Factor & Components Analysis

Lecture 8

Multivariate statistics

Psychology 613 – Spring 2022

Why use factor and components analysis?

1. Identify sets of variables that are *convergent* and *discriminant* (both)

2. Test/*confirm* a specific measurement model
   → Factor analysis

3. Reduce the dimensionality of a data set
   → Component analysis

Differences between FA and CA

Components analysis	Factor analysis
Data driven	Hypothesis driven
Model free	Model based (SEM)
No latent variables	Latent and observed vars
Orthogonal or oblique	Oblique only
Arbitrary number of components	Number of factors specified in advance
No unique solution	Unique solution possible
Exploratory	Confirmatory

Selection primarily depends on the phase of your research

(Principal) components analysis

Purpose

Summarize the patterns of correlations / covariances among a large set of variables

Steps

Select / measure variables, compute correlation matrix, extract components from matrix, rotate components, interpret

(Principal) components analysis

<u>Output</u>

A regression-like equation that combines the scores on each variable into one or more *components* that explains as much of the variance as possible.

The “betas” in this equation are called ***component scores***.

PCA: The math

Analytic solution to PCA based on *eigenvectors* and *eigenvalues* (“own” in German)

A vector **V** is an eigenvector of **X** if:

$$\mathbf{XV} = \lambda\mathbf{V}$$

Where λ is a scalar called the “eigenvalue”

PCA: The math

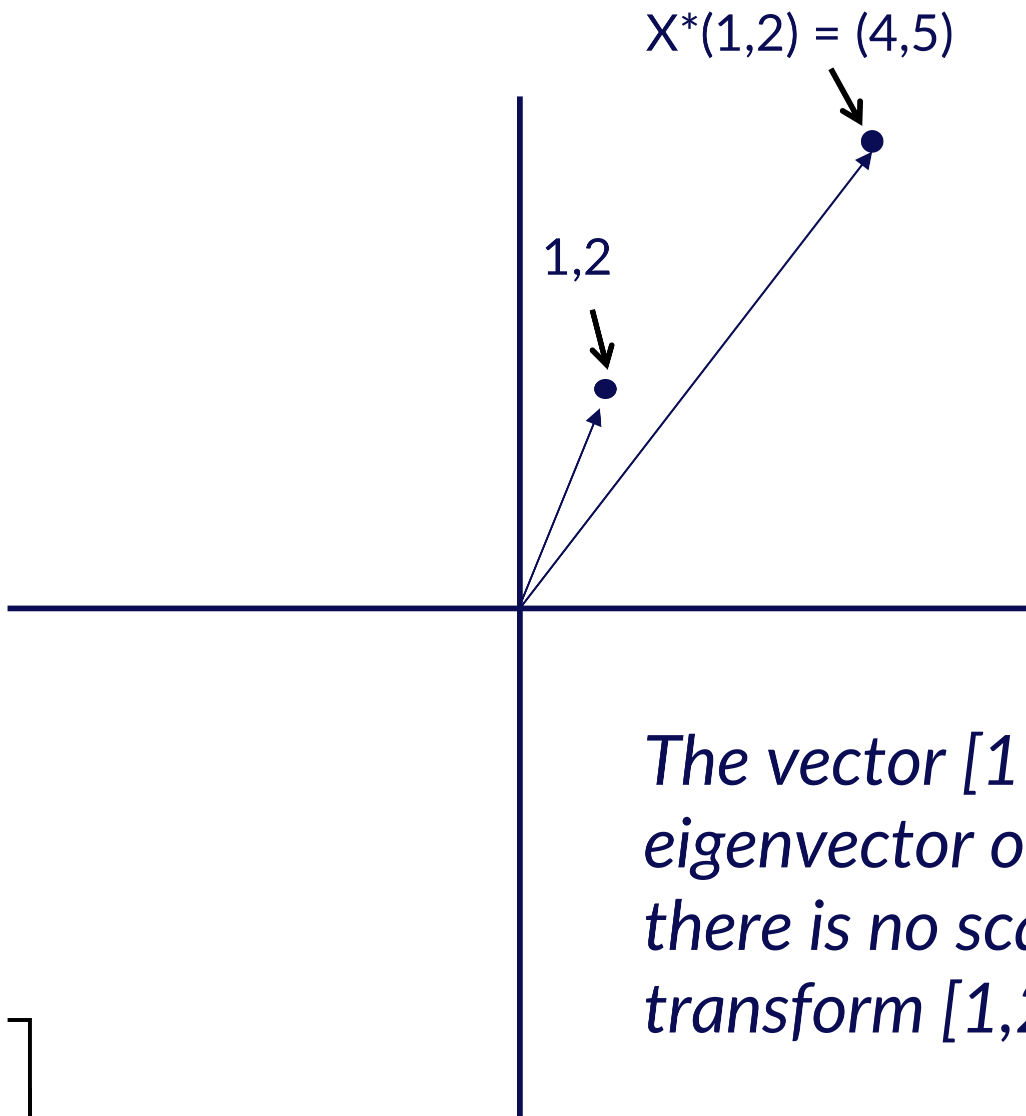
Suppose X = [2 1 ; 2 1]

Then the vector **V** is an eigenvector of X because:

$$X * V = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} * \begin{bmatrix} -\sqrt{2}/2 \\ \sqrt{2}/2 \end{bmatrix} = 1 * \begin{bmatrix} -\sqrt{2}/2 \\ \sqrt{2}/2 \end{bmatrix}$$

Where $\lambda=1$

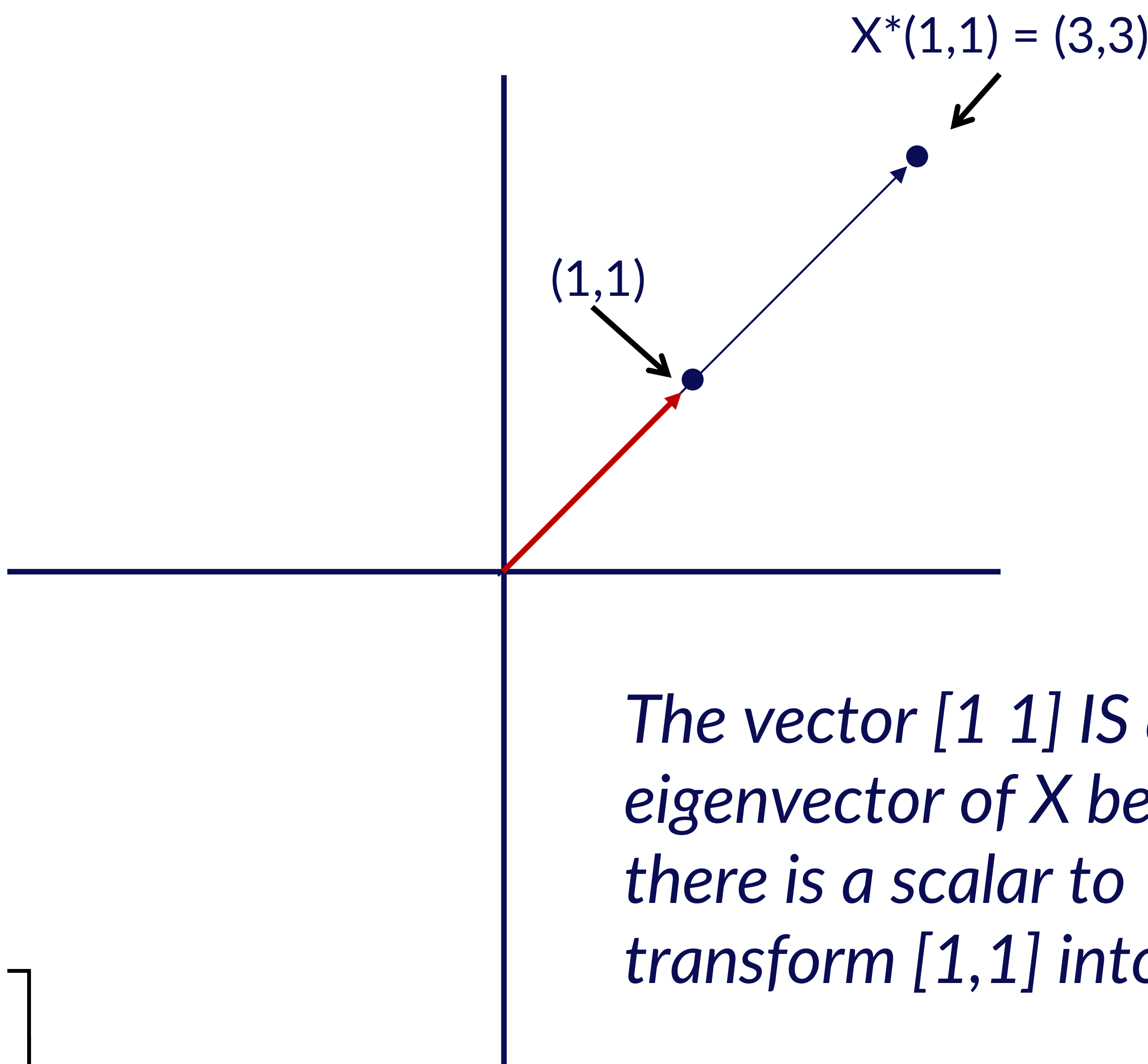
Geometrical interpretation



The vector [1 2] is NOT an eigenvector of X because there is no scalar to transform [1,2] into [4,5]

$$X = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

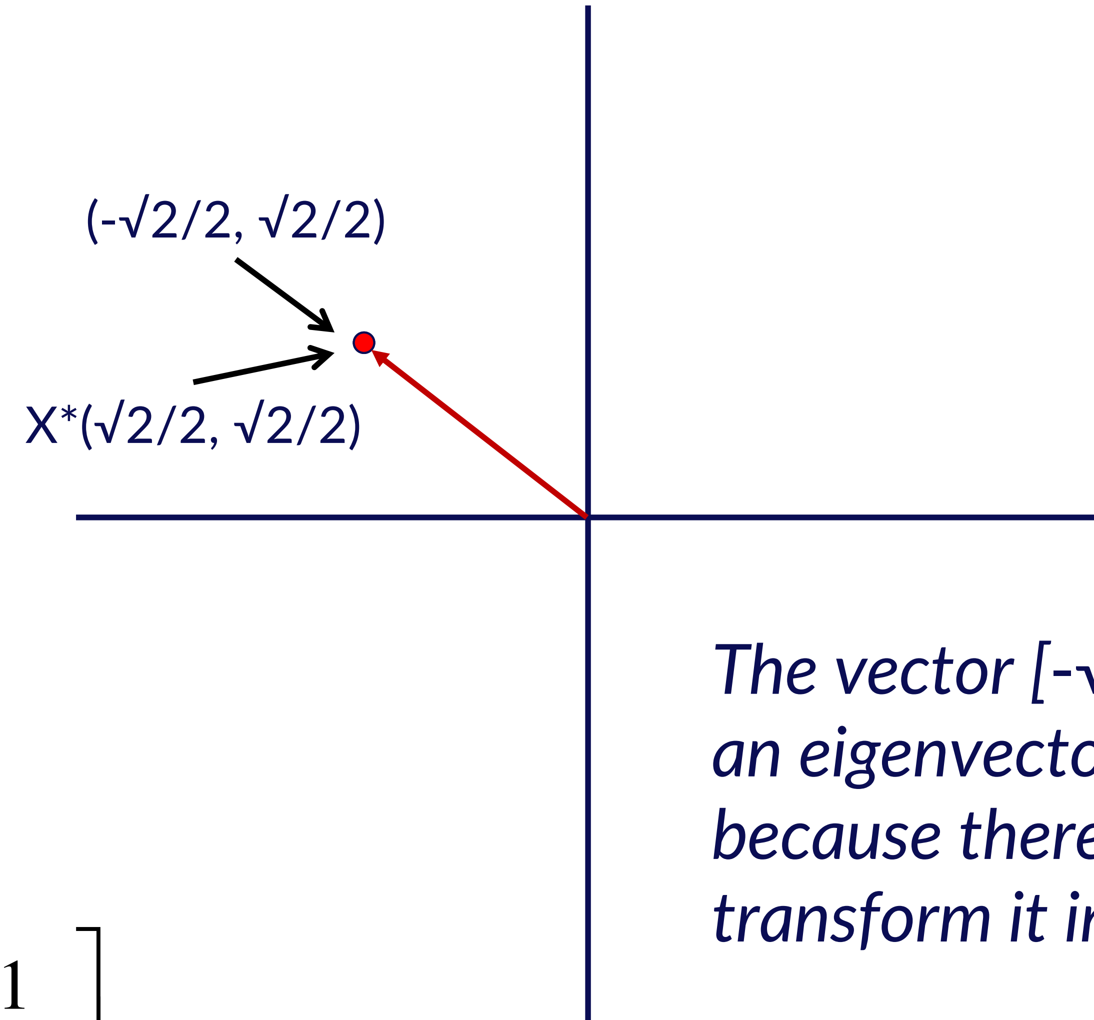
Geometrical interpretation



The vector [1 1] IS an eigenvector of X because there is a scalar to transform [1,1] into [3,3]: 3

$$X = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

Geometrical interpretation



The vector [-√2/2 √2/2] IS an eigenvector of X because there is a scalar to transform it into itself: 1

$$X = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

In R

```
> require(OpenMx)
> X <- matrix(c(2,1,1,2), nrow=2, byrow=T)
> eigen(X)
```

```
> X
     [,1] [,2]
[1,]    2    1
[2,]    1    2
> eigen(X)
$values
[1] 3 1

$vectors
          [,1]       [,2]
[1,] 0.7071068 -0.7071068
[2,] 0.7071068  0.7071068
```

The two eigenvalues are 1 and 3.

The complete set of eigenvectors is the two columns of "vectors".

Properties of eigen-stuffs

1. Eigvector * Eigvector’ = Identity

2. Eigvalue = Eigvector’ * X * Eigvector

3. **Original X =**
   Eigvector * Eigvalue * Eigvector’

A simple example

Subject	Cost of ticket	Lift speed	Powder depth	Powder moisture
S1	32	64	65	67
S2	61	37	62	65
S3	59	40	45	43
S4	36	62	34	35
S5	62	46	43	40

Correlations

	Cost of ticket	Lift speed	Powder depth	Powder moisture
Cost	1	-.953	-.055	-.13
Speed		1	-.091	-.036
Depth			1	.99
Moisture				1

Calculate the eigen-stuffs

> eigen(R)

The first two eigenvalues are by far the largest (usually use a cutoff of 1)

```
> eigs
$values
[1] 2.016305104 1.941513814 0.037812306 0.004368776

$vectors
           [,1]       [,2]       [,3]       [,4]
[1,]  0.3524130  0.6143298  0.6624913  0.2439451
[2,] -0.2511248 -0.6637642  0.6758934  0.1988000
[3,] -0.6273987  0.3222291  0.2754625 -0.6531919
[4,] -0.6473888  0.2796147 -0.1685044  0.6887014
```

The first two eigenvalues correspond to the first two columns of the eigenvector matrix

The component loading matrix

The **component loading matrix** describes the relationship between each of the 4 variables and the principal components

Component loading matrix = eigvector * sqrt(eigvals)

Conceptually, this is just the vectors weighted according to the size of the eigenvalues

Calculate the component loadings

> loadings = eigvec %*% sqrt(eigval)

```
> eigval <- vec2diag(eigs$values)
> eigvec <- eigs$vectors
> loadings <- eigvec %*% sqrt(eigval)
> loadings
           [,1]       [,2]        [,3]        [,4]
[1,]  0.5004147  0.8559962  0.12882400  0.01612397
[2,] -0.3565888 -0.9248772  0.13143009  0.01314003
[3,] -0.8908852  0.4489882  0.05356475 -0.04317384
[4,] -0.9192705  0.3896101 -0.03276633  0.04552090
```

Calculate the component loadings

By the “Kaiser rule”, only look at factors whose eigenvalue is > 1 (they were 1.94, 2.02, 0.04, and 0)

> loadings = loadings[,c(1:2)]

```
> loadings[,c(1:2)]
           [,1]       [,2]
[1,]  0.5004147  0.8559962
[2,] -0.3565888 -0.9248772
[3,] -0.8908852  0.4489882
[4,] -0.9192705  0.3896101
```

Calculate the component loadings

These first two vectors correspond to the two components that explain the most variance

loadings =

-0.5004	**-0.8560**	COST
0.3566	**0.9249**	LIFT SPEED
0.8909	-0.4490	SNOW DEPTH
0.9193	-0.3896	MOISTURE
Comp 1	Comp 2	

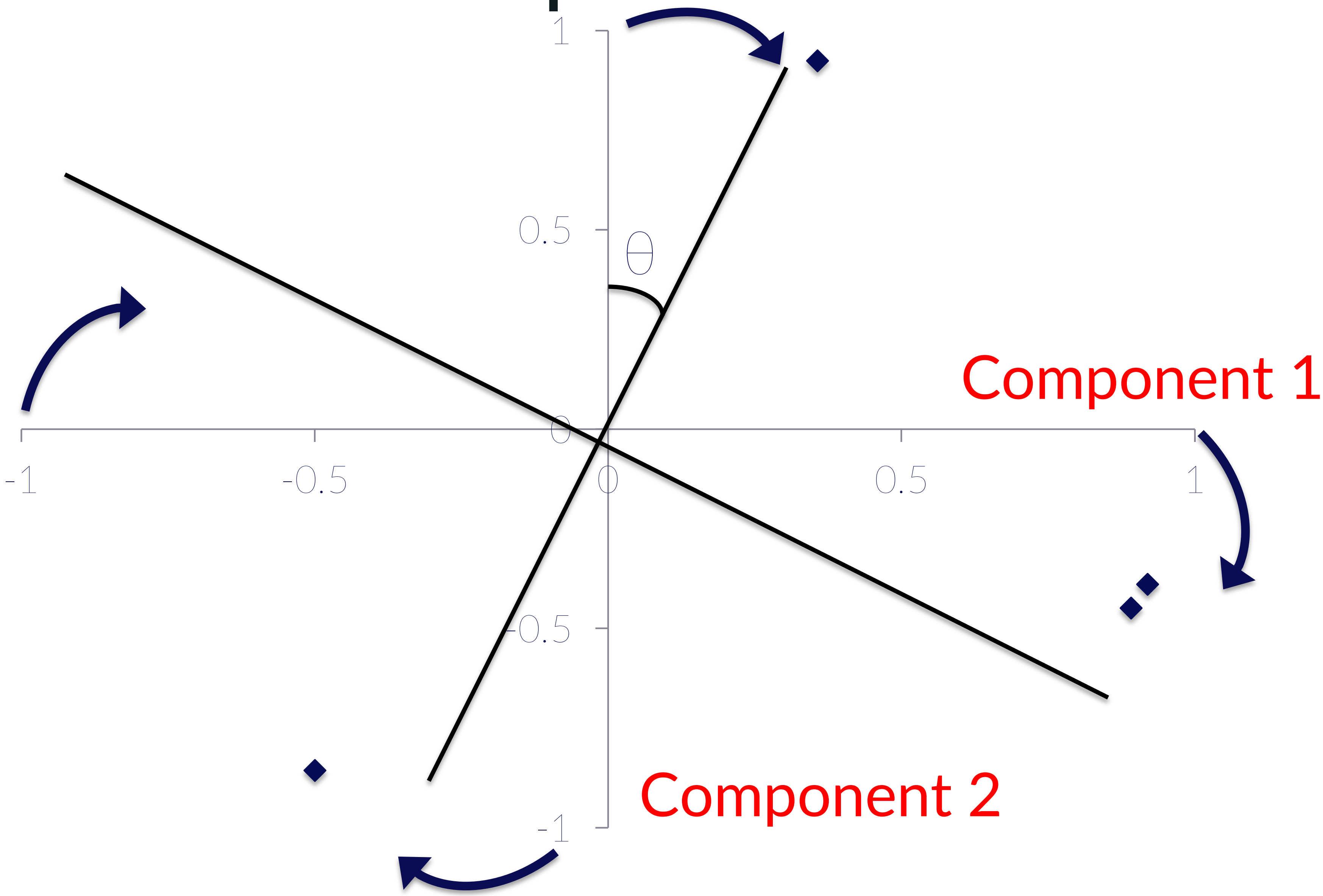
Rotation

Transform the components with respect to the variables to maximize the high loadings and minimize the low loadings to simplify the factors.

Varimax: Algorithm that maximizes the variance of loadings between items on a component

Note: does not actually improve FIT as the eigenvariates don't change. All orthogonal rotations are mathematically equivalent. Rotation is only to aid in interpretation.

Rotation: Graphical interpretation



Rotation: Matrix formulation

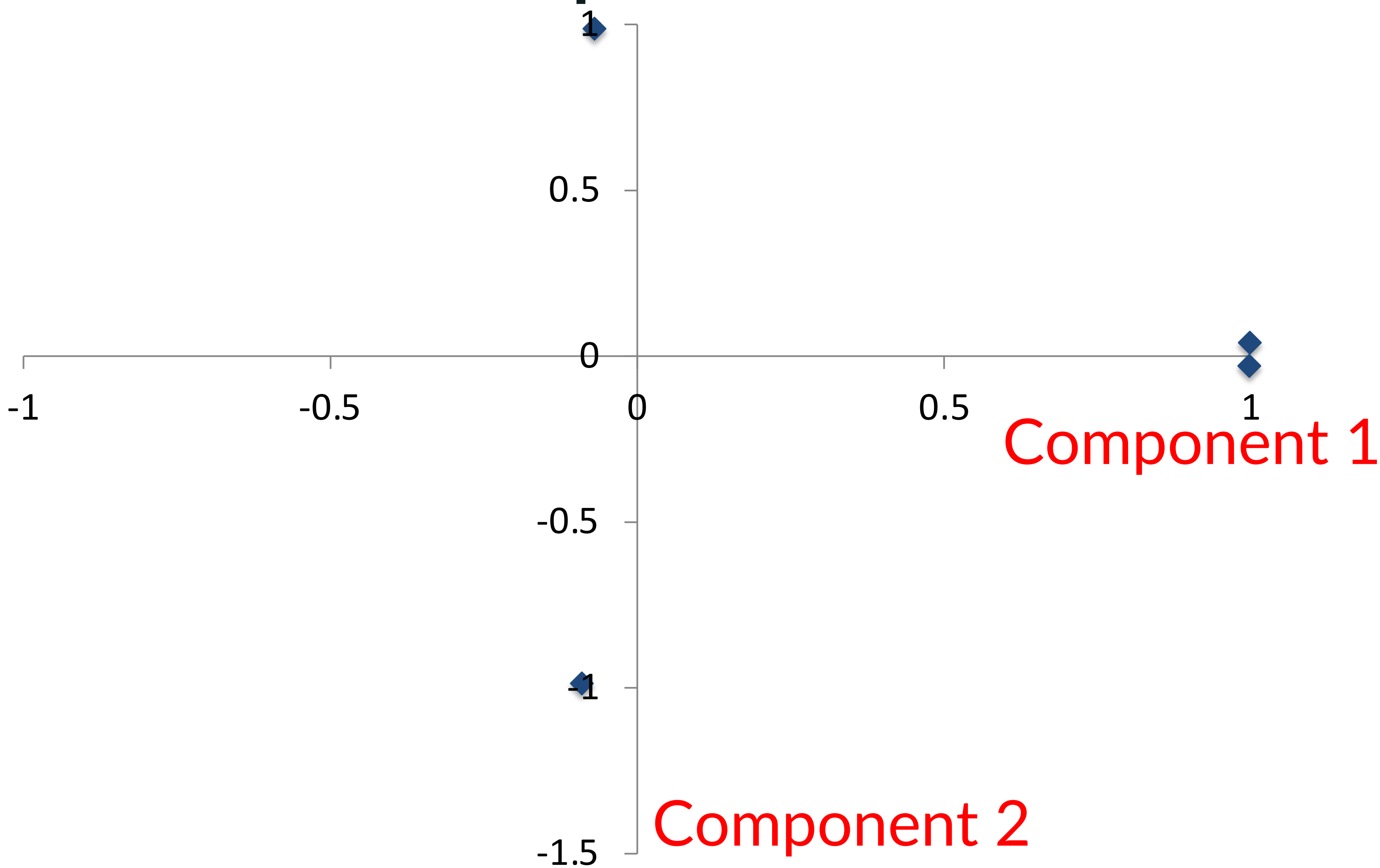
Rotated component loadings = loadings*$\boldsymbol{\Theta}$

Where $\boldsymbol{\Theta}$ (capital theta) = $\begin{bmatrix} \cos(\theta) & \sin(\theta) \\ \sin(\theta) & -\cos(\theta) \end{bmatrix}$

In this case, θ=0.44 (~25 degrees), so rotated

$$= \begin{bmatrix} -.50 & -.86 \\ .36 & .93 \\ .89 & -.45 \\ .92 & -.39 \end{bmatrix} \begin{bmatrix} \cos(.44) & \sin(.44) \\ \sin(.44) & -\cos(.44) \end{bmatrix} = \begin{bmatrix} -.09 & -.987 \\ -.07 & .988 \\ .997 & -.03 \\ .998 & .04 \end{bmatrix}$$

Rotation: Graphical interpretation



Alternative rotations

Orthogonal: Components uncorrelated

Use *varimax* to maximize item-to-item variance on each factor

Oblique: Components correlated

Use *direct oblimin* (with *delta* as the parameter) to minimize cross-products of loadings, OR

Promax, which starts with varimax, then raises loadings to powers (*kappa*) to simplify structure

Interpretation

Remember, these are not the same as *latent* factors. They are just linear combinations of the items.

Don't fall victim to the *reification fallacy*: the belief that a hypothetical construct must correspond to a real thing.

Usually, people label components based on the variables that load most highly on them.

Accounting for variance

	Comp 1	Comp 2	Communality (h^2)
COST	-.09	-.987	Σa^2=.97
SPEED	-.07	.988	Σa^2=.96
DEPTH	.997	-.03	Σa^2=.989
MOISTURE	.998	.04	Σa^2=.996
Sum of squared loading	Σa^2=1.994	Σa^2=1.919	3.915
Proportion of variance (#vars)	.50	.48	.98
Proportion of covariance (h^2)	.51	.49	

Residuals

Recall that the original matrix is given by:

X = Eigvec * Eigval * Eigvec’

This will reproduce the original X exactly if all of the eigenvectors are used

But without all of them, we get an approximation…

Residuals

eigvec =

-0.6143	-0.3524
0.6638	0.2511
-0.3222	0.6274
-0.2796	0.6474

eigval = [1.9415; 2.0163]

X (approximated) = eigvec * eigval * eigvec'

Residual correlation matrix = X – X (approx.)

```
> R
            [,1]        [,2]        [,3]        [,4]
[1,]  1.00000000 -0.95299048 -0.05527555 -0.12999882
[2,] -0.95299048  1.00000000 -0.09110654 -0.03624823
[3,] -0.05527555 -0.09110654  1.00000000  0.99017435
[4,] -0.12999882 -0.03624823  0.99017435  1.00000000
> reproducedR
            [,1]        [,2]        [,3]        [,4]
[1,]  0.98314440 -0.97013370 -0.06147984 -0.12651171
[2,] -0.97013370  0.98255347 -0.09757925 -0.03253989
[3,] -0.06147984 -0.09757925  0.99526684  0.99389478
[4,] -0.12651171 -0.03253989  0.99389478  0.99685421
```

```
> residualCorrs
             [,1]         [,2]         [,3]         [,4]
[1,]  0.016855604  0.017143219  0.006204291 -0.003487112
[2,]  0.017143219  0.017446530  0.006472714 -0.003708336
[3,]  0.006204291  0.006472714  0.004733163 -0.003720433
[4,] -0.003487112 -0.003708336 -0.003720433  0.003145785
> meanResidual
[1] 0.00499936
> sqrt(mean(residualCorrs^2))
[1] 0.009515963
```

Practical advice

Start with principal components with varimax rotation

Experiment with different numbers of components

Try using oblique but don't necessarily keep it

Trial and error is OK—this is not hypothesis testing!

Differences between FA and CA

Components analysis	Factor analysis - **TUESDAY**
Data driven	Hypothesis driven
Model free	Model based (SEM)
No latent variables	Latent and observed vars
Orthogonal or oblique	Oblique only
Arbitrary number of components	Number of factors specified in advance
No unique solution	Unique solution possible
Exploratory	Confirmatory

Selection primarily depends on the phase of your research