

Course Project

Daniel

11 6 2020

Introduction

This analysis serves as the final course project of the Johns Hopkins' Coursera course "Practical Machine Learning" in the Data Science Specialization.

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 dumbbell 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://web.archive.org/web/20161224072740/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://web.archive.org/web/20161224072740/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.

Goal of the analysis

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.

```
library(caret)
```

Loading relevant libraries and data

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

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

```
library(rpart)
library(rpart.plot)
library(RColorBrewer)
library(rattle)
```

```
## Loading required package: tibble
```

```
## Loading required package: bitops
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.4.0 Copyright (c) 2006-2020 Togaware Pty Ltd.
## Geben Sie 'rattle()' ein, um Ihre Daten mischen.
```

```
library(randomForest)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:rattle':
##
##      importance
```

```
## The following object is masked from 'package:ggplot2':
##
##      margin
```

```
trainUrl <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
testUrl <- "http://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"

training <- read.csv(url(trainUrl), na.strings=c("NA", "#DIV/0!", ""))
testing <- read.csv(url(testUrl), na.strings=c("NA", "#DIV/0!", ""))
set.seed(987)
```

Exploratory Data Analysis

```
## 'data.frame':   19622 obs. of  160 variables:
## $ X               : int  1 2 3 4 5 6 7 8 9 10 ...
## $ user_name        : Factor w/ 6 levels "adelmo","carlitos",...: 2 2 2 2 2 2 2 2 2 2 ...
## $ raw_timestamp_part_1 : int  1323084231 1323084231 1323084231 1323084232 1323084232 1323084232 ...
## $ raw_timestamp_part_2 : int  788290 808298 820366 120339 196328 304277 368296 440390 484323 484 ...
## $ cvtd_timestamp      : Factor w/ 20 levels "02/12/2011 13:32",...: 9 9 9 9 9 9 9 9 9 9 ...
## $ new_window          : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
## $ num_window          : int  11 11 11 12 12 12 12 12 12 12 ...
## $ 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 ...
## $ kurtosis_roll_belt  : num  NA NA NA NA NA NA NA NA NA NA ...
## $ kurtosis_pitch_belt : num  NA NA NA NA NA NA NA NA NA NA ...
## $ kurtosis_yaw_belt   : logi  NA NA NA NA NA NA ...
## $ skewness_roll_belt  : num  NA NA NA NA NA NA NA NA NA NA ...
## $ skewness_roll_belt.1 : num  NA NA NA NA NA NA NA NA NA NA ...
## $ skewness_yaw_belt   : logi  NA NA NA NA NA NA ...
## $ max_roll_belt       : num  NA NA NA NA NA NA NA NA NA NA ...
## $ max_pitch_belt      : int  NA NA NA NA NA NA NA NA NA NA ...
## $ max_yaw_belt        : num  NA NA NA NA NA NA NA NA NA NA ...
## $ min_roll_belt       : num  NA NA NA NA NA NA NA NA NA NA ...
## $ min_pitch_belt      : int  NA NA NA NA NA NA NA NA NA NA ...
## $ min_yaw_belt        : num  NA NA NA NA NA NA NA NA NA NA ...
## $ amplitude_roll_belt : num  NA NA NA NA NA NA NA NA NA NA ...
## $ amplitude_pitch_belt : int  NA NA NA NA NA NA NA NA NA NA ...
## $ amplitude_yaw_belt   : num  NA NA NA NA NA NA NA NA NA NA ...
## $ var_total_accel_belt : num  NA NA NA NA NA NA NA NA NA NA ...
## $ avg_roll_belt       : num  NA NA NA NA NA NA NA NA NA NA ...
## $ stddev_roll_belt    : num  NA NA NA NA NA NA NA NA NA NA ...
## $ var_roll_belt       : num  NA NA NA NA NA NA NA NA NA NA ...
## $ avg_pitch_belt      : num  NA NA NA NA NA NA NA NA NA NA ...
## $ stddev_pitch_belt   : num  NA NA NA NA NA NA NA NA NA NA ...
## $ var_pitch_belt      : num  NA NA NA NA NA NA NA NA NA NA ...
## $ avg_yaw_belt        : num  NA NA NA NA NA NA NA NA NA NA ...
## $ stddev_yaw_belt     : num  NA NA NA NA NA NA NA NA NA NA ...
## $ var_yaw_belt        : num  NA NA NA NA NA NA NA NA NA NA ...
## $ 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 ...
## $ var_accel_arm       : num  NA NA NA NA NA NA NA NA NA NA ...
## $ avg_roll_arm        : num  NA NA NA NA NA NA NA NA NA NA ...
```

```

## $ stddev_roll_arm      : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ var_roll_arm         : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ avg_pitch_arm        : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ stddev_pitch_arm     : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ var_pitch_arm        : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ avg_yaw_arm          : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ stddev_yaw_arm       : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ var_yaw_arm          : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ 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 ...
## $ kurtosis_roll_arm    : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ kurtosis_pitch_arm   : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ kurtosis_yaw_arm     : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ skewness_roll_arm    : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ skewness_pitch_arm   : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ skewness_yaw_arm     : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ max_roll_arm         : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ max_pitch_arm        : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ max_yaw_arm          : int   NA NA NA NA NA NA NA NA NA NA NA ...
## $ min_roll_arm         : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ min_pitch_arm        : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ min_yaw_arm          : int   NA NA NA NA NA NA NA NA NA NA NA ...
## $ amplitude_roll_arm   : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ amplitude_pitch_arm  : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ amplitude_yaw_arm     : int   NA NA NA NA NA NA NA NA NA NA NA ...
## $ 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 ...
## $ kurtosis_roll_dumbbell : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ kurtosis_pitch_dumbbell : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ kurtosis_yaw_dumbbell : logi  NA NA NA NA NA NA ...
## $ skewness_roll_dumbbell : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ skewness_pitch_dumbbell : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ skewness_yaw_dumbbell : logi  NA NA NA NA NA NA ...
## $ max_roll_dumbbell    : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ max_pitch_dumbbell   : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ max_yaw_dumbbell     : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ min_roll_dumbbell    : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ min_pitch_dumbbell   : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ min_yaw_dumbbell     : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ amplitude_roll_dumbbell : num  NA NA NA NA NA NA NA NA NA NA NA ...
## [list output truncated]

## 'data.frame': 20 obs. of 160 variables:
## $ X : int 1 2 3 4 5 6 7 8 9 10 ...
## $ user_name : Factor w/ 6 levels "adelmo","carlitos",...: 6 5 5 1 4 5 5 2 3 ...
## $ raw_timestamp_part_1 : int 1323095002 1322673067 1322673075 1322832789 1322489635 1322673149 ...

```

```

## $ raw_timestamp_part_2 : int 868349 778725 342967 560311 814776 510661 766645 54671 916313 3842
## $ cvtd_timestamp      : Factor w/ 11 levels "02/12/2011 13:33",...: 5 10 10 1 6 11 11 10 3 2 ...
## $ new_window          : Factor w/ 1 level "no": 1 1 1 1 1 1 1 1 1 ...
## $ num_window          : int 74 431 439 194 235 504 485 440 323 664 ...
## $ roll_belt           : num 123 1.02 0.87 125 1.35 -5.92 1.2 0.43 0.93 114 ...
## $ pitch_belt          : num 27 4.87 1.82 -41.6 3.33 1.59 4.44 4.15 6.72 22.4 ...
## $ yaw_belt            : num -4.75 -88.9 -88.5 162 -88.6 -87.7 -87.3 -88.5 -93.7 -13.1 ...
## $ total_accel_belt    : int 20 4 5 17 3 4 4 4 4 18 ...
## $ kurtosis_roll_belt  : logi NA NA NA NA NA NA ...
## $ kurtosis_pitch_belt : logi NA NA NA NA NA NA ...
## $ kurtosis_yaw_belt   : logi NA NA NA NA NA NA ...
## $ skewness_roll_belt  : logi NA NA NA NA NA NA ...
## $ skewness_roll_belt.1 : logi NA NA NA NA NA NA ...
## $ skewness_yaw_belt   : logi NA NA NA NA NA NA ...
## $ max_roll_belt       : logi NA NA NA NA NA NA ...
## $ max_pitch_belt      : logi NA NA NA NA NA NA ...
## $ max_yaw_belt        : logi NA NA NA NA NA NA ...
## $ min_roll_belt       : logi NA NA NA NA NA NA ...
## $ min_pitch_belt      : logi NA NA NA NA NA NA ...
## $ min_yaw_belt        : logi NA NA NA NA NA NA ...
## $ amplitude_roll_belt : logi NA NA NA NA NA NA ...
## $ amplitude_pitch_belt : logi NA NA NA NA NA NA ...
## $ amplitude_yaw_belt  : logi NA NA NA NA NA NA ...
## $ var_total_accel_belt : logi NA NA NA NA NA NA ...
## $ avg_roll_belt       : logi NA NA NA NA NA NA ...
## $ stddev_roll_belt    : logi NA NA NA NA NA NA ...
## $ var_roll_belt       : logi NA NA NA NA NA NA ...
## $ avg_pitch_belt      : logi NA NA NA NA NA NA ...
## $ stddev_pitch_belt   : logi NA NA NA NA NA NA ...
## $ var_pitch_belt      : logi NA NA NA NA NA NA ...
## $ avg_yaw_belt        : logi NA NA NA NA NA NA ...
## $ stddev_yaw_belt     : logi NA NA NA NA NA NA ...
## $ var_yaw_belt        : logi NA NA NA NA NA NA ...
## $ gyros_belt_x        : num -0.5 -0.06 0.05 0.11 0.03 0.1 -0.06 -0.18 0.1 0.14 ...
## $ gyros_belt_y        : num -0.02 -0.02 0.02 0.11 0.02 0.05 0 -0.02 0 0.11 ...
## $ gyros_belt_z        : num -0.46 -0.07 0.03 -0.16 0 -0.13 0 -0.03 -0.02 -0.16 ...
## $ accel_belt_x        : int -38 -13 1 46 -8 -11 -14 -10 -15 -25 ...
## $ accel_belt_y        : int 69 11 -1 45 4 -16 2 -2 1 63 ...
## $ accel_belt_z        : int -179 39 49 -156 27 38 35 42 32 -158 ...
## $ magnet_belt_x       : int -13 43 29 169 33 31 50 39 -6 10 ...
## $ magnet_belt_y       : int 581 636 631 608 566 638 622 635 600 601 ...
## $ magnet_belt_z       : int -382 -309 -312 -304 -418 -291 -315 -305 -302 -330 ...
## $ roll_arm            : num 40.7 0 0 -109 76.1 0 0 0 -137 -82.4 ...
## $ pitch_arm           : num -27.8 0 0 55 2.76 0 0 0 11.2 -63.8 ...
## $ yaw_arm             : num 178 0 0 -142 102 0 0 0 -167 -75.3 ...
## $ total_accel_arm     : int 10 38 44 25 29 14 15 22 34 32 ...
## $ var_accel_arm       : logi NA NA NA NA NA NA ...
## $ avg_roll_arm        : logi NA NA NA NA NA NA ...
## $ stddev_roll_arm     : logi NA NA NA NA NA NA ...
## $ var_roll_arm        : logi NA NA NA NA NA NA ...
## $ avg_pitch_arm       : logi NA NA NA NA NA NA ...
## $ stddev_pitch_arm    : logi NA NA NA NA NA NA ...
## $ var_pitch_arm       : logi NA NA NA NA NA NA ...
## $ avg_yaw_arm         : logi NA NA NA NA NA NA ...

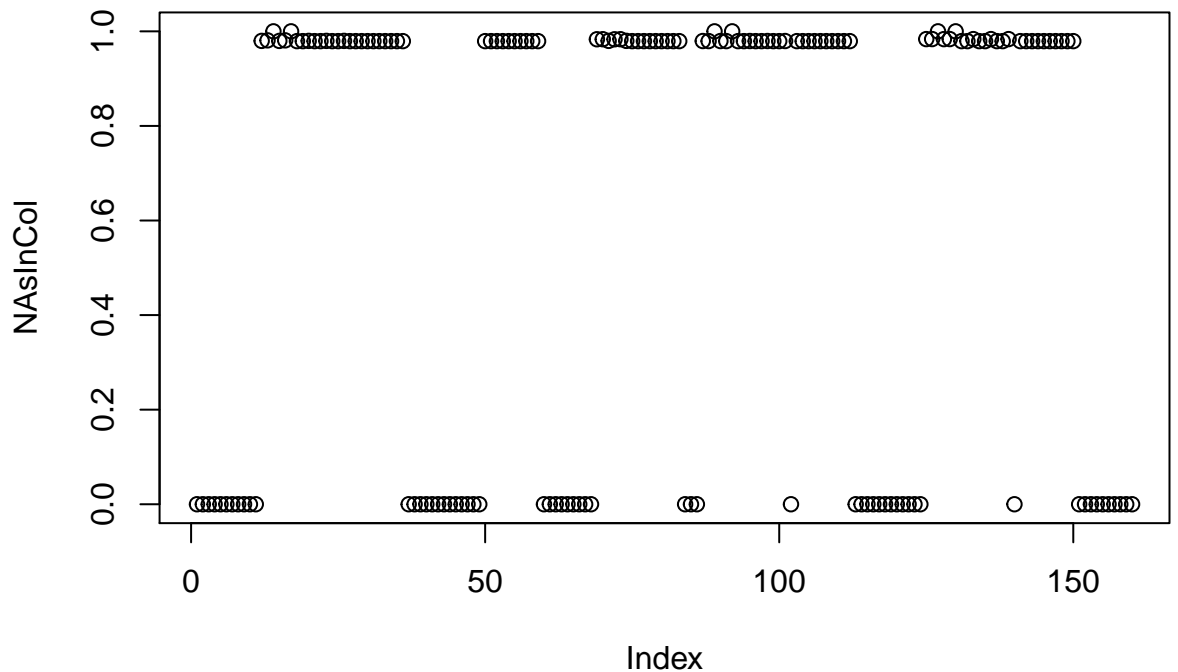
```

```
## $ stddev_yaw_arm      : logi  NA NA NA NA NA NA ...
## $ var_yaw_arm         : logi  NA NA NA NA NA NA ...
## $ gyros_arm_x         : num   -1.65 -1.17 2.1 0.22 -1.96 0.02 2.36 -3.71 0.03 0.26 ...
## $ gyros_arm_y         : num    0.48 0.85 -1.36 -0.51 0.79 0.05 -1.01 1.85 -0.02 -0.5 ...
## $ gyros_arm_z         : num   -0.18 -0.43 1.13 0.92 -0.54 -0.07 0.89 -0.69 -0.02 0.79 ...
## $ accel_arm_x         : int    16 -290 -341 -238 -197 -26 99 -98 -287 -301 ...
## $ accel_arm_y         : int    38 215 245 -57 200 130 79 175 111 -42 ...
## $ accel_arm_z         : int    93 -90 -87 6 -30 -19 -67 -78 -122 -80 ...
## $ magnet_arm_x        : int   -326 -325 -264 -173 -170 396 702 535 -367 -420 ...
## $ magnet_arm_y        : int   385 447 474 257 275 176 15 215 335 294 ...
## $ magnet_arm_z        : int   481 434 413 633 617 516 217 385 520 493 ...
## $ kurtosis_roll_arm   : logi   NA NA NA NA NA NA ...
## $ kurtosis_pitch_arm  : logi   NA NA NA NA NA NA ...
## $ kurtosis_yaw_arm    : logi   NA NA NA NA NA NA ...
## $ skewness_roll_arm   : logi   NA NA NA NA NA NA ...
## $ skewness_pitch_arm  : logi   NA NA NA NA NA NA ...
## $ skewness_yaw_arm    : logi   NA NA NA NA NA NA ...
## $ max_roll_arm        : logi   NA NA NA NA NA NA ...
## $ max_pitch_arm       : logi   NA NA NA NA NA NA ...
## $ max_yaw_arm         : logi   NA NA NA NA NA NA ...
## $ min_roll_arm        : logi   NA NA NA NA NA NA ...
## $ min_pitch_arm       : logi   NA NA NA NA NA NA ...
## $ min_yaw_arm         : logi   NA NA NA NA NA NA ...
## $ amplitude_roll_arm  : logi   NA NA NA NA NA NA ...
## $ amplitude_pitch_arm : logi   NA NA NA NA NA NA ...
## $ amplitude_yaw_arm   : logi   NA NA NA NA NA NA ...
## $ roll_dumbbell       : num   -17.7 54.5 57.1 43.1 -101.4 ...
## $ pitch_dumbbell      : num    25 -53.7 -51.4 -30 -53.4 ...
## $ yaw_dumbbell        : num   126.2 -75.5 -75.2 -103.3 -14.2 ...
## $ kurtosis_roll_dumbbell : logi   NA NA NA NA NA NA ...
## $ kurtosis_pitch_dumbbell : logi   NA NA NA NA NA NA ...
## $ kurtosis_yaw_dumbbell : logi   NA NA NA NA NA NA ...
## $ skewness_roll_dumbbell : logi   NA NA NA NA NA NA ...
## $ skewness_pitch_dumbbell : logi   NA NA NA NA NA NA ...
## $ skewness_yaw_dumbbell : logi   NA NA NA NA NA NA ...
## $ max_roll_dumbbell   : logi   NA NA NA NA NA NA ...
## $ max_pitch_dumbbell  : logi   NA NA NA NA NA NA ...
## $ max_yaw_dumbbell    : logi   NA NA NA NA NA NA ...
## $ min_roll_dumbbell   : logi   NA NA NA NA NA NA ...
## $ min_pitch_dumbbell  : logi   NA NA NA NA NA NA ...
## $ min_yaw_dumbbell    : logi   NA NA NA NA NA NA ...
## $ amplitude_roll_dumbbell : logi   NA NA NA NA NA NA ...
## [list output truncated]
```

```
## [1] 1925102
```

```
## [1] 0.6131835
```

```
# Calculate proportion of missing values in each column
NASInCol <- apply(training, 2, function(x) sum(is.na(x)/length(x)))
plot(NASInCol)
```



Preprocessing

```
# Remove variables with too many missing values
training2 <- training[,NAsInCol < 0.3]
sum(is.na(training2))
```

```
## [1] 0
```

```
testing2 <- testing[, NAsInCol < 0.3]
sum(is.na(testing2))
```

```
## [1] 0
```

```
# Remove ID and time stamps column
train.fin <- training2[, -c(1,3,4)]
test.fin <- testing2[, -c(1,3,4)]
# Split training set into two
inTrain <- createDataPartition(y=train.fin$classe, p=0.8, list = FALSE)
myTrain <- train.fin[inTrain,]
myTest <- train.fin[-inTrain,]
```

```
# In order to make the Random Forest Model work, we need to coerce the testing data into the same type
for (i in 1:length(test.fin) ) {
```

```

    for(j in 1:length(train.fin)) {
      if( length( grep(names(train.fin[i]), names(test.fin)[j]) ) ==1) {
        class(test.fin[j]) <- class(train.fin[i])
      }
    }
  }
names(test.fin) <- names(train.fin)
test.fin <- rbind(train.fin[1, ], test.fin)

```

Coerce testing data into the same type

```

## Warning in `[<-.factor`(`*tmp*`, ri, value = 1:20): invalid factor level, NA
## generated

```

```

test.fin <- test.fin[-1,]

```

```

rffit <- randomForest(classe ~., data = myTrain)
# Prediction:
predRF <- predict(rffit, myTest, type = "class")
confusionMatrix(predRF, myTest$classe)

```

Fit Random Forest Model

```

## Confusion Matrix and Statistics
##
##           Reference
## Prediction   A    B    C    D    E
##           A 1115    1    0    0    0
##           B   1  756    1    0    0
##           C   0   2  683    2    0
##           D   0   0   0  640    4
##           E   0   0   0   1  717
##
## Overall Statistics
##
##               Accuracy : 0.9969
##               95% CI : (0.9947, 0.9984)
##       No Information Rate : 0.2845
##       P-Value [Acc > NIR] : < 2.2e-16
##
##               Kappa : 0.9961
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##               Class: A Class: B Class: C Class: D Class: E
## Sensitivity           0.9991  0.9960  0.9985  0.9953  0.9945
## Specificity           0.9996  0.9994  0.9988  0.9988  0.9997
## Pos Pred Value        0.9991  0.9974  0.9942  0.9938  0.9986

```


## Neg Pred Value	0.9996	0.9991	0.9997	0.9991	0.9988
## Prevalence	0.2845	0.1935	0.1744	0.1639	0.1838
## Detection Rate	0.2842	0.1927	0.1741	0.1631	0.1828
## Detection Prevalence	0.2845	0.1932	0.1751	0.1642	0.1830
## Balanced Accuracy	0.9994	0.9977	0.9987	0.9971	0.9971

```
predictRfTest <- predict(rffit, test.fin, type = "class")
```

```
predictRfTest
```

Predict on Testing Data

```
##  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21
##  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
```