KDE and BoxCox

jdt

12/29/2020

Contents

Theory

Kernel Density Estimation

A procedure used for estimating a probability density function using the observed data is considered. As with histograms, the procedure is used to provide a graphical representation of the pdf of X using the observed data, $x_1, x_2, \dots, x_n$. The estimate is called the *kernel density estimate*. Kernel density estimation is a nonparametric technique for density estimation in which a known density function (the kernel) is averaged across the observed data points to create a smooth approximation for $f_X(x)$. SAS PROC KDE uses a Gaussian density as the kernel, and its assumed variance determines the smoothness of the resulting estimate. See Silverman (1986) for a thorough review and discussion.

Computational Methods

Univariate Kernel Density Estimates - SAS

Let (X_i, W_i), denote the observed sample of X_i with specified weight W_i for $i = 1, 2, \dots, n$. The weighted kernel density estimate of $f(x)$, the density of X, is

$$\hat{f}(x) = \frac{1}{\sum_{i=1}^{n} W_i} \sum_{i=1}^{n} \varphi_h(x - X_i)$$

where h is the bandwidth and

$$\varphi_h(x) = \frac{1}{\sqrt{2\pi}h} \exp\left(-\frac{x^2}{2h^2}\right)$$

is the normal density rescaled by the bandwidth ($N(0, h^2)$). If $h \to 0$ and $nh \to \infty$, then the optimal bandwidth is

$$h_{AMISE} = \left[\frac{1}{2\sqrt{\pi}n \int (f'')^2}\right]^{1/5}$$

where $\frac{\partial^2 f}{\partial x} = f''$. Since the optimal value is unknown, approximations methods are needed. For a derivation and discussion of these methods, see Silverman (1986) and Jones, Marron, and Sheather (1996).

General Univariate Kernel Density Estimate

Assume that $W_i = 1$, in which case the kernel density estimate for $f(x)$ is,

$$\hat{f}(x) = \frac{1}{nh}\sum_{i=1}^{n} K(\frac{x - X_i}{h})$$

where the kernel, K, satisfies $\int K(x)dx = 1$ and the smoothing parameter, h, is called the bandwidth. In practice, the kernel is an unimodal function satisfying, $\int xK(x)dx = 0$.[1] A popular choice for the *normal* kernel given by

$$K(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right).$$

If one assumes that the underlying density is sufficiently smooth and that the kernel has finite forth moments, then an asymptotic expansion for the bias and variance of a kernel estimate are given by

$$Bias_{asy}\{\hat{f}_h(x)\} = \frac{h^2}{2}\mu_2(K)^2 f''(x)$$

and

$$Var_{asy}\{\hat{f}_h(x)\} = \frac{1}{nh}R(K)f(x)$$

where $R(K) = \int K^2(y)dy$, $\mu_2(K) = \int y^2 K(y)dy$ and f'' is the second derivative of f.

A widely used criteria for measuring the discrepancy between f and $\hat{f}$ is the mean integrated squared error (MISE) is given by,

$$\begin{aligned} MISE(\hat{f}) &= E\{\int \left(f(y) - \hat{f}(y)\right)^2 dy\} \\ &= \int Bias(\hat{f}(y))^2 dy + \int Var(\hat{f}(y))dy. \end{aligned}$$

If one assumes that the function f is integrable, then the asymptotic mean integrated squared error (AMISE) is given by

$$AMISE(\hat{f}(_h) = \frac{1}{nh}R(K) = \frac{h^2}{4}\mu_2(K)^2 R(f'').$$

The bandwidth that minimizes the AMISE is given by

$$h_{AMISE} = \{\frac{R(K)}{\mu_2(K)^2 R(f'')}\}^{1/3} n^{-1/3}.$$

Bandwidth Selection

Several different bandwidth selection methods are available in PROC KDE in the univariate case. Following the recommendations of Jones, Marron, and Sheather (1996), the default method follows a plug-in formula of Sheather and Jones.

This method solves the fixed-point equation

$$h = \left[\frac{R(\varphi)}{nR\left(\hat{f}''_{g(h)}\right)\left(\int x^2\varphi(x)dx\right)^2}\right]^{1/5}$$

[1] Note: the kernel, $K(x)$ is a unimodal density function with expected value, $E_K(X) = \int xK(x)dx = 0$.

where $R(\varphi) = \int \varphi^2(x)dx$ and $g(h) = C(K)[R(f'')/R(f''')]^{1/7}h^{5/7}$ is the bandwidth for the estimate of $R(\hat{f}'')$.

PROC KDE solves this equation by first evaluating it on a grid of values spaced equally on a log scale. The largest two values from this grid that bound a solution are then used as starting values for a bisection algorithm. The simple normal reference rule works by assuming $\hat{f}$ is Gaussian in the preceding fixed-point equation. This results in

$$\begin{aligned} h &= \hat{\sigma}[4/(3n)]^{1/5} \\ &= 1.06\,\hat{\sigma}n^{-1/5} \end{aligned}$$

where $\hat{\sigma}$ is the sample standard deviation.

Alternatively, the bandwidth can be computed using the interquartile range,

$$\begin{aligned} h &= 1.06\hat{\sigma}n^{-1/5} \\ &\approx 1.06\hat{\sigma}n^{-1/5} \\ &\approx 1.06\,(Q/1.34)n^{-1/5} \end{aligned}$$

Silverman's rule of thumb (Silverman, 1986, Section 3.4.2) is computed as

$$h = 0.9\min[\hat{\sigma}, Q/1.34]n^{-1/5}$$

The oversmoothed bandwidth is computed as

$$h = 3\hat{\sigma}[1/(70\sqrt{\pi}n)]^{1/5}$$

When you specify a WEIGHT variable, PROC KDE uses weighted versions of Q_3, Q_1, and $\hat{\sigma}$ in the preceding expressions. The weighted quartiles are computed as weighted order statistics, and the weighted variance

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^{n} W_i(X_i - \bar{X})^2}{\sum_{i=1}^{n} W_i}$$

where $\bar{X} = (\sum_{i=1}^{n} W_i X_i)/(\sum_{i=1}^{n} W_i)$ is the weighted sample mean.

Box Cox Transformations

Suppose that $y > 0$, define the Box-Cox transformation as

$$y_i^{(\lambda)} = \begin{cases} (y_i^{\lambda} - 1)/\lambda & when \quad \lambda \neq 0, \\ \ln y_i & when \quad \lambda = 0, \end{cases}$$

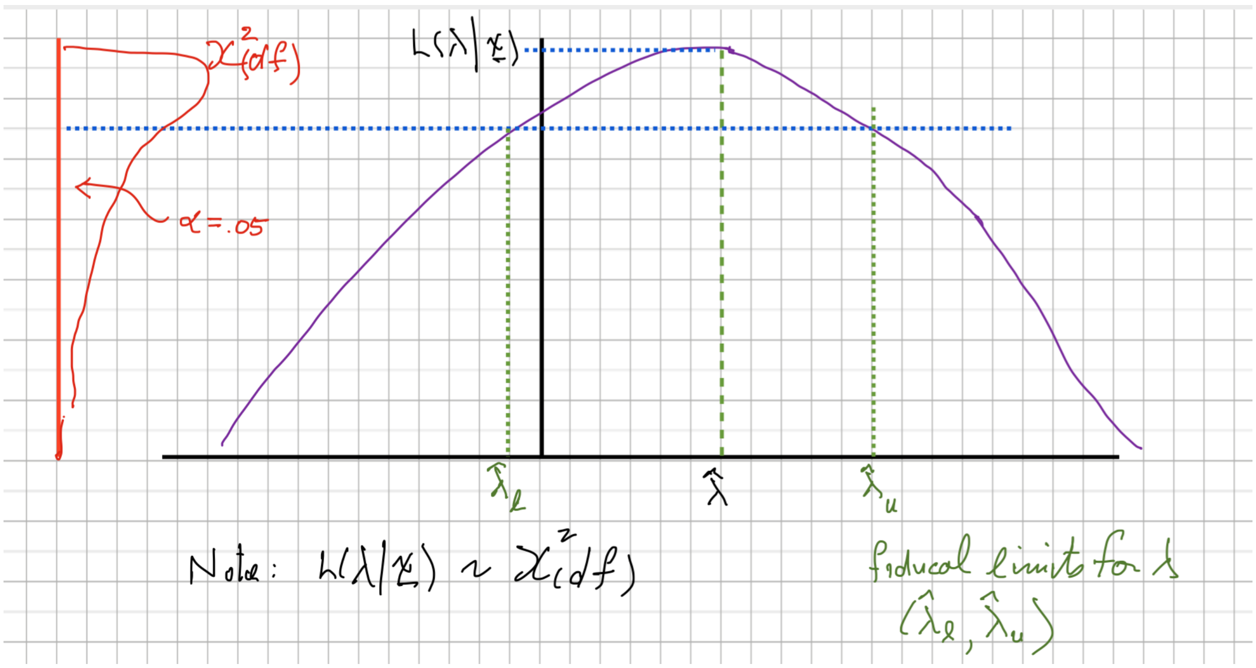
where $i = 1, 2, \ldots, n$. One determines λ by maximizing

$$-n/2\log[s^2(\lambda)] = (\lambda - 1)\sum_{i=1}^{n}\ln(y_i) - n/2\log[\hat{\sigma}^2(\lambda)],$$

and $\hat{\sigma}^2(\lambda) = 1/n\vec{y}^{(\lambda)\prime}[I - H]\vec{y}^{(\lambda)}$ i.e., it is the sum of squares for the error term when $y_i^{(\lambda)}$ is used instead of y_i and $\vec{y}^{(\lambda)} = (y_1^{(\lambda)}, y_2^{(\lambda)}, \ldots, y_n^{(\lambda)})'$.

Since, there is not a close form solution to the above maximization, one usually plots $-n/2\log[s^2(\lambda)]$ vs λ. Another approach is to compute a confidence interval using the fact that $-n/2\log[s^2(\lambda)] \sim \chi^2(df = 1)$. One can then use any λ which is contained in the confidence interval.

Often the output for the BoxCox results is the following figure.



Comment

I did not perform the BoxCox method in SAS.

R Code and Output

Sheather Simulated KDE Example

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

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

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

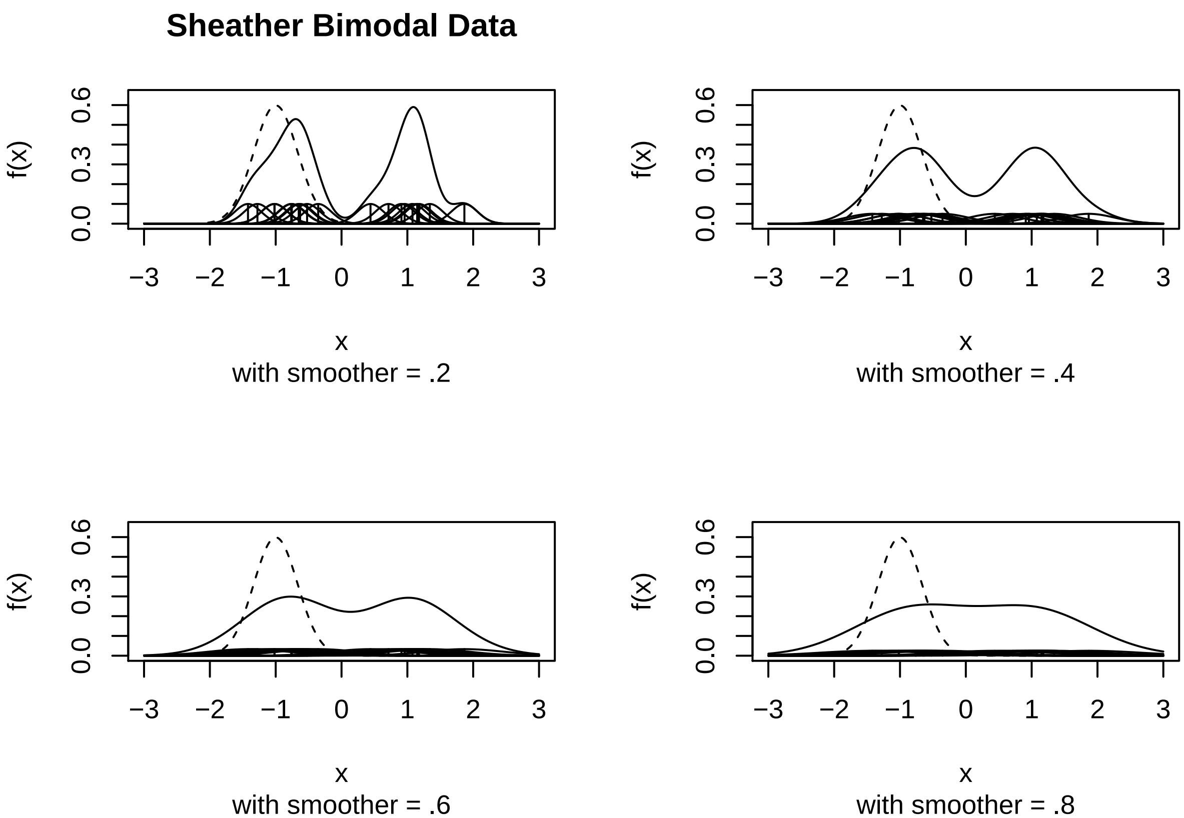
```
library("KernSmooth")
bimodal <- read.table("bimodal.txt", header=TRUE)
attach(bimodal)
```

```
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( x=c(-3,3),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]
     y = (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)
}
```

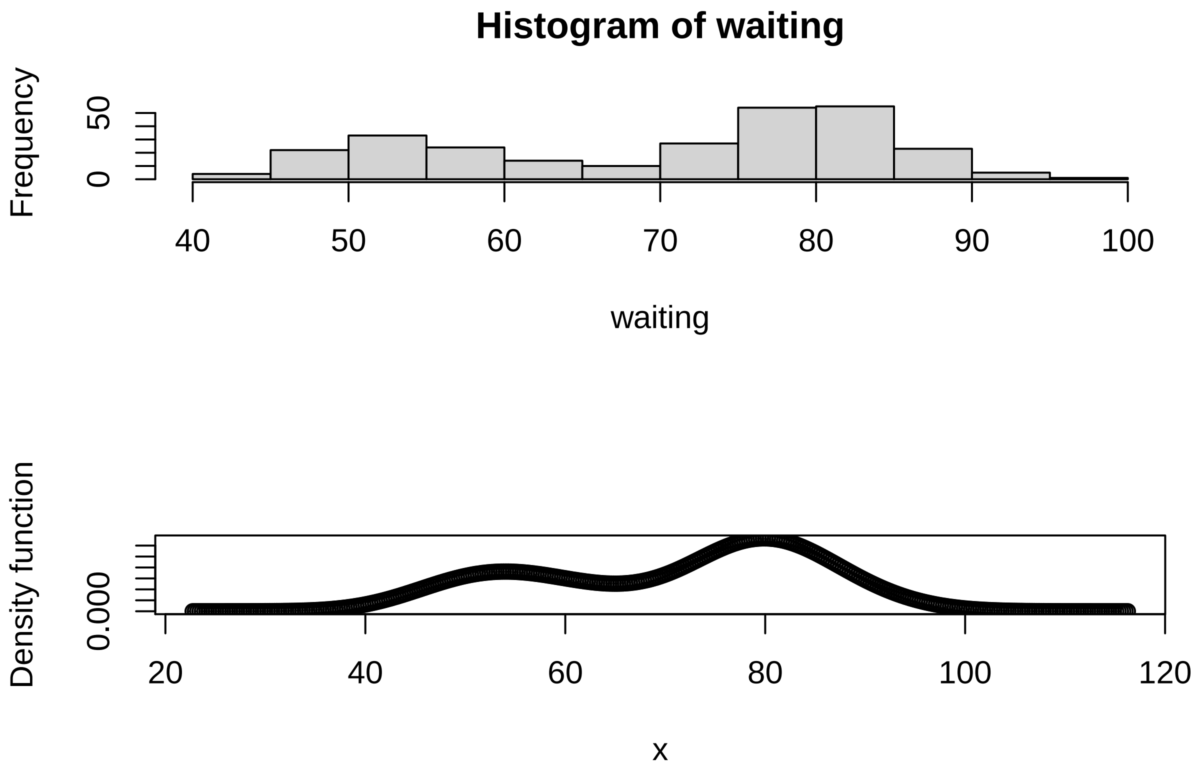
```
par(mfrow=c(2,2))
 sheather.curve(.2, "Sheather Bimodal Data", "with smoother = .2")
 sheather.curve(.4, " ", "with smoother = .4")
 sheather.curve(.6, " ", "with smoother = .6")
 sheather.curve(.8, " ", "with smoother = .8")
```



Old Faithful geyser data

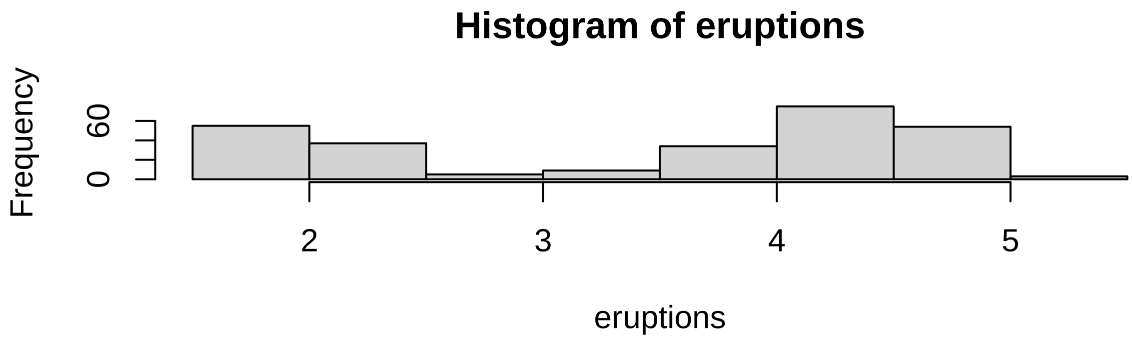
Waiting Time

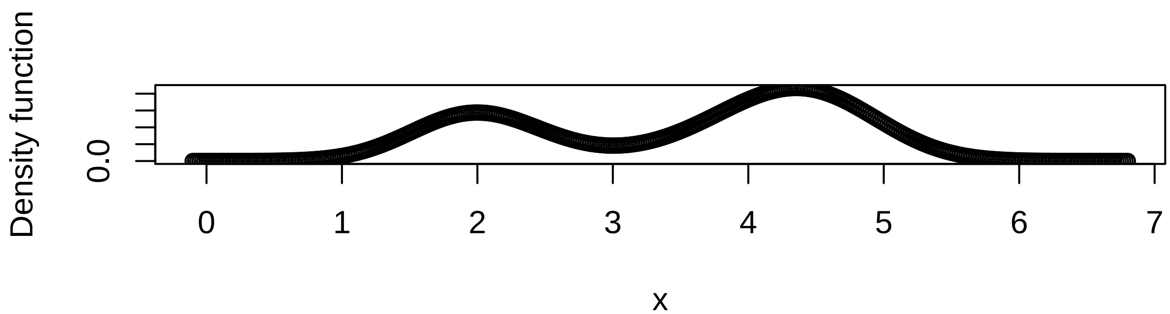
```
par(mfrow=c(2,1))
 library(KernSmooth)
 attach(faithful)
 hist(x=waiting)
 fhat <- bkde(x=waiting)
 plot (fhat, xlab="x", ylab="Density function")
```



Eruption Time

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





Regression

```
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.mod1 = coef(mod1)

covb = vcov(mod1)
covb
```

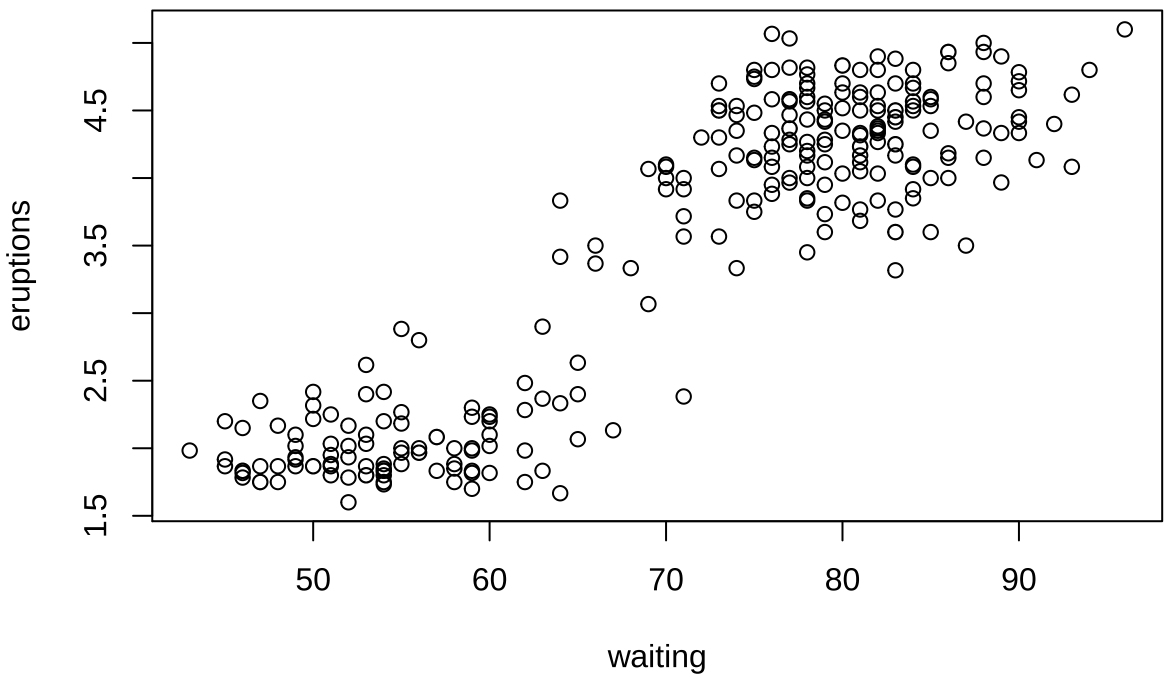
```
##                (Intercept)   eruptions
## (Intercept)   1.3337328 -0.34553365
## eruptions    -0.3455336  0.09906971
```

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

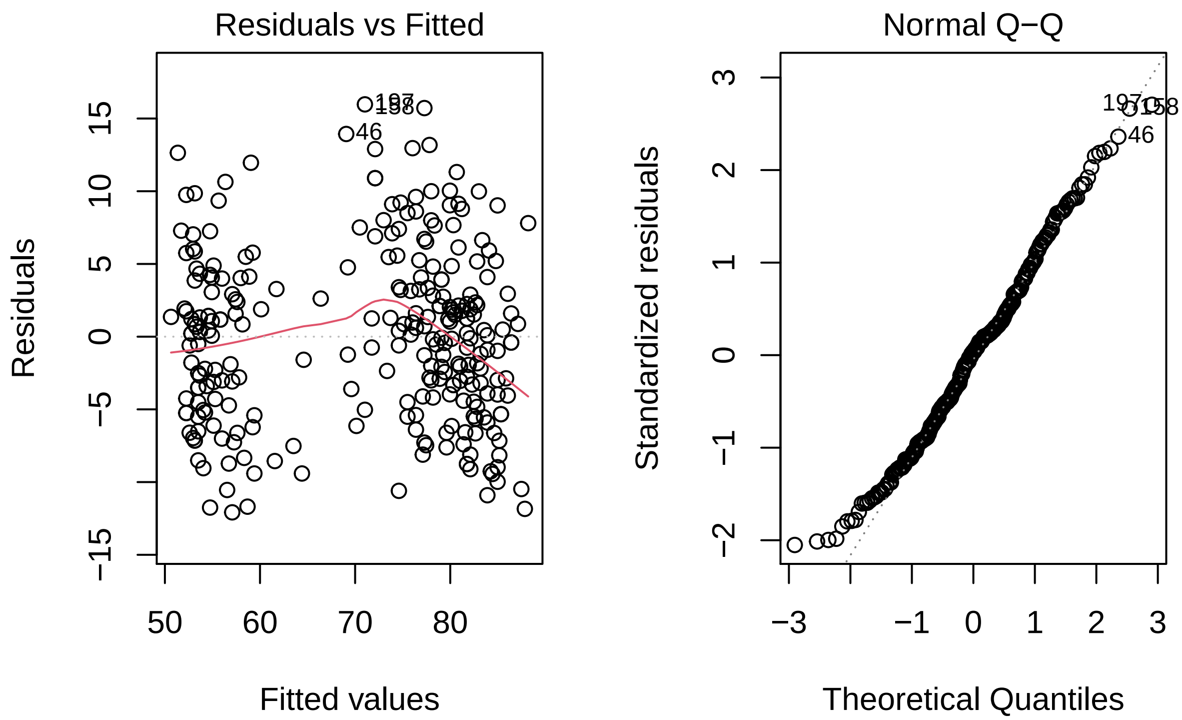
```
##      Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -12.0796  -4.4831   0.2122   0.0000   3.9246  15.9719
```

Plots of regression

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

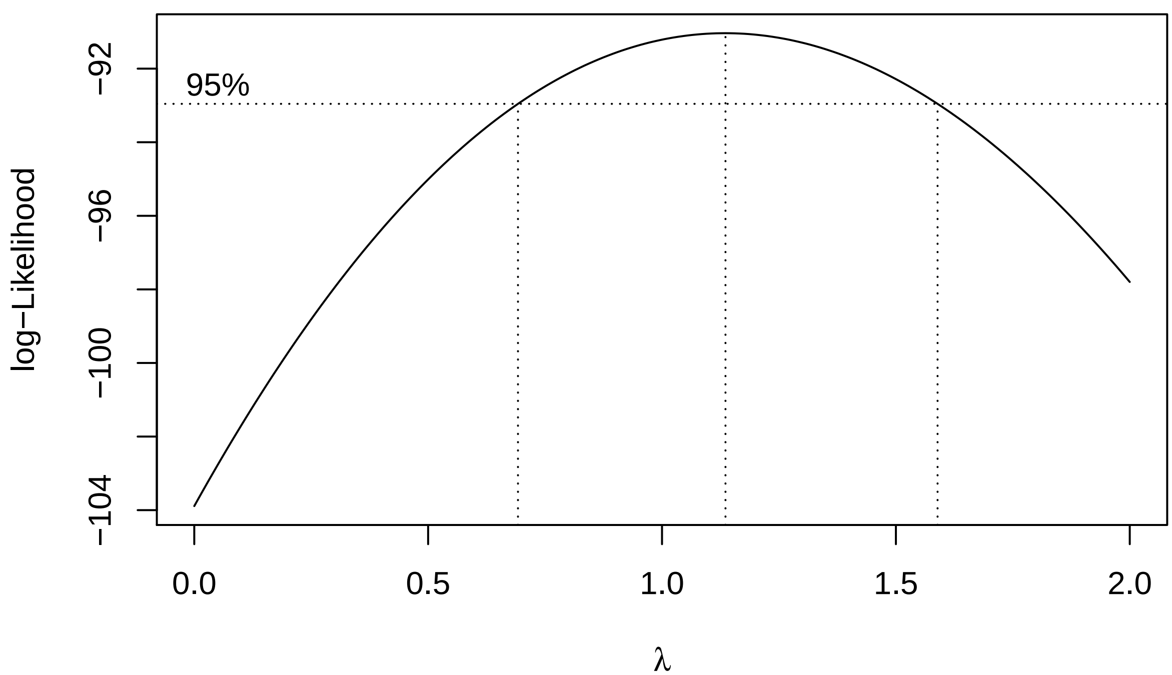



```
par(mfrow=c(1,2))
plot(mod1, which=c(1,2))
```



Box Cox transformation

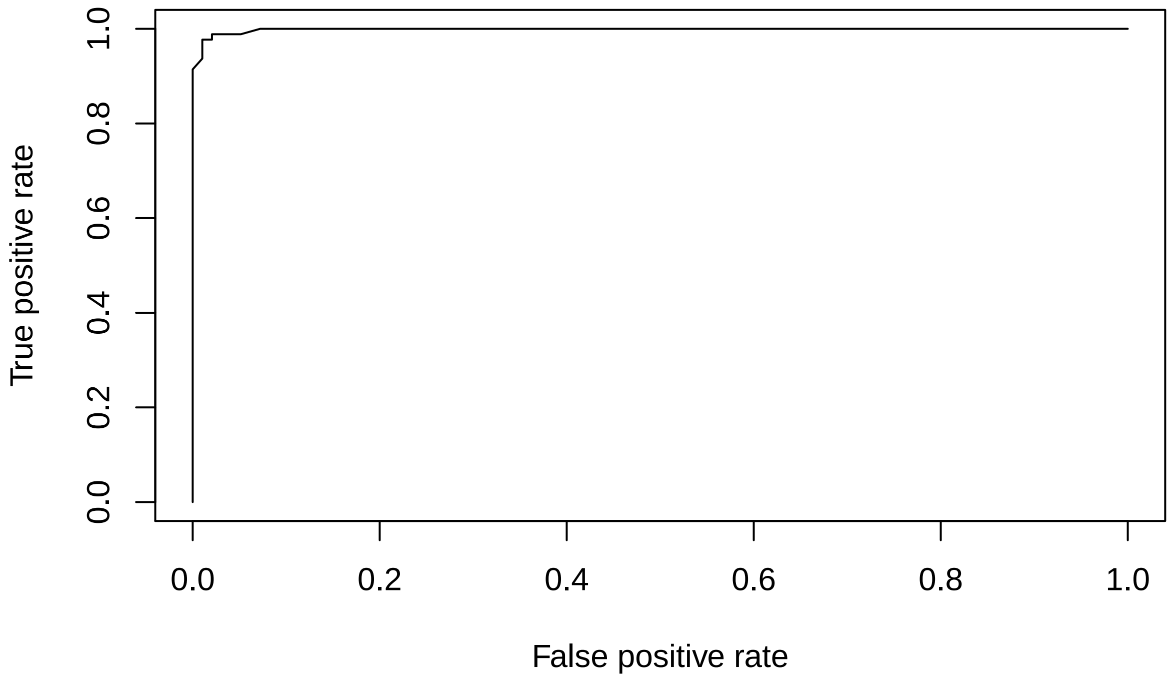
```
library(MASS)
boxcox(waiting ~. ,data=faithful, lambda=seq(0, 2.0, length=200))
```



ROC Curves

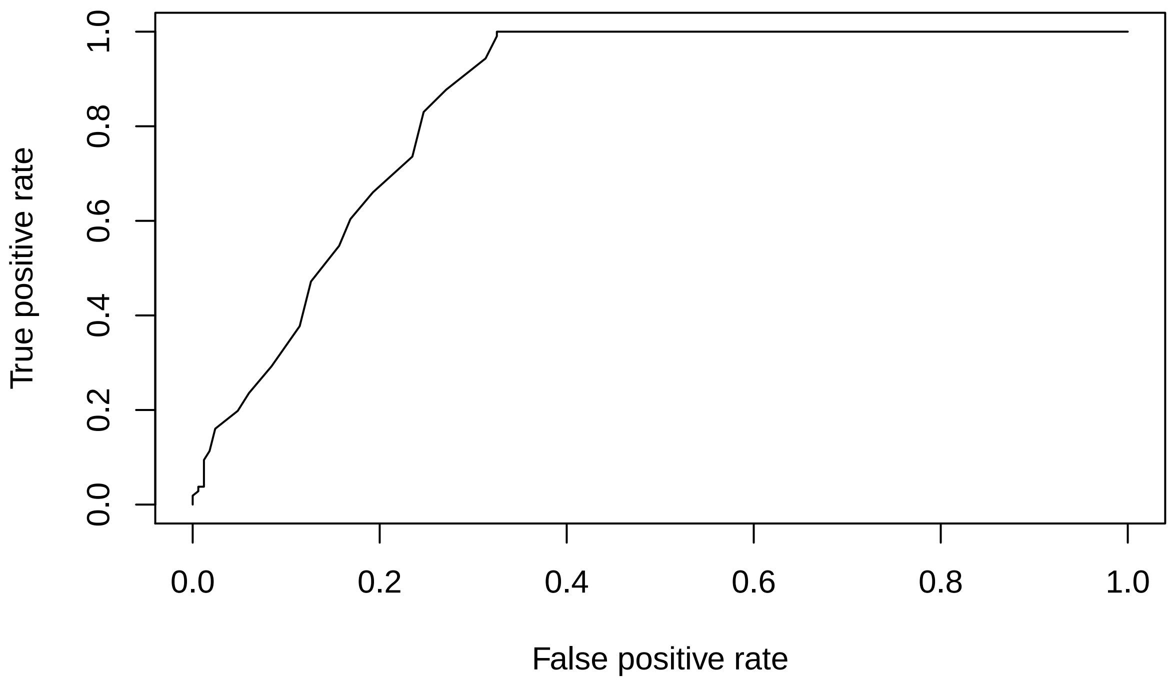
ROC Curves for eruption > 3 minutes

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



ROC Curves for eruption > 4.2 minutes

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



SAS

SAS Code

```
options center nodate pagesize=100 ls=70;
libname LDATA '/home/jacktubbs/my_shared_file_links/jacktubbs/LaTeX/';

/* Simplified LaTeX output that uses plain LaTeX tables */

ods tagsets.simplelatex
file="/home/jacktubbs/my_shared_file_links/jacktubbs/LaTeX/sheather_faithful.tex"
stylesheet="/home/jacktubbs/my_shared_file_links/jacktubbs/LaTeX/sas.sty"
(url="sas");


title 'Sheather KDE Simulated Data';
ods graphics on;

data bimodal; set ldata.bimodal;
run;

proc kde data=bimodal;
   univar x(bwm=.2) x(bwm=0.4) x(bwm=.6) x(bwm=0.8);
run;

title 'Old Faithful Data';

data faithful; set ldata.faithful;
run;

proc sgplot data=faithful;
    scatter y=wait x=duration;
    reg y=wait x=duration/ clm;
```

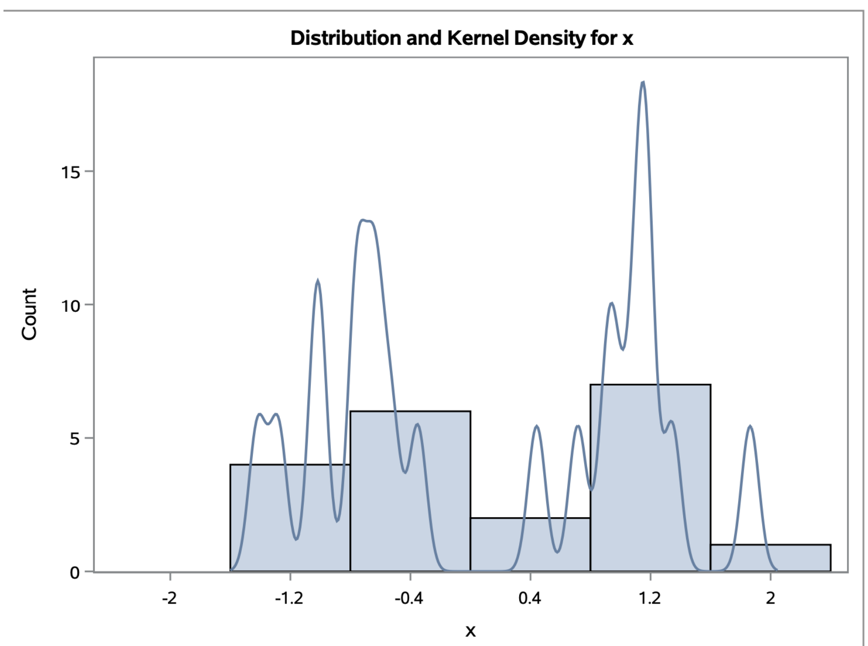
```
    loess y=wait x=duration;
run;

proc sgplot data=faithful;
histogram wait;
density wait;
run;

proc kde data=faithful;
univar wait;
run;

proc kde data=faithful;
   bivar wait duration  ;
run;
quit;
```

Output

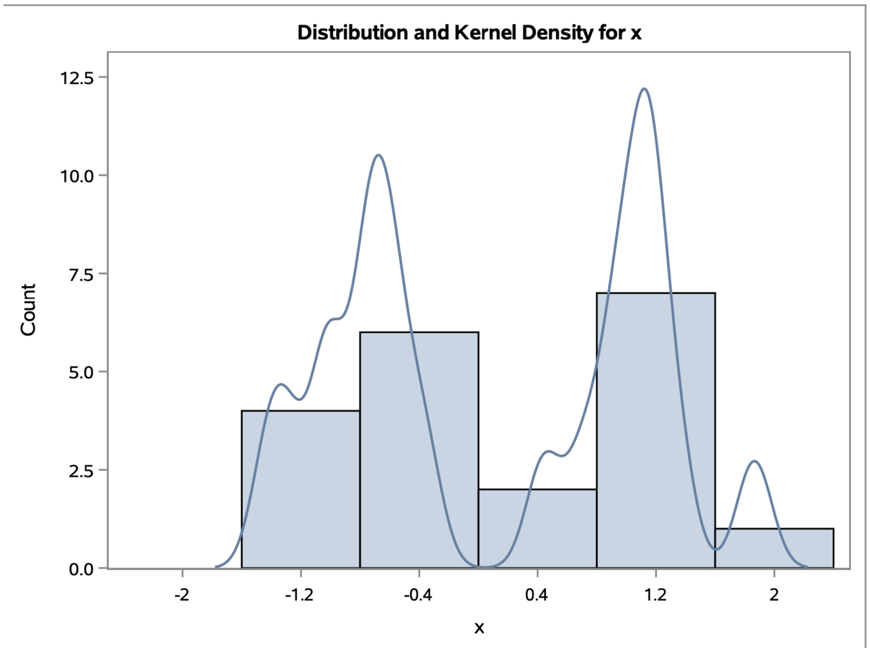
Sheather KDE Simulated Data

The KDE Procedure

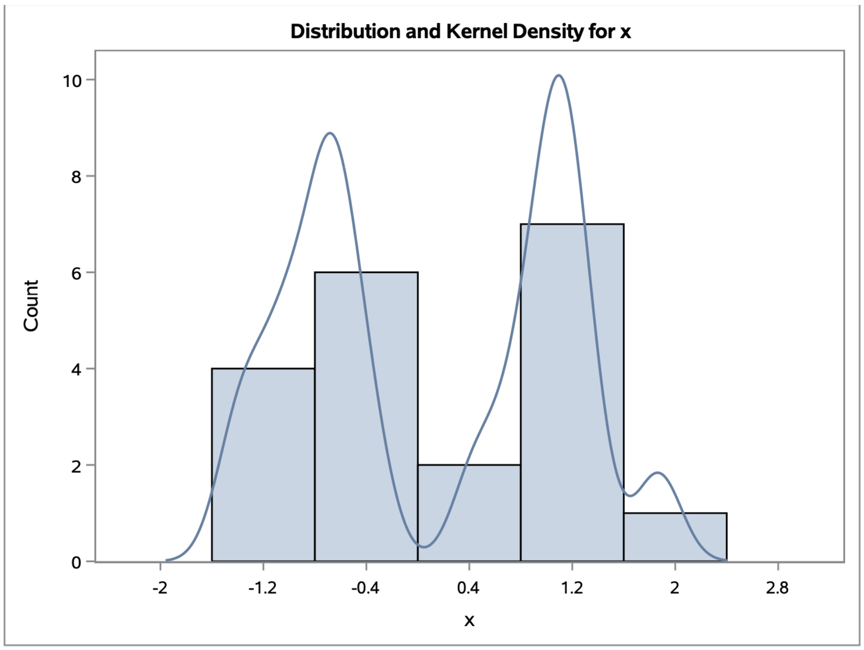
Controls	
	x
Grid Points	401
Lower Grid Limit	−1.601
Upper Grid Limit	2.0459
Bandwidth Multiplier	0.2



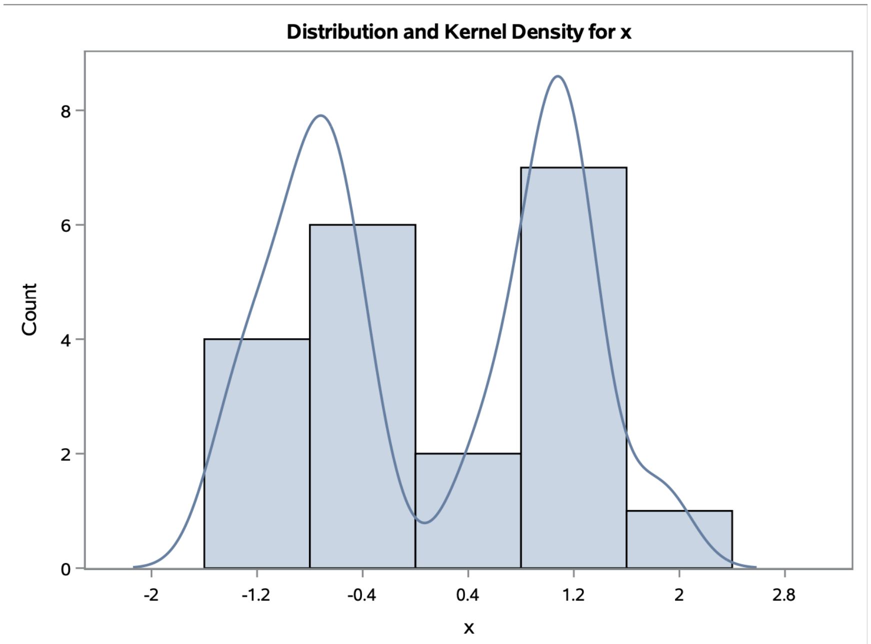
Controls	
	x
Grid Points	401
Lower Grid Limit	−1.601
Upper Grid Limit	2.0459
Bandwidth Multiplier	0.4



Controls	
	x
Grid Points	401
Lower Grid Limit	−1.601
Upper Grid Limit	2.0459
Bandwidth Multiplier	0.6

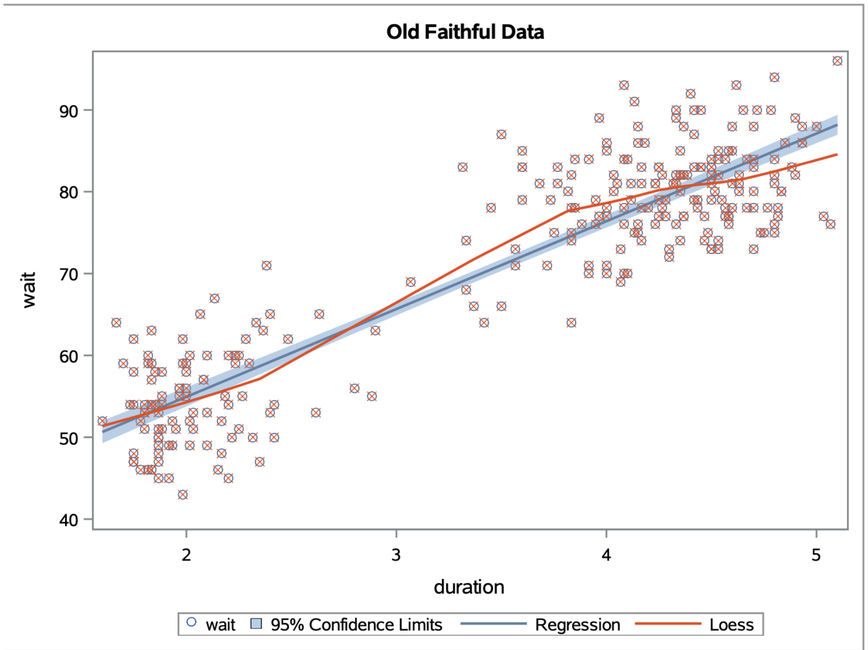


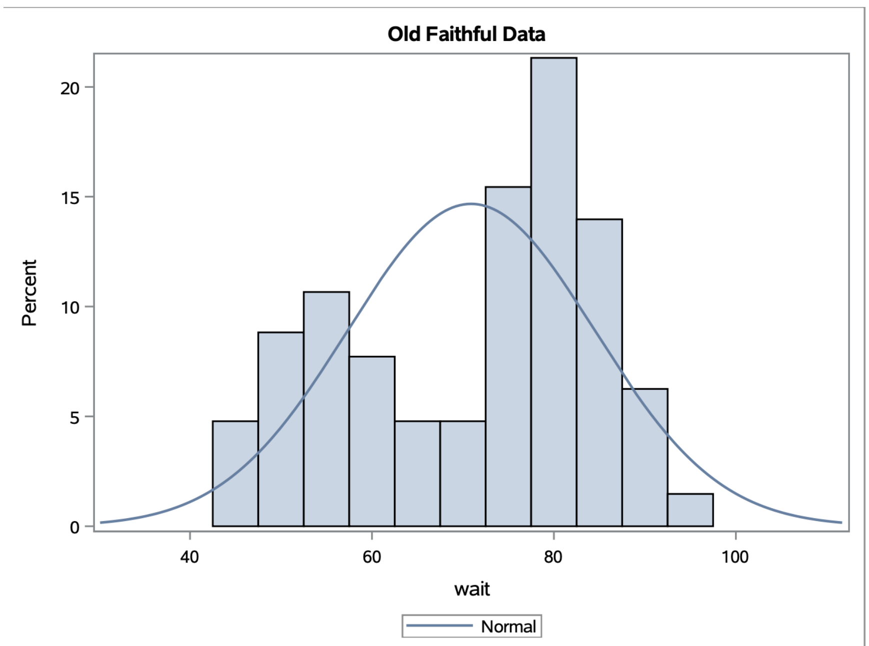
Controls	
	x
Grid Points	401
Lower Grid Limit	−1.601
Upper Grid Limit	2.0459
Bandwidth Multiplier	0.8

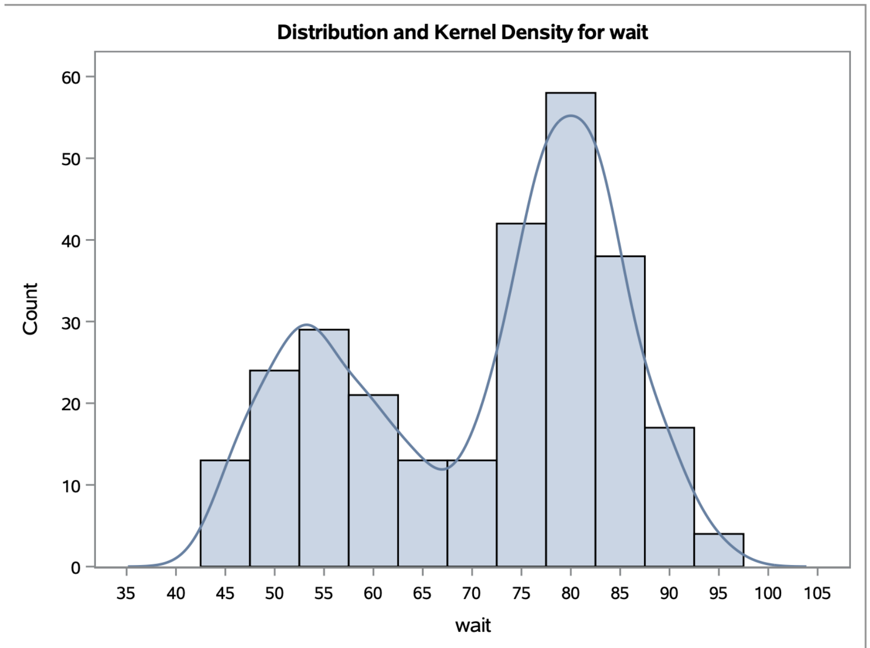


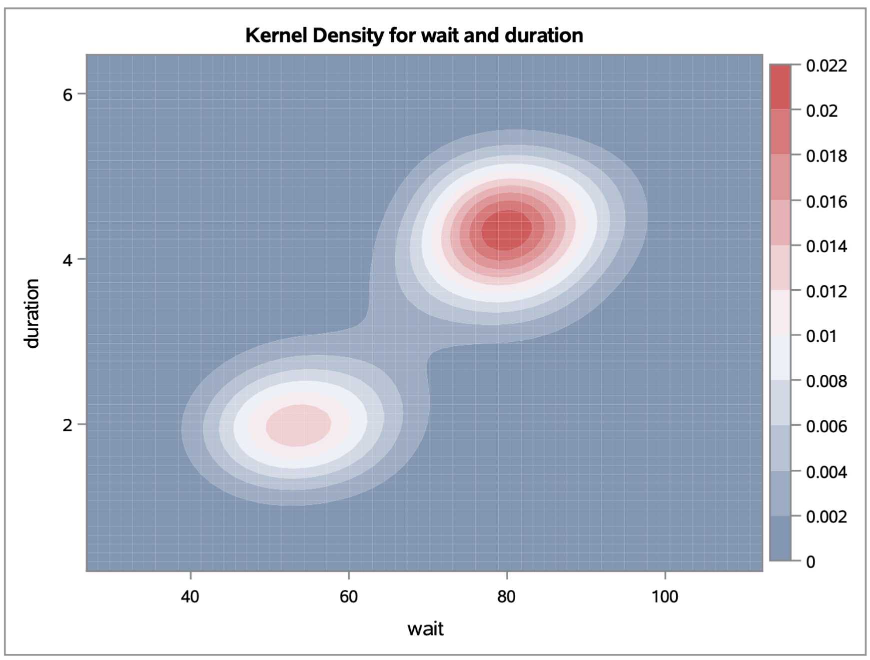
Old Faithful Data

The KDE Procedure






Distribution and Kernel Density for wait
Count
60
50
40
30
20
10
0
35
40
45
50
55
60
65
70
75
80
85
90
95
100
105
wait


Kernel Density for wait and duration
duration
6
4
2
40
60
80
100
wait
0.022
0.02
0.018
0.016
0.014
0.012
0.01
0.008
0.006
0.004
0.002
0