

# Feature-based time series analysis

Rob J Hyndman

21 June 2018

# Outline

1 M3 forecasting competition

2 Yahoo email servers

3 Irish smart metres

4 Packages

# M3 competition



ELSEVIER

International Journal of Forecasting 16 (2000) 451–476

international journal  
of forecasting

[www.elsevier.com/locate/ijforecast](http://www.elsevier.com/locate/ijforecast)

## The M3-Competition: results, conclusions and implications

Spyros Makridakis, Michèle Hibon\*

*INSEAD, Boulevard de Constance, 77305 Fontainebleau, France*

### Abstract

This paper describes the M3-Competition, the latest of the M-Competitions. It explains the reasons for conducting the competition and summarizes its results and conclusions. In addition, the paper compares such results/conclusions with those of the previous two M-Competitions as well as with those of other major empirical studies. Finally, the implications of these results and conclusions are considered, their consequences for both the theory and practice of forecasting are explored and directions for future research are contemplated. © 2000 Elsevier Science B.V. All rights reserved.

**Keywords:** Comparative methods — time series: univariate; Forecasting competitions; M-Competition; Forecasting methods, Forecasting accuracy

# M3 competition



ELSEVIER

International Journal of Forecasting 16 (2000) 451–476

international  
of forecasting

[www.elsevier.com/locate/ijforecast](http://www.elsevier.com/locate/ijforecast)



## etition: results, conclusions

Spyros Makridakis, Michèle Hibon

EAD, Boulevard de Constance, 77305 Fontainebleau

Abstr

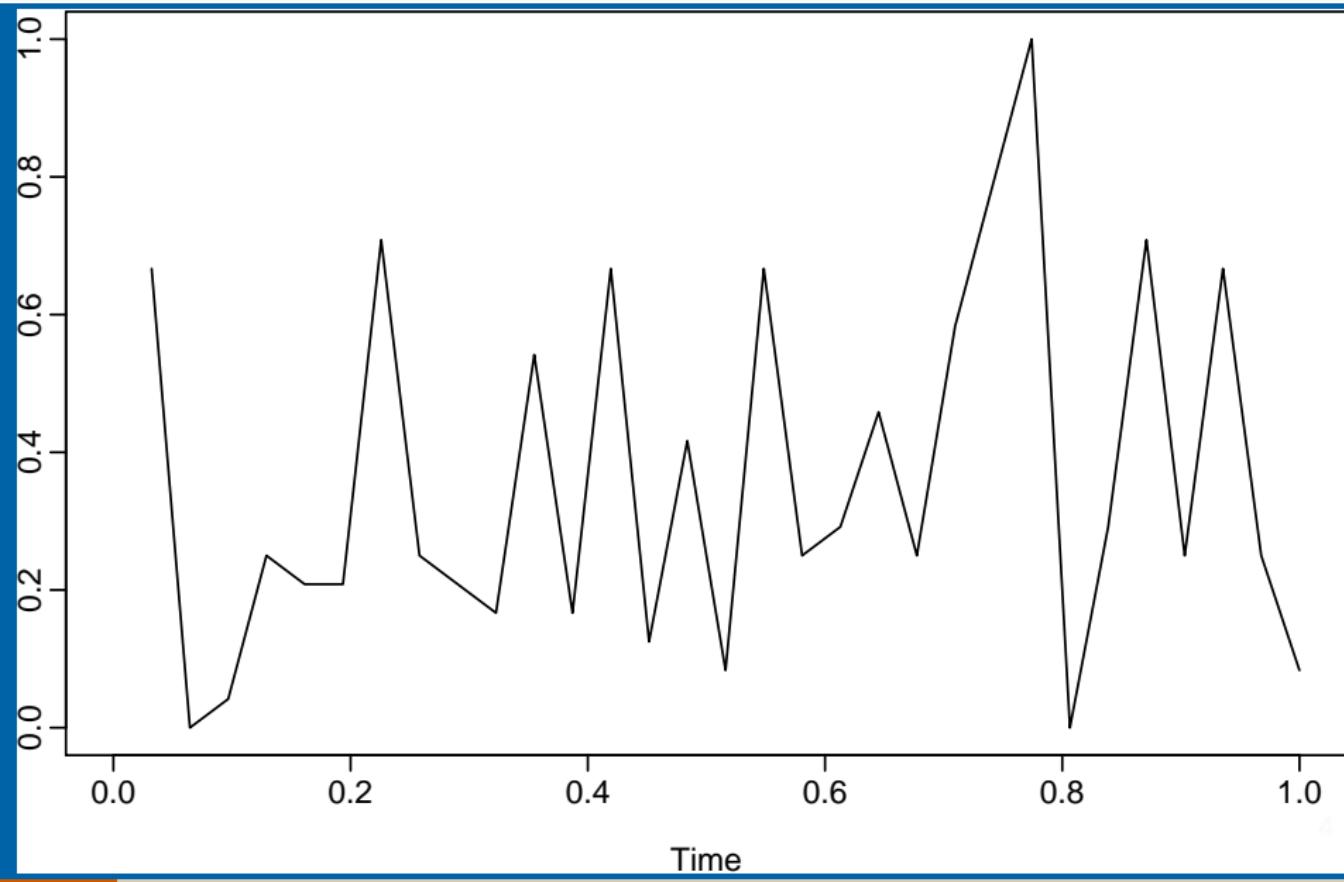


ons

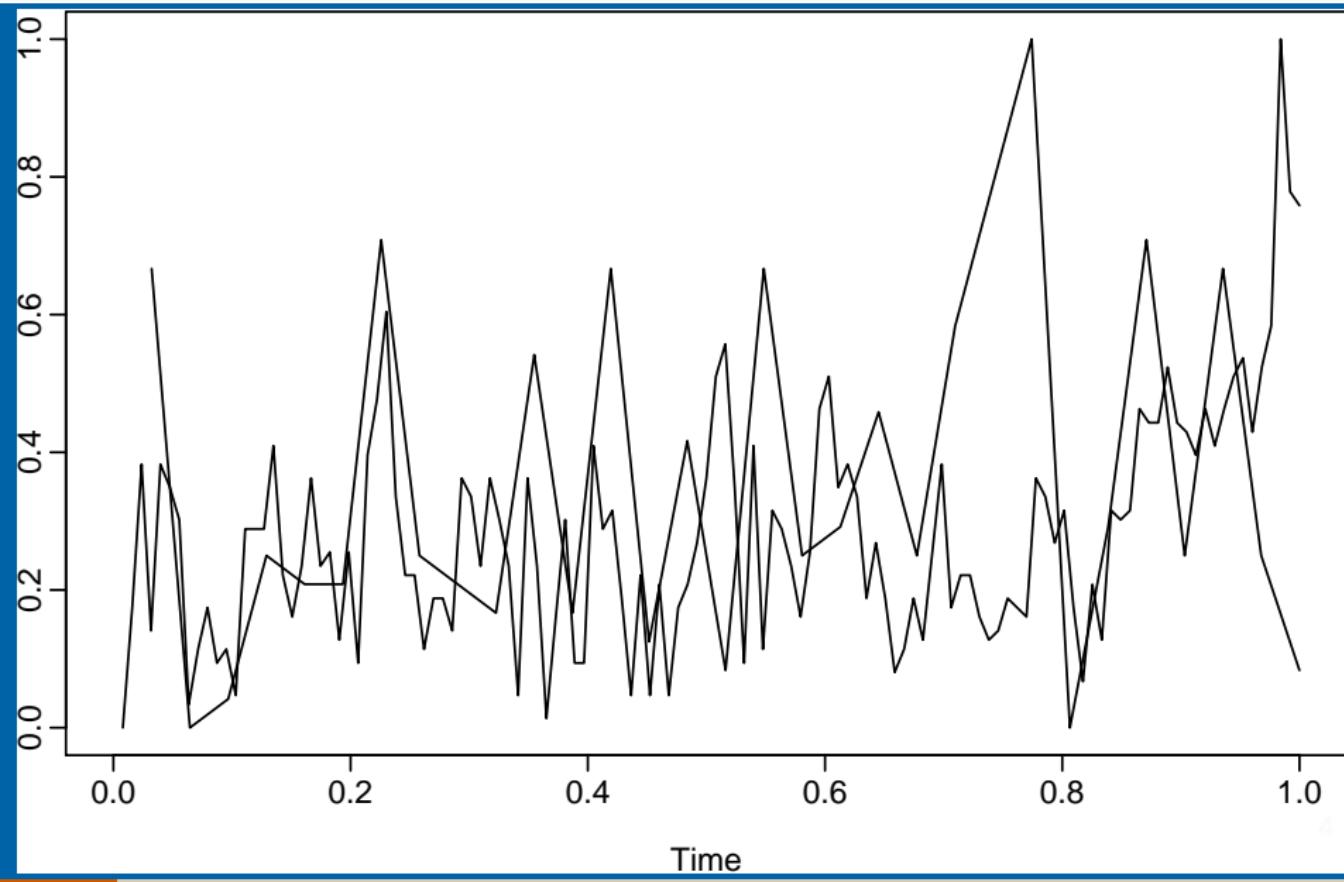
This paper describes the M3-Competition, the latest of the M-Competitions. It explains the reasons for conducting the competition and summarizes its results and conclusions. In addition, the paper compares such results/conclusions with those of the previous two M-Competitions as well as with those of other major empirical studies. Finally, the implications of these results and conclusions are considered, their consequences for both the theory and practice of forecasting are explored and directions for future research are contemplated. © 2000 Elsevier Science B.V. All rights reserved.

**Keywords:** Comparative methods — time series: univariate; Forecasting competitions; M-Competition; Forecasting methods, Forecasting accuracy

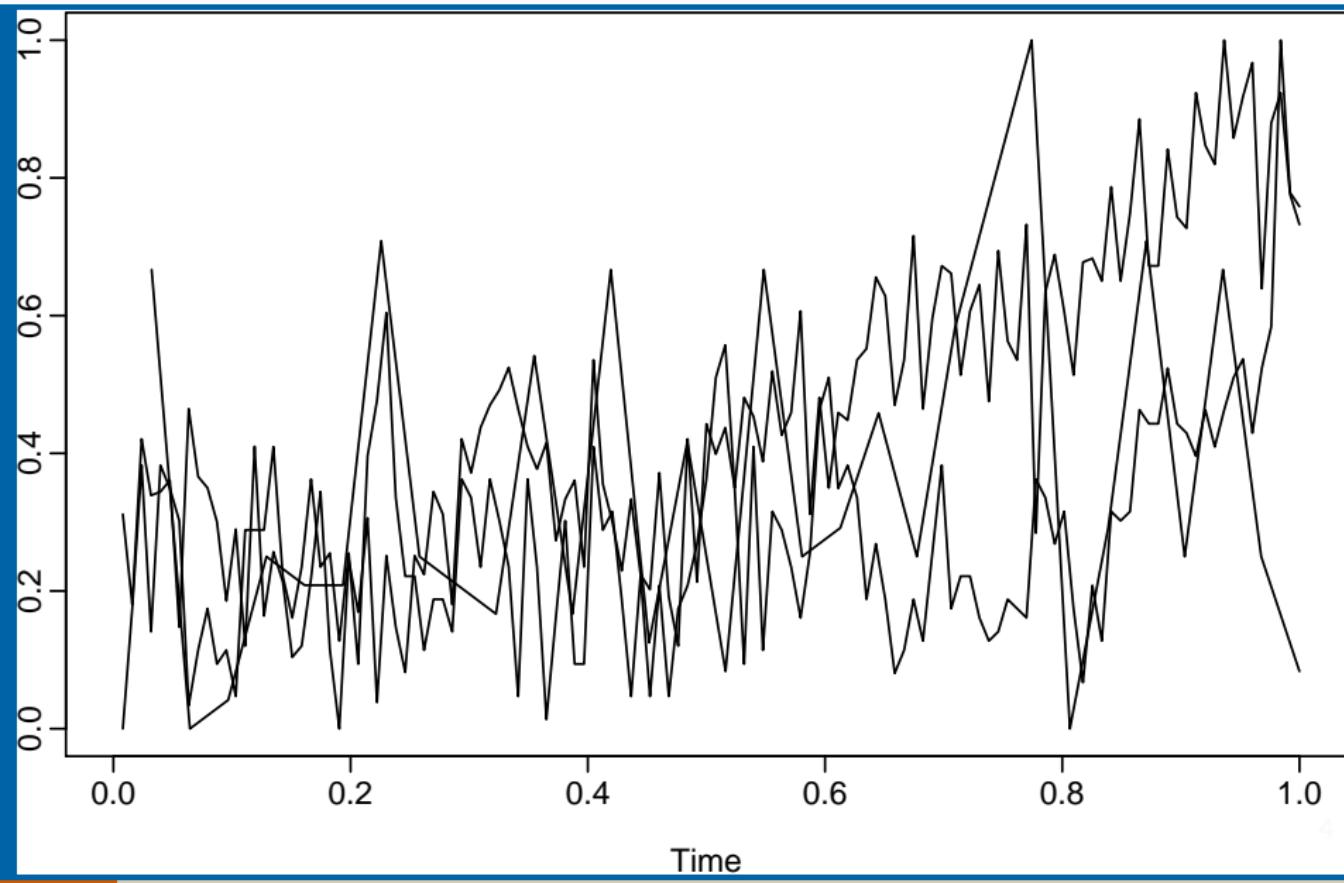
# How to plot lots of time series?



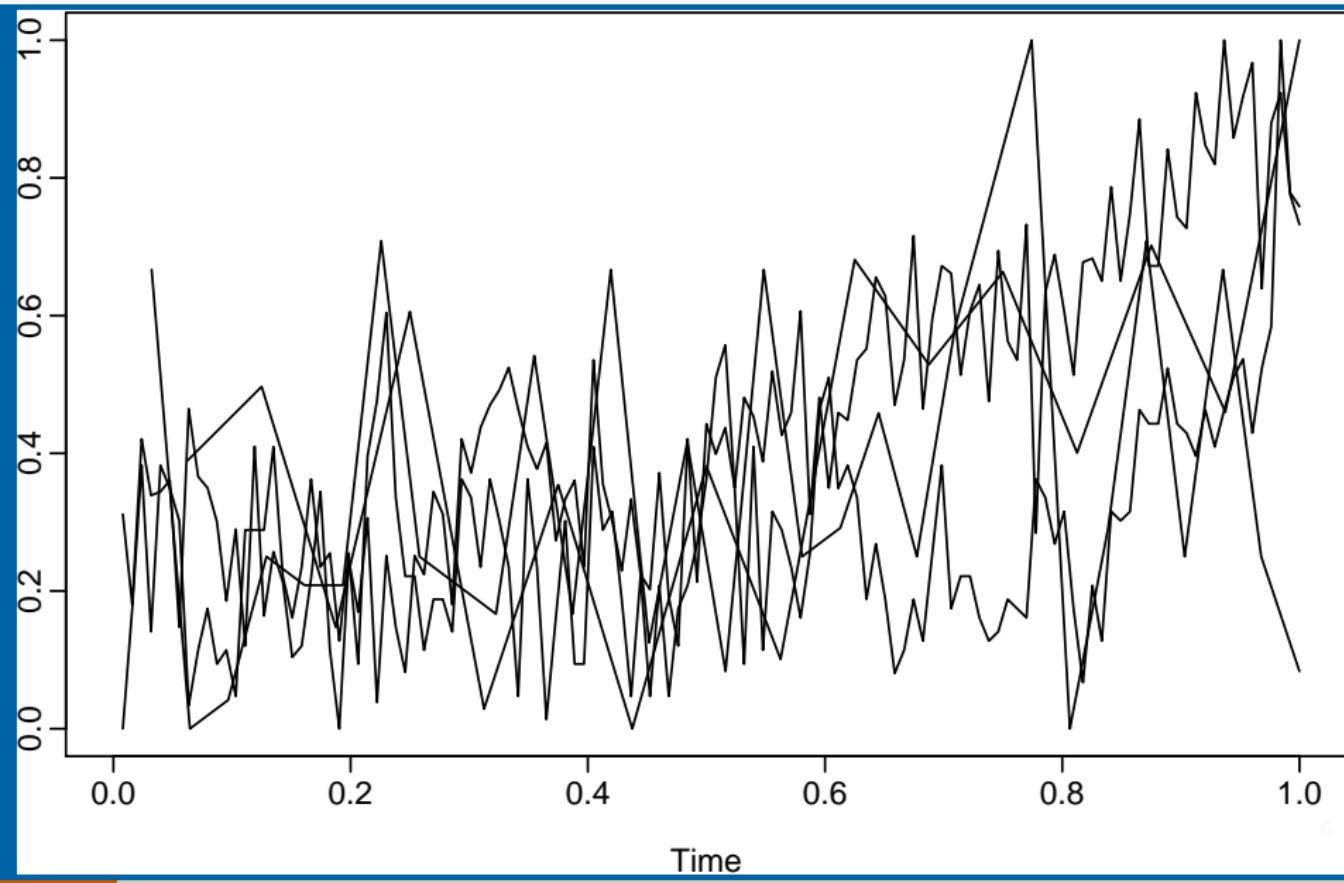
# How to plot lots of time series?



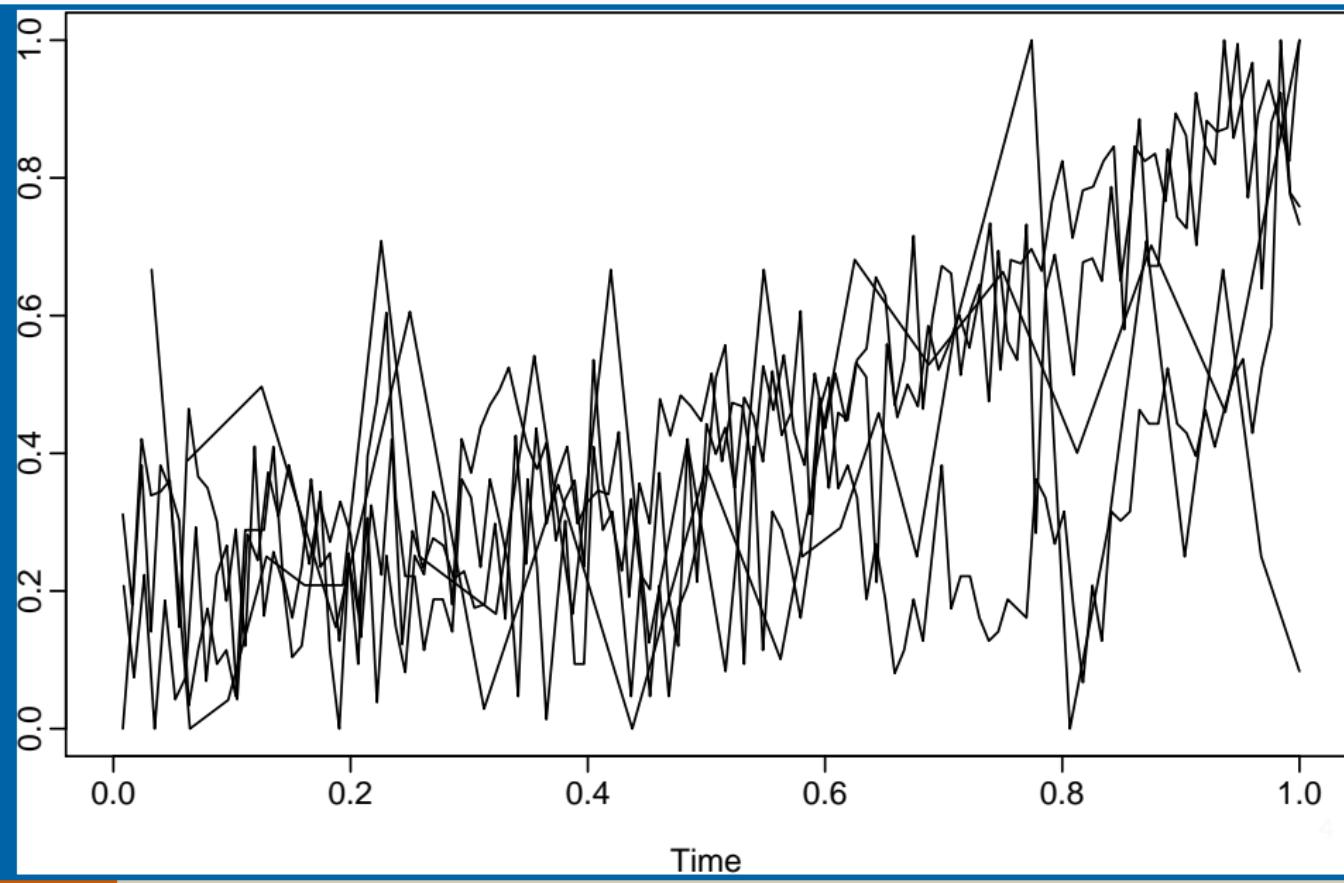
# How to plot lots of time series?



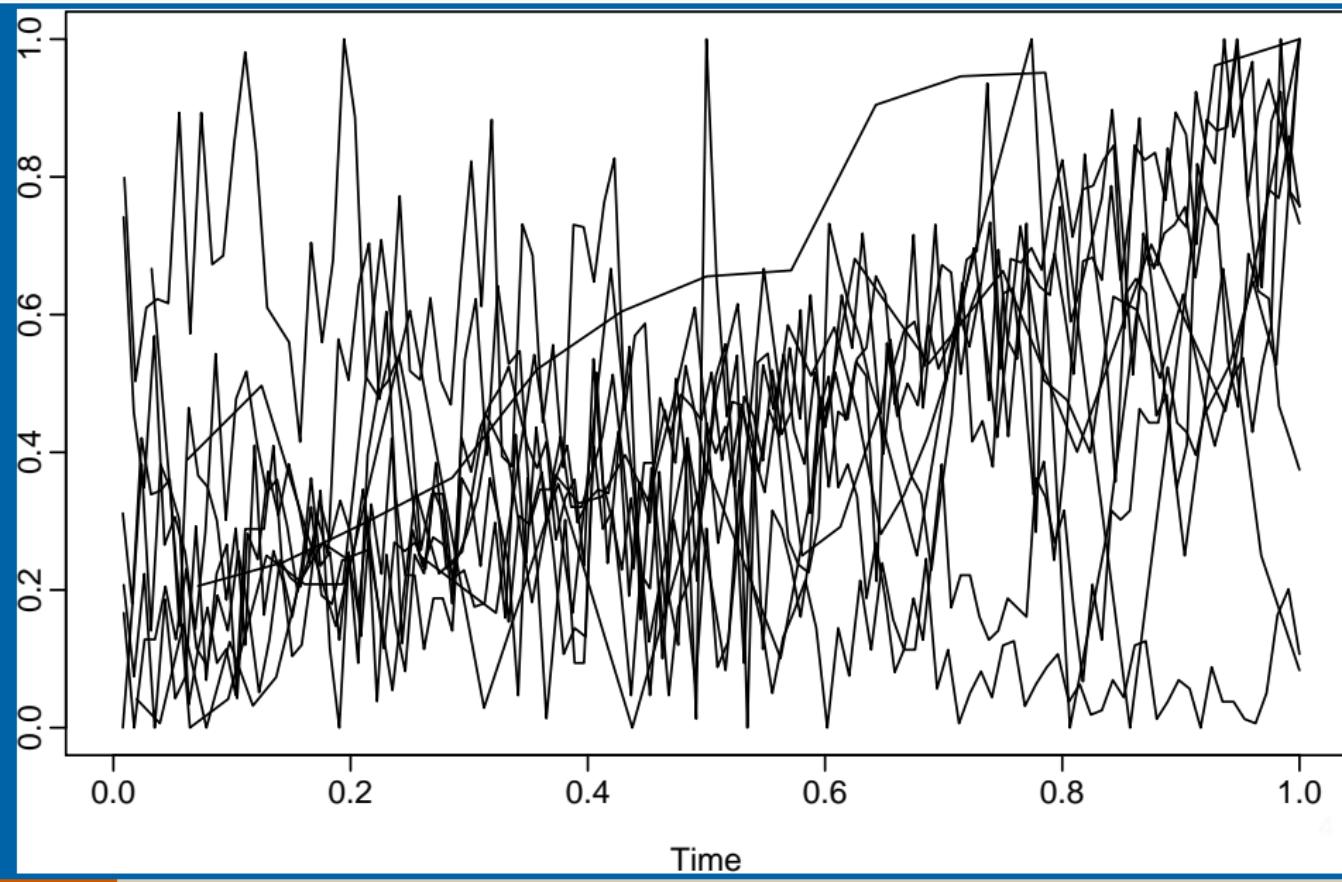
# How to plot lots of time series?



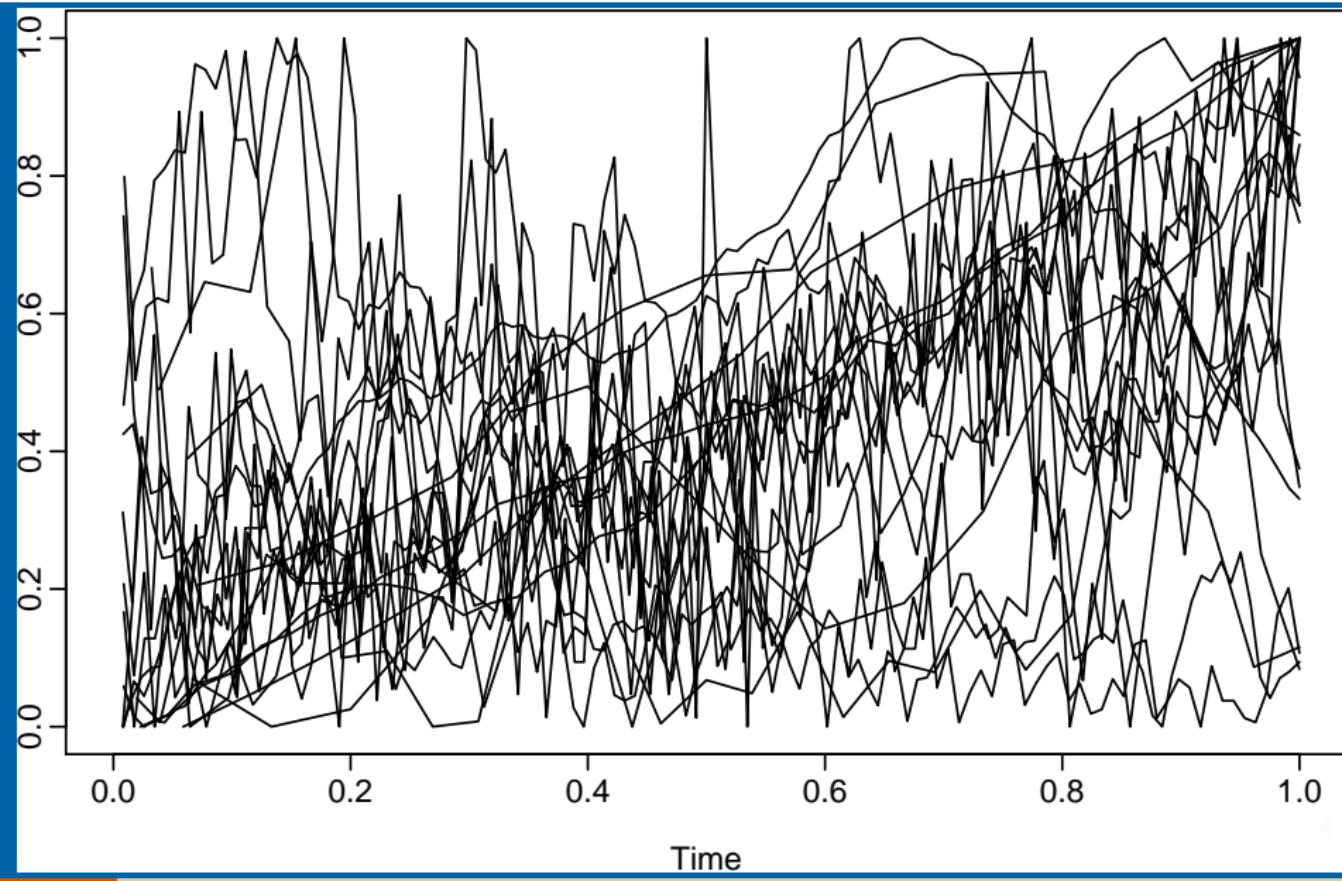
# How to plot lots of time series?



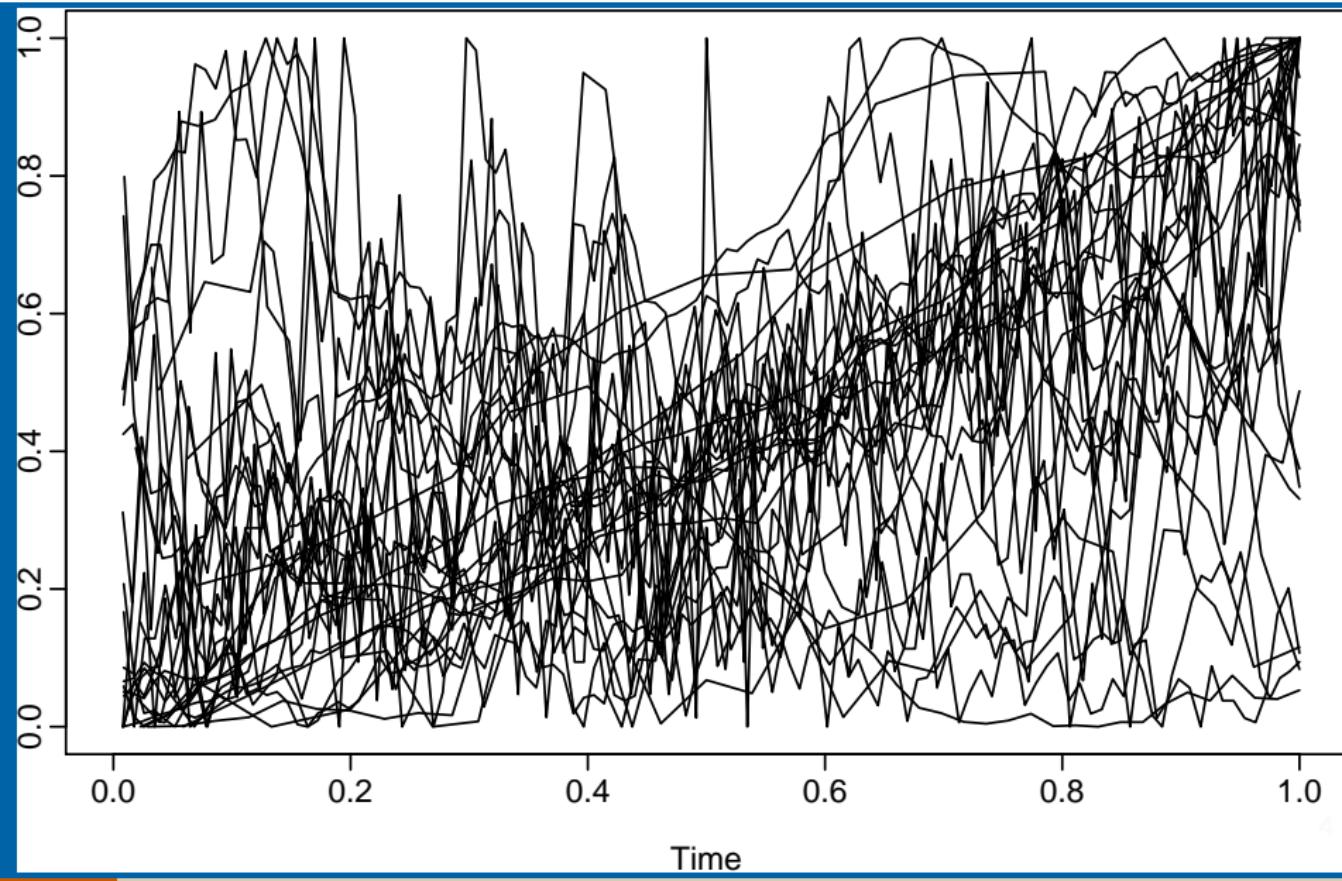
# How to plot lots of time series?



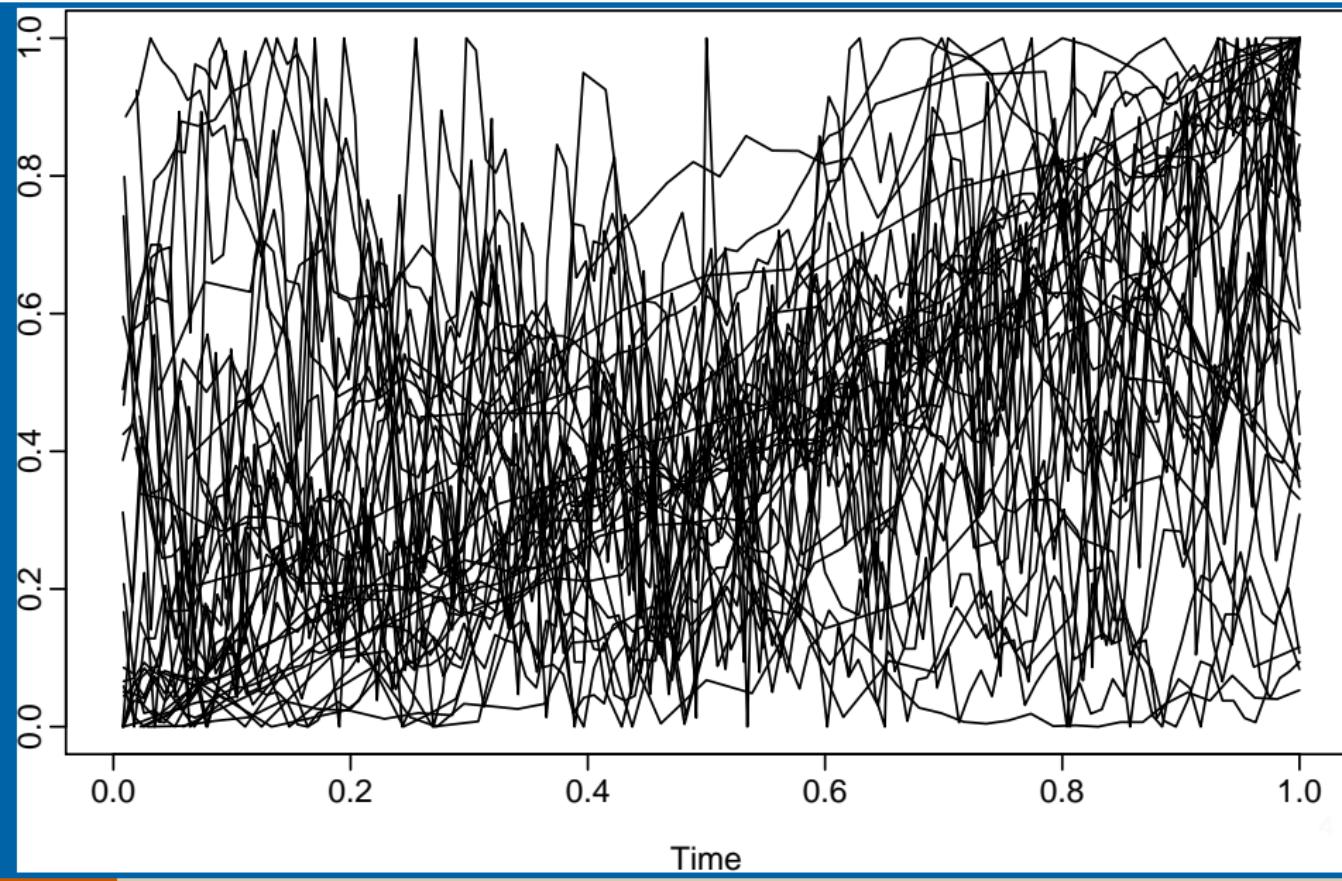
# How to plot lots of time series?



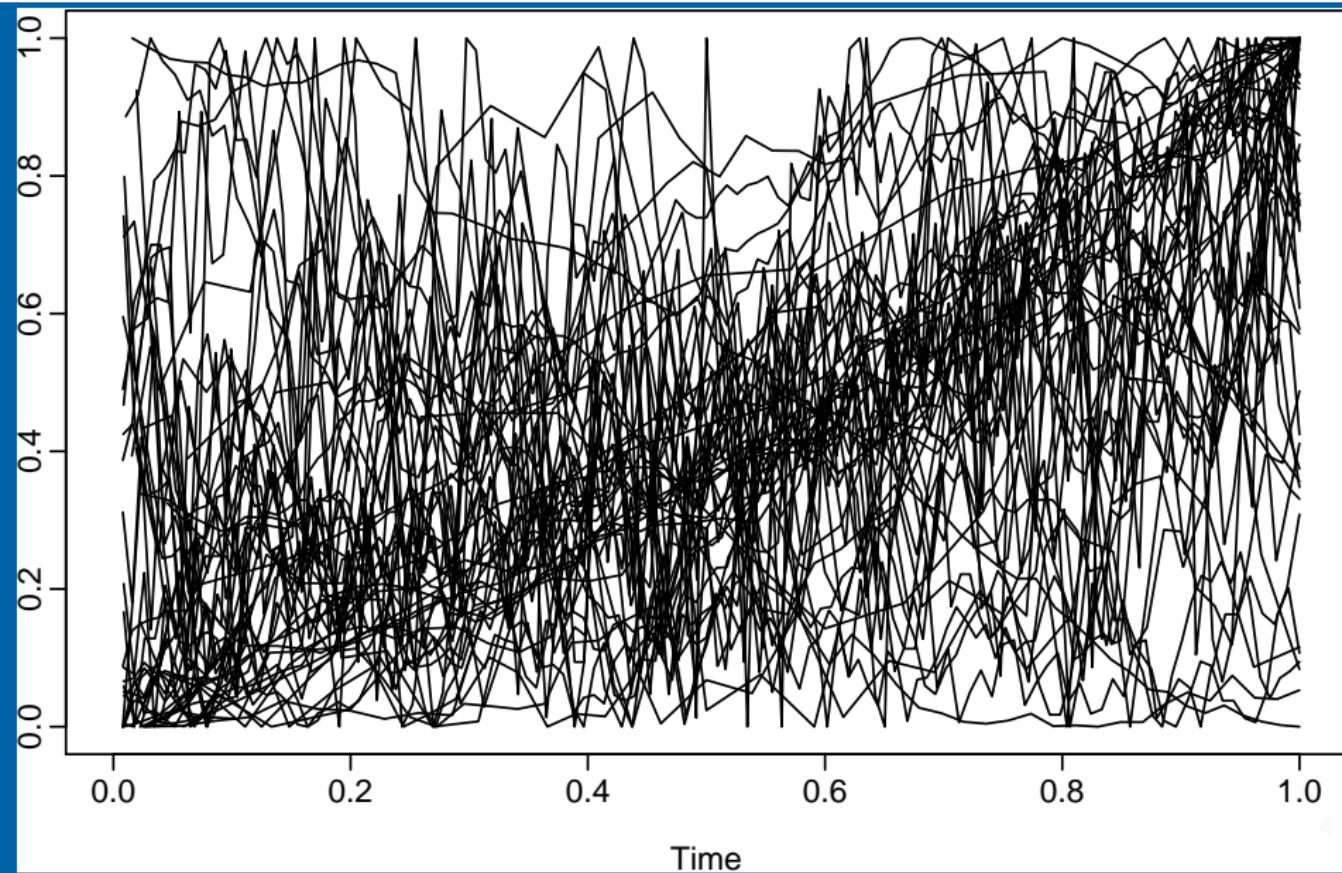
# How to plot lots of time series?



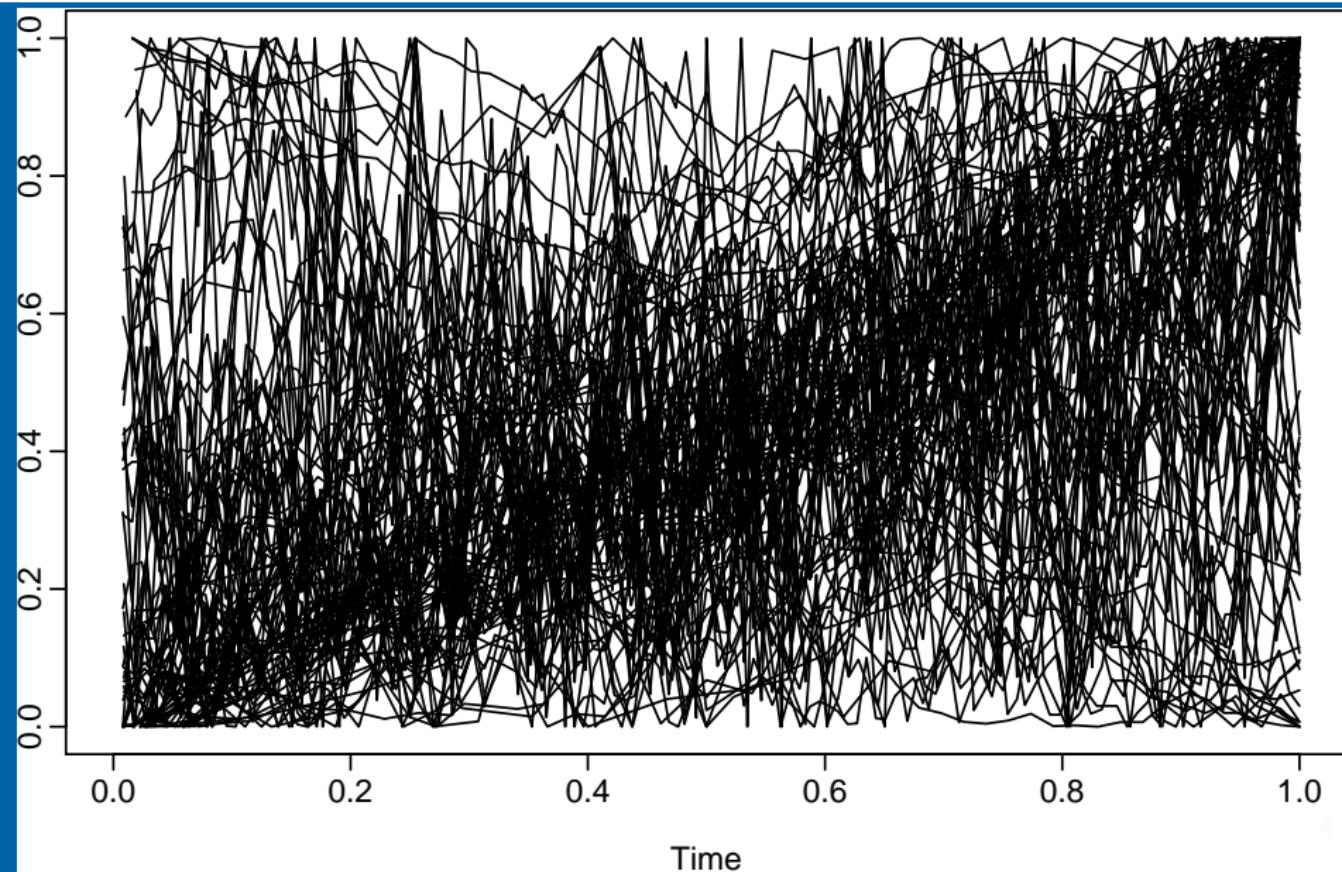
# How to plot lots of time series?



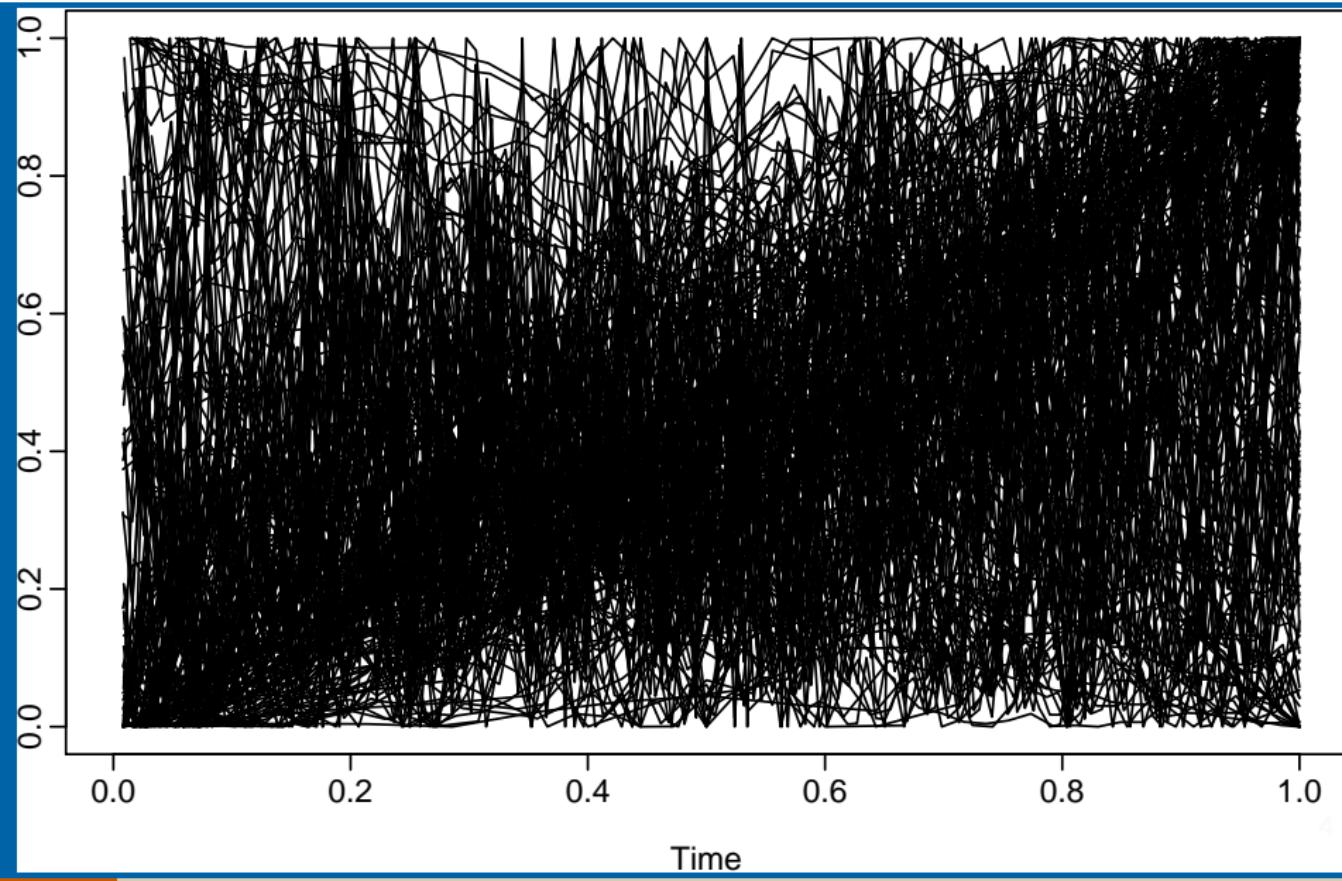
# How to plot lots of time series?



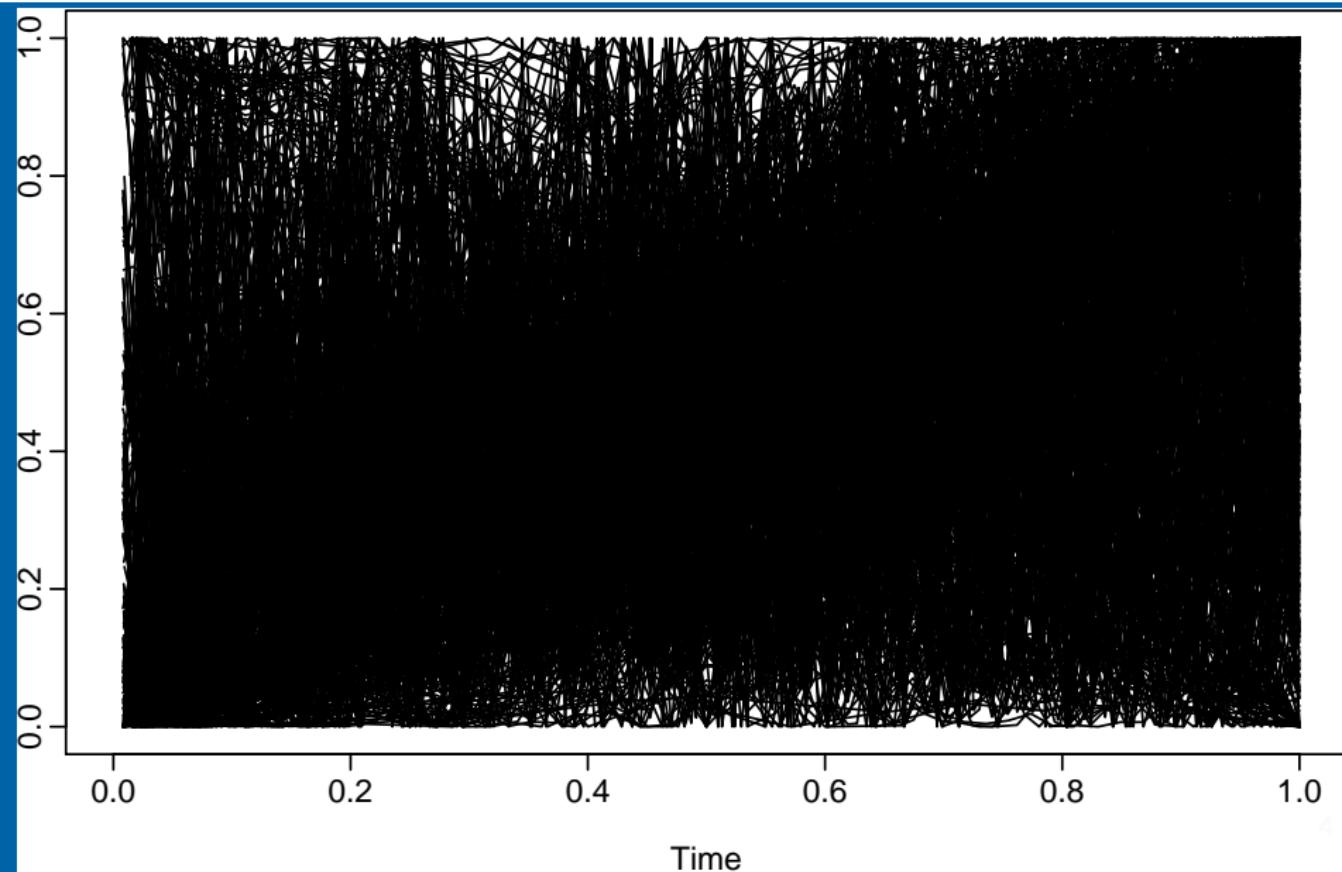
# How to plot lots of time series?



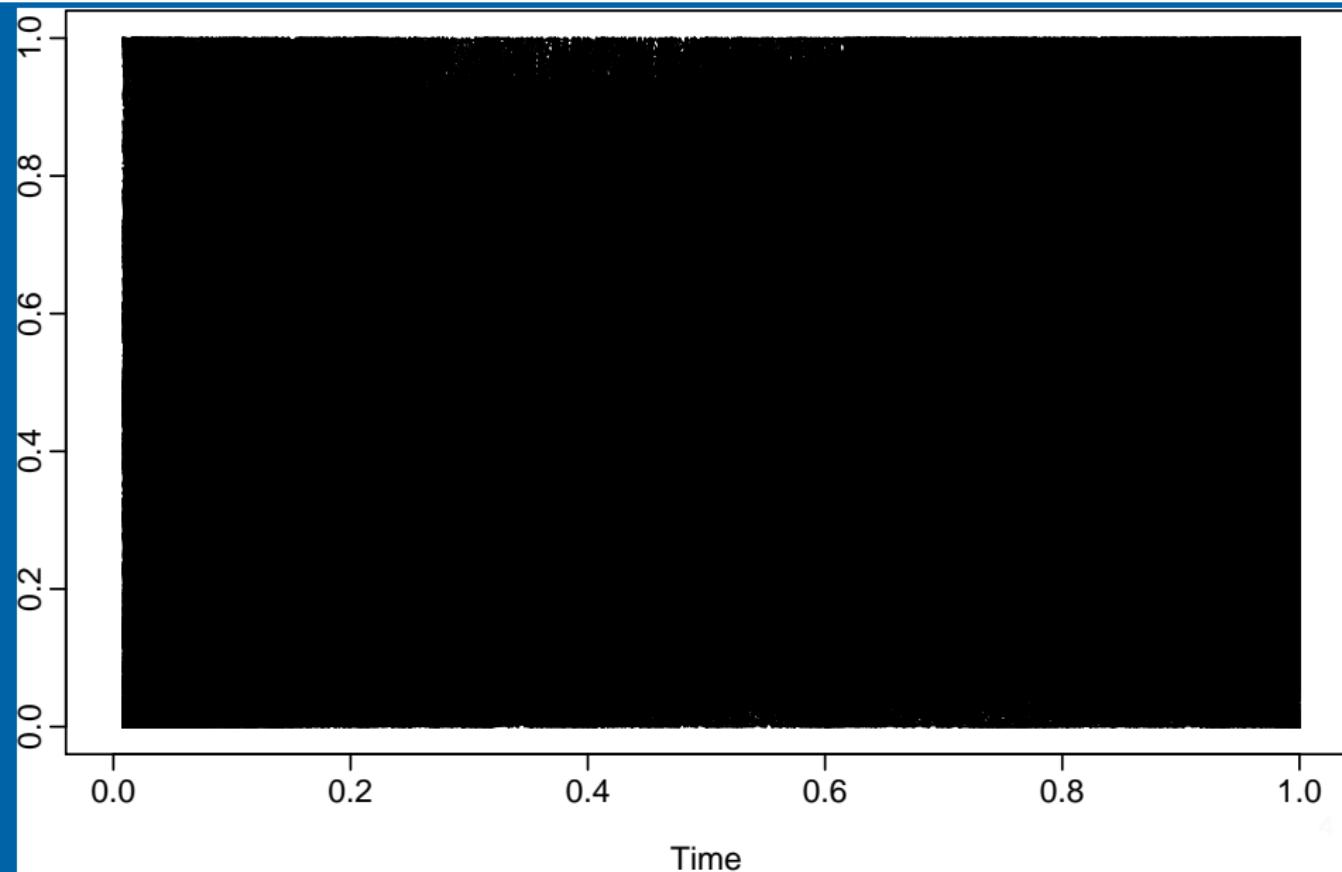
# How to plot lots of time series?



# How to plot lots of time series?



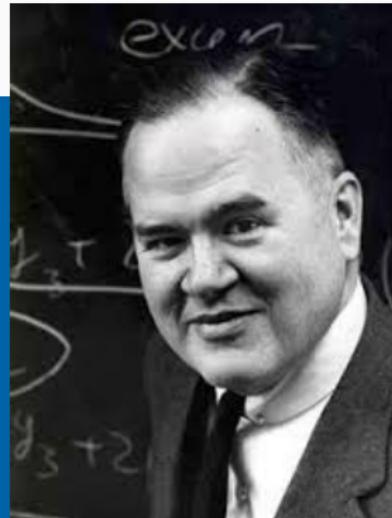
# How to plot lots of time series?



# Key idea

## Cognostics

Computer-produced diagnostics  
(Tukey and Tukey, 1985).



*John W Tukey*

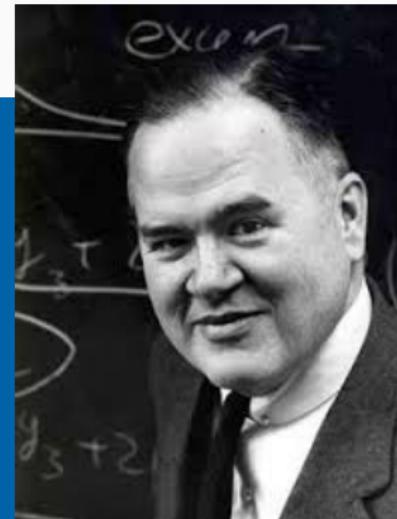
# Key idea

## Cognostics

Computer-produced diagnostics  
(Tukey and Tukey, 1985).

## Examples for time series

- lag correlation
- size and direction of trend
- strength of seasonality
- timing of peak seasonality
- spectral entropy



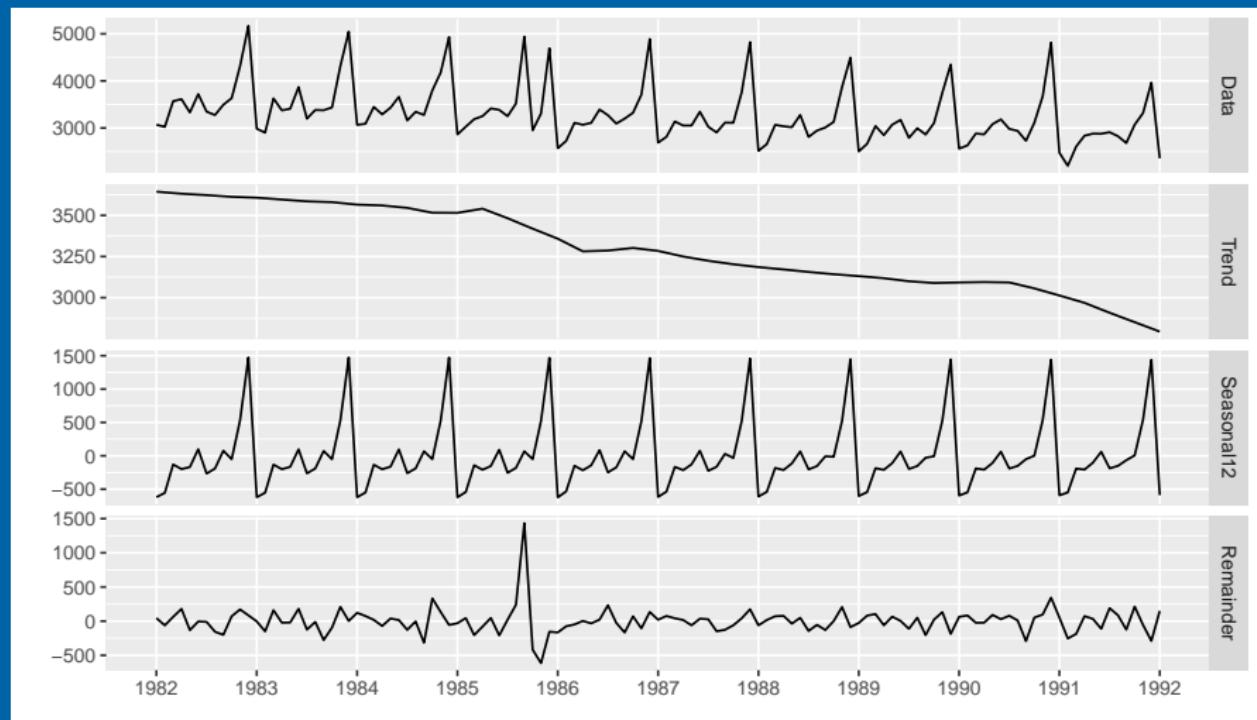
John W Tukey

Called “features” in the machine learning literature.

# An STL decomposition: N2096

$$Y_t = S_t + T_t + R_t$$

$S_t$  is periodic with mean 0



# Candidate features

## STL decomposition

$$Y_t = S_t + T_t + R_t$$

# Candidate features

## STL decomposition

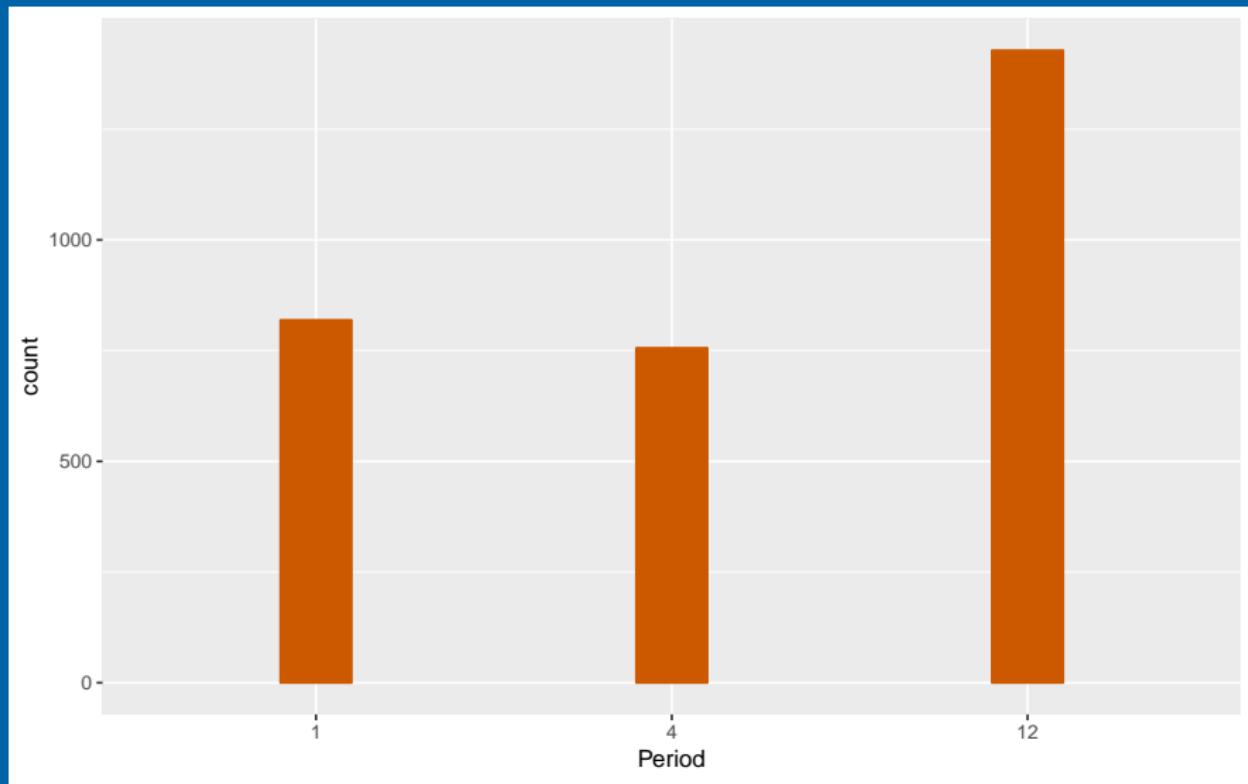
$$Y_t = S_t + T_t + R_t$$

- Seasonal period
- Autocorrelations of data  $(Y_1, \dots, Y_T)$
- Autocorrelations of data  $(R_1, \dots, R_T)$
- Strength of seasonality:  $\max \left( 0, 1 - \frac{\text{Var}(R_t)}{\text{Var}(Y_t - T_t)} \right)$
- Strength of trend:  $\max \left( 0, 1 - \frac{\text{Var}(R_t)}{\text{Var}(Y_t - S_t)} \right)$
- Spectral entropy:  $H = - \int_{-\pi}^{\pi} f_y(\lambda) \log f_y(\lambda) d\lambda$ , where  $f_y(\lambda)$  is spectral density of  $Y_t$ .  
Low values of  $H$  suggest a time series that is easier to forecast (more signal).
- Optimal Box-Cox transformation of data

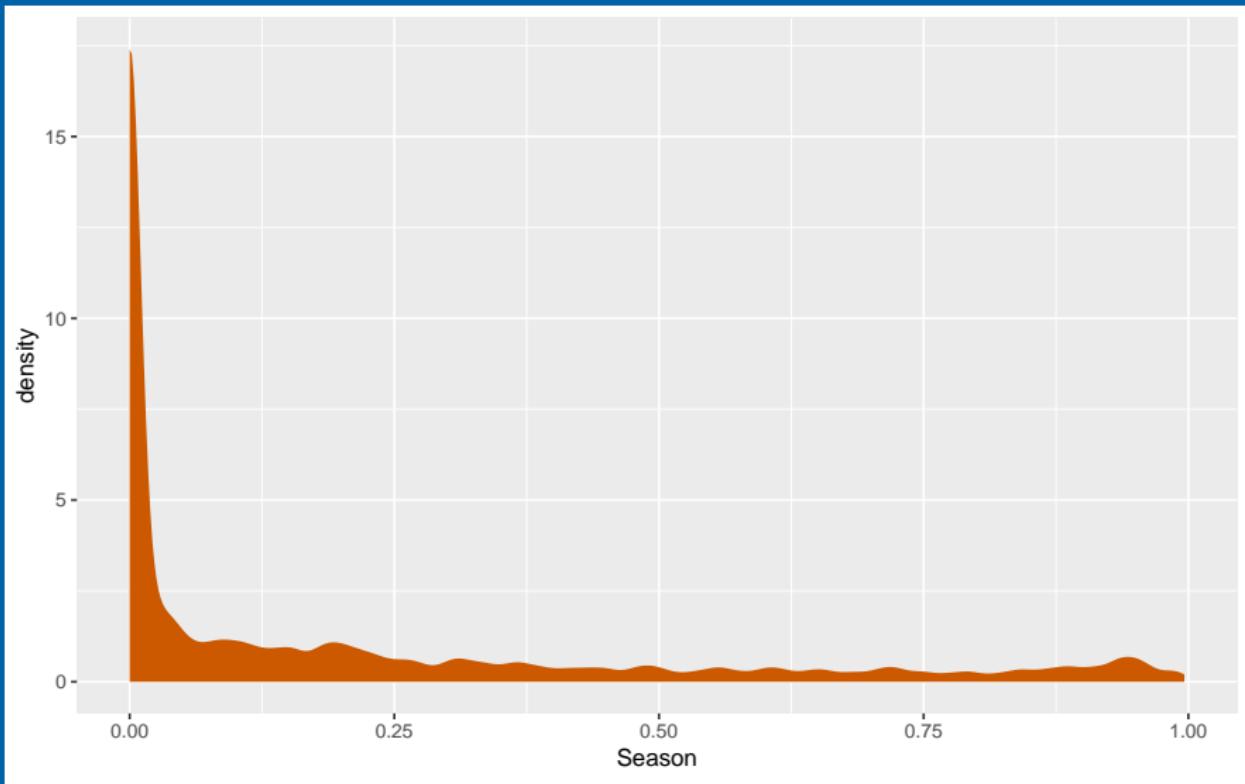
# tsfeatures package

```
library(tsfeatures)
lambda_stl <- function(x,...) {
  lambda <- forecast::BoxCox.lambda(x,
    lower=0, upper=1, method='loglik')
  y <- forecast::BoxCox(x, lambda)
  c(stl_features(y,s.window='periodic', robust=TRUE, ...),
    lambda=lambda)
}
M3Features <- bind_cols(
  tsfeatures(M3data, c("frequency", "entropy")),
  tsfeatures(M3data, "lambda_stl", scale=FALSE)) %>%
  select(frequency, entropy, trend, seasonal_strength,
    e_acf1, lambda) %>%
  replace_na(list(seasonal_strength=0)) %>%
  rename(
    Frequency = frequency,
    Entropy = entropy,
    Trend = trend,
    Season = seasonal_strength,
    ACF1 = e_acf1,
    Lambda = lambda) %>%
  mutate(Period = as.factor(Frequency))
```

# Distribution of Period for M3

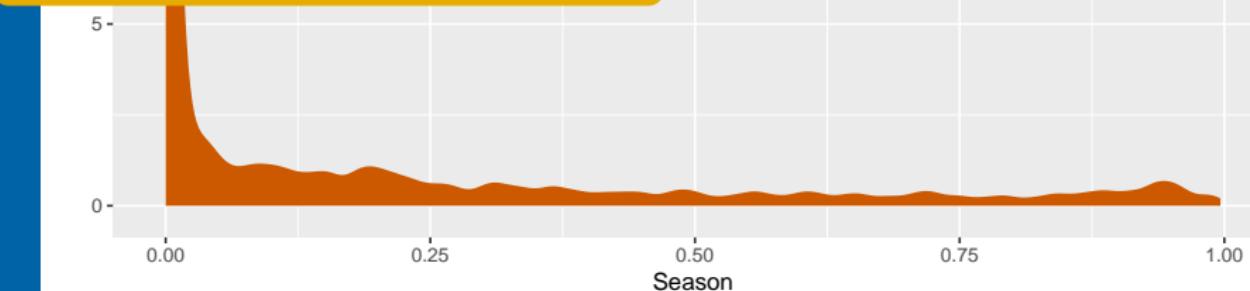
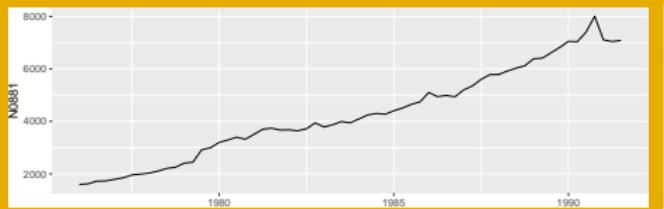


# Distribution of Seasonality for M3

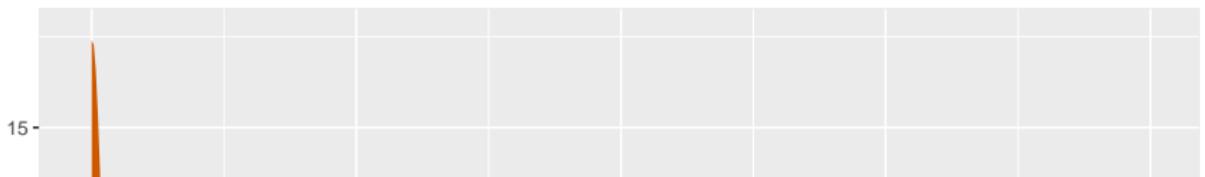


# Distribution of Seasonality for M3

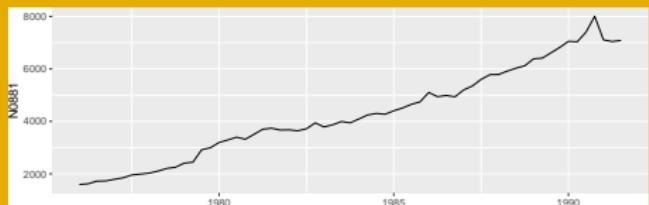
Low Seasonality



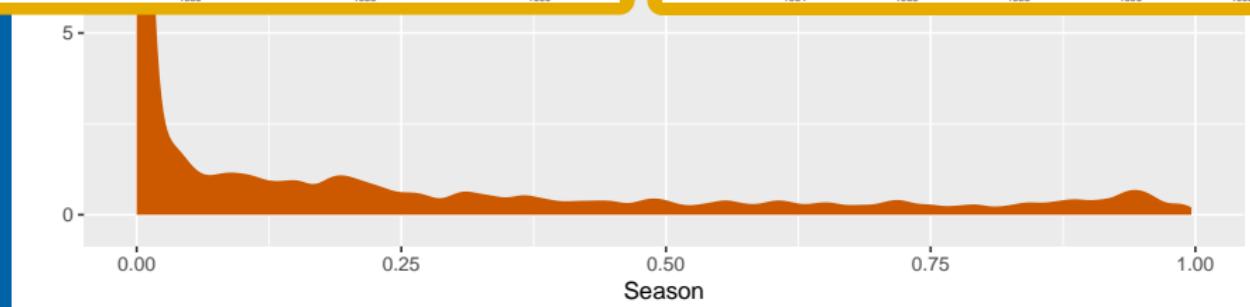
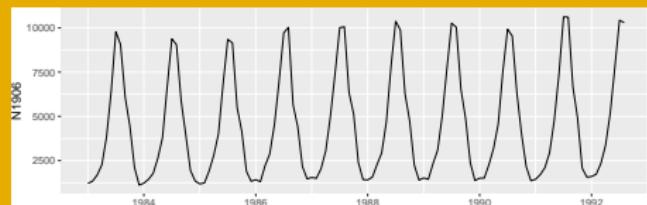
# Distribution of Seasonality for M3



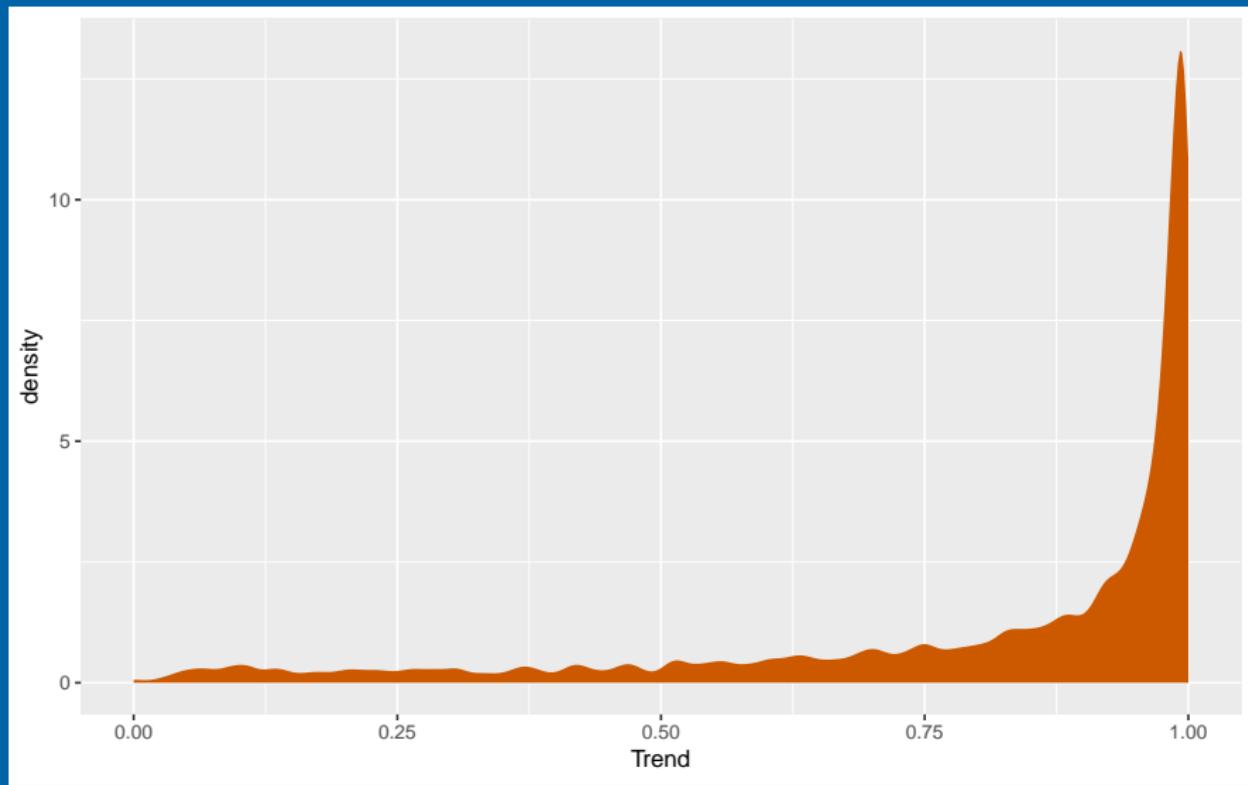
Low Seasonality



High Seasonality

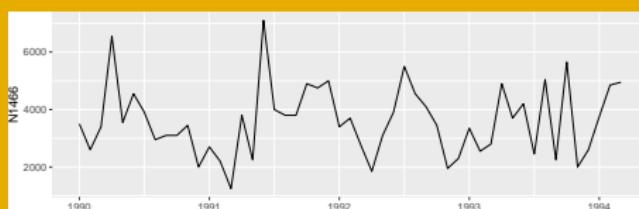


# Distribution of Trend for M3



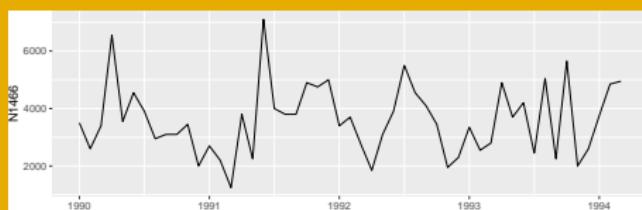
# Distribution of Trend for M3

Low Trend

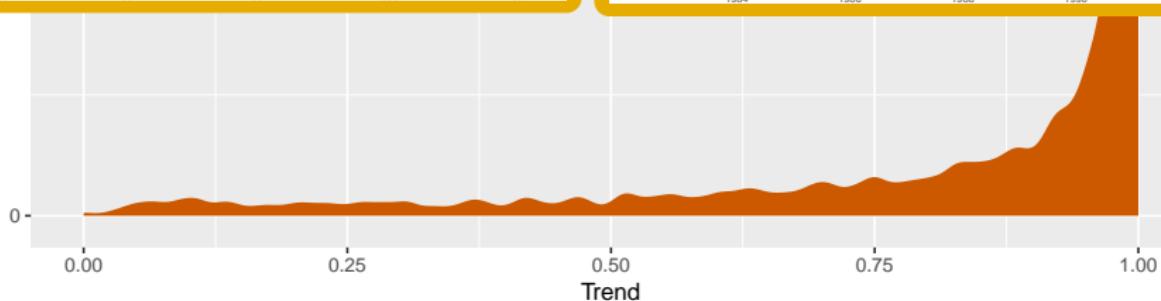
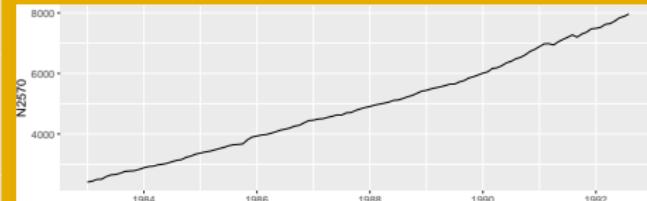


## Distribution of Trend for M3

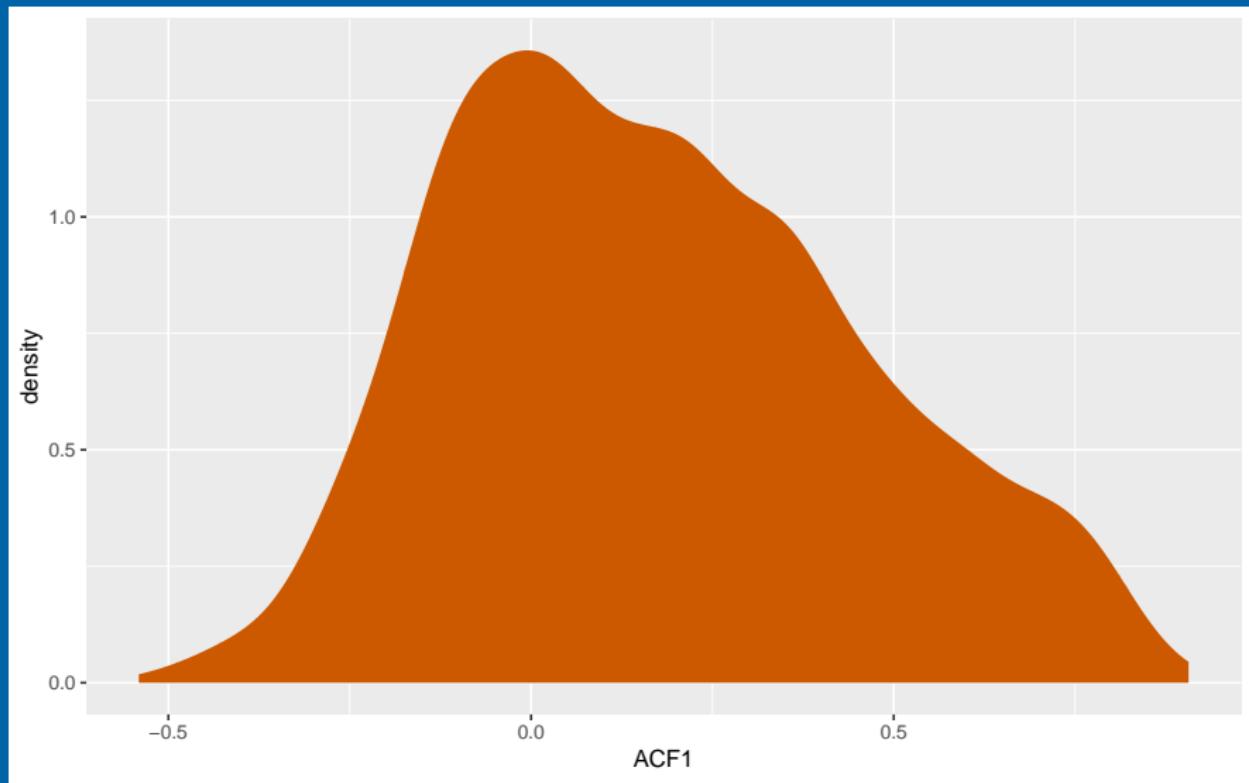
## Low Trend



## High Trend

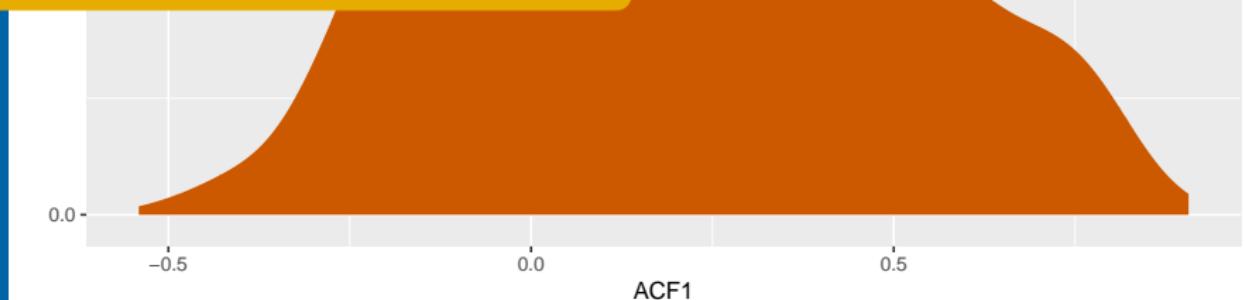
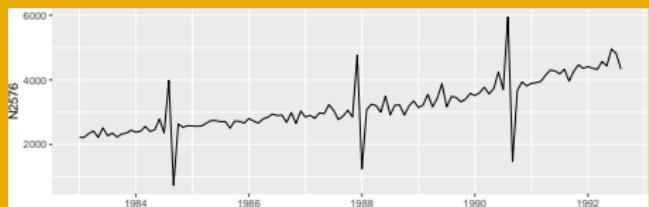


# Distribution of Residual ACF1 for M3



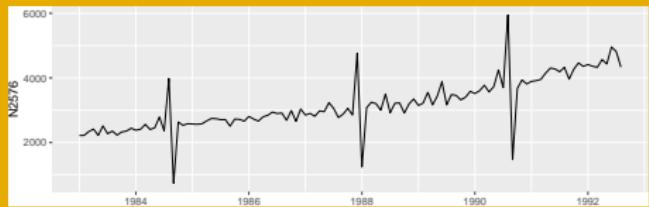
# Distribution of Residual ACF1 for M3

Low ACF1

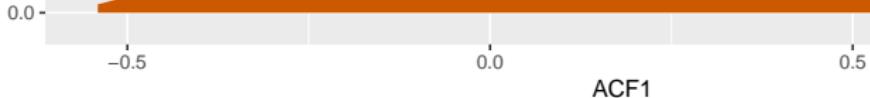
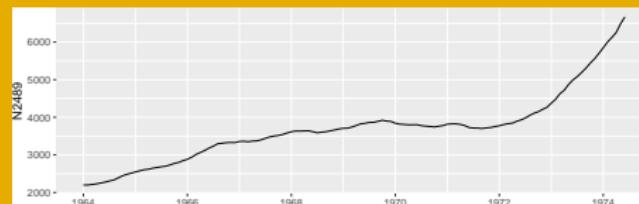


# Distribution of Residual ACF1 for M3

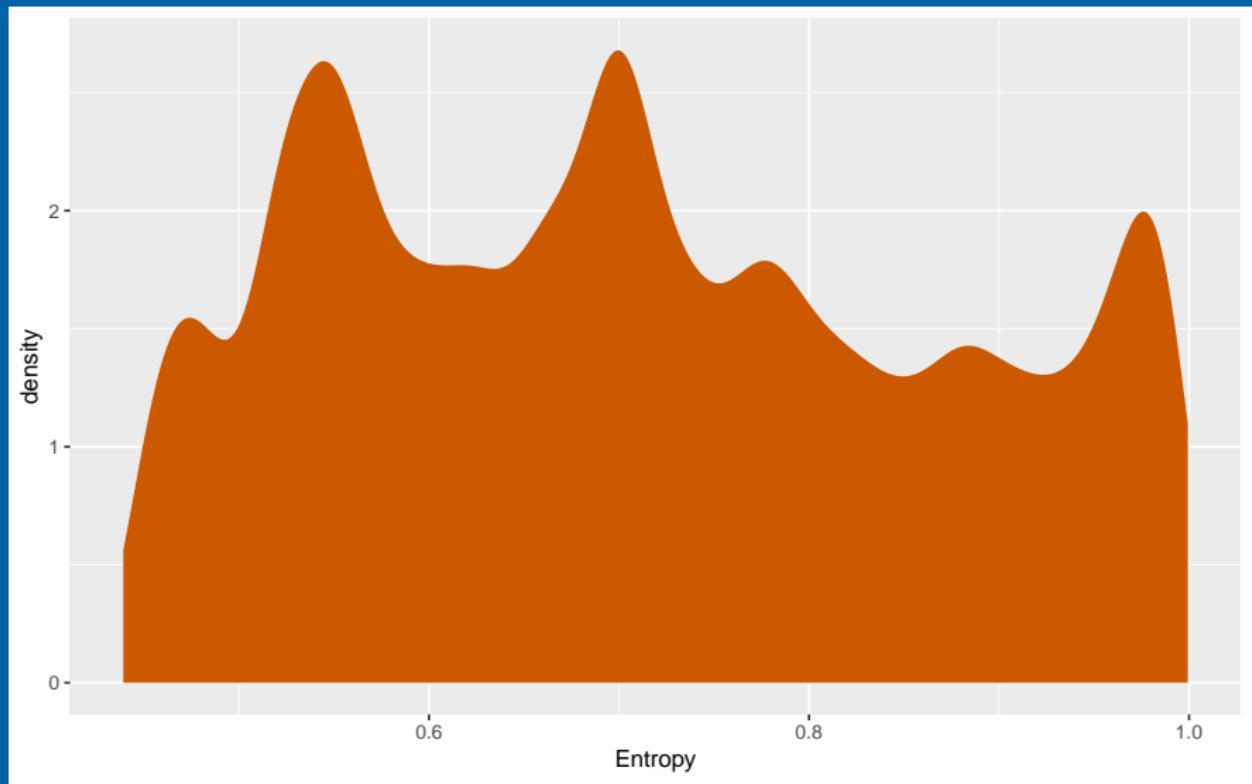
Low ACF1



High ACF1

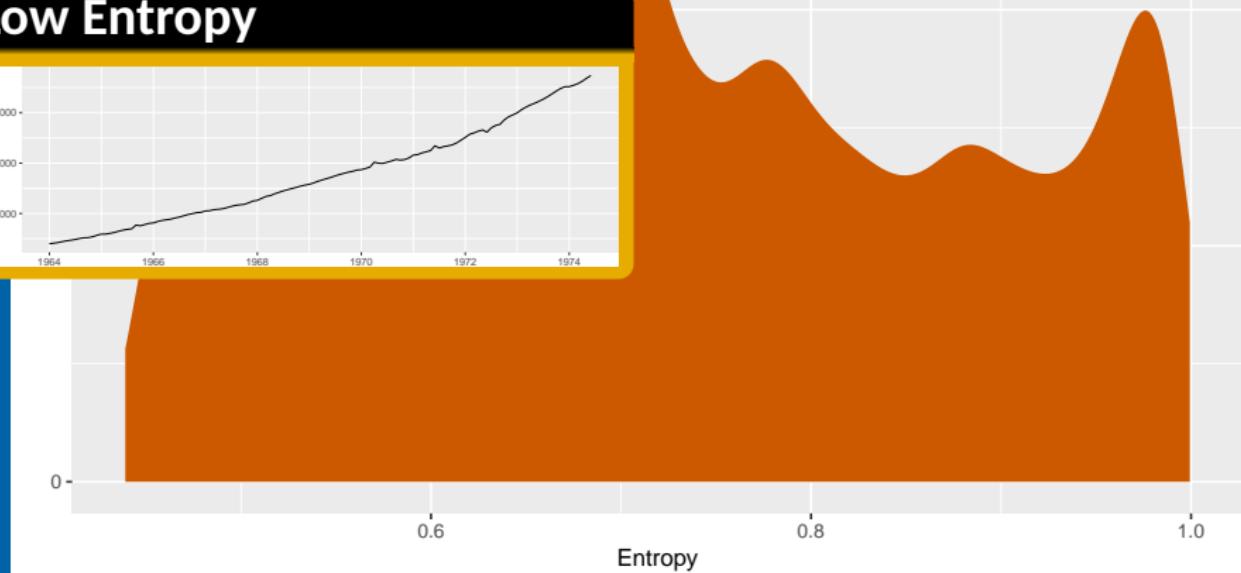
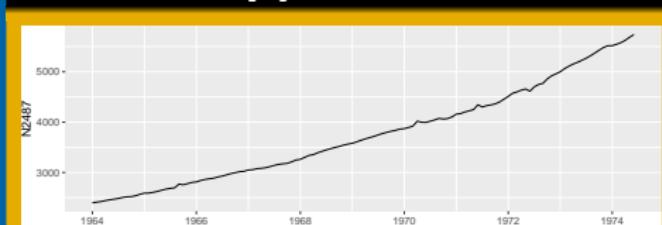


# Distribution of Spectral Entropy for M3



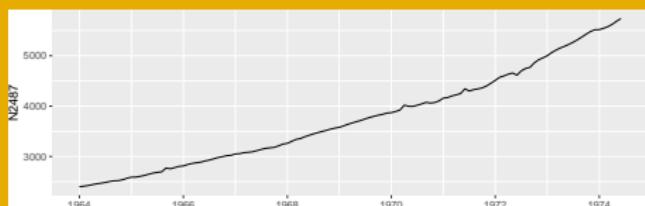
# Distribution of Spectral Entropy for M3

Low Entropy

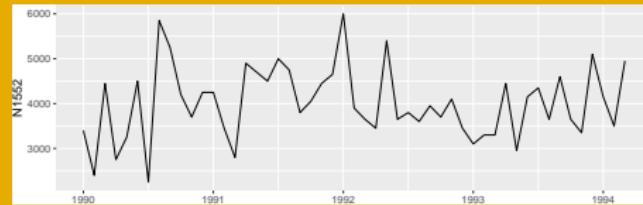


# Distribution of Spectral Entropy for M3

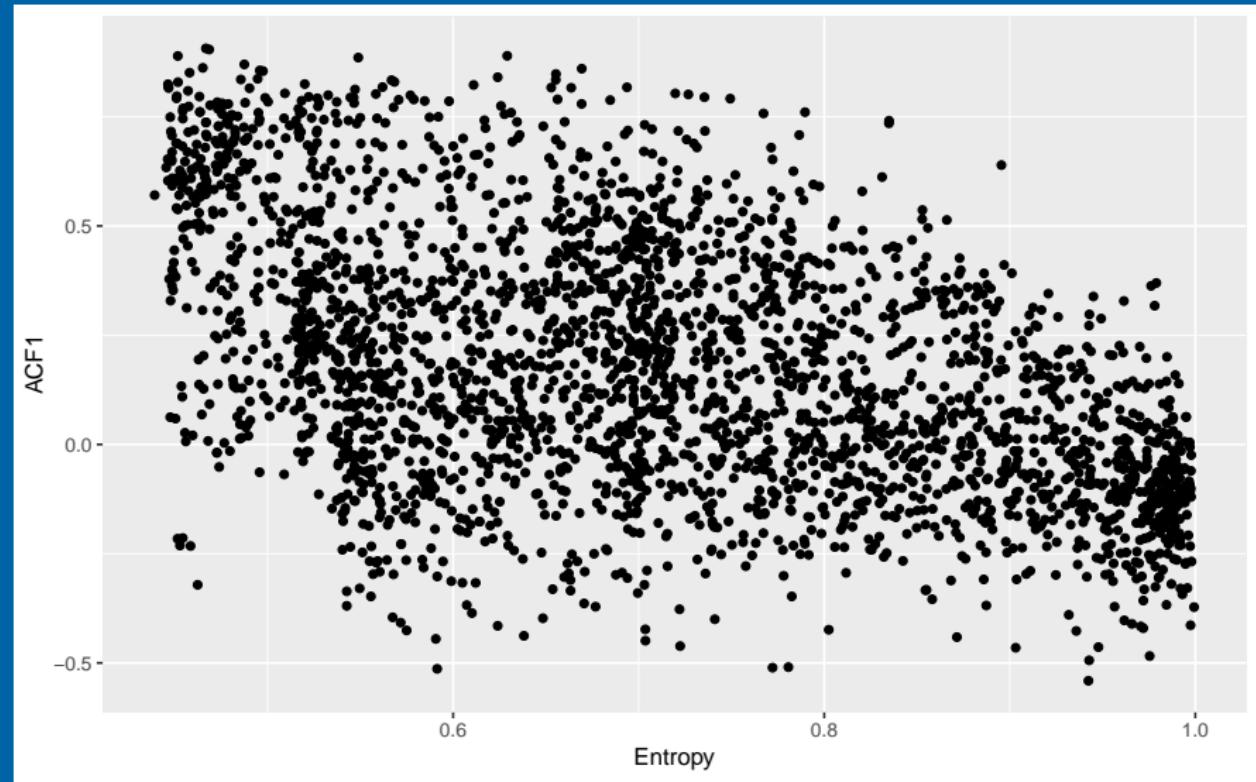
Low Entropy



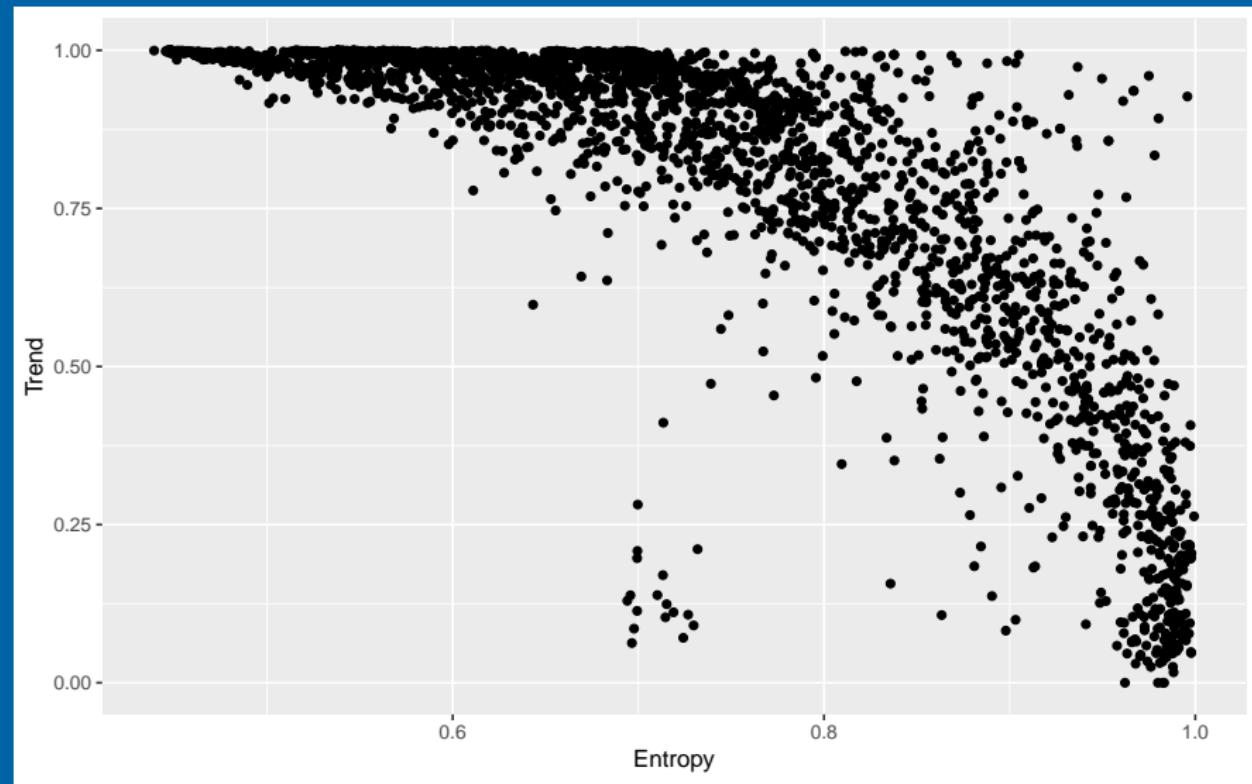
High Entropy



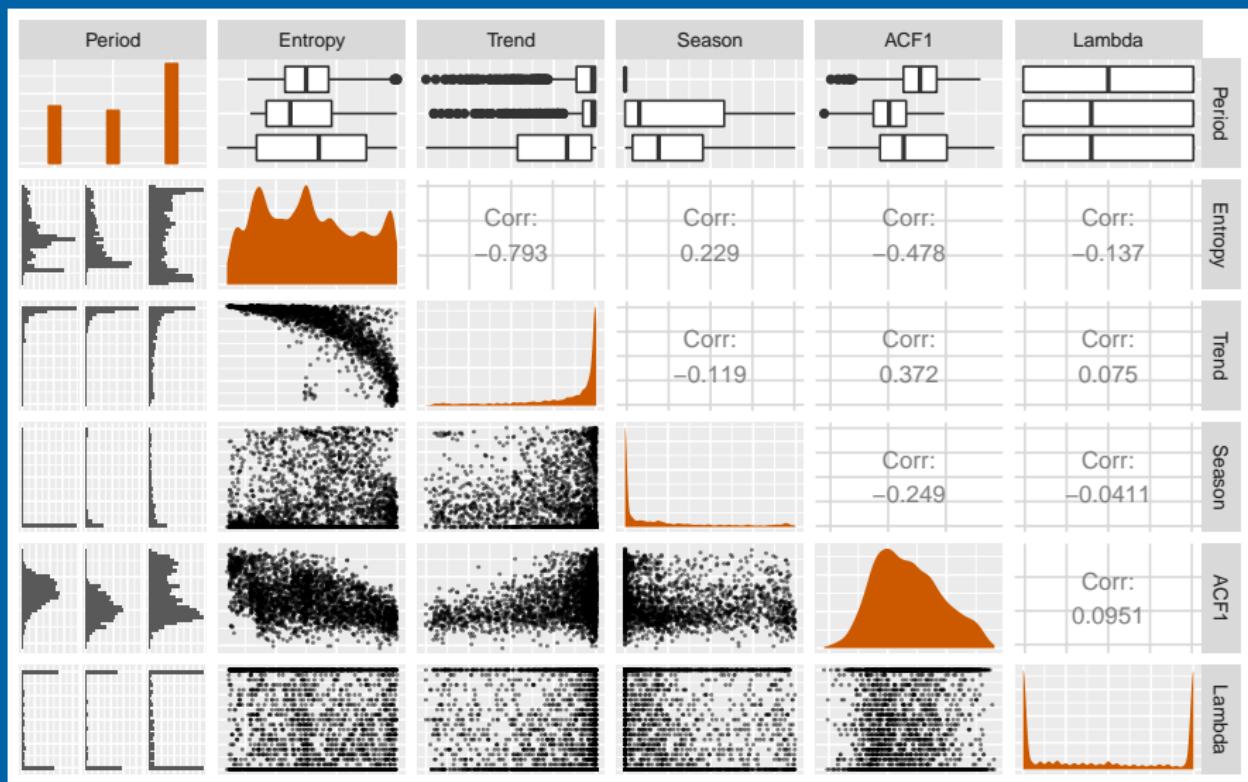
# Feature distributions



# Feature distributions



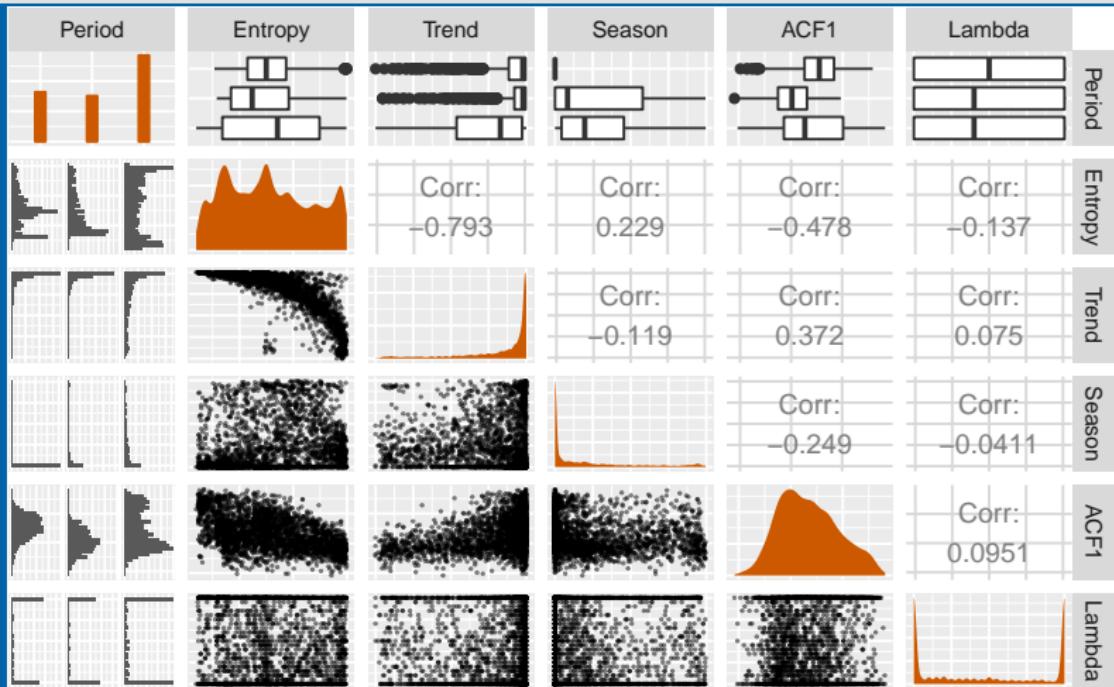
# Feature distributions



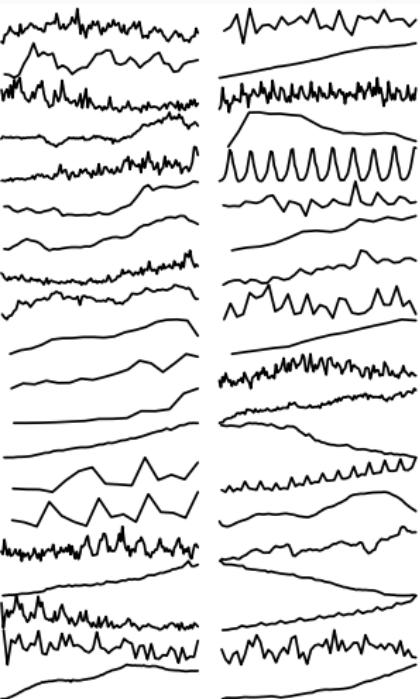
# Feature distributions

M3Features %>%

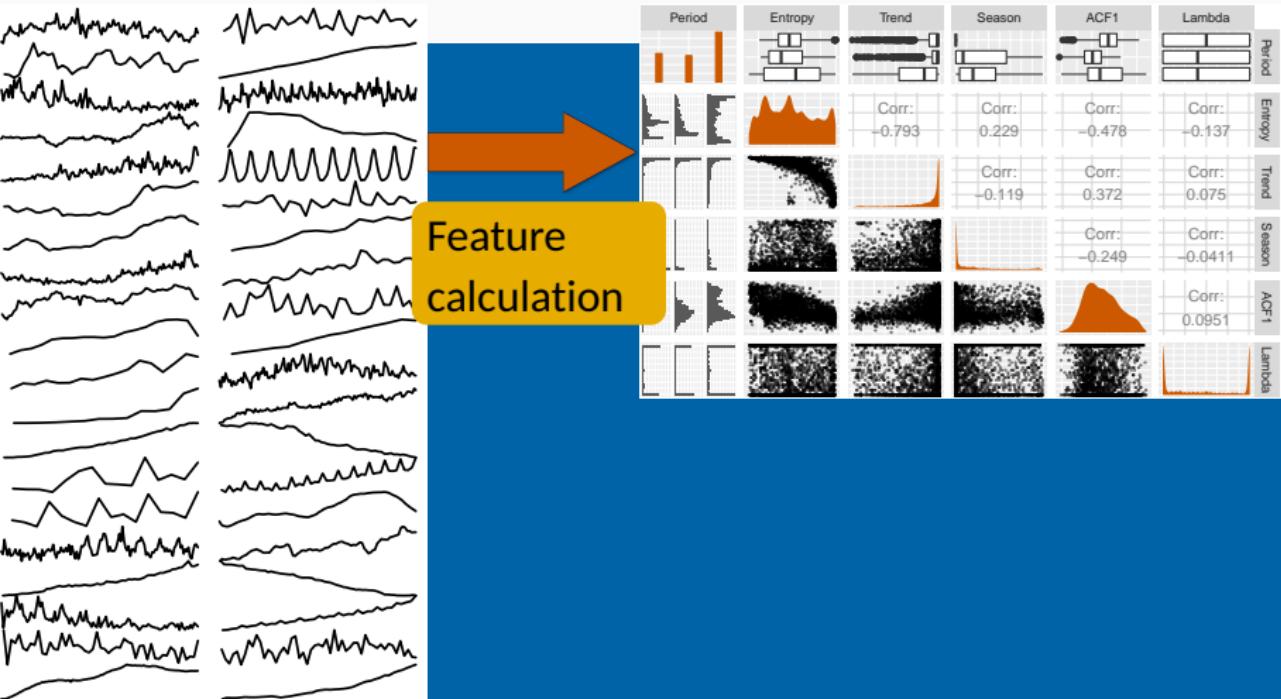
```
select(Period, Entropy, Trend, Season, ACF1, Lambda) %>%  
GGally::ggpairs()
```



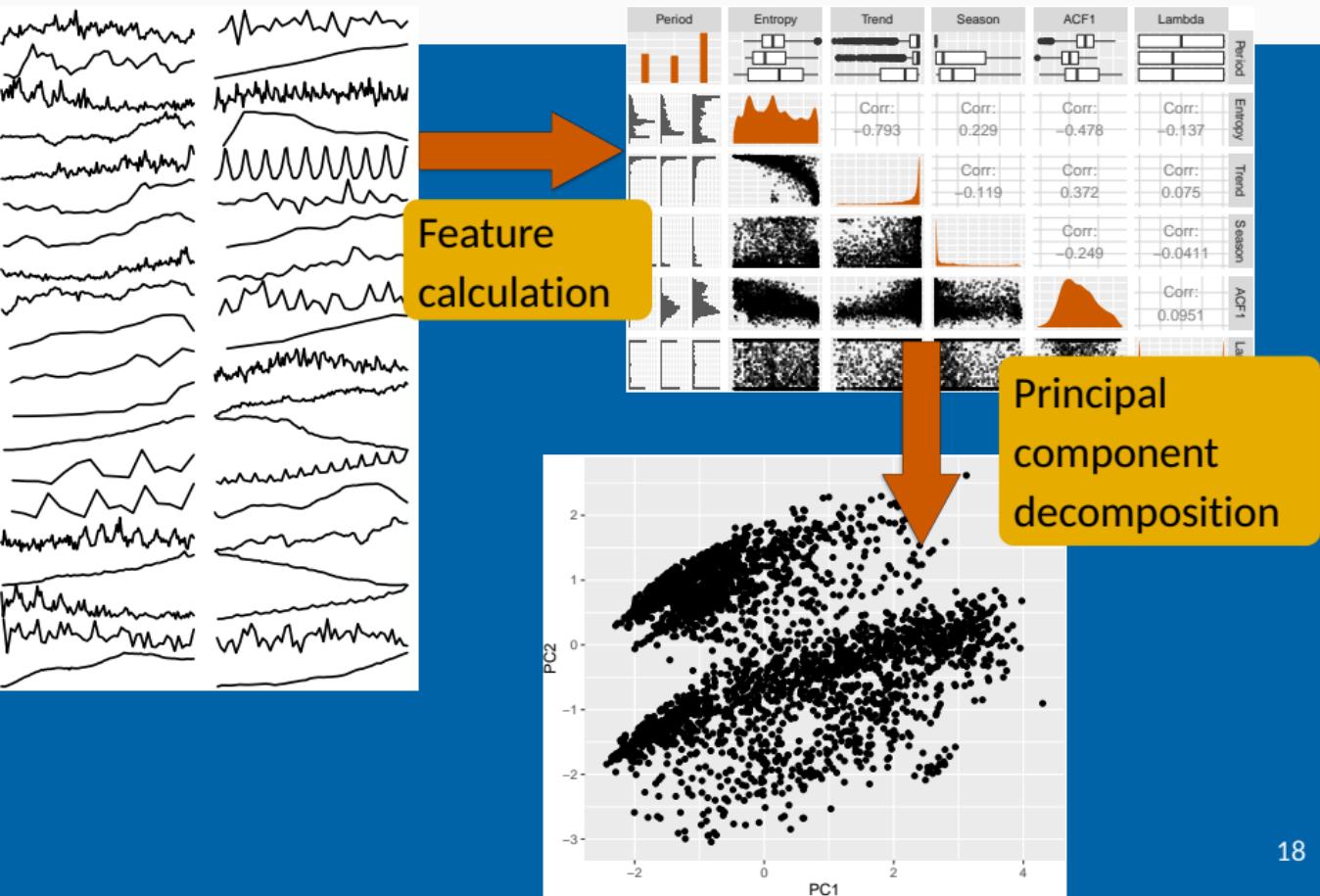
# Dimension reduction for time series



# Dimension reduction for time series

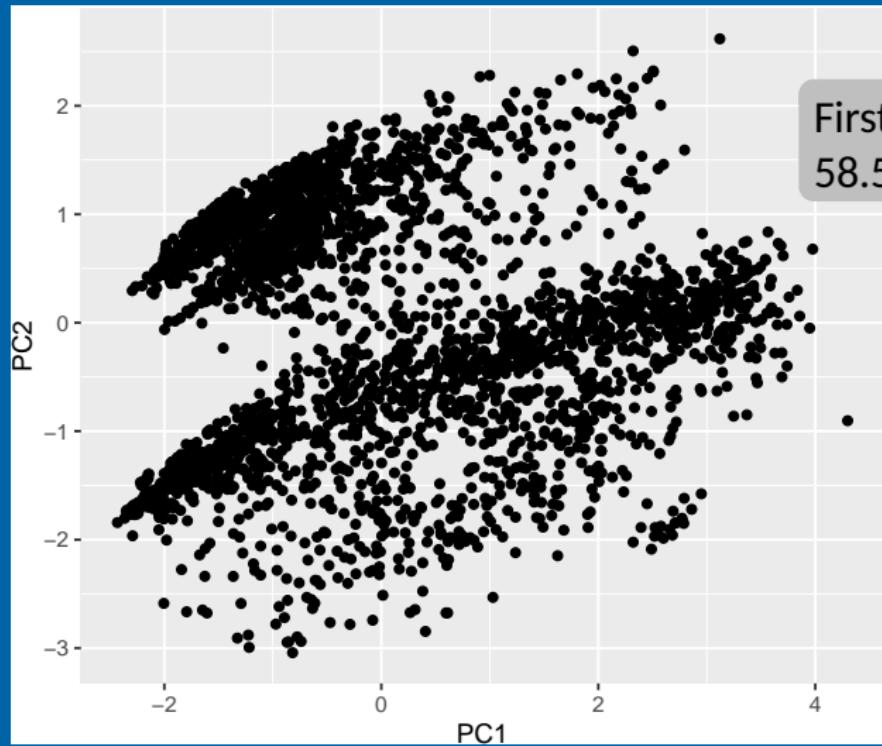


# Dimension reduction for time series



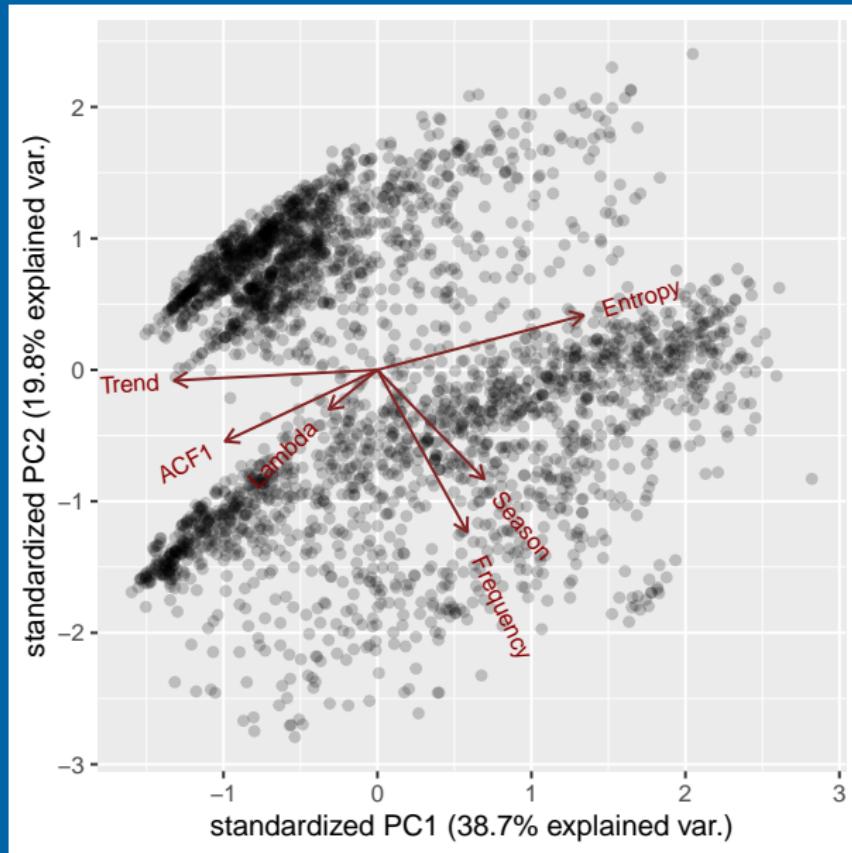
# M3 feature space

```
prcomp(select(M3Features, -Period), scale=TRUE)$x %>%  
  ggplot(aes(x=PC1, y=PC2))
```

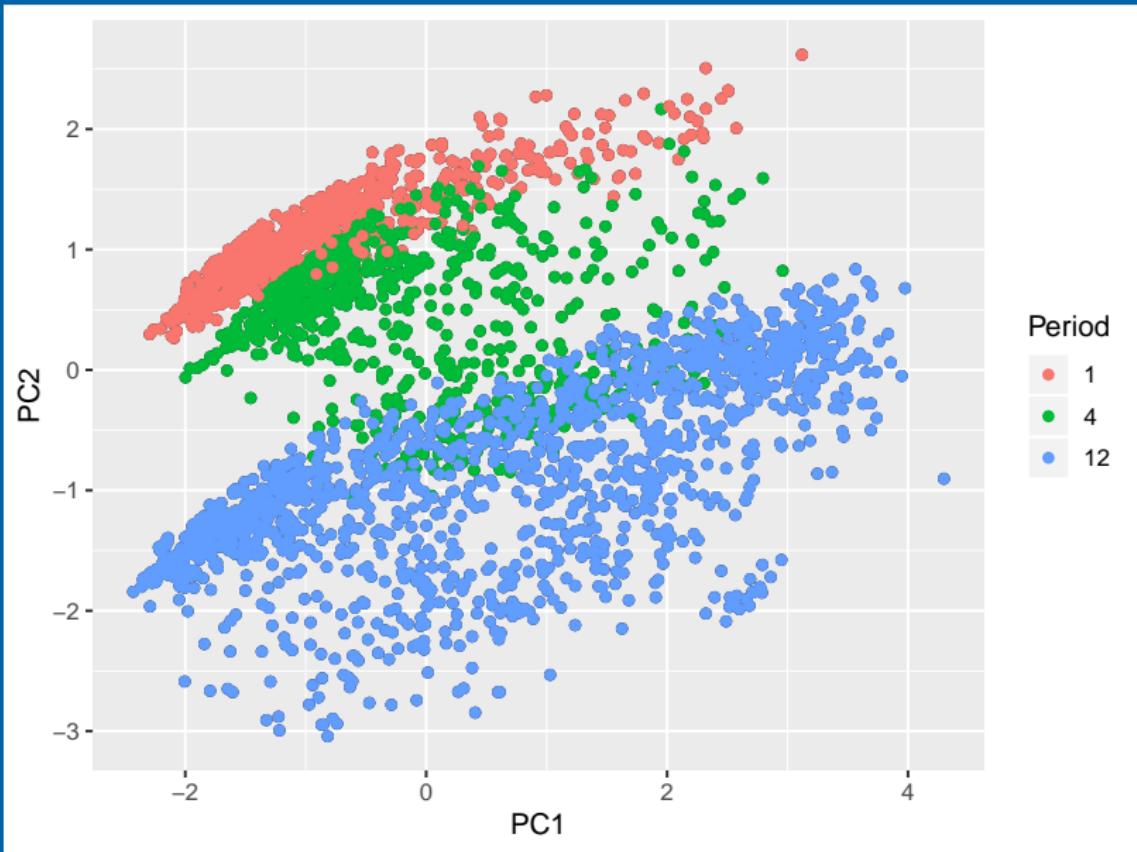


First two PCs explain  
58.5% of the variance.

# M3 feature space



# M3 feature space



# Papers and packages



Kang, Hyndman, & Smith-Miles, K. (2017)  
Visualising forecasting algorithm performance using time series instance spaces.  
*IJF*, 33(2) 345–358.



Hyndman, Wang, Kang, Talagala &  
Montero-Manso (2018). **tsfeatures**: Time  
Series Feature Extraction.  
[github.com/robjhyndman/tsfeatures/](https://github.com/robjhyndman/tsfeatures)

# Outline

1 M3 forecasting competition

2 Yahoo email servers

3 Irish smart metres

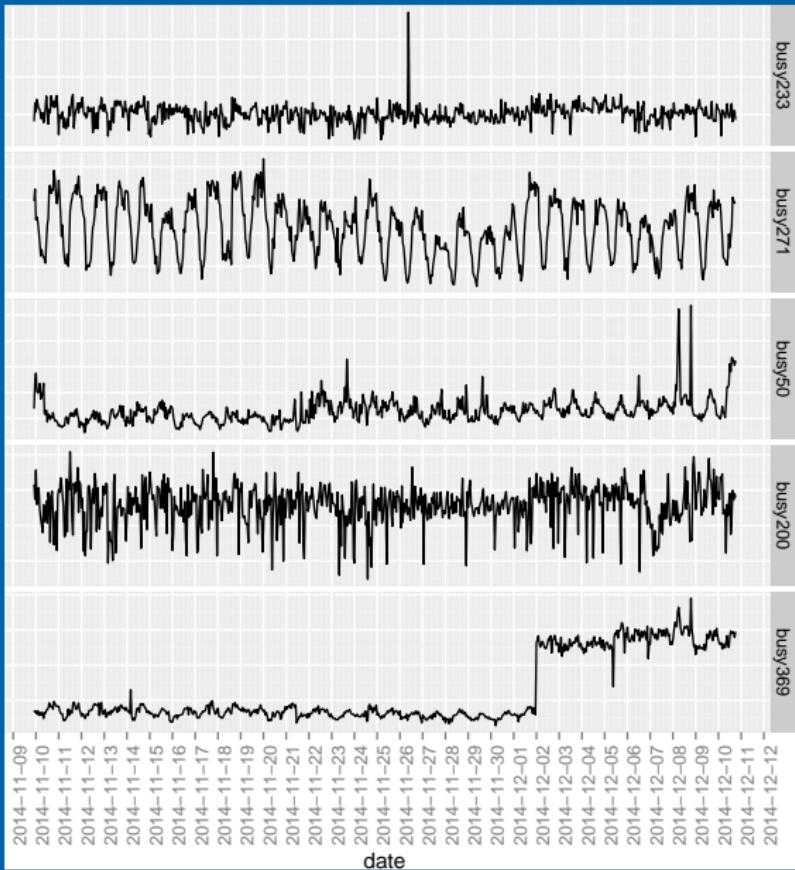
4 Packages

# Yahoo email servers

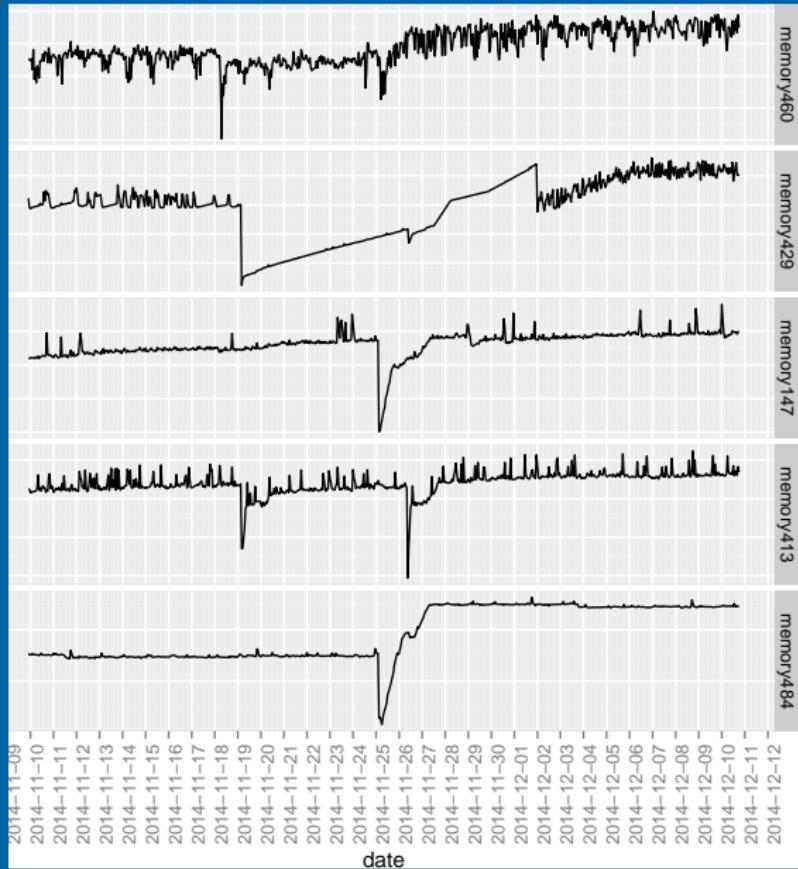
- Tens of thousands of time series collected at one-hour intervals over one month.
- Consisting of several server metrics (e.g. CPU usage and paging views) from many server farms globally.
- Aim: find unusual (anomalous) time series.



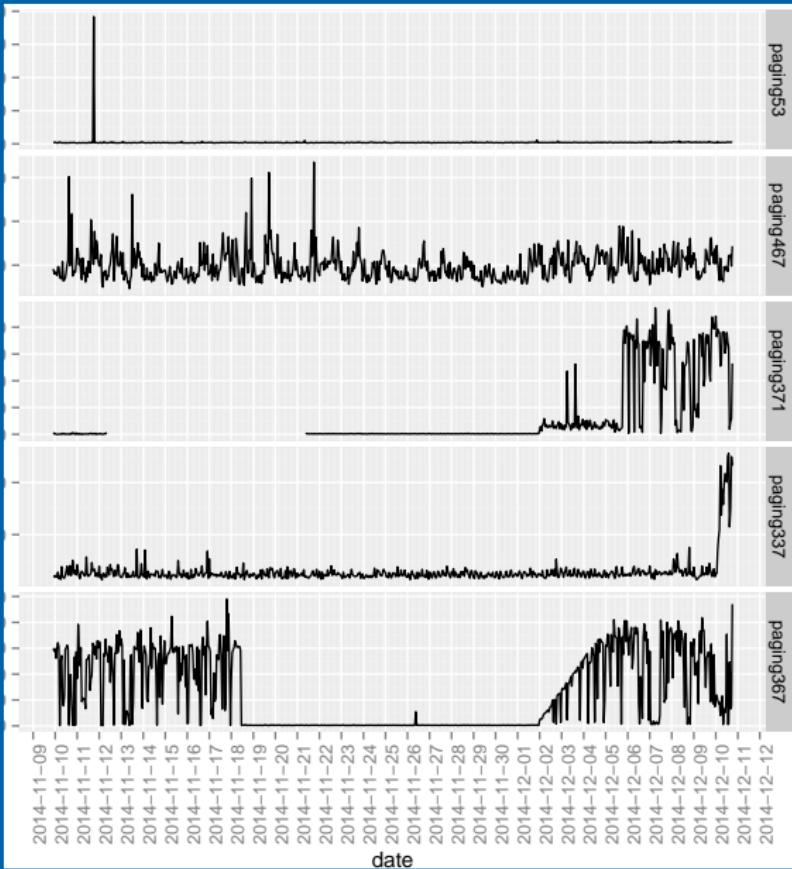
# Yahoo email servers



# Yahoo email servers



# Yahoo email servers



# Yahoo email servers

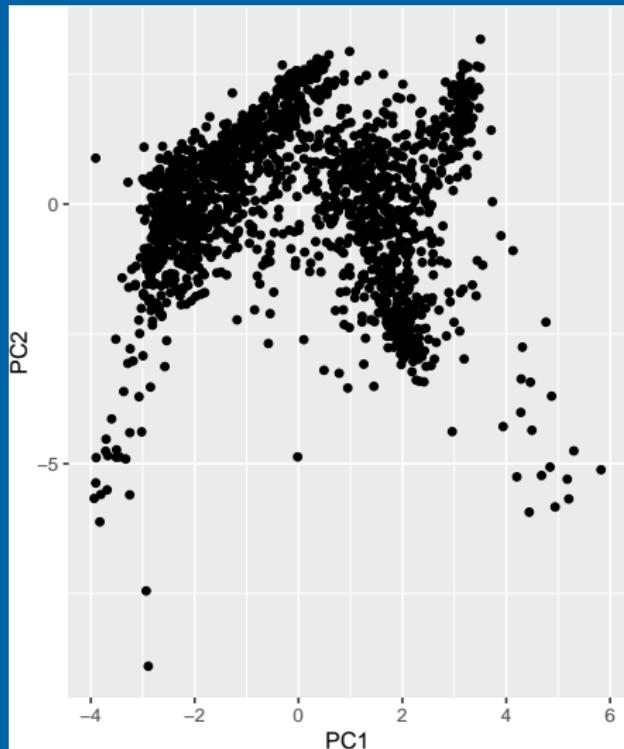
- **ACF1:** first order autocorrelation =  $\text{Corr}(Y_t, Y_{t-1})$
- Strength of **trend** and **seasonality** based on STL
- Size of seasonal **peak** and **trough**
- Spectral **entropy**
- **Lumpiness:** variance of block variances (block size 24).
- **Spikiness:** variances of leave-one-out variances of STL remainders.
- **Level shift:** Maximum difference in trimmed means of consecutive moving windows of size 24.
- **Variance change:** Max difference in variances of consecutive moving windows of size 24.
- **Flat spots:** Discretize sample space into 10 equal-sized intervals. Find max run length in any interval.
- Number of **crossing points** of mean line.
- **Kullback-Leibler score:** Maximum of  $D_{KL}(P\|Q) = \int P(x) \ln P(x)/Q(x)dx$  where  $P$  and  $Q$  are estimated by kernel density estimators applied to consecutive windows of size 48.
- **Change index:** Time of maximum KL score

# Feature space

```
library(tsfeatures)
library(tidyverse)
library(anomalous)
yahoo <- cbind(dat0, dat1, dat2, dat3)
hwl <- bind_cols(
    tsfeatures(yahoo,
        c("acf_features", "entropy", "lumpiness",
            "flat_spots", "crossing_points")),
    tsfeatures(yahoo, "stl_features",
        s.window='periodic', robust=TRUE),
    tsfeatures(yahoo, "max_kl_shift", width=48),
    tsfeatures(yahoo,
        c("mean", "var"), scale=FALSE, na.rm=TRUE),
    tsfeatures(yahoo,
        c("max_level_shift", "max_var_shift"), trim=TRUE)) %>%
select(mean, var, x_acf1, trend,
       seasonal_strength, peak, trough,
       entropy, lumpiness, spike, max_level_shift,
       max_var_shift, flat_spots, crossing_points,
       max_kl_shift, time_kl_shift)
```

# Feature space

```
pc <- prcomp(na.omit(hwl), scale=TRUE)$x %>% as_tibble()  
ggplot(pc, aes(x=PC1, y=PC2)) + geom_point()
```

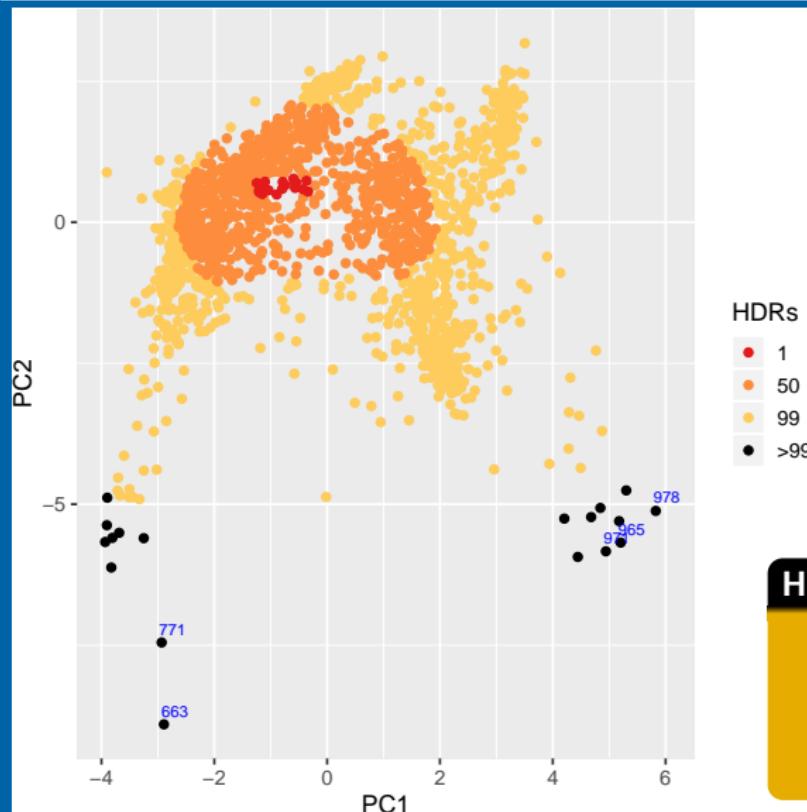


## What is “anomalous”?

- We need a measure of the “anomalousness” of a time series.
- Rank points based on their local density using a bivariate kernel density estimate.

# Finding weird time series

```
hdrcde::hdrscatterplot(pc[,1], pc[,2], noutliers=5)
```



HDRs

- 1
- 50
- 99
- >99

## Highest Density Regions

- Estimate using `hdrcde` package
- Highlight outlying points as those with lowest density.

# Stray algorithm



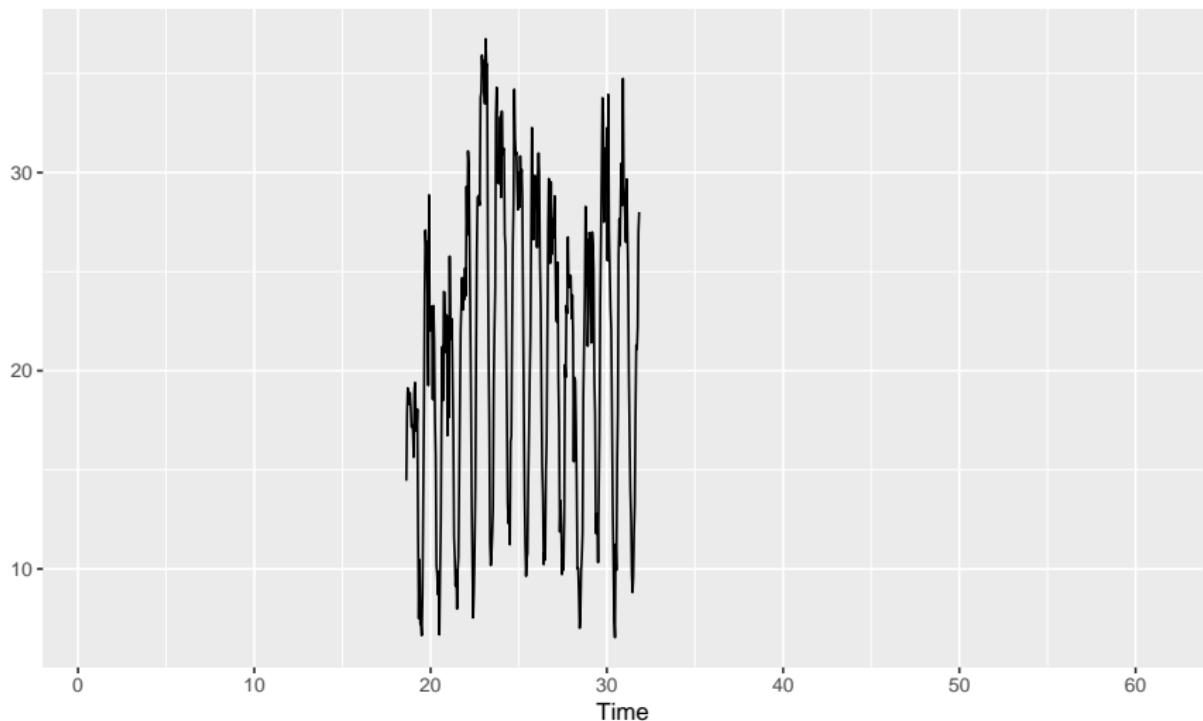
```
library(stray)  
find_HDoutliers(hwl)
```

```
## [1] 311 392 411 413 583 594 601 1151
```

- Uses extreme value theory applied to nearest neighbour distances between observations.
- Works directly on high-dimensional data (no need to do PCA).
- Modification of HDoutliers algorithm of Lee Wilkinson (HDoutliers package).

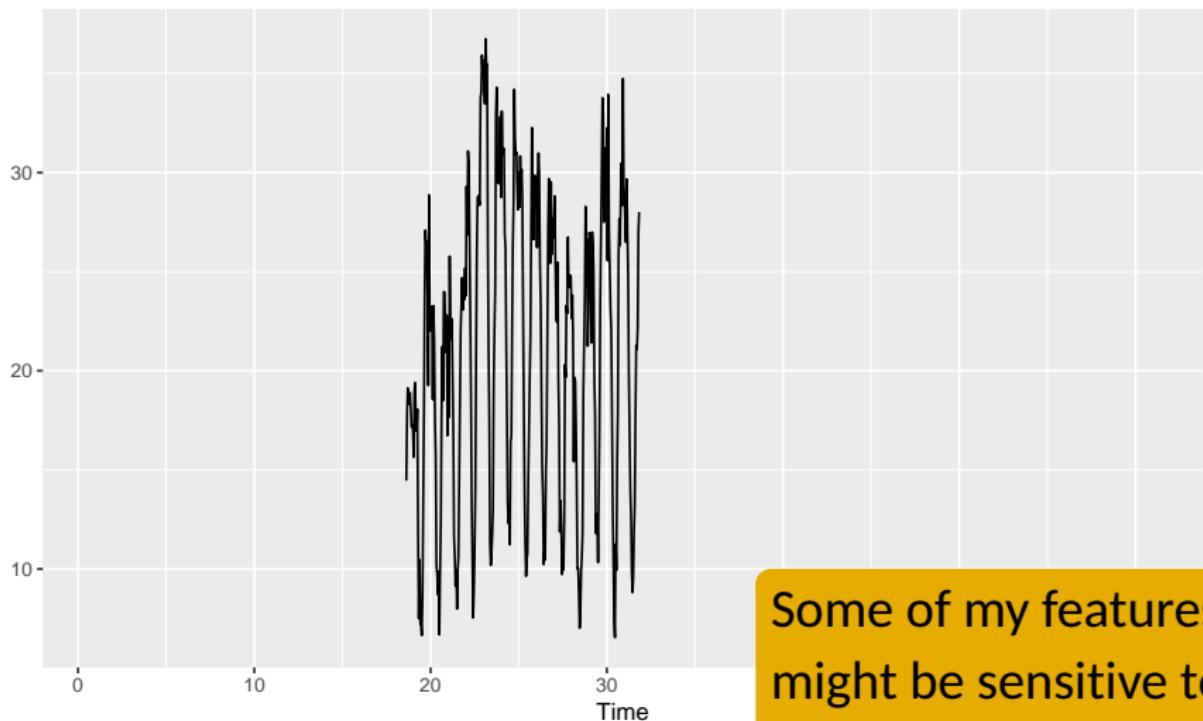
# Stray algorithm

Series 311



# Stray algorithm

Series 311



Some of my features  
might be sensitive to  
missing values.

# Outline

1 M3 forecasting competition

2 Yahoo email servers

3 Irish smart metres

4 Packages

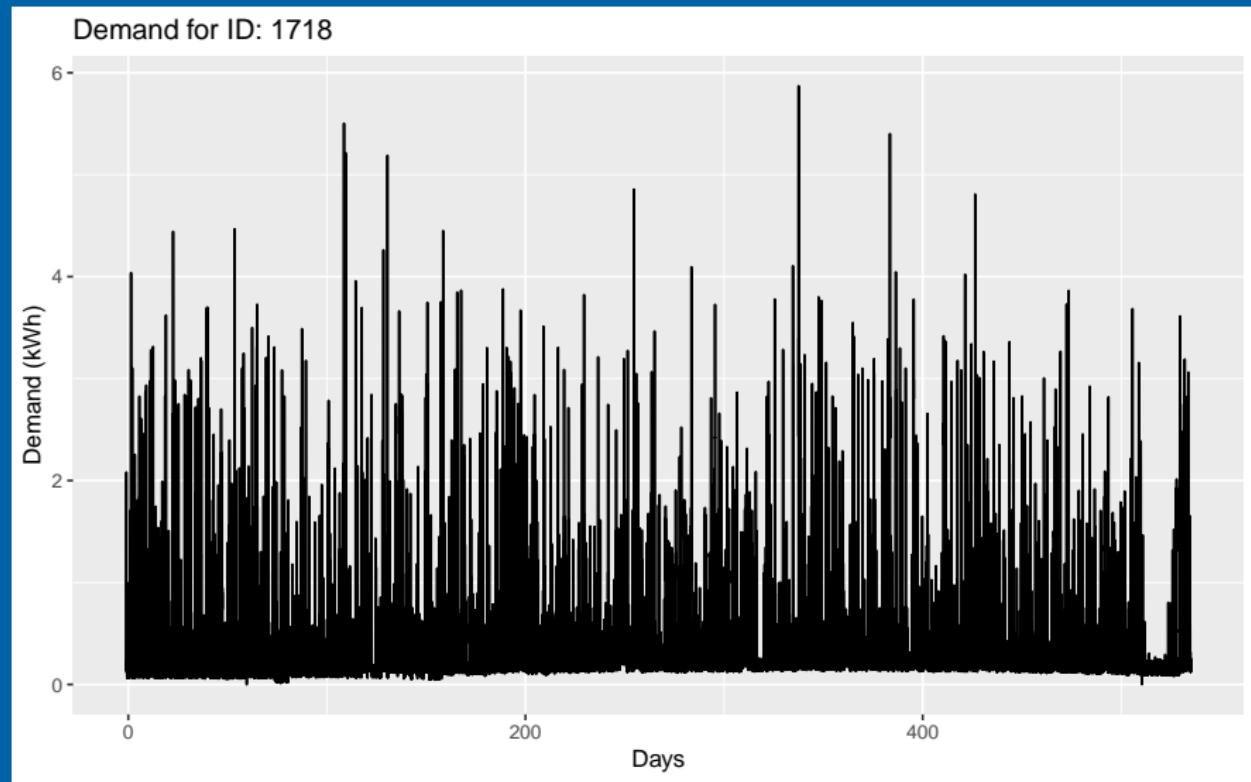
# Irish smart metre data



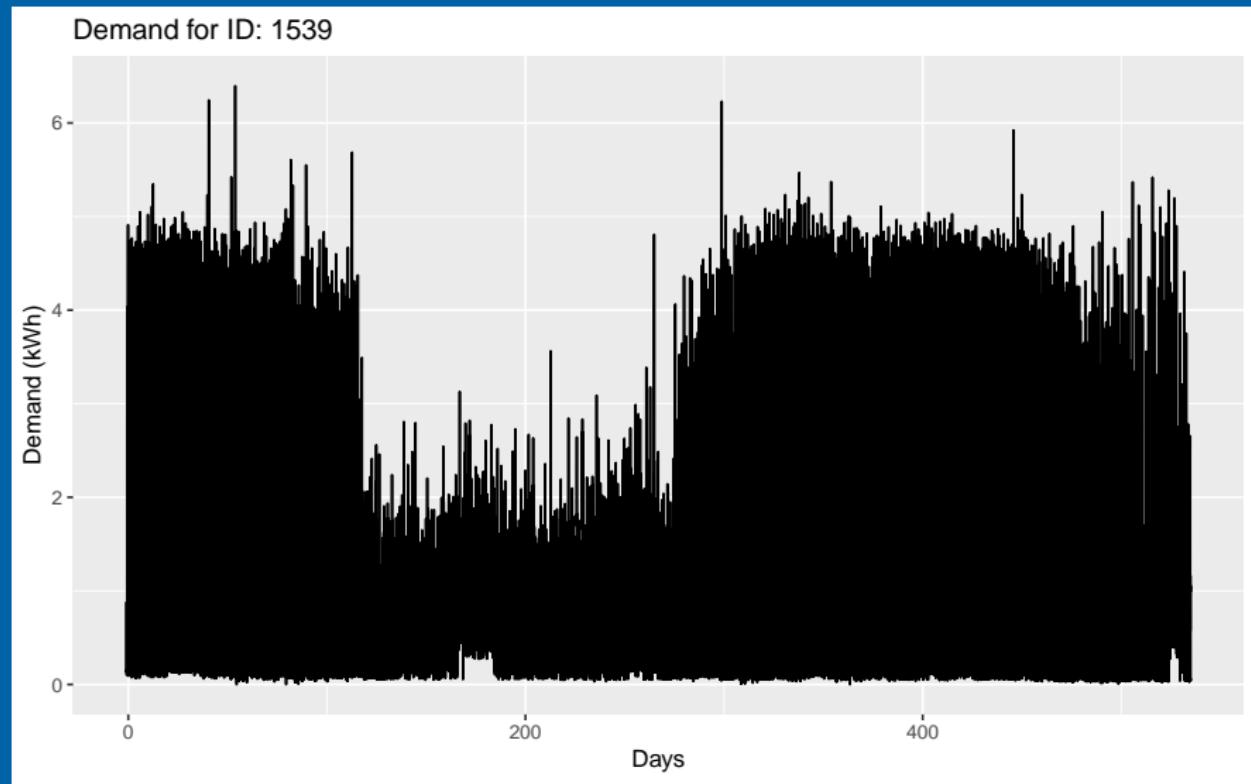
Figure: <http://solutions.3m.com>

- 500 households from smart metering trial
- Electricity consumption at 30-minute intervals between 14 July 2009 and 31 December 2010
- Heating/cooling energy usage excluded

# Irish smart metre data

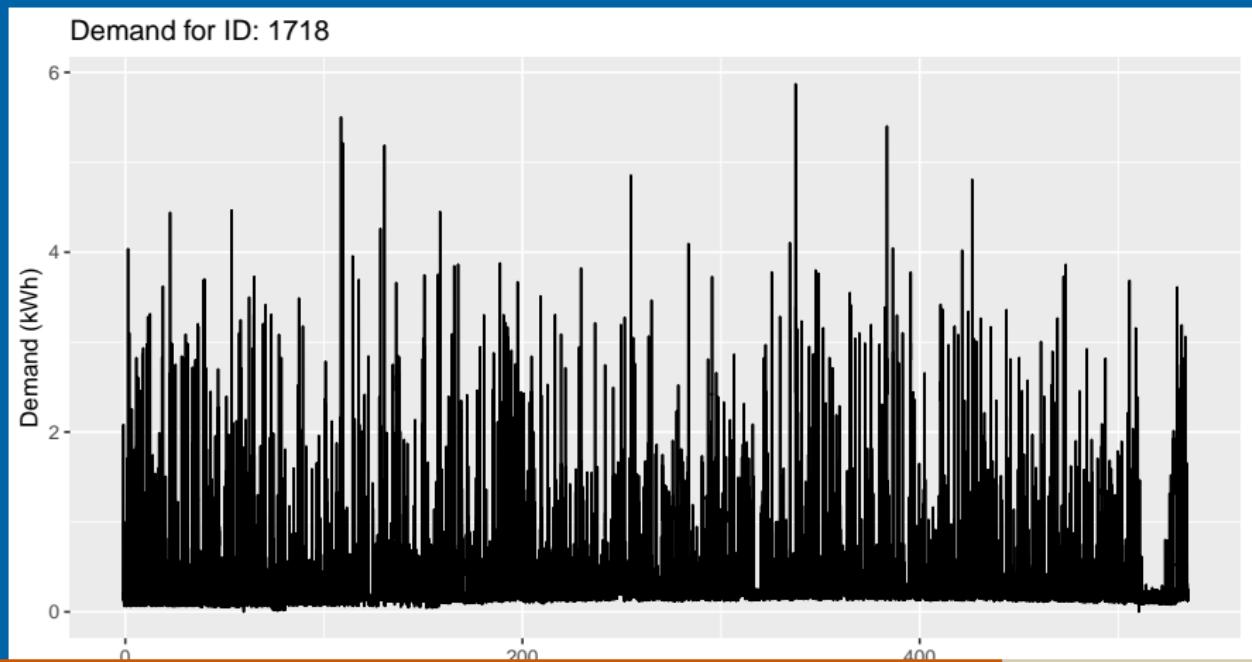


# Irish smart metre data



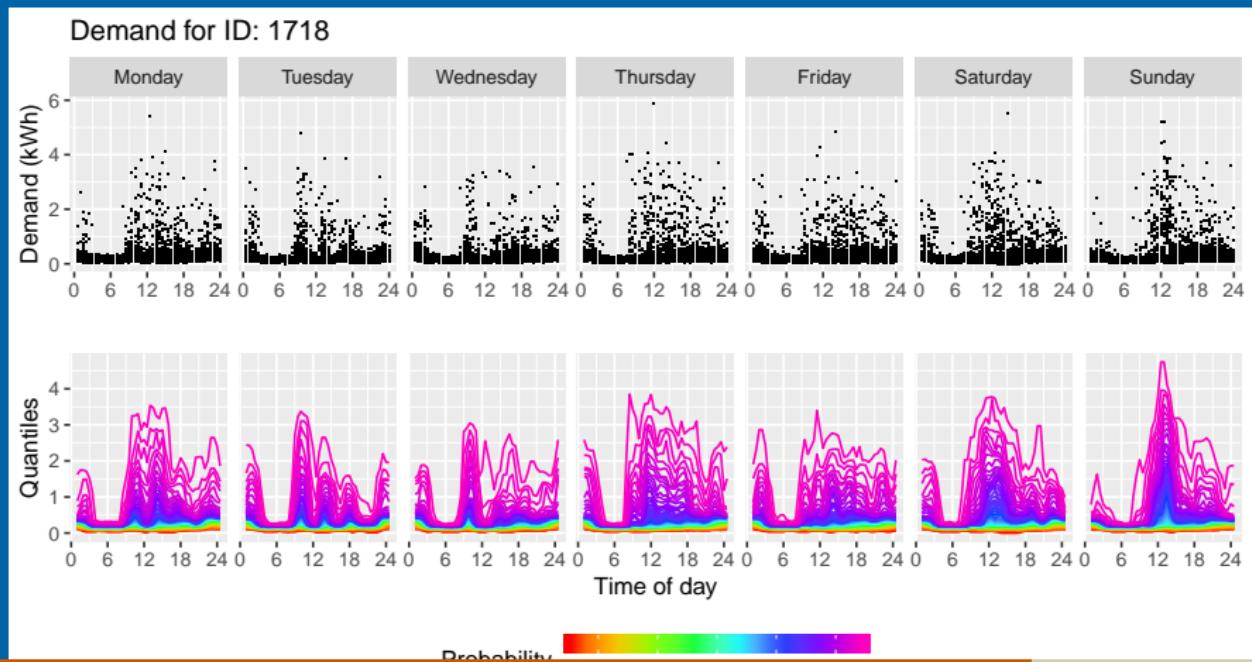
# Quantiles conditional on time of week

- Compute sample quantiles at  $p = 0.01, 0.02, \dots, 0.99$  for each household and each half-hour of the week.
- 336 probability distributions per household.



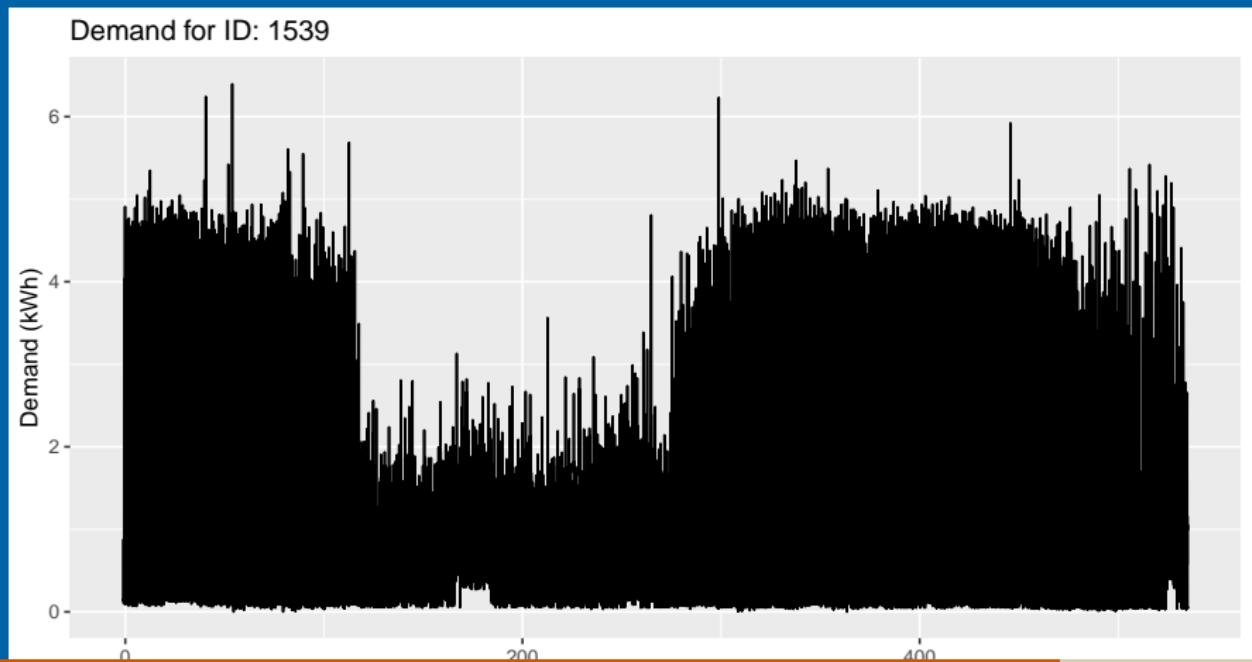
# Quantiles conditional on time of week

- Compute sample quantiles at  $p = 0.01, 0.02, \dots, 0.99$  for each household and each half-hour of the week.
- 336 probability distributions per household.



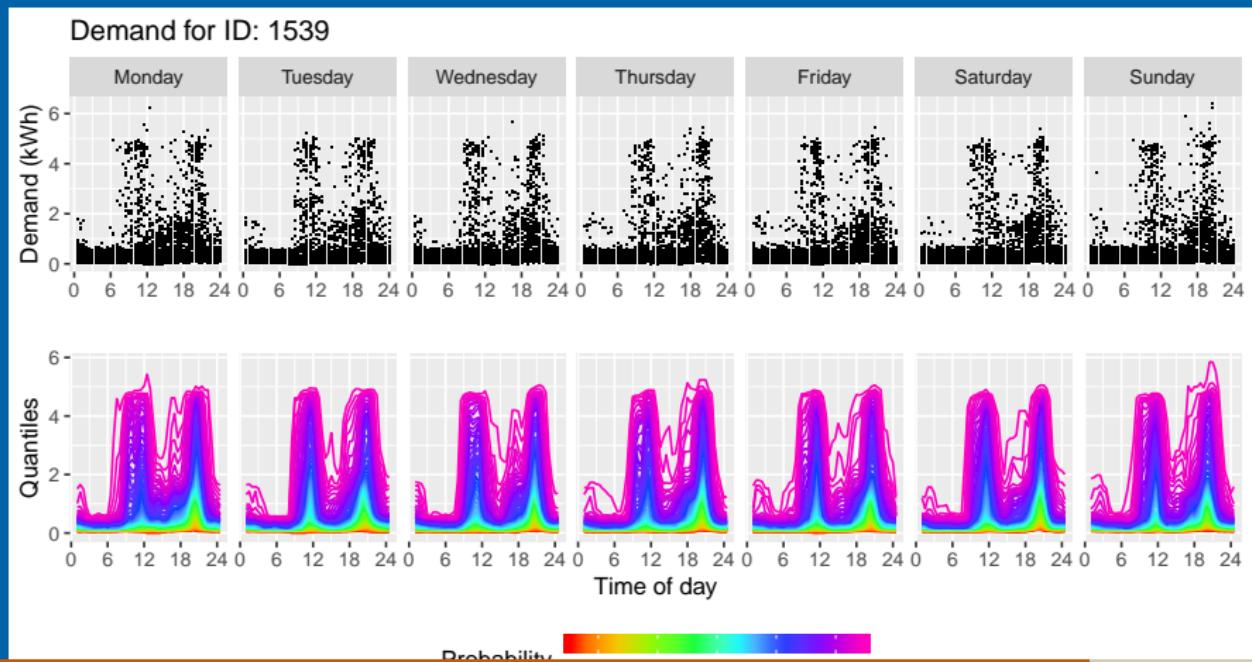
# Quantiles conditional on time of week

- Compute sample quantiles at  $p = 0.01, 0.02, \dots, 0.99$  for each household and each half-hour of the week.
- 336 probability distributions per household.

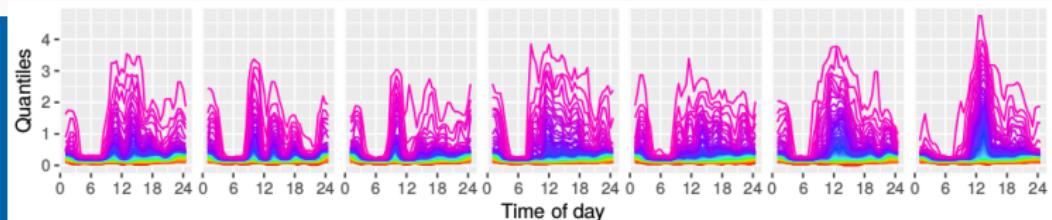


# Quantiles conditional on time of week

- Compute sample quantiles at  $p = 0.01, 0.02, \dots, 0.99$  for each household and each half-hour of the week.
- 336 probability distributions per household.

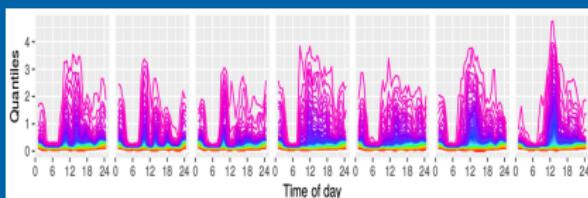
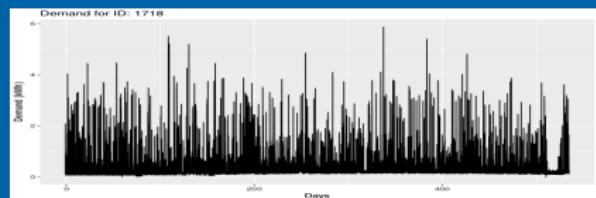


# Quantiles conditional on time of week

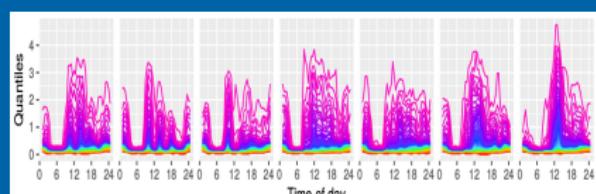


- Sample quantiles better than kernel density estimate:
  - ▶ presence of zeros
  - ▶ non-negative support
  - ▶ high skewness
- Avoids missing data issues and variation in series length
- Avoids timing of household events, holidays, etc.
- Allows clustering of households based on probabilistic behaviour rather than coincident behaviour.
- Allows identification of anomalous households.
- Allows estimation of typical household behaviour.

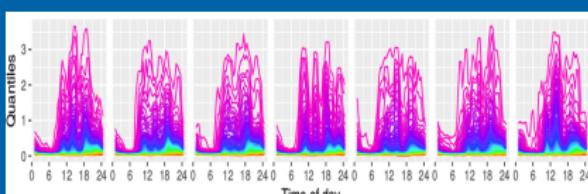
# Pairwise distances



- The time series of  $535 \times 48$  observations per household is mapped to a set of  $7 \times 48 \times 99$  quantiles giving a bivariate surface for each household.
- Can we compute pairwise distances between all households?



← ? →  
Distance



# Pairwise distances

## Jensen-Shannon distance between two households

$$\Delta_{ij} = \sum_{t=1}^{7 \times 48} JS(p_t, q_t)$$

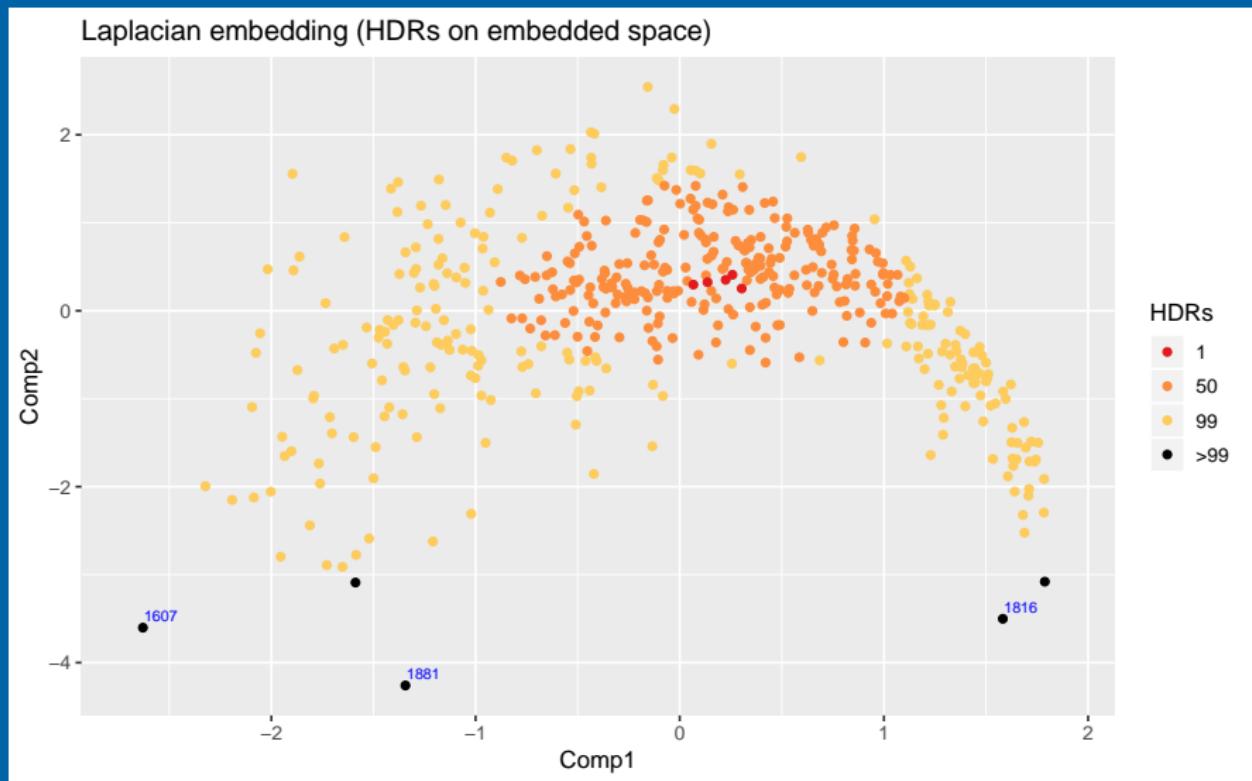
## Similarity between two households

$$w_{ij} = \exp(-\Delta_{ij}^2/h^2).$$

## Laplacian eigenmaps

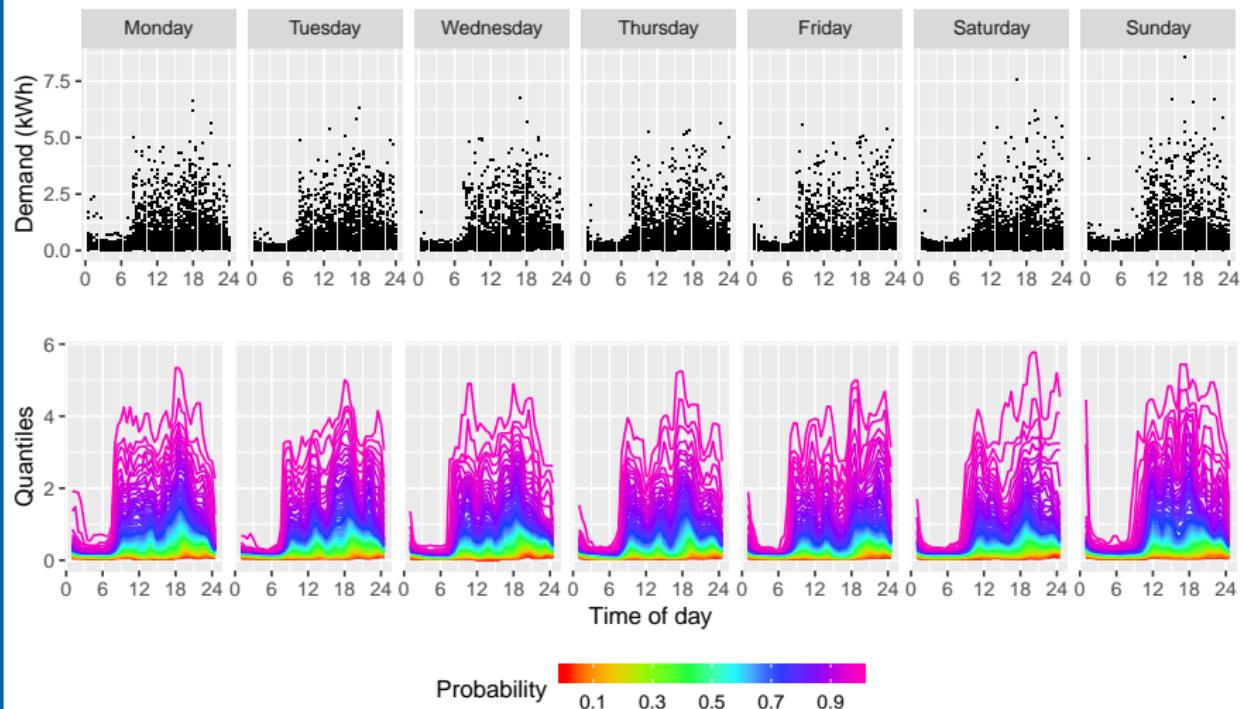
- Laplacian eigenmap maps high-dimensional space to 2d space while preserving smallest distances, but not largest distances.
- We can use this to view the distances between the conditional densities of households

# Outliers computed in embedded space:



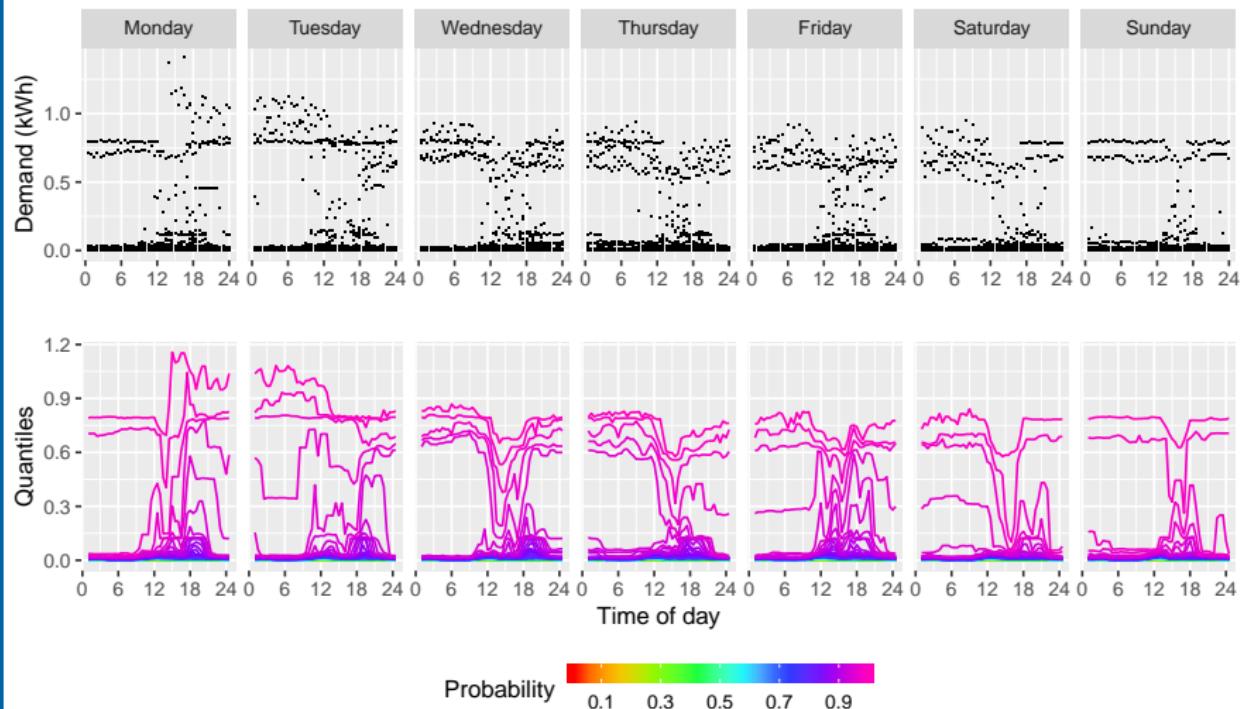
# Most typical household

Demand for ID: 1672



# Most anomalous household

Demand for ID: 1881



# Outline

1 M3 forecasting competition

2 Yahoo email servers

3 Irish smart metres

4 Packages

# Packages

- **tsfeatures**: compute time series features.  
[github.com/robjhyndman/tsfeatures](https://github.com/robjhyndman/tsfeatures)
- **hdrcde**: scatterplots with bivariate HDRs.  
[CRAN](https://CRAN.R-project.org/package=hdrcde) | [github.com/robjhyndman/hdrcde](https://github.com/robjhyndman/hdrcde)
- **stray**: finding outliers in high dimensions.  
[github.com/pridiltal/stray](https://github.com/pridiltal/stray)
- **oddstream**: finding outliers in streaming data.  
[github.com/pridiltal/oddstream](https://github.com/pridiltal/oddstream)
- **Mcomp**: M3 data.  
[CRAN](https://CRAN.R-project.org/package=Mcomp) | [github.com/robjhyndman/Mcomp](https://github.com/robjhyndman/Mcomp)
- **anomalous**: yahoo data.  
[github.com/robjhyndman/anomalous](https://github.com/robjhyndman/anomalous)