TD2_R

March 31, 2021

```
[1]: prostate<-read.table('prostate.data',header = TRUE)
```

```
[2]: prostate
```

A data.frame: 97 × 10

	lcavol	lweight	age	lbph	svi	lcp	gleason	pgg4
	<dbl>	<dbl>	<int>	<dbl>	<int>	<dbl>	<int>	<int
1	-0.5798185	2.769459	50	-1.3862944	0	-1.38629436	6	0
2	-0.9942523	3.319626	58	-1.3862944	0	-1.38629436	6	0
3	-0.5108256	2.691243	74	-1.3862944	0	-1.38629436	7	20
4	-1.2039728	3.282789	58	-1.3862944	0	-1.38629436	6	0
5	0.7514161	3.432373	62	-1.3862944	0	-1.38629436	6	0
6	-1.0498221	3.228826	50	-1.3862944	0	-1.38629436	6	0
7	0.7371641	3.473518	64	0.6151856	0	-1.38629436	6	0
8	0.6931472	3.539509	58	1.5368672	0	-1.38629436	6	0
9	-0.7765288	3.539509	47	-1.3862944	0	-1.38629436	6	0
10	0.2231436	3.244544	63	-1.3862944	0	-1.38629436	6	0
11	0.2546422	3.604138	65	-1.3862944	0	-1.38629436	6	0
12	-1.3470736	3.598681	63	1.2669476	0	-1.38629436	6	0
13	1.6134299	3.022861	63	-1.3862944	0	-0.59783700	7	30
14	1.4770487	2.998229	67	-1.3862944	0	-1.38629436	7	5
15	1.2059708	3.442019	57	-1.3862944	0	-0.43078292	7	5
16	1.5411591	3.061052	66	-1.3862944	0	-1.38629436	6	0
17	-0.4155154	3.516013	70	1.2441546	0	-0.59783700	7	30
18	2.2884862	3.649359	66	-1.3862944	0	0.37156356	6	0
19	-0.5621189	3.267666	41	-1.3862944	0	-1.38629436	6	0
20	0.1823216	3.825375	70	1.6582281	0	-1.38629436	6	0
21	1.1474025	3.419365	59	-1.3862944	0	-1.38629436	6	0
22	2.0592388	3.501043	60	1.4747630	0	1.34807315	7	20
23	-0.5447272	3.375880	59	-0.7985077	0	-1.38629436	6	0
24	1.7817091	3.451574	63	0.4382549	0	1.17865500	7	60
25	0.3852624	3.667400	69	1.5993876	0	-1.38629436	6	0
26	1.4469190	3.124565	68	0.3001046	0	-1.38629436	6	0
27	0.5128236	3.719651	65	-1.3862944	0	-0.79850770	7	70
28	-0.4004776	3.865979	67	1.8164521	0	-1.38629436	7	20
29	1.0402767	3.128951	67	0.2231435	0	0.04879016	7	80
30	2.4096442	3.375880	65	-1.3862944	0	1.61938824	6	0
...	...	...	...	...	...	...	...	...
68	2.1983351	4.050915	72	2.30757263	0	-0.4307829	7	10
69	-0.4462871	4.408547	69	-1.38629436	0	-1.3862944	6	0
70	1.1939225	4.780383	72	2.32630162	0	-0.7985077	7	5
71	1.8640801	3.593194	60	-1.38629436	1	1.3217558	7	60
72	1.1600209	3.341093	77	1.74919985	0	-1.3862944	7	25
73	1.2149127	3.825375	69	-1.38629436	1	0.2231435	7	20
74	1.8389611	3.236716	60	0.43825493	1	1.1786550	9	90
75	2.9992262	3.849083	69	-1.38629436	1	1.9095425	7	20
76	3.1411305	3.263849	68	-0.05129329	1	2.4203681	7	50
77	2.0108950	4.433789	72	2.12226154	0	0.5007753	7	60
78	2.5376572	4.354784	78	2.32630162	0	-1.3862944	7	10
79	2.6483002	3.582129	69	-1.38629436	1	2.5839975	7	70
80	2.7794402	3.823192	63	-1.38629436	0	0.3715636	7	50
81	1.4678743	3.070376	66	0.55961579	0	0.2231435	7	40
82	2.5136561	3.473518	57	0.43825493	0	2.3272777	7	60
83	2.6130067	3.888754	77	-0.52763274	1	0.5596158	7	30
84	2.6775910	3.838376	65	1.11514159	0	1.7491998	9	70
85	1.5623463	3.709907	60	1.69561561	0	0.8109302	7	30
86	3.3028493	3.518980	64	-1.38629436	1	2.3272777	7	60
87	2.0241931	3.731699	58	1.63899671	0	-1.3862944	6	0

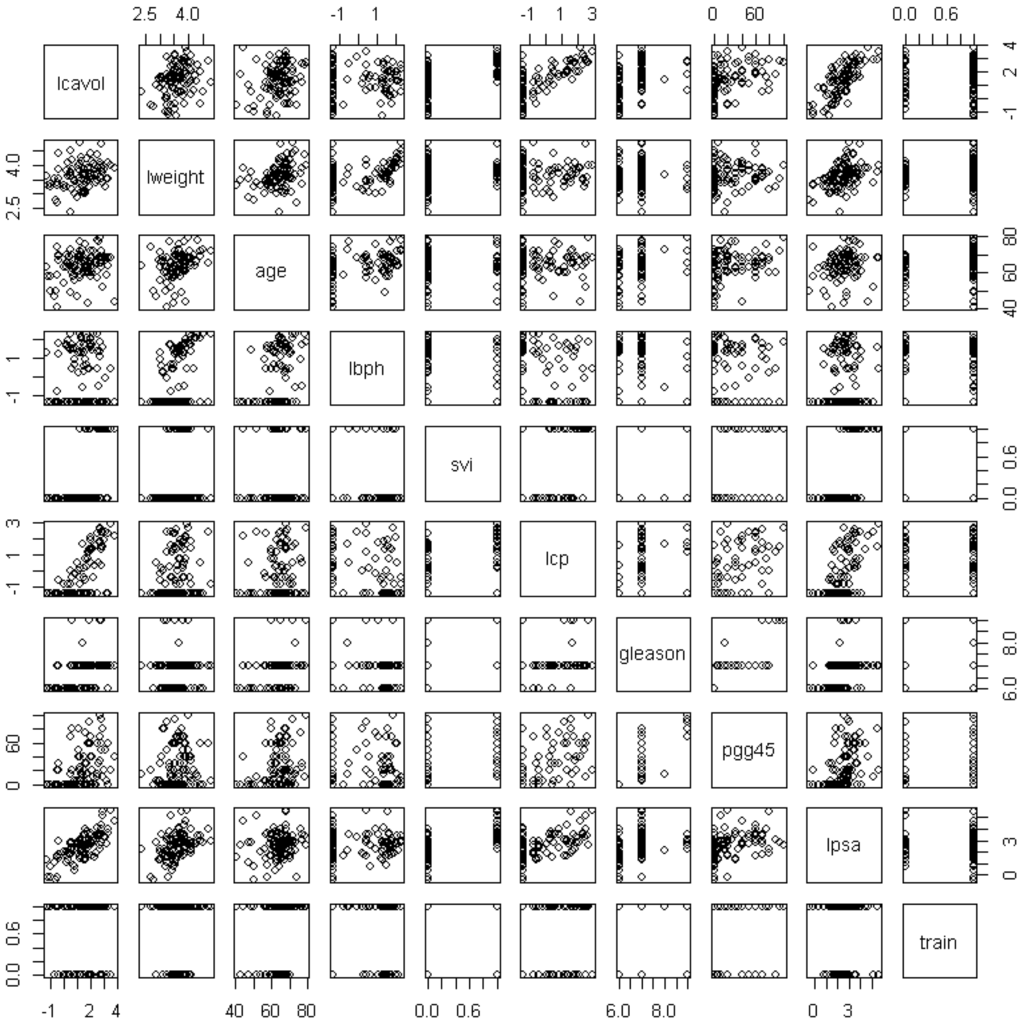
[3]:

```
summary(prostate)
```

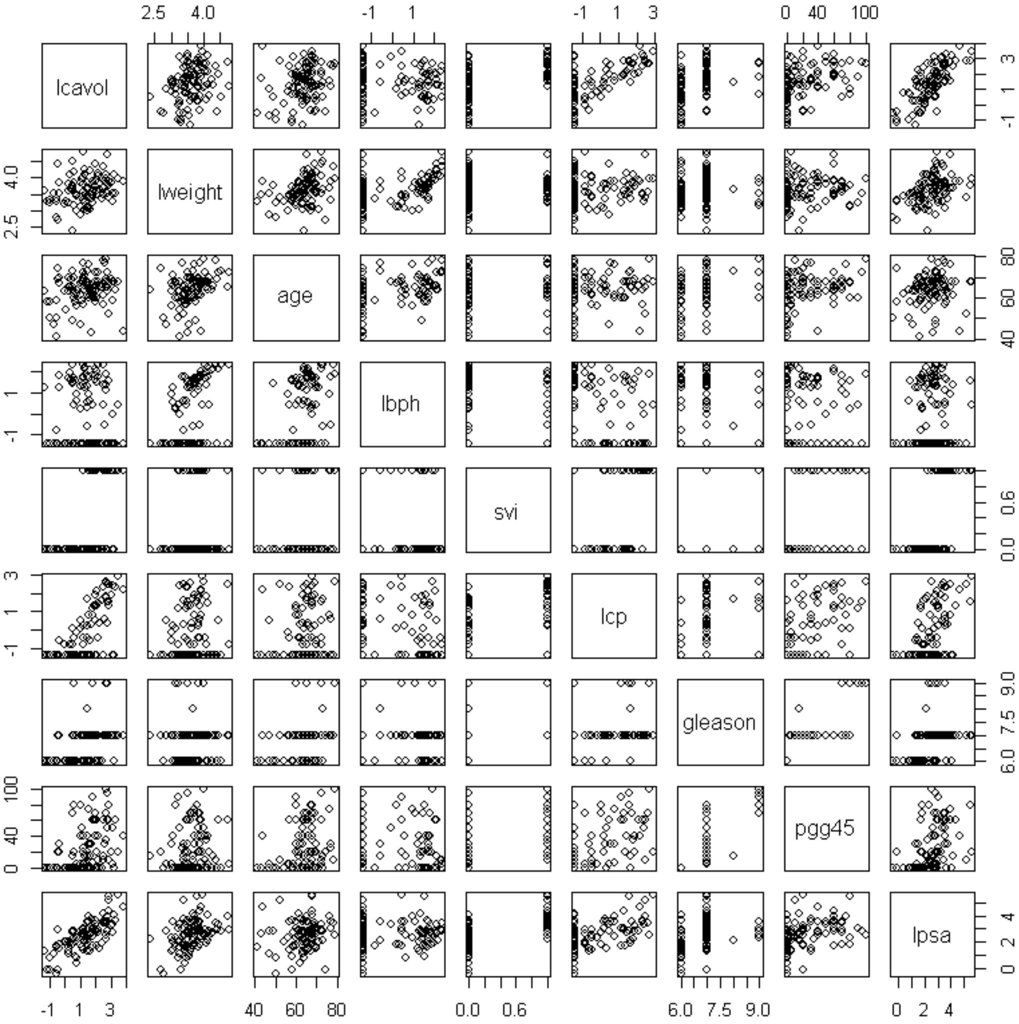
```
    lcavol           lweight            age              lbph        
Min.   :-1.3471   Min.   :2.375   Min.   :41.00   Min.   :-1.3863  
1st Qu.: 0.5128   1st Qu.:3.376   1st Qu.:60.00   1st Qu.:-1.3863  
Median : 1.4469   Median :3.623   Median :65.00   Median : 0.3001  
Mean   : 1.3500   Mean   :3.629   Mean   :63.87   Mean   : 0.1004  
3rd Qu.: 2.1270   3rd Qu.:3.876   3rd Qu.:68.00   3rd Qu.: 1.5581  
Max.   : 3.8210   Max.   :4.780   Max.   :79.00   Max.   : 2.3263  
     svi              lcp             gleason          pgg45       
Min.   :0.0000   Min.   :-1.3863   Min.   :6.000   Min.   :  0.00  
1st Qu.:0.0000   1st Qu.:-1.3863   1st Qu.:6.000   1st Qu.:  0.00  
Median :0.0000   Median :-0.7985   Median :7.000   Median : 15.00  
Mean   :0.2165   Mean   :-0.1794   Mean   :6.753   Mean   : 24.38  
3rd Qu.:0.0000   3rd Qu.: 1.1787   3rd Qu.:7.000   3rd Qu.: 40.00  
Max.   :1.0000   Max.   : 2.9042   Max.   :9.000   Max.   :100.00  
     lpsa           train        
Min.   :-0.4308   Mode :logical  
1st Qu.: 1.7317   FALSE:30       
Median : 2.5915   TRUE :67       
Mean   : 2.4784                  
3rd Qu.: 3.0564                  
Max.   : 5.5829                  
```

[4]:

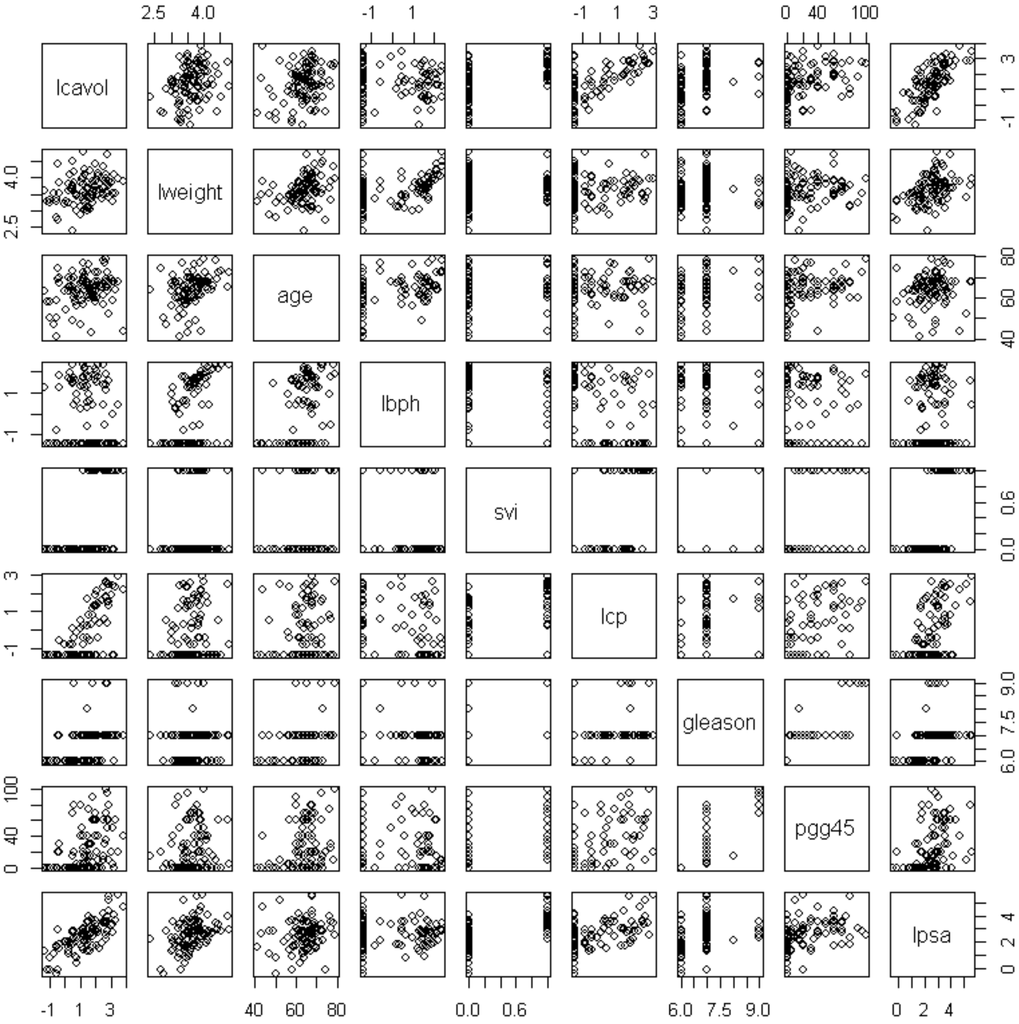
```
plot(prostate)
```



```
[5]: plot(prostate[,-10])
```

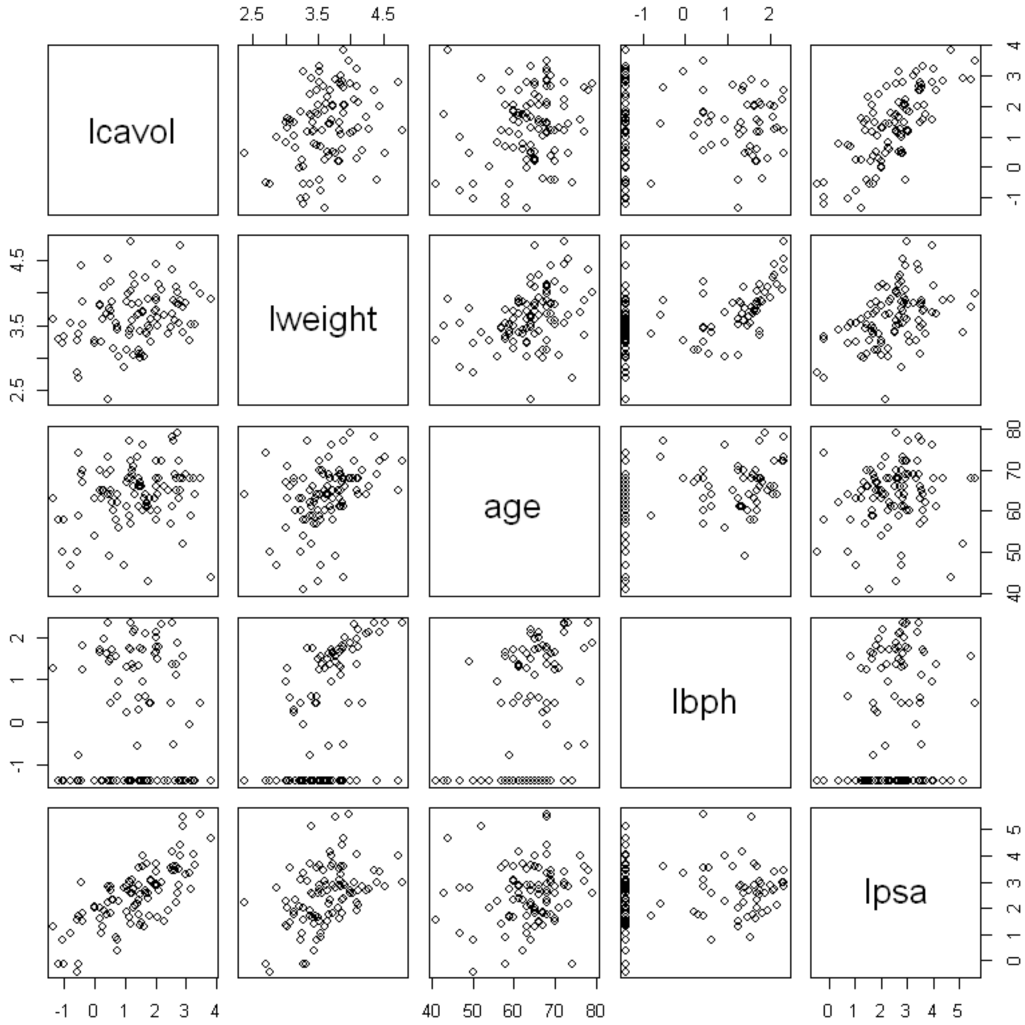


```
[6]: plot(prostate[,-10])
```

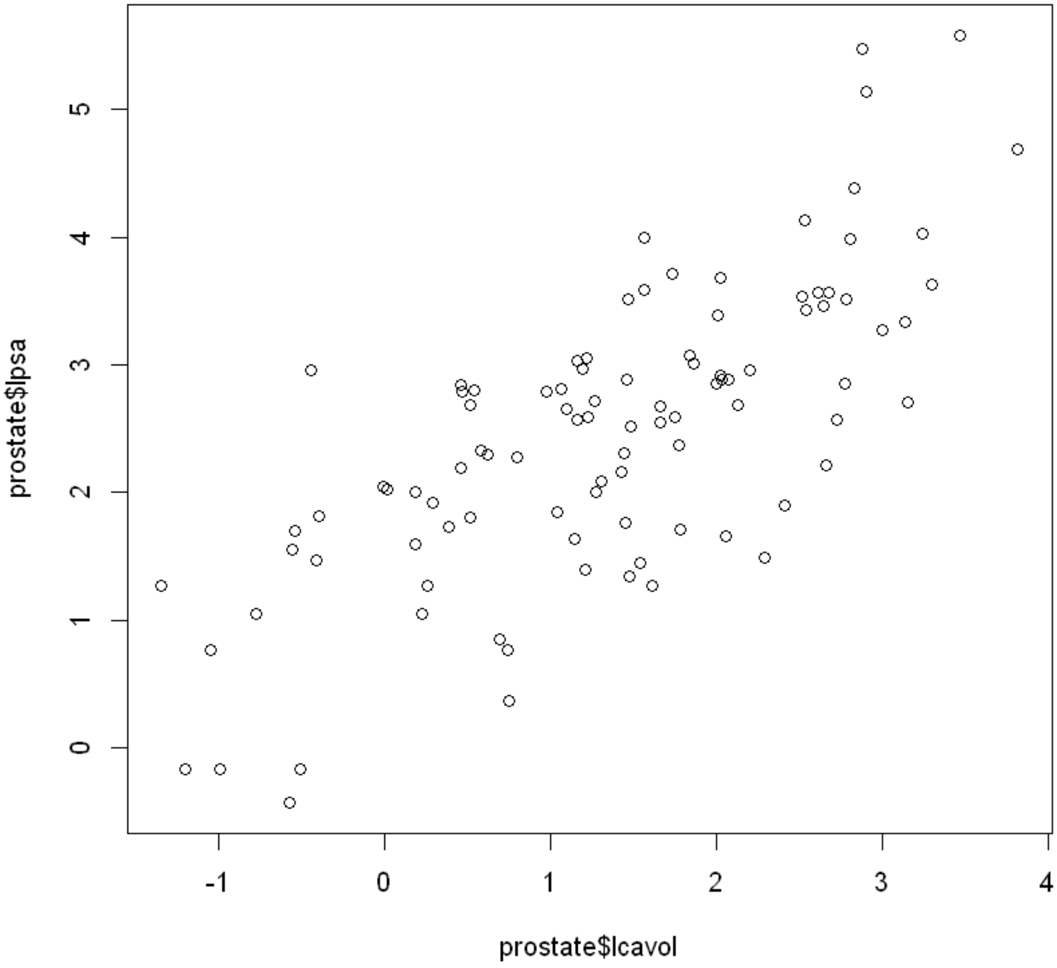


[7]:

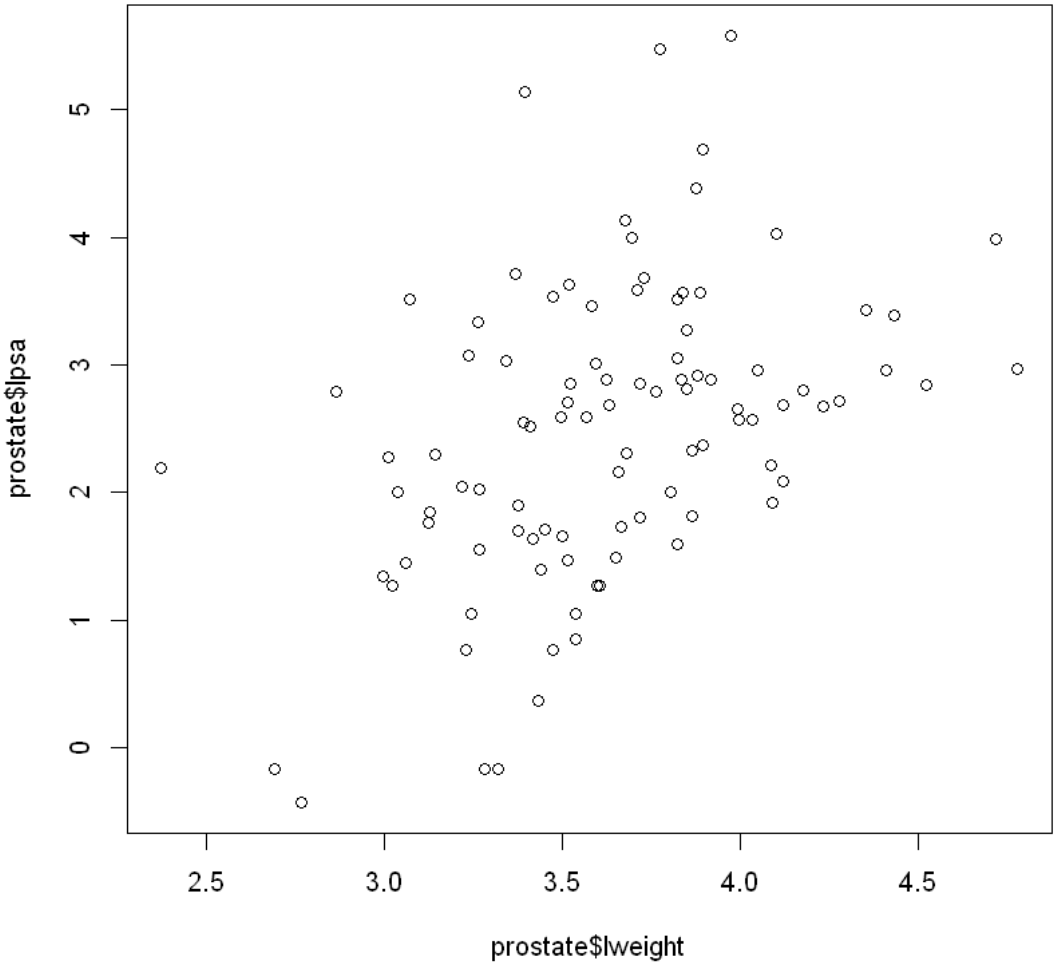
```
plot(prostate[,c(1,2,3,4,9)])
```



[8]:
```
plot(prostate$lcavol,prostate$lpsa)
```

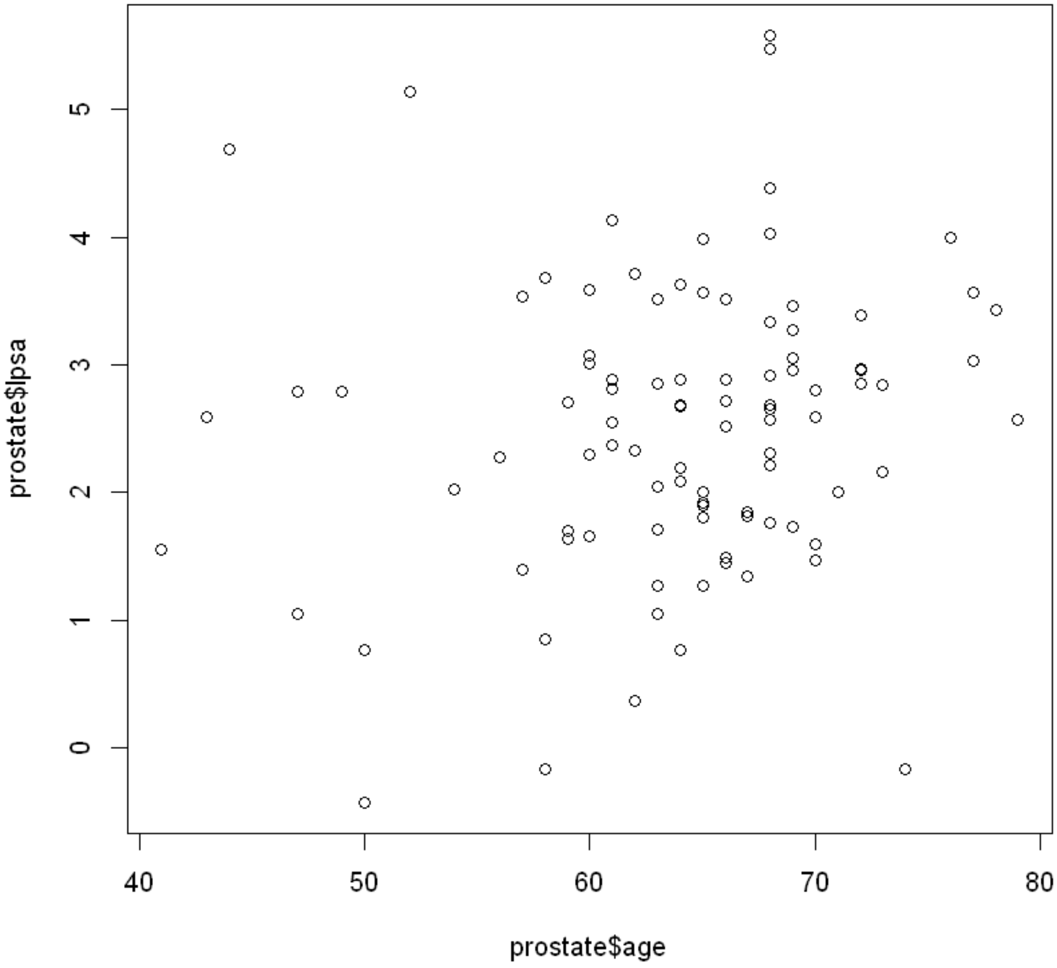


[9]:
```
plot(prostate$lweight,prostate$lpsa)
```



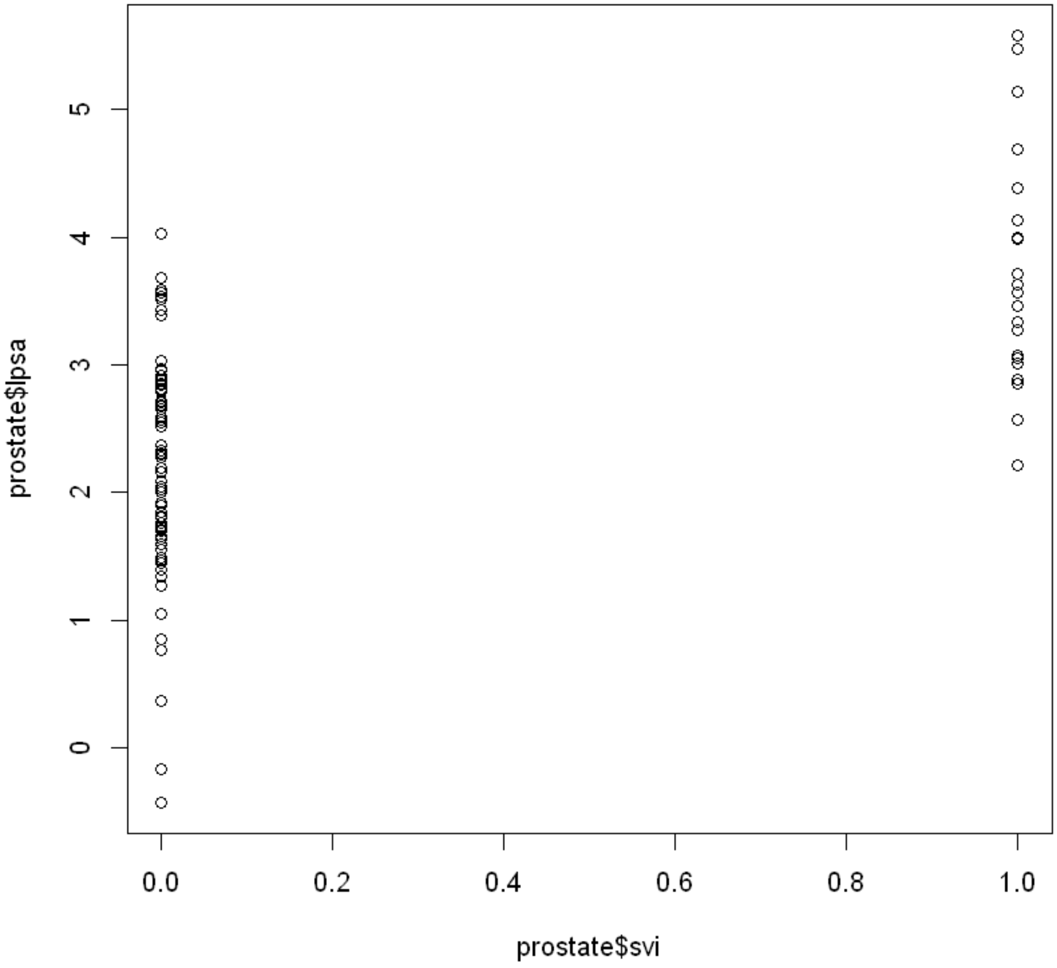
[10]:

```
plot(prostate$age,prostate$lpsa)
```

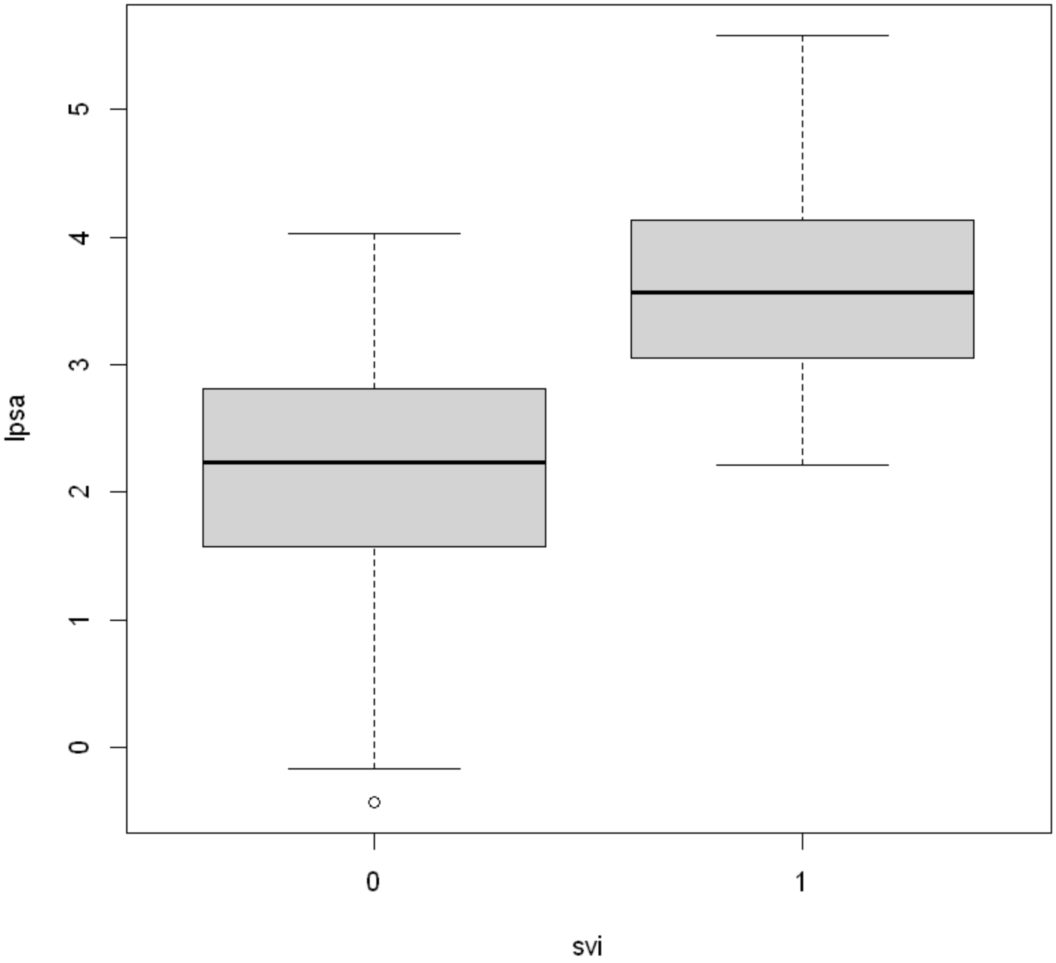


[11]:

```
plot(prostate$svi,prostate$lpsa)
```



[12]:
```
boxplot(lpsa~svi,data=prostate,xlab="svi",ylab="lpsa") # label
```



[13]:
```
# Q3
```

[14]:
```
library('FNN')
```

[15]:
```
data<-prostate[,c('lcavol','lweight','age','lbph','lpsa','train')]
```

[17]:
```
x.train<-scale(data[data$train==T,1:4])
y.train<-data[data$train==T,5]
x.test<-scale(data[data$train==F,1:4])
y.test<-data[data$train==F,5]
```

[18]:
```
# https://www.cnblogs.com/listenfwind/p/10311496.html
reg<-knn.reg(train=x.train, test = x.test, y=y.train, k = 5)
```

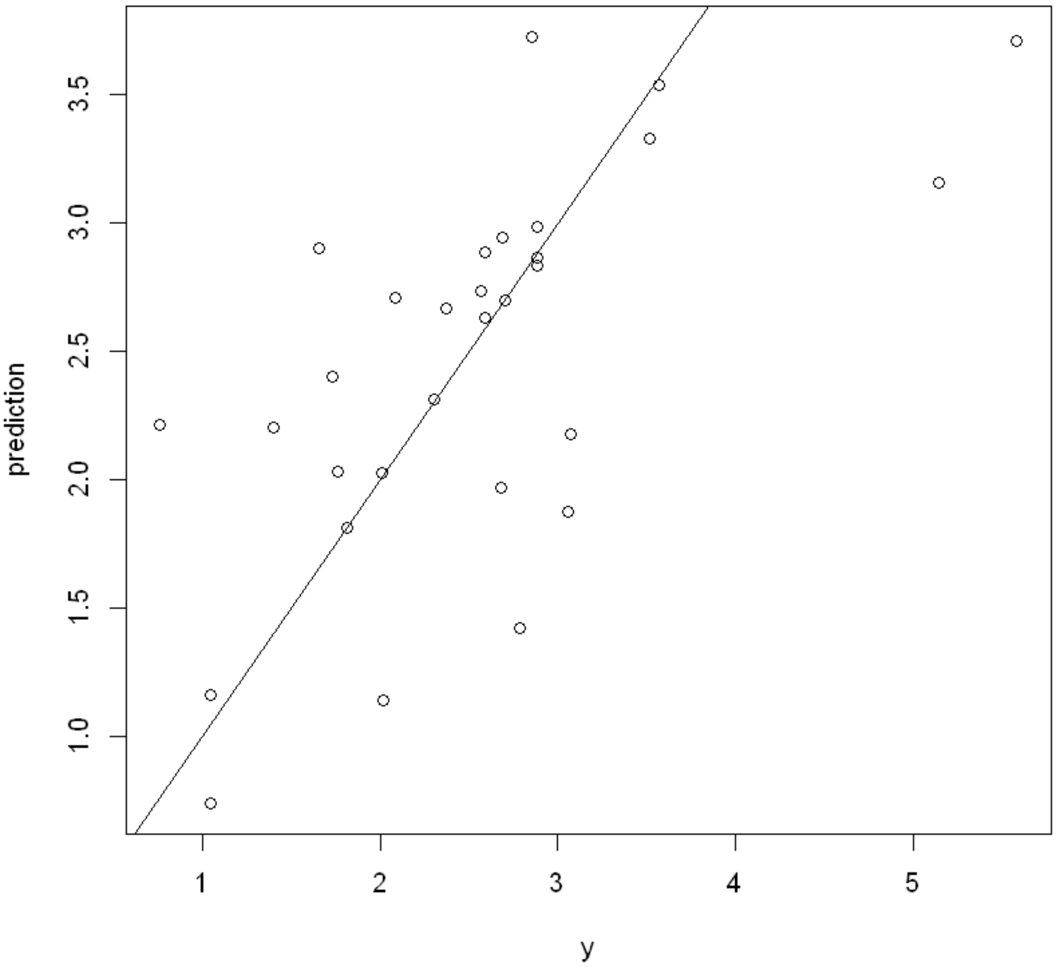
[19]:

```
mean((y.test-reg$pred)^2)
```

0.639996519785868

[20]:

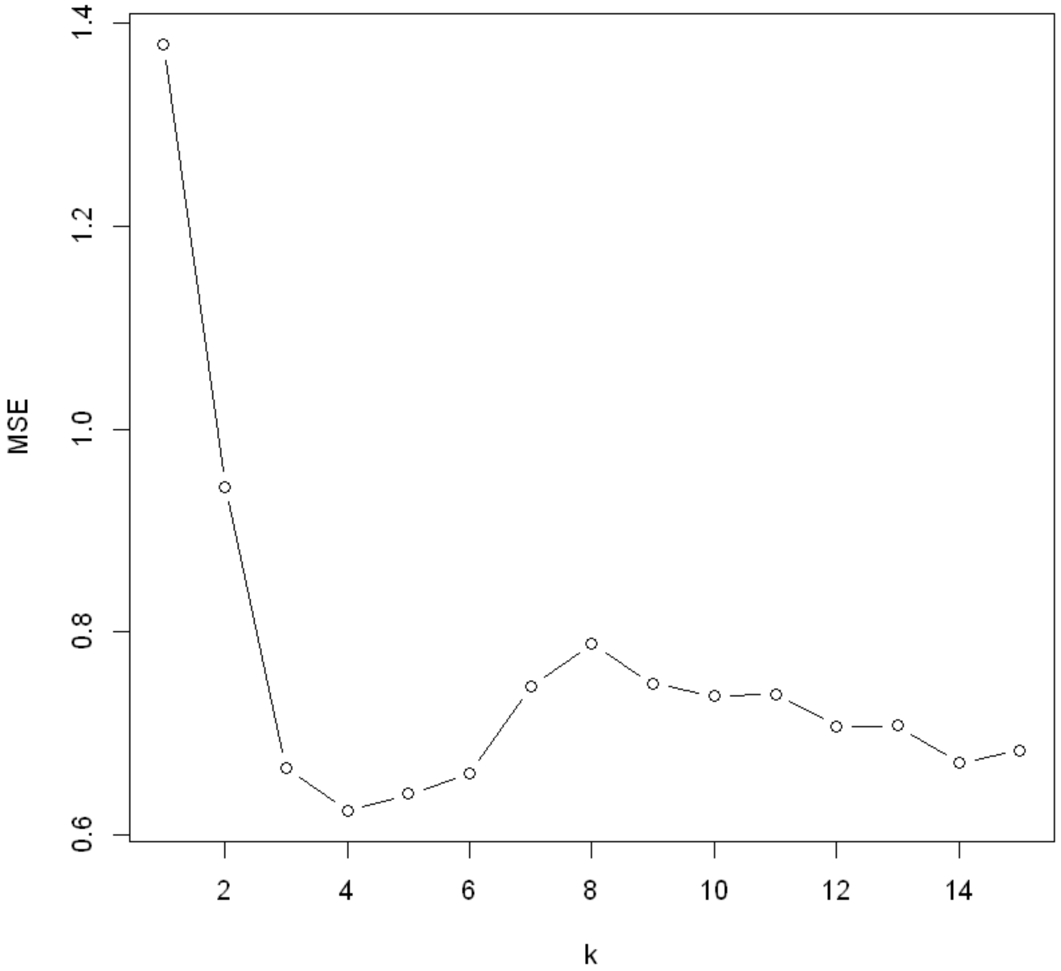
```
plot(y.test,reg$pred,xlab='y',ylab='prediction')
abline(0,1)
```



[21]:

```
MSE<-rep(0,15)
for(k in 1:15){
  reg<-knn.reg(train=x.train, test = x.test,
               y=y.train, k = k)
  MSE[k]<-mean((y.test-reg$pred)^2)
}
```

```
plot(1:15,MSE,type='b',xlab='k',ylab='MSE')
```



[22]:

```
#-------------------------------------------------
# Part II

# Q1
```

[23]:

```
library(mvtnorm)
mu1<-c(0,0)
mu2<-c(0,2)
mu3<-c(2,0)
```

[24]:

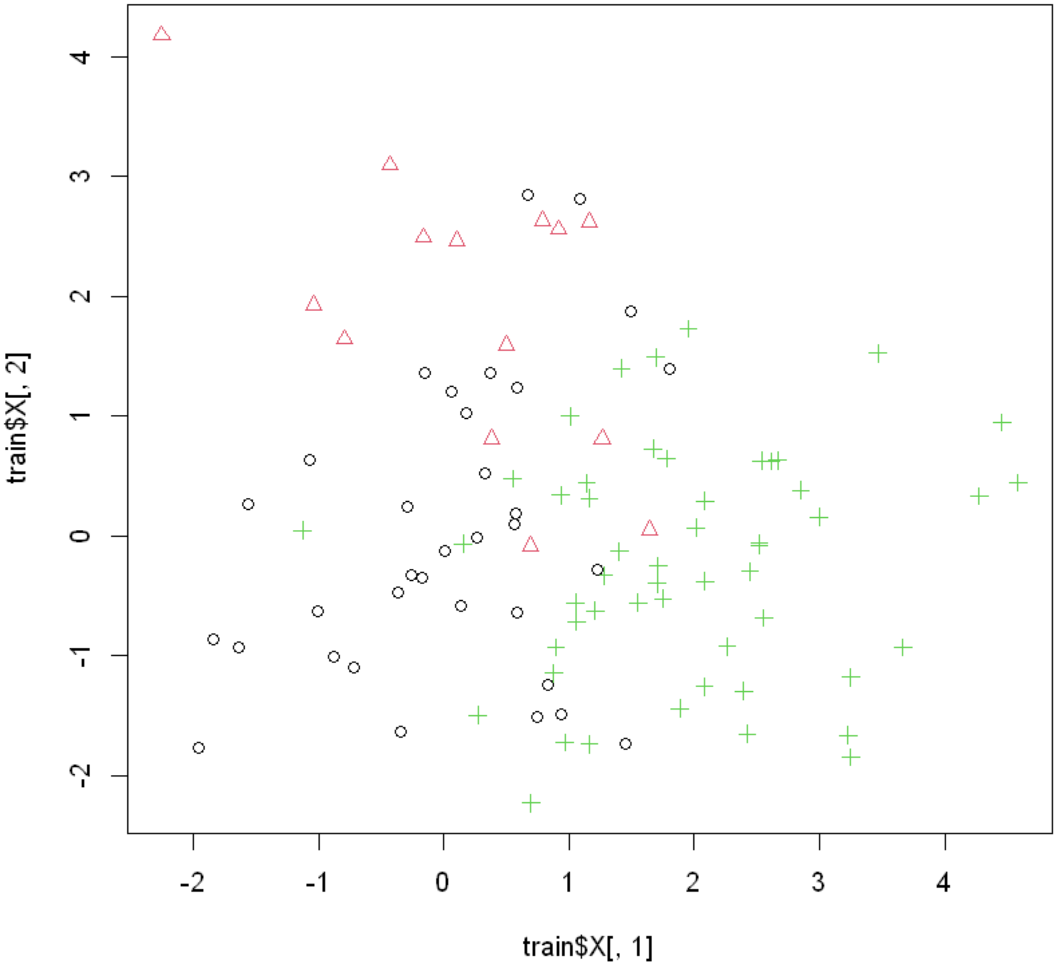
```
Sigma1<-matrix(c(1,0.5,0.5,2),2,2)
Sigma2<-matrix(c(2,-0.5,-0.5,1),2,2)
Sigma3<-diag(c(1,1))
```

[25]:

```
# Function to generate a data set
gen.data<- function(N,mu1,mu2,mu3,Sigma1,Sigma2,Sigma3,p1,p2){
  y<-sample(3,N,prob=c(p1,p2,1-p1-p2),replace=TRUE)
  X<-matrix(0,N,2)
  N1<-length(which(y==1)) # number of objects from class 1
  N2<-length(which(y==2))
  N3<-length(which(y==3))
  X[y==1,]<-rmvnorm(N1,mu1,Sigma1)
  X[y==2,]<-rmvnorm(N2,mu2,Sigma2)
  X[y==3,]<-rmvnorm(N3,mu3,Sigma3)
  return(list(X=X,y=y))
}
```

[26]:

```
# Training set
train<-gen.data(N=100,mu1,mu2,mu3,Sigma1,Sigma2,Sigma3,p1=0.3,p2=0.2)
plot(train$X[,1],train$X[,2],col=train$y,pch=train$y)
```



```
[27]: # Test set
      test<-gen.data(N=1000,mu1,mu2,mu3,Sigma1,Sigma2,Sigma3,p1=0.3,p2=0.2)
```

```
[30]: # Q2-3
```

```
[28]: ypred<-knn(train$X,test$X,factor(train$y),k=5)
      table(test$y,ypred)
```

```
   ypred
      1   2   3
  1 220  22  70
  2  68 105  29
  3  53  12 421
```

[29]:

```
err<-mean(test$y != ypred)
print(err)
```

```
[1] 0.254
```

[31]:

```
# Q4
```

[32]:

```
M<-10
Kmax<-20
ERR100<-matrix(0,M,Kmax)
ERR500<-ERR100
```

[33]:

```
for(m in 1:M){
  print(m)
  train100<-gen.data(N=100,mu1,mu2,mu3,Sigma1,Sigma2,Sigma3,p1=0.3,p2=0.2)
  train500<-gen.data(N=500,mu1,mu2,mu3,Sigma1,Sigma2,Sigma3,p1=0.3,p2=0.2)
  for(k in 1:Kmax){
    ypred<-knn(train100$X,test$X,factor(train100$y),k=k)
    ERR100[m,k]<-mean(test$y != ypred)
    ypred<-knn(train500$X,test$X,factor(train500$y),k=k)
    ERR500[m,k]<-mean(test$y != ypred)
  }
}
```

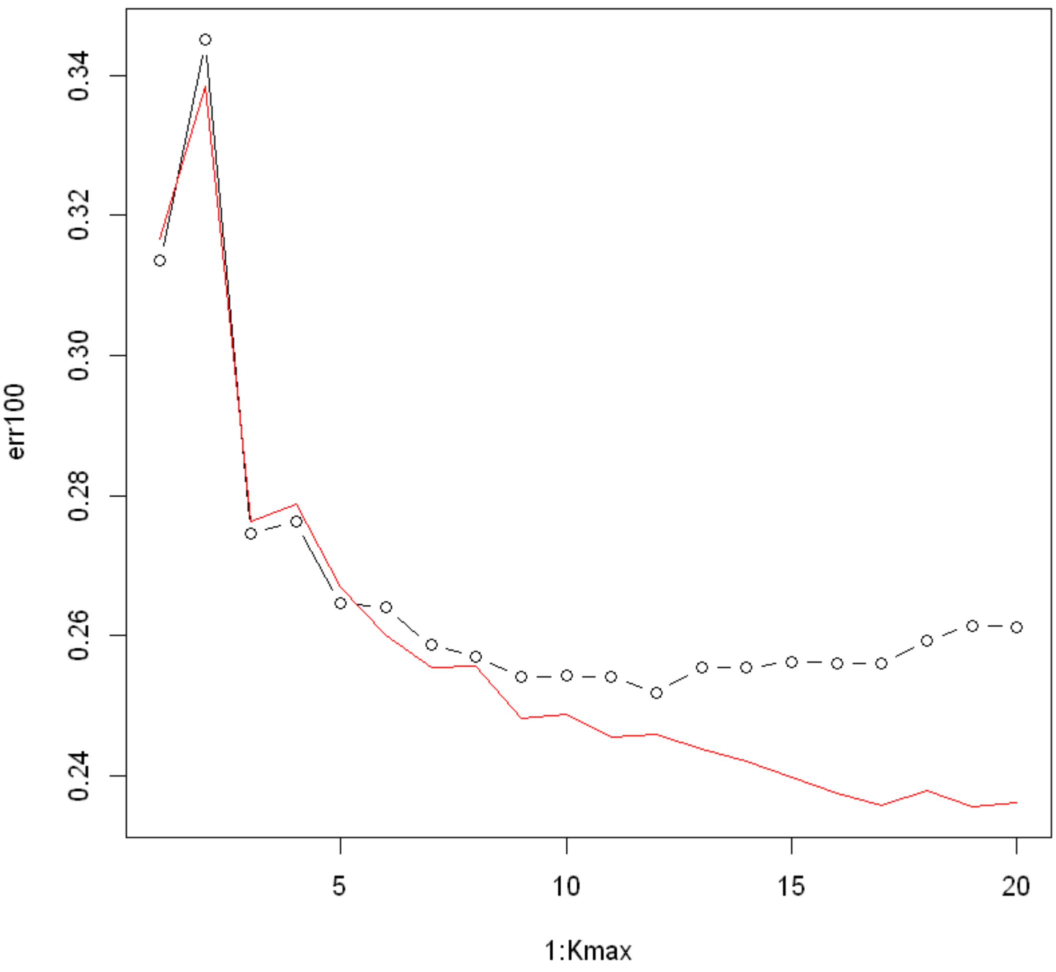
```
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
[1] 6
[1] 7
[1] 8
[1] 9
[1] 10
```

[34]:

```
err100<-colMeans(ERR100)
err500<-colMeans(ERR500)
```

[35]:

```
plot(1:Kmax,err100,type="b",ylim=range(err100,err500))
lines(1:Kmax,err500,col="red")
```


0.34
0.32
0.30
0.28
0.26
0.24
err100
5
10
15
20
1:Kmax