Statistical Consulting Service

Department of Statistics

Faculty of Applied Sciences

University of Sri Jayewardenepura

Report for: P.A.D. Savindi Prathibha

Date: 3rd March 2023

Client Information

Name: P.A.D. Savindi Prathibha

Department: Department of Food Science and Technology, Faculty of Applied Sciences, University of Sri Jayewardenepura

E-mail: savi.prathibha@gmail.com

Internal Supervisor

Dr. W.L. Isuru Wijesekara

Department of Food Science and Technology, Faculty of Applied Sciences, University of Sri Jayewardenepura

isuruw@sci.sjp.ac.lk

External Supervisor

Mrs. Champa K. Dissanayake

Director, Life Sciences Division, Sri Lanka Atomic Energy Board

Project Team Members

SCS Coordinator: - Dr. Thiyanga Talagala

ttalagala@sjp.ac.lk

1. Sadrushi Dissanayake
2. Thisaakhya Jayakody
3. Lakna Perera
4. Menasha Senanayaka
5. Kalani Siriwardena
6. Trishika Wickramarathne

Contents

Abbreviations 4
Report Summary 5
Project Timeline 5
Section 1- Introduction 5
1.1 Background of the Study 5
1.2 Objectives of the Study 6
1.3 Description of Variables 6
Section 2 - Data Structure 6
Section 3 – Data Analysis Methodology 7
3.1 Data Filtering 7
3.2 Principal Component Analysis (PCA) 7
3.3 Quadratic Discriminant Analysis (QDA) 8
3.4 Partial Least Squares (PLS) Regression 8
Section 4 – Guidelines for using R Software 8
4.1 Order of running scripts 8
4.2 Use the following datasets for PC_QD_Analysis1 and PLS_Analysis1. 8
4.3 Use the following dataset for PC_QD_Analysis2 and PLS_Analysis2 8
4.4 Seven outputs will be produced from the function PC_QD_Analysis1. They are, ... 8
4.5 One outputs will be produced from the function PC_QD_Analysis2. They are, ... 8
4.6 Three outputs will be produced from the PLS_Analysis1. They are, ... 9
4.7 One output will be produced from the PLS_Function2. They are, ... 9
Section 5 – Outputs 9
5.1 PC_QD_Analysis1 Outputs 9
5.2 PC_QD_Analysis2 Outputs 13
5.3 PLS_Analysis1 Output 14
5.4 PLS_Analysis2 Output 16
References 16

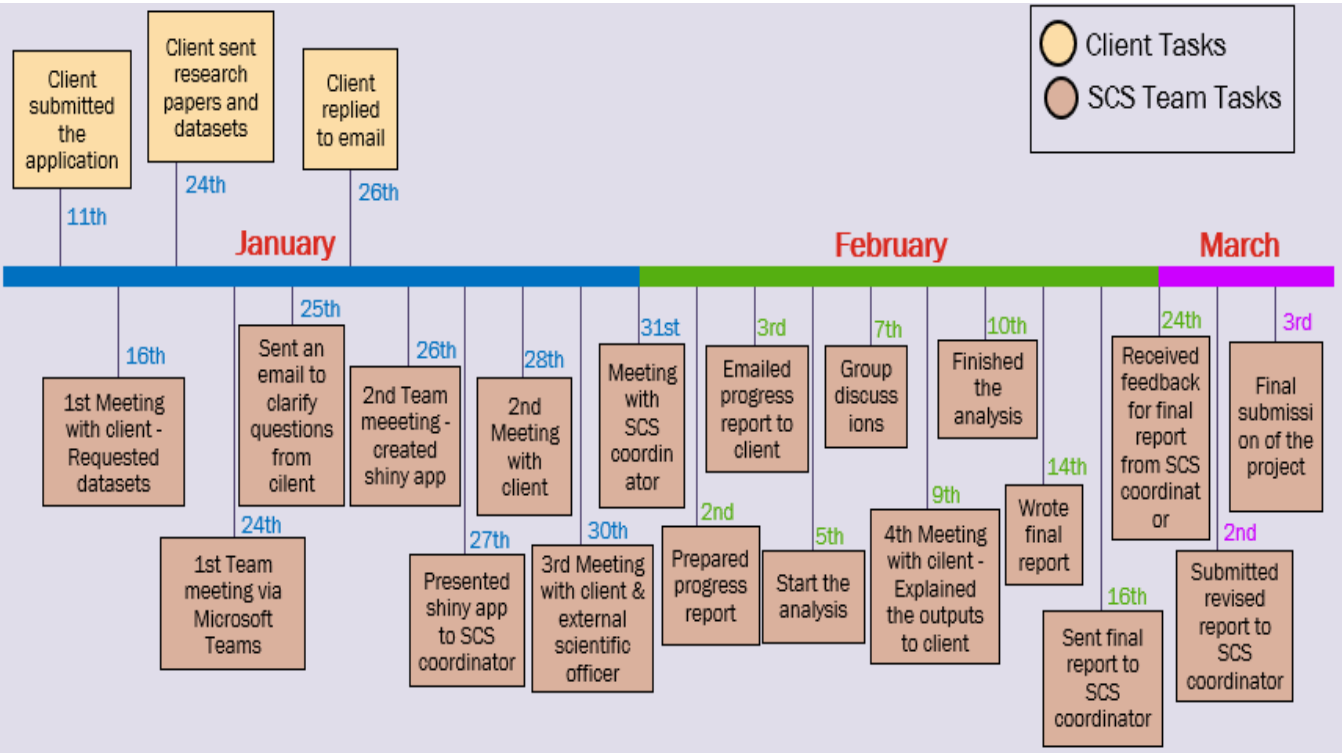
Abbreviations

- PCA - Principal Component Analysis
- PC - Principal Component
- LDA - Linear Discriminant Analysis
- QDA - Quadratic Discriminant Analysis
- RMSE - Root Mean Squared Error
- PLS - Partial Least Squares

Report Summary

This report explains in detail the statistical methodology that SCS followed to conduct the analysis requested by the client. The analysis was done using R programming. This report also explains the guidelines on how the client can understand and use the R codes used for the analysis.

Project Timeline



Section 1- Introduction

1.1 Background of the Study

Lankan Coconut Oil is of high demand in Sri Lanka as well as foreign countries, mainly due to its numerous health benefits. However, production of Lankan Coconut Oil is quite costly when compared to alternatives such as Palm Oil , Palm Kernal Oil and imported edible oils. Therefore, original coconut Oil is often adulterated with cheaper alternatives in order to minimize the production cost. This could result in food fraud, which could harm customers and damage customer trust.

Further, selling Palm Oil under the ‘Coconut Oil’ label is another problem associated in the Sri Lankan market. This could lead to numerous health risks to consumers. According to multiple studies, consuming Palm Oil is said to increase the risk of Cardiovascular Disease, Low-Density Lipoprotein Cholesterol, Ischemic Heart Disease Mortality, and bring many other negative impacts.

Hence, it is vital to investigate whether original Coconut Oil, known as Virgin Coconut Oil (VCO) is adulterated with any other type of Oil before it reaches the market. Developing an analytical approach for screening such adulterations efficiently and accurately will be helpful. Therefore, this study aims to build procedures to examine if a given solution of VCO is adulterated with Pure Palm Oil or not.

1.2 Objectives of the Study

- To analyze the ATR-FTIR spectrum of pure VCO - the purpose of this analysis is to identify the characteristic peaks of VCO.
- To analyze the ATR-FTIR spectrum of Pure Palm Oil - the purpose of this analysis is to identify the characteristic peaks of palm Oil.
- Identify the characteristic peaks of palm Oil - this will help in differentiating the adulterated VCO sample from pure VCO sample.
- To develop a model to determine the palm Oil content in VCO (this is done in connection with PLS regression).

1.3 Description of Variables

Virgin Coconut Oil (VCO), Pure Palm Oil and 12 adulterated samples with Palm Oil concentrations consistently increasing from 5% to 60% are considered.

Dependent Variable

1. Absorption - This is the absorption for a given wavelength in the spectrum

Independent Variables

1. Wavelength (cm^{-1}) – This is the wavelength against which the absorption is plotted
2. Concentration (v/v) – The Palm Oil concentration mixed with Virgin Coconut Oil
3. Replicates – The number of replicates for VCO, Pure Palm Oil and each adulterated sample.

Section 2 - Data Structure

The client provided us with data pertaining to Pure VCO, Adulterated VCO and Pure Palm Oil, in 3 separate excel workbooks. Data relating to each replicate (each sample) under these three types of Oil were entered in a separate sheet. Since it is difficult to conduct the analysis using the data structure provided by the client due to inconsistencies and high probability of error, we organized the data according to the structure shown in Figure 2. 1.

Series	Palm olein concentration(C)	Replicate No	Wave Number (cm-1)(W)	Absorption(A)
Pure Palm Oil	1	1	5500	0.0017
Pure Palm Oil	1	1	5499	0.0017
Pure Palm Oil	1	1	5498	0.0018
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
Pure Palm Oil	0	15	5499	0.0017
Pure Palm Oil	0	15	5498	0.0018

Figure 2. 1: Tidy Data Structure

As shown in Figure 2.1, all workbooks were combined into a single excel sheet. All variables such as Wave Number and Absorption were entered in columns, with each replicate of each oil type occupying a single row. The replicates of each oil type were entered in the order, Pure Palm Oil, Adulterated VCO and Pure VCO. The observed values for each replicated were recorded in the cells.

The number of replicates per oil type were as follows,

- Pure VCO – 15 replicates
- Pure Palm Oil - 15 replicates
- Adulterated series – 3 replicates for each concentration. (12 concentrations were considered starting from 5% adulteration, up to 60% adulteration.)

Section 3 – Data Analysis Methodology

The following steps were followed for the data analysis procedure.

3.1 Data Filtering

The tidy dataset was taken, and the identified peak regions based on (Rohman & Man, 2009) were extracted.

Next, Pure VCO and Adulterated series were filtered, and the dataset was narrowed down to include only the variables corresponding to the peak regions identified by Rohman and Che Man. Thereby, the data set consisting of over 5000 variables was narrowed down to 38 variables.

3.2 Principal Component Analysis (PCA)

PCA was performed in order to further reduce the dimensions of the dataset (number of variables), and thereby make the analysis simpler. A new set of uncorrelated variables (Principal Components) were created such that they successively maximize the variability that they explain in the given model. If a significant proportion (approximately 80%) of the variability of the model can be explained by the first few PC's, we are able to omit the remaining PCs from the model.

The PC calculation was done and a Scree Plot was plotted in order to identify the number of PCs that need to be included in the model. It was identified that the first 3 PCs collectively explain over 80% of the variability. Therefore, it was decided that it is sufficient to include only the first 3 PCs in the model.

The first three PCs; PC1, PC2 and PC3 were extracted and include in the model.

3.3 Quadratic Discriminant Analysis (QDA)

Then the identified three Principal Components were taken as input variables for the QDA, in (using QDA over Linear Discriminant Analysis (LDA) since the variance of Pure VCO and Adulterated series were not equal).

3.4 Partial Least Squares (PLS) Regression

To identify the Palm Oil concentration of the adulterated Oil group, Partial Least Squares (PLS) Regression was used as in the literature. The concentration level was considered as the dependent variable and the Wavelengths were considered as the independent variables. Three PLS components which minimizes the RMSE and gives a high R squared value were selected for the final model.

Section 4 – Guidelines for using R Software

Data Analysis was done on Rstudio using R programming language. The packages required for the analysis and a basic explanation of the usage of R was explained to the client. Four functions which includes all the steps explained in Section 3 has been constructed. The following general guidelines can be followed to execute the functions.

4.1 Order of running scripts

1. Load.R
2. PC_QD_Analysis1.R
3. PC_QD_Analysis2.R
4. PLS_Analysis1.R
5. PLS_Analysis2.R

4.2 Use the following datasets for PC_QD_Analysis1 and PLS_Analysis1.

1. Training dataset: - “Training Data.csv”
2. Testing dataset: - “Testing Data.csv”

4.3 Use the following dataset for PC_QD_Analysis2 and PLS_Analysis2

1. Training dataset: - “Training Data.csv”
2. Prediction dataset: - “Prediction Data.csv”

4.4 Seven outputs will be produced from the function PC_QD_Analysis1. They are,

1. Box plot
2. Scree plot
3. Summary of PCA results
4. Bi plot
5. Summary of QDA
6. Levene’s test results for 3 PCAs (Y1, Y2, Y3)
7. Confusion matrix (Predictions of the Testing dataset along with the Actuals)

4.5 One outputs will be produced from the function PC_QD_Analysis2. They are,

1. Predicted class (Pure VCO or Adulterated) for new dataset

4.6 Three outputs will be produced from the PLS_Analysis1. They are,

1. Summary of PLS
2. Predicted values for testing dataset
3. Model performance measures

4.7 One output will be produced from the PLS_Function2. They are,

1. Predicted concentration values for new dataset

The confusion matrix, RMSE and R^2 values (model performance measures) obtained from the PC_QD_Analysis1(see Section 4.1) and PLS_Analysis1(see Section 4.1) respectively is used to check the accuracy of the Discriminant Analysis algorithm and PLS regression model. Due to the small number of observations included in the dataset provided by the client, the entire dataset was used as a training dataset. Furthermore, the client notified us that she will create a new testing dataset. Therefore, functions have been written in such a way that model performance measures can be calculated on any Testing dataset that will be entered into the functions by the client.

Refer Appendix to see the codes related to the 4 functions referred to in Section 4.1.

Section 5 – Outputs

5.1 PC_QD_Analysis1 Outputs

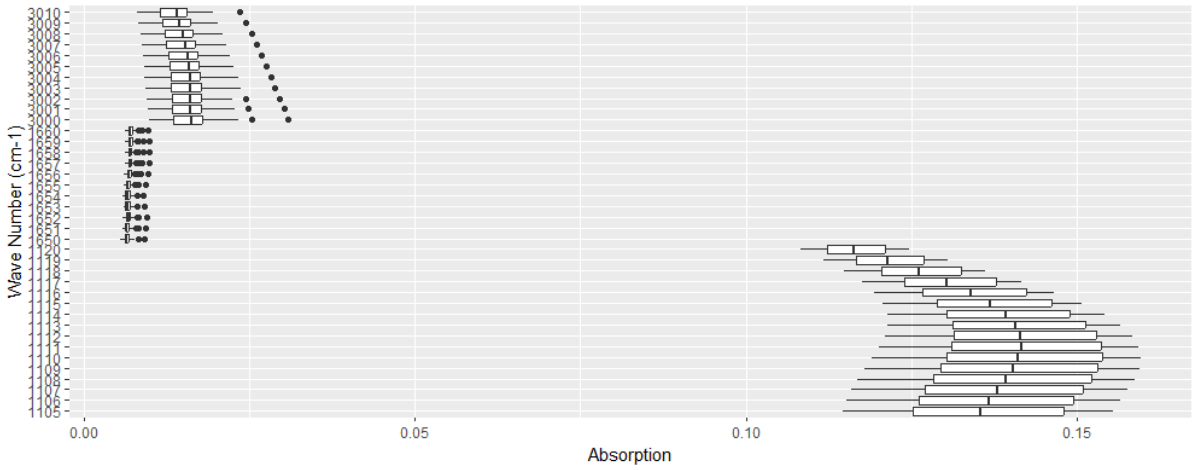


Figure 5. 1: Box Plot

Figure 5.1 shows the box plot of the absorption level of each of the 38 wave numbers (variables) considered for the analysis. The box plot is drawn to obtain a quick understanding of the variability of the absorption level among variables. Since a high variability among variables can be identified, it was decided that PCs have to be extracted from a Correlation Matrix instead of the usual Covariance Matrix. We have to perform standardization on the variables in order to obtain the Correlation Matrix.

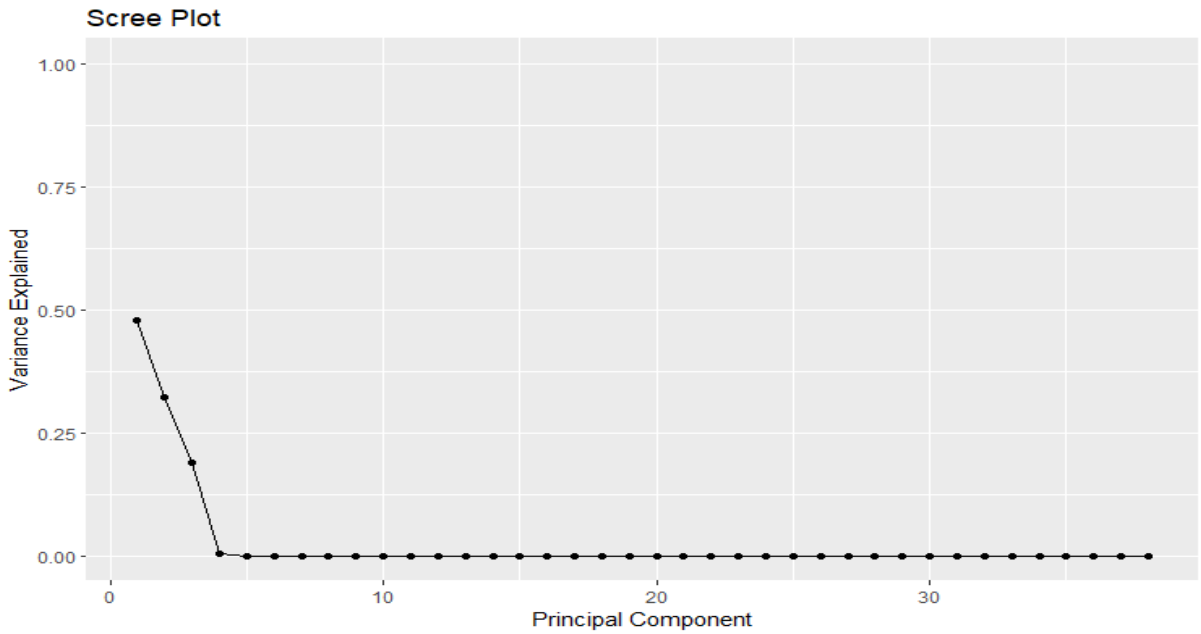


Figure 5. 2: Scree Plot

A Scree Plot is drawn in order to identify the PCs needed for the analysis. The Scree Plot in Figure 5.1 shows that the variance explained by each of the 38 Principal Components, given that the previous Principal Component is already included in the model. The point at which the slope of the curve is clearly leveling off indicates that a negligible proportion of variance is explained by the Principal Components beyond that point. Since the slope levels off after the 3rd PC, it was identified that it is optimal to include only the first 3 PC's in the analysis,

```
$`PCA Summary`
Importance of components:
                          PC1    PC2    PC3     PC4     PC5     PC6     PC7     PC8     PC9    PC10    PC11
Standard deviation     4.2704 3.4977 2.6940 0.46255 0.17636 0.09692 0.09265 0.05354 0.03685 0.03257 0.02637
Proportion of Variance 0.4799 0.3220 0.1910 0.00563 0.00082 0.00025 0.00023 0.00008 0.00004 0.00003 0.00002
Cumulative Proportion  0.4799 0.8019 0.9929 0.99849 0.99931 0.99955 0.99978 0.99985 0.99989 0.99992 0.99994
                          PC12    PC13    PC14    PC15    PC16   PC17     PC18     PC19    PC20     PC21     PC22
Standard deviation     0.02471 0.02283 0.02005 0.01666 0.01362 0.0112 0.008568 0.008073 0.00726 0.004997 0.003966
Proportion of Variance 0.00002 0.00001 0.00001 0.00001 0.00000 0.0000 0.000000 0.000000 0.00000 0.000000 0.000000
Cumulative Proportion  0.99995 0.99997 0.99998 0.99998 0.99999 1.0000 0.999990 1.000000 1.00000 1.000000 1.000000
                          PC23     PC24     PC25     PC26    PC27   PC28     PC29      PC30      PC31      PC32
Standard deviation     0.003524 0.003215 0.002645 0.002262 0.00165 0.0015 0.001192 0.0008695 0.0007335 0.0003955
Proportion of Variance 0.000000 0.000000 0.000000 0.000000 0.00000 0.0000 0.000000 0.0000000 0.0000000 0.0000000
Cumulative Proportion  1.000000 1.000000 1.000000 1.000000 1.00000 1.0000 1.000000 1.0000000 1.0000000 1.0000000
                          PC33       PC34       PC35       PC36       PC37       PC38
Standard deviation     0.000357 0.0002835 0.0002504 0.0001356 0.0001101 6.494e-05
Proportion of Variance 0.000000 0.0000000 0.0000000 0.0000000 0.0000000 0.000e+00
Cumulative Proportion  1.000000 1.0000000 1.0000000 1.0000000 1.0000000 1.000e+00
```

Table 5.1: PCA Summary

The above summary explains that the first 3 principal components jointly explain 99.29% of the total variability. So, it was confirmed that it is sufficient for only the first 3 principal components (PC1, PC2, PC3) to be included in the analysis.

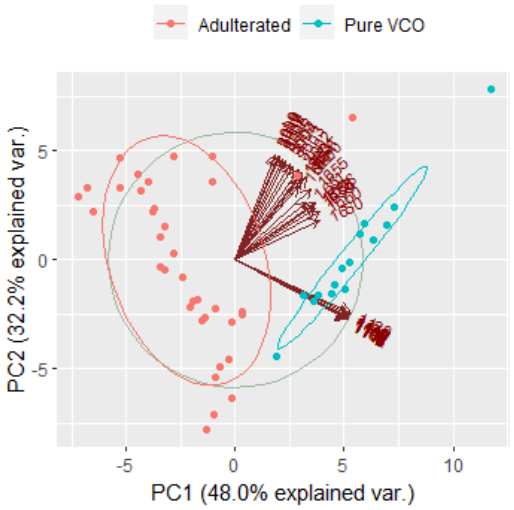


Figure 5. 3: Biplot

The biplot in figure 5.3 is used to illustrate the first two PCs; the scores of each sample of Pure VCO and Adulterated VCO with respect to the first two PCs, and the loading vectors of the 38 original variables with respect to the PCs. It can be identified that the scores relating to the two groups Pure VCO and Adulterated VCO are clearly separated into two groups, showing that they behave differently with respect to the PC's. The red arrows show the direction and strength of association between each variable and the PCs. It can be seen that all variables are in the same direction as PC1 and are therefore positively correlated with PC1, and majority of the variables are also positively correlated with PC2.

Refer this link for information on biplot interpretations.

```
$`Levene test for PCA1`
Levene's Test for Homogeneity of Variance (center = median)
      Df F value Pr(>F)
group  1  0.6419 0.4269
      49

$`Levene test for PCA2`
Levene's Test for Homogeneity of Variance (center = median)
      Df F value Pr(>F)
group  1  5.0217 0.0296 *
      49
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

$`Levene test for PCA3`
Levene's Test for Homogeneity of Variance (center = median)
      Df F value Pr(>F)
group  1  1.6937 0.1992
      49
```

Output 5.1: Results of Levene's Test

Output 5.1 shows the results of Adam Levene's test for PC1, PC2 and PC3. Adam Leven's test checks if the variance of these PCs are homogenous across the all the samples. According to the results, the homogeneity of variance was violated for PC2. Since the homogeneity of variance is violated for at least one PC, we cannot use LDA for the data. Therefore, QDA was applied for the Discriminant Analysis. Furthermore, it should be kept in mind that Adam Leven's test does not require normality assumption for underlying data.

```
$`QDA Results`
Call:
qda(Series ~ ., data = QDA_data)

Prior probabilities of groups:
Adulterated    Pure VCO
  0.7058824   0.2941176

Group means:
                    Y1          Y2         Y3
Adulterated -2.234961 -0.03625099  0.6598691
Pure VCO     5.363907  0.08700237 -1.5836858
```

Output 5. 2: Results of QDA model

Output 5.2 shows the outputs of the QDA analysis. It contains the group means of the two groups; Adulterated and Pure VCO, and not the coefficients of the Linear Discriminants. This is because the QDA classifier involves a quadratic function of the predictors, rather than a linear function.

```
$`QDA Prediction`
   Actual_Group Predicted_Group
1      Pure VCO        Pure VCO
2      Pure VCO        Pure VCO
3      Pure VCO        Pure VCO
4      Pure VCO        Pure VCO
5      Pure VCO        Pure VCO
6      Pure VCO        Pure VCO
7      Pure VCO        Pure VCO
8      Pure VCO        Pure VCO
9      Pure VCO        Pure VCO
10     Pure VCO        Pure VCO
11     Pure VCO        Pure VCO
12     Pure VCO        Pure VCO
13     Pure VCO        Pure VCO
14     Pure VCO        Pure VCO
15     Pure VCO        Pure VCO
16  Adulterated     Adulterated
17  Adulterated     Adulterated
18  Adulterated     Adulterated
19  Adulterated     Adulterated
20  Adulterated     Adulterated
21  Adulterated     Adulterated
```

Output 5. 3: QDA predictions

Output 3.5 shows the prediction of the QDA for the test data set, which are obtained by the ranges identified through the QDA analysis. We have only included part of the output in this report. The predictions of the whole data set are included in the output of the function.

```
$`Confusion Matrix`
             Actual
Predicted     Adulterated Pure VCO
  Adulterated          36        0
  Pure VCO              0       15
```

Output 5.4: Result of the confusion matrix

Output 5.4 shows a Confusion Matrix, which indicates the number of correct an incorrect classification made by the QDA model. The output shows that all 36 adulterated samples have been correctly classified as 'Adulterated' and all Pure VCO samples have also been correctly classified, leaving zero misclassification. This is also understood because the above outputs relate to the training set.

5.2 PC_QD_Analysis2 Outputs

Index	Predicted_Group
1	Adulterated
2	Adulterated
3	Adulterated
4	Pure VCO
5	Pure VCO
6	Pure VCO
7	Adulterated
.	.
.	.
.	.

Output 5.5: Result of Prediction Dataset

Output 5.5 shows the predictions obtained from a new dataset. It must be noted that this new dataset will also go through the variable reduction technique using the three selected PCs, followed by a classification using the QDA classifier that was produced in Section 5.1. Thus, making the new predictions.

5.3 PLS_Analysis1 Output

```
$PLS_Model
Partial Least Squares

66 samples
38 predictors

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 66, 66, 66, 66, 66, 66, ...
Resampling results across tuning parameters:

  ncomp  RMSE         Rsquared   MAE
  1      0.020580509  0.9971173  0.015389619
  2      0.007714185  0.9995856  0.006049408
  3      0.007591998  0.9996073  0.006100827

RMSE was used to select the optimal model using the smallest value.
The final value used for the model was ncomp = 3.
```

```
Data:   X dimension: 66 38
        Y dimension: 66 1
Fit method: oscorespls
Number of components considered: 3
TRAINING: % variance explained
          1 comps  2 comps  3 comps
X           98.06    99.94    99.99
.outcome    99.71    99.96    99.96
```

Output 5,6: PLS Summary

Partial Least Square (PLS) Regression is used to determine the concentration of Palm Oil in a given oil sample, after determining whether the sample is adulterated or not. PLS regression is done independently of the PCA or QDA. The wavelengths are considered as input variables, while the predicted Palm Oil Concentration is the outcome variable. The PLS regression is also a variable reduction technique. However, the PLS regression requires that the selected PCs explain a significant proportion (at least 80%) of the predictor variables (X) as well as the outcome variable.

Output 5.6 shows the summary of the PLS regression performed on our training dataset. It shows that three PCs have been identified , and that they collectively explain more than 99% of the variability of both the outcome and the predictor variables.

Series	Predicted.Concentration
Pure Palm Oil	99.89098726
Pure Palm Oil	99.98582626
Pure Palm Oil	99.92635361
.	.
.	.
.	.
Pure VCO	1.02240179
Pure VCO	0.36026133
Pure VCO	0.66455334
.	.
.	.
.	.
Adulterated	4.52843673
Adulterated	2.93100368
Adulterated	4.12373873
.	.
.	.
.	.

Output 5.7: Predicted Values for Test Dataset

Output 5.7 shows the predicted Palm Oil concentrations in the oil samples in the testing dataset, which have been determined by the PLS regression. It can be identified that all the Pure Palm Oil samples have given a very high Palm oil concentration value, that Pure VCO given a negligible Palm Oil concentration value and that the adulterated samples have taken intermediate values. The complete output can be used to verify that the predicted Palm Oil concentrations of this dataset have been very close to the actual concentrations.

```
$`Model Performance`
        RMSE   Rsquare
1 0.006950048 0.9996487
```

Output 5.8: Model Performance Measures

Output 5.8 shows the cross-validation error RMSE obtained with the PLS model is 0.00695, and the R squared value is 0.9996. This shows that the model explained more than 99% of the variability of the observed concentration variable, with minimal error.

5.4 PLS_Analysis2 Output

Series	Predicted.Concentration
Pure Palm Oil	104.0213903
Pure Palm Oil	104.1586968
Pure Palm Oil	104.3210992
.	.
.	.
.	.
Pure VCO	-0.01390208
Pure VCO	-0.10875619
Pure VCO	-0.87427274
.	.
.	.
.	.
Adulterated	4.57906425
Adulterated	2.78222499
Adulterated	4.21542112
.	.
.	.
.	.

Output 5.9: PLS Predictions for New Dataset

Output 5.9 presents the predictions that will be produced from a new dataset. The predicted Palm oil concentration in Pure Palm Oil has slightly surpassed 100%, while that in pure VCO contains negative values, indicating that there may be slight inefficiencies in the fitted model. One main reason for this could be the small sample size used for the analysis. This analysis can be re-done using a larger sample to test if these inefficiencies are eliminated. In the case of this dataset, all negative values should be replaced by zero.

References

Rohman, A., & Man, Y. B. C. (2009). Monitoring ofvirgin coconut oil (VCO) adulteration with palm oil using Fourier transform infrared spectroscopy. *Journal of Food Lipids*, *16*(4), 618–628. https://doi.org/10.1111/j.1745-4522.2009.01170.x

Appendix

This section includes the codes related to the 4 functions produced for the client.

Loading Packages

```
library(tidyverse)
library(ggplot2)
library(GGally)
library(plotly)
library(MASS)
library(car)
library(pls)
library(caret)
```

PC_QD_Analysis1

```
# Importing Data setets
Training<-read_csv("Training Data.csv") # This is for training set
Testing<-read_csv("Testing Data.csv") # This is for testing set

#=============================================================================
# Function for Data analysis
PC_QD_Analysis1<-function(Training, Testing){

  ## Training Data prep
  Training <- rename(Training, Concentration = `Palm olein concentration(C)`,
  Replicate = `Replicate No`, W=`Wave Number (cm-1)(W)`, A=`Absorption (A)`)

  # Filtering Wavelenghts
  filtertraindata2<-Training %>% filter(W>=3000 & W<=3010)
  filtertraindata3<-Training %>% filter(W>=1650 & W<=1660)
  filtertraindata4<-Training %>% filter(W>=1105 & W<=1120)

  # Combining filtered datasets
  filtertraindata5<-rbind(filtertraindata2, filtertraindata3, filtertraindata4)

  # Selecting VCO and adulterated
  VCOtraindata<-filtertraindata5 %>% filter(Series=="Pure VCO")
  Adulttraindata<-filtertraindata5 %>% filter(Series=="Adulterated")

  # Combining datasets
```

```r
PCAtraindata<-rbind(VCOtraindata, Adulttraindata)

# Putting PCA data in wider format
PCAtraindata<-pivot_wider(PCAtraindata, names_from = W, values_from = A)
PCAtraindata<-PCAtraindata %>% mutate(Index=1:n()) %>%
  relocate(Index, .before = Series)

pcatraindata <-  PCAtraindata
pcatraindata_v2 <- pcatraindata %>%
  subset(select = -c(Index, Series, Concentration, Replicate))

#############################################################################
# Testing data prep
Testing <- rename(Testing, Concentration = `Palm olein concentration(C)`,
Replicate = `Replicate No`, W=`Wave Number (cm-1)(W)`, A=`Absorption (A)`)

# Filtering Wavelenghts
filtertestdata2<-Testing %>% filter(W>=3000 & W<=3010)
filtertestdata3<-Testing %>% filter(W>=1650 & W<=1660)
filtertestdata4<-Testing %>% filter(W>=1105 & W<=1120)

# Combining filtered datasets
filtertestdata5<-rbind(filtertestdata2, filtertestdata3, filtertestdata4)

# Selecting VCO and adulterated
VCOtestdata<-filtertestdata5 %>% filter(Series=="Pure VCO")
Adulttestdata<-filtertestdata5 %>% filter(Series=="Adulterated")

# Combining datasets
PCAtestdata<-rbind(VCOtestdata, Adulttestdata)

# Putting PCA data in wider format
PCAtestdata<-pivot_wider(PCAtestdata, names_from = W, values_from = A)
PCAtestdata<-PCAtestdata %>% mutate(Index=1:n()) %>%
  relocate(Index, .before = Series)

pcatestdata <-  PCAtestdata
pcatestdata_v2 <- pcatestdata %>%
  subset(select = -c(Index, Series, Concentration, Replicate))

#############################################################################

## PCA Analysis

# Box plot

# data set for box plot
data_boxplot <- pcatraindata %>%
  pivot_longer(cols = 5:ncol(pcatraindata), names_to = "W", values_to = "A")

# Box plot
boxplot <- data_boxplot %>% ggplot(aes(x = A, y = W)) + geom_boxplot() +
  xlab("Absorption") + ylab("Wave Number (cm-1)")
```

```r
# PC calculation
pc <- prcomp(pcatraindata_v2,
             center = TRUE,
             scale. = TRUE)

# Scree Plot
#calculate total variance explained by each principal component
var_explained <- pc$sdev^2 / sum(pc$sdev^2)

#create scree plot
p<-qplot(c(1:38), var_explained) +
  geom_line() +
  xlab("Principal Component") +
  ylab("Variance Explained") +
  ggtitle("Scree Plot") +
  ylim(0, 1)

# Summary of PCA
Summary_PCA<-summary(pc)

# extracting PCA scores
Y1 <- pc$x[ , 1]
Y2 <- pc$x[ , 2]
Y3 <- pc$x[ , 3]


# PC data
PC_Scores <- cbind(pcatraindata[,1:4], Y1, Y2, Y3) %>% as.data.frame()


##############################################################################

## DA Analysis

# load the data
DA_data <- PC_Scores

# Group into Pure VCO and Adulterated
Pure_VCO <- DA_data %>% filter(Series == "Pure VCO")
Adulterated <- DA_data %>% filter(Series == "Adulterated")

# Checking the Assumption of Equal Covariance
# Levene's test
levene_data <- rbind(Pure_VCO, Adulterated)
levene_result_Y1 = leveneTest(Y1 ~ Series, levene_data)
levene_result_Y2 = leveneTest(Y2 ~ Series, levene_data)
levene_result_Y3 = leveneTest(Y3 ~ Series, levene_data)

# QDA
QDA_data = subset(DA_data, select = -c(Index, Concentration, Replicate))

qda_results <- qda(Series~., QDA_data)
```

```r
  ##############################################################################

  ## Prediction from testing set

  # Predicting PCs
  test_pcs <- predict(pc, newdata = pcatestdata_v2)
  test_pcs3 <- data.frame(Y1 = test_pcs[,1],
                          Y2 = test_pcs[,2],
                          Y3 = test_pcs[,3])
  test_pcs3 <- as.data.frame(cbind(Series=pcatestdata$Series, test_pcs3))

  # Confusion Matrix
  pred <- predict(qda_results, test_pcs3)$class

  confusion_matrix <- table(Predicted = pred, Actual = test_pcs3$Series)

  # Prediction Data Frame
  qda_prediction_df <- data.frame("Actual_Group" = test_pcs3$Series,
                                  "Predicted_Group" = pred)


  ##############################################################################
  ## Outputs
  list(`Box Plot`=  boxplot, `Scree Plot`=p, `PCA Summary`=Summary_PCA,
       `Bi Plot`=biplot, `Levene test for PCA1` =levene_result_Y1,
       `Levene test for PCA2` =levene_result_Y2,
       `Levene test for PCA3` =levene_result_Y3,
       `QDA Results`= qda_results, `QDA Prediction`= qda_prediction_df,
       `Confusion Matrix`= confusion_matrix)
}

#=============================================================================

PC_QD_Analysis1(Training, Testing)
```

PC_QD_Analysis2

```
# Importing Data sets
Training<-read_csv("Training Data.csv") # This is for training set
Prediction<-read_csv("Prediction Data.csv") # This is for prediction set

#=============================================================================

# Function for Data analysis
PC_QD_Analysis2<-function(Training, Prediction){

  ## Training Data prep
  Training <- rename(Training, Concentration = `Palm olein concentration(C)`,
   Replicate = `Replicate No`, W=`Wave Number (cm-1)(W)`, A=`Absorption (A)`)

  # Filtering Wavelenghts
  filtertraindata2<-Training %>% filter(W>=3000 & W<=3010)
  filtertraindata3<-Training %>% filter(W>=1650 & W<=1660)
  filtertraindata4<-Training %>% filter(W>=1105 & W<=1120)

  # Combining filtered datasets
  filtertraindata5<-rbind(filtertraindata2, filtertraindata3, filtertraindata4)

  # Selecting VCO and adulterated
  VCOtraindata<-filtertraindata5 %>% filter(Series=="Pure VCO")
  Adulttraindata<-filtertraindata5 %>% filter(Series=="Adulterated")

  # Combining datasets
  PCAtraindata<-rbind(VCOtraindata, Adulttraindata)

  # Putting PCA data in wider format
  PCAtraindata<-pivot_wider(PCAtraindata, names_from = W, values_from = A)
  PCAtraindata<-PCAtraindata %>% mutate(Index=1:n()) %>%
    relocate(Index, .before = Series)

  pcatraindata <-  PCAtraindata
  pcatraindata_v2 <- pcatraindata %>%
    subset(select = -c(Index, Series, Concentration, Replicate))

 ##############################################################################

  # Validation data prep
 Prediction <- rename(Prediction, Concentration = `Palm olein concentration(C)`,
    Replicate = `Replicate No`, W=`Wave Number (cm-1)(W)`, A=`Absorption (A)`)

  # Filtering Wavelengths
  filterpredictiondata2<-Prediction %>% filter(W>=3000 & W<=3010)
  filterpredictiondata3<-Prediction %>% filter(W>=1650 & W<=1660)
  filterpredictiondata4<-Prediction %>% filter(W>=1105 & W<=1120)

  # Combining filtered datasets
  PCApredictiondata<-rbind( filterpredictiondata2, filterpredictiondata3,
                            filterpredictiondata4)
```

```r
# Putting PCA data in wider format
PCApredictiondata<-pivot_wider( PCApredictiondata, names_from = W,
                                values_from = A)
PCApredictiondata<- PCApredictiondata %>% mutate(Index=1:n()) %>%
  relocate(Index, .before = Series)

pcapredictiondata <-  PCApredictiondata
pcapredictiondata_v2 <-  pcapredictiondata %>%
  subset(select = -c(Index, Series, Concentration, Replicate))

nrow(pcapredictiondata_v2)
##############################################################################

## PCA Analysis

# PC calculation
pc <- prcomp(pcatraindata_v2,
             center = TRUE,
             scale. = TRUE)

# extracting PCA scores
Y1 <- pc$x[ , 1]
Y2 <- pc$x[ , 2]
Y3 <- pc$x[ , 3]

# PC data
PC_Scores <- cbind(pcatraindata[,1:4], Y1, Y2, Y3) %>% as.data.frame()


##############################################################################

## DA Analysis

# load the data
DA_data <- PC_Scores

# QDA
QDA_data = subset(DA_data, select = -c(Index, Concentration, Replicate))
qda_results <- qda(Series~., QDA_data)

##############################################################################

## Prediction from prediction set

# Predicting PCs
prediction_pcs <- predict(pc, newdata = pcapredictiondata_v2)
prediction_pcs3 <- data.frame(Y1 = prediction_pcs[,1],
                        Y2 = prediction_pcs[,2],
                        Y3 = prediction_pcs[,3])


# Prediction
pred <- predict(qda_results, prediction_pcs3)$class
```

```
  # Prediction Data Frame
  qda_prediction_df <- data.frame("Predicted_Group" = pred)


  ##############################################################################

  ## Outputs
  list( `QDA Prediction`= qda_prediction_df)
}

#==============================================================================

# Inputs for function

PC_QD_Analysis2(Training, Prediction)
```

PLS_Analysis 1

```r
# Importing Data setets
Training<-read_csv("Training Data.csv") # This is for training set
Testing<-read_csv("Testing Data.csv") # This is for testing set

#=============================================================================

# Function
PLS_Analysis1<-function(Training, Testing){
  ## Training Data
  Training <- rename(Training, Concentration = `Palm olein concentration(C)`,
    Replicate = `Replicate No`, W=`Wave Number (cm-1)(W)`, A=`Absorption (A)`)

  # Filtering Wavelenghts
  filtertraindata2<-Training %>% filter(W>=3000 & W<=3010)
  filtertraindata3<-Training %>% filter(W>=1650 & W<=1660)
  filtertraindata4<-Training %>% filter(W>=1105 & W<=1120)

  # Combining filtered datasets
  filtertraindata5<-rbind(filtertraindata2, filtertraindata3, filtertraindata4)

  # Putting PCA data in wider format
  PLStraindata<-pivot_wider(filtertraindata5, names_from = W, values_from = A)
  PLStraindata <-PLStraindata %>% mutate(Index=1:n()) %>%
    relocate(Index, .before = Series)

  # Converting concentration to numeric vector in training data
  PLStraindata <- PLStraindata %>% separate(Concentration,
                                            into = c("Concentration", "percent"))
  PLStraindata$Concentration <- as.numeric(PLStraindata$Concentration)/100

  PLStraindata <- PLStraindata %>% subset(select = -c(Index, Series,
                                                      percent, Replicate))


  ##############################################################################

  ## Testing Data prep
  Testing <- rename(Testing, Concentration = `Palm olein concentration(C)`,
    Replicate = `Replicate No`, W=`Wave Number (cm-1)(W)`, A=`Absorption (A)`)

  # Filtering Wavelenghts
  filtertestdata2<- Testing %>% filter(W>=3000 & W<=3010)
  filtertestdata3<- Testing %>% filter(W>=1650 & W<=1660)
  filtertestdata4<- Testing %>% filter(W>=1105 & W<=1120)

  # Combining filtered datasets
  filtertestdata5<-rbind(filtertestdata2, filtertestdata3, filtertestdata4)

  # Putting PCA data in wider format
  PLStestdata<-pivot_wider(filtertestdata5, names_from = W, values_from = A)
  PLStestdata <-PLStestdata %>% mutate(Index=1:n()) %>%
```

```
    relocate(Index, .before = Series)

  # Coverting concentration to numeric vector
  PLStestdata <- PLStestdata %>% separate(Concentration,
                                          into = c("Concentration", "percent"))
  PLStestdata$Concentration <- as.numeric(PLStestdata$Concentration)/100

  PLStestdata <- PLStestdata %>% subset(select = -c(Index, percent, Replicate))
  ncolplstest<-ncol(PLStestdata)

  ##############################################################################

  # Fitting model for training data set

  set.seed(123)
  model <- train(
    Concentration ~ .,
    data = PLStraindata,
    method = 'pls'
  )

  # Summarize the final model
  summary <- summary(model$finalModel)

  ##############################################################################

  ## Prediction for testing dataset

  predictions = predict(model, newdata = PLStestdata[,3: ncolplstest])
  predicted_PLS <- predictions*100

  predictionTable <- data.frame(Series = PLStestdata$Series,
                                `Predicted Concentration (%)` = predicted_PLS)


  # Model performance metrics
  peformance_values <- data.frame(
    RMSE = caret::RMSE(predictions, PLStestdata$Concentration),
    Rsquare = caret::R2(predictions, PLStestdata$Concentration)
  )

  ##############################################################################

  ## Outputs

  list(PLS_Model = model, `Predicted values for testing set` = predictionTable,
       `Model Performance` = peformance_values)

}

#=============================================================================

# Inputs for function
```

```
PLS_Analysis1(Training, Testing)
```

PLS__Analysis2

```
# Importing Data setets
Training<-read_csv("Training Data.csv") # This is for training set
Validation <- read_csv("Prediction Data.csv") # This is for Prediction data set

#=============================================================================

# Function
PLS_Analysis2 <-function(Training, Validation){
  ## Training Data
  Training <- rename(Training, Concentration = `Palm olein concentration(C)`,
   Replicate = `Replicate No`, W=`Wave Number (cm-1)(W)`, A=`Absorption (A)`)

  # Filtering Wavelenghts
  filtertraindata2<-Training %>% filter(W>=3000 & W<=3010)
  filtertraindata3<-Training %>% filter(W>=1650 & W<=1660)
  filtertraindata4<-Training %>% filter(W>=1105 & W<=1120)

  # Combining filtered datasets
  filtertraindata5<-rbind(filtertraindata2, filtertraindata3, filtertraindata4)

  # Putting PCA data in wider format
  PLStraindata<-pivot_wider(filtertraindata5, names_from = W, values_from = A)
  PLStraindata <-PLStraindata %>% mutate(Index=1:n()) %>%
    relocate(Index, .before = Series)

  # Converting concentration to numeric vector in training data
  PLStraindata <- PLStraindata %>% separate(Concentration,
                                            into = c("Concentration", "percent"))
  PLStraindata$Concentration <- as.numeric(PLStraindata$Concentration)/100

  PLStraindata <- PLStraindata %>% subset(select = -c(Index, Series,
                                                      percent, Replicate))


  ###########################################################################

  # Fitting model for training data set
  set.seed(123)
  model <- train(
    Concentration ~ .,
    data = PLStraindata,
    method = 'pls'
  )


  ###########################################################################
```

```
  # Prediction for validation data set (Validation --> Prediction Data)

  ## Validation Data
  Validation <- rename(Validation, Concentration = `Palm olein concentration(C)`,
   Replicate = `Replicate No`, W=`Wave Number (cm-1)(W)`, A=`Absorption (A)`)

  # Filtering Wavelenghts
  filterValidationdata2<-Validation %>% filter(W>=3000 & W<=3010)
  filterValidationdata3<-Validation %>% filter(W>=1650 & W<=1660)
  filterValidationdata4<-Validation %>% filter(W>=1105 & W<=1120)

  # Combining filtered datasets
  filterValidationdata5<-rbind(filterValidationdata2, filterValidationdata3,
                               filterValidationdata4)

  # Putting PCA data in wider format
  PLSValidationdata<-pivot_wider(filterValidationdata5, names_from = W, values_from = A)
  PLSValidationdata <-PLSValidationdata %>% mutate(Index=1:n()) %>%
    relocate(Index, .before = Series)

  # Converting concentration to numeric vector in training data
  PLSValidationdata <- PLSValidationdata %>% separate(Concentration,
                                        into = c("Concentration", "percent"))
  PLSValidationdata$Concentration <- as.numeric(PLSValidationdata$Concentration)/100

  PLSValidationdata <- PLSValidationdata %>% subset(select = -c(Index,
                                                    percent, Replicate))
  ncolplsvalidation<-ncol(PLSValidationdata)

  ##############################################################################

  ## Prediction for validation dataset

  predictions = predict(model, newdata = PLSValidationdata[,2:ncolplsvalidation])
  predicted_PCR <- predictions*100

  predictionTable <- data.frame(`Series Label` = PLSValidationdata$Series,
                                `Predicted Concentration (%)` = predicted_PCR)

  ##############################################################################
  ## Outputs

  list(`Predicted values for validation set` = predictionTable)
}

#=============================================================================

# Inputs for function

PLS_Analysis2(Training, Validation)
```