

# Ordination

- **Principal component analysis (PCA)**
- **Factor analysis (FA)**
- **Correspondence analysis (CA)**
- **Principal coordinate analysis (PCoA) or multidimensional scaling (MDS)**
- **Non-metric multidimensional scaling (NMDS)**
- **Redundancy analysis (RDA)**
- **Canonical correspondence analysis (CCA)**
- **Generalized Joint Attribute Modeling (GJAM)**



## History of ordination methods

- In 1930, Ramensky began to use informal ordination techniques for vegetation. Such informal and largely subjective methods became widespread in the early 1950's (Whittaker 1967).
- In 1951, Curtis and McIntosh (1951) developed the 'continuum index', which later lead to conceptual links between species responses to gradients and multivariate methods. Shortly thereafter, Goodall (1954) introduced the term 'ordination' in an ecological context for Principal Components Analysis.
- Bray and Curtis (1957) developed polar ordination, which became the first widely-used ordination technique in ecology.
- Austin (1968) used canonical correlation to assess plant-environment relationships in what may have been the first example of multivariate direct gradient analysis in ecology.
- In 1973, Hill introduced correspondence analysis, a technique originating in the 1930's, to ecologists. Correspondence analysis gradually supplanted polar ordination, which today has few practitioners.
- Fasham (1977) and Prentice (1977) independently discovered and demonstrated the utility of Kruskal's (1964) nonmetric multidimensional scaling, originally intended as a psychometric technique, for community ecology.
- Hill (1979) corrected some of the flaws of Correspondence Analysis and thereby created Detrended Correspondence Analysis, which is the most widely used indirect gradient analysis technique today. The software to implement Detrended Correspondence Analysis, DECORANA, became the backbone of many later software packages.
- Gauch's (1982) book "Multivariate Analysis in Community Ecology" described ordination in non-technical terms to the average practitioner, and allowed ordination techniques to enter the mainstream.
- Fuzzy set theory, introduced to ecologists by Roberts (1986), is a promising approach with ties to polar ordination, but has yet to gain many adherents.
- Ter Braak (1986) ushered in the biggest modern revolution in ordination methods with Canonical Correspondence Analysis. This technique coupled Correspondence Analysis with regression methodologies, and provides for hypothesis testing.
- Ter Braak and Prentice (1988) developed a theoretical unification of ordination techniques, hence placing gradient analysis on a firm theoretical foundation.

# Principal Component Analysis (PCA)

Invented by Pearson (1901) and Hotelling (1933)

Use in ecology by Goodall (1954) under the name “factor analysis” (“principal factor analysis” is a synonym of PCA).

# Principal component analysis

Principal component analysis (PCA) is a technique that is useful for the compression and classification of data. The purpose is to **reduce the dimensionality of a data set** (sample) by finding a new set of variables, smaller than the original set of variables, that nonetheless retains most of the sample's information.

By information we mean the variation present in the sample, given by the correlations between the original variables. The new variables, called principal components (PCs), are uncorrelated, and are ordered by the fraction of the total information each retains.

## Example

In this example wildlife (moose) population density was measured over time (once a year) in three areas.

Year	Area 1	Area 2	Area 3
1	11.3	14.1	6.9
2	10.4	14	11.2
3	9.9	13	8.7
4	8.2	11.4	3.3
5	10.1	11.9	8.7
6	10.7	13.8	12.5
7	11	14.9	8.9
8	7.1	8.5	3.7
9	14.7	14.5	12.1
10	5.4	9	4.1
11	7.3	7.6	5.6
12	10.2	10.9	7.3
13	6.1	9.9	6.8
14	9.7	13.2	6.6
15	8.1	9.4	4
16	11.3	11.8	4.9
17	8.8	11.5	8.8
18	9.4	11.6	5.7
19	7.5	11.4	4.9
20	8.8	10.7	7.2
21	7.5	11.1	7
22	9.1	13.2	8.9
23	6.8	9.8	7.6



[https://timgsa.baidu.com/timg?image&quality=80&size=b9999\\_10000&sec=1491476303003&di=f903f1e5bc129bcff1201fa4fdeb8b8&imgtype=0&src=http%3A%2F%2Fwww.bigthings.ca%2Fscotia%2Fpictures%2Fmoose1.jpg](https://timgsa.baidu.com/timg?image&quality=80&size=b9999_10000&sec=1491476303003&di=f903f1e5bc129bcff1201fa4fdeb8b8&imgtype=0&src=http%3A%2F%2Fwww.bigthings.ca%2Fscotia%2Fpictures%2Fmoose1.jpg)

# Habitats



Year	Area 1	Area 2	Area 3
1	11.3	14.1	6.9
2	10.4	14	11.2
3	9.9	13	8.7
4	8.2	11.4	3.3
5	10.1	11.9	8.7
6	10.7	13.8	12.5
7	11	14.9	8.9
8	7.1	8.5	3.7
9	14.7	14.5	12.1
10	5.4	9	4.1
11	7.3	7.6	5.6
12	10.2	10.9	7.3
13	6.1	9.9	6.8
14	9.7	13.2	6.6
15	8.1	9.4	4
16	11.3	11.8	4.9
17	8.8	11.5	8.8
18	9.4	11.6	5.7
19	7.5	11.4	4.9
20	8.8	10.7	7.2
21	7.5	11.1	7
22	9.1	13.2	8.9
23	6.8	9.8	7.6

# The Sample Statistics

$$\vec{\bar{x}} = \begin{bmatrix} 9.10 \\ 11.62 \\ 7.19 \end{bmatrix}$$

```
A = A[, c(2:4)] # The covariance matrix
apply(A, 2, mean) # mean
```

$$S = \begin{bmatrix} 4.297 & 3.307 & 3.295 \\ & 4.015 & 3.527 \\ & & 6.566 \end{bmatrix}$$

```
# variance-covariance matrix
S = var(A)
```

$$R = \begin{bmatrix} 1 & .796 & .620 \\ & 1 & .687 \\ & & 1 \end{bmatrix}$$

```
cor(A) # correlation matrix
```

```
A = read.table(header = T, text = "
Year   Area1   Area2   Area3
1      11.3    14.1    6.9
2      10.4    14      11.2
3       9.9    13      8.7
4       8.2    11.4    3.3
5      10.1    11.9    8.7
6      10.7    13.8    12.5
7       11     14.9    8.9
8       7.1     8.5    3.7
9      14.7    14.5    12.1
10      5.4     9      4.1
11      7.3     7.6    5.6
12     10.2    10.9    7.3
13      6.1     9.9    6.8
14      9.7    13.2    6.6
15      8.1     9.4     4
16     11.3    11.8    4.9
17      8.8    11.5    8.8
18      9.4    11.6    5.7
19      7.5    11.4    4.9
20      8.8    10.7    7.2
21      7.5    11.1     7
22      9.1    13.2    8.9
23      6.8     9.8    7.6 ")
```

# Eigenvalues and eigenvectors $\text{eigen}(S)$

$$S = \begin{bmatrix} 4.297 & 3.307 & 3.295 \\ & 4.015 & 3.527 \\ & & 6.566 \end{bmatrix}$$

The eigenvalues of  $S$

$$\lambda_1 = 11.85974, \quad \lambda_2 = 2.204232, \quad \lambda_3 = 0.814249$$

The eigenvectors of  $S$

$$a_1 = \begin{bmatrix} 0.522 \\ 0.523 \\ 0.674 \end{bmatrix} \quad a_2 = \begin{bmatrix} -0.582 \\ -0.359 \\ 0.730 \end{bmatrix} \quad a_3 = \begin{bmatrix} 0.624 \\ -0.773 \\ 0.117 \end{bmatrix}$$

The principal components

$$C_1 = 0.522X_1 + 0.523X_2 + 0.674X_3$$

$$C_2 = -0.582X_1 - 0.359X_2 + 0.730X_3$$

$$C_3 = 0.624X_1 - 0.733X_2 + 0.117X_3$$

	Comp.1	Comp.2	Comp.3
[1.]	-2.2489	2.3813	-0.5841
[2.]	-4.6230	-1.3164	-0.5654
[3.]	-2.1549	-0.1418	-0.3964
[4.]	3.2071	2.2356	-0.8506
[5.]	-1.6840	-0.4203	0.5787
[6.]	-5.5508	-2.1624	-0.0718
[7.]	-3.8578	1.0343	-1.1559
[8.]	5.0289	0.2627	0.7520
[9.]	-7.7363	0.7082	1.8344
[10.]	5.3857	-0.8389	-0.6477
[11.]	4.1154	-1.3306	1.7945
[12.]	-0.2701	0.3006	1.2504
[13.]	2.7307	-2.0790	-0.5914
[14.]	-0.7405	1.3462	-0.9211
[15.]	3.8339	0.9487	0.7149
[16.]	0.3013	3.0153	0.9601
[17.]	-0.8633	-1.3932	0.0890
[18.]	0.8592	1.2541	0.0235
[19.]	2.4949	0.6607	-1.1001
[20.]	0.6329	-0.5127	0.5204
[21.]	1.2373	-0.9796	-0.6227
[22.]	-1.9765	-0.6814	-1.0264
[23.]	1.8786	-2.2914	0.0159



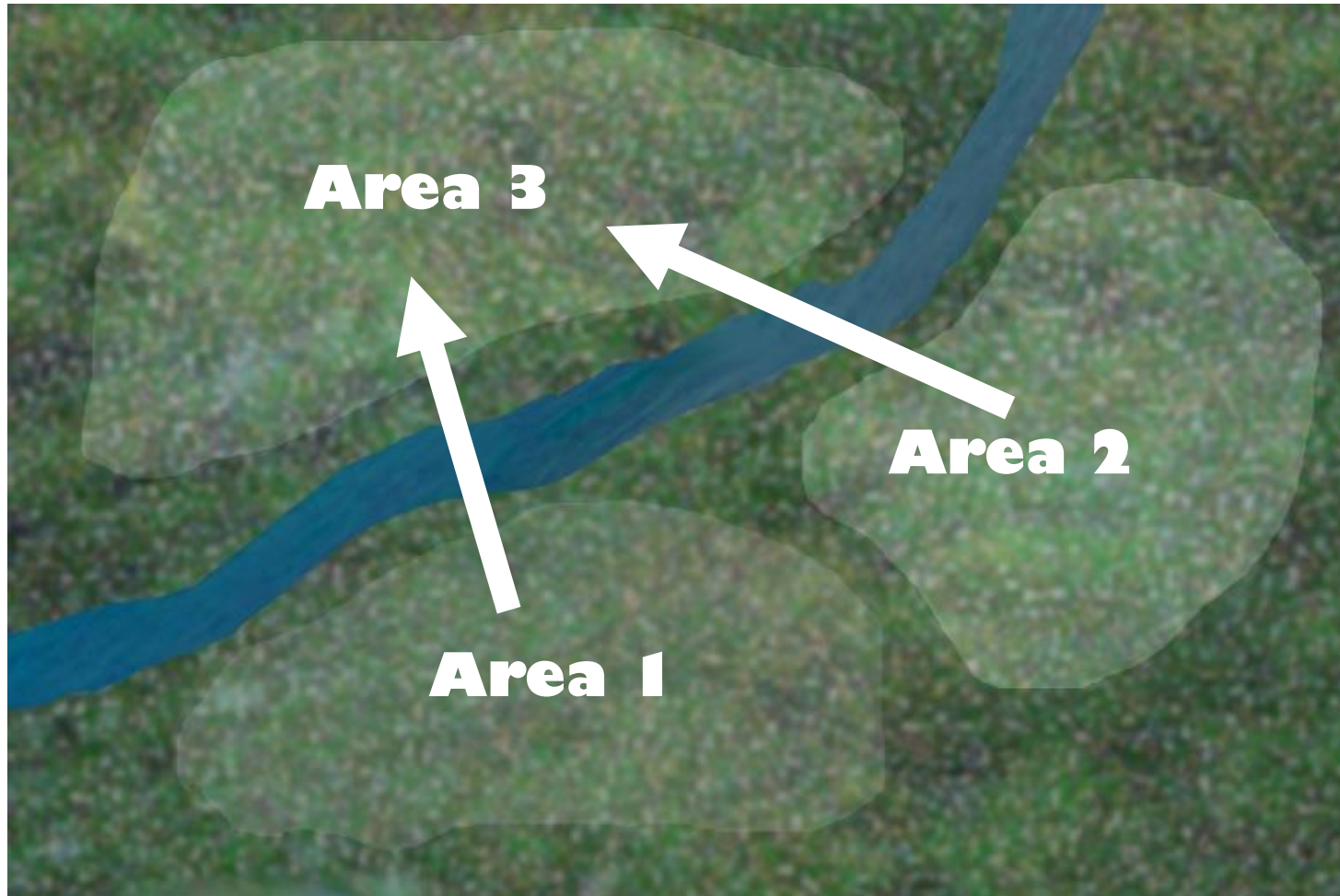
# Example 1/3

$$C_1 = 0.522X_1 + 0.523X_2 + 0.674X_3$$



## Example 2/3

$$C_2 = -0.582X_1 - 0.359X_2 + 0.730X_3$$



## Example 3/3

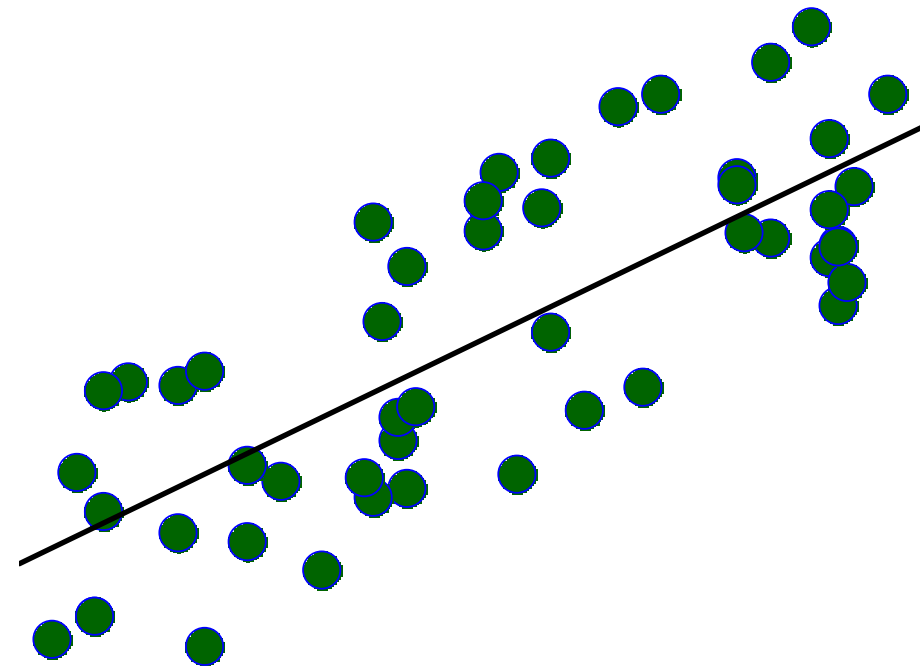
$$C_3 = .624x_1 - .733x_2 + .117x_3$$





# Principal Component Analysis

- Given  $m$  points in a  $n$  dimensional space, for large  $n$ , how does one project on to a 1 dimensional space?
- Choose a line that fits the data so the points are spread out well along the line.

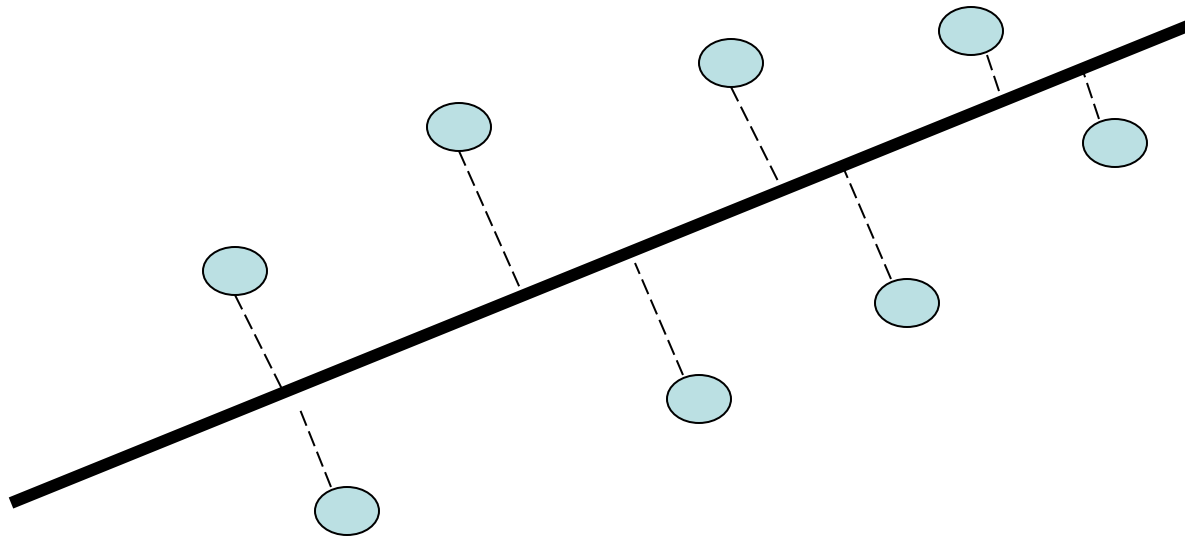


```
X = sample(1:100, 50, rep = TRUE)
# Y = X + rnorm(50, 0, 20)
Y = jitter(X, factor = 1, amount = 40)
```

```
plot(X, Y, pch = 21, cex = 3, col = "blue",
      bg = "darkgreen", lwd = 1,
      axes = FALSE, xlab = "", ylab = "")
abline(lm(Y~X), lwd=3)
```

# Principal Component Analysis

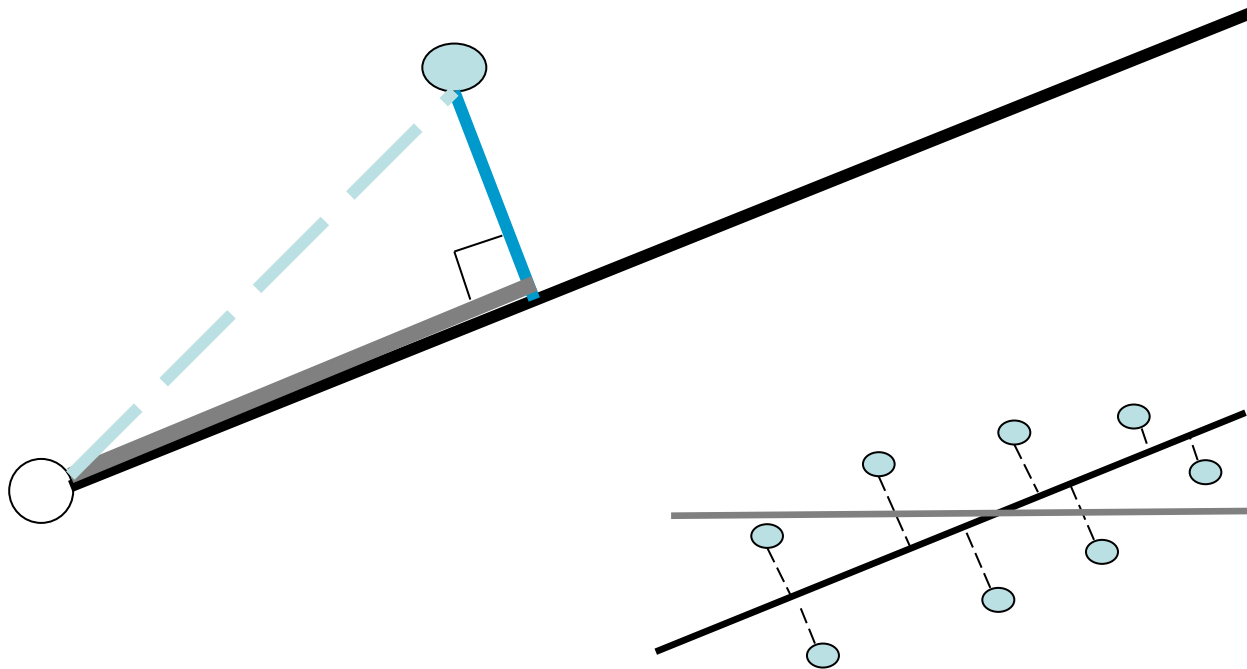
- Formally, minimize sum of squares of distances to the line.



- Why sum of squares? Because it allows fast minimization.

# Principal Component Analysis

- For one data point and a line through point  $(0,0)$ , minimizing sum of squares of distances to the line is the same as maximizing the sum of squares of the projections on that line (Pythagoras, long ago)



# PCA: General methodology

From  $k$  original variables:  $x_1, x_2, \dots, x_k$ :

Produce  $k$  new variables:  $y_1, y_2, \dots, y_k$ :

$$y_1 = a_{11}x_1 + a_{12}x_2 + \dots + a_{1k}x_k$$

$$y_2 = a_{21}x_1 + a_{22}x_2 + \dots + a_{2k}x_k$$

...

$$y_k = a_{k1}x_1 + a_{k2}x_2 + \dots + a_{kk}x_k$$

$y_k$ 's are  
**Principal Components**

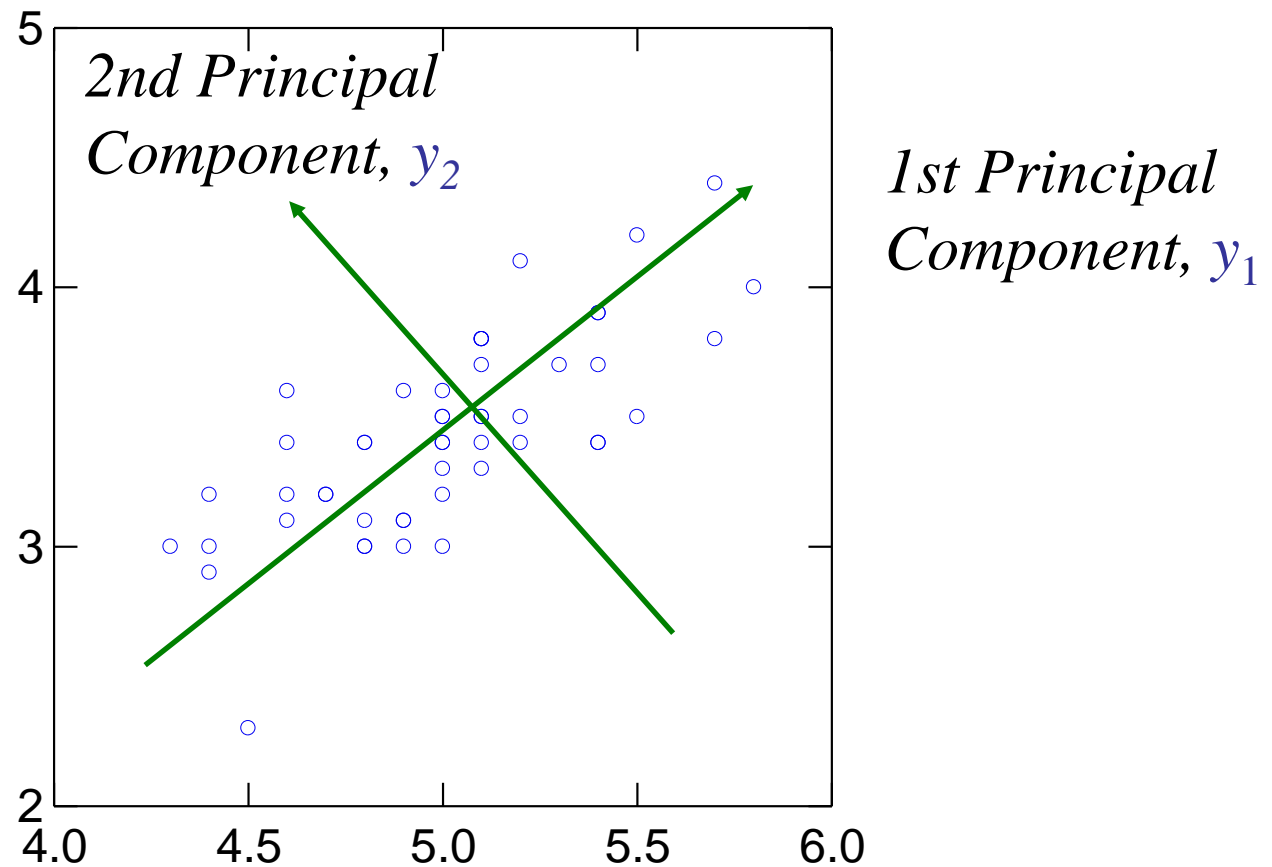
*such that:*

$y_k$ 's are uncorrelated (orthogonal)

$y_1$  explains as much as possible of original variance in data set

$y_2$  explains as much as possible of remaining variance

# Graphical interpretation





# Eigenvalues

Amount of **variance accounted for** by:

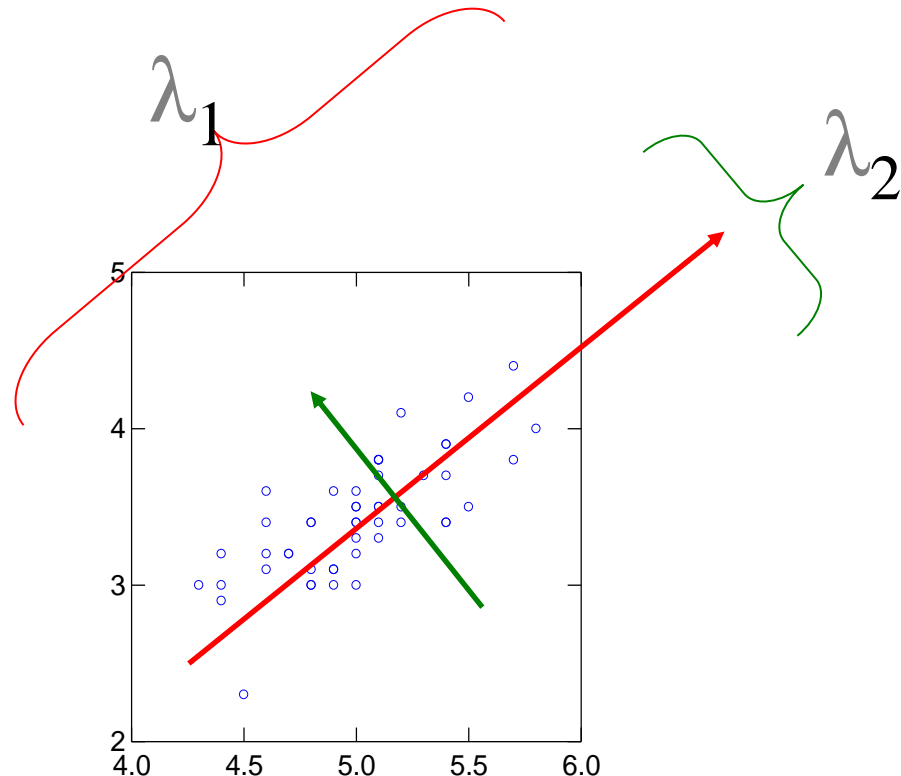
1st principal component,  $\lambda_1$ , 1st **eigenvalue**

2nd principal component,  $\lambda_2$ , 2nd **eigenvalue**

...

$$\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \lambda_4 \geq \dots$$

Average  $\lambda_j = 1$  if the matrix is standardized.



# Eigenvectors

$\{a_{11}, a_{12}, \dots, a_{1k}\}$  is 1st **Eigenvector** of covariance matrix,  
and **coefficients** of first principal component

$\{a_{21}, a_{22}, \dots, a_{2k}\}$  is 2nd **Eigenvector** of covariance matrix,  
and **coefficients** of 2nd principal component

...

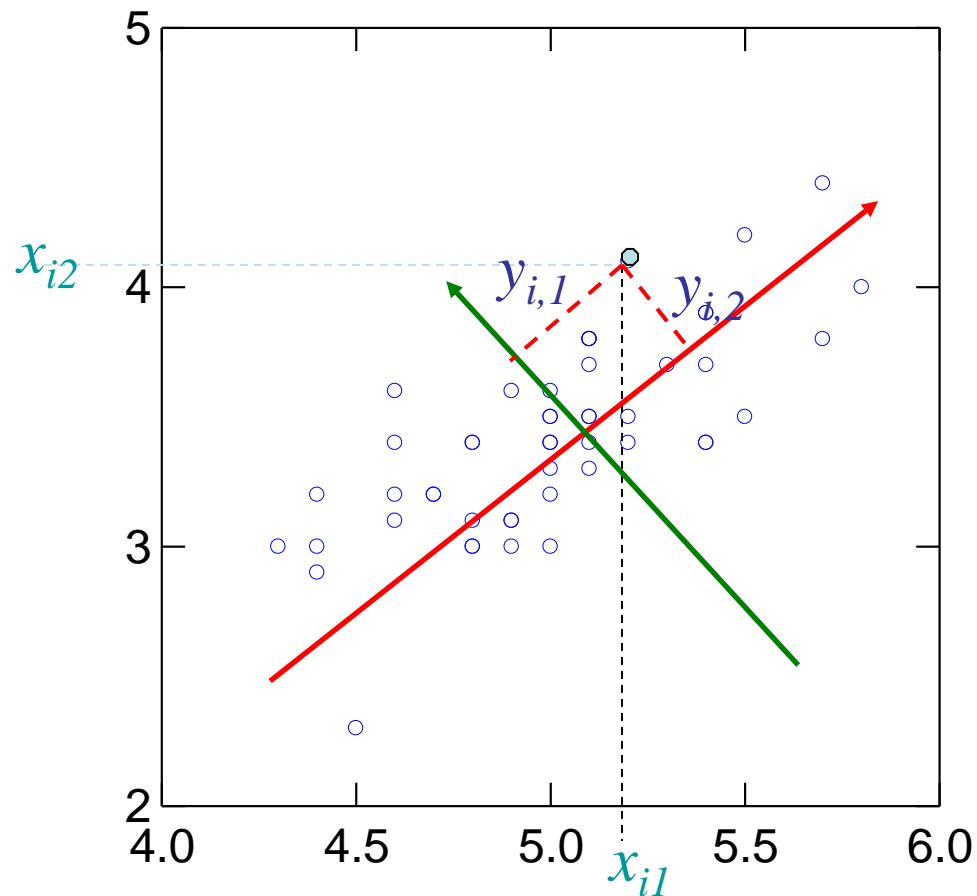
$\{a_{k1}, a_{k2}, \dots, a_{kk}\}$  is  $k$ th **Eigenvector** of covariance matrix,  
and **coefficients** of  $k$ th principal component

# PCA: Terminology

- $j$ th **principal component** is linear combination of all variables
- **coefficients**,  $a_{jk}$ , are elements of eigenvectors and relate original variables (standardized if using correlation matrix) to components
- **scores** are values of units on components (produced using coefficients)
- **amount of variance accounted for** by component is given by eigenvalue,  $\lambda_j$
- **proportion of variance accounted for** by component is given by  $\lambda_j / \sum \lambda_j$
- **loading** of  $k$ th original variable on  $j$ th component is given by  $a_{jk}$   
correlation between variable and component

# Scores

**Score** of  $i$ th unit on  $j$ th principal component



# R code: PCA

`head(mtcars)`

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda_RX4	21	6	160	110	3.9	2.62	16.46	0	1	4	4
Mazda_RX4_Wag	21	6	160	110	3.9	2.875	17.02	0	1	4	4
Datsun_710	22.8	4	108	93	3.85	2.32	18.61	1	1	4	1
Hornet_4_Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
Hornet_Sportabout	18.7	8	360	175	3.15	3.44	17.02	0	0	3	2
Valiant	18.1	6	225	105	2.76	3.46	20.22	1	0	3	1

**##### PCA #####**

**## vary by orders of magnitude, so scaling is appropriate**

`pca1 <- princomp(mtcars) # inappropriate`

**# use the correlation matrix NOT the covariance matrix**

**# =^= prcomp(iris.pre, scale=TRUE)**

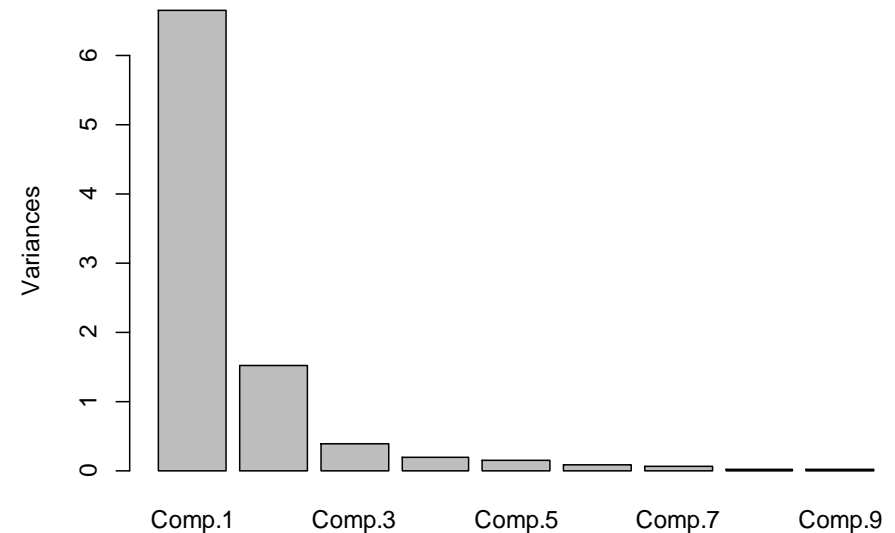
`pca2 <- princomp(mtcars, cor = TRUE)`

<b>mpg</b>	Miles/(US) gallon
<b>cyl</b>	Number of cylinders
<b>disp</b>	Displacement (cu.in.)
<b>hp</b>	Gross horsepower
<b>drat</b>	Rear axle ratio
<b>wt</b>	Weight (lb/1000)
<b>qsec</b>	1/4 mile time
<b>vs</b>	V/S
<b>am</b>	Transmission (0 = automatic, 1 = manual)
<b>gear</b>	Number of forward gears
<b>carb</b>	Number of carburetors

# PCA scree plot

```
cars = mtcars[1:21, ]
cars = cars[-c(9:13), ]
cars = cars[, c(1:7, 10,11)]
head(cars)
pca <- princomp(cars, cor = TRUE)
plot(pca) # shows a scree plot

summary(pca2)
pca2$loadings # eigen vector
pca2$scores # principal component vector
```



	Comp.1	Comp.2	Comp.3	Comp.4	Comp.5	Comp.6	Comp.7	Comp.8	Comp.9
Standard deviation	2.578	1.23	0.6143	0.4374	0.3643	0.26917	0.21161	0.1298	0.092411
Proportion of Variance	0.738	0.168	0.0419	0.0213	0.0147	0.00805	0.00498	0.00187	0.000949
Cumulative Proportion	0.738	0.906	0.9482	0.9694	0.9842	0.9922	0.99718	0.99905	1.000000

# PCA biplot

```
out = summary(pca)
```

	Comp.1	Comp.2	Comp.3	Comp.4	Comp.5	Comp.6	Comp.7	Comp.8	Comp.9
Standard deviation	2.578	1.23	0.6143	0.4374	0.3643	0.26917	0.21161	0.1298	0.092411
Proportion of Variance	0.738	0.168	0.0419	0.0213	0.0147	0.00805	0.00498	0.00187	0.000949
Cumulative Proportion	0.738	0.906	0.9482	0.9694	0.9842	0.9922	0.99718	0.99905	1.000000

```
Prop.var1 = out[[1]][1]^2/sum(out[[1]]^2) ## using variance not SD
```

```
Prop.var2 = out[[1]][2]^2/sum(out[[1]]^2) ## using variance not SD
```

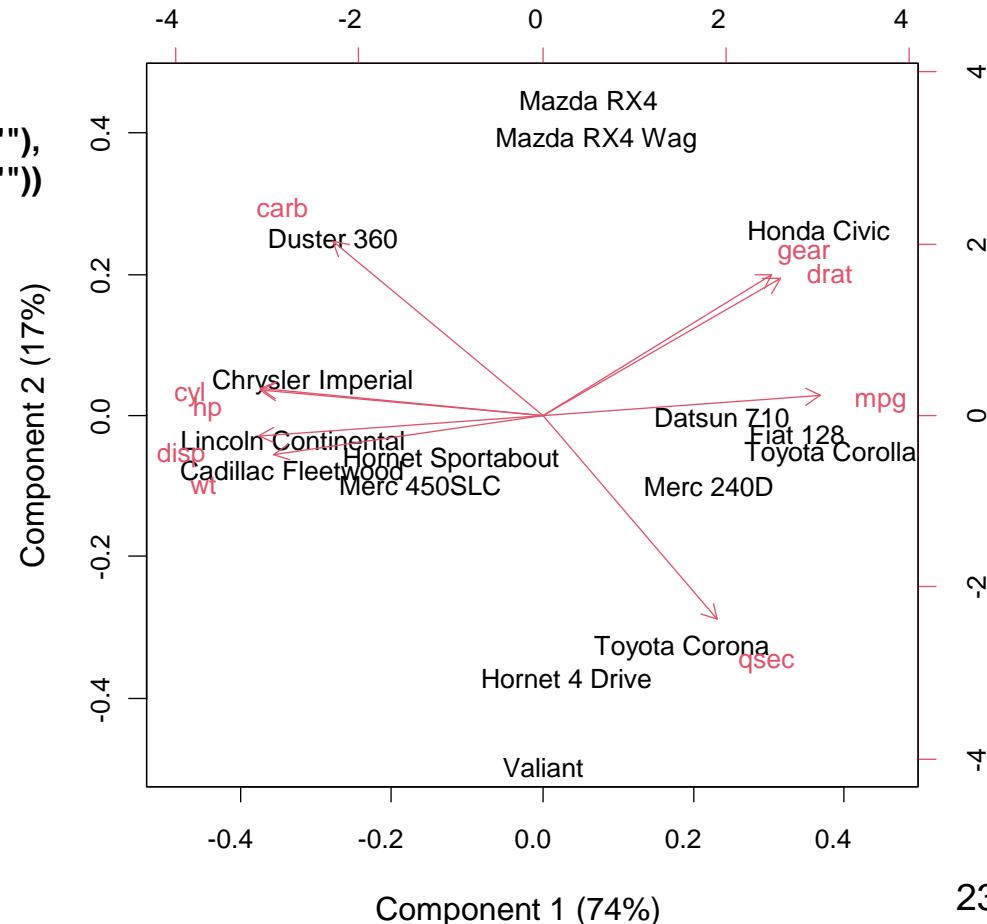
```
Prop.var1 = round(Prop.var1, 2)*100
```

```
Prop.var2 = round(Prop.var2, 2)*100
```

```
biplot(pca, cex=0.7,
```

```
  xlab = paste("Component 1 (", Prop.var1, "%)", sep=""),
```

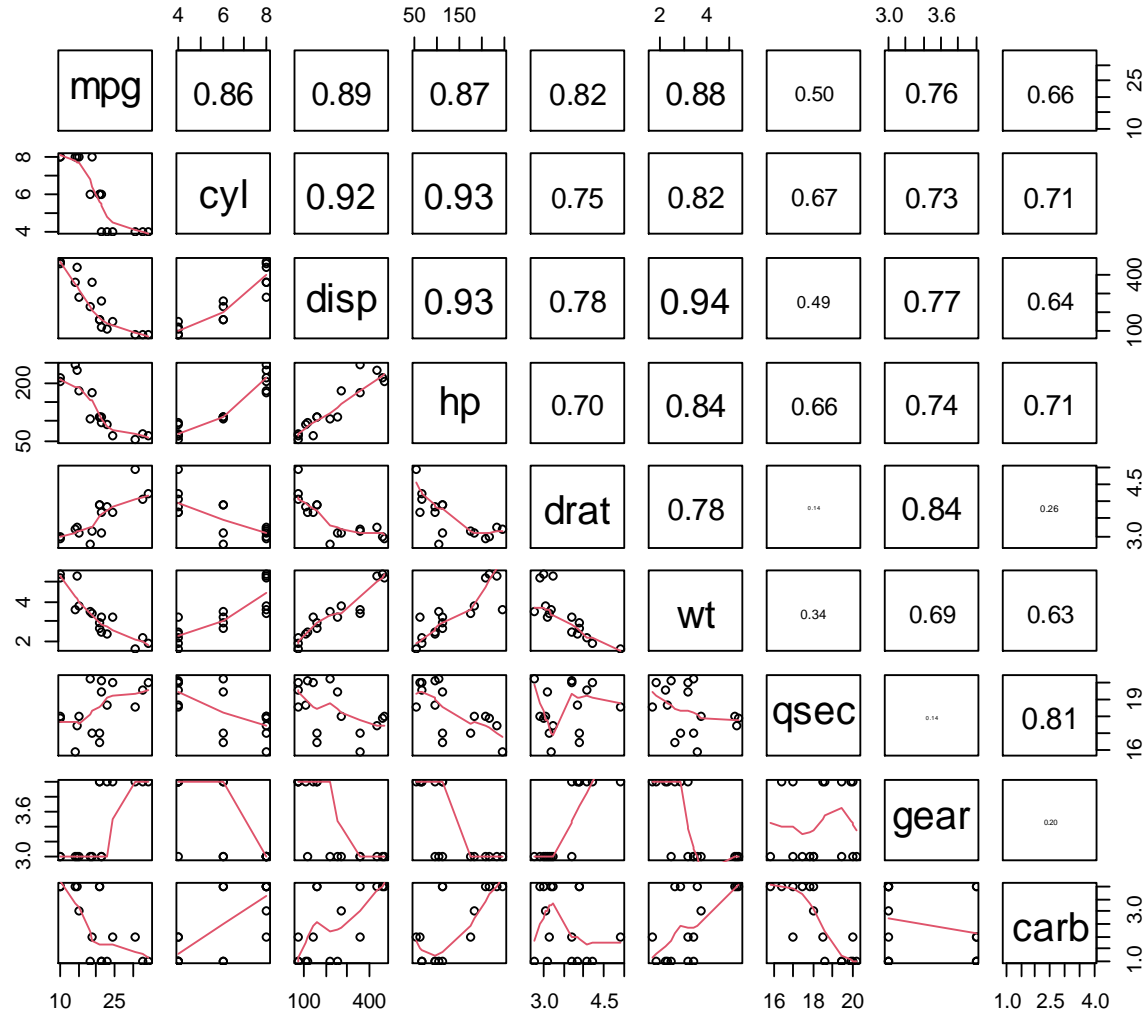
```
  ylab = paste("Component 2 (", Prop.var2, "%)", sep=""))
```



# Association of variables

```
## put (absolute) correlations on the upper panels,
## with size proportional to the correlations.
panel.cor <- function(x, y, digits = 2, prefix = "", cex.cor, ...)
{
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(0, 1, 0, 1)) # ranges for x-axis and y-axis in plots
  r <- abs(cor(x, y))
  txt <- format(c(r, 0.123456789), digits = digits)[1]
  txt <- paste(prefix, txt, sep = " ")
  if(missing(cex.cor)) cex.cor <- 0.8/strwidth(txt)
  text(0.5, 0.5, txt, cex = cex.cor * r)
}
```

```
pairs(cars, lower.panel = panel.smooth,
      upper.panel = panel.cor)
```





# R code: PCA

`pca2$loadings` # in R `princomp()`, the loadings are coefficients (eigenvectors)

	Comp.1	Comp.2	Comp.3	Comp.4	Comp.5	Comp.6	Comp.7	Comp.8	Comp.9	Comp.10	Comp.11
mpg	0.363		-0.226		-0.103	0.109	0.368	0.754	0.236	0.139	0.125
cyl	-0.374		-0.175			-0.169		0.231		-0.846	0.141
disp	-0.368			0.257	-0.394	0.336	0.214		0.198		-0.661
hp	-0.33	-0.249	0.14		-0.54			0.222	-0.576	0.248	0.256
drat	0.294	-0.275	0.161	0.855		-0.244				-0.101	
wt	-0.346	0.143	0.342	0.246		0.465			0.359		0.567
qsec	0.2	0.463	0.403		0.165	0.33		0.232	-0.528	-0.271	-0.181
vs	0.307	0.232	0.429	-0.215	-0.6	-0.194	-0.266		0.359	-0.159	
am	0.235	-0.429	-0.206			0.571	-0.587			-0.178	
gear	0.207	-0.462	0.29	-0.265		0.244	0.605	-0.336		-0.214	
carb	-0.214	-0.414	0.529	-0.127	0.361	-0.184	-0.175	0.396	0.171		-0.32

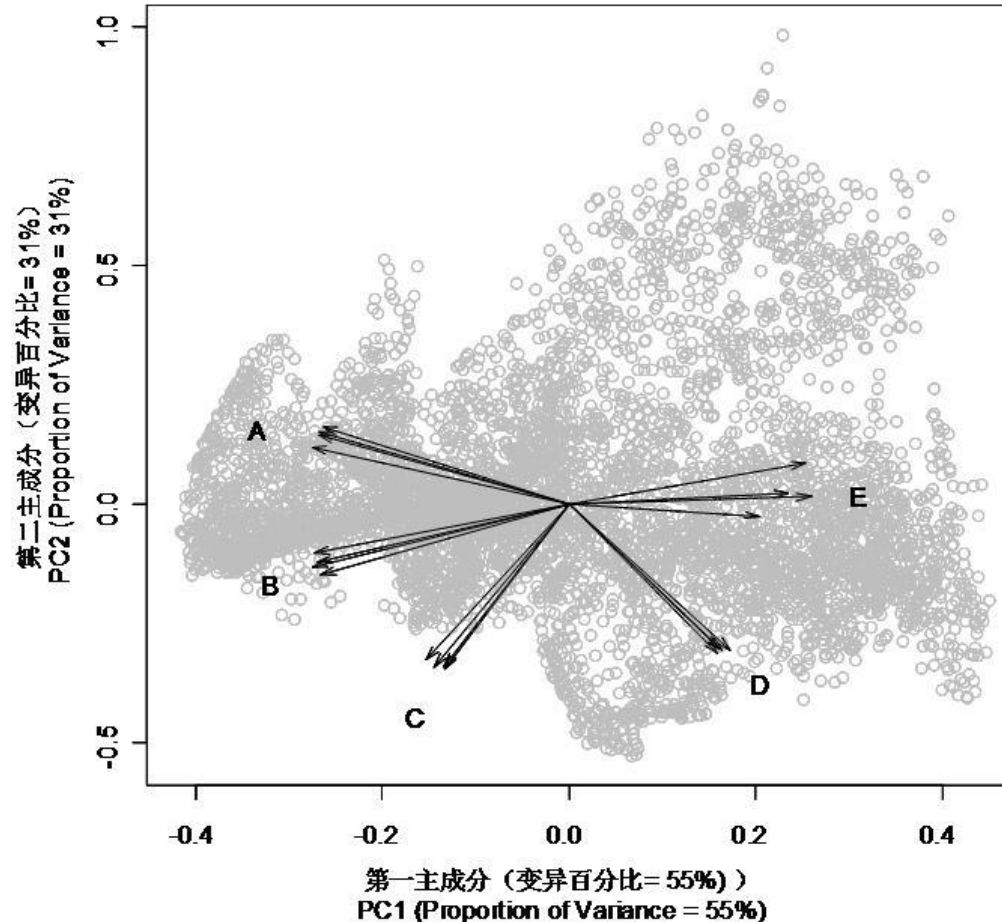
`eigen(var(scale(mtcars))) [[2]]` # coefficients, eigenvectors

# R code: PCA

**pca2\$scores**

	<b>Comp.1</b>	<b>Comp.2</b>	<b>Comp.3</b>	<b>Comp.4</b>	<b>Comp.5</b>
<b>Mazda RX4</b>	0.657213	-1.73545	-0.6012	0.115522	0.960653
<b>Mazda RX4 Wag</b>	0.629396	-1.55003	-0.38232	0.202307	1.032949
<b>Datsun 710</b>	2.779397	0.146457	-0.24124	-0.24914	-0.40514
<b>Hornet 4 Drive</b>	0.311771	2.363019	-0.13576	-0.51186	-0.558
<b>Hornet Sportabout</b>	-1.97449	0.754402	-1.1344	0.075653	-0.21084
<b>Valiant</b>	0.056138	2.786	0.163826	-0.99077	-0.21505
<b>Duster 360</b>	-3.00267	-0.33489	-0.36276	-0.05235	-0.34935
<b>Merc 240D</b>	2.055329	1.465181	0.943895	-0.1444	0.321718
<b>Merc 230</b>	2.287408	1.983527	1.797241	0.291807	0.339022
<b>Merc 280</b>	0.526381	0.162013	1.49277	0.067324	0.070738
<b>Merc 280C</b>	0.509205	0.323895	1.683585	0.095867	0.151185
<b>Merc 450SE</b>	-2.24781	0.683474	-0.37538	-0.13187	0.384669
<b>Merc 450SL</b>	-2.04782	0.683221	-0.48446	-0.21437	0.361302
<b>Merc 450SLC</b>	-2.14854	0.80174	-0.29511	-0.17814	0.439055
<b>Cadillac Fleetwood</b>	-3.89979	0.827948	0.647291	0.295154	0.049017
<b>Lincoln Continental</b>	-3.95412	0.733382	0.72061	0.411823	-0.00396
<b>Chrysler Imperial</b>	-3.59297	0.421135	0.548891	0.676317	-0.21136
<b>Fiat 128</b>	3.856284	0.296752	-0.42283	0.056074	-0.2235
<b>Honda Civic</b>	4.254033	-0.68841	-0.20684	1.186208	-0.09924
<b>Toyota Corolla</b>	4.234221	0.279288	-0.46626	0.186246	-0.22571

## Example: study on climate change consequences to the crested ibis



The loadings of 20 variables (five climate variables, i.e. annual total precipitation (A), annual minimum temperature (B), annual maximum temperature (C), seasonal variance of temperature (D), and seasonal variance of precipitation (E), at four time periods, i.e. present, 2020, 2050, and 2080 at the first and second principal components space. The grey circles are the scores of 5751 sites in Yang county at the first and second principal components space. (Zhai & Li 2012)

# 3D plot

```
library(pca3d)
```

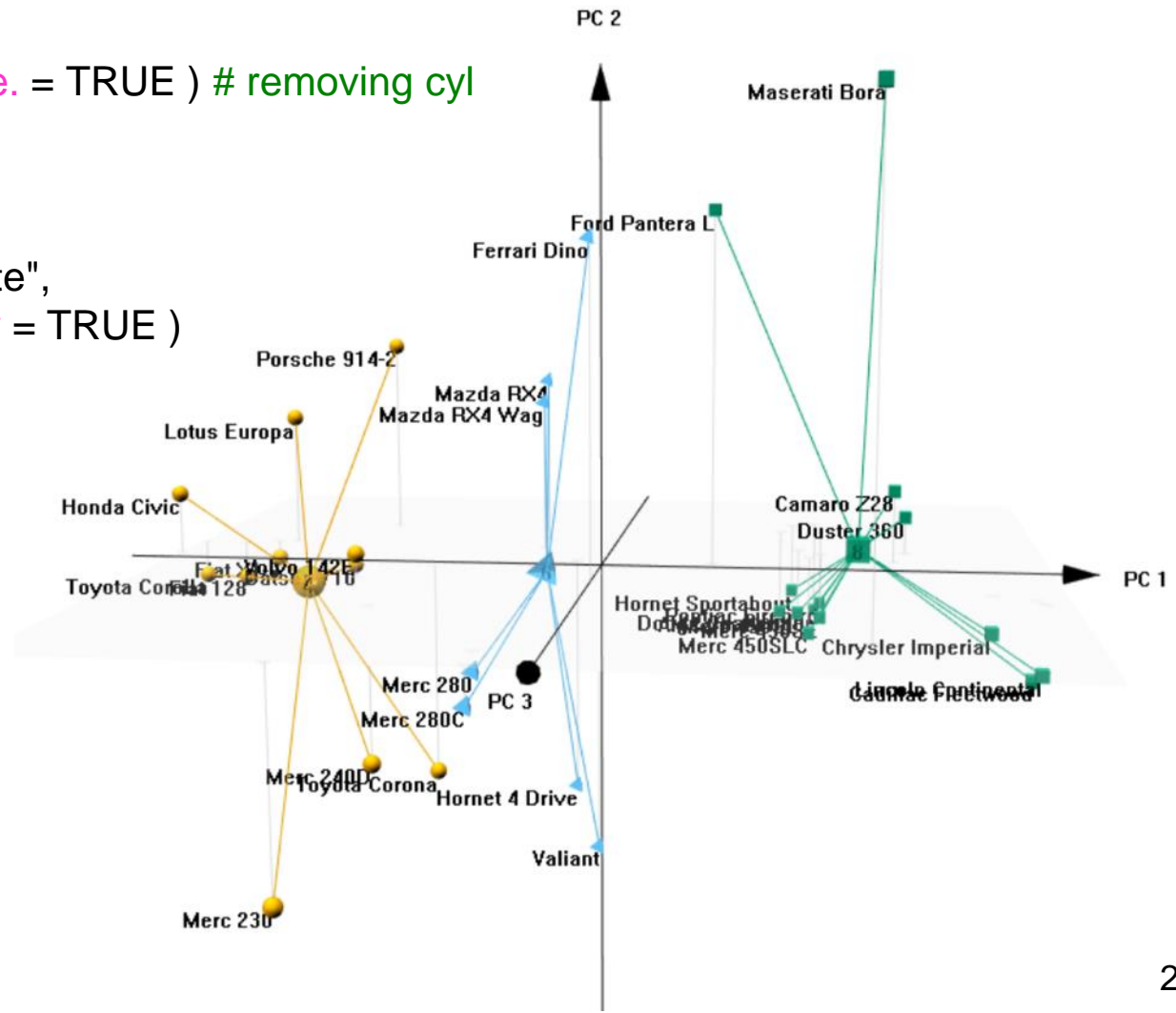
```
data(mtcars)
```

```
pca <- prcomp( mtcars[,-2], scale. = TRUE ) # removing cyl
```

```
# using cyl as the three groups
```

```
pca3d( pca, group = mtcars[,2] )
```

```
pca3d( pca, group = mtcars[,2],  
       fancy = TRUE, bg = "white",  
       axes.color = "black", new = TRUE )
```



# PCA: potential problems

- Lack of independence between variables
  - **No problem**
- Lack of normality
  - Normality desirable but not essential
- Not monotonous between any two variables
  - **Problem** (use correspondence analysis)
- Many zeroes or identical values in data matrix
  - **Problem** (use correspondence analysis)

## Note

- The principal components are dependent on the units used to measure the original variables as well as on the range of values they assume.
- We **usually** standardize the data prior to using PCA.

# Hourly records of sperm whale behaviour

- Variables:
  - Mean cluster size
  - Max. cluster size
  - Mean speed
  - Heading consistency
  - Fluke-up rate
  - Breach rate
  - Lobtail rate
  - Spyhop rate
  - Sidefluke rate
  - Coda rate
  - Creak rate
  - High click rate
- Data collected:
  - Off Galapagos Islands
  - 1985 and 1987
- Units:
  - hours spent following sperm whales
  - 440 hours

# Principal Components

	Principal Components:			
	1	2	3	4
% of variance accounted for	31.09	13.41	12.08	10.52
<i>Loadings:</i>				
Mean cluster size	0.82	0.35	0.01	-0.14
Max. cluster size	0.83	0.24	0.17	-0.12
Mean speed	-0.38	0.30	0.44	-0.09
Heading consistency	-0.48	0.39	0.19	-0.04
Fluke-up rate	-0.65	-0.19	0.30	0.25
Breach rate	0.24	0.24	-0.13	0.74
Lobtail rate	0.29	0.30	-0.09	0.71
Spyhop rate	0.46	-0.60	-0.23	0.01
Sidefluke rate	0.49	-0.57	-0.20	0.07
Coda rate	0.68	-0.08	0.53	0.03
Creak rate	0.57	-0.11	0.70	0.02
High click rate	-0.41	-0.55	0.47	0.31



	Principal Components:			
	1	2	3	4
% of variance accounted for	31.09	13.41	12.08	10.52
<i>Loadings:</i>				
Mean cluster size	0.82	0.35	0.01	-0.14
Max. cluster size	0.83	0.24	0.17	-0.12
Mean speed	-0.38	0.30	0.44	-0.09
Heading consistency	-0.48	0.39	0.19	-0.04
Fluke-up rate	-0.65	-0.19	0.30	0.25
Breach rate	0.24	0.24	-0.13	0.74
Lobtail rate	0.29	0.30	-0.09	0.71
Spyhop rate	0.46	-0.60	-0.23	0.01
Sidefluke rate	0.49	-0.57	-0.20	0.07
Coda rate	0.68	-0.08	0.53	0.03
Creak rate	0.57	-0.11	0.70	0.02
High click rate	-0.41	-0.55	0.47	0.31
<b>Principal Components meanings</b>	“Socializing/ foraging”	“Directed movement”	“Vocal”	“Aerial”

# Factor Analysis (FA)

# Factor Analysis

- Data reduction tool
- Removes redundancy or duplication from a set of correlated variables
- Represents correlated variables with a smaller set of “derived” variables.
- Factors are formed that are relatively independent of one another.
- Two types of “variables”:
  - latent variables: factors
  - observed variables

# Some applications of factor analysis

## 1. Identification of underlying factors:

- clusters variables into homogeneous sets
- creates new variables (i.e. factors)
- allows us to gain insight to categories

## 2. Screening of variables:

- identifies groupings to allow us to select one variable to represent many
- useful in regression (recall collinearity)

## 3. Summary:

- Allows us to describe many variables using a few factors

## 4. Clustering of objects:

- Helps us to put objects (people) into categories depending on their factor scores

## R code: Factor Analysis

```
# Exploratory Factor Analysis (Maximum Likelihood)
# extracting 3 factors, with varimax rotation
fit <- factanal(scale(mtcars), 3, rotation = "varimax")
print(fit, digits = 2, cutoff = .01, sort = TRUE)
```

Call:

```
factanal(x = scale(mtcars), factors = 3, rotation = "varimax")
```

Uniquenesses:

```
mpg cyl  disp  hp  drat  wt   qsec  vs   am  gear  carb
0.13 0.06 0.09  0.13 0.29 0.06 0.05  0.22 0.21 0.12 0.16
```

Loadings:

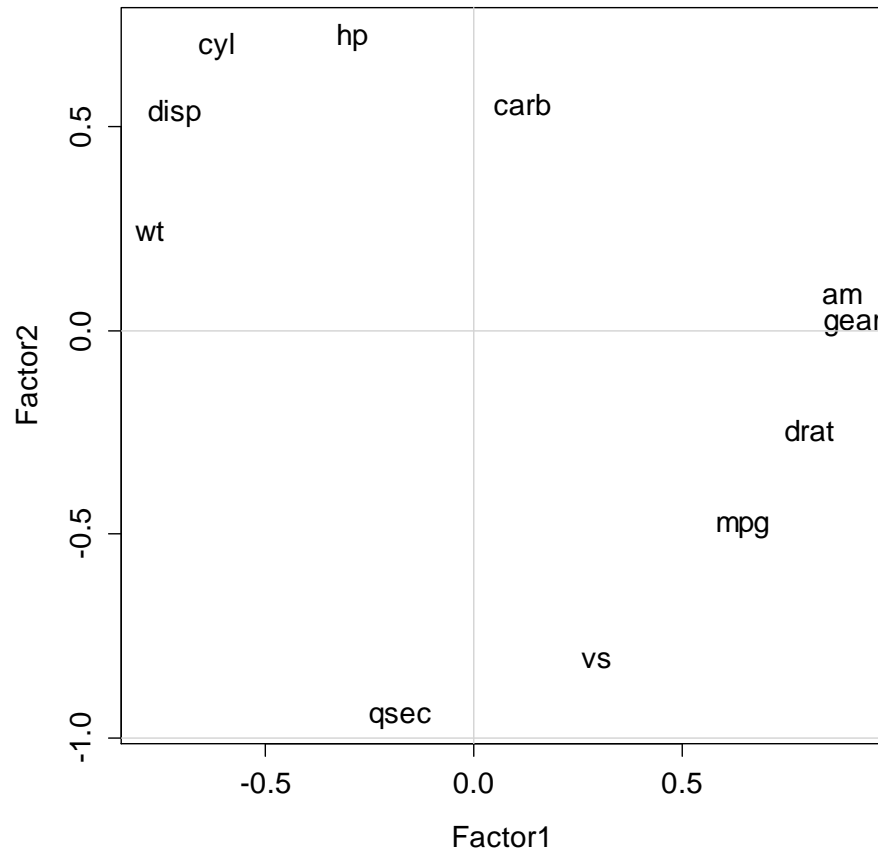
	Factor1	Factor2	Factor3
mpg	0.64	-0.48	-0.47
disp	-0.72	0.54	0.32
drat	0.8	-0.24	-0.07
wt	-0.78	0.25	0.52
am	0.88	0.09	-0.09
gear	0.91	0.02	0.22
cyl	-0.62	0.7	0.26
hp	-0.29	0.72	0.51
qsec	-0.18	-0.95	-0.15
vs	0.3	-0.8	-0.2
carb	0.11	0.56	0.72

	Factor1	Factor2	Factor3
SS loadings	4.38	3.52	1.58
Proportion Var	0.40	0.32	0.14
Cumulative Var	0.40	0.72	0.86

Test of the hypothesis that 3 factors are sufficient. The chi square statistic is 30.53 on 25 degrees of freedom. The p-value is 0.205

# R: Result

```
# plot factor 1 by factor 2  
load <- fit$loadings[, 1:2]  
plot(load, type = "n") # set up plot  
text(load, labels = names(mtcars), cex = 1) # add variable names  
abline(h = -1:1, v = -1:1, col = "lightgray", lty=1)
```



# R code: Factor Analysis

```
library(MASS)
COV = cov(mtcars)
FA <- factanal(covmat = COV, factor = 3, rotation = "varimax")
load <- loadings(FA); rot <- FA$rot
```

# Pairs of factor loadings to plot

```
ind <- combn(1:3,2)
```

```
par(mfrow = c(2,2))
```

```
nms <- row.names(load)
```

# Loop over pairs of factors and draw each plot

```
for (i in 1:3){
```

```
  eqscplot(load[,ind[1,i]], load[,ind[2,i]], xlim = c(-1,1),
```

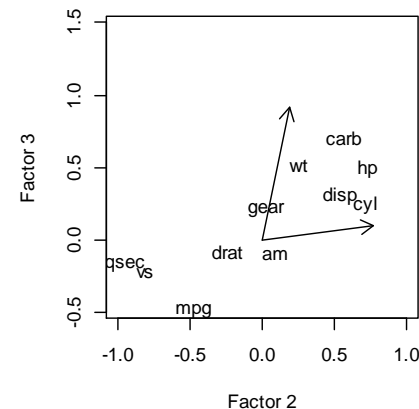
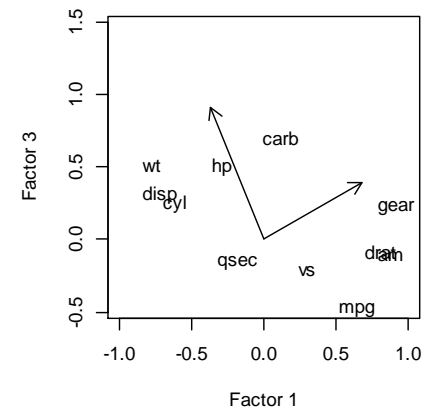
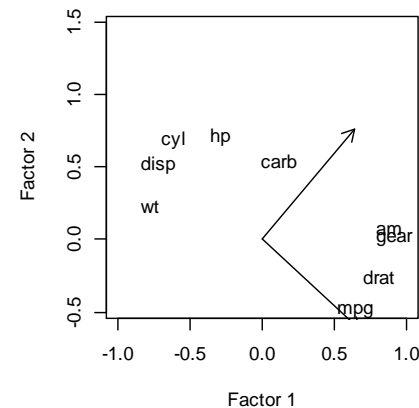
```
    ylim = c(-0.5,1.5), type = "n",
```

```
    xlab = paste("Factor", as.character(ind[1,i])),
```

```
    ylab = paste("Factor", as.character(ind[2,i])))
```

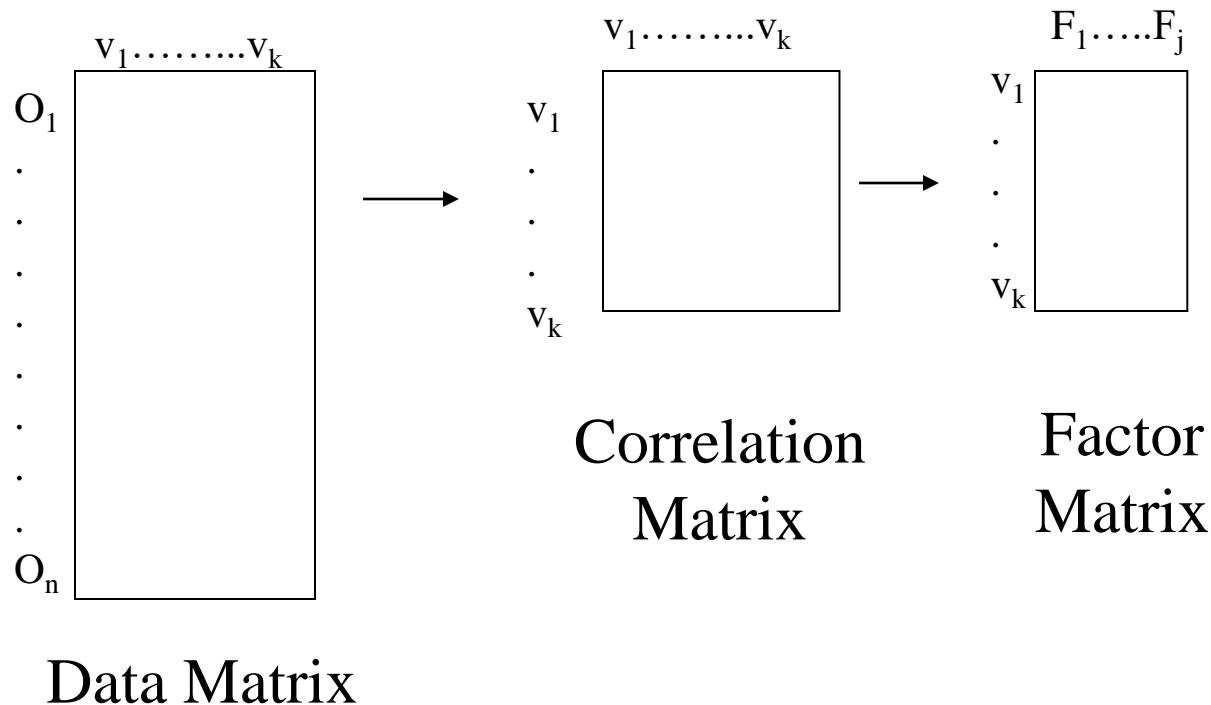
```
  text(load[,ind[1,i]],load[,ind[2,i]], labels = nms)
```

```
  arrows(c(0,0), c(0,0), rot[ind[,i], ind[,i]][,1],
    rot[ind[,i], ind[,i]][,2], length = 0.1)}
```



# Data Matrix

- Factor analysis is **totally dependent** on correlations between variables.
- Factor analysis summarizes correlation structure





# Choosing number of factors

- Intuitively: The number of uncorrelated constructs that are jointly measured by the X's.
- Only useful if number of factors is less than number of X's (recall “data reduction”).

Use “principal components” to help decide

- number of factors is equivalent to number of variables
- each factor is a weighted combination of the input variables:

$$F_1 = a_{11}X_1 + a_{12}X_2 + \dots$$

# Eigenvalues

- To select how many factors to use, consider *eigenvalues* from a principal components analysis
- Two interpretations:
  - eigenvalue  $\cong$  equivalent number of variables which the factor represents
  - eigenvalue  $\cong$  amount of variance in the data described by the factor.
- Rules to go by:
  - number of eigenvalues  $> 1$
  - scree plot
  - % variance explained
  - comprehensibility
- Note: sum of eigenvalues is equal to the number of items

# Steps in Factor Analysis

- Factor analysis usually proceeds in four steps:
  - 1: the correlation matrix for all variables is computed
  - 2: Factor extraction
  - 3: Factor rotation
  - 4: Make final decisions about the number of underlying factors

# The Correlation Matrix

## The first step: the correlation matrix

- Generate a correlation matrix for all variables
- Identify variables not related to other variables
- If the correlation between variables are small, it is unlikely that they share common factors (variables must be related to each other for the factor model to be appropriate).
- Correlation coefficients greater than 0.3 in absolute value are indicative of acceptable correlations.
- Examine visually the appropriateness of the factor model.

# Factor Extraction

## The second step: factor extraction

- The principal components analysis is the most commonly used extraction method. Other factor extraction methods include:
  - ✓ Maximum likelihood method
  - ✓ Principal axis factoring
  - ✓ Alpha method
  - ✓ Unweighted least squares method
  - ✓ Generalized least square method
  - ✓ Image factoring.

# Factor Extraction

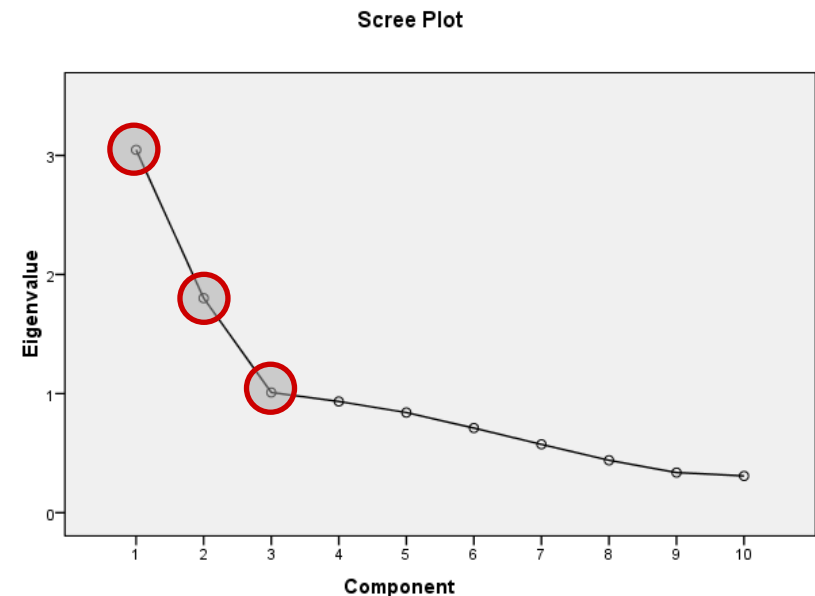
- In principal components analysis, **linear combinations** of the observed variables are formed.
- The 1<sup>st</sup> principal component is the combination that accounts for the **largest amount of variance** in the sample (1<sup>st</sup> extracted factor).
- The 2<sup>nd</sup> principle component accounts for the next largest amount of variance and **is uncorrelated with the first** (2<sup>nd</sup> extracted factor).
- Successive components explain progressively smaller portions of the total sample variance, and all are uncorrelated with each other.

# Factor Extraction

- To decide on how many factors we need to represent the data, we use 2 statistical criteria:
  - **Eigen Values**, and
  - The **Scree Plot**.
- The determination of the number of factors is usually done by considering only factors with Eigen values greater than 1.
- Factors with a variance less than 1 are no better than a single variable, since each variable is expected to have a variance of 1.

# Factor Extraction

- The examination of the **Scree plot** provides a visual of the total variance associated with each factor.
- The steep slope shows the large factors.
- The gradual trailing off (scree) shows the rest of the factors usually lower than an Eigen value of 1.
- In choosing the number of factors, in addition to the statistical criteria, one should make initial decisions based on conceptual and theoretical grounds.
- At this stage, the decision about the number of factors is not final.

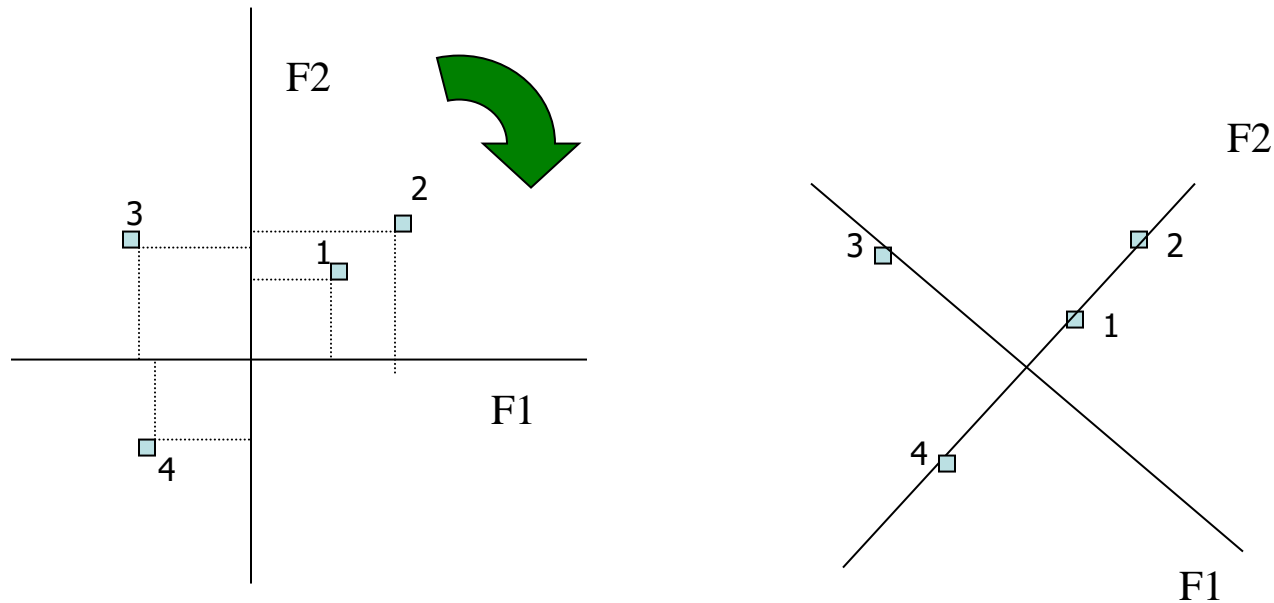




# Factor Rotation

- **3<sup>rd</sup> Step: Factor rotation.**
- In this step, factors are rotated.
- Un-rotated factors are typically not very interpretable (most factors are correlated with many variables).
- Factors are rotated to make them more meaningful and easier to interpret (each variable is associated with a minimal number of factors).
- Different rotation methods may result in the identification of somewhat different factors.

# Factor Rotation



	Factor 1	Factor 2
x1	0.5	0.5
x2	0.8	0.8
x3	-0.7	0.7
x4	-0.5	-0.5

	Factor 1	Factor 2
x1	0	0.6
x2	0	0.9
x3	-0.9	0
x4	0	-0.9

# Factor Rotation

- Quartimax
  - Simplify rows – variable loads high on one factor, low on others
- Varimax
  - Simplify columns – clearer separation of factors – each factor has variables that either load high or load very low
- Equimax
  - Compromise between the two – rarely used
- The most popular rotational method is **Varimax rotations**.
- Varimax use orthogonal rotations yielding uncorrelated factors/components.
- Varimax attempts to minimize the number of variables that have high loadings on a factor. This enhances the interpretability of the factors.

# Making Final Decisions

- 4<sup>th</sup> Step: Making final decisions
  - The final decision about the number of factors to choose is the number of factors for the rotated solution that is **most interpretable**.
  - To identify factors, group variables that have large loadings for the same factor.
  - Plots of loadings provide a visual for variable clusters.
  - Interpret factors according to the meaning of the variables
- This decision should be guided by:
  - A priori conceptual beliefs about the number of factors from past research or theory
  - Eigen values computed in step 2.
  - The relative interpretability of rotated solutions computed in step 3.

# Assumptions

- Assumption underlying factor analysis include:
  - The measured variables are linearly related to the factors + errors.
  - The data should have a bivariate normal distribution for each pair of variables.
  - Observations are independent.
  - The factor analysis model assumes that variables are determined by common factors and unique factors. All unique factors are assumed to be **uncorrelated** with each other.

# Factor Analysis (FA) vs. PCA

- PCA analyzes variance; FA analyzes covariance.
- PCA is to extract as much variance with the least amount of factors.
- FA is to explain as much of the correlations with a minimum number of factors.
- PCA gives a unique solution. If all components are retained, all variance is explained.
- FA can give multiple solutions depending on the method and the estimates of communality.

# Factor Analysis vs. Cluster Analysis

- Both are data reduction techniques.
- Factor Analysis is to reduce original set of variables to smaller set of factors.
- Cluster Analysis is to form groups from the observations or records, thus reducing original number of elements to fewer groups.
- Factor Analysis can be seen as a clustering technique that is focused on the columns of data frame, rather than the rows.

# Correspondence Analysis (CA)



# Correspondence Analysis

- Also called Reciprocal Averaging (Hill 1973)
- Weighted averaging of site scores to yield species scores and vice versa
- Simultaneous ordination of both rows and columns of a matrix
- Used to examine relationship of species assemblages to site characteristics
- Sites typically span an environmental gradient

## Using Chi-Square approach to measure correspondence between rows and columns

Observed					
	Sp1	Sp2	Sp3	Sp4	Plot total
Plot1	6	12	18	24	60
Plot2	5	10	15	20	50
Plot3	4	8	12	16	40
Plot4	3	6	9	12	30
Plot5	2	4	6	8	20
Species total	20	40	60	80	200

Expected				
	Sp1	Sp2	Sp3	Sp4
Plot1	6	12	18	24
Plot2	5	10	15	20
Plot3	4	8	12	16
Plot4	3	6	9	12
Plot5	2	4	6	8

Chi Square				
	Sp1	Sp2	Sp3	Sp4
Plot1	0	0	0	0
Plot2	0	0	0	0
Plot3	0	0	0	0
Plot4	0	0	0	0
Plot5	0	0	0	0

$$6/60 = 20/200$$

$$= 20 \times 60 / 200$$

$$= (\text{Exp} - \text{Obs})^2 / \text{Exp}$$

$$\begin{aligned} \text{Inertia} &= \text{Chi Square} / \text{Grand Sum} \\ &= 0 / 200 = 0 \end{aligned}$$

In this dataset, rows and columns are not independent. Thus, the contents of each cell are predictable based on row total, column totals, and the grand total.

## As rows and column deviate (more independent), Chi Square values (and inertia) grows

Observed					
	Sp1	Sp2	Sp3	Sp4	Plot total
Plot1	6	8	12	15	41
Plot2	5	2	20	7	34
Plot3	4	8	11	3	26
Plot4	3	6	8	5	22
Plot5	2	5	1	8	16
Species total	20	29	52	38	139
Expected					
	Sp1	Sp2	Sp3	Sp4	
Plot1	5.899	8.554	15.338	11.209	
Plot2	4.892	7.094	12.719	9.295	
Plot3	3.741	5.424	9.727	7.108	
Plot4	3.165	4.590	8.230	6.014	
Plot5	2.302	3.338	5.986	4.374	
Chi Square					
	Sp1	Sp2	Sp3	Sp4	
Plot1	0.002	0.036	0.726	1.282	
Plot2	0.002	3.657	4.167	0.567	
Plot3	0.018	1.223	0.167	2.374	
Plot4	0.009	0.433	0.006	0.171	
Plot5	0.040	0.827	4.153	3.006	
Sum					22.867

$$6/41 \sim 20/139$$

$$= 20 \times 41 / 139$$

$$= (\text{Exp} - \text{Obs})^2 / \text{Exp}$$

$$\begin{aligned} \text{Inertia} &= \text{Chi Square} / \text{Grand Sum} \\ &= 22.867 / 139 = 0.165 \end{aligned}$$

Total variance the CA will attempt to explain.

The chi square matrix describes all the variability in the dataset not explainable by row or column profiles (totals).

## Correspondence Analysis

Chi-Sqr	sp1	sp2	sp3	sp4	Row totals
sam1	0.07877	0.124363	0.0806	0.232143	<b>0.515876</b>
sam2	0.501505	0.341336	0.256398	1.193828	<b>2.293067</b>
sam3	4.892877	0.300778	1.172794	1.028178	<b>7.394627</b>
sam4	3.462503	0.590862	0.791607	0.224873	<b>5.069845</b>
sam5	0.557292	0.005016	0.132378	0.473542	<b>1.168228</b>
Col totals	<b>9.492948</b>	<b>1.362354</b>	<b>2.433777</b>	<b>3.152565</b>	

Look at where the variability is in the Chi Sqr matrix...

Chi Sqr    Inertia  
16.44164    0.08519

Partitioning of mean squared contingency coefficient:

	Inertia	Proportion
Total	0.08519	1
Unconstrained	0.08519	1

...

Importance of components:

	CA1	CA2	CA3
Eigenvalue	0.0748	0.0100	0.000414
Proportion Explained	0.8776	0.1176	0.004850
Cumulative Proportion	0.8776	0.9951	1.000000

...

Species scores

	CA1	CA2	CA3
sp1	<b>-0.39331</b>	-0.030492	-0.0008905
sp2	0.09946	0.141064	0.0219980
sp3	0.19632	0.007359	-0.0256591
sp4	0.29378	-0.197766	0.0262108

CA does an eigenvalue decomposition to summarize this variability in fewer axes (components).

Species and sites that contribute most to the inertia have the largest magnitude CA1 scores.

Site scores (weighted averages of species scores)

	CA1	CA2	CA3
sam1	-0.2405	-1.9357	3.4903
sam2	0.9471	-2.4310	-1.6574
sam3	<b>-1.3920</b>	-0.1065	-0.2535
sam4	0.8520	0.5769	0.1625
sam5	-0.7355	0.7884	-0.3974

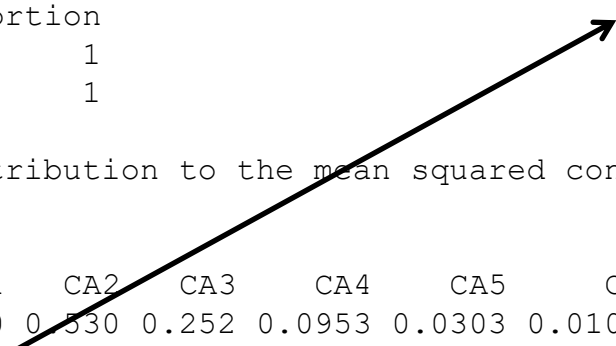
Scores are centered and scaled to be directly comparable.

# Correspondence Analysis

- Output
  - Row and column sums, total Chi Square
  - Species and sample scores that can be plotted in the same space. Interpretation is similar to sample scores and species weighted averages in NMDS.
  - # axes = n-1 for whichever dimension of the data matrix is lower (samples or species).
  - Eigenvalues – relative importance of each axis, interpreted as the percentage of total **inertia** explained.

Partitioning of mean squared contingency coefficient:

	Inertia	Proportion
Total	1.780	1
Unconstrained	1.780	1

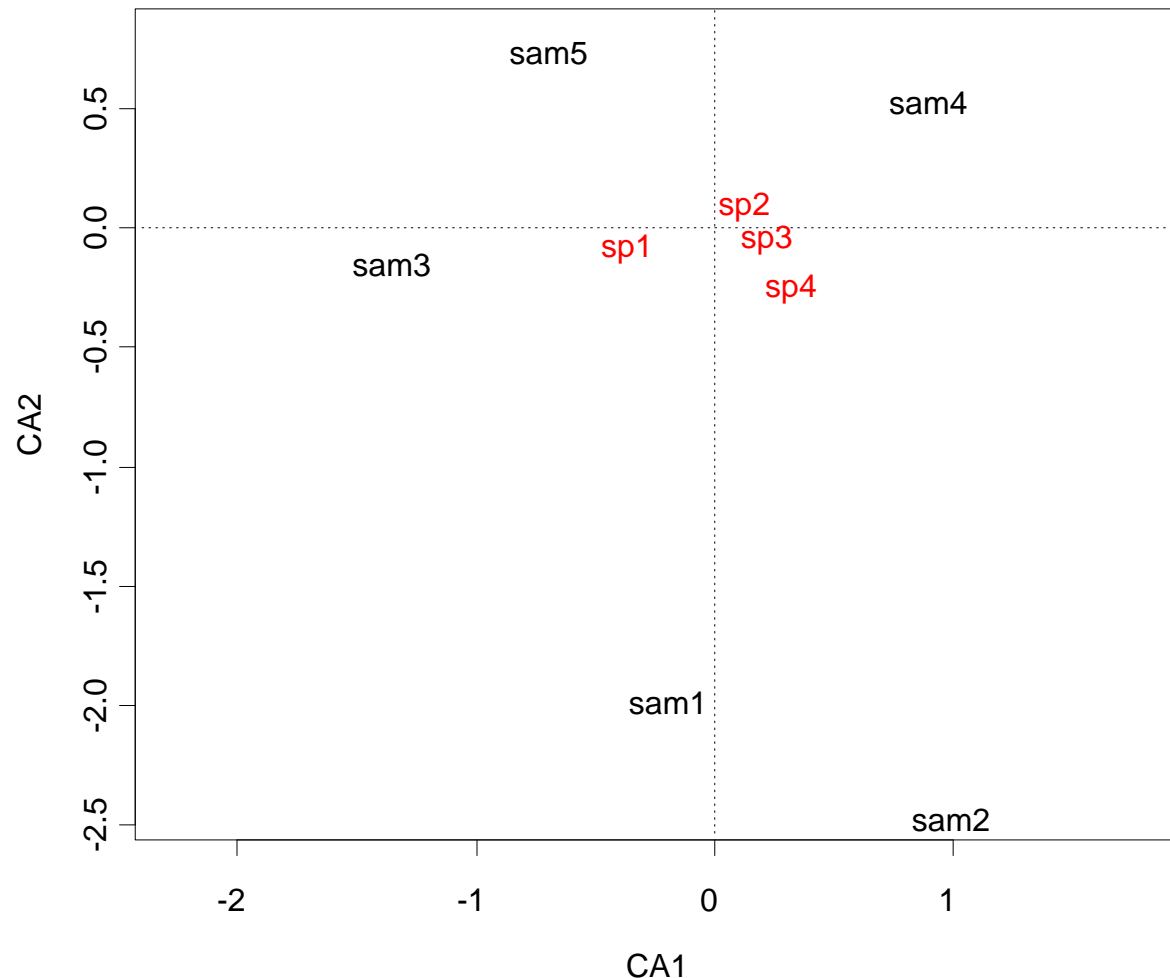
$$0.85/1.780=0.478$$


Eigenvalues, and their contribution to the mean squared contingency coefficient

Importance of components:

	CA1	CA2	CA3	CA4	CA5	CA6	CA7	CA8	CA9
Eigenvalue	0.850	0.530	0.252	0.0953	0.0303	0.01097	0.00638	0.00307	0.00179
Proportion Explained	0.478	0.298	0.142	0.0536	0.0170	0.00616	0.00358	0.00173	0.00101
Cumulative Proportion	0.478	0.775	0.917	0.9705	0.9875	0.99368	0.99727	0.99899	1.00000

## Correspondence Analysis



Distances between species are two-dimensional approximations of their Chi-square distances. Distances between samples are also two-dimensional approximations of Chi-square distances. Distances between species and sites cannot be interpreted.

# How does CA work?

- Site-species matrix
- Eigen analysis

similar to PCA but differs in some details

axes rotated through species and sample space  
with the goal of maximizing their correspondence

- Reciprocal averaging

calculate species scores as weighted averages of  
the sites in which they occur

calculate new site scores by weighted averaging  
of the species scores

# Output of CA

- Yields principal axes and scores
  - scores for rows (sites) and columns (species)
  - 1<sup>st</sup> axis has the largest eigenvalue (and accounts for largest variance); maximizes association between rows and columns
  - subsequent axes account for residual variation and have smaller eigenvalues
  - rarely use more than 2-3 axes in CA



## R code: Correspondence analysis

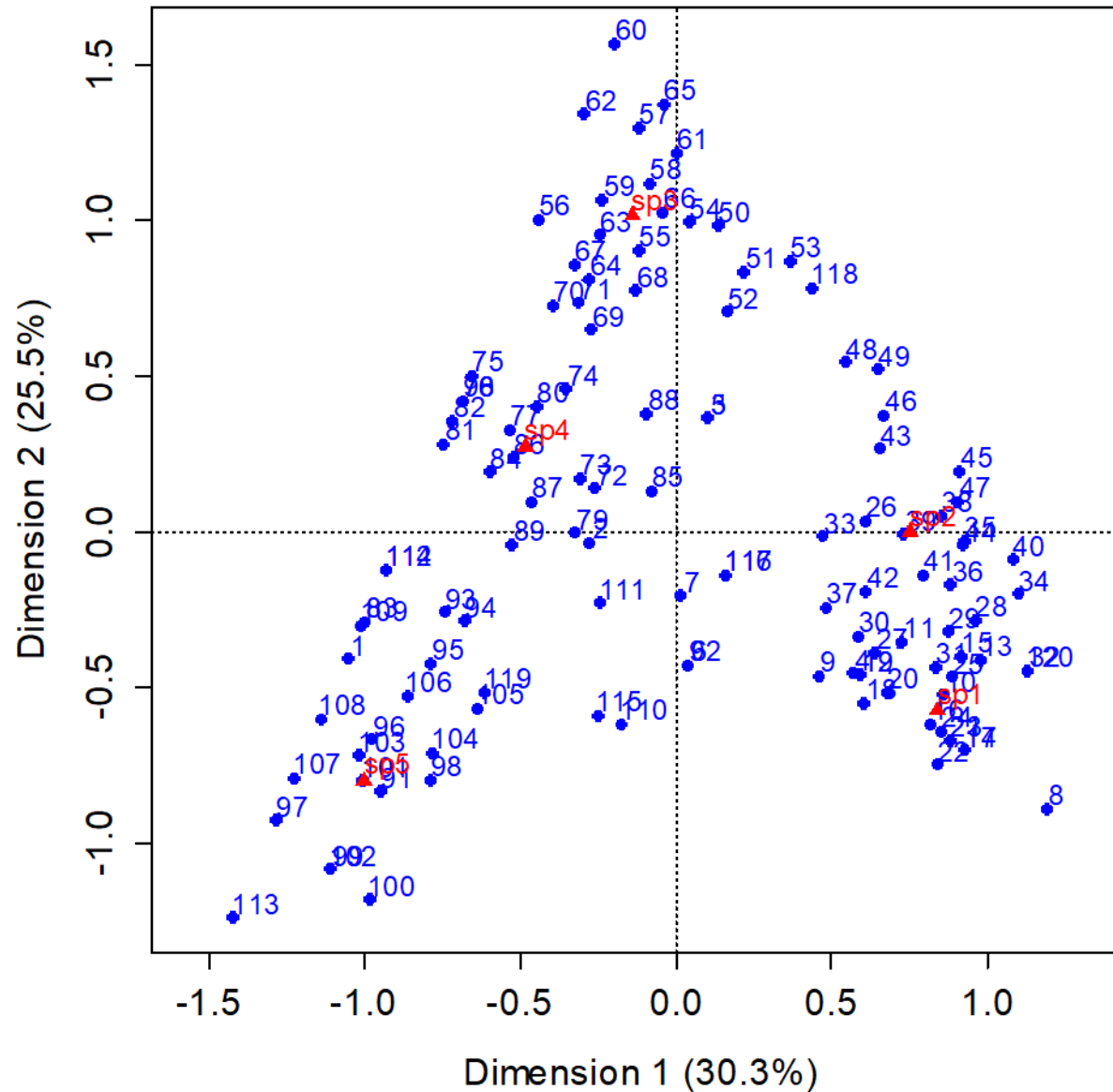
```
library(ca) # Package for Correspondence analysis
data(author); ca(author); plot(ca(author)) # One example

# Generate 5 species density data at 120 sites
sp1 = round(rnorm(120, 20, 7)); sp1 = abs(sp1)
sp2 = round(rnorm(120, 40, 7))
sp3 = round(rnorm(120, 60, 7))
sp4 = round(rnorm(120, 80, 7))
sp5 = round(rnorm(120, 100, 7))
species = data.frame(sp1, sp2, sp3, sp4, sp5); species = species * 0
sp.1 = table(sp1)
sp.2 = table(sp2)
sp.3 = table(sp3)
sp.4 = table(sp4)
sp.5 = table(sp5); sp.5 = sp.5[1:(nrow(sp.5)-2)]
species$sp1[c(as.numeric(names(sp.1)))] = sp.1
species$sp2[c(as.numeric(names(sp.2)))] = sp.2
species$sp3[c(as.numeric(names(sp.3)))] = sp.3
species$sp4[c(as.numeric(names(sp.4)))] = sp.4
species$sp5[c(as.numeric(names(sp.5)))] = sp.5

#Generate a noise
random = matrix(sample(c(0,1), 600, rep=T), nrow=120, ncol=5)
species = species + random
rownames(species) = c(1:120) #define row names
ca(species) #Correspondence analysis
plot(ca(species))
```

	sp1	sp2	sp3	sp4	sp5
1	0	0	0	1	1
2	1	0	1	1	1
3	1	0	1	1	0
4	2	0	0	1	0
5	1	0	1	1	0
6	1	1	0	1	1
7	2	0	1	1	1
8	2	0	0	0	0
9	5	0	1	1	1
10	4	1	0	1	0
11	5	0	1	1	0
12	6	1	1	1	1
13	5	1	1	0	0
14	6	0	0	1	0
15	4	0	1	0	0
16	8	1	1	0	1
17	6	0	0	1	0
18	7	0	1	1	1
19	6	1	1	1	1
20	5	1	1	0	1
21	5	0	0	1	0
22	10	1	0	1	1
23	5	0	0	1	0
24	9	1	1	0	1
25	8	1	1	1	0
26	2	2	1	1	0
27	3	4	0	1	1
28	4	2	1	0	0
29	3	3	0	1	0
30	3	2	1	0	1
31	3	5	0	0	1
32	4	4	0	0	0
33	1	5	1	1	1
34	2	7	0	0	0
35	2	4	1	0	0
36	2	5	0	1	0
37	1	4	0	1	1
38	0	7	0	1	0
39	1	12	1	1	1
40	1	9	0	0	0
41	0	8	0	0	1
42	1	6	0	1	1
43	1	6	2	1	0
44	1	9	0	1	0
45	0	7	1	0	0
46	1	3	2	0	0
47	1	5	1	0	0
48	1	3	3	0	0
49	0	2	1	0	0
50	0	3	6	1	0

## Five species at 120 sites



# Correspondence Analysis

- First axis always most informative
- Number of axes produced is set by the dimensionality of the data, not a user option
- Not distance based, data transformations typically more important
- Ordinates both samples and species directly

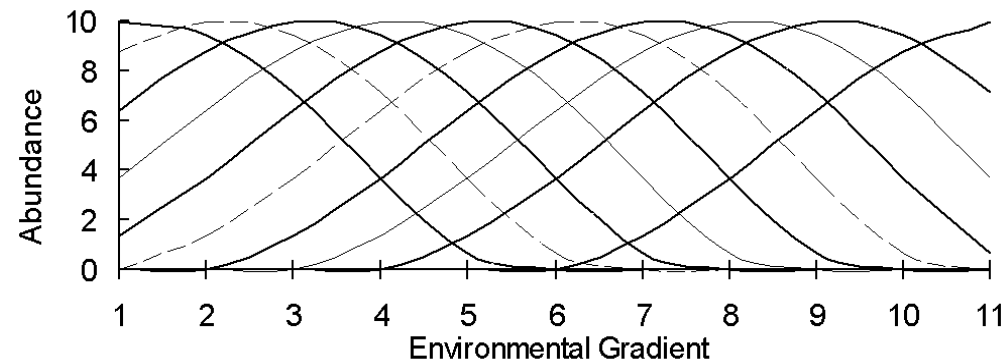
# CA vs. PCA

- Both are eigenvector methods
- PCA uses Euclidean distance, while CA uses Chi-square distance
- Underlying assumption – species abundance distributions are Gaussian (normal and unimodal)
- CA orders the points correctly on the first axis
- Curvature and “tucking in” of the ends of the gradient in PCA results in failure to order points correctly (horseshoe effect)

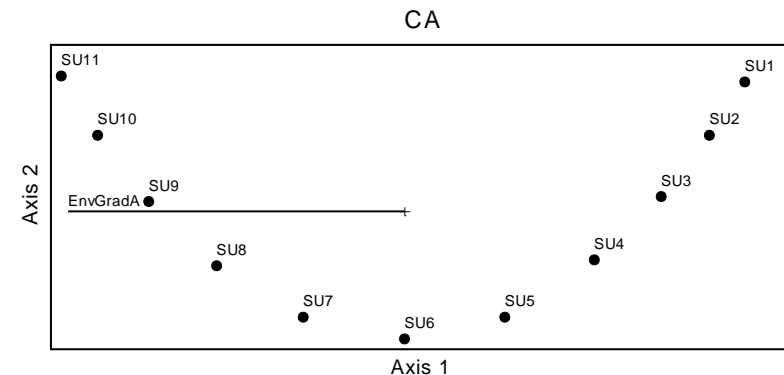
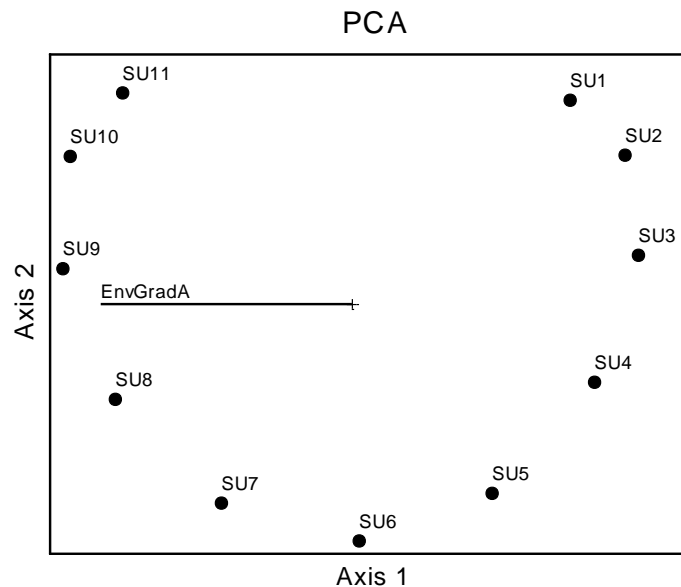
# Advantage of CA

- Weaknesses of PCA:
  - Assumes species are linearly related with each other and/or gradients
  - **Samples are ordinated in species space**
  - Results in “horseshoe effect” where ends of ordination axes are distorted
- Correspondence analysis allows for non-linear unimodal relationships
  - Both samples and species handled similarly, axes do not explicitly represent species-space

## Represent environmental gradient

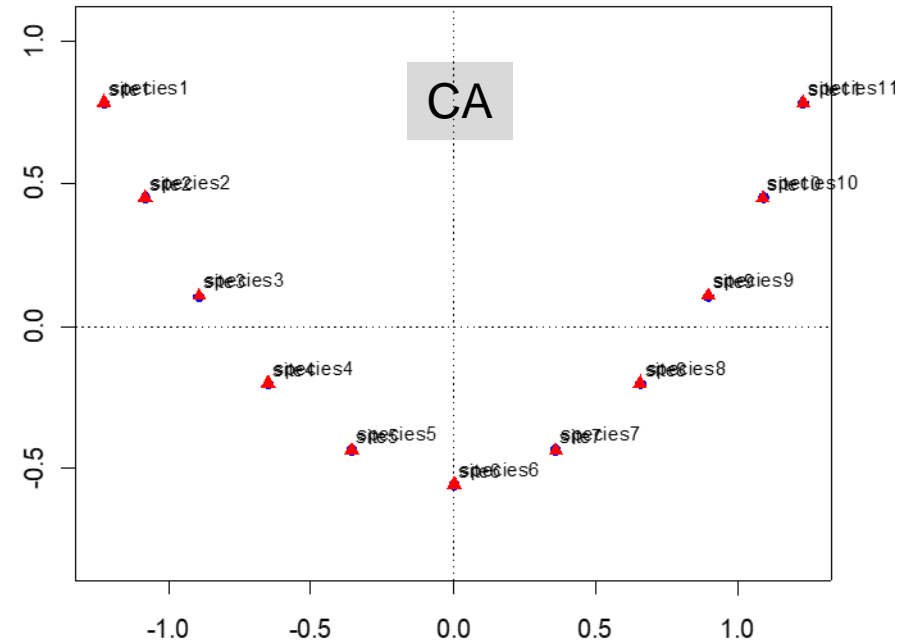
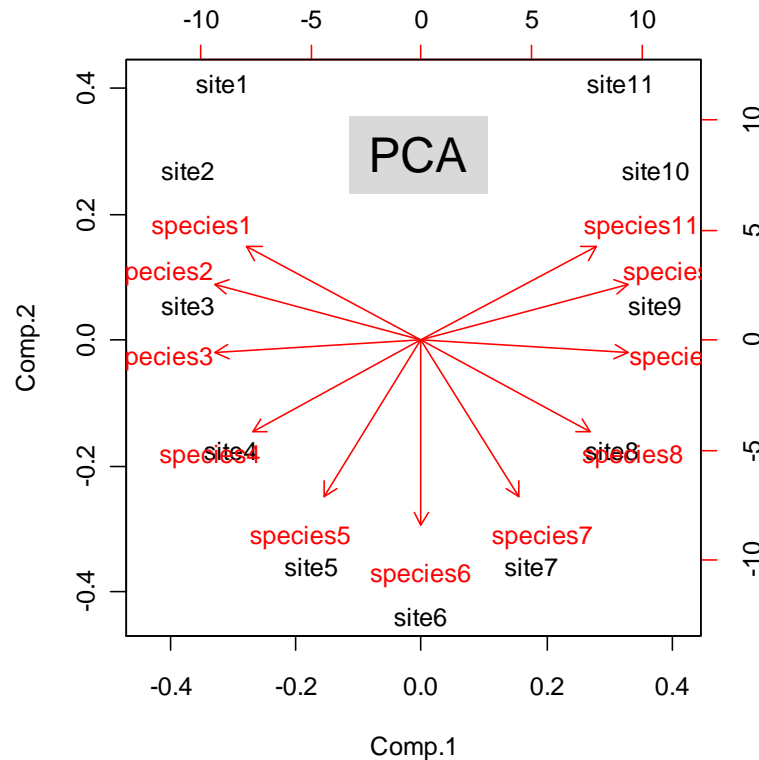


A synthetic data set of eleven species with noiseless hump-shaped responses to an environmental gradient. The gradient was sampled at eleven points (sample units), numbered 1-11.



Comparison of PCA and CA of the data set shown in Figure 19.1. PCA curves the ends of the gradient in, while CA does not. The vectors indicate the correlations of the environmental gradient with the axis scores.

# Horseshoe effect



**# horseshoe effect**

`species <- read.csv('D:/PCA species environment gradient.csv', header = T)` **# see following table for data**

`rownames(species) <- species$site`

`species <- species[,-1]` **#remove the first column (row names)**

`pca <- princomp(species)`

`biplot(pca)`

`library(ca)` **# Package for CA**

`ca(species)` **#Correspondence analysis**

`plot(ca(species))`

site	species1	species2	species3	species4	species5	species6	species7	species8	species9	species10	species11
site1	10	8	6	4	2	0	0	0	0	0	0
site2	8	10	8	6	4	2	0	0	0	0	0
site3	6	8	10	8	6	4	2	0	0	0	0
site4	4	6	8	10	8	6	4	2	0	0	0
site5	2	4	6	8	10	8	6	4	2	0	0
site6	0	2	4	6	8	10	8	6	4	2	0
site7	0	0	2	4	6	8	10	8	6	4	2
site8	0	0	0	2	4	6	8	10	8	6	4
site9	0	0	0	0	2	4	6	8	10	8	6
site10	0	0	0	0	0	2	4	6	8	10	8
site11	0	0	0	0	0	0	2	4	6	8	10

## Problems with CA

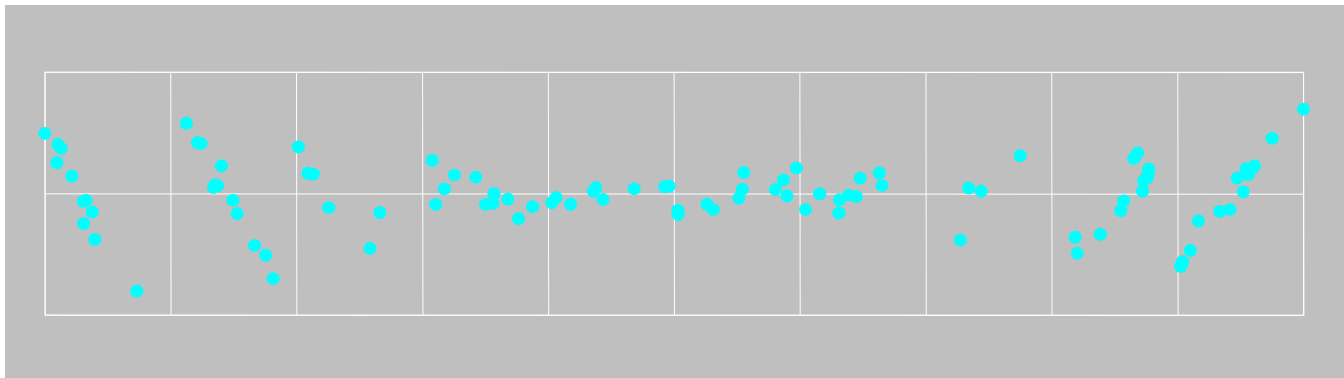
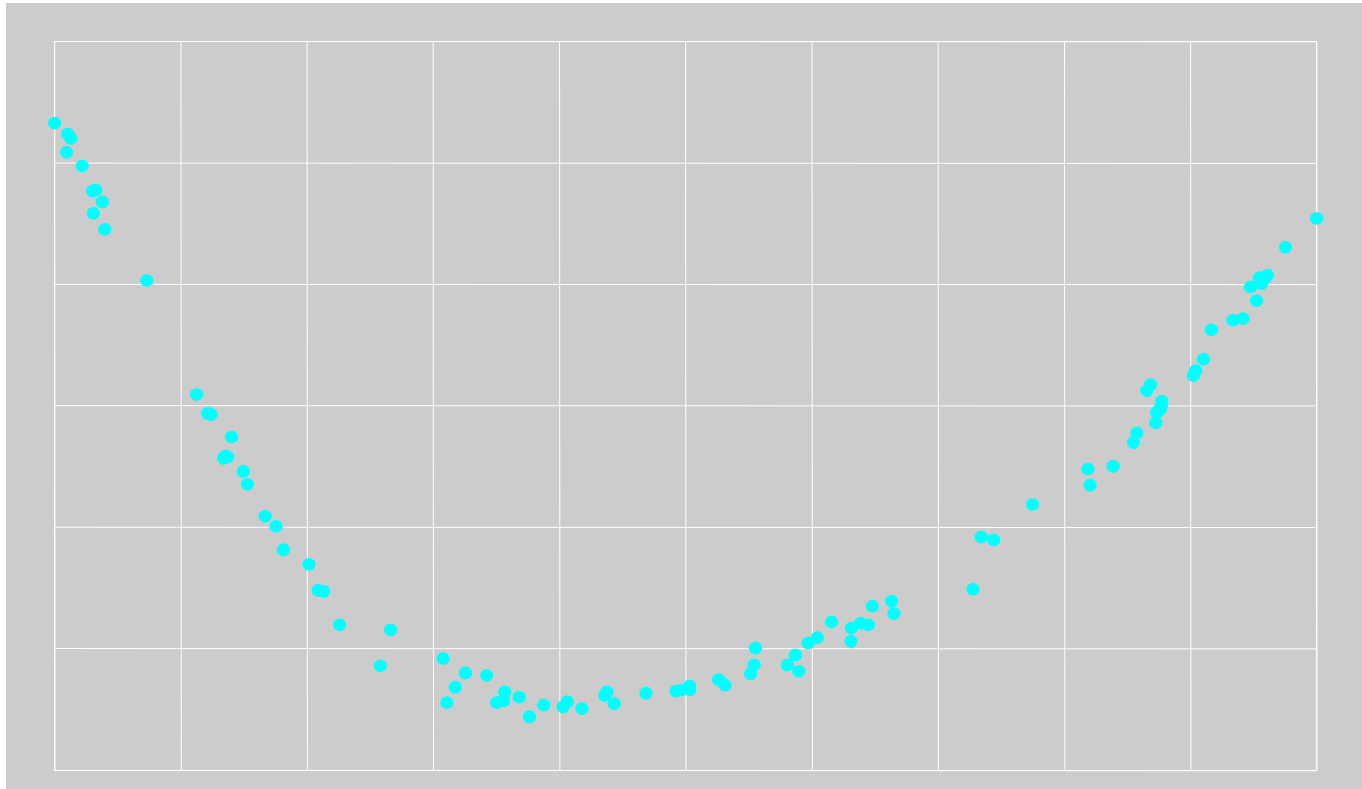
- The 1st CA axis is reliable, but 2nd and later axes are quadratic distortions of the first – produces the “arch effect”
- Distances compressed toward the ends of the axes and stretched in the middle
- Chi-square distance gives high weight to species with low abundance, which exaggerates distinctiveness of samples containing several rare species (Faith et al. 1987, Minchin 1987)



# Detrended correspondence analysis (DCA)

- The “arch effect” of CA is unwanted; the ends of the axes in CA are also compressed
- Detrending (detrended correspondence analysis, DCA) deals with the arch by:
  - 5 segment smoothing of 1<sup>st</sup> axis. Divide into segments (weights of 1,2,3,2,1), center each at 0.
  - Rescaling of axis into “standard deviation” units of species turnover.
- Only first 4 axes are adjusted, the rest are discarded
- Assumptions
  - Same as for CA
  - DCA is not really an analysis. It is a post hoc modification of a CA

# Detrended correspondence analysis (DCA)

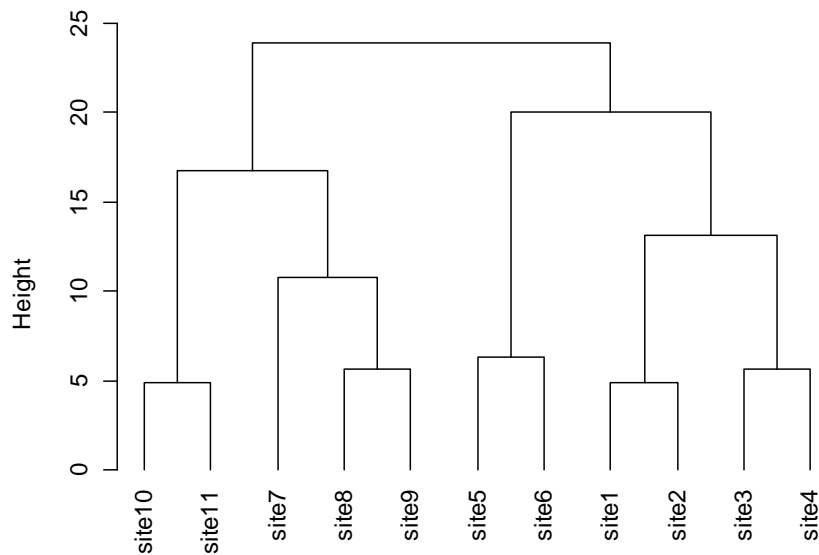


# R code: Detrended correspondence analysis (DCA)

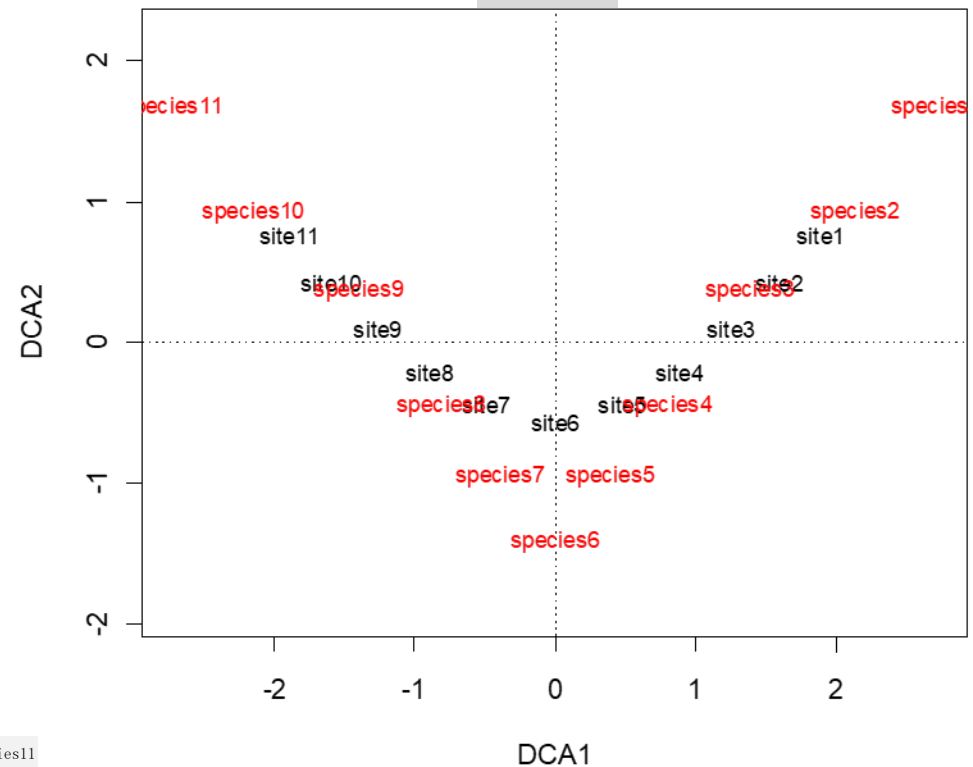
```
library(vegan); plot(decorana(species))
```

## Cluster analysis

### Cluster Dendrogram



## DCA



site	species1	species2	species3	species4	species5	species6	species7	species8	species9	species10	species11
site1	10	8	6	4	2	0	0	0	0	0	0
site2	8	10	8	6	4	2	0	0	0	0	0
site3	6	8	10	8	6	4	2	0	0	0	0
site4	4	6	8	10	8	6	4	2	0	0	0
site5	2	4	6	8	10	8	6	4	2	0	0
site6	0	2	4	6	8	10	8	6	4	2	0
site7	0	0	2	4	6	8	10	8	6	4	2
site8	0	0	0	2	4	6	8	10	8	6	4
site9	0	0	0	0	2	4	6	8	10	8	6
site10	0	0	0	0	0	2	4	6	8	10	8
site11	0	0	0	0	0	0	2	4	6	8	10

# Assignment

General objectives: learn about PCA

- Develop a dataset to perform PCA
- Describe your data, e.g. X1, X2, X3, etc.
- Plot explained variance (by PCs), loadings and scores, explain the relationship between variables, between scores, and variable-score association.