Correspondence Analysis & Related Methods

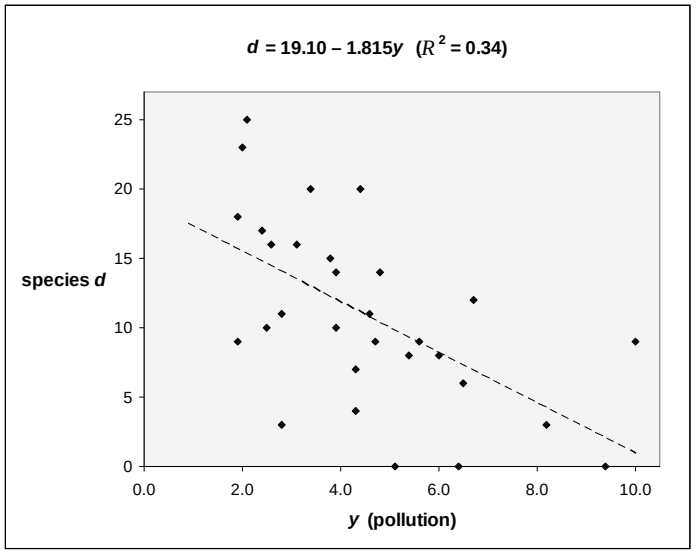
Michael Greenacre

SESSION 5:

REGRESSION BIPLOTS

d	*y* (pollution)
14	4.8
11	2.8
8	5.4
3	8.2
10	3.9
16	2.6
11	4.6
0	5.1
14	3.9
9	10.0
6	6.5
15	3.8
0	9.4
9	4.7
12	6.7
3	2.8
0	6.4
20	4.4
16	3.1
9	5.6
4	4.3
9	1.9
17	2.4
7	4.3
23	2.0
10	2.5
25	2.1
20	3.4
8	6.0
18	1.9

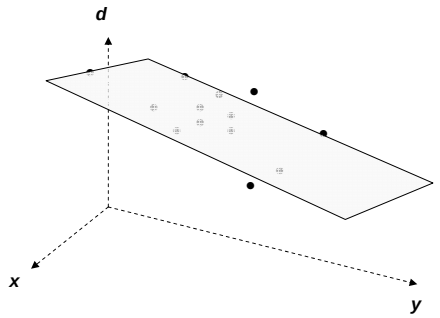
Simple linear regression species d versus pollution (y)



d	*y*	*x*
14	4.8	72
11	2.8	75
8	5.4	59
3	8.2	64
10	3.9	61
16	2.6	94
11	4.6	53
0	5.1	61
14	3.9	68
9	10.0	69
6	6.5	57
15	3.8	84
0	9.4	53
9	4.7	83
12	6.7	100
3	2.8	84
0	6.4	96
20	4.4	74
16	3.1	79
9	5.6	73
4	4.3	59
9	1.9	54
17	2.4	95
7	4.3	64
23	2.0	97
10	2.5	78
25	2.1	85
20	3.4	92
8	6.0	51
18	1.9	99

Multiple linear regression

$$\boldsymbol{d} = 6.135 - 1.388\boldsymbol{y} + 0.148\boldsymbol{x}$$



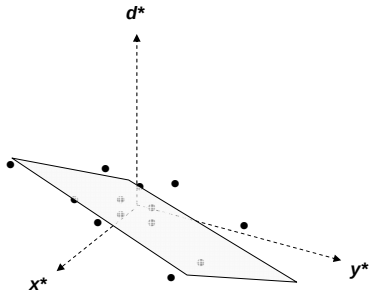
Regression model is a (hyper)plane

$R^2 = 0.442$

d *	*y**	*x**
0.503	0.132	-0.156
0.052	-0.802	0.036
-0.400	0.413	-0.988
-1.152	1.720	-0.668
-0.099	-0.288	-0.860
0.804	-0.895	1.253
0.052	0.039	-1.373
-1.603	0.272	-0.860
0.503	-0.288	-0.412
-0.249	2.561	-0.348
-0.701	0.926	-1.116
0.654	-0.335	0.613
-1.603	2.281	-1.373
-0.249	0.086	0.549
0.202	1.020	1.637
-1.152	-0.802	0.613
-1.603	0.880	1.381
1.406	-0.054	-0.028
0.804	-0.662	0.292
-0.249	0.506	-0.092
-1.001	-0.101	-0.988
-0.249	-1.222	-1.309
0.955	-0.989	1.317
-0.550	-0.101	-0.668
1.858	-1.175	1.445
-0.099	-0.942	0.228
2.159	-1.129	0.677
1.406	-0.522	1.125
-0.400	0.693	-1.501
1.105	-1.222	1.573

Multiple linear regression, variables standardized

$$\boldsymbol{d^*} = -.446\boldsymbol{y^*} + 0.347\boldsymbol{x^*}$$



Explanatory variables ***x*** and ***y*** and response variable ***d*** standardized

From usual regression coefficients to standardized ones

$$d = a + bx + cy$$

$$\bar{d} = a + b\bar{x} + c\bar{y}$$

$$d - \bar{d} = b(x - \bar{x}) + c(y - \bar{y}) = bs_x \frac{(x - \bar{x})}{s_x} + cs_y \frac{(y - \bar{y})}{s_y}$$

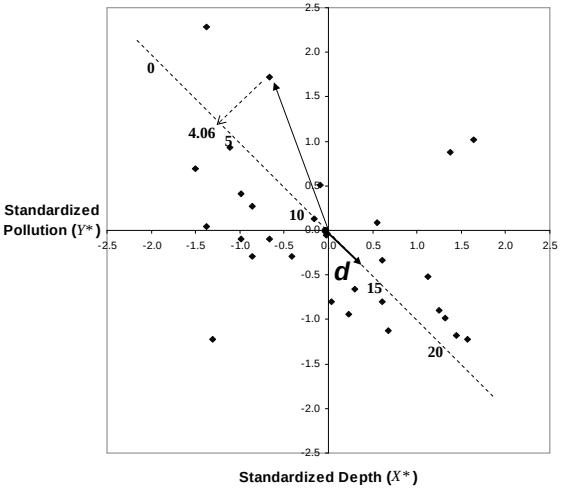
$$\frac{d - \bar{d}}{s_d} = \frac{bs_x}{s_d} \frac{(x - \bar{x})}{s_x} + \frac{cs_y}{s_D} \frac{(y - \bar{y})}{s_y}$$

$$\boxed{d^* = b\frac{s_x}{s_d} x^* + c\frac{s_y}{s_d} y^*}$$

Another view of regression & prediction

d	y	x
14	4.8	72
11	2.8	75
8	5.4	59
3	8.2	64
10	3.9	61
16	2.6	94
11	4.6	53
0	5.1	61
14	3.9	68
9	10.0	69
6	6.5	57
15	3.8	84
0	9.4	53
9	4.7	83
12	6.7	100
3	2.8	84
0	6.4	96
20	4.4	74
16	3.1	79
9	5.6	73
4	4.3	59
9	1.9	54
17	2.4	95
7	4.3	64
23	2.0	97
10	2.5	78
25	2.1	85
20	3.4	92
8	6.0	51
18	1.9	99

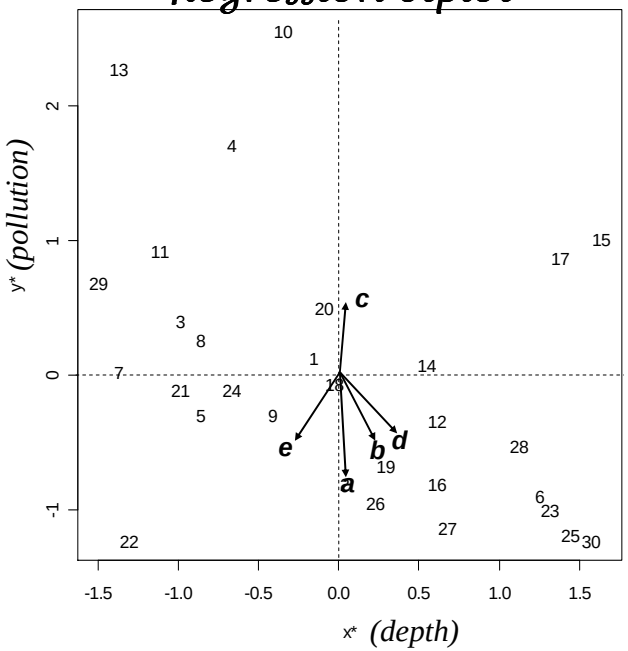
$$d^* = -0.446y^* + 0.347x^*$$



variance explained (R^2): 44.2%

y*	x*
0.132	-0.156
-0.802	0.036
0.413	-0.988
1.720	-0.668
-0.288	-0.860
-0.895	1.253
0.039	-1.373
0.272	-0.860
-0.288	-0.412
2.561	-0.348
0.926	-1.116
-0.335	0.613
2.281	-1.373
0.086	0.549
1.020	1.637
-0.802	0.613
0.880	1.381
-0.054	-0.028
-0.662	0.292
0.506	-0.092
-0.101	-0.988
-1.222	-1.309
-0.989	1.317
-0.101	-0.668
-1.175	1.445
-0.942	0.228
-1.129	0.677
-0.522	1.125
0.693	-1.501
-1.222	1.573

Regression biplot



variance explained:

- ***a***: 52.9%
- ***b***: 39.1%
- ***c***: 21.8%
- ***d***: 44.2%
- ***e***: 23.5%

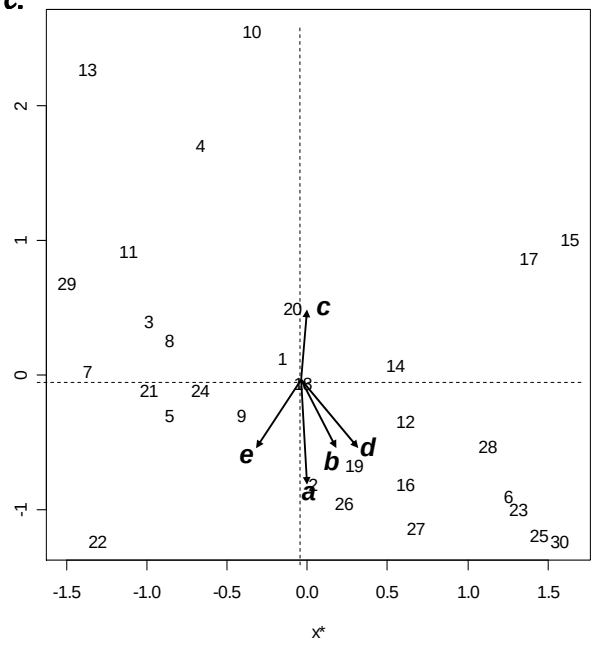
overall: 36.3%

significance:

	y	x
a:	**	
b:	**	
c:	*	
d:	**	*
e:	*	*

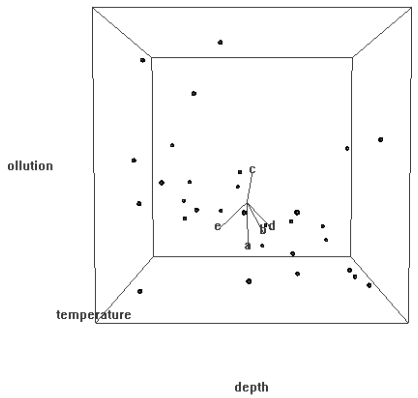
Regression biplot: summary

- A regression model can be represented as a point in space (in this case two-dimensional space because two explanatory variables)
- Reconstruct the data from projections of cases onto variable directions, but only as well as measured by R^2
- If *x* and *y* are uncorrelated, then the regression coefficients are just correlation coefficients (in this case, of the species with the explanatory variables)



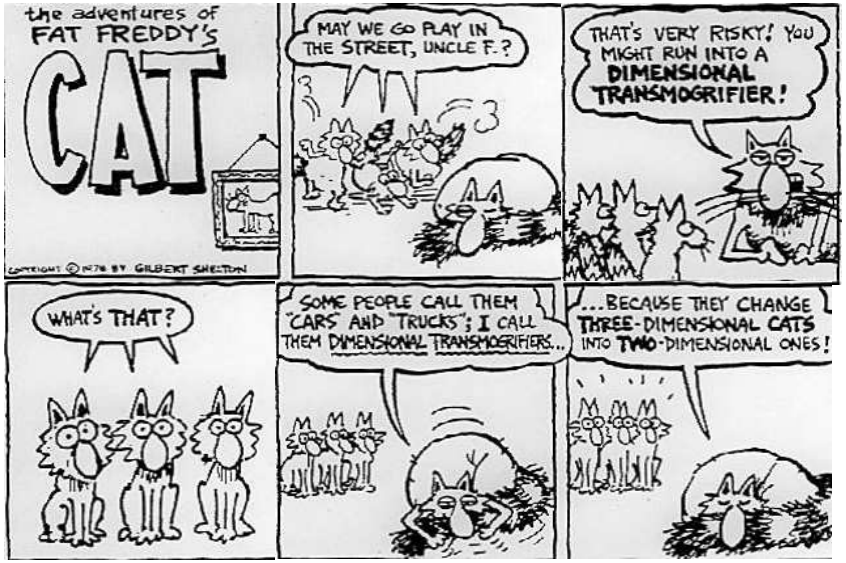
What happens for three predictors?

- Each regression model can be represented as a point in three-dimensional space.
- Reconstruct the data from projections of cases onto variable directions, but only as well as measured by R^2 ; in this example the increase in explained variance from two-dimensional to three-dimensional (adding temperature as an explanatory variable) is from 36.3% to 37.1%, hence temperature is explaining very little extra variance.



There will be a particular orientation of the vectors that gives maximum variance explained in the two-dimensional projection

Dimensional Transmogrifier



with thanks to Jörg Blasius

Interpretation of biplots

- A biplot usually represents cases as points and variables as vectors.
  (in this example, the points are the 30 sites and the variables are the 5 species)
- Reconstruct approximations to the data from projections of cases onto the biplot axes given by the variable vectors. This gives a measure of good ness of fit, usually expressed as a percentage.
- The biplot needs a system of axes for its construction: here we used the predictor variables pollution, depth and temperature.
- If the system of axes is more than two-dimensional we generally project ("transmogrify") the high-dimensional display onto a plane.
- If we don't have a given system of axes, we can derive a set of latent axes with the same idea: optimize the reconstruction of the original data set in the biplot – this is the idea behind principal component analysis (PCA) and correspondence analysis (CA).

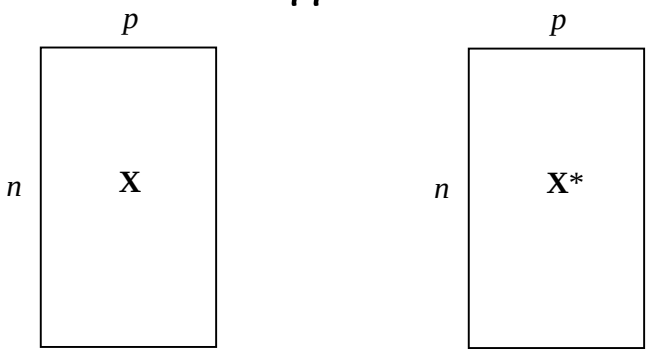
Correspondence Analysis & Related Methods

Michael Greenacre

SESSION 6:

SINGULAR VALUE DECOMPOSITION

Matrix approximation



usually of "full rank" p

Geometrically, we need p dimensions to represent the rows of $\mathbf{X}$

"reduced rank" $p^* < p$

Geometrically, we need p^* dimensions to represent the rows of $\mathbf{X}^*$

Question: how do we find the $\mathbf{X}^*$ that is "closest" to $\mathbf{X}$ **?**

Defining distance betwen matrices

Question: how do we find the matrix $\mathbf{X}^*$ that is "closest" to $\mathbf{X}$ **?** Let's define "closeness" by sum of squared differences (for example): this sounds like the squared Euclidean distance, and it is, except it is calculated between two matrices, not between two vectors[1]. If you thought of a $n \times p$ matrix as a vector of np elements, then it is the squared Euclidean distance.

$$\text{RSS} = \sum_{i=1}^{n}\sum_{j=1}^{p}(x_{ij} - x_{ij}^*)^2$$

So we need to find the $\mathbf{X}^*$ that minimizes the residual sum of squares RSS

[1] In this context it is often called the Frobenius distance

Singular value decomposition

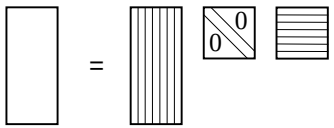
The matrix decomposition called the "singular value decomposition", or SVD, provides the solution to this least-squares problem. It does so in a globally optimum way, and provides solutions for any rank (i.e., any dimensionality). The SVD is the rectangular matrix equivalent of the eigenvalue-eigenvector decomposition of a square matrix.

$$\mathbf{X} = \mathbf{U}\mathbf{D}_\alpha\mathbf{V}^\mathsf{T} \quad \text{where} \quad \mathbf{U}^\mathsf{T}\mathbf{U} = \mathbf{V}^\mathsf{T}\mathbf{V} = \mathbf{I}\,,\ \alpha_1 \geq \alpha_2 \geq \cdots \geq 0$$

left singular vectors; right singular vectors; singular values

singular vectors are orthonormal (sum of squares =1 and orthogonal)

singular values are positive, in descending order



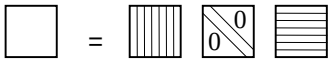
R function **svd**

Eigen-decomposition

Compare the SVD with the eigen-decomposition of a square symmetric matrix:

$$\mathbf{A} = \mathbf{V}\mathbf{D}_{\lambda}\mathbf{V}^{\mathsf{T}} \quad \text{where } \mathbf{V}^{\mathsf{T}}\mathbf{V} = \mathbf{I},\ \lambda_1 \geq \lambda_2 \geq \cdots$$

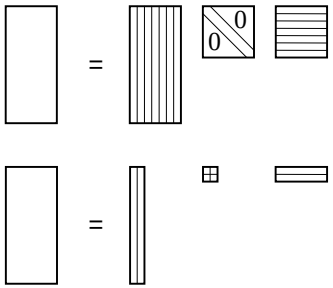
eigen-vectors; eigen-vectors; eigenvalues; eigenvectors are orthonormal (sum of squares =1 and orthogonal); eigenvalues are in descending order



R function **`eigen`**

Low-rank approximation

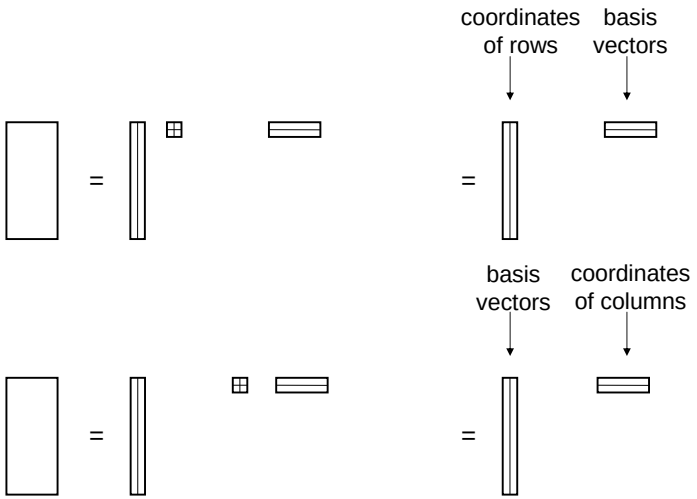
$$\mathbf{X} = \mathbf{U}\mathbf{D}_{\alpha}\mathbf{V}^{\mathsf{T}} \quad \text{where } \mathbf{U}^{\mathsf{T}}\mathbf{U} = \mathbf{V}^{\mathsf{T}}\mathbf{V} = \mathbf{I},\ \alpha_1 \geq \alpha_2 \geq \cdots \geq 0$$



$\mathbf{X}^{*} = \mathbf{U}_{[2]}\mathbf{D}_{\alpha_{[2]}}\mathbf{V}_{[2]}^{\mathsf{T}}$ is the best rank 2 approximation
(Eckart-Young theorem, Psychometrika, 1936)

Coordinates and basis vectors

$\mathbf{X}^{*} = \mathbf{U}_{[2]}\mathbf{D}_{\alpha_{[2]}}\mathbf{V}_{[2]}^{\mathsf{T}}$ is the best rank 2 approximation



Quality of approximation

$$\mathbf{X} = \mathbf{U}\mathbf{D}_{\alpha}\mathbf{V}^{\mathsf{T}} \quad \text{where } \mathbf{U}^{\mathsf{T}}\mathbf{U} = \mathbf{V}^{\mathsf{T}}\mathbf{V} = \mathbf{I},\ \alpha_1 \geq \alpha_2 \geq \cdots \geq 0$$

$\mathbf{X}^{*} = \mathbf{U}_{[2]}\mathbf{D}_{\alpha_{[2]}}\mathbf{V}_{[2]}^{\mathsf{T}}$ is the best rank 2 approximation

$$\sum_i \sum_j x_{ij}^2 = \text{trace}(\mathbf{X}\mathbf{X}^{\mathsf{T}}) = \text{trace}(\mathbf{U}\mathbf{D}_{\alpha}\mathbf{V}^{\mathsf{T}}\mathbf{V}\mathbf{D}_{\alpha}\mathbf{U}^{\mathsf{T}})$$

$$= \text{trace}(\mathbf{D}_{\alpha}^2) = \sum_k \alpha_k^2$$ is the total sum of squares

$$\sum_i \sum_j x_{ij}^{*2} = \sum_{k=1}^{2} \alpha_k^2$$ is the sum of squares in two dimensions

$$\sum_i \sum_j (x_{ij} - x_{ij}^{*})^2 = \sum_{k=3}^{\cdots} \alpha_k^2$$ is the error sum of squares

Generalized SVD

We often want to associate weights on the rows and columns, so that the fit is by weighted least-squares, not ordinary least squares, that is we want to minimize

$$\text{RSS} = \sum_{i=1}^{n}\sum_{j=1}^{p} r_i c_j (x_{ij} - x_{ij}^*)^2$$

$$\mathbf{D}_r^{1/2}\mathbf{X}\mathbf{D}_c^{1/2} = \mathbf{U}\mathbf{D}_\alpha\mathbf{V}^\mathsf{T} \quad \text{where } \mathbf{U}^\mathsf{T}\mathbf{U} = \mathbf{V}^\mathsf{T}\mathbf{V} = \mathbf{I}, \alpha_1 \geq \alpha_2 \geq \cdots \geq 0$$

$$\mathbf{X} = \mathbf{D}_r^{-1/2}\mathbf{U}\mathbf{D}_\alpha(\mathbf{D}_c^{-1/2}\mathbf{V})^\mathsf{T}$$

$$\mathbf{X}^* = etc...$$

Generalized principal component analysis

We take the case of points defined in the rows of $\mathbf{X}$; that is, n rows of dimensionality p. First we need to center $\mathbf{X}$ w.r.t. column means:

$$\mathbf{Y} = (\mathbf{I} - (1/n)\mathbf{1}\mathbf{1}^\mathsf{T})\mathbf{X}$$

Suppose the distance between (centered) rows is defined by a weighted Euclidean distance with weights $1/m_j$, and that each row has a mass of r_i.

$$\sum_{i=1}^{n} r_i \sum_{j=1}^{p} \frac{(y_{ij} - y_{ij}^*)^2}{m_j} \qquad \text{RSS} = \sum_{i=1}^{n}\sum_{j=1}^{p} (r_i / m_j)(y_{ij} - y_{ij}^*)^2$$

$$\mathbf{D}_r^{1/2}\mathbf{Y}\mathbf{D}_m^{-1/2} = \mathbf{U}\mathbf{D}_\alpha\mathbf{V}^\mathsf{T} \quad \text{where } \mathbf{U}^\mathsf{T}\mathbf{U} = \mathbf{V}^\mathsf{T}\mathbf{V} = \mathbf{I}, \alpha_1 \geq \alpha_2 \geq \cdots \geq 0$$

$$\mathbf{Y} = \underbrace{\mathbf{D}_r^{-1/2}\mathbf{U}\mathbf{D}_\alpha}_{\text{coordinates of rows}}\underbrace{(\mathbf{D}_m^{1/2}\mathbf{V})^\mathsf{T}}_{\text{basis vectors}} \qquad \mathbf{Y}^* = etc...$$

Example: Economic indicators

	Unemp	GDPH	PCH	PCP	RULC
Belgium	8.8	102	104.9	3.3	89.7
Denmark	7.6	134.4	117.1	1	92.4
Germany	5.4	128.1	126	3	90
Greece	8.5	37.7	40.5	2	105.6
Spain	16.5	67.1	68.7	4	86.2
France	9.1	112.4	110.1	2.8	89.7
Ireland	16.2	64	60.1	4.5	81.9
Italy	10.6	105.8	106	3.8	97.4
Luxembourg	1.7	119.5	110.7	2.8	95.9
Netherlands	9.6	99.6	96.7	3.3	86.6
Portugal	5.2	32.6	34.8	3.5	78.3
U.K.	6.5	95.3	99.7	2.1	98.9

$$\sum_{i=1}^{n}(1/12)\sum_{j=1}^{p}\frac{(y_{ij} - y_{ij}^*)^2}{s_j^2} \qquad r_i = 1/12 \quad m_j = s_j^2$$

$$\mathbf{D}_r^{1/2}\mathbf{Y}\mathbf{D}_m^{-1/2} = \mathbf{U}\mathbf{D}_\alpha\mathbf{V}^\mathsf{T} \quad \text{where } \mathbf{U}^\mathsf{T}\mathbf{U} = \mathbf{V}^\mathsf{T}\mathbf{V} = \mathbf{I}, \alpha_1 \geq \alpha_2 \geq \cdots \geq 0$$

$$\mathbf{Y} = \underbrace{\mathbf{D}_r^{-1/2}\mathbf{U}\mathbf{D}_\alpha}_{\text{coordinates of rows}}\underbrace{(\mathbf{D}_m^{1/2}\mathbf{V})^\mathsf{T}}_{\text{basis vectors}} \qquad \mathbf{Y}^* = etc...$$

R code

```
# read in E.U. economic indicators into data.frame EU

n     <- nrow(EU)
p     <- ncol(EU)

Dr    <- diag(rep(1, n)/n)
Drm1  <- diag(rep(n, n))

Y     <- sweep(EU, 2, apply(EU, 2, mean))

s2    <- ((n-1)/n) * apply(EU, 2, var)

Dsm1  <- diag(1/sqrt(s2))

S     <- sqrt(Dr) %*% as.matrix(Y) %*% Dsm1

S.svd <- svd(S)

FF    <- sqrt(Drm1) %*% S.svd$u %*% diag(S.svd$d)

plot(FF[,1], FF[,2], type="n")
text(FF[,1], FF[,2], labels=rownames(EU))
```

This shows the countries (rows) as points; now you have to add the variables (columns) as biplot vectors – see homework