Kernel Density Estimation and Non-Parametric Regression

Patrick Mellady

This explanation will explore kernels, kernel density estimation, and non-parametric regression. We start with the definition of a kernel.

Kernels

A *kernel* is a function $k(x)$ such that $\int_{\mathbb{R}} k(x)dx = 1$, $k(x)$ is symmetric, $k(x) \geq 0$, and $\int_{\mathbb{R}} x^2 k(x)dx < \infty$. Given a kernel and a *bandidth*, h, we can define

$$k_h(x) = \frac{1}{h}k(\frac{x}{h})$$

If $k : \mathbb{R}^d \to \mathbb{R}$ and we have a *bandwidth matrix*, H, we define

$$k_H(x) = \frac{1}{|H|}k(H^{-1}x)$$

Frequently, we will use $H = hI$ and implement the following

$$k_H(x) = \frac{1}{h^d}\prod_{i=1}^{n} k(\frac{x_i}{h})$$

Kernel Density Estimation

Suppose we observe $Y_1, Y_2, \cdots, Y_n \sim P$ with an unknown density. The kernel density estimate for the density of P is given by

$$\hat{p}(y) = \frac{1}{n}\sum_{i=1}^{n} k_h(Y_i - y)$$

Non-Parametric Regression

We will implement the estimator of the form

$$\hat{r}(x) = \sum_{i=1}^{n} w_i(x)Y_i$$

where the weights, $w_i(x)$ are given by

$$w_i(x) = \frac{k_h(X_i - x)}{\sum_{j=1}^{n} k_h(X_j - x)}$$

Implementation

We now implement the above methods to find kernel density estimates and non-paramteric regression methods in a variety of examples. First, in the single dimension case, we implement general functions for calculating the functions above.

```r
k<-function(x, type="gaussian"){
  if(type=="gaussian"){
    temp<-dnorm(x)
  }
  if(type=="logistic"){
    temp<-(2/pi)*exp(x)/(1+exp(2*x))
  }
  if(type=="epanechnikov"){
    temp<-.75*(1-x^2)
  }
  return(temp)
}

# Non-parametric regression
w<-function(x, h, X, type){
  num<-k((X-x)/h, type)/h
  denom<-sum(k((X-x)/h, type)/h)
  temp<-num/denom
  return(temp)
}

r_x<-function(x, h, X, Y, type){
  weight<-w(x, h, X, type)
  temp<-sum(weight*Y)
  return(temp)
}


r<-function(X, Y, n=100, h=1/10, type="gaussian"){
  n<-length(X)
  x<-seq(min(X), max(X), length.out=n)
  y_hat<-c()
  for(point in x){
    y_hat<-c(y_hat, r_x(point, h, X, Y, type))
  }
  return(list(x=x, y_hat=y_hat))
}

# Density estimation
p_x<-function(x, h, X, type){
  n<-length(X)
  temp<-sum(k((x-X)/h, type)/h)/n
}

p<-function(X, n=1000, h=1/10, type="gaussian"){
  x<-seq(-10*max(abs(X)),10*max(abs(X)), length.out=n)
  p_hat<-c()
  for(point in x){
    p_hat<-c(p_hat, p_x(point, h, X, type))
  }
  return(list(x=x, p_hat=p_hat))
}
```

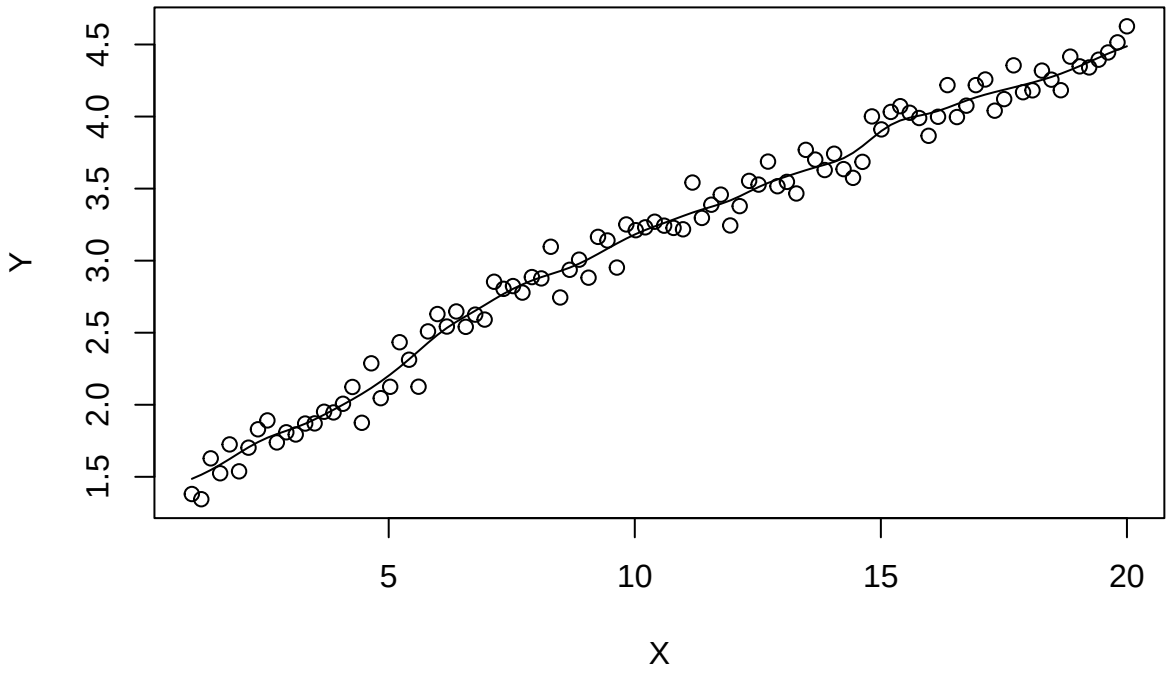
First Example

With the above functions in place, we examine some examples. The first example comes from simple normal data.

```
# Normal Example
## Non-parametric regression
n<-100
X<-seq(1, 20, length.out=n)
Y<-sqrt(X)+X*exp(-X)*sin(X)+rnorm(n, 0, .1)
```

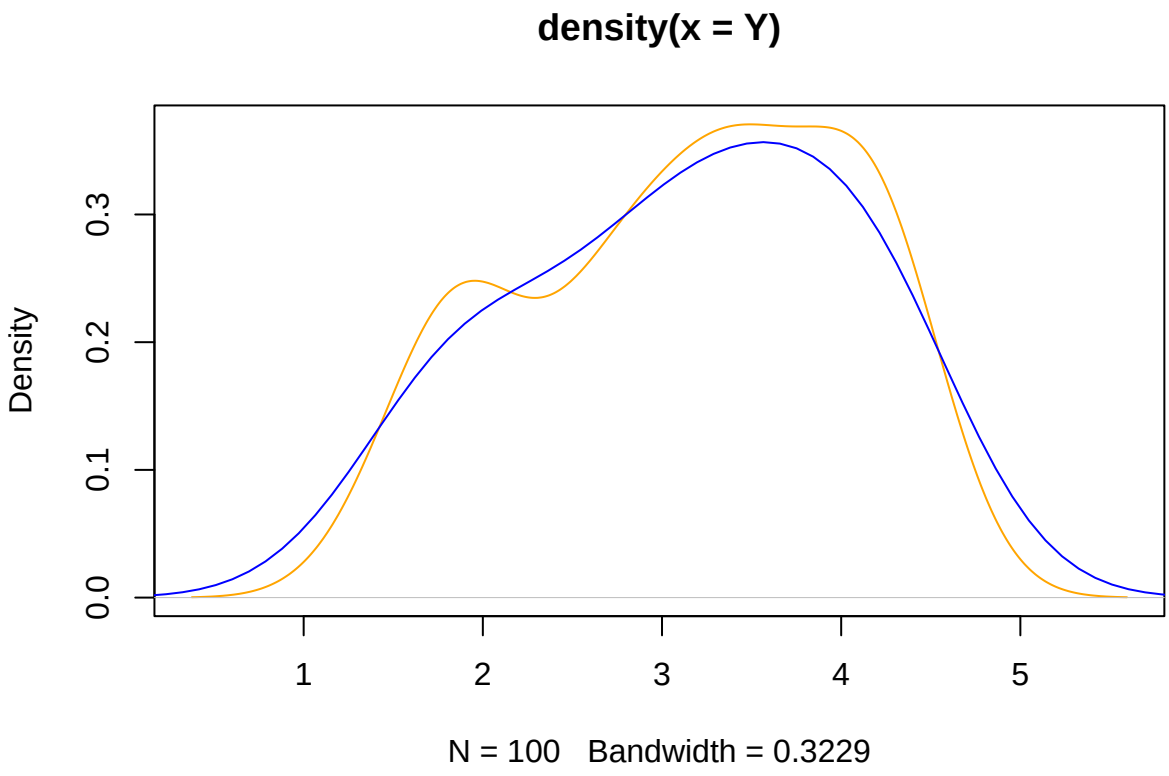
The non-parametric regression is

```
Y_hat<-r(X, Y, h=1/2)
plot(X, Y)
points(Y_hat$x, Y_hat$y_hat, type="l")
```



The kernel density estimate of the Y_is is

```
## KDE
p_hat<-p(Y, h=1/2)
plot(density(Y), col="orange")
points(p_hat$x, p_hat$p_hat, type="l", col="blue")
```



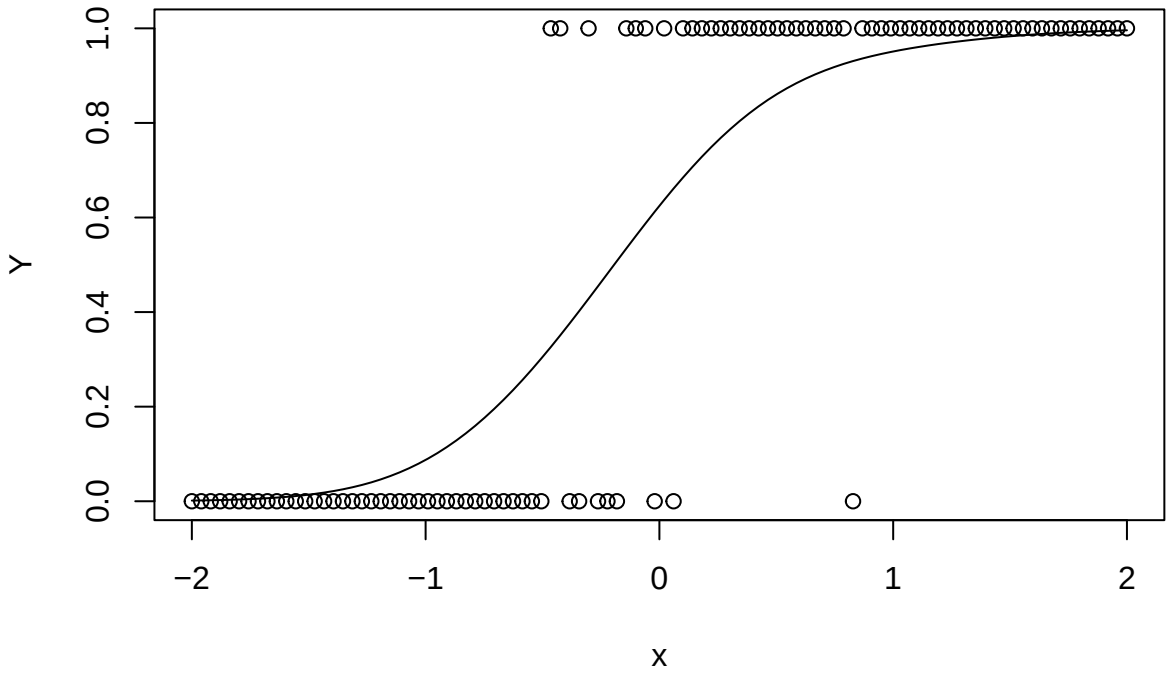
Second Example

We now try an example of logistically distributed data.

```
# Logistic Example
## Non-parametric regression
n<-100
h<-1/10
b<-5.01
x<-seq(-2, 2, length.out=n)
pi<-exp(x*b)/(1+exp(x*b))
Y<-c()
for(i in 1:n){
  Y<-c(Y, rbinom(1,1,pi[i]))
}
```

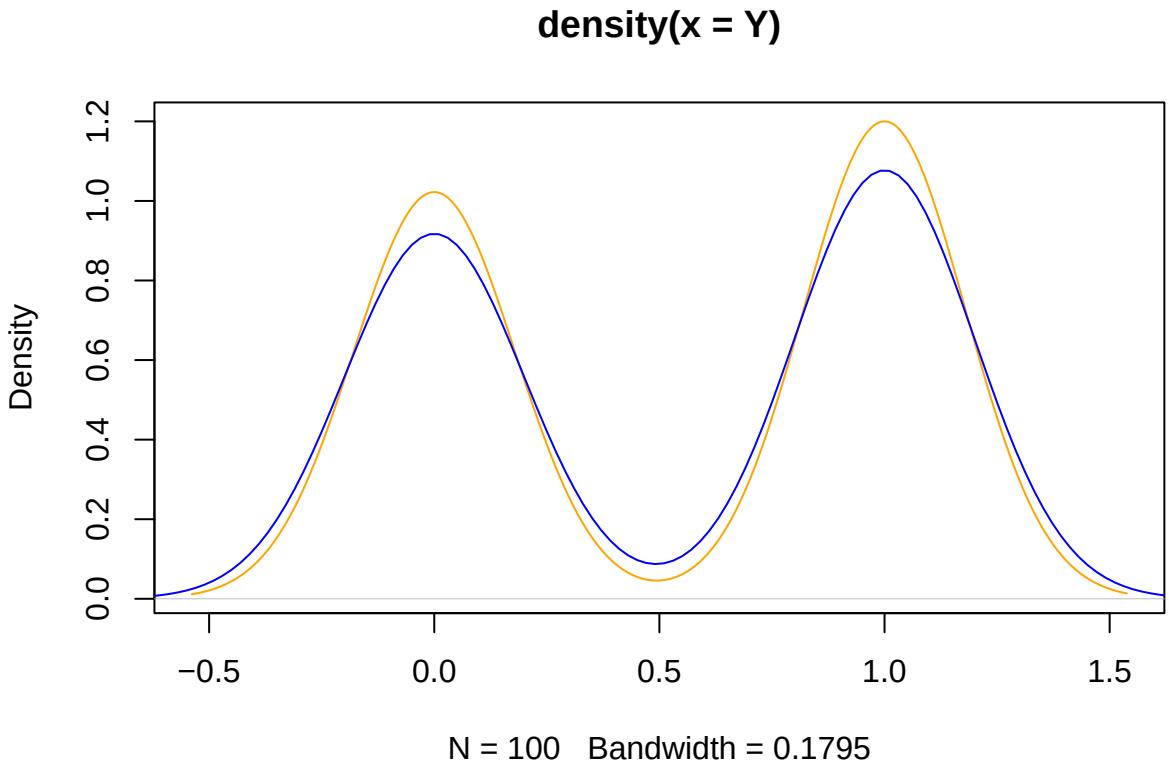
The non-parametric regression is

```
Y_hat<-r(x, Y, h=1/2)
plot(x, Y)
points(Y_hat$x, Y_hat$y_hat, type="l")
```



and the corresponding density estimate of the Y_is is

```
## KDE
p_hat<-p(Y, h=1/5)
plot(density(Y), col="orange")
points(p_hat$x, p_hat$p_hat, type="l", col="blue")
```



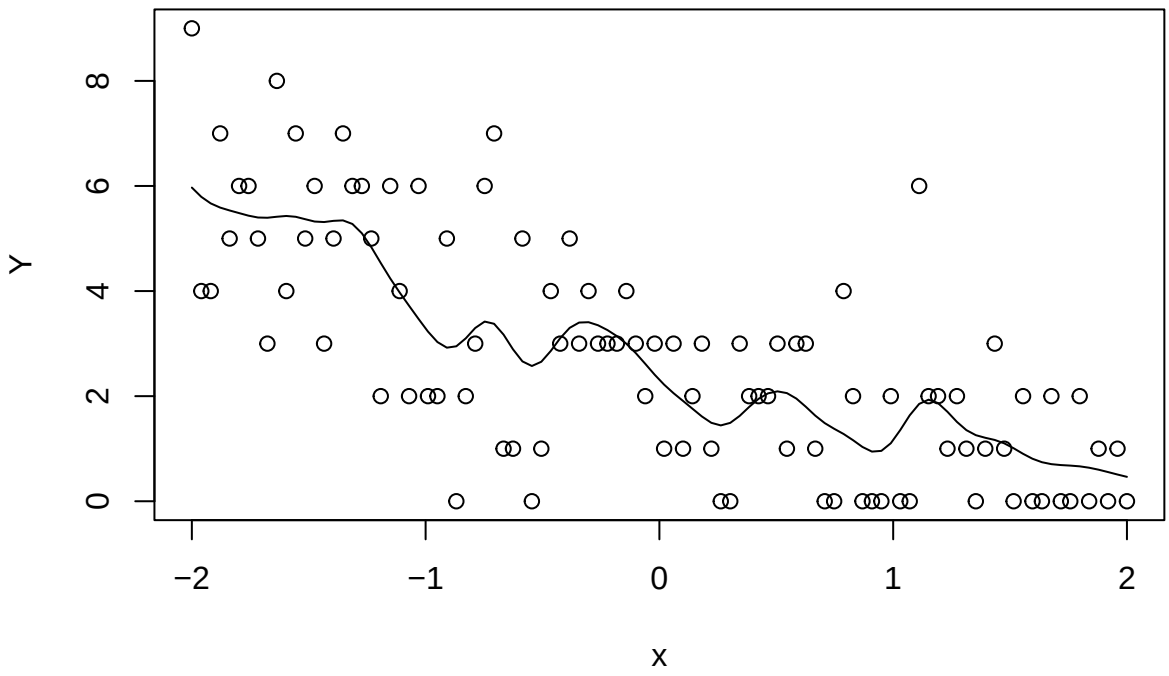
Example 3

In this example, we look at Poisson distributed data.

```
# Poisson Example
## Non-parametric regression
n<-100
x<-seq(-2,2,length.out=n)
X<-matrix(c(rep(1,n), x),ncol=2)
b<-c(.922,-.501)
lam<-exp(X%*%b)
Y<-c()
for(i in 1:n){
  Y<-c(Y, rpois(1, lam[i]))
}
```

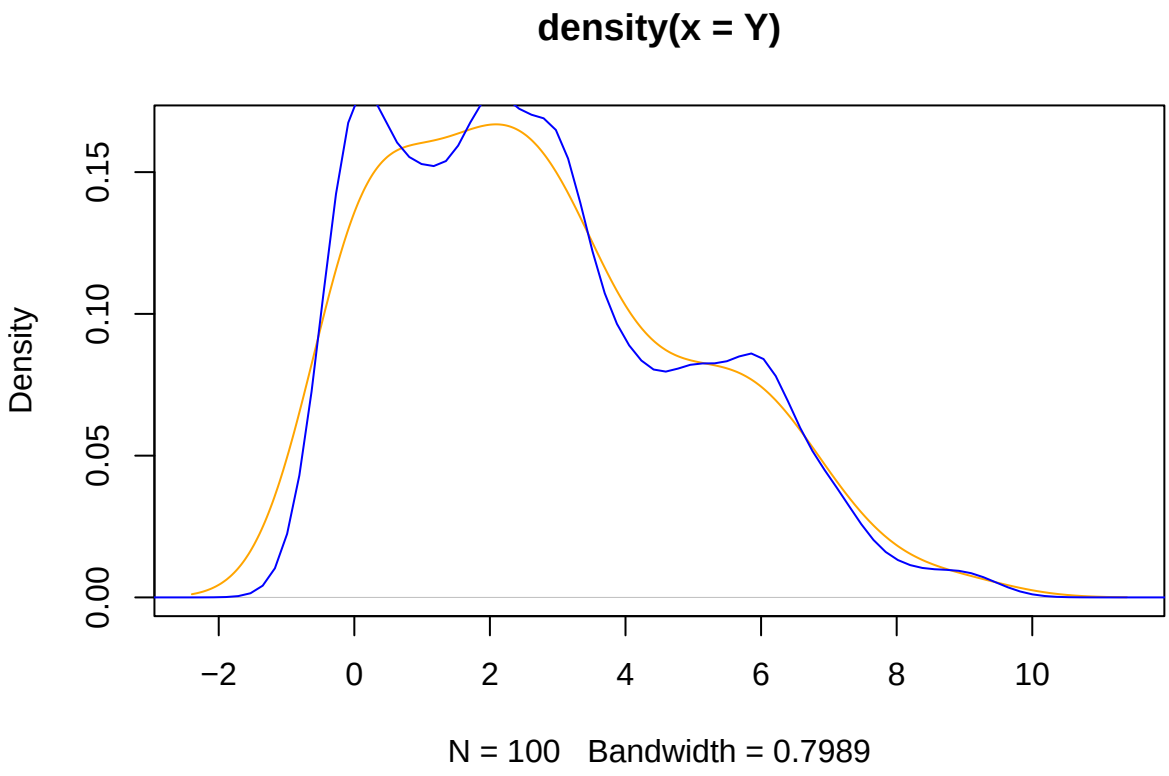
The regression is

```
# Non-Parametric regression
Y_hat<-r(x, Y)
plot(x, Y)
points(Y_hat$x, Y_hat$y_hat, type="l")
```



and the density of the Y_is is

```
# KDE
p_hat<-p(Y, h=1/2)
plot(density(Y), col="orange")
points(p_hat$x, p_hat$p_hat, type="l", col="blue")
```



A Real Data Example

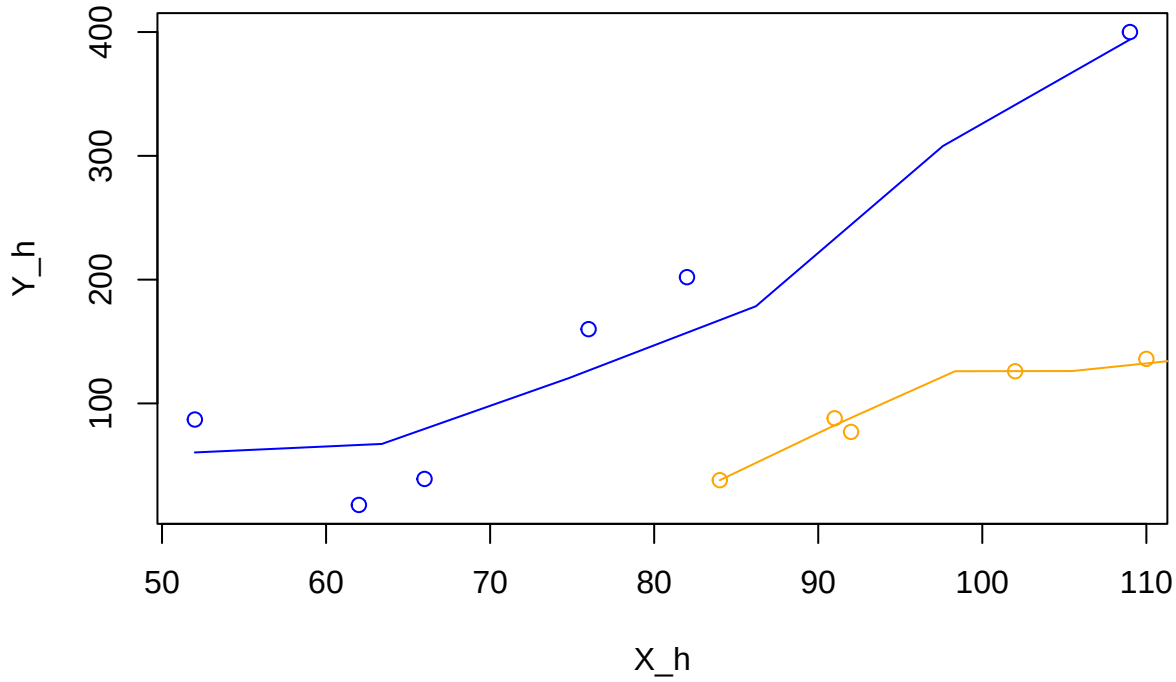
In this example, we fit non-parametric regression to data of alcohol consumption at events run by a certain company. We start with importing and partitioning the data.

```
# Alcohol Example
## Read data
mvalc <- read.csv("alc_data.csv", header=T)

## Tidy Data
rownames(mvalc) <- mvalc$Wedding
mvalc<-t(mvalc[,-1])

## Define Variables
X<-mvalc[,1]
X2<-X^2

Y<-mvalc[,62]
Y_w<-mvalc[,63]

high<-mvalc[c(2,6,7,9,12,13),c(1,62)]
low<-mvalc[-c(2,6,7,9,12,13), c(1,62)]

Y_h<-high[,2]
X_h<-high[,1]
X_h2<-X_h^2
```

```
Y_l<-low[,2]
X_l<-low[,1]
X_l2<-X_l^2
```

We now fit a non-parametric regression and plot

```
## Non-parametric regression
x_h<-seq(min(X_h), max(X_h), length.out=10*length(X_h))
Y_h_hat<-r(X_h, Y_h, h=10)
plot(X_h, Y_h, col="blue")
points(Y_h_hat$x, Y_h_hat$y_hat, type="l", col="blue")

x_l<-seq(min(X_l), max(X_l), length.out=10*length(X_l))
Y_l_hat<-r(X_l, Y_l, h=1)
points(X_l, Y_l, col="orange")
points(Y_l_hat$x, Y_l_hat$y_hat, type="l", col="orange")
```



Multivariate Example

We implement the non-parametric regression method below

```
r<-function(X, Y, intercept, n=100, h=1/10, type="gaussian"){
  if(is.matrix(X)==FALSE){
    x<-seq(min(X), max(X), length.out=n)
    y_hat<-c()
    for(point in x){
      y_hat<-c(y_hat, r_x(point, h, X, Y, type))
```

```
    }
  }
  if(is.matrix(X)==TRUE){
    if(intercept==TRUE){
      d<-ncol(X)-1
      x<-matrix(rep(0,n*d), ncol=d)
      for(j in 1:d){
        x[,j]<-seq(min(X[,j+1]), max(X[,j+1]), length.out=n)
      }
      X<-X[,-1]
    }
    if(intercept==FALSE){
      d<-ncol(X)
      x<-matrix(rep(0,n*d), ncol=d)
      for(j in 1:d){
        x[,j]<-seq(min(X[,j]), max(X[,j]), length.out=n)
      }
    }
    y_hat<-c()
    for(i in 1:n){
      point<-x[i,]
      num<-c()
      for(l in 1:nrow(X)){
        num<-c(num, prod(k((X[l,]-point)/h, type)/h))
      }
      denom<-sum(num)
      W<-num/denom
      y_hat<-c(y_hat,sum(W*Y))
    }

  }
  return(list(x=x, y_hat=y_hat))
}
```

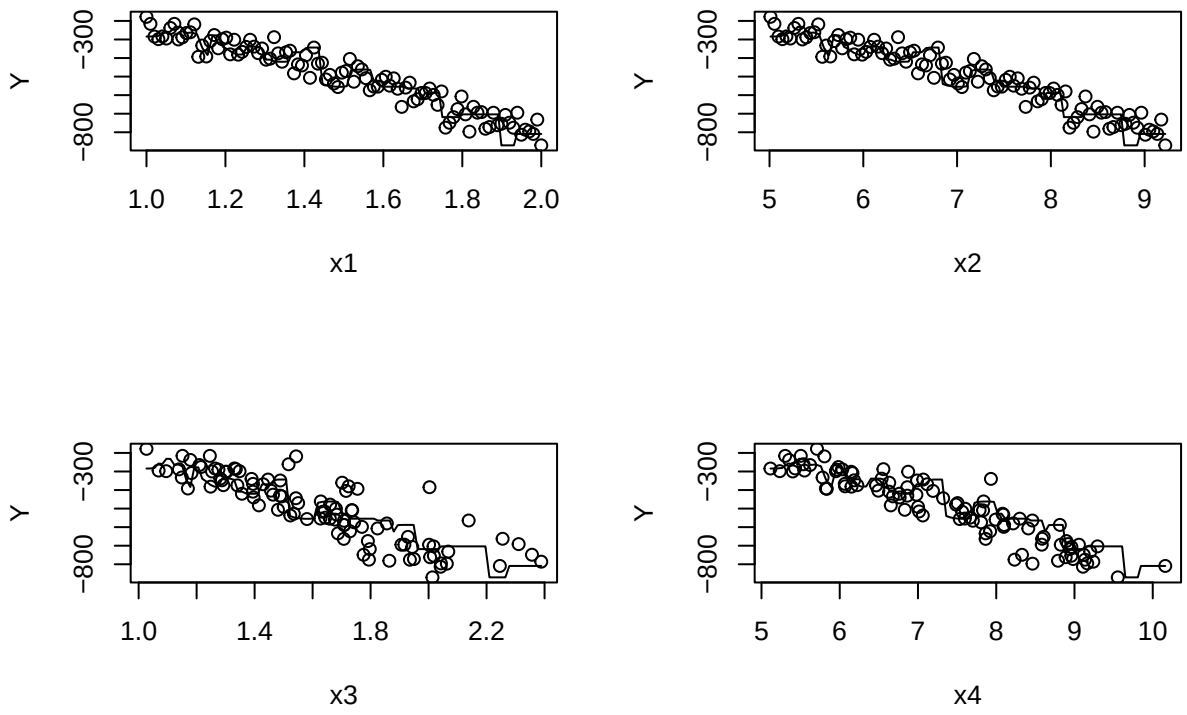
Using this, we generate some normal data and show and examine the non-parametric fit

```
# Normal Example
## Non-parametric regression
n<-100
b<-matrix(c(5.01, 9.22, -12.22, 1, 2), ncol=1)
x1<-seq(1, 2, length.out=n)
x2<-seq(5.01,9.22,length.out=n)
x3<-x1+rgamma(n,1,8)
x4<-x2+rgamma(n,1,4)
X<-matrix(c(rep(1,n),x1^2, x2^2, x3, x4^2), ncol=5)
Y<-X%*%b+rnorm(n, 0, 50)

Y_hat<-r(X, Y, intercept=TRUE, h=c(.75, .5), n=100)
```

To examine the fit, we plot the profile of the non-parametric regression

```
par(mfrow=c(2,2))
plot(x1, Y)
points(sqrt(Y_hat$x[,1]), Y_hat$y_hat, type="l")
plot(x2,Y)
```

```
points(sqrt(Y_hat$x[,2]), Y_hat$y_hat, type="l")
plot(x3, Y)
points(Y_hat$x[,3], Y_hat$y_hat, type="l")
plot(x4,Y)
points(sqrt(Y_hat$x[,4]), Y_hat$y_hat, type="l")
```



Further, we can examine how Poisson distributed data is fit by a non-parametric regression

```
# Poisson Example
## Non-parametric regression
n<-100
b<-matrix(c(5.01, 9.22, 12.22), ncol=1)
x1<-seq(1, 2, length.out=n)
x2<-seq(5.01,9.22,length.out=n)
X<-matrix(c(rep(1,n),x1^2, x2), ncol=3)
lam<-exp(X%*%b)
Y<-c()
for(i in 1:n){
  Y<-c(Y, rpois(1, lam[i]))
}

Y_hat<-r(X, Y, intercept=TRUE, h=c(.75, .5), n=100)
```

We now plot the profile of the non-parametric regression

```
par(mfrow=c(1,2))
plot(x1, Y)
```

```
points(sqrt(Y_hat$x[,1]), Y_hat$y_hat, type="l")
plot(x2,Y)
points(Y_hat$x[,2], Y_hat$y_hat, type="l")
```