PML project

```
library(gdata)
```

```
## gdata: read.xls support for 'XLS' (Excel 97-2004) files ENABLED.
##
## gdata: read.xls support for 'XLSX' (Excel 2007+) files ENABLED.
##
## Attaching package: 'gdata'
##
## The following object is masked from 'package:stats':
##
##     nobs
##
## The following object is masked from 'package:utils':
##
##     object.size
```

```
library(randomForest)
```

```
## randomForest 4.6-10
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
##
## The following object is masked from 'package:gdata':
##
##     combine
```

read data

```
fitData<-read.csv('pml-training.csv')

testing<-read.csv('pml-testing.csv')
```

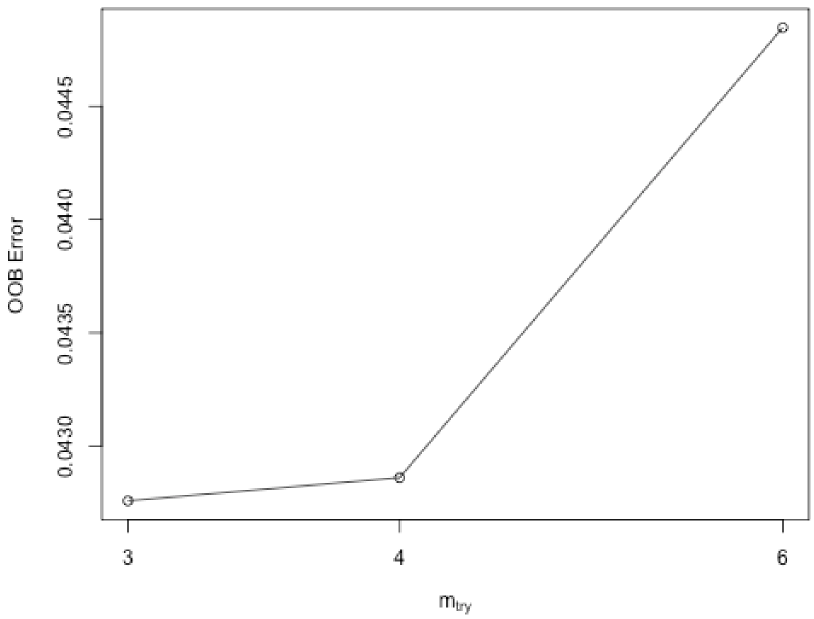
selecting columns with 'accel' but without 'var'

```
myvar<-matchcols(fitData,with='accel',without='var')

mydata<-fitData[c(myvar,'classe')]

testdata<-testing[myvar]
```

Tuning the random forest to find the best parameter mtry with the smallest OOB error

```
fitRF<-tuneRF(mydata[myvar],mydata$classe,stepFactor=1.5,ntreeTry=500)
```

```
## mtry = 4  OOB error = 4.29%
## Searching left ...
## mtry = 3     OOB error = 4.28%
## 0.002378121 0.05
## Searching right ...
## mtry = 6     OOB error = 4.48%
## -0.04637337 0.05
```



Fit the best random forest model and decide the variable importance

```
bestfit<-
randomForest(classe~.,data=mydata,mtry=3,ntree=500,keep.forest=TRUE,importance=TRUE)
```

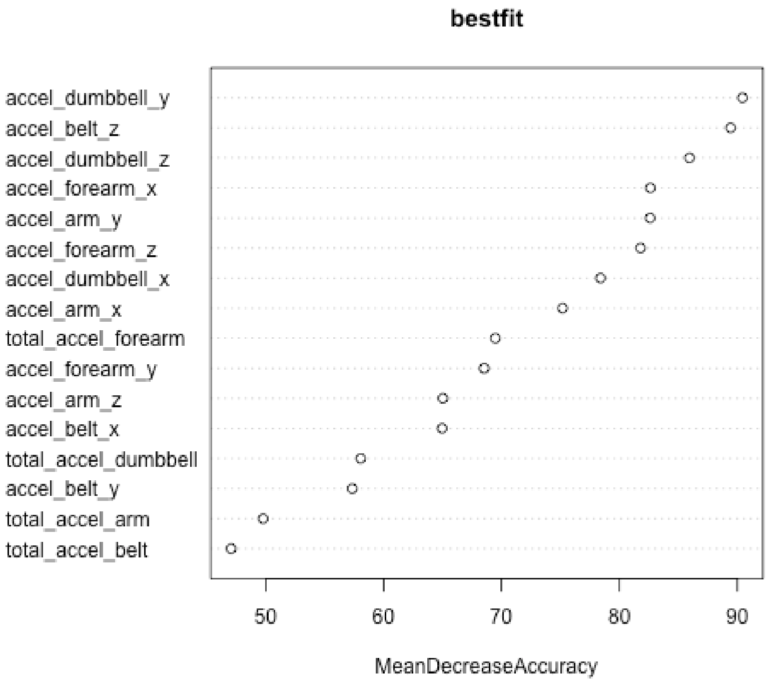
```
print(bestfit)
```

```
##
## Call:
##  randomForest(formula = classe ~ ., data = mydata, mtry = 3, ntree = 500, 
keep.forest = TRUE, importance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 4.22%
## Confusion matrix:
##      A    B    C    D    E class.error
## A 5429   31   57   59    4  0.02706093
## B  111 3551   93   22   20  0.06478799
## C   49   70 3274   24    5  0.04324956
## D   59   16  107 3020   14  0.06094527
## E    5   40   19   24 3519  0.02439701
```

```
importance(bestfit,type=1)
```

```
##                      MeanDecreaseAccuracy
## total_accel_belt                 47.07419
## accel_belt_x                     64.96055
## accel_belt_y                     57.34487
## accel_belt_z                     89.42953
## total_accel_arm                  49.77731
## accel_arm_x                      75.18321
## accel_arm_y                      82.60441
## accel_arm_z                      65.03131
## total_accel_dumbbell             58.05267
## accel_dumbbell_x                 78.40396
## accel_dumbbell_y                 90.43351
## accel_dumbbell_z                 85.94595
## total_accel_forearm              69.48146
## accel_forearm_x                  82.62991
## accel_forearm_y                  68.53126
## accel_forearm_z                  81.79422
```

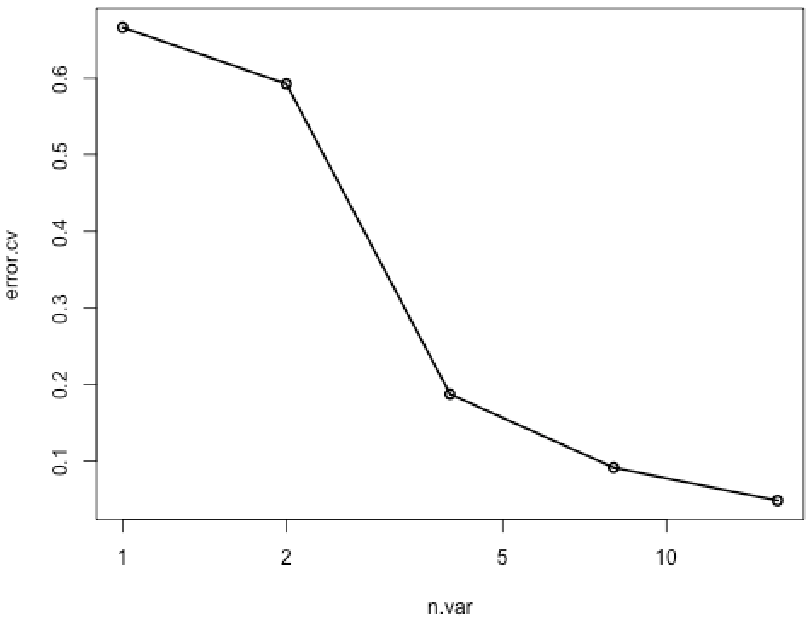
```
varImpPlot(bestfit,type=1)
```



Using random forest cross validation to see if we could possibly reduce the number of predictors

```
featurefit<-rfcv(mydata[myvar],mydata$classe,ntree=500,cv.fold=5)
```

```
with(featurefit,plot(n.var,error.cv,log='x',type='o',lwd=2))
```



predict the testing set

```
pred<-predict(bestfit, testdata)
```