

Supplemental material for

“Understanding the Heterogeneity of Swiss Alpine Summer Farms for Tailored Agricultural Policies: A Typology”, by Maximilian Meyer, Christian Gazzarin, Pierrick Jan, and Nadja El Benni, published in *Mountain Research and Development* 44(1), 2024. (See <https://bioone.org/toc/mred/44/1>)

APPENDIX S1 A typology of Swiss alpine summer farms: cluster analysis documentation

This document describes the cluster analysis used to create a typology of Swiss Alpine summer farms. First, we read our dataframe that has 5892 farms and 62 variables available.

Introduction

The results of the operational typology of Swiss alpine farms are based on a cluster analysis. First, the question is raised about how to determine the optimal number of operational types (clusters). The decision criteria used include both the measure of *internal validity* and the *interpretability* of the clusters. Both criteria were selected based on the following guidelines:

<https://stats.stackexchange.com/questions/195456/how-to-select-a-clustering-method-how-to-validate-a-cluster-solution-to-warran>

Internal validity is used because we have a census of all farms, meaning we have information about all farms in the population. Conversely, this means that external validity is not necessary because it is not a sample of a population.

Interpretability is used to ensure the meaningfulness of the typology, especially in practice, which is warranted from expert assessment.

Variables selection

Here we select factors for a potential typology using cluster analysis. These include: Total number of Normal Stocking Units, Has dairy cows, Has sheep, Has cattle, Is private, Elevation, Has road access.

```
vs <- c("NST_total", "milking_cows", "sheep", "cattle", "private_count",  
       "elevation_mean", "has_road_access")  
# Subset  
vars <- subset(df, select = vs)  
vars[] <- lapply(vars, as.numeric)
```

Summary statistics for each of the clustering variables

```
knitr::kable(summary(vars), caption = "", digits = 2,  
             col.names = c("Total NST", "Has dairy cows",  
                           "Has sheep", "Has cattle",  
                           "Is private", "Elevation [m]",  
                           "Has road access"))
```

Total NST	Has dairy cows	Has sheep	Has cattle	Is private	Elevation [m]	Has road access
Min. : 0.1411	Min. :0.0000	Min. :0.0000	Min. :0.0000	Min. :0.0000	Min. : 197.7	Min. :0.0000
1st Qu.: 15.5596	1st Qu.:0.0000	1st Qu.:0.0000	1st Qu.:1.0000	1st Qu.:0.0000	1st Qu.:1183.4	1st Qu.:1.0000
Median : 29.1541	Median :1.0000	Median :0.0000	Median :1.0000	Median :1.0000	Median :1410.0	Median :1.0000

Total NST	Has dairy cows	Has sheep	Has cattle	Is private	Elevation [m]	Has road access
Mean :	Mean	Mean	Mean	Mean	Mean	Mean
43.4728	:0.6763	:0.1314	:0.8854	:0.6417	:1456.4	:0.8057
3rd Qu.:	3rd	3rd	3rd	3rd	3rd	3rd
55.2794	Qu.:1.0000	Qu.:0.0000	Qu.:1.0000	Qu.:1.0000	Qu.:1740.0	Qu.:1.0000
Max.	Max.	Max.	Max.	Max.	Max.	Max.
:846.4796	:1.0000	:1.0000	:1.0000	:1.0000	:2935.6	:1.0000

Normalization

```
max = apply(vars, 2, max, na.rm = T)
min = apply(vars, 2, min, na.rm = T)
vars_scaled = as.data.frame(scale(vars, center = min, scale = max - min))
rm(max, min)
```

Analysis

Before we start the analysis, we use the Hopkins statistic to evaluate whether the data set contains any meaningful clusters. This is a statistical hypothesis test in which the null hypothesis is that the data are uniformly randomly distributed. A value close to 1 indicates that the data are highly clustered, random data are more likely to have values around 0.5, and uniformly distributed data are more likely to have values close to 0. https://en.wikipedia.org/wiki/Hopkins_statistic

```
res <- get_clust_tendency(vars, n = nrow(vars)-1, graph = FALSE)
res$hopkins_stat
```

```
## [1] 0.9762781
```

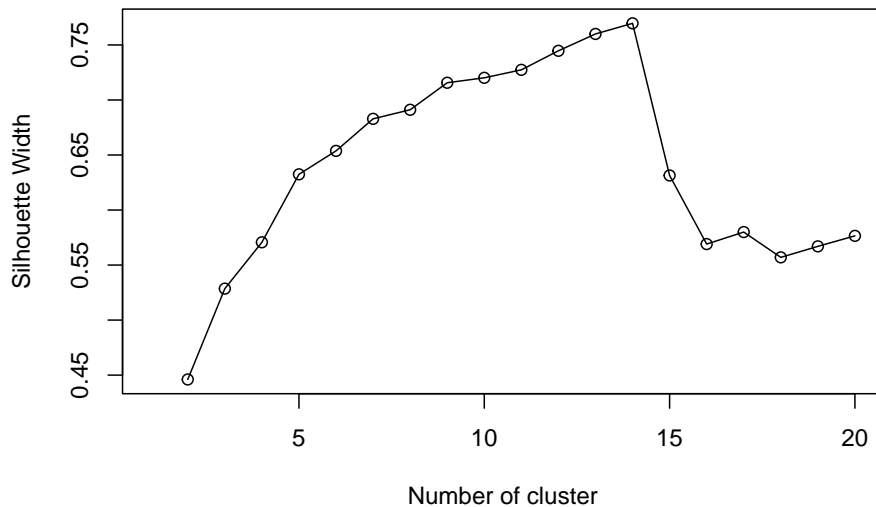
The value indicates strong structuring or clusterability..

We use a partitioning clustering method (partitioning around medoids) with the general similarity coefficient of Gower (Gower distance). See *Gower, J. C. (1971): A General Coefficient of Similarity and Some of Its Properties. In Biometrics 27 (4), p. 857.*

The general principle of the algorithm is to minimize the average dissimilarity of a representative farm to all other farms in the same cluster. In addition, PAM is well suited because representative farms are also of interest (called medoids) to which other farms are assigned by clustering. This is relevant for more advanced full cost accounting.

```
gds <- as.matrix(daisy(vars_scaled,metric = "gower"))
```

The selection of 6 clusters/types each is based on the graph below, which describes the size of the silhouette coefficient, dependent on the number of clusters. The silhouette is described as “a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). The silhouette ranges from - 1 to +1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters. If most objects have a high value, then the clustering configuration is appropriate. If many points have a low or negative value, then the clustering configuration may have too many or too few clusters.” See [https://en.wikipedia.org/wiki/Silhouette_\(clustering\)](https://en.wikipedia.org/wiki/Silhouette_(clustering)) for more information, including calculation details.



Although a 14 cluster solution would be ideal, its interpretability is very difficult, which is highlighted by expert assessment. Therefore, 6 clusters are a compromise between interpretability and statistical accuracy.

```
no_clusters <- 6
pam_fit <- pam(gds, diss = TRUE, k = no_clusters)
df <- cbind(df, clusterNumber = pam_fit$clustering)
```

Results

We now interpret the clusters and name them according to one to three characteristics that are unique and delimit the type towards others types to make it distinct.

```
Type <- c("Private dairy farms ",
          "Communal mixed cattle and dairy farms ",
          "Communal cattle farms ",
          "Remote farms",
          "Small, private cattle farms",
          "Sheep farms")

t <- cbind(Type,t)
t <- subset(t, select = -c(clusterNumber))

t <- t %>%
  mutate(Has_dairy_cows = Has_dairy_cows*100,
         Has_sheep = Has_sheep*100,
         Has_cattle = Has_cattle*100,
         Private = Private*100,
         Has_road_access = Has_road_access*100)
```

Type	N	Total NST	has dairy cows	Has sheep	Has cattle	Is private	Elevation [m]	has road access
Private dairy farms	2180	37.01	100.00	4.45	96.74	100.00	1372.49	100.00
Communal mixed cattle and dairy farms	1350	73.82	100.00	6.37	96.52	0.00	1464.16	96.44
Communal cattle farms	467	36.43	0.00	5.78	94.00	0.00	1355.45	80.09
Remote farms	523	46.08	86.81	12.81	98.47	71.89	1800.94	0.00
Small, private cattle farms	855	20.57	0.00	2.81	95.79	100.00	1275.48	90.88

Type	N	Total NST	has dairy cows	Has sheep	Has cattle	Is private	Elevation [m]	has road access
Sheep farms	517	33.08	0.19	91.49	6.19	71.57	1831.42	22.05

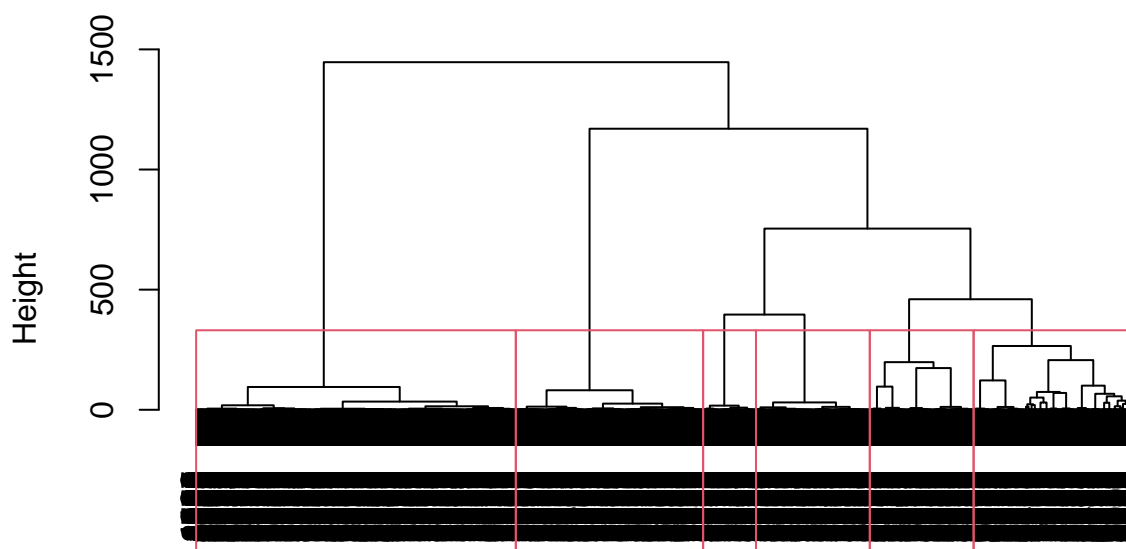
Robustness check

We also perform hierarchical clustering as a robustness check and to graphically display the partitioning. For this we use the euclidean distance and ward agglomeration.

```
vars_dist <- dist(vars_scaled, method = "euclidean")

plot(hclust(vars_dist,method = "ward.D"), xlab = "Ward D")
rect.hclust(hclust(vars_dist,method = "ward.D"), k = 6)
```

Cluster Dendrogram



Ward D
hclust (*, "ward.D")

```
hclust_ward_d2 <- hclust(vars_dist,method = "ward.D")

clusters <- as.data.frame(cutree(hclust_ward_d2, k = 6))
colnames(clusters) <- "cluster_k6_hierarchical_ward_d"
df <- cbind(df,clusters)

t_h <- df %>%
  group_by(cluster_k6_hierarchical_ward_d) %>%
  summarize(N = n(),
            Total_normal_stocking_unit = mean(NST_total, na.rm = TRUE),
            Has_dairy_cows = mean(milking_cows, na.rm = TRUE)*100,
            Has_sheep = mean(sheep, na.rm = TRUE)*100,
            Has_cattle = mean(cattle, na.rm = TRUE)*100,
            Private = mean(private_count, na.rm = TRUE)*100,
```

```

Elevation = mean(elevation_mean, na.rm = TRUE),
Has_road_access = mean(has_road_access, na.rm = TRUE)*100)

Type <- c("Private dairy farms ",
         "Communal mixed cattle and dairy farms ",
         "Communal cattle farms ",
         "Remote farms",
         "Small, private cattle farms",
         "Sheep farms")

t_h <- cbind(Type,t_h)
t_h <- subset(t_h, select = -c(cluster_k6_hierarchical_ward_d))

knitr::kable(t_h, caption = "", digits = 2,
             col.names = c("Type",
                          "N","Total NST","has dairy cows",
                          "Has sheep", "Has cattle",
                          "Is private", "Elevation [m]",
                          "has road access"))

```

Type	N	Total NST	has dairy cows	Has sheep	Has cattle	Is private	Elevation [m]	has road access
Private dairy farms	2014	36.00	100.00	0.00	100.00	100.00	1362.79	100.0
Communal mixed cattle and dairy farms	1180	73.93	100.00	0.00	100.00	0.00	1449.22	100.0
Communal cattle farms	333	37.00	0.00	0.00	100.00	0.00	1259.63	100.0
Remote farms	654	37.84	64.22	0.00	100.00	62.08	1699.84	0.0
Small, private cattle farms	717	21.32	0.00	0.00	100.00	100.00	1256.35	100.0
Sheep farms	994	44.31	37.32	77.87	32.09	64.79	1704.47	50.6

The resulting interpretation is similar to the one of the final results, as the cluster would correspond to very similar type names, when, again, using one to three main characteristics of the farms. However, the type “sheep farms” becomes less distinct, wherefore we prefer the typology presented in the main results.

REVIEW 1

Correlation of variables

```

cor(vars)

##          NST_total milking_cows          sheep          cattle private_count
## NST_total      1.00000000  0.23007705  0.059899521  0.10244681 -0.319362689
## milking_cows  0.23007705  1.00000000 -0.292617057  0.37532365 -0.052385548
## sheep         0.05989952 -0.29261706  1.000000000 -0.57789777  0.005564645
## cattle        0.10244681  0.37532365 -0.577897770  1.00000000 -0.030940837
## private_count -0.31936269 -0.05238555  0.005564645 -0.03094084  1.000000000
## elevation_mean 0.14731766 -0.01680681  0.292096377 -0.25248331 -0.112265583
## has_road_access 0.05067726  0.24882102 -0.360094890  0.34049477  0.004263975
## clusterNumber -0.13875457 -0.81903825  0.432764033 -0.47716099  0.014555524

```

```
##          elevation_mean has_road_access clusterNumber
## NST_total      0.14731766      0.050677265 -0.13875457
## milking_cows   -0.01680681      0.248821018 -0.81903825
## sheep          0.29209638     -0.360094890  0.43276403
## cattle         -0.25248331      0.340494775 -0.47716099
## private_count  -0.11226558      0.004263975  0.01455552
## elevation_mean  1.00000000     -0.418993880  0.20014349
## has_road_access -0.41899388      1.000000000 -0.54284615
## clusterNumber  0.20014349     -0.542846155  1.00000000
```

Omit Elevation

```
vs <- c("NST_total", "milking_cows", "sheep", "cattle", "private_count",
        #"elevation_mean",
        "has_road_access")
# Subset
vars <- subset(df, select = vs)
vars[] <- lapply(vars, as.numeric)
```

Normalization

```
max = apply(vars, 2, max, na.rm = T)
min = apply(vars, 2, min, na.rm = T)
vars_scaled = as.data.frame(scale(vars, center = min, scale = max - min))
rm(max, min)
```

```
gds <- as.matrix(daisy(vars_scaled, metric = "gower"))
```

```
no_clusters <- 6
pam_fit <- pam(gds, diss = TRUE, k = no_clusters)
df <- cbind(df, clusterNumber = pam_fit$clustering)
```

```
t <- t %>%
  mutate(Has_dairy_cows = Has_dairy_cows*100,
         Has_sheep = Has_sheep*100,
         Has_cattle = Has_cattle*100,
         Private = Private*100,
         Has_road_access = Has_road_access*100)
```

Type	N	Total NST	has dairy cows	Has sheep	Has cattle	Is private	has road access
1	2180	37.01	100.00	4.45	96.74	100.00	100.00
2	1350	73.82	100.00	6.37	96.52	0.00	96.44
3	467	36.43	0.00	5.78	94.00	0.00	80.09
4	523	46.08	86.81	12.81	98.47	71.89	0.00
5	855	20.57	0.00	2.81	95.79	100.00	90.88
6	517	33.08	0.19	91.49	6.19	71.57	22.05

Omit Sheep

```
vs <- c("NST_total", "milking_cows",
        #"sheep",
        "cattle", "private_count",
        "elevation_mean",
        "has_road_access")
# Subset
```

```
vars <- subset(df, select = vs)
vars[] <- lapply(vars, as.numeric)
```

Normalization

```
max = apply(vars, 2, max, na.rm = T)
min = apply(vars, 2, min, na.rm = T)
vars_scaled = as.data.frame(scale(vars, center = min, scale = max - min))
rm(max, min)
```

```
gds <- as.matrix(daisy(vars_scaled, metric = "gower"))
```

```
no_clusters <- 6
pam_fit <- pam(gds, diss = TRUE, k = no_clusters)
df <- cbind(df, clusterNumber = pam_fit$clustering)
```

```
t <- t %>%
  mutate(Has_dairy_cows = Has_dairy_cows*100,
         #Has_sheep = Has_sheep*100,
         Has_cattle = Has_cattle*100,
         Private = Private*100,
         Has_road_access = Has_road_access*100)
```

Type	N	Total NST	has dairy cows	Has cattle	Is private	Elevation [m]	has road access
1	2180	37.01	100.00	96.74	100.00	1372.49	100.00
2	1350	73.82	100.00	96.52	0.00	1464.16	96.44
3	467	36.43	0.00	94.00	0.00	1355.45	80.09
4	523	46.08	86.81	98.47	71.89	1800.94	0.00
5	855	20.57	0.00	95.79	100.00	1275.48	90.88
6	517	33.08	0.19	6.19	71.57	1831.42	22.05

Copyright by the authors, 2024