489 Midterm Exam

Nicholas Thompson

10/8/2021

#Question 1  
set.seed(49)  
str(Boston)

## 'data.frame': 506 obs. of 14 variables:  
## $ crim : num 0.00632 0.02731 0.02729 0.03237 0.06905 ...  
## $ zn : num 18 0 0 0 0 0 12.5 12.5 12.5 12.5 ...  
## $ indus : num 2.31 7.07 7.07 2.18 2.18 2.18 7.87 7.87 7.87 7.87 ...  
## $ chas : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ nox : num 0.538 0.469 0.469 0.458 0.458 0.458 0.524 0.524 0.524 0.524 ...  
## $ rm : num 6.58 6.42 7.18 7 7.15 ...  
## $ age : num 65.2 78.9 61.1 45.8 54.2 58.7 66.6 96.1 100 85.9 ...  
## $ dis : num 4.09 4.97 4.97 6.06 6.06 ...  
## $ rad : int 1 2 2 3 3 3 5 5 5 5 ...  
## $ tax : num 296 242 242 222 222 222 311 311 311 311 ...  
## $ ptratio: num 15.3 17.8 17.8 18.7 18.7 18.7 15.2 15.2 15.2 15.2 ...  
## $ black : num 397 397 393 395 397 ...  
## $ lstat : num 4.98 9.14 4.03 2.94 5.33 ...  
## $ medv : num 24 21.6 34.7 33.4 36.2 28.7 22.9 27.1 16.5 18.9 ...

0.6\*nrow(Boston)

## [1] 303.6

attach(Boston)  
bos.rand=sample(1:nrow(Boston), 304)  
  
bos.train=Boston[bos.rand,]  
bos.test=Boston[-bos.rand,]  
  
#response is crim  
  
#a  
plot(indus, crim, data=bos.train)

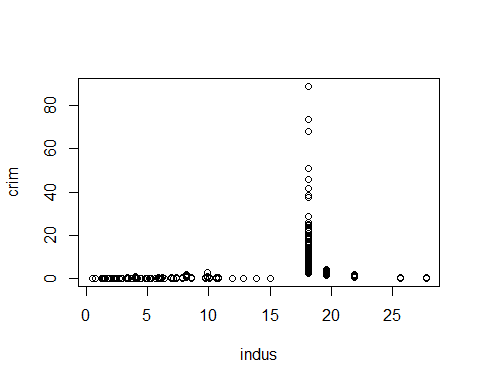
## Warning in plot.window(...): "data" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "data" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "data" is not a  
## graphical parameter  
  
## Warning in axis(side = side, at = at, labels = labels, ...): "data" is not a  
## graphical parameter

## Warning in box(...): "data" is not a graphical parameter

## Warning in title(...): "data" is not a graphical parameter



plot(zn, crim, data=bos.train)

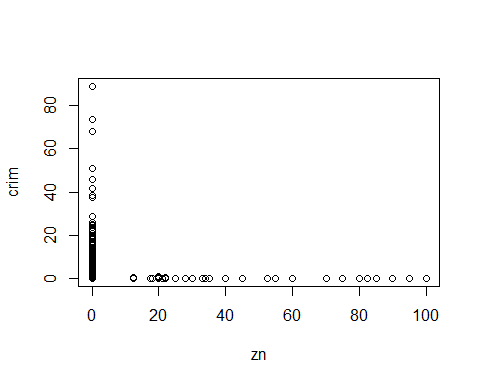
## Warning in plot.window(...): "data" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "data" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "data" is not a  
## graphical parameter  
  
## Warning in axis(side = side, at = at, labels = labels, ...): "data" is not a  
## graphical parameter

## Warning in box(...): "data" is not a graphical parameter

## Warning in title(...): "data" is not a graphical parameter



plot(black, crim, data= bos.train)

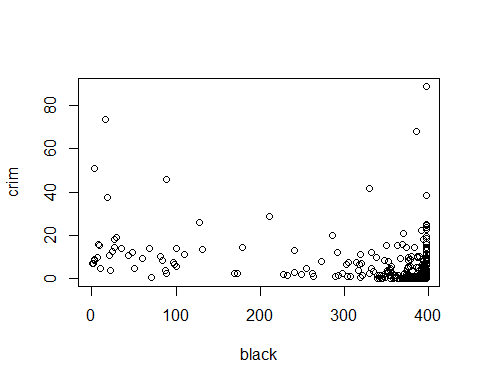
## Warning in plot.window(...): "data" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "data" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "data" is not a  
## graphical parameter  
  
## Warning in axis(side = side, at = at, labels = labels, ...): "data" is not a  
## graphical parameter

## Warning in box(...): "data" is not a graphical parameter

## Warning in title(...): "data" is not a graphical parameter



plot(medv, crim, data=bos.train)

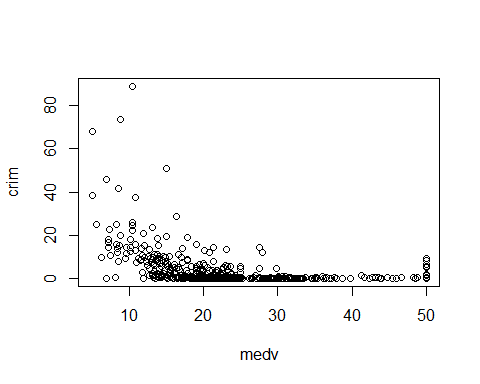
## Warning in plot.window(...): "data" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "data" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "data" is not a  
## graphical parameter  
  
## Warning in axis(side = side, at = at, labels = labels, ...): "data" is not a  
## graphical parameter

## Warning in box(...): "data" is not a graphical parameter

## Warning in title(...): "data" is not a graphical parameter



plot(dis, crim, data=bos.train)

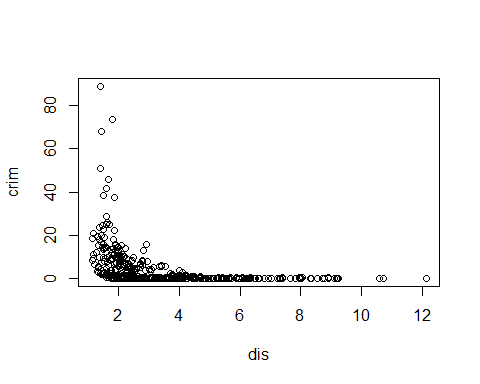
## Warning in plot.window(...): "data" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "data" is not a graphical parameter

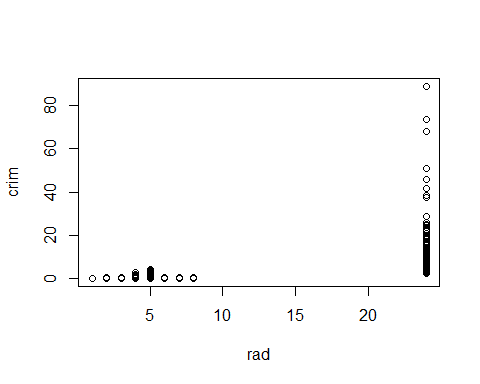
## Warning in axis(side = side, at = at, labels = labels, ...): "data" is not a  
## graphical parameter  
  
## Warning in axis(side = side, at = at, labels = labels, ...): "data" is not a  
## graphical parameter

## Warning in box(...): "data" is not a graphical parameter

## Warning in title(...): "data" is not a graphical parameter



plot(rad, crim)



plot(tax, crim, data=bos.train)

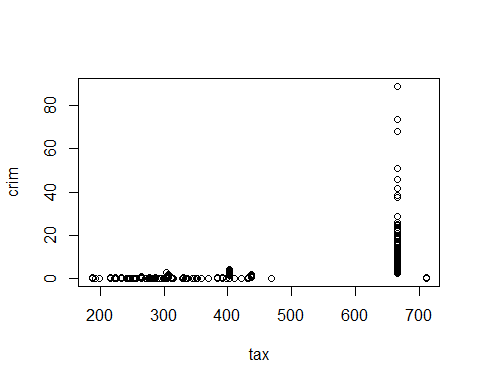
## Warning in plot.window(...): "data" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "data" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "data" is not a  
## graphical parameter  
  
## Warning in axis(side = side, at = at, labels = labels, ...): "data" is not a  
## graphical parameter

## Warning in box(...): "data" is not a graphical parameter

## Warning in title(...): "data" is not a graphical parameter



#Of the variables tested, none seem to have a linear relationship with crime rate. Further,  
#it seems that certain values of the predictors (like for example when tax about 670) lead to a varied   
#levels of crime rate. Of these variables plotted, I cannot confidently say that they share a relationship with crime rate  
  
  
fit1b=lm(crim~., data = bos.train)  
vif(fit1b)

## zn indus chas nox rm age dis rad   
## 2.239505 3.824030 1.095264 4.503442 3.009994 3.108173 4.212620 6.108067   
## tax ptratio black lstat medv   
## 7.629317 1.884900 1.620625 3.827348 4.835764

#appears to be multicollinearity issues for rad and tax (vif >5)  
cor(tax,crim)

## [1] 0.5827643

cor(rad,crim)

## [1] 0.6255051

#b  
summary(fit1b)

##   
## Call:  
## lm(formula = crim ~ ., data = bos.train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -11.435 -1.935 -0.489 1.021 56.300   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 29.613421 9.135736 3.241 0.00133 \*\*   
## zn 0.041000 0.021806 1.880 0.06109 .   
## indus -0.072769 0.098669 -0.738 0.46141   
## chas -0.616870 1.522317 -0.405 0.68562   
## nox -12.408783 6.627330 -1.872 0.06216 .   
## rm -1.749248 0.943861 -1.853 0.06486 .   
## age 0.023773 0.022157 1.073 0.28419   
## dis -0.791069 0.338626 -2.336 0.02017 \*   
## rad 0.585762 0.103838 5.641 4.01e-08 \*\*\*  
## tax -0.003092 0.005819 -0.531 0.59555   
## ptratio -0.265279 0.233442 -1.136 0.25674   
## black -0.013227 0.004697 -2.816 0.00519 \*\*   
## lstat 0.055651 0.098121 0.567 0.57104   
## medv -0.083833 0.088787 -0.944 0.34585   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 6.139 on 290 degrees of freedom  
## Multiple R-squared: 0.4745, Adjusted R-squared: 0.4509   
## F-statistic: 20.14 on 13 and 290 DF, p-value: < 2.2e-16

#based on the t-test at 0.05 level, the only significant predictors are dis, rad, and black.  
  
#c  
  
reg.best.subset=regsubsets(crim~.,data=bos.train, nvmax=14)  
summary(reg.best.subset)

## Subset selection object  
## Call: regsubsets.formula(crim ~ ., data = bos.train, nvmax = 14)  
## 13 Variables (and intercept)  
## Forced in Forced out  
## zn FALSE FALSE  
## indus FALSE FALSE  
## chas FALSE FALSE  
## nox FALSE FALSE  
## rm FALSE FALSE  
## age FALSE FALSE  
## dis FALSE FALSE  
## rad FALSE FALSE  
## tax FALSE FALSE  
## ptratio FALSE FALSE  
## black FALSE FALSE  
## lstat FALSE FALSE  
## medv FALSE FALSE  
## 1 subsets of each size up to 13  
## Selection Algorithm: exhaustive  
## zn indus chas nox rm age dis rad tax ptratio black lstat medv  
## 1 ( 1 ) " " " " " " " " " " " " " " "\*" " " " " " " " " " "   
## 2 ( 1 ) " " " " " " " " " " " " " " "\*" " " " " " " "\*" " "   
## 3 ( 1 ) " " " " " " " " "\*" " " " " "\*" " " " " "\*" " " " "   
## 4 ( 1 ) " " " " " " " " "\*" " " " " "\*" " " " " "\*" "\*" " "   
## 5 ( 1 ) "\*" " " " " " " "\*" " " "\*" "\*" " " " " "\*" " " " "   
## 6 ( 1 ) "\*" " " " " "\*" "\*" " " "\*" "\*" " " " " "\*" " " " "   
## 7 ( 1 ) "\*" " " " " "\*" "\*" "\*" "\*" "\*" " " " " "\*" " " " "   
## 8 ( 1 ) "\*" "\*" " " "\*" "\*" "\*" "\*" "\*" " " " " "\*" " " " "   
## 9 ( 1 ) "\*" "\*" " " "\*" "\*" "\*" "\*" "\*" " " " " "\*" "\*" " "   
## 10 ( 1 ) "\*" "\*" " " "\*" "\*" "\*" "\*" "\*" " " "\*" "\*" " " "\*"   
## 11 ( 1 ) "\*" "\*" " " "\*" "\*" "\*" "\*" "\*" " " "\*" "\*" "\*" "\*"   
## 12 ( 1 ) "\*" "\*" " " "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*"   
## 13 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*"

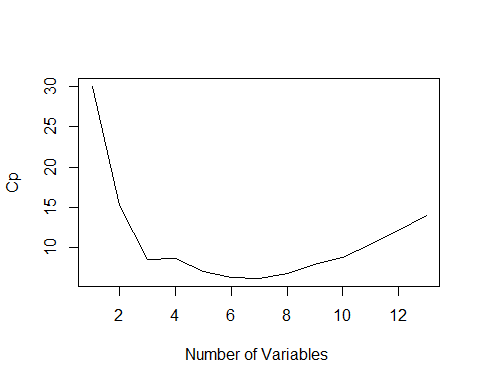
cbind(  
 summary(reg.best.subset)$cp,  
 summary(reg.best.subset)$rsq  
)

## [,1] [,2]  
## [1,] 29.986292 0.4019998  
## [2,] 15.257727 0.4323153  
## [3,] 8.604659 0.4479963  
## [4,] 8.639242 0.4515580  
## [5,] 7.084326 0.4580002  
## [6,] 6.385003 0.4628920  
## [7,] 6.236153 0.4667861  
## [8,] 6.887719 0.4692297  
## [9,] 8.007175 0.4708254  
## [10,] 8.799601 0.4730138  
## [11,] 10.406263 0.4737266  
## [12,] 12.164201 0.4741653  
## [13,] 14.000000 0.4744628

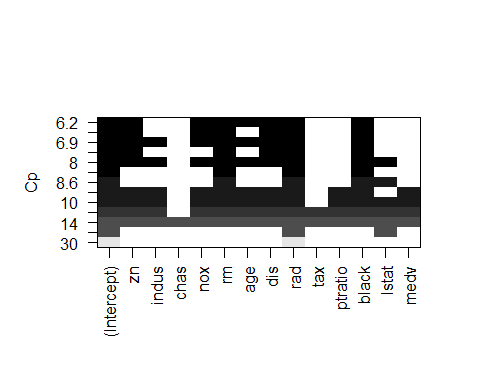
min(summary(reg.best.subset)$cp)

## [1] 6.236153

plot(summary(reg.best.subset)$cp ,xlab="Number of Variables ",ylab="Cp", type="l")



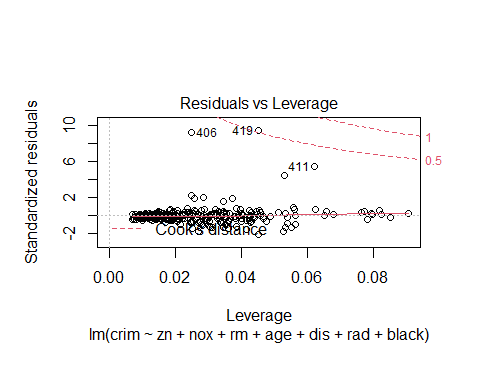
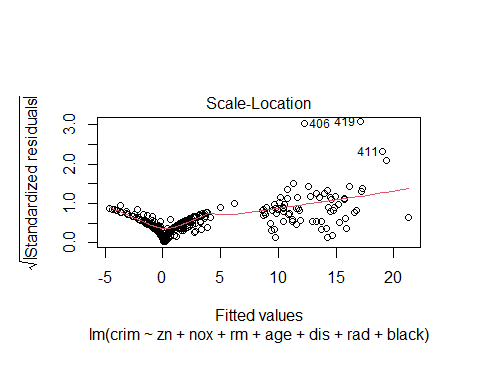
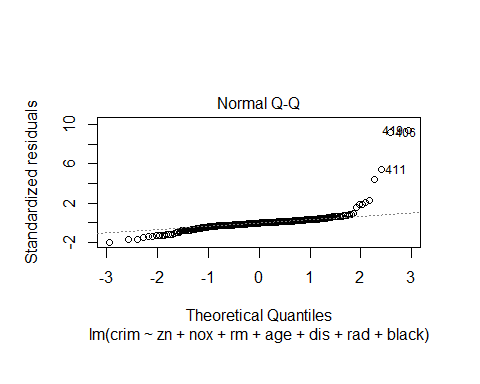
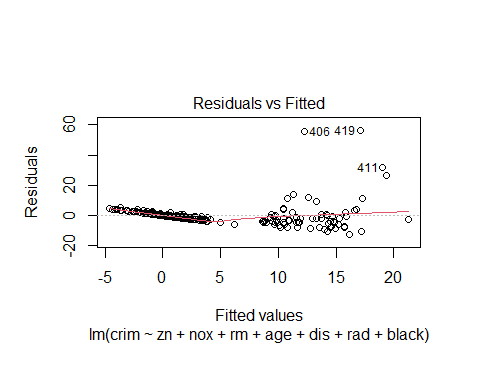
plot(reg.best.subset,scale="Cp")



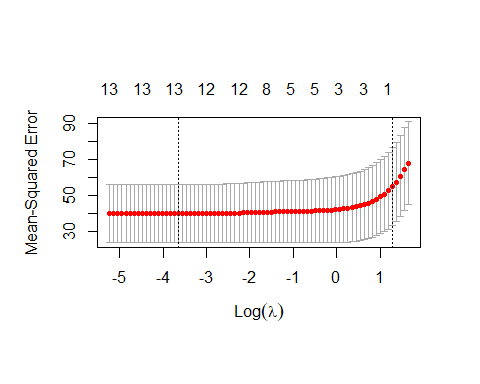
#The model with the minimum Cp value is the model with 7 predictors  
#predictors in this model are: zn, nox, rm, age, dis, rad, and black  
  
#d  
fit1d=lm(crim~zn+nox+rm+age+dis+rad+black, data = bos.train)  
coef(summary(fit1d))

## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 25.62979858 6.172216259 4.152447 4.306849e-05  
## zn 0.04607786 0.020037477 2.299584 2.216773e-02  
## nox -12.23777359 6.026644689 -2.030611 4.318812e-02  
## rm -2.40072256 0.593736774 -4.043412 6.724262e-05  
## age 0.02949649 0.020061780 1.470283 1.425473e-01  
## dis -0.63888320 0.305722182 -2.089751 3.749359e-02  
## rad 0.52208337 0.060107993 8.685756 2.592777e-16  
## black -0.01502153 0.004390948 -3.421022 7.116234e-04

#Median value of owner-occupied homes, or medv, has no effect on the per capita   
#crime rate since it is not a predictor used in the model from the best subset selection.  
  
#e  
#from the best subset selection matrix created in 1c: R-square=0.4667861  
#46% of the variation in crime rate can be explained by the 7 predictors  
plot(fit1d)



#based on the diagnostic plots, a recommendation would be to remove outlier rows from the training data  
#Another way to improve the model fit would be to add interaction terms or quadratic terms  
#We may want to try a LASSO regression  
  
#f  
fitfx=model.matrix(crim~.,data=bos.train)[,-1] #remove intercept  
fitfy=bos.train$crim  
cv.lasso=cv.glmnet(fitfx,fitfy,alpha=1)  
plot(cv.lasso)



best\_lambda=cv.lasso$lambda.min  
lasso=glmnet(fitfx,fitfy,alpha=1,lambda=best\_lambda, standardize=T)   
coef(lasso)

## 14 x 1 sparse Matrix of class "dgCMatrix"  
## s0  
## (Intercept) 26.478946629  
## zn 0.036711395  
## indus -0.075335336  
## chas -0.483947579  
## nox -10.499152552  
## rm -1.685911792  
## age 0.020533883  
## dis -0.700937755  
## rad 0.547722523  
## tax -0.001259889  
## ptratio -0.223173456  
## black -0.013118141  
## lstat 0.062901513  
## medv -0.068908350

#the lasso removed the tax predictor  
  
#g  
x=model.matrix(crim~., data = Boston)[,-1]  
  
  
lasso.pred=predict(lasso, newx=x[bos.rand,])  
y=Boston$crim  
mean((lasso.pred - y[-bos.rand]))#test error rate

## Warning in lasso.pred - y[-bos.rand]: longer object length is not a multiple of  
## shorter object length

## Warning in lasso.pred - y[-bos.rand]: longer object length is not a multiple of  
## shorter object length

## [1] 0.3747655

#0.3747655  
  
bss.pred=predict(fit1d,newx= x[bos.rand,])  
mean(bss.pred - bos.test[,1])

## Warning in bss.pred - bos.test[, 1]: longer object length is not a multiple of  
## shorter object length

## Warning in bss.pred - bos.test[, 1]: longer object length is not a multiple of  
## shorter object length

## [1] 0.3747655

#0.3747655  
#based on the test error rate, the models produce the same test error.

#Question 2  
  
m=median(crim)  
crime.level=I(crim>m)  
  
c1=crime.level[bos.rand]  
c2=crime.level[-bos.rand]  
train2=bos.train[,-1]  
test2=bos.test[,-1]  
train2=mutate(train2,crime\_level=c1)  
test2=mutate(test2,crime\_level=c2)  
#a  
reg.bwd=regsubsets(crime\_level~.,data=train2, nvmax=14, method="backward")  
summary(reg.bwd)

## Subset selection object  
## Call: regsubsets.formula(crime\_level ~ ., data = train2, nvmax = 14,   
## method = "backward")  
## 13 Variables (and intercept)  
## Forced in Forced out  
## zn FALSE FALSE  
## indus FALSE FALSE  
## chas FALSE FALSE  
## nox FALSE FALSE  
## rm FALSE FALSE  
## age FALSE FALSE  
## dis FALSE FALSE  
## rad FALSE FALSE  
## tax FALSE FALSE  
## ptratio FALSE FALSE  
## black FALSE FALSE  
## lstat FALSE FALSE  
## medv FALSE FALSE  
## 1 subsets of each size up to 13  
## Selection Algorithm: backward  
## zn indus chas nox rm age dis rad tax ptratio black lstat medv  
## 1 ( 1 ) " " " " " " "\*" " " " " " " " " " " " " " " " " " "   
## 2 ( 1 ) " " " " " " "\*" " " " " " " "\*" " " " " " " " " " "   
## 3 ( 1 ) " " " " " " "\*" " " "\*" " " "\*" " " " " " " " " " "   
## 4 ( 1 ) " " " " " " "\*" " " "\*" " " "\*" " " " " " " " " "\*"   
## 5 ( 1 ) " " " " " " "\*" " " "\*" " " "\*" " " "\*" " " " " "\*"   
## 6 ( 1 ) " " " " " " "\*" " " "\*" " " "\*" " " "\*" "\*" " " "\*"   
## 7 ( 1 ) "\*" " " " " "\*" " " "\*" " " "\*" " " "\*" "\*" " " "\*"   
## 8 ( 1 ) "\*" "\*" " " "\*" " " "\*" " " "\*" " " "\*" "\*" " " "\*"   
## 9 ( 1 ) "\*" "\*" " " "\*" " " "\*" " " "\*" "\*" "\*" "\*" " " "\*"   
## 10 ( 1 ) "\*" "\*" "\*" "\*" " " "\*" " " "\*" "\*" "\*" "\*" " " "\*"   
## 11 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" " " "\*" "\*" "\*" "\*" " " "\*"   
## 12 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" " " "\*" "\*" "\*" "\*" "\*" "\*"   
## 13 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*"

summary(reg.bwd)$bic

## [1] -215.5482 -232.5801 -236.9997 -239.6684 -236.7087 -232.3328 -227.9345  
## [8] -223.2067 -217.8120 -212.1971 -206.4922 -200.7885 -195.0716

min(summary(reg.bwd)$bic)

## [1] -239.6684

#4th model  
model2a=glm(crime\_level~nox+age+rad+medv, data = train2, family = binomial)  
summary(model2a)

##   
## Call:  
## glm(formula = crime\_level ~ nox + age + rad + medv, family = binomial,   
## data = train2)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.70549 -0.35762 -0.10854 0.01047 2.24020   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -18.12289 2.62520 -6.903 5.08e-12 \*\*\*  
## nox 25.42988 4.72715 5.380 7.47e-08 \*\*\*  
## age 0.01396 0.01039 1.343 0.179145   
## rad 0.47911 0.13191 3.632 0.000281 \*\*\*  
## medv 0.03692 0.02828 1.305 0.191785   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 420.37 on 303 degrees of freedom  
## Residual deviance: 163.07 on 299 degrees of freedom  
## AIC: 173.07  
##   
## Number of Fisher Scoring iterations: 8

#from this model, nox and rad are statistically significant  
  
#b  
bs =function(formula, data, indices) {   
 d =data[indices,]  
fit = glm(formula, data=d, family = "binomial")  
return(coef(fit))   
}  
res=boot(data=train2, statistic=bs,  
 R=500, formula=crime\_level~nox+age+rad+medv)  
summary(res)

##   
## Number of bootstrap replications R = 500   
## original bootBias bootSE bootMed  
## 1 -18.122894 -0.56808543 2.258443 -18.421385  
## 2 25.429882 0.83245860 4.742068 25.926686  
## 3 0.013964 0.00080058 0.012088 0.014481  
## 4 0.479106 0.01520379 0.108451 0.494600  
## 5 0.036918 -0.00049099 0.023049 0.037561

boot.ci(res, index = 5, type = "norm")#CI for medv

## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS  
## Based on 500 bootstrap replicates  
##   
## CALL :   
## boot.ci(boot.out = res, type = "norm", index = 5)  
##   
## Intervals :   
## Level Normal   
## 95% (-0.0078, 0.0826 )   
## Calculations and Intervals on Original Scale

#We estimate that the true coefficient for median value of owner-occupied homes is between the interval [-0.0111,0.0835]  
boot.ci(res, index = 3, type = "norm")#CI for age

## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS  
## Based on 500 bootstrap replicates  
##   
## CALL :   
## boot.ci(boot.out = res, type = "norm", index = 3)  
##   
## Intervals :   
## Level Normal   
## 95% (-0.0105, 0.0369 )   
## Calculations and Intervals on Original Scale

boot.ci(res, index=4,type = "norm")#CI for rad

## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS  
## Based on 500 bootstrap replicates  
##   
## CALL :   
## boot.ci(boot.out = res, type = "norm", index = 4)  
##   
## Intervals :   
## Level Normal   
## 95% ( 0.2513, 0.6765 )   
## Calculations and Intervals on Original Scale

boot.ci(res, index = 2, type = "norm")#CI for nox

## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS  
## Based on 500 bootstrap replicates  
##   
## CALL :   
## boot.ci(boot.out = res, type = "norm", index = 2)  
##   
## Intervals :   
## Level Normal   
## 95% (15.30, 33.89 )   
## Calculations and Intervals on Original Scale

#c  
logreg.probs=predict(model2a,test2)  
logreg.pred= rep(1, length(logreg.probs))  
logreg.pred[logreg.probs < 0.5] = 0  
table(logreg.pred, test2$crime\_level)

##   
## logreg.pred FALSE TRUE  
## 0 87 16  
## 1 5 94

mean(logreg.pred != test2$crime\_level)

## [1] 0.1039604

lda2c=lda(crime\_level~nox+age+rad+medv, data = train2, subset = bos.rand)  
lda2c

## Call:  
## lda(crime\_level ~ nox + age + rad + medv, data = train2, subset = bos.rand)  
##   
## Prior probabilities of groups:  
## FALSE TRUE   
## 0.5215054 0.4784946   
##   
## Group means:  
## nox age rad medv  
## FALSE 0.4743619 54.33093 4.28866 24.97216  
## TRUE 0.6342472 82.90225 13.67416 20.91910  
##   
## Coefficients of linear discriminants:  
## LD1  
## nox 8.24661059  
## age 0.01118235  
## rad 0.07753323  
## medv 0.03414159

lda.pred=predict(lda2c,test2)$class  
table(lda.pred, test2$crime\_level)

##   
## lda.pred FALSE TRUE  
## FALSE 91 22  
## TRUE 1 88

mean(lda.pred != test2$crime\_level)

## [1] 0.1138614

#logistic regression test error is about a tenth smaller

boston=Boston[,-1]  
boston=mutate(boston,crime.level)  
model3=glm(crime.level~.,data = boston)  
kfoldcv.err1=cv.glm(boston,model3,K=10)$delta[1]  
kfoldcv.err1

## [1] 0.1049615

cv.glm(boston,model3,K=9)$delta[1]

## [1] 0.1039871

cv.glm(boston,model3,K=11)$delta[1]

## [1] 0.102858

cv.glm(boston,model3,K=15)$delta[1]

## [1] 0.1046707

cv.glm(boston,model3,K=25)$delta[1]

## [1] 0.1035838

cv.glm(boston,model3,K=16)$delta[1]

## [1] 0.1037329

#by k-fold cross validation, we have determined best K is 15  
  
cly=boston$crime.level  
knn.pred=knn(boston,boston,cly, k=15)  
table(knn.pred, boston$crime.level)

##   
## knn.pred FALSE TRUE  
## FALSE 226 24  
## TRUE 27 229

mean(knn.pred != boston$crime.level)

## [1] 0.1007905

#error rate was 10.07%