Code

Predictions using the Weight Lifting Exercises Dataset

1. Explore the data, especially focussing on the two paramaters we are interested in
2. Model selection, where we try different models to help us answer our questions
3. Model examination, to see wether our best model holds up to our standards
4. A Conclusion where we answer the questions based on the data
5. Predicting the classification of the model on test set

Importing data and exploation

Hide

```
training <- read.csv("./pml-training.csv")
testing <- read.csv("./pml-testing.csv")
```

Hide

```
dim(training)
```

```
[1] 19622   160
```

Hide

```
head(training)
```

	X <int>	user_na... <fctr>	raw_timestamp_part_1 <int>	raw_timestamp_part_2 <int>	cvtd_timestamp <fctr>	new_wi <fctr>
1	1	carlitos	1323084231	788290	05/12/2011 11:23	no
2	2	carlitos	1323084231	808298	05/12/2011 11:23	no
3	3	carlitos	1323084231	820366	05/12/2011 11:23	no
4	4	carlitos	1323084232	120339	05/12/2011 11:23	no
5	5	carlitos	1323084232	196328	05/12/2011 11:23	no
6	6	carlitos	1323084232	304277	05/12/2011 11:23	no

6 rows | 1-8 of 160 columns

Cleaning data

Hide

```
maxNAPerc = 20
maxNACount <- nrow(training) / 100 * maxNAPerc
removeColumns <- which(colSums(is.na(training) | training=="") > maxNACount)
training.cleaned01 <- training[,-removeColumns]
testing.cleaned01 <- testing[,-removeColumns]
```

Hide

```
removeColumns <- grep("timestamp", names(training.cleaned01))
training.cleaned02 <- training.cleaned01[,-c(1, removeColumns )]
testing.cleaned02 <- testing.cleaned01[,-c(1, removeColumns )]
```

Hide

```
classeLevels <- levels(training.cleaned02$classe)
training.cleaned03 <- data.frame(data.matrix(training.cleaned02))
training.cleaned03$classe <- factor(training.cleaned03$classe, labels=classeLevels)
testing.cleaned03 <- data.frame(data.matrix(testing.cleaned02))
```

Hide

```
training.cleaned <- training.cleaned03
testing.cleaned <- testing.cleaned03
```

Hide

```
set.seed(19791108)
library(caret)
```

```
le package <U+393C><U+3E31>caret<U+393C><U+3E32> a <U+653C><U+3E39>t<U+653C><U+3E39> compil<U
+653C><U+3E39> avec la version R 3.6.3Le chargement a n<U+653C><U+3E39>cessit<U+653C><U+3E39>
le package : lattice
le package <U+393C><U+3E31>lattice<U+393C><U+3E32> a <U+653C><U+3E39>t<U+653C><U+3E39> compil
<U+653C><U+3E39> avec la version R 3.6.3Le chargement a n<U+653C><U+3E39>cessit<U+653C><U+3E3
9> le package : ggplot2
le package <U+393C><U+3E31>ggplot2<U+393C><U+3E32> a <U+653C><U+3E39>t<U+653C><U+3E39> compil
<U+653C><U+3E39> avec la version R 3.6.3
```

Hide

```
classeIndex <- which(names(training.cleaned) == "classe")
partition <- createDataPartition(y=training.cleaned$classe, p=0.75, list=FALSE)
training.subSetTrain <- training.cleaned[partition, ]
training.subSetTest <- training.cleaned[-partition, ]
```

Feature correlations

Hide

```
correlations <- cor(training.subSetTrain[, -classeIndex], as.numeric(training.subSetTrain$cla
sse))
bestCorrelations <- subset(as.data.frame(as.table(correlations)), abs(Freq)>0.3)
bestCorrelations
```

	Var1 <fctr>	Var2 <fctr>	Freq <dbl>
44	pitch_forearm	A	0.336018

1 row

Some graphical representations

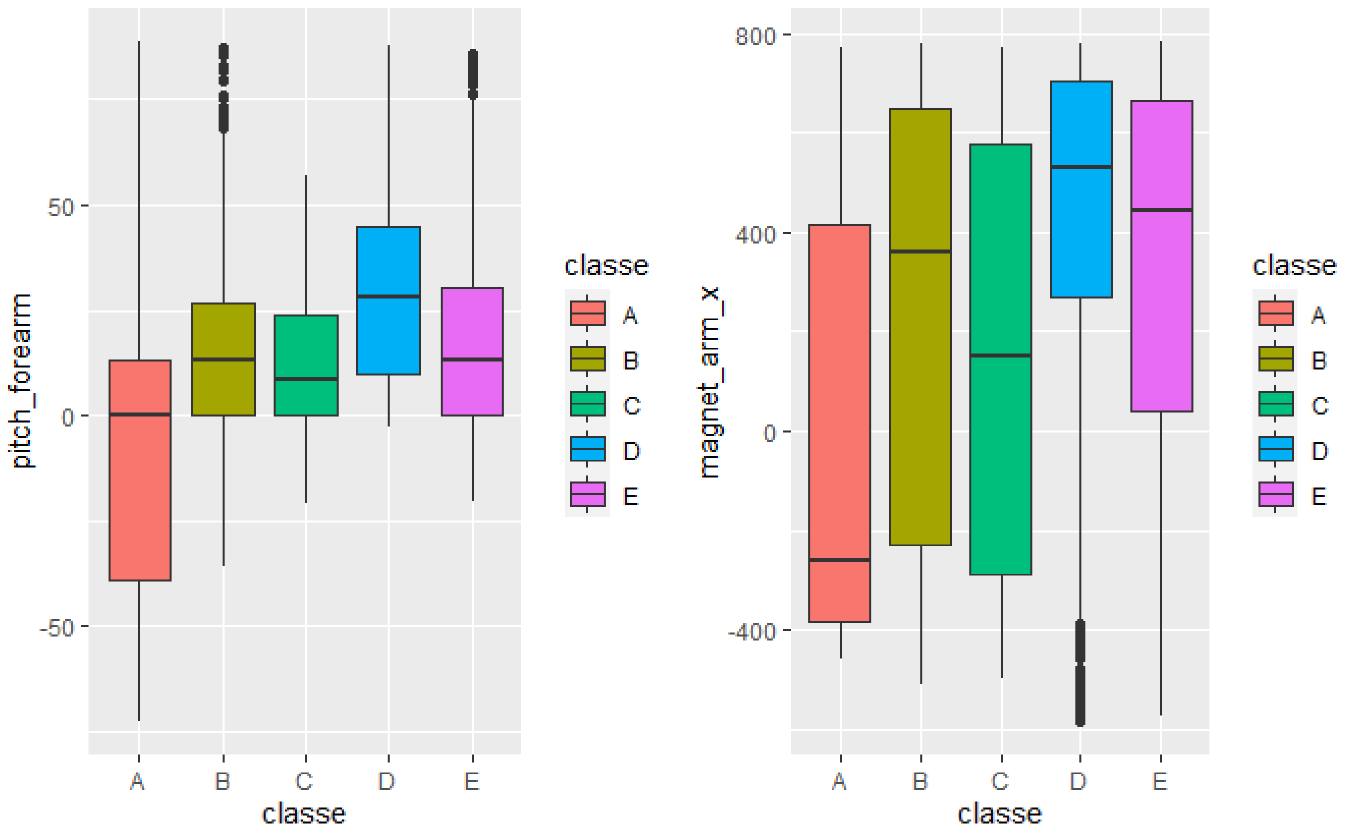
Hide

```
library(Rmisc)
```

```
le package <U+393C><U+3E31>Rmisc<U+393C><U+3E32> a <U+653C><U+3E39>t<U+653C><U+3E39> compil<U
+653C><U+3E39> avec la version R 3.6.3Le chargement a n<U+653C><U+3E39>cessit<U+653C><U+3E39>
le package : plyr
le package <U+393C><U+3E31>plyr<U+393C><U+3E32> a <U+653C><U+3E39>t<U+653C><U+3E39> compil<U+
653C><U+3E39> avec la version R 3.6.3
```

Hide

```
library(ggplot2)
p1 <- ggplot(training.subSetTrain, aes(classe,pitch_forearm)) +
geom_boxplot(aes(fill=classe))
p2 <- ggplot(training.subSetTrain, aes(classe, magnet_arm_x)) +
geom_boxplot(aes(fill=classe))
multiplot(p1,p2,cols=2)
```



The correlations heatmap

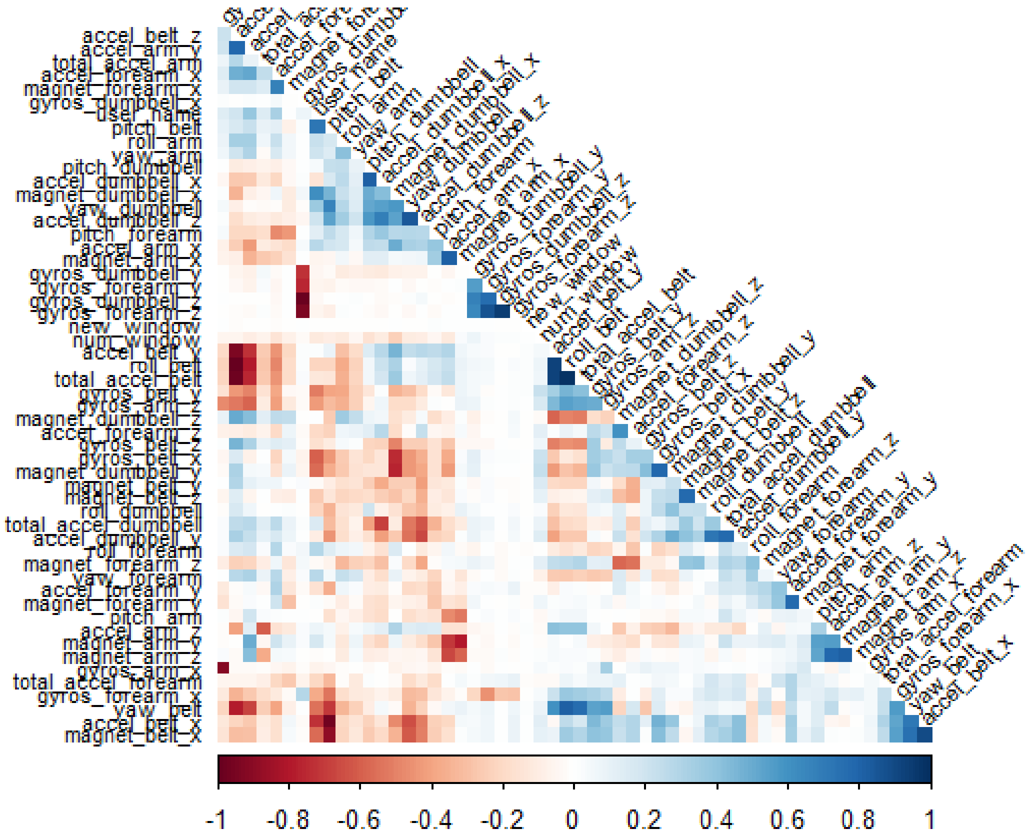
Hide

```
library(corrplot)
```

```
le package <U+393C><U+3E31>corrplot<U+393C><U+3E32> a <U+653C><U+3E39>t<U+653C><U+3E39> compi
l<U+653C><U+3E39> avec la version R 3.6.3corrplot 0.84 loaded
```

Hide

```
correlationMatrix <- cor(training.subSetTrain[, -classeIndex])
highlyCorrelated <- findCorrelation(correlationMatrix, cutoff=0.9, exact=TRUE)
excludeColumns <- c(highlyCorrelated, classeIndex)
corrplot(correlationMatrix, method="color", type="lower", order="hclust", tl.cex=0.70, tl.col
="black", tl.srt = 45, diag = FALSE)
```



Classification methods

Random Forest

Hide

```
library(rpart)
```

```
le package <U+393C><U+3E31>rpart<U+393C><U+3E32> a <U+653C><U+3E39>t<U+653C><U+3E39> compil<U
+653C><U+3E39> avec la version R 3.6.3
```

Hide

```
library(rpart.plot)
```

```
le package <U+393C><U+3E31>rpart.plot<U+393C><U+3E32> a <U+653C><U+3E39>t<U+653C><U+3E39> com
pil<U+653C><U+3E39> avec la version R 3.6.3
```

Hide

```
library(rattle)
```

```
le package <U+393C><U+3E31>rattle<U+393C><U+3E32> a <U+653C><U+3E39>t<U+653C><U+3E39> compil<
U+653C><U+3E39> avec la version R 3.6.3Rattle: A free graphical interface for data science wi
th R.
Version 5.3.0 Copyright (c) 2006-2018 Togaware Pty Ltd.
Entrez 'rattle()' pour secouer, faire vibrer, et faire d<U+653C><U+3E39>filer vos donn<U+653C
><U+3E39>es.
```

Hide

```
training <- read.csv("./pml-training.csv")
testing <- read.csv("./pml-testing.csv")
label <- createDataPartition(training$classe, p = 0.7, list = FALSE)
train <- training[label, ]
test <- training[-label, ]
```

Hide

```
NZV <- nearZeroVar(train)
train <- train[ ,-NZV]
test <- test[ ,-NZV]
label <- apply(train, 2, function(x) mean(is.na(x))) > 0.95
train <- train[, -which(label, label == FALSE)]
test <- test[, -which(label, label == FALSE)]
train <- train[ , -(1:5)]
test <- test[ , -(1:5)]
```

Hide

```
library(caret)
set.seed(13908)
control <- trainControl(method = "cv", number = 3, verboseIter=FALSE)
modelRF <- train(classe ~ ., data = train, method = "rf", trControl = control)
modelRF$finalModel
```

```
Call:
 randomForest(x = x, y = y, mtry = param$mtry)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 27

        OOB estimate of  error rate: 0.2%
Confusion matrix:
     A    B    C    D    E  class.error
A 3905    0    0    0    1 0.0002560164
B    6 2650    2    0    0 0.0030097818
C    0    5 2391    0    0 0.0020868114
D    0    0    8 2244    0 0.0035523979
E    0    0    0    5 2520 0.0019801980
```

Hide

```
predictRF <- predict(modelRF, test)
confMatRF <- confusionMatrix(predictRF, test$classe)
confMatRF
```

```
Confusion Matrix and Statistics

          Reference
Prediction    A    B    C    D    E
         A 1674    6    0    0    0
         B    0 1131    1    0    0
         C    0    2 1025    6    0
         D    0    0    0  958    0
         E    0    0    0    0 1082

Overall Statistics

               Accuracy : 0.9975
                 95% CI : (0.9958, 0.9986)
    No Information Rate : 0.2845
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9968

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: A Class: B Class: C Class: D Class: E
Sensitivity            1.0000   0.9930   0.9990   0.9938   1.0000
Specificity            0.9986   0.9998   0.9984   1.0000   1.0000
Pos Pred Value         0.9964   0.9991   0.9923   1.0000   1.0000
Neg Pred Value         1.0000   0.9983   0.9998   0.9988   1.0000
Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
Detection Rate         0.2845   0.1922   0.1742   0.1628   0.1839
Detection Prevalence   0.2855   0.1924   0.1755   0.1628   0.1839
Balanced Accuracy      0.9993   0.9964   0.9987   0.9969   1.0000
```

Gradient Boosting

Hide

```
library(rpart)
library(rpart.plot)
library(rattle)
```

Hide

```
control <- trainControl(method = "repeatedcv", number = 5, repeats = 1, verboseIter = FALSE)
modelGBM <- train(classe ~ ., data = train, trControl = control, method = "gbm", verbose = FA
LSE)
modelGBM$finalModel
```

```
A gradient boosted model with multinomial loss function.
150 iterations were performed.
There were 53 predictors of which 52 had non-zero influence.
```

Hide

```
predictGBM <- predict(modelGBM, test)
confMatGBM <- confusionMatrix(predictGBM, test$classe)
confMatGBM
```

```
Confusion Matrix and Statistics

          Reference
Prediction    A    B    C    D    E
         A 1667   16    0    0    0
         B    6 1108    5    3    3
         C    0   15 1015   16    4
         D    1    0    6  945    4
         E    0    0    0    0 1071

Overall Statistics

               Accuracy : 0.9866
                 95% CI : (0.9833, 0.9894)
    No Information Rate : 0.2845
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.983

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: A Class: B Class: C Class: D Class: E
Sensitivity            0.9958   0.9728   0.9893   0.9803   0.9898
Specificity            0.9962   0.9964   0.9928   0.9978   1.0000
Pos Pred Value         0.9905   0.9849   0.9667   0.9885   1.0000
Neg Pred Value         0.9983   0.9935   0.9977   0.9961   0.9977
Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
Detection Rate         0.2833   0.1883   0.1725   0.1606   0.1820
Detection Prevalence   0.2860   0.1912   0.1784   0.1624   0.1820
Balanced Accuracy      0.9960   0.9846   0.9910   0.9890   0.9949
```