Principal Component Analysis

Rosario Barone*

*Email: rosario.barone@uniroma2.it

Suppose to have n observations of p variables, such that $\mathbf{X}' = (X_1, \ldots, X_p)$.

- Graphical representations are useful with small number of variables
- Summarizing data in presence of too many variables can be difficult: Suppose we have data on athletes and their scores in 5 different specialties. How to summarize such information in order to rank them?
- Furthermore, the explanatory variables can be highly correlated: issues when applying multivariate techniques (e.g., regression)

Suppose to have n observations of p variables, such that $\mathbf{X}' = (X_1, \ldots, X_p)$.

- Graphical representations are useful with small number of variables
- Summarizing data in presence of too many variables can be difficult: Suppose we have data on athletes and their scores in 5 different specialties. How to summarize such information in order to rank them?
- Furthermore, the explanatory variables can be highly correlated: issues when applying multivariate techniques (e.g., regression)

Suppose to have n observations of p variables, such that $\mathbf{X}' = (X_1, \ldots, X_p)$.

- Graphical representations are useful with small number of variables
- Summarizing data in presence of too many variables can be difficult: Suppose we have data on athletes and their scores in 5 different specialties. How to summarize such information in order to rank them?
- Furthermore, the explanatory variables can be highly correlated: issues when applying multivariate techniques (e.g., regression)

Suppose to have n observations of p variables, such that $\mathbf{X}' = (X_1, \ldots, X_p)$.

- Graphical representations are useful with small number of variables
- Summarizing data in presence of too many variables can be difficult: Suppose we have data on athletes and their scores in 5 different specialties. How to summarize such information in order to rank them?
- Furthermore, the explanatory variables can be highly correlated: issues when applying multivariate techniques (e.g., regression)

Solutions

- Simplest: take just one element and discard all others: loss of information and in interpretation
- Consider a summary measure $\frac{1}{p}\sum_{j=1}^{p} X_j$: same importance to all the variables
- Cosider a summary measure $\mathbf{w}'\mathbf{X} = \sum_{j=1}^{p} w_j X_j$ such that $\mathbf{w}'\mathbf{w} = \sum_{j=1}^{p} w_j^2 = 1$
  - How do we choose the weights?
  - Is a single summary measure enough?
  - A more advanced solution: Principal Component Analysis (PCA)

Solutions

- Simplest: take just one element and discard all others: loss of information and in interpretation
- Consider a summary measure $\frac{1}{p}\sum_{j=1}^{p} X_j$: same importance to all the variables
- Cosider a summary measure $\mathbf{w}'\mathbf{X} = \sum_{j=1}^{p} w_j X_j$ such that $\mathbf{w}'\mathbf{w} = \sum_{j=1}^{p} w_j^2 = 1$
  - How do we choose the weights?
  - Is a single summary measure enough?
  - A more advanced solution: Principal Component Analysis (PCA)

Solutions

- Simplest: take just one element and discard all others: loss of information and in interpretation
- Consider a summary measure $\frac{1}{p}\sum_{j=1}^{p} X_j$: same importance to all the variables
- Cosider a summary measure $\mathbf{w}'\mathbf{X} = \sum_{j=1}^{p} w_j X_j$ such that $\mathbf{w}'\mathbf{w} = \sum_{j=1}^{p} w_j^2 = 1$
  - How do we choose the weights?
  - Is a single summary measure enough?
  - A more advanced solution: Principal Component Analysis (PCA)

Solutions

- Simplest: take just one element and discard all others: loss of information and in interpretation
- Consider a summary measure $\frac{1}{p}\sum_{j=1}^{p} X_j$: same importance to all the variables
- Cosider a summary measure $\mathbf{w}'\mathbf{X} = \sum_{j=1}^{p} w_j X_j$ such that $\mathbf{w}'\mathbf{w} = \sum_{j=1}^{p} w_j^2 = 1$
  - How do we choose the weights?
  - Is a single summary measure enough?
  - A more advanced solution: Principal Component Analysis (PCA)

Principal Component Analysis

Principal Component Analysis (PCA) is an unsupervised exploratory MVA technique that aims at:

- Data reduction: reduce the dimension of the data matrix without (almost) losing information;
- Data visualization: identify a lower dimensional space where to project the points without (almost) deforming the distance between them;
- Data interpretation: reducing the original data, we create a smaller number of new variables (artificial variables) that might have a more direct interpretation than the original ones.

Suppose to have a p-dimensional set of correlated variables $\mathbf{X}' = (X_1, \ldots, X_p)$. PCA is used to extract and express important information of the original variables with a set of few new variables $\mathbf{Y}' = (Y_1, \ldots, Y_p)$ called principal components (PCs), that:

- correspond to linear combinations of $\mathbf{X}$:

$$Y_i = \mathbf{a}_i' \mathbf{X} = \sum_{j=1}^{p} a_{ij} X_j \quad i = 1, \ldots, q \quad (q \leq p)$$

- are uncorrelated (orthogonal): $Y_i' Y_j = 0 \quad \forall i, j = 1, \ldots, q$
- are build in a sequential way and ordered:
  - Y_1 is the variable with the highest variability;
  - Y_2 is orthogonal to Y_1 and $Var(Y_2) \leq Var(Y_1)$

The information of a data matrix X corresponds to its total variation:

$$I_{tot} = \sum_{j=1}^{p} Var(X_j)$$

The goal of PCA is to identify directions (or principal components) along which the variation in the data is maximal

- Y_1 is s.t. $Var(Y_1) = Var(\mathbf{a}_1' X) = \mathbf{a}_1' \mathbf{\Sigma}_x \mathbf{a}_1$ is maximal;
- Y_2 is s.t. $Var(Y_2) = Var(\mathbf{a}_2' X) = \mathbf{a}_2' \mathbf{\Sigma}_x \mathbf{a}_2$ is maximal s.t. $Cov(Y_1, Y_2) = 0$

where $\mathbf{\Sigma}_x = Var(X)$.

The information of a data matrix X corresponds to its total variation:

$$I_{tot} = \sum_{j=1}^{p} Var(X_j)$$

The goal of PCA is to identify directions (or principal components) along which the variation in the data is maximal

- Y_1 is s.t. $Var(Y_1) = Var(\mathbf{a}_1' X) = \mathbf{a}_1' \mathbf{\Sigma}_x \mathbf{a}_1$ is maximal;
- Y_2 is s.t. $Var(Y_2) = Var(\mathbf{a}_2' X) = \mathbf{a}_2' \mathbf{\Sigma}_x \mathbf{a}_2$ is maximal s.t. $Cov(Y_1, Y_2) = 0$

where $\mathbf{\Sigma}_x = Var(X)$.

The "interesting directions" are found through the spectral decomposition of $\mathbf{\Sigma}_x$. They are given by the eigenvectors $\mathbf{v}$ corresponding to the largest eigenvalues λ of the covariance matrix $\mathbf{\Sigma}_x$.

1. Find the eigenvalues of $\mathbf{\Sigma}_x$ and order them:

$$\lambda_1 \geq \cdots \geq \lambda_p$$

2. Find the eigenvectors $\phi_1 \ldots \phi_p$ corresponding to the eigenvalues $\lambda_1, \ldots, \lambda_p$ and define the first component as:

$$Y_1 = \phi_1 \mathbf{X} = \sum_{j=1}^{p} \phi_{1j} X_j$$

3. Repeat the second step for each component.

Note that:

- $Var(Y_j) = \lambda_j$
- $\sum_{j=1}^{p} \lambda_j = I_{tot}$

When we choose the number of PCs, we agree on a maximum number of information to lose:

- We generally want to preserve 80% of the total information I_{tot}
- We select the first q components such that:

$$\frac{Var(Y_1) + \cdots + Var(Y_q)}{I_{tot}} \approx 0.8$$

- You can work with the original variables (variance-covariance matrix) or with the standardized variables (correlation matrix). The results are, in general, different.

- Original variables $\mathbf{X}$ strongly correlated: maximum dimensionality reduction;

- Original variables $\mathbf{X}$ uncorrelated: $p = q$

- PCs are linear combination of the original variables:

$$Y_{ij} = \phi_{1j} X_{i1} + \phi_{2j} X_{i2} + \cdots + \phi_{qj} X_{iq}$$

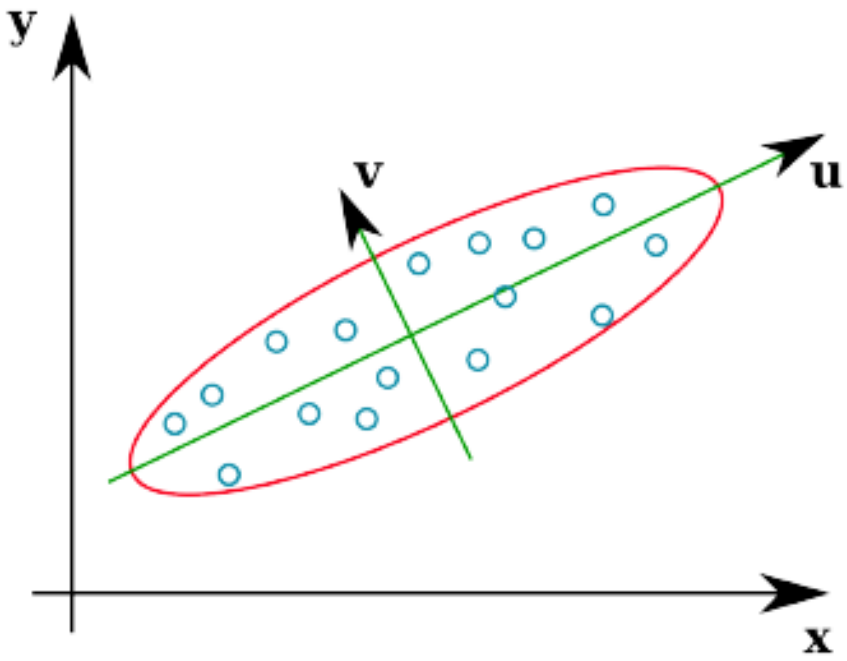
  where the generic coefficient ϕ_{ij} represents the weights that the variable X_j has in determining the j-th principal component;

- The component Y_j will be mostly represented by the variables with the larger coefficients.

Geometrical interpretation

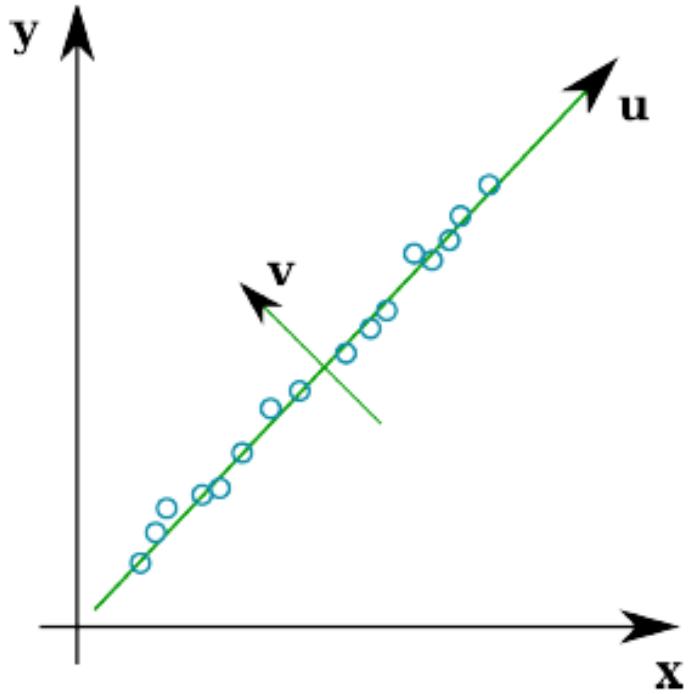
- PCs represent a selection of a new coordinate system obtained by rotating the original axes to a set of new axes (to provide a simpler structure).
    - The first PC represents the direction of maximum variability, followed by the others, all orthogonal.
- "Best" fit hyper-plane: minimizes the sum of squared distances between points that represent cases and space defined by PCs
    - The first PC defines a line; the first Two PCs define a plane.

PCA for Data Representation



- the principal direction in which the data varies is shown by the U axis;
- the second most important direction is the V axis (orthogonal to U).
- If we place the $U - V$ axis system at the mean of the data it gives us a compact representation.

Data Representation and dimension Reduction



- If the variation in the data is caused by some other relationship then PCA gives us a way of reducing the dimensionality of a data set
- the principal direction in which the data varies is shown by the U axis
- in this case all the V coordinates are all very close to zero: we can represent the data set by one variable U and discard V

PC calculation

Required background for PCA: (a little of) linear algebra:

- The principal components are found by calculating the eigenvectors and eigenvalues of the data covariance matrix: this process is equivalent to finding the axis system in which the co-variance matrix is diagonal.

- The eigenvector with the largest eigenvalue is the direction of greatest variation, the one with the second largest eigenvalue is the (orthogonal) direction with the next highest variation and so on.

Eigendecomposition

Let A be a $p \times p$ matrix. The eigenvalues of A are defined as the roots of:

$$\det(A - \lambda I) = |(A - \lambda I)| = 0$$

where I is a $p \times p$ matrix and the equation has p roots.

Let λ be an eigenvalue of A. Then there exists a vector ϕ such that:

$$A\phi = \lambda\phi$$

The vector ϕ is called an eigenvector of A associated with the eigenvalue λ.

Let consider a simple p=3 case. Suppose we have a 3×3 matrix A with eigenvectors ϕ_1, ϕ_2, ϕ_3, and eigenvalues λ_1, λ_2, λ_3 so:

$$A\phi_1 = \lambda_1\phi_1 \quad A\phi_2 = \lambda_2\phi_2 \quad A\phi_3 = \lambda_3\phi_3.$$

By defining:

$$\Phi = \begin{bmatrix} \phi_1 & \phi_2 & \phi_3 \end{bmatrix} \text{ and } \Lambda = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}$$

we get the matrix equation

$$A\Phi = \Phi\Lambda$$

Since eigenvectors are normalized to unit magnitude, and they are orthogonal, so $\Phi^T\Phi = I$. Therefore

$$A\Phi = \Phi\Lambda$$

$$\Phi^T A\Phi = \Phi^T\Phi\Lambda$$

$$\Phi^T A\Phi = \Lambda$$

and

$$A = \Phi^T\Lambda\Phi$$

The matrix A is represented in terms of its eigenvalues and eigenvectors; this factorization[†] is defined eigendecomposition. If A is a symmetric matrix, the decomposition is called spectral decomposition.

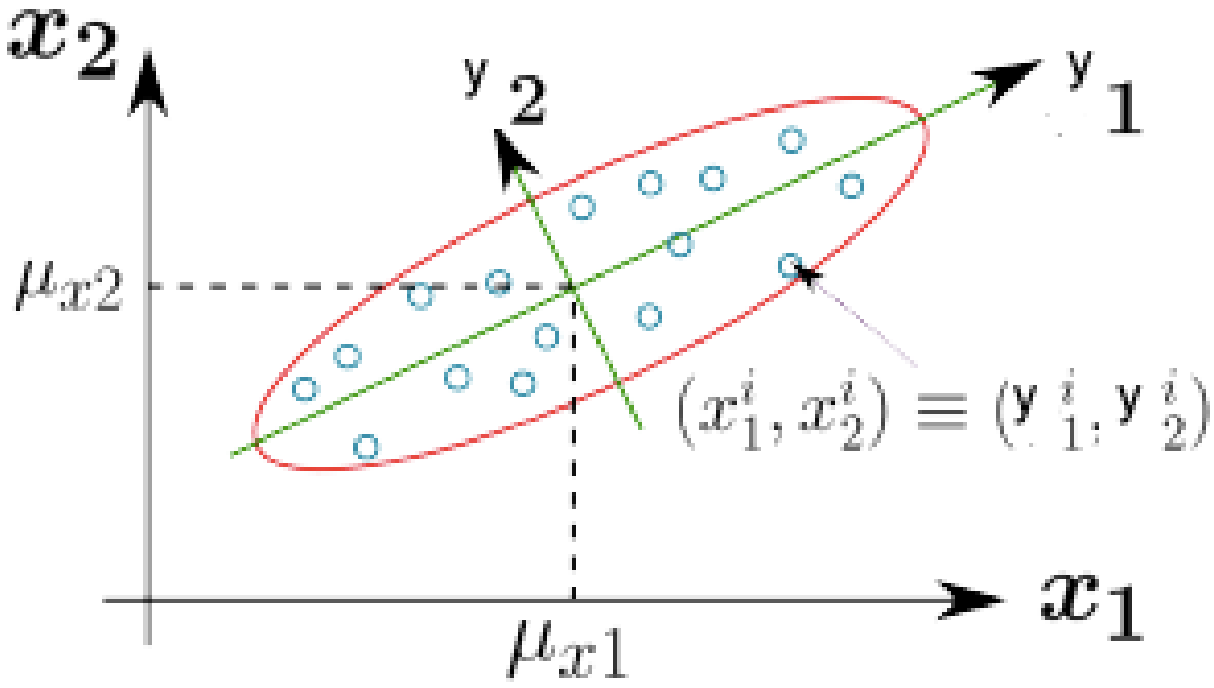
[†] a matrix factorization is a factorization of a matrix into a product of matrices

Spectral decomposition in PCA

Let Σ_x be an $p \times p$ covariance matrix. There is an orthogonal $p \times p$ matrix Φ, whose columns are eigenvectors of Σ_x and a diagonal matrix Λ whose non-zero elements are the eigenvalues of Σ_x such that

$$\Phi^T \Sigma_x \Phi = \Lambda$$

We can look on the matrix of eigenvectors as a linear transformation which transforms data points in the $[x_1; x_2]$ axis system into the $[y_1, y_2]$ axis system.



In the general case the linear transformation given by Φ transforms the data points into a data set where the variables are uncorrelated: the covariance matrix of the data in the new coordinate system is Λ which has zeros in all the off diagonal elements.

Summary

We start with the p $(X_1, \ldots, X_p)$ original variables (quantitative), and we obtain $q \leq p$ new variables, where q is a good compromise between

- the minimum number of variables (max dimensionality reduction)
- minimum lost of information (max variability)

$$(X_1, \ldots, X_p) \rightarrow (Y_1, \ldots, Y_q)$$

When?

- High-dimension datasets;
- Quantitative variables;
- Variables with a correlation pattern (redundancy in the data);
- $n > p$

Example: Brand ratings data

We investigate dimensionality using a simulated data set that is typical of consumer brand perception surveys. The data comprise ratings of 10 brands (*a* to *j*) on 9 adjectives (`performance, leader, fun, etc`), for n = 100 respondents, as expressed on survey items with the following form: scale from 1 (least) to 10 (most).

Example:

How trendy is Intelligentsia Coffee?

```
> brand.ratings <- read.csv("http://goo.gl/IQl8nc")
> head(brand.ratings)

  perform leader latest fun serious bargain value trendy rebuy brand
1       2      4      8   8       2       9     7      4     6     a
2       1      1      4   7       1       1     1      2     2     a
3       2      3      5   9       2       9     5      1     6     a
4       1      6     10   8       3       4     5      2     1     a
5       1      1      5   8       1       9     9      1     1     a
6       2      8      9   5       3       8     7      1     2     a
```

```
> str(brand.ratings)

'data.frame':        1000 obs. of  10 variables:
 $ perform: int  2 1 2 1 1 2 1 2 2 3 ...
 $ leader : int  4 1 3 6 1 8 1 1 1 1 ...
 $ latest : int  8 4 5 10 5 9 5 7 8 9 ...
 $ fun    : int  8 7 9 8 8 5 7 5 10 8 ...
 $ serious: int  2 1 2 3 1 3 1 2 1 1 ...
 $ bargain: int  9 1 9 4 9 8 5 8 7 3 ...
 $ value  : int  7 1 5 5 9 7 1 7 7 3 ...
 $ trendy : int  4 2 1 2 1 1 1 7 5 4 ...
 $ rebuy  : int  6 2 6 1 1 2 1 1 1 1 ...
 $ brand  : chr  "a" "a" "a" "a" ...
```

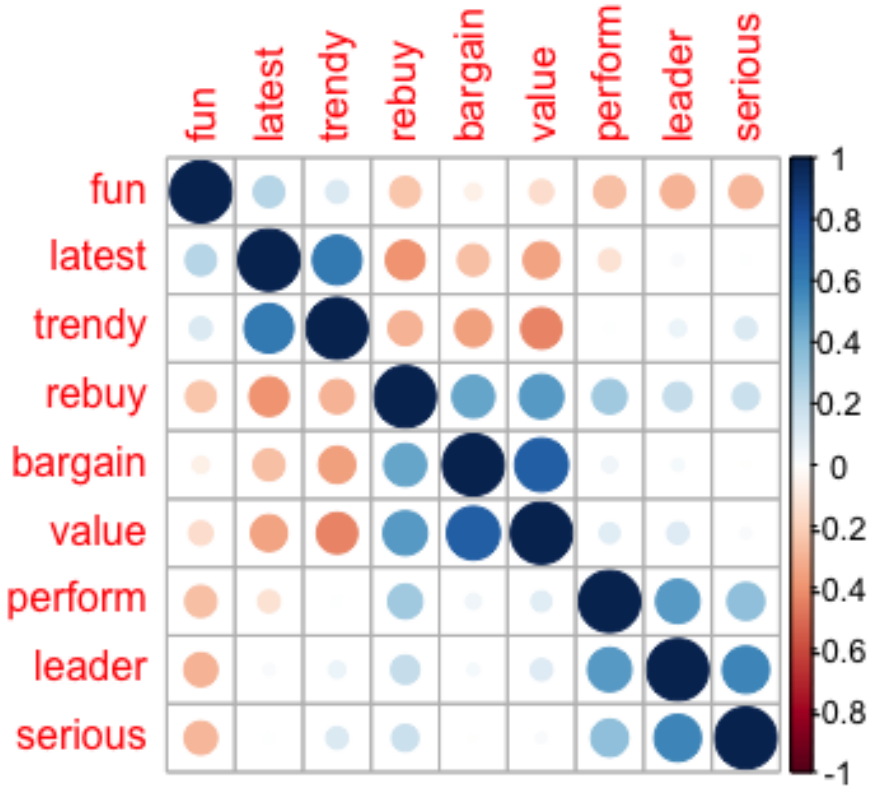
```
> library(psych)
> describe(brand.ratings)
```

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew
perform	1	1000	4.49	3.20	4.0	4.24	4.45	1	10	9	0.43
leader	2	1000	4.42	2.61	4.0	4.27	2.97	1	10	9	0.28
latest	3	1000	6.20	3.08	7.0	6.37	4.45	1	10	9	-0.35
fun	4	1000	6.07	2.74	6.0	6.18	2.97	1	10	9	-0.24
serious	5	1000	4.32	2.78	4.0	4.07	2.97	1	10	9	0.58
bargain	6	1000	4.26	2.67	4.0	4.07	2.97	1	10	9	0.37
value	7	1000	4.34	2.40	4.0	4.21	2.97	1	10	9	0.33
trendy	8	1000	5.22	2.74	5.0	5.19	2.97	1	10	9	0.02
rebuy	9	1000	3.73	2.54	3.0	3.43	2.97	1	10	9	0.74
brand*	10	1000	5.50	2.87	5.5	5.50	3.71	1	10	9	0.00

```
> brand.sc <- brand.ratings
> brand.sc[, 1:9] = apply(brand.ratings[, 1:9], 2, scale)
> summary(brand.sc)
```

```
    perform            leader            latest             fun              se
 Min.   :-1.0888   Min.   :-1.3100   Min.   :-1.6878   Min.   :-1.84677   Min.
 1st Qu.:-1.0888   1st Qu.:-0.9266   1st Qu.:-0.7131   1st Qu.:-0.75358   1st Q
 Median :-0.1523   Median :-0.1599   Median : 0.2615   Median :-0.02478   Media
 Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.00000   Mean
 3rd Qu.: 0.7842   3rd Qu.: 0.6069   3rd Qu.: 0.9113   3rd Qu.: 0.70402   3rd Q
 Max.   : 1.7206   Max.   : 2.1404   Max.   : 1.2362   Max.   : 1.43281   Max.
    bargain             value              trendy             rebuy              b
 Min.   :-1.22196   Min.   :-1.3912   Min.   :-1.53897   Min.   :-1.0717   Leng
 1st Qu.:-0.84701   1st Qu.:-0.9743   1st Qu.:-0.80960   1st Qu.:-1.0717   Clas
 Median :-0.09711   Median :-0.1405   Median :-0.08023   Median :-0.2857   Mode
 Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.0000
 3rd Qu.: 0.65279   3rd Qu.: 0.6933   3rd Qu.: 0.64914   3rd Qu.: 0.5003
 Max.   : 2.15258   Max.   : 2.3610   Max.   : 1.74319   Max.   : 2.4652
```

> library(corrplot) > corrplot(cor(brand.sc[, 1:9]), order = "hclust")



Perform PCA

```
> brand.pc <- prcomp(brand.sc[, 1:9])
> summary(brand.pc)

Importance of components:
                          PC1    PC2    PC3    PC4     PC5
Standard deviation     1.726 1.4479 1.0389 0.8528 0.79846
Proportion of Variance 0.331 0.2329 0.1199 0.0808 0.07084
Cumulative Proportion  0.331 0.5640 0.6839 0.7647 0.83554
                          PC6     PC7     PC8     PC9
Standard deviation     0.73133 0.62458 0.55861 0.49310
Proportion of Variance 0.05943 0.04334 0.03467 0.02702
Cumulative Proportion  0.89497 0.93831 0.97298 1.00000
```

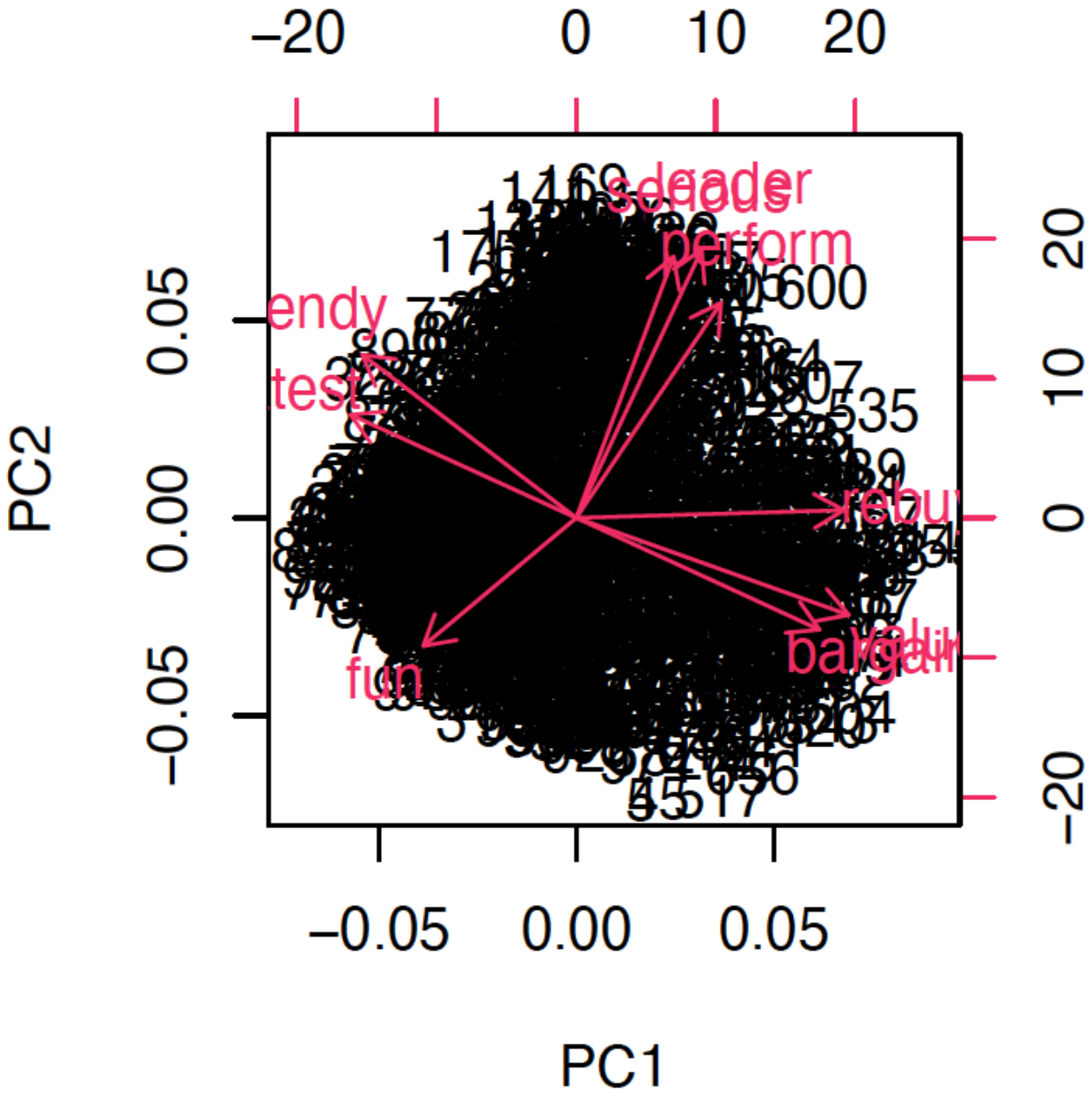
Check orthogonality

```
> cor(brand.pc$x)
```

	PC1	PC2	PC3	PC4
PC1	1.000000e+00	-1.027928e-16	-8.324536e-17	8.071781e-16
PC2	-1.027928e-16	1.000000e+00	5.124408e-16	2.578368e-15
PC3	-8.324536e-17	5.124408e-16	1.000000e+00	-3.101350e-16
PC4	8.071781e-16	2.578368e-15	-3.101350e-16	1.000000e+00
PC5	-3.999033e-16	-9.154739e-16	3.040767e-16	2.044069e-15
PC6	1.196430e-15	2.621610e-15	-2.423218e-16	-1.758965e-15
PC7	-2.677865e-16	5.669396e-16	-6.452385e-16	1.236603e-15
PC8	-1.195225e-16	5.143978e-16	-6.455569e-16	-4.750556e-16
PC9	-8.130889e-17	7.586994e-16	-4.401351e-16	6.530373e-16

	PC5	PC6	PC7	PC8	PC9
PC1	-3.999033e-16	1.196430e-15	-2.677865e-16	-1.195225e-16	-8.130889e-17
PC2	-9.154739e-16	2.621610e-15	5.669396e-16	5.143978e-16	7.586994e-16
PC3	3.040767e-16	-2.423218e-16	-6.452385e-16	-6.455569e-16	-4.401351e-16
PC4	2.044069e-15	-1.758965e-15	1.236603e-15	-4.750556e-16	6.530373e-16
PC5	1.000000e+00	1.154086e-15	-2.006218e-15	1.290515e-15	-1.117524e-15
PC6	1.154086e-15	1.000000e+00	1.167053e-15	2.354267e-16	5.737252e-16
PC7	-2.006218e-15	1.167053e-15	1.000000e+00	1.504508e-15	-5.307298e-16
PC8	1.290515e-15	2.354267e-16	1.504508e-15	1.000000e+00	7.434271e-16
PC9	-1.117524e-15	5.737252e-16	-5.307298e-16	7.434271e-16	1.000000e+00

```
> biplot(brand.pc)
```



The plot of individual respondents' ratings is too dense and it does not tell us about the brand positions! Biplots are especially helpful when:

- there are a smaller number of points
- when there are clusters

Better solution: perform PCA using aggregated ratings by brand!

```
brand.mean <- aggregate(brand.sc[,1:9], list(brand.sc[,10]), mean)
rownames(brand.mean) = brand.mean[, 1]
brand.mean
```

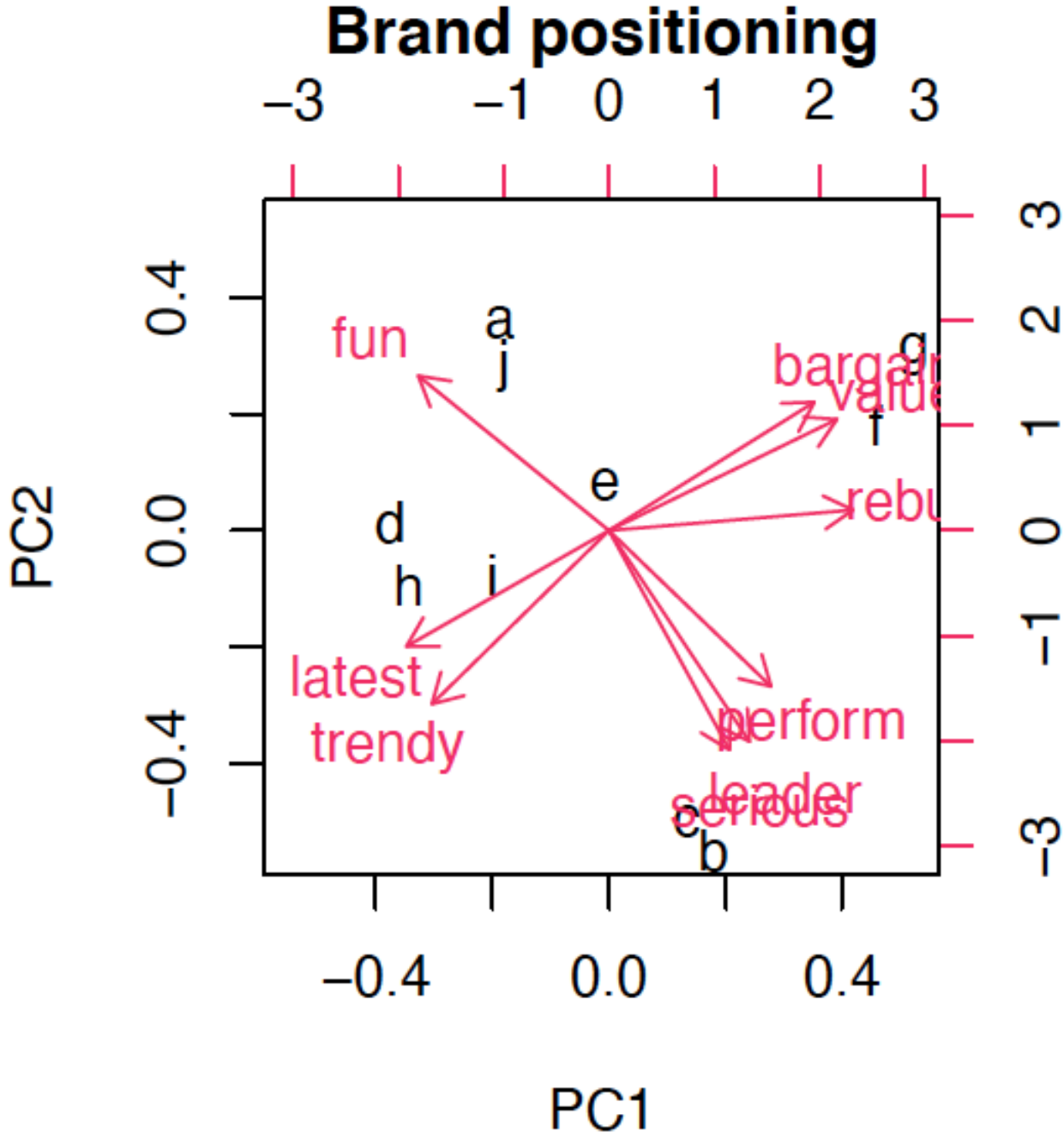
	Group.1	perform	leader	latest	fun	serious
a	a	-0.88591874	-0.5279035	0.4109732	0.6566458	-0.91894067
b	b	0.93087022	1.0707584	0.7261069	-0.9722147	1.18314061
c	c	0.64992347	1.1627677	-0.1023372	-0.8446753	1.22273461
d	d	-0.67989112	-0.5930767	0.3524948	0.1865719	-0.69217505
e	e	-0.56439079	0.1928362	0.4564564	0.2958914	0.04211361
f	f	-0.05868665	0.2695106	-1.2621589	-0.2179102	0.58923066
g	g	0.91838369	-0.1675336	-1.2849005	-0.5167168	-0.53379906
h	h	-0.01498383	-0.2978802	0.5019396	0.7149495	-0.14145855
i	i	0.33463879	-0.3208825	0.3557436	0.4124989	-0.14865746
j	j	-0.62994504	-0.7885965	-0.1543180	0.2849595	-0.60218870

	bargain	value	trendy	rebuy
a	0.21409609	0.18469264	-0.52514473	-0.59616642
b	0.04161938	0.15133957	0.74030819	0.23697320
c	-0.60704302	-0.44067747	0.02552787	-0.13243776
d	-0.88075605	-0.93263529	0.73666135	-0.49398892
e	0.55155051	0.41816415	0.13857986	0.03654811
f	0.87400696	1.02268859	-0.81324496	1.35699580
g	0.89650392	1.25616009	-1.27639344	1.3609257
h	-0.73827529	-0.78254646	0.86430070	-0.60402622
i	-0.25459062	-0.80339213	0.59078782	-0.20317603
j	-0.09711188	-0.07379367	-0.48138267	-0.96164748

```
> brand.mu.pc <- prcomp(brand.mean[, 2:10], scale = TRUE)
> summary(brand.mu.pc)

Importance of components:
                          PC1    PC2    PC3     PC4     PC5
Standard deviation     2.1345 1.7349 0.7690 0.61498 0.50983
Proportion of Variance 0.5062 0.3345 0.0657 0.04202 0.02888
Cumulative Proportion  0.5062 0.8407 0.9064 0.94842 0.97730
                          PC6     PC7     PC8     PC9
 Standard deviation    0.36662 0.21506 0.14588 0.04867
Proportion of Variance 0.01493 0.00514 0.00236 0.00026
Cumulative Proportion  0.99223 0.99737 0.99974 1.00000
```

> biplot(brand.mu.pc, main = "Brand positioning")



Interpretation

What does the map tell us? First we interpret the adjective clusters and relationships and see four areas with well differentiated sets of adjectives and brands that are positioned in proximity. Brands f and g are high on value, for instance, while a and j are relatively high on fun, which is opposite in direction from leadership adjectives (leader and serious).

Let suppose that you are the brand manager for brand *e*. What does the map tell you?

Your brand is in the center and thus appears not to be well-differentiated on any of the dimensions. That could be good or bad, depending on your strategic goals.

If your goal is to be a safe brand that appeals to many consumers, then a relatively undifferentiated position like e could be desirable.

On the other hand, if you wish your brand to have a strong, differentiated perception, this finding would be unwanted.

Let suppose that you are the brand manager for brand *e*. What does the map tell you?

Your brand is in the center and thus appears not to be well-differentiated on any of the dimensions. That could be good or bad, depending on your strategic goals.

If your goal is to be a safe brand that appeals to many consumers, then a relatively undifferentiated position like e could be desirable.

On the other hand, if you wish your brand to have a strong, differentiated perception, this finding would be unwanted.

Let suppose that you are the brand manager for brand *e*. What does the map tell you?

Your brand is in the center and thus appears not to be well-differentiated on any of the dimensions. That could be good or bad, depending on your strategic goals.

If your goal is to be a safe brand that appeals to many consumers, then a relatively undifferentiated position like e could be desirable.

On the other hand, if you wish your brand to have a strong, differentiated perception, this finding would be unwanted.

Let suppose that you are the brand manager for brand *e*. What does the map tell you?

Your brand is in the center and thus appears not to be well-differentiated on any of the dimensions. That could be good or bad, depending on your strategic goals.

If your goal is to be a safe brand that appeals to many consumers, then a relatively undifferentiated position like e could be desirable.

On the other hand, if you wish your brand to have a strong, differentiated perception, this finding would be unwanted.

Suppose you wanted to move in the direction of brand c. You could look at the specific differences from c in the data:

```
> brand.mean <- aggregate(. ~ brand, data = brand.sc, mean)
> brand.mean[3, -1] - brand.mean[5, -1]

   perform    leader     latest       fun  serious
3 1.214314 0.9699315 -0.5587936 -1.140567 1.180621
  bargain      value    trendy      rebuy
-1.158594 -0.8588416 -0.113052 -0.1689859
```

This shows you that *e* is relatively stronger than *c* on value and fun, which suggests dialing down messaging or other attributes that reinforce those. Similarly, *c* is stronger on perform and serious.