Abstract

Keywords:

0.1 Materials and methods

0.1.1 Simulations

To test clustering algorithm solutions against ground truth cluster assignments, we generate simulations to represent important problems that could be encountered in electricity data contexts. The code to generate these simulations are included in the Supplementary paper. When we move from the linear to the cyclic world of temporal granularities, we can see patterns across different categories of the granularities which gets lost in the linear representation. Populations are modeled by a collection of these cyclic granularities. Each cyclic granularity might or might not have patterns across its categories. Each cluster is characterized by similar patterns across one or more of these cyclic granularities.

A small example is given to setup the problem.

We constructed the simulation parameters to represent common patterns in electricity data. 50, 200 and 500 time series were chosen for different simulation designs with different granularities and patterns changing across different granularities. The data type is fixed to be "continuous".We generated independent replications of all combinations of the simulation parameters.

Consider a continuous time series variable y of length T indexed by $0, 1, \dots T-1$. Three circular granularities $g1$, $g2$ and $g3$ are considered with 2, 3 and 5 levels respectively. Categories of g1, g2 and g3 are represented by $0, 1$, $0, 1, 2$ and $0, 1, 2, 3, 4$. These categories could be integers or some more meaningful labels. For example, the granularity "day-of-week" could be either represented by $0, 1, 2, \dots, 6$ or $Mon, Tue, \dots, Sun$.

Consider a case where distribution of y would vary across levels of $g2$ for all designs, across levels of $g1$ for few designs and $g3$ does not change across designs. Figure **??** shows the linear and cyclic representation of y. The first panel shows raw plot of y in a linear scale and the second panel shows distribution of y across cyclic granularities namely $g1$, $g2$ and $g3$. As could be seen from the plots, it is impossible to decipher from the raw time plot that the time series variable shows such pattern across different granularities.

A subset of many possible designs are shown in Figure **??**. For the parameter space (XXX unique combinations shown in table YYY), 100, 500 independent replications of all possible combination of simulation parameters were generated. The clustering methodolo-

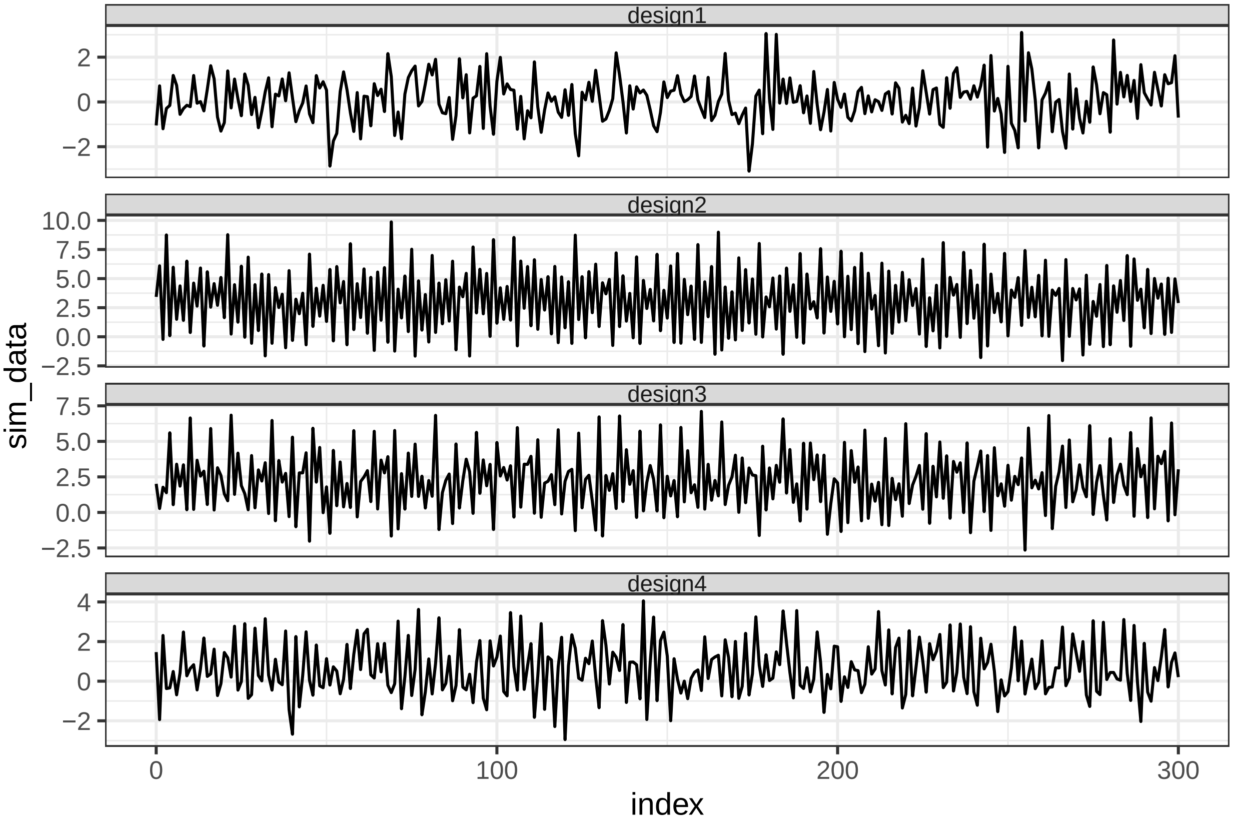


Figure 1: The linear and cyclic representation of the time series variable y. It is not possible to comprehend these patterns across cyclic granularities g1, g2 and g3 or group similar series just by looking at the linear plots.
(#fig:plot-linear -1)

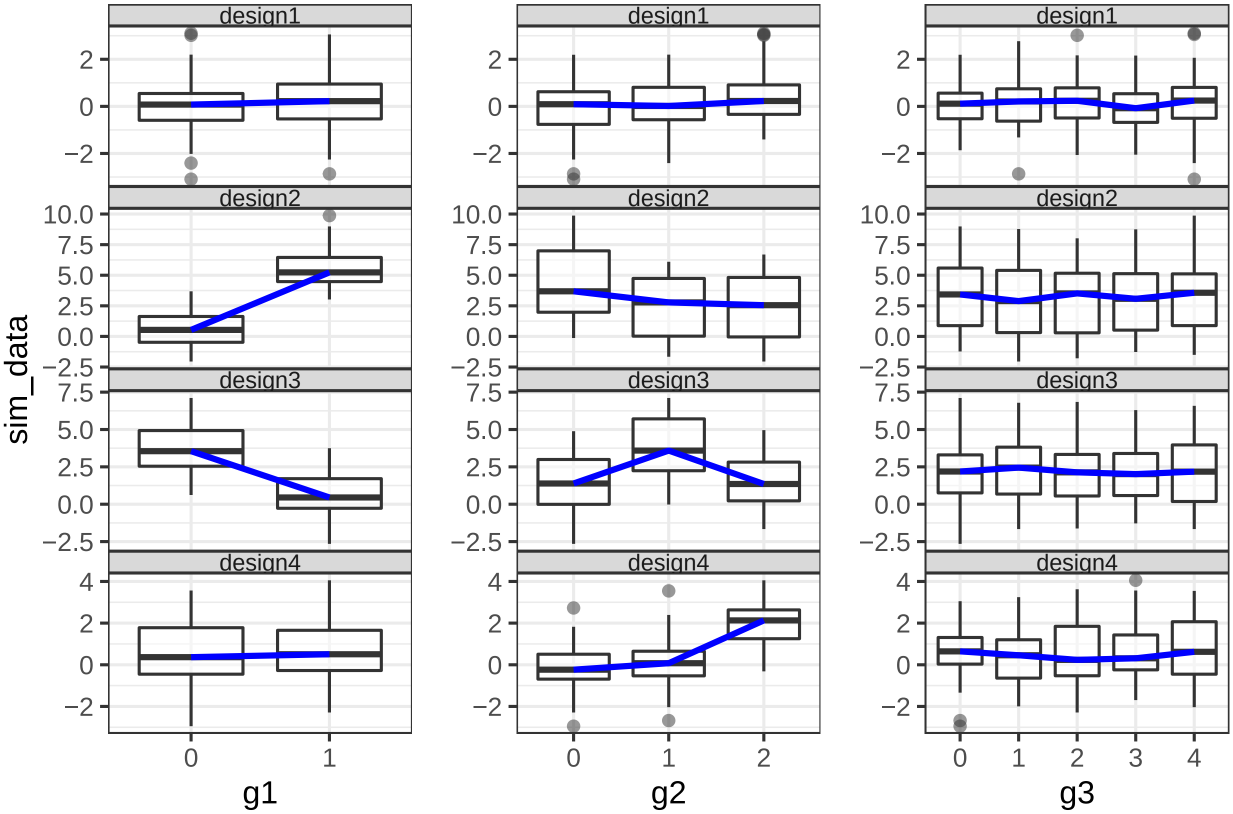


Figure 2: The linear and cyclic representation of the time series variable y. It is not possible to comprehend these patterns across cyclic granularities g1, g2 and g3 or group similar series just by looking at the linear plots.

(#fig:plot-linear -2)

gies were run all these unique combinations and subsets of these to verify if the methodologies work as expected.

0.1.2 Clustering algorithms

We choose the algorithms with most common use in smart meter clustering. . First, we used agglomerative hierarchical clustering with Ward's criterion (HC), a dominant approach for smart meter clustering. Second, we also used k-means clustering. For each clustering algorithm, we assumed the number of clusters was known, recovering the number of clusters given from the simulation parameters.

Granularity type	# Significant	# Replications
Individual # obs: 300, 500, 2000 # clusters: 6/7	1/2/3	25, 100, 200
Interaction # obs: 500, 2000 # clusters: 4	1/2	25, 100, 200

0.1.3 Distance metrics

A single or pair of granularities together (change names) The methodology can be summarized in the following steps:

- *Pre-processing step*

Robust scaling method or NQT used for each customer.

- *NQT*
- *Treatment to outliers*
- *Handling trend, seasonality, non-stationarity and auto-correlation* Trend and seasonality are common features of time series, and it is natural to characterize a time series by its degree of trend and seasonality. By considering the probability distributions through the use of $wpd_{norm_{s,t}}(A, B)$, these features of the time series are lost and hence there is no need to de-trend or de-seasonalize the data before performing the clustering algorithm. No need to exclude holiday or weekend patterns.

Many granularities together (change names) The methodology can be summarized in the following steps:

1. Compute quantiles of distributions across each category of the cyclic granularity
2. Compute JS distance between households for each each category of the cyclic granularity
3. Total distance between households computed as sum of JS distances for all hours
4. Cluster using this distance with hierarchical clustering algorithm (method "Ward.D")

Pro:

- distance metric makes sense to group different shapes together
- simulation results look great on typical designs

Cons:

- Can only take one granularity at once
- Clustering a big blob of points together whereas the aim is to groups these big blob into smaller ones

Multiple-granularities *Description:*

Choose all significant granularities and compute wpd for all these granularities for all customers. Distance between customers is taken as the euclidean distances between them with the granularities being the variables and wpd being the value under each variable for which Euclidean distance needs to be measured.

Pro:

- Can only take many granularities at once - can apply variable selection PCP and other interesting clustering techniques - simulation results look great on typical designs - splitting the data into similar sized groups

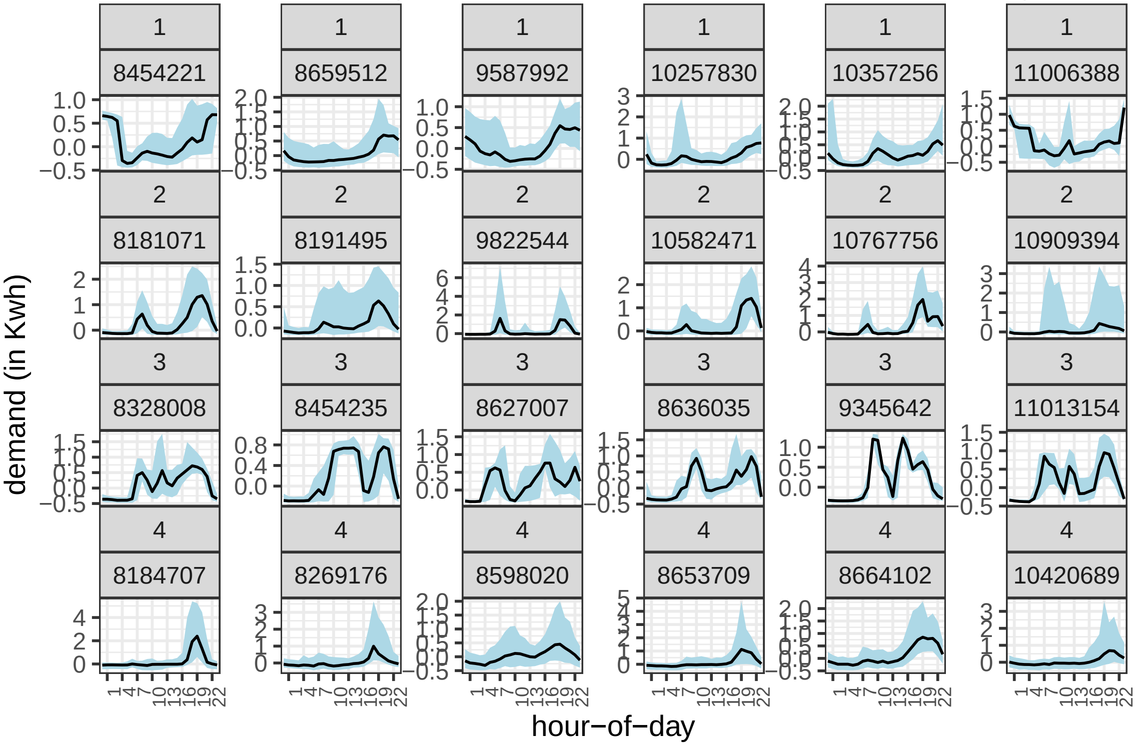
Cons:

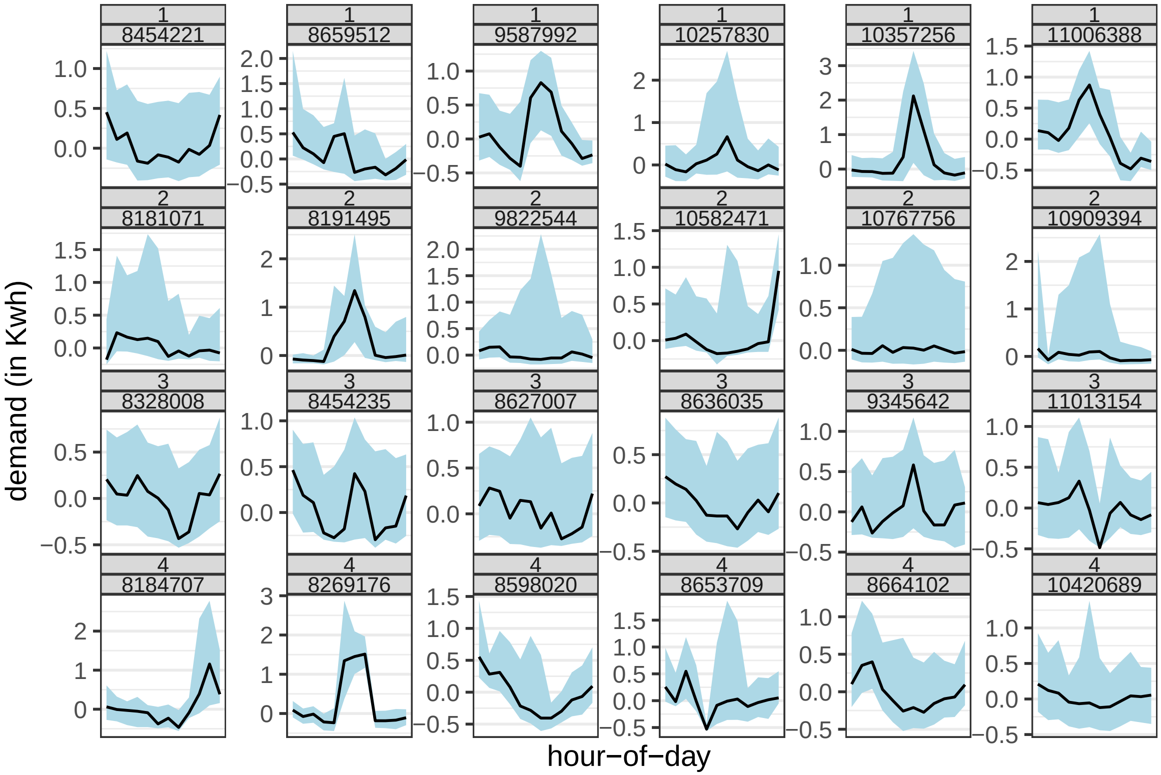
- distance metric does not make sense to split the data into similar shaped clusters

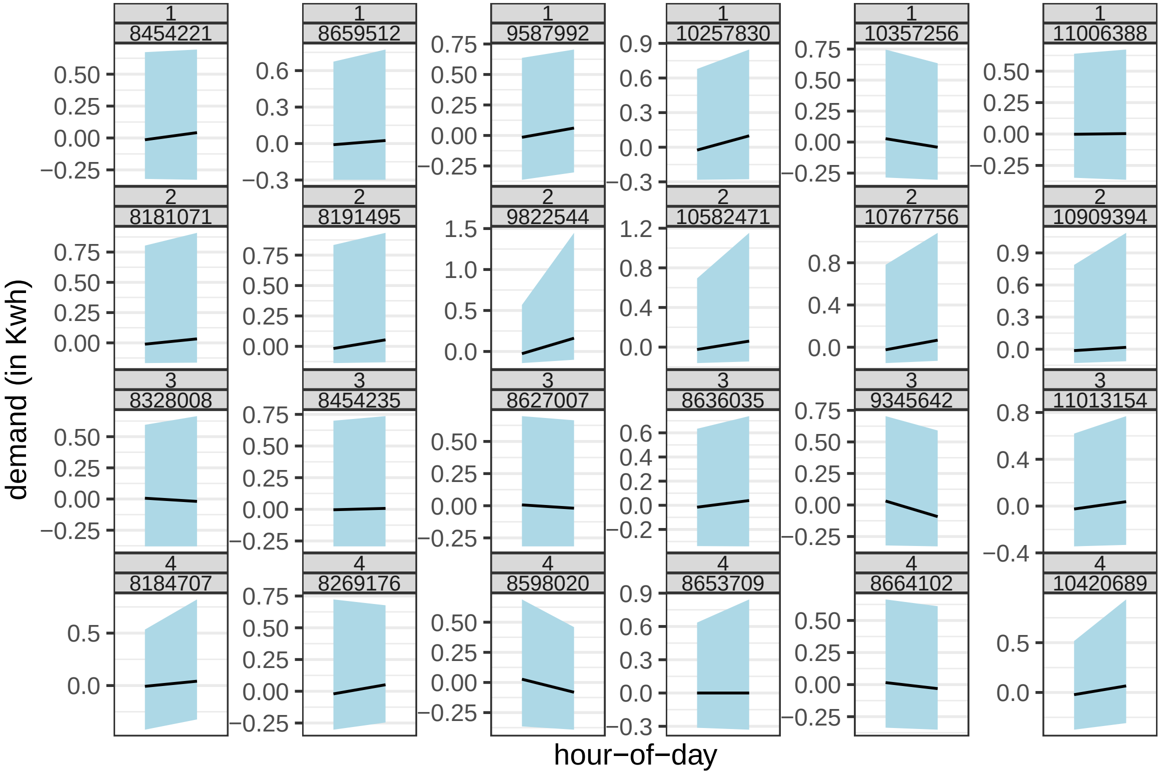
0.1.4 Prototype application

A clean data set is obtained by carefully choosing customers which shows similar shapes across one or more cyclic granularity. Since this is unlabeled data, there is no way to do external validation of our methodologies. Thus, we chose this way to see how well our methodology works in a cleaner data set as this one.

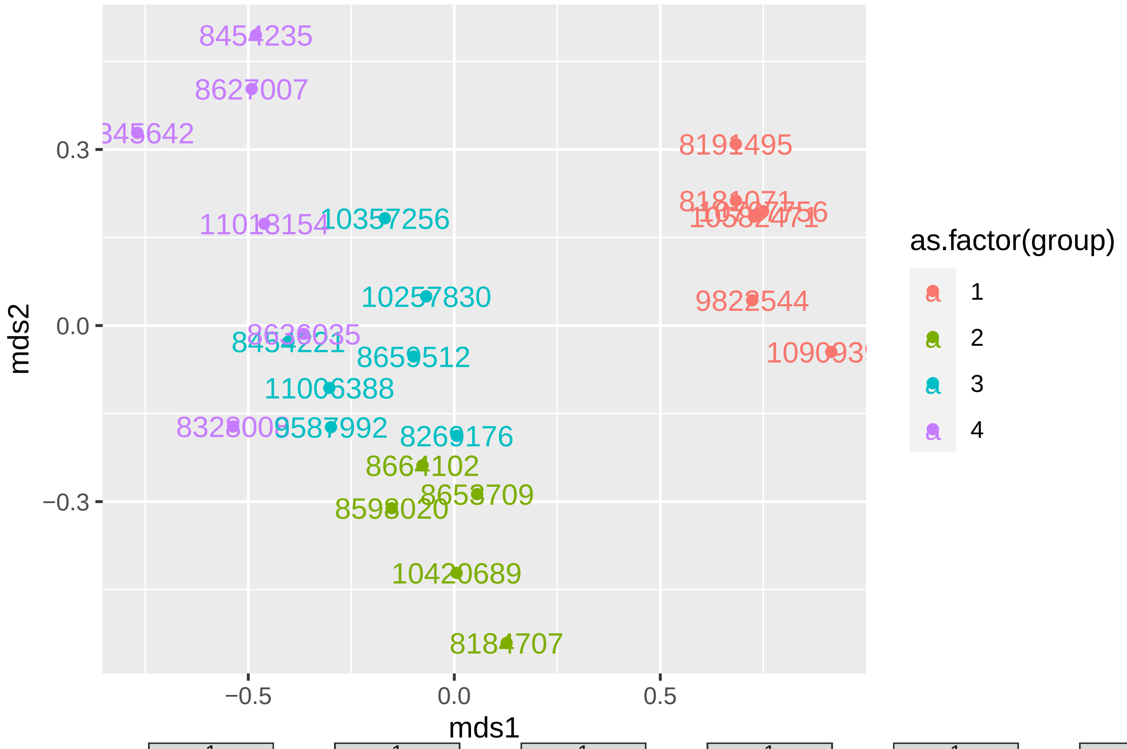
Fig **??**, **??** and **??** shows the distribution of 24 customers across hour-of-day, month-of-year and wknd-wday respectively. Every row in **??** shows different shapes across hour-of-day and across columns show similar shapes for each row. We use our methodology to see if the customers are allocated to same group have similar shapes across one or more granularity.

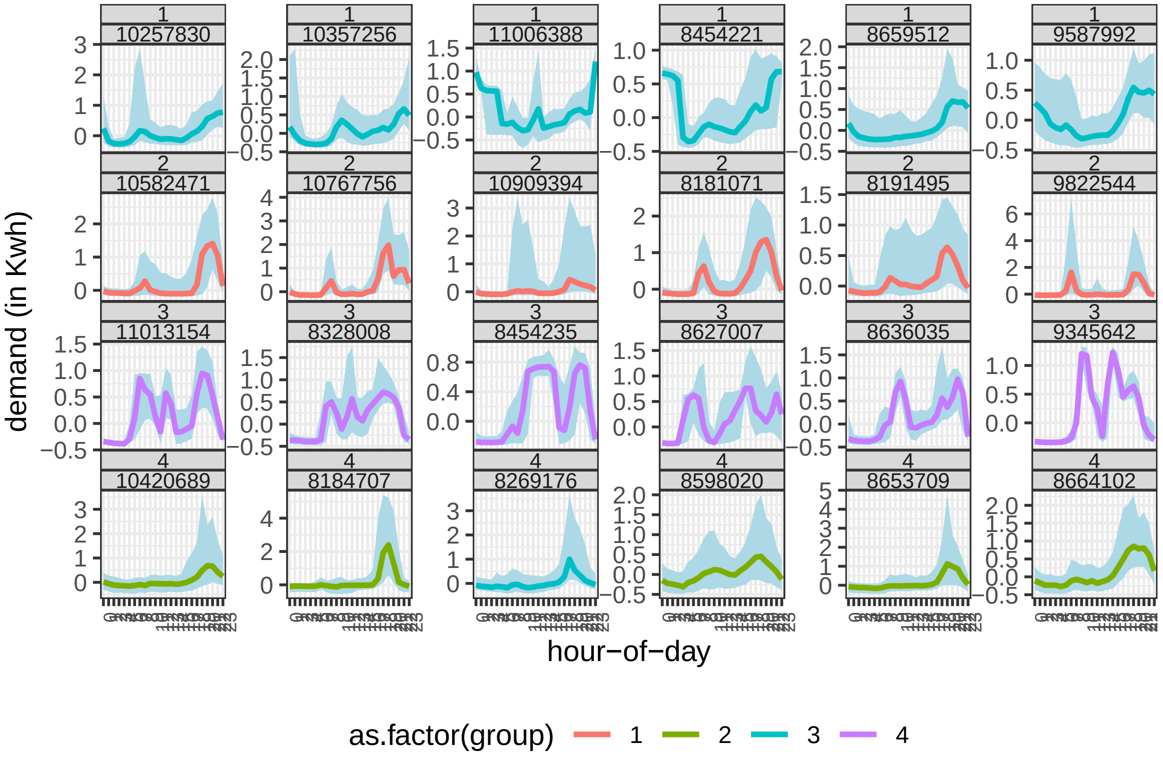


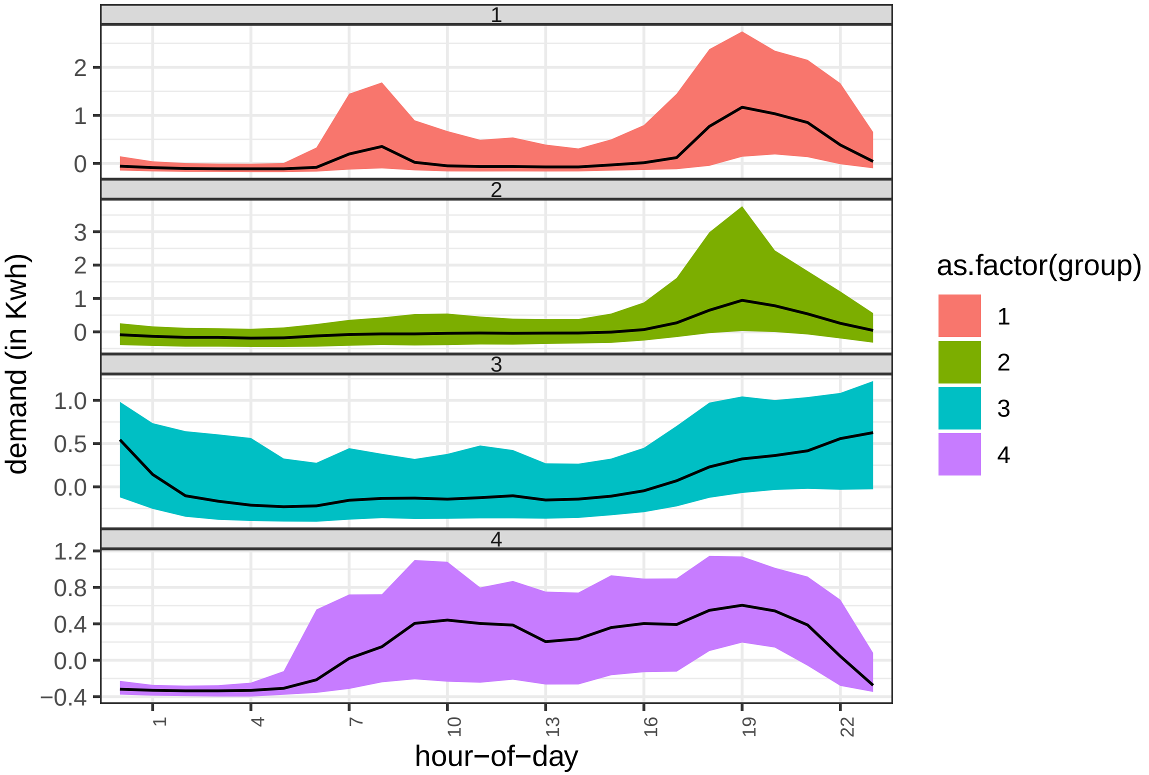

demand (in Kwh)
hour−of−day
1 8454221
1 8659512
1 9587992
1 10257830
1 10357256
1 11006388
2 8181071
2 8191495
2 9822544
2 10582471
2 10767756
2 10909394
3 8328008
3 8454235
3 8627007
3 8636035
3 9345642
3 11013154
4 8184707
4 8269176
4 8598020
4 8653709
4 8664102
4 10420689

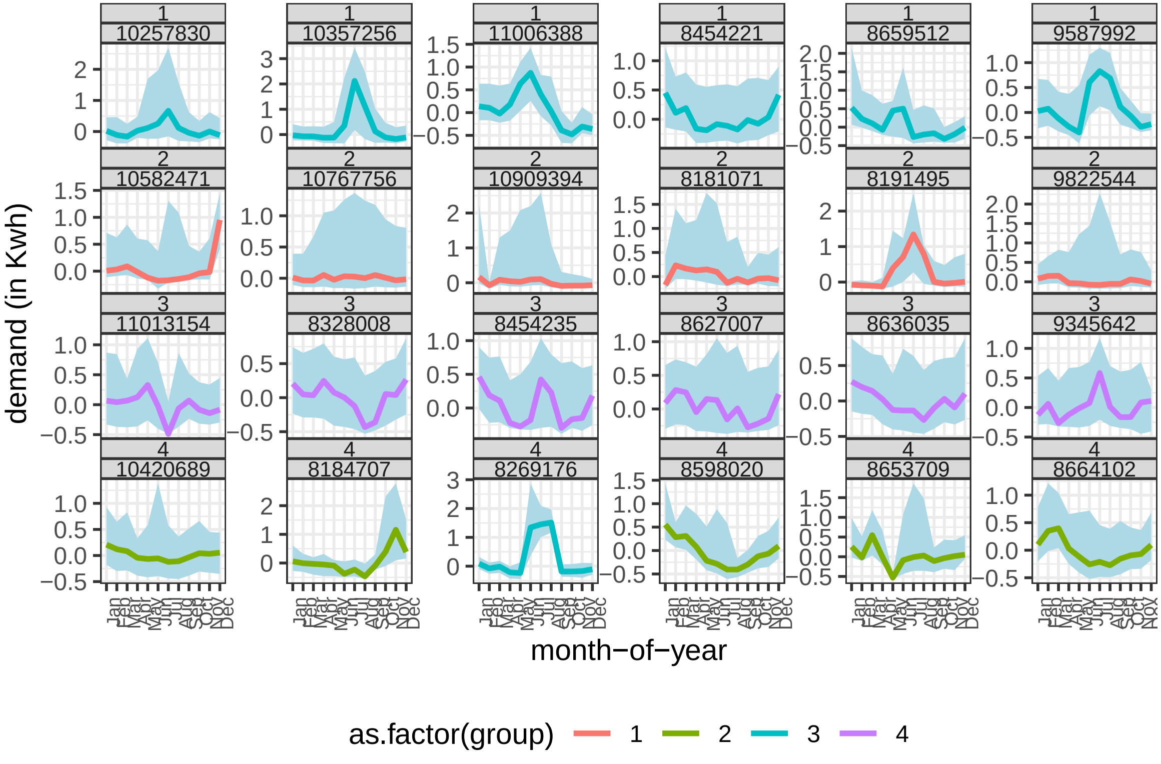

demand (in Kwh)
hour−of−day
1 8454221
1 8659512
1 9587992
1 10257830
1 10357256
1 11006388
2 8181071
2 8191495
2 9822544
2 10582471
2 10767756
2 10909394
3 8328008
3 8454235
3 8627007
3 8636035
3 9345642
3 11013154
4 8184707
4 8269176
4 8598020
4 8653709
4 8664102
4 10420689

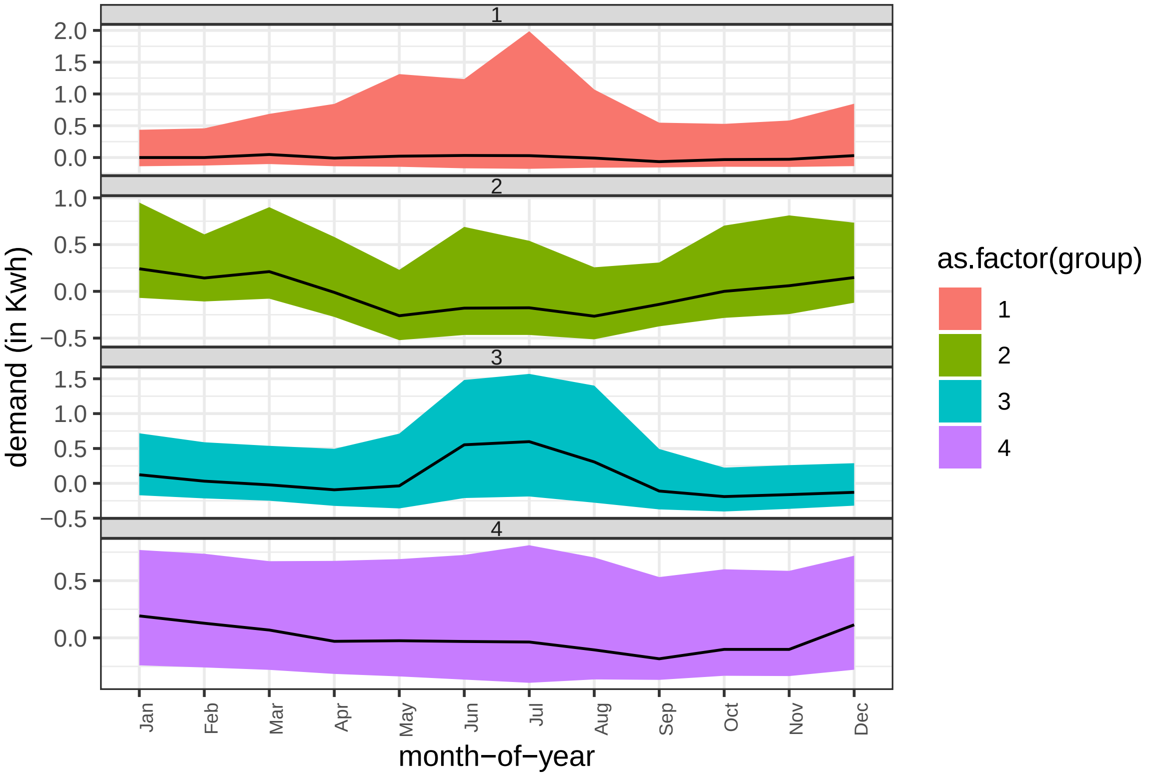
1 Clustering by method 1

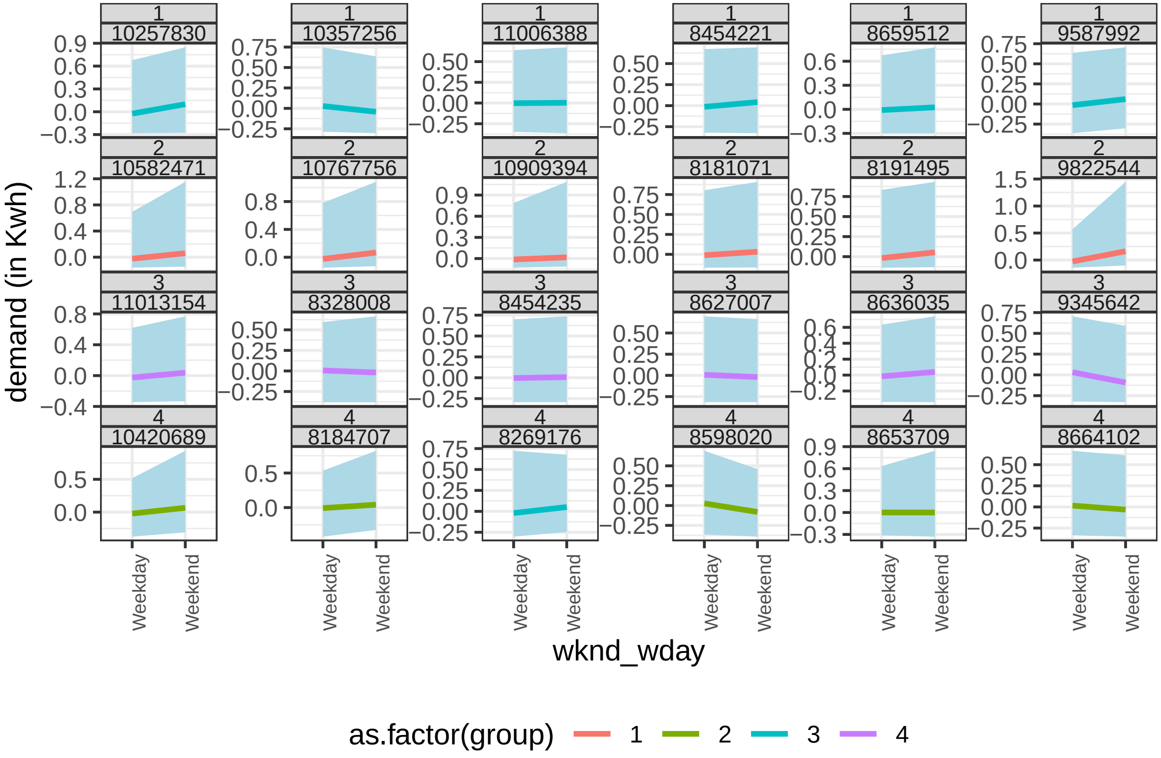


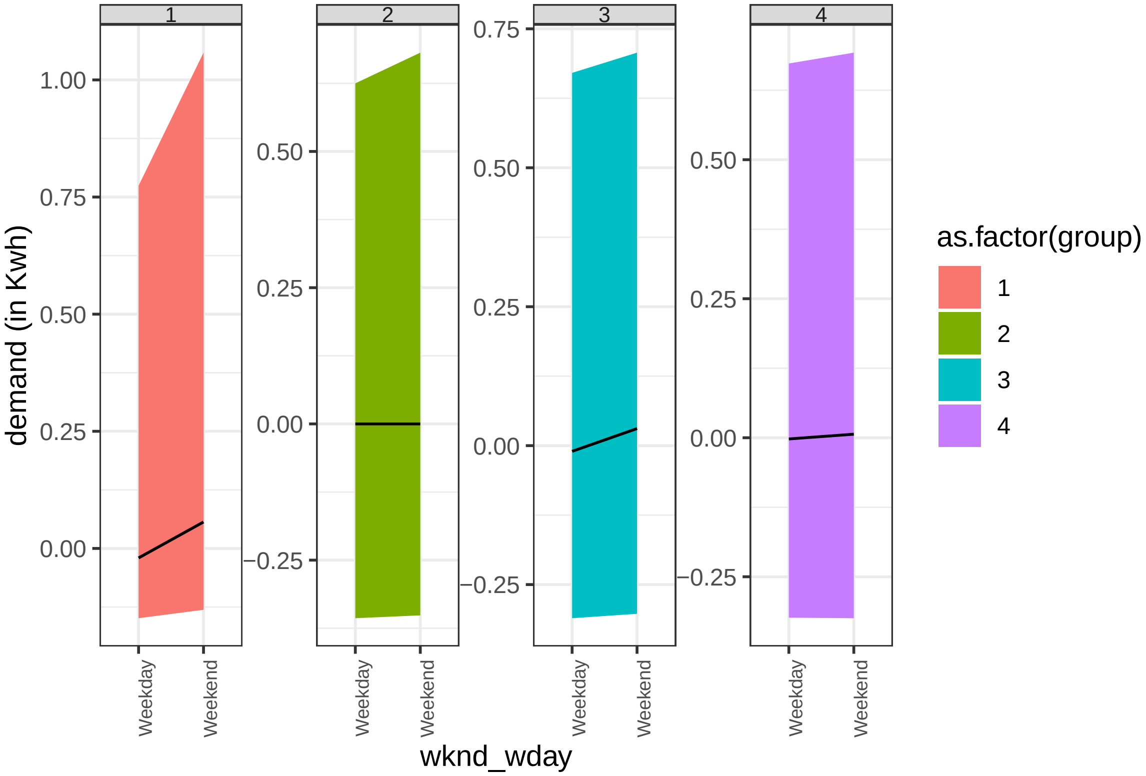



1
2
3
4
demand (in Kwh)
hour−of−day
1
4
7
10
13
16
19
22
2
1
0
3
2
1
0
1.0
0.5
0.0
1.2
0.8
0.4
0.0
−0.4
as.factor(group)
1
2
3
4

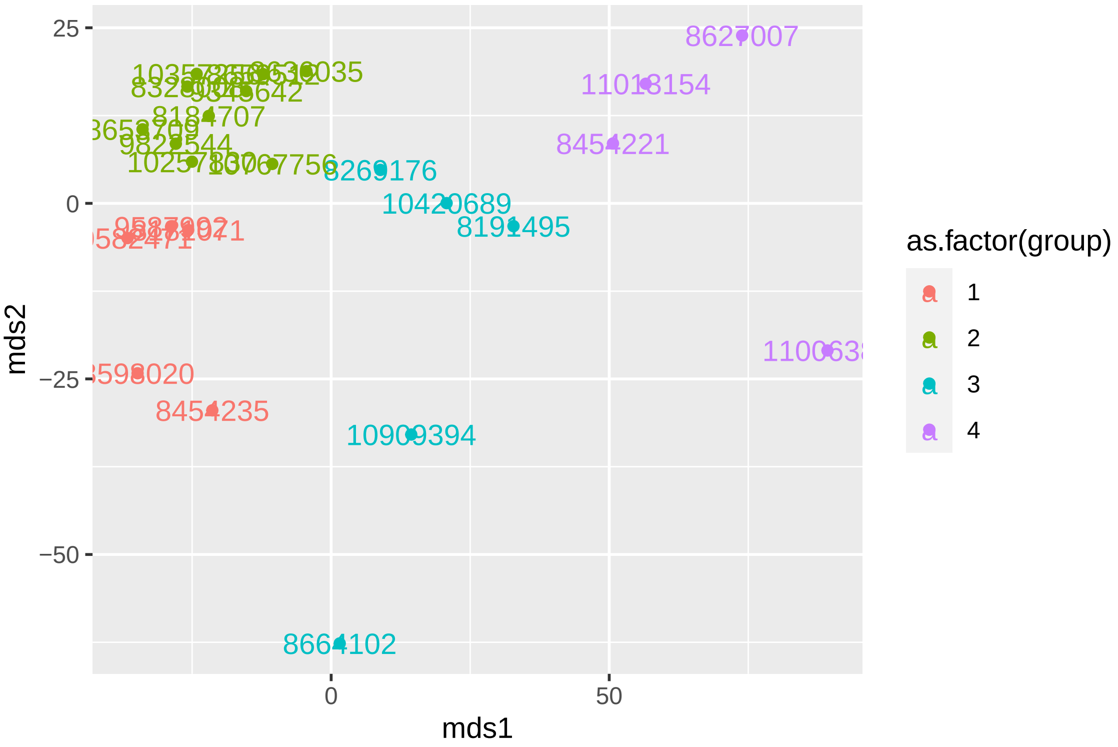

1 10257830
1 10357256
1 11006388
1 8454221
1 8659512
1 9587992
2 10582471
2 10767756
2 10909394
2 8181071
2 8191495
2 9822544
3 11013154
3 8328008
3 8454235
3 8627007
3 8636035
3 9345642
4 10420689
4 8184707
4 8269176
4 8598020
4 8653709
4 8664102
demand (in Kwh)
month−of−year
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
as.factor(group)
1
2
3
4

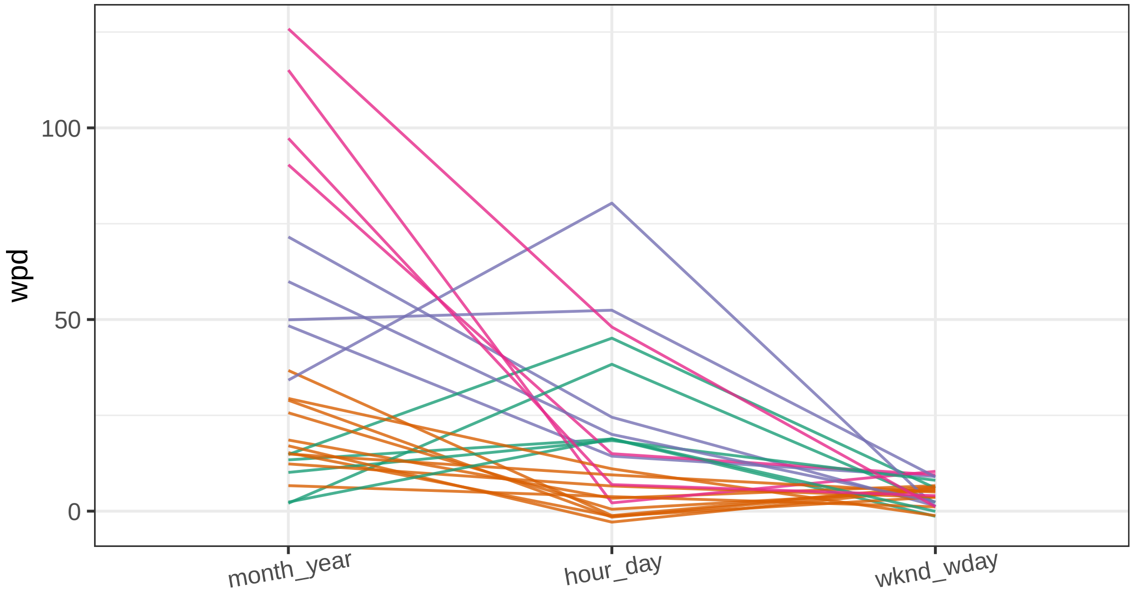

1
2
3
4
2.0
1.5
1.0
0.5
0.0
1.0
0.5
0.0
−0.5
1.5
1.0
0.5
0.0
−0.5
0.5
0.0
demand (in Kwh)
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
month−of−year
as.factor(group)
1
2
3
4

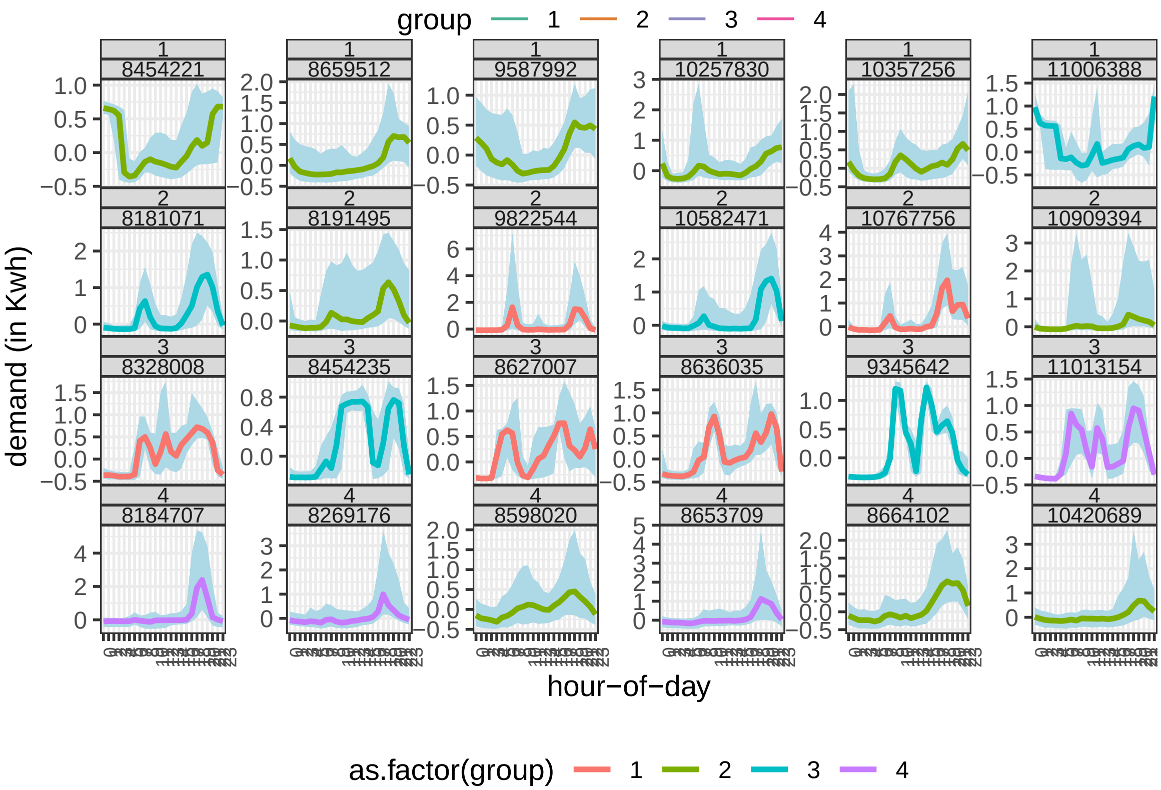

1
10257830
1
10357256
1
11006388
1
8454221
1
8659512
1
9587992
2
10582471
2
10767756
2
10909394
2
8181071
2
8191495
2
9822544
3
11013154
3
8328008
3
8454235
3
8627007
3
8636035
3
9345642
4
10420689
4
8184707
4
8269176
4
8598020
4
8653709
4
8664102
demand (in Kwh)
Weekday
Weekend
wknd_wday
as.factor(group)
1
2
3
4

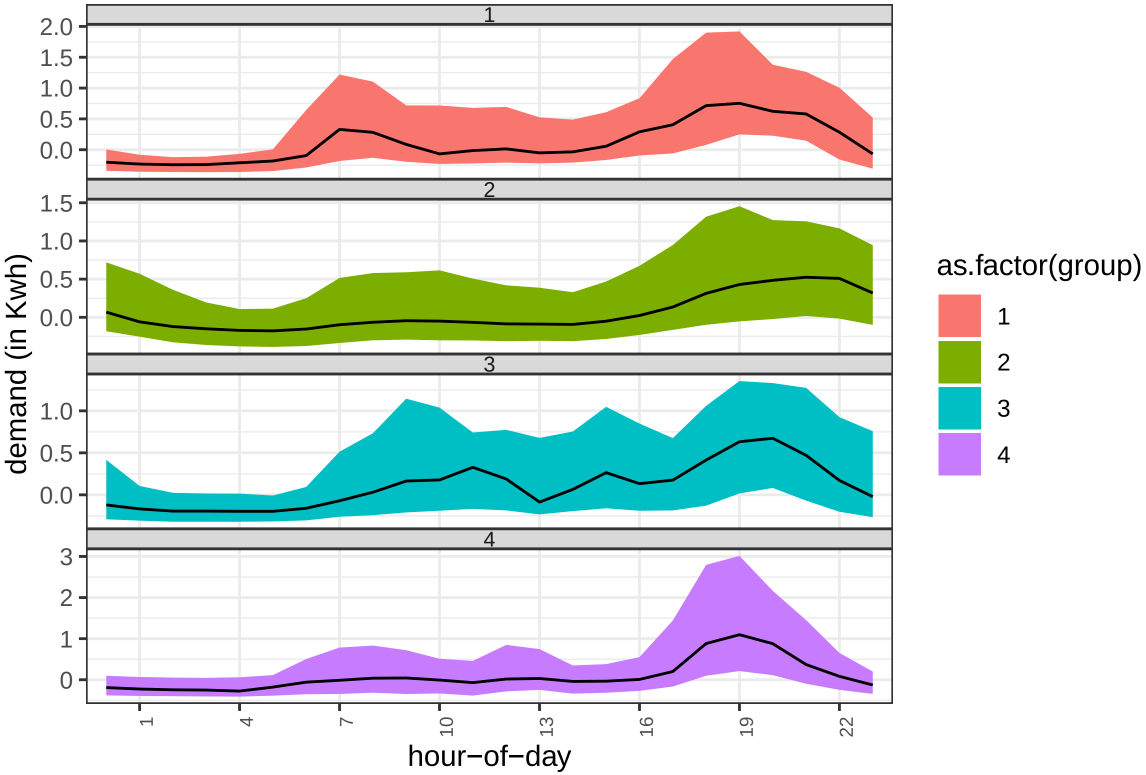


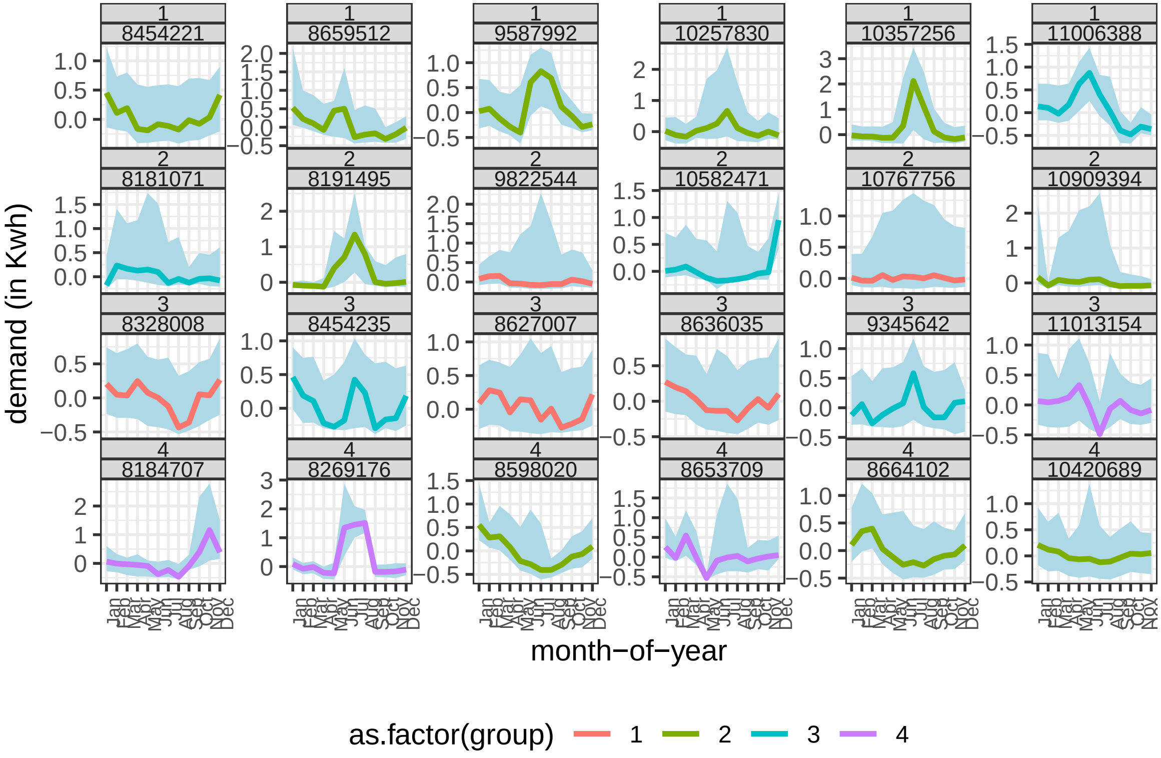
2 Clustering by method 2

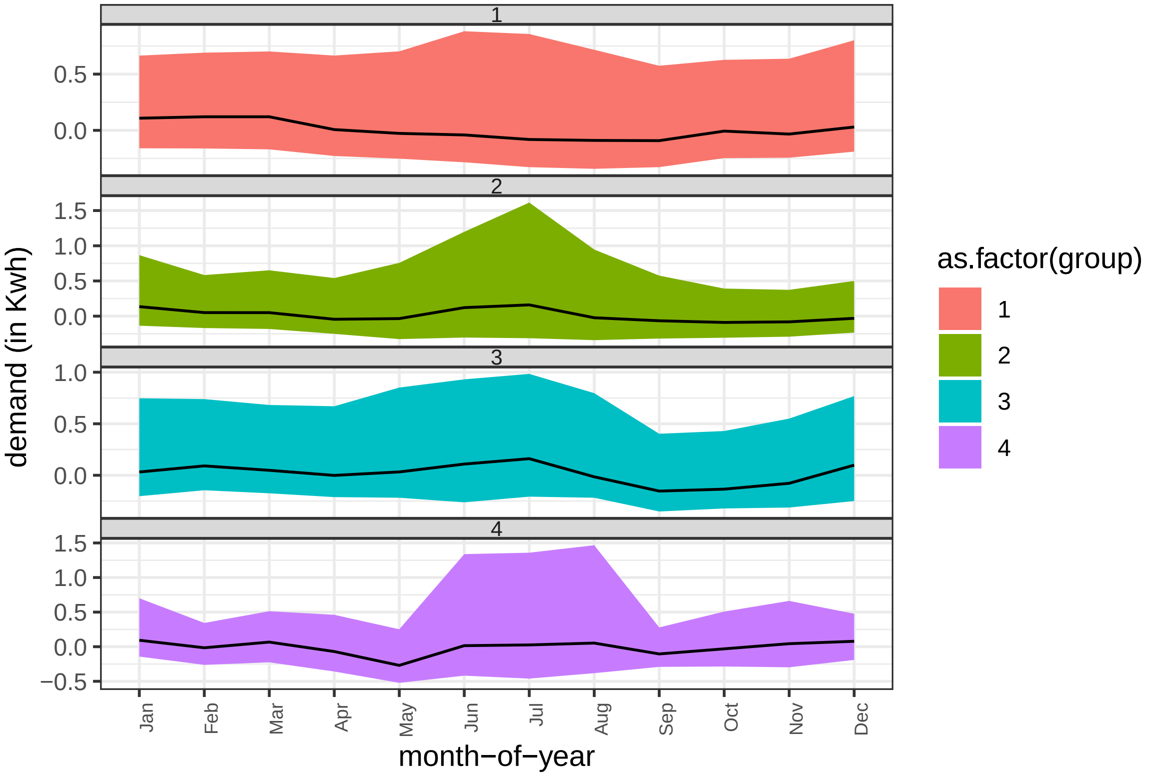


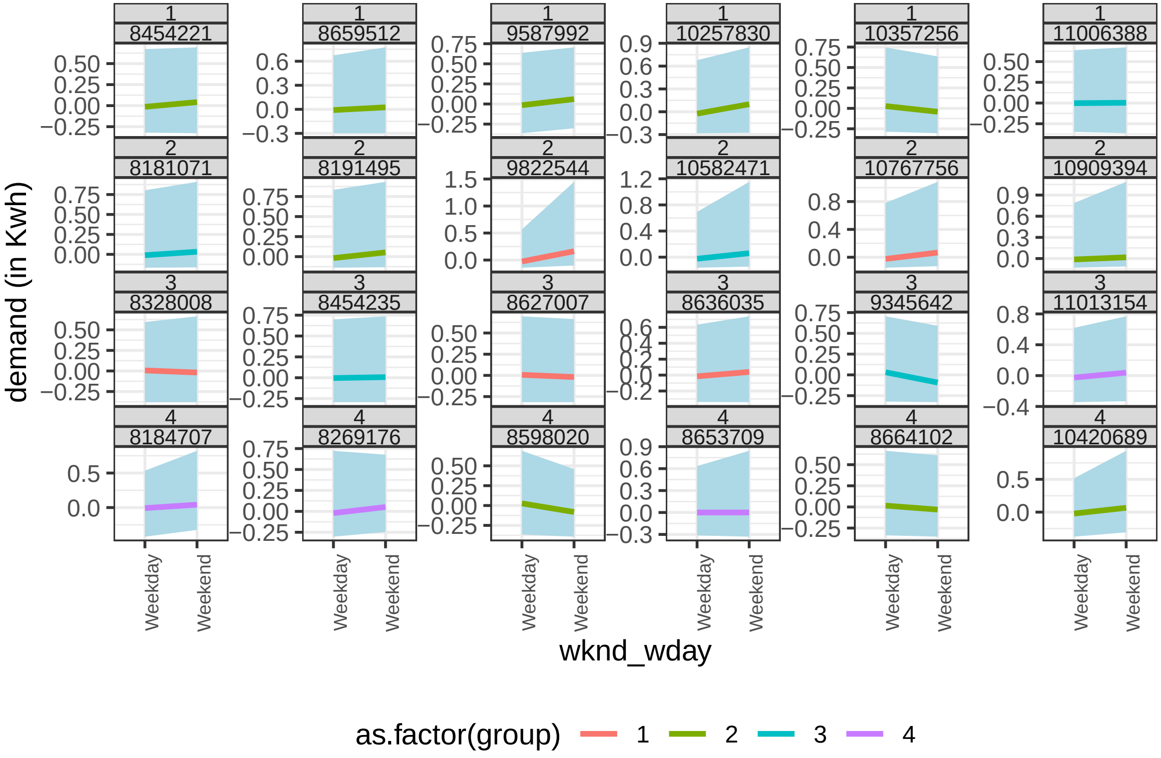

wpd
100
50
0
month_year
hour_day
wknd_wday

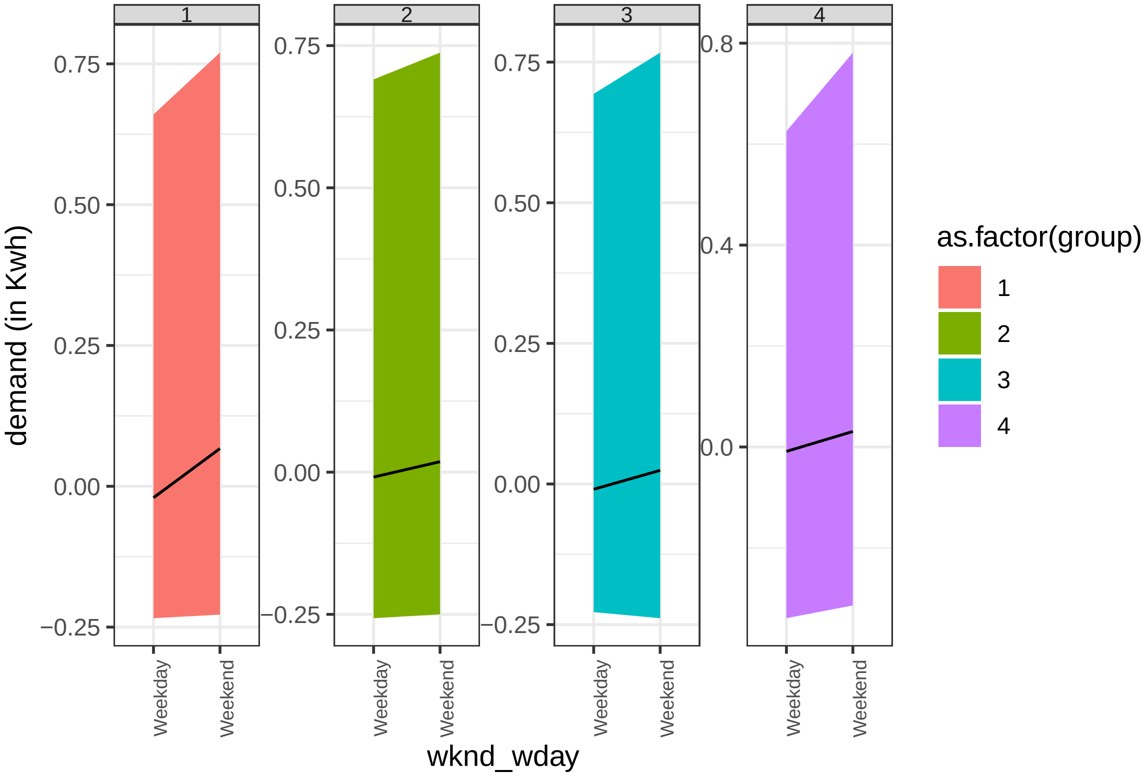

group 1 2 3 4
1 8454221
1 8659512
1 9587992
1 10257830
1 10357256
1 11006388
2 8181071
2 8191495
2 9822544
2 10582471
2 10767756
2 10909394
3 8328008
3 8454235
3 8627007
3 8636035
3 9345642
3 11013154
4 8184707
4 8269176
4 8598020
4 8653709
4 8664102
4 10420689
demand (in Kwh)
hour−of−day
as.factor(group) 1 2 3 4


1
2
3
4
demand (in Kwh)
hour−of−day
1
4
7
10
13
16
19
22
as.factor(group)
🟥 1
🟩 2
🟦 3
🟪 4


1 8454221
1 8659512
1 9587992
1 10257830
1 10357256
1 11006388
2 8181071
2 8191495
2 9822544
2 10582471
2 10767756
2 10909394
3 8328008
3 8454235
3 8627007
3 8636035
3 9345642
3 11013154
4 8184707
4 8269176
4 8598020
4 8653709
4 8664102
4 10420689
demand (in Kwh)
month−of−year
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
as.factor(group)
1
2
3
4


1
0.5
0.0
2
1.5
1.0
0.5
0.0
3
1.0
0.5
0.0
4
1.5
1.0
0.5
0.0
−0.5
demand (in Kwh)
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
month−of−year
as.factor(group)
1
2
3
4


1
8454221
0.50
0.25
0.00
−0.25
1
8659512
0.6
0.3
0.0
−0.3
1
9587992
0.75
0.50
0.25
0.00
−0.25
1
10257830
0.9
0.6
0.3
0.0
−0.3
1
10357256
0.75
0.50
0.25
0.00
−0.25
1
11006388
0.50
0.25
0.00
−0.25
2
8181071
0.75
0.50
0.25
0.00
2
8191495
0.75
0.50
0.25
0.00
2
9822544
1.5
1.0
0.5
0.0
2
10582471
1.2
0.8
0.4
0.0
2
10767756
0.8
0.4
0.0
2
10909394
0.9
0.6
0.3
0.0
3
8328008
0.50
0.25
0.00
−0.25
3
8454235
0.75
0.50
0.25
0.00
−0.25
3
8627007
0.50
0.25
0.00
−0.25
3
8636035
0.6
0.4
0.2
0.0
−0.2
3
9345642
0.75
0.50
0.25
0.00
−0.25
3
11013154
0.8
0.4
0.0
−0.4
4
8184707
0.5
0.0
4
8269176
0.75
0.50
0.25
0.00
−0.25
4
8598020
0.50
0.25
0.00
−0.25
4
8653709
0.9
0.6
0.3
0.0
−0.3
4
8664102
0.50
0.25
0.00
−0.25
4
10420689
0.5
0.0
demand (in Kwh)
Weekday
Weekend
wknd_wday
as.factor(group)
1
2
3
4



```
## # A tibble: 1,800 x 6
##    customer_serial_id facet_variable x_variable    wpd customer_id group
##    <chr>              <lgl>          <chr>       <dbl>       <int> <int>
##  1 1                  NA             month_year 35.1       8143599    NA
##  2 1                  NA             hour_day   10.4       8143599    NA
##  3 1                  NA             wknd_wday   3.86      8143599    NA
##  4 10                 NA             hour_day   10.1       8160755    NA
##  5 10                 NA             month_year  7.19      8160755    NA
##  6 10                 NA             wknd_wday  -0.613     8160755    NA
##  7 100                NA             hour_day   41.2       8272324    NA
##  8 100                NA             wknd_wday   4.82      8272324    NA
##  9 100                NA             month_year  0.747     8272324    NA
## 10 101                NA             month_year 50.5       8273636    NA
## # ... with 1,790 more rows
```