Practical Machine Learning - Assignment report

The assignment is asking us to predict the manner in which participant did an exercise - the "classe" variable in the training set

I first read the testing and training data:

```
training<- read.csv("pml-training.csv", na.strings=c("NA"," ","#DIV/0!"))

testing<- read.csv("pml-testing.csv", na.strings=c("NA"," ","#DIV/0!"))
```

From visual inspection of the training data I noticed that many of the input fields are actually NA, blank or equal to "#DIV/0!", therefore I used these as NA while reading the files, and then removed the unused columns

```
training<-training[, colSums(is.na(training)) != nrow(training)]
```

From inspecting the data I also noticed that rows with the parameter new_window = "yes" are rare and the data in them looks completely different, therefor I decided to remove these lines from the training data as well

```
training<-training[training$new_window =="no",] ## clean rows with strange data
```

Some more inspection indicated that the first few columns are less relevant therefore I removed them as well

```
training<-subset( training, select = -c(X,user_name,raw_timestamp_part_1,raw_timestamp_part_2,cvtd_timestamp,new_window))
```

I then created a training and testing subset in the data:

```
inTrain <- createDataPartition(y=training$classe,p=0.7, list=FALSE)

traininggroup <- training[inTrain,]

testinggroup <- training[-inTrain,]
```

I then used the Caret package and its Train function, using Random Forest algorithm to predict the Classe parameter

```
tr1<-train(traininggroup$classe ~ ., data = traininggroup, method = "rf")
```

The result were as follows:

```
Random Forest

13453 samples
   54 predictor
    5 classes: 'A', 'B', 'C', 'D', 'E'
```

```
No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 13453, 13453, 13453, 13453, 13453, 13453
, ...
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa      Accuracy SD  Kappa SD
   2    0.9911399  0.9887894  0.001793225  0.002270393
  28    0.9956940  0.9945522  0.001316920  0.001665525
  54    0.9919410  0.9898052  0.002631727  0.003324420

Accuracy was used to select the optimal model using  the largest
value.
The final value used for the model was mtry = 28.
```

I used the (internal) testing group to validate the model:

```
confusionMatrix(testinggroup$classe, predict(tr1, testinggroup))
```

Which resulted with the following accuracy of 0.9986:

		Reference				
Prediction		A	B	C	D	E
	A	1640	0	0	0	1
	B	1	1114	0	0	0
	C	0	2	1003	0	0
	D	0	0	3	941	0
	E	0	0	0	1	1057

```
Overall Statistics

               Accuracy : 0.9986
                 95% CI : (0.9973, 0.9994)
    No Information Rate : 0.2847
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9982
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: A Class: B Class: C Class: D Class: E
Sensitivity            0.9994   0.9982   0.9970   0.9989   0.9991
Specificity            0.9998   0.9998   0.9996   0.9994   0.9998
Pos Pred Value         0.9994   0.9991   0.9980   0.9968   0.9991
Neg Pred Value         0.9998   0.9996   0.9994   0.9998   0.9998
Prevalence             0.2847   0.1936   0.1746   0.1635   0.1836
Detection Rate         0.2846   0.1933   0.1740   0.1633   0.1834
Detection Prevalence   0.2847   0.1935   0.1744   0.1638   0.1836
```

```
Balanced Accuracy      0.9996   0.9990   0.9983   0.9992   0.9994
```

Finally I used the prediction model on the test cases

```
predict(tr1, testing)
```

And got the following results

```
[1] 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
```

Further analyzing the reulst I was looking for the most relevant parameters in the model:

```
varImp(tr1)
```

Which had the following results:

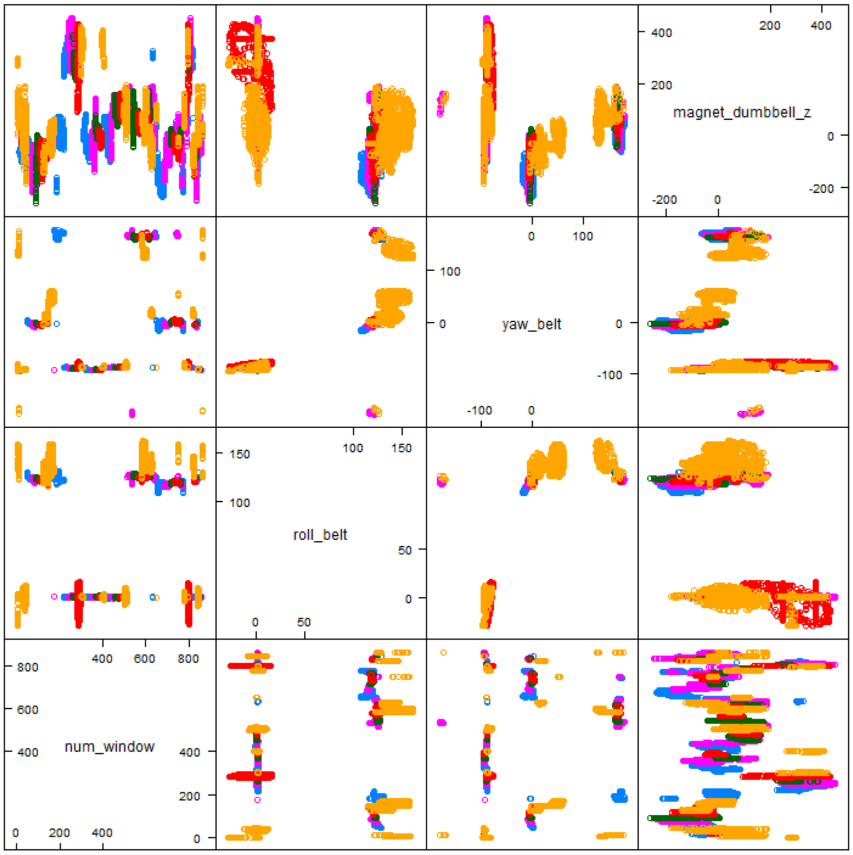
```
rf variable importance

  only 20 most important variables shown (out of 53)

                     Overall
num_window           100.000
roll_belt             60.683
pitch_forearm         36.901
yaw_belt              30.530
magnet_dumbbell_z     29.093
pitch_belt            28.035
magnet_dumbbell_y     27.036
roll_forearm          21.887
accel_dumbbell_y      12.423
roll_dumbbell         10.856
magnet_dumbbell_x     10.511
accel_forearm_x        9.847
accel_belt_z           9.292
total_accel_dumbbell   8.647
accel_dumbbell_z       7.412
magnet_belt_y          7.130
magnet_belt_z          6.792
magnet_forearm_z       6.601
magnet_belt_x          5.686
roll_arm               4.819
```

I than draw the realtions between the most significant parameters:

```
featurePlot(x=training[,c("num_window","roll_belt","yaw_belt","magnet_dumbbell_z")],
y = training$classe,plot="pairs")
```



Scatter Plot Matrix