Here's the GitHub link to my repository.
https://github.com/Lifelearner8109/PML-Version2.0.git

Installed Libraries

1. caret
2. e1071
3. rattle
4. rpart
5. rpart.plot
6. randomForest

Data Cleaning
After loading the training and testing data, the summary() function on the training set revealed a significant pattern in missing data. 19,216 N/A values appeared throughout the data set. Relative to the entire collection of 19,622 training observations, 97.9% of the data was missing. N/A values or null values are common to large data sets, but the same number of missing values across the entire data set implied errors and/or biases that occurred during the data collection process. As a way to eliminate these errors/and or biases, I removed all variables containing 19,216 N/A values. The following R commands completed this task.

```
#Remove the NA and Near Zero Variance Variables
nzvCols = nearZeroVar(train)
trainClean=train[,-nzvCols]
testClean=test[,-nzvCols]
ColsToRemove=which(sapply(trainClean, function(x) sum(is.na(x)))>=19216)
trainReady=trainClean[,-ColsToRemove]
testReady=testClean[,-ColsToRemove]
```

Training, Validation and Test Data
Before creating any models, I separated the training set into training and validation sets. The training set was used to create the models, and the validation served as my test set before running the model on the actual test data. I did this so that the classification results from the validation set would serve as my best estimate of the out-of-sample error. Here is the corresponding R code for creating these data sets.

```
#Create Training, Validation & Test Data
inTraining=createDataPartition(y=trainReady$classe, p=0.7, list=FALSE)
training=trainReady[inTraining,]
validation=trainReady[-inTraining,]
testing=testReady
```

Model Construction
I created the first model using the randomForest package in R (see below for the syntax and output).

```
mod1=randomForest(classe~.-X - user_name - raw_timestamp_part_1 - raw_timestamp_part_2 -
cvtd_timestamp, data=training, ntree=200, method="class")

pred1=predict(mod1, newdata=validation, type="class")
confusionMatrix(pred1, validation$classe)
```

Confusion Matrix and Statistics

```
          Reference
Prediction    A    B    C    D    E
         A 1673    0    0    0    0
         B    0 1139    1    0    0
         C    0    0 1025    9    0
         D    0    0    0  955    0
         E    1    0    0    0 1082
```

Accuracy : 0.9981

Out of Sample Error & Cross Validation
The random forest model (mod1) was extremely accurate on the validation set with an overall accuracy of 99.81%. Cross validation yielded similar results (see below for syntax and output).

```
mod2=train(classe~.-X - user_name - raw_timestamp_part_1 - raw_timestamp_part_2 -
cvtd_timestamp, data=training, method="rf", trControl=trainControl(method="cv", number=3))

pred2=predict(mod2, newdata=validation)
confusionMatrix(pred2, validation$classe)
```

```
Reference
Prediction    A    B    C    D    E
         A 1672    0    0    0    0
         B    1 1137    1    0    0
         C    0    2 1022    8    0
         D    0    0    3  955    0
         E    1    0    0    1 1082
```

Accuracy : 0.9971

In summary, cross validation yielded similar results to the initial model construction. One of the potential downsides to random forest models is overfitting. Therefore, I assume that I will achieve ~90% accuracy on the actual test set. Here are my predictions on the test set.

```
finalPred=predict(mod1, newdata=test)
finalPred
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
 B  A  B  A  A  E  D  B  A  A  B  C  B  A  E  E  A  B  B  B
```