Assignment4

Prahlad

6/13/2020

Load Packages

```
suppressMessages(library(ggplot2))
suppressMessages(library(caret))
suppressMessages(library(e1071))
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:ggplot2':
##
##     margin
```

Read Data

```
setwd("C:/Users/SPRAHLA2/Desktop/ml_rcourse")
#Load DataSet
df_training <- read.csv("C:/Users/SPRAHLA2/Desktop/ml_rcourse/pml-
training.csv",stringsAsFactors = FALSE)
df_testing <- read.csv("C:/Users/SPRAHLA2/Desktop/ml_rcourse/pml-
testing.csv",stringsAsFactors = FALSE)
```

Understanding Data

dimension of data

```
#colnames of data
dim(df_training)

## [1] 19622   160
```

160 variables

no of unique users

```
#no of unique users

unique(df_training$user_name)

## [1] "carlitos" "pedro"    "adelmo"   "charles"  "eurico"   "jeremy"
```

6 users in dataset

no of classes in classe

```
#classe dependent variable

unique(df_training$classe)

## [1] "A" "B" "C" "D" "E"
```

6 classe classes and number of users != number of classe

range of data

```
#data recording start and end time

min(df_training$cvtd_timestamp)

## [1] "02/12/2011 13:32"

max(df_training$cvtd_timestamp)

## [1] "30/11/2011 17:12"
```

Data is from november 30th to december 2nd

Are There Any Missing Value In Data?

```
which(sapply(df_training,function(x) sum(is.na(x))/nrow(df_training)*100)>95)

##            max_roll_belt           max_picth_belt            min_roll_belt
##                       18                       19                       21
##           min_pitch_belt      amplitude_roll_belt     amplitude_pitch_belt
##                       22                       24                       25
##     var_total_accel_belt           avg_roll_belt        stddev_roll_belt
##                       27                       28                       29
##            var_roll_belt           avg_pitch_belt       stddev_pitch_belt
##                       30                       31                       32
##           var_pitch_belt             avg_yaw_belt         stddev_yaw_belt
##                       33                       34                       35
##             var_yaw_belt           var_accel_arm            avg_roll_arm
##                       36                       50                       51
##          stddev_roll_arm            var_roll_arm           avg_pitch_arm
##                       52                       53                       54
##         stddev_pitch_arm           var_pitch_arm             avg_yaw_arm
##                       55                       56                       57
```

```
##           stddev_yaw_arm              var_yaw_arm             max_roll_arm
##                       58                       59                       75
##            max_picth_arm             max_yaw_arm             min_roll_arm
##                       76                       77                       78
##            min_pitch_arm             min_yaw_arm      amplitude_roll_arm
##                       79                       80                       81
##      amplitude_pitch_arm       amplitude_yaw_arm       max_roll_dumbbell
##                       82                       83                       93
##       max_picth_dumbbell       min_roll_dumbbell       min_pitch_dumbbell
##                       94                       96                       97
##  amplitude_roll_dumbbell amplitude_pitch_dumbbell      var_accel_dumbbell
##                       99                      100                      103
##        avg_roll_dumbbell     stddev_roll_dumbbell       var_roll_dumbbell
##                      104                      105                      106
##       avg_pitch_dumbbell    stddev_pitch_dumbbell       var_pitch_dumbbell
##                      107                      108                      109
##         avg_yaw_dumbbell      stddev_yaw_dumbbell         var_yaw_dumbbell
##                      110                      111                      112
##         max_roll_forearm        max_picth_forearm         min_roll_forearm
##                      131                      132                      134
##        min_pitch_forearm   amplitude_roll_forearm  amplitude_pitch_forearm
##                      135                      137                      138
##        var_accel_forearm         avg_roll_forearm      stddev_roll_forearm
##                      141                      142                      143
##         var_roll_forearm        avg_pitch_forearm     stddev_pitch_forearm
##                      144                      145                      146
##        var_pitch_forearm          avg_yaw_forearm       stddev_yaw_forearm
##                      147                      148                      149
##          var_yaw_forearm
##                      150
```

Yes there are missing values.67 variables have missing value % greater than 95%. We have to be careful of variable selection for model building

Taking only set of variables which are not having high percentage of missing value and leaving out statistical derived variables like (max,min,avg,skewness,kurtosis,variance,standard deviation) for analysis

```
colnames(df_training)
```

```
##   [1] "X"                        "user_name"
##   [3] "raw_timestamp_part_1"     "raw_timestamp_part_2"
##   [5] "cvtd_timestamp"           "new_window"
##   [7] "num_window"               "roll_belt"
##   [9] "pitch_belt"               "yaw_belt"
##  [11] "total_accel_belt"         "kurtosis_roll_belt"
##  [13] "kurtosis_picth_belt"      "kurtosis_yaw_belt"
```

```
##  [15] "skewness_roll_belt"       "skewness_roll_belt.1"
##  [17] "skewness_yaw_belt"        "max_roll_belt"
##  [19] "max_picth_belt"           "max_yaw_belt"
##  [21] "min_roll_belt"            "min_pitch_belt"
##  [23] "min_yaw_belt"             "amplitude_roll_belt"
##  [25] "amplitude_pitch_belt"     "amplitude_yaw_belt"
##  [27] "var_total_accel_belt"     "avg_roll_belt"
##  [29] "stddev_roll_belt"         "var_roll_belt"
##  [31] "avg_pitch_belt"           "stddev_pitch_belt"
##  [33] "var_pitch_belt"           "avg_yaw_belt"
##  [35] "stddev_yaw_belt"          "var_yaw_belt"
##  [37] "gyros_belt_x"             "gyros_belt_y"
##  [39] "gyros_belt_z"             "accel_belt_x"
##  [41] "accel_belt_y"             "accel_belt_z"
##  [43] "magnet_belt_x"            "magnet_belt_y"
##  [45] "magnet_belt_z"            "roll_arm"
##  [47] "pitch_arm"                "yaw_arm"
##  [49] "total_accel_arm"          "var_accel_arm"
##  [51] "avg_roll_arm"             "stddev_roll_arm"
##  [53] "var_roll_arm"             "avg_pitch_arm"
##  [55] "stddev_pitch_arm"         "var_pitch_arm"
##  [57] "avg_yaw_arm"              "stddev_yaw_arm"
##  [59] "var_yaw_arm"              "gyros_arm_x"
##  [61] "gyros_arm_y"              "gyros_arm_z"
##  [63] "accel_arm_x"              "accel_arm_y"
##  [65] "accel_arm_z"              "magnet_arm_x"
##  [67] "magnet_arm_y"             "magnet_arm_z"
##  [69] "kurtosis_roll_arm"        "kurtosis_picth_arm"
##  [71] "kurtosis_yaw_arm"         "skewness_roll_arm"
##  [73] "skewness_pitch_arm"       "skewness_yaw_arm"
##  [75] "max_roll_arm"             "max_picth_arm"
##  [77] "max_yaw_arm"              "min_roll_arm"
##  [79] "min_pitch_arm"            "min_yaw_arm"
##  [81] "amplitude_roll_arm"       "amplitude_pitch_arm"
##  [83] "amplitude_yaw_arm"        "roll_dumbbell"
##  [85] "pitch_dumbbell"           "yaw_dumbbell"
##  [87] "kurtosis_roll_dumbbell"   "kurtosis_picth_dumbbell"
##  [89] "kurtosis_yaw_dumbbell"    "skewness_roll_dumbbell"
##  [91] "skewness_pitch_dumbbell"  "skewness_yaw_dumbbell"
##  [93] "max_roll_dumbbell"        "max_picth_dumbbell"
##  [95] "max_yaw_dumbbell"         "min_roll_dumbbell"
##  [97] "min_pitch_dumbbell"       "min_yaw_dumbbell"
##  [99] "amplitude_roll_dumbbell"  "amplitude_pitch_dumbbell"
## [101] "amplitude_yaw_dumbbell"   "total_accel_dumbbell"
## [103] "var_accel_dumbbell"       "avg_roll_dumbbell"
## [105] "stddev_roll_dumbbell"     "var_roll_dumbbell"
## [107] "avg_pitch_dumbbell"       "stddev_pitch_dumbbell"
## [109] "var_pitch_dumbbell"       "avg_yaw_dumbbell"
## [111] "stddev_yaw_dumbbell"      "var_yaw_dumbbell"
## [113] "gyros_dumbbell_x"         "gyros_dumbbell_y"
```

```
## [115] "gyros_dumbbell_z"          "accel_dumbbell_x"
## [117] "accel_dumbbell_y"          "accel_dumbbell_z"
## [119] "magnet_dumbbell_x"         "magnet_dumbbell_y"
## [121] "magnet_dumbbell_z"         "roll_forearm"
## [123] "pitch_forearm"             "yaw_forearm"
## [125] "kurtosis_roll_forearm"     "kurtosis_picth_forearm"
## [127] "kurtosis_yaw_forearm"      "skewness_roll_forearm"
## [129] "skewness_pitch_forearm"    "skewness_yaw_forearm"
## [131] "max_roll_forearm"          "max_picth_forearm"
## [133] "max_yaw_forearm"           "min_roll_forearm"
## [135] "min_pitch_forearm"         "min_yaw_forearm"
## [137] "amplitude_roll_forearm"    "amplitude_pitch_forearm"
## [139] "amplitude_yaw_forearm"     "total_accel_forearm"
## [141] "var_accel_forearm"         "avg_roll_forearm"
## [143] "stddev_roll_forearm"       "var_roll_forearm"
## [145] "avg_pitch_forearm"         "stddev_pitch_forearm"
## [147] "var_pitch_forearm"         "avg_yaw_forearm"
## [149] "stddev_yaw_forearm"        "var_yaw_forearm"
## [151] "gyros_forearm_x"           "gyros_forearm_y"
## [153] "gyros_forearm_z"           "accel_forearm_x"
## [155] "accel_forearm_y"           "accel_forearm_z"
## [157] "magnet_forearm_x"          "magnet_forearm_y"
## [159] "magnet_forearm_z"          "classe"
```

data variable selection

```
df_training_selected <-
df_training[,c("user_name","cvtd_timestamp","new_window","num_window","roll_b
elt","pitch_belt","yaw_belt","total_accel_belt","roll_arm","pitch_arm","yaw_a
rm","total_accel_arm","roll_dumbbell","pitch_dumbbell","yaw_dumbbell","total_
accel_dumbbell","roll_forearm","pitch_forearm","yaw_forearm","total_accel_for
earm","classe")]
```

check missing value %

```
sapply(df_training_selected,function(x) sum(is.na(x))/nrow(df_training)*100)
```

```
##            user_name       cvtd_timestamp           new_window
##                    0                    0                    0
##           num_window            roll_belt           pitch_belt
##                    0                    0                    0
##             yaw_belt     total_accel_belt             roll_arm
##                    0                    0                    0
##            pitch_arm              yaw_arm      total_accel_arm
##                    0                    0                    0
##        roll_dumbbell       pitch_dumbbell         yaw_dumbbell
##                    0                    0                    0
## total_accel_dumbbell         roll_forearm        pitch_forearm
##                    0                    0                    0
##          yaw_forearm  total_accel_forearm               classe
##                    0                    0                    0
```

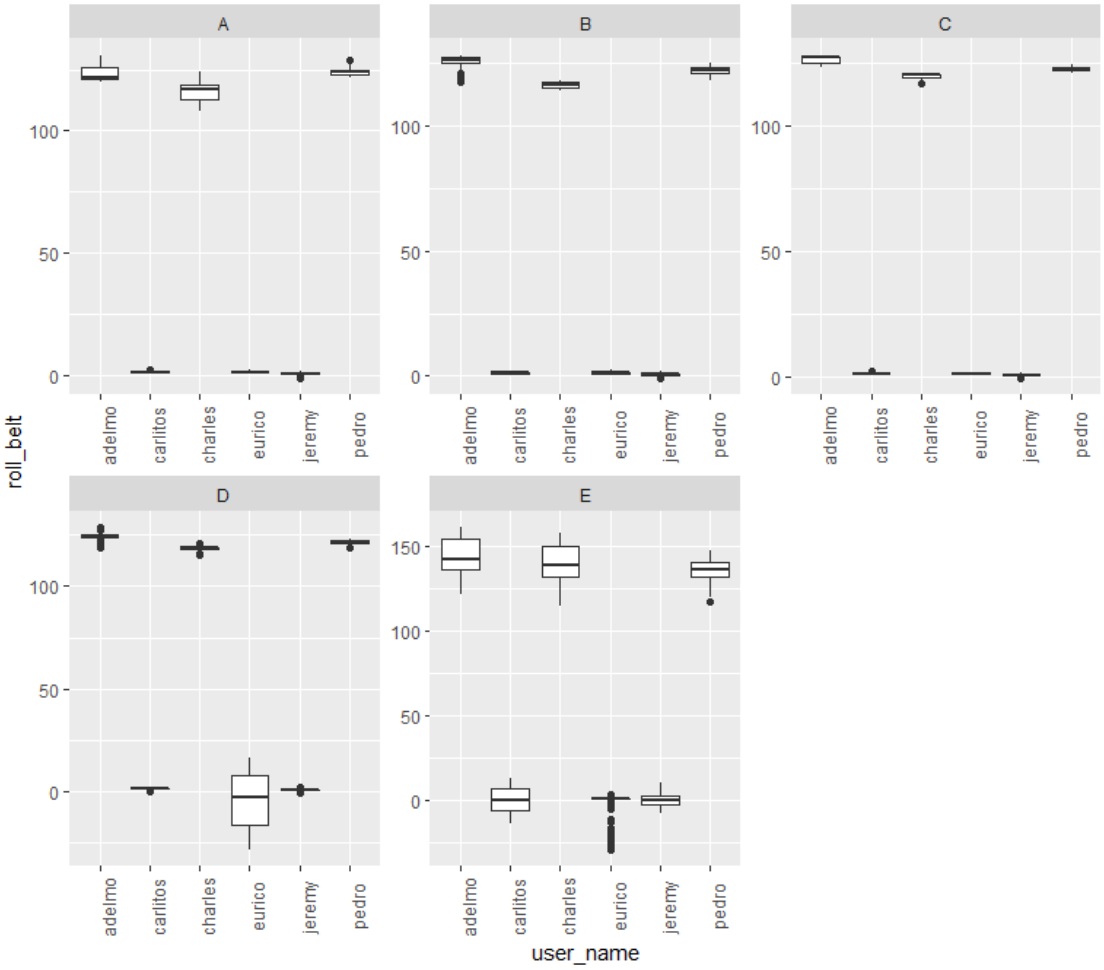
Exploratory Analysis(Only 5 plots are recommended so i am picking just one variable in each of arm,dumbbell,forearm,belt from selected variable done in the above step)

Usage of belt,arm,dumbbell,forearm among users each day(4 days!)

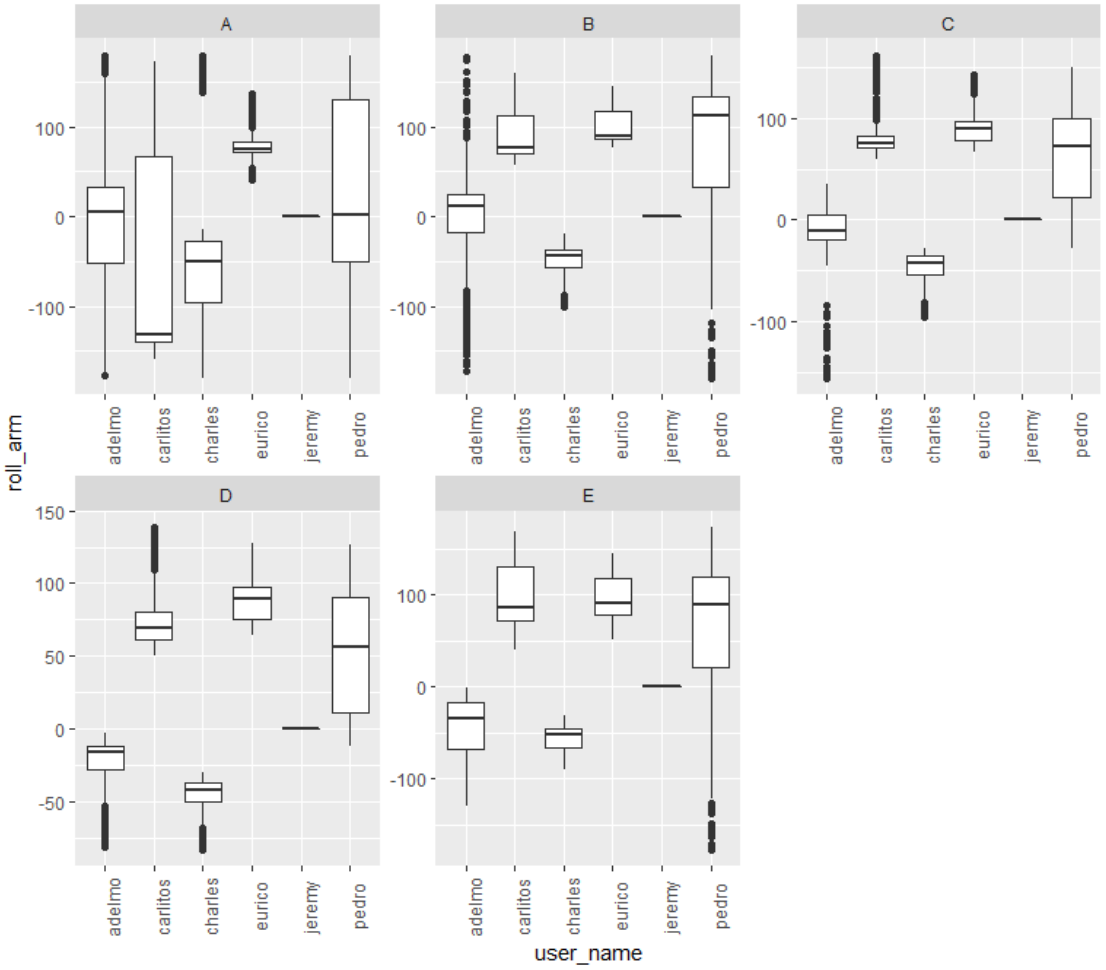
```
df_training_selected$day <-
as.character(as.Date(df_training_selected$cvtd_timestamp,"%d/%m/%Y"))
table(df_training_selected$user_name,df_training_selected$day)

##
##            2011-11-28 2011-11-30 2011-12-02 2011-12-05
##   adelmo            0          0       3892          0
##   carlitos          0          0          0       3112
##   charles           0          0       3536          0
##   eurico         3070          0          0          0
##   jeremy            0       3402          0          0
##   pedro             0          0          0       2610
```

In distribution of 4 days usage each user has tried on different days

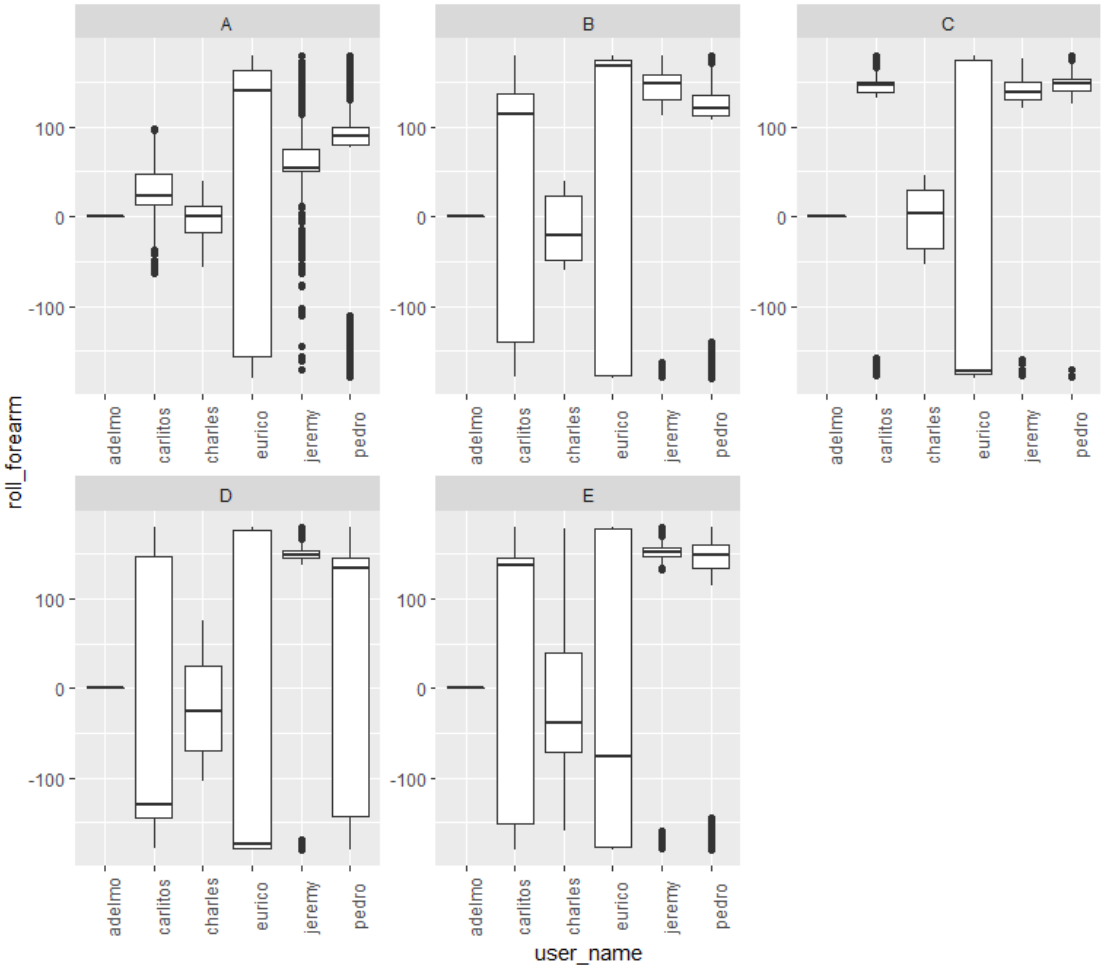
Performance of classe exercise around belt



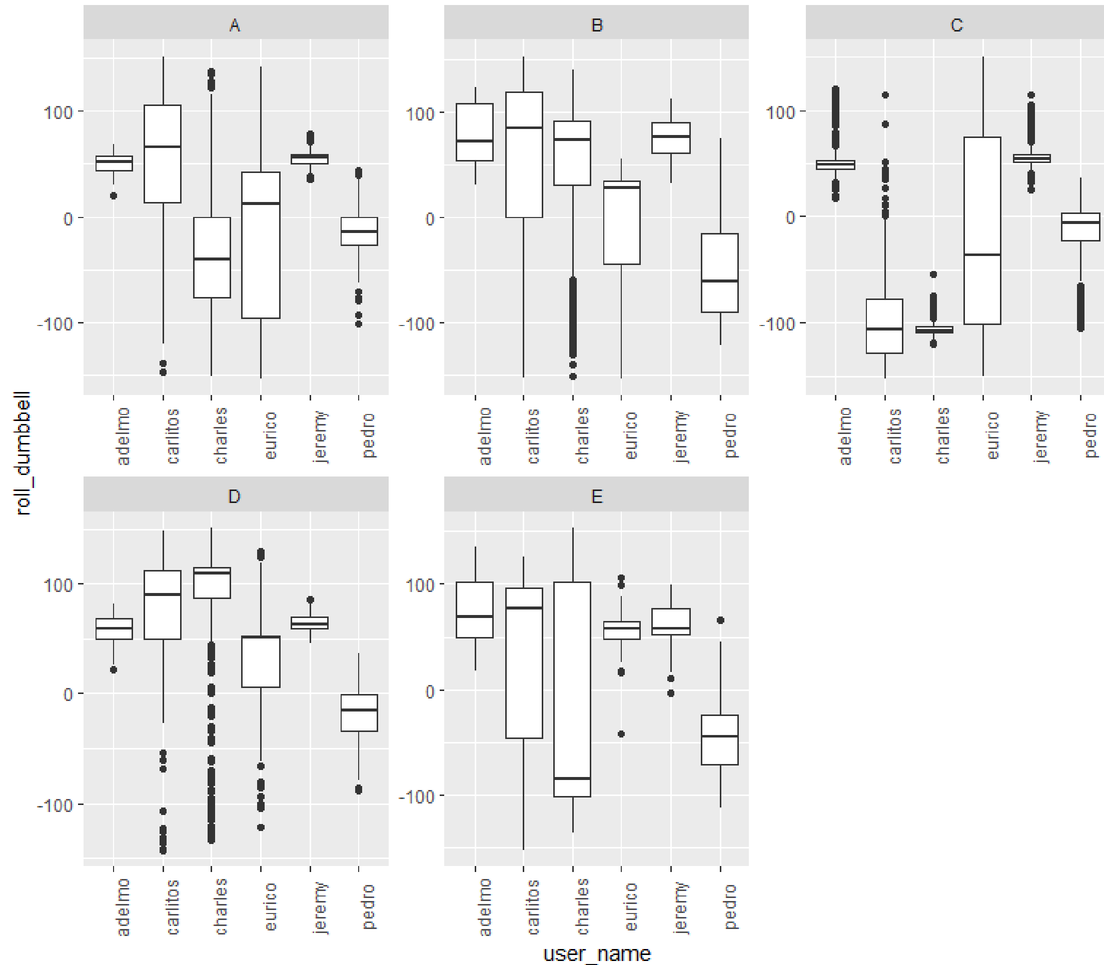
Performance of classe exercise around arm



Performance of classe exercise around fore arm



Performance of classe exercise around dumbbell



There is variablity in each user for each classe

Check For Multicollinearity Between Numerical Variables

```
suppressMessages(library(pander))
p <- sapply(df_training_selected,function(x) is.numeric(x))
t <- df_training_selected[,p]
#Remove Na's
q1 <- na.omit(t)
c <-as.data.frame(cor(q1))
panderOptions('table.split.table', Inf)
pander(c)
```

	num_window	roll_bel	pitch_bel	yaw_bel	total_accel_bel	roll_ar	pitch_arm	yaw_ar	total_accel_ar	roll_dumbb	pitch_dumbbel	yaw_dumbbel	total_accel_dumbbell	roll_forear	pitch_forear	yaw_forearm	total_accel_forearm

		t	t	t	t	m		m	m	ell	l	l		m	m		
num_window	1	0.071998	-0.102389	0.085889	0.066003	-0.119899	-0.033888	-0.047998	-0.064555	0.034007	-0.1527	-0.1084	0.08873	-0.011133	-0.046997	-0.084133	-0.02651
roll_belt	0.07198	1	-0.215959	0.815252	0.980809	-0.37272	-0.059777	-0.22259	-0.27833	-0.126265	0.064003	0.025664	-0.19021	-0.15026	0.1746	-0.26664	0.06556
pitch_belt	-0.10239	-0.215959	1	-0.669898	-0.13939	0.170202	-0.161615	0.143434	0.093302	-0.350202	0.2316	0.66	-0.31175	0.14152	0.2536	0.048511	-0.32993
yaw_belt	0.085889	0.815252	-0.669898	1	0.7621	-0.333669	0.136262	-0.228286	-0.240767	0.097713	-0.093312	-0.3477	0.04576	-0.18007	-0.030771	-0.19757	0.2278
total_accel_belt	0.066603	0.980809	-0.139	0.7621	1	-0.361515	0.053611	-0.210104	-0.26666	-0.131314	0.058995	0.060443	-0.17114	-0.11127	0.1845	-0.23887	0.03222
roll_arm	-0.11999	-0.37272	0.170202	-0.333669	-0.3615	1	-0.041136	0.40707	0.057171	-0.149292	0.2286	0.2796	-0.15554	0.075888	0.1657	0.2365	-0.01191
pitch_arm	-0.03388	0.059797	-0.16162	0.136262	0.053611	-0.041	1	-0.08202	0.030889	0.021554	-0.1526	-0.1784	0.19772	0.022887	-0.1545	0.079777	0.1089

		7	5			36		35								
yaw_arm	-0.04798	-0.22559	0.14334	-0.22866	-0.2104	0.4007	-0.08235	1	-0.01491	-0.1125	0.08696	0.1443	-0.06311	0.05716	0.06093	0.1845	-0.11129
total_accel_arm	-0.06455	-0.27833	0.09302	-0.24007	-0.266	0.05771	0.03089	-0.01491	1	-0.00529	-0.07226	-0.02883	0.137	0.02375	-0.18677	0.09288	-0.07364
roll_dumbbell	0.03407	-0.12665	-0.35023	0.09713	-0.13144	-0.14925	0.02154	-0.1255	-0.00529	1	0.2129	-0.2771	0.3599	-0.02392	-0.1083	-0.00023	0.1871
pitch_dumbbell	-0.1527	0.06403	0.23166	-0.09312	0.05895	0.2286	-0.15266	0.08696	-0.07226	0.2129	1	0.5174	-0.4052	-0.07236	0.2908	-0.09264	0.01484
yaw_dumbbell	-0.1084	0.02564	0.666	-0.34777	0.06043	0.2796	-0.17843	0.14443	-0.02883	-0.2771	0.5174	1	-0.5267	0.01276	0.2804	-0.05579	-0.1894
total_accel_dumbbell	0.08873	-0.19211	-0.31756	-0.04576	-0.1714	-0.15554	0.19722	-0.06311	0.1137	0.3599	-0.4052	-0.5267	1	0.2199	-0.3449	0.22889	0.1257
roll_fore	-0.	-0.	0.1	-0.	-0.1	0.0	0.0	0.0	0.023	-0.0	-0.0	0.012	0.219	1	-0.0	0.34	-0.08

arm	01113	1502	452	1807	127	7588	2287	5716	75	2392	7236	76			5441	67	146
pitch_forearm	-0.04697	0.17446	0.25336	-0.030071	0.1845	0.16557	-0.15445	0.060993	-0.1867	-0.1083	0.2908	0.2804	-0.3449	-0.05441	1	-0.2208	-0.1793
yaw_forearm	-0.08413	-0.26664	0.048551	-0.19957	-0.2387	0.23665	0.079777	0.18445	0.09288	-0.002381	-0.09264	-0.05579	0.2289	0.3467	-0.2208	1	0.1998
total_accel_forearm	-0.02651	0.065556	-0.32293	0.2278	0.03222	-0.011191	0.1089	-0.11129	-0.07364	0.1871	0.01484	-0.1894	0.1257	-0.08146	-0.1793	0.1998	1

Multi collinearity exists but since we are building tree based models including rf where each tree is different.we can try to remove them in further iterations

Model Building

distribution of classe variable

```
table(df_training_selected$classe)
```

```
##
##    A    B    C    D    E
## 5580 3797 3422 3216 3607
```

Equal distribution of classes. We can go for accuracy as metric

Building a default RF with no hyper parameters and OOB is a CV metric

1)Since the user names are same in both train and test we can use user name

2) dropping new wndow as test set only has value “no”

```
#convert dependent to factor
df_training_selected$classe <- as.factor(df_training_selected$classe)

# dropping unwanted column

df_training_selected$day <- NULL

#character to factor
df_training_selected$new_window <- as.factor(df_training_selected$new_window)
df_training_selected$user_name <- as.factor(df_training_selected$user_name)

set.seed(3457)

#training
model_rf <- randomForest(classe~.-cvtd_timestamp-
new_window,data=df_training_selected)

model_rf

##
## Call:
##  randomForest(formula = classe ~ . - cvtd_timestamp - new_window,
data = df_training_selected)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 0.09%
## Confusion matrix:
##      A    B    C    D    E  class.error
## A 5580    0    0    0    0 0.0000000000
## B    1 3795    0    1    0 0.0005267316
## C    0    4 3416    2    0 0.0017533606
## D    0    1    1 3213    1 0.0009328358
## E    0    0    3    3 3601 0.0016634322
```

The accuracy is 0.99 and the oob error is 1% wich is equivalent to validation test error

Varimp

```
varImp(model_rf)
```

```
##                        Overall
## user_name             457.9678
## num_window           2942.2121
## roll_belt            1936.1633
## pitch_belt           1253.7556
## yaw_belt             1541.9041
## total_accel_belt      538.2202
## roll_arm              524.8097
## pitch_arm             272.3909
## yaw_arm               437.8473
## total_accel_arm       233.3761
## roll_dumbbell         843.3334
## pitch_dumbbell        451.4661
## yaw_dumbbell          626.2025
## total_accel_dumbbell  689.6410
## roll_forearm          930.6746
## pitch_forearm        1225.7259
## yaw_forearm           384.6047
## total_accel_forearm   220.3053
```

we could see from model_rf that variables user_name,num_window,roll_belt,pitch_belt,yaw_belt have high importance compared to other predictors

Prediction on test data

```
df_testing_selected <-
df_testing[,c("user_name","cvtd_timestamp","new_window","num_window","roll_belt","pitch_belt","yaw_belt","total_accel_belt","roll_arm","pitch_arm","yaw_arm","total_accel_arm","roll_dumbbell","pitch_dumbbell","yaw_dumbbell","total_accel_dumbbell","roll_forearm","pitch_forearm","yaw_forearm","total_accel_forearm")]

#character to factor
df_testing_selected$new_window <- as.factor(df_testing_selected$new_window)
df_testing_selected$user_name <- as.factor(df_testing_selected$user_name)

predictions <- predict(model_rf,df_testing_selected)

predictions
```

```
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
##  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
```