Using Machine Learning Technique to predict Health Wellness

### Executive Summary

Thie objective of this project is to analyze and model the data coming from accelerometers on the belt, forearm, arm, and dumbell of 6 subjects participating in a study conducted by **Velloso, E.; Bulling, A.; Gellersen, H.; Ugulino, W.; Fuks, H.** as a part of the research study **Qualitative Activity Recognition of Weight Lifting Exercises**. The subjects were asked to perform one set of 10 repetitions of the Unilateral Dumbbell Biceps Curl in five different fashions: exactly according to the specification (Class A), throwing the elbows to the front (Class B), lifting the dumbbell only halfway (Class C), lowering the dumbbell only halfway (Class D) and throwing the hips to the front (Class E).

### Approach and steps in the project:

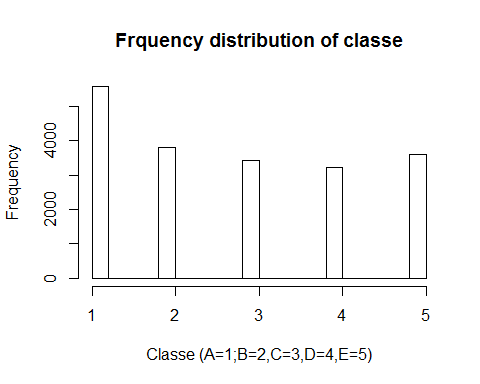
1. First load the training data from the CSV file **pml-training.csv**
2. Decide on a strategy to select significant predictors from the whole bunch of predictors in the training dataset which can be used to predict the outcome variable "classe"
3. Pre-process predictor variables
4. Come up with the predictive model from the training dataset
5. Load the testing data from the CSV file **pml-testing.csv**
6. Apply the model to predict the outcome variable "classe" for each row of the testing dataset

**Load all required libraries first:**

library(dplyr)  
library(caret)

**Step 1 : Load the training and test datasets**

training\_data <- read.csv("./pml-training.csv")  
training\_data <- mutate(training\_data, classe = as.numeric(classe))  
hist(training\_data$classe, main = "Frquency distribution of classe", xlab = "Classe (A=1;B=2,C=3,D=4,E=5)")



testing\_data <- read.csv("./pml-testing.csv")

**Step 2 : Strategize and select significant predictors**

1. First of all, take a look at the training dataset to see if there are fields having all missing values. In that case those variables need to be dropped from the set of predictor variables as anyway they will be of no use while prediction. So, as a first step, filter out all those fields from the testing dataset.
2. Next take out all variables for which it is anyway known that will not have any impact on predicted outcome e.g. user name, time stamp, etc. This is because goal of machine learning classification is to produce a classifier that will predict well *on future data* - on data that is *not in the training set*. This means the goal is to have the classifier work *on new users* who are *not part of that study*. Or for that matter, the timestamps won't be related.
3. The num\_window variable is only in there as bookkeeping info - it is not a predictor. It identifies a short time sequence of samples that are part of the same trial. The experimenters split up the trial (one exercise repetition) into windows, and the window number is in there so they can easily compute the summary statistics (mean, variance, skewness, kurtosis) over each window.
4. Trim down the training dataset to retain only the significant predictors as determined above.

# Finding fields with missing data from testing dataset ...  
  
ColNums\_NotAllMissing <- function(df){as.vector(which(colSums(is.na(df)) != nrow(df)))}  
trim\_testing\_data <- testing\_data %>% select(ColNums\_NotAllMissing(.))  
  
# Taking out all other irrelevant book-keeping fields ...  
  
trim\_testing\_data <- select(trim\_testing\_data, -X, -user\_name, -raw\_timestamp\_part\_1, -raw\_timestamp\_part\_2, -cvtd\_timestamp, -new\_window, -num\_window, -problem\_id)  
  
test\_varnames <- names(trim\_testing\_data)  
  
# Trim down the training dataset to consist only the significant predictors ...  
  
trim\_training\_data1 <- training\_data[test\_varnames]  
trim\_training\_data2 <- select(training\_data, classe)  
trim\_training\_data <- cbind(trim\_training\_data1, trim\_training\_data2)

**Step 3 : Pre-process predictor variables**

preproc <- preProcess(trim\_training\_data[ , -53], method = "pca")

**Step 4 : Train a model**

This is a clear problem of classification as we are trying to predict categories from labeled data. So, I've planned to use prediction with basic decision tree.

trainPC <- predict(preproc, trim\_training\_data[ , -53])  
modelFit <- train(trim\_training\_data$classe ~ ., method = "rpart", data = trainPC)

## Loading required package: rpart

## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info =  
## trainInfo, : There were missing values in resampled performance measures.

**Step 5 : Load the testing data**

Already done above as a part of step 2.

**Step 6 : Predict "classe" for each tect case of test dataset**

testPC <- predict(preproc, trim\_testing\_data)  
predict(modelFit, testPC)

## [1] 2.306074 2.306074 2.306074 2.306074 3.034536 2.306074 3.926483  
## [8] 2.306074 2.306074 2.306074 2.306074 2.306074 3.034536 2.306074  
## [15] 3.926483 3.034536 2.306074 3.034536 2.306074 3.034536