I downloaded training and testing datasets

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

#make datasets readable as csv files
training <- read.csv(url(url_train), na.strings=c("NA","#DIV/0!",""))
testing <- read.csv(url(url_test), na.strings=c("NA","#DIV/0!",""))
#get rid of NAs
training<-training[,colSums(is.na(training)) == 0]
testing <-testing[,colSums(is.na(testing)) == 0]
```

I created a training (60% of the original data) and testing dataset (40%)

```
inTrain = createDataPartition(y=training$classe, p = 0.6, list=FALSE)
myTraining = training[inTrain, ]
mytesting = training[-inTrain, ]
mytraining<-myTraining

#check structure of the datasets for exploration purposes
dim(mytraining); dim(mytesting); dim(testing)
```

```
## [1] 11776    60
```

```
## [1] 7846   60
```

```
## [1] 20 60
```

```
str(mytraining)
```

```
## 'data.frame':    11776 obs. of  60 variables:
##  $ X                   : int  1 3 5 7 8 10 11 13 14 16 ...
##  $ 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 1323084232 1323084232 1323084232 1323084232 1323084232 1323084232 1323084232 1323084232
##  $ raw_timestamp_part_2: int  788290 820366 196328 368296 440390 484434 500302 560359 576390 644302 ...
##  $ 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 12 12 12 12 12 12 12 12 ...
##  $ roll_belt           : num  1.41 1.42 1.48 1.42 1.42 1.45 1.45 1.42 1.42 1.48 ...
##  $ pitch_belt          : num  8.07 8.07 8.07 8.09 8.13 8.17 8.18 8.2 8.21 8.15 ...
##  $ 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 0.02 0.02 0.02 0.03 0.03 0.02 0.02 0 ...
##  $ gyros_belt_y        : num  0 0 0.02 0 0 0 0 0 0 0 ...
##  $ gyros_belt_z        : num  -0.02 -0.02 -0.02 -0.02 -0.02 0 -0.02 0 -0.02 0 ...
##  $ accel_belt_x        : int  -21 -20 -21 -22 -22 -21 -21 -22 -22 -21 ...
##  $ accel_belt_y        : int  4 5 2 3 4 4 2 4 4 4 ...
##  $ accel_belt_z        : int  22 23 24 21 21 22 23 21 21 23 ...
##  $ magnet_belt_x       : int  -3 -2 -6 -4 -2 -3 -5 -3 -8 0 ...
##  $ magnet_belt_y       : int  599 600 600 599 603 609 596 606 598 592 ...
##  $ magnet_belt_z       : int  -313 -305 -302 -311 -313 -308 -317 -309 -310 -305 ...
##  $ roll_arm            : num  -128 -128 -128 -128 -128 -128 -128 -128 -128 -129 ...
##  $ pitch_arm           : num  22.5 22.5 22.1 21.9 21.8 21.6 21.5 21.4 21.4 21.3 ...
##  $ 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 0 0.02 0.02 0.02 0.02 0.02 0.02 ...
##  $ gyros_arm_y         : num  0 -0.02 -0.03 -0.03 -0.02 -0.03 -0.03 -0.02 0 0 ...
##  $ gyros_arm_z         : num  -0.02 -0.02 0 0 0 -0.02 0 -0.02 -0.03 -0.03 ...
##  $ accel_arm_x         : int  -288 -289 -289 -289 -289 -288 -290 -287 -288 -289 ...
##  $ accel_arm_y         : int  109 110 111 111 111 110 110 111 111 109 ...
##  $ accel_arm_z         : int  -123 -126 -123 -125 -124 -124 -123 -124 -124 -121 ...
##  $ magnet_arm_x        : int  -368 -368 -374 -373 -372 -376 -366 -372 -371 -367 ...
##  $ magnet_arm_y        : int  337 344 337 336 338 334 339 338 331 340 ...
##  $ magnet_arm_z        : int  516 513 506 509 510 516 509 509 523 509 ...
##  $ roll_dumbbell       : num  13.1 12.9 13.4 13.1 12.8 ...
##  $ pitch_dumbbell      : num  -70.5 -70.3 -70.4 -70.2 -70.3 ...
##  $ yaw_dumbbell        : num  -84.9 -85.1 -84.9 -85.1 -85.1 ...
##  $ 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 0.02 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 0 0 0 -0.02 -0.02 0 ...
##  $ accel_dumbbell_x    : int  -234 -232 -233 -232 -234 -235 -233 -234 -234 -233 ...
##  $ accel_dumbbell_y    : int  47 46 48 47 46 48 47 48 48 48 ...
##  $ accel_dumbbell_z    : int  -271 -270 -270 -270 -272 -270 -269 -269 -268 -271 ...
##  $ magnet_dumbbell_x   : int  -559 -561 -554 -551 -555 -558 -564 -552 -554 -554 ...
##  $ magnet_dumbbell_y   : int  293 298 292 295 300 291 299 302 295 297 ...
##  $ magnet_dumbbell_z   : num  -65 -63 -68 -70 -74 -69 -64 -69 -68 -73 ...
##  $ roll_forearm        : num  28.4 28.3 28 27.9 27.8 27.7 27.6 27.2 27.2 27.1 ...
##  $ pitch_forearm       : num  -63.9 -63.9 -63.9 -63.9 -63.8 -63.8 -63.8 -63.9 -63.9 -64 ...
##  $ yaw_forearm         : num  -153 -152 -152 -152 -152 -152 -152 -151 -151 -151 ...
##  $ total_accel_forearm : int  36 36 36 36 36 36 36 36 36 36 ...
##  $ gyros_forearm_x     : num  0.03 0.03 0.02 0.02 0.02 0.02 0.02 0 0 0.02 ...
##  $ gyros_forearm_y     : num  0 -0.02 0 0 -0.02 0 -0.02 0 -0.02 0 ...
##  $ gyros_forearm_z     : num  -0.02 0 -0.02 -0.02 0 -0.02 -0.02 -0.03 -0.03 0 ...
##  $ accel_forearm_x     : int  192 196 189 195 193 190 193 193 193 194 ...
##  $ accel_forearm_y     : int  203 204 206 205 205 205 205 205 202 204 ...
##  $ accel_forearm_z     : int  -215 -213 -214 -215 -213 -215 -214 -215 -214 -215 ...
##  $ magnet_forearm_x    : int  -17 -18 -17 -18 -9 -22 -17 -15 -14 -13 ...
##  $ magnet_forearm_y    : num  654 658 655 659 660 656 657 655 659 656 ...
##  $ magnet_forearm_z    : num  476 469 473 470 474 473 465 472 478 471 ...
##  $ classe              : Factor w/ 5 levels "A","B","C","D",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
colnames(mytraining)
```

```
##  [1] "X"                    "user_name"            "raw_timestamp_part_1"
##  [4] "raw_timestamp_part_2" "cvtd_timestamp"       "new_window"
##  [7] "num_window"           "roll_belt"            "pitch_belt"
## [10] "yaw_belt"             "total_accel_belt"     "gyros_belt_x"
## [13] "gyros_belt_y"         "gyros_belt_z"         "accel_belt_x"
## [16] "accel_belt_y"         "accel_belt_z"         "magnet_belt_x"
## [19] "magnet_belt_y"        "magnet_belt_z"        "roll_arm"
## [22] "pitch_arm"            "yaw_arm"              "total_accel_arm"
## [25] "gyros_arm_x"          "gyros_arm_y"          "gyros_arm_z"
## [28] "accel_arm_x"          "accel_arm_y"          "accel_arm_z"
## [31] "magnet_arm_x"         "magnet_arm_y"         "magnet_arm_z"
## [34] "roll_dumbbell"        "pitch_dumbbell"       "yaw_dumbbell"
## [37] "total_accel_dumbbell" "gyros_dumbbell_x"     "gyros_dumbbell_y"
## [40] "gyros_dumbbell_z"     "accel_dumbbell_x"     "accel_dumbbell_y"
## [43] "accel_dumbbell_z"     "magnet_dumbbell_x"    "magnet_dumbbell_y"
## [46] "magnet_dumbbell_z"    "roll_forearm"         "pitch_forearm"
## [49] "yaw_forearm"          "total_accel_forearm"  "gyros_forearm_x"
## [52] "gyros_forearm_y"      "gyros_forearm_z"      "accel_forearm_x"
## [55] "accel_forearm_y"      "accel_forearm_z"      "magnet_forearm_x"
## [58] "magnet_forearm_y"     "magnet_forearm_z"     "classe"
```

```
colnames(testing)
```

```
##  [1] "X"                    "user_name"            "raw_timestamp_part_1"
##  [4] "raw_timestamp_part_2" "cvtd_timestamp"       "new_window"
##  [7] "num_window"           "roll_belt"            "pitch_belt"
## [10] "yaw_belt"             "total_accel_belt"     "gyros_belt_x"
## [13] "gyros_belt_y"         "gyros_belt_z"         "accel_belt_x"
## [16] "accel_belt_y"         "accel_belt_z"         "magnet_belt_x"
## [19] "magnet_belt_y"        "magnet_belt_z"        "roll_arm"
## [22] "pitch_arm"            "yaw_arm"              "total_accel_arm"
## [25] "gyros_arm_x"          "gyros_arm_y"          "gyros_arm_z"
## [28] "accel_arm_x"          "accel_arm_y"          "accel_arm_z"
## [31] "magnet_arm_x"         "magnet_arm_y"         "magnet_arm_z"
## [34] "roll_dumbbell"        "pitch_dumbbell"       "yaw_dumbbell"
## [37] "total_accel_dumbbell" "gyros_dumbbell_x"     "gyros_dumbbell_y"
## [40] "gyros_dumbbell_z"     "accel_dumbbell_x"     "accel_dumbbell_y"
## [43] "accel_dumbbell_z"     "magnet_dumbbell_x"    "magnet_dumbbell_y"
## [46] "magnet_dumbbell_z"    "roll_forearm"         "pitch_forearm"
## [49] "yaw_forearm"          "total_accel_forearm"  "gyros_forearm_x"
## [52] "gyros_forearm_y"      "gyros_forearm_z"      "accel_forearm_x"
## [55] "accel_forearm_y"      "accel_forearm_z"      "magnet_forearm_x"
## [58] "magnet_forearm_y"     "magnet_forearm_z"     "problem_id"
```

I createed a list of variables with 0 observations or not useful predictors (NZV)

```
myDataNZV <- nearZeroVar(mytraining, saveMetrics=TRUE)
```

Then I took the list of variables with NZV and concatenate them as a list for readability purposes

```
myNZVvars <- names(mytraining) %in% c("new_window", "kurtosis_roll_belt", "kurtosis_picth_belt",
                                      "kurtosis_yaw_belt", "skewness_roll_belt", "skewness_roll_belt.1", "skewness_yaw_belt",
                                      "max_yaw_belt", "min_yaw_belt", "amplitude_yaw_belt", "avg_roll_arm", "stddev_roll_arm",
                                      "var_roll_arm", "avg_pitch_arm", "stddev_pitch_arm", "var_pitch_arm", "avg_yaw_arm",
                                      "stddev_yaw_arm", "var_yaw_arm", "kurtosis_roll_arm", "kurtosis_picth_arm",
                                      "kurtosis_yaw_arm", "skewness_roll_arm", "skewness_pitch_arm", "skewness_yaw_arm",
                                      "max_roll_arm", "min_roll_arm", "min_pitch_arm", "amplitude_roll_arm", "amplitude_pitch_arm",
                                      "kurtosis_roll_dumbbell", "kurtosis_picth_dumbbell", "kurtosis_yaw_dumbbell", "skewness_roll_dumbbell",
                                      "skewness_pitch_dumbbell", "skewness_yaw_dumbbell", "max_yaw_dumbbell", "min_yaw_dumbbell",
                                      "amplitude_yaw_dumbbell", "kurtosis_roll_forearm", "kurtosis_picth_forearm", "kurtosis_yaw_forearm",
                                      "skewness_roll_forearm", "skewness_pitch_forearm", "skewness_yaw_forearm", "max_roll_forearm",
                                      "max_yaw_forearm", "min_roll_forearm", "min_yaw_forearm", "amplitude_roll_forearm",
                                      "amplitude_yaw_forearm", "avg_roll_forearm", "stddev_roll_forearm", "var_roll_forearm",
                                      "avg_pitch_forearm", "stddev_pitch_forearm", "var_pitch_forearm", "avg_yaw_forearm",
                                      "stddev_yaw_forearm", "var_yaw_forearm")


mytraining <- mytraining[!myNZVvars]#updates the training dataset without NZV variables
```

I checked that the new N of observations is the same as in the old training dataset

```
dim(mytraining)
```

```
## [1] 11776    59
```

I removed ID from dataset so that it does not interfere with machine learning analyses

```
mytraining <- mytraining[c(-1)]
dim(mytraining)
```

```
## [1] 11776    58
```

I got rid of NA; I decided to remove those with more than 60% NA

```
trainingV3 <- mytraining #creating another clean data subset (training V3) with less than 60% NA
for(i in 1:length(mytraining)) { #for every column in the training dataset, if NAs>60%  get rid of it
if( sum( is.na( mytraining[, i] ) ) /nrow(mytraining) >= .6 ) {
for(j in 1:length(trainingV3)) {
if( length( grep(names(mytraining[i]), names(trainingV3)[j]) ) ==1)  { # compare training and training V3
        #and if a column is found to have too many NAs remove it and save it all in training V3
trainingV3 <- trainingV3[ , -j]
}
}
}
}

dim(trainingV3)
```

```
## [1] 11776    58
```

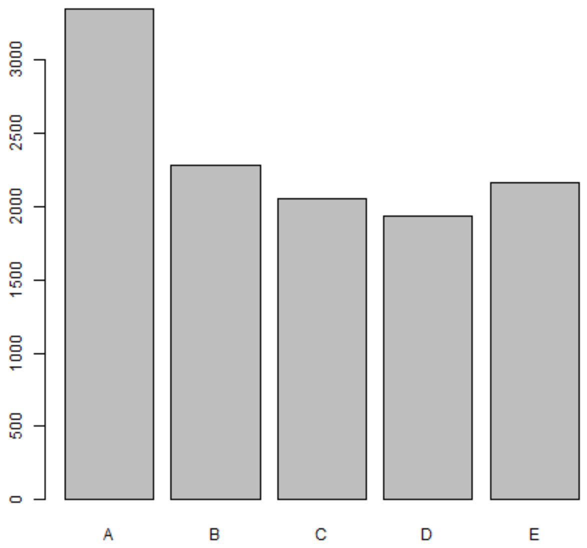
I checked the new N of observations in training V3

```
dim(mytraining)
```

```
## [1] 11776    58
```

I visualized the data

```
plot(mytraining$classe)
```



```
eval()
```

```
## Error in eval(): argument "expr" is missing, with no default
```

```
summary(mytraining$classe)#plot data
```

```
##    A    B    C    D    E 
## 3348 2279 2054 1930 2165
```

```
myTraining <- trainingV3
```

```
rm(trainingV3)
```

I created a table with the list of variables that are included in the training dataset, tested that my testing had all the variables included in my training

```
clean1 <- colnames(mytraining)
mytesting <- mytesting[clean1]
clean2<-colnames(mytraining[,-58])#make the mytesting and initial
#testing dataset compatible with same variables
testing<-testing[clean2]
```

I checked the new N of observations, which should be 57

```
dim(mytesting)
```

```
## [1] 7846   58
```

```
dim(testing)
```

```
## [1] 20 57
```

```
#coerce data into the same data type
for (i in 1:length(testing) ) {
  for(j in 1:length(mytraining)) {
    if( length( grep(names(mytraining[i]), names(testing)[j]) ) ==1)  {
      class(mytesting[j]) <- class(mytraining[i])
    }
  }
}

testing<-rbind(mytraining[2, -58], testing)
testing<-testing[-1,]
```

I tested hardness by using a 3-fold cross-validation to estimate accuracy. This is set in subsequent code using the “fitControl” object

```
fitControl <- trainControl(method='cv', number = 3)
metric <-"fitControl"
```

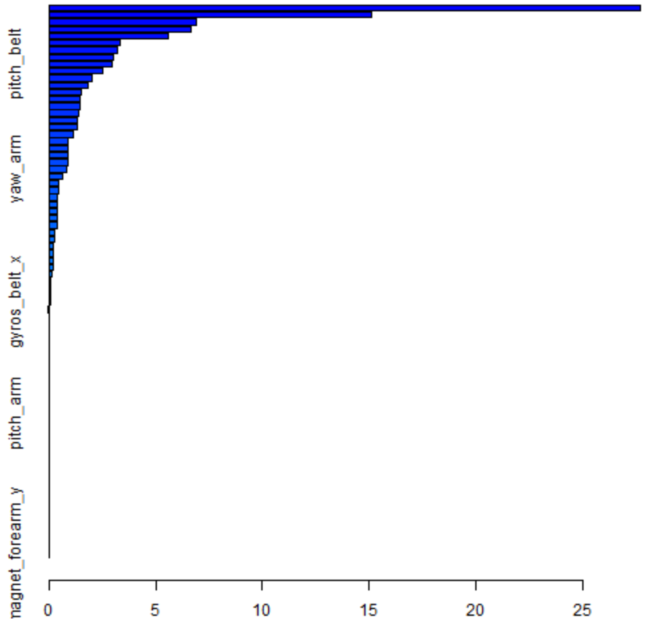
I applied a first ML algorithm for prediction: generalized boosted regression

```
#(gbm)
set.seed(7)
```

```
modFitA1 <- train(classe ~ ., data=mytraining, method="gbm", trControl=fitControl)
```

```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.6094             nan     0.1000    0.1290
##      2        1.5231             nan     0.1000    0.0903
##      3        1.4628             nan     0.1000    0.0664
##      4        1.4188             nan     0.1000    0.0524
##      5        1.3835             nan     0.1000    0.0489
##      6        1.3506             nan     0.1000    0.0499
##      7        1.3198             nan     0.1000    0.0439
##      8        1.2924             nan     0.1000    0.0392
##      9        1.2645             nan     0.1000    0.0345
##     10        1.2424             nan     0.1000    0.0394
##     20        1.0575             nan     0.1000    0.0223
##     40        0.8364             nan     0.1000    0.0107
##     60        0.6986             nan     0.1000    0.0074
##     80        0.5978             nan     0.1000    0.0052
##    100        0.5192             nan     0.1000    0.0033
##    120        0.4543             nan     0.1000    0.0020
##    140        0.4033             nan     0.1000    0.0023
##    150        0.3800             nan     0.1000    0.0035
## 
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.6094             nan     0.1000    0.1947
##      2        1.4827             nan     0.1000    0.1361
##      3        1.3926             nan     0.1000    0.1090
##      4        1.3216             nan     0.1000    0.1083
##      5        1.2524             nan     0.1000    0.0809
##      6        1.2007             nan     0.1000    0.0945
##      7        1.1392             nan     0.1000    0.0703
##      8        1.0944             nan     0.1000    0.0557
##      9        1.0573             nan     0.1000    0.0657
##     10        1.0164             nan     0.1000    0.0476
##     20        0.7593             nan     0.1000    0.0327
##     40        0.4628             nan     0.1000    0.0140
##     60        0.3164             nan     0.1000    0.0080
##     80        0.2219             nan     0.1000    0.0060
##    100        0.1589             nan     0.1000    0.0045
##    120        0.1159             nan     0.1000    0.0035
##    140        0.0866             nan     0.1000    0.0016
##    150        0.0767             nan     0.1000    0.0010
## 
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.6094             nan     0.1000    0.2429
##      2        1.4490             nan     0.1000    0.1849
##      3        1.3322             nan     0.1000    0.1514
##      4        1.2348             nan     0.1000    0.1219
##      5        1.1550             nan     0.1000    0.0972
##      6        1.0926             nan     0.1000    0.0986
##      7        1.0304             nan     0.1000    0.0850
##      8        0.9774             nan     0.1000    0.0800
##      9        0.9276             nan     0.1000    0.0784
##     10        0.8794             nan     0.1000    0.0717
##     20        0.5795             nan     0.1000    0.0347
##     40        0.2997             nan     0.1000    0.0194
##     60        0.1664             nan     0.1000    0.0090
##     80        0.1008             nan     0.1000    0.0031
##    100        0.0671             nan     0.1000    0.0020
##    120        0.0476             nan     0.1000    0.0006
##    140        0.0347             nan     0.1000    0.0005
##    150        0.0296             nan     0.1000    0.0003
## 
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.6094             nan     0.1000    0.1265
##      2        1.5242             nan     0.1000    0.0831
##      3        1.4674             nan     0.1000    0.0679
##      4        1.4222             nan     0.1000    0.0543
##      5        1.3863             nan     0.1000    0.0524
##      6        1.3501             nan     0.1000    0.0480
##      7        1.3189             nan     0.1000    0.0446
##      8        1.2906             nan     0.1000    0.0360
##      9        1.2669             nan     0.1000    0.0371
##     10        1.2412             nan     0.1000    0.0390
##     20        1.0597             nan     0.1000    0.0205
##     40        0.8386             nan     0.1000    0.0138
##     60        0.7004             nan     0.1000    0.0080
##     80        0.6039             nan     0.1000    0.0066
##    100        0.5242             nan     0.1000    0.0032
##    120        0.4618             nan     0.1000    0.0032
##    140        0.4105             nan     0.1000    0.0036
##    150        0.3869             nan     0.1000    0.0027
## 
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.6094             nan     0.1000    0.1891
##      2        1.4848             nan     0.1000    0.1404
##      3        1.3920             nan     0.1000    0.1057
##      4        1.3227             nan     0.1000    0.0945
##      5        1.2617             nan     0.1000    0.0943
##      6        1.2020             nan     0.1000    0.0813
##      7        1.1514             nan     0.1000    0.0699
##      8        1.1049             nan     0.1000    0.0663
##      9        1.0627             nan     0.1000    0.0645
##     10        1.0229             nan     0.1000    0.0536
##     20        0.7600             nan     0.1000    0.0400
##     40        0.4767             nan     0.1000    0.0132
##     60        0.3178             nan     0.1000    0.0098
##     80        0.2227             nan     0.1000    0.0044
##    100        0.1595             nan     0.1000    0.0023
##    120        0.1196             nan     0.1000    0.0028
##    140        0.0911             nan     0.1000    0.0011
##    150        0.0796             nan     0.1000    0.0009
## 
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.6094             nan     0.1000    0.2447
##      2        1.4512             nan     0.1000    0.1772
##      3        1.3362             nan     0.1000    0.1380
##      4        1.2468             nan     0.1000    0.1289
##      5        1.1644             nan     0.1000    0.1034
##      6        1.0991             nan     0.1000    0.0878
##      7        1.0432             nan     0.1000    0.0957
##      8        0.9817             nan     0.1000    0.0821
##      9        0.9307             nan     0.1000    0.0810
##     10        0.8803             nan     0.1000    0.0651
##     20        0.5839             nan     0.1000    0.0345
##     40        0.3046             nan     0.1000    0.0131
##     60        0.1760             nan     0.1000    0.0049
##     80        0.1114             nan     0.1000    0.0021
##    100        0.0732             nan     0.1000    0.0022
##    120        0.0519             nan     0.1000    0.0010
##    140        0.0383             nan     0.1000    0.0006
##    150        0.0336             nan     0.1000    0.0003
## 
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.6094             nan     0.1000    0.1248
##      2        1.5261             nan     0.1000    0.0861
##      3        1.4691             nan     0.1000    0.0662
##      4        1.4253             nan     0.1000    0.0554
##      5        1.3901             nan     0.1000    0.0468
##      6        1.3590             nan     0.1000    0.0474
##      7        1.3279             nan     0.1000    0.0368
##      8        1.3034             nan     0.1000    0.0441
##      9        1.2741             nan     0.1000    0.0407
##     10        1.2472             nan     0.1000    0.0326
##     20        1.0663             nan     0.1000    0.0200
##     40        0.8527             nan     0.1000    0.0143
##     60        0.7112             nan     0.1000    0.0082
##     80        0.6073             nan     0.1000    0.0062
##    100        0.5272             nan     0.1000    0.0053
##    120        0.4640             nan     0.1000    0.0015
##    140        0.4140             nan     0.1000    0.0031
##    150        0.3907             nan     0.1000    0.0019
## 
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.6094             nan     0.1000    0.1833
##      2        1.4883             nan     0.1000    0.1359
##      3        1.3990             nan     0.1000    0.1074
##      4        1.3276             nan     0.1000    0.1059
##      5        1.2601             nan     0.1000    0.0752
##      6        1.2099             nan     0.1000    0.0894
##      7        1.1534             nan     0.1000    0.0737
##      8        1.1082             nan     0.1000    0.0668
##      9        1.0662             nan     0.1000    0.0579
##     10        1.0301             nan     0.1000    0.0531
##     20        0.7780             nan     0.1000    0.0340
##     40        0.4803             nan     0.1000    0.0187
##     60        0.3226             nan     0.1000    0.0079
##     80        0.2245             nan     0.1000    0.0058
##    100        0.1629             nan     0.1000    0.0040
##    120        0.1186             nan     0.1000    0.0019
##    140        0.0893             nan     0.1000    0.0016
##    150        0.0786             nan     0.1000    0.0011
## 
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.6094             nan     0.1000    0.2512
##      2        1.4482             nan     0.1000    0.1773
##      3        1.3340             nan     0.1000    0.1454
##      4        1.2423             nan     0.1000    0.1151
##      5        1.1683             nan     0.1000    0.0985
##      6        1.1041             nan     0.1000    0.0858
##      7        1.0496             nan     0.1000    0.0937
##      8        0.9888             nan     0.1000    0.0756
##      9        0.9398             nan     0.1000    0.0621
##     10        0.8992             nan     0.1000    0.0687
##     20        0.5843             nan     0.1000    0.0413
##     40        0.3028             nan     0.1000    0.0200
##     60        0.1702             nan     0.1000    0.0080
##     80        0.1062             nan     0.1000    0.0029
##    100        0.0712             nan     0.1000    0.0018
##    120        0.0497             nan     0.1000    0.0010
##    140        0.0361             nan     0.1000    0.0009
##    150        0.0311             nan     0.1000    0.0002
## 
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1        1.6094             nan     0.1000    0.2421
##      2        1.4520             nan     0.1000    0.1913
##      3        1.3325             nan     0.1000    0.1443
##      4        1.2403             nan     0.1000    0.1199
##      5        1.1631             nan     0.1000    0.1121
##      6        1.0924             nan     0.1000    0.1097
##      7        1.0257             nan     0.1000    0.0845
##      8        0.9732             nan     0.1000    0.0775
##      9        0.9247             nan     0.1000    0.0787
##     10        0.8771             nan     0.1000    0.0579
##     20        0.5765             nan     0.1000    0.0362
##     40        0.2935             nan     0.1000    0.0136
##     60        0.1683             nan     0.1000    0.0062
##     80        0.1045             nan     0.1000    0.0038
##    100        0.0699             nan     0.1000    0.0015
##    120        0.0497             nan     0.1000    0.0013
##    140        0.0368             nan     0.1000    0.0004
##    150        0.0319             nan     0.1000    0.0009
```

```
par(mar = rep(2, 4))
summary(modFitA1)
```



```
##                                                            var     rel.inf
## raw_timestamp_part_1                      raw_timestamp_part_1 27.69599113
## roll_belt                                            roll_belt 15.10849171
## pitch_forearm                                    pitch_forearm  6.89824165
## num_window                                          num_window  6.67076094
## magnet_dumbbell_z                            magnet_dumbbell_z  5.56839977
## roll_forearm                                      roll_forearm  3.34786853
## cvtd_timestamp30/11/2011 17:12 cvtd_timestamp30/11/2011 17:12  3.21859929
## cvtd_timestamp28/11/2011 14:15 cvtd_timestamp28/11/2011 14:15  3.05444880
## pitch_belt                                          pitch_belt  2.94325798
## magnet_dumbbell_y                            magnet_dumbbell_y  2.49878434
## cvtd_timestamp02/12/2011 13:33 cvtd_timestamp02/12/2011 13:33  2.01219317
## roll_dumbbell                                    roll_dumbbell  1.80778203
## cvtd_timestamp02/12/2011 13:34 cvtd_timestamp02/12/2011 13:34  1.49895339
## accel_forearm_x                                accel_forearm_x  1.45215565
## gyros_belt_z                                      gyros_belt_z  1.44757356
## yaw_belt                                              yaw_belt  1.37166795
## gyros_dumbbell_y                              gyros_dumbbell_y  1.33398345
## cvtd_timestamp02/12/2011 14:57 cvtd_timestamp02/12/2011 14:57  1.30422402
## cvtd_timestamp02/12/2011 14:58 cvtd_timestamp02/12/2011 14:58  1.15395287
## cvtd_timestamp05/12/2011 11:24 cvtd_timestamp05/12/2011 11:24  0.89754902
## cvtd_timestamp30/11/2011 17:11 cvtd_timestamp30/11/2011 17:11  0.89696630
## accel_dumbbell_y                              accel_dumbbell_y  0.88994795
## magnet_belt_z                                    magnet_belt_z  0.87629352
## yaw_arm                                                yaw_arm  0.83794950
## cvtd_timestamp05/12/2011 14:23 cvtd_timestamp05/12/2011 14:23  0.67331550
## cvtd_timestamp05/12/2011 14:22 cvtd_timestamp05/12/2011 14:22  0.45523558
## cvtd_timestamp05/12/2011 14:24 cvtd_timestamp05/12/2011 14:24  0.43555367
## magnet_belt_x                                    magnet_belt_x  0.37873388
## accel_dumbbell_x                              accel_dumbbell_x  0.37363124
## accel_arm_x                                        accel_arm_x  0.36913603
## cvtd_timestamp02/12/2011 13:35 cvtd_timestamp02/12/2011 13:35  0.36875164
## accel_dumbbell_z                              accel_dumbbell_z  0.36437301
## gyros_belt_y                                      gyros_belt_y  0.28816728
## magnet_belt_y                                    magnet_belt_y  0.25904752
## magnet_forearm_x                              magnet_forearm_x  0.20473281
## roll_arm                                              roll_arm  0.19923693
## magnet_dumbbell_x                            magnet_dumbbell_x  0.19455893
## cvtd_timestamp05/12/2011 11:25 cvtd_timestamp05/12/2011 11:25  0.18372765
## cvtd_timestamp02/12/2011 14:59 cvtd_timestamp02/12/2011 14:59  0.13753741
## magnet_arm_z                                      magnet_arm_z  0.10584046
## yaw_dumbbell                                      yaw_dumbbell  0.08206215
## magnet_forearm_z                              magnet_forearm_z  0.05354295
## gyros_belt_x                                      gyros_belt_x  0.05155651
## cvtd_timestamp28/11/2011 14:14 cvtd_timestamp28/11/2011 14:14  0.03522232
## user_namecarlitos                            user_namecarlitos  0.00000000
## user_namecharles                              user_namecharles  0.00000000
## user_nameeurico                                user_nameeurico  0.00000000
## user_namejeremy                                user_namejeremy  0.00000000
## user_namepedro                                  user_namepedro  0.00000000
## raw_timestamp_part_2                      raw_timestamp_part_2  0.00000000
## cvtd_timestamp02/12/2011 14:56 cvtd_timestamp02/12/2011 14:56  0.00000000
## cvtd_timestamp05/12/2011 11:23 cvtd_timestamp05/12/2011 11:23  0.00000000
## cvtd_timestamp28/11/2011 14:13 cvtd_timestamp28/11/2011 14:13  0.00000000
## cvtd_timestamp30/11/2011 17:10 cvtd_timestamp30/11/2011 17:10  0.00000000
## total_accel_belt                             total_accel_belt  0.00000000
## accel_belt_x                                     accel_belt_x  0.00000000
## accel_belt_y                                     accel_belt_y  0.00000000
## accel_belt_z                                     accel_belt_z  0.00000000
## pitch_arm                                           pitch_arm  0.00000000
## total_accel_arm                               total_accel_arm  0.00000000
## gyros_arm_x                                       gyros_arm_x  0.00000000
## gyros_arm_y                                       gyros_arm_y  0.00000000
## gyros_arm_z                                       gyros_arm_z  0.00000000
## accel_arm_y                                       accel_arm_y  0.00000000
## accel_arm_z                                       accel_arm_z  0.00000000
## magnet_arm_x                                     magnet_arm_x  0.00000000
## magnet_arm_y                                     magnet_arm_y  0.00000000
## pitch_dumbbell                                 pitch_dumbbell  0.00000000
## total_accel_dumbbell                     total_accel_dumbbell  0.00000000
## gyros_dumbbell_x                             gyros_dumbbell_x  0.00000000
## gyros_dumbbell_z                             gyros_dumbbell_z  0.00000000
## yaw_forearm                                       yaw_forearm  0.00000000
## total_accel_forearm                       total_accel_forearm  0.00000000
## gyros_forearm_x                               gyros_forearm_x  0.00000000
## gyros_forearm_y                               gyros_forearm_y  0.00000000
## gyros_forearm_z                               gyros_forearm_z  0.00000000
## accel_forearm_y                               accel_forearm_y  0.00000000
## accel_forearm_z                               accel_forearm_z  0.00000000
## magnet_forearm_y                             magnet_forearm_y  0.00000000
```

```
install.packages("e1071")
```

```
## Error in install.packages : Updating loaded packages
```

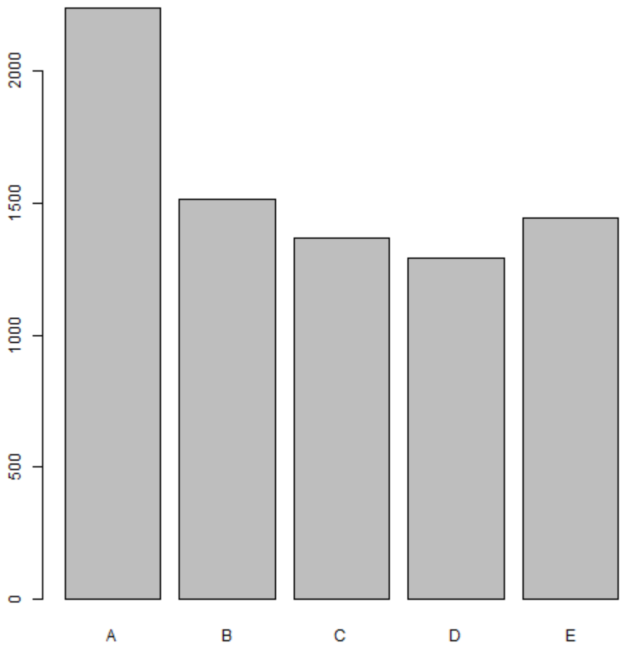
```
library(e1071)
```

I used predictions on testing dataset using the training dataset

```
predsA1 <- predict(modFitA1, mytesting)
#compare predictions and real data using the confusion matrix
CMA1<-confusionMatrix(predictionsA1, mytesting$classe)
```

```
## Error in confusionMatrix(predictionsA1, mytesting$classe): object 'predictionsA1' not found
```

```
par(mar = rep(2, 4))
plot(predsA1)
```

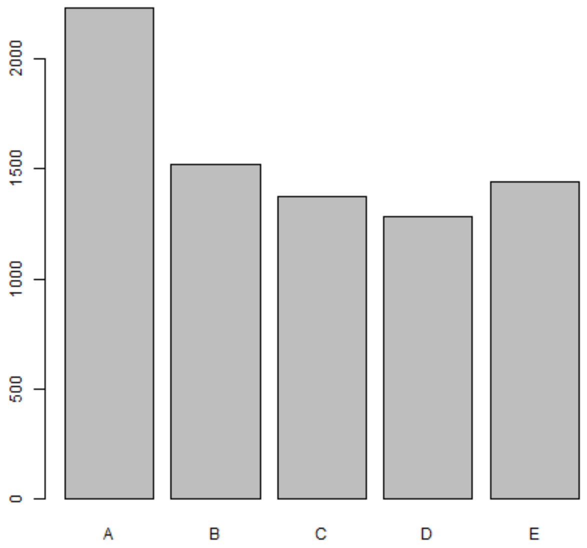


I used a second ML algorithm for prediction: random forest

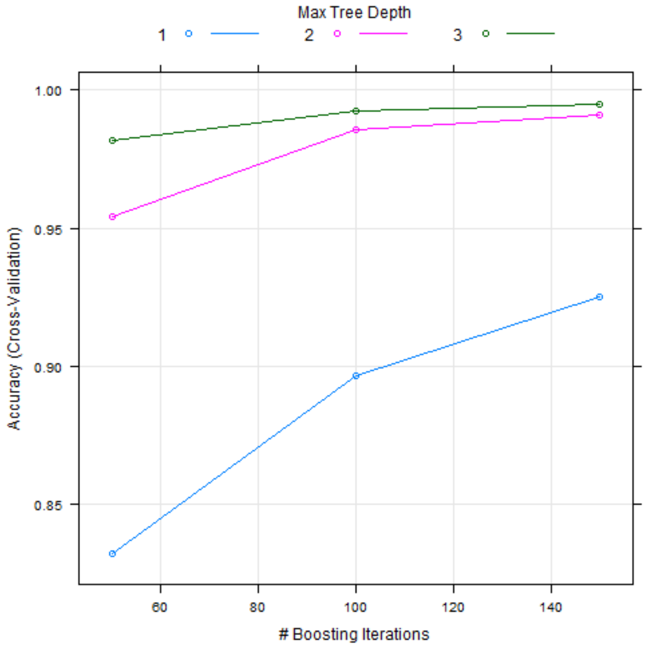
```
set.seed(7)
modFitB1 <- randomForest(classe ~. , data=mytraining, trControl=fitControl)
#use predictions on training set on testing dataset
predsB1 <- predict(modFitB1, mytesting, type = "class")
#calculate accuracy of model use confusion matrix
CMB1<-confusionMatrix(predictionsB1, mytesting$classe)
```

```
## Error in confusionMatrix(predictionsB1, mytesting$classe): object 'predictionsB1' not found
```

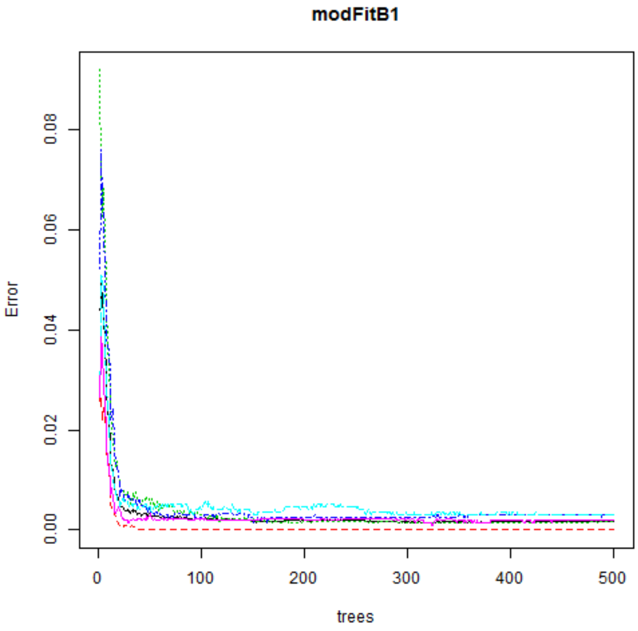
```
plot(predsB1)
```



```
plot(modFitA1)
```



```
plot(modFitB1)
```



```
AccuracyResults<-data.frame(Model=c('GBR', 'RF'), Accuracy=rbind(CMA1$overall[1], CMB1$overall[1]))
```

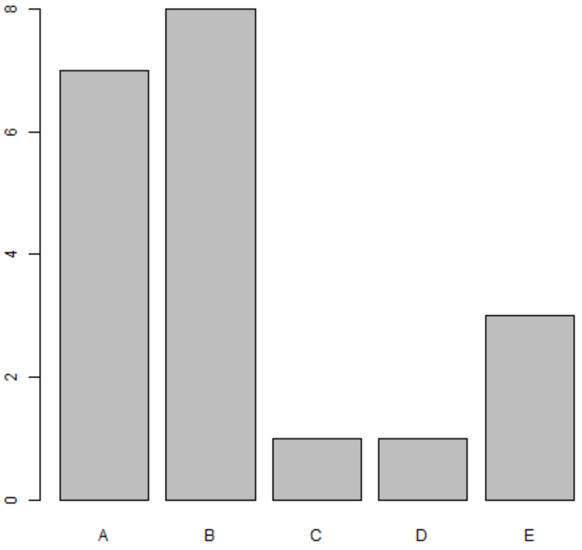
```
## Error in rbind(CMA1$overall[1], CMB1$overall[1]): object 'CMA1' not found
```

```
print(AccuracyResults)
```

```
## Error in print(AccuracyResults): object 'AccuracyResults' not found
```

The out-of-sample error is equal to 1 - accuracy against cross-validation dataset. RF is the best model with 99.89% accuracy e.g. OSE=.11% now use RF prediction in the “initial” testing dataset that we downloaded at the beginning -independent sample)

```
predictB2<-predict(modFitB1, testing, type="class")
plot(predictB2)
```



```
pml_write_files = function(x){
  n = length(x)
  for(i in 1:n){
    filename = paste0("problem_id_",i,".txt")
    write.table(x[i],file=filename,quote=FALSE,row.names=FALSE,col.names=FALSE)
  }
}

pml_write_files(predictB2)

rmarkdown::render("FinalMLassignment.Rmd")
```

```
##
##
```

```
## processing file: FinalMLassignment.Rmd
##
  |
  |                                                                  |   0%
  |
  |..                                                                |   3%
##   ordinary text without R code
##
##
  |
  |....                                                              |   5%
## label: unnamed-chunk-32 (with options)
## List of 1
##  $ include: logi FALSE
##
##
  |
  |.....                                                             |   8%
##   ordinary text without R code
##
##
  |
  |.......                                                           |  11%
## label: unnamed-chunk-33
##
  |
  |.........                                                         |  14%
##   ordinary text without R code
##
##
  |
  |...........                                                       |  16%
## label: unnamed-chunk-34
##
  |
  |............                                                      |  19%
##   ordinary text without R code
##
##
  |
  |..............                                                    |  22%
## label: unnamed-chunk-35
##
  |
  |................                                                  |  24%
##   ordinary text without R code
##
##
  |
  |..................                                                |  27%
## label: unnamed-chunk-36
##
  |
  |...................                                               |  30%
##   ordinary text without R code
##
##
  |
  |.....................                                             |  32%
## label: unnamed-chunk-37
##
  |
  |.......................                                           |  35%
##   ordinary text without R code
##
##
  |
  |.........................                                         |  38%
## label: unnamed-chunk-38
##
  |
  |..........................                                        |  41%
##   ordinary text without R code
##
##
  |
  |............................                                      |  43%
## label: unnamed-chunk-39
##
  |
  |..............................                                    |  46%
##   ordinary text without R code
##
##
  |
  |................................                                  |  49%
## label: unnamed-chunk-40
##
  |
  |.................................                                 |  51%
##   ordinary text without R code
##
##
  |
  |...................................                               |  54%
## label: unnamed-chunk-41
## Quitting from lines 126-133 (FinalMLassignment.Rmd)
```

```
## Error in eval(): argument "expr" is missing, with no default
```