Cluster analysis in Python

Or, how to see neat patterns in everything, whether or not they're really there

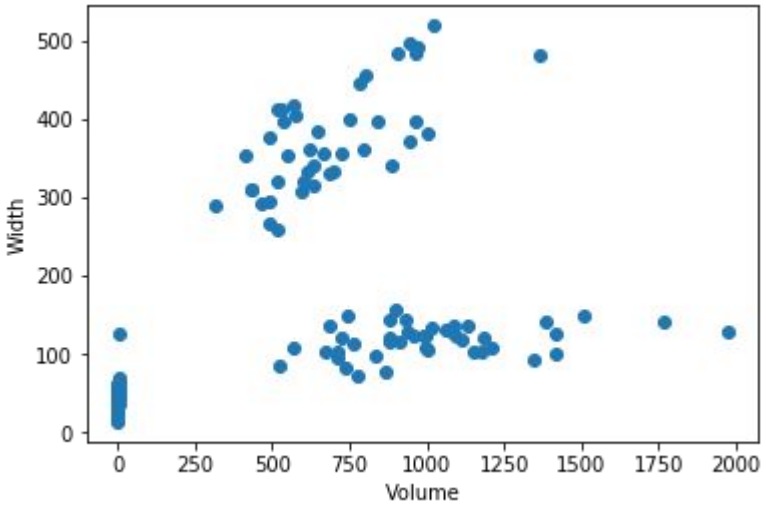
Tal Yarkoni

Supervised vs. unsupervised learning

- So far in this course, you've mostly covered *supervised* learning
  - The data are labeled—i.e., we already know the answer we're looking for
- In *unsupervised* learning, we have no knowledge of the ground truth
  - Our goal is to detect interesting/useful structure within the data

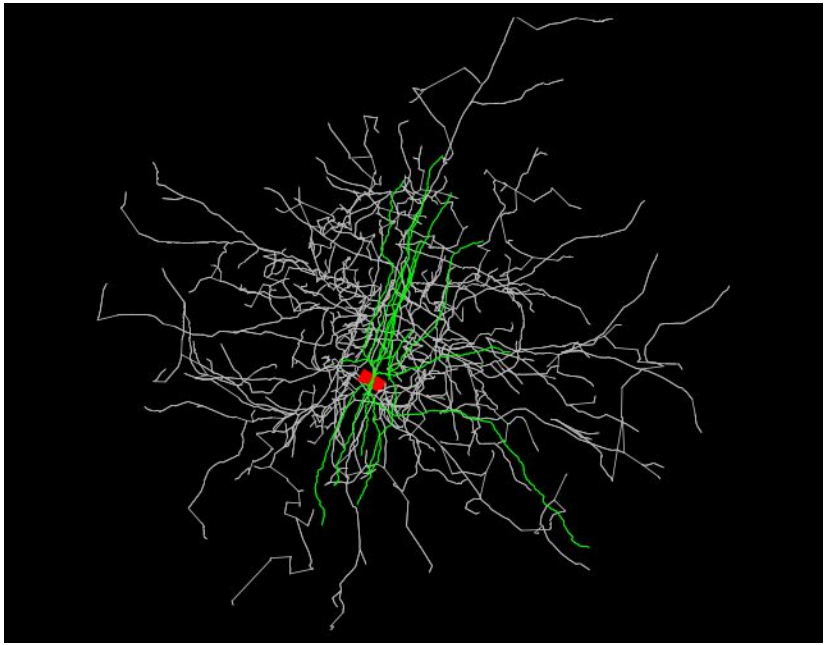
Clustering analysis

- One common form of unsupervised learning
- The goal is to partition a continuous space into discrete clusters and assign each data point to a cluster



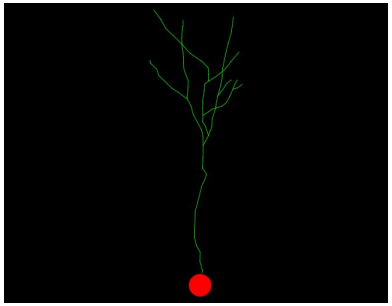
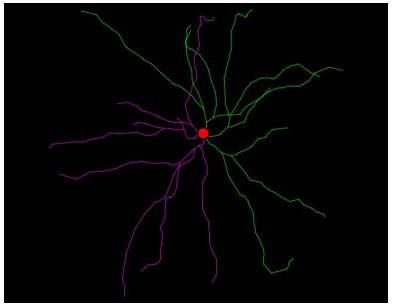
Neuromorpho.org

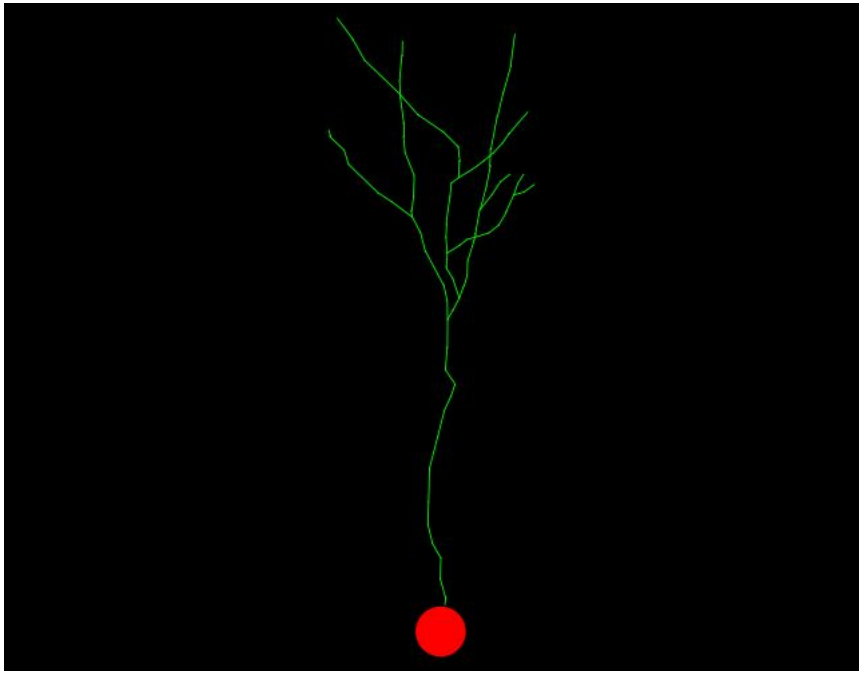
- A database of digitally reconstructed neurons
- > 100k neurons from > 400 labs



Our Neuromorpho sample

- 122 neurons from 3 groups:
  - 37 mouse hippocampal granule neurons
  - 42 human neocortical pyramidal neurons
  - 43 zebrafish olfactory bulb neurons

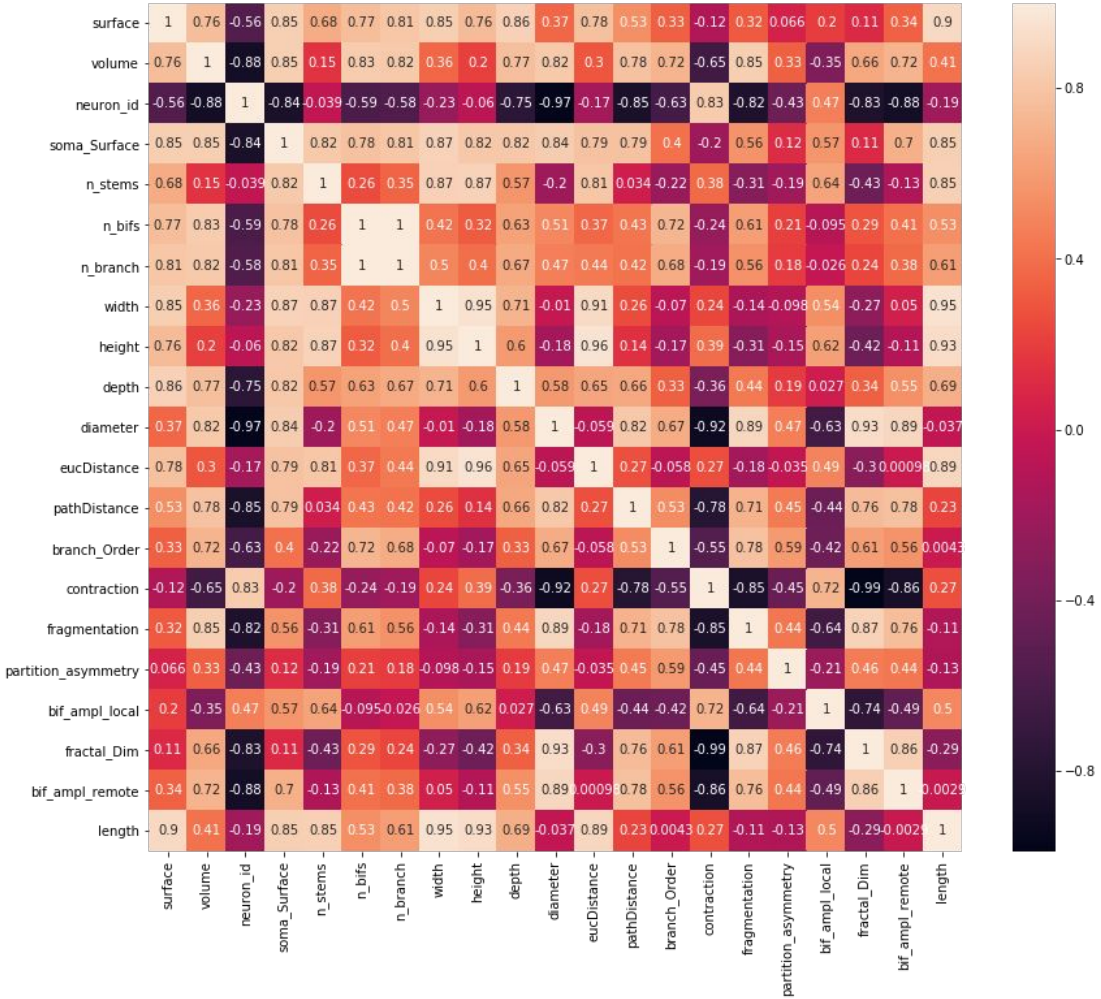



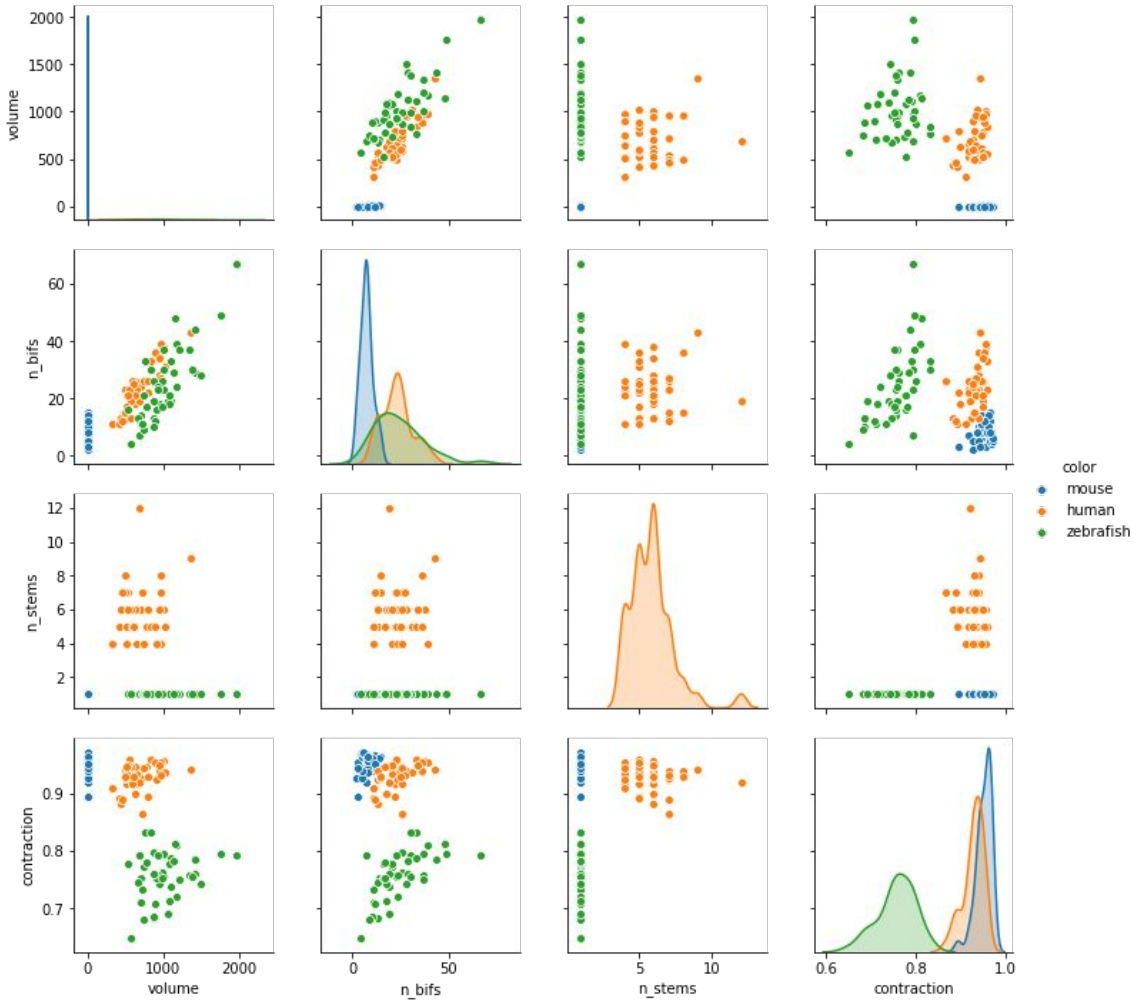



Measurements	
Soma Surface :	213.96 μm^2
Number of Stems :	1
Number of Bifurcations :	7
Number of Branches :	15
Overall Width :	29.98 µm
Overall Height :	93.6 µm
Overall Depth :	6.38 µm
Average Diameter :	0.09 µm
Total Length :	278.18 µm
Total Surface :	78.65 μm^2
Total Volume :	1.77 μm^3
Max Euclidean Distance :	99.2 µm
Max Path Distance :	113.64 µm
Max Branch Order :	4
Average Contraction :	0.92
Total Fragmentation :	131
Partition Asymmetry :	0.17
Average Rall's Ratio :	2
Average Bifurcation Angle Local :	53.03°
Average Bifurcation Angle Remote :	43.41°
Fractal Dimension :	1.03

http://neuromorpho.org/neuron_info.jsp?neuron_name=DD13-10-c6-3

- How do these various measures relate to one another in our sample?
- Useful for picking out non-redundant features to cluster on



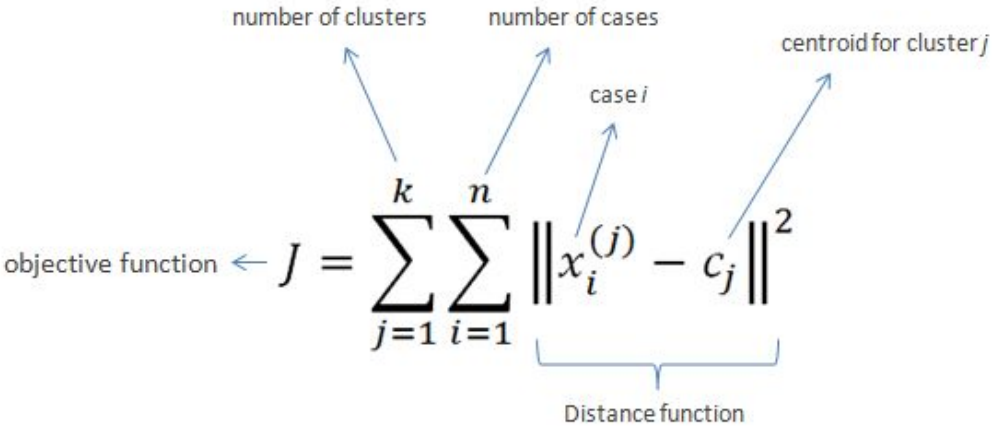

volume
n_bifs
n_stems
contraction
color
mouse
human
zebrafish

Let’s cluster our data!

- Let’s see if we can recapture the true labels using clustering methods
- There are a *lot* of clustering methods out there
- We’ll start with one of the simplest and most popular: k-means

K-Means clustering

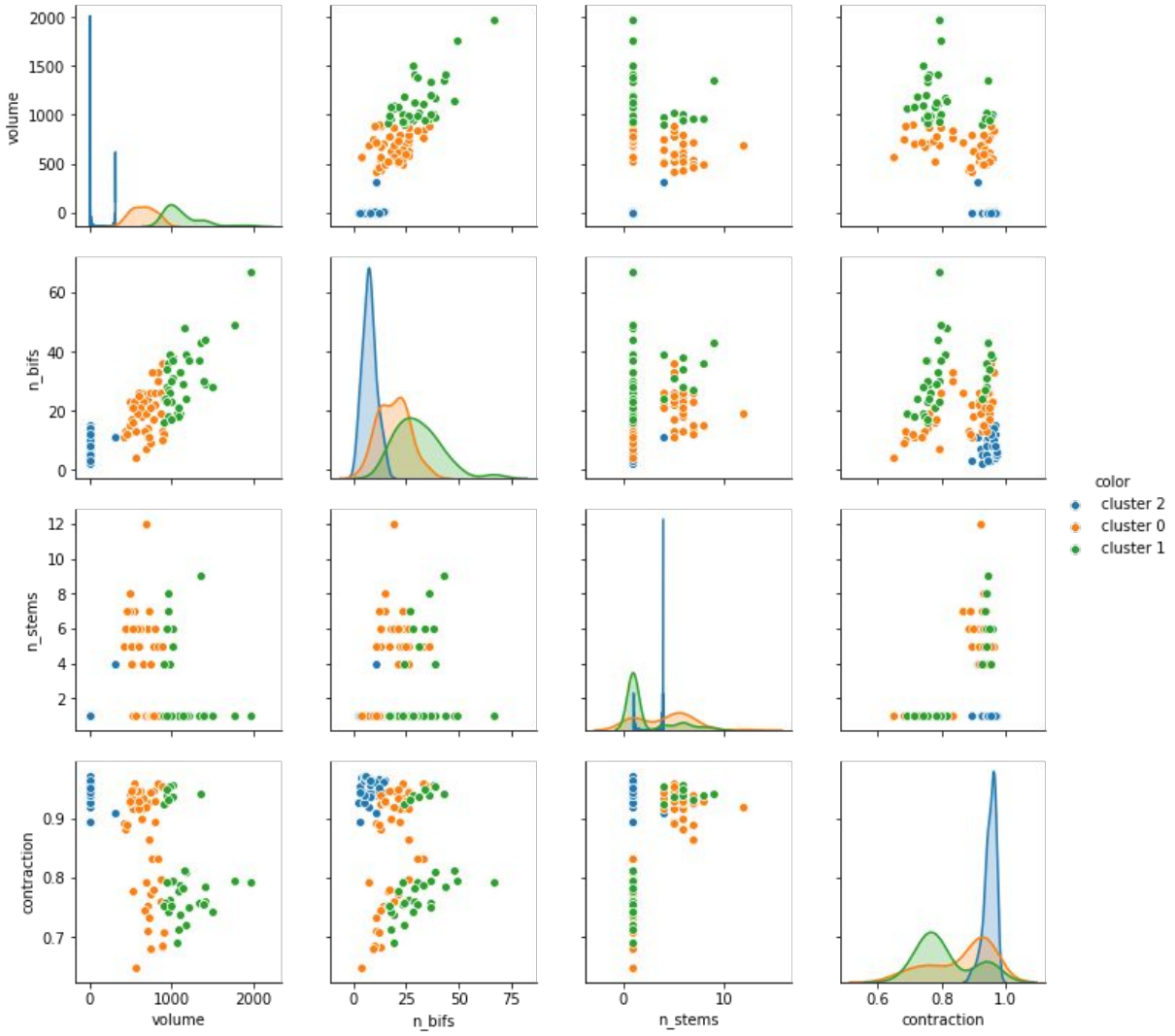
Intuition: a good clustering should minimize distance between the points *within* each cluster



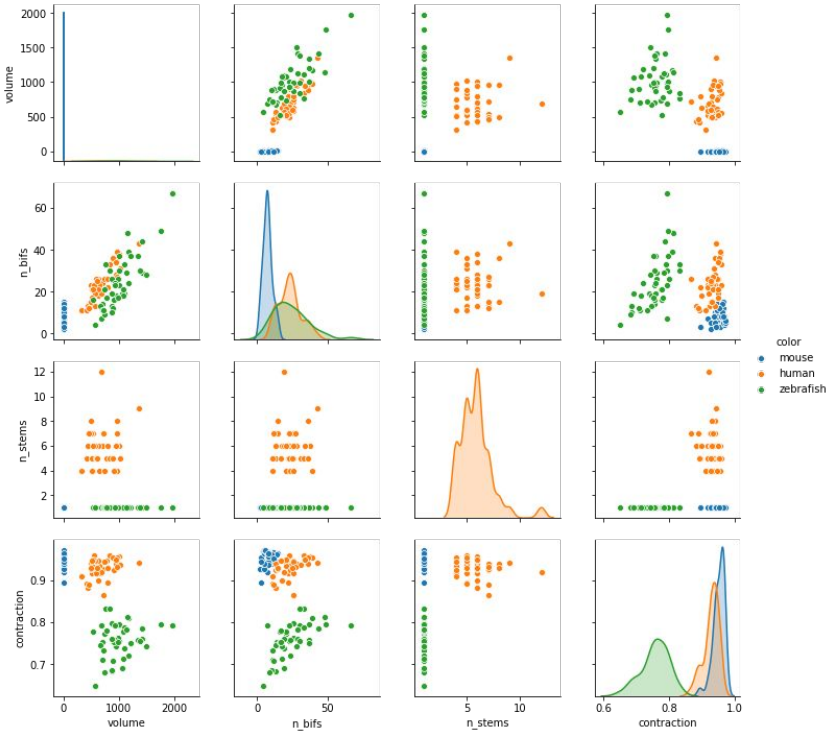
https://www.saedsayad.com/clustering_kmeans.htm

Clustering neurons with k-means

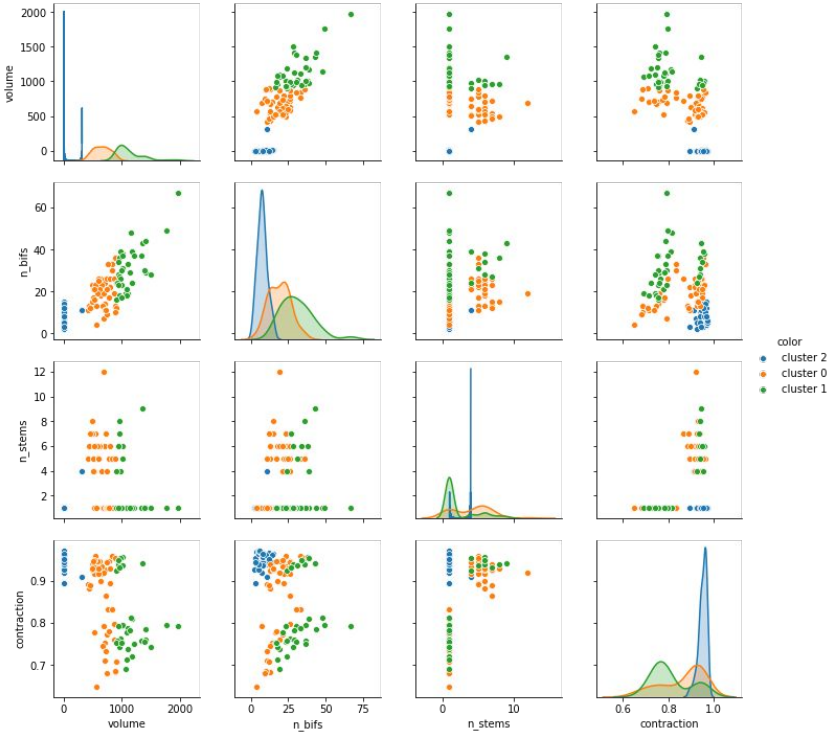
- So what do our Neuromorpho data look like if we cluster them using these 4 variables?
  - Volume
  - No. of bifurcations
  - No. of stems
  - Contraction


volume
n_bifs
n_stems
contraction
color
cluster 2
cluster 0
cluster 1

Ground truth

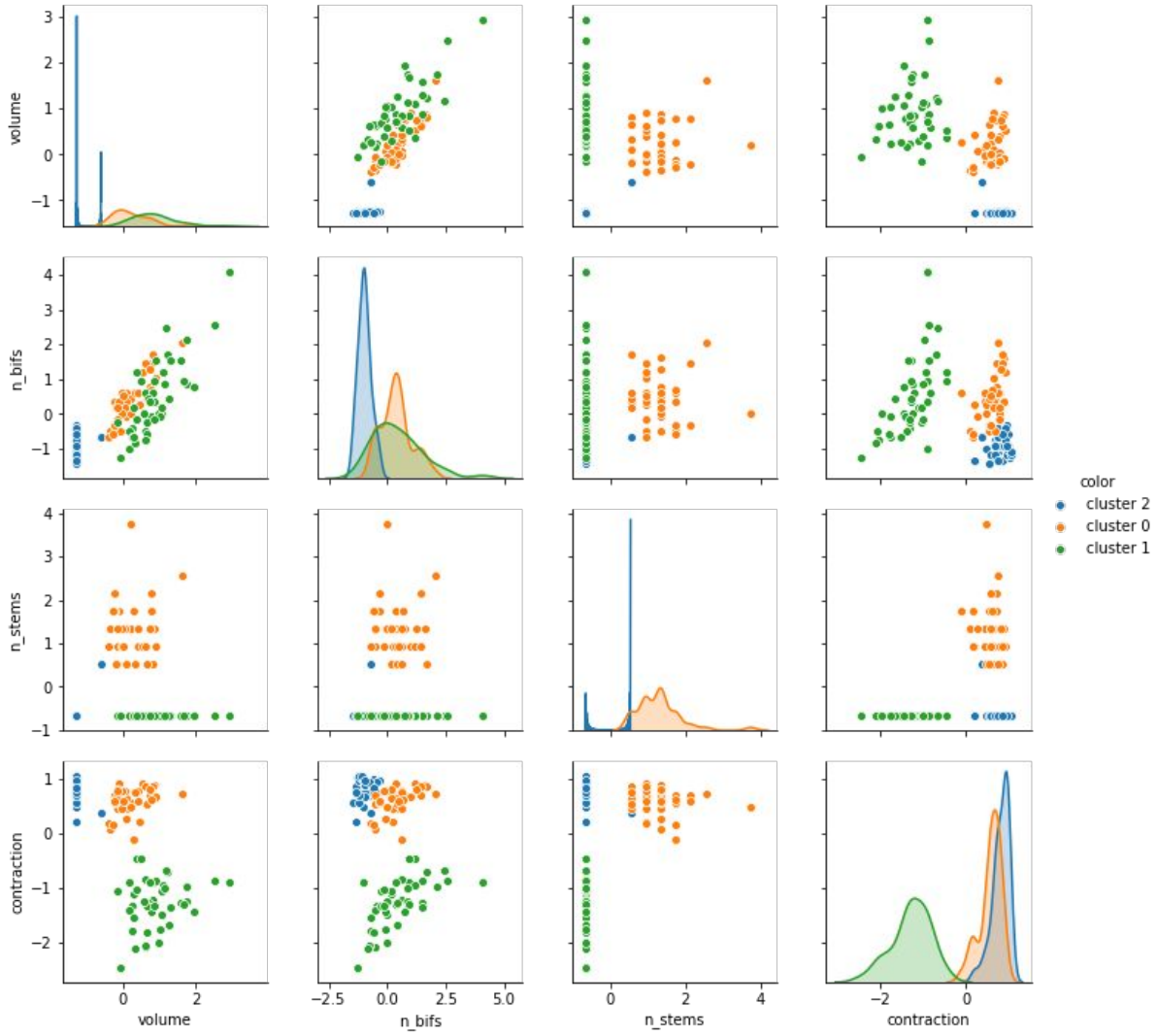


K-means clustering



What do we think?

- K-means is clearly finding some meaningful boundaries between our neuron types
- But it's far from perfect
  - Should we expect perfection, given the data we started with?
- Can we improve performance by selecting different features or otherwise transforming the data?
  - E.g., k-means is sensitive to scale… maybe we should standardize our features?

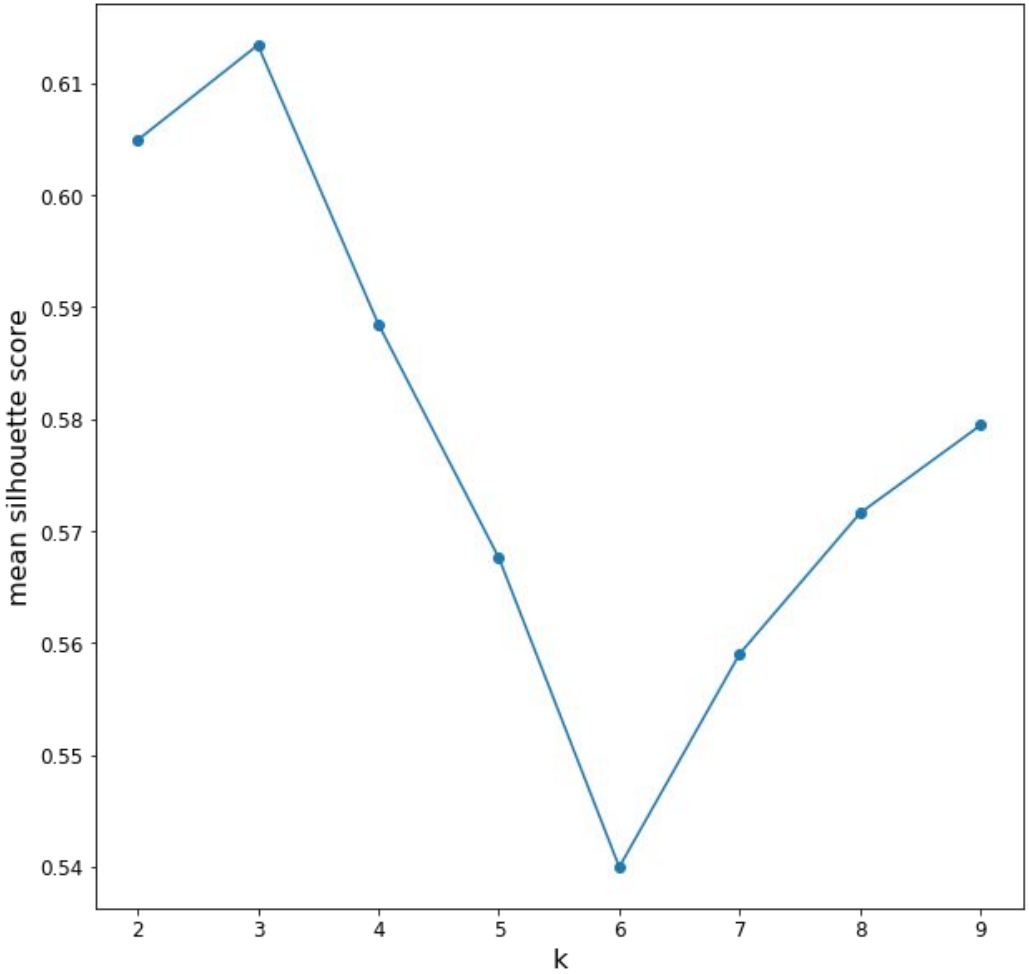

volume
n_bifs
n_stems
contraction
color
cluster 2
cluster 0
cluster 1

How many clusters?

- How do we know what the right value of *k* is?
  - In our example, we cheated and used the ground truth of 3
  - We usually don't know this!
  - Worse: there often (usually?) *isn't* any single right value
- If we don't have good priors, we can try to determine *k* from the data
  - There are a *lot* of metrics we can use
  - scikit-learn's metrics module contains quite a few (`sklearn.metrics`)
- E.g., silhouette score:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

- Here are the mean silhouette scores for our sample as a function of *k*
- *k* = 3 seems optimal, right?
- Any reservations about this? Is anything important missing?



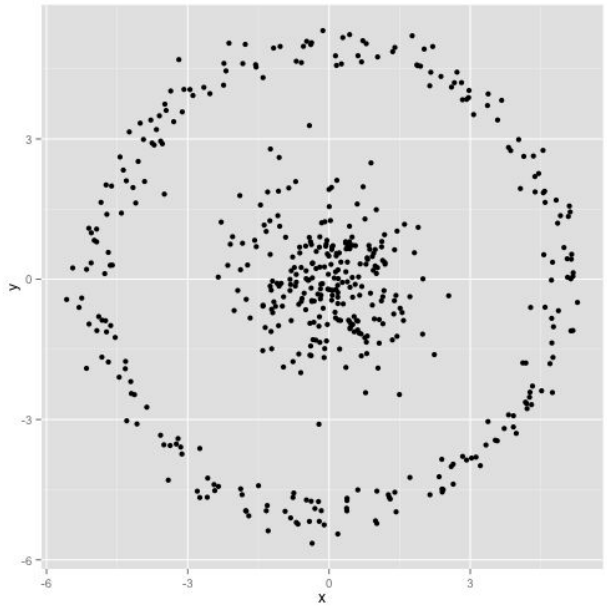
Buyer beware!

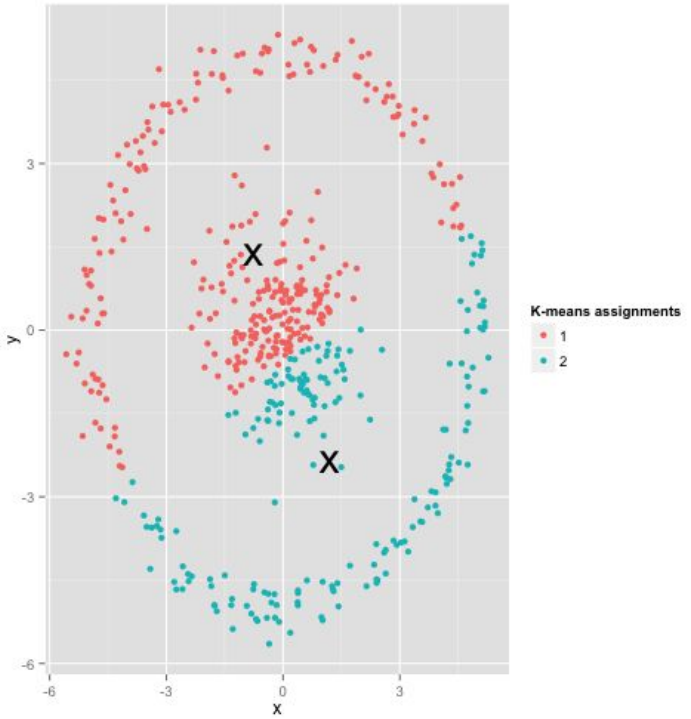
- Most clustering methods are utterly lacking in self-awareness
  - They won't tell you when they're unable to give a sensible answer
  - Beware of pareidolia: it's *really* easy to see interesting patterns in data
  - As an exercise, you can play with the clustering code and see how many sweet little lies you can convince yourself of

Buyer beware!

- If you apply a give to data that don’t fit its assumptions, you’re gonna have a bad time!
  - E.g., k-means doesn’t like clusters of unequal sizes or within-cluster variance
- It’s critical to think about the match between estimator and data
  - Use a method that makes sane assumptions for your use case
  - Consider transforming your data

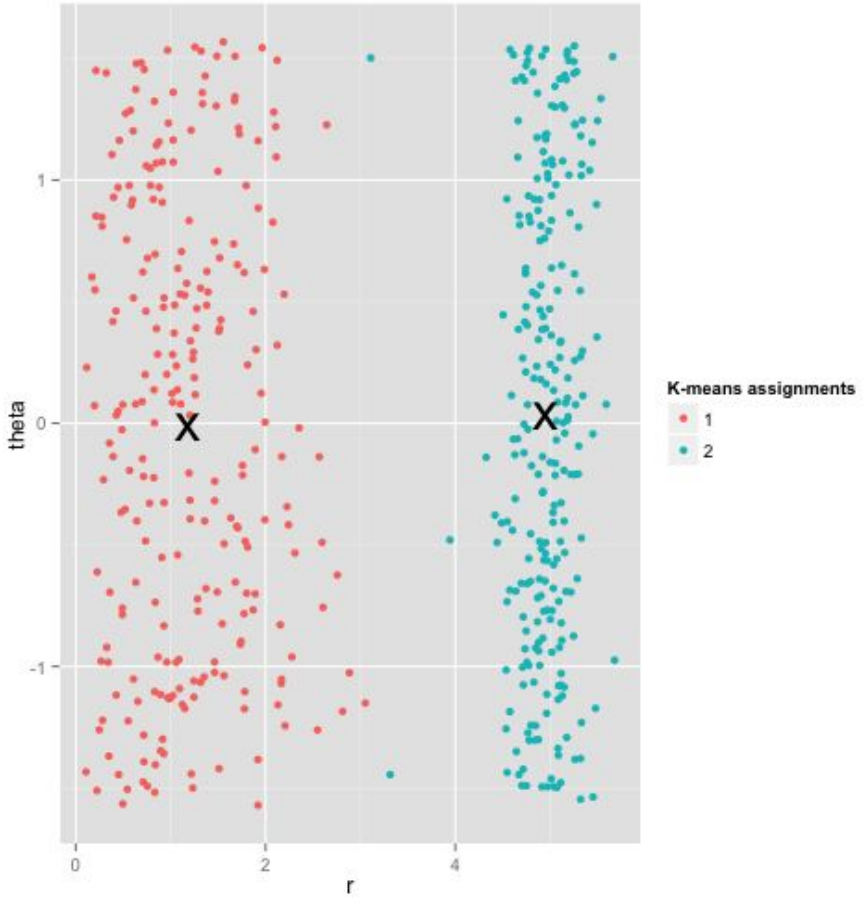
Ruh roh...





- http://varianceexplained.org/r/kmeans-free-lunch/

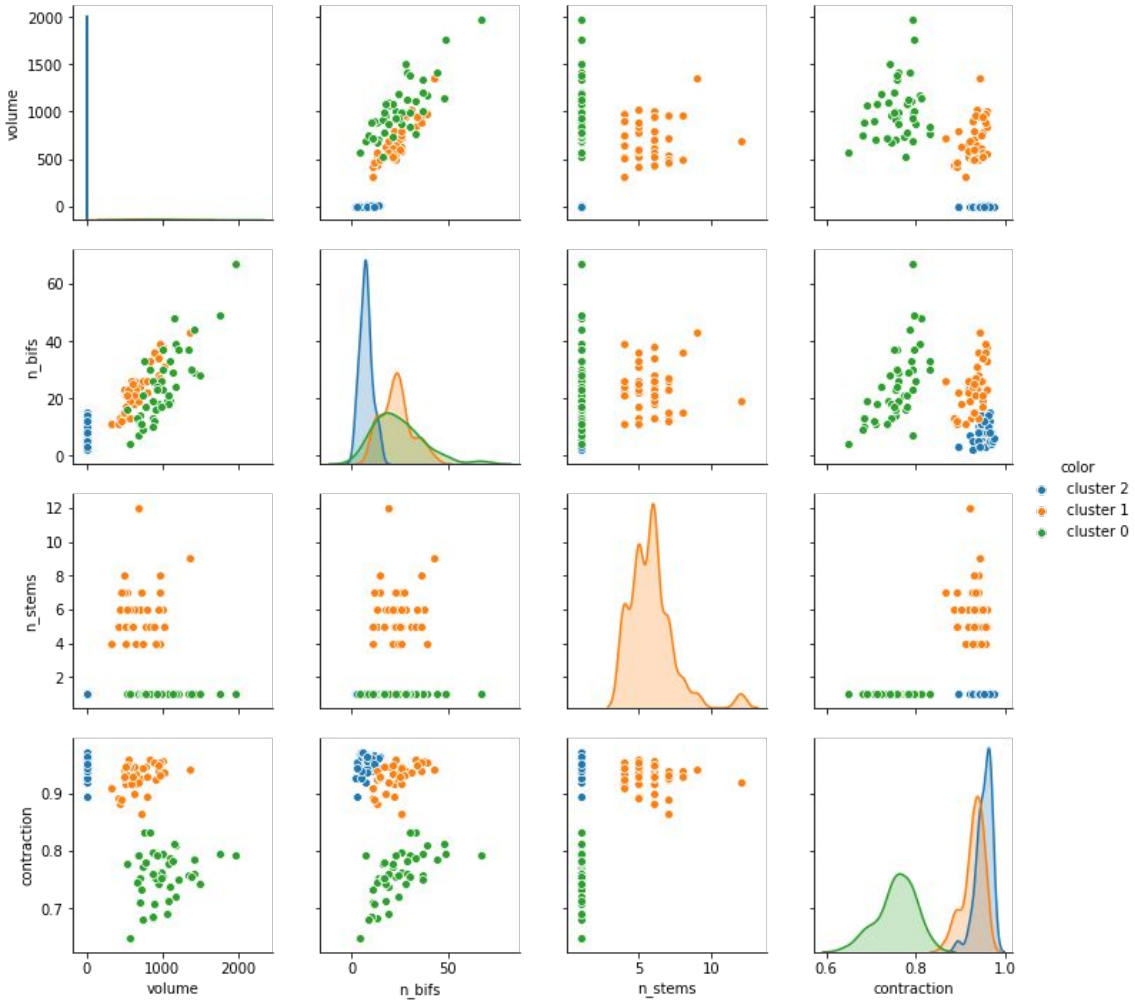
In polar coordinates...

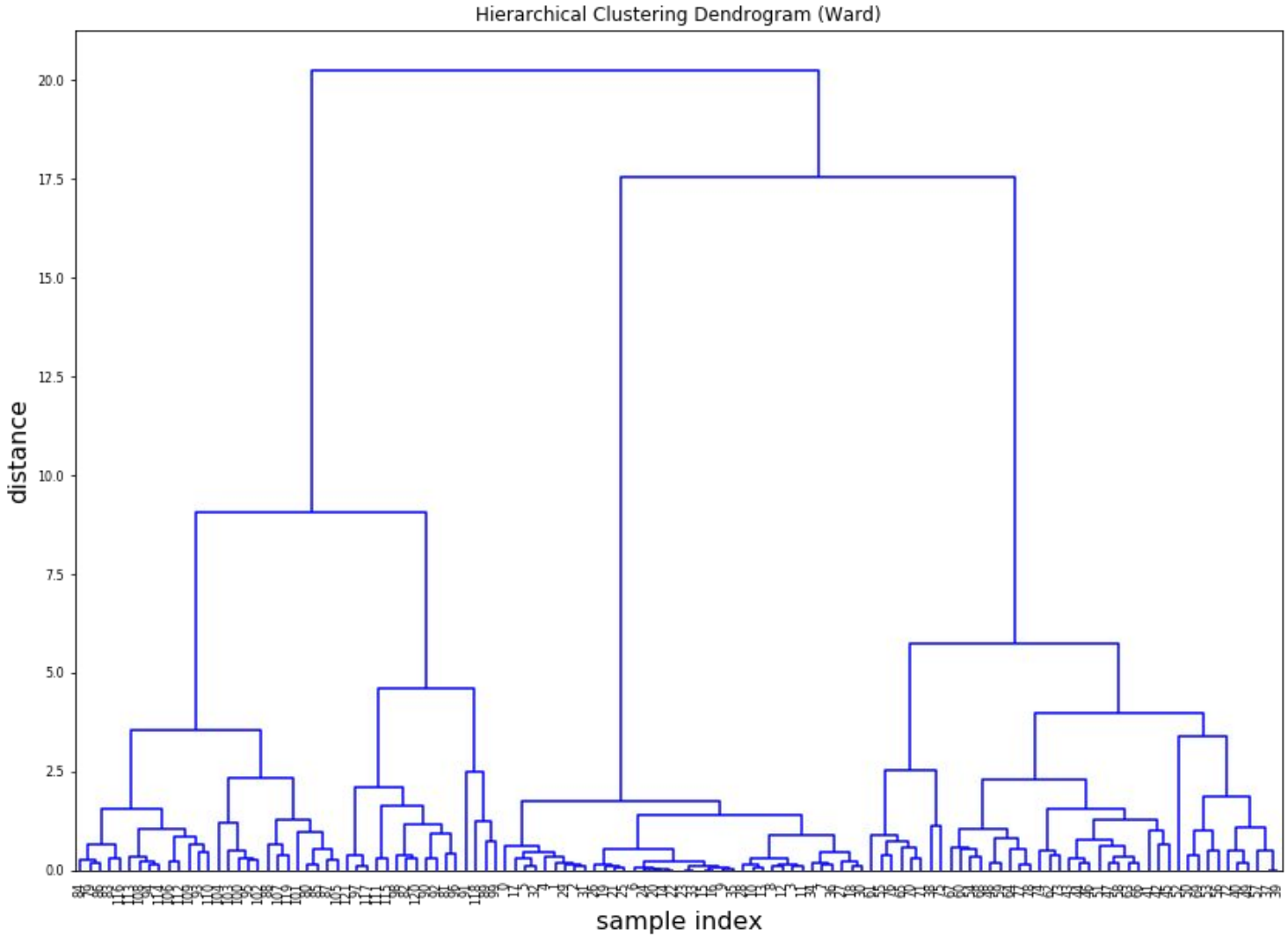


- http://varianceexplained.org/r/kmeans-free-lunch/

Agglomerative clustering

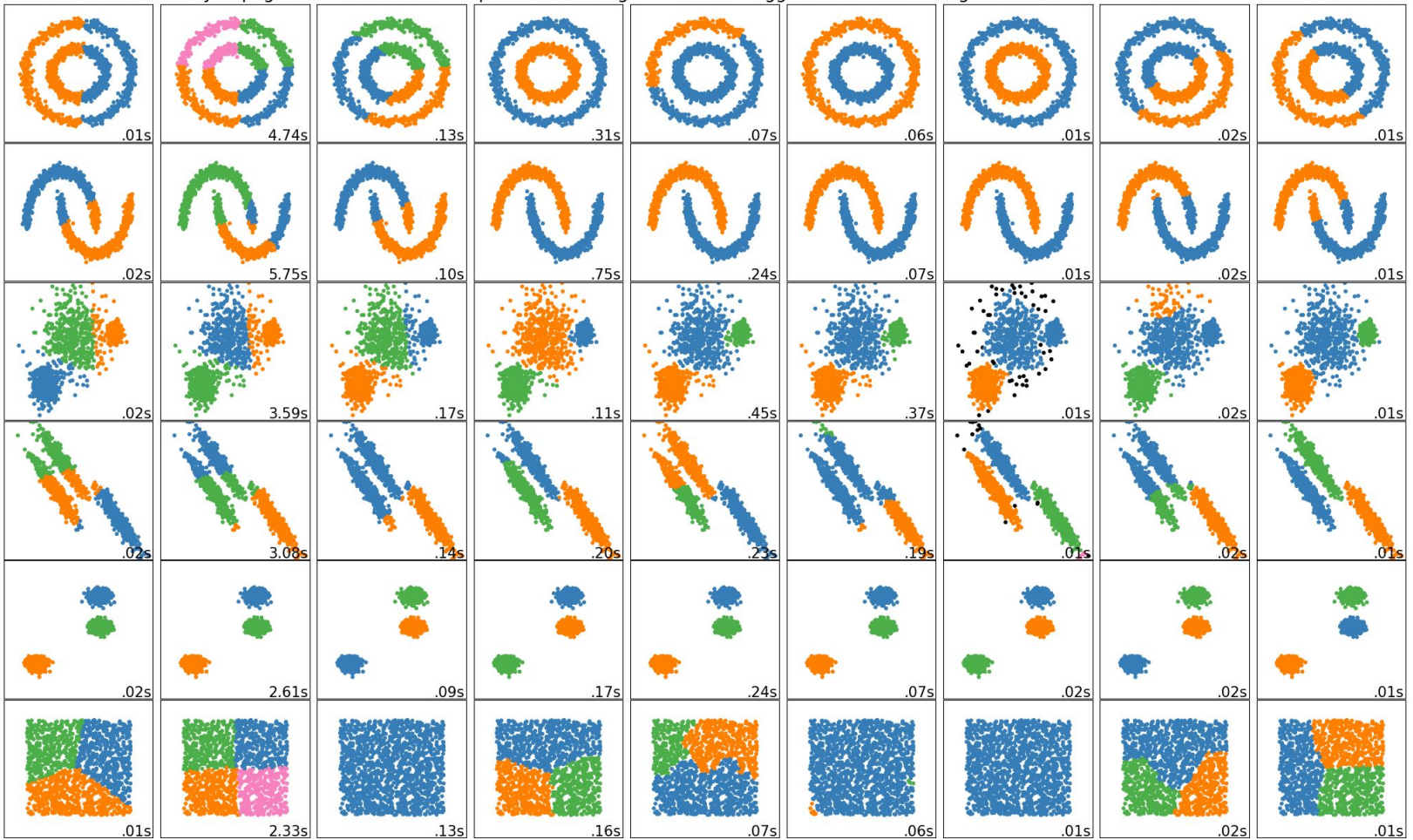
- A “bottom-up” form of hierarchical clustering
- The name reflects the core idea: we start with individual data points and gradually agglomerate or merge them into ever-larger clusters
- Ingredients:
  - A distance metric (e.g., Euclidian distance in feature space)
  - A linkage rule (e.g., minimize the variance of merged clusters)
- Proceed sequentially, merging the two closest pairs at each step
- Depending on algorithm, this can be very slow (O(n^3))


volume
n_bifs
n_stems
contraction
color
cluster 2
cluster 1
cluster 0


Hierarchical Clustering Dendrogram (Ward)
distance
sample index
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0

Other clustering methods

- There are hundreds of clustering methods out there
- Common packages like scikit-learn implement many common methods
- Vary widely in distance metric, assumptions, scalability, etc.
    - Some have free parameters that require tuning for good performance



https://scikit-learn.org/stable/modules/clustering.html#clustering

So… what's the best overall clustering method?

- Widely acknowledged to be the Smithsonian Kernel Popper
- No, not really
- Real answer: there isn't one*

No Free Lunch

- A set of theorems demonstrating that, over the space of all possible datasets, no estimator is better (on average) than any other
- Does this mean that the estimator we choose makes no difference?
  - No! The datasets we use aren't sampled at random from all possible datasets
- The key is selecting the right tool for the job

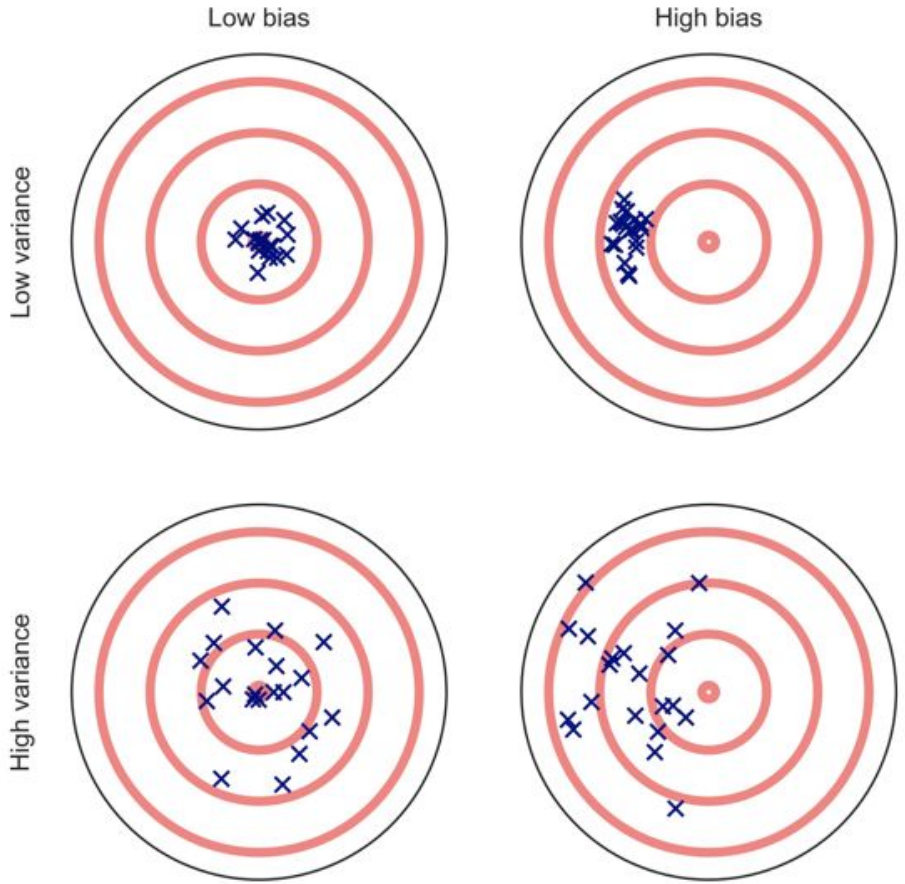
Remember regularization?

- Ridge regression: add an L2 penalty to ordinary least squares regression
- Is ridge regression better than OLS?
- In many cases, yes, but… it depends!
  - How big is the penalty?
  - How much data do you have available for hyperparameter optimization?
  - What does your data look like?

The bias-variance tradeoff

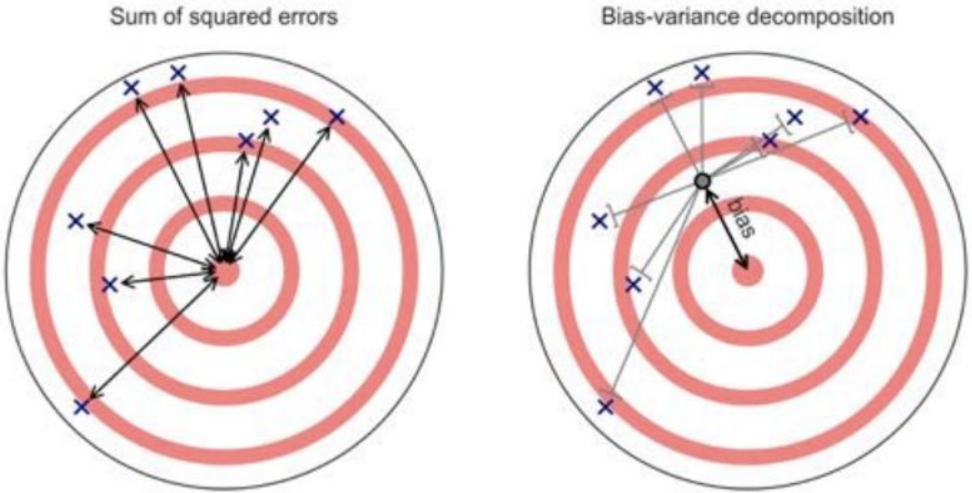
- Scientists learning classical statistics are often taught that bias is bad
- Bias *can* be bad, but it doesn't have to be
- If you have good prior knowledge, you should *bias* your data in the direction of your priors
    - E.g., when you misplace your keys, you don't look *everywhere* with equal probability

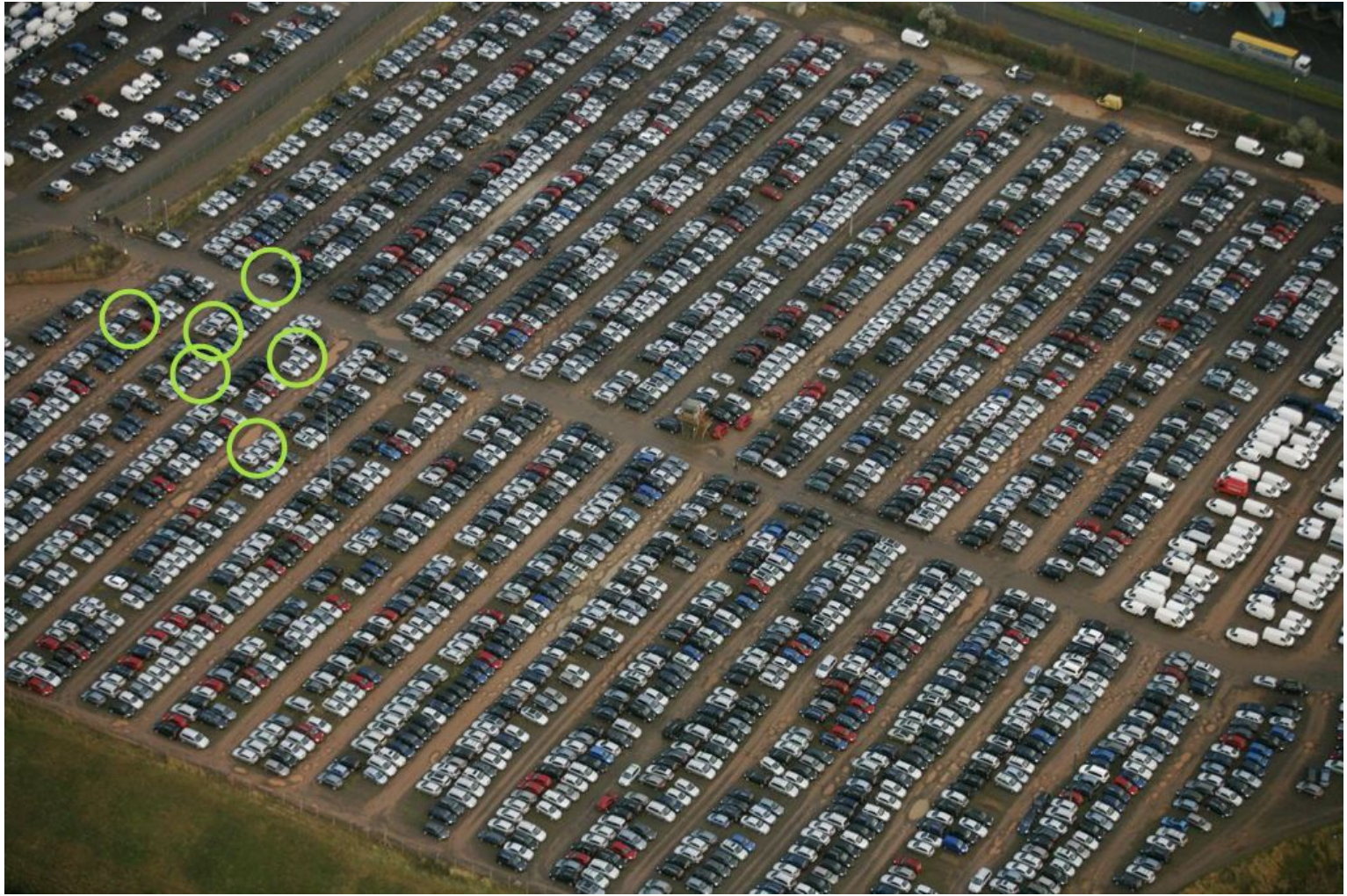
Bias vs. variance

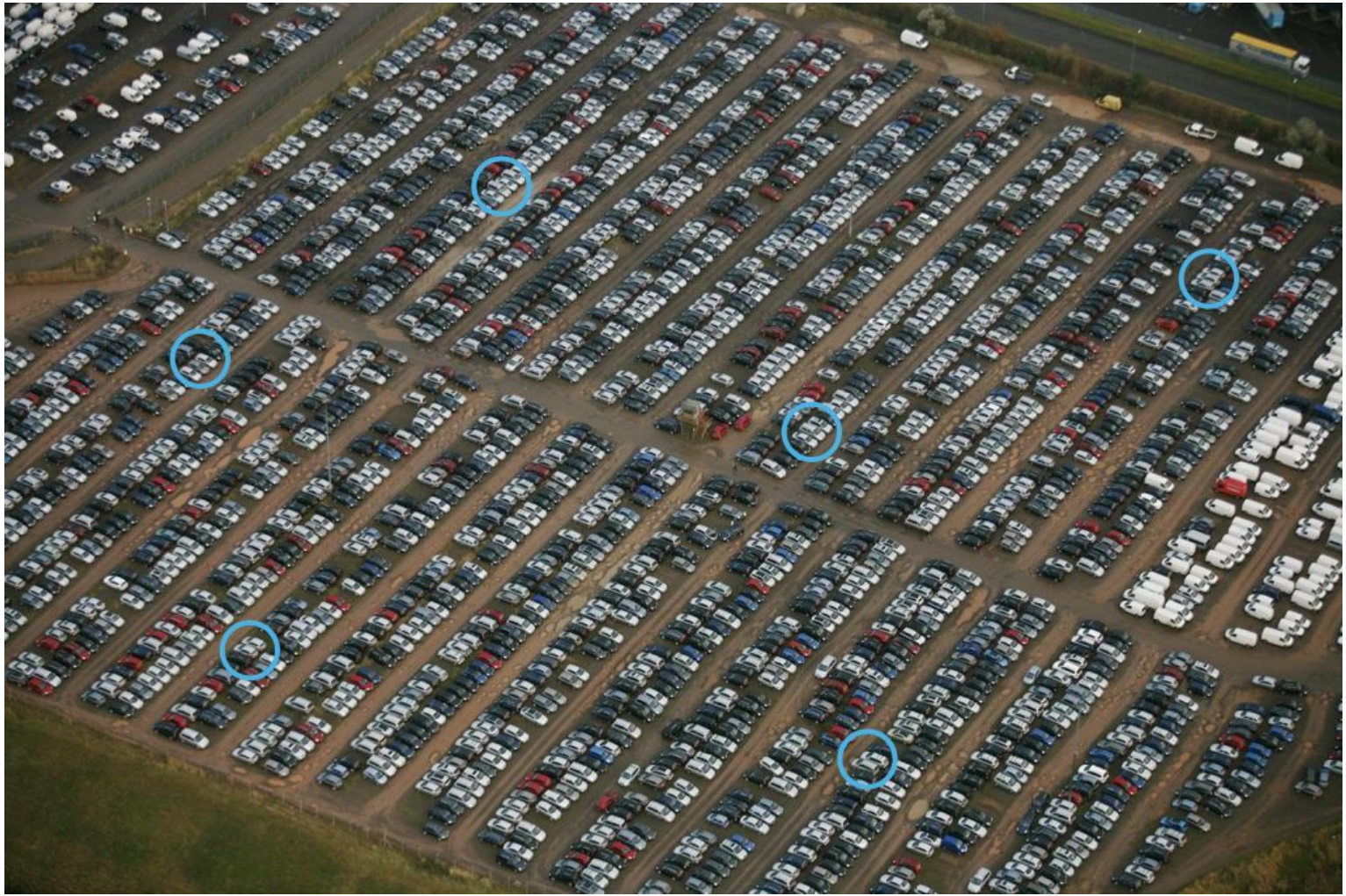


The bias-variance decomposition

- Total error in the SSE model can be decomposed into bias and variance components
- There is a tradeoff between these things: you can (often) reduce one by increasing the other







So, how do you select the optimal estimator?

- The key is to match the estimator to the context and data
- This requires careful thought, analysis, and/or domain expertise
- Remember: there’s no free lunch!