**Project**

**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> (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>

The test data are available here:   
  
<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>. If you use the document you create for this class for any purpose please cite them as they have been very generous in allowing their data to be used for this kind of assignment.

**What you should submit**

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.

1. Your submission should consist of a link to a Github repo with your R markdown and compiled HTML file describing your analysis. Please constrain the text of the writeup to < 2000 words and the number of figures to be less than 5. It will make it easier for the graders if you submit a repo with a gh-pages branch so the HTML page can be viewed online (and you always want to make it easy on graders :-).
2. You should also apply your machine learning algorithm to the 20 test cases available in the test data above. Please submit your predictions in appropriate format to the programming assignment for automated grading. See the programming assignment for additional details.

**Reproducibility**

Due to security concerns with the exchange of R code, your code will not be run during the evaluation by your classmates. Please be sure that if they download the repo, they will be able to view the compiled HTML version of your analysis.

**Prediction Assignment Submission: Instructions**

Please apply the machine learning algorithm you built to each of the 20 test cases in the testing data set. For more information and instructions on how to build your model see the prediction assignment writeup. For each test case you should submit a text file with a single capital letter (A, B, C, D, or E) corresponding to your prediction for the corresponding problem in the test data set. You get 1 point for each correct answer. You may submit up to 2 times for each problem. I know it is a lot of files to submit. It may be helpful to use the following function to create the files. If you have a character vector with your 20 predictions in order for the 20 problems. So something like (note these are not the right answers!):

answers = rep("A", 20)

then you can load this function by copying and pasting it into R:

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=FALSE)

}

}

then create a folder where you want the files to be written. Set that to be your working directory and run:

pml\_write\_files(answers)

and it will create one file for each submission. **Note: if you use this script, please make sure the files that get written out have one character each with your prediction for the corresponding problem ID. I have noticed the script produces strange results if the answers variable is not a character vector.**