Time Series Representations for Better Data Mining

What can we do with time series data?

- Classification
- Clustering
- Anomaly (outlier) detection
- Forecasting

What are the problems with time series data?

- High-dimension
- Noise
- Concept-drift (trend-shift etc.)

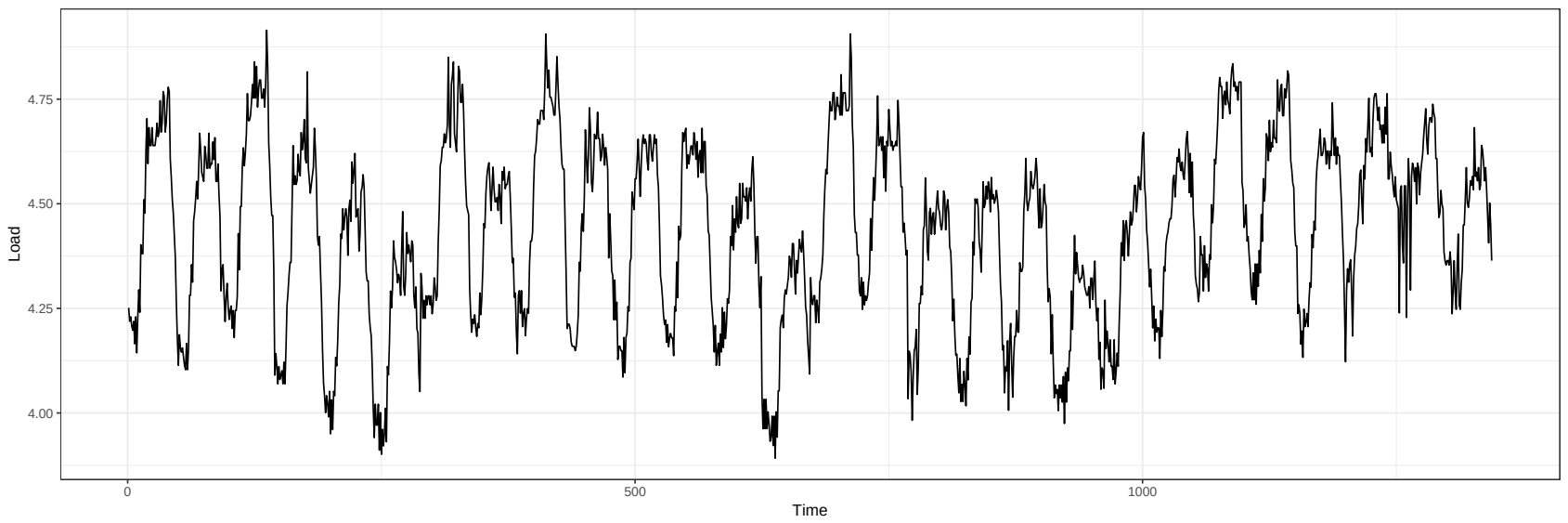
Time Series Representations

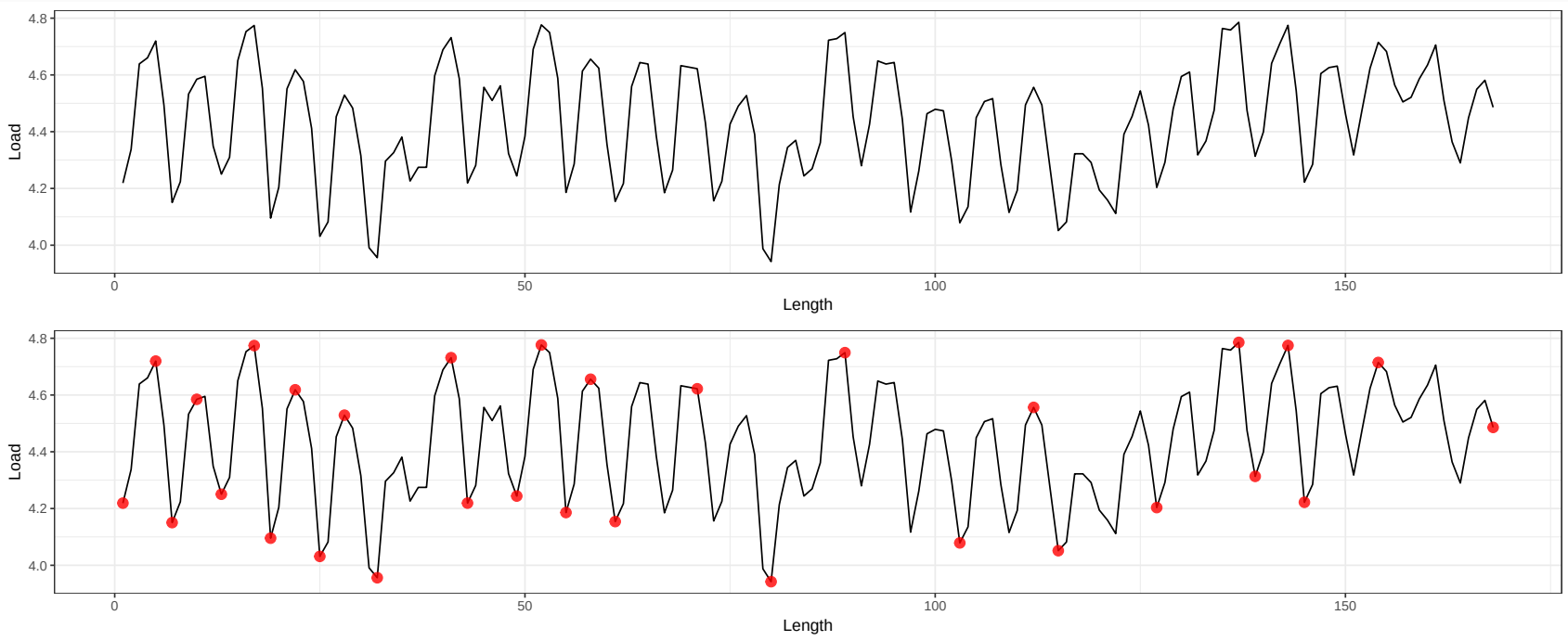
What can we do for solving these problems?

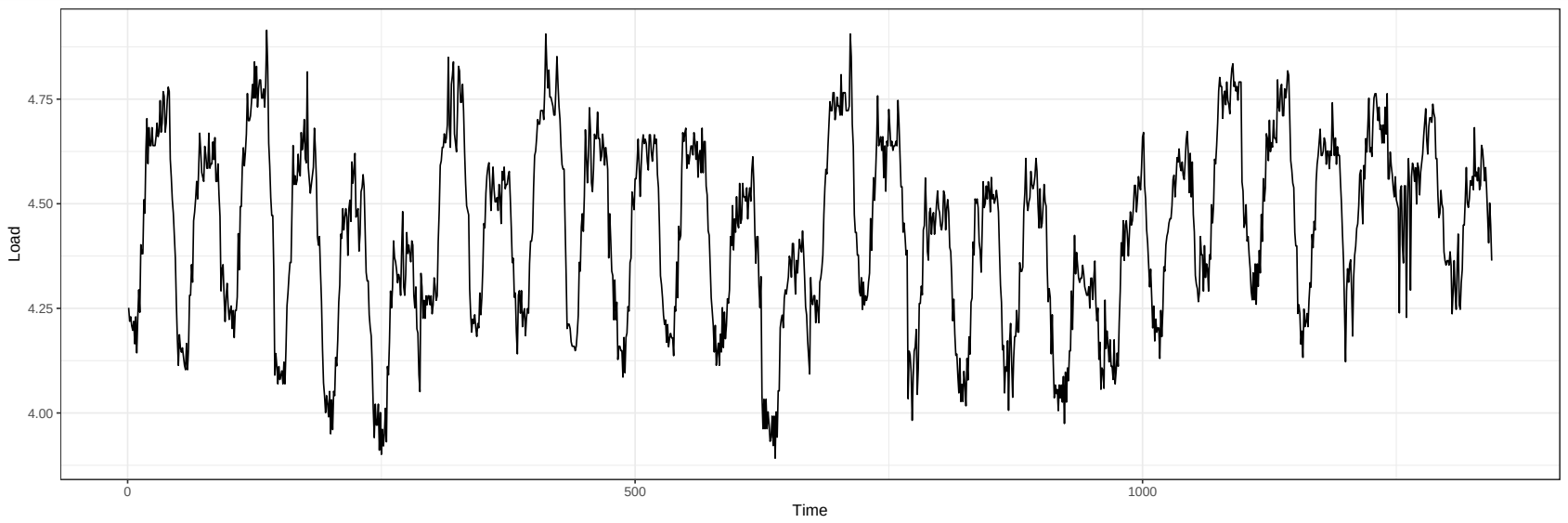
- **Use time series representations!**

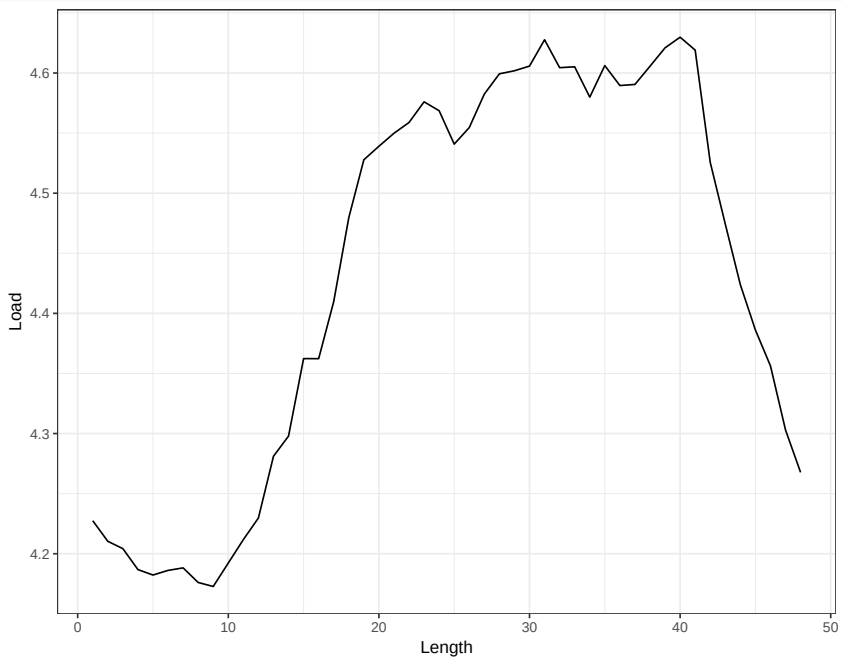
They are excellent to:

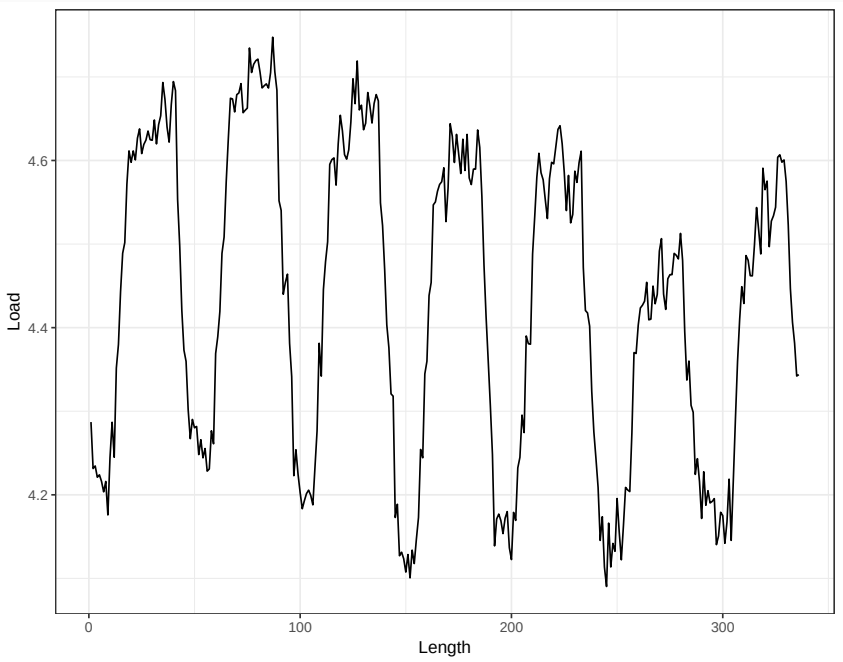
- Reduce memory load.
- Accelerate subsequent machine learning algorithms.
- Implicitly remove noise from the data.
- Emphasize the essential characteristics of the data.
- Help to find patterns in data (or motifs).


Load
4.75
4.50
4.25
4.00
0
500
1000
Time


Load
4.8
4.6
4.4
4.2
4.0
0
50
100
150
Length
Load
4.8
4.6
4.4
4.2
4.0
0
50
100
150
Length


Load
4.75
4.50
4.25
4.00
0
500
1000
Time


Load
4.6
4.5
4.4
4.3
4.2
0
10
20
30
40
50
Length


Load
4.6
4.4
4.2
0
100
200
300
Length

TSrepr - CRAN[1], GitHub[2]

- R package for time series representations computing
- Large amount of various methods are implemented
- Several useful support functions are also included
- Easy to extend and to use

```
data <- rnorm(1000)
repr_paa(data, func = median, q = 10)
```

[1] https://CRAN.R-project.org/package=TSrepr
[2] https://github.com/PetoLau/TSrepr/

All type of time series representations methods are implemented, so far these:

- PAA - Piecewise Aggregate Approximation (`repr_paa`)
- DWT - Discrete Wavelet Transform (`repr_dwt`)
- DFT - Discrete Fourier Transform (`repr_dft`)
- DCT - Discrete Cosine Transform (`repr_dct`)
- PIP - Perceptually Important Points (`repr_pip`)
- SAX - Symbolic Aggregate Approximation (`repr_sax`)
- PLA - Piecewise Linear Approximation (`repr_pla`)
- Mean seasonal profile (`repr_seas_profile`)
- Model-based seasonal representations based on linear model (`repr_lm`)
- FeaClip - Feature extraction from clipping representation (`repr_feaclip`)

Additional useful functions are implemented as:

- Windowing (`repr_windowing`)
- Matrix of representations (`repr_matrix`)
- Normalisation functions - z-score (`norm_z`), min-max (`norm_min_max`)

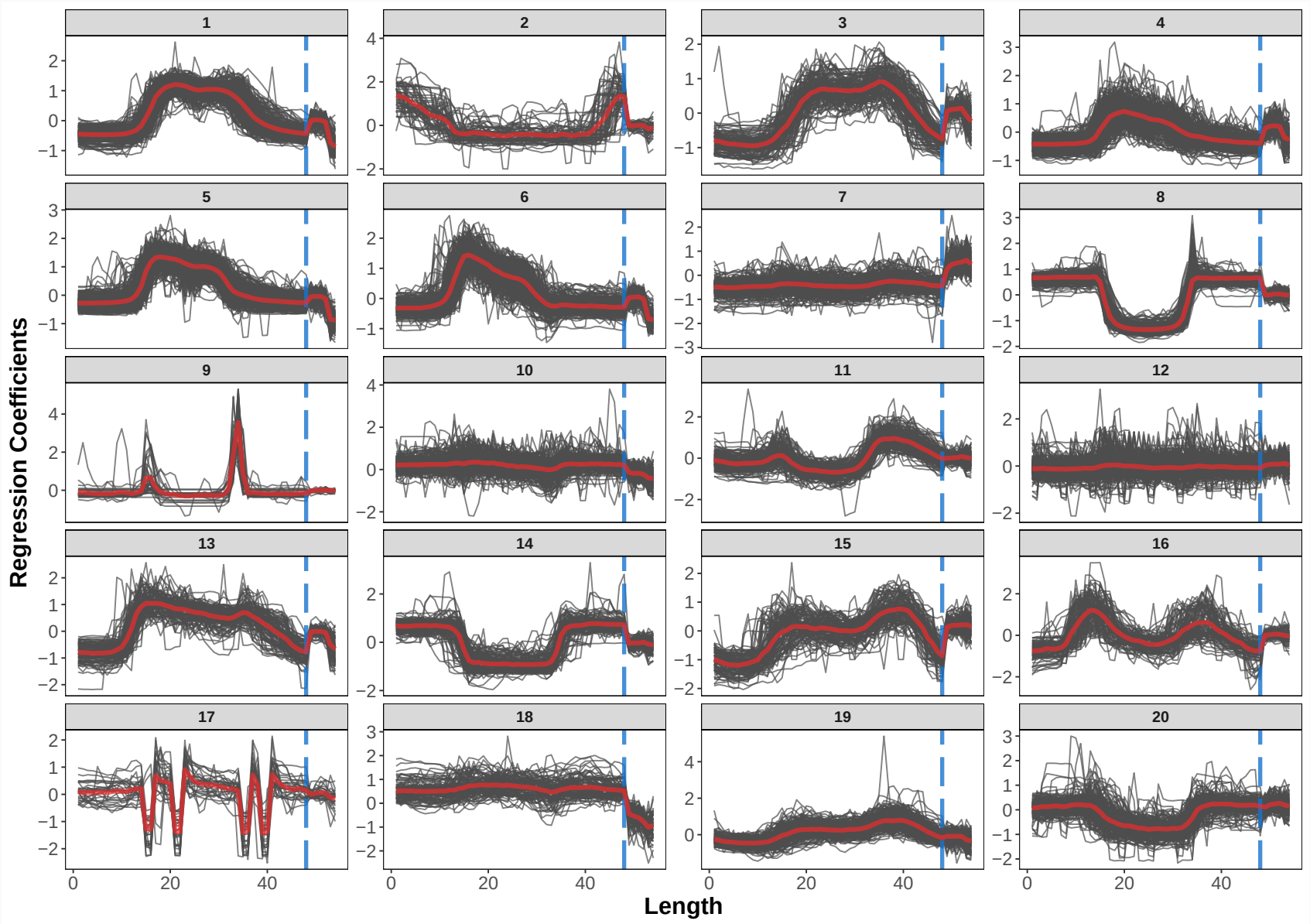
Usage of TSrepr

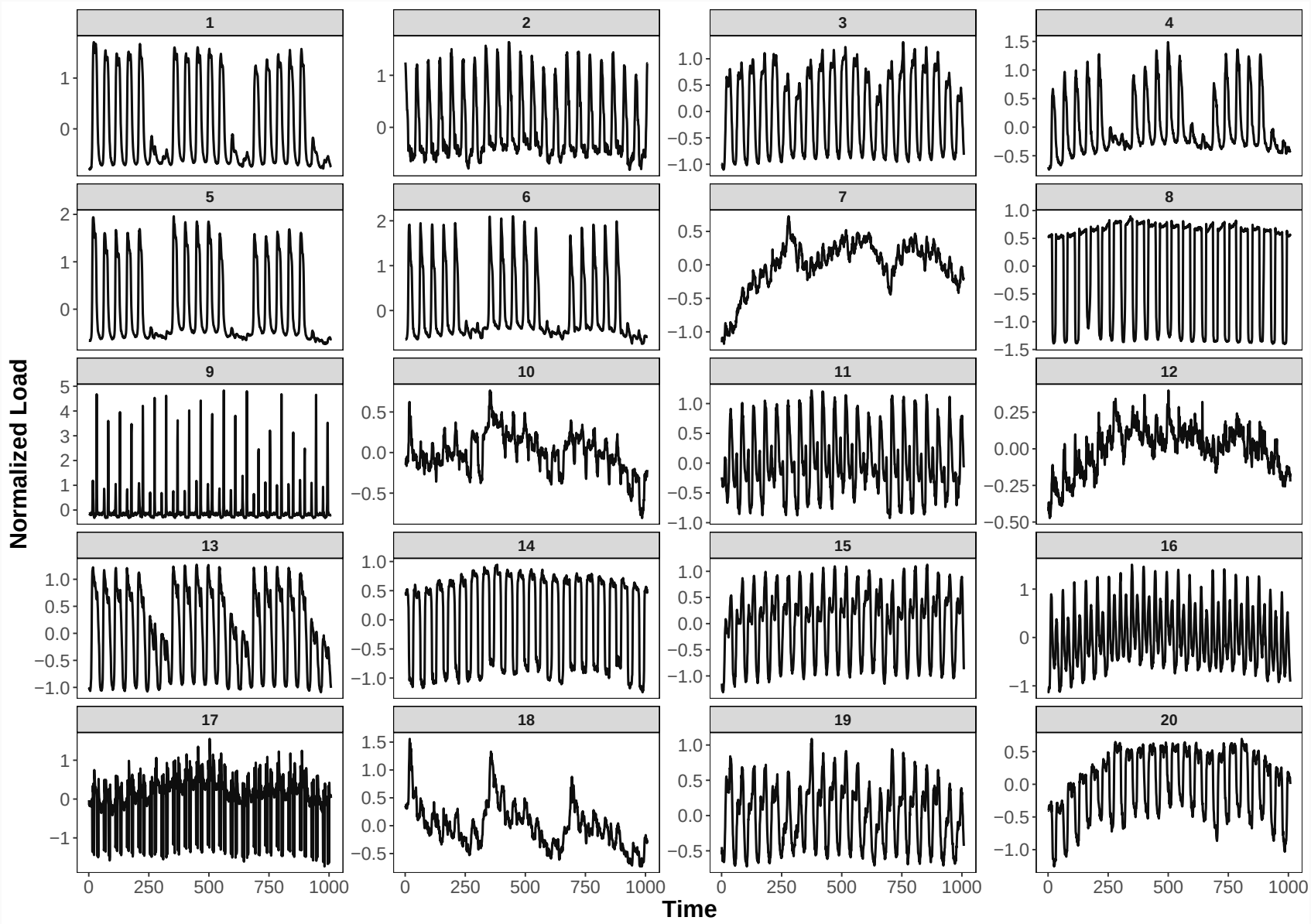
```
mat <- "some matrix with lot of time series"
```

```
mat_reprs <- repr_matrix(mat, func = repr_lm,
       args = list(method = "rlm", freq = c(48, 48*7)),
       normalise = TRUE, func_norm = norm_z)
```

```
mat_reprs <- repr_matrix(mat, func = repr_feaclip,
       windowing = TRUE, win_size = 48)
```

```
clustering <- kmeans(mat_reprs, 20)
```


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
Regression Coefficients
Length


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
Normalized Load
Time
0
250
500
750
1000

Simple extensibility of TSrepr

Example #1:

```
library(moments)
```

```
data_ts_skew <- repr_paa(data, q = 48, func = skewness)
```

Example #2:

```
repr_fea_extract <- function(x)
      c(mean(x), median(x), max(x), min(x), sd(x))
```

```
data_fea <- repr_windowing(data,
      win_size = 100, func = repr_fea_extract)
```

Conclusions

Time Series Representations:

- They are our fiends in clustering, forecasting, classification etc.
- Implemented in TSrepr

Questions: Peter Laurinec `tsreprpackage@gmail.com`
Code: **`https://github.com/PetoLau/TSrepr/`**
More research: **`https://petolau.github.io/research`**
Blog: **`https://petolau.github.io`**

And of course: `install.packages("TSrepr")`