Background

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. More information is available from the website here: http://groupware.les.inf.puc-rio.br/har (http://groupware.les.inf.puc-rio.br/har) (see the section on the Weight Lifting Exercise Dataset).

Data

The training data for this project are available here: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv)

The test data are available here: https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv (https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv)

The data for this project come from this source: http://groupware.les.inf.puc-rio.br/har (http://groupware.les.inf.puc-rio.br/har).

Objective

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.

Analysis Approach

I will be using an 8 step approach for this project.

1. Reproducibility: Load necessary packages and set seed
2. Load and explore data set
3. Cross Validation by using 70% of original data for model building (training data) while the rest of 30% will be used for testing (testing data).
4. Data cleaning: To remove variables which have little predictive value or information.
5. Apply PCA to reduce the number of variables.
6. Apply random forest method to build a model.
7. Check the model with the testing data set
8. Apply model to estimate classes of 20 observations.

Step 1: Reproducibility

```
set.seed(1234)
library(caret)
```

```
## Loading required package: lattice
## Loading required package: ggplot2
```

```
library(e1071)
library(randomForest)
```

```
## randomForest 4.6-10
## Type rfNews() to see new features/changes/bug fixes.
```

Step 2:

```
#Results hidden
data <- read.csv("pml-training.csv", na.strings=c("NA","#DIV/0!", ""))
colnames(data)
summary(data)
```

Step 3:

```
#Results hidden
train <- createDataPartition(y=data$classe,p=.70,list=F)
training <- data[train,]
testing <- data[-train,]
```

Step 4:

```
#Results hidden
#Remove variables which have little predicitive value: identifier, timestamp, and window data
Cl <- grep("name|timestamp|window|X", colnames(training), value=F)
trainingCl <- training[,-Cl]

#Remove variables with over 95% missing data
trainingCl[trainingCl==""] <- NA
NArate <- apply(trainingCl, 2, function(x) sum(is.na(x)))/nrow(trainingCl)
trainingCl <- trainingCl[!(NArate>0.95)]
summary(trainingCl)
```

Step 5:

```
#Results hidden
preProc <- preProcess(trainingCl[,1:52],method="pca",thresh=.8) #12 components are required
preProc

preProc <- preProcess(trainingCl[,1:52],method="pca",thresh=.9) #18 components are required
preProc

preProc <- preProcess(trainingCl[,1:52],method="pca",thresh=.95) #25 components are required
preProc

preProc <- preProcess(trainingCl[,1:52],method="pca",pcaComp=25) #Use 25 components to achieve 95% of variance
preProc
```

```
preProc$rotation
trainingPC <- predict(preProc,trainingCl[,1:52])
```

Step 6:

```
#Results hidden
modFitRF <- randomForest(trainingCl$classe ~ .,   data=trainingPC, do.trace=F)
print(modFitRF) # view results
importance(modFitRF) # importance of each predictor
```

Step 7:

```
#Results hidden
testingCl <- testing[,-Cl]
testingCl[testingCl==""] <- NA
NArate <- apply(testingCl, 2, function(x) sum(is.na(x)))/nrow(testingCl)
testingCl <- testingCl[!(NArate>0.95)]
testingPC <- predict(preProc,testingCl[,1:52])
confusionMatrix(testingCl$classe,predict(modFitRF,testingPC))
```

Step 8:

```
#Results hidden
testdata <- read.csv("pml-testing.csv", na.strings=c("NA","#DIV/0!", ""))
testdataCl <- testdata[,-Cl]
testdataCl[testdataCl==""] <- NA
NArate <- apply(testdataCl, 2, function(x) sum(is.na(x)))/nrow(testdataCl)
testdataCl <- testdataCl[!(NArate>0.95)]
testdataPC <- predict(preProc,testdataCl[,1:52])
testdataCl$classe <- predict(modFitRF,testdataPC)
```

Write files for submission

```
pml_write_files = function(x){
  n = length(x)
  for(i in 1:n){
    filename = paste0("problem_id_",i,".txt")
    write.table(x[i],file=filename,quote=FALSE,row.names=FALSE,col.names=FALS
E)
  }
}

pml_write_files(testdataCl$classe)
```

Conclusion

In order to analyze and predict correct body movement during the exercise, 19622 observations from weight lifting exercise were collected. These data are randomly split into 2 sets: 70% (13737 observations) to build a model by random forest method, and the remaining 30% (5885 observations) to be used as the testing set for model validation.

The model statistics showed that the model had the overall accuracy of 97% for the testing set. The sensitivity is 92%-99% and the specificity was over 99% for all classes.

One limitation of the study is that observation data used in the analyses was collected from 6 young healthy participants in an experiment using Microsoft Kinect. Under the same conditions, the model is expected to perform over 95% accuracy. However, under different conditions (e.g. different age group or another measuring device), the model might not be able to perform to that accuracy.