Clustering Template

PMS

15 May, 2023

Preliminary

Loading & Cleaning Data

```
# loading libraries
set.seed(2023)
library(cluster)
library(ggplot2)
library(factoextra)
library(clusterCrit)

# loading data
# ...

# subsampling data (if needed)
# perc = 20 #percentage of data to subsample
# subsample = (nrow(master)/100)*perc
# subsam = master[sample(nrow(master), subsample), idx]

# variables to cluster with
clust_vars = c('CSDTYPE_grouped, PMS_DBPOP, IOR_Index_of_remoteness')
```

Assumptions of the Alogrithm

text

Employment

Implementation

```
# remove NA values
# amen = master[!is.na(master$PMS_prox_idx_emp),]

# algorithm
#

# plot
#pass = list(data = <cluster_data>, cluster = <cluster_assignments>)
#fviz_cluster(pass, ellipse.type = "norm") + theme_minimal()
```

Cut-off Values

```
# need to code this still
```

Silhouette Plot

```
# sil = silhouette(<cluster_assignments>, dist(<cluster_data>))
# fviz_silhouette(sil)
```

Cluster Profiles

```
# for (k in sort(unique(<cluster_assignments>))){
#   temp = amen[<cluster_assignments> == k,]
#   print(paste('Cluster #', k))
#   print(paste('Num of DBs in cluster: ', as.character(nrow(temp))))
#   print('CSD Type:')
#   print(table(temp$CSDTYPE_grouped))
#   cat('\n DB Population: \n')
#   print(summary(temp$PMS_DBPOP))
#   cat('\n Index of Remoteness: \n')
#   print(summary(temp$IOR_Index_of_remoteness))
#   cat('\n Provinces: \n')
#   print(table(temp$PROVINCE))
#   cat('\n Amenity dense: \n')
#   print(table(temp$PMS_amenity_dense))
#   cat('\n\n\n ')
# }

# save memory
# rm(amen)
```

Pharmacy

Implementation

```
# remove NA values
# amen = master[!is.na(master$PMS_prox_idx_pharma),]

# algorithm
#

# plot
#pass = list(data = <cluster_data>, cluster = <cluster_assignments>)
#fviz_cluster(pass, ellipse.type = "norm") + theme_minimal()
```

Cut-off Values

```
# need to code this still
```

Silhouette Plot

```
# sil = silhouette(<cluster_assignments>, dist(<cluster_data>))
# fviz_silhouette(sil)
```

Cluster Profiles

```
# for (k in sort(unique(<cluster_assignments>))){
#   temp = amen[<cluster_assignments> == k,]
#   print(paste('Cluster #', k))
#   print(paste('Num of DBs in cluster: ', as.character(nrow(temp))))
#   print('CSD Type:')
#   print(table(temp$CSDTYPE_grouped))
#   cat('\n DB Population: \n')
#   print(summary(temp$PMS_DBPOP))
#   cat('\n Index of Remoteness: \n')
#   print(summary(temp$IOR_Index_of_remoteness))
#   cat('\n Provinces: \n')
#   print(table(temp$PROVINCE))
#   cat('\n Amenity dense: \n')
#   print(table(temp$PMS_amenity_dense))
#   cat('\n\n\n ')
# }

# save memory
# rm(amen)
```

Childcare

Implementation

```
# remove NA values
# amen = master[!is.na(master$PMS_prox_idx_childcare),]

# algorithm
#

# plot
#pass = list(data = <cluster_data>, cluster = <cluster_assignments>)
#fviz_cluster(pass, ellipse.type = "norm") + theme_minimal()
```

Cut-off Values

```
# need to code this still
```

Silhouette Plot

```
# sil = silhouette(<cluster_assignments>, dist(<cluster_data>))
# fviz_silhouette(sil)
```

Cluster Profiles

```
# for (k in sort(unique(<cluster_assignments>))){
#   temp = amen[<cluster_assignments> == k,]
#   print(paste('Cluster #', k))
#   print(paste('Num of DBs in cluster: ', as.character(nrow(temp))))
#   print('CSD Type:')
#   print(table(temp$CSDTYPE_grouped))
#   cat('\n DB Population: \n')
#   print(summary(temp$PMS_DBPOP))
#   cat('\n Index of Remoteness: \n')
#   print(summary(temp$IOR_Index_of_remoteness))
#   cat('\n Provinces: \n')
#   print(table(temp$PROVINCE))
#   cat('\n Amenity dense: \n')
#   print(table(temp$PMS_amenity_dense))
#   cat('\n\n\n ')
# }

# save memory
# rm(amen)
```

Healthcare

Implementation

```
# remove NA values
# amen = master[!is.na(master$PMS_prox_idx_health),]

# algorithm
#

# plot
#pass = list(data = <cluster_data>, cluster = <cluster_assignments>)
#fviz_cluster(pass, ellipse.type = "norm") + theme_minimal()
```

Cut-off Values

```
# need to code this still
```

Silhouette Plot

```
# sil = silhouette(<cluster_assignments>, dist(<cluster_data>))
# fviz_silhouette(sil)
```

Cluster Profiles

```
# for (k in sort(unique(<cluster_assignments>))){
#   temp = amen[<cluster_assignments> == k,]
#   print(paste('Cluster #', k))
#   print(paste('Num of DBs in cluster: ', as.character(nrow(temp))))
#   print('CSD Type:')
#   print(table(temp$CSDTYPE_grouped))
#   cat('\n DB Population: \n')
#   print(summary(temp$PMS_DBPOP))
#   cat('\n Index of Remoteness: \n')
#   print(summary(temp$IOR_Index_of_remoteness))
#   cat('\n Provinces: \n')
#   print(table(temp$PROVINCE))
#   cat('\n Amenity dense: \n')
#   print(table(temp$PMS_amenity_dense))
#   cat('\n\n\n ')
# }

# save memory
# rm(amen)
```

Grocery

Implementation

```
# remove NA values
# amen = master[!is.na(master$PMS_prox_idx_grocery),]

# algorithm
#

# plot
#pass = list(data = <cluster_data>, cluster = <cluster_assignments>)
#fviz_cluster(pass, ellipse.type = "norm") + theme_minimal()
```

Cut-off Values

```
# need to code this still
```

Silhouette Plot

```
# sil = silhouette(<cluster_assignments>, dist(<cluster_data>))
# fviz_silhouette(sil)
```

Cluster Profiles

```
# for (k in sort(unique(<cluster_assignments>))){
#   temp = amen[<cluster_assignments> == k,]
#   print(paste('Cluster #', k))
#   print(paste('Num of DBs in cluster: ', as.character(nrow(temp))))
#   print('CSD Type:')
#   print(table(temp$CSDTYPE_grouped))
#   cat('\n DB Population: \n')
#   print(summary(temp$PMS_DBPOP))
#   cat('\n Index of Remoteness: \n')
#   print(summary(temp$IOR_Index_of_remoteness))
#   cat('\n Provinces: \n')
#   print(table(temp$PROVINCE))
#   cat('\n Amenity dense: \n')
#   print(table(temp$PMS_amenity_dense))
#   cat('\n\n\n ')
# }

# save memory
# rm(amen)
```

Primary Education

Implementation

```
# remove NA values
# amen = master[!is.na(master$PMS_prox_idx_educpri),]

# algorithm
#

# plot
#pass = list(data = <cluster_data>, cluster = <cluster_assignments>)
#fviz_cluster(pass, ellipse.type = "norm") + theme_minimal()
```

Cut-off Values

```
# need to code this still
```

Silhouette Plot

```
# sil = silhouette(<cluster_assignments>, dist(<cluster_data>))
# fviz_silhouette(sil)
```

Cluster Profiles

```
# for (k in sort(unique(<cluster_assignments>))){
#   temp = amen[<cluster_assignments> == k,]
#   print(paste('Cluster #', k))
#   print(paste('Num of DBs in cluster: ', as.character(nrow(temp))))
#   print('CSD Type:')
#   print(table(temp$CSDTYPE_grouped))
#   cat('\n DB Population: \n')
#   print(summary(temp$PMS_DBPOP))
#   cat('\n Index of Remoteness: \n')
#   print(summary(temp$IOR_Index_of_remoteness))
#   cat('\n Provinces: \n')
#   print(table(temp$PROVINCE))
#   cat('\n Amenity dense: \n')
#   print(table(temp$PMS_amenity_dense))
#   cat('\n\n\n ')
# }

# save memory
# rm(amen)
```

Secondary Education

Implementation

```
# remove NA values
# amen = master[!is.na(master$PMS_prox_idx_educsec),]

# algorithm
#

# plot
#pass = list(data = <cluster_data>, cluster = <cluster_assignments>)
#fviz_cluster(pass, ellipse.type = "norm") + theme_minimal()
```

Cut-off Values

```
# need to code this still
```

Silhouette Plot

```
# sil = silhouette(<cluster_assignments>, dist(<cluster_data>))
# fviz_silhouette(sil)
```

Cluster Profiles

```
# for (k in sort(unique(<cluster_assignments>))){
#   temp = amen[<cluster_assignments> == k,]
#   print(paste('Cluster #', k))
#   print(paste('Num of DBs in cluster: ', as.character(nrow(temp))))
#   print('CSD Type:')
#   print(table(temp$CSDTYPE_grouped))
#   cat('\n DB Population: \n')
#   print(summary(temp$PMS_DBPOP))
#   cat('\n Index of Remoteness: \n')
#   print(summary(temp$IOR_Index_of_remoteness))
#   cat('\n Provinces: \n')
#   print(table(temp$PROVINCE))
#   cat('\n Amenity dense: \n')
#   print(table(temp$PMS_amenity_dense))
#   cat('\n\n\n ')
# }

# save memory
# rm(amen)
```

Library

Implementation

```
# remove NA values
# amen = master[!is.na(master$PMS_prox_idx_lib),]

# algorithm
#

# plot
#pass = list(data = <cluster_data>, cluster = <cluster_assignments>)
#fviz_cluster(pass, ellipse.type = "norm") + theme_minimal()
```

Cut-off Values

```
# need to code this still
```

Silhouette Plot

```
# sil = silhouette(<cluster_assignments>, dist(<cluster_data>))
# fviz_silhouette(sil)
```

Cluster Profiles

```
# for (k in sort(unique(<cluster_assignments>))){
#   temp = amen[<cluster_assignments> == k,]
#   print(paste('Cluster #', k))
#   print(paste('Num of DBs in cluster: ', as.character(nrow(temp))))
#   print('CSD Type:')
#   print(table(temp$CSDTYPE_grouped))
#   cat('\n DB Population: \n')
#   print(summary(temp$PMS_DBPOP))
#   cat('\n Index of Remoteness: \n')
#   print(summary(temp$IOR_Index_of_remoteness))
#   cat('\n Provinces: \n')
#   print(table(temp$PROVINCE))
#   cat('\n Amenity dense: \n')
#   print(table(temp$PMS_amenity_dense))
#   cat('\n\n\n ')
# }

# save memory
# rm(amen)
```

Parks

Implementation

```
# remove NA values
# amen = master[!is.na(master$PMS_prox_idx_parks),]

# algorithm
#

# plot
#pass = list(data = <cluster_data>, cluster = <cluster_assignments>)
#fviz_cluster(pass, ellipse.type = "norm") + theme_minimal()
```

Cut-off Values

```
# need to code this still
```

Silhouette Plot

```
# sil = silhouette(<cluster_assignments>, dist(<cluster_data>))
# fviz_silhouette(sil)
```

Cluster Profiles

```
# for (k in sort(unique(<cluster_assignments>))){
#   temp = amen[<cluster_assignments> == k,]
#   print(paste('Cluster #', k))
#   print(paste('Num of DBs in cluster: ', as.character(nrow(temp))))
#   print('CSD Type:')
#   print(table(temp$CSDTYPE_grouped))
#   cat('\n DB Population: \n')
#   print(summary(temp$PMS_DBPOP))
#   cat('\n Index of Remoteness: \n')
#   print(summary(temp$IOR_Index_of_remoteness))
#   cat('\n Provinces: \n')
#   print(table(temp$PROVINCE))
#   cat('\n Amenity dense: \n')
#   print(table(temp$PMS_amenity_dense))
#   cat('\n\n\n ')
# }

# save memory
# rm(amen)
```

Transit

Implementation

```
# remove NA values
# amen = master[!is.na(master$PMS_prox_idx_transit),]

# algorithm
#

# plot
#pass = list(data = <cluster_data>, cluster = <cluster_assignments>)
#fviz_cluster(pass, ellipse.type = "norm") + theme_minimal()
```

Cut-off Values

```
# need to code this still
```

Silhouette Plot

```
# sil = silhouette(<cluster_assignments>, dist(<cluster_data>))
# fviz_silhouette(sil)
```

Cluster Profiles

```
# for (k in sort(unique(<cluster_assignments>))){
#   temp = amen[<cluster_assignments> == k,]
#   print(paste('Cluster #', k))
#   print(paste('Num of DBs in cluster: ', as.character(nrow(temp))))
#   print('CSD Type:')
#   print(table(temp$CSDTYPE_grouped))
#   cat('\n DB Population: \n')
#   print(summary(temp$PMS_DBPOP))
#   cat('\n Index of Remoteness: \n')
#   print(summary(temp$IOR_Index_of_remoteness))
#   cat('\n Provinces: \n')
#   print(table(temp$PROVINCE))
#   cat('\n Amenity dense: \n')
#   print(table(temp$PMS_amenity_dense))
#   cat('\n\n\n ')
# }

# save memory
# rm(amen)
```

Conclusion

text