~~~
---
title: "Practical Machine Learning Project"
author: "Joseph Coggins"
date: "September 26, 2015"
output: html_document
---

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word
documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of
any embedded R code chunks within the document. You can embed an R code chunk like this:
```{r}

```


```{r}
```
The machine learning package is the caret package. The ggplot2
package is a good standardized plotting package.
```{r}
library(ggplot2)
library(caret)
set.seed(323)

```
Here is the URL path to the training.csv file
```{r}
trainingURL = "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
```
Here is the URL path to the testing.csv file
```{r}
testingURL = "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
```

Extracting the csv files
Now we can read the csv files into memory
training
```{r}
training = read.csv(url(trainingURL),, header=T, sep=",", na.strings=c("NA", "#DIV/0!", ""))
```
testing
```{r}
testing = read.csv(url(testingURL), header=T, sep=",", na.strings=c("NA", "#DIV/0!", ""))
```
Exploratory Analysis:
```{r}
#summary(training, 5)
#head(training, 5)
#names(training)
```
The head() without using na.strings shows NA, blanks, #Div/0! and near zero data.
This data should be cleaned from the training data
to train the machine on clean data. Colums without
data should be removed as well.



Let's Partition the given Training Data
```{r}
inTrain = createDataPartition(y=training$classe, p=0.7, list=FALSE)
trainingData = training[inTrain,]
testingData = training[-inTrain,]
dim(trainingData)
dim(testingData)
```

Determine the datatype for each column
```{r}
#sapply(trainingData, class)

```
Using Google search the correct link was found at
Human Activity Recognition - Groupware@LES - PUC-Rio
groupware.les.inf.puc-rio.br/har
Human Activity Recognition - HAR - has emerged as a key research area in the ... dataset

Checking the names of the first 7 columns
```{r}
trainingData_first7Colums = trainingData[,1:7]
names(trainingData_first7Colums)
# Check names for physical exercises
trainingData = trainingData[,8:length(colnames(trainingData))]
names(trainingData)
```
~~~

~~~
Cleaning out first 7 unnecessary columns
```{r}
trainingData = trainingData[,8:length(colnames(trainingData))]
testingData = testingData[,8:length(colnames(testingData))]
#head(trainingData)
```

Find all the columns that are factors and ignore the last column
which is the classe column
```{r}
col_names <- c()
n <- ncol(trainingData)-1
print(n)
for (i in 1:n) {
     if (is.factor(trainingData[,i])){
             col_names = c(col_names,i)
           }
}
```

Remove the factor columns
```{r}
#trainingData = trainingData[,-col_names]
```




Cleaning out columns with NAs
```{r}
trainingData = trainingData[, colSums(is.na(trainingData)) == 0]
testingData = testingData[, colSums(is.na(testingData)) == 0]
#head(trainingData)
```
Cleaning out columns with near zero variance
```{r}
nzVariance = nearZeroVar(trainingData, saveMetrics = TRUE)
zeroVar = sum(nzVariance$nzVariance)

if((zeroVar >0)) {
  trainingData = trainingData[,nzVariance$nzVariance==FALSE]
}

#head(trainingData)
```
Data Plot
```{r, echo=FALSE}
library(ggplot2)
table(trainingData$classe)
qplot(roll_forearm, roll_arm, colour=classe, data=trainingData)

```

Decision Tree Plot
```{r, echo=FALSE}
library(caret)

modFit = train(classe ~., method="rpart", data=trainingData)
library(rattle)
fancyRpartPlot(modFit$finalModel)

```

Prediction:
```{r}
prediction = predict(modFit, testingData)
#table(prediction, testingData$classe)
```

Use confusion matrix to check Accuracy
```{r}
#confusionMatrix(predict(modFit, testing), testing$classe)
```
~~~