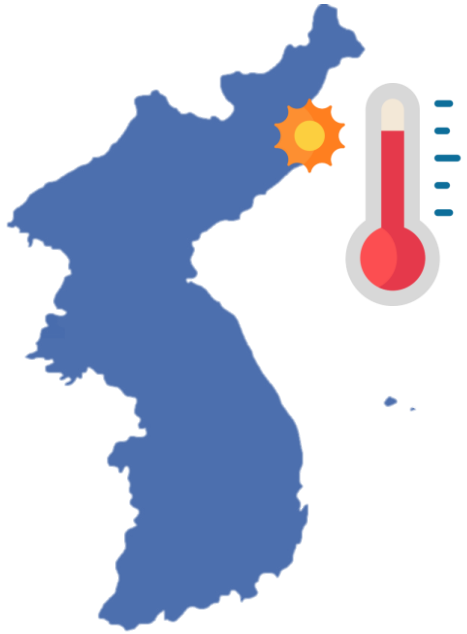
한국 기온 데이터
Functional Principal Component Analysis



1조

정영욱 최예림 오희준 배현주

INDEX

00 Introduction
분석 목적

- Functional Data Analysis에서 가장 기본적인 분석은 Principal Component Analysis

- 교재 내에서 주로 분석한 데이터는 Canadian Weather 데이터,
but 캐나다의 지형, 위치 등에 대한 배경지식 부족으로 결과의 해석에 어려움을 겪음

- 따라서, 한국의 기온 데이터를 직접 수집하여 FPCA를 적용해보고자 함

Topic:

FPCA를 통해 지역 간의 변동성을 설명하는 PC를 찾고 PC의 의미를 해석

1

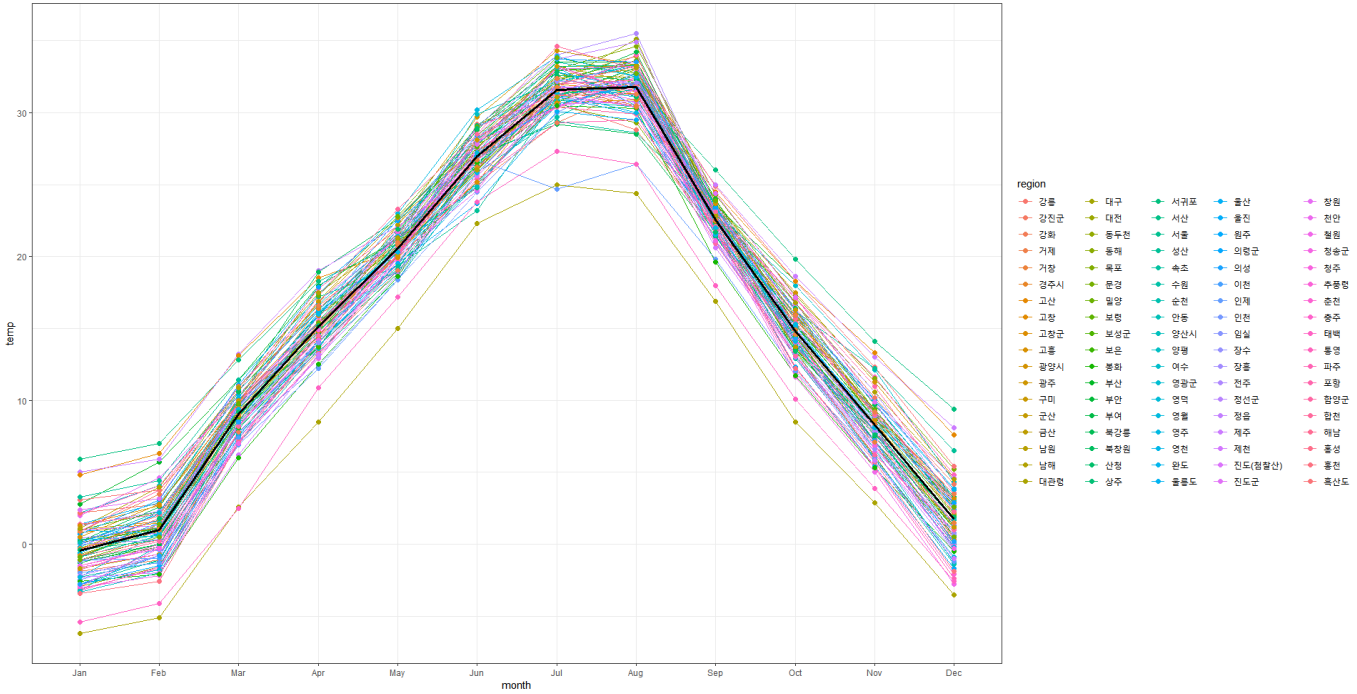
데이터 소개 및 시각화

데이터 수집

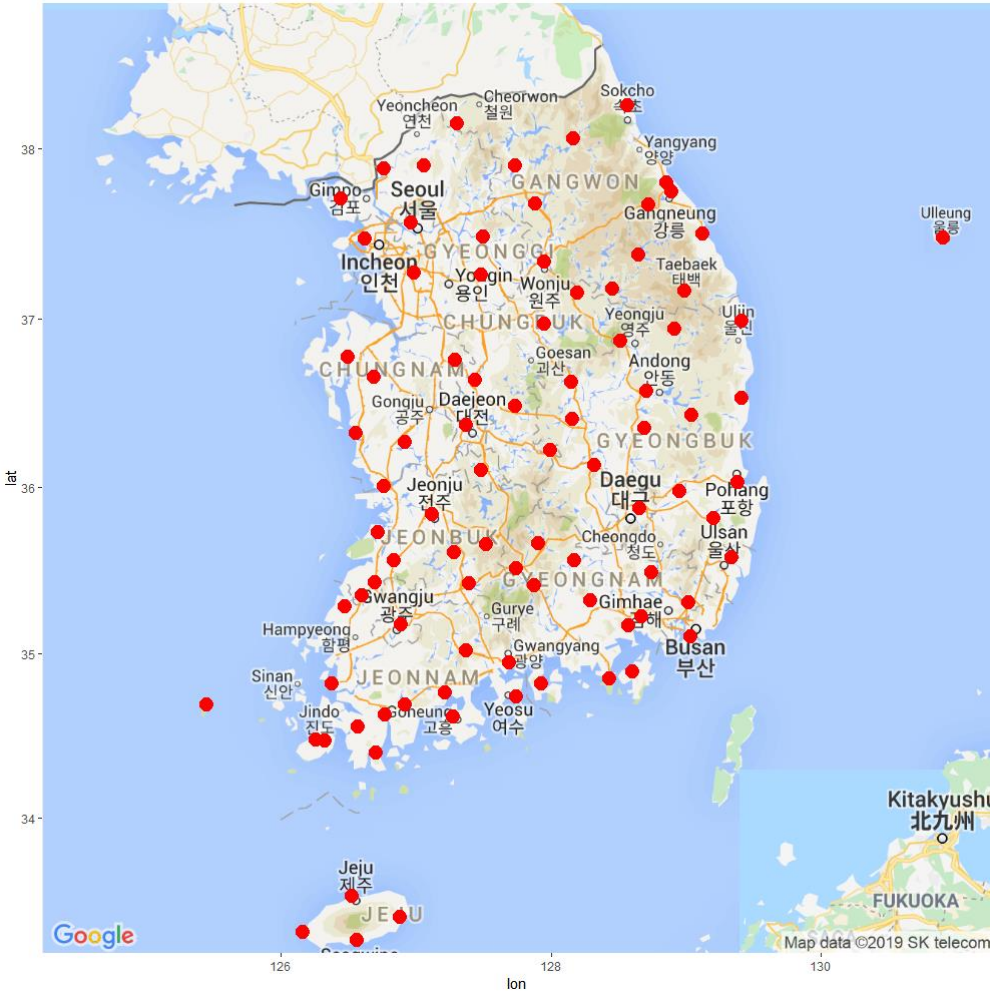
2018년 90개의 관측소에서 관측한 월별 기온 데이터 수집

데이터 출처: 기상자료개방포털
(https://data.kma.go.kr/cmmn/main.do)

Plot of Raw Data (검은색 선은 평균값)

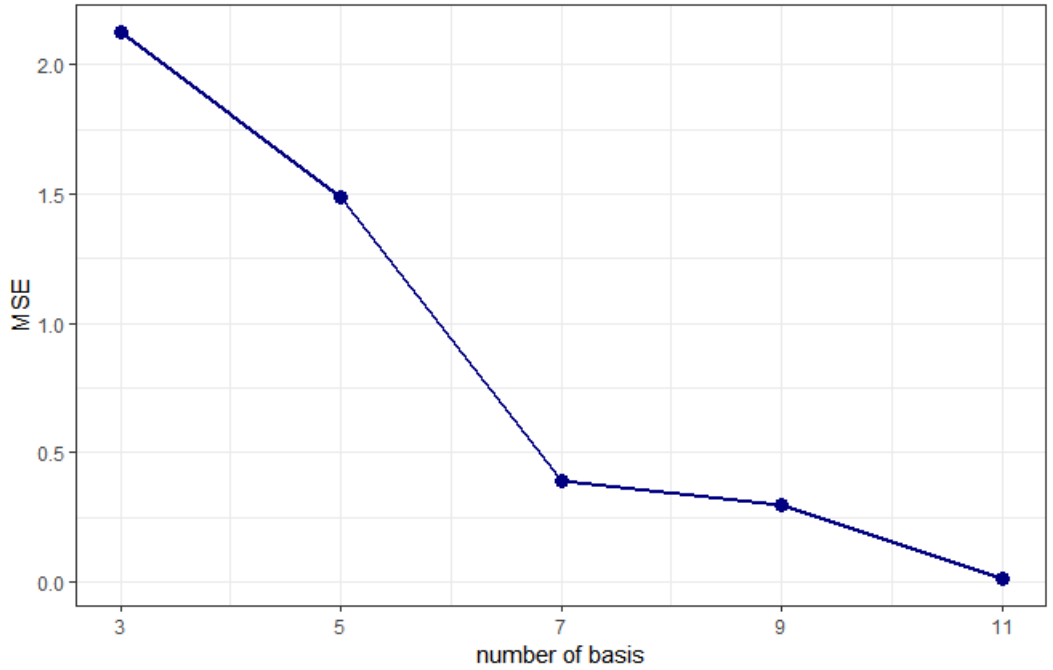


90개 관측소 위치



Smoothing

- Smooth by Fourier basis expansion
  : MSE를 통한 basis 개수 선정 (후보: 3,5,7,9,11)

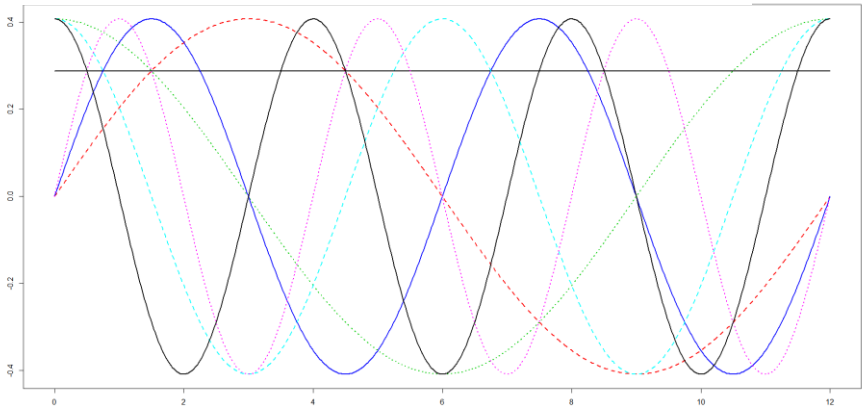


# basis	3	5	7	9	11
MSE	2.218	1.488	0.389	0.300	0.012

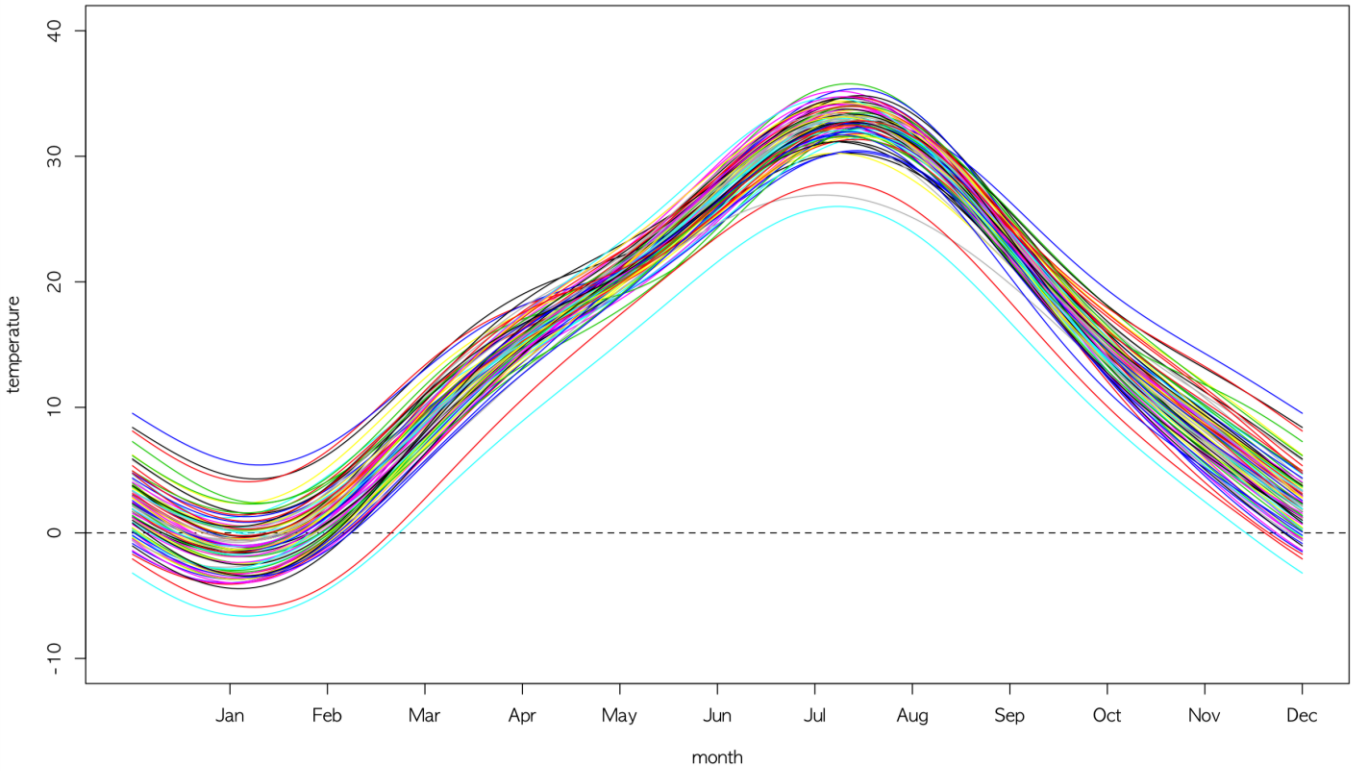
➔ MSE의 값이 가장 크게 감소하는 지점인 7개의 basis 결정 (overfitting 방지)

- Smooth by Fourier basis expansion (# of basis : 7)

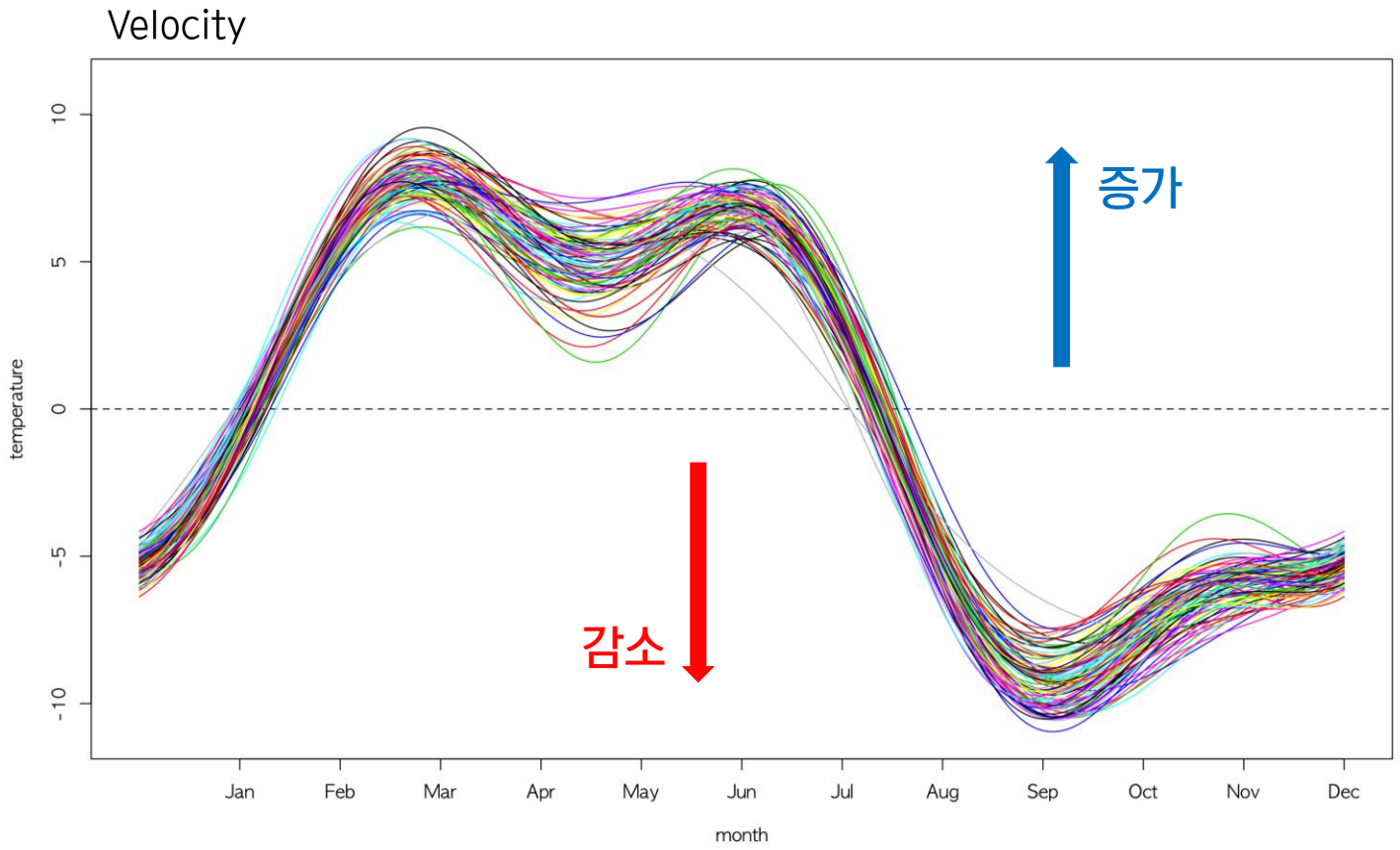
7 Fourier Basis



Smoothed Functions



1차 미분(First Derivative) Plot

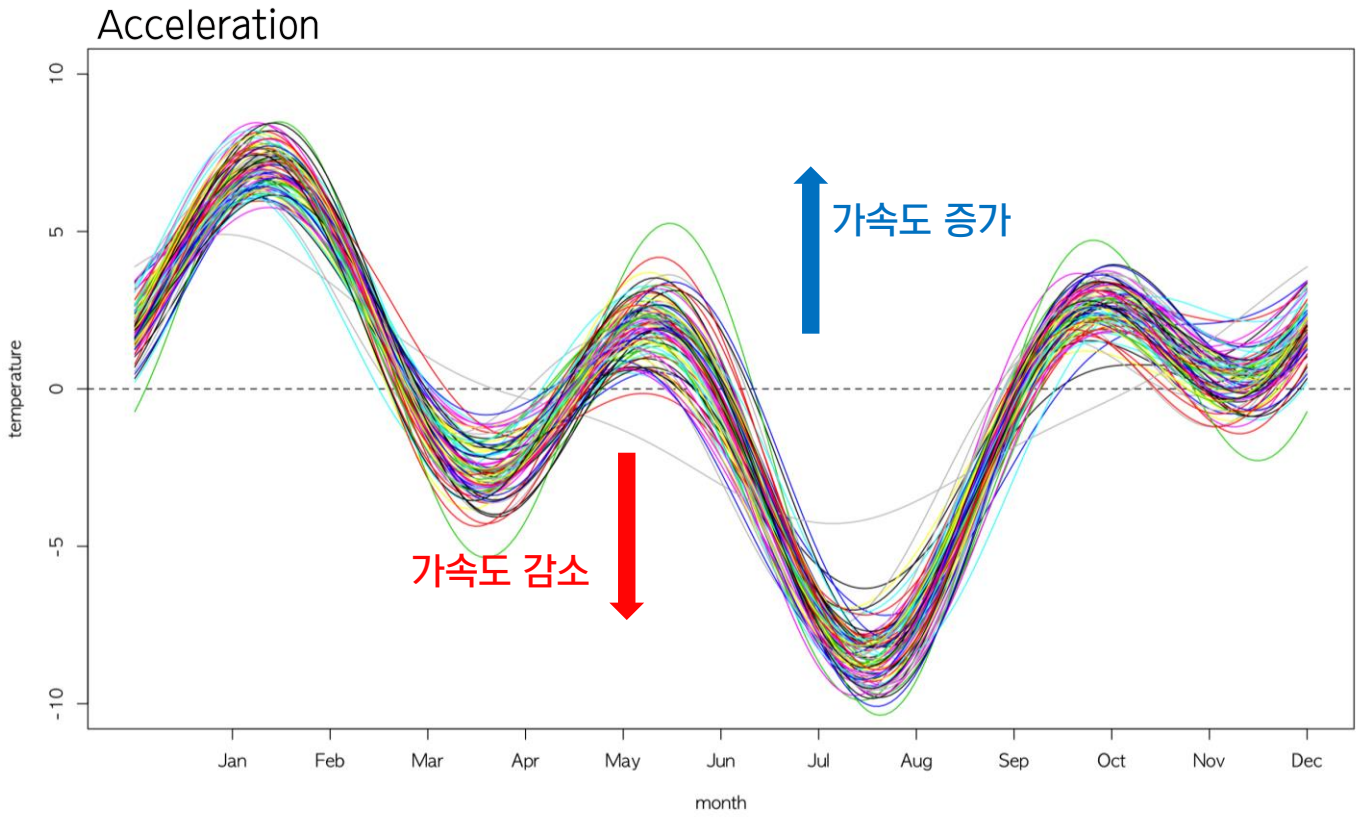


→

2월 - 7월: 기온 증가

8월 - 1월: 기온 감소

2차 미분(Second Derivative) Plot



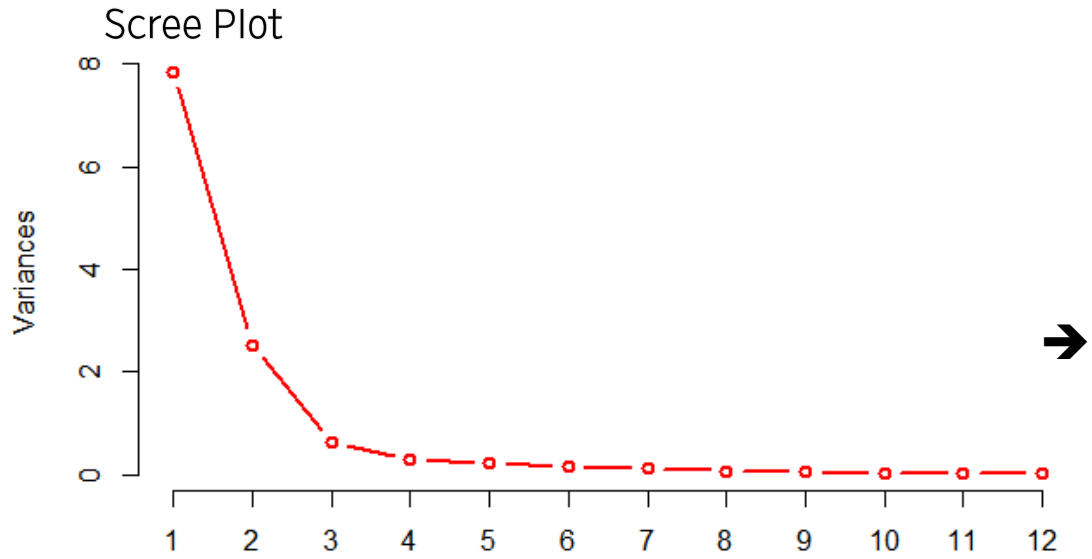
➔ 1월 - 2월 / 6월 / 10월: 양의 가속도
3월 - 4월 / 7월 - 8월 / 12월: 음의 가속도
5월 / 11월: 0에 가까운 가속도

➔ 대체적으로 1월에 기온이 빠르게 감소하고
5월에 온도가 가장 빠르게 상승한다.

2

Multivariate PCA

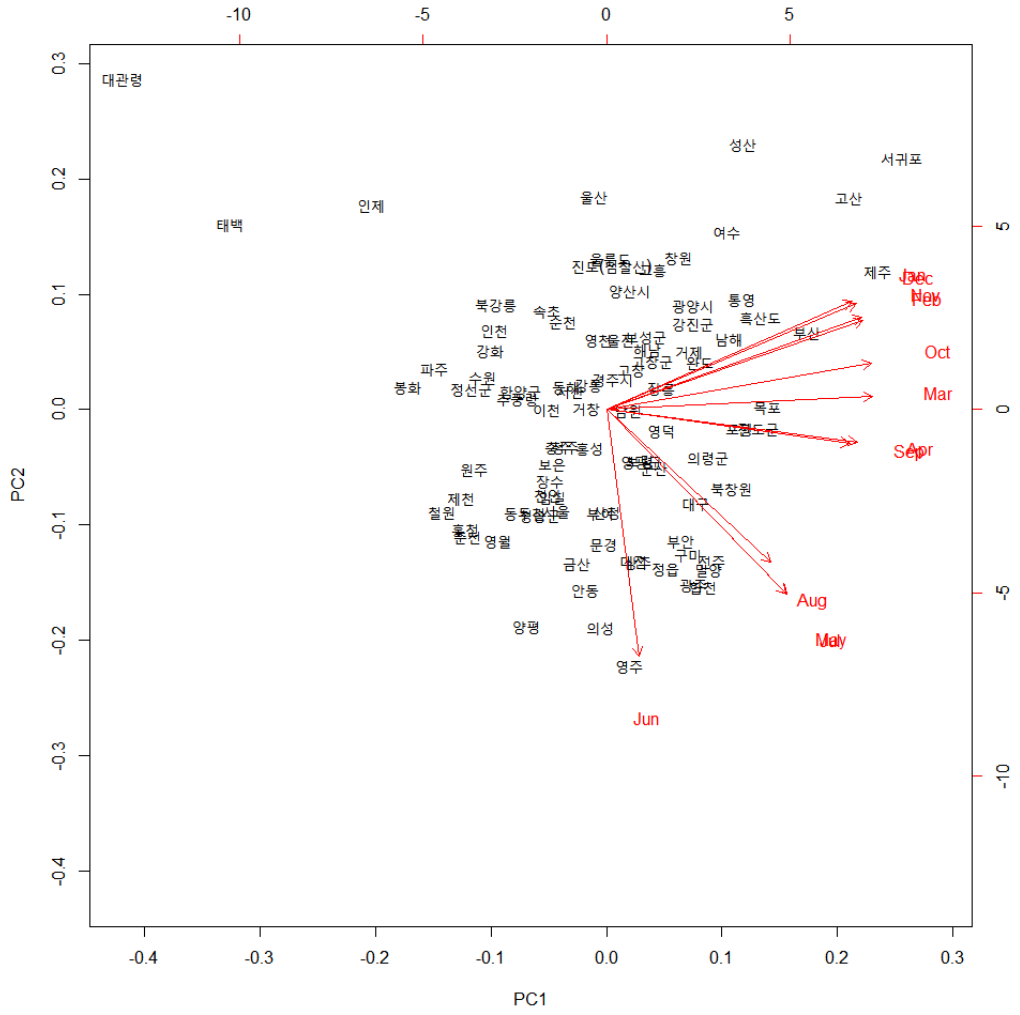
Scree Plot



➔ **주성분이 3개인 지점부터 분산의 변동성이 완만해짐**
(PC 개수 3개로 선택)

	PC1	PC2	PC3		PC11	PC12
Standard deviation	2.804071	1.592239	0.793123	...	0.154103	0.118509
Proportion of Variance	0.65523	0.21127	0.05242		0.00198	0.00117
Cumulative Proportion	0.65523	0.8665	0.91892		0.99883	1

각 PC의 분산 설명력

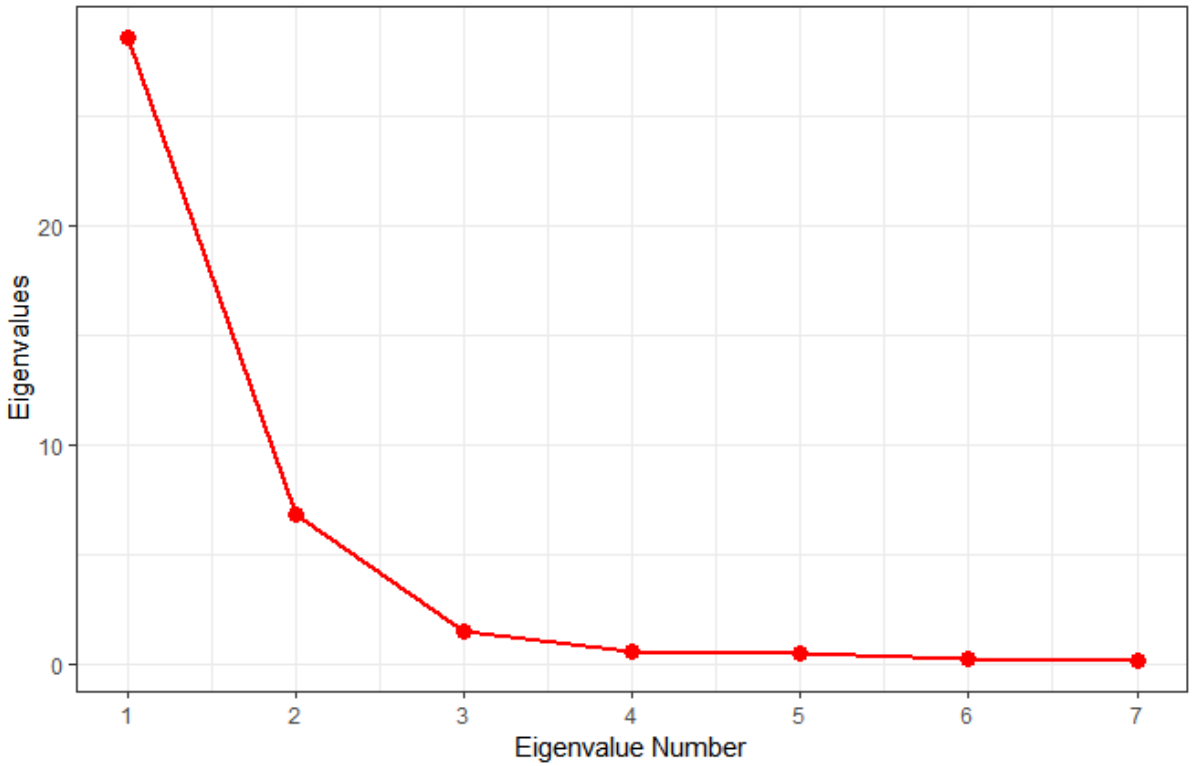


- 제 1 주성분에 대하여 4 계절이 모두 강한 양의 상관
- 제 2 주성분에 대하여
  ‘여름’의 계절이 강한 음의 상관관계를,
  ‘겨울’의 계절이 강한 양의 상관관계를 보임
- Multivariate PCA는 월(month) 사이의 변동을 설명하지만, 월 사이의 변동을 보는 것은 분석의 본래 목적과 다르며 관측소의 개수가 12보다 크기 때문에 transpose하여 PCA 불가능

3

Functional PCA

Scree Plot

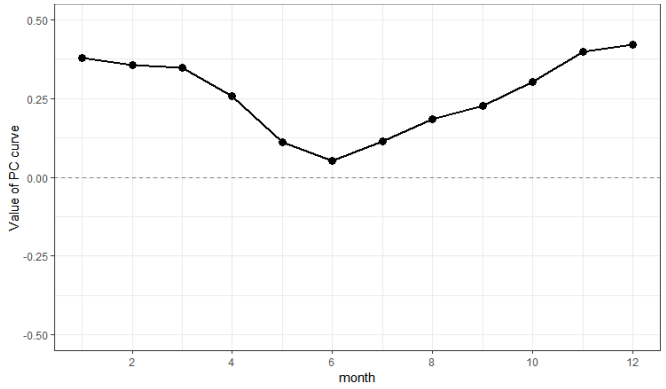


- basis의 개수에 따라 PC의 개수도 7개로 설정

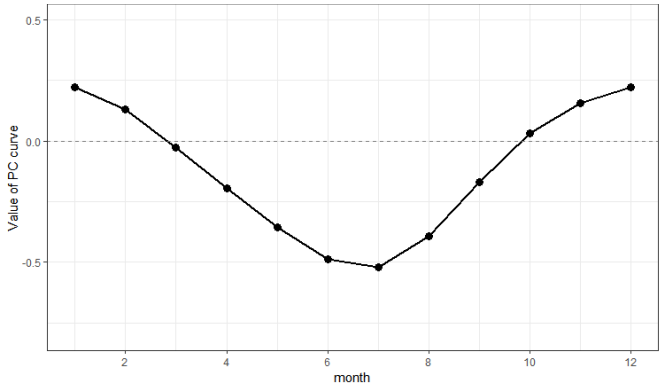
  ➔ 적절한 PC의 개수는 3개로 판단

03 Functional PCA
Eigenfunctions

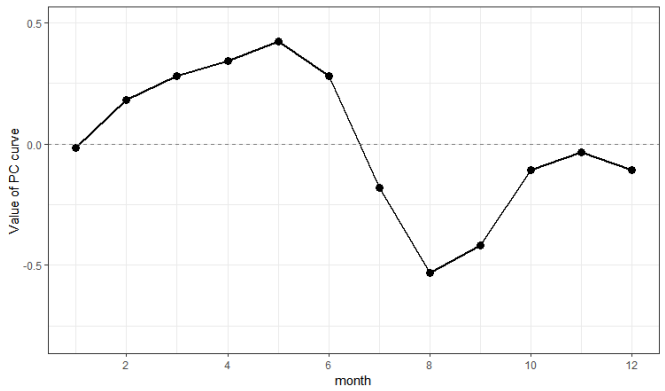
PC 1 (74.1%)



PC 2 (17.8%)

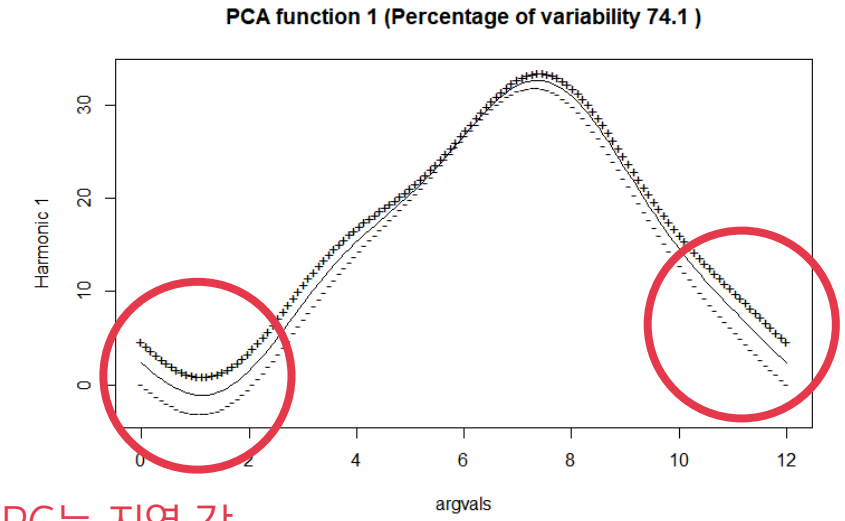


PC 3 (3.9%)

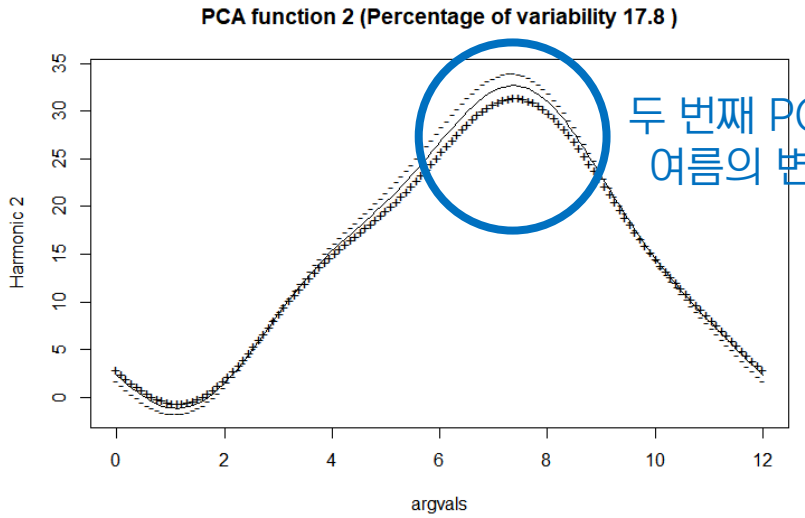


- centered된 데이터에 PCA 적용
- 3개의 PC로 전체 변동의 96% 설명
- 첫 세 개의 PC의 eigenfunction
  - 관측소 간 가장 큰 변동은 겨울에 발생
  - 첫 번째 eigenfunction이 모두 양수이기 때문에 두 번째 eigenfunction에 음수 부분 존재

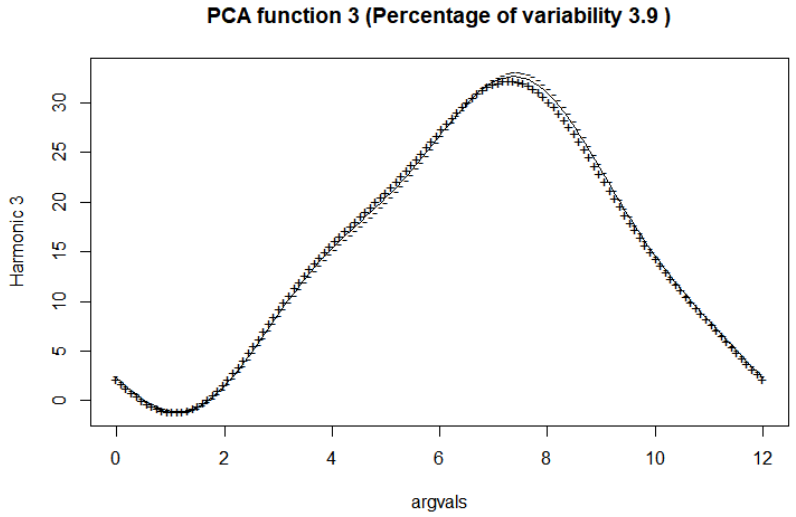
03 Functional PCA
Applying to Mean Function



첫 번째 PC는 지역 간
겨울의 변동을 설명



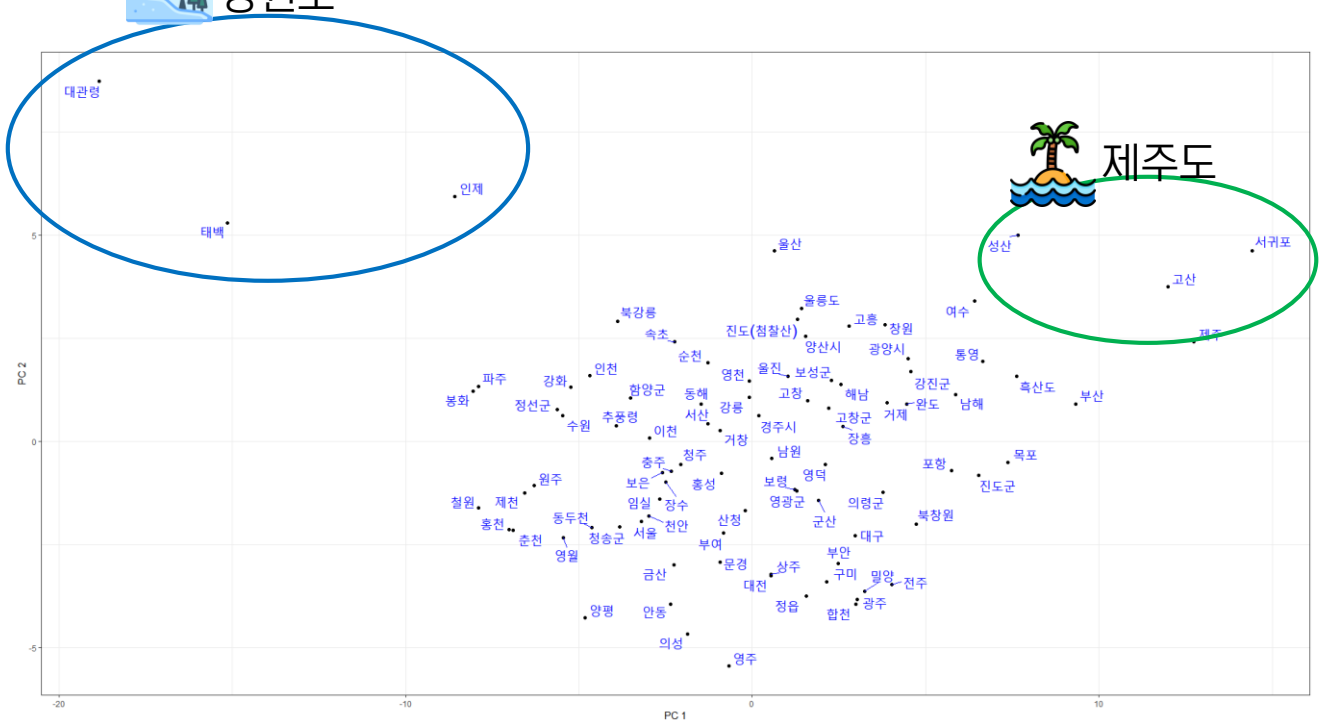
두 번째 PC는 지역 간
여름의 변동을 설명

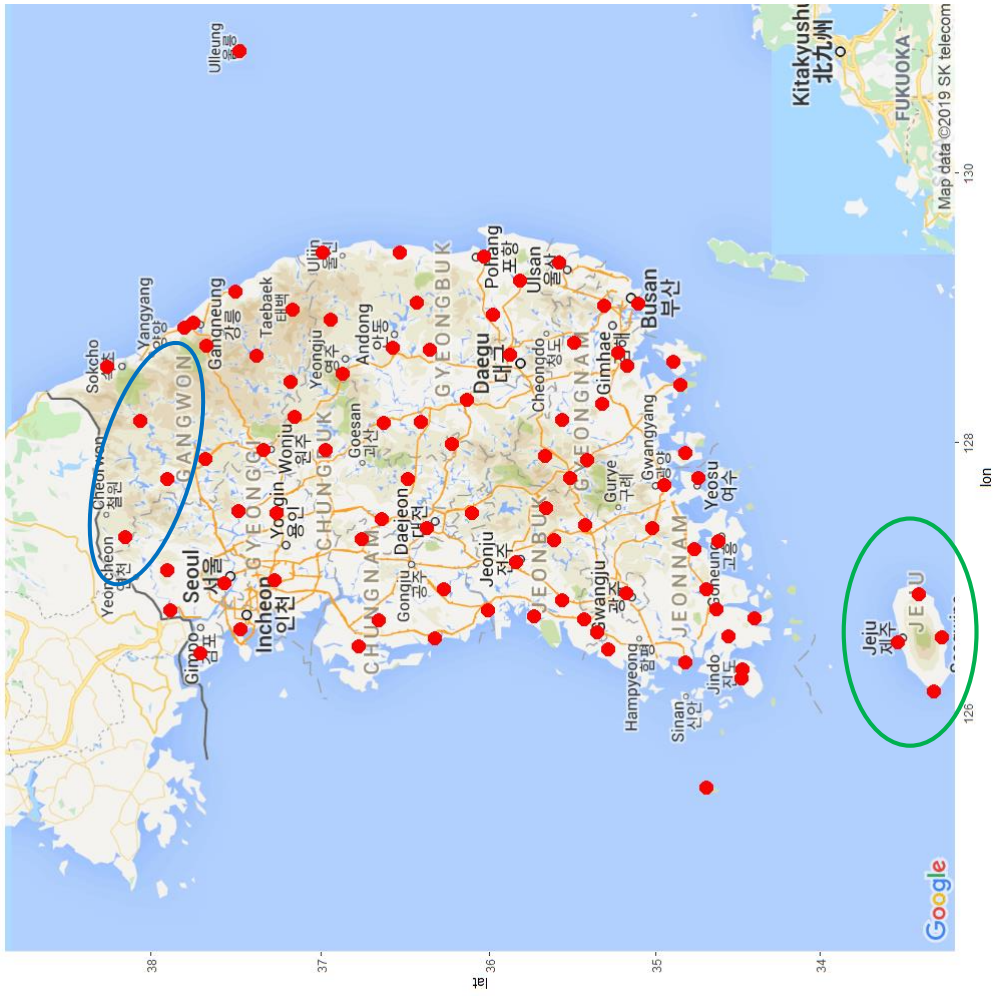


03 Functional PCA
Scores for Two PC Functions

강원도

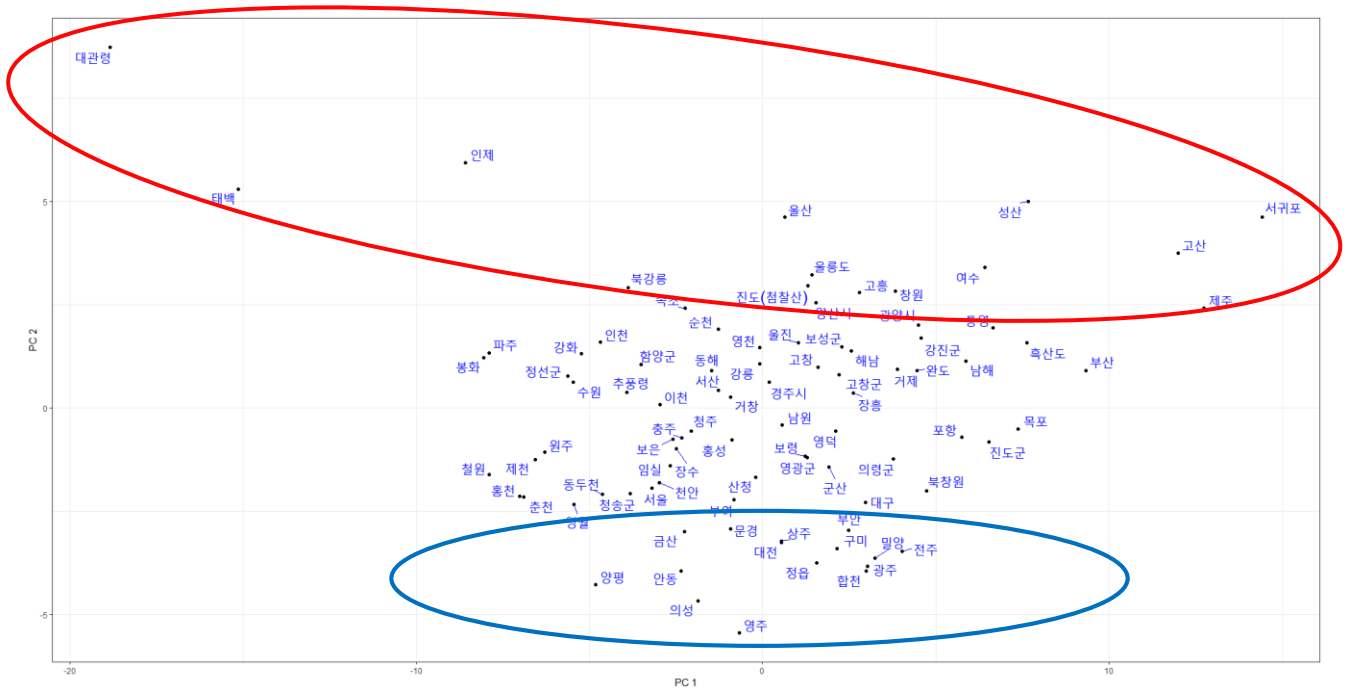
제주도





➔ 첫 번째 PC는 위도(latitude)를 반영

03 Functional PCA
Scores for Two PC Functions



Region	diff
서귀포	26.1
고산	26.7
제주	27.3
성산	28.3
인제	28.4
강릉	28.6
창원	28.9
속초	29.5
울산	29.5
고흥	29.6
강진군	29.7
북강릉	29.8
울진	29.8
여수	29.8
진도(첨찰산)	29.8

영주	34.8
문경	34.4
청송군	34.4
영덕	30.7
의성	36.3
구미	34.1
영천	30.6
경주시	31.2
거창	30.8
합천	33.4
밀양	33.6
산청	31.5
거제	30.4
남해	32

8월 기온 – 1월 기온

➔ 두 번째 PC는 여름과 겨울의 기온차를 반영

한국 기온 데이터를 이용해 Multivariate PCA와 Functional PCA 적용

Multivariate PCA

- 월(month) 간 변동을 설명하는 principal component 탐색
- 첫 번째 PC는 65%, 두 번째 PC는 21%의 변동 설명
- 월 간 변동을 보는 것은 큰 의미가 없음 (사계절 존재)
- 지역 간 변동을 보기 위해 transpose한 데이터를 사용하는 것은 불가능 ($n < p$)

Functional PCA

- 지역 간 변동을 설명하는 principal component 탐색
- 첫 번째 PC는 74.1%, 두 번째 PC는 17.8%의 변동 설명
- 첫 번째 PC는 위도를 반영, 두 번째 PC는 여름과 겨울의 기온차 반영

Appendix – R code

```
# loading packages
library(fda)
library(ggplot2)
library(ggmap)
library(reshape2)
library(dplyr)
library(ggrepel)
library(data.table)

# 0. data preprocessing
data <- read.csv("KoreanDataSheet_지점추가.csv") # 2018 data

region <- data$Region
rownames(data) <- region ; data <- data[,-1]
head(data)

# 1. Plotting
# 1.1 Raw data
data.mlt <- melt(data[,-1]) # remain only month(variable) and value(temp)
head(data.mlt)
data.mlt <- cbind(region = rep(region, 12), data.mlt)
colnames(data.mlt) <- c("region", "month", "temp")
data.mlt$region <- as.factor(data.mlt$region)
head(data.mlt)

mean.dat <- data.mlt %>% group_by(month) %>% summarise(m = mean(temp))
with(mean.dat, month <- as.character(month))

ggplot(data.mlt, aes(x = month, y = temp, group = region, color = region)) +
  geom_line() + geom_point(size = 2) + ggtitle("Raw Data") + theme_bw() +
  geom_line(data = mean.dat, aes(x = month, y = m, group = 1), inherit.aes = FALSE,
                size = 1.3)
```

```
# 1.2 Normalized data
nor_data <- t(apply(data[,-1], 1, scale))
data.mlt <- melt(nor_data)
head(data.mlt)
colnames(data.mlt) <- c("region", "month", "temp")
data.mlt$region <- as.factor(data.mlt$region)
head(data.mlt)

mean.dat <- data.mlt %>% group_by(month) %>% summarise(m = mean(temp))
with(mean.dat, month <- as.character(month))

ggplot(data.mlt, aes(x = month, y = temp, group = region, color = region)) +
  geom_line() + geom_point(size = 2) + ggtitle("Scaled Data") + theme_bw() +
  geom_line(data = mean.dat, aes(x = month, y = m, group = 1), inherit.aes = FALSE,
                size = 1.3)

# 1.3 Observatory
register_google(key='AIzaSyDLbJFdxvyywERerFK2piCIDPLjNLMIbQk')

obs <- fread("관측지점.csv")
df <- obs[종료일 == ""&지점 %in% data$RegionID, .(name = 지점명, lon = 경도, lat = 위도)]
head(df)

cen <- c(mean(df$lon),mean(df$lat))
map <- get_googlemap(center=cen,
                     maptype="roadmap",
                     zoom=7)

ggmap(map) + geom_point(data = df, aes(x = lon, y = lat), size = 4.5, col = "red")
```

Appendix – R code

```
# 2. smoothing using regression analysis
# Select the number of basis
z <- NULL
for(i in 1:5){
  monthbasis <- create.fourier.basis(c(0, 12), nbasis = (2*i +1), period = 12)
  fd_obj <- smooth.basis(1:12, t(data[,-1]), monthbasis, fdnames =
list("month","region","Deg C"))$fd
  est <- eval.fd(1:12, fd_obj)
  mse <- mean((t(data[,-1])-est)^2)
  z <- c(z, mse)
}

df <- data.frame(x = c(3,5,7,9,11), y = z)
ggplot(df,aes(x,y)) + geom_point(size = 3, col = "navy") + geom_line(size = 1,col =
"navy") +
  theme_bw() + scale_x_continuous(breaks = c(3,5,7,9,11)) + xlab("number of
basis") + ylab("MSE")

monthbasis <- create.fourier.basis(c(0, 12), nbasis = 7, period = 12)
plot(monthbasis, lwd = 2)
fd_obj <- smooth.basis(1:12, t(data[,-1]), monthbasis, fdnames =
list("month","region","Deg C"))$fd
plot(fd_obj, lty = 1)

smooth.basis.fun <- function(x) {
  result <- smooth.basis(1:12, x, monthbasis)
  return(result$fd)
}
data.smooth <- apply(data[,-1], 1, smooth.basis.fun)
# str(data.smooth)
plot(data.smooth[[1]], xlab="month", ylab="temperature",
     col=1, main="smooth temperature", ylim=c(-10, 40))
for (i in 2:nrow(data)) lines(data.smooth[[i]], col=i)
```

```
# 2.1. first derivative of smoothing data

# ex
# plot(deriv.fd(data.smooth[[1]], 1))
# plot(deriv.fd(data.smooth[[1]]))

data.smooth.1 <- list()
for (i in 1:90){
  data.smooth.1[[i]] <- deriv.fd(data.smooth[[i]], 1)
}
plot(data.smooth.1[[1]], xlab="month", ylab="temperature",
     col=1, main="the first derivative curves", ylim=c(-11, 11))
for (i in 2:90) lines(data.smooth.1[[i]], col=i)

# 2.2. second derivative of smoothing data

data.smooth.2 <- list()
for (i in 1:90){
  data.smooth.2[[i]] <- deriv.fd(data.smooth[[i]], 2)
}
plot(data.smooth.2[[1]], xlab="month", ylab="temperature",
     col=1, main="the second derivative curves", ylim=c(-10, 10))
for (i in 2:90) lines(data.smooth.2[[i]], col=i)
```

Appendix – R code

```
# 3. Principal Component Analysis
# 3.1 Multivariate PCA
pc.cr <- prcomp(data[,-1], scale. = T)
summary(pc.cr)
df <- as.data.frame(pc.cr$x[,1:2])
ggplot(df, aes(PC1, PC2))+ theme_bw() +
  geom_text(aes(label = rownames(df)), col = "purple", size= 6)

screeplot(pc.cr, main = "", col = "red", type = "lines", pch = 1, npcs =
length(pc.cr$sdev), lwd = 2)
biplot(pc.cr)

# 3.2 Functional PCA
monthbasis <- create.fourier.basis(c(0,12), nbasis = 7, period = 12)
plot(monthbasis)
fd_obj <- smooth.basis(1:12, t(data[,-1]), monthbasis, fdnames =
list("month","region","Deg C"))$fd
pca_obj <- pca.fd(fd_obj, nharm = 7, centerfns = T )

# (1) Eigenfunctions
fdmat <- as.data.frame(eval.fd(1:12, pca_obj[[1]]))
ggplot(fdmat, aes(x = 1:12, y = fdmat[,1])) + geom_point(size = 3) + geom_line(size
= 1) +
  xlab("month") + ylab("Value of PC curve") + ggtitle(paste("PC 1")) + theme_bw()+
  ylim(c(-0.5, 0.5)) + geom_abline(slope = 0, intercept = 0, linetype = "dashed", col
= "grey50")+
  scale_x_continuous(breaks = c(2,4,6,8,10,12))

ggplot(fdmat, aes(x = 1:12, y = fdmat[,2])) + geom_point(size = 3) + geom_line(size =
1) +
  xlab("month") + ylab("Value of PC curve") + ggtitle(paste("PC 2")) + theme_bw()+
  ylim(c(-0.8, 0.5)) + geom_abline(slope = 0, intercept = 0, linetype = "dashed", col =
"grey50")+
  scale_x_continuous(breaks = c(2,4,6,8,10,12))

ggplot(fdmat, aes(x = 1:12, y = fdmat[,3])) + geom_point(size = 3) + geom_line(size =
1) +
  xlab("month") + ylab("Value of PC curve") + ggtitle(paste("PC 3")) + theme_bw()+
  ylim(c(-0.8, 0.5)) + geom_abline(slope = 0, intercept = 0, linetype = "dashed", col =
"grey50")+
  scale_x_continuous(breaks = c(2,4,6,8,10,12))

# (2) Mean +/- eigenfunctions
plot.pca.fd(pca_obj)

# (3) Eigenvalues
eigvals <- pca_obj[[2]]
df <- data.frame(x = 1:7, y = eigvals)

ggplot(df, aes(x,y)) + geom_point(size = 3, col = "red") + geom_line(size = 1, col =
"red") +
  xlab("Eigenvalue Number") + ylab("Eigenvalues") + theme_bw() +
  scale_x_continuous(breaks = 1:7)

# (4) Score functions
harmscr <- data.frame(name = region, pca_obj[[3]])

ggplot(harmscr, aes(X1, X2)) + theme_bw() + xlab("PC 1") + ylab("PC 2") +
  geom_text(aes(label = name), col = "blue", size= 6)
```

Appendix – R code

```
# 4. Clustering - Region
set.seed(123)
str(data)
data.clustering <- data[, -1]

data.kmeans <- kmeans(data.clustering, centers=3)
data.kmeans$centers

data.clustering$cluster <- as.factor(data.kmeans$cluster)

ggplot(aes(data.clustering$Jan, data.clustering$Aug, colour=cluster),
      data=data.clustering) + theme_bw() +
  geom_text(label=rownames(data.clustering), nudge_x = 0.25, nudge_y = 0.25,
check_overlap = T,
            size = 7)
```