Supplementary materials for the main submission entitled - Clustering time series based on probability distributions across temporal granularities

Contents


```
#theme_set(theme_bw())
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.0.2
```

```
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 4.0.2
```

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

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

```
library(gravitas)
library(gracsr)
library(ggdendro)
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.0.2
```

```
##
## 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(readr)
```

```
## Warning: package 'readr' was built under R version 4.0.2
```

```
library(visdat)
```

```
## Warning: package 'visdat' was built under R version 4.0.2
```

```
library(ggplot2)
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.0.2
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v tibble  3.1.0     v stringr 1.4.0
## v tidyr   1.1.3     v forcats 0.5.1
## v purrr   0.3.4
```

```
## Warning: package 'tibble' was built under R version 4.0.2
```

```
## Warning: package 'tidyr' was built under R version 4.0.2
```

```
## Warning: package 'forcats' was built under R version 4.0.2
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x lubridate::as.difftime() masks base::as.difftime()
## x lubridate::date()        masks base::date()
## x dplyr::filter()          masks stats::filter()
## x lubridate::intersect()   masks base::intersect()
## x dplyr::lag()             masks stats::lag()
## x lubridate::setdiff()     masks base::setdiff()
## x lubridate::union()       masks base::union()
```

```
library(naniar)
```

```
## Warning: package 'naniar' was built under R version 4.0.2
```

```
library(here)
```

```
## Warning: package 'here' was built under R version 4.0.2
```

```
## here() starts at /Users/sgup0008/Documents/paper-gracsR
```

```
library(tsibble)
```

```
## Warning: package 'tsibble' was built under R version 4.0.2
```

```
##
## Attaching package: 'tsibble'
```

```
## The following object is masked from 'package:naniar':
##
##     pedestrian
```

```
## The following object is masked from 'package:lubridate':
##
##     interval
```

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

```
library(knitr)
```

```
## Warning: package 'knitr' was built under R version 4.0.2
```

```
library(patchwork)
```

```
## Warning: package 'patchwork' was built under R version 4.0.2
```

```
library(GGally)
```

```
## Warning: package 'GGally' was built under R version 4.0.2
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(distributional)
```

```
## Warning: package 'distributional' was built under R version 4.0.2
```

```
library(viridis)
```

```
## Warning: package 'viridis' was built under R version 4.0.2
```

```
## Loading required package: viridisLite
```

```r
library(forcats)
library(tidyr)
library(purrr)
library(stringr)
library(pals)
```

```
## Warning: package 'pals' was built under R version 4.0.2
```

```
##
## Attaching package: 'pals'
```

```
## The following objects are masked from 'package:viridis':
##
##     cividis, inferno, magma, plasma, viridis
```

```
## The following objects are masked from 'package:viridisLite':
##
##     cividis, inferno, magma, plasma, viridis
```

```r
theme_application <- function() {

  theme_light() + # setting theme
    #theme(strip.text = element_text(margin = margin(b = 0, t = 0))) + # narrow facet space
    theme(axis.title.y=element_blank(),
          axis.text.y=element_blank(),
          axis.ticks.y=element_blank()) + # no axis ticks
    theme(panel.spacing =unit(0, "lines")) +  # to ensure no gap between facets
    theme(axis.ticks.x = element_blank(),
          axis.text.x = element_blank()) + # no x-axis labels to further reduce the gap between facets
    #theme(axis.text.x = element_text(angle=90, hjust=1, size = 9)) + # rotate the x-axis text
    theme(plot.margin = margin(0, 0, 0, 0, "cm")) +
    #theme(axis.text.x = element_text(size=5)) +
    theme(strip.background = element_blank(),
          strip.text.x = element_blank())
}
```

```r
theme_characterisation <- function() {

  theme_bw() + # seeting theme
    theme(strip.text = element_text(size = 10,
                                    margin = margin(b = 0, t = 0))) + # narrow facet space
    theme(axis.title.y=element_blank(),
          axis.text.y=element_blank(),
          axis.ticks.y=element_blank()) + # no axis ticks
    theme(panel.spacing =unit(0, "lines")) +  # to ensure no gap between facets
    theme(axis.text.x = element_text(angle=90, hjust=1, size = 10)) + # rotate the x-axis text
    theme(legend.position = "bottom")+
    theme(plot.margin = margin(0, 0, 0, 0, "cm")) +
    theme(axis.text.x = element_text(size=5))
}
```

```
theme_validation <- function() {
  theme_bw() +
    theme(
      strip.text = element_text(size = 8, margin = margin(b = 0, t = 1)),
      plot.margin = margin(0, 0, 0, 0, "cm"),
      axis.title.y = element_blank(),
      #axis.text.x = element_blank(),
      axis.text.y = element_blank(),
      axis.ticks = element_blank(),
      legend.position = "bottom"
    )
}
```

0.1 Raw time series plot

To get a sense of how the raw time series data looks, we plot the energy usage for 50 sampled households is plotted along the y-axis versus time from past to future in Figure ref{fig:raw-data-50}. Each of these series is associated with a single customer. Energy consumption for each customer is given at fine temporal resolution (every 30 minutes) for a period of 2-3 years.

```
knitr::include_graphics("figs/raw_plot_cust.png") # look at smart-meter.R for the code
```

0.2 Plot displaying missing observations

Prototype selection method

S1. Robust scaling is applied to each customer.

S2. 50^{th} percentile for each category for each granularity is obtained for each customers. So we have a data structure with 356 rows and $(24 + 12 + 2)$ variables corresponding to 50^{th} percentile for each hour-of-day, month-of-year and weekend-weekday.

S3. Apply principal components and restrict the results down to the first six principal components (which makes up approximately 85% of the variance explained in the data) to use with the grand tour.

S4. Run t-SNE using the default arguments on the complete data (sets the perplexity to equal 30 and performs random initialisation). We then create a linked tour with t-SNE layout with liminal as shown in Figure 4.

S5. We inspect of the subspace generated by the set of low-dimensional projections in tour by looking for a simplex shape while the visualization moves from one basis to another. When we brush the corners of the simplex, we find they fall on the edge of the t-SNE point cloud. Hall, Marron, and Neeman (2005) have shown that in the extreme case of high-dimension, low-sample size data, observations are on the vertices of a simplex.

This is because in high-dimensional data analysis the curse of dimensionality reasons that points tend to be far away from the center of the distribution and on the edge of high-dimensional space. Contrary to this, is that projected data tends to clump at the center.

S6. These points should ideally correspond to different behavior with respect to all the variables considered while running PCA.

```
knitr::include_graphics("figs/liminal_view.png")
```

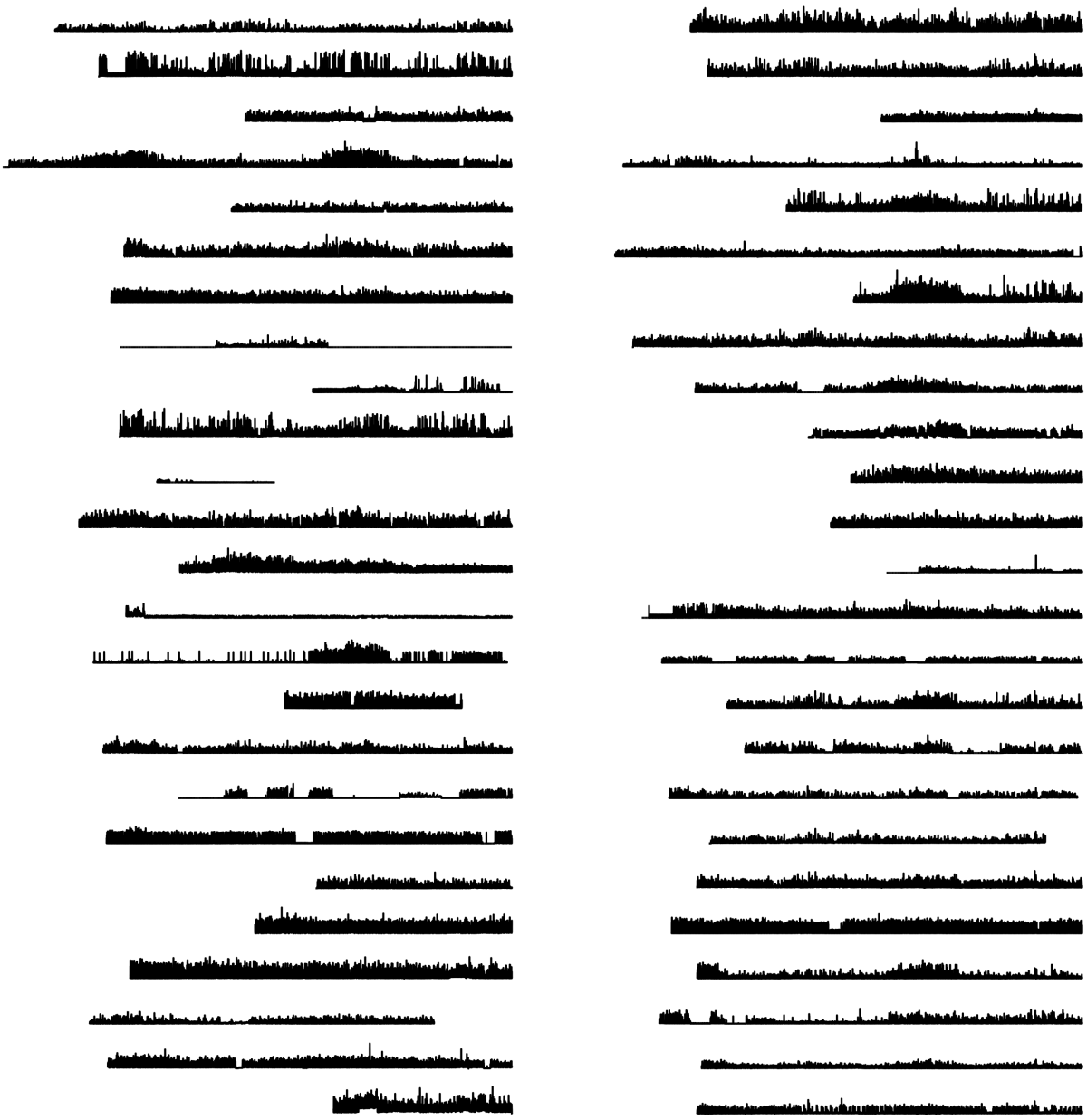


Figure 1: The raw data for 50 households are shown. It looks like there is a lot of missing values and unequal length of time series along with asynchronous periods for which data is observed. No insightful behavioral pattern could be discerned from this view other than when the customer is not at home.

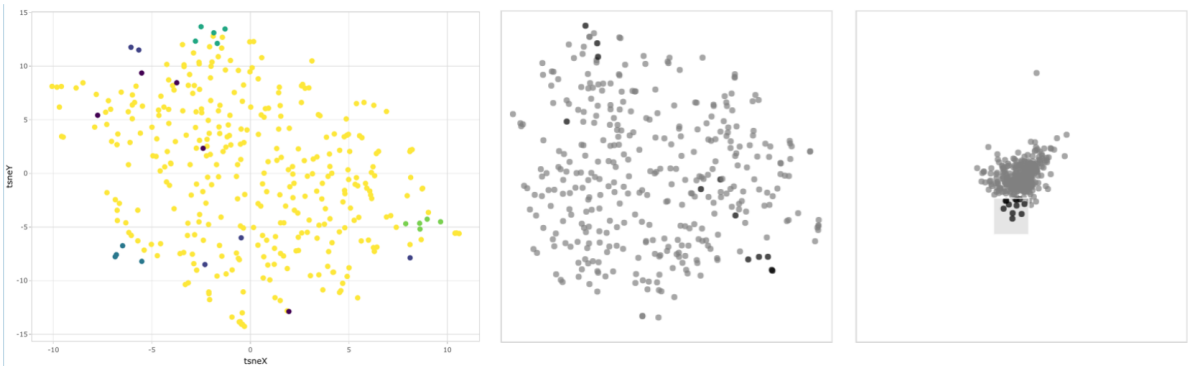


Figure 2: Instance selection using tours and projecting the points in a lower dimensional tsne cloud.

0.3 Clustering on 350 customers

```
# Read the nqt distances

wkndwday <- read_rds(here("data/dist_gran_wkndwday_356cust_nqt.rds")) %>% broom::tidy()

moy <- read_rds(here("data/dist_gran_moy_356cust_nqt.rds")) %>% broom::tidy()

hod <- read_rds(here("data/dist_gran_hod_356cust_nqt.rds")) %>% broom::tidy()


# Make the distance metrics

distance <- wkndwday %>%
  left_join(moy, by = c("item1", "item2")) %>%
  left_join(hod, by = c("item1", "item2")) %>%
  rename("wkndwday" ="distance.x",
         "moy" = "distance.y",
         "hod" = "distance") %>%
  mutate(item1 = as.integer(as.character(item1)),
         item2 = as.integer(as.character(item2)))

filtered_distance <- distance %>%
  filter(!(item1 %in% c(8196183, 8508008, 8680538))) %>%
  filter(!(item2 %in% c(8196183, 8508008, 8680538)))

total_distance <- filtered_distance %>%
  mutate(total = wkndwday/2 + moy/12 + hod/24)


total_distance_wide <- total_distance %>% pivot_wider(-c(2:5),
                                                      names_from = item2,
                                                      values_from = total)


rownames(total_distance_wide) <- total_distance_wide$item1
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
mds_data <- total_distance_wide %>%
  mutate_all(~replace(., is.na(.), 0)) %>%
  tibble::rownames_to_column() %>%
  dplyr::select(-item1) %>%
  pivot_longer(-rowname) %>%
  pivot_wider(names_from=rowname, values_from=value)

rownames(mds_data) <- total_distance_wide$item1
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
df <- mds_data[-1] %>% as.matrix()
DM <- matrix(0, ncol(mds_data), ncol(mds_data))
DM[lower.tri(DM)] = df[lower.tri(df, diag=TRUE)] # distance metric
f = as.dist(DM)


first_lot <- mds_data %>% names()

id <- c(first_lot[-1], mds_data$name[nrow(mds_data)])


# Find optimal number of clusters

library(fpc)
```

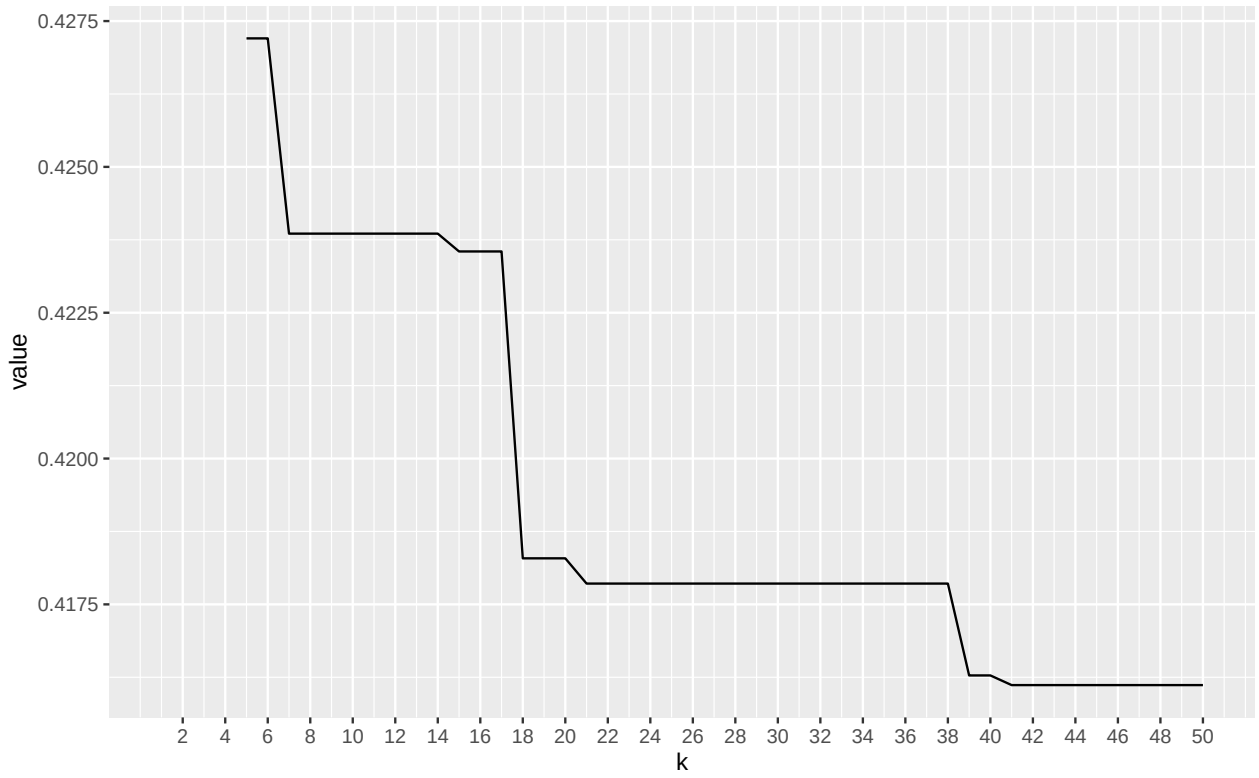
```
## Warning: package 'fpc' was built under R version 4.0.2
```

```
library(cluster)
```

```
## Warning: package 'cluster' was built under R version 4.0.2
```

```
k = array()
for(i in 5:50)
{
  group <- f %>% hclust (method = "ward.D") %>% cutree(k=i)
  p <- cluster.stats(f, clustering = group, silhouette = TRUE)
  k[i]=p$sindex
}

ggplot(k %>% as_tibble %>% mutate(k = row_number()), aes(x=k, y = value)) + geom_line() + scale_x_conti
```

```
## Warning: Removed 4 row(s) containing missing values (geom_path).
```




```
# plot(k, type = "l")
# 6 coming as the number of clusters with maximum silwidth

group <- f %>% hclust (method = "ward.D") %>% cutree(k=17)
cluster_result <- bind_cols(customer_id = id, group = group)

# cluster_result %>% group_by(group) %>% count()
```

```
legend_title <- "group"

data_pick <- read_rds(here::here("data/elec_nogap_2013_clean_356cust.rds")) %>%
  mutate(customer_id = as.character(customer_id)) %>%
  gracsr::scale_gran( method = "robust",
                      response = "general_supply_kwh")

data_group <- data_pick  %>%
  mutate(customer_id = as.character(customer_id)) %>%
   gracsr::scale_gran( method = "robust",
                       response = "general_supply_kwh") %>%
  left_join(cluster_result, by = c("customer_id"))

data_heatmap_hod_group <- quantile_gran(data_group,
                                gran1="hour_day",
                                quantile_prob_val = c(0.25, 0.5, 0.75),
                                group="group") %>%
  pivot_wider(names_from = quantiles, values_from = quantiles_values)


data_heatmap_hod_group$category <- factor(data_heatmap_hod_group$category, levels = 0:23)
```

```
data_heatmap_hod_group$group <- paste("group", data_heatmap_hod_group$group, sep = "-")

hod_group_entire <- data_heatmap_hod_group %>%
  ggplot(aes(x = category)) +
  geom_ribbon(aes(ymin = `25%`,
                  ymax = `75%`,
                  group=group,
                  fill = as.factor(group), alpha = 0.5),
              alpha = 0.5) +
  geom_line(aes(y = `50%`,
                group=group,
                color = as.factor(group)), size = 1)+
  facet_wrap(~group,
             scales = "free_y",
             ncol = 17) +
              #labeller = labeller(xfacet = c(`1` = "Group 2", `2` = "Group 4",`3` = "Group 1",`4` = "G
    theme(strip.text = element_text(size = 10, margin = margin(b = 0, t = 0))) + xlab("hour-of-day") +
  ylab("demand (in Kwh)") +
  theme_bw()  +
  scale_x_discrete(breaks = seq(1, 24, 3))+
  #theme(strip.text = element_text(size = 8, margin = margin(b = 0, t = 0)))+
  theme_application() +
  scale_fill_manual(values=as.vector(polychrome(20)))+
  scale_color_manual(values=as.vector(polychrome(20)))+
  theme(legend.position = "bottom")
```

```
data_heatmap_moy_group <- quantile_gran(data_group,
                                gran1="month_year",
                                quantile_prob_val = c(0.25, 0.5, 0.75),
                                group="group") %>%
  pivot_wider(names_from = quantiles, values_from = quantiles_values)

data_heatmap_moy_group$category <- factor(data_heatmap_moy_group$category, levels = c("Jan", "Feb", "Ma


data_heatmap_moy_group$group <- paste("group", data_heatmap_moy_group$group, sep = "-")


moy_group_entire <- data_heatmap_moy_group %>%
  ggplot(aes(x = category)) +
  geom_ribbon(aes(ymin = `25%`,
                  ymax = `75%`, group=group, fill = as.factor(group)), alpha = 0.5) +
  geom_line(aes(y = `50%`, group=group, color = as.factor(group)), size = 1 ) +
  facet_wrap(~group,
             scales = "free_y",
             labeller = "label_value",
             ncol = 17) +
    theme(strip.text = element_text(size = 10, margin = margin(b = 0, t = 0))) + xlab("month-of-year")
  ylab("demand (in Kwh)") +
  theme_bw() + theme_application() +
  scale_fill_manual(values=as.vector(polychrome(20)))+
  scale_color_manual(values=as.vector(polychrome(20)))+
  theme(legend.position = "bottom")
```

```
wkndwday_data <- data_group %>%
  create_gran("wknd_wday")
wkndwday_data$group <- as.factor(wkndwday_data$group)
# %>%
#   create_gran("hour_day")

ylim1 = boxplot.stats(wkndwday_data$general_supply_kwh)$stats[c(1, 5)]

wkndwday_group_entire <- wkndwday_data%>%
  ggplot(aes(x=wknd_wday, y = general_supply_kwh)) +
  #lvplot::geom_lv(aes(fill = as.factor(group)), k=5) +
  geom_boxplot(aes(fill = group, color = group),alpha = 0.5, outlier.alpha = 0.05)+
  #geom_boxplot(outlier.size = 1) +
  coord_cartesian(ylim = ylim1*1.05)+
  #ggridges::geom_density_ridges2(aes(x = general_supply_kwh, y = wknd_wday,fill = as.factor(group))) +
#geom_boxplot(aes(fill = as.factor(group))) +
  #scale_fill_lv() +
 xlab("wknd-wday") +
  ylab("demand (in Kwh)") +
   facet_wrap(~group,
              scales = "free_y",
              labeller = "label_both",
              ncol = 17) +
  theme_bw() + theme_application() +
  scale_fill_manual(values=as.vector(polychrome(20)))+
  scale_color_manual(values=as.vector(polychrome(20)))+
  theme(legend.position = "none")
```

```
 (hod_group_entire/moy_group_entire/ wkndwday_group_entire) +
  plot_annotation(tag_levels = 'a', tag_prefix = '(', tag_suffix = ')')+
  plot_layout(guides = "collect")& theme(legend.position = 'none')
```

1 contribution

```
# data_validation <- hod %>%
#   rename("hod" = "distance") %>%
#   left_join(moy,
#             by = c("item1", "item2"))%>%
#   rename("moy" = "distance") %>%
#   left_join((wkndwday),
#             by = c("item1", "item2"))%>%
#   rename("wkndwday" = "distance") %>%
#   left_join(cluster_result, by = c("item1" = "customer_id")) %>%
#   rename("group_item1" = "group") %>%
#   left_join(cluster_result, by = c("item2" = "customer_id")) %>%
#   rename("group_item2" = "group") %>%
#   mutate(hod = hod/24, moy = moy/12, wkndwday = wkndwday/2) %>%
#   filter(!is.na(group_item1)| !is.na(group_item2)) %>%
#   pivot_longer(3:5,names_to="gran",
#                values_to = "distance")
```

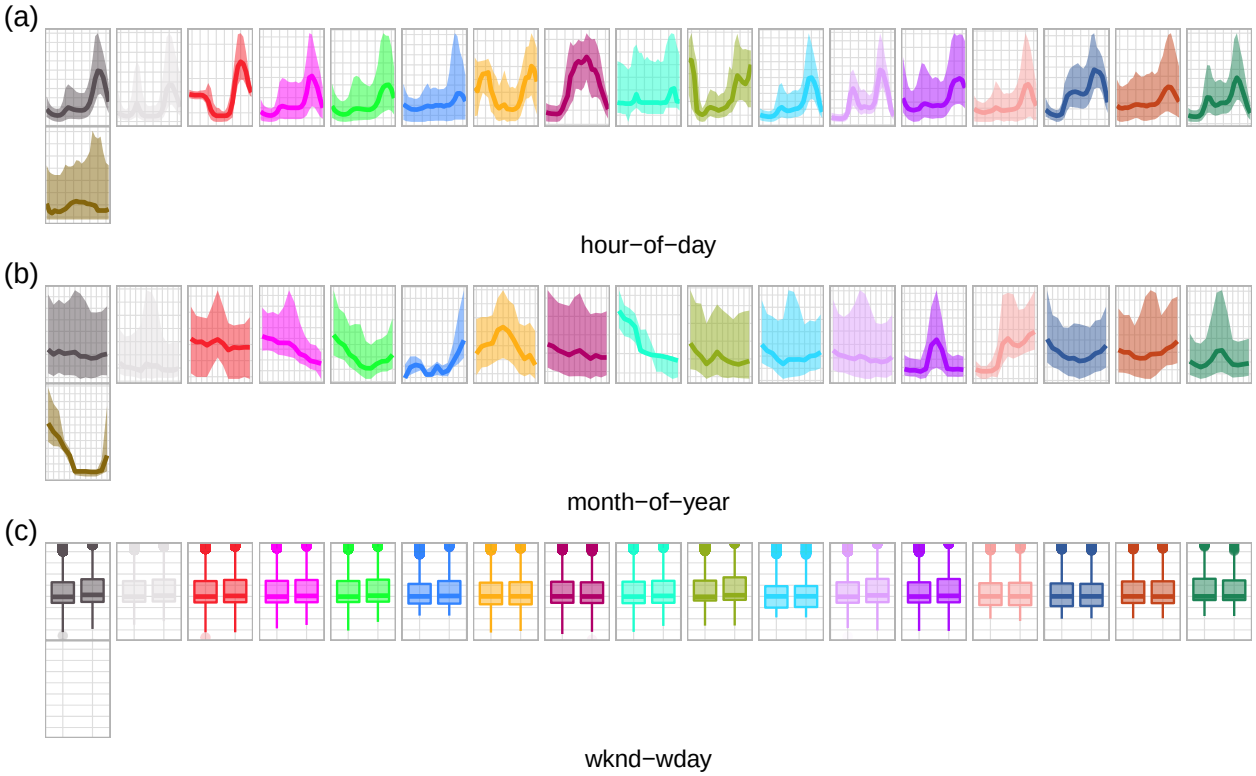


Figure 3: The distribution of electricity demand for the clusters across hod (a), moy (b) and wkndwday (c). It seems like group 2 and 5 have a hod pattern across its members, while group 1, 3, 5 have a moy pattern. Wknd-wday variations across groups are not distinguishable, indicating that it is not a critical variable for clustering. It is helpful to compare the summarised distributions of groups to that of individuals to confirm that the most of individuals in the group have the same characterisation.

```
#
```

```
data_validation_cat <- read_rds(here::here("validation/data_validation_cat_all_cluster.rds"))
```

```
# hod_cat <- data_pick %>%
#   dist_gran_cat(gran1 = "hour_day", response = "general_supply_kwh") %>%
#   mutate(gran = "hod")
#
# moy_cat <- data_pick %>%
#   dist_gran_cat(gran1 = "month_year", response = "general_supply_kwh")%>%
#   mutate(gran = "moy")
#
# wkndwday_cat <- data_pick %>%
#   dist_gran_cat(gran1 = "wknd_wday", response = "general_supply_kwh")%>%
#   mutate(gran = "wnwd")
#
#
# data_validation_cat <- bind_rows(hod_cat, moy_cat, wkndwday_cat) %>%
#   mutate(customer_from = as.character(customer_from),
#          customer_to = as.character(customer_to))
```

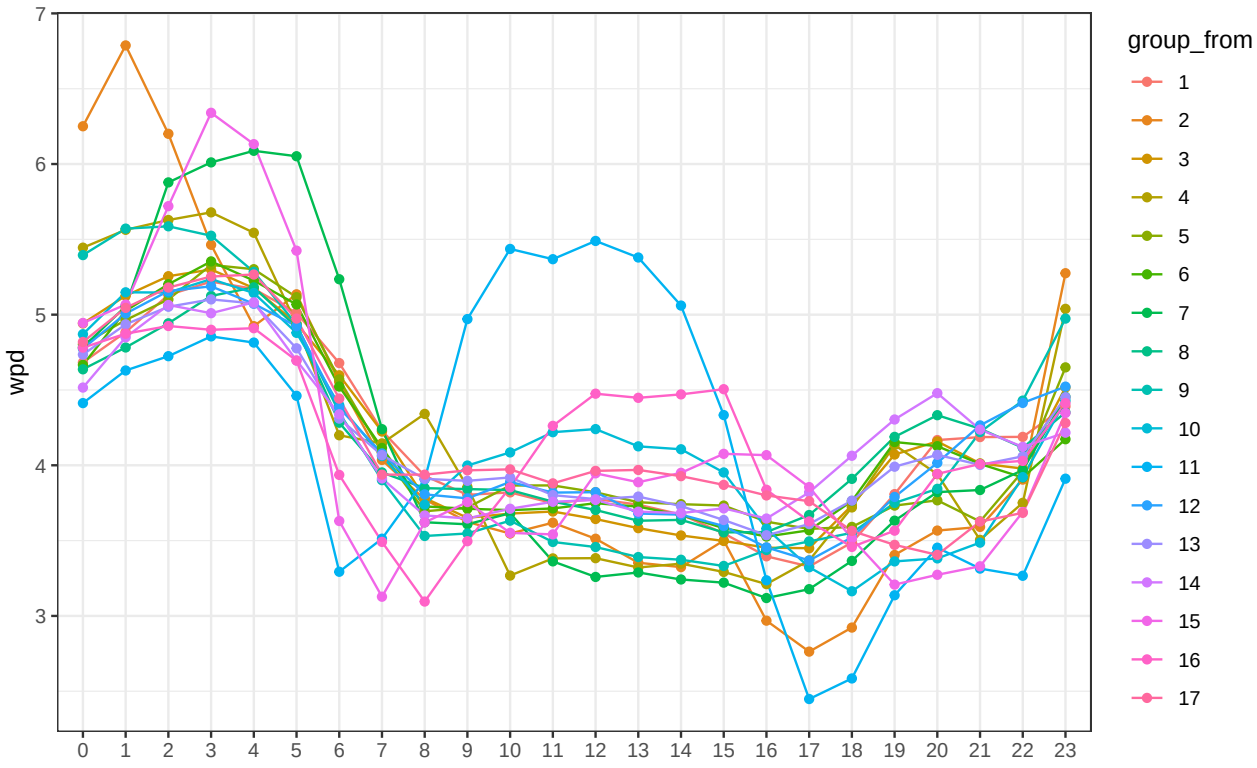
```
gran_cat_dist <- data_validation_cat %>%
  mutate(customer_from = as.character(customer_from),
         customer_to = as.character(customer_to)) %>%
  left_join(cluster_result, by = c("customer_from" = "customer_id")) %>%
  rename("group_from" = "group") %>%
  left_join(cluster_result, by = c("customer_to" = "customer_id")) %>%
  rename("group_to" = "group") %>%
  group_by(gran,
           category,
           group_from,
           group_to) %>%
  summarise(sum = sum(distance)) %>%
  pivot_wider(names_from = group_to, values_from = sum) %>%
  mutate(distance = sum(`1`, `2`, `3`, `4`, `5`,`6`, `7`, `8`, `9`, `10`, `11`, `12`, `13`, `14`, `15`,
  select(-(4:8)) %>%
  arrange(-distance)
```

```
## `summarise()` has grouped output by 'gran', 'category', 'group_from'. You can override using the `.gr
```

```
# Contribution of individual category for hod
```

```
gran_cat_hod <-   gran_cat_dist %>%
  filter(gran=="hod") %>%
  mutate(category = as.integer(category)) %>%
  arrange(group_from, category)

gran_cat_hod$category <- factor(gran_cat_hod$category, levels = 0:23)
```

```r
# for each group, percent contribution of hours

group_total <- gran_cat_hod %>%
  ungroup() %>%
  group_by(group_from) %>%
  summarise(group_total = sum(distance))


data_pcp <- gran_cat_hod %>%
  ungroup %>%
  left_join(group_total, by = "group_from") %>%
  mutate(cat_contibution = distance*100/group_total) %>%
  group_by(group_from) %>%
  #arrange(-distance) %>%
  select(group_from, category, cat_contibution) %>%
  pivot_wider(names_from = "category",
              values_from = "cat_contibution") %>% mutate(group_from = as.factor(group_from))

parcoord <- GGally::ggparcoord(data_pcp ,
                               columns = 2:ncol(data_pcp),
                               groupColumn = "group_from",
                               showPoints = TRUE,
                               alphaLines = 1,
                               scale = "globalminmax"
) +
  ggplot2::theme(
    plot.title = ggplot2::element_text(size=10)
  )+
  ggplot2::theme(axis.text.x = ggplot2::element_text(angle = 10)) +
  theme(legend.position = "bottom") +
  xlab("") +
  ylab("wpd")+
  scale_fill_manual(values=as.vector(polychrome(20)))+
  theme_bw()
parcoord
```


wpd
7
6
5
4
3
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
group_from
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17