# Predicting exercise quality with tracking devices

## Summary

The goal of this project is to generate a model that can predict if an exercise is well performed based on data collected from accelerometers on the belt, forearm, arm, and dumbell of 6 participants.The variable “classe” is the one to be predicted. First, the documentation about this study was explored to find more specific information about the dataset and the experiment: http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har (http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har) PaperVelloso, E.; Bulling, A.; Gellersen, H.; Ugulino, W.; Fuks, H. Qualitative Activity Recognition of Weight Lifting Exercises. Proceedings of 4th International Conference in Cooperation with SIGCHI (Augmented Human ’13) . Stuttgart, Germany: ACM SIGCHI, 2013. In this study, six young health participants were asked to perform one set of 10 repetitions of the unilateral dumbbell biceps curl in five different fashions: (“classe”" variable description) exactly according to the specification (Class A), throwing the elbows to the front (Class B), lifting the dumbbell only halfway (Class C), lowering the dumbbell only halfway (Class D) and throwing the hips to the front (Class E).

```
#load libraries
library(readr)
library(caret)
library(data.table)
library(dplyr)
library(GGally)
#set up directory: setwd("/Users/ireneramoslopez/Documents/Machine learning 2018")
#read training data
training <- read_csv("pml-training.csv")
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Warning in rbind(names(probs), probs_f): number of columns of result is not
## a multiple of vector length (arg 1)
```

```
## Warning: 185 parsing failures.
## row # A tibble: 5 x 5 col     row               col expected  actual               fi
le expected   <int>             <chr>    <chr>   <chr>              <chr> actual 1  2231
kurtosis_roll_arm a double #DIV/0! 'pml-training.csv' file 2  2231 skewness_roll_arm a d
ouble #DIV/0! 'pml-training.csv' row 3  2255 kurtosis_roll_arm a double #DIV/0! 'pml-tra
ining.csv' col 4  2255 skewness_roll_arm a double #DIV/0! 'pml-training.csv' expected 5
2282 kurtosis_roll_arm a double #DIV/0! 'pml-training.csv'
## ... ................. ...
.......................................................... ........
.......................................................... ......
.......................................................... ....
.......................................................... ...
.......................................................... ...
.......................................................... ........
..........................................................
## See problems(...) for more details.
```

# Exploratory analysis and feature selection

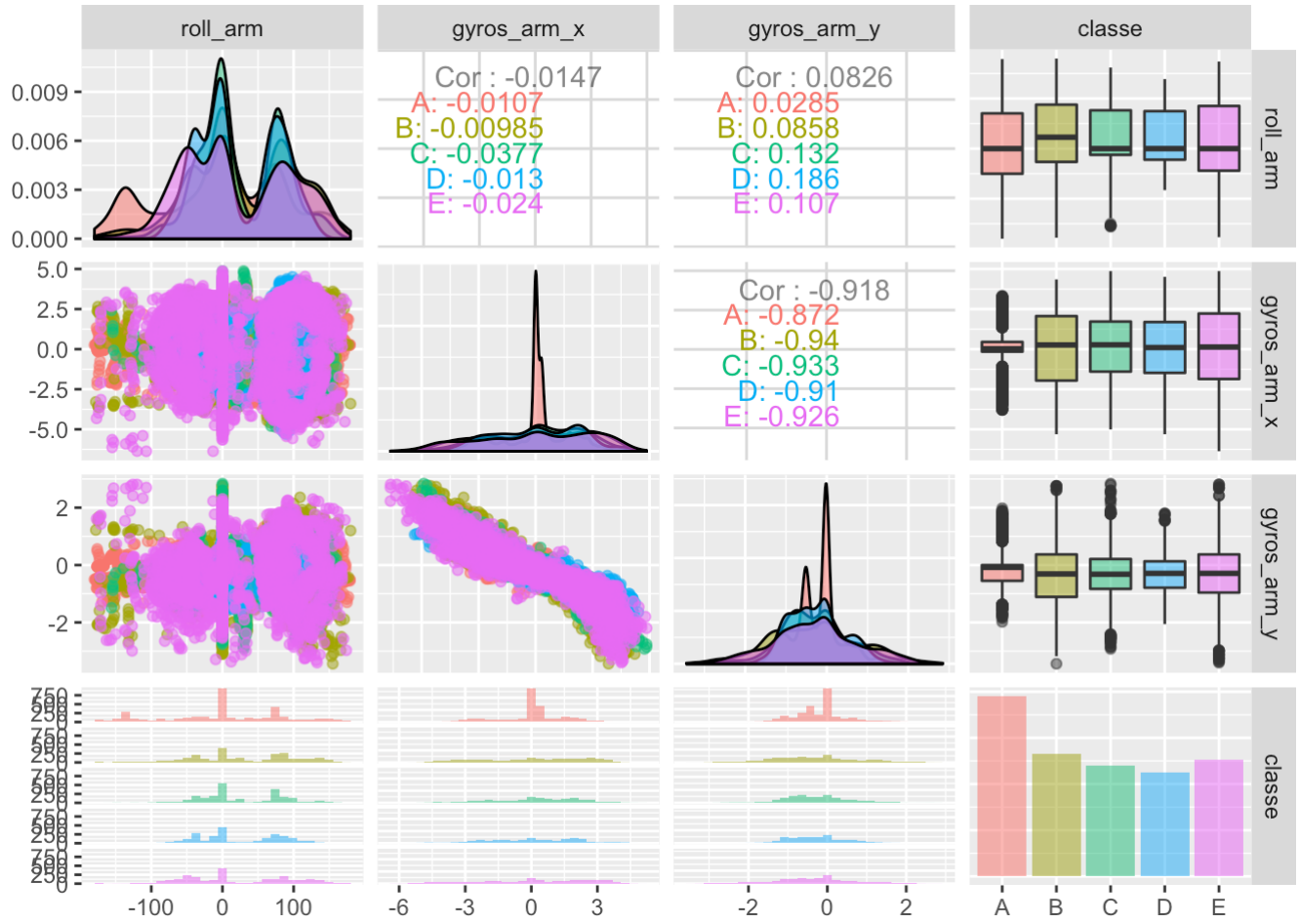
The training dataset has 160 variables. Therefore, to reduce the variables number of variables, those variables with near zero variance were identified and removed. Also some of the variables had most of the values missing. Those variables were also eliminated from the analysis. Identification of highly correlated variables was also done but they were left in the model since there was not a high number and they shouldn't affect the model. Lastly, the first 6 columns were also included as they did not contain any predictive information. The final dataset after this selection proscess had 52 predictors.

```
#exclude variables from the data set with no variability:
nzv <- nearZeroVar(training, saveMetrics = TRUE)
nzv <- setDT(nzv, keep.rownames = TRUE)[]
nzv_vars <- filter(nzv, nzv == FALSE)
# select variables no nzv in training
training2 <- training[nzv_vars$rn] #reduced to 119 vars (including classe)
#Many columns have mostly NA values, so we remove those from the model:
no_NA_columns <- colnames(training2)[colSums(is.na(training2)) < 19000]
training3 <- select(training2, no_NA_columns)
#reduced to 59 vars (including classe)
training4 <- training3[,7:59] #remove variables 1-6 from data set
```

As the purpose of the testing set provided in the assignment is to be used in the final quiz and does not contain the classe column, a partition of the training set in new "training" and "testing" sets was created to provide additional evaluation of the model.

```
#create data partition
trainingFinal <- createDataPartition(training4$classe, p = 0.7, list = FALSE)
training <- training4[trainingFinal, ]
testing <- training4[-trainingFinal, ]
```

Some more exploratory analysis was done plotting all the pre-selected variables against the others. Examples of the variables that seemed more predictive are shown below.


roll_arm
gyros_arm_x
gyros_arm_y
classe
Cor : -0.0147
A: -0.0107
B: -0.00985
C: -0.0377
D: -0.013
E: -0.024
Cor : 0.0826
A: 0.0285
B: 0.0858
C: 0.132
D: 0.186
E: 0.107
Cor : -0.918
A: -0.872
B: -0.94
C: -0.933
D: -0.91
E: -0.926
accel_arm_x
magnet_arm_x
magnet_arm_y
Cor : 0.814
A: 0.869
B: 0.774
C: 0.764
D: 0.791
E: 0.719
Cor : -0.699
A: -0.72
B: -0.702
C: -0.653
D: -0.595
E: -0.711
Cor : -0.79
A: -0.85
B: -0.721
C: -0.769
D: -0.82
E: -0.728

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removing 1 row that contained a missing value
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removing 1 row that contained a missing value
```

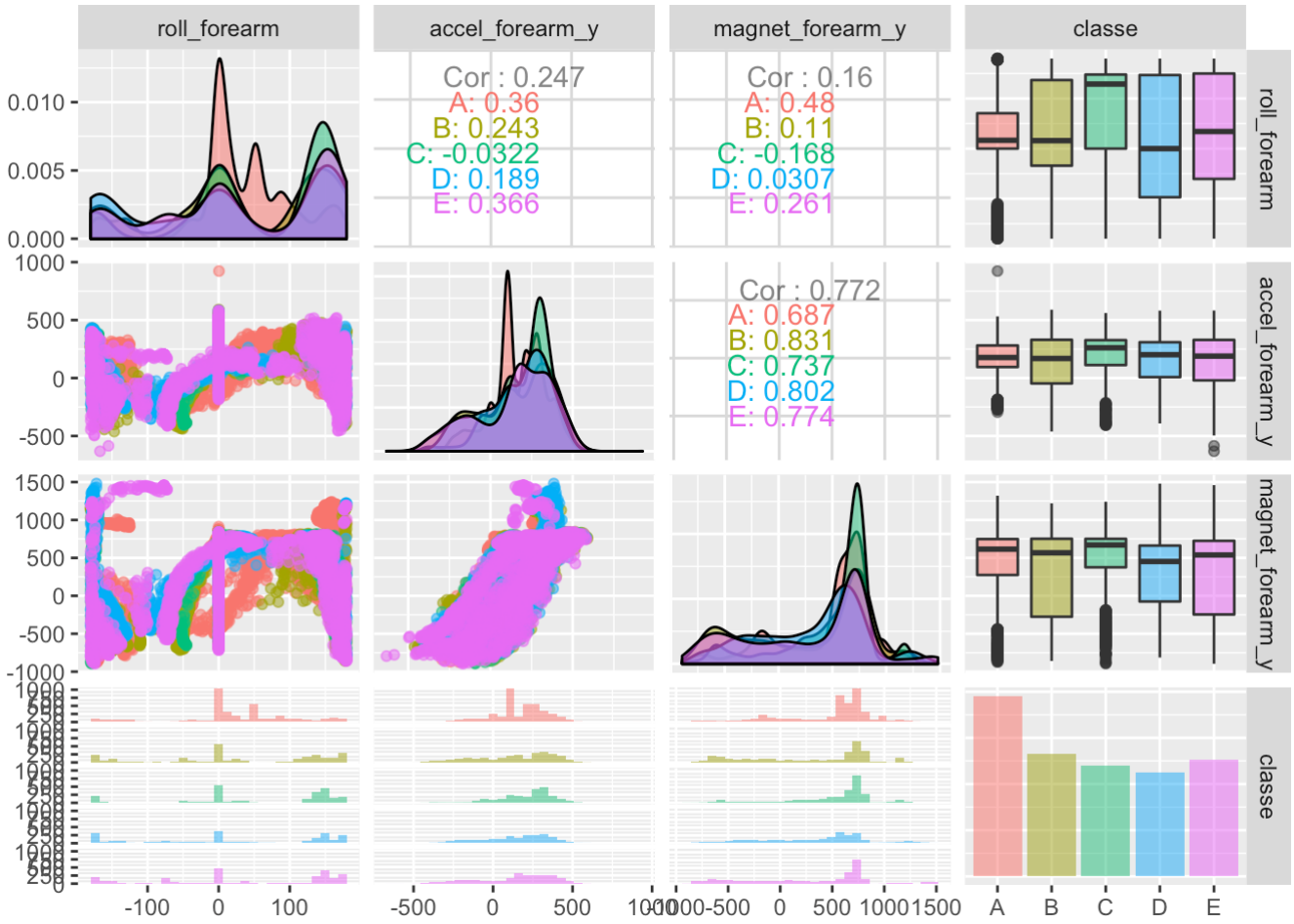
```
## Warning: Removed 1 rows containing missing values (geom_point).

## Warning: Removed 1 rows containing missing values (geom_point).
```

```
## Warning: Removed 1 rows containing non-finite values (stat_density).
```

```
## Warning: Removed 1 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 1 rows containing non-finite values (stat_bin).
```




# Building the model

Several models were tested, including classification trees ("rpart"), boosting with trees ("gmb") and random forest ("rf"). However the one that showed better accuracy was random trees, which is described below. 5-fold cross validation and parallel processing in caret was performed following these recomendations:

https://github.com/lgreski/datasciencectacontent/blob/master/markdown/pml-randomForestPerformance.md (https://github.com/lgreski/datasciencectacontent/blob/master/markdown/pml-randomForestPerformance.md)

```
##ramdon forest model
modelfit_rf <- train(as.factor(classe) ~ ., data = training, method = "rf",trControl = f
itControl,na.action = na.omit)
```

# Accuracy, cross validation and expected out of sample error

Accuracy, values helding out folds (cross validation) during the model building phase and classification errors can be observed below

```
## Random Forest
##
## 13737 samples
##    52 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 10989, 10990, 10990, 10988, 10987
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.9908276  0.9883965
##   27    0.9910455  0.9886734
##   52    0.9824553  0.9778080
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 27.
```

```
##    Accuracy     Kappa Resample
## 1 0.9905351 0.9880247    Fold1
## 2 0.9905317 0.9880249    Fold3
## 3 0.9916242 0.9894051    Fold2
## 4 0.9890869 0.9861978    Fold5
## 5 0.9934498 0.9917143    Fold4
```

```
##
## Call:
##  randomForest(x = x, y = y, mtry = param$mtry)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 27
##
##         OOB estimate of  error rate: 0.71%
## Confusion matrix:
##      A    B    C    D    E class.error
## A 3898    4    2    0    1 0.001792574
## B   17 2633    8    0    0 0.009405568
## C    0   13 2377    6    0 0.007929883
## D    1    1   27 2221    2 0.013765542
## E    0    1    4   10 2510 0.005940594
```

Finally, the model was evaluated in the testing dataset that was particioned from the initial training dataset. The different types of errors such as sensitivity, specificity, or positive and negative predictive values are shown below. Overall, the predictive model showed very good performance accross the board.

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1673   11    0    0    0
##          B    1 1124    6    0    0
##          C    0    4 1011    5    0
##          D    0    0    9  957    3
##          E    0    0    0    2 1079
##
## Overall Statistics
##
##                Accuracy : 0.993
##                  95% CI : (0.9906, 0.995)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9912
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            0.9994   0.9868   0.9854   0.9927   0.9972
## Specificity            0.9974   0.9985   0.9981   0.9976   0.9996
## Pos Pred Value         0.9935   0.9938   0.9912   0.9876   0.9981
## Neg Pred Value         0.9998   0.9968   0.9969   0.9986   0.9994
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2843   0.1910   0.1718   0.1626   0.1833
## Detection Prevalence   0.2862   0.1922   0.1733   0.1647   0.1837
## Balanced Accuracy      0.9984   0.9927   0.9918   0.9952   0.9984
```