

KDE and BoxCox

```
if (!require("KernSmooth")) install.packages("KernSmooth", dep=TRUE)
```

```
## Loading required package: KernSmooth
```

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

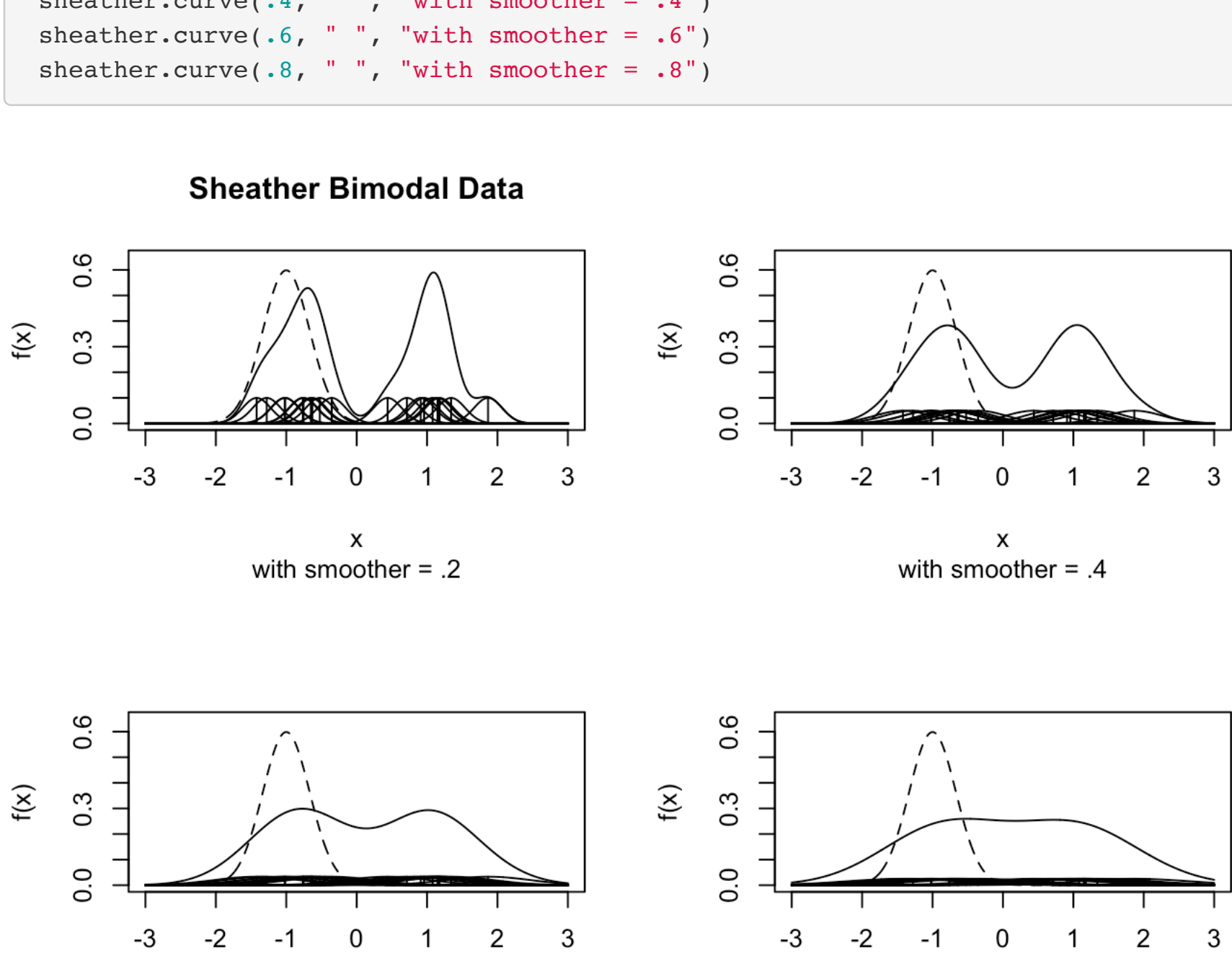
```
library("KernSmooth")
bimodal <- read.table("bimodal.txt", header=TRUE)
attach(bimodal)
summary(bimodal)
```

```
##           x
## Min.   -1.42099
## 1st Qu.: 0.75715
## Median: 0.04272
## Mean   : 0.11878
## 3rd Qu.: 1.09593
## Max.    1.86610
```

```
x <- bimodal$x
n<-length(x)
xx <- c(-300:300)/100
```

```
sheather.curve = function(h, main=" ", sub = " ") {
  truedensity = 0.5*(3/(sqrt(2*pi)))*exp(-0.5*((xx+1)/(1/3))^2)
  + 0.5*(3/(sqrt(2*pi)))*exp(-0.5*((xx-1)/(1/3))^2)
  plot( xx=(-2:3), y=c(0,0.65), type="n", xlab="x", ylab="f(x)")
  title(main=main, sub = sub)
  ysum = numeric(601)
  for (i in 1:n)
  {points(x[i], 1/(n*h*sqrt(2*pi)), type="h")
    x1 = numeric(601)*x[i]
    y1 = (1/(h*sqrt(2*pi)))*exp(-0.5*((xx-x1)/h)^2)
    ysum = y/n + ysum
    lines(xx,y/n,lty=1)}
  lines(xx,ysum,lty=1)
  lines(xx,truedensity,lty=2)
}
```

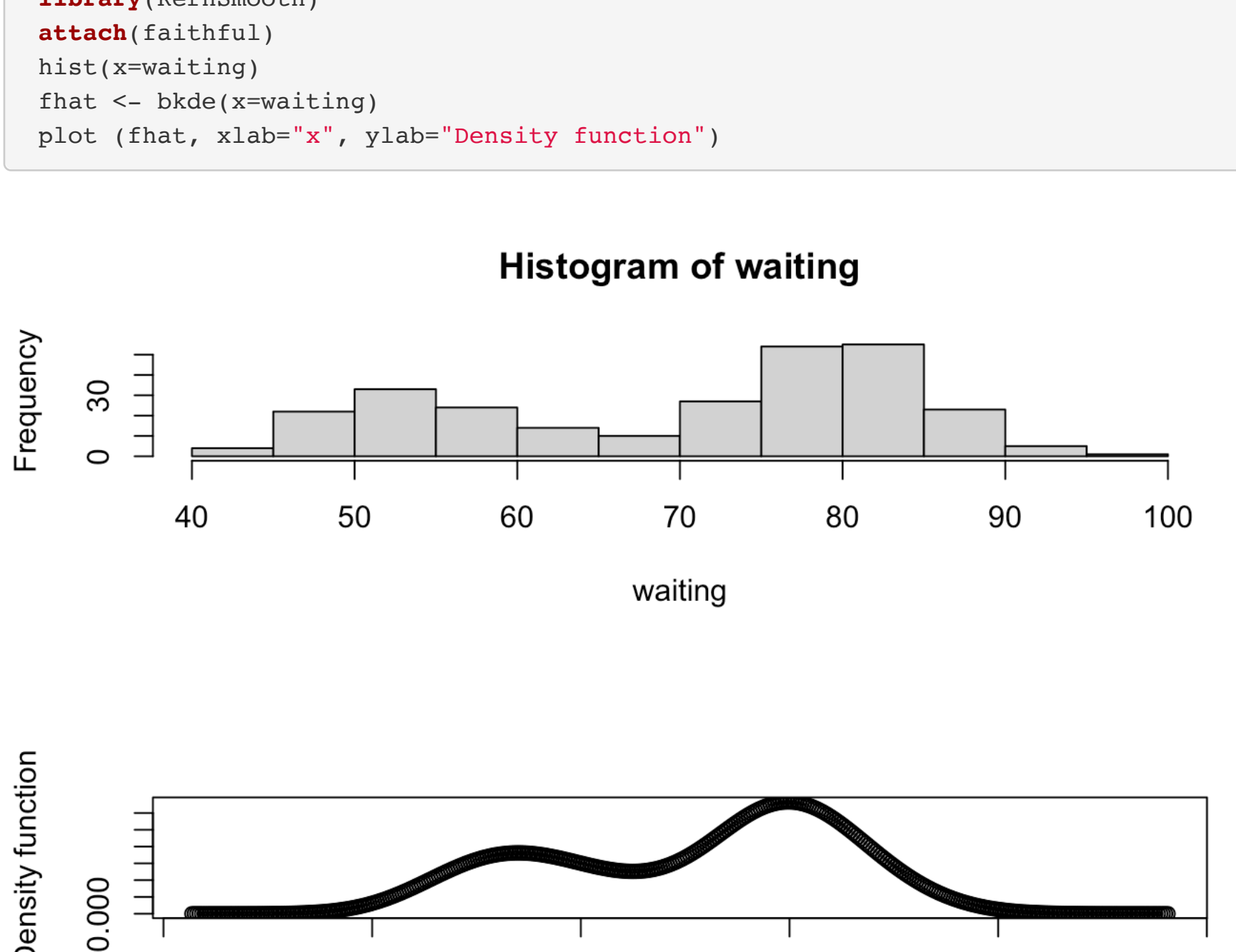
```
par(mfrow=c(2,2))
sheather.curve(.2, "Sheather Bimodal Data", "with smoother = .2")
sheather.curve(.4, " ", "with smoother = .4")
sheather.curve(.6, " ", "with smoother = .6")
sheather.curve(.8, " ", "with smoother = .8")
```



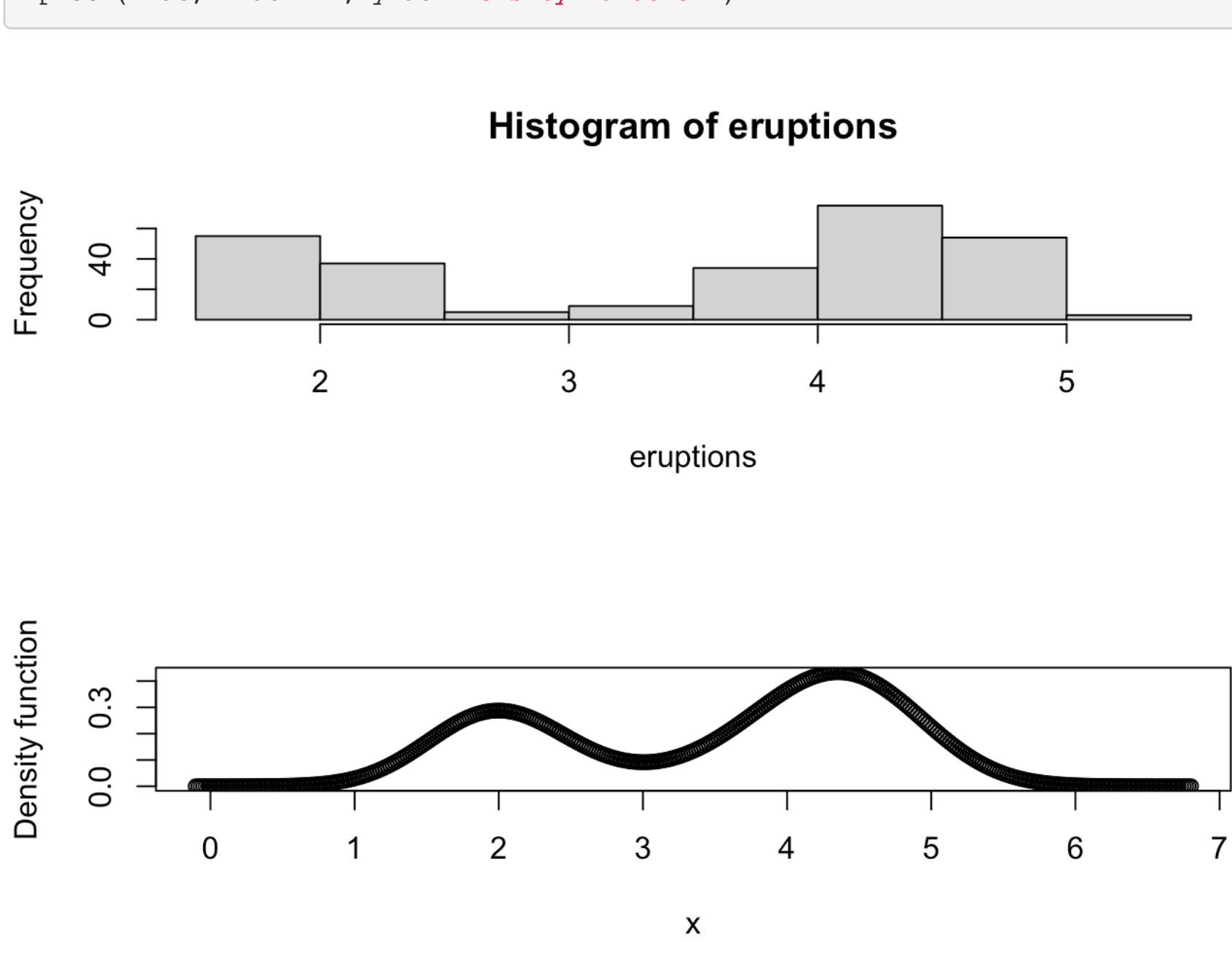
The Old Faithful geyser data

Waiting Time

```
par(mfrow=c(2,1))
library(KernSmooth)
attach(faithful)
hist(x=waiting)
fhat <- bkde(x=waiting)
plot(fhat, xlab="x", ylab="Density function")
```



```
par(mfrow=c(2,1))
hist(x=eruptions)
fhat <- bkde(x=eruptions)
plot(fhat, xlab="x", ylab="Density function")
```



Regression model for Old Faithful data

```
mod1 = lm(waiting ~ eruptions, data=faithful)
summary(mod1)
```

```
## Call:
## lm(formula = waiting ~ eruptions, data = faithful)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.0796  -4.4831   0.2122   3.9246  15.9719
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  33.4744      1.1549   28.98  <2e-16 ***
## eruptions    10.7296      0.3148   34.09  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.914 on 270 degrees of freedom
## Multiple R-squared:  0.8115, Adjusted R-squared:  0.8108
## F-statistic: 1162 on 1 and 270 DF, p-value: < 2.2e-16
```

```
covb = vcov(mod1)
coeff = coef(mod1)
covb = vcov(mod1)
covb
```

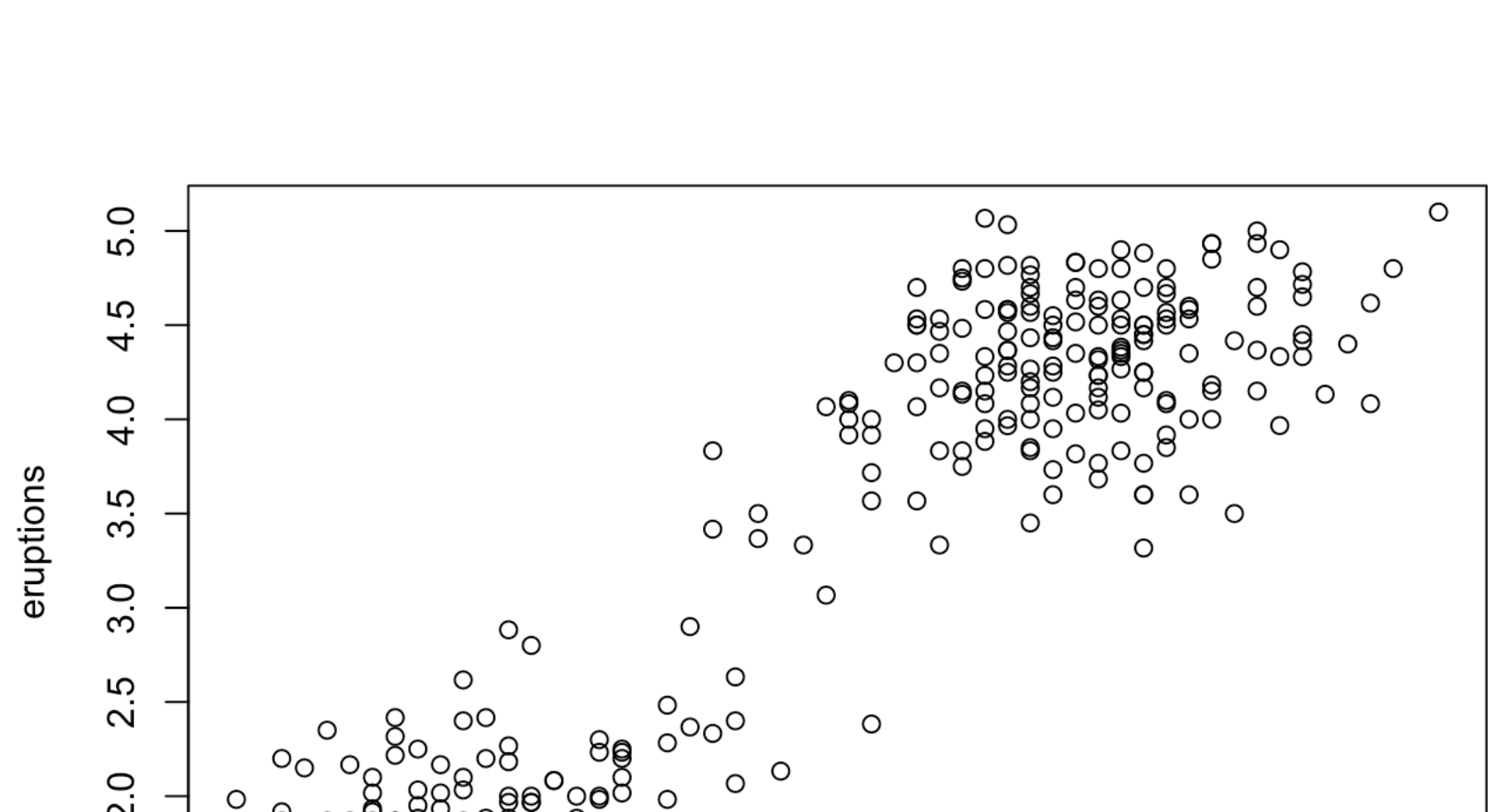
```
##              (Intercept) eruptions
## (Intercept)  1.337328 -0.34553365
## eruptions    -0.3455336  0.09906971
```

```
pred.per_fat = predict(mod1)
res.per_fat = residuals(mod1)
summary(res.per_fat)
```

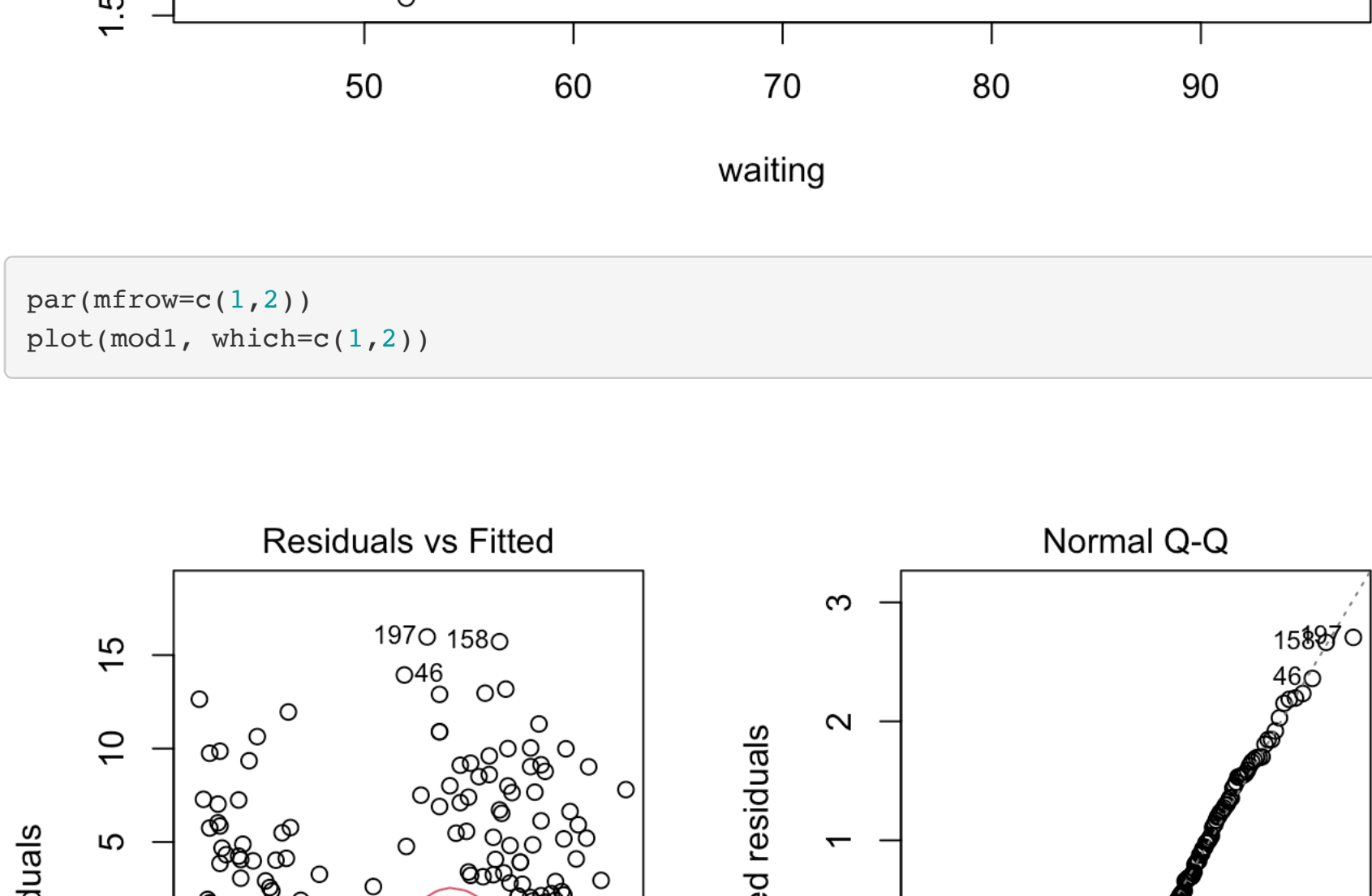
```
##      Min.   1st Qu.   Median     Mean   3rd Qu.     Max.
## -12.0796  -4.4831   0.2122   0.0000   3.9246   15.9719
```

Plots of regression

```
par(mfrow=c(1,1))
plot(waiting,eruptions)
```



```
par(mfrow=c(1,2))
plot(mod1, which=c(1,2))
```



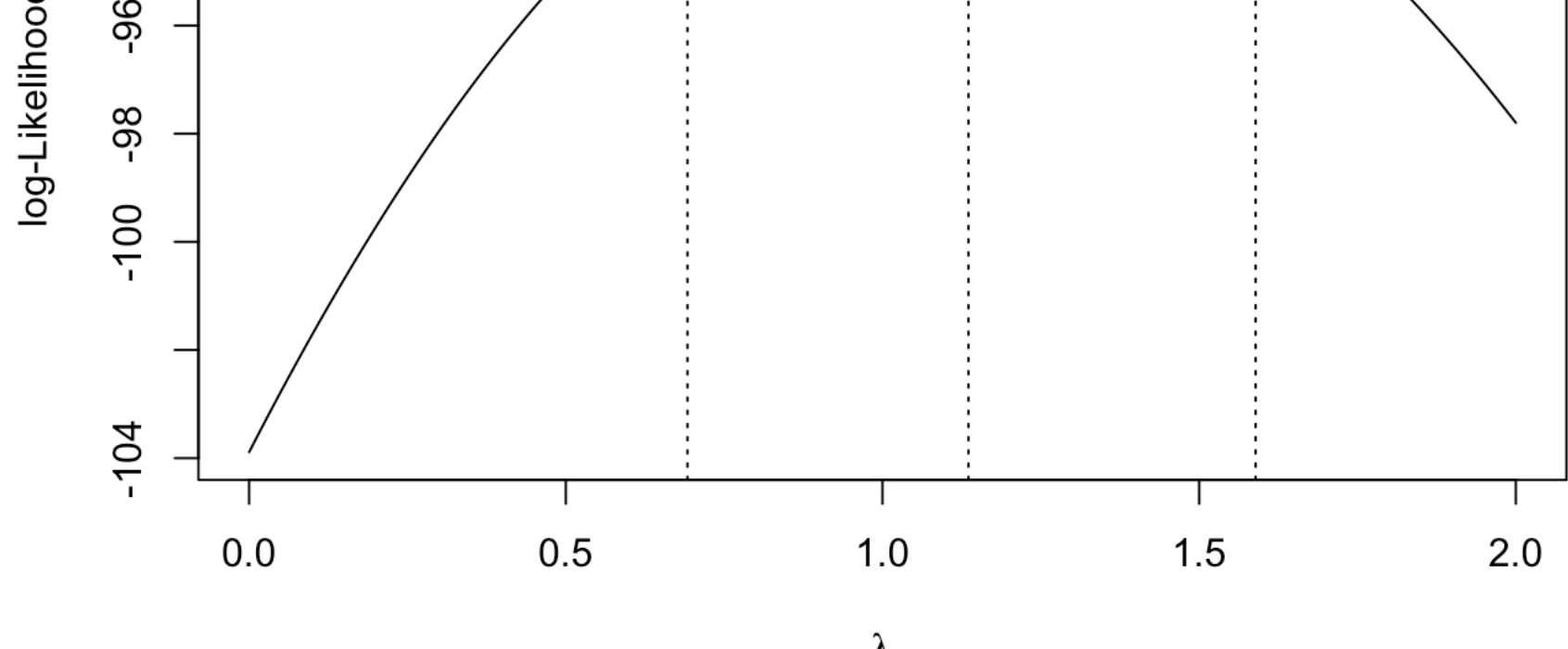
Box Cox transformation

```
library(MASS)
boxcox(waiting ~., data=faithful, lambda=seq(0, 2.0, length=200))
```



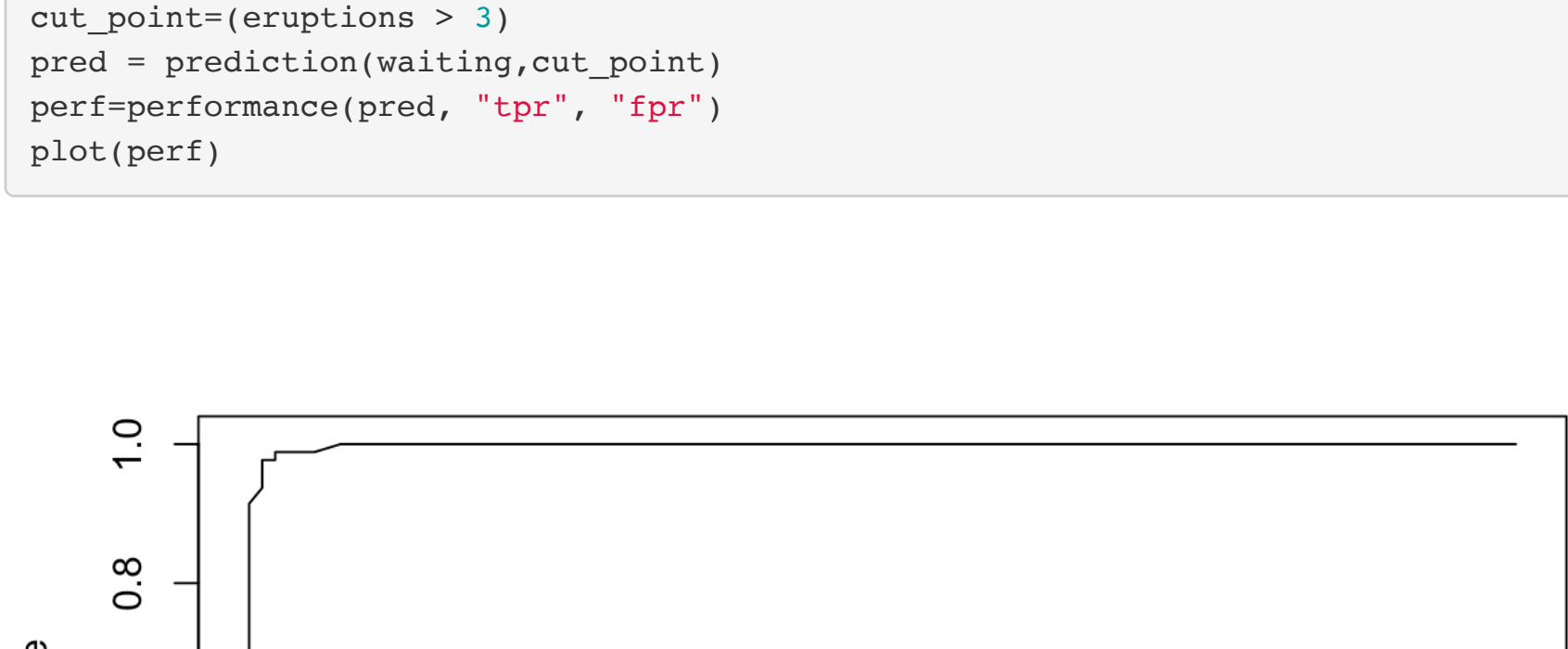
ROC Curves for eruption > 3 minutes

```
library(ROCR)
cut_point=(eruptions > 3)
pred = prediction(waiting,cut_point)
perf=performance(pred, "tpr", "fpr")
plot(perf)
```



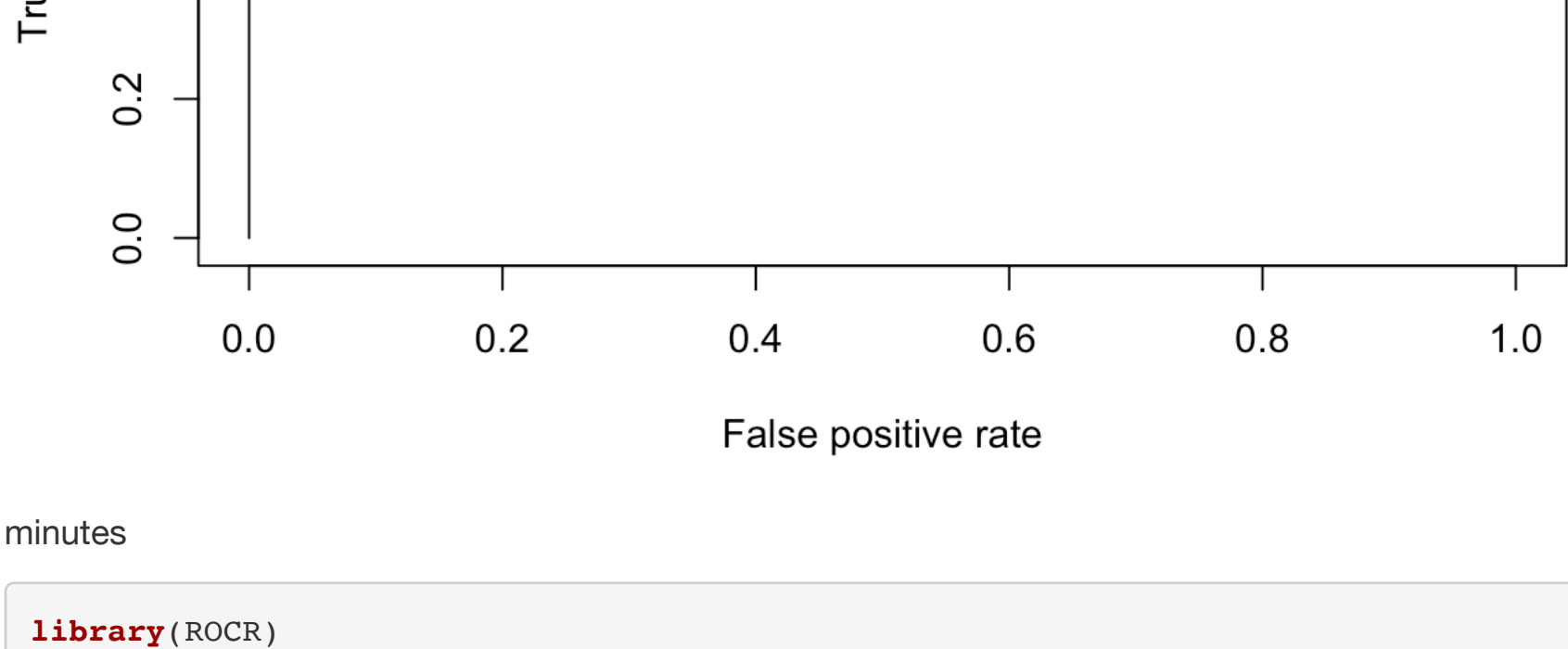
ROC Curves for eruption > 4.2

```
library(ROCR)
cut_point=(eruptions > 4.2)
pred = prediction(waiting,cut_point)
perf=performance(pred, "tpr", "fpr")
plot(perf)
```



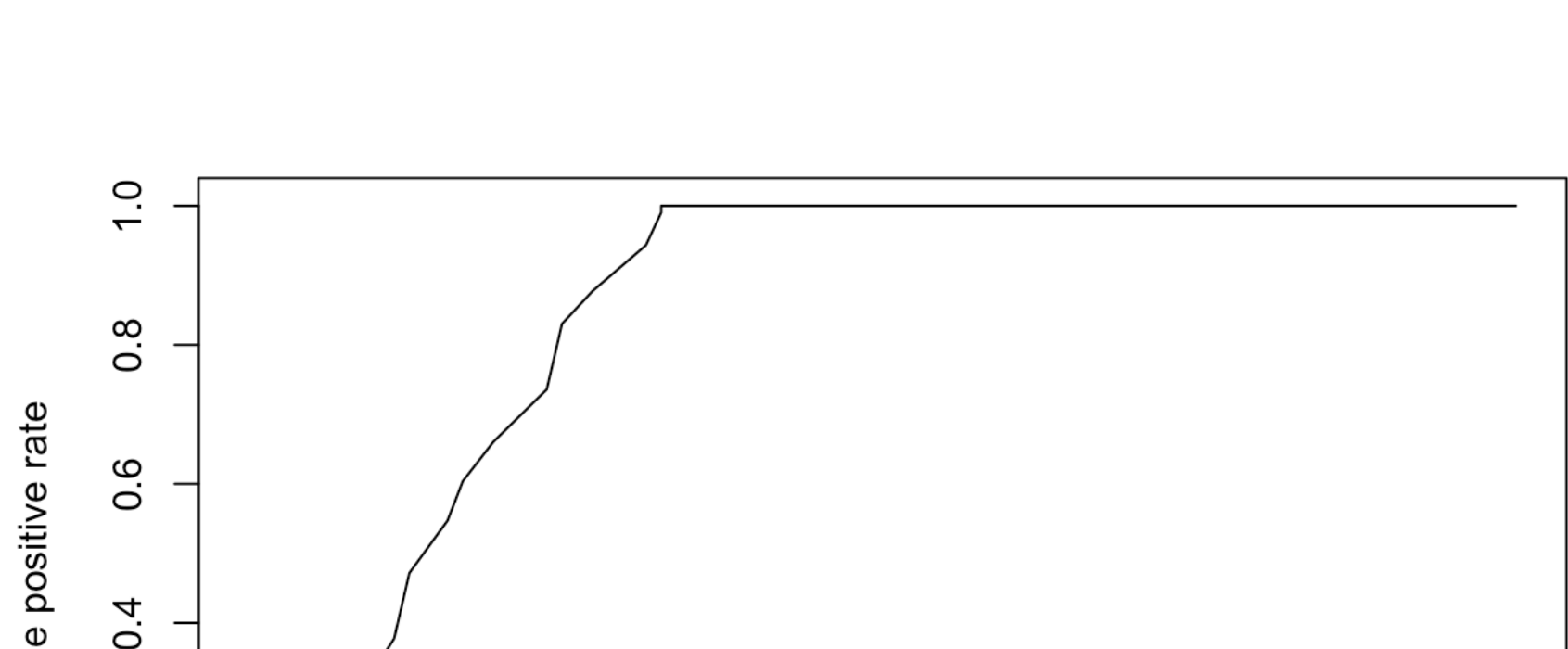
minutes

```
library(ROCR)
cut_point=(eruptions > 4.2)
pred = prediction(waiting,cut_point)
perf=performance(pred, "tpr", "fpr")
plot(perf)
```



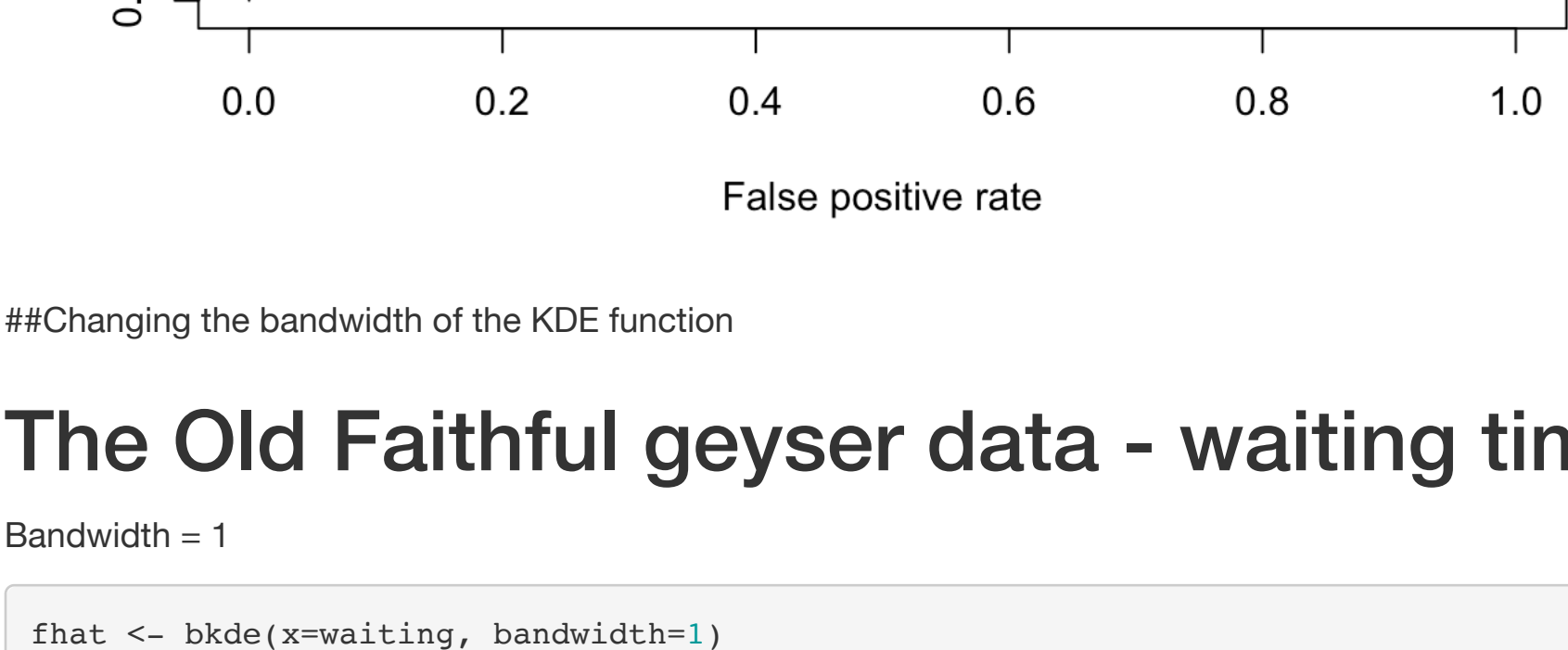
minutes

```
library(ROCR)
cut_point=(eruptions > 4.2)
pred = prediction(waiting,cut_point)
perf=performance(pred, "tpr", "fpr")
plot(perf)
```



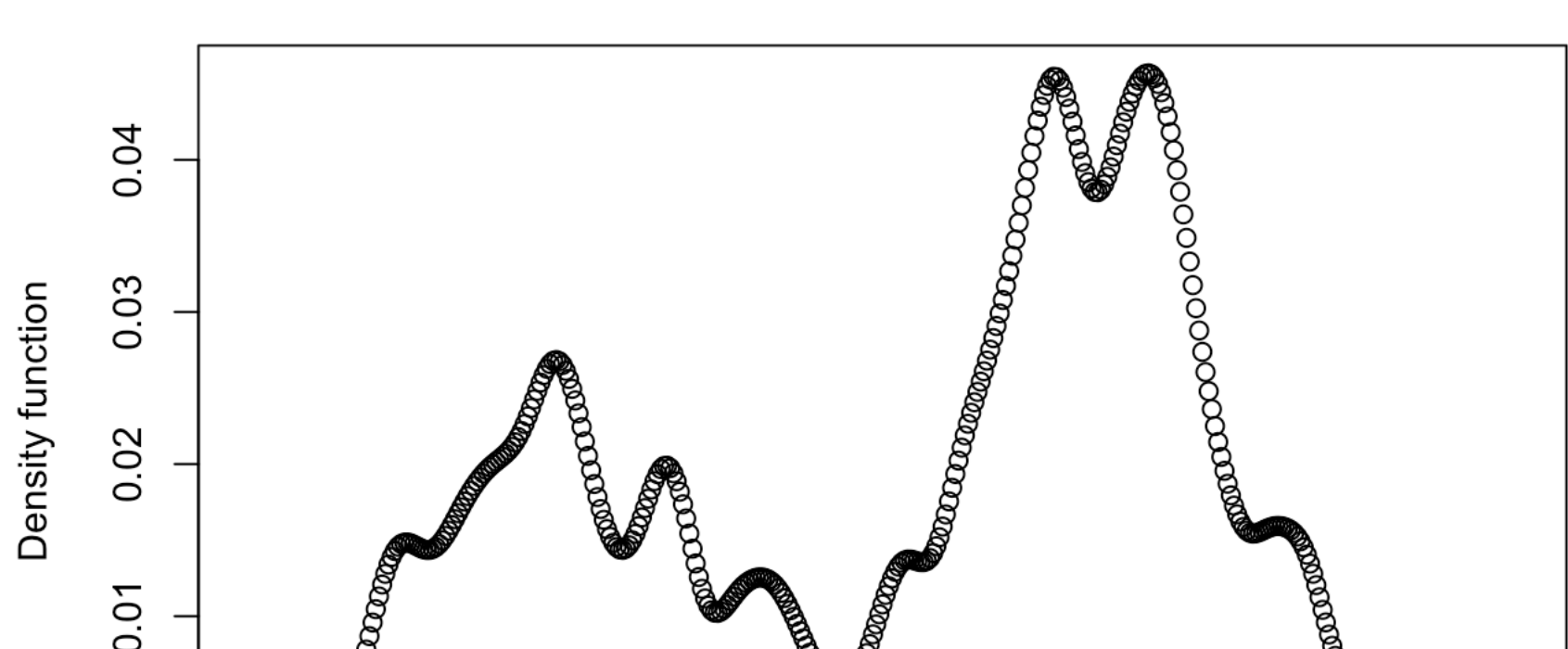
minutes

```
library(ROCR)
cut_point=(eruptions > 4.2)
pred = prediction(waiting,cut_point)
perf=performance(pred, "tpr", "fpr")
plot(perf)
```



minutes

```
library(ROCR)
cut_point=(eruptions > 4.2)
pred = prediction(waiting,cut_point)
perf=performance(pred, "tpr", "fpr")
plot(perf)
```



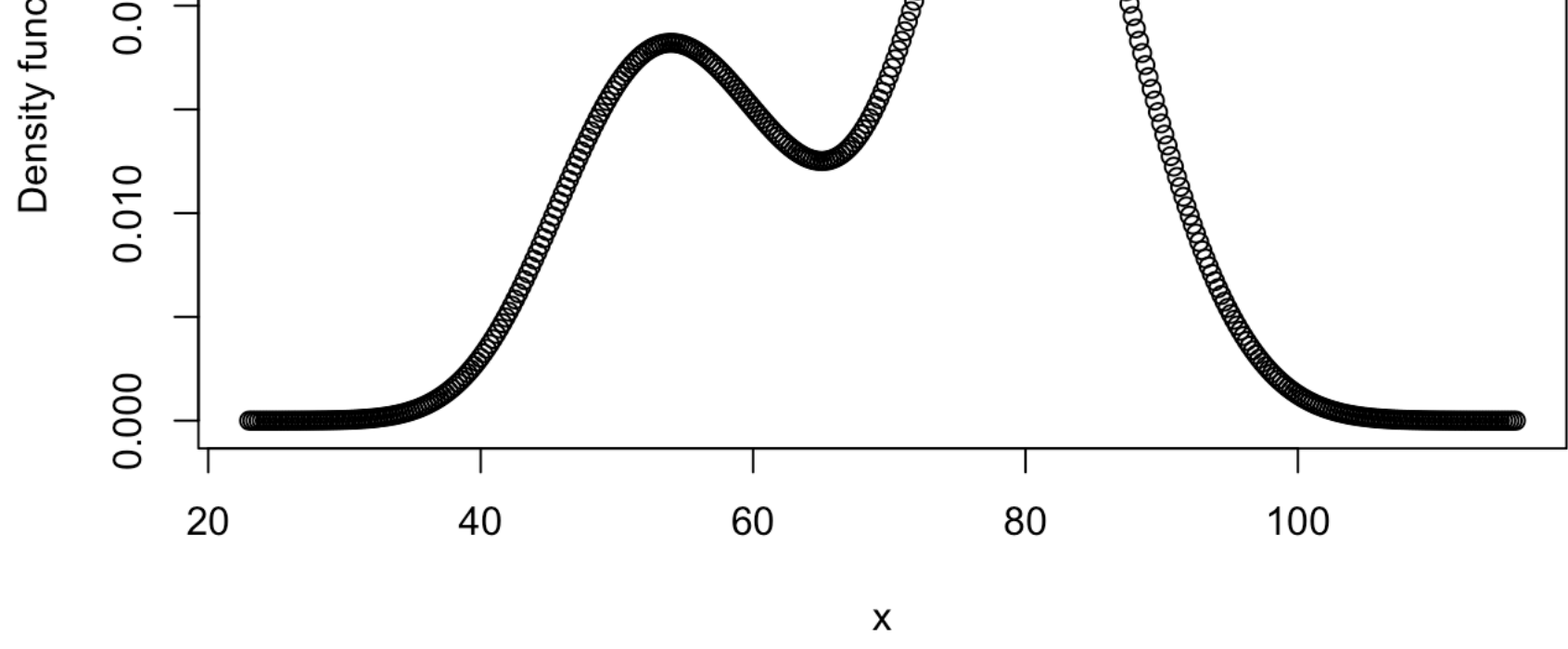
minutes

```
library(ROCR)
cut_point=(eruptions > 4.2)
pred = prediction(waiting,cut_point)
perf=performance(pred, "tpr", "fpr")
plot(perf)
```



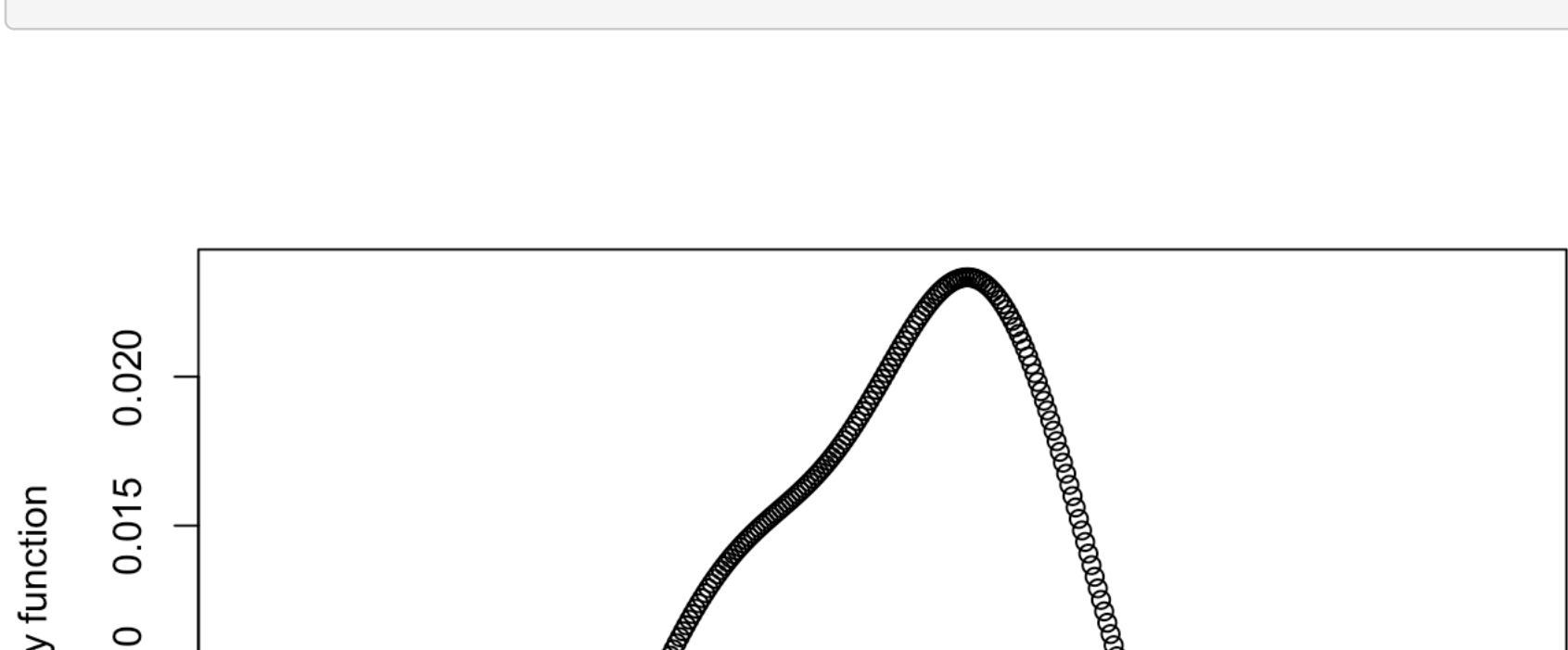
minutes

```
library(ROCR)
cut_point=(eruptions > 4.2)
pred = prediction(waiting,cut_point)
perf=performance(pred, "tpr", "fpr")
plot(perf)
```



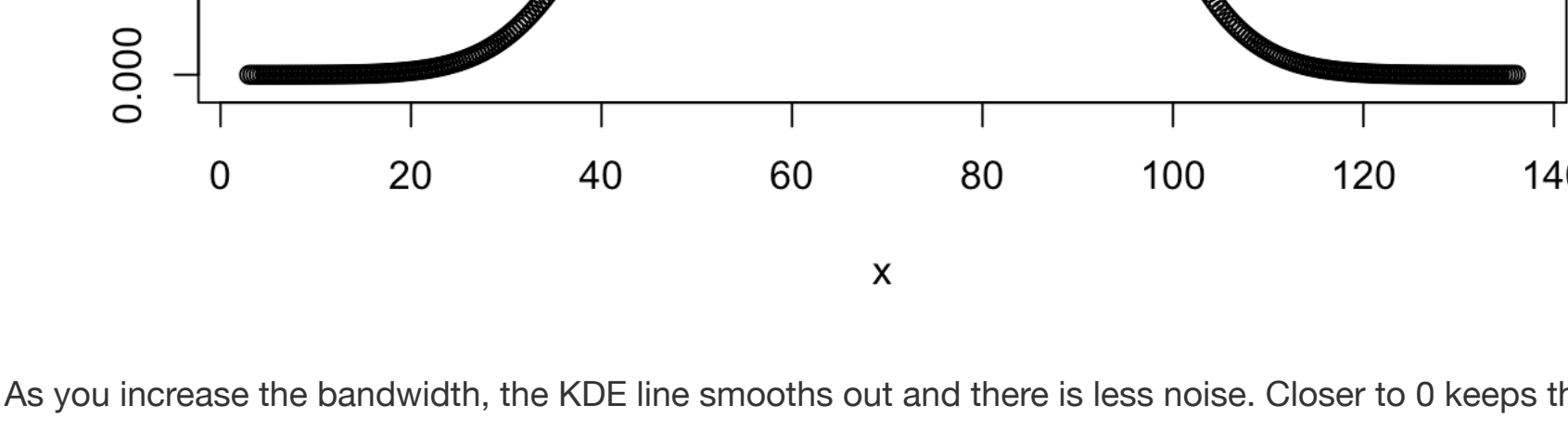
minutes

```
library(ROCR)
cut_point=(eruptions > 4.2)
pred = prediction(waiting,cut_point)
perf=performance(pred, "tpr", "fpr")
plot(perf)
```



minutes

```
library(ROCR)
cut_point=(eruptions > 4.2)
pred = prediction(waiting,cut_point)
perf=performance(pred, "tpr", "fpr")
plot(perf)
```



As you increase the bandwidth, the KDE line smooths out and there is less noise. Closer to 0 keeps the noise.