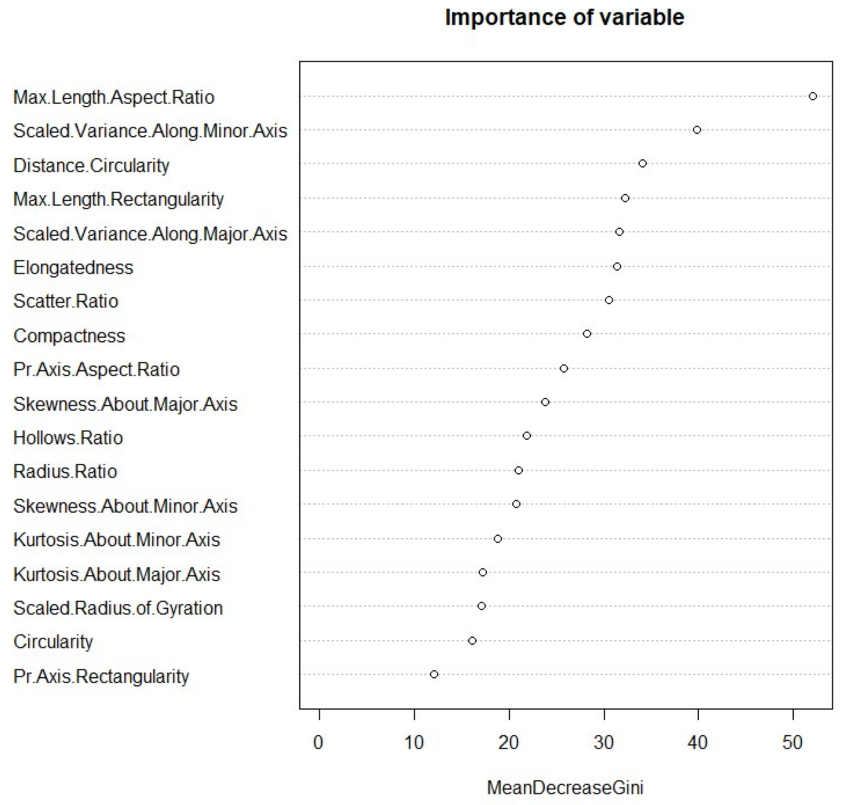
2. Fit a default random forest

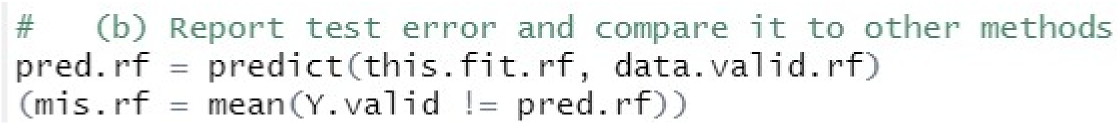
(a) **Report variable importance and comment: which variables seem most important and which seem least important?**

```
### The function that fits random forests requires that our response
### variable be a factor. We need to make a copy of our dataset and
### use the factor() function on quality.
data.rf = vehdata
data.rf$class = factor(data.rf$class)

### Split the data into training and validation sets
data.train.rf = data.rf[which ( perm <= 3* nrow ( vehdata )/4) , ]
data.valid.rf = data.rf[which ( perm > 3* nrow ( vehdata )/4) , ]
Y.train.rf = data.train.rf$class
Y.valid.rf = data.valid.rf$class



this.fit.rf = randomForest(class ~ ., data = data.train.rf)
importance(this.fit.rf)
varImpPlot(this.fit.rf, main="Importance of variable")
```

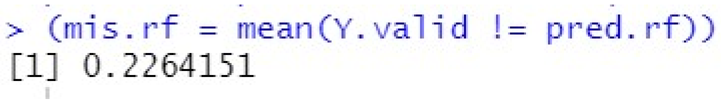


Max.Length.Aspect.Ratio looks most important and Pr.Axis.Rectangularity seems less important.

(b) **Report test error and compare it to other methods**



```
#   (b) Report test error and compare it to other methods
pred.rf = predict(this.fit.rf, data.valid.rf)
(mis.rf = mean(Y.valid != pred.rf))
```



```
> (mis.rf = mean(Y.valid != pred.rf))
[1] 0.2264151
```

It shows quite lower error compare to classification tree and Naïve Bayes.