STAT460 – Homework 8
Due: Mar. 25 at the start of class.

For this assignment, let's attempt to make a spam filter[1]. Usually, this would involve a lot of text processesing on a huge number of emails, creating a large design matrix, $\mathbb{X}$, which has rows given by individual emails and columns given by the number of each word or character that appears in that email, as well as three different numerical measures regarding capital letters (average length of consecutive capitals, longest sequence of consecutive capitals, and total number of capital letters). The response, Y, is given by the user supplied label marking that email as either spam ($Y = 1$) or not ($Y = 0$).

1. Read in the `R` data set `spam.Rdata` and read the documentation file `spambase.Documentation`. What object is loaded into memory? What objects are inside that object? How many emails do we have total? What words/characters/other covariates are in this data set?

2. Let's make a training and test set.

```
train = spam$train
test  = !train
X     = spam$XdataF[train,]
X_0   = spam$XdataF[test,]
Y     = factor(spam$Y[train])
Y_0   = factor(spam$Y[test])
```

   How many observations are in the training set (that is, what is n)? How many observations are in the test set?

3. Run the following code

```
out.tree = tree(Y~.,data=X)
tmp.tree = prune.tree(out.tree,best=3)
plot(tmp.tree)
text(tmp.tree)
```

   What covariate is split on first? Interpet the split point. Make the corresponding partition view for this dendrogram (you don't need to use `R` for this, just draw the right rectangles and be neat about it)

4. Fit an unpruned classification tree to the training data (hint: you've already done that on this h/w). Get the associated test misclassification rate and test confusion matrix using the following function

```
miss.class =function(pred.class,true.class,produceOutput=FALSE){
  confusion.mat = table(pred.class,true.class)
  if(produceOutput){
    return(1-sum(diag(confusion.mat))/sum(confusion.mat))
  }
  else{
```

[1] It has been estimated that spam (that is, unsolicited bulk) emails cost businesses $\approx$ \$10 billion a year

```
    print('miss-class')
    print(1-sum(diag(confusion.mat))/sum(confusion.mat))
    print('confusion mat')
    print(confusion.mat)
  }
}
# this can be called using:
#     (assuming you make the appropriately named test predictions)
miss.class(Y.hat,Y_0)
```

5. Lastly, attempt to prune the tree via weakest-link pruning (i.e. using the `cv.tree` and `prune.misclass` pair of functions as shown in lecture). What are this tree's test misclassification rate and test confusion matrix? How does this compare to the original tree `out.tree`?