ICT513 - Assignment 3

Chew Jian Yue

Contents

1 Question 1

The objective of the Principal Component Analysis (PCA) is to reduce the large number of variables to a smaller set of Principal Components (PCs) that capture as much variability and information as possible across the set of variables.

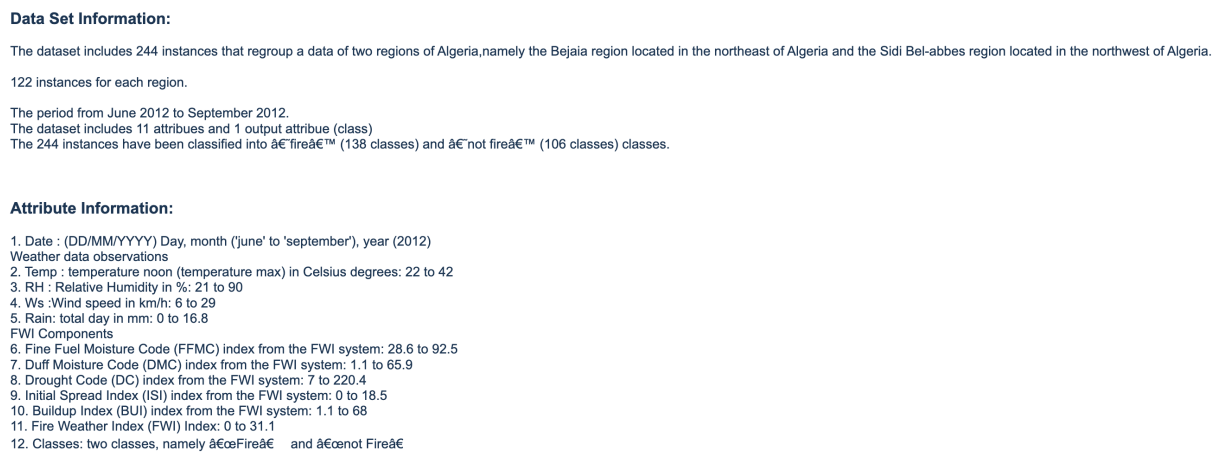
1.1 Understanding the dataset

For the Algerian Forest Fires Dataset, there are a total of 244 instances/observations regrouped into two *balanced* regions. The dataset represents data collected during the period of June 2012 and Sept 2012.

```
glimpse(forestf)
```

```
Rows: 244
Columns: 14
$ day         <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,~
$ month       <int> 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6~
$ year        <int> 2012, 2012, 2012, 2012, 2012, 2012, 2012, 2012, 2012, 2012~
$ Temperature <int> 29, 29, 26, 25, 27, 31, 33, 30, 25, 28, 31, 26, 27, 30, 28~
$ RH          <int> 57, 61, 82, 89, 77, 67, 54, 73, 88, 79, 65, 81, 84, 78, 80~
$ Ws          <int> 18, 13, 22, 13, 16, 14, 13, 15, 13, 12, 14, 19, 21, 20, 17~
$ Rain        <dbl> 0.0, 1.3, 13.1, 2.5, 0.0, 0.0, 0.0, 0.0, 0.2, 0.0, 0.0, 0.~
$ FFMC        <dbl> 65.7, 64.4, 47.1, 28.6, 64.8, 82.6, 88.2, 86.6, 52.9, 73.2~
$ DMC         <dbl> 3.4, 4.1, 2.5, 1.3, 3.0, 5.8, 9.9, 12.1, 7.9, 9.5, 12.5, 1~
$ DC          <chr> "7.6", "7.6", "7.1", "6.9", "14.2", "22.2", "30.5", "38.3"~
$ ISI         <dbl> 1.3, 1.0, 0.3, 0.0, 1.2, 3.1, 6.4, 5.6, 0.4, 1.3, 4.0, 4.8~
$ BUI         <dbl> 3.4, 3.9, 2.7, 1.7, 3.9, 7.0, 10.9, 13.5, 10.5, 12.6, 15.8~
$ FWI         <chr> "0.5", "0.4", "0.1", "0", "0.5", "2.5", "7.2", "7.1", "0.3~
$ Classes     <chr> "not fire   ", "not fire   ", "not fire   ", "not fire   "~
```

Ensuring that all the observations in the dataset are within the range of values, by looking at the maximum and minimum values, as found on the website.



Data Set Information:

The dataset includes 244 instances that regroup a data of two regions of Algeria,namely the Bejaia region located in the northeast of Algeria and the Sidi Bel-abbes region located in the northwest of Algeria.

122 instances for each region.

The period from June 2012 to September 2012.
The dataset includes 11 attribues and 1 output attribue (class)
The 244 instances have been classified into â€˜fireâ€™ (138 classes) and â€˜not fireâ€™ (106 classes) classes.

Attribute Information:

1. Date : (DD/MM/YYYY) Day, month ('june' to 'september'), year (2012)
Weather data observations
2. Temp : temperature noon (temperature max) in Celsius degrees: 22 to 42
3. RH : Relative Humidity in %: 21 to 90
4. Ws :Wind speed in km/h: 6 to 29
5. Rain: total day in mm: 0 to 16.8
FWI Components
6. Fine Fuel Moisture Code (FFMC) index from the FWI system: 28.6 to 92.5
7. Duff Moisture Code (DMC) index from the FWI system: 1.1 to 65.9
8. Drought Code (DC) index from the FWI system: 7 to 220.4
9. Initial Spread Index (ISI) index from the FWI system: 0 to 18.5
10. Buildup Index (BUI) index from the FWI system: 1.1 to 68
11. Fire Weather Index (FWI) Index: 0 to 31.1
12. Classes: two classes, namely â€œFireâ€ and â€œnot Fireâ€

Figure 1: Screenshot of information about the dataset

```
      day             month            year       Temperature          RH
 Min.   : 1.00   Min.   :6.0   Min.   :2012   Min.   :22.00   Min.   :21.00
 1st Qu.: 8.00   1st Qu.:7.0   1st Qu.:2012   1st Qu.:30.00   1st Qu.:52.00
 Median :16.00   Median :7.5   Median :2012   Median :32.00   Median :63.00
 Mean   :15.75   Mean   :7.5   Mean   :2012   Mean   :32.17   Mean   :61.94
 3rd Qu.:23.00   3rd Qu.:8.0   3rd Qu.:2012   3rd Qu.:35.00   3rd Qu.:73.25
 Max.   :31.00   Max.   :9.0   Max.   :2012   Max.   :42.00   Max.   :90.00

       Ws              Rain              FFMC            DMC
 Min.   : 6.0   Min.   : 0.0000   Min.   :28.60   Min.   : 0.70
 1st Qu.:14.0   1st Qu.: 0.0000   1st Qu.:72.08   1st Qu.: 5.80
 Median :15.0   Median : 0.0000   Median :83.50   Median :11.30
 Mean   :15.5   Mean   : 0.7607   Mean   :77.89   Mean   :14.67
 3rd Qu.:17.0   3rd Qu.: 0.5000   3rd Qu.:88.30   3rd Qu.:20.75
 Max.   :29.0   Max.   :16.8000   Max.   :96.00   Max.   :65.90

       DC              ISI              BUI             FWI
 Min.   :  6.90   Min.   : 0.000   Min.   : 1.10   Min.   : 0.000
 1st Qu.: 12.35   1st Qu.: 1.400   1st Qu.: 6.00   1st Qu.: 0.700
 Median : 33.10   Median : 3.500   Median :12.25   Median : 4.200
 Mean   : 49.43   Mean   : 4.774   Mean   :16.66   Mean   : 7.035
 3rd Qu.: 69.10   3rd Qu.: 7.300   3rd Qu.:22.52   3rd Qu.:11.450
 Max.   :220.40   Max.   :19.000   Max.   :68.00   Max.   :31.100
 NA's   :1                                        NA's   :1
   Classes                date
 Length:244          Min.   :2012-06-01
```

```
 Class :character   1st Qu.:2012-07-01
 Mode  :character   Median :2012-07-31
                    Mean   :2012-07-31
                    3rd Qu.:2012-08-31
                    Max.   :2012-09-30
```

There should only be two factor levels for `Classes` variable, either `not fire` or `fire`. The dataset provided has many levels because of trailing and leading whitespaces. Remove the trailing and leading spaces for the `Classes` variable, factor levels. Also, I remove observations with empty factor levels (containing only whitespaces).

```
[1] not fire fire
Levels: fire not fire
```

After removing the single observation with `Classes` as not labelled whitespace, we are left with 137 observations of `Classes` labelled `fire`, and 106 observations of `not fire`.

Classes	Count
fire	137
not fire	106

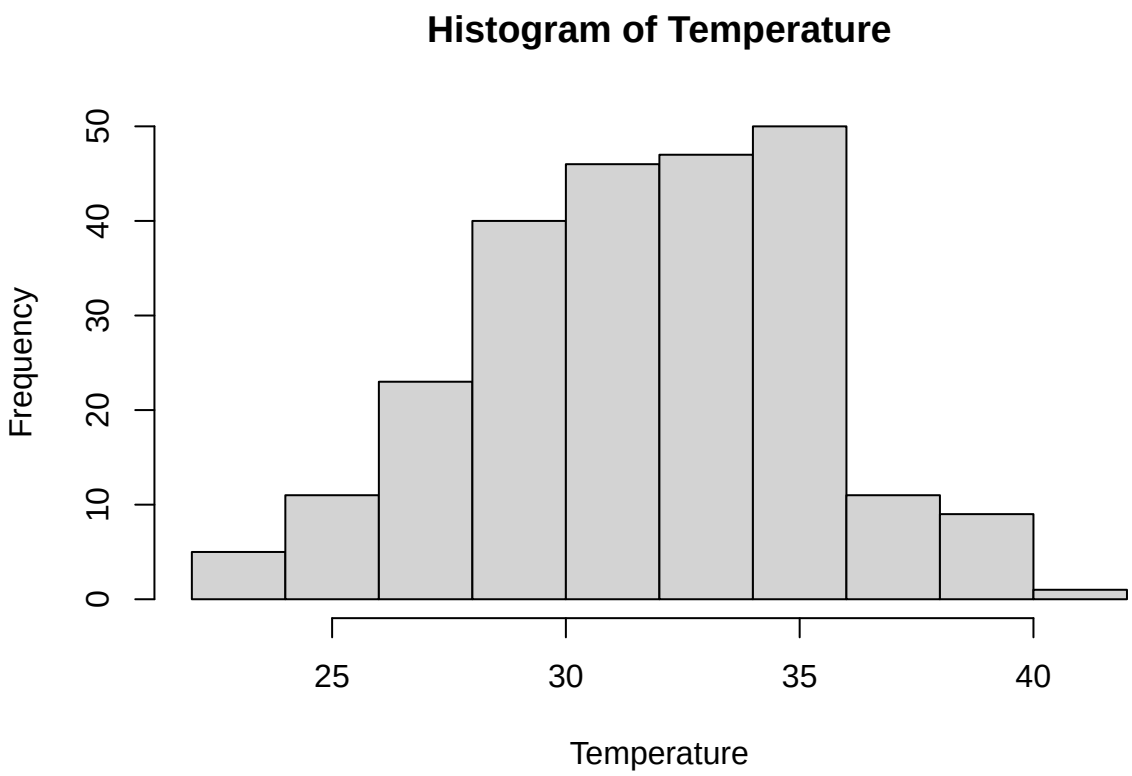
Doesn't make sense to apply PCA on index-like value such as day, month and year.

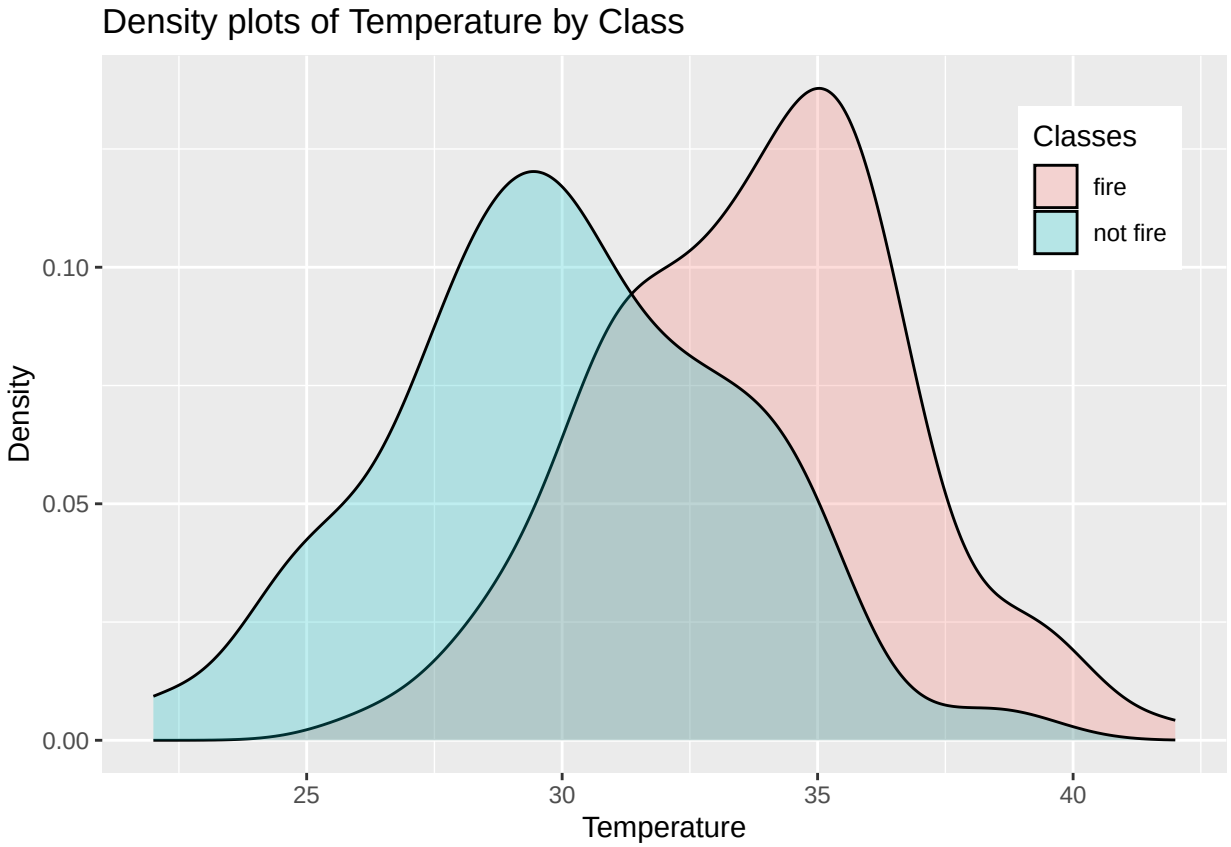
I will remove these data through `select` method and excluding the variables `day`, `month` and `year`.

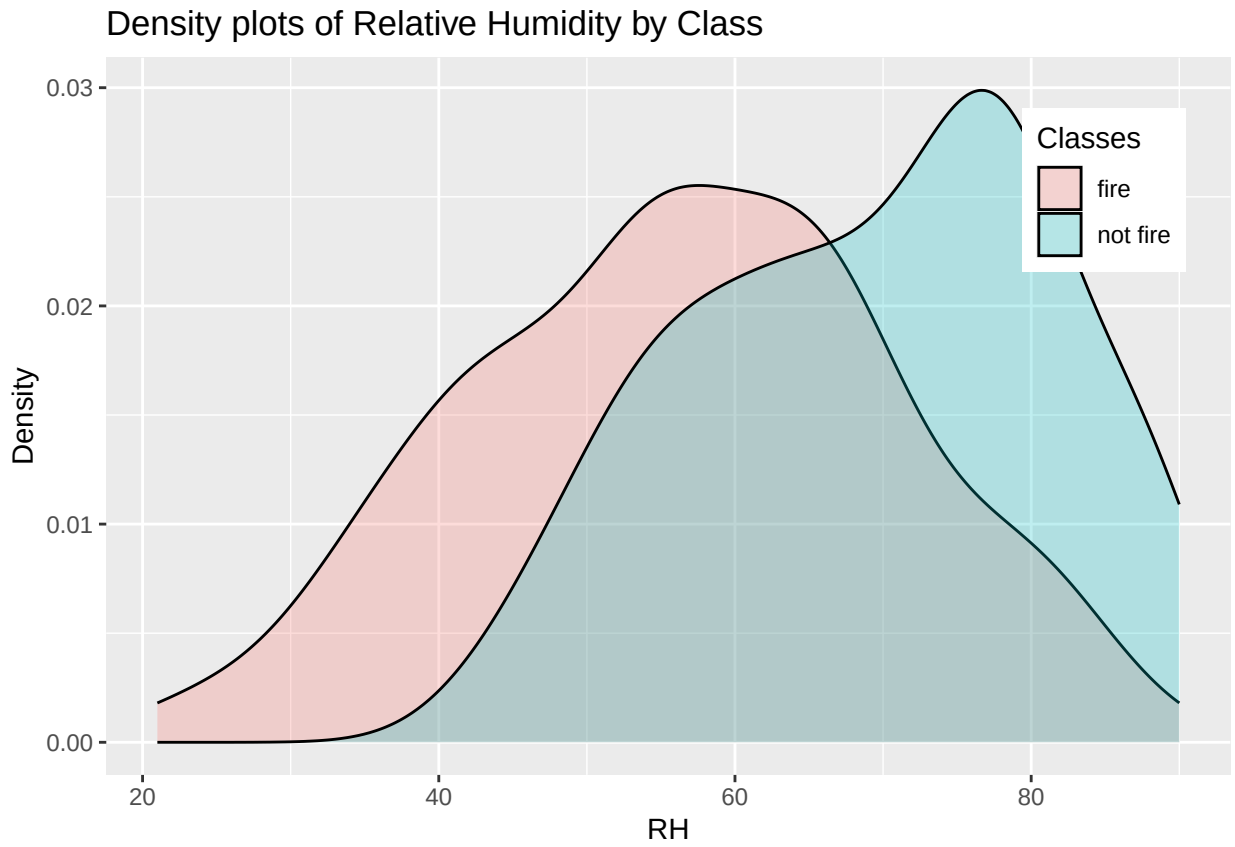
```
[1] 0
```

There is no "NA" values in the dataset.

Understanding the data through data visualisation

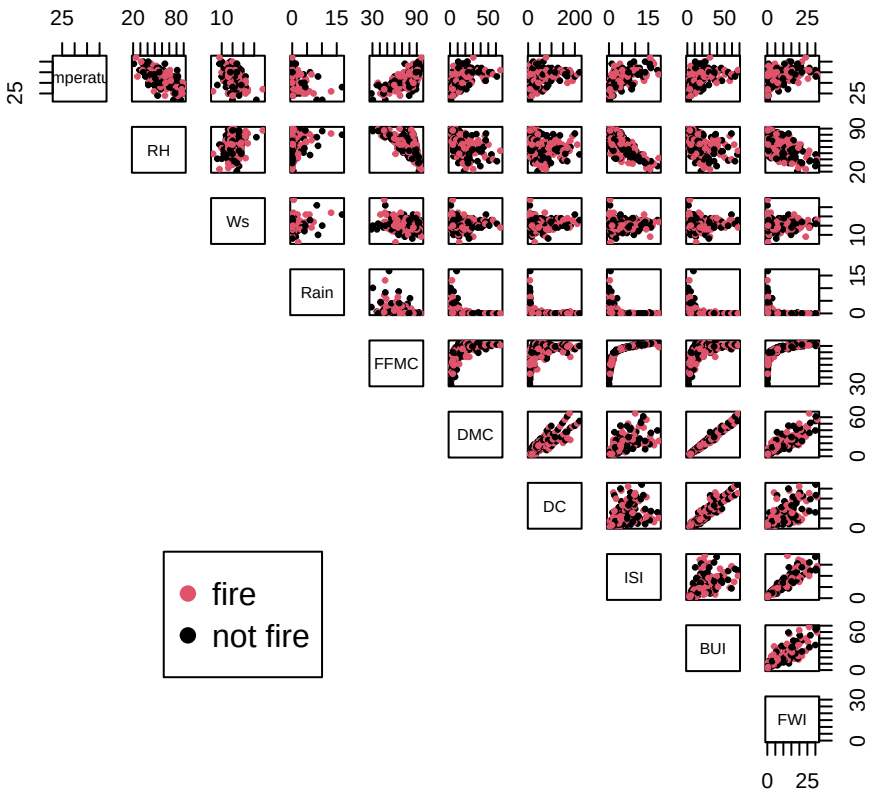

Histogram of Temperature
Frequency
0
10
20
30
40
50
25
30
35
40
Temperature


Density plots of Temperature by Class
Classes
fire
not fire
Density
0.10
0.05
0.00
25
30
35
40
Temperature



From the scatterplot matrix, most of the variables, visually, seems to have some correlation with one another.

```
[1] "fire"     "not fire"
```



1.2 (a) Should this data be scaled prior to running a PCA or not?

A key point to note, is that the covariance, $Cov(X_i, Y)$ is highly dependent on and highly affected by the scale of X_i and Y, where X_i is each of the independent variables in the dataset. There are reasons to believe that FFMC and DMC (for example) are of different scale, which may affect the results of Covariance.

The variables are of different units and have different ranges, hence scaling through standardisation is required prior to running PCA. However, when running PCA using the `prcomp` function, we can tell the function to standardise the values by specifying `scale = TRUE`. Hence, we do not need to manually scale the variables. However, for another function that conducts PCA without scaling, I will have to scale the data before running PCA. This is because variables such as `FWI` that is of significantly higher scale than `Temperature` may dominate measures of distance (sometimes due to unit differences), variables with lower scale would have reduced usefulness in the analysis.

```
forestf.cov <- cov(cbind(forestf[, 1:10]))
forestf.cov
```

```
            Temperature          RH          Ws        Rain        FFMC
```

```
Temperature   13.162670  -35.043482 -2.90194878  -2.3728497   35.222858
RH           -35.043482  219.874333 10.17380880   6.6048362 -137.215388
Ws            -2.901949   10.173809  7.90388736   0.9658861   -6.718952
Rain          -2.372850    6.604836  0.96588613   4.0128375  -15.634746
FFMC          35.222858 -137.215388 -6.71895215 -15.6347459  205.912204
DMC           21.837668  -75.071928 -0.02511989  -7.1690251  107.342972
DC            65.071727 -160.400449 10.60453015 -28.4564555  347.051372
ISI            9.101371  -42.298446  0.09964289  -2.8916881   44.113113
BUI           23.734918  -74.653741  1.25758596  -8.5465090  120.872510
FWI           15.297068  -64.096917  0.67707887  -4.8355020   73.791799
                    DMC         DC          ISI        BUI         FWI
Temperature  21.83766793   65.07173   9.10137061  23.734918  15.2970683
RH          -75.07192803 -160.40045 -42.29844574 -74.653741 -64.0969170
Ws           -0.02511989   10.60453   0.09964289   1.257586   0.6770789
Rain         -7.16902510  -28.45646  -2.89168809  -8.546509  -4.8355020
FFMC        107.34297198  347.05137  44.11311329 120.872510  73.7917995
DMC         153.58743428  517.42775  35.03222817 173.203205  80.7645303
DC          517.42774819 2272.00999 100.71852107 638.862525 262.2777461
ISI          35.03222817  100.71852  17.25765874  38.071147  28.5265515
BUI         173.20320460  638.86252  38.07114665 202.447968  90.8314934
FWI          80.76453032  262.27775  28.52655154  90.831493  55.3620481
```

Most covariances are non-zero, hence there is some correlation and dependence between each pair of variables.

1.3 (b) Consider the eigenvalues for the principal component analysis and answer the following (provide your relevant R output as well)

```
forestf.pca <- prcomp(~Temperature + RH + Ws + Rain + FFMC +
    DMC + DC + ISI + BUI + FWI, data = forestf, scale = TRUE)
# Eigenvalues
forestf.pca$sdev^2
```

```
 [1] 5.726855877 1.583287174 0.917148658 0.796524081 0.391380735 0.250211226
 [7] 0.222264816 0.092326109 0.016155238 0.003846086
```

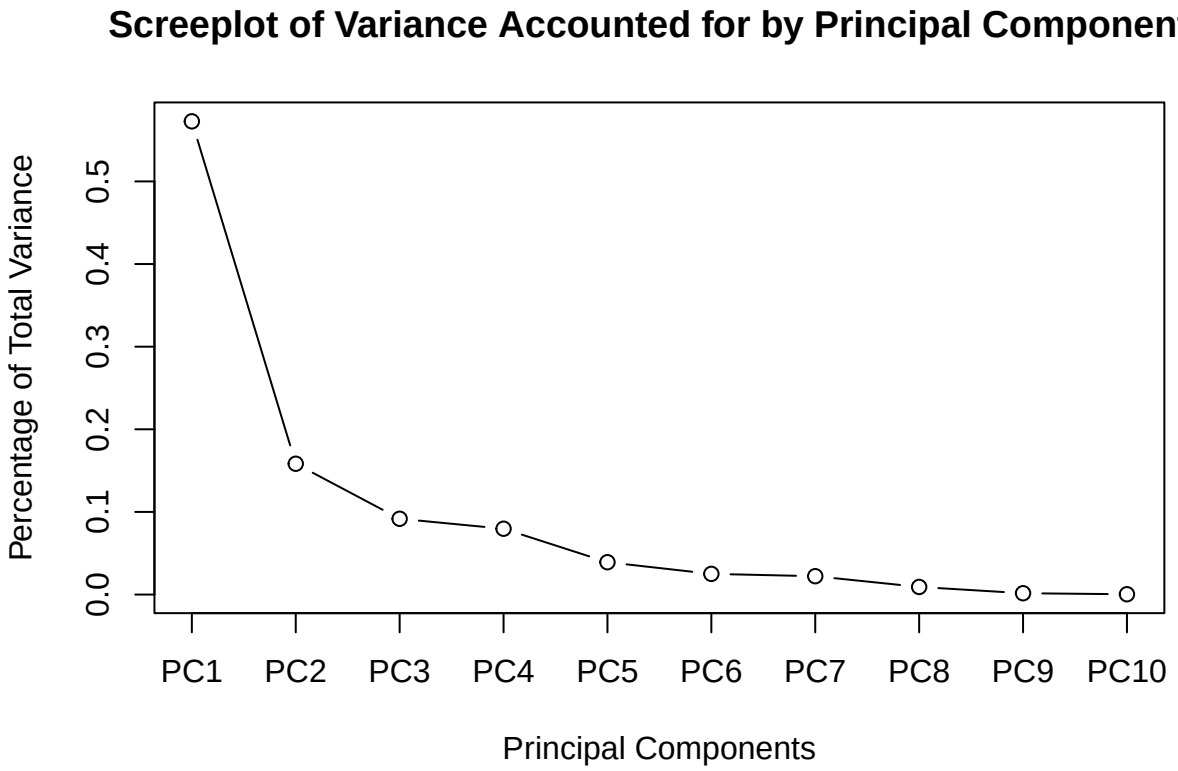


Figure 2: Scree plot of PCA

1.3.1 How many principal components would you select if using the "elbow" method?

Using visual analysis of the scree plot, the additional percentage of total variance (relative contribution) explained by PC4 as compared to PC3 is minimal. The percentage of total variance explained by PC1, PC2, and PC3 can be considered quite substantial.

Hence, through the "elbow" method, I would select three principal components, PC1, PC2 and PC3.

1.3.2 How many principal components would you select if attempting to account for 70% of total variation?

```
(forestf.pca$sdev^2)/sum(forestf.pca$sdev^2)
```

```
 [1] 0.5726855877 0.1583287174 0.0917148658 0.0796524081 0.0391380735
 [6] 0.0250211226 0.0222264816 0.0092326109 0.0016155238 0.0003846086
```

```
cumsum((forestf.pca$sdev^2)/sum(forestf.pca$sdev^2))
```

```
[1] 0.5726856 0.7310143 0.8227292 0.9023816 0.9415197 0.9665408 0.9887673
[8] 0.9979999 0.9996154 1.0000000
```

From the above output, I would select two Principal Components, PC1 and PC2. Both principal components will account for 73.10% (above 70%) of the total variation in the variables. PC1 will account for about 57.27% and PC2 will account for 15.83% of total variation.

1.4 (c) Produce a biplot of the first two principal components. What variable groupings load onto these components similarly?

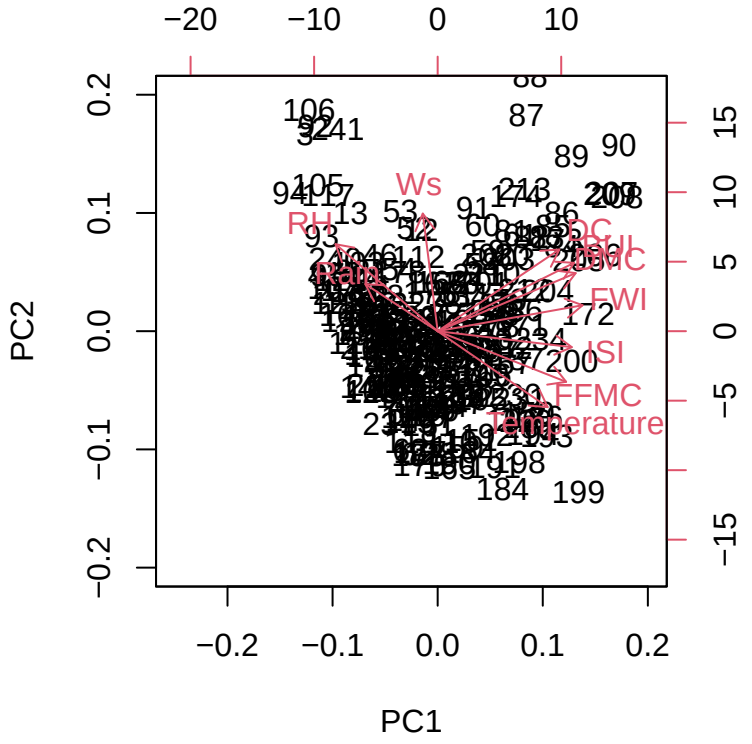


Figure 3: Biplot of the first two principal components

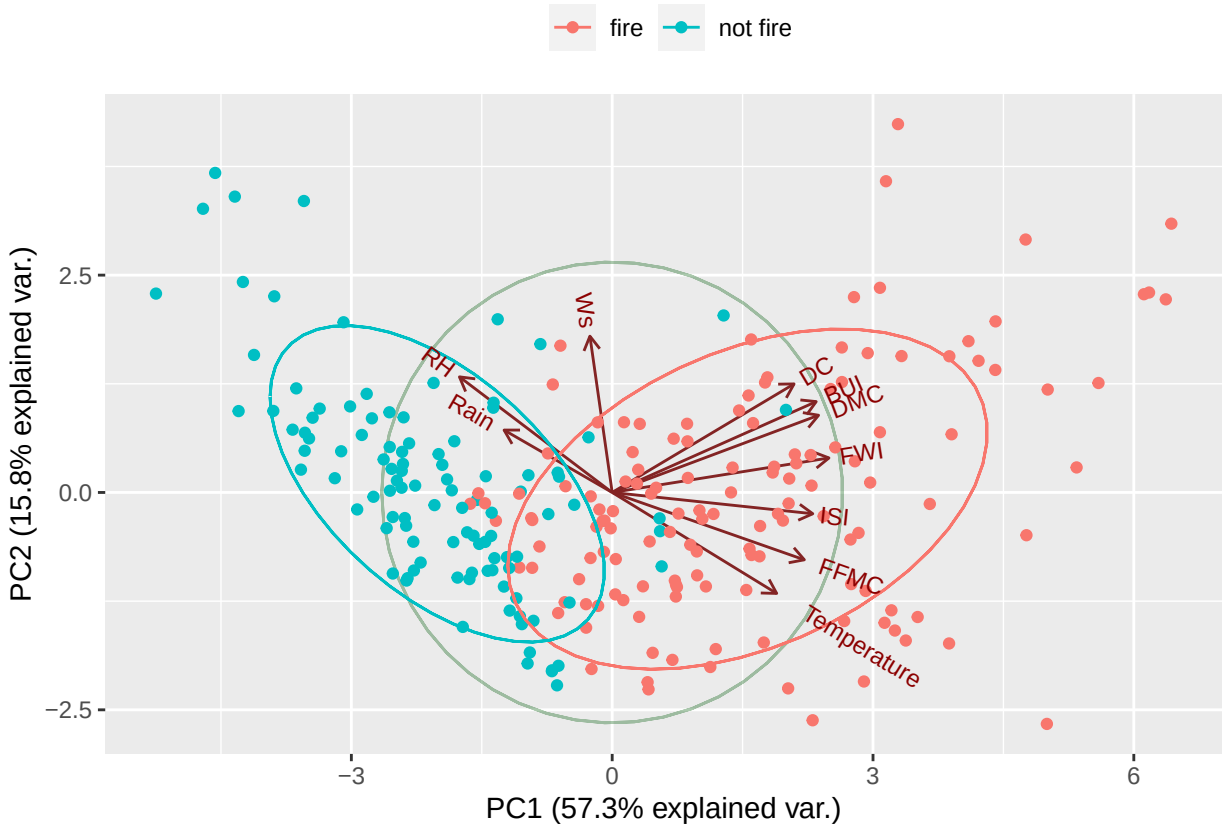


Figure 4: Alternate biplot using ggbiplot

The most recent biplot produces clearer results. Based on visual analyses of the bi-plots, the vector for `Ws` extend further from the origin in the vertical direction (corresponding to the second Principal Component, PC2) than the other vector, this means there is greater loading of `Ws` on PC2 but weak loading on PC1. `Temperature` and `Rain`, `Temperature` and `RH` are uncorrelated with each other.

`Temperature`, `RH`, `Ws`, `Rain`, `DC` are more loaded into PC2, because the vertical component of their vectors extend further than the horizontal component (representing PC1). The rest of the variables, `FFMC`, `DMC`, `ISI`, `BUI` and `FWI` are more strongly loaded on `PC1` than `PC2`. Similar results are reflected on the coefficients of the eigenvectors for PC1 and PC2.

1.5 (d) What are the percentage contributions (loadings) of the first five variables of PC1?

```
Temperature          RH          Ws        Rain        FFMC
 0.29834510 -0.27701600 -0.04037637 -0.19606535  0.34846690
```

```
sum(forestf.pca$rotation[1:5, "PC1"]^2)
```

```
[1] 0.3272487
```

First five variables	Coefficients of eigenvectors of PC1
Temperature	0.29834510
RH	-0.27701600
Ws	-0.04037637
Rain	-0.19606535
FFMC	0.34846690

The percentage contributions (loadings) for the first five variables (`Temperature`, `RH`, `Ws`, `Rain`, `FFMC`) of PC1 is 32.72%.

2 Question 2

We are interested to know if `BabyBirthWeight`, `NumberFeeds` and `MotherConcern` can aid in classifying `ProductionCategory`. In this case, `MotherID` is not an important variable.

```
Rows: 160
Columns: 5
$ MotherID           <dbl> 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 100~
$ ProductionCategory <chr> "Low", "Low", "Low", "Low", "Low", "Low", "Low", "L~
$ BabyBirthweight    <dbl> 3610, 3400, 3950, 3500, 3160, 3630, 2985, 3120, 248~
$ NumberFeeds        <dbl> 4, 10, 2, 4, 7, 6, 9, 5, 6, 7, 8, 9, 10, 12, 7, 6, ~
$ MotherConcern      <dbl> 83, 71, 86, 91, 87, 89, 80, 79, 77, 79, 76, 66, 83,~
```

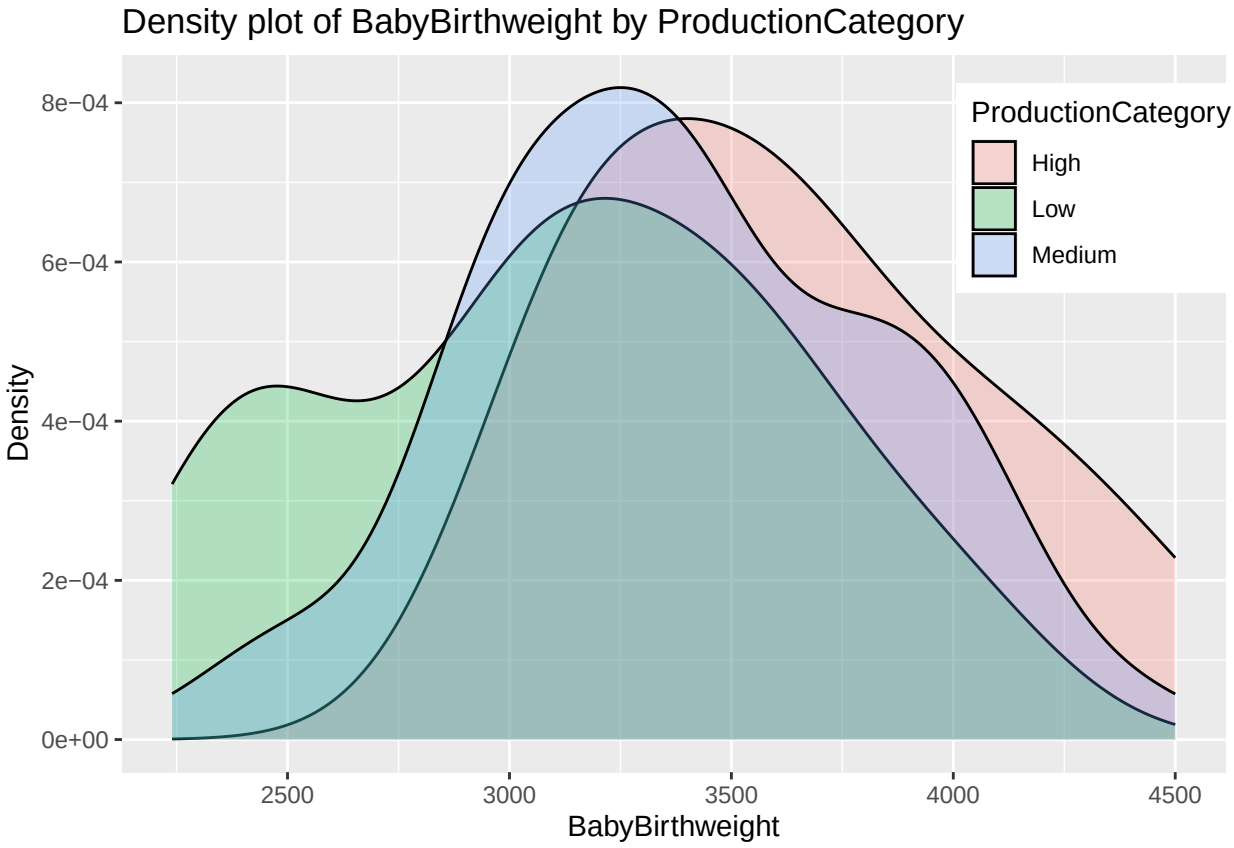
```
[1] 3
```

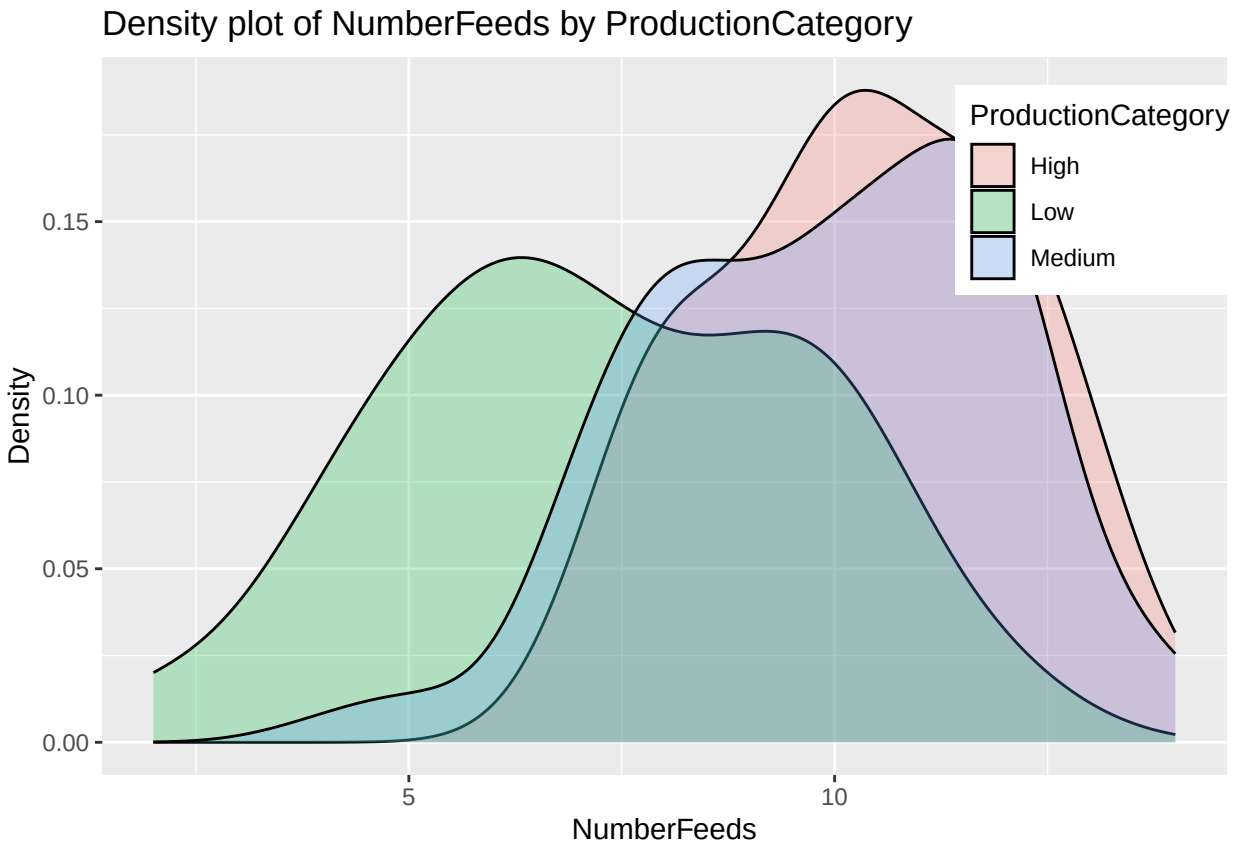
2.1 (a) Discuss the assumptions of linear discriminant analysis as they relate to this data set.

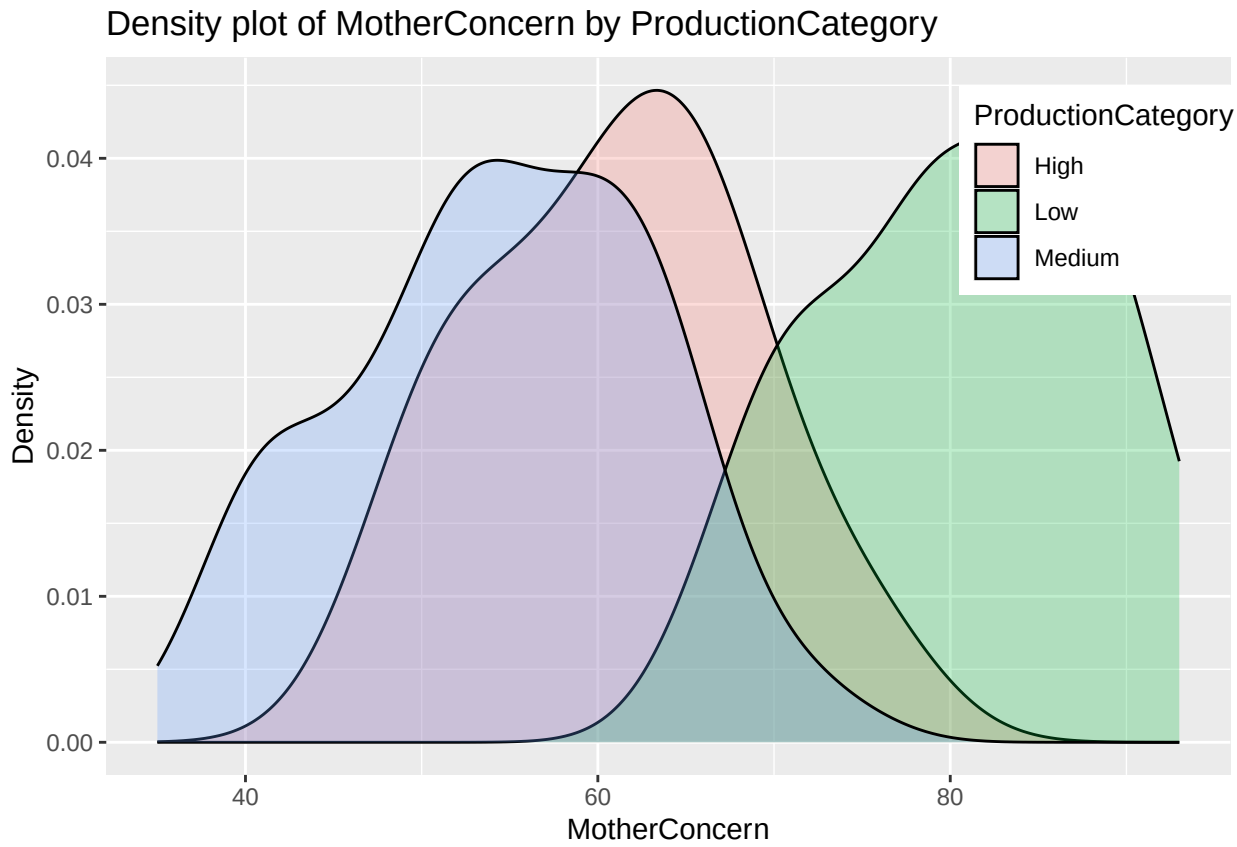
Assumptions of Linear Discriminant Analysis (**LDA**):

2.1.1 Normality

Explanatory variables $(X_1, X_2, ..., X_k)$ must follow a multivariate normal distribution for each value of the grouping variable Y.




Density plot of NumberFeeds by ProductionCategory
ProductionCategory
High
Low
Medium
Density
0.15
0.10
0.05
0.00
5
10
NumberFeeds



The assumption of linear discriminant analysis following a multivariate normal distribution is violated.

For `BabyBirthWeight`, the density plots for `ProductionCategory` that is `Low` and `Medium` shows a bimodal distribution. In particular, the bimodality of production categories `High` and `Low` for `NumberFeeds` and `BabyBirthweight` violates the assumption of normality.

There is no noticeable right-skewness in most, if not all of the sub-groups throughout the variables. Hence, log-transformation will not be applied for this analyses. Furthermore, it is assumed that multivariate normality assumption is not violated for the purposes of this assignment.

2.1.2 Homoscedasticity / equality of variances

The covariance matrices of $X_1, X_2, ..., X_k$ are equivalent for each value of the grouping variable Y. This means that the general shape/scatter and direction of points are relatively the same.

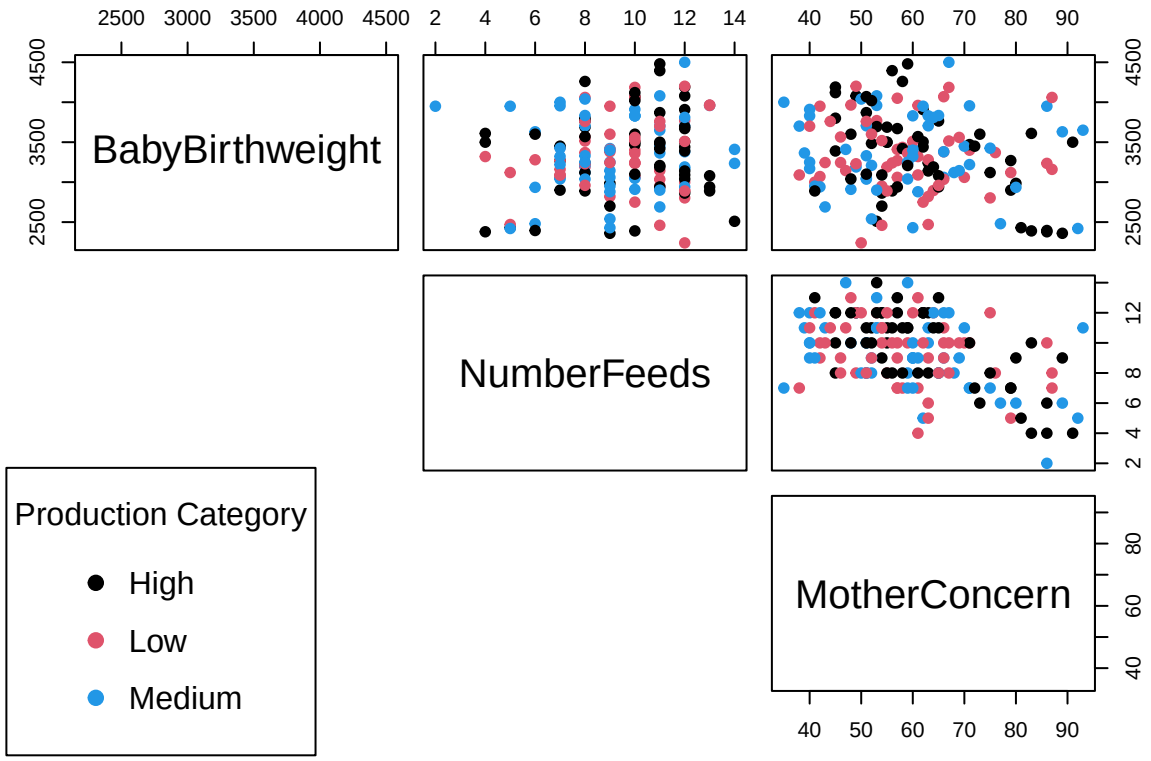


Figure 5: Scatterplot matrix between explanatory variables

From the scatterplot matrix, there is no significant distinction and many overlaps between points of different groups. From visual observation, homoscedasticity seems to have been violated for some groups, for example, for the plot `BabyBirthWeight` vs. `NumberFeeds`, the production category `High` data points seems to show an increasing variance with funnel-like shape.

Furthermore, for `NumberFeeds` vs. `MotherConcern`, `Medium` production category, seems to show similarly shows increasing variance, heteroscedasticity. The direction of points for each of the groups are unclear. Hence, it is likely that the assumption of homoscedasticity is violated.

2.1.3 Independence of observations

Experimental units, or, equivalently paired observations $(X_{1i}, X_{2i}, ..., X_{ki}, Y_i)$ for $i = 1, 2, ..., n$ are independent.

Independence of observations assumption is assumed not to be violated, each observation is an independent observation. There is no indication that this assumption is violated in the question. It is assumed that every observation "represents" and different mother, and no two or more observations come from the "same" mother.

2.2 (b) Using linear discriminant analysis, determine the hit rate when considering the variables baby birthweight, number of feeds and mother concern in trying to predict the outcome.

```
# Use LDA function to conduct LDA
milk.reduced.lda <- lda(ProductionCategory ~ BabyBirthweight +
    NumberFeeds + MotherConcern, data = milk.reduced)

# Get LDA values
milk.lda.values <- as.data.frame(predict(milk.reduced.lda)$x)

# To get LDA classes predict(milk.reduced.lda)

# Create confusion matrix (or classification matrix)
confusion.matrix.milk.LD1 <- table(predict(milk.reduced.lda,
    dimen = 1)$class, milk.reduced$ProductionCategory)

confusion.matrix.milk.LD2 <- table(predict(milk.reduced.lda,
    dimen = 2)$class, milk.reduced$ProductionCategory)

confusion.matrix.milk.LD1.LD2 <- table(predict(milk.reduced.lda)$class,
    milk.reduced$ProductionCategory)


# View the confusion matrix
confusion.matrix.milk.LD1
```

```
        High Low Medium
  High     0   0      0
  Low      1  23      3
  Medium  11   6    114
```

```
confusion.matrix.milk.LD2
```

```
        High Low Medium
  High     0   0      1
  Low      1  23      3
  Medium  11   6    113
```

```
confusion.matrix.milk.LD1.LD2
```

```
        High Low Medium
  High     0   0      1
  Low      1  23      3
  Medium  11   6    113
```

```
# Calculate the hit rate and misclassification rate
hit.rate.LD1 <- (0 + 23 + 114)/(1 + 23 + 3 + 11 + 6 + 114)
hit.rate.LD1
```

```
[1] 0.8670886
```

```
hit.rate.LD2 <- (0 + 23 + 113)/(1 + 1 + 23 + 3 + 11 + 6 + 113)
hit.rate.LD2
```

```
[1] 0.8607595
```

```
hit.rate.LD1.LD2 <- hit.rate.LD2

misclassification.rate.LD1 <- 1 - hit.rate.LD1
misclassification.rate.LD1
```

```
[1] 0.1329114
```

The highest hit rate when considering the possibly relevant variables is 0.8670886 or 0.867. This corresponds with using the first linear discriminant function, LD1.

2.3 (c) Using the group means, describe the three outcomes and how they typically differ.

```
milk.reduced.lda$means
```

```
       BabyBirthweight NumberFeeds MotherConcern
High          3622.083   10.333333      61.00000
Low           3127.483    7.275862      80.48276
Medium        3382.393    9.905983      54.47009
```

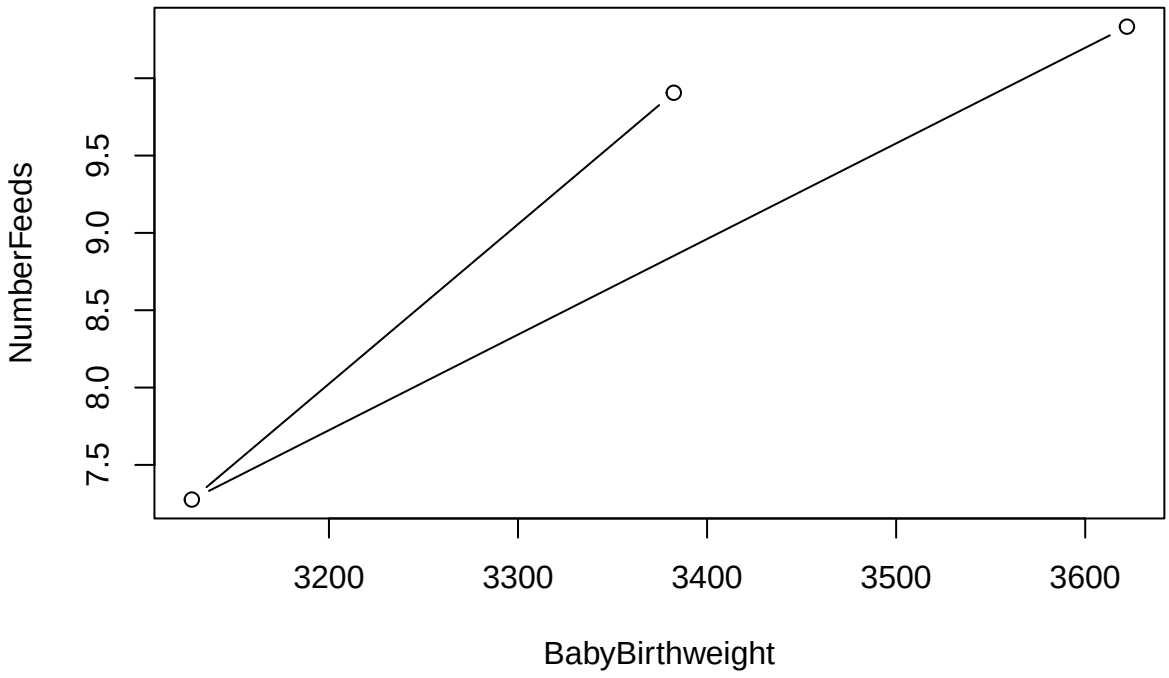
Group means shows the group center of gravity and shows the mean of each variable in each group.

The means of `BabyBirthweighht` is generally larger by a substantial magnitude compared to `MotherConcern` followed by `NumberFeeds`. It has been suggested in the group means that with low maternal production, there are low baby birth weight and lower number of feeds (on average per day) but mother's concern is the highest. This is because, the variables are all measured on different scales.

For `High` maternal production, there is the highest baby birth weight, highest number of feeds with medium mother's concern. This is surprising as the mother's concern for high maternal production is lower than for low maternal production but higher than for medium maternal production. Generally, mothers are more concerned when their maternal production is too high, and mothers are highly concerned when their maternal production is too low. It is interesting to note that although mothers are highly concerned, the number of feeds are low, possibly due to the low maternal production, that could not trigger more feeds or the baby's reluctance on accepting more feeds. This may have aggravated the mother concern measure.

Generally, the maternal production seems to correspond with the baby birth weight and the number of feeds per day on average.

Within each variable,



2.4 (d) How does this change if we say that the costs of misdiagnosing the high or low production mothers are 5 times that of medium production.

2.4.1 What are the new priors?

For the purposes of predictive LDA, LOOCV is applied.

```
# Define the cost of misclassification of high and low is 5
# times more than medium
cost <- c(5, 5, 1)

# Run LDA with LOOCV
milk.lda.cv <- lda(ProductionCategory ~ BabyBirthweight + NumberFeeds +
    MotherConcern, data = milk.reduced, CV = TRUE)

cm.milk.cv <- table(milk.lda.cv$class, milk.reduced$ProductionCategory)
cm.milk.cv
```

```
         High Low Medium
  High      0   0      1
  Low       1  23      3
  Medium   11   6    113
```

```
cm.milk.cv.hitrate <- (0 + 23 + 113)/(1 + 1 + 23 + 3 + 11 + 6 +
    113)
cm.milk.cv.hitrate
```

```
[1] 0.8607595
```

```
# Calculate loss based on costs and misclassifications
loss <- 0

for (i in 1:ncol(cm.milk.cv)) {
    loss <- loss + cost[i] * sum(cm.milk.cv[-i, i])
}
loss
```

```
[1] 94
```

```
# Compute old prior probabilities
old.prior <-
↪  table(milk.reduced$ProductionCategory)/sum(table(milk.reduced$ProductionCategory))
old.prior
```

```
      High        Low     Medium
0.07594937 0.18354430 0.74050633
```

```
# Create new priors based on costs
new.prior <- cost * old.prior/sum(cost * old.prior)
new.prior
```

```
     High       Low    Medium
0.1863354 0.4503106 0.3633540
```

The new prior probabilities are as follows:

High	Low	Medium
0.1863354	0.4503106	0.3633540

2.4.2 What is the new hit rate?

```
# Re-run LDA with new prior
milk.lda.cv.newprior <- lda(ProductionCategory ~ BabyBirthweight +
    NumberFeeds + MotherConcern, data = milk.reduced, CV = TRUE,
    prior = new.prior)

cm.lda.cv.newprior <- table(milk.lda.cv.newprior$class,
↪  milk.reduced$ProductionCategory)

cm.lda.cv.newprior
```

```
        High Low Medium
  High     3   0     12
  Low      1  28      6
  Medium   8   1     99
```

```
cm.lda.cv.newprior.hitrate <- (3 + 28 + 99)/(3 + 12 + 1 + 28 +
    6 + 8 + 1 + 99)
cm.lda.cv.newprior.hitrate
```

```
[1] 0.8227848
```

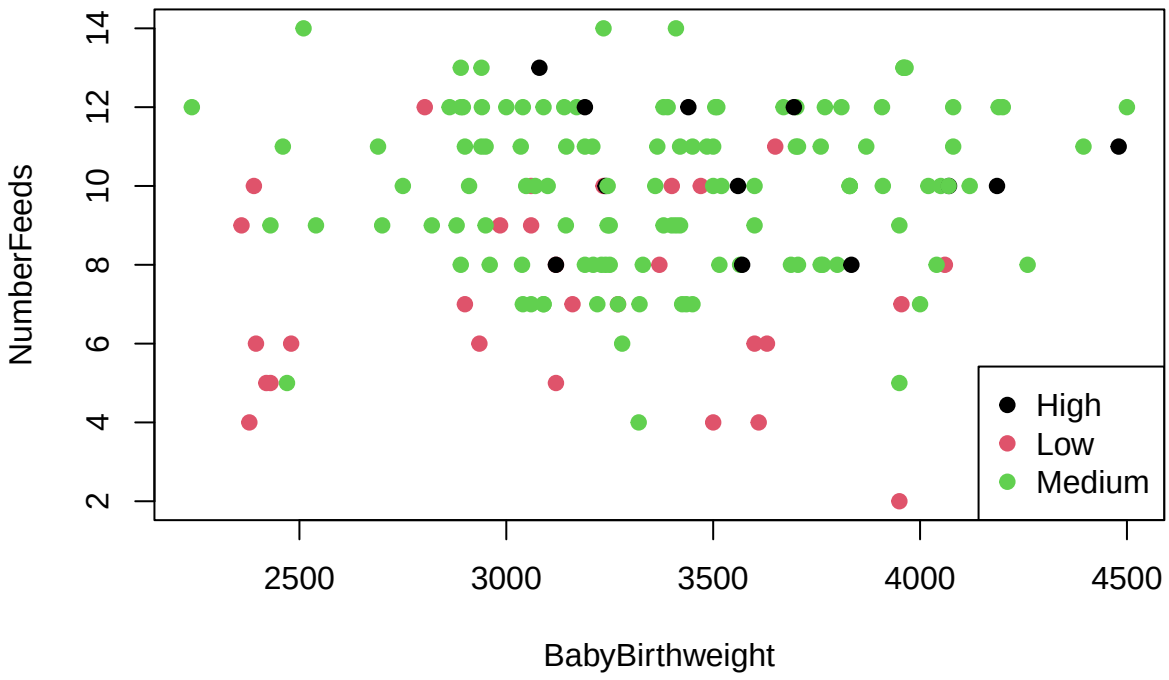
```
# Calculate loss based on costs and misclassification
loss <- 0

for (i in 1:ncol(cm.lda.cv.newprior)) {
    loss <- loss + cost[i] * sum(cm.lda.cv.newprior[-i, i])
}
loss
```

```
[1] 68
```

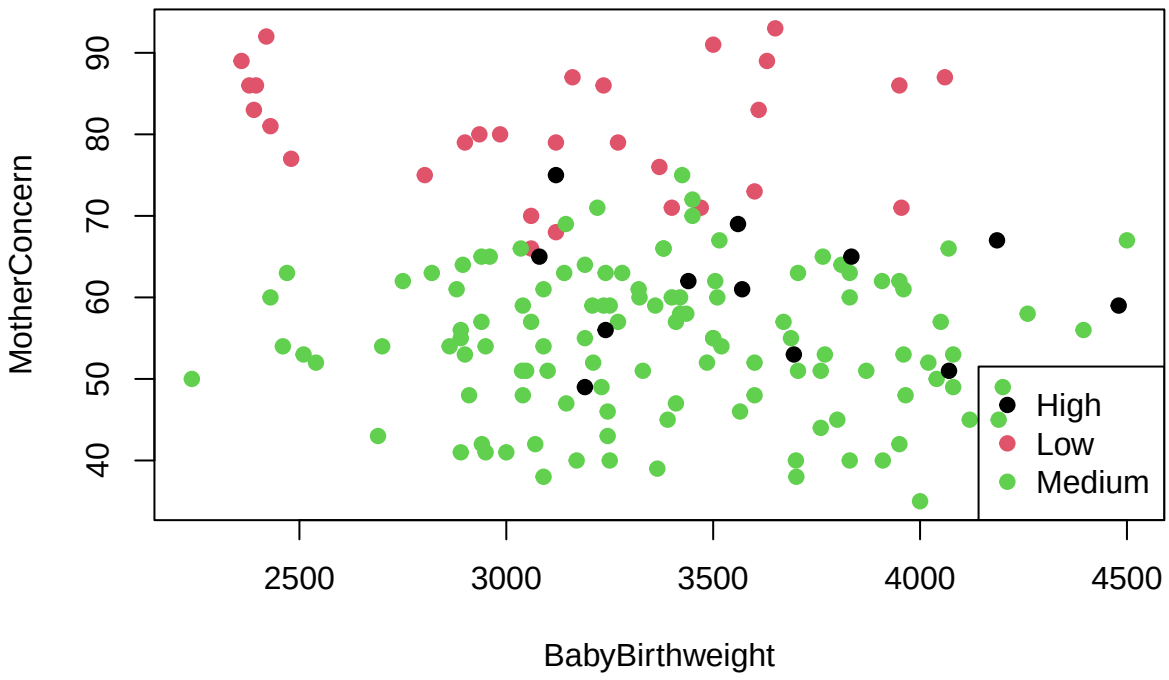
From the confusion matrix above, the new hit rate is 0.8227848 or 0.823. As expected the prior probabilities for `High` and `Low` are higher than before, while for `Medium`, it has reduced. The loss has also been reduced.

2.5 (e) Is linear discriminant analysis effective in this context? Provide at least one visualisation to support your answer.

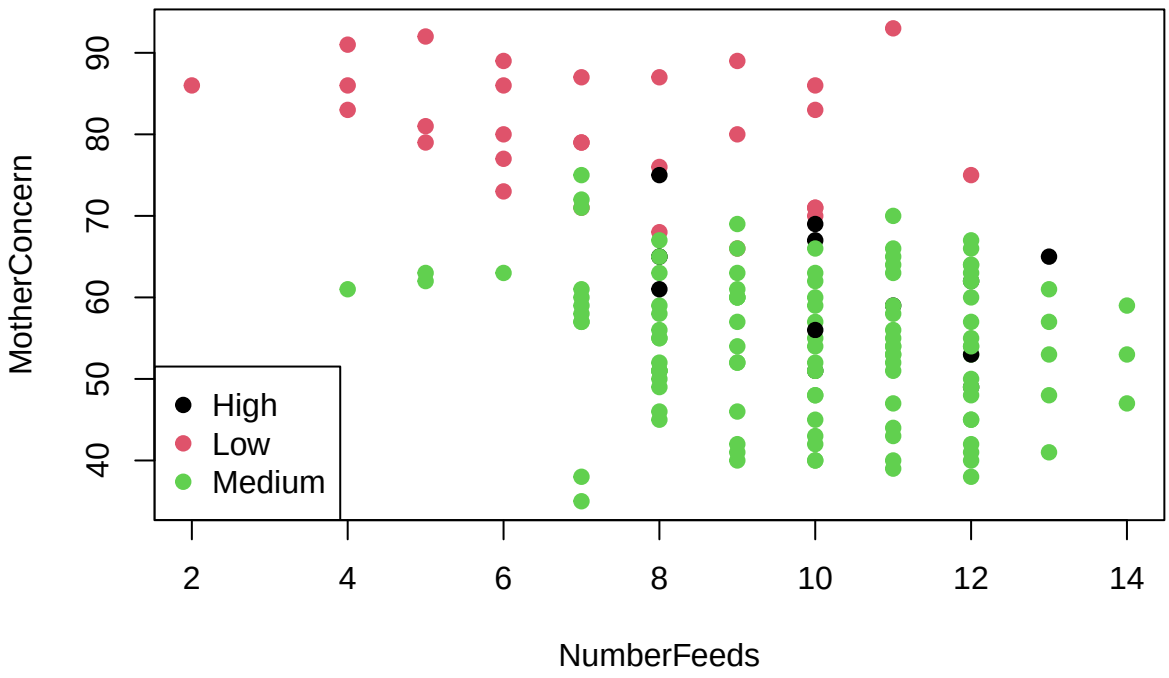
```
plot(milk.reduced$BabyBirthweight, milk.reduced$NumberFeeds,
    col = milk.reduced$ProductionCategory, xlab = "BabyBirthweight",
    ylab = "NumberFeeds", pch = 19, cex = 1)
legend(x = "bottomright", legend = levels(milk.reduced$ProductionCategory),
    pch = 19, col = 1:3)
```

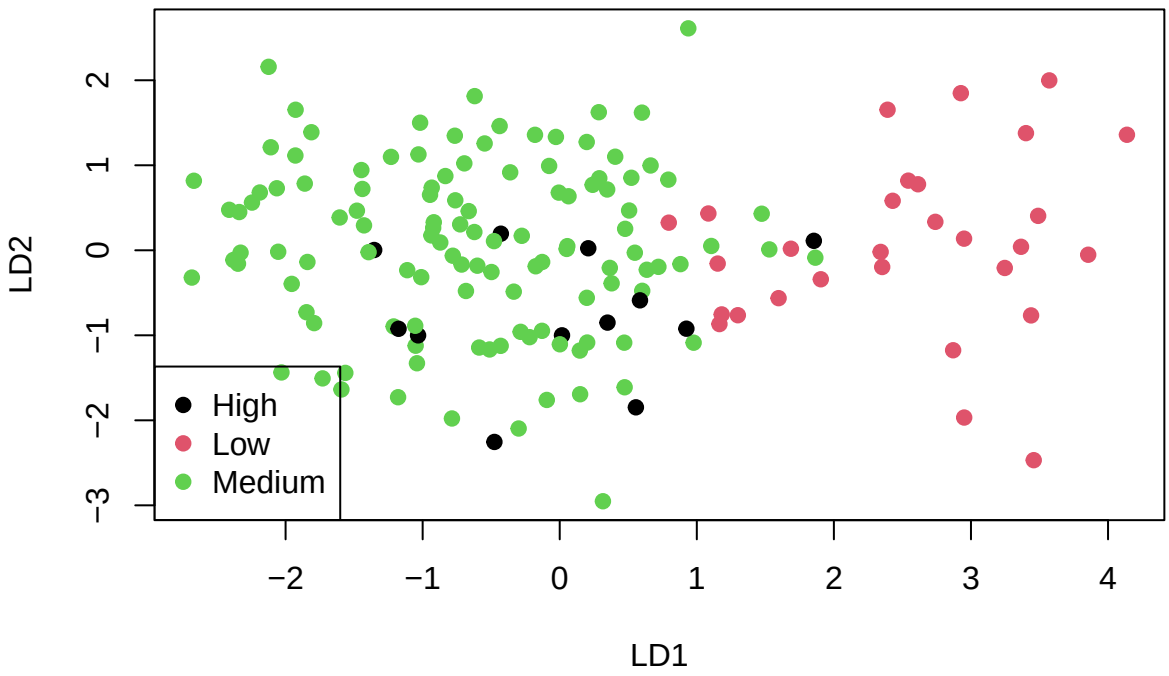


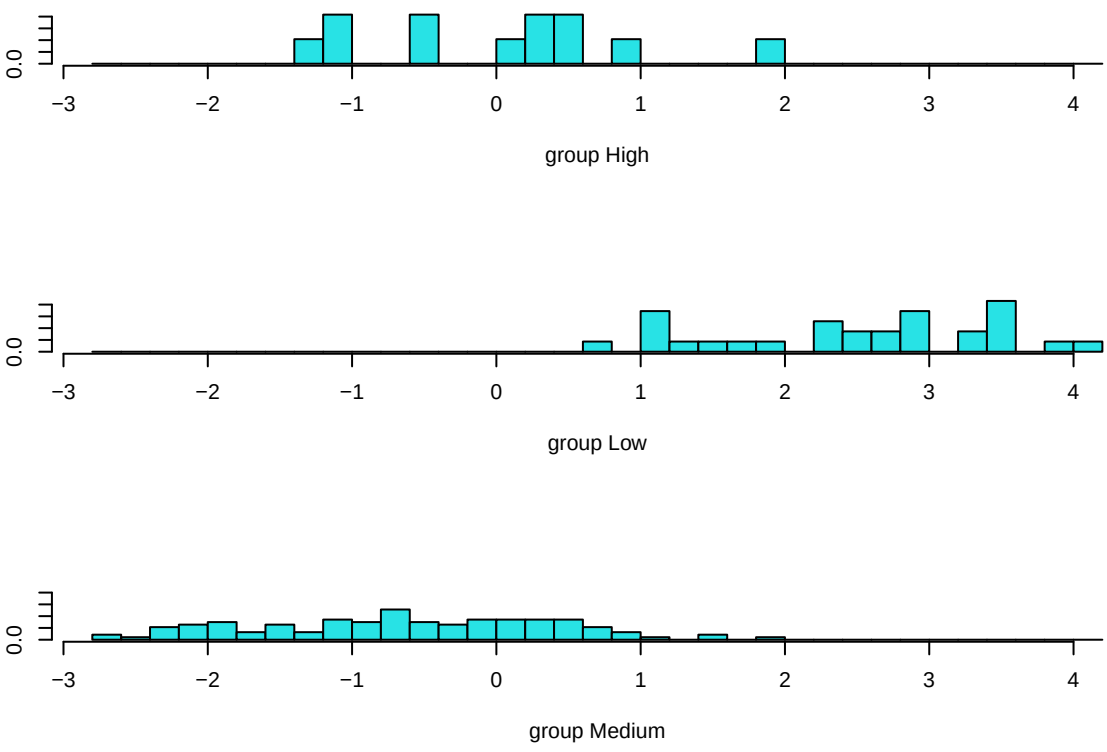

```
plot(milk.reduced$BabyBirthweight, milk.reduced$MotherConcern,
    col = milk.reduced$ProductionCategory, xlab = "BabyBirthweight",
    ylab = "MotherConcern", pch = 19, cex = 1)
legend(x = "bottomright", legend = levels(milk.reduced$ProductionCategory),
    pch = 19, col = 1:3)
```

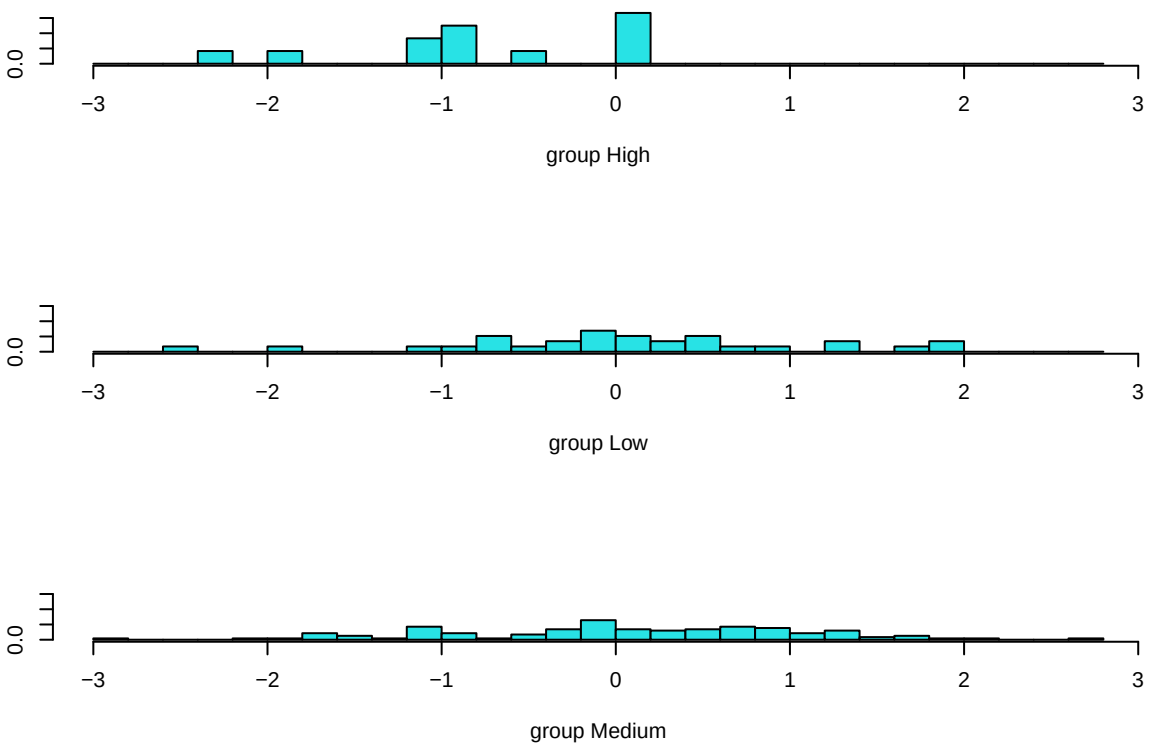



```
plot(milk.reduced$NumberFeeds, milk.reduced$MotherConcern, col =
↪  milk.reduced$ProductionCategory,
    xlab = "NumberFeeds", ylab = "MotherConcern", pch = 19, cex = 1)
legend(x = "bottomleft", legend = levels(milk.reduced$ProductionCategory),
    pch = 19, col = 1:3)
```


MotherConcern
40
50
60
70
80
90
2
4
6
8
10
12
14
NumberFeeds
High
Low
Medium


LD2
LD1
High
Low
Medium
−3
−2
−1
0
1
2
−2
−1
0
1
2
3
4


0.0
−3
−2
−1
0
1
2
3
4
group High
0.0
−3
−2
−1
0
1
2
3
4
group Low
0.0
−3
−2
−1
0
1
2
3
4
group Medium



From the three scatterplots, there is substantial overlap between the distributions of values for each of the three production levels, this suggests that LDA may struggle to produce splits/classifications that clearly distinguish groups.

From the plot of second discriminant function values against the first discriminant function values (LD2 vs. LD1), there is some clustering (not very distinct) for `Medium` and `Low` groups. A linear separating line can be drawn in between with some allowance for misclassification. Unfortunately, there is no clear/distinct clustering for `High` group. LDA is more effective at revealing between `Low` and `Medium` groupings, but not so for `High`.

From the stacked histogram figures, the first linear discriminant function seems better at discriminating between the `Low` and `Medium` group, however, it seems that there is some overlap between `High` and `Low`, and `High` and `Medium` groups. There is too much alignment and overlap for the stacked histogram between `Low` and `Medium` group in the second discriminant function. This suggests that the second discriminant function is poor at creating separation for the grouping variable.

Linear discriminant analysis can be considered effective if there are only `Low` and `Medium` groups in the dataset. However, in unseen and unlabelled datasets, and the goal is for prediction, there could be `High` maternal production that may be incorrectly classified to `Low` or `Medium` using LDA.

In conclusion, LDA is not very effective, especially for unlabelled data for prediction, especially when we are unsure if it contains data that is of `High` maternal production.

3 Question 3

For supervised learning problems, there are predictor measurement(s) x_i and at least a response variable, y_i. A model is fitted to relate the response to its predictors. The goal is to better understand the relationship between the response and predictors (inference) and predicting the response of future observations (prediction) (James et al. 2013).

In a *unsupervised* machine learning problem, there is no response variable to **supervise** our analysis. Given no associated response variable, y_i to a set of measurements, x_i, it is not possible to utilise supervised machine learning techniques, which expect a response variable to be fed to train a model. Hence, the presence of the response variable used in the machine learning analysis is a key difference between supervised and unsupervised learning problems.

The goals of supervised and unsupervised learning problems are different. In a supervised learning problem, our goal is either inference or for prediction. This is in contrast to the goal of unsupervised learning, to understand relationships between variables or between observations (James et al. 2013). For example, *cluster analysis* such as K-means clustering algorithm is a unsupervised machine learning technique. The aim is to determine with only the response variables x_i whether the observations can be clustered into distinct groups. The algorithm tries to group the data points into "k" clusters without knowledge of the grouping labels. Without a grouping variable Y, the algorithm attempts to assign observations to clusters by minimising the total distance from points to cluster centroids. Because, the algorithm requires cycling through each observation, if the dataset is large by length, it may be more computationally intensive compared to KNN.

This is different from K-nearest neighbours (KNN) classification algorithm. KNN is a supervised machine learning technique, as the labelled points can be supplied to the algorithm which will be utilised to learn how to label other points. For new points, the algorithm may calculate the euclidean distance to the labelled points and determine the shortest distance to neighbouring clusters and tries to find the optimal number of nearest neighbours for correct classification of new observations. In this case, the classification of other points are known because the data points are labelled with the response variable.

Another limitation to unsupervised learning, particularly K-means clustering, is that the cluster assignment of observations are not guaranteed to be unique as the same WSS could be achieved with different cluster assignments, unlike for KNN (objective distance calculation). Hence, we may not possibly know the global optimal for K-means clustering classification.

References

James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2013. *An Introduction to Statistical Learning*. Vol. 112. Springer.