Aggretation1

Xin

2024-05-15

```
library(electBook)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(proxy)
```

```
##
## Attaching package: 'proxy'
```

```
## The following objects are masked from 'package:stats':
##
##     as.dist, dist
```

```
## The following object is masked from 'package:base':
##
##     as.matrix
```

```
library(tibble)
library(ggplot2)
```

```
data(Irish)
col_zero_counts <- colSums(Irish$indCons == 0)
cols_to_remove <- which(col_zero_counts > 30*48)
df <- Irish$indCons
df$date <- date(Irish$extra$dateTime)
df <- df[,-cols_to_remove]
```

```
df$date <- as.Date(df$date)

# Gather the data into long format
df_long <- df %>%
  pivot_longer(cols = -date, names_to = "household_id", values_to = "demand")

# Ensure the date column is of Date type in df_long
df_long$date <- as.Date(df_long$date)

# Group by household and date, then summarize to create daily profiles
daily_profiles <- df_long %>%
  group_by(household_id, date) %>%
  summarise(daily_demand = sum(demand, na.rm = TRUE)) %>%
  pivot_wider(names_from = date, values_from = daily_demand) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'household_id'. You can override using the
## `.groups` argument.
```

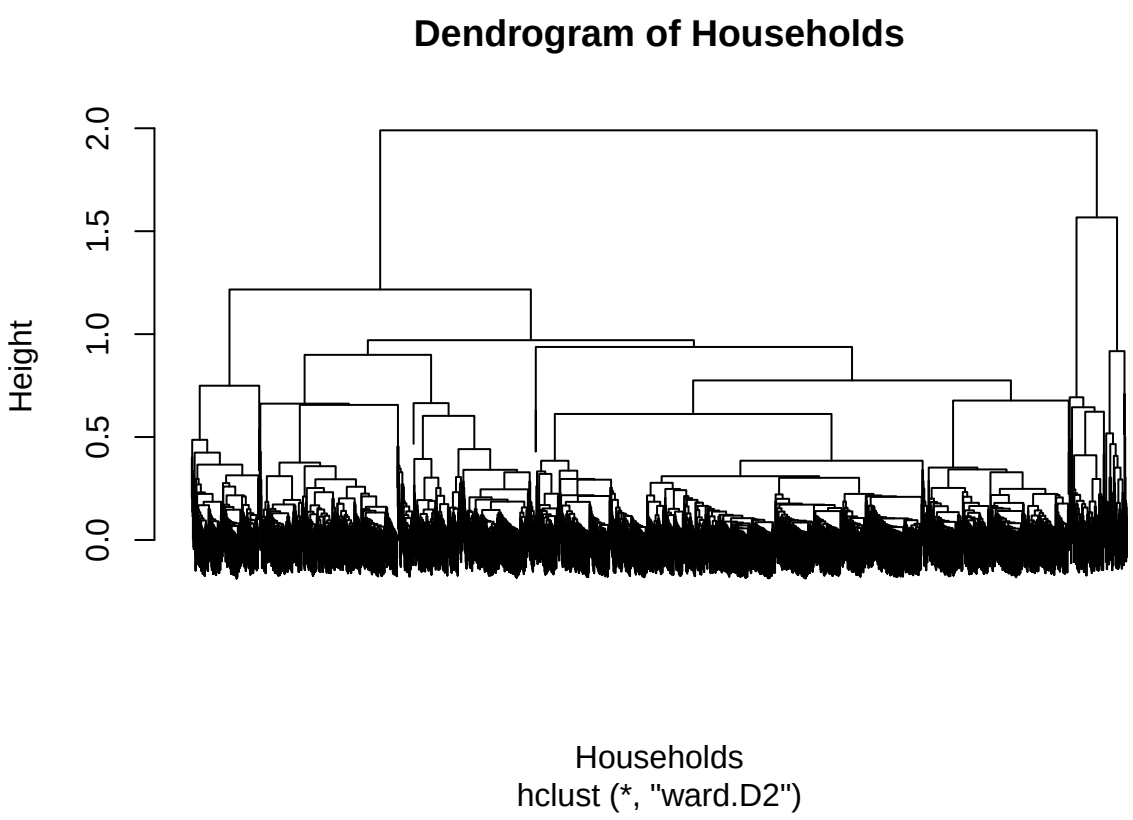
```
# Replace NA values with zeros (assuming no demand means 0 demand)
daily_profiles[is.na(daily_profiles)] <- 0

# Compute the cosine similarity matrix
compute_cosine_similarity_matrix <- function(data) {
  data_matrix <- as.matrix(data[-1])  # Remove the household_id column
  similarity_matrix <- proxy::simil(data_matrix, method = "cosine")
  dist_matrix <- 1 - similarity_matrix
  return(as.matrix(dist_matrix))
}

cosine_distances <- compute_cosine_similarity_matrix(daily_profiles)

# Hierarchical clustering
hc <- hclust(as.dist(cosine_distances), method = "ward.D2")

# Plot the dendrogram
plot(hc, labels = FALSE, main = "Dendrogram of Households", xlab = "Households", ylab = "Height")
```




```
# Create clusters
clusters <- cutree(hc, k = 5)
daily_profiles$cluster <- clusters
# Summarize the number of households in each cluster
cluster_summary <- daily_profiles %>%
  group_by(cluster) %>%
  summarise(num_households = n())

# Display the summary
print(cluster_summary)
```

```
## # A tibble: 5 x 2
##   cluster num_households
##     <int>          <int>
## 1       1           1509
## 2       2            780
## 3       3            192
## 4       4            101
## 5       5             63
```

```
# Reshape daily_profiles back to long format
daily_profiles_long <- daily_profiles %>%
  pivot_longer(cols = -c(household_id, cluster), names_to = "date", values_to = "daily_demand")

# Ensure the date column is of Date type in daily_profiles_long
```

```r
daily_profiles_long$date <- as.Date(daily_profiles_long$date)

# Join cluster information back to the original dataframe
df_with_clusters <- df_long %>%
  left_join(daily_profiles_long, by = c("household_id", "date"))

# Analyze cluster characteristics
cluster_analysis <- df_with_clusters %>%
  group_by(cluster) %>%
  summarise(
    average_demand = mean(daily_demand, na.rm = TRUE)
  )

print(cluster_analysis)
```

```
## # A tibble: 5 x 2
##   cluster average_demand
##     <int>          <dbl>
## 1       1           24.6
## 2       2           24.0
## 3       3           23.1
## 4       4           23.5
## 5       5           16.6
```

```r
df_t <- as.data.frame(t(df[,-ncol(df)]))

# Step 2: Add the clusters as a new column to the transposed data frame
df_t$cluster <- clusters

# Step 3: Group by cluster and calculate the mean for each row within each cluster
mean_by_cluster <- df_t %>%
  group_by(cluster) %>%
  summarise(across(everything(), mean, na.rm = TRUE))
```

```
## Warning: There was 1 warning in `summarise()`.
## i In argument: `across(everything(), mean, na.rm = TRUE)`.
## i In group 1: `cluster = 1`.
## Caused by warning:
## ! The `...` argument of `across()` is deprecated as of dplyr 1.1.0.
## Supply arguments directly to `.fns` through an anonymous function instead.
##
##   # Previously
##   across(a:b, mean, na.rm = TRUE)
##
##   # Now
##   across(a:b, \(x) mean(x, na.rm = TRUE))
```

```r
mean_by_cluster <- as.data.frame(mean_by_cluster[,-1])
rownames(mean_by_cluster) <- c("Cluster 1","Cluster 2","Cluster 3","Cluster 4","Cluster 5")
```

```r
df0 <- Irish$extra
df0 <- df0 %>% mutate(dow = ifelse(dow %in% c("Sat", "Sun"), "True", "False")) %>% select(-c(holy,time,
df0 <-t(df0)
```

```r
colnames(df0) <- colnames(mean_by_cluster)
mean_by_cluster <- rbind(mean_by_cluster,df0)
```

```r
# Save as CSV file
write.csv(mean_by_cluster, file = "AggregatedData.csv")
```