Course 8 project

R Markdown

The goal of your project is to predict the manner in which people did the exercise. the data is provided by: http://web.archive.org/web/20161224072740/http:/groupware.les.inf.puc-rio.br/har

data load and preprocessing

below I load the data. I see that there are a lot of NA's within variables.

data load and preprocessing

there are still 60 variables which is a lot. To see if this could be reduced I perform a test for zero covariates.

```
##    freqRatio       percentUnique        zeroVar           nzv
##  Min.   : 1.000   Min.   :  0.01019   Mode :logical   Mode :logical
##  1st Qu.: 1.022   1st Qu.:  1.25497   FALSE:60        FALSE:59
##  Median : 1.069   Median :  3.36102                   TRUE :1
##  Mean   : 6.192   Mean   : 11.08467
##  3rd Qu.: 1.135   3rd Qu.:  7.96937
##  Max.   :87.256   Max.   :100.00000
```

apply on test set

these changes are aso performed with the test dataset

create training and validation set

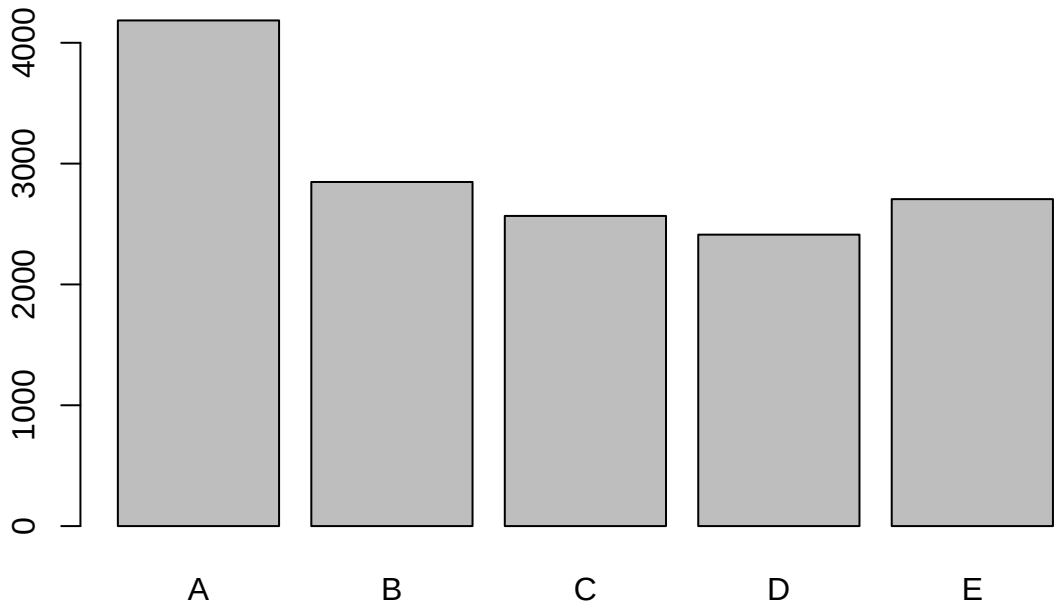
the dataset is seperated into 2 dataset, the training dataset is used to train different models. The validation dataset is used to estimate how accurate the different models are for out sample errors.

```
set.seed(22)
inTrainIndex <- createDataPartition(df_train$classe, p=0.75)[[1]]
df_training <- df_train[inTrainIndex,]
df_validation <- df_train[-inTrainIndex,]
```

exploration

before perfroming Machine learning models, I do some explorative analysis to get to know the data

```
library(ggplot2)
df_training$classe <- as.factor(df_training$classe)
plot(df_training$classe)
```



most of the datapoints are assigned to classe A. but it seems no problem for further analysis because there are no rare events.

train machine learning models

to start i perform a " normal " decision tree. decision tree

```
decisiontree <- train(classe ~., data= df_training, method="rpart", trControl=trainControl("cv", number=
decisiontree
```

```
## CART
##
## 14718 samples
##    58 predictor
##     5 classes: 'A', 'B', 'C', 'D', 'E'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 13247, 13246, 13247, 13246, 13246, 13246, ...
## Resampling results across tuning parameters:
##
##   cp          Accuracy   Kappa
##   0.00000000  0.9997283  0.99965630
##   0.03004314  0.9997283  0.99965630
##   0.06008629  0.9997283  0.99965630
##   0.09012943  0.9997283  0.99965630
##   0.12017258  0.9997283  0.99965630
##   0.15021572  0.9997283  0.99965630
```

```
##   0.18025887  0.9997283  0.99965630
##   0.21030201  0.9997283  0.99965630
##   0.24034516  0.8359832  0.79209417
##   0.27038830  0.3229458  0.06623846
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was cp = 0.210302.
```

```
dt_predict <- predict(decisiontree, newdata = df_validation)
df_validation$classe <- as.factor(df_validation$classe)
confusionMatrix(df_validation$classe, dt_predict)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    A    B    C    D    E
##          A 1395    0    0    0    0
##          B    0  949    0    0    0
##          C    0    0  855    0    0
##          D    0    0    0  804    0
##          E    0    0    0    0  901
##
## Overall Statistics
##
##                Accuracy : 1
##                  95% CI : (0.9992, 1)
##     No Information Rate : 0.2845
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: A Class: B Class: C Class: D Class: E
## Sensitivity            1.0000   1.0000   1.0000   1.0000   1.0000
## Specificity            1.0000   1.0000   1.0000   1.0000   1.0000
## Pos Pred Value         1.0000   1.0000   1.0000   1.0000   1.0000
## Neg Pred Value         1.0000   1.0000   1.0000   1.0000   1.0000
## Prevalence             0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Rate         0.2845   0.1935   0.1743   0.1639   0.1837
## Detection Prevalence   0.2845   0.1935   0.1743   0.1639   0.1837
## Balanced Accuracy      1.0000   1.0000   1.0000   1.0000   1.0000
```

The accurancy of the final model on the training set is 99% the accurancy is really high, be aware of overfitting. the acuurancy of the final mode on the validation set is 99%

model selection

because the accurancy of the mdel is high for out of sample, it seems to be a good prediction model. therefor i did not perform any other methods. There is no cross validation performed due to high compution time.

apply model to test set

apply model to the test set to make predicitons

```r
test_predict <- predict(decisiontree, newdata= df_test)
summary(test_predict)
```

```
##  A  B  C  D  E
## 20  0  0  0  0
```