## Time Series Encoding (encode)

This example shows how to transform a time series into fixed-size windows and train an autoencoder to learn a compact latent representation (p -> k) of these windows.

Prerequisites - R packages: daltoolbox, ggplot2 - Python with PyTorch accessible via reticulate (the backend is loaded by internal functions)

# Loading required packages  
library(daltoolbox)

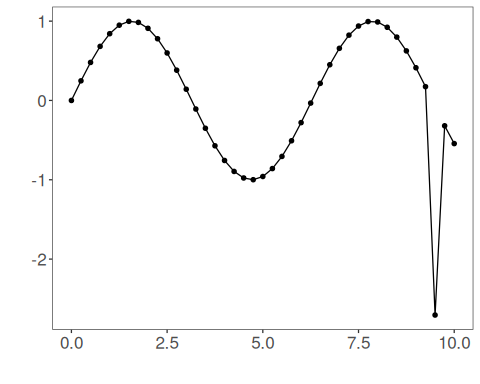
## Series for study

data(tsd)  
tsd$y[39] <- tsd$y[39] \* 6 # inject a synthetic outlier for illustration in the plot

sw\_size <- 5 # sliding window size (p)  
ts <- ts\_data(tsd$y, sw\_size) # convert the series into windows with p columns  
ts\_head(ts, 3) # view the first 3 windows

## t4 t3 t2 t1 t0  
## [1,] 0.0000000 0.2474040 0.4794255 0.6816388 0.8414710  
## [2,] 0.2474040 0.4794255 0.6816388 0.8414710 0.9489846  
## [3,] 0.4794255 0.6816388 0.8414710 0.9489846 0.9974950

library(ggplot2)  
plot\_ts(x = tsd$x, y = tsd$y) + # series plot with the outlier peak  
 theme(text = element\_text(size = 16))



## Data sampling

samp <- ts\_sample(ts, test\_size = 5) # hold out the last 5 windows for test  
train <- as.data.frame(samp$train)  
test <- as.data.frame(samp$test)

## Train the model

auto <- autoenc\_e(5, 3) # reduce from 5 -> 3 dimensions (p -> k)  
auto <- fit(auto, train)

## Encoding evaluation (train)

print(head(train)) # original windows (p columns)

## t4 t3 t2 t1 t0  
## 1 0.0000000 0.2474040 0.4794255 0.6816388 0.8414710  
## 2 0.2474040 0.4794255 0.6816388 0.8414710 0.9489846  
## 3 0.4794255 0.6816388 0.8414710 0.9489846 0.9974950  
## 4 0.6816388 0.8414710 0.9489846 0.9974950 0.9839859  
## 5 0.8414710 0.9489846 0.9974950 0.9839859 0.9092974  
## 6 0.9489846 0.9974950 0.9839859 0.9092974 0.7780732

result <- transform(auto, train) # encodings (k columns)  
print(head(result))

## [,1] [,2] [,3]  
## [1,] -0.03470621 -0.2017126 -0.7972909  
## [2,] 0.17020221 -0.4402158 -0.9277116  
## [3,] 0.37237298 -0.6695251 -0.9982935  
## [4,] 0.54281580 -0.8716210 -0.9945154  
## [5,] 0.66290563 -1.0294850 -0.9247539  
## [6,] 0.72029185 -1.1298401 -0.7931669

## Encoding of the test set

print(head(test))

## t4 t3 t2 t1 t0  
## 1 0.9893582 0.9226042 0.7984871 0.6247240 0.4121185  
## 2 0.9226042 0.7984871 0.6247240 0.4121185 0.1738895  
## 3 0.7984871 0.6247240 0.4121185 0.1738895 -2.7054403  
## 4 0.6247240 0.4121185 0.1738895 -2.7054403 -0.3195192  
## 5 0.4121185 0.1738895 -2.7054403 -0.3195192 -0.5440211

result <- transform(auto, test)  
print(head(result))

## [,1] [,2] [,3]  
## [1,] 0.65379596 -1.14049983 -0.4247846  
## [2,] 0.53606588 -1.06513941 -0.1803204  
## [3,] 0.08953688 -1.99327374 1.7490469  
## [4,] -0.79318631 0.05811567 1.6759746  
## [5,] -0.78187072 -0.28948379 1.3296857