MONASH University

MONASH BUSINESS SCHOOL

Feature-based time series analysis

Rob J Hyndman

21 June 2018

Outline

1 M3 forecasting competition

2 Yahoo server metrics

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

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.

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

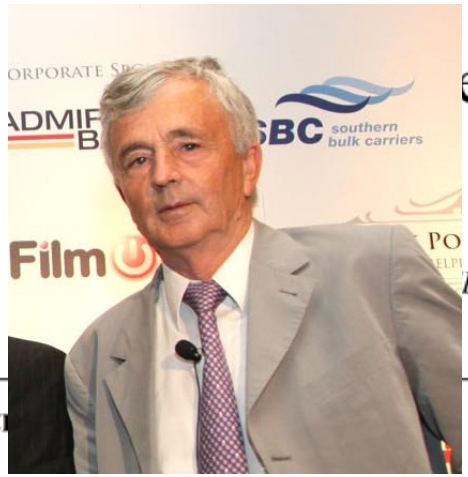
M3 competition

ELSEVIER

International Journal of Forecasting 16 (2000) 451–476

international journal of forecasting

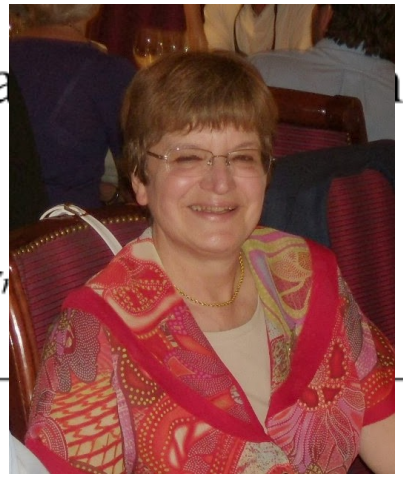
www.elsevier.com/locate/ijforecast



etition: results, conclusions a ns

Spyros Makridakis, Michèle Hibon*

EAD, Boulevard de Constance, 77305 Fontainebleau, Fr

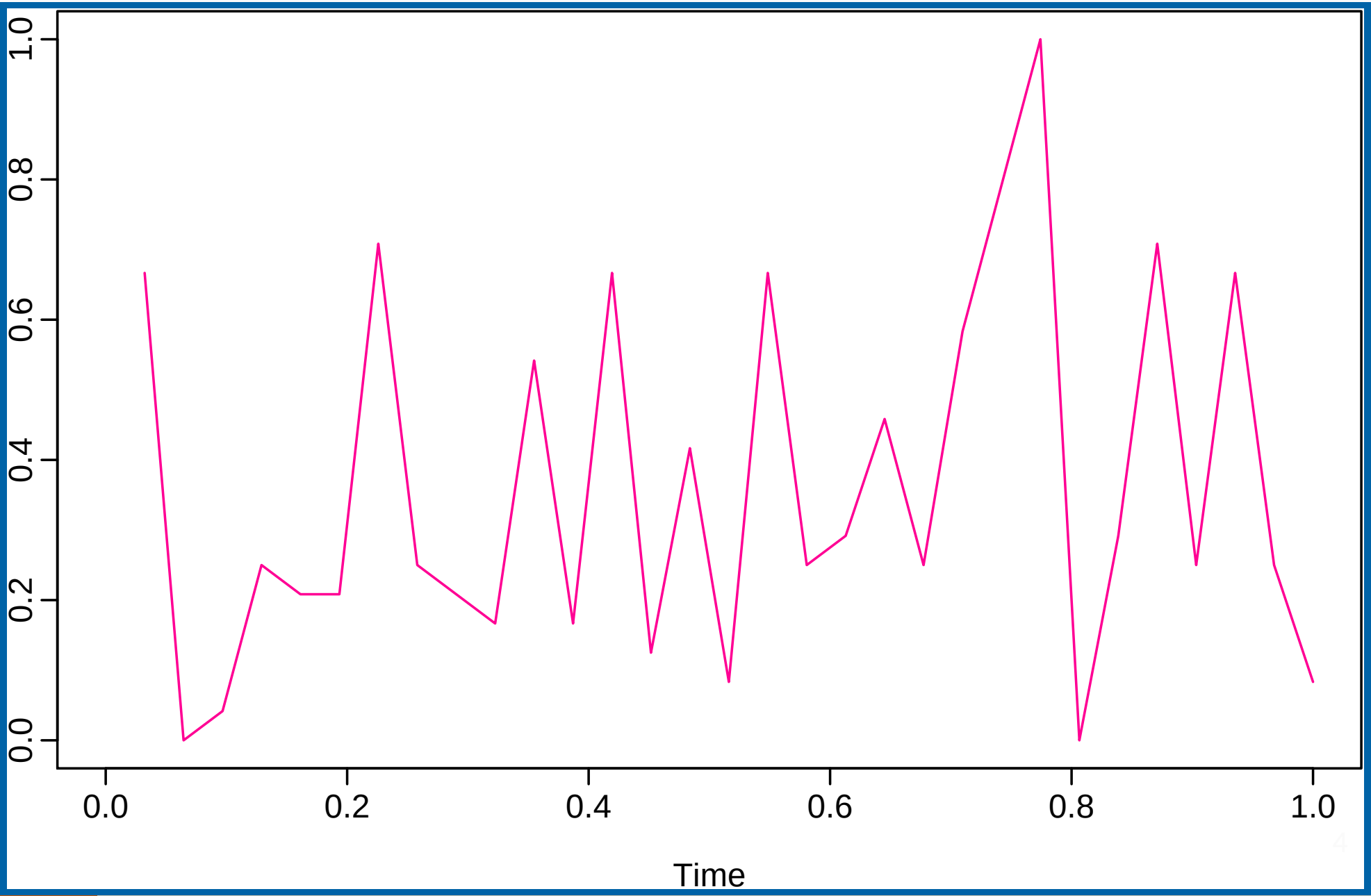


Abstr

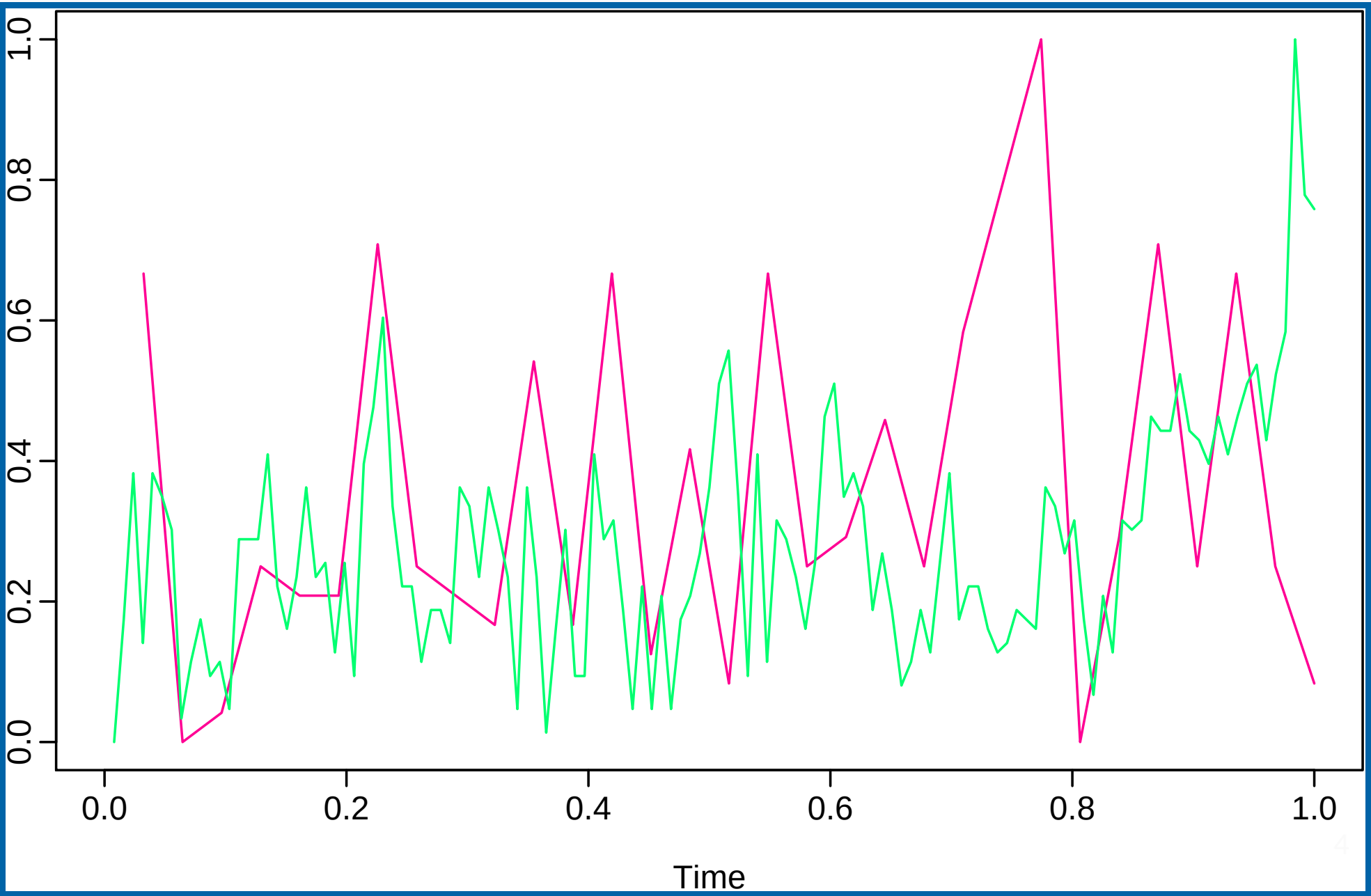
This paper describes the M3-Competition, the latest of the M-Competitions. It explains the reasons for conducting the competition and summarizes its results and conclusions. In addition, the paper compares such results/conclusions with those of the previous two M-Competitions as well as with those of 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.

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

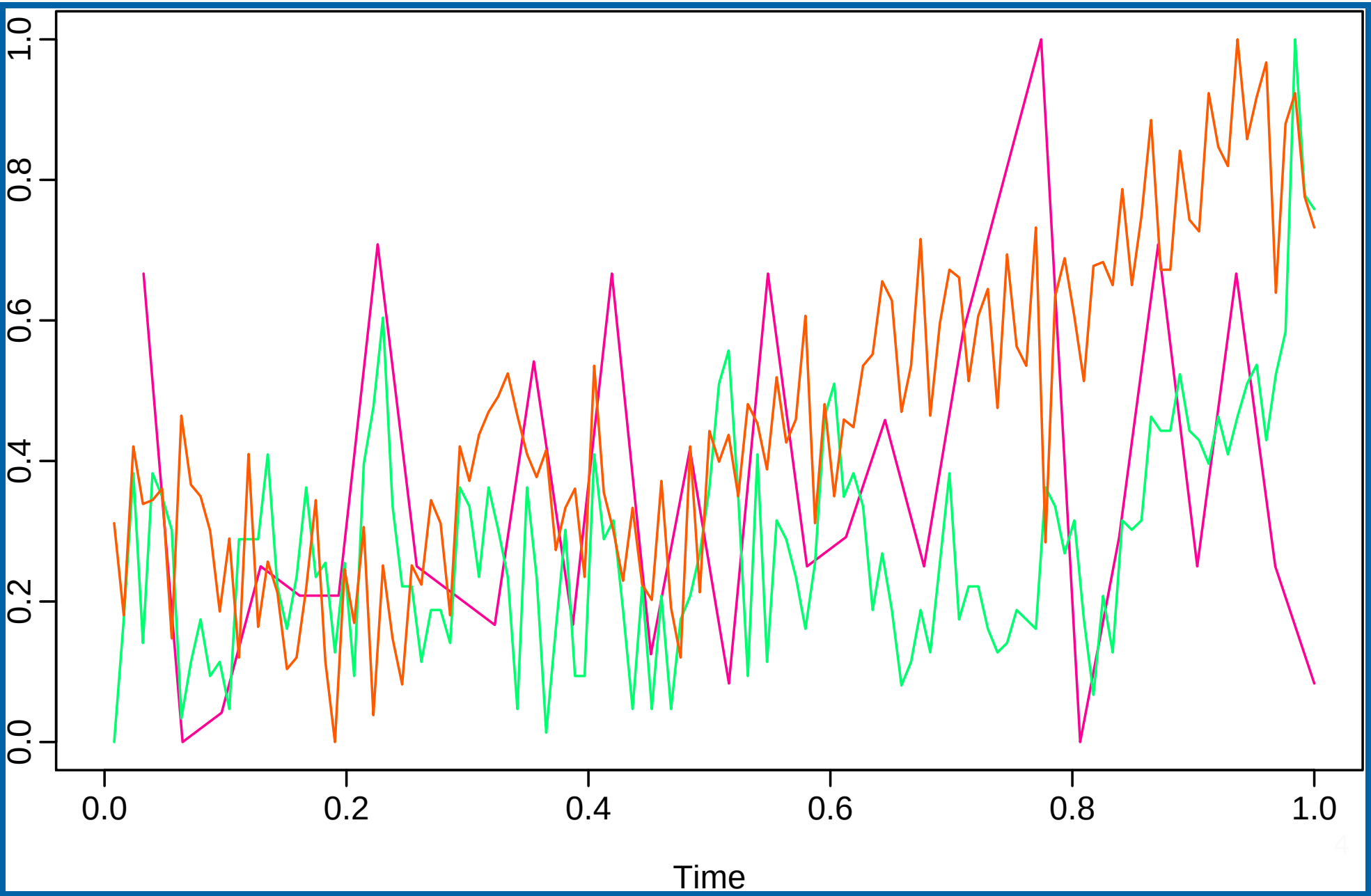
How to plot lots of time series?



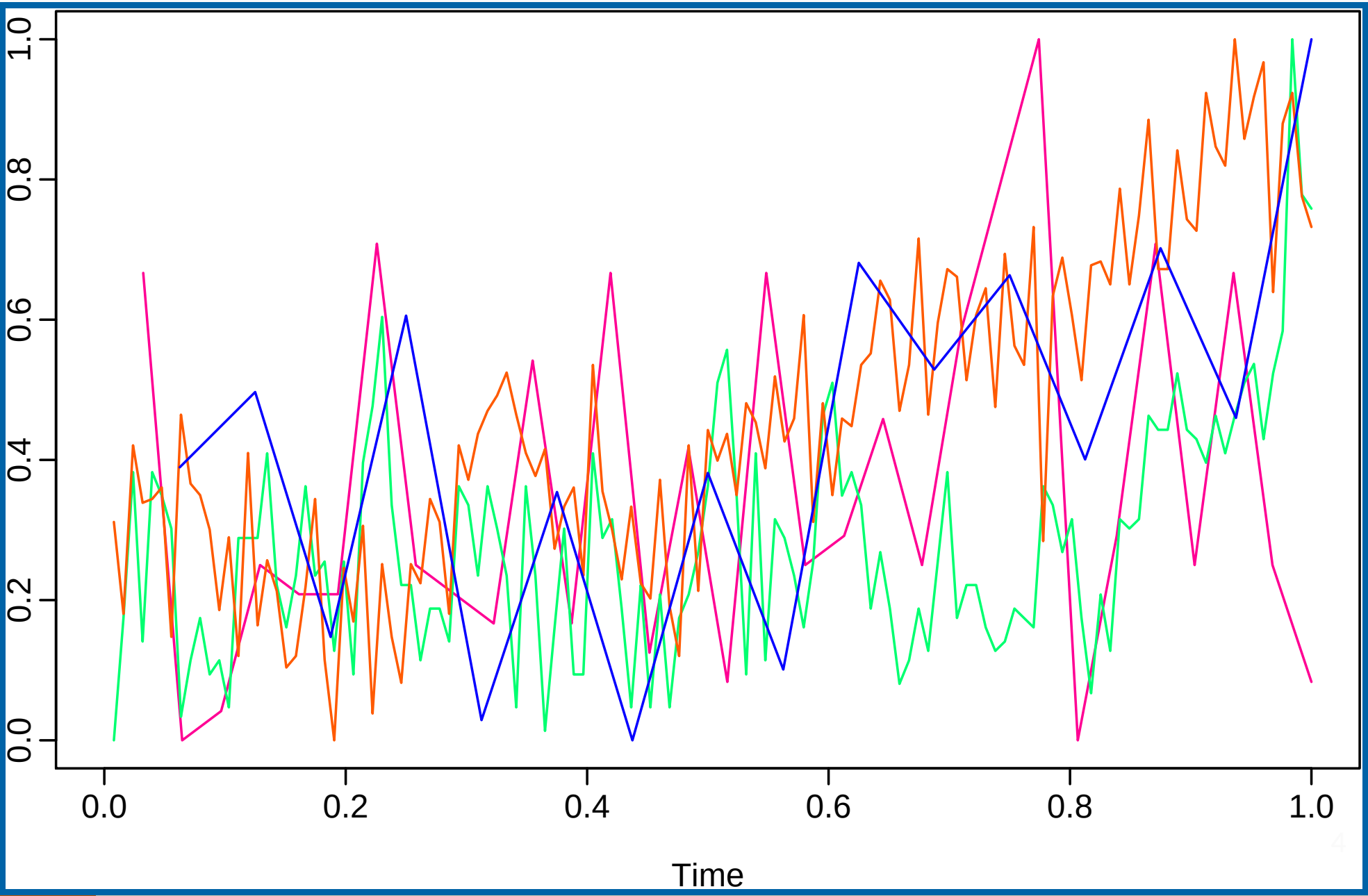
How to plot lots of time series?



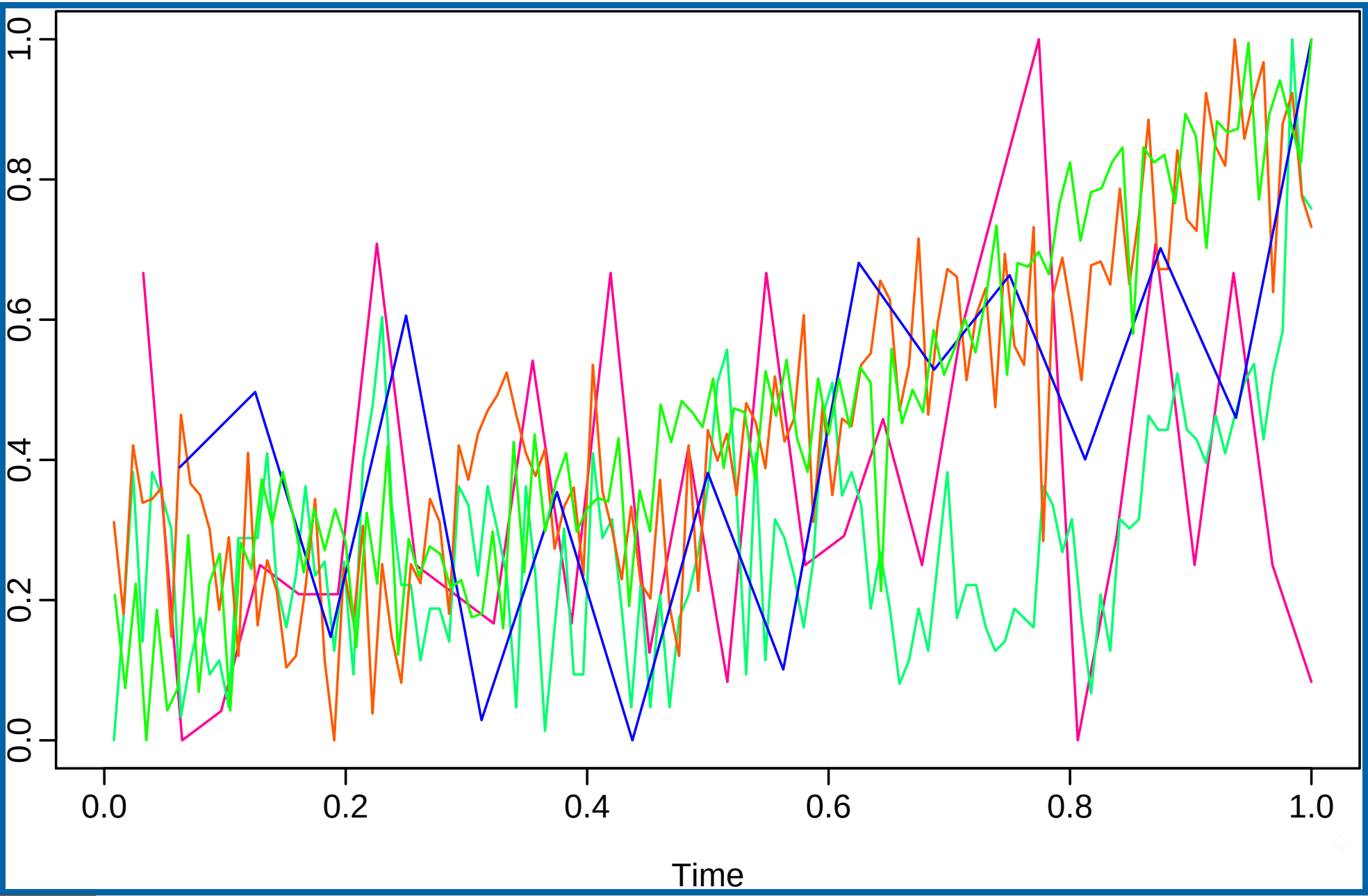
How to plot lots of time series?



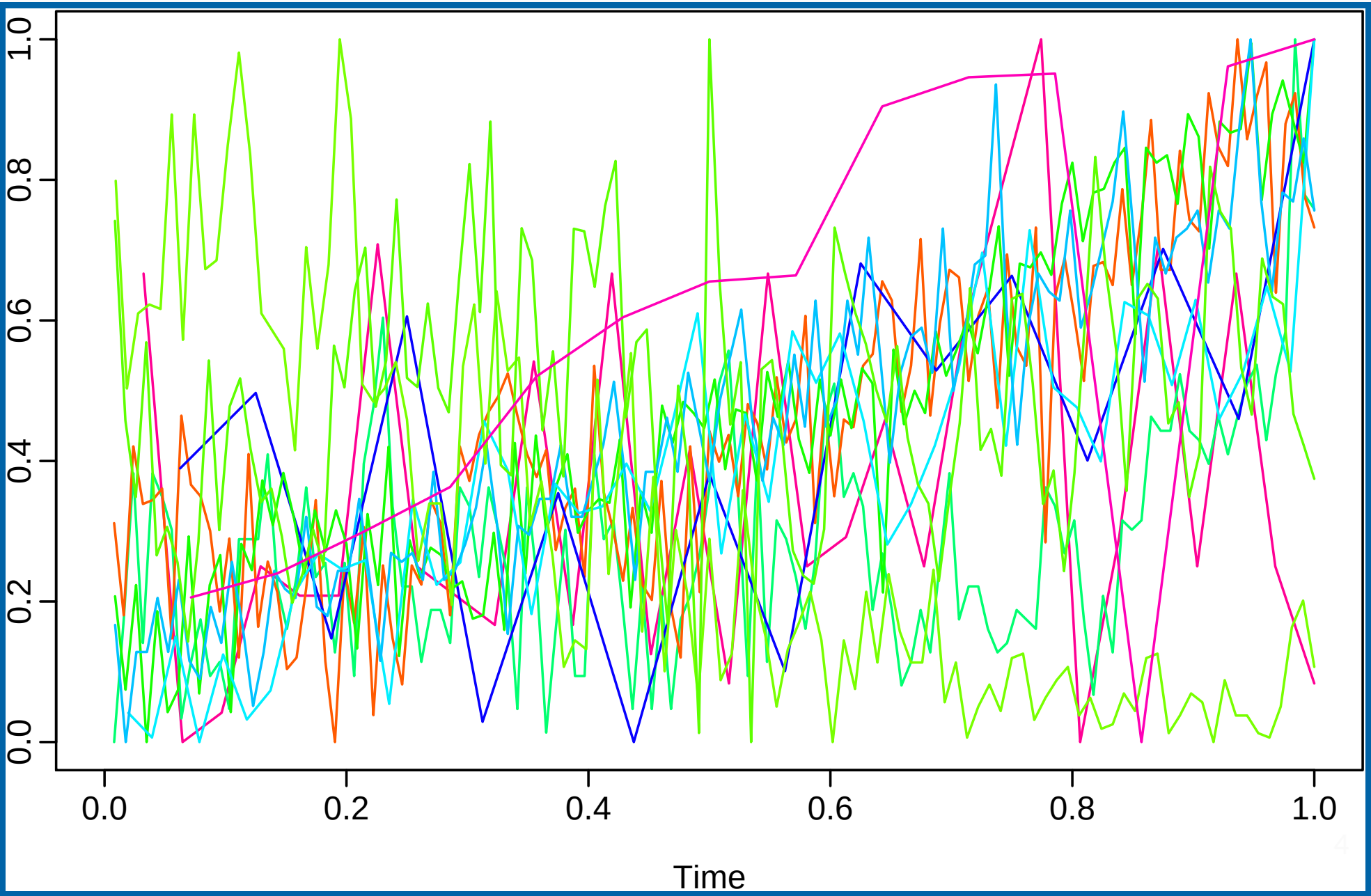
How to plot lots of time series?



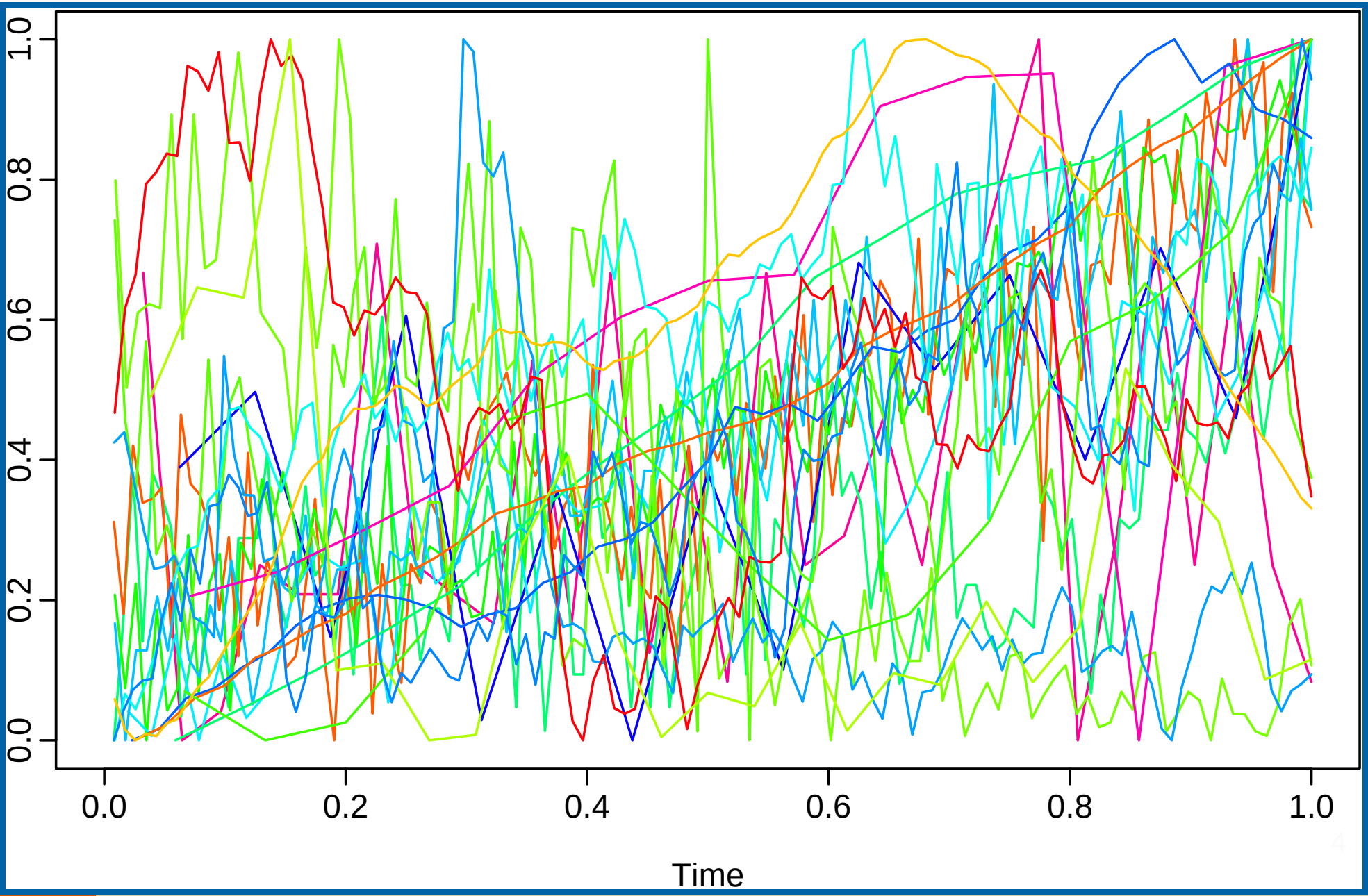
How to plot lots of time series?



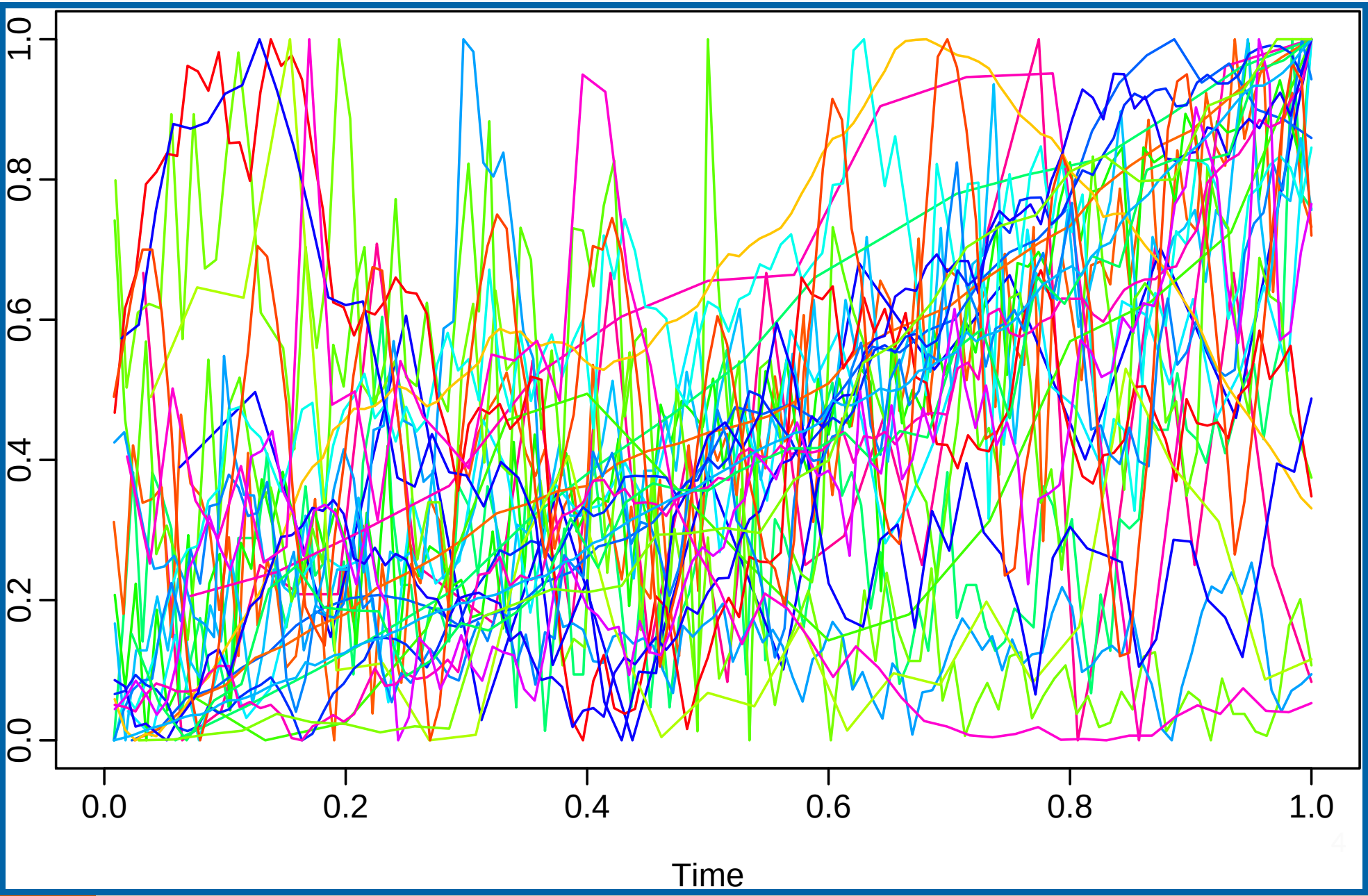
How to plot lots of time series?



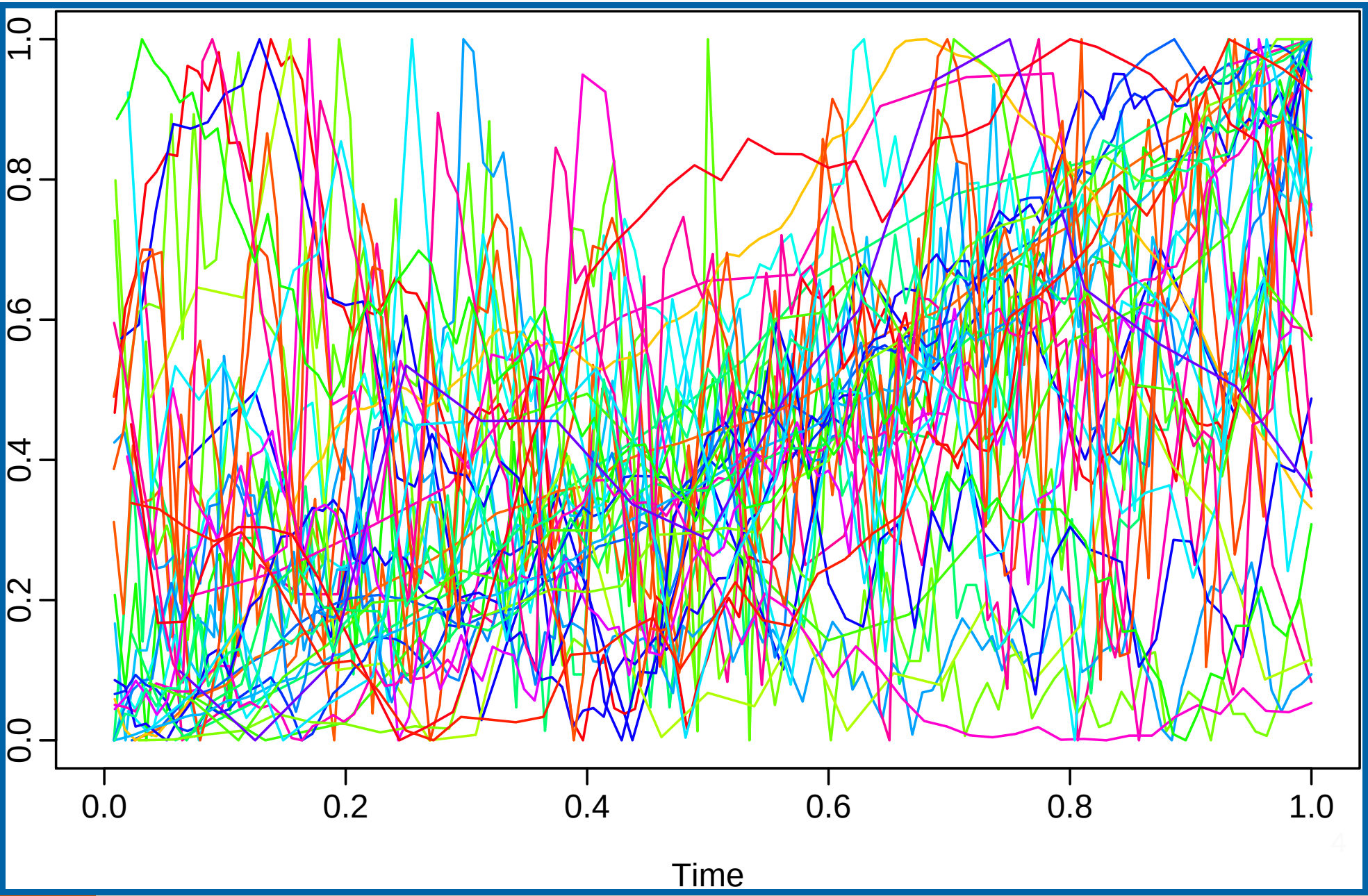
How to plot lots of time series?



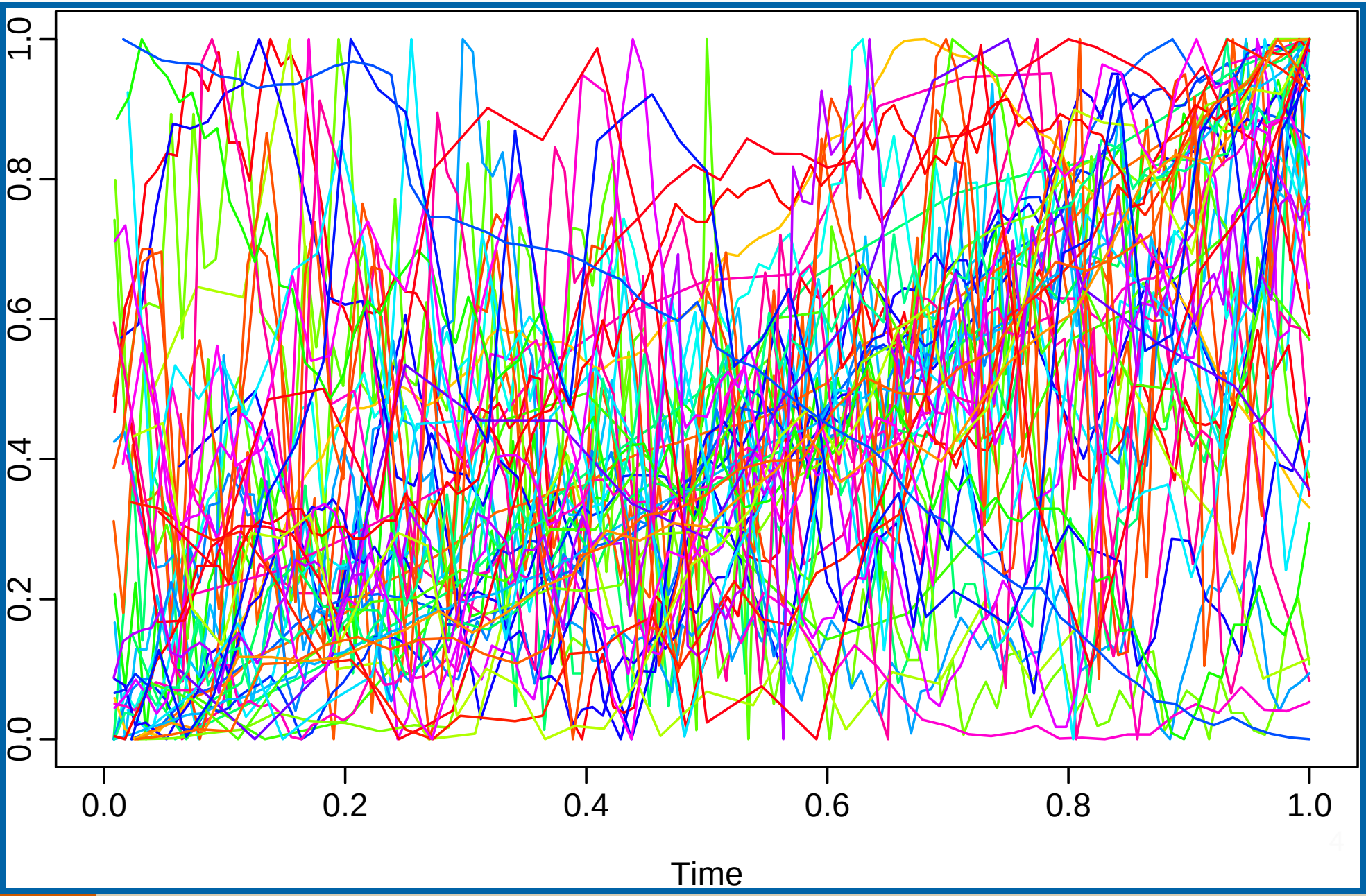
How to plot lots of time series?



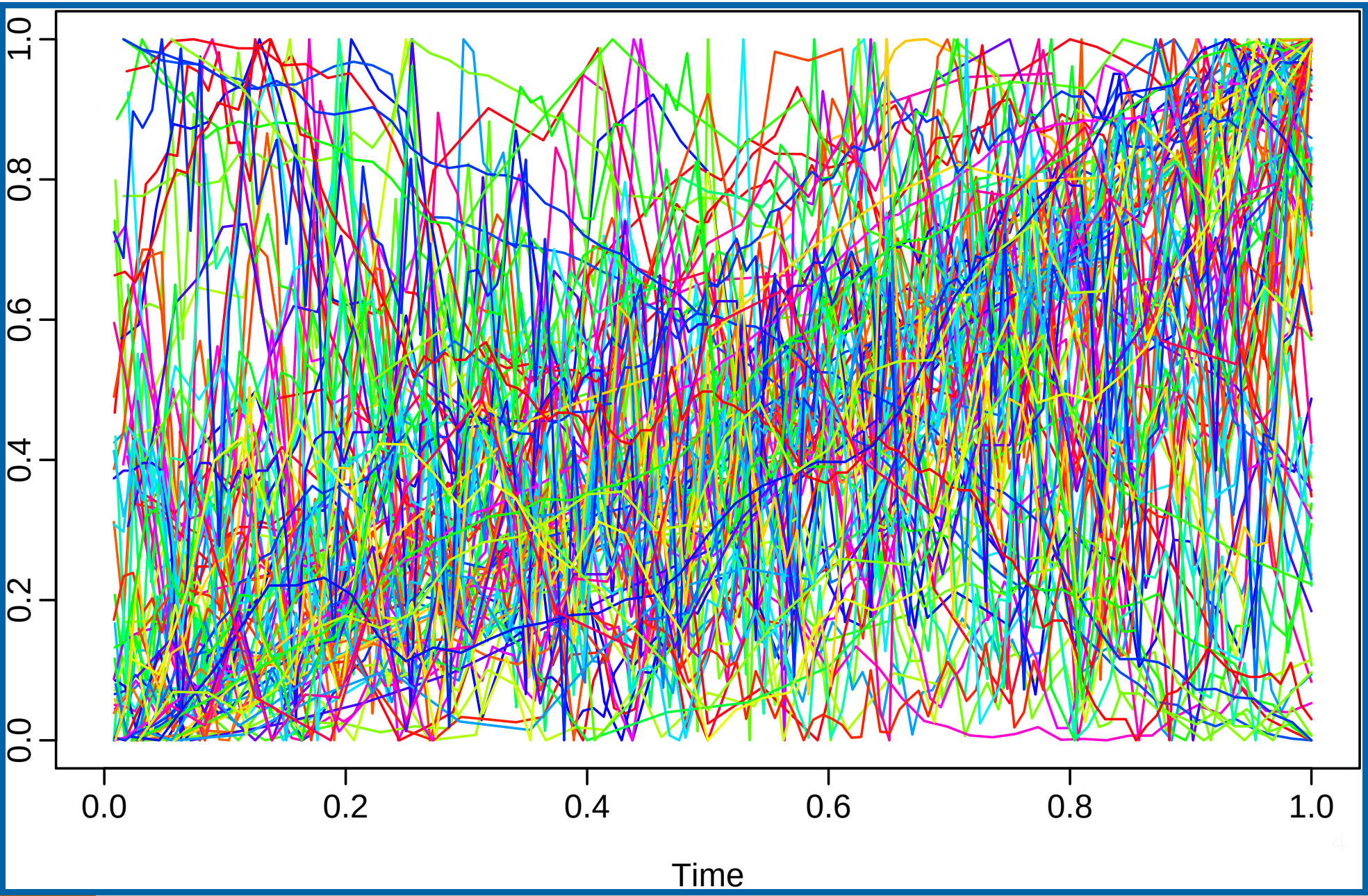
How to plot lots of time series?



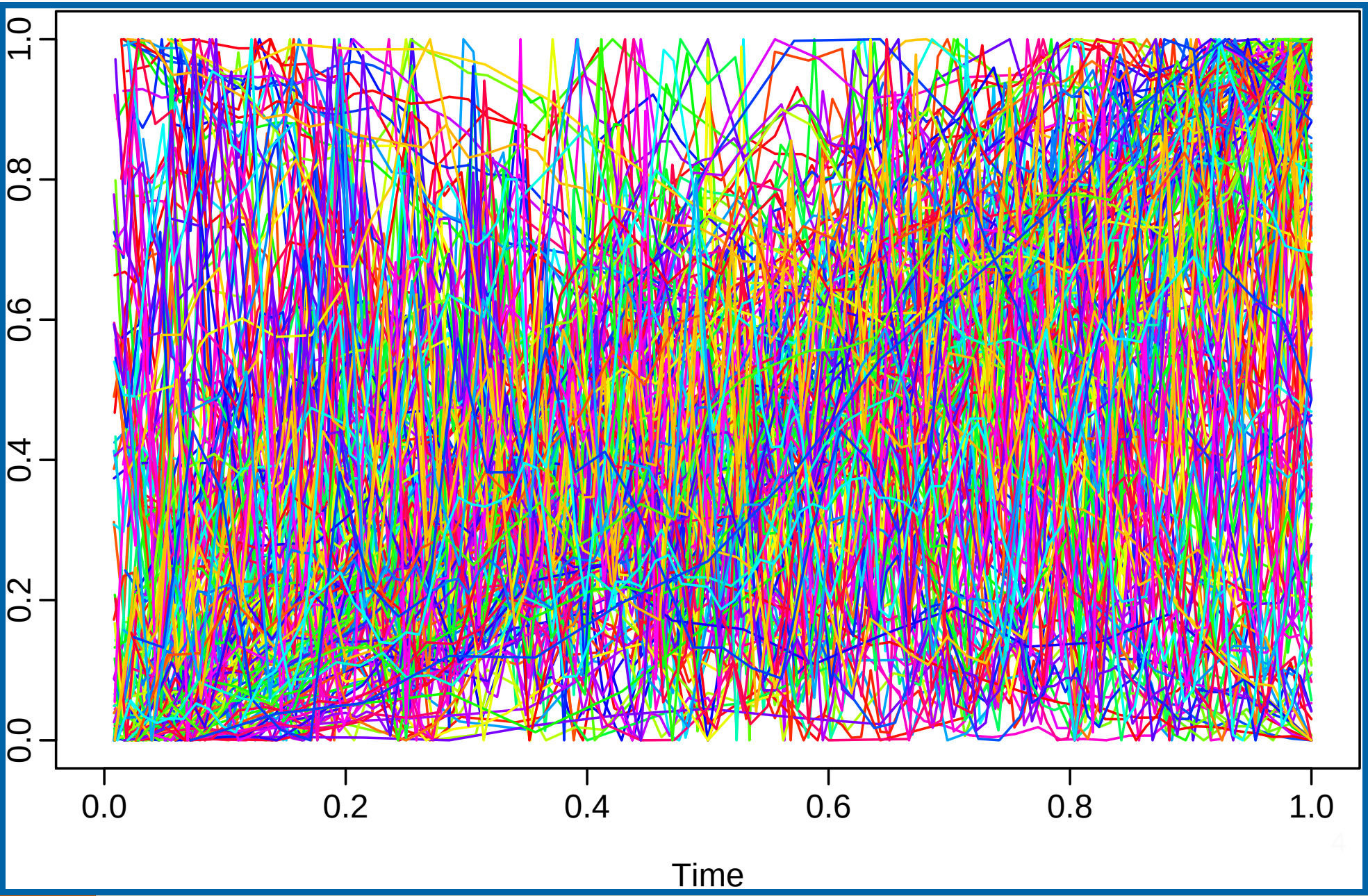
How to plot lots of time series?



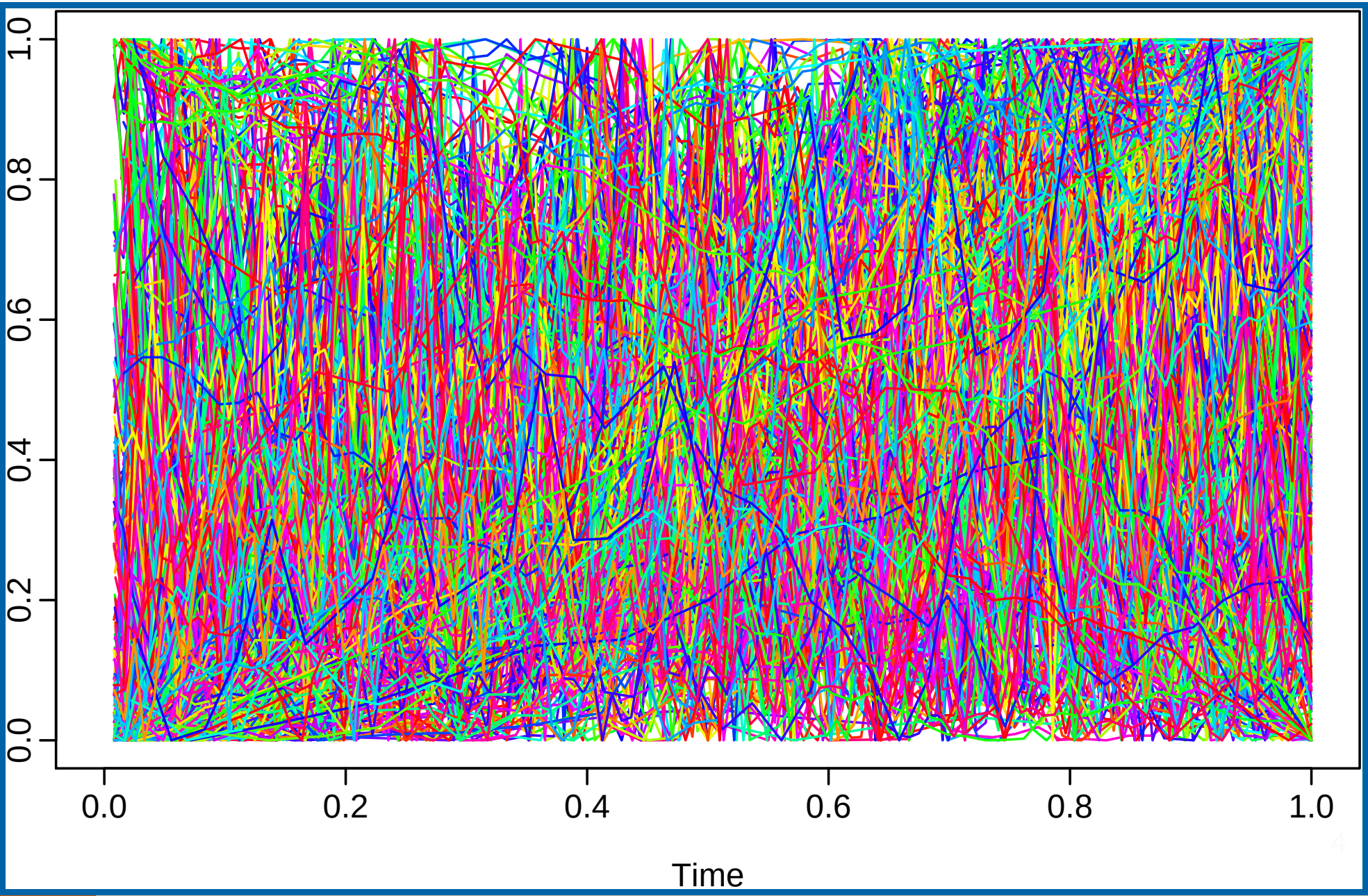
How to plot lots of time series?



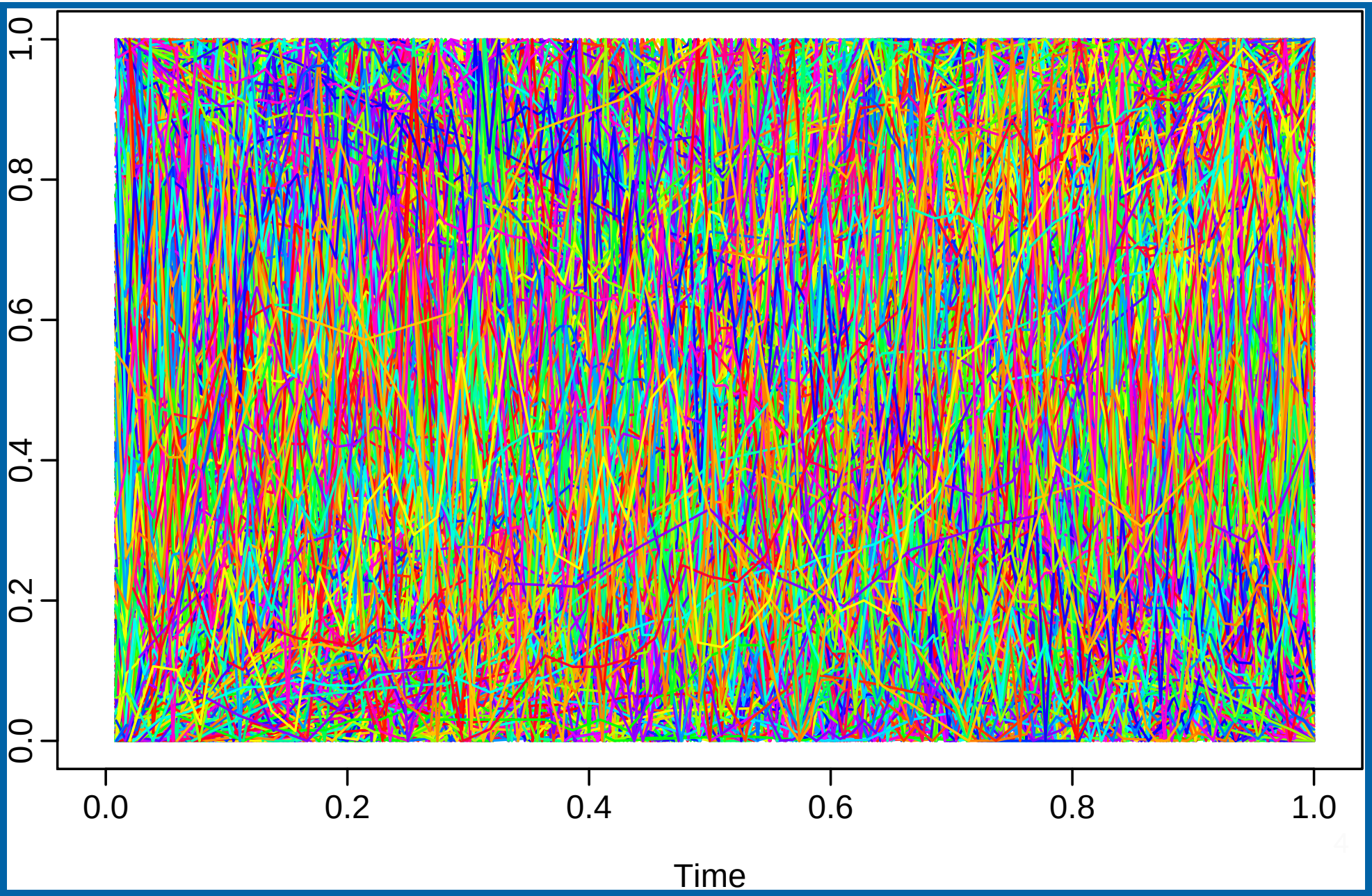
How to plot lots of time series?



How to plot lots of time series?



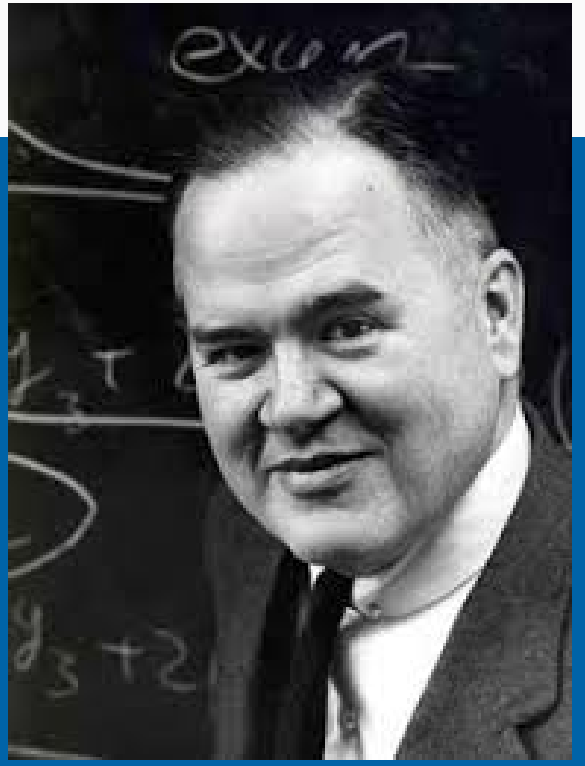
How to plot lots of time series?



Key idea

Cognostics

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

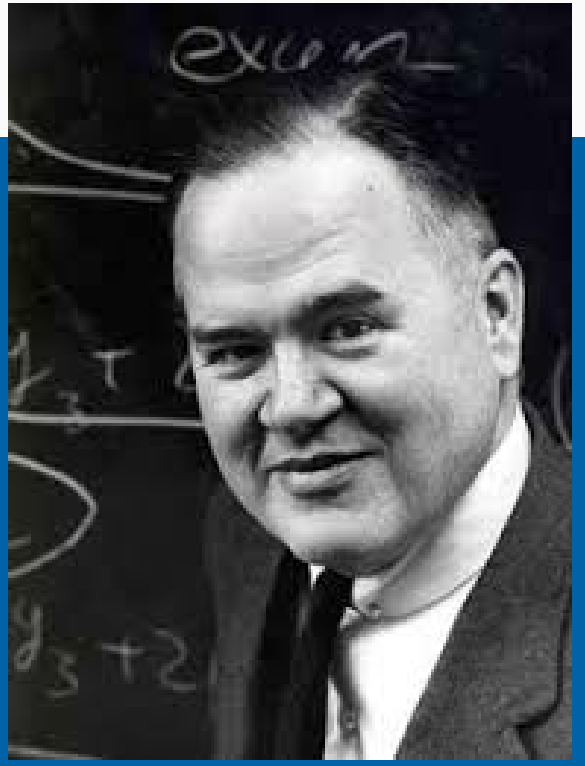


John W Tukey

Key idea

Cognostics

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



John W Tukey

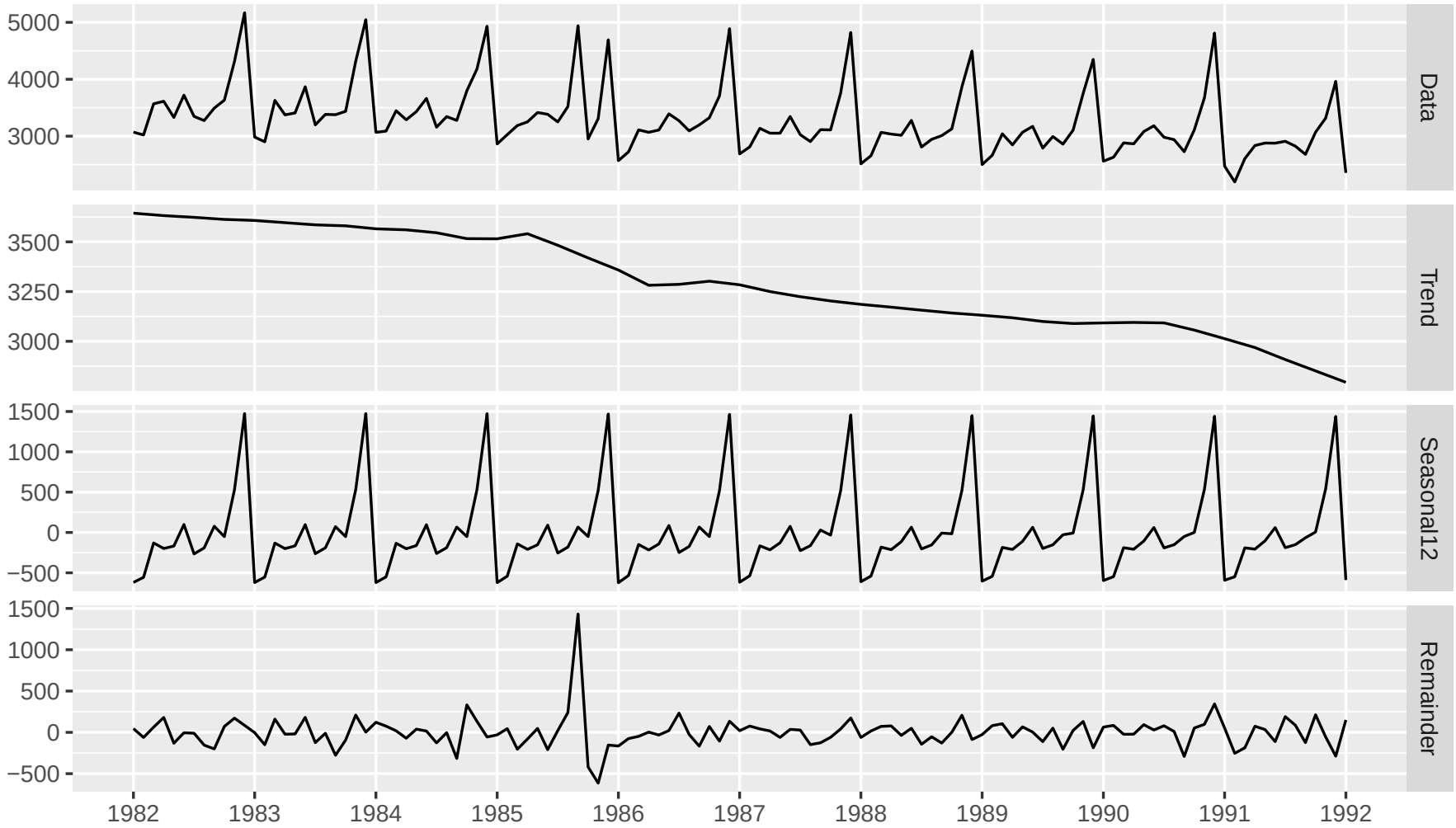
Examples for time series

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

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

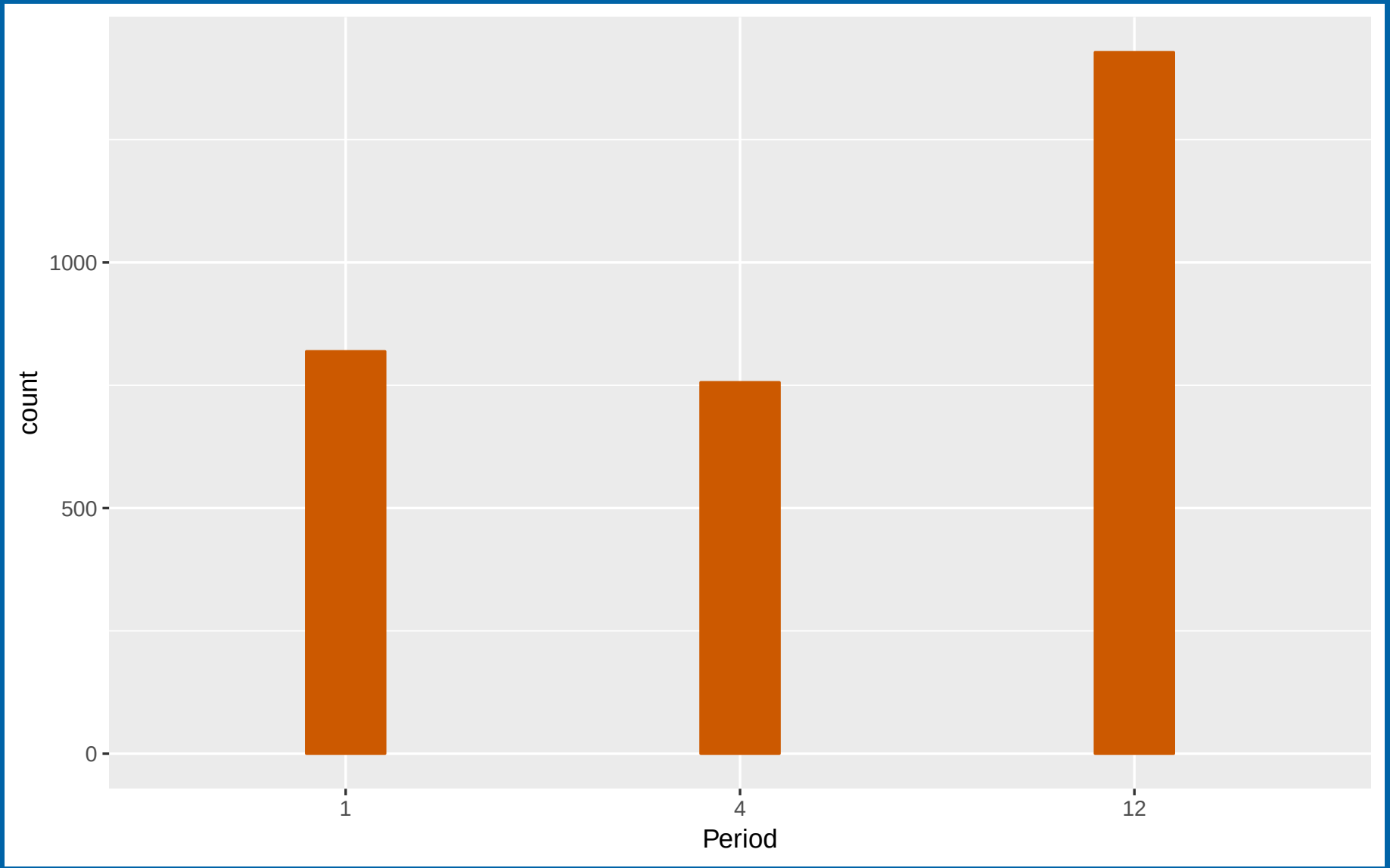
$$Y_t = S_t + T_t + R_t$$

- 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

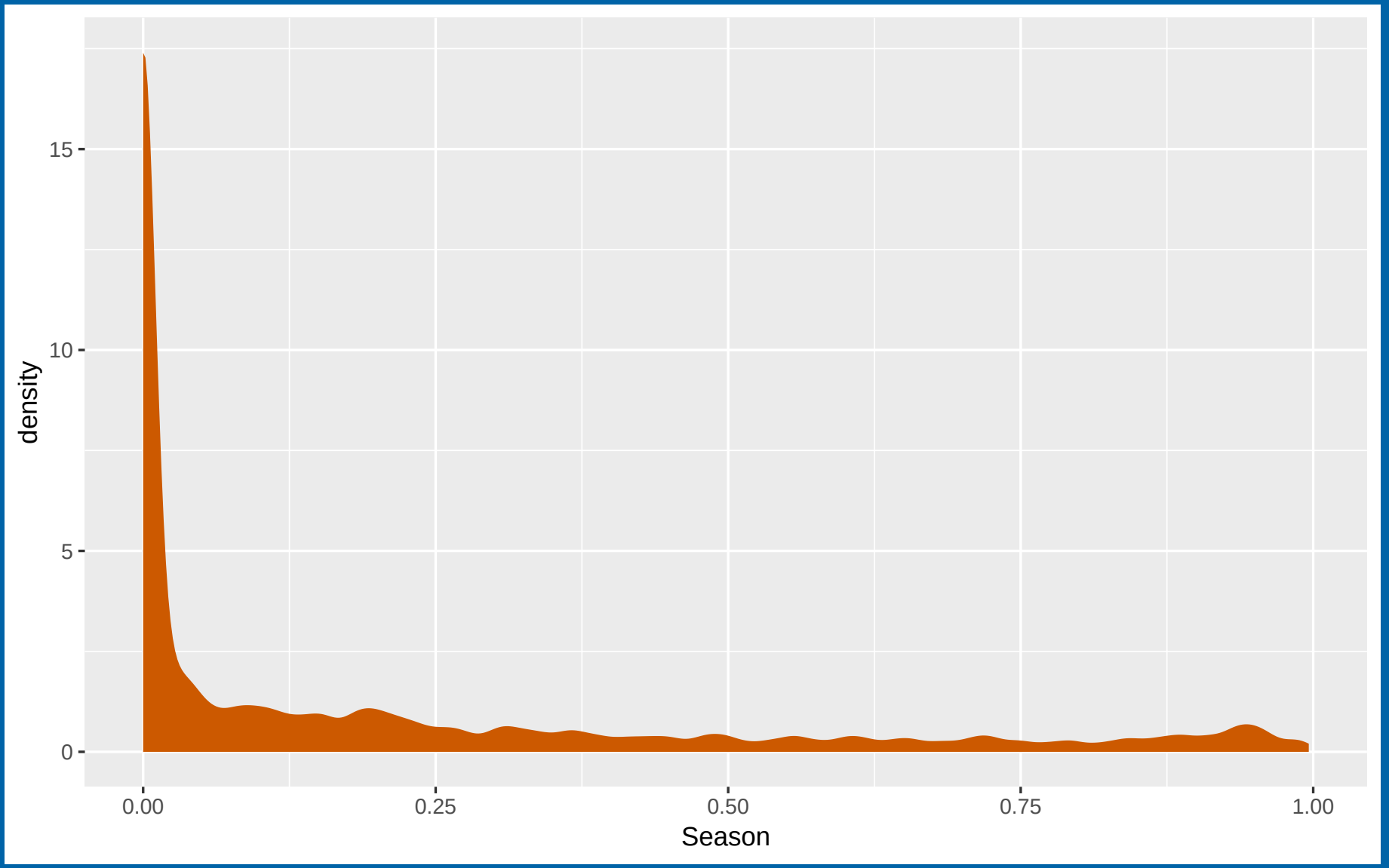
tsfeatures package

```
library(tsfeatures); library(tidyverse)
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))
```

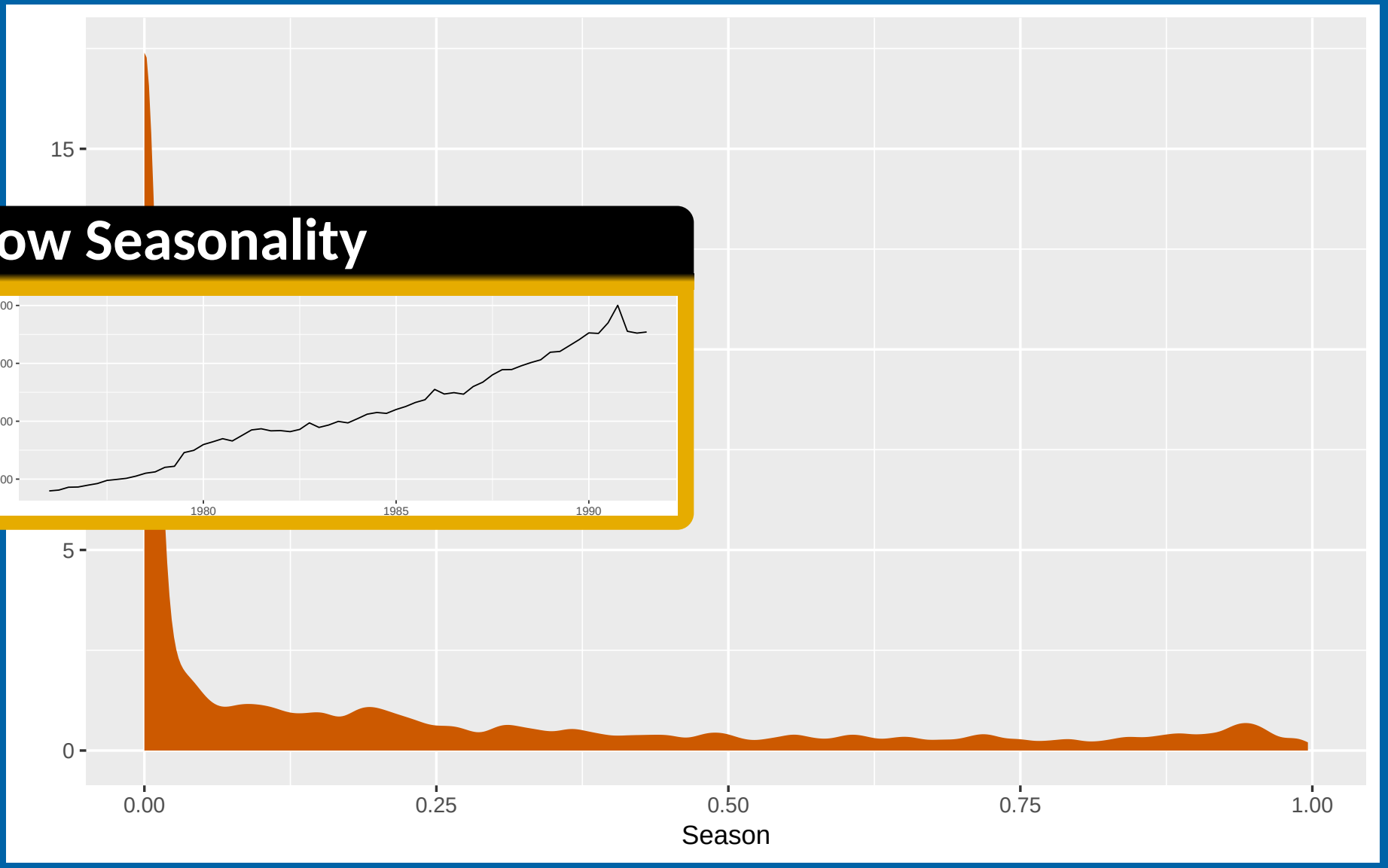
Distribution of Period for M3



Distribution of Seasonality for M3

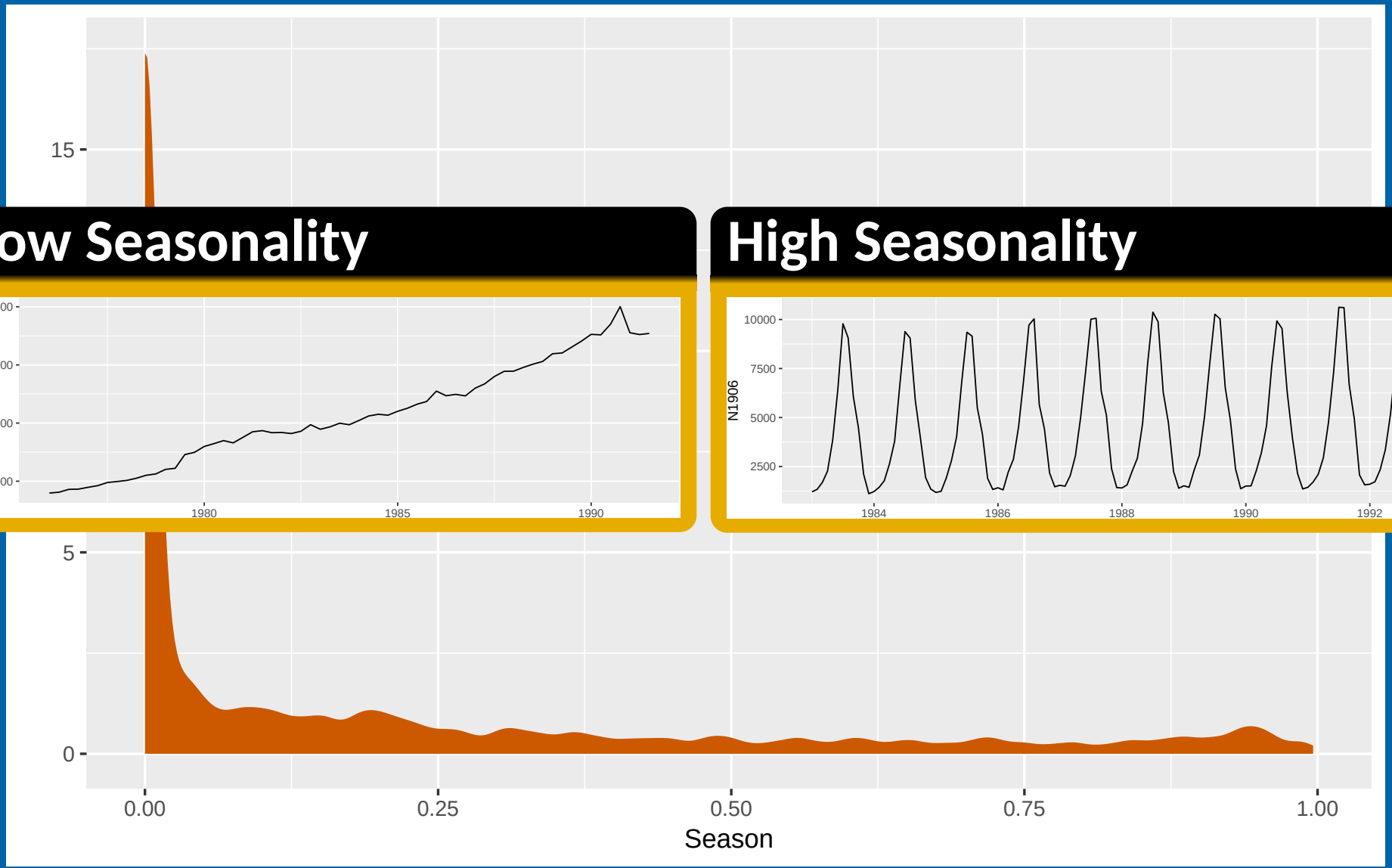


Distribution of Seasonality for M3



Low Seasonality

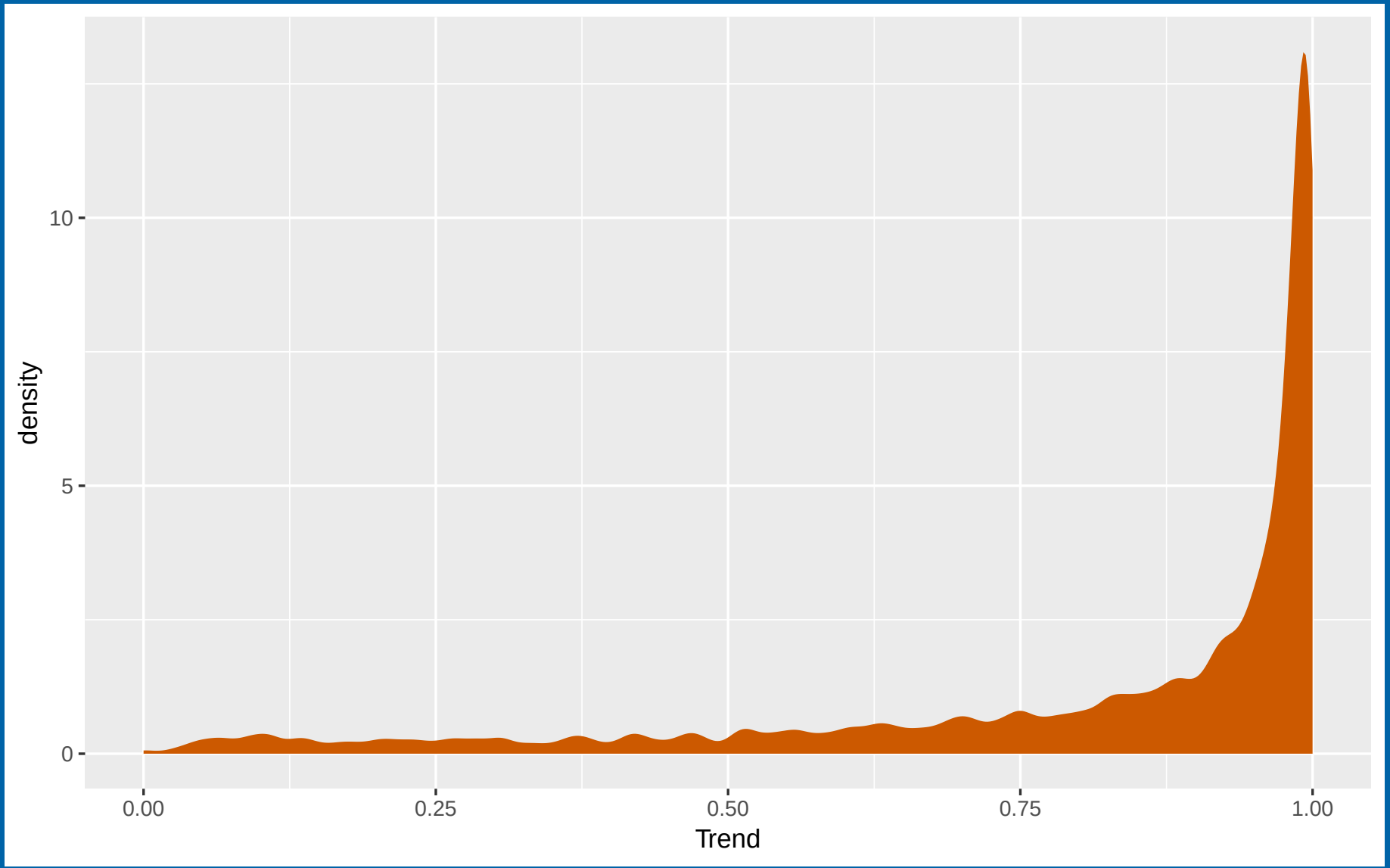
Distribution of Seasonality for M3



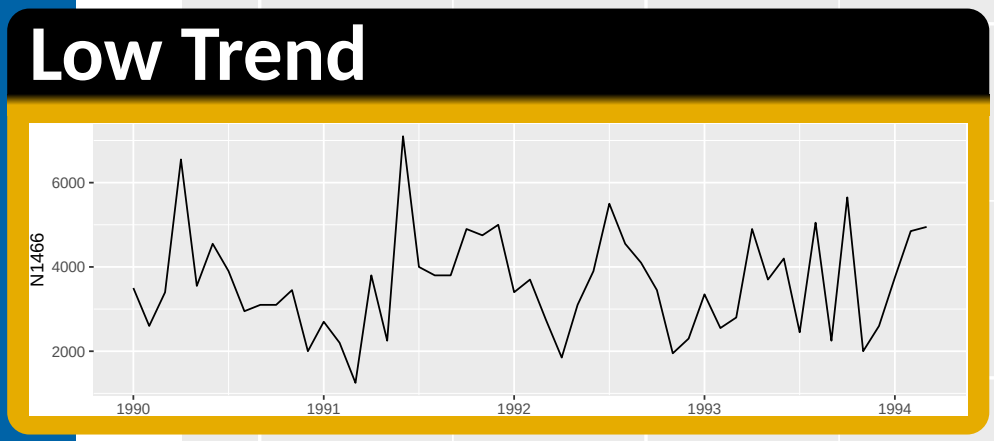
Low Seasonality

High Seasonality

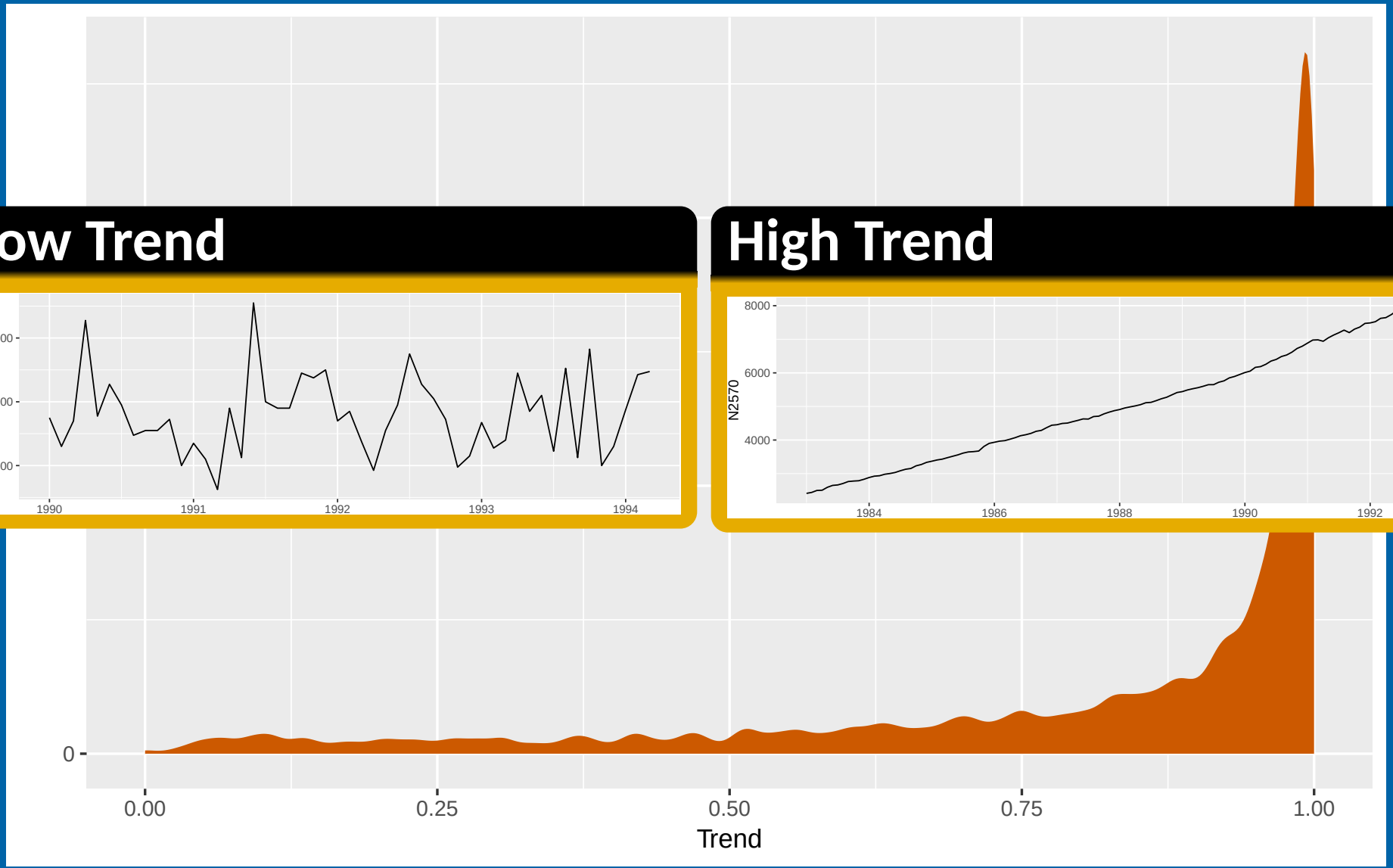
Distribution of Trend for M3



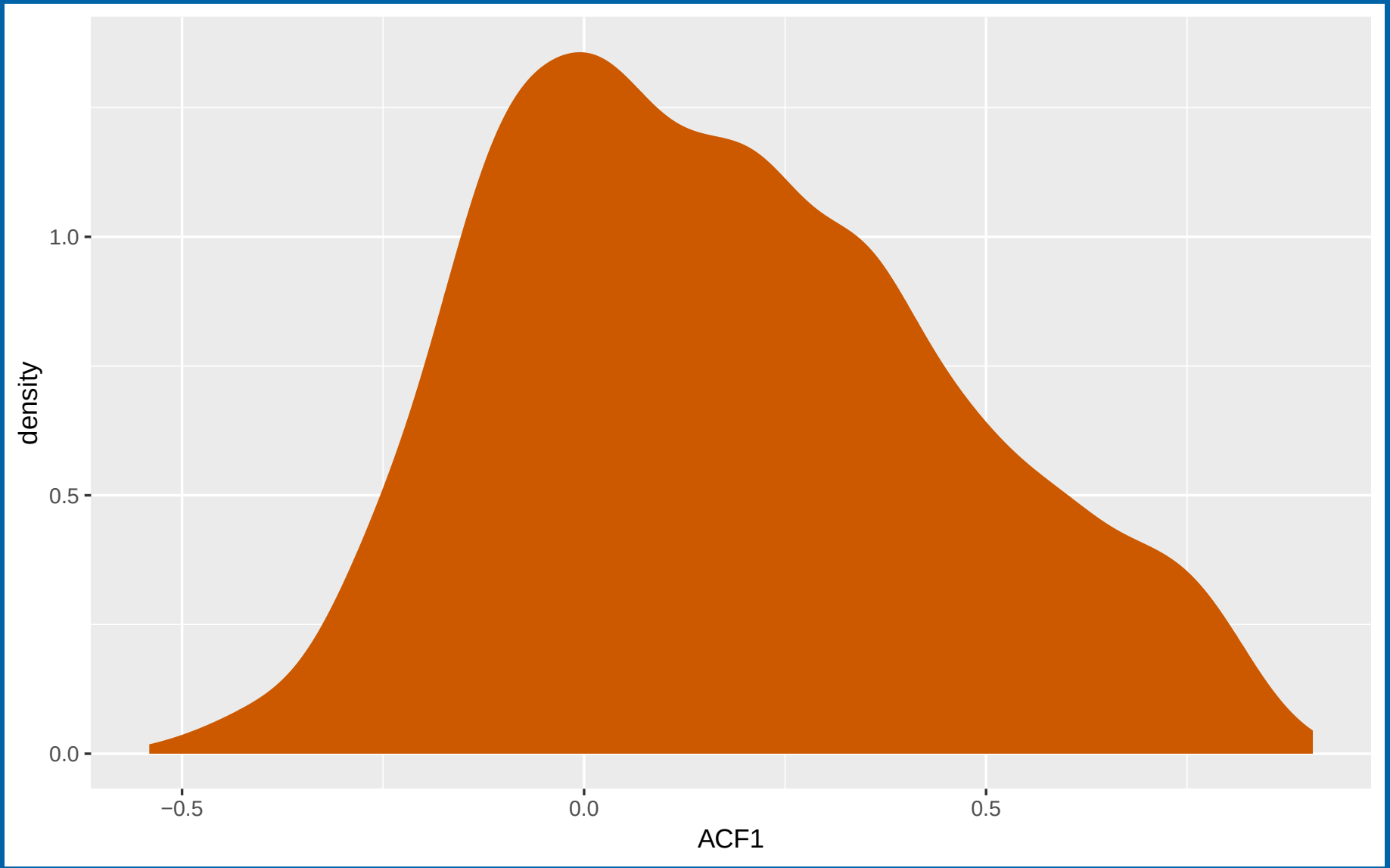
Distribution of Trend for M3



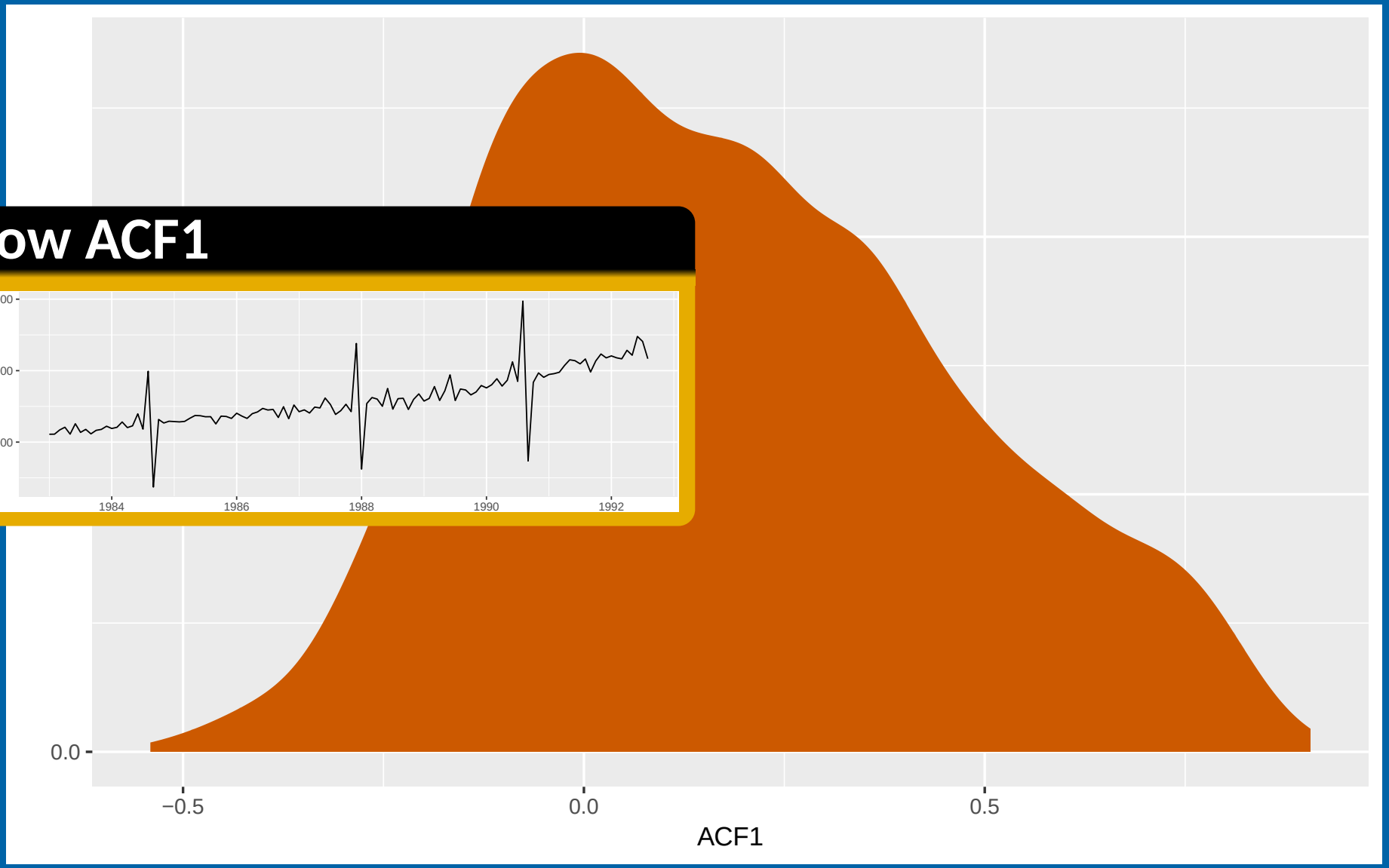
Distribution of Trend for M3



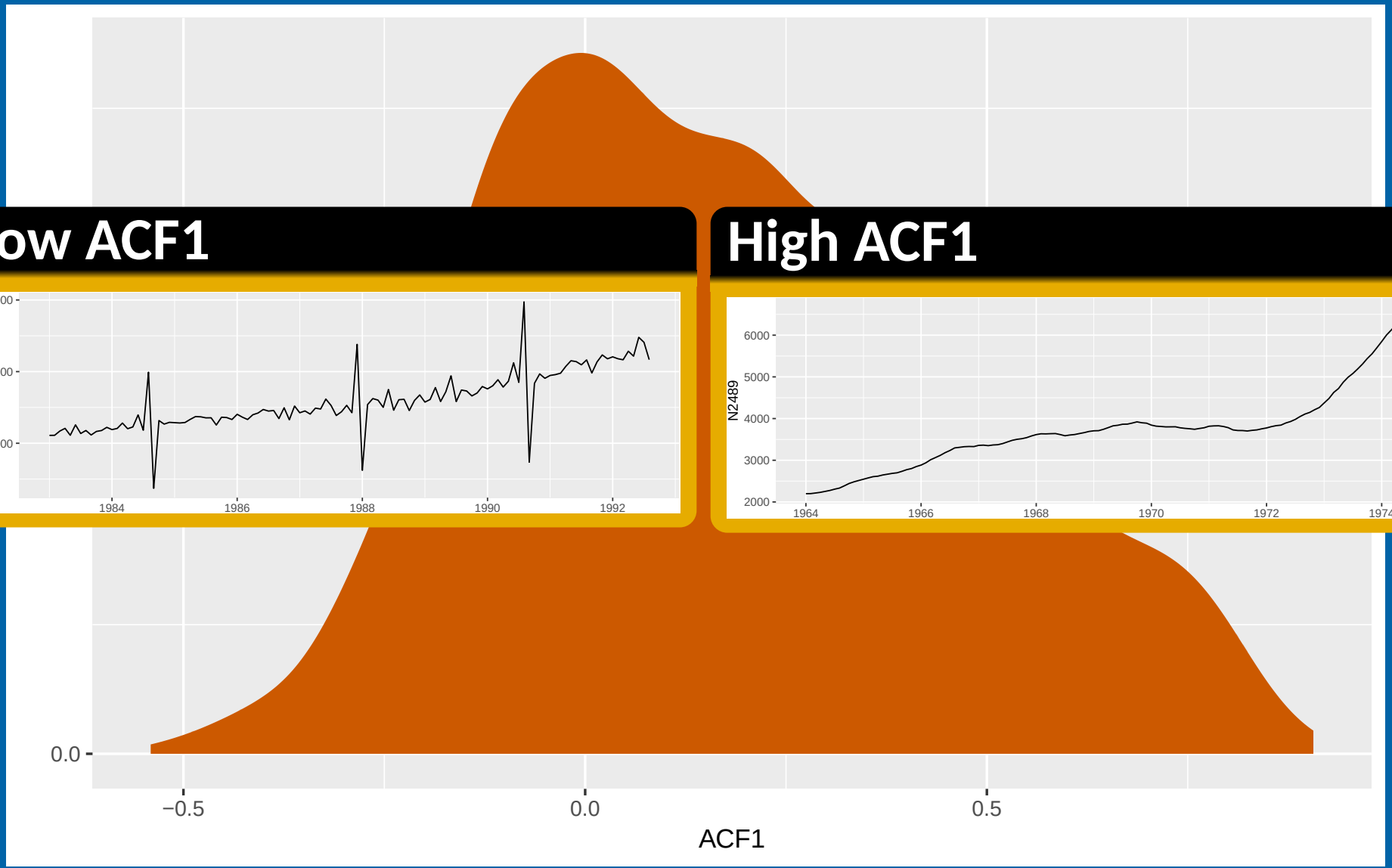
Distribution of Residual ACF1 for M3



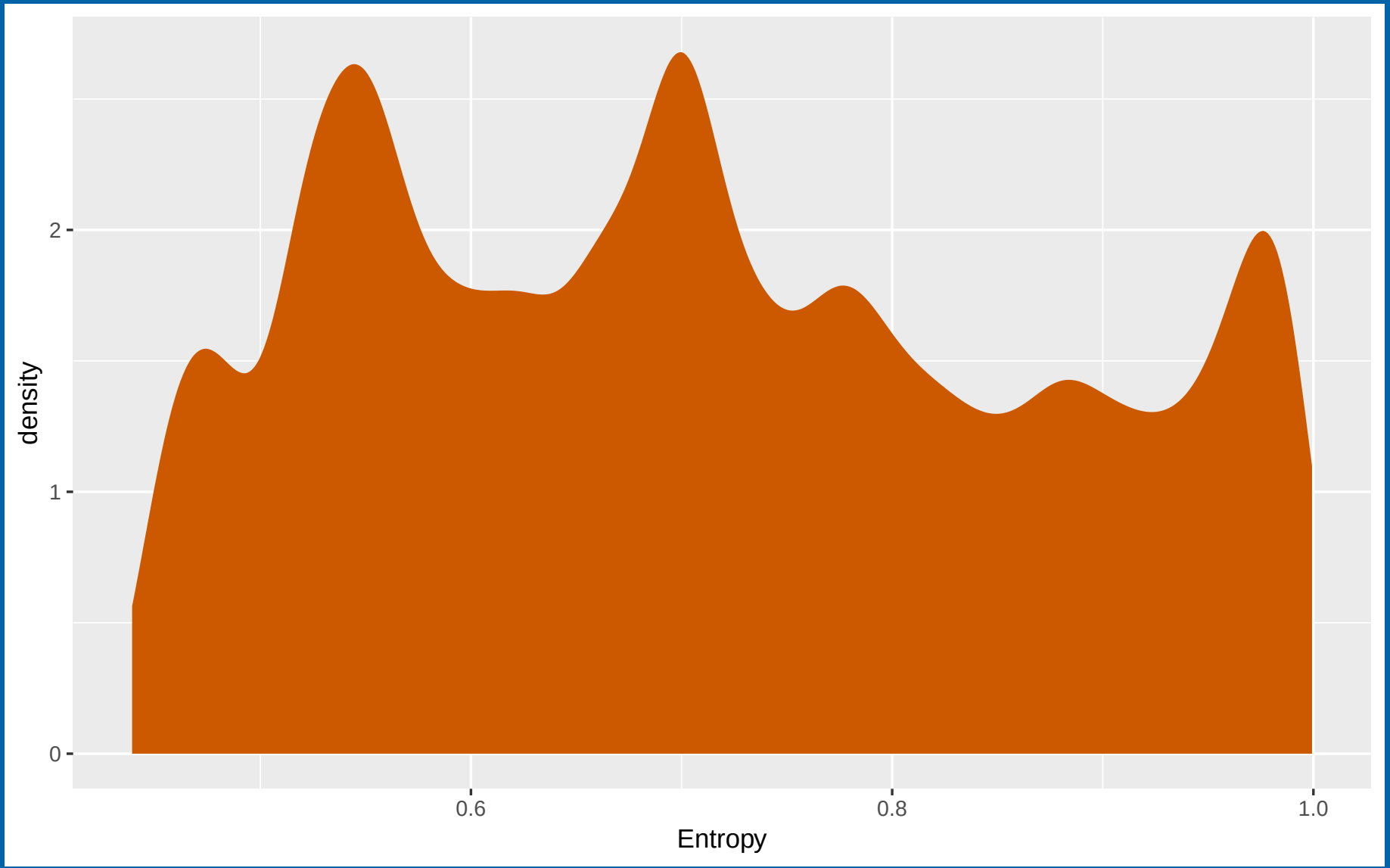
Distribution of Residual ACF1 for M3



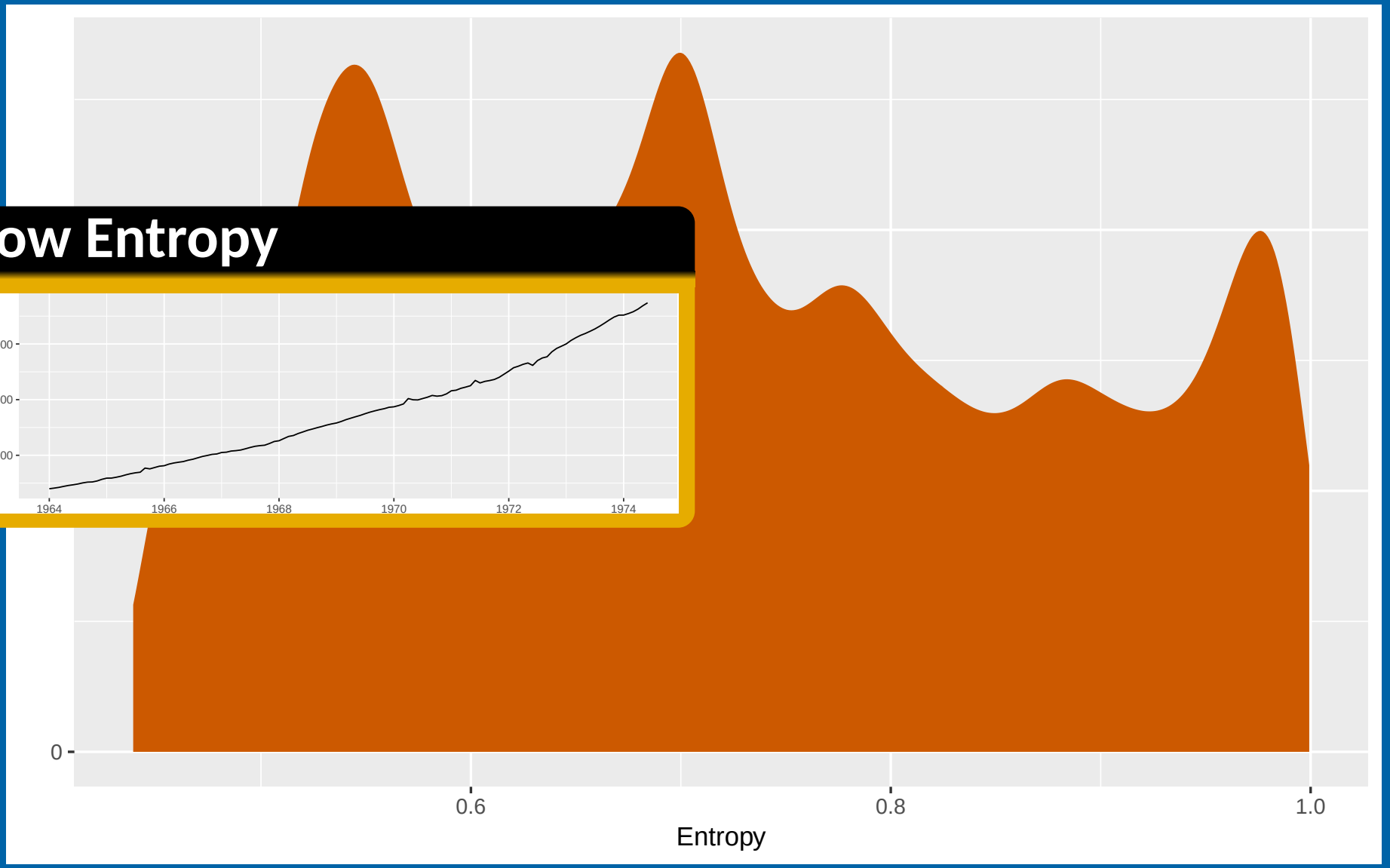
Distribution of Residual ACF1 for M3



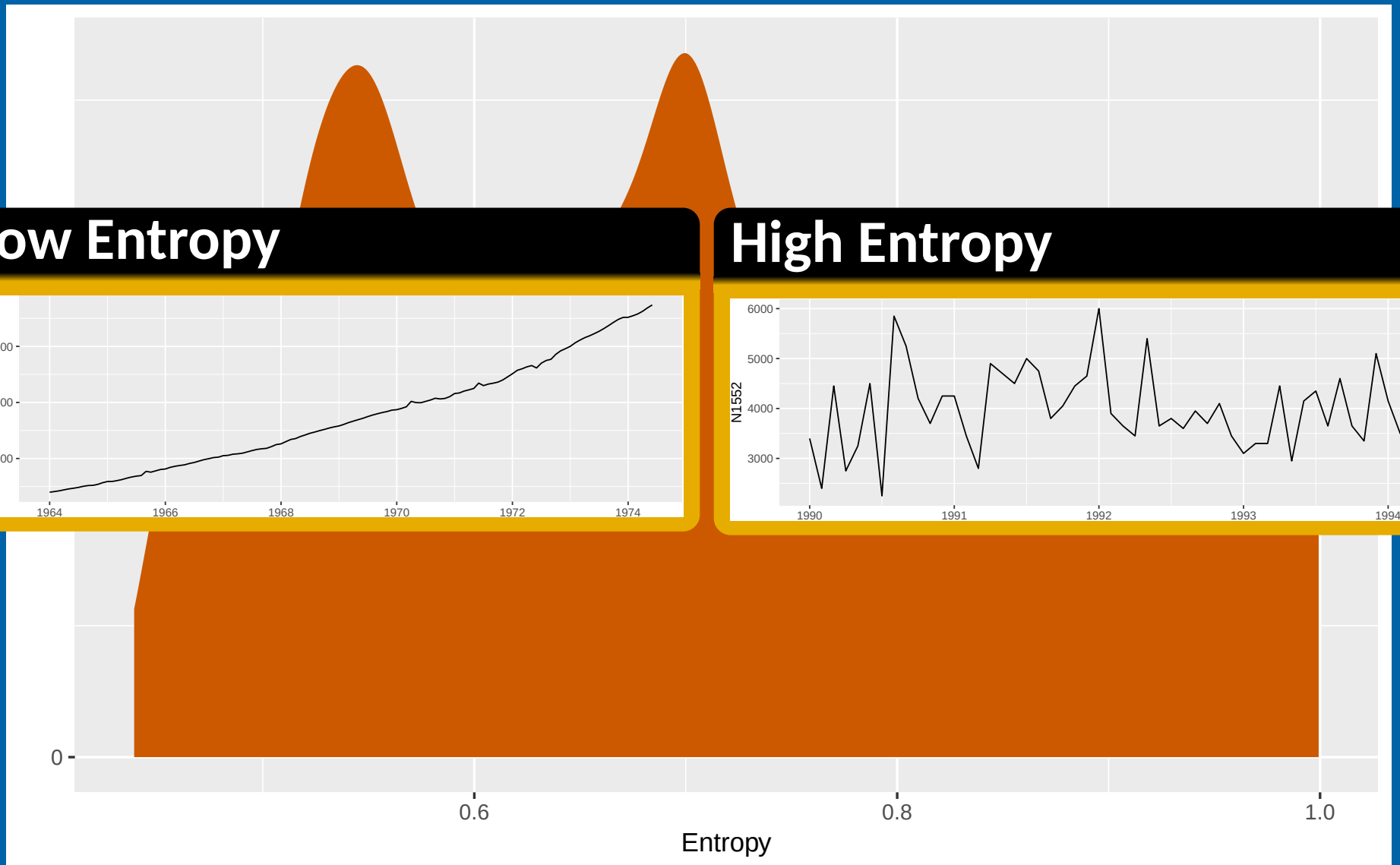
Distribution of Spectral Entropy for M3



Distribution of Spectral Entropy for M3



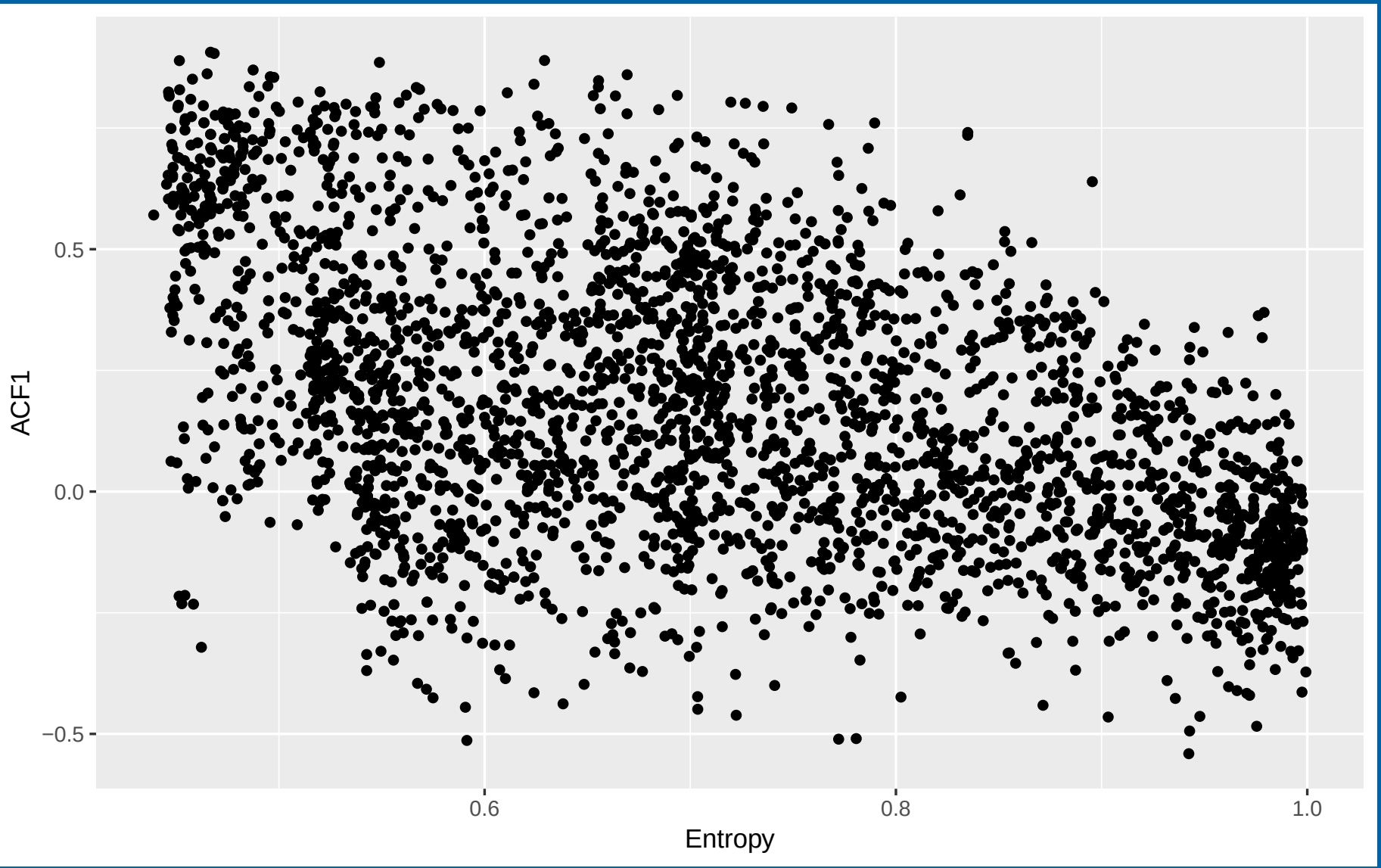
Distribution of Spectral Entropy for M3



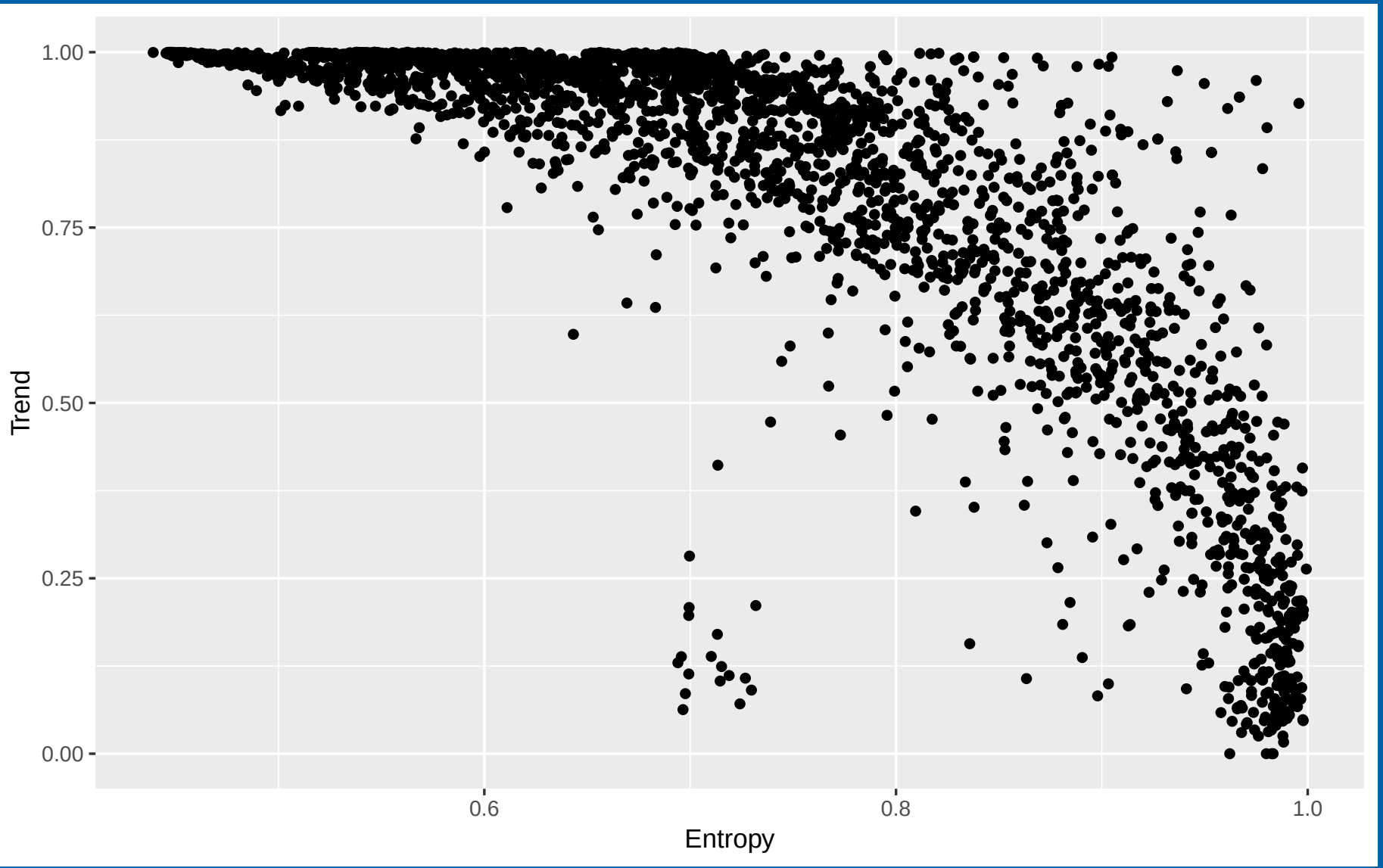
Low Entropy

High Entropy

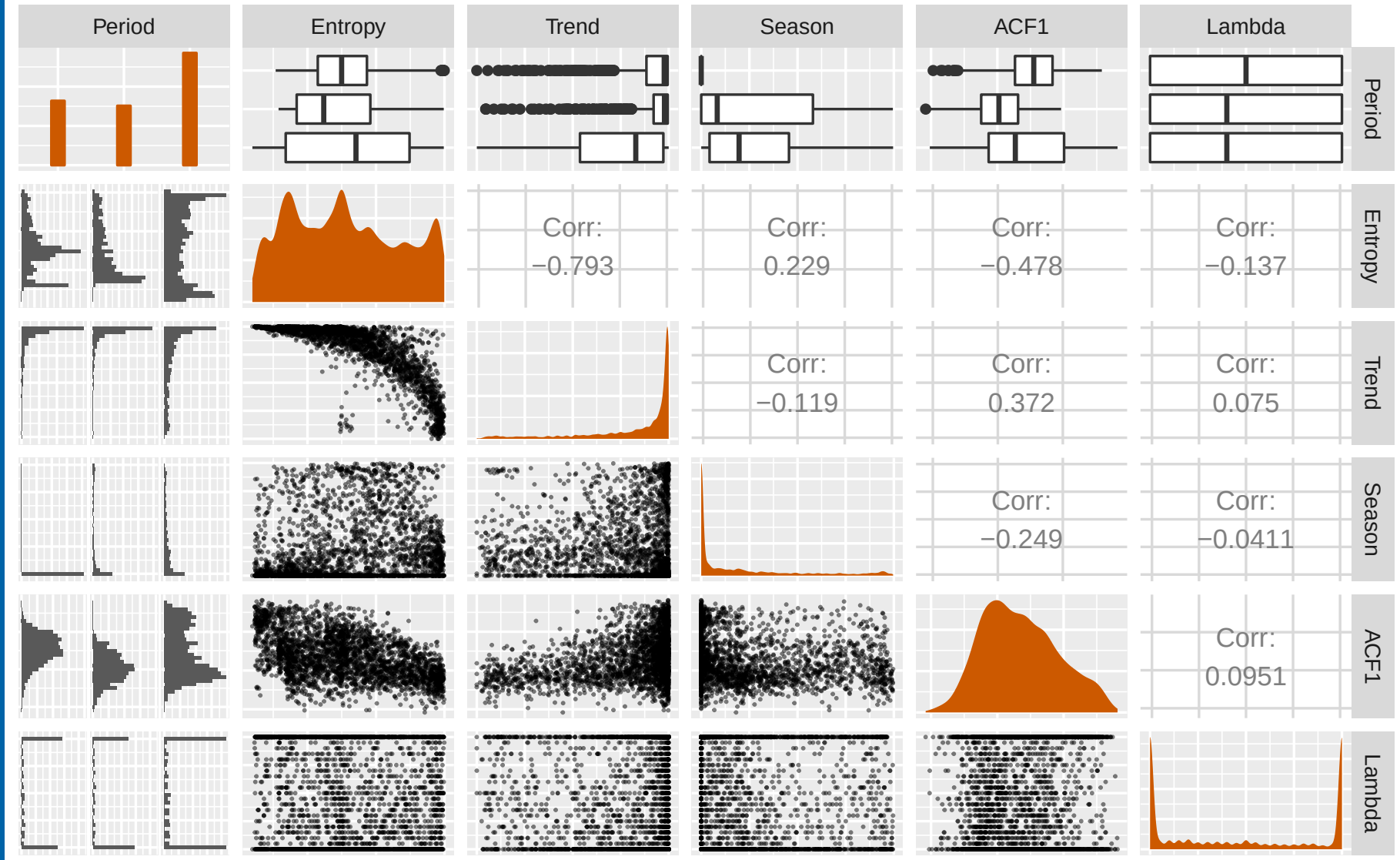
Feature distributions



Feature distributions

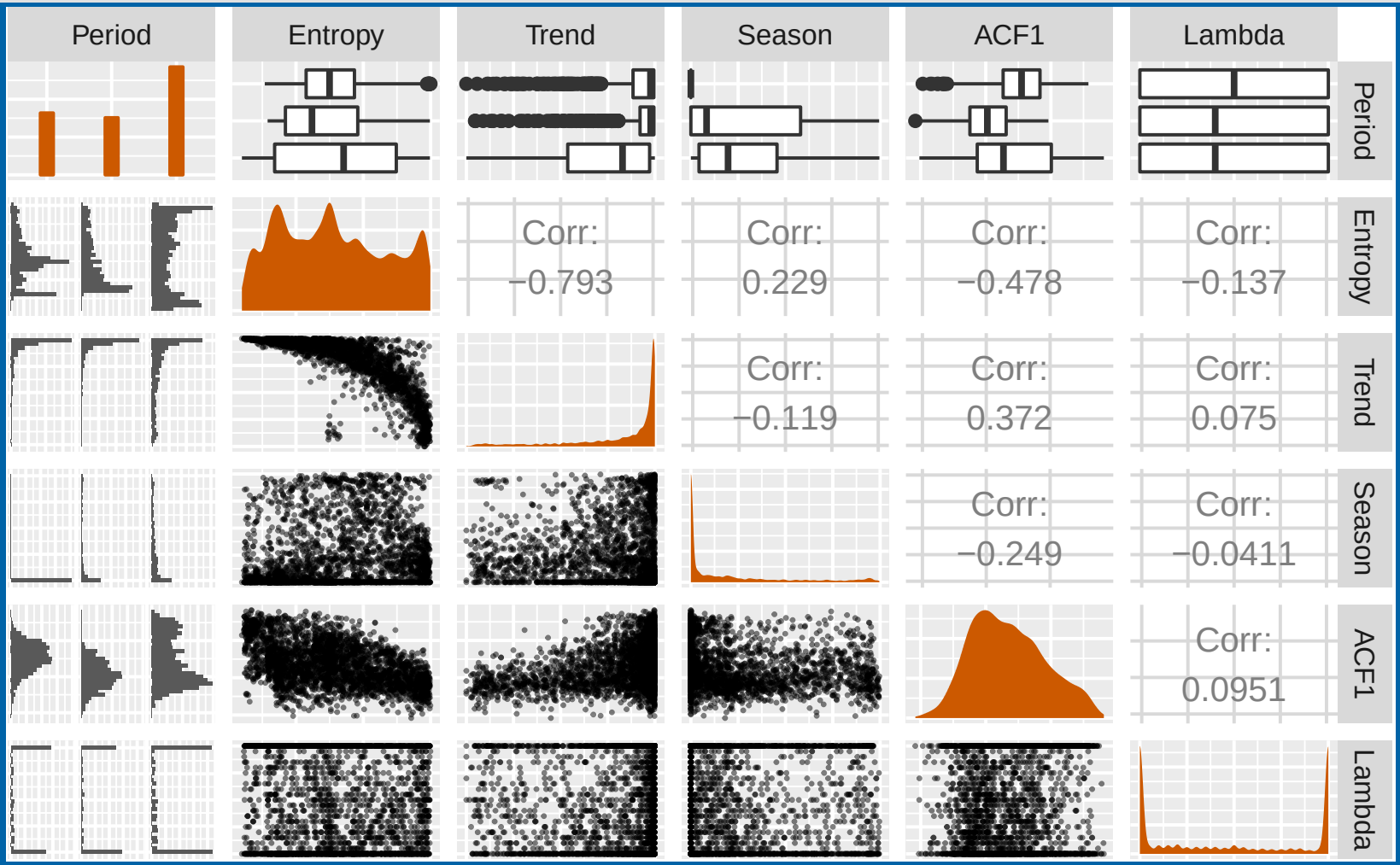


Feature distributions

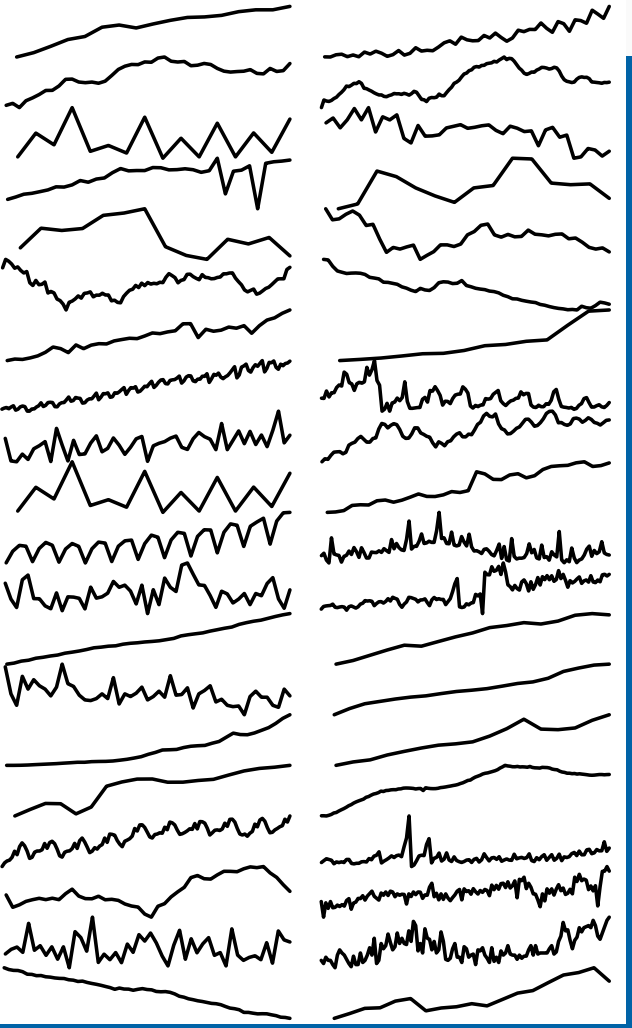


Feature distributions

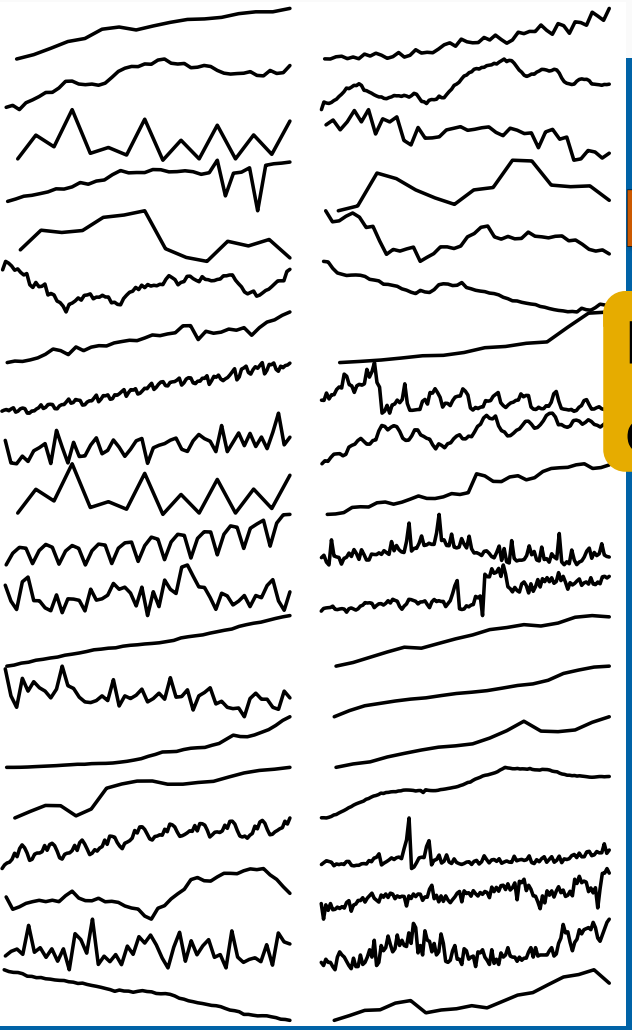
```
M3Features %>%
  select(Period, Entropy, Trend, Season, ACF1, Lambda) %>%
  GGally::ggpairs()
```



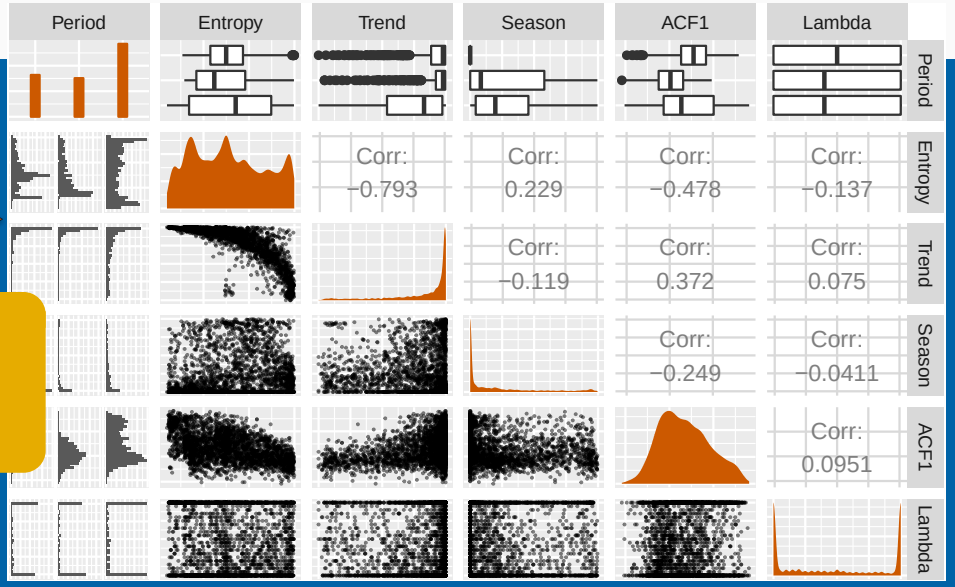
Dimension reduction for time series



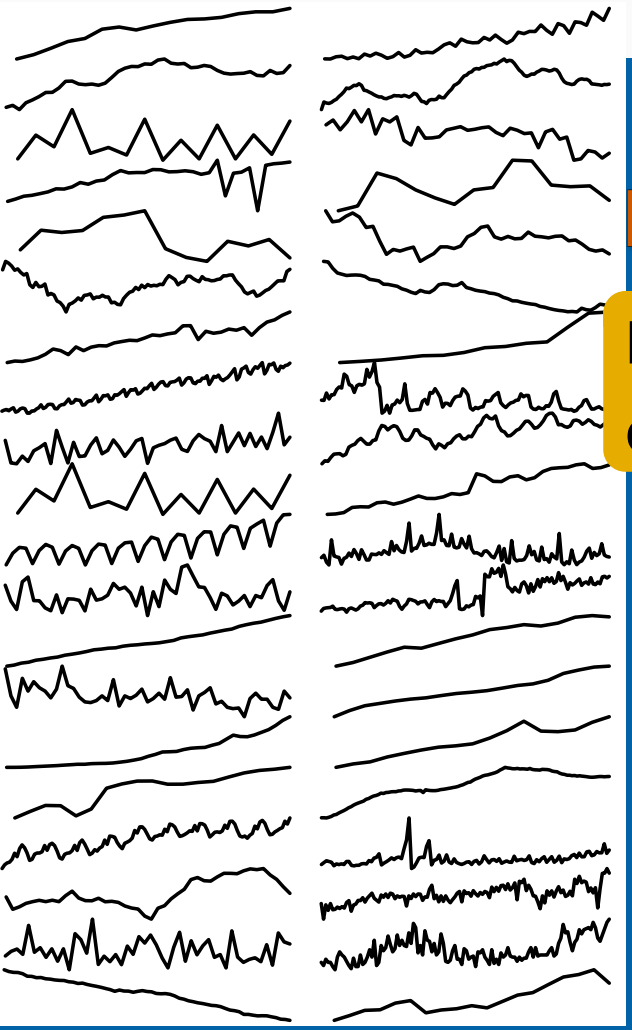
Dimension reduction for time series



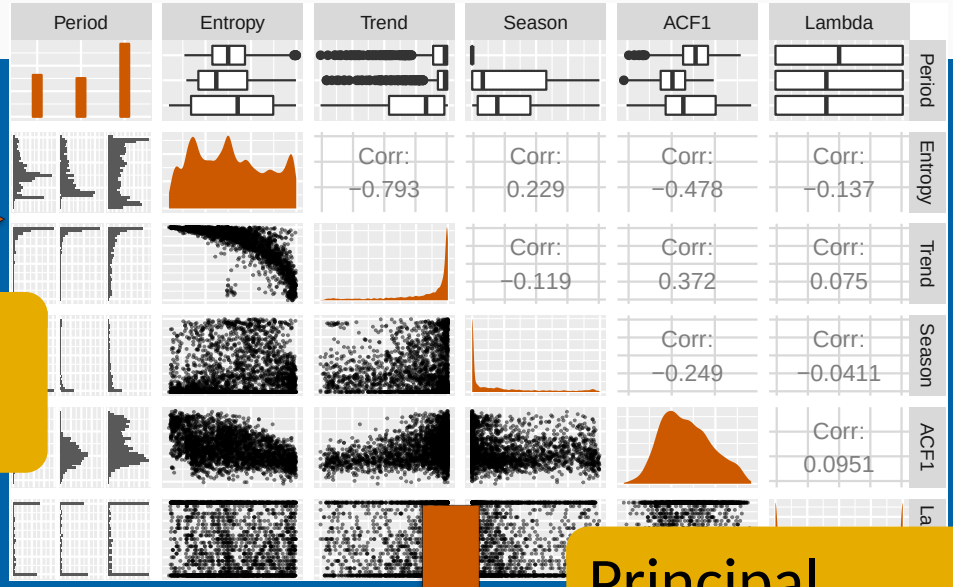
Feature calculation



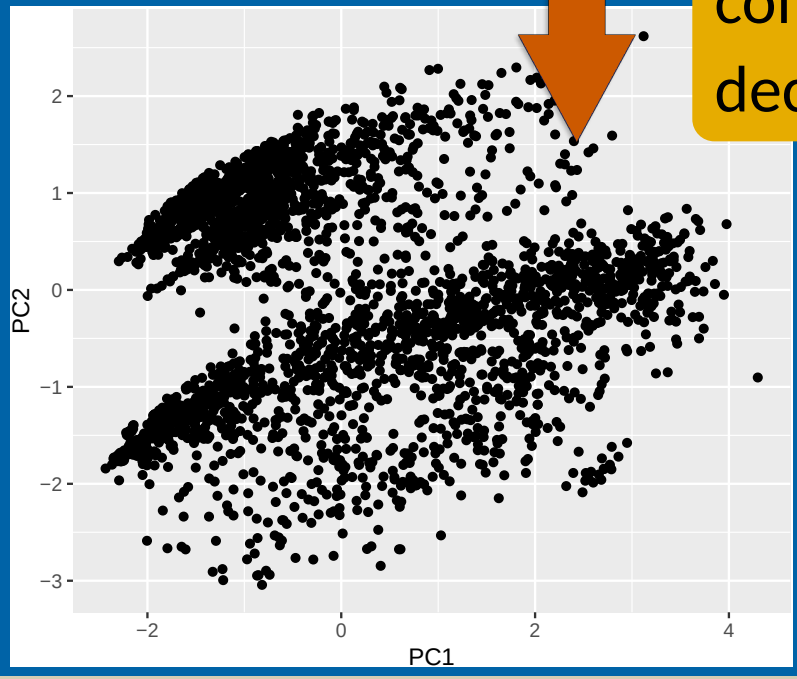
Dimension reduction for time series



Feature calculation

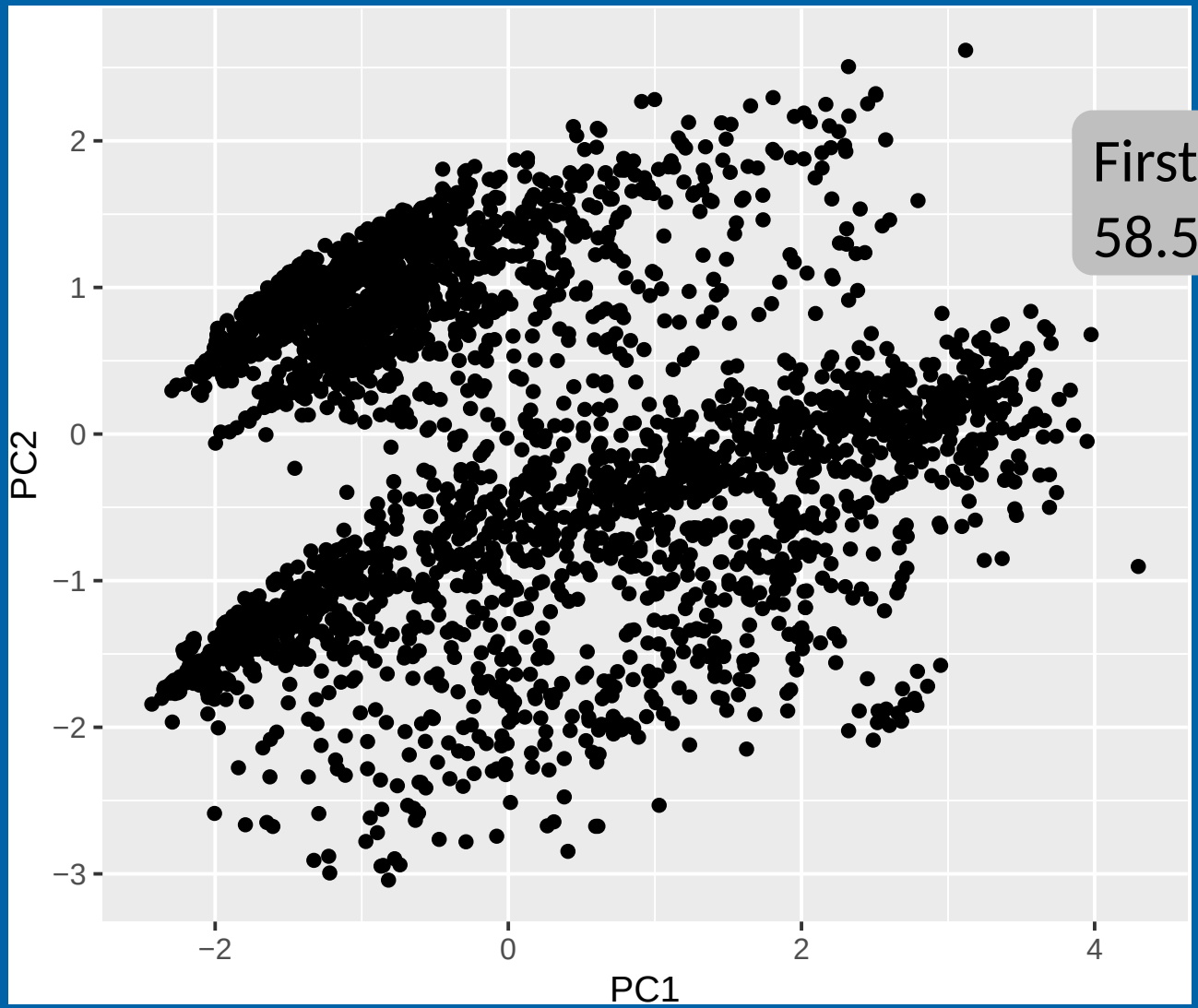


Principal component decomposition



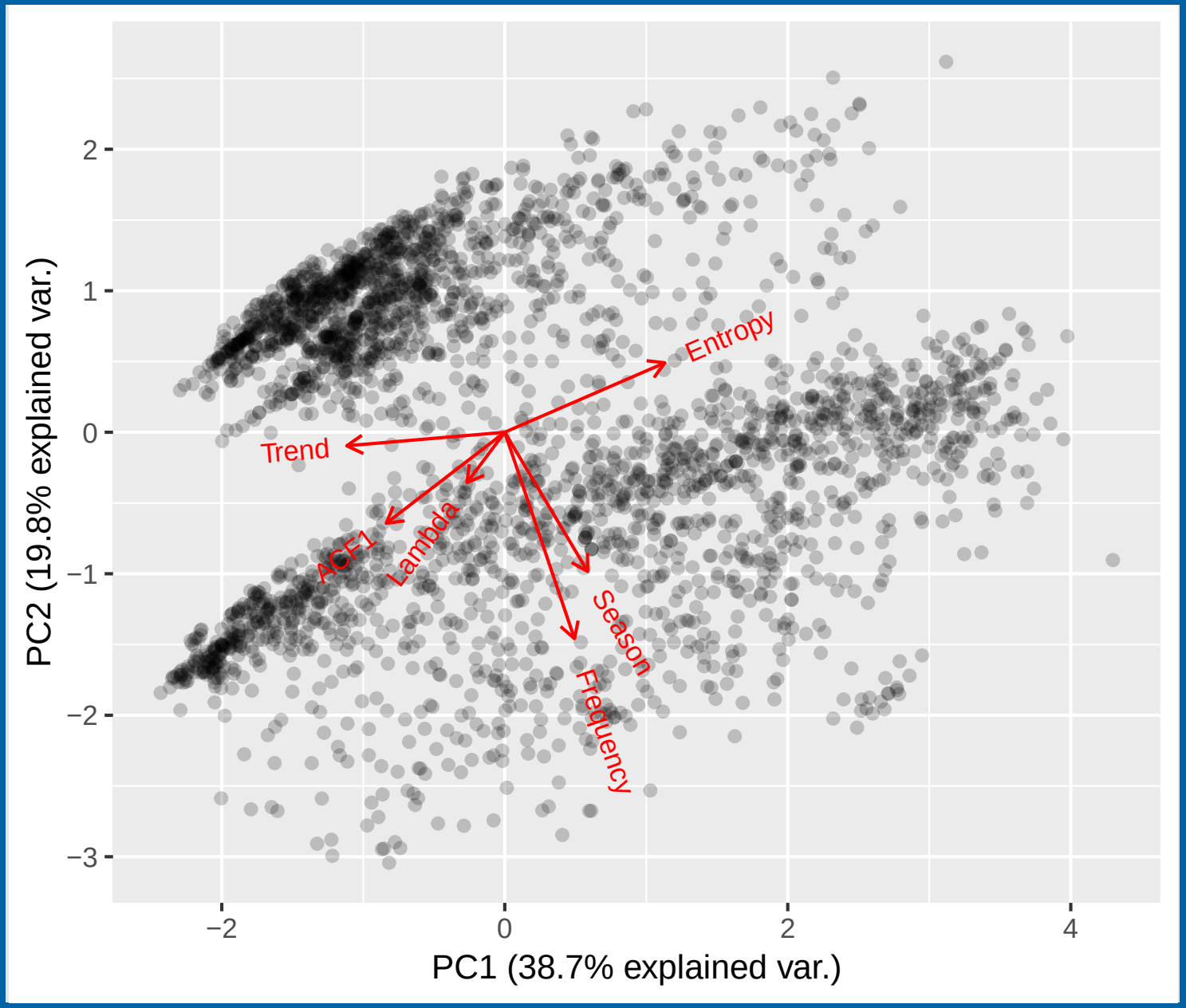
M3 feature space

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

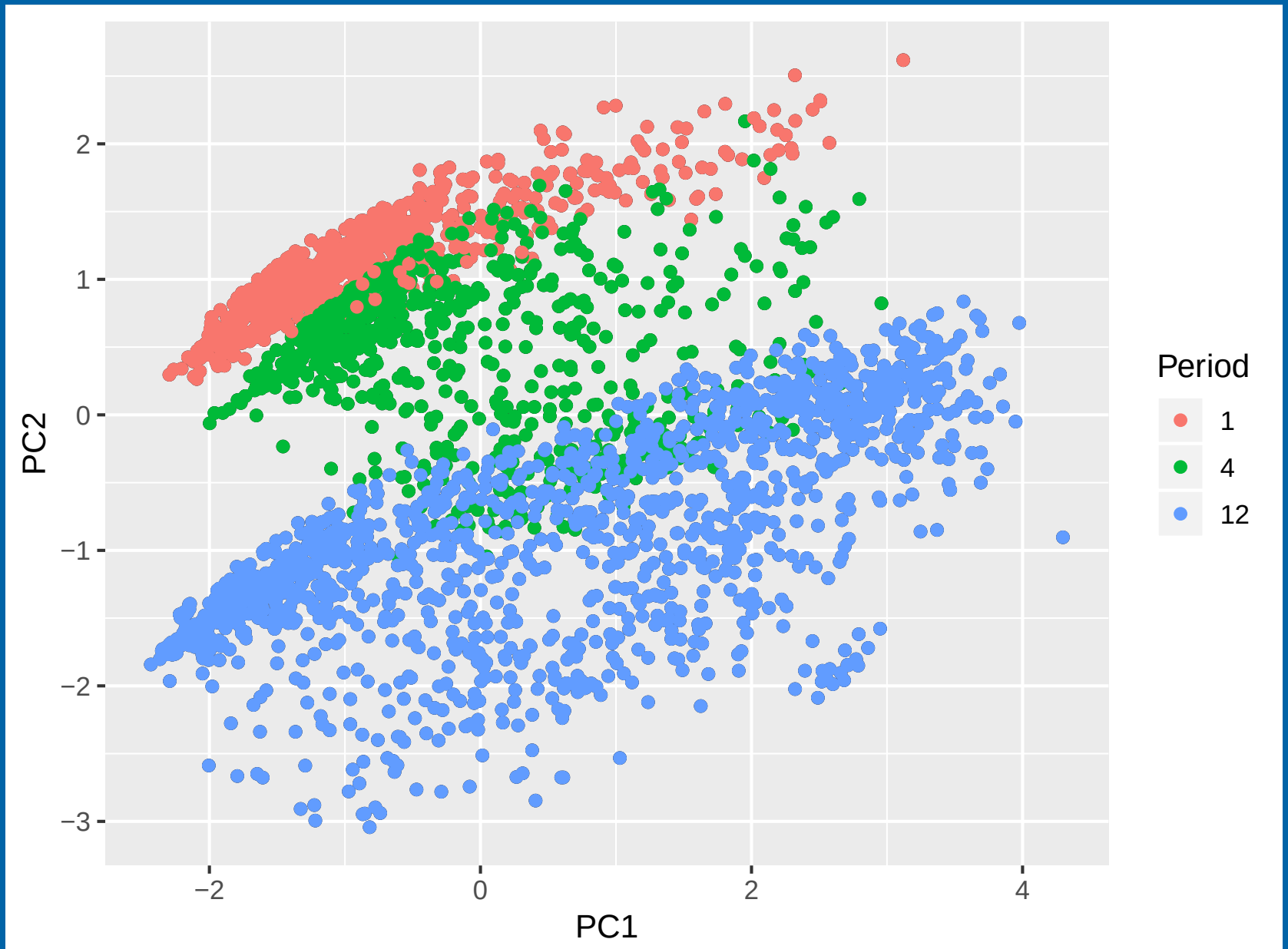


First two PCs explain 58.5% of the variance.

M3 feature space



M3 feature space



Selecting a forecasting model using seer

```
library(seer)
cal_features(subset(M3,"yearly"),
             h=6, database="M3",  highfreq=FALSE) %>%
  head()
```

```
##   entropy lumpiness stability hurst trend spikiness linearity
## 1   0.773         0         0 0.971 0.995  5.89e+05      4497
## 2   0.837         0         0 0.947 0.869  6.02e+08      2781
## 3   0.825         0         0 0.949 0.865  6.69e+08      2389
## 4   0.854         0         0 0.949 0.853  1.24e+09      3891
## 5   0.899         0         0 0.855 0.586  8.96e+08      -701
## 6   0.798         0         0 0.964 0.964  8.16e+06      2785
##   curvature e_acf1 y_acf1 diff1y_acf1 diff2y_acf1 y_pacf5
## 1       532  0.412  0.762      0.5974    -0.00481   0.615
## 2     -2824  0.324  0.751      0.2400    -0.39825   0.809
## 3     -3078  0.457  0.769      0.4461    -0.21180   0.917
## 4     -1697  0.281  0.692      0.1111    -0.32978   0.580
## 5     -1620  0.192  0.609     -0.0160    -0.44559   0.502
## 6      -449  0.181  0.701      0.0967    -0.29606   0.510
##   diff1y_pacf5 diff2y_pacf5 nonlinearity lmres_acf1 ur_pp
```

Selecting a forecasting model using seer

```
fcast_accuracy(M3[1:4],
  models=c("arima","ets","rw","rwd","theta","nn","snaive"),
  database="M3", cal_MASE, h=6, length_out=1)
```

```
## $accuracy
##       arima   ets    rw   rwd theta   nn snaive
## N0001 1.567 1.564 7.704 4.204 6.017 2.57  7.704
## N0002 1.698 0.923 1.698 0.612 1.096 0.28  1.698
## N0003 0.375 0.375 0.375 1.308 0.815 2.88  0.375
## N0004 0.868 1.498 0.868 0.884 0.450 1.77  0.868
##
## $ARIMA
##          N0001          N0002          N0003          N0004
## "ARIMA(0,2,0)" "ARIMA(0,1,0)" "ARIMA(0,1,0)" "ARIMA(0,1,0)"
##
## $ETS
##        N0001        N0002        N0003        N0004
## "ETS(M,A,N)" "ETS(M,A,N)" "ETS(A,N,N)" "ETS(M,A,N)"
```

Selecting a forecasting model using seer

FFORMS: Feature-based FORecast Model Selection

- We train a random forest using feature inputs and labels equal to best-performing method on test sets.
- Training data augmented by simulating from fitted models.
- Entered M4 competition after training on M1, M3, and truncated M4 series + augmented series.

Forecast model averaging

FFORMA: Feature-based FORecast Model Averaging

- We use xgboost with feature inputs and labels equal to best-performing method on test sets. The optimization criterion is forecast accuracy not classification accuracy.
- The probability of each model being best is used to construct a model weight.
- All forecasts are averaged using weights.
- **Came second in the M4 competition**

Outline

1 M3 forecasting competition

2 Yahoo server metrics

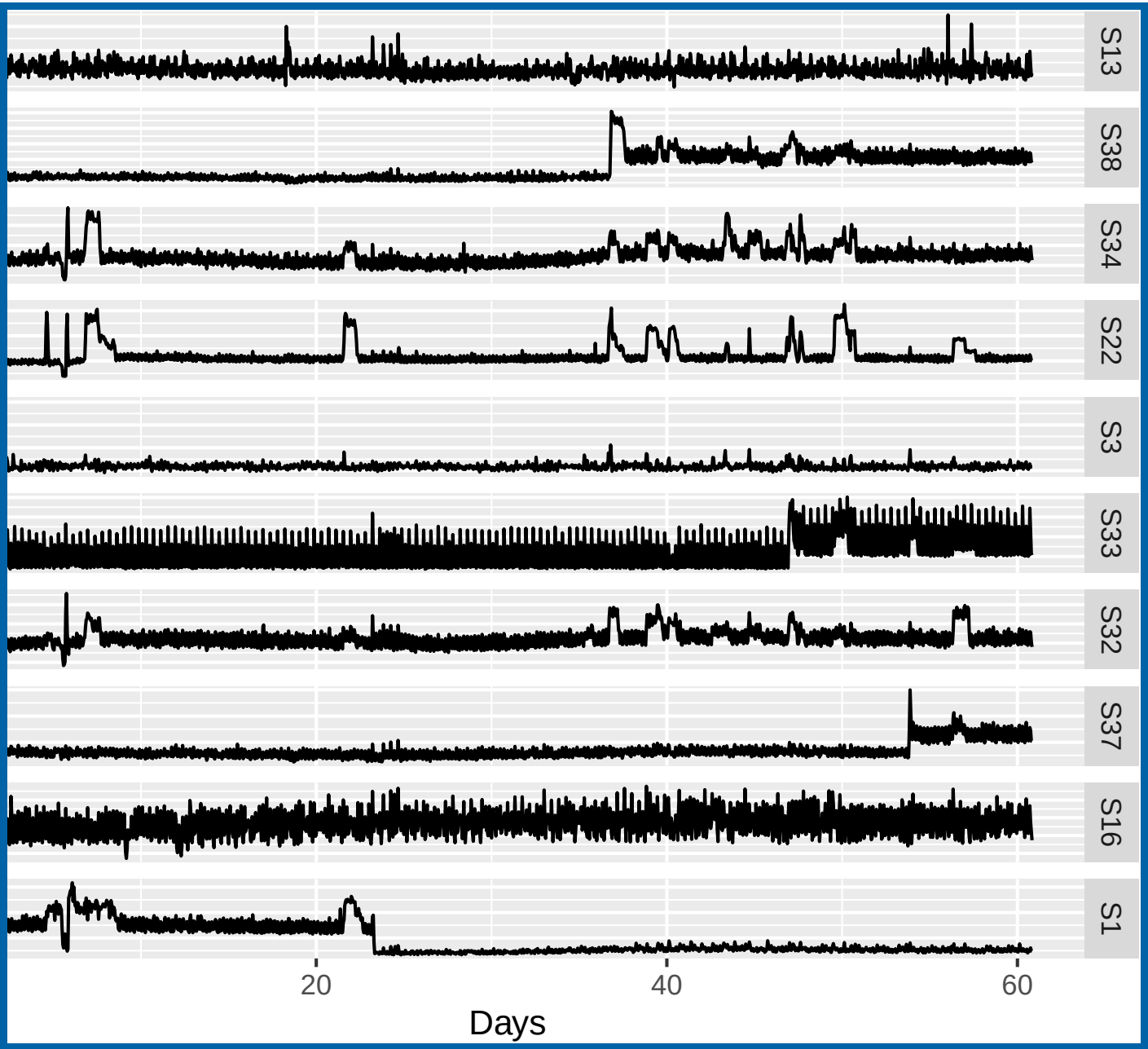
3 Irish smart metres

4 Packages

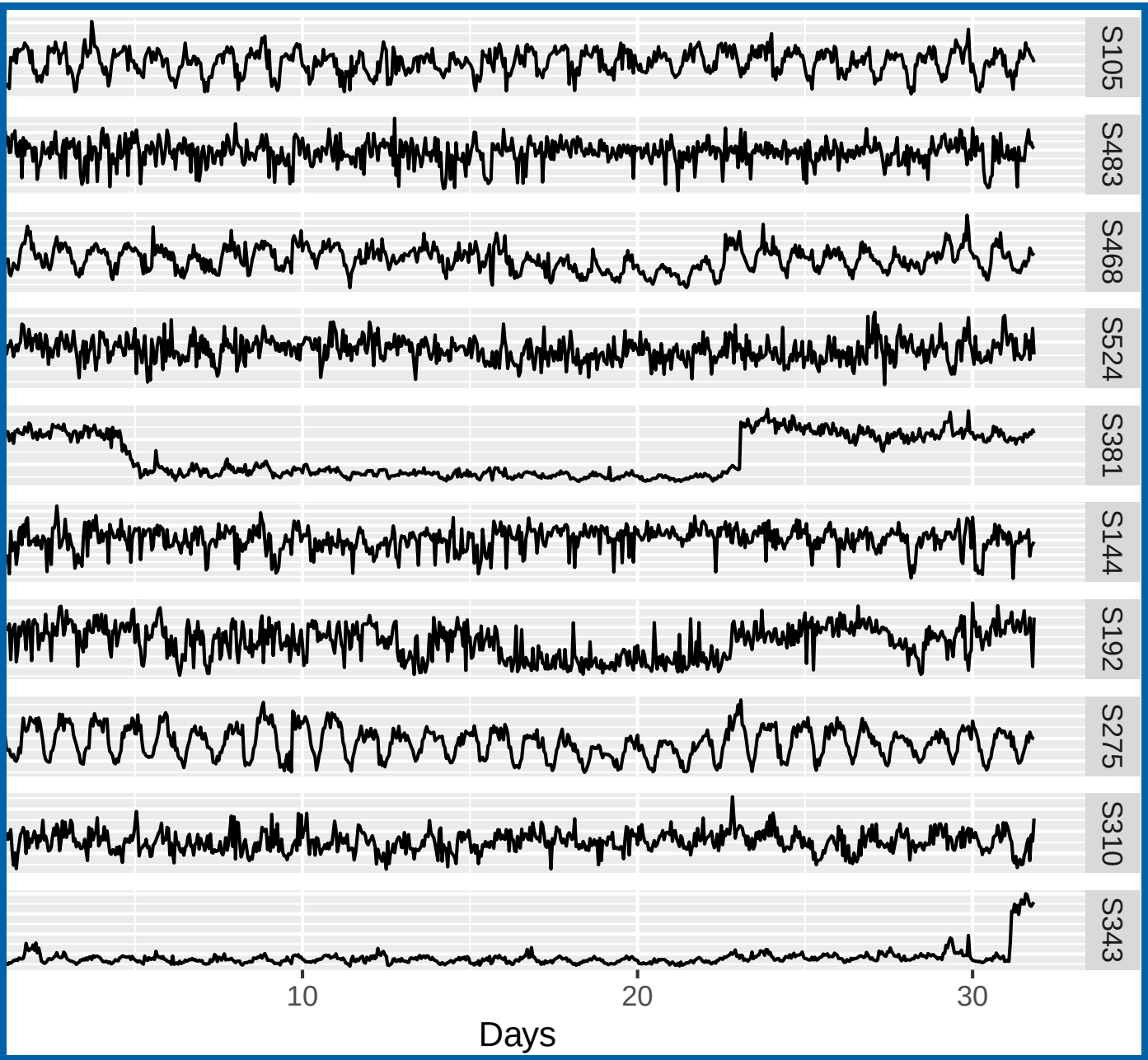
Yahoo server metrics

- Tens of thousands of time series collected at one-hour intervals over 1–2 months.
- Consisting of several server metrics (e.g. CPU usage and paging views) from many server farms globally.
- Aim: find unusual (anomalous) time series.

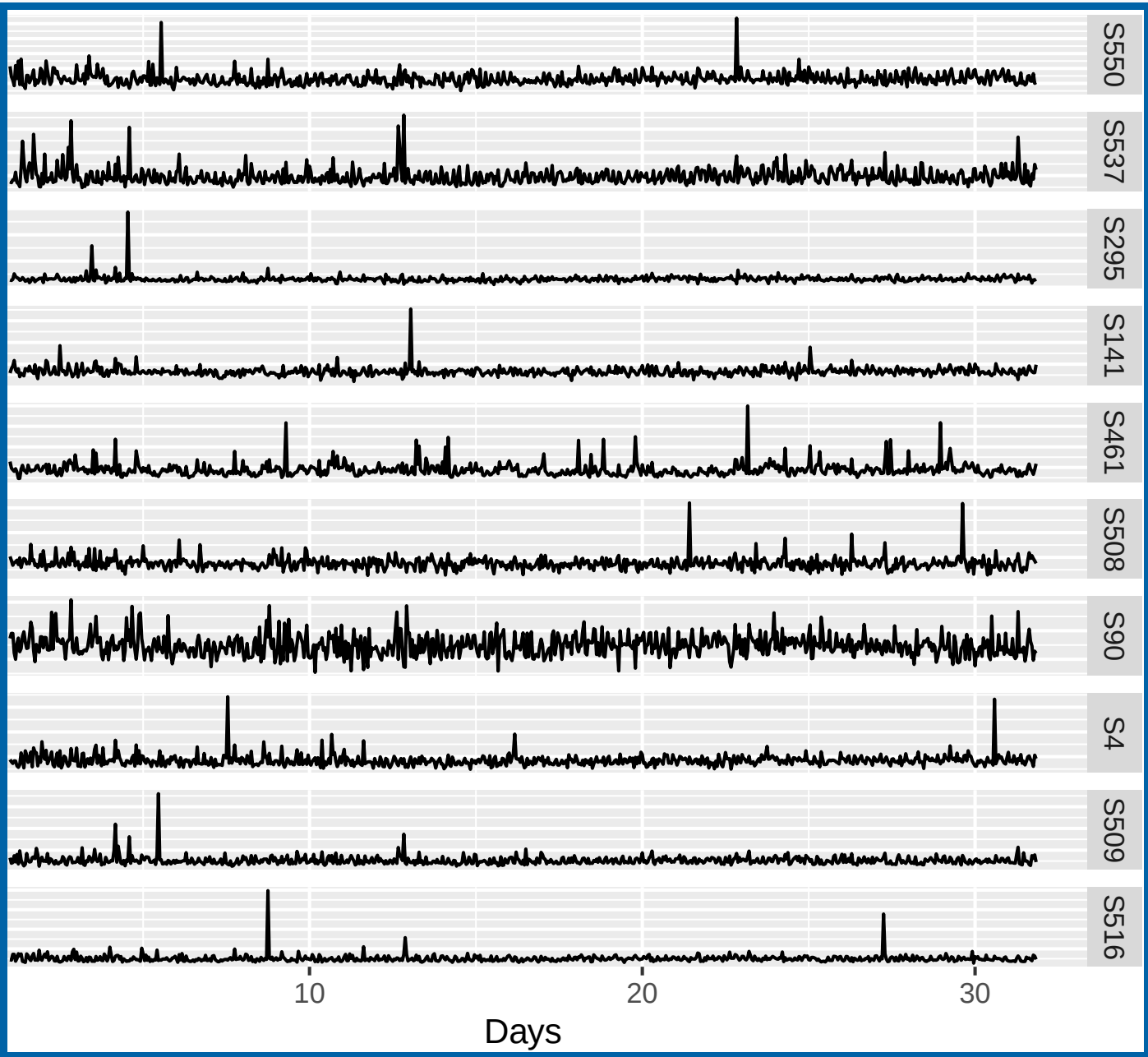
Yahoo server metrics



Yahoo server metrics



Yahoo server metrics



Yahoo server metrics

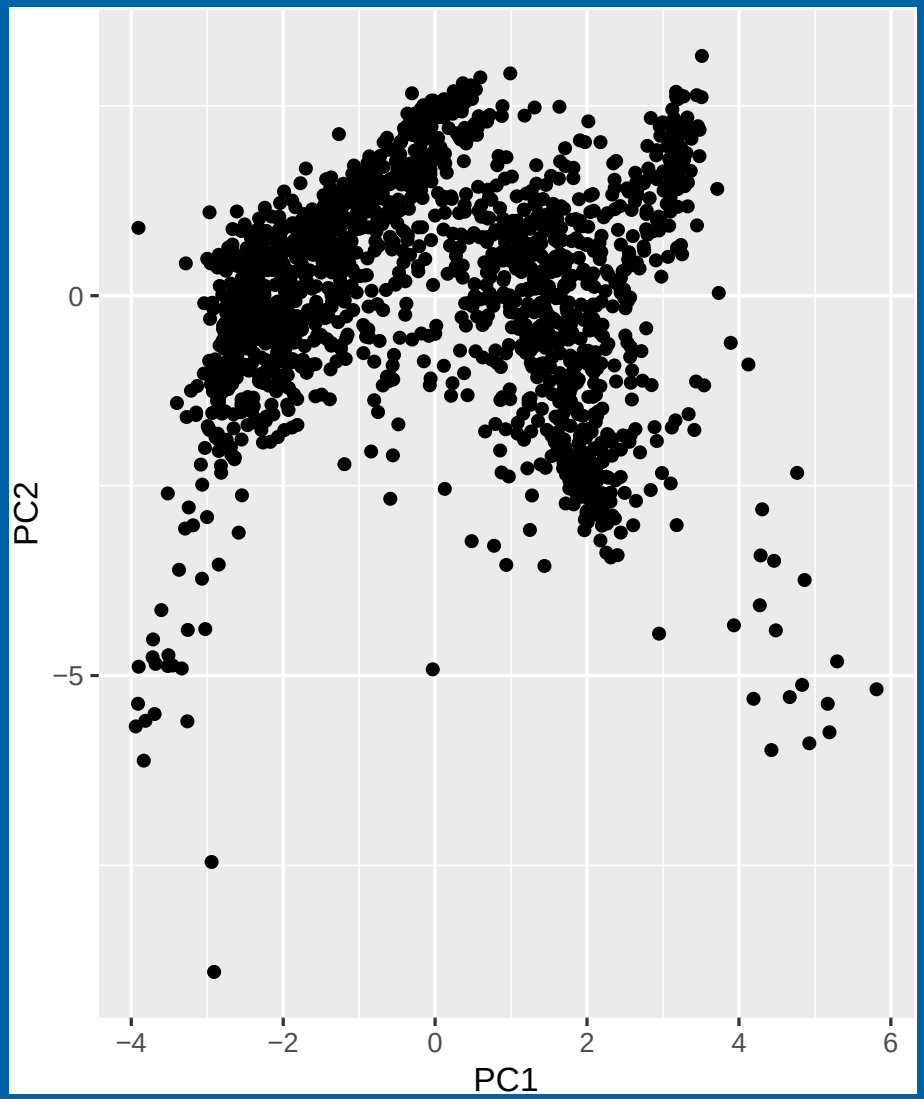
- **ACF1**: first order autocorrelation = Corr(Y_t, Y_{t-1})
- Strength of **trend** and **seasonality** based on STL
- Size of seasonal **peak** and **trough**
- Spectral **entropy**
- **Lumpiness**: variance of block variances (block size 24).
- **Spikiness**: variances of leave-one-out variances of 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

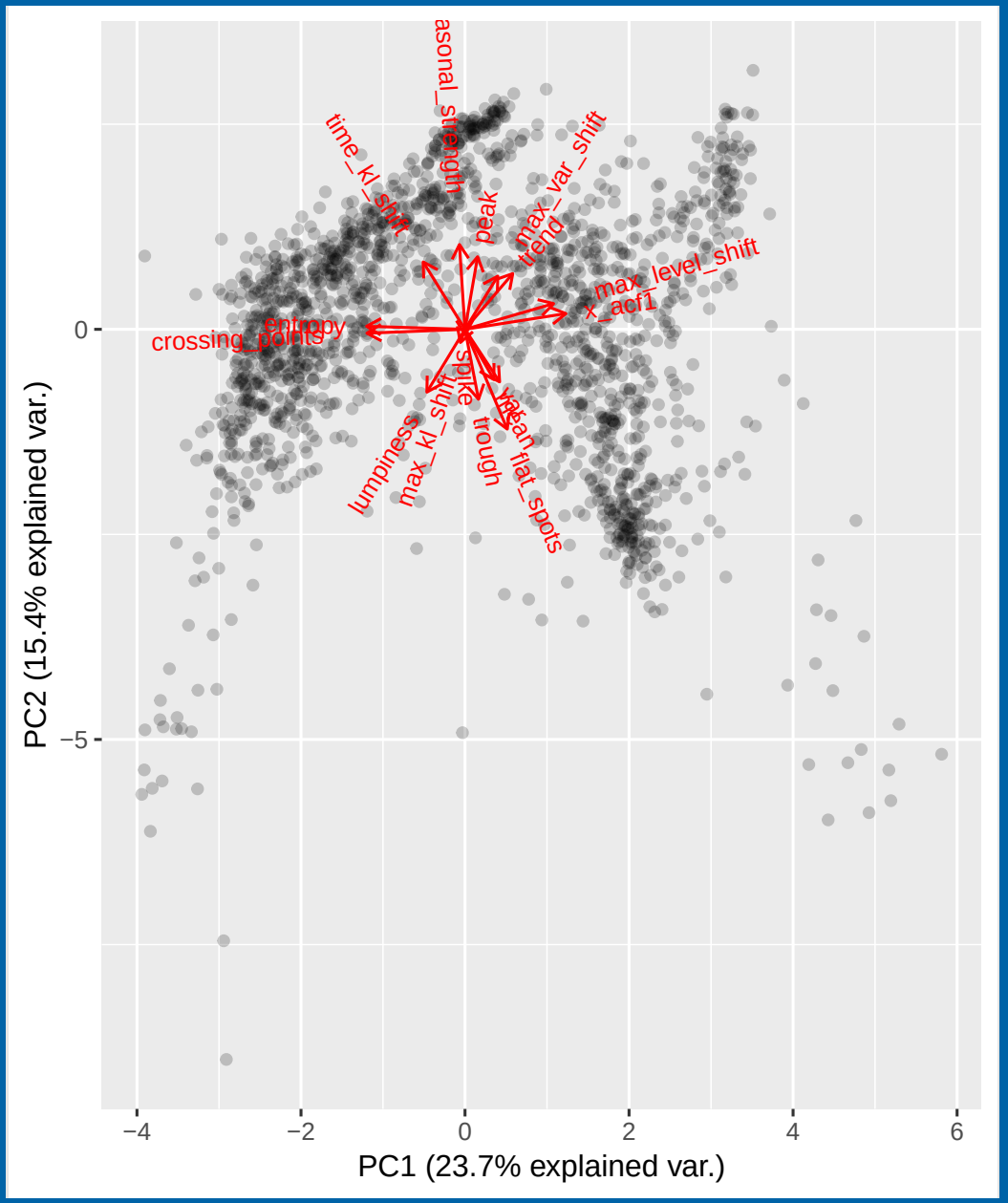
```
library(tsfeatures); library(tidyverse)
library(anomalous) # For data
yahoo <- cbind(dat0, dat1, dat2, dat3)
hwl <- bind_cols(
         tsfeatures(yahoo,
           c("acf_features","entropy","lumpiness",
             "flat_spots","crossing_points")),
         tsfeatures(yahoo,"stl_features",
                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

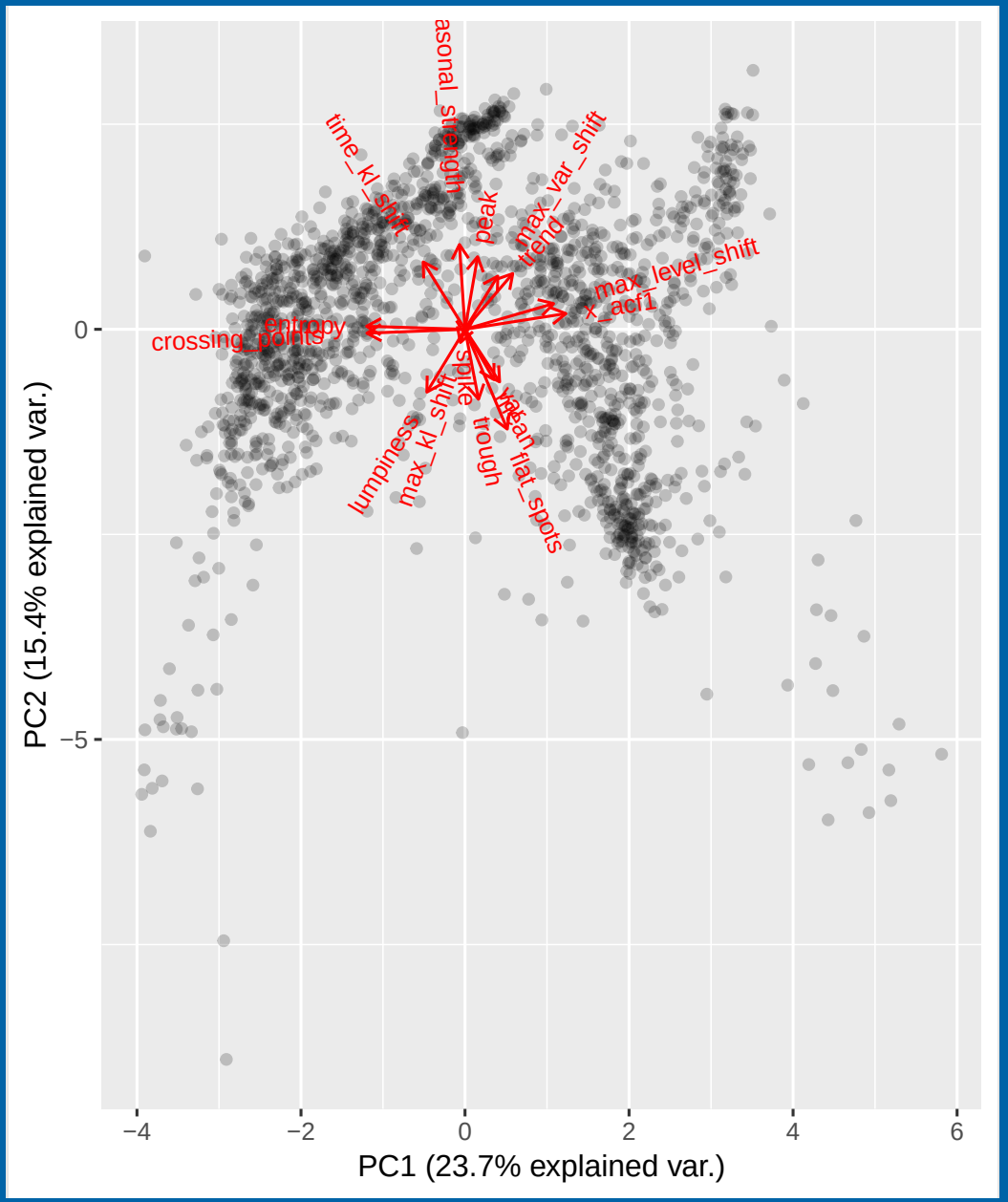
```
pc <- prcomp(na.omit(hwl), scale=TRUE)$x %>% as_tibble()
ggplot(pc, aes(x=PC1, y=PC2)) + geom_point()
```



Feature space



Feature space

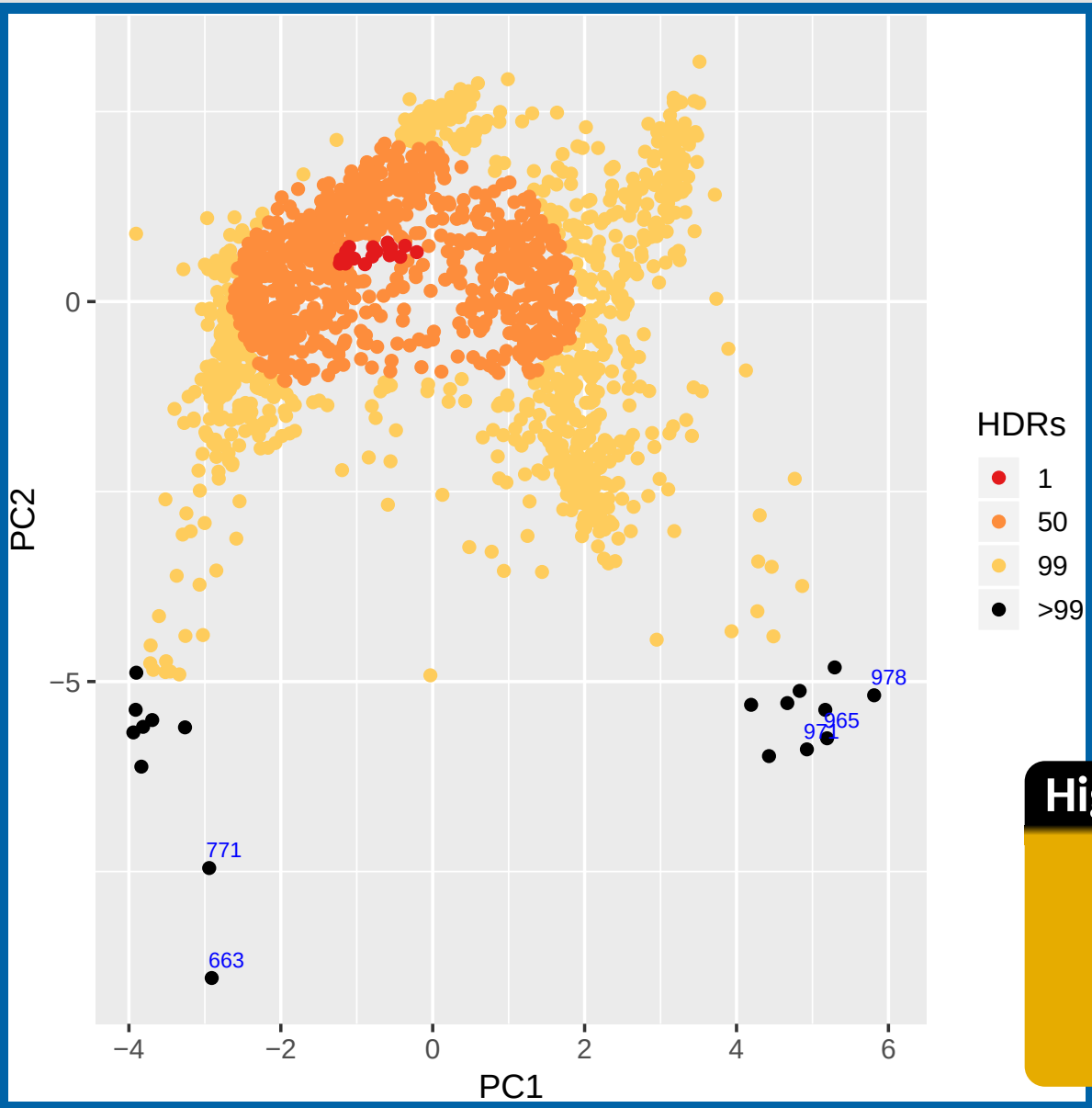


What is “anomalous”?

- 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)
```



Highest Density Regions

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

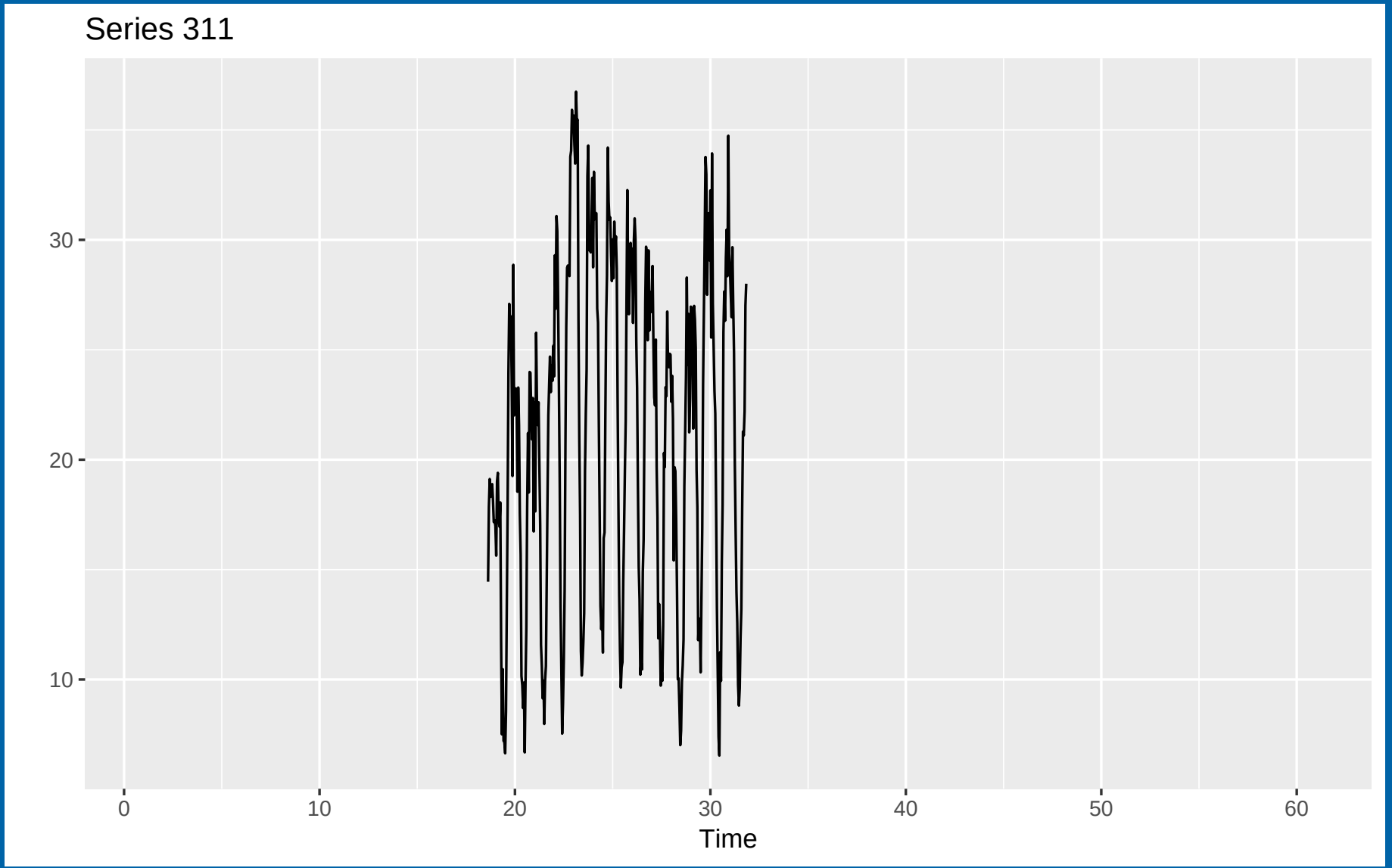
Stray algorithm


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

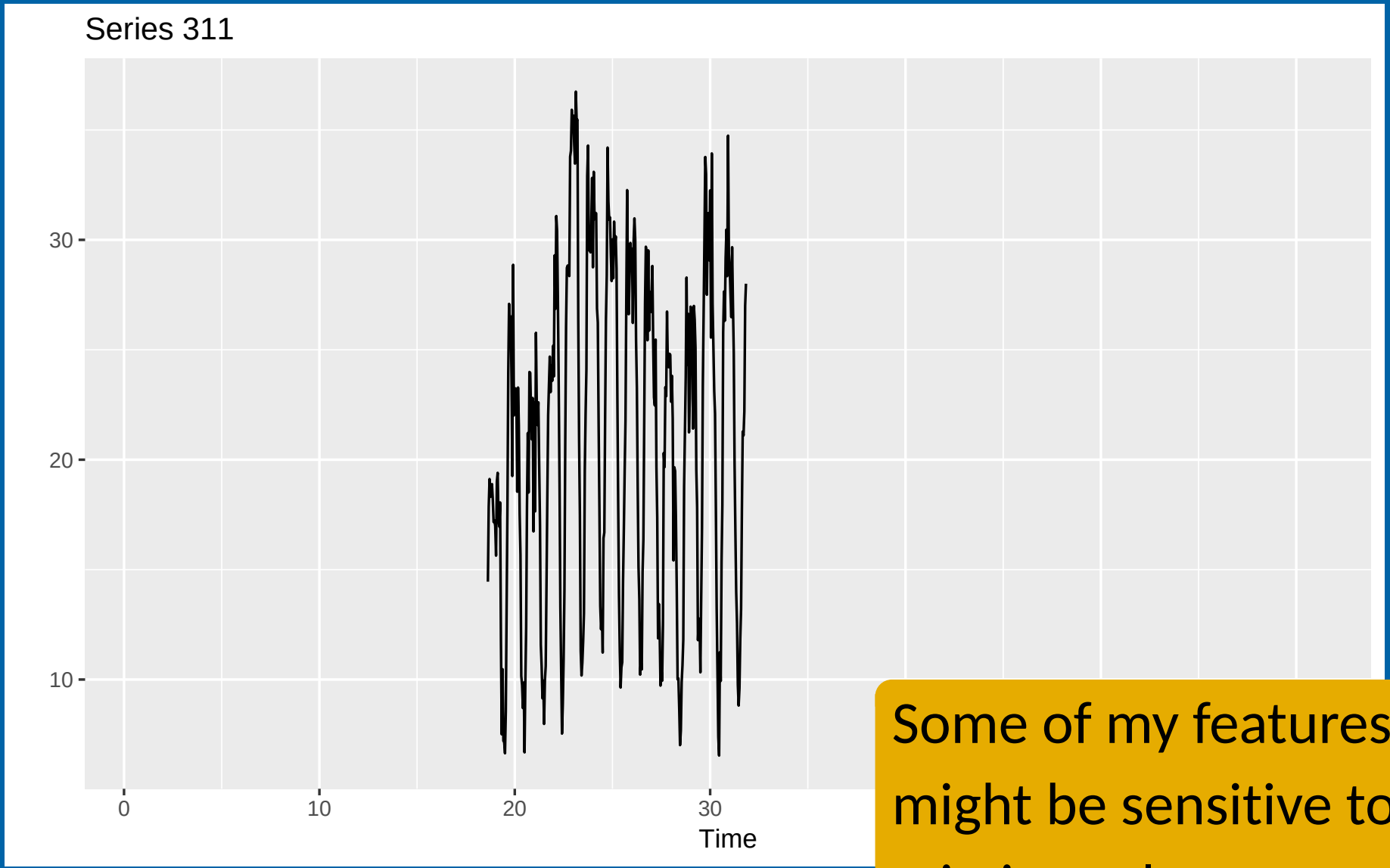
```
##   [1]  311  392  411  413  583  594  601  944
##  [12] 1151 1679 1719 1730 1737
```

- 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



Stray algorithm



Some of my features might be sensitive to missing values.

Outline

1 M3 forecasting competition

2 Yahoo server metrics

3 Irish smart metres

4 Packages

Irish smart metre data

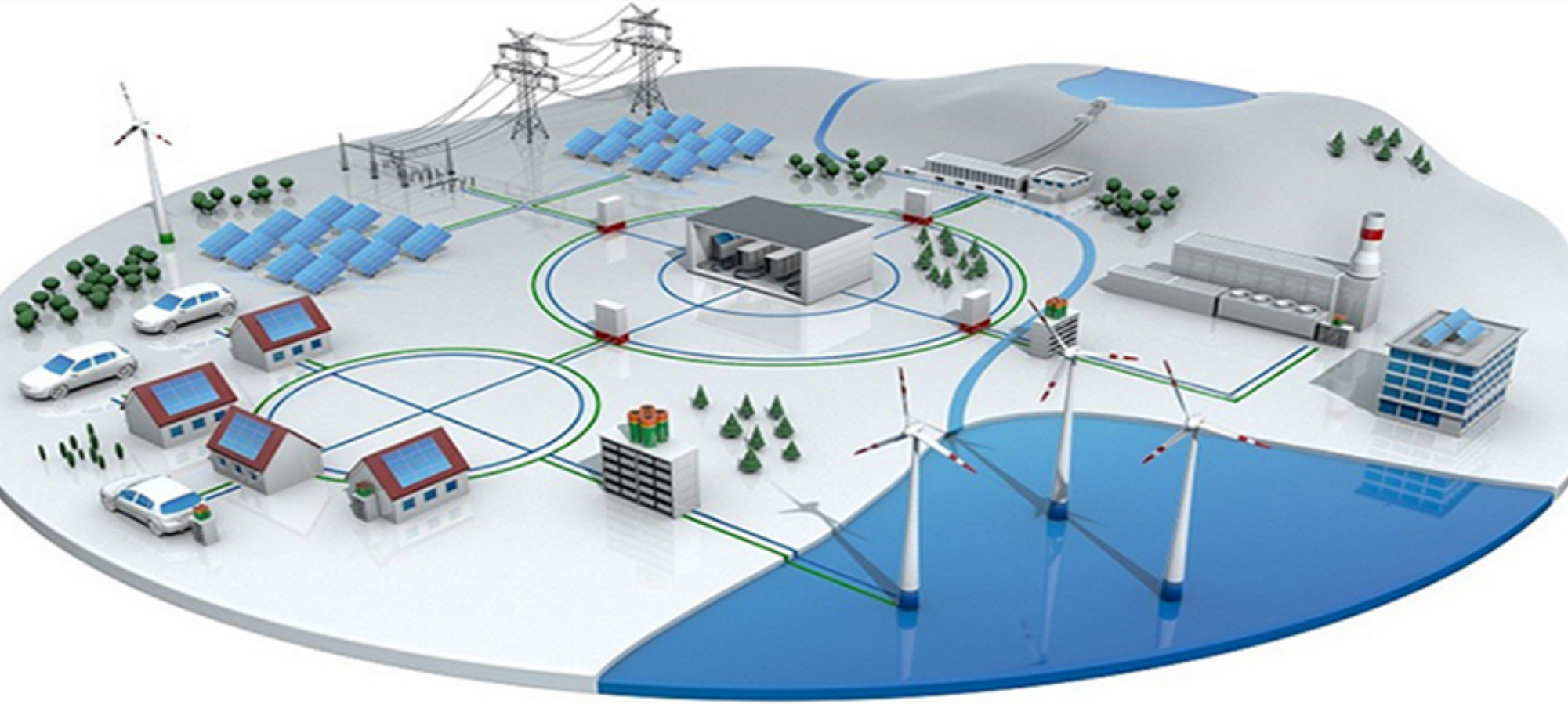
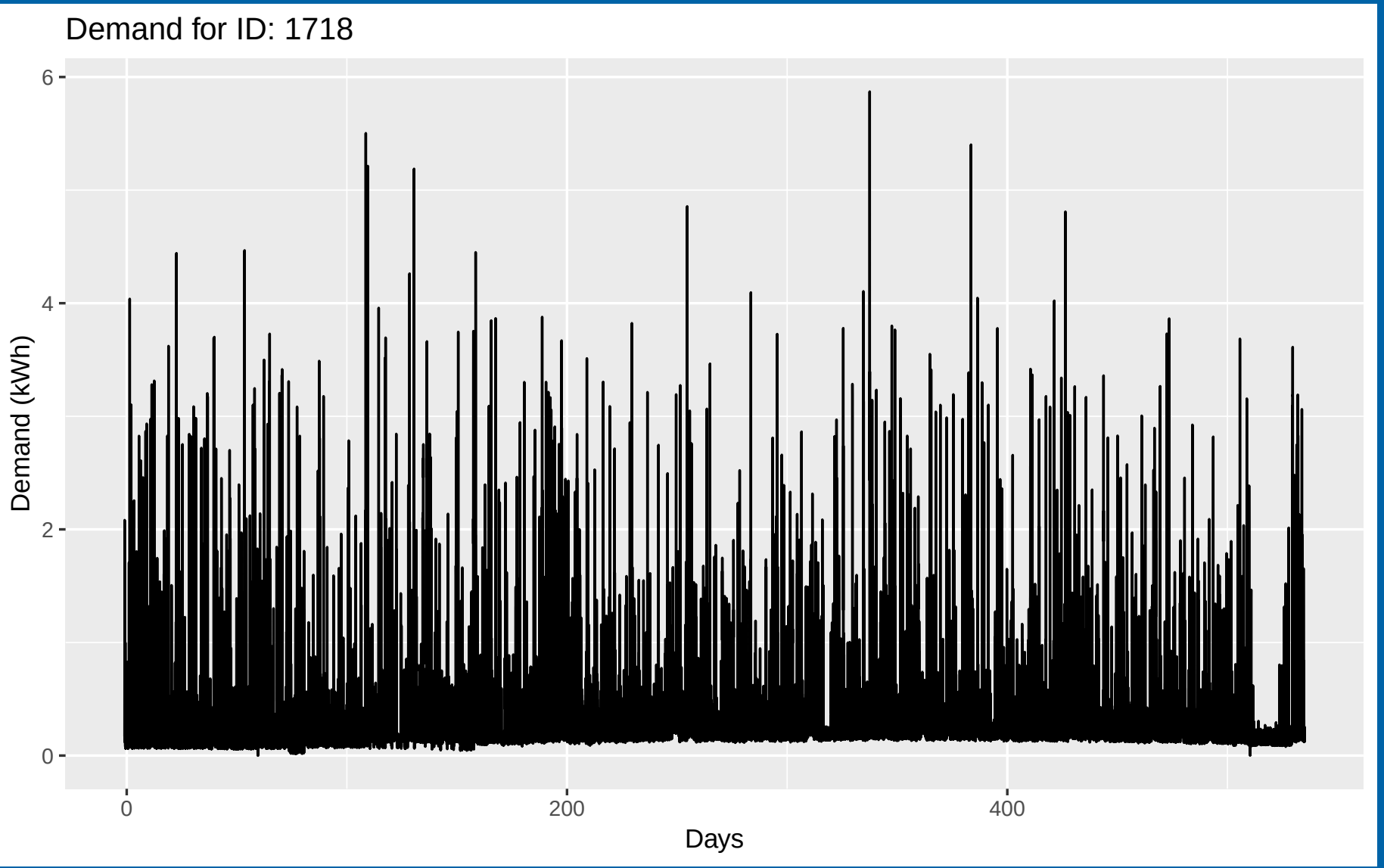


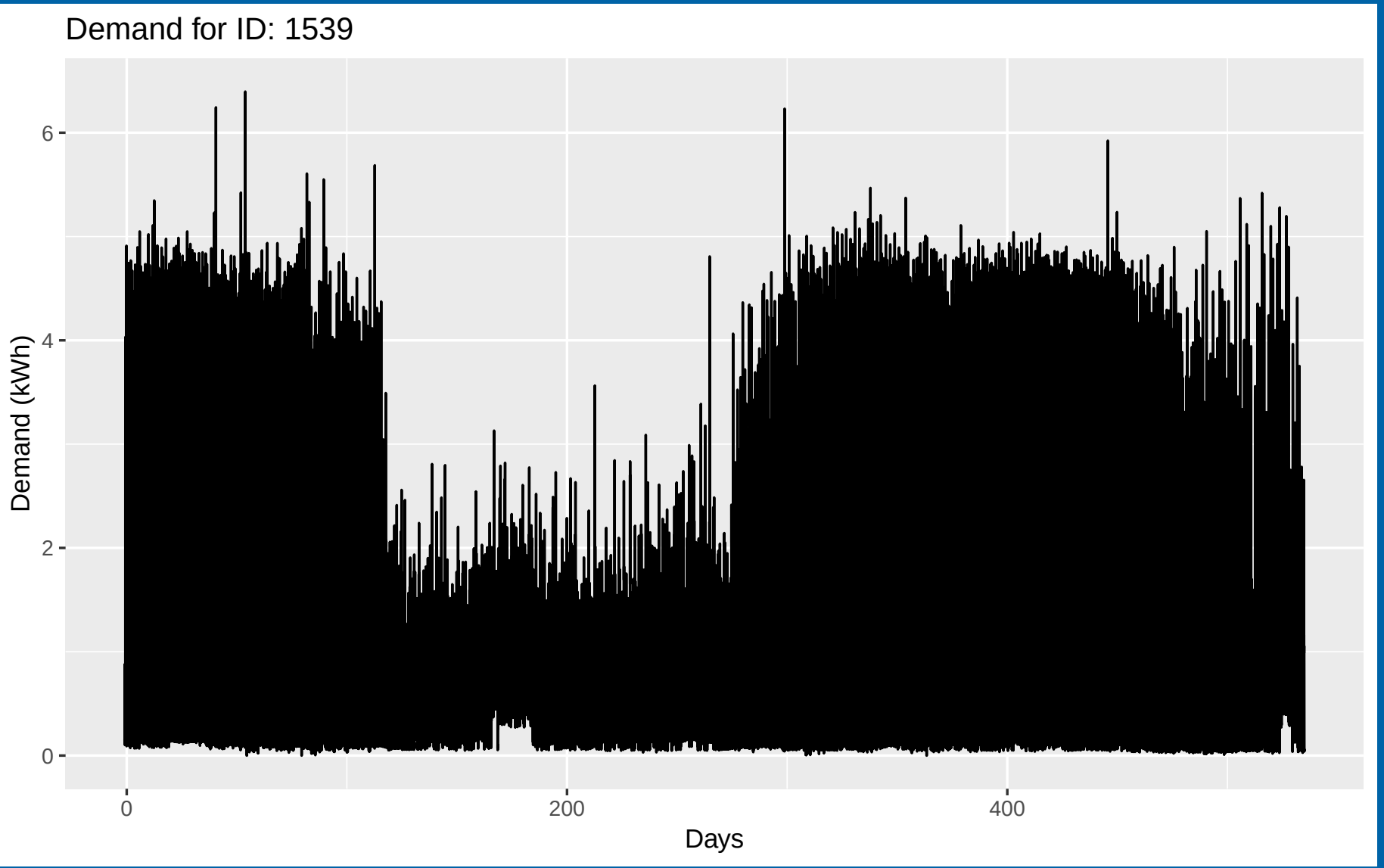
Figure: http://solutions.3m.com

- 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

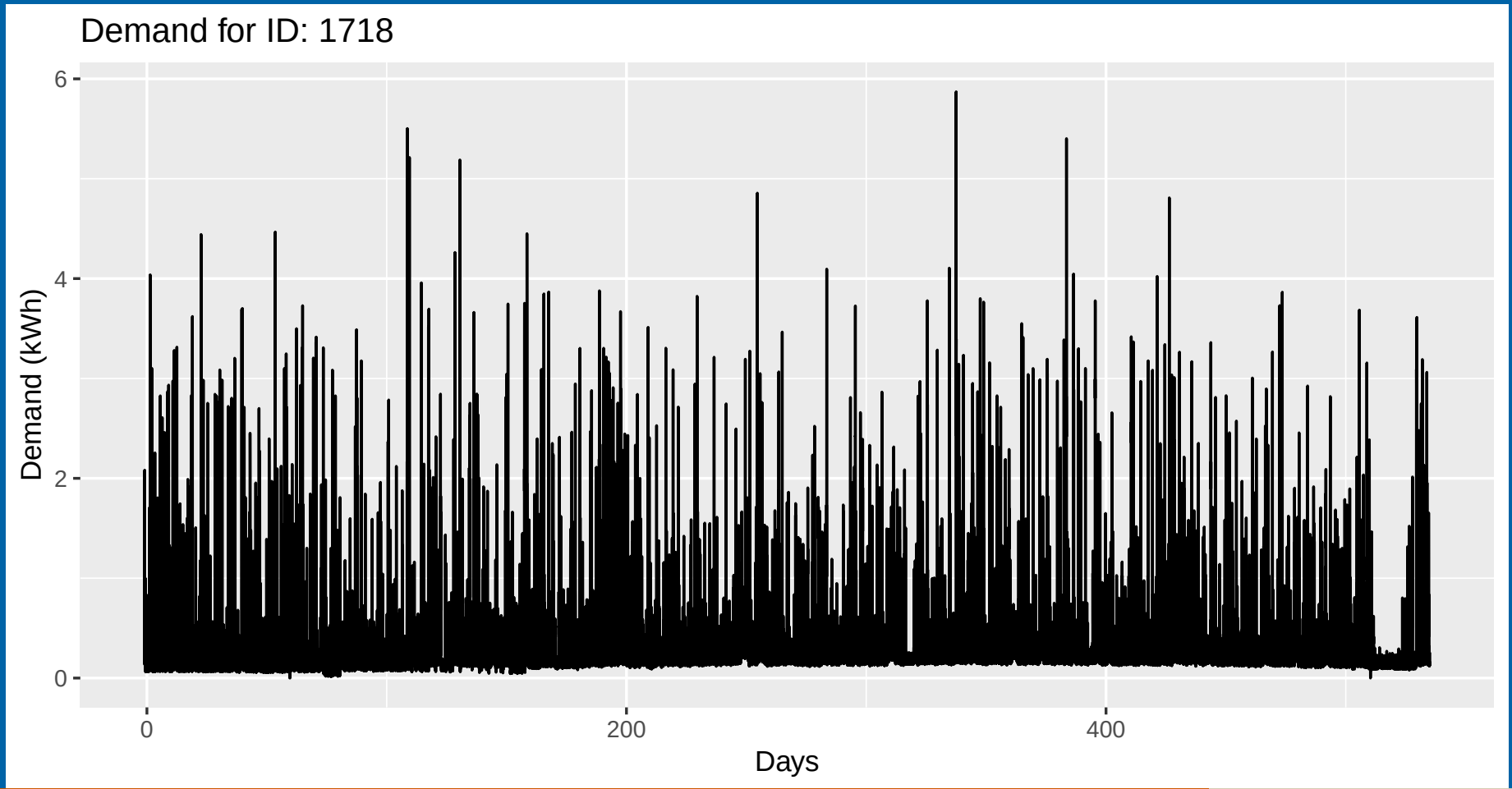


Irish smart metre data



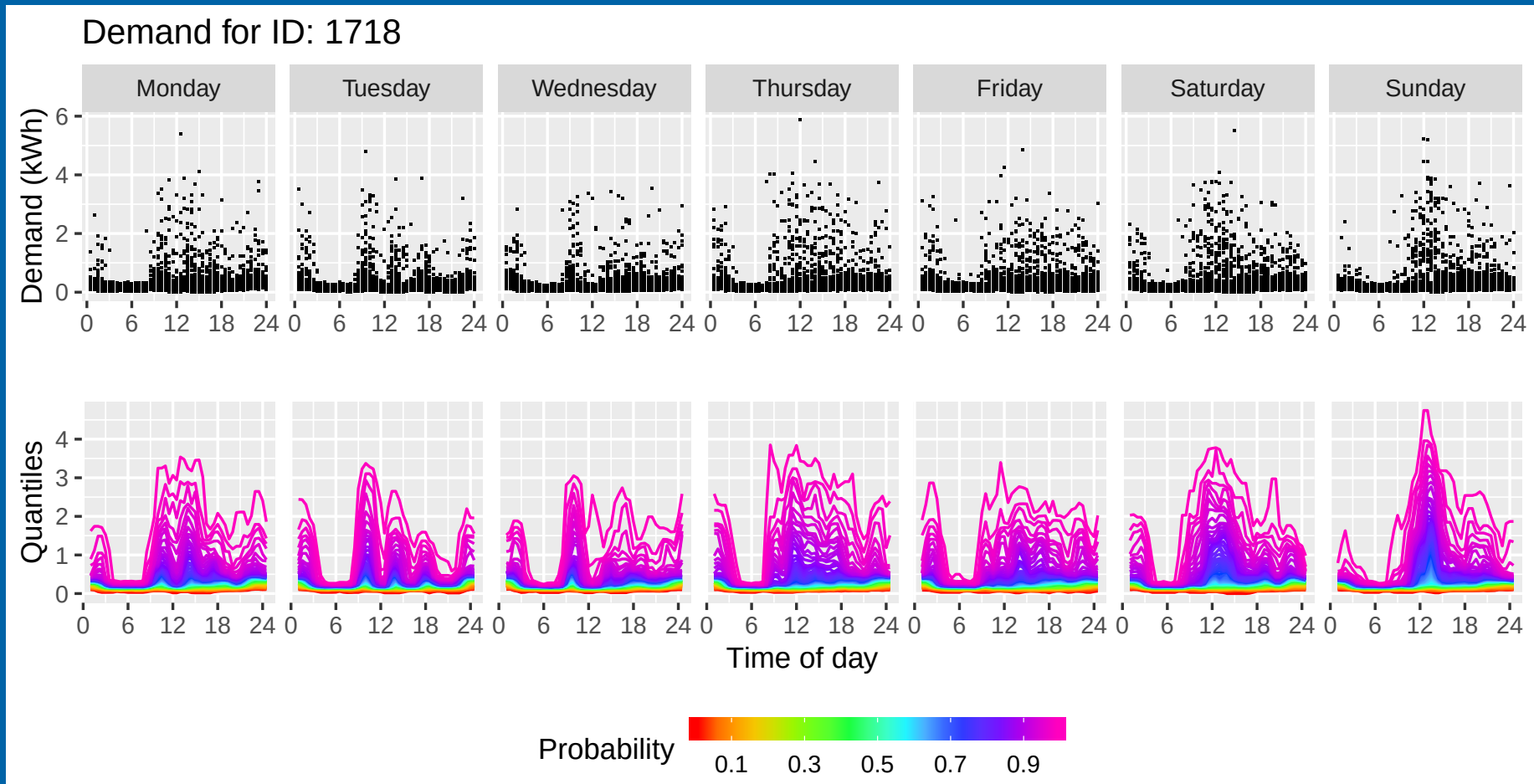
Quantiles as features

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



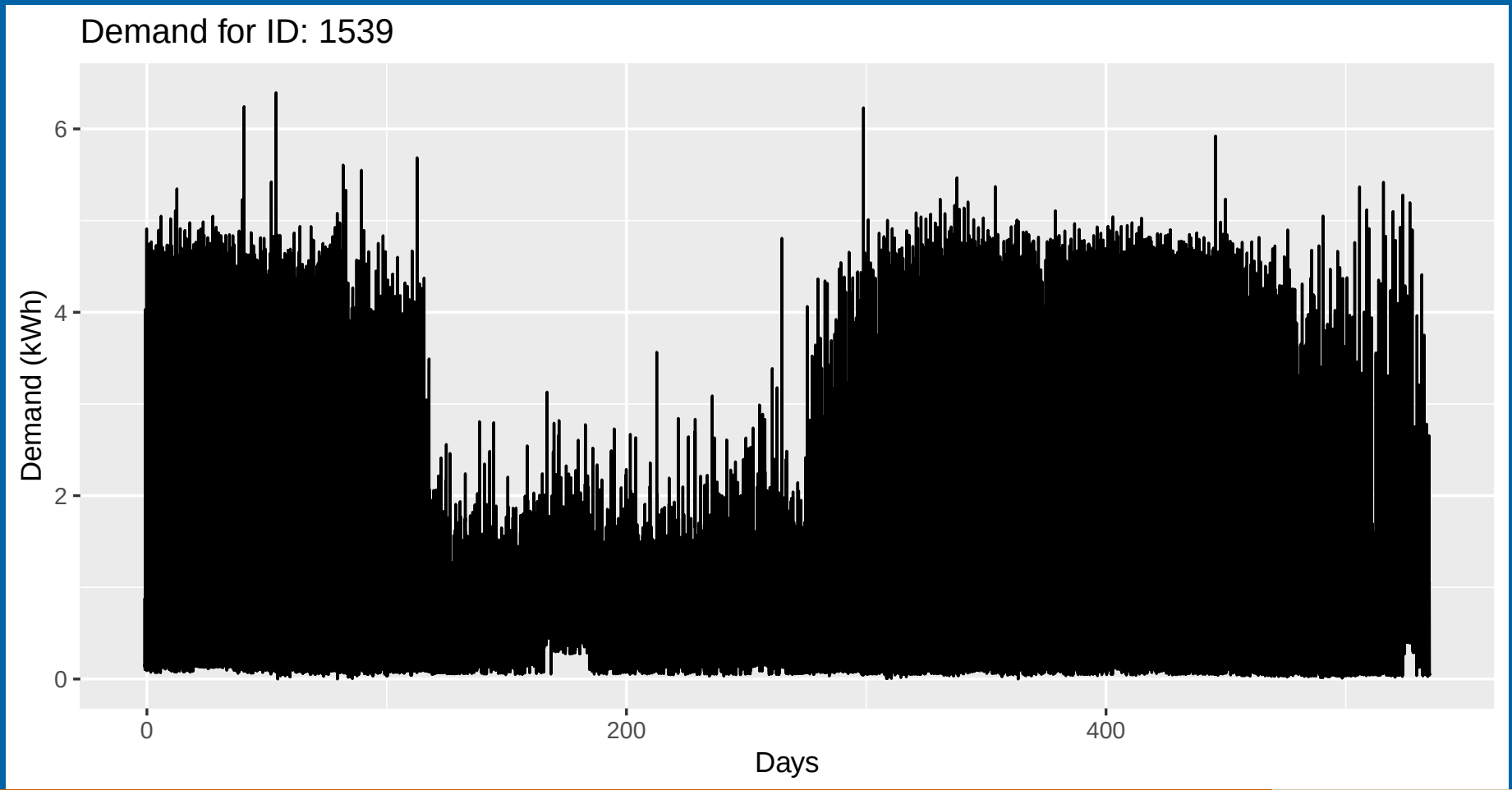
Quantiles as features

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



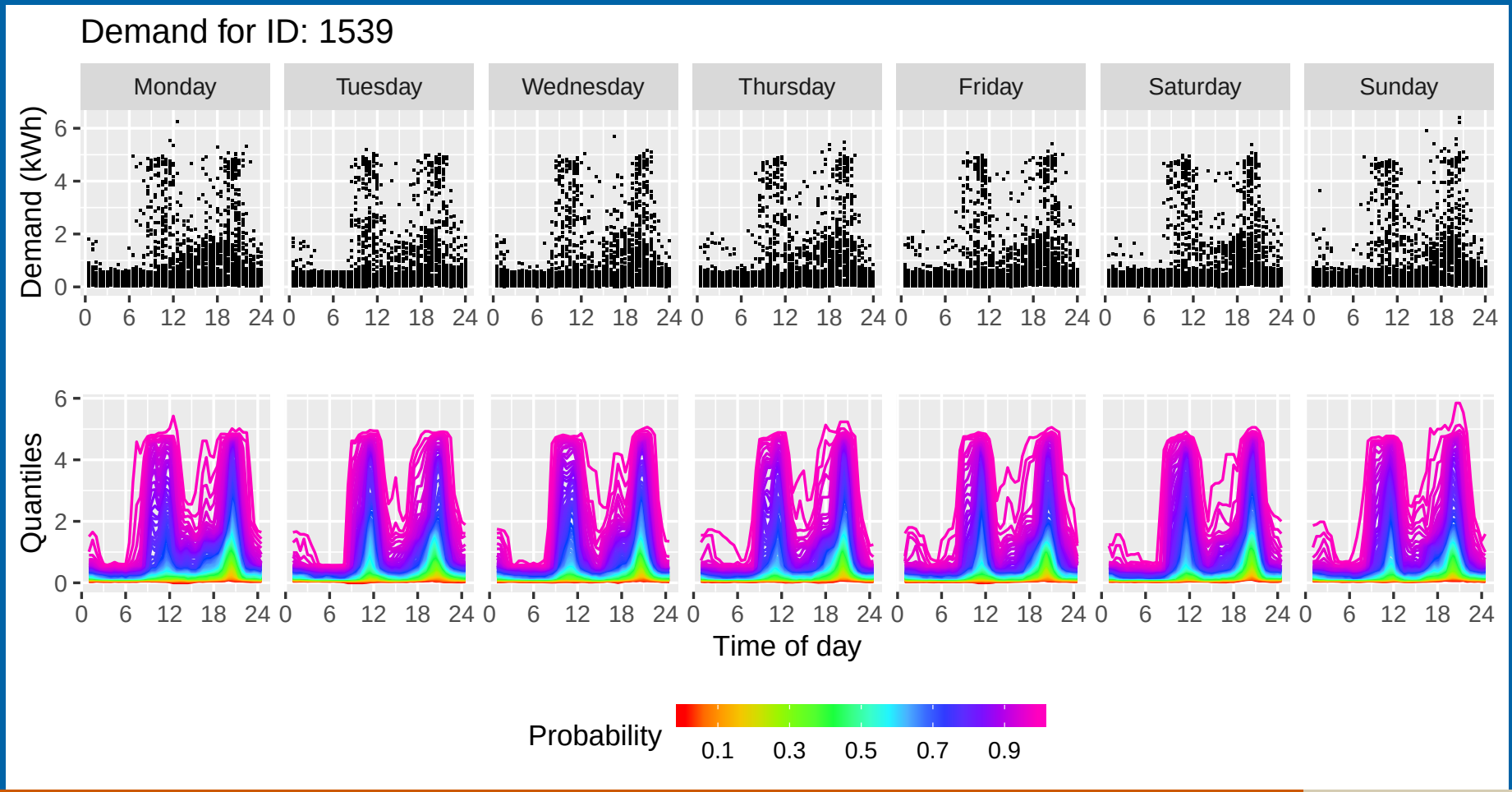
Quantiles as features

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

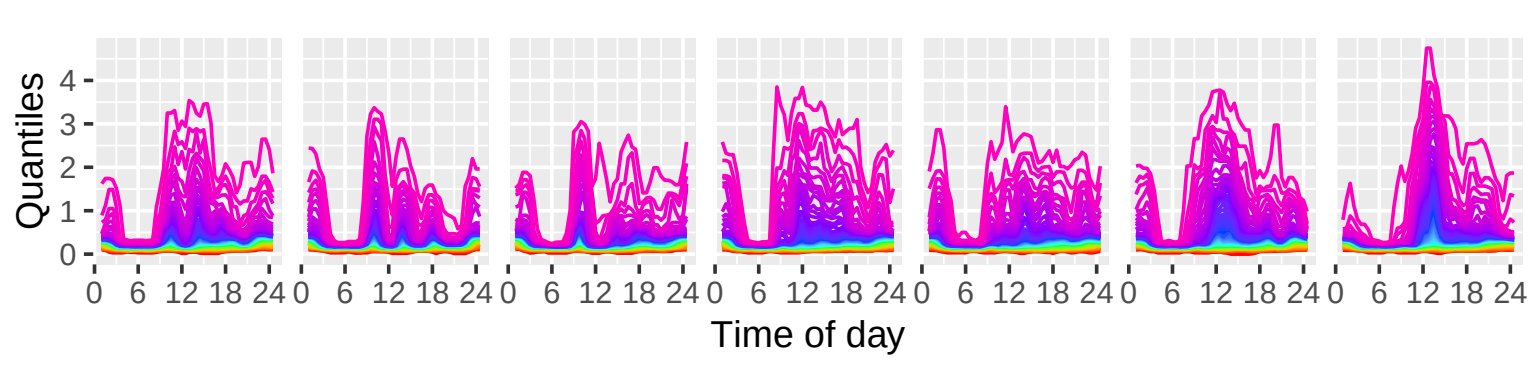


Quantiles as features

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

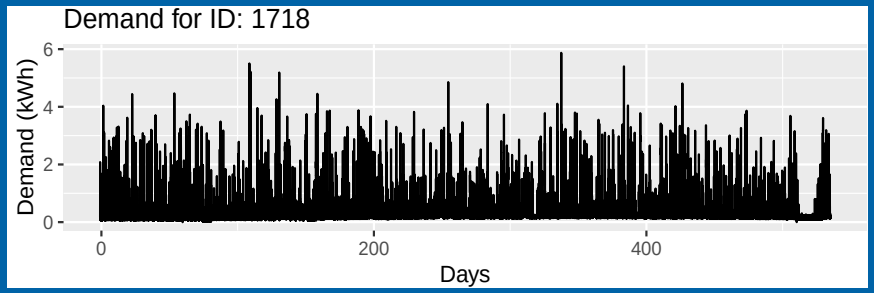


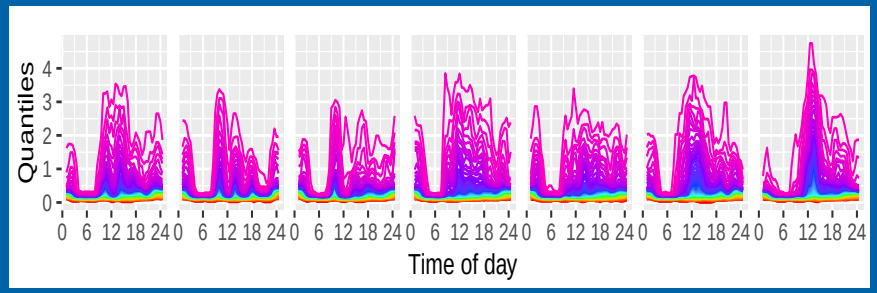
Quantiles as features



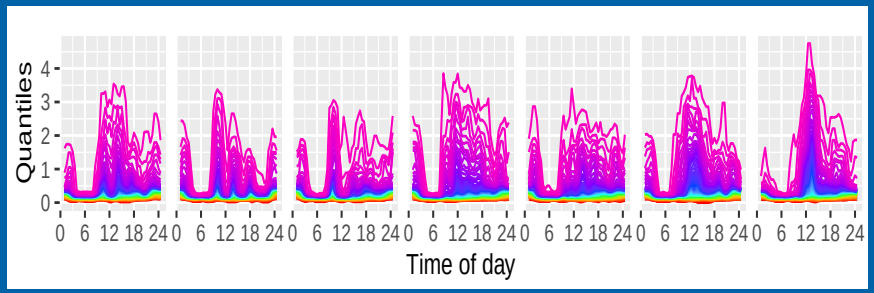
- Sample quantiles better than kernel density estimate:
  - presence of zeros
  - non-negative support
  - high skewness
- Avoids missing data issues and variation in series length
- Avoids timing of household events, holidays, etc.
- Allows clustering of households based on probabilistic behaviour rather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

Pairwise distances

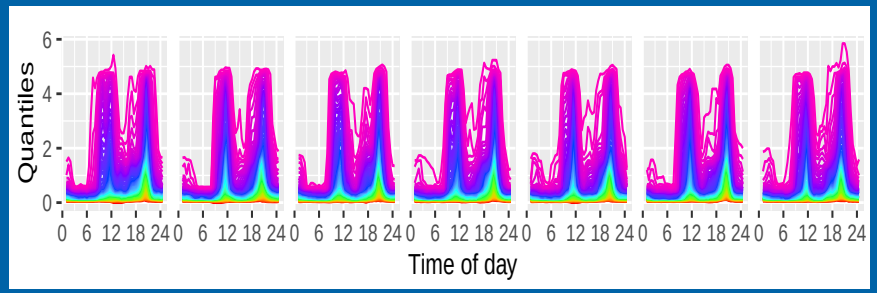




- The time series of 535×48 observations per household is mapped to a set of $7 \times 48 \times 99$ quantiles giving a bivariate surface for each household.
- Can we compute pairwise distances between all households?



$\leftarrow$? $\rightarrow$
Distance



Pairwise distances

Jensen-Shannon distance between two households

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

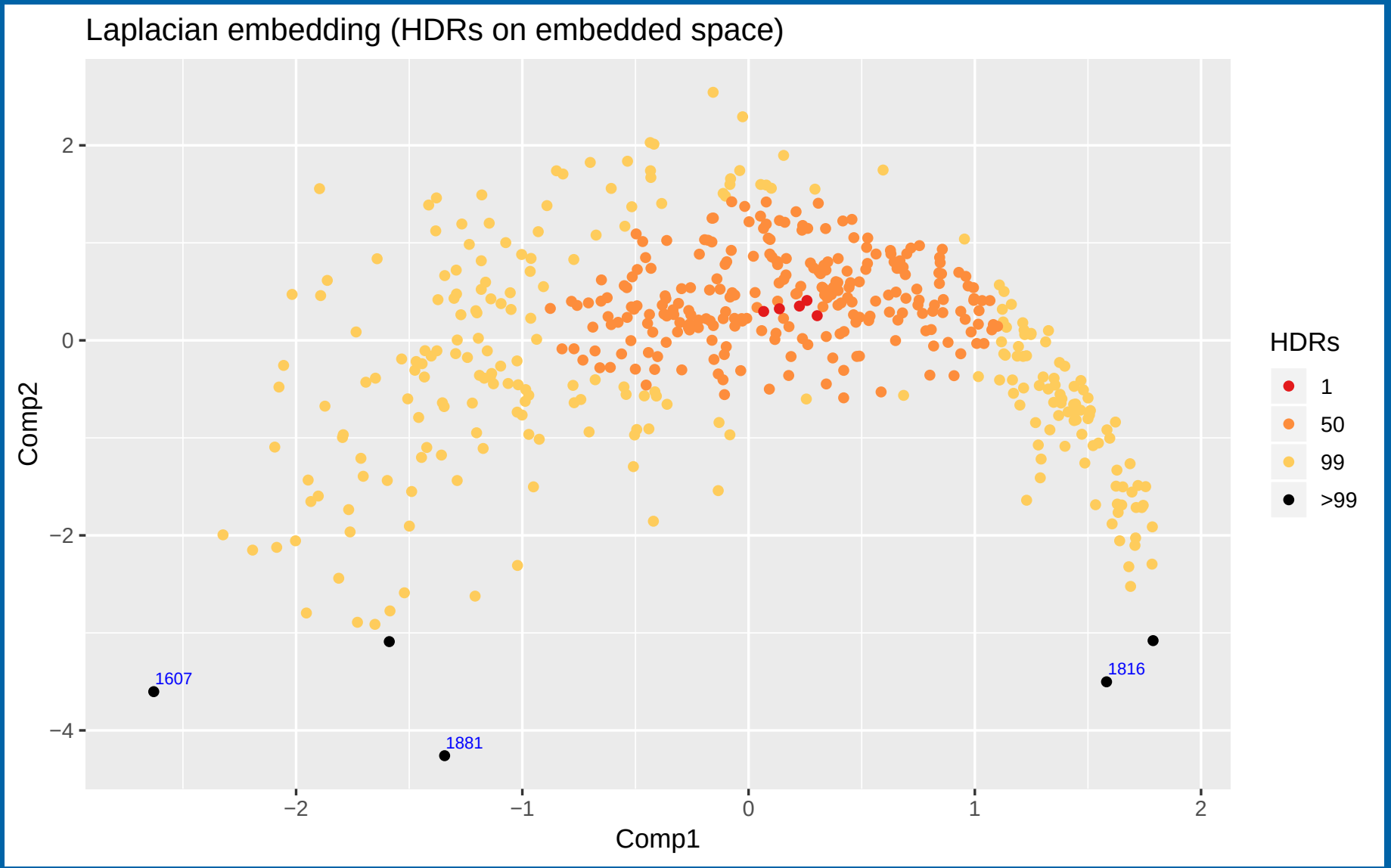
Similarity between two households

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

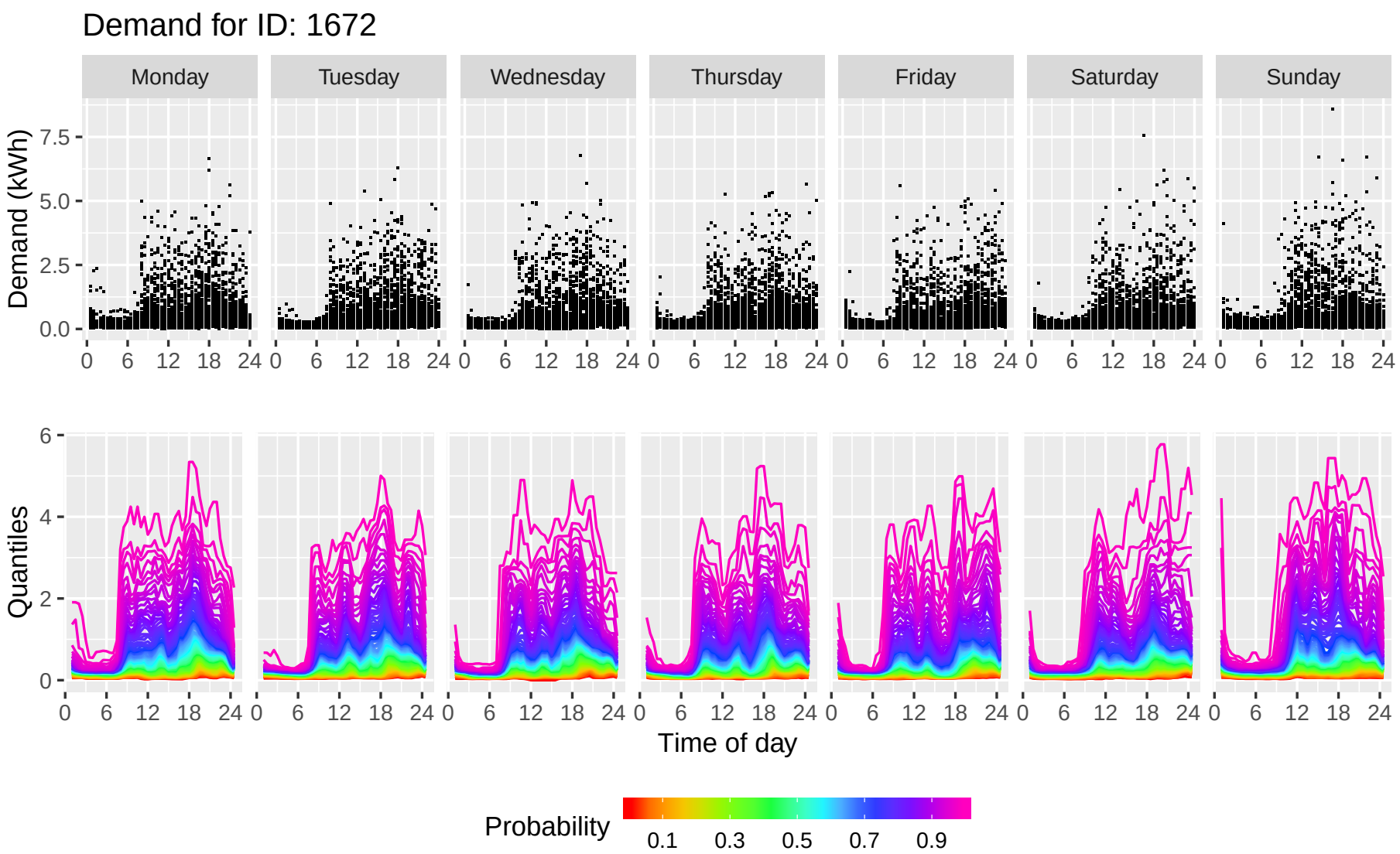
Laplacian eigenmaps

- 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

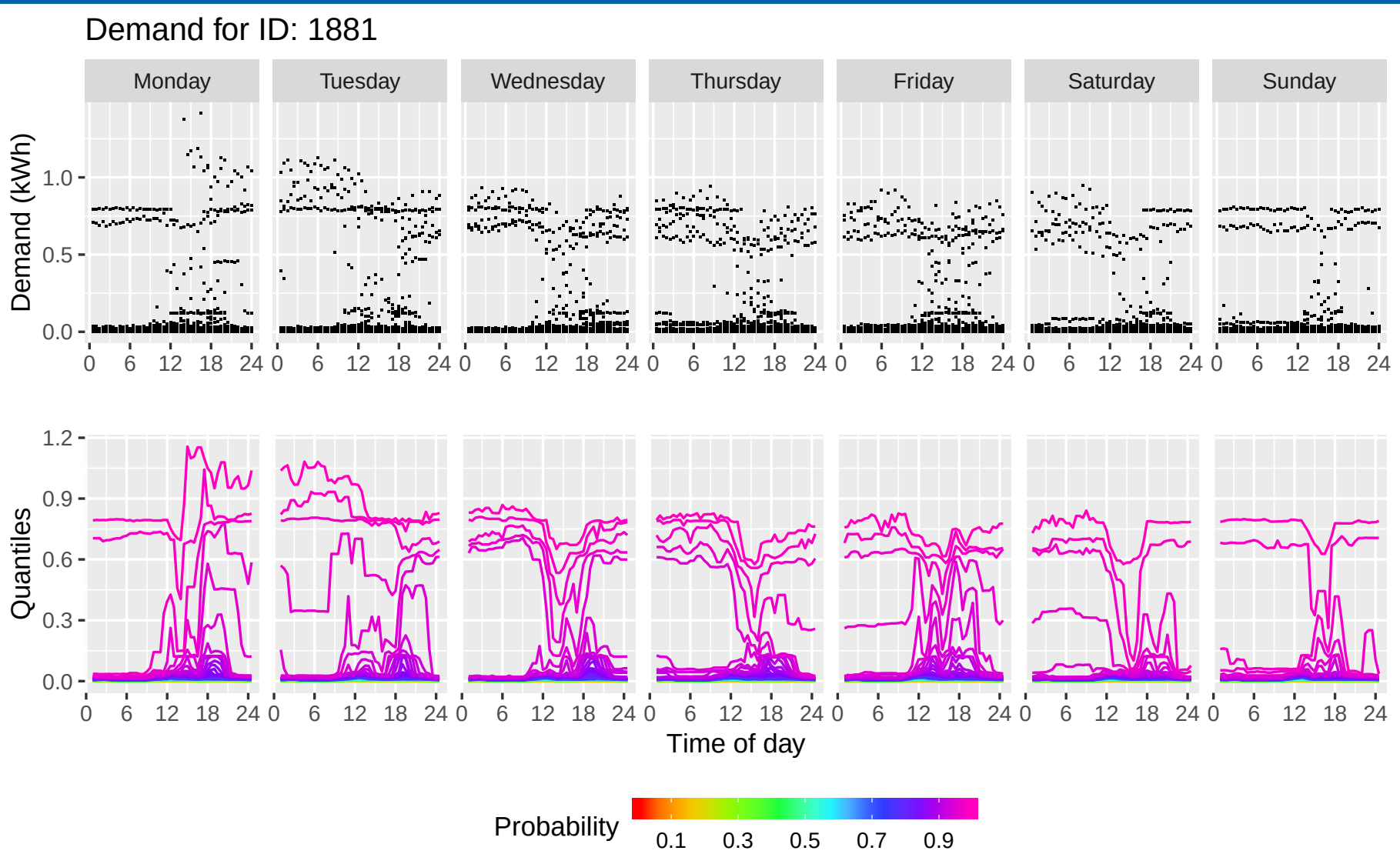
Outliers computed in embedded space:



Most typical household



Most anomalous household



Outline

1 M3 forecasting competition

2 Yahoo server metrics

3 Irish smart metres

4 Packages

Packages

- **tsfeatures**: compute time series features.
  github.com/robjhyndman/tsfeatures
- **hdrcde**: scatterplots with bivariate HDRs.
  CRAN | github.com/robjhyndman/hdrcde
- **stray**: finding outliers in high dimensions.
  github.com/pridiltal/stray
- **oddstream**: finding outliers in streaming data.
  github.com/pridiltal/oddstream
- **seer**: selecting forecasting model using features.
  github.com/thiyangt/seer
- **Mcomp**: M3 data.
  CRAN | github.com/robjhyndman/Mcomp
- **anomalous**: yahoo data.
  github.com/robjhyndman/anomalous

Acknowledgments

Earo Wang

Yanfei Kang

Dilini Talagala

Thiyanga Talagala

Pablo Montero-Manso

Mitchell O'Hara-Wild