Second Group Assignment

This is your second of three homework assignments. The value of this assignment is 8%. Each group will receive an extended version of the electricity consumption dataset to be analyzed using the R language and environment for statistical computing and graphics.

Please complete the tasks described below and submit an electronic copy of your solution, one per group, through CourSys by end of Sunday, 20 February 2022.

1. Feature scaling is an essential preprocessing technique of input data for machine learning algorithms and beyond. Two widely used feature scaling techniques are *normalization* and *standardization*. For each of the following questions explain clearly and concisely: (I) What is feature scaling? - (II) Why scaling features of a dataset is necessary? - (III) What does normalization and standardization do to the data and the noise? Your answers to all three questions should not exceed 2 pages in total but provide technical descriptions including the use of mathematical notation.

   (Please check the information available online.)

2. For the dataset assigned to your group, complete the following tasks. Slice the dataset into as many complete weeks (Monday-Sunday) as possible. Consider the *Moving Average* and apply this method on **Global_intensity** only to reduce the relative effect of single point anomalies (which are oftentimes just noise). To do this, compute the moving average for each complete week in the dataset using a fixed size **moving time window** of 7-10 consecutive observations. The outcome will be a 'smoothened' version of the input week, referred to as *smoothened week #n* (for n = 1, 2, …).

   Now, create a new time series by computing the average value of all observations over each of the n smoothened weeks, referred to as the *average smoothened week*; that is, the value for time t of the new time series is obtained as the average value at time t calculated over all n smoothened weeks. The *average smoothened week* is used here as an approximation of what a 'normal week' looks like for the time period of the dataset.

   Identify the most and the least anomalous weeks among the n weeks (hint: compare each smoothened week to the *average smoothened week* and quantify the deviation in terms of a meaningful numerical score allowing you to rank all n weeks according to how closely they resemble the average week. Provide a brief rational for choosing your scoring.

   Represent all anomaly scores in a table and plot the smoothened versions of the most and the least anomalous weeks against the average smoothened week.

There are a number of packages and functions you can use in order to calculate the moving average over a given univariate time series. Below are a few suggestions.

Use simple moving average, **SMA()**, in the TTR package.
https://cran.r-project.org/web/packages/TTR/TTR.pdf#page40

Use rolling mean, **rollmean()**, in the zoo package.
https://cran.r-project.org/web/packages/zoo/zoo.pdf#page49

Use moving average, **MA()**, in the forecast package.
https://cran.r-project.org/web/packages/forecast/forecast.pdf#page87

3. Given a Hidden Markov Model, or HMM, as defined in the tutorial introduction by Rabiner, there are three basic problems of interest to be solved for a model used in real-world applications. Based on your understanding from reading the tutorial introduction clearly and concisely explain in plain English words the meaning of Problem 1 for detection of anomalous patterns in stream data from a continuously operating system, such as a supervisory control system, i.e. what does $P(O \mid \lambda)$ refer to in terms of intrusion detection in this application context.

   Problem 1: Given the observation sequence $O = O_1, O_2, \ldots, O_T$ and a model $\lambda = (A, B, \pi)$, how do we **efficiently** compute $P(O \mid \lambda)$, the probability of the observation sequence given the model?

   Limit your explanations to not more than one 1/2 pages.

Please submit a short report for your solution and the R code through CourSys by FEB 20th.

Thank you!