Correspondence analysis

Michael Friendly

Psych 6136

http://friendly.github.io/psy6136

Correspondence analysis: Basic ideas

Analog of PCA for frequency data

- Account for maximum % of χ^2 in few (2-3) dimensions
- Finds scores for row (x_{im}) and col (y_{jm}) categories on these dimensions
- Uses Singular Value Decomposition of residuals from independence,

$$d_{ij} = (n_{ij} - \widehat{m}_{ij})/\sqrt{\widehat{m}_{ij}} \implies d_{ij} = \sqrt{n} \sum_{m=1}^{M} \lambda_m x_{im} y_{jm} \quad \leftrightarrow \quad D = X \Lambda Y^{\mathsf{T}}$$

- Optimal scaling: each pair of scores for rows (x_{im}) and col (y_{jm}) have highest possible correlation (= λ_m)
- Plots of the row and column scores show associations
  - Row point (x_{im}) near col point (y_{jm}) → positive association $d_{ij} > 0$

2

Correspondence analysis: History

- Mathematical foundations: "Geometric data analysis", J. P. Benzecri, ~ 1960s
  - The French school: L' Analyse des Donnes
  - Popularized in European social science
- Multidimensional EDA
  - More descriptive than inferential
  - "models should follow the data, not vice versa"
  - High-D phenomena → Low-D approximations
- CARME conferences: every 4 years
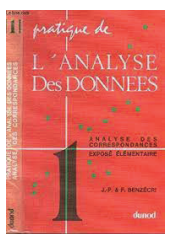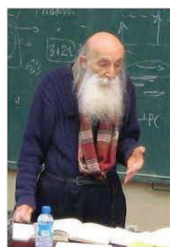
CA software for R

- ca package
  - ca() – two-way tables; plot(ca()) for graphs
  - mjca() – multiple & joint CA; vcdExtra::mcaplot() for plots
- FactoMineR & factoextra packages
  - CA() – many options for graphical displays
  - fviz_ca() – uses ggplot2; can ggrepel point labels
- ade4 package
  - dudi.coa() – very nice graphics, but somewhat quirky

4

Example: Hair color, eye color

```
> library(ca)
> haireye <- margin.table(HairEyeColor, 1:2)
> (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       0.1824  0.4831   0.1199 0.215
ChiDist    0.5512  0.1595   0.3548 0.838
Inertia    0.0554  0.0123   0.0151 0.151
Dim. 1    -1.1043 -0.3245  -0.2835 1.828
Dim. 2     1.4409 -0.2191  -2.1440 0.467

 Columns:
            Brown  Blue   Hazel   Green
Mass       0.3716 0.363  0.1571  0.1081
ChiDist    0.5005 0.554  0.2887  0.3857
Inertia    0.0931 0.111  0.0131  0.0161
Dim. 1    -1.0771 1.198 -0.4653  0.3540
Dim. 2     0.5924 0.556 -1.1228 -2.2741
```
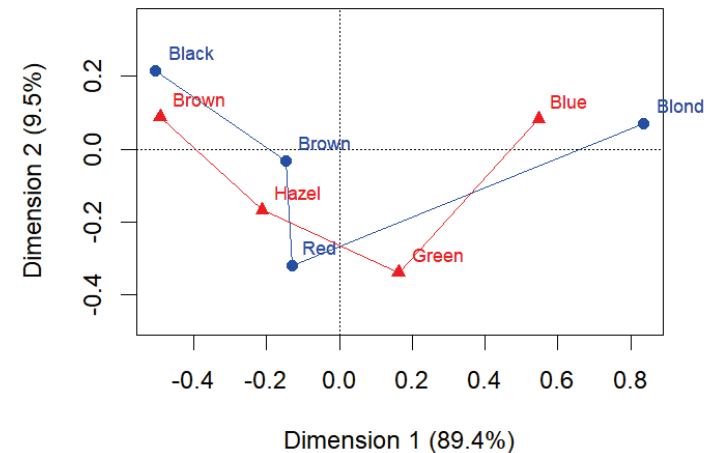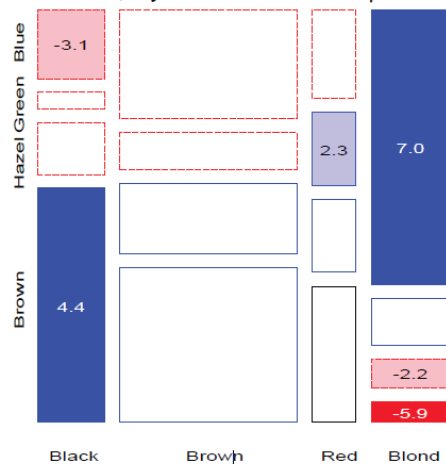
χ^2 % for dimensions

Hair category scores, Dim1-2

Eye category scores, Dim1-2

```
plot(haireye.ca, lines=TRUE)
```


- Rough interpretation: row/col points "near" each other are positively associated (independence residuals $d_{ij} \gg 0$)
- 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 & column profiles

- For a two-way table, row profiles & column profiles give relative proportions of the 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 profiles pertain to the differences among brand preference
- Column profiles pertain to the differences among regions

```
 Region                              Region
Brand        R1    R2    R3 Sum     Brand           R1     R2     R3
  Brand A 12.5 12.5 75.0 100          Brand A  12.5   12.5   75.0
  Brand B 14.3 71.4 14.3 100          Brand B  12.5   62.5   12.5
  Brand C 60.0 20.0 20.0 100          Brand C  37.5   12.5   12.5
  Brand D 75.0 25.0  0.0 100          Brand D  37.5   12.5    0.0
                                      Sum      100.0  100.0  100.0
```

There is clearly an association: → the row (& column) profiles differ

```
> chisq.test(toothpaste)

        Pearson's Chi-squared test

data:  toothpaste
X-squared = 79.6, df = 6, p-value = 4.3e-15
```
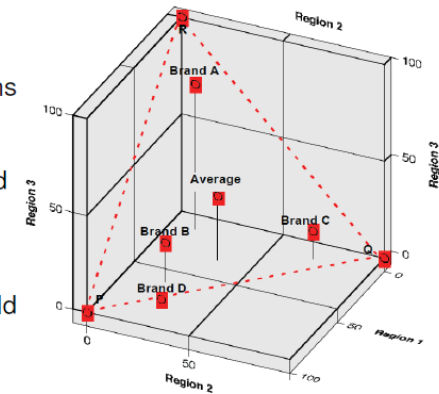
Plotting profiles

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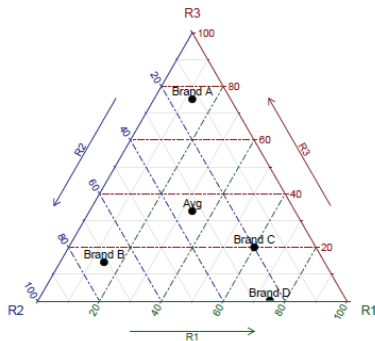
- The Avg profile has coordinates of 33.3% for each region
- Brand preferences by region can be seen by their positions wrt the R1–R3 axes
- This suggests that differences among brands can be measured by their (squared) distances from the centroid, weighted by their row margins (mass)
- Physical analogy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

CA solution

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

```
> 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
```
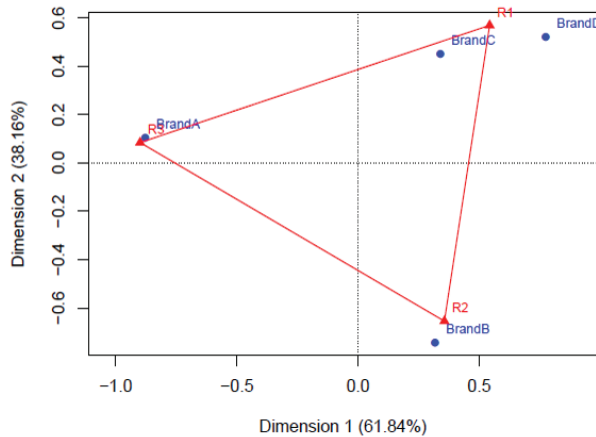
Pearson $\chi^2 = N \sum \lambda^2$

```
> # reproduce chi-square
> sum(tp.ca$sv^2) * sum(toothpaste)
[1] 79.607
```

CA solution

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


Brand A: most in R3
Brand B: most in R2
Brands C, D: most in R1

Profiles & inertia

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.
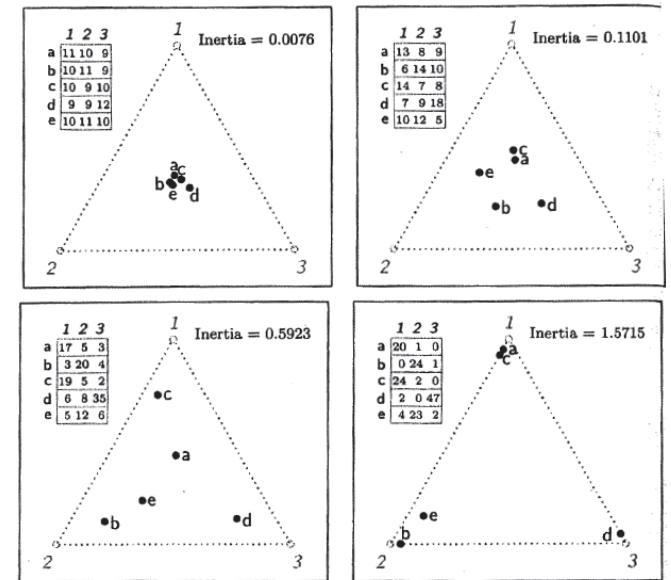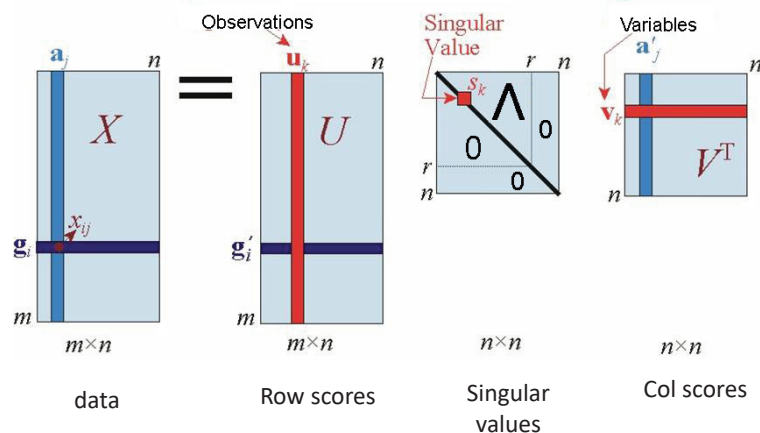
Singular value decomposition

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

$$\mathbf{X} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{V}^T$$

| data | Row scores | Singular values | Col scores |

Properties of the SVD

- **U**: columns are eigenvectors of $\mathbf{X}\mathbf{X}^T$ and form an orthonormal basis for observation profiles such that $\mathbf{U}^T\mathbf{U} = \mathbf{I}$
- **Λ**: diagonal, r singular values = sqrt eigenvalues of both $\mathbf{X}\mathbf{X}^T$ and $\mathbf{X}^T\mathbf{X}$
- **V**: columns are eigenvectors of $\mathbf{X}^T\mathbf{X}$, orthonormal: $\mathbf{V}^T\mathbf{V} = \mathbf{I}$

$$\mathbf{X} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{V}^T$$
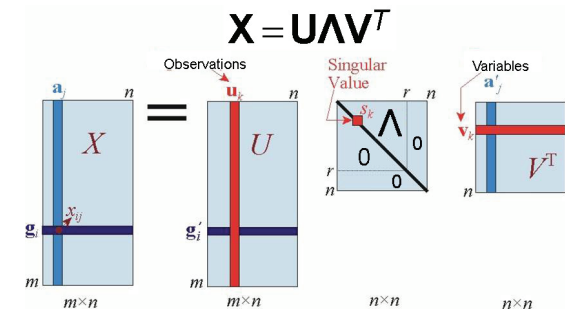
SVD: Matrix approximation

- Let $\mathbf{X}$ be an $m \times n$ matrix such that rank($\mathbf{X}$) = r
- If $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_r$ are the singular values of $\mathbf{X}$, then $\hat{\mathbf{X}}$, the rank q approximation of $\mathbf{X}$ that minimizes $\| \mathbf{X} - \hat{\mathbf{X}} \|$, is
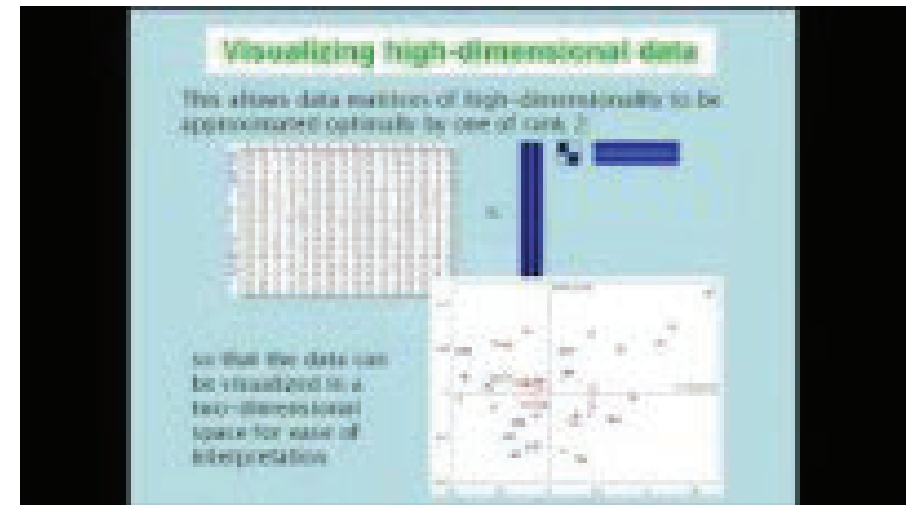
$$\hat{\mathbf{X}}_{m \times n} = \sum_{i=1}^{q} \lambda_i \begin{pmatrix} u_{i1} \\ \vdots \\ u_{im} \end{pmatrix} \begin{pmatrix} v_{i1} & \cdots & v_{in} \end{pmatrix} = \lambda_1 u_1 v_1^T + \cdots + \lambda_q u_q v_q^T$$

column scores (pointing to $v_{i1} \cdots v_{in}$)
row scores (pointing to $u_{i1} \ldots u_{im}$)

a sum of q rank=1 (outer) products. The variance in $\mathbf{X}$ accounted for each term is λ_1^2

SVD song: It had to be U ...

Michael Greenacre, *It had to be U - the SVD song*, https://www.youtube.com/watch?v=JEYLfIVvR9I

CA notation & terminology

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

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

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

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

Principal & standard coordinates

Two types of coordinates are used in CA, based on re-scalings of A and B. Principal coordinates are most commonly used in plotting CA solutions.

Principal coordinates

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

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

- 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, $\mathbf{F}$ and $\mathbf{G}$, is called the symmetric map because both row and column profiles are overlaid in the same coordinate system.

Standard coordinates

Standard coordinates

The standard coordinates (Φ, Γ) are a rescaling of the principal coordinates to unit inertia along each axis,

$$\Phi = D_r^{-1} A \quad \text{scaled so that} \quad \Phi^T D_r \Phi = I$$
$$\Gamma = D_c^{-1} B \quad \text{scaled so that} \quad \Gamma^T D_c \Gamma = I$$

- 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 & 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 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, distances between two row profiles, r_i and $r_{i'}$ are χ^2 distances
  - The squared difference $(r_{ij} - r_{i'j})^2$ between two row profiles is inversely weighted by the column frequency, to account for the different relative frequency of the column categories.
- **Plotting**: For distances to be interpretable, it's crucial to scale the axes equally, so 1^{cm} is the same on both axes (aspect ratio = 1). This is standard in most packages.

The `ca` package in R

`ca()` calculates CA solutions, returning a "ca" object with all the details

```
> names(haireye.ca)
 [1] "sv"        "nd"        "rownames"  "rowmass"   "rowdist"
 [6] "rowinertia" "rowcoord"  "rowsup"    "colnames"  "colmass"
[11] "coldist"    "colinertia" "colcoord" "colsup"    "N"
[16] "call"
```

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

```
> haireye.ca$rowcoord
       Dim1    Dim2    Dim3
Black -1.104  1.441 -1.089
Brown -0.324 -0.219  0.957
Red   -0.283 -2.144 -1.631
Blond  1.828  0.467 -0.318
```

```
> haireye.ca$colcoord
        Dim1    Dim2    Dim3
Brown -1.077  0.592 -0.4240
Blue   1.198  0.556  0.0924
Hazel -0.465 -1.123  1.9719
Green  0.354 -2.274 -1.7184
```

`ca` plots

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

- "symmetric" – both rows & cols in principal coordinates (default)
- "rowprincipal" or "colprincipal" – asymmetric maps with rows in principal coordinates and cols in std coordinates, or vice versa
- "symbiplot" – scales both rows and cols to have variances equal to the singular value

The `mjca()` function is used for multiple correspondence analysis (MCA) for 3+ way tables. Has analogous print(), summary() and plot() methods
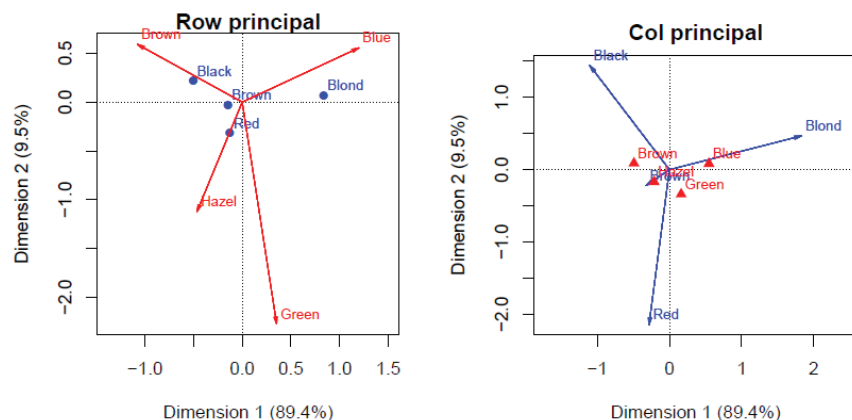- `vcdExtra::mcaplot()` does a nicer job of plotting MCA solutions

Slide 25

Asymmetric row/col principal plots are biplots – can interpret the projection of points on vectors for the other variable

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

Slide 26

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, **X**1 and **Y**1 to the row/column categories: $\rightarrow$ Max. possible correlation, λ_1
  - Assign Dim 2 scores, **X**2 and **Y**2 to the row/column categories: $\rightarrow$ Max. possible correlation, λ_2, but uncorrelated with **X**1, **Y**1
  - All association between row/col categories is captured by the scores
- This optimal scaling interpretation can be used to quantify categorical variables, particularly if Dim 1 is large
- Mosaics: Permute rows / cols by Dim 1 scores

Slide 27

Optimal category scores

```
> haireye.ca <- ca(haireye)
> round(haireye.ca$sv, 3)
[1] 0.457 0.149 0.051
```

The singular values λ_i = canonical correlations

To demonstrate category scores, extract row/col coordinates to a data frame

```
HE.df <- as.data.frame(haireye)

RC <- haireye.ca$rowcoord  # row coordinates
CC <- haireye.ca$colcoord  # col coordinates

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

HE.df <- cbind(HE.df, X1, Y1, X2, Y2)
```

Slide 28

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
. . .
13  Black Green    5   0.354  -1.104  -2.274   1.441
14  Brown Green   29   0.354  -0.324  -2.274  -0.219
15    Red Green   14   0.354  -0.283  -2.274  -2.144
16  Blond Green   16   0.354   1.828  -2.274   0.467
```

Calculate Freq-weighted correlations. All are zero except r(X1, Y1) = λ_1 & r(X2, Y2) = λ_2

```
> corr <- cov.wt(HE.df[,4:7], wt=HE.df$Freq, cor=TRUE)$cor
> round(zapsmall(corr), 3)
      X1    Y1    X2    Y2
X1 1.000 0.457 0.000 0.000
Y1 0.457 1.000 0.000 0.000
X2 0.000 0.000 1.000 0.149
Y2 0.000 0.000 0.149 1.000
```

Permuting for a mosaic

```
data(housetasks, package="factoextra")
res.ca <- FactoMiner::CA(housetasks, graph=FALSE)
fviz_ca(res.ca, repel = TRUE,
        geom.col = c("point", "text", "arrow"))  +
    theme_minimal()
```
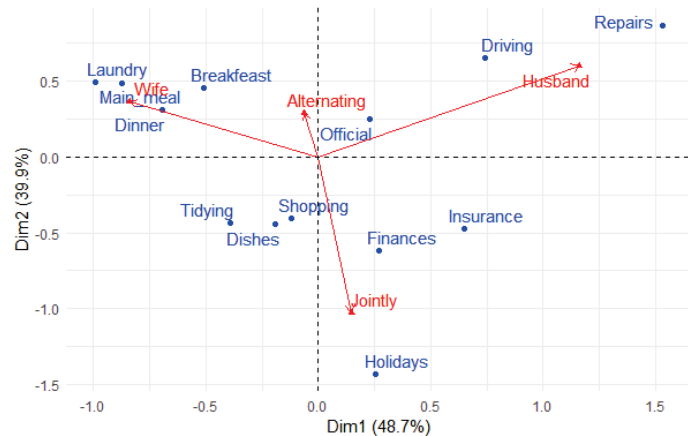

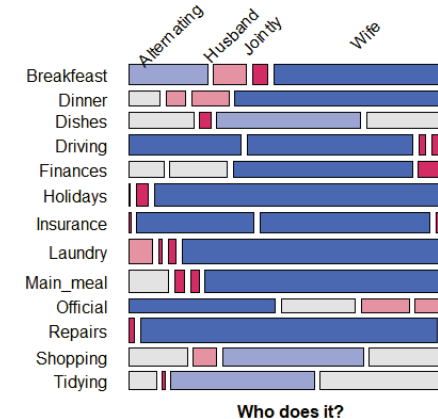Dim1: H vs Wife

Dim2: single vs jointly
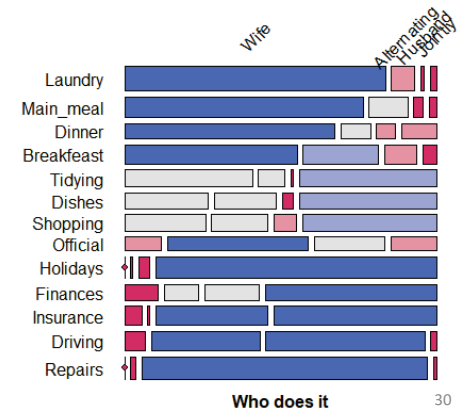
Permuting for a mosaic

The seriate package has a CA method to permute rows/cols of a df or matrix

```
library(seriation)
order <- seriate(housetasks, method = "CA")
ht <- permute(housetasks, order, margin=1)
mosaic(ht, shade = TRUE, ...)
```
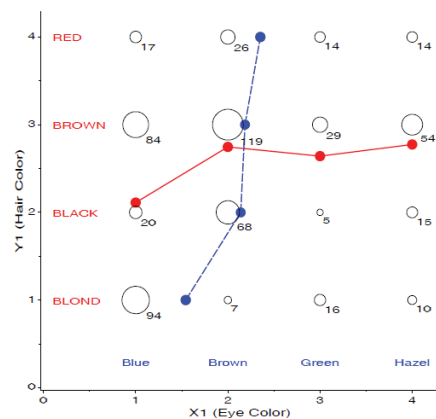
Alpha ordered

CA ordered

Who does it?

Who does it

Simultaneous linear regression

Assign linear 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

$Y1 = 0 + 0.457 \; X1$
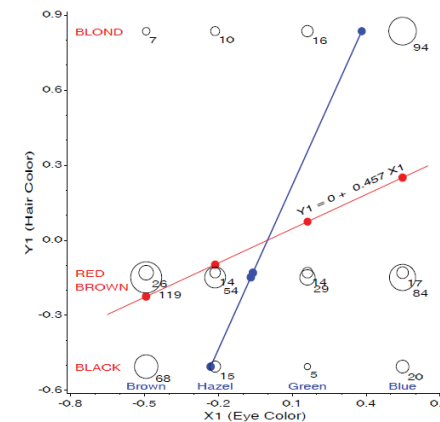
- The regression of Y1 on X1 is linear, with slope λ_1
- The regression of X1 on Y1 is linear, with slope $1/\lambda_1$
- λ_1 is the (canonical) correlation between X1 and Y1
- The angle between the two lines would be 0 if perfect correlation
- The conditional means (dots) are the principal coordinates

Example: Mental impairment & parent' SES

Data on mental health status of 1660 young NYC residents, by parents' SES, a 6 x 4 table. Is higher SES associated with better kids' mental health?

```
> 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 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 ...
```

Both ses and mental are ordered factors in a frequency data frame
- For ca(), convert this to a table using xtabs()

```
> (mental.tab <- xtabs(Freq ~ ses + mental, data=Mental))
    mental
ses Well Mild Moderate Impaired
  1   64   94       58       46
  2   57   94       54       40
  3   57  105       65       60
  4   72  141       77       94
  5   36   97       54       78
  6   21   71       54       71
```

Mental data: CA solution

```
> mental.ca <- ca(mental.tab)
> summary(mental.ca, rows=FALSE, columns=FALSE)

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
```
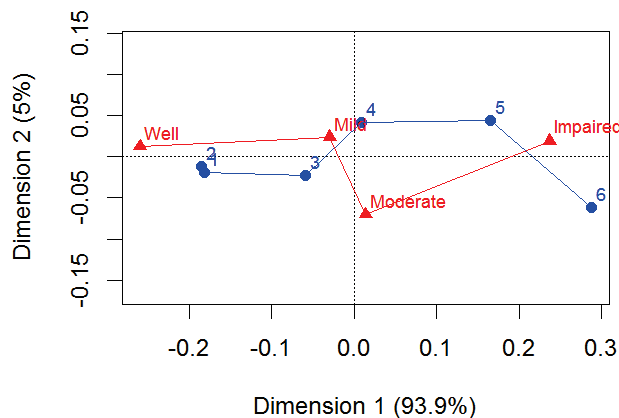
- The exact CA solution requires min(r-1, c-1) = 3 dimensions
- Total Pearson χ^2 is $n\Sigma \lambda_i^2$ = 1660 x 0.0277 = 45.98 with 15 df
- Of this, 93.9% is accounted for by the 1st dimension

Mental data: CA plot

```
plot(mental.ca, lines = TRUE)
```


Category spacing:

SES: perhaps collapse categories (1,2) ??

Mental: Smaller diff betw. Mild, Moderate ??

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 a hypothesis that the row/column scores are have some particular spacing (e.g., are `mental` and `ses` equally spaced?)
- These 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

A 3-way table of size $I \times J \times K$ can be sliced and stacked as a two-way table in several ways
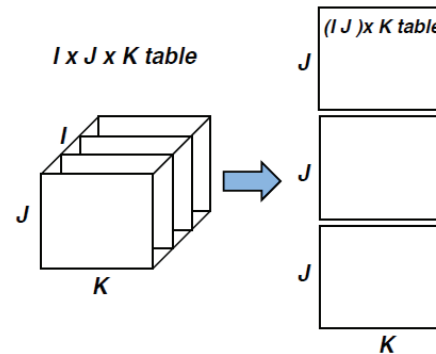
- The variables combined are treated "interactively"
- Each way of stacking corresponds to a loglinear model
  - $(I \times J) \times K \rightarrow [AB][C]$
  - $I \times (J \times K) \rightarrow [A][BC]$
  - $J \times (I \times K) \rightarrow [B][AC]$
- Only the associations in separate [ ] terms are analyzed and displayed
- The stacked table is analyzed with ordinary CA of the two-way stacked table

---

Interactive coding in R

- Data in table or array form: use
  `as.matrix(structable(rows ~ cols))`

```
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 as frequency data frame: use **interaction()** or `paste()` 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

- **vcd::Suicide** gives a 2 x 5 x 8 table of `sex` by `age.group` by `method` for 53,158 suicides in Germany, in a frequency data frame
- Use `paste()` to join `age.group` and `sex` → `age_sex` in the form '10-20 M'

```
> Suicide <- within(Suicide, {
    age_sex <- paste(age.group, toupper(substr(sex,1,1)))
  })
> head(Suicide)
  Freq  sex    method age age.group method2 age_sex
1    4 male    poison  10     10-20  poison 10-20 M
2    0 male   cookgas  10     10-20     gas 10-20 M
3    0 male  toxicgas  10     10-20     gas 10-20 M
4  247 male      hang  10     10-20    hang 10-20 M
5    1 male     drown  10     10-20   drown 10-20 M
6   17 male       gun  10     10-20     gun 10-20 M
```

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

Suicide rates in Germany

```
> suicide.ca <- ca(suicide.tab)
> summary(suicide.ca, rows=FALSE, columns = FALSE)

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
```
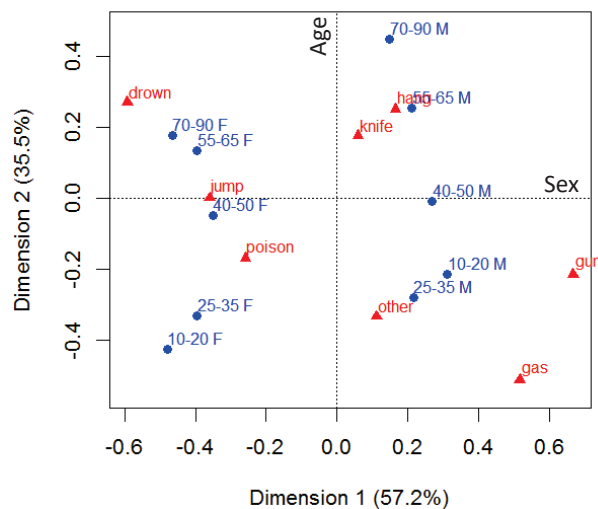
For this table χ^2 (63) = 8946. Of this, 92.6% is accounted for in the first two dimensions
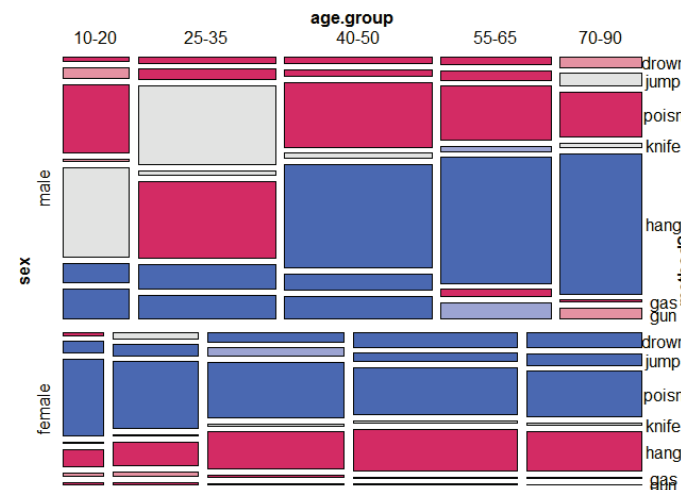
```
> plot(suicide.ca)
```


- Dim 1: Sex
- Dim 2: Age
- Can interpret method use by age-sex combination
- young M: gas, gun,
- young F: poison

Compare with a mosaic plot, also fitting the model [Age Sex][Method]

```
suicide.tab3 <- xtabs(Freq ~ sex + age.group + method2, data=Suicide)
mosaic(suicide.tab3, shade=TRUE, legend=FALSE,
       expected=~age.group*sex + method2, ... )
```


DDAR Fig 6.7, p 238

(I permuted methods by CA Dim1 & deleted "Other")

Marginal tables & supplementary variables

- **Supplementary variables** provide a way to include more info in CA
  - 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
- E.g., age by method, ignoring sex as the main analysis

```
> 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
```

Also have data on 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)
```

```
          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
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
```

Main analysis table

Supplementary rows

Supplementary variables

Call ca(table, suprow =) to treat some rows as supplementary variables

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

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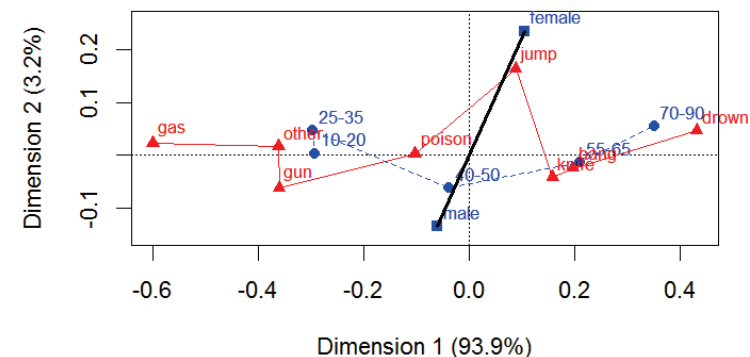
The relation of age and method is now essentially 1 dimensional
The inertia of Dim 1 here (0.604) is nearly the same as that of Dim 2 (0.596) for age in the stacked table

```
res <- plot(suicide.ca2s,
            pch=c(16, 15, 17, 24),
            lines = c(FALSE, TRUE))
lines(res$rows[1:5,], col = "blue", lty=2)
lines(res$rows[6:7,], col = "black", lwd=3)
```

Plotting the solution shows points for row, col & supplementary rows

Ignoring Sex has collapsed Sim 1 (Sex) of the [Age Sex][Method] analysis
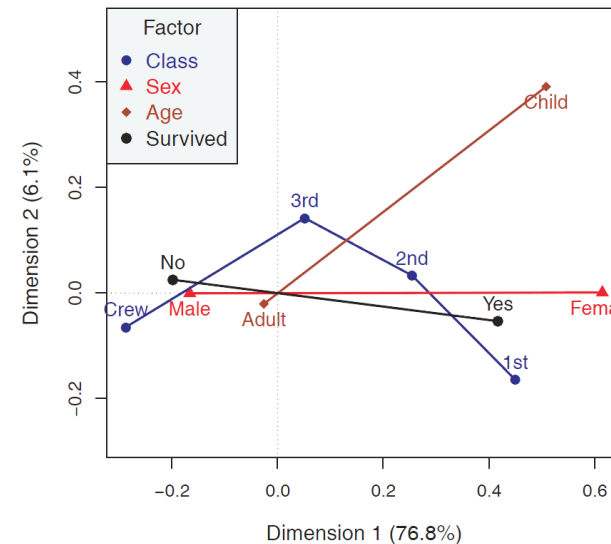Supp. points for Sex show the association of Method with Sex in this space

Multiple correspondence analysis

- 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

Example: Titanic data

Plot of MCA for the Titanic data

All 4 variables represented in a single plot

Dim 1: Sex
Dim 2: Class & Age

Distance from origin = inertia ~ 1/category freq

CA → MCA: Indicator & Burt

Two ways to think about MCA:

Indicator matrix (dummy variables)

- A given categorical variable, q, can be represented by an indicator matrix $Z(n \times J_q)$ of dummy variables, $z_{ij} = 1$ if case i is in category j
- Let $Z_1, Z_2, \ldots, Z_Q$ be the indicator matrices for Q variables
- MCA is then a simple CA applied to the partitioned matrix $Z = [Z_1, Z_2, \ldots, Z_Q]$

Burt matrix

- The Bert matrix is the product of the indicator matrix Z and its transpose

$$B = Z^T Z$$

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

Indicator matrix: Hair Eye color

- For the hair-eye data, the indicator matrix Z has n=592 rows (observations) and 4 + 4 = 8 columns (categories).
  - Shown below in frequency form: h1 — h4 indicators for hair color, e1—e4 for eye color
  - E.g., 1st row represents 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 to get **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 $\mathbf{Z} = [\mathbf{Z}_1; \mathbf{Z}_2]$, corresponding to the hair, eye categories, then the contingency table is given by $\mathbf{N} = \mathbf{Z}^\mathsf{T}_1 \mathbf{Z}_2$.
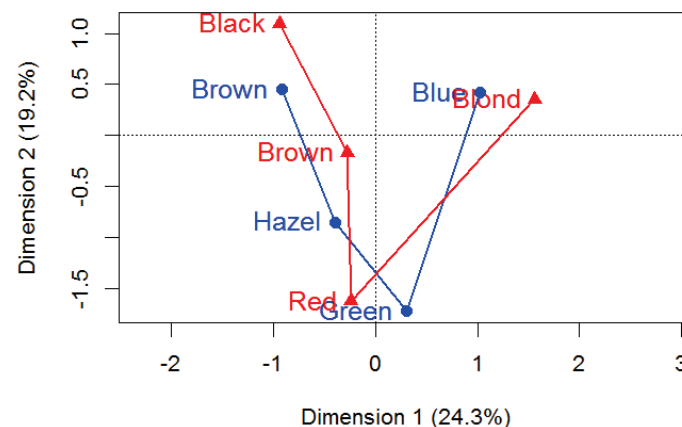
```
> Z1 <- as.matrix(Z[,1:4])
> Z2 <- as.matrix(Z[,5:8])
> (N <- t(Z1) %*% Z2)
      Brown Blue Hazel Green
Black    68   20    15     5
Brown   119   84    54    29
Red      26   17    14    14
Blond     7   94    10    16
```

- 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")) # plus customization
```

The Burt matrix

For two categorical variables, the Burt matrix is

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

- $\mathbf{N}_1$ and $\mathbf{N}_2$ 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 **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 $(Z_q) = J_q - 1$
- 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 $2 \times 2 \times 2 \times 2$ 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, rows=FALSE, columns = FALSE)
```

```
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

```
vcdExtra::mcaplot(presex.mca,
               legend=TRUE, legend.pos = "bottomright")
```
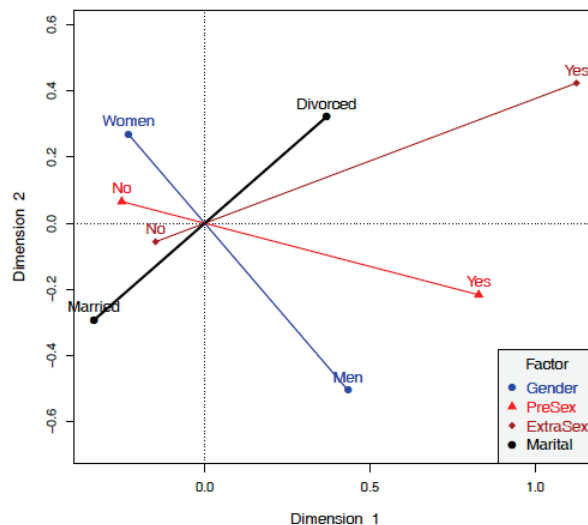
Accounts for 76% of total inertia

Women less likely to report pre- and/or extra-marital sex

Divorced associated with pre- and extra- sex

Gender $\perp$ Marital

NB: This only analyzes bivariate associations, i.e., no 3-way associations

Inertia in MCA

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

Not so straight-forward in MCA:

- For a given indicator matrix, Z_q, the inertia is $J_q - 1$
- For all variables, with $J = \Sigma J_q$ categories, the total inertia of $Z = [Z_1, \ldots, Z_Q]$ is the average of the inertias of the sub-tables

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

- The average inertia per dimension is therefore $1/Q$
- $\implies$ 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: Details

Two solutions:

Adjusted inertia

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

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

- 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

NB: JCA solutions aren't nested. I generally use adjusted inertia

---

MCA example: Survival on the *Titanic*

Analyse the Titanic data using `ca::mcja()`
- The default inertia method is lambda = "adjusted"
- Other methods: "indicator", "Burt", "JCA"

```
data(Titanic)
titanic.mca <- mjca(Titanic)
summary(titanic.mca, columns = FALSE)
```
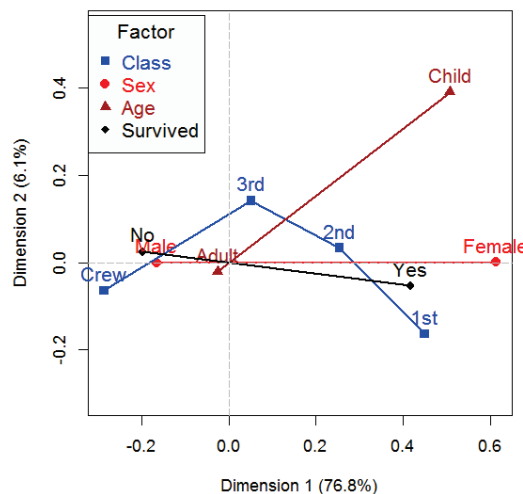
```
Principal inertias (eigenvalues):

 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
```

Using adjusted inertia, the 2D solution accounts for ~ 83% of total, bivariate association.

Plot the solution with `vcdExtra::mcaplot()`

```
mcaplot(titanic.mca, legend=TRUE, legend.pos = "topleft")
```


Dim 1 perfectly aligned with Sex
Also strongly aligned w/ survival & class

Dim 2: reflects class & age

→ Survival associated with Female, 1st vs 3rd class, child

Biplots for contingency tables

The biplot is a related visualization that also uses the SVD to give a low-rank (2D) approximation.
- In CA, the weighted χ^2 distances between row (column) points reflect the differences among row (column) profiles
- In the biplot, rows (columns) are represented by vectors from the origin, with an inner-product (projection) interpretation – row point $\mathbf{a}_i$ is fit by projection on col point $\mathbf{b}_j$
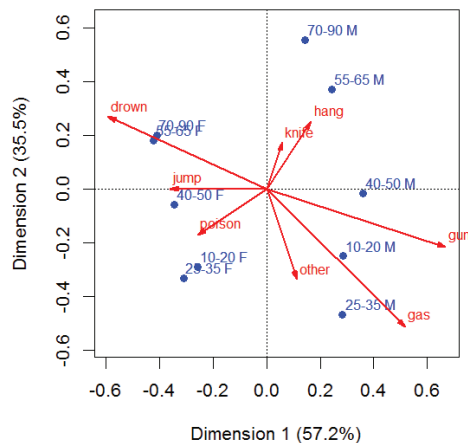
$$\mathbf{Y} \approx \mathbf{AB}^{\mathsf{T}} \iff y_{ij} \approx \mathbf{a}_i^{\mathsf{T}} \mathbf{b}_j$$
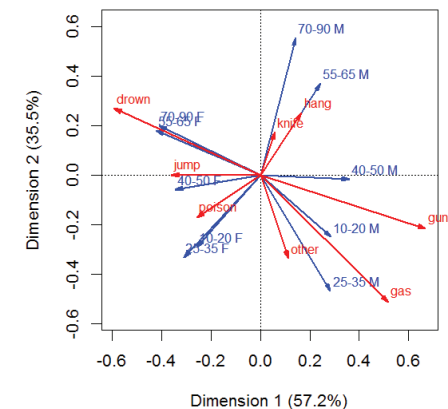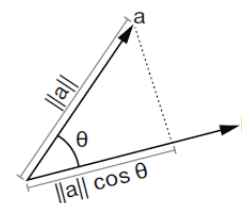
Example: Suicide rates

There are different scalings for CA biplots. Here I use the 'contribution' biplot
I find the plot less messy to plot arrows for only rows or cols and imagine the projection

```
plot(suicide.ca, map="colgreen", arrows=c(FALSE, TRUE), lwd=2)
```


Associations between age-sex categories and suicide methods can be read as projections of the points on the vectors

Lengths of vectors for suicide 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 $\rightarrow$ a loglinear model
  - MCA analyzes the full $n - way$ table using an indicator matrix or the Burt matrix

Given a new 2-way table, my first thought is nearly always: `plot(ca(mytable))`