#Step0: Using R to calculate a number based on the initial.  
 which(letters=="n")

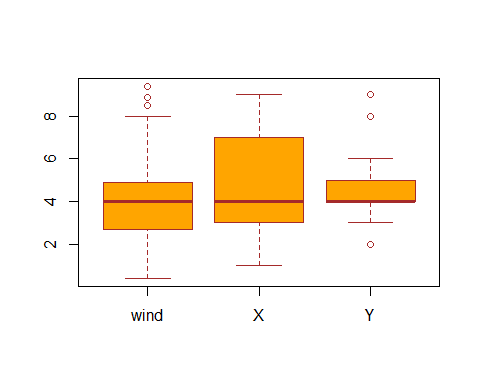
## [1] 14

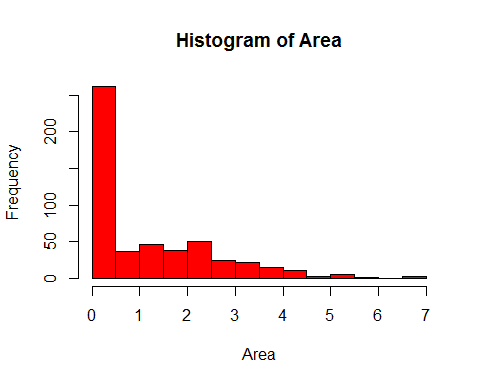
which(letters=="a")

## [1] 1

**So the number is 141 which is equal to (14\*10)+1.**   
#Step1:Dataset name: Forestfires  
#Number of Attributes: 12 + output attribute  
#1. X - x-axis spatial coordinate within the Montesinho park map: 1 to 9  
#2. Y - y-axis spatial coordinate within the Montesinho park map: 2 to 9  
#3. month - month of the year: "jan" to "dec"   
#4. day - day of the week: "mon" to "sun"  
#5. FFMC - FFMC index from the FWI system: 18.7 to 96.20  
#6. DMC - DMC index from the FWI system: 1.1 to 291.3   
#7. DC - DC index from the FWI system: 7.9 to 860.6   
#8. ISI - ISI index from the FWI system: 0.0 to 56.10  
#9. temp - temperature in Celsius degrees: 2.2 to 33.30  
#10. RH - relative humidity in %: 15.0 to 100  
#11. wind - wind speed in km/h: 0.40 to 9.40   
#12. rain - outside rain in mm/m2 : 0.0 to 6.4   
#Response variable is "Area"-the burned area of the forest (in ha): 0.00 to 1090.84   
#Number of Instances: 517

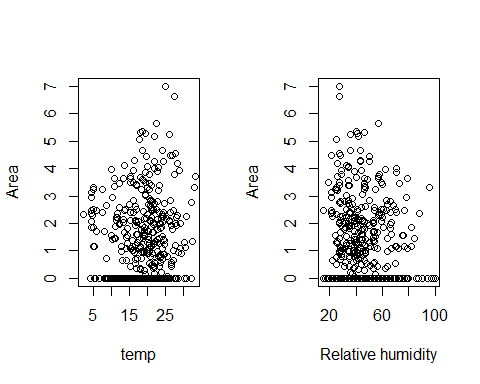
**This is a Regression problem and the model will explain the relationship between the explanatory variables and the response variable and to predict the burned area of forest fires, in the northeast region of Portugal, by using meteorological and other data.Later on to use various methods to enhance model performance, we do recoding of the response variable as the binary categorical variable.**   
**I took log of response var area as 'log(area+1)' transformation due to the heavy skewness and many zeroes in that variable.(fires that burnt less than a hectare).**   
  
#Step 2: Graphical EDA  
#Boxplots,scatterplots and histogram





**The histogram shows the frequency of smaller fires is more than larger fires.**

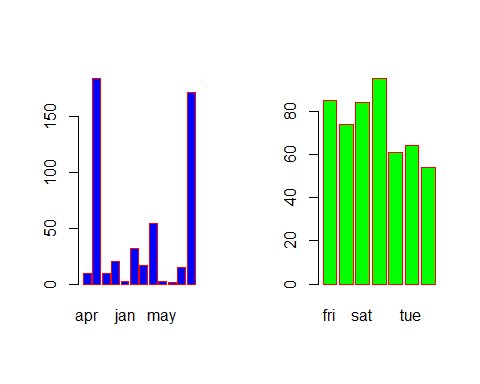
**The scatterplot below shows as the temperature increases, burned forest area is larger. Also, relative humidity of 20%-40% cause larger fires than higher values of humidity.**



**Categorical Variables: Number of fires in each month and according to days.**

## month  
## apr aug dec feb jan jul jun mar may nov oct sep   
## 9 184 9 20 2 32 17 54 2 1 15 172

## day  
## fri mon sat sun thu tue wed   
## 85 74 84 95 61 64 54



**The month of August and September show the most fires and that too more on weekends.**

The Correlation matrix of the predictors in dataset:  
## X Y FFMC DMC DC ISI temp RH wind rain  
## X 1.00 0.54 -0.02 -0.05 -0.09 0.01 -0.05 0.09 0.02 0.07  
## Y 0.54 1.00 -0.05 0.01 -0.10 -0.02 -0.02 0.06 -0.02 0.03  
## FFMC -0.02 -0.05 1.00 0.38 0.33 0.53 0.43 -0.30 -0.03 0.06  
## DMC -0.05 0.01 0.38 1.00 0.68 0.31 0.47 0.07 -0.11 0.07  
## DC -0.09 -0.10 0.33 0.68 1.00 0.23 0.50 -0.04 -0.20 0.04  
## ISI 0.01 -0.02 0.53 0.31 0.23 1.00 0.39 -0.13 0.11 0.07  
## temp -0.05 -0.02 0.43 0.47 0.50 0.39 1.00 -0.53 -0.23 0.07  
## RH 0.09 0.06 -0.30 0.07 -0.04 -0.13 -0.53 1.00 0.07 0.10  
## wind 0.02 -0.02 -0.03 -0.11 -0.20 0.11 -0.23 0.07 1.00 0.06  
## rain 0.07 0.03 0.06 0.07 0.04 0.07 0.07 0.10 0.06 1.00

**Some amount of positive correlation is seen between X and Y and between DC and DMC variables and negative correlation between RH and temp(but it is not much significant).**

#Step 3 : The statistical method I used here is 'Multiple linear regression' using least square approach.I did transfomration of response variable 'area' as 'log(area+1) since the model fit is better after the transformation.  
# So since least square approach is used the tuning parameter value is "0" in this case.  
  
Fit1<-lm(Area~.,data=mydata)  
#the summary of the linear model formed has DMC and monthdec as two significant variables, but the value of adjusted r-sq was very low, so variable selection should be done, besides considering other statistical learning methods like Ridge Regression and Lasso.  
  
# Method of subset selection for variable selection

M1=regsubsets(Area~.,data=mydata,nvmax =12)  
M1summary<-summary(M1)  
which.max (M1summary$adjr2)

## [1] 10

which.min (M1summary$cp )#5 variable model

## [1] 5

which.min (M1summary$bic )#1 variable model

## [1] 1

**Thus,adjusted rsq gave 10 variable model, Cp pick 5 variable and BIC pick single variable(temp) model. So, to be sure I performed forward and backward stepwise selection methods as well.**

step(lm(Area~.,data=mydata),direction="both")

**Forward and backward stepwise selection pick 6 variable model("X","month","DMC", "DC","Wind","temp")**

## Call:  
## lm(formula = Area ~ X + month + DMC + DC + temp + wind, data = mydata)  
##   
## Coefficients:  
## (Intercept) X monthaug monthdec monthfeb   
## 0.185102 0.047093 0.320707 2.203821 0.198425   
## monthjan monthjul monthjun monthmar monthmay   
## -0.443735 0.127736 -0.361830 -0.341704 0.751387   
## monthnov monthoct monthsep DMC DC   
## -0.913303 0.864504 0.974630 0.004151 -0.001988   
## temp wind   
## 0.034672 0.052480

**Now we need to apply some statistical methods which first train on the training dataset and then with the help of best tuning parameter predict the response Area for the validation data as well as its prediction accuracy.This include subset selection method and regularization methods like ridge regression and Lasso.**

#Setting seed, forming training and validation sets   
set.seed (141)train1=sample (c(TRUE ,FALSE), nrow(mydata),rep=TRUE)  
 test1 =(! train1 )  
  
#apply regsubsets() to the training set:  
regfit.best=regsubsets(Area~.,data=mydata[train1 ,], nvmax =12)

#computing the validation set error for the best model of each model size.  
test.mat=model.matrix (Area~.,data=mydata [test1 ,])  
#run a loop  
val.errors =rep(NA ,12)  
 for(i in 1:12){  
 coefi=coef(regfit.best,id=i)  
 pred=test.mat[,names(coefi)]%\*% coefi  
 val.errors[i]=mean(( mydata$Area[test1]-pred)^2)  
}  
val.errors#best model is the one that contains two variable.

## [1] 1.930706 1.878855 1.959689 1.958912 1.955176 1.999365 2.073011  
## [8] 2.073555 1.969580 1.990716 2.042386 2.126126

which.min (val.errors )

## [1] 2

**So model with 2 variables have the lowest test MSE.**

coef(regfit.best ,2)

## (Intercept) monthdec DC   
## 0.6536766211 1.6904481817 0.0006611361

**Thus, in case of training data best model will have monthdec and DC as two variables.**

#Finally, we perform best subset selection on the full data set, and select the best two-variable model.

## (Intercept) monthdec temp   
## 0.57120277 1.87906042 0.02684671

**Thus we got monthdec and temp as the two significant variable model by this method.**   
#Ridge regression on training data  
x=model.matrix (Area~.,mydata )[,-1]#produces matrix of 12 predictors  
y<-mydata$Area  
set.seed (141)  
train=sample (1: nrow(x), nrow(x)/2)  
test=(- train )  
y.test=y[test]  
library(glmnet)

grid <- 10 ^ seq(4, -2, length = 100)  
ridge.mod =glmnet (x[train ,],y[train],alpha =0, lambda =grid ,thresh=1e-12)

#cv.glmnet() func performs ten fold cross validation  
cv.out =cv.glmnet (x[train ,],y[train],alpha=0,lambda=grid)  
  
 bestlam =cv.out$lambda.min  
 bestlam

**The value of lambda(tuning parameter) that results in the smallest crossvalidation error is 10000**

## [1] 10000

## Lasso  
lasso.mod=glmnet (x[train,],y[train],alpha=1, lambda=grid)  
set.seed (141)  
cv.out =cv.glmnet (x[train ,],y[train],alpha =1)  
bestlam =cv.out$lambda.min  
bestlam

**The value of lambda(tuning parameter) that results in the smallest crossvalidation error is 0.1741**

## [1] 0.1741563

#Step4 Evaluating model performance on the validation set, first ridge regrsn  
ridge.pred=predict(ridge.mod ,s=bestlam ,newx=x[test ,])  
mean(( ridge.pred -y.test)^2)# the test MSE is 2.28

## [1] 2.287357

# refitting the ridge regression model on the full data set using the value of lambda chosen by cross-validation, and examine the coefficient estimates.  
out=glmnet (x,y,alpha =0)  
predict (out,type="coefficients",s=bestlam )[1:28,] #none of the coefficients are zero as ridge regression does not perform variable selection.

## (Intercept) X Y monthaug monthdec   
## -0.3430415654 0.0399067129 0.0053455466 -0.1785690264 1.4753565032   
## monthfeb monthjan monthjul monthjun monthmar   
## 0.1930431007 -0.3300707991 -0.1542867973 -0.3279660578 -0.2590039819   
## monthmay monthnov monthoct monthsep daymon   
## 0.7225784376 -0.9851887831 -0.0663261556 0.1856616773 0.0633422420   
## daysat daysun daythu daytue daywed   
## 0.2203729394 0.1384918204 -0.0120483477 0.2198452656 0.0993029289   
## FFMC DMC DC ISI temp   
## 0.0074958789 0.0019787675 -0.0002857309 -0.0105935431 0.0200421646   
## RH wind rain   
## -0.0017971656 0.0522333897 0.0479249166

#Then Lasso, evaluation on test data  
lasso.pred=predict (lasso.mod ,s=bestlam ,newx=x[test,])  
mean(( lasso.pred -y.test)^2)

## [1] 1.73854

**# the test MSE is 1.73 which is lower than that of ridge regression but slightly higher than that of least squares. So the best two variable model seems to be the appropriate one.**

## Step 5

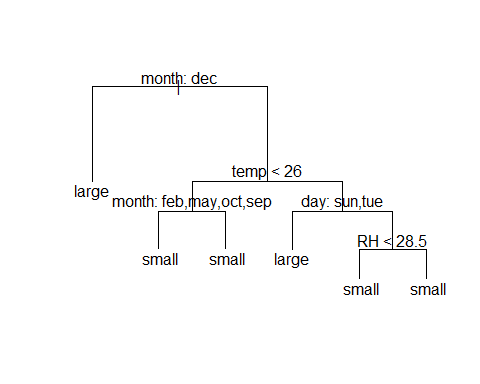
# We first use classification trees to analyze the Forestfires data set. In these data, area is a continuous variable, and so we begin by recoding it as a binary variable. We use the ifelse() function to create a variable.  
   
 Fire=ifelse (area <=5," small"," large ")   
 #Finally, we use the data.frame() function to merge Fire with the rest of the data.  
d1 =data.frame(d1 ,Fire)

tree.d1 =tree(Fire~.-area ,d1 )  
summary (tree.d1 )  
## Classification tree:  
## tree(formula = Fire ~ . - area, data = d1)  
## Variables actually used in tree construction:  
## [1] "month" "temp" "day" "RH"   
## Number of terminal nodes: 6   
## Residual mean deviance: 1.116 = 570.3 / 511   
## Misclassification error rate: 0.2592 = 134 / 517

**Decision tree uses 4 variables and the misclassification error rate is 0.25 which is good, that is lower rate of incorrect classification**.

plot(tree.d1 )# to display the tree structure

**The most important indicator of burned forest area appears to be monthdec since the first branch differentiates into small and large fires in dec**.



#Prediction accuracy for test set  
pred<-predict(dt,test,type="class")  
tab<-table(predictions=pred,actual=test$Fire)  
tab

## actual  
## predictions large small  
## large 10 16  
## small 35 94

sum(diag(tab))/sum(tab**)**

**Tthe prediction accuracy of decision tree is 67% for this model.**

## [1] 0.6709677

## Random forest  
##Fitting the training model  
library(randomForest)

rf<-randomForest(Fire~.-area,data=train,mtry=3,ntree=20)  
rf  
## randomForest(formula = Fire ~ . - area, data = train, mtry = 3, ntree = 20)   
## Type of random forest: classification  
## Number of trees: 20  
## No. of variables tried at each split: 3  
##   
## OOB estimate of error rate: 33.15%  
## Confusion matrix:  
## large small class.error  
## large 30 76 0.7169811  
## small 44 212 0.1718750

**Tthe Misclassification rate is 33% .**

# prediction for test set  
P<-predict(rf,test,type="class")  
t<-table(predictions=P,actual=test$Fire)  
t

## actual  
## predictions large small  
## large 10 19  
## small 35 91

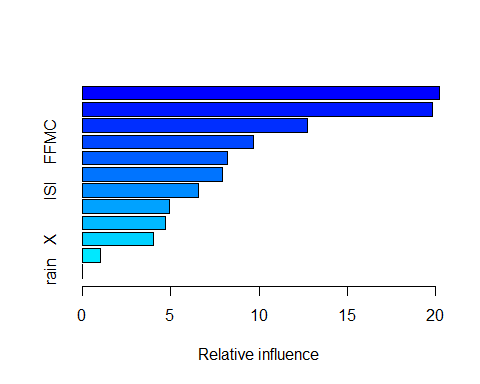
sum(diag(tab))/sum(tab)

## [1] 0.6709677

**It has same prediction accuracy as of decision tree.**

## Boosting  
library(gbm)

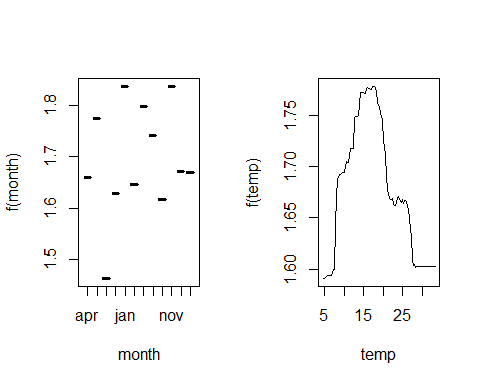
set.seed(141)  
  
bt<-gbm(Fire ~.-area,train, distribution="gaussian",n.trees =5000 , interaction.depth =2)  
summary(bt)



## var rel.inf  
## month month 20.224618  
## temp temp 19.808834  
## day day 12.726886  
## FFMC FFMC 9.716389  
## DMC DMC 8.212477  
## RH RH 7.944327  
## ISI ISI 6.603223  
## wind wind 4.946124  
## DC DC 4.723207  
## X X 4.035565  
## Y Y 1.058348  
## rain rain 0.000000

**Here also we see that month and temp are the most important variables.**

par(mfrow =c(1,2))  
plot(bt,i="month")  
plot(bt,i="temp")#These plots illustrate the marginal effect of the selected variables on the response after integrating out the other variables.



#We now use the boosted model to predict Fires on the test set  
PR<-predict(bt,test,n.trees=5000)  
ta<-table(predictions=PR,actual=test$Fire)  
  
sum(diag(ta))/sum(ta)

## [1] 0.6451613

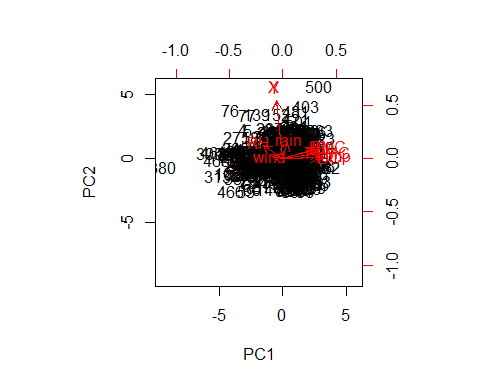
**The prediction accuracy for the boosted model is around 64%.**

# Principal component analysis: This is a part of exploratory data analysis but I tried this just to visualize data as a whole.  
apply(data , 2, mean)# we can see the mean values of vriables like temp wind, relative humidity which gives us the indication of forestfires.

## X Y FFMC DMC DC   
## 4.66924565 4.29980658 90.64468085 110.87234043 547.94003868   
## ISI temp RH wind rain   
## 9.02166344 18.88916828 44.28820116 4.01760155 0.02166344

pr.out=prcomp(data,scale =TRUE)  
  
biplot (pr.out,scale =0)

#The orange arrows indicate the first two principal component loading vectorsThe variables which are close to each other are correlated while some like X,Y, wind they are far away but important in prediction.

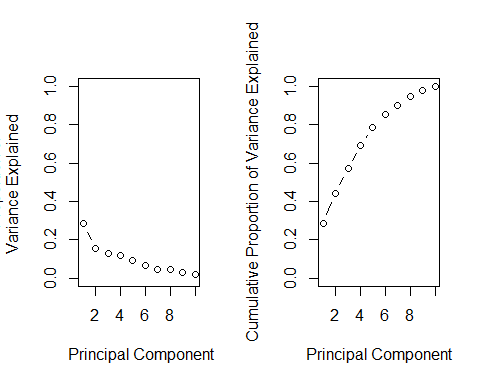


pr.var =pr.out$sdev ^2  
pve=pr.var/sum(pr.var )  
pve

## [1] 0.28596963 0.15593806 0.12961244 0.12124636 0.09315200 0.06926170  
## [7] 0.04755913 0.04630938 0.02946948 0.02148181

#We see that the first principal component explains 28.0% of the variance in the data, the next principal component explains 15% of the variance, and so forth. We can plot the PVE explained by each component, as well as the cumulative PVE, as follows

par(mfrow=c(1,2))  
plot(pve,xlab=" Principal Component", ylab="Proportion of  
Variance Explained ",ylim=c(0,1),type='b')  
plot(cumsum (pve ),xlab="Principal Component",ylab ="  
Cumulative Proportion of Variance Explained",ylim=c(0,1),type='b')



**Conclusion: The best model according to me will be with four predictors model (“temp”, ”month”, ”day” and “RH”) if categorical response variable is used. There might be a little bit change in variable selection when the transformed response variable is used where (“X”, ”month”, ”DC”, ”DMC”, ”wind”, ”temp”) seem to be important. Again, it depends if we treat the categorical variables as numeric or as factors in the latter case “monthdec” and “temp” are shown significant by most of the methods.**