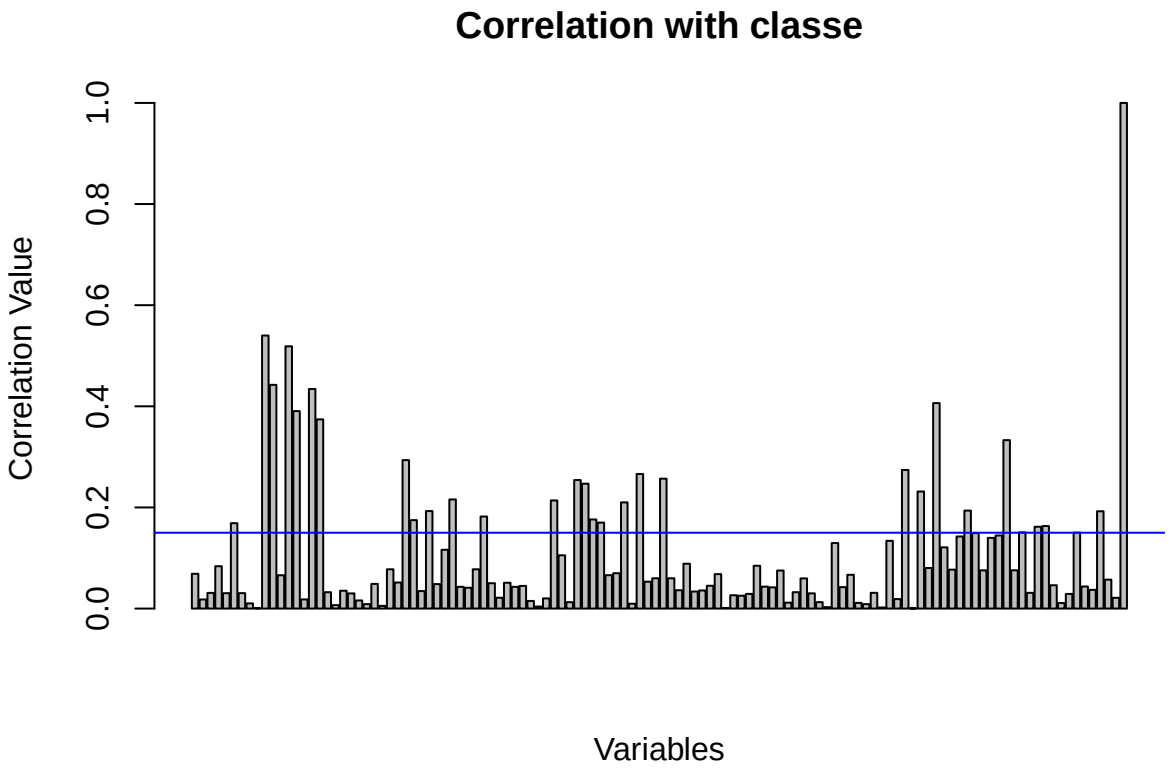
Data Analysis- Weight Llifting Exercise

In this report, data from weight lifting exercise dataset is analyzed. The objective is to find whether a given volunteer has performed a given exercise according to the specified instructions to quantify how "well" the exercise is carried out. The output of whether an exercise is done well is stored in variable "classe" where 'A' means the exercise is done well and letters 'B' to 'E' imply the exercise is not done according to the instructions.

The data is loaded. The training data has 19622 observations of 162 variables.

In order to calssify the test data, a model needs to be developed using the training data. To achieve that, it is important to appreciate that not all 160 variables may contribute anything significant to the classification of output variable 'classe'. Therefore, a correlation matrix is found for all those variables which passed the complete.case test.

From the correlation matrix, the following barplot is generated which shows how classe variable is linked to other variables.



The threshold is applied at 0.15 (the blue horizontal line). As can be seen from the barplot there are a significant number of variables that may have an affect on classe outcome based on the variables with correlation coefficient of at least 0.15. They are:

```
library(caret)
library(rattle)
library(randomForest)
```

```
dtrain <- read.csv("pml-training.csv", na.strings = c("NA", "")) ##Testing data
dtest <- read.csv("pml-testing.csv", na.strings = c("NA", "")) ##Training data

## Using only those variables as predictors whose correlation with classe variable is greater than 0.15
var_ind <- which(cor2$classe >0.15)
print(colnames(cor2[var_ind])[1:18])
```

```
##  [1] "max_picth_belt"       "amplitude_pitch_belt" "var_total_accel_belt"
##  [4] "stddev_roll_belt"     "var_roll_belt"        "stddev_pitch_belt"
##  [7] "var_pitch_belt"       "var_accel_arm"        "accel_arm_x"
## [10] "magnet_arm_x"         "amplitude_yaw_arm"    "pitch_forearm"
## [13] "max_roll_forearm"     "min_roll_forearm"     "total_accel_forearm"
## [16] "avg_pitch_forearm"    "stddev_yaw_forearm"   "var_yaw_forearm"
```

```
train_var <- colnames(cor2[var_ind])
dtrain1 <- dtrain[,train_var]

##k fold cross validation
trControl <- trainControl(method = "cv", number = 10)
rf1 <- train(dtrain1$classe ~ ., method = "rf", trControl = trControl, dtrain1)
rf1$finalModel
```

```
##
## 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: 2
##
##         OOB estimate of  error rate: 33%
## Confusion matrix:
##    A  B  C  D  E class.error
## A 79  9 17  3  1   0.2752294
## B 12 43 17  4  3   0.4556962
## C 10 15 35  8  2   0.5000000
## D  8  4  3 52  2   0.2463768
## E  4  3  6  3 63   0.2025316
```

As we can see the random forest is giving significant numbers for missclassification of classe variable.