~~~
---
title: "Practical ML Course Project - Weight Lifting Excercise Dataset"
author: "JJanzen"
date: "May 22, 2015"
output: html_document
---

Using the dataset comes from "wearables", such as FitBit and Jawbone Up,
six participants measured themselves doing a barbel bicep curl.  They
were asked to perform the lifts correctly and incorrectly five differnet
ways.  Using the training data, I'm to create a model which will then
predict on a testing set of 20 test cases to see how accurate my model
was.

## 1. Download the data
```{r}
setwd("/Users/a149174/JHDataScience/machine_learning/cp")
training <- read.csv("training.csv", header=T, na.strings=c("NA",""))
testing <- read.csv("testing.csv", header=T, na.strings=c("NA",""))
# get dimesions before cleaning
dim(training)
dim(testing)
```
## 2. Understand the data
```{r, echo=FALSE}
# of the 19,622 rows of data, I wanted to understand the frequency of
classe
plot(training$classe, col="green", main="Frequency of Classe Levels",
xlab="classe levels", ylab="Frequency")
```

## 3. Clean the data
```{r}
# delete columns with all missing values
training <- training[,colSums(is.na(training)) == 0]
testing <- testing[,colSums(is.na(testing)) == 0]

# remove first seven columns not valid for machine learning (x,
username, timestamps, new_window, and num_window )
training   <-training[,-c(1:7)]
testing <-testing[,-c(1:7)]
```

## 4. Split the data to create a training and test set
```{r}
library(caret);
set.seed(1000)
inTrain <- createDataPartition(y=training$classe, p=0.75, list=F)
my_training <- training[inTrain,]
my_testing <- training[-inTrain,]
~~~

~~~~
```

## 5. Create the using model Random Forest
```{r}
set.seed(1000)
# use random forest with resampling with cross-validation 4-fold
modelFit_rf <- train(classe~., data=my_training, method="rf",
trControl=trainControl(method="cv", number = 4))
modelFit_rf
# final model
modelFit_rf$finalModel
# prediction
predictions_rf <- predict(modelFit_rf, newdata=my_testing)
# confusion matrix
confusionMatrix(predictions_rf, my_testing$classe)
```

## 6. Create the using K-Nearest Neighbor
```{r}
set.seed(1000)
# use k-nearest neighbor with resampling with cross-validation 4-fold
modelFit_knn <- train(classe~., data=my_training, method="knn", metric =
"Accuracy", trControl=trainControl(method="cv", number = 4))
modelFit_knn
# final model
modelFit_knn$finalModel
# prediction
predictions_knn <- predict(modelFit_knn, newdata=my_testing)
# confusion matrix
confusionMatrix(predictions_knn, my_testing$classe)
```

## 7. Out of sample error
The model with highest accuracy was random forest.  I wanted to measure
out of sample error on my testing set (which is 25% of the inital
training set).  Accurancy of random forest was 99.27%, so the out of
sample error is 1 - 0.9927 or 0.0073.  Based on this low out of sample
error, we should have very few to no misclassified on the test samples.

## 8. Predict the Samples
Using random forest
```{r}
# test on original testing set
predictions_final <- predict(modelFit, newdata=testing)
predictions_final
```
~~~~