Anomaly Detection

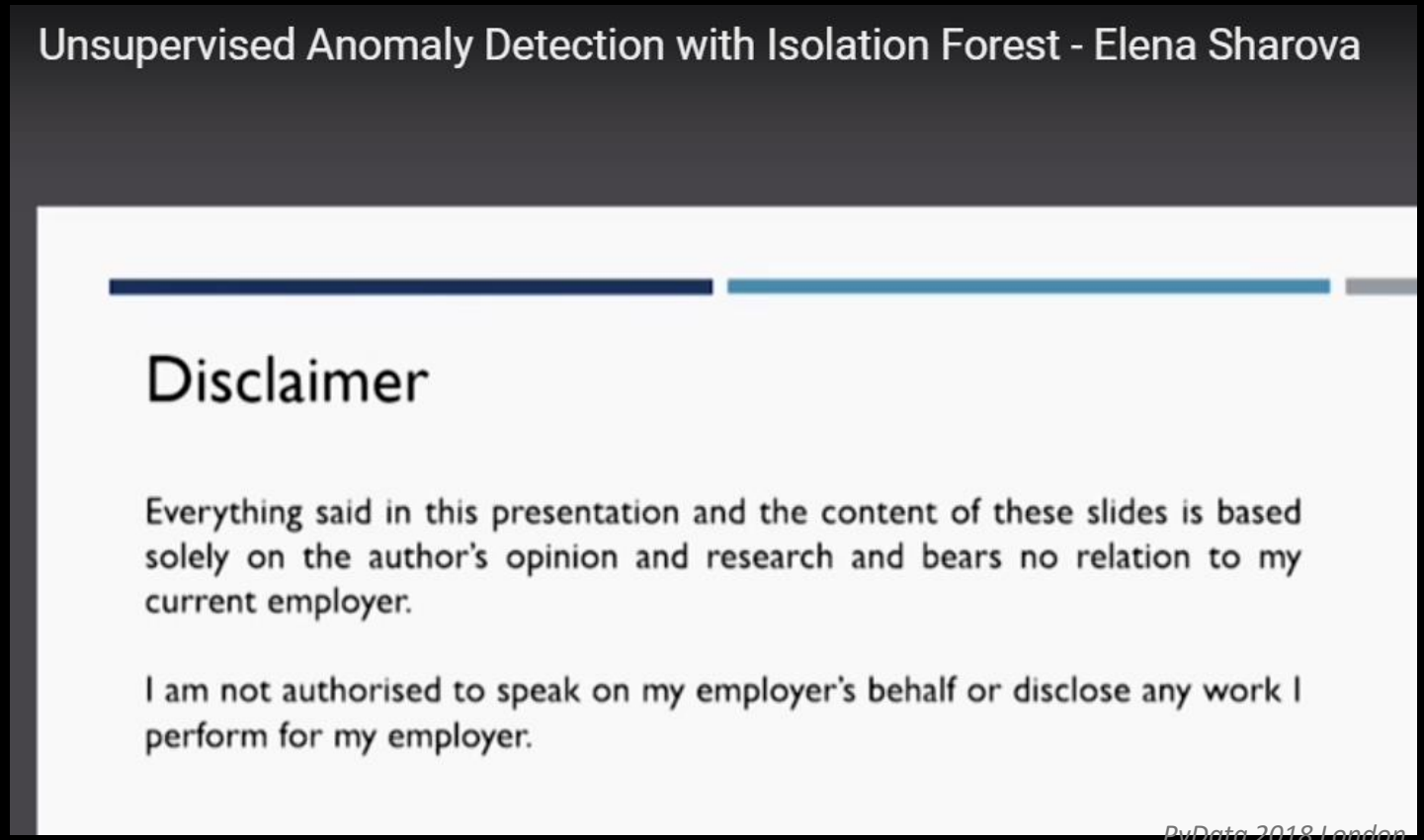
My Goal:

Present a lot of the math behind Anomaly Detection algorithms and some example approaches.

Nothing proprietary in this slide deck, it is all based on open source type concepts.

https://www.youtube.com/watch?v=5p8B2Ikcw-k

Tom is reusing this Disclaimer !
Please read - - - - - - - - - >



Small Print:
I'm just an engineer interested in this topic and I'm sharing my thoughts about it, as well as talking about a tool that seems to help you understand said topic. Nothing presented here is in any form associated with the company I work for currently.
- Tom

Name: Tom Bresee

Email: *tom.bresee@gmail.com*

Github: *https://github.com/tombresee*

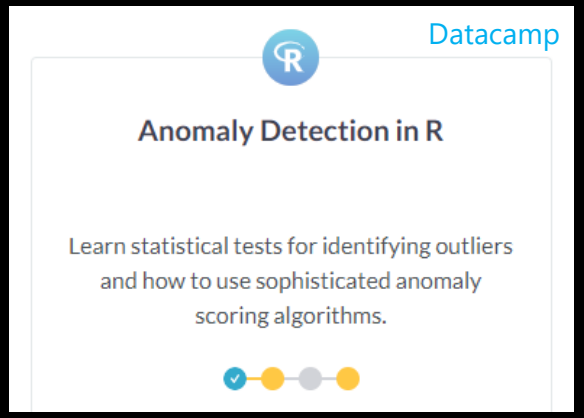
LinkedIn: *https://www.linkedin.com/in/tombresee/*

Things to Check Out

These types of blogs, whitepapers, etc are helpful to understand this topic

Test-Driven Data Analysis (Python TDDA library)

THE NETFLIX TECH BLOG



Uber Engineering

DATADOG

PyData London 2018

anodot

Twitter Engineering

Defining the term ‘Anomaly’…

I really can’t define it,
I just know it when I see it

Elements not behaving
as they normally would ?

Elements (?complicated?) that
aren’t inliers

Rare items, events or observations which raise suspicions by differing significantly from the majority of the data

Deviations from the
expected pattern
of a dataset from the
rest of the values

Everything, depending on
your reference frame or subject
matter expert’s opinion

Values that are unusually distant
from the rest of the values

A datapoint or collection of datapoints that do not follow the same pattern or have the same structure as the rest of the data

Rare.
Distinct.
Doesn’t fit.
Stranger (danger)

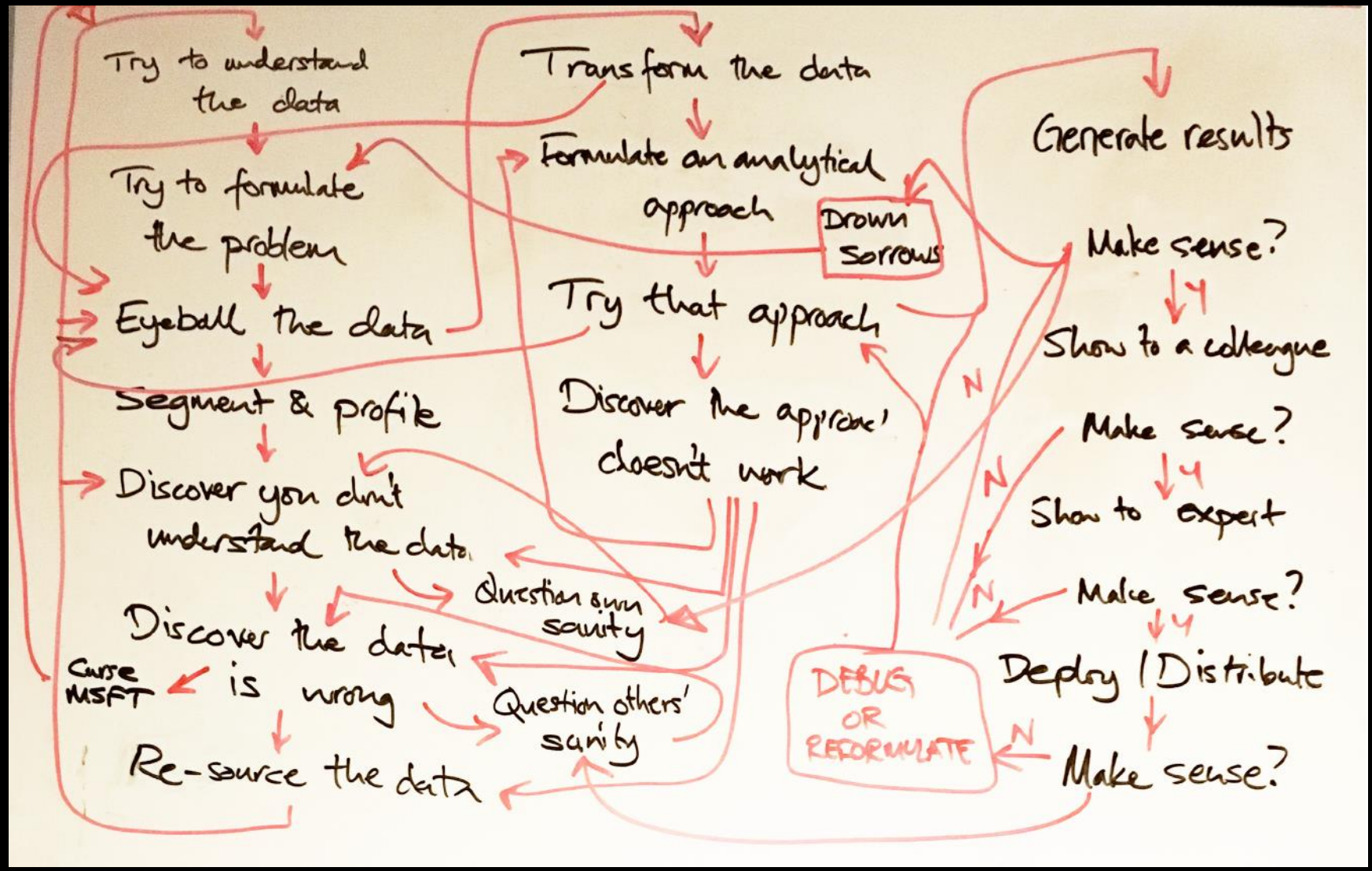
Patterns that do not conform
to expected behavior ?

There is no 100% agreed upon formal definition for an outlier/anomaly

Maybe it just depends…on your…
Subject Matter
Use Case
Application
Problem
Domain

Then let's be careful and try to be precise with our language:
Call it a boxplot outlier, or Grubbs outlier,
or mathematical anomaly, or etc

This stuff is nowhere near easy



Nick Radcliffe - https://www.youtube.com/watch?v=5p8B2Ikcw-k

New Vocabulary Terms ?

- **Highly Statistically Improbable Combinations (H-SIC)** – very low probability events that occur simultaneously in combination (ex: You are in your car in a thunderstorm, scratching off your lottery ticket, and the second you realize you won $100M, lighting also strikes your car)
- **Crazy Ivan** – An anomaly that surfaces, is detected, but then completely disappears for quite some time
- **Chapacabra** – Something bad that you think might be happening in your data, but can't prove. You can neither prove it exists, **nor** can you prove it **doesn't** exist. Has a tendency to lurk.
- **R3** – short for refine, refine, refine (your approach, algorithm, fill in the blank, see last slide…)
- **EKG** – very periodic data that is absolutely not gaussian/normal in any way, highly proprietary like
- **Fly** – Odd annoying observations in data, but harmless
- **Drift** – The concept that what is normal data today may not be normal in the future (not auto related)
- **Kobe dataset** - Severe class imbalance scenario. This is especially imp in AD.
- **Wolf Matrix** – Another name for the confusing Confusion Matrix
- **TML** – a library/philosophy that helps you on your ML journey

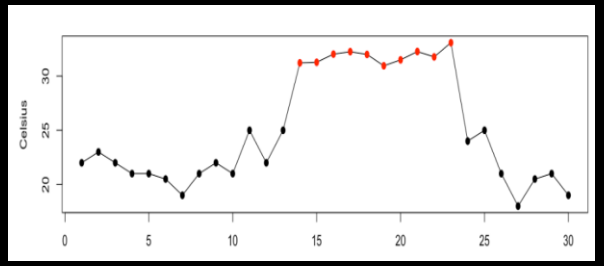
Anomaly Types

Point: Observation that deviates from the 'trend'. A single instance of data is anomalous if it's far off from the rest (unusual when compared to the rest)

Example: It hits 90F in the middle of winter. Think extreme blacksheep.

Collective: Anomalous collection of data instances, unusual when considered together. Conditional/contextual anomalies with respect to the entire dataset, but not each other. i.e. individually not bad, but in the big picture yes its bad.

Example: 10 consecutive high daily temperatures (usually time series related)



Contextual: Observation unusual in a certain context, but NOT in other context.

Example: Something pops during a 3am maintenance window vs middle of day

Location: Global or Local

General Approaches

Viz or Stats

Dev statistical models

Supervised Learning

Classify data as normal or not

But do you have existing labelled data ???

Unsupervised Learning

Clustering

Maybe a bit better of an approach ?

Semi-Supervised Learning

Or a combination maybe ?

Simulating anomalies is tricky

Statistical Methods

- Z-ish score
- Median-ish score
- 3-sigma
- Grubbs
- EST

Distance-Based

- K-NN
- Regression hyperplane dist
- K-Means

Density-Based

- DBSCAN
- LOF (local outlier factor)

Spatial Proximity

Remember: In general, anomalies should not be very prevalent . . .I think we can agree on that . . .

Visualization Assessment Approaches

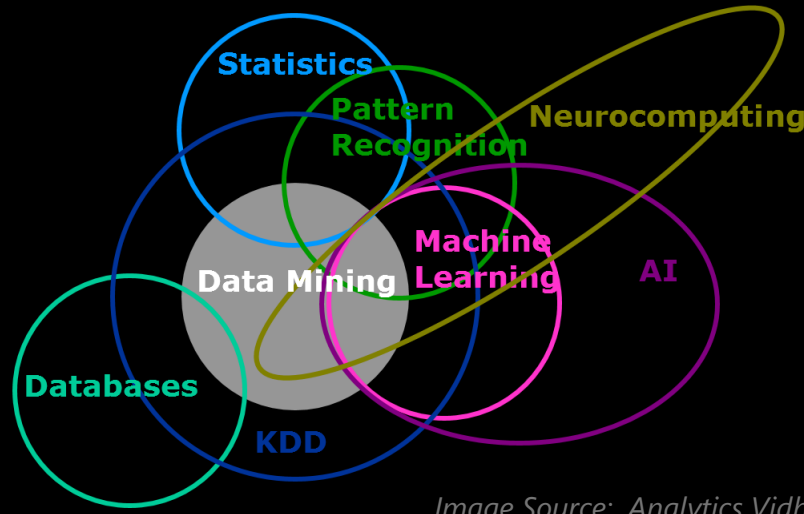
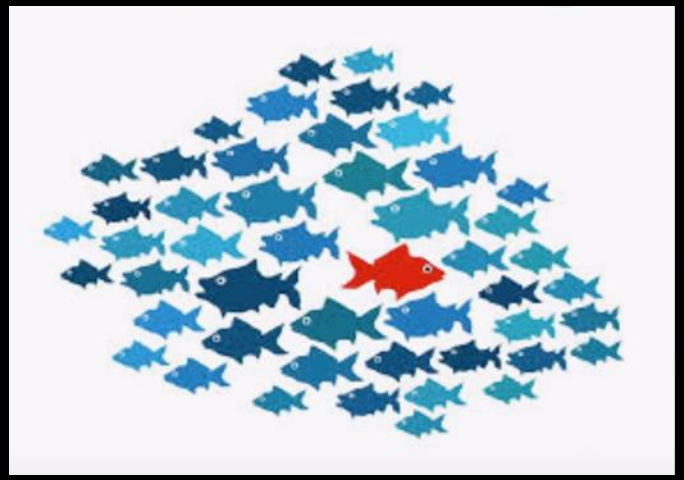


Image Source: Analytics Vidhya

Visuality

- HUMANINT
  - Properly Graph
  - Eyeball
  - SME Eyeball



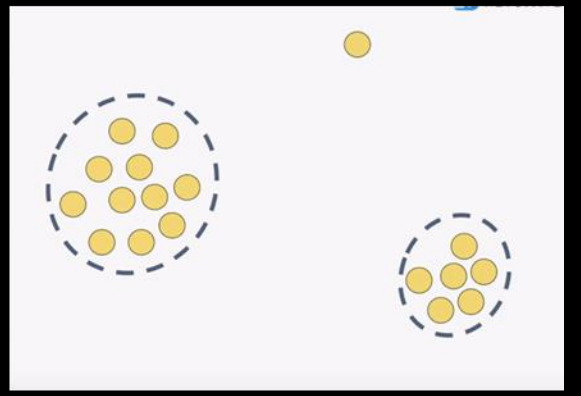
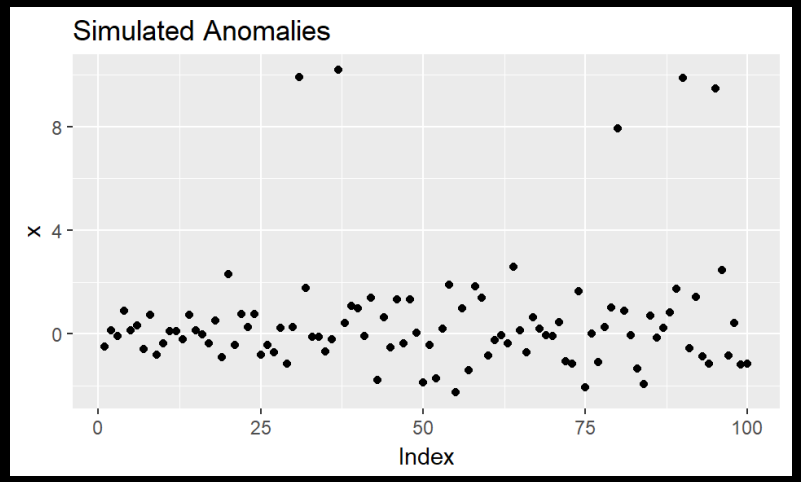


Image Source: Numenta



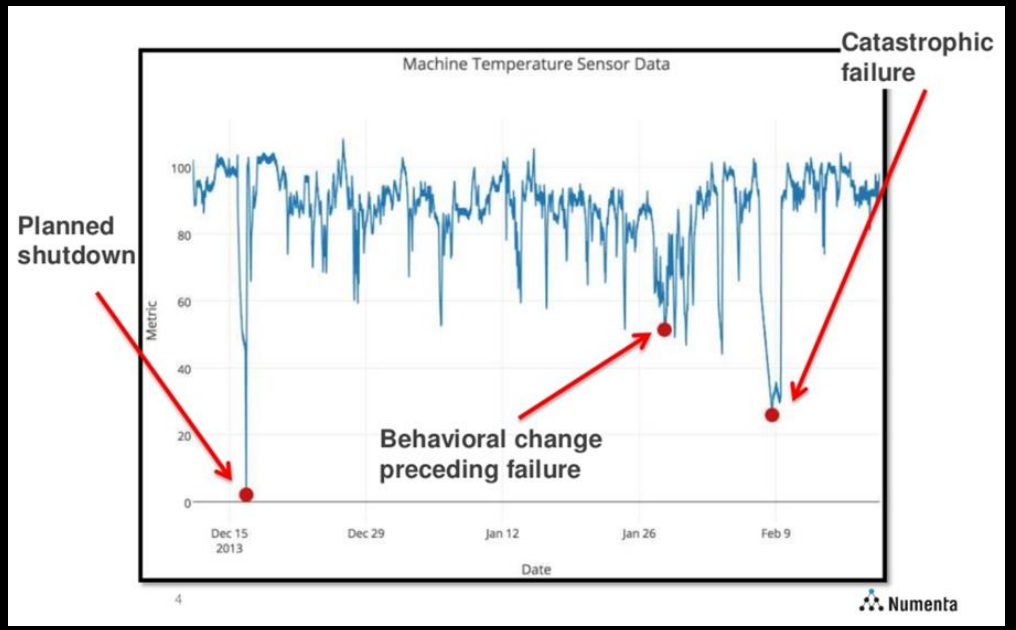


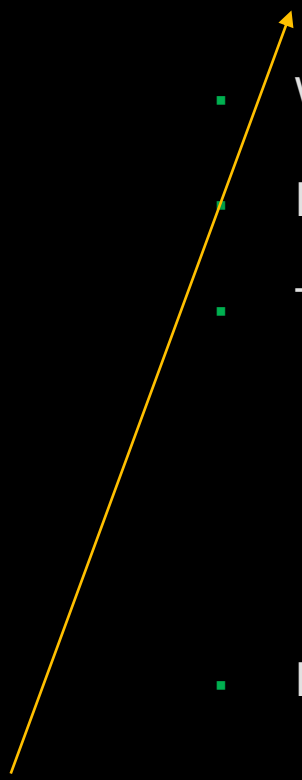
Image Source: Numenta

Dashboards

- Gathering the appropriate **metrics** is absolutely essential
- Must have solid baselines
- TCA alert (if you cross value X a certain number of times within Y timeframe, then alert)
- What happens when things start increasing/growing or 'drift' ?
- False Alarms are taxing on people (Klaxon)
- Tableau Effect:

  Don't worry, we have things under control because we have all of the raw data in a dashboard, and its pleasant to gaze upon because of its aesthetic beauty

  Moreover: I look smart because the graphs look like something from a sci-fi movie
- Eventually hit some scaling limits, BUT not bad for most metrics you are watching



Would be nice to have a 'dynamic' envelope around data, so we could re-use threshold crossing concept (not static) . . . See time series slides …

Where exactly is the outlier ?

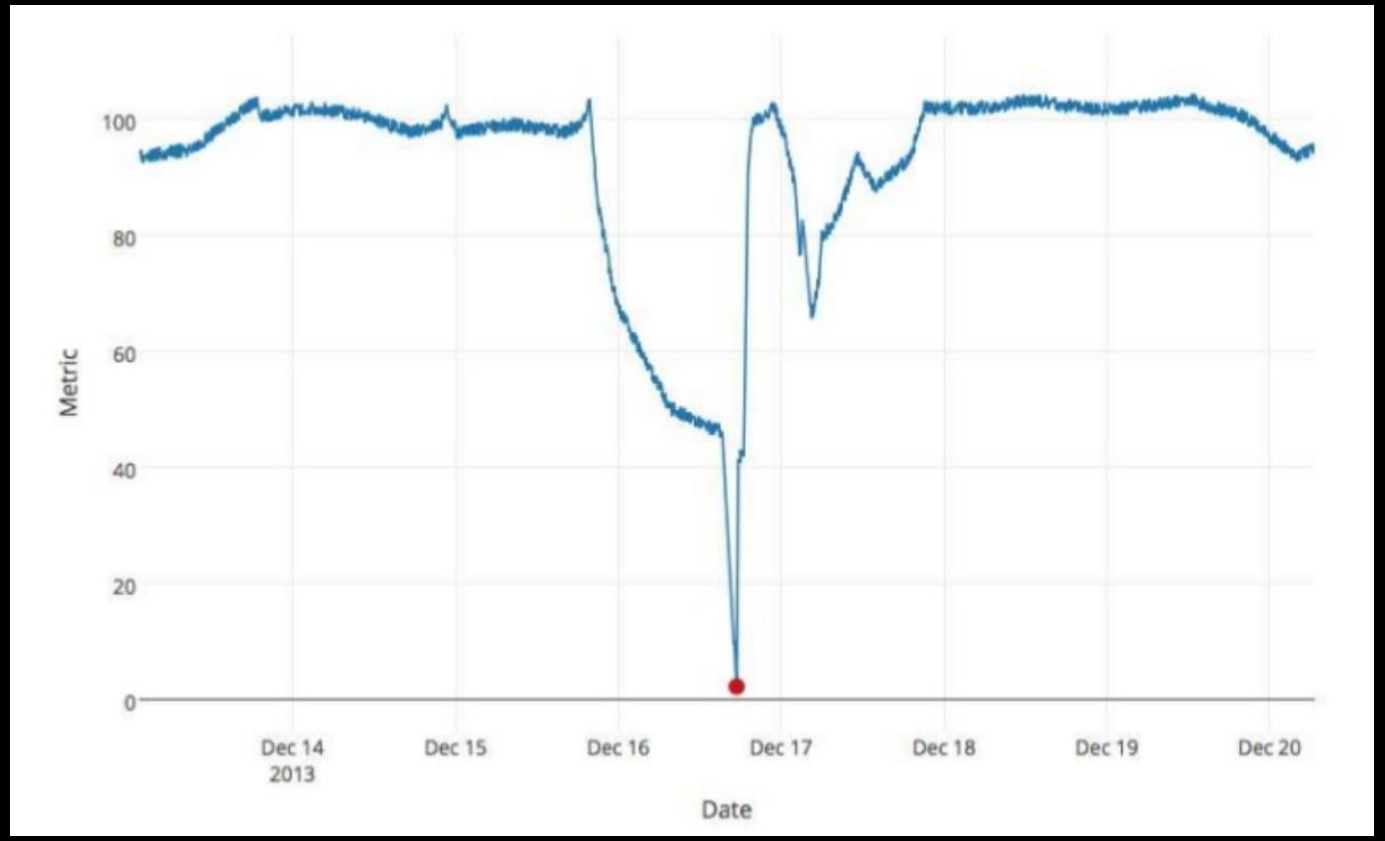


Image Source: Anomaly Detection - Elizabeth (Betsy) Nichols
https://www.youtube.com/watch?v=5vrY4RbeWkM

A person can't just eyeball graphs. Doesn't scale and its unreliable.

Pairplot ?

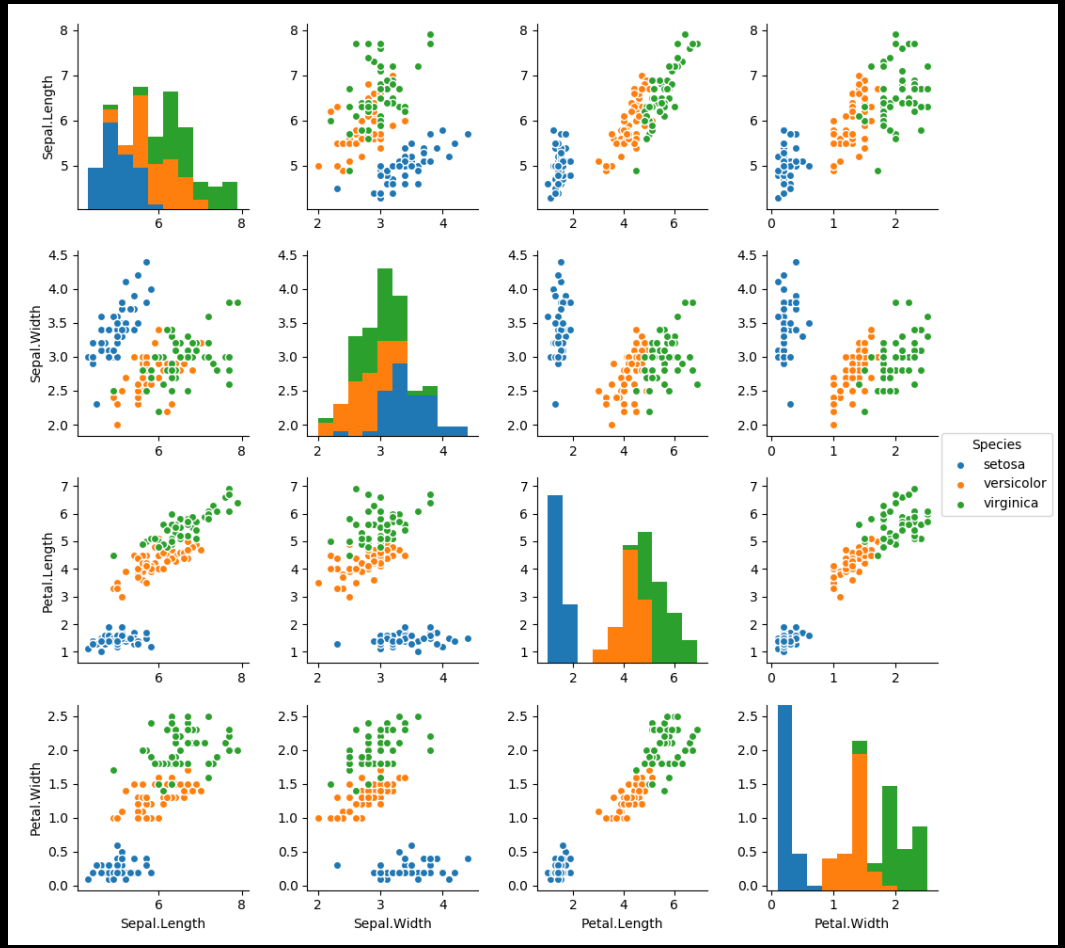


Image Source: Scikit Learn Iris Dataset

Metric A vs B type comparisons can be helpful…

Apophenia: the spontaneous perception of connections and meaningfulness of unrelated phenomena.

Patternicity: the tendency to find meaningful patterns in meaningless noise

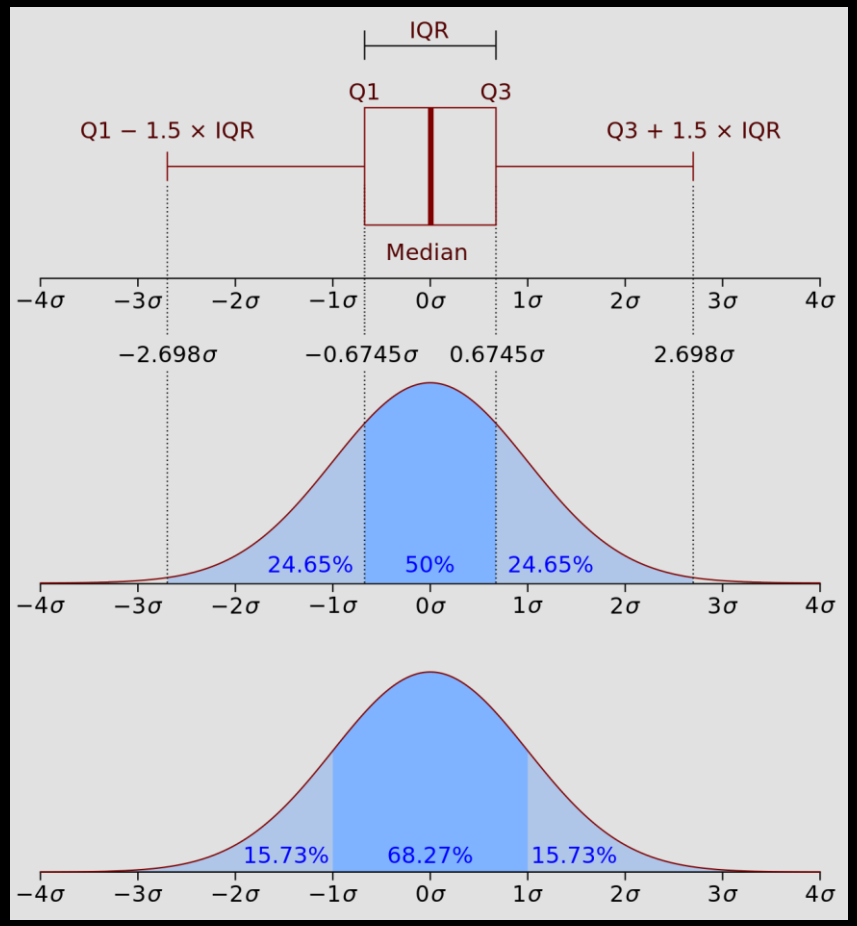
Pareidolia: a psychological phenomenon in which the mind responds to a stimulus, usually an image, by perceiving a familiar pattern where none exists.

Agenticity

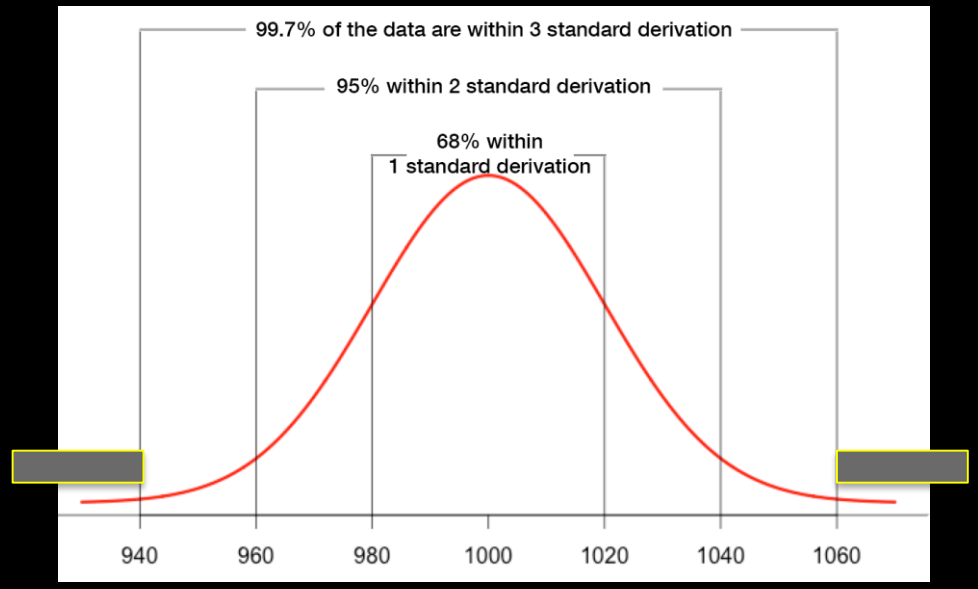
Wednesday, March 1, 1950
Beatrice, Nebraska

Statistical Approaches

Statistics



Source: Wikipedia - Normal Distribution



Source: Wikipedia - Normal Distribution

Think extremities and Z-score…

Could couple this with windowing for RT

But aren't mean/SD sensitive ???

Go turn in a late assignment

Boxplots/IQR

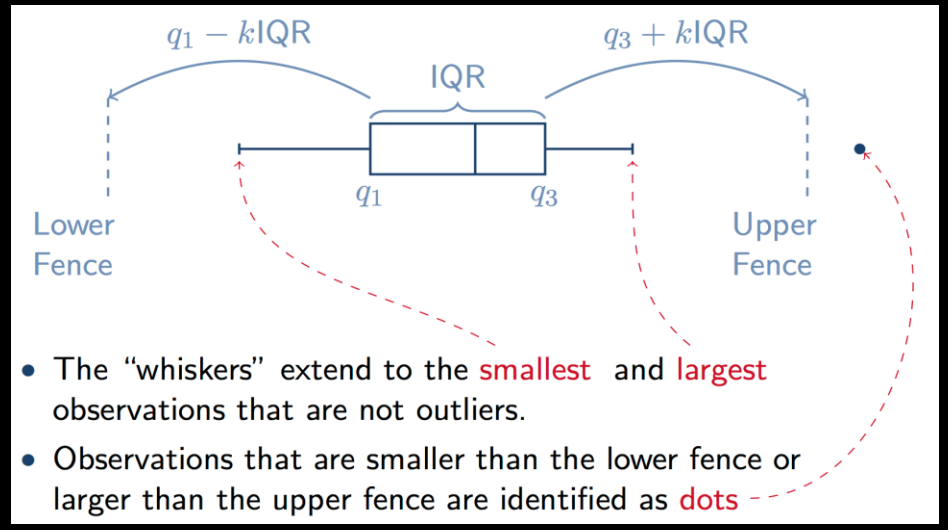
matplotlib.pyplot.boxplot

matplotlib.pyplot.boxplot(*x*, *notch=None*, *sym=None*, *vert=None*, *whis=None*, *positions=None*, *widths=None*, *patch_artist=None*, *bootstrap=None*, *usermedians=None*, *conf_intervals=None*, *meanline=None*, *showmeans=None*, *showcaps=None*, *showbox=None*, *showfliers=None*, *boxprops=None*, *labels=None*, *flierprops=None*, *medianprops=None*, *meanprops=None*, *capprops=None*, *whiskerprops=None*, *manage_xticks=True*, *autorange=False*, *zorder=None*, *, *data=None*) [source]

whis : float, sequence, or string (default = 1.5)

As a float, determines the reach of the whiskers to the beyond the first and third quartiles. In other words, where IQR is the interquartile range (`Q3-Q1`), the upper whisker will extend to last datum less than `Q3 + whis*IQR`). Similarly, the lower whisker will extend to the first datum greater than `Q1 - whis*IQR`. Beyond the whiskers, data are considered outliers and are plotted as individual points. Set this to an

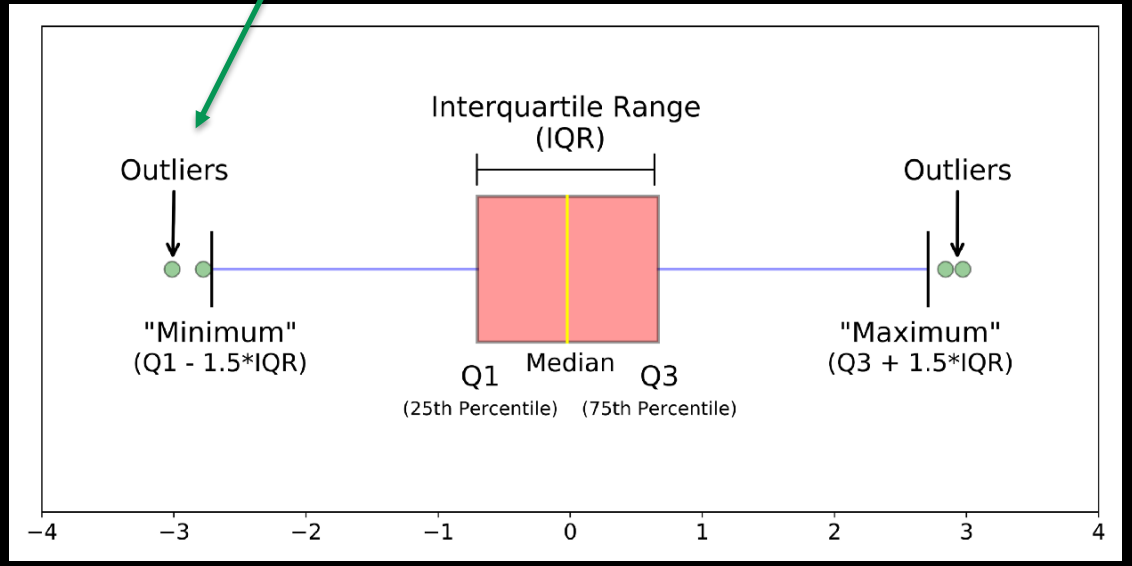
Term: "Boxplot Anomaly" ?



Images Source: Matplotlib Library

- **Boxplot**
  - Use boxplot to visualize ?
  - Several graphical techniques can, and should, be used to detect outliers.
  - Helpful for identifying point anomalies
  - z = (x - u) / s

Once we have identified our anomaly, we are done right ?



Source: Wikipedia Boxplot

```
summary(temperature)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  18.00   20.45   22.45   22.30   22.98   30.00
```

Source: R

Grubbs tests (for one outlier)

- Also known as the **maximum normalized residual test** or extreme studentized deviate test, is a statistical test used to detect outliers in a **univariate** data set
- Statistical Test to decide if a single (extreme) point is an outlier (point furthest from the mean, hi or low)
- 'Maximum Normed Residual Test'
- <u>Assumptions:</u>
  - Data comes from a **normally** distributed population (check normality first, look for the L…)
  - Check normality with histogram, you are looking for symmetry and shape
  - Maybe good idea to call it the Grubbs Outlier ?

The Grubbs' test statistic is defined as follows:

$$C = \frac{\max_t \mid x_t - \overline{x} \mid}{s} \qquad (3)$$

where, $\overline{x}$ and s denote the mean and variance of the time series X. For the two-sided test, the hypothesis of no outliers is rejected at significance level α if

$$C > \frac{(N-1)}{\sqrt{N}} \sqrt{\frac{(t_{\alpha/(2N),N-2})^2}{N-2+(t_{\alpha/(2N),N-2})^2}} \qquad (4)$$

Source: Wikipedia – Grubbs Test

Anomaly_Detection / R_examples / Grubbs_Test_with_R.ipynb

```
grubbs.test(x)
```

```
        Grubbs test for one outlier

data:  x
G = 2.40630, U = 0.42576, p-value = 0.0258
alternative hypothesis: lowest value 4.02 is an outlier
```

Source: Tom

Median (not mean!) Absolute Deviation (MAD)

- *Robust/Resilient* measure of the variability of a univariate sample of quantitative data
- Measure of statistical dispersion (much better than SD for our apps)
- *MAD: quasi robust measure of the variability or spread of the data*

Ultimately, we are looking for a solid better measure of our variability/extremes not so thrown off by large values…

Very good -> Source: OSCON 2016 Homin Lee (Datadog)

$$MAD(D) = median(\{ |d_i - median(D)| \})$$

$$MAD(D) = median(\{ |d_i - median(D)| \})$$

$$D = \{ 1, 2, 3, 4, 5, 6, 100 \}$$

$$median = 4$$

$$deviations = \{ -3, -2, -1, 0, 1, 2, 96 \}$$

$$abs\ deviations = \{ 0, 1, 1, 2, 2, 3, 96 \}$$

$$MAD = 2$$

https://www.youtube.com/watch?v=mG4ZpEhRKHA

*Starting with the residual (deviations) from the data's median, the MAD is the **median** of their absolute values*

MAD

- I specifically am not saying Mean, I'm saying Median
- Hard-core Robusticity
- Very few parameters ~ simplicity
- Standardize features scale

Median stays solid…despite how much I mess with values

Dataset = { 1 3 5 7 9 50 91 93 95 97 99 }

Dataset = { 1 3 5 7 9 50 91 93 95 97 150 }

Dataset = { 1 3 5 7 9 50 91 93 95 97 8,456,234,121 }

In statistics, the **median absolute deviation** (**MAD**) is a robust measure of the variability of a univariate sample of quantitative data. It can also refer to the population parameter that is estimated by the MAD calculated from a sample.

For a univariate data set X_1, X_2, ..., X_n, the MAD is defined as the median of the absolute deviations from the data's median $\tilde{X} = \text{median}(X)$:

$$\text{MAD} = \text{median}(|X_i - \tilde{X}|)$$

that is, starting with the residuals (deviations) from the data's median, the MAD is the median of their absolute values.

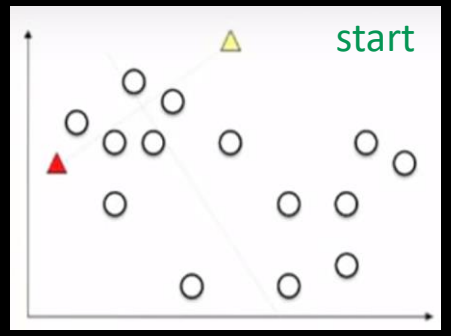
Source: 'Median Absolute Deviation' - Wikipedia

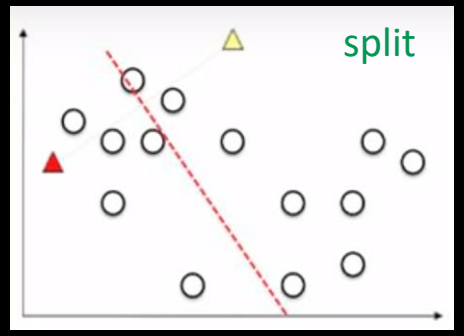
Unsupervised Machine Learning

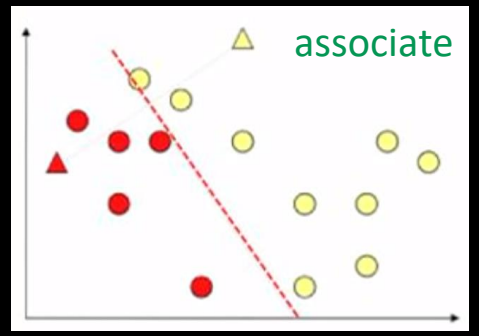
k-Means

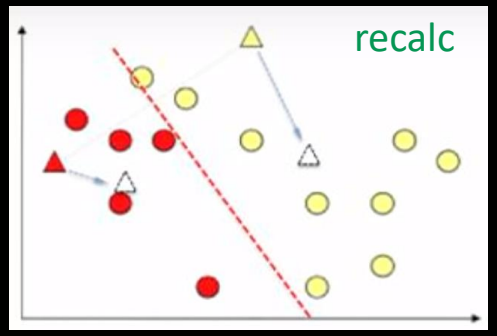
- Need to know the number of clusters (k) you want to find
- Categorical/Ordinal ? No (how do you take the average of cat and dog ?
- Scales well to large number of samples
- All-purpose wrench

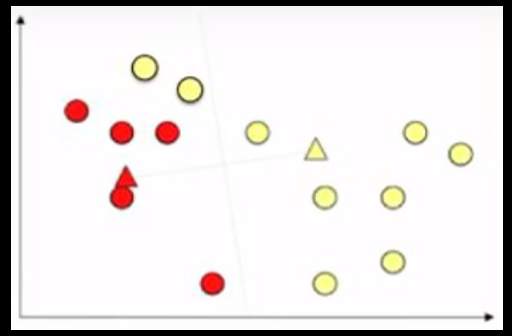
See full lecture: Victor Lavrenko - http://bit.ly/K-means

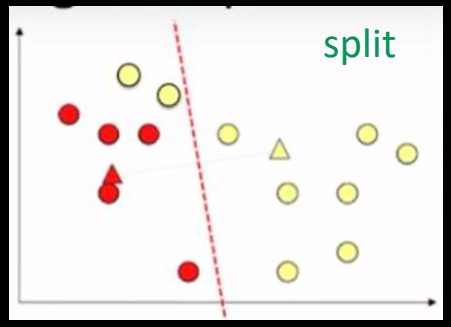


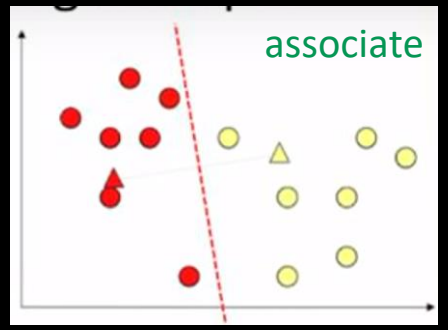


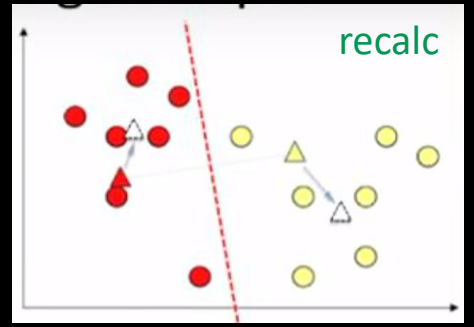


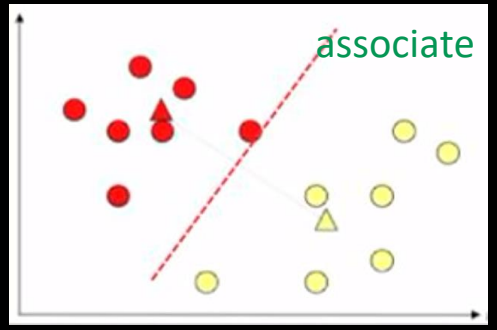








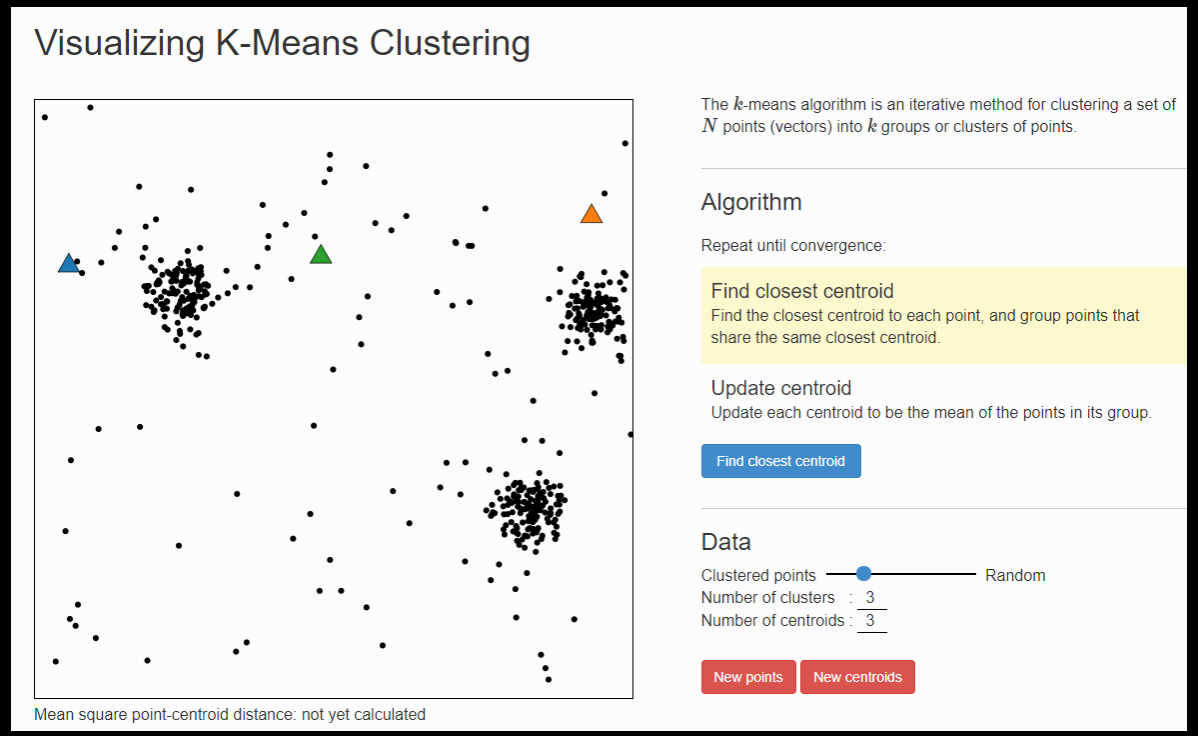




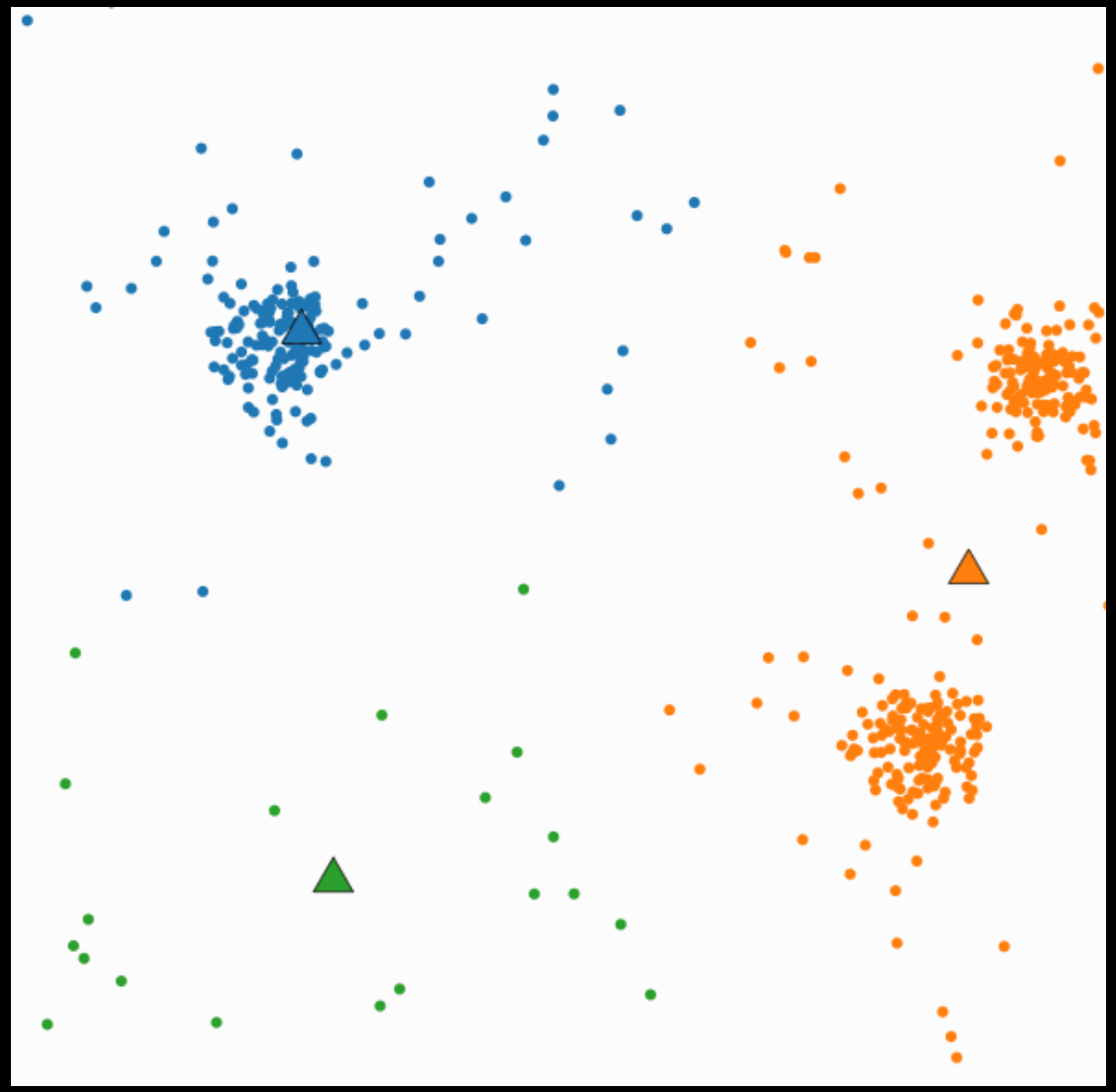
Converged !

k-Means

Go here and mess around with various k input values:
http://web.stanford.edu/class/ee103/visualizations/kmeans/kmeans.html



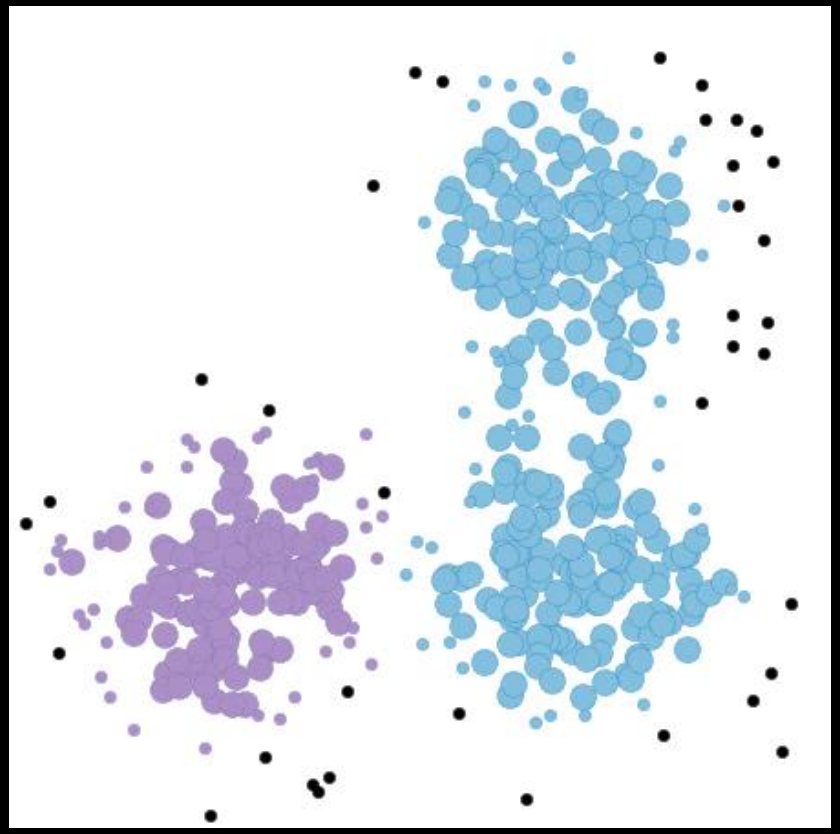
I have converged !



Another good visualization link for k-means:
https://www.naftaliharris.com/blog/visualizing-k-means-clustering/

DBSCAN (`96)

Lets cluster things that are like each other, **and** I don't want to guess on the number of clusters



Black dots = outliers/anomalies

Big circles = core points

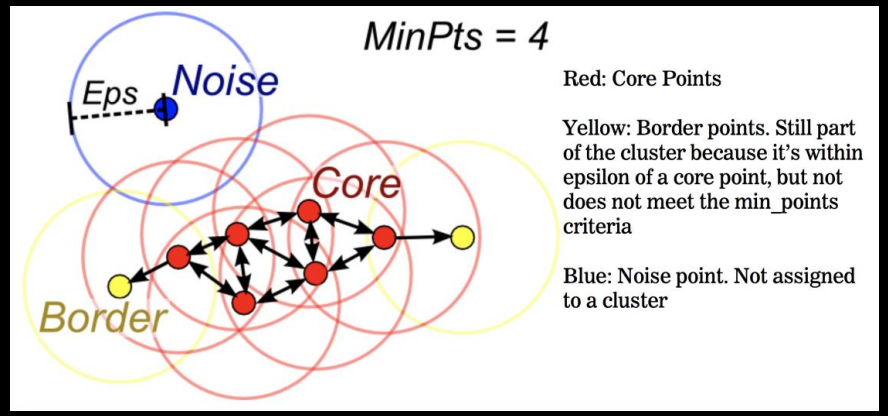
Small circles = border points

Check this out, it's very very good for understanding how this stuff works:
https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/

Image Source: OSCON 2016 Homin Lee (Datadog)

DBSCAN

The How



DBSCAN: Unsupervised learning method utilized in model building and machine learning algorithms

Density-based clustering algorithm: Seeking areas in the data that have a high density of observations, versus areas of the data that are not very dense with observations.

minPts - the minimum number of data points needed to determine a single cluster. Maybe call it the **min**_buddy value ? Overall specifies how many neighbors a point should have, to thus be included into a cluster.

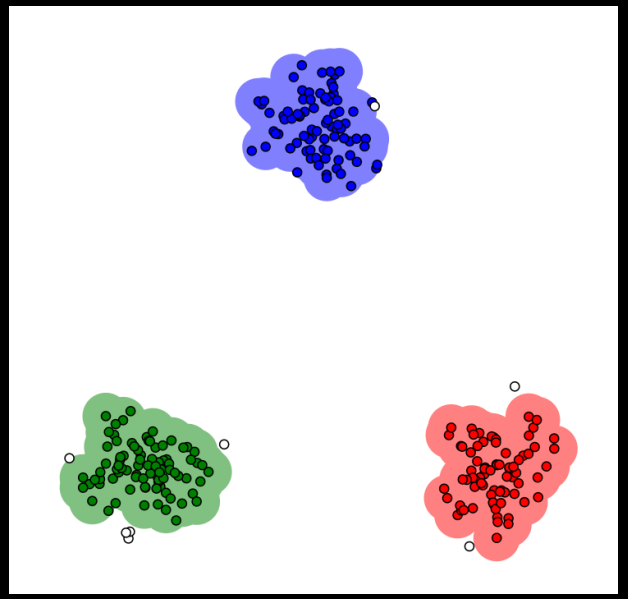
epsilon - specifies how close points should be to each other to be considered a part of a cluster. Maybe call it ***p****ersonal-****s****pace* Radius ?

Effectively anomalies are like *'misunderstood loners'*

Anti-blob support: **Can** sort data into clusters of *varying* shapes/sizes.

Diff approach than k-means, and I like it

In contrast to k-means, which modeled clusters as sets of points near to their center (centroid), density-based approaches like DBSCAN model clusters as high-density **clumps** of points

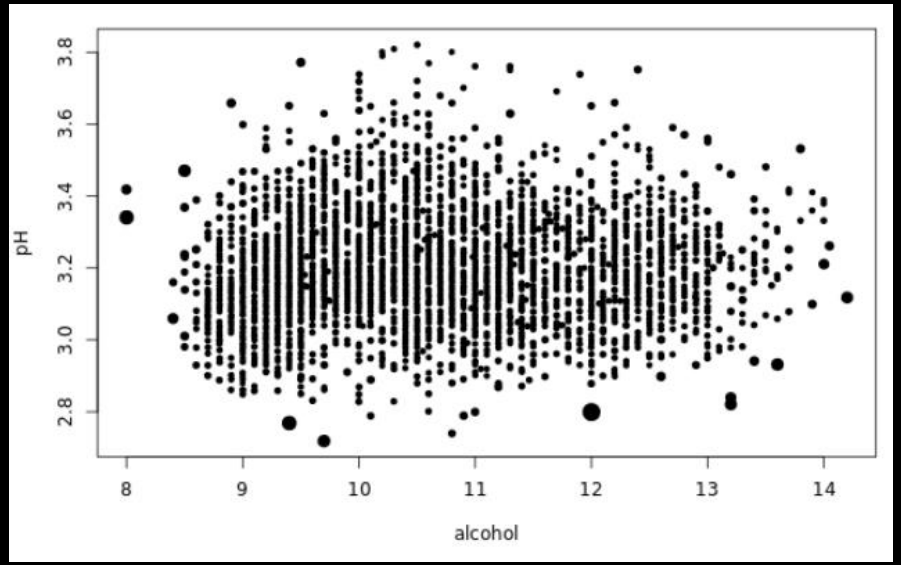


ToT.DE.

Image Source Ref:

Local Outlier Factor (LOF)

- Unsupervised outlier detection method which computes the local **density** deviation of a given data point with respect to its neighbors. It considers as outlier samples that have a substantially lower density than their neighbors.
- Uses density (not distance) to calculate outlier '*score*'
- Molecular approach ! ! ! D = m / V
- In general, a universal alg for finding global and local is not easy
- Avg density around my neighbors / density around me (so it's a ratio)
- LOF >1 more likely to be A, and <1 less likely to be an A
- Need k number of neighbors
- Doesn't rely on distance, so picks up local outliers quite well



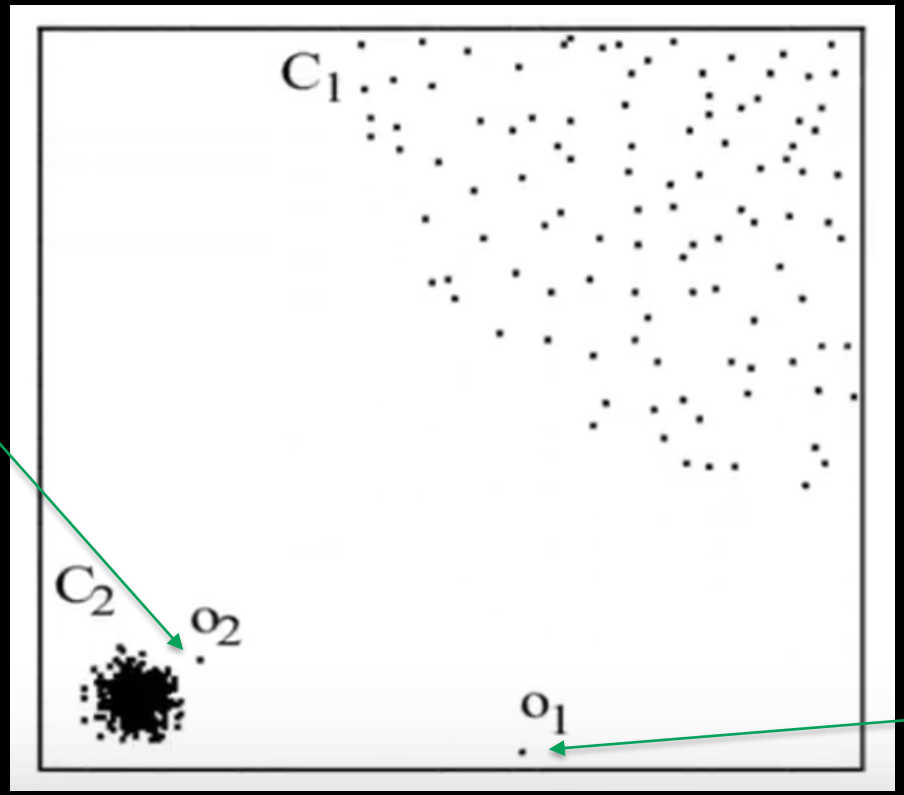
Goal: Do a better job of finding local outliers

Quantify the relative density about a particular data point. Anomaly should be more isolated. Examines neighborhood.

M = mass (number of points) V = K-distance or vol of circle. Divide !

Local Outlier: Only far from nearby points in one area
Solutions based on abs density cannot detect local outliers !
LOF works on global and local outliers…

We need to make sure we don't forget about 'local' type outliers



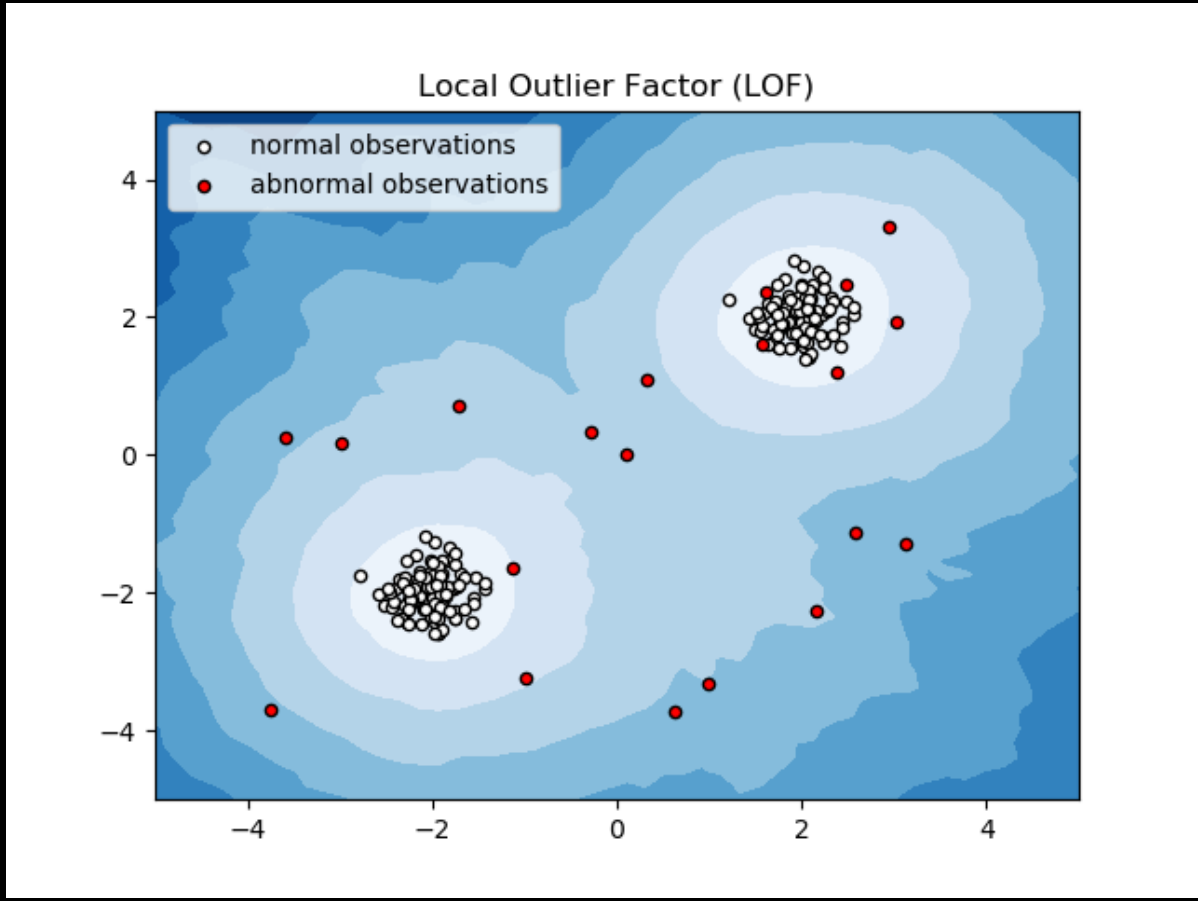
This is easy, its obviously a global outlier

LOF = indicator for a **degree** of local **outlier-ness**…

LOF(k) ~ 1 means **Similar density as neighbors**

LOF(k) < 1 means **Higher density than neighbors (Inlier)**

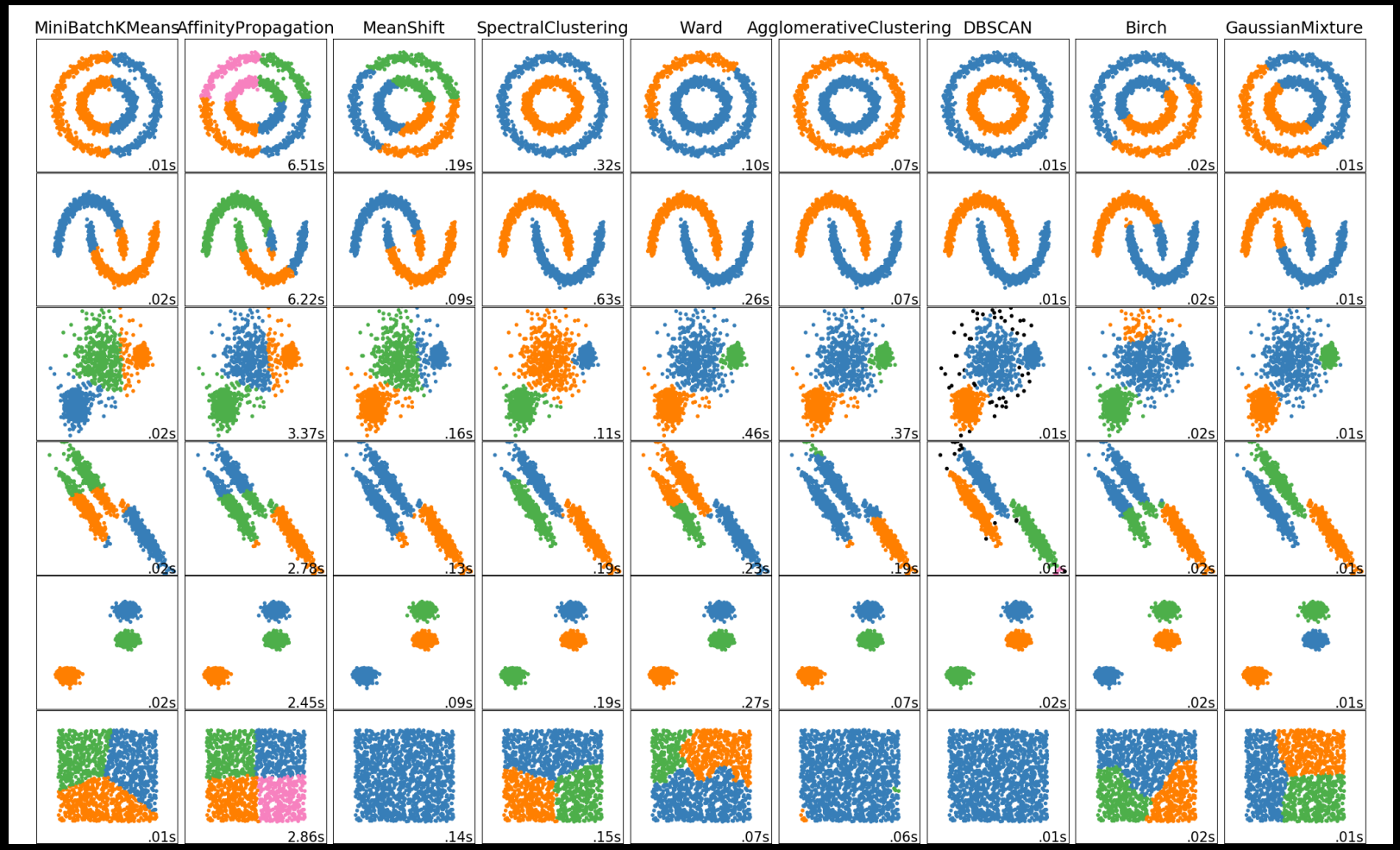
LOF(k) > 1 means **Lower density than neighbors (Outlier)**



Source: Scikit-Learn LOF

SpectralClustering, etc

Reference: Clustering



Source: Scikit Learn - Clustering

Pre-Processing

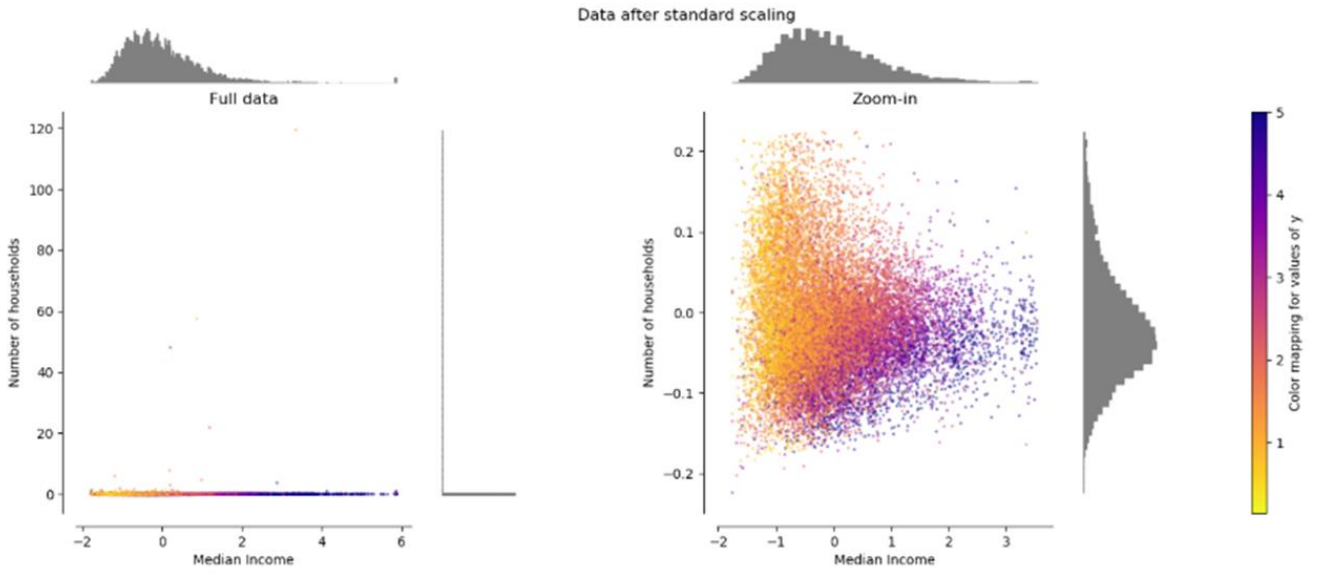
Zeroize…

StandardScaler

`StandardScaler` removes the mean and scales the data to unit variance. However, the outliers have an influence when computing the empirical mean and standard deviation which shrink the range of the feature values as shown in the left figure below. Note in particular that because the outliers on each feature have different magnitudes, the spread of the transformed data on each feature is very different: most of the data lie in the [-2, 4] range for the transformed median income feature while the same data is squeezed in the smaller [-0.2, 0.2] range for the transformed number of households.

`StandardScaler` therefore cannot guarantee balanced feature scales in the presence of outliers.

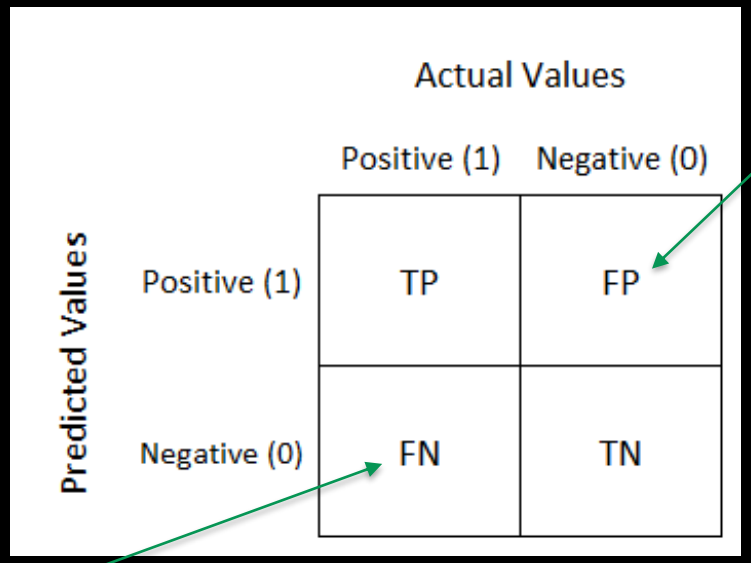
```
make_plot(1)
```



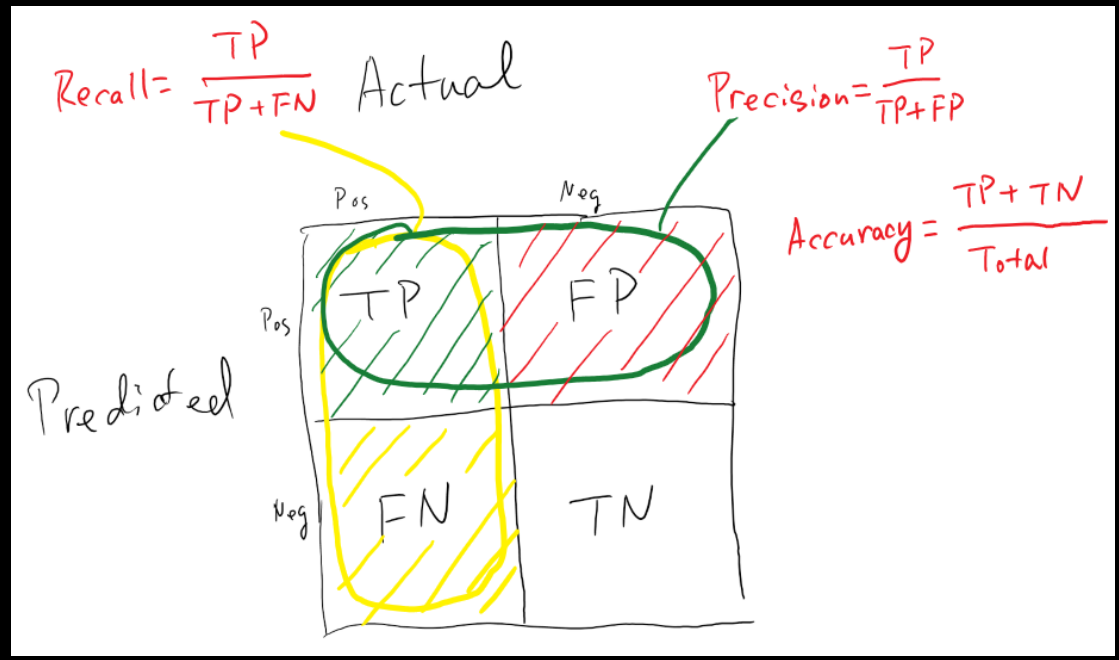
Source: Scikit Learn StandardScaler

Maybe call it Wolf Matrix ?

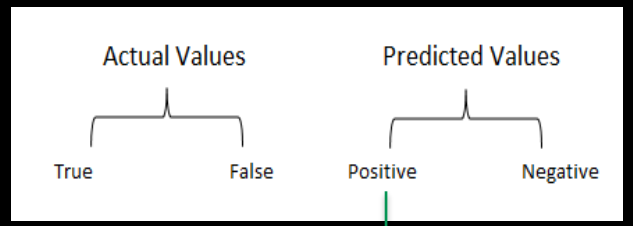
Confusion Matrix



Type 1 Error:
I cried (predicted) wolf (anomaly)
but in **actual**ity there is no wolf (anomaly)



Type 2 Error:
I look
incoMpitent



Anomaly Present
(I think)

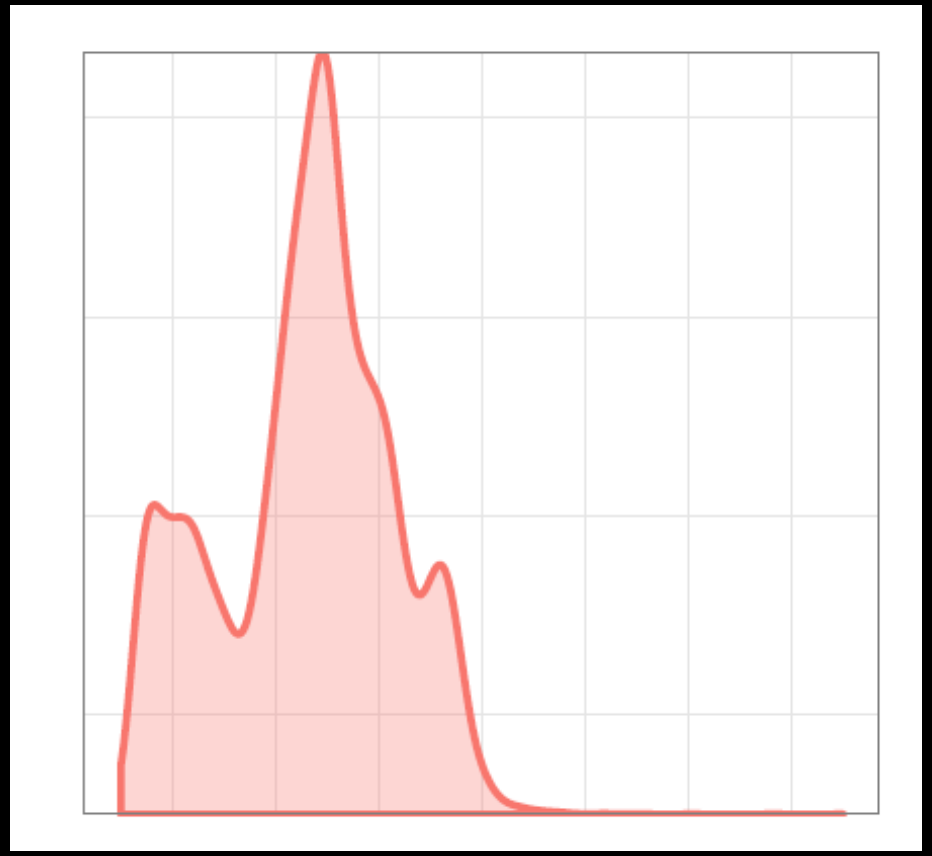
Images Source: Wikipedia Confusion Matrix

Time Series

Things are about to get a lot more complex…

Cog ?

Does this look like a Gaussian distribution to you ?



Ummmm, now what ?

Time Series

- Sequence time series data: set of data in a sequence
- A term called ‘Seasonality’ is introduced
- Local/Global combinations of anomalies are difficult to detect
- Network Anomaly Detection becomes a thing
- AD too sensitive: Get ready for a million screaming eagles

    Too sensitive and you get false positives; too robust and you miss them

- Robust:

    Approach: less influenced by one-off spikes

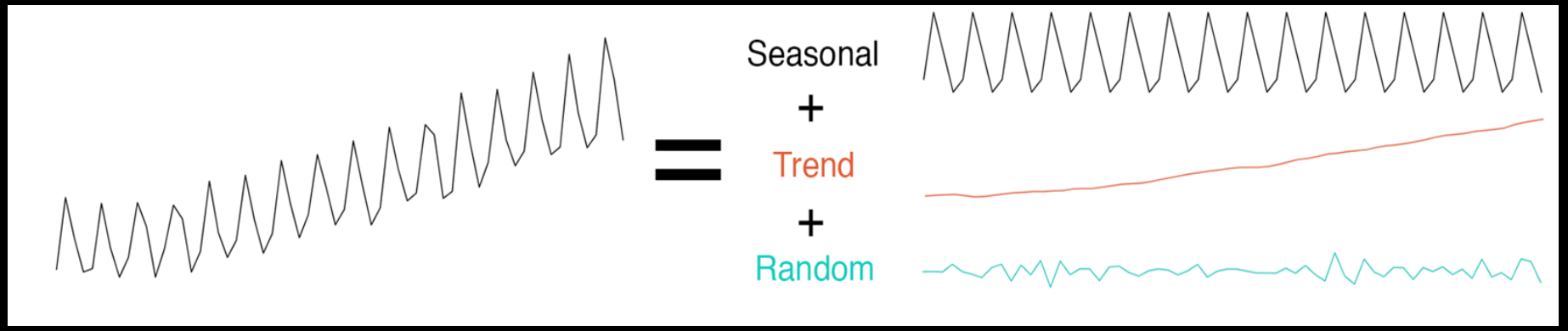
    Alg: performance is stable after adding some noise to the dataset

Time Series

- Need windows and need envelopes
- What is your periodicity ? It's the first thing I care about
- TCA isn't dead !
  - Just kinda automatically setting thresholds
  - Phat Envelopes
- Need instant historical context
- **Power of last week** + robust statistics

Decompose into components !

a. What is your overall trend/rate ?
b. What is your seasonality/periodicity ?
c. Lets get our arms around the random component



Source: Twitter

Twitter (2015)

Search for the machine learning part . . . You won't find it

Automatic Anomaly Detection in the Cloud Via Statistical Learning

Jordan Hochenbaum Owen S. Vallis Arun Kejariwal
Twitter Inc.

ABSTRACT

Performance and high availability have become increasingly important drivers, amongst other drivers, for user retention in the context of web services such as social networks, and web search. Exogenic and/or endogenic factors often give rise to anomalies, making it very challenging to maintain high availability, while also delivering high performance.

ing initiatives [7, 8] have been undertaken. Likewise, there has been an increasing emphasis on developing techniques for detection, and root cause analysis, of performance issues in the cloud [9, 10, 11, 12, 13, 14, 15].

A lot of research has been done in the context of anomaly detection in various domains such as, but not limited to, statistics, signal processing, finance, econometrics, manufac-

https://github.com/twitter/AnomalyDetection

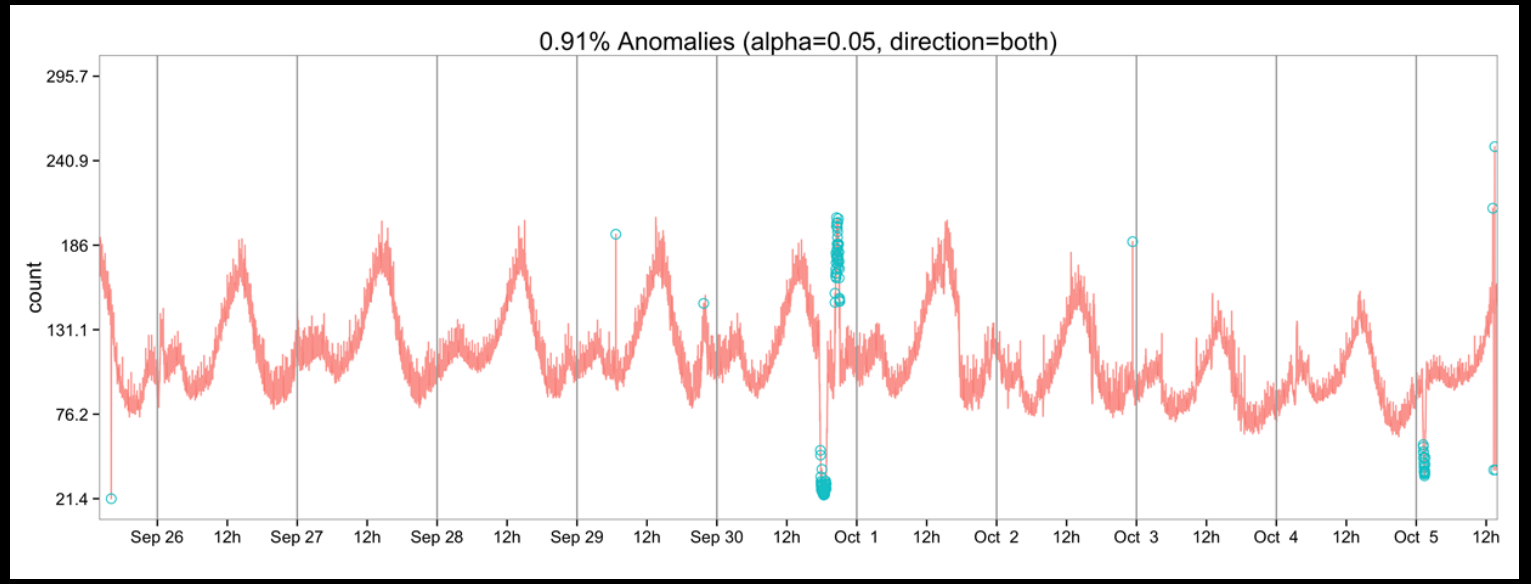
Twitter

AnomalyDetection R package

build passing | pending pull-requests 9 | open issues 30

AnomalyDetection is an open-source R package to detect anomalies which is robust, from a statistical standpoint, in the presence of seasonality and an underlying trend. The AnomalyDetection package can be used in wide variety of contexts. For example, detecting anomalies in system metrics after a new software release, user engagement post an A/B test, or for problems in econometrics, financial engineering, political and social sciences.

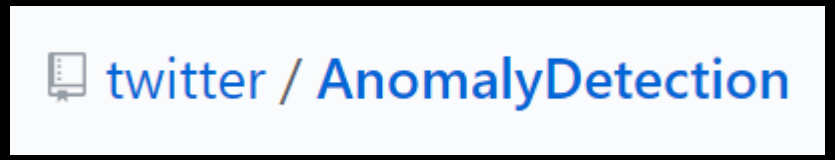
https://github.com/twitter/AnomalyDetection



Seasonality

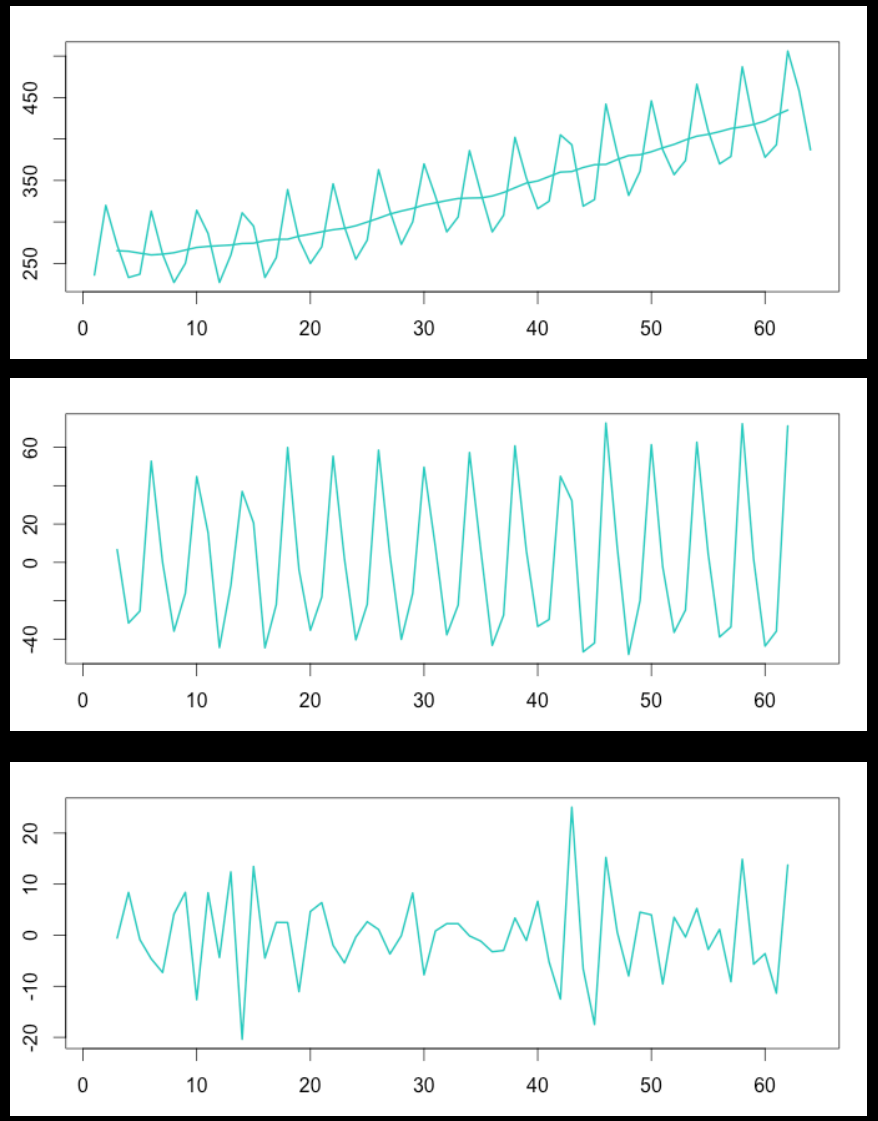
- Seasonal-Hybrid ESD algorithm (https://github.com/twitter/AnomalyDetection)
  - Input vector of values, period of repeating pattern, direction (find anomalies that are small/big/both)
  - ALL statistical, no ML

Github

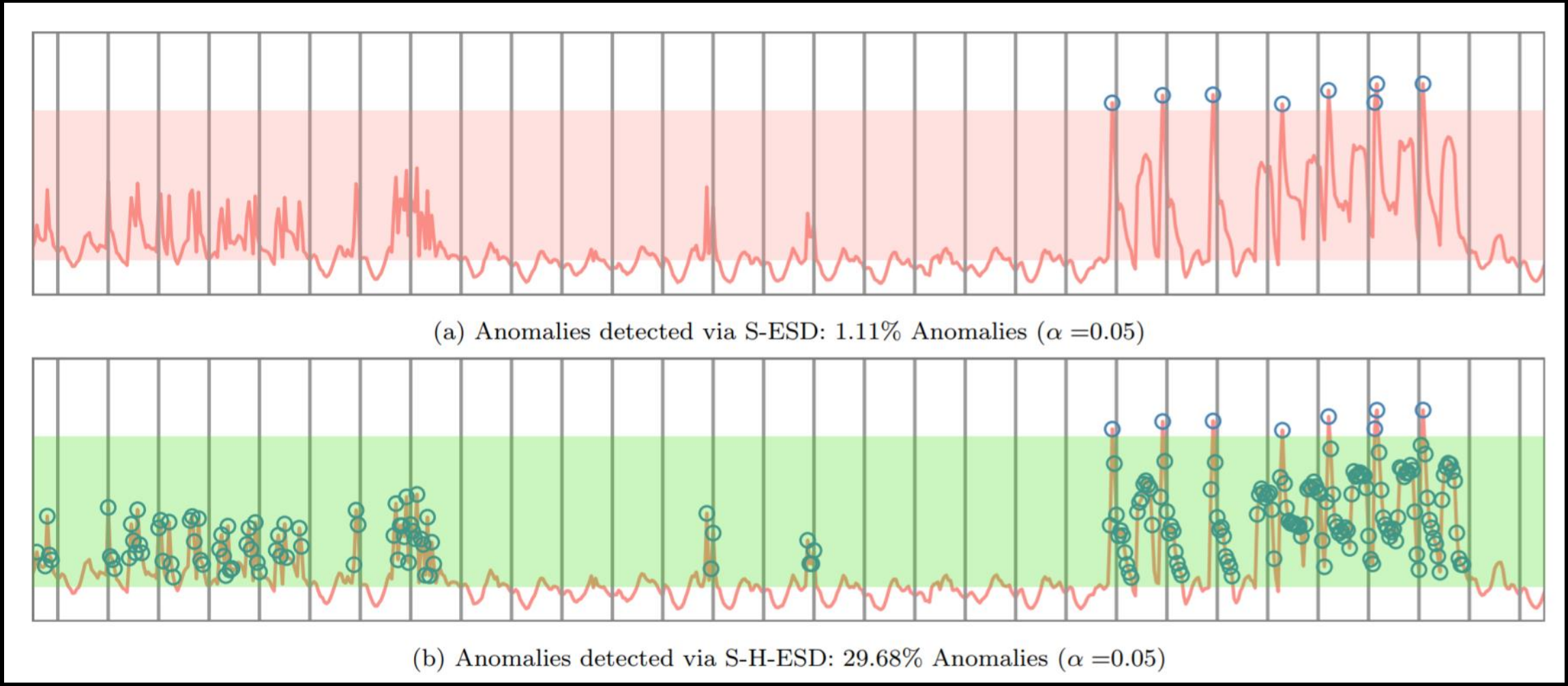


Decomposition: The Concept

y(t) = Level + Trend + Seasonality + Noise



Twitter Data (Source: Whitepaper about S-H-ESD)

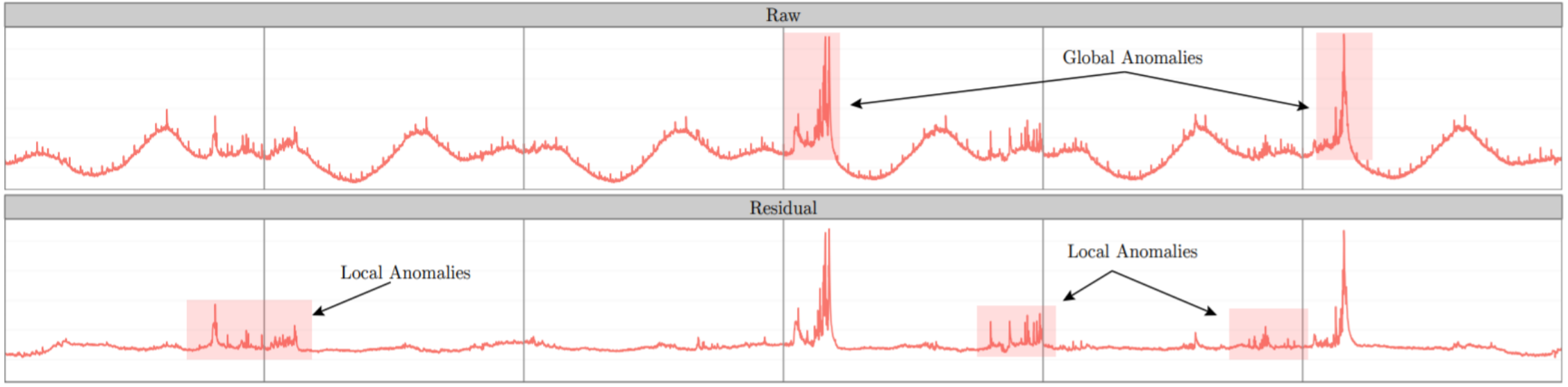


(a) Anomalies detected via S-ESD: 1.11% Anomalies (α =0.05)

(b) Anomalies detected via S-H-ESD: 29.68% Anomalies (α =0.05)

Using MAD exposes the local outliers ! (look at the bottom graph)

Source: https://blog.twitter.com/engineering/en_us/a/2015/introducing-practical-and-robust-anomaly-detection-in-a-time-series.html



Success !

I can capture global AND local anomalies !

Guidelines

- Synthetic datasets (create/capture/find)
- Differing Environments
- Refine (R3)
- How well do you think your approach scales ?
- GIGO
- Talk to people !

Outlier Detection DataSets (ODDS)

In ODDS, we openly provide access to a large collection of outlier detection datasets with ground truth (if available). Our focus is to provide datasets from different domains and present them under a single umbrella for the research community. As such, we arrange the datasets based on their types into different tables in the order as listed below. [read more about ODDS]

Key Takeaways

- Anomalies come in a lot of shapes and sizes !
- Statistical methods can be leveraged effectively in the right context,
- But context is everything
- Figure out a plan for `Slowly changing dimensions` !
- Manual anomaly detection is neither fast nor scalable
- **Stop trying to be absolutely perfect. Focus on capturing the *underlying structure* and everything will work out.**
  - The perfect detector:
    - Detects EVERY single anomaly (and it's real-time)
    - Triggers NO false alarms
    - Requires NO parameter tuning
    - Auto-magically adapts to changing statistics on the fly
    - And lastly, doesn't exist

https://github.com/TomBresee/Anomaly_Detection