STSCI 4740 Final Project

Ben Liu (BL652), Yang Shen (YS773), Xihan Peng (XP49)

Department of Statistical Science, Cornell University

Abstract

In this report, we are going to identify important genes and investigate their relationships with the target gene Mapk1. First, we fit a multiple linear regression model. Independence and normality test are made at the beginning. Best subset selection and forward selection are used to choose the best variables for linear regression. Next, we apply shrinkage methods to perform variable selection and compare the mean squared test error with ridge regression. We also build a simple decision tree to enhance model interpretability. Bagging and random forests are applied to improve predictive performance. Last, we compare all possible models, and find that the linear model performs better than others based on the test MSE. Since the linear model is quite simple and interpretable, we decide to choose the linear model with 4 predictors: Akt2, Rik, Pik3r3 and Rac1 as our best model.

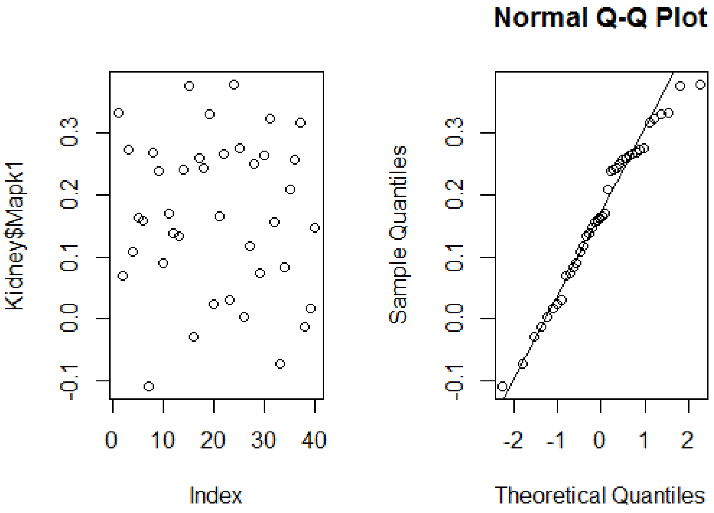
Keywords: Linear Model, Shrinkage Methods, Random Forest, Test MSE

1. Data Preparation

To identify important genes and investigate their relationships with the target gene Mapk1, we first transpose the original kidney dataset in R so that we have 24 predictors (Gene.Name) with 40 observations.

2. Linear Model

Before fitting linear regression, we first test if the response values are independent and if they satisfy the normality assumption. The values of Mapk1 and the Q-Q plot of Mapk1 against normal distribution are plotted. The results are displayed in the following graphs.

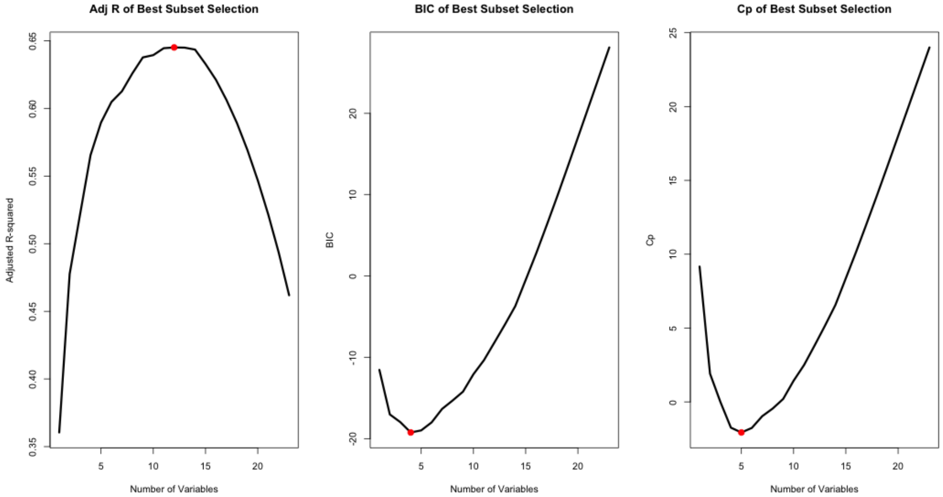


Preprint submitted to Data Mining and Machine Learning *December 2, 2016*

2.1. ***Best Subset Selection***

To select important variables for the linear regression, best subset selection is a valid option.We first apply the “regsubset” function to the dataset and use Mapk1 as the response. We summarize the best subset selection procedure by considering the number of predictors that makes the regression have the lowest BIC, C_p value and the highest adjusted R^2.

By looking at the graph below, we got 4 predictors based on BIC criterion, and they are Akt2, Rik, Pik3r3, Rac1. C_p gives us 5 predictors, Akt2, Rik, Pik3r3, Rac1 and Pik3r1. The largest adjusted R^2 value suggests 12 variables. But if we train a linear model with 12 variables, many of them will become insignificant. Therefore, we don’t choose predictors based on adjusted R^2.



2.2. ***Forward Selection***

Apart from best subset selection, forward and backward selection are also applied as alternative methods to find the best predictors. Applying forward selection, we get different numbers of predictors based on different criteria. The minimum value of BIC suggests 4 predictors for the model, and they are exactly the same with the result from best subset selection. The minimum value of C_p suggests 5 predictors, which are also exactly the same with the previous result. The maximum adjusted R^2 suggests 12 predictors. We won’t use 12 predictors because many of them will be insignificant in the model.

The backward selection is not used in this case, because the number of observations is close to the number of predictors, which will not produce robust results for the method.

2.3. ***Linear Regression Diagnostics***

Based on the selection results from the previous section, there are a total of 4 possible linear regression models.

Model 1 with 4 predictors[1]:

$$\text{Mapk1} = -0.445 - 0.405 \times \text{Akt2}^{*} + 0.221 \times \text{Rik}^{*} + 0.244 \times \text{Pik3r3}^{*} + 0.317 \times \text{Rac1}^{***}$$

Model 2 with 5 predictors:

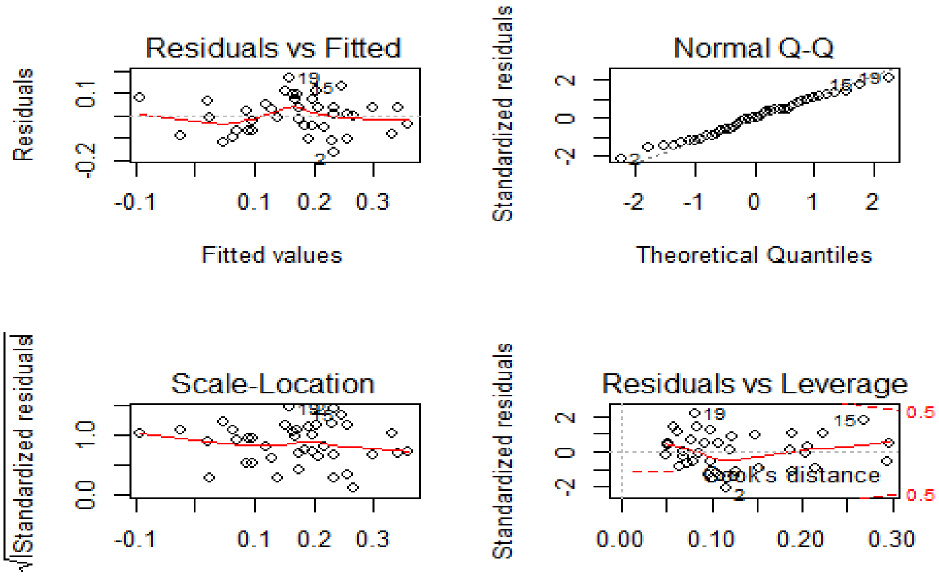
$$\text{Mapk1} = -0.618 - 0.397 \times \text{Akt2}^{*} + 0.250 \times \text{Rik}^{*} + 0.224 \times \text{Pik3r3}^{\cdot} + 0.330 \times \text{Rac1}^{***} - 0.162 \times \text{Pik3r1}^{\cdot}$$

[1]”∗” means parameter is significant at 0.05 level, ” ·” means parameter is significant at 0.1 level

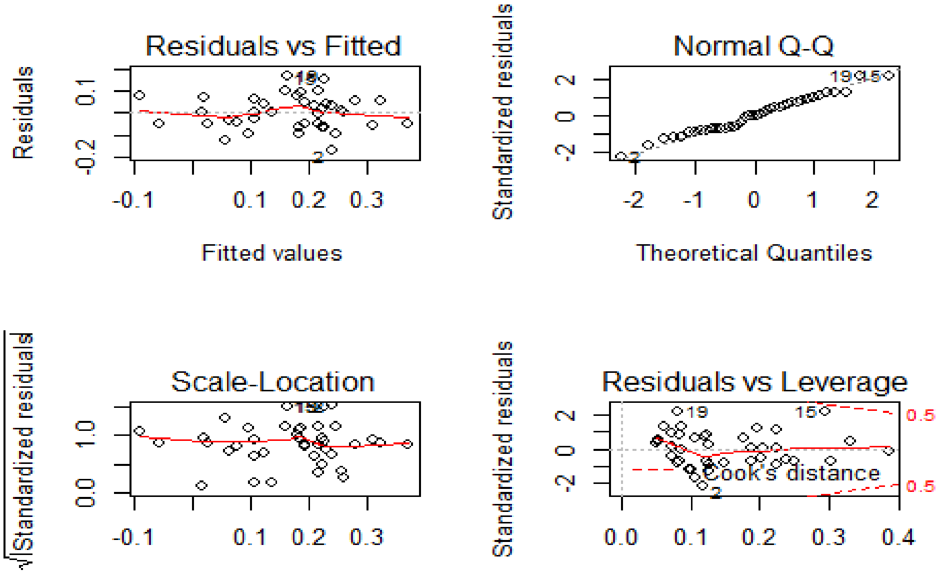
Predictors in the first two models are all significant, and for the other two models with 12 predictors, many predictors are insignificant. Therefore only Model 1 and Model 2 are kept for further comparison.

The residual plot, Q-Q plot, standardized residual plot and leverage plot for both models are shown below:

Model 1



Model 2



From both plots, the residuals are randomly distributed around 0. Most of the residuals in Q-Q plot fall on the standard theoretical quantile line. The fitted line of standardized residuals shows no significant pattern related to fitted values. And finally, all the points have Cook's distance that are less than 0.5. Therefore, diagnostics for these two regression show good results, which means the two regression models are both plausible for further evaluation.

3. Shrinkage Methods

*3.1. **Lasso and Ridge Regression***

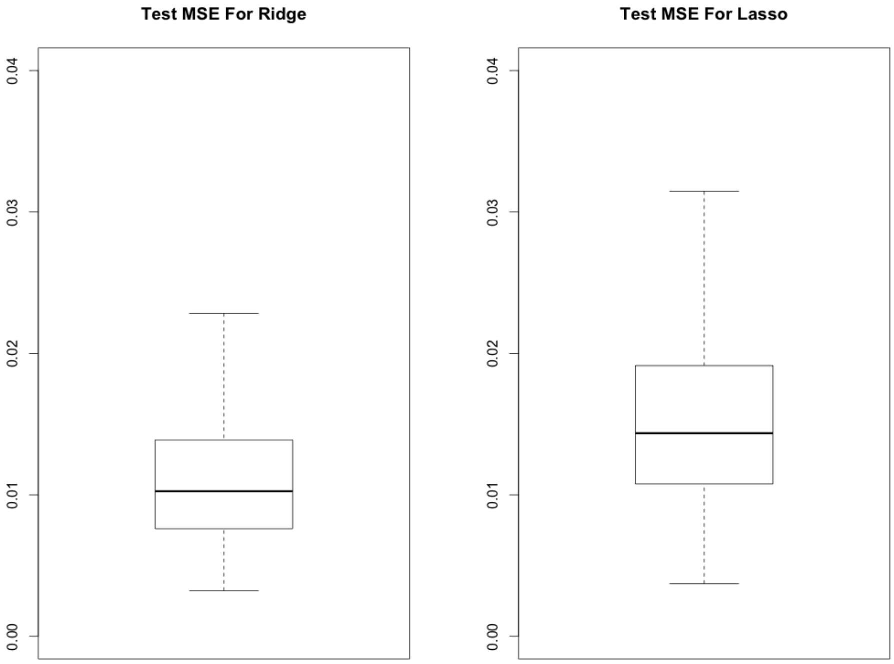
In this section, we use lasso at first to perform variable selection. Recall that the lasso coefficients, $\hat{\beta}^L_\lambda$, minimize the following equation

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j|$$

Compared with ridge regression, the lasso has the effect of shrinking some of coefficients estimated to be exactly to zero when λ is very large. Hence, much like best subset selection, the lasso performs variable selection.

```
> coef.lasso[coef.lasso!=0]
(Intercept)        Akt2       Plcg2         Rik      Pik3cd
-0.21075215 -0.05789074  0.01200671  0.12095833 -0.03413748
     Pik3r3        Rac1       Nfat5
 0.17018699  0.15622264  0.06521329
```

From the output above, the seven variables selected by lasso are Akt2, Plcg2, Rik, Pik3d, Pik3r3, Rac1 and Nfat5. We can notice that some predictors are also identified using best subset selection and forward selection. Similarly, we perform ridge regression. 50 randoms seeds are set and 80% of the original data is split into training set with remaining going to test set before applying cross validation. The MSE boxplots are as follows.



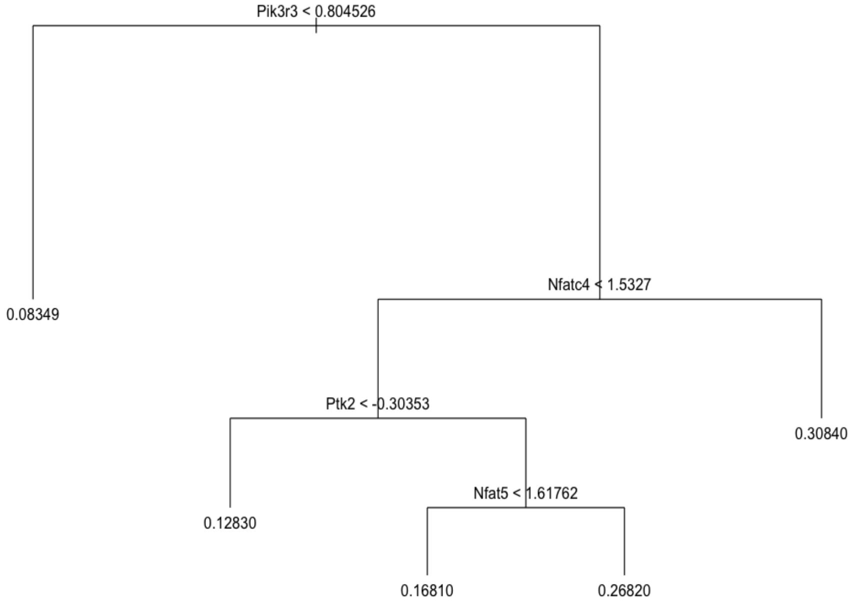
In this case, we can see that ridge performs better lasso, but ridge regression coefficient estimates solve the problem

$$\min_{\beta} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \quad \text{subject to} \quad \sum_{j=1}^{p} \beta_j^2 \leq s$$

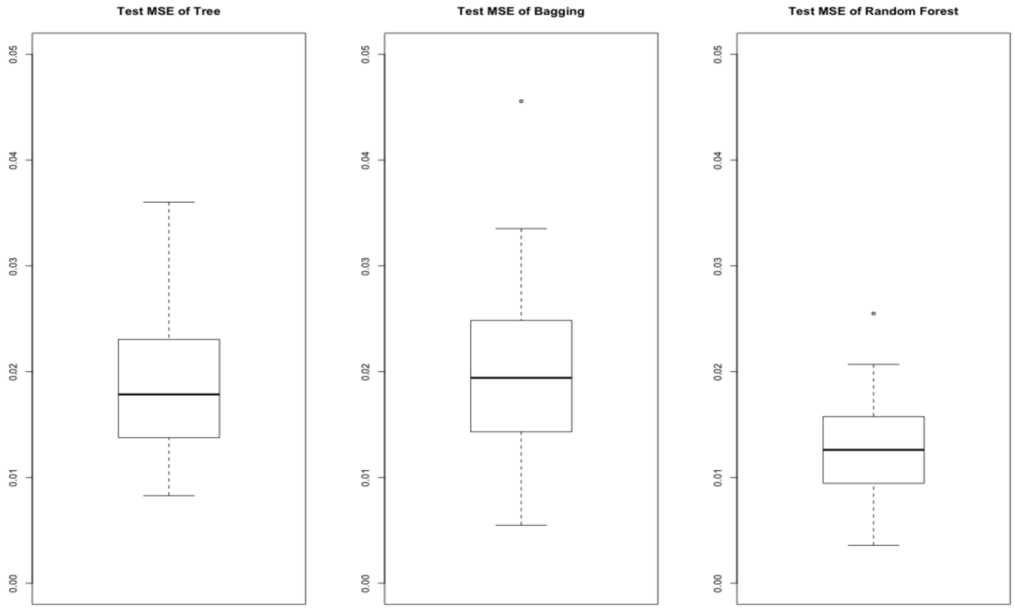
The solution for ridge regression is the joint point of a circle and an ellipse, which doesn't have the chance to force coefficients to be 0. To keep the balance between model simplicity and accuracy, we still choose the lasso model.

4. Decision Tree, Bagging and Random Forest

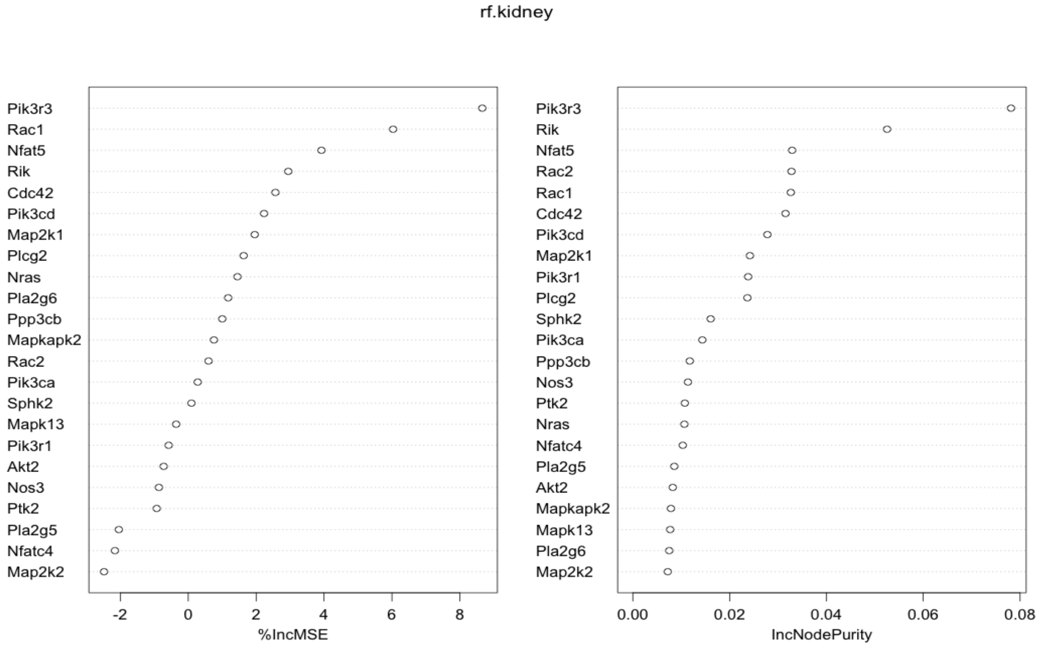
Since trees are intuitively clear to explain the model to people without statistical background, and can be graphically displayed, we train several tree models and compare their prediction performance of different models using bagging and random forests methods.



The simple decision tree only selects four important variables. It suggests that if Pik3r3 is the most important factor in determining Mapk1, and Nfatc4 with higher values lead to higher values of Mapk1. Given that Pik3r3 is smaller, then Nfatc4 seems to play little role in deciding Mapk1. However, trees generally do not have the same level of predictive accuracy, so we try to aggregate many decision trees and use bagging and random forest to improve our model accuracy. The box plots of test MSE for all three models are shown below.



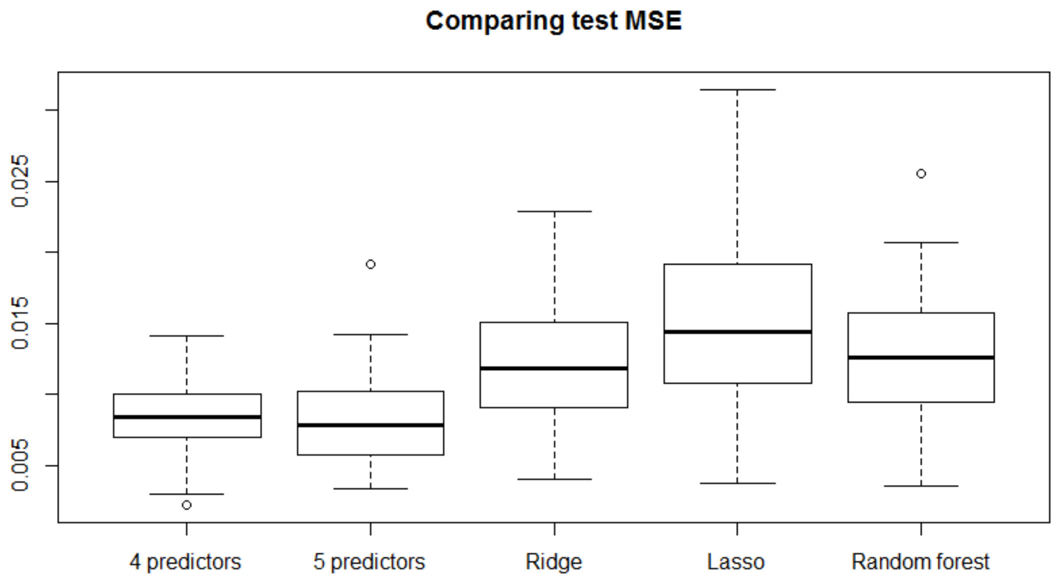
From above, we can find that the random forest performs much better than bagging and simple decision trees. As a result, we plot the importance of each variable using the random forest model with the smallest MSE.



The top five most important variables are Pik3r3, Nfat5, Rac1, Rik, Cdc42. Three of them are also included in the best subset selection of five variables, which suggests us to compare random forest with the previous two linear regression models.

5. Model Comparison and Justification

In this section, the test MSE of two linear regression models with 4 and 5 predictors, ridge regression, lasso regression and random forest will be used to assess which model is the best. We can get the following graph:



From the graph, we can see that both linear regression models have lower test MSE. While random forest and lasso regression have many applications, they don't perform as good as linear regression for this dataset. And for the choice between the two linear models, while the median test MSE with 5 predictors is slightly lower than that with 4 predictors, the distribution of the test MSE is more scattered. Besides, with 5 predictors, not all of them are significant at 5% level. Therefore, to get a more stable model, we prefer the model with 4 predictors.

6. Conclusion

The **final model** to predict the expression of Mapk1 is:

$$\text{Mapk1} = -0.445 - 0.405 \times \text{Akt2}^{*} + 0.221 \times \text{Rik}^{*} + 0.244 \times \text{Pik3r3}^{*} + 0.317 \times \text{Rac1}^{***}$$

Based on the model, we can say that the expression of Akt2 will suppress the expression of Mapk1, while the expression of Rik, Pik3r3 and Rac1 will boost the expression of Mapk1.

References

[1] G. James, D. Witten, T. Hastie, R. Tibshirani: An Introduction to Statistical Learning, ISSN 1431-875X

[2] T.Hastie, R. Tibshirani, J. Friedman: The Elements of Statistical Learning, ISSN 0172-7397

Appendices

R Code

```
kidney=read.csv("Kidney.csv",header = TRUE)
# Transpose the dataset kidney
kidney=t(kidney)
colnames(kidney) = as.matrix(kidney[1, ])
kidney =kidney[-1, ]
# Transform the dataset into dataframe
kidney=data.frame(kidney)
# Turn factor variable into numeric
for(i in 1:24){
kidney[,i]=as.numeric(levels(kidney[,i]))[kidney[,i]]
}

cor(kidney[,1:24])
#variables are not strongly correlated.

#check normality for regression
qqnorm(kidney$Mapk1)
qqline(kidney$Mapk1)

shapiro.test(kidney$Mapk1)
#Shapiro-Wilk’s test p-value is greater than 0.05, so we don’t reject the null hypothesi


#Best Subset Selection
library(leaps)
regfit=regsubsets(Mapk1˜.,data=kidney,nvmax=23)
reg.summary=summary(regfit)
# Find the largest adjusted R-squared
which.max(reg.summary$adjr2)
# Find the smallest BIC
which.min(reg.summary$bic)
# Find the smallest Cp
which.min(reg.summary$cp)
# Plot adjusted R-squared and BIC
par(mfrow=c(1,3))

plot(reg.summary$adjr2,xlab="Number of Variables",ylab="Adjusted R-squared",
type="l",main="Adj R of Best Subset Selection",lwd=3)
points(12,reg.summary$adjr2[12],col="red",cex=2,pch=20)

plot(reg.summary$bic,xlab="Number of Variables",ylab="BIC",type="l",
main="BIC of Best Subset Selection",lwd=3)
points(4,reg.summary$bic[4],col="red",cex=2,pch=20)
```

```
plot(reg.summary$cp,xlab="Number of Variables",ylab="Cp",type="l"
,main="Cp of Best Subset Selection",lwd=3)
points(5,reg.summary$cp[5],col="red",cex=2,pch=20)
par(mfrow=c(1,1))


#Forward selection
regfit.fwd = regsubsets(Mapk1~.,data=kidney,nvmax=20,method = "forward")
fwd.sum = summary(regfit.fwd)
fwd.sum
which.min(fwd.sum$cp)
which.min(fwd.sum$bic)
which.max(fwd.sum$adjr2)
#5 for cp, 4 for bic and 10 for adjr2
#5: Akt2 Rik Pik3r3 Rac1 Pik3r1
#4: Akt2 Rik Pik3r3 Rac1
#same as best subset selection


#Backward selection
regfit.bwd = regsubsets(Mapk1~.,data=kidney,nvmax=20,method = "backward")
bwd.sum = summary(regfit.bwd)
bwd.sum
which.min(bwd.sum$cp)
which.min(bwd.sum$bic)
which.max(bwd.sum$adjr2)
#7 for both cp and bic, 10 for adjr2
#variables: Cdc42 Akt2 Plcg2 Rac2 Sphk2 Ppp3cb Rac1
#very different from best subset selection and forward selection


#Best subset: BIC -- Akt2 Rik Pik3r3 Rac1
#             Cp  -- Akt2 Rik Pik3r3 Rac1 Pik3r1
#Lasso: Akt2, Plcg2, Rik, Pik3cd, Pik3r3, Rac1, Nfat5
#Tree:  Bagging -- Pik3r3, Rac1, Rik, Cdc42, Nfat5
#       Random Forest -- Pik3r3, Rik, Pik3cd, Rac1 and Cdc42
#Common variables in all the models: Akt2 Rik Pik3r3 Rac1

library(glmnet)
library(caTools)
grid=10^seq(10,-2,length=100)


#Ridge Regression
mse.ridge=rep(NA,50)
bestlam=rep(NA,50)
```

```
for (i in 1:50){
set.seed(i)
spl=sample.split(kidney$Mapk1,SplitRatio = 0.8)
train=subset(kidney,spl==TRUE)
test=subset(kidney,spl==FALSE)
x.train=model.matrix(Mapk1~.,train)[,-1]
y.train=train$Mapk1
x.test=model.matrix(Mapk1~.,test)[,-1]
y.test=test$Mapk1
ridge.mod=glmnet(x.train,y.train,alpha=0,lambda=grid)
# Use Cross Validation to choose the tuning parameter
cv.out=cv.glmnet(x.train,y.train,alpha=0)
bestlam[i]=cv.out$lambda.min
ridge.pred=predict(ridge.mod,s=bestlam[i],newx=x.test)
mse.ridge[i]=mean((ridge.pred-y.test)^2)
}


#Lasso Regression
mse.lasso=rep(NA,50)
bestlam.lasso=rep(NA,50)
for (i in 1:50){
set.seed(i)
spl=sample.split(kidney$Mapk1,SplitRatio = 0.8)
train=subset(kidney,spl==TRUE)
test=subset(kidney,spl==FALSE)
x.train=model.matrix(Mapk1~.,train)[,-1]
y.train=train$Mapk1
x.test=model.matrix(Mapk1~.,test)[,-1]
y.test=test$Mapk1
lasso.mod=glmnet(x.train,y.train,alpha=1,lambda=grid)
# Use Cross Validation to choose the tuning parameter
cv.out.lasso=cv.glmnet(x.train,y.train,alpha=1)
bestlam.lasso[i]=cv.out.lasso$lambda.min
lasso.pred=predict(lasso.mod,s=bestlam.lasso[i],newx=x.test)
mse.lasso[i]=mean((lasso.pred-y.test)^2)
}

bestlambda.lasso=bestlam.lasso[which.min(mse.lasso)]
out=glmnet(model.matrix(Mapk1~.,kidney),kidney$Mapk1, alpha=1)
coef.lasso=predict(out,type="coefficients",s=bestlambda.lasso)[1:25,]
coef.lasso[coef.lasso!=0]

#The seven variables selected by lasso are Akt2, Plcg2, Rik, Pik3d, Pik3r3, Rac1 and Nfa

#Tree
library(MASS)
library(tree)
set.seed(1)
spl.tree=sample.split(kidney$Mapk1,SplitRatio = 0.8)
train.tree=subset(kidney,spl==TRUE)
test.tree=subset(kidney,spl==FALSE)
tree.kidney=tree(Mapk1~.,data=train.tree)
summary(tree.kidney)

plot(tree.kidney)
text(tree.kidney,pretty=0)

yhat=predict(tree.kidney,newdata = test.tree)
mean((yhat-test.tree$Mapk1)^2)

#Bagging
library(randomForest)
set.seed(1)
spl.bag=sample.split(kidney$Mapk1,SplitRatio = 0.8)
train.bag=subset(kidney,spl.bag==TRUE)
test.bag=subset(kidney,spl.bag==FALSE)
bag.kidney=randomForest(Mapk1~.,data =train.bag,mtry = 23,importance = TRUE)
yhat.bag=predict(bag.kidney,newdata = test.bag)
mean((yhat.bag-test.bag$Mapk1)^2)


#Random Forest
set.seed(1)
spl.rf=sample.split(kidney$Mapk1,SplitRatio = 0.8)
train.rf=subset(kidney,spl.rf==TRUE)
test.rf=subset(kidney,spl.rf==FALSE)
rf.kidney=randomForest(Mapk1~.,data =train.rf,mtry = 7,importance = TRUE)
yhat.rf=predict(rf.kidney,newdata = test.rf)
mean((yhat.rf-test.rf$Mapk1)^2)
#The test set MSE is 0.008698898.
varImpPlot(bag.kidney)
varImpPlot(rf.kidney)


#Tree, Bagging and Random Forest Performance
# Simple Tree Performance
mse.simpletree=rep(NA,50)
for (i in 1:50){
set.seed(i)
spl.tree1=sample.split(kidney$Mapk1,SplitRatio = 0.8)
train.tree1=subset(kidney,spl.tree1==TRUE)
test.tree1=subset(kidney,spl.tree1==FALSE)
tree.kidney1=tree(Mapk1~.,data=train.tree1)
```

```
yhat1=predict(tree.kidney1,newdata = test.tree1)
mse.simpletree[i]=mean((yhat1-test.tree1$Mapk1)^2)
}

# Bagging MSE Performance
mse.bagging=rep(NA,50)
for (i in 1:50){
set.seed(i)
spl.bag1=sample.split(kidney$Mapk1,SplitRatio = 0.8)
train.bag1=subset(kidney,spl.bag1==TRUE)
test.bag1=subset(kidney,spl.bag1==FALSE)
bag.kidney1=randomForest(Mapk1~.,data =train.bag,mtry = 23,importance = TRUE)
yhat.bag1=predict(bag.kidney1,newdata = test.bag)
mse.bagging[i]=mean((yhat.bag1-test.bag1$Mapk1)^2)
}

# Random Forest Performance
mse.rf=rep(NA,50)
for (i in 1:50){
set.seed(i)
spl.rf1=sample.split(kidney$Mapk1,SplitRatio = 0.8)
train.rf1=subset(kidney,spl.rf1==TRUE)
test.rf1=subset(kidney,spl.rf1==FALSE)
rf.kidney1=randomForest(Mapk1~.,data =train.rf1,mtry = 7,importance = TRUE)
yhat.rf1=predict(rf.kidney1,newdata = test.rf1)
mse.rf[i]=mean((yhat.rf1-test.rf1$Mapk1)^2)
}


#linear regression fit

#fit the model with 12 variables picked by backward selection
fit.12 = lm(Mapk1~Cdc42+Pla2g6+Akt2+Plcg2+Rac2+Rik+Pla2g5+Sphk2+Map2k1+Ptk2+ Nos3+Rac1,d
summary(fit.12)

#fit the model with the four variables picked by lasso and best subset selection
par(mfrow=c(2,2))
gene.fit = lm(Mapk1~Akt2+Rik+Pik3r3+Rac1,data=kidney)
summary(gene.fit)
plot(gene.fit,1)
plot(gene.fit,2)
plot(gene.fit,3)
plot(gene.fit,4)

#fit the model with five variables picked by best subset selection based on Cp.
five.fit = lm(Mapk1~Akt2+Rik+Pik3r3+Rac1+Pik3r1,data=kidney)
summary(five.fit)
```

```
plot(five.fit,1)
plot(five.fit,2)
plot(five.fit,3)
plot(five.fit,4)

#The performance of the two models
lm.err = rep(NA,50)
lm5.err = rep(NA,50)
for(i in 1:50){
  set.seed(i)
  train = sample(40,32,replace = FALSE)
  kidney.t = kidney[train,]
  kidney.te = kidney[-train,]
  gene.fit = lm(Mapk1~Akt2+Rik+Pik3r3+Rac1,data=kidney.t)
  bic.fit = lm(Mapk1~Akt2+Rik+Pik3r3+Rac1+Pik3r1,data=kidney.t)
  bw.fit = lm(Mapk1~Cdc42+Akt2+Plcg2+Rac2+Sphk2+Ppp3cb+Rac1,data=kidney)
  lm.err[i] = mean((kidney.te$Mapk1-predict(gene.fit,newdata=kidney.te))^2)
  lm5.err[i] =  mean((kidney.te$Mapk1-predict(bic.fit,newdata=kidney.te))^2)
}

#boxplot to compare the performance of each model
boxplot(lm.err,lm5.err,mse.ridge,mse.lasso,mse.rf,names = c("4 predictors","5 predictors
"Ridge","Lasso","Random forest"),main="Comparing test MSE")
```