Análisis de Series Temporals usando R: Cluster, Clasificación y Contraste de Hipótesis

by

Pablo Montero José A. Vilar

26 Outubro, 2017

Financiación

- Ministerio de Economía y Competitividad
  MTM2014-52876-R
- Xunta de Galicia ED431C 2016-015

1. Cluster
2. Clasificación
3. Contraste de Hipótesis

Cluster

1. Propósito del Cluster
2. Métodos Cluster en Series Temporales
3. Validación
    - Con valores verdaderos
    - Sin valores verdaderos

Propósito de Cluster

No hay una definición globalmente aceptada sobre qué es cluster

- “Agrupar los datos de manera que los objetos dentro de cada grupo son más similares entre sí que a los de los otros grupos”
- “Extraer patrones interesantes de los datos”
- “Reducir la complejidad de los datos”
- Christian Hennig

Cluster Basado en Distancias

Contamos con alguna manera para medir cuánto se parecen dos series.
Muchos métodos de cluster se reducen a trabajar con estas comparaciones directamente, no con los datos originales.

- K-means
- K-medoids (PAM)
- Jerárquico
- Versiones Fuzzy

Distancias Entre Series Temporales

Podemos categorizar las distancias en las siguientes familias

- Basadas en Modelos
  - Ajustar un modelo de series temporales para cada serie, y se comparan los modelos
  - Interpretabilidad
- Basadas en características o datos en bruto
- Basadas en Complejidad
  - Teoría de la información: Información Conjunta
- Basadas en Pronóstico (forecast)
  - Característico de Series Temporales
  - Asumen algún tipo de modelo para las series

Práctica: Distancia + Método de Cluster

Métodos de Cluster

- K-means
  - Cuidado en como se calcula el centro del grupo
  - Hacer la media de características o modelos suele no tener sentido
  - Cada distancia necesita de su método para obtener el centro, fuera de la distancia euclídea no es trivial
- K-medoides
  - Se substituye el centro por un centroide, un dato existente en la muestra
  - Permite trabajar con disimilaridades en general
  - Más robusto a atípicos
- Jerárquico
  - Permite trabajar con disimilaridades en general
  - Suele producir resultados interpretables que ayudan a su vez a escoger el número de clusters

Distancias basadas en modelos

1. Se asume un modelo o familia de modelos
2. Se ajustan los parámetros del modelo para cada serie
   - Idealmente de manera automática
3. Se comparan los parámetros del modelo
   - Distancia entre parámetros
   - P-valores al contrastar igualdad de modelos

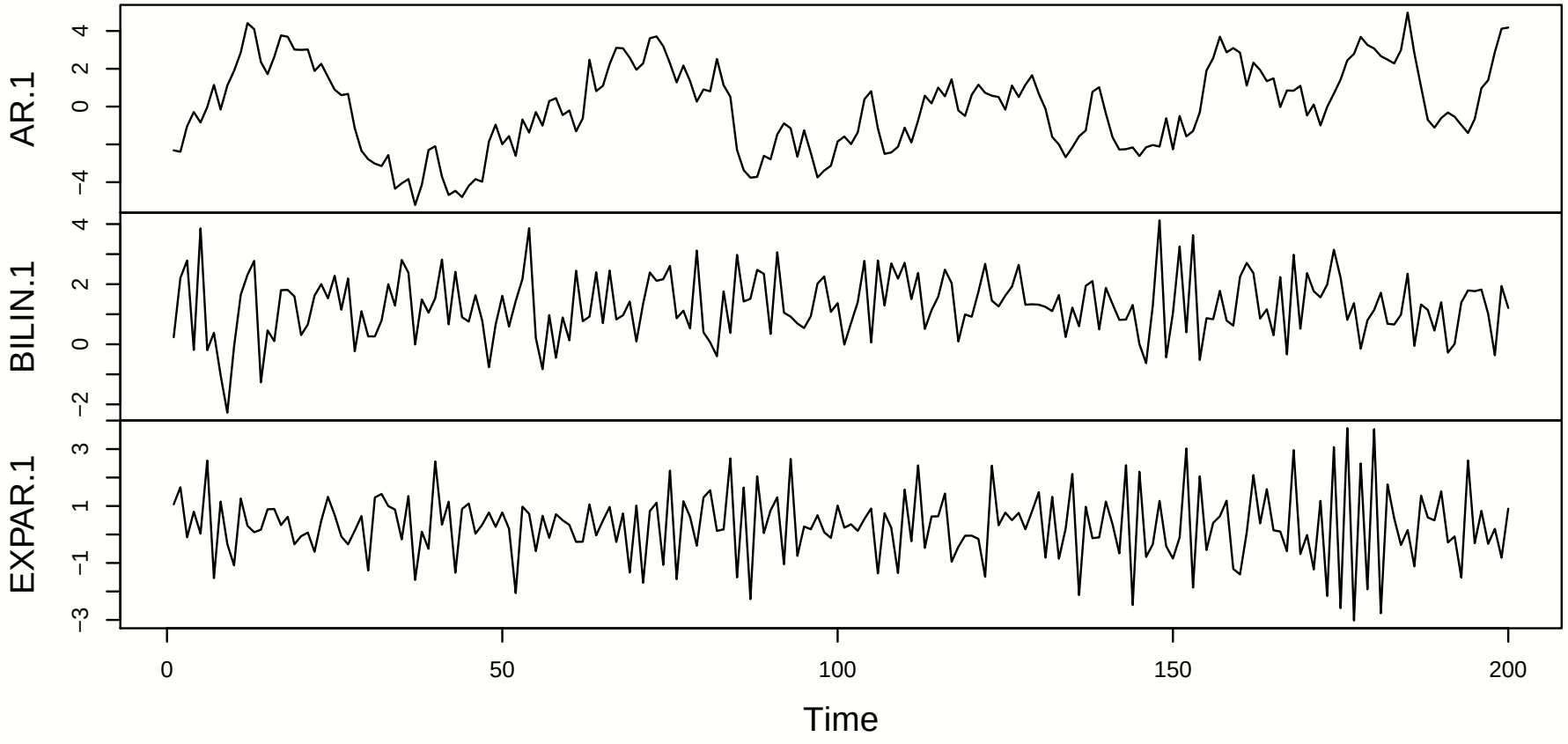
- AR
- ARMA
- ARIMA
- Modelos autoregresivos no-paramétricos

Permiten al usuario capturar o despreciar comportamientos que no le interesan, como la parte estacional de las series

Ejemplo R

```r
#paquete TSclust contiene distancias entre series,
# métodos de cluster, validación
#y datasets de ejemplo
library(TSclust)

#cargamos un dataset de datos simulados
data(synthetic.tseries)

#pintamos unas series
plot(synthetic.tseries[,c(1,4,7)])
```

Series Simuladas de Modelos Autoregresivos



Ejemplo R

```
#aplicamos una distancia usando diss():
#diss() aplica una distancia de TSclust
#a todos los pares posibles de series
# en nuestro dataset
#y devuelve un objeto dist de R
#que suelen usar los métodos de cluster

dCEPS <- diss(synthetic.tseries, "AR.LPC.CEPS")

#tomamos una distancia basada en modelos autoregres

#con el objeto dist dCEPS podemos hacer cluster
#por ejemplo cluster jerárquico
#se presta a la comparación visual

hcCEPS <- hclust(dCEPS)
```

Distancias entre características o datos en bruto

Características como:

- Autocorrelaciones
- Periodograma
- Correlación temporal entre series
- Densidad Espectral

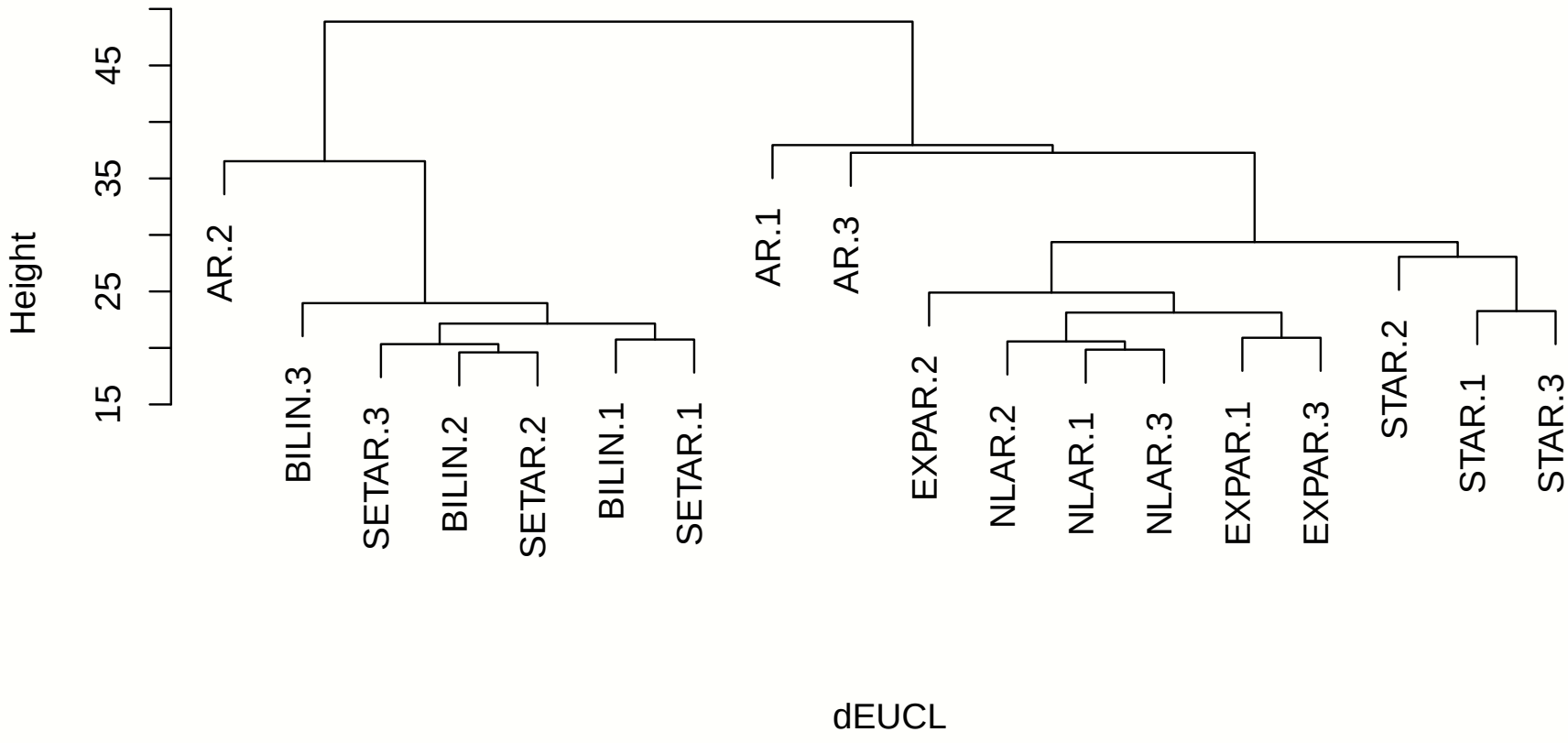
Datos en bruto:

- Distancia Euclídea
- Dynamic Time Warping
- Distancia Levehshtein (Edit Distance) (paquete TSdist)
- Longest Common Subsequences (paquete TSdist)

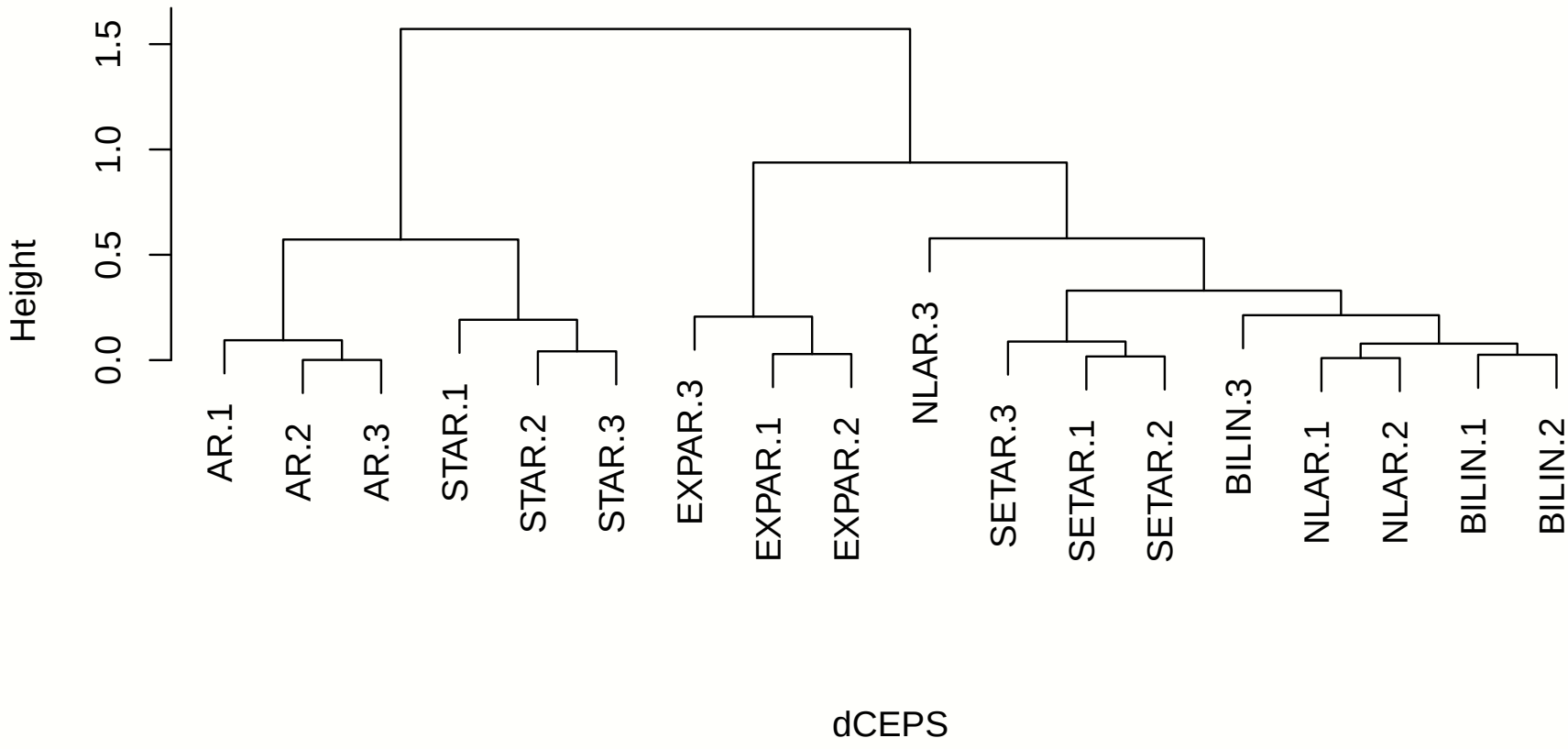
Ejemplo R

```
#apliquemos una distancia de datos en bruto
# como la Euclídea
dEUCL <- diss(synthetic.tseries, "EUCL")

#aplicamos el mismo método de cluster
# que con la distancia basada en modelos
hcEUCL <- hclust(dEUCL)

#comparemos visualmente las dos
plot(hcEUCL)
plot(hcCEPS)
```

Resultado con distancia Euclídea



Resultado con distancia basada en modelo AR



Ejemplo R

```
#las distancias en TSclust tienen su
# función individual
#documentada y con referencia al artículo
# en el que fueron publicadas
?diss.AR.MAH

#esta distancia basada en modelos
# produce un p-valor
#del contraste que compara si dos series
#tienen el mismo modelo
dMAH <- diss(synthetic.tseries, "AR.MAH")
```

Ejemplo R

```
#TSclust tiene un método de cluster
# basado en p-valores
#que agrupa fijando un nivel de significación

#interés por interpretabilidad
#en función de la significación
#crea automáticamente
#los clusters sin evidencias para separarlos

pvalues.clust(dMAH$p_value, 0.01)
[1] 3 3 3 1 1 1 2 2 1 1 1 1 1 1 1 3 3 3

#menos conservador -> más clusters
pvalues.clust(dMAH$p_value, 0.05)
[1] 2 2 2 1 1 1 3 3 1 4 4 1 1 1 1 4 4 4
```

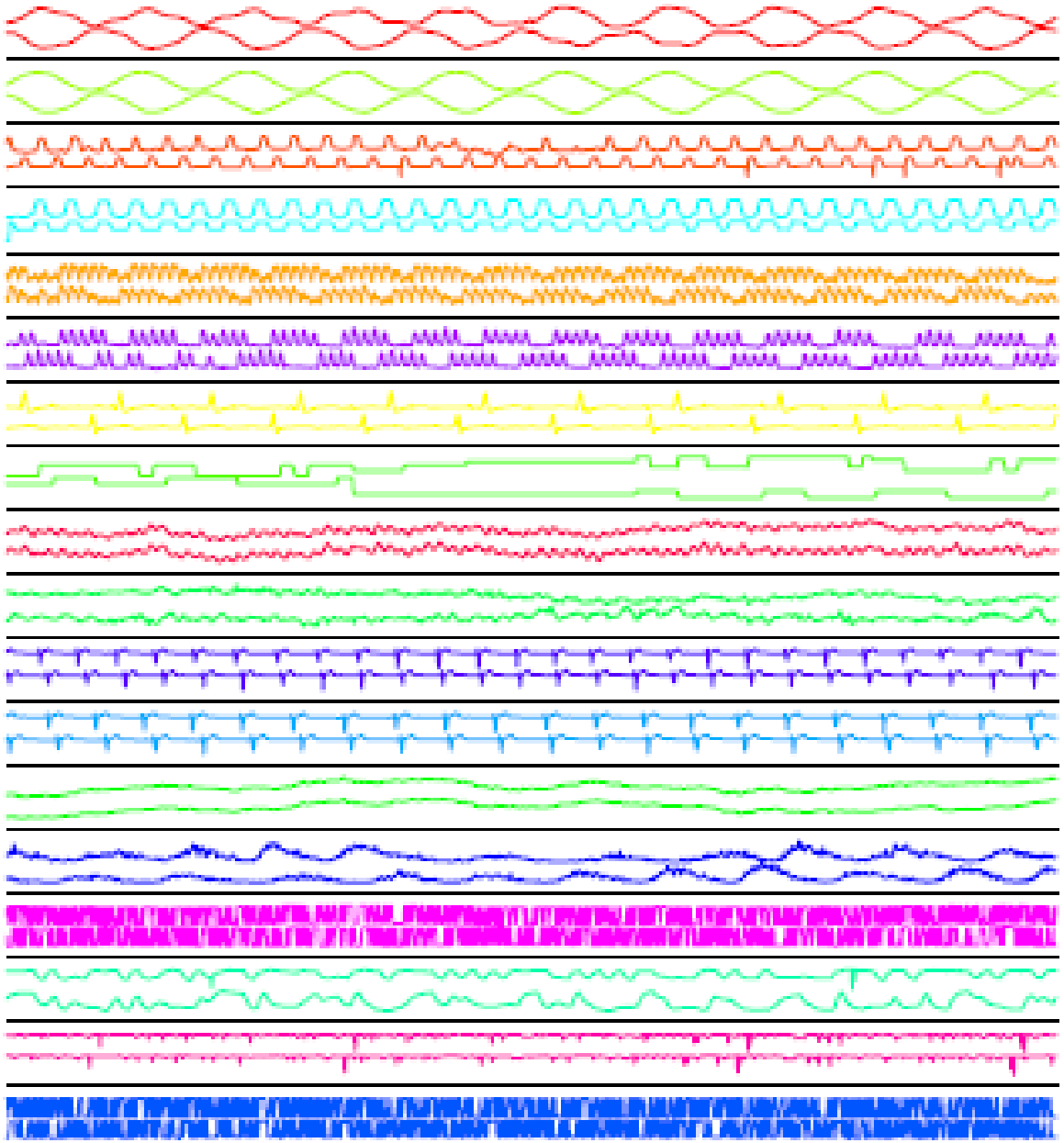
Distancias basadas en Complejidad

- Basadas en compresión para aproximar la complejidad
  - Número de bits al comprimir las dos series concatenadas
  - Relativo al número de bits al comprimir las series por separado
- Basada en la distribución de permutaciones
  - Qué maneras hay de ordenar las subsecuencias de las series

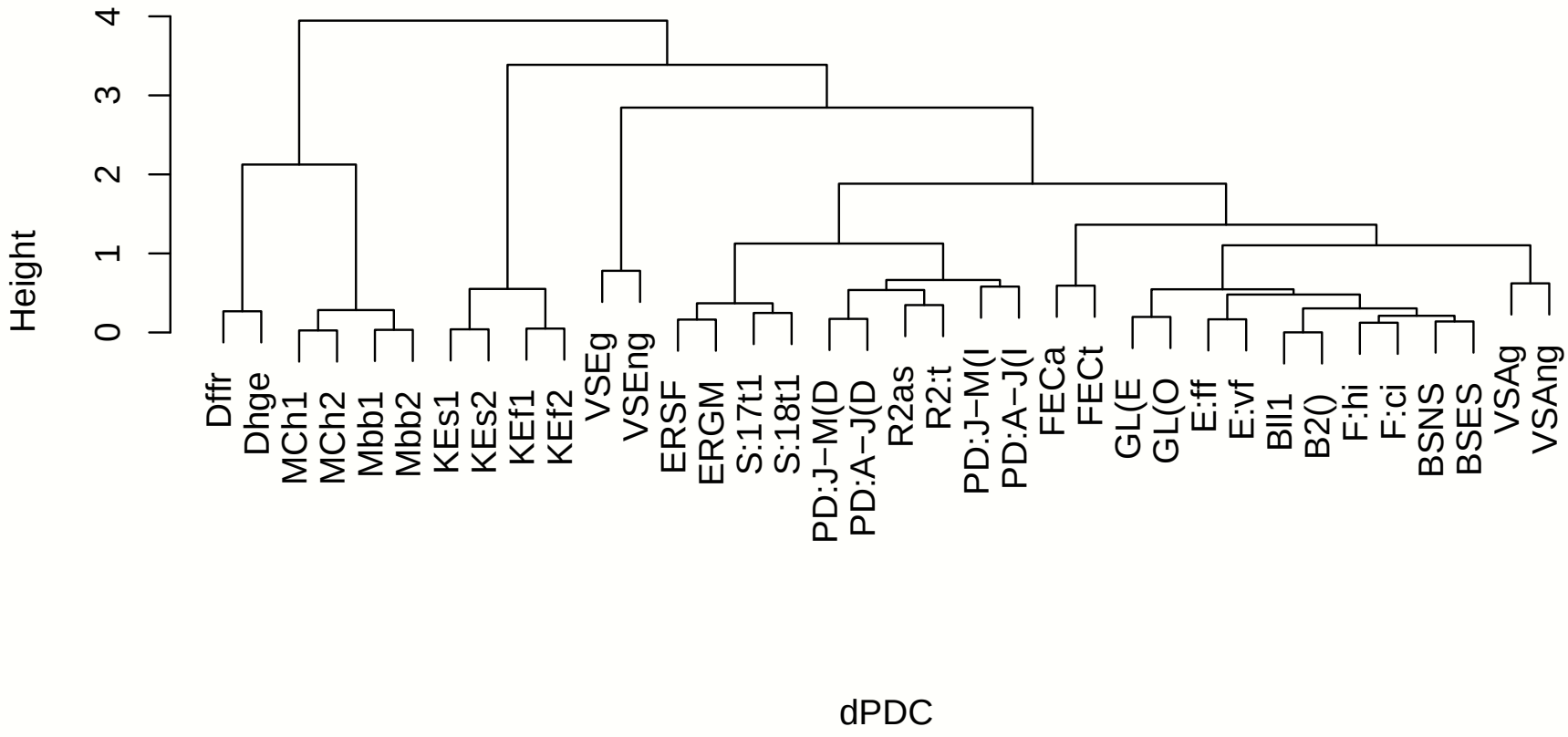
Ejemplo R

```
#cargar un dataset con series
#de diferentes dominios
#ECG, Sensores de movimiento,
#Demanda eléctrica...
data(”paired.tseries”)

#usamos la distancia PDC
#del paquete pdc
#incluida en TSclust por conveniencia
dPDC = diss(paired.tseries, ”PDC”, m=5,t=8)

plot(hclust(dPDC))
```

Series de diferentes dominios



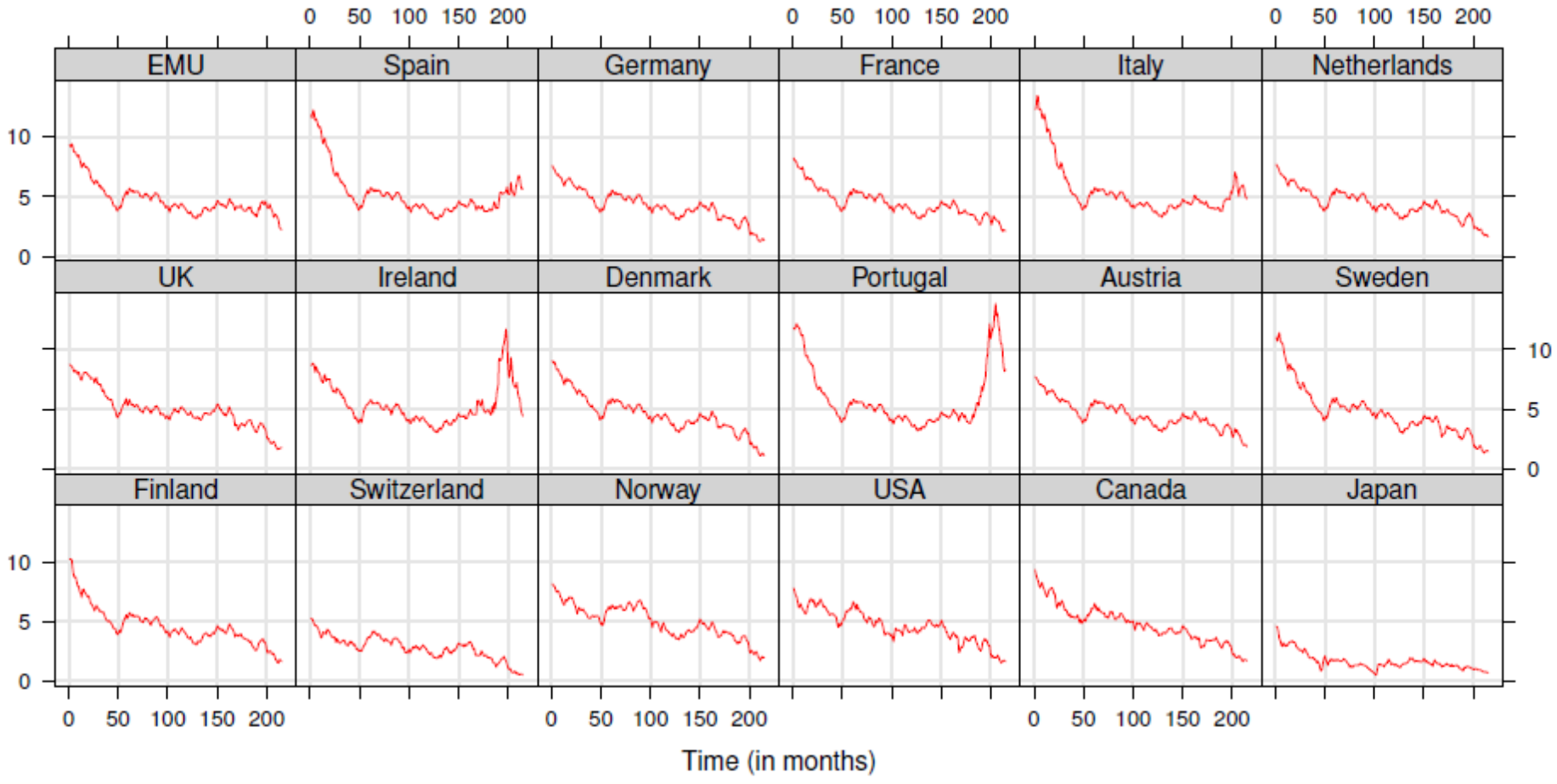
Resultado con distancia basada en complejidad



Distancias basadas en forecast

- Se basan en comparar el comportamiento de las series en un horizonte futuro.
- Ejemplo motivador: Protocolo de Kyoto
  - Queremos agrupar países por como será su comportamiento en 2020
  - No nos interesa tanto toda su trayectoria pasada
- Se ajusta un modelo predictivo probabilístico y se generan predicciones en el horizonte de interés
- Se comparan las densidades de las predicciones de cada serie

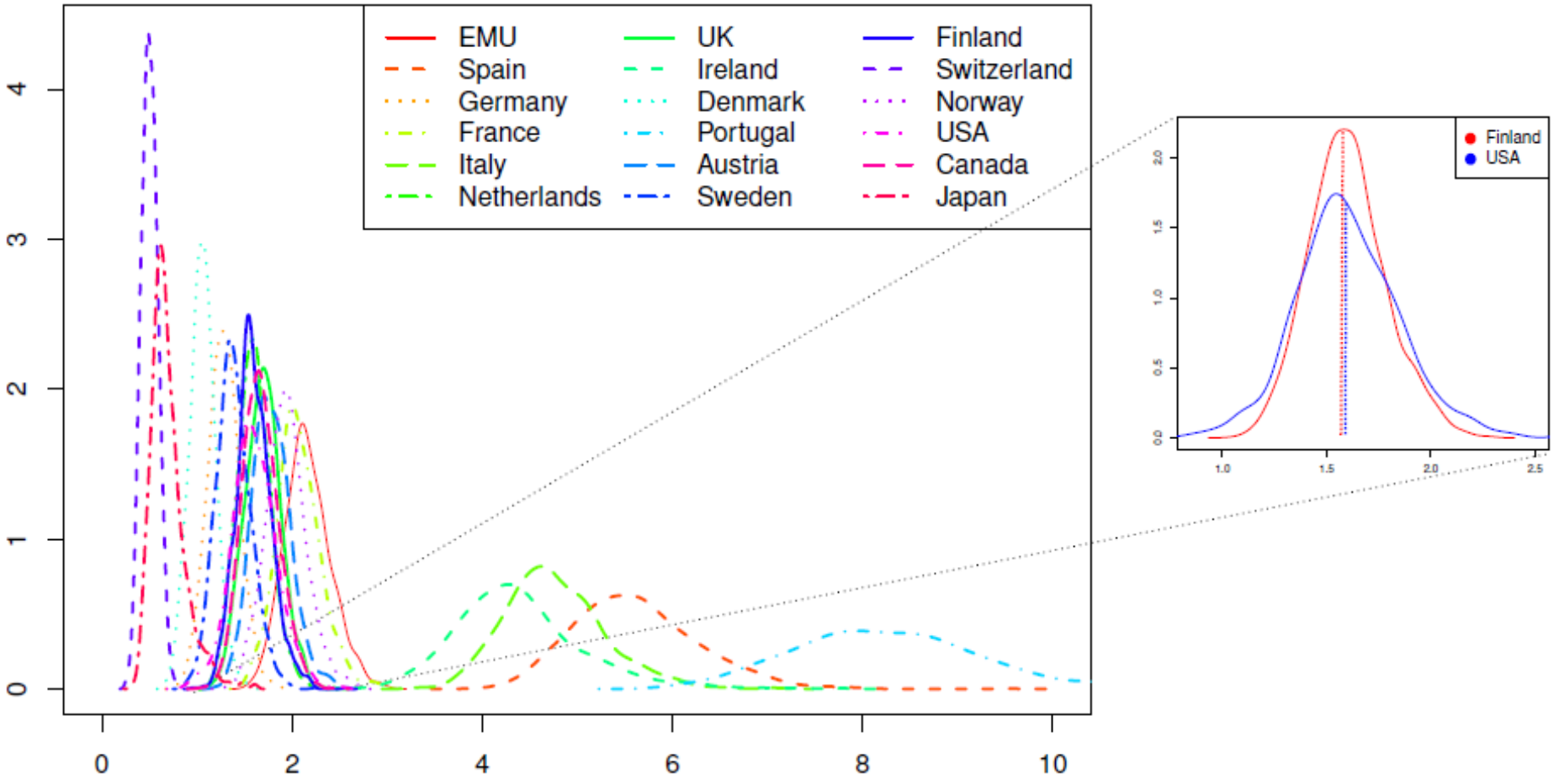
Series de tipos de interés



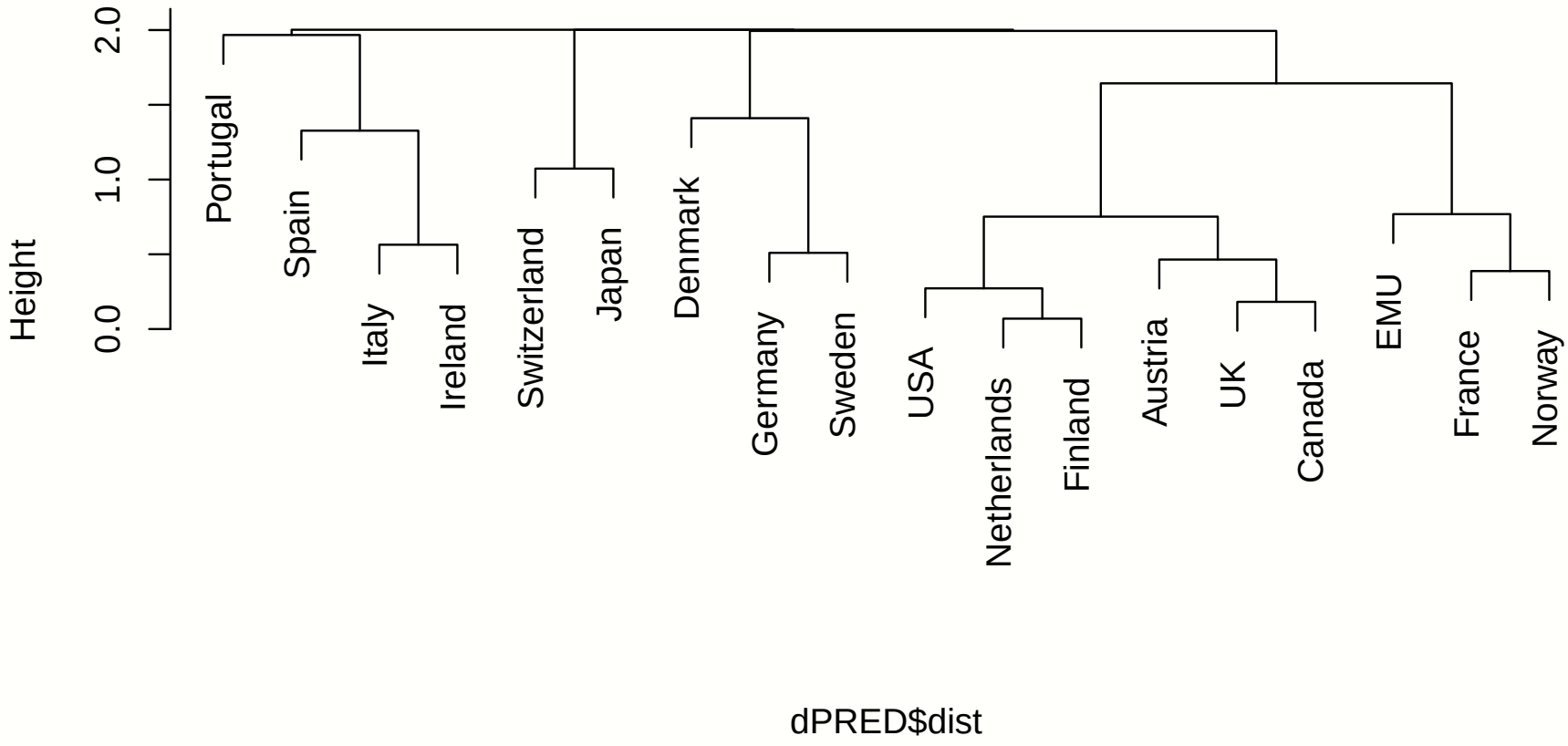
Ejemplo R

```
#aplicamos cluster basado en predicción
#el método de predicción es un
#autoregresivo no-paramétrico
#que acepta transformaciones clásicas
#diferenciación y logaritmos

#en este caso aplicamos las dos transformaciones
diffs <- rep(1, ncol(interest.rates))
logs <- rep(TRUE, ncol(interest.rates))

?diss.PRED
#comparamos las predicciones
# a un horizonte de 6 años
dPRED <- diss(interest.rates, "PRED", h=6, B=1200,
              differences=diffs, plot=TRUE)

plot(hclust(dPRED$dist))
```

Densidades de predicción



Resultado con distancia de predicción



Validación

- Con valor de verdad (externos): Medimos lo lejos que está una solución dada de la ideal
  - Índice de Rand Ajustado
- Sin valor de verdad (internos): Criterios como homogeneidad dentro de los cluster, separación entre clusters..
  - Coeficiente de silueta

Ejemplo R

```
#probamos con el datasaet inicial
#de series temporales simuladas

#creamos el valor de verdad "real"
tssynthetic_truth <- rep(1:6, each=3)

#aplicamos cluster usando PAM
pamCEPS <- pam(dCEPS, k=6)$clustering
pamEUCL <- pam(dEUCL, k=6)$clustering

#medida de evaluación implementada en TSclust
cluster.evaluation(tssynthetic_truth, pamCEPS)
[1] 0.875
cluster.evaluation(tssynthetic_truth, pamEUCL)
[1] 0.6944444
```

Ejemplo R

```
#el paquete fpc contiene medidas más populares
#de evaluación
#con valor de verdad como sin el
library(fpc)

cluster.stats(dCEPS, clustering=pamCEPS,
                alt.clustering=tssynthetic_truth )
$corrected.rand
[1] 0.7702703
$avg.silwidth
[1] 0.6442922
cluster.stats(dEUCL, clustering=pamEUCL,
                alt.clustering=tssynthetic_truth )
$corrected.rand
[1] 0.5142857
$avg.silwidth
[1] 0.1225102
```

Clasificación de Series Temporales

- El state of the art avanza muy rápido
- Se mide comparando los ratios de clasificación correcta en las bases de datos de timeseriesclassification.com
- Buenos resultados se pueden obtener usando distancias entre series temporales + k-NN
  - Especialmente usando ensembles de distancias

Ejemplo R

```
#obtenemos un dataset de ejemplo
download.file("http://timeseriesclassification.com/
               "ECG200.zip")
unzip("ECG200.zip")

trainset <- as.matrix(
read.arff("ECG200/ECG200_TRAIN.arff"))
class(trainset) <- "numeric"
trainclasses <- trainset[, ncol(trainset)]
trainset <- trainset[,-ncol(trainset)]

dDTW <- diss(trainset, "DTWARP")
#calculamos crossvalidation accuracy con TSclust
loo1nn.cv(dDTW, trainclasses)

dCORT <- diss(trainset, "CORT")
loo1nn.cv(dCORT, trainclasses)
```

Ejemplo R

```
#con TSclust y TSdist tenemos una gama de distancia
#métodos publicados
dEDR <- proxy::dist(trainset, EDRDistance, epsilon=
loo1nn.cv(dEDR, trainclasses)

#cargamos el conjunto de test
testset <- as.matrix(
read.arff("ECG200/ECG200_TEST.arff"))
class(testset) <- "numeric"
testclasses <- testset[, ncol(testset)]
testset <- testset[,-ncol(testset)]
#1-NN para las 3 distancias de ejemplo
predclassesDTW <- apply(testset, 1,
 function (xts) {
  distances <- apply(trainset, 1,
      function(yts) diss.DTWARP(xts, yts))
  trainclasses[which.min(distances)]})
```

Ejemplo R

```
#ensemble, majority voting ponderado por
# loo1nn crossvalidation
C <- unique(trainclasses)

C1 <- (predclassesEDR==C[1]) * loo1nn.cv(dEDR, trai
  (predclassesCORT==C[1]) * loo1nn.cv(dCORT, trainc
  (predclassesDTW==C[1]) * loo1nn.cv(dDTW, traincla

C2 <- (predclassesEDR==C[2]) * loo1nn.cv(dEDR, trai
  (predclassesCORT==C[2]) * loo1nn.cv(dCORT, trainc
  (predclassesDTW==C[2]) * loo1nn.cv(dDTW, traincla

mean(C[((C2 - C1) > 0)  + 1] == testclasses)
[1] 0.83
```

Contraste de Hipótesis de Homogeneidad

Motivación

Se pretende responder a la siguiente pregunta: ¿Tienen dos poblaciones de series temporales la misma distribución?

- Electrocardiogramas de pacientes y grupo de control
- Capturas de pesca de distintos caladeros
- Patrones de voz de dos zonas geográficas
- Patrones de lluvia en distintas zonas

Necesidad de aproximación específica a series temporales

Constrastes clásicos no son útiles por muchos motivos

- Complejidad inherente a las series temporales
- Distinta longitud de las series
- Se desean algunas invarianzas, por ejemplo: cambios de fase, aceleraciones/deceleraciones, volumen

Método

- Para atacar el problema utilizaremos también distancias entre series
- Combinada con un test de hipótesis que acepta distintos tipos de distnacias para comparar objetos
  - El estadístico energy

Ejemplo R

```
#usamos el energy statistic
library(energy)


dSPEC = diss(synthetic.tseries, ”PER”)
energy::eqdist.etest(dSPEC, c(9,9), R=10000)
 p-value = 0.0436

dEUCL = diss(synthetic.tseries, ”EUCL”)
energy::eqdist.etest(dEUCL, c(9,9), R=10000)
 p-value = 0.478
```

Conclusiones

- Presentados métodos basados en distancias
- Combinación de Distancia + Algoritmo
  - Cluster
  - Clasificación
  - Contraste de Hipótesis de Homogeneidad
- Paquetes
  - TSclust
  - TSdist
  - fpc, cluster