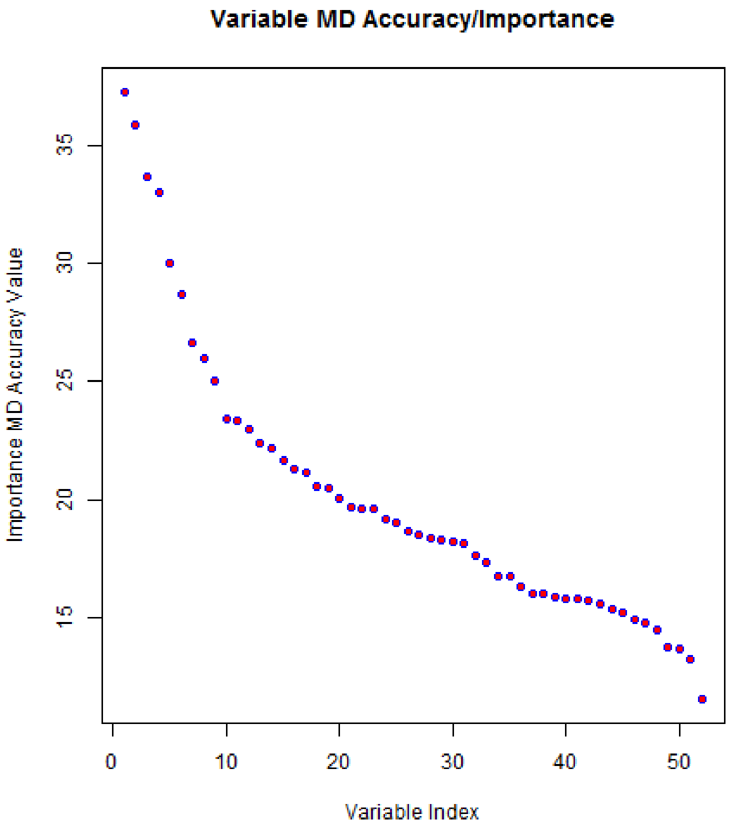
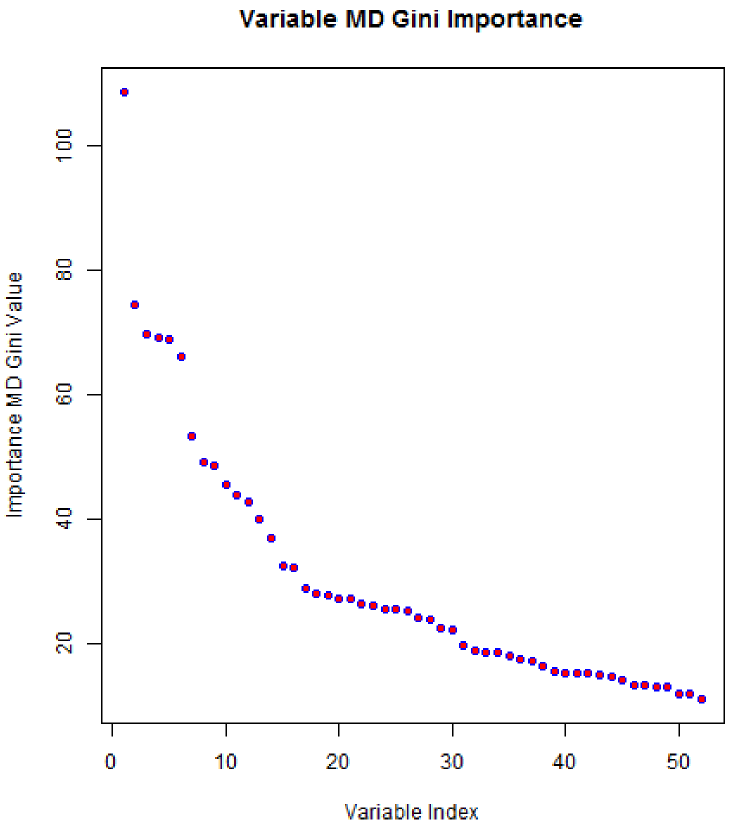
```
> t <- cpScript()
[1] ---> loading data...
[1] ...done reading data and creating data frames

[1] plotting Random Forest variable importance() metrics
[1] no clear dividing line differentiating important vs. unimportant variables
```

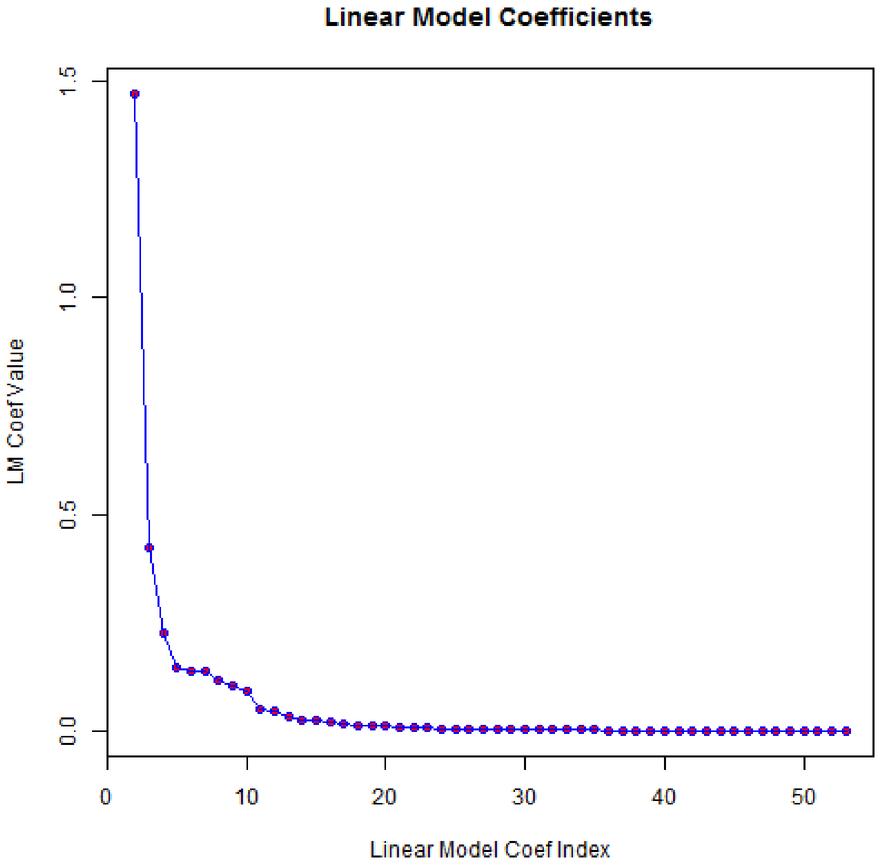


Variable MD Accuracy/Importance



Variable MD Gini Importance

```
[1] ---> assessing whether Linear Model coefficients offer useful variable importance rankings...
[1] plotting Linear Model coefficients sorted in decreasing order
```



Linear Model Coefficients

```
[1] ---> examining Random Forest performance for a range of 'mtry' and 'ntree' parameters...
[1] evaluating Random Forest w/ mtry=2 ntree=1
[1] evaluating Random Forest w/ mtry=2 ntree=2
[1] evaluating Random Forest w/ mtry=2 ntree=3
[1] evaluating Random Forest w/ mtry=2 ntree=5
[1] evaluating Random Forest w/ mtry=2 ntree=10
[1] evaluating Random Forest w/ mtry=2 ntree=20
[1] evaluating Random Forest w/ mtry=5 ntree=1
[1] evaluating Random Forest w/ mtry=5 ntree=2
[1] evaluating Random Forest w/ mtry=5 ntree=3
[1] evaluating Random Forest w/ mtry=5 ntree=5
[1] evaluating Random Forest w/ mtry=5 ntree=10
[1] evaluating Random Forest w/ mtry=5 ntree=20
[1] evaluating Random Forest w/ mtry=10 ntree=1
[1] evaluating Random Forest w/ mtry=10 ntree=2
[1] evaluating Random Forest w/ mtry=10 ntree=3
[1] evaluating Random Forest w/ mtry=10 ntree=5
[1] evaluating Random Forest w/ mtry=10 ntree=10
[1] evaluating Random Forest w/ mtry=10 ntree=20

[1] --> Random Forest accuracy values for evaluated (mtry, ntree) grid pairs:
[1] --> NOTE: row names are mtry values; column names are ntree values
[1] --> NOTE: matrix entries are classification accuracy on train set

           1         2         3         5        10        20
2  0.9548321 0.9512009 0.9906989 0.9978977 0.9997452 1.0000000
5  0.9720966 0.9693572 0.9950946 0.9984710 0.9998726 1.0000000
10 0.9684016 0.9678282 0.9948398 0.9984073 0.9998726 0.9999363

[1] --->parameters from best 52-feature Random Forest:
[1] resultant rf52: train accuracy=1.000000 mtry=2 ntree=20 OOB error=0.023445

[1] ---> printing best (rf52) Random Forest for above parameters:

Call:
 randomForest(formula = classe ~ ., data = df, mtry = mt, ntree = nt) 
               Type of random forest: classification
                     Number of trees: 20
No. of variables tried at each split: 2

        OOB estimate of  error rate: 2.34%
Confusion matrix:
     A    B    C    D    E class.error
A 4394   22    9   14    2  0.01058320
B   45 2951   32   14   10  0.03309305
C    7   45 2610   30    7  0.03297518
D   12    3   63 2480   10  0.03426791
E    2   15    4   22 2893  0.01464578

[1] ---> computing rfcv() cross-validation error - this may take several minutes...
[1] ...done computing rfcv() output
[1] rfcv() cross-validation estimates for training set vs. number variables used:
         52          26          13           6           3           1 
0.005542460 0.007581066 0.009683379 0.045104160 0.109320252 0.596929350 

[1] accuracy of best (rf52) Random Forest on 20% TEST set: 0.991847

[1] ---> As additional exercise fit Random Forest to top-20 features from Linear Model
[1] evaluating Random Forest w/ mtry=2 ntree=10
[1] evaluating Random Forest w/ mtry=2 ntree=50
[1] evaluating Random Forest w/ mtry=2 ntree=100
[1] evaluating Random Forest w/ mtry=2 ntree=200
[1] evaluating Random Forest w/ mtry=5 ntree=10
[1] evaluating Random Forest w/ mtry=5 ntree=50
[1] evaluating Random Forest w/ mtry=5 ntree=100
[1] evaluating Random Forest w/ mtry=5 ntree=200
[1] evaluating Random Forest w/ mtry=10 ntree=10
[1] evaluating Random Forest w/ mtry=10 ntree=50
[1] evaluating Random Forest w/ mtry=10 ntree=100
[1] evaluating Random Forest w/ mtry=10 ntree=200
```

```
[1] --> Random Forest accuracy values for evaluated (mtry, ntree) grid pairs:
[1] --> NOTE: row names are mtry values; column names are ntree values
[1] --> NOTE: matrix entries are classification accuracy on train set

          10 50 100 200
2  0.9997452  1   1   1
5  0.9998726  1   1   1
10 0.9998726  1   1   1

[1] --->parameters from best 20-feature Random Forest:
[1] resultant rf20: train accuracy=1.000000 mtry=2 ntree=50 OOB error=0.009110

[1] ---> printing best Random Forest for above parameters:

Call:
 randomForest(formula = classe ~ ., data = df, mtry = mt, ntree = nt)
               Type of random forest: classification
                     Number of trees: 50
No. of variables tried at each split: 2

        OOB estimate of  error rate: 0.91%
Confusion matrix:
     A    B    C    D    E class.error
A 4430    7    0    3    1 0.002476920
B   21 3020    8    1    2 0.010484928
C    3   28 2663    5    1 0.013703704
D    3    0   42 2519    4 0.019080997
E    0    3    1   10 2922 0.004768392

[1] ---> evalute top-20 feature Random Forest:
[1] accuracy of best (rf20) Random Forest on 20% TEST set: 0.994395

[1] ---> use rf52 to predict labels for course project 20-row data set (pml-testing.csv
 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 
Levels: A B C D E

[1] ---> use rf20 to predict labels for course project 20-row data set (pml-testing.csv
 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 
Levels: A B C D E
```