Practical Machine Learning - Write-Up Project

Carlos Alberto Guevara Díez

November, 2015

The goal of this project is to use data from accelerometers on the belt, forearm, arm, and dumbell of 6 participants and predict the manner in which people sis the exercise, to be more specific I´m going to predict the classe variable of the data sets, for further information about the description of the project and data used, please refer to the ReadMe file included in this repo.

Data Pre-Processing and Exploratory Analysis

The first thing to do is to load all the libraries needed to run the functions used in the project.

```
library(caret)
```

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

```
library(ggplot2)
library(corrplot)
library(randomForest)
```

```
## randomForest 4.6-12
## Type rfNews() to see new features/changes/bug fixes.
```

```
library (rattle)
```

```
## Loading required package: RGtk2
## Rattle: A free graphical interface for data mining with R.
## Versión 3.5.0 Copyright (c) 2006-2015 Togaware Pty Ltd.
## Escriba 'rattle()' para agitar, sacudir y  rotar sus datos.
```

Once loaded I´m going to load the data into R. Note: If you are going to run the code in your own environment please change the first line of the following chunk:

```
#Adjust this line to your own environment.
setwd("C:/Users/220194/Documents/Data Science Specialization/08 Practical Machine Learning/PracticalMachineLearn
#Read csv files provided
training <- read.csv("pml-training.csv",na.strings = c("NA", ""))
testing <- read.csv("pml-testing.csv",na.strings = c("NA", ""))
```

At this point I need to make some exploratory analysis on the data in order to identify those columns that do not support the analysis because they are non-numerical. Using what I´ve seen now I'm going to remove columns using the grep function an create a new vector. Note: To make the report easier to read I´m hiding the results of the sapply function, please refer to the appendix section (1) of this report to see the results.

```
#Explore the class of the columns
sapply(testing,class)
#Using previous exploration generate newe data sets removing non-numerical columns
training_aux <- training[, -(grep("timestamp|X|user_name|num_window|new_window", names(training)))]
testing_aux <- testing[, -(grep("timestamp|X|user_name|num_window|new_window", names(testing)))]
```

As part of the pre-processing tasks I need to get rid of the NA values, as you know The NA's records make our machine learning algorithm less precise, that´s why is so important to take care of it, once removed I'm updating the vectors I´m going to use in my analysis. Finally I'm going to create my working vectors using a 60%-40% relation between training and testing data sets.

```
#Identify NAs
NAs <- apply(training_aux, 2, function(x) {
    sum(is.na(x))})
#Remove NAs
training_aux <- training_aux[, which(NAs == 0)]
testing_aux <- testing_aux[, which(NAs == 0)]
#Create Working training and testing data sets.
trainingWS <- training_aux[createDataPartition(y = training_aux$classe, p = 0.6, list = FALSE), ]
testingWS <- training_aux[-createDataPartition(y = training_aux$classe, p = 0.6, list = FALSE), ]
```

Now it´s time to look the changes that I've done, once again for more readability of the document I´m hiding the results, please refer to the appendix section (2) of this report to see the results.

```
#Review the changes in the DataSets
dim(trainingWS)
head(trainingWS)
```

To end with the exploratory tasks I´m showing the following graphic, this one is useful to understand the correlation among the variables, please note the code of color in which the dark blue indicates strong correlation and red negative correlation.

```
#Remove NAs and then plot a correlation graphic
Graph <- trainingWS
NAs <- apply(trainingWS, 2, function(x) {
    sum(is.na(x))
})
Graph<- trainingWS[, which(NAs == 0)]
CorrePlot = cor( Graph[,-c(grep("timestamp|X|user_name|num_window|new_window",names(Graph)), length(Graph))])
corrplot(CorrePlot, method="circle",tl.cex=1)
```
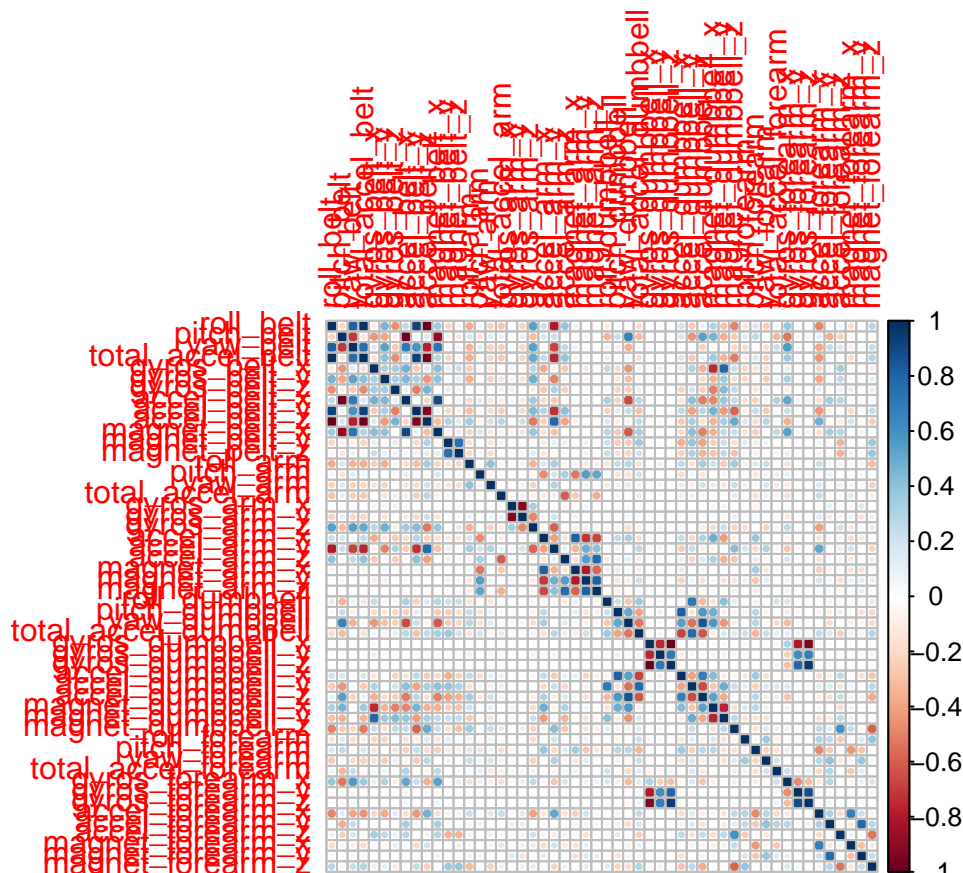

Model creation

Now that I have collected enough information to understand the datasets it´s time to build my model using Random Forest.

2

```
#Set seed for reproduction in other environments
set.seed(10)
#Create Model using trainControl and boot as method
rfModel<- train(trainingWS$classe ~ ., data = trainingWS, method = "rf",
    prof = TRUE, trControl = trainControl(method = "boot", number = 5, allowParallel = TRUE))
#Check for the model, final results and accuracy
rfModel
```

```
## Random Forest
##
## 11776 samples
##    52 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Bootstrapped (5 reps)
## Summary of sample sizes: 11776, 11776, 11776, 11776, 11776
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa       Accuracy SD  Kappa SD
##    2    0.9855974  0.9817599   0.002441620  0.003094780
##   27    0.9867116  0.9831738   0.002450084  0.003108448
##   52    0.9741038  0.9672035   0.005255190  0.006693790
##
## Accuracy was used to select the optimal model using  the largest value.
## The final value used for the model was mtry = 27.
```

```
rfModel.Final<- rfModel$results
round(max(rfModel.Final$Accuracy), 3) * 100
```

```
## [1] 98.7
```
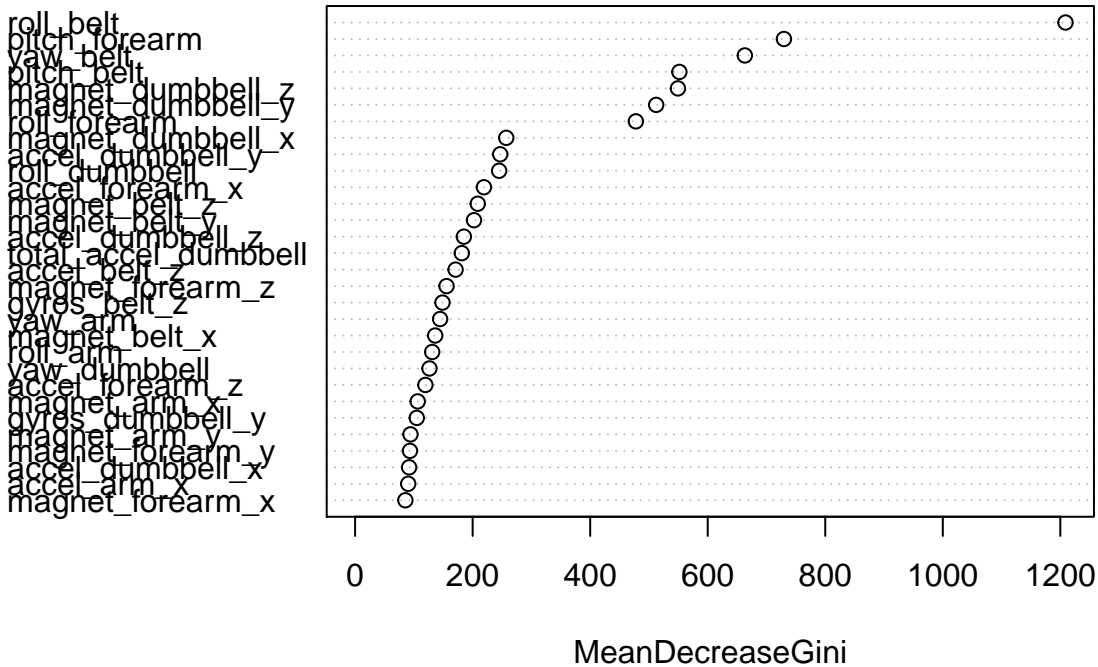
Now that I ran the model it´s possible to say that we have great results, the accuracy of the model is of 98.6 % which is a very nice value. Next I´m plotting a Variable Importance Plot which helps to understand the importance of variables (significance) measured by the Random Forest Model.

```
varImpPlot(rfModel$finalModel,
           main = "Principal Components with high importance")
```

Principal Components with high importance

roll_belt
pitch_forearm
yaw_belt
pitch_belt
magnet_dumbbell_z
magnet_dumbbell_y
roll_forearm
magnet_dumbbell_x
accel_dumbbell_y
roll_dumbbell
accel_forearm_x
magnet_belt_y
magnet_belt_z
accel_dumbbell_z
total_accel_dumbbell
accel_belt_z
magnet_forearm_z
gyros_belt_z
yaw_arm
magnet_belt_x
roll_arm
yaw_dumbbell
accel_forearm_z
magnet_arm_x
gyros_dumbbell_y
magnet_arm_y
magnet_forearm_y
accel_dumbbell_x
accel_arm_x
magnet_forearm_x

MeanDecreaseGini

Testing Evaluations

With our model built now it´s time to execute some validation to test the accuracy of our model, which will be the following:

Cross Validation.

Now let´s use cross validation to test accuracy on the testing data set created in the past section of this document, as we can see in the results I also got a high level of accuracy

```
testingWS$predRight <- (predict(rfModel, testingWS))== testingWS$classe
table(predict(rfModel, testingWS), testingWS$classe)
```

```
##
##        A    B    C    D    E
##   A 2230    5    0    0    0
##   B    2 1509    3    0    0
##   C    0    4 1364    6    1
##   D    0    0    1 1278    2
##   E    0    0    0    2 1439
```

```
CrossValidation<- postResample((predict(rfModel, testingWS)), testingWS$classe)
CrossValidation
```

```
## Accuracy    Kappa
## 0.9966862 0.9958083
```

Confusion Matrix

Now it´s time to test the performance of the model, once again its showed that the results of the predictions are very close with the result of the matrix with accuracy of 99.6 %, all this test results shows that the model was built correctly.

4

```r
#set seed for reproduction in opther environments
set.seed(10)
#build the matrix
CrossValidationError <- confusionMatrix((predict(rfModel, testingWS)), testingWS$classe)
CrossValidationError
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 2230    5    0    0    0
##          B    2 1509    3    0    0
##          C    0    4 1364    6    1
##          D    0    0    1 1278    2
##          E    0    0    0    2 1439
##
## Overall Statistics
##
##                Accuracy : 0.9967
##                  95% CI : (0.9951, 0.9978)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9958
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9991   0.9941   0.9971   0.9938   0.9979
## Specificity            0.9991   0.9992   0.9983   0.9995   0.9997
## Pos Pred Value         0.9978   0.9967   0.9920   0.9977   0.9986
## Neg Pred Value         0.9996   0.9986   0.9994   0.9988   0.9995
## Prevalence             0.2845   0.1935   0.1744   0.1639   0.1838
## Detection Rate         0.2842   0.1923   0.1738   0.1629   0.1834
## Detection Prevalence   0.2849   0.1930   0.1752   0.1633   0.1837
## Balanced Accuracy      0.9991   0.9966   0.9977   0.9967   0.9988
```

```r
#Calculate Accuracy
postResample((predict(rfModel, testingWS)), testingWS$classe)[[1]]
```

```
## [1] 0.9966862
```

```r
1- postResample((predict(rfModel, testingWS)), testingWS$classe)[[1]]
```

```
## [1] 0.00331379
```

Submission Part of The Project, prediction of the 20 cases

Once that I´ve built and tested my model it´s time to use the 20 cases test set to predict the behavior of the people using the devices, first I´m going to prepare the function to write the files that I´m going to submit to Coursera.

```r
pml_write_files = function(x, directory="solutionfiles"){
  dir.create (directory)
  n = length(x)
  for(i in 1:n){
    filename = paste0("problem_id_",i,".txt")
```

```
    filename=file.path(directory, filename)
    write.table(x[i],file=filename,quote=FALSE,row.names=FALSE,col.names=FALSE)
  }
}
```

Now that the function is written all I have to do is to use the predict function in the Random Forest model built using de TESTING data set provided (20 cases) and create the files, in the "solutionfiles"" directory provided in this repo are the 20 files that contains the prediction of each case provided in the TESTING data set.

```
Model.Prediction <- predict(rfModel, testing)
Model.Prediction
```

```
## [1] B A B A A E D B A A B C B A E E A B B B
## Levels: A B C D E
```

```
pml_write_files(Model.Prediction)
```

```
## Warning in dir.create(directory): 'solutionfiles' already exists
```

Final Thoughts

In this assignment I´ve put in practice several concepts reviewed in the entire Data Science Specialization, finally I´ve put into practice them and understood the relation among training, testing and prediction datasets. This project really gave me a complete view of the data science goal and also gave me a lot of ideas to put in practice in the real world, I really hope that you enjoy reading this report as much as I've enjoy writing it, and of course that may help you to understand your own concepts and applications.

Apendix

Section (1) sapply function to explore the data

```
#Explore the class of the columns
sapply(testing,class)
```

```
##                      X             user_name    raw_timestamp_part_1
##              "integer"              "factor"               "integer"
##   raw_timestamp_part_2        cvtd_timestamp              new_window
##              "integer"              "factor"                "factor"
##             num_window             roll_belt              pitch_belt
##              "integer"             "numeric"               "numeric"
##               yaw_belt       total_accel_belt      kurtosis_roll_belt
##              "numeric"             "integer"               "logical"
##     kurtosis_picth_belt      kurtosis_yaw_belt     skewness_roll_belt
##              "logical"             "logical"               "logical"
##   skewness_roll_belt.1      skewness_yaw_belt          max_roll_belt
##              "logical"             "logical"               "logical"
##          max_picth_belt           max_yaw_belt          min_roll_belt
##              "logical"             "logical"               "logical"
##          min_pitch_belt           min_yaw_belt     amplitude_roll_belt
##              "logical"             "logical"               "logical"
##     amplitude_pitch_belt    amplitude_yaw_belt    var_total_accel_belt
##              "logical"             "logical"               "logical"
##           avg_roll_belt        stddev_roll_belt           var_roll_belt
##              "logical"             "logical"               "logical"
```

```
##              avg_pitch_belt            stddev_pitch_belt               var_pitch_belt
##                   "logical"                  "logical"                    "logical"
##                avg_yaw_belt              stddev_yaw_belt                 var_yaw_belt
##                   "logical"                  "logical"                    "logical"
##                gyros_belt_x                 gyros_belt_y                 gyros_belt_z
##                   "numeric"                  "numeric"                    "numeric"
##                accel_belt_x                 accel_belt_y                 accel_belt_z
##                   "integer"                  "integer"                    "integer"
##               magnet_belt_x               magnet_belt_y                magnet_belt_z
##                   "integer"                  "integer"                    "integer"
##                    roll_arm                    pitch_arm                      yaw_arm
##                   "numeric"                  "numeric"                    "numeric"
##              total_accel_arm               var_accel_arm                 avg_roll_arm
##                   "integer"                  "logical"                    "logical"
##              stddev_roll_arm                var_roll_arm                avg_pitch_arm
##                   "logical"                  "logical"                    "logical"
##             stddev_pitch_arm               var_pitch_arm                  avg_yaw_arm
##                   "logical"                  "logical"                    "logical"
##               stddev_yaw_arm                 var_yaw_arm                  gyros_arm_x
##                   "logical"                  "logical"                    "numeric"
##                 gyros_arm_y                  gyros_arm_z                  accel_arm_x
##                   "numeric"                  "numeric"                    "integer"
##                 accel_arm_y                  accel_arm_z                 magnet_arm_x
##                   "integer"                  "integer"                    "integer"
##                magnet_arm_y                 magnet_arm_z             kurtosis_roll_arm
##                   "integer"                  "integer"                    "logical"
##           kurtosis_picth_arm            kurtosis_yaw_arm            skewness_roll_arm
##                   "logical"                  "logical"                    "logical"
##           skewness_pitch_arm            skewness_yaw_arm                  max_roll_arm
##                   "logical"                  "logical"                    "logical"
##                max_picth_arm                 max_yaw_arm                  min_roll_arm
##                   "logical"                  "logical"                    "logical"
##                min_pitch_arm                 min_yaw_arm            amplitude_roll_arm
##                   "logical"                  "logical"                    "logical"
##           amplitude_pitch_arm           amplitude_yaw_arm                roll_dumbbell
##                   "logical"                  "logical"                    "numeric"
##               pitch_dumbbell                yaw_dumbbell         kurtosis_roll_dumbbell
##                   "numeric"                  "numeric"                    "logical"
##      kurtosis_picth_dumbbell      kurtosis_yaw_dumbbell       skewness_roll_dumbbell
##                   "logical"                  "logical"                    "logical"
##       skewness_pitch_dumbbell      skewness_yaw_dumbbell           max_roll_dumbbell
##                   "logical"                  "logical"                    "logical"
##            max_picth_dumbbell           max_yaw_dumbbell            min_roll_dumbbell
##                   "logical"                  "logical"                    "logical"
##            min_pitch_dumbbell           min_yaw_dumbbell      amplitude_roll_dumbbell
##                   "logical"                  "logical"                    "logical"
##      amplitude_pitch_dumbbell     amplitude_yaw_dumbbell         total_accel_dumbbell
##                   "logical"                  "logical"                    "integer"
##           var_accel_dumbbell           avg_roll_dumbbell        stddev_roll_dumbbell
##                   "logical"                  "logical"                    "logical"
##           var_roll_dumbbell           avg_pitch_dumbbell       stddev_pitch_dumbbell
##                   "logical"                  "logical"                    "logical"
##           var_pitch_dumbbell            avg_yaw_dumbbell         stddev_yaw_dumbbell
##                   "logical"                  "logical"                    "logical"
##            var_yaw_dumbbell            gyros_dumbbell_x            gyros_dumbbell_y
##                   "logical"                  "numeric"                    "numeric"
##            gyros_dumbbell_z            accel_dumbbell_x            accel_dumbbell_y
##                   "numeric"                  "integer"                    "integer"
##            accel_dumbbell_z           magnet_dumbbell_x           magnet_dumbbell_y
##                   "integer"                  "integer"                    "integer"
##           magnet_dumbbell_z                 roll_forearm                pitch_forearm
```

```
##            "integer"                 "numeric"                 "numeric"
##          yaw_forearm    kurtosis_roll_forearm   kurtosis_picth_forearm
##            "numeric"                 "logical"                 "logical"
##   kurtosis_yaw_forearm   skewness_roll_forearm   skewness_pitch_forearm
##            "logical"                 "logical"                 "logical"
##   skewness_yaw_forearm        max_roll_forearm        max_picth_forearm
##            "logical"                 "logical"                 "logical"
##         max_yaw_forearm        min_roll_forearm        min_pitch_forearm
##            "logical"                 "logical"                 "logical"
##         min_yaw_forearm   amplitude_roll_forearm  amplitude_pitch_forearm
##            "logical"                 "logical"                 "logical"
##   amplitude_yaw_forearm      total_accel_forearm        var_accel_forearm
##            "logical"                 "integer"                 "logical"
##         avg_roll_forearm     stddev_roll_forearm        var_roll_forearm
##            "logical"                 "logical"                 "logical"
##        avg_pitch_forearm    stddev_pitch_forearm       var_pitch_forearm
##            "logical"                 "logical"                 "logical"
##          avg_yaw_forearm      stddev_yaw_forearm         var_yaw_forearm
##            "logical"                 "logical"                 "logical"
##          gyros_forearm_x          gyros_forearm_y          gyros_forearm_z
##            "numeric"                 "numeric"                 "numeric"
##          accel_forearm_x          accel_forearm_y          accel_forearm_z
##            "integer"                 "integer"                 "integer"
##         magnet_forearm_x         magnet_forearm_y         magnet_forearm_z
##            "integer"                 "integer"                 "integer"
##               problem_id
##            "integer"
```

```r
#Using previous exploration generate newe data sets removing non-numerical columns
training_aux <- training[, -(grep("timestamp|X|user_name|num_window|new_window", names(training)))]
testing_aux <- testing[, -(grep("timestamp|X|user_name|num_window|new_window", names(testing)))]
```

Section (2) Review working data sets without NAs

```r
#Review the changes in the DataSets
dim(trainingWS)
```

```
## [1] 11776    53
```

```r
head(trainingWS)
```

```
##    roll_belt pitch_belt yaw_belt total_accel_belt gyros_belt_x
## 1       1.41       8.07    -94.4                3         0.00
## 3       1.42       8.07    -94.4                3         0.00
## 4       1.48       8.05    -94.4                3         0.02
## 5       1.48       8.07    -94.4                3         0.02
## 9       1.43       8.16    -94.4                3         0.02
## 10      1.45       8.17    -94.4                3         0.03
##    gyros_belt_y gyros_belt_z accel_belt_x accel_belt_y accel_belt_z
## 1          0.00        -0.02          -21            4           22
## 3          0.00        -0.02          -20            5           23
## 4          0.00        -0.03          -22            3           21
## 5          0.02        -0.02          -21            2           24
## 9          0.00        -0.02          -20            2           24
## 10         0.00         0.00          -21            4           22
##    magnet_belt_x magnet_belt_y magnet_belt_z roll_arm pitch_arm yaw_arm
## 1             -3           599          -313     -128      22.5    -161
## 3             -2           600          -305     -128      22.5    -161
```

8

```
## 4                 -6          604            -310         -128     22.1      -161
## 5                 -6          600            -302         -128     22.1      -161
## 9                  1          602            -312         -128     21.7      -161
## 10                -3          609            -308         -128     21.6      -161
##    total_accel_arm gyros_arm_x gyros_arm_y gyros_arm_z accel_arm_x
## 1               34        0.00        0.00       -0.02        -288
## 3               34        0.02       -0.02       -0.02        -289
## 4               34        0.02       -0.03        0.02        -289
## 5               34        0.00       -0.03        0.00        -289
## 9               34        0.02       -0.03       -0.02        -288
## 10              34        0.02       -0.03       -0.02        -288
##    accel_arm_y accel_arm_z magnet_arm_x magnet_arm_y magnet_arm_z
## 1          109        -123         -368          337          516
## 3          110        -126         -368          344          513
## 4          111        -123         -372          344          512
## 5          111        -123         -374          337          506
## 9          109        -122         -369          341          518
## 10         110        -124         -376          334          516
##    roll_dumbbell pitch_dumbbell yaw_dumbbell total_accel_dumbbell
## 1       13.05217      -70.49400    -84.87394                   37
## 3       12.85075      -70.27812    -85.14078                   37
## 4       13.43120      -70.39379    -84.87363                   37
## 5       13.37872      -70.42856    -84.85306                   37
## 9       13.15463      -70.42520    -84.91563                   37
## 10      13.33034      -70.85059    -84.44602                   37
##    gyros_dumbbell_x gyros_dumbbell_y gyros_dumbbell_z accel_dumbbell_x
## 1                 0            -0.02             0.00             -234
## 3                 0            -0.02             0.00             -232
## 4                 0            -0.02            -0.02             -232
## 5                 0            -0.02             0.00             -233
## 9                 0            -0.02             0.00             -232
## 10                0            -0.02             0.00             -235
##    accel_dumbbell_y accel_dumbbell_z magnet_dumbbell_x magnet_dumbbell_y
## 1                47             -271              -559               293
## 3                46             -270              -561               298
## 4                48             -269              -552               303
## 5                48             -270              -554               292
## 9                47             -269              -549               292
## 10               48             -270              -558               291
##    magnet_dumbbell_z roll_forearm pitch_forearm yaw_forearm
## 1                -65         28.4         -63.9        -153
## 3                -63         28.3         -63.9        -152
## 4                -60         28.1         -63.9        -152
## 5                -68         28.0         -63.9        -152
## 9                -65         27.7         -63.8        -152
## 10               -69         27.7         -63.8        -152
##    total_accel_forearm gyros_forearm_x gyros_forearm_y gyros_forearm_z
## 1                   36            0.03            0.00           -0.02
## 3                   36            0.03           -0.02            0.00
## 4                   36            0.02           -0.02            0.00
## 5                   36            0.02            0.00           -0.02
## 9                   36            0.03            0.00           -0.02
## 10                  36            0.02            0.00           -0.02
##    accel_forearm_x accel_forearm_y accel_forearm_z magnet_forearm_x
## 1              192             203            -215              -17
## 3              196             204            -213              -18
## 4              189             206            -214              -16
## 5              189             206            -214              -17
## 9              193             204            -214              -16
## 10             190             205            -215              -22
##    magnet_forearm_y magnet_forearm_z classe
```

```
## 1             654              476      A
## 3             658              469      A
## 4             658              469      A
## 5             655              473      A
## 9             653              476      A
## 10            656              473      A
```