Summary: PCA

PCA

Find an orthogonal basis \( \{v_1, v_2, \ldots, v_p\} \) for the data space such that:

- The first principal component (PC) \( v_1 \) is the **direction of greatest variance** of data.
- The \( j \)-th PC \( v_j \) is the **direction orthogonal to** \( v_1, v_2, \ldots, v_{j-1} \) of greatest variance, for \( j = 2, \ldots, p \).

- **Eigendecomposition of the sample covariance matrix**
  \[
  S = \frac{1}{n-1} \sum_{i=1}^{n} x_i x_i^\top.
  \]
  \[
  S = V \Lambda V^\top.
  \]
  - \( \Lambda \) is a diagonal matrix with eigenvalues (variances along each principal component) \( \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0 \)
  - \( V \) is a \( p \times p \) orthogonal matrix whose columns are the \( p \) eigenvectors of \( S \), i.e. the principal components \( v_1, \ldots, v_p \)
- **Dimensionality reduction by projecting** \( x_i \in \mathbb{R}^p \) onto first \( k \) principal components:
  \[
  z_i = [v_1^\top x_i, \ldots, v_k^\top x_i]^\top \in \mathbb{R}^k.
  \]
Summary: PCA

\[ S = \frac{1}{n-1} \sum_{i=1}^{n} x_i x_i^\top = \frac{1}{n-1} X^\top X. \]

- \( S \) is a **real and symmetric** matrix, so there exist \( p \) eigenvectors \( v_1, \ldots, v_p \) that are pairwise orthogonal and \( p \) associated eigenvalues \( \lambda_1, \ldots, \lambda_p \) which satisfy the eigenvalue equation \( S v_i = \lambda_i v_i \). In particular, \( V \) is an orthogonal matrix:
  \[ V V^\top = V^\top V = I_p. \]

- \( S \) is a **positive-semidefinite** matrix, so the eigenvalues are non-negative:
  \[ \lambda_i \geq 0, \ \forall i. \]

Why is \( S \) symmetric? Why is \( S \) positive-semidefinite?

**Reminder:** A symmetric \( p \times p \) matrix \( R \) is said to be positive-semidefinite if
\[ \forall a \in \mathbb{R}^p, a^\top R a \geq 0. \]
Singular Value Decomposition (SVD)

**SVD**

Any real-valued $n \times p$ matrix $X$ can be written as $X = UDV^\top$ where

- $U$ is an $n \times n$ orthogonal matrix: $UU^\top = U^\top U = I_n$
- $D$ is a $n \times p$ matrix with decreasing **non-negative** elements on the diagonal (the singular values) and zero off-diagonal elements.
- $V$ is a $p \times p$ orthogonal matrix: $VV^\top = V^\top V = I_p$

- SVD **always** exists, even for non-square matrices.
- Fast and numerically stable algorithms for SVD are available in most packages. The relevant R command is `svd`. 
SVD and PCA

- Let \( X = UDV^\top \) be the SVD of the \( n \times p \) data matrix \( X \).
- Note that

\[
(n - 1)S = X^\top X = (UDV^\top)^\top (UDV^\top) = VD^\top U^\top UDV^\top = VD^\top DV^\top,
\]

using orthogonality \( U^\top U = I_n \) of \( U \).
- The eigenvalues of \( S \) are thus the diagonal entries of \( \Lambda = \frac{1}{n-1} D^\top D \).
- We also have (using orthogonality \( V^\top V = I_p \))

\[
XX^\top = (UDV^\top)(UDV^\top)^\top = UDV^\top VD^\top U^\top = UD \mathbf{D} D^\top U^\top,
\]

Gram matrix

\( B = XX^\top \), \( B_{ij} = x_i^\top x_j \) is called the Gram matrix of dataset \( X \).
\( B \) and \( (n - 1)S = X^\top X \) have the same nonzero eigenvalues, equal to the non-zero squared singular values of \( X \).

Projection:

\[
Z = XV = UDV^\top V = UD.
\]

Can be obtain by eigendecomposition of \( B \), less computation if \( p > n \).
Biplots

PCA plots show the data items (rows of $X$) in the space spanned by PCs.

Biplots allow us to visualize the original variables $X^{(1)}, \ldots, X^{(p)}$ (corresponding to columns of $X$) in the same plot.
Recall that $X = [X^{(1)}, \ldots, X^{(p)}]^{\top}$ and $X = UDV^{\top}$ is the SVD of the data matrix.

- The 'full' PC projection of $x_i$ is the $i$-th row of $UD$:
  $$z_i = V^{\top} x_i = D^{\top} U_i^{\top},$$ 
equivalently: $XV = UD$.

- The $j$-th unit vector $e_j \in \mathbb{R}^p$ points in the direction of the original variable $X^{(j)}$. Its PC projection $\eta_j$ is:
  $$\eta_j = V^{\top} e_j = V_j^{\top} \quad \text{(the } j\text{-th row of } V)$$

- The projection of $e_j$ indicates the weighting each PC gives to the original variable $X^{(j)}$.
- Dot products between these projections give entries of the data matrix:
  $$x_{ij} = \sum_{k=1}^{\min\{n,p\}} U_{ik} D_{kk} V_{jk} = \langle D^{\top} U_i^{\top}, V_j^{\top} \rangle = \langle z_i, \eta_j \rangle.$$

- Biplots focus on the first two PCs and the quality depends on the
  **proportion of variance explained** by the first two PCs.
Iris Data

50 samples from each of the 3 species of iris: *setosa*, *versicolor*, and *virginica*

Each measuring the length and widths of both sepal and petals

Collected by E. Anderson (1935) and analysed by R.A. Fisher (1936)
Iris Data

```
> data(iris)
> iris[sample(150,20),]

<table>
<thead>
<tr>
<th>Sepal.Length</th>
<th>Sepal.Width</th>
<th>Petal.Length</th>
<th>Petal.Width</th>
<th>Species</th>
</tr>
</thead>
<tbody>
<tr>
<td>54</td>
<td>5.5</td>
<td>2.3</td>
<td>4.0</td>
<td>1.3 versicolor</td>
</tr>
<tr>
<td>33</td>
<td>5.2</td>
<td>4.1</td>
<td>1.5</td>
<td>0.1 setosa</td>
</tr>
<tr>
<td>30</td>
<td>4.7</td>
<td>3.2</td>
<td>1.6</td>
<td>0.2 setosa</td>
</tr>
<tr>
<td>73</td>
<td>6.3</td>
<td>2.5</td>
<td>4.9</td>
<td>1.5 versicolor</td>
</tr>
<tr>
<td>107</td>
<td>4.9</td>
<td>2.5</td>
<td>4.5</td>
<td>1.7 virginica</td>
</tr>
<tr>
<td>4</td>
<td>4.6</td>
<td>3.1</td>
<td>1.5</td>
<td>0.2 setosa</td>
</tr>
<tr>
<td>90</td>
<td>5.5</td>
<td>2.5</td>
<td>4.0</td>
<td>1.3 versicolor</td>
</tr>
<tr>
<td>83</td>
<td>5.8</td>
<td>2.7</td>
<td>3.9</td>
<td>1.2 versicolor</td>
</tr>
<tr>
<td>50</td>
<td>5.0</td>
<td>3.3</td>
<td>1.4</td>
<td>0.2 setosa</td>
</tr>
<tr>
<td>92</td>
<td>6.1</td>
<td>3.0</td>
<td>4.6</td>
<td>1.4 versicolor</td>
</tr>
<tr>
<td>128</td>
<td>6.1</td>
<td>3.0</td>
<td>4.9</td>
<td>1.8 virginica</td>
</tr>
<tr>
<td>57</td>
<td>6.3</td>
<td>3.3</td>
<td>4.7</td>
<td>1.6 versicolor</td>
</tr>
<tr>
<td>9</td>
<td>4.4</td>
<td>2.9</td>
<td>1.4</td>
<td>0.2 setosa</td>
</tr>
<tr>
<td>2</td>
<td>4.9</td>
<td>3.0</td>
<td>1.4</td>
<td>0.2 setosa</td>
</tr>
<tr>
<td>86</td>
<td>6.0</td>
<td>3.4</td>
<td>4.5</td>
<td>1.6 versicolor</td>
</tr>
<tr>
<td>66</td>
<td>6.7</td>
<td>3.1</td>
<td>4.4</td>
<td>1.4 versicolor</td>
</tr>
<tr>
<td>85</td>
<td>5.4</td>
<td>3.0</td>
<td>4.5</td>
<td>1.5 versicolor</td>
</tr>
<tr>
<td>147</td>
<td>6.3</td>
<td>2.5</td>
<td>5.0</td>
<td>1.9 virginica</td>
</tr>
<tr>
<td>8</td>
<td>5.0</td>
<td>3.4</td>
<td>1.5</td>
<td>0.2 setosa</td>
</tr>
<tr>
<td>41</td>
<td>5.0</td>
<td>3.5</td>
<td>1.3</td>
<td>0.3 setosa</td>
</tr>
</tbody>
</table>
```
Iris data biplot

```r
> iris.pca <- princomp(iris[, -5], cor=TRUE)
> biplot(iris.pca, scale=0)
```
Biplots

- There are other projections we can consider for biplots (assuming $p < n$ to simplify notation):

$$x_{ij} = \sum_{k=1}^{p} U_{ik} D_{kk} V_{jk} = \langle D_{1:p,1:p}^\top U_{i,1:p}, V_{j}^\top \rangle = \langle D_{1:p,1:p}^{1-\alpha} U_{i,1:p}, D_{1:p,1:p}^{\alpha} V_{j}^\top \rangle.$$

where $0 \leq \alpha \leq 1$, i.e., we change representation to

$$\tilde{z}_i = D_{1:p,1:p}^{1-\alpha} U_{i,1:p}, \quad \tilde{\eta}_j = D_{1:p,1:p}^{\alpha} V_{j}^\top$$

- **case $\alpha = 1$:**
  - Sample covariance of the projected points is:

$$\hat{\text{Cov}}(\tilde{Z}) = \frac{1}{n-1} U_{1:n,1:p}^\top U_{1:n,1:p} = \frac{1}{n-1} I_p.$$

Projected points are uncorrelated and dimensions are equi-variance.

- Sample covariance between $X^{(i)}$ and $X^{(j)}$ is:

$$\hat{E}(X^{(i)} X^{(j)}) = \frac{1}{n-1} \left( V D^\top DV^\top \right)_{i,j} = \frac{1}{n-1} \langle D_{1:p,1:p} V_{i}^\top, D_{1:p,1:p} V_{j}^\top \rangle$$

The angle between the projected variables maps to their correlation.
Iris Data biplot - scaled

> ?biplot
...
scale: The variables are scaled by lambda ^ scale and the observations are scaled by lambda ^ (1-scale) where lambda are the singular values as computed by princomp. (default=1)
...
> biplot(iris.pca, scale=1)
Crabs Data biplots

> biplot(Crabs.pca, scale=0)

> biplot(Crabs.pca, scale=1)
US Arrests Data

This data set contains statistics, in arrests per 100,000 residents for assault, murder, and rape in each of the 50 US states in 1973. Also given is the percent of the population living in urban areas.

```r
pairs(USArrests)
usarrests.pca <- princomp(USArrests, cor=T)
plot(usarrests.pca)
biplot(usarrests.pca)
pairs(predict(usarrests.pca))
```
US Arrests Data Pairs Plot

> pairs(USArrests)
US Arrests Data Biplot

> biplot(usarrests.pca, scale=1)
Suppose there are $n$ points $X$ in $\mathbb{R}^p$, but we are only given the $n \times n$ matrix $D$ of inter-point distances.

Can we reconstruct $X$?
Multidimensional Scaling

Rigid transformations (translations, rotations and reflections) do not change inter-point distances so cannot recover $X$ exactly. However $X$ can be recovered up to these transformations!

- Let $d_{ij} = \|x_i - x_j\|_2$ be the distance between points $x_i$ and $x_j$.

$$d_{ij}^2 = \|x_i - x_j\|_2^2 = (x_i - x_j)^\top (x_i - x_j) = x_i^\top x_i + x_j^\top x_j - 2x_i^\top x_j$$

- Let $B = XX^\top$ be the $n \times n$ matrix of dot-products, $b_{ij} = x_i^\top x_j$. The above shows that $D$ can be computed from $B$.

- Some algebraic exercise shows that $B$ can be recovered from $D$ if we assume $\sum_{i=1}^nx_i = 0$. 
Multidimensional Scaling

- If we knew $X$, then SVD gives $X = UDV^\top$. As $X$ has rank at most $r = \min(n, p)$, we have at most $r$ non-zero singular values in $D$ and we can assume $U \in \mathbb{R}^{n \times r}$, $D \in \mathbb{R}^{r \times r}$ and $V^\top \in \mathbb{R}^{r \times p}$.

- The eigendecomposition of $B$ is then:

$$B = XX^\top = U D^2 U^\top = U \Lambda U^\top.$$  

This eigendecomposition can be obtained from $B$ without knowing $X$!

- Let $\tilde{x}_i^\top = U_i \Lambda_i^{\frac{1}{2}} \in \mathbb{R}^r$. If $r < p$, pad $\tilde{x}_i$ with 0s so that it has length $p$. Then,

$$\tilde{x}_i^\top \tilde{x}_j = U_i \Lambda U_j^\top = b_{ij} = x_i^\top x_j$$

and we have found a set of vectors with dot-products given by $B$, as desired.

- The vectors $\tilde{x}_i$ differs from $x_i$ only via the orthogonal matrix $V^\top$ (recall that $x_i^\top = U_i D V^\top = \tilde{x}_i^\top V^\top$) so are equivalent up to rotation and reflections.
We present a table of flying mileages between 10 American cities, distances calculated from our 2-dimensional world. Using $D$ as the starting point, metric MDS finds a configuration with the same distance matrix.

<table>
<thead>
<tr>
<th></th>
<th>ATLA</th>
<th>CHIG</th>
<th>DENV</th>
<th>HOUS</th>
<th>LA</th>
<th>MIAM</th>
<th>NY</th>
<th>SF</th>
<th>SEAT</th>
<th>DC</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>587</td>
<td>1212</td>
<td>701</td>
<td>1936</td>
<td>604</td>
<td>748</td>
<td>2139</td>
<td>2182</td>
<td>543</td>
<td></td>
</tr>
<tr>
<td>587</td>
<td>0</td>
<td>920</td>
<td>940</td>
<td>1745</td>
<td>1188</td>
<td>713</td>
<td>1858</td>
<td>1737</td>
<td>597</td>
<td></td>
</tr>
<tr>
<td>1212</td>
<td>920</td>
<td>0</td>
<td>879</td>
<td>831</td>
<td>1726</td>
<td>1631</td>
<td>949</td>
<td>1021</td>
<td>1494</td>
<td></td>
</tr>
<tr>
<td>701</td>
<td>940</td>
<td>879</td>
<td>0</td>
<td>1374</td>
<td>968</td>
<td>1420</td>
<td>1645</td>
<td>1891</td>
<td>1220</td>
<td></td>
</tr>
<tr>
<td>1936</td>
<td>1745</td>
<td>831</td>
<td>1374</td>
<td>0</td>
<td>2339</td>
<td>2451</td>
<td>347</td>
<td>959</td>
<td>2300</td>
<td></td>
</tr>
<tr>
<td>604</td>
<td>1188</td>
<td>1726</td>
<td>968</td>
<td>2339</td>
<td>0</td>
<td>1092</td>
<td>2594</td>
<td>2734</td>
<td>923</td>
<td></td>
</tr>
<tr>
<td>748</td>
<td>713</td>
<td>1631</td>
<td>1420</td>
<td>2451</td>
<td>1092</td>
<td>0</td>
<td>2571</td>
<td>2408</td>
<td>205</td>
<td></td>
</tr>
<tr>
<td>2139</td>
<td>1858</td>
<td>949</td>
<td>1645</td>
<td>347</td>
<td>2594</td>
<td>2571</td>
<td>0</td>
<td>678</td>
<td>2442</td>
<td></td>
</tr>
<tr>
<td>2182</td>
<td>1737</td>
<td>1021</td>
<td>1891</td>
<td>959</td>
<td>2734</td>
<td>2408</td>
<td>678</td>
<td>0</td>
<td>2329</td>
<td></td>
</tr>
<tr>
<td>543</td>
<td>597</td>
<td>1494</td>
<td>1220</td>
<td>2300</td>
<td>923</td>
<td>205</td>
<td>2442</td>
<td>2329</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
library(MASS)

us <- read.csv("http://www.stats.ox.ac.uk/~palamara/teaching/SML19/uscities.csv")

## use classical MDS to find lower dimensional views of the data
## recover X in 2 dimensions

us.classical <- cmdscale(d=us,k=2)
plot(us.classical)
text(us.classical,labels=names(us))
US City Flight Distances
Lower-dimensional Reconstructions

In classical MDS derivation, we used all eigenvalues in the eigendecomposition of $B$ to reconstruct

$$\tilde{x}_i = U_i \Lambda^{\frac{1}{2}}.$$ 

We can use only the largest $k < \min(n, p)$ eigenvalues and eigenvectors in the reconstruction, giving the ‘best’ $k$-dimensional view of the data.

This is analogous to PCA, where only the largest eigenvalues of $X^\top X$ are used, and the smallest ones effectively suppressed.

Indeed, PCA and classical MDS are duals and yield effectively the same result.
library(MASS)
crabs$spsex=paste(crabs$sp,crabs$sex,sep=""")
varnames<-c('FL','RW','CL','CW','BD')
Crabs <- crabs[,varnames]
Crabs.class <- factor(crabs$spsex)
crabsmds <- cmdscale(d= dist(Crabs),k=2)
plot(crabsmds, pch=20, cex=2,col=unclass(Crabs.class))
Crabs Data

Compare with previous PCA analysis. Classical MDS solution corresponds to the first 2 PCs.
Varieties of MDS

Generally, MDS is a class of dimensionality reduction techniques which represents data points $x_1, \ldots, x_n \in \mathbb{R}^p$ in a lower-dimensional space $z_1, \ldots, z_n \in \mathbb{R}^k$ which tries to preserve inter-point (dis)similarities.

- It requires only the matrix $D$ of pairwise dissimilarities $d_{ij} = d(x_i, x_j)$. For example, we can use Euclidean distance $d_{ij} = \|x_i - x_j\|_2$, but other dissimilarities are possible.
- MDS finds representations $z_1, \ldots, z_n \in \mathbb{R}^k$ such that

$$\|z_i - z_j\|_2 \approx d(x_i, x_j) = d_{ij},$$

and differences in dissimilarities are measured by the appropriate loss $\Delta(d_{ij}, \|z_i - z_j\|_2)$.
- Goal: Find $Z$ which minimizes the stress function

$$S(Z) = \sum_{i \neq j} \Delta(d_{ij}, \|z_i - z_j\|_2).$$
Varieties of MDS

- Choices of (dis)similarities and stress functions $S(Z)$ lead to different algorithms.
  - **Classical/Torgerson**: preserves inner products instead - strain function (cmdscale)
    $$S(Z) = \sum_{i \neq j} (b_{ij} - \langle z_i - \bar{z}, z_j - \bar{z} \rangle)^2$$
  - **Metric Shephard-Kruskal**: preserves distances w.r.t. squared stress
    $$S(Z) = \sum_{i \neq j} (d_{ij} - \| z_i - z_j \|^2)^2$$
  - **Sammon**: preserves shorter distances more ($\text{sammon}$)
    $$S(Z) = \sum_{i \neq j} \left( \frac{d_{ij} - \| z_i - z_j \|^2}{d_{ij}} \right)^2$$
  - **Non-Metric Shephard-Kruskal**: ignores actual distance values, only preserves ranks ($\text{isoMDS}$)
    $$S(Z) = \min_{g \text{ increasing}} \frac{\sum_{i \neq j} (g(d_{ij}) - \| z_i - z_j \|^2)^2}{\sum_{i \neq j} \| z_i - z_j \|^2}$$
Example: Language data

Presence or absence of 2867 homologous traits in 87 Indo-European languages.

> X<-read.table("http://www.stats.ox.ac.uk/~palamara/teaching/SML19/cognate.txt")
> X[1:15,1:16]

<table>
<thead>
<tr>
<th></th>
<th>V1</th>
<th>V2</th>
<th>V3</th>
<th>V4</th>
<th>V5</th>
<th>V6</th>
<th>V7</th>
<th>V8</th>
<th>V9</th>
<th>V10</th>
<th>V11</th>
<th>V12</th>
<th>V13</th>
<th>V14</th>
<th>V15</th>
<th>V16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Irish_A</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Irish_B</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Welsh_N</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Welsh_C</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Breton_List</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Breton_SE</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Breton_ST</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Romanian_List</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Vlach</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Italian</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Ladin</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Provencal</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>French</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Walloon</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>French_Creole_C</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Example: Language data

Using MDS with non-metric (Sammon) scaling.

MDS (i.e. cmdscale) which minimizes \((d_{ij}^2 - \tilde{d}_{ij}^2)^2\). Sammon thereby puts more weight on reproducing the separation of points which are close by forcing them apart.

Projection by MDS (Jaccard/sammon) with cluster discovery by k-means (Jaccard): There is an obvious east to west (top-left to bottom-right) separation of languages in the MDS and the clusters in the MDS grouping agree with the clusters discovered by agglomerative clustering and k-means. The two clustering methods group languages differently with k-means splitting the Germanic languages.

For more details, see the alternative/MDS## alternative/MDS# make a field to display the clusters..

Alternative: MDS-sammon does this nicely.

Diagrame: `eqscplot`\(\text{di.sam,points,pch=km,cluster,col=km,cluster}\)\n
Text: `text`\(\text{di.sam,points,labels=row.names(X),pos=4,col=km,cluster}\)\n
5
Nonlinear Dimensionality Reduction

Two aims of different varieties of MDS:

- To visualize the (dis)similarities among items in a dataset, where these (dis)similarities may not have Euclidean geometric interpretations.
- To perform **nonlinear** dimensionality reduction.

Many high-dimensional datasets exhibit low-dimensional structure (“live on a low-dimensional manifold”).
Isomap is a non-linear dimensional reduction technique based on classical MDS. Differs from other MDSs as it uses estimates of *geodesic distances* between the data points.

---

Tenenbaum et al. (2000)
Isomap

- Calculate Euclidean distances $D_{ij}$ for $i, j = 1, \ldots, n$ between all data points.
- Form a graph $G$ with $n$ samples as nodes, and edges between the respective $K$ nearest neighbours ($K$-Isomap) or between $i$ and $j$ if $D_{ij} < \epsilon$ ($\epsilon$-Isomap).
- For $i, j$ linked by an edge, set $D_{ij}^G = D_{ij}$. Otherwise, set $D_{ij}^G$ to be the shortest-path distance between $i$ and $j$ in $G$.
- Run classical MDS using distances $D_{ij}^G$.

**R function**: `isomap{vegan}`.
Faces

For faraway points, geodesic distance can be approximated by adding up a sequence of "short distances". This isometric step-by-step approach leads to the geodesic distance.

These methods are based on the intrinsic geometry of the manifold where the data are embedded. For instance, the technique of Isomap estimates the geodesic distances between points using the shortest Euclidean distances among their neighbors.

The algorithm proceeds by constructing a graph of Euclidean distances and then estimating the geodesic distances by minimizing the distortion of the local geometric structures.

A variant of Isomap is known as Locally Linear Embedding (LLE), which aims to preserve the local linear relationships between data points.

In summary, methods like Isomap and LLE provide a way to visualize high-dimensional data in a lower-dimensional space while preserving the intrinsic distances between points.
Handwritten Characters

Visualisation and Dimensionality Reduction

Isomap

The image shows a layout of handwritten characters, presumably digits, with axes labeled for 'Top arch articulation' and 'Bottom loop articulation'. The characters are spread out in a two-dimensional space, illustrating how different handwriting features can be visualised and dimensionally reduced for pattern recognition or analysis.