European Commission

Software demonstration for clustering of distributional SD

A. Irpino, R. Verde

ESTP Cologne 14-16 May 2024

Hard-partitive algorithms

Dynamic clustering (a generalization of k-means algorithm)

The dynamic clustering algorithm: after initialization, a two-step algorithm looks for the best partition into k classes and the best representation of clusters.

The DCA algorithm

1. **Initialize the algorithm**
    1. Set a number k of clusters
    2. Set $T = 0$
    3. Generate a random partition of the objects $P(0)$
    4. Compute the criterion (the Within-cluster sum of Squares), $CRIT(0)$
2. **Representation step**
    1. Set $T = T + 1$
    2. Compute the prototypes of each cluster using $P(T-1)$
3. **Allocation step**
    1. Allocate objects to the nearest prototype obtaining the partition $P(T)$
    2. Compute $CRIT(T)$
4. **STOP CONDITION**
    - If $CRIT(T) < CRIT(T-1)$ goto **step 2**, else return results.

The WH_kmeans function

The function uses L_2 Wasserstein-based statistics

```r
library("HistDAWass")
```

```r
results=WH_kmeans(x,                 # A MatH object
                  k,                 # The number of required clusters
               rep=5,             # How many time it is initialized
               simplify=FALSE,    # A flag for speeding up,
                                  # approximating data
               qua=10,            # If symplify=TRUE how many quantiles
                                  # are used for approximating the
                                  # distributions
               standardize=FALSE) # Do you need to standardize variables?
```

The output of `WH_kmeans` function

- `results` A list. It contains the best solution among the repetitions, i.e. the one having the minimum criterion.
  - `results$IDX` A vector. The clusters at which the objects are assigned.
  - `results$cardinality` A vector. The cardinality of each final cluster.
  - `results$centers` A MatH object with the description of centers.
  - `results$Crit` A number. The criterion (Within-cluster Sum od squared distances from the centers).
  - `results$quality` A number. The percentage of Total SS explained by the model. (The higher the better)

Adaptive distances-based dynamic clustering (Irpino, Verde, and De Carvalho 2014)

A system of weights are calculated for the variables, for their components, clusterwise or globally. The system of weights is useful if data are clustered into non-spherical classes.

The Adaptive DCA algorithm

1. **Initialize the algorithm**
    1. Set $T = 0$, a number k of clusters, initialize weights $W(0)$.
    2. Generate a random partition of the objects $P(0)$
    3. Compute the criterion (the Within-cluster sum of Squares), $CRIT(0)$
2. **Representation step** (Fix the Partition and the Weights)
    1. Set $T = T + 1$. Compute the prototypes $G(T)$ of each cluster using $P(T-1)$ and $W(T-1)$.
3. **Weighting step** (Fix the Prototypes and the Weights)
    1. Compute the weight system $W(T)$ using $G(T)$ and $P(T-1)$
4. **Allocation step** (Fix the Weights and Prototypes)
    1. Assing objects to the nearest prototype in $G(T)$ using $W(T)$, obtaining the partition $P(T)$
    2. Compute $CRIT(T)$
5. **STOP CONDITION** If $CRIT(T) < CRIT(T-1)$ goto **step 2**, else return results.

Two possible functions for computing the weights and four possible combinations of weights

The system of weights may be

- Multiplicative: the product of weights is fixed (generally equal to one)
- Additive: the sum of weights is fixed (generally equal to one)

Ways for assigning weights.

1. A weight for **each variable**
2. A weight for **each variable** and **each cluster**
3. A weight for **each component** of a distributional variable (we mean the *position* and the *variability* component related to the decomposition of the L_2 Wasserstein distance)
4. A weight for **each component** and **each cluster**

The WH_adaptive_kmeans function

```
results= WH_adaptive.kmeans(x, k,schema = 1, init, rep, simplify = FALSE,
                  qua = 10,standardize = FALSE,
                  weight.sys = "PROD",
                  theta = 2, init.weights = "EQUAL")
```

Parameter	Description
x	A MatH object (a matrix of distributionH).
k	An integer, the number of groups.
schema	a number from 1 to 4:
1	A weight for each variable (default)
2	A weight for the average and the dispersion component of each variable
3	Same as 1 but a set of weights for each cluster
4	Same as 2 but a set of weights for each cluster
init	(optional, do not use) initialization for partitioning the data default is ’RPART’
rep	Maximum number of repetitions of the algorithm (default rep=5).

Parameter	Description
simplify	A logic value (default is FALSE), if TRUE histograms are recomputed in order to speed-up the algorithm.
qua	An integer, if simplify=TRUE is the number of quantiles used for recoding the histograms.
standardize	A logic value (default is FALSE). If TRUE, histogram-valued data are standardized,variable by variable, using the Wassertein based standard deviation.
weight.sys	a string. Weights may add to one (’SUM’) or their product is equal to 1 (’PROD’, default).
theta	a number. A parameter if weight.sys=’SUM’, default is 2.
init.weights	a string how to initialize weights: ’EQUAL’ (default), all weights are the same, ’RANDOM’, weights are initalised at random.

The output

Name	description
results	A list.Returns the best solution among the repetitions, i.e. the one having the minimum sum of squares criterion.
results$IDX	A vector. The final clusters labels of the objects.
results$cardinality	A vector. The cardinality of each final cluster.
results$proto	A MatH object with the description of centers.
results$weights	A matrix of weights for each component of each variable and each cluster.
results$Crit	A number. The criterion (Weighted Within-cluster SS) value at the end of the run.
results$TOTSSQ	The total SSQ computed with the system of weights.
results$BSQ	The Between-clusters SSQ computed with the system of weights.
results$WSQ	The Within-clusters SSQ computed with the system of weights.
results$quality	A number. The proportion of TSS explained by the model. (The higher the better)

Hierarchical clustering

Hierarchical clustering

```
results= WH_hclust (x, simplify=FALSE, qua=10,
                     standardize=FALSE, distance="WDIST",
                     method="complete")
```

Input param.	Description
x	A MatH object (a matrix of distributionH)
simplify	As before.
qua	As before.
standardize	As before.
distance	A string default WDIST the L_2 Wasserstein distance (other distances will be implemented)
method	A string, default="complete", is the the agglomeration method to be used. This should be (an unambiguous abbreviation of) one of ward.D, ward.D2, single, complete, average (= UPGMA), mcquitty (= WPGMA), median (= WPGMC) or centroid (= UPGMC).

Output : An object of the class `hclust` which describes the tree produced by the method.

An example: Time Use Data

We will use data from the IPUMS https://www.ipums.org/, which provides census and survey data from around the world integrated across time and space.

- Among the collection, we use micro data from the Annual American Time Use Survey (ATUS).
- We considered two waves 2010, 2022.
- The original sample extracted from the collection (https://www.atusdata.org,(Flood et al. 2023)) was of about 21,4K respondents.
- We extracted classes of respondents accordingly to the YEAR, the Occupation (OCC2_CPS8, General occupation category, main job (CPS)), and the Sex of workers (workers are people who declared to work at least 10 minutes in the previous day).
    - We considered only concepts with a size grater than 30.
    - We obtained 34 concepts of respondents.
- We considered only five activities: Food, Personal Care, Social and leisure, Traveling, and Working. The remaining activities time use are summed into an “All the rest of time”.

The 34 concepts and the time use variables

The concepts

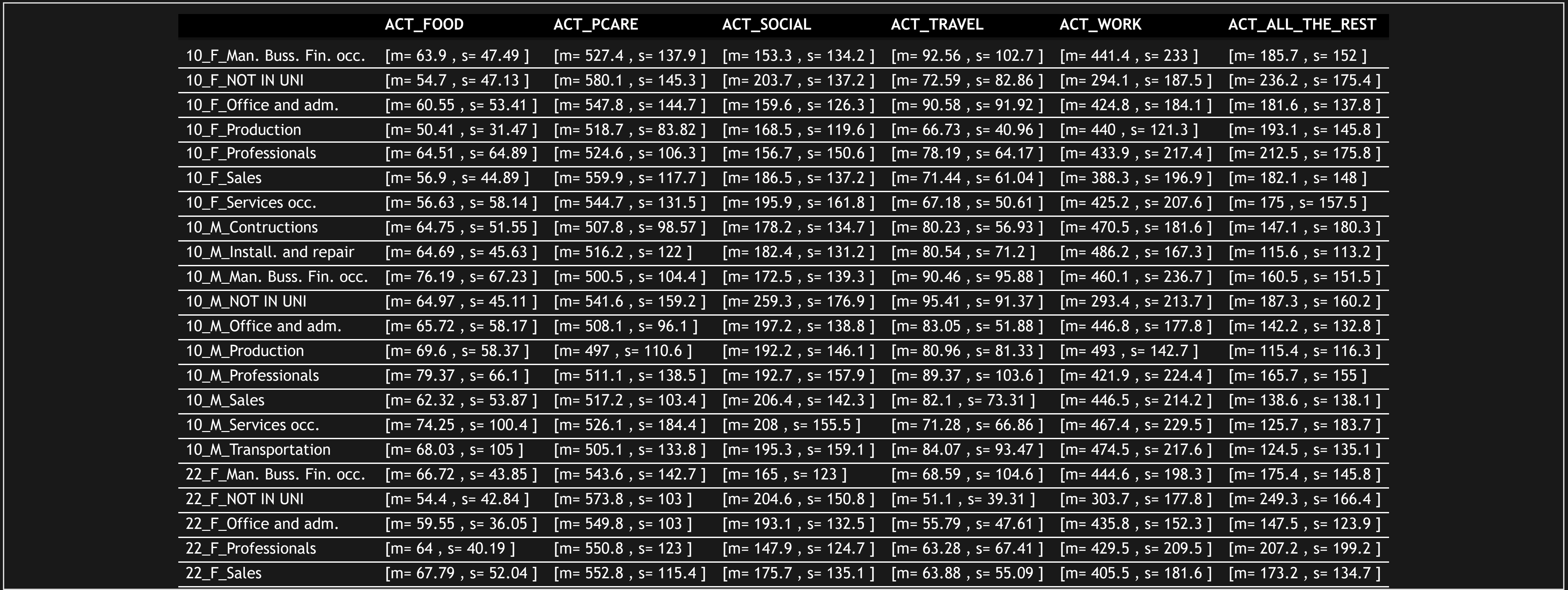
```
 [1] "2010_Female_Man. Buss. Fin. occ." "2010_Female_NOT IN UNI"
 [3] "2010_Female_Office and adm."      "2010_Female_Production"
 [5] "2010_Female_Professionals"        "2010_Female_Sales"
 [7] "2010_Female_Services occ."        "2010_Male_Contructions"
 [9] "2010_Male_Install. and repair"    "2010_Male_Man. Buss. Fin. occ."
[11] "2010_Male_NOT IN UNI"             "2010_Male_Office and adm."
[13] "2010_Male_Production"             "2010_Male_Professionals"
[15] "2010_Male_Sales"                  "2010_Male_Services occ."
[17] "2010_Male_Transportation"         "2022_Female_Man. Buss. Fin. occ."
[19] "2022_Female_NOT IN UNI"           "2022_Female_Office and adm."
[21] "2022_Female_Professionals"        "2022_Female_Sales"
[23] "2022_Female_Services occ."        "2022_Female_Transportation"
[25] "2022_Male_Contructions"           "2022_Male_Install. and repair"
[27] "2022_Male_Man. Buss. Fin. occ."   "2022_Male_NOT IN UNI"
[29] "2022_Male_Office and adm."        "2022_Male_Production"
[31] "2022_Male_Professionals"          "2022_Male_Sales"
[33] "2022_Male_Services occ."          "2022_Male_Transportation"
```

Note: this is a didactic example.

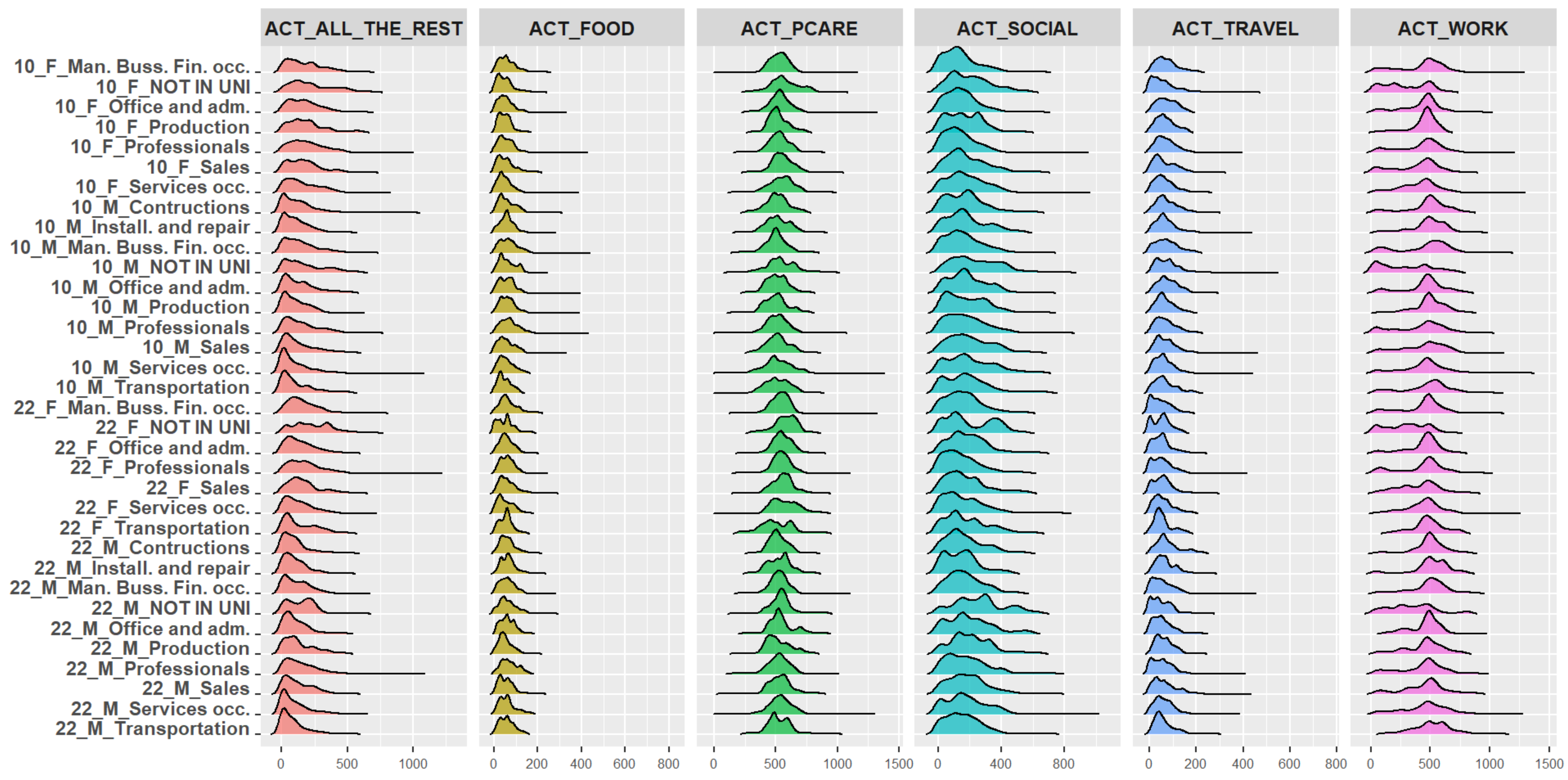
The variables

- ACT_PCARE ACT: Personal care
- ACT_WORK ACT: Working and Work-related Activities
- ACT_FOOD ACT: Eat and drinking
- ACT_SOCIAL ACT: Socializing, relaxing, and leisure
- ACT_TRAVEL ACT: Traveling
- ACT_ALL_THE_REST ACT: all the other activities

The (histogram) data table



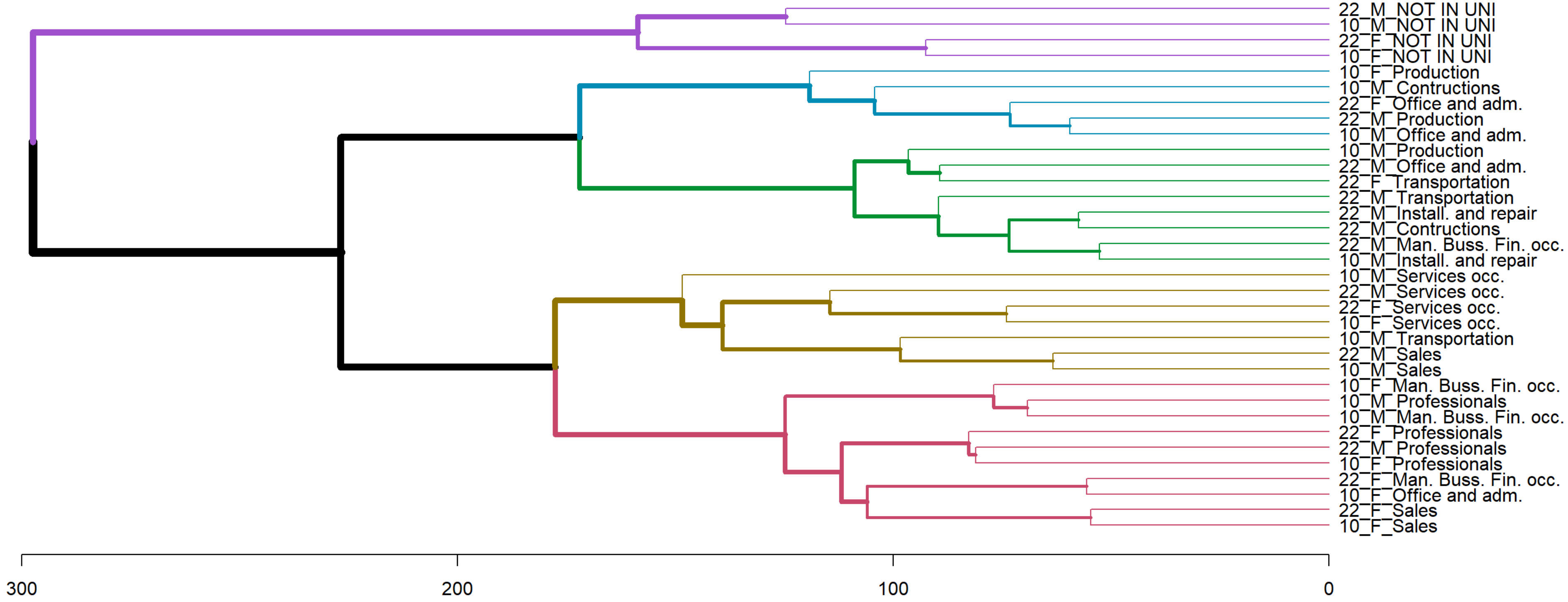
	ACT_FOOD	ACT_PCARE	ACT_SOCIAL	ACT_TRAVEL	ACT_WORK	ACT_ALL_THE_REST
10_F_Man. Buss. Fin. occ.	[m= 63.9 , s= 47.49]	[m= 527.4 , s= 137.9]	[m= 153.3 , s= 134.2]	[m= 92.56 , s= 102.7]	[m= 441.4 , s= 233]	[m= 185.7 , s= 152]
10_F_NOT IN UNI	[m= 54.7 , s= 47.13]	[m= 580.1 , s= 145.3]	[m= 203.7 , s= 137.2]	[m= 72.59 , s= 82.86]	[m= 294.1 , s= 187.5]	[m= 236.2 , s= 175.4]
10_F_Office and adm.	[m= 60.55 , s= 53.41]	[m= 547.8 , s= 144.7]	[m= 159.6 , s= 126.3]	[m= 90.58 , s= 91.92]	[m= 424.8 , s= 184.1]	[m= 181.6 , s= 137.8]
10_F_Production	[m= 50.41 , s= 31.47]	[m= 518.7 , s= 83.82]	[m= 168.5 , s= 119.6]	[m= 66.73 , s= 40.96]	[m= 440 , s= 121.3]	[m= 193.1 , s= 145.8]
10_F_Professionals	[m= 64.51 , s= 64.89]	[m= 524.6 , s= 106.3]	[m= 156.7 , s= 150.6]	[m= 78.19 , s= 64.17]	[m= 433.9 , s= 217.4]	[m= 212.5 , s= 175.8]
10_F_Sales	[m= 56.9 , s= 44.89]	[m= 559.9 , s= 117.7]	[m= 186.5 , s= 137.2]	[m= 71.44 , s= 61.04]	[m= 388.3 , s= 196.9]	[m= 182.1 , s= 148]
10_F_Services occ.	[m= 56.63 , s= 58.14]	[m= 544.7 , s= 131.5]	[m= 195.9 , s= 161.8]	[m= 67.18 , s= 50.61]	[m= 425.2 , s= 207.6]	[m= 175 , s= 157.5]
10_M_Contructions	[m= 64.75 , s= 51.55]	[m= 507.8 , s= 98.57]	[m= 178.2 , s= 134.7]	[m= 80.23 , s= 56.93]	[m= 470.5 , s= 181.6]	[m= 147.1 , s= 180.3]
10_M_Install. and repair	[m= 64.69 , s= 45.63]	[m= 516.2 , s= 122]	[m= 182.4 , s= 131.2]	[m= 80.54 , s= 71.2]	[m= 486.2 , s= 167.3]	[m= 115.6 , s= 113.2]
10_M_Man. Buss. Fin. occ.	[m= 76.19 , s= 67.23]	[m= 500.5 , s= 104.4]	[m= 172.5 , s= 139.3]	[m= 90.46 , s= 95.88]	[m= 460.1 , s= 236.7]	[m= 160.5 , s= 151.5]
10_M_NOT IN UNI	[m= 64.97 , s= 45.11]	[m= 541.6 , s= 159.2]	[m= 259.3 , s= 176.9]	[m= 95.41 , s= 91.37]	[m= 293.4 , s= 213.7]	[m= 187.3 , s= 160.2]
10_M_Office and adm.	[m= 65.72 , s= 58.17]	[m= 508.1 , s= 96.1]	[m= 197.2 , s= 138.8]	[m= 83.05 , s= 51.88]	[m= 446.8 , s= 177.8]	[m= 142.2 , s= 132.8]
10_M_Production	[m= 69.6 , s= 58.37]	[m= 497 , s= 110.6]	[m= 192.2 , s= 146.1]	[m= 80.96 , s= 81.33]	[m= 493 , s= 142.7]	[m= 115.4 , s= 116.3]
10_M_Professionals	[m= 79.37 , s= 66.1]	[m= 511.1 , s= 138.5]	[m= 192.7 , s= 157.9]	[m= 89.37 , s= 103.6]	[m= 421.9 , s= 224.4]	[m= 165.7 , s= 155]
10_M_Sales	[m= 62.32 , s= 53.87]	[m= 517.2 , s= 103.4]	[m= 206.4 , s= 142.3]	[m= 82.1 , s= 73.31]	[m= 446.5 , s= 214.2]	[m= 138.6 , s= 138.1]
10_M_Services occ.	[m= 74.25 , s= 100.4]	[m= 526.1 , s= 184.4]	[m= 208 , s= 155.5]	[m= 71.28 , s= 66.86]	[m= 467.4 , s= 229.5]	[m= 125.7 , s= 183.7]
10_M_Transportation	[m= 68.03 , s= 105]	[m= 505.1 , s= 133.8]	[m= 195.3 , s= 159.1]	[m= 84.07 , s= 93.47]	[m= 474.5 , s= 217.6]	[m= 124.5 , s= 135.1]
22_F_Man. Buss. Fin. occ.	[m= 66.72 , s= 43.85]	[m= 543.6 , s= 142.7]	[m= 165 , s= 123]	[m= 68.59 , s= 104.6]	[m= 444.6 , s= 198.3]	[m= 175.4 , s= 145.8]
22_F_NOT IN UNI	[m= 54.4 , s= 42.84]	[m= 573.8 , s= 103]	[m= 204.6 , s= 150.8]	[m= 51.1 , s= 39.31]	[m= 303.7 , s= 177.8]	[m= 249.3 , s= 166.4]
22_F_Office and adm.	[m= 59.55 , s= 36.05]	[m= 549.8 , s= 103]	[m= 193.1 , s= 132.5]	[m= 55.79 , s= 47.61]	[m= 435.8 , s= 152.3]	[m= 147.5 , s= 123.9]
22_F_Professionals	[m= 64 , s= 40.19]	[m= 550.8 , s= 123]	[m= 147.9 , s= 124.7]	[m= 63.28 , s= 67.41]	[m= 429.5 , s= 209.5]	[m= 207.2 , s= 199.2]
22_F_Sales	[m= 67.79 , s= 52.04]	[m= 552.8 , s= 115.4]	[m= 175.7 , s= 135.1]	[m= 63.88 , s= 55.09]	[m= 405.5 , s= 181.6]	[m= 173.2 , s= 134.7]



Time use distributional data table

Time use data: Hierarchical clustering

```
results=WH_hclust(x = HMAT2, method="complete")
```



Time use dendrogram using complete linkage

Time use data: DCA (or Kmeans)

```r
DCA.5k.resu<-WH_kmeans(HMAT2,k = 5,rep = 50)
```

Cluster 1 # 4

10_F_NOT IN UNI; 10_M_NOT IN UNI; 22_F_NOT IN UNI; 22_M_NOT IN UNI

Cluster 2 # 6

10_M_Man. Buss. Fin. occ.; 10_M_Professionals; 10_M_Sales; 10_M_Services occ.; 10_M_Transportation; 22_M_Services occ.

Cluster 3 # 9

10_F_Man. Buss. Fin. occ.; 10_F_Office and adm.; 10_F_Professionals; 10_F_Sales; 10_F_Services occ.; 22_F_Man. Buss. Fin. occ.; 22_F_Professionals; 22_F_Sales; 22_M_Professionals

Cluster 4 # 8

10_F_Production; 10_M_Contructions; 10_M_Office and adm.; 22_F_Office and adm.; 22_F_Services occ.; 22_F_Transportation; 22_M_Production; 22_M_Sales

Cluster 5 # 7

10_M_Install. and repair; 10_M_Production; 22_M_Contructions; 22_M_Install. and repair; 22_M_Man. Buss. Fin. occ.; 22_M_Office and adm.; 22_M_Transportation

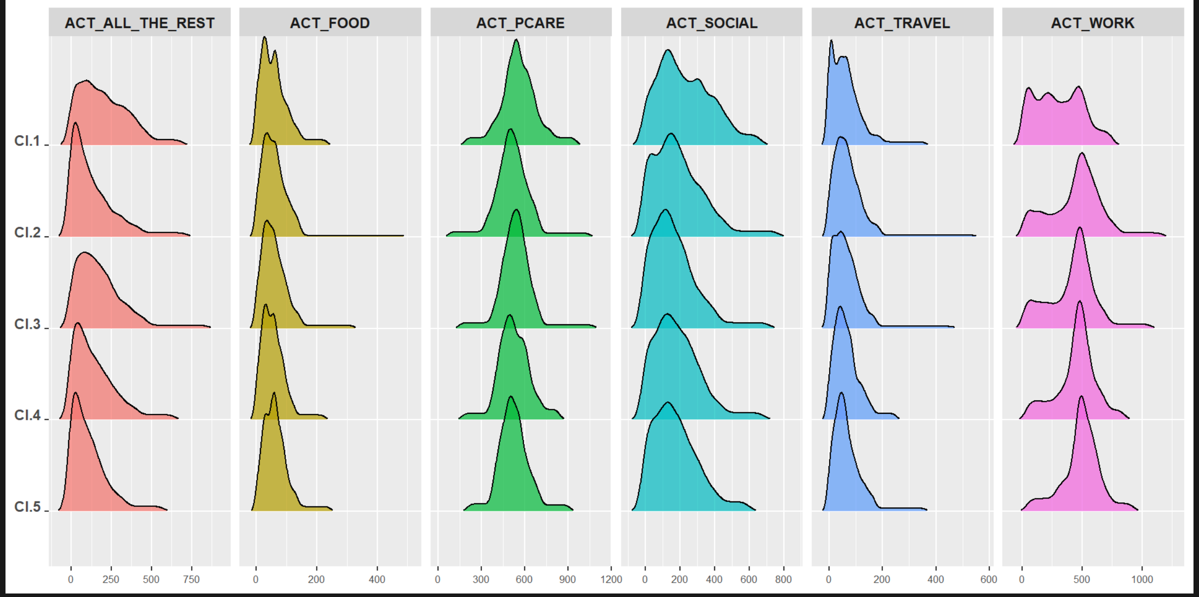
The clusters centers

```
DCA.5k.resu$solution$centers
```

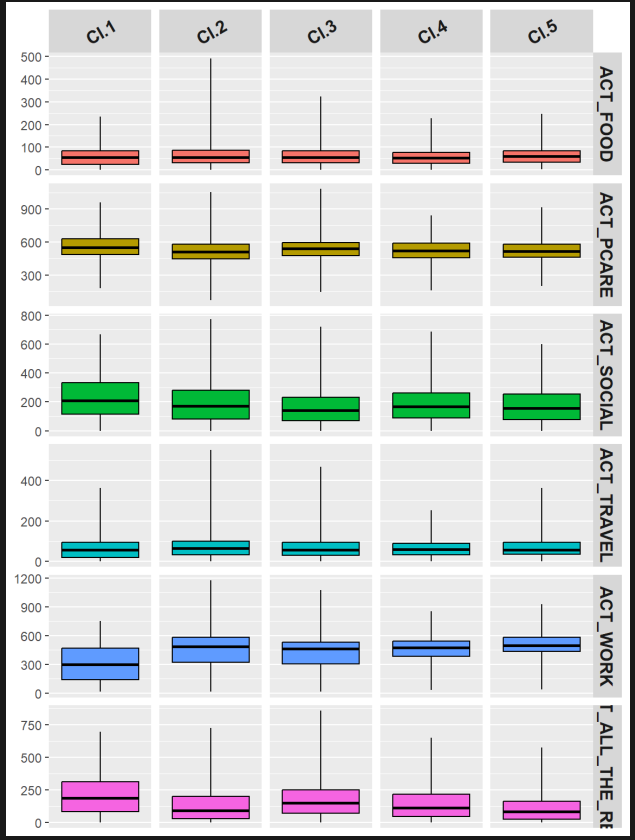
	ACT_FOOD	ACT_PCARE	ACT_SOCIAL	ACT_TRAVEL	ACT_WORK	ACT_ALL_THE_REST
Cl.1	[m= 60.48 , s= 46.23]	[m= 559.9 , s= 131.9]	[m= 233.4 , s= 154.8]	[m= 69.13 , s= 66.4]	[m= 309.4 , s= 193.8]	[m= 206.4 , s= 154.2]
Cl.2	[m= 70.02 , s= 70.1]	[m= 516.8 , s= 136.6]	[m= 196 , s= 153.6]	[m= 81.06 , s= 82.22]	[m= 458 , s= 223.4]	[m= 137.1 , s= 147.3]
Cl.3	[m= 64.93 , s= 52.57]	[m= 542 , s= 125.3]	[m= 169.8 , s= 137.1]	[m= 73.29 , s= 72.83]	[m= 424.9 , s= 201.3]	[m= 185.1 , s= 158.4]
Cl.4	[m= 57.59 , s= 40.38]	[m= 524.8 , s= 112.8]	[m= 188.8 , s= 138.8]	[m= 68.03 , s= 49.51]	[m= 453 , s= 162.7]	[m= 148.3 , s= 135.5]
Cl.5	[m= 64.77 , s= 42.48]	[m= 525.4 , s= 111.2]	[m= 177 , s= 129.5]	[m= 72.05 , s= 60.68]	[m= 491.8 , s= 159.3]	[m= 112.7 , s= 114.8]

The plots of the centers

```
plot(DCA.5k.resu$solution$centers, type="DENS")
```



```
plot(DCA.5k.resu$solution$centers, type="BOXPLOT")
```



Tools of interpretation

The Within SUM OF SQUARES

```
DCA.5k.resu$solution$Crit
```

```
[1] 152252.3
```

The Quality of partition index $R^2 = 1 - \frac{WSS}{TSS}$

```
DCA.5k.resu$quality
```

```
[1] 0.5938733
```

Other tools of interpretation

- We can explore for each variable, what is the QPI
- We can check for each center how far each local mean distribution is from the grand mean. The highest is the distance, the more the cluster is characterized by that variable for that cluster.
- We can compute if the variability is mainly due to the variability of the averages or by that of the shapes of the distributions.

That can enrich the interpretation of the results.

Fuzzy c_means

We observed that the QPI is not so high, meaning that a strong separation of the clusters is not observed.

Fuzzy c-means, is a generalization of the k-means algorithm wich allows the unit to belong to a cluster with a **membership** degree (0= the concept doesn't belong at all to the cluster, 1= the concept belongs exclusively to that cluster).

In fuzzy c-means, where c represents the user-defined number of clusters, a further output is produced: the membership of concepts to clusters matrix.

The code to call in R for the fuzzy cmeans is

```r
WH_fcmeans(x, k, m = 1.6, rep, simplify = FALSE, qua = 10, standardize = FALSE)
```

Fcmeans: arguments of the function

```r
set.seed(1234)
WH_fcmeans(x, k, m = 1.6, rep, simplify = FALSE, qua = 10, standardize = FALSE)
```

Parameter	Description
x	The distributional data table
k	The number of fuzzy clusters
m	The fuzzyness parameter $1 < m < \infty$, 1=sharp clusters
rep	As for K-means
simplify	As for K-means
qua	As for K-means
standardize	As for K-means

Fcmeans: main outputs

Slots of the output	Description
`solution`	A list. It returns the best solution among the repetitions, i.e. the one having the minimum sum of squared deviations.
`solution$membership`	A matrix. The membership degree of each unit to each cluster.
`solution$IDX`	A vector. The crisp assignment to a cluster.
`solution$cardinality`	A vector. The size of each final cluster (after the crisp assignment).
`solution$Crit`	A number. The criterion (Sum of square deviation from the prototypes) value at the end of the run.
`quality`	A number. The percentage of Sum of square deviation explained by the model. (The higher the better)

Fcmeans: other outputs

Slots of the output	Description
`Crisp_clu`	A list, containing the concepts belonging to the crisp clusters, accordingly to the highest membership.
`TSQ`	The Total Sum of Squares.
`WSQ`	The Within Sum of Squares.
`BSQ`	The Between Sum of Squares.
`ProtoGEN`	The mean distribution of each distributional variable.

Time use: fcmeans

```
DCA.5k.fc_resu<-WH_fcmeans(HMAT2,k = 5,rep = 50,m=1.5)
```

	Cl.1	Cl.2	Cl.3	Cl.4	Cl.5	assign
10_F_Man. Buss. Fin. occ.	0.645	0.045	0.275	0.027	0.008	1
10_F_Office and adm.	0.701	0.088	0.159	0.042	0.010	1
10_F_Professionals	0.749	0.054	0.164	0.023	0.010	1
10_F_Sales	0.561	0.184	0.152	0.035	0.068	1
10_F_Services occ.	0.449	0.141	0.351	0.041	0.017	1
22_F_Man. Buss. Fin. occ.	0.684	0.071	0.194	0.045	0.006	1
22_F_Professionals	0.782	0.061	0.113	0.029	0.016	1
22_F_Sales	0.394	0.350	0.169	0.057	0.029	1
22_M_Professionals	0.581	0.076	0.310	0.026	0.007	1
10_F_Production	0.135	0.538	0.085	0.216	0.026	2
10_M_Contructions	0.155	0.365	0.204	0.270	0.007	2
10_M_Office and adm.	0.035	0.779	0.079	0.105	0.002	2
22_F_Office and adm.	0.026	0.873	0.023	0.075	0.003	2

Crisp membership

Cluster 1 # 9

10_F_Man. Buss. Fin. occ.; 10_F_Office and adm.; 10_F_Professionals; 10_F_Sales; 10_F_Services occ.; 22_F_Man. Buss. Fin. occ.; 22_F_Professionals; 22_F_Sales; 22_M_Professionals

Cluster 2 # 7

10_F_Production; 10_M_Contructions; 10_M_Office and adm.; 22_F_Office and adm.; 22_F_Services occ.; 22_M_Production; 22_M_Sales

Cluster 3 # 6

10_M_Man. Buss. Fin. occ.; 10_M_Professionals; 10_M_Sales; 10_M_Services occ.; 10_M_Transportation; 22_M_Services occ.

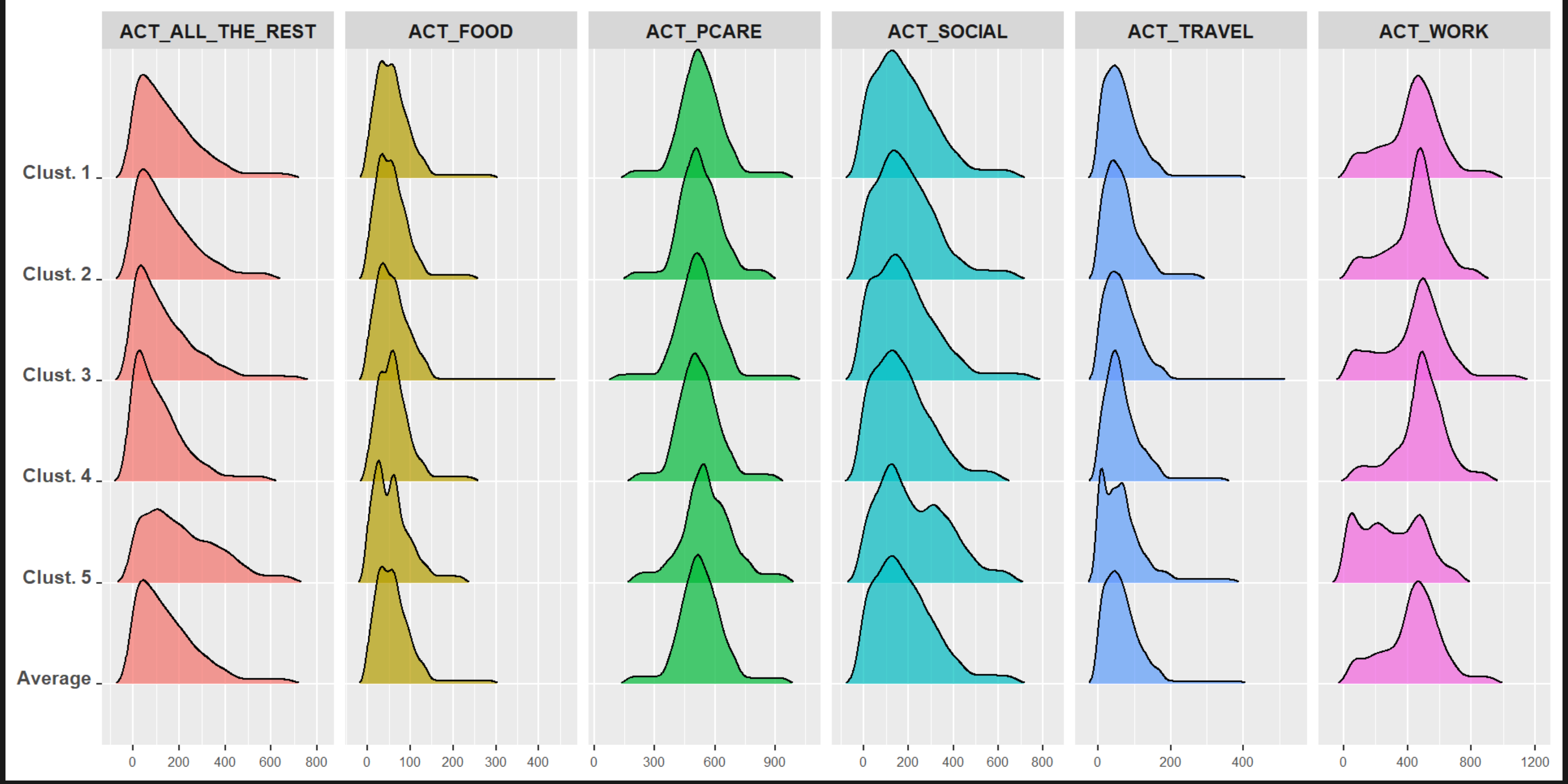
Cluster 4 # 8

10_M_Install. and repair; 10_M_Production; 22_F_Transportation; 22_M_Contructions; 22_M_Install. and repair; 22_M_Man. Buss. Fin. occ.; 22_M_Office and adm.; 22_M_Transportation

Cluster 5 # 4

10_F_NOT IN UNI; 10_M_NOT IN UNI; 22_F_NOT IN UNI; 22_M_NOT IN UNI

The means of each cluster and the grand mean



Cluster 1 # 9

10_F_Man. Buss. Fin. occ.; 10_F_Office and adm.; 10_F_Professionals; 10_F_Sales; 10_F_Services occ.; 22_F_Man. Buss. Fin. occ.; 22_F_Professionals; 22_F_Sales; 22_M_Professionals

Cluster 2 # 7

10_F_Production; 10_M_Contructions; 10_M_Office and adm.; 22_F_Office and adm.; 22_F_Services occ.; 22_M_Production; 22_M_Sales

Cluster 3 # 6

10_M_Man. Buss. Fin. occ.; 10_M_Professionals; 10_M_Sales; 10_M_Services occ.; 10_M_Transportation; 22_M_Services occ.

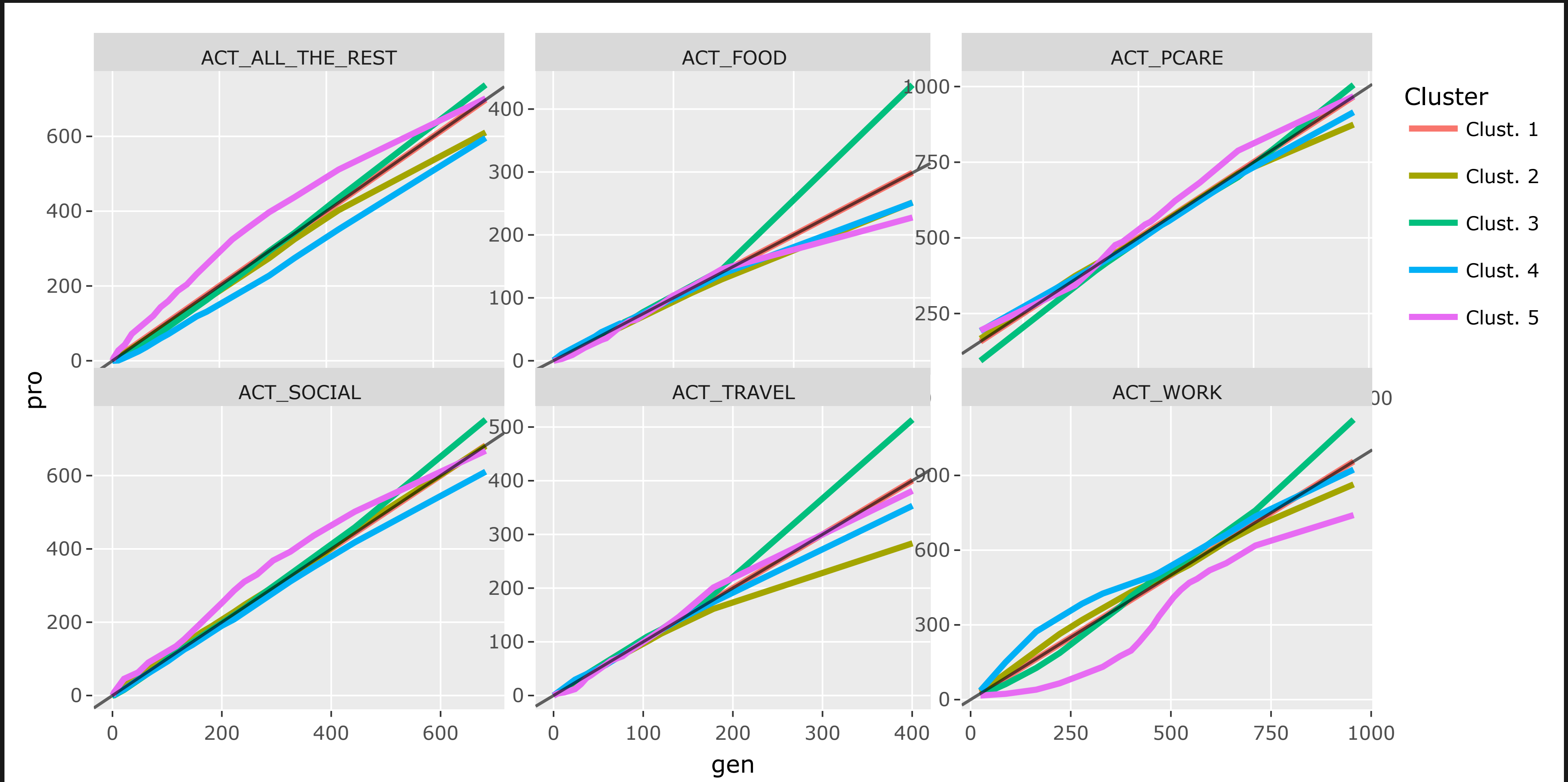
Cluster 4 # 8

10_M_Install. and repair; 10_M_Production; 22_F_Transportation; 22_M_Contructions; 22_M_Install. and repair; 22_M_Man. Buss. Fin. occ.; 22_M_Office and adm.; 22_M_Transportation

Cluster 5 # 4

10_F_NOT IN UNI; 10_M_NOT IN UNI; 22_F_NOT IN UNI; 22_M_NOT IN UNI

Comparing via QQ plots



Cluster 1 # 9

10_F_Man. Buss. Fin. occ.; 10_F_Office and adm.; 10_F_Professionals; 10_F_Sales; 10_F_Services occ.; 22_F_Man. Buss. Fin. occ.; 22_F_Professionals; 22_F_Sales; 22_M_Professionals

Cluster 2 # 7

10_F_Production; 10_M_Contructions; 10_M_Office and adm.; 22_F_Office and adm.; 22_F_Services occ.; 22_M_Production; 22_M_Sales

Cluster 3 # 6

10_M_Man. Buss. Fin. occ.; 10_M_Professionals; 10_M_Sales; 10_M_Services occ.; 10_M_Transportation; 22_M_Services occ.

Cluster 4 # 8

10_M_Install. and repair; 10_M_Production; 22_F_Transportation; 22_M_Contructions; 22_M_Install. and repair; 22_M_Man. Buss. Fin. occ.; 22_M_Office and adm.; 22_M_Transportation

Cluster 5 # 4

10_F_NOT IN UNI; 10_M_NOT IN UNI; 22_F_NOT IN UNI; 22_M_NOT IN UNI

Other clustering methods implemented in HistDAWass

Kohonen Self Organizing Maps

Adaptive distances-based k-means

Adaptive distances-based Fuzzy c-means

References

Flood, Sarah M., Liana C. Sayer, Daniel Backman, and Annie Chen. 2023. “American Time Use Survey Data Extract Builder: Version 3.2 [Dataset].” College Park, MD: University of Maryland; Minneapolis, MN: IPUMS. https://doi.org/https://doi.org/10.18128/D060.V3.2.

Irpino, A., R. Verde, and F. A. T. De Carvalho. 2014. “Dynamic Clustering of Histogram Data Based on Adaptive Squared Wasserstein Distances.” *Expert Systems with Applications* 41 (7): 3351–66. https://doi.org/http://dx.doi.org/10.1016/j.eswa.2013.12.001.