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)
```