Consensus Clustering

• • •

An Ensemble Approach to a Practitioner's Dilemma

The Problem

For many real-world datasets, and for high-dimensional data (think text, image) in particular:

- **Different algorithms rarely agree** upon the cluster solution
- Most algorithms **require the user to input the number of clusters**
- **Distance metrics suffer** as the dimensionality of the data increases
    - Difficult to evaluate and compare cluster solutions
    - Algorithms become unpredictable, likely to get stuck at local optima

A Practitioner's Dilemma

A researcher pulls research abstracts from a web database
$\approx$ 4,000 documents containing $\approx$ 11,000 terms (variables).

The documents were pulled from 3 research domains
(forming 3 major themes/clusters in the data)

X=

Document	Count of Term 1	Count of Term 2	Count of Term 3	...
Document 1	1	2	0	...
Document 2	0	0	2	...
Document 3	3	1	0	...
Document 4	0	0	1	...
⋮	⋮	⋮	⋮	⋮

The goal: Partition the documents according to dominant themes.

A Practitioner's Dilemma

After a survey of literature, the researcher compiles a list of 7 algorithms which have been heavily cited for document clustering:

1. PDDP
2. Spherical k-means
    1. With random initialization
    2. Initialized with centroids from PDDP clustering
3. Nonnegative Matrix Factorization (NMF)
4. Power Iteration Clustering (PIC)
5. Spectral Clustering
    1. Normalized Cuts of Meila and Shi (NCut)
    2. Normalized Cuts of Ng-Jordan-Weiss (NJW)

A Practitioner's Dilemma

<u>**The Plan:**</u> Use all 7 algorithms and compare the results using **3 heavily cited metrics** for cluster evaluation to choose a final solution

1. **The Silhouette Coefficient** (SC)
    1. Range: $-1 \leq SC \leq 1$
    2. Values closer to +1 are desired
    3. Computationally intensive - involves many distance calcs for every point
2. **Ray & Turi's Validity Metric** (V)
    1. Range: $V>0$
    2. Smaller values desired
3. **Sum of Squared Error Criterion** (k-means objective function, aka **Inertia**)
    1. Range: $SSE>0$
    2. Smaller values are desired

A Practitioner's Dilemma

Let's **rank** the solutions according to these metrics:

	Silhouette	Ray&Turi	SumSqError
PDDP	3	3	3
PDDP-kmeans	7	1	1
Rand-kmeans	6	5	6
NMF	2	6	7
PIC	4	7	4
NCUT	1	4	5
NJW	5	2	2

A Practitioner's Dilemma

And now compare how those cluster validity measures mapped to the accuracy of the clustering:

	Silhouette	Ray&Turi	SumSqError	Accuracy
PDDP	3	3	3	83.0
PDDP-kmeans	7	1	1	69.8
Rand-kmeans	2	6	7	50.9
NMF	6	5	6	70.7
PIC	4	7	4	88.9
NCUT	1	4	5	96.6
NJW	5	2	2	85.0

A Practitioner's Dilemma

And now compare how those cluster validity measures mapped to the accuracy of the clustering:

	Silhouette	Ray&Turi	SumSqError	Accuracy
PDDP	3	3	3	83.0
PDDP-kmeans	7	1	1	69.8
Rand-kmeans	2	6	7	50.9
NMF	6	5	6	70.7
PIC	4	7	4	88.9
NCUT	1	4	5	96.6
NJW	5	2	2	85.0

A Practitioner's Dilemma

And now compare how those cluster validity measures mapped to the accuracy of the clustering:

	Silhouette	Ray&Turi	SumSqError	Accuracy
PDDP	3	3	3	83.0
PDDP-kmeans	7	1	1	69.8
Rand-kmeans	2	6	7	50.9
NMF	6	5	6	70.7
PIC	4	7	4	88.9
NCUT	1	4	5	96.6
NJW	5	2	2	85.0

Dimension Reduction

- Shouldn't the researcher reduce the dimensions first? **YES.**
- Almost as many options for dim. reductions as there are for clustering!
- How can we compare clusterings for two different dimension reductions? The underlying data is different!
- Have to compare using metrics on full data. Metrics suffer due to data dimensionality.

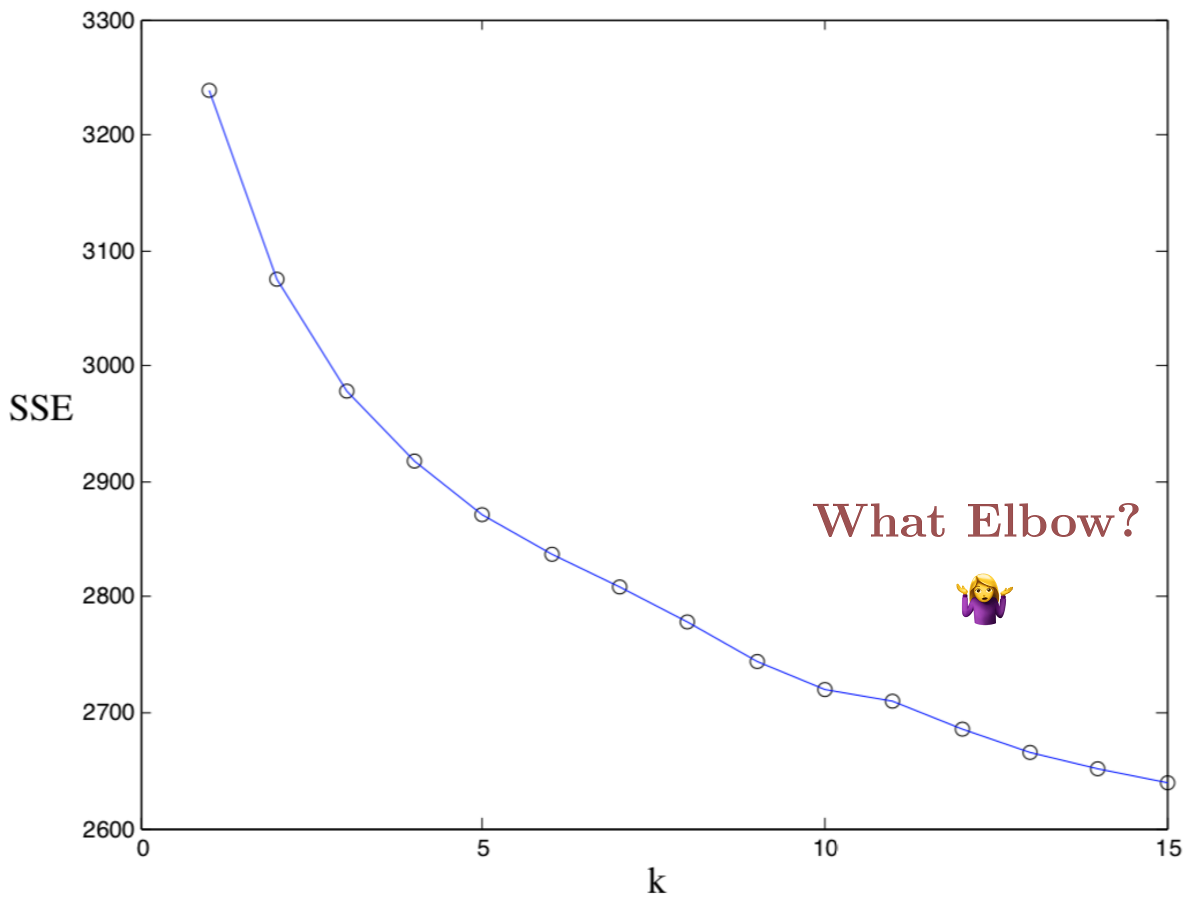
Choosing k

- Backing up - how did the research determine how many clusters to create?
- Let's approach this problem using some recommended tools from the literature:
  1. Sum Squared Error (SSE aka Inertia) Plots
  2. Ray and Turi's Plots
  3. Statistical Hypothesis Testing (generally bad for big data)
     1. SPSS
     2. SAS's Cubic Clustering Criterion
     3. The Gap Statistic

Choosing *k*: Sum Squared Error (SSE) Plots

Visual is dependent on choice of clustering algorithm

Plot the SSE for k=1,2,3,... and look for an "elbow" in the graph



Choosing *k*: Ray and Turi's Plot

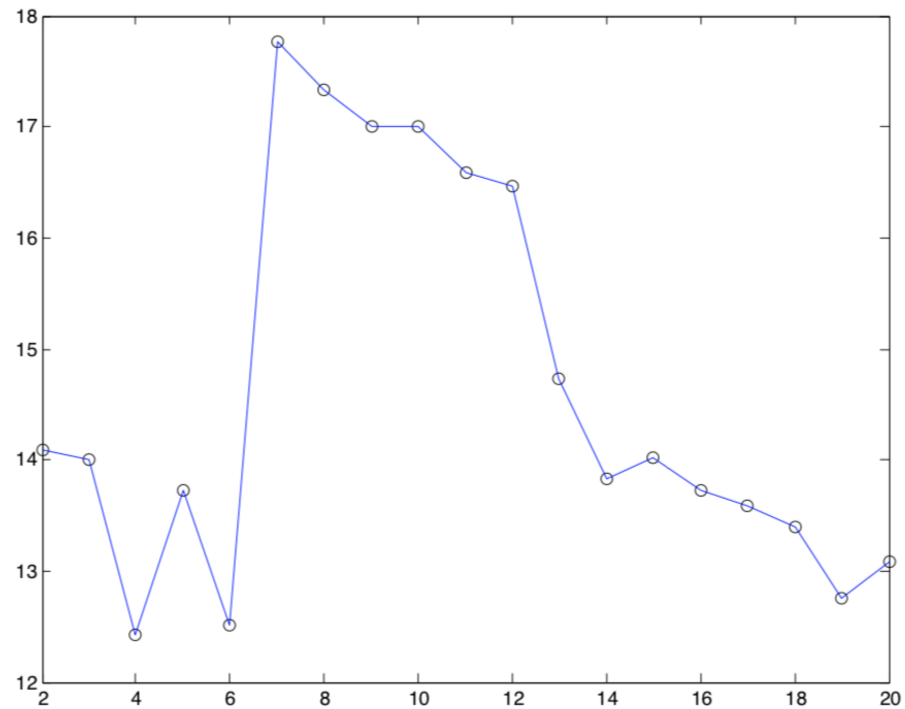
Visual is dependent on choice of clustering algorithm

Plot Ray and Turi's statistic for k=1,2,3, ... and identify *either:*

1. The **global minimum**
2. The **modified minimum**: The local minimum following the first local maximum

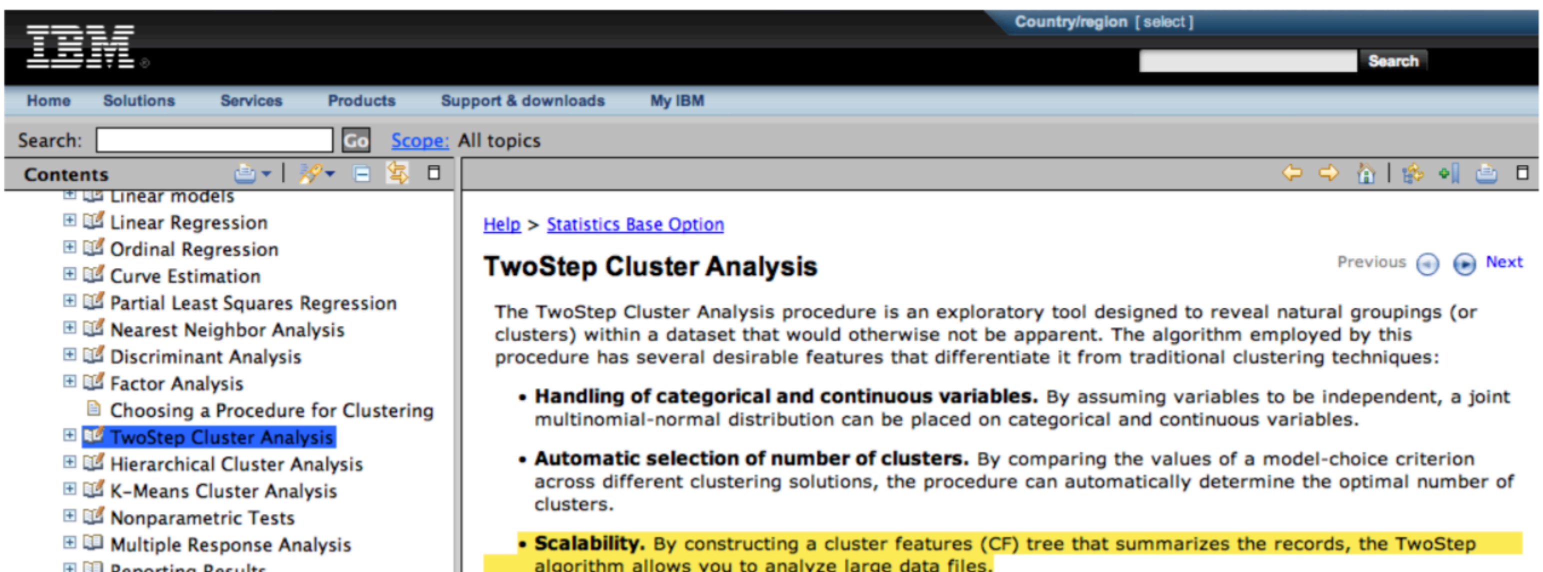
Global Min: 4

Modified Min: 6



Choosing *k*: Statistical Hypothesis Testing

- The Gap Statistic
  - Too inefficient for large datasets
- SAS's Cubic Clustering Criterion
  - Chosen number of clusters: 50
- SPSS 2-Step cluster procedure



Cool, let's try that!

Choosing *k*: Statistical Hypothesis Testing

(4 days later) The response: "Go home, you have no clusters"

Model Summary

Algorithm	TwoStep
Inputs	11001
Clusters	1

Cluster Quality

Cluster quality cannot be computed for a single-cluster solution.

Practitioners need a more practical way to explore clusters in their data.

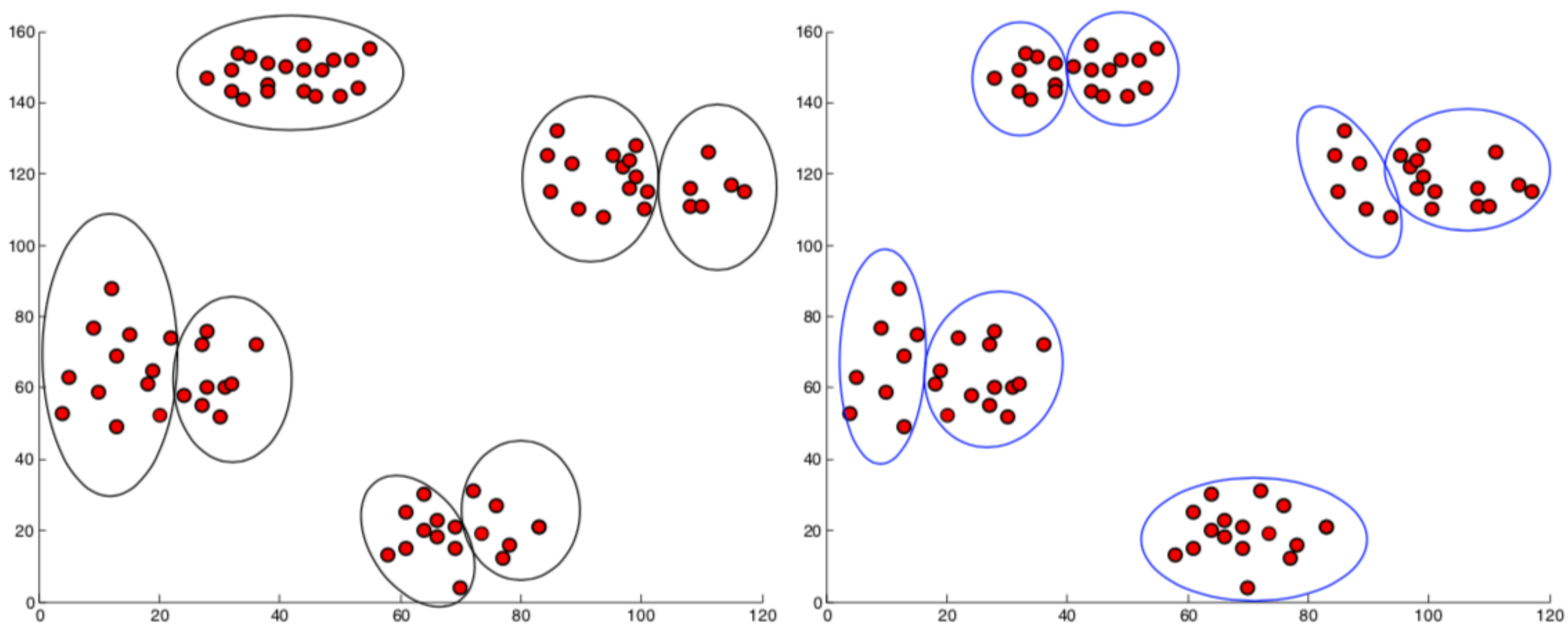
Consensus Clustering

• • •

How can we combine the input from multiple clusterings into one final solution?

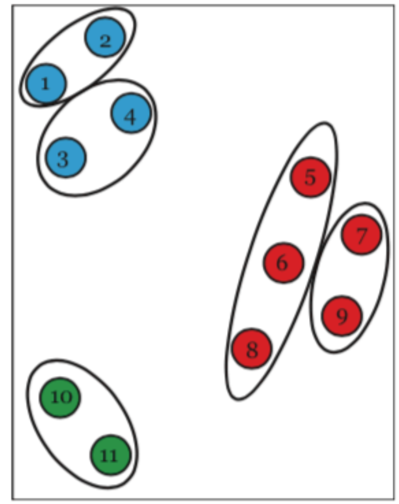
Assumptions of Consensus Clustering

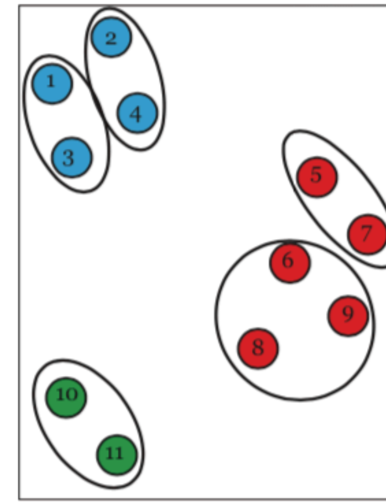
- If there are **truly k clusters** in a given dataset and a clustering **algorithm is set to find $\hat{k} > k$ clusters** then the original k **clusters will be broken apart into smaller clusters** to form $\hat{k}$ total clusters.
- In the absence of sub cluster structure, **different algorithms will do this in different ways.**



The Consensus Matrix, $\mathbf{C}$

First create many clusterings of your data



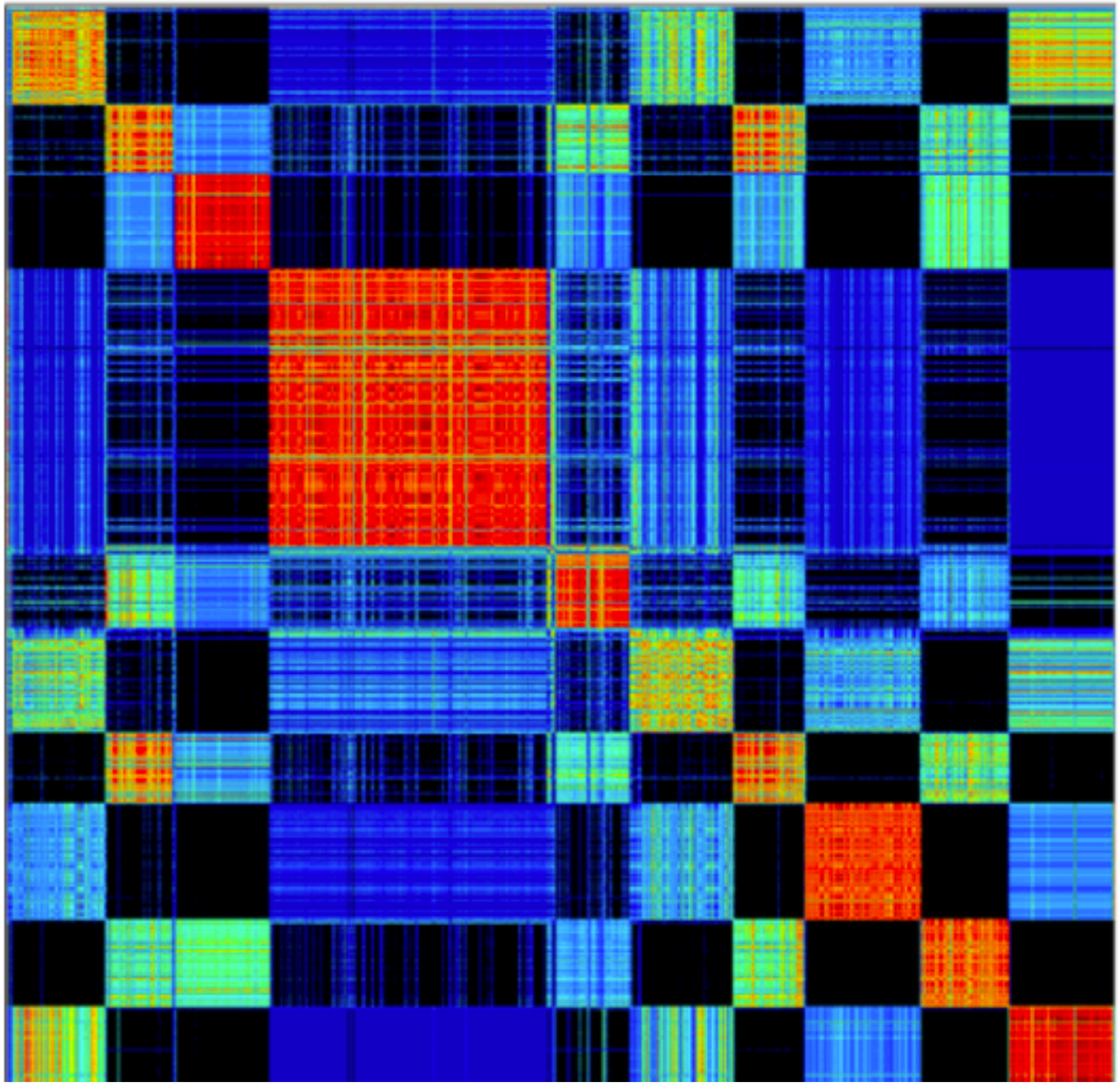


Consensus Clustering

- You can make it as simple as using k-means with many random initializations.
- You need not limit yourself to one value of k.
- You need not limit yourself to one algorithm.
- You need not limit yourself to one dimension reduction.

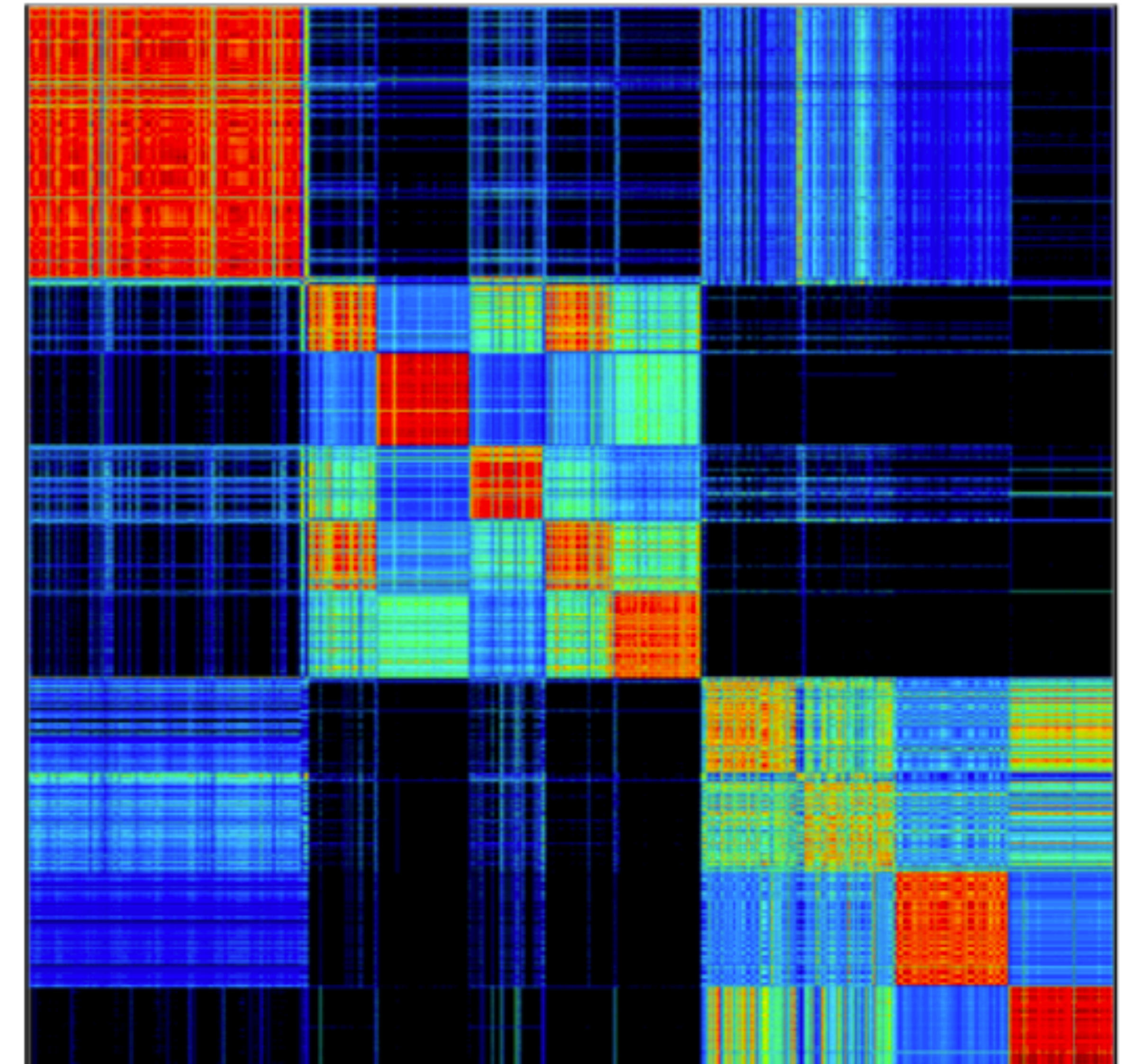
The Consensus Matrix

- Consensus Matrix of the text data
- Each Row/Column is one document
- Each pixel is a value in the matrix, blue<red
- Matrix ordered by a solution with k=10 clusters



The Consensus Matrix

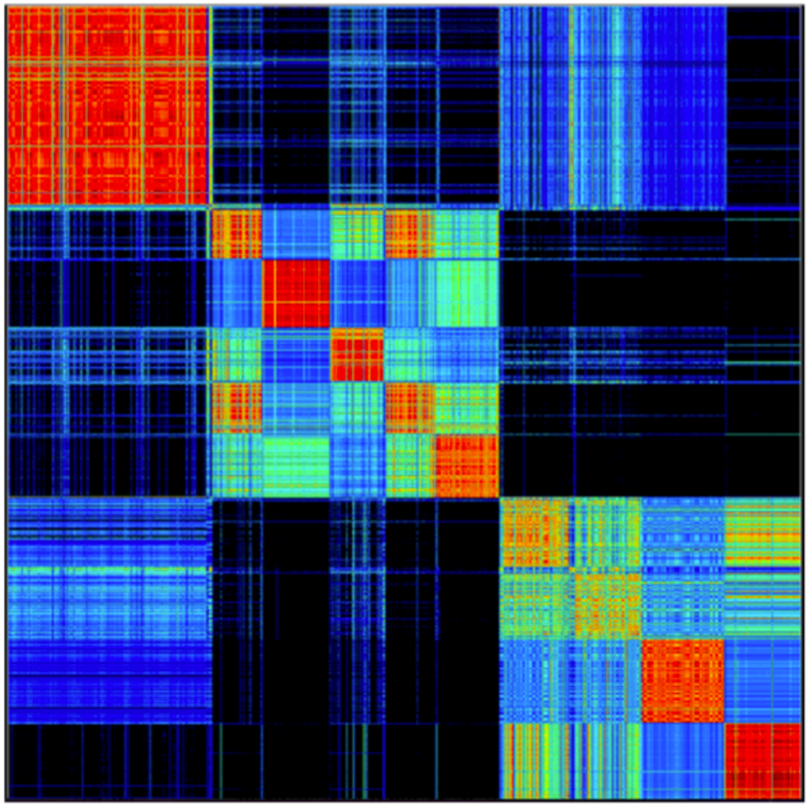
Same exact matrix, reordered according to a k=3 cluster solution, keeping the k=10 solution within that larger solution

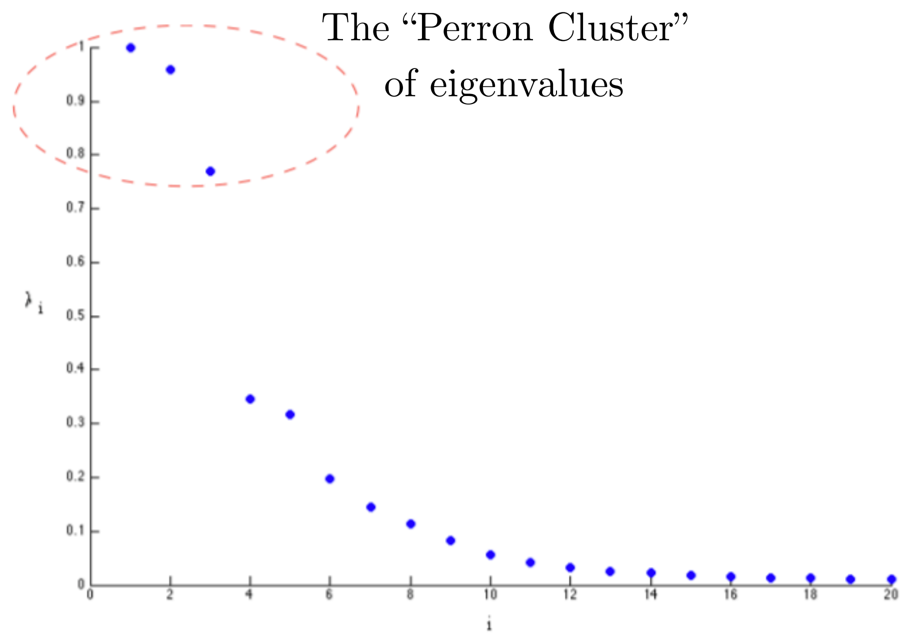


Determining number of clusters

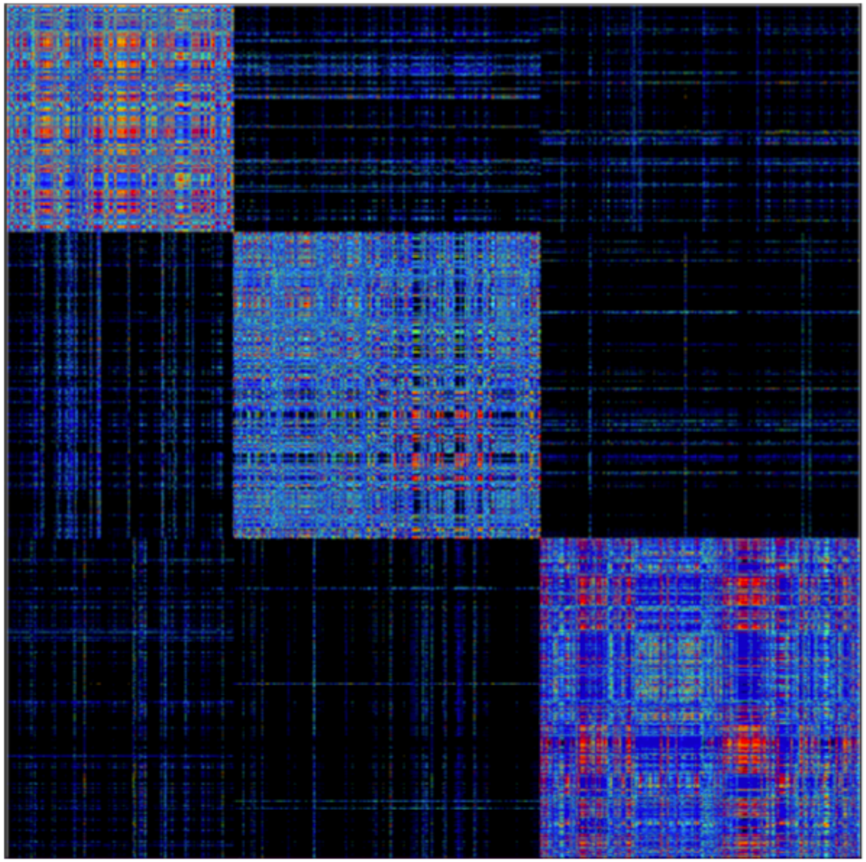
- Visuals are pretty but not practical solution to counting clusters.
- Instead, create the matrix $\mathbf{P} = \mathbf{D}^{-1}\mathbf{C}$ where $\mathbf{D}$ is a diagonal matrix containing the row sums of $\mathbf{C}$.
  Now the row sums are 1.
- Observe the eigenvalues of $\mathbf{P}$, look for a group of them near 1 followed by a gap.

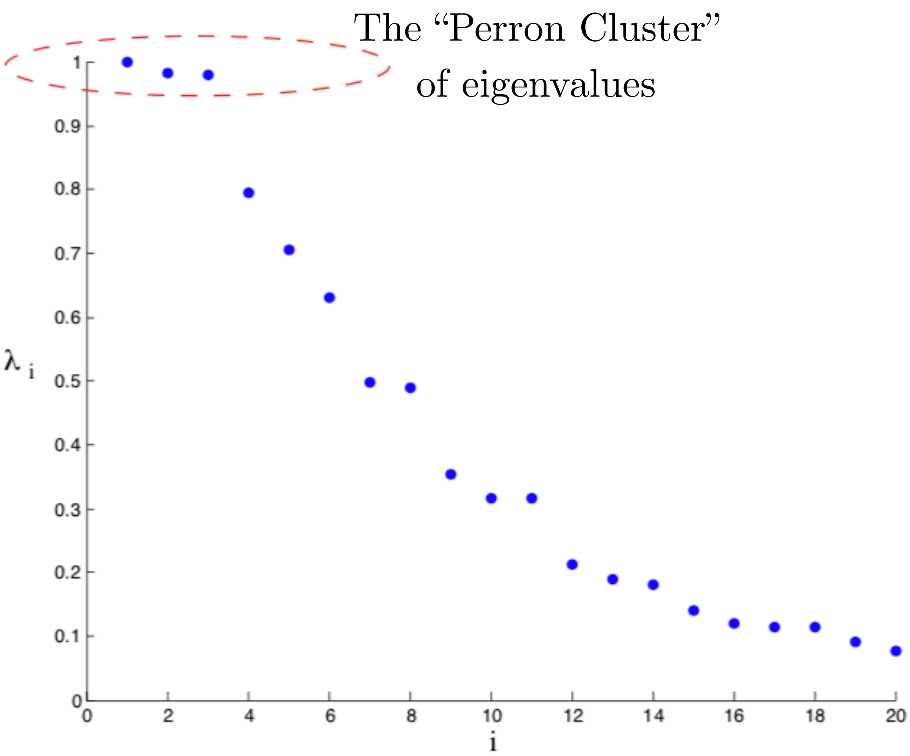
Answer: k=3 ✅




The "Perron Cluster"
of eigenvalues
λ_i
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
2
4
6
8
10
12
14
16
18
20
i

Same Data, New Consensus Matrix (using k=10, 11, ..., 20 clusters)





Final Clustering?

The consensus matrix clarifies the cluster solution, making it easier for algorithms to find.

Algorithm	Accuracy
PDDP	88%
PDDP-kmeans	97%
Rand-kmeans	97%
NMF	97%
PIC	73%
NCUT	97%
NJW	96%

Final Clustering?

The consensus matrix clarifies the cluster solution, making it easier for algorithms to find.

Algorithm	Accuracy	Original Accuracy
PDDP	88%	83.0
PDDP-kmeans	97%	69.8
Rand-kmeans	97%	50.9
NMF	97%	70.7
PIC	73%	88.9
NCUT	97%	96.6
NJW	96%	85.0

Final Clustering?

Usually, any clustering algorithm performed on the **consensus matrix** will have **better stability and performance than** the same algorithm on **the raw data.**

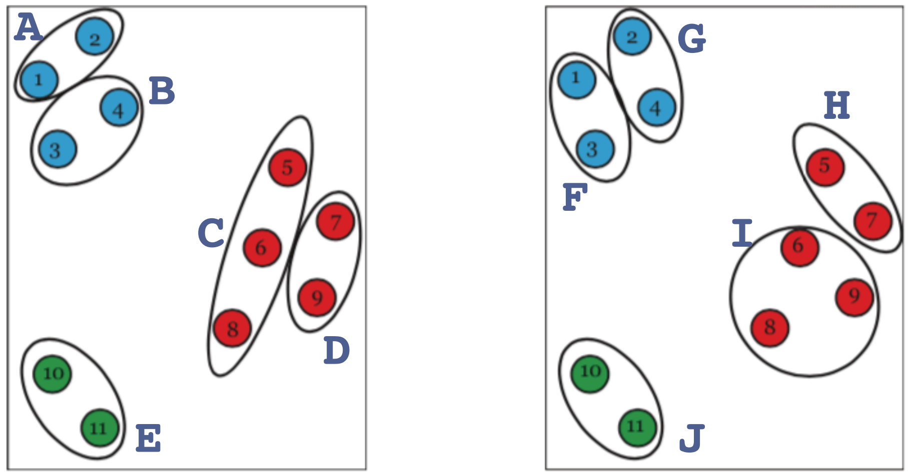
One **can *iterate* this process** until algorithmic consensus by clustering the consensus matrix many times and **forming a *new* consensus matrix**. Repeat **until many clustering algorithms agree** upon a common solution.

In particularly tricky problems, a **drop tolerance** parameter, ϱ can be introduced, where entries in the consensus matrix less than ϱ are set to 0. (i.e. two observations must be clustered together at least ϱ times to be considered related in consensus matrix)

Can't afford the Consensus Matrix? Try the "pre-consensus" matrix

- On large datasets, the consensus matrix requires a lot of storage >15Gb for 45K observations without sparse matrix magic.
- Try the **"pre-consensus" matrix**, **H**, since $\mathbf{H}\mathbf{H}^T = \mathbf{C}$
- **H** is a binary matrix with rows corresponding to observations and columns corresponding to clusters, having one column for every cluster created (across many clusterings).
- (i,j) entry of **H** is 1 if observation i was placed in cluster j

Can't afford the Consensus Matrix? Try the "pre-consensus" matrix



$$
\mathbf{H}= \begin{array}{c} \\ 1\\2\\3\\4\\5\\6\\7\\8\\9\\10\\11 \end{array}
\begin{array}{c}
\begin{array}{cccccccccc} \mathbf{A} & \mathbf{B} & \mathbf{C} & \mathbf{D} & \mathbf{E} & \mathbf{F} & \mathbf{G} & \mathbf{H} & \mathbf{I} & \mathbf{J} \end{array} \\
\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}
\end{array}
$$

May not get a nice clear "Perron Cluster," but **singular values** of this matrix will inform the choice of k.

(**They are the eigenvalues of C**)

This matrix is also easy to update with additional clusterings!

Just Remember

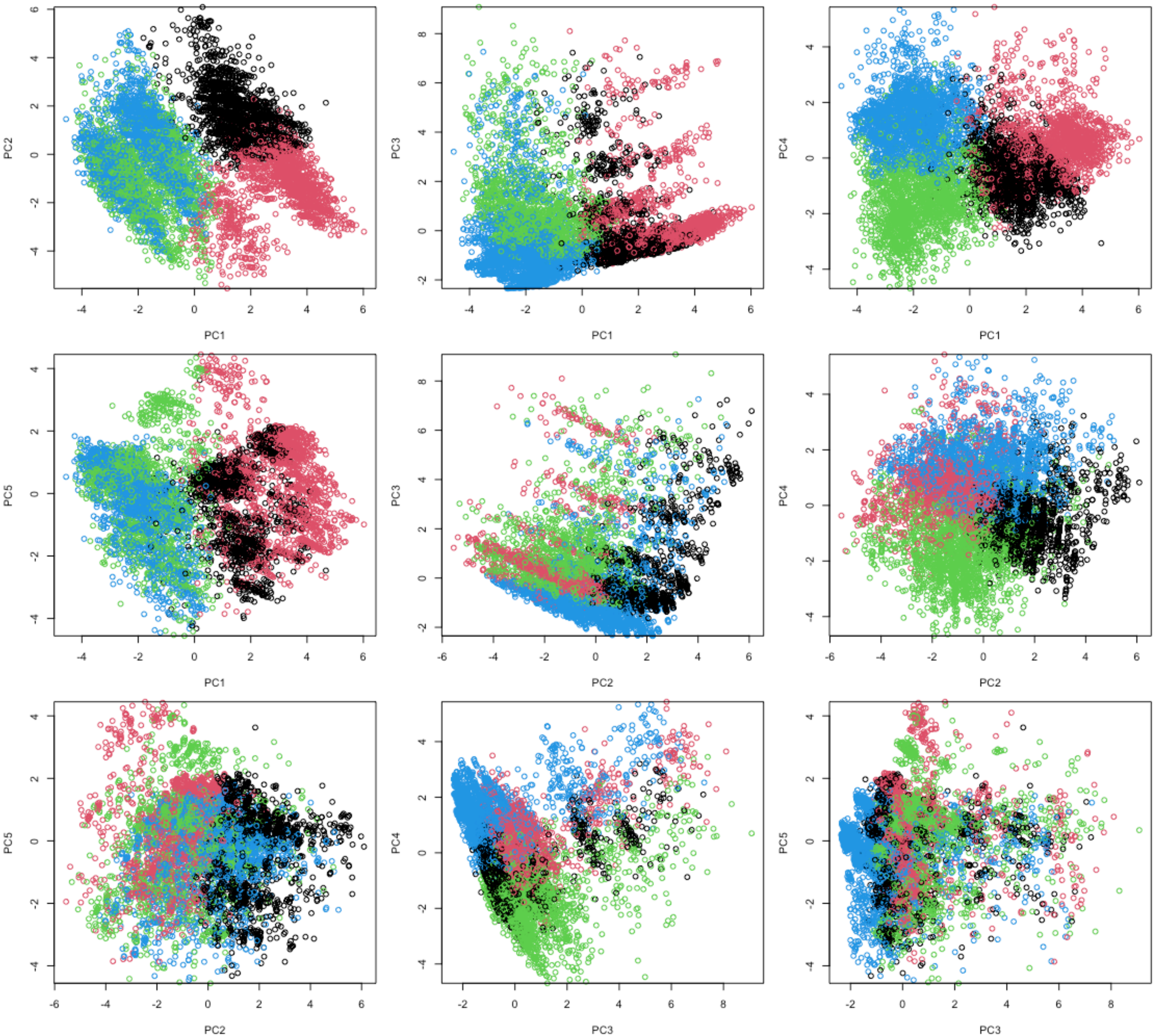
- There are so many options
- You can always do a clustering with k=2 or 3 and work with those larger clusters individually (i.e. cluster the larger clusters of data into smaller clusters) if you see something you'd like to explore in the PC visualizations

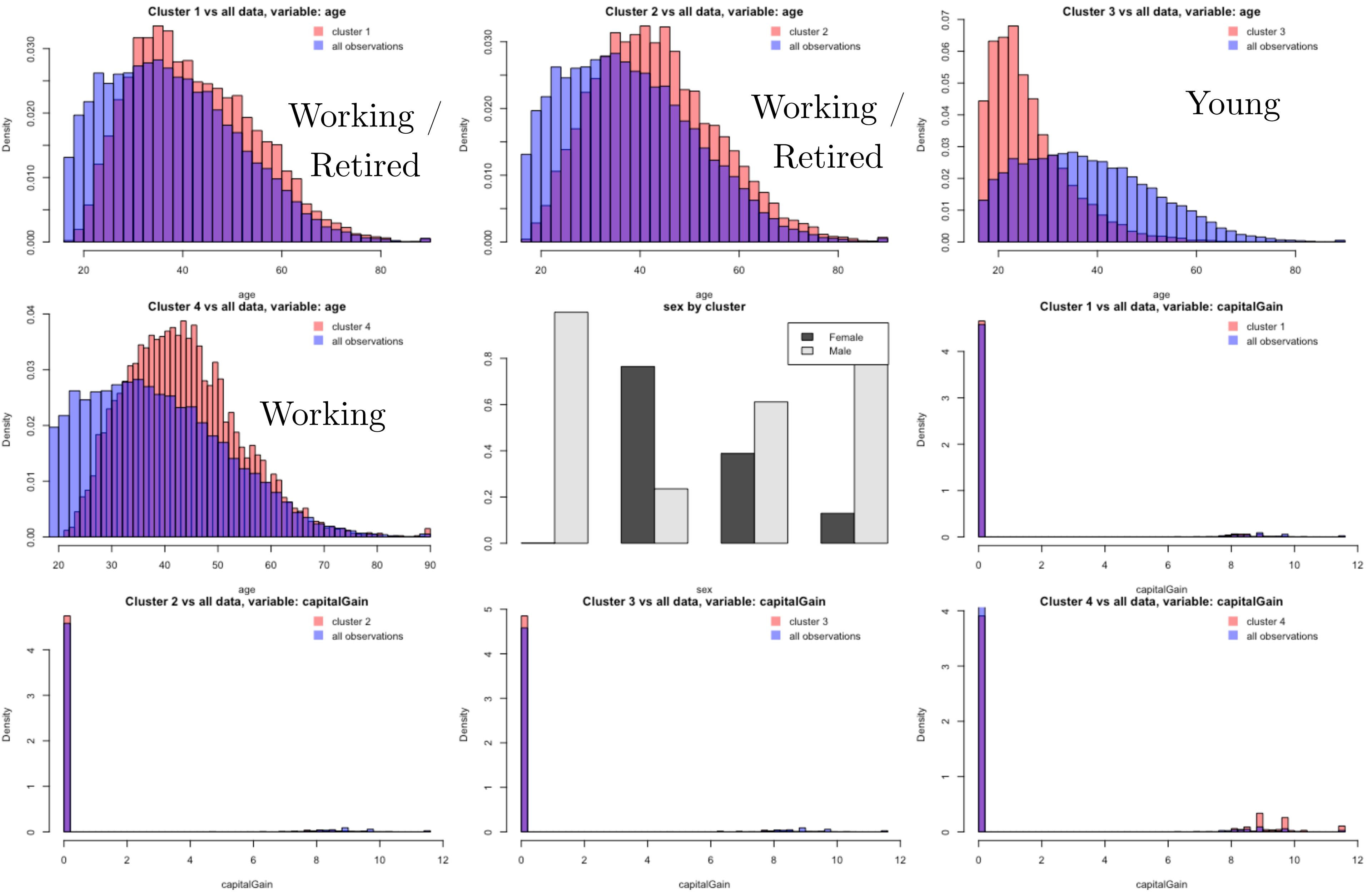
Adult Dataset

• • •

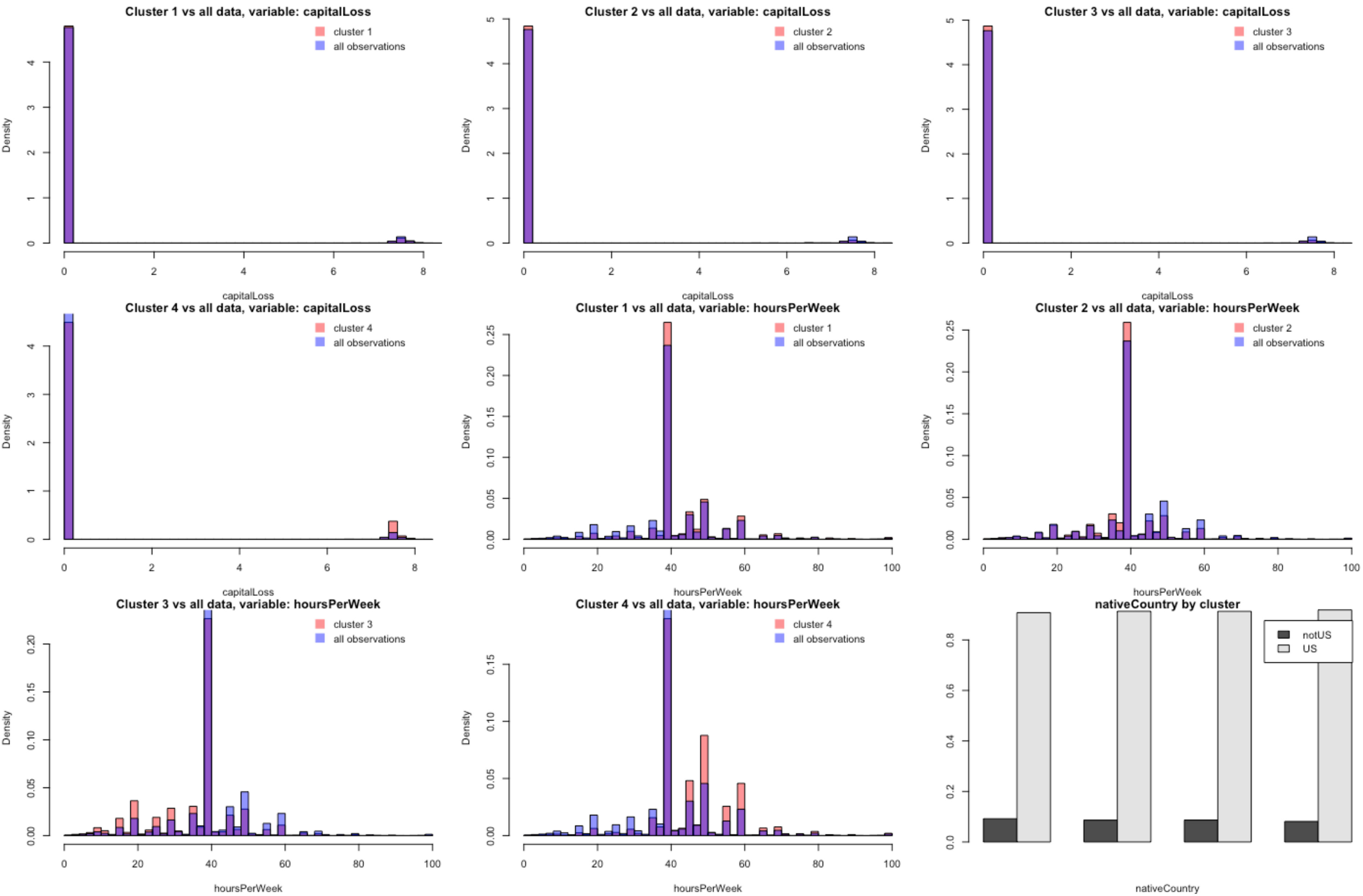
Steps

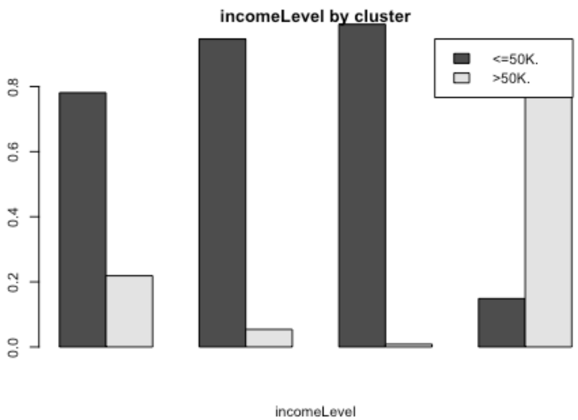
1. Explore your data. Make any transformations necessary
2. Create dummy columns for categorical variables as necessary
3. Reduce the dimensionality of your data if desired
    1. PCA
    2. SVD
4. Create a bunch of clustering using the *scores* from step 3 as input and a range of possible values for k.
5. Create the Consensus Matrix, $\mathbf{C}$, or the "Pre-consensus" Matrix, $\mathbf{H}$
6. Observe the singular values for a drop-off/elbow to find k
7. Cluster the matrix from step 5 for k-clusters (even better, cluster its first k singular vectors).
8. Visualize those clusters using scores on first 2 components from step 4


PC1
PC2
PC3
PC4
PC5

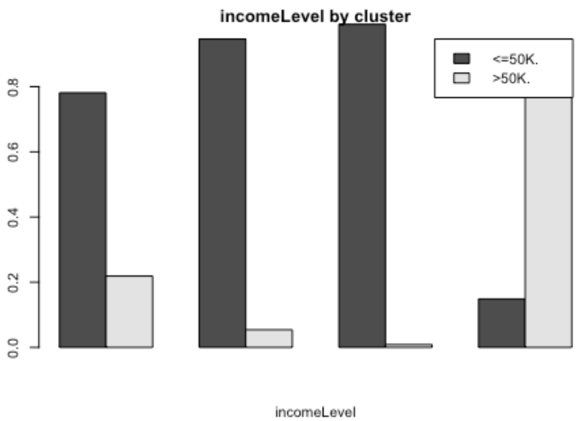


Cluster 1 vs all data, variable: age
cluster 1
all observations
Working / Retired
Density
age
Cluster 2 vs all data, variable: age
cluster 2
all observations
Working / Retired
Density
age
Cluster 3 vs all data, variable: age
cluster 3
all observations
Young
Density
age
Cluster 4 vs all data, variable: age
cluster 4
all observations
Working
Density
age
sex by cluster
Female
Male
sex
Cluster 1 vs all data, variable: capitalGain
cluster 1
all observations
Density
capitalGain
Cluster 2 vs all data, variable: capitalGain
cluster 2
all observations
Density
capitalGain
Cluster 3 vs all data, variable: capitalGain
cluster 3
all observations
Density
capitalGain
Cluster 4 vs all data, variable: capitalGain
cluster 4
all observations
Density
capitalGain


Cluster 1 vs all data, variable: capitalLoss
cluster 1
all observations
Density
capitalLoss
Cluster 2 vs all data, variable: capitalLoss
cluster 2
all observations
Density
capitalLoss
Cluster 3 vs all data, variable: capitalLoss
cluster 3
all observations
Density
capitalLoss
Cluster 4 vs all data, variable: capitalLoss
cluster 4
all observations
Density
capitalLoss
Cluster 1 vs all data, variable: hoursPerWeek
cluster 1
all observations
Density
hoursPerWeek
Cluster 2 vs all data, variable: hoursPerWeek
cluster 2
all observations
Density
hoursPerWeek
Cluster 3 vs all data, variable: hoursPerWeek
cluster 3
all observations
Density
hoursPerWeek
Cluster 4 vs all data, variable: hoursPerWeek
cluster 4
all observations
Density
hoursPerWeek
nativeCountry by cluster
notUS
US
nativeCountry



	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Age	Working+Retired	Working+Retired	Young	Working Ages
Sex	Male	Female	Balanced (Skew Female)	Same as population
Work hours	40 hours	35-40	<<40 hours	>>40 hours
Income	population par	<50K	<<50K	>>50K
Race	par	more diverse	more diverse	par
Workclass	higher % self-empt		more Bachelors, some college	more non private
Marital	Married	mostly divorced, few married	Single! Never Married	Married
Occupation	Craft/Repair	Admin/clerical/ service	Services	Professional - specialty
Relationship	Husband	Not in Family/ unmarried/wife	Not in Family / child of	Husband



	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Age	Working+Retired	Working+Retired	Young	Working Ages
Sex	Male	Female	Balanced (Skew Female)	Same as population
Work hours	40 hours	35-40	<<40 hours	>>40 hours
Income	population par	<50K	<<50K	>>50K
Race	par	more diverse	more diverse	par
Workclass	higher % self-empt		more Bachelors, some college	more non private
Marital	Married	mostly divorced, few married	Single! Never Married	Married
Occupation	Craft/Repair	Admin/clerical/ service	Services	Professional - specialty
Relationship	Husband	Not in Family/ unmarried/wife	Not in Family / child of	Husband