Lab 7: k-Nearest Neighbors Regression

Stat 154, Fall 2019

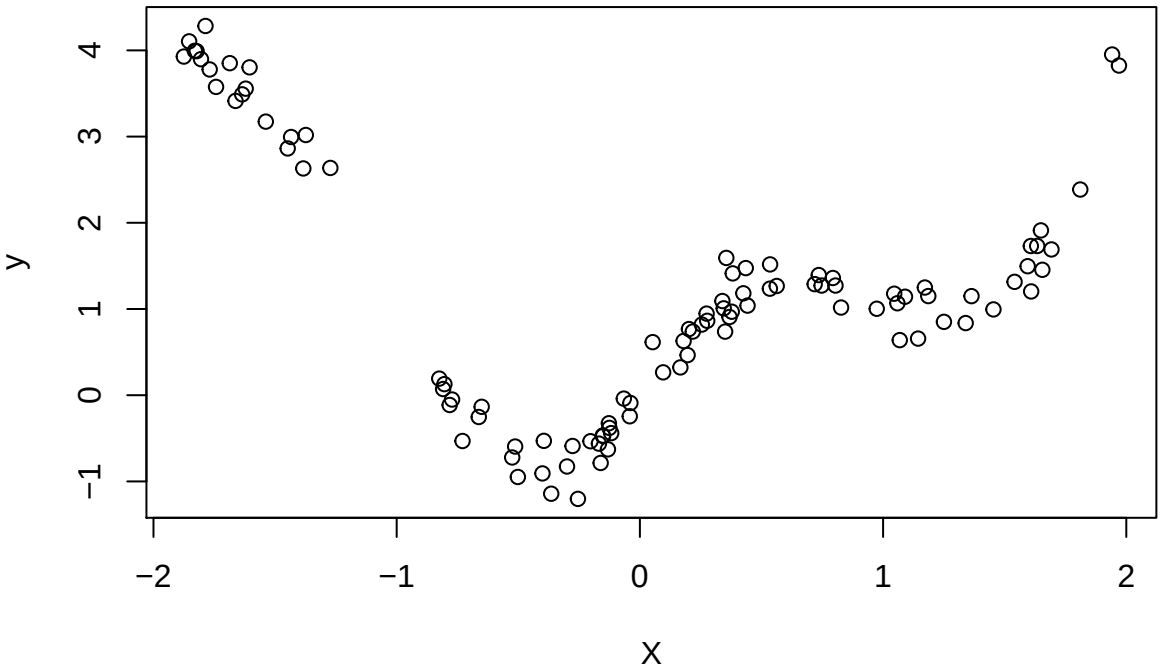
So far, we have considered only *parametric* regression methods, and more specifically, ones that have assumed that the true function f is approximately linear. In this lab, we will investigate perhaps the simplest *nonparametric* regression method: k-Nearest Neightbors regression.

Part 1: Implementing kNN Regression

In this part, we will implement kNN regression, and test it on the following synthetic dataset.

```
X <- runif(-2,2,n=100)
f <- function(x){ return(sin(pi*x) +x^2)}
y <- f(X) + rnorm(length(X), sd=.2)

plot(X, y)
```



1.1

Implement a function `kNNR(z,k)` that finds the set of indices $\{i_1, ..., i_k\}$ of the k nearest points to z, and returns $\frac{1}{k}\sum_{j=1}^{k} y_{i_j}$. Plot the function kNNR for $k = 5$ over a reasonable range of values of z values.

1.2

Consider the following test set of 50 points.

```
Xtest <- runif(-2,2,n=50)
ytest <- f(Xtest) + rnorm(length(Xtest), sd=.2)
```

Compute the out of sample MSE for k ranging from 1 to 100. Which value of k makes the MSE smallest?

Part 2: Comparison to Linear Regression

In this part, we will compare kNN regression to linear regression.

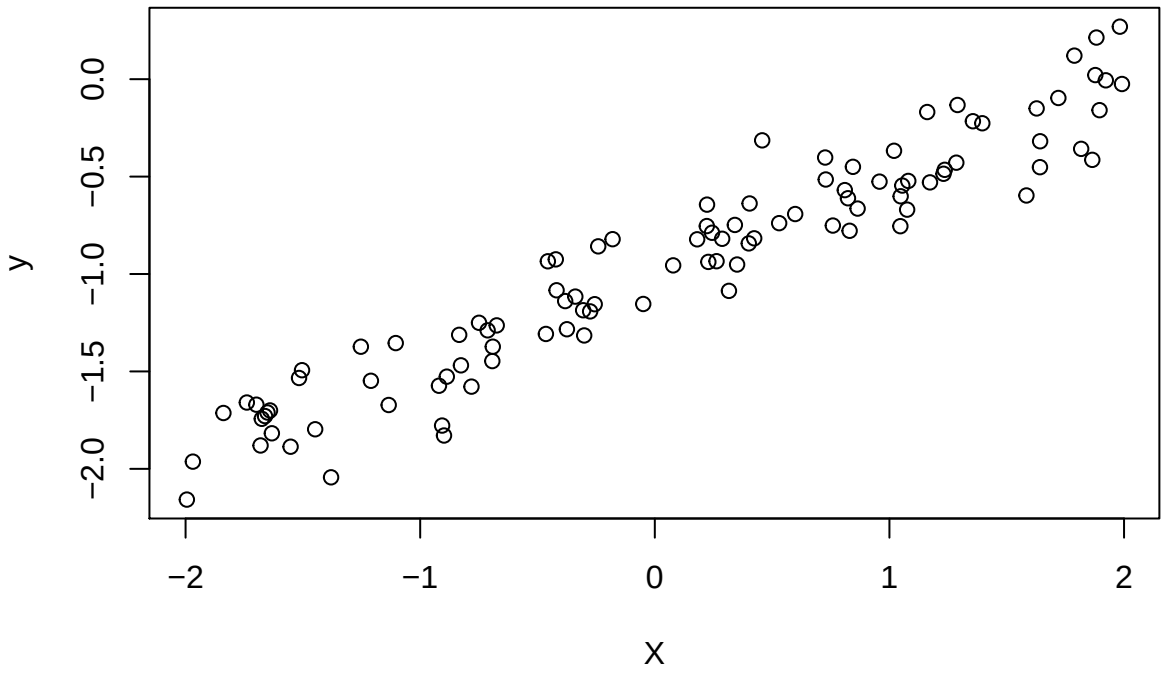
2.1 (linearly generated data)

First, consider the following data, generated from a linear system $y = X\beta + \epsilon$, with $\epsilon \sim N(0, .04)$:

```
X <- runif(-2,2,n=100)
f <- function(x){ return(.5*x -1)}
y <- f(X) + rnorm(length(X), sd=.2)

Xtest <- runif(-2,2,n=100)
ytest <- f(Xtest) + rnorm(length(Xtest), sd=.2)

plot(X, y)
```



Compute the least squares coefficients, and compute the corresponding test MSE for the linear model.

Now for k ranging from 1 to 100, compute the test MSE for the k-Nearest Neighbors model. What is the best value of k? How do the two models (linear regression and kNN regression) compare in terms of MSE?

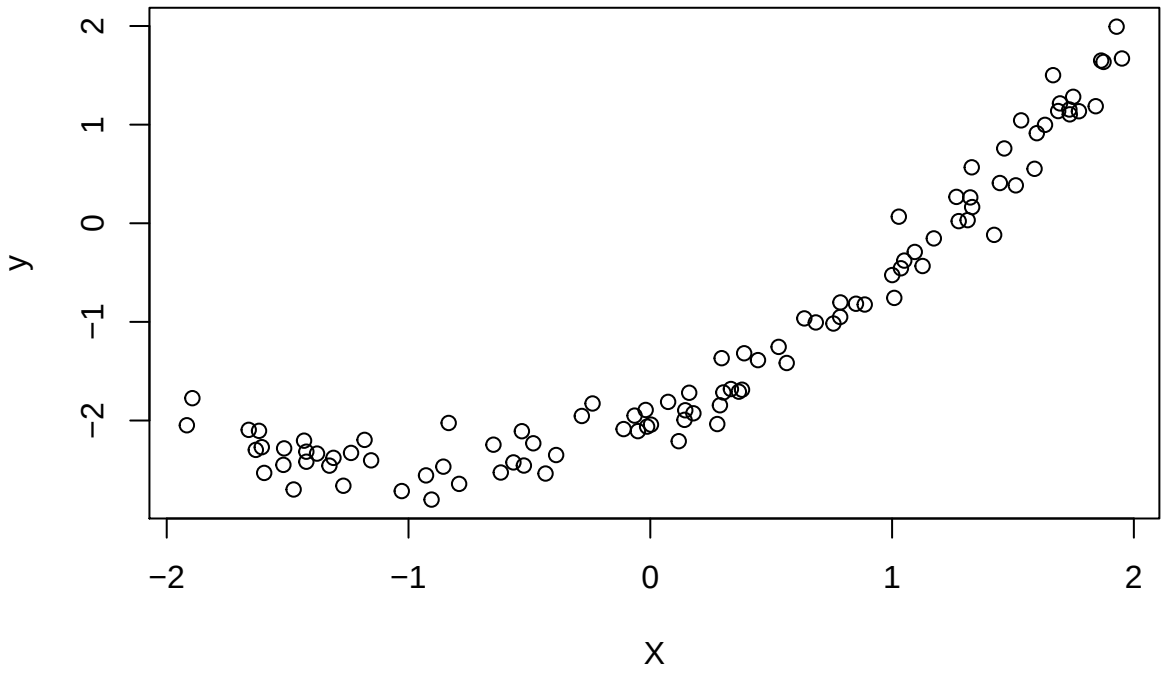
2.2 (non-linearly generated data)

Repeat the steps of the previous problem, but now with the following data:

```
X <- runif(-2,2,n=100)
f <- function(x){ return(.5*x^2 + x -2)}
y <- f(X) + rnorm(length(X), sd=.2)

Xtest <- runif(-2,2,n=100)
ytest <- f(Xtest) + rnorm(length(Xtest), sd=.2)

plot(X, y)
```



Which method performs better now?