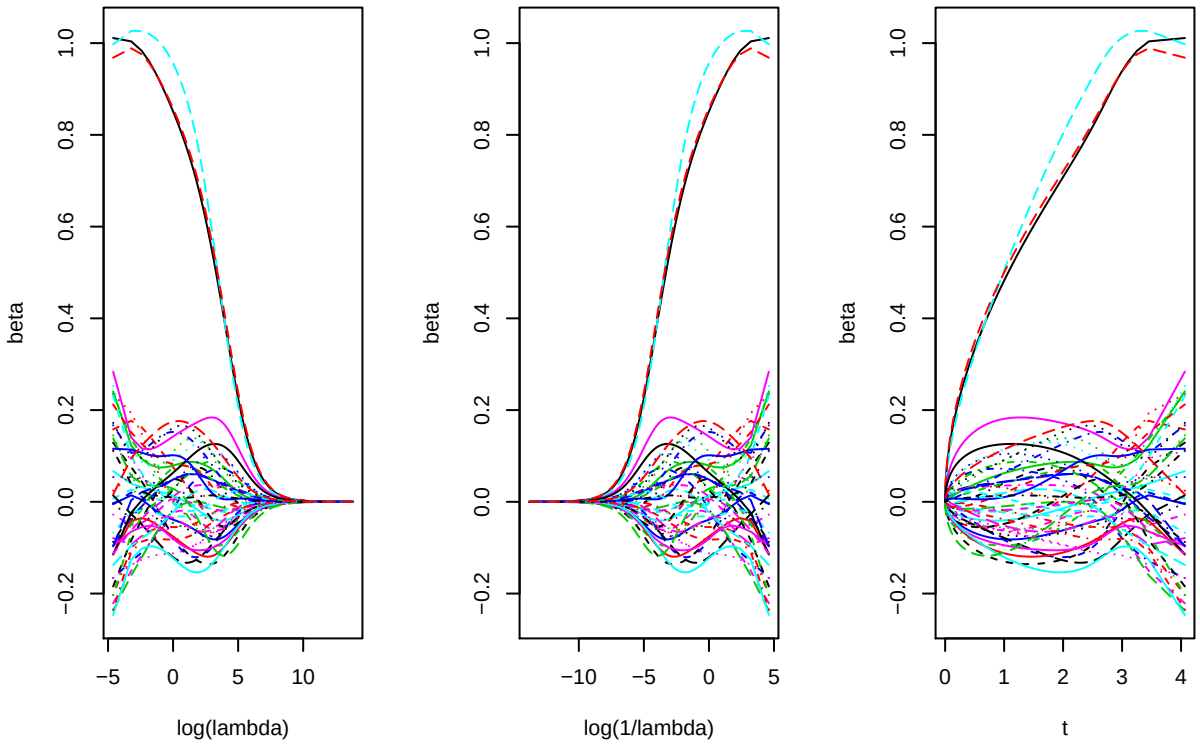
ST_5227_Tut_2_code

Name:Zhu Xu
User ID:E0337988
Student ID:A0131944H

2019/2/14

Q6:

```
data = read.csv("/Users/xuzhu/Desktop/Notes/Sem2/ST5227-Applied_Data_Mining/Tut/Tut2/mydataT02a.csv", h
x = data.matrix(data[, 1:50])
y = data.matrix(data[, 201])

#a)
p = 50
I = diag(p)
#lambda=0.1
b1 = solve(t(x)%*%x+0.1*I)%*%t(x)%*%y

#lambda=1
b2 = solve(t(x)%*%x+1*I)%*%t(x)%*%y

#lambda=10
b3 = solve(t(x)%*%x+10*I)%*%t(x)%*%y

#lambda=100
b4 = solve(t(x)%*%x+100*I)%*%t(x)%*%y

#lambda=1000
b5 = solve(t(x)%*%x+1000*I)%*%t(x)%*%y

lambda = matrix(0, 10000, 1)
beta = matrix(0, 10000, 50)
for (i in 1:10000){
  lambdai=i^2/100
  br=solve(t(x)%*%x+lambdai*I)%*%t(x)%*%y
  beta[i,]=br
  lambda[i]=lambdai
}
par(mfrow=c(1,3))
matplot(log(lambda), beta, type="l")
matplot(log(1/lambda), beta, type="l")
t = apply(beta^2, 1, sum)
matplot(t, beta, type="l")
```




```
#b)
n = length(y)
cv = matrix(0,100,1)
lambda = matrix(0,100,1)
for(i in 1:100){
  lambdai=i/20
  err=c()
  for(j in 1:n){
    xnew=x[-j,]
    ynew=y[-j,]
    betaj=solve(t(xnew)%*%xnew+lambdai*I)%*%t(xnew)%*%ynew
    err[j]=y[j,]-x[j,]%*%betaj
  }
  cv[i]=sum(err^2)/n
  lambda[i]=lambdai
}
bestlambda = lambda[which(cv==min(cv))]
Ridge = solve(t(x)%*%x+bestlambda*I)%*%t(x)%*%y
Ridge
```

```
##              [,1]
## V1   0.8662060873
## V2   0.0841500822
## V3   0.1024369882
## V4  -0.0263531756
## V5   0.9683085113
## V6   0.1381084502
## V7  -0.0649610032
## V8   0.0134194307
## V9   0.0218902270
```

```
## V10  0.0447492037
## V11  0.0155157889
## V12 -0.0302093376
## V13 -0.0436463972
## V14 -0.0550423638
## V15 -0.0259067525
## V16 -0.0364637545
## V17 -0.0022362941
## V18 -0.0685143238
## V19  0.0189721929
## V20  0.8717201141
## V21  0.0773086289
## V22 -0.0721520152
## V23  0.0873682199
## V24  0.0054173101
## V25 -0.1156470919
## V26 -0.0712003066
## V27  0.0689753596
## V28  0.0457691205
## V29 -0.0309568591
## V30 -0.0335321747
## V31  0.0579101119
## V32  0.0840609752
## V33 -0.0413807880
## V34  0.1517873585
## V35 -0.0294703747
## V36 -0.0747182471
## V37  0.0472901523
## V38  0.0036003027
## V39  0.0243900572
## V40  0.0448674817
## V41 -0.1174032093
## V42  0.0049808042
## V43  0.1570167818
## V44 -0.0822609684
## V45  0.0309613301
## V46  0.1012952924
## V47  0.0003184543
## V48 -0.1129817703
## V49  0.0101566955
## V50  0.1728811551
```

```
bestlambda
```

```
## [1] 0.8
```

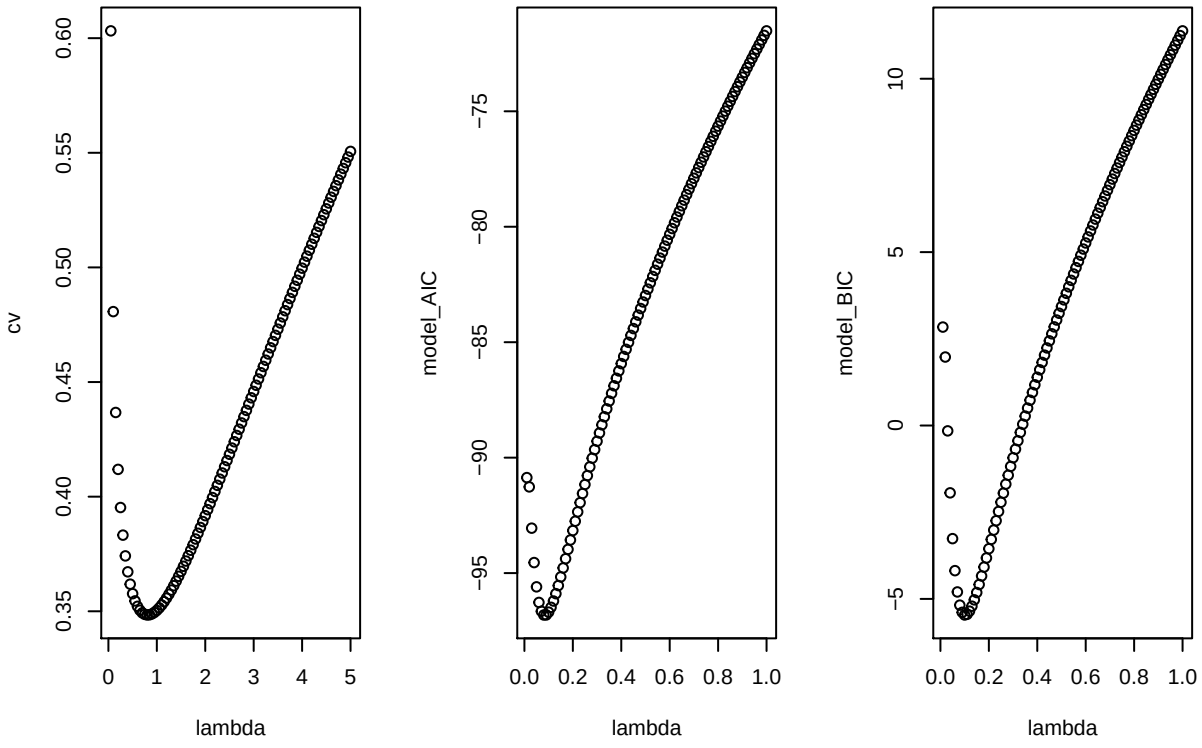
```
plot(lambda, cv)

#c)
newdata = read.csv("/Users/xuzhu/Desktop/Notes/Sem2/ST5227-Applied_Data_Mining/Tut/Tut2/mydataT02b.csv")
newx = as.matrix(newdata[,1:50])
newy = as.matrix(newdata[,201])
prey=newx%*%Ridge
errorRidge=mean((newy-prey)^2)
errorRidge
```

```
## [1] 0.3903278
```

```
#d)

lambda = matrix(0,100,1)
model_AIC=c()
model_BIC=c()
for (i in 1:100){
  lambdai=i/100
  br=solve(t(x)%*%x+lambdai*I)%*%t(x)%*%y
  residuals=y-x%*%br
  p=sum(diag(x%*%solve(t(x)%*%x+lambdai*I)%*%t(x)))
  model_AIC[i]=n*log(sum(residuals^2)/(n-p))+2*p
  model_BIC[i]=n*log(sum(residuals^2)/(n-p))+p*log(n)
  lambda[i]=lambdai
}
plot(lambda,model_AIC)
bestlambda_AIC = lambda[which(model_AIC==min(model_AIC))]
bestlambda_AIC
```

```
## [1] 0.08
```

```
plot(lambda,model_BIC)
```




```
bestlambda_BIC = lambda[which(model_BIC==min(model_BIC))]
bestlambda_BIC
```

```
## [1] 0.1
```

```
fold_cv = matrix(0,100,1)
lambda=matrix(0,100,1)
for(i in 1:100){
  lambdai=i/20
```

```
  err=c()
  for(j in 1:5){
    p=((j-1)*floor(n/5)+1):(j*floor(n/5))
    xnew=x[-p,]
    ynew=y[-p,]
    betaj=solve(t(xnew)%*%xnew+lambdai*I)%*%t(xnew)%*%ynew
    err[j]=mean((y[p,]-x[p,]%*%betaj)^2)
  }
  fold_cv[i]=sum(err)/5
  lambda[i]=lambdai
}
plot(lambda,fold_cv)
bestlambda_f = lambda[which(fold_cv==min(fold_cv))]
bestlambda_f
```

```
## [1] 0.6
```