Using Clustering Of Electricity Consumers To Produce More Accurate Predictions

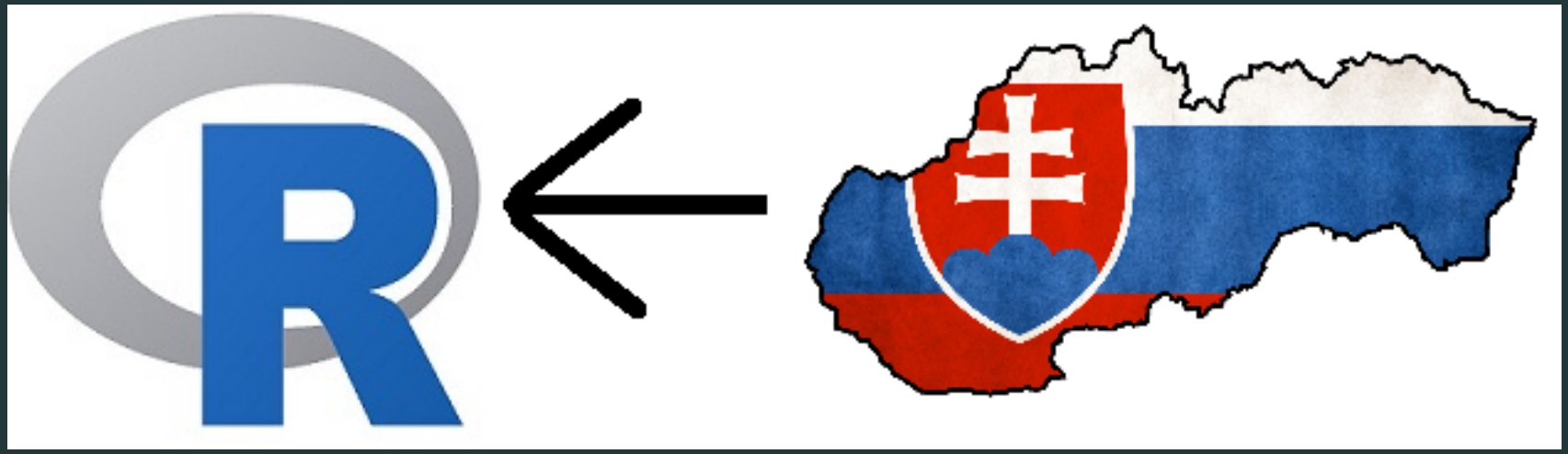
#1 R<-Slovakia meetup

Peter Laurinec

22.3.2017

FIIT STU

R <- Slovakia



Why are we here? Rrrr

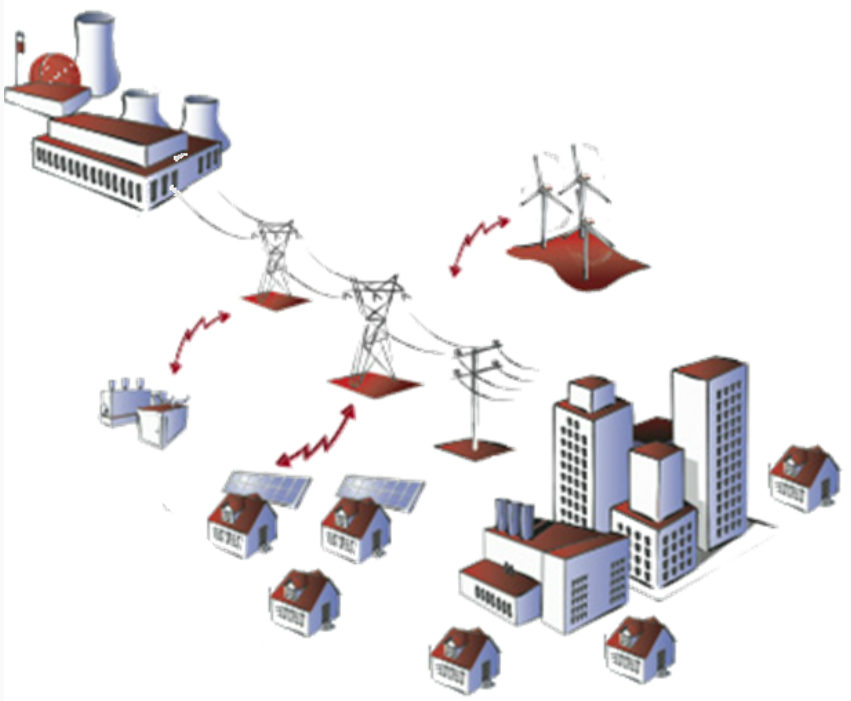
CRAN - about 10300 packages - **https://cran.r-project.org/web/packages/**.

Popularity - **http://redmonk.com/sogrady/**:

Why prediction of electricity consumption?

Important for:

- Distribution (utility) companies. Producers of electricity. Overload.
- Deregularization of market
- Buy and sell of electricity.
- Source of energy - wind and photo-voltaic power-plant. Hardly predictable.
- Active consumers. **pro**ducer + con**sumer** $\Rightarrow$ **prosumer**.
- Optimal settings of bill.



Smart grid

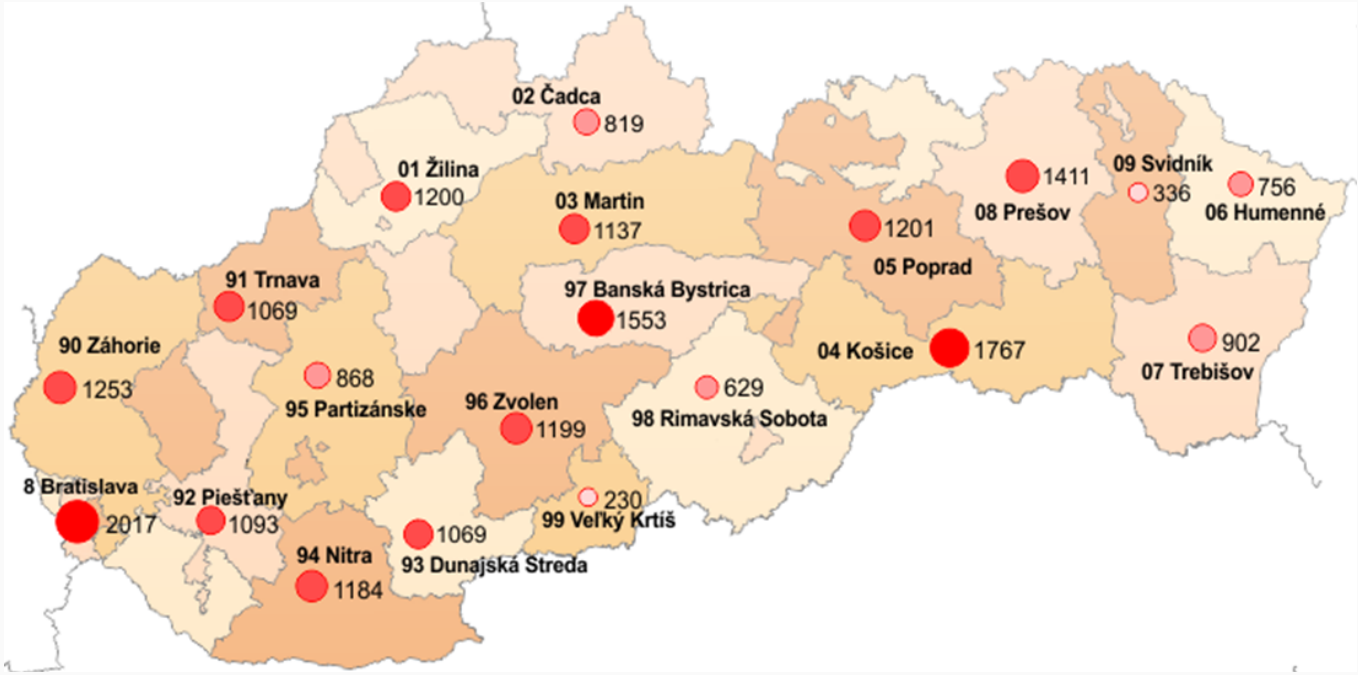
- smart meter
- demand response
- smart cities

Good for:

- Sustainability - Blackouts...
- Green energy
- Saving energy
- Dynamic tariffs $\Rightarrow$ saving money

Slovak data

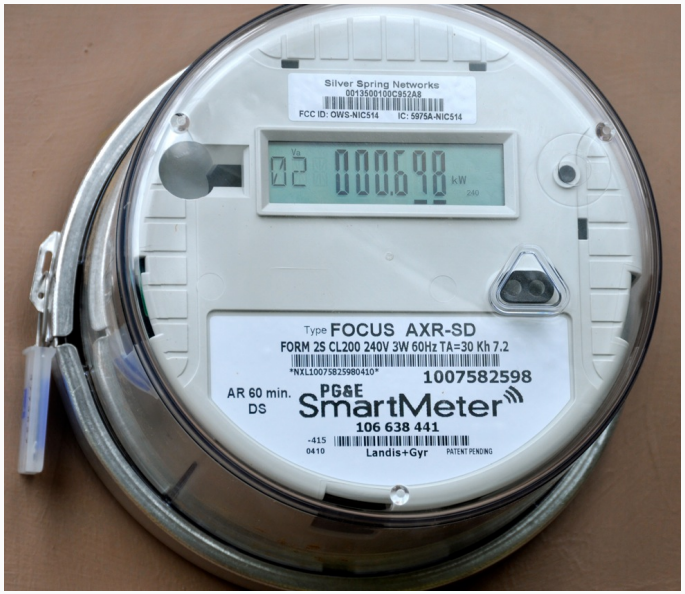
- Number of consumers are 21502. From that 11281 are OK. Enterprises.
- Time interval of measurements: 01.07.2011 – 14.05.2015. But mainly from the 01.07.2013. 96 measurements per day.



Irish data:

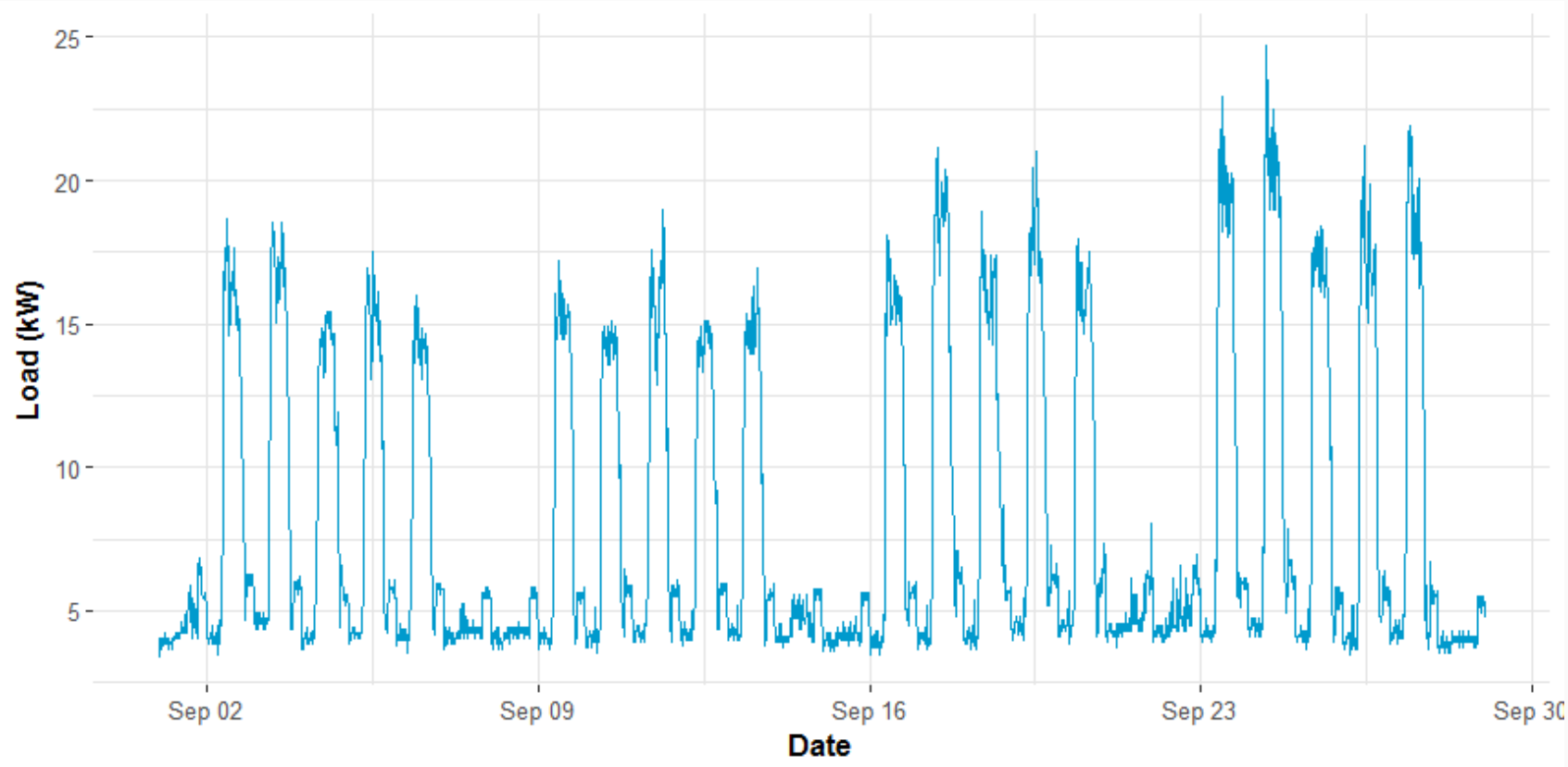
- Number of consumers are 6435. From that 3639 are residences.
- Time interval: 14.7.2009 – 31.12.2010. 48 measurements per day.

Smart meter data

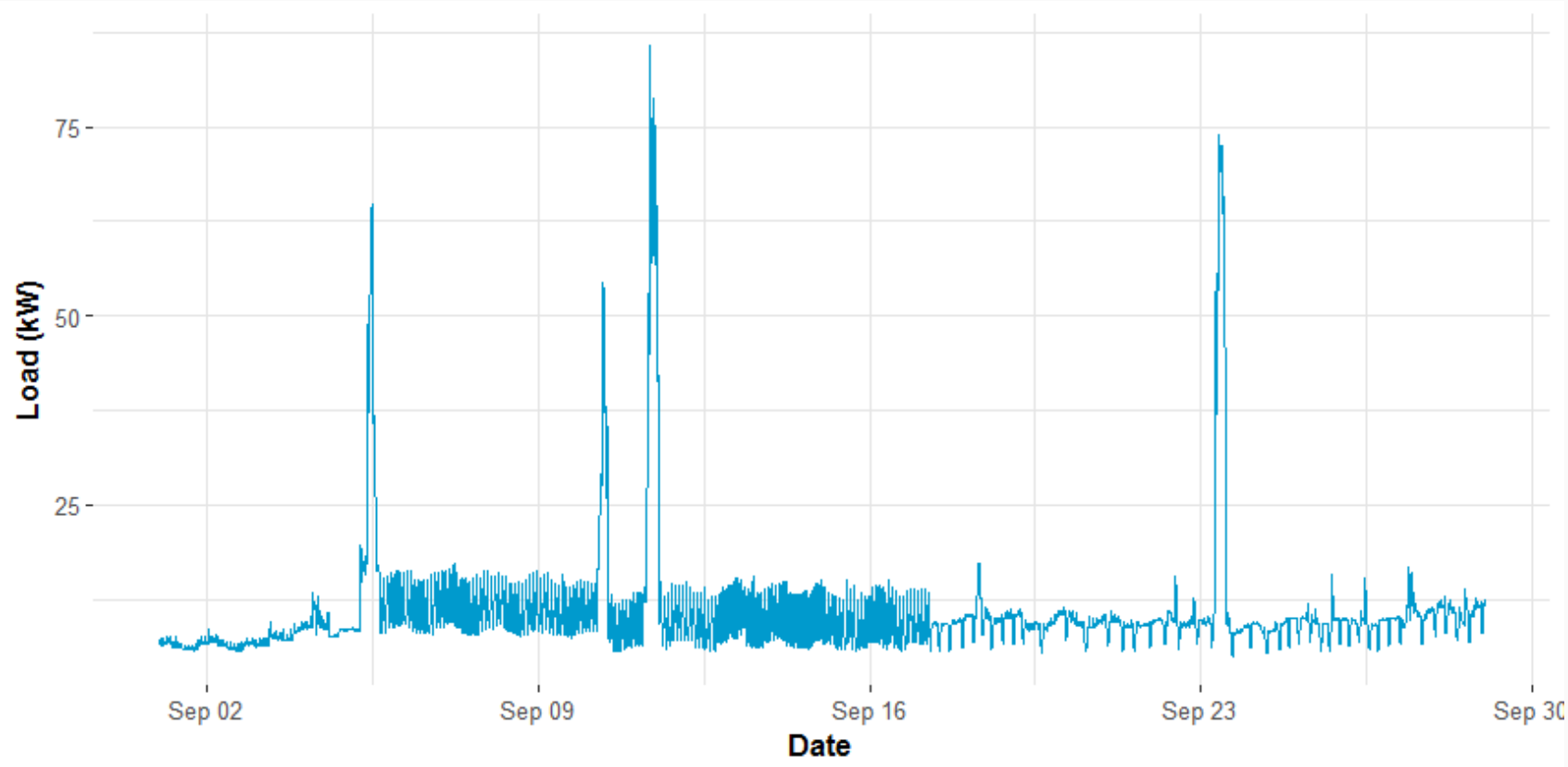


	OOM_ID	DIAGRAM_ID	TIME	LOAD	TYPE_OF_OOM	DATE	ZIP
1:	11	202004	−45	4.598	O	01/01/2014	4013
2:	11	202004	195	4.087	O	01/01/2014	4013
3:	11	202004	−30	5.108	O	01/01/2014	4013
4:	11	202004	345	4.598	O	01/01/2014	4013
5:	11	202004	825	2.554	O	01/01/2014	4013
6:	11	202004	870	2.554	O	01/01/2014	4013
41312836:	20970	14922842	90	18.783	O	14/02/2015	4011
41312837:	20970	14922842	75	20.581	O	14/02/2015	4011
41312838:	20970	14922842	60	18.583	O	14/02/2015	4011
41312839:	20970	14922842	45	18.983	O	14/02/2015	4011
41312840:	20970	14922842	30	17.384	O	14/02/2015	4011
41312841:	20970	14922842	15	18.583	O	14/02/2015	4011

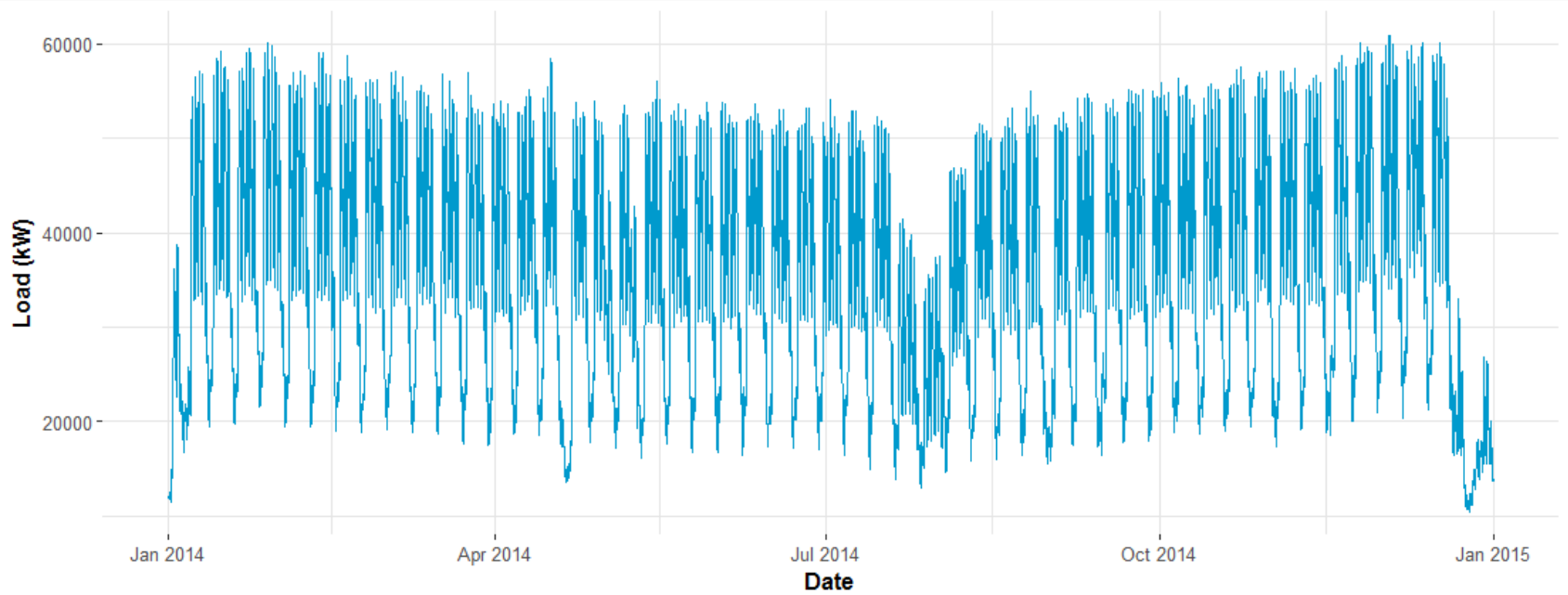
Random consumer from Kosice vol. 1



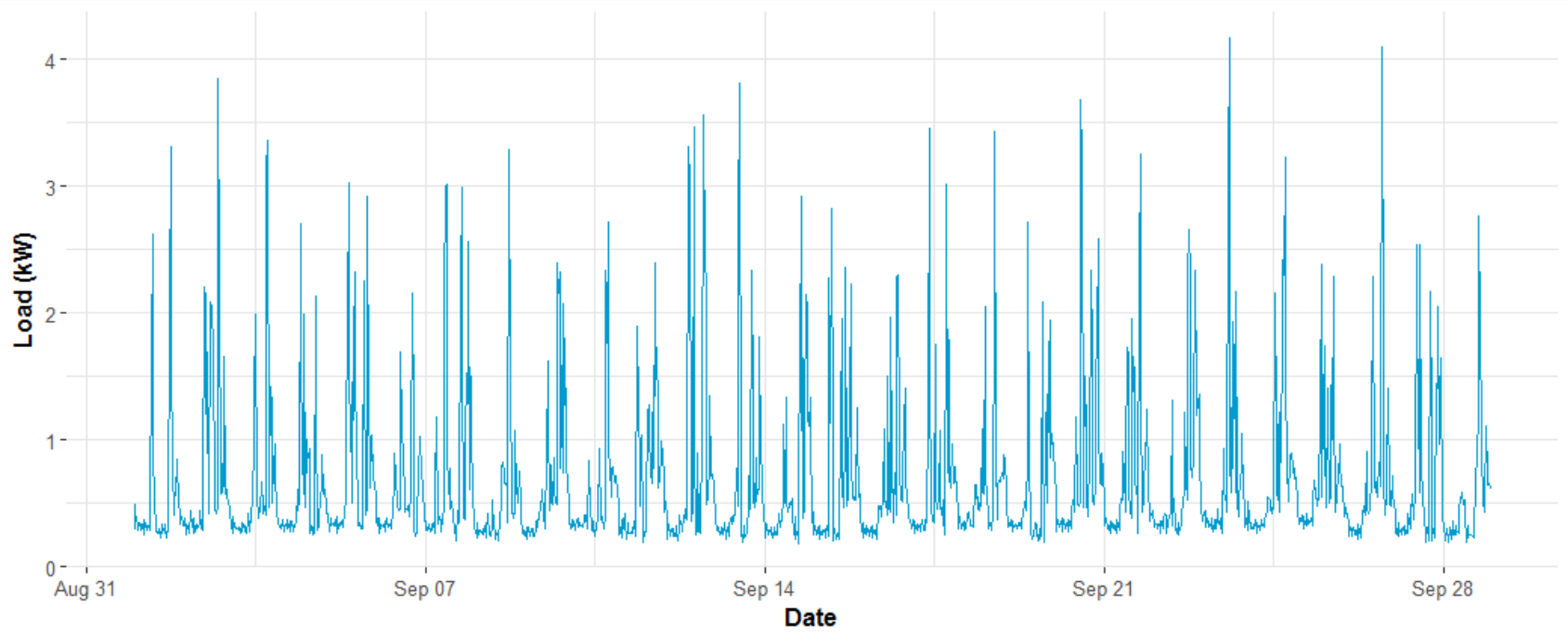
Random consumer from Kosice vol. 2



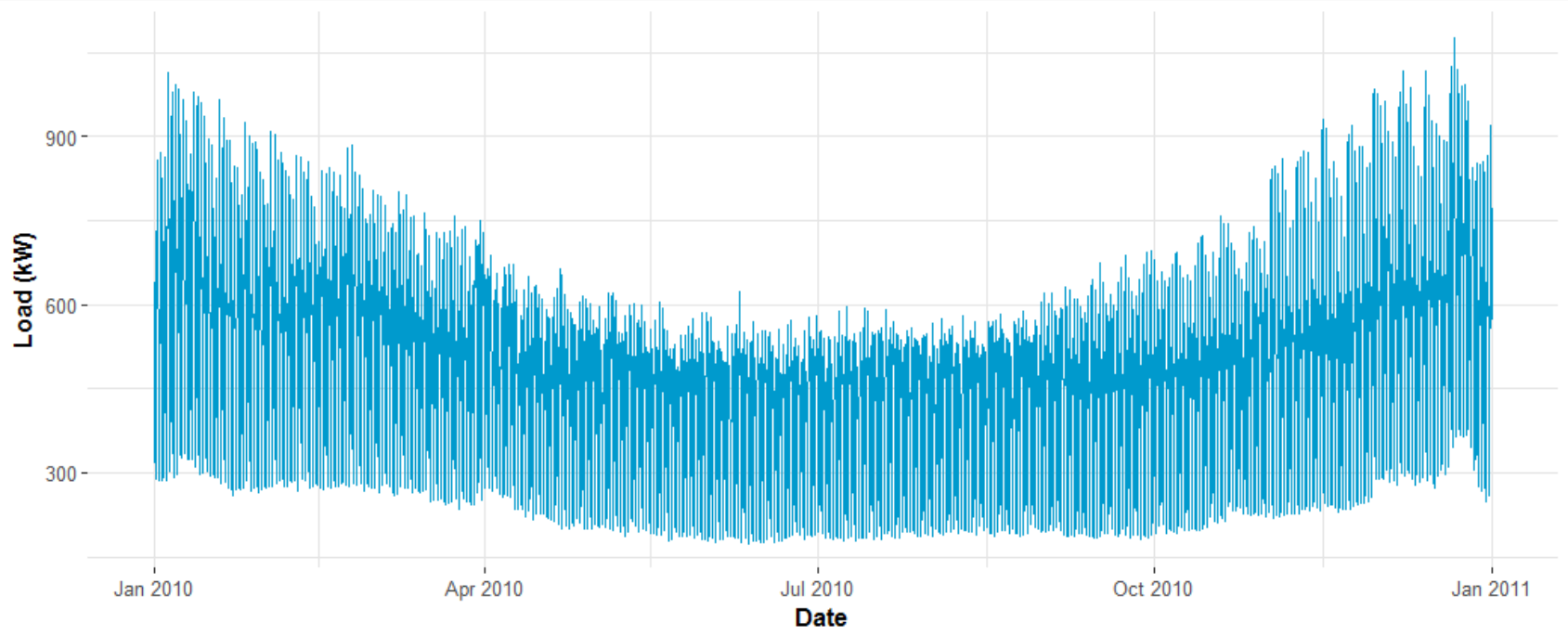
Aggregate load from Zilina



Random consumer from Ireland



Aggregate load from Ireland

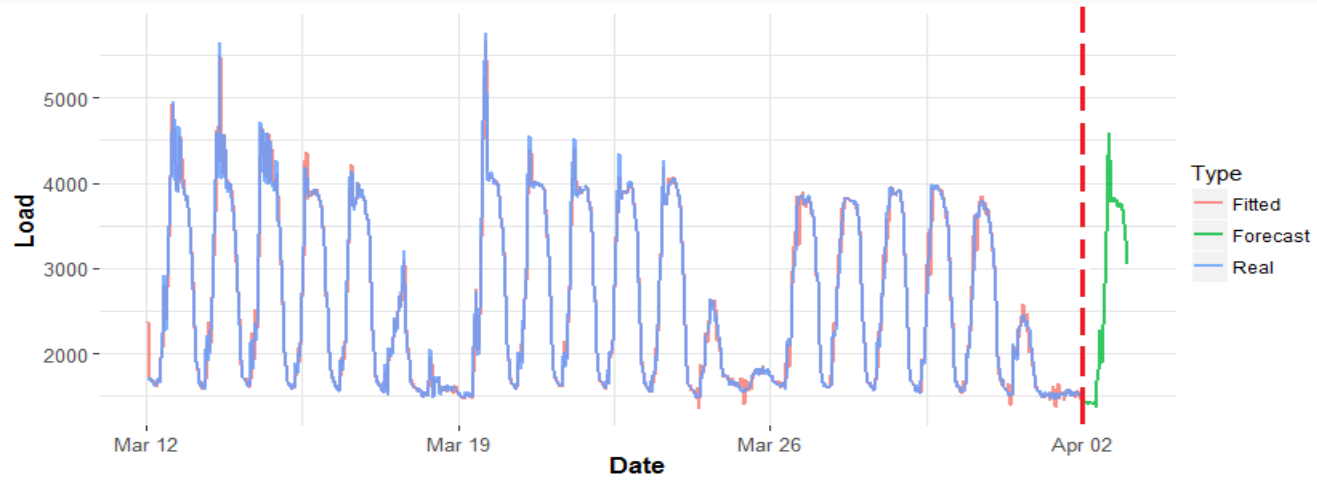


R <- ”Forecasting”

STLF - Short Term Load Forecasting

- Prediction (forecast) for 1 day ahead (96 resp. 48 measurements). Short term forecasting.
- Strong time dependence. Seasonalities (daily, weekly, yearly). Holidays. Weather.
- Accuracy of predictions (in %) - MAPE (Mean Absolute Percentage Error).

$$\text{MAPE} = 100 \times \frac{1}{n} \sum_{t=1}^{n} \frac{|x_t - \hat{x}_t|}{x_t}$$



Methods for prediction of time series

Time series analysis

ARIMA, Holt-Winters exponential smoothing, decomposition TS...

Linear regression

Multiple linear regression, robust LR, GAM (Generalized Additive Model).

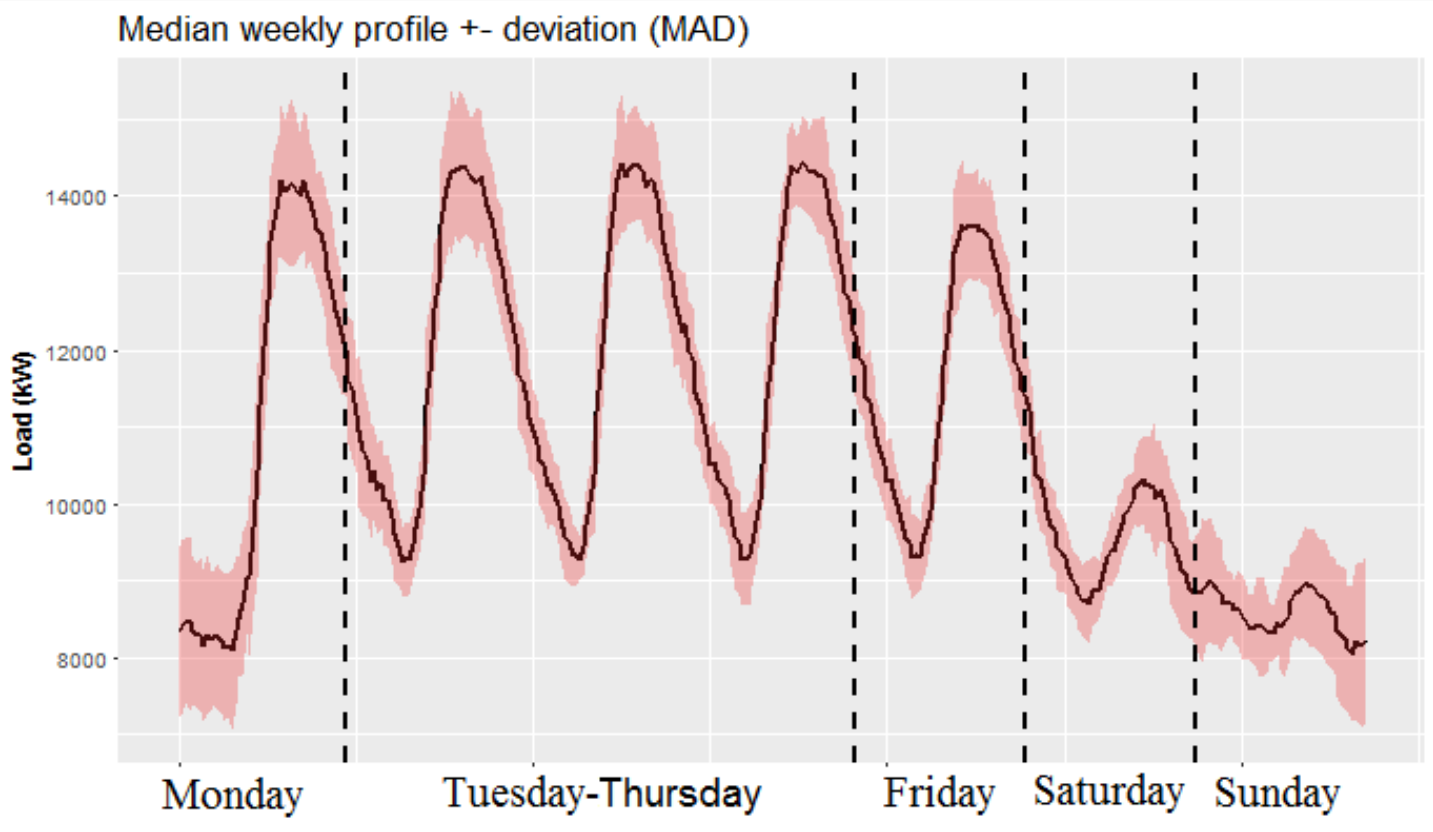
Machine learning

- Neural nets
- Support Vector Regression
- Regression trees and forests

Ensemble learning

Linear combination of predictions.

Weekly profile



Time series analysis methods

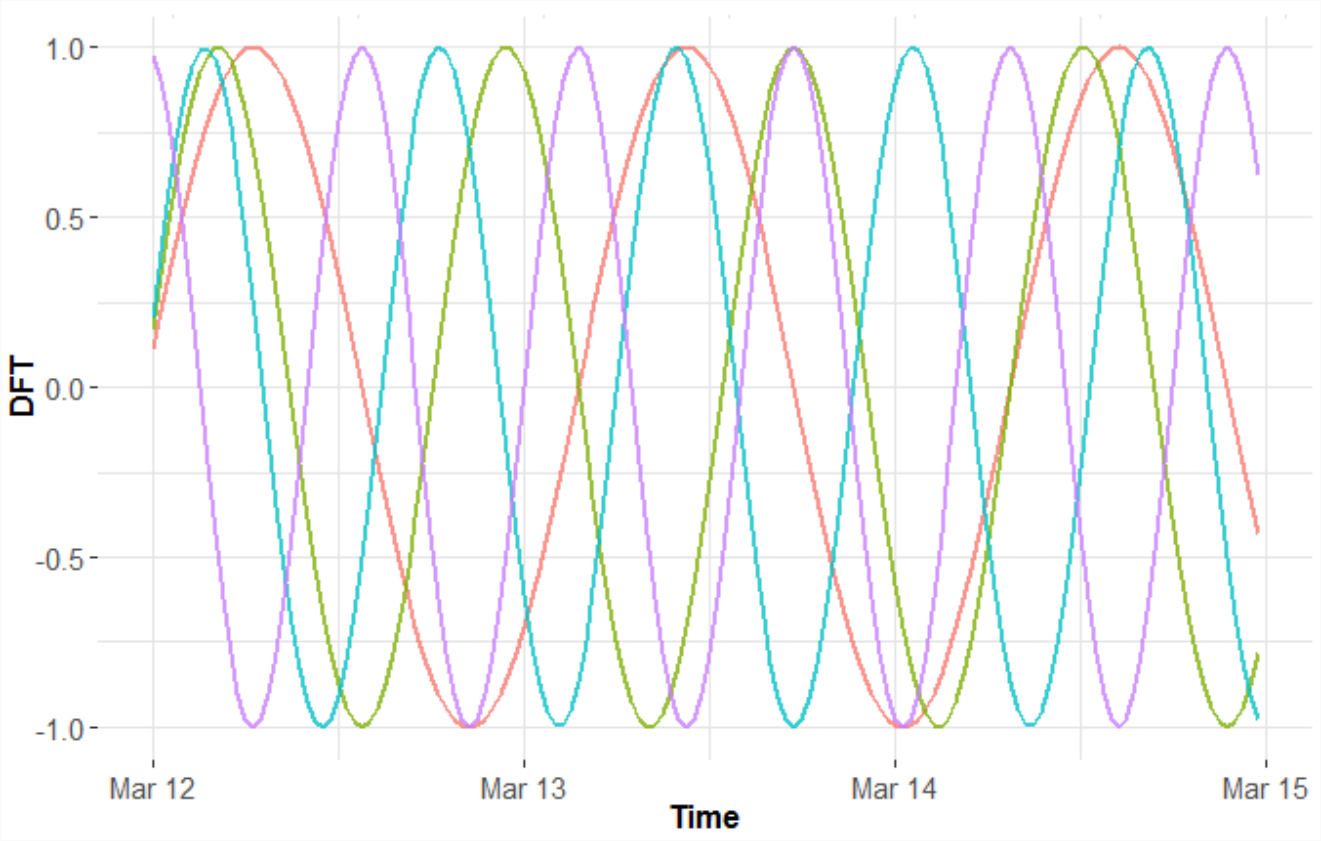
ARIMA, exponential smoothing and co.

- Not best for multiple seasonal time series.
- Solution: Creation of separate models for different days in week.
- Solution: Set higher period - $7 * period$. We need at least 3 periods in training set.
- Solution: Double-seasonal Holt-Winters (**`dshw, tbats`**).
- Solution: SARIMA. ARIMAX - ARIMA plus extra predictors by FFT (**`auto.arima(ts, xreg)`**).

Fourier transform

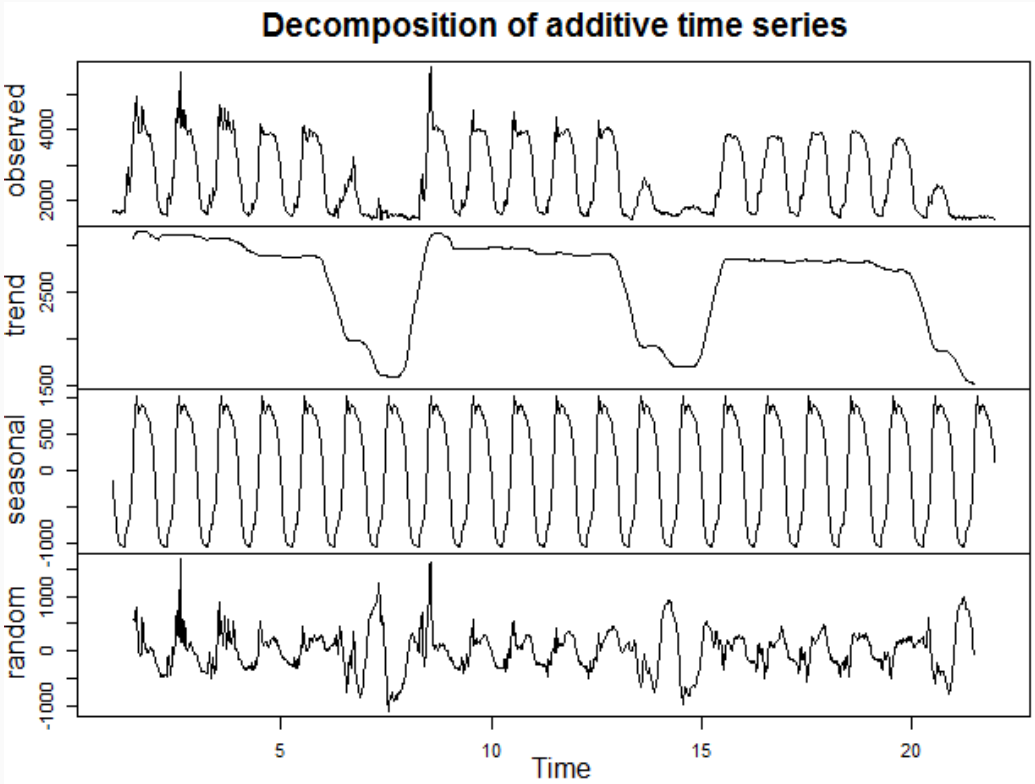
`fourier` in `forecast`

```
load_msts <- msts(load,
       seasonal.periods = c(freq, freq * 7))
fourier(load_msts, K = c(6, 6*2))
```



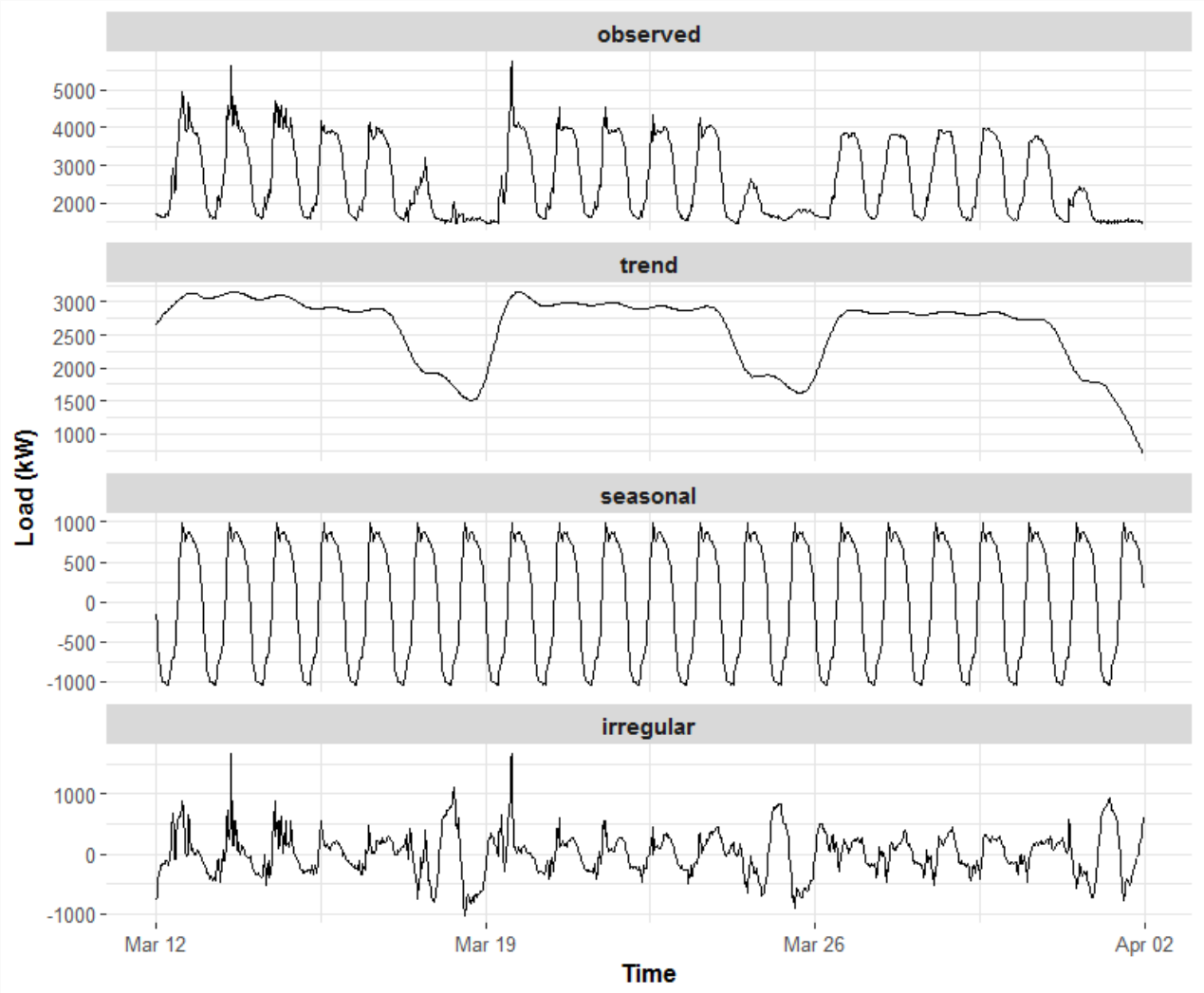
Time series analysis

- Solution: Decomposition of time series to three parts - seasonal, trend, remainder (noise).
- Many different methods. STL decomposition (seasonal decomposition of time series by loess).
- Moving Average (**`decompose`**). Exponential smoothing...



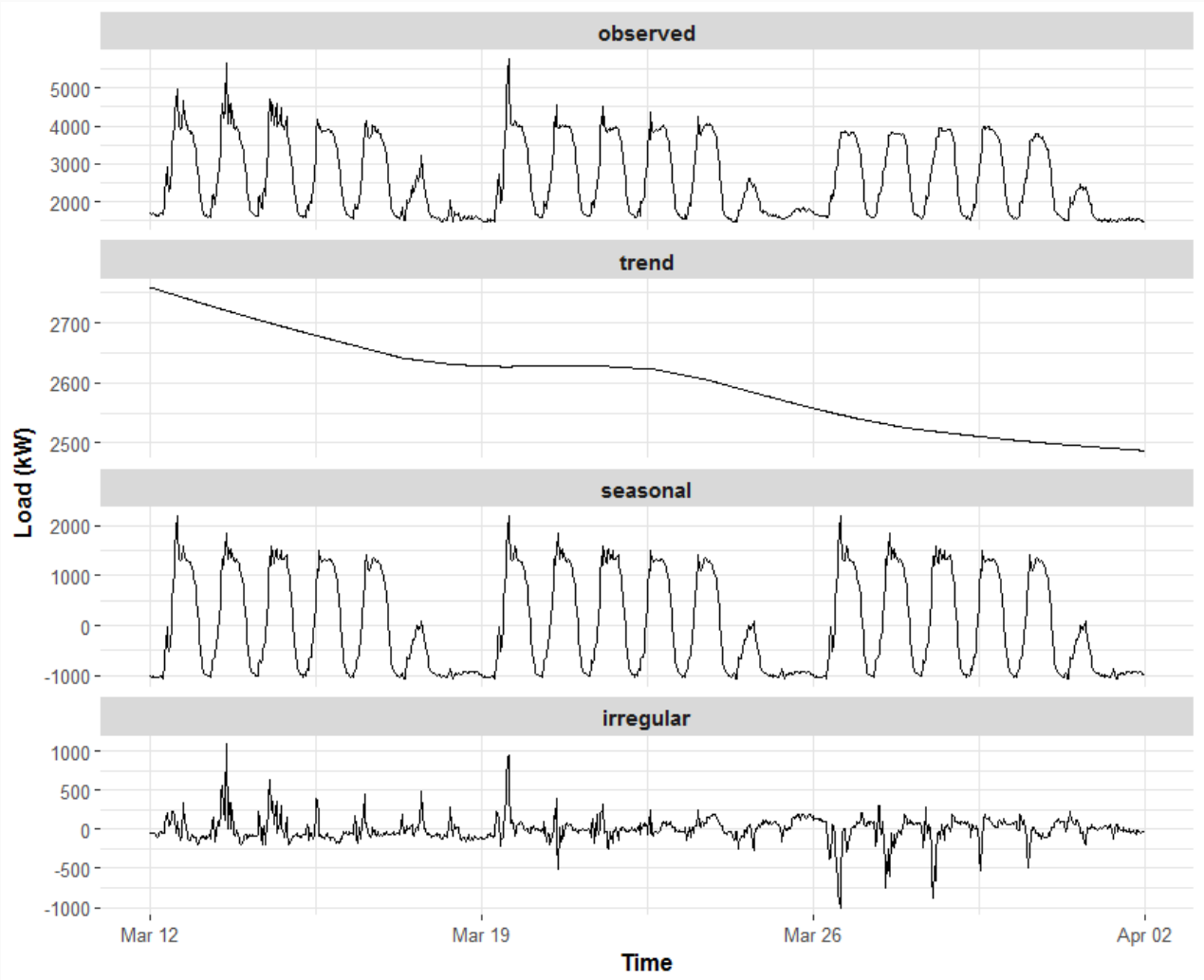
Decomposition - STL

`stl`, `ggseas` with period 48



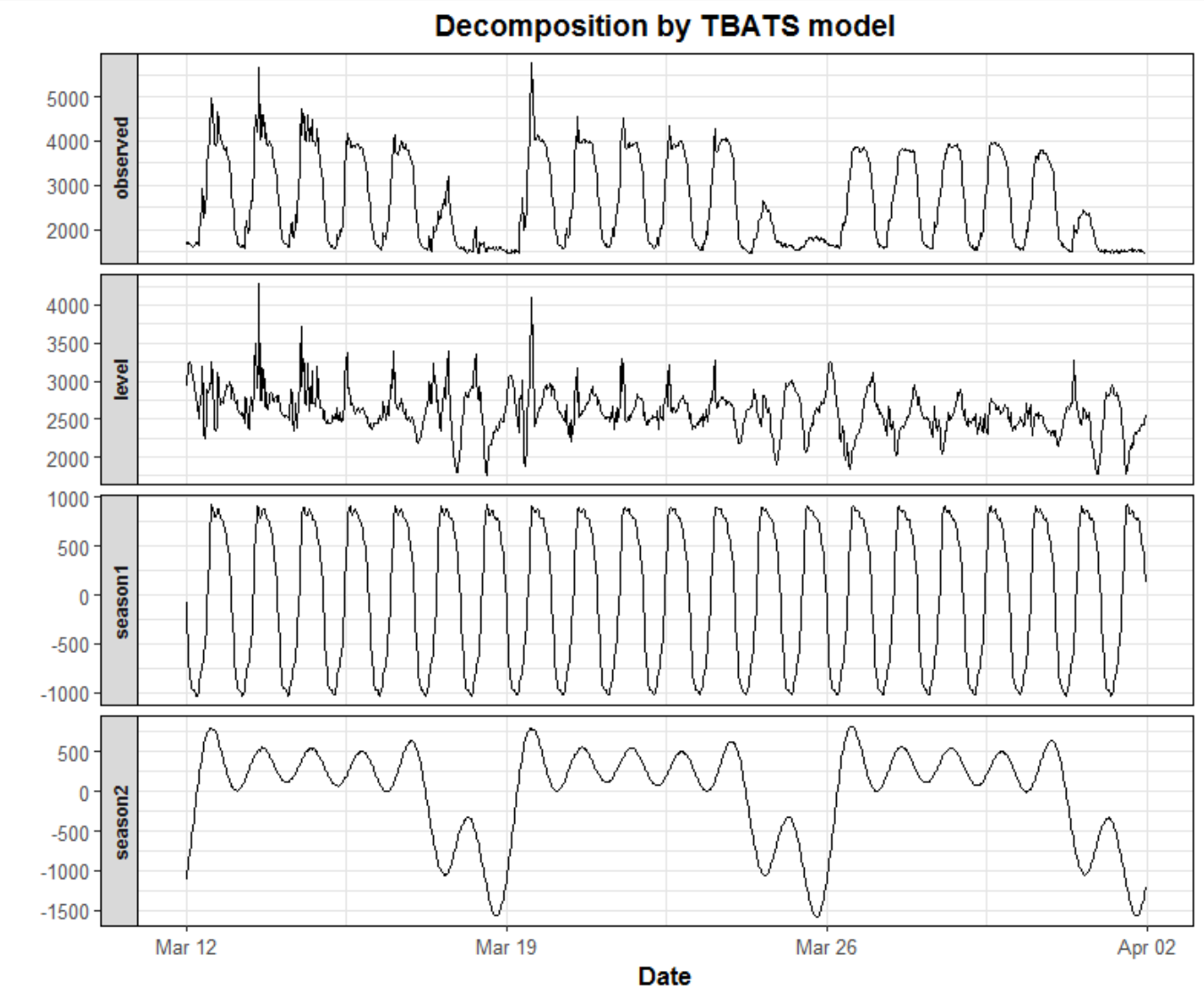
Decomposition - STL vol.2

`stl`, `ggseas` with period $48 * 7$



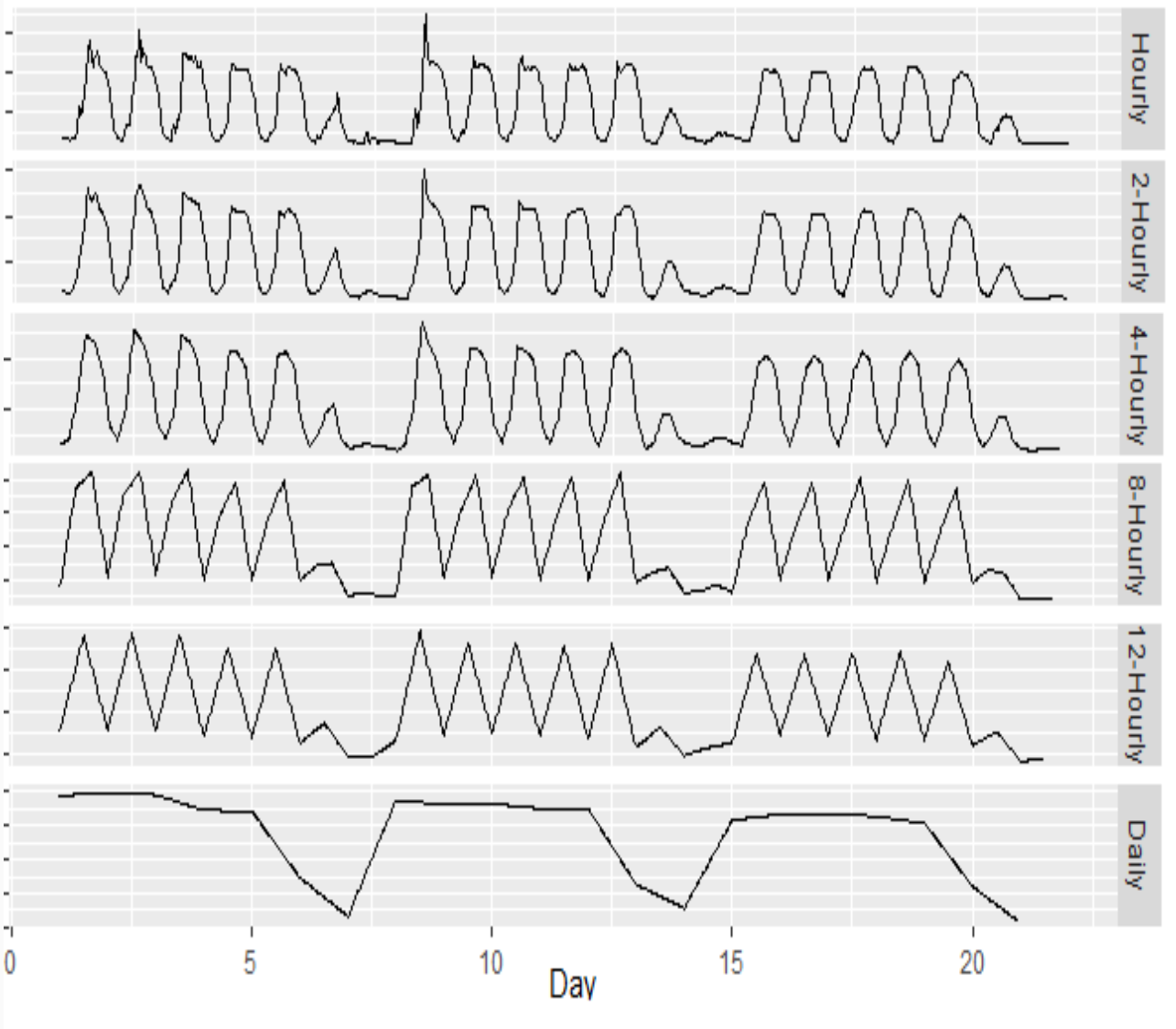
Decomposition - EXP

`tbats` in `forecast`



Temporal Hierarchical Forecasting

`thief` - `http://robjhyndman.com/hyndsight/thief/`



Regression methods

- MLR (OLS), RLM [1] (M-estimate), SVR [2]
- Dummy (binary) variables[3]

	Load	D1	D2	D3	D4	D5	D6	D7	D8	D9	...	W1	W2	...
1:	0.402	1	0	0	0	0	0	0	0	0		1	0	
2:	0.503	0	1	0	0	0	0	0	0	0		1	0	
3:	0.338	0	0	1	0	0	0	0	0	0		1	0	
4:	0.337	0	0	0	1	0	0	0	0	0		1	0	
5:	0.340	0	0	0	0	1	0	0	0	0		1	0	
6:	0.340	0	0	0	0	0	1	0	0	0		1	0	
7:	0.340	0	0	0	0	0	0	1	0	0		1	0	
8:	0.338	0	0	0	0	0	0	0	1	0		1	0	
9:	0.339	0	0	0	0	0	0	0	0	1		1	0	
⋮											⋱	⋮	⋮	

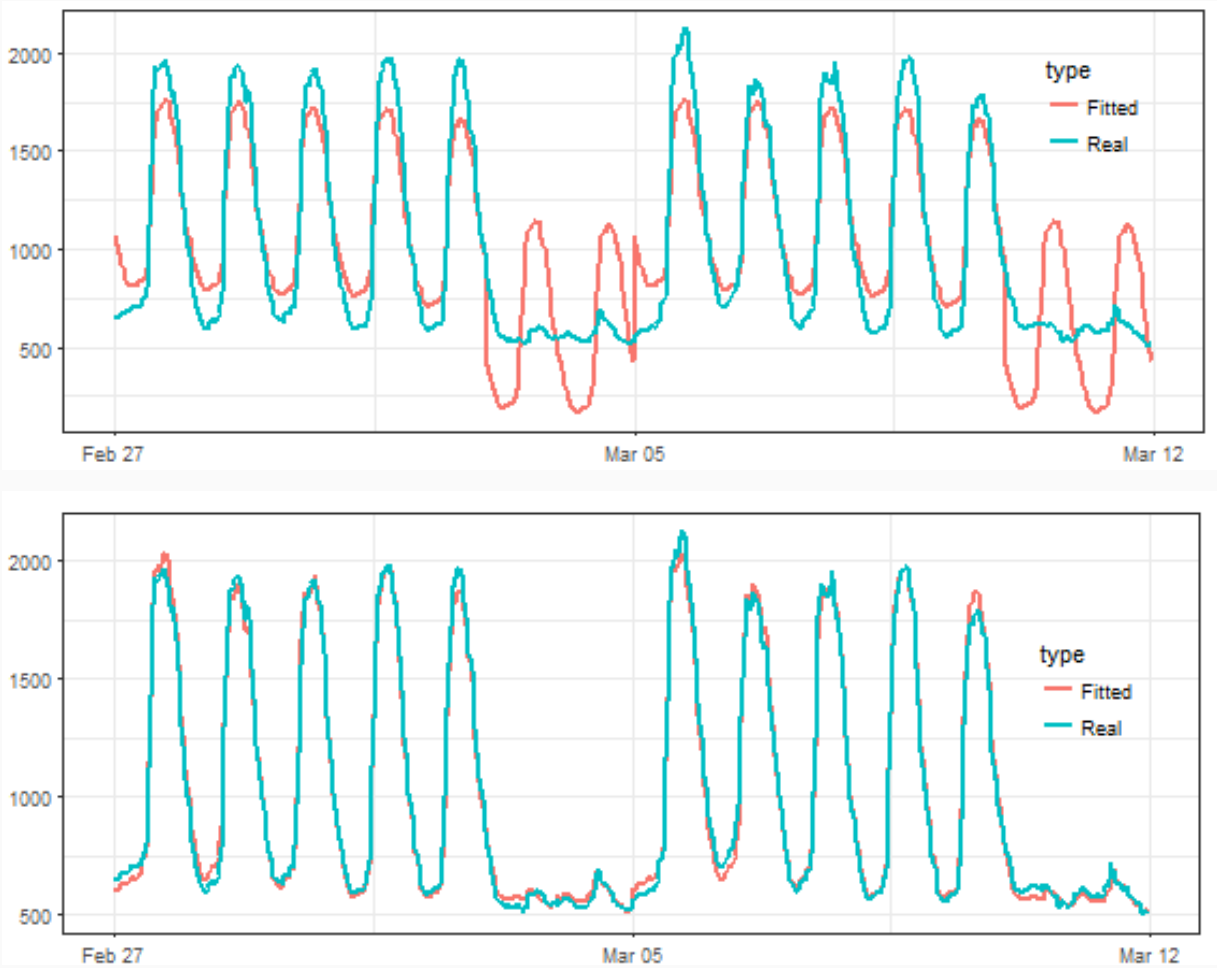
[1] rlm - MASS

[2] **`ksvm(..., eps, C)`** - kernlab

[3] **`model.matrix(~ 0 + as.factor(Day) + as.factor(Week))`**

Linear Regression

Interactions! MLR and GAM [4].



[4] https://petolau.github.io/

Trees and Forests

- Dummy variables aren't appropriate.
- CART [5], Extremely Randomized Trees [6], Bagging [7].

Daily and Weekly seasonal vector:

$$\boldsymbol{Day} = (1, 2, \ldots, \mathit{freq}, 1, 2, \ldots, \mathit{freq}, \ldots)$$

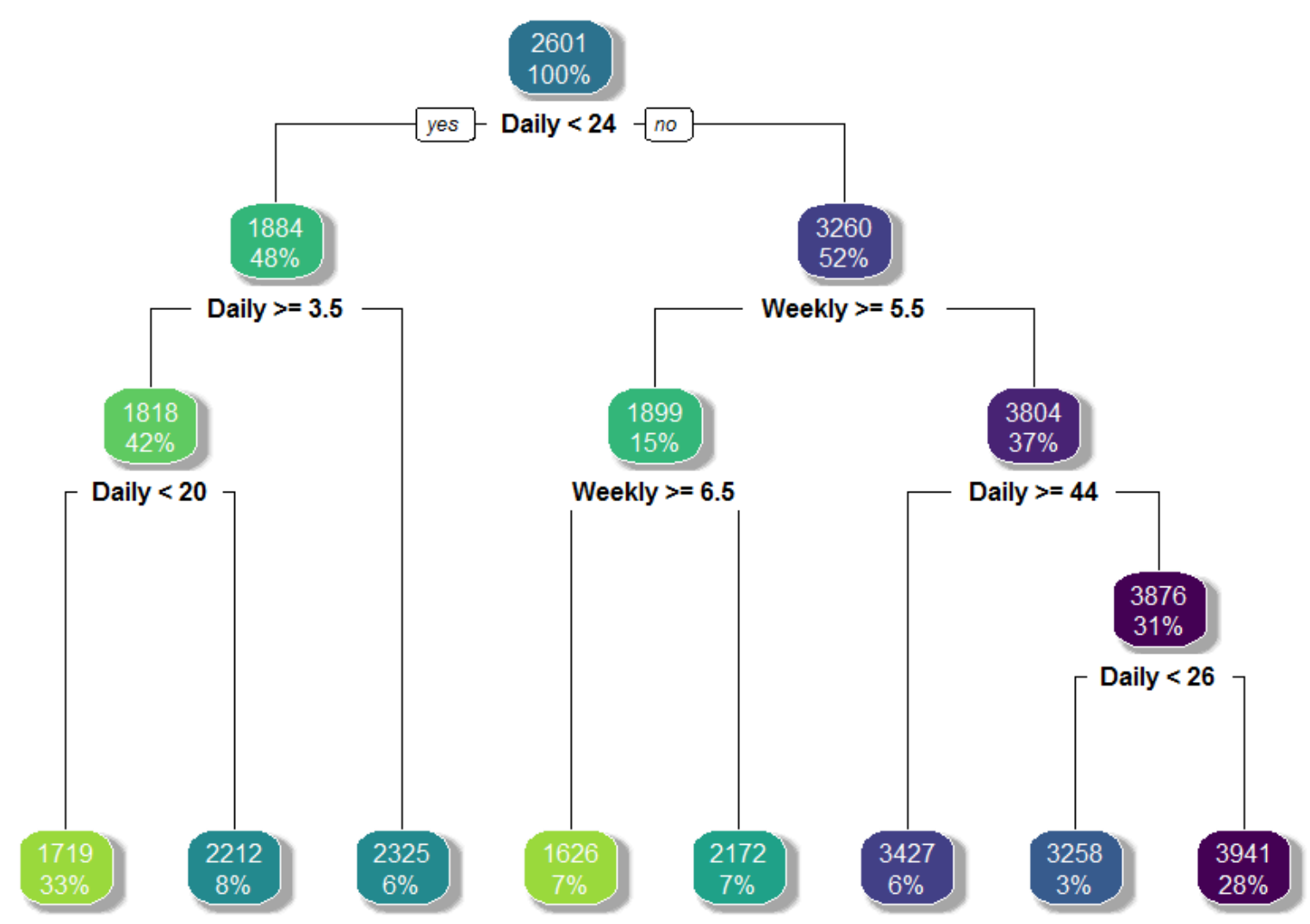
$$\boldsymbol{Week} = (1, 1, \ldots, 1, 2, 2, \ldots, 2, \ldots, 7, 1, 1, \ldots),$$

where *freq* is a period (48 resp. 96).

[5] `rpart`
[6] `extraTrees`
[7] `RWeka`

rpart

`rpart.plot::rpart.plot`



Forests and ctree

- Ctree (**party**), Extreme Gradient Boosting (**xgboost**), Random Forest.

Daily and weekly seasonal vector in form of sine and cosine:

$$\boldsymbol{Day} = (1, 2, \ldots, freq, 1, 2, \ldots, freq, \ldots)$$

$$\boldsymbol{Week} = (1, 1, \ldots, 1, 2, 2, \ldots, 2, \ldots, 7, 1, 1, \ldots)$$

$$\frac{\sin(2\pi \frac{\boldsymbol{Day}}{freq}) + 1}{2} \quad \text{resp.} \quad \frac{\cos(2\pi \frac{\boldsymbol{Day}}{freq}) + 1}{2}$$

$$\frac{\sin(2\pi \frac{\boldsymbol{Week}}{7}) + 1}{2} \quad \text{resp.} \quad \frac{\cos(2\pi \frac{\boldsymbol{Week}}{7}) + 1}{2},$$

where *freq* is a period (48 resp. 96).

- lag, seasonal part from decomposition

Neural nets

- Feedforward, recurrent, multilayer perceptron (MLP), deep.
- Lagged load. Denoised (detrended) load.
- Model for different days - similar day approach.
- Sine and cosine
- Fourier

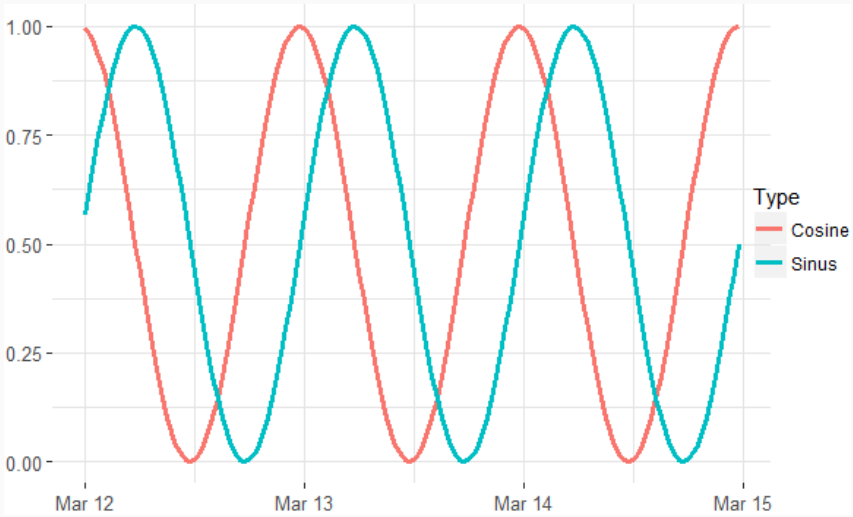


Figure 1: Day

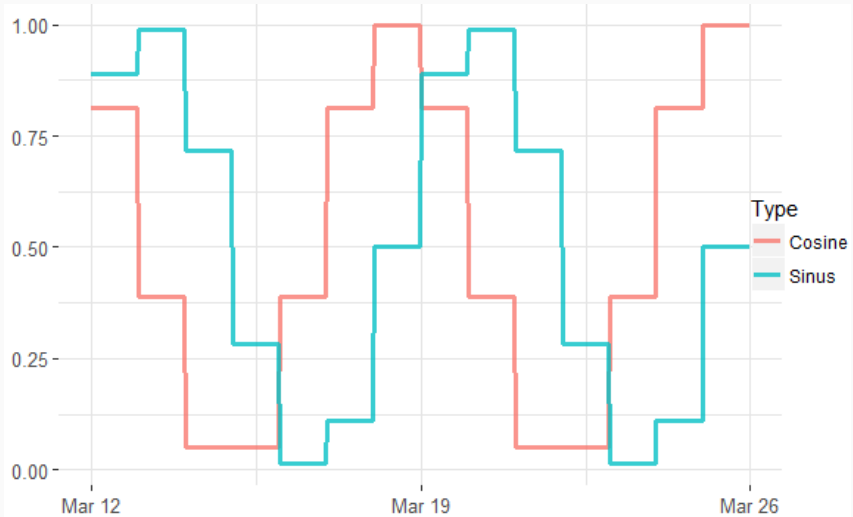


Figure 2: Week

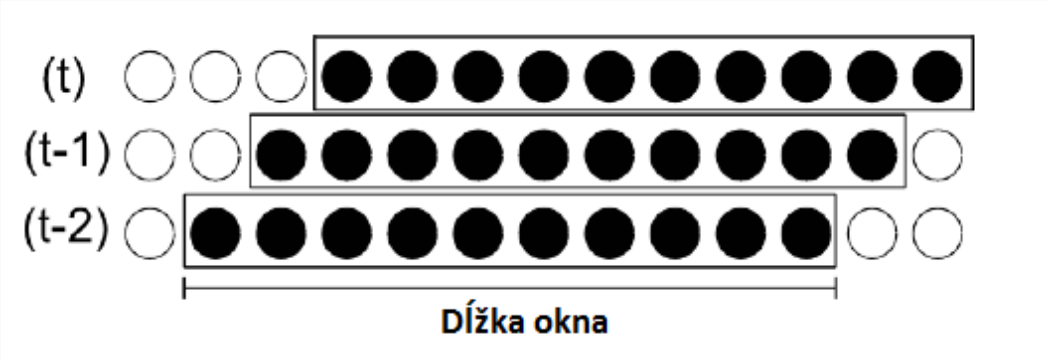
Ensemble learning

- We can't determine which model is best. Overfitting.
- Wisdom of crowds - set of models. Model weighting $\Rightarrow$ linear combination.
- Adaptation of weights of prediction methods based on prediction error - median weighting.

$$e_j^t = median(|\, \boldsymbol{x}^t - \hat{\boldsymbol{x}}_j^t \,|)$$

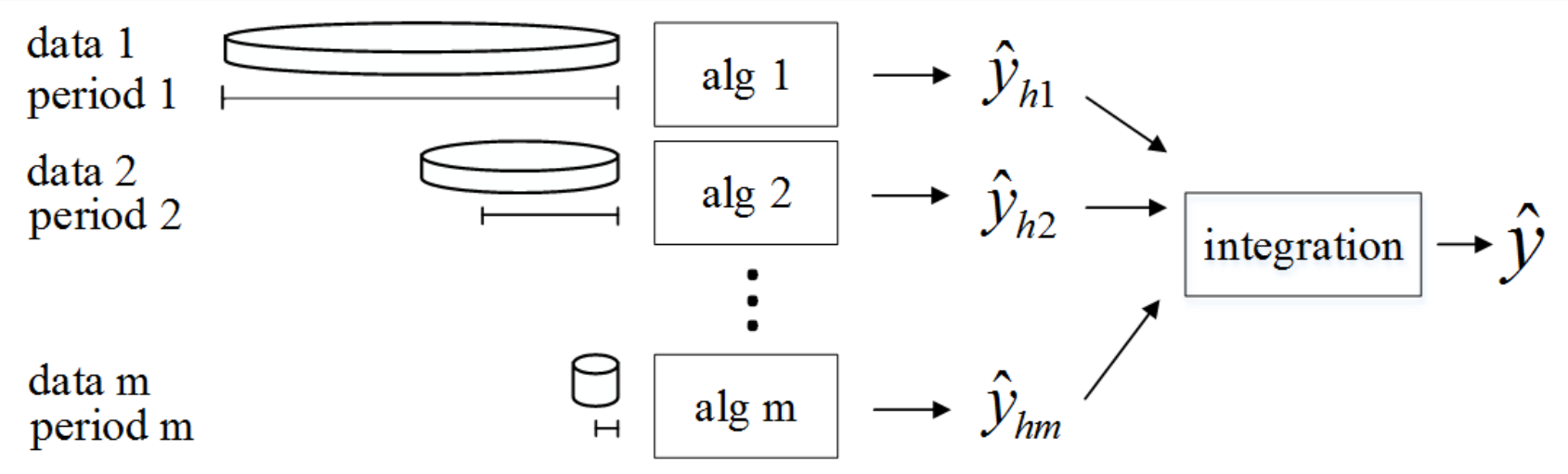
$$w_j^{t+1} = w_j^t \frac{median(\boldsymbol{e}^t)}{e_j^t}$$

Training set – sliding window.



Ensemble learning

- Heterogeneous ensemble model.

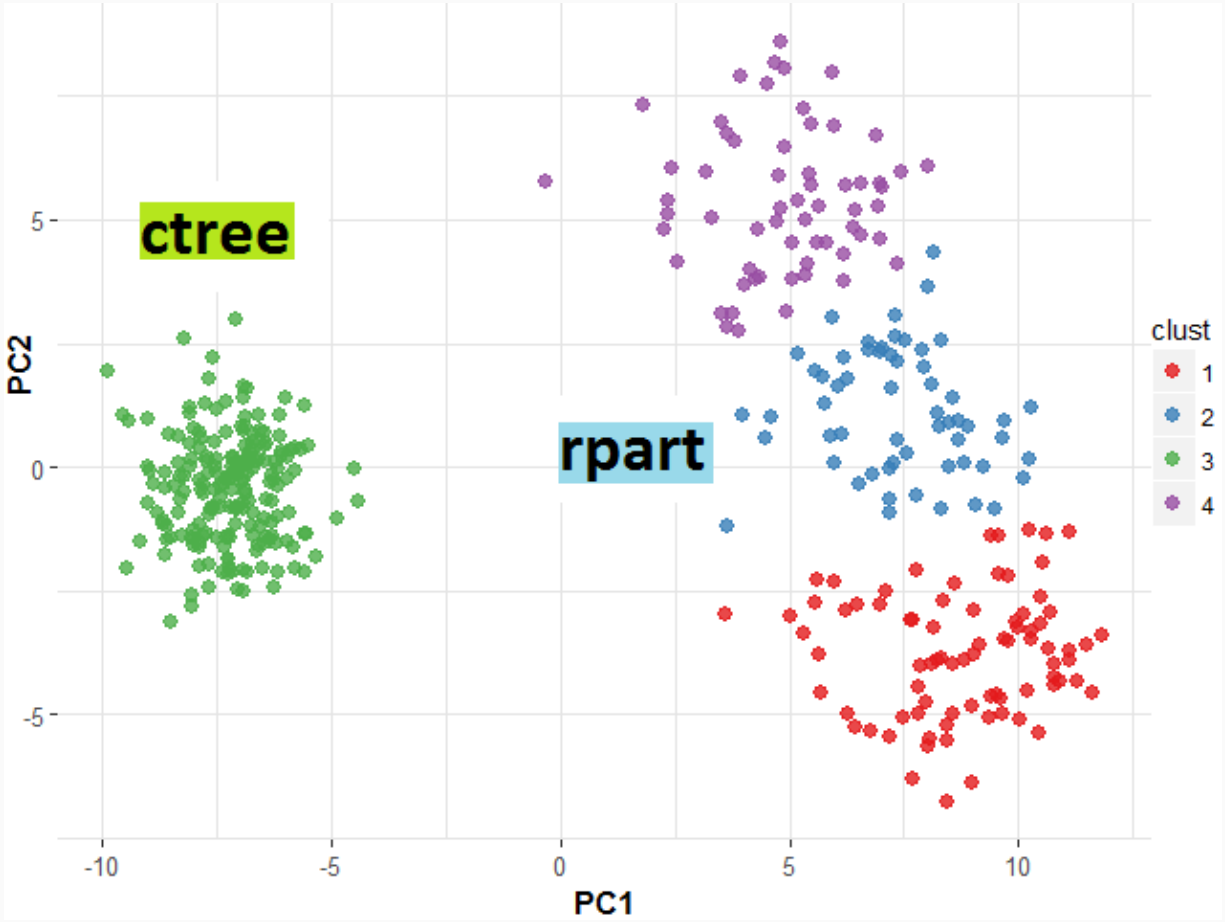


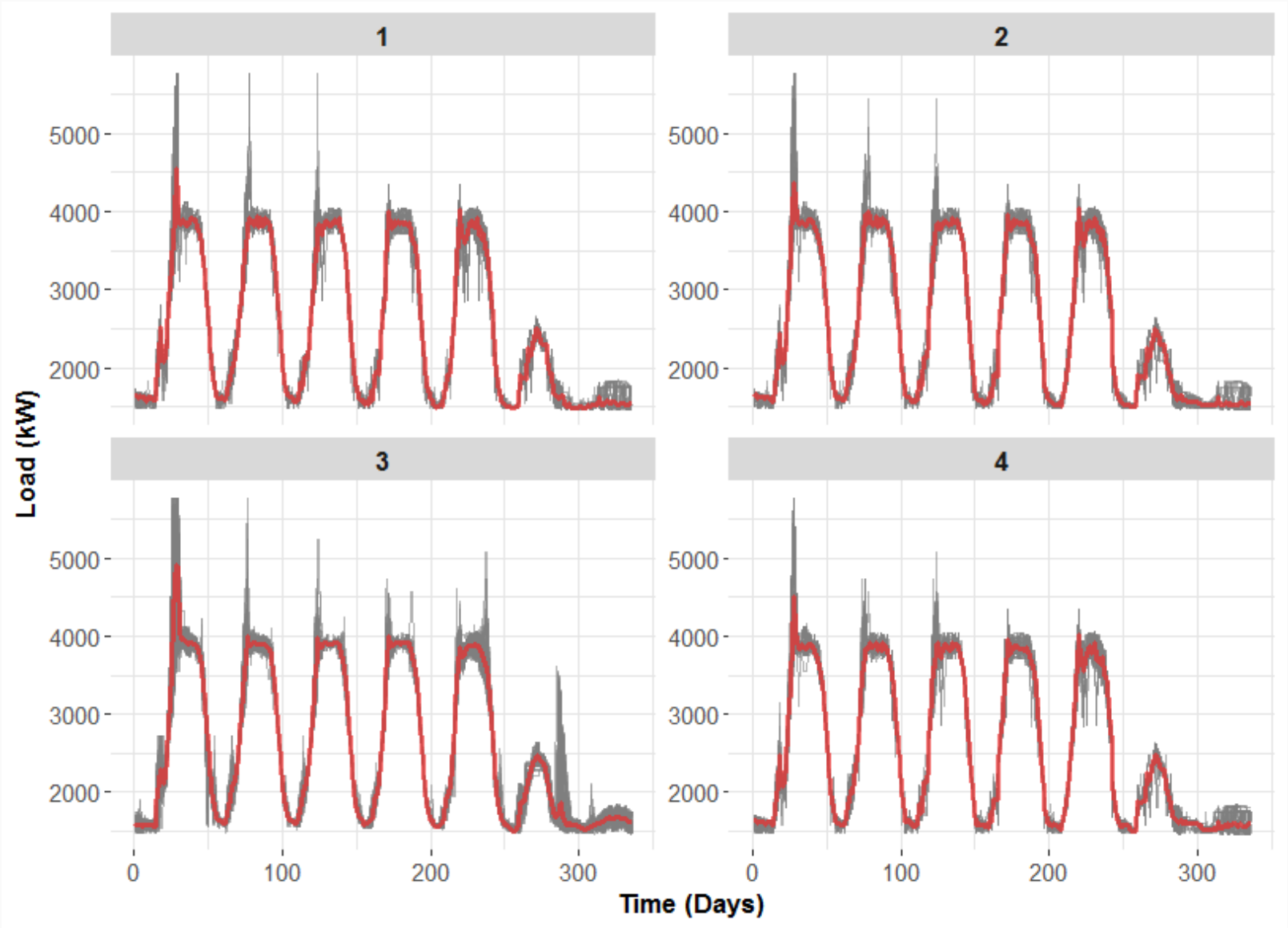
- Improvement of 8 – 12%.
- Weighting of methods – optimization task.
  Genetic algorithm, PSO and others.
- Improvement of 5,5%. Possible overfit.

Ensemble learning with clustering

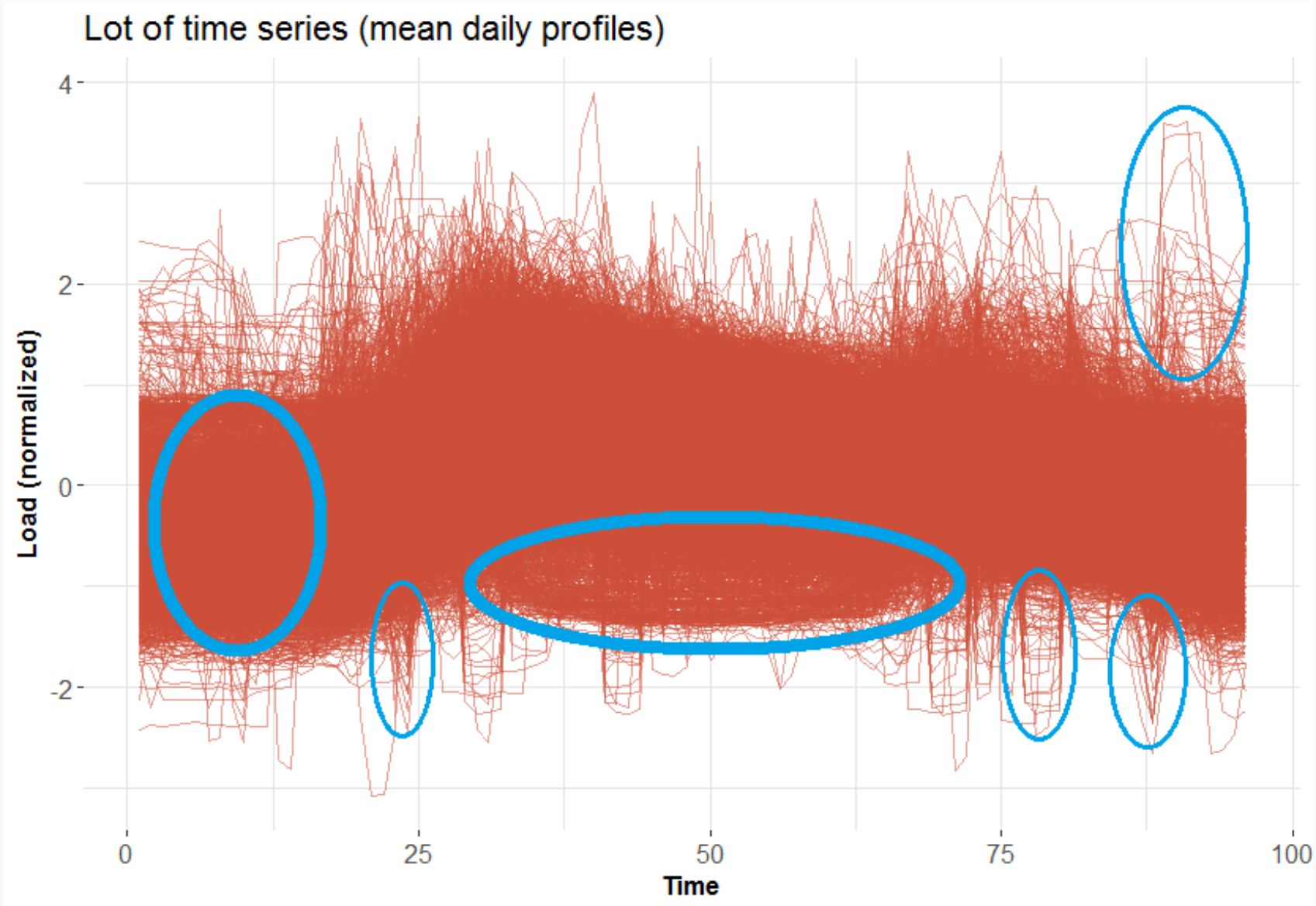
`ctree` + **`rpart`** trained on different data sets (sampling with replacement). ”Weak learners”.

Pre-process - PCA $\rightarrow$ K-means. + unsupervised.



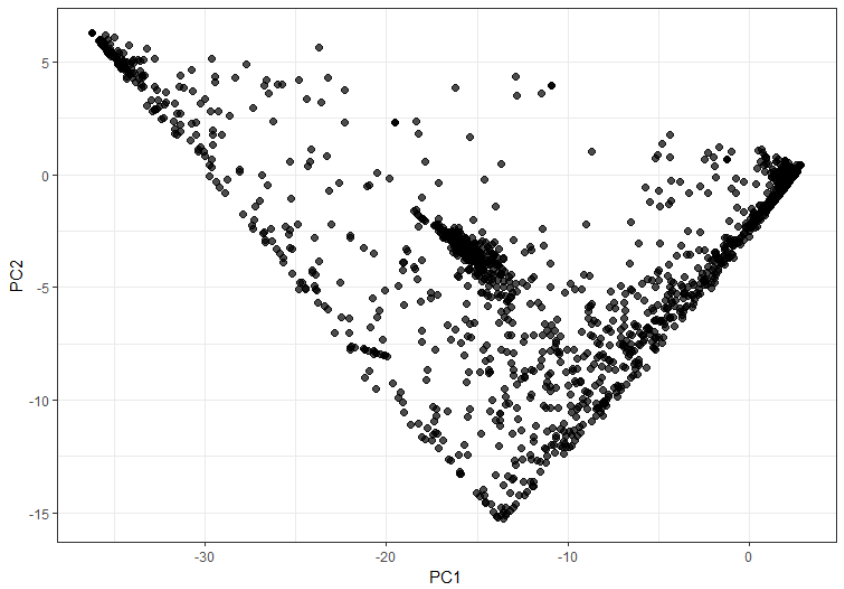

1
2
3
4
Load (kW)
5000
4000
3000
2000
0
100
200
300
Time (Days)

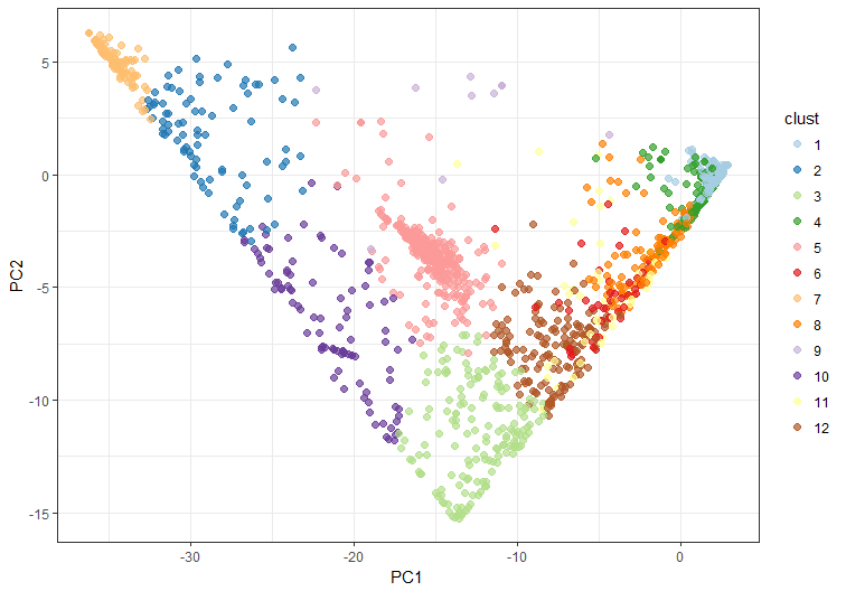
R <- ”Cluster Analysis”


Lot of time series (mean daily profiles)
Load (normalized)
4
2
0
-2
0
25
50
75
100
Time

Creation of more predictable groups of consumers

Cluster Analysis! Unsupervised learning...





Cluster analysis in smart grids

Good for:

Cluster analysis in smart grids

Good for:

- Creation of consumer profiles, which can help by recommendation to customers to save electricity.

Cluster analysis in smart grids

Good for:

- Creation of consumer profiles, which can help by recommendation to customers to save electricity.
- Detection of anomalies.

Cluster analysis in smart grids

Good for:

- Creation of consumer profiles, which can help by recommendation to customers to save electricity.
- Detection of anomalies.
- Neater monitoring of the smart grid (as a whole).

Cluster analysis in smart grids

Good for:

- Creation of consumer profiles, which can help by recommendation to customers to save electricity.
- Detection of anomalies.
- Neater monitoring of the smart grid (as a whole).
- Emergency analysis.

Cluster analysis in smart grids

Good for:

- Creation of consumer profiles, which can help by recommendation to customers to save electricity.
- Detection of anomalies.
- Neater monitoring of the smart grid (as a whole).
- Emergency analysis.
- Generation of more realistic synthetic data.

Cluster analysis in smart grids

Good for:

- Creation of consumer profiles, which can help by recommendation to customers to save electricity.
- Detection of anomalies.
- Neater monitoring of the smart grid (as a whole).
- Emergency analysis.
- Generation of more realistic synthetic data.
- Last but not least, improve prediction methods to produce more accurate predictions.

Aggregation with clustering

Aggregation with clustering

1. Set of time series of electricity consumption

Aggregation with clustering

1. Set of time series of electricity consumption
2. Normalization (z-score)

Approach

Aggregation with clustering

1. Set of time series of electricity consumption
2. Normalization (z-score)
3. Computation of representations of time series

Aggregation with clustering

1. Set of time series of electricity consumption
2. Normalization (z-score)
3. Computation of representations of time series
4. Determination of optimal number of clusters K

Aggregation with clustering

1. Set of time series of electricity consumption
2. Normalization (z-score)
3. Computation of representations of time series
4. Determination of optimal number of clusters K
5. Clustering of representations

Aggregation with clustering

1. Set of time series of electricity consumption
2. Normalization (z-score)
3. Computation of representations of time series
4. Determination of optimal number of clusters K
5. Clustering of representations
6. Summation of K time series by found clusters

Aggregation with clustering

1. Set of time series of electricity consumption
2. Normalization (z-score)
3. Computation of representations of time series
4. Determination of optimal number of clusters K
5. Clustering of representations
6. Summation of K time series by found clusters
7. Training of K forecast models and the following forecast

Approach

Aggregation with clustering

1. Set of time series of electricity consumption
2. Normalization (z-score)
3. Computation of representations of time series
4. Determination of optimal number of clusters K
5. Clustering of representations
6. Summation of K time series by found clusters
7. Training of K forecast models and the following forecast
8. Summation of forecasts and evaluation

Pre-processing

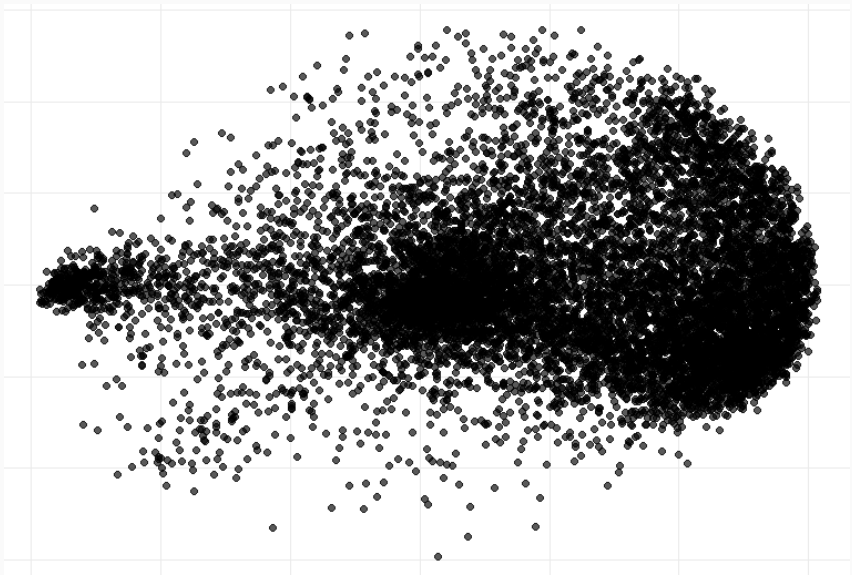
Normalization of time series - consumers.

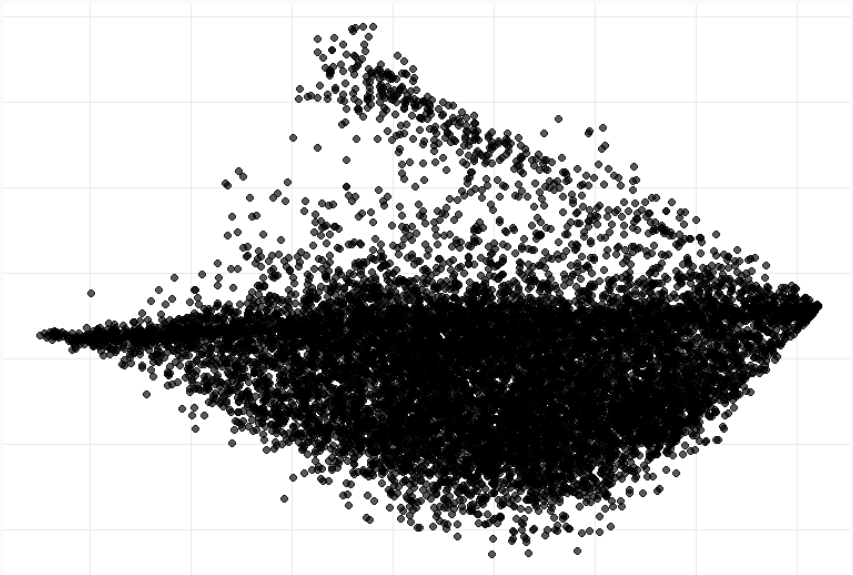
Best choice is to use z-score:

$$\frac{x - \mu}{\sigma}$$

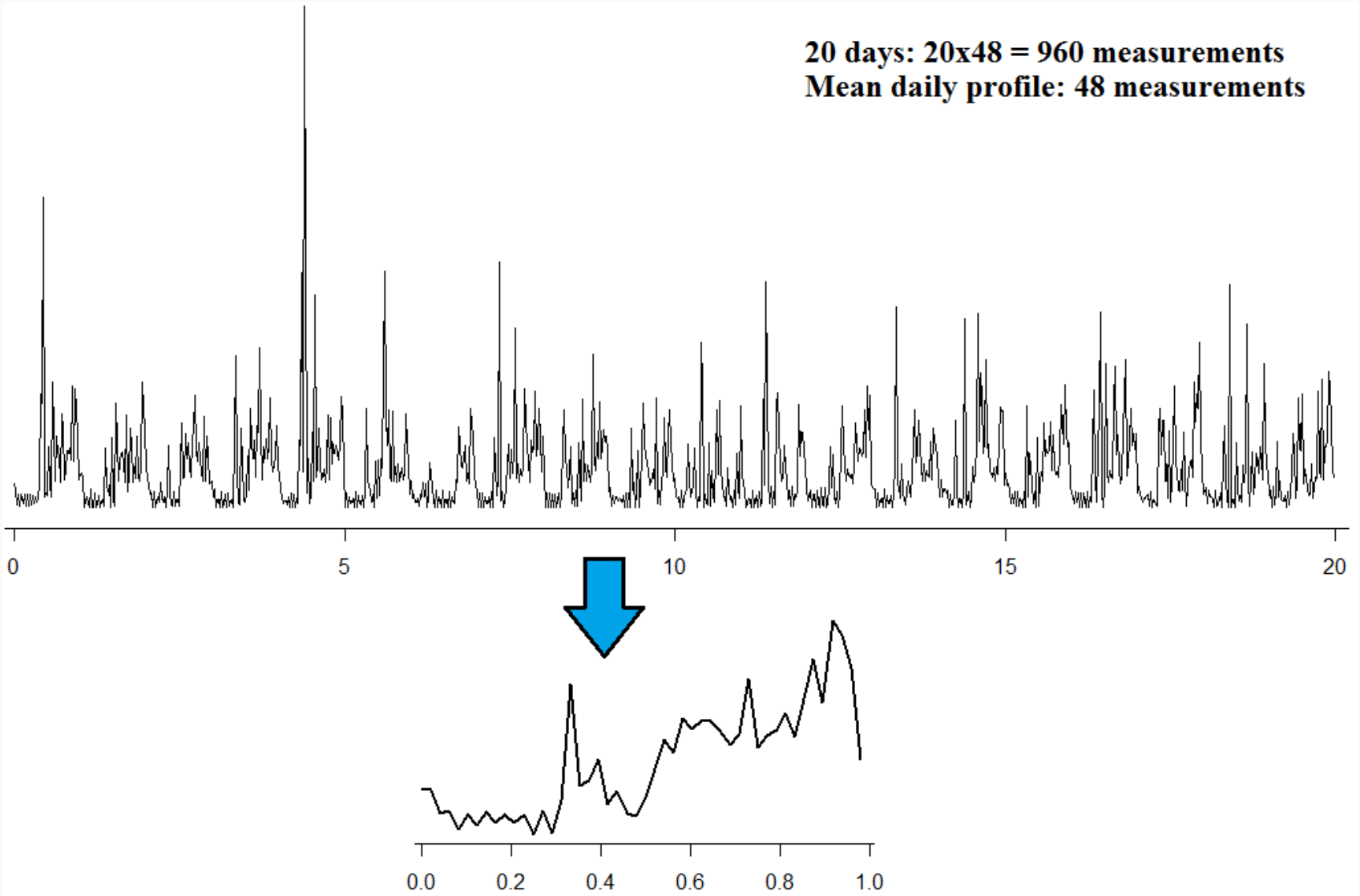
Alternative:

$$\frac{x}{\max(x)}$$





Representations of time series



Representations of time series

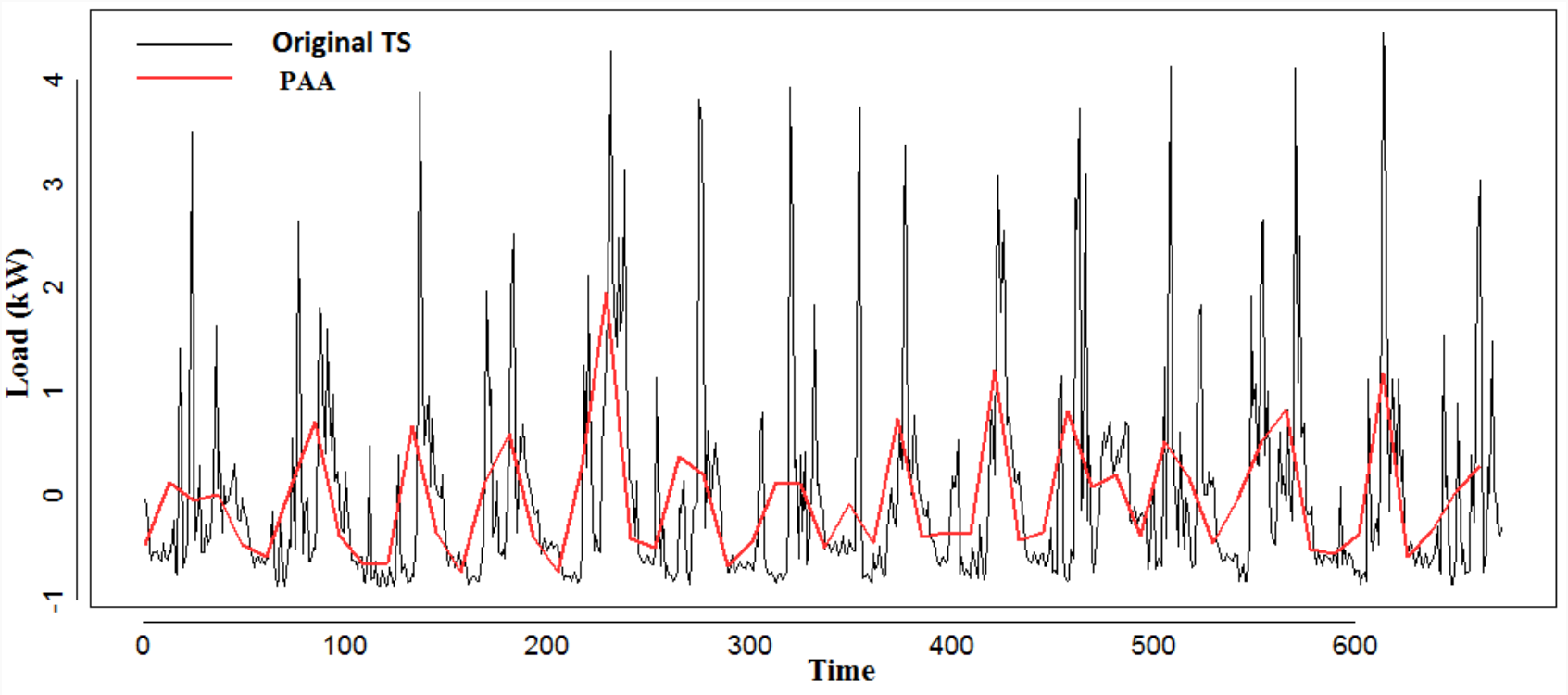
Why time series representations?

1. Reduce memory load.
2. Accelerate subsequent machine learning algorithms.
3. Implicitly remove noise from the data.
4. Emphasize the essential characteristics of the data.

Methods of representations of time series

PAA - Piecewise Aggregate Approximation. Non data adaptive.
n -> d. $\hat{X} = (\hat{x}_1, \ldots, \hat{x}_d)$.

$$\hat{x}_i = \frac{d}{n} \sum_{j=(n/d)(i-1)+1}^{(n/d)i} x_j.$$



Not only average. Median, standard deviation, maximum...
DWT, DFT...

Back to models - model based methods

- Representations based on statistical model.
- Extraction of regression coefficients $\Rightarrow$ creation of daily profiles.
- Creation of representation which is long as frequency of time series.

$$y_i = \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_{freq} x_{ifreq} + \beta_{week1} x_{iweek1} + \cdots + \beta_{week7} x_{iweek7} + \varepsilon_i,$$

$$where\ i = 1, \ldots, n$$

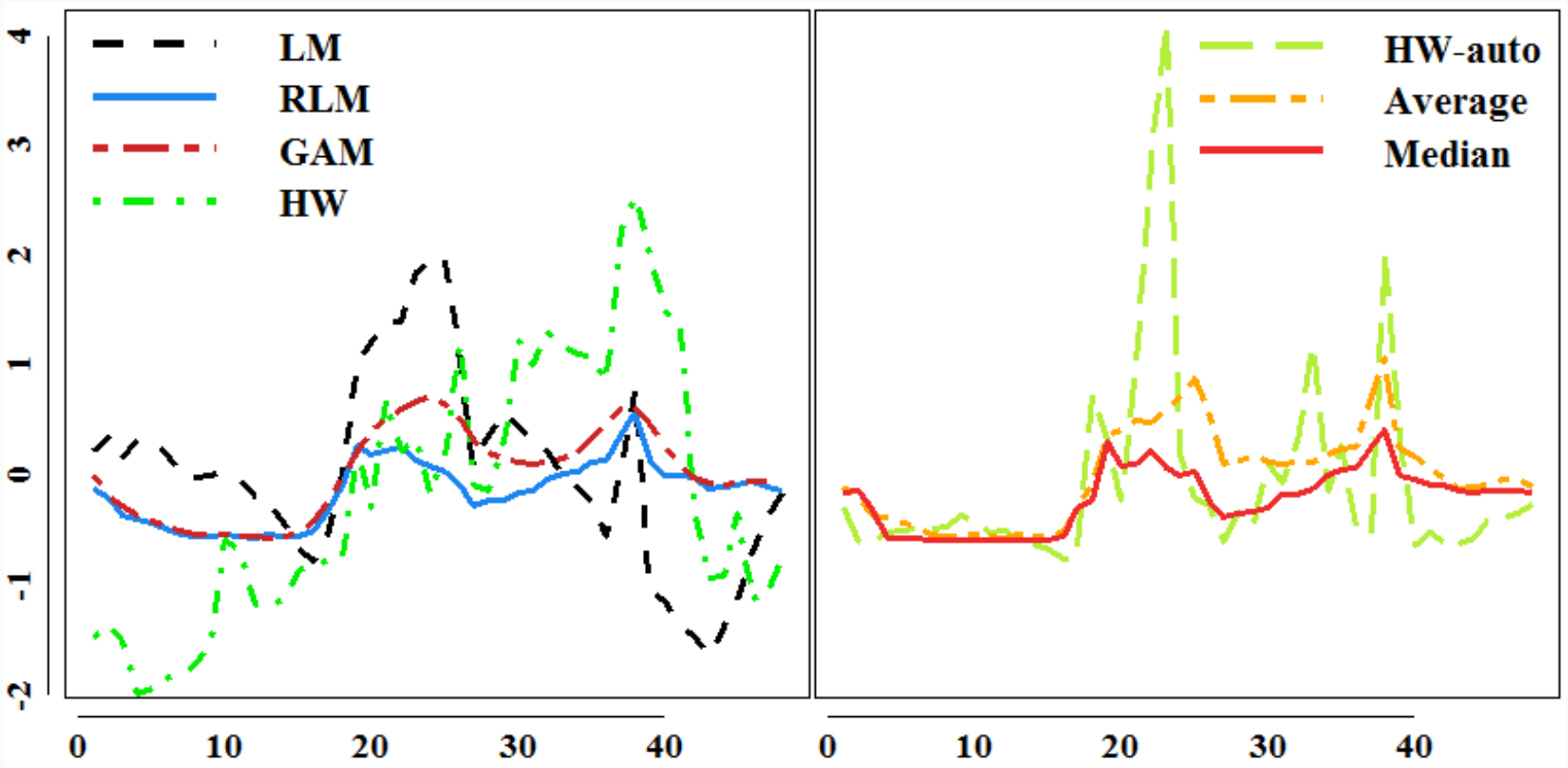
New representation: $\hat{\boldsymbol{\beta}} = (\hat{\beta}_1, \ldots, \hat{\beta}_{freq}, \hat{\beta}_{week1}, \ldots, \hat{\beta}_{week7})$.

Possible applicable methods:
MLR, Robust Linear Model, L1-regression, Generalized Additive Model (GAM)...

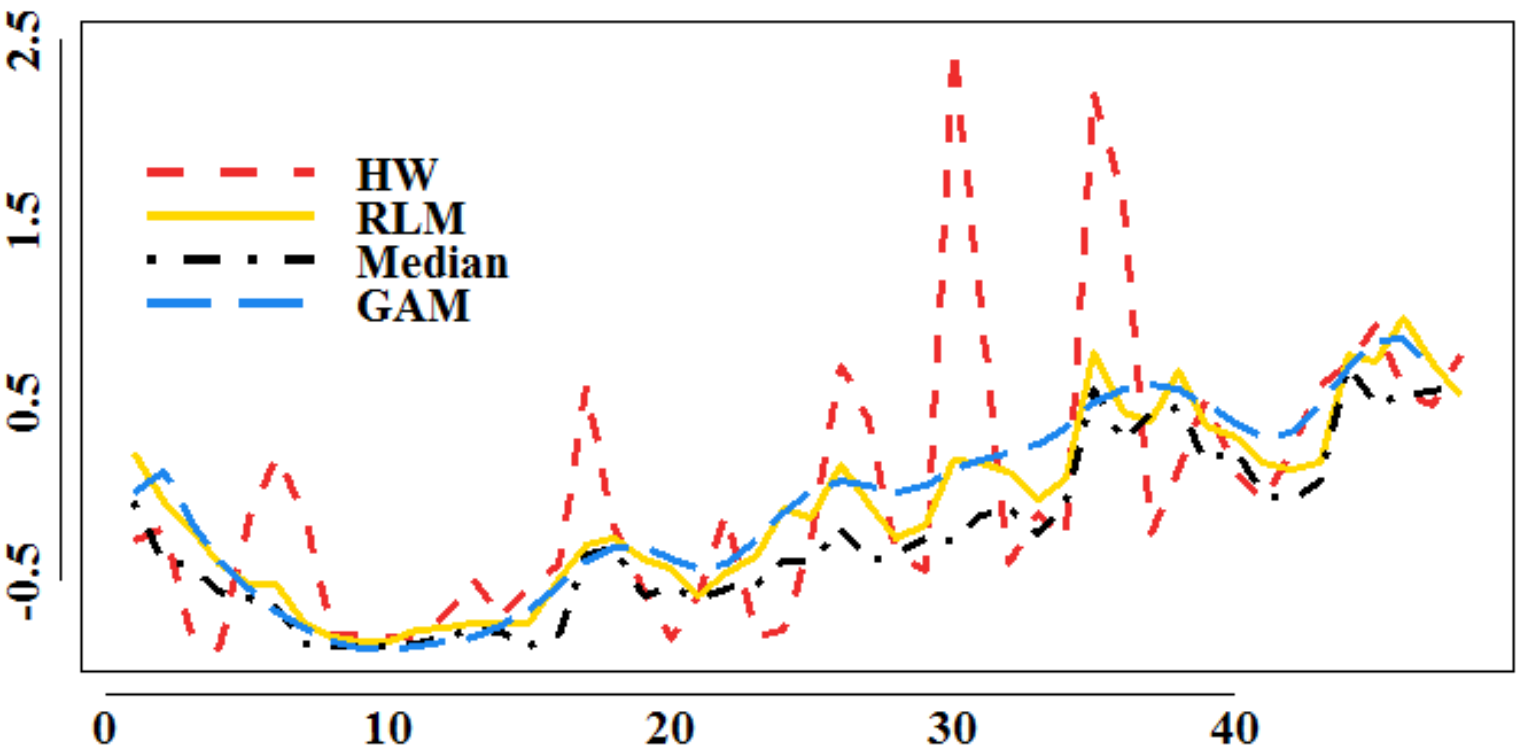
Model based methods

- Triple Holt-Winters Exponential Smoothing. Last seasonal coefficients as representation.
    1. Smoothing factors can be set automatically or manually.
- Mean and median daily profile.

Comparison


LM
RLM
GAM
HW
HW-auto
Average
Median
4
3
2
1
0
-1
-2
0
10
20
30
40
0
10
20
30
40

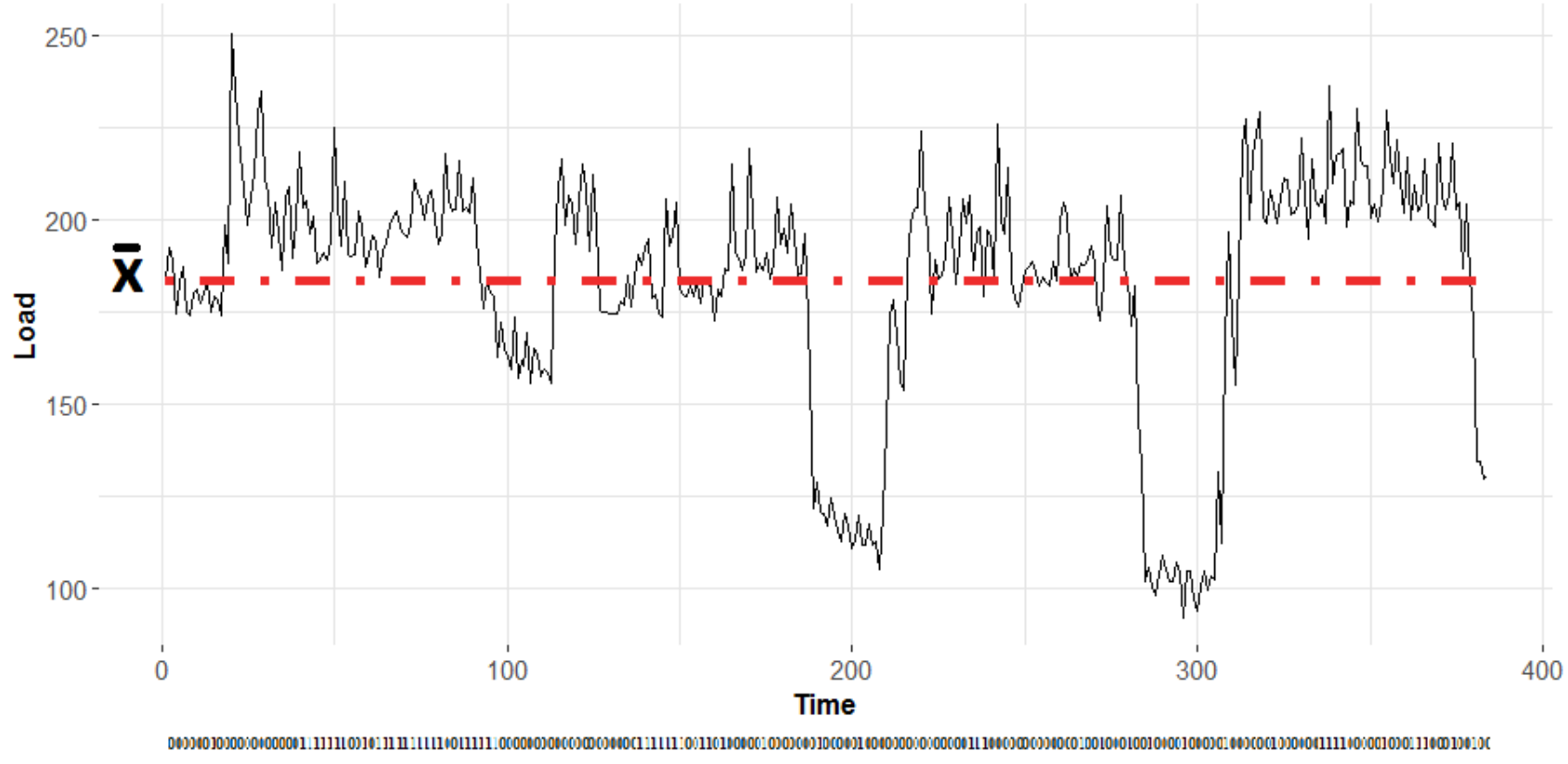
Comparison



Clipping - bit level representation

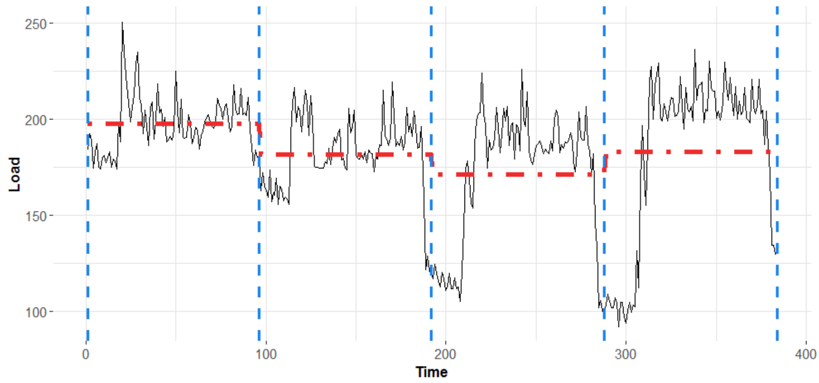
Data dictated.

$$\hat{x}_t = \begin{cases} 1 & \text{if } x_t > \mu \\ 0 & \text{otherwise} \end{cases}$$



Clipping - RLE

RLE - Run Length Encoding. Sliding window - one day.



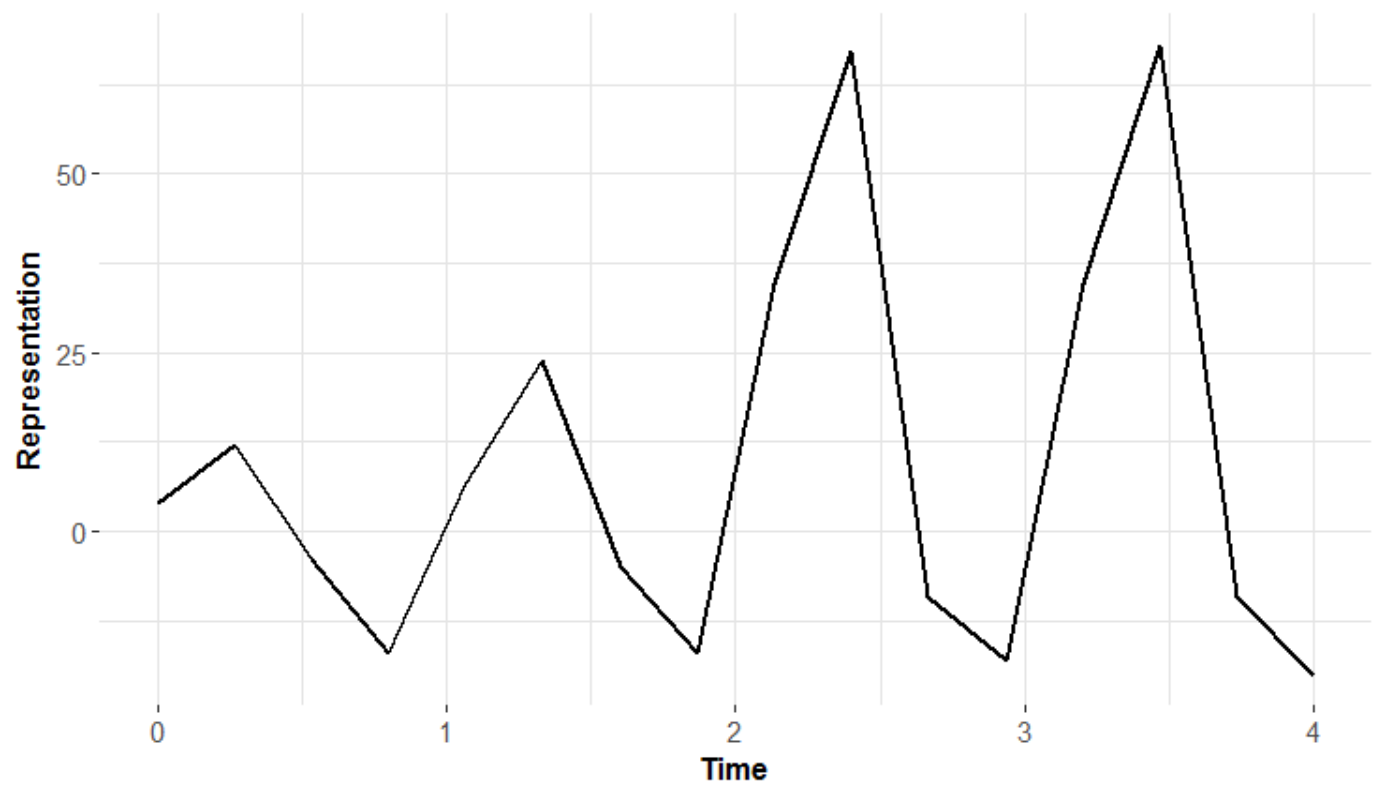
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	1	0	1	0	1	0	1	0	1	0	1	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
6	1	12	7	2	1	1	10	2	6	20	7	2	2	1	1	5	1	7	1	1	4	1	15	3	13	1	2	1	3	1	2	1	1	3	1	5	1	6	1	6	4	5	1	3	3	3	1	2	1	2

Clipping - final representation

1	7	1	10	6	7	2	1	1	1	1	3	1	1	1	1	1	1	1	4	1	3	1	1

6	12	2	1	2	20	2	1	5	7	0	4	15	13	2	3	2	1	3	5	6	6	5	3

Feature extraction: average, maximum, -average, -maximum ...



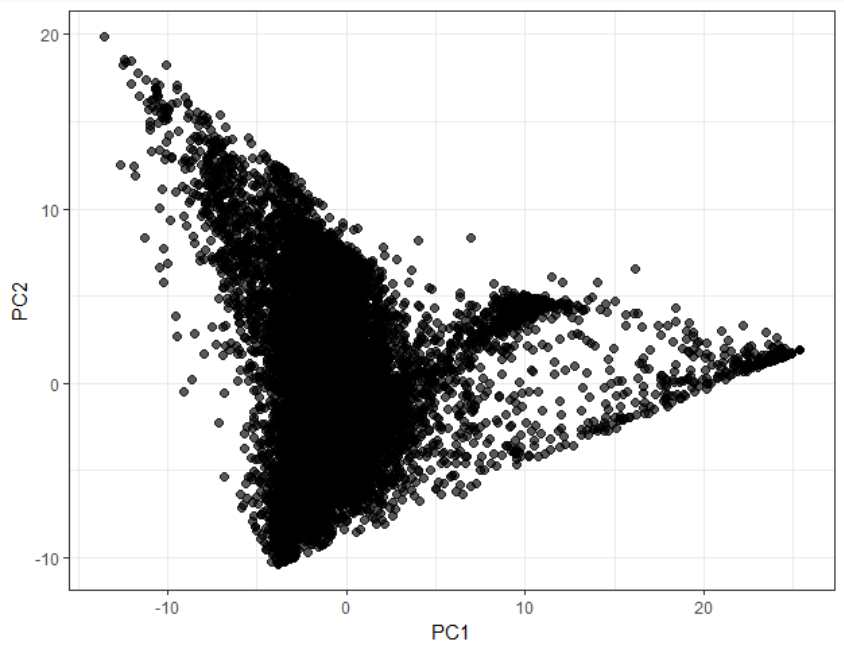


Figure 3: Slovakia - factories

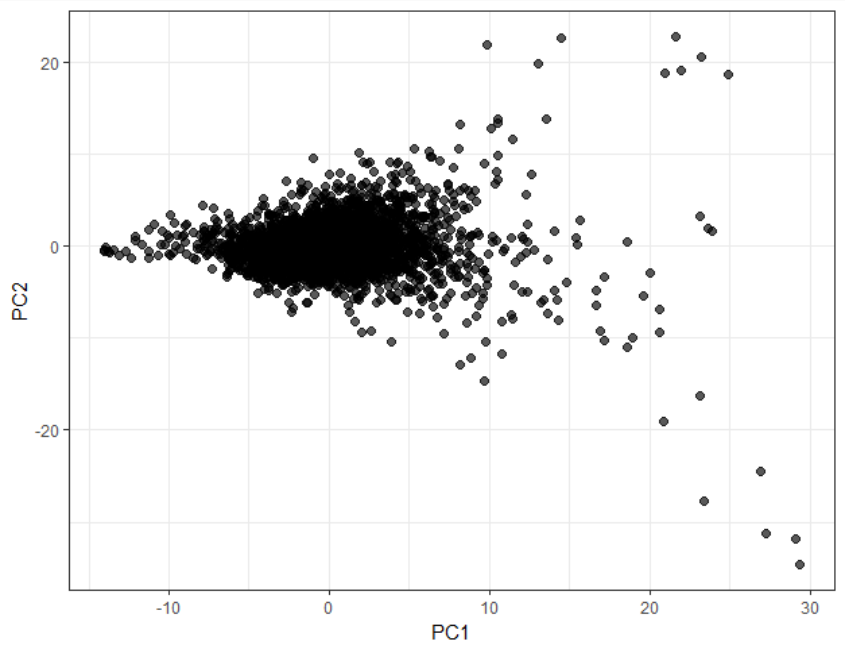


Figure 4: Ireland - residential

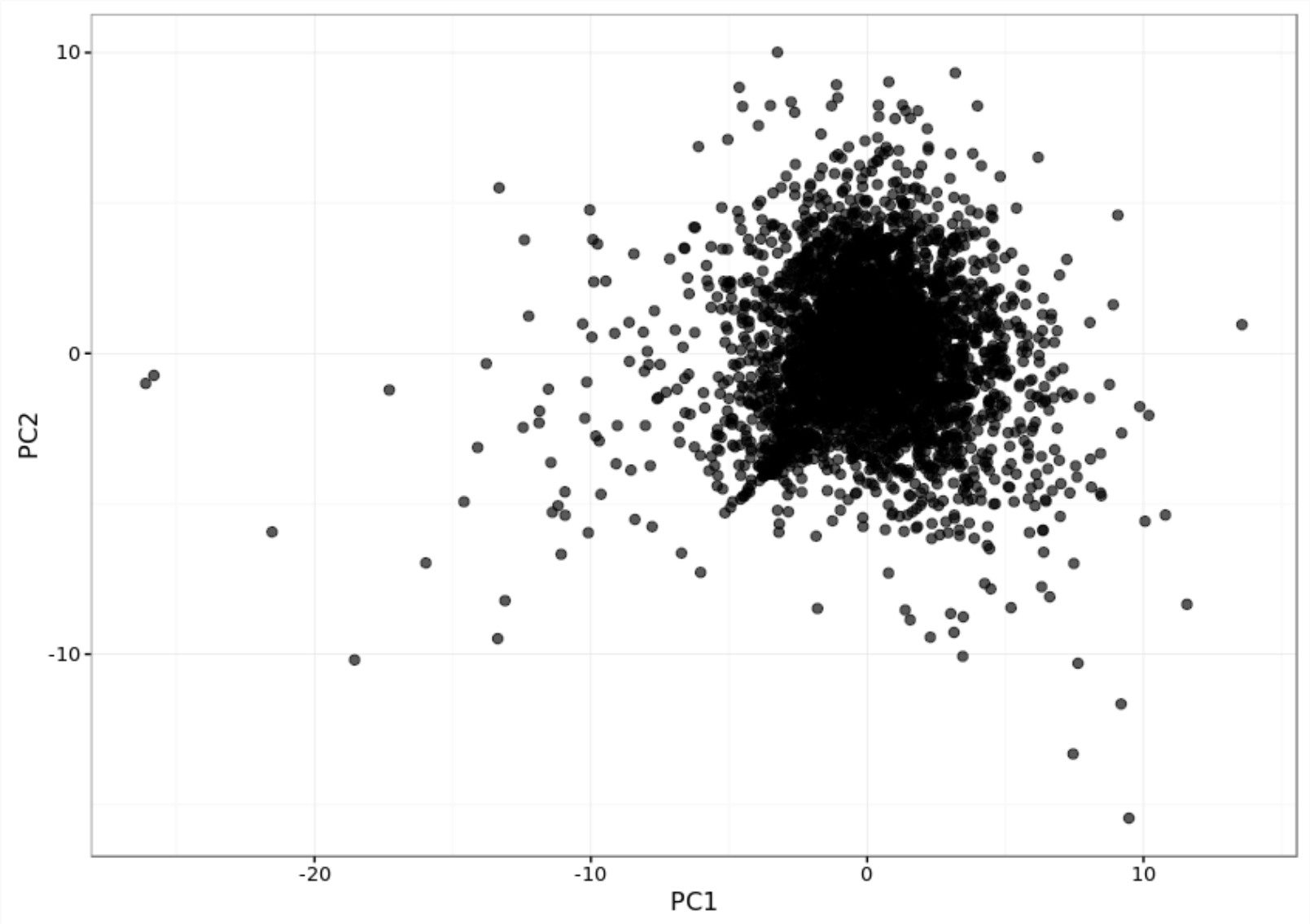
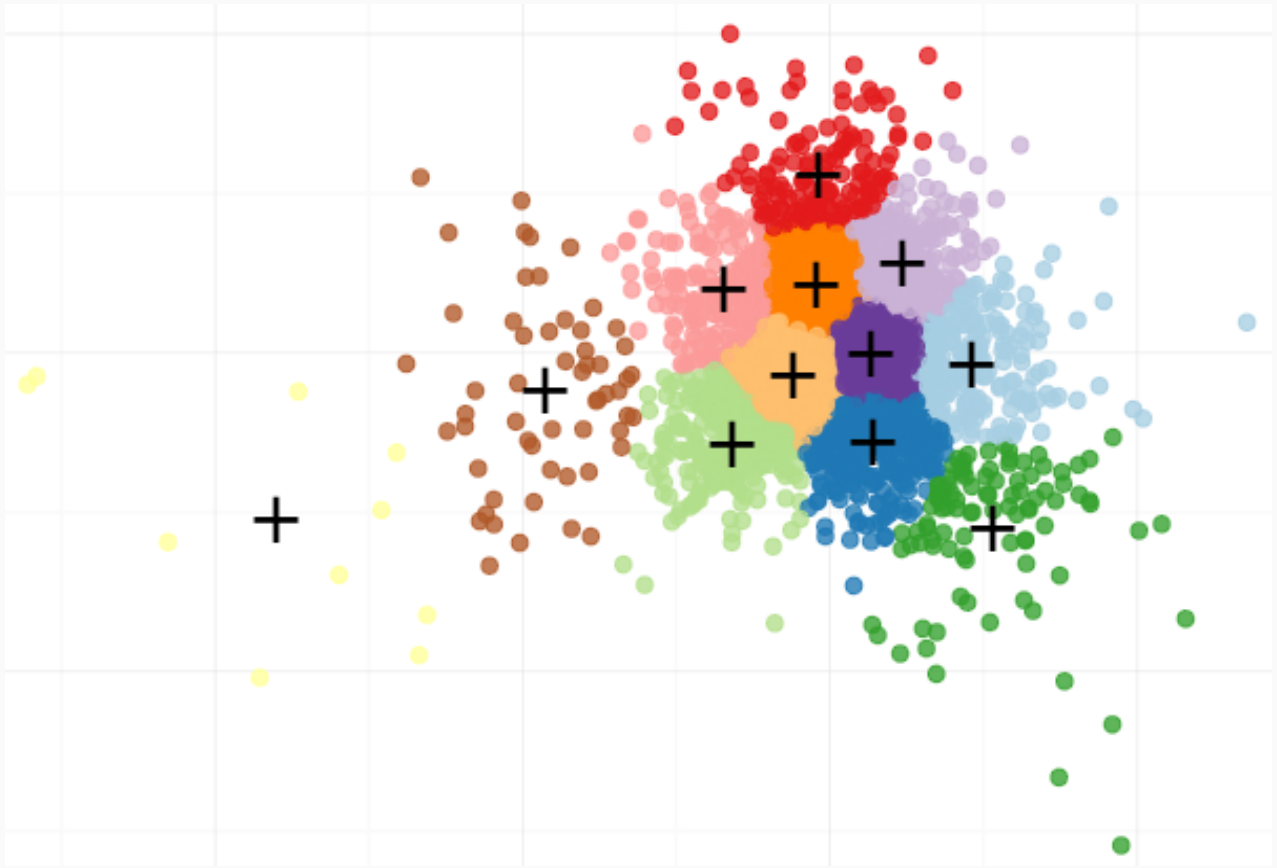


Figure 5: Ireland - median daily profile

K-means

Euclidean distance. Centroids.

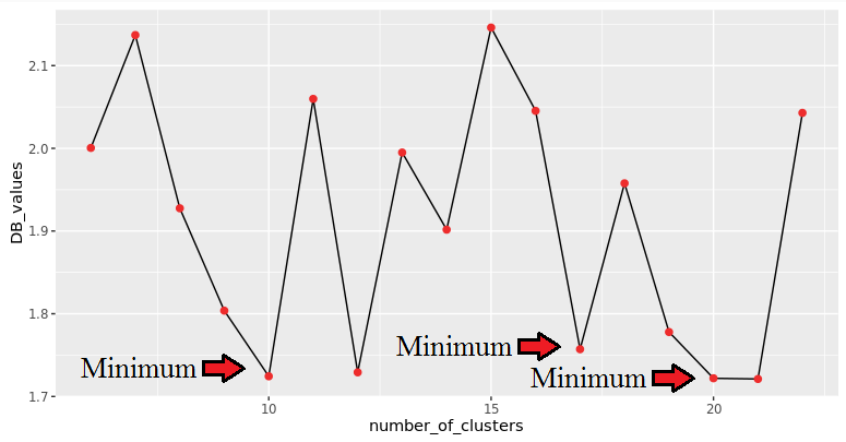


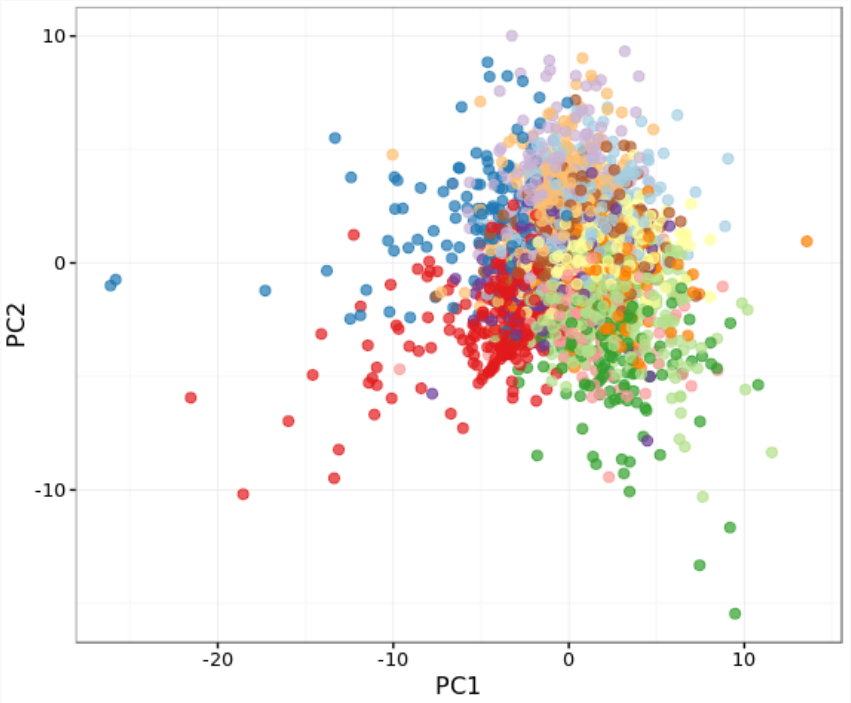
K-means++ - careful seeding of centroids.

K-medoids, DBSCAN...

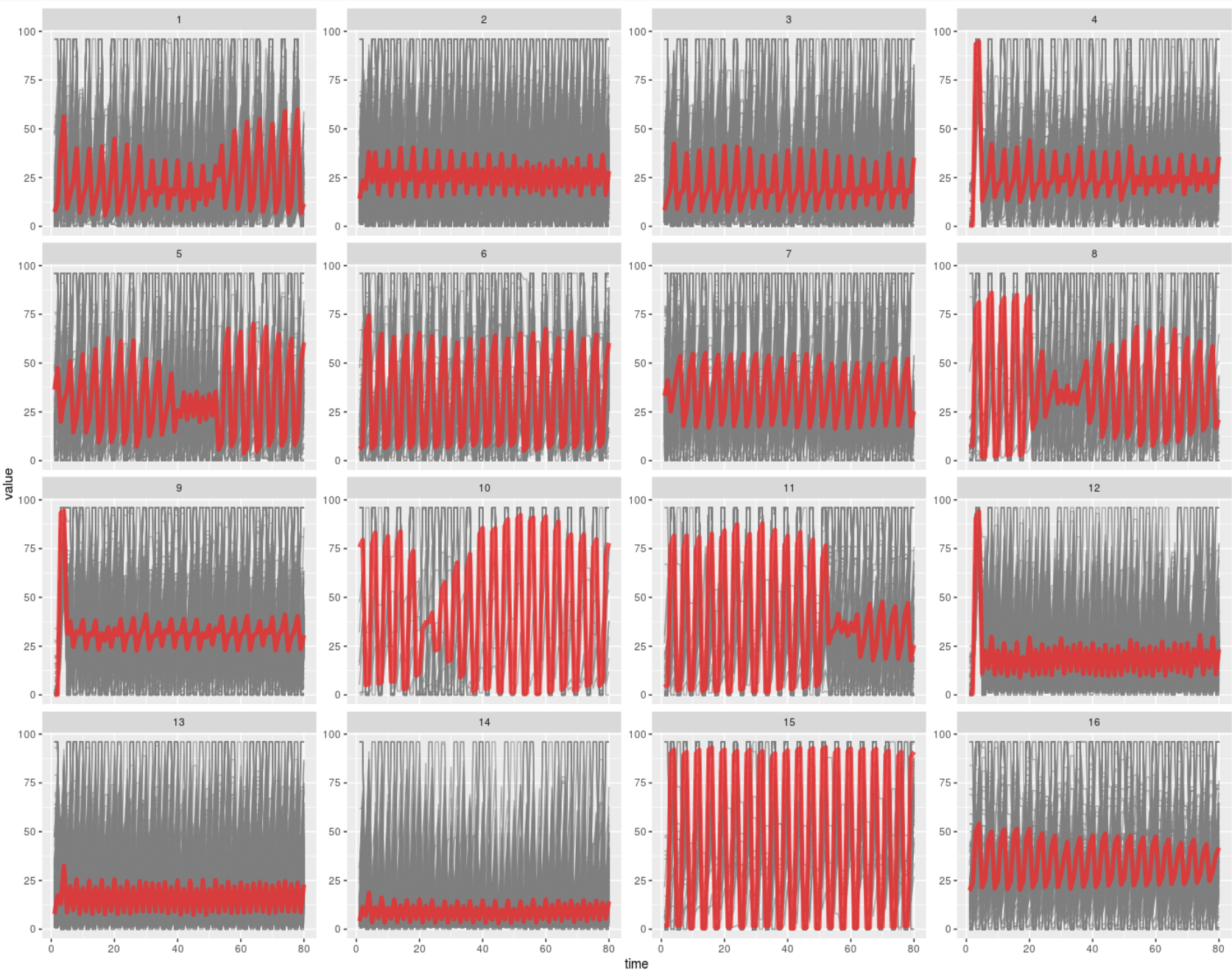
Finding optimal K

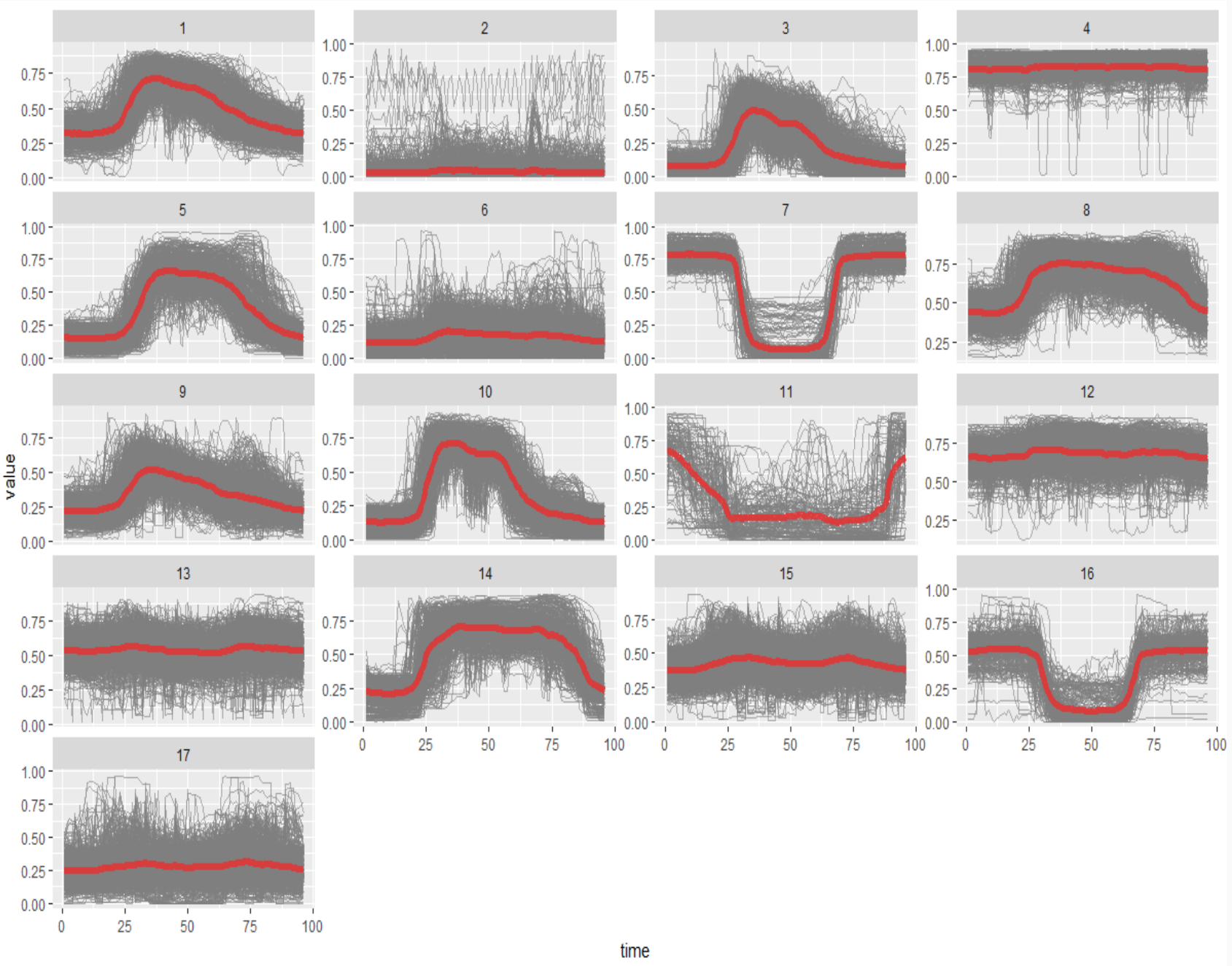
Manually or automatically set number of clusters. By intern validity measure. Davies-Bouldin index...

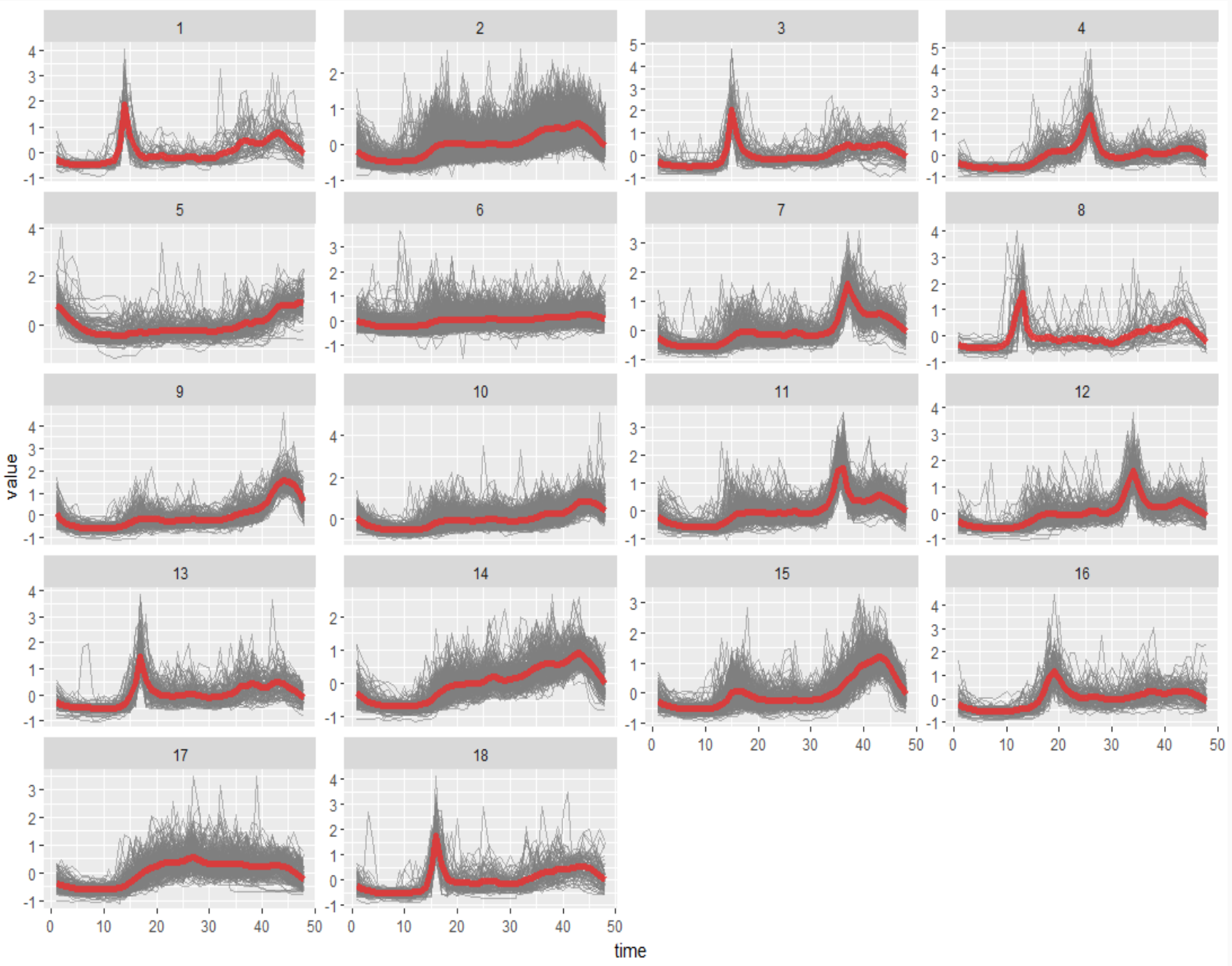


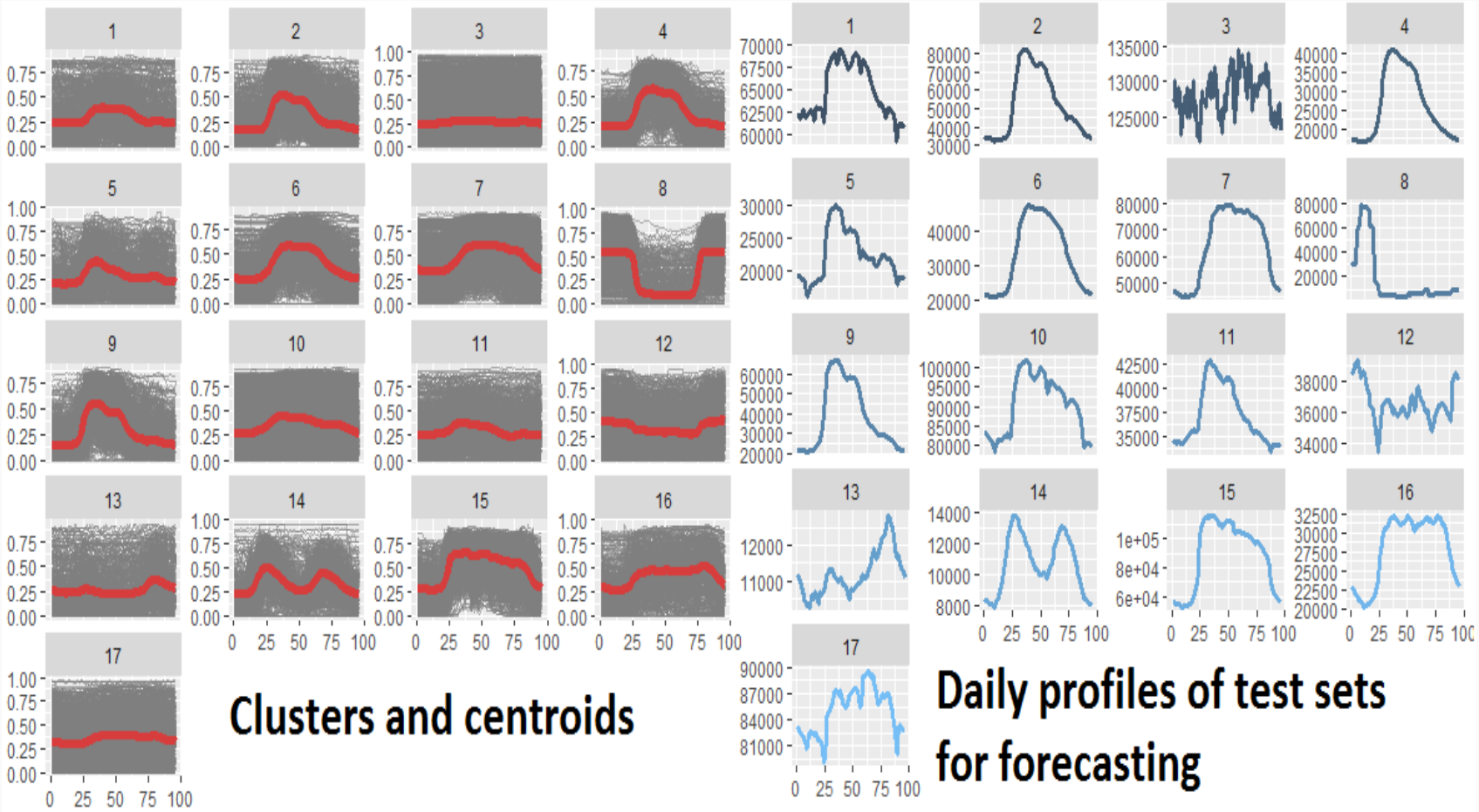


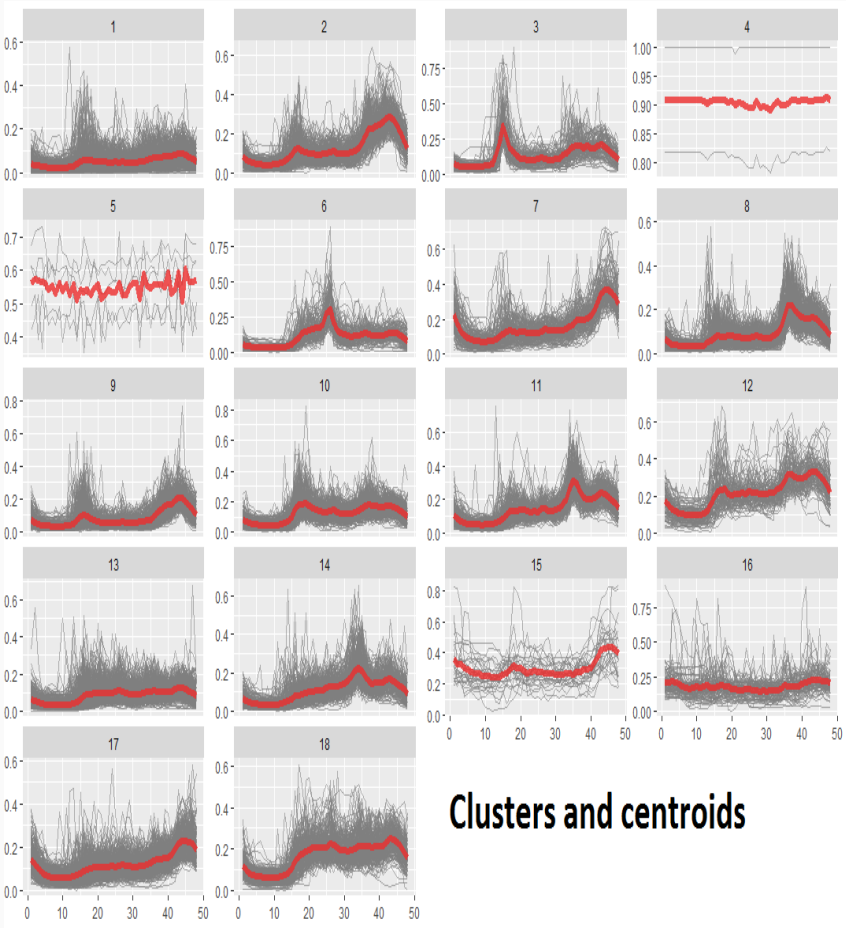
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
123	90	139	477	11	468	263	184	94	136	765	224	195	55	81	82	160	90	2


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
value
time
0
25
50
75
100
0
20
40
60
80

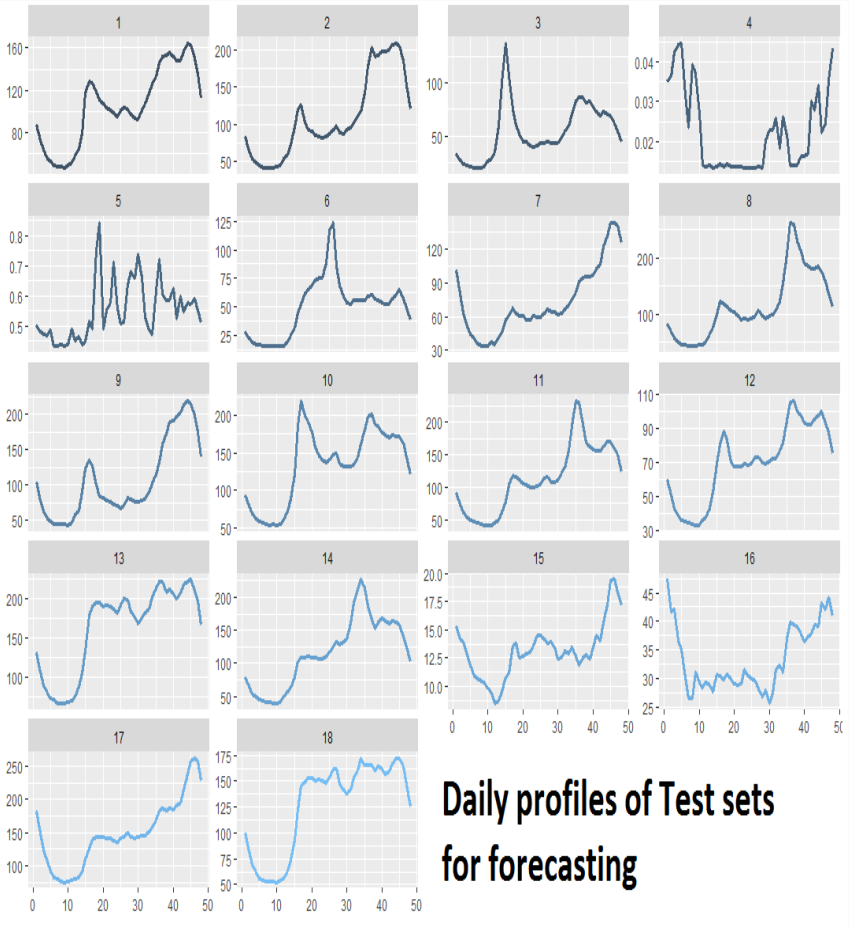

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
value
time
0.00
0.25
0.50
0.75
1.00
0
25
50
75
100


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
value
time


Clusters and centroids
Daily profiles of test sets for forecasting



Clusters and centroids



Daily profiles of Test sets for forecasting

Results

Static data set

One clustering → prediction with sliding window.
Irish. GAM, RLM, PAA - 5.23% → okolo 4.22% MAPE.
Slovak. Clip10, LM, DWT - 4.15% → okolo 3.97% MAPE.
STL + EXP, STL + ARIMA, SVR, Bagging. Comparison of 17 representations.

Batch processing

Batch clustering → prediction with sliding window (14 days).
Irish. RLM, Median, GAM. Comparison of 10 prediction methods.
Significant improvement in DSHW, STL + ARIMA, STL + ETS, SVR, RandomForest, Bagging and MLP.
Best result: Bagging with GAM 3.68% against 3.82% MAPE.

R <- Tips and Tricks

Packages

List of all packages by name + short description.

(`https://cran.r-project.org/web/packages/available_packages_by_name.html`)

Available CRAN Packages By Name

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

A3	Accurate, Adaptable, and Accessible Error Metrics for Predictive Models
abbyyR	Access to Abbyy Optical Character Recognition (OCR) API
abc	Tools for Approximate Bayesian Computation (ABC)
ABCanalysis	Computed ABC Analysis
abc.data	Data Only: Tools for Approximate Bayesian Computation (ABC)
abcdeFBA	ABCDE_FBA: A-Biologist-Can-Do-Everything of Flux Balance Analysis
ABCoptim	Implementation of Artificial Bee Colony (ABC) Optimization
ABCp2	Approximate Bayesian Computational Model for Estimating P2
ABC.RAP	Array Based CpG Region Analysis Pipeline
abcrf	Approximate Bayesian Computation via Random Forests
abctools	Tools for ABC Analyses
abd	The Analysis of Biological Data

Task Views

List of specific tasks (domains) - description of available packages and functions. (**`https://cran.r-project.org/web/views/`**)

CRAN Task View: Machine Learning & Statistical Learning

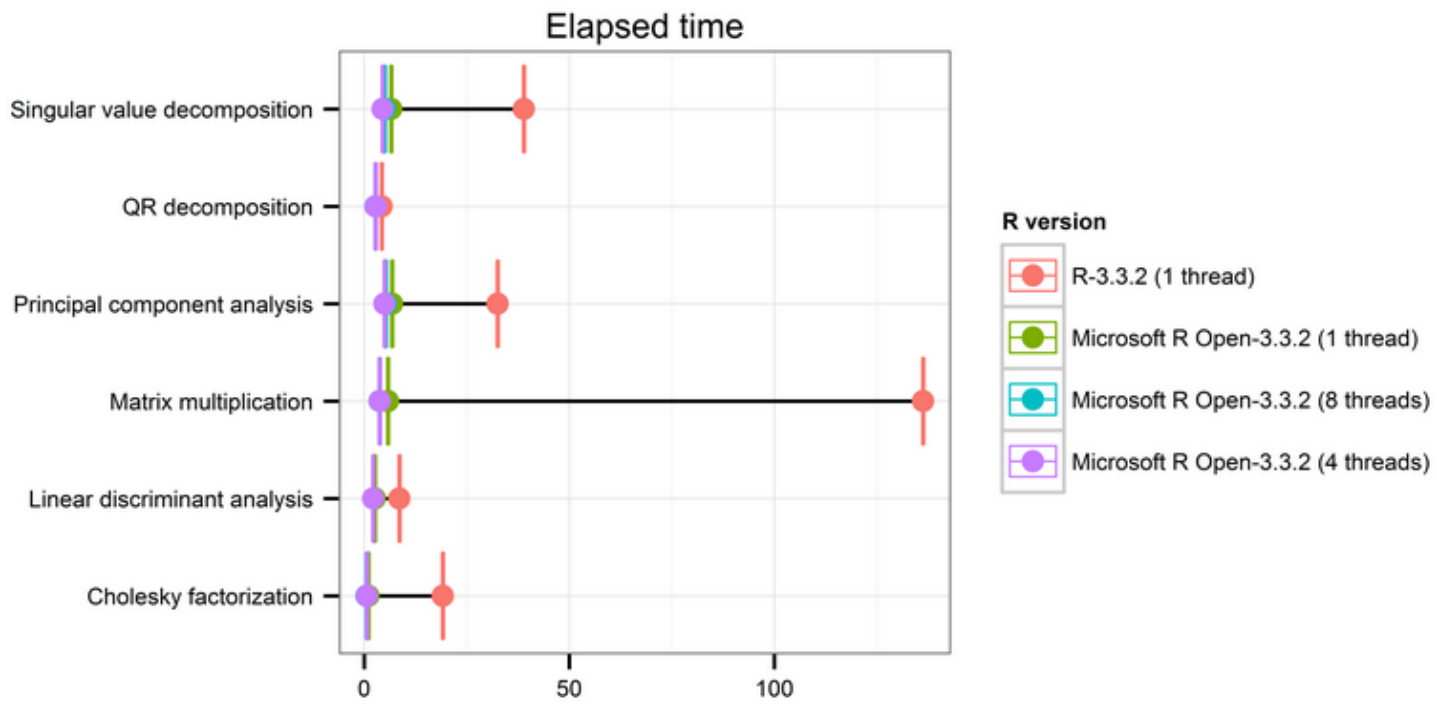
Maintainer: Torsten Hothorn
Contact: Torsten.Hothorn at R-project.org
Version: 2017-03-17
URL: https://CRAN.R-project.org/view=MachineLearning

Several add-on packages implement ideas and methods developed at the borderline between computer science and statistics - this field of research is usually referred to as machine learning. The packages can be roughly structured into the following topics:

- *Neural Networks* : Single-hidden-layer neural network are implemented in package nnet (shipped with base R). Package RSNNS offers an interface to the Stuttgart Neural Network Simulator (SNNS). An interface to the FCNN library allows user-extensible artificial neural networks in package FCNN4R. rnn implements recurrent neural networks.
- *Recursive Partitioning* : Tree-structured models for regression, classification and survival analysis, following the ideas in the CART book, are implemented in rpart (shipped with base R) and tree. Package rpart is recommended for computing CART-like trees. A rich toolbox of partitioning algorithms is available in Weka , package RWeka provides an interface to this implementation, including the J4.8-variant of C4.5 and M5. The Cubist package fits rule-based models (similar to trees) with linear regression models in the terminal leaves, instance-based corrections and boosting. The C50 package can fit C5.0 classification trees, rule-based models, and boosted versions of these.
  Two recursive partitioning algorithms with unbiased variable selection and statistical stopping criterion are implemented in package party. Function `ctree()` is based on non-parametrical conditional inference procedures for testing independence between

Matrix Calculus

MRO [8]. The Intel Math Kernel Library (MKL).



$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T y$$

[8] https://mran.microsoft.com/documents/rro/multithread/

Fast! SVD and PCA

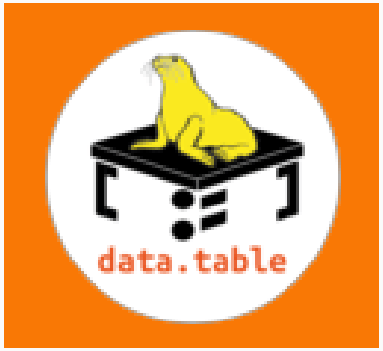
Alternative to **prcomp**:

```
library(rARPACK)

pc <- function(x, k) {
 # First center data
 xc <- scale(x, center = TRUE, scale = TRUE)
 # Partial SVD
 decomp <- svds(xc, k, nu = 0, nv = k)
 return(xc %*% decomp$v)
}

pc_mat <- pc(your_matrix, 2)
```

data.table



Nice syntax[9]:

`DT[i, j, by]`

R	i	j	by
SQL	where	select or update	group by

[9] `https://github.com/Rdatatable/data.table/wiki/Getting-started`

Advantages

- Memory efficient
- Fast
- Less code and functions

Reading large amount of .csv:

```
files <- list.files(pattern = "*.csv")
DT <- rbindlist(lapply(files, function(x)
        cbind(fread(x), gsub(".csv", "", x))))
```

data.table

Easy subsetting

`data.frame` style:

```
DF[DF$col1 > 4 & DF$col2 == "R", "col3"]
```

`data.table` style:

```
DT[col1 > 4 & col2 == "R", .(col3)]
```

Expressions

`.N, .SD`

```
DT[, .N, by = .(INDUSTRY, SUB_INDUSTRY)]

DT[, lapply(.SD, sumNA)]
```

Changing DT by reference[10]

:=, set(), setkey(), setnames()

```
DF$R <- rep("love", Inf)
DT[, R := rep("love", Inf)]

for(i in from:to)
      set(DT, row, column, new value)
```

[10] shallow vs deep copy

Vectorization and apply

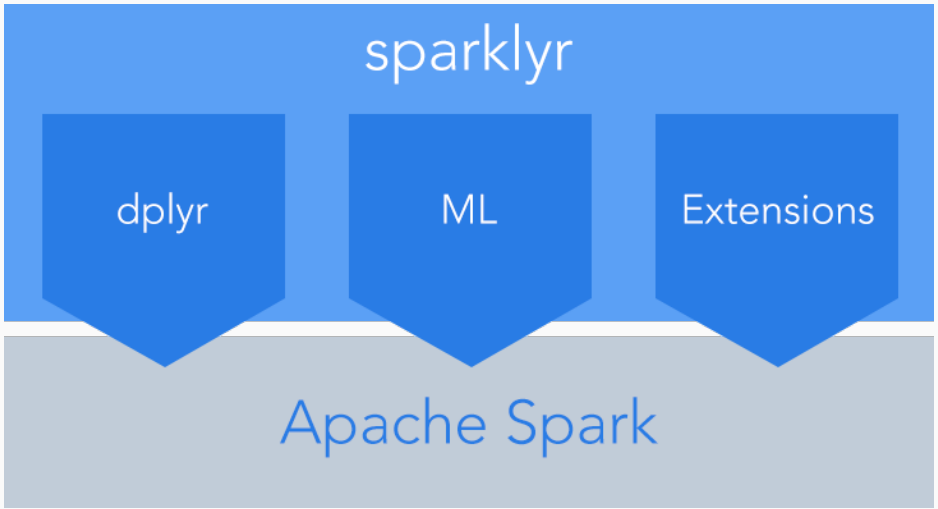
```
library(parallel)


lmCoef <- function(X, Y) {
   beta <- solve(t(X) %*% X) %*% t(X) %*% as.vector(Y)
    as.vector(beta)
}

cl <- makeCluster(detectCores())
clusterExport(cl, varlist = c("lmCoef", "modelMatrix",
       "loadMatrix"), envir = environment())
lm_mat <- parSapply(cl, 1:nrow(loadMatrix), function(i)
       lmCoef(modelMatrix, as.vector(loadMatrix[i,])))
stopCluster(cl)
```

Spark + R

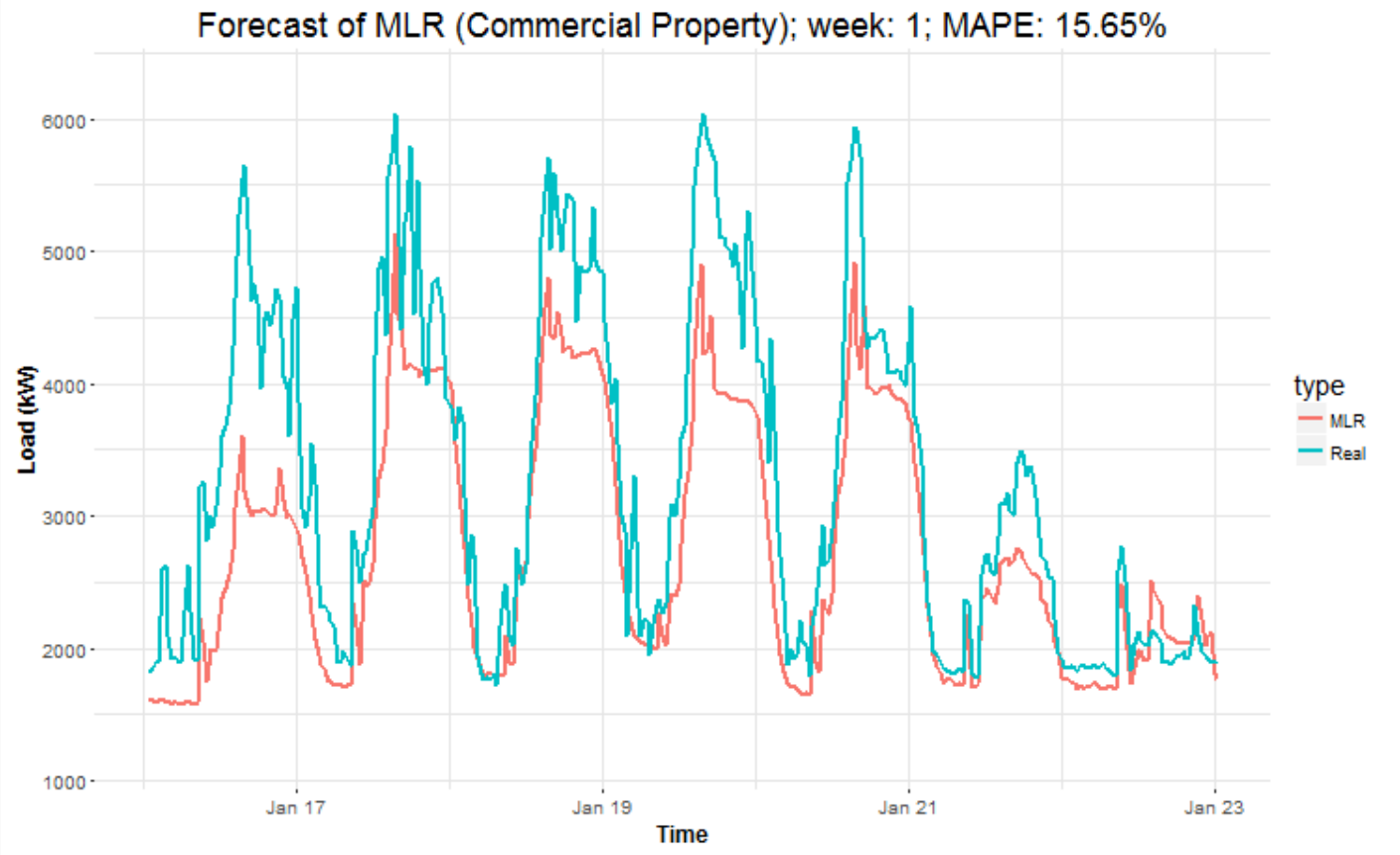
sparklyr - Perform SQL queries through the sparklyr dplyr interface. MLlib + extensions.



sparklyr **http://spark.rstudio.com/index.html**
SparkR **https://spark.apache.org/docs/2.0.1/sparkr.html**
Comparison **http://stackoverflow.com/questions/39494484/sparkr-vs-sparklyr**
HPC **https://cran.r-project.org/web/views/HighPerformanceComputing.html**

Graphics

ggplot2, googleVis, ggVis, dygraphs, plotly, animation, shiny...



Summary

- More accurate predictions are important.
- Clustering of electricity consumers have lot of applications.
- Prediction methods - different methods needs different approaches.
- Tips for more effective computing (programming) in R.

Blog: **https://petolau.github.io/**

Publications:
https://www.researchgate.net/profile/Peter_Laurinec

Contact: laurinec.peter@gmail.com