European Commission

Summary of R software for Symbolic Data Analysis

A. Irpino

Introduction

Three main packages in R are available for the Analysis of Symbolic Data:

- `RSDA` R to Symbolic Data Analysis
- `symbolicDA` Symbolic Data Analysis
- `HistDAWass` Histogram data analysis

RSDA R to Symbolic Data Analysis

R to Symbolic Data Analysis: Symbolic Data Analysis (SDA) was proposed by professor Edwin Diday in 1987, the main purpose of SDA is to substitute the set of rows (cases) in the data table for a concept (second order statistical unit). This package implements, to the symbolic case, certain techniques of automatic classification, as well as some linear models.

symbolicDA Symbolic Data Analysis

Symbolic Data Analysis: Symbolic data analysis methods: importing/exporting data from ASSO XML Files, distance calculation for symbolic data (Ichino-Yaguchi, de Carvalho measure), zoom star plot, 3d interval plot, multidimensional scaling for symbolic interval data, dynamic clustering based on distance matrix, HINoV method for symbolic data, Ichino's feature selection method, principal component analysis for symbolic interval data, decision trees for symbolic data based on optimal split with bagging, boosting and random forest approach (+visualization), kernel discriminant analysis for symbolic data, Kohonen's self-organizing maps for symbolic data, replication and profiling, artificial symbolic data generation.

HistDAWass Histogram data analysis

Histogram data analysis: In the framework of Symbolic Data Analysis, a relatively new approach to the statistical analysis of multi-valued data, we consider histogram-valued data, i.e., data described by univariate histograms. The methods and the basic statistics for histogram-valued data are mainly based on the L2 Wasserstein metric between distributions, i.e., the Euclidean metric between quantile functions. The package contains unsupervised classification techniques, least square regression and tools for histogram-valued data and for histogram time series.

Some packages for download offical statistics microdata

- `tidycensus` package
  - An integrated R interface to several United States Census Bureau APIs (https://www.census.gov/data/developers/data-sets.html) and the US Census Bureau's geographic boundary files. Allows R users to return Census and ACS data as tidyverse-ready data frames, and optionally returns a list-column with feature geometry for mapping and spatial analysis.
- `ipumsr` package
  - An easy way to work with census, survey, and geographic data provided by IPUMS in R. Generate and download data through the IPUMS API and load IPUMS files into R with their associated metadata to make analysis easier. IPUMS data describing 1.4 billion individuals drawn from over 750 censuses and surveys is available free of charge from the IPUMS website https://www.ipums.org.

- `eurostat` R Tools for Eurostat Open Data
  - Tools to download data from the Eurostat database https://ec.europa.eu/eurostat together with search and manipulation utilities.
- rOpenGov is a community of R package developers on open government data analytics and related topics. See https://ropengov.github.io/giscoR/
  - `giscoR` is an API package that helps to retrieve data from Eurostat - GISCO (the Geographic Information System of the COmmission). It also provides some lightweight data sets ready to use without downloading.
    GISCO is a geospatial open data repository including several data sets as countries, coastal lines, labels or NUTS levels. The data sets are usually provided at several resolution levels (60M/20M/10M/03M/01M) and in 3 different projections (4326/3035/3857).
    Note that the package does not provide metadata on the downloaded files, the information is available on the API webpage.
    Full site with examples and vignettes on https://ropengov.github.io/giscoR/

RSDA R to Symbolic Data Analysis: examples

The package is provided with some datasets for showing some main procedures developed during the SODAS and ASSO project for the application of Symbolic Data Analysis to Official Statistics.

How to generated a symbolic data table from a classic data table in RSDA?

The `classic.to.sym` function allows to convert a traditional table into a symbolic one, to this we must indicate the following parameters.

- `x` = a data.frame
- `concept` = variables to be used as a concept
- `variables` = variables to be used, conceptible with tidyselect options
- `default.numeric` = function that will be used by default for numerical values (sym.interval)
- `default.categorical` = functions to be used by default for categorical values (sym.model)

RSDA from classical to Symbolic Data Table (USCrime dataset 1,994 rows an 103 columns)

Show 10 entries Search:

	state	fold	population	householdsize	racepctblack	racePctWhite	racePctAsian	racePctHisp	agePct12t21	agePct12t29	agePct16t24	ageP
1	8	1	0.19	0.33	0.02	0.9	0.12	0.17	0.34	0.47	0.29	
2	53	1	0	0.16	0.12	0.74	0.45	0.07	0.26	0.59	0.35	
3	24	1	0	0.42	0.49	0.56	0.17	0.04	0.39	0.47	0.28	
4	34	1	0.04	0.77	1	0.08	0.12	0.1	0.51	0.5	0.34	
5	42	1	0.01	0.55	0.02	0.95	0.09	0.05	0.38	0.38	0.23	
6	6	1	0.02	0.28	0.06	0.54	1	0.25	0.31	0.48	0.27	
7	44	1	0.01	0.39	0	0.98	0.06	0.02	0.3	0.37	0.23	
8	6	1	0.01	0.74	0.03	0.46	0.2	1	0.52	0.55	0.36	
9	21	1	0.03	0.34	0.2	0.84	0.02	0	0.38	0.45	0.28	
10	29	1	0.01	0.4	0.06	0.87	0.3	0.03	0.9	0.82	0.8	

Showing 1 to 10 of 1,994 entries Previous 1 2 3 4 5 … 200 Next

RSDA: the symbolic data table

Each row is a state (46 states are recorded)

```
# A tibble: 46 × 4
  ViolentCrimesPerPop_hist NumInShelters      NumImmig ViolentCrimesPerPop
                <symblc_h>    <symblc_n>    <symblc_n>          <symblc_n>
1                    <hist> [0.00 : 0.32] [0.00 : 0.04]       [0.01 : 1.00]
2                    <hist> [0.01 : 0.18] [0.01 : 0.09]       [0.05 : 0.36]
3                    <hist> [0.00 : 1.00] [0.00 : 0.57]       [0.05 : 0.57]
4                    <hist> [0.00 : 0.08] [0.00 : 0.02]       [0.02 : 1.00]
5                    <hist> [0.00 : 1.00] [0.00 : 1.00]       [0.01 : 1.00]
6                    <hist> [0.00 : 0.68] [0.00 : 0.23]       [0.07 : 0.75]
# ℹ 40 more rows
```

The first <hist> codify

```
$breaks
[1] 0.0 0.2 0.4 0.6 0.8 1.0

$props
[1] 0.37209302 0.20930233 0.18604651 0.04651163 0.04651163
```

Example

A dataset with 130 rows and 5 variables about some suspects.

Show 10 entries Search:

	suspect	age	hair	eyes	region
1	1	42	h_red	e_brown	Bronx
2	2	20	h_black	e_green	Bronx
3	3	64	h_brown	e_brown	Brooklyn
4	4	55	h_blonde	e_brown	Bronx
5	5	4	h_brown	e_green	Manhattan
6	6	61	h_blonde	e_green	Bronx
7	7	61	h_white	e_black	Queens
8	8	32	h_blonde	e_brown	Manhattan
9	9	39	h_blonde	e_black	Brooklyn
10	10	50	h_brown	e_brown	Manhattan

Showing 1 to 10 of 130 entries Previous 1 2 3 4 5 … 13 Next

The symbolic data table

```
# A tibble: 100 × 4
              age               hair              eyes              region
        <symblc_n>         <symblc_s>        <symblc_s>           <symblc_s>
 1 [22.00 : 42.00]            {h_red} {e_brown,e_black}              {Bronx}
 2 [20.00 : 57.00] {h_black,h_blonde} {e_green,e_black}    {Bronx,Manhattan}
 3 [29.00 : 64.00]  {h_brown,h_white} {e_brown,e_green}    {Brooklyn,Queens}
 4 [14.00 : 55.00]         {h_blonde} {e_brown,e_black}    {Bronx,Manhattan}
 5  [4.00 : 47.00]    {h_brown,h_red}         {e_green}    {Manhattan,Bronx}
 6 [32.00 : 61.00] {h_blonde,h_white}  {e_green,e_blue}       {Bronx,Queens}
 7 [49.00 : 61.00]    {h_white,h_red}  {e_black,e_blue}       {Queens,Bronx}
 8  [8.00 : 32.00] {h_blonde,h_white} {e_brown,e_black} {Manhattan,Brooklyn}
 9 [39.00 : 67.00] {h_blonde,h_white} {e_black,e_brown}     {Brooklyn,Bronx}
10 [50.00 : 68.00]  {h_brown,h_black} {e_brown,e_green}    {Manhattan,Bronx}
# ℹ 90 more rows
```

Basic statistics

Symbolic Mean

```
data(example3)
head(example3)
```

```
# A tibble: 6 × 7
     F1              F2                        F3    F4        F5                F6
  <dbl>      <symblc_n>              <symblc_m> <dbl> <symblc_>       <symblc_n>
1   2.8   [1.00 : 2.00] M1:0.10 M2:0.70 M3:0.20   6   {e,g,i,k}   [0.00 : 90.00]
2   1.4   [3.00 : 9.00] M1:0.60 M2:0.30 M3:0.10   8   {a,b,c,d} [-90.00 : 98.00]
3   3.2  [-1.00 : 4.00] M1:0.20 M2:0.20 M3:0.60  -7   {2,b,1,c}  [65.00 : 90.00]
4  -2.1   [0.00 : 2.00] M1:0.90 M2:0.00 M3:0.10   0   {a,3,4,c}  [45.00 : 89.00]
5  -3   [-4.00 : -2.00] M1:0.60 M2:0.00 M3:0.40  -9.5 {e,g,i,k}  [20.00 : 40.00]
6   0.1 [10.00 : 21.00] M1:0.00 M2:0.70 M3:0.30  -1     {e,1,i}    [5.00 : 8.00]
# ℹ 1 more variable: F7 <symblc_n>
```

```
mean(example3$F1)#mean(example3[,1])
```

```
[1] 1.628571
```

```
mean(example3$F2)#mean(example3[,2])
```

```
[1] 5
```

```
mean(example3$F2,method = "interval")#mean(example3[,2],method = "interval")
```

```
<symbolic_interval[1]>
[1] [1.86 : 8.14]
```

Symbolic median

```
median(example3$F1) #median(example3[,1])
```

```
[1] 1.4
```

```
median(example3$F2) #median(example3[,2])
```

```
[1] 1.5
```

```
median(example3$F6, method = 'interval')# median(example3[,6], method = 'interval')
```

```
<symbolic_interval[1]>
[1] [5.00 : 89.00]
```

Variance and standard deviation

```
1 var(example3[,1])
```

```
[1] 15.98238
```

```
1 var(example3[,2])
```

```
[1] 90.66667
```

```
1 var(example3$F6)
```

```
[1] 1872.358
```

```
1 var(example3$F6, method = 'interval')
```

```
<symbolic_interval[1]>
[1] [2,408.97 : 1,670.51]
```

```
1 var(example3$F6, method = 'billard')
```

```
[1] 1355.143
```

```
1 sd(example3$F1)
```

```
[1] 3.997797
```

```
1 sd(example3$F2)
```

```
[1] 6.733003
```

```
1 sd(example3$F6)
```

```
[1] 30.59704
```

```
1 sd(example3$F6, method = 'interval')
```

```
<symbolic_interval[1]>
[1] [49.08 : 40.87]
```

```
1 sd(example3$F6, method = 'billard')
```

```
[1] 36.81226
```

Symbolic correlation

```
1  cor(example3$F1, example3$F4)#cor(example3[,1], example3[,4])
```

```
[1] 0.2864553
```

```
1  cor(example3$F2, example3$F6, method = 'centers')
```

```
[1] -0.6693648
```

```
1  cor(example3$F2, example3$F6, method = 'billard')
```

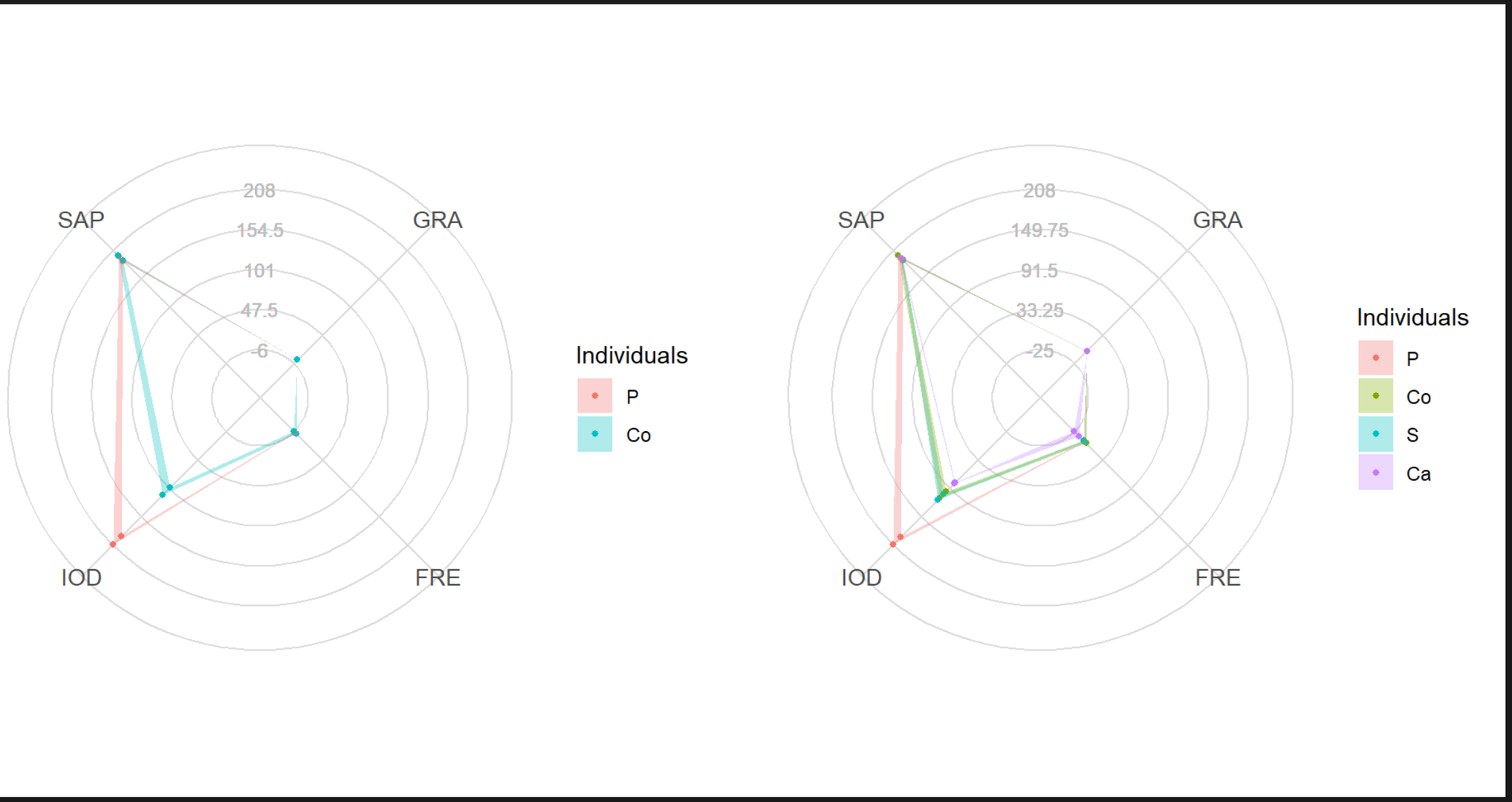
```
[1] -0.6020041
```

Radar plot for intervals (Oils data)

```
1 library(ggpolypath)
2 data(oils)
3 oils
```

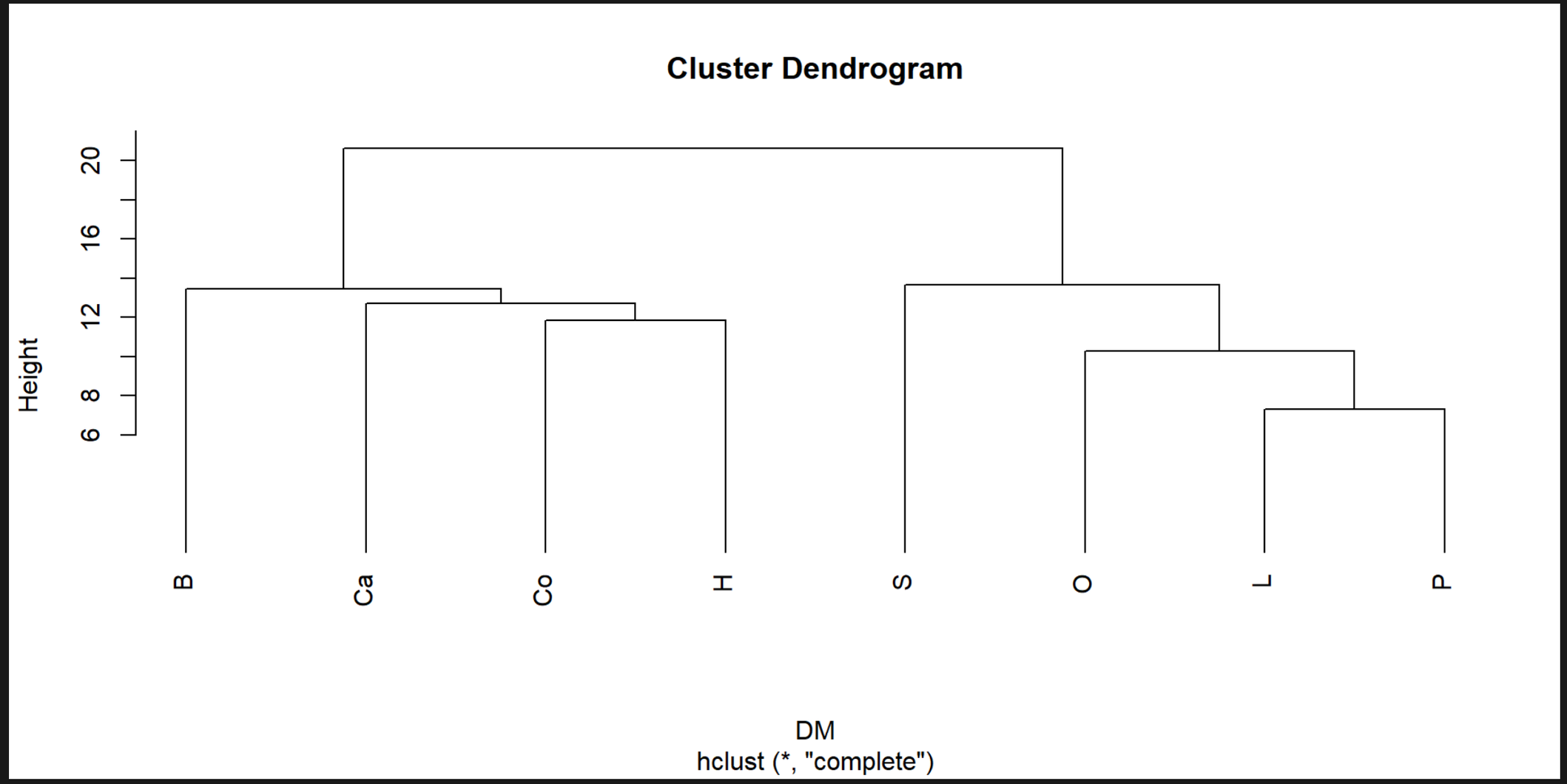
```
# A tibble: 8 × 4
            GRA                 FRE                 IOD                 SAP
*     <symblc_n>          <symblc_n>          <symblc_n>          <symblc_n>
1 [0.93 : 0.94] [-27.00 : -18.00] [170.00 : 204.00] [118.00 : 196.00]
2 [0.93 : 0.94]   [-5.00 : -4.00] [192.00 : 208.00] [188.00 : 197.00]
3 [0.92 : 0.92]   [-6.00 : -1.00]  [99.00 : 113.00] [189.00 : 198.00]
4 [0.92 : 0.93]   [-6.00 : -4.00] [104.00 : 116.00] [187.00 : 193.00]
5 [0.92 : 0.92] [-25.00 : -15.00]   [80.00 : 82.00] [189.00 : 193.00]
6 [0.91 : 0.92]     [0.00 : 6.00]   [79.00 : 90.00] [187.00 : 196.00]
7 [0.86 : 0.87]   [30.00 : 38.00]   [40.00 : 48.00] [190.00 : 199.00]
8 [0.86 : 0.86]   [22.00 : 32.00]   [53.00 : 77.00] [190.00 : 202.00]
```

```
oils <- RSDA:::to.v3(RSDA:::to.v2(oils))
p1<-sym.radar.plot(oils[2:3,])
p2<-sym.radar.plot(oils[2:5,])
p1+p2
```

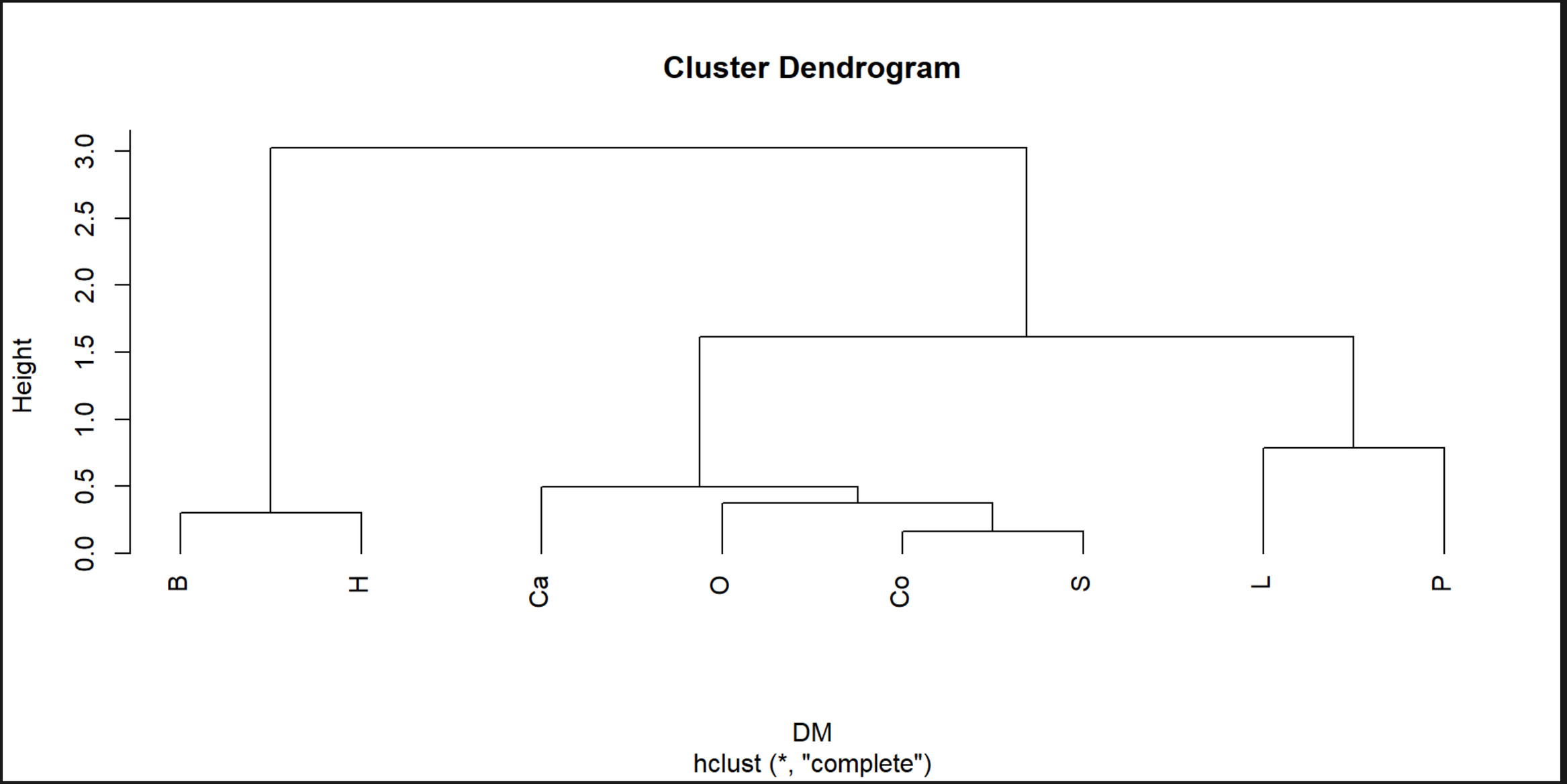


Distances for intervals

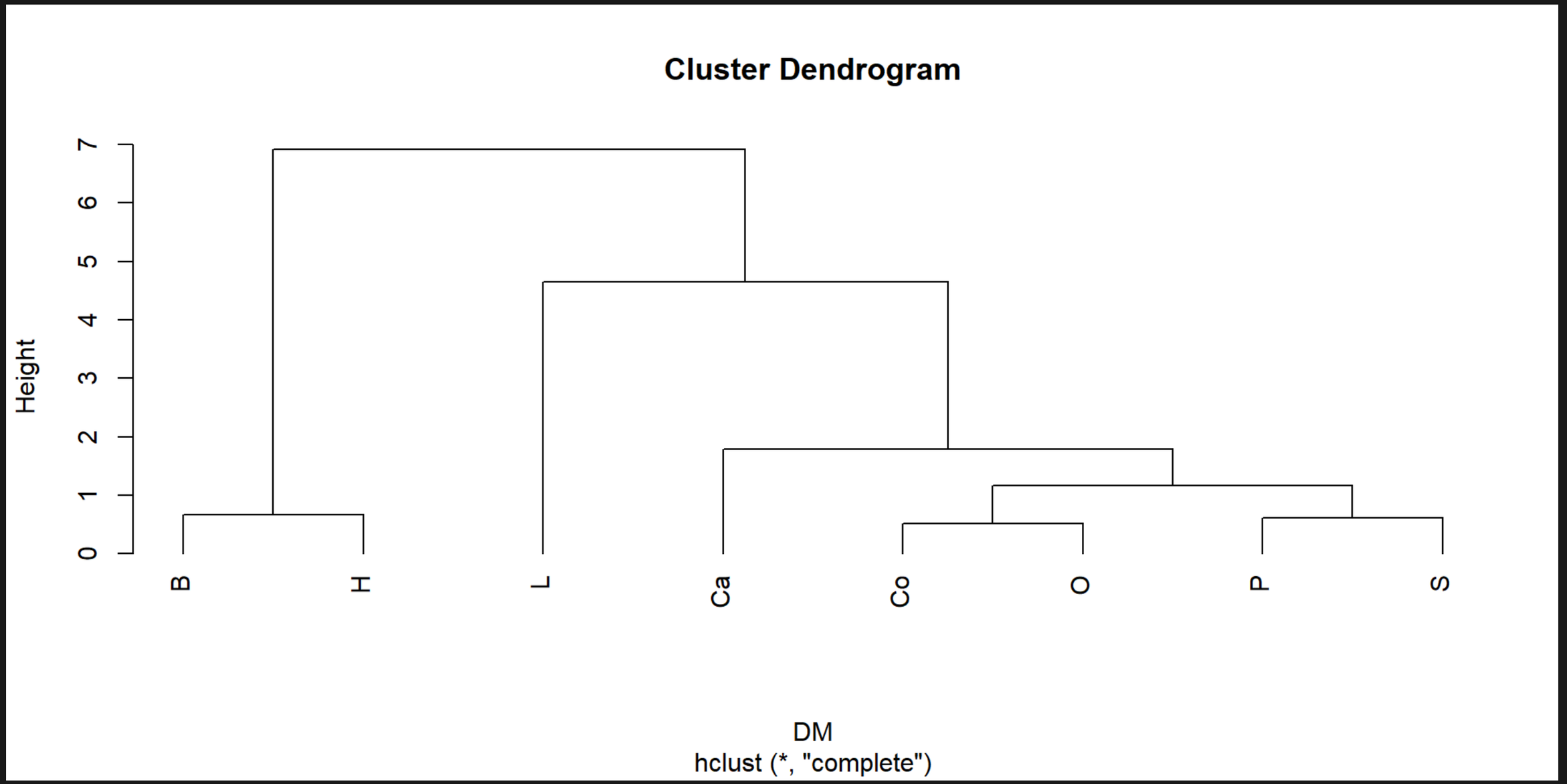
Gowda-Diday



Ichino


Cluster Dendrogram
Height
0.0
0.5
1.0
1.5
2.0
2.5
3.0
B
H
Ca
O
Co
S
L
P
DM
hclust (*, "complete")

Hausdorff


Cluster Dendrogram
Height
0
1
2
3
4
5
6
7
B
H
L
Ca
Co
O
P
S
DM
hclust (*, "complete")

Linear regression for intervals

The data is an object of class symbolic_tbl with 67 rows and 9 columns.

The data for this example come from a study by Stamey et al. (1989) that examined the correlation between the level of prostate specific antigen (PSA) and a number of clinical measures, in 97 men who were about to receive a radical prostatectomy. The goal is to predict the log of PSA (`lpsa`) from a number of measurements including log cancer volume (`lcavol`), log prostate weight `lweight`, `age`, log of benign prostatic hyperplasia amount `lbph`, seminal vesicle invasion `svi`, log of capsular penetration `lcp`, Gleason score `gleason`, and percent of Gleason scores 4 or 5 `pgg45`.

```
# A tibble: 67 × 9
           lcavol       lweight             age            lbph           svi
        <symblc_n>    <symblc_n>      <symblc_n>      <symblc_n>    <symblc_n>
 1 [-0.58 : -0.58] [2.77 : 2.78] [50.00 : 50.01] [-1.39 : -1.37] [0.00 : 0.02]
 2 [-1.00 : -0.99] [3.31 : 3.32] [58.00 : 58.00] [-1.39 : -1.32] [0.00 : 0.02]
 3 [-0.52 : -0.48] [2.69 : 2.70] [74.00 : 74.00] [-1.39 : -1.33] [0.00 : 0.01]
 4 [-1.21 : -1.17] [3.28 : 3.29] [58.00 : 58.00] [-1.39 : -1.34] [0.00 : 0.03]
 5   [0.73 : 0.78] [3.43 : 3.44] [61.99 : 62.01] [-1.39 : -1.33] [0.00 : 0.03]
 6 [-1.05 : -1.01] [3.23 : 3.23] [50.00 : 50.01] [-1.39 : -1.39] [0.00 : 0.03]
 7   [0.69 : 0.72] [3.53 : 3.55] [58.00 : 58.01]   [1.52 : 1.54] [0.00 : 0.02]
 8   [0.24 : 0.28] [3.60 : 3.61] [64.99 : 65.00] [-1.39 : -1.35] [0.00 : 0.03]
 9 [-1.35 : -1.30] [3.59 : 3.60] [62.99 : 63.00]   [1.24 : 1.29] [0.00 : 0.03]
10   [1.59 : 1.62] [3.02 : 3.02] [63.00 : 63.00] [-1.39 : -1.38] [0.00 : 0.02]
# ℹ 57 more rows
# ℹ 4 more variables: lcp <symblc_n>, gleason <symblc_n>, pgg45 <symblc_n>,
#   lpsa <symblc_n>
```

Training

```
1  data(int_prost_train)
2  data(int_prost_test)
3  res.cm <- sym.lm(formula = lpsa~., sym.data = int_prost_train, method = 'cm')
4  res.cm
```

```
Call:
stats::lm(formula = formula, data = centers)

Coefficients:
(Intercept)       lcavol      lweight          age         lbph          svi
   0.411537     0.579327     0.614128    -0.018659     0.143918     0.730937
        lcp      gleason        pgg45
  -0.205536    -0.030924     0.009507
```

Prediction

```
1 pred.cm <- sym.predict(model = res.cm, new.sym.data = int_prost_test)
```

Testing

```
1 RMSE.L(int_prost_test$lpsa, pred.cm$Fitted)
```

```
[1] 0.7229999
```

```
1 RMSE.U(int_prost_test$lpsa, pred.cm$Fitted)
```

```
[1] 0.7192467
```

```
1 R2.L(int_prost_test$lpsa, pred.cm$Fitted)
```

```
[1] 0.501419
```

```
1 R2.U(int_prost_test$lpsa, pred.cm$Fitted)
```

```
[1] 0.5058389
```

```
1 deter.coefficient(int_prost_test$lpsa, pred.cm$Fitted)
```

```
[1] 0.4962964
```

PCA for intervals

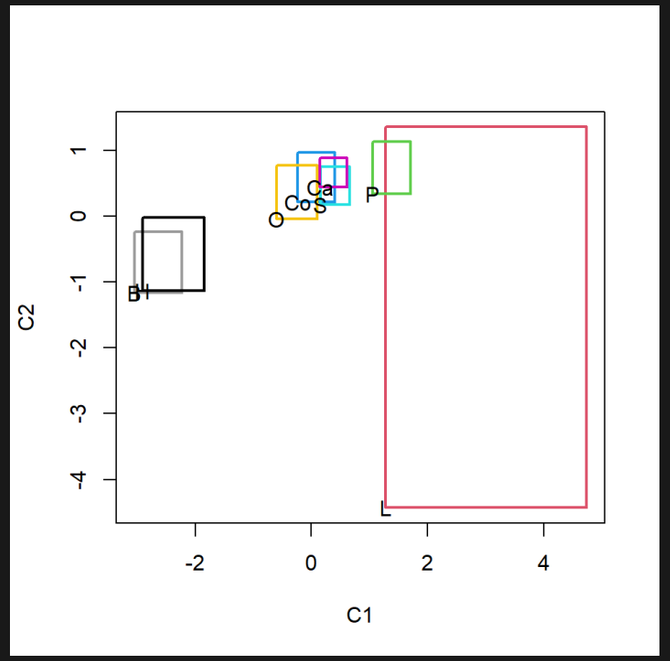
Example Oils data

```
data("oils")
oils
```

```
# A tibble: 8 × 4
                GRA                 FRE                 IOD                 SAP
*       <symblc_n>          <symblc_n>          <symblc_n>          <symblc_n>
1 [0.93 : 0.94]  [-27.00 : -18.00]  [170.00 : 204.00]  [118.00 : 196.00]
2 [0.93 : 0.94]    [-5.00 : -4.00]  [192.00 : 208.00]  [188.00 : 197.00]
3 [0.92 : 0.92]    [-6.00 : -1.00]   [99.00 : 113.00]  [189.00 : 198.00]
4 [0.92 : 0.93]    [-6.00 : -4.00]  [104.00 : 116.00]  [187.00 : 193.00]
5 [0.92 : 0.92]  [-25.00 : -15.00]    [80.00 : 82.00]  [189.00 : 193.00]
6 [0.91 : 0.92]      [0.00 : 6.00]    [79.00 : 90.00]  [187.00 : 196.00]
7 [0.86 : 0.87]    [30.00 : 38.00]    [40.00 : 48.00]  [190.00 : 199.00]
8 [0.86 : 0.86]    [22.00 : 32.00]    [53.00 : 77.00]  [190.00 : 202.00]
```

```
res <- sym.pca(oils,'centers')
```

```
plot(res, choix = "ind")
```




```
plot(res, choix = "var")
```