In [1]:
```
library(caret)
```

```
Warning message:
"package 'caret' was built under R version 3.6.3"Loading required packa
ge: lattice
Loading required package: ggplot2
Warning message:
"package 'ggplot2' was built under R version 3.6.3"
```

In [2]:
```
library(rpart)
```

In [3]:
```
library(rpart.plot)
```

```
Error in library(rpart.plot): there is no package called 'rpart.plot'
Traceback:

1. library(rpart.plot)
```

In [4]:
```
install.packages("rpart.plot", repos='http://cran.us.r-project.org')
```

```
package 'rpart.plot' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Acer\AppData\Local\Temp\RtmpgFPMMX\downloaded_packages
```

In [5]:
```
library(rpart.plot)
```

```
Warning message:
"package 'rpart.plot' was built under R version 3.6.3"
```

In [6]:
```
library(rpart.plot)
```

```
Warning message:
"package 'randomForest' was built under R version 3.6.3"randomForest 4.
6-14
```

```
Type rfNews() to see new features/changes/bug fixes.

Attaching package: 'randomForest'

The following object is masked from 'package:ggplot2':

    margin
```

In [7]:
```
library(corrplot)
```

```
Error in library(corrplot): there is no package called 'corrplot'
Traceback:

1. library(corrplot)
```

In [8]:
```
install.packages("corrplot", repos='http://cran.us.r-project.org')
```

```
package 'corrplot' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Acer\AppData\Local\Temp\RtmpgFPMMX\downloaded_packages
```

In [9]:
```
library(corrplot)
```

```
Warning message:
"package 'corrplot' was built under R version 3.6.3"corrplot 0.84 loade
d
```

In [10]:
```
trainingURL <-"https://d396qusza40orc.cloudfront.net/predmachlearn/pml-
training.csv"
testingURL <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-
testing.csv"
trainingFile <- "./data/pml-training.csv"
testingFile  <- "./data/pml-testing.csv"
if (!file.exists("./data")) {
  dir.create("./data")
}
```

```
if (!file.exists(trainingFile)) {
  download.file(trainingURL, destfile=trainingFile, method="curl")
}
if (!file.exists(testingFile)) {
  download.file(testingURL, destfile=testingFile, method="curl")
}
```

In [11]:
```
trainRaw <- read.csv("./data/pml-training.csv")
testRaw <- read.csv("./data/pml-testing.csv")
dim(trainRaw)
```

```
19622  160
```

In [12]:
```
dim(testRaw)
```

```
20  160
```

In [13]:
```
sum(complete.cases(trainRaw))
```

```
406
```

In [14]:
```
trainRaw <- trainRaw[, colSums(is.na(trainRaw)) == 0]
```

In [15]:
```
testRaw <- testRaw[, colSums(is.na(testRaw)) == 0]
```

In [16]:
```
classe <- trainRaw$classe
trainRemove <- grepl("^X|timestamp|window", names(trainRaw))
trainRaw <- trainRaw[, !trainRemove]
trainCleaned <- trainRaw[, sapply(trainRaw, is.numeric)]
trainCleaned$classe <- classe
testRemove <- grepl("^X|timestamp|window", names(testRaw))
testRaw <- testRaw[, !testRemove]
testCleaned <- testRaw[, sapply(testRaw, is.numeric)]
```

In [17]:
```
set.seed(2108) # For reproducibile purpose
inTrain <- createDataPartition(trainCleaned$classe, p=0.70, list=F)
```

```
trainData <- trainCleaned[inTrain, ]
testData <- trainCleaned[-inTrain, ]
```

In [18]:
```
controlRf <- trainControl(method="cv", 5)
modelRf <- train(classe ~ ., data=trainData, method="rf", trControl=con
trolRf, ntree=250)
modelRf
```

```
Error: package e1071 is required
Traceback:

1. train(classe ~ ., data = trainData, method = "rf", trControl = contr
olRf,
 .     ntree = 250)
2. train.formula(classe ~ ., data = trainData, method = "rf", trControl
= controlRf,
 .     ntree = 250)
3. train(x, y, weights = w, ...)
4. train.default(x, y, weights = w, ...)
5. evalSummaryFunction(y, wts = weights, ctrl = trControl, lev = classL
evels,
 .     metric = metric, method = method)
6. ctrl$summaryFunction(testOutput, lev, method)
7. postResample(data[, "pred"], data[, "obs"])
8. requireNamespaceQuietStop("e1071")
9. stop(paste("package", package, "is required"), call. = FALSE)
```

In [19]:
```
install.packages("e1071", repos='http://cran.us.r-project.org')
```

```
package 'e1071' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Acer\AppData\Local\Temp\RtmpgFPMMX\downloaded_packages
```

In [20]:
```
controlRf <- trainControl(method="cv", 5)
modelRf <- train(classe ~ ., data=trainData, method="rf", trControl=con
trolRf, ntree=250)
modelRf
```

```
Random Forest

13737 samples
   52 predictor
    5 classes: 'A', 'B', 'C', 'D', 'E'

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 10989, 10989, 10990, 10990, 10990
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
   2    0.9892259  0.9863692
  27    0.9913372  0.9890413
  52    0.9826747  0.9780829

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 27.
```

In [21]:
```
predictRf <- predict(modelRf, testData)
confusionMatrix(testData$classe, predictRf)
```

```
Confusion Matrix and Statistics

          Reference
Prediction    A    B    C    D    E
         A 1671    1    1    0    1
         B   13 1124    2    0    0
         C    0    3 1022    1    0
         D    0    0    7  957    0
         E    0    0    4    3 1075

Overall Statistics

               Accuracy : 0.9939
                 95% CI : (0.9915, 0.9957)
    No Information Rate : 0.2862
    P-Value [Acc > NIR] : < 2.2e-16
```

```
                  Kappa : 0.9923

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: A Class: B Class: C Class: D Class: E
Sensitivity            0.9923   0.9965   0.9865   0.9958   0.9991
Specificity            0.9993   0.9968   0.9992   0.9986   0.9985
Pos Pred Value         0.9982   0.9868   0.9961   0.9927   0.9935
Neg Pred Value         0.9969   0.9992   0.9971   0.9992   0.9998
Prevalence             0.2862   0.1917   0.1760   0.1633   0.1828
Detection Rate         0.2839   0.1910   0.1737   0.1626   0.1827
Detection Prevalence   0.2845   0.1935   0.1743   0.1638   0.1839
Balanced Accuracy      0.9958   0.9967   0.9928   0.9972   0.9988
```

In [22]:
```
accuracy <- postResample(predictRf, testData$classe)
accuracy
```

Accuracy	0.993882752761257
Kappa	0.992261060800132

In [23]:
```
oose <- 1 - as.numeric(confusionMatrix(testData$classe, predictRf)$overall[1])
oose
```

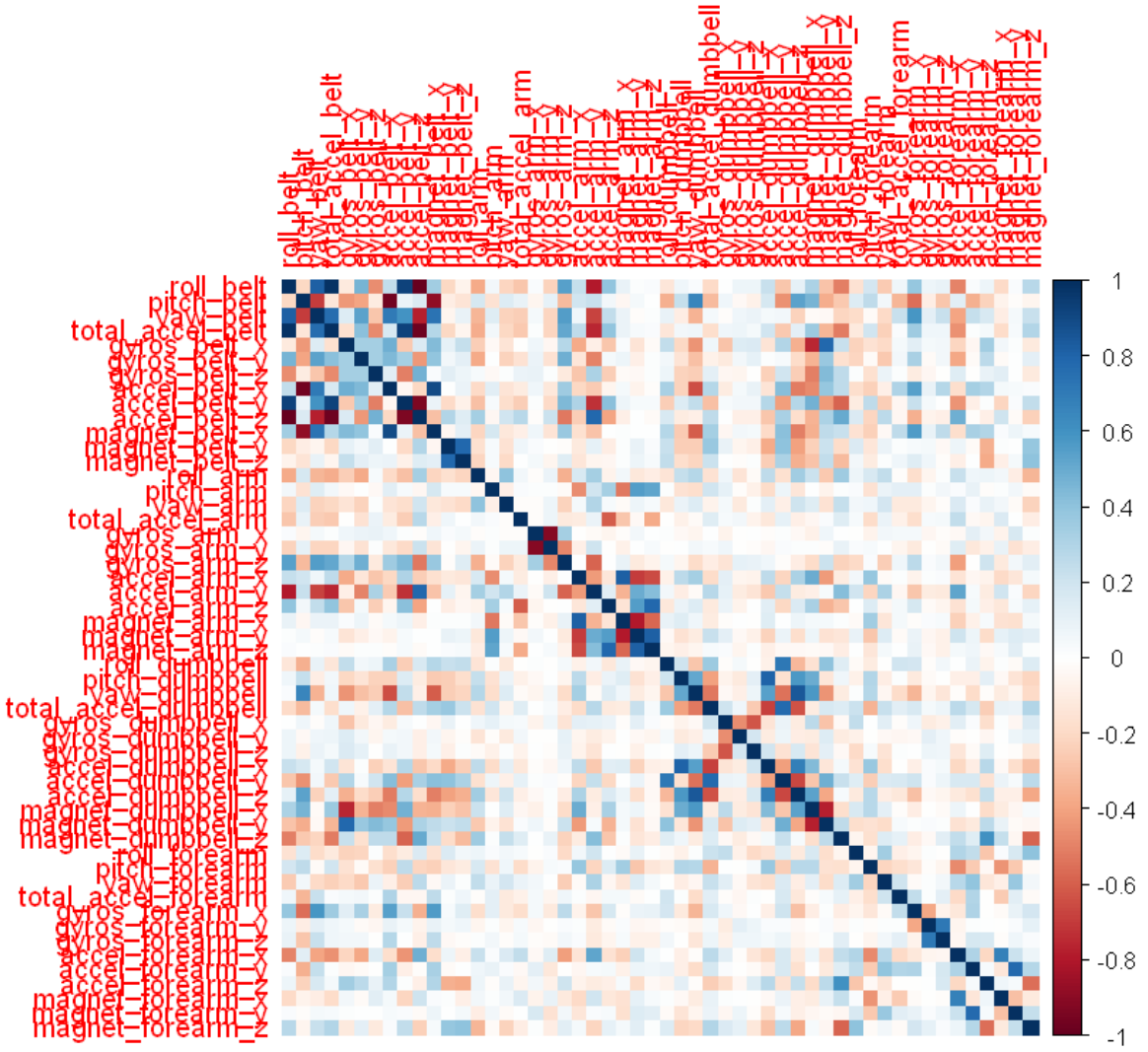
0.00611724723874252

In [24]:
```
result <- predict(modelRf, testCleaned[, -length(names(testCleaned))])
result
```

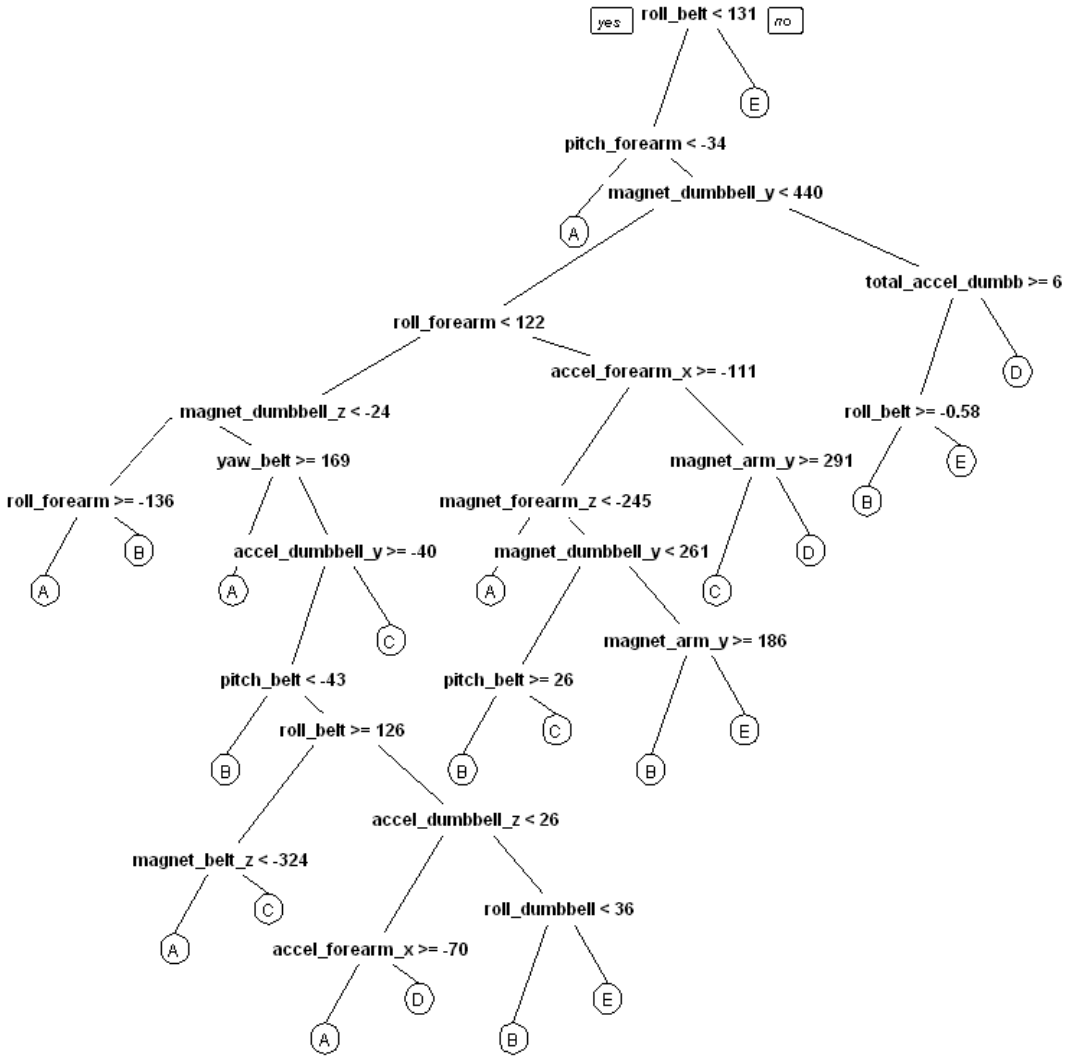
B A B A A E D B A A B C B A E E A B B B

▶ **Levels**:

In [25]:
```
corrPlot <- cor(trainData[, -length(names(trainData))])
corrplot(corrPlot, method="color")
```



In [26]:
```
treeModel <- rpart(classe ~ ., data=trainData, method="class")
prp(treeModel)
```



In [ ]: