Application of Functional Clustering Methods to Climate Data

2023년 8월 29일

서울대학교 수리과학부 김수현

Table of Contents

1

Introduction

1 Overview

- Due to the infinite-dimensional nature of function spaces, prevalent clustering methods for Euclidean data cannot be directly utilized to cluster functional data.
- This project aims to examine and compare various functional clustering methods that effectively handles problems that arises from the nature of functional data.
- In particular, we will discuss different algorithms such as FunFEM and FADPclust and apply it to climate data to see their performance.

2

Functional Clustering Methods

Functional Data
FunFEM / FADPclust / Curve Alignment / fPCA
Evaluation Metrics

2 Overview of Functional Clustering Methods

The following four functional clustering methods were mainly used in this project:

1. **FunFEM**: A functional mixture model utilizing the Fisher-EM algorithm.
2. **FADPclust**: A multivariate functional data clustering method using adaptive density peak detection.
3. **Clustering with curve alignment:** An algorithm that jointly aligns and clusters curves, using existing clustering methods such as K-means and hierarchical clustering.
4. **Functional PCA:** Applying existing vector-valued clustering methods to fPCA scores.

The following evaluation metrics were used to determine the optimal clustering method:

- **Silhouette Coefficients**
- **Dunn Index**

2 Transformation of Observed Data

- Let $\{x_1, \cdots, x_n\}$ denote the observed curves, all belonging to $L_2[0, T]$.
- In practice, the functional expressions of the observed curves are not known, and we only have access to the discrete observations $x_{ij} = x_i(t_{is})$ at a finite set of ordered times $\{t_{is}: s = 1, \cdots, m_i\}$.
- A common way to reconstruct the functional forms is to assume that the curves belong to a finite-dimensional space spanned by a basis of functions $\{\psi_1, \cdots, \psi_p\}$.
- The basis coefficients of each observed curve can be estimated by least squares:

$$\widehat{\gamma_i} = (\Theta^T \Theta)^{-1} \Theta_i X_i^{obs}$$

  with $\Theta_i = \left(\psi_j(t_{is})\right)$ and $X_i^{obs} = (x_i^{obs}(t_{i1}), \cdots, x_i^{obs}(t_{im}))$.
- B-splines and Fourier basis functions are widely used for non-periodic and periodic data, respectively.

FunFEM

- FunFEM is a discriminative functional mixture model that utilizes the Fisher-EM algorithm, which iteratively updates (1) the most discriminative subspace of the original function space (Fisher) and (2) model parameters for the assumed Gaussian distribution (EM).
- Our goal is to cluster the observed curves $\{x_1, \cdots, x_n\}$ into K homogeneous groups.

2 FunFEM

Model Assumptions

- Known fact: A subspace of $d = K - 1$ dimensions is sufficient to discriminate K groups (Fisher (1936), Fukunaga (1990)).
- We aim to find the most discriminative subspace of $d = K - 1$ $(d \leq p)$ dimensions, spanned by a basis of d basis functions $\{\phi_j\}_{j=1,\cdots,d}$.
- The basis $\{\phi_j\}_{j=1,\cdots,d}$ is obtained from the original basis $\{\psi_j\}_{j=1,\cdots,p}$ through a linear transformation $\phi_j = \sum_{l=1}^{p} u_{jl}\psi_l$ such that the $p \times d$ matrix $U = (u_{jl})$ is orthogonal.
- Let Γ and Λ be the coefficient expansion matrices of the observed curves $\{x_1, \cdots, x_n\}$ on the bases $\{\psi_j\}_{j=1,\cdots,p}$ and $\{\phi_j\}_{j=1,\cdots,d}$, respectively. Then, Γ and Λ are linked by $\Gamma = U\Lambda + \epsilon$, where $\epsilon \in \mathbb{R}^p$ is an independent and random noise term.

Model Assumptions - Gaussian density assumption

- Let $Z = (Z_1, \cdots, Z_k) \in \{0,1\}^K$ be the latent variable indicating the assignment of clusters.
- Conditioned on Z, Λ is assumed to be normally distributed:

$$\Lambda_{|Z=k} \sim N(\mu_k, \Sigma_k).$$

- ϵ is also assumed to be normally distributed: $\epsilon \sim N(0, \Xi)$.
- The noise covariance matrix Ξ is assumed that $\Delta_k = cov(W^T\Gamma \,|\, Z = k) = W^T\Sigma_k W$ has the following form:

$$\Delta_k = \begin{pmatrix} \Sigma_k & \mathbf{0} & \\ \mathbf{0} & \begin{matrix} \beta & & 0 \\ & \ddots & \\ 0 & & \beta \end{matrix} \end{pmatrix} \begin{matrix} \}\ d \\ \}\ p-d \end{matrix}$$

with $W = [U, V]$, where V is the orthogonal complement of U.

2 FunFEM

Model Assumptions – Mixture Model

- With these distributional assumptions, the marginal distribution of Γ is a mixture of Gaussians:

$$p(\gamma) = \sum_{k=1}^{K} \pi_k \phi(\gamma; U\mu_k, U^T \Sigma_k U + \Xi)$$

where ϕ is the standard Gaussian density function, and $\pi_k = P(Z = k)$ is the prior probability of the k-th group.

Model Inference

- A classical solution for model inference is to use the EM algorithm.
- In this context, however, we cannot directly apply the EM algorithm due to the nature of the functional subspace F.
- We estimate the most discriminative subspace F and the corresponding matrix U separately, and subsequently apply the EM algorithm on the subspace.
- The FunFEM algorithm alternates over the three following steps: the F-step, the M-step and the E-step.

2 FunFEM

Model Inference – The F-step

- Assume that the posterior probabilities $t_{ik}^{(q)} = \mathrm{E}[\mathrm{z_{ik}} \mid \gamma_i, \theta^{(q-1)}]$ are already estimated in the E-step estimation of iteration q-1.
- The F-step aims to determine the orientation matrix U, conditioned on the posterior probabilities, in which the K clusters are best separated.
- Fisher criterion: We look for a subspace that minimizes the variance within the groups and maximizes the variance between groups.

2 FunFEM

Model Inference – The F-step

- The Fisher criterion looks for the discriminative function $u \in L_2[0,T]$, where u is a solution of

$$\max_u \frac{Var(E[\Phi(X)|Z])}{Var(\Phi(X))}$$

  where $\Phi(X) = \int_0^T X(t)u(t)dt$ is the projection of X on the discriminative function u.
- Obtaining the complete set of basis functions u can be done in a PCA-like manner, by consecutively solving a constrained eigenproblem.

Model Inference – The M-step

- Conditioned on the orientation matrix $U^{(q)}$ obtained in the previous step, the M-step aims to maximize the conditional expectation of the complete data log-likelihood

$$Q(\theta, \theta^{(q-1)}) = E\left[\ell(\theta; \Gamma, z_1, \cdots, z_n \mid \Gamma, \theta^{(q-1)}\right],$$

where $\theta = (\pi_k, \mu_k, \Sigma_k, \beta)_{k=1,\cdots,K}$.

- The model parameters are updated as follows:

  - $\pi_k^{(q)} = n_k^{(q-1)}/n,$
  - $\mu_k^{(q)} = \frac{1}{n_k^{(q-1)}} \sum_{i=1}^{n} t_{ik}^{(q-1)} U^{(q)t} \gamma_i,$
  - $\Sigma_k^{(q)} = U^{(q)t} C_k^{(q)} U^{(q)},$
  - $\beta^{(q)} = (\text{trace}(C^{(q)}) - \sum_{j=1}^{d} u_j^{(q)t} C^{(q)} u_j^{(q)})/(p-d)$

where $C_k = \frac{1}{n_k^{(q-1)}} \sum_{i=1}^{n} t_{ik}^{(q-1)} \left(\gamma_i - \mu_i^{(q-1)}\right)\left(\gamma_i - \mu_i^{(q-1)}\right)^T$.

Model Inference – The E-step

- The E-steps updates the posterior probabilities $t_{ik}^{(q)} = \mathrm{E}[\mathrm{z_{ik}} \mid \gamma_i, \theta^{(q-1)}]$.
- Using Bayes' theorem, the posterior probabilities $t_{ik}^{(q)}$ can be expressed as follows:

$$t_{ik}^{(q)} = \frac{\pi_k^{(q)} \phi(\gamma_i, \theta_k^{(q)})}{\sum_{l=1}^{K} \pi_l^{(q)} \phi(\gamma_i, \theta_l^{(q)})},$$

where $\theta_k^{(q)}$ is the set of parameters for the k-th component updated in the M-step.

2 FADPclust

FADPclust

- FADPclust is a set of algorithms that implement an adaptive density peak detection technique.
- There are two algorithms, namely FADP1 and FADP2:
  - **FADP1** uses an L2 distance between raw functional curves.
  - **FADP2** uses a semimetric of functional principal components.
- FADPclust algorithms can be applied to multivariate functional data.

2 FADPclust

Density Peak Detection

- Clustering by density peak detection has the following advantages:
  - Like the K-medoids method, it has its basis only in the distance between data points.
  - Like DBSCAN, it is able to detect nonspherical clusters and to automatically find the correct number of clusters.

- The algorithm is based on two main assumptions:
  1. Cluster centres are surrounded by neighbours with lower local density.
  2. Cluster centres are at a relatively large distance from any points with a higher local density.

2 FADPclust

Density Peak Detection in a Simplified Setting

- For each data point i, we compute two quantities: its local density ρ_i and its distance δ_i from points of higher density.
- The local density ρ_i is defined as

$$\rho_i = \sum_j \chi(d_{ij} - d_c)$$

where $\chi(x) = 1$ if $x < 0$ and $\chi(x) = 0$ otherwise, and d_c is a cutoff distance.

- δ_i is measured by computing the minimum distance between the point i and any other point with higher density:

$$\delta_i = \min_{j:\rho_j > \rho_i} d_{ij} .$$

Density Peak Detection in a Simplified Setting

- The pairs (ρ_i, δ_i) naturally yields the cluster centres.
- After the cluster centres have been found, each remaining point is assigned to the same cluster as its nearest neighbour of higher density.

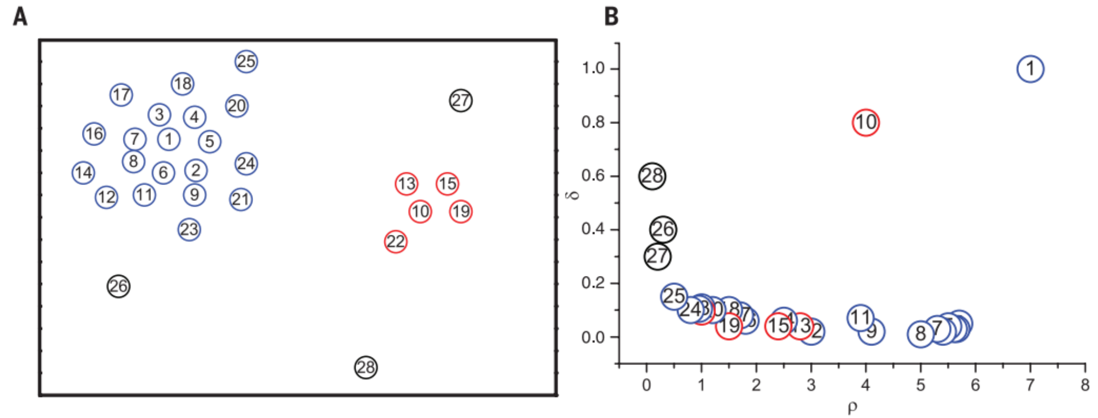
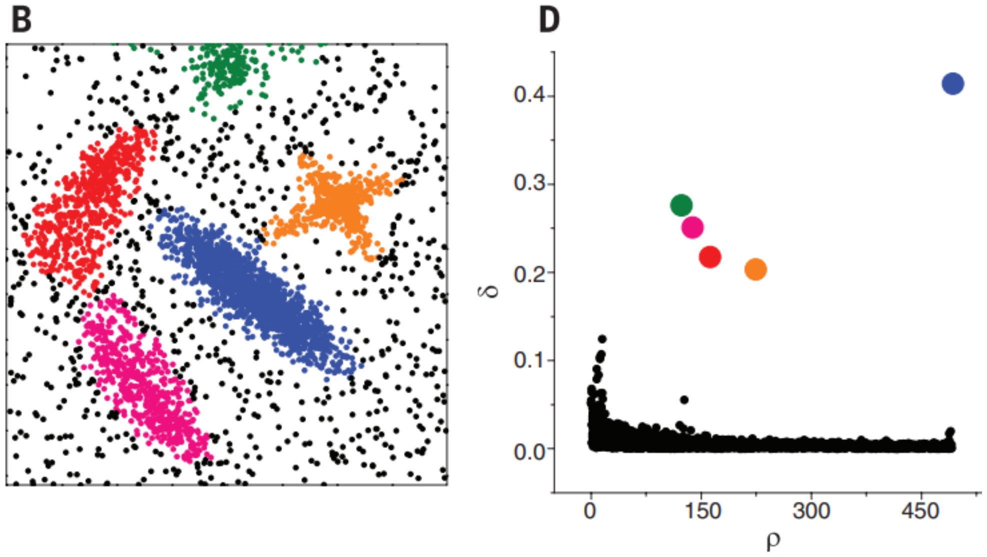


Fig. 1. The algorithm in two dimensions. (**A**) Point distribution. Data points are ranked in order of decreasing density. (**B**) Decision graph for the data in (A). Different colors correspond to different clusters.



2 FADPclust

Functional Extension of Density Peak Detection

- The FADPclust algorithm applies the idea of density peak detection by introducing appropriate metrics for evaluating the density-distance pairs (ρ_i, δ_i) for functional data.
- Note that various methods can be used for determining the local density of a data point.
- Another widely-used method is the K-NN density estimation:

$$\rho_x = \frac{k}{n} \cdot \frac{1}{V_d \cdot R_k^d(x)} = \frac{k}{n} \cdot \frac{1}{\text{Volume of a d-dimensional ball with radius } R_k(x))}$$

where $R_k(x)$ denotes the distance from x to its k-th nearest neighbour point.

2 FADPclust

Functional Extensions of Density Peak Detection

- The main problem is to decide the local density metric ρ_i ; the distance component δ_i can be easily defined analogously.
- FADP1 and FADP2 both use K-NN density estimation in order to determine the local density ρ_i, but adopt different metric (or semimetric) for measuring the distance between functional curves.

FADP1

- The density-distance pair $(\tilde{f}(X_i), \tilde{\delta}_i)$ for FADP1 is defined as follows:

$$\tilde{f}(\mathbf{X}_i) \sim \frac{1}{n h_{k,\mathbf{X}_i}} \sum_{j=1}^{n} K\left(\frac{d\left(\mathbf{X}_i, \mathbf{X}_j\right)}{h_{k,\mathbf{X}_i}}\right); \ \tilde{\delta}_i = \min_{j:\tilde{f}(\mathbf{X}_i)<\tilde{f}(\mathbf{X}_j)} d\left(\mathbf{X}_i, \mathbf{X}_j\right)$$

- Notations:
  - $\boldsymbol{X}_i$: data point (curve)
  - $K(x)$: kernel function satisfying $\int K(x)dx = 1$
  - $h_{k,X} = \min\{h \in \mathbb{R}^+, \sum_{j=1}^{n} I_{B(X,h_{k,X})}(\boldsymbol{X}_j) = k\}$: used instead of the volume of the d-ball.
- The choice of the density parameter k and the number of clusters is determined via a grid-search procedure.

2 FADPclust

FADP2

- The density-distance pair $(\tilde{f}(X_i), \tilde{\delta}_i)$ for FADP2 is defined as follows:

$$\tilde{f}(\mathbf{X}_i) \sim \frac{1}{n\check{h}_{k,\mathbf{X}_i}^{M}\hat{f}_{\mathbf{Y}}\left(\boldsymbol{y}_i;k\right)}\sum_{j=1}^{n}K\left(\frac{\left\|\boldsymbol{y}_i-\boldsymbol{y}_j\right\|}{\check{h}_{k,\mathbf{X}_i}}\right);\ \tilde{\delta}_i=\min_{j:\tilde{f}(\mathbf{X}_i)<\tilde{f}(\mathbf{X}_j)}\left\|\boldsymbol{y}_i-\boldsymbol{y}_j\right\|$$

- Whereas FADP1 uses the raw distance metric (L2-metric), FADP2 uses a semimetric based on functional principal components.
- The number of principal components, M, is selected based on the estimated eigenvalues or the percentage of variance explained.
- However, M has little impact on the clustering performance when the percentage of variance explained exceeds 90%.

2 FADPclust

FADP2

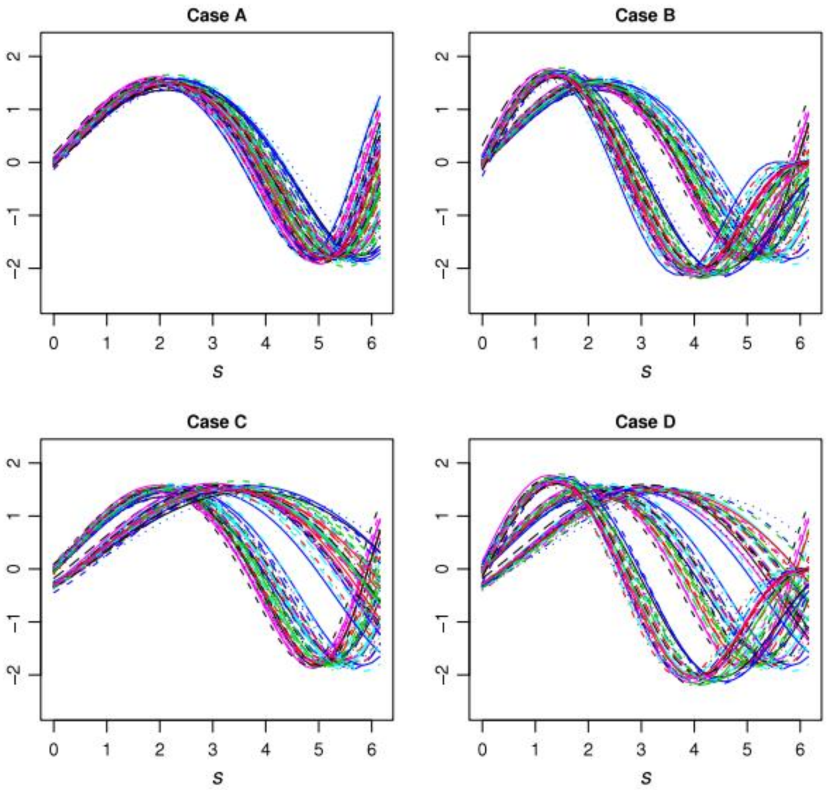
- The density-distance pair $(\tilde{f}(X_i), \tilde{\delta}_i)$ for FADP2 is defined as follows:

$$\tilde{f}(\mathbf{X}_i) \sim \frac{1}{n\check{h}^M_{k,\mathbf{X}_i}\hat{f}_{\mathbf{Y}}(\boldsymbol{y}_i;k)} \sum_{j=1}^{n} K\left(\frac{\left\|\boldsymbol{y}_i - \boldsymbol{y}_j\right\|}{\check{h}_{k,\mathbf{X}_i}}\right);\ \tilde{\delta}_i = \min_{j:\tilde{f}(\mathbf{X}_i)<\tilde{f}(\mathbf{X}_j)} \left\|\boldsymbol{y}_i - \boldsymbol{y}_j\right\|$$

- Notations:
  - $\boldsymbol{y}_i \in \mathbb{R}^M$: fPC scores corresponding to the curve $\boldsymbol{X}_i$
  - $\check{h}_{k,X} = \min\{h \in \mathbb{R}^+, \sum_{j=1}^n I_{[0,1]}\left(\frac{\|\boldsymbol{y} - \boldsymbol{y}_j\|}{h}\right) = k\}$
  - $\hat{f}_Y(\boldsymbol{y}, k)$: k-nearest neighbour density of the fPC vector $\boldsymbol{y}$

Clustering Misaligned Curves

- Problem of clustering misaligned curves based on usual metrics (or semimetrics):



Clustering Misaligned Curves

- In this approach, we aim to (1) suitably align similar curves and (2) assign them to homogeneous clusters. This approach can be applied to multivariate functional data.
- The notion of similarity between curves is defined by a similarity index $\rho(\cdot,\cdot)$ that measures the similarity between two curves, and a class W of warping functions h.

2 Clustering with Curve Alignment

Alignment of Curves

- We consider the cosine similarity of the 1st derivative of the curves as the similarity index between two curves $c_1, c_2 \in C$ ($C = \{c : c \in L_2(\mathbb{R}, \mathbb{R}^d), c \neq 0\}$):

$$\rho(\mathbf{c}_1, \mathbf{c}_2) = \frac{1}{d} \sum_{p=1}^{d} \frac{\int_{\mathbb{R}} c'_{1p}(s) c'_{2p}(s) \mathrm{d}s}{\sqrt{\int_{\mathbb{R}} c'_{1p}(s)^2 \mathrm{d}s} \sqrt{\int_{\mathbb{R}} c'_{2p}(s)^2 \mathrm{d}s}}$$

- This can be interpreted as the average of cosine similarity of the derivatives of the curves.
- The two curves are said to be similar when the index is close to its maximal value.
  In our case, the maximal value 1 is attained when the two curves are identical except for shifts and dilations:

$$\rho(\mathbf{c}_1, \mathbf{c}_2) = 1 \quad \Leftrightarrow \quad \begin{array}{l} \text{for } p = 1, \ldots, d, \exists A_p \in \mathbb{R}^+, \exists B_p \in \mathbb{R} : \\ c_{1p} = A_p c_{2p} + B_p. \end{array}$$

2 Clustering with Curve Alignment

Alignment of Curves

- Aligning $c_1 \in C$ to $c_2 \in C$ means finding a warping function $h(s)$ such that the two curves $c_1 \circ h$ and c_2 are the most similar.
- We choose the group of strictly increasing affine transformations as the class W of warping functions:

$$W = \{h : h(s) = ms + q, m \in \mathbb{R}^+, q \in \mathbb{R}\}.$$

- Our choice of (ρ, W) satisfies the following properties:
  - The similarity index ρ is bounded with maximum 1, so that two curves c_1 and c_2 are similar when $\rho(c_1, c_2) = 1$. Moreover, it is an equivalence relation.
  - W is a convex vector space and has a group structure with respect to $\circ$.
  - (ρ, W) is consistent in the sense that $\rho(c_1, c_2) = \rho(c_1 \circ h, c_2 \circ h), \forall h \in W$.

2 Clustering with Curve Alignment

Curve Clustering When Curves are Misaligned

- Constider the problem of clustering and aligning a set of N curves $\{c_1, \cdots, c_N\}$ with respect to k template curves $\underline{\phi} = \{\phi_1, \cdots, \phi_k\} \subset C$.
- Notations:
  - Template: An ideal representation of curves in each cluster.
  - For each template curve ϕ_j, define the domain of attraction

$$\Delta_j(\underline{\boldsymbol{\varphi}}) = \{\mathbf{c} \in \mathcal{C} : \sup_{h \in W} \rho(\boldsymbol{\varphi}_j, \mathbf{c} \circ h) \geq \sup_{h \in W} \rho(\boldsymbol{\varphi}_r, \mathbf{c} \circ h), \forall r \neq j\}, \quad j = 1, \ldots, k.$$

  - Define the labelling function (for well-definedness)

$$\lambda(\underline{\boldsymbol{\varphi}}, \mathbf{c}) = \min\{r : \mathbf{c} \in \Delta_r(\underline{\boldsymbol{\varphi}})\}.$$

2 Clustering with Curve Alignment

Curve Clustering When Curves are Misaligned

- In order to cluster and align the set of N curves $\{c_1, \cdots, c_N\}$ with respect to k unknown templates, we should solve the following optimization problem:

  1) Find $\underline{\phi} = \{\phi_1, \cdots, \phi_k\} \subset C$ and $\underline{h} = \{\mathrm{h}_1, \cdots, h_N\} \subset W$ such that

  $$\frac{1}{N}\sum_{i=1}^{N} \rho(\boldsymbol{\varphi}_{\lambda(\underline{\boldsymbol{\varphi}},\mathbf{c}_i)}, \mathbf{c}_i \circ h_i) \geq \frac{1}{N}\sum_{i=1}^{N} \rho(\boldsymbol{\psi}_{\lambda(\underline{\boldsymbol{\psi}},\mathbf{c}_i)}, \mathbf{c}_i \circ g_i)$$

     for any other set of k templates $\underline{\psi} = \{\psi_1, \cdots, \psi_k\} \subset C$ and N warping functions $\underline{g} = \{g_1, \cdots, g_k\} \subset W$.

  2) Assign c_i to the cluster $\lambda\left(\underline{\phi}, \mathrm{c_i}\right)$ and align it to the corresponding template $\phi_{\lambda\left(\underline{\phi}, \mathrm{c_i}\right)}$ using the warping function h_i.

Curve Clustering When Curves are Misaligned

- The optimization problem 1) is not analytically solvable.
- For this reason, we simultaneously deal with 1) and 2) via a k-mean alignment algorithm that iteratively alternates template identification steps and assignment and alignment steps.

2 Clustering with Curve Alignment

K-Mean Alignment Algorithm

- In the **template identification step**, we estimate the set of k templates associated to the k clusters identified at the previous assignment and alignment step.
- In the **assignment and alignment step**, we align the N curves to the set of the k templates obtained in the previous template identification step, and we assign each of the curves to one of the k clusters.
- However, such a solution may not be unique; this problem is resolved via a **normalization step.**
- The initial templates are chosen at random, with the only requirement that none of the template curves are similar.
- The algorithm stops when the increments of the similarity indices are all lower than 0.01.

2 Clustering with Curve Alignment

K-Mean Alignment Algorithm – Template Identification Step

- For $j = 1, \cdots, k$, the template of the j-th cluster $\phi_{j[q]}$ is estimated using all curves assigned to cluster j at iteration q-1.
- Ideally, the template $\phi_{j[q]}$ should be estimated as the curve $\phi \in C$ that maximizes the total similarity:

$$\sum_{i:\lambda(\underline{\boldsymbol{\varphi}}[q-1], \mathbf{c}_{i[q-1]})=j} \rho(\boldsymbol{\varphi}, \mathbf{c}_{i[q-1]}).$$

- In practice, we estimate $\phi_{j[q]}$ via Loess, instead of directly maximizing the total similarity.
- Since only first derivatives are required in the definition of ρ, it is sufficient to estimate the first derivative $\phi'_{j[q]}$.

2 Clustering with Curve Alignment

K-Mean Alignment Algorithm – Assignment and Alignment Step

- The set of curves $\left\{c_1^{[q-1]}, \cdots, c_N^{[q-1]}\right\}$ is clustered and aligned to the set of templates $\underline{\phi_{[q]}} = \left\{\phi_1^{[q]}, \cdots, \phi_k^{[q]}\right\}$:
  - $c_i^{[q-1]}$ is aligned to $\phi_{\lambda\left(\underline{\phi_{[q]}}, c_i^{[q-1]}\right)}$.
  - The aligned curve $\widetilde{c_i^{[q]}} = c_i^{[q-1]} \circ h_i^{[q]}$ is assigned to cluster $\lambda\left(\underline{\phi_{[q]}}, c_i^{[q-1]}\right) \equiv \lambda\left(\underline{\phi_{[q]}}, \widetilde{c_i^{[q]}}\right)$.

K-Mean Alignment Algorithm – Normalization Step

- The normalization step is implemented in order to choose, among all candidate solutions to the optimization problem, the one that leaves the average locations of the clusters unchanged.
- After the template identification step and the assignment and alignment step, a warping function is applied to all the curves of each cluster respectively, so that the average warping undergone by curves to each cluster is the identity transformation.
- This avoids the drifting apart of clusters of the global drifting of the overall set of curves.

2 Clustering via Functional PCA

Functional PCA + Finite-Dimensional Methods

- The main difficulty in clustering functional data comes from the fact the function space is infinite-dimensional.
- Dimension reduction techniques, such as principal component analysis, can be utilized to address this problem.
- Functional PCA allows to interpret functional data as finite-dimensional vectors (fPC scores), spanned by a set of principal component functions.
- Common clustering methods, such as K-Means or hierarchical clustering, can be applied to fPC scores.

2 Evaluation Metrics

Evaluating Clustering Results

- Determining the best clustering method and the best number of clusters is a practical issue that arises in performing clustering.
- Various evaluation metrics that can be used to measure how well the data points are clustered.
- The **silhouette coefficient** and the **Dunn index** is used as evaluation metrics in this project.
  - Both metrics involve a distance metric $d(\cdot, \cdot)$.
  - L_2-norm is used as the distance metric d throughout this project.

Silhouette Coefficient

- The silhouette of a data point $i \in C_I$ is defined as

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

where $a(i) = \frac{1}{|C_I|-1} \sum_{j \in C_I, i \neq j} d(i,j)$ and $b(i) = \min_{J \neq I} \sum_{j \in C_J} d(i,j)$

- The silhouette coefficient of a clustering result is the average of silhouettes of data points.
- The silhouette coefficient ranges from -1 to 1, where a higher value indicates better clustering results.

Dunn Index

- The Dunn index is defined as a ratio of the inter-cluster distance and intra-cluster distance of a data point:

$$DI_K = \frac{\min_{1 \le i \le j \le K} \delta(C_i, C_j)}{\max_{1 \le k \le K} \Delta_K}.$$

  - Inter-cluster distance: $\delta(C_i, C_j) = \min_{x \in C_i, y \in C_j} d(x, y)$
  - Intra-cluster distance (diameter): $\Delta_I = \max_{x, y \in C_I} d(x, y)$

- The Dunn index ranges from 0 to infinity. Like the silhouette coefficient, higher value of the Dunn index indicates better clustering results.

3

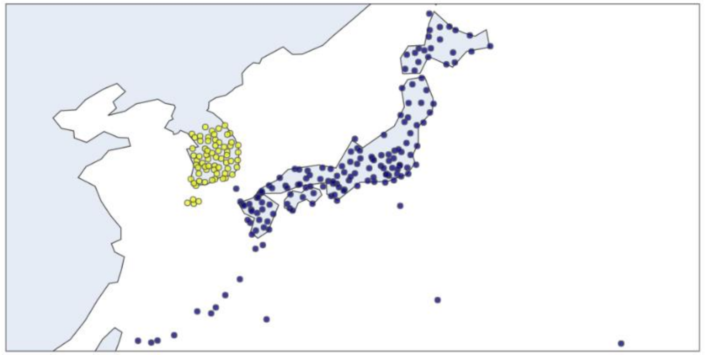
Application to Climate Data

Comparison via Evaluation Metrics / Interpretation of Results

3 Dataset

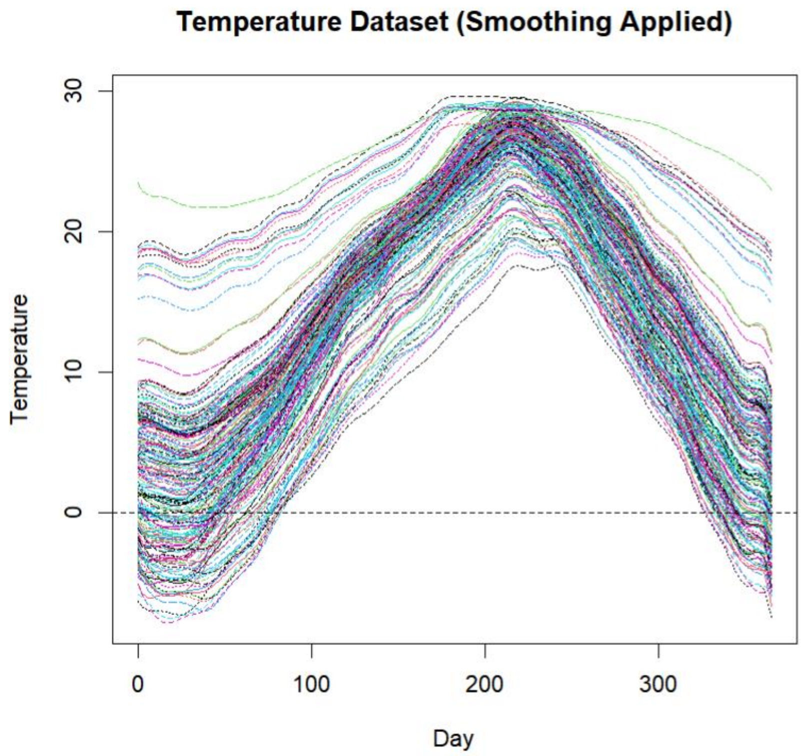
Mean Temperature Data of Korea and Japan

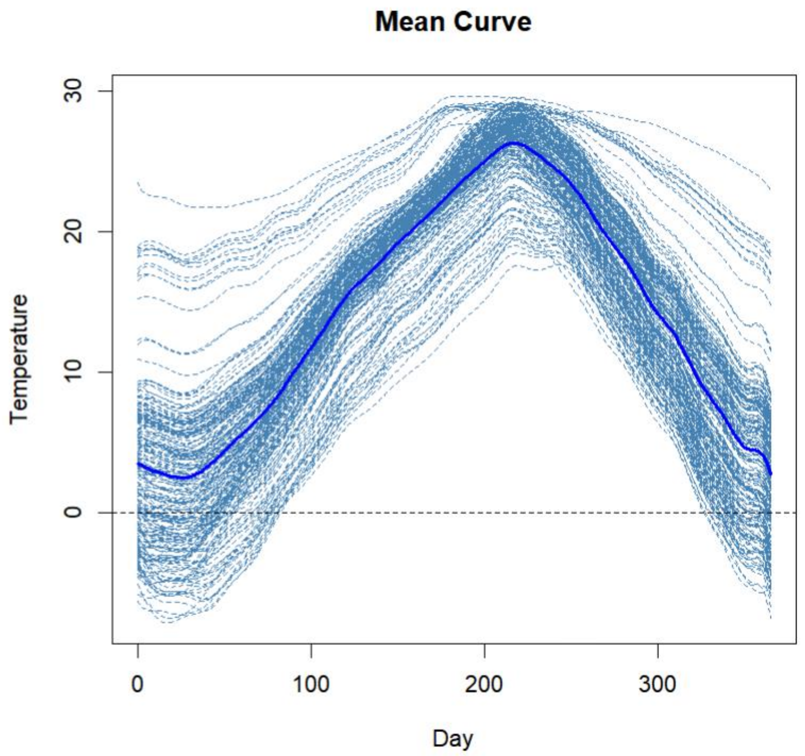
- In this project, we aim to cluster mean annual temperature data of Korea and Japan.
- The raw dataset consists of 365 rows and 225 columns:
  - The rows correspond to the daily mean temperature.
  - The columns correspond to the weather stations - 66 stations in Korea and 159 stations in Japan.
- Each column is smoothed and made into a functional curve using B-splines (cubic splines).



Functional Representation of Data

- All the curves are defined on an identical time grid [0, 365].
- Curves are unimodal and roughly symmetrical.





3 Clustering Methods

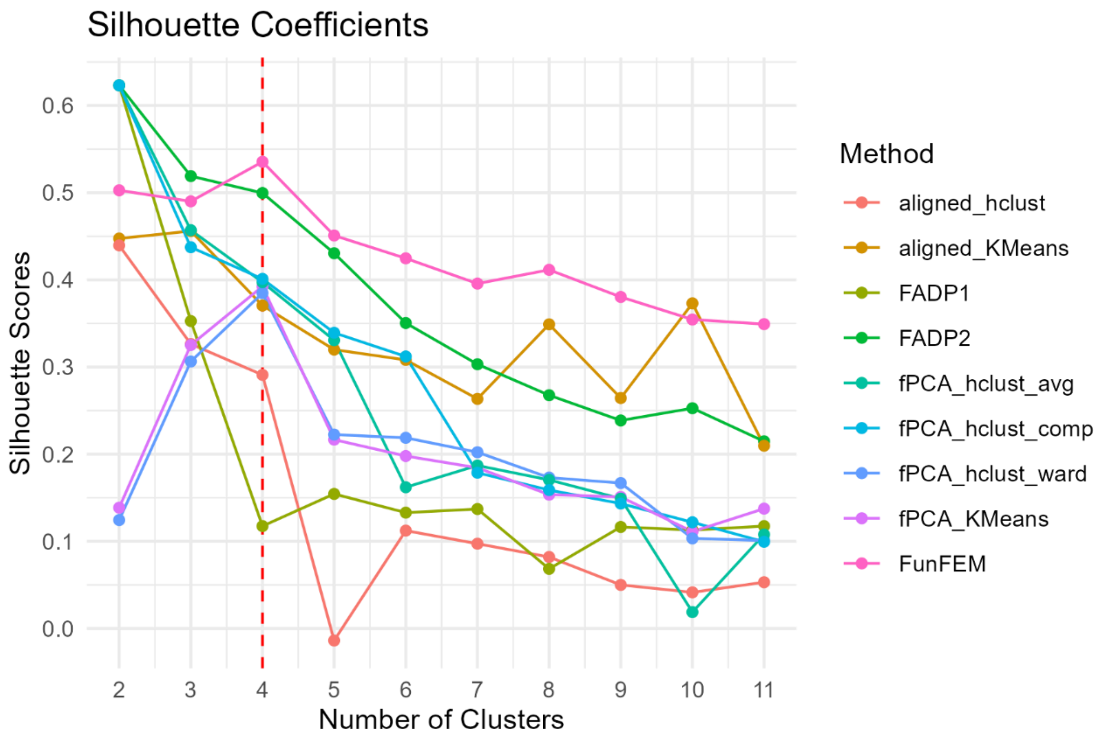
The following clustering methods were applied to the dataset:

- **FunFEM**
- **FADPclust:** FADP1, FADP2
- **Clustering with curve alignment:** K-means, hierarchical clustering (linkage = complete)
- **Functional PCA + finite-dimensional methods:** K-means, hierarchical clustering (linkage = Ward, complete, average)

3 Clustering Results

Comparison via Silhouette Coefficient

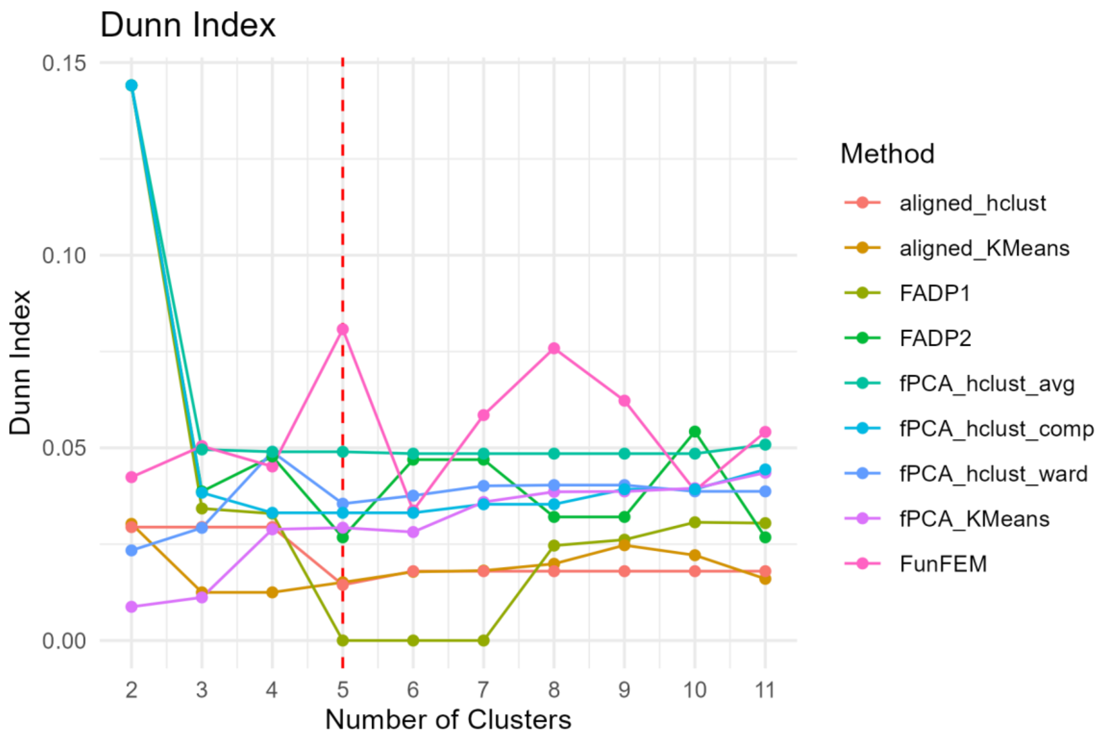
- Since clustering with K = 2 does not yield meaningful results, we only consider K ≥ 3.
- Best method: FunFEM, K = 4



	Method	# Clusters	Silhouette Score
0	FunFEM	4	0.5355
2	FADP2	3	0.5191
7	fPCA_hclust_avg	3	0.457
3	aligned_KMeans	3	0.456
8	fPCA_hclust_comp	3	0.4374
5	fPCA_KMeans	4	0.3925
6	fPCA_hclust_ward	4	0.3848
1	FADP1	3	0.3527
4	aligned_hclust	3	0.3264

Comparison via Dunn Index

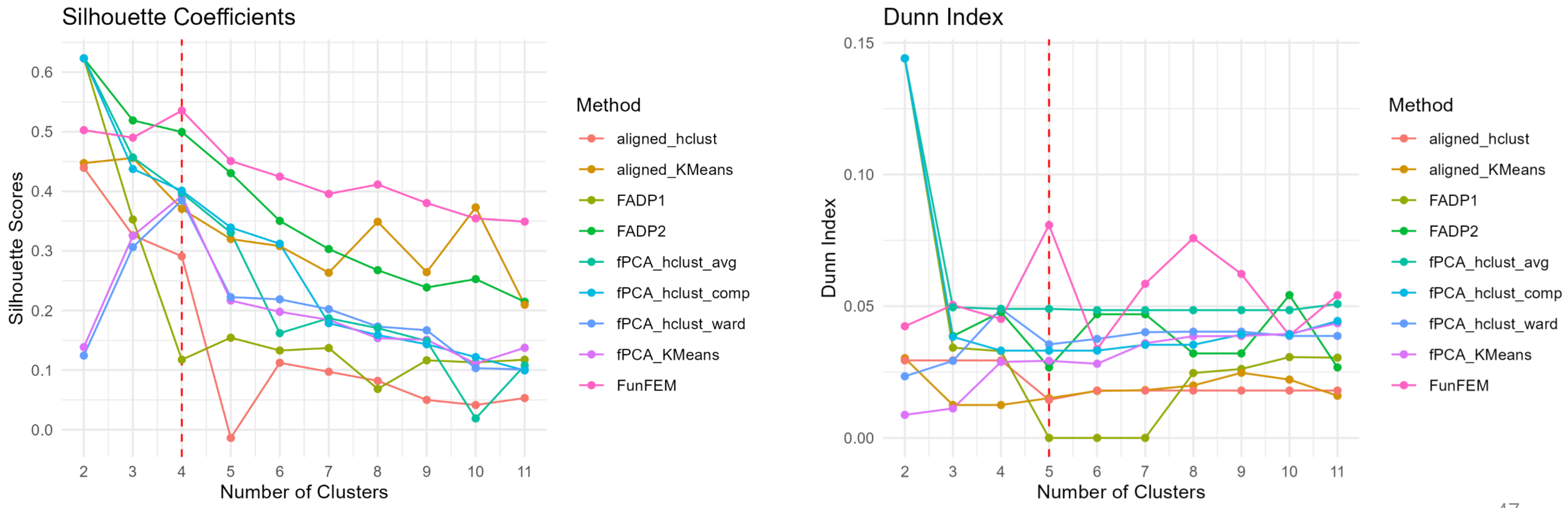
- The results are quite different from the previous result from silhouette coefficients.
- Best method: FunFEM, K = 5



	Method	# Clusters	Dunn Index
0	FunFEM	5	0.0808
2	FADP2	10	0.0542
7	fPCA_hclust_avg	11	0.0508
6	fPCA_hclust_ward	4	0.049
8	fPCA_hclust_comp	11	0.0444
5	fPCA_KMeans	11	0.0436
1	FADP1	3	0.0343
4	aligned_hclust	3	0.0294
3	aligned_KMeans	9	0.0247

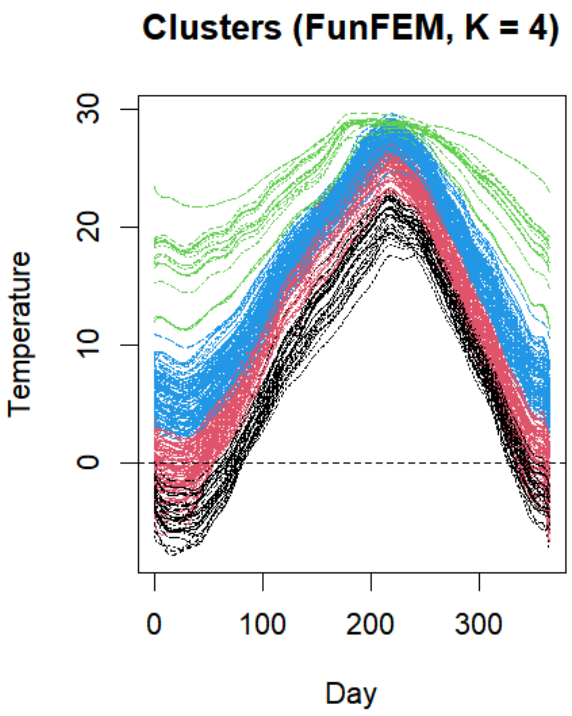
Comparison via Dunn Index

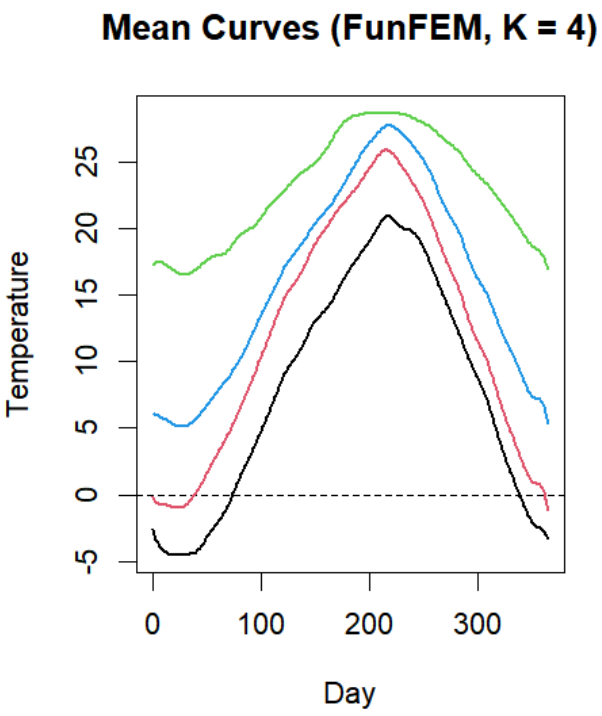
- **FunFEM** performed best according to both criteria, followed by **FADP2** and **fPCA + hierarchical clustering**.
- FADP1 and curve alignment algorithms showed poor performance.

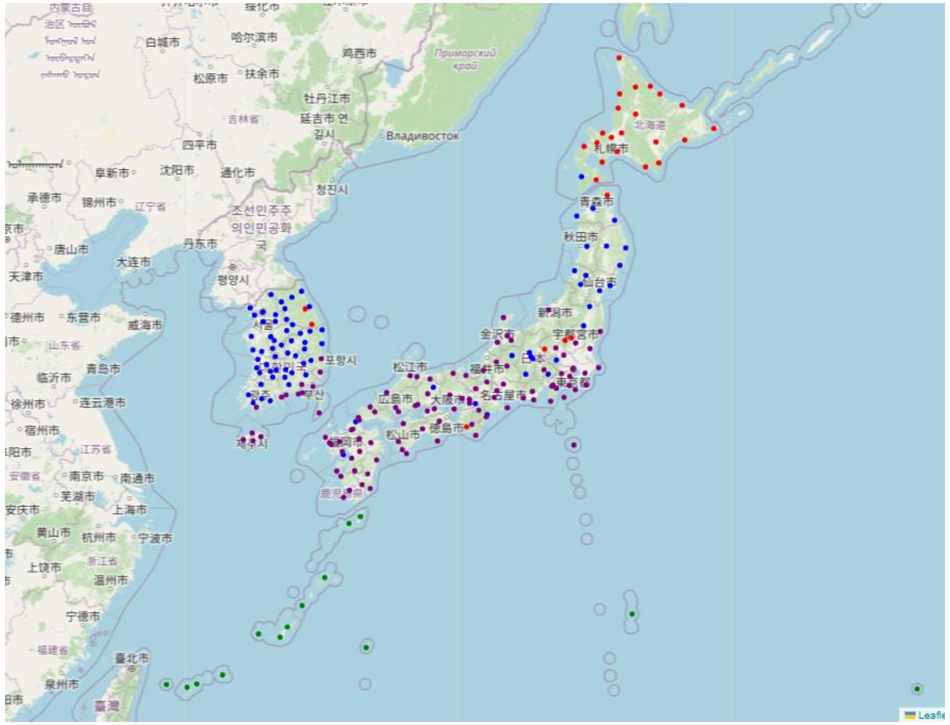


3 Clustering Results

Best Results – FunFEM, K = 4

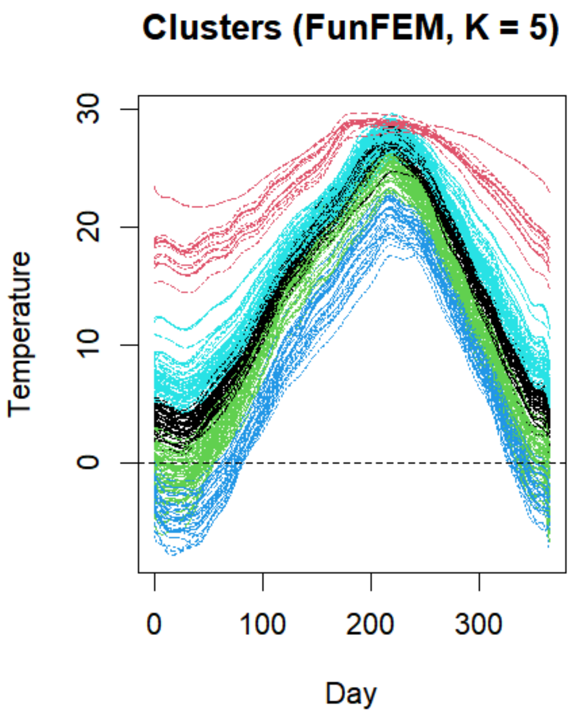


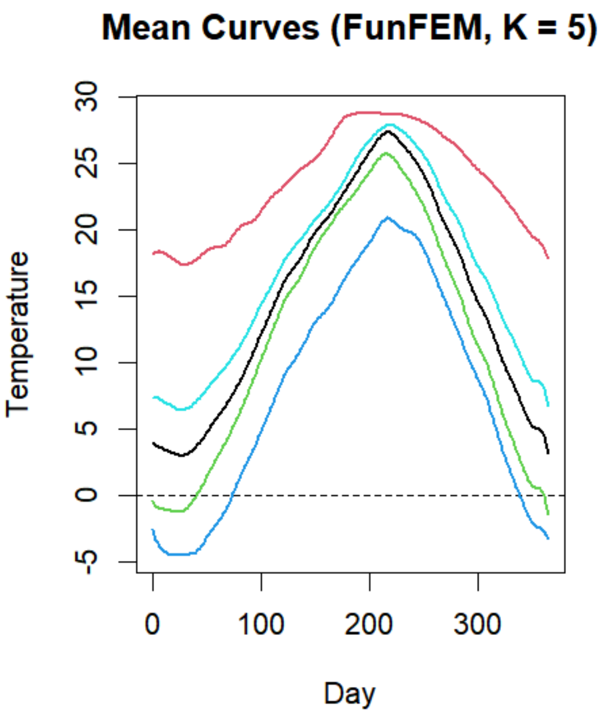


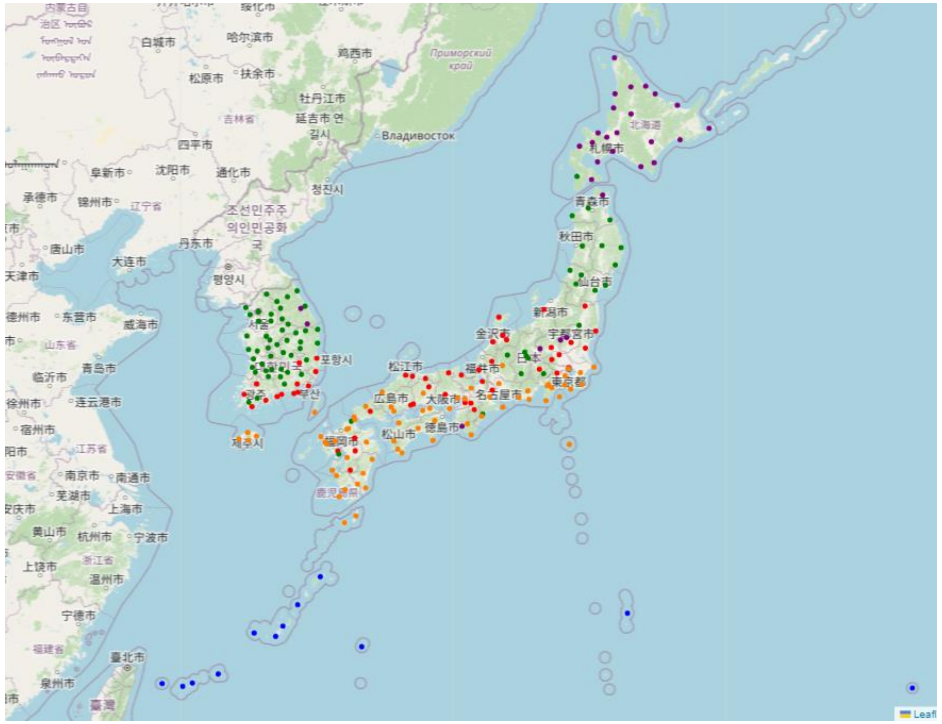


3 Clustering Results

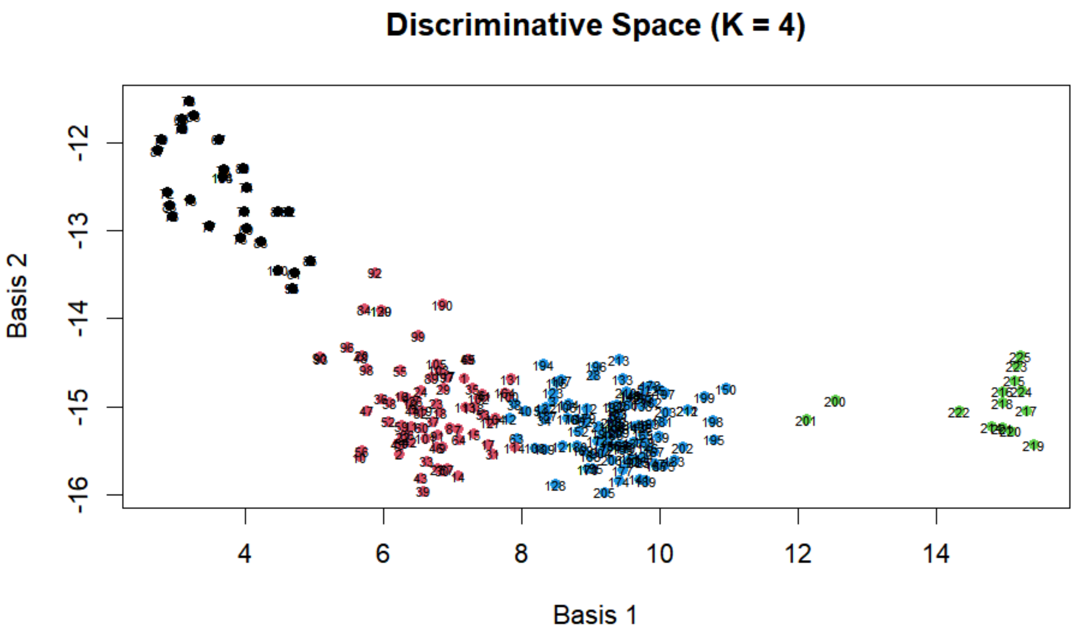
Best Results – FunFEM, K = 5

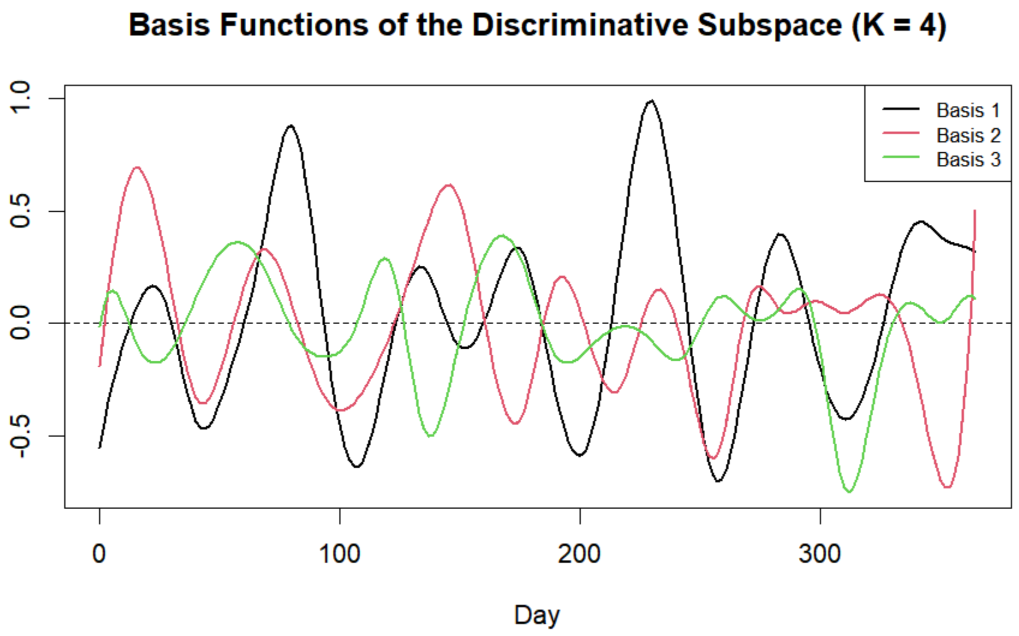




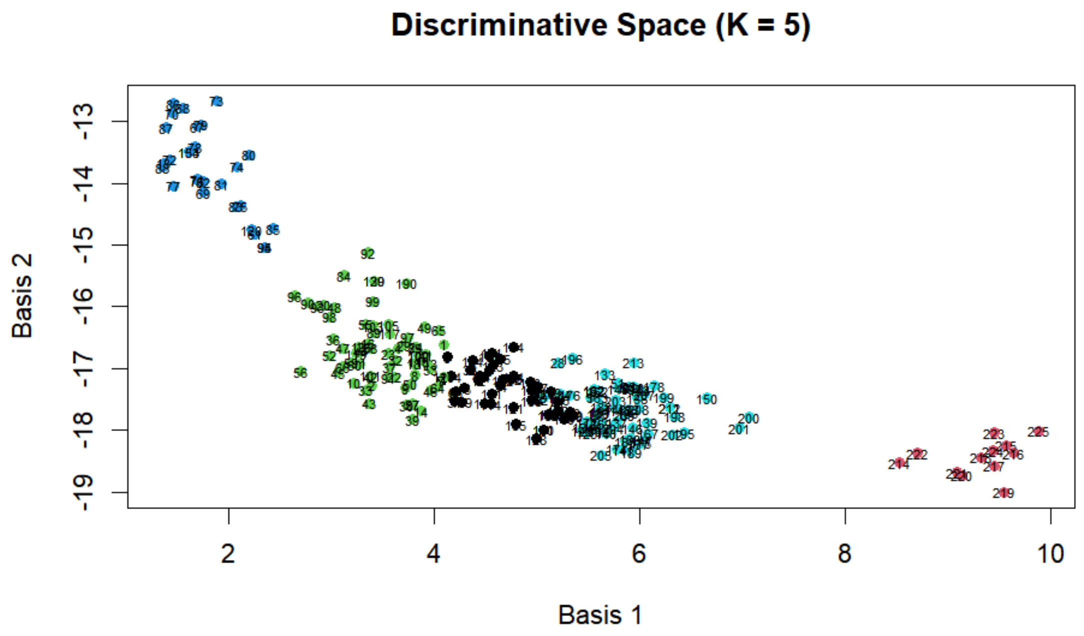


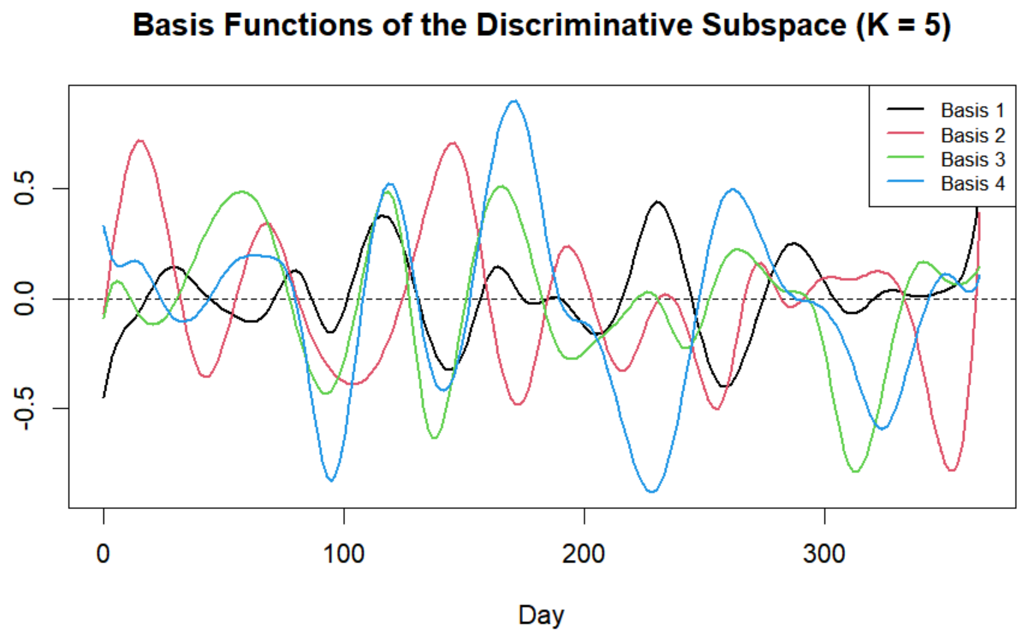
Best Results – FunFEM, K = 4





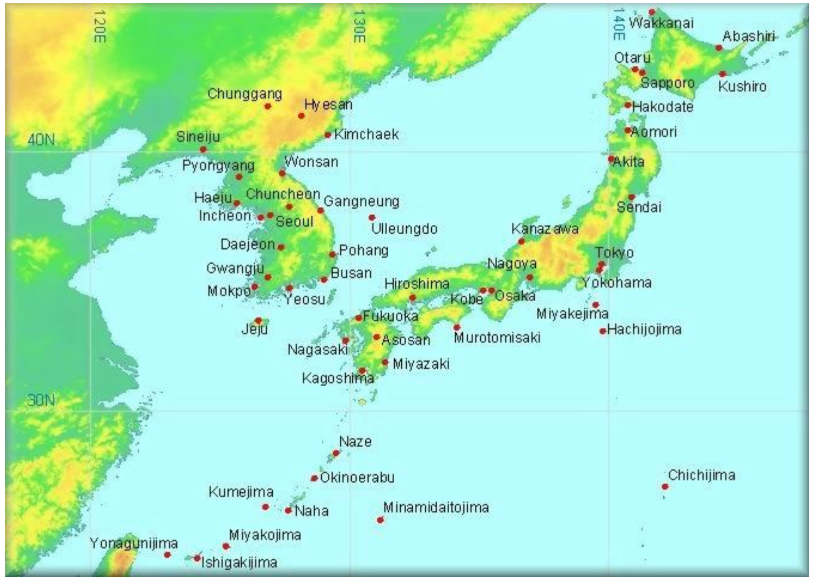
Best Results – FunFEM, K = 5

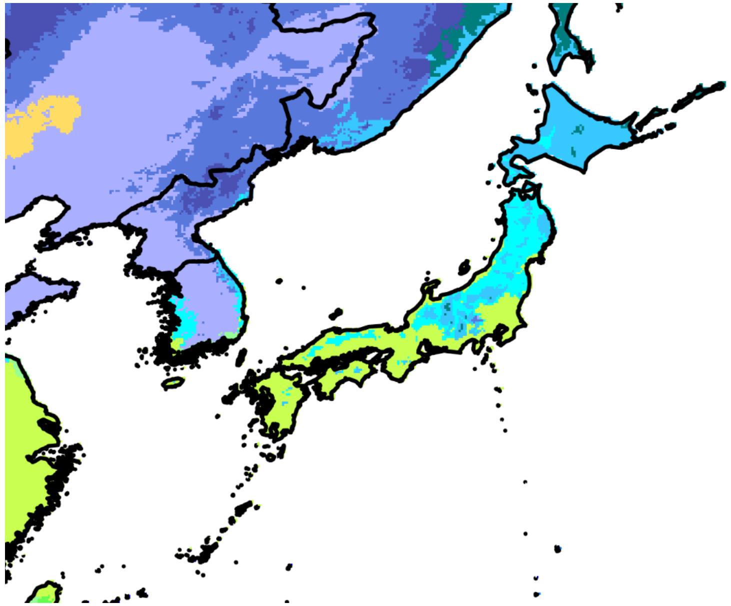




Interpreting the Clusters

1. **Geographical/Topographical Connectedness**: See whether each cluster is visually (and 'intuitively') identifiable when viewed on a map.
2. **The Köppen climate classification:** Comparison with existing climate classification systems.





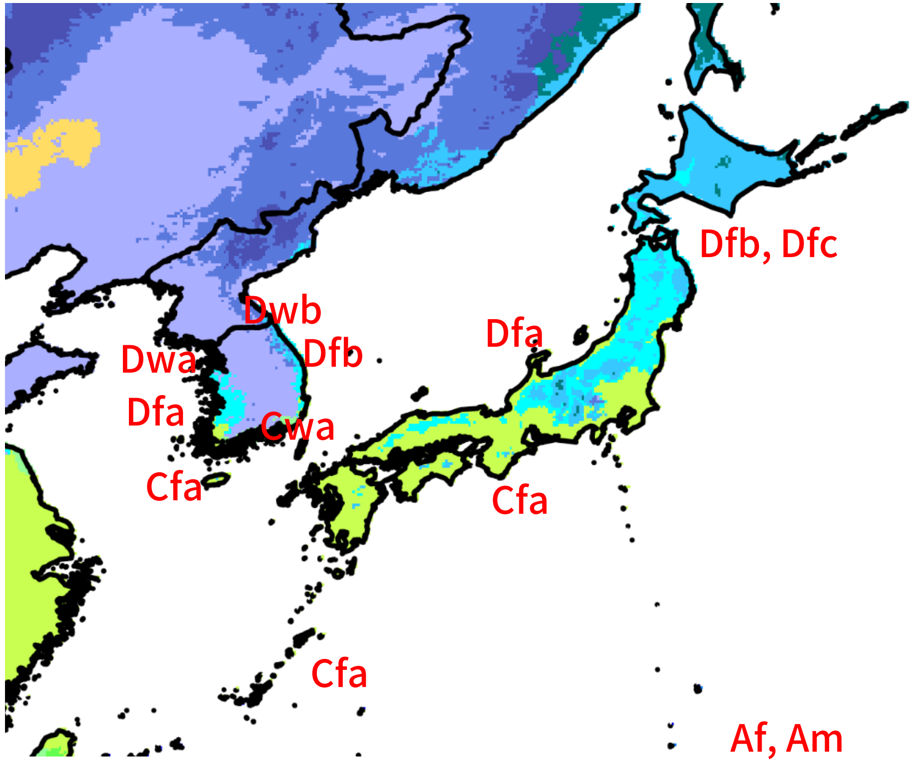
Köppen climate classification

- The Köppen climate classification is one of the most widely used climate classification systems, based on seasonal temperature and precipitation.
- The first letter indicates the five main climate groups.
- The second letter indicates the seasonal precipitation type, while the third letter indicates the level of heat.

1st	2nd	3rd
A (Tropical)	f (Rainforest) m (Monsoon) w (Savanna, dry winter) s (Savanna, dry summer)	
B (Dry)	W (Arid Desert) S (Semi-Arid or steppe)	h (Hot) k (Cold)
C (Temperate)	w (Dry winter) f (No dry season) s (Dry summer)	a (Hot summer) b (Warm summer) c (Cold summer)
D (Continental)	w (Dry winter) f (No dry season) s (Dry summer)	a (Hot summer) b (Warm summer) c (Cold summer) d (Very cold winter)
E (Polar)		T (Tundra) F (Ice cap)

3 Interpretation of Results

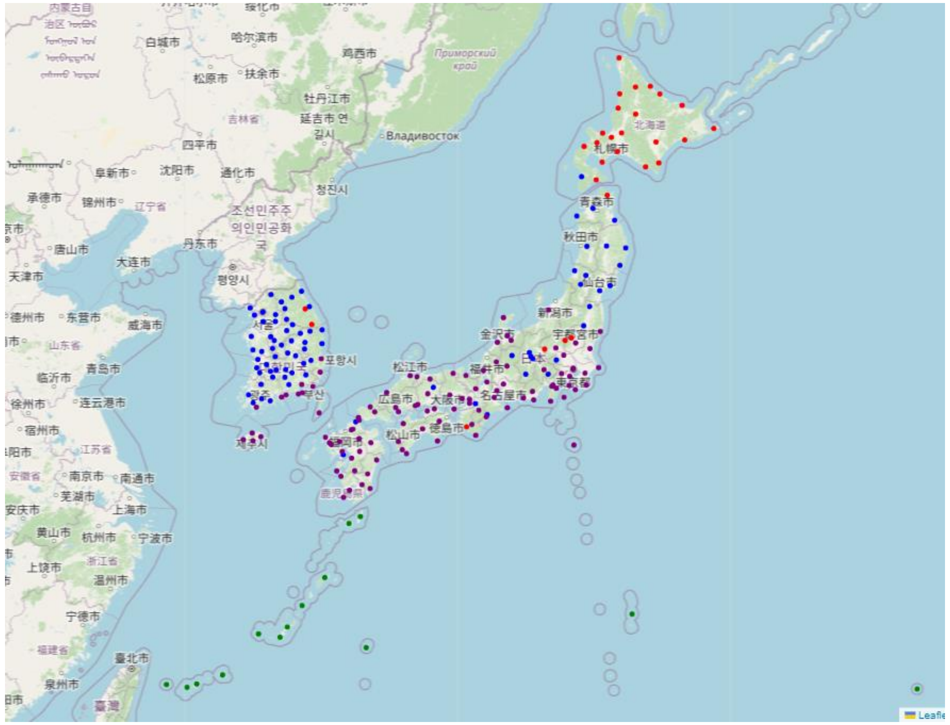
Köppen climate classification

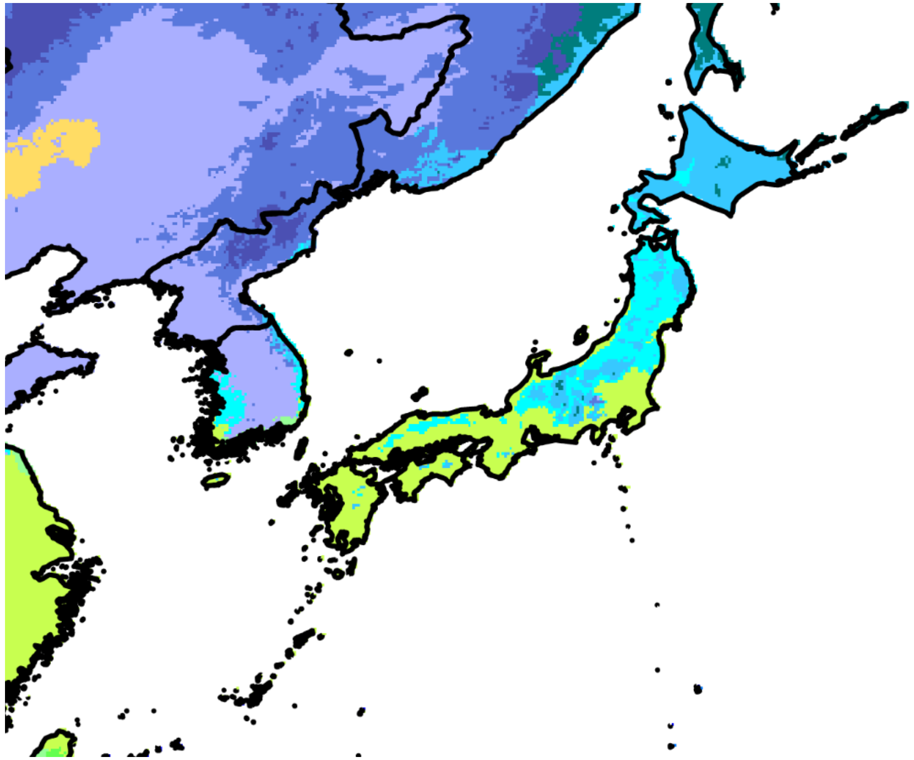


1st	2nd	3rd
A (Tropical)	f (Rainforest) m (Monsoon) w (Savanna, dry winter) s (Savanna, dry summer)	
B (Dry)	W (Arid Desert) S (Semi-Arid or steppe)	h (Hot) k (Cold)
C (Temperate)	w (Dry winter) f (No dry season) s (Dry summer)	a (Hot summer) b (Warm summer) c (Cold summer)
D (Continental)	w (Dry winter) f (No dry season) s (Dry summer)	a (Hot summer) b (Warm summer) c (Cold summer) d (Very cold winter)
E (Polar)		T (Tundra) F (Ice cap)

Clustering Results vs Köppen climate classification

- FunFEM, K = 4

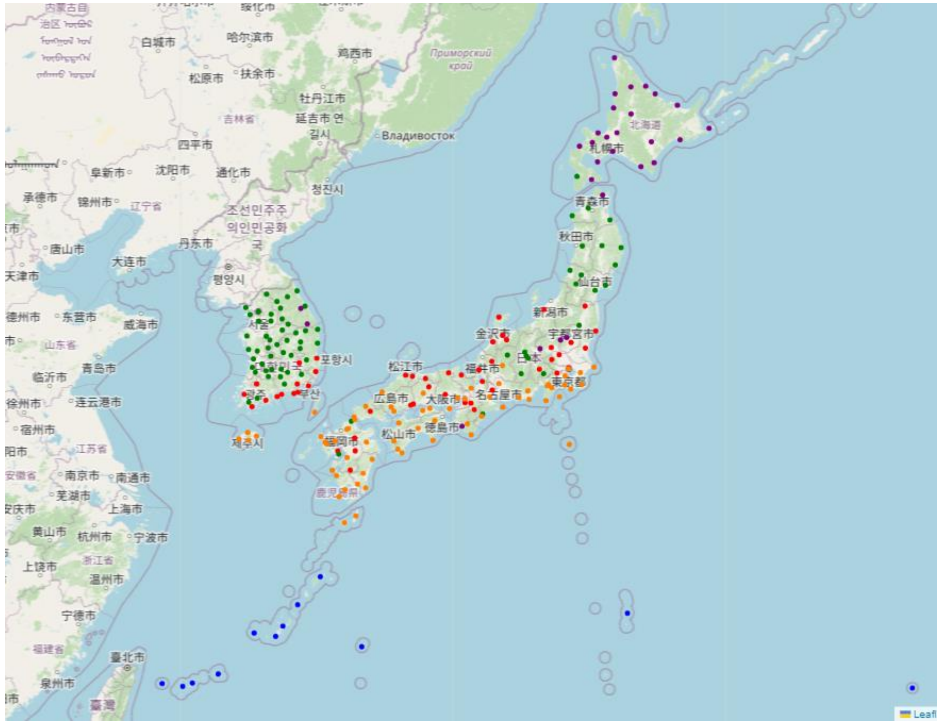


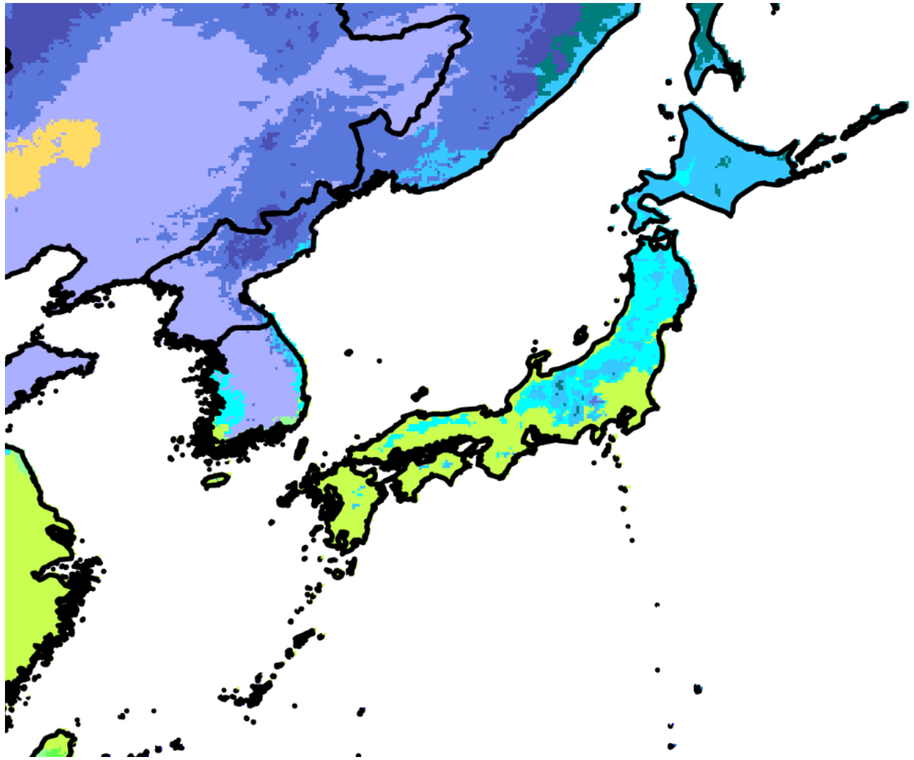


3 Interpretation of Results

Clustering Results vs Köppen climate classification

- FunFEM, K = 5

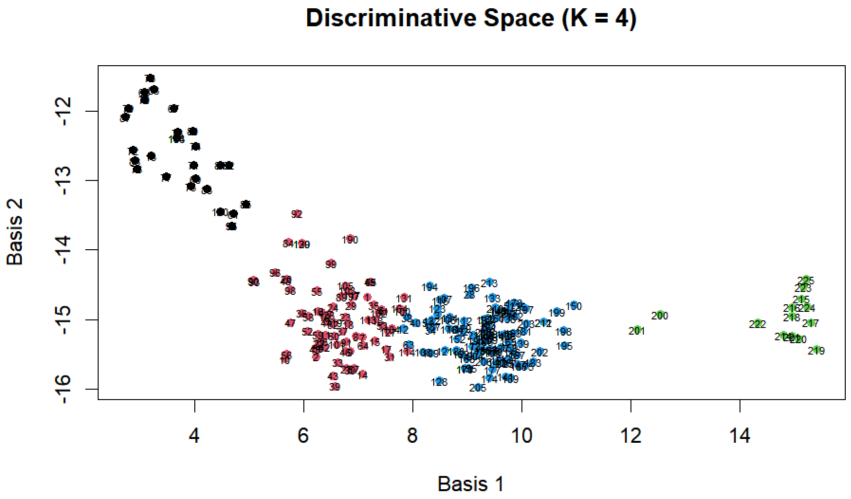




3 Interpretation of Results

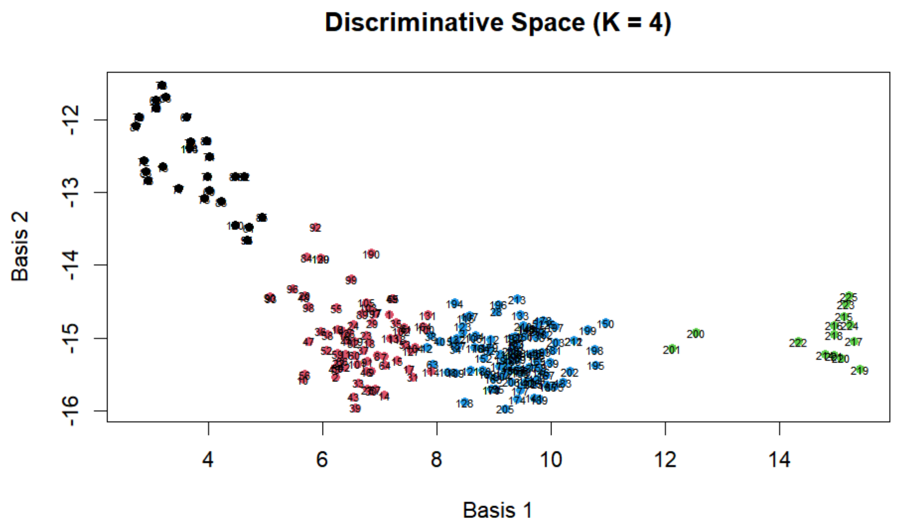
Why did FunFEM perform so well?

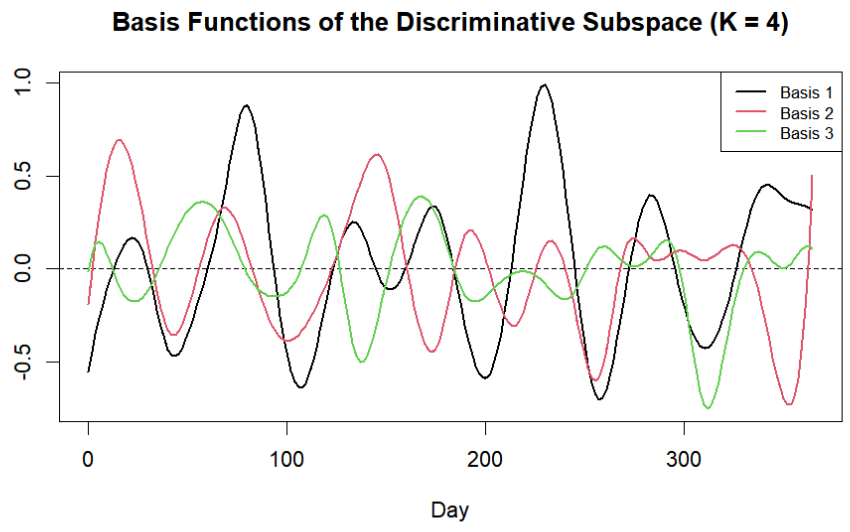
- Few possible reasons:
  - The Gaussian distribution assumption was well met.
  - All the data points were defined on the same time grid, and it was sufficient to only consider the amplitude variability of the data.



Why did FunFEM perform so well?

- Disadvantages:
    - FunFEM can only be applied to univariate functional data.
    - Unlike PCA, it is difficult to interpret the discriminative subspace and its basis functions.





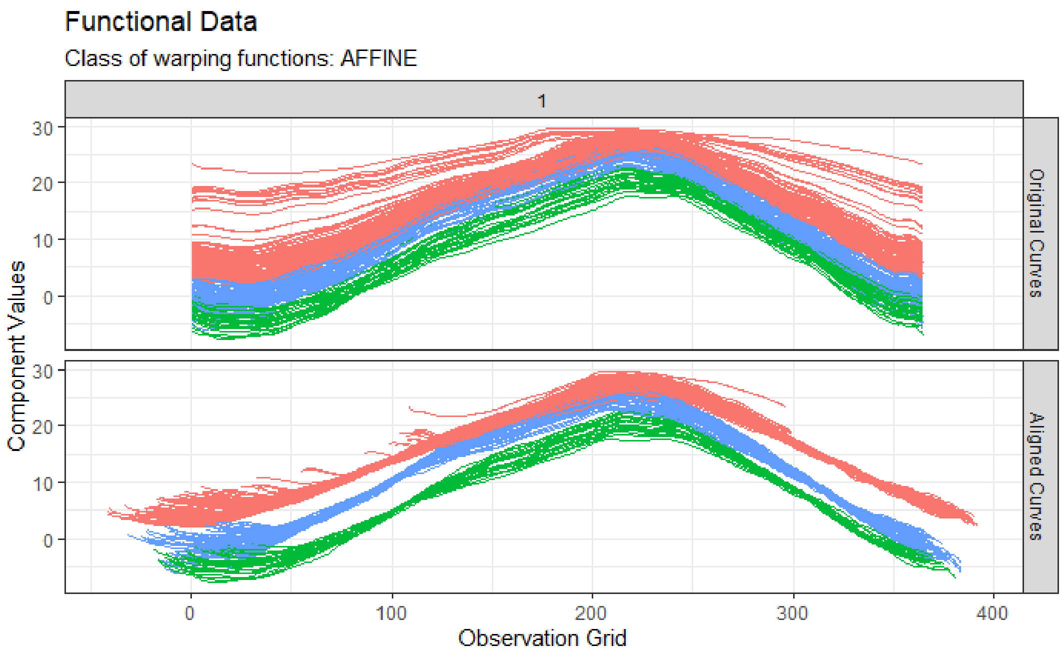
Poor performance of FADP1 and curve aligning algorithms

- **FADPclust**
  - Since it is a density-based method, it has its strength in its ability to detect non-spherical clusters.
  - Hence, FADPclust may perform better than other methods in terms of multivariate functional clustering.

3 Interpretation of Results

Poor performance of FADP1 and curve aligning algorithms

- **Curve aligning algorithms**:
  - Since all of the curves are symmetric and unimodal, they are somehow similar; hence much of the variability of our data is ignored when affine transformation is applied.



3 Summary

- Among all the clustering algorithms, FunFEM performed best, in terms of both the silhouette coefficient and the Dunn index.
- The optimal numbers of clusters were chosen as 4 and 5, respectively, based on the silhouette coefficient and the Dunn index.
- Even though FADP1 and curve alignment methods showed poor performance for this dataset, these algorithms still have some advantages:
  - FADPclust and curve alignment methods can be used to cluster multivariate functional data.
  - FADPclust can detect nested clusters efficiently.
  - Curve alignment methods are more suitable for data with phase and amplitude variability, for which we want to identify each cluster by the ‘overall shape’.

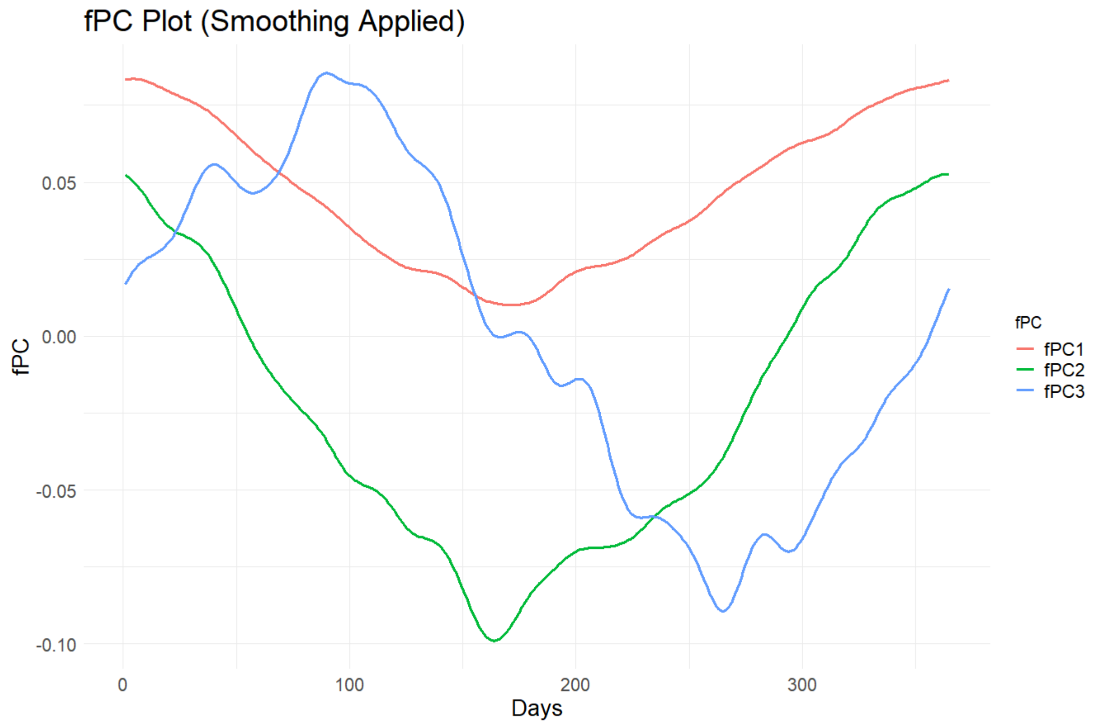
Appendix

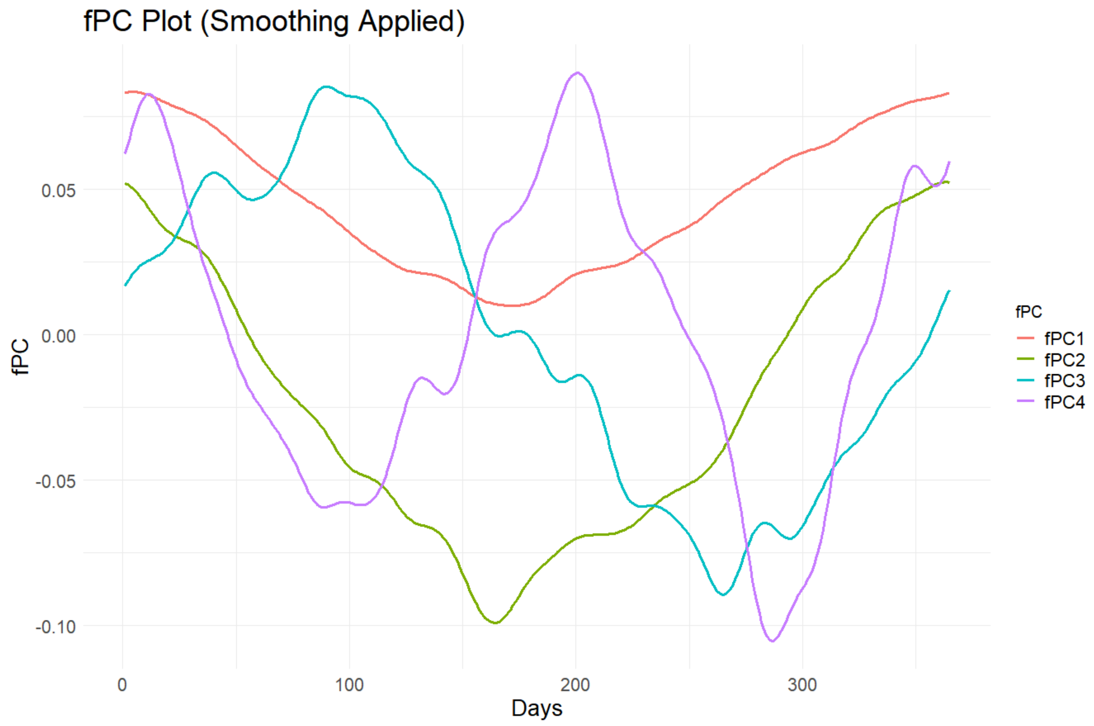
fPCA / More Clustering Results

A fPCA

Dimension Reduction via fPCA

- fPC 1, fPC 2는 전반적인 경향성을 반영
- fPC 3는 봄과 가을의 차이를, fPC 4는 봄/가을과 여름/겨울의 차이를 반영

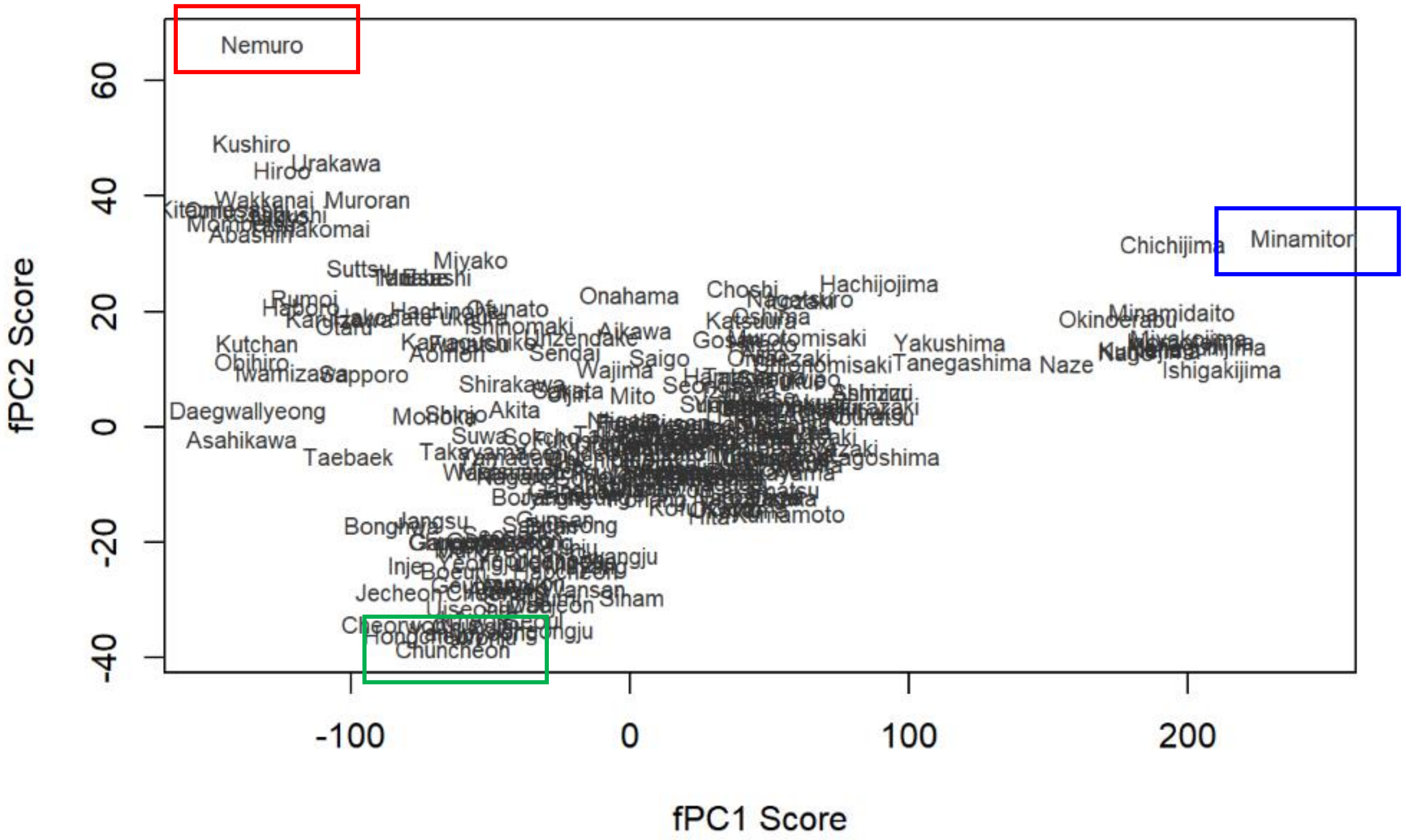




A **fPCA**

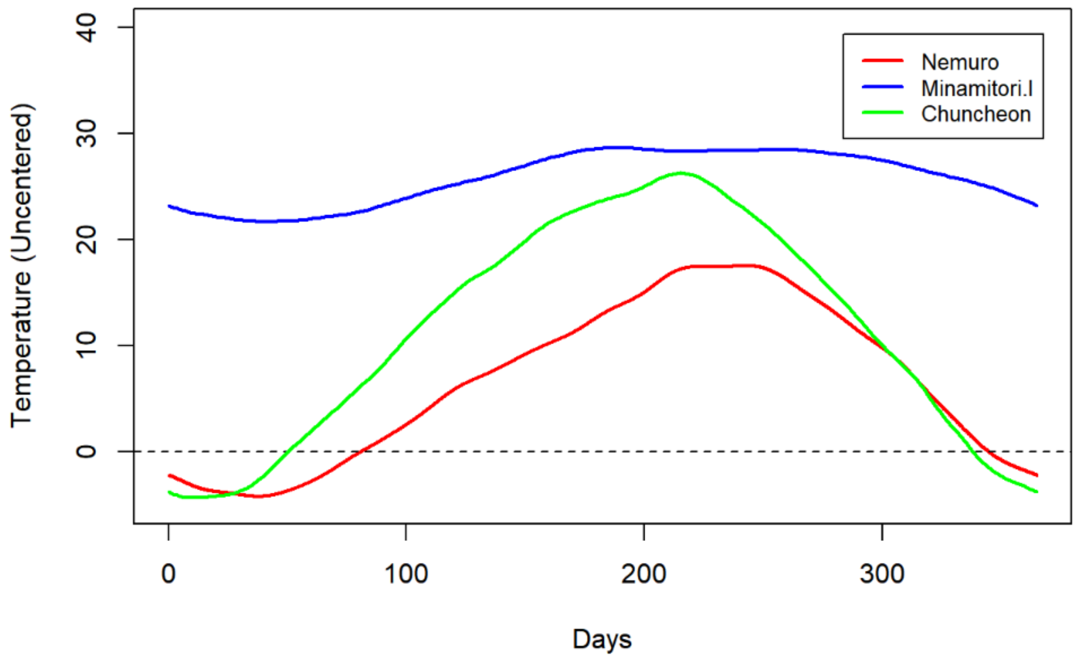
Dimension Reduction via fPCA

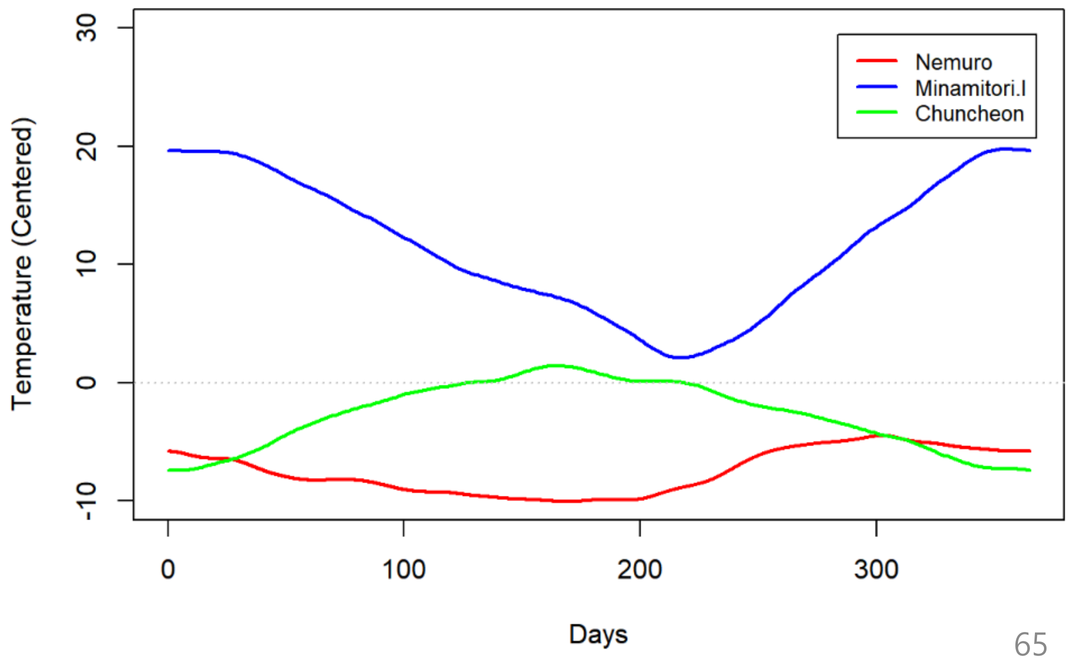
- Scatterplot of fPC 1 and fPC 2(cumulative variance 98.6%)



Dimension Reduction via fPCA

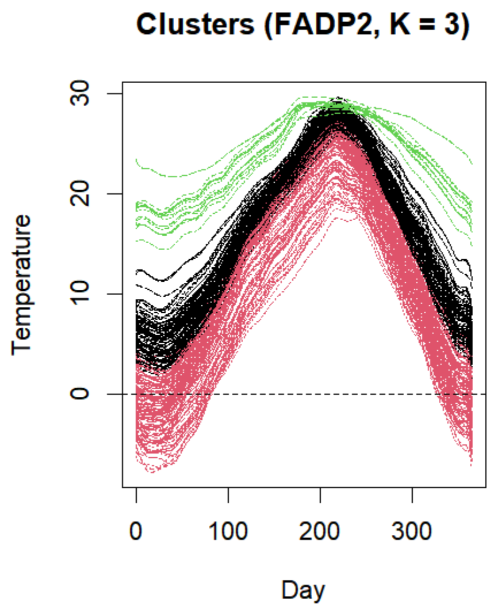
- Examples: Nemuro, Minamitori Island, Chuncheon
  - fPC 1 (Nemuro vs Minamitori I.)
  - fPC 2 (Chuncheon vs Nemuro)

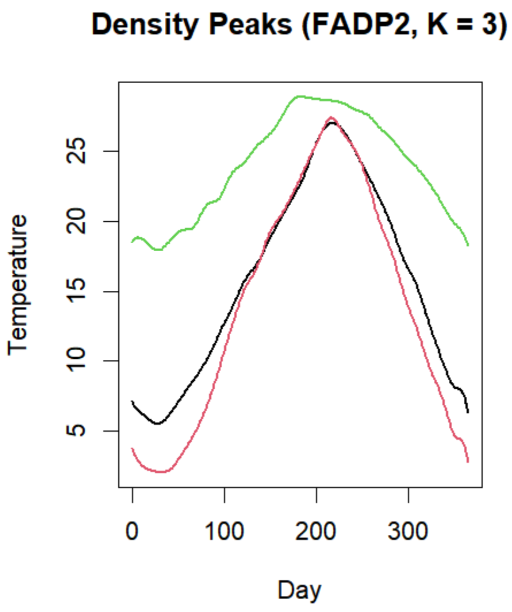


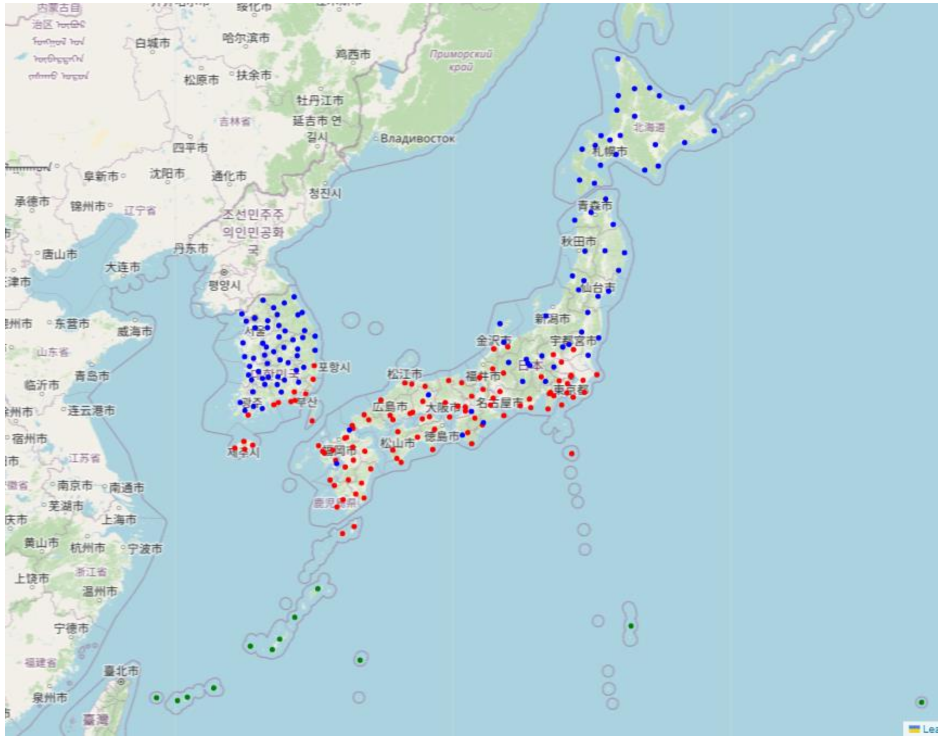


A Clustering Results

FADP2, K = 3







A Clustering Results

FADP2, K = 10