Investigation of IDA-SE with 3-dimensional Dataset Intelligent Data Analysis (extended) report

Shubhangi Birajdar (Student ID: 2032649)

School of Computer Science, University of Birmingham

Abstract

The motive of these observations was to use dimension-reducing methods such as Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) to examine magnetometer data on Galileo MAG.

This is official NASA's dataset of Galileo MAG high-resolution magnetic field. The Galileo Orbiter supports a complement of fields and particles instruments mapped out to supply data required from magnetometer. The dataset that I have worked on to examine variations of data from Solar Equatorial (ISE) coordinates of Ida astroid.

Keywords: **PCA, LDA, dataset, Galileo, MAG**

1. Introduction

The purpose behind this dataset is to examine values from 3 co-ordinates (X, Y and Z) of ISE. I followed J. W. Belcher's example and he researched on magnetosphere of Jupiter. In dataset, there are 4 sections which includes labels of "TIME", "X COMP OF B IN IDA-SE COORDS", " Y COMP OF B IN IDA-SE COORDS" and "Z COMP OF B IN IDA-SE COORDS"[1].

According to Space Science Reviews, to display the strategy, consider that the data from the space-probe are achieved in a rotating coordinate system (1, 2, 3) so that the measured elements of the magnetic field are (B_1, B_2, B_3). The data are represented into an inertial coordinate system (X, Y, Z) to obtain (B_X, B_Y, B_Z)[2].

2. Results

The RStudio environment is an IDE for R which includes console, syntax-highlighting editor that runs direct expressions of code execution. It has special tools that gives special features like plotting, summary of data, debugging and it has in-built libraries like pca3d, MASS, ggplot2 to enhance output. It's easy to import any kind of data and examine it.

2.1. Complete Data Set:

In each column, there are 90 sets. From "TIME" column, 90 sets are divided into 3 groups (each group has 30 sets) with same date(yyyy-mm-dd) but different time(hh). The title "TIME" started from 15.00 and ends with 17.59 (which indicated as 1993-08-28/15, 1993-08-28/16 & 1993-08-28/17). Each row of "TIME" differs by 2 minutes and capture measurements from all axes.

This dataset has meant to pre-process the data for calculating sdv, mean, co-variance , probabilities and many more to get results and methods like PCA, LDA to generate projection and plot data respectively.

2.2. Pre-processing:

In RStudio, it was necessary to put finite data from dataset to get accurate result and plot it to check how dimensions looks like. To create projections, **pca3d** output was created from raw data that can be seen in Figure 1.

It displays position of data from axes and somehow, X COMP data found in Z COMP region. These attributes are expressed as PC 1, PC 2 and PC 3 which define as principle components of 3-dimensional architecture. In pca3d library, it is essential to convert dataset into matrix using **prcomp** or **princomp** functions uses singular value decomposition(SVD) and spectral decomposition approach respectively.

In second, Figure 2, it shows 2-dimensional architecture PCA using **pca2d** function that displays the representation of 3 axes according to time.

2.3. Labelling:

The original dataset is actual big and it was hard to apply PCA and LDA for code execution. So dataset has replaced with finite data with 90 rows and

same no. of columns. In each column, there are 90 records. From "TIME" column, 90 data are divided into 3 sections (each section has 30 sets) got from same date(yyyy-mm-dd) but different time(hh). The column "TIME" started from 15.00 and ends with 17.59 (which indicated as 1993-08-28/15, 1993-08-28/16 & 1993-08-28/17). Each row of "TIME" differs by 2 minutes and capture measurements from all axes. Therefore, it was necessary to apply a new labelling structure and filter it with countable rows to achieve projections.

3. Principle Component Analysis

Principle Component Analysis (PCA) is effective technique to manipulate datasets and using RStudio, it allows some special features to add customization for plotting dataset. By using **pca3d** package in R, **eigen** function helps to convey accurate eigenvalues and eigenvectors of co-variance matrix which are executed. Let's understand the concept that how PCA and **eigen** function works.

- Step 1: Calculate co-variance matrix of IDA-SE dataset.
- Step 2: To get result, we need co-variance matrix of given dataset which has performed procedure of function **prcomp** or **princomp** & defined group by using **factor/as.factor** function to set single column which help to get graphical representation of output by applying **pca3d** function.

Summary of PCA (IDA–SE COORDS):
Importance of components:

	PC1	PC2	PC3
Standard deviation	1.4420	0.9576	0.06170
Proportion of Variance	0.6931	0.3056	0.00127
Cumulative Proportion	0.6931	0.9987	1.00000

- Step 3: Use function **eigen** to get eigenvalue and eigenvector to compute co-variance matrix. Below result is summary of Eigenvalue, Eigenvector and co-variance of IDA-SE dataset.

```
> Eigenvalues
[1] 2.079287764 0.916905689 0.003806548

> Eigenvectors
      [,1]      [,2]           [,3]
[1,]    1  0.000000e+00  0.000000e+00
[2,]    0 -1.000000e+00 -1.110223e-16
[3,]    0 -1.110223e-16  1.000000e+00

> cov(pca\$x)[1:3, 1:3]
          PC1           PC2           PC3
PC1  2.079288e+00 -1.260836e-16 1.055024e-16
PC2 -1.260836e-16  9.169057e-01 4.507713e-17
PC3  1.055024e-16  4.507713e-17 3.806548e-03
```

- Step 4: As we can be seen that that the variance of **cov(pca$x)** is equal to Eigenvalue and co-variances value must be 0 (Principal Components need to be uncorrelated).

- Step 5: Measure proportion of variation which has determined by given components.

```
> print(round(Eigenvalues/sum(Eigenvalues) * 100, digits = 2))
[1] 69.31 30.56  0.13

> round(cumsum(Eigenvalues)/sum(Eigenvalues) * 100, digits = 2)
[1]  69.31  99.87 100.00
```

From this step, it actually defines that
round(Eigenvalues[1]/sum(Eigenvalues)*100, digits=2) prints 69.31 and rest of values depicts total variation in the data.
This recommend that the productive dimension of the space of yield curves may be three and more of the yield curves from data set can be explained by a linear combination of all values, besides the reality, relative error is very small[3].
Further on Figure 3 and Figure 4, applying **screeplot(Eigenvalues)** captures result with representation of graphics using types “biplot” and “barplot” respectively.

To see how implications of output works, below Figure 5, explains the differentiation of each axes using **pca$rotation** function in pca3d. All of these output has computed in RStudio by using pca which is covariance matrix of actual database. This element returns the matrix of variable loadings (columns are eigenvectors). In rotation matrix, each column consists loading vector of given principle component. By simply applying **pca$rotation[, 1]**, **pca$rotation[, 2]**, **pca$rotation[, 3]** statement, X COMP has dramatic biplot line while Y COMP ratio raised slowly till end point and at the end, Z COMP picked highest value and continued with same level as mentioned in Figure 5.

4. Linear Discriminant Analysis

Linear Discriminant Analysis is another analysis technique of machine learning and Dimensionality reduction technique which reduce dimensions by taking out dependent features by converting it from high dimensions to low dimensions. To compute this data execution by using R, another package called MASS applying function **lda()**. The main goal while choosing this method was to show directions that maximizes class separation and utilize directions to predict single class. Let's find out how function **lda()** works,

- Step 1: Apply MASS library to compute **lda()** function using IDA-SE dataset.
- Step 2: Transform matrix to data frame and take group of 1st column of dataset to perform **lda()** function.
- Step 3: Perform **lda()** by applying **lda(gr ~, pca.lda** syntax to get resultant summary of LDA method.

  As it has shown below,

  a) Prior probabilities of groups: In this section, it represents proportion of training observations in individual groups. As can be seen that 33% training has examined in 1st group.
  b) Group means: It displays mean of variables with each group of IDA-SE dataset.
  c) Coefficients of linear discriminants: It calculates linear integration of predicted variable for LDA decision rule. For example,

$+15.5 * \text{X COMP} - 1.18 * \text{Y COMP} - 16.6 * \text{Z COMP}$

d) Proportion of trace: IDA-SE dataset separated by data with proportion with 99.97% and 0.03% of LD1 and LD2 resp.

```
                    LDA of IDA-SE COORDS:

Prior probabilities of groups:
    1993-08-28/15     1993-08-28/16     1993-08-28/17
        0.3333333         0.3333333         0.3333333

Group means:
                    X COMP      Y COMP       Z COMP
1993-08-28/15   -0.9581578   0.46880741 -0.002448176
1993-08-28/16   -1.0624466  -0.39401265  0.006335906
1993-08-28/17    2.0206043  -0.07479476 -0.003887730

Coefficients of linear discriminants:
             LD1         LD2
X COMP   15.57512  0.03179052
Y COMP   -1.18847  1.09970355
Z COMP -16.65302  -2.66321482

Proportion of trace:
    LD1    LD2
 0.9997 0.0003
```

- Step 4: To see plot, library **ggplot2** use function **ggplot()** to create plots of LD such as LD1 and LD2 respectively. Given Figure 6 shows graphical representation of LDA of IDA-SE dataset.

- Step 5: After finding LDA of IDA-SE dataset, we can create prediction value which is part of LDA and reserve it in an object using **predict()** function. This function generate measurements from dataset. The size of predicted value will be correlate with length of processing data. Figure 7 and Figure 8 are histogram of LDA of IDA-SE dataset using function **ldahist()**.

5. Conclusion:

In many fields specially in Intelligent Data Analysis domain, somehow it could be important that how much size of dataset with no. of sets that goes in processing and detect result using any methods of analysis. The main focus is to see how LDA and PCA gives certain output that shows different projections and dimensions. It seems that both methods of analysis are useful to get accurate output and perform code execution to run dataset. When IDA-SE dataset has gone through pre-processing with PCA and their functions, PCA ignores existed class labels. While LDA aims to discover a characteristics and separates each group differently. Above Figure 2 illustrates how PCA reducing number of samples in each group. In my opinion, PCA method is top and it has better performance than LDA . The reason why PCA outperforms LDA, is the number of training data per group is small. It relies on what IDE environment attends functions of PCA and LDA to give best result.

References

[1] Belcher, J., *The low-energy plasma in the Jovian magnetosphere. Physics of the Jovian Magnetosphere*, pp.68-105.

[2] Kivelson, M., Khurana, K., Means, J., Russell, C. and Snare, R. (1992). *The Galileo magnetic field investigation.*

[3] Anon. (2009). *Principal Component Analysis using R.*

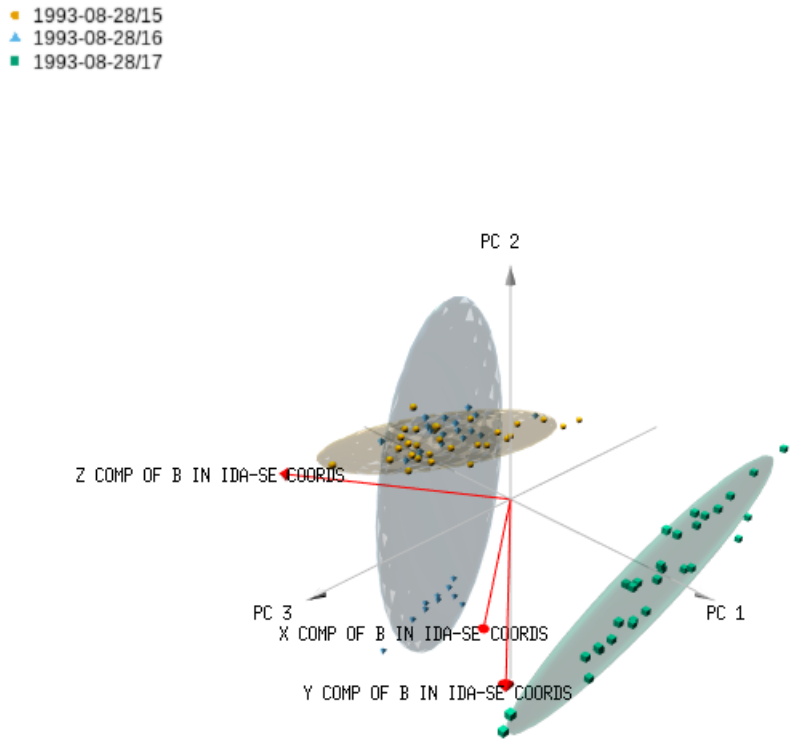


Figure 1: A 3-dimensional architecture PCA of IDA-SE data using **pca3d** function

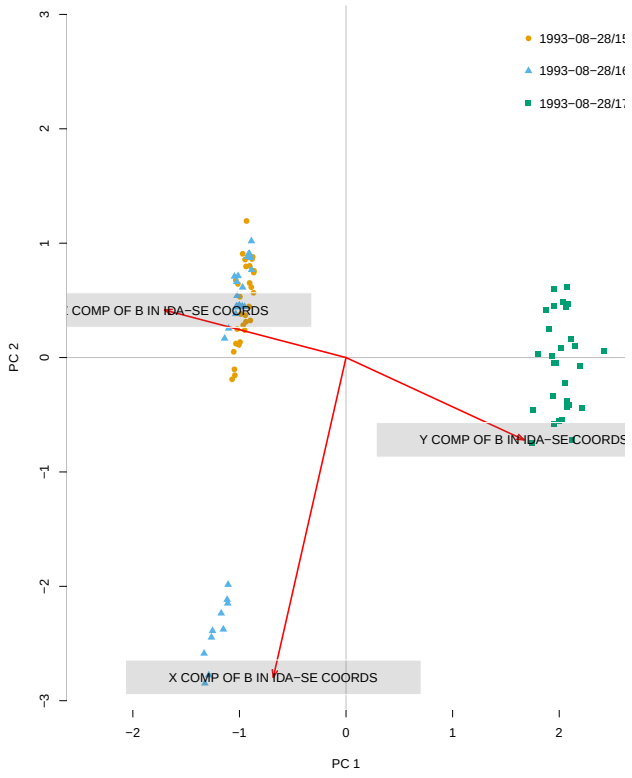


Figure 2: A 3-dimensional architecture PCA of IDA-SE data using **pca2d** function

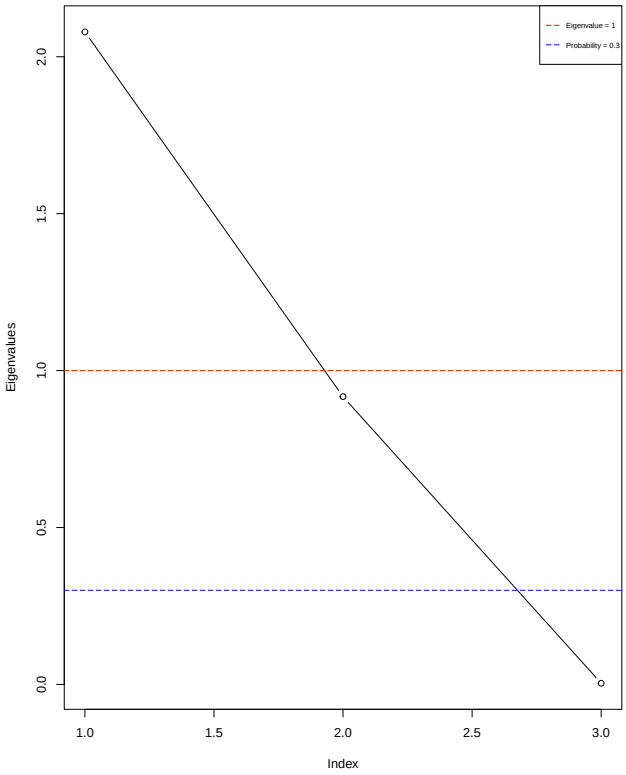


Figure 3: Plot of Eigenvalue of IDA-SE data with type “biplot”

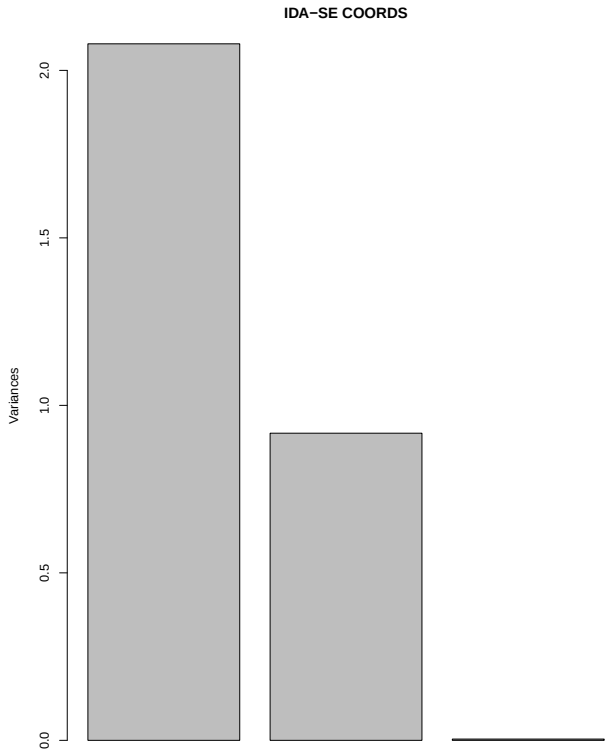
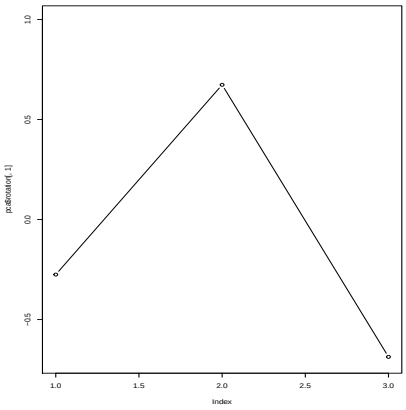
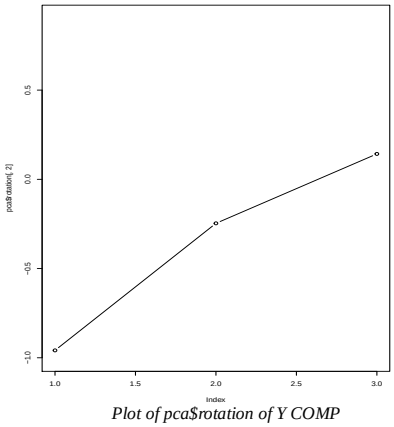


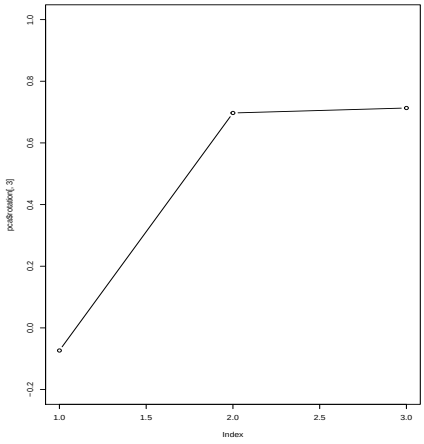
Figure 4: Plot of Eigenvalue of IDA-SE data with type “barplot”



Plot of pca$rotation of X COMP



Plot of pca$rotation of Y COMP



Plot of pca$rotation of Y COMP

Figure 5: **pca$rotation** of all axex in IDA-SE dataset

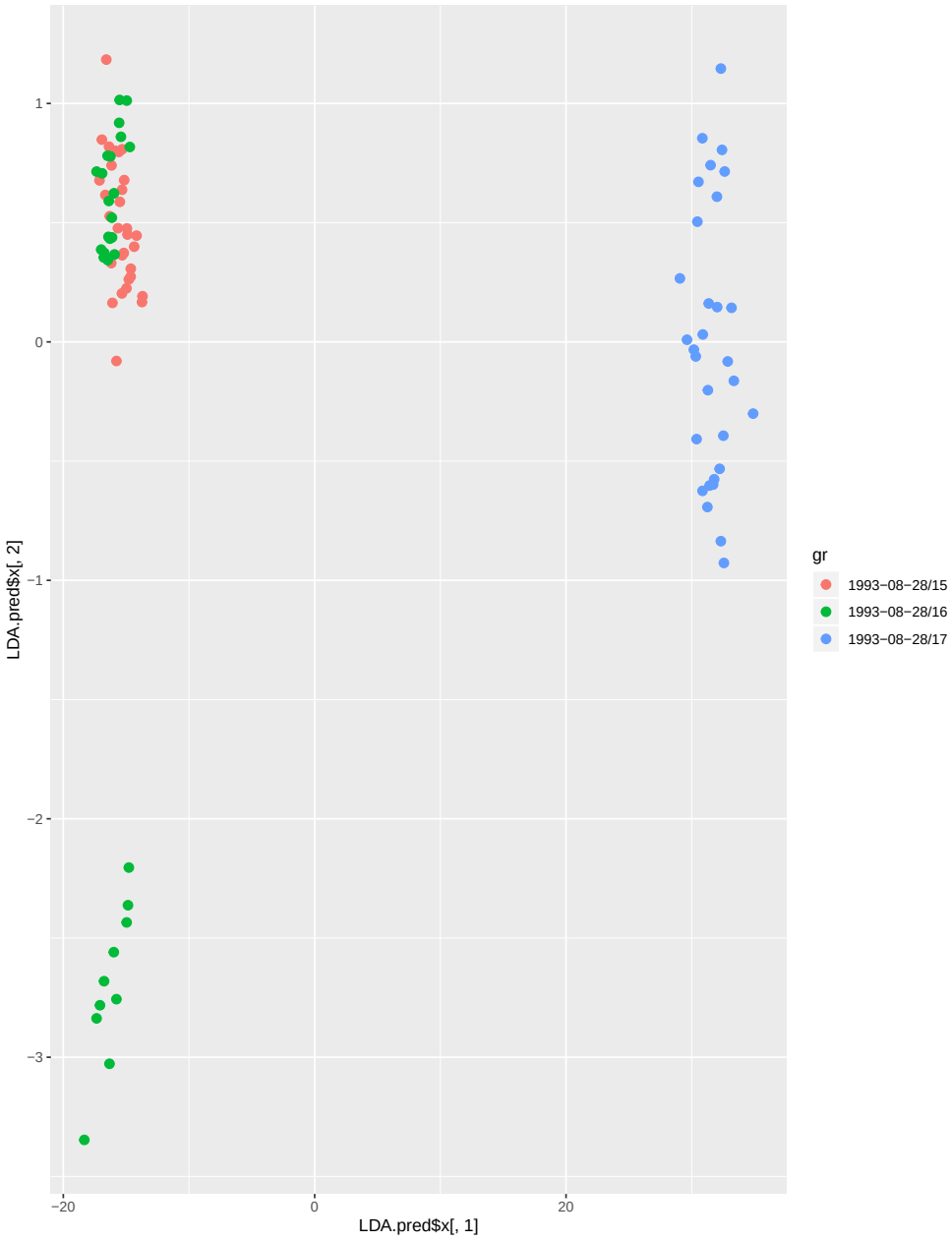


Figure 6: Plot of LDA of IDA-SE dataset using **ggplot()**

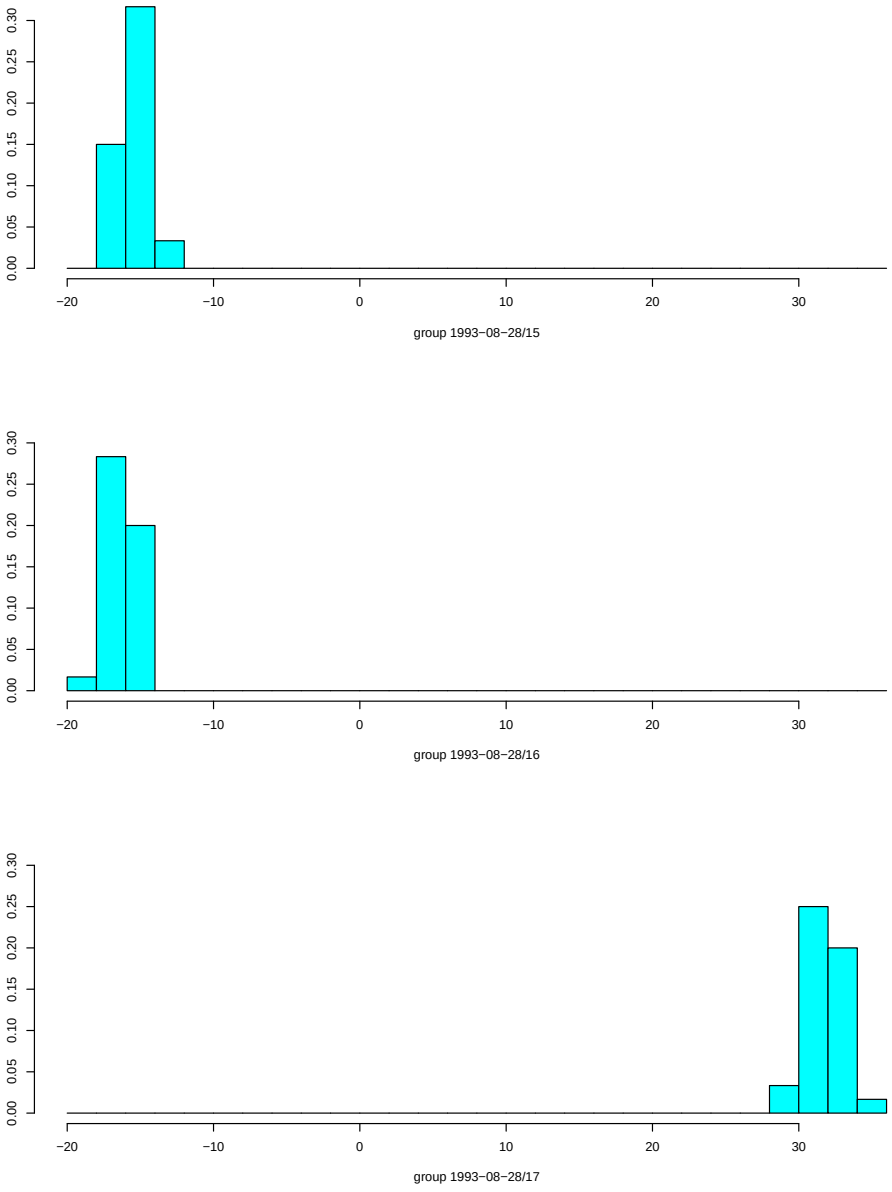


Figure 7: Plot of LDA.pred[, 1] of IDA-SE dataset using **ldahist()**

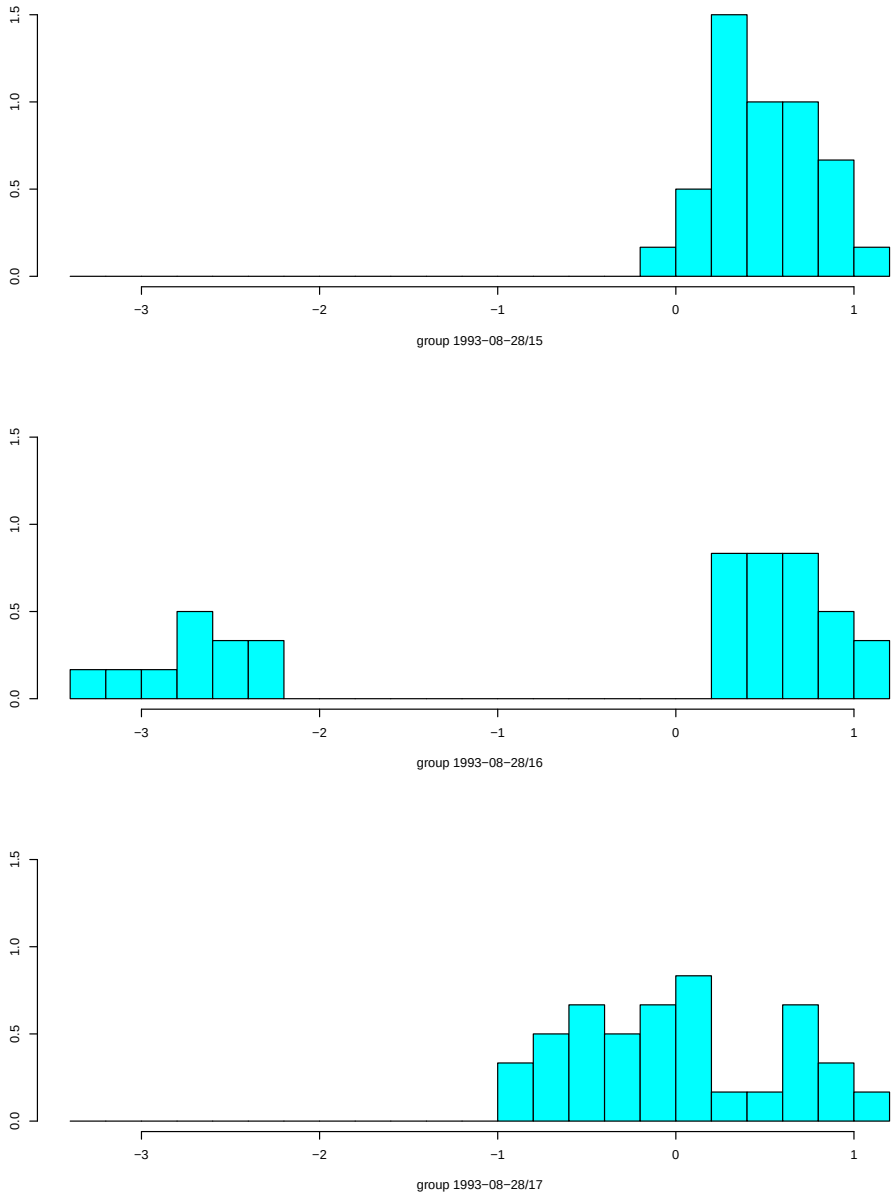


Figure 8: Plot of LDA.pred[, 2] of IDA-SE dataset using **ldahist()**