Paper SAS2997-2019

Independent Component Analysis Using the ICA Procedure

Ning Kang, SAS Institute Inc., Cary, NC

ABSTRACT

Independent component analysis (ICA) attempts to extract from observed multivariate data independent components (also called factors or latent variables) that are as statistically independent from each other as possible. You can use ICA to reveal the hidden structure of the data in applications in many different fields, such as audio processing, biomedical signal processing, and image processing. This paper briefly covers the underlying principles of ICA and then discusses how to use the ICA procedure, available in SAS® Visual Statistics 8.3 in SAS® Viya®, to perform independent component analysis.

INTRODUCTION

Independent component analysis (ICA) is a method of finding underlying factors or components from observed multivariate data. The components that ICA looks for are both non-Gaussian and as statistically independent from each other as possible. ICA is one of the most widely used techniques for performing blind source separation, where “source” means an original signal or independent component, and “blind” means that the mixing process of the source signals is unknown and few assumptions about the source signals are made. You can use ICA to analyze the multidimensional data in many fields and applications, including image processing, biomedical imaging, econometrics, and psychometrics. Typical examples of multidimensional data are mixtures of simultaneous speech signals that are recorded by several microphones; brain imaging data from fMRI, MEG, or EEG studies; radio signals that interfere with a mobile phone; or parallel time series that are obtained from some industrial processes.

This paper first defines ICA and discusses its underlying principles. It then introduces the numeric method of estimating the independent components, which the ICA procedure implements. Finally, the paper gives a few examples to demonstrate how to use the ICA procedure to perform independent component analysis.

The ICA procedure is available in SAS Visual Statistics 8.3 in SAS Viya. SAS Viya is the third generation of SAS® software for high-performance in-memory analytics, and the analytic engine in SAS Viya is SAS® Cloud Analytic Services (CAS). Because the SAS Viya statistical procedures were developed specifically for CAS, they enable you to do the following:

- run on a cluster of machines that distribute the data and the computations
- run in single-machine mode
- exploit all the available cores and concurrent threads

INDEPENDENT COMPONENT ANALYSIS

Independent component analysis defines a generative model for the observed multivariate data. Let $\mathbf{X}$ be the data matrix that contains observations of n random variables; that is, $\mathbf{X} = (\mathrm{x}_1, \ldots, \mathrm{x}_n)$, where x_i is the random variable. Assume that the observed data variables are generated as a linear mixture of n independent components (for simplicity, the number of independent components is assumed to be equal to the number of observed data variables),

$$\mathbf{X} = \mathbf{SA}$$

where $\mathbf{A}$ is the unknown mixing matrix and $\mathbf{S} = (\mathrm{s}_1, \ldots, \mathrm{s}_n)$, where s_i is the latent variable or independent component of the observed data. Independent component analysis estimates both matrix $\mathbf{A}$ and matrix $\mathbf{S}$, when only the data matrix $\mathbf{X}$ is observed.

An alternative way to define the ICA model is to find a linear transformation given by a matrix $\mathbf{W}$ as in

$$\mathbf{S} = \mathbf{XW}$$

such that the latent variables or components $\mathrm{s}_i, i = 1, \ldots, n$, are as statistically independent from each other as possible. In this formulation, $\mathbf{W}$, which is called the demixing matrix, is the pseudoinverse of $\mathbf{A}$.

It can be shown that the ICA problem is well defined, and it can be estimated if and only if the latent variables or components $\mathbf{s}_i$ have non-Gaussian distributions. This is the fundamental requirement for the ICA model to be possible. In other words, the mixing matrix $\mathbf{A}$ is not identifiable for Gaussian independent components because any linear combination of Gaussian random variables is itself Gaussian.

Another thing to note is that independence implies uncorrelatedness, but not the other way around. In principal component analysis (PCA) or factor analysis, the data are assumed to have a Gaussian distribution, and the uncorrelated components that they find are always independent. However, if the data do not follow a Gaussian distribution, PCA or factor analysis cannot reveal the underlying factors or sources, because they find components that are uncorrelated, but little more. So, for non-Gaussian data, uncorrelatedness in itself is not enough to find the independent components. That explains the main difference between ICA and PCA or factor analysis, in which the non-Gaussianity of the data is not taken into account.

One approach to estimating the independent components is based on nonlinear decorrelation. If $\mathbf{s}_1$ and $\mathbf{s}_2$ are independent, then any nonlinear transformations $g(\mathbf{s}_1)$ and $h(\mathbf{s}_2)$ are uncorrelated; that is, their covariance is zero. This leads to the method of estimating ICA: finding the demixing matrix $\mathbf{W}$ such that for any $i \neq j$, the components $\mathbf{s}_i$ and $\mathbf{s}_j$ are uncorrelated, and the transformed components $g(\mathbf{s}_i)$ and $h(\mathbf{s}_j)$ are uncorrelated, where g and h are some suitable nonlinear functions.

Another approach to estimating the independent components is based on maximum non-Gaussianity. According to the central limit theorem, a sum of two independent random variables usually has a distribution that is closer to a Gaussian distribution than any of the two original random variables. Therefore, maximizing the non-Gaussianity of a linear combination $\mathbf{y} = \mathbf{X}\mathbf{w}_i = \mathbf{S}(\mathbf{A}\mathbf{w}_i)$ produces one independent component, where $\mathbf{w}_i$ is the ith column vector in the demixing matrix $\mathbf{W}$. This occurs because $\mathbf{y}$ is more Gaussian than any of $\mathbf{s}_i$ according to the central limit theorem; $\mathbf{y}$ becomes least Gaussian when it equals one of $\mathbf{s}_i$. In practice, non-Gaussianity is usually measured by kurtosis or negentropy.

ICA is closely related to projection pursuit, a technique that usually attempts to find the most non-Gaussian projections of multidimensional data. If the ICA model holds for the data, optimizing the non-Gaussianity measures produces independent components; if the model is absent for the data, ICA produces the projection pursuit directions.

COMPUTING INDEPENDENT COMPONENTS

The ICA procedure implements the FastICA algorithm of Hyvärinen and Oja (2000), which applies the principle of maximum non-Gaussianity.

Let $\mathbf{X}$ be the centered and scaled data matrix that is whitened by using the eigenvalue decomposition, and let $\mathbf{w}_i$ be an orthonormal weight vector for the ith independent component. The FastICA algorithm of Hyvärinen and Oja (2000) that the ICA procedure implements is based on a fixed-point iteration scheme for finding a maximum of the non-Gaussianity of $\mathbf{X}\mathbf{w}_i$, which is measured by the approximation of negentropy,

$$J(u) \propto \Big(E[G(u)] - E[G(\nu)]\Big)^2$$

where the random variable u is assumed to be of zero mean and unit variance, ν is a Gaussian variable of zero mean and unit variance, and G is a nonquadratic function. PROC ICA provides the following choices of G as options: $G(u) = -e^{-u^2/2}$ and $G(u) = \log\cosh(u)$. The fixed-point iteration scheme in the FastICA algorithm is derived as an approximative Newton iteration,

$$\mathbf{w}_i \leftarrow E[\mathbf{X}^T g(\mathbf{X}\mathbf{w}_i)] - E[g'(\mathbf{X}\mathbf{w}_i)]\mathbf{w}_i$$

where *g* is the derivative of G.

To prevent the estimated weight vectors $\mathbf{w}_1, \ldots, \mathbf{w}_n$ from converging to the same maxima, the projections $\mathbf{X}\mathbf{w}_1, \ldots, \mathbf{X}\mathbf{w}_n$ must be decorrelated after every iteration. To achieve this, PROC ICA implements two methods: the deflationary decorrelation method and the symmetric decorrelation method. The deflationary method achieves decorrelation on the basis of the Gram-Schmidt orthogonalization method, which means that the independent components are computed sequentially:

$$\mathbf{w}_i \leftarrow \mathbf{w}_i - \sum_{j=1}^{i-1}(\mathbf{w}_j \mathbf{w}_j^T)\mathbf{w}_i$$

A problem with the deflationary decorrelation approach is the accumulation of numeric errors.

The symmetric decorrelation method estimates the independent components simultaneously by performing symmetric orthogonalization of the matrix $\mathbf{W} = (\mathbf{w}_1, \ldots, \mathbf{w}_n)$. This is accomplished by the classic method that involves matrix square roots:

$$\mathbf{W} \leftarrow \mathbf{W}(\mathbf{W}^T\mathbf{W})^{-\frac{1}{2}}$$

The inverse square root is obtained from the eigenvalue decomposition of $\mathbf{W}^T\mathbf{W} = \mathbf{E}\mathbf{D}\mathbf{E}^T$ as $(\mathbf{W}^T\mathbf{W})^{-1/2} = \mathbf{E}\mathbf{D}^{-1/2}\mathbf{E}^T$, where $\mathbf{E}$ is an orthogonal matrix whose entries are the eigenvectors of $\mathbf{W}^T\mathbf{W}$ and $\mathbf{D}$ is a diagonal matrix whose entries are the eigenvalues of $\mathbf{W}^T\mathbf{W}$.

In the ICA model, the variances and the order of the independent components cannot be determined. Each of the independent components is thus assumed to have unit variance in the computation because they are random variables. However, the sign is still indeterminable. By default, PROC ICA calculates the independent component scores by using the scaled input data and the estimated demixing matrix. You can use the NOSCALE option in the PROC ICA statement to suppress scaling; the ICA procedure then calculates the independent component scores by using the original input data and the estimated demixing matrix.

Whitening can greatly simplify the complexity of the problem of independent component analysis by reducing the number of parameters to be estimated. A zero-mean random vector is said to be *white* if its elements are uncorrelated and have unit variances. A synonym for white is sphered.

By default, the input data variables are centered and scaled to have a zero mean and a unit standard deviation. The centered and scaled data matrix $\mathbf{X}$ is then transformed linearly to become a new data matrix $\tilde{\mathbf{X}}$, which is white. In other words, the covariance matrix of $\tilde{\mathbf{X}}$ equals the identity matrix: $E[\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}] = \mathbf{I}$. PROC ICA performs the whitening transformation by using the eigenvalue decomposition of the covariance matrix $E[\mathbf{X}^T\mathbf{X}] = \mathbf{E}\mathbf{D}\mathbf{E}^T$, where $\mathbf{E}$ is the orthogonal matrix of its eigenvectors and $\mathbf{D}$ is the diagonal matrix of its eigenvalues. Thus the whitening transformation matrix is given by $\mathbf{E}\mathbf{D}^{-1/2}$, and whitening is done by $\tilde{\mathbf{X}} = \mathbf{X}\mathbf{E}\mathbf{D}^{-1/2}$.

It can also be useful to reduce the dimensionality of the input data before you extract the independent components; this step can usually reduce noise and prevent overlearning. PROC ICA performs dimension reduction and the whitening transformation at the same time by looking at the eigenvalues of $E[\mathbf{X}^T\mathbf{X}]$ and discarding those that are too small. You can use the EIGTHRESH= option in the PROC ICA statement to specify a threshold for the proportion of variance explained by the eigenvalues. An eigenvalue is discarded if the proportion of variance that it explains is less than the threshold.

When you use the EIGTHRESH= option for dimension reduction, you should usually suppress scaling by using the NOSCALE option in the PROC ICA statement. Scaling places all dimensions of the space on an equal footing relative to their variation in the data. Performing dimension reduction with scaling poses the risk of dropping eigenvalues for the dimensions of the space that are spanned by the independent components but retaining dimensions that are filled by noise.

EXAMPLES USING THE ICA PROCEDURE

Estimating Source Signals

This example uses simulated signal data to illustrate the basic features of the ICA procedure. The simulated signal data are saved in a CAS table named **Signals1**, which are the result of multiplying the source signal matrix $\mathbf{S}$ and the mixing matrix $\mathbf{A}$; that is, the data matrix $\mathbf{X} = \mathbf{S}\mathbf{A}$. Figure 1 shows the original source signals. Figure 2 shows the observed mixtures of the source signals. The problem is to recover the original source signals (**s1**, **s2**, and **s3**) shown in Figure 1 only from the observed signal mixtures (**x1**, **x2**, and **x3**) shown in Figure 2.

Figure 1 Original Source Signals

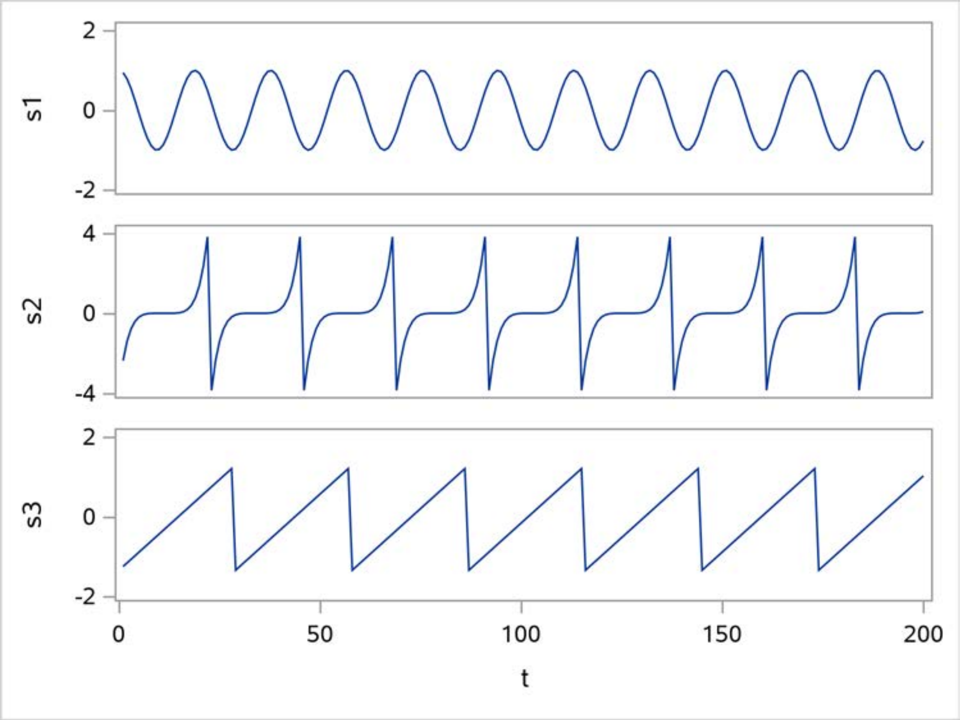
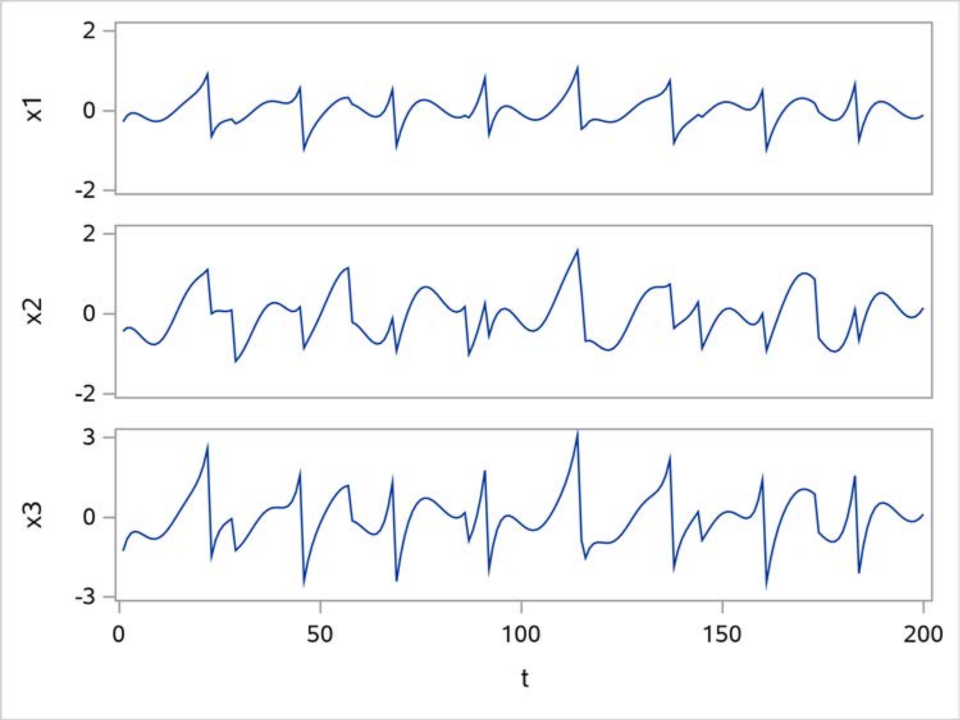


Figure 2 Observed Signal Mixtures



The following statements invoke the ICA procedure, which requests the independent component analysis of the data, outputs the computed independent components to an output data table, and produces the tables in Figure 3 through Figure 6:

```
proc ica data=mycas.Signals1 seed=345;
   var x1-x3;
   output out=mycas.Scores1 component=c copyvar=t;
run;
```

The DATA= option specifies a CAS table named **mycas.Signals1**. The OUT= option in the OUTPUT statement specifies a CAS table named **mycas.Scores1** to store the computed independent components. (The first level of the name is the CAS engine libref, and the second level is the table name.) The VAR statement lists the numeric variables to be analyzed. If you omit the VAR statement, all numeric variables that are not specified in other statements are analyzed.

Figure 3 displays the “Model Information,” “Dimensions,” “Number of Observations,” and “Centering and Scaling Information” tables.

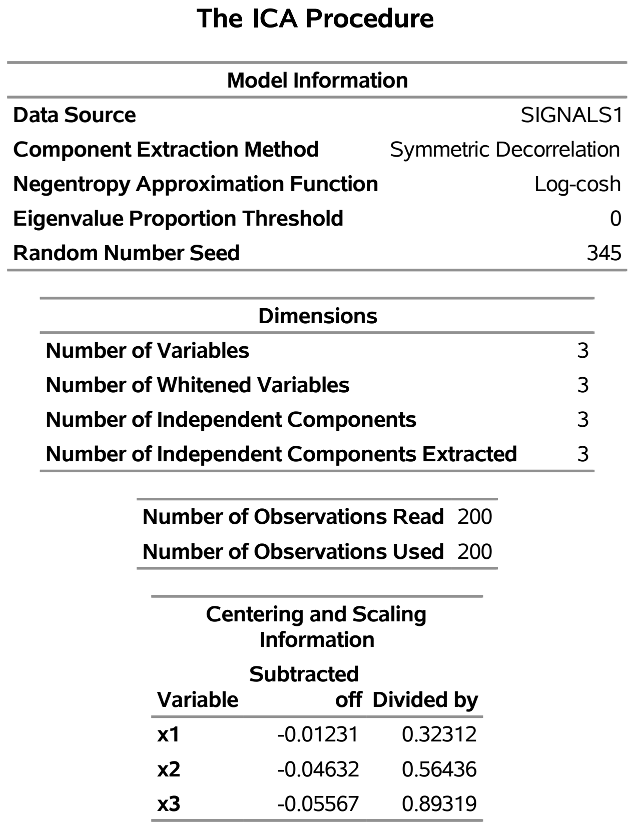
The “Model Information” table identifies the data source and shows that the independent component extraction method is symmetric decorrelation, which is the default. The nonquadratic function $\log \cosh$ is used to approximate negentropy, and the eigenvalue proportion threshold is set to 0; these are the defaults. The random number seed is set to 345. Random number generation is used to initialize the demixing matrix. Changing the random number seed value changes the initial demixing matrix; this yields a different estimated demixing matrix.

The “Dimensions” table indicates that there are three variables to be analyzed and three independent components to be computed. If you omit the N= option in the PROC ICA statement, the default is the number of numeric variables to be analyzed. The table also shows that there are three whitened variables and that three independent components are actually extracted.

The “Number of Observations” table shows that all 200 of the sample observations in the input data are used in the analysis; all the samples are used because they all contain complete data.

The “Centering and Scaling Information” table displays the centering and scaling information of the analysis variables.

Figure 3 Model Information, Dimensions, Number of Observations, and Centering and Scaling Information



The ICA Procedure

Model Information	
Data Source	SIGNALS1
Component Extraction Method	Symmetric Decorrelation
Negentropy Approximation Function	Log-cosh
Eigenvalue Proportion Threshold	0
Random Number Seed	345

Dimensions	
Number of Variables	3
Number of Whitened Variables	3
Number of Independent Components	3
Number of Independent Components Extracted	3

Number of Observations Read	200
Number of Observations Used	200

Centering and Scaling Information

Variable	Subtracted off	Divided by
x1	-0.01231	0.32312
x2	-0.04632	0.56436
x3	-0.05567	0.89319

Figure 4 displays the “Eigenvalues” table.

Figure 4 Eigenvalues Table

Eigenvalues				
	Eigenvalue	**Difference**	**Proportion**	**Cumulative**
1	2.663628	2.352835	0.8879	0.8879
2	0.310793	0.285215	0.1036	0.9915
3	0.025578		0.0085	1.0000

Figure 5 displays the “Whitening Transformation Matrix” and “Dewhitening Transformation Matrix” tables. In the whitening transformation matrix that is shown, the whitened variables are represented as a linear combination of the original variables that are centered and scaled. The dewhitening transformation matrix is the pseudoinverse of the whitening matrix.

Figure 5 Whitening and Dewhitening Transformation Matrices

Whitening Transformation Matrix			
Variable	**White1**	**White2**	**White3**
x1	-0.35446	-1.01630	3.66902
x2	-0.33541	1.44039	1.47348
x3	-0.37052	-0.33165	-4.84386

Dewhitening Transformation Matrix			
Whitened Variable	**x1**	**x2**	**x3**
White1	-0.94415	-0.89341	-0.98693
White2	-0.31586	0.44766	-0.10307
White3	0.09385	0.03769	-0.12390

Figure 6 displays the “Demixing Matrix” and “Mixing Matrix” tables. In the demixing matrix that is shown, the independent components are represented as a linear combination of the original variables that are standardized. The mixing matrix is the pseudoinverse of the demixing matrix.

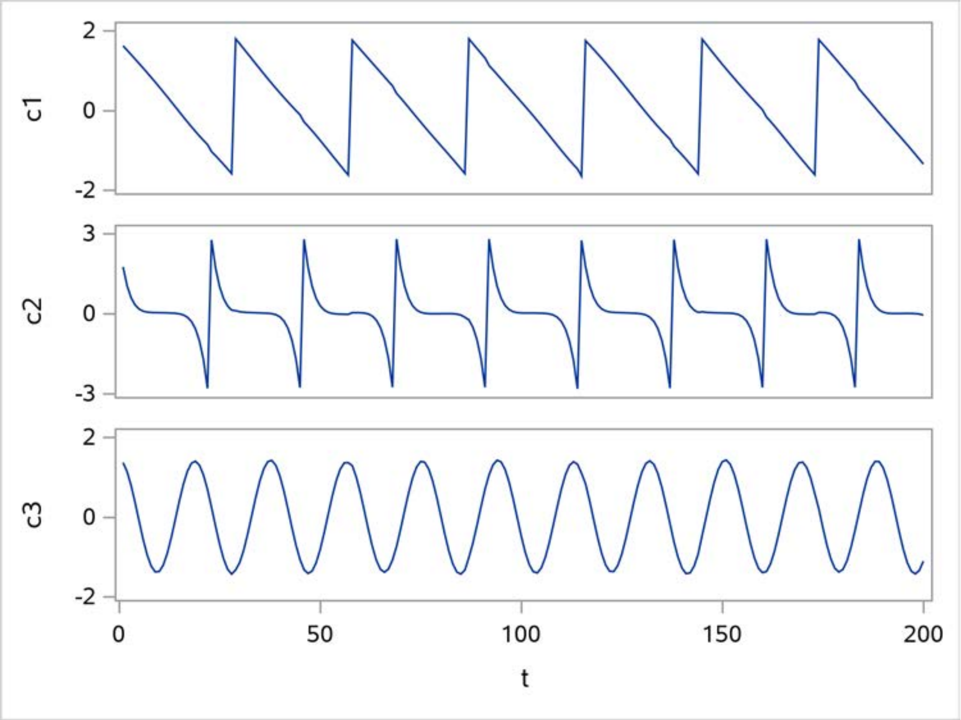
Figure 6 Demixing and Mixing Matrices

Demixing Matrix			
Variable	**Comp1**	**Comp2**	**Comp3**
x1	2.51490	0.32078	2.86227
x2	-0.43425	1.17723	1.66851
x3	-2.44766	-2.07527	-3.66231

Mixing Matrix			
Component	**x1**	**x2**	**x3**
Comp1	-0.12351	-0.69340	-0.41244
Comp2	-0.82569	-0.32078	-0.79147
Comp3	0.55043	0.64520	0.45108

Using the output data table **mycas.Scores1** and the SGPLOT or SGRENDER procedure, you can plot the computed independent components. Figure 7 shows three independent components (**c1**, **c2**, and **c3**) that are the estimates of the three original source signals (**s1**, **s2**, and **s3**) in Figure 1. The original source signals are accurately estimated from the observed signal mixtures shown in Figure 2, up to multiplicative signed scalars. The order of the computed independent components cannot be determined in the ICA model.

Figure 7 Estimated Source Signals



Estimating Source Signals with Dimension Reduction

This example demonstrates that applying dimension reduction in PROC ICA provides a better estimate of the independent components than you get by extracting the independent components without reducing the dimension of the data. The data that this example uses are from the example on page 3, and an extra noise signal (**x4**) is added to the observed signal mixtures. The signal data are saved in a CAS table named **Signals2**.

The following statements extract the independent components by using dimension reduction and output the computed independent components to an output data table:

```
proc ica data=mycas.Signals2 eigthresh=0.004 noscale seed=345;
   var x1-x4;
   output out=mycas.Scores2 component=c copyvar=t;
run;
```

The EIGTHRESH= option specifies a threshold for the proportion of variance explained by the eigenvalues. An eigenvalue is discarded if the proportion of variance that it explains is less than the threshold. You can use this option to reduce the dimensionality of the input data. When you use it for dimension reduction, you should usually suppress scaling by using the NOSCALE option at the same time. Performing dimension reduction with scaling poses the risk of dropping eigenvalues for the dimensions of the space that are spanned by the independent components but retaining dimensions that are filled by noise.

Figure 8 displays the PROC ICA output. The “Model Information” table shows that the eigenvalue proportion threshold is set to 0.004. The “Dimensions” table indicates that there are four variables to be analyzed and four independent components to be computed. It also shows that three whitened variables are generated by using whitening and three independent components are actually extracted, thanks to the use of dimension reduction.

Figure 8 Results of Independent Component Analysis with Dimension Reduction

The ICA Procedure

Model Information	
Data Source	SIGNALS2
Component Extraction Method	Symmetric Decorrelation
Negentropy Approximation Function	Log-cosh
Eigenvalue Proportion Threshold	0.004
Random Number Seed	345

Dimensions	
Number of Variables	4
Number of Whitened Variables	3
Number of Independent Components	4
Number of Independent Components Extracted	3

Number of Observations Read	200
Number of Observations Used	200

Centering and Scaling Information

Variable	Subtracted off	Divided by
x1	-0.01231	1.00000
x2	-0.04632	1.00000
x3	-0.05567	1.00000
x4	0.05224	1.00000

Eigenvalues

	Eigenvalue	Difference	Proportion	Cumulative
1	1.129934	1.045020	0.9251	0.9251
2	0.084915	0.079040	0.0695	0.9946
3	0.005874	0.005123	0.0048	0.9994
4	0.000752		0.0006	1.0000

Whitening Transformation Matrix

Variable	White1	White2	White3
x1	-0.26624	1.12655	11.73306
x2	-0.44684	-2.96560	2.16685
x3	-0.78387	1.30782	-5.22151
x4	-0.00124	0.04711	0.78387

Dewhitening Transformation Matrix

Whitened Variable	x1	x2	x3	x4
White1	-0.30083	-0.50490	-0.88572	-0.00140
White2	0.09566	-0.25182	0.11105	0.00400
White3	0.06892	0.01273	-0.03067	0.00460

Figure 8 *continued*

Demixing Matrix

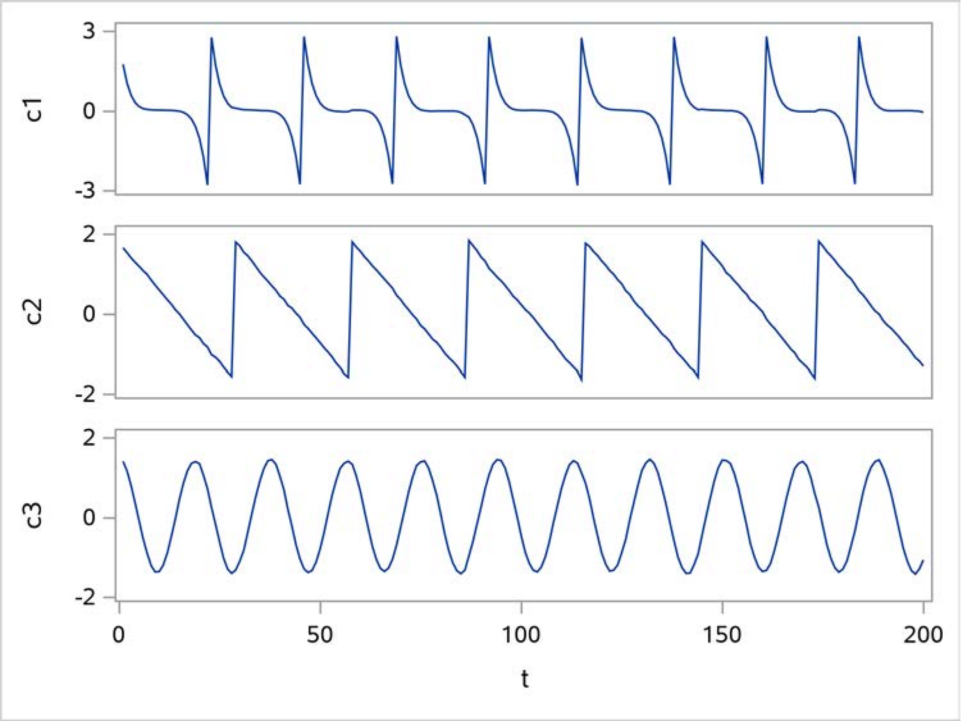
Variable	Comp1	Comp2	Comp3
x1	0.96285	7.75333	8.82969
x2	2.08133	-0.76730	2.96125
x3	-2.31137	-2.73321	-4.09587
x4	0.09540	0.50802	0.59118

Mixing Matrix

Component	x1	x2	x3	x4
Comp1	-0.26709	-0.18133	-0.70731	-0.00299
Comp2	-0.04024	-0.39162	-0.36922	0.00468
Comp3	0.17732	0.36367	0.40146	0.00289

You can produce a plot of the computed independent components by using the output data table **mycas.Scores2** and PROC SGRENDER. Figure 9 displays the computed independent components. The original source signals (**s1**, **s2**, and **s3**) in Figure 1 are accurately estimated by the computed independent components (**c1**, **c2**, and **c3**) up to multiplicative signed scalars.

Figure 9 Independent Components Computed with Dimension Reduction

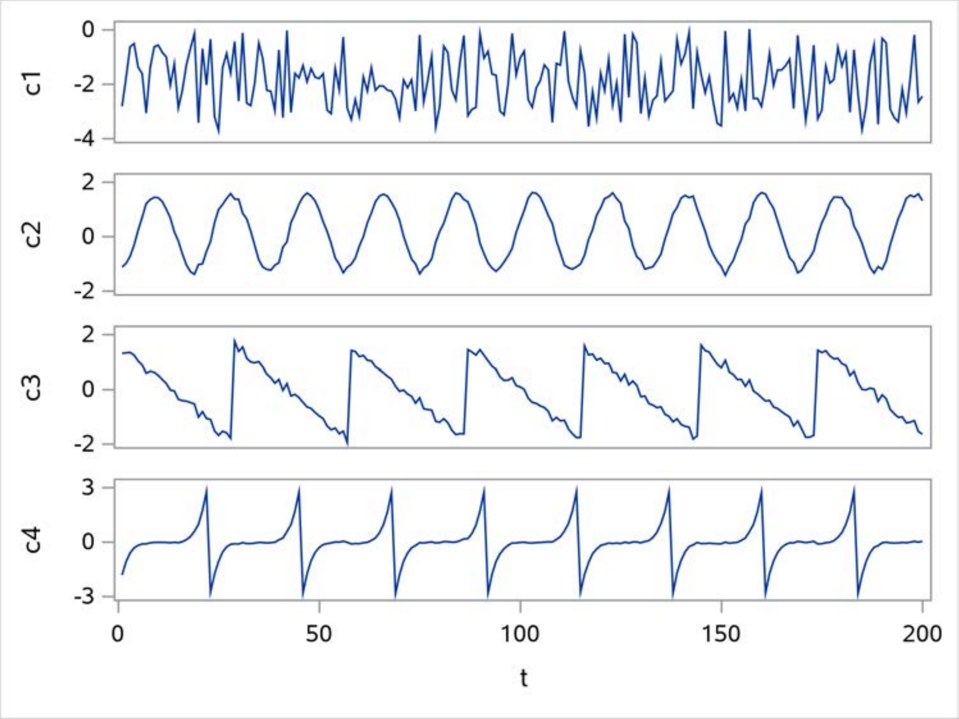


For comparison, the following statements extract the independent components by using all dimensions of the input data (**x1**–**x4**) and output the computed independent components to an output data table:

```
proc ica data=mycas.Signals2 noscale seed=345;
   var x1-x4;
   output out=mycas.Scores2a component=c copyvar=t;
run;
```

Figure 10 displays the computed independent components by using the output data table **mycas.Scores2a** and PROC SGRENDER. The original source signals (**s1**, **s2**, and **s3**) in Figure 1 are closely estimated by the three computed independent components (**c2**, **c3**, and **c4**) up to multiplicative signed scalars. However, you can clearly see that the estimates are more affected by the addition of the noise signal (**x4**) than the independent components that are computed using dimension reduction in Figure 9. Because there are fewer independent components than analysis variables in the input data, the computed component **c1** is the projection pursuit direction.

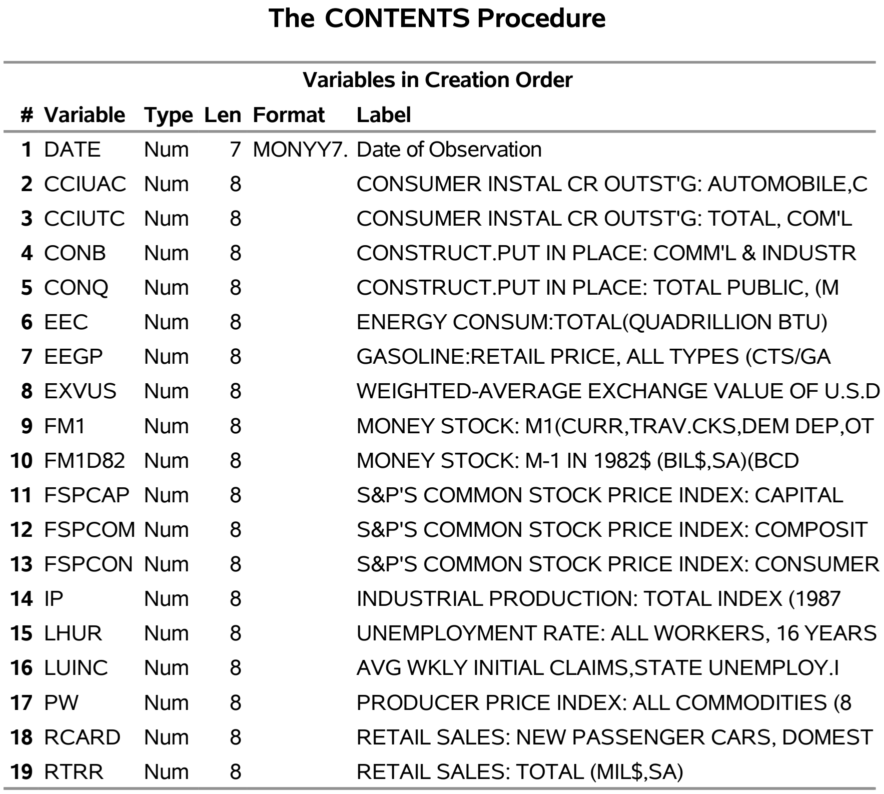
Figure 10 Independent Components Computed with Full Dimensions



Finding Underlying Factors in Macroeconomic Data

This example uses PROC ICA to explore whether independent component analysis can reveal the underlying structure of macroeconomic data. Such hidden factors could include news, major events, and unexplained noise. The data that this example uses are from the Sashelp data set **Sashelp.citimon**, which provides Citibase monthly macroeconomic indicators from January 1980 to January 1992. Figure 11 displays variable information in the data set, which shows that there are 18 monthly indicators. Figure 12 depicts the time series data.

Figure 11 Sashelp.citimon—Variable Information

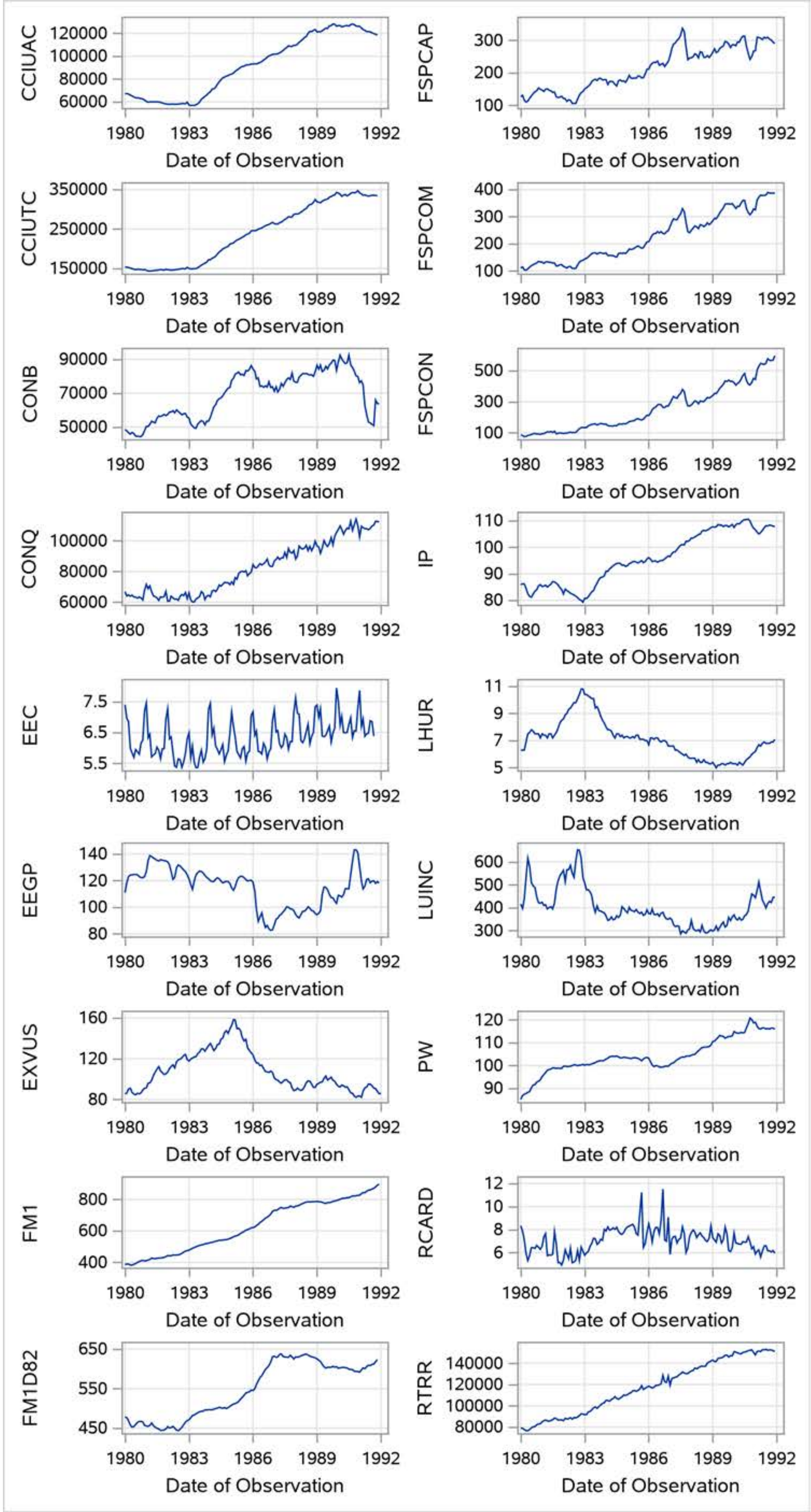


The CONTENTS Procedure

Variables in Creation Order

#	Variable	Type	Len	Format	Label
1	DATE	Num	7	MONYY7.	Date of Observation
2	CCIUAC	Num	8		CONSUMER INSTAL CR OUTST'G: AUTOMOBILE,C
3	CCIUTC	Num	8		CONSUMER INSTAL CR OUTST'G: TOTAL, COM'L
4	CONB	Num	8		CONSTRUCT.PUT IN PLACE: COMM'L & INDUSTR
5	CONQ	Num	8		CONSTRUCT.PUT IN PLACE: TOTAL PUBLIC, (M
6	EEC	Num	8		ENERGY CONSUM:TOTAL(QUADRILLION BTU)
7	EEGP	Num	8		GASOLINE:RETAIL PRICE, ALL TYPES (CTS/GA
8	EXVUS	Num	8		WEIGHTED-AVERAGE EXCHANGE VALUE OF U.S.D
9	FM1	Num	8		MONEY STOCK: M1(CURR,TRAV.CKS,DEM DEP,OT
10	FM1D82	Num	8		MONEY STOCK: M-1 IN 1982$ (BIL$,SA)(BCD
11	FSPCAP	Num	8		S&P'S COMMON STOCK PRICE INDEX: CAPITAL
12	FSPCOM	Num	8		S&P'S COMMON STOCK PRICE INDEX: COMPOSIT
13	FSPCON	Num	8		S&P'S COMMON STOCK PRICE INDEX: CONSUMER
14	IP	Num	8		INDUSTRIAL PRODUCTION: TOTAL INDEX (1987
15	LHUR	Num	8		UNEMPLOYMENT RATE: ALL WORKERS, 16 YEARS
16	LUINC	Num	8		AVG WKLY INITIAL CLAIMS,STATE UNEMPLOY.I
17	PW	Num	8		PRODUCER PRICE INDEX: ALL COMMODITIES (8
18	RCARD	Num	8		RETAIL SALES: NEW PASSENGER CARS, DOMEST
19	RTRR	Num	8		RETAIL SALES: TOTAL (MIL$,SA)

Figure 12 Sashelp.citimon—Citibase Monthly Indicators: Jan80–Jan92



The following statements perform the independent component analysis of the macroeconomic data and output the computed independent components to an output data table:

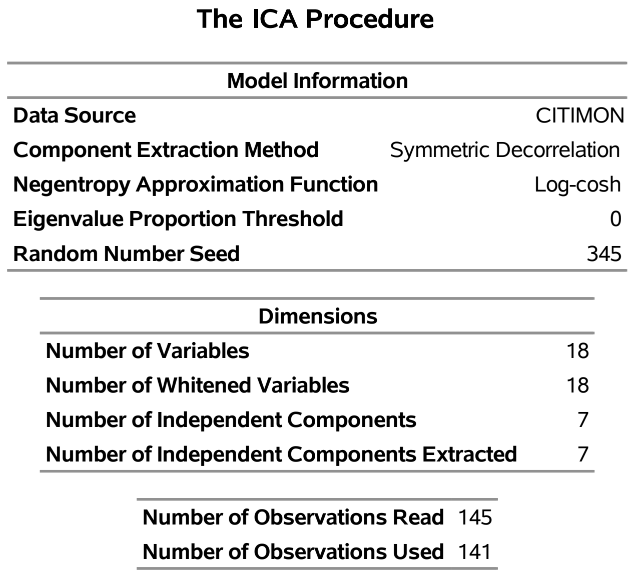
```
data mycas.citimon;
   set sashelp.citimon;
run;

proc ica data=mycas.citimon n=7 seed=345;
   var CCIUAC CCIUTC CONB CONQ
       EEC EEGP EXVUS
       FM1 FM1D82 FSPCAP FSPCOM FSPCON
       IP LHUR LUINC PW RCARD RTRR;
   output out=mycas.Scores3 component=c copyvar=DATE;
run;
```

The DATA step creates a CAS table named **mycas.citimon**, which loads the data set **Sashelp.citimon** into your CAS session. The N= option specifies the number of independent components to be estimated; the analysis assumes that there are seven hidden factors in the ICA mixing model. The variables that the VAR statement lists are centered and scaled in the analysis to have mean 0 and standard deviation 1, which are the defaults in PROC ICA.

Figure 13 displays the "Model Information," "Dimensions," and "Number of Observations" tables. The "Dimensions" table indicates that all 18 monthly indicators are used as input variables to be analyzed and seven independent components are extracted.

Figure 13 Model Information, Dimensions, and Number of Observations



The ICA Procedure

Model Information	
Data Source	CITIMON
Component Extraction Method	Symmetric Decorrelation
Negentropy Approximation Function	Log-cosh
Eigenvalue Proportion Threshold	0
Random Number Seed	345

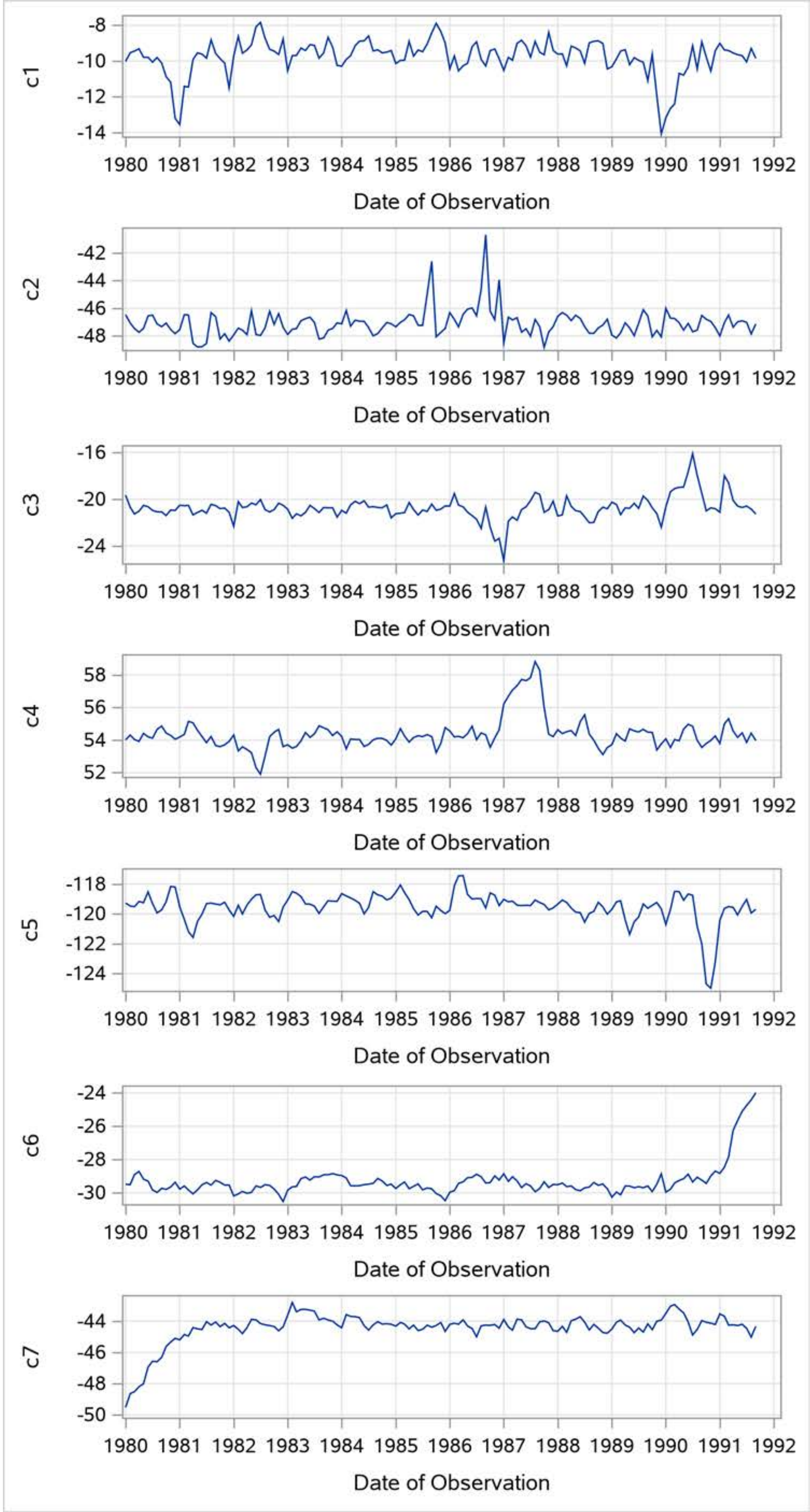
Dimensions	
Number of Variables	18
Number of Whitened Variables	18
Number of Independent Components	7
Number of Independent Components Extracted	7

Number of Observations Read	145
Number of Observations Used	141

You can produce a plot of the computed independent components by using the output data table **mycas.Scores3** and the SGRENDER procedure. Figure 14 displays the seven extracted independent components. These independent components can be seen as very distinct thrust signals that constitute the underlying structure of the macroeconomic indicators. The infrequent but large and abrupt changes depict the impact of the news or major events on the monthly indicators, which are responsible for the major shifts in the indicators. The frequent but smaller fluctuations contribute much less to the overall levels of the monthly indicators.

The computed independent component **c4** shows a large and sudden bump in 1987 that is caused by the stock market crash that year. The independent component **c5** shows a large and sudden drop in 1990, and the independent component **c3** also shows a large thrust that same year; both sudden changes most likely reflect the impact of the Gulf War in 1990. For other computed independent components, the large and sudden changes in certain years can have different interpretations, depending on the major events of that year.

Figure 14 Independent Components Extracted from Citibase Monthly Indicators



REFERENCES

Hyvärinen, A., Karhunen, J., and Oja, E. (2001). *Independent Component Analysis*. New York: John Wiley & Sons.

Hyvärinen, A., and Oja, E. (2000). “Independent Component Analysis: Algorithms and Applications.” *Neural Networks* 13:411–430.

ACKNOWLEDGMENTS

The author thanks Ed Huddleston for his editorial assistance.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author:

Ning Kang
SAS Institute Inc.
SAS Campus Drive
Cary, NC 27513
ning.kang@sas.com