Data loading and processing

```
setwd("C:/Stats/R")

library(caret)

trainURL <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-trainin
g.csv"

testURL <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.
csv"

training <- read.csv(url(trainURL))

testing <- read.csv(url(testURL))



lbl <- createDataPartition(training$classe , p = 0.7, list = FALSE)

train <- training[lbl, ]

test <- training[-lbl, ]
```

Removing columns that contains NA values and irrelevant variables

```
NZeroV <- nearZeroVar(train)

train <- train[ ,-NZeroV]

test <- test[ ,-NZeroV]



lbl <- apply(train, 2, function(x) mean(is.na(x))) > 0.95

train <- train[, -which(lbl, lbl == FALSE)]

test <- test[, -which(lbl, lbl == FALSE)]

train <- train[ , -(1:5)]

test <- test[ , -(1:5)]

library(caret)

## Loading required package: lattice

## Loading required package: ggplot2

set.seed(1704)
```

Building model and cross validation

Modelling with regression tree (“rpart”)

```
f1 <- train(classe ~ ., method="rpart", data=train)
v1 <- predict(fit1, validation)
confusionMatrix(validation$classe, v1)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1530   21  120    0    3
##          B  456  425  258    0    0
##          C  452   36  538    0    0
##          D  429  165  370    0    0
##          E  154  150  291    0  487
##
## Overall Statistics
##
##                Accuracy : 0.5064
##                  95% CI : (0.4935, 0.5192)
##     No Information Rate : 0.5133
##     P-Value [Acc > NIR] : 0.8604
##
##                   Kappa : 0.3554
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.5065  0.53325  0.34115       NA  0.99388
## Specificity            0.9497  0.85967  0.88672   0.8362  0.88971
## Pos Pred Value         0.9140  0.37313  0.52437       NA  0.45009
## Neg Pred Value         0.6459  0.92162  0.78617       NA  0.99938
## Prevalence             0.5133  0.13543  0.26797   0.0000  0.08326
## Detection Rate         0.2600  0.07222  0.09142   0.0000  0.08275
## Detection Prevalence   0.2845  0.19354  0.17434   0.1638  0.18386
## Balanced Accuracy      0.7281  0.69646  0.61394       NA  0.94180
```

random forest (“rf”)

```
set.seed(14807)
control <- trainControl(method = "cv", number = 3, verboseIter=FALSE)
modelRF <- train(classe ~ ., data = train, method = "rf", trControl = contr
ol)
modelRF$finalModel
predictRF <- predict(modelRF, test)
```

```
## Confusion Mtrx and Statis
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1674    0    0    0    0
##          B   14 1121    4    0    0
##          C    0    3 1015    3    0
##          D    0    0   11  952    0
##          E    0    0    1    3 1078
##
## Overall Statistics
##
##                Accuracy : 0.9921
##                  95% CI : (0.991, 0.9953)
##     No Information Rate : 0.2867
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9916
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9917   0.9973   0.9846   0.9937   1.0000
## Specificity            1.0000   0.9962   0.9988   0.9978   0.9992
## Pos Pred Value         1.0000   0.9842   0.9942   0.9886   0.9963
## Neg Pred Value         0.9967   0.9994   0.9967   0.9988   1.0000
## Prevalence             0.2868   0.1910   0.1760   0.1630   0.1832
## Balanced Accuracy      0.9959   0.9968   0.9917   0.9958   0.9996
```

The above result show that the random forest model has the highest accuracy in cross validation. Therefore, we will use the random forest model for predicting test samples.

Prediction

We used the random forest model for prediction

```
pred <- predict(fit2, newdata=testing)
pred
```

```
##  [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
```