HiOA Big Data Course

Session 5 - Time Series 101 + Unsupervised Learning

Dirk Hesse

Time Series Analysis

- We actually don't have much time to talk about it TS analysis.
- So we'll be very, very brief.
  - But it's an important topic.
  - So read up on it!

Time Series Analysis

*A **time series** is a series of data points indexed (or listed or graphed) in time order.*

- Typically $X(t)$
.
- Or X_t.

Time Series: Examples

- Temperature measurements.
- Visitor counts.
- Things sold.
- Ocean tides.
- Sun spot counts.
- Stock prices.

How can we analyze it?

- What is the time scale?
  - Hours?
  - Years?
  - Microseconds?
- Is there a general trend?
  - Linear fit?
  - We'll generally assume there isn't (stationary process).
- Is there periodicity?
  - Fourier transform?
  - Autocorrelation function?

Correlation and Autocorrelation

$$\mathrm{Cor}(X, Y) = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y}$$

$$\mathrm{R}(\tau) = \frac{E[(X_t - \mu)(X_{t+\tau} - \mu)}{\sigma^2}$$

- Time series with periodicity have a non-zero ACF.

A simple time series model.

$$X(t) = a + bt + ct^2.$$

- Doesn't take periodicity into account.

Direct (Component) Modeling

$$X(t) = \alpha_0 \,\mathrm{hour}(t) + \alpha_1 \,\mathrm{monday}(t) + \ldots$$

$$\mathrm{monday}(t) = \begin{cases} 1 & \text{if } t \text{ on a Monday} \\ 0 & \text{else} \end{cases}$$

- Periodicity explicitly assumed.

Auto-regressive Moving Average (ARMA)

$$X(t) = c + \sum_{i=1}^{p} \alpha_i X(t-i) + \epsilon_t + \sum_{i=1}^{q} \theta_i \epsilon_{t-i}$$

- Auto-regressive + moving average.
- Can be used separately.

Unsupervised Learning

- Supervised learning:
  - Training data, $(x_i, y_i), i = 1, \ldots, N$.
  - Target: y_i.
  - Variables: $x_i = (x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(K)})^T$.
  - Want to find $\hat{y}(X) = \mathrm{E}(Y|X)$.
- Unsupervised learning:
  - Have a bunch of x_i, no target.
- Try to make sense of the x_i.
  - Sometimes means finding an approximation of $p(X = x)$ given the training data.
  - Gives a measure how improbable the observation is.
  - Sometimes means finding groups in the data.

Clustering

Definition

Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters).

Examples

- Types of tissues in medical imaging.
- Cluster consumers in market segments.
- Analyze social networks.
- Group search results.
- Image segmentation.

Clustering

- Objective: Find k clusters of points that are close together.
  - Some methods find k automatically.
  - Most methods need it as input.
- Needed: Some measure for distance between measurements.
  - I.e. a metric.
- Needed: A measure for distance of clusters (linkage).
  - Total.
  - Minimal.
  - Maximal.
  - Average.

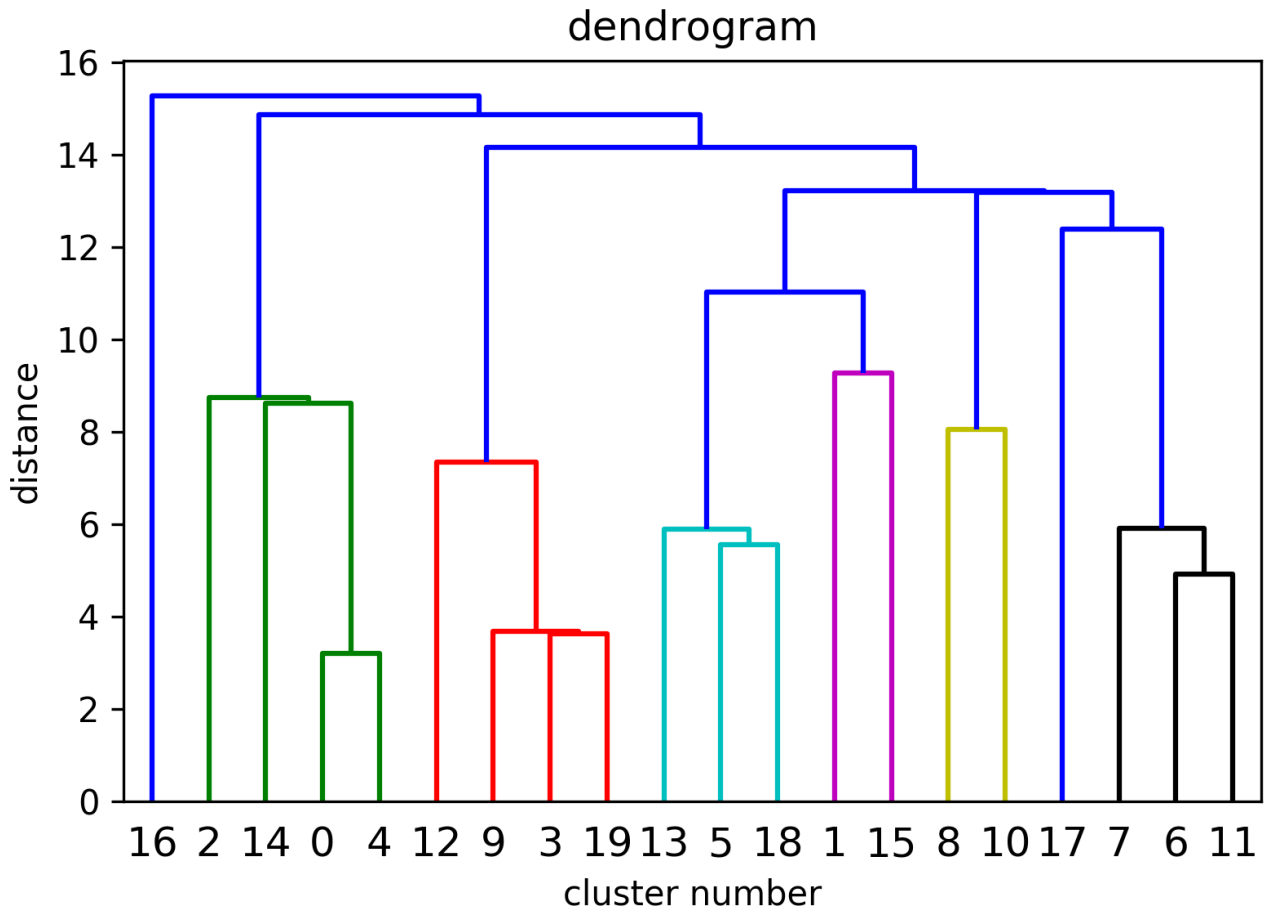
Types of Clustering

- Hierarchical
  - Arranges samples in a hierarchy, according to distance.
  - Types:
    - Agglomerative (bottom-up)
    - Divisive (top-down, not terribly common)
- Non-hierarchical
  - Examples:
    - K-Means.
    - DBSCAN.

Agglomerative Clustering

- Start with each x_i defining its own cluster.
- At each step, merge the closest two clusters.
  - Closest according to linkage you've chosen.
- Keep going until you have only one cluster.
- Choose a place to 'cut' the resulting tree.

Agglomerative Clustering



Distance between two clusters?

- Linkage
  - Single (closest points of clusters).
  - Full (furthest points of clusters).
  - Average (average distance of points).
  - Centroid (distance of cluster centers).

How to find the number of clusters?

- Sometimes given.
  - Want to distribute customers to k service people.
- Sometimes can be 'eyeballed' by looking at the dendrogram.
- Sometimes we need to work a little.
  - Number of classes might be unknown.
  - Might be questionable if there is structure at all.

Preparations for finding k numerically.

- Given clustering with k cluster centers μ_l.
- Assign each x_i a cluster $C(x_i)$, such that

$$C(x_i) = \operatorname*{argmin}_l \|x_i - \mu_l\|$$

- For agglomerative clustering we'll often have the $C(x_i)$ first and calculate

$$\mu_l = \frac{1}{N_l} \sum_{i; C(x_i)=l} x_i$$

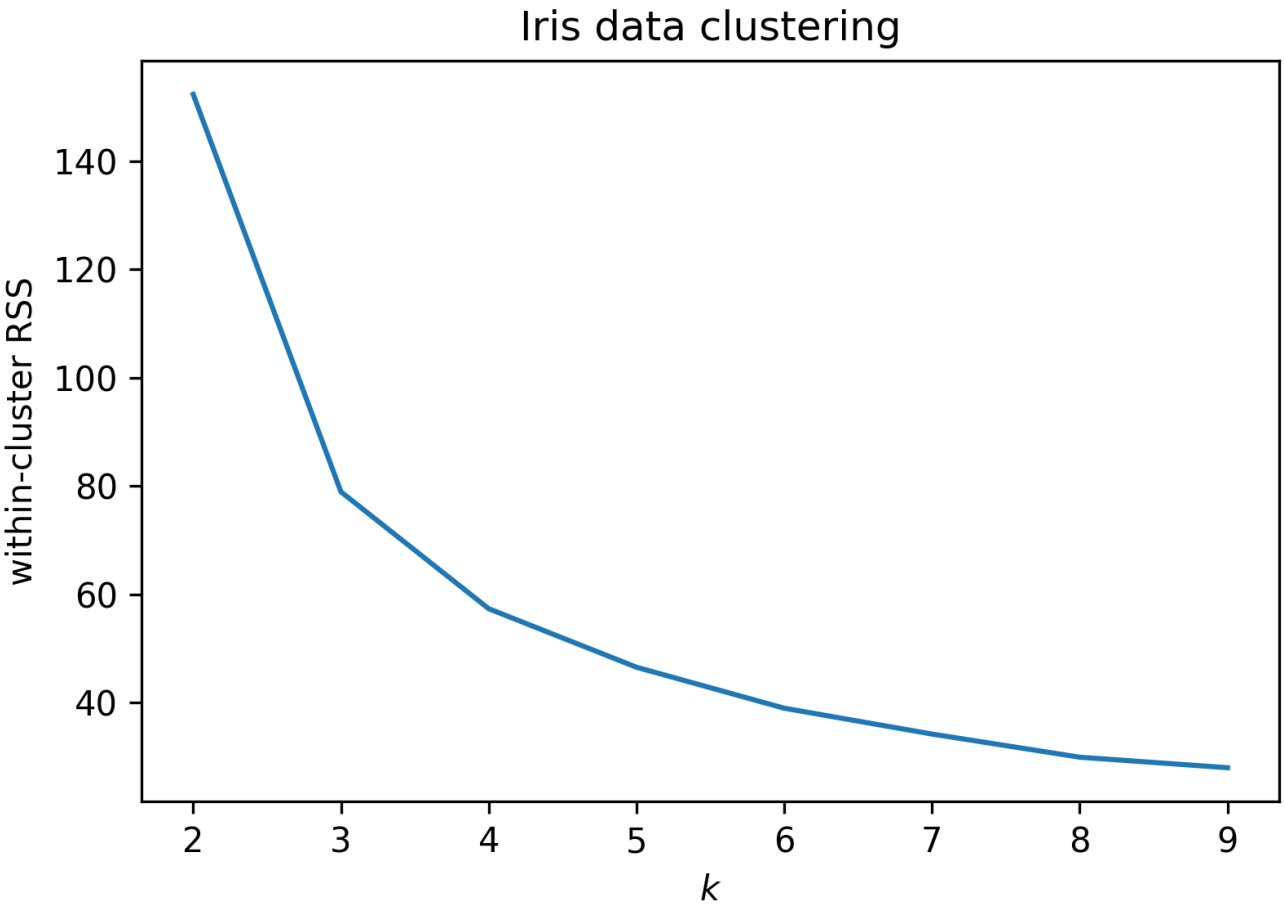
- Define the inertia (or within-cluster RSS)

$$\sum_l \sum_{i; C(x_i)=l} \|\mu_l - x_i\|^2$$

Finding k numerically.

- Set a range k you want to explore.
- Calculate the inertia for each of them.
- Plot vs. k.
- Often you'll see a knee-shape.
  - Less clusters than optimal: High reduction in variance.
  - More clusters than optimal: Don't gain much.
- Choose k at or close to 'knee'.

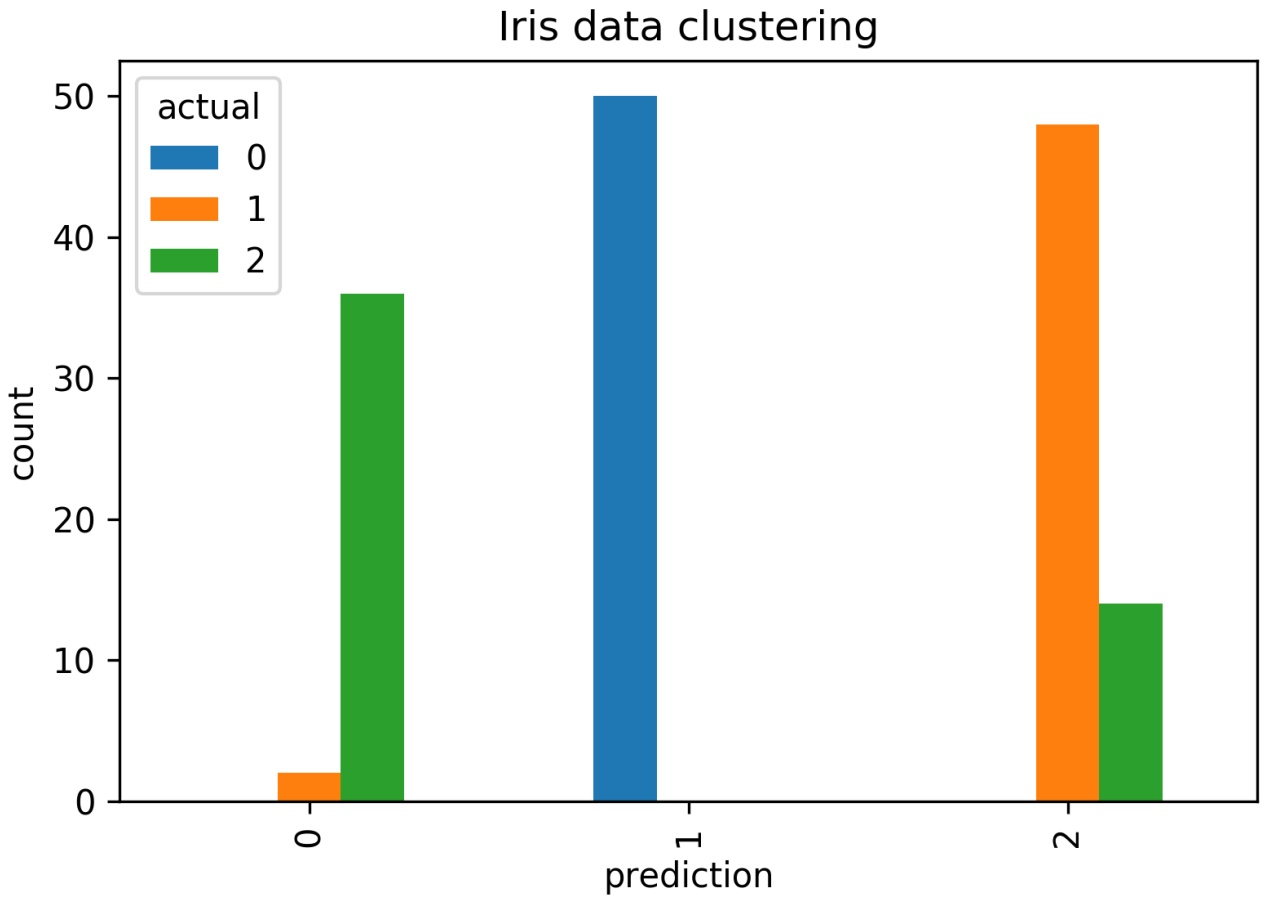
K-scan for the iris data set.



How do you know you did well?

- Inspect per-cluster means of $X^{(i)}$.
  - Do they separate well?
- Inspect known labels.
  - Do we classify correctly?
  - Beware of cluster label mismatch.

Evaluating the iris clusters.



K-Means Clustering

- Start with k *random* cluster means μ_l.
- Given the data x_i, calculate new cluster centers, again using

$$C(x_i) = \underset{l}{\operatorname{argmin}} \|x_i - \mu_l\|$$

$$\mu_l' = \frac{1}{N_l} \sum_{i;C(x_i)=l} x_i$$

- Iterate until the assignment stops changing.

Pros and Cons

- K-Means
  - Fast ($O(N^2)$).
  - Good for sphere shaped clusters.
  - Some randomness.
    - Starts with random center assignments.
    - Outcome might vary from run to run.
- Hierarchical clustering.
  - Slower ($O(n^2)$).
  - Works well with any cluster shape.
  - Can eyeball k from dendrogram.

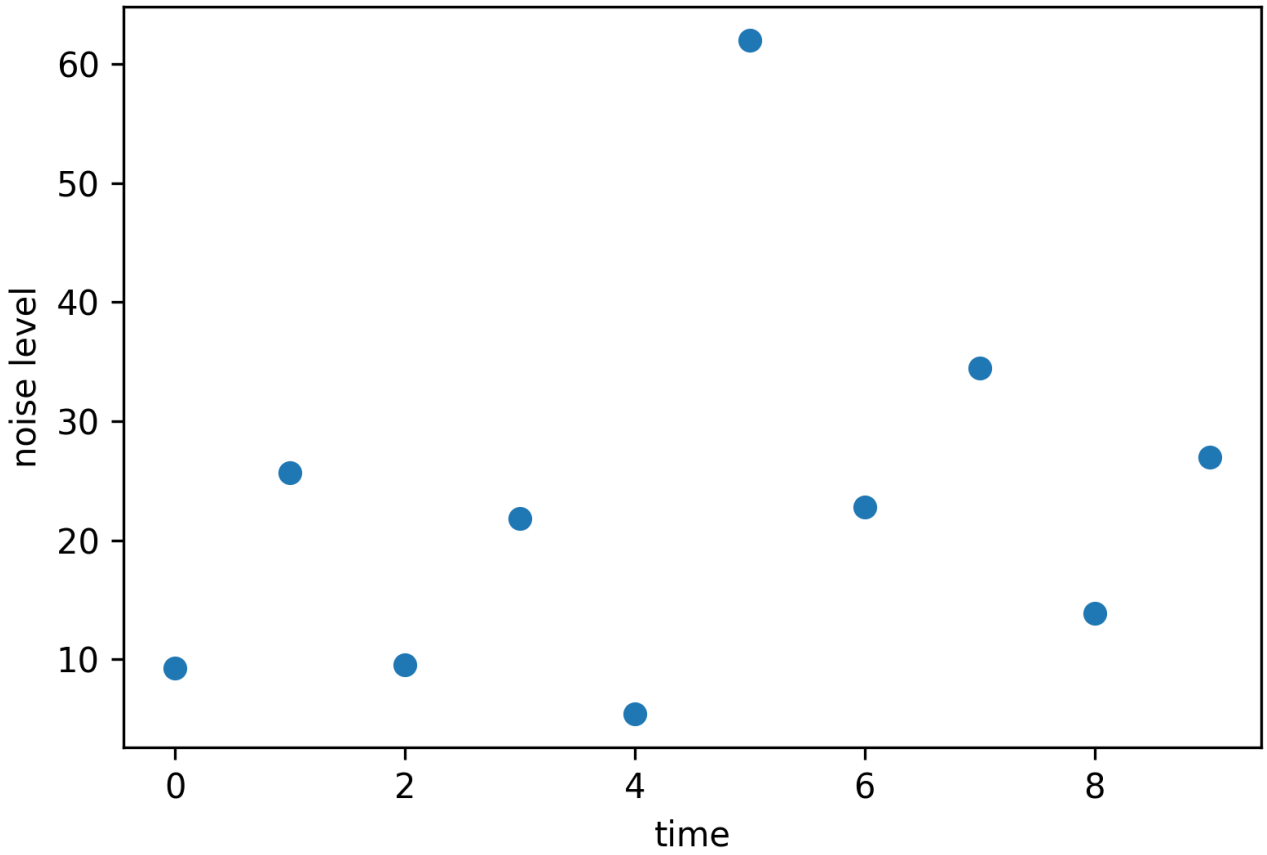
Practical Considerations

- Needs a good metric.
- Observations might need to be standardized.
- Metric chosen might be of interest.
  - 1-Norm (aka Manhattan distance).
  - l-Norm (usually Euclidean).

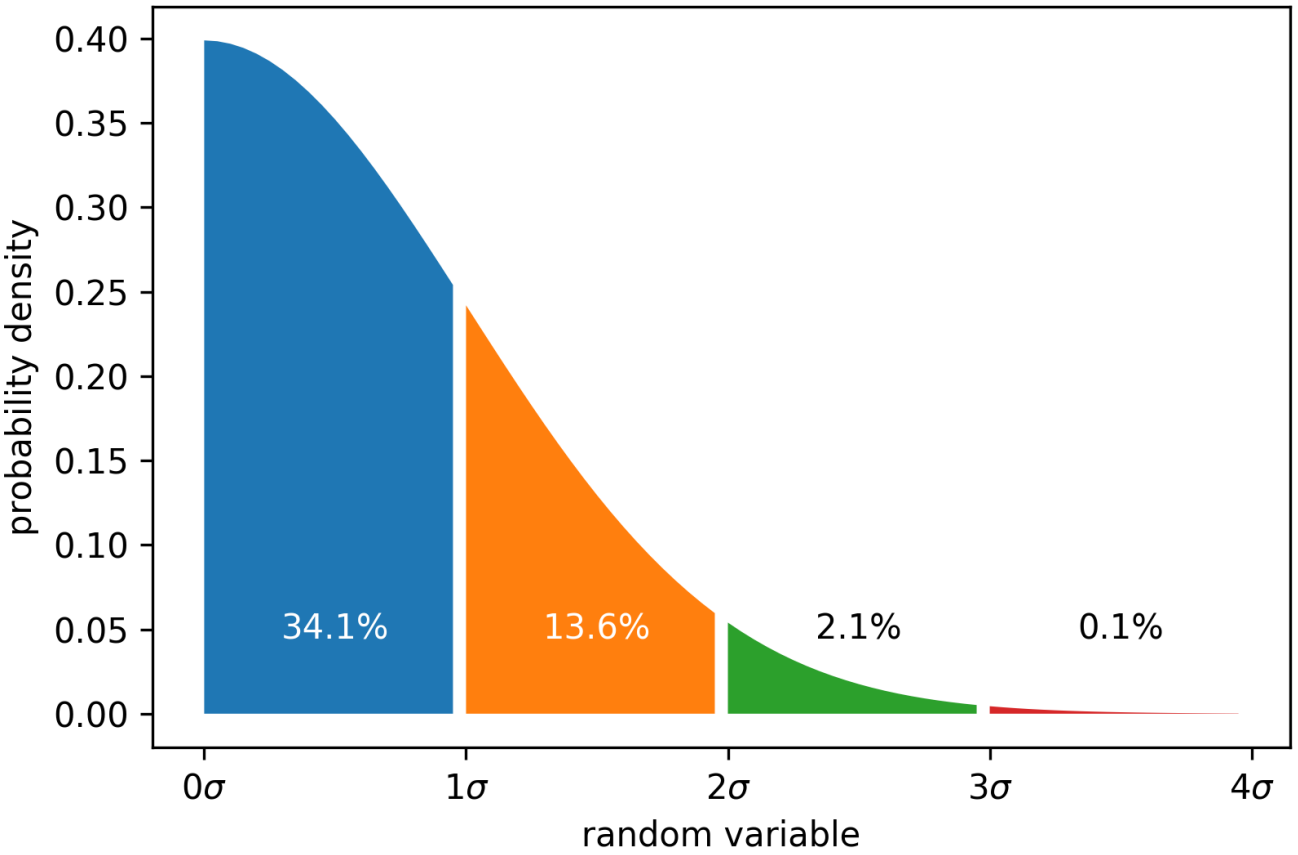
Questions?

Outlier Detection

Example: Predicting machine malfunction.



The Normal Distribution



The z Value

- Assume we have data $x_1, \ldots, x_N$.
- Calculate the mean $\overline{x} = \frac{1}{N} \sum_i x_i$.
- Calculate the standard deviation
  $\sigma = \sqrt{\frac{1}{N} \sum_i (x_i - \overline{x})^2}$
- Calculate the z-value $z_i = (x_i - \overline{x})/\sigma$.
- Flag everything with $z > z_{\mathrm{max}}$ as anomaly.

Chebyshev's inequality

- Valid for a wide variety of *probability distributions*.
- Statement:

$$Pr(|X - \mu| \geq k\sigma) \leq \frac{1}{k^2}$$

- I.e. looking at z values for anomaly detection makes sense.

Sigmas and Probabilities

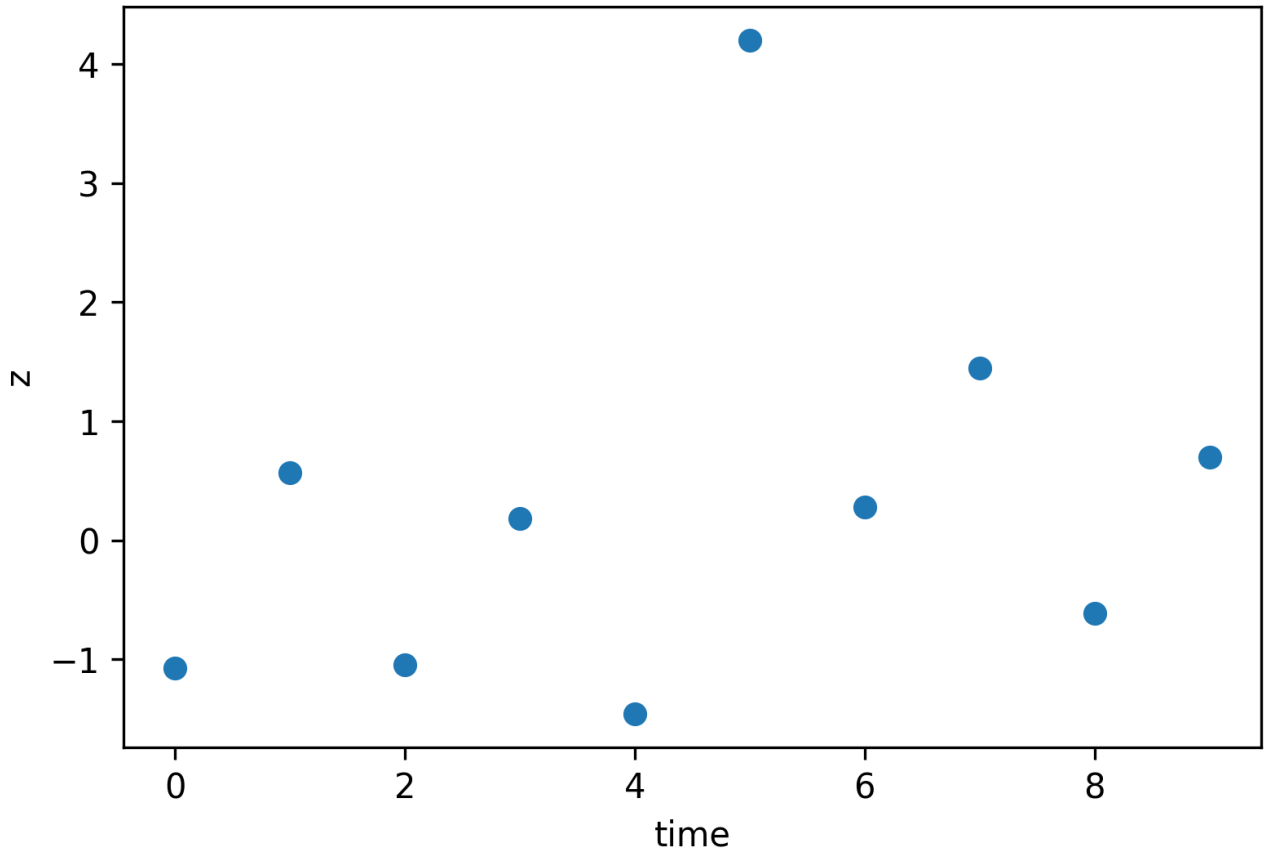
For the normal distribution:

Threshold	Fraction Outside
3σ	1 / 370
4σ	1 / 15 787
5σ	1 / 1 744 278
6σ	1 / 506 797 346

Chebyshev guarantees

Threshold	Percent Outside
3σ	11.1111%
4σ	6.25%
5σ	4%
6σ	2.7778%

Example: Predicting machine malfunction.



The Need For Clustering

- There might be natural variations in data.
  - Weekend vs. weekday spending patterns.
  - Heart rhythms at rest vs. during sport.
- Fit a bunch of (multivariate) normal distributions.
  - How? We generally don't have labels.
- Enter: Cluster methods.