STAT 388 Final

STAT 351/488

Predictive Analytics - Exam 2

Due December 10, 2019

Name: Charles Hwang

ID Number: 00001447912

Problem 1

The main ideas of STAT 388: Predictive Analytics use sampling theory, non-parametric methods, multivariate analysis, and decision theory to make predictions on a given set of data. Oftentimes, a data set is randomly divided into two subsets, a training set and a test set. The training set is used to "train" and fit the model created through various methods (including—but not limited to—cross validation, bootstrapping, regression, random forests, and support vector machines), and the test set (with the response variable removed) is used to test that model.

Problem 2

We may choose to use a more restrictive method like a linear model if we are afraid of overfitting (for example, if the consequences of overfitting are far greater than those of underfitting). Additionally, linear models in particular can be easier to interpret and explain.

Problem 3

In supervised learning, we have a set of "n" observations with "p" predictors and a response variable "Y" and are supposed to predict the response variable for future observations. In unsupervised learning, we are not given the response variable and thus are unable to predict it. Unsupervised learning is often more challenging because of this, and we can instead choose to discover interesting trends and subgroups in the data or easier and more informative ways to visualize the data.

Problem 4

Problem 4a

Generalized additive models (GAMs), logistic regression, principal component analysis (PCA), k-nearest neighbors (KNN), random forests, linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), CART, support vector machines (SVM), and hierarchical clustering can be used for classification problems.

Problem 4b

Generalized additive models (GAMs), LASSO, ridge regression, random forests, and CART can be used for regression problems.

Problem 4c

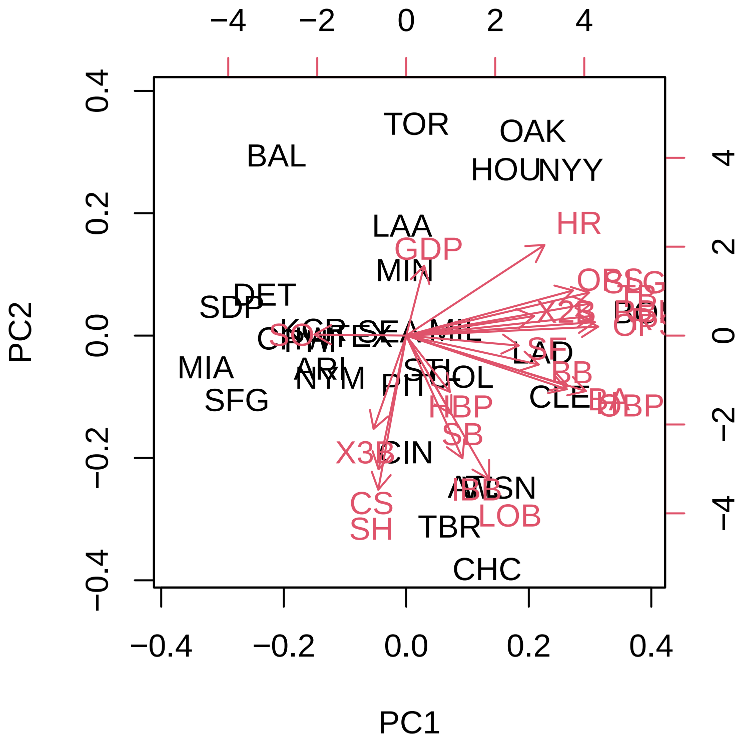
Principal component analysis (PCA) and hierarchical clustering are considered unsupervised learning techniques.

Problem 5

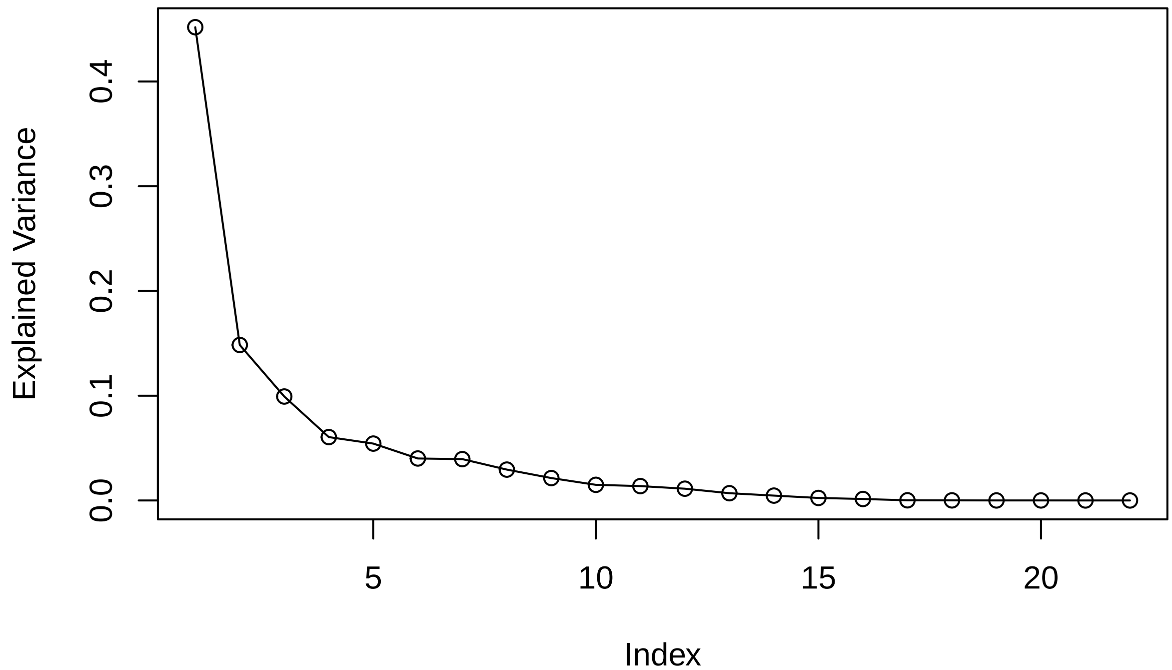
```r
data <- read.csv(file="/Users/newuser/Desktop/Notes/Undergraduate/STAT 338 - Predictive Analytics/MLB20
data <- data[-(31:32),]
row.names(data) <- data$Tm
data <- data[,-(1:7)]
summary(prcomp(data,scale.=TRUE))
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     3.1527 1.8069 1.47755 1.15377 1.09271 0.93897 0.93166
## Proportion of Variance 0.4518 0.1484 0.09923 0.06051 0.05427 0.04008 0.03945
## Cumulative Proportion  0.4518 0.6002 0.69944 0.75995 0.81422 0.85430 0.89375
##                            PC8     PC9    PC10    PC11    PC12    PC13    PC14
## Standard deviation     0.80496 0.68555 0.57310 0.54911 0.49714 0.39002 0.31890
## Proportion of Variance 0.02945 0.02136 0.01493 0.01371 0.01123 0.00691 0.00462
## Cumulative Proportion  0.92321 0.94457 0.95950 0.97321 0.98444 0.99135 0.99598
##                          PC15    PC16    PC17    PC18    PC19    PC20    PC21
## Standard deviation     0.2297 0.17469 0.05617 0.04042 0.01472 0.01291 0.00788
## Proportion of Variance 0.0024 0.00139 0.00014 0.00007 0.00001 0.00001 0.00000
## Cumulative Proportion  0.9984 0.99976 0.99991 0.99998 0.99999 1.00000 1.00000
##                             PC22
## Standard deviation     6.743e-16
## Proportion of Variance 0.000e+00
## Cumulative Proportion  1.000e+00
```

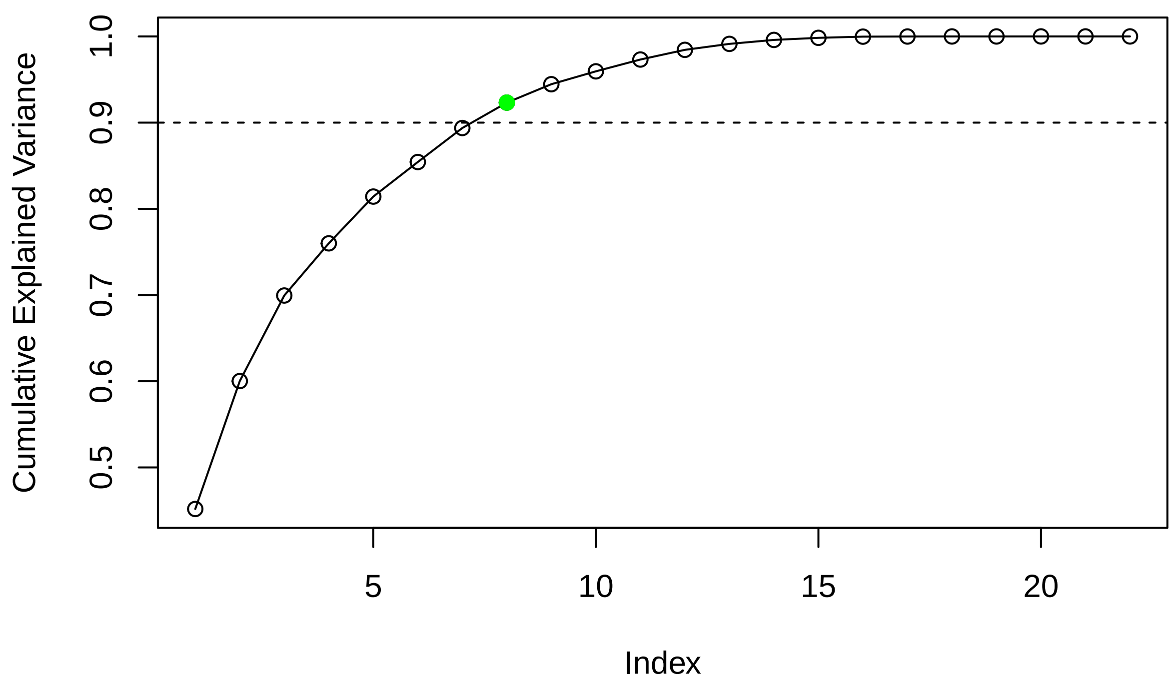
```r
biplot(prcomp(data,scale.=TRUE))
```



```r
plot(prcomp(data,scale.=TRUE)$sdev^2/sum(prcomp(data,scale.=TRUE)$sdev^2),ylab="Explained Variance",typ
```




```
# I would choose to keep 10 principal components. Ten principal components would explain over 95 percen
plot(cumsum(prcomp(data,scale.=TRUE)$sdev^2/sum(prcomp(data,scale.=TRUE)$sdev^2)),ylab="Cumulative Expla
abline(.9,0,lty=2)
points(8,cumsum(prcomp(data,scale.=TRUE)$sdev^2/sum(prcomp(data,scale.=TRUE)$sdev^2))[8],col="green",pch
```




```
cumsum(prcomp(data,scale.=TRUE)$sdev^2/sum(prcomp(data,scale.=TRUE)$sdev^2))[5:9] # Checking variance v
```

```
## [1] 0.8142246 0.8543007 0.8937546 0.9232072 0.9445701
```

```r
# Eight principal components are needed to explain at least 90 percent of the variance.
```

Problem 6

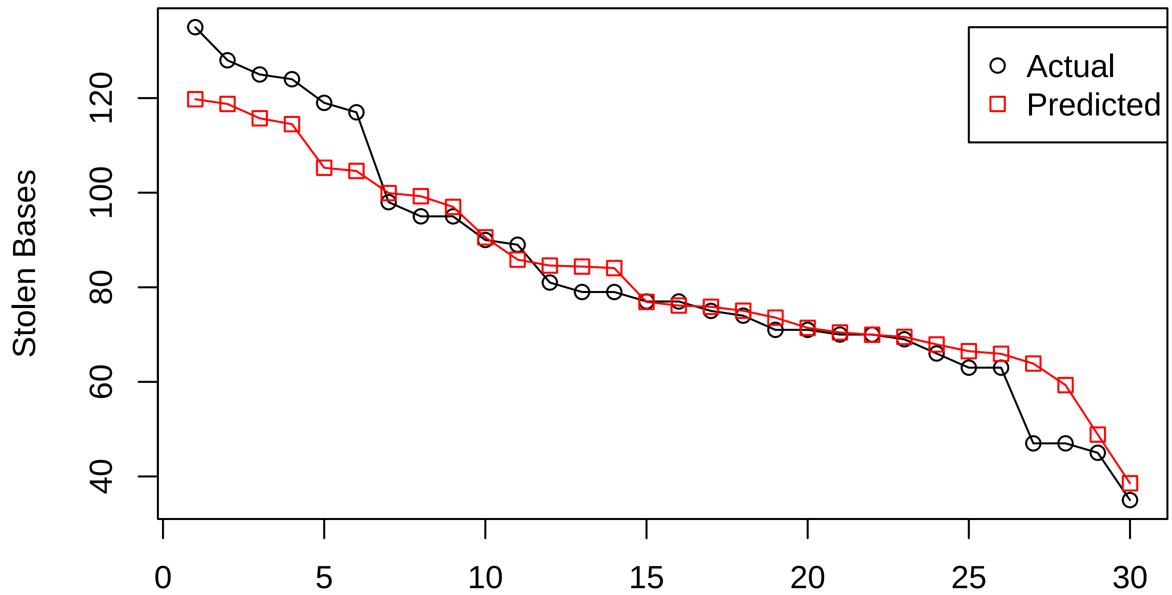
```r
library(glmnet)
set.seed(1012)
datam <- model.matrix(SB~.,data=data)
ridge <- glmnet(datam,data$SB,alpha=0,lambda=10^seq(10,-2,length=100))
best <- min(ridge$lambda)
error <- mean((predict(ridge,s=best,newx=datam)-data$SB)^2)
c(best,error)
```

```
## [1]   0.0100 232.1994
```

```r
predict(ridge,s=best,newx=datam)
```

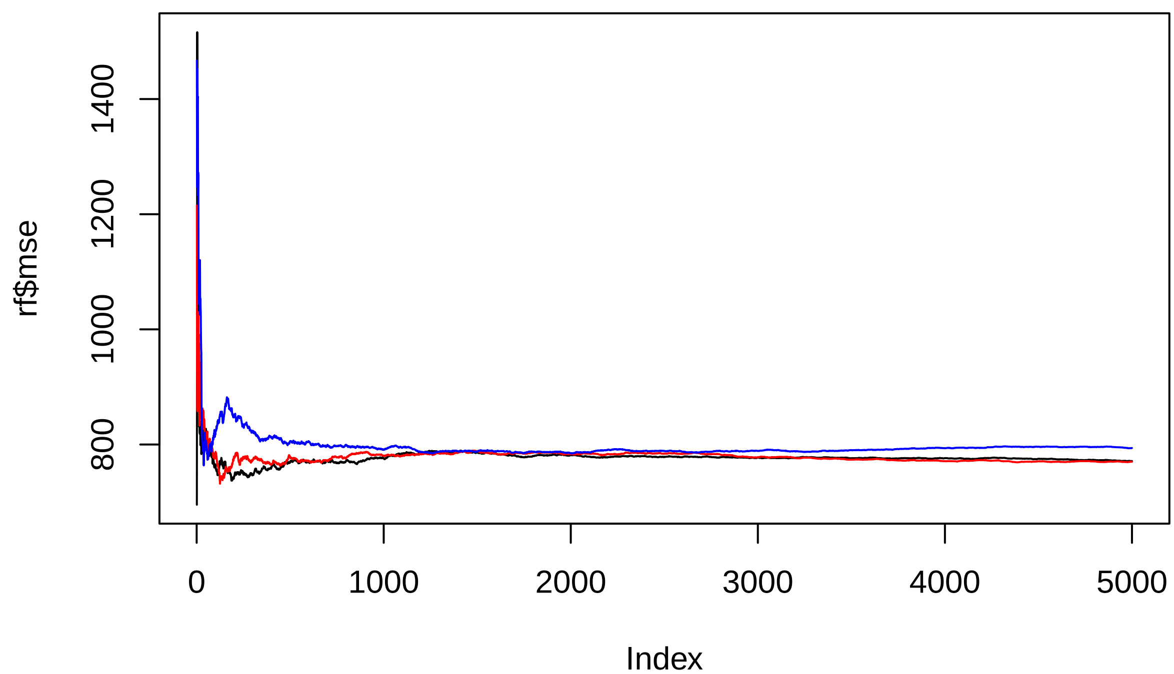
```
##            s1
## ARI  69.50173
## ATL  75.04257
## BAL  66.48307
## BOS 105.26975
## CHC  99.25818
## CHW  85.83858
## CIN  76.90235
## CLE 119.76569
## COL 115.73317
## DET  67.91607
## HOU  69.96026
## KCR  90.56135
## LAA  84.59150
## LAD  76.13351
## MIA  63.86756
## MIL 114.49542
## MIN  65.93084
## NYM  84.37697
## NYY  73.58157
## OAK  38.55491
## PHI  48.85468
## PIT  75.86908
## SDP  96.98848
## SEA 104.59534
## SFG  70.45173
## STL  59.31710
## TBR 118.76694
## TEX  84.05801
## TOR  71.41790
## WSN  99.91569
```

```r
plot(sort(data$SB,decreasing=TRUE),xlab="",ylab="Stolen Bases",type="o")
points(sort(predict(ridge,s=best,newx=datam),decreasing=TRUE),pch=0,col="red",type="o")
legend(25,135,c("Actual","Predicted"),col=c("black","red"),pch=c(1,0))
```



Problem 7

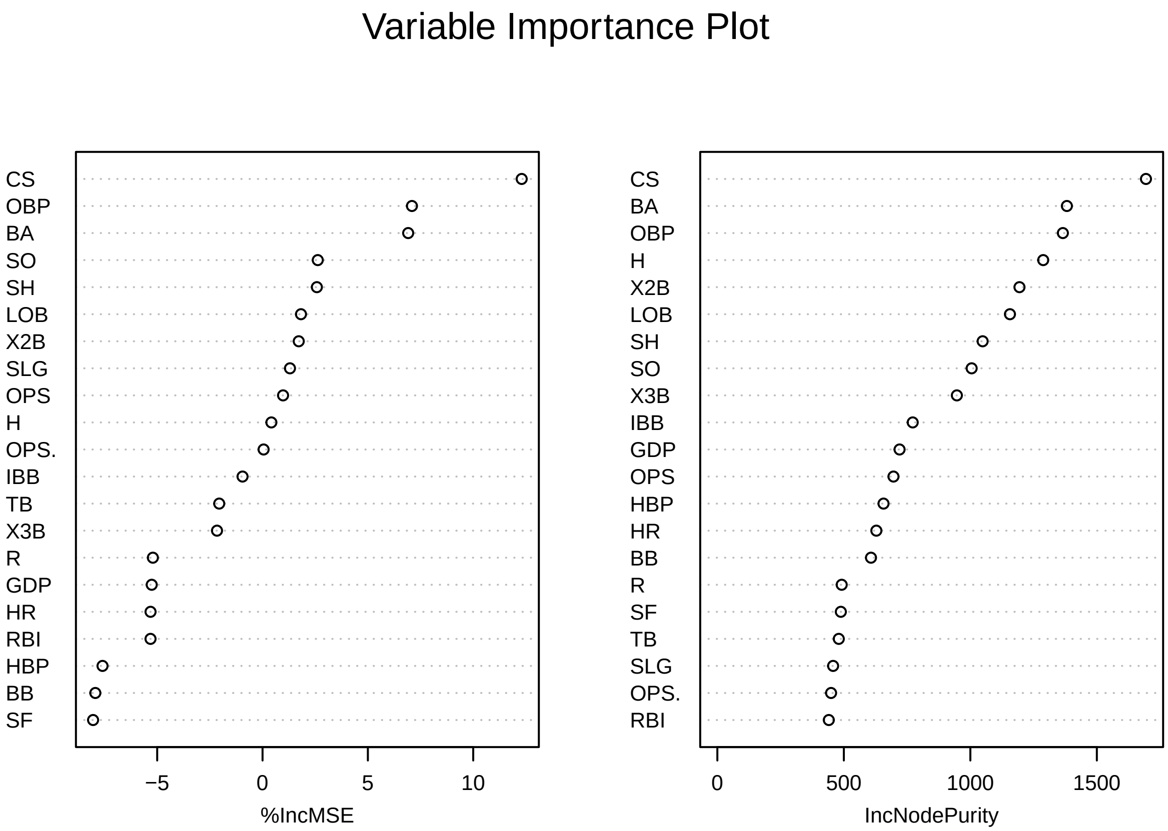
```
library(randomForest)
set.seed(1012)
rf <- randomForest(SB~.,data=data,ntree=5000,importance=TRUE) # Choosing arbitrary number of trees
rf4 <- randomForest(SB~.,data=data,ntree=5000,importance=TRUE,mtry=4) # Choosing different numbers of v
rf11 <- randomForest(SB~.,data=data,ntree=5000,importance=TRUE,mtry=11)
plot(rf$mse,type="l")
points(1:5000,rf4$mse,type="l",col="red")
points(1:5000,rf11$mse,type="l",col="blue")
```




```
summary(rf$mse)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   695.4   774.6   776.7   777.1   779.1  1516.1
```

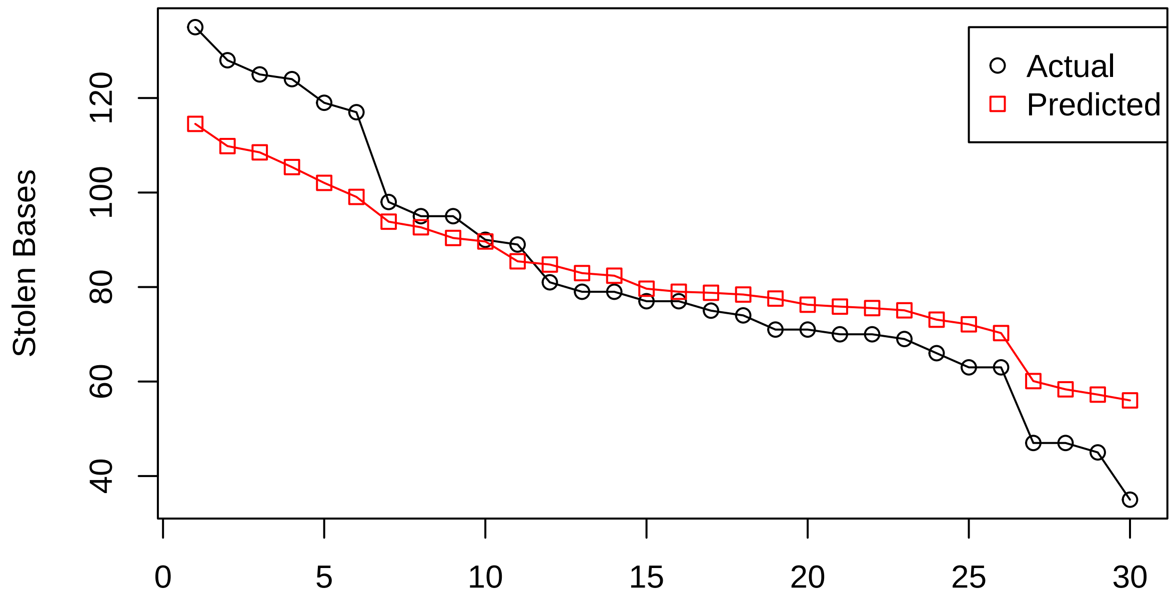
```
varImpPlot(rf,main="Variable Importance Plot",cex=.7)
```




```
predict(rf,data)
```

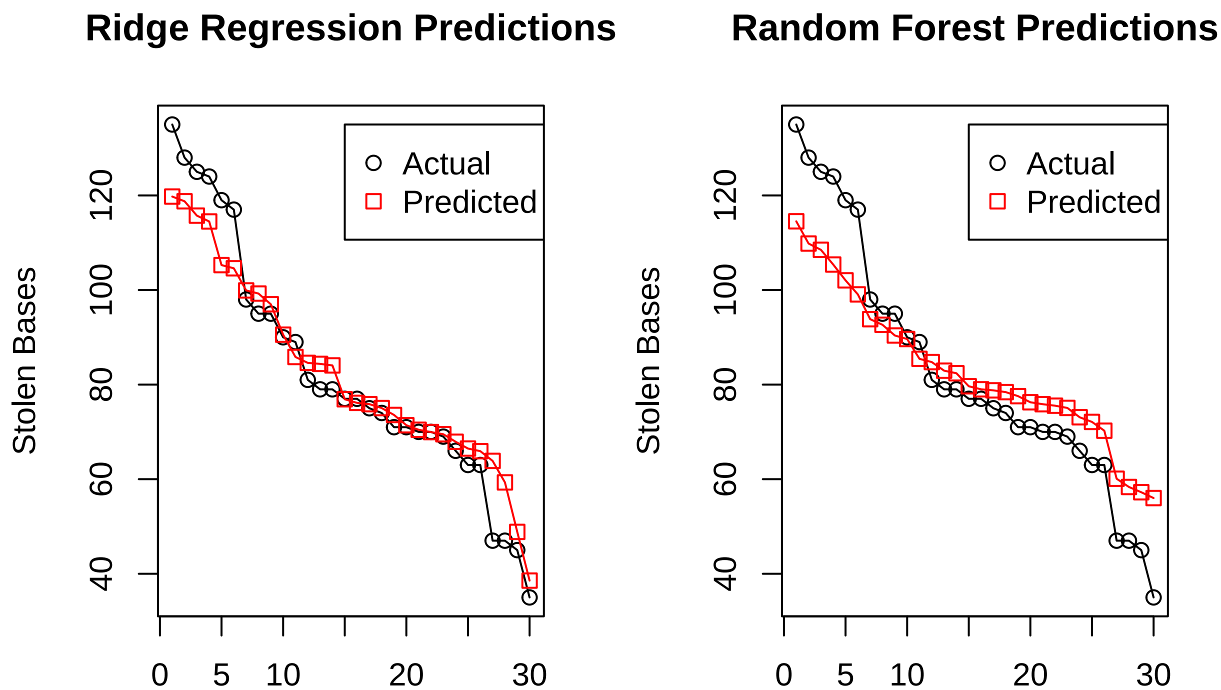
```
##       ARI       ATL       BAL       BOS       CHC       CHW       CIN       CLE
##  77.56218  90.38296  79.64770 109.82626  84.76034  92.64620  82.94535 114.53256
##       COL       DET       HOU       KCR       LAA       LAD       MIA       MIL
##  93.83679  75.07166  75.85750  99.07408  79.00810  82.40049  60.09915 102.04553
##       MIN       NYM       NYY       OAK       PHI       PIT       SDP       SEA
##  57.24135  75.54448  72.10698  58.34966  73.09732  78.78305  89.64224  85.44632
##       SFG       STL       TBR       TEX       TOR       WSN
##  78.41400  70.25515 108.50573  76.26723  56.00902 105.39279
```

```
plot(sort(data$SB,decreasing=TRUE),xlab="",ylab="Stolen Bases",type="o")
points(sort(predict(rf,data),decreasing=TRUE),pch=0,col="red",type="o")
legend(25,135,c("Actual","Predicted"),col=c("black","red"),pch=c(1,0))
```



Problem 8

```
par(mfrow=c(1,2))
plot(sort(data$SB,decreasing=TRUE),xlab="",ylab="Stolen Bases",main="Ridge Regression Predictions",type=
points(sort(predict(ridge,s=best,newx=datam),decreasing=TRUE),pch=0,col="red",type="o")
legend(15,135,c("Actual","Predicted"),col=c("black","red"),pch=c(1,0))
plot(sort(data$SB,decreasing=TRUE),xlab="",ylab="Stolen Bases",main="Random Forest Predictions",type="o
points(sort(predict(rf,data),decreasing=TRUE),pch=0,col="red",type="o")
legend(15,135,c("Actual","Predicted"),col=c("black","red"),pch=c(1,0))
```



```r
# It looks like the ridge regression model is better in this case.
```

Problem 9

```r
rm(list=ls())
n=30                                                                                        # Problem 9a
n
```

```
## [1] 30
```

```r
x2=0                                                                                        # Problem 9b
x2 < 0.0396957
```

```
## [1] TRUE
```

```r
x1=1
x1 > 0.0682064
```

```
## [1] TRUE
```

```r
cat("Example point: x1=",x1,", x2=",x2,sep="")
```

```
## Example point: x1=1, x2=0
```

```r
# Pruning the tree or choosing a smaller number of trees can help avoid overfitting. # Problem 9c
```

Problems 10-11

```r
"Problem 10:" # The number of variables per split "m", usually sqrt(p) for a classification tree and p/
"Problem 11a:" # No, this classifier is not a maximal margin classifier. There is no maximal margin cla
"Problem 11b:" # The boundary could shift slightly to the right or rotate slightly counterclockwise (or
```

Problems 12-13

```r
knitr::knit_hooks$set(error = function(x, options) {
  paste0("<pre style=\"color: red;\"><code>", x, "</code></pre>")
  })
Picture_With_Dr._Matthews_And_Dr._Perry_Being_BFFs             # Problem 12
```

```r
Three_Reasons_Why_Smash_Mouth_Is_The_Greatest_Band_Of_All_Time # Problem 13
```