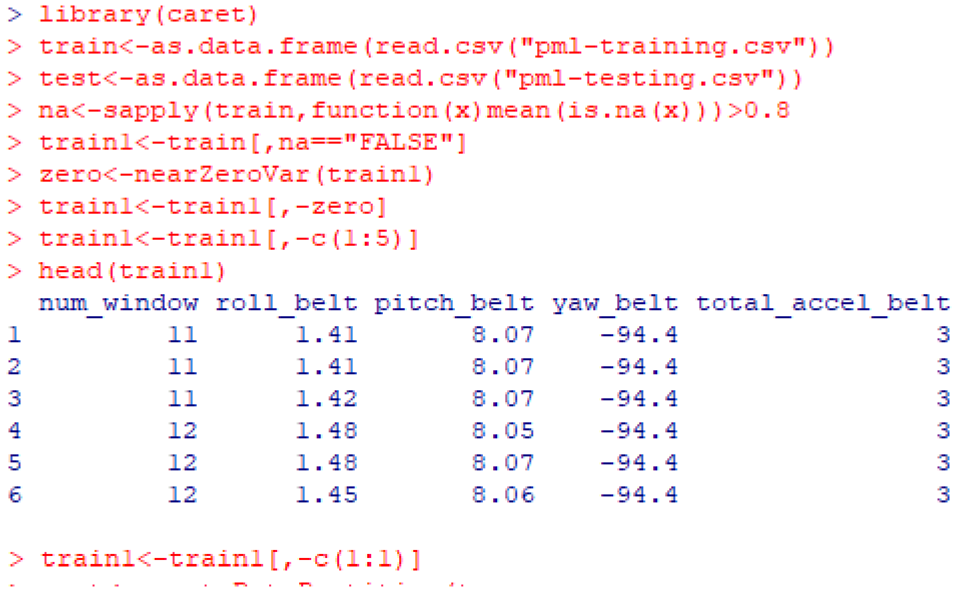
PEER GRADED ASSIGNMENT

NEED:

Using devices such as *Jawbone Up*, *Nike FuelBand*, and *Fitbit* it is now possible to collect a large amount of data about personal activity relatively inexpensively. These type of devices are part of the quantified self movement – a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. One thing that people regularly do is quantify how *much* of a particular activity they do, but they rarely quantify *how well they do it*. In this project, your goal will be to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants. They were asked to perform barbell lifts correctly and incorrectly in 5 different ways

The goal of your project is to predict the manner in which they did the exercise. This is the "classe" variable in the training set. You may use any of the other variables to predict with. You should create a report describing how you built your model, how you used cross validation, what you think the expected out of sample error is, and why you made the choices you did. You will also use your prediction model to predict 20 different test cases.

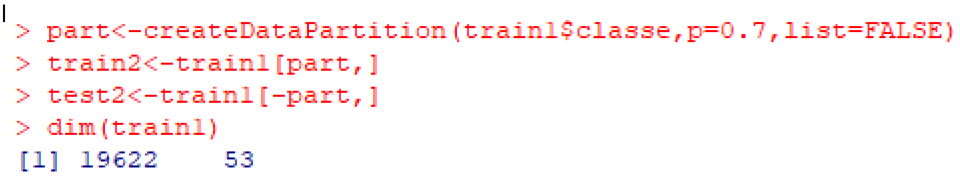
DATA PROCESSING



```
> library(caret)
> train<-as.data.frame(read.csv("pml-training.csv"))
> test<-as.data.frame(read.csv("pml-testing.csv"))
> na<-sapply(train,function(x)mean(is.na(x)))>0.8
> train1<-train[,na=="FALSE"]
> zero<-nearZeroVar(train1)
> train1<-train1[,-zero]
> train1<-train1[,-c(1:5)]
> head(train1)
  num_window roll_belt pitch_belt yaw_belt total_accel_belt
1         11      1.41       8.07    -94.4                3
2         11      1.41       8.07    -94.4                3
3         11      1.42       8.07    -94.4                3
4         12      1.48       8.05    -94.4                3
5         12      1.48       8.07    -94.4                3
6         12      1.45       8.06    -94.4                3

> train1<-train1[,-c(1:1)]
```

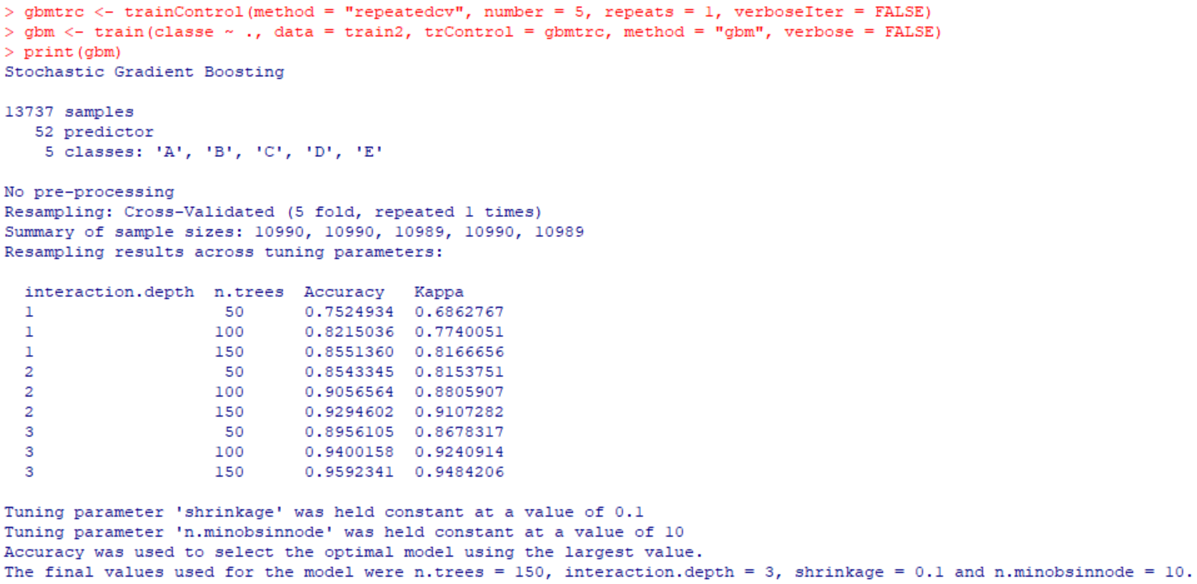
data and packages are loaded now & NA values and near zero values taken out from data. Timestatmp & other dummy label columns taken out. There was num_column and taken out by above code

DIVIDING TRAIN DATA INTO FURTHUR TWO (TRAIN & TEST)



```
> part<-createDataPartition(train1$classe,p=0.7,list=FALSE)
> train2<-train1[part,]
> test2<-train1[-part,]
> dim(train1)
[1] 19622    53
```

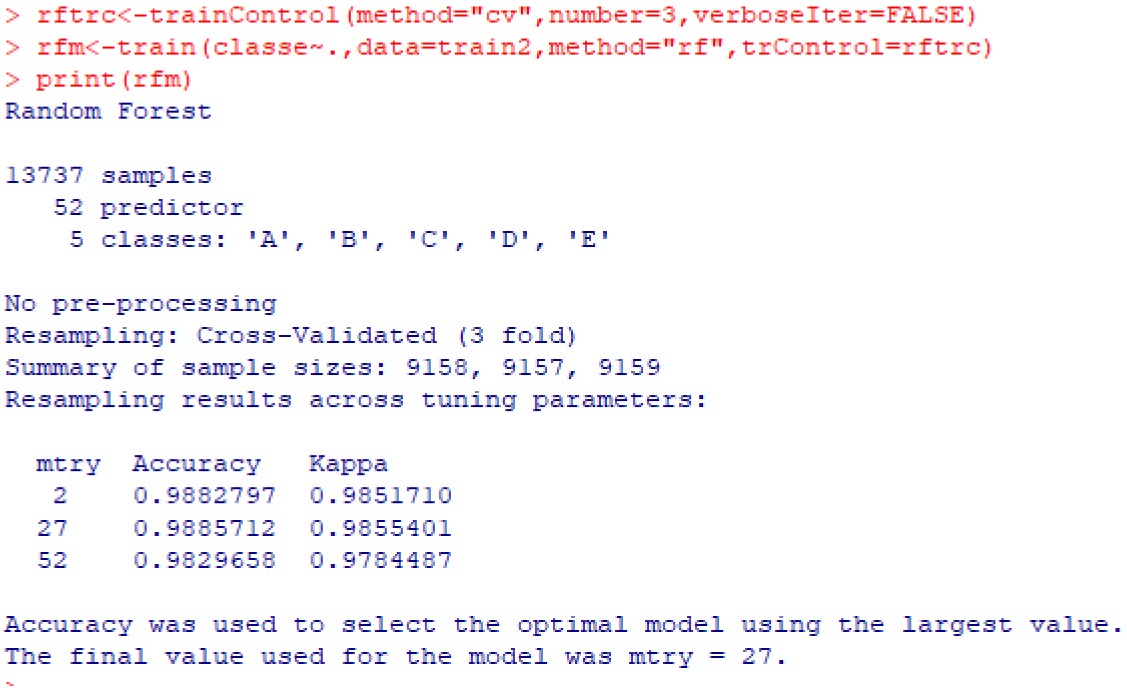
CREATING GBM MODEL



```
> gbmtrc <- trainControl(method = "repeatedcv", number = 5, repeats = 1, verboseIter = FALSE)
> gbm <- train(classe ~ ., data = train2, trControl = gbmtrc, method = "gbm", verbose = FALSE)
> print(gbm)
Stochastic Gradient Boosting

13737 samples
   52 predictor
    5 classes: 'A', 'B', 'C', 'D', 'E'

No pre-processing
Resampling: Cross-Validated (5 fold, repeated 1 times)
Summary of sample sizes: 10990, 10990, 10989, 10990, 10989
Resampling results across tuning parameters:

  interaction.depth  n.trees  Accuracy   Kappa
  1                   50      0.7524934  0.6862767
  1                  100      0.8215036  0.7740051
  1                  150      0.8551360  0.8166656
  2                   50      0.8543345  0.8153751
  2                  100      0.9056564  0.8805907
  2                  150      0.9294602  0.9107282
  3                   50      0.8956105  0.8678317
  3                  100      0.9400158  0.9240914
  3                  150      0.9592341  0.9484206

Tuning parameter 'shrinkage' was held constant at a value of 0.1
Tuning parameter 'n.minobsinnode' was held constant at a value of 10
Accuracy was used to select the optimal model using the largest value.
The final values used for the model were n.trees = 150, interaction.depth = 3, shrinkage = 0.1 and n.minobsinnode = 10.
```

generalized boosted model created to see the accuracy of the prediction.

CREATING RANDOM FOREST MODEL



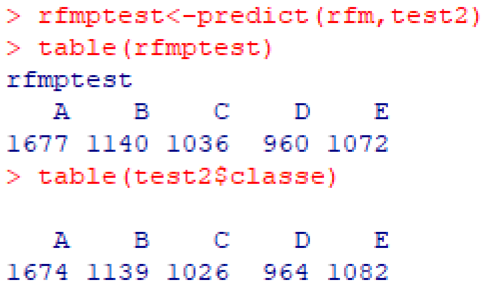
```
> rftrc<-trainControl(method="cv",number=3,verboseIter=FALSE)
> rfm<-train(classe~.,data=train2,method="rf",trControl=rftrc)
> print(rfm)
Random Forest

13737 samples
   52 predictor
    5 classes: 'A', 'B', 'C', 'D', 'E'

No pre-processing
Resampling: Cross-Validated (3 fold)
Summary of sample sizes: 9158, 9157, 9159
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
   2    0.9882797  0.9851710
  27    0.9885712  0.9855401
  52    0.9829658  0.9784487

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 27.
```

random forest model created to see the accuracy of the prediction

random forest gave considerably high accuracy than generalized boosted model.

hence random forest chosen for validation

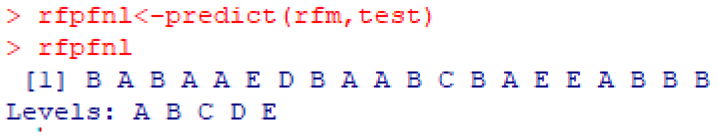
WITH RANDOM FOREST PREDICTING WITH TEST DATA



```
> rfmptest<-predict(rfm,test2)
> table(rfmptest)
rfmptest
   A    B    C    D    E 
1677 1140 1036  960 1072 
> table(test2$classe)

   A    B    C    D    E 
1674 1139 1026  964 1082 
```

Considering only few errors same model used for 20 test case

FINAL VALIDATION WITH TEST DATA



```
> rfpfnl<-predict(rfm,test)
> rfpfnl
 [1] B A B A A E D B A A B C B A E E A B B B
Levels: A B C D E
```