```
knitr::opts_chunk$set(echo = TRUE)
```

To address the project, I chose 'Random forest' to predict the test set. Beforehand, I also measured the accuracy of 'Decision tree' and 'generalized boosted model.' Since 'Random forest' demonstrated the best accuracy (0.9997), I selected this model.

A summary of building up the model.

1) The crossvalidation was performed using 25% of the training sample to evaluate the overfitting.

2) Apply random forest to the train, and crossvallidation samples and evaluation

3) Perform test sample prediction

4) Find the important variables using 'importance' function.

Download packages

```
install.packages("caret") ; require(caret)
install.packages("randomForest")
require(randomForest)
install.packages("e1071")
library(e1071)
```

DATA Loading, remove NA columns for the training and testing data

```
traindata <- read.csv("pml-training.csv", na.strings = c("NA", "#DIV/0!", ""))
testdata  <- read.csv("pml-testing.csv", na.strings = c("NA", "#DIV/0!", ""))
comps <- complete.cases(t(traindata)) & complete.cases(t(testdata))
traindata1 <- traindata[,comps]
testdata1  <- testdata[,comps]
# Drop the first 7 columns as they're unnecessary for predicting.
traindata2<- traindata1[,8:length(colnames(traindata1))]
testdata2 <- testdata1[,8:length(colnames(testdata1))]
```

Cross Validation

```
set.seed(32323)
in.training <- createDataPartition(traindata2$classe, p=0.75, list=F)
Train <- traindata2[in.training, ]
Cross <- traindata2[-in.training, ]
Train$classe = as.factor(Train$classe)
```

Modeling

```
modFit <- randomForest(Train$classe~., data = Train, importance=T)
modFit    #Please, take time. It may takes a few minitues:) #
testdata2$classe = as.factor(testdata2$classe)
pred1 <-predict(modFit,Train)
table(pred1)
table(Train$classe)
confusionMatrix(pred1,Train$classe) #Accuray : 0.9997
```

##Call:
randomForest(formula = Train$classe ~ ., data = Train, importance = T)
Type of random forest: classification
Number of trees: 500
No. of variables tried at each split: 7

```
    OOB estimate of  error rate: 0.46%
```

Confusion matrix:

	A	B	C	D	E	class.error
A	4183	1	0	1	0	0.0004778973
B	7	2837	4	0	0	0.0038623596
C	0	15	2551	1	0	0.0062329568
D	0	0	30	2379	3	0.0136815920
E	0	0	0	5	2701	0.0018477458

The results on the validation set

```
Cross$classe = as.factor(Cross$classe)
valid <- predict(modFit, newdata=Cross)
confusionMatrix(valid,Cross$classe)  #Accuracy : 0.9951
```

The results on the test set

```
testresults <- predict(modFit, newdata=testdata2)
print("Classifications on the test set:"); testresults
```

##[1] "Classifications on the test set:"
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
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

Importance of the variables

```
varImp <- importance(modFit)
varImp[1:10,]
```

In conclusion, 'Random Forest' in this model conducted an entirely accurate prediction. The importance of the variables revealed that yaw_belt, rell_ belt, and pitch_belt are the essential variables in sequence and the upper ten variables seem to decide the classification. After testing 'Course project prediction quiz', I was sure the model is completely accurate.

Appendix, Plot

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

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.