Arif Ali
21922601
Stat 154
02/21/14
Section 101

Homework 2

Question 1

The first analysis conducted was hierarchical clustering based on the three political parties that were present in the 2005 Congress, Republican, Democrat, and Independent. The clustering first reveals that there were only three political parties to analyze. Figure 1 reveals to expectation that the value of the votes was held by one of the three parties, the Republican in the case of 2005. The value of Democratic and Independent votes was at the same level. Therefore, the Republicans had a stronger influence on voting during this time period. This is consistent with the notation that the Republican must have been the majority party at the time.

The Second Analysis conducted was the PCA of the voting data as shown in figure 2. This revealed the same conclusions as the first analysis. The Republicans were of far greater influence compared to wither the Democrats or Independent. However, the interesting part of this representation was the spread of the Republicans vs. the Democrats. The Republican Party seemed to be clustered in a smaller area, which indicates strong record of consistency with regards to voting. There are noticeable outliers in this plot with respect to Republicans indicating some differences in voting record but not as diverse when compared to the Democrats. It seems that the minority of the Democrats has slightly more influence on voting outcomes compared to the majority.

The Third analysis was the use of Kernel PCA (figure 3) which helped to reveal some insight regarding the majority of Democrats vs. the minority and is further backed up by analysis of Spectral clustering (figure 4). The Minority described in the second analysis has an exact number of 72 Democrats. This indicates that this group was more influential with regards to voting outcomes compared to the rest of the Democrats in the House of Representatives.

Question 2

Part 1

In order to plot the three different Regression functions, the first step was to ensure that x_i was properly computed. Using a fixed N, i was created as a vector from 1 to 10 and then applied to the formula for $x_i = (i - 1/2)/N * 2 * \pi$. Using these x_i values, a response vector was created a called f which was $f = \cos(10 * x) + 2$. Later a separate vector was designed using the normal distribution $N\ (0, 1)$. When using the K-NN regression function, only the prediction was kept into account along with the x and y coordinates. The test vector was composed of $2\pi/100$, which were equally distribution 100 times.

Figure 5 depicts the graph when analyzing the predicted outcomes when k =1. The graphs show distinct slops between every few clusters of points. This is attributed to each point relying on the single closest neighbor as opposed to multiple neighbors. Figure 6 depicts the regression when k = 3, which shows less sloping because to dramatic slopes that were shown in figure 5 are determined by the three nearest neighbors which may reduce the differences in prediction, thus reducing the overall slopes

Figure 7 shows the other extreme when compared to figure 6 and figure 5. The predictions for all the points are shown to be the same or marginally different. This is because, k =10 which would mean that each points is determined by the 10 nearest neighbors, or 10% of x_i. This significantly alters the predictions because the magnitude of k renders the use of the k nearest neighbor algorithm as obsolete.

Part 1.1

In order to calculate $EPE\ (\pi)$, I used the formula $EPE_k(x_0) = \sigma^2 + [\text{Bias}^2\left(\hat{\text{f}}(\text{x}_0)\right) + \text{Var}_T\left(\hat{f}_k(\text{x}_0\ )\right)] = \sigma^2 + [f\ (x_0)\ - \frac{1}{k}\sum_1^k\ f(x_\ell)]^2 + \frac{\sigma^2}{k}$ from page 37 in "Elements of Statistical Learning". The lowest $EPE\ (\pi)$ occurred when k =1. This would be attributed to the fact that we only analyzed one x.

Calculated the $\boldsymbol{E}\ (EPE(X))$ required analyzing each variable of X, which was given to be a uniform distribution of $(0,2\pi)$ For each x in the Uniform distribution, the equation used above was applied given the k value. Unsurprisingly, the lowest $\boldsymbol{E}\ (EPE(X))$ occurred when k =10. However, the Bias was greater.

$EPE\ (\pi)$	$\boldsymbol{E}\ (EPE(X))$ (Note: for Bias^2 and Var, the mean was taken, which may affect results
K=1 Bias^2 Var EPE 35.91725 1 37.91725	K =1 Bias^2 Var EPE 78.59098 1.00000 80.59098
K=3 Bias^2 Var EPE	K =3 Bias^2 Var EPE

9.669802 0.3333333 11.00314	3.6156218 0.3333333 4.9489551
K=10 Bias^2 Var EPE 3.823453 0.1 4.923453	K = 10 Bias^2 Var EPE 3.543937 0.100000 4.643937

Part 1.2

Constant function (note Bias indicates Bias^2)

$EPE\ (\pi)$	$\boldsymbol{E}\ (EPE(X))$ (Note: for Bias^2 and Var, the mean was taken, which may affect results
K=1 Bias Var EPE 180.8864 1 182.8864	Bias Var EPE 67.27978 1.00000 69.27978
K=3 Bias Var EPE 1.559431 0.3333333 2.892764	Bias Var EPE 6.2338808 0.3333333 7.5672141
K=10 Bias Var EPE 0.03531186 0.1 1.135312	Bias Var EPE 2.292326 0.100000 3.392326

Linear function (note Bias indicates Bias^2)

$EPE\ (\pi)$	$\boldsymbol{E}\ (EPE(X))$ (Note: for Bias^2 and Var, the mean was taken, which may affect results
K=1 Bias Var EPE 448.8684 1 450.8684	Bias Var EPE 90.08286 1.00000 92.08286
K=3 Bias Var EPE 2.885933 0.3333333 4.219266	Bias Var EPE 5.0158042 0.3333333 6.3491375
K=10 Bias Var EPE 5.726577 0.1 6.826577	Bias Var EPE 1.403866 0.100000 2.503866

Quadratic function (note Bias indicates Bias^2)

$EPE\ (\pi)$	$\boldsymbol{E}\ (EPE(X))$ (Note: for Bias^2 and Var, the mean was taken, which may affect results
K=1 Bias Var EPE 12.34045 1 14.34045	Bias Var EPE 0.05672754 1.00000000 2.05672754
K=3 Bias Var EPE 0.3882076 0.3333333 1.721541	Bias Var EPE 0.02903191 0.33333333 1.36236525
K=10 Bias Var EPE 0.1999003 0.1 1.2999	Bias Var EPE 0.04394939 0.10000000 1.14394939

Part 1.3

For $EPE\ (\pi)$, the lower a polynomial is in power, the lower the EPE will be. However for $\boldsymbol{E}\ (EPE(X))$, the opposite is true, has the quadratic yields lower EPE means yet a higher bias. The linear equation is slightly higher in the value of $\boldsymbol{E}\ (EPE(X))$; however, it seems that the more points in the vector, the higher polynomial results in lower values of $\boldsymbol{E}\ (EPE(X))$.

Part 2.1

The first three observations for $EPE\ (\pi)$ were extremely high compared to the k =1.3.4 when N=10. However, when looking at k=20, 50, the bias and EPE were both reduced. This was also consistent when analyzing the polynomials.

	Bias (f0)	Var (f0)	EPE(f0)	Bias (x^2)	Var (x^2)	EPE(x^2)
K=1	85158.97	1	85160.97	43609.45	1	43611.45
K=3	6407.382	0.3333333	6408.716	4751.189	0.3333333	4752.522
K=10	277.4659	0.1	278.5659	294.345	0.1	295.445
K=20	48.59119	0.05	49.64119	67.68676	0.05	68.73676
K=50	1.407453	0.02	2.427453	4.789247	0.02	5.809247

The same should be true for polynomials with degree 5.

	Bias (x^5)	Var (x^5)	EPE(x^5)
K=1	0.023446	1	2.023446
K=3	0.01852797	0.3333333	1.351861
K=10	0.004422169	0.1	1.104422
K=20	0.0002435558	0.05	1.050244
K=50	0.0001039057	0.02	1.020104

When analyzing the E(EPE(X)), the following output is given

```
sapply(EPE.X.X(X, 1, y6form.X), mean)
        EPE        Bias         Var
2.001924304 0.001924304 1.000000000
sapply(EPE.X.X(X, 3, y6form.X), mean)
       EPE       Bias        Var
43.9155276 42.5821943  0.3333333
sapply(EPE.X.X(X, 10, y6form.X), mean)
     EPE     Bias      Var
70.40573 69.30573  0.10000
sapply(EPE.X.X(X, 20, y6form.X), mean)
     EPE     Bias      Var
83.28693 82.23693  0.05000
sapply(EPE.X.X(X, 50, y6form.X), mean)
```

EPE Bias Var
92.04681 91.02681 0.02000

Part 2.2

Based on the formula given by Part 1.1, the if $\sigma^2 > 1$, EPE would increase by $\frac{(\sigma^2-1)(k+1)}{k}$ $when\ \sigma^2 > 1$

Question 3

When analyzing smaller values, the ability for Bias and EPE to be found is found to surprising contain integers as is the case when k =1. However, since the sample contains only 5 since p =5, as k approaches 5, the EPE lowers, values also tend to lower.

K=1

Bias	Var	Var
0	1	2
0	1	2
0	1	2
0	1	2
0	1	2

.

K=4

Bias	Var	Var
0.014208916	0.25	1.264209
0.001989070	0.25	1.251989
0.112938431	0.25	1.362938
0.002472867	0.25	1.252473
0.045594241	0.25	1.252473

Question 4

The two formulas given signify the standard deviation; therefore, upper and lower bounds that any of $\left|\left|X_{i0} - X_j\right|\right|$. Since this was normally constructed i.i.d N(0,1), it should be distributed as normal. This is observed in figure 8.

Appendix (Graphs and Code)

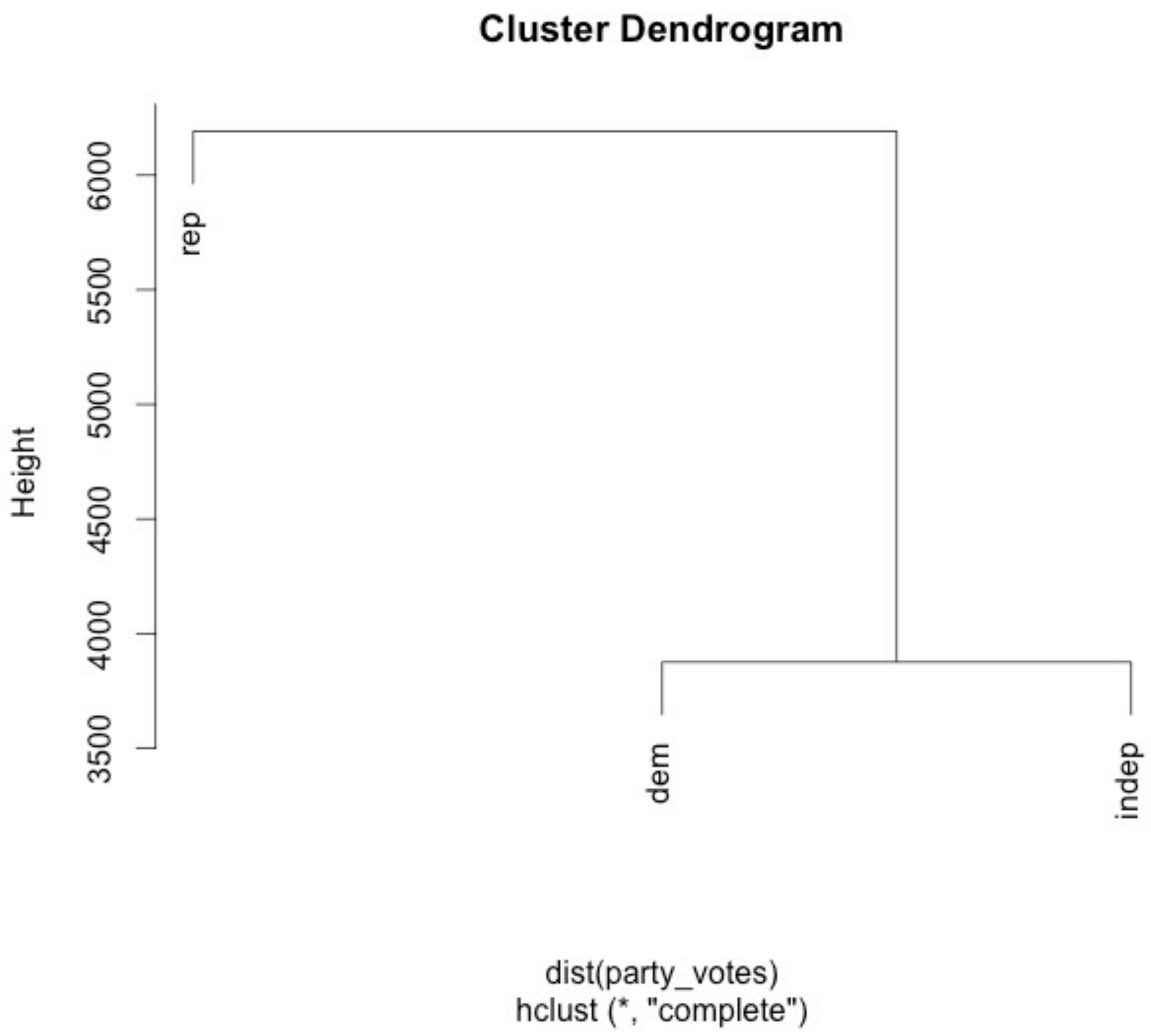

Figure 1

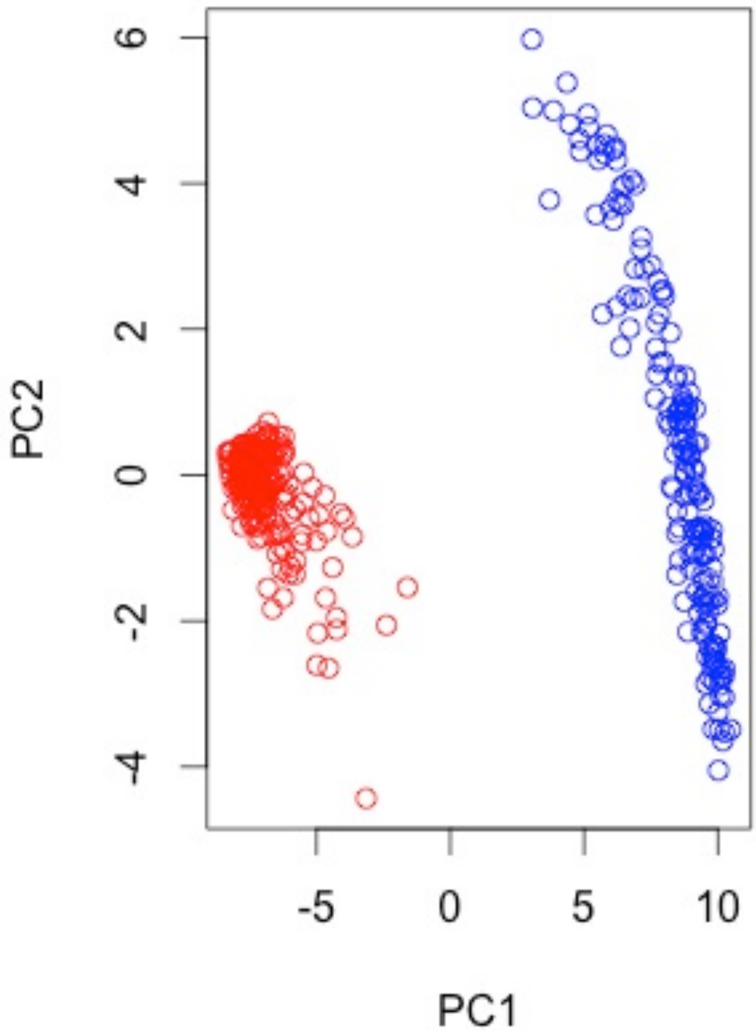

Figure 2

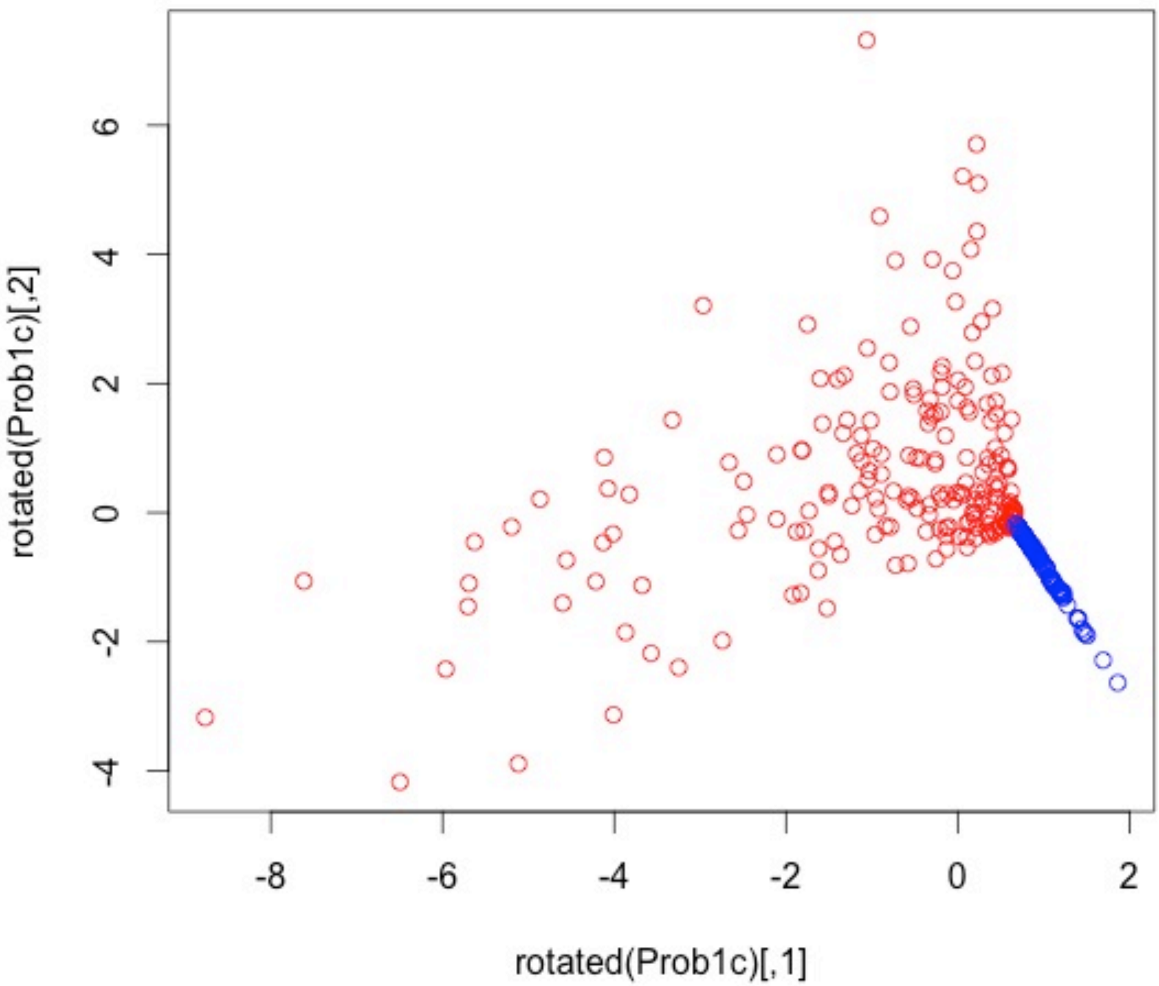

Figure 3

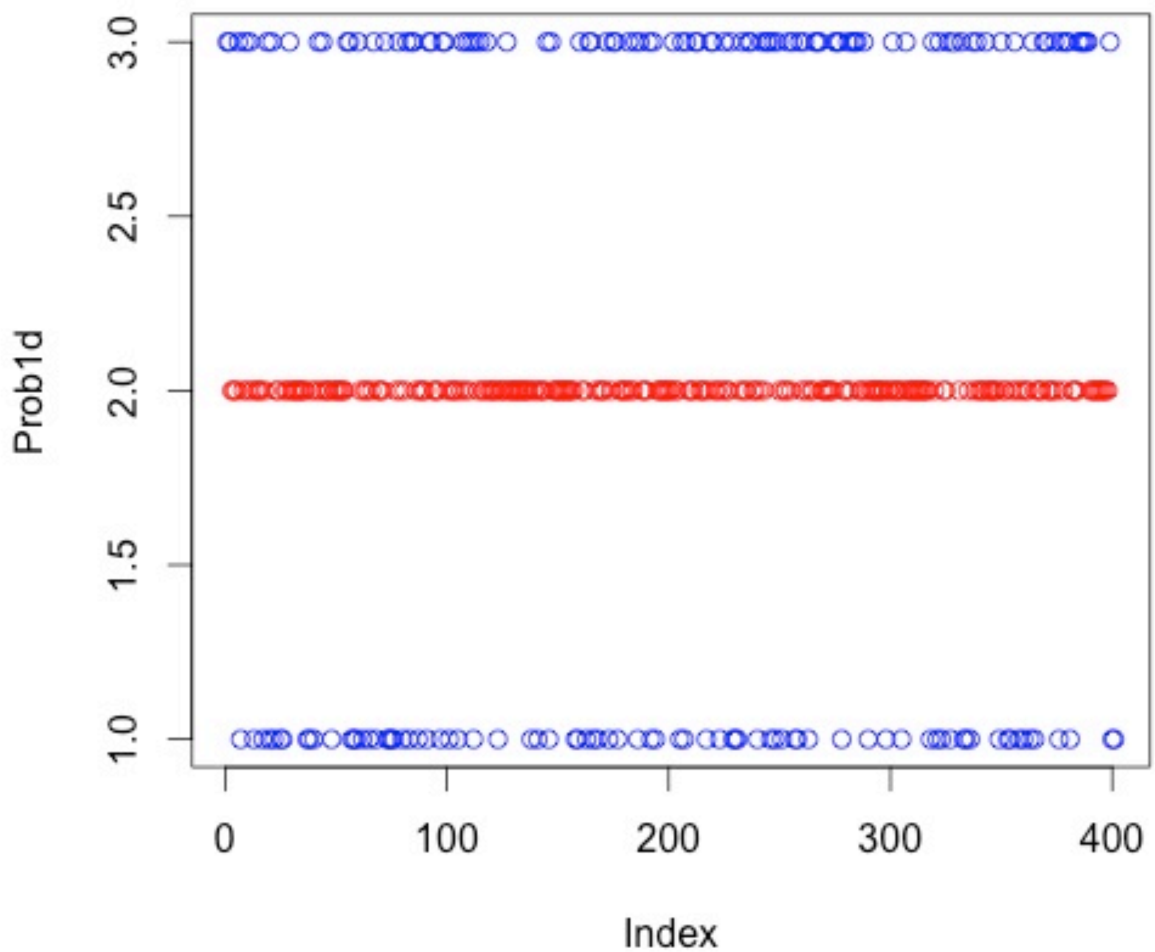

Figure 4

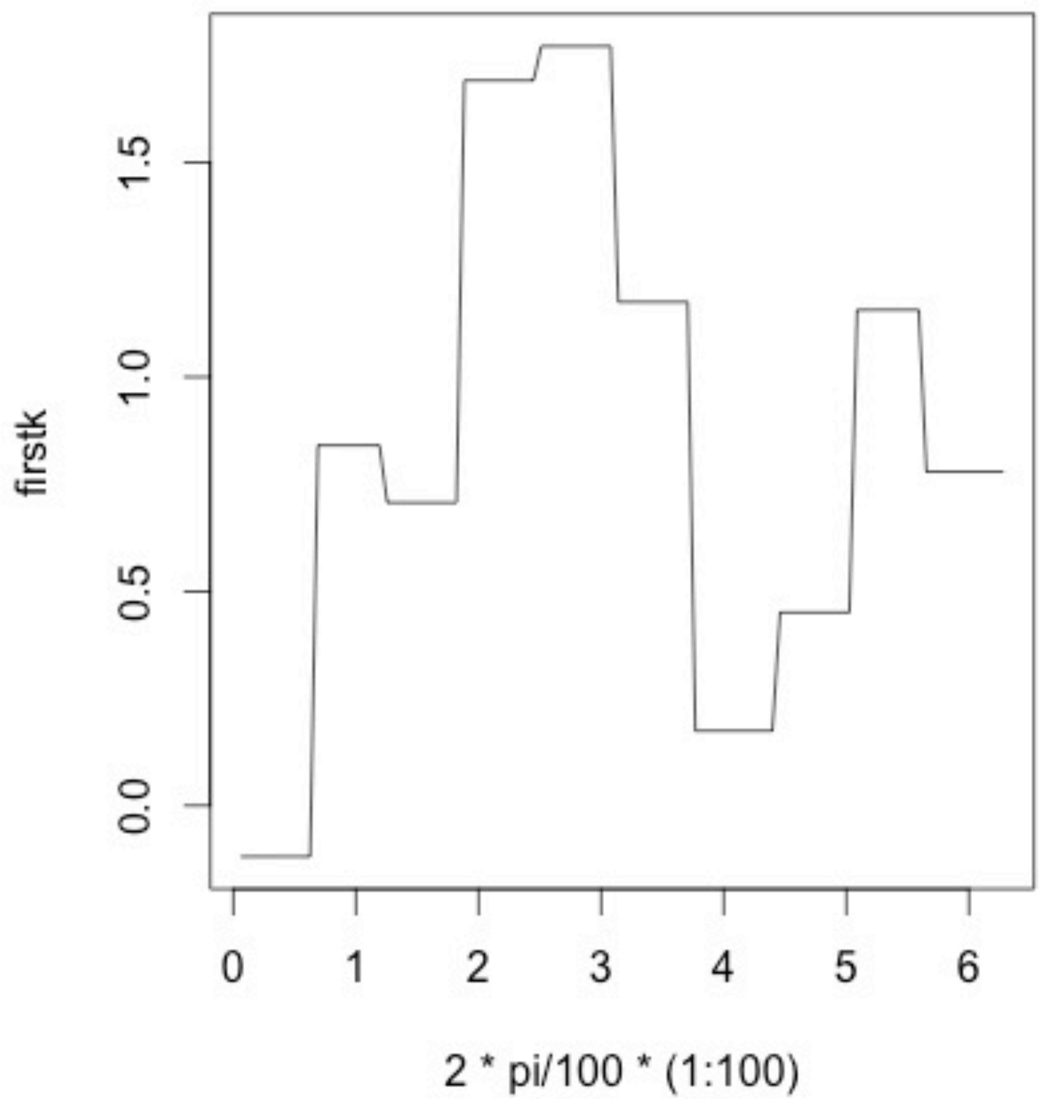

Figure 5

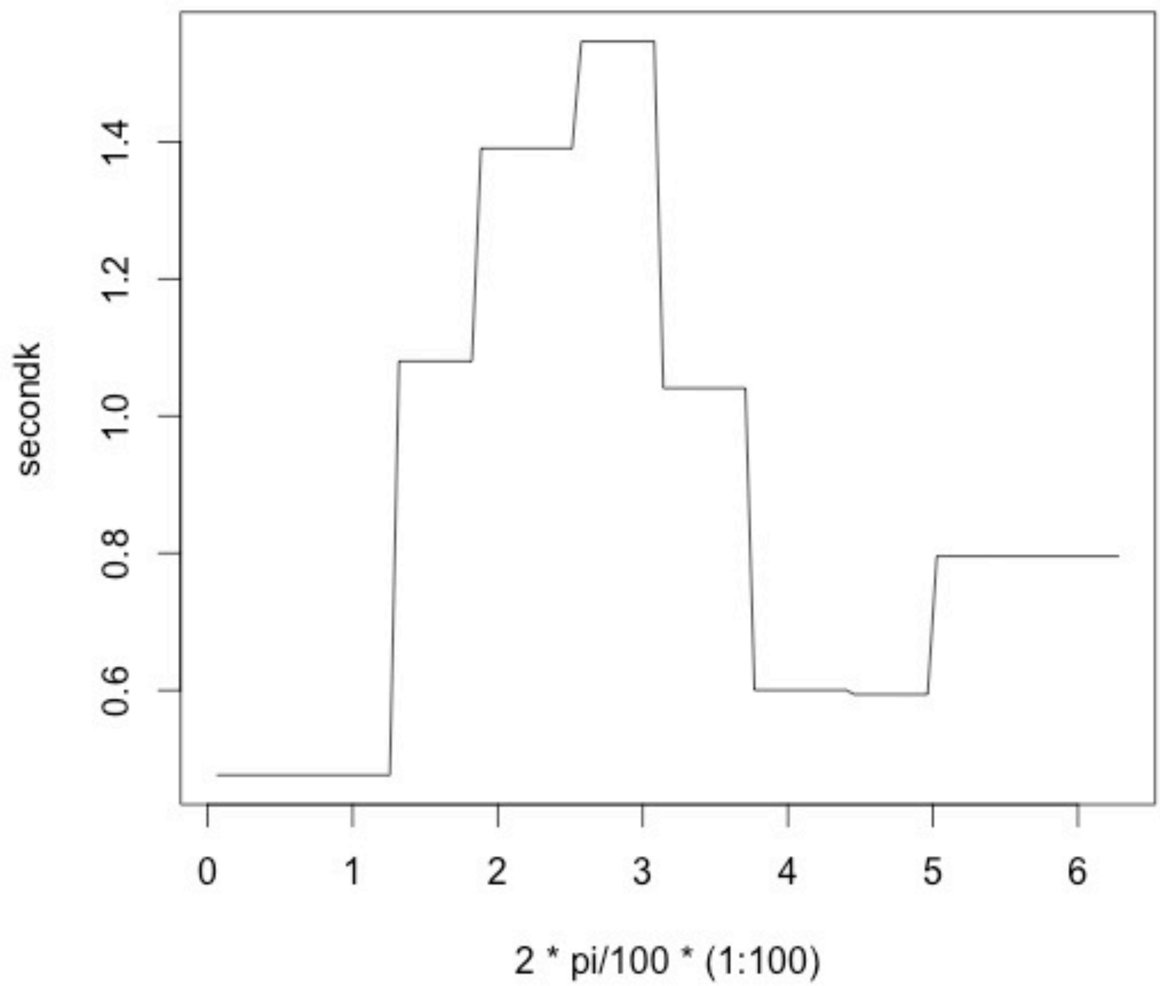

Figure 6

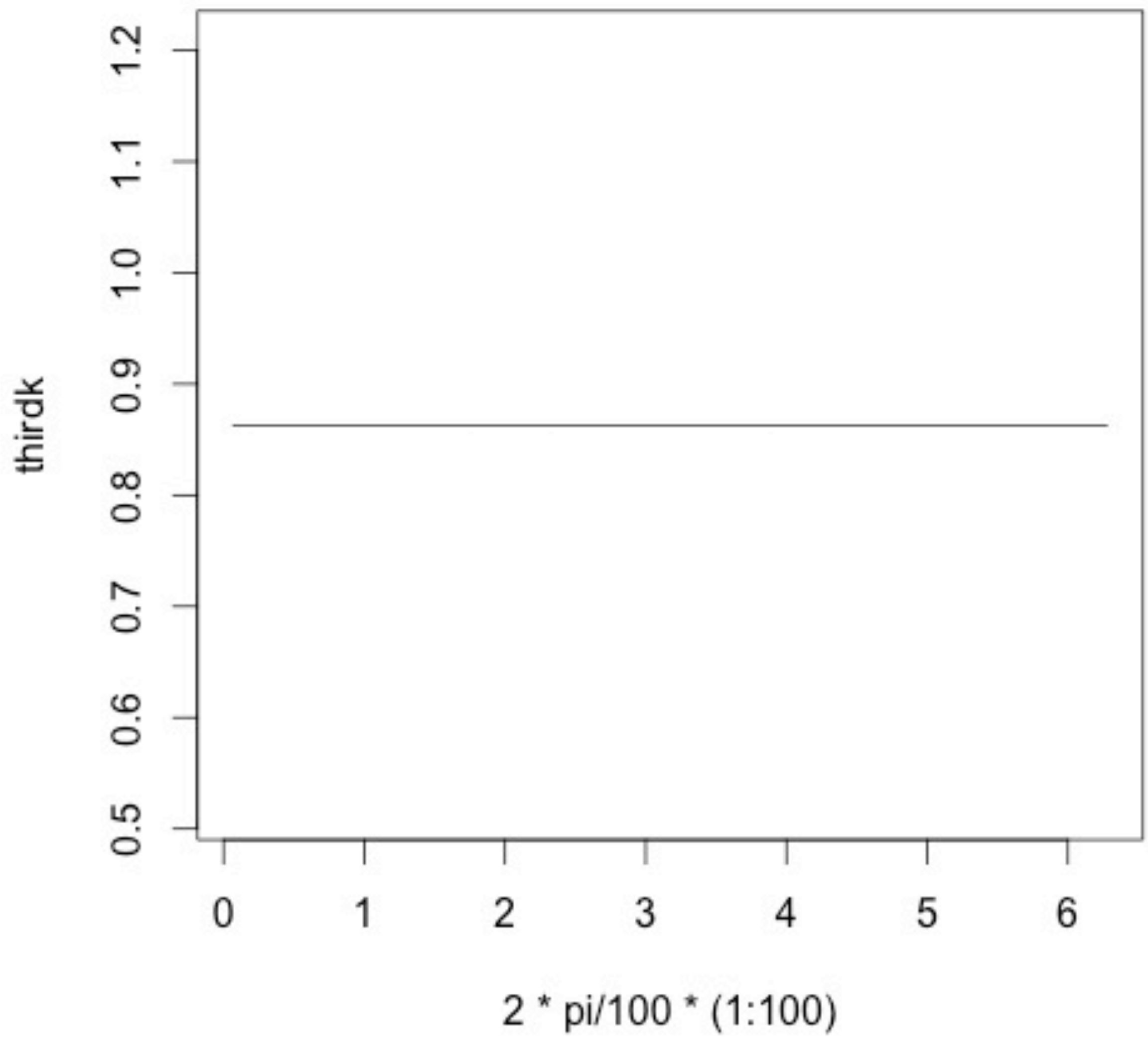

Figure 7

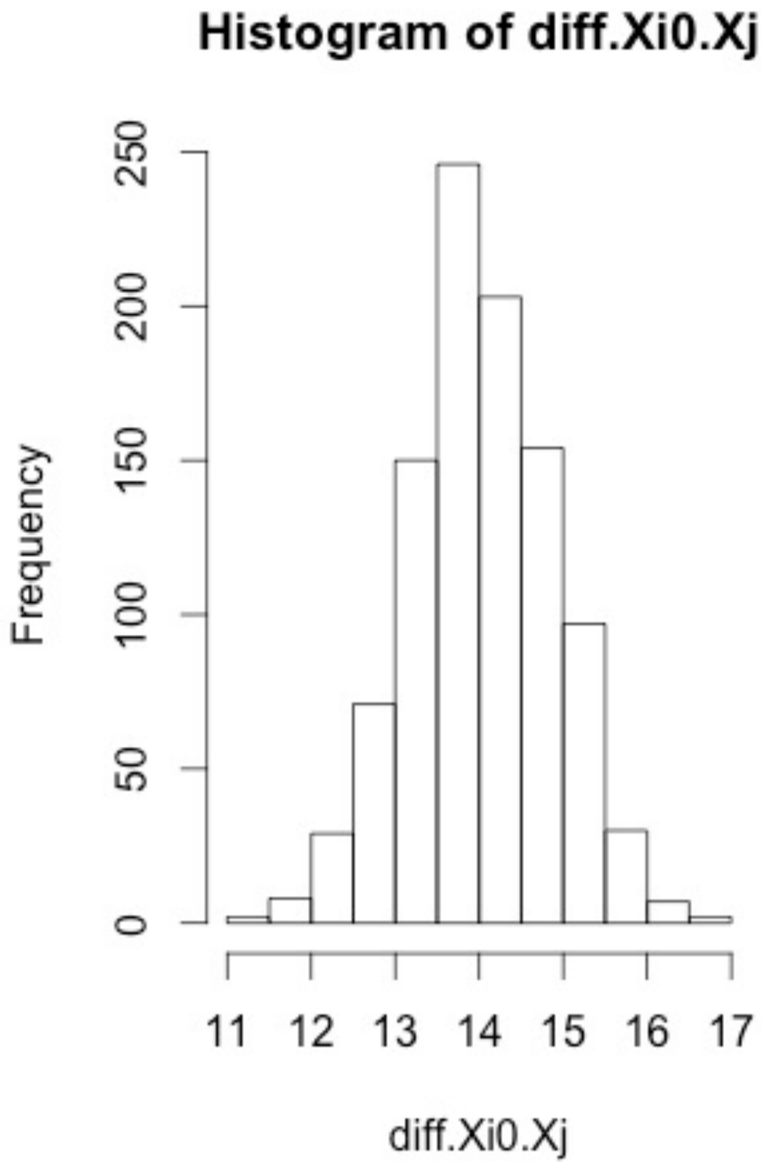
Histogram of diff.Xi0.Xj
Frequency
0
50
100
150
200
250
11
12
13
14
15
16
17
diff.Xi0.Xj

Figure 8

```
#####Problem 1
library("cluster")
library("kernlab")
reduced_voting_record2005 <- read.delim("~/Dropbox/School/Statistics/Stat 154 Spring 2014/HW2/
reduced_voting_record2005.txt", header=F)
reduced_house_party2005 <- na.omit(t(read.delim("~/Dropbox/School/Statistics/Stat 154 Spring 2014/
HW2/reduced_house_party2005.txt", header=F)))
reduced_voting_record2005$V670 = reduced_house_party2005
colorsRep = 1:401
colorsRep[reduced_house_party2005==1] = "blue"
colorsRep[reduced_house_party2005==0] = "red"
colorsRep[reduced_house_party2005==2] = "grey"
party_votes = matrix(nrow=3, ncol=669)
row.names(party_votes) = c("dem", "rep", "indep")
for(i in 1:669){
  party_votes[1, i] = 2*sum(reduced_voting_record2005[,i][reduced_voting_record2005$V670==1])
  party_votes[2, i] = 2*sum(reduced_voting_record2005[,i][reduced_voting_record2005$V670==0])
  party_votes[3, i] = 2*sum(reduced_voting_record2005[,i][reduced_voting_record2005$V670==2])
}
Prob1a = hclust(dist(party_votes))
plot(Prob1a)
Problb = prcomp(reduced_voting_record2005)
plot(Problb$x, col = colorsRep)
Prob1c = kpca(as.matrix(reduced_voting_record2005))
plot(rotated(Prob1c), col=colorsRep)
Prob1d=specc(as.matrix(reduced_voting_record2005), centers = 3)
plot(Prob1d, col = colorsRep)
####Problem 2
library(FNN)
N=10
error1 = rnorm(N, 0, sqrt(1))
Problem2.1 = function(N, k, sigma, test = 2*pi/100*(1:100), error = error1){
  i = 1:N
  x = (i - 1/2)/N*2*pi
  f = cos(10*x)+2
  y = f + error
  results = list()
  for(i in (1:length(test))){
    firstset = list()
    firstset[[1]] = knn.reg(train = x, test = test[i], y=y, k = k)$pred
    firstset[[2]] = x
    firstset[[3]] = y
    results[[i]] = firstset
  }
  return(results)
}

firstk =c()
secondk = c()
thirdk = c()
P2.1 = Problem2.1(10, 1, 0.1)
P2.2 = Problem2.1(10, 3, 0.1)
P2.3 = Problem2.1(10, 10, 0.1)
for(i in 1:100){
  firstk = c(firstk, P2.1[[i]][[1]])
```

```
  secondk= c(secondk, P2.2[[i]][[1]])
  thirdk = c(thirdk, P2.3[[i]][[1]])
}

plot(x = 2*pi/100*(1:100), firstk, type="l")
plot(x = 2*pi/100*(1:100), secondk, type="l")
plot(x = 2*pi/100*(1:100), thirdk, type="l")
###Problem 2.1.1
EPE.X= function(k, f){
  i = 1:N
  x = (i - 1/2)/N*2*pi
    Bias.x = (f(pi, n = 1)-1/k*sum(f(x[order(abs(X[i]-X))[1:k]])))^2
    Var.x = 1/k
    EPE= 1 + Bias.x + Var.x
    Bias[i] = Bias.x
    Var = c(Var, Var.x)
  final.df = data.frame(Bias, Var, EPE)
  return(final.df)
}
N=10
X = runif(N, 0, 2*pi)
f0 = function(x, n = N){
  cos(10*x)+2+ rnorm(n, 0, sqrt(1))}



i = 1:N
x = (i - 1/2)/N*2*pi
y = f0(x)
y1 = lm(y ~ 1)
y1form.X = function(x, n=N){
  y1$coefficients[1]+ rnorm(n, 0, sqrt(1))}

y2 = lm(y ~ 1 + x)
y2form.X = function(x, n=N){
  y2$coefficients[2]*x+y2$coefficients[1]+ rnorm(n, 0, sqrt(1))}

y3 = lm(y ~ 1 + x + I(x^2))
y3form.X = function(x, n=N){
  y3$coefficients[3]*x^2+y3$coefficients[2]*x+y3$coefficients[1] + rnorm(n, 0, sqrt(1))
}
EPE.X(1, f0)
EPE.X(3, f0)
EPE.X(10, f0)

EPE.X(1, y1form.X)
EPE.X(3, y1form.X)
EPE.X(10, y1form.X)

EPE.X(1, y2form.X)
EPE.X(3, y2form.X)
EPE.X(10, y2form.X)

EPE.X(1, y3form.X)
EPE.X(3, y3form.X)
```

```
EPE.X(10, y3form.X)

y = f0(X)
y1 = lm(y ~ 1)
y1form.X = function(x){
  y1$coefficients[1]+ rnorm(N, 0, sqrt(1))}

y2 = lm(y ~ 1 + X)
y2form.X = function(x){
  y2$coefficients[2]*x+y2$coefficients[1]+ rnorm(N, 0, sqrt(1))}

y3 = lm(y ~ 1 + X + I(X^2))
y3form.X = function(x){
  y3$coefficients[3]*x^2+y3$coefficients[2]*x+y3$coefficients[1]
}
y = y3form.X(X)

EPE.X.X= function(X, k, f){
  Bias = c()
  Var = c()
  EPE = c()
  for(i in 1:length(X)){
    Bias.x = (f(X[i]) - 1/k*sum(f(
          X[order(abs(X[i]-X))[1:k]])))^2
    Var.x = 1/k
    EPE[i] = 1 + Bias.x + Var.x
    Bias[i] = Bias.x
    Var = c(Var, Var.x)
}
return(data.frame(EPE, Bias, Var))
}
EPE.X.X(X, 1, f0)
sapply(EPE.X.X(X, 1, f0), mean)
EPE.X.X(X, 3, f0)
sapply(EPE.X.X(X, 3, f0), mean)
EPE.X.X(X, 10, f0)
sapply(EPE.X.X(X, 10, f0), mean)

sapply(EPE.X.X(X, 1, y1form.X), mean)
sapply(EPE.X.X(X, 3, y1form.X), mean)
sapply(EPE.X.X(X, 10, y1form.X), mean)

sapply(EPE.X.X(X, 1, y2form.X), mean)
sapply(EPE.X.X(X, 3, y2form.X), mean)
sapply(EPE.X.X(X, 10, y2form.X), mean)

sapply(EPE.X.X(X, 1, y3form.X), mean)
sapply(EPE.X.X(X, 3, y3form.X), mean)
sapply(EPE.X.X(X, 10, y3form.X), mean)


###For EPE(pi)), where X is unformily distributed from 0 to 2*pi, the linear function gives the lowest output;
####however, the quadratic is only marginally higher.
```

```
###For E(EPE(X)), where X is unformily distributed from 0 to 2*pi, the quadratic function gives the lowest
output;
####however, the linear is only marginally higher.
###Problem 2.2
N=100
error1 = rnorm(N, 0, sqrt(1))
Problem2.1 = function(N, k, sigma, test = 2*pi/100*(1:100), error = error1){
  i = 1:N
  x = (i - 1/2)/N*2*pi
  f = cos(10*x)+2
  y = f + error
  results = list()
  for(i in (1:length(test))){
    firstset = list()
    firstset[[1]] = knn.reg(train = x, test = test[i], y=y, k = k)$pred
    firstset[[2]] = x
    firstset[[3]] = y
    results[[i]] = firstset
  }
  return(results)
}

firstk =c()
secondk = c()
thirdk = c()
P2.1 = Problem2.1(10, 1, 0.1)
P2.2 = Problem2.1(10, 3, 0.1)
P2.3 = Problem2.1(10, 10, 0.1)
for(i in 1:100){
  firstk = c(firstk, P2.1[[i]][[1]])
  secondk= c(secondk, P2.2[[i]][[1]])
  thirdk = c(thirdk, P2.3[[i]][[1]])
}

plot(x = 2*pi/100*(1:100), firstk, type="l")
plot(x = 2*pi/100*(1:100), secondk, type="l")
plot(x = 2*pi/100*(1:100), thirdk, type="l")
###Problem 2.1.1
EPE.X= function(k, f){
  i = 1:N
  x = (i - 1/2)/N*2*pi
  Bias.x = (f(pi, n = 1)-1/k*sum(f(x[order(abs(X[i]-X))[1:k]])))^2
  Var.x = 1/k
  EPE= 1 + Bias.x + Var.x
  Bias[i] = Bias.x
  Var = c(Var, Var.x)
  final.df = list(Bias, Var, EPE)
  return(final.df)
}
N=100
X = runif(N, 0, 2*pi)
f0 = function(x, n = N){
  cos(10*x)+2+ rnorm(n, 0, sqrt(1))}
```

```
i = 1:N
x = (i - 1/2)/N*2*pi
y = f0(x)
y1 = lm(y ~ 1)
y1form.X = function(x, n=N){
  y1$coefficients[1]+ rnorm(n, 0, sqrt(1))}

y2 = lm(y ~ 1 + x)
y2form.X = function(x, n=N){
  y2$coefficients[2]*x+y2$coefficients[1]+ rnorm(n, 0, sqrt(1))}

y3 = lm(y ~ 1 + x + I(x^2))
y3form.X = function(x, n=N){
  y3$coefficients[3]*x^2+y3$coefficients[2]*x+y3$coefficients[1] + rnorm(n, 0, sqrt(1))
}
EPE.X(1, f0)[1,]
EPE.X(3, f0)[1,]
EPE.X(10, f0)[1,]
EPE.X(20, f0)[1,]
EPE.X(50, f0)[1,]


EPE.X(1, y3form.X)[1,]
EPE.X(3, y3form.X)[1,]
EPE.X(10, y3form.X)[1,]
EPE.X(20, y3form.X)[1,]
EPE.X(50, y3form.X)[1,]


y6 = lm(y ~ 1 + x + I(x^2)+I(x^3)+I(x^4)+I(x^5))
y6form.X = function(x, n=N){
  sum(as.numeric(y6$coefficients)*(rep(x, 6))^(0:5))
}
EPE.X(1, y6form.X)
EPE.X(3, y6form.X)
EPE.X(10, y6form.X)
EPE.X(20, y6form.X)
EPE.X(50, y6form.X)

y = f0(X)
y1 = lm(y ~ 1)
y1form.X = function(x){
  y1$coefficients[1]+ rnorm(N, 0, sqrt(1))}

y2 = lm(y ~ 1 + X)
y2form.X = function(x){
  y2$coefficients[2]*x+y2$coefficients[1]+ rnorm(N, 0, sqrt(1))}

y3 = lm(y ~ 1 + X + I(X^2))
y3form.X = function(x){
  y3$coefficients[3]*x^2+y3$coefficients[2]*x+y3$coefficients[1]
}
y = y3form.X(X)
```

```
y6 = lm(y ~ 1 + X + I(X^2)+I(X^3)+I(X^4)+I(X^5))
y6form.X = function(x, n=N){
  sum(as.numeric(y6$coefficients)*(rep(x, 6))^(0:5))
}

EPE.X.X= function(X, k, f){
  Bias = c()
  Var = c()
  EPE = c()
  for(i in 1:length(X)){
    Bias.x = (f(X[i]) - 1/k*sum(f(
      X[order(abs(X[i]-X))[2:(k+1)]])))^2
    Var.x = 1/k
    EPE[i] = 1 + Bias.x + Var.x
    Bias[i] = Bias.x
    Var = c(Var, Var.x)
  }
  return(data.frame(EPE, Bias, Var))
}
EPE.X.X(X, 1, f0)
sapply(EPE.X.X(X, 1, f0), mean)
EPE.X.X(X, 3, f0)
sapply(EPE.X.X(X, 3, f0), mean)
EPE.X.X(X, 10, f0)
sapply(EPE.X.X(X, 10, f0), mean)

sapply(EPE.X.X(X, 1, y1form.X), mean)
sapply(EPE.X.X(X, 3, y1form.X), mean)
sapply(EPE.X.X(X, 10, y1form.X), mean)

sapply(EPE.X.X(X, 1, y2form.X), mean)
sapply(EPE.X.X(X, 3, y2form.X), mean)
sapply(EPE.X.X(X, 10, y2form.X), mean)

sapply(EPE.X.X(X, 1, y3form.X), mean)
sapply(EPE.X.X(X, 3, y3form.X), mean)
sapply(EPE.X.X(X, 10, y3form.X), mean)

sapply(EPE.X.X(X, 1, y6form.X), mean)
sapply(EPE.X.X(X, 3, y6form.X), mean)
sapply(EPE.X.X(X, 10, y6form.X), mean)
sapply(EPE.X.X(X, 20, y6form.X), mean)
sapply(EPE.X.X(X, 50, y6form.X), mean)


###Probem2.2.2

EPE.X.X= function(X, k, f){
  Bias = c()
  Var = c()
  EPE = c()
  for(i in 1:length(X)){
    Bias.x = (
      f(X[i]) -
        1/k*sum(f(
```

```
        X[order(abs(X[i]-X))[1:k]]
        ))
          )^2
    Var.x = 1/k
    EPE[i] = 1 + Bias.x + Var.x
    Bias[i] = Bias.x
    Var = c(Var, Var.x)
}
return(data.frame(Bias, Var, EPE))
}


###Problem 3
N = 50
x = runif(5, 0, 2*pi)
f0 = function(x, k){
  summing = c()
  for(i in 1:4){
    summing[i] = sum(sin(sqrt(k)*x[i]))+sum(cos(x[i]*x[(i+1)]))
}
return(summing)
}
y = f0(x, 1)

y1 = lm(y ~ 1 + x[1:4])
y1form.X = function(x, n = N){
  y1$coefficients[2]*x+ y1$coefficients[1]}
EPE.X.X(x, 1, y1form.X)

y = f0(x, 4)

y1 = lm(y ~ 1 + x)
y1form.X = function(x){
  y1$coefficients[2]*x+ y1$coefficients[1]}
EPE.X.X(x, 4, y1form.X)

y = f0(x, 5)

y1 = lm(y ~ 1 + x)
y1form.X = function(x){
  y1$coefficients[2]*x+ y1$coefficients[1]}

EPE.X.X= function(X, k, f){
  Bias = c()
  Var = c()
  EPE = c()
  for(i in 1:length(X)){
    Bias.x = (f(X[i])-1/k*sum(f(x[order(abs(X[i]-X))[1:k]])))^2
    Var.x = 1/k
    EPE[i] = 1 + Bias.x + Var.x
    Bias[i] = Bias.x
    Var = c(Var, Var.x)
  }
  final.df = data.frame(Bias, Var, EPE)
  return(final.df)
```

```
}

EPE.X.X(x, 5, y1form.X)
####Problem 4
Problem4 = data.frame(v1=rnorm(100, 0, sqrt(1)))
for( i in 2:1000){
  Problem4[,i]=rnorm(100, 0, sqrt(1))
}
i0 = sample(x=1:1000, 1)
Xi0= Problem4[,i0]
Xj= Problem4[,-c(i0)]

diff.Xi0.Xj = c()
for(i in 1:999){
  diff.Xi0.Xj[i] = sqrt(sum((Xi0-Xj[,i])^2))
}
1/sqrt(100)*min(diff.Xi0.Xj)
1/sqrt(100)*max(diff.Xi0.Xj)
```