####################################################################################################

# 複数の階層型クラスター分析結果におけるクラスター同定手法「クラスター重心間距離最小法」

#

# 「標準化データ行列」(N x P)と結果「クラスターベクトルN-K」(N)を用いて クラスター同定を実施する。（ただし、

#「クラスターベクトルN-K」は「標準化データ行列」 からの生成物である。したがって、「標準化データ行列」さえあれば良い。）

#

#　 　(1)「標準化データ行列」(N x P)から「クラスター重心行列」(K x P)を作る。

# 　　　　このためには「クラスターベクトルN-K」が必要である。

# 　　(2)次に「クラスター重心行列」同士の距離行列CCV(K x K)を作る。

# 　　(3)もっとも距離が近い( min(CCV) )クラスター同士を類似クラスターとして同定する。

# 　　(4)同定されたクラスター対を対象外にして同じことを繰り返す

# 　　(5)クラスター数分類似クラスター対が同定されれば終了。

# 元データとしてe-Statから単年度データをダウンロードしたファイルを読み込んで、３年平均化する処理からスタート。

# これを元に、2017to2019(213項目)をRefND として項目数を対象データに合わせ、

# 2007to2009(207項目)と、2012to2014(208項目)をNDとして使用する。

#

# 2017to2019(213項目)から「しめじ(26B)」[90]+「えのきたけ(26X)」[91]→「他のきのこ(267)」[92]、

# 「スポーツドリンク(38Y)」[191]→「他の飲料のその他(389)」[192]、

# 「チューハイ・カクテル(3X8)」[199]→「他の酒(3XX)」[200]

# 「焼肉(399)」[208]→「洋食(395)」[207]

# 　　　　　にそれぞれ合算する。(208項目化)。さらに、

# 「ドレッシング(332)」[138]→「マヨネーズ・マヨネーズ風調味料(330)」[137]

# 　　　　　に合算する。(207項目化)

# 　　　　　　　　　 []の番号は、colnames(ssdse.p)の数字

#

# 次元合わせこみは、初めの金額値で実施している。その後、パーセント化と標準化を行う処理を実装した。

#

#######

# 本Rスクリプトを走らせるために必要なデータ

# eStat-20xx-onlyC.csv e-Statからダウンロードした20xx年単年度の家計調査データ(47都道府県 × 該当年次調査品目数)

#

# 本スクリプトで生成される生成物（行列、ベクトル、等）

# RefCN: SSDSE-2020C相当のデータをward法クラスター分析(hclust)により12クラスターに

# 分類した結果の「クラスターベクトル47-12」　　　　　　　　＜本稿の　図１　のクラスターベクトル＞

# CN\_2007to2009: e-Statからダウンロードした2007年から2009年の３年分の家計調査

#　 　　　　　　　　　　　データをward法クラスター分析(hclust)により12クラスターに分類した結果

#　 　　　　　　　　　　　の「クラスターベクトル47-12」　　　　　　　　　　＜本稿の　図２　のクラスターベクトル＞

# CN\_2007to2009\_renum:　CN-2007to2009をRefCNを参照用クラスターベクトルとして

#　 　　　　　　　　　　　　　「重心間距離最小法」によってクラスター同定しなおした

#　　　　　　　　　　　　　　　「クラスターベクトル47-12」　　　　　　　　　＜本稿の　図３　のクラスターベクトル＞

# CN\_2012to2014:　e-Statからダウンロードした2012年から2014年の３年分の家計調査

#　 　　　　　　　　　　　データをward法クラスター分析(hclust)により12クラスターに分類した結果

#　 　　　　　　　　　　　の「クラスターベクトル47-12」　　　　　　　　　　＜本稿の　図４　のクラスターベクトル＞

# CN\_2012to2014\_renum:　CN-2007to2009をRefCNを参照用クラスターベクトルとして

#　 　　　　　　　　　　　　　「重心間距離最小法」によってクラスター同定しなおした

#　　　　　　　　　　　　　　　「クラスターベクトル47-12」　　　　　　　　　＜本稿の　図５　のクラスターベクトル＞

####################################################################################################

library(dplyr)

#################################################################################################

# 元データの読み込み（e-Statの家計調査データから落とした単年度データから

# 　　　　　　　 　　　　　　　　　　　　SSDSE-C相当のデータセットを作成する）

# 2007年,2008年,2009年の単年度ファイルを読み込む(207食料品目)

ssdse\_2007 <- read.csv("eStat-2007-onlyC.csv", row.name = 1)

ssdse\_2008 <- read.csv("eStat-2008-onlyC.csv", row.name = 1)

ssdse\_2009 <- read.csv("eStat-2009-onlyC.csv", row.name = 1)

# 2012年,2013年,2014年の単年度ファイルを読み込む(208食料品目) 「ドレッシング」が追加された

ssdse\_2012 <- read.csv("eStat-2012-onlyC.csv", row.name = 1)

ssdse\_2013 <- read.csv("eStat-2013-onlyC.csv", row.name = 1)

ssdse\_2014 <- read.csv("eStat-2014-onlyC.csv", row.name = 1)

# 2017年,2018年,2019年の単年度ファイルを読み込む(213食料品目)　さらに５品目が追加された

ssdse\_2017 <- read.csv("eStat-2017-onlyC.csv", row.name = 1)

ssdse\_2018 <- read.csv("eStat-2018-onlyC.csv", row.name = 1)

ssdse\_2019 <- read.csv("eStat-2019-onlyC.csv", row.name = 1)

# ３年平均ファイルの作成

ssdse\_2007to2009 <- (ssdse\_2007 + ssdse\_2008 + ssdse\_2009)/3

ssdse\_2012to2014 <- (ssdse\_2012 + ssdse\_2013 + ssdse\_2014)/3

ssdse\_2017to2019 <- (ssdse\_2017 + ssdse\_2018 + ssdse\_2019)/3

#####################################################

######################################################

# クラスター分析（各市毎の食品支出額比率で分析）2017to2019

year\_chr <- "2017to2019"

ssdse\_colname <- colnames(ssdse\_2017to2019)

#（以下の２行はRスクリプトとしては１行であるので注意。）

ssdse.p <- (ssdse\_2017to2019 %>% select(ssdse\_colname[2]:ssdse\_colname[length(ssdse\_colname)]) %>% sweep(.,1,c(as.numeric(ssdse\_2017to2019[,1])),"/")\*100)

ssdse.std <- scale(ssdse.p)

ssdse.dist <- dist(ssdse.std)

# hclust関数のmethodでward.D2を指定することから「平方ユークリッド距離」の1/2の平方根をとる。

ssdse.dist <- sqrt(1/2)\*(ssdse.dist)

kbest.p <- 12 # クラスター数を指定

methname <- "ward.D2" # method を指定

ssdse.hclust <- hclust(ssdse.dist, method=methname)

##########################

# 「クラスターベクトル47-12」を作成する（県とクラスタ番号を切り取る）

RefCN <- cutree(ssdse.hclust, k=kbest.p)　　 # これを参照用CNとする　　　 　　＜本稿の　図１　のクラスターベクトル＞

######################################################

# クラスター分析（各市毎の食品支出額比率で分析）2007to2009

year\_chr <- "2007to2009"

ssdse\_colname\_2007to2009 <- colnames(ssdse\_2007to2009)

#（以下の３行はRスクリプトとしては１行であるので注意。）

ssdse2007to2009.p <- (ssdse\_2007to2009 %>% select(ssdse\_colname\_2007to2009[2]:ssdse\_colname\_2007to2009[length(ssdse\_colname\_2007to2009)]) %>% sweep(.,1,c(as.numeric(ssdse\_2007to2009[,1])),"/")\*100)

ssdse2007to2009.std <- scale(ssdse2007to2009.p)

ssdse2007to2009.dist <- dist(ssdse2007to2009.std)

# hclust関数のmethodでward.D2を指定することから「平方ユークリッド距離」の1/2の平方根をとる。

ssdse2007to2009.dist <- sqrt(1/2)\*(ssdse2007to2009.dist)

ssdse2007to2009.hclust <- hclust(ssdse2007to2009.dist, method=methname)

##########################

# 「クラスターベクトル47-12」を作成する（県とクラスタ番号を切り取る）

CN\_2007to2009 <- cutree(ssdse2007to2009.hclust, k=kbest.p)　 # これを対象用CNとする＜本稿の　図2　のクラスターベクトル＞

#############################################################################################

# RefNDを対象データND2007to2009の変数サイズに合わせこむ

ssdse\_2017to2019\_deitemFor2007to2009 <- ssdse\_2017to2019

#（以下の２行はRスクリプトとしては１行であるので注意。）

ssdse\_2017to2019\_deitemFor2007to2009[, "X267.他のきのこ.2014年までは26B.26Xを含む..円."] <- ssdse\_2017to2019[,"X26B.しめじ.円."]+ssdse\_2017to2019[, "X267.他のきのこ.2014年までは26B.26Xを含む..円." ]

#（以下の2行はRスクリプトとしては１行であるので注意。）

ssdse\_2017to2019\_deitemFor2007to2009[, "X267.他のきのこ.2014年までは26B.26Xを含む..円."] <- ssdse\_2017to2019[,"X26X.えのきたけ.円."]+ssdse\_2017to2019[, "X267.他のきのこ.2014年までは26B.26Xを含む..円." ]

#（以下の2行はRスクリプトとしては１行であるので注意。）

ssdse\_2017to2019\_deitemFor2007to2009[, "X389.他の飲料のその他.2014年までは38Yを含む..円."] <- ssdse\_2017to2019[,"X38Y.スポーツドリンク.円." ]+ssdse\_2017to2019[, "X389.他の飲料のその他.2014年までは38Yを含む..円."]

#（以下の２行はRスクリプトとしては１行であるので注意。）

ssdse\_2017to2019\_deitemFor2007to2009[, "X3XX.他の酒.2014年までは3X8を含む..円."] <- ssdse\_2017to2019[,"X3X8.チューハイ.カクテル.円."]+ssdse\_2017to2019[, "X3XX.他の酒.2014年までは3X8を含む..円."]

#（以下の２行はRスクリプトとしては１行であるので注意。）

ssdse\_2017to2019\_deitemFor2007to2009[, "X395.洋食.2014年までは399を含む..円."] <- ssdse\_2017to2019[,"X399.焼肉.円."]+ssdse\_2017to2019[, "X395.洋食.2014年までは399を含む..円."]

#（以下の２行はRスクリプトとしては１行であるので注意。）

ssdse\_2017to2019\_deitemFor2007to2009[, "X330.マヨネーズ.マヨネーズ風調味料.円."] <- ssdse\_2017to2019[,"X332.ドレッシング.円."]+ssdse\_2017to2019[, "X330.マヨネーズ.マヨネーズ風調味料.円."]

#（以下の2行はRスクリプトとしては１行であるので注意。）

ssdse\_2017to2019\_deitemFor2007to2009\_207items <- select(ssdse\_2017to2019\_deitemFor2007to2009, -X26B.しめじ.円., -X26X.えのきたけ.円., -X38Y.スポーツドリンク.円., -X3X8.チューハイ.カクテル.円., -X399.焼肉.円., -X332.ドレッシング.円.)

ssdse\_colname\_2017to2019\_deitemFor2007to2009\_207items <- colnames(ssdse\_2017to2019\_deitemFor2007to2009\_207items)

#（以下の４行はRスクリプトとしては１行であるので注意。）

ssdse\_2017to2019\_deitemFor2007to2009\_207items.p <- (ssdse\_2017to2019\_deitemFor2007to2009\_207items %>% select(ssdse\_colname\_2017to2019\_deitemFor2007to2009\_207items[2]:ssdse\_colname\_2017to2019\_deitemFor2007to2009\_207items[length(ssdse\_colname\_2017to2019\_deitemFor2007to2009\_207items)]) %>% sweep(.,1,c(as.numeric(ssdse\_2017to2019\_deitemFor2007to2009\_207items[,1])),"/")\*100)

ssdse\_2017to2019\_deitemFor2007to2009\_207items.std <- scale(ssdse\_2017to2019\_deitemFor2007to2009\_207items.p)

##########################

# 参照用クラスター要素数ベクトル( RefCNV[i]:クラスター識別番号iのメンバー数=|{iRefCN}| )を作成する

RefCNV <- rep(0, times = kbest.p)

for(i in 1:length(RefCN)){　　　# 都道府県数分繰り返す

RefCNV[RefCN[i]] <- RefCNV[RefCN[i]] + 1

}

##########################

# 参照用クラスター重心行列(K x P)の(RefCCVM)作成　　　　　　　　　　　　 　　　　＜本稿の（式４）と（式３）＞

# 同一クラスターのstdベクトルを全て加算する

RefCCVM <- matrix(0, nrow = kbest.p, ncol = ncol(ssdse\_2017to2019\_deitemFor2007to2009\_207items.std))

for(i in 1:length(RefCN)){ # 都道府県数分繰り返す

RefCCVM[RefCN[i],] <- RefCCVM[RefCN[i],] + ssdse\_2017to2019\_deitemFor2007to2009\_207items.std[i,]

}

# 同一クラスターの平均値ベクトルを計算する

for(i in 1:kbest.p){ # クラスター数分繰り返す

RefCCVM[i,] <- RefCCVM[i,]/RefCNV[i] # クラスターの要素数で割る

}

##########################

# 対象用クラスター要素数ベクトル(CNV[i]:クラスター識別番号iのメンバー数=|{iCN}|)を作成する

CNV <- rep(0, times = kbest.p)

for(i in 1:length(CN\_2007to2009)){　　　# 都道府県数分繰り返す

CNV[CN\_2007to2009[i]] <- CNV[CN\_2007to2009[i]] + 1

}

##########################

# 対象用クラスター重心行列(K x P)の(CCVM)作成　　　　　　　　　　　　 　　　　　　＜本稿の（式２）と（式１）＞

# 同一クラスターのstdベクトルを全て加算する

CCVM <- matrix(0, nrow = kbest.p, ncol = ncol(ssdse2007to2009.std))

for(i in 1:length(CN\_2007to2009)){ # 都道府県数分繰り返す

CCVM[CN\_2007to2009[i],] <- CCVM[CN\_2007to2009[i],] + ssdse2007to2009.std[i,]

}

# 同一クラスターの平均値ベクトルを計算する

for(i in 1:kbest.p){ # クラスター数分繰り返す

CCVM[i,] <- CCVM[i,]/CNV[i] # クラスターの要素数で割る

}

##########################################################

# 参照用重心行列と対象用重心行列(K x K)間の重心距離行列を作成する　　　　　　　　　 　　　　＜本稿の（式５）＞

# CCDij = sqrt( sum( (RefCCVM[i,] - CCVM[j,])^2 ) )の処理

CCD <- matrix(0, nrow = kbest.p, ncol = kbest.p)

for(i in 1:kbest.p){

for(j in 1:kbest.p){

CCD[i,j] <- sqrt( sum( (RefCCVM[i,] - CCVM[j,])^2 ) )

}

}

##########################################################

# 重心距離行列CCD[i,j]の最小要素の行数と列数を調べる　　　　　　　　　　　 　　　　＜本稿のアルゴリズム（１）～（３）＞

# クラスター識別番号対応ベクトル(CN\_trans)をつくる

# CN\_transを用いて、対象用クラスターベクトルN-Kを変換して、

# 参照用クラスターと同じ識別番号に入れ替える

CN\_trans <- rep(0, times = kbest.p)

CN\_trans\_order <- matrix(0, nrow = kbest.p, ncol = 3) # 距離の近いクラスタ順位を記録

# CCDのコピーを作る（同定されたクラスター対に関する重心距離をmax(CCD)で上書きするため）

CCD2 <- CCD

for(z in 1:kbest.p){

# CCD2の最小要素の行と列を求める

# y[1,1]に min\_row, y[1,2]に min\_col が入る

y <- which(CCD2 == min(CCD2), arr.ind=TRUE)

# CN\_2007to2009 の　クラスター番号に対して参照クラスター番号

# を対応づける　CN\_2007to2009[x] = j のとき、CN\_trans[j] が

# 変換後のクラスター識別番号(類似対応の参照用データのクラスタ

# 識別番号i)である。

CN\_trans[y[1,2]] <- y[1,1]

# CN\_trans\_order(参照クラスタ番号,対象クラスタ番号,クラスタ重心間距離)　確認用

CN\_trans\_order[z,] <- c(y[1,1],y[1,2],min(CCD2))

# 最小距離の行と列に最大値を埋める　解決済みの類似クラスター対を対象外にするため

CCD2[y[1,1],] <- max(CCD) + 1

CCD2[,y[1,2]] <- max(CCD) + 1

}

#############################################################

# 対象クラスターベクトルn-kの変換処理 　　　　　　　 ＜本稿の　図3　のクラスターベクトルを作成する＞

# CN\_transベクトルによって CN\_2007to2009 を　類似ベクトル対応変換し、　CN\_2007to2009\_renum をつくりだす

#

CN\_2007to2009\_renum <- CN\_2007to2009

for(z in 1:length(CN\_2007to2009)){

CN\_2007to2009\_renum[z] <- CN\_trans[CN\_2007to2009[z]]

}

########################## 以上、2007to2009データの処理

############################

#####################################################

# クラスター分析（各市毎の食品支出額比率で分析）2012to2014

year\_chr <- "2012to2014"

ssdse\_colname\_2012to2014 <- colnames(ssdse\_2012to2014)

#（以下の３行はRスクリプトとしては１行であるので注意。）

ssdse2012to2014.p <- (ssdse\_2012to2014 %>% select(ssdse\_colname\_2012to2014[2]:ssdse\_colname\_2012to2014[length(ssdse\_colname\_2012to2014)]) %>% sweep(.,1,c(as.numeric(ssdse\_2012to2014[,1])),"/")\*100)

ssdse2012to2014.std <- scale(ssdse2012to2014.p)

ssdse2012to2014.dist <- dist(ssdse2012to2014.std)

# hclust関数のmethodでward.D2を指定することから「平方ユークリッド距離」の1/2の平方根をとる。

ssdse2012to2014.dist <- sqrt(1/2)\*(ssdse2012to2014.dist)

ssdse2012to2014.hclust <- hclust(ssdse2012to2014.dist, method=methname)

##########################

# 「クラスターベクトル47-12」を作成する（県とクラスタ番号を切り取る）

CN\_2012to2014 <- cutree(ssdse2012to2014.hclust, k=kbest.p) 　# これを対象用CNとする＜本稿の　図4　のクラスターベクトル＞

##########################

############################

#############################################################################################

# RefCCVMを対象データCCVM2012to2014の変数サイズに合わせこむ

ssdse\_2017to2019\_deitemFor2012to2014 <- ssdse\_2017to2019

#（以下の２行はRスクリプトとしては１行であるので注意。）

ssdse\_2017to2019\_deitemFor2012to2014[, "X267.他のきのこ.2014年までは26B.26Xを含む..円."] <- ssdse\_2017to2019[,"X26B.しめじ.円."]+ssdse\_2017to2019[, "X267.他のきのこ.2014年までは26B.26Xを含む..円." ]

#（以下の2行はRスクリプトとしては１行であるので注意。）

ssdse\_2017to2019\_deitemFor2012to2014[, "X267.他のきのこ.2014年までは26B.26Xを含む..円."] <- ssdse\_2017to2019[,"X26X.えのきたけ.円."]+ssdse\_2017to2019[, "X267.他のきのこ.2014年までは26B.26Xを含む..円." ]

#（以下の2行はRスクリプトとしては１行であるので注意。）

ssdse\_2017to2019\_deitemFor2012to2014[, "X389.他の飲料のその他.2014年までは38Yを含む..円."] <- ssdse\_2017to2019[,"X38Y.スポーツドリンク.円." ]+ssdse\_2017to2019[, "X389.他の飲料のその他.2014年までは38Yを含む..円."]

#（以下の２行はRスクリプトとしては１行であるので注意。）

ssdse\_2017to2019\_deitemFor2012to2014[, "X3XX.他の酒.2014年までは3X8を含む..円."] <- ssdse\_2017to2019[,"X3X8.チューハイ.カクテル.円."]+ssdse\_2017to2019[, "X3XX.他の酒.2014年までは3X8を含む..円."]

#（以下の２行はRスクリプトとしては１行であるので注意。）

ssdse\_2017to2019\_deitemFor2012to2014[, "X395.洋食.2014年までは399を含む..円."] <- ssdse\_2017to2019[,"X399.焼肉.円."]+ssdse\_2017to2019[, "X395.洋食.2014年までは399を含む..円."]

#（以下の２行はRスクリプトとしては１行であるので注意。）

ssdse\_2017to2019\_deitemFor2012to2014\_208items <- select(ssdse\_2017to2019\_deitemFor2012to2014, -X26B.しめじ.円., -X26X.えのきたけ.円., -X38Y.スポーツドリンク.円., -X3X8.チューハイ.カクテル.円., -X399.焼肉.円.)

ssdse\_colname\_2017to2019\_deitemFor2012to2014\_208items <- colnames(ssdse\_2017to2019\_deitemFor2012to2014\_208items)

#（以下の４行はRスクリプトとしては１行であるので注意。）

ssdse\_2017to2019\_deitemFor2012to2014\_208items.p <- (ssdse\_2017to2019\_deitemFor2012to2014\_208items %>%

select(ssdse\_colname\_2017to2019\_deitemFor2012to2014\_208items[2]:ssdse\_colname\_2017to2019\_deitemFor2012to2014\_208items[length(ssdse\_colname\_2017to2019\_deitemFor2012to2014\_208items)]) %>% sweep(.,1,c(as.numeric(ssdse\_2017to2019\_deitemFor2012to2014\_208items[,1])),"/")\*100)

ssdse\_2017to2019\_deitemFor2012to2014\_208items.std <- scale(ssdse\_2017to2019\_deitemFor2012to2014\_208items.p)

##########################

# 参照用クラスター要素数ベクトル(RefCNV:クラスター識別番号毎の要素数)を作成する

RefCNV <- rep(0, times = kbest.p)

for(i in 1:length(RefCN)){　　　# 都道府県数分繰り返す

RefCNV[RefCN[i]] <- RefCNV[RefCN[i]] + 1

}

##########################

# 参照用クラスター重心行列(K x P)の(RefCCVM)作成 　　　　　　　　 　 　　 ＜本稿の（式４）と（式３）＞

# 同一クラスターのstdベクトルを全て加算する

RefCCVM <- matrix(0, nrow = kbest.p, ncol = ncol(ssdse\_2017to2019\_deitemFor2012to2014\_208items.std))

for(i in 1:length(RefCN)){ # 都道府県数分繰り返す

RefCCVM[RefCN[i],] <- RefCCVM[RefCN[i],] + ssdse\_2017to2019\_deitemFor2012to2014\_208items.std[i,]

}

# 同一クラスターの平均値ベクトルを計算する

for(i in 1:kbest.p){ # クラスター数分繰り返す

RefCCVM[i,] <- RefCCVM[i,]/RefCNV[i] # クラスターの要素数で割る

}

##########################

# 対象用クラスター要素数ベクトル(CNV:クラスター識別番号毎の要素数)を作成する

CNV <- rep(0, times = kbest.p)

for(i in 1:length(CN\_2012to2014)){　　　# 都道府県数分繰り返す

CNV[CN\_2012to2014[i]] <- CNV[CN\_2012to2014[i]] + 1

}

##########################

# 対象用クラスター重心行列(K x P)の(CCVM)作成 　　　　　　　　　　　　　 　　　　＜本稿の（式２）と（式１）＞

# 同一クラスターのstdベクトルを全て加算する

CCVM <- matrix(0, nrow = kbest.p, ncol = ncol(ssdse2012to2014.std))

for(i in 1:length(CN\_2012to2014)){ # 都道府県数分繰り返す

CCVM[CN\_2012to2014[i],] <- CCVM[CN\_2012to2014[i],] + ssdse2012to2014.std[i,]

}

# 同一クラスターの平均値ベクトルを計算する

for(i in 1:kbest.p){ # クラスター数分繰り返す

CCVM[i,] <- CCVM[i,]/CNV[i] # クラスターの要素数で割る

}

##############################################################################################

##########################################################

# 参照用重心行列と対象用重心行列(K x K)間の重心距離行列を作成する 　　　　　　　　　 　　　　　＜本稿の（式５）＞

# CCDij = sqrt( sum( (RefCCVM[i,] - CCVM[j,])^2 ) )

#

CCD <- matrix(0, nrow = kbest.p, ncol = kbest.p)

for(i in 1:kbest.p){

for(j in 1:kbest.p){

CCD[i,j] <- sqrt( sum( (RefCCVM[i,] - CCVM[j,])^2 ) )

}

}

##########################################################

# 重心距離行列CCD[i,j]の最小要素の行数と列数を調べる　　　　　　　　　　 　　　＜本稿のアルゴリズム（１）～（３）＞

#

# クラスター識別番号対応ベクトル(CN\_trans)をつくる

# CN\_transを用いて、対象用クラスターベクトルn-kを変換して、

# 参照用クラスターと同じ識別番号に入れ替える

CN\_trans <- rep(0, times = kbest.p)

CN\_trans\_order <- matrix(0, nrow = kbest.p, ncol = 3) # 距離の近いクラスタ順位を記録

# CCDのコピーを作る（同定されたクラスター対に関する重心距離をmax(CCD)で上書きするため）

CCD2 <- CCD

for(z in 1:kbest.p){

# CCD2の最小要素の行と列を求める

# y[1,1]に min\_row, y[1,2]に min\_col が入る

y <- which(CCD2 == min(CCD2), arr.ind=TRUE)

# CN\_2012to2014 の　クラスター番号に対して参照クラスター番号

# を対応づける　CN\_2012to2014[x] = j のとき、CN\_trans[j] が

# 変換後のクラスター識別番号(類似対応の参照用データのクラスタ

# 識別番号i)である。

CN\_trans[y[1,2]] <- y[1,1]

# CN\_trans\_order(参照クラスタ番号,対象クラスタ番号,クラスタ重心間距離)　確認用

CN\_trans\_order[z,] <- c(y[1,1],y[1,2],min(CCD2))

# 最小距離の行と列に最大値を埋める　解決済みの類似クラスター対を対象外にするため

CCD2[y[1,1],] <- max(CCD) + 1

CCD2[,y[1,2]] <- max(CCD) + 1

}

#############################################################

# 対象クラスターベクトルN-Kの変換処理　　　　　　　　 　　＜本稿の　図5　のクラスターベクトルを作成する＞

# CN\_transベクトルによって CN\_2012to2014 を　類似ベクトル対応変換し、　CN\_2012to2014\_renum をつくりだす

#

CN\_2012to2014\_renum <- CN\_2012to2014

for(z in 1:length(CN\_2012to2014)){

CN\_2012to2014\_renum[z] <- CN\_trans[CN\_2012to2014[z]]

}

##########################　以上、2012to2014データの処理

############################