Math 189: Final Project

Zhenjian Wang A15880200

```
defaultW <- getOption("warn")

options(warn = -1)
```

Introduction

Our study is about predicting whether a note is false or counterfeit using supervised learning. We are using the Swiss banknote datasets. The dataset contains six variables measured on 100 genuine and 100 counterfeit old Swiss 1000-franc banknotes:

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

We will be using K-fold cross-validation, LDA, logistic regression, PCA, factor model in this study, and we will present the results and conclusion at the end.

Descriptive Statistics

Import Data

```
swiss <- read.table("D:\\ucsd\\math189\\ma189-main\\Data\\SBN.txt")
swiss
```

```
swiss$label = c(rep(0, 100),rep(1,100))
swiss
```

The sample mean of the Genuine sample and the Counterfeit sample.

```
mu_mat <- cbind(colMeans(swiss[1:100,]),colMeans(swiss[101:200,]))
colnames(mu_mat) <- c("Genuine Sample Mean","Counterfeit Sample Mean")
mu_mat
```

```
##          Genuine Sample Mean Counterfeit Sample Mean
## Length               214.969                 214.823
## Left                 129.943                 130.300
## Right                129.720                 130.193
## Bottom                 8.305                  10.530
## Top                   10.168                  11.133
## Diagonal             141.517                 139.450
## label                  0.000                   1.000
```

The variance-covariance matrix of the Genuine sample and the Counterfeit sample.

```
var_genuine <- var(swiss[1:100,])
var_counterfeit <- var(swiss[101:200,])
var_genuine
```

```
##               Length        Left       Right        Bottom         Top
## 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   0.057126263  0.05665152  0.05818182  0.4132070707 -0.26347475
## Top      0.014452525  0.04906667  0.03064646 -0.2634747475  0.42118788
## Diagonal 0.005481818 -0.04306162 -0.02377778 -0.0001868687 -0.07530909
## label    0.000000000  0.00000000  0.00000000  0.0000000000  0.00000000
##               Diagonal label
## Length    0.0054818182     0
## Left     -0.0430616162     0
## Right    -0.0237777778     0
## Bottom   -0.0001868687     0
## Top      -0.0753090909     0
## Diagonal  0.1998090909     0
## label     0.0000000000     0
```

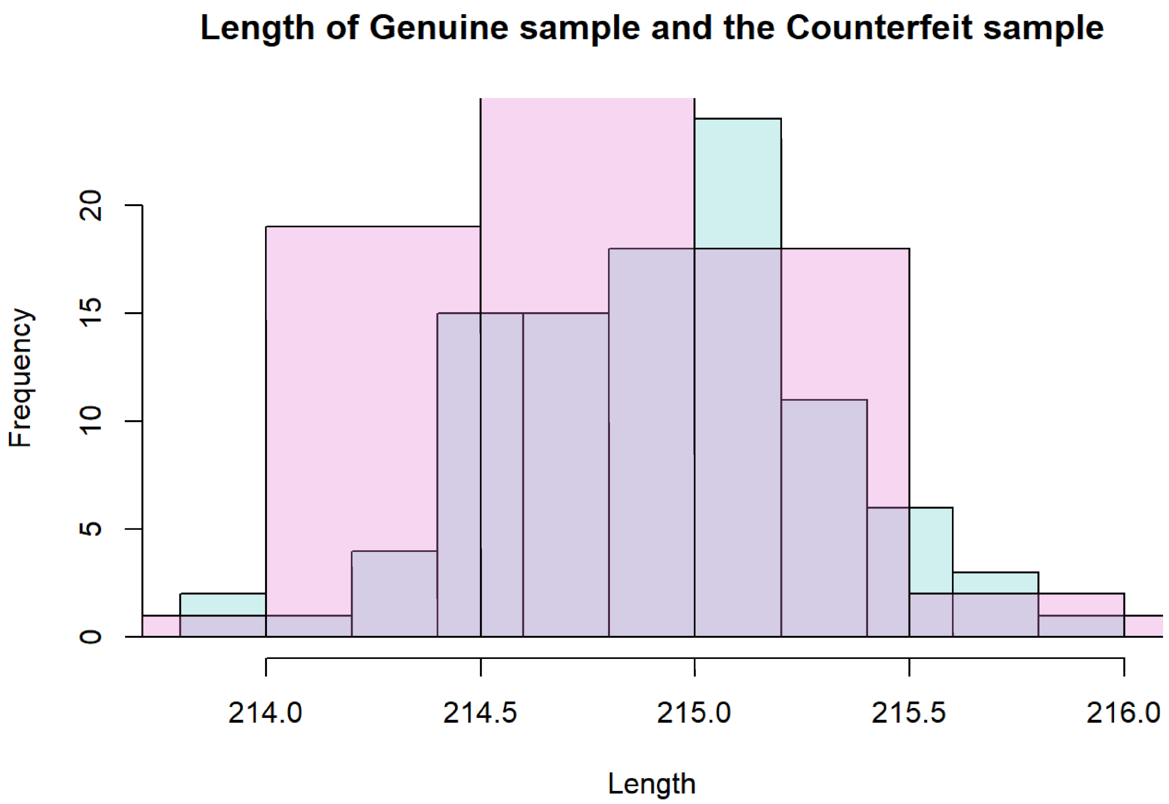
```
var_counterfeit
```

```
##               Length         Left         Right      Bottom           Top
## Length    0.12401111  0.031515152  0.0240010101 -0.10059596  0.0194353535
## Left      0.03151515  0.065050505  0.0467676768 -0.02404040 -0.0119191919
## Right     0.02400101  0.046767677  0.0889404040 -0.01857576  0.0001323232
## Bottom   -0.10059596 -0.024040404 -0.0185757576  1.28131313 -0.4901919192
## Top       0.01943535 -0.011919192  0.0001323232 -0.49019192  0.4044555556
## Diagonal  0.01156566 -0.005050505  0.0341919192  0.23848485 -0.0220707071
## label     0.00000000  0.000000000  0.0000000000  0.00000000  0.0000000000
##              Diagonal label
## Length    0.011565657     0
## Left     -0.005050505     0
## Right     0.034191919     0
## Bottom    0.238484848     0
## Top      -0.022070707     0
## Diagonal  0.311212121     0
## label     0.000000000     0
```

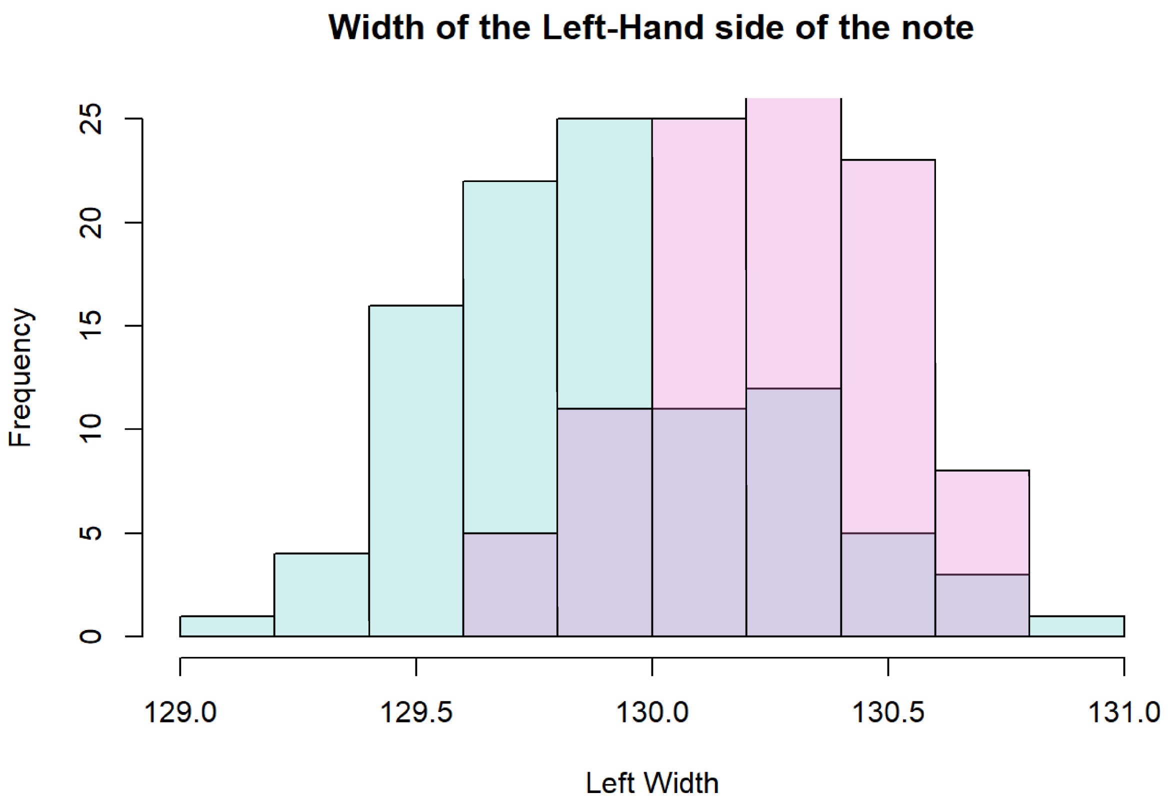
Visualization

Histogram of the variables in the samples. The color of the Genuine sample: Blue The Color of Counterfeit sample: Red

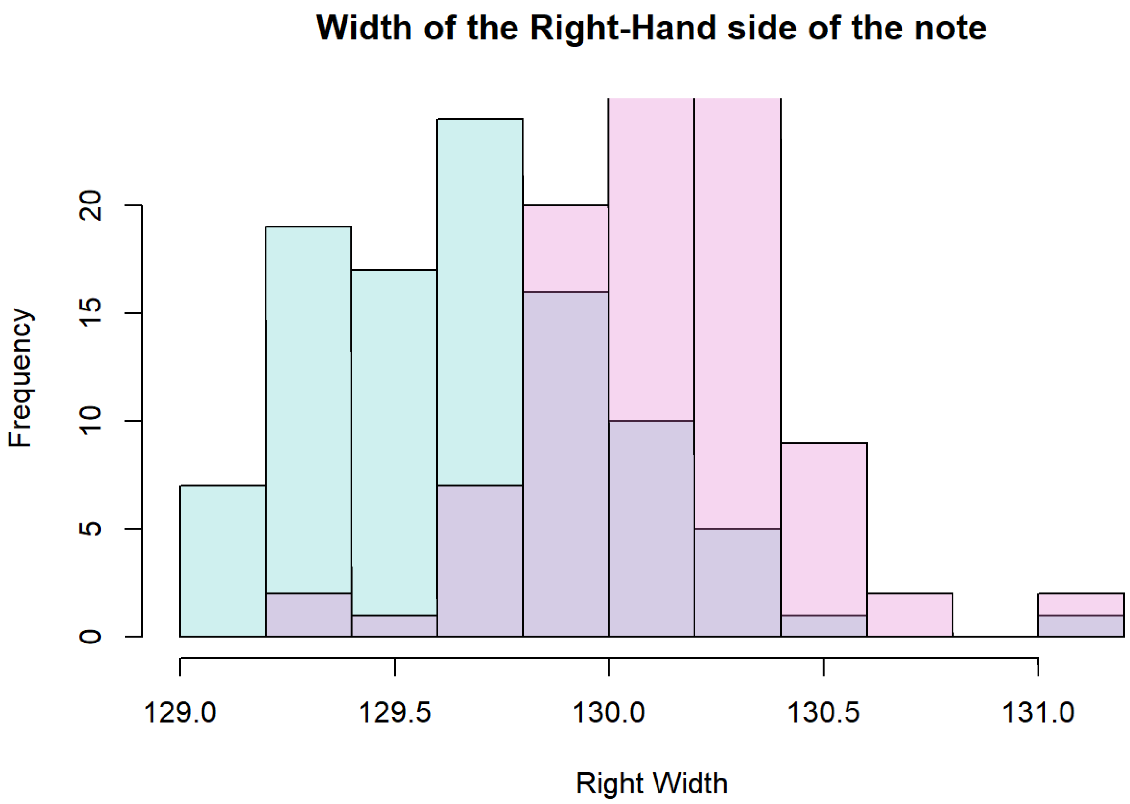
```
c1 <- rgb(100,213,210,max = 265, alpha = 80, names = "lt.blue")
c2 <- rgb(235,125,213, max = 265, alpha = 80, names = "lt.red")
hist(swiss[1:100,]$Length, col = c1, main = 'Length of Genuine sample and th
e Counterfeit sample', xlab = 'Length')
hist(swiss[101:200,]$Length, col = c2, add = TRUE)
```



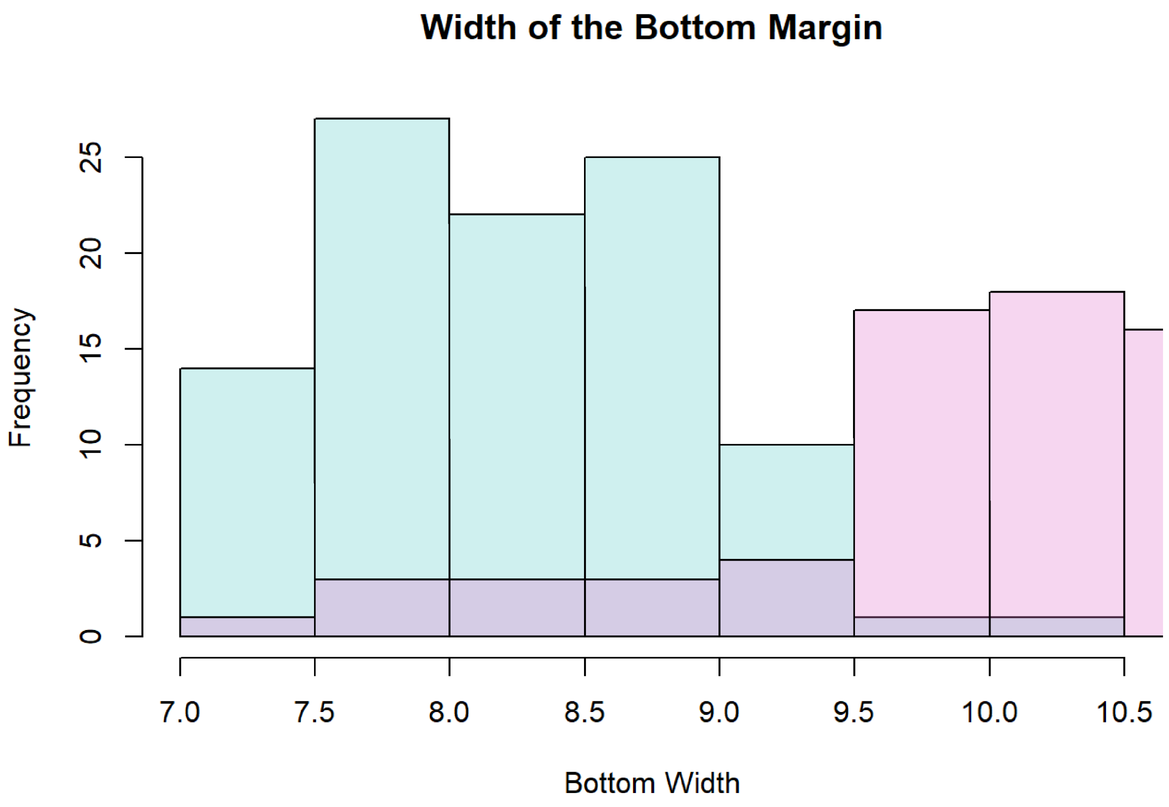
```
hist(swiss[1:100,]$Left, col = c1, main = 'Width of the Left-Hand side of th
e note', xlab = 'Left Width')
hist(swiss[101:200,]$Left, col = c2, add = TRUE)
```



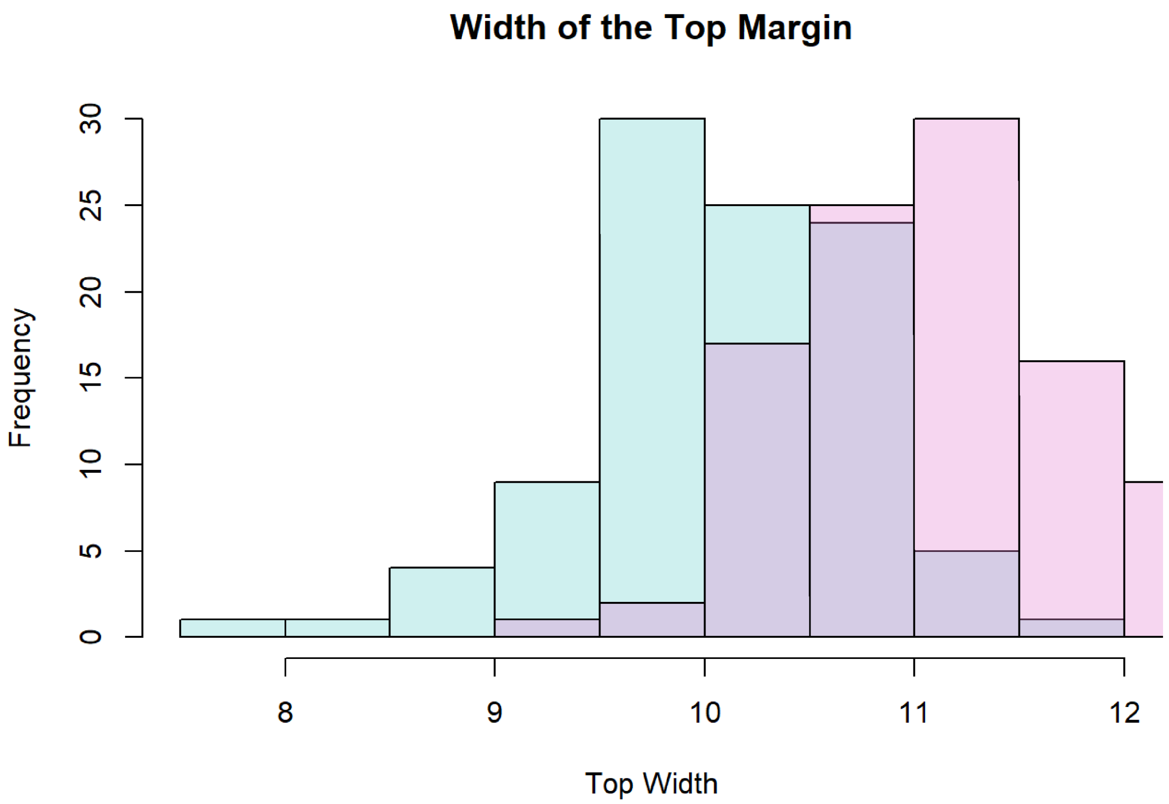
```
hist(swiss[1:100,]$Right, col = c1, main = 'Width of the Right-Hand side of
the note', xlab = 'Right Width')
hist(swiss[101:200,]$Right, col = c2, add = TRUE)
```



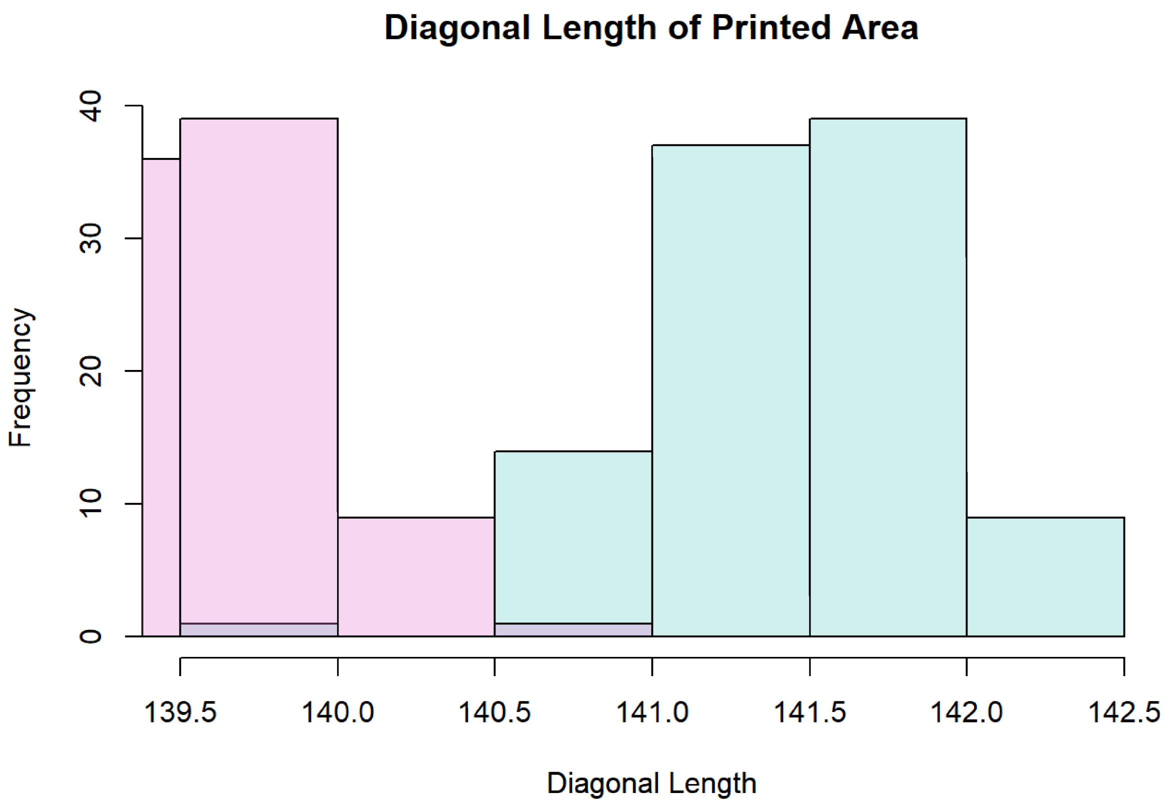
```
hist(swiss[1:100,]$Bottom, col = c1, main = 'Width of the Bottom Margin', xl
ab = 'Bottom Width')
hist(swiss[101:200,]$Bottom, col = c2, add = TRUE)
```



```
hist(swiss[1:100,]$Top, col = c1, main = 'Width of the Top Margin', xlab =
'Top Width')
hist(swiss[101:200,]$Top, col = c2, add = TRUE)
```



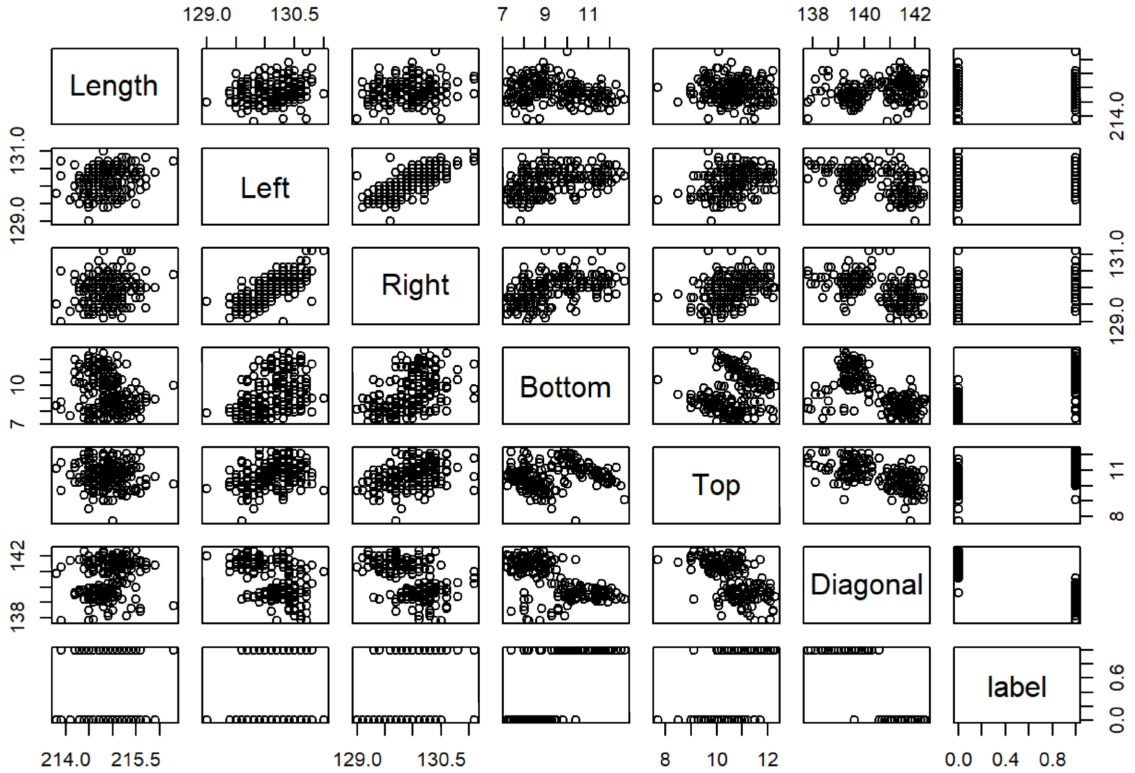
```
hist(swiss[1:100,]$Diagonal, col = c1, main = 'Diagonal Length of Printed Ar
ea', xlab = 'Diagonal Length')
hist(swiss[101:200,]$Diagonal, col = c2, add = TRUE)
```



From the histograms above, we could clearly see the difference between the genuine sample and the counterfeit sample by looking at the two colors.

Then we are presenting a pairwise scatterplot to show if there is any correlation between the two variables.

```
pairs(swiss)
```



We could see that left and right are highly correlated, and other variables may be weakly related. Besides, the scatter plot that shows the relationship between Diagonal and any other variable looks interesting. It looks like that it contains two clusters. We could guess that Diagonal could possibly be used to distinguish the genuine sample and the counterfeit samples.

K-fold cross-validation, LDA and Logistic Regression

We are now going to use K-fold cross-validation and apply the LDA and Logistic Regression to the data by the package called caret. we have 200 rows in our data.

we use k = 4:

```
library(caret)
```

```
## Loading required package: lattice
```

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

we then apply LDA and logistic regression to each fold

```
# K-fold CV
fitControl <- trainControl(method = "cv",
                           number = 4,
                           )
# LDA
set.seed(123)
swiss$label = as.factor(swiss$label)
lda.fit = train(label ~ ., data=swiss, method="lda",
                trControl = fitControl)


# Logistic Regression

glm.fit = train(label ~ ., data=swiss, method="glm",
                trControl = fitControl)

lda_accuracy = lda.fit$resample$Accuracy
glm_accuracy = glm.fit$resample$Accuracy

df = data.frame(c('fold1','fold2','fold3','fold4'),lda_accuracy,glm_accurac
y)
df
```

we put the results in a data frame called df. We output the accuracy of the LDA model, which is 1.00, 1.00, 0.98, 1.00 for fold 1,2,3 and 4. This is good because the accuracy is perfect for 3 fold and only 1 fold has a minor mistake.

We also output the accuracy of the logistic regression model, which is 0.98, 1.00, 0.98, 1.00 for fold 1, 2, 3 and 4. This is also good because we only have 2 folds that are not perfect and they have high accuracy.

Factor Model

Principal component analysis First of all, we get rid of the label colume.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
swiss_no_label = select(swiss,Length,Left,Right,Bottom,Top,Diagonal)
```

```
#"scale=True" means the data are standardized first

pca_result <- prcomp(swiss_no_label, scale = TRUE)
pca_var <- pca_result$sdev^2
pve <- pca_var/sum(pca_var)
out2 <- cbind(pca_var,pve,cumsum(pve))
colnames(out2) <- c("Eigenvalue","Proportion","Cumulative")
rownames(out2) <- c("PC1","PC2","PC3","PC4","PC5","PC6")
out2
```

```
##     Eigenvalue Proportion Cumulative
## PC1  2.9455582 0.49092637  0.4909264
## PC2  1.2780838 0.21301396  0.7039403
## PC3  0.8690326 0.14483876  0.8487791
## PC4  0.4497687 0.07496145  0.9237405
## PC5  0.2686769 0.04477948  0.9685200
## PC6  0.1888799 0.03147998  1.0000000
```

From the “Cumulative” column, we could clearly see that PC1 and PC2 could cover 70.39 percent of variance and 84.88 percent of the variance is covered by PC 1,2 and 3, which are good.

```
t(pca_result$rotation)
```

```
##           Length       Left      Right     Bottom         Top   Diagonal
## PC1  0.006987029 -0.4677582 -0.4866787 -0.4067583 -0.36789112  0.4934583
## PC2 -0.815494969 -0.3419671 -0.2524586  0.2662288  0.09148667 -0.2739407
## PC3  0.017680661 -0.1033829 -0.1234747 -0.5835383  0.78757147 -0.1138754
## PC4  0.574617276 -0.3949225 -0.4302783  0.4036735  0.11022672 -0.3919305
## PC5 -0.058796102  0.6394961 -0.6140972 -0.2154756 -0.21984942 -0.3401601
## PC6  0.031056981 -0.2977477  0.3491529 -0.4623536 -0.41896754 -0.6317985
```

From the table above, we could see that PC1 focuses more on the Left, right and bottom and PC2 focuses more on Length.

So We will choose to use two factors and the model is as following:

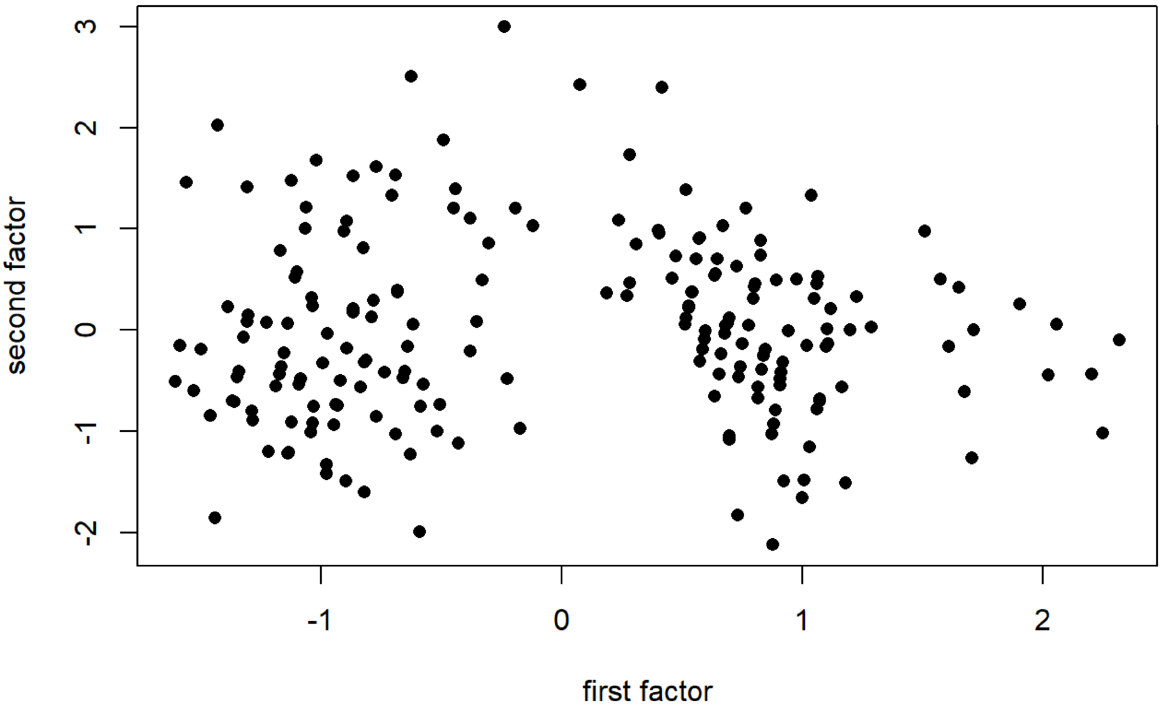
```
# MLE for factor model
n.factors = 2
fa_fit = factanal(swiss_no_label, n.factors,scores = 'regression',rotation
= 'varimax')
fa_fit$loadings
```

```
##
## Loadings:
##          Factor1 Factor2
## Length   -0.167   0.431
## Left      0.553   0.711
## Right     0.560   0.614
## Bottom    0.632   0.105
## Top       0.599
## Diagonal -0.995
##
##                Factor1 Factor2
## SS loadings      2.397   1.085
## Proportion Var   0.399   0.181
## Cumulative Var   0.399   0.580
```

we could see that factor 1 assigns high loading to diagonal, with 0.995. The second factor puts more weight on the Left and Right the most.

the factor scores:

```
score = fa_fit$scores
plot(score[,1],score[,2],pch = 16, xlab = 'first factor', ylab = 'second factor')
```



```
score_df = as.data.frame(score)
```

From the graph, we could clearly see the data points gathering into two clusters. One is on the left and one is on the right, and the left cluster has a negative first factor and the right cluster has a positive first factor. Thus, the first factor is very important for our training model.

Assumptions

Assumptions for PCA:

1. variables are continuous;
2. there’s a linear relationship between all variables;
3. sampling adequacy;
4. no significant outliers.

Assumptions for the Factor Model:

- Assumptions on mean vector:

1. The common factors all have mean zero: \[{\mathbb E} [\underline{f}] = 0. \]
2. The specific factors (or random errors) all have mean zero: \[{\mathbb E} [\underline{f}] = 0. \]

- Assumptions on variance-covariance matrix:

1. Common factors satisfy: \[\mbox{Cov} [\underline{f}] = I_m, \] where \(I_m\) is an \(m \times m\)-dimensional identity matrix.
2. Random errors satisfy: \[\mbox{Cov} [\underline{\epsilon}] = {\mathbf \Psi} = \mbox{diag} (\psi_1, \ldots, \psi_p). \]
3. Common factors and random errors are uncorrelated: \[\mbox{Cov} [\underline{f}, \underline {\epsilon}] = 0. \]

Assumptions for the LDA:

1. The data from group \(k\) has common mean vector \(\underline{\mu}^{(k)}\), i.e., \[{\mathbb E} [x_{ij}^{(k)}] = \underline{\mu}_j^{(k)}. \] (The \(m\) components of the vector correspond to the \(m\) variables.)
2. Homoskedasticity: The data from all groups have common covariance matrix \({\mathbf \Sigma}\), i.e., \[{\mathbf \Sigma} = \mbox{Cov} [\underline{x}_i^{(k)}, \underline{x}_i^{(k)}] \] for any record \(i\), and the matrix does not depend on \(k\) (the group index).
3. Independence: The observations are independently sampled.
4. Normality: The data are multivariate normally distributed.

Run LDA and Logistic Regression on factor scores

```
score_df$label = swiss$label
score_df
```

```
# LDA

set.seed(123)

score_df$label = as.factor(score_df$label)

lda.fit_0 = train(label ~ ., data=score_df, method="lda",
                trControl = fitControl)



# Logistic Regression

glm.fit_1 = train(label ~ ., data=score_df, method="glm",
                trControl = fitControl)

new_lda = lda.fit_0$resample$Accuracy
new_glm = glm.fit_1$resample$Accuracy

new_lda
```

```
## [1] 1.00 1.00 0.98 1.00
```

```
new_glm
```

```
## [1] 1.00 1.00 0.98 1.00
```

Results

```
df_new = data.frame(c('fold1','fold2','fold3','fold4'),new_lda,new_glm)

df_new
```

```
df
```

we output the data frame contains new LDA and new GLM accuracy, and the original data frame that shows the old accuracy.

we could see there is no change in the LDA model and some improvements in the logistic regression model. The fold 1’s accuracy improves from 0.98 to 1.00

Thus, we could say that the factor analysis is useful, and the LDA and GLM are both very accurate with 1.00 accuracy on fold 1, 2, and 4 and 0.98 accuracy on the 3rd fold.

Conclusion

Our study is about whether or if we could predict a note is false or counterfeit using supervised learning. Firstly we give labels to the data where 0 means genuine notes and 1 means counterfeit notes. Then we used k-fold cross-validation to put data into training sets and test sets, dividing them into 4 folds. On each fold, we applied both LDA and logistic regression models, and we have some high accuracy on both models. For the LDA model, we have 1.00, 1.00, 0.98, 1.00 for fold 1,2,3 and 4. For the logistic regression model, we have 0.98, 1.00, 0.98, 1.00 for fold 1,2,3 and 4. Afterward, we use PCA to find if we could possibly reduce any variables, and we see that pc1 and pc2 could cover 70.39 percent of the variance. We choose to reduce the dimension to two and use the factor model. We get out factor scores and run the LDA and logistic regression model again, and we see that there is no change in the LDA model and some improvements in the logistic regression model. We have an accuracy of 1.00 for three folds and 0.98 for 1 fold for both models, which are accurate. For future work, we may want to explore how to improve the overall accuracy to 1.00.

Data Citation

the dataset was downloaded from https://github.com/tuckermcelroy/ma189/blob/main/Data/SBN.txt (https://github.com/tuckermcelroy/ma189/blob/main/Data/SBN.txt)

```
options(warn = defaultW)
```