
         bw=bw.SJ(software$Minutes), lower=15, upper=185)
```

```
Loading required package: KernSmooth
```

```
KernSmooth 2.23 loaded
Copyright M. P. Wand 1997-2009
```

```

        Fan's test

data:  software$Minutes
Ig = -1.5057, p-value = 0.9339
```

5.6 For the data in peabody use fan.test to test the hypothesis that the data are normally distributed using the sample mean and sample standard deviation for the parameter values. Compare the densities over a range that is slightly larger than the range of the data and use the Sheather-Jones method to chose the smoothing paramter. What do you conclude about the assumption that the test scores are normally distributed?

Based on these results the data do not follow a normal distribution as we reject the null.

In [100]:
```
fan.test(pea$peabody, fun.den=dnorm, par=list(mean=46.41, sd=15.92),
         bw=bw.SJ(pea$peabody), lower=18, upper=108)
```

```
        Fan's test

data:  pea$peabody
Ig = 14.101, p-value < 2.2e-16
```

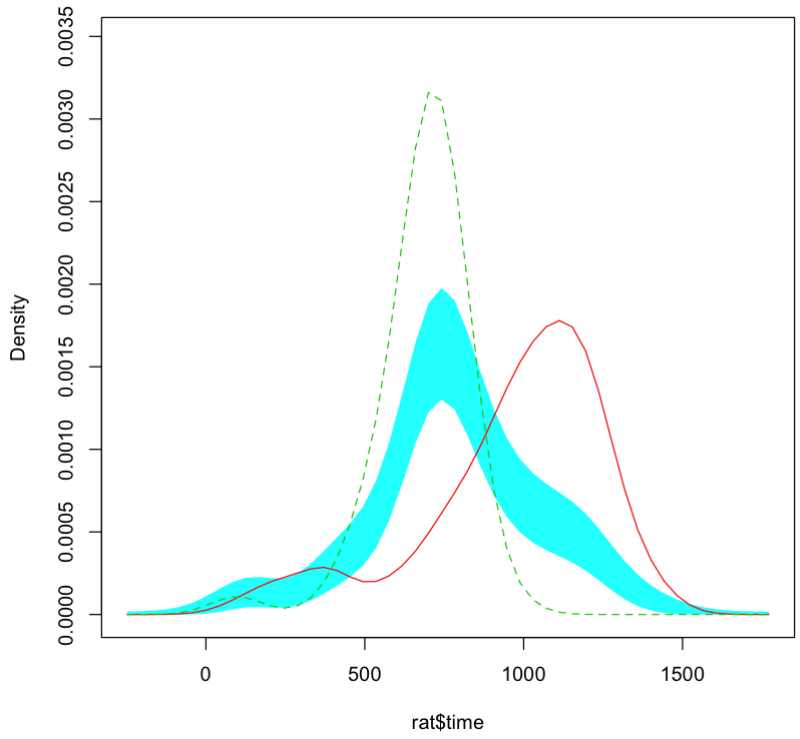
5.7 The dataset rats contains the lifetime (in days) of 195 rats, each which received either an unrestricted diet (diet=1) or a restricted diet (diet=0). The goal of this exercise is to determine if there is evidence that the distribution of lifetimes is different for the two diets Following the procedure described in example 5.8 compute the p-value for the test of the hypothesis that the density function of lifetime is the same for the two groups defined by the type of diet. Choose the smoothing parameters using the Sheather-Jones method. Based on this results what do you conclude regarding the hypothesis?

Using the standard .05 criterion for significance, the p-value of 0 indicates we would reject the hypothesis the density function for the two groups are equal thus there apperas to be a difference in distribution of time for rats with restricted diet and rats without restricted diets.

In [119]:
```
library('sm')
data_loc<-'/Users/Alexis/Documents/Spring2020/nonparametrics/data/rats.csv'
rat<-read.csv(data_loc)
no_diet=bw.SJ(rat$time[rat$diet==0])
yes_diet=bw.SJ(rat$time[rat$diet==1])
h=2*(1/no_diet+1/yes_diet)^-1

sm.density.compare(rat$time, group=rat$diet, model="equal", bw=h, nboot=10000)
```

```
Test of equal densities:  p-value =  0
```



In [ ]:
```

```