(Cluster Analysis)
&
(Classification And Regression Trees = CART)

James McCreight
mccreigh >at< gmail >dot< com

Why talk about them together?

- Partitioning data:
  - cluster analysis partitions vectors of data based on the properties of the vectors.
  - CART partitions a response (one entry in a vector) variable based on predictor variables (other entries in a vector)
- K-means clustering and CART select clusters which minimize variance.
- Continuous or categorical partitioning (regression vs classification).
- Hierarchical clustering and CART have the same partition structure

If we are going to talk about clustering it is worth the time to expose you to CART.

Cluster Analysis

Overview from wikipedia (font of all fact checks) reveals a broad topic with lots of applications.

Clusters and clusterings

The notion of a **cluster** varies between algorithms and is one of the many decisions to take when choosing the appropriate algorithm for a particular problem. At first the terminology of a cluster seems obvious: a group of data objects. However, the clusters found by different algorithms vary significantly in their properties, and understanding these **cluster models** is key to understanding the differences between the various algorithms. Typical cluster models include:

- Connectivity models: for example hierarchical clustering builds models based on distance connectivity.
- Centroid models: for example the k-means algorithm represents each cluster by a single mean vector.
- Distribution models: clusters are modeled using statistic distributions, such as multivariate normal distributions used by the Expectation-maximization algorithm.
- Density models: for example DBSCAN and OPTICS defines clusters as connected dense regions in the data space.
- Subspace models: in Biclustering (also known as Co-clustering or two-mode-clustering), clusters are modeled with both cluster members and relevant attributes.
- Group models: some algorithms (unfortunately) do not provide a refined model for their results and just provide the grouping information.

A **clustering** is essentially a set of such clusters, usually containing all objects in the data set. Additionally, it may specify the relationship of the clusters to each other, for example a hierarchy of clusters embedded in each other. Clusterings can be roughly distinguished in:

- **hard clustering**: each object belongs to a cluster or not
- **soft clustering** (also: **fuzzy clustering**): each object belongs to each cluster to a certain degree (e.g. a likelihood of belonging to the cluster)

There are also finer distinctions possible, for example:

- **strict partitioning clustering**: here each object belongs to exactly one cluster
- **strict partitioning clustering with outliers**: object can also belong to no cluster, and are considered outliers.
- **overlapping clustering** (also: **alternative clustering**, **multi-view clustering**): while usually a hard clustering, objects may belong to more than one cluster.
- **hierarchical clustering**: objects that belong to a child cluster also belong to the parent cluster
- **subspace clustering**: while an overlapping clustering, within a uniquely defined subspace, clusters are not expected to overlap.

Some Nomenclature

- Clustering is unsupervised learning: dosent require predictor variables; there's no reward function, no training examples; it's not regression.
- Elements of Statistical Learning (5th ed.)
  - ch 14 on unsupervised learning
  - chapter 14.3 (p501-528) focuses on the two most popular kinds of clustering for a wide variety of applications:

K-Means	K-Medoids
• hard clustering • centroid model • quantitative variables	• hard clustering • medoid model (cluster member) • quantative + ordinal + categorical variables

Both require a distance/dissimilarity metric.

Outline

- 1-d non-example: the idea of variance and clusters
- 2-d example, dissimilarity/variance in 2-d
- Dissimilarity / variance in N-d
- The algorithm
- Problem of a priori selection of K
    - hierarchical clustering

1-D Clusters and Variance

The 1-D squared euclidean distance/dissimilarity

$$d(x_i, \overline{x}_i) = (x_i - \overline{x}_i)^2$$

between any data point and its associated centroid x_i.

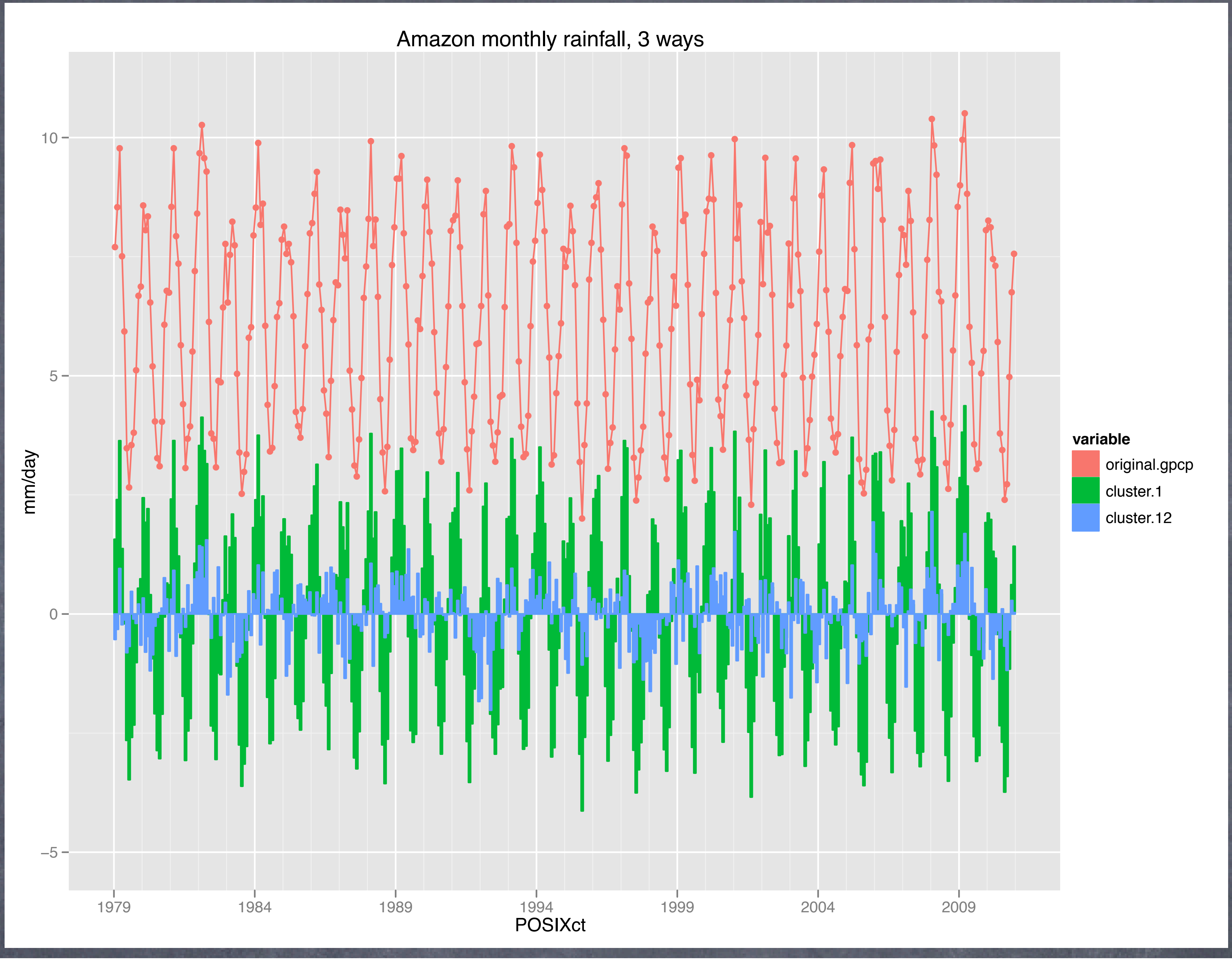
For a single 1-D cluster with centroid μ, k-means clustering minimizes the within-cluster scatter which looks like the (unnormalized) variance

$$W(C) = \sum_i d(x_i, \mu) = \sum_i (x_i - \mu)^2$$

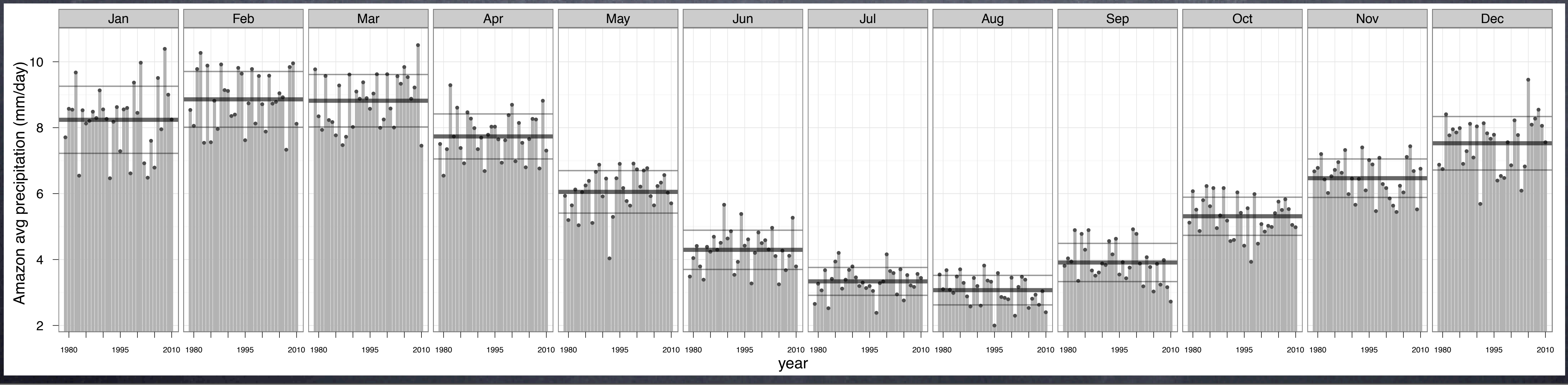
For K clusters (K centroids), we have:

$$W(C) = \sum_{i_1} (x_{i_1} - u_1)^2 + ... + \sum_{i_K} (x_{i_K} - u_K)^2$$

$$= \sum_{k=1}^{K} N_k \sum_{C(i)=k} (x_i - u_k)^2 \qquad (N_K = \sum_{i=1}^{N} I(C(i) = k))$$



Amazon monthly rainfall, 3 ways
mm/day
10
5
0
−5
1979
1984
1989
1994
1999
2004
2009
POSIXct
variable
original.gpcp
cluster.1
cluster.12



Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Amazon avg precipitation (mm/day)
10
8
6
4
2
1980
1995
2010
year

non-example:

- example was a priori clustering.
- "cluster analysis" is machine learning driven by an algorithm.
- for a specified number of clusters, machine learning would have found different centroids.
- the algorithm minimizes the scatter about the centroids.

illustrates:

- The total scatter, T, is a constant function of the data points, under euclidean norm it is proportional to their total variance
- T is the sum of the within-cluster scatter and between cluster scatter

$$T = W(C) + B(C)$$

- To minimize W is to maximize B.
- W and B are functions of the specific cluster centers, C(K), and their number, K.

Clustering in 2-d

The 2-d euclidean measure has x_i as 2-d vector, and the within-cluster scatter is minimized:

$$W(C) = \sum_{k=1}^{K} N_k \sum_{C(i)=k} (x_{i1} - u_{i1})^2 + (x_{i2} - u_{i2})^2$$

$$= \sum_{k=1}^{K} N_k \sum_{C(i)=k} \sum_{d=1}^{2} (x_{id} - \mu_{kd})^2$$

$$= \sum_{k=1}^{K} N_k \sum_{C(i)=k} ||x_i - \mu_k||^2$$

... example in R.

Clustering in D-d

Let x_i be a D-dimensional vector:

$$W(C) = \sum_{k=1}^{K} N_k \sum_{C(i)=k} (x_{i1} - u_{i1})^2 + ... + (x_{iD} - u_{iD})^2$$

$$= \sum_{k=1}^{K} N_k \sum_{C(i)=k} \sum_{d=1}^{D} (x_{id} - \mu_{kd})^2$$

$$= \sum_{k=1}^{K} N_k \sum_{C(i)=k} ||x_i - \mu_k||^2$$

Examples:

- 1-d: O rainfall observations
- 2-d: P points in 2-d space
- 3-d: P points in 3-d space
- 11-d: mtcars 32 obs of 11 vars (rows=obs in dataframe)
- T-d: P points with length T timeseries (homework)

Lloyd's "hill-climbing" algorithm

K-means Clustering Algorithm:
0. Assign an initial set of cluster centers, $\{\mu_1, ..., \mu_k\}$.
1. Assign each observation to its closest centroid in $\{\mu_1, ..., \mu_k\}$.
2. Update the centroids based on the last assignment.
3. Iterate steps 1 and 2 until the assignments (1) do not change.

- the algorithm is expensive (NP-hard: $O(n^{dk+1} \log n)$)
- this is a stochastic algorithm because of the 1st step,
  - results may vary from run to run!
- convergence depends on the assumptions of the model and the nature of the data:
  - model: spherical clusters which are separable so that their centroids converge.
  - data: try clustering a smooth gradient.

... on and on ...

- note: gaussian mixtures as soft k-means clustering (Hastie et al. p. 510),
    - mclust package: model based clustering, BIC...
- recent link of k-means and PCA under certain assumptions. see:http://en.wikipedia.org/wiki/K-means_clustering
- clustering built in to R (stats): kmeans, hclust
- clustering packages in R:
  clust, flexclust, mclust, pvclust, fpc, som, clusterfly
  see: http://cran.r-project.org/web/views/Multivariate.html
- QuickR page on clustering has some useful overview:
  http://www.statmethods.net/advstats/cluster.html

The problem of K

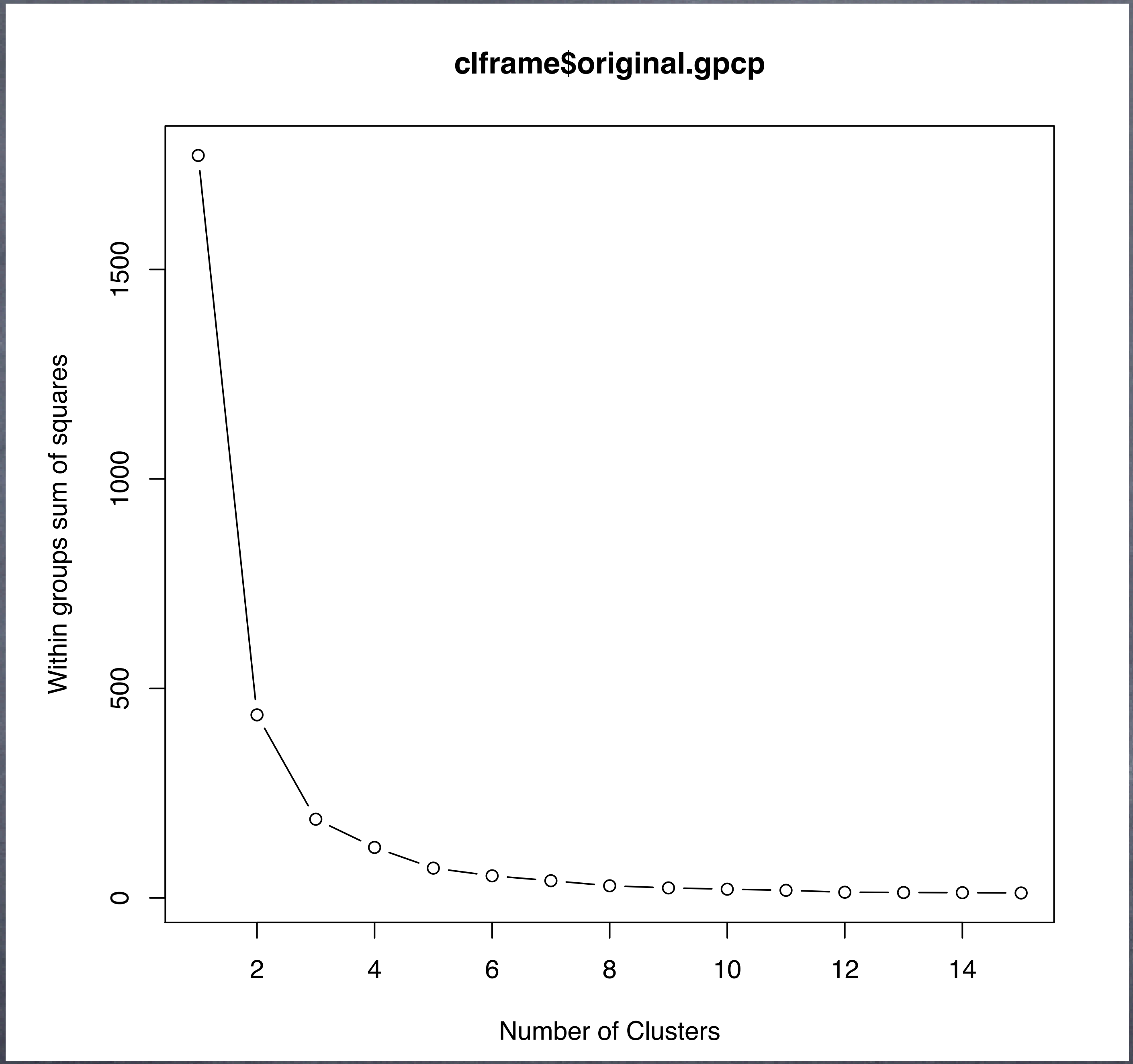
- in some situations, k is known. Fine.
- when k is not known we have a new problem, some approaches:
  - graph kink
  - model clustering EM/BIC approach
  - hierarchical approach

Amazon Rainfall redux

- A priori, we had a reason for 12 clusters: months of the year
- Consider we dont know anything about the physical problem, then consider
  - W(K)

```
## Determine number of clusters, adapted
kink.wss <- function(data, maxclusts=15) {
t <- kmeans(data,1)$totss
w <- laply( as.list(2:maxclusts),  function(nc) kmeans(data,nc)$tot.withinss )
plot(1:maxclusts, c(t,w), type="b",
    xlab="Number of Clusters", ylab="Within groups sum of squares",
    main=paste(deparse(substitute(data))) )
}
```

Amazon Rainfall redux continued

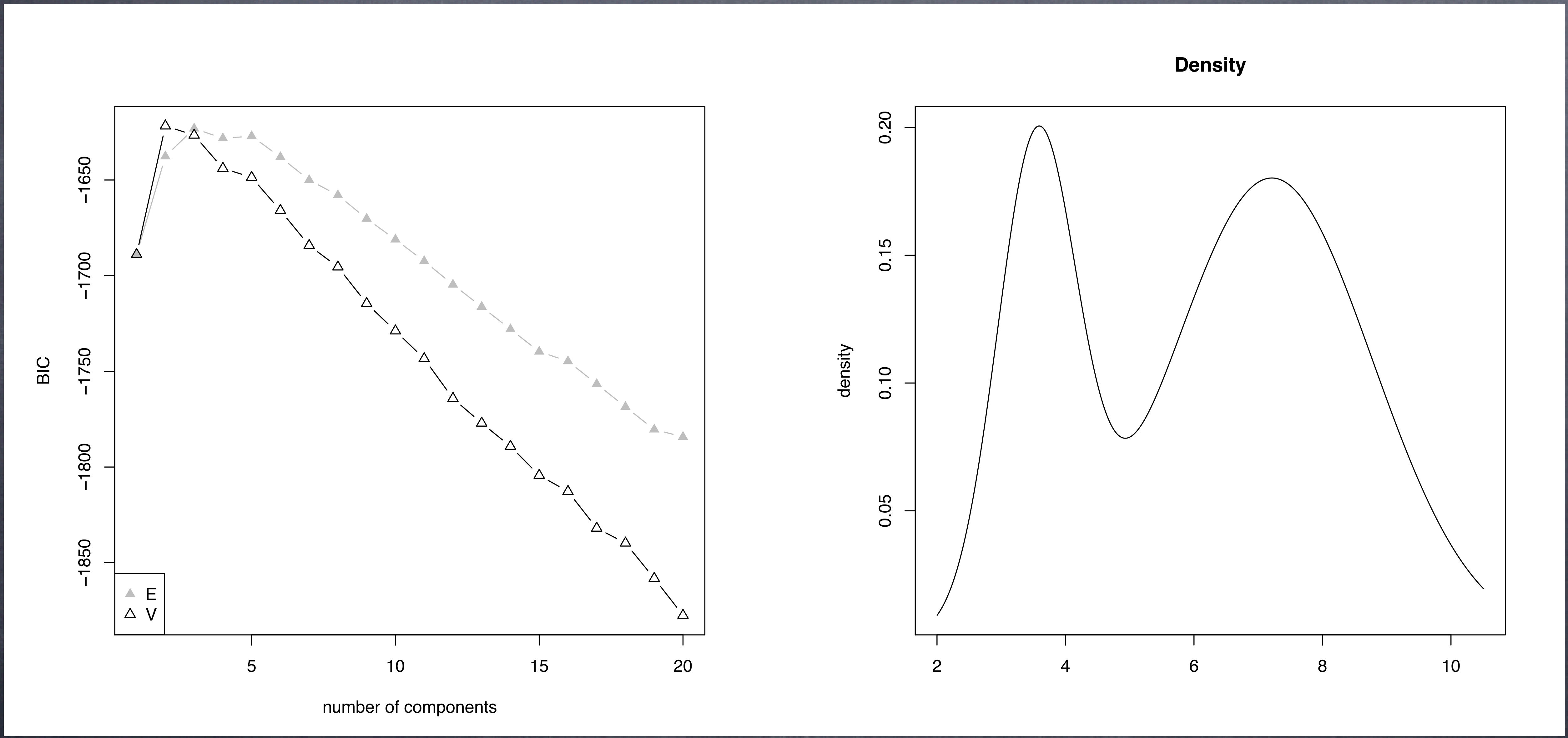


- looking for a number of clusters after which W dosent decrease much.

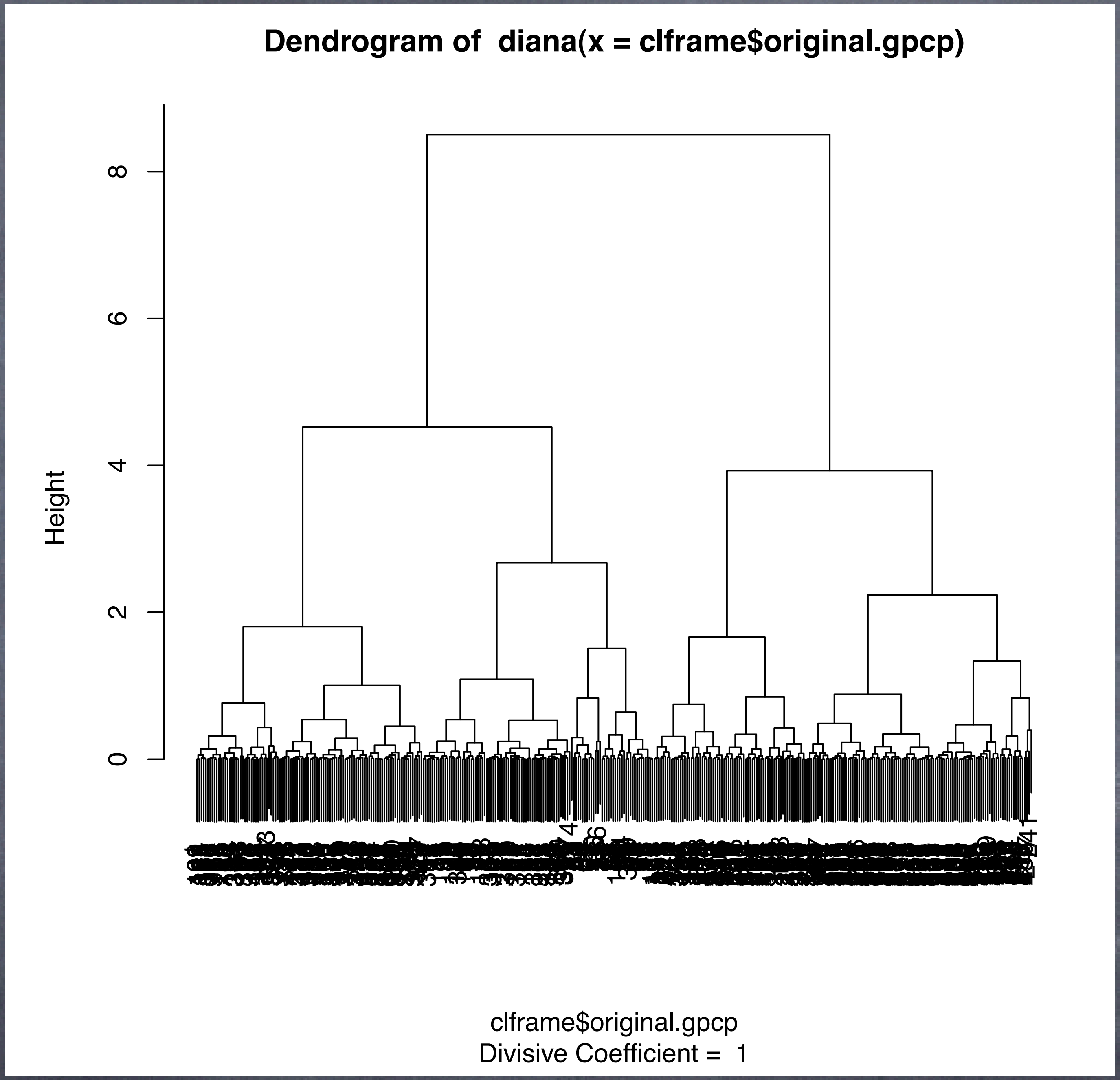
aside...
EOF/PCA vs Cluster Analysis

- Dominant variability (modes) vs similar observations (clusters),
  - one <u>chooses</u> the # of clusters but not the # of modes.
- EOF/PCA: data subspaces which explain maximum variance.
- Cluster analysis: similarities/differences in observations
  - identify observations which vary similarly,
  - decompose non-stationarity, homogenize a variable.

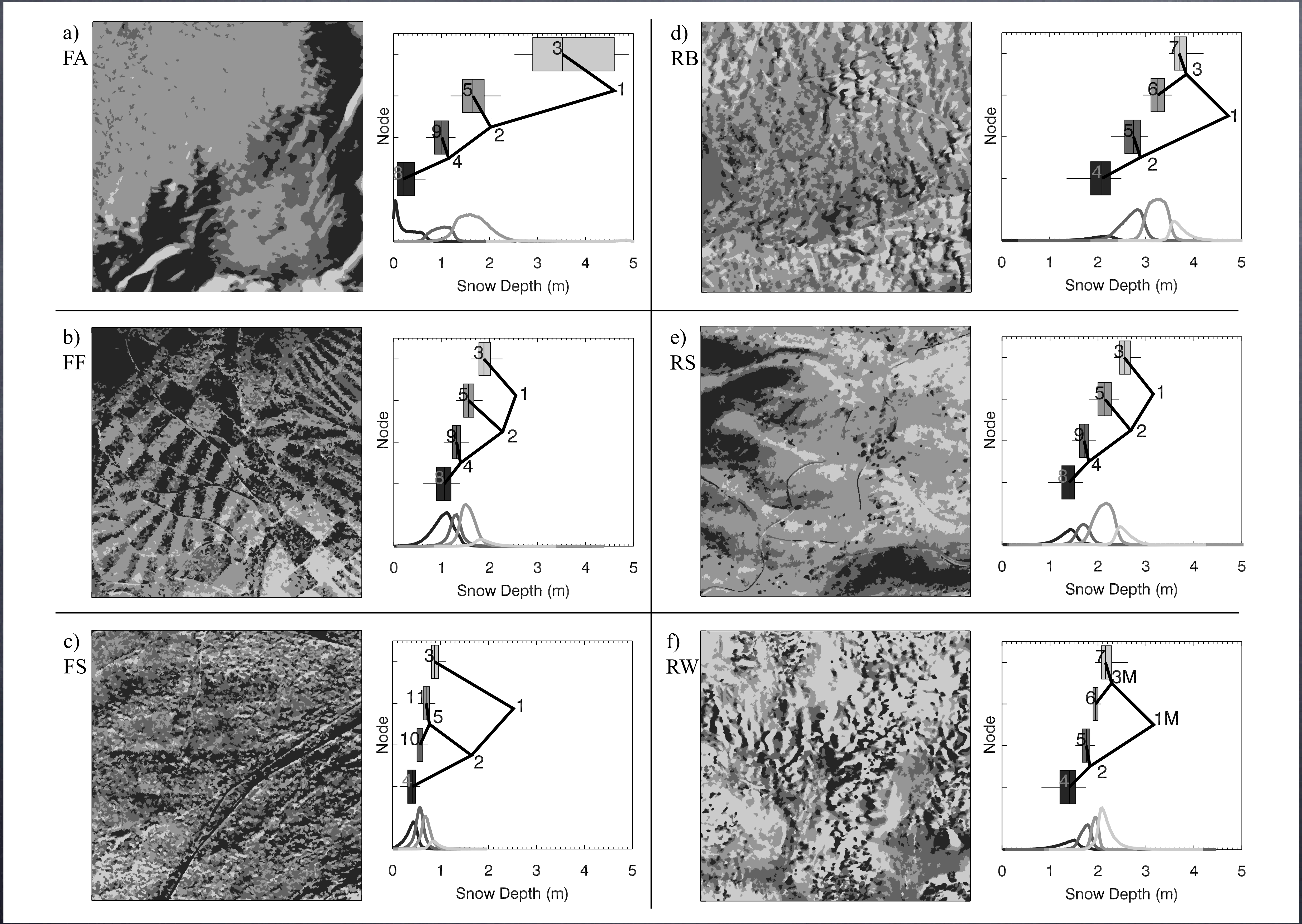
mclust: 2 cluster mixture model via EM



cluster: diana



Resolving spatial non-stationarity in snow depth distribution



New Observations

- Classification: assign a new observation to its closest centroid of an existing clustering.

But what does that get you??

- Typically we want an estimate or prediction of some variable from new data, not just a classification.

-> CART

```r
require(ggplot2)

## generate 3 random clusters about fixed centroids (5,5), (5,-5) and (-5,-5)
clust.2d <- function(var=0) {
  data <- as.data.frame(rbind( cbind(x=rnorm(10, +5, var), y=rnorm(10, 5, var)),
                              cbind(rnorm(15, +5, var), rnorm(15,-5, var)),
                              cbind(rnorm(12, -5, var), rnorm(12,-5, var)) )  )
  plot.frame <- as.data.frame(data); plot.frame$orig.clust <- factor(c(rep(1,10),rep(2,15),rep(3,12)) )
  plot.frame$k.clust <- factor(kmeans( data, 3)$cluster) ## make it a factor, since it's categorical
  ggplot( plot.frame, aes(x=x,y=y,color=orig.clust,shape=k.clust) ) + geom_point(size=3)
}


clust.2d()
clust.2d(var=2)
clust.2d(var=3)
clust.2d(var=10)

# what is the total scatter?
var=1
data <- as.data.frame(rbind( cbind(x=rnorm(10, +5, var), y=rnorm(10, 5, var)),
                            cbind(rnorm(15, +5, var), rnorm(15,-5, var)),
                            cbind(rnorm(12, -5, var), rnorm(12,-5, var)) ) )

## calculate T = W + B
kdata <- kmeans(data,3)
str(data)
str(kdata)

T <- sum(diag(var(data))*(length(data[,1])-1)) ## unbiased sample variance is used in var()
T
T2 <- sum( (data$x-mean(data$x))^2 + (data$y-mean(data$y))^2 )
T2

W <- sum((data-kdata$centers[kdata$cluster,])^2)
kdata$tot.withinss


# Determine number of clusters, adapted
kink.wss <- function(data, maxclusts=15) {
  t <- kmeans(data,1)$totss
  w <- laply( as.list(2:maxclusts),  function(nc) kmeans(data,nc)$tot.withinss )
  plot(1:maxclusts, c(t,w), type="b",
     xlab="Number of Clusters", ylab="Within groups sum of squares",
     main=paste(deparse(substitute(data))) ) ## oooh, fancy!
}

kink.wss(data, max=8)
```