Project: Practical Machine Learning

Anne Racel

April 18, 2019

Summary

The dataset provided with workout data. The purpose of this study is to use this data to predict what type of activity (classe) the participant is engaging in. There are 159 independent variables and one dependent variable provided. A quick ‘eyeballing’ of the data, however, shows many columns where the values are ‘NA’. It will be determined how important those values are to the predictive model and how they should be handled.

This is a classification problem, and we will try applying 3 or 4 different algorithms to the problem, in order to find the best solution. We will also try combining the results of the predictions to see if that assists us in improving the predictive possibilities.

Notes: I tried using median values to replace the ‘NAs’ in a number of columns. This was abandoned after the results proved to be poor. Those efforts have been removed from this writeup in order to stay within the limits given within the directions for this project.

```
exTrain <- read.csv('/home/anne/Downloads/pml-training.csv')
exTest <- read.csv('/home/anne/Downloads/pml-testing.csv')

head(exTrain)
```

```
##   X user_name raw_timestamp_part_1 raw_timestamp_part_2   cvtd_timestamp
## 1 1  carlitos           1323084231               788290 05/12/2011 11:23
## 2 2  carlitos           1323084231               808298 05/12/2011 11:23
## 3 3  carlitos           1323084231               820366 05/12/2011 11:23
## 4 4  carlitos           1323084232               120339 05/12/2011 11:23
## 5 5  carlitos           1323084232               196328 05/12/2011 11:23
## 6 6  carlitos           1323084232               304277 05/12/2011 11:23
##   new_window num_window roll_belt pitch_belt yaw_belt total_accel_belt
## 1         no         11      1.41       8.07    -94.4                3
## 2         no         11      1.41       8.07    -94.4                3
## 3         no         11      1.42       8.07    -94.4                3
## 4         no         12      1.48       8.05    -94.4                3
## 5         no         12      1.48       8.07    -94.4                3
## 6         no         12      1.45       8.06    -94.4                3
##   kurtosis_roll_belt kurtosis_picth_belt kurtosis_yaw_belt
## 1
## 2
## 3
## 4
## 5
## 6
##   skewness_roll_belt skewness_roll_belt.1 skewness_yaw_belt max_roll_belt
## 1                                                                      NA
## 2                                                                      NA
## 3                                                                      NA
## 4                                                                      NA
## 5                                                                      NA
## 6                                                                      NA
##   max_picth_belt max_yaw_belt min_roll_belt min_pitch_belt min_yaw_belt
## 1             NA                        NA             NA
## 2             NA                        NA             NA
## 3             NA                        NA             NA
## 4             NA                        NA             NA
## 5             NA                        NA             NA
## 6             NA                        NA             NA
##   amplitude_roll_belt amplitude_pitch_belt amplitude_yaw_belt
## 1                  NA                   NA
## 2                  NA                   NA
## 3                  NA                   NA
## 4                  NA                   NA
## 5                  NA                   NA
## 6                  NA                   NA
##   var_total_accel_belt avg_roll_belt stddev_roll_belt var_roll_belt
## 1                   NA            NA               NA            NA
## 2                   NA            NA               NA            NA
## 3                   NA            NA               NA            NA
## 4                   NA            NA               NA            NA
## 5                   NA            NA               NA            NA
## 6                   NA            NA               NA            NA
##   avg_pitch_belt stddev_pitch_belt var_pitch_belt avg_yaw_belt
## 1             NA                NA             NA           NA
## 2             NA                NA             NA           NA
## 3             NA                NA             NA           NA
## 4             NA                NA             NA           NA
## 5             NA                NA             NA           NA
## 6             NA                NA             NA           NA
##   stddev_yaw_belt var_yaw_belt gyros_belt_x gyros_belt_y gyros_belt_z
## 1              NA           NA         0.00         0.00        -0.02
## 2              NA           NA         0.02         0.00        -0.02
## 3              NA           NA         0.00         0.00        -0.02
## 4              NA           NA         0.02         0.00        -0.03
## 5              NA           NA         0.02         0.02        -0.02
## 6              NA           NA         0.02         0.00        -0.02
##   accel_belt_x accel_belt_y accel_belt_z magnet_belt_x magnet_belt_y
## 1          -21            4           22            -3           599
## 2          -22            4           22            -7           608
## 3          -20            5           23            -2           600
## 4          -22            3           21            -6           604
## 5          -21            2           24            -6           600
## 6          -21            4           21             0           603
##   magnet_belt_z roll_arm pitch_arm yaw_arm total_accel_arm var_accel_arm
## 1          -313     -128      22.5    -161              34            NA
## 2          -311     -128      22.5    -161              34            NA
## 3          -305     -128      22.5    -161              34            NA
## 4          -310     -128      22.1    -161              34            NA
## 5          -302     -128      22.1    -161              34            NA
## 6          -312     -128      22.0    -161              34            NA
##   avg_roll_arm stddev_roll_arm var_roll_arm avg_pitch_arm stddev_pitch_arm
## 1           NA              NA           NA            NA               NA
## 2           NA              NA           NA            NA               NA
## 3           NA              NA           NA            NA               NA
## 4           NA              NA           NA            NA               NA
## 5           NA              NA           NA            NA               NA
## 6           NA              NA           NA            NA               NA
##   var_pitch_arm avg_yaw_arm stddev_yaw_arm var_yaw_arm gyros_arm_x
## 1            NA          NA             NA          NA        0.00
## 2            NA          NA             NA          NA        0.02
## 3            NA          NA             NA          NA        0.02
## 4            NA          NA             NA          NA        0.02
## 5            NA          NA             NA          NA        0.00
## 6            NA          NA             NA          NA        0.02
##   gyros_arm_y gyros_arm_z accel_arm_x accel_arm_y accel_arm_z magnet_arm_x
## 1        0.00       -0.02        -288         109        -123         -368
## 2       -0.02       -0.02        -290         110        -125         -369
## 3       -0.02       -0.02        -289         110        -126         -368
## 4       -0.03        0.02        -289         111        -123         -372
## 5       -0.03        0.00        -289         111        -123         -374
## 6       -0.03        0.00        -289         111        -122         -369
##   magnet_arm_y magnet_arm_z kurtosis_roll_arm kurtosis_picth_arm
## 1          337          516
## 2          337          513
## 3          344          513
## 4          344          512
## 5          337          506
## 6          342          513
##   kurtosis_yaw_arm skewness_roll_arm skewness_pitch_arm skewness_yaw_arm
## 1
## 2
## 3
## 4
## 5
## 6
##   max_roll_arm max_picth_arm max_yaw_arm min_roll_arm min_pitch_arm
## 1           NA            NA          NA           NA            NA
## 2           NA            NA          NA           NA            NA
## 3           NA            NA          NA           NA            NA
## 4           NA            NA          NA           NA            NA
## 5           NA            NA          NA           NA            NA
## 6           NA            NA          NA           NA            NA
##   min_yaw_arm amplitude_roll_arm amplitude_pitch_arm amplitude_yaw_arm
## 1          NA                 NA                  NA                NA
## 2          NA                 NA                  NA                NA
## 3          NA                 NA                  NA                NA
## 4          NA                 NA                  NA                NA
## 5          NA                 NA                  NA                NA
## 6          NA                 NA                  NA                NA
##   roll_dumbbell pitch_dumbbell yaw_dumbbell kurtosis_roll_dumbbell
## 1      13.05217      -70.49400    -84.87394
## 2      13.13074      -70.63751    -84.71065
## 3      12.85075      -70.27812    -85.14078
## 4      13.43120      -70.39379    -84.87363
## 5      13.37872      -70.42856    -84.85306
## 6      13.38246      -70.81759    -84.46500
##   kurtosis_picth_dumbbell kurtosis_yaw_dumbbell skewness_roll_dumbbell
## 1
## 2
## 3
## 4
## 5
## 6
##   skewness_pitch_dumbbell skewness_yaw_dumbbell max_roll_dumbbell
## 1                                                              NA
## 2                                                              NA
## 3                                                              NA
## 4                                                              NA
## 5                                                              NA
## 6                                                              NA
##   max_picth_dumbbell max_yaw_dumbbell min_roll_dumbbell min_pitch_dumbbell
## 1                 NA                                 NA                 NA
## 2                 NA                                 NA                 NA
## 3                 NA                                 NA                 NA
## 4                 NA                                 NA                 NA
## 5                 NA                                 NA                 NA
## 6                 NA                                 NA                 NA
##   min_yaw_dumbbell amplitude_roll_dumbbell amplitude_pitch_dumbbell
## 1                                       NA                       NA
## 2                                       NA                       NA
## 3                                       NA                       NA
## 4                                       NA                       NA
## 5                                       NA                       NA
## 6                                       NA                       NA
##   amplitude_yaw_dumbbell total_accel_dumbbell var_accel_dumbbell
## 1                                          37                 NA
## 2                                          37                 NA
## 3                                          37                 NA
## 4                                          37                 NA
## 5                                          37                 NA
## 6                                          37                 NA
##   avg_roll_dumbbell stddev_roll_dumbbell var_roll_dumbbell
## 1                NA                   NA                NA
## 2                NA                   NA                NA
## 3                NA                   NA                NA
## 4                NA                   NA                NA
## 5                NA                   NA                NA
## 6                NA                   NA                NA
##   avg_pitch_dumbbell stddev_pitch_dumbbell var_pitch_dumbbell
## 1                 NA                    NA                 NA
## 2                 NA                    NA                 NA
## 3                 NA                    NA                 NA
## 4                 NA                    NA                 NA
## 5                 NA                    NA                 NA
## 6                 NA                    NA                 NA
##   avg_yaw_dumbbell stddev_yaw_dumbbell var_yaw_dumbbell gyros_dumbbell_x
## 1               NA                  NA               NA                0
## 2               NA                  NA               NA                0
## 3               NA                  NA               NA                0
## 4               NA                  NA               NA                0
## 5               NA                  NA               NA                0
## 6               NA                  NA               NA                0
##   gyros_dumbbell_y gyros_dumbbell_z accel_dumbbell_x accel_dumbbell_y
## 1            -0.02             0.00             -234               47
## 2            -0.02             0.00             -233               47
## 3            -0.02             0.00             -232               46
## 4            -0.02            -0.02             -232               48
## 5            -0.02             0.00             -233               48
## 6            -0.02             0.00             -234               48
##   accel_dumbbell_z magnet_dumbbell_x magnet_dumbbell_y magnet_dumbbell_z
## 1             -271              -559               293               -65
## 2             -269              -555               296               -64
## 3             -270              -561               298               -63
## 4             -269              -552               303               -60
## 5             -270              -554               292               -68
## 6             -269              -558               294               -66
##   roll_forearm pitch_forearm yaw_forearm kurtosis_roll_forearm
## 1         28.4         -63.9        -153
## 2         28.3         -63.9        -153
## 3         28.3         -63.9        -152
## 4         28.1         -63.9        -152
## 5         28.0         -63.9        -152
## 6         27.9         -63.9        -152
##   kurtosis_picth_forearm kurtosis_yaw_forearm skewness_roll_forearm
## 1
## 2
## 3
## 4
## 5
## 6
##   skewness_pitch_forearm skewness_yaw_forearm max_roll_forearm
## 1                                                          NA
## 2                                                          NA
## 3                                                          NA
## 4                                                          NA
## 5                                                          NA
## 6                                                          NA
##   max_picth_forearm max_yaw_forearm min_roll_forearm min_pitch_forearm
## 1                NA                               NA                NA
## 2                NA                               NA                NA
## 3                NA                               NA                NA
## 4                NA                               NA                NA
## 5                NA                               NA                NA
## 6                NA                               NA                NA
##   min_yaw_forearm amplitude_roll_forearm amplitude_pitch_forearm
## 1                                    NA                      NA
## 2                                    NA                      NA
## 3                                     NA                      NA
## 4                                     NA                      NA
## 5                                     NA                      NA
## 6                                     NA                      NA
##   amplitude_yaw_forearm total_accel_forearm var_accel_forearm
## 1                                        36                NA
## 2                                        36                NA
## 3                                        36                NA
## 4                                        36                NA
## 5                                        36                NA
## 6                                        36                NA
##   avg_roll_forearm stddev_roll_forearm var_roll_forearm avg_pitch_forearm
## 1               NA                  NA               NA                NA
## 2               NA                  NA               NA                NA
## 3               NA                  NA               NA                NA
## 4               NA                  NA               NA                NA
## 5               NA                  NA               NA                NA
## 6               NA                  NA               NA                NA
##   stddev_pitch_forearm var_pitch_forearm avg_yaw_forearm
## 1                   NA                NA              NA
## 2                   NA                NA              NA
## 3                   NA                NA              NA
## 4                   NA                NA              NA
## 5                   NA                NA              NA
## 6                   NA                NA              NA
##   stddev_yaw_forearm var_yaw_forearm gyros_forearm_x gyros_forearm_y
## 1                 NA              NA            0.03            0.00
## 2                 NA              NA            0.02            0.00
## 3                 NA              NA            0.03           -0.02
## 4                 NA              NA            0.02           -0.02
## 5                 NA              NA            0.02            0.00
## 6                 NA              NA            0.02           -0.02
##   gyros_forearm_z accel_forearm_x accel_forearm_y accel_forearm_z
## 1           -0.02             192             203            -215
## 2           -0.02             192             203            -216
## 3            0.00             196             204            -213
## 4            0.00             189             206            -214
## 5           -0.02             189             206            -214
## 6           -0.03             193             203            -215
##   magnet_forearm_x magnet_forearm_y magnet_forearm_z classe
## 1              -17              654              476      A
## 2              -18              661              473      A
## 3              -18              658              469      A
## 4              -16              658              469      A
## 5              -17              655              473      A
## 6               -9              660              478      A
```

There are 100 columns with mostly ‘NAs’ or ‘#DIV/0’, rather than acutal values. As mentioned above, I did try removing the ‘DIV/0’ and change the ‘NAs’ to the median values of the column. But the analysis results were poor, so, instead, I’m removing these columns:

```
exParedTrain <- exTrain[,c(1:11,37:49,60:68,84:86,102, 113:124,140,151:160)]

# Note: I found by looking at the data in a spreadsheet, that the
exParedTest <- exTest[c(1:11,37:49,60:68,84:86,102, 113:124,140,151:160)]
```

The provided test set does not include the categorization (classe) values. So we’ll divide up the training dataset to allow us to have a ‘test’ set to use for evaluating our models.

```
library(caTools)
set.seed(1234)
split <- sample.split(exParedTrain$classe, SplitRatio = 0.80)
exSplitTrain <- subset(exParedTrain, split == TRUE)
exSplitTest <- subset(exParedTrain, split == FALSE)
```

Other values that will be needed for most, if not all, the models:

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

```
## Loading required package: ggplot2
```

Visualizations of data

Let’s do some discovery on our data with a heatmap.

```
library(reshape2)
library(ggplot2)
library(plyr)
library(scales)

exTrain.m <- melt(exParedTrain)
```

```
## Using user_name, cvtd_timestamp, new_window, classe as id variables
```

```
exTrain.m <- ddply(exTrain.m, .(variable), transform, rescale = rescale(value))
p <- ggplot(exTrain.m, aes(classe, variable))
p = p + geom_tile(aes(fill = rescale), colour = "white")
```

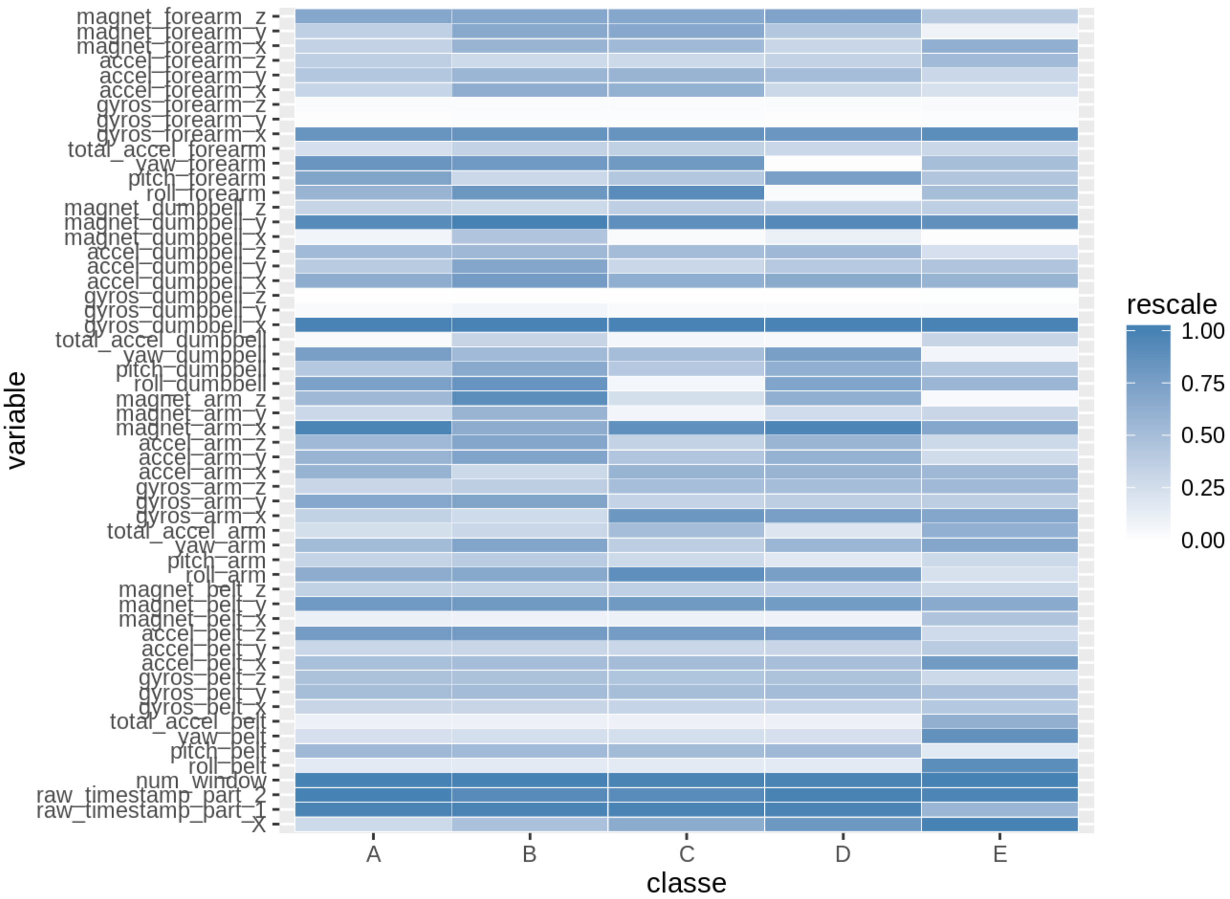
```
## Warning in structure(NULL, class = "waiver"): Calling 'structure(NULL, *)' i
s deprecated, as NULL cannot have attributes.
##   Consider 'structure(list(), *)' instead.
```

```
p <- p + scale_fill_gradient(low = "white", high = "steelblue")
```

```
## Warning in structure(NULL, class = "waiver"): Calling 'structure(NULL, *)' i
s deprecated, as NULL cannot have attributes.
##   Consider 'structure(list(), *)' instead.

## Warning in structure(NULL, class = "waiver"): Calling 'structure(NULL, *)' i
s deprecated, as NULL cannot have attributes.
##   Consider 'structure(list(), *)' instead.

## Warning in structure(NULL, class = "waiver"): Calling 'structure(NULL, *)' i
s deprecated, as NULL cannot have attributes.
##   Consider 'structure(list(), *)' instead.

## Warning in structure(NULL, class = "waiver"): Calling 'structure(NULL, *)' i
s deprecated, as NULL cannot have attributes.
##   Consider 'structure(list(), *)' instead.

## Warning in structure(NULL, class = "waiver"): Calling 'structure(NULL, *)' i
s deprecated, as NULL cannot have attributes.
##   Consider 'structure(list(), *)' instead.
```

```
p
```

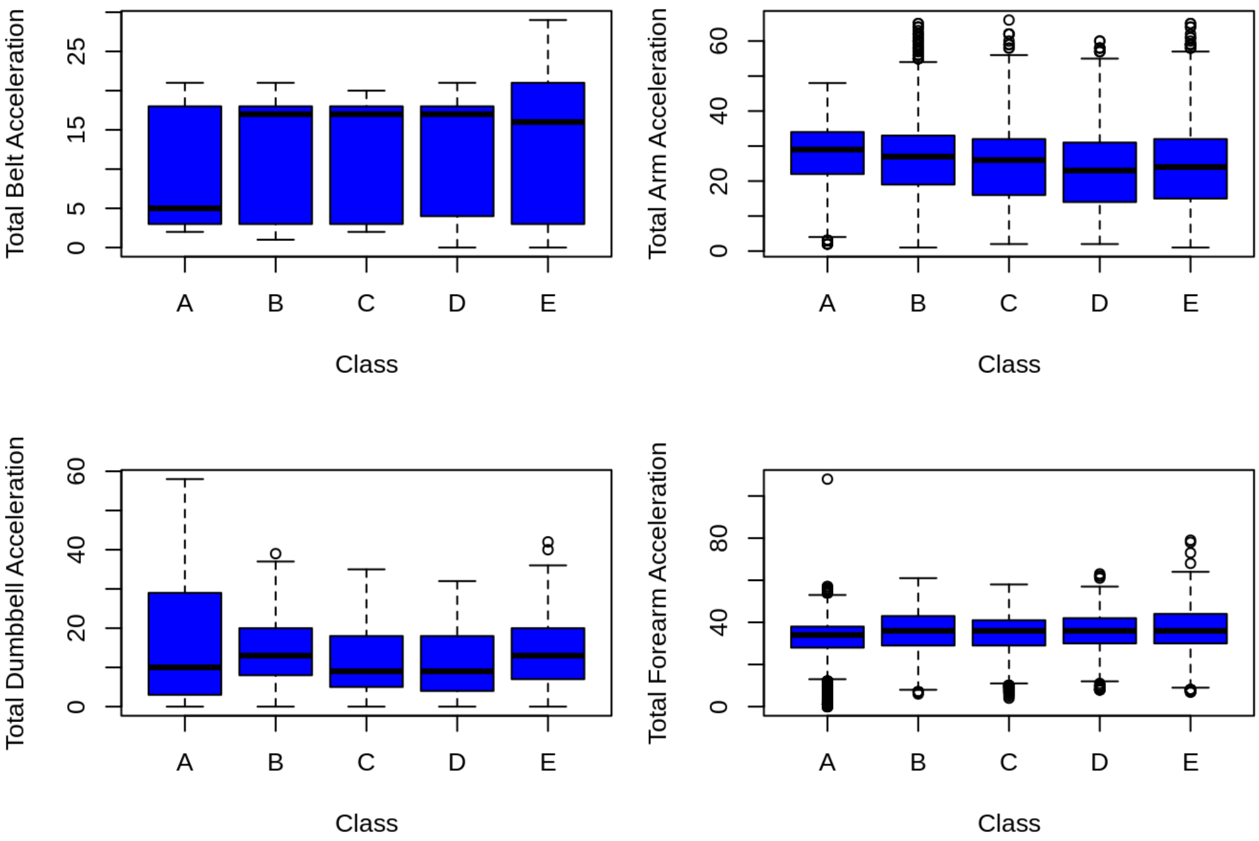
```
## Warning in structure(NULL, class = "waiver"): Calling 'structure(NULL, *)' i
s deprecated, as NULL cannot have attributes.
##   Consider 'structure(list(), *)' instead.

## Warning in structure(NULL, class = "waiver"): Calling 'structure(NULL, *)' i
s deprecated, as NULL cannot have attributes.
##   Consider 'structure(list(), *)' instead.

## Warning in structure(NULL, class = "waiver"): Calling 'structure(NULL, *)' i
s deprecated, as NULL cannot have attributes.
##   Consider 'structure(list(), *)' instead.

## Warning in structure(NULL, class = "waiver"): Calling 'structure(NULL, *)' i
s deprecated, as NULL cannot have attributes.
##   Consider 'structure(list(), *)' instead.

## Warning in structure(NULL, class = "waiver"): Calling 'structure(NULL, *)' i
s deprecated, as NULL cannot have attributes.
##   Consider 'structure(list(), *)' instead.

## Warning in structure(NULL, class = "waiver"): Calling 'structure(NULL, *)' i
s deprecated, as NULL cannot have attributes.
##   Consider 'structure(list(), *)' instead.

## Warning in structure(NULL, class = "waiver"): Calling 'structure(NULL, *)' i
s deprecated, as NULL cannot have attributes.
##   Consider 'structure(list(), *)' instead.

## Warning in structure(NULL, class = "waiver"): Calling 'structure(NULL, *)' i
s deprecated, as NULL cannot have attributes.
##   Consider 'structure(list(), *)' instead.

## Warning in structure(NULL, class = "waiver"): Calling 'structure(NULL, *)' i
s deprecated, as NULL cannot have attributes.
##   Consider 'structure(list(), *)' instead.
```



Note: I'm following the examples in the help for the heatmap. I have tried modifying the arguments but have not been able to get rid of the error messages. However, the final product looks to be correct.

The following shows 'total' values for all the major categories. Since it would be difficult to review all the values for each class, these boxplots of the 'totals' vs. class does give us some information as to where divisions may lie.

```
par(mfrow = c(2,2), mar = c(5,4,2,1))
boxplot(total_accel_belt ~ classe, exParedTrain, xlab = 'Class', ylab = 'Total
Belt Acceleration', col = "blue")
boxplot(total_accel_arm ~ classe, exParedTrain, xlab = 'Class', ylab = 'Total A
rm Acceleration', col = "blue")
boxplot(total_accel_dumbbell ~ classe, exParedTrain, xlab = 'Class', ylab = 'To
tal Dumbbell Acceleration', col = "blue")
boxplot(total_accel_forearm ~ classe, exParedTrain, xlab = 'Class', ylab = 'Tot
al Forearm Acceleration', col = "blue")
```



The Total Arm Acceleration shows few differences between the different classes, although there are fewer outliers for A than the other classes. The Total Belt Acceleration shows the widest variety. And there is a great difference in the median between A and the others. The Dumbell medians are more widely spaced than the other readings, although they are still close.

Machine Learning

K-Means Clustering

```
set.seed(1234)
kmFit <- kmeans(exSplitTrain[,7:59], 5, nstart = 20)

table(kmFit$cluster, exSplitTrain$classe)
```

```
##
##        A    B    C    D    E
##   1  764 1021  809 1128 1052
##   2  505  383  392  376  354
##   3  448  235  194  222  153
##   4  567  833  341  499  824
##   5 2180  566 1002  348  503
```

K Means Clustering is very ineffective, it seems. Or at least with the hyperparameters I've selected.

Support Vector Machine

```
library(e1071)

svmModel <- svm(formula = classe ~ .,
                data = exSplitTrain,
                type = 'C-classification',
                kernel = 'polynomial')

SvmPred = predict(svmModel, newdata = exSplitTest)
table(exSplitTest$classe, SvmPred)
```

```
##    SvmPred
##        A    B    C    D    E
##   A 1111    5    0    0    0
##   B   26  721   12    0    0
##   C    0   12  670    2    0
##   D    0    0   48  594    1
##   E    0    0    1   17  703
```

K-NN

```
library(caret)
Sys.time()
```

```
## [1] "2019-04-27 10:16:10 EDT"
```

```
knnModel <- train(classe ~ ., data = exSplitTrain, method = "knn",
                  preProcess = c("center", "scale"),
                  trControl = trainValues,
                  metric = "Accuracy",
                  tuneLength = tunel)
Sys.time()
```

```
## [1] "2019-04-27 10:36:38 EDT"
```

```
knnModel
```

```
## k-Nearest Neighbors
##
## 15699 samples
##    59 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## Pre-processing: centered (81), scaled (81)
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 14129, 14130, 14129, 14130, 14129, 14128, ...
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    5  0.9783640  0.9726284
##    7  0.9725037  0.9652077
##    9  0.9668139  0.9580036
##   11  0.9613785  0.9511209
##   13  0.9553903  0.9435415
##   15  0.9502944  0.9370946
##   17  0.9455376  0.9310742
##   19  0.9408662  0.9251704
##   21  0.9377027  0.9211699
##   23  0.9339442  0.9164212
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 5.
```

Predicting values with the K-NN model:

```
knnPredict <- predict(knnModel, exSplitTest)
table(exSplitTest$classe, knnPredict)
```

```
##    knnPredict
##        A    B    C    D    E
##   A 1102   10    4    0    0
##   B   19  727   13    0    0
##   C    1   11  667    5    0
##   D    0    0   15  626    2
##   E    0    0    0    7  714
```

Random Forest

```
library(caret)
Sys.time()
```

```
## [1] "2019-04-27 10:36:53 EDT"
```

```
rfFit <- train(classe ~ .,
                data = exSplitTrain,
                method = "rf",
                prox = TRUE,
                preProcess = c("center","scale"),
                trControl = trainValues)
Sys.time()
```

```
## [1] "2019-04-27 15:08:53 EDT"
```

```
rfFit
```

```
## Random Forest
##
## 15699 samples
##    59 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## Pre-processing: centered (81), scaled (81)
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 14131, 14128, 14129, 14129, 14128, 14129, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.9960504  0.9950041
##   41    0.9998301  0.9997850
##   81    0.9997664  0.9997045
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was mtry = 41.
```

Random Forest Prediction:

```
rfPred <- predict(rfFit, exSplitTest)
table(rfPred,exSplitTest$classe)
```

```
##
## rfPred    A    B    C    D    E
##      A 1116    0    0    0    0
##      B    0  759    0    0    0
##      C    0    0  684    0    0
##      D    0    0    0  643    0
##      E    0    0    0    0  721
```

Random Forest took quite awhile (4-5 hrs) to run. But the results were perfect for the test set. Therefore, this is the model I will be using for the test set.

Comparing Results from Different Algorithms

```
tmpKnn <- exSplitTest$classe == knnPredict
tmpSVM <- SvmPred == exSplitTest$classe
tmpRf <- rfPred == exSplitTest$classe

print('K-NN')
```

```
## [1] "K-NN"
```

```
sum(tmpKnn == TRUE)/length(exSplitTest$classe) * 100
```

```
## [1] 97.78231
```

```
print('SVM')
```

```
## [1] "SVM"
```

```
sum(tmpSVM == TRUE)/length(exSplitTest$classe) * 100
```

```
## [1] 96.83915
```

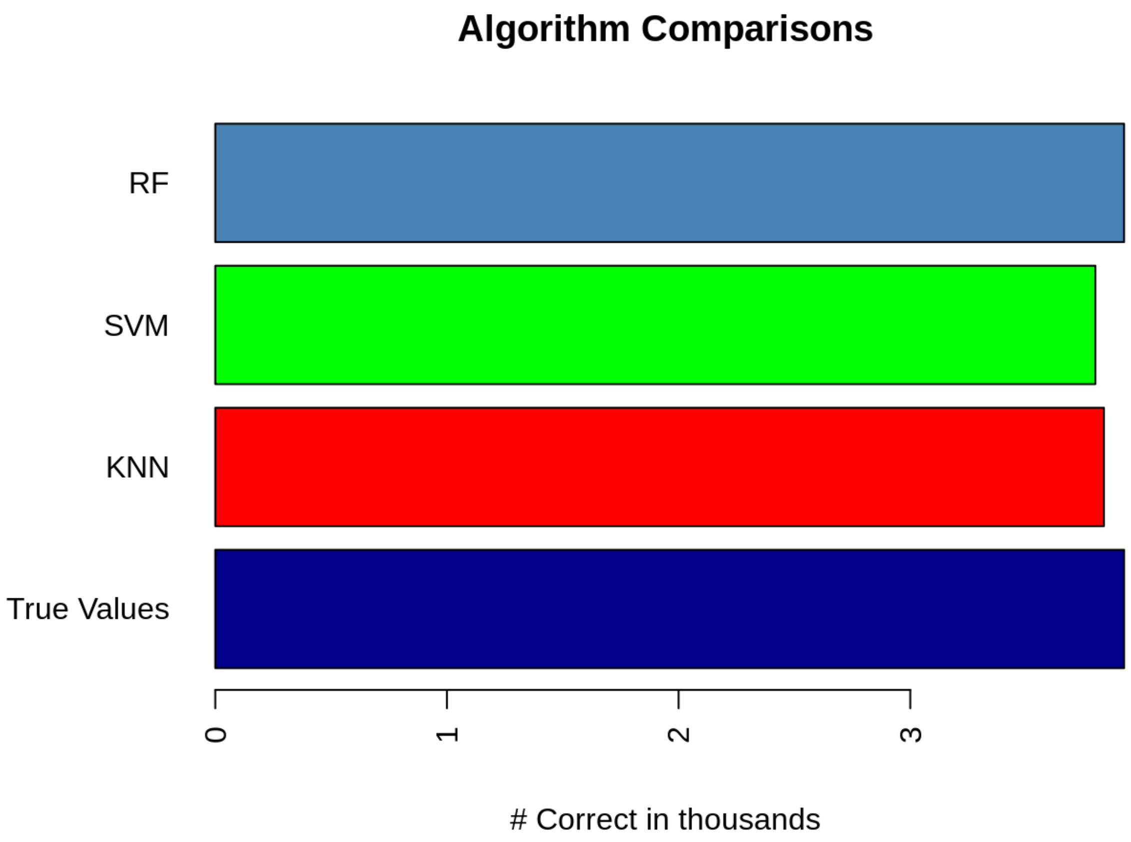
```
print('RF')
```

```
## [1] "RF"
```

```
sum(tmpRf == TRUE)/length(exSplitTest$classe) * 100
```

```
## [1] 100
```

```
counts <- c(length(exSplitTest$classe)/1000, sum(tmpKnn == TRUE)/1000, sum(tmpSVM == TRUE)/1000, sum(tmpRf == TRUE)/1000)

par(las=2)
par(mar=c(5,8,4,2))
barplot(counts,
        main = "Algorithm Comparisons",
        xlab = "# Correct in thousands",
        horiz = TRUE,
        names.arg=c("True Values",'KNN','SVM','RF'),
        col=c('darkblue','red','green','steelblue'))
```



Summary

The training data was divided into ‘test’ and ‘train’, since the test data provided did not include the classification. 4 algorithms were tested on the data. KMeans clustering did so poorly on the training data that it wasn’t even tried on the test data. K-NN and SVM did fairly well: 97.83% and 96.84% respectively. Random Forest, although it took 6 hours to run, was the most successful, getting 100% of the test samples correct.