# Human Activity Recognition

*J. Peyton*

*16 Sept 2017*

## Introduction

Weight lifting excercise data has been gathered for Human Activity Recognition (HAR) research. The data contain measurements from accelerometers and gyroscopes attached to individuals as they carry out weight lifting. The individuals were asked to perform the excercise in a range of different ways, intentionally replicating five common weight lifting mistakes, which are classified in the data as Class A, B, C, D or E. The data will be analysed and a model will be fit to it in order to predict the class for a particular measurement. The model will be applied to a test data set and the predictions used in Quiz 4.

### Loading the data

The training and quiz test data is downloaded.

```
if(!file.exists("pml-training.csv")) {
  fileURL <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
  download.file(fileURL, "pml-training.csv")
}
if(!file.exists("pml-testing.csv")) {
  fileURL <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
  download.file(fileURL, "pml-testing.csv")
}

set.seed(15846)

data<-read.csv("pml-training.csv", header=TRUE, colClasses = "character")
nacols<-sum(apply(apply(data, 2, is.na), 2, sum)!=0)
dim<-dim(data)
```

The data contains 160 variables and 19622 observations. Notably, 67 variables contain missing data. The first 6 variables, including user identifiers and time stamps will be discarded from the analysis as it is not relevant to predicting the activity class. The variables will be converted to numeric data before the training and test sets are created. 50% of the data will be used for each set, with the training set used for training the model and the testing set used for cross-validation of the model. Finally, the quiz test set will also be loaded.

```
sub<-data[,c(7:159)]
df<-as.data.table(sub)
df<-as.data.frame(df[, lapply(.SD, as.numeric)])
nacols<-apply(apply(df, 2, is.na),2,sum)
df<-df[,names(nacols[nacols==0])]
data<-mutate(df, classe=data$classe)

trainIndex<-createDataPartition(data$classe, p=.5, list=FALSE)
training<-data[trainIndex,]
testing<-data[-trainIndex,]

quizset<-read.csv("pml-testing.csv", header=TRUE, colClasses = "character")
```

```
quizsub<-quizset[,c(7:159)]
quizdf<-as.data.table(quizsub)
quizdf<-as.data.frame(quizdf[, lapply(.SD, as.numeric)])
quizdf<-quizdf[,names(nacols[nacols==0])]
quizset<-quizdf
```

## Pre-processing

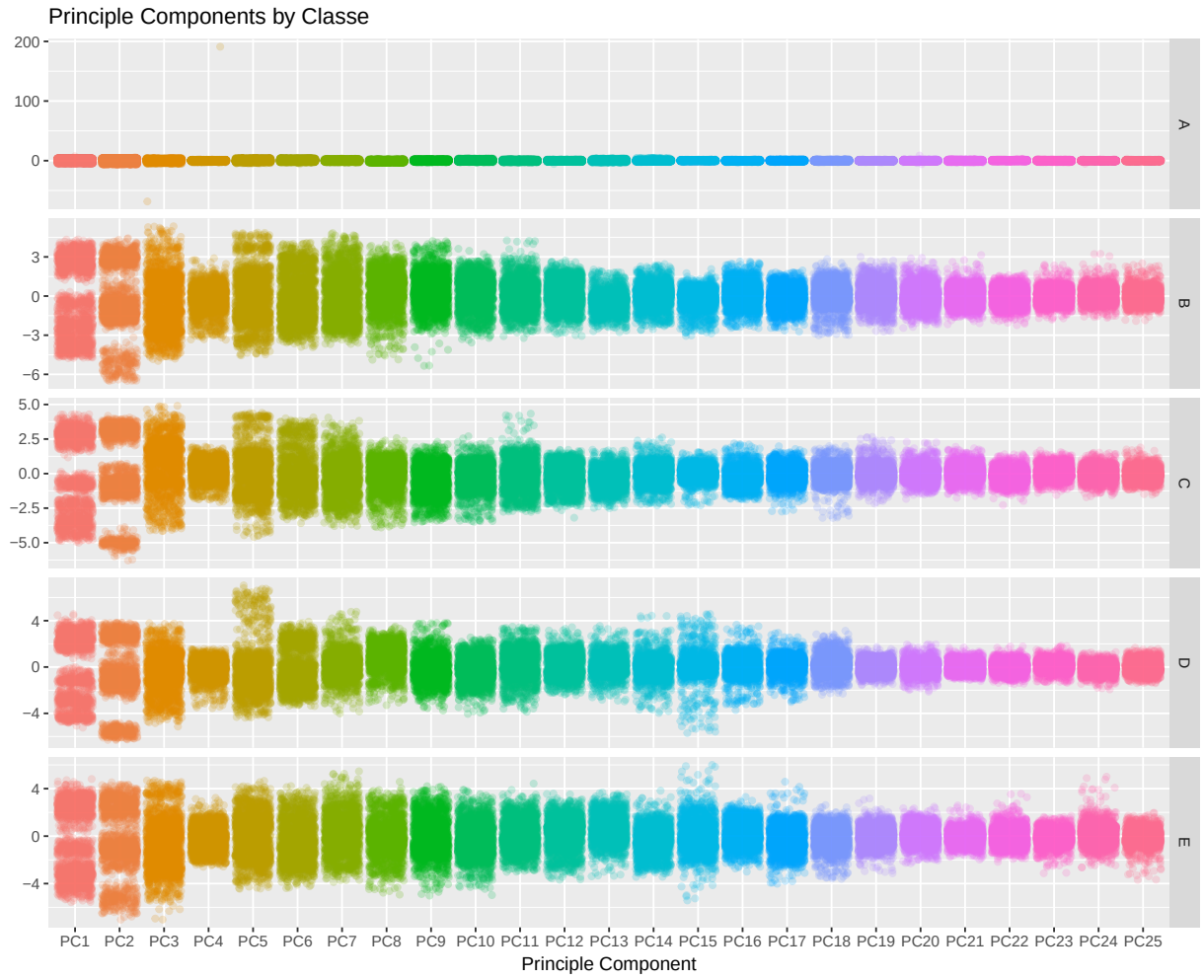
To determine if the data can be simplified before training the model, Principle Components Analysis (PCA) will be performed.

```
preProc<-preProcess(training[,-54], method="pca")
trainPCA<-predict(preProc, training[,-54])
trainPCA<-mutate(trainPCA, classe=training$classe)
numComp<-preProc$numComp
thresh<-preProc$thresh
```

Using PCA shows the number of variables required to explain 0.95 of the variance is 25. As this significantly reduces the number of varialbes involved, the model will be trained using the 25 principle components to reduce computational time.

From the plot below, it is clear that for each class the principle components exibit unique characteristics.

```
melt<-melt(trainPCA, id.vars="classe")
ggplot() +
geom_point(data=melt, aes(x=variable, y=value, colour=variable), alpha=0.2, position="jitter") +
facet_grid(classe~., scales="free") +
theme(legend.position="none") +
labs(title="Principle Components by Classe", x = "Principle Component", y="")
```




## Training

A Random Forest model will be fit to the data. A Random Forest has been selected as it is generally one of the best performing prediction algorithms.

```
modfit<-train(classe~., method="rf", data=trainPCA)
confusionMatrix(training$classe, predict(modfit,trainPCA))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 2790    0    0    0    0
##          B    0 1899    0    0    0
##          C    0    0 1711    0    0
##          D    0    0    0 1608    0
##          E    0    0    0    0 1804
##
## Overall Statistics
##
##                Accuracy : 1
```

```
##                  95% CI : (0.9996, 1)
##     No Information Rate : 0.2843
##     P-Value [Acc > NIR] : < 0.00000000000000022
##
##                   Kappa : 1
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   1.0000   1.0000   1.0000   1.0000
## Specificity            1.0000   1.0000   1.0000   1.0000   1.0000
## Pos Pred Value         1.0000   1.0000   1.0000   1.0000   1.0000
## Neg Pred Value         1.0000   1.0000   1.0000   1.0000   1.0000
## Prevalence             0.2843   0.1935   0.1744   0.1639   0.1839
## Detection Rate         0.2843   0.1935   0.1744   0.1639   0.1839
## Detection Prevalence   0.2843   0.1935   0.1744   0.1639   0.1839
## Balanced Accuracy      1.0000   1.0000   1.0000   1.0000   1.0000
```

```
insampleerror <-  (1 - confusionMatrix(training$classe, predict(modfit,trainPCA))$overall[1]) * 100
```

After fitting the Random Forest model, the in-sample error is 0%.

## Cross-validation

The model will be cross-validated against the training data to determine the expected accuracy.

```
testPCA<-predict(preProc, testing[,-54])
confusionMatrix(testing$classe, predict(modfit,testPCA))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 2748   16   14   12    0
##          B   56 1775   53    3   11
##          C    4   41 1647   17    2
##          D    4    9   71 1520    4
##          E    0    8    9   16 1770
##
## Overall Statistics
##
##                Accuracy : 0.9643
##                  95% CI : (0.9605, 0.9679)
##     No Information Rate : 0.2866
##     P-Value [Acc > NIR] : < 0.00000000000000022
##
##                   Kappa : 0.9549
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9772   0.9600   0.9181   0.9694   0.9905
## Specificity            0.9940   0.9845   0.9920   0.9893   0.9959
```

```
## Pos Pred Value         0.9849   0.9352   0.9626   0.9453   0.9817
## Neg Pred Value         0.9909   0.9906   0.9818   0.9941   0.9979
## Prevalence             0.2866   0.1885   0.1829   0.1598   0.1822
## Detection Rate         0.2801   0.1809   0.1679   0.1549   0.1804
## Detection Prevalence   0.2844   0.1935   0.1744   0.1639   0.1838
## Balanced Accuracy      0.9856   0.9723   0.9550   0.9794   0.9932
```

```
  outofsampleerror <- (1 -  confusionMatrix(testing$classe, predict(modfit,testPCA))$overall[1]) * 100
```

The expected out of sample error is 3.5779817%. This is error to expect when the model is used on data outside the training and test sets.

## Predictions

The results of applying the model to the quiz test data set are given below.

```
  quizPCA<-predict(preProc, quizset)
  quizpredictions<-predict(modfit, quizPCA)
  names(quizpredictions)<-paste(rep("Test ", 20), seq(1:20))
  print(quizpredictions)
```

```
##  Test  1  Test  2  Test  3  Test  4  Test  5  Test  6  Test  7  Test  8
##        B        A        A        A        A        E        D        B
##  Test  9 Test  10 Test  11 Test  12 Test  13 Test  14 Test  15 Test  16
##        A        A        B        C        B        A        E        E
## Test  17 Test  18 Test  19 Test  20
##        A        B        B        B
## Levels: A B C D E
```