Independent Component Analysis (ICA) for Blind Source Separation

---

A Comprehensive Evaluation Report
Presented to
The Statistics Faculty
Amherst College

In Partial Fulfillment
of the Requirements for the Degree
Bachelor of Arts
in
Statistics

Sara J. Culhane

September 17, 2017

Acknowledgements

I want to thank a few people.

Table of Contents

List of Tables

List of Figures

Abstract

Most of statistics and even some aspects of machine learning rely on linear models or Gaussian distributions in order to successfully model and separate signal from noise. While our focus in coursework at Amherst has mostly been involved analysis based on known or applied variables, Independent Component Analaysis (ICA) and other methods of Blind Source Separation (BSS) call for the decomposition of their components with much more limited knowledge than we have seen previously. By identifying independent, non-Gaussians components of a mixture with ICA, the desired/original source can be uncovered. R packages like FastICA, make its application to more efficient and usable.

Introduction

0.1 Overview : What is Blind Source Separation (BSS)?

Blind Source Separation is a set of methods used to recompose unknown original sources. This is done from known observed sources with consideration of a set of unknown weights applied to the unknown source that generate the observations.

In other words, there are always two unknowns (vector and matrix) and one known vector.

On a fundamental level, BSS is a method of solving a system of linear equations in which:

$x = A \cdot s$ (1)

Where we are looking for a solution to this system in the form:

$s = A^{-1} \cdot x$

However, these s_i are not known values but instead a vector of distinct source signals defined by an also unknown operator matrix A. The x_i are values that we can observe from the signal without decomposition.

In general we have,

• N unknown sources s_j (eg. the original sources to recover)

• One unknown operator A (an "un-mixing" or "de-mixing" matrix)

• P observed signals x_i with some relation:

$x = A(s)$

With this, the chosen BSS method takes the P known observations, performs a seperation technique (also known as un-mixing) and outputs the unknown N sources that have been obscured by some unobserved process (i.e. the data collection proccess or limits of a set of data in most cases).

These y_k outputs should effectively estimate the original sources that make up the s vector.

(Puigt, 2011)

0.2 ICA Generally

0.2.1 Definition: Independent Component Analysis (ICA)

While ICA follows the same basic matrix model as described above in figure 1, it can also often also be written as:

$$x = \sum_{i=1}^{n} a_i s_i$$

The key qualities differentiating ICA from general BSS are the following:

• Assumption of statistical independence amongst the unknown source components s_j

• These independent compontents are also assumed to have strictly Non-gaussian distributions (or at most one independent component with a gaussian distribution)

(p. 2, Hyvärinen & Oja, 2000)

0.3 Primary Features of ICA

0.3.1 Independence

As independence is critical condition of ICA for the unknown s_j source variables, it is important to define it for clarity.

Looking at random variables (RV) y_1 and y_2 in general, they will be independent if and only if their joint probability density function (pdf) can be rewritten as:

$$p(y_1, y_2) = p_1(y_1)p_2(y_2)$$

This can be extended to the expectation of these two functions and is denoted:

$$E(h_1(y_1)h_2(y_2)) = E(h_1(y_1))E(h_2(y_2))$$

(p. 3-4, Hyvärinen & Oja, 2000)

0.3.2 Non-Gaussianity and Measurement

Though the Central Limit Theorem asserts that the sum of two or more independent RV's will tend towards a Gaussian distribution, the process of un-mixing actually produces non-gaussian signals. (Puigt, 2011)

ICA requires Non-Gaussianity for the distributions of all source signals, as accurate estimation of the mixing matrix A is not possible without meeting this condition.

To show this, assume that the known mixing matrix A is orthogonal (eg. $A^T A = AA^T = I$) and that all of the s_i are Gaussian.

Then, x_1 and x_2 have joint density:

$$p(x_1, x_2) = \frac{1}{2\pi} exp(\frac{x_1^2 + x_2^2}{2})$$

The symmetry of this joint density function makes it impossible to know the directionality of the columns of matrix A, and therefore, the matrix cannot be estimated, rendering ICA uneffective in the Gaussian case.

(p. 5, Hyvärinen & Oja, 2000)

Kurtosis

Kurtosis is defined as the 4th order cumulant:

$$kurt(y) = E(y^4) - 3(E(y^2))^2$$

Since y= 1, it can be simplied to:

$$kurt(y) = E(y^4) - 3$$

Which is equivalent to the 4th moment of y

In a Gaussian Distribution, the 4th moment equals $3(E(y^2))^2$

(p. 6, Hyvärinen & Oja, 2000)

Thus,

$$kurt(y) = 3(E(y^2))^2 - 3(E(y^2))^2 = 0$$

All Gaussian distributions have kurtosis equal to 0, and therefore, it is expected that most non-Gaussian distributions will have a nonzero Kurtosis with some rare exceptions. Beyond this, Kurtosis essentially measures the deviation of RV from Gaussianity. (Puigt, 2011)

It's use in ICA is popular due to the ease of estimation as well as its linearity property. This holds two properties, assuming that x_1 and x_2 are independent:

$$kurt(x_1 + x_2) = kurt(x_1) + kurt(x_2)$$

$$kurt(\alpha x_1) = \alpha^4 kurt(x_1)$$

Issues with Kurtosis - Highly sensitive to outliers as just a few extreme values can dramatically impact its calculation. Therefore, its application to skewed data may not be accuarate.

(p. 7, Hyvärinen & Oja, 2000)

Given this and the preference for another method in R packages to be utilized later, kurtosis will not be discussed much further in this paper, and more robust methods of measuring non-gaussianity will be explored instead.

Negentropy

Since Kurtosis cannot always successfully determine non-gaussianity, the use of negentrophy as a method of estimation is often necessary.

Negentrophy dervives from the theortical "differential" entrophy or level of randomness of a given varible. In generally, Gaussian variables tend to possess the largest entrophy given control for an equal variance condition. (8, Hyvärinen & Oja, 2000)

Theoritical entrophy is defined for discrete and continuous respectively as:

$$H(Y) = -\sum_{i=1} P(Y = a_i) log P(Y = a_i)$$

$$H(\mathbf{y}) = -\int f(\mathbf{y}) log f(\mathbf{y}) dy$$

To obtain the desired negative differential entrophy, the calculation modifies to:

$$J(\mathbf{y}) = H(\mathbf{y}_{gauss}) - H(\mathbf{y})$$

Critically in the above equation, $\mathbf{y}_{gauss}$ is the Gaussian RV of with identical covariance as $\mathbf{y}$.

It is always:

• Non-negative and will be zero iff $\mathbf{y}$ is Gaussian.

• Invariant for invertible linear transformations

In general, negentrophy is the best estimator of non-gaussianity but can be difficult or expensive to compute. However, the ease of computation has improved over time with technology, and thus the fastICA package relies on the estimator in its calculation.

(p. 8, Hyvärinen & Oja, 2000)

Minimization of Mutual Information

Though Negentrophy will be primarily used in subsequent application in this paper through fastICA, other forms of ICA estimation are still important to note for thoroughness purposes. A third commonly used estimator is the minimization of mutual information technique.

Mutual Information - The MI between m scalar variables $y_i, i = 1, ..m$ is indicated by:

$$I(y_1, y_2, ..., y_m) = \sum_{i=1}^{m} H(y_i) - H(\mathbf{y})$$

(p. 9, Hyvärinen & Oja, 2000)

Maximum Likelihood Estimation (MLE)

Similar to the mutual information method is MLE for the ICA model.

Particularly, thelikelihood of the noise-free model can be formulated then used to find an estimate with MLE.

Log-likelihood for ICA:

Denote:

$$\mathbf{W} = (\mathbf{w}_1, ...\mathbf{w}_n) = \mathbf{A}^{-1}$$

$$L = \sum_{t=1}^{T} \sum_{i=1}^{n} log f_j(\mathbf{w}_i^T \mathbf{x}(t)) + T log|det \mathbf{W}|$$

(p. 10 Hyvärinen & Oja, 2000)

In this, the f_i signify the density functions of the s_i, which would generally be unknown but is known here.

$$\mathbf{x}(t), t = 1, ..., N$$

are the observations of x

$$log|det\mathbf{W}|$$

is linear transformation of an RV and its density

(p. 11 Hyvärinen & Oja, 2000)

Connect to Mutual Information

ICA and Projection Pursuit

0.4 Pre-proccessing in ICA

0.4.1 Centering

Practically speaking, subtracting the expectation/mean vector $\mathbf{m} = E(\mathbf{x})$ center the observations and make x zero-mean.

This is strictly for algorithmic simplification purposes.

(p. 12, Hyvärinen & Oja, 2000)

0.4.2 Whitening

As a concept, whitening is the process by which generating vectors with components that are both uncorrelated and equal variance with theortical notation for the covariance matrix of a whiten vector as stated below:

$$E(\tilde{x}\tilde{x}^T) = \mathbf{I}$$

(p. 12, Hyvärinen & Oja, 2000)

Whitening, als known as *Sphering*, removes the scale and correlation structure from **X**. In general, the proccess has been criticized for manipulating the distribution too far but remains standard practice for pre-processing data for ICA. (p. 5, Izenman, 2003)

0.5 Toy Example and Simulation

0.5.1 Quick Look at FastICA

Though the next chapter will dig deeper in demonstrating the power of the FastICA R package in generating ICA models, key features should be pointed out before their application to two toy examples in this chapter.

Key Details

fastICA takes the data matrix X of the model and attempts to un-mix components. Explicitly, this matrix is a linear combination of matrices S (original source signals) and A (mixing matrix) by generating matrix W that maximized non-Gaussianity, which is $\mathbf{WX = S}$

This non-gaussian approximation is based on Negentrophy calculations discussed in prior sections and it is used over kurtosis because of efficiency according to the authors of the package. They did not explicit indicate why they chose it over MLE or Minimum information.

Specically for the FastICA algorithm, the $H(\mathbf{y})$ function options are

•$G(u) = \frac{1}{\alpha} logcosh(\alpha u)$

•$G(u) = -exp(u^2/2)$

(see J L Marchini <marchini@stats.ox.ac.uk> & <ripley@stats.ox.ac.uk>, 2017)

0.5.2 Arguments in the fastICA function

fastICA employs several different techincal arguments in its function that must be understood in order to interrupt the results of its output rigorously.

X simply the data matrix of interest

n.comp Number of components to be extracted

alg.typ

• if == "parallel" - components are extracted simultaneously(default)

• if == "deflation" - components are extracted one at a time

fun Function form of G to use (see above)

alpha Constant range[1,2] used in apporx. to neg-entrophy when fun == "logcosh"

method

• if == "R" then computations are done exclusively in R

• if == "C" then C code is used to perform computations (runs faster)

row.norm logical valuve indicating whether rows of X should be standardized

maxit Maximum number of iterations

tol A positive scalar giving tolerance at which the un-mix is considered to have converaged

verbose level of output

w.init Initialized un-mixing matrix of dim(n.comp,n.comp) if Null matrix of RV's used

0.5.3 JADE and Infomax

Although fastICA has largely become the primary R packages for ICA modeling, JADE and Infomax methods are also avaliable to perform the same task. In subsequent sections, their effectiveness will be evaluated in relation to fastICA.

Particularly, the original Cardoso paper describes the algorithm in 4 steps:

1 Create sample covariance $\hat{R}_x$ and compute whitening matrix $\hat{W}$

2 Create sample of 4th-order cumulants of whitening; compute significant eigenpairs

3 Jointly diagonialize the set with unitary matrix

4 Estimate A with $\hat{A} = \hat{W}\hat{U}$
(366 **???**)

JADE -

Uses *Joint Approximate Diagonalization of Eigenmatrices* (JADE) dervived by Cardoso and Souloumiac in 1993. This approach involves finding a orthologonal rotation matrix R that diagonalizes the cumulant arry of source signals. (Helwig, 2015)

Infomax -

Also utilizes the orthogonal rotation matrix R that maximizes the joint entrophy of a non-linear function of the estimated source signals. (Helwig, 2015)

0.5.4 Simple Example: Two Random uniform Distributions

To demonstrate effectiveness of ICA on decomposing source signals and to develop a better undering, a very simple toy example is simulated. The"unknown" source components in this model are just two random uniform variables in the matrix S, distorted by some arbitary "unknown"

Here it is a 2x2 matrix A

$$A = \begin{bmatrix} 1 & 1 \\ -1 & 3 \end{bmatrix}$$

.

The known observations is the product of these two matrices, or matrix X.

To test ICA, apply FastICA to this simple model and generate 5 plots:

- The pre-processed data
- PCA components (for comparison)
- ICA components

  - fastICA

  - JADE method

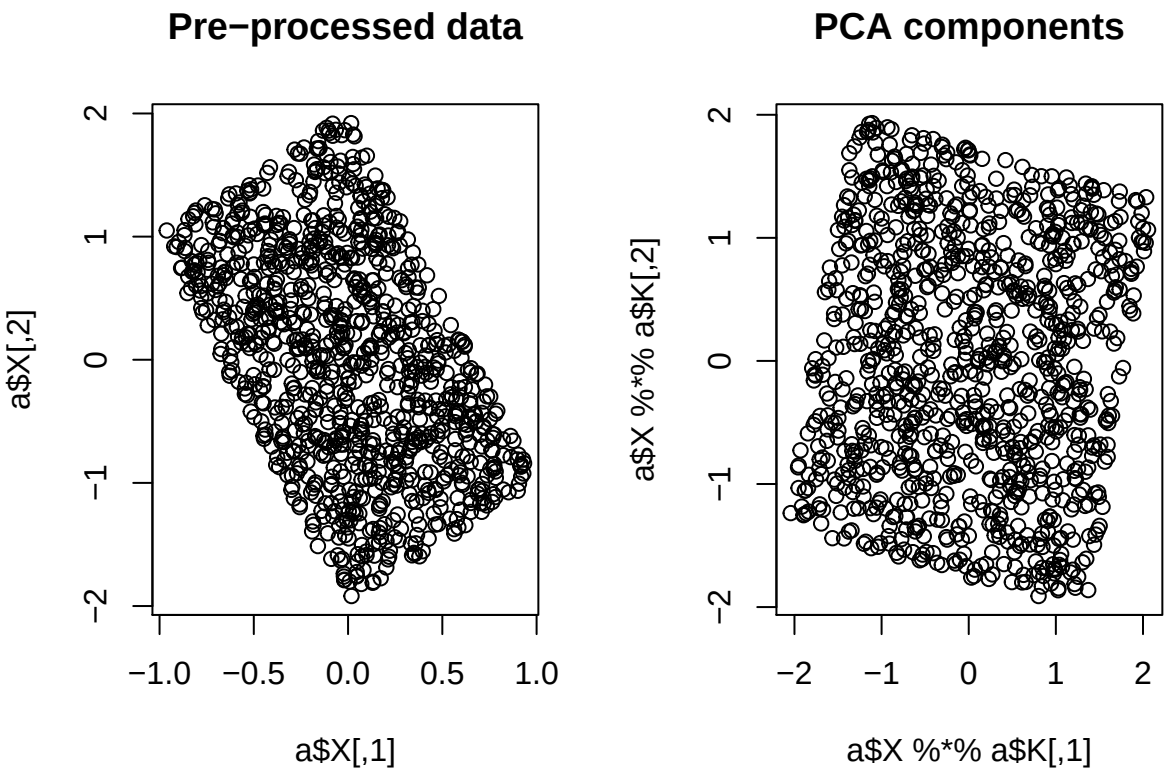
  - Infomax

```
set.seed(10)
library(fastICA)
S <- matrix(runif(10000), 1000, 2)
A <- matrix(c(1, 1, -1, 3), 2, 2, byrow = TRUE)
X <- S %*% A

a <- fastICA(X, 2, alg.typ = "parallel", fun = "logcosh", alpha = 1,
             method = "C", row.norm = FALSE, maxit = 200,
             tol = 0.0001, verbose = TRUE)
```

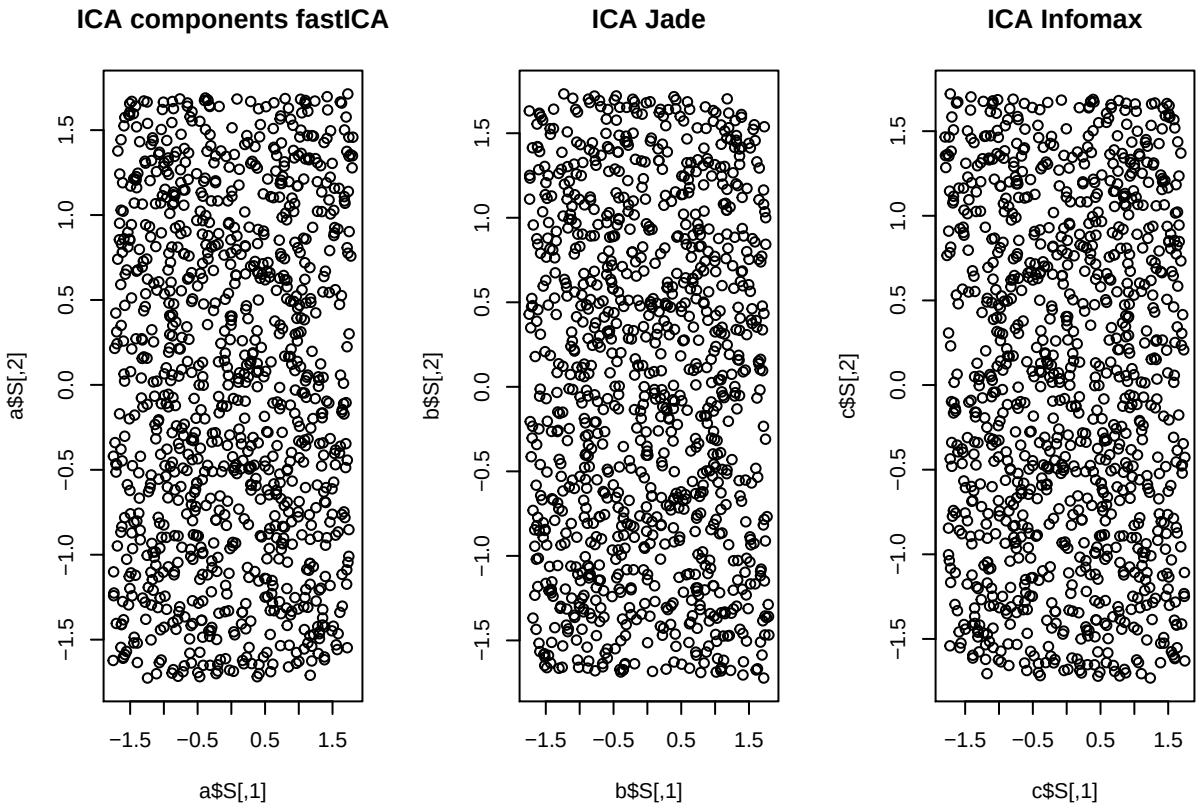
```
Centering
Whitening
```

```
Symmetric FastICA using logcosh approx. to neg-entropy function
Iteration 1 tol=0.000001
```

```
b <- icajade(X,2,center=TRUE,maxit=200,tol=0.0001)

c <- icaimax(X,nc=2,center=TRUE,maxit=200,tol=0.0001,alg="newton",fun="log")

par(mfrow = c(1, 2))
plot(a$X, main = "Pre-processed data")
plot(a$X %*% a$K, main = "PCA components")
```




```
par(mfrow = c(1, 3))

plot(a$S, main = "ICA components fastICA")
plot(b$S,main="ICA Jade")
plot(c$S, main="ICA Infomax")
```



Plot Interpretation

Looking at these plots, we see that the preprocessed data is rotated and no signals are clear from it.

PCA appear to have flattened the data slightly and creates almost a mirror of the original data, but the true signals from the distributions still do not appear completely evident. PCA does not effectively separate sources given how it rotates data.

However, looking at the ICA transformation, we can which clearly see the shape of a uniform distribution forming after the decomposition. ICA was able to recover the original signals quite easily in this simple simulation. The results for all 3 ICA packages (JADE,Infomax and fastICA) appear to produce approximately the same plots for this example.

0.5.5 (More) Complex Example: The Cocktail Party Problem

The most common example used to explain and motive the use of ICA is the "Cocktail Party Problem."

Two microphones are placed in a room where to different people are speaking at the same time. Each microphone signal is defined as $x_1(t)$ and $x_2(t)$, with some observed (i.e. known) amplitudes x_1 and x_2 with time as an index. The unknown speech signals by the speakers are $s_1(t)$ and $s_2(t)$ to make up the following system of equations:

$$x_1(t) = a_{11}s_1 + a_{12}s_2$$

$$x_2(t) = a_{21}s_1 + a_{22}s_2$$

This can also be rewritten as:

$$x_i = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$s_i = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}$$

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

If we knew the weighted a_{ij}, this problem could be easily solved but since we do not, we must use blind source separation.

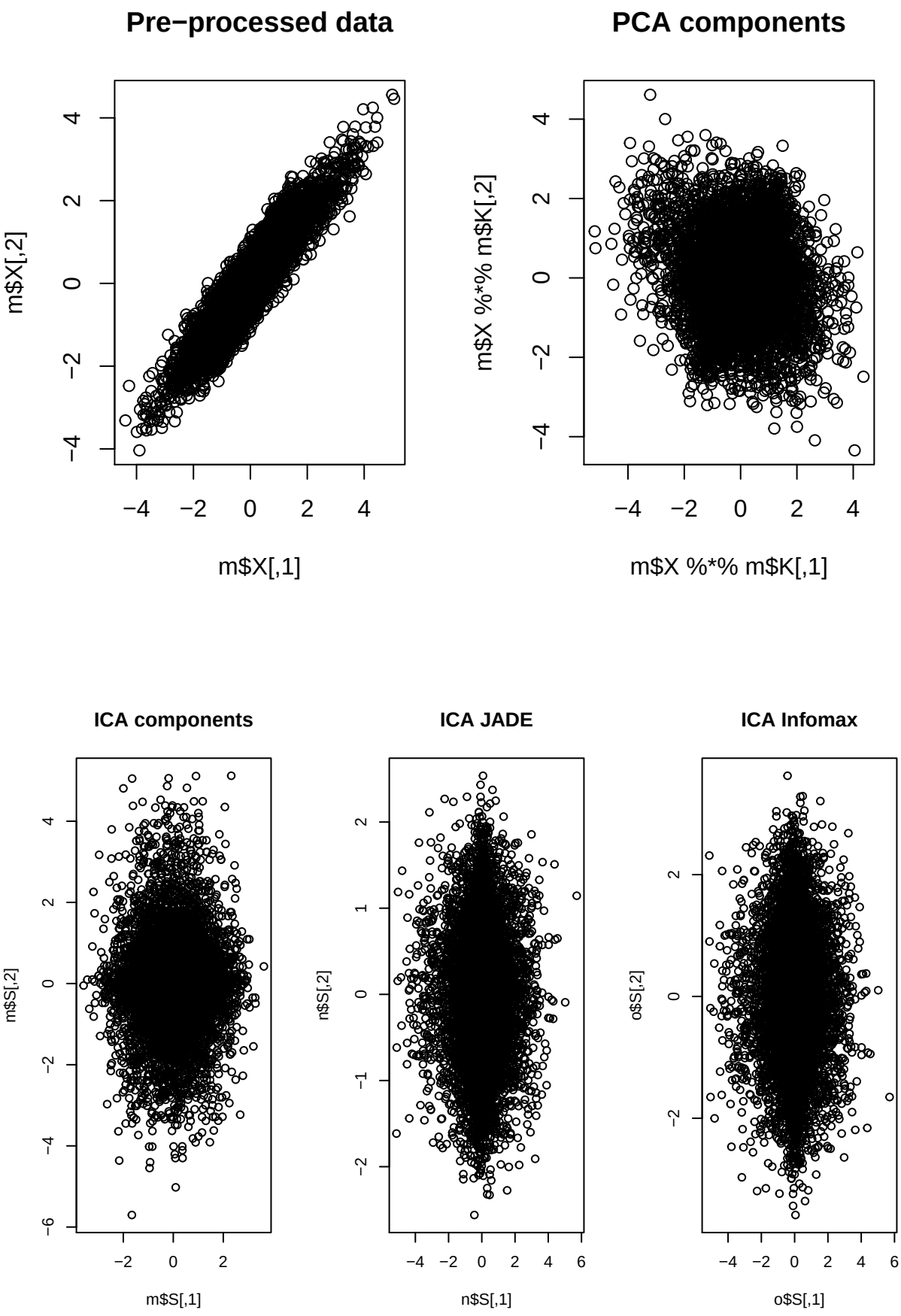
Example with Cocktail Party Data

As a test example/simulation, we can use the CPPdata from the JADE package that contains 50,000 observations from 4 different microphones.

```
Centering
Whitening
Symmetric FastICA using logcosh approx. to neg-entropy function
Iteration 1 tol=0.047273
Iteration 2 tol=0.007242
Iteration 3 tol=0.000012
```

```
Warning in icajade(dat, 4, center = TRUE, maxit = 200, tol = 1e-04): Numerical rank of X
  Number of components has been redefined as the numerical rank of X.
```

```
Warning in icaimax(dat, nc = 4, center = TRUE, maxit = 200, tol = 1e-04, : Numerical ran
  Number of components has been redefined as the numerical rank of X.
```


Pre−processed data
m$X[,2]
m$X[,1]
PCA components
m$X %*% m$K[,2]
m$X %*% m$K[,1]
ICA components
m$S[,2]
m$S[,1]
ICA JADE
n$S[,2]
n$S[,1]
ICA Infomax
o$S[,2]
o$S[,1]

Plot Interpretation

Again, we have a highly rotated set of pre-processed data and a PCA rotation that only transforms the data enough to separate the mixture ever so slightly.

With ICA, we see a full rotation and a much cleaner shape beginning to form, almost diamond-like. Like with the random uniform example, the superior performance of any one of the three ICA methods is not identifible just from these primitive plots of the model.

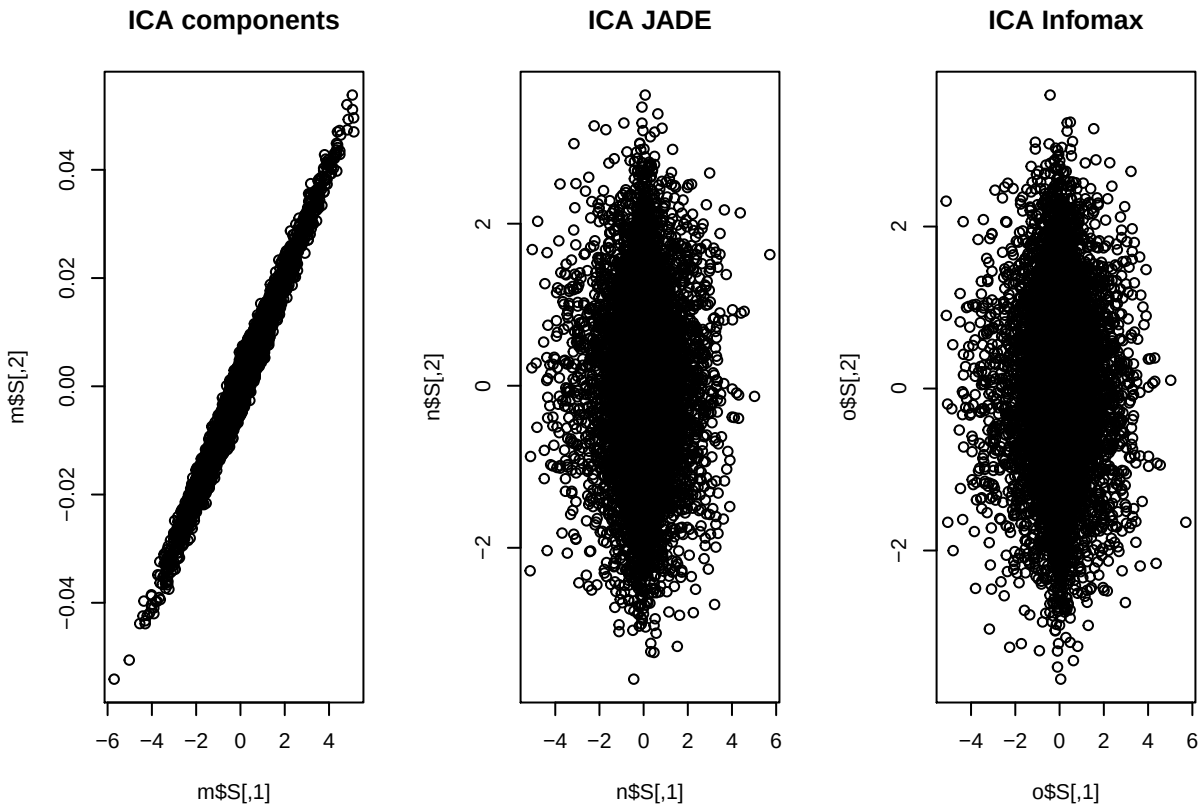
Of note with the ICA algorithms example is the "nc" (number of components) input of the function for each of these different methods. Particularly, both the JADE and Infomax produce similar plots at $nc = 4$, which makes sense as the dataset $CPPdata$ has observations from 4 microphones. However, to generate a similar plot for fastICA, $nc = 2$ was used. The results for $nc = 4$ with ICA appear as below (JADE and Infomax shown at $nc = 2$ for reverse comparison:

```
set.seed(10)
library(fastICA)
S <- matrix(runif(10000), 1000, 2)
A <- matrix(c(1, 1, -1, 3), 2, 2, byrow = TRUE)
X <- S %*% A
m <- fastICA(dat,4,alg.typ = "parallel", fun = "logcosh", alpha = 1,
             method = "C", row.norm = FALSE, maxit = 200,
             tol = 0.0001, verbose = TRUE)
```

```
Centering
Whitening
Symmetric FastICA using logcosh approx. to neg-entropy function
Iteration 1 tol=0.380613
Iteration 2 tol=0.254897
Iteration 3 tol=0.146511
Iteration 4 tol=0.072136
Iteration 5 tol=0.031287
Iteration 6 tol=0.012479
Iteration 7 tol=0.004745
Iteration 8 tol=0.001764
Iteration 9 tol=0.000650
Iteration 10 tol=0.000238
Iteration 11 tol=0.000087
```

```
n <- icajade(dat,2,center=TRUE,maxit=200,tol=0.0001)

o <- icaimax(dat,nc=2,center=TRUE,maxit=200,tol=0.0001,alg="newton",fun="log")

par(mfrow = c(1,3))
plot(m$S, main = "ICA components")
```

```
plot(n$S, main = "ICA JADE")
plot(o$S, main ="ICA Infomax")
```



While JADE and Infomax at $nc = 2$ do not differ greatly from $nc = 4$, the fastICA plot appears more similar to the pre-processed data when the number of components is set at 4.

Explain what this could mean

0.6 Simulation

Given the simplistic nature of both toy examples, a simulation can provide more robust insight into the effectiveness of ICA in extracting original source components.

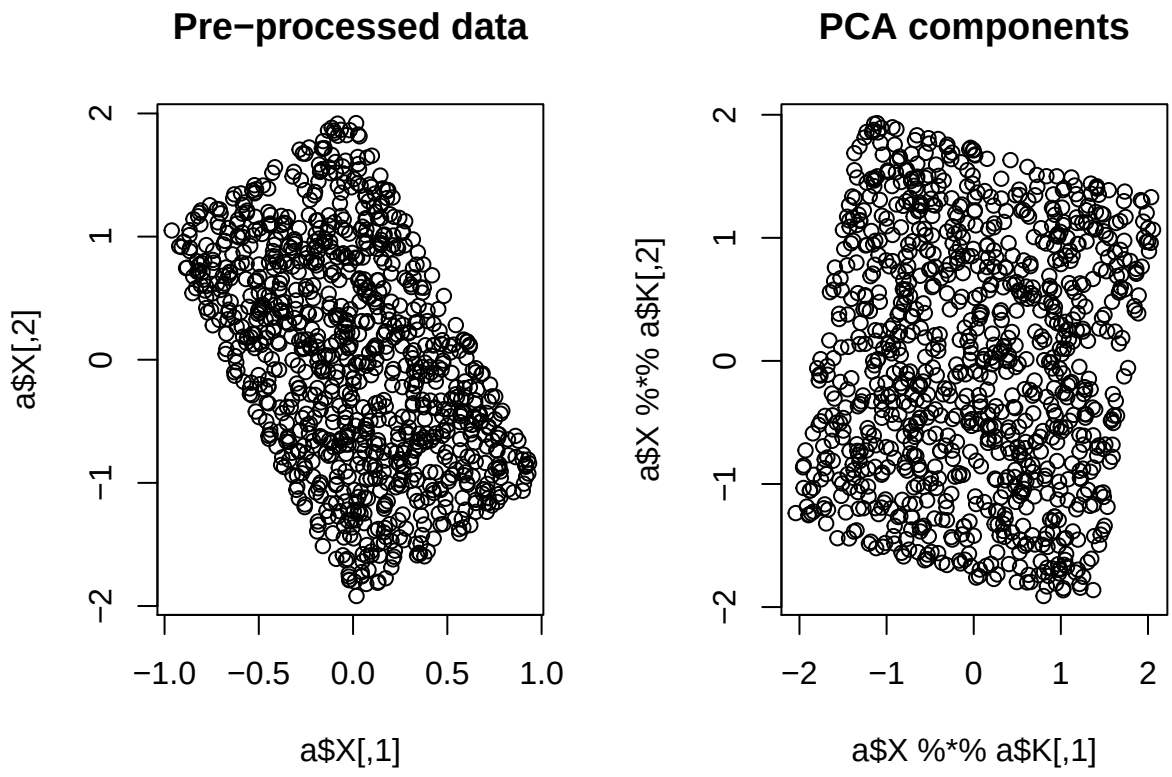
For this, we will use the Monte Carlo Markov Chain (MCMC) and see how ICA performs in a "noisier" setting.

0.6.1 Random Uniform Distribution

```
a <- fastICA(X, 2, alg.typ = "parallel", fun = "logcosh", alpha = 1,
             method = "C", row.norm = FALSE, maxit = 200,
             tol = 0.0001, verbose = TRUE)
```

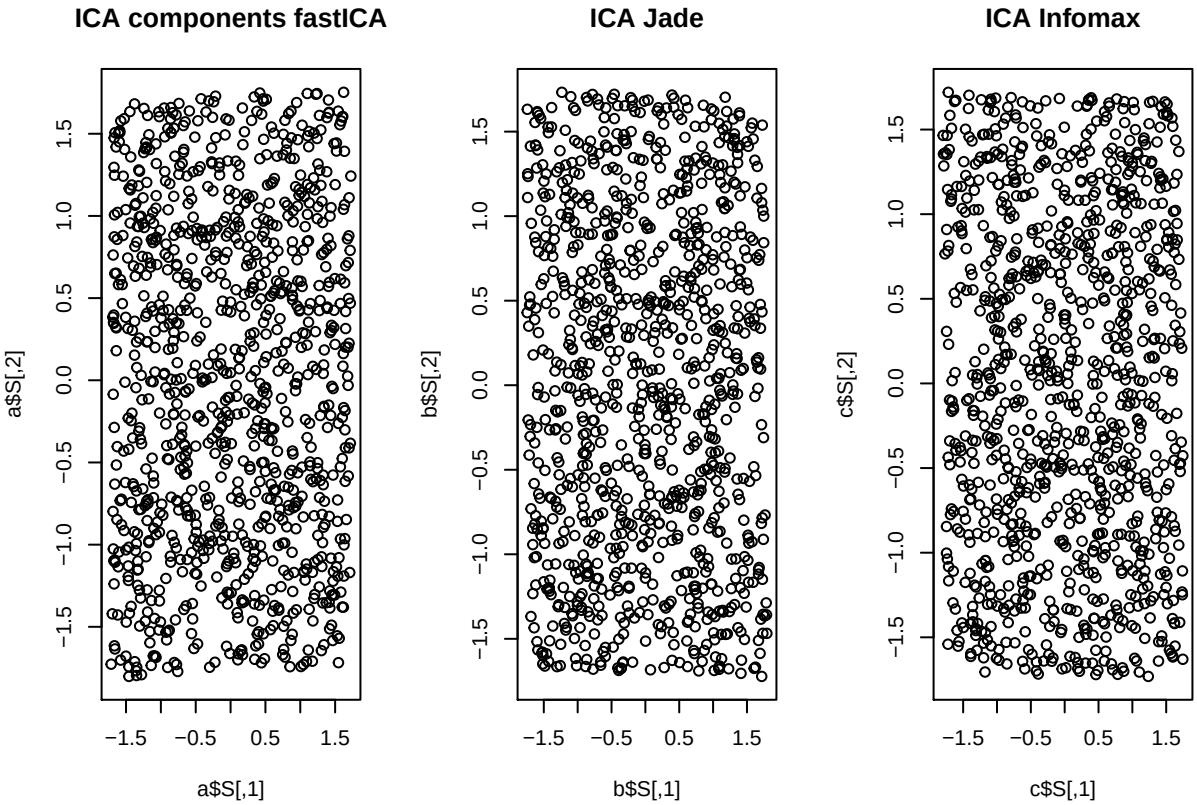
```
Centering
Whitening
Symmetric FastICA using logcosh approx. to neg-entropy function
Iteration 1 tol=0.116217
Iteration 2 tol=0.000607
Iteration 3 tol=0.000007
```

```
b <- icajade(X,2,center=TRUE,maxit=200,tol=0.0001)

c <- icaimax(X,nc=2,center=TRUE,maxit=200,tol=0.0001,alg="newton",fun="log")

par(mfrow = c(1, 2))
plot(a$X, main = "Pre-processed data")
plot(a$X %*% a$K, main = "PCA components")
```




```
par(mfrow = c(1, 3))

plot(a$S, main = "ICA components fastICA")
```

```
plot(b$S,main="ICA Jade")
plot(c$S, main="ICA Infomax")
```



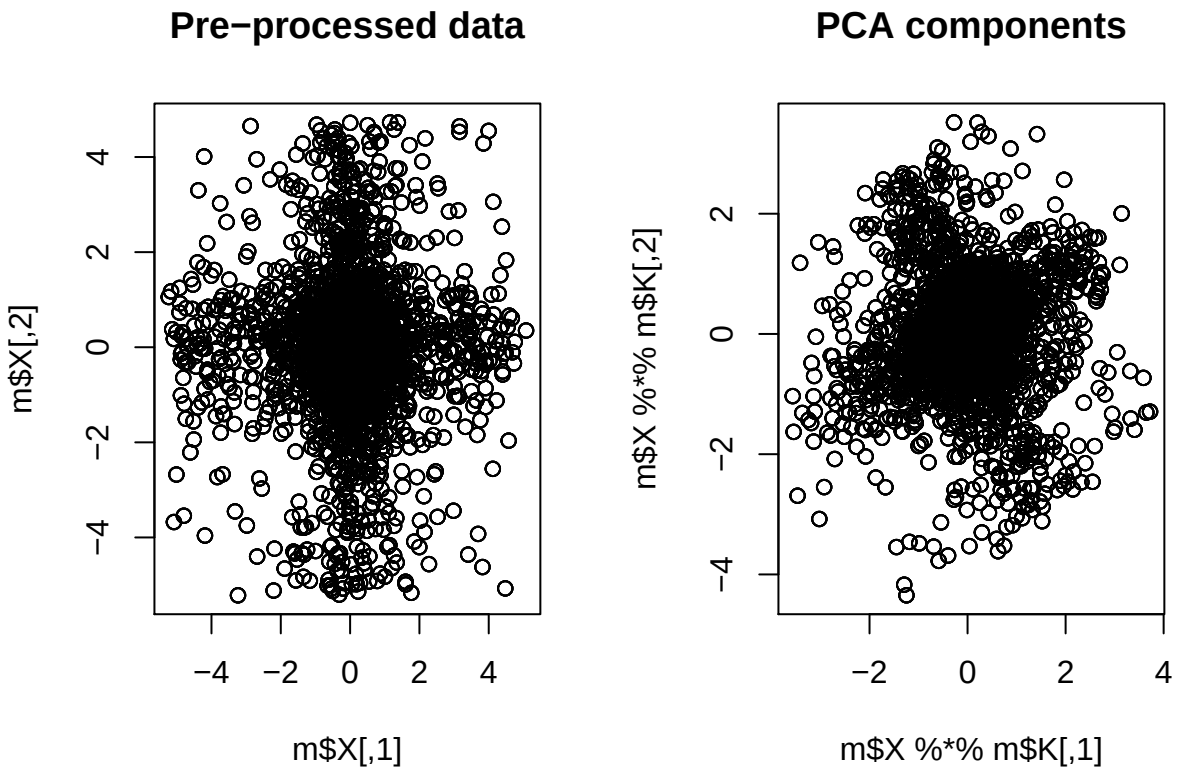
0.7 CPP Simulation

```
m <- fastICA(r1,2,alg.typ = "parallel", fun = "logcosh", alpha = 1,
             method = "C", row.norm = FALSE, maxit = 200,
             tol = 0.0001, verbose = TRUE)
```

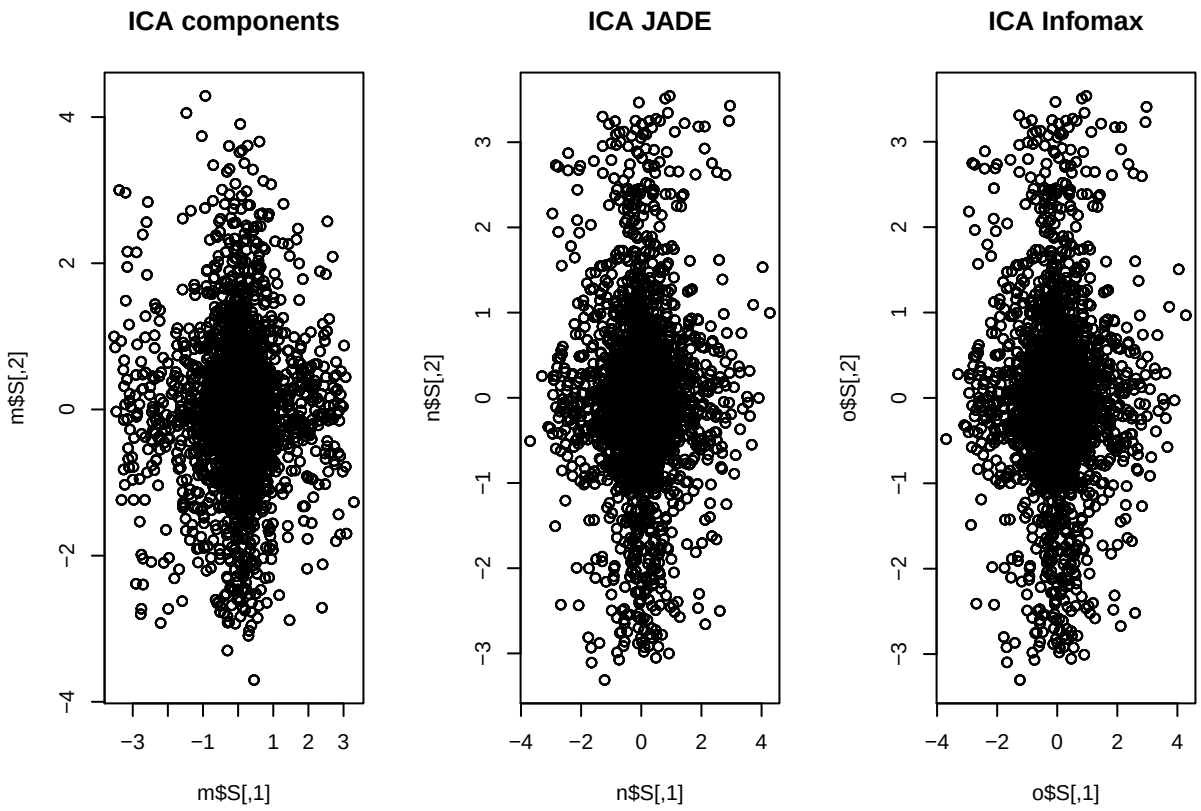
```
Centering
Whitening
Symmetric FastICA using logcosh approx. to neg-entropy function
Iteration 1 tol=0.006395
Iteration 2 tol=0.000040
```

```
n <- icajade(r1,2,center=TRUE,maxit=200,tol=0.0001)

o <- icaimax(r1,nc=2,center=TRUE,maxit=200,tol=0.0001,alg="newton",fun="log")
```

```
par(mfrow = c(1,2))
plot(m$X, main = "Pre-processed data")
plot(m$X %*% m$K, main = "PCA components")
```



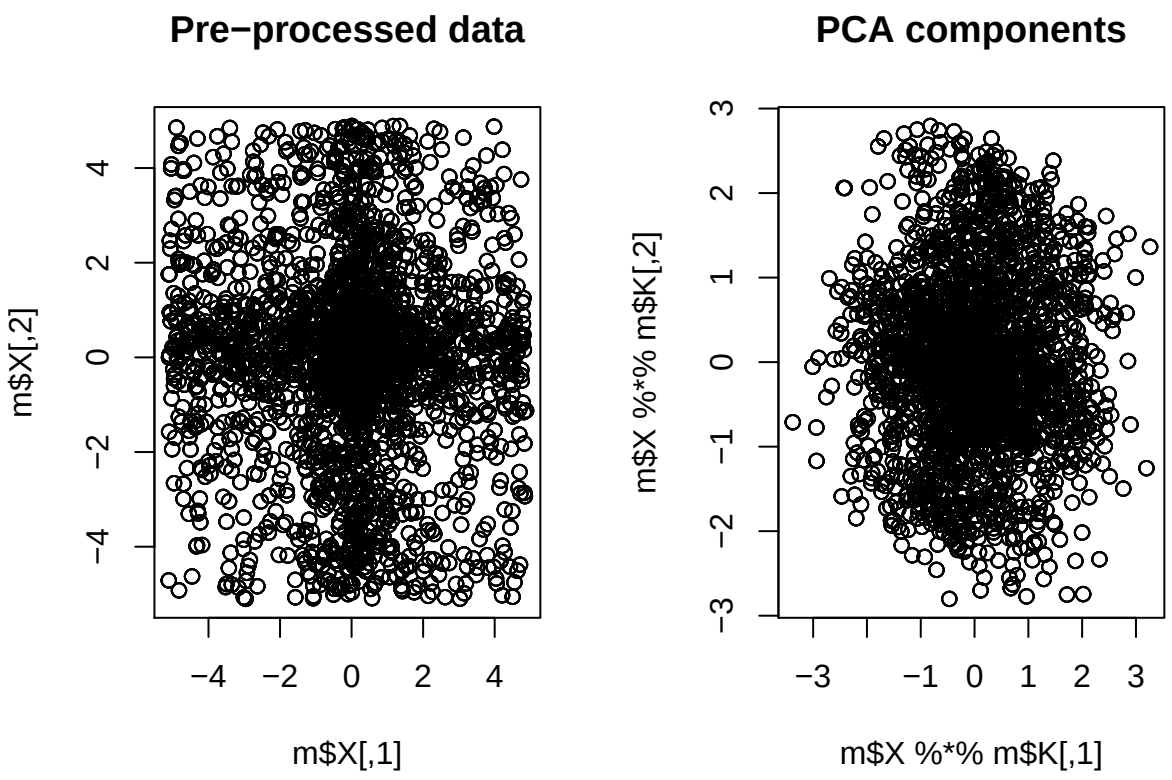

```
par(mfrow = c(1,3))

plot(m$S, main = "ICA components")
plot(n$S, main = "ICA JADE")
plot(o$S, main ="ICA Infomax")
```



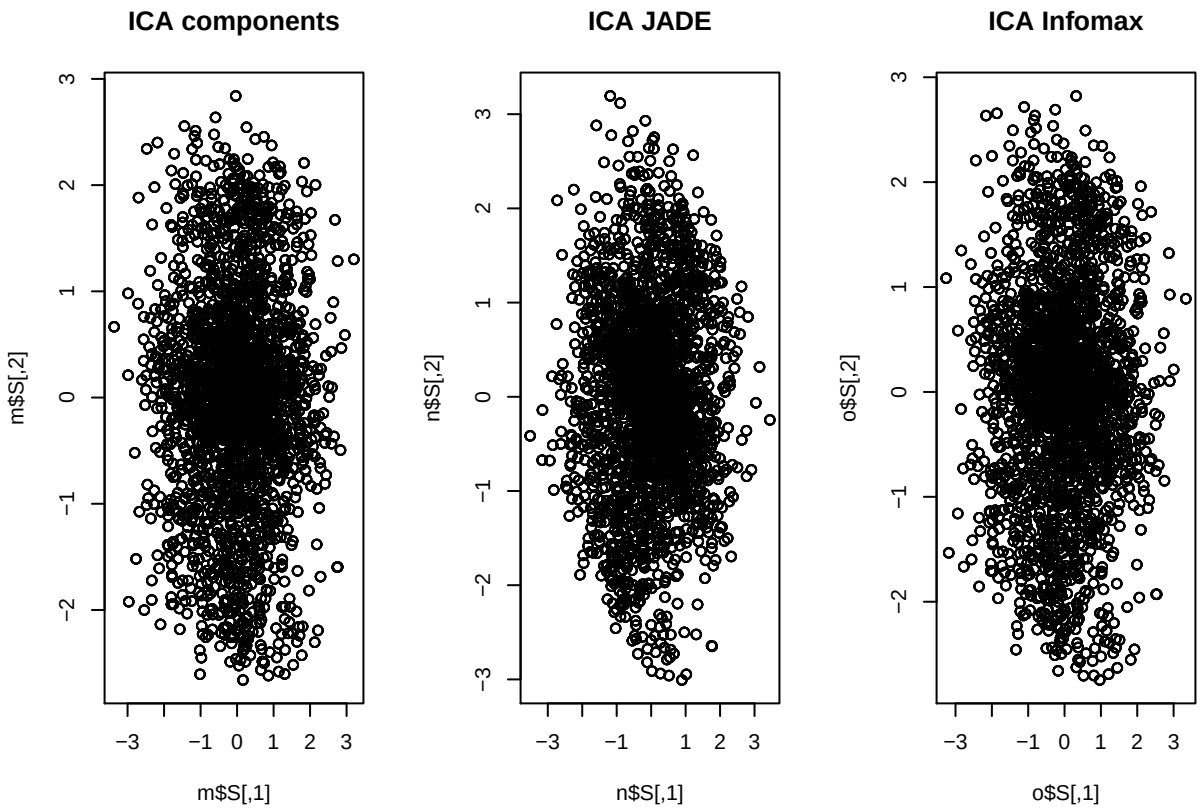

```
m <- fastICA(r2,2,alg.typ = "parallel", fun = "logcosh", alpha = 1,
             method = "C", row.norm = FALSE, maxit = 200,
             tol = 0.0001, verbose = TRUE)
```

```
Centering
Whitening
Symmetric FastICA using logcosh approx. to neg-entropy function
Iteration 1 tol=0.022903
Iteration 2 tol=0.002771
Iteration 3 tol=0.000168
Iteration 4 tol=0.000008
```

```
n <- icajade(r2,2,center=TRUE,maxit=200,tol=0.0001)

o <- icaimax(r2,nc=2,center=TRUE,maxit=200,tol=0.0001,alg="newton",fun="log")

par(mfrow = c(1,2))
plot(m$X, main = "Pre-processed data")
plot(m$X %*% m$K, main = "PCA components")
```



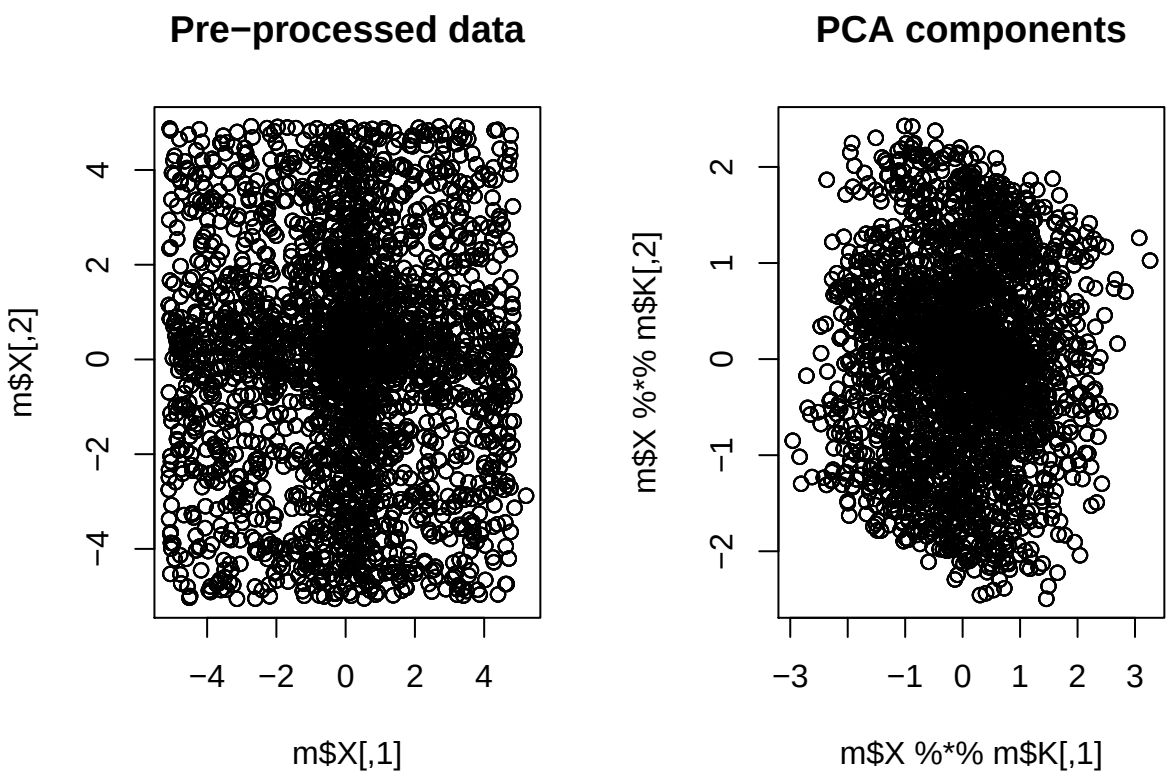

```
par(mfrow = c(1,3))

plot(m$S, main = "ICA components")
plot(n$S, main = "ICA JADE")
plot(o$S, main ="ICA Infomax")
```



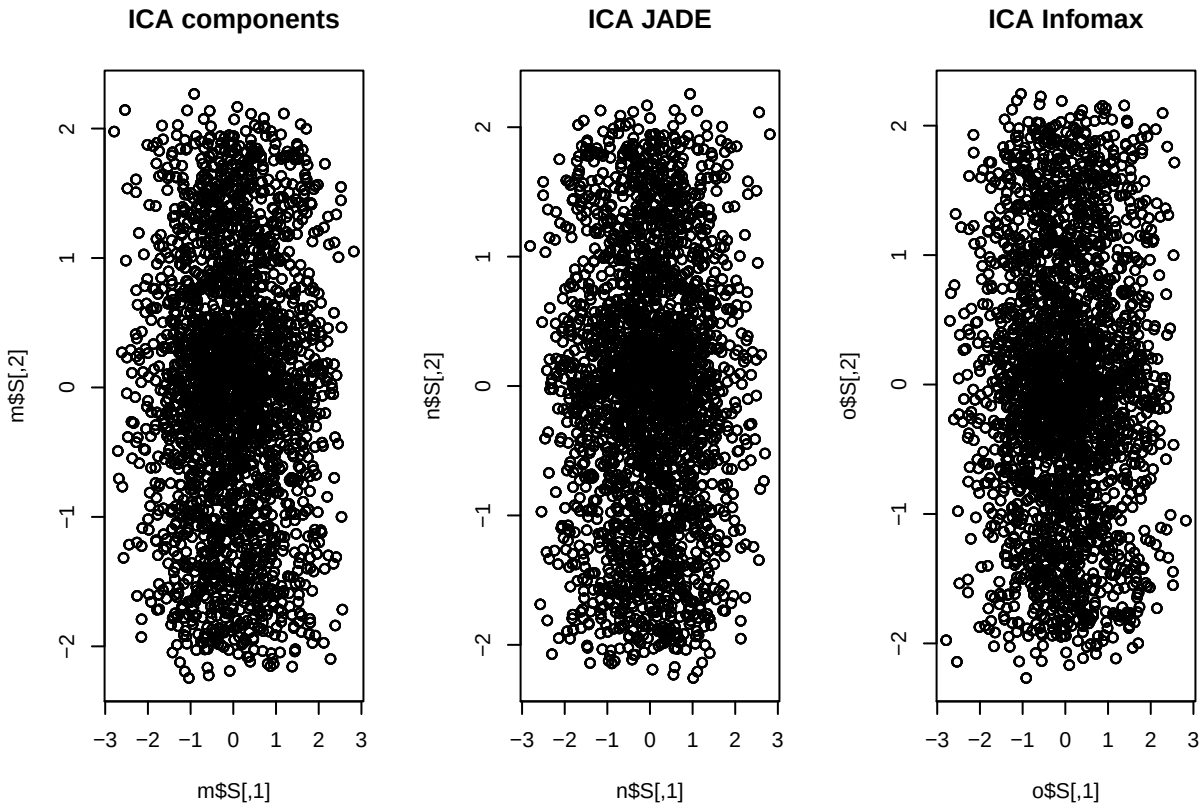

```
m <- fastICA(r3,2,alg.typ = "parallel", fun = "logcosh", alpha = 1,
             method = "C", row.norm = FALSE, maxit = 200,
             tol = 0.0001, verbose = TRUE)
```

```
Centering
Whitening
Symmetric FastICA using logcosh approx. to neg-entropy function
Iteration 1 tol=0.050045
Iteration 2 tol=0.014569
Iteration 3 tol=0.000013
```

```
n <- icajade(r3,2,center=TRUE,maxit=200,tol=0.0001)

o <- icaimax(r3,nc=2,center=TRUE,maxit=200,tol=0.0001,alg="newton",fun="log")

par(mfrow = c(1,2))
plot(m$X, main = "Pre-processed data")
plot(m$X %*% m$K, main = "PCA components")
```




```
par(mfrow = c(1,3))

plot(m$S, main = "ICA components")
plot(n$S, main = "ICA JADE")
plot(o$S, main ="ICA Infomax")
```


ICA components
ICA JADE
ICA Infomax
m$S[,2]
m$S[,1]
n$S[,2]
n$S[,1]
o$S[,2]
o$S[,1]
−3 −2 −1 0 1 2 3
−2 −1 0 1 2

Chapter 1

Computation and Application to Dataset

1.1 The Dataset: Walmart Store Data from Kaggle

From a 2014 Kaggle competition in attempt to forecast sales , the Walmart Store dataset provides information on sales for $n = 45$ located in different regions of the United States. Each store has Weekly Sales data from $2010-02-05$ to $2012-11-01$, information by department and a binary IsHoliday variable.

1.1.1 Prior Analysis : Kimmo Kiviluoto and Erkki Oja

In 1998, Kiviluoto and Erkki Oja used ICA on parallel Financial Time Series from a retail chain of 40 stores. They claim in their paper *Independent Comp onent Analysis for Parallel Financial Time Series* that by removing the fundamental factors of the data, they were able to see the impact of management decisions of a particular store more clearly.

With the walmart store data, an attempt to show a similar result using retail data will be made. However, this will look at weekly sales not cashflow data, which will probably not generate as robust results since cashflow provides more information about financial activities than just sales.

1.1.2 Exploration

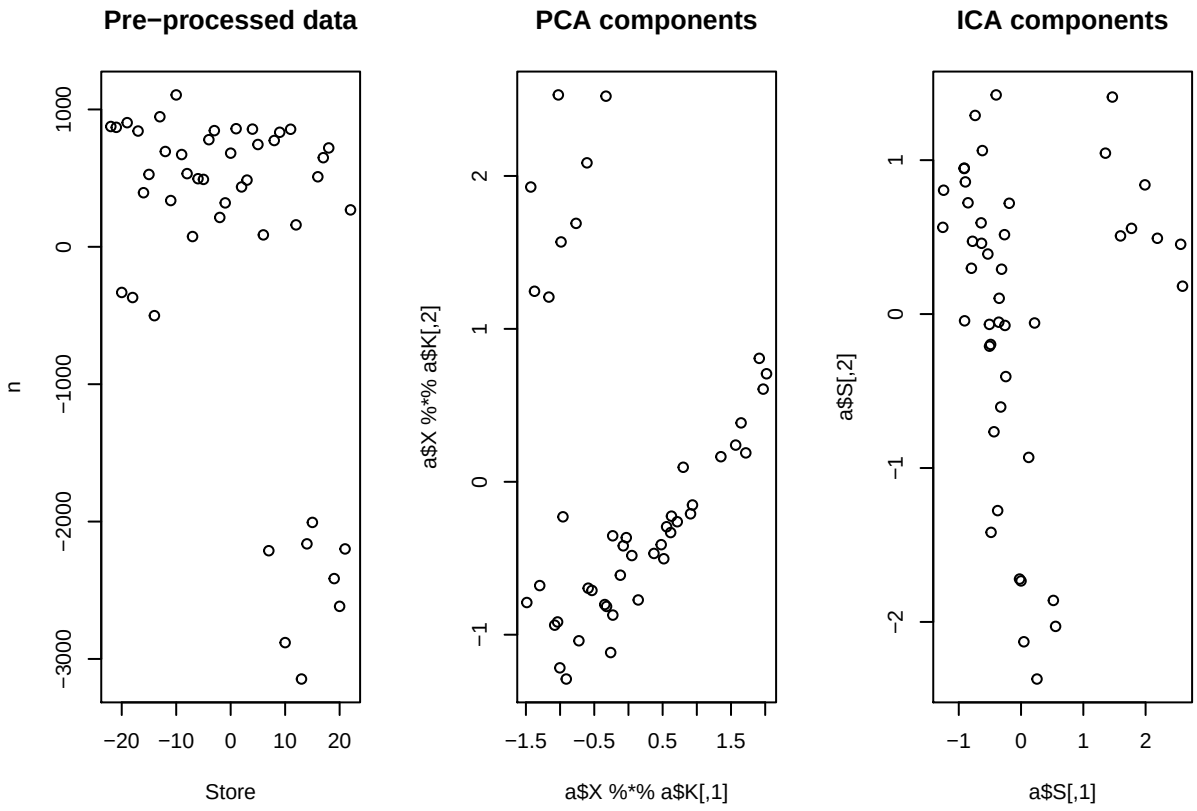
1.2 FastICA Algorithm

1.3 FastICA Application/Simulation

```
a <-fastICA(w2,3, alg.typ="parallel",fun= "logcosh",,row.norm=FALSE,maxit=5,tol= 0.0001,
```

```
Centering
Whitening
Symmetric FastICA using logcosh approx. to neg-entropy function
Iteration 1 tol = 0.1452916
Iteration 2 tol = 0.2135624
Iteration 3 tol = 0.3292911
Iteration 4 tol = 0.5126155
```

```
par(mfrow = c(1, 3))
plot(a$X, main = "Pre-processed data")
plot(a$X %*% a$K, main = "PCA components" )
plot(a$S, main = "ICA components")
```



```
bank <- read_csv("bank-full.csv")
```

```
Parsed with column specification:
cols(
  `age";"job";"marital";"education";"default";"balance";"housing";"loan";"contact"
)
```

1.4 Comparison to other Methods for ICA and for other BSS

Chapter 2

Visualization

2.1 Shiny App: Comparison of Methods

2.2 Summary of Performance from App

2.2.1 ICA

2.2.2 PCA

2.2.3 Sparse Component Analysis (SCA)

2.2.4 Non-negative Matrix Factorization (NMF)

Chapter 3

3.1 Tables

In addition to the tables that can be automatically generated from a data frame in **R** that you saw in [R Markdown Basics] using the `kable` function, you can also create tables using *pandoc*. (More information is available at `http://pandoc.org/README.html#tables`.) This might be useful if you don't have values specifically stored in **R**, but you'd like to display them in table form. Below is an example. Pay careful attention to the alignment in the table and the use of the hyphens to create the rows and columns.

Table 3.1: Correlation of Inheritance Factors for Parents and Child

Factors	Correlation between Parents & Child	Inherited
Education	-0.49	Yes
Socio-Economic Status	0.28	Slight
Income	0.08	No
Family Size	0.18	Slight
Occupational Prestige	0.21	Slight

We can also create a link to the table by doing the following: Table 3.1. If you go back to [Loading and exploring data] and look at the `kable` function code, you'll see that I added in a similar `\\label` to be able to reference that table later. (The extra backslash there is a way that *Markdown* interfaces with LaTeX.) We can create a reference to the max delays table: **??**.

The addition of the `\label{}` option to the end of the table caption allows us to then use the LaTeX `autoref` function to produce the link. The `ref` function in **R** allows for tables and figures to be referenced in the document easily without having to directly use the `autoref` function. It will automatically add "Table" before your number if you add the "tab:" prefix to your label. Note that this reference could appear anywhere throughout the document.

3.2 Figures

If your thesis has a lot of figures, *R Markdown* might behave better for you than that other word processor. One perk is that it will automatically number the figures accordingly in each chapter. You'll also be able to create a label for each figure, add a caption, and then reference the figure in a way similar to what we saw with tables earlier. If you label your figures, you can move the figures around and *R Markdown* will automatically adjust the numbering for you. No need for you to remember! So that you don't have to get too far into LaTeX to do this, a couple **R** functions have been created for you to assist. You'll see their use below.

In the **R** chunk below, we will load in a picture stored as `amherst.png` in our main directory. We then give it the caption of "Amherst logo", the label of "amherst", and specify that this is a figure. Note again the use of the `results = "asis"` specification to automatically include and compile the LaTeX code.

```
label(path = "figure/amherst.png", caption = "Amherst logo",
      label = "amherst", type = "figure")
```

Figure 3.1: Amherst logo

Here is a reference to the Amherst logo: Figure 3.1. Note the use of the inline **R** code here. By default "figure" is specified as the type. For clarity, we could have also added the `label` and `type` to the parameter specifications and this would give us the same result: Figure 3.1.

Below we will investigate how to save the output of an **R** plot and label it in a way similar to that done above. Recall the `flights` dataset from . (Note that we've shown a different way to reference a section or chapter here.) We will next explore a bar graph with the mean flight departure delays by airline from Portland for 2014. Note also the use of the `scale` parameter which is discussed on the next page.

```
delay_airline <- flights %>% group_by(carrier) %>%
  summarize(mean_dep_delay = mean(dep_delay)) %>%
  ggplot(aes(x = carrier, y = mean_dep_delay)) +
  geom_bar(position = "identity", stat = "identity", fill = "red")
ggsave("figure/delays.png", plot = delay_airline,
       width = 5, height = 3)
```

```
label(path = "figure/delays.png",
      caption = "Mean Delays by Airline",
      label = "delays", type = "figure",
      scale = 0.3)
```

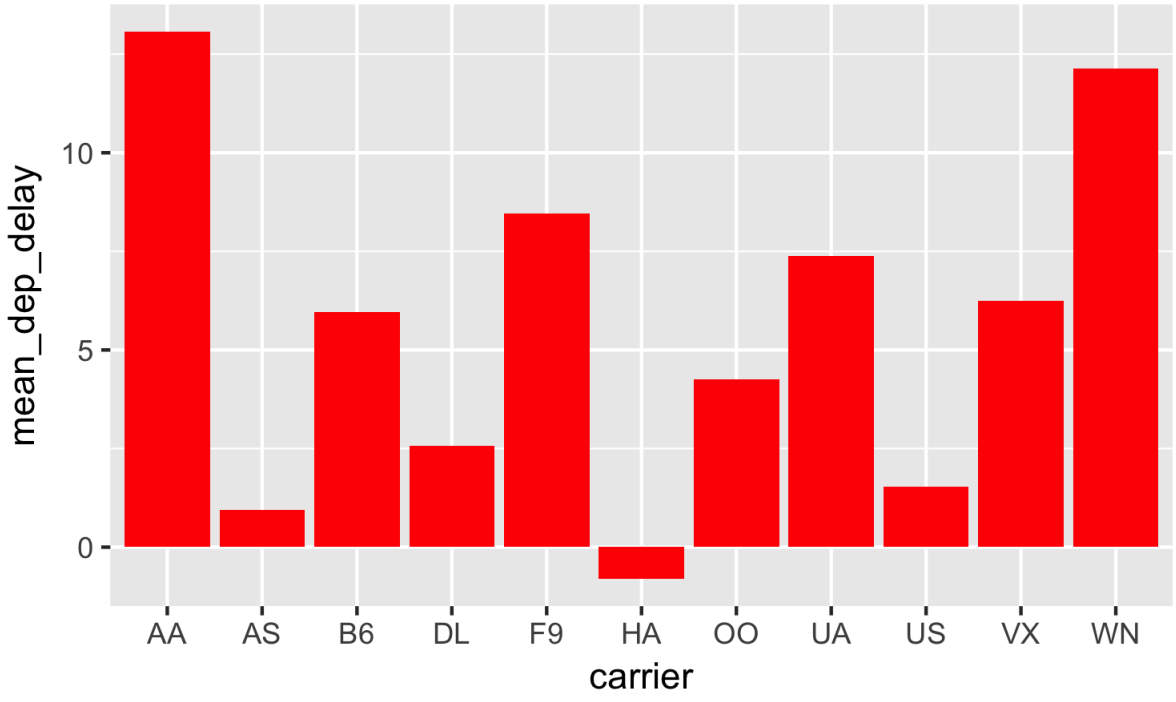


Figure 3.2: Mean Delays by Airline

A table linking these carrier codes to airline names is available at `https://github.com/ismayc/pnwflights14/blob/master/data/airlines.csv`.

Next, we will explore the use of the `scale` parameter which can be used to shrink or expand an image. Here we use the mathematical graph stored in the "subdivision.pdf" file. Note that we didn't specify the `caption =` or `label =` here, but we could have.

```
label("figure/subdivision.pdf", "Subdiv. graph", "subd",
      scale = 0.75)
```

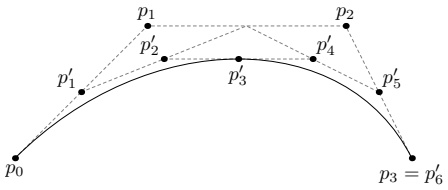


Figure 3.3: Subdiv. graph

Here is a reference to this image: Figure 3.3. (Move this around throughout the document as you wish.)

More Figure Stuff

Lastly, we will explore how to rotate figures using the `angle` parameter.

```
label("figure/subdivision.pdf",
      "A Larger Figure, Flipped Upside Down",
      scale = 1.5,
      angle = 180,
      label = "subd2")
```

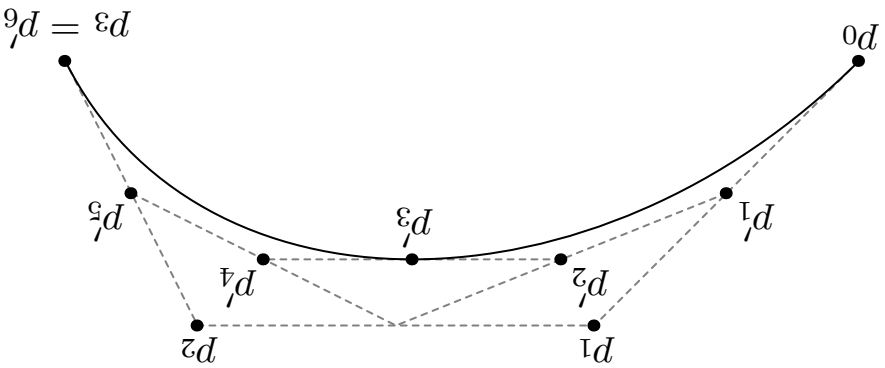


Figure 3.4: A Larger Figure, Flipped Upside Down

As another example, here is a reference to this figure: Figure 3.4.

Common Modifications

The following figure features the more popular changes thesis students want to their figures. We can add math to the caption that displays below the picture, specify the size of our caption to display below the figure (list of sizes available at this link), and also specify that a different caption `alt.cap` be what appears in the Table of Figures for this figure.

If you'd like to make further tweaks to figures, you might need to invoke some LaTeX code.

```
label("figure/subdivision.pdf",
      caption = "Subdivision of arc segments",
      alt.cap = "You can see that $p_3 = p_6^\\prime$",
      cap.size = "footnotesize",
      label = "subd3")
```

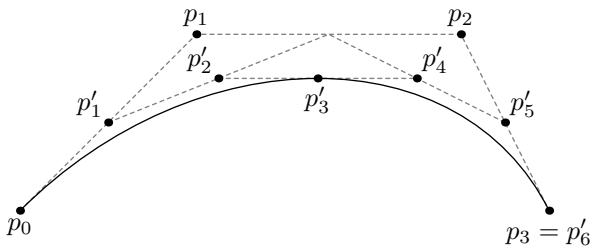


Figure 3.5: You can see that $p_3 = p_6'$

3.3 Footnotes and Endnotes

You might want to footnote something.[1] The footnote will be in a smaller font and placed appropriately. Endnotes work in much the same way.

3.4 Bibliographies

Of course you will need to cite things, and you will probably accumulate an armful of sources. There are a variety of tools available for creating a bibliography database (stored with the .bib extension). In addition to BibTeX suggested below, you may want to consider using the free and easy-to-use tool called Zotero.

R Markdown uses *pandoc* (`http://pandoc.org/`) to build its bibliographies. One nice caveat of this is that you won't have to do a second compile to load in references as standard LaTeX requires. To cite references in your thesis (after creating your bibliography database), place the reference name inside square brackets and precede it by the "at" symbol. For example, here's a reference to a book about worrying: (Molina

[1] footnote text

& Borkovec, 1994). This `Molina1994` entry appears in a file called `thesis.bib` in the `bib` folder. This bibliography database file was created by a program called BibTeX. You can call this file something else if you like (look at the YAML header in the main .Rmd file) and, by default, is to placed in the `bib` folder.

For more information about BibTeX and bibliographies, see Reed College's CUS site (`http://web.reed.edu/cis/help/latex/index.html`)[2]. There are three pages on this topic: *bibtex* (which talks about using BibTeX, at `http://web.reed.edu/cis/help/latex/bibtex.html`), *bibtexstyles* (about how to find and use the bibliography style that best suits your needs, at `http://web.reed.edu/cis/help/latex/bibtexstyles.html`) and *bibman* (which covers how to make and maintain a bibliography by hand, without BibTeX, at `http://web.reed.edu/cis/help/latex/bibman.html`). The last page will not be useful unless you have only a few sources.

If you look at the YAML header at the top of the main .Rmd file you can see that we can specify the style of the bibliography by referencing the appropriate csl file. You can download a variety of different style files at `https://www.zotero.org/styles`. Make sure to download the file into the csl folder.

Tips for Bibliographies

- Like with thesis formatting, the sooner you start compiling your bibliography for something as large as thesis, the better. Typing in source after source is mind-numbing enough; do you really want to do it for hours on end in late April? Think of it as procrastination.
- The cite key (a citation's label) needs to be unique from the other entries.
- When you have more than one author or editor, you need to separate each author's name by the word "and" e.g. `Author = {Noble, Sam and Youngberg, Jessica},`.
- Bibliographies made using BibTeX (whether manually or using a manager) accept LaTeX markup, so you can italicize and add symbols as necessary.
- To force capitalization in an article title or where all lowercase is generally used, bracket the capital letter in curly braces.

[2] Reed College (2007)

Conclusion

If we don't want Conclusion to have a chapter number next to it, we can add the `{.unnumbered}` attribute. This has an unintended consequence of the sections being labeled as 3.6 for example though instead of 4.1. The LaTeX commands immediately following the Conclusion declaration get things back on track.

More info

And here's some other random info: the first paragraph after a chapter title or section head *shouldn't be* indented, because indents are to tell the reader that you're starting a new paragraph. Since that's obvious after a chapter or section title, proper typesetting doesn't add an indent there.

Appendix A

The First Appendix

This first appendix includes all of the R chunks of code that were hidden throughout the document (using the `include = FALSE` chunk tag) to help with readibility and/or setup.

In the main Rmd file:

```
# This chunk ensures that the acstats package is
# installed and loaded. This acstats package includes
# the template files for the thesis and also two functions
# used for labeling and referencing
if(!require(devtools))
  install.packages("devtools", repos = "http://cran.rstudio.com")
if(!require(acstats)){
  library(devtools)
  devtools::install_github("Amherst-Statistics/acstats")
}
library(acstats)
library(knitr)
library(readr)
library(fastICA)
library(ica)
library(gridExtra)
```

In :

```
# This chunk ensures that the acstats package is
# installed and loaded. This acstats package includes
# the template files for the thesis and also two functions
# used for labeling and referencing
if(!require(devtools))
```

```
  install.packages("devtools", repos = "http://cran.rstudio.com")
if(!require(dplyr))
    install.packages("dplyr", repos = "http://cran.rstudio.com")
if(!require(ggplot2))
    install.packages("ggplot2", repos = "http://cran.rstudio.com")
if(!require(acstats)){
  library(devtools)
  devtools::install_github("Amherst-Statistics/acstats")
  }
library(acstats)
flights <- read.csv("data/flights.csv")
```

Appendix B

The Second Appendix, for Fun

References

Helwig, N. E. (2015, August). Independent component analysis. *"Proceedings of" SIGGRAPH 2000*. Retrieved from `https://cran.r-project.org/web/packages/ica/ica.pdf`

Hyvärinen, A., & Oja, E. (2000). Independent component analysis: Algorithms and applications. *"'Independent Component Analysis: Algorithms and Applications' Neural Netw. 200*, 411–30.

Izenman, A. J. (2003). What is independent component analysis?

J L Marchini <marchini@stats.ox.ac.uk>, C. H. <. r, & <ripley@stats.ox.ac.uk>, B. D. R. (2017, June). FastICA algorithms to perform ica and projection pursuit. Retrieved from `https://cran.r-project.org/web/packages/fastICA/fastICA.pdf`

Molina, S. T., & Borkovec, T. D. (1994). The Penn State worry questionnaire: Psychometric properties and associated characteristics. In G. C. L. Davey & F. Tallis (Eds.), *Worrying: Perspectives on theory, assessment and treatment* (pp. 265–283). New York: Wiley.

Puigt, M. (2011). *A very short introduction to blind source separation*. Heraklion, Crete, Greece: Foundation for Research; Technology – Hellas Institute of Computer Science.

Reed College. (2007, March). LaTeX your document. Retrieved from `http://web.reed.edu/cis/help/LaTeX/index.html`