# Practical Machine Learning Course Project

*Marc Vaglio-Laurin*

*Wednesday, Sept 23, 2015*

## Introduction

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. These different techniques or classes are defined as:

Class A - exercise performed exactly according to the specification Class B - exercise performed incorrectly; subject throwing the elbows to the front Class C - exercise performed incorrectly; subject lifting the dumbbell only halfway Class D - exercise performed incorrectly; subject lowering the dumbbell only halfway Class E - exercise performed incorrectly; subject throwing the hips to the front

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

The goal of this project is to predict how well each participant performs each exercise. We will use 2 different modeling techniques - CART and Random Forest - as well as partition our data into training and testing to help increase the accuracy of model predictions.

## Load required libraries

```
library(knitr)
library(caret)
```

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

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

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

## Load source data

```
testing<-read.csv("C:/Users/staples/Documents/Coursera/PracticalMachineLearning/Data/pml-testing.csv")
training<-read.csv("C:/Users/staples/Documents/Coursera/PracticalMachineLearning/Data/pml-training.csv")
```

## Preview source data

**Let's take a quick look at the data we will be analyzing.**

```r
tail(training)
```

```
##           X user_name raw_timestamp_part_1 raw_timestamp_part_2
## 19617 19617    adelmo           1322832937               588324
## 19618 19618    adelmo           1322832937               588376
## 19619 19619    adelmo           1322832937               596287
## 19620 19620    adelmo           1322832937               636283
## 19621 19621    adelmo           1322832937               964299
## 19622 19622    adelmo           1322832937               972293
##         cvtd_timestamp new_window num_window roll_belt pitch_belt yaw_belt
## 19617 02/12/2011 13:35         no        864       148      -34.7      129
## 19618 02/12/2011 13:35         no        864       147      -34.8      129
## 19619 02/12/2011 13:35         no        864       145      -35.3      130
## 19620 02/12/2011 13:35         no        864       145      -35.5      130
## 19621 02/12/2011 13:35         no        864       143      -35.9      131
## 19622 02/12/2011 13:35        yes        864       143      -36.0      132
##       total_accel_belt kurtosis_roll_belt kurtosis_picth_belt
## 19617               21
## 19618               21
## 19619               19
## 19620               19
## 19621               18
## 19622               18          -1.175902           -1.063259
##       kurtosis_yaw_belt skewness_roll_belt skewness_roll_belt.1
## 19617
## 19618
## 19619
## 19620
## 19621
## 19622           #DIV/0!           0.196860            -0.572396
##       skewness_yaw_belt max_roll_belt max_picth_belt max_yaw_belt
## 19617                              NA             NA
## 19618                              NA             NA
## 19619                              NA             NA
## 19620                              NA             NA
## 19621                              NA             NA
## 19622           #DIV/0!           132             25         -1.2
##       min_roll_belt min_pitch_belt min_yaw_belt amplitude_roll_belt
## 19617            NA             NA                               NA
## 19618            NA             NA                               NA
## 19619            NA             NA                               NA
## 19620            NA             NA                               NA
## 19621            NA             NA                               NA
## 19622           123             18         -1.2                   9
##       amplitude_pitch_belt amplitude_yaw_belt var_total_accel_belt
## 19617                   NA                                      NA
## 19618                   NA                                      NA
## 19619                   NA                                      NA
## 19620                   NA                                      NA
```

```
## 19621                  NA                                      NA
## 19622                   7              0.00             5.6268
##       avg_roll_belt stddev_roll_belt var_roll_belt avg_pitch_belt
## 19617            NA               NA            NA             NA
## 19618            NA               NA            NA             NA
## 19619            NA               NA            NA             NA
## 19620            NA               NA            NA             NA
## 19621            NA               NA            NA             NA
## 19622      151.1481           4.7532       22.5926       -33.6259
##       stddev_pitch_belt var_pitch_belt avg_yaw_belt stddev_yaw_belt
## 19617                NA             NA           NA              NA
## 19618                NA             NA           NA              NA
## 19619                NA             NA           NA              NA
## 19620                NA             NA           NA              NA
## 19621                NA             NA           NA              NA
## 19622            1.3952         1.9466     126.8889          2.7503
##       var_yaw_belt gyros_belt_x gyros_belt_y gyros_belt_z accel_belt_x
## 19617           NA         0.37         0.00        -0.62           49
## 19618           NA         0.37        -0.02        -0.67           50
## 19619           NA         0.39        -0.02        -0.67           47
## 19620           NA         0.37         0.00        -0.64           47
## 19621           NA         0.37        -0.02        -0.59           46
## 19622       7.5641         0.35        -0.02        -0.57           42
##       accel_belt_y accel_belt_z magnet_belt_x magnet_belt_y magnet_belt_z
## 19617           25         -195           191           540          -415
## 19618           26         -193           190           552          -412
## 19619           15         -179           192           558          -389
## 19620           13         -177           191           560          -386
## 19621           18         -172           190           565          -370
## 19622           25         -171           194           566          -349
##       roll_arm pitch_arm yaw_arm total_accel_arm var_accel_arm
## 19617    -99.1     -33.7    79.4              48            NA
## 19618    -99.4     -33.8    79.0              47            NA
## 19619    -99.6     -34.5    77.3              45            NA
## 19620    -99.6     -35.1    76.3              44            NA
## 19621    -98.6     -36.7    73.5              41            NA
## 19622    -97.6     -37.7    71.5              41       54.2564
##       avg_roll_arm stddev_roll_arm var_roll_arm avg_pitch_arm
## 19617           NA              NA           NA            NA
## 19618           NA              NA           NA            NA
## 19619           NA              NA           NA            NA
## 19620           NA              NA           NA            NA
## 19621           NA              NA           NA            NA
## 19622     -91.6481          9.1687      84.0649      -37.6519
##       stddev_pitch_arm var_pitch_arm avg_yaw_arm stddev_yaw_arm
## 19617               NA            NA          NA             NA
## 19618               NA            NA          NA             NA
## 19619               NA            NA          NA             NA
## 19620               NA            NA          NA             NA
## 19621               NA            NA          NA             NA
## 19622           3.6161       13.0764     66.3111        15.4797
##       var_yaw_arm gyros_arm_x gyros_arm_y gyros_arm_z accel_arm_x
## 19617          NA        0.31       -0.45        0.28          67
## 19618          NA        0.55       -0.51        0.25          75
```

```
## 19619          NA        0.88       -0.71        0.21          52
## 19620          NA        0.98       -0.82        0.23          62
## 19621          NA        1.35       -1.00        0.49          70
## 19622     239.621        1.51       -1.06        0.59          58
##       accel_arm_y accel_arm_z magnet_arm_x magnet_arm_y magnet_arm_z
## 19617        -181        -432          268         -138         -566
## 19618        -184        -415          272         -134         -562
## 19619        -163        -406          288         -112         -559
## 19620        -167        -391          309         -103         -541
## 19621        -164        -359          339          -91         -543
## 19622        -152        -365          362          -84         -539
##       kurtosis_roll_arm kurtosis_picth_arm kurtosis_yaw_arm
## 19617
## 19618
## 19619
## 19620
## 19621
## 19622          -1.32631            0.50959         -0.62736
##       skewness_roll_arm skewness_pitch_arm skewness_yaw_arm max_roll_arm
## 19617                                                                 NA
## 19618                                                                 NA
## 19619                                                                 NA
## 19620                                                                 NA
## 19621                                                                 NA
## 19622          -0.51721           -1.26872         -0.77150        -33.7
##       max_picth_arm max_yaw_arm min_roll_arm min_pitch_arm min_yaw_arm
## 19617            NA          NA           NA            NA          NA
## 19618            NA          NA           NA            NA          NA
## 19619            NA          NA           NA            NA          NA
## 19620            NA          NA           NA            NA          NA
## 19621            NA          NA           NA            NA          NA
## 19622          79.5          49        -43.5          27.5          25
##       amplitude_roll_arm amplitude_pitch_arm amplitude_yaw_arm
## 19617                 NA                  NA                NA
## 19618                 NA                  NA                NA
## 19619                 NA                  NA                NA
## 19620                 NA                  NA                NA
## 19621                 NA                  NA                NA
## 19622                9.8                  52                24
##       roll_dumbbell pitch_dumbbell yaw_dumbbell kurtosis_roll_dumbbell
## 19617      38.60998      -22.79150    -111.6131
## 19618      36.41318      -22.86197    -113.4998
## 19619      35.15281      -22.97191    -114.5256
## 19620      30.06028      -20.99018    -120.0318
## 19621      22.86333      -21.75662    -125.2459
## 19622      20.80000      -19.70000    -128.2000                -1.1322
##       kurtosis_picth_dumbbell kurtosis_yaw_dumbbell skewness_roll_dumbbell
## 19617
## 19618
## 19619
## 19620
## 19621
## 19622                 -0.7225               #DIV/0!                 0.0955
##       skewness_pitch_dumbbell skewness_yaw_dumbbell max_roll_dumbbell
```

```
## 19617                                                             NA
## 19618                                                             NA
## 19619                                                             NA
## 19620                                                             NA
## 19621                                                             NA
## 19622                 0.1057              #DIV/0!             -19.7
##       max_picth_dumbbell max_yaw_dumbbell min_roll_dumbbell
## 19617                 NA                                 NA
## 19618                 NA                                 NA
## 19619                 NA                                 NA
## 19620                 NA                                 NA
## 19621                 NA                                 NA
## 19622                -92             -1.1             -33.1
##       min_pitch_dumbbell min_yaw_dumbbell amplitude_roll_dumbbell
## 19617                 NA                                       NA
## 19618                 NA                                       NA
## 19619                 NA                                       NA
## 19620                 NA                                       NA
## 19621                 NA                                       NA
## 19622             -128.2             -1.1                   13.41
##       amplitude_pitch_dumbbell amplitude_yaw_dumbbell total_accel_dumbbell
## 19617                       NA                                          19
## 19618                       NA                                          19
## 19619                       NA                                          18
## 19620                       NA                                          19
## 19621                       NA                                          19
## 19622                     36.2                   0.00                   19
##       var_accel_dumbbell avg_roll_dumbbell stddev_roll_dumbbell
## 19617                 NA                NA                   NA
## 19618                 NA                NA                   NA
## 19619                 NA                NA                   NA
## 19620                 NA                NA                   NA
## 19621                 NA                NA                   NA
## 19622             0.4217           37.3418               9.7828
##       var_roll_dumbbell avg_pitch_dumbbell stddev_pitch_dumbbell
## 19617                NA                 NA                    NA
## 19618                NA                 NA                    NA
## 19619                NA                 NA                    NA
## 19620                NA                 NA                    NA
## 19621                NA                 NA                    NA
## 19622           95.7038           -26.8182                4.0098
##       var_pitch_dumbbell avg_yaw_dumbbell stddev_yaw_dumbbell
## 19617                 NA               NA                  NA
## 19618                 NA               NA                  NA
## 19619                 NA               NA                  NA
## 19620                 NA               NA                  NA
## 19621                 NA               NA                  NA
## 19622            16.0788        -109.9671              9.7475
##       var_yaw_dumbbell gyros_dumbbell_x gyros_dumbbell_y gyros_dumbbell_z
## 19617               NA             0.34            -0.31            -0.51
## 19618               NA             0.32            -0.26            -0.36
## 19619               NA             0.24            -0.24             0.05
## 19620               NA             0.22            -0.27             0.21
## 19621               NA             0.13            -0.14             0.34
```

```
## 19622          95.0143             0.02             0.02             0.36
##       accel_dumbbell_x accel_dumbbell_y accel_dumbbell_z magnet_dumbbell_x
## 19617              -42               70             -167              -624
## 19618              -42               66             -168              -618
## 19619              -41               62             -164              -618
## 19620              -38               54             -170              -621
## 19621              -40               42             -176              -628
## 19622              -36               38             -176              -627
##       magnet_dumbbell_y magnet_dumbbell_z roll_forearm pitch_forearm
## 19617               127                 8            0             0
## 19618               134                 0            0             0
## 19619               116                 7            0             0
## 19620               113                -9            0             0
## 19621               116                 0            0             0
## 19622               119                 2            0             0
##       yaw_forearm kurtosis_roll_forearm kurtosis_picth_forearm
## 19617           0
## 19618           0
## 19619           0
## 19620           0
## 19621           0
## 19622           0               #DIV/0!                #DIV/0!
##       kurtosis_yaw_forearm skewness_roll_forearm skewness_pitch_forearm
## 19617
## 19618
## 19619
## 19620
## 19621
## 19622              #DIV/0!               #DIV/0!                #DIV/0!
##       skewness_yaw_forearm max_roll_forearm max_picth_forearm
## 19617                                    NA                NA
## 19618                                    NA                NA
## 19619                                    NA                NA
## 19620                                    NA                NA
## 19621                                    NA                NA
## 19622              #DIV/0!                0                 0
##       max_yaw_forearm min_roll_forearm min_pitch_forearm min_yaw_forearm
## 19617                               NA                NA
## 19618                               NA                NA
## 19619                               NA                NA
## 19620                               NA                NA
## 19621                               NA                NA
## 19622         #DIV/0!                0                 0         #DIV/0!
##       amplitude_roll_forearm amplitude_pitch_forearm amplitude_yaw_forearm
## 19617                     NA                      NA
## 19618                     NA                      NA
## 19619                     NA                      NA
## 19620                     NA                      NA
## 19621                     NA                      NA
## 19622                      0                       0               #DIV/0!
##       total_accel_forearm var_accel_forearm avg_roll_forearm
## 19617                  27                NA               NA
## 19618                  29                NA               NA
## 19619                  29                NA               NA
```

```
## 19620                  29              NA              NA
## 19621                  32              NA              NA
## 19622                  33        30.10541               0
##       stddev_roll_forearm var_roll_forearm avg_pitch_forearm
## 19617                  NA               NA                NA
## 19618                  NA               NA                NA
## 19619                  NA               NA                NA
## 19620                  NA               NA                NA
## 19621                  NA               NA                NA
## 19622                   0                0                 0
##       stddev_pitch_forearm var_pitch_forearm avg_yaw_forearm
## 19617                   NA                NA              NA
## 19618                   NA                NA              NA
## 19619                   NA                NA              NA
## 19620                   NA                NA              NA
## 19621                   NA                NA              NA
## 19622                    0                 0               0
##       stddev_yaw_forearm var_yaw_forearm gyros_forearm_x gyros_forearm_y
## 19617                 NA              NA            1.75           -1.91
## 19618                 NA              NA            1.73           -1.75
## 19619                 NA              NA            1.59           -1.36
## 19620                 NA              NA            1.54           -1.20
## 19621                 NA              NA            1.48           -0.90
## 19622                  0               0            1.38           -0.64
##       gyros_forearm_z accel_forearm_x accel_forearm_y accel_forearm_z
## 19617           -0.38            -255             -50             -30
## 19618           -0.25            -271             -68             -37
## 19619            0.00            -271             -91             -43
## 19620            0.05            -263             -99             -45
## 19621            0.05            -270            -141             -51
## 19622            0.08            -278            -159             -52
##       magnet_forearm_x magnet_forearm_y magnet_forearm_z classe
## 19617             -226             -570               27      E
## 19618             -205             -587                6      E
## 19619             -151             -635              -36      E
## 19620             -116             -654              -70      E
## 19621              -68             -678              -98      E
## 19622              -60             -686             -110      E
```

### Cleanse source data

**As seen above, a number of the data values in the source data are N/A and some are DIV/0! or blank, so let's remove those.**

```
train<-read.csv("C:/Users/staples/Documents/Coursera/PracticalMachineLearning/Data/pml-training.csv",hea
test<-read.csv("C:/Users/staples/Documents/Coursera/PracticalMachineLearning/Data/pml-testing.csv",head
trainNA<-apply(train,2,function(x) {sum(is.na(x))})
testNA<-apply(test,2,function(y) {sum(is.na(y))})
train<-train[,which(trainNA == 0)]
test<-test[,which(testNA == 0)]
```

After cleansing data, 60 variables remain (out of an original count of 160 variables) in both the train and test data.

## Eliminate unnecessary variables

**Since we are only concerned with how well participants did the exercise (data from accelerometers from belt, forearm, arm, and dumbell from each of the participants) and are not concerned with the names of the participants or when they did the exercises, we can remove the first 7 variables (X, user_name, raw_timestamp_part_1, raw_timestamp_part_2, cvtd_timestamp, new_window, num_window) from the train and test data.**

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

**The final variable in the test data set, problem_id, is not needed, so let's also remove that one.**

```
test<-test[,c(1:52)]
```

**And now a quick look at the variables in our source data.**

```
str(train)
```

```
## 'data.frame':    19622 obs. of  53 variables:
##  $ roll_belt           : num  1.41 1.41 1.42 1.48 1.48 1.45 1.42 1.42 1.43 1.45 ...
##  $ pitch_belt          : num  8.07 8.07 8.07 8.05 8.07 8.06 8.09 8.13 8.16 8.17 ...
##  $ yaw_belt            : num  -94.4 -94.4 -94.4 -94.4 -94.4 -94.4 -94.4 -94.4 -94.4 -94.4 ...
##  $ total_accel_belt    : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ gyros_belt_x        : num  0 0.02 0 0.02 0.02 0.02 0.02 0.02 0.02 0.03 ...
##  $ gyros_belt_y        : num  0 0 0 0 0.02 0 0 0 0 0 ...
##  $ gyros_belt_z        : num  -0.02 -0.02 -0.02 -0.03 -0.02 -0.02 -0.02 -0.02 -0.02 0 ...
##  $ accel_belt_x        : int  -21 -22 -20 -22 -21 -21 -22 -22 -20 -21 ...
##  $ accel_belt_y        : int  4 4 5 3 2 4 3 4 2 4 ...
##  $ accel_belt_z        : int  22 22 23 21 24 21 21 21 24 22 ...
##  $ magnet_belt_x       : int  -3 -7 -2 -6 -6 0 -4 -2 1 -3 ...
##  $ magnet_belt_y       : int  599 608 600 604 600 603 599 603 602 609 ...
##  $ magnet_belt_z       : int  -313 -311 -305 -310 -302 -312 -311 -313 -312 -308 ...
##  $ roll_arm            : num  -128 -128 -128 -128 -128 -128 -128 -128 -128 -128 ...
##  $ pitch_arm           : num  22.5 22.5 22.5 22.1 22.1 22 21.9 21.8 21.7 21.6 ...
##  $ yaw_arm             : num  -161 -161 -161 -161 -161 -161 -161 -161 -161 -161 ...
##  $ total_accel_arm     : int  34 34 34 34 34 34 34 34 34 34 ...
##  $ gyros_arm_x         : num  0 0.02 0.02 0.02 0 0.02 0 0.02 0.02 0.02 ...
##  $ gyros_arm_y         : num  0 -0.02 -0.02 -0.03 -0.03 -0.03 -0.03 -0.02 -0.03 -0.03 ...
##  $ gyros_arm_z         : num  -0.02 -0.02 -0.02 0.02 0 0 0 0 -0.02 -0.02 ...
##  $ accel_arm_x         : int  -288 -290 -289 -289 -289 -289 -289 -289 -288 -288 ...
##  $ accel_arm_y         : int  109 110 110 111 111 111 111 111 109 110 ...
##  $ accel_arm_z         : int  -123 -125 -126 -123 -123 -122 -125 -124 -122 -124 ...
##  $ magnet_arm_x        : int  -368 -369 -368 -372 -374 -369 -373 -372 -369 -376 ...
##  $ magnet_arm_y        : int  337 337 344 344 337 342 336 338 341 334 ...
```

```
##  $ magnet_arm_z        : int  516 513 513 512 506 513 509 510 518 516 ...
##  $ roll_dumbbell       : num  13.1 13.1 12.9 13.4 13.4 ...
##  $ pitch_dumbbell      : num  -70.5 -70.6 -70.3 -70.4 -70.4 ...
##  $ yaw_dumbbell        : num  -84.9 -84.7 -85.1 -84.9 -84.9 ...
##  $ total_accel_dumbbell: int  37 37 37 37 37 37 37 37 37 37 ...
##  $ gyros_dumbbell_x    : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ gyros_dumbbell_y    : num  -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.02 ...
##  $ gyros_dumbbell_z    : num  0 0 0 -0.02 0 0 0 0 0 0 ...
##  $ accel_dumbbell_x    : int  -234 -233 -232 -232 -233 -234 -232 -234 -232 -235 ...
##  $ accel_dumbbell_y    : int  47 47 46 48 48 48 47 46 47 48 ...
##  $ accel_dumbbell_z    : int  -271 -269 -270 -269 -270 -269 -270 -272 -269 -270 ...
##  $ magnet_dumbbell_x   : int  -559 -555 -561 -552 -554 -558 -551 -555 -549 -558 ...
##  $ magnet_dumbbell_y   : int  293 296 298 303 292 294 295 300 292 291 ...
##  $ magnet_dumbbell_z   : num  -65 -64 -63 -60 -68 -66 -70 -74 -65 -69 ...
##  $ roll_forearm        : num  28.4 28.3 28.3 28.1 28 27.9 27.9 27.8 27.7 27.7 ...
##  $ pitch_forearm       : num  -63.9 -63.9 -63.9 -63.9 -63.9 -63.9 -63.9 -63.8 -63.8 -63.8 ...
##  $ yaw_forearm         : num  -153 -153 -152 -152 -152 -152 -152 -152 -152 -152 ...
##  $ total_accel_forearm : int  36 36 36 36 36 36 36 36 36 36 ...
##  $ gyros_forearm_x     : num  0.03 0.02 0.03 0.02 0.02 0.02 0.02 0.02 0.03 0.02 ...
##  $ gyros_forearm_y     : num  0 0 -0.02 -0.02 0 -0.02 0 -0.02 0 0 ...
##  $ gyros_forearm_z     : num  -0.02 -0.02 0 0 -0.02 -0.03 -0.02 0 -0.02 -0.02 ...
##  $ accel_forearm_x     : int  192 192 196 189 189 193 195 193 193 190 ...
##  $ accel_forearm_y     : int  203 203 204 206 206 203 205 205 204 205 ...
##  $ accel_forearm_z     : int  -215 -216 -213 -214 -214 -215 -215 -213 -214 -215 ...
##  $ magnet_forearm_x    : int  -17 -18 -18 -16 -17 -9 -18 -9 -16 -22 ...
##  $ magnet_forearm_y    : num  654 661 658 658 655 660 659 660 653 656 ...
##  $ magnet_forearm_z    : num  476 473 469 469 473 478 470 474 476 473 ...
##  $ classe              : Factor w/ 5 levels "A","B","C","D",..: 1 1 1 1 1 1 1 1 1 1 ...
```

We are left with 53 numeric and integer variables in the train data and 52 num and int variables in the test data, with the exception of the 'classe' (factor) variable in the train data, which is the evaluation of how well each participant performed each exercise and is the variable we want to predict.

### Set seed

**Setting a seed value should enable us to get identical results in subsequent runs of the code**

```r
set.seed(12345)
```

### Partition training data into training and testing (cross validation)

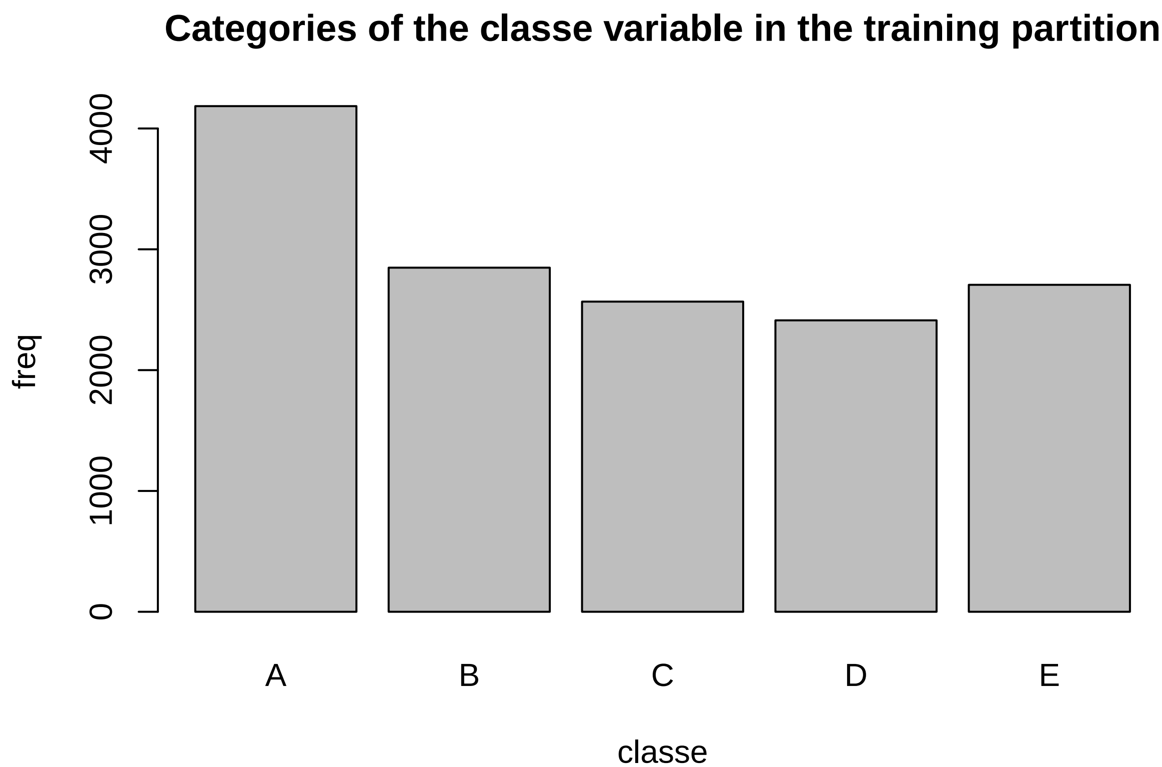
**We will partition our training data into trainPart and testPart, with a 75%/25% split, respectively. The testPart data will be used as validation to help ensure we don't overfit our model to the trainPart data; that is, we will train the models with the trainPart data and then test or validate with the testPart data to help ensure the model generalizes well to data other than the testPart.**

```r
partition<-createDataPartition(y=train$classe,p=0.75,list=FALSE)
trainPart<-train[partition,]
testPart<-train[-partition,]
```

## Exploratory Data Analysis

**Let's evaluate the "classe" results from the training partition.**

```
plot(trainPart$classe,main="Categories of the classe variable in the training partition",xlab="classe",
```




In our training partition, we see that class A has more than 4000 observations, significantly higher than each of the other classes, which range between about 2500 and 3000. This is good, because class A is correct technique, but the combined frequencies of classes B - E (incorrect technique) exceeds the total for class A.

## Model comparisons

**Let's construct a simple CART model to predict exercise results:**

```
cartTrain<-rpart(classe~.,data=trainPart,method="class")
cartPredict<-predict(cartTrain,testPart,type="class")
confusionMatrix(cartPredict,testPart$classe)
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction    A    B    C    D    E
##            A 1260  156   33   40   23
```

```
##          B   52  555   73   52   52
##          C   24  136  575   83   95
##          D   40   33  150  513   89
##          E   19   69   24  116  642
##
## Overall Statistics
##
##                Accuracy : 0.7229
##                  95% CI : (0.7101, 0.7354)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6486
##  Mcnemar's Test P-Value : < 2.2e-16
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9032   0.5848   0.6725   0.6381   0.7125
## Specificity            0.9282   0.9421   0.9165   0.9239   0.9430
## Pos Pred Value         0.8333   0.7079   0.6298   0.6218   0.7379
## Neg Pred Value         0.9602   0.9044   0.9298   0.9287   0.9358
## Prevalence             0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Rate         0.2569   0.1132   0.1173   0.1046   0.1309
## Detection Prevalence   0.3083   0.1599   0.1862   0.1682   0.1774
## Balanced Accuracy      0.9157   0.7635   0.7945   0.7810   0.8278
```

Our CART model has a prediction accurary of 72.3% and a reasonably tight 95% confidence interval. We see that 1260 particiipants are correctly predicted for class A, 555 for class B, and so on. The specificity is above 0.91 for all 5 classes, but the sensitivity varies between 0.58 (class B) and 0.90 (class A).

**Let's now construct a random forest model to predict exercise results:**

```r
rfTrain<-train(classe~.,data=trainPart,method="rf")
rfPredict<-predict(rfTrain,testPart)
confusionMatrix(rfPredict,testPart$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1393    6    0    0    0
##          B    2  938    2    0    0
##          C    0    5  849   10    2
##          D    0    0    4  794    5
##          E    0    0    0    0  894
##
## Overall Statistics
##
##                Accuracy : 0.9927
##                  95% CI : (0.9899, 0.9949)
##     No Information Rate : 0.2845
```

```
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9907
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9986   0.9884   0.9930   0.9876   0.9922
## Specificity            0.9983   0.9990   0.9958   0.9978   1.0000
## Pos Pred Value         0.9957   0.9958   0.9804   0.9888   1.0000
## Neg Pred Value         0.9994   0.9972   0.9985   0.9976   0.9983
## Prevalence             0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Rate         0.2841   0.1913   0.1731   0.1619   0.1823
## Detection Prevalence   0.2853   0.1921   0.1766   0.1637   0.1823
## Balanced Accuracy      0.9984   0.9937   0.9944   0.9927   0.9961
```

Our Random Forest model has a prediction accurary of 99.3% and a very tight 95% confidence interval. We see that 1393 particiipants are correctly predicted for class A, 938 for class B, and so on, and we have very few misclassification errors. The specificity is above .995 for all 5 classes, and the sensitivity varies between .987 and .998. This is a fairly dramatic improvement over the results from our CART model, so we select the Random Forest model for our predictions.

## Accuracy and out of sample error

We see above that the Random Forest algorithm peCformed better than our CART model. Accuracy for the Random Forest model was 0.993 (95% CI: (.9899,.9949)) compared to 0.723 for CART (95% CI: (.7101,.7354)). The accuracy of the Random Forest model suggests the expected out-of-sample error is estimated at 0.007, or 0.7%.

## Predictions of 20 observations based on Random Forest model

### Using the code provided in the prediction submission instructions

This will write out the predicted classe results (e.g., A, B, C…) for 20 observations - each to an individual file - which will be submitted in the subsequent step of the assignment.