Functional Data Analysis
Project 1
Shuenn Siang Ng

Introduction

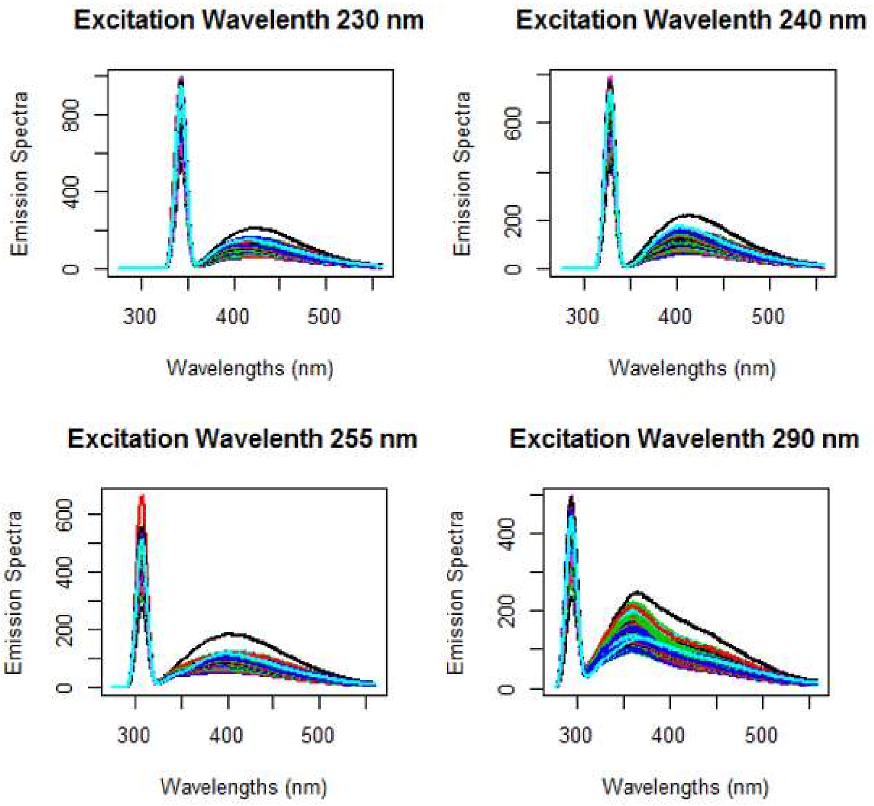
There are two parts in this project. Both are analysis of real data sets. The first part is an analysis of the sugar process data along with two response variables, which was generated by Bro in 1999. The sugar was dissolved in un-buffered water (2.25g/15mL) and the solution was measured spectrofluorometrically in a 10 by 10 mm cuvette on a PE LS50B spectrofluorometer. Raw non-smoothed data was output from the spectrofluorometer. For every sample the emission spectra from 275-560 nm were measured in 0.5 nm intervals (571 wavelengths) at seven excitation wavelengths (230, 240, 255, 290, 305, 325, 340 nm). The smoothed FPCA is used to build a regression model to study the association between the fluorescence data and the response variables.

For the second part we analyze the pharmaceutical tablets data generated by Dyrby et al. in 2002. The spectral curves of Near Infrared Transmittance for 310 tablets were measured. These 310 tablets belong to four classes. Linear Discriminant Analysis is used to build a classification rule which assigns a tablet to one of four classes based on its spectral curve.

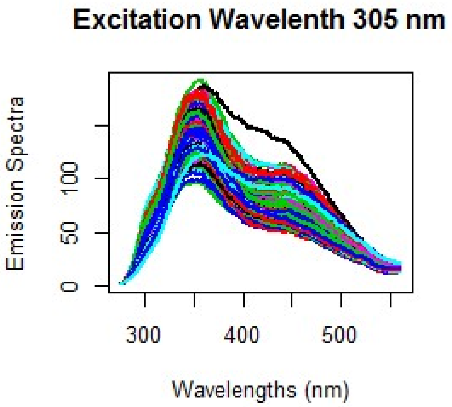
First Part: Sugar Process Data

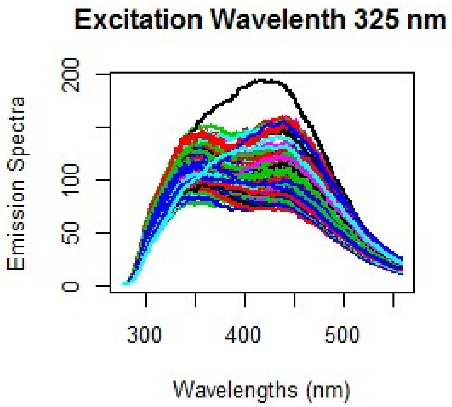
The first data set is a 197 by 3997 matrix, where n = 197 is the sample size and 3997 is the number of observation points for the seven curves. That is, the first 571 number in each row is the observed values of the sample curves for the excitation wavelength 230, the next 571 numbers for the excitation wavelength 240, and so on. The aforementioned response variables are “color” and “ash measurements”.

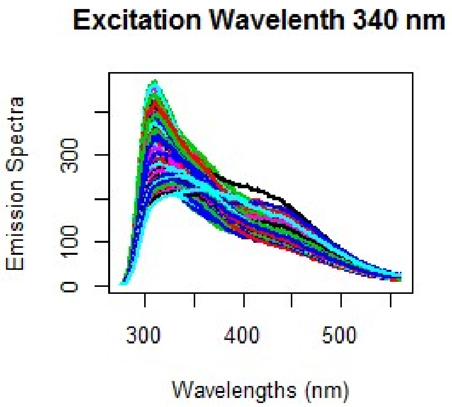
We start by plotting the data for each excitation wavelength:



Functional Data Analysis
Project 1
Shuenn Siang Ng







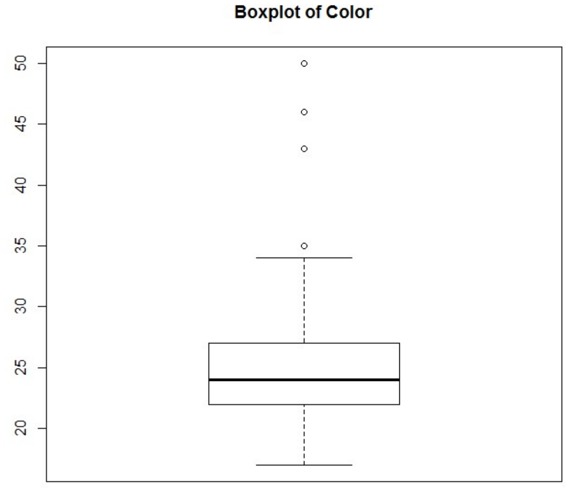
All the graphs have some similarities. Overall, they all start increasing to a peak and eventually decrease to 0 at the end. In particular, first four graphs are really similar to each other. They all have a high peak at the beginning followed by a small round bump. The last three graphs do not have the small round bump like the first four.

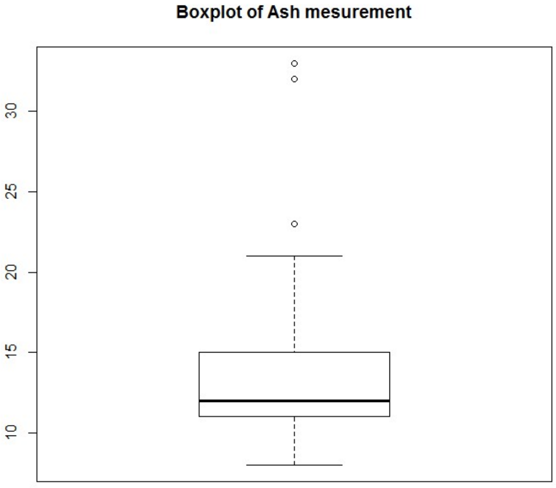
Furthermore, we make a summary and boxplots of the two response variables:

Table 1.1: Summary of response variables

Statistics	Color	Ash
Minimum	17	8
First Quartile	22	11
Median	24	12
Mean	24.5	13.2
Third Quartile	27	15
Maximum	50	33

Functional Data Analysis
Project 1
Shuenn Siang Ng





From the boxplots, there are a few outliers for both response variables. It seems that the response variable, color is more evenly distributed compared to ash measurement.

We use the smooth FPCA to calculate the PC scores for each seven functional data as predictor variables to build a regression model for the both response variables. We decided to first use 1000 basis functions to smooth the data. Then, we choose smoothing parameters and number of PC components by cross-validation. The following tables summarize the cross-validation errors.

Table 1.2: Cross-Validation Errors for Response Variable "Color"

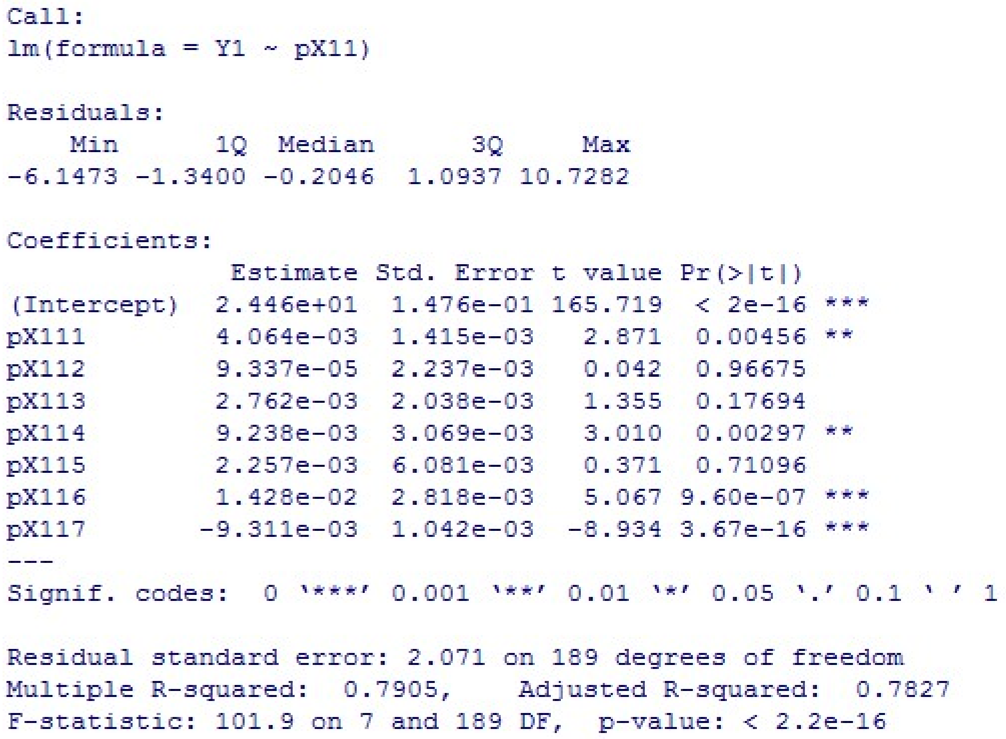
λ \ M	1	2	3	4	5	6	7	8	9	10
10^{-10}	1427	1479	1715	1513	1824	2408	2143	2632	2101	2470
10^{-8}	1427	1479	1715	1513	1824	2408	2143	2632	2101	2470
10^{-6}	1427	1479	1715	1513	1824	2408	2142	2632	2101	2470
10^{-4}	1427	1479	1715	1513	1824	2407	2141	2632	2101	2468
10^{-2}	1427	1479	1717	1516	1902	2375	1981	2324	2175	2353
1	1427	1479	1721	1522	1841	2234	1630	1897	2003	2344
10	1427	1492	1739	1521	1835	2058	1567	1570	1962	2005

Table 1.3: Cross Validation Errors for Response Variable “Ash Measurement”

λ \ M	1	2	3	4	5	6	7	8	9	10
10^{-10}	1277	705	416	455	443	474	528	612	730	781
10^{-8}	1277	705	416	455	443	474	528	612	730	781
10^{-6}	1277	705	416	455	443	474	528	612	730	781
10^{-4}	1277	705	416	455	443	474	528	612	728	781
10^{-2}	1278	705	416	455	472	492	501	568	636	730
1	1278	705	416	456	477	474	511	564	617	578
10	1278	711	427	456	477	466	570	638	560	756

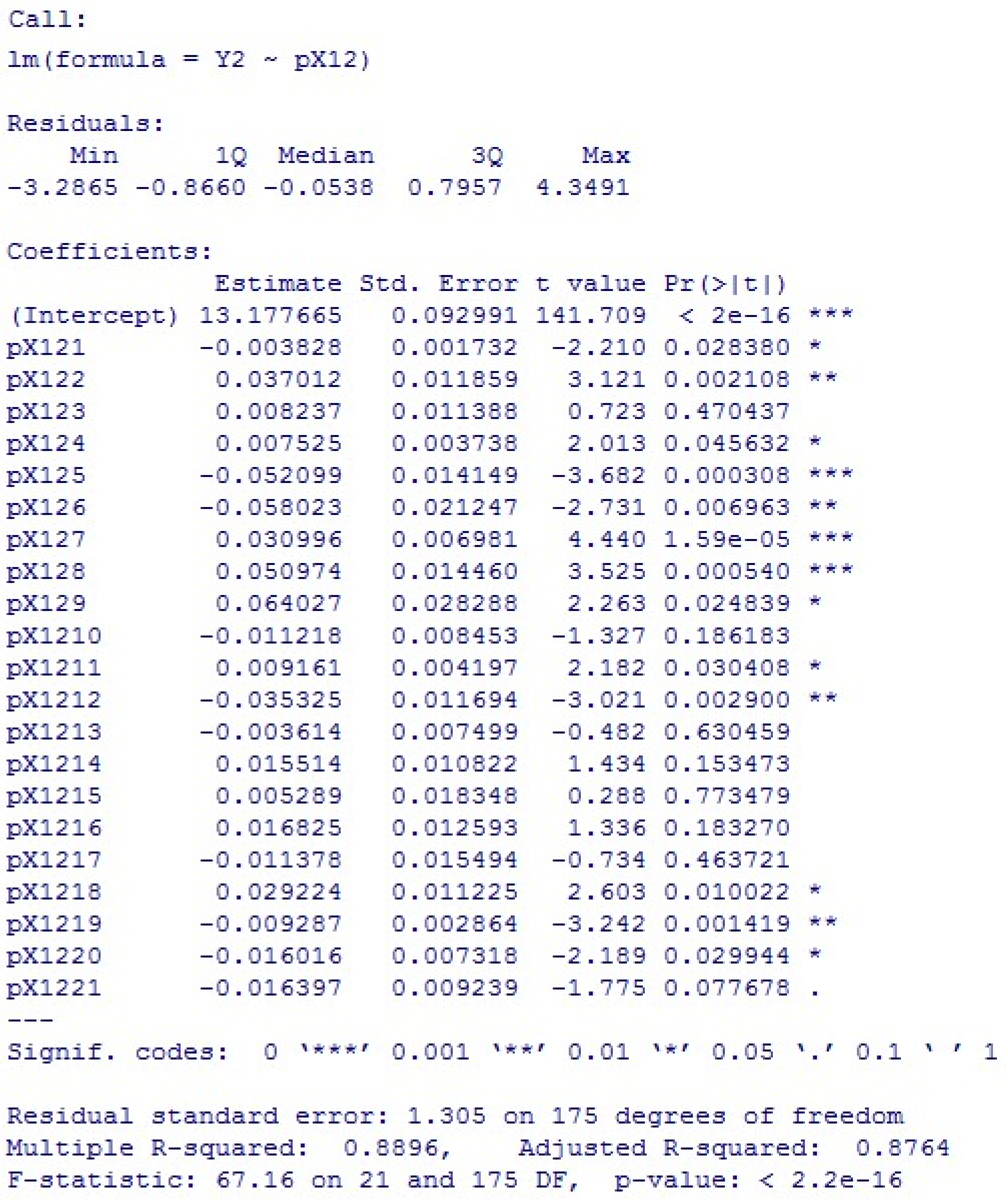
We found that the optimal tuning parameter and number of PC components for response variable “color” are 10^{-2} and 1, respectively. Also, optimal tuning parameter and number of PC components for response variable “ash measurement” is 10^{-10} and 3, respectively. Using these optimal values, we calculate the PC scores for each of seven functional data and use these scores as predictor variables to build a regression model for both response variables.

Diagram 1.2: Summary of Regression with Response Variable “Color”



```
Call:
lm(formula = Y1 ~ pX11)

Residuals:
    Min      1Q  Median      3Q     Max 
-6.1473 -1.3400 -0.2046  1.0937 10.7282 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  2.446e+01  1.476e-01 165.719  < 2e-16 ***
pX111        4.064e-03  1.415e-03   2.871  0.00456 ** 
pX112        9.337e-05  2.237e-03   0.042  0.96675    
pX113        2.762e-03  2.038e-03   1.355  0.17694    
pX114        9.238e-03  3.069e-03   3.010  0.00297 ** 
pX115        2.257e-03  6.081e-03   0.371  0.71096    
pX116        1.428e-02  2.818e-03   5.067 9.60e-07 ***
pX117       -9.311e-03  1.042e-03  -8.934 3.67e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.071 on 189 degrees of freedom
Multiple R-squared:  0.7905,    Adjusted R-squared:  0.7827 
F-statistic: 101.9 on 7 and 189 DF,  p-value: < 2.2e-16
```

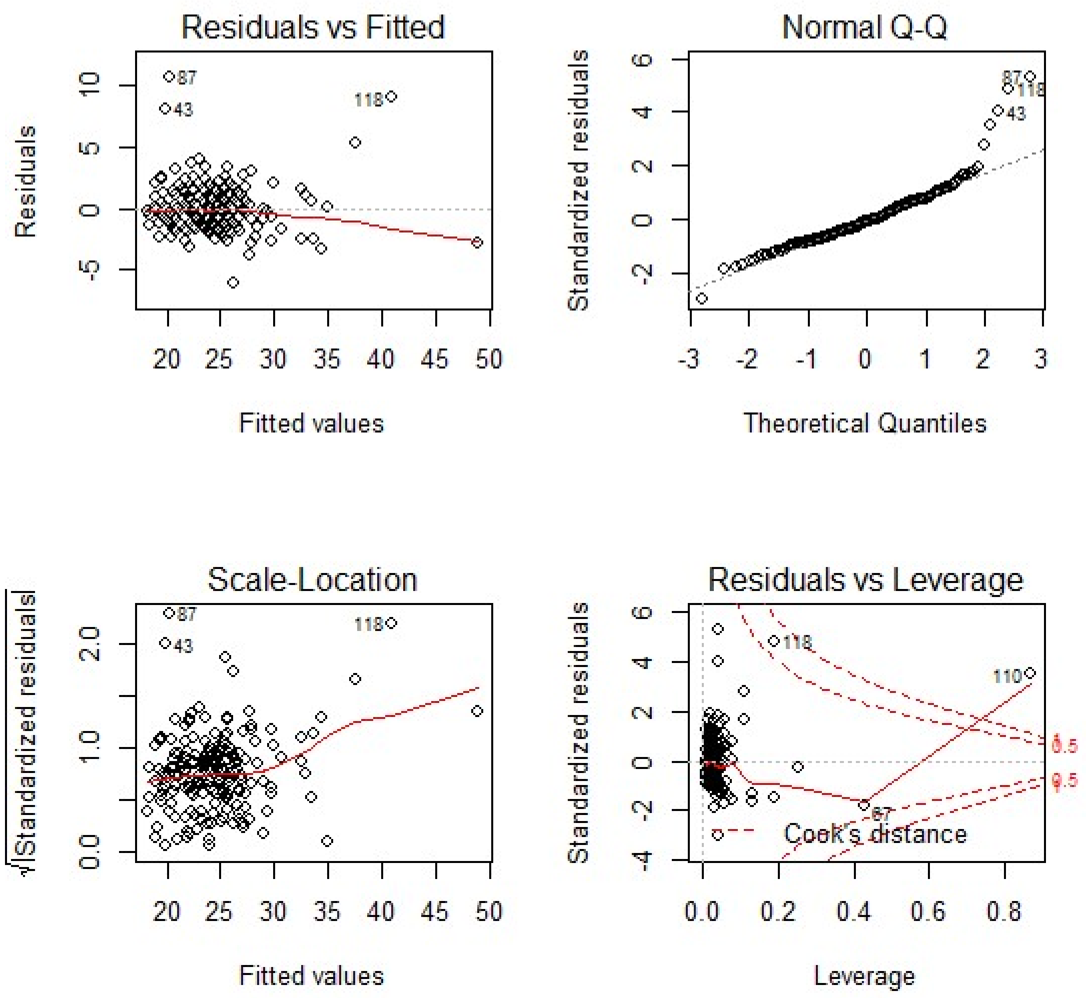
Diagram 1.2: Summary of Regression with Response Variable "Ash Measurement"



```
Call:
lm(formula = Y2 ~ pX12)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.2865 -0.8660 -0.0538  0.7957  4.3491 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 13.177665   0.092991 141.709  < 2e-16 ***
pX121       -0.003828   0.001732  -2.210 0.028380 *  
pX122        0.037012   0.011859   3.121 0.002108 ** 
pX123        0.008237   0.011388   0.723 0.470437    
pX124        0.007525   0.003738   2.013 0.045632 *  
pX125       -0.052099   0.014149  -3.682 0.000308 ***
pX126       -0.058023   0.021247  -2.731 0.006963 ** 
pX127        0.030996   0.006981   4.440 1.59e-05 ***
pX128        0.050974   0.014460   3.525 0.000540 ***
pX129        0.064027   0.028288   2.263 0.024839 *  
pX1210      -0.011218   0.008453  -1.327 0.186183    
pX1211       0.009161   0.004197   2.182 0.030408 *  
pX1212      -0.035325   0.011694  -3.021 0.002900 ** 
pX1213      -0.003614   0.007499  -0.482 0.630459    
pX1214       0.015514   0.010822   1.434 0.153473    
pX1215       0.005289   0.018348   0.288 0.773479    
pX1216       0.016825   0.012593   1.336 0.183270    
pX1217      -0.011378   0.015494  -0.734 0.463721    
pX1218       0.029224   0.011225   2.603 0.010022 *  
pX1219      -0.009287   0.002864  -3.242 0.001419 ** 
pX1220      -0.016016   0.007318  -2.189 0.029944 *  
pX1221      -0.016397   0.009239  -1.775 0.077678 .  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.305 on 175 degrees of freedom
Multiple R-squared:  0.8896,    Adjusted R-squared:  0.8764 
F-statistic: 67.16 on 21 and 175 DF,  p-value: < 2.2e-16
```

We plotted both regressions. The plots are below.

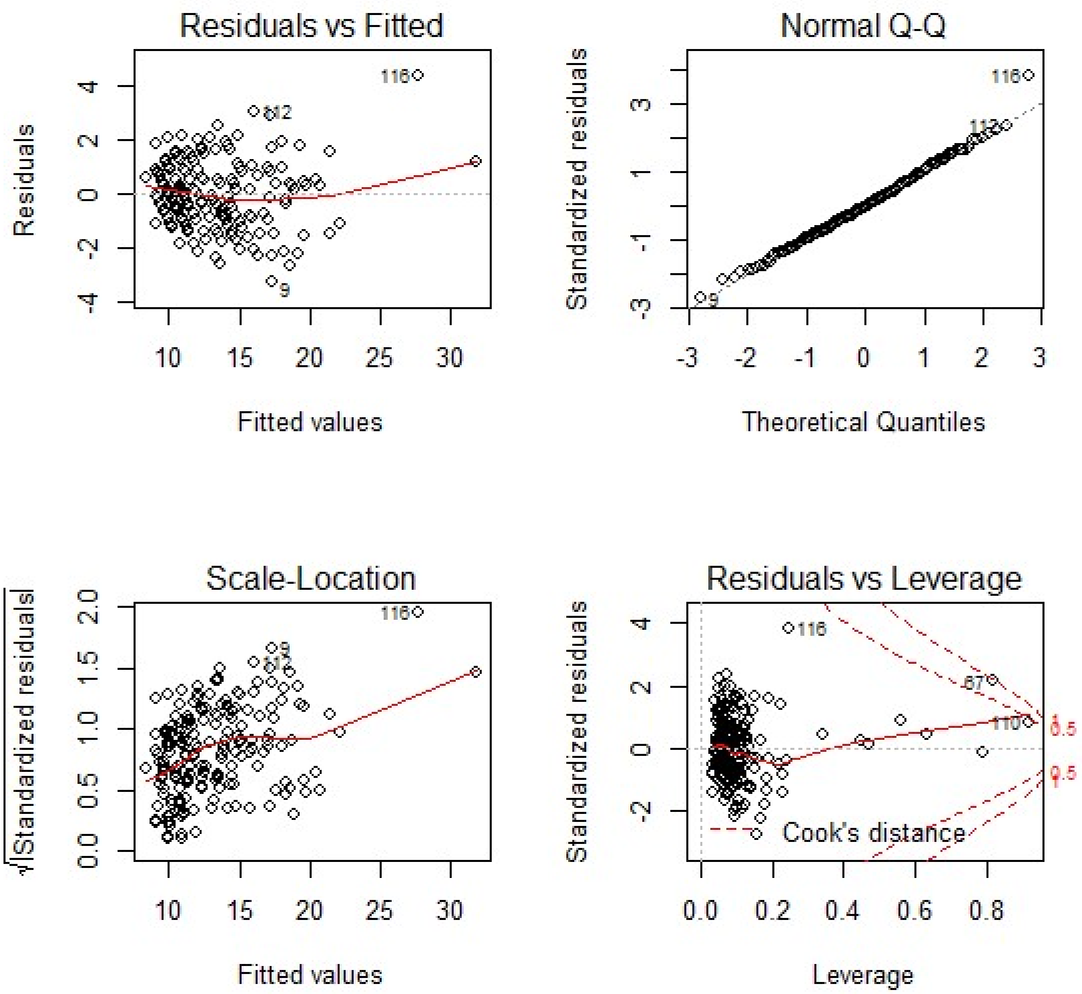
Diagram 1.3: Regression with Response Variable “Color”



Based on the Residue vs Fitted plot above, there does not seem to be a non-linear relationship between predictor variables and the response variable. Also, Q-Q plot suggests that the residuals are not normally distributed because of the heavy tails. Furthermore, the Scale-Location plot does not support the assumption of equal variance. Overall, the regression model is a pretty good fit.

Diagram 1.1 shows that those significant coefficients come from first, fourth, sixth, and seventh curves. In other words, the emission spectra are affected mostly by excitation wavelengths of 230 nm, 290 nm, 325 nm, and 340 nm.

Diagram 1.4: Regression with Response Variable "Ash Measurement"



Based on the Residue vs Fitted plot above, there seem to be a non-linear relationship between predictor variables and the response variable, which might be a quadratic relationship. However, the line is extremely close to a straight line. Also, Q-Q plot suggests that the residuals are normally distributed. Furthermore, the Scale-Location plot does not support the assumption of equal variance. Overall, the regression model is a pretty good fit.

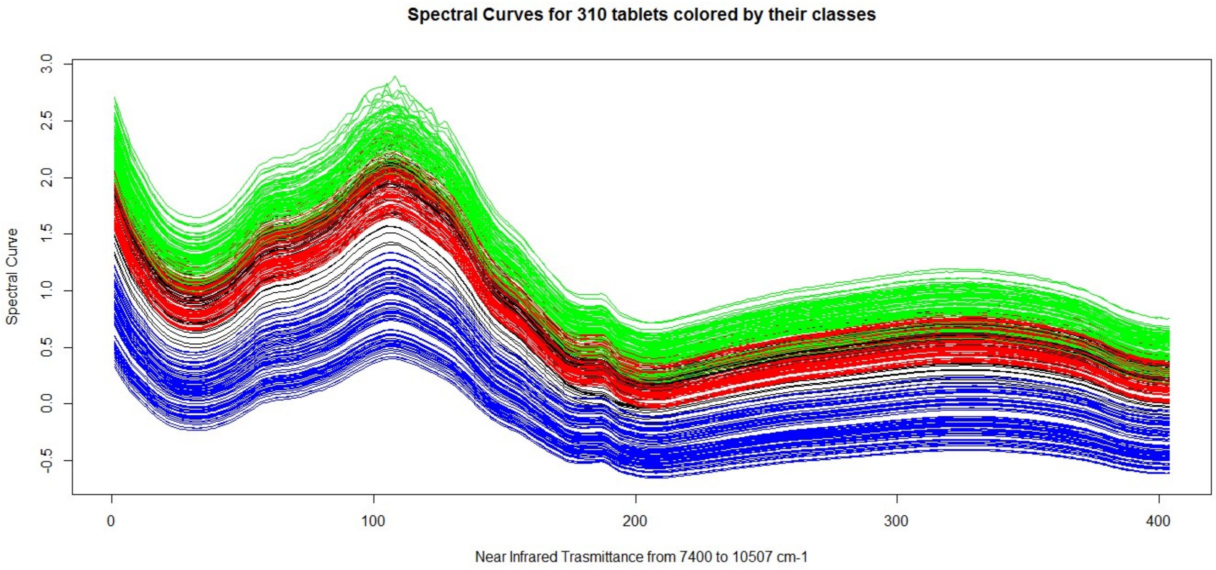
Diagram 1.2 shows that those significant coefficients come from first, second, third, fourth, and seventh curves. In other words, the emission spectra are affected mostly by excitation wavelengths of 230 nm, 240 nm, 255 nm, 290 nm, and 340 nm.

Functional Data Analysis
Project 1
Shuenn Siang Ng

Second Part: Pharmaceutical Tablets Data

The pharmaceutical tablets data are 310 by 404 matrix, where 310 is the same size and 404 is the number of equally spaced observation points from 7400 to 10507 cm^{-1}. These tablets belong to four classes. First, we plot all 310 curves and color them based on their classes, where class 1 is colored black, class 2 is colored blue, class 3 is colored red, and class 4 is colored green.

Diagram 2.1: Spectral Curves by Colors



Each curve clearly has the same pattern. Class 2, 3, and 4 seem to be clearly separated from each other. Class 2 curves are in the bottom, class 3 curves are in the middle and class 4 curves are at the top. On the other hand, class 1 curves are mixed in between class 2 and class 4.

We use the smooth FPCA to calculate the PC scores as predictor values for the linear discriminant analysis. We decided to first use 1000 basis functions to smooth the data. Then, we choose smoothing parameters and number of PC components by cross-validation. The following tables summarize the cross-validation classification errors:

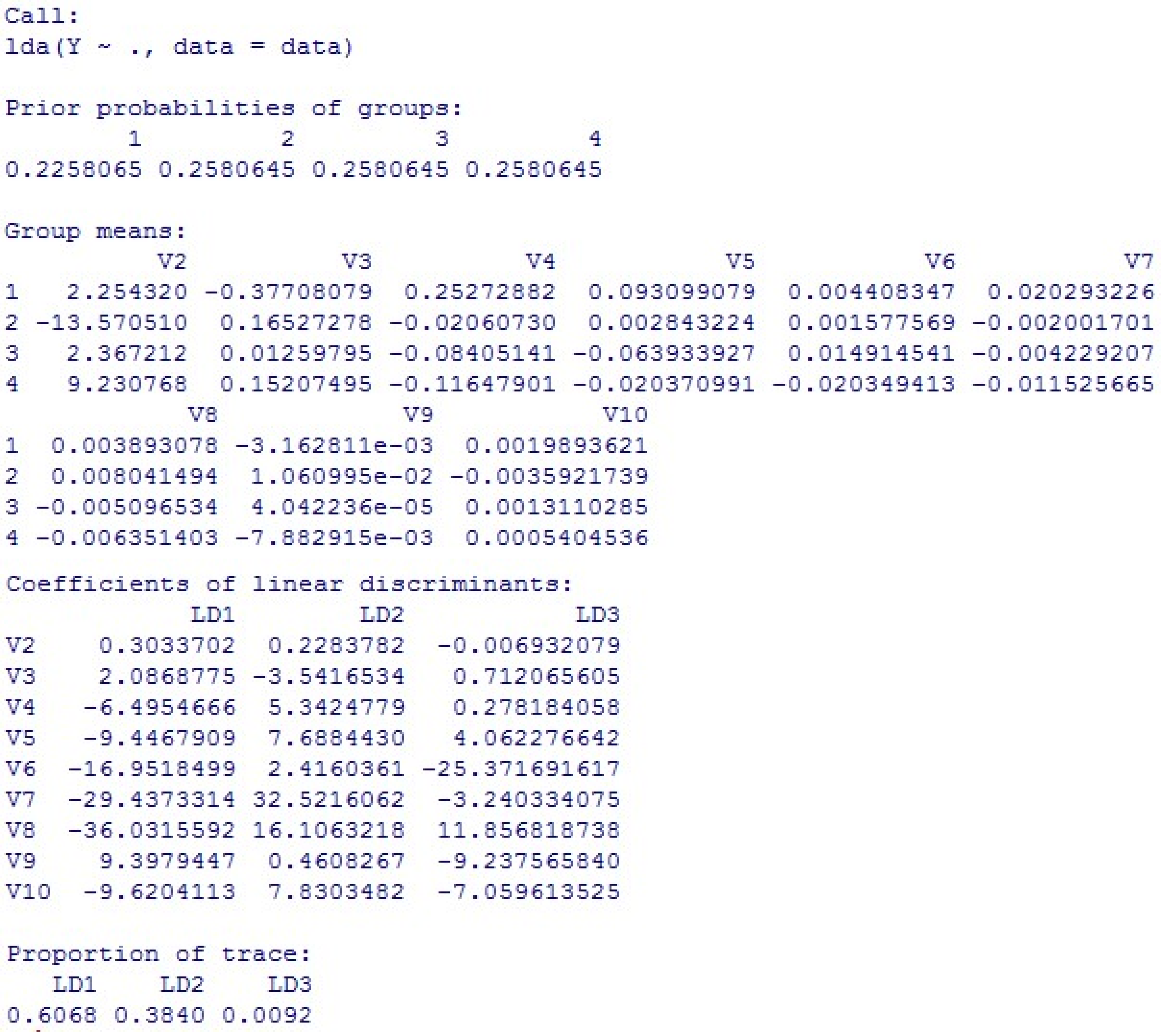
Table 2.1: Cross-Validation Classification Errors

λ \ M	1	2	3	4	5	6	7	8	9	10
10^{-10}	1.452	0.758	0.565	0.290	0.242	0.258	0.242	0.226	0.177	0.194
10^{-8}	1.452	0.758	0.565	0.290	0.242	0.258	0.242	0.226	0.177	0.194
10^{-6}	1.452	0.758	0.565	0.290	0.242	0.258	0.242	0.226	0.177	0.194
10^{-4}	1.452	0.758	0.565	0.290	0.242	0.258	0.242	0.226	0.177	0.194
10^{-2}	1.452	0.758	0.565	0.290	0.242	0.258	0.242	0.226	0.177	0.194
1	1.452	0.758	0.565	0.290	0.242	0.258	0.194	0.194	0.194	0.194
10	1.452	0.758	0.565	0.323	0.210	0.177	0.210	0.210	0.194	0.210

Functional Data Analysis
Project 1
Shuenn Siang Ng

We chose 10^{-6} and 9 to be the optimal tuning parameter and number of PC components, respectively. Using these optimal values, we calculate the PC scores and use these scores as predictor variables to build a classification model using Linear Discriminant Analysis.

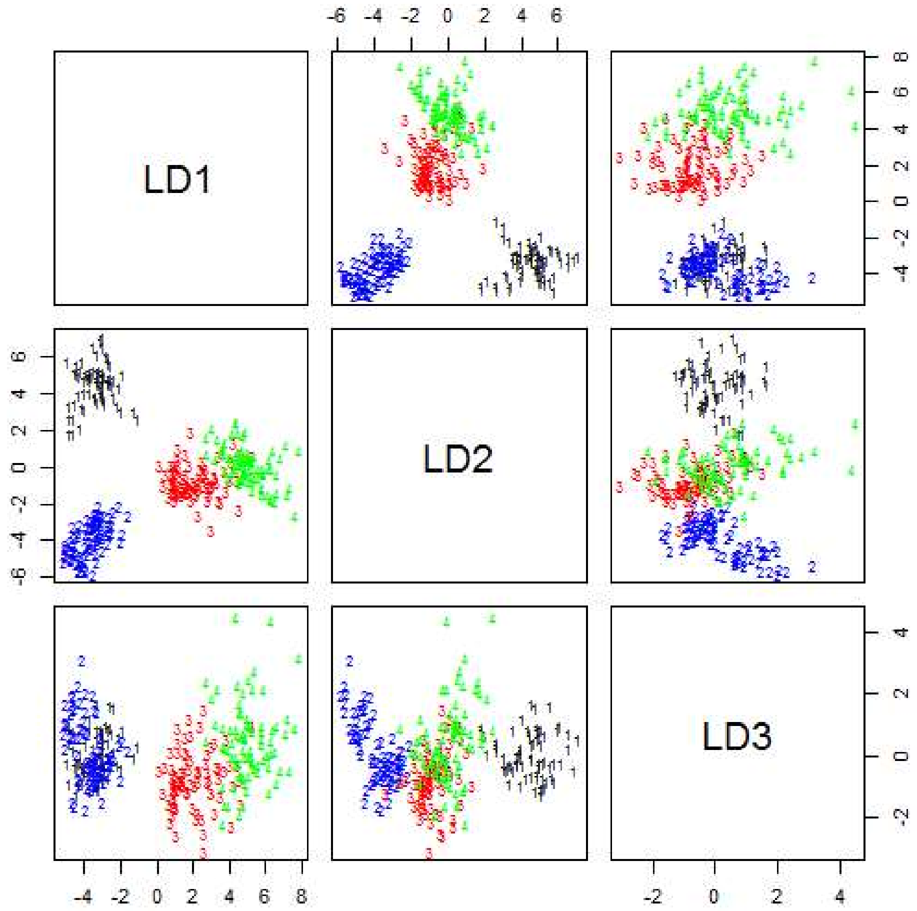
Diagram 2.2: Summary of Linear Discriminant Analysis



```
Call:
lda(Y ~ ., data = data)

Prior probabilities of groups:
        1         2         3         4 
0.2258065 0.2580645 0.2580645 0.2580645 

Group means:
         V2          V3          V4           V5           V6           V7
1  2.254320 -0.37708079  0.25272882  0.093099079  0.004408347  0.020293226
2 -13.570510  0.16527278 -0.02060730  0.002843224  0.001577569 -0.002001701
3  2.367212  0.01259795 -0.08405141 -0.063933927  0.014914541 -0.004229207
4  9.230768  0.15207495 -0.11647901 -0.020370991 -0.020349413 -0.011525665
           V8            V9          V10
1  0.003893078 -3.162811e-03  0.0019893621
2  0.008041494  1.060995e-02 -0.0035921739
3 -0.005096534  4.042236e-05  0.0013110285
4 -0.006351403 -7.882915e-03  0.0005404536

Coefficients of linear discriminants:
            LD1        LD2          LD3
V2    0.3033702  0.2283782  -0.006932079
V3    2.0868775 -3.5416534   0.712065605
V4   -6.4954666  5.3424779   0.278184058
V5   -9.4467909  7.6884430   4.062276642
V6  -16.9518499  2.4160361 -25.371691617
V7  -29.4373314 32.5216062  -3.240334075
V8  -36.0315592 16.1063218  11.856818738
V9    9.3979447  0.4608267  -9.237565840
V10  -9.6204113  7.8303482  -7.059613525

Proportion of trace:
   LD1    LD2    LD3 
0.6068 0.3840 0.0092 
```

To calculate the classification error, we use the leave-one-out method. We get that the classification error is 0.04516129. Also, we plot the class of the linear discriminant analysis using the color representation from above.

Diagram 2.3: Linear Discriminant Analysis Graphs



From Diagram 2.3, there are some overlaps between classes but they are pretty distinctively separated most of the times. I think the reason for this phenomenal is that the original curves have some overlaps as mentioned before. With a relatively small classification error, I would say this classification rule is a decent one.

Functional Data Analysis
Project 1
Shuenn Siang Ng

Appendix

Part 1:

```
X=read.table("project.one.1.X.txt")
s=seq(275,560,0.5)
X1=as.matrix(X[,1:571])
X2=as.matrix(X[,572:1142])
X3=as.matrix(X[,1143:1713])
X4=as.matrix(X[,1714:2284])
X5=as.matrix(X[,2285:2855])
X6=as.matrix(X[,2856:3426])
X7=as.matrix(X[,3427:3997])

matplot(s,t(X1),type="l",lwd=2,lty=1,
xlab="Wavelengths (nm)",ylab="Emission Spectra",
main="Excitation Wavelenth 230 nm")

matplot(s,t(X2),type="l",lwd=2,lty=1,
xlab="Wavelengths (nm)",ylab="Emission Spectra",
main="Excitation Wavelenth 240 nm")

matplot(s,t(X3),type="l",lwd=2,lty=1,
xlab="Wavelengths (nm)",ylab="Emission Spectra",
main="Excitation Wavelenth 255 nm")

matplot(s,t(X4),type="l",lwd=2,lty=1,
xlab="Wavelengths (nm)",ylab="Emission Spectra",
main="Excitation Wavelenth 290 nm")

matplot(s,t(X5),type="l",lwd=2,lty=1,
xlab="Wavelengths (nm)",ylab="Emission Spectra",
main="Excitation Wavelenth 305 nm")

matplot(s,t(X6),type="l",lwd=2,lty=1,
xlab="Wavelengths (nm)",ylab="Emission Spectra",
main="Excitation Wavelenth 325 nm")

matplot(s,t(X7),type="l",lwd=2,lty=1,
xlab="Wavelengths (nm)",ylab="Emission Spectra",
main="Excitation Wavelenth 340 nm")

Y=read.table("project.one.1.Y.txt")
Y1=as.vector(as.matrix(Y[,1]))
Y2=as.matrix(Y[,2])
n=1:197
summary(Y)
```

Functional Data Analysis
Project 1
Shuenn Siang Ng

```
par(mfrow=c(1,2))
boxplot(Y1,main="Boxplot of Color")
boxplot(Y2,main="Boxplot of Ash mesurement")

t=0:570
basis.obj=create.bspline.basis(c(0,571), nbasis=1000)
fdParobj=fdPar(basis.obj, 2, 1e-10)
X=scale(X,center=TRUE, scale=FALSE)
X1=as.matrix(X[,1:571])
X2=as.matrix(X[,572:1142])
X3=as.matrix(X[,1143:1713])
X4=as.matrix(X[,1714:2284])
X5=as.matrix(X[,2285:2855])
X6=as.matrix(X[,2856:3426])
X7=as.matrix(X[,3427:3997])

fit.list.1=smooth.basis(t,t(X1),fdParobj)
X1.fd=fit.list.1$fd
fit.list.2=smooth.basis(t,t(X2),fdParobj)
X2.fd=fit.list.2$fd
fit.list.3=smooth.basis(t,t(X3),fdParobj)
X3.fd=fit.list.3$fd
fit.list.4=smooth.basis(t,t(X4),fdParobj)
X4.fd=fit.list.4$fd
fit.list.5=smooth.basis(t,t(X5),fdParobj)
X5.fd=fit.list.5$fd
fit.list.6=smooth.basis(t,t(X6),fdParobj)
X6.fd=fit.list.6$fd
fit.list.7=smooth.basis(t,t(X7),fdParobj)
X7.fd=fit.list.7$fd

lambda=10^(c(-10,-8,-6,-4,-2,0,1))
M=1:10
cv.fold=split(sample(197),rep(1:5,length=197))
CV.error.1=array(0, c(length(lambda),length(M)))
CV.error.2=array(0, c(length(lambda),length(M)))

{for(j in 1:length(lambda))
{for(k in 1:length(M))
{ print(c("j,k=",j,k))

pca.Par=fdPar(basis.obj, 2, lambda[j])

pca.list.1=pca.fd(X1.fd, M[k], pca.Par)
pX1=pca.list.1$scores
pca.list.2=pca.fd(X2.fd, M[k], pca.Par)
pX2=pca.list.2$scores

pca.list.3=pca.fd(X3.fd, M[k], pca.Par)
pX3=pca.list.3$scores

pca.list.4=pca.fd(X4.fd, M[k], pca.Par)
pX4=pca.list.4$scores

pca.list.5=pca.fd(X5.fd, M[k], pca.Par)
pX5=pca.list.5$scores

pca.list.6=pca.fd(X6.fd, M[k], pca.Par)
pX6=pca.list.6$scores

pca.list.7=pca.fd(X7.fd, M[k], pca.Par)
pX7=pca.list.7$scores

pX=cbind(pX1,pX2,pX3,pX4,pX5,pX6,pX7)

for(l in 1:5)
{test=cv.fold[[l]]
train=setdiff(1:197, test)

fit.1=lm(Y1[train]~pX[train,])
Y1.test.pred=cbind(1,pX[test,])%*%coef(fit.1)
CV.error.1[j,k]=CV.error.1[j,k]+sum((Y1.test.pred-Y1[test])^2)
print(c("CV.error.1[j,k]=",CV.error.1[j,k]))

fit.2=lm(Y2[train]~pX[train,])
Y2.test.pred=cbind(1,pX[test,])%*%coef(fit.2)
CV.error.2[j,k]=CV.error.2[j,k]+sum((Y2.test.pred-Y2[test])^2)
print(c("CV.error.2[j,k]=",CV.error.2[j,k]))
}
}
}
}
cv1=which(CV.error.1==min(CV.error.1), arr.ind=TRUE)
cv2=which(CV.error.2==min(CV.error.2), arr.ind=TRUE)

pca.Par=fdPar(basis.obj, 2, lambda[5])

pca.list.1=pca.fd(X1.fd, M[1], pca.Par)
pX1=pca.list.1$scores
```

```
pca.list.2=pca.fd(X2.fd, M[1], pca.Par)
pX2=pca.list.2$scores

pca.list.3=pca.fd(X3.fd, M[1], pca.Par)
pX3=pca.list.3$scores

pca.list.4=pca.fd(X4.fd, M[1], pca.Par)
pX4=pca.list.4$scores

pca.list.5=pca.fd(X5.fd, M[1], pca.Par)
pX5=pca.list.5$scores

pca.list.6=pca.fd(X6.fd, M[1], pca.Par)
pX6=pca.list.6$scores

pca.list.7=pca.fd(X7.fd, M[1], pca.Par)
pX7=pca.list.7$scores

pX11=cbind(pX1,pX2,pX3,pX4,pX5,pX6,pX7)

fit.1=lm(Y1~pX11)
summary(fit.1)
par(mfrow=c(2,2))
plot(fit.1)

pca.Par=fdPar(basis.obj, 2, lambda[1])

pca.list.1=pca.fd(X1.fd, M[3], pca.Par)
pX1=pca.list.1$scores

pca.list.2=pca.fd(X2.fd, M[3], pca.Par)
pX2=pca.list.2$scores

pca.list.3=pca.fd(X3.fd, M[3], pca.Par)
pX3=pca.list.3$scores

pca.list.4=pca.fd(X4.fd, M[3], pca.Par)
pX4=pca.list.4$scores

pca.list.5=pca.fd(X5.fd, M[3], pca.Par)
pX5=pca.list.5$scores

pca.list.6=pca.fd(X6.fd, M[3], pca.Par)
pX6=pca.list.6$scores

pca.list.7=pca.fd(X7.fd, M[3], pca.Par)
```

```
pX7=pca.list.7$scores

pX12=cbind(pX1,pX2,pX3,pX4,pX5,pX6,pX7)

fit.2=lm(Y2~pX12)
summary(fit.2)
par(mfrow=c(2,2))
plot(fit.2)
```

Part 2:

```
X=as.matrix(read.table("project.one.2.X.txt"))
Y=as.vector(as.matrix(read.table("project.one.2.Y.txt")))
color=rep("black",length(Y))
color[Y==2]="blue"
color[Y==3]="red"
color[Y==4]="green"
par(mfrow=c(1,1))
matplot(t(X),type="l",lty=1,col=color,main="Spectral Curves for 310
tablets colored by their classes
",ylab="Spectral Curve", xlab="Near Infrared Trasmittance from 7400 to
10507 cm-1")

t=0:403
X=scale(X,center=TRUE,scale=FALSE)
basis.obj=create.bspline.basis(c(0,404), nbasis=1000)
fdParobj=fdPar(basis.obj, 2, 1e-10)
fit.list=smooth.basis(t,t(X),fdParobj)
X.fd=fit.list$fd

lambda=10^(c(-10,-8,-6,-4,-2,0,1))
M=1:10
cv.fold=split(sample(310),rep(1:5,length=310))
CV.error=array(0,c(length(lambda),length(M)))
for(j in 1:length(lambda))
{for(k in 1:length(M))
{ print(c("j,k=",j,k))

pca.Par=fdPar(basis.obj,2,lambda[j])
pca.list=pca.fd(X.fd,M[k],pca.Par)
pX=pca.list$scores
data=as.data.frame(cbind(Y,pX))

for(l in 1:5)
{test=cv.fold[[l]]
train=setdiff(1:310,test)
fit=lda(Y~.,data=data)
CV.error[j,k]=CV.error[j,k]+
sum(predict(fit, data[-train, ])$class!=data[-train,1])/length(test)
print(c("CV.error[j,k]=",CV.error[j,k]))
}
}
}
cv=which(CV.error==min(CV.error), arr.ind=TRUE)

pca.Par=fdPar(basis.obj,2,lambda[3])
pca.list=pca.fd(X.fd,M[9],pca.Par)
pX=pca.list$scores
data=as.data.frame(cbind(Y,pX))
fit=lda(Y~.,data=data)
plot(fit)

fit=lda(Y~.,data=data,CV=TRUE)
sum(fit$class!=data$Y)/length(Y)
```