Final Report

Cheolmin Hwang

3/11/2021

Introduction

In the final project for Math 189, we will analyze the Swiss bank notes dataset. The objective is to answer the question: Can we predit whether a note is false or counterfeit using supervised learning? We will attempt to answer this question by using techniques and tools learned from lectures 13 through 24. We will implement K-fold cross-validation method. On each fold, we will train Linear Discriminant Analysis (LDA) classifier and logistic regression classifier and look at their accuracies for each fold to decided on a better method of classification. After that, we will repeat the K-fold cross-validation on the dataset that has been preprocessed: factor analysis through maximum likelihood estimation and see its effects on the accuracies of the two models in each fold of the cross-validation.

Body

Data

The dataset was acquired from the course repository (https://github.com/tuckermcelroy/ma189/blob/main/Data/SBN.txt (https://github.com/tuckermcelroy/ma189/blob/main/Data/SBN.txt)), which was originally extracted from Flury, B. and Riedwyl, H. (1988). Multivariate Statistics: A practical approach. London: Chapman & Hall, Tables 1.1 and 1.2, pp. 5-8.

In our data, we have a total of 200 observations of old 1000-franc Swiss bank notes, where 100 of them are genuine Swiss bank notes and the other 100 are counterfeit. Each observation contains six variables measured of the bank notes:

1. Length of the note
2. Width of the Left-Hand side of the note
3. Width of the Right-Hand side of the note
4. Width of the Bottom Margin
5. Width of the Top Margin
6. Diagonal Length of Printed Area

```
notes <- read.table('C:\\Users\\cheol\\Repository\\ma189\\Data\\SBN.txt')
colnames(notes) <- c('Length', 'Left', 'Right', 'Bottom Margin', 'Top Margin', 'Diagonal')
head(notes)
```

```
##     Length  Left Right Bottom Margin Top Margin Diagonal
## BN1  214.8 131.0 131.1           9.0        9.7    141.0
## BN2  214.6 129.7 129.7           8.1        9.5    141.7
## BN3  214.8 129.7 129.7           8.7        9.6    142.2
## BN4  214.8 129.7 129.6           7.5       10.4    142.0
## BN5  215.0 129.6 129.7          10.4        7.7    141.8
## BN6  215.7 130.8 130.5           9.0       10.1    141.4
```

Source: Flury, B. and Riedwyl, H. (1988). Multivariate Statistics: A practical approach. London: Chapman & Hall, Tables 1.1 and 1.2, pp. 5-8.

We know from lecture that observations with index BN1 to BN100 are genuine banknotes and that observations with index BN101 to 200 are counterfeit banknotes. So we can divide them and show separate basic statistics and visualizations separately.We will divide the dataset into genuine bank notes and counterfeit bank notes.

Here are the sample means and variance matrices of each genuine bank notes and counterfeit bank notes:

```
genuine_sbn <- notes[1:100,]
counterfeit_sbn <- notes[101:200,]

sbn_mat <- cbind(colMeans(genuine_sbn), colMeans(counterfeit_sbn))
colnames(sbn_mat) <- c("Genuine Sample Mean","Counterfeit Sample Mean")
sbn_mat
```

```
##                Genuine Sample Mean Counterfeit Sample Mean
## Length                     214.969                 214.823
## Left                       129.943                 130.300
## Right                      129.720                 130.193
## Bottom Margin                8.305                  10.530
## Top Margin                  10.168                  11.133
## Diagonal                   141.517                 139.450
```

```
var(notes[1:100,])
```

```
##                    Length        Left       Right Bottom Margin  Top Margin
## Length        0.150241414  0.05801313  0.05729293  0.0571262626  0.01445253
## Left          0.058013131  0.13257677  0.08589899  0.0566515152  0.04906667
## Right         0.057292929  0.08589899  0.12626263  0.0581818182  0.03064646
## Bottom Margin 0.057126263  0.05665152  0.05818182  0.4132070707 -0.26347475
## Top Margin    0.014452525  0.04906667  0.03064646 -0.2634747475  0.42118788
## Diagonal      0.005481818 -0.04306162 -0.02377778 -0.0001868687 -0.07530909
##                   Diagonal
## Length         0.0054818182
## Left          -0.0430616162
## Right         -0.0237777778
## Bottom Margin -0.0001868687
## Top Margin    -0.0753090909
## Diagonal       0.1998090909
```

```
var(notes[101:200,])
```

```
##                   Length         Left         Right Bottom Margin
## Length         0.12401111  0.031515152  0.0240010101   -0.10059596
## Left           0.03151515  0.065050505  0.0467676768   -0.02404040
## Right          0.02400101  0.046767677  0.0889404040   -0.01857576
## Bottom Margin -0.10059596 -0.024040404 -0.0185757576    1.28131313
## Top Margin     0.01943535 -0.011919192  0.0001323232   -0.49019192
## Diagonal       0.01156566 -0.005050505  0.0341919192    0.23848485
##                 Top Margin     Diagonal
## Length         0.0194353535  0.011565657
## Left          -0.0119191919 -0.005050505
## Right          0.0001323232  0.034191919
## Bottom Margin -0.4901919192  0.238484848
## Top Margin     0.4044555556 -0.022070707
## Diagonal      -0.0220707071  0.311212121
```

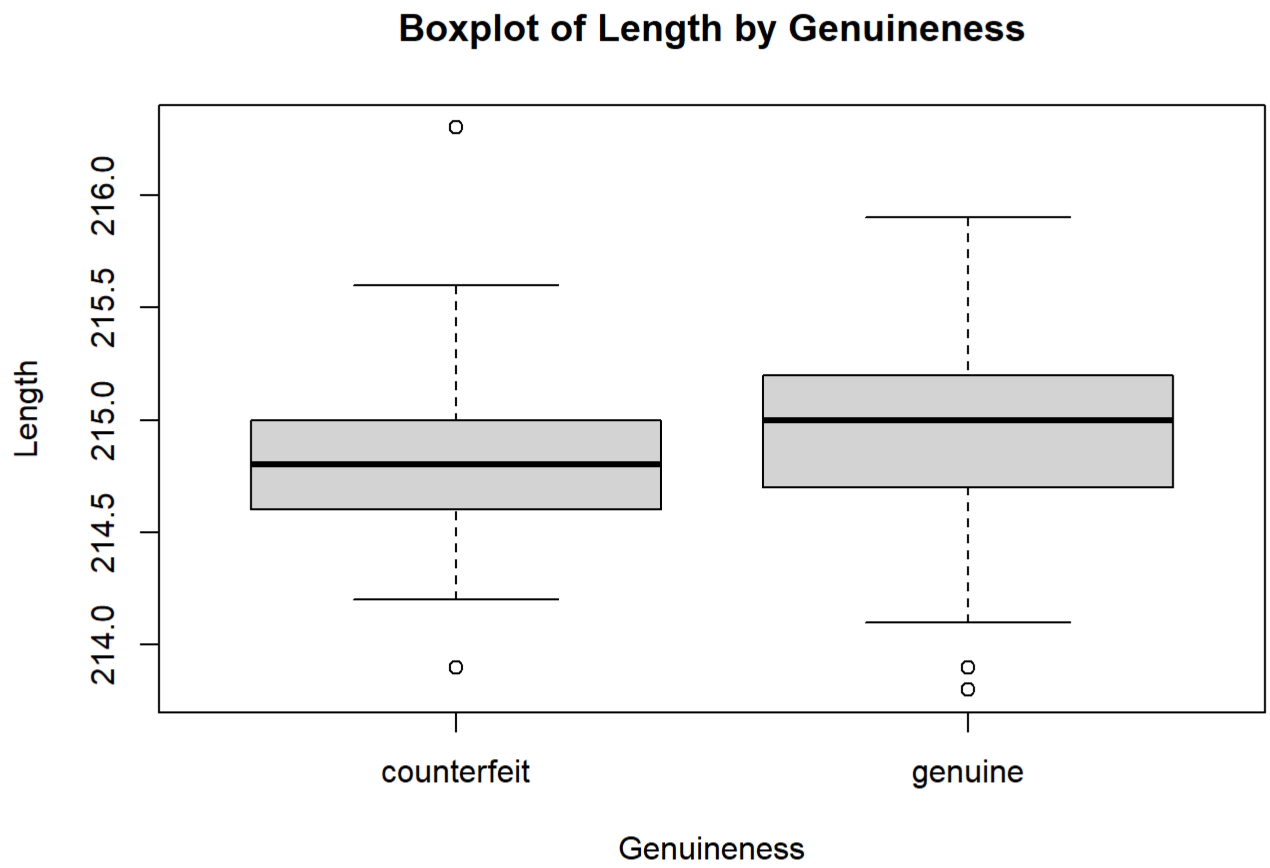
To show and indication for whether an observation is of a genuine bank note or a counterfeit bank note, we will add a column for indication.

```
Indicator <- c()

for(count in 1:100){
  Indicator <- c(Indicator, 'genuine')
}

for(count in 1:100){
  Indicator <- c(Indicator, 'counterfeit')
}

notes_indc <- cbind(notes, Indicator)
head(notes_indc)
```

```
##     Length  Left Right Bottom Margin Top Margin Diagonal Indicator
## BN1  214.8 131.0 131.1           9.0        9.7    141.0   genuine
## BN2  214.6 129.7 129.7           8.1        9.5    141.7   genuine
## BN3  214.8 129.7 129.7           8.7        9.6    142.2   genuine
## BN4  214.8 129.7 129.6           7.5       10.4    142.0   genuine
## BN5  215.0 129.6 129.7          10.4        7.7    141.8   genuine
## BN6  215.7 130.8 130.5           9.0       10.1    141.4   genuine
```

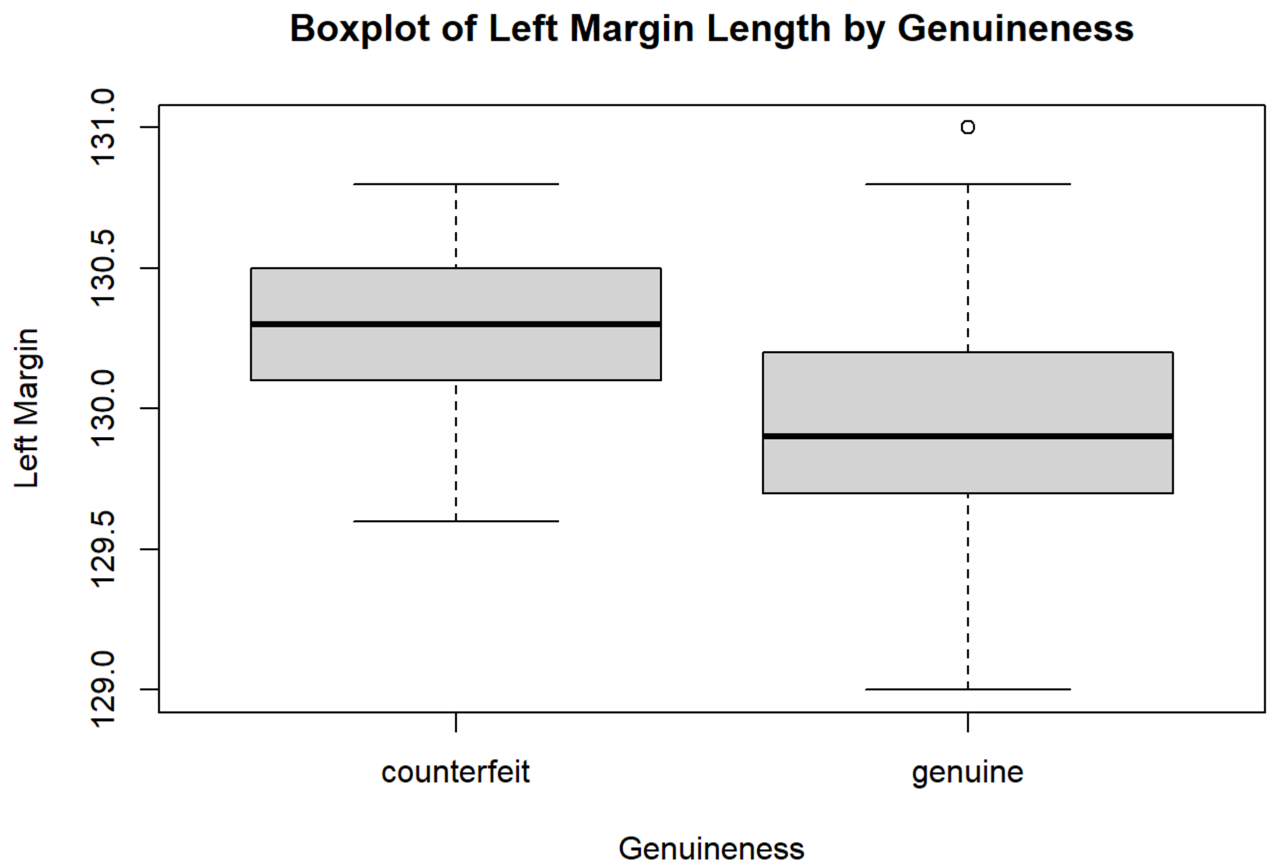
Here are the comparisons between the two groups visualized:

```
boxplot(notes_indc$Length ~ notes_indc$Indicator, xlab = 'Genuineness', ylab = 'Length')
title('Boxplot of Length by Genuineness')
```



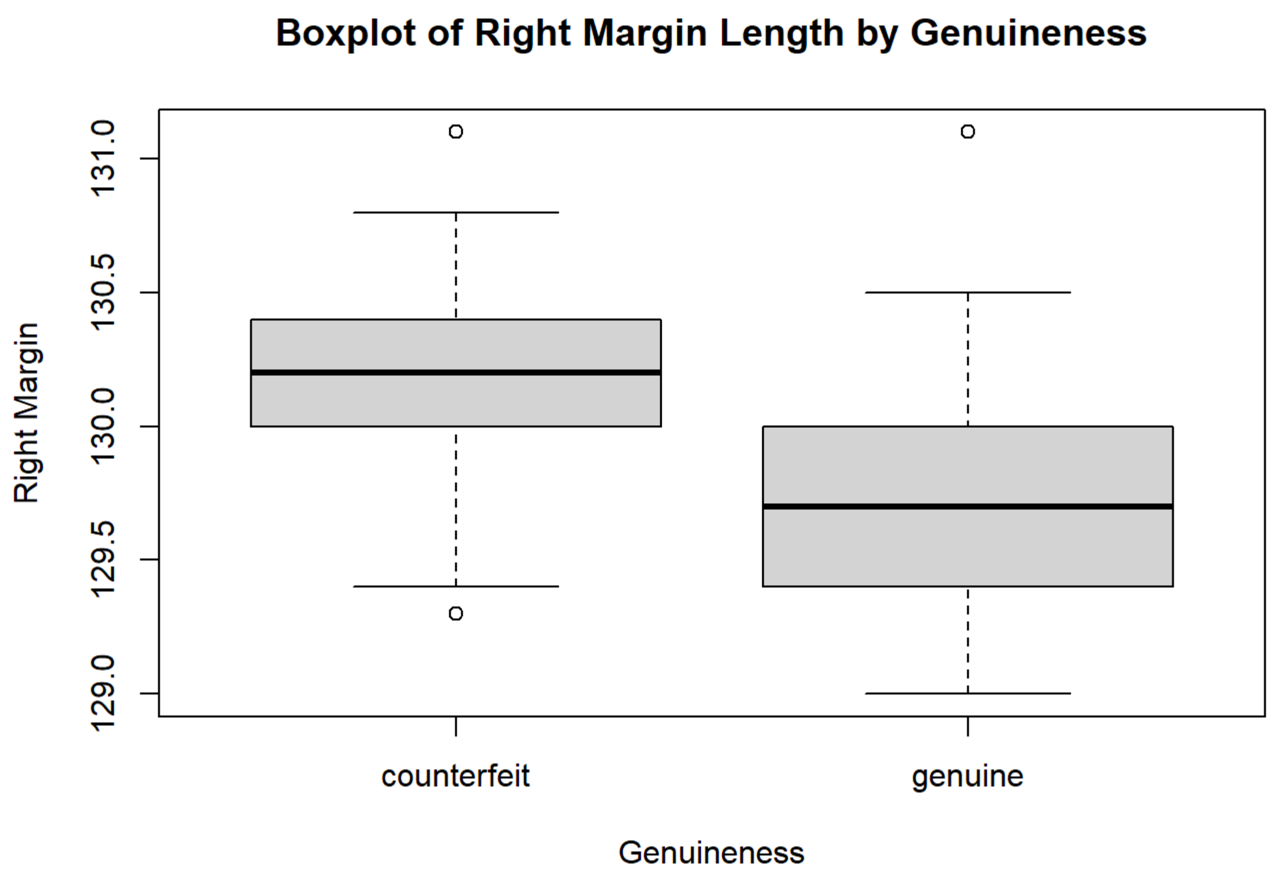
We can observe that the Length of geuine notes is slightly higher.

```
boxplot(notes_indc$Left ~ notes_indc$Indicator, xlab = 'Genuineness', ylab = 'Left Margin')
title('Boxplot of Left Margin Length by Genuineness')
```



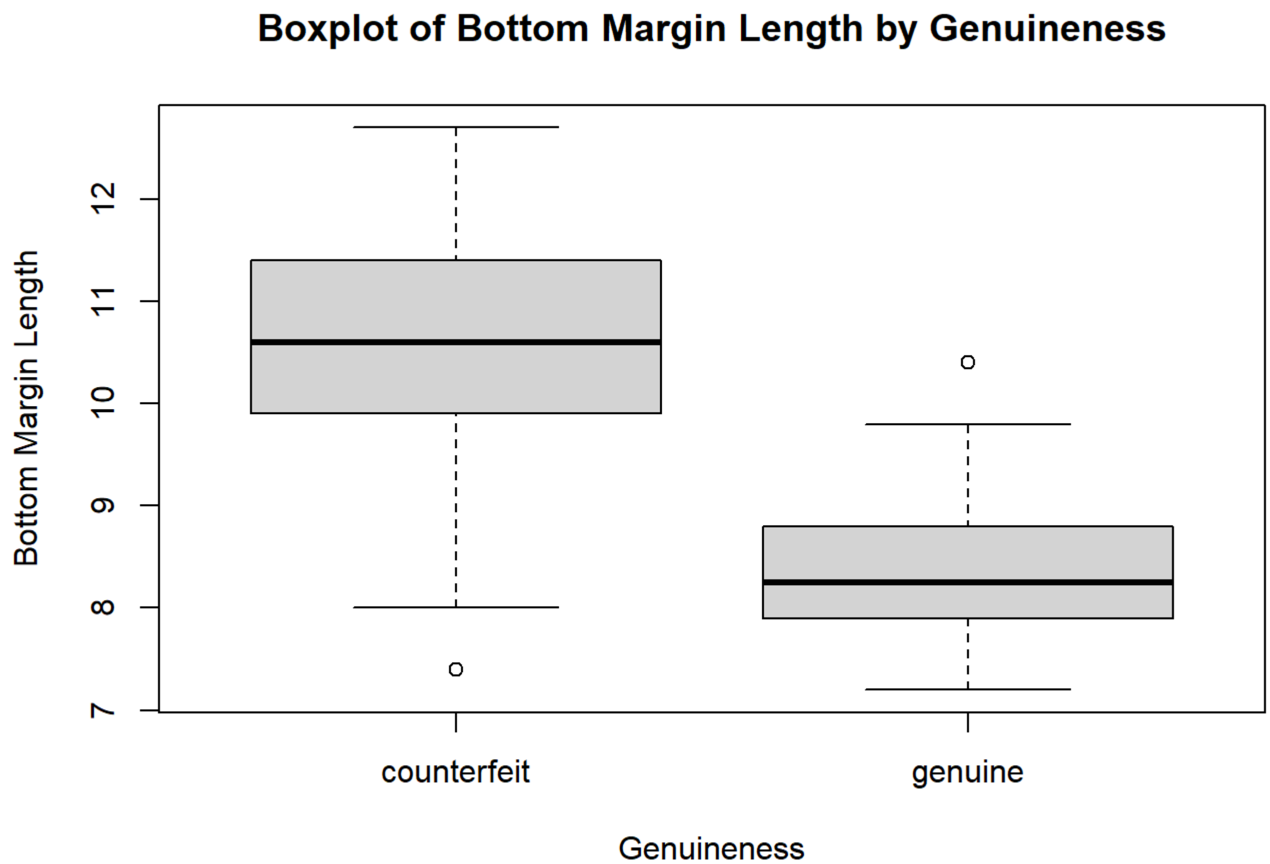
From this we can observe that the length of the left margin is much shorter for genuine bank notes, and longer for the counterfeit notes.

```
boxplot(notes_indc$Right ~ notes_indc$Indicator, xlab = 'Genuineness', ylab = 'Right Margin')
title('Boxplot of Right Margin Length by Genuineness')
```



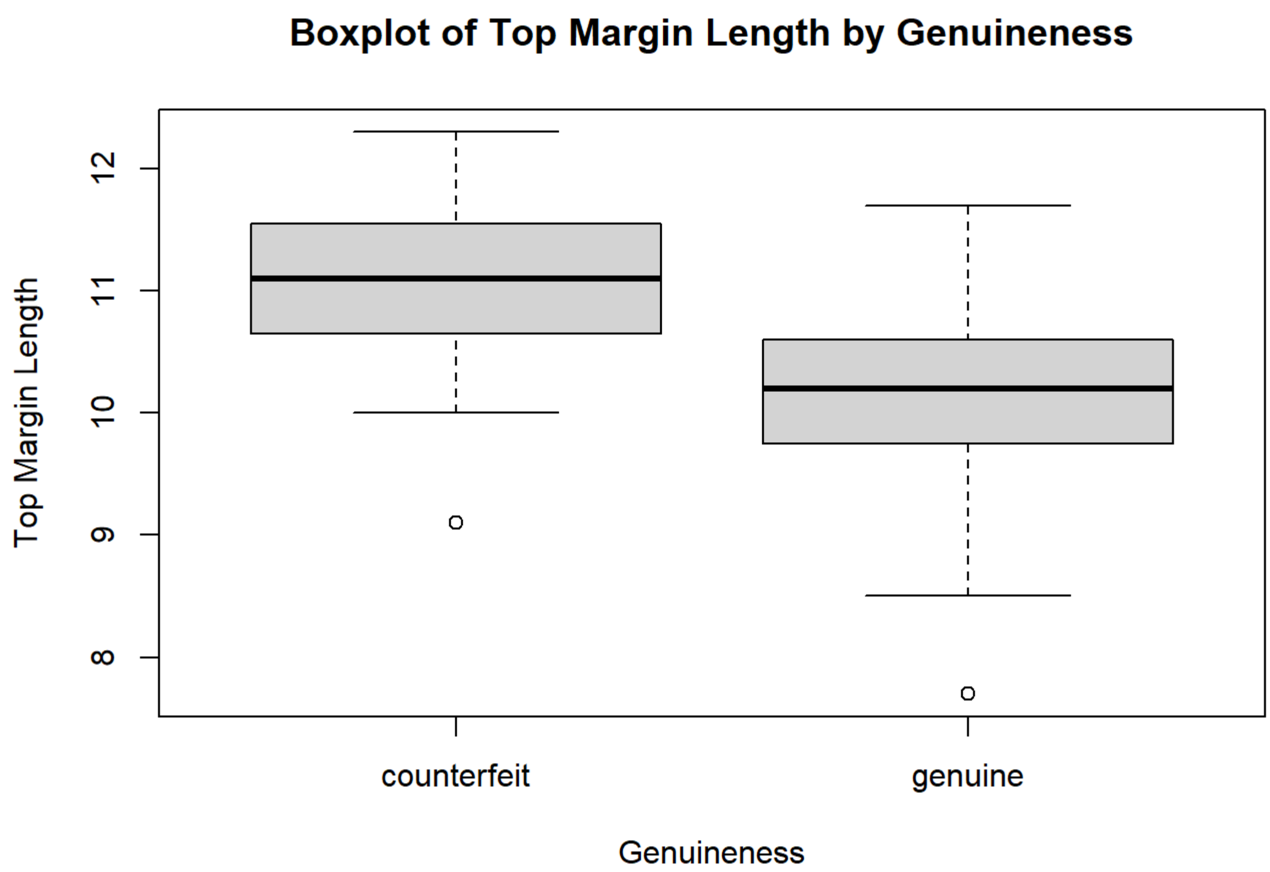
From this we can observe again that the length of the right margin is much shorter for genuine bank notes, and longer for the counterfeit notes.

```
boxplot(notes_indc$`Bottom Margin` ~ notes_indc$Indicator, xlab = 'Genuineness', ylab = 'Bottom
 Margin Length')
title('Boxplot of Bottom Margin Length by Genuineness')
```



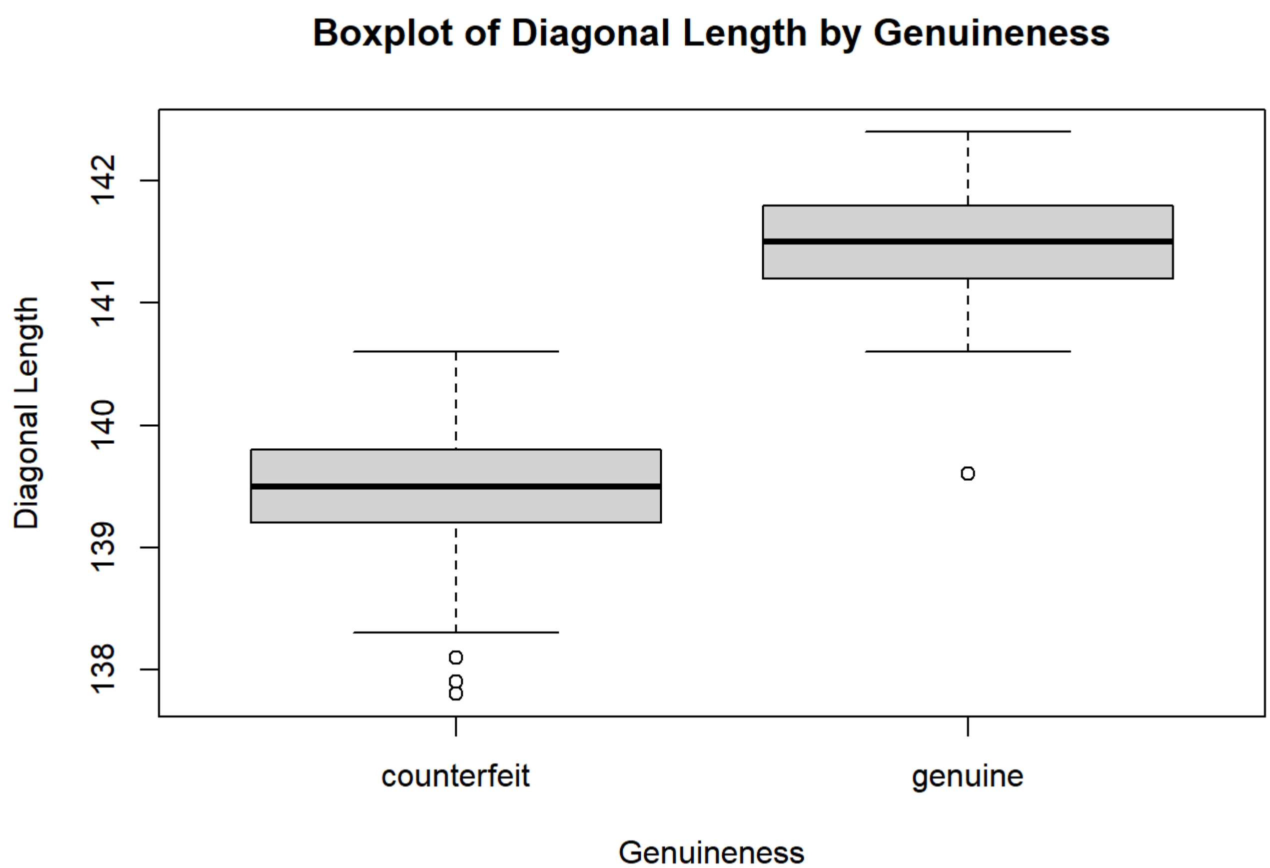
From this we can observe similarly that the length of the bottom margin is much shorter for genuine bank notes, and longer for the counterfeit notes.

```
boxplot(notes_indc$`Top Margin` ~ notes_indc$Indicator, xlab = 'Genuineness', ylab = 'Top Margin
Length')
title('Boxplot of Top Margin Length by Genuineness')
```



From this we can observe that the length of the top margin is slightly, but noticeably shorter for genuine bank notes, and longer for the counterfeit notes.

```
boxplot(notes_indc$Diagonal ~ notes_indc$Indicator, xlab = 'Genuineness', ylab = 'Diagonal Lengt
h')
title('Boxplot of Diagonal Length by Genuineness')
```



However, from this we can observe that the length of the diagonal is significantly longer for genuine bank notes, and much shorter for counterfeit notes.

From the above visualizations, we may gain insight to developing a model to distinguish between the counterfeit and geunine bank notes using these attributes of each group.

We can also visualize the correlation between attributes:

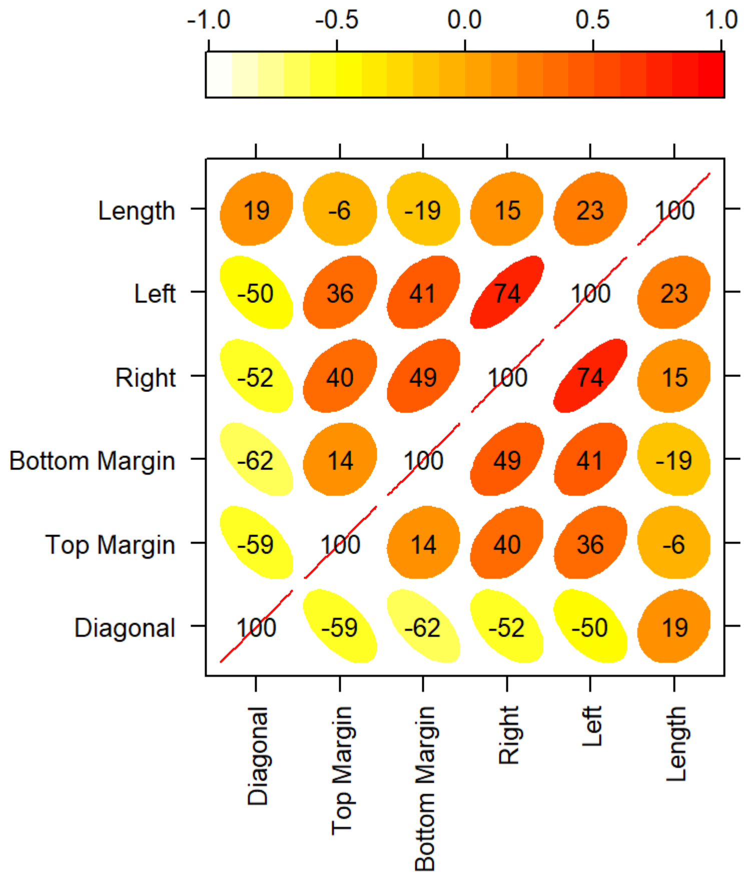
```
library(lattice)
library(ellipse)
```

```
## Warning: package 'ellipse' was built under R version 4.0.4
```

```
##
## Attaching package: 'ellipse'
```

```
## The following object is masked from 'package:graphics':
##
##     pairs
```

```
cor_df <- cor(notes)

# Function to generate correlation plot
panel.corrgram <- function(x, y, z, subscripts, at, level = 0.9, label = FALSE, ...) {
     require("ellipse", quietly = TRUE)
     x <- as.numeric(x)[subscripts]
     y <- as.numeric(y)[subscripts]
     z <- as.numeric(z)[subscripts]
     zcol <- level.colors(z, at = at,  ...)
     for (i in seq(along = z)) {
         ell=ellipse(z[i], level = level, npoints = 50,
                     scale = c(.2, .2), centre = c(x[i], y[i]))
         panel.polygon(ell, col = zcol[i], border = zcol[i], ...)
     }
     if (label)
         panel.text(x = x, y = y, lab = 100 * round(z, 2), cex = 0.8,
                    col = ifelse(z < -1, "white", "black"))
 }

# generate correlation plot
print(levelplot(cor_df[seq(6,1), seq(6,1)], at = do.breaks(c(-1.01, 1.01), 20),
           xlab = NULL, ylab = NULL, colorkey = list(space = "top"), col.regions=rev(heat.colors
(100)),
           scales = list(x = list(rot = 90)),
           panel = panel.corrgram, label = TRUE))
```



The level plot above shows us the correlation between variables through the use of hue and shape, darker colors and narrower ovals indicating stronger correlations between the corresponding two variables. By looking at the level plot it appears that the length of the Left margin and the length of the Right margin are strongly correlated. We can also observe that the attribute paris length of the bottom margin and length of the right margin are some what correlated, and so are left margin & bottom margin, right margin & top margin, and top margin & left margin pairs as well. We can also observe that the Diagonal attribute, with the exception of the length attribute, is negatively correlated with all other attributes. The Length attribute seems to show very liitle correlation with all other attributes.

From this, we can gain intuition for factor analysis that some attributes may be redundant or some attributes may not be contributing to the decision of genuineness of a bank note.

Analysis

Assumptions

There are some assumptions that have to be make before the analysis.

For our Linear Discriminant Analysis we make the following assumptions:

1. The data from group k has common mean vector $\underline{\mu}^{(k)}$, i.e.,

$$\mathbb{E}[x_{ij}^{(k)}] = \underline{\mu}_j^{(k)}.$$

   There were no inconsistencies when selecting observations from each group of genuinity.

2. Homoskedasticity: The data from all groups have common covariance matrix $\mathbf{\Sigma}$, i.e.,

$$\mathbf{\Sigma} = \mathrm{Cov}[\underline{x}_i^{(k)}, \underline{x}_i^{(k)}]$$

```
cov(x = notes, y = notes)
```

```
##                   Length        Left       Right Bottom Margin Top Margin
## Length         0.14179296  0.03144322  0.02309146    -0.1032462 -0.0185407
## Left           0.03144322  0.13033945  0.10842739     0.2158028  0.1050394
## Right          0.02309146  0.10842739  0.16327412     0.2841319  0.1299967
## Bottom Margin -0.10324623  0.21580276  0.28413191     2.0868781  0.1645389
## Top Margin    -0.01854070  0.10503945  0.12999673     0.1645389  0.6447234
## Diagonal       0.08430553 -0.20934196 -0.24047010    -1.0369962 -0.5496148
##                  Diagonal
## Length         0.08430553
## Left          -0.20934196
## Right         -0.24047010
## Bottom Margin -1.03699623
## Top Margin    -0.54961482
## Diagonal       1.32771633
```

3. Independence: The observations are independently sampled.
4. Normality: The data are multivariate normally distributed.

Quantile-Quantile Plot for Length

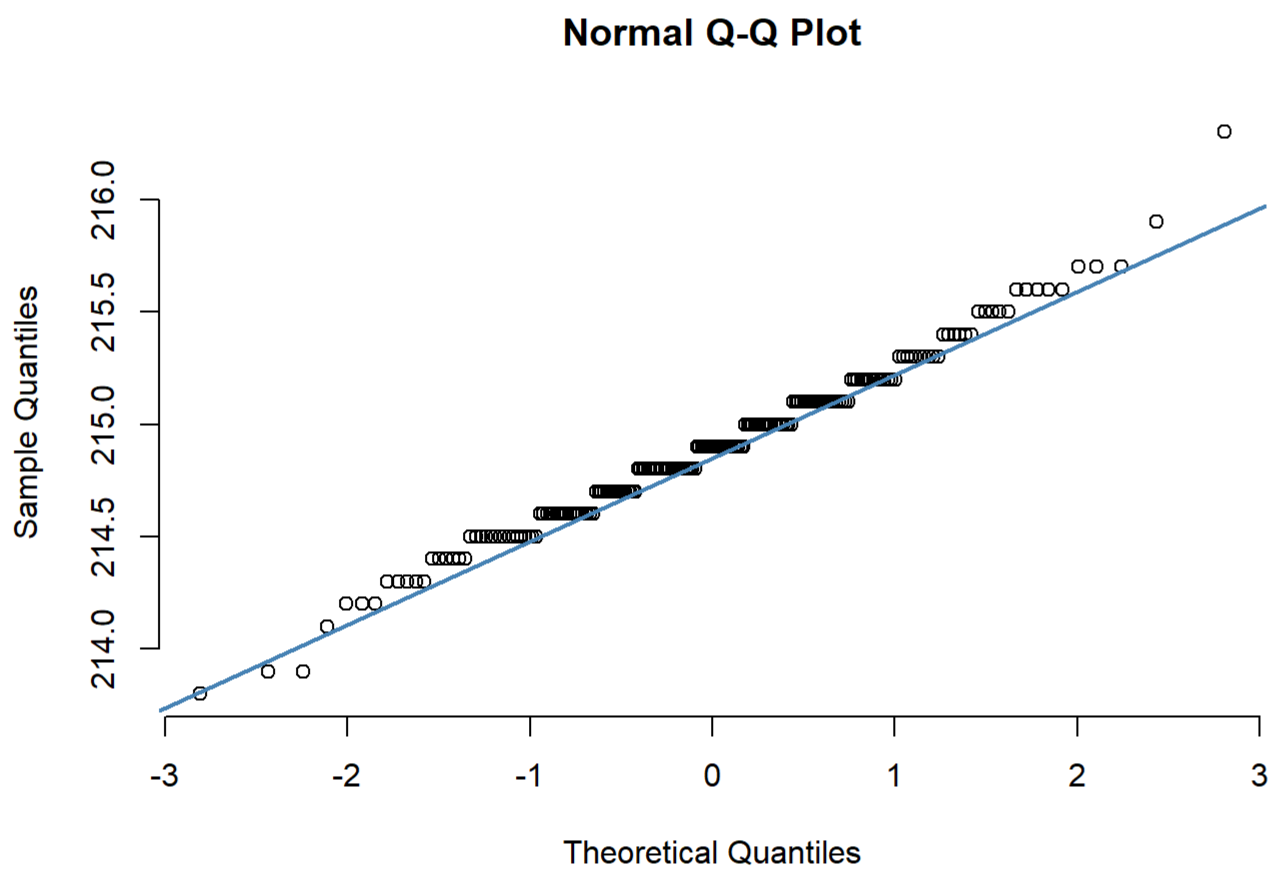
```
library('car')
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

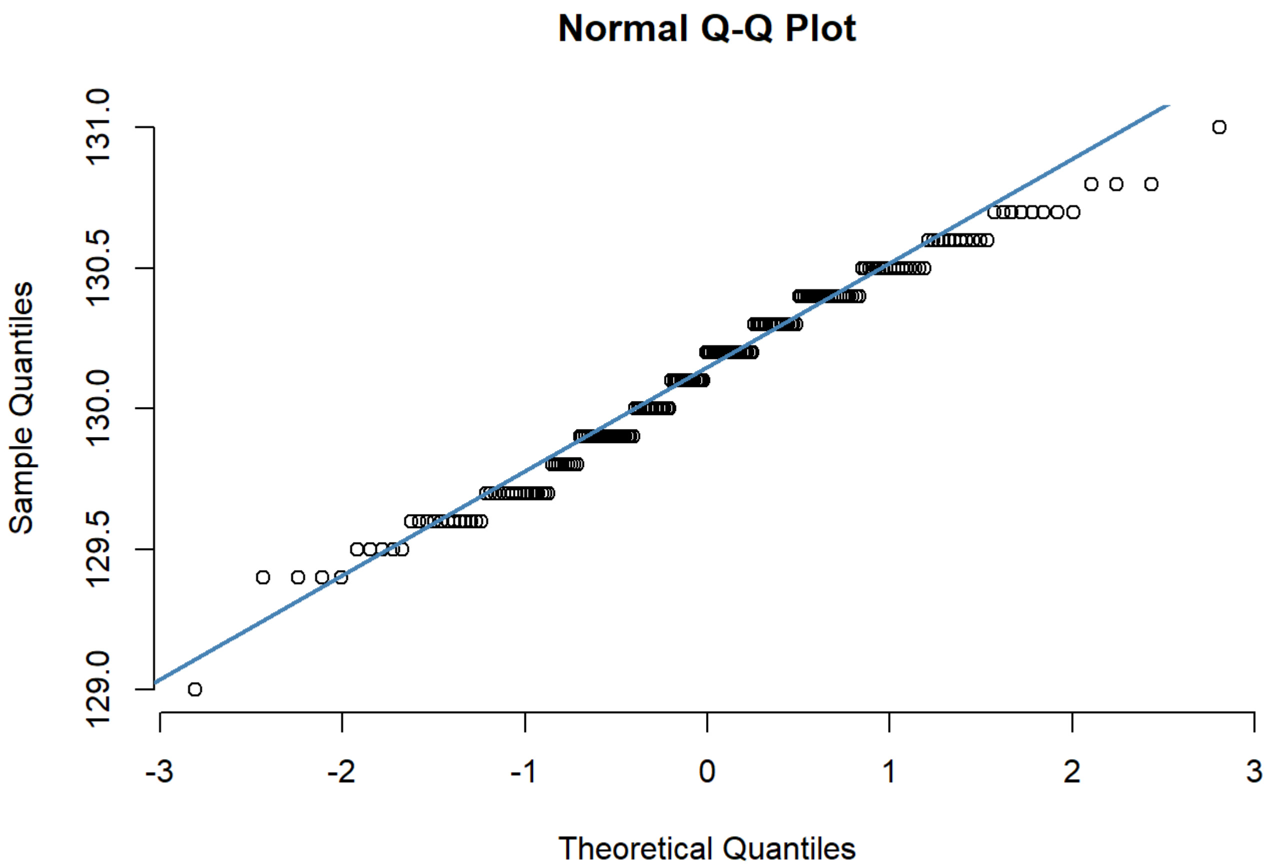
```
## The following object is masked from 'package:ellipse':
##
##     ellipse
```

```
qqnorm(notes$Length, pch = 1, frame = FALSE)
qqline(notes$Length, col = "steelblue", lwd = 2)
```



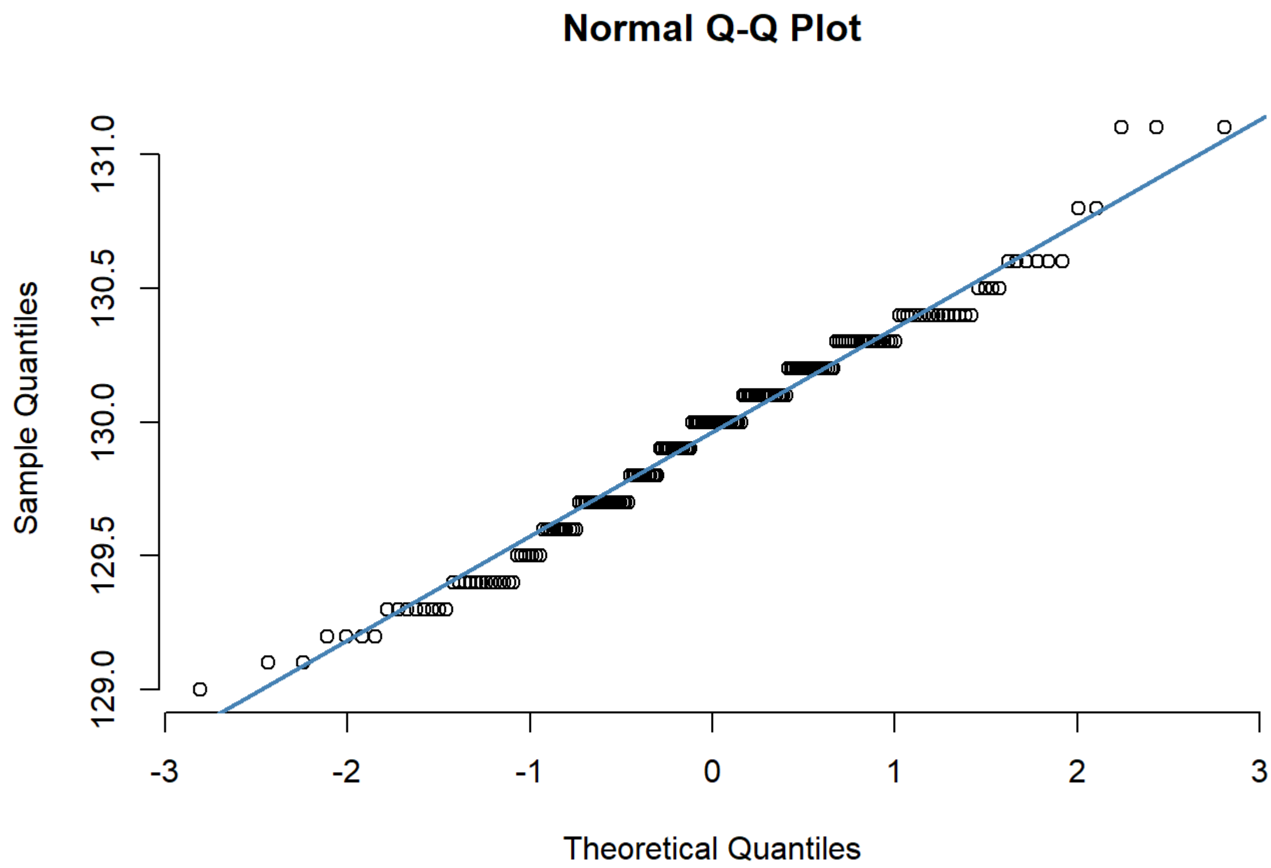
Quantile-Quantile Plot for Left Margin

```
library('car')
qqnorm(notes$Left, pch = 1, frame = FALSE)
qqline(notes$Left, col = "steelblue", lwd = 2)
```



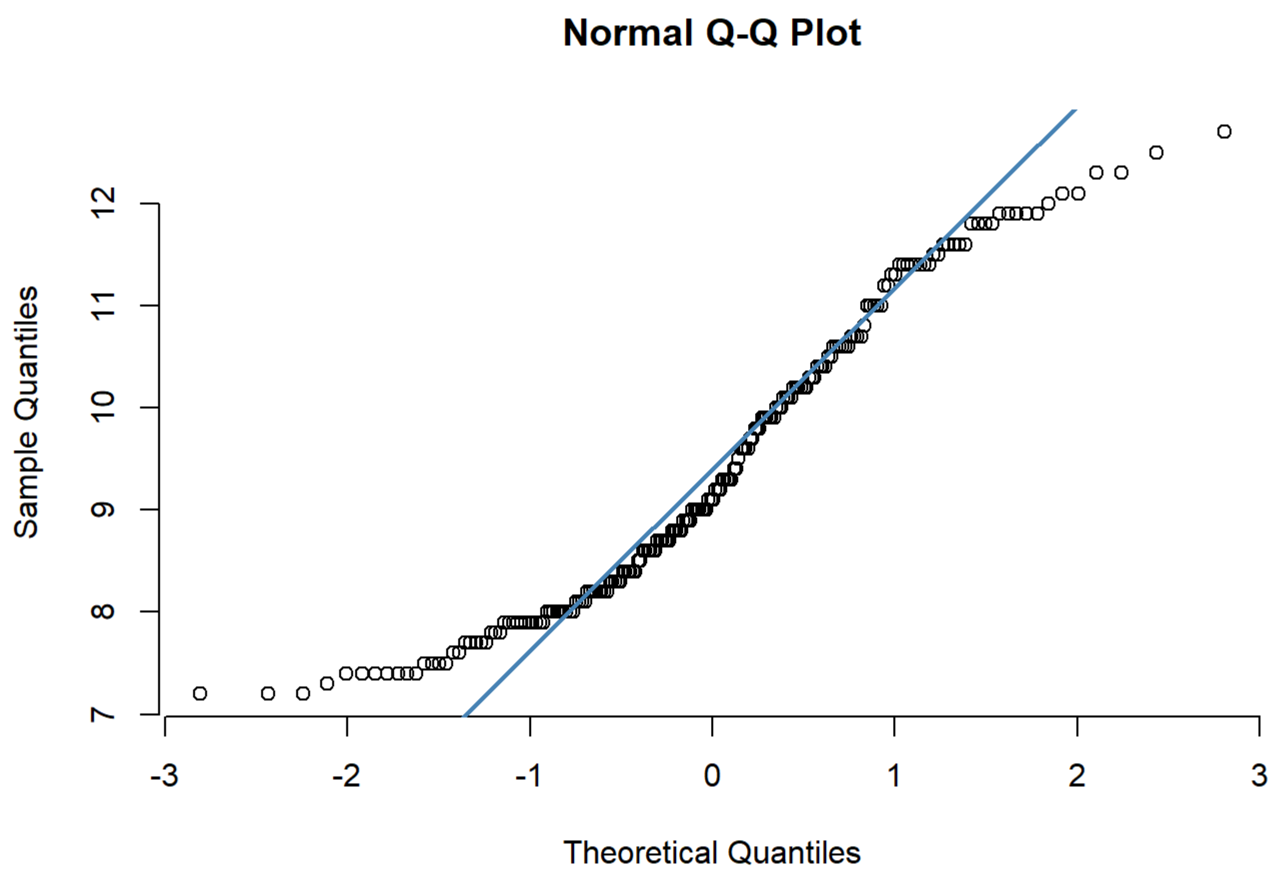
Quantile-Quantile Plot for Right Margin

```
library('car')
qqnorm(notes$Right, pch = 1, frame = FALSE)
qqline(notes$Right, col = "steelblue", lwd = 2)
```



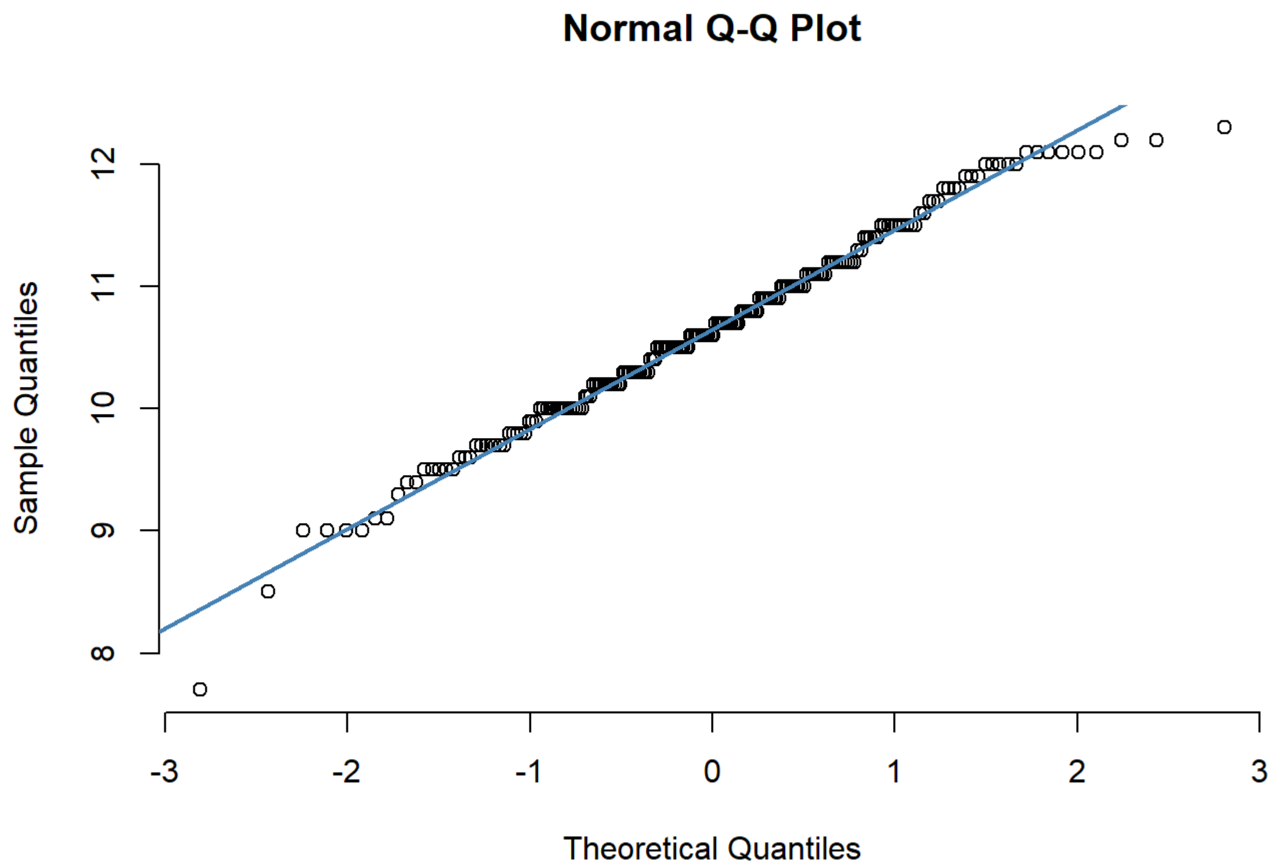
Quantile-Quantile Plot for Bottom Margin

```
library('car')
qqnorm(notes$`Bottom Margin`, pch = 1, frame = FALSE)
qqline(notes$`Bottom Margin`, col = "steelblue", lwd = 2)
```



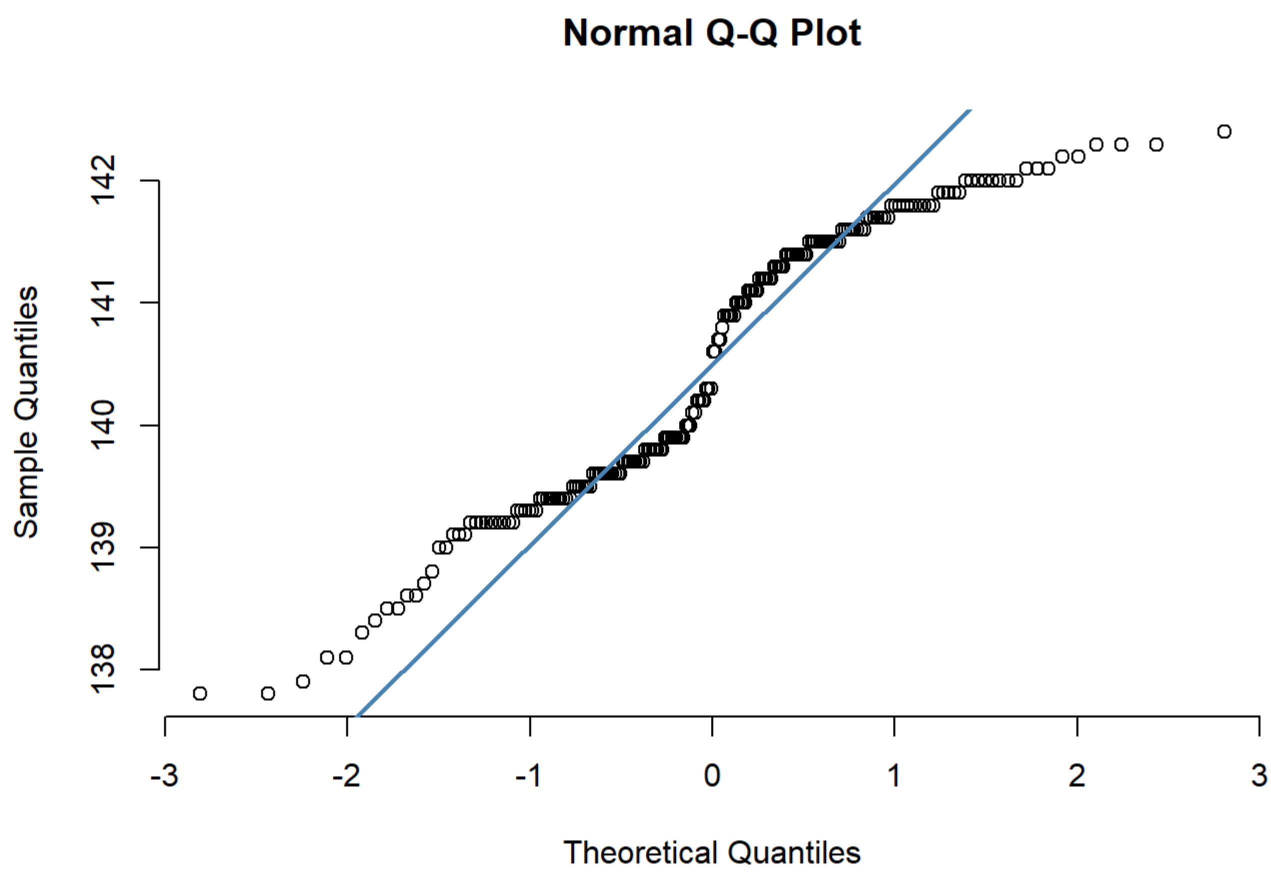
Quantile-Quantile Plot for Top Margin

```
library('car')
qqnorm(notes$`Top Margin`, pch = 1, frame = FALSE)
qqline(notes$`Top Margin`, col = "steelblue", lwd = 2)
```



Quantile-Quantile Plot for Diagonal

```
library('car')
qqnorm(notes$Diagonal, pch = 1, frame = FALSE)
qqline(notes$Diagonal, col = "steelblue", lwd = 2)
```



For our Logistic Regression we make the following assumption:

$\mathbf{P}[y_i = 1|x_i] = p(x_i)$ and $\mathbf{P}[y_i = 0|x_i] = 1 - p(x_i)$.

For our Maximum Likelihood Estimator wemake the following assumption:

The dataset is independently sampled from a multivariate normal distribution, which allows for the establishment of the likelihood function for the factor model.

```
n_factors <- 2
fa_fit <-factanal(notes, n_factors, rotation = 'varimax')
loading <- fa_fit$loadings[,1:2]
t(loading)
```

```
##              Length      Left     Right Bottom Margin Top Margin    Diagonal
## Factor1 -0.1670846 0.5527990 0.5603285     0.6323679 0.59914467 -0.99529478
## Factor2  0.4305069 0.7105278 0.6142669     0.1047625 0.05087066  0.06627958
```

Not an assumption, but a caveat for K-fold cross-validation is:

1. K-fold cross-validation with K < n has a smaller variance than Leave-one-out cross-validation. We are averaging the outputs of K fitted models that are somewhat less correlated with each other, since the overlap between the training sets in each model is smaller.
2. Performing K-fold cross-validation will lead to an intermediate level of bias compared to Leave-one-out cross-validation. Each training set contains (K−1)n/K observations; fewer than in the LOOCV approach, but substantially more than in the validation set approach.

Given these considerations, we will make the choice of k = 10 to yield test error estimates that suffer neither from excessively high bias nor from very high variance.

K-fold cross-validation

We will perform K-fold cross-validation. For each fold, we will use both Linear discriminant analysis (LDA) and logistic regression for classification. We use LDA because it is a supervised classification tool with an objective to solve classification problems when the groups are know as a priori, which is used to predict the group membership of an observation, which in this case would be the group of genuine notes and group of counterfeit notes. We use Logistic Regression because we wish to build a model to predict whether a bank note is genuine or not given the attributes of an observation. Logistic Regression is a supervised classification model that models the probability that the observation will be either genuine or counterfeit.

We will ignore K-fold cross-validation of k = 1, since it make the entire dataset to be a testing and validation set at the same time. We will to a K-fold cross validation of k = 10, and compare the accuracies of both models for each fold.

We will also set seed so that our partition is random, and this will remove the chance that our testing data is either all genuine or all counterfeit, thus letting the test be representative.

For each fold, we will conduct LDA and logistic regression:

```
# K-fold cross-validation using Linear Discriminant Analysis and Logistic Regression
# k = 2

lda_accuracy = c()
lr_accuracy = c()

library(caret)
```

```
## Warning: package 'caret' was built under R version 4.0.4
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 4.0.4
```

```
set.seed(42)

train_control_2 <- trainControl(method="cv", number=2)
# train the model
model_lda_2 <- train(Indicator~., data=notes_indc, trControl=train_control_2, method="lda")
model_lr_2 <- train(Indicator~., data=notes_indc, trControl=train_control_2, method="glm")
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
# validate model
predict_lda_2 <- predict(model_lda_2, notes_indc)
predict_lr_2 <-predict(model_lr_2, notes_indc)

# create confusion matrix
Indicator_fac <- as.factor(Indicator)
conf_lda_2 <- confusionMatrix(predict_lda_2, Indicator_fac)
conf_lr_2 <- confusionMatrix(predict_lr_2, Indicator_fac)

# summarize results and show confusion matrix
print(model_lda_2)
```

```
## Linear Discriminant Analysis
##
## 200 samples
##   6 predictor
##   2 classes: 'counterfeit', 'genuine'
##
## No pre-processing
## Resampling: Cross-Validated (2 fold)
## Summary of sample sizes: 100, 100
## Resampling results:
##
##   Accuracy  Kappa
##   0.995     0.99
```

```
print(conf_lda_2)
```

```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    counterfeit genuine
##   counterfeit         100       1
##   genuine               0      99
##
##                Accuracy : 0.995
##                  95% CI : (0.9725, 0.9999)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.99
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 1.0000
##             Specificity : 0.9900
##          Pos Pred Value : 0.9901
##          Neg Pred Value : 1.0000
##              Prevalence : 0.5000
##          Detection Rate : 0.5000
##    Detection Prevalence : 0.5050
##       Balanced Accuracy : 0.9950
##
##        'Positive' Class : counterfeit
##
```

```
print(model_lr_2)
```

```
## Generalized Linear Model
##
## 200 samples
##   6 predictor
##   2 classes: 'counterfeit', 'genuine'
##
## No pre-processing
## Resampling: Cross-Validated (2 fold)
## Summary of sample sizes: 100, 100
## Resampling results:
##
##   Accuracy  Kappa
##   0.99      0.98
```

```
print(conf_lr_2)
```

```
## Confusion Matrix and Statistics
## 
##              Reference
## Prediction    counterfeit genuine
##   counterfeit         100       0
##   genuine               0     100
##                                       
##                Accuracy : 1           
##                  95% CI : (0.9817, 1)
##     No Information Rate : 0.5         
##     P-Value [Acc > NIR] : < 2.2e-16   
##                                       
##                   Kappa : 1           
##                                       
##  Mcnemar's Test P-Value : NA          
##                                       
##             Sensitivity : 1.0         
##             Specificity : 1.0         
##          Pos Pred Value : 1.0         
##          Neg Pred Value : 1.0         
##              Prevalence : 0.5         
##          Detection Rate : 0.5         
##    Detection Prevalence : 0.5         
##       Balanced Accuracy : 1.0         
##                                       
##        'Positive' Class : counterfeit 
## 
```

```
# append accuracies of each method to accuracy list
lda_accuracy <- c(lda_accuracy, model_lda_2$results$Accuracy)
lr_accuracy <- c(lr_accuracy, model_lr_2$results$Accuracy)
```

```
# K-fold cross-validation using Linear Discriminant Analysis and Logistic Regression
# k = 3

train_control_3 <- trainControl(method="cv", number=3)
# train the model
model_lda_3 <- train(Indicator~., data=notes_indc, trControl=train_control_3, method="lda")
model_lr_3 <- train(Indicator~., data=notes_indc, trControl=train_control_3, method="glm")
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
# validate model
predict_lda_3 <- predict(model_lda_3, notes_indc)
predict_lr_3 <-predict(model_lr_3, notes_indc)

# create confusion matrix
conf_lda_3 <- confusionMatrix(predict_lda_3, Indicator_fac)
conf_lr_3 <- confusionMatrix(predict_lr_3, Indicator_fac)

# summarize results and show confusion matrix
print(model_lda_3)
```

```
## Linear Discriminant Analysis
##
## 200 samples
##   6 predictor
##   2 classes: 'counterfeit', 'genuine'
##
## No pre-processing
## Resampling: Cross-Validated (3 fold)
## Summary of sample sizes: 132, 134, 134
## Resampling results:
##
##   Accuracy   Kappa
##   0.9949495  0.989899
```

```
print(conf_lda_3)
```

```
## Confusion Matrix and Statistics
##
##               Reference
## Prediction    counterfeit genuine
##   counterfeit         100       1
##   genuine               0      99
##
##                Accuracy : 0.995
##                  95% CI : (0.9725, 0.9999)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.99
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 1.0000
##             Specificity : 0.9900
##          Pos Pred Value : 0.9901
##          Neg Pred Value : 1.0000
##              Prevalence : 0.5000
##          Detection Rate : 0.5000
##    Detection Prevalence : 0.5050
##       Balanced Accuracy : 0.9950
##
##        'Positive' Class : counterfeit
##
```

```
print(model_lr_3)
```

```
## Generalized Linear Model
##
## 200 samples
##   6 predictor
##   2 classes: 'counterfeit', 'genuine'
##
## No pre-processing
## Resampling: Cross-Validated (3 fold)
## Summary of sample sizes: 133, 134, 133
## Resampling results:
##
##   Accuracy   Kappa
##   0.9849992  0.9699941
```

```
print(conf_lr_3)
```

```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    counterfeit genuine
##   counterfeit         100       0
##   genuine               0     100
##
##                Accuracy : 1
##                  95% CI : (0.9817, 1)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0
##             Specificity : 1.0
##          Pos Pred Value : 1.0
##          Neg Pred Value : 1.0
##              Prevalence : 0.5
##          Detection Rate : 0.5
##    Detection Prevalence : 0.5
##       Balanced Accuracy : 1.0
##
##        'Positive' Class : counterfeit
##
```

```
# append accuracies of each method to accuracy list
lda_accuracy <- c(lda_accuracy, model_lda_3$results$Accuracy)
lr_accuracy <- c(lr_accuracy, model_lr_3$results$Accuracy)
```

```
# K-fold cross-validation using Linear Discriminant Analysis and Logistic Regression
# k = 4

train_control_4 <- trainControl(method="cv", number=4)
# train the model
model_lda_4 <- train(Indicator~., data=notes_indc, trControl=train_control_4, method="lda")
model_lr_4 <- train(Indicator~., data=notes_indc, trControl=train_control_4, method="glm")
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
# validate model
predict_lda_4 <- predict(model_lda_4, notes_indc)
predict_lr_4 <-predict(model_lr_4, notes_indc)

# create confusion matrix
Indicator <- as.factor(Indicator)
conf_lda_4 <- confusionMatrix(predict_lda_4, Indicator)
conf_lr_4 <- confusionMatrix(predict_lr_4, Indicator)

# summarize results and show confusion matrix
print(model_lda_4)
```

```
## Linear Discriminant Analysis
##
## 200 samples
##   6 predictor
##   2 classes: 'counterfeit', 'genuine'
##
## No pre-processing
## Resampling: Cross-Validated (4 fold)
## Summary of sample sizes: 150, 150, 150, 150
## Resampling results:
##
##   Accuracy  Kappa
##   0.995     0.99
```

```
print(conf_lda_4)
```

```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    counterfeit genuine
##   counterfeit         100       1
##   genuine               0      99
##
##                Accuracy : 0.995
##                  95% CI : (0.9725, 0.9999)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.99
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 1.0000
##             Specificity : 0.9900
##          Pos Pred Value : 0.9901
##          Neg Pred Value : 1.0000
##              Prevalence : 0.5000
##          Detection Rate : 0.5000
##    Detection Prevalence : 0.5050
##       Balanced Accuracy : 0.9950
##
##        'Positive' Class : counterfeit
##
```

```
print(model_lr_4)
```

```
## Generalized Linear Model
##
## 200 samples
##   6 predictor
##   2 classes: 'counterfeit', 'genuine'
##
## No pre-processing
## Resampling: Cross-Validated (4 fold)
## Summary of sample sizes: 150, 150, 150, 150
## Resampling results:
##
##   Accuracy  Kappa
##   0.98      0.96
```

```
print(conf_lr_4)
```

```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    counterfeit genuine
##   counterfeit         100       0
##   genuine               0     100
##
##                Accuracy : 1
##                  95% CI : (0.9817, 1)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0
##             Specificity : 1.0
##          Pos Pred Value : 1.0
##          Neg Pred Value : 1.0
##              Prevalence : 0.5
##          Detection Rate : 0.5
##    Detection Prevalence : 0.5
##       Balanced Accuracy : 1.0
##
##        'Positive' Class : counterfeit
##
```

```
# append accuracies of each method to accuracy list
lda_accuracy <- c(lda_accuracy, model_lda_4$results$Accuracy)
lr_accuracy <- c(lr_accuracy, model_lr_4$results$Accuracy)
```

```
# K-fold cross-validation using Linear Discriminant Analysis and Logistic Regression
# k = 5

train_control_5 <- trainControl(method="cv", number=4)
# train the model
model_lda_5 <- train(Indicator~., data=notes_indc, trControl=train_control_5, method="lda")
model_lr_5 <- train(Indicator~., data=notes_indc, trControl=train_control_5, method="glm")
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
# validate model
predict_lda_5 <- predict(model_lda_5, notes_indc)
predict_lr_5 <-predict(model_lr_5, notes_indc)

# create confusion matrix
Indicator <- as.factor(Indicator)
conf_lda_5 <- confusionMatrix(predict_lda_5, Indicator)
conf_lr_5 <- confusionMatrix(predict_lr_5, Indicator)

# summarize results and show confusion matrix
print(model_lda_5)
```

```
## Linear Discriminant Analysis
##
## 200 samples
##   6 predictor
##   2 classes: 'counterfeit', 'genuine'
##
## No pre-processing
## Resampling: Cross-Validated (4 fold)
## Summary of sample sizes: 150, 150, 150, 150
## Resampling results:
##
##   Accuracy  Kappa
##   0.995     0.99
```

```
print(conf_lda_5)
```

```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    counterfeit genuine
##   counterfeit         100       1
##   genuine               0      99
##
##                Accuracy : 0.995
##                  95% CI : (0.9725, 0.9999)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.99
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 1.0000
##             Specificity : 0.9900
##          Pos Pred Value : 0.9901
##          Neg Pred Value : 1.0000
##              Prevalence : 0.5000
##          Detection Rate : 0.5000
##    Detection Prevalence : 0.5050
##       Balanced Accuracy : 0.9950
##
##        'Positive' Class : counterfeit
##
```

```
print(model_lr_5)
```

```
## Generalized Linear Model
##
## 200 samples
##   6 predictor
##   2 classes: 'counterfeit', 'genuine'
##
## No pre-processing
## Resampling: Cross-Validated (4 fold)
## Summary of sample sizes: 150, 150, 150, 150
## Resampling results:
##
##   Accuracy  Kappa
##   0.985     0.97
```

```
print(conf_lr_5)
```

```
## Confusion Matrix and Statistics
## 
##              Reference
## Prediction    counterfeit genuine
##   counterfeit         100       0
##   genuine               0     100
##                                      
##                Accuracy : 1          
##                  95% CI : (0.9817, 1)
##     No Information Rate : 0.5        
##     P-Value [Acc > NIR] : < 2.2e-16  
##                                      
##                   Kappa : 1          
##                                      
##  Mcnemar's Test P-Value : NA         
##                                      
##             Sensitivity : 1.0        
##             Specificity : 1.0        
##          Pos Pred Value : 1.0        
##          Neg Pred Value : 1.0        
##              Prevalence : 0.5        
##          Detection Rate : 0.5        
##    Detection Prevalence : 0.5        
##       Balanced Accuracy : 1.0        
##                                      
##        'Positive' Class : counterfeit
## 
```

```
# append accuracies of each method to accuracy list
lda_accuracy <- c(lda_accuracy, model_lda_5$results$Accuracy)
lr_accuracy <- c(lr_accuracy, model_lr_5$results$Accuracy)
```

```
# K-fold cross-validation using Linear Discriminant Analysis and Logistic Regression
# k = 6

train_control_6 <- trainControl(method="cv", number=6)
# train the model
model_lda_6 <- train(Indicator~., data=notes_indc, trControl=train_control_6, method="lda")
model_lr_6 <- train(Indicator~., data=notes_indc, trControl=train_control_6, method="glm")
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
# validate model
predict_lda_6 <- predict(model_lda_6, notes_indc)
predict_lr_6 <-predict(model_lr_6, notes_indc)

# create confusion matrix
Indicator <- as.factor(Indicator)
conf_lda_6 <- confusionMatrix(predict_lda_6, Indicator)
conf_lr_6 <- confusionMatrix(predict_lr_6, Indicator)

# summarize results and show confusion matrix
print(model_lda_6)
```

```
## Linear Discriminant Analysis
##
## 200 samples
##   6 predictor
##   2 classes: 'counterfeit', 'genuine'
##
## No pre-processing
## Resampling: Cross-Validated (6 fold)
## Summary of sample sizes: 168, 166, 167, 166, 166, 167, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.9947917  0.9895833
```

```
print(conf_lda_6)
```

```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    counterfeit genuine
##   counterfeit         100       1
##   genuine               0      99
##
##                Accuracy : 0.995
##                  95% CI : (0.9725, 0.9999)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.99
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 1.0000
##             Specificity : 0.9900
##          Pos Pred Value : 0.9901
##          Neg Pred Value : 1.0000
##              Prevalence : 0.5000
##          Detection Rate : 0.5000
##    Detection Prevalence : 0.5050
##       Balanced Accuracy : 0.9950
##
##        'Positive' Class : counterfeit
##
```

```
print(model_lr_6)
```

```
## Generalized Linear Model
##
## 200 samples
##   6 predictor
##   2 classes: 'counterfeit', 'genuine'
##
## No pre-processing
## Resampling: Cross-Validated (6 fold)
## Summary of sample sizes: 166, 166, 167, 167, 167, 167, ...
## Resampling results:
##
##   Accuracy  Kappa
##   0.974896  0.9498661
```

```
print(conf_lr_6)
```

```
## Confusion Matrix and Statistics
## 
##              Reference
## Prediction    counterfeit genuine
##   counterfeit         100       0
##   genuine               0     100
##                                      
##                Accuracy : 1          
##                  95% CI : (0.9817, 1)
##     No Information Rate : 0.5        
##     P-Value [Acc > NIR] : < 2.2e-16  
##                                      
##                   Kappa : 1          
##                                      
##  Mcnemar's Test P-Value : NA         
##                                      
##             Sensitivity : 1.0        
##             Specificity : 1.0        
##          Pos Pred Value : 1.0        
##          Neg Pred Value : 1.0        
##              Prevalence : 0.5        
##          Detection Rate : 0.5        
##    Detection Prevalence : 0.5        
##       Balanced Accuracy : 1.0        
##                                      
##        'Positive' Class : counterfeit
## 
```

```
# append accuracies of each method to accuracy list
lda_accuracy <- c(lda_accuracy, model_lda_6$results$Accuracy)
lr_accuracy <- c(lr_accuracy, model_lr_6$results$Accuracy)
```

```
# K-fold cross-validation using Linear Discriminant Analysis and Logistic Regression
# k = 7

train_control_7 <- trainControl(method="cv", number=7)
# train the model
model_lda_7 <- train(Indicator~., data=notes_indc, trControl=train_control_7, method="lda")
model_lr_7 <- train(Indicator~., data=notes_indc, trControl=train_control_7, method="glm")
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
# validate model
predict_lda_7 <- predict(model_lda_7, notes_indc)
predict_lr_7 <-predict(model_lr_7, notes_indc)

# create confusion matrix
Indicator <- as.factor(Indicator)
conf_lda_7 <- confusionMatrix(predict_lda_7, Indicator)
conf_lr_7 <- confusionMatrix(predict_lr_7, Indicator)

# summarize results and show confusion matrix
print(model_lda_7)
```

```
## Linear Discriminant Analysis
##
## 200 samples
##   6 predictor
##   2 classes: 'counterfeit', 'genuine'
##
## No pre-processing
## Resampling: Cross-Validated (7 fold)
## Summary of sample sizes: 171, 171, 171, 172, 172, 172, ...
## Resampling results:
##
##   Accuracy  Kappa
##   0.994898  0.9897959
```

```
print(conf_lda_7)
```

```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    counterfeit genuine
##   counterfeit         100       1
##   genuine               0      99
##
##                Accuracy : 0.995
##                  95% CI : (0.9725, 0.9999)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.99
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 1.0000
##             Specificity : 0.9900
##          Pos Pred Value : 0.9901
##          Neg Pred Value : 1.0000
##              Prevalence : 0.5000
##          Detection Rate : 0.5000
##    Detection Prevalence : 0.5050
##       Balanced Accuracy : 0.9950
##
##        'Positive' Class : counterfeit
##
```

```
print(model_lr_7)
```

```
## Generalized Linear Model
##
## 200 samples
##   6 predictor
##   2 classes: 'counterfeit', 'genuine'
##
## No pre-processing
## Resampling: Cross-Validated (7 fold)
## Summary of sample sizes: 171, 172, 170, 171, 172, 172, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.9802838  0.9604971
```

```
print(conf_lr_7)
```

```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    counterfeit genuine
##   counterfeit         100       0
##   genuine               0     100
##
##                Accuracy : 1
##                  95% CI : (0.9817, 1)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0
##             Specificity : 1.0
##          Pos Pred Value : 1.0
##          Neg Pred Value : 1.0
##              Prevalence : 0.5
##          Detection Rate : 0.5
##    Detection Prevalence : 0.5
##       Balanced Accuracy : 1.0
##
##        'Positive' Class : counterfeit
##
```

```
# append accuracies of each method to accuracy list
lda_accuracy <- c(lda_accuracy, model_lda_7$results$Accuracy)
lr_accuracy <- c(lr_accuracy, model_lr_7$results$Accuracy)
```

```
# K-fold cross-validation using Linear Discriminant Analysis and Logistic Regression
# k = 8

train_control_8 <- trainControl(method="cv", number=8)
# train the model
model_lda_8 <- train(Indicator~., data=notes_indc, trControl=train_control_8, method="lda")
model_lr_8 <- train(Indicator~., data=notes_indc, trControl=train_control_8, method="glm")
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
# validate model
predict_lda_8 <- predict(model_lda_8, notes_indc)
predict_lr_8 <-predict(model_lr_8, notes_indc)

# create confusion matrix
Indicator <- as.factor(Indicator)
conf_lda_8 <- confusionMatrix(predict_lda_8, Indicator)
conf_lr_8 <- confusionMatrix(predict_lr_8, Indicator)

# summarize results and show confusion matrix
print(model_lda_8)
```

```
## Linear Discriminant Analysis
##
## 200 samples
##   6 predictor
##   2 classes: 'counterfeit', 'genuine'
##
## No pre-processing
## Resampling: Cross-Validated (8 fold)
## Summary of sample sizes: 175, 174, 175, 175, 176, 175, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.9947917  0.9895833
```

```
print(conf_lda_8)
```

```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    counterfeit genuine
##   counterfeit         100       1
##   genuine               0      99
##
##                Accuracy : 0.995
##                  95% CI : (0.9725, 0.9999)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.99
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 1.0000
##             Specificity : 0.9900
##          Pos Pred Value : 0.9901
##          Neg Pred Value : 1.0000
##              Prevalence : 0.5000
##          Detection Rate : 0.5000
##    Detection Prevalence : 0.5050
##       Balanced Accuracy : 0.9950
##
##        'Positive' Class : counterfeit
##
```

```
print(model_lr_8)
```

```
## Generalized Linear Model
##
## 200 samples
##   6 predictor
##   2 classes: 'counterfeit', 'genuine'
##
## No pre-processing
## Resampling: Cross-Validated (8 fold)
## Summary of sample sizes: 176, 174, 174, 176, 174, 176, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.9799679  0.9599359
```

```
print(conf_lr_8)
```

```
## Confusion Matrix and Statistics
## 
##              Reference
## Prediction    counterfeit genuine
##   counterfeit         100       0
##   genuine               0     100
##                                         
##                Accuracy : 1         
##                  95% CI : (0.9817, 1)
##     No Information Rate : 0.5       
##     P-Value [Acc > NIR] : < 2.2e-16 
##                                         
##                   Kappa : 1         
##                                         
##  Mcnemar's Test P-Value : NA        
##                                         
##             Sensitivity : 1.0       
##             Specificity : 1.0       
##          Pos Pred Value : 1.0       
##          Neg Pred Value : 1.0       
##              Prevalence : 0.5       
##          Detection Rate : 0.5       
##    Detection Prevalence : 0.5       
##       Balanced Accuracy : 1.0       
##                                         
##        'Positive' Class : counterfeit
## 
```

```
# append accuracies of each method to accuracy list
lda_accuracy <- c(lda_accuracy, model_lda_8$results$Accuracy)
lr_accuracy <- c(lr_accuracy, model_lr_8$results$Accuracy)
```

```
# K-fold cross-validation using Linear Discriminant Analysis and Logistic Regression
# k = 9

train_control_9 <- trainControl(method="cv", number=9)
# train the model
model_lda_9 <- train(Indicator~., data=notes_indc, trControl=train_control_9, method="lda")
model_lr_9 <- train(Indicator~., data=notes_indc, trControl=train_control_9, method="glm")
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
# validate model
predict_lda_9 <- predict(model_lda_9, notes_indc)
predict_lr_9 <-predict(model_lr_9, notes_indc)

# create confusion matrix
Indicator <- as.factor(Indicator)
conf_lda_9 <- confusionMatrix(predict_lda_9, Indicator)
conf_lr_9 <- confusionMatrix(predict_lr_9, Indicator)

# summarize results and show confusion matrix
print(model_lda_9)
```

```
## Linear Discriminant Analysis
##
## 200 samples
##   6 predictor
##   2 classes: 'counterfeit', 'genuine'
##
## No pre-processing
## Resampling: Cross-Validated (9 fold)
## Summary of sample sizes: 178, 178, 178, 178, 178, 178, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.9949495  0.989899
```

```
print(conf_lda_9)
```

```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    counterfeit genuine
##   counterfeit         100       1
##   genuine               0      99
##
##                Accuracy : 0.995
##                  95% CI : (0.9725, 0.9999)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.99
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 1.0000
##             Specificity : 0.9900
##          Pos Pred Value : 0.9901
##          Neg Pred Value : 1.0000
##              Prevalence : 0.5000
##          Detection Rate : 0.5000
##    Detection Prevalence : 0.5050
##       Balanced Accuracy : 0.9950
##
##        'Positive' Class : counterfeit
##
```

```
print(model_lr_9)
```

```
## Generalized Linear Model
##
## 200 samples
##   6 predictor
##   2 classes: 'counterfeit', 'genuine'
##
## No pre-processing
## Resampling: Cross-Validated (9 fold)
## Summary of sample sizes: 177, 178, 178, 178, 178, 178, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.9850681  0.9701544
```

```
print(conf_lr_9)
```

```
## Confusion Matrix and Statistics
## 
##              Reference
## Prediction    counterfeit genuine
##   counterfeit         100       0
##   genuine               0     100
##                                   
##                Accuracy : 1       
##                  95% CI : (0.9817, 1)
##     No Information Rate : 0.5     
##     P-Value [Acc > NIR] : < 2.2e-16
##                                   
##                   Kappa : 1       
##                                   
##  Mcnemar's Test P-Value : NA      
##                                   
##             Sensitivity : 1.0     
##             Specificity : 1.0     
##          Pos Pred Value : 1.0     
##          Neg Pred Value : 1.0     
##              Prevalence : 0.5     
##          Detection Rate : 0.5     
##    Detection Prevalence : 0.5     
##       Balanced Accuracy : 1.0     
##                                   
##        'Positive' Class : counterfeit
## 
```

```
# append accuracies of each method to accuracy list
lda_accuracy <- c(lda_accuracy, model_lda_9$results$Accuracy)
lr_accuracy <- c(lr_accuracy, model_lr_9$results$Accuracy)
```

```
# K-fold cross-validation using Linear Discriminant Analysis and Logistic Regression
# k = 10

train_control_10 <- trainControl(method="cv", number=10)
# train the model
model_lda_10 <- train(Indicator~., data=notes_indc, trControl=train_control_10, method="lda")
model_lr_10 <- train(Indicator~., data=notes_indc, trControl=train_control_10, method="glm")
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
# validate model
predict_lda_10 <- predict(model_lda_10, notes_indc)
predict_lr_10 <-predict(model_lr_10, notes_indc)

# create confusion matrix
Indicator <- as.factor(Indicator)
conf_lda_10 <- confusionMatrix(predict_lda_10, Indicator)
conf_lr_10 <- confusionMatrix(predict_lr_10, Indicator)

# summarize results and show confusion matrix
print(model_lda_10)
```

```
## Linear Discriminant Analysis
##
## 200 samples
##   6 predictor
##   2 classes: 'counterfeit', 'genuine'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 180, 180, 180, 180, 180, 180, ...
## Resampling results:
##
##   Accuracy  Kappa
##   0.995     0.99
```

```
print(conf_lda_10)
```

```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    counterfeit genuine
##   counterfeit         100       1
##   genuine               0      99
##
##                Accuracy : 0.995
##                  95% CI : (0.9725, 0.9999)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.99
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 1.0000
##             Specificity : 0.9900
##          Pos Pred Value : 0.9901
##          Neg Pred Value : 1.0000
##              Prevalence : 0.5000
##          Detection Rate : 0.5000
##    Detection Prevalence : 0.5050
##       Balanced Accuracy : 0.9950
##
##        'Positive' Class : counterfeit
##
```

```
print(model_lr_10)
```

```
## Generalized Linear Model
##
## 200 samples
##   6 predictor
##   2 classes: 'counterfeit', 'genuine'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 180, 180, 180, 180, 180, 180, ...
## Resampling results:
##
##   Accuracy  Kappa
##   0.98      0.96
```

```
print(conf_lr_10)
```

```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    counterfeit genuine
##   counterfeit         100       0
##   genuine               0     100
##
##                Accuracy : 1
##                  95% CI : (0.9817, 1)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0
##             Specificity : 1.0
##          Pos Pred Value : 1.0
##          Neg Pred Value : 1.0
##              Prevalence : 0.5
##          Detection Rate : 0.5
##    Detection Prevalence : 0.5
##       Balanced Accuracy : 1.0
##
##        'Positive' Class : counterfeit
##
```

```
# append accuracies of each method to accuracy list
lda_accuracy <- c(lda_accuracy, model_lda_10$results$Accuracy)
lr_accuracy <- c(lr_accuracy, model_lr_10$results$Accuracy)
```

```
num_fold = c(2, 3, 4, 5, 6, 7, 8, 9, 10)
results = cbind(num_fold, lda_accuracy, lr_accuracy)
results
```

```
##       num_fold lda_accuracy lr_accuracy
##  [1,]        2    0.9950000   0.9900000
##  [2,]        3    0.9949495   0.9849992
##  [3,]        4    0.9950000   0.9800000
##  [4,]        5    0.9950000   0.9850000
##  [5,]        6    0.9947917   0.9748960
##  [6,]        7    0.9948980   0.9802838
##  [7,]        8    0.9947917   0.9799679
##  [8,]        9    0.9949495   0.9850681
##  [9,]       10    0.9950000   0.9800000
```

As we can observe from the results of each fold, we are easily able to observe that the Linear Discriminant Analysis consistantly yields a model with an accuracy over 99.49%, whereas the logsitic regression model struggles to consistantly do the same, yielding accuracies varying between 99% and 97.49%.

Therefore from the K-fold cross-validation of 10 folds with LDA model and logistic regression model from each fold, we are able to conclude that LDA is a better model of classification of genuine and counterfeit bank notes.

Factor Analysis (Maximum Likelihood Estimator)

Here, we will perform factor analysis on our Swiss Bank Notes dataset through the Maximum Likelihood Estimator (MLE) method. Through factor analysis, we may attempt to remove redundant attributes or attributes that do not impact, or disrupt the decision of the genuinenity of a Swiss Bank Note.

```
Indicator_int <- c()

for(count in 1:100){
  Indicator_int <- c(Indicator_int, 1)
}

for(count in 1:100){
  Indicator_int <- c(Indicator_int, 0)
}

notes_indc_int <- cbind(notes, Indicator_int)

mle <- factanal(notes_indc_int, factors = 2, scores = 'regression')
scores <- mle$scores
scores <- as.array(scores)
```

```
indic <- rep(c(1, 0), each=100)
mle_indic <- data.frame(scores, Indicator)
mle_indic
```

```
##          Factor1     Factor2   Indicator
## BN1   -1.0863751  3.29481358     genuine
## BN2   -0.9757616 -0.73173771     genuine
## BN3   -0.9838926 -0.70093227     genuine
## BN4   -0.9826998 -0.74011051     genuine
## BN5   -0.9688717 -0.93070467     genuine
## BN6   -1.0810730  2.64434275     genuine
## BN7   -0.9700301 -1.05160266     genuine
## BN8   -0.9603059 -1.25027044     genuine
## BN9   -0.9502164 -1.39262122     genuine
## BN10  -1.0232316  1.46731501     genuine
## BN11  -1.0455487  1.49014362     genuine
## BN12  -0.9717105 -1.17868021     genuine
## BN13  -1.0679109  2.07015579     genuine
## BN14  -0.9731215 -0.68916058     genuine
## BN15  -0.9984447 -0.12383150     genuine
## BN16  -0.9794531 -0.46206877     genuine
## BN17  -0.9953379 -0.01533874     genuine
## BN18  -0.9993019 -0.17322491     genuine
## BN19  -0.9657270 -0.88169366     genuine
## BN20  -1.0209824  0.63551764     genuine
## BN21  -0.9846073 -0.36417841     genuine
## BN22  -1.0567385  1.61057468     genuine
## BN23  -1.0550953  1.81219596     genuine
## BN24  -1.0298819  0.88553115     genuine
## BN25  -0.9718627 -0.49491426     genuine
## BN26  -1.0587200  1.52207099     genuine
## BN27  -1.0429645  0.88248523     genuine
## BN28  -1.0345057  1.16957539     genuine
## BN29  -1.0135470  0.27988565     genuine
## BN30  -0.9810884 -0.92577923     genuine
## BN31  -1.0174757  0.44165755     genuine
## BN32  -0.9887262 -0.71206873     genuine
## BN33  -0.9852014  0.06708585     genuine
## BN34  -1.0359484  1.43135122     genuine
## BN35  -1.0422162  1.43206926     genuine
## BN36  -1.0173555  0.77845662     genuine
## BN37  -1.0427410  1.08423647     genuine
## BN38  -1.0043425 -0.26674927     genuine
## BN39  -1.0479069  1.08774344     genuine
## BN40  -1.0018478  0.24761170     genuine
## BN41  -0.9831546 -0.80668729     genuine
## BN42  -0.9908442  0.26408320     genuine
## BN43  -0.9630821 -1.09049182     genuine
## BN44  -1.0185332  1.07630920     genuine
## BN45  -0.9290872 -1.73311462     genuine
## BN46  -0.9484108 -1.43422687     genuine
## BN47  -0.9876307 -0.18343682     genuine
## BN48  -0.9999876 -0.18398896     genuine
## BN49  -0.9629833 -0.65140070     genuine
## BN50  -0.9188028 -2.53384218     genuine
## BN51  -0.9768566 -0.63325694     genuine
## BN52  -1.0458765  1.80388589     genuine
```

```
## BN53  -0.9923772  0.43977098     genuine
## BN54  -1.0303597  0.95050991     genuine
## BN55  -0.9394667 -1.58505734     genuine
## BN56  -0.9804088 -0.78639641     genuine
## BN57  -0.9936531  0.08258290     genuine
## BN58  -0.9562156 -1.06044830     genuine
## BN59  -1.0062733  0.49270411     genuine
## BN60  -0.9998511  0.17798285     genuine
## BN61  -0.9649110 -1.09162023     genuine
## BN62  -0.9776989 -0.74209805     genuine
## BN63  -0.9842284 -0.47339843     genuine
## BN64  -0.9870775 -0.09027770     genuine
## BN65  -0.9964750  0.16330611     genuine
## BN66  -1.0389058  1.80524002     genuine
## BN67  -0.9442759 -1.47559126     genuine
## BN68  -0.9643791 -0.90458524     genuine
## BN69  -0.9726546 -0.82502321     genuine
## BN70  -0.9815929  0.90355191     genuine
## BN71  -0.9465564 -0.69955453     genuine
## BN72  -0.9894023 -0.21286865     genuine
## BN73  -0.9626132 -0.61038109     genuine
## BN74  -0.9901159 -0.31642497     genuine
## BN75  -1.0095405 -0.11591506     genuine
## BN76  -0.9532635 -1.27058704     genuine
## BN77  -0.9854917  0.05681904     genuine
## BN78  -0.9671577 -0.61596829     genuine
## BN79  -1.0341452  1.45073241     genuine
## BN80  -0.9906708 -0.39506400     genuine
## BN81  -0.9552096 -0.59901559     genuine
## BN82  -0.9743111 -0.45554678     genuine
## BN83  -0.9751476 -0.94258847     genuine
## BN84  -1.0086161  0.50809975     genuine
## BN85  -1.0565374  2.19956789     genuine
## BN86  -0.9871093 -0.17948403     genuine
## BN87  -0.9959688 -0.12113534     genuine
## BN88  -0.9698822 -1.16967679     genuine
## BN89  -1.0266944  0.94644878     genuine
## BN90  -1.0003674 -0.15140749     genuine
## BN91  -0.9674696 -0.92508920     genuine
## BN92  -1.0061021  0.27928226     genuine
## BN93  -0.9456286 -1.51243493     genuine
## BN94  -0.9477006 -1.43801699     genuine
## BN95  -0.9685014 -1.06511910     genuine
## BN96  -1.0038010  0.06194035     genuine
## BN97  -1.0276612  1.42447508     genuine
## BN98  -0.9582325 -0.85408995     genuine
## BN99  -0.9966726  0.17645490     genuine
## BN100 -0.9900017 -0.11726058     genuine
## BN101  1.0074505 -0.47932605 counterfeit
## BN102  0.9795680  0.55388206 counterfeit
## BN103  0.9714871 -0.01076140 counterfeit
## BN104  0.9578452  0.51352283 counterfeit
## BN105  1.0056438 -0.18249035 counterfeit
## BN106  0.9903775 -0.19080010 counterfeit
```

```
## BN107  0.9713891  0.07350895 counterfeit
## BN108  0.9985254 -0.35463265 counterfeit
## BN109  1.0049348 -0.32388279 counterfeit
## BN110  0.9400403  1.16009666 counterfeit
## BN111  0.9701443  0.41678228 counterfeit
## BN112  0.9618603  0.63264574 counterfeit
## BN113  0.9181623  1.61029774 counterfeit
## BN114  0.9839940  0.12805108 counterfeit
## BN115  0.9866927 -0.02793601 counterfeit
## BN116  0.9689450  0.30573328 counterfeit
## BN117  0.9693943  0.68664899 counterfeit
## BN118  0.9861979  0.19011270 counterfeit
## BN119  0.9691922  0.51581279 counterfeit
## BN120  1.0182745 -0.43125780 counterfeit
## BN121  0.9915235  0.15638564 counterfeit
## BN122  0.9670739  0.87114833 counterfeit
## BN123  0.9245913  1.83008655 counterfeit
## BN124  0.9490356  1.18318815 counterfeit
## BN125  0.9554858  0.64270793 counterfeit
## BN126  1.0145113 -0.63133296 counterfeit
## BN127  0.9620541  0.41795990 counterfeit
## BN128  0.9397224  1.20813664 counterfeit
## BN129  0.9811110 -0.17084828 counterfeit
## BN130  0.9947604 -0.04539214 counterfeit
## BN131  1.0039770 -0.43314199 counterfeit
## BN132  1.0277410 -0.47515780 counterfeit
## BN133  0.9695378  0.54459175 counterfeit
## BN134  1.0021787 -0.13452592 counterfeit
## BN135  1.0313490 -0.81753941 counterfeit
## BN136  1.0203553 -0.51108163 counterfeit
## BN137  1.0425971 -1.23579166 counterfeit
## BN138  0.9720406  1.11686732 counterfeit
## BN139  0.9832392  0.34129928 counterfeit
## BN140  0.9821984  0.45526023 counterfeit
## BN141  1.0053809 -0.14227735 counterfeit
## BN142  1.0326550 -1.15706639 counterfeit
## BN143  0.9975823 -0.15625997 counterfeit
## BN144  0.9641289  0.66515235 counterfeit
## BN145  1.0493615 -1.48762741 counterfeit
## BN146  0.9639812  0.88778288 counterfeit
## BN147  0.9671891  0.66014716 counterfeit
## BN148  0.9792989  0.95171163 counterfeit
## BN149  1.0071554 -0.28603071 counterfeit
## BN150  1.0324990 -1.02187421 counterfeit
## BN151  0.9957592 -0.12239144 counterfeit
## BN152  1.0495771 -1.26707253 counterfeit
## BN153  1.0604468 -1.97209903 counterfeit
## BN154  1.0239935 -0.57936914 counterfeit
## BN155  1.0046691 -0.12396714 counterfeit
## BN156  1.0142189 -0.60784203 counterfeit
## BN157  1.0603540 -1.88208639 counterfeit
## BN158  1.0271984 -0.64349673 counterfeit
## BN159  0.9914008  0.48713302 counterfeit
## BN160  0.9914464  0.72573974 counterfeit
```

```
## BN161  1.0115803 -0.54647558 counterfeit
## BN162  0.9943155  0.50274327 counterfeit
## BN163  1.0194613 -0.61310613 counterfeit
## BN164  1.0254938 -0.47168301 counterfeit
## BN165  0.9996250 -0.18461487 counterfeit
## BN166  0.9840233  0.31852161 counterfeit
## BN167  0.9409254  1.40817483 counterfeit
## BN168  0.9864312  0.39498110 counterfeit
## BN169  1.0290211 -1.15967513 counterfeit
## BN170  1.0219455 -0.85652581 counterfeit
## BN171  0.9911961  1.05043389 counterfeit
## BN172  0.9768724  0.75443556 counterfeit
## BN173  0.9820739  0.57579109 counterfeit
## BN174  1.0634892 -1.78793332 counterfeit
## BN175  1.0136637 -0.57934908 counterfeit
## BN176  0.9995025  0.30567873 counterfeit
## BN177  1.0189958 -0.62116519 counterfeit
## BN178  0.9896784  0.12670372 counterfeit
## BN179  0.9853526  0.60610420 counterfeit
## BN180  1.0253992 -0.12005925 counterfeit
## BN181  0.9941898  0.26690914 counterfeit
## BN182  0.9642716  1.00617853 counterfeit
## BN183  0.9825430  0.52082890 counterfeit
## BN184  0.9794531  0.60587819 counterfeit
## BN185  1.0057282 -0.28633476 counterfeit
## BN186  0.9944327  0.14985552 counterfeit
## BN187  0.9997380  0.14810541 counterfeit
## BN188  1.0335599 -0.97909166 counterfeit
## BN189  1.0280696 -0.93992323 counterfeit
## BN190  1.0060904 -0.02811401 counterfeit
## BN191  1.0062857 -0.35703258 counterfeit
## BN192  0.9694077  0.87003043 counterfeit
## BN193  1.0022949  0.02120368 counterfeit
## BN194  0.9850741  0.63542262 counterfeit
## BN195  0.9875071  0.20291285 counterfeit
## BN196  0.9798782  0.38610825 counterfeit
## BN197  0.9891717 -0.07070904 counterfeit
## BN198  0.9823772  0.13176922 counterfeit
## BN199  0.9603973  1.33287504 counterfeit
## BN200  1.0363801 -1.20721534 counterfeit
```

K-fold cross-validation after factor analysis (MLE)

Here, we will perform the exact same K-fold cross-validation from above, but on the data set that we performed factor analysis on, and compare the outcomes of the models from the cross-validations to see the effects of factor analysis.

```
mle_lda_accuracy <- c()
mle_lr_accuracy <- c()
# K-fold cross-validation using Linear Discriminant Analysis and Logistic Regression
for(fold in c(2:10)){
  train_control <- trainControl(method="cv", number=fold)
  # train the model
  model_lda <- train(Indicator~., data=mle_indic, trControl=train_control, method="lda")
  model_lr <- train(Indicator~., data=mle_indic, trControl=train_control, method="glm")

  # validate model
  predict_lda <- predict(model_lda, mle_indic)
  predict_lr <-predict(model_lr, mle_indic)

  # create confusion matrix
  Indicator_fac <- as.factor(Indicator)
  conf_lda <- confusionMatrix(predict_lda, Indicator_fac)
  conf_lr <- confusionMatrix(predict_lr, Indicator_fac)

  # summarize results and show confusion matrix
  print(model_lda)
  print(conf_lda)

  print(model_lr)
  print(conf_lr)

  # append accuracies of each method to accuracy list
  mle_lda_accuracy <- c(mle_lda_accuracy, model_lda$results$Accuracy)
  mle_lr_accuracy <- c(mle_lr_accuracy, model_lr$results$Accuracy)
}
```

```
## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge
```

```
## Linear Discriminant Analysis
##
## 200 samples
##   2 predictor
##   2 classes: 'counterfeit', 'genuine'
##
## No pre-processing
## Resampling: Cross-Validated (2 fold)
## Summary of sample sizes: 100, 100
## Resampling results:
##
##   Accuracy  Kappa
##   1         1
##
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    counterfeit genuine
##   counterfeit         100       0
##   genuine               0     100
##
##                Accuracy : 1
##                  95% CI : (0.9817, 1)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0
##             Specificity : 1.0
##          Pos Pred Value : 1.0
##          Neg Pred Value : 1.0
##              Prevalence : 0.5
##          Detection Rate : 0.5
##    Detection Prevalence : 0.5
##       Balanced Accuracy : 1.0
##
##        'Positive' Class : counterfeit
##
## Generalized Linear Model
##
## 200 samples
##   2 predictor
##   2 classes: 'counterfeit', 'genuine'
##
## No pre-processing
## Resampling: Cross-Validated (2 fold)
## Summary of sample sizes: 100, 100
## Resampling results:
##
##   Accuracy  Kappa
```

```
##   1         1
##
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    counterfeit genuine
##   counterfeit         100       0
##   genuine               0     100
##
##                Accuracy : 1
##                  95% CI : (0.9817, 1)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0
##             Specificity : 1.0
##          Pos Pred Value : 1.0
##          Neg Pred Value : 1.0
##              Prevalence : 0.5
##          Detection Rate : 0.5
##    Detection Prevalence : 0.5
##       Balanced Accuracy : 1.0
##
##        'Positive' Class : counterfeit
##
```

```
## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge
```

```
## Linear Discriminant Analysis
##
## 200 samples
##   2 predictor
##   2 classes: 'counterfeit', 'genuine'
##
## No pre-processing
## Resampling: Cross-Validated (3 fold)
## Summary of sample sizes: 134, 132, 134
## Resampling results:
##
##   Accuracy  Kappa
##   1         1
##
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    counterfeit genuine
##   counterfeit         100       0
##   genuine               0     100
##
##                Accuracy : 1
##                  95% CI : (0.9817, 1)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0
##             Specificity : 1.0
##          Pos Pred Value : 1.0
##          Neg Pred Value : 1.0
##              Prevalence : 0.5
##          Detection Rate : 0.5
##    Detection Prevalence : 0.5
##       Balanced Accuracy : 1.0
##
##        'Positive' Class : counterfeit
##
## Generalized Linear Model
##
## 200 samples
##   2 predictor
##   2 classes: 'counterfeit', 'genuine'
##
## No pre-processing
## Resampling: Cross-Validated (3 fold)
## Summary of sample sizes: 133, 133, 134
## Resampling results:
##
##   Accuracy  Kappa
```

```
##   1         1
##
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    counterfeit genuine
##   counterfeit         100       0
##   genuine               0     100
##
##                Accuracy : 1
##                  95% CI : (0.9817, 1)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0
##             Specificity : 1.0
##          Pos Pred Value : 1.0
##          Neg Pred Value : 1.0
##              Prevalence : 0.5
##          Detection Rate : 0.5
##    Detection Prevalence : 0.5
##       Balanced Accuracy : 1.0
##
##        'Positive' Class : counterfeit
##
```

```
## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge
```

```
## Linear Discriminant Analysis
##
## 200 samples
##   2 predictor
##   2 classes: 'counterfeit', 'genuine'
##
## No pre-processing
## Resampling: Cross-Validated (4 fold)
## Summary of sample sizes: 150, 150, 150, 150
## Resampling results:
##
##   Accuracy  Kappa
##   1         1
##
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    counterfeit genuine
##   counterfeit         100       0
##   genuine               0     100
##
##                Accuracy : 1
##                  95% CI : (0.9817, 1)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0
##             Specificity : 1.0
##          Pos Pred Value : 1.0
##          Neg Pred Value : 1.0
##              Prevalence : 0.5
##          Detection Rate : 0.5
##    Detection Prevalence : 0.5
##       Balanced Accuracy : 1.0
##
##        'Positive' Class : counterfeit
##
## Generalized Linear Model
##
## 200 samples
##   2 predictor
##   2 classes: 'counterfeit', 'genuine'
##
## No pre-processing
## Resampling: Cross-Validated (4 fold)
## Summary of sample sizes: 150, 150, 150, 150
## Resampling results:
##
##   Accuracy  Kappa
```

```
##   1         1
##
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    counterfeit genuine
##   counterfeit         100       0
##   genuine               0     100
##
##                Accuracy : 1
##                  95% CI : (0.9817, 1)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0
##             Specificity : 1.0
##          Pos Pred Value : 1.0
##          Neg Pred Value : 1.0
##              Prevalence : 0.5
##          Detection Rate : 0.5
##    Detection Prevalence : 0.5
##       Balanced Accuracy : 1.0
##
##        'Positive' Class : counterfeit
##
```

```
## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge
```

```
## Linear Discriminant Analysis
##
## 200 samples
##   2 predictor
##   2 classes: 'counterfeit', 'genuine'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 160, 160, 160, 160, 160
## Resampling results:
##
##   Accuracy  Kappa
##   1         1
##
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    counterfeit genuine
##   counterfeit         100       0
##   genuine               0     100
##
##                Accuracy : 1
##                  95% CI : (0.9817, 1)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0
##             Specificity : 1.0
##          Pos Pred Value : 1.0
##          Neg Pred Value : 1.0
##              Prevalence : 0.5
##          Detection Rate : 0.5
##    Detection Prevalence : 0.5
##       Balanced Accuracy : 1.0
##
##        'Positive' Class : counterfeit
##
## Generalized Linear Model
##
## 200 samples
##   2 predictor
##   2 classes: 'counterfeit', 'genuine'
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 160, 160, 160, 160, 160
## Resampling results:
##
##   Accuracy  Kappa
```

```
##   1         1
##
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    counterfeit genuine
##   counterfeit         100       0
##   genuine               0     100
##
##                Accuracy : 1
##                  95% CI : (0.9817, 1)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0
##             Specificity : 1.0
##          Pos Pred Value : 1.0
##          Neg Pred Value : 1.0
##              Prevalence : 0.5
##          Detection Rate : 0.5
##    Detection Prevalence : 0.5
##       Balanced Accuracy : 1.0
##
##        'Positive' Class : counterfeit
##
```

```
## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge
```

```
## Linear Discriminant Analysis
##
## 200 samples
##   2 predictor
##   2 classes: 'counterfeit', 'genuine'
##
## No pre-processing
## Resampling: Cross-Validated (6 fold)
## Summary of sample sizes: 166, 167, 167, 166, 166, 168, ...
## Resampling results:
##
##   Accuracy  Kappa
##   1         1
##
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    counterfeit genuine
##   counterfeit         100       0
##   genuine               0     100
##
##                Accuracy : 1
##                  95% CI : (0.9817, 1)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0
##             Specificity : 1.0
##          Pos Pred Value : 1.0
##          Neg Pred Value : 1.0
##              Prevalence : 0.5
##          Detection Rate : 0.5
##    Detection Prevalence : 0.5
##       Balanced Accuracy : 1.0
##
##        'Positive' Class : counterfeit
##
## Generalized Linear Model
##
## 200 samples
##   2 predictor
##   2 classes: 'counterfeit', 'genuine'
##
## No pre-processing
## Resampling: Cross-Validated (6 fold)
## Summary of sample sizes: 166, 166, 167, 166, 167, 168, ...
## Resampling results:
##
##   Accuracy  Kappa
```

```
##   1         1
##
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    counterfeit genuine
##   counterfeit         100       0
##   genuine               0     100
##
##                Accuracy : 1
##                  95% CI : (0.9817, 1)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0
##             Specificity : 1.0
##          Pos Pred Value : 1.0
##          Neg Pred Value : 1.0
##              Prevalence : 0.5
##          Detection Rate : 0.5
##    Detection Prevalence : 0.5
##       Balanced Accuracy : 1.0
##
##        'Positive' Class : counterfeit
##
```

```
## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge
```

```
## Linear Discriminant Analysis
##
## 200 samples
##   2 predictor
##   2 classes: 'counterfeit', 'genuine'
##
## No pre-processing
## Resampling: Cross-Validated (7 fold)
## Summary of sample sizes: 171, 171, 172, 171, 171, 172, ...
## Resampling results:
##
##   Accuracy  Kappa
##   1         1
##
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    counterfeit genuine
##   counterfeit         100       0
##   genuine               0     100
##
##                Accuracy : 1
##                  95% CI : (0.9817, 1)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0
##             Specificity : 1.0
##          Pos Pred Value : 1.0
##          Neg Pred Value : 1.0
##              Prevalence : 0.5
##          Detection Rate : 0.5
##    Detection Prevalence : 0.5
##       Balanced Accuracy : 1.0
##
##        'Positive' Class : counterfeit
##
## Generalized Linear Model
##
## 200 samples
##   2 predictor
##   2 classes: 'counterfeit', 'genuine'
##
## No pre-processing
## Resampling: Cross-Validated (7 fold)
## Summary of sample sizes: 170, 172, 171, 172, 172, 172, ...
## Resampling results:
##
##   Accuracy  Kappa
```

```
##   1         1
##
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    counterfeit genuine
##   counterfeit         100       0
##   genuine               0     100
##
##                Accuracy : 1
##                  95% CI : (0.9817, 1)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0
##             Specificity : 1.0
##          Pos Pred Value : 1.0
##          Neg Pred Value : 1.0
##              Prevalence : 0.5
##          Detection Rate : 0.5
##    Detection Prevalence : 0.5
##       Balanced Accuracy : 1.0
##
##        'Positive' Class : counterfeit
##
```

```
## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge
```

```
## Linear Discriminant Analysis
##
## 200 samples
##   2 predictor
##   2 classes: 'counterfeit', 'genuine'
##
## No pre-processing
## Resampling: Cross-Validated (8 fold)
## Summary of sample sizes: 175, 174, 175, 175, 176, 176, ...
## Resampling results:
##
##   Accuracy  Kappa
##   1         1
##
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    counterfeit genuine
##   counterfeit         100       0
##   genuine               0     100
##
##                Accuracy : 1
##                  95% CI : (0.9817, 1)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0
##             Specificity : 1.0
##          Pos Pred Value : 1.0
##          Neg Pred Value : 1.0
##              Prevalence : 0.5
##          Detection Rate : 0.5
##    Detection Prevalence : 0.5
##       Balanced Accuracy : 1.0
##
##        'Positive' Class : counterfeit
##
## Generalized Linear Model
##
## 200 samples
##   2 predictor
##   2 classes: 'counterfeit', 'genuine'
##
## No pre-processing
## Resampling: Cross-Validated (8 fold)
## Summary of sample sizes: 175, 176, 175, 174, 175, 174, ...
## Resampling results:
##
##   Accuracy  Kappa
```

```
##   1         1
##
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    counterfeit genuine
##   counterfeit         100       0
##   genuine               0     100
##
##                Accuracy : 1
##                  95% CI : (0.9817, 1)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0
##             Specificity : 1.0
##          Pos Pred Value : 1.0
##          Neg Pred Value : 1.0
##              Prevalence : 0.5
##          Detection Rate : 0.5
##    Detection Prevalence : 0.5
##       Balanced Accuracy : 1.0
##
##        'Positive' Class : counterfeit
##
```

```
## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge
```

```
## Linear Discriminant Analysis
##
## 200 samples
##   2 predictor
##   2 classes: 'counterfeit', 'genuine'
##
## No pre-processing
## Resampling: Cross-Validated (9 fold)
## Summary of sample sizes: 178, 178, 178, 177, 177, 178, ...
## Resampling results:
##
##   Accuracy  Kappa
##   1         1
##
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    counterfeit genuine
##   counterfeit         100       0
##   genuine               0     100
##
##                Accuracy : 1
##                  95% CI : (0.9817, 1)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0
##             Specificity : 1.0
##          Pos Pred Value : 1.0
##          Neg Pred Value : 1.0
##              Prevalence : 0.5
##          Detection Rate : 0.5
##    Detection Prevalence : 0.5
##       Balanced Accuracy : 1.0
##
##        'Positive' Class : counterfeit
##
## Generalized Linear Model
##
## 200 samples
##   2 predictor
##   2 classes: 'counterfeit', 'genuine'
##
## No pre-processing
## Resampling: Cross-Validated (9 fold)
## Summary of sample sizes: 178, 177, 178, 178, 177, 178, ...
## Resampling results:
##
##   Accuracy  Kappa
```

```
##   1         1
##
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    counterfeit genuine
##   counterfeit         100       0
##   genuine               0     100
##
##                Accuracy : 1
##                  95% CI : (0.9817, 1)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0
##             Specificity : 1.0
##          Pos Pred Value : 1.0
##          Neg Pred Value : 1.0
##              Prevalence : 0.5
##          Detection Rate : 0.5
##    Detection Prevalence : 0.5
##       Balanced Accuracy : 1.0
##
##        'Positive' Class : counterfeit
##
```

```
## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge
```

```
## Linear Discriminant Analysis
##
## 200 samples
##   2 predictor
##   2 classes: 'counterfeit', 'genuine'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 180, 180, 180, 180, 180, 180, ...
## Resampling results:
##
##   Accuracy  Kappa
##   1         1
##
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    counterfeit genuine
##   counterfeit         100       0
##   genuine               0     100
##
##                Accuracy : 1
##                  95% CI : (0.9817, 1)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0
##             Specificity : 1.0
##          Pos Pred Value : 1.0
##          Neg Pred Value : 1.0
##              Prevalence : 0.5
##          Detection Rate : 0.5
##    Detection Prevalence : 0.5
##       Balanced Accuracy : 1.0
##
##        'Positive' Class : counterfeit
##
## Generalized Linear Model
##
## 200 samples
##   2 predictor
##   2 classes: 'counterfeit', 'genuine'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 180, 180, 180, 180, 180, 180, ...
## Resampling results:
##
##   Accuracy  Kappa
```

```
##   1         1
##
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    counterfeit genuine
##   counterfeit         100       0
##   genuine               0     100
##
##                Accuracy : 1
##                  95% CI : (0.9817, 1)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0
##             Specificity : 1.0
##          Pos Pred Value : 1.0
##          Neg Pred Value : 1.0
##              Prevalence : 0.5
##          Detection Rate : 0.5
##    Detection Prevalence : 0.5
##       Balanced Accuracy : 1.0
##
##        'Positive' Class : counterfeit
##
```

```
results = cbind(results, mle_lda_accuracy, mle_lr_accuracy)
results
```

```
##       num_fold lda_accuracy lr_accuracy mle_lda_accuracy mle_lr_accuracy
##  [1,]        2    0.9950000   0.9900000                1               1
##  [2,]        3    0.9949495   0.9849992                1               1
##  [3,]        4    0.9950000   0.9800000                1               1
##  [4,]        5    0.9950000   0.9850000                1               1
##  [5,]        6    0.9947917   0.9748960                1               1
##  [6,]        7    0.9948980   0.9802838                1               1
##  [7,]        8    0.9947917   0.9799679                1               1
##  [8,]        9    0.9949495   0.9850681                1               1
##  [9,]       10    0.9950000   0.9800000                1               1
```

By comparing the results of LDA classification model and logistic regression classification from both before and after the reduction of dimensions through maximum likelihood estimation, we are able to observe that the both classification models show a slightly more accuracy when trained on dataset preprocessed through maximum likelihood estimation, which is done to remove any attributes that are redundant or do not contribute to the outcome, thus excluding over-fitting issues.

Although the increase in accuracy can be seen as a very ignorably small amount, there is a significance in this increase because before MLE, the models would get one or two predictions wrong. However, after the preprocessing of MLE on the dataset, we can see none of those errors.

We can realize from the above comparison that factor analysis does help in increasing the accuracy of models LDA and logistic regression by slight amounts, and therefore factor analysis is definitely not a waste of time. For each folds when training models on raw, unprocessed datasets, LDA is definitely the model that yields the better answer. After preprocessing factor analysis through MLE, both LDA and logistic regression classification models yield similar accuracies. Then the better model would be the logistic regression model.

This is because even though they both yield similar results, the binary logistic regression (BLR) classification model has less constraints on the dataset and conditions. First, the BLR model is not so exigent to the level of the scale and form of distributions in predictors, where as the LDA desires interval levels with multivariate normal distribution. Second, the BLR model has no requirements about within-group covariance matrices of the predictors, where as the LDA covariance matrices should be identical to that of the population. Third, the BLR is much less sensitive to outliers, whereas the LDA is very sensitive to outliers. If any of these conditions that are not easy to satisfy all in reality are not met for LDA, the model has a chance to produce misleading results.

Conclusion

Our question was: Can we predict whether a note is false or counterfeit using supervised learning? The answer is yes, through building a model of K-fold cross-validation: implementing Linear Discrimimant Analysis (LDA) and (Binary) Logistic Regression (BLR) for each fold. Both of these models were able to make predictions that very much accurately guess the genuinenity of each bank note. However after factor analyzing the dataset through the Maximum Likelihood estimator and then performing K-fold cross-validation on that processed dataset resulted to show models with even better accuracies. From this Final Project, we are able to conclude that we can predict whether a note is false or counterfeit using supervised learning like LDA and BLR, and also that factor analysis to reduce the dimension and remove and redundancy through MLE, does have a significant effect on increasing the accuracy of the LDA and BLR models from K-fold cross-validation. Therefore, future research could explore specifically which variables are best fitted, redundant, or not needed to predict the genuineity of a bank note to further increase the accuracy with even larger datasets.

However a caveat is that this conclusion does need to be approached with caution since we made 4 assumptions in the beginning of LDA and 1 before MLE. And for our conclusion to hold true, these assumptions must be true.