Course8Project

John David

8/23/2019

Weight Lifting - Machine Learning Project

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 dumbell 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).

Approach: My approach was to explore the data set and analyze what actions best describe how to correctly curl weight. So I explored cleaned and explored the data and used that subset of data to form my models. I did a K fold cross validation, K=20 and I built an qda model and a lda model and had well ok results then I used a random forest model and tuned it to achieve a 98.65% accuracy on my test data.

Conclusion: It was fun to analyze the data myself but I could have used the randomforest to do it. All in all I enjoyed this lab and used all my previous John Hopkins classes to find this success.

References: http://www-personal.umich.edu/~johannb/Papers/paper63.pdf https://www.google.com/search?q=pitch+roll+and+yaw&tbm=isch&source=iu&ictx=1&fir=4R5jctF0uP_q5M%253A%252CrwdN0Ut4Lf6FUM%252C%252Fm%252F04gmp6w&vet=1&usg=AI4_-kR4vLK3GkH5rhntURhlMELz4kTd0g&sa=X&ved=2ahUKEwiF-4jEhZnkAhVldt8KHU2fB3UQ_B0wG3oECAEQAw#imgrc=4R5jctF0uP_q5M:

```
library("dplyr")

## Warning: package 'dplyr' was built under R version 3.5.2

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
## 
##     intersect, setdiff, setequal, union
```

```
library("ggplot2")
library("mlbench")
library("caret")
```

```
## Warning: package 'caret' was built under R version 3.5.2
```

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

```
#library("MASS")

apply.type.rules <- function(tmpDF){
  if (tmpDF["classe"] == "A"){
    return (1)
  }else if (tmpDF["classe"] == "B"){
    return (2)
  }else if (tmpDF["classe"] == "C"){
    return (3)
  }else if (tmpDF["classe"] == "D"){
    return (4)
  }else if (tmpDF["classe"] == "E"){
    return (5)
  }else if (tmpDF["classe"] == "F"){
    return (6)
  }else{
    return (100)
  }

}




trainingDF <- read.csv(file="/Course_8_Machine_Learning/Project/pml-training.
csv", stringsAsFactors = FALSE)

#summary(trainingDF)
#str(trainingDF)
names(trainingDF)
```

```
##   [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"
```

```
str(trainingDF)
```

```
## 'data.frame':    19622 obs. of  160 variables:
##  $ X                       : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ user_name               : chr  "carlitos" "carlitos" "carlitos" "carlit
os" ...
##  $ raw_timestamp_part_1    : int  1323084231 1323084231 1323084231 1323084
232 1323084232 1323084232 1323084232 1323084232 1323084232 1323084232 ...
##  $ raw_timestamp_part_2    : int  788290 808298 820366 120339 196328 30427
7 368296 440390 484323 484434 ...
##  $ cvtd_timestamp          : chr  "05/12/2011 11:23" "05/12/2011 11:23" "0
5/12/2011 11:23" "05/12/2011 11:23" ...
##  $ new_window              : chr  "no" "no" "no" "no" ...
##  $ 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      : chr  "" "" "" "" ...
##  $ kurtosis_picth_belt     : chr  "" "" "" "" ...
##  $ kurtosis_yaw_belt       : chr  "" "" "" "" ...
##  $ skewness_roll_belt      : chr  "" "" "" "" ...
##  $ skewness_roll_belt.1    : chr  "" "" "" "" ...
##  $ skewness_yaw_belt       : chr  "" "" "" "" ...
##  $ max_roll_belt           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ max_picth_belt          : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ max_yaw_belt            : chr  "" "" "" "" ...
##  $ 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            : chr  "" "" "" "" ...
##  $ 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      : chr  "" "" "" "" ...
##  $ 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 -0.0
2 -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 ...
##  $ var_roll_arm            : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ avg_pitch_arm           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ stddev_pitch_arm        : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ var_pitch_arm           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ avg_yaw_arm             : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ stddev_yaw_arm          : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ var_yaw_arm             : num  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       : chr  "" "" "" "" ...
##  $ kurtosis_picth_arm      : chr  "" "" "" "" ...
##  $ kurtosis_yaw_arm        : chr  "" "" "" "" ...
##  $ skewness_roll_arm       : chr  "" "" "" "" ...
##  $ skewness_pitch_arm      : chr  "" "" "" "" ...
##  $ skewness_yaw_arm        : chr  "" "" "" "" ...
##  $ max_roll_arm            : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ max_picth_arm           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ max_yaw_arm             : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ min_roll_arm            : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ min_pitch_arm           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ min_yaw_arm             : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ amplitude_roll_arm      : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ amplitude_pitch_arm     : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ amplitude_yaw_arm       : int  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  : chr  "" "" "" "" ...
##  $ kurtosis_picth_dumbbell : chr  "" "" "" "" ...
##  $ kurtosis_yaw_dumbbell   : chr  "" "" "" "" ...
##  $ skewness_roll_dumbbell  : chr  "" "" "" "" ...
##  $ skewness_pitch_dumbbell : chr  "" "" "" "" ...
##  $ skewness_yaw_dumbbell   : chr  "" "" "" "" ...
##  $ max_roll_dumbbell       : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ max_picth_dumbbell      : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ max_yaw_dumbbell        : chr  "" "" "" "" ...
##  $ min_roll_dumbbell       : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ min_pitch_dumbbell      : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ min_yaw_dumbbell        : chr  "" "" "" "" ...
##  $ amplitude_roll_dumbbell : num  NA NA NA NA NA NA NA NA NA NA ...
##   [list output truncated]
```

```
classeADF <- trainingDF[trainingDF['classe'] == "A",]
classeBDF <- trainingDF[trainingDF['classe'] == "B",]
classeCDF <- trainingDF[trainingDF['classe'] == "C",]
classeDDF <- trainingDF[trainingDF['classe'] == "D",]
classeEDF <- trainingDF[trainingDF['classe'] == "E",]
classeFDF <- trainingDF[trainingDF['classe'] == "F",]

#classADumbbellDF <- select(classeADF,roll_dumbbell,pitch_dumbbell,yaw_dumbbe
ll)
#colnames(classADumbbellDF) <- c("Roll.Dumbbell","Pitch.Dumbbell","Yaw.Dumbbe
ll")

#head(classADumbbellDF)
#dim(classADumbbellDF)

#n = 3510:(3510+470)

#classeAForearmDF <- select(classeADF,roll_forearm, pitch_forearm, yaw_forear
m)
#colnames(classeAForearmDF) <- c("Roll.Forearm","Pitch.Forearm","Yaw.Forearm"
)
#plot(classeAForearmDF$pitch_forearm[n])

#minPV <- apply(classeAForearmDF,2,min)
#abs(minPV)
#tempDF2 <- mutate(classeAForearmDF, log2Pitch = log2(pitch_forearm+abs(minPV
)))

#head(tempDF2,10)
#plot(tempDF2$log2Pitch[n])
#useless
#classAArmDF <- select(classeADF, roll_arm, pitch_arm, yaw_arm)
#plot(classAArmDF$pitch_arm)

#classeABeltDF <- select(classeADF,roll_belt,pitch_belt,yaw_belt)
#plot(classeABeltDF$pitch_belt[n])

#useless
#classeAGyroForearmDF <- select(classeADF, gyros_forearm_x, gyros_forearm_y,
gyros_forearm_z)
#plot(classeAGyroForearmDF$gyros_forearm_z)

#classeAGyroArmDF <- select(classeADF, gyros_arm_x, gyros_arm_y, gyros_arm_z)
#plot(classeAGyroArmDF$gyros_arm_z)

#classeAGyroBeltDF <- select(classeADF, gyros_belt_x, gyros_belt_y, gyros_bel
t_z)
#plot(classeAGyroBeltDF$gyros_belt_z)



#correlMtrx <- cor(trainingDF[,8:159])
#highlyCorrel <- findCorrelation(correlMtrx, cutoff=0.5)
#print(highlyCorrel)



selectedInputDF = select(trainingDF, roll_forearm, pitch_forearm, yaw_forearm
, roll_arm, pitch_arm, yaw_arm, roll_belt, pitch_belt, yaw_belt, classe)

typeAry <- c()
for (k in 1:nrow(selectedInputDF)){
  typeAry <- c(typeAry,"NA")
}

#selectedInputDF$Type=typeAry

head(selectedInputDF,10)
```

```
##    roll_forearm pitch_forearm yaw_forearm roll_arm pitch_arm yaw_arm
## 1          28.4         -63.9        -153     -128      22.5    -161
## 2          28.3         -63.9        -153     -128      22.5    -161
## 3          28.3         -63.9        -152     -128      22.5    -161
## 4          28.1         -63.9        -152     -128      22.1    -161
## 5          28.0         -63.9        -152     -128      22.1    -161
## 6          27.9         -63.9        -152     -128      22.0    -161
## 7          27.9         -63.9        -152     -128      21.9    -161
## 8          27.8         -63.8        -152     -128      21.8    -161
## 9          27.7         -63.8        -152     -128      21.7    -161
## 10         27.7         -63.8        -152     -128      21.6    -161
##    roll_belt pitch_belt yaw_belt classe
## 1       1.41       8.07    -94.4      A
## 2       1.41       8.07    -94.4      A
## 3       1.42       8.07    -94.4      A
## 4       1.48       8.05    -94.4      A
## 5       1.48       8.07    -94.4      A
## 6       1.45       8.06    -94.4      A
## 7       1.42       8.09    -94.4      A
## 8       1.42       8.13    -94.4      A
## 9       1.43       8.16    -94.4      A
## 10      1.45       8.17    -94.4      A
```

```
tail(selectedInputDF)
```

```
##       roll_forearm pitch_forearm yaw_forearm roll_arm pitch_arm yaw_arm
## 19617            0             0           0    -99.1     -33.7    79.4
## 19618            0             0           0    -99.4     -33.8    79.0
## 19619            0             0           0    -99.6     -34.5    77.3
## 19620            0             0           0    -99.6     -35.1    76.3
## 19621            0             0           0    -98.6     -36.7    73.5
## 19622            0             0           0    -97.6     -37.7    71.5
##       roll_belt pitch_belt yaw_belt classe
## 19617       148      -34.7      129      E
## 19618       147      -34.8      129      E
## 19619       145      -35.3      130      E
## 19620       145      -35.5      130      E
## 19621       143      -35.9      131      E
## 19622       143      -36.0      132      E
```

```
#str(selectedInputDF)

Type <- apply(X=selectedInputDF,MARGIN=1,FUN=apply.type.rules)

str(Type)
```

```
##  num [1:19622] 1 1 1 1 1 1 1 1 1 1 ...
```

```
finalDF <- cbind(selectedInputDF,Type)

head(finalDF)
```

```
##   roll_forearm pitch_forearm yaw_forearm roll_arm pitch_arm yaw_arm
## 1         28.4         -63.9        -153     -128      22.5    -161
## 2         28.3         -63.9        -153     -128      22.5    -161
## 3         28.3         -63.9        -152     -128      22.5    -161
## 4         28.1         -63.9        -152     -128      22.1    -161
## 5         28.0         -63.9        -152     -128      22.1    -161
## 6         27.9         -63.9        -152     -128      22.0    -161
##   roll_belt pitch_belt yaw_belt classe Type
## 1      1.41       8.07    -94.4      A    1
## 2      1.41       8.07    -94.4      A    1
## 3      1.42       8.07    -94.4      A    1
## 4      1.48       8.05    -94.4      A    1
## 5      1.48       8.07    -94.4      A    1
## 6      1.45       8.06    -94.4      A    1
```

```
tail(finalDF)
```

```
##       roll_forearm pitch_forearm yaw_forearm roll_arm pitch_arm yaw_arm
## 19617            0             0           0    -99.1     -33.7    79.4
## 19618            0             0           0    -99.4     -33.8    79.0
## 19619            0             0           0    -99.6     -34.5    77.3
## 19620            0             0           0    -99.6     -35.1    76.3
## 19621            0             0           0    -98.6     -36.7    73.5
## 19622            0             0           0    -97.6     -37.7    71.5
##       roll_belt pitch_belt yaw_belt classe Type
## 19617       148      -34.7      129      E    5
## 19618       147      -34.8      129       E    5
## 19619       145      -35.3      130       E    5
## 19620       145      -35.5      130       E    5
## 19621       143      -35.9      131       E    5
## 19622       143      -36.0      132       E    5
```

```
str(finalDF)
```

```
## 'data.frame':    19622 obs. of  11 variables:
##  $ roll_forearm : num  28.4 28.3 28.3 28.1 28 27.9 27.9 27.8 27.7 27.7 ...
##  $ pitch_forearm: num  -63.9 -63.9 -63.9 -63.9 -63.9 -63.9 -63.9 -63.8 -63
.8 -63.8 ...
##  $ yaw_forearm  : num  -153 -153 -152 -152 -152 -152 -152 -152 -152 -152 .
..
##  $ 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 .
..
##  $ 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 ...
##  $ classe       : chr  "A" "A" "A" "A" ...
##  $ Type         : num  1 1 1 1 1 1 1 1 1 1 ...
```

```
summary(finalDF)
```

```
##   roll_forearm       pitch_forearm     yaw_forearm         roll_arm
##  Min.   :-180.0000   Min.   :-72.50   Min.   :-180.00   Min.   :-180.00
##  1st Qu.:  -0.7375   1st Qu.:  0.00   1st Qu.: -68.60   1st Qu.: -31.77
##  Median :  21.7000   Median :  9.24   Median :   0.00   Median :   0.00
##  Mean   :  33.8265   Mean   : 10.71   Mean   :  19.21   Mean   :  17.83
##  3rd Qu.: 140.0000   3rd Qu.: 28.40   3rd Qu.: 110.00   3rd Qu.:  77.30
##  Max.   : 180.0000   Max.   : 89.80   Max.   : 180.00   Max.   : 180.00
##    pitch_arm          yaw_arm            roll_belt        pitch_belt
##  Min.   :-88.800   Min.   :-180.0000   Min.   :-28.90   Min.   :-55.8000
##  1st Qu.:-25.900   1st Qu.: -43.1000   1st Qu.:  1.10   1st Qu.:  1.7600
##  Median :  0.000   Median :   0.0000   Median :113.00   Median :  5.2800
##  Mean   : -4.612   Mean   :  -0.6188   Mean   : 64.41   Mean   :  0.3053
##  3rd Qu.: 11.200   3rd Qu.:  45.8750   3rd Qu.:123.00   3rd Qu.: 14.9000
##  Max.   : 88.500   Max.   : 180.0000   Max.   :162.00   Max.   : 60.3000
##     yaw_belt          classe               Type
##  Min.   :-180.00   Length:19622       Min.   :1.000
##  1st Qu.: -88.30   Class :character   1st Qu.:1.000
##  Median : -13.00   Mode  :character   Median :3.000
##  Mean   : -11.21                      Mean   :2.769
##  3rd Qu.:  12.90                      3rd Qu.:4.000
##  Max.   : 179.00                      Max.   :5.000
```

```
tempDF2 <- finalDF[,1:10]
Type <- finalDF['Type']
#tdf <- data.frame({"Type"=typeLst})
#tempDF2 <- cbind(tempDF2, Type)

head(tempDF2)
```

```
##   roll_forearm pitch_forearm yaw_forearm roll_arm pitch_arm yaw_arm
## 1         28.4          -63.9        -153     -128      22.5    -161
## 2         28.3          -63.9        -153     -128      22.5    -161
## 3         28.3          -63.9        -152     -128      22.5    -161
## 4         28.1          -63.9        -152     -128      22.1    -161
## 5         28.0          -63.9        -152     -128      22.1    -161
## 6         27.9          -63.9        -152     -128      22.0    -161
##   roll_belt pitch_belt yaw_belt classe
## 1      1.41       8.07    -94.4      A
## 2      1.41       8.07    -94.4      A
## 3      1.42       8.07    -94.4      A
## 4      1.48       8.05    -94.4      A
## 5      1.48       8.07    -94.4      A
## 6      1.45       8.06    -94.4      A
```

```
correlMtrx <- cor(tempDF2[,1:9])
highlyCorrel <- findCorrelation(correlMtrx, cutoff=0.5)
print(highlyCorrel)
```

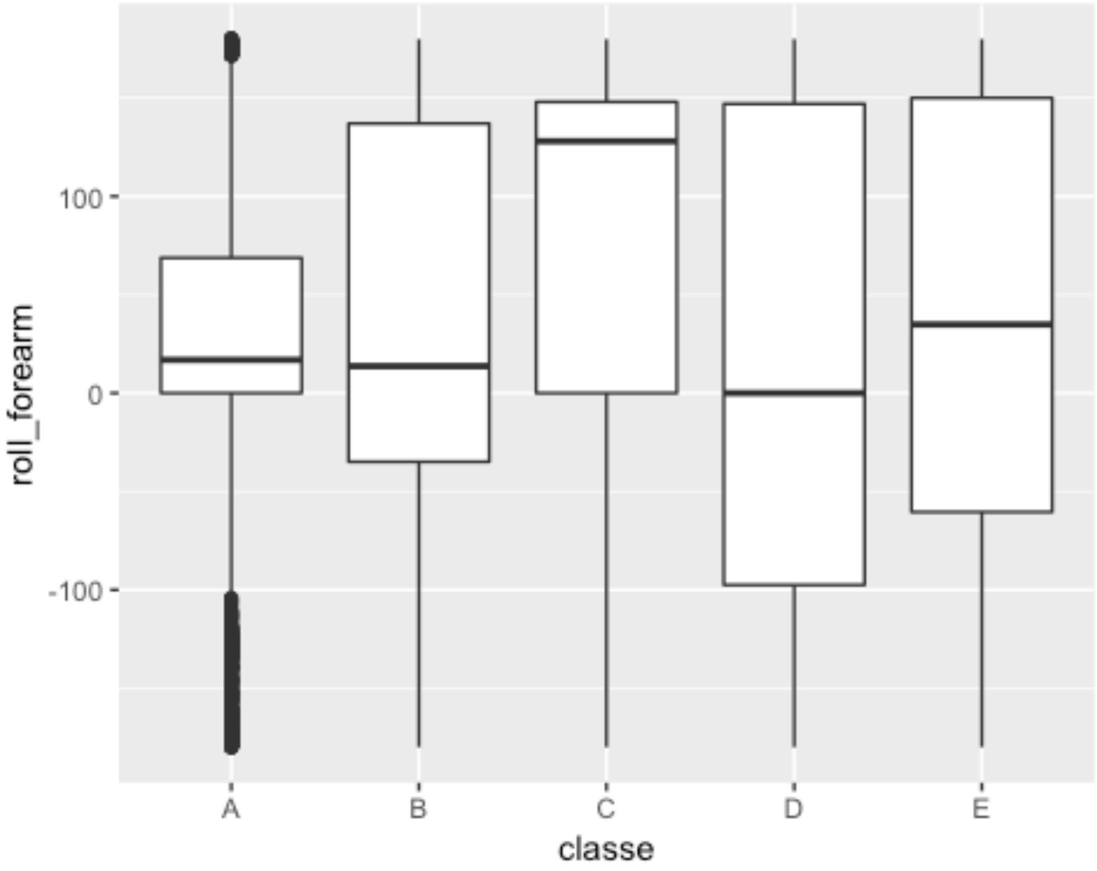
```
## [1] 9
```

```
#data(PimaIndiansDiabetes)
#head(PimaIndiansDiabetes)


#control <- trainControl(method = "repeatedcv",number = 10,repeats=3)
#model <- train(classe~.,data=tempDF2,method="lvq",preProcess="scale",trContr
ol=control)
#importance <- varImp(model,scale=FALSE)
#print(importance)
#plot(importance)

################################ Exploring the Data #########################
#############

plotDF <- finalDF
plotDF$classe <- as.factor(plotDF$classe)
str(plotDF)
```

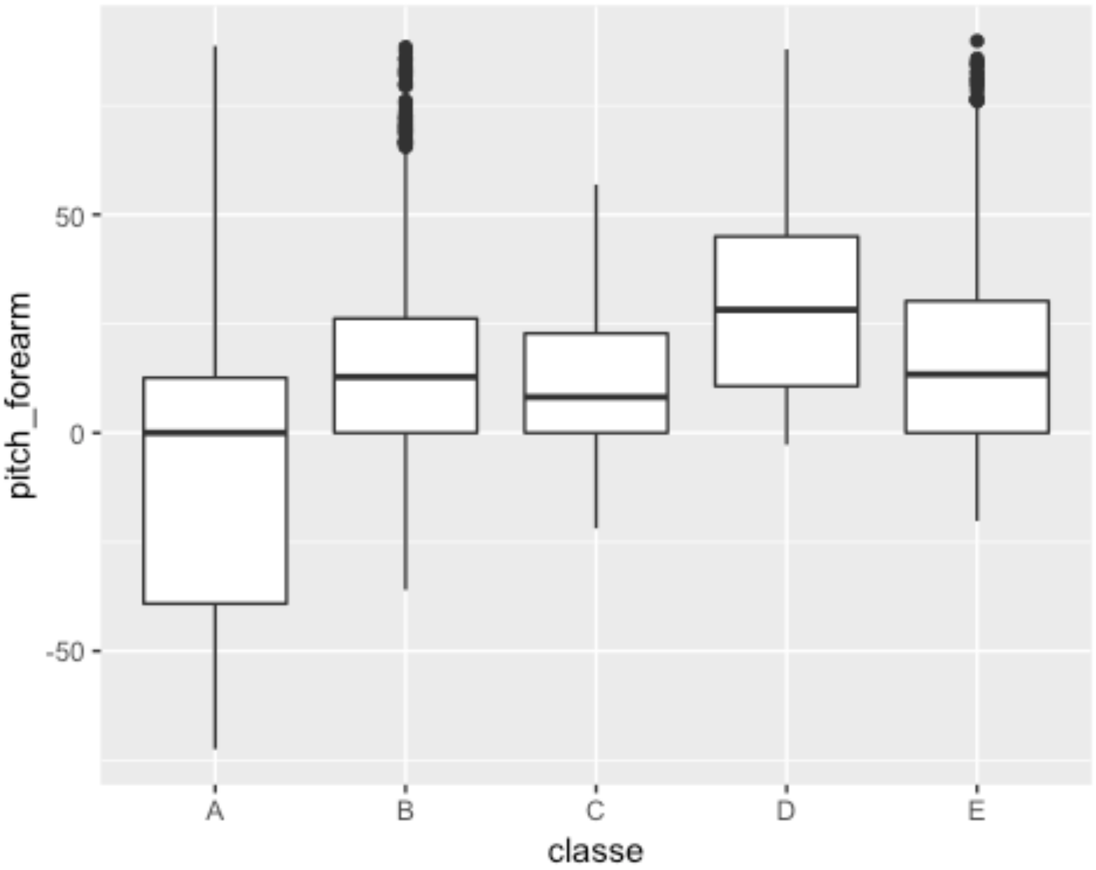
```
## 'data.frame':    19622 obs. of  11 variables:
##  $ roll_forearm : num  28.4 28.3 28.3 28.1 28 27.9 27.9 27.8 27.7 27.7 ...
##  $ pitch_forearm: num  -63.9 -63.9 -63.9 -63.9 -63.9 -63.9 -63.9 -63.8 -63
.8 -63.8 ...
```

```
##  $ yaw_forearm  : num  -153 -153 -152 -152 -152 -152 -152 -152 -152 -152 .
..
##  $ 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 .
..
##  $ 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 ...
##  $ classe       : Factor w/ 5 levels "A","B","C","D",..: 1 1 1 1 1 1 1 1 1
1 ...
##  $ Type         : num  1 1 1 1 1 1 1 1 1 1 ...
```

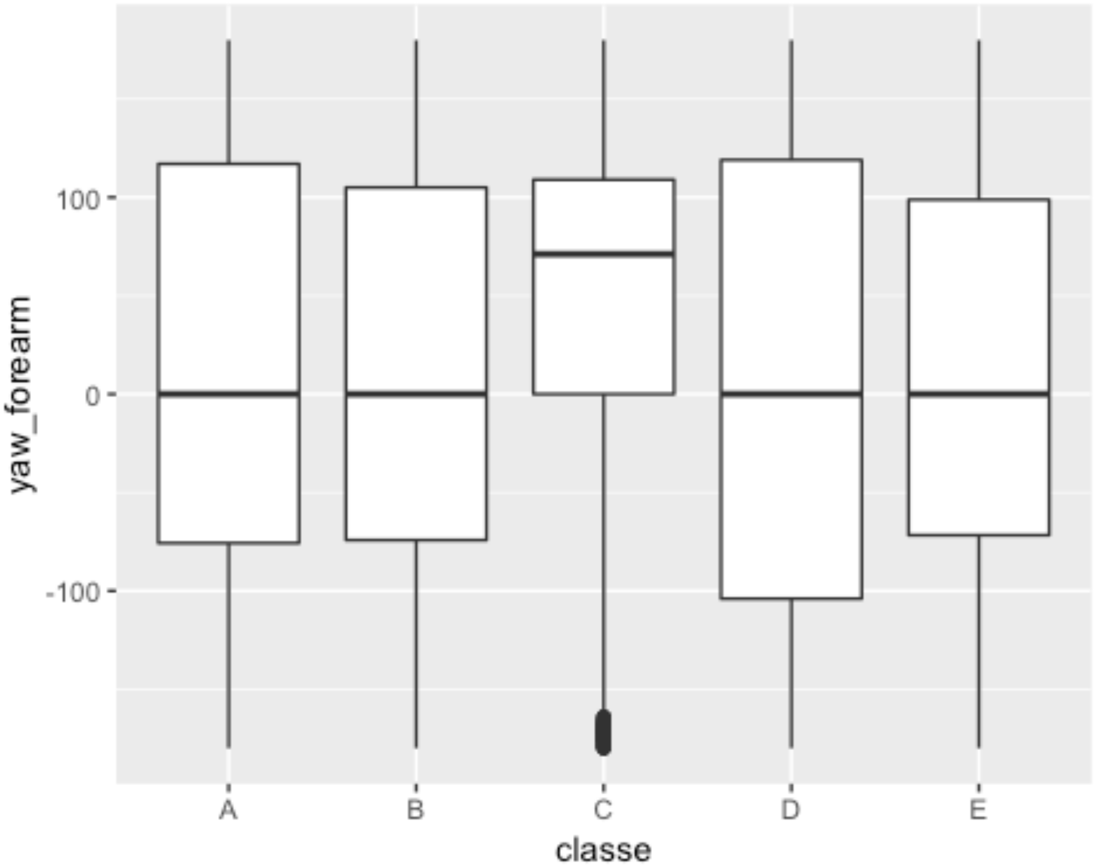
```
#print(names(plotDF))

bxpRFA <- ggplot(plotDF,aes(x=classe,y=roll_forearm)) + geom_boxplot()
plot(bxpRFA)
```



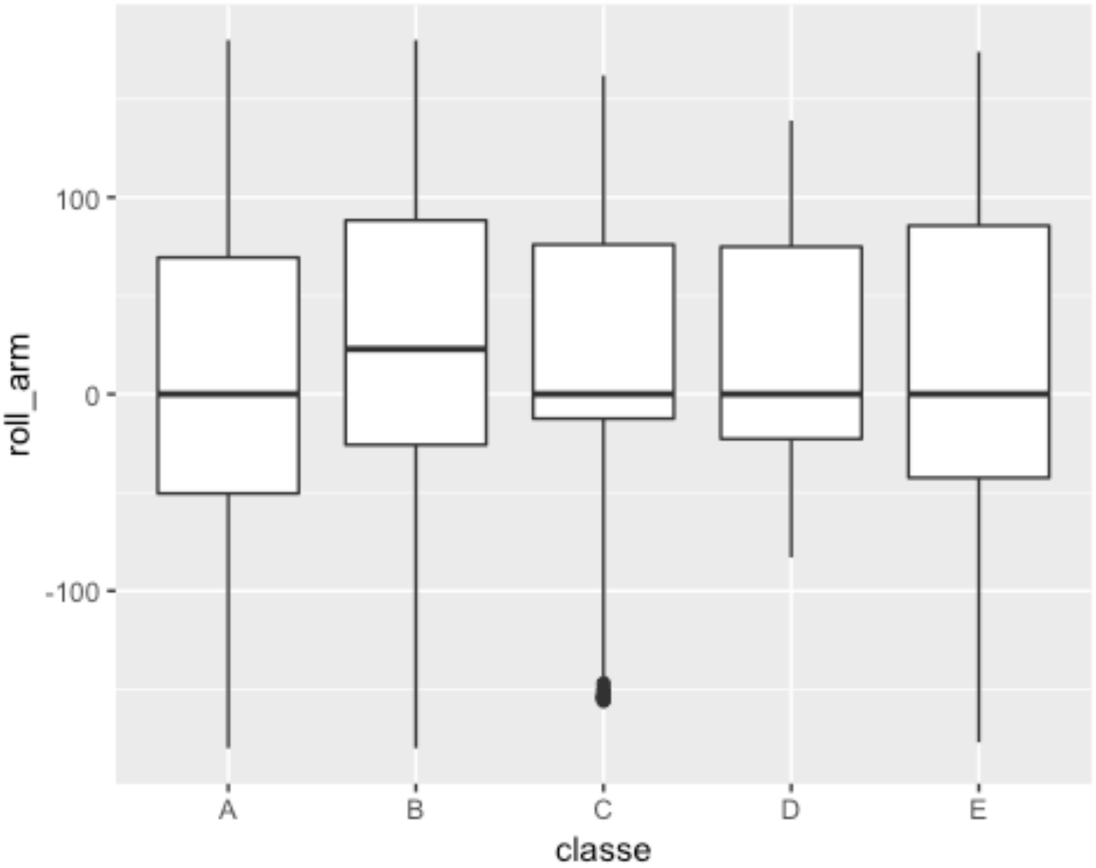
```
bxpPFA <- ggplot(plotDF,aes(x=classe,y=pitch_forearm)) + geom_boxplot()
plot(bxpPFA)
```



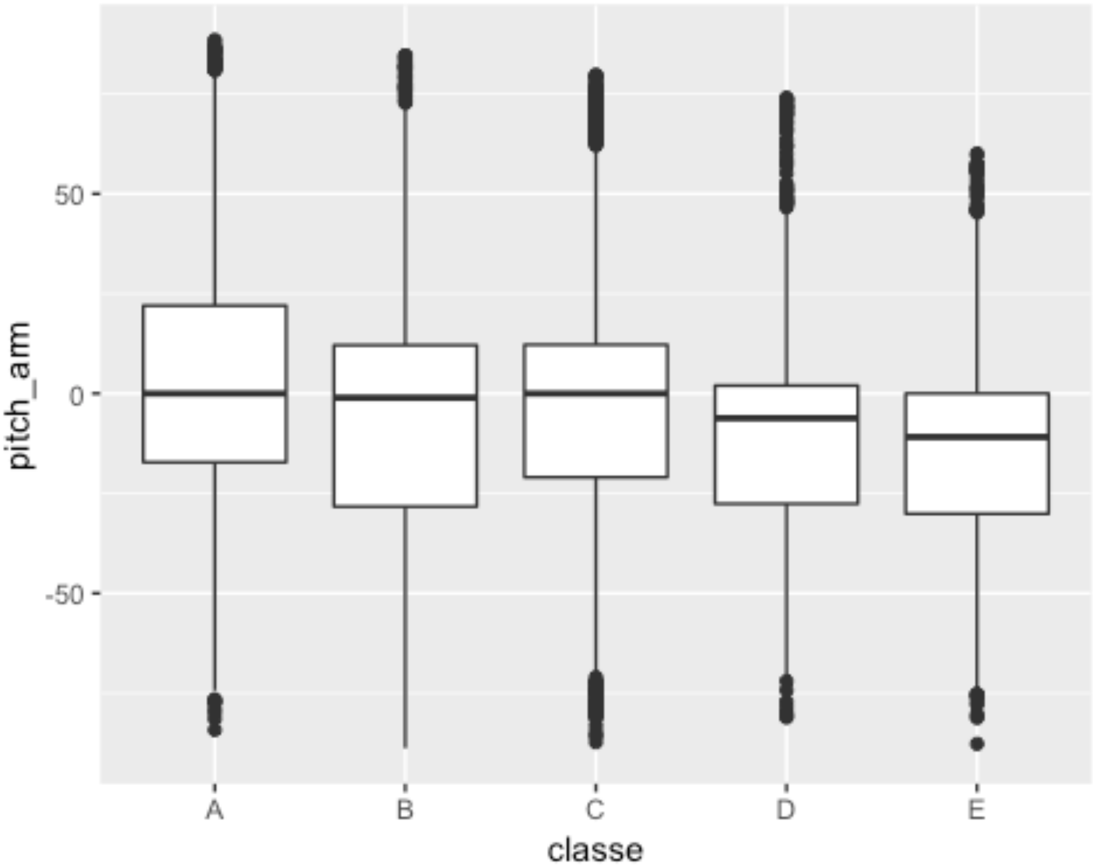
```
bxpYFA <- ggplot(plotDF,aes(x=classe,y=yaw_forearm)) + geom_boxplot()
plot(bxpYFA)
```



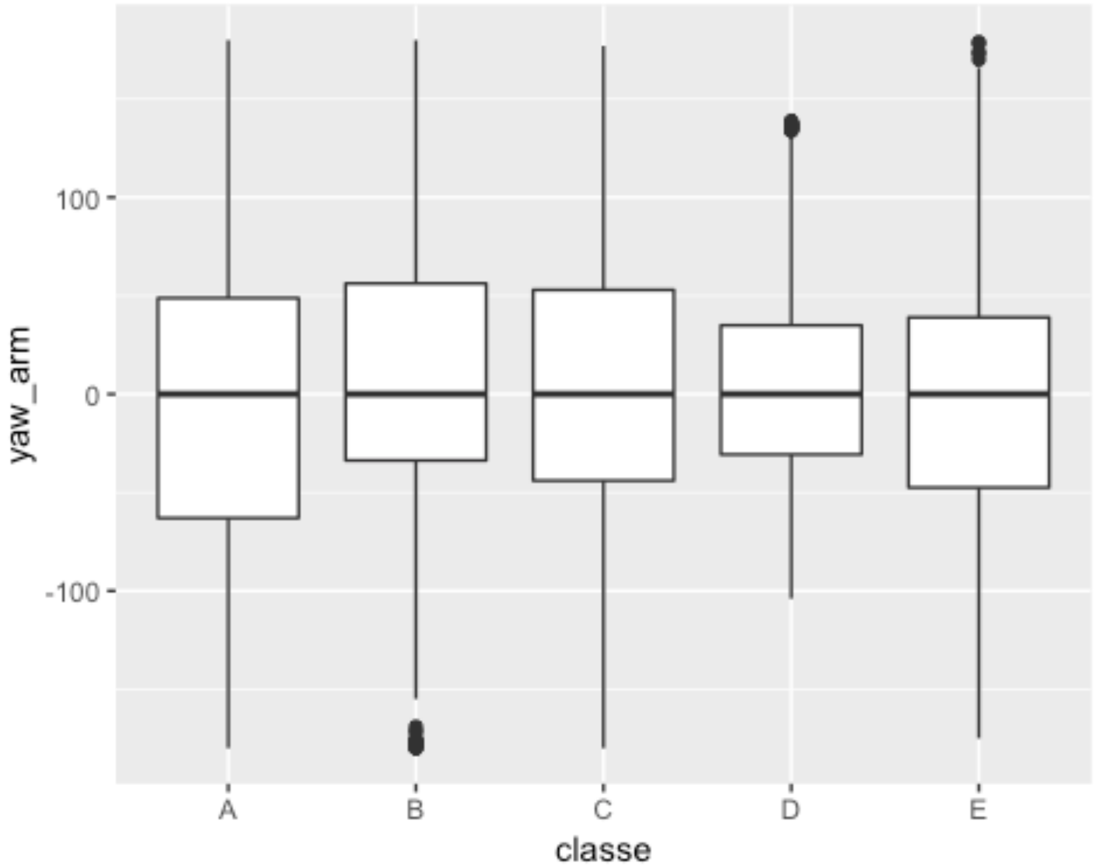
```
bxpRA <- ggplot(plotDF,aes(x=classe,y=roll_arm)) + geom_boxplot()
plot(bxpRA)
```



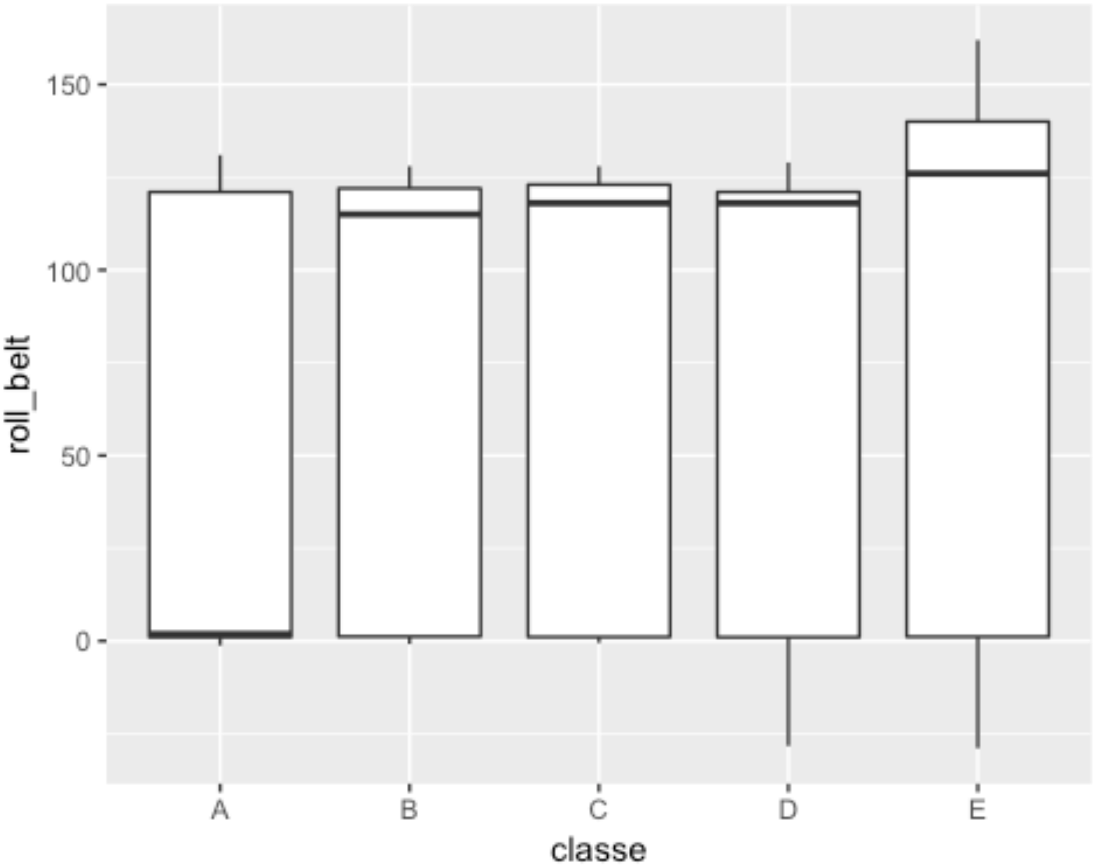
```
bxpPA <- ggplot(plotDF,aes(x=classe,y=pitch_arm)) + geom_boxplot()
plot(bxpPA)
```



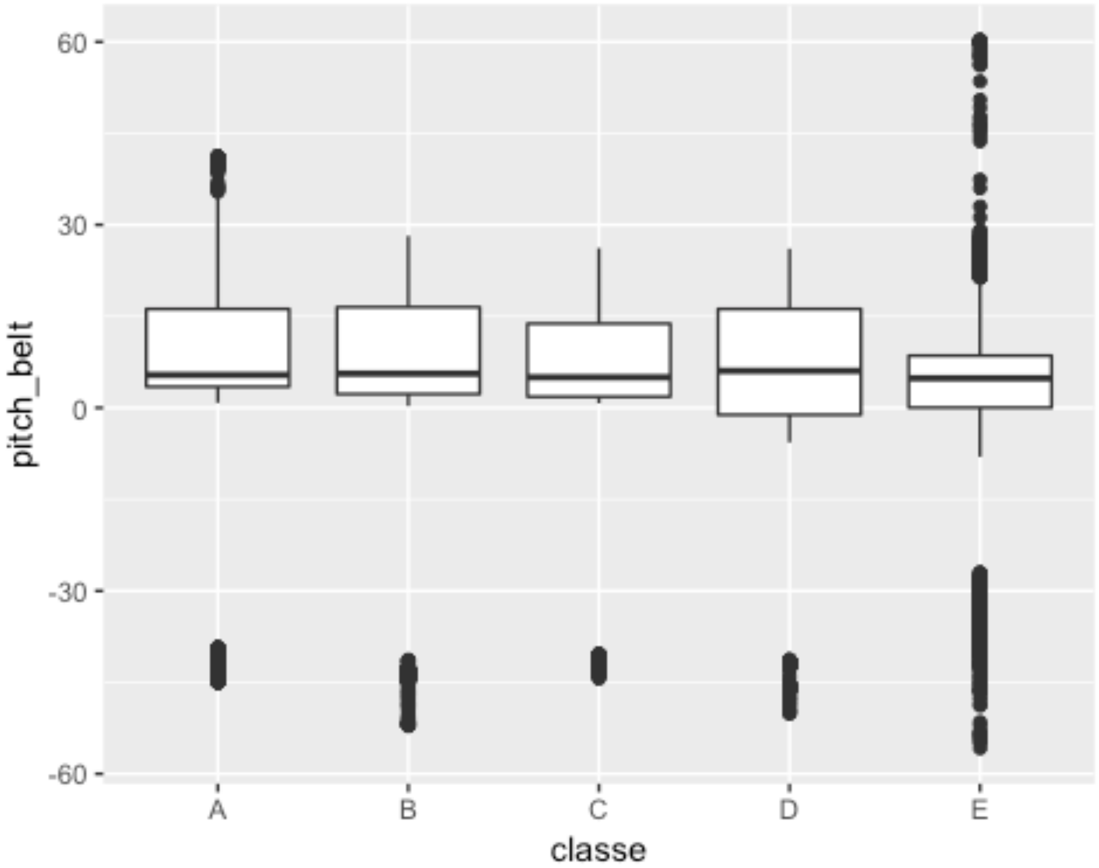
```
bxpYA <- ggplot(plotDF,aes(x=classe,y=yaw_arm)) + geom_boxplot()
plot(bxpYA)
```



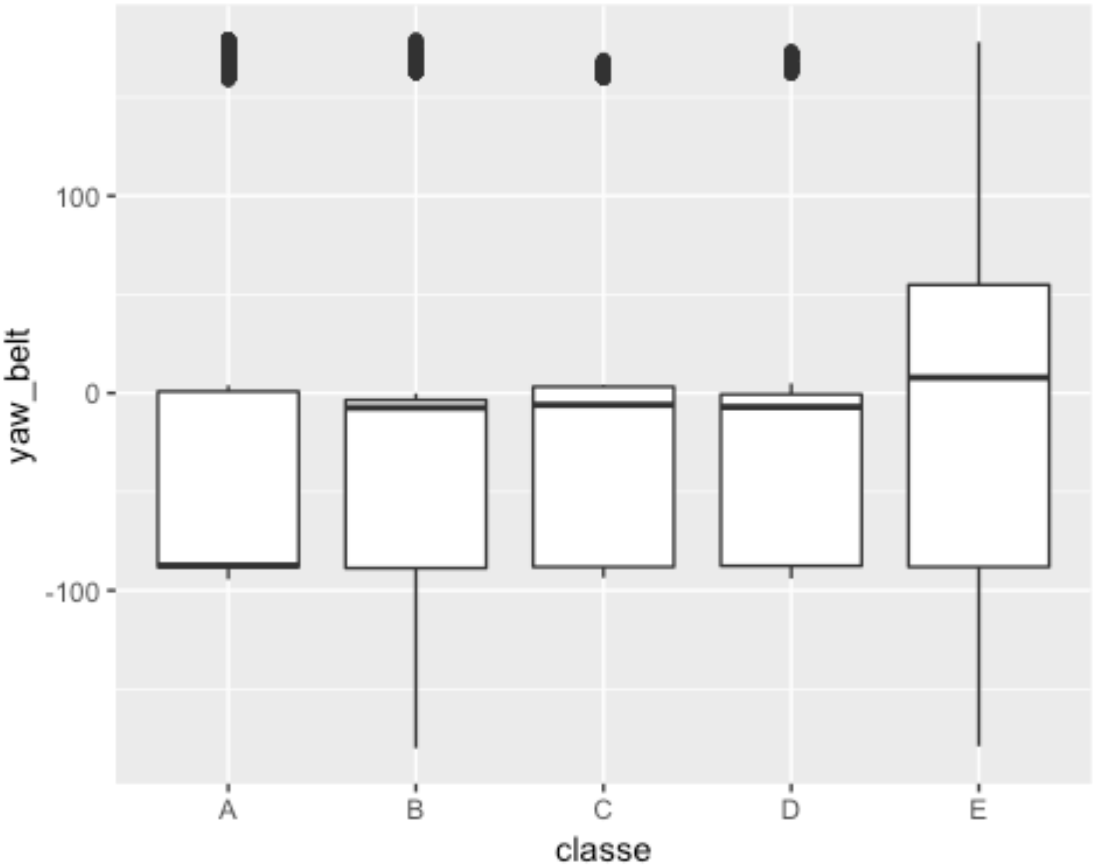
```
bxpRB <- ggplot(plotDF,aes(x=classe,y=roll_belt)) + geom_boxplot()
plot(bxpRB)
```



```
bxpPB <- ggplot(plotDF,aes(x=classe,y=pitch_belt)) + geom_boxplot()
plot(bxpPB)
```



```
bxpYB <- ggplot(plotDF,aes(x=classe,y=yaw_belt)) + geom_boxplot()
plot(bxpYB)
```



```
###################### K cross over validation and qda and lda analysis



library(MASS)

## Warning: package 'MASS' was built under R version 3.5.2

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select

tempDF2$classe=factor(tempDF2$classe)
str(tempDF2)

## 'data.frame':    19622 obs. of  10 variables:
##  $ roll_forearm : num  28.4 28.3 28.3 28.1 28 27.9 27.9 27.8 27.7 27.7 ...
##  $ pitch_forearm: num  -63.9 -63.9 -63.9 -63.9 -63.9 -63.9 -63.9 -63.8 -63
.8 -63.8 ...
##  $ yaw_forearm  : num  -153 -153 -152 -152 -152 -152 -152 -152 -152 -152 .
..
```

```
##  $ 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 .
..
##  $ 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 ...
##  $ classe       : Factor w/ 5 levels "A","B","C","D",..: 1 1 1 1 1 1 1 1 1
1 ...
```

```
K <- 20
folds <- cut(seq(1,nrow(tempDF2)),breaks = K,labels=FALSE)
head(folds)
```

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

```
set.seed(1)
cv.qda <- sapply(1:K, FUN = function(i) {
  testID <- which(folds == i, arr.ind = TRUE)
  test <- tempDF2[testID,]
  train <- tempDF2[-testID,]
  qdaf <- qda(classe~.,data=train)
  qda.pred <- predict(qdaf, test)
  cv.est.qda <- mean(qda.pred$class != test$classe)
  return(cv.est.qda)
})
print(cv.qda)
```

```
##  [1] 0.6568228 0.6065240 0.2548420 0.4740061 0.6870540 0.5932722 1.0000000
##  [8] 0.9418960 1.0000000 0.9724771 0.7930683 0.9714577 0.7431193 0.6309888
## [15] 0.7339450 0.6697248 0.6095821 0.9633028 0.8623853 0.7474542
```

```
print(mean(cv.qda))
```

```
## [1] 0.7455961
```

```
K <- 20
folds <- cut(seq(1,nrow(tempDF2)),breaks = K,labels=FALSE)
head(folds)
```

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

```
set.seed(1)
cv.lda <- sapply(1:K, FUN = function(i) {
  testID <- which(folds == i, arr.ind = TRUE)
  test <- tempDF2[testID,]
  train <- tempDF2[-testID,]
  ldaf <- lda(classe~.,data=train)
  lda.pred <- predict(ldaf, test)
  cv.est.lda <- mean(lda.pred$class != test$classe)
  return(cv.est.lda)
})
print(cv.lda)
```

```
##  [1] 0.5488798 0.7237513 0.2721713 0.4485219 0.7543323 0.6075433 1.0000000
##  [8] 1.0000000 1.0000000 0.9714577 1.0000000 1.0000000 0.9653415 0.8695209
## [15] 0.9785933 0.9653415 0.7747197 0.9633028 0.9989806 0.9725051
```

```
print(mean(cv.lda))
```

```
## [1] 0.8407481
```

```
str(finalDF)
```

```
## 'data.frame':    19622 obs. of  11 variables:
##  $ roll_forearm : num  28.4 28.3 28.3 28.1 28 27.9 27.9 27.8 27.7 27.7 ...
##  $ pitch_forearm: num  -63.9 -63.9 -63.9 -63.9 -63.9 -63.9 -63.9 -63.8 -63
.8 -63.8 ...
##  $ yaw_forearm  : num  -153 -153 -152 -152 -152 -152 -152 -152 -152 -152 .
..
##  $ 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 .
..
##  $ 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 ...
##  $ classe       : chr  "A" "A" "A" "A" ...
##  $ Type         : num  1 1 1 1 1 1 1 1 1 1 ...
```

```
names(finalDF)
```

```
##  [1] "roll_forearm"  "pitch_forearm" "yaw_forearm"   "roll_arm"
##  [5] "pitch_arm"     "yaw_arm"       "roll_belt"     "pitch_belt"
##  [9] "yaw_belt"      "classe"        "Type"
```

```
################################ Random Forest ##############################
###3


DF51 <- finalDF[,1:9]
DF52 <- finalDF[,10]
names(DF51)
```

```
## [1] "roll_forearm"  "pitch_forearm" "yaw_forearm"   "roll_arm"
## [5] "pitch_arm"     "yaw_arm"       "roll_belt"     "pitch_belt"
## [9] "yaw_belt"
```

```
names(DF52)
```

```
## NULL
```

```
rndforestDF <- cbind(DF51,Type)
rndforestDF$Type <- as.factor(rndforestDF$Type)
names(rndforestDF)
```

```
##  [1] "roll_forearm"  "pitch_forearm" "yaw_forearm"   "roll_arm"
##  [5] "pitch_arm"     "yaw_arm"       "roll_belt"     "pitch_belt"
##  [9] "yaw_belt"      "Type"
```

```
str(rndforestDF)
```

```
## 'data.frame':    19622 obs. of  10 variables:
##  $ roll_forearm : num  28.4 28.3 28.3 28.1 28 27.9 27.9 27.8 27.7 27.7 ...
##  $ pitch_forearm: num  -63.9 -63.9 -63.9 -63.9 -63.9 -63.9 -63.9 -63.8 -63
.8 -63.8 ...
##  $ yaw_forearm  : num  -153 -153 -152 -152 -152 -152 -152 -152 -152 -152 .
..
##  $ 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 .
..
##  $ 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 ...
##  $ Type         : Factor w/ 5 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1
1 ...
```

```
table(rndforestDF$Tpe)
```

```
## < table of extent 0 >
```

```
set.seed(325)
ind <- sample(2,nrow(rndforestDF),replace = TRUE,prob = c(0.7, 0.3))
train <- rndforestDF[ind==1,]
test <- rndforestDF[ind==2,]
```

```
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
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```r
set.seed(757)
rf <- randomForest(Type~.,data=train)
print(rf)
```

```
##
## Call:
##  randomForest(formula = Type ~ ., data = train)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 1.28%
## Confusion matrix:
##      1    2    3    4    5 class.error
## 1 3893   10    1    0    0 0.002817623
## 2   16 2557   56    3    0 0.028495441
## 3    0   21 2370   15    4 0.016597510
## 4    0    1   11 2189    5 0.007706256
## 5    1   12    7   12 2503 0.012623274
```

```r
################################ Predict using the train set
attributes(rf)
```

```
## $names
##  [1] "call"            "type"            "predicted"
##  [4] "err.rate"        "confusion"       "votes"
##  [7] "oob.times"       "classes"         "importance"
## [10] "importanceSD"    "localImportance" "proximity"
## [13] "ntree"           "mtry"            "forest"
## [16] "y"               "test"            "inbag"
## [19] "terms"
##
## $class
## [1] "randomForest.formula" "randomForest"
```

```r
p1 <- predict(rf, train)
head(p1)
```

```
## 1 3 4 5 6 7
## 1 1 1 1 1 1
## Levels: 1 2 3 4 5
```

```r
head(train$Type)
```

```
## [1] 1 1 1 1 1 1
## Levels: 1 2 3 4 5
```

```r
confusionMatrix(p1,train$Type)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction    1    2    3    4    5
##          1 3904    0    0    0    0
##          2    0 2632    0    0    0
##          3    0    0 2410    0    0
##          4    0    0    0 2206    0
##          5    0    0    0    0 2535
## 
## Overall Statistics
## 
##                Accuracy : 1
##                  95% CI : (0.9997, 1)
##     No Information Rate : 0.2852
##     P-Value [Acc > NIR] : < 2.2e-16
## 
##                   Kappa : 1
## 
##  Mcnemar's Test P-Value : NA
## 
## Statistics by Class:
## 
##                      Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
## Sensitivity            1.0000   1.0000   1.0000   1.0000   1.0000
## Specificity            1.0000   1.0000   1.0000   1.0000   1.0000
## Pos Pred Value         1.0000   1.0000   1.0000   1.0000   1.0000
## Neg Pred Value         1.0000   1.0000   1.0000   1.0000   1.0000
## Prevalence             0.2852   0.1923   0.1761   0.1612   0.1852
## Detection Rate         0.2852   0.1923   0.1761   0.1612   0.1852
## Detection Prevalence   0.2852   0.1923   0.1761   0.1612   0.1852
## Balanced Accuracy      1.0000   1.0000   1.0000   1.0000   1.0000
```

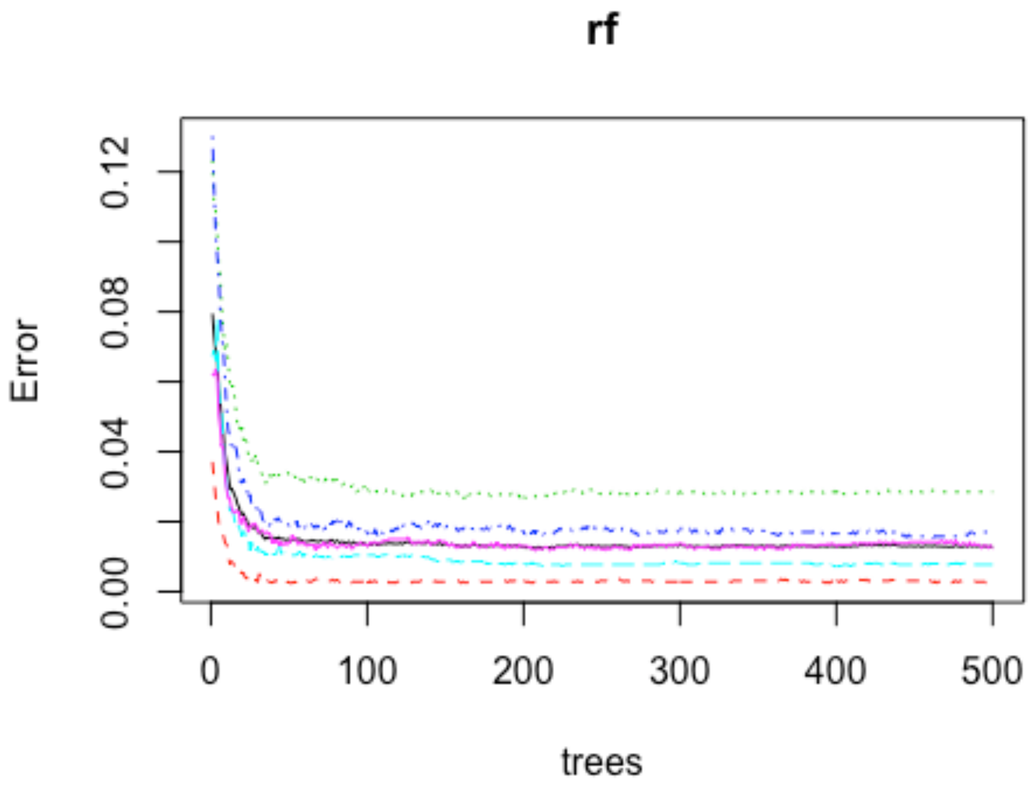
```r
############################## Predict using the test set

p2 <- predict(rf, test)
confusionMatrix(p2,test$Type)
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction    1    2    3    4    5
##          1 1669    7    0    0    0
##          2    7 1128   14    1    1
##          3    0   26  992    8    1
##          4    0    4    4 1000    5
##          5    0    0    2    1 1065
## 
## Overall Statistics
##                                           
##                Accuracy : 0.9864          
##                  95% CI : (0.9831, 0.9891)
##     No Information Rate : 0.2824          
##     P-Value [Acc > NIR] : < 2.2e-16       
##                                           
##                   Kappa : 0.9827          
##                                           
##  Mcnemar's Test P-Value : NA              
## 
## Statistics by Class:
## 
##                      Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
## Sensitivity            0.9958   0.9682   0.9802   0.9901   0.9935
## Specificity            0.9984   0.9952   0.9929   0.9974   0.9994
## Pos Pred Value         0.9958   0.9800   0.9659   0.9872   0.9972
## Neg Pred Value         0.9984   0.9923   0.9959   0.9980   0.9986
## Prevalence             0.2824   0.1963   0.1705   0.1702   0.1806
## Detection Rate         0.2812   0.1901   0.1671   0.1685   0.1794
## Detection Prevalence   0.2824   0.1939   0.1730   0.1707   0.1799
## Balanced Accuracy      0.9971   0.9817   0.9866   0.9937   0.9964
```

```
names(train)
```

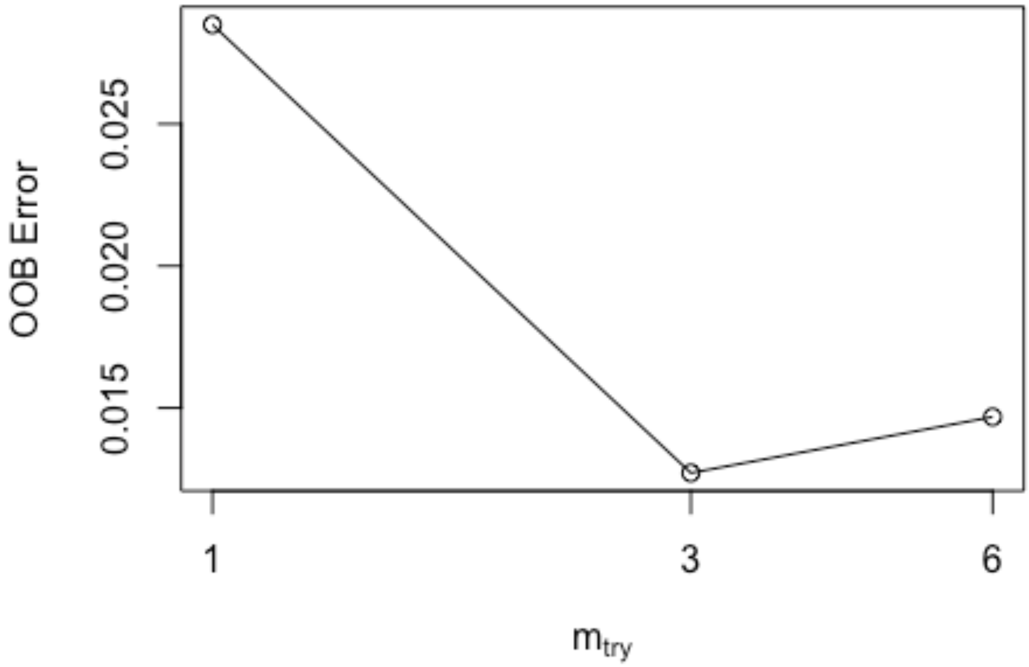
```
##  [1] "roll_forearm"  "pitch_forearm" "yaw_forearm"   "roll_arm"     
##  [5] "pitch_arm"     "yaw_arm"       "roll_belt"     "pitch_belt"   
##  [9] "yaw_belt"      "Type"
```

```
########################################### Tuning the Random Forest ########
###############3

plot(rf)
```



```
t <- tuneRF(train[,-10],train[,10],stepFactor = 0.5,plot = TRUE,ntreeTry = 250,trace=TRUE,improve=0.05)
```

```
## mtry = 3  OOB error = 1.27%
## Searching left ...
## mtry = 6     OOB error = 1.47%
## -0.1551724 0.05
## Searching right ...
## mtry = 1     OOB error = 2.85%
## -1.241379 0.05
```



```
############################################ Tuning using the train sret

rf2 <- randomForest(Type~.,data=train,ntree = 250,mtry = 3, importance = TRUE
, proximity = TRUE)
print(rf2)

##
## Call:
##  randomForest(formula = Type ~ ., data = train, ntree = 250, mtry = 3,
importance = TRUE, proximity = TRUE)
##                Type of random forest: classification
##                      Number of trees: 250
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 1.28%
## Confusion matrix:
##      1    2    3    4    5 class.error
## 1 3894   10    0    0    0 0.002561475
## 2   15 2555   58    4    0 0.029255319
## 3    0   24 2368   14    4 0.017427386
## 4    0    1    9 2191    5 0.006799637
## 5    0   11    6   14 2504 0.012228797
```

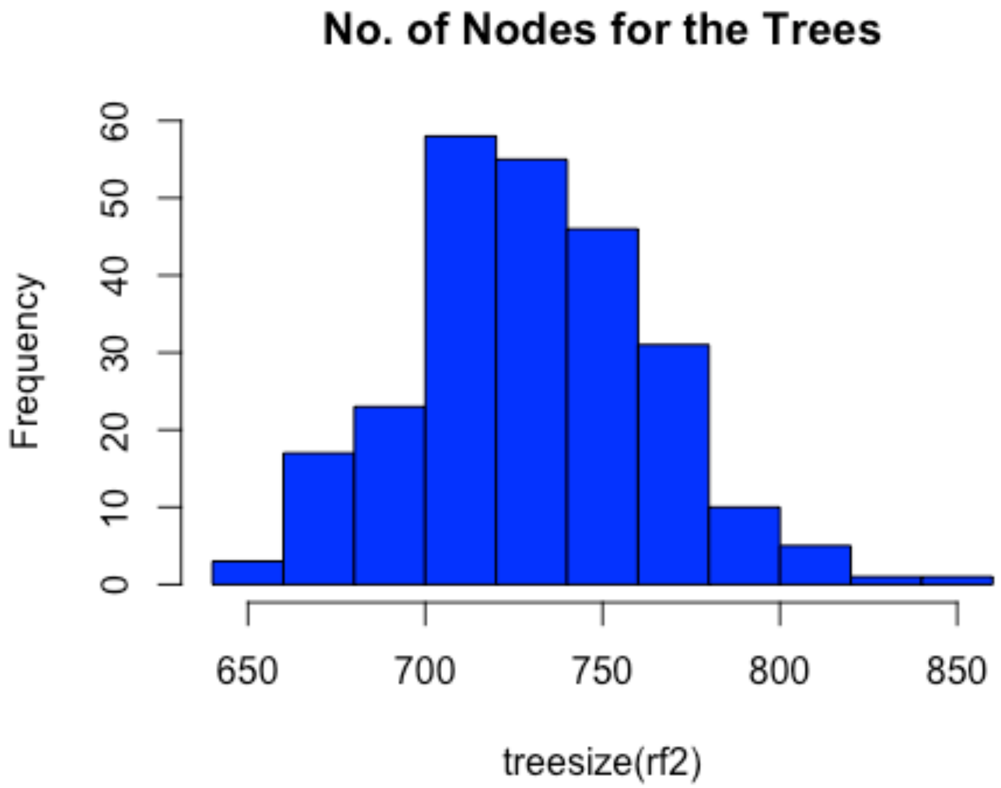
```
p21 <- predict(rf2,train)
confusionMatrix(p21, train$Type)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    1    2    3    4    5
##          1 3904    0    0    0    0
##          2    0 2632    0    0    0
##          3    0    0 2410    0    0
##          4    0    0    0 2206    0
##          5    0    0    0    0 2535
##
## Overall Statistics
##
##                Accuracy : 1
##                  95% CI : (0.9997, 1)
##     No Information Rate : 0.2852
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
## Sensitivity            1.0000   1.0000   1.0000   1.0000   1.0000
## Specificity            1.0000   1.0000   1.0000   1.0000   1.0000
## Pos Pred Value         1.0000   1.0000   1.0000   1.0000   1.0000
## Neg Pred Value         1.0000   1.0000   1.0000   1.0000   1.0000
## Prevalence             0.2852   0.1923   0.1761   0.1612   0.1852
## Detection Rate         0.2852   0.1923   0.1761   0.1612   0.1852
## Detection Prevalence   0.2852   0.1923   0.1761   0.1612   0.1852
## Balanced Accuracy      1.0000   1.0000   1.0000   1.0000   1.0000
```

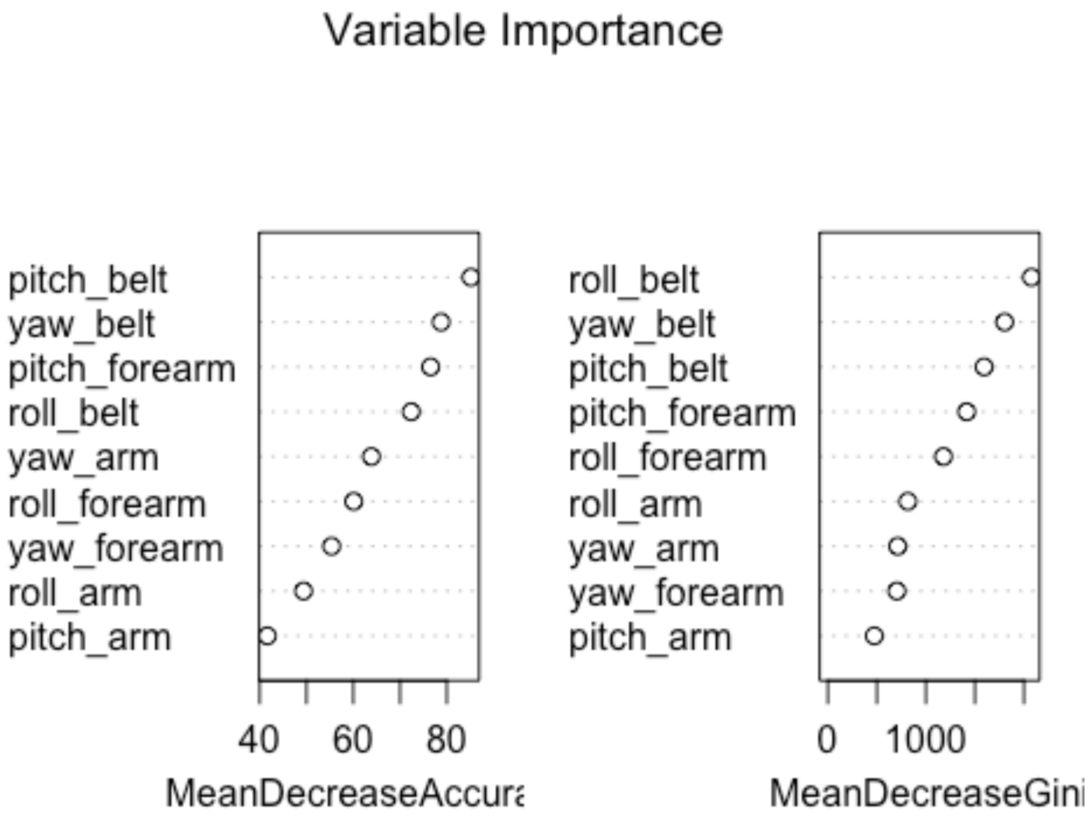
```
########################################## Tuning using the predict set

p22 <- predict(rf2,test)
confusionMatrix(p22,test$Type)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    1    2    3    4    5
##          1 1668    7    0    0    0
##          2    8 1128   12    1    1
##          3    0   26  995    9    1
##          4    0    4    5  999    5
##          5    0    0    0    1 1065
```

```
##
## Overall Statistics
##
##                Accuracy : 0.9865
##                  95% CI : (0.9833, 0.9893)
##     No Information Rate : 0.2824
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.983
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
## Sensitivity            0.9952   0.9682   0.9832   0.9891   0.9935
## Specificity            0.9984   0.9954   0.9927   0.9972   0.9998
## Pos Pred Value         0.9958   0.9809   0.9651   0.9862   0.9991
## Neg Pred Value         0.9981   0.9923   0.9965   0.9978   0.9986
## Prevalence             0.2824   0.1963   0.1705   0.1702   0.1806
## Detection Rate         0.2810   0.1901   0.1676   0.1683   0.1794
## Detection Prevalence   0.2822   0.1938   0.1737   0.1707   0.1796
## Balanced Accuracy      0.9968   0.9818   0.9879   0.9931   0.9966

hist(treesize(rf2),main = "No. of Nodes for the Trees", col="blue")
```

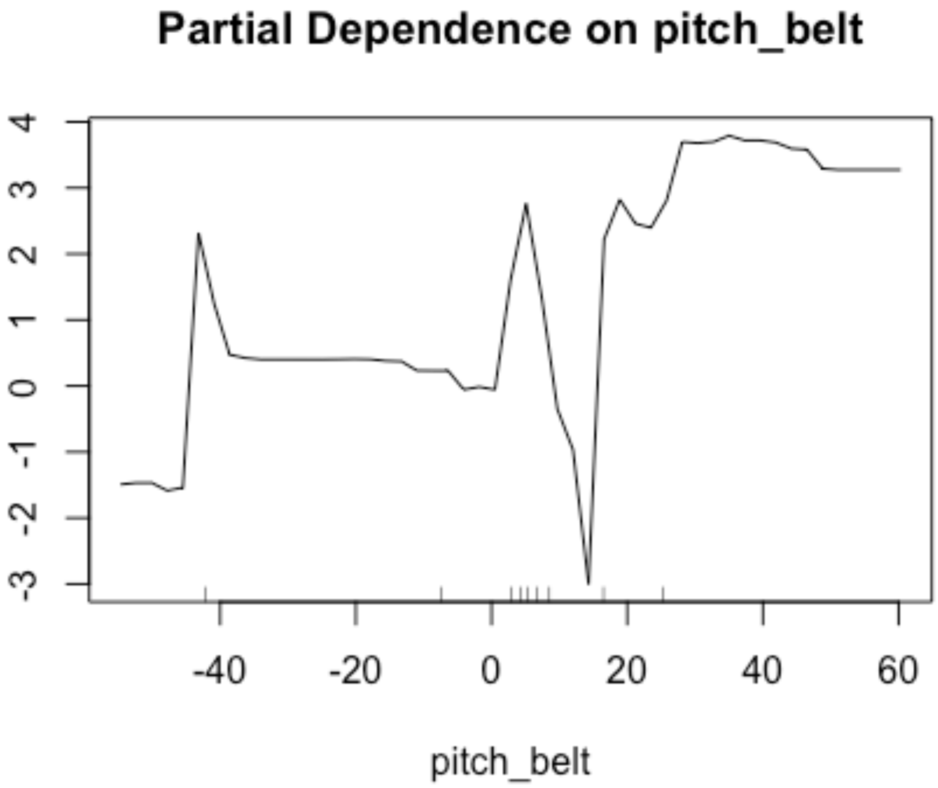


```r
varImpPlot(rf2, main = "Variable Importance")
```

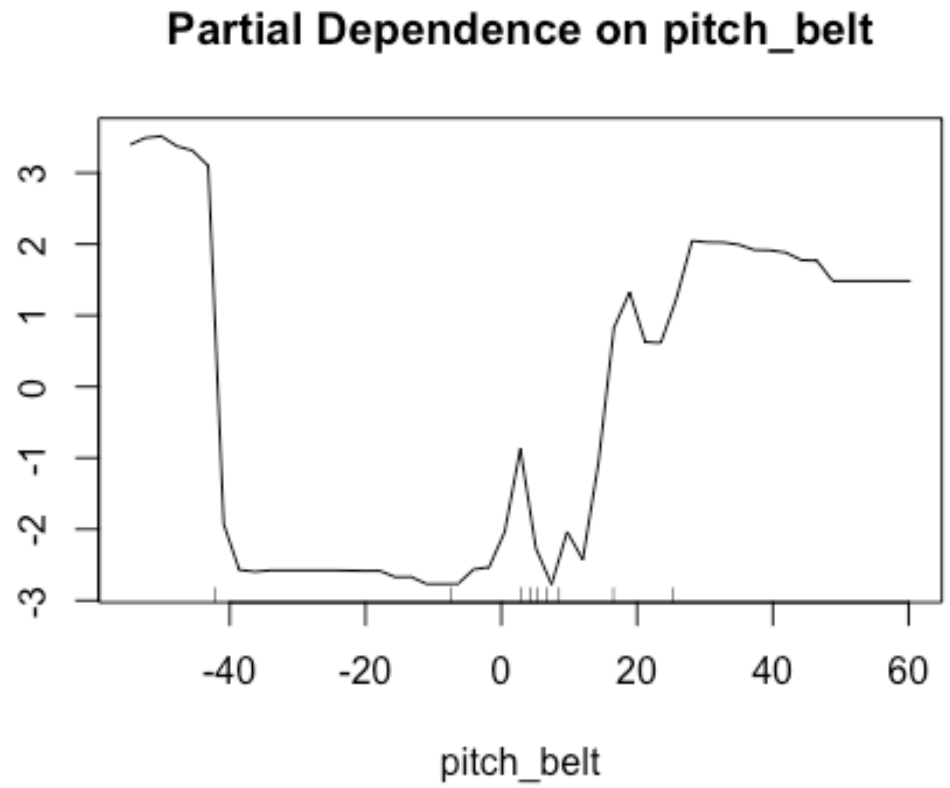


```
importance(rf2)

##                      1        2        3        4        5
## roll_forearm  42.85669 51.26740 76.88669 50.68116 49.68210
## pitch_forearm 43.50019 57.18189 62.42005 85.07532 57.29369
## yaw_forearm   34.63824 41.26679 50.80003 42.62494 45.06413
## roll_arm      26.39943 42.96575 39.21809 43.95813 31.43289
## pitch_arm     21.72842 36.78679 39.61017 34.61318 33.93804
## yaw_arm       43.65869 50.29782 51.96163 45.27550 31.45772
## roll_belt     45.45110 61.69194 57.25247 49.89291 58.52817
## pitch_belt    50.46518 76.36023 68.35774 60.21233 42.53164
## yaw_belt      66.17040 65.77808 58.22214 68.60817 33.66951
##               MeanDecreaseAccuracy MeanDecreaseGini
## roll_forearm              60.09964        1179.6245
## pitch_forearm             76.53942        1416.2912
## yaw_forearm               55.37273         704.0048
## roll_arm                  49.37771         816.1285
## pitch_arm                 41.62751         474.1469
## yaw_arm                   63.88046         713.1944
## roll_belt                 72.39013        2077.0343
## pitch_belt                85.16214        1594.0007
## yaw_belt                  78.77275        1802.9883
```

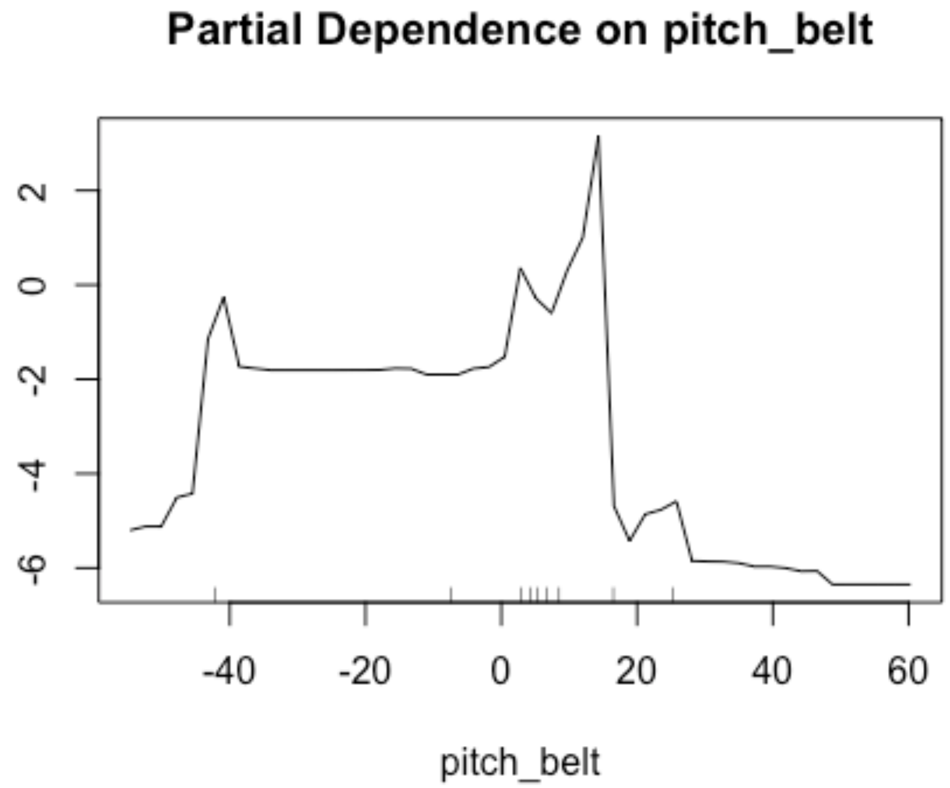
```r
varUsed(rf2)

## [1] 19386 19848 18608 18456 16435 16639 21963 25299 25660

###########################3 How does pitch belt predict the various classe

############# classe = A means Type = 1
############# classe = B means Type = 2
############# classe = C means Type = 3
############# classe = D means Type = 4
############# classe = E means Type = 5

partialPlot(rf2, train, pitch_belt, "1")
```



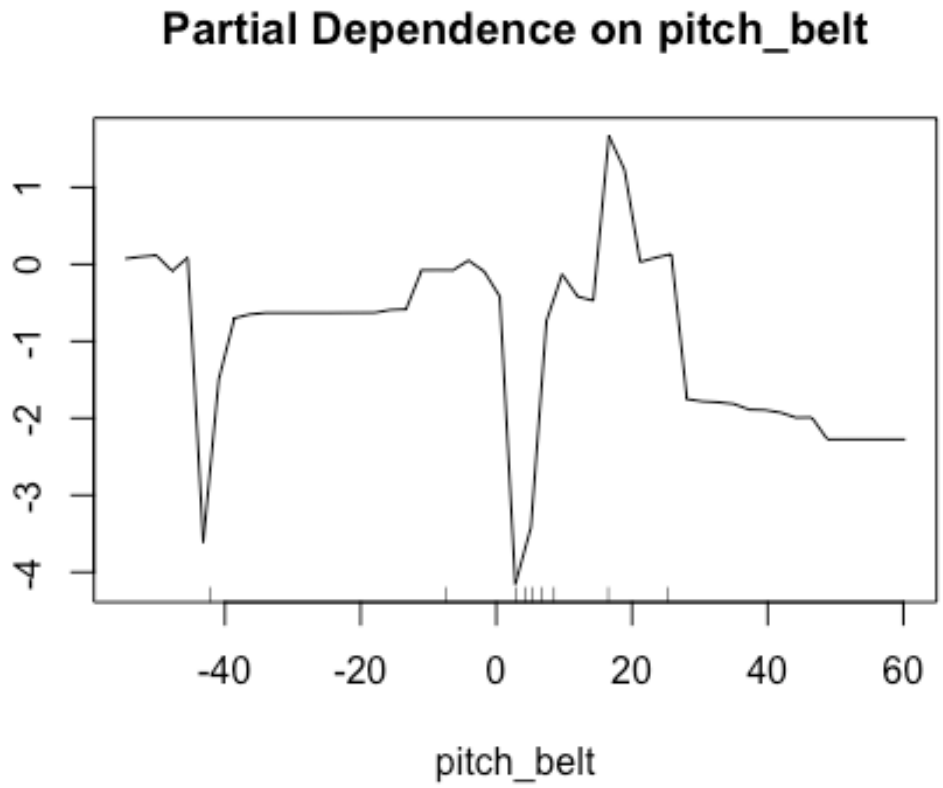
```r
partialPlot(rf2, train, pitch_belt, "2")
```



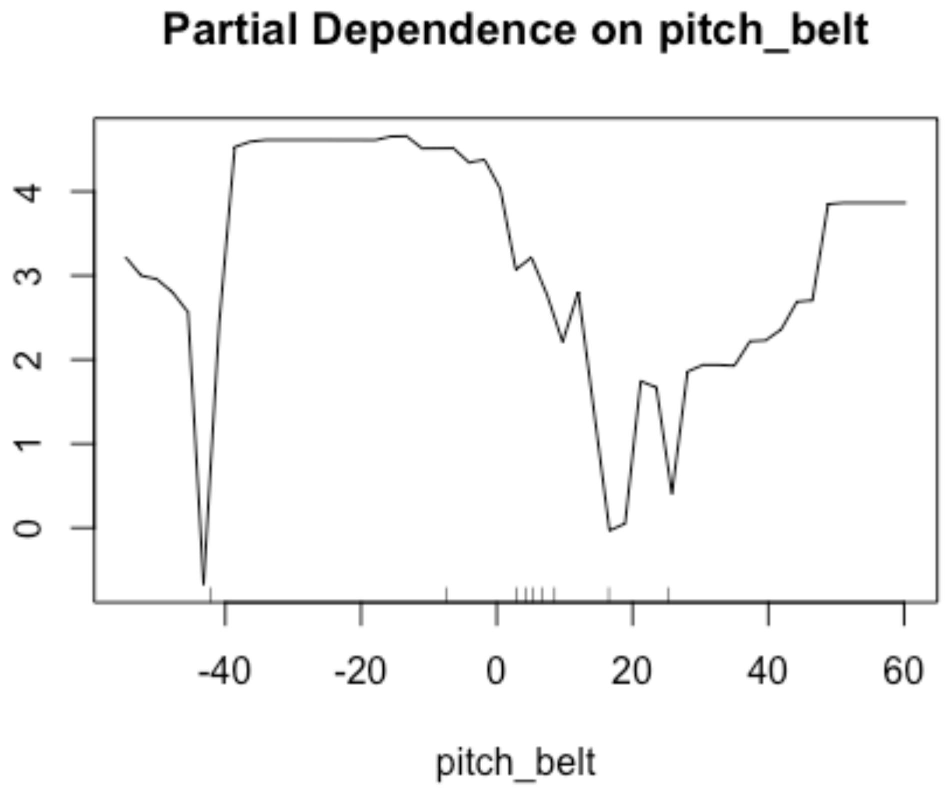
```r
partialPlot(rf2, train, pitch_belt, "3")
```



```r
partialPlot(rf2, train, pitch_belt, "4")
```



```r
partialPlot(rf2, train, pitch_belt, "5")
```



Including Plots

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.