M6L4 Homework Assignment

Joshua Conte

November 5, 2017

1 M6L4 Homework Assignment

R studio was configured with the following parameters before beginning the project:

```
# clears the console in RStudio
cat("\014")
```

```
# clears environment
rm(list = ls())

# Load required packages
require(ggplot2)
require(MASS)
require(car)
```

1.1 Load Data.

I opened the Wholesale customers Data Set using read.csv2 and dodfWCDloaded it directly from the UC Irvine Machine Learning Repository.

To format the data, the data is separated by ',', stringsAsFactors = FALSE so that the strings in a data frame will be treated as plain strings and not as factor variables. I set na strings for missing data. Once the data was loaded I added the column names and changed the data types to numeric and finally removed the text data type.

Below is my R code:

```
# Some csv files are really big and take a while to open.  This command checks to
# see if it is already opened, if it is, it does not open it again.
# I also omitted the first column
if (!exists("dfWCD")) {
dfWCD <-
  read.csv2("Wholesale customers data.csv",
    sep = ",",
    stringsAsFactors = FALSE,
    na.strings=c("","NA")
  )
}

# DodfWCDload directly from site (unreliable from Ecuador)
# if (!exists("dfWCD")) {
# dfWCD <-
#   read.csv2(
#     url(
#       "https://archive.ics.uci.edu/ml/machine-learning-databases/00292/Wholesale customers data.csv"
#     ),
#     sep = ",",
#     stringsAsFactors = FALSE,
#     na.strings=c("","NA")
#   )
# # Add a column so I know which study the data is referring to
# study <- sprintf("study_%s",seq(1:440))
# dfWCD$study<-study
# }
```

```r
# change 2 to 24 to numeric
dfWCD[1:8] <- sapply(dfWCD[1:8], as.numeric)

# Print first lines
str(dfWCD)
```

```
## 'data.frame':    440 obs. of  8 variables:
##  $ Channel         : num  2 2 2 1 2 2 2 2 1 2 ...
##  $ Region          : num  3 3 3 3 3 3 3 3 3 3 ...
##  $ Fresh           : num  12669 7057 6353 13265 22615 ...
##  $ Milk            : num  9656 9810 8808 1196 5410 ...
##  $ Grocery         : num  7561 9568 7684 4221 7198 ...
##  $ Frozen          : num  214 1762 2405 6404 3915 ...
##  $ Detergents_Paper: num  2674 3293 3516 507 1777 ...
##  $ Delicassen      : num  1338 1776 7844 1788 5185 ...
```

1.1.1 Understanding the data

The data set refers to clients of a wholesale distributor in Portugal. It includes the annual spending in monetary units (m.u.) on diverse product categories. The data has the following attribute information:

1. FRESH: annual spending (m.u.) on fresh products (Continuous);
2. MILK: annual spending (m.u.) on Fresh products (Continuous);
3. GROCERY: annual spending (m.u.)on grocery products (Continuous);
4. FROZEN: annual spending (m.u.)on frozen products (Continuous)
5. DETERGENTS_PAPER: annual spending (m.u.) on detergents and paper products (Continuous)
6. DELICATESSEN: annual spending (m.u.)on and delicatessen products (Continuous);
7. CHANNEL: customer channel - 1 = Horeca (Hotel/Restaurant/Cafe) or 2 = Retail
8. REGION: Customers Region - 1= Lisnon 2 = Oporto or 3 = Other (Nominal)

1.2 Linear Discriminant Analysis in R

Linear Discriminant Analysis (LDA) is a generalization of Fisher's linear discriminant to find a linear combination of features that characterizes or separates two or more classes of objects or events. Discriminant analysis seeks to generate lines that are efficient for discrimination.

LDA is also closely related to principal component analysis (PCA) and factor analysis in that they both look for linear combinations of variables which best explain the data. In the case of LDA, we are maximizing the linear compenent axes for class discrimination. In the case of PCA, we are finding basis that maximize the variance.

LDA can also be used as a supervised technique by finding a discriminant projection that maximizing between-class distance and minimizing within-class distance.

Below is the R code:

```r
head(dfWCD)
```

```
##   Channel Region Fresh Milk Grocery Frozen Detergents_Paper Delicassen
## 1       2      3 12669 9656    7561    214             2674       1338
## 2       2      3  7057 9810    9568   1762             3293       1776
## 3       2      3  6353 8808    7684   2405             3516       7844
## 4       1      3 13265 1196    4221   6404              507       1788
## 5       2      3 22615 5410    7198   3915             1777       5185
## 6       2      3  9413 8259    5126    666             1795       1451
```

```
summary(dfWCD)
```

```
##     Channel          Region          Fresh             Milk      
##  Min.   :1.000   Min.   :1.000   Min.   :     3   Min.   :   55  
##  1st Qu.:1.000   1st Qu.:2.000   1st Qu.:  3128   1st Qu.: 1533  
##  Median :1.000   Median :3.000   Median :  8504   Median : 3627  
##  Mean   :1.323   Mean   :2.543   Mean   : 12000   Mean   : 5796  
##  3rd Qu.:2.000   3rd Qu.:3.000   3rd Qu.: 16934   3rd Qu.: 7190  
##  Max.   :2.000   Max.   :3.000   Max.   :112151   Max.   :73498  
##     Grocery          Frozen        Detergents_Paper    Delicassen     
##  Min.   :    3   Min.   :   25.0   Min.   :    3.0   Min.   :    3.0  
##  1st Qu.: 2153   1st Qu.:  742.2   1st Qu.:  256.8   1st Qu.:  408.2  
##  Median : 4756   Median : 1526.0   Median :  816.5   Median :  965.5  
##  Mean   : 7951   Mean   : 3071.9   Mean   : 2881.5   Mean   : 1524.9  
##  3rd Qu.:10656   3rd Qu.: 3554.2   3rd Qu.: 3922.0   3rd Qu.: 1820.2  
##  Max.   :92780   Max.   :60869.0   Max.   :40827.0   Max.   :47943.0
```

```
length(dfWCD)
```

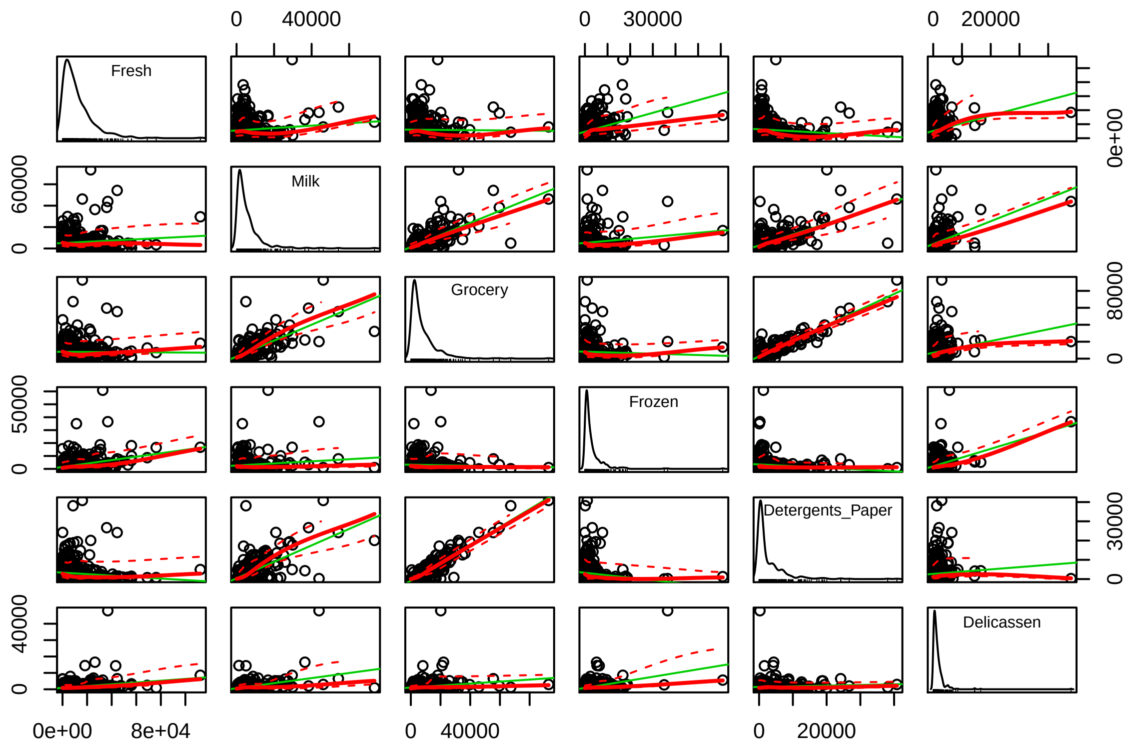
```
## [1] 8
```

This plots the data:

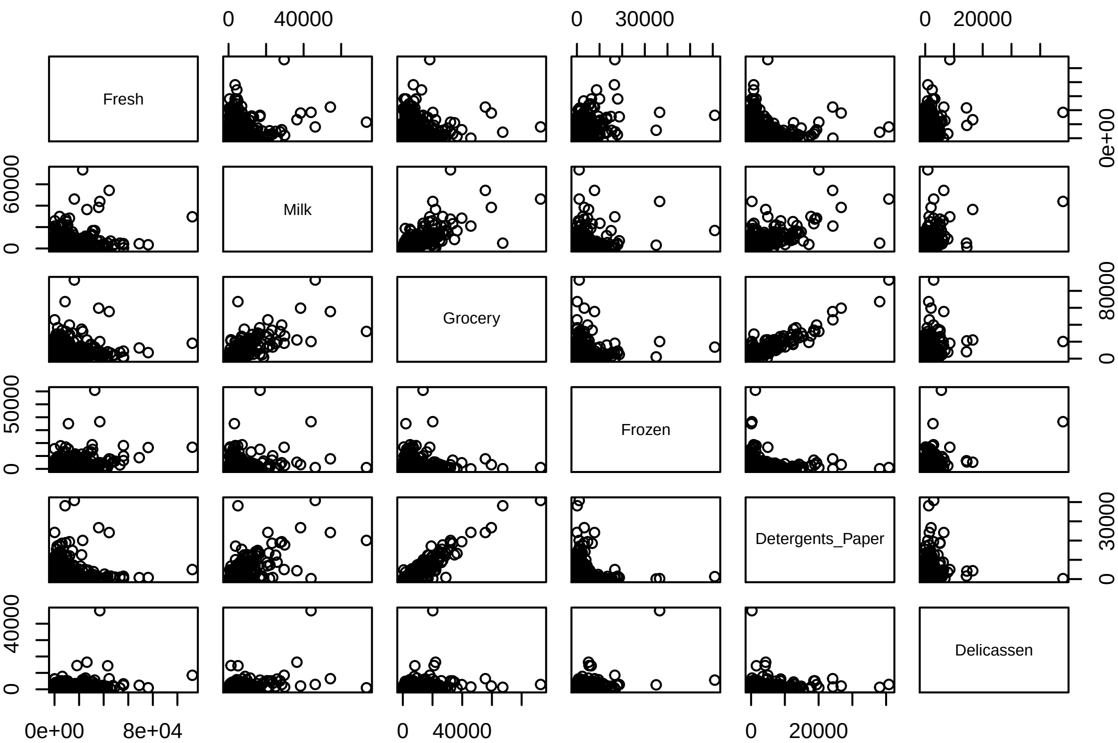
```
names(dfWCD)
```

```
## [1] "Channel"          "Region"           "Fresh"           
## [4] "Milk"             "Grocery"          "Frozen"          
## [7] "Detergents_Paper" "Delicassen"
```

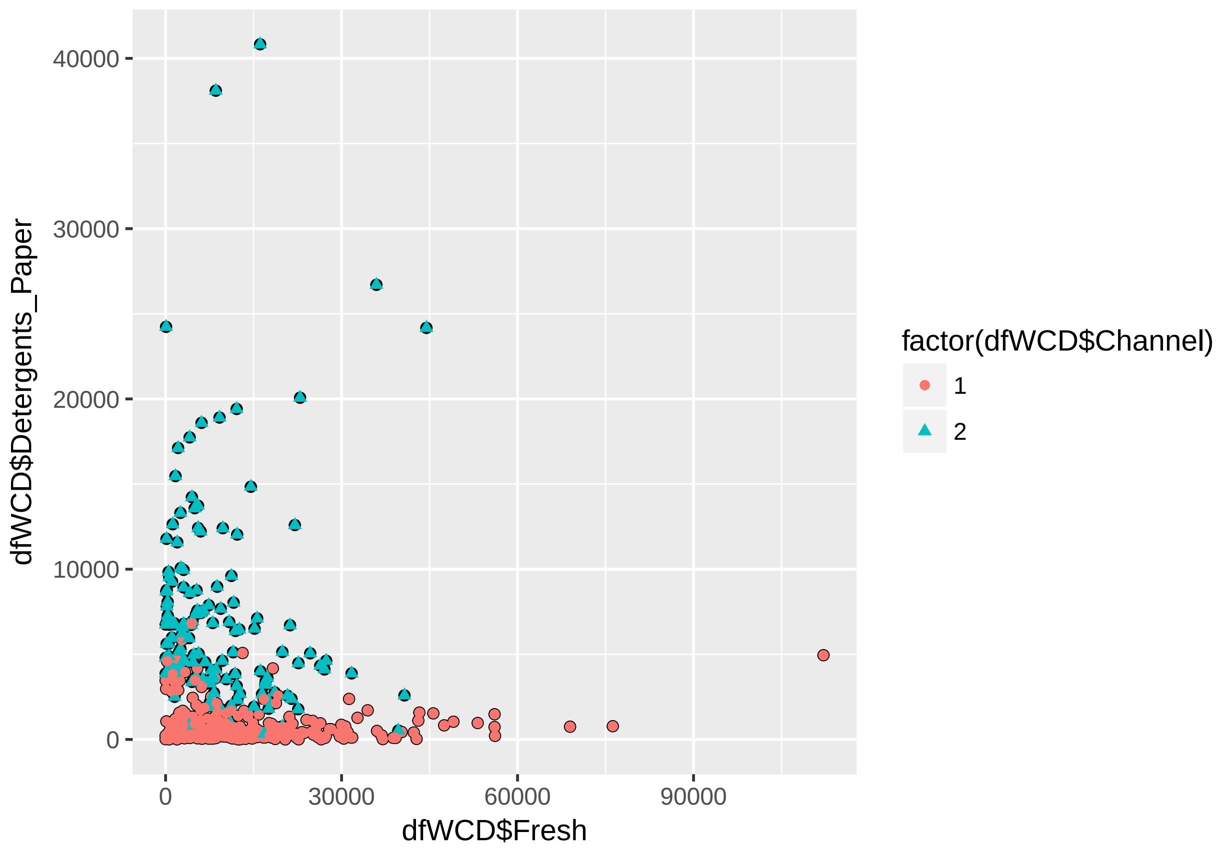
```
scatterplotMatrix(dfWCD[3:8])
```



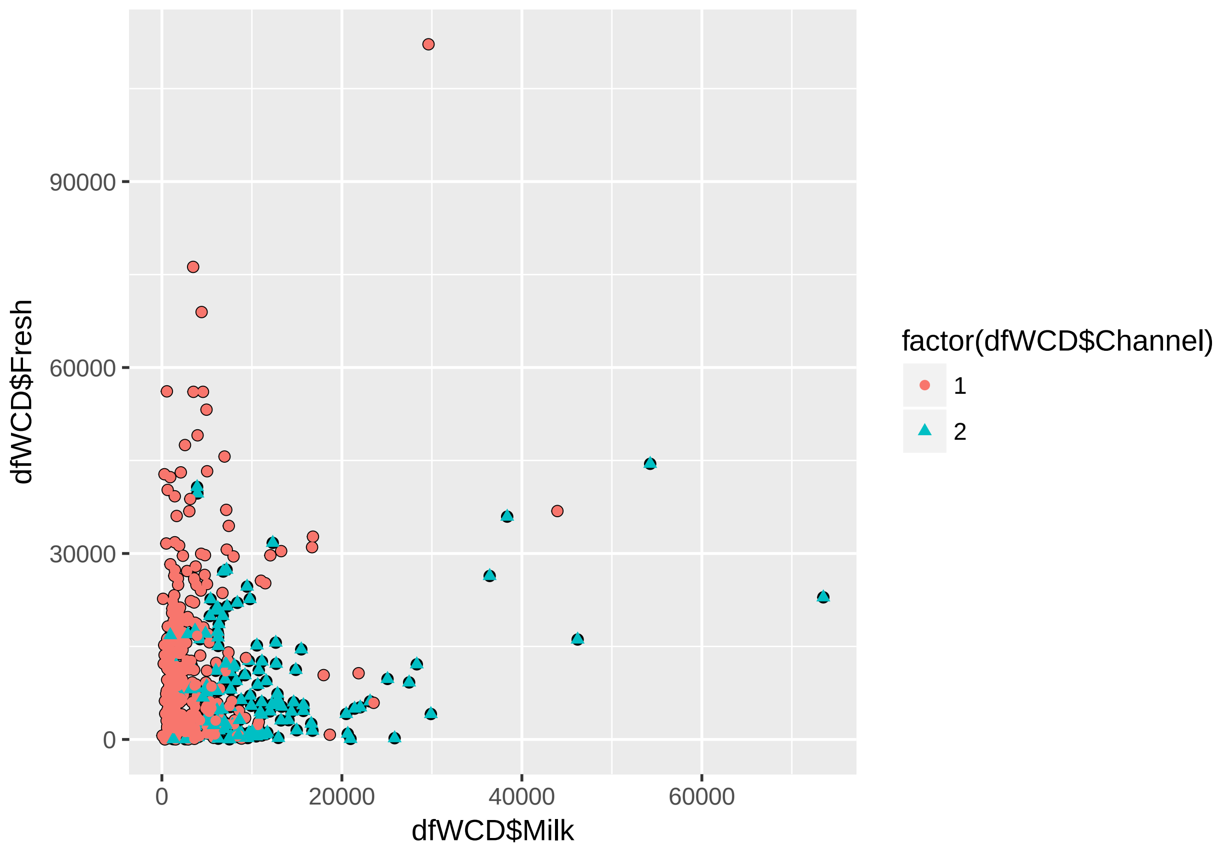

```r
pairs(dfWCD[,3:8])
```




```r
qplot(dfWCD$Fresh,dfWCD$Detergents_Paper,data=dfWCD)+geom_point(aes(colour = factor(dfWCD$Channel),shap
```




```
qplot(dfWCD$Milk,dfWCD$Fresh,data=dfWCD)+geom_point(aes(colour = factor(dfWCD$Channel),shape = factor(d
```



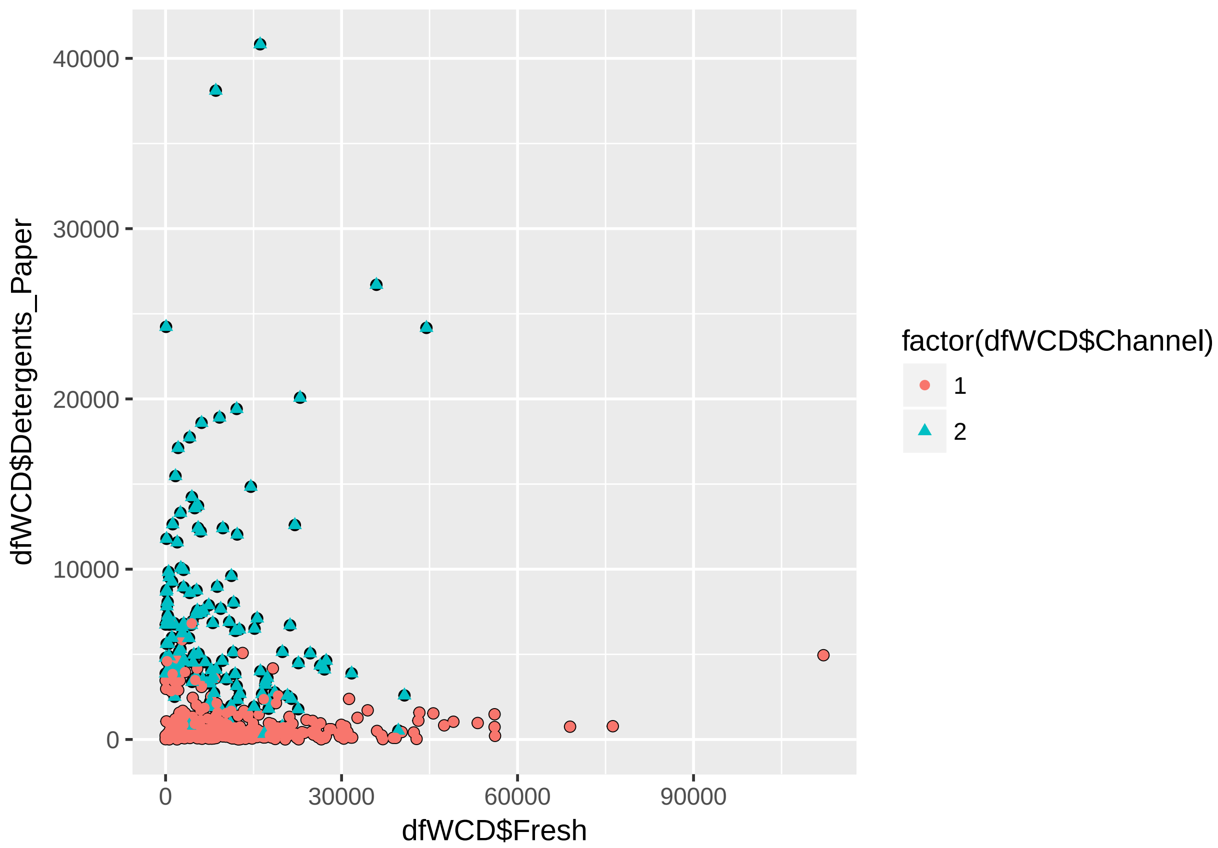
Here I am using two predictor variables and Channel as the response variable.

```
lsa.m1<-lda(Channel ~ Fresh + Detergents_Paper, data=dfWCD)
lsa.m1
```

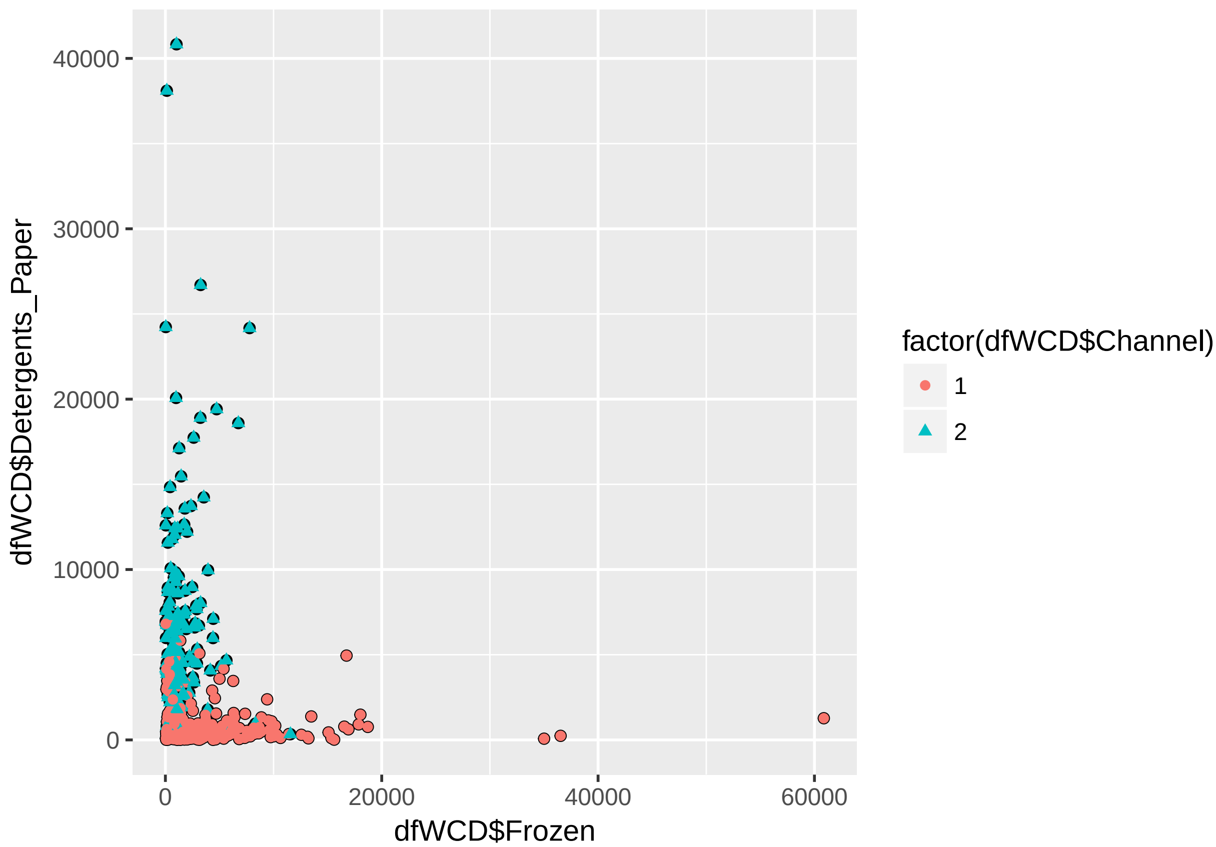
```
## Call:
## lda(Channel ~ Fresh + Detergents_Paper, data = dfWCD)
##
## Prior probabilities of groups:
##         1         2
## 0.6772727 0.3227273
##
## Group means:
##       Fresh Detergents_Paper
## 1 13475.560         790.5604
## 2  8904.324        7269.5070
##
## Coefficients of linear discriminants:
##                            LD1
## Fresh            -1.689506e-05
## Detergents_Paper  2.658111e-04
```

Here I plotted to see what Fresh and Detergents_Paper looked like

```
qplot(dfWCD$Fresh,dfWCD$Detergents_Paper,data=dfWCD)+geom_point(aes(colour = factor(dfWCD$Channel),shap
```



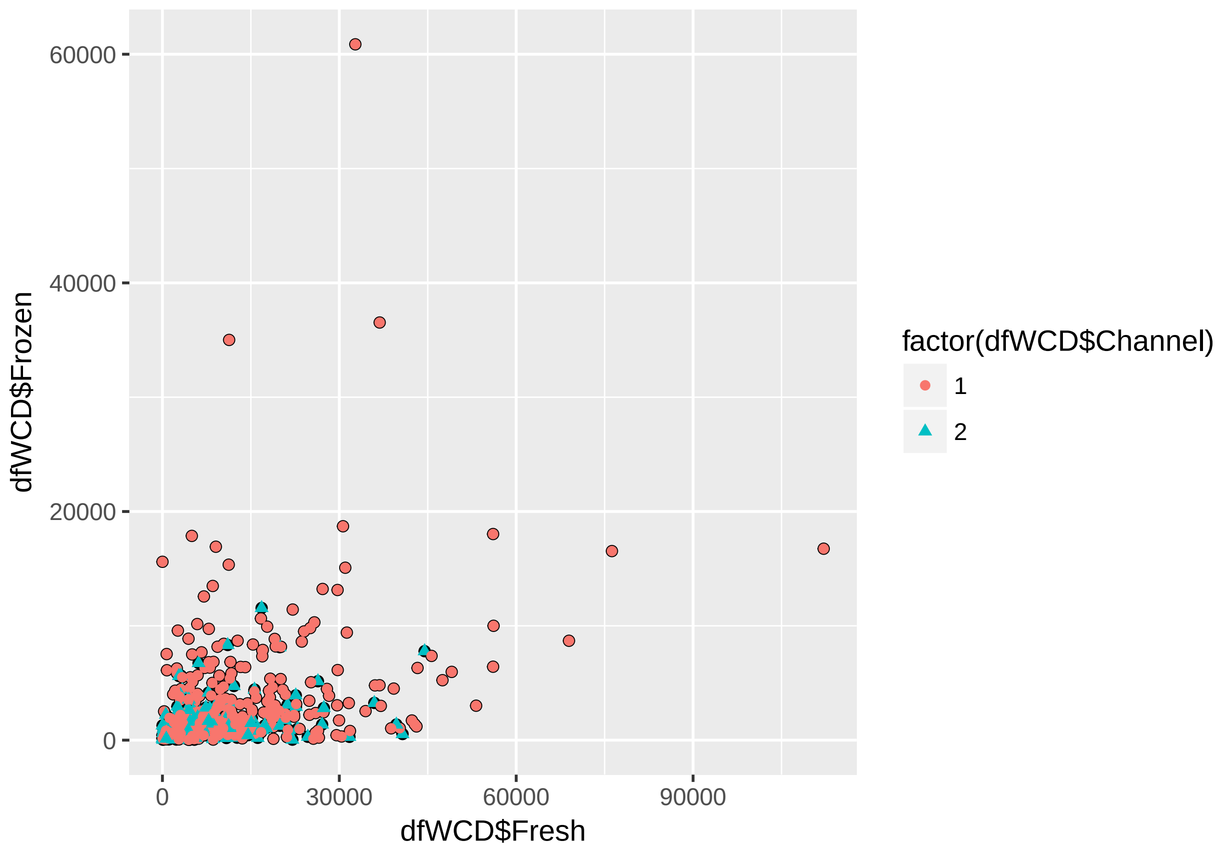

```
qplot(dfWCD$Frozen,dfWCD$Detergents_Paper,data=dfWCD)+geom_point(aes(colour = factor(dfWCD$Channel),shap
```




```
lsa.m2<-lda(Channel ~ Detergents_Paper + Frozen, data=dfWCD)
lsa.m2
```

```
## Call:
## lda(Channel ~ Detergents_Paper + Frozen, data = dfWCD)
##
## Prior probabilities of groups:
##         1         2
## 0.6772727 0.3227273
##
## Group means:
##   Detergents_Paper   Frozen
## 1         790.5604 3748.252
## 2        7269.5070 1652.613
##
## Coefficients of linear discriminants:
##                            LD1
## Detergents_Paper  2.633510e-04
## Frozen           -5.024385e-05
```

```
qplot(dfWCD$Fresh,dfWCD$Frozen,data=dfWCD)+geom_point(aes(colour = factor(dfWCD$Channel),shape = factor
```




```
lsa.m3<-lda(Channel ~ Fresh + Detergents_Paper, data=dfWCD)
lsa.m3
```

```
## Call:
## lda(Channel ~ Fresh + Detergents_Paper, data = dfWCD)
##
## Prior probabilities of groups:
##         1         2
## 0.6772727 0.3227273
##
## Group means:
##       Fresh Detergents_Paper
## 1 13475.560        790.5604
## 2  8904.324       7269.5070
##
## Coefficients of linear discriminants:
##                           LD1
## Fresh            -1.689506e-05
## Detergents_Paper  2.658111e-04
```

```
names(dfWCD) # Fresh (2) + Malic.acid(3) + Detergents_Paper (4)
```

```
## [1] "Channel"          "Region"           "Fresh"
## [4] "Milk"             "Grocery"          "Frozen"
## [7] "Detergents_Paper" "Delicassen"
```

```
lsa.m2.p<-predict(lsa.m2, newdata = dfWCD[,c(6,7)])
summary(lsa.m2.p)
```

```
##           Length Class  Mode
## class     440    factor numeric
## posterior 880    -none- numeric
## x         440    -none- numeric
```

```
#lsa.m2.p$class
```

This uses predict to see what it would have gotton based on the model

```
lsa.m1.p<-predict(lsa.m1, newdata = dfWCD[,c(3,7)])
```

This evaluates the models.

```
cm.m1<-table(lsa.m1.p$class,dfWCD[,c(1)])
cm.m1
```

```
##
##       1   2
##   1 295  68
##   2   3  74
```

```
cm.m2<-table(lsa.m2.p$class,dfWCD[,c(1)])
cm.m2
```

```
##
##       1   2
##   1 295  70
##   2   3  72
```

1.2.1 Additional predictors

Three predictors:

```
lsa.m4<-lda(Channel ~ Fresh + Detergents_Paper + Frozen, data=dfWCD)
lsa.m4
```

```
## Call:
## lda(Channel ~ Fresh + Detergents_Paper + Frozen, data = dfWCD)
##
## Prior probabilities of groups:
##         1         2
## 0.6772727 0.3227273
##
## Group means:
##       Fresh Detergents_Paper   Frozen
## 1 13475.560         790.5604 3748.252
## 2  8904.324        7269.5070 1652.613
##
## Coefficients of linear discriminants:
##                            LD1
## Fresh            -1.171229e-05
## Detergents_Paper  2.611651e-04
## Frozen           -3.984417e-05
```

```
lsa.m4.p<-predict(lsa.m4, newdata = dfWCD[,c(3,7,6)])
cm.m4<-table(lsa.m4.p$class,dfWCD[,c(1)])
```

```
cm.m4
```

```
##
##       1   2
##   1 295  67
##   2   3  75
```

Four predictors:

```
lsa.m5<-lda(Channel ~ Fresh + Detergents_Paper + Frozen + Milk, data=dfWCD)
lsa.m5
```

```
## Call:
## lda(Channel ~ Fresh + Detergents_Paper + Frozen + Milk, data = dfWCD)
##
## Prior probabilities of groups:
##         1         2
## 0.6772727 0.3227273
##
## Group means:
##       Fresh Detergents_Paper   Frozen      Milk
## 1 13475.560         790.5604 3748.252  3451.725
## 2  8904.324        7269.5070 1652.613 10716.500
##
## Coefficients of linear discriminants:
##                            LD1
## Fresh            -1.418387e-05
## Detergents_Paper  2.183278e-04
## Frozen           -5.025145e-05
## Milk              3.901105e-05
```

```
lsa.m5.p<-predict(lsa.m5, newdata = dfWCD[,c(3,7,6, 4)])
cm.m5<-table(lsa.m5.p$class,dfWCD[,c(1)])
cm.m5
```

```
##
##       1   2
##   1 295  63
##   2   3  79
```

Five predictors:

```
lsa.m6<-lda(Channel ~ Fresh + Detergents_Paper + Frozen + Milk + Grocery, data=dfWCD)
lsa.m6
```

```
## Call:
## lda(Channel ~ Fresh + Detergents_Paper + Frozen + Milk + Grocery,
##     data = dfWCD)
##
## Prior probabilities of groups:
##         1         2
## 0.6772727 0.3227273
##
## Group means:
##       Fresh Detergents_Paper   Frozen      Milk   Grocery
## 1 13475.560         790.5604 3748.252  3451.725  3962.138
## 2  8904.324        7269.5070 1652.613 10716.500 16322.852
##
```

```
## Coefficients of linear discriminants:
##                           LD1
## Fresh            -1.538619e-05
## Detergents_Paper  1.562846e-04
## Frozen           -5.238007e-05
## Milk              2.993809e-05
## Grocery           3.823092e-05
```

```
lsa.m6.p<-predict(lsa.m6, newdata = dfWCD[,c(3,7,6,4,5)])
cm.m6<-table(lsa.m6.p$class,dfWCD[,c(1)])
cm.m6
```

```
##
##       1   2
##   1 295  62
##   2   3  80
```

1.2.2 Normalizing and mixing up the data

```
# create a random sample for training and test data
set.seed(12345)
dfWCD_rand <- dfWCD[order(runif(440)), ]

# normalize
normalize<- function(x) {
  return((x-min(x))/(max(x)-min(x)))
}
dfWCD_rand.normalized<-as.data.frame(lapply(dfWCD_rand,normalize))

lsa.m1n<-lda(Channel ~ Fresh + Detergents_Paper, data=dfWCD_rand.normalized)
lsa.m1n
```

```
## Call:
## lda(Channel ~ Fresh + Detergents_Paper, data = dfWCD_rand.normalized)
##
## Prior probabilities of groups:
##         0         1
## 0.6772727 0.3227273
##
## Group means:
##        Fresh Detergents_Paper
## 0 0.12013197        0.0192916
## 1 0.07937122        0.1779960
##
## Coefficients of linear discriminants:
##                        LD1
## Fresh            -1.894747
## Detergents_Paper 10.851471
```

```
lsa.m1n.p<-predict(lsa.m1n, newdata = dfWCD[,c(3,7)])
cm.m1n<-table(lsa.m1n.p$class,dfWCD[,c(1)])
cm.m1n
```

```
##
##       1   2
```

```
##   0 230  18
##   1  68 124
```

```
lsa.m2n<-lda(Channel ~ Detergents_Paper + Frozen, data=dfWCD_rand.normalized)
lsa.m2n
```

```
## Call:
## lda(Channel ~ Detergents_Paper + Frozen, data = dfWCD_rand.normalized)
##
## Prior probabilities of groups:
##         0         1
## 0.6772727 0.3227273
##
## Group means:
##   Detergents_Paper     Frozen
## 0        0.0192916 0.06119341
## 1        0.1779960 0.02675059
##
## Coefficients of linear discriminants:
##                        LD1
## Detergents_Paper 10.751039
## Frozen           -3.057037
```

```
lsa.m2n.p<-predict(lsa.m2n, newdata = dfWCD[,c(6,7)])
cm.m2n<-table(lsa.m2n.p$class,dfWCD[,c(1)])
cm.m2n
```

```
##
##       1   2
##   0 186   4
##   1 112 138
```

1.3 Questions

1. Does the number of predictor variables for LDA make a difference? Try for a range of models using differing numbers of predictor variables.
   - It makes a little difference, but not significant. For 3 predictors, it was the same. With 4 predictors, the numbers improved a little and same with 6.
2. What determines the number of linear discriminants in LDA.
   - LDA finds at most k???1 linear discriminants, where k is the number of classes. In my data I have two classes (the Channel variable) hence only 1 linear discriminants can be resolved.
3. Does scaling, normalization or leaving the data unscaled make a difference for LDA?
   - Normalizing the data made it different, arguably better for this data. Without normalizing the data lsa.m1 had 1% error 295 of 298 good in the first column (48% error 74 of 142 in the second), when it was normalized it became 23% error 230 of 298 (3% error 138 of 142 in the second). lsa.m2 was worse, 1% in the first column and 47% in the second, when it was normalized 37% in the first and 3% in the second.