# An accessment : How do you know you exercise correctly ?

# 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).

# Dataset

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>.

Download both training dataset :-

set.seed(11111)  
library(caret)

## Warning: package 'caret' was built under R version 3.2.2

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

## Warning: package 'ggplot2' was built under R version 3.2.2

library(rpart)  
library(rpart.plot)

## Warning: package 'rpart.plot' was built under R version 3.2.2

library(RColorBrewer)  
library(rattle)

## Warning: package 'rattle' was built under R version 3.2.2

## Loading required package: RGtk2

## Warning: package 'RGtk2' was built under R version 3.2.2

## Rattle: A free graphical interface for data mining with R.  
## Version 3.5.0 Copyright (c) 2006-2015 Togaware Pty Ltd.  
## Type 'rattle()' to shake, rattle, and roll your data.

library(randomForest)

## Warning: package 'randomForest' was built under R version 3.2.2

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

curdir <-getwd()  
file.url<-'http://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv'  
download.file(file.url,destfile=paste(curdir,'/pml-training.csv',sep=""))  
  
curdir <-getwd()  
file.url<-'http://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv'  
download.file(file.url,destfile=paste(curdir,'/pml-testing.csv',sep=""))

Load both dataset :-

train <- read.csv(paste(curdir,'/pml-training.csv',sep=""),na.strings=c("NA","#DIV/0!",""))  
test <-read.csv(paste(curdir,'/pml-testing.csv',sep=""),na.strings=c("NA","#DIV/0!",""))

Cheking the dimension of training and test dataset :-

dim(train)

## [1] 19622 160

dim(test)

## [1] 20 160

Checking the columns which have all missing values

train<-train[,colSums(is.na(train)) == 0]  
test <-test[,colSums(is.na(test)) == 0]

We remove 6 of the variables which is irrelevant like :-

1. user\_name
2. raw\_timestamp\_part\_1
3. raw\_timestamp\_part\_2
4. cvtd\_timestamp
5. new\_window
6. num\_window

which resides on the column 1-7.

train <-train[,-c(1:7)]  
test <-test[,-c(1:7)]

Check again the dimension

dim(train)

## [1] 19622 53

dim(test)

## [1] 20 53

Now we obtain the several rows to preview

head(train)

## roll\_belt pitch\_belt yaw\_belt total\_accel\_belt gyros\_belt\_x gyros\_belt\_y  
## 1 1.41 8.07 -94.4 3 0.00 0.00  
## 2 1.41 8.07 -94.4 3 0.02 0.00  
## 3 1.42 8.07 -94.4 3 0.00 0.00  
## 4 1.48 8.05 -94.4 3 0.02 0.00  
## 5 1.48 8.07 -94.4 3 0.02 0.02  
## 6 1.45 8.06 -94.4 3 0.02 0.00  
## gyros\_belt\_z accel\_belt\_x accel\_belt\_y accel\_belt\_z magnet\_belt\_x  
## 1 -0.02 -21 4 22 -3  
## 2 -0.02 -22 4 22 -7  
## 3 -0.02 -20 5 23 -2  
## 4 -0.03 -22 3 21 -6  
## 5 -0.02 -21 2 24 -6  
## 6 -0.02 -21 4 21 0  
## magnet\_belt\_y magnet\_belt\_z roll\_arm pitch\_arm yaw\_arm total\_accel\_arm  
## 1 599 -313 -128 22.5 -161 34  
## 2 608 -311 -128 22.5 -161 34  
## 3 600 -305 -128 22.5 -161 34  
## 4 604 -310 -128 22.1 -161 34  
## 5 600 -302 -128 22.1 -161 34  
## 6 603 -312 -128 22.0 -161 34  
## gyros\_arm\_x gyros\_arm\_y gyros\_arm\_z accel\_arm\_x accel\_arm\_y accel\_arm\_z  
## 1 0.00 0.00 -0.02 -288 109 -123  
## 2 0.02 -0.02 -0.02 -290 110 -125  
## 3 0.02 -0.02 -0.02 -289 110 -126  
## 4 0.02 -0.03 0.02 -289 111 -123  
## 5 0.00 -0.03 0.00 -289 111 -123  
## 6 0.02 -0.03 0.00 -289 111 -122  
## magnet\_arm\_x magnet\_arm\_y magnet\_arm\_z roll\_dumbbell pitch\_dumbbell  
## 1 -368 337 516 13.05217 -70.49400  
## 2 -369 337 513 13.13074 -70.63751  
## 3 -368 344 513 12.85075 -70.27812  
## 4 -372 344 512 13.43120 -70.39379  
## 5 -374 337 506 13.37872 -70.42856  
## 6 -369 342 513 13.38246 -70.81759  
## yaw\_dumbbell total\_accel\_dumbbell gyros\_dumbbell\_x gyros\_dumbbell\_y  
## 1 -84.87394 37 0 -0.02  
## 2 -84.71065 37 0 -0.02  
## 3 -85.14078 37 0 -0.02  
## 4 -84.87363 37 0 -0.02  
## 5 -84.85306 37 0 -0.02  
## 6 -84.46500 37 0 -0.02  
## gyros\_dumbbell\_z accel\_dumbbell\_x accel\_dumbbell\_y accel\_dumbbell\_z  
## 1 0.00 -234 47 -271  
## 2 0.00 -233 47 -269  
## 3 0.00 -232 46 -270  
## 4 -0.02 -232 48 -269  
## 5 0.00 -233 48 -270  
## 6 0.00 -234 48 -269  
## magnet\_dumbbell\_x magnet\_dumbbell\_y magnet\_dumbbell\_z roll\_forearm  
## 1 -559 293 -65 28.4  
## 2 -555 296 -64 28.3  
## 3 -561 298 -63 28.3  
## 4 -552 303 -60 28.1  
## 5 -554 292 -68 28.0  
## 6 -558 294 -66 27.9  
## pitch\_forearm yaw\_forearm total\_accel\_forearm gyros\_forearm\_x  
## 1 -63.9 -153 36 0.03  
## 2 -63.9 -153 36 0.02  
## 3 -63.9 -152 36 0.03  
## 4 -63.9 -152 36 0.02  
## 5 -63.9 -152 36 0.02  
## 6 -63.9 -152 36 0.02  
## gyros\_forearm\_y gyros\_forearm\_z accel\_forearm\_x accel\_forearm\_y  
## 1 0.00 -0.02 192 203  
## 2 0.00 -0.02 192 203  
## 3 -0.02 0.00 196 204  
## 4 -0.02 0.00 189 206  
## 5 0.00 -0.02 189 206  
## 6 -0.02 -0.03 193 203  
## accel\_forearm\_z magnet\_forearm\_x magnet\_forearm\_y magnet\_forearm\_z  
## 1 -215 -17 654 476  
## 2 -216 -18 661 473  
## 3 -213 -18 658 469  
## 4 -214 -16 658 469  
## 5 -214 -17 655 473  
## 6 -215 -9 660 478  
## classe  
## 1 A  
## 2 A  
## 3 A  
## 4 A  
## 5 A  
## 6 A

head(test)

## roll\_belt pitch\_belt yaw\_belt total\_accel\_belt gyros\_belt\_x gyros\_belt\_y  
## 1 123.00 27.00 -4.75 20 -0.50 -0.02  
## 2 1.02 4.87 -88.90 4 -0.06 -0.02  
## 3 0.87 1.82 -88.50 5 0.05 0.02  
## 4 125.00 -41.60 162.00 17 0.11 0.11  
## 5 1.35 3.33 -88.60 3 0.03 0.02  
## 6 -5.92 1.59 -87.70 4 0.10 0.05  
## gyros\_belt\_z accel\_belt\_x accel\_belt\_y accel\_belt\_z magnet\_belt\_x  
## 1 -0.46 -38 69 -179 -13  
## 2 -0.07 -13 11 39 43  
## 3 0.03 1 -1 49 29  
## 4 -0.16 46 45 -156 169  
## 5 0.00 -8 4 27 33  
## 6 -0.13 -11 -16 38 31  
## magnet\_belt\_y magnet\_belt\_z roll\_arm pitch\_arm yaw\_arm total\_accel\_arm  
## 1 581 -382 40.7 -27.80 178 10  
## 2 636 -309 0.0 0.00 0 38  
## 3 631 -312 0.0 0.00 0 44  
## 4 608 -304 -109.0 55.00 -142 25  
## 5 566 -418 76.1 2.76 102 29  
## 6 638 -291 0.0 0.00 0 14  
## gyros\_arm\_x gyros\_arm\_y gyros\_arm\_z accel\_arm\_x accel\_arm\_y accel\_arm\_z  
## 1 -1.65 0.48 -0.18 16 38 93  
## 2 -1.17 0.85 -0.43 -290 215 -90  
## 3 2.10 -1.36 1.13 -341 245 -87  
## 4 0.22 -0.51 0.92 -238 -57 6  
## 5 -1.96 0.79 -0.54 -197 200 -30  
## 6 0.02 0.05 -0.07 -26 130 -19  
## magnet\_arm\_x magnet\_arm\_y magnet\_arm\_z roll\_dumbbell pitch\_dumbbell  
## 1 -326 385 481 -17.73748 24.96085  
## 2 -325 447 434 54.47761 -53.69758  
## 3 -264 474 413 57.07031 -51.37303  
## 4 -173 257 633 43.10927 -30.04885  
## 5 -170 275 617 -101.38396 -53.43952  
## 6 396 176 516 62.18750 -50.55595  
## yaw\_dumbbell total\_accel\_dumbbell gyros\_dumbbell\_x gyros\_dumbbell\_y  
## 1 126.23596 9 0.64 0.06  
## 2 -75.51480 31 0.34 0.05  
## 3 -75.20287 29 0.39 0.14  
## 4 -103.32003 18 0.10 -0.02  
## 5 -14.19542 4 0.29 -0.47  
## 6 -71.12063 29 -0.59 0.80  
## gyros\_dumbbell\_z accel\_dumbbell\_x accel\_dumbbell\_y accel\_dumbbell\_z  
## 1 -0.61 21 -15 81  
## 2 -0.71 -153 155 -205  
## 3 -0.34 -141 155 -196  
## 4 0.05 -51 72 -148  
## 5 -0.46 -18 -30 -5  
## 6 1.10 -138 166 -186  
## magnet\_dumbbell\_x magnet\_dumbbell\_y magnet\_dumbbell\_z roll\_forearm  
## 1 523 -528 -56 141  
## 2 -502 388 -36 109  
## 3 -506 349 41 131  
## 4 -576 238 53 0  
## 5 -424 252 312 -176  
## 6 -543 262 96 150  
## pitch\_forearm yaw\_forearm total\_accel\_forearm gyros\_forearm\_x  
## 1 49.30 156.0 33 0.74  
## 2 -17.60 106.0 39 1.12  
## 3 -32.60 93.0 34 0.18  
## 4 0.00 0.0 43 1.38  
## 5 -2.16 -47.9 24 -0.75  
## 6 1.46 89.7 43 -0.88  
## gyros\_forearm\_y gyros\_forearm\_z accel\_forearm\_x accel\_forearm\_y  
## 1 -3.34 -0.59 -110 267  
## 2 -2.78 -0.18 212 297  
## 3 -0.79 0.28 154 271  
## 4 0.69 1.80 -92 406  
## 5 3.10 0.80 131 -93  
## 6 4.26 1.35 230 322  
## accel\_forearm\_z magnet\_forearm\_x magnet\_forearm\_y magnet\_forearm\_z  
## 1 -149 -714 419 617  
## 2 -118 -237 791 873  
## 3 -129 -51 698 783  
## 4 -39 -233 783 521  
## 5 172 375 -787 91  
## 6 -144 -300 800 884  
## problem\_id  
## 1 1  
## 2 2  
## 3 3  
## 4 4  
## 5 5  
## 6 6

In order to run cross-validation , the training dataset need to partition into 2 sets . We set the 1st partition for training dataset to 75% and test dataset to 25%. Training dataset contains 53 variables with 19622 obs and test dataset contains 53 variables with 20 obs.

This will do the randomize sub-sampling without replacement

PartTrain <- createDataPartition(y=train$classe, p=0.75, list=FALSE)  
train\_part <- train[PartTrain, ];   
test\_part <- train[-PartTrain, ]  
dim(train\_part)

## [1] 14718 53

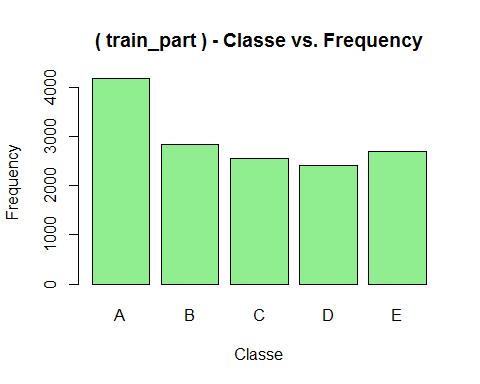
dim(test\_part)

## [1] 4904 53

# Visualization

We try to plot into the histogram to see the trending frequency of each sub-training & test dataset by comparing with each other. The variable classe contains 5 levels which is A,B,C,D & E

plot(train\_part$classe, col="lightgreen",   
main="( train\_part ) - Classe vs. Frequency",   
xlab="Classe", ylab="Frequency")



The graph above shows that A ~ 4000x occurrences is most frequent while D is the lest frequent ~ 2500x occurrences

# Decision Tree

We choose Decision Tree machine learning algorithm as a support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility.

fit\_model <- rpart(classe ~ ., data=train\_part, method="class")  
fit\_model

## n= 14718   
##   
## node), split, n, loss, yval, (yprob)  
## \* denotes terminal node  
##   
## 1) root 14718 10533 A (0.28 0.19 0.17 0.16 0.18)   
## 2) roll\_belt< 130.5 13487 9311 A (0.31 0.21 0.19 0.18 0.11)   
## 4) pitch\_forearm< -33.95 1173 7 A (0.99 0.006 0 0 0) \*  
## 5) pitch\_forearm>=-33.95 12314 9304 A (0.24 0.23 0.21 0.2 0.12)   
## 10) magnet\_dumbbell\_y< 439.5 10462 7512 A (0.28 0.18 0.24 0.19 0.11)   
## 20) roll\_forearm< 124.5 6549 3896 A (0.41 0.18 0.18 0.17 0.061)   
## 40) magnet\_dumbbell\_z< -27.5 2216 750 A (0.66 0.21 0.011 0.082 0.032)   
## 80) roll\_forearm>=-136.5 1839 418 A (0.77 0.17 0.012 0.033 0.0076) \*  
## 81) roll\_forearm< -136.5 377 225 B (0.12 0.4 0.008 0.32 0.15) \*  
## 41) magnet\_dumbbell\_z>=-27.5 4333 3146 A (0.27 0.17 0.27 0.21 0.076)   
## 82) accel\_dumbbell\_y>=-40.5 3815 2632 A (0.31 0.19 0.19 0.24 0.081)   
## 164) yaw\_belt>=169.5 529 68 A (0.87 0.059 0 0.068 0.0019) \*  
## 165) yaw\_belt< 169.5 3286 2417 D (0.22 0.21 0.22 0.26 0.093)   
## 330) pitch\_belt< -42.95 394 68 B (0.018 0.83 0.089 0.038 0.028) \*  
## 331) pitch\_belt>=-42.95 2892 2038 D (0.25 0.12 0.23 0.3 0.1)   
## 662) roll\_belt>=125.5 673 280 C (0.38 0.024 0.58 0.0074 0.0015)   
## 1324) magnet\_belt\_z< -323.5 230 6 A (0.97 0 0.022 0 0.0043) \*  
## 1325) magnet\_belt\_z>=-323.5 443 55 C (0.077 0.036 0.88 0.011 0) \*  
## 663) roll\_belt< 125.5 2219 1370 D (0.21 0.15 0.13 0.38 0.13)   
## 1326) yaw\_belt< -85.55 1387 1055 A (0.24 0.22 0.14 0.21 0.19)   
## 2652) yaw\_forearm>=-95.15 1114 782 A (0.3 0.25 0.14 0.09 0.22)   
## 5304) accel\_dumbbell\_z< 21.5 625 315 A (0.5 0.19 0.24 0.046 0.027)   
## 10608) magnet\_forearm\_z>=-159.5 403 97 A (0.76 0.15 0.025 0.03 0.032) \*  
## 10609) magnet\_forearm\_z< -159.5 222 81 C (0.018 0.25 0.64 0.077 0.018) \*  
## 5305) accel\_dumbbell\_z>=21.5 489 259 E (0.045 0.34 0 0.15 0.47)   
## 10610) roll\_dumbbell< 36.19782 150 25 B (0.02 0.83 0 0.02 0.13) \*  
## 10611) roll\_dumbbell>=36.19782 339 128 E (0.056 0.12 0 0.2 0.62) \*  
## 2653) yaw\_forearm< -95.15 273 76 D (0 0.062 0.16 0.72 0.051) \*  
## 1327) yaw\_belt>=-85.55 832 280 D (0.15 0.046 0.1 0.66 0.041)   
## 2654) yaw\_arm< -102.85 114 1 A (0.99 0 0.0088 0 0) \*  
## 2655) yaw\_arm>=-102.85 718 166 D (0.017 0.053 0.11 0.77 0.047) \*  
## 83) accel\_dumbbell\_y< -40.5 518 64 C (0.0077 0.037 0.88 0.035 0.044) \*  
## 21) roll\_forearm>=124.5 3913 2612 C (0.076 0.18 0.33 0.23 0.19)   
## 42) magnet\_dumbbell\_y< 290.5 2268 1155 C (0.093 0.13 0.49 0.14 0.14)   
## 84) magnet\_forearm\_z< -251 175 36 A (0.79 0.074 0 0.023 0.11) \*  
## 85) magnet\_forearm\_z>=-251 2093 980 C (0.034 0.13 0.53 0.15 0.15)   
## 170) pitch\_belt>=26.15 145 33 B (0.12 0.77 0.028 0 0.076) \*  
## 171) pitch\_belt< 26.15 1948 839 C (0.027 0.086 0.57 0.17 0.15) \*  
## 43) magnet\_dumbbell\_y>=290.5 1645 1082 D (0.053 0.25 0.11 0.34 0.24)   
## 86) accel\_forearm\_x>=-90.5 1015 666 E (0.05 0.32 0.15 0.13 0.34)   
## 172) magnet\_arm\_y>=188.5 417 184 B (0.0096 0.56 0.23 0.091 0.11) \*  
## 173) magnet\_arm\_y< 188.5 598 296 E (0.079 0.15 0.1 0.17 0.51) \*  
## 87) accel\_forearm\_x< -90.5 630 204 D (0.057 0.13 0.049 0.68 0.084) \*  
## 11) magnet\_dumbbell\_y>=439.5 1852 912 B (0.032 0.51 0.043 0.23 0.19)   
## 22) total\_accel\_dumbbell>=5.5 1317 457 B (0.046 0.65 0.059 0.02 0.22)   
## 44) roll\_belt>=-0.58 1120 260 B (0.054 0.77 0.07 0.023 0.086) \*  
## 45) roll\_belt< -0.58 197 0 E (0 0 0 0 1) \*  
## 23) total\_accel\_dumbbell< 5.5 535 143 D (0 0.15 0.0037 0.73 0.11) \*  
## 3) roll\_belt>=130.5 1231 9 E (0.0073 0 0 0 0.99) \*

Displays the (Complexity) cp table for fitted model .

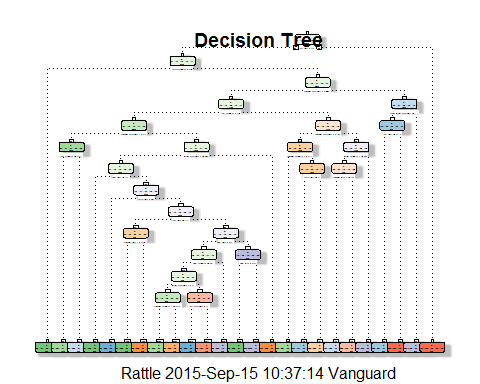
printcp(fit\_model)

##   
## Classification tree:  
## rpart(formula = classe ~ ., data = train\_part, method = "class")  
##   
## Variables actually used in tree construction:  
## [1] accel\_dumbbell\_y accel\_dumbbell\_z accel\_forearm\_x   
## [4] magnet\_arm\_y magnet\_belt\_z magnet\_dumbbell\_y   
## [7] magnet\_dumbbell\_z magnet\_forearm\_z pitch\_belt   
## [10] pitch\_forearm roll\_belt roll\_dumbbell   
## [13] roll\_forearm total\_accel\_dumbbell yaw\_arm   
## [16] yaw\_belt yaw\_forearm   
##   
## Root node error: 10533/14718 = 0.71565  
##   
## n= 14718   
##   
## CP nsplit rel error xerror xstd  
## 1 0.115162 0 1.00000 1.00000 0.0051957  
## 2 0.059622 1 0.88484 0.88541 0.0055494  
## 3 0.035602 4 0.70597 0.74547 0.0057460  
## 4 0.029621 5 0.67037 0.66467 0.0057521  
## 5 0.021361 6 0.64075 0.62508 0.0057269  
## 6 0.020792 11 0.51771 0.53290 0.0055945  
## 7 0.020127 12 0.49691 0.52682 0.0055820  
## 8 0.018703 13 0.47679 0.50109 0.0055239  
## 9 0.017659 14 0.45808 0.48030 0.0054704  
## 10 0.013924 15 0.44043 0.46160 0.0054173  
## 11 0.013197 18 0.39865 0.42533 0.0052999  
## 12 0.013007 19 0.38546 0.42220 0.0052889  
## 13 0.010728 20 0.37245 0.39343 0.0051803  
## 14 0.010253 21 0.36172 0.36998 0.0050819  
## 15 0.010159 22 0.35147 0.36533 0.0050612  
## 16 0.010064 23 0.34131 0.35792 0.0050276  
## 17 0.010000 24 0.33124 0.35792 0.0050276

To visualize the decision tree , we use this fancyRpartPlot command below :-

fancyRpartPlot(fit\_model,main="Decision Tree")

## Warning: labs do not fit even at cex 0.15, there may be some overplotting



Green nodes represent individuals classified by the tree as A, blue nodes are those classified as B and orange nodes are classified as C. The gradient is a visual representation of the three numbers in the middle of the nodes: bearing in mind that levels of a factor are by default in alphabetical order, the first of these three numbers is the proportion of individuals in that node that were actually classified as the first level, (A), in my data; the second number is the proportion that were actually classified as B, and the third the proportion that were C.

# Reference

Velloso, E.; Bulling, A.; Gellersen, H.; Ugulino, W.; Fuks, H. Qualitative Activity Recognition of Weight Lifting Exercises. Proceedings of 4th International Conference in Cooperation with SIGCHI (Augmented Human '13) . Stuttgart, Germany: ACM SIGCHI, 2013. Read more: <http://groupware.les.inf.puc-rio.br/har#ixzz3lj0hACeI>