Exploratory Analysis

```
## Loading required package: Rcpp
## ##
## ## Amelia II: Multiple Imputation
## ## (Version 1.7.3, built: 2014-11-14)
## ## Copyright (C) 2005-2015 James Honaker, Gary King and Matthew Blackwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
## ##
##
## Attaching package: 'dplyr'
##
## The following objects are masked from 'package:stats':
##
##     filter, lag
##
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
##
## Loading required package: lattice
## Loading required package: foreach
## Loading required package: iterators
## Loading required package: parallel
```

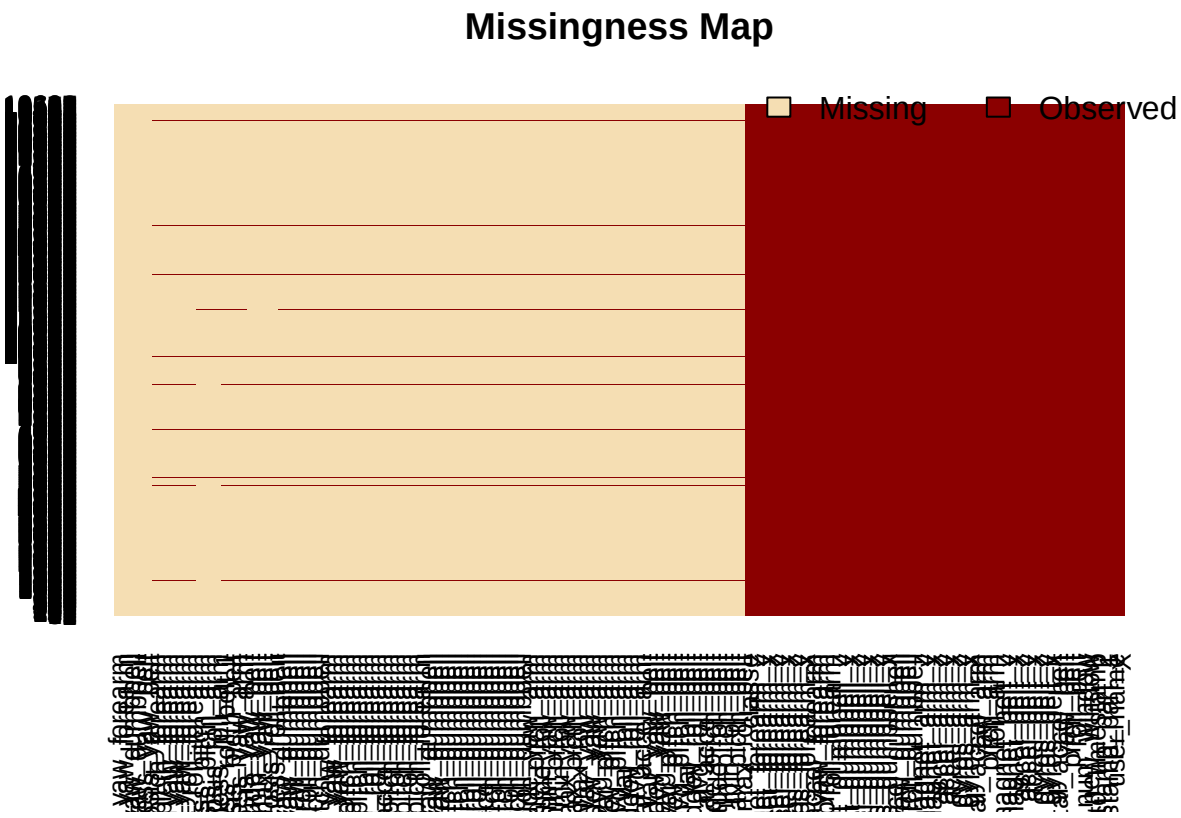
Lets read raw data

```
train_file <- "pml-training.csv"
test_file <- "pml-testing.csv"

train_raw <- read.csv(train_file, na.strings=c("NA", "", "#DIV/0!"))
test_raw <- read.csv(test_file, na.strings=c("NA", "", "#DIV/0!"))
```

What is missing:

```
missmap(train_raw)
```



Data Cleaning

We need to remove features with many NAs.

```
all_features <- names(train_raw)
num_train_cases <- nrow(train_raw)

mostly_na_features <- sapply(all_features, function(x) {
    sum(is.na(train_raw[, x])) > num_train_cases*0.9
})

non_na_features <- all_features[ ! mostly_na_features]
features <- setdiff(non_na_features, c("X", "classe"))

train_no_na <- train_raw[ , c(features, "classe")]
test_no_na <- test_raw[ , c(features, "problem_id")]
dim(train_no_na)
```

```
## [1] 19622    59
```

```
dim(test_no_na)
```

```
## [1] 20 59
```

Model Building

Lets split data into test(to estimate out of sample error) and training dataset.

```
in_train <- createDataPartition(train_no_na$classe, p=0.7, list=FALSE)
train_set <- train_no_na[in_train, ]
out_of_sample <- train_no_na[-in_train, ]
```

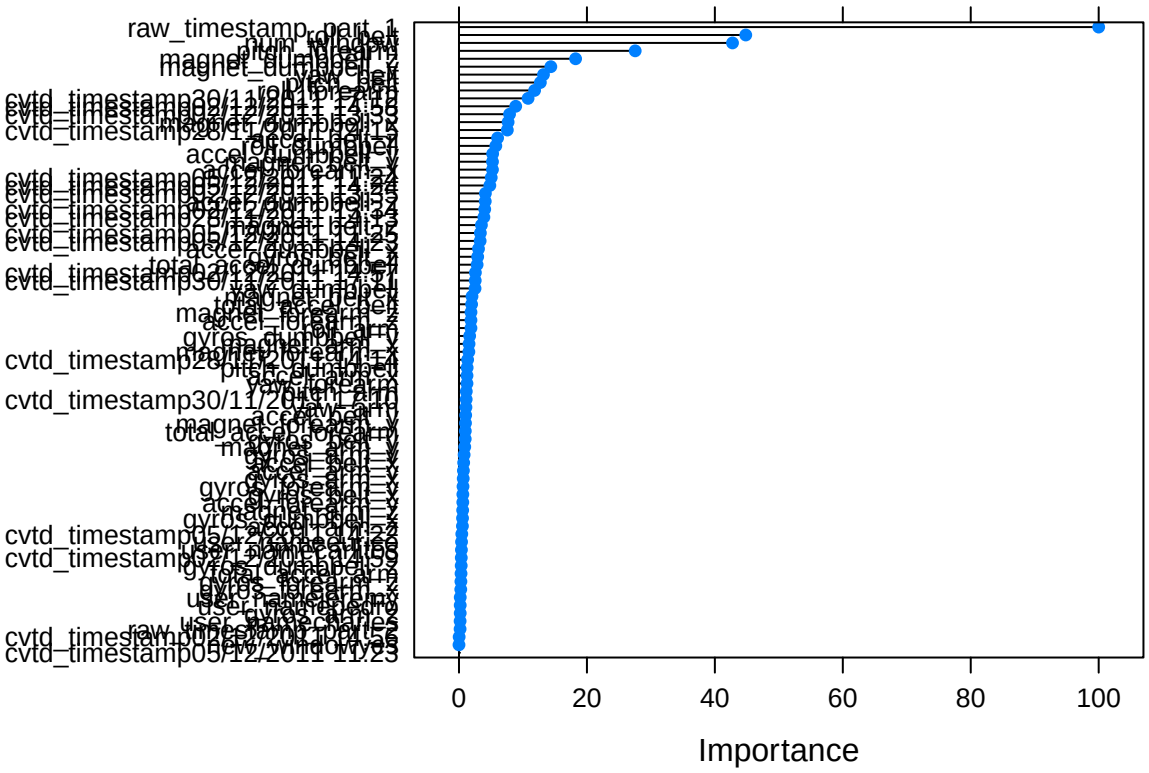
Lets start with simple random forest:

```
control <- trainControl("repeatedcv", repeats=3)
rf_model <- train(classe ~ ., data=train_set, method="rf",
    trControl=control)
```

```
## Loading required package: randomForest
## randomForest 4.6-12
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
##
## The following object is masked from 'package:dplyr':
##
##     combine
```

Feature Importance

```
plot(varImp(rf_model))
```



Out of Sample Error

Lets estimate error on unseen examples, but for the cases when we know truth:

```
out_of_sample_prediction <- predict(rf_model, newdata=out_of_sample)
confusionMatrix(data=out_of_sample_prediction, out_of_sample$classe)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1674    1    0    0    0
##          B    0 1136    1    0    0
##          C    0    2 1025    0    0
##          D    0    0    0  963    0
##          E    0    0    0    1 1082
##
## Overall Statistics
##
##                Accuracy : 0.9992
##                  95% CI : (0.998, 0.9997)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9989
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   0.9974   0.9990   0.9990   1.0000
## Specificity            0.9998   0.9998   0.9996   1.0000   0.9998
## Pos Pred Value         0.9994   0.9991   0.9981   1.0000   0.9991
## Neg Pred Value         1.0000   0.9994   0.9998   0.9998   1.0000
## Prevalence             0.2845   0.1935   0.1743   0.1638   0.1839
## Detection Rate         0.2845   0.1930   0.1742   0.1636   0.1839
## Detection Prevalence   0.2846   0.1932   0.1745   0.1636   0.1840
## Balanced Accuracy      0.9999   0.9986   0.9993   0.9995   0.9999
```

So the model that we build is highly accurate $> 98\%$.

Prediction

```
pml_write_files = function(x){
  n = length(x)
  for(i in 1:n){
    filename = paste0("problem_id_",i,".txt")
    write.table(x[i],file=filename,quote=FALSE,row.names=FALSE,col.names=FALSE)
  }
}
```

```
prediction <- predict(rf_model, newdata=test_no_na)
pml_write_files(prediction)
```