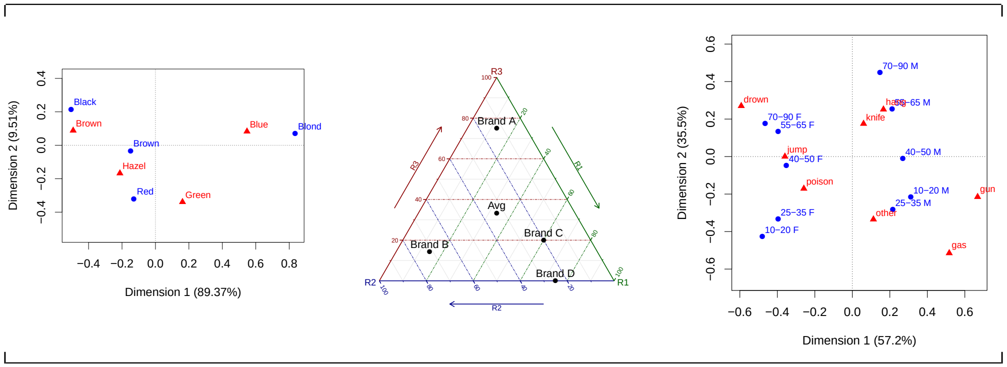
Correspondence analysis

Michael Friendly

Psych 6136

April 13, 2015



Correspondence analysis: Basic ideas

Correspondence analysis (CA)

Analog of PCA for frequency data:

- account for maximum % of χ^2 in few (2-3) dimensions
- finds scores for row (x_{im}) and column (y_{jm}) categories on these dimensions
- uses Singular Value Decomposition of residuals from independence, $d_{ij} = (n_{ij} - \widehat{m}_{ij}) / \sqrt{\widehat{m}_{ij}}$

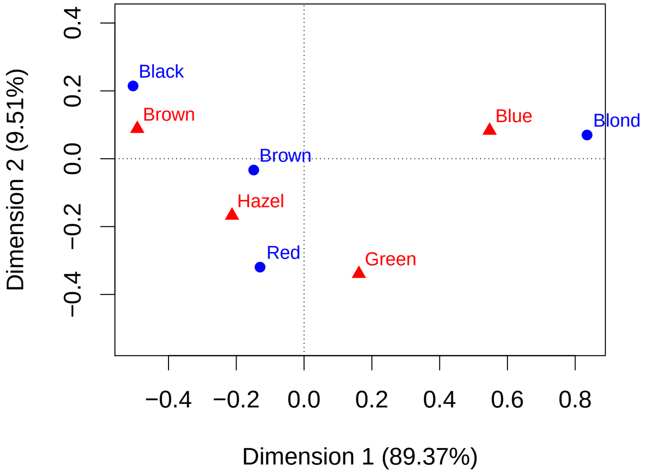
$$\frac{d_{ij}}{\sqrt{n}} = \sum_{m=1}^{M} \lambda_m \, x_{im} \, y_{jm}$$

- *optimal scaling*: each pair of scores for rows (x_{im}) and columns (y_{jm}) have highest possible correlation ($= \lambda_m$).
- plots of the row (x_{im}) and column (y_{jm}) scores show associations

Example: Hair color, eye color data

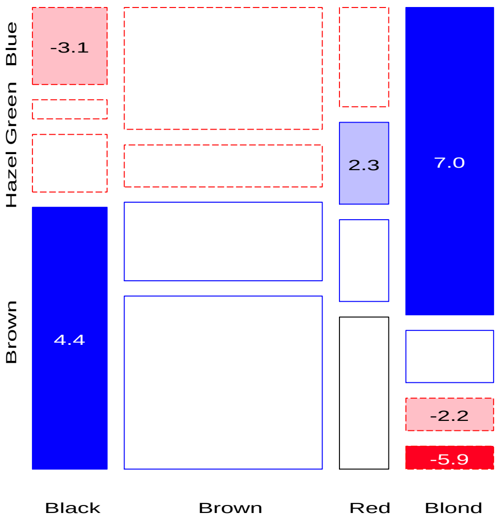
```
library(ca)
(haireye.ca <- ca(haireye))

##
##  Principal inertias (eigenvalues):
##           1        2        3
## Value     0.208773 0.022227 0.002598
## Percentage 89.37%   9.52%    1.11%
##
##
##  Rows:
##              Black     Brown       Red   Blond
## Mass      0.182432  0.483108  0.119932 0.21453
## ChiDist   0.551192  0.159461  0.354770 0.83840
## Inertia   0.055425  0.012284  0.015095 0.15079
## Dim. 1   -1.104277 -0.324463 -0.283473 1.82823
## Dim. 2    1.440917 -0.219111 -2.144015 0.46671
##
##
##  Columns:
##              Brown    Blue     Hazel     Green
## Mass      0.371622 0.36318  0.157095  0.108108
## ChiDist   0.500487 0.55368  0.288654  0.385727
## Inertia   0.093086 0.11134  0.013089  0.016085
## Dim. 1   -1.077128 1.19806 -0.465286  0.354011
## Dim. 2    0.592420 0.55642 -1.122783 -2.274122
```

Hair color, Eye color data:



- Rough interpretation: row/column points “near” each other are positively associated
- Dim 1: 89.4% of χ^2 (dark ↔ light)
- Dim 2: 9.5% of χ^2 (Red/Green vs. others)

Hair color, Eye color data: Compare with mosaic display



- The main dark–light dimension is reflected in the opposite-corner pattern of residuals
- The 2nd dimension is reflected in deviations from this pattern (e.g., Red hair–Green eyes)
- CA is "accounting for" residuals (deviations) from independence

Row and column profiles

- For a two-way table, row profiles and column profiles give the relative proportions of the column/row categories.
- An association is present to the extent that the row/col profiles differ
- Profiles add to 1.0 (100%), and can be visualized in profile space

Example: Toothpaste purchases by region

120 people in three regions where asked which of four brands of toothpaste, A–D, they had most recently purchased. Is there a difference among regions?

```
toothpaste

##            Region
## Brand       R1 R2 R3
##    Brand A   5  5 30
##    Brand B   5 25  5
##    Brand C  15  5  5
##    Brand D  15  5  0
```

Row and column profiles

- Row profiles pertain to the differences among brand preference
- Column profiles pertain to the differences among regions

```
##            R1   R2   R3 Sum
## Brand A 12.5 12.5 75.0 100
## Brand B 14.3 71.4 14.3 100
## Brand C 60.0 20.0 20.0 100
## Brand D 75.0 25.0  0.0 100
## Avg     33.3 33.3 33.3 100
```

```
##              R1    R2    R3   Avg
## Brand A    12.5  12.5  75.0  33.3
## Brand B    12.5  62.5  12.5  29.2
## Brand C    37.5  12.5  12.5  20.8
## Brand D    37.5  12.5   0.0  16.7
## Sum       100.0 100.0 100.0 100.0
```

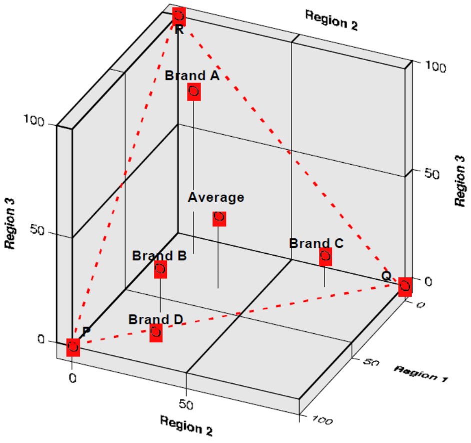
There is clearly an association, meaning that the row (column) profiles differ

```
chisq.test(toothpaste)

##
##  Pearson's Chi-squared test
##
## data:  toothpaste
## X-squared = 79.607, df = 6, p-value = 4.307e-15
```

Plotting profiles

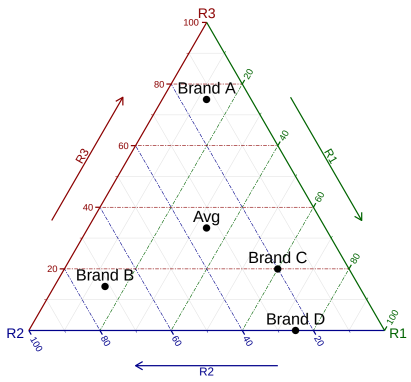
In this simple example we can plot the row profiles as points in 3D space, with axes corresponding to regions, R1, R2, R3

- Each brand is positioned in this space according to its proportions for the regions
- Because proportions sum to 100%, all points lie in the dashed plane PQR
- The Average profile is at the (weighted) centroid
- If no association, all brands would appear at the centroid



Plotting profiles

Analogous 2D plot is a trilinear plot that automatically scales the R1–R3 values so they sum to 100%



- The Avg profile has coordinates of 33.3% for each region
- Brand preferences by region can be seen by their positions wrt the R1–R3 axes
- This suggests that differences among brands can be measured by their (squared) distances from the centroid, weighted by their row margins (mass)
- Physical analogy suggests the term inertia for this weighted variation

CA solution

- The CA solution has at most min$(r-1, c-1)$ dimensions
- A 2D solution here is exact, i.e., accounts for 100% of Pearson X^2

```
library(ca)
tp.ca <- ca(toothpaste)
summary(tp.ca, rows=FALSE, columns=FALSE)

##
## Principal inertias (eigenvalues):
##
##  dim    value      %   cum%   scree plot
##  1      0.410259  61.8  61.8   ***************
##  2      0.253134  38.2 100.0   **********
##         -------- -----
##  Total: 0.663393 100.0
```

Pearson X^2:

```
sum(tp.ca$sv^2) * 120

## [1] 79.607
```

CA solution

```
res <- plot(tp.ca)
polygon(res$cols, border="red", lwd=2)
```

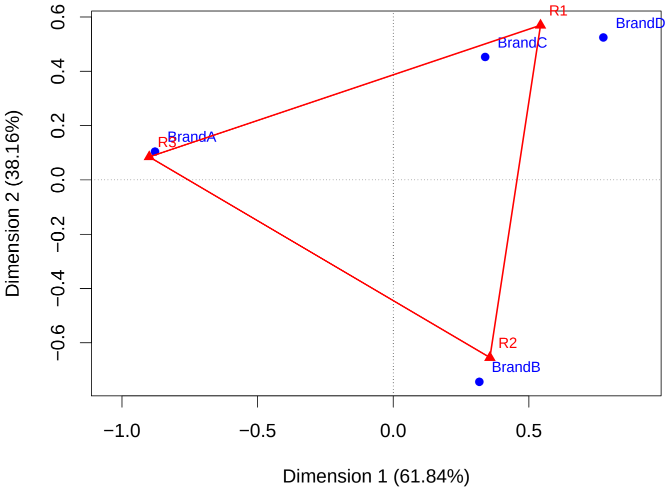
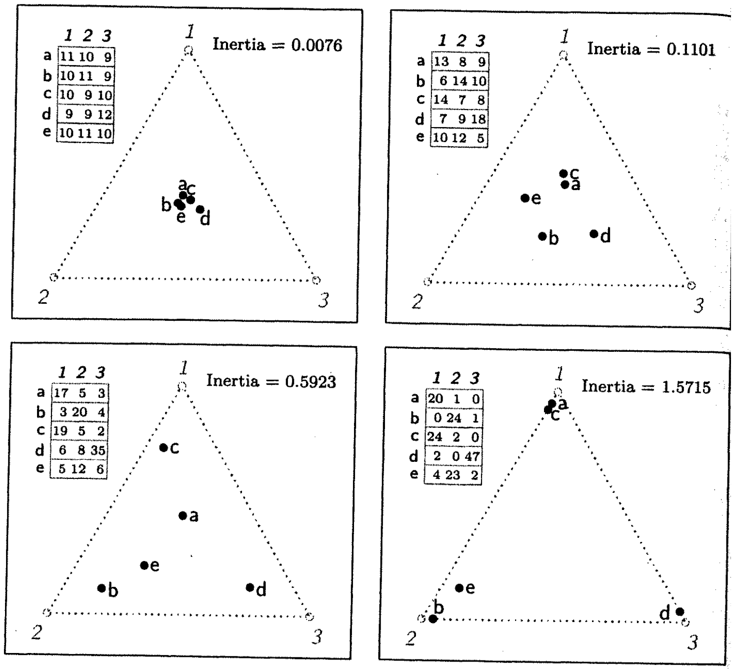


Exhibit 4.2: *A series of data tables with increasing total inertia. The higher the total inertia, the greater is the association between the rows and columns, displayed by the higher dispersion of the profile points in the profile space. The values in these tables have been chosen specifically so that the column sums are all equal, so the weights in the χ^2-distance formulation are the same, and hence distances we observe in these maps are true χ^2-distances.*



	1	2	3
a	11	10	9
b	10	11	9
c	10	9	10
d	9	9	12
e	10	11	10

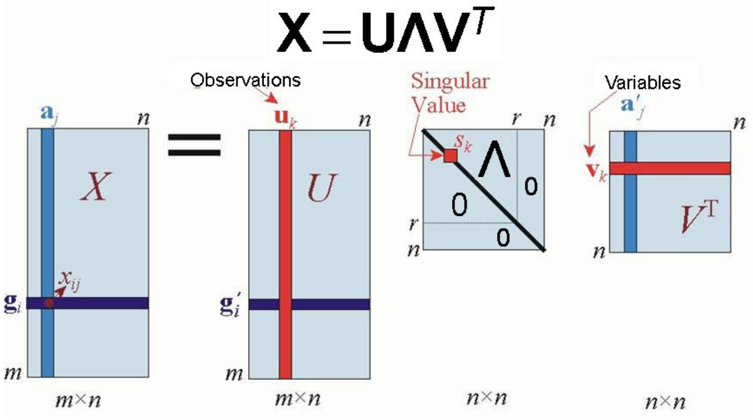
	1	2	3
a	13	8	9
b	6	14	10
c	14	7	8
d	7	9	18
e	10	12	5

	1	2	3
a	17	5	3
b	3	20	4
c	19	5	2
d	6	8	35
e	5	12	6

	1	2	3
a	20	1	0
b	0	24	1
c	24	2	0
d	2	0	47
e	4	23	2

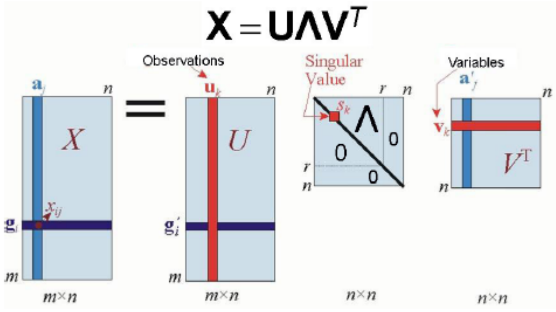
Singular value decomposition

The singular value decomposition (SVD) is a basic technique for factoring a matrix and matrix approximation.
For an $m \times n$ matrix $\boldsymbol{X}$ of rank $r \leq \min(m, n)$ the SVD of $\boldsymbol{X}$ is:



Properties of the SVD

- **V**: columns are the eigenvectors of $\boldsymbol{X}^T\boldsymbol{X}$ and form an orthonormal basis ($\mathbf{V}^T\mathbf{V} = \mathbf{I}$) for the variables
- **Λ**: diagonal, r singular values are the square roots of the eigenvalues of both $\boldsymbol{XX}^T$ and $\boldsymbol{X}^T\boldsymbol{X}$
- **U**: columns are the eigenvectors of $\boldsymbol{XX}^T$ and form an orthonormal basis for the observation profiles, so that $\mathbf{U}^T\mathbf{U} = \mathbf{I}$



SVD: Matrix approximation

- Let **X** be an m x n matrix such that rank(**X**) = r
- If $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_r$ are the singular values of **X**, then $\hat{\mathbf{X}}$, the rank q approximation of **X** that minimizes $\| \mathbf{X} - \hat{\mathbf{X}} \|$, is

column scores

$$\hat{\mathbf{X}}_{m \times n} = \sum_{i=1}^{q} \lambda_i \begin{pmatrix} u_{i1} \\ \vdots \\ u_{im} \end{pmatrix} \begin{pmatrix} v_{i1} & \cdots & v_{in} \end{pmatrix} = \lambda_1 u_1 v_1^T + \cdots + \lambda_q u_q v_q^T$$

row scores

a sum of q rank=1 (outer) products. The variance in **X** accounted for each term is $\lambda_1{}^2$

CA notation and terminology

Notation:

- Contingency table: $\boldsymbol{N} = \{n_{ij}\}$
- Correspondence matrix (cell probabilities): $\boldsymbol{P} = \{p_{ij}\} = \boldsymbol{N}/n$
- Row/column masses (marginal probabilities): $\boldsymbol{r} = \sum_j p_{ij}$ and $\boldsymbol{c} = \sum_i p_{ij}$
- Diagonal weight matrices: $\boldsymbol{D}_r = \mathrm{diag}\,(\boldsymbol{r})$ and $\boldsymbol{D}_c = \mathrm{diag}\,(\boldsymbol{c})$

The SVD is then applied to the correspondence matrix of cell probabilities as:

$$\boldsymbol{P} = \boldsymbol{A} \boldsymbol{D}_\lambda \boldsymbol{B}^\mathsf{T}$$

where

- Singular values: $\boldsymbol{D}_\lambda = \mathrm{diag}\,(\boldsymbol{\lambda})$ is the diagonal matrix of singular values $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_M$
- Row scores: $\boldsymbol{A}_{I \times M}$, normalized so that $\boldsymbol{A}\boldsymbol{D}_r^{-1}\boldsymbol{A}^\mathsf{T} = \boldsymbol{I}$
- Column scores: $\boldsymbol{B}_{J \times M}$, normalized so that $\boldsymbol{B}\boldsymbol{D}_c^{-1}\boldsymbol{B}^\mathsf{T} = \boldsymbol{I}$

Principal and standard coordinates

Two types of coordinates are commonly used in CA, based on re-scalings of $\boldsymbol{A}$ and $\boldsymbol{B}$.

Principal coordinates

Coordinates of the row ($\boldsymbol{F}$) and column ($\boldsymbol{G}$) profiles *wrt* their own principal axes

$$\begin{aligned} \boldsymbol{F} &= \boldsymbol{D}_r^{-1}\boldsymbol{A}\boldsymbol{D}_\lambda \quad \text{scaled so that} \quad \boldsymbol{F}^\mathsf{T}\boldsymbol{D}_r\boldsymbol{F} = \boldsymbol{D}_\lambda \\ \boldsymbol{G} &= \boldsymbol{D}_c^{-1}\boldsymbol{B}\boldsymbol{D}_\lambda \quad \text{scaled so that} \quad \boldsymbol{G}^\mathsf{T}\boldsymbol{D}_c\boldsymbol{G} = \boldsymbol{D}_\lambda \end{aligned}$$

- Defined so that the inertia along each axis is the corresponding singular value, λ_i,
- i.e., weighted average of squared principal coordinates = λ_i on dim. i
- The joint plot in principal coordinates, $\boldsymbol{F}$ and $\boldsymbol{G}$, is called the symmetric map because both row and column profiles are overlaid in the same coordinate system.

Principal and standard coordinates

Standard coordinates

The standard coordinates $(\boldsymbol{\Phi}, \boldsymbol{\Gamma})$ are a rescaling of the principal coordinates to unit inertia along each axis,

$$\begin{aligned} \boldsymbol{\Phi} &= \boldsymbol{D}_r^{-1}\boldsymbol{A} \quad \text{scaled so that} \quad \boldsymbol{\Phi}^\mathsf{T}\boldsymbol{D}_r\boldsymbol{\Phi} = \boldsymbol{I} \\ \boldsymbol{\Gamma} &= \boldsymbol{D}_c^{-1}\boldsymbol{B} \quad \text{scaled so that} \quad \boldsymbol{\Gamma}^\mathsf{T}\boldsymbol{D}_c\boldsymbol{\Gamma} = \boldsymbol{I} \end{aligned}$$

- The weighted average of squared standard coordinates = 1 on each dimension
- An asymmetric map shows one set of points (say, the rows) in principal coordinates and the other set in standard coordinates.
-

Geometric and statistical properties

nested solutions: CA solutions are *nested*, meaning that the first two dimensions of a 3D solution will be identical to the 2D solution (similar to PCA)

centroids at the origin: In both principal coordinates and standard coordinates the points representing the row and column profiles have their centroids (weighted averages) at the origin. The origin represents the (weighted) average row profile and column profile.

chi-square distances: In principal coordinates, the row coordinates are equal to the row profiles $\boldsymbol{D}_r^{-1}\boldsymbol{P}$, rescaled inversely by the square-root of the column masses, $\boldsymbol{D}_c^{-1/2}$. Distances between two row profiles, $\boldsymbol{R}_i$ and $\boldsymbol{R}_{i'}$ are χ^2 distances, where the squared difference $[\boldsymbol{R}_{ij} - \boldsymbol{R}_{i'j}]^2$ is inversely weighted by the column frequency, to account for the different relative frequency of the column categories.

The ca package in R

- `ca()` calculates CA solutions, returning a `"ca"` object

```
names(haireye.ca)

##  [1] "sv"         "nd"         "rownames"   "rowmass"    "rowdist"
##  [6] "rowinertia" "rowcoord"   "rowsup"     "colnames"   "colmass"
## [11] "coldist"    "colinertia" "colcoord"   "colsup"     "call"
```

- The result contains the standard row coordinates (`rowcoord`: Φ) and column coordinates (`colcoord`: Γ)

```
haireye.ca$rowcoord

##            Dim1     Dim2     Dim3
## Black -1.10428  1.44092 -1.08895
## Brown -0.32446 -0.21911  0.95742
## Red   -0.28347 -2.14401 -1.63122
## Blond  1.82823  0.46671 -0.31809
```

The ca package in R

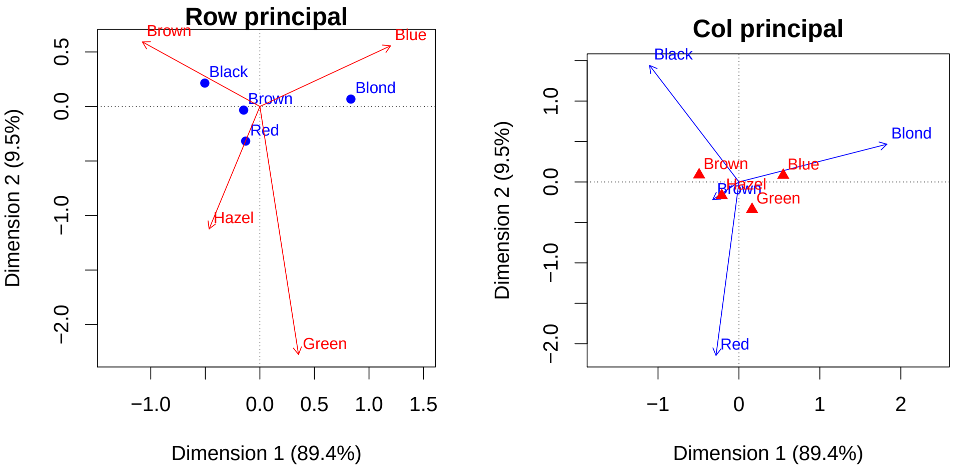
The `plot()` method provides a wide variety of scalings (`map=`), with different interpretive properties. Some of these are:

- `"symmetric"` — both rows and columns in pricipal coordinates (default)
- `"rowprincipal"` or `"colprincipal"` — asymmetric maps, with either rows in principal coordinates and columns in standard coordinates, or vice versa
- `"symbiplot"` — scales both rows and columns to have variances equal to the singular value

The `mcja()` function is used for multiple correspondence analysis (MCA) and has analogous `print()`, `summary()` and `plot()` methods.

Asymmetric row/col principal plots are biplots — can interpret projection of points on vectors

```r
plot(haireye.ca, map="rowprincipal", arrows=c(FALSE,TRUE))
plot(haireye.ca, map="colprincipal", arrows=c(TRUE,FALSE))
```



Optimal category scores

- CA has a close relation to canonical correlation analysis, applied to dummy variables representing the categories.
- The singular values, λ_i, are the correlations between the category scores
  - Assign Dim 1 scores, X1 and Y1 to the row/column categories: $\Longrightarrow$ max. possible correlation, λ_1
  - Assign Dim 2 scores, X2 and Y2 to the row/column categories: $\Longrightarrow$ max. possible correlation, λ_2, but uncorrelated with X1, Y1
  - Thus all association between the row/col categories is captured by the scores
- This optimal scaling interpretation can be used to quantify categorical variables

Optimal category scores

Singular values = canonical correlations

```r
haireye.ca <- ca(haireye)
round(haireye.ca$sv, 4)

## [1] 0.4569 0.1491 0.0510
```

Extract the row and column coordinates to a data frame

```r
RC <- haireye.ca$rowcoord  # row coordinates
CC <- haireye.ca$colcoord  # col coordinates
HE.df <- as.data.frame(haireye)

Y1 <- RC[match(HE.df$Hair, haireye.ca$rownames),1]
X1 <- CC[match(HE.df$Eye, haireye.ca$colnames),1]
Y2 <- RC[match(HE.df$Hair, haireye.ca$rownames),2]
X2 <- CC[match(HE.df$Eye, haireye.ca$colnames),2]
```

Optimal category scores

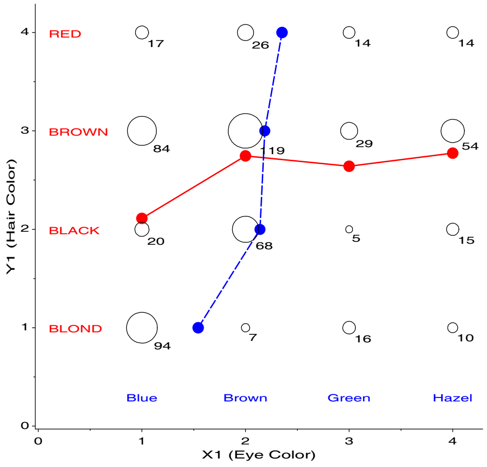
```
HE.df <- cbind(HE.df, X1, Y1, X2, Y2)
print(HE.df, digits=3)

##      Hair   Eye Freq     X1     Y1     X2     Y2
## 1  Black Brown   68 -1.077 -1.104  0.592  1.441
## 2  Brown Brown  119 -1.077 -0.324  0.592 -0.219
## 3    Red Brown   26 -1.077 -0.283  0.592 -2.144
## 4  Blond Brown    7 -1.077  1.828  0.592  0.467
## 5  Black  Blue   20  1.198 -1.104  0.556  1.441
## 6  Brown  Blue   84  1.198 -0.324  0.556 -0.219
## 7    Red  Blue   17  1.198 -0.283  0.556 -2.144
## 8  Blond  Blue   94  1.198  1.828  0.556  0.467
...
```

Calculate correlations—all zero except r(X1,Y1)=λ_1 and r(X2,Y2)=λ_2

```
corr <- cov.wt(HE.df[,4:7], wt=HE.df$Freq, cor=TRUE)$cor
round(zapsmall(corr), 4)

##        X1     Y1     X2     Y2
## X1 1.0000 0.4569 0.0000 0.0000
## Y1 0.4569 1.0000 0.0000 0.0000
## X2 0.0000 0.0000 1.0000 0.1491
## Y2 0.0000 0.0000 0.1491 1.0000
```

Simultaneous linear regressions

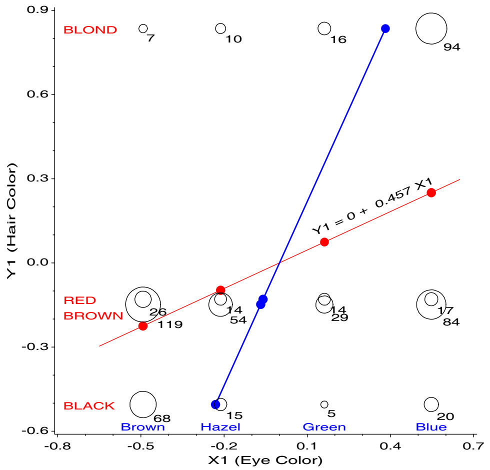
Assign arbitrary scores (1–4) X1 to eye color and Y1 to hair color



- Lines connecting the weighted (conditional) means of $Y1 \mid X1$ and $X1 \mid Y1$ are not-linear
- The scatterplot uses bubble symbols showing frequency in each cell
- Is it possible to assign row and column scores so that both regressions are linear?

Simultaneous linear regressions

Yes, use CA scores on the first dimension



- The regression of Y1 on X1 is linear, with slope λ_1
- The regression of X1 on Y1 is linear, with slope $1/\lambda_1$
- λ_1 is the (canonical) correlation between X1 and Y1
- The angle between the two lines would be 0 if perfect correlation
- The conditional means (dots) are the principal coordinates

Example: Mental impairment and parents' SES

Data on mental health status (`mental`) of 1660 young NYC residents by parents' SES (`ses`), a 6 × 4 table.

- Both `mental` and `ses` are ordered factors
- Convert from frequency data frame to table using **`xtabs()`**

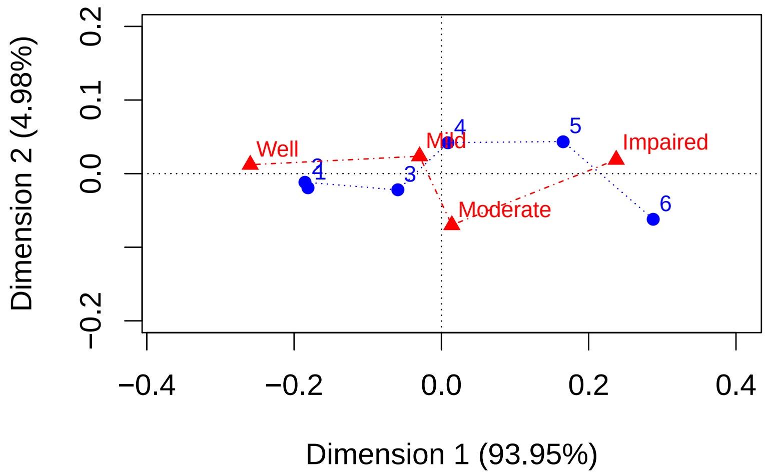
```
data("Mental", package="vcdExtra")
str(Mental)

## 'data.frame': 24 obs. of  3 variables:
##  $ ses   : Ord.factor w/ 6 levels "1"<"2"<"3"<"4"<..: 1 1 1 1 2 2 2 2 3
##  $ mental: Ord.factor w/ 4 levels "Well"<"Mild"<..: 1 2 3 4 1 2 3 4 1 2
##  $ Freq  : int  64 94 58 46 57 94 54 40 57 105 ...

mental.tab <- xtabs(Freq ~ ses + mental, data=Mental)
```

Example: Mental impairment and parents' SES

```
mental.ca <- ca(mental.tab)
summary(mental.ca)

##
## Principal inertias (eigenvalues):
##
##  dim    value      %   cum%   scree plot
##  1      0.026025  93.9  93.9  ***********************
##  2      0.001379   5.0  98.9  *
##  3      0.000298   1.1 100.0
##         -------- -----
##  Total: 0.027702 100.0
...
```

- The exact CA solution has $\min(r-1, c-1) = 3$ dimensions
- The total Pearson X^2 is $n \Sigma \lambda_i^2 = 1660 \times 0.0277 = 45.98$ with 15 df
- Of this, 93.9% is accounted for by the first dimension

```
res <- plot(mental.ca)
lines(res$rows, col="blue", lty=3)
lines(res$cols, col="red", lty=4)
```



Looking ahead

- CA is largely an exploratory method — row/column scores are not parameters of a statistical model; no confidence intervals
- Only rough tests for the number of CA dimensions
- Can't test an hypothesis that the row/column scores are have some particular spacing (e.g., are `mental` and `ses` equally spaced?)
- These kinds of questions can be answered with specialized loglinear models
- Nevertheless, `plot(ca(table))` gives an excellent quick view of associations

Multi-way tables

Correspondence analysis can be extended to *n*-way tables in several ways:

Stacking approach

- *n*-way table flattened to a 2-way table, combining several variables "interactively"
- Each way of stacking corresponds to a *loglinear model*
- Ordinary CA of the flattened table → visualization of that model
- Associations among stacked variables are *not visualized*

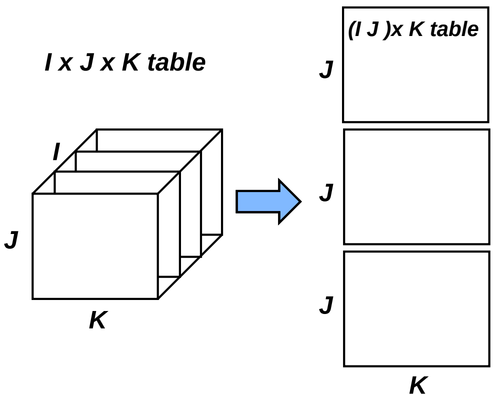
Multiple correspondence analysis (MCA)

- Extends CA to *n*-way tables
- Analyzes all pairwise bivariate associations
- Can plot all factors in a single plot
- An extension, joint correspondence analysis, gives a better account of inertia for each dimension

Multi-way tables: Stacking

Stacking approach:

- three-way table, of size $I \times J \times K$ can be sliced and stacked as a two-way table in different ways



- The variables combined are treated “interactively”
- Each way of stacking corresponds to a loglinear model
  - $(I \times J) \times K \rightarrow$ [AB][C]
  - $I \times (J \times K) \rightarrow$ [A][BC]
  - $J \times (I \times K) \rightarrow$ [B][AC]
- Only the associations in separate [ ] terms are analyzed and displayed
- The stacked table is analyzed with ordinary CA of the two-way stacked table

Interactive coding in R

- Data in table (array) form: Use `as.matrix(structable())`

```
mat1 <- as.matrix(structable(A + B ~ C, data=mytable))  # [A B][C]
mat2 <- as.matrix(structable(A + C ~ B + D, data=mytable)) # [A C][B D
ca(mat2)
```

- Data in frequency data frame form: Use `paste()` or `interaction()`, followed by `xtabs()`

```
mydf$AB <- interaction(mydf$A, mydf$B, sep='.')  # levels: A.B
mydf$AB <- paste(mydf$A, mydf$B, sep=':')        # levels: A:B
...
mytab <- xtabs(Freq ~ AB + C, data=mydf)         # [A B] [C}
```

Example: Suicide rates in Germany

- `Suicide` in vcd gives a $2 \times 5 \times 8$ table of `sex` by `age.group` by `method` of suicide for 53,182 suicides in Germany, in a frequency data frame
- With the data in this form, you can use `paste()` to join `age.group` and `sex` together to form a new variable `age_sex` consisting of their combinations.

```
data("Suicide", package="vcd")
# interactive coding of sex and age.group
Suicide <- within(Suicide, {
        age_sex <- paste(age.group, toupper(substr(sex,1,1)))
        })
```

Example: Suicide rates in Germany

```
suicide.tab <- xtabs(Freq ~ age_sex + method2, data=Suicide)
suicide.tab

##           method2
## age_sex    poison  gas hang drown  gun knife jump other
##   10-20 F     921   40  212    30   25    11  131   100
##   10-20 M    1160  335 1524    67  512    47  189   464
##   25-35 F    1672  113  575   139   64    41  276   263
##   25-35 M    2823  883 2751   213  852   139  366   775
##   40-50 F    2224   91 1481   354   52    80  327   305
##   40-50 M    2465  625 3936   247  875   183  244   534
##   55-65 F    2283   45 2014   679   29   103  388   296
##   55-65 M    1531  201 3581   207  477   154  273   294
##   70-90 F    1548   29 1355   501    3    74  383   106
##   70-90 M     938   45 2948   212  229   105  268   147
```

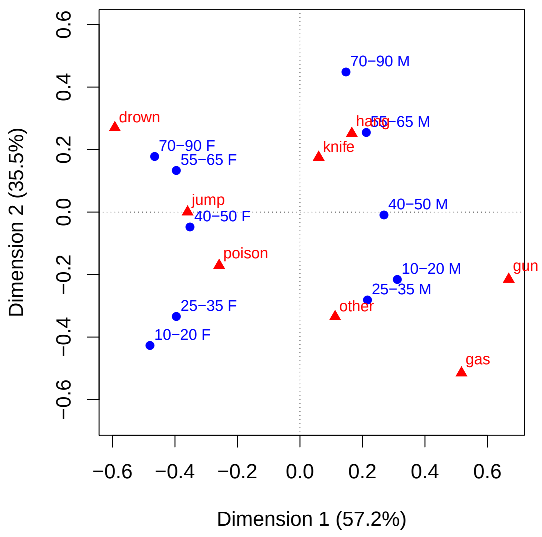
- The CA analysis will be that of the loglinear model [Age Sex] [Method]
- It will show associations between the age–sex combinations and method of suicide
- Associations between age and sex will not be shown in this analysis

Example: Suicide rates in Germany

```
suicide.ca <- ca(suicide.tab)
summary(suicide.ca)

##
## Principal inertias (eigenvalues):
##
##  dim    value      %   cum%   scree plot
##  1      0.096151  57.2  57.2  **************
##  2      0.059692  35.5  92.6  *********
##  3      0.008183   4.9  97.5  *
##  4      0.002158   1.3  98.8
##  5      0.001399   0.8  99.6
##  6      0.000557   0.3 100.0
##  7      6.7e-050   0.0 100.0
##         -------- -----
##  Total: 0.168207 100.0
...
```

It can be seen that 92.6% of the Pearson X^2 for this model is accounted for in the first two dimensions.

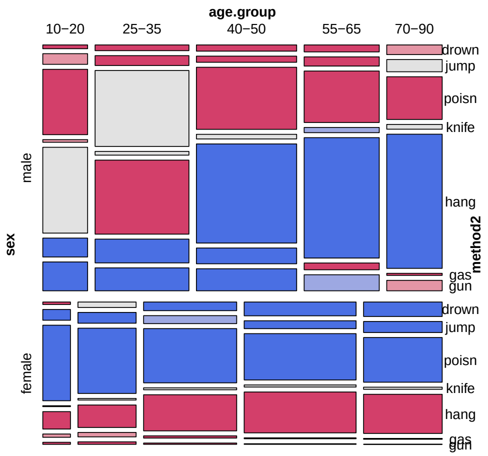
```
plot(suicide.ca)
```



- Dim 1: Sex
- Dim 2: Age
- Interpret method use by age-sex combinations, e.g., young males: gas, gun; young females: poison

Compare with a mosaic plot also fitting the model [Age Sex][Suicide]:

```
suicide.tab3 <- xtabs(Freq ~ sex + age.group + method2, data=Suicide)
mosaic(suicide.tab3, shade=TRUE, legend=FALSE,
       expected=~age.group*sex + method2,
       labeling_args=list(abbreviate_labs=c(FALSE, FALSE, 5)),
                          rot_labels = c(0, 0, 0, 90))
```



Marginal tables and supplementary variables

- An *n*-way table is collapsed to a marginal table by ignoring factors
- Omitted variables can be included by treating them as supplementary
- These are projected into the space of the marginal CA

Age by method, ignoring sex:

```
suicide.tab2 <- xtabs(Freq ~ age.group + method2, data=Suicide)
suicide.tab2

##          method2
## age.group poison  gas hang drown  gun knife jump other
##     10-20   2081  375 1736    97  537    58  320   564
##     25-35   4495  996 3326   352  916   180  642  1038
##     40-50   4689  716 5417   601  927   263  571   839
##     55-65   3814  246 5595   886  506   257  661   590
##     70-90   2486   74 4303   713  232   179  651   253
```

Relation of sex and method:

```
(suicide.sup <- xtabs(Freq ~ sex + method2, data=Suicide))

##         method2
## sex      poison   gas  hang drown   gun knife  jump other
##   male     8917  2089 14740   946  2945   628  1340  2214
##   female   8648   318  5637  1703   173   309  1505  1070

suicide.tab2s <- rbind(suicide.tab2, suicide.sup)
```

Marginal tables and supplementary variables

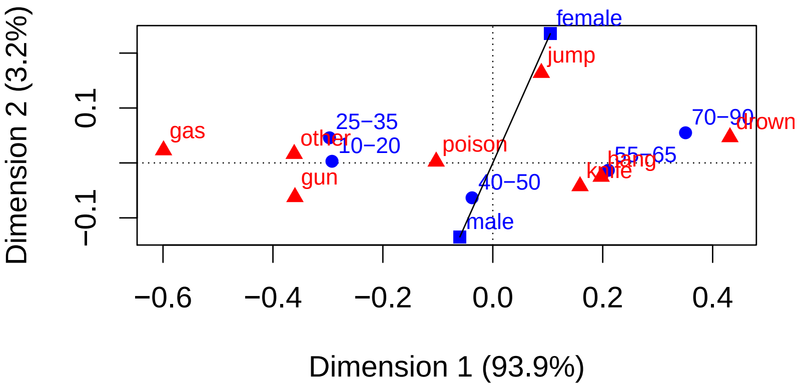
The rows for sex by method are treated as supplementary rows:

```
suicide.ca2s <- ca(suicide.tab2s, suprow=6:7)
summary(suicide.ca2s)

##
## Principal inertias (eigenvalues):
##
##  dim    value      %   cum%   scree plot
##  1      0.060429  93.9  93.9  ***********************
##  2      0.002090   3.2  97.1  *
##  3      0.001479   2.3  99.4  *
##  4      0.000356   0.6 100.0
##         -------- -----
##  Total: 0.064354 100.0
##
...
```

- the relation of age and method is now essentially 1 dimensional
- the inertia of Dim 1 (0.604) is nearly the same as that of Dim 2 (0.596) in the stacked table

- Plot the 2D CA solution for the [Age] [Method] marginal table.
- Add category points for Sex (supplementary points)

```
res <- plot(suicide.ca2s, pch=c(16, 15, 17, 24))
lines(res$rows[6:7,])
```



Ignoring Sex has collapsed Dim 1 (Sex) of the [Age Sex][Method] analysis. Supp. points show associations of Method with Sex.

Multiple correspondence analysis (MCA)

- Extends CA to *n*-way tables
- Useful when simpler stacking approach doesn't work well, e.g., 10 categorical attitude items
- Analyzes all pairwise bivariate associations. Analogous to:
  - Correlation matrix (numbers)
  - Scatterplot matrix (graphs)
  - All pairwise χ^2 tests (numbers)
  - Mosaic matrix (graphs)
- Provides an optimal scaling of the category scores for each variable
- Can plot all factors in a single plot
- An extension, joint correspondence analysis, gives a better account of inertia for each dimension

The indicator matrix and the Burt matrix

Two ways to think about MCA:

Indicator matrix (dummy variables)

- A given categorical variable, q, can be represented by an indicator matrix $\boldsymbol{Z}(n \times J_q)$ of dummy variables, $z_{ij} = 1$ if case i is in category j
- Let $\boldsymbol{Z}_1, \boldsymbol{Z}_2, \dots, \boldsymbol{Z}_Q$ be the indicator matrices for Q variables
- MCA is then a simple CA applied to the partitioned matrix $\boldsymbol{Z} = [\boldsymbol{Z}_1, \boldsymbol{Z}_2, \dots, \boldsymbol{Z}_Q]$

Burt matrix

- The Bert matrix is the product of the indicator matrix $\boldsymbol{Z}$ and its transpose

$$\boldsymbol{B} = \boldsymbol{Z}^{\mathsf{T}}\boldsymbol{Z}$$

- MCA can be defined using the SVD of $\boldsymbol{B}$, giving category scores for all variables accounting for the largest proportion of all bivariate associations.

Bivariate MCA: Hair Eye color

- For the hair-eye color data, the indicator matrix ***Z*** has n=592 rows (observations) and $4 + 4 = 8$ columns (categories).
- Shown below in frequency form, using h1–h4 for hair color and e1–e4 for eye color
- E.g., first row reflects the 68 observations with black hair and brown eyes

```
##     Hair   Eye Freq h1 h2 h3 h4 e1 e2 e3 e4
## 1  Black Brown   68  1  0  0  0  1  0  0  0
## 2  Brown Brown  119  0  1  0  0  1  0  0  0
## 3    Red Brown   26  0  0  1  0  1  0  0  0
## 4  Blond Brown    7  0  0  0  1  1  0  0  0
## 5  Black  Blue   20  1  0  0  0  0  1  0  0
## 6  Brown  Blue   84  0  1  0  0  0  1  0  0
## 7    Red  Blue   17  0  0  1  0  0  1  0  0
## 8  Blond  Blue   94  0  0  0  1  0  1  0  0
...
```

Expand this to case form for ***Z*** (592 x 8)

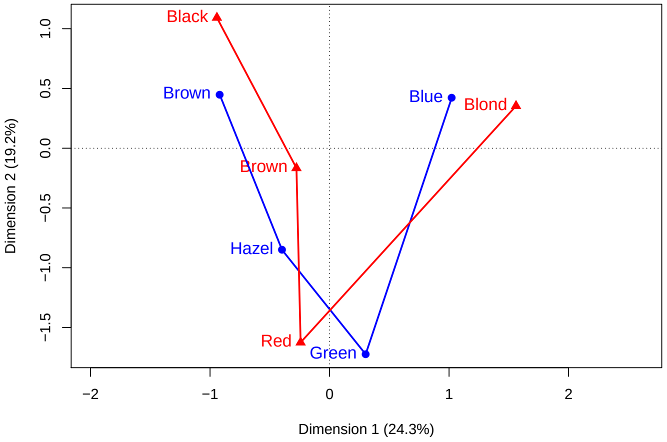
```
Z <- expand.dft(haireye.df)[,-(1:2)]
vnames <- c(levels(haireye.df$Hair), levels(haireye.df$Eye))
colnames(Z) <- vnames
dim(Z)

## [1] 592   8
```

If the indicator matrix is partitioned as $\boldsymbol{Z} = [\boldsymbol{Z}_1, \boldsymbol{Z}_2]$, corresponding to the hair, eye categories, then the contingency table is given by $\boldsymbol{N} = \boldsymbol{Z}_1^\mathsf{T}\boldsymbol{Z}_2$.

```
(N <- t(as.matrix(Z[,1:4])) %*% as.matrix(Z[,5:8]))

##        Brown Blue Hazel Green
## Black     68   20    15     5
## Brown    119   84    54    29
## Red       26   17    14    14
## Blond      7   94    10    16
```

- We can then use ordinary CA on the indicator matrix, ***Z***
- Except for scaling, this is the same as the CA of ***N***
- The inertia contributions differ, and this is handled better by MCA

```
Z.ca <- ca(Z)
res <- plot(Z.ca, what=c("none", "all"))
```



The Burt matrix

For two categorical variables, the Burt matrix is

$$\boldsymbol{B} = \boldsymbol{Z}^\mathsf{T}\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{N}_1 & \boldsymbol{N} \\ \boldsymbol{N}^\mathsf{T} & \boldsymbol{N}_2 \end{bmatrix} .$$

- $\boldsymbol{N}_1$ and $\boldsymbol{N}_2$ are diagonal matrices containing the marginal frequencies of the two variables
- The contingency table, ***N*** appears in the off-diagonal block

A similar analysis to that of the indicator matrix ***Z*** is produced by:

```
Burt <- t(as.matrix(Z)) %*% as.matrix(Z)
rownames(Burt) <- colnames(Burt) <- vnames
Burt.ca <- ca(Burt)
plot(Burt.ca)
```

- Standard coords are the same
- Singular values of ***B*** are the squares of those of ***Z***

Multivariate MCA

For Q categorical variables, the Burt matrix is

$$\mathbf{B} = \mathbf{Z}^{\mathsf{T}}\mathbf{Z} = \begin{bmatrix} \mathbf{N}_1 & \mathbf{N}_{[12]} & \cdots & \mathbf{N}_{[1Q]} \\ \mathbf{N}_{[21]} & \mathbf{N}_2 & \cdots & \mathbf{N}_{[2Q]} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{N}_{[Q1]} & \mathbf{N}_{[Q2]} & \cdots & \mathbf{N}_Q \end{bmatrix} .$$

- The diagonal blocks $\mathbf{N}_i$ contain the one-way marginal frequencies
- The off-diagonal blocks $\mathbf{N}_{[ij]}$ contain the bivariate contingency tables for each pair (i, j) of variables.
- Classical MCA can be defined as a SVD of the matrix $\mathbf{B}$
- It produces scores for the categories of *all* variables accounting for the greatest proportion of the bivariate associations in off-diagonal blocks in a small number of dimensions.

MCA properties

- The inertia contributed by a given variable increases with the number of response categories: inertia $(\mathbf{Z}_q) = J_q - 1$
- The centroid of the categories for each variable is at the origin of the display.
- For a given variable, the inertia contributed by a given category increases as the marginal frequency in that category *decreases*. Low frequency points therefore appear further from the origin.
- The category points for a binary variable lie on a line through the origin.

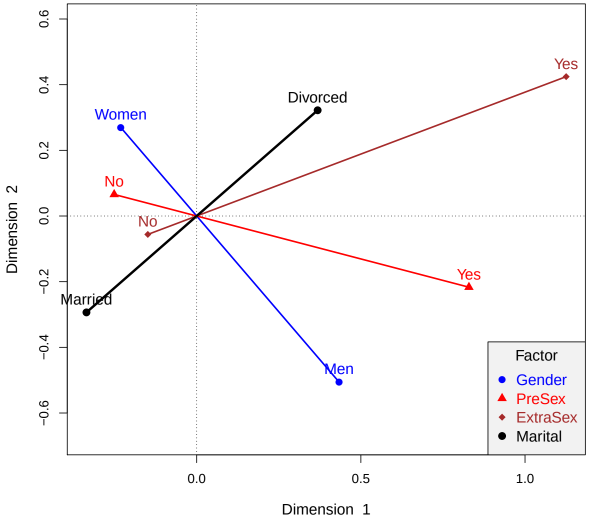
MCA example: Pre- and extramarital sex

- `PreSex` data: the 2 × 2 × 2 × 2 table of gender, premarital sex, extramatrial sex and marital status (divorced, still married)
- The function `mjca()` provides several scalings for the singular values
- Here I use `lambda="Burt"`

```
data("PreSex", package="vcd")
PreSex <- aperm(PreSex, 4:1)   # order variables G, P, E, M
presex.mca <- mjca(PreSex, lambda="Burt")
summary(presex.mca)

##
## Principal inertias (eigenvalues):
##
##  dim    value      %   cum%   scree plot
##  1      0.149930  53.6  53.6  *************
##  2      0.067201  24.0  77.6  ******
##  3      0.035396  12.6  90.2  ***
##  4      0.027365   9.8 100.0  **
##         -------- -----
##  Total: 0.279892 100.0
...
```

MCA example: Pre- and extramarital sex

```
plot(presex.mca)
```



Inertia in MCA

- In simple CA, total inertia $= \Sigma \lambda_i^2 = \chi^2 / n$
- $\Longrightarrow$ sensible to consider % inertia for each dimension

Not so straight-forward in MCA:

- For a given indicator matrix, $\boldsymbol{Z}_q$, the inertia is $J_q - 1$
- For all variables, with $J = \Sigma J_q$ categories, the total inertia of $\boldsymbol{Z} = [\boldsymbol{Z}_1, \dots, \boldsymbol{Z}_Q]$ is the average of the inertias of the sub-tables

$$inertia(\boldsymbol{Z}) = \frac{1}{Q} \sum_q inertia(\boldsymbol{Z}_q) = \frac{1}{Q} \sum_q (J_q - 1) = \frac{J - Q}{Q}$$

- The average inertia per dimension is therefore 1/Q
- $\Longrightarrow$ Interpret dimensions with inertia $> 1/Q$ (as in PCA: $\lambda > 1$)
- In analysis of the Burt matrix, average inertia is inflated by the diagonal blocks

Inertia in MCA

Two solutions:

Adjusted inertia

- Ignores the diagonal blocks in the Burt matrix
- Calculates adjusted inertia as

$$(\lambda_i^\star)^2 = \left[\frac{Q}{Q-1} (\lambda_i^Z - \frac{1}{Q}) \right]^2$$

- Express contributions of dimensions as $(\lambda_i^\star)^2 / \sum (\lambda_i^\star)^2$, with summation over only dimensions with $(\lambda^Z)^2 > 1/Q$.

Joint correspondence analysis

- Start with MCA analysis of the Burt matrix
- Replace diagonal blocks with values estimated from that solution
- Repeat until solution converges, improving the fit to off-diagonal blocks

MCA example: Survival on the Titanic

- Analyze the `Titanic` data, using `mjca()`
- The default inertia method is `lambda="adjusted"`
- Other methods are `"indicator", "Burt", "JCA"`

```
data(Titanic)
titanic.mca <- mjca(Titanic)
summary(titanic.mca)

##
## Principal inertias (eigenvalues):
##
##  dim    value      %   cum%   scree plot
##  1      0.067655  76.8  76.8  **********************
##  2      0.005386   6.1  82.9  **
##  3      00000000   0.0  82.9
##         -------- -----
##  Total: 0.088118
...
```

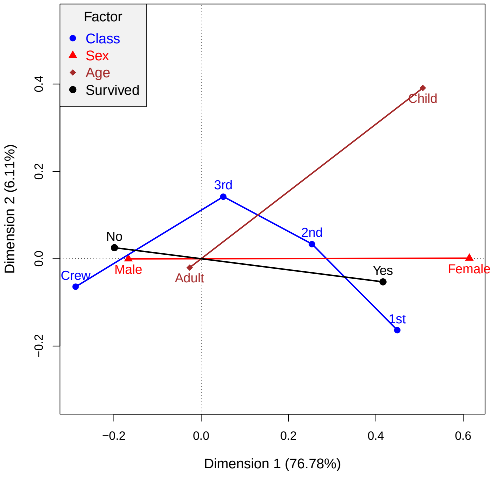
Compare adjusted inertias with other methods:

```
summary(mjca(Titanic, lambda="indicator"), columns=FALSE)

##
## Principal inertias (eigenvalues):
##
##  dim    value      %   cum%   scree plot
##  1      0.445079  29.7  29.7  *******
##  2      0.305044  20.3  50.0  *****
##  3      0.250006  16.7  66.7  ****
##  4      0.205037  13.7  80.3  ***
##  5      0.178515  11.9  92.2  ***
##  6      0.116318   7.8 100.0  **
##         -------- -----
##  Total: 1.500000 100.0
```

- Total inertia is `sum(dim(Titanic)-1)` / 4 = 6/4 = 1.5
- should only interpret dimensions with inertia $> 1/4$

MCA example: Survival on the Titanic

```
plot(titanic.mca)
```



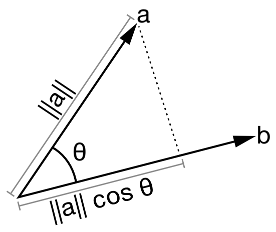
- Dim 1 is perfectly aligned with sex
- This is also strongly aligned with survival and class
- Dim 2 pertains largely to class and age effects
- $\Longrightarrow$ Survival associated with being female, upper class and child
- Using adjusted inertia, the 2D solution accounts for 83%

Biplots for contingency tables

The ***biplot*** is another visualization method that also uses the SVD to give a low-rank (2D) representation.

- In CA, the (weighted) χ^2 distances between row (column) points reflect the differences among row (column) profiles
- In the biplot, rows and columns are represented by vectors from the origin with an inner product (projection) interpretation

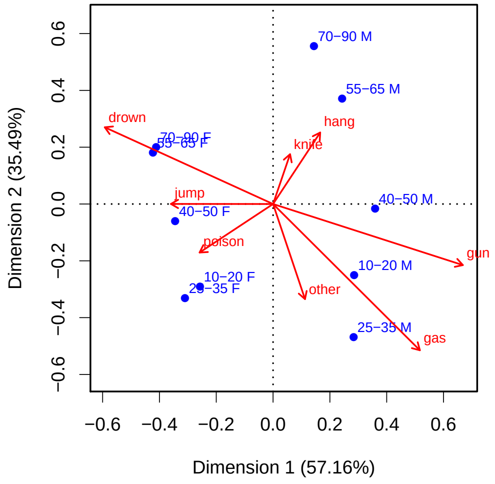
$$\boldsymbol{Y} \approx \boldsymbol{A}\boldsymbol{B}^{\mathsf{T}} \Longleftrightarrow y_{ij} \approx \boldsymbol{a}_i^{\mathsf{T}} \boldsymbol{b}_j$$



Example: suicide rates in Germany

There are a variety of different scalings for biplots. Here we use the contribution biplot

```
plot(suicide.ca, map="colgreen", arrows=c(FALSE, TRUE))
```



- Associations between age-sex categories and suicide methods can be read as projections of the points on the vectors.
- Lengths of vectors for suicide categories reflect their contributions to this 2D plot

Summary

- CA is an exploratory method designed to account for association (Pearson χ^2) in a small number of dimensions
  - Row and column scores provide an optimal scaling of the category levels
  - Plots of these can suggest an explanation for association
- CA uses the singular value decomposition to approximate the matrix of residuals from independence
- Standard and principal coordinates have different geometric properties, but are essentially re-scalings of each other
- Multi-way tables can be handled by:
  - Stacking approach— collapse some dimensions interactively to a 2-way table
  - Each way of stacking → a loglinear model
  - MCA analyzes the full $n-$ *way* table using an indicator matrix or the Burt matrix