

Stat 8053, Chapter 14 Canonical Correlation Analysis, rev January 23, 2012

Let's begin with a small artificial example in which X and Y each have two components:

```
> sxx <- matrix(c(1.,.7,.7,1),nrow=2)
> syy <- matrix(c(1,.6,.6,1),nrow=2)
> sxy <- matrix(c(.5,.6,.3,.4),nrow=2,byrow=T)
> (R <- cbind( rbind(sxx,sxy), rbind(t(sxy),syy)))
```

```
      [,1] [,2] [,3] [,4]
[1,]  1.0  0.7  0.5  0.3
[2,]  0.7  1.0  0.6  0.4
[3,]  0.5  0.6  1.0  0.6
[4,]  0.3  0.4  0.6  1.0
```

```
> eigen(R)$values
```

```
[1] 2.5633 0.8049 0.3495 0.2823
```

Here X and Y have been standardized so the covariance matrices are in correlation form. There is modest correlation, and one relatively large eigenvalue.

To do canonical correlation analysis, we need to get $\Sigma_{xx}^{-1/2}$ and $\Sigma_{yy}^{-1/2}$, for which we can use the spectral decomposition.

```
> eigen.sxx <- eigen(sxx)
> eigen.syy <- eigen(syy)
> (sxxinvsqrt <- eigen.sxx$vectors %*% diag(1/sqrt(eigen.sxx$values)) %*%
+      t(eigen.sxx$vectors))
```

```
      [,1] [,2]
[1,]  1.2964 -0.5294
[2,] -0.5294  1.2964
```

```
> (syyinvsqrt <- eigen.syy$vectors %*% diag(1/sqrt(eigen.syy$values)) %*%
+      t(eigen.syy$vectors))
```

```
      [,1] [,2]
[1,]  1.1859 -0.3953
[2,] -0.3953  1.1859
```

Finally, we need to compute $\Sigma_{xx}^{-1/2}\Sigma_{xy}\Sigma_{yy}^{-1/2}$,

```
> (mat <- sxxinvsqrt %*% sxy %*% syyinvsqrt)
```

```
      [,1] [,2]
[1,] 0.35656 0.4778
[2,] 0.06788 0.1891
```

```
> (sv <- svd(mat))
```

```
$d
[1] 0.62667 0.05586
```

```
$u
      [,1] [,2]
[1,] -0.9510 -0.3093
[2,] -0.3093  0.9510
```

```
$v
      [,1] [,2]
[1,] -0.5746 -0.8185
[2,] -0.8185  0.5746
```

The singular values are the canonical correlations. The first canonical correlation of 0.63 is large relative to the second correlation of 0.056. Consequently, most of the information about the dependence between X and Y is contained in the first canonical variate.

The canonical variates for the (standardized) X and Y are given by $u'\Sigma_{xx}^{-1/2}$ and $v'\Sigma_{yy}^{-1/2}$, respectively. We transpose the results before printing, so the canonical vectors are column vectors:

```
> t(t(sv$u) %*% sxxinvsqrt)
```

```
      [,1] [,2]
[1,] -1.0691 -0.9044
[2,]  0.1025  1.3965
```

```
> t(t(sv$v) %*% syyinvsqrt)
```

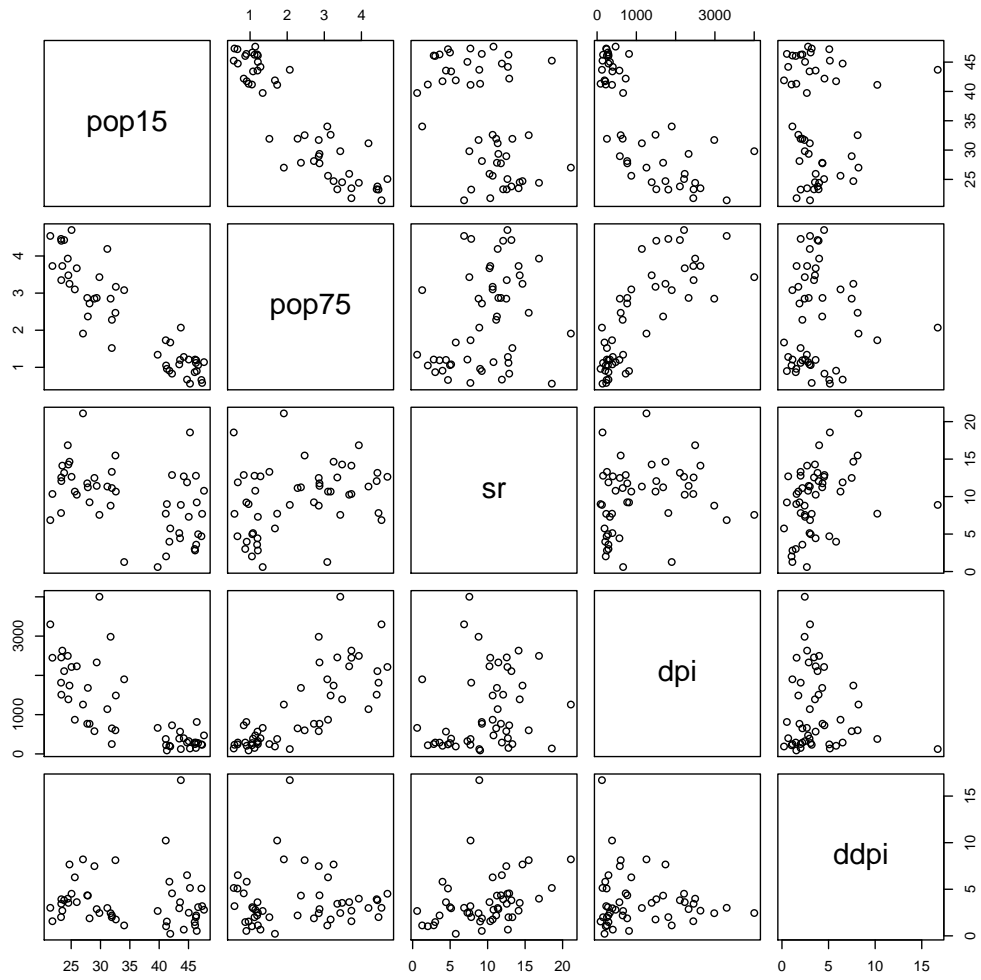
```
      [,1] [,2]
[1,] -0.3578 -1.198
[2,] -0.7434  1.005
```

The first canonical variate for X is nearly the first column of X , and the second canonical vector is more or less the difference between the columns. For Y , the first canonical vector is a weighted sum with the second column given double weight.

Life Cycle Savings

The `LifeCycleSavings` data in the `data` package provide data on aggregate personal savings in 50 countries. `Pop15` and `Pop75` are population variables (% of population under 15 and % of population over 75), and the other three variables are economic (aggregate savings `sr`, per-capita disposable income `dpi`, and % growth rate of disposable income `ddpi`). We can fit canonical correlation as we have done above, or we can use the `cancor` function in the `stats` package.

```
> pairs(LifeCycleSavings[,c(2,3,1,4,5)])
```



```
> pop <- LifeCycleSavings[, 2:3]
> oec <- LifeCycleSavings[, -(2:3)]
> (c1<-cancor(pop, oec))
```

```
$cor
[1] 0.8248 0.3653
```

```
$xcoef
      [,1]      [,2]
pop15 -0.009111 -0.03622
pop75  0.048648 -0.26031
```

```
$ycoef
      [,1]      [,2]      [,3]
sr    0.0084710  3.338e-02 -5.157e-03
dpi   0.0001307 -7.588e-05  4.544e-06
ddpi  0.0041706 -1.227e-02  5.188e-02
```

```
$xcenter
```

```
pop15 pop75
35.090 2.293
```

```
$ycenter
      sr      dpi      ddpi
9.671 1106.758 3.758
```

Repeat the analysis, but in correlation scale:

```
> (c2<- cancor(scale(pop),scale(oec)))
```

```
$cor
[1] 0.8248 0.3653
```

```
$xcoef
      [,1] [,2]
pop15 -0.08338 -0.3315
pop75 0.06279 -0.3360
```

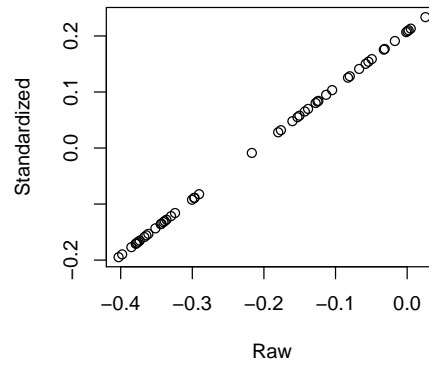
```
$ycoef
      [,1] [,2] [,3]
sr 0.03795 0.14955 -0.023106
dpi 0.12955 -0.07519 0.004502
ddpi 0.01197 -0.03521 0.148898
```

```
$xcenter
      pop15      pop75
3.073e-16 -8.941e-17
```

```
$ycenter
      sr      dpi      ddpi
1.538e-16 7.272e-17 -1.790e-17
```

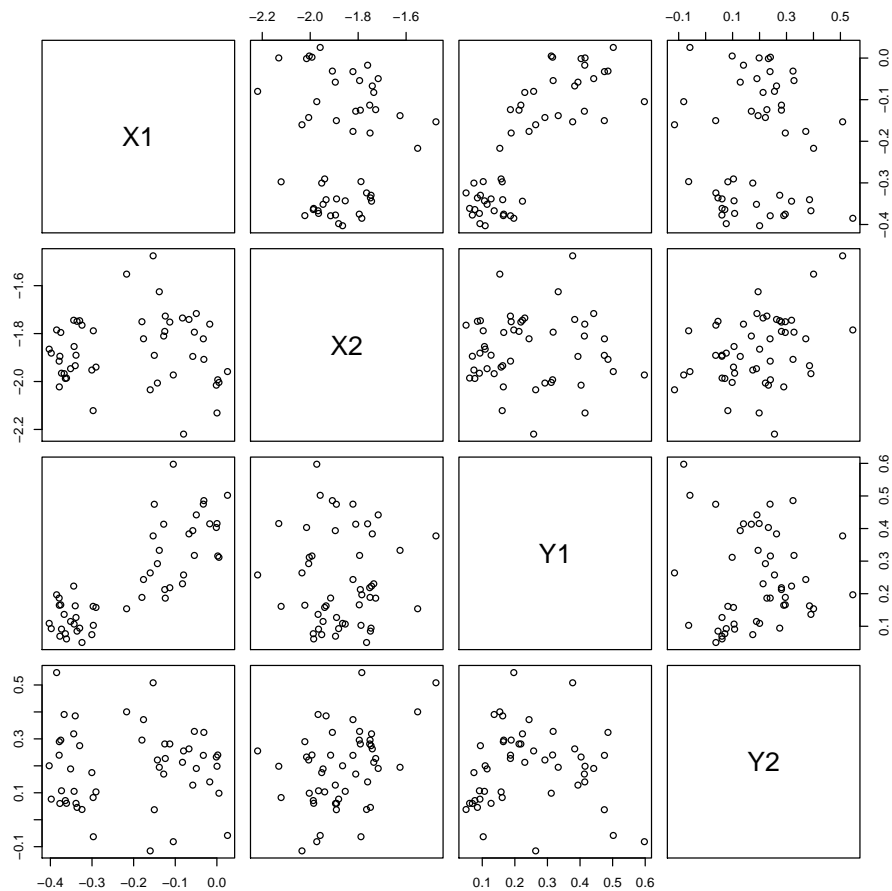
The canonical correlations are the same for both fits, so unlike many of the methods we have been studying recently the canonical correlations are invariant under rescaling, although the canonical variates differ by a scale:

```
> plot(as.matrix(pop) %*% c1$xcoef[,1],
+      as.matrix(scale(pop))%*% c2$xcoef[,1],xlab="Raw",ylab="Standardized")
```



The canonical vectors, (X_1, X_2) and (Y_1, Y_2) are uncorrelated by construction. The first canonical correlation is large, and so the plot of X_1 versus Y_1 should, and does, display a linear relationship, which we will examine more carefully in the next plot.

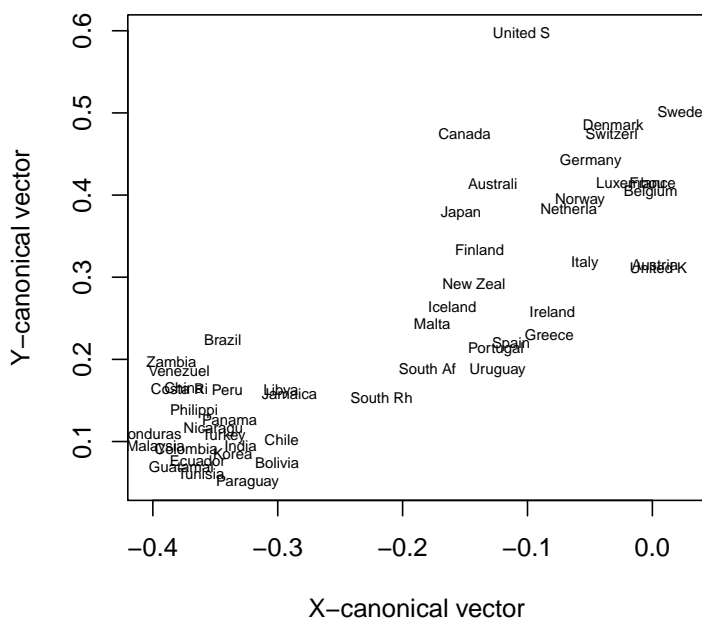
```
> d <- cbind(as.matrix(pop) %*% c1$xcoef, as.matrix(oec) %*% c1$ycoef[,1:2])
> colnames(d) <- c("X1", "X2", "Y1", "Y2")
> pairs(d)
```



```

> plot( as.matrix(pop) %%% c1$xcoef[,1],
+       as.matrix(oec) %%% c1$ycoef[,1],type="n",
+       xlab="X-canonical vector",ylab="Y-canonical vector")
> text( as.matrix(pop) %%% c1$xcoef[,1],
+       as.matrix(oec) %%% c1$ycoef[,1],substr(rownames(oec),1,8),cex=.6)

```



This may suggest that countries generally cluster into two groups: The lower-left less-developed countries that are very similar on both the population and economic variables and the upper-right more developed countries that are similar on the population variables but variable on the economic ones.

Tests

Under normality, test of the hypothesis that X is independent of Y is rejected if the canonical correlations λ_i are too large in absolute value. The test statistic is

$$-(n - (p + q + 3)/2) \log \left[\prod_{i=1}^k (1 - \lambda_i^2) \right] \sim \chi^2(pq)$$

where $k = \min(p, q)$. The `cancor` function is very primitive, and does not compute this test for you:

```

> n <- dim(pop)[1]
> p <- 2
> q <- 3
> (test <- -(n-(p+q+3)/2) *log(prod(1-c1$cor^2)))

```

```
[1] 59.04
```

```
> pchisq(test,p*q,lower.tail=FALSE)
```

```
[1] 7.04e-11
```

The hypothesis of independence is rejected. The hypothesis of exactly one non-zero canonical correlation is computed as

$$-(n - (p + q + 3)/2) \log \left[\prod_{i=2}^k (1 - \lambda_i^2) \right] \sim \chi^2((p - 1)(q - 1))$$

where $k = \min(p, q)$.

```
> (test <- -(n-(p+q+3)/2) *log(prod(1-c1$cor[-1]^2)))
```

```
[1] 6.588
```

```
> pchisq(test,(p-1)*(q-1),lower.tail=FALSE)
```

```
[1] 0.03711
```

This provides modest evidence that the second “population” canonical correlation is non-zero.

Seemingly Unrelated Regressions

We imagine an $n \times a$ response matrix $Y = (Y_1, \dots, Y_q)$, and for each component of the response we have a linear model,

$$\begin{aligned} Y_1 &= X_1\beta_1 + \varepsilon_1 \\ Y_2 &= X_2\beta_2 + \varepsilon_2 \\ &\vdots \\ Y_q &= X_q\beta_q + \varepsilon_q \end{aligned}$$

where Y_j is a vector of length n , X_j has p_j columns (we assume $p_j = p$ below), and $\text{Var}(\varepsilon_j) = \sigma^2 I_n$. The important feature here is that the Y s are measured on the **same n units** and hence they may be correlated. Suppose that the correlation matrix for the q observations on the same subject is $\text{Cov}(y_i|X) = \Sigma$. Let

$$\tilde{\beta} = (X_j'X_j)^{-1}X_j'Y$$

be the $p \times q$ matrix of the multivariate OLS estimator for $(\beta_1, \dots, \beta_q)$.

Although we usually treat this as q separate regressions, but we can combine them into a single regression as follows. Let \mathbf{Y} be the $qn \times 1$ vector obtained by stacking the responses, and write

$$\mathbf{Y} = \begin{pmatrix} X_1 & 0 & 0 & \cdots & 0 \\ 0 & X_2 & 0 & \cdots & 0 \\ 0 & 0 & X_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & 0 & \cdots & X_q \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \vdots \\ \beta_q \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \vdots \\ \varepsilon_q \end{pmatrix}$$

The variance of \mathbf{Y} is given by $\text{Var}(\mathbf{Y}) = \Omega = \Sigma \otimes I_n$ where Σ is the $q \times q$ covariance matrix of the q responses. If A is an $m \times n$ matrix and B is a $p \times q$ matrix, then the Kronecker product $A \otimes B$ is the $mp \times nq$ block matrix with the blocks are given by $a_{ij}B$. Thus we have

$$\Omega = \begin{pmatrix} \sigma_{11}I_n & \sigma_{12}I_n & \cdots & \sigma_{1q}I_n \\ \sigma_{12}I_n & \sigma_{22}I_n & \cdots & \sigma_{2q}I_n \\ \vdots & \vdots & \cdots & \vdots \\ \sigma_{1q}I_n & \sigma_{2q}I_n & \cdots & \sigma_{11}I_n \end{pmatrix}$$

The non-zero off-diagonal elements of Ω provide the extra information that makes this methodology provide different answers than OLS.

The generalized least squares estimator of $\beta = (\beta_1', \dots, \beta_q)'$, assuming Σ known, is

$$\hat{\beta} = (X'\Omega^{-1}X)^{-1}X'\Omega^{-1}Y$$

This estimator is not the same as the OLS estimator unless either $\Sigma = \sigma^2 I_q$ or $X = I_n \otimes X_1$, equivalent to $X_1 = X_2 = \dots = X_q$, all the design matrices the same. The proof of these results requires only substituting into the last displayed equation, and then using the definition of Kronecker products.

Assuming neither of the conditions of the last paragraph are satisfied, this is called the *seemingly unrelated regressions* or SUR problem: to estimate the coefficients for the regression of a response of interest on a set of predictors, we should take into account the regression of other responses on (other) predictors for the same experimental units, potentially *even if the other responses are not of interest*.

A practical problem is that Σ is unknown, and must be estimated from data in some way. Of course if the estimate is poor, the SUR estimates can be worse than OLS, at least in some problems. The simplest estimate is to compute ols residuals $e_j = Y_j - X_j\tilde{\beta}_j$, and then an estimate of the (j, j') element of Σ is given by $\sum e_{ji}e_{j'i}/n$. An estimator is obtained by substituting this estimate for Σ (or perhaps some other estimate, or using an iterative procedure in its place) to get estimates.

The following demonstration uses an example taken from Kmenta, J. (1986), *Elements of Econometrics*, Second Edition, Macmillan, New York, p. 685. We want to estimate a small model of the US food market, which has a *demand* equation,

$$\text{consump} = \beta_{10} + \beta_{11}\text{price} + \beta_{12}\text{income} + \varepsilon_1$$

and also a *supply* equation,

$$\text{consump} = \beta_{20} + \beta_{21}\text{price} + \beta_{22}\text{farmPrice} + \beta_{23}\text{trend} + \varepsilon_2$$

Variable *consump* (food consumption per capita) is the dependent variable in each of the equations. The predictors are price (ratio of food prices to general consumer prices) and income (disposable income) as well as a constant. The supply side of the food market also uses *consump* as the dependent variable, but different predictors, including *farmPrice* (ratio of preceding year's prices received by farmers to general consumer prices) and a time *trend*; thus the errors in the two equations for the same time period are correlated. The `systemfit` package estimates using both OLS and SUR:

```
> library("systemfit")
> data(Kmenta)
> eqDemand <- consump ~ price + income
> eqSupply <- consump ~ price + farmPrice + trend
> eqSystem <- list(demand = eqDemand, supply = eqSupply)
> summary(fitols <- systemfit(eqSystem, data=Kmenta))
```

systemfit results

method: OLS

	N	DF	SSR	detRCov	OLS-R2	McElroy-R2
system	40	33	156	4.43	0.709	0.558

	N	DF	SSR	MSE	RMSE	R2	Adj R2
demand	20	17	63.3	3.73	1.93	0.764	0.736
supply	20	16	92.6	5.78	2.40	0.655	0.590

The covariance matrix of the residuals

	demand	supply
demand	3.73	4.14
supply	4.14	5.78

The correlations of the residuals

	demand	supply
demand	1.000	0.891
supply	0.891	1.000

OLS estimates for 'demand' (equation 1)

Model Formula: `consump ~ price + income`

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	99.8954	7.5194	13.29	2.1e-10
price	-0.3163	0.0907	-3.49	0.0028
income	0.3346	0.0454	7.37	1.1e-06

Residual standard error: 1.93 on 17 degrees of freedom

Number of observations: 20 Degrees of Freedom: 17

SSR: 63.332 MSE: 3.725 Root MSE: 1.93

Multiple R-Squared: 0.764 Adjusted R-Squared: 0.736

OLS estimates for 'supply' (equation 2)

Model Formula: `consump ~ price + farmPrice + trend`

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	58.2754	11.4629	5.08	0.00011
price	0.1604	0.0949	1.69	0.11039
farmPrice	0.2481	0.0462	5.37	6.2e-05
trend	0.2483	0.0975	2.55	0.02157

Residual standard error: 2.405 on 16 degrees of freedom

Number of observations: 20 Degrees of Freedom: 16

SSR: 92.551 MSE: 5.784 Root MSE: 2.405

Multiple R-Squared: 0.655 Adjusted R-Squared: 0.59

```
> summary(fitsur <- systemfit(eqSystem, method = "SUR", data=Kmenta))
```

systemfit results

method: SUR

	N	DF	SSR	detRCov	OLS-R2	McElroy-R2
system	40	33	170	0.879	0.683	0.789

	N	DF	SSR	MSE	RMSE	R2	Adj R2
demand	20	17	65.7	3.86	1.97	0.755	0.726
supply	20	16	104.1	6.50	2.55	0.612	0.539

The covariance matrix of the residuals used for estimation

	demand	supply
demand	3.73	4.14
supply	4.14	5.78

The covariance matrix of the residuals

	demand	supply
--	--------	--------

```
demand 3.86 4.92
supply 4.92 6.50
```

The correlations of the residuals

```
      demand supply
demand 1.000 0.982
supply 0.982 1.000
```

SUR estimates for 'demand' (equation 1)

Model Formula: `consump ~ price + income`

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	99.3329	7.5145	13.22	2.3e-10
price	-0.2755	0.0885	-3.11	0.0063
income	0.2986	0.0419	7.12	1.7e-06

Residual standard error: 1.966 on 17 degrees of freedom

Number of observations: 20 Degrees of Freedom: 17

SSR: 65.683 MSE: 3.864 Root MSE: 1.966

Multiple R-Squared: 0.755 Adjusted R-Squared: 0.726

SUR estimates for 'supply' (equation 2)

Model Formula: `consump ~ price + farmPrice + trend`

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	61.9662	11.0808	5.59	4.0e-05
price	0.1469	0.0944	1.56	0.13941
farmPrice	0.2140	0.0399	5.37	6.3e-05
trend	0.3393	0.0679	5.00	0.00013

Residual standard error: 2.55 on 16 degrees of freedom

Number of observations: 20 Degrees of Freedom: 16

SSR: 104.058 MSE: 6.504 Root MSE: 2.55

Multiple R-Squared: 0.612 Adjusted R-Squared: 0.539

```
> compareCoefs(fitols, fitsur)
```

Call:

```
1:"systemfit(formula = eqSystem, data = Kmenta)"
```

```
2:"systemfit(formula = eqSystem, method = \"SUR\", data = Kmenta)"
```

	Est. 1	SE 1	Est. 2	SE 2
demand_(Intercept)	99.8954	7.5194	99.3329	7.5145
demand_price	-0.3163	0.0907	-0.2755	0.0885
demand_income	0.3346	0.0454	0.2986	0.0419
supply_(Intercept)	58.2754	11.4629	61.9662	11.0808
supply_price	0.1604	0.0949	0.1469	0.0944
supply_farmPrice	0.2481	0.0462	0.2140	0.0399
supply_trend	0.2483	0.0975	0.3393	0.0679