

MAT 4376/5314E
Techniques of Data Analysis

Module 5
Anomaly Detection and Outlier Analysis

P.Boily (uOttawa, IACS, DAL)
with Y.Cissokkho, S.Fadel, R.Millson, R.Pourhasan

Fall 2020

Outline

With the advent of automatic data collection, it is now possible to store and process large troves of data. There are technical issues associated to massive data sets, such as the speed and efficiency of analytical methods, but there are also problems related to the detection of **anomalous observations** and the **analysis of outliers**.

Unexpected observations can spoil analyses and/or be indicative of data collection and data processing issues.

Extreme and irregular values behave very differently from the majority of observations: they can represent criminal attacks, fraud attempts, targeted attacks, or data collection errors. As a result, anomaly detection and outlier analysis plays a crucial role in cyber-security, quality control, etc.

5.1 – Basic Notions and Overview (p.5)

- Anomaly Detection as Statistical Learning (p.44)

5.2 – Quantitative Methods of Anomaly Detection (p.84)

- Distance-Based Methods (p.85)
- Density-Based Methods (p.121)

5.3 – Qualitative Methods (p.170)

- AVF Algorithm (p.175)
- Greedy Algorithm (p.180)

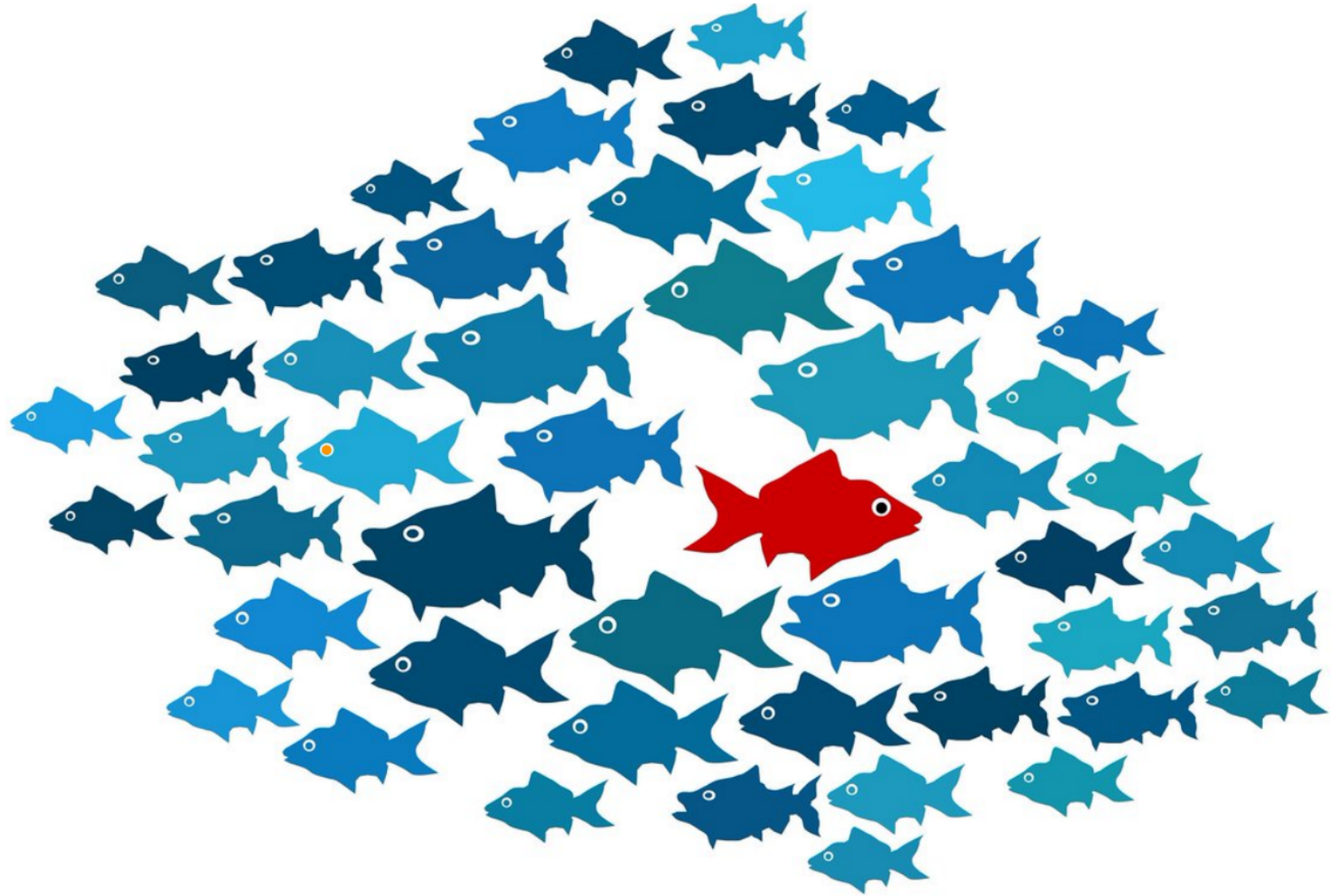
5.4 – Anomalies in High-Dimensional Datasets (p.184)

- Definitions and Challenges (p.185)
- Projection-Based Methods (p.187)
- Subspace Methods (p.207)

5.5 – Advanced Topics (p.212)

- Outlier Ensembles (p.213)
- Anomalies in Text Datasets (p.224)

References and other details can be found in Cissokho, Y., Fadel, S., Millson, R., Pourhasan, R., Boily, P. [2020], *Anomaly Detection and Outlier Analysis*, Data Science Report Series, Data Action Lab.



5.1 – Basic Notions and Overview

Isaac Asimov, the prolific American author, once wrote that

The most exciting phrase to hear [...], the one that heralds the most discoveries, is not “Eureka!” but “That’s funny...”.

Important Goals: establish anomaly detection protocols and to identify strategies to deal with such observations.

Outlying observations: data points that are atypical within-unit or between-units, or as part of a collective subset of observations.

In other words, outliers are observations which are **dissimilar to other cases** or which contradict **known dependencies** or rules.

Outlying observations may be anomalous along any of the individual variables, or in combination.

Observations could be anomalous in one context, but not in another:

- an adult male who is 6-foot tall falls in the 86th percentile among Canadian males \implies tall, but not unusually so;
- in Bolivia, the same man would land in the 99.9th percentile \implies extremely tall; a rarity.

Anomaly detection points towards interesting questions for analysts and subject matter experts: in this case, why is there such a large discrepancy in the two populations?

What's an **outlier/anomalous observation**? (reprise)

- **“bad” object/measurement:** data artifacts, spelling mistakes, poorly imputed values, etc.
- **misclassified observation:** according to the existing data patterns: the observation should have been labeled differently in the
- an observation whose measurements are found in the **distribution tails**, in a large enough number of features;
- **unknown unknowns:** completely new type of observations whose existence was hertofore unsuspected.

A common mistake that analysts make when dealing with outlying observations is to remove them from the dataset without carefully studying whether they are **influential data points**.

Influential observations are points whose absence leads to **markedly different** analysis results.

Points can be influential for one analytical methods, but not for another.

Remedial measures (data transformation strategies, etc.) may need to be applied to minimize any undue effect.

Outliers may be influential, and influential data points may be outliers, but the conditions are **neither necessary nor sufficient**.

Anomalies

Anomalies are **infrequent** and typically shrouded in **uncertainty** due to their relatively low numbers.

This makes it difficult to differentiate anomalies from banal **noise** or **data collection errors**.

The boundary between normal and deviant observations is usually **fuzzy**.

Example: before the advent of e-shops, a purchase which was recorded at 3AM (local time) would probably raise a red flag for a credit card company; but with online shops, that is not necessarily the case.

If anomalies are actually associated with **malicious activities**, they are often **disguised** to blend in with normal observations \implies this obviously complicates the detection process.

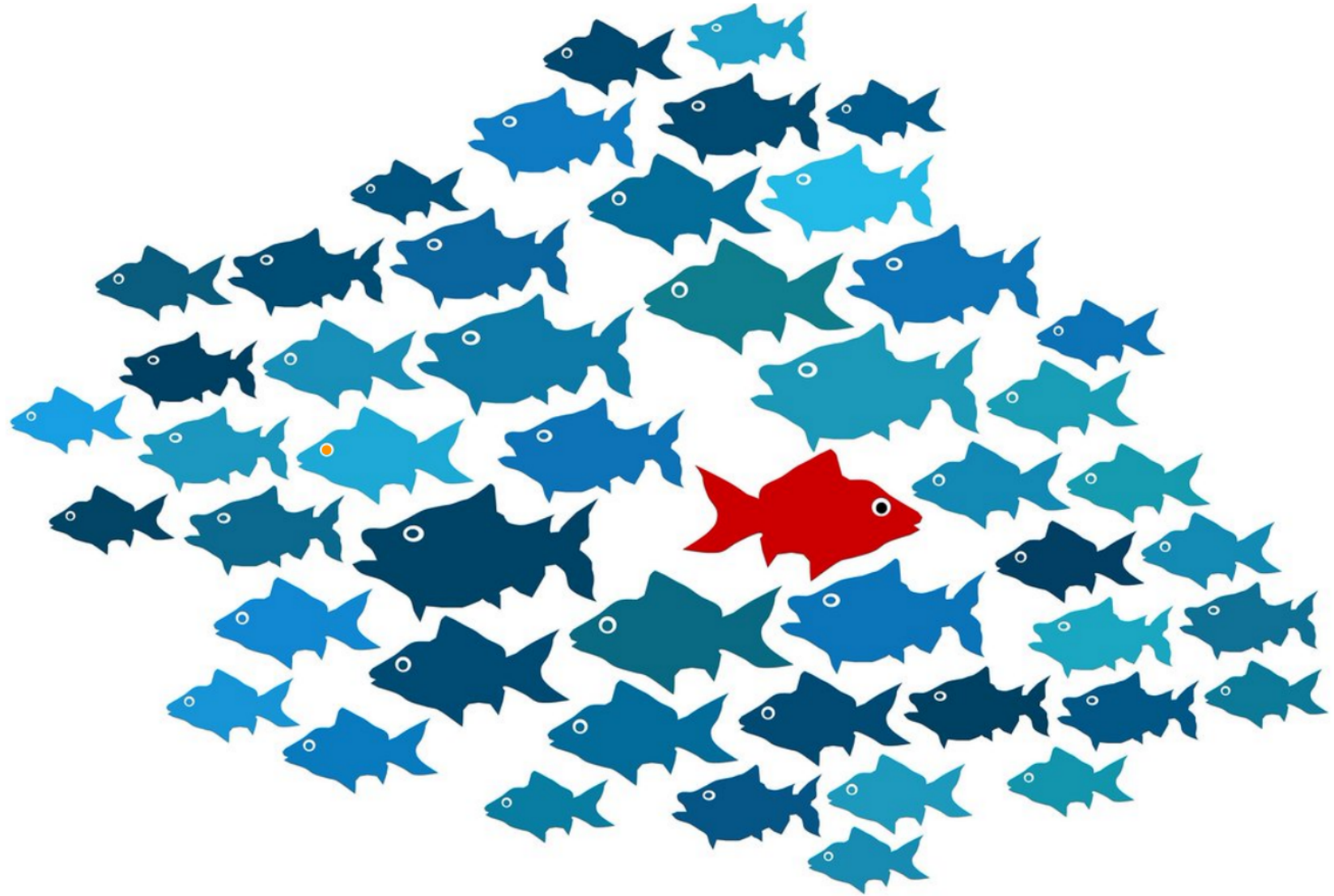
Numerous methods exist to identify anomalous observations; **none of them are foolproof** and judgement must be used.

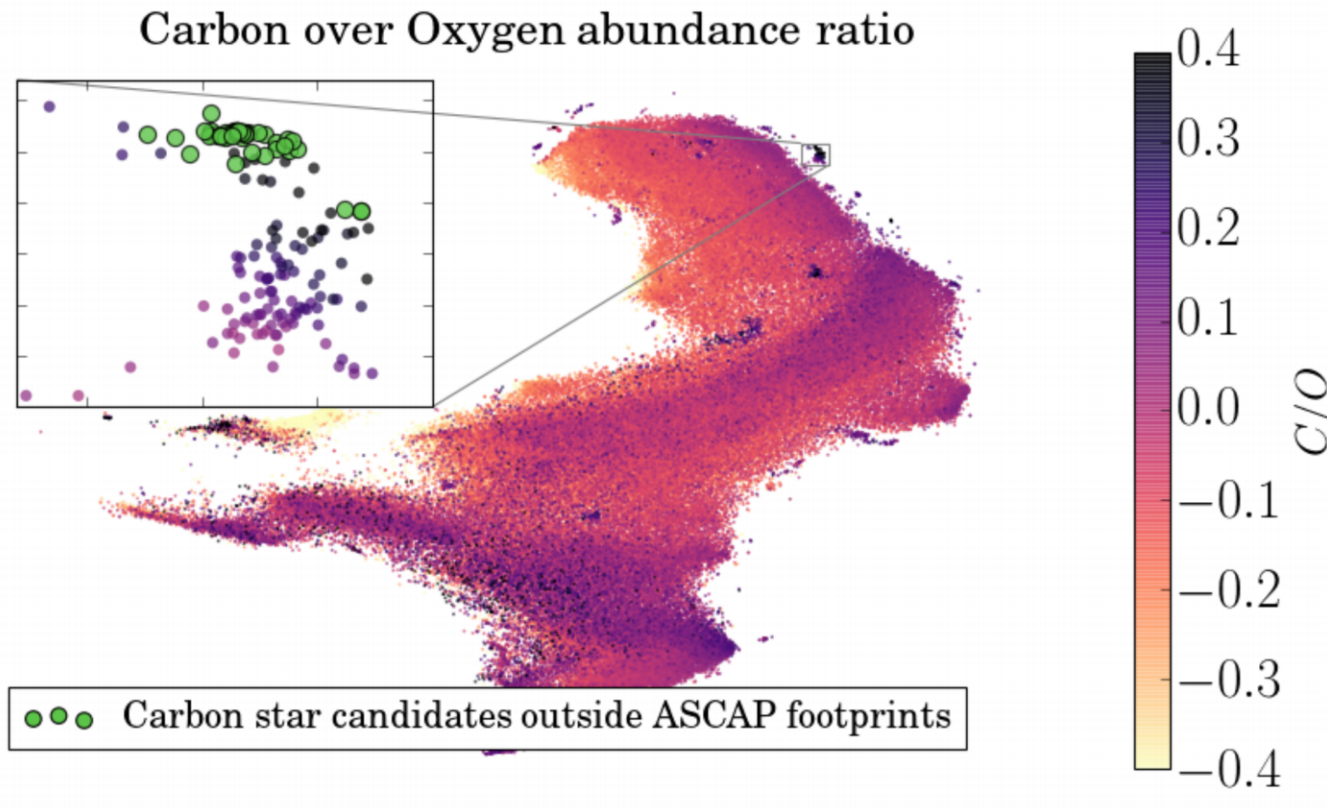
Graphical methods to identify outliers are particularly easy to implement:

- boxplots, scatterplots, scatterplot matrices, and 2D tours

usually require a low-dimensional setting for **interpretability**.

They also usually find those anomalies that “**shout the loudest**” [Baron].





Derived-score anomaly detection may help (... or it may not) [Baron]

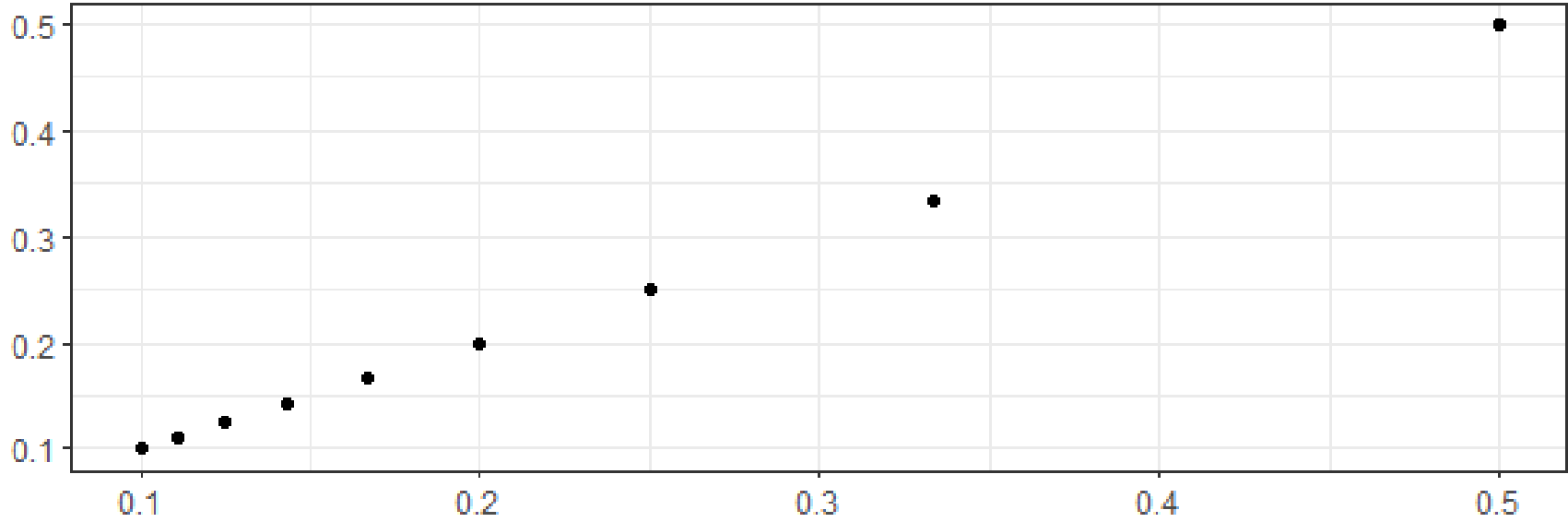
Simple analytical methods using Cooke's or Mahalanobis' distances are sometimes used, but more sophisticated analysis is usually required, especially when trying to identify influential points (*cf.* **leverage**).

In small datasets, detection can be conducted on a case-by-case basis.

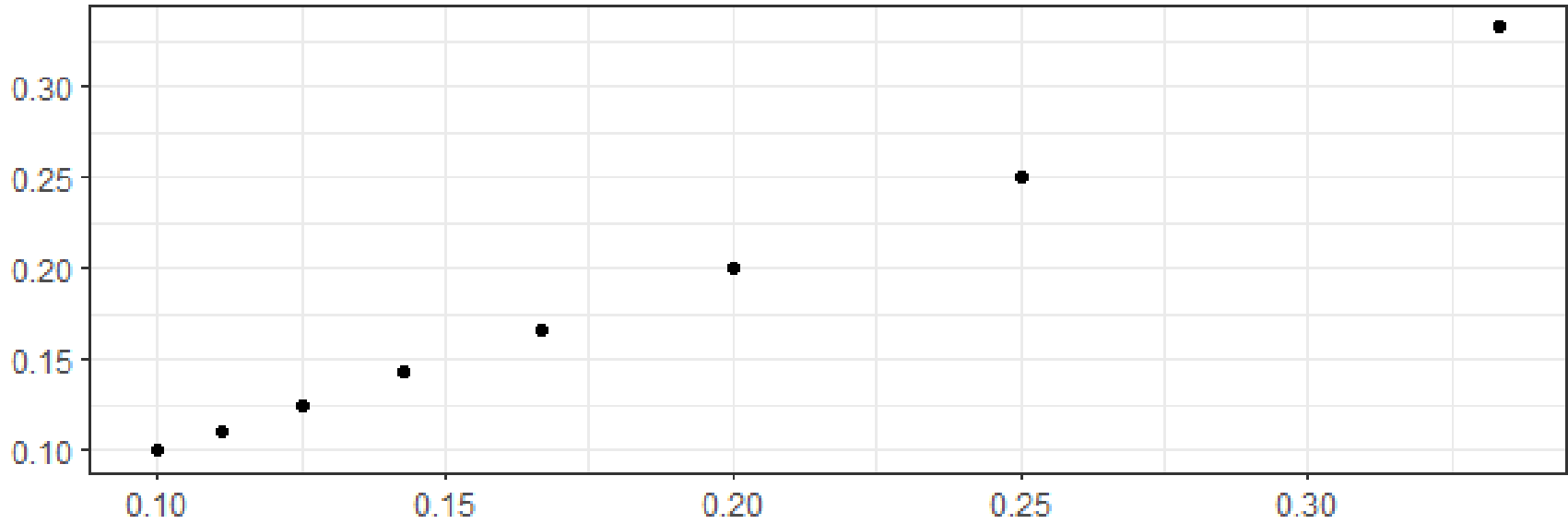
Questions: how many anomalies are too many to find? How many cases are you willing to inspect manually?

It is tempting to use **automated detection/removal** with large datasets, but doing so may be catastrophic from a data analysis perspective!

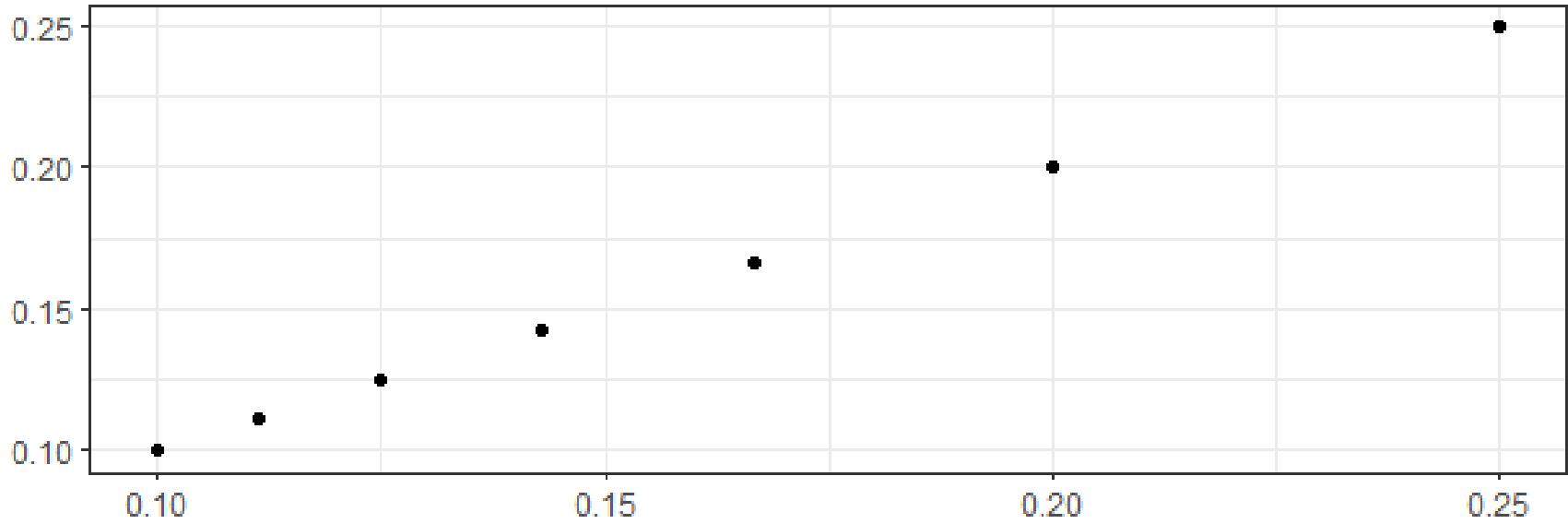
If once “anomalous” observations have been removed from the dataset, previously “regular” observations can become anomalous in turn in the smaller dataset – when does the runaway train?



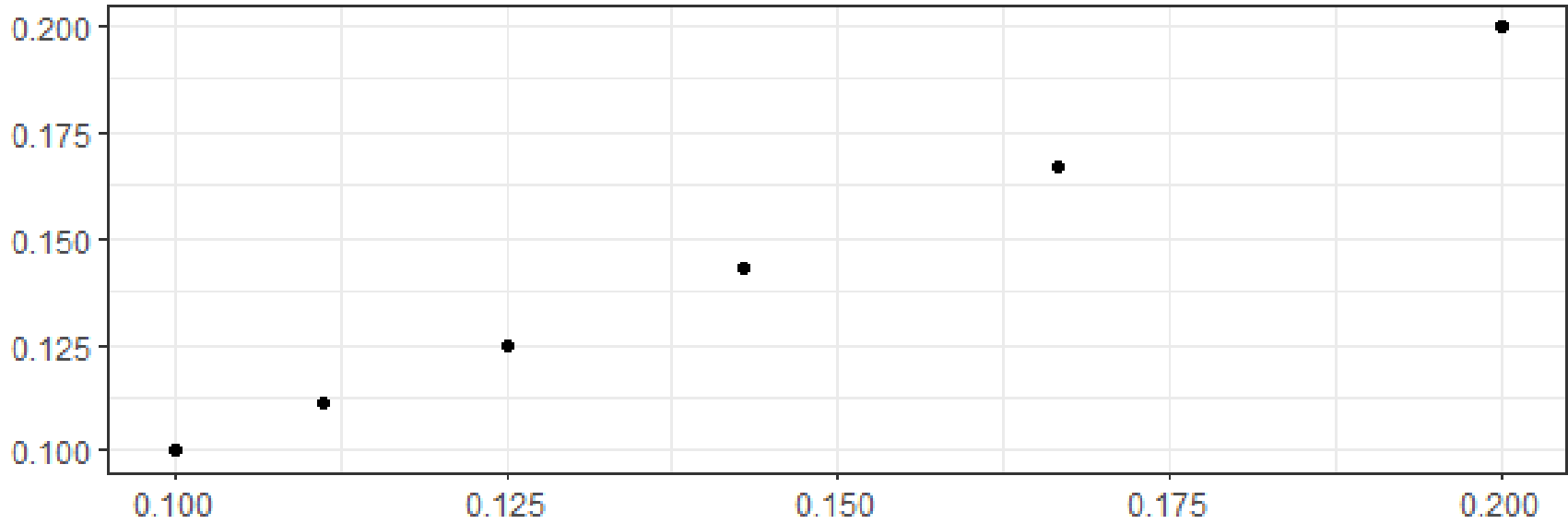
Automatic renewal of anomaly (step 1)



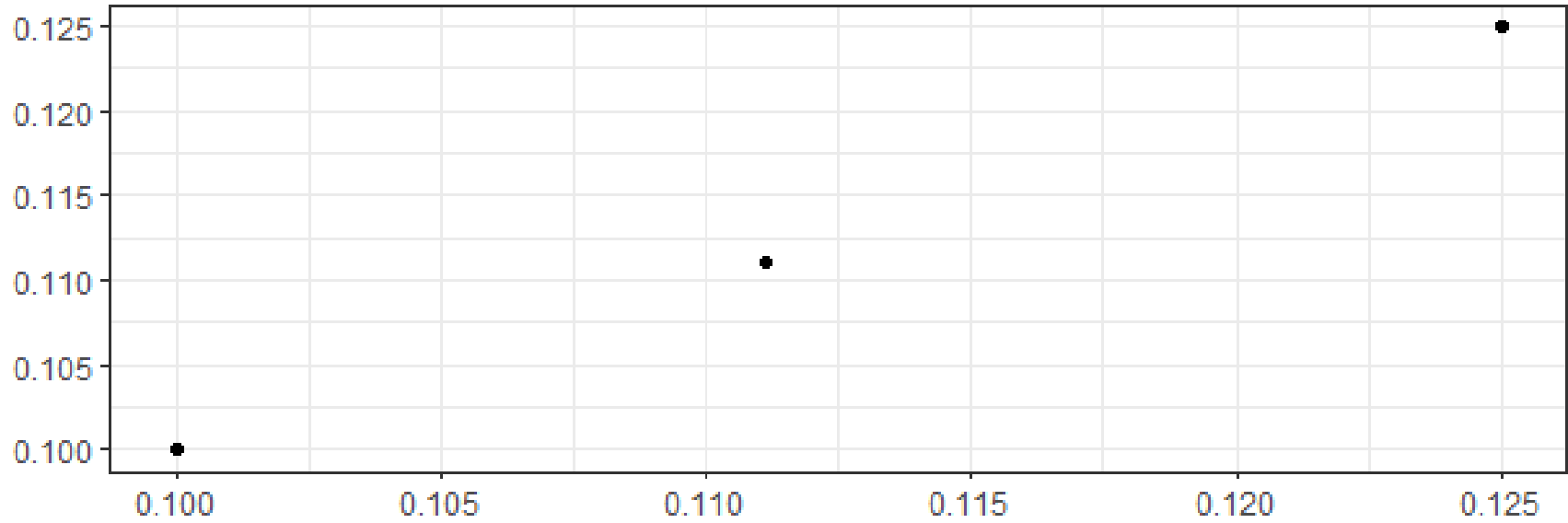
Automatic renewal of anomaly (step 2)



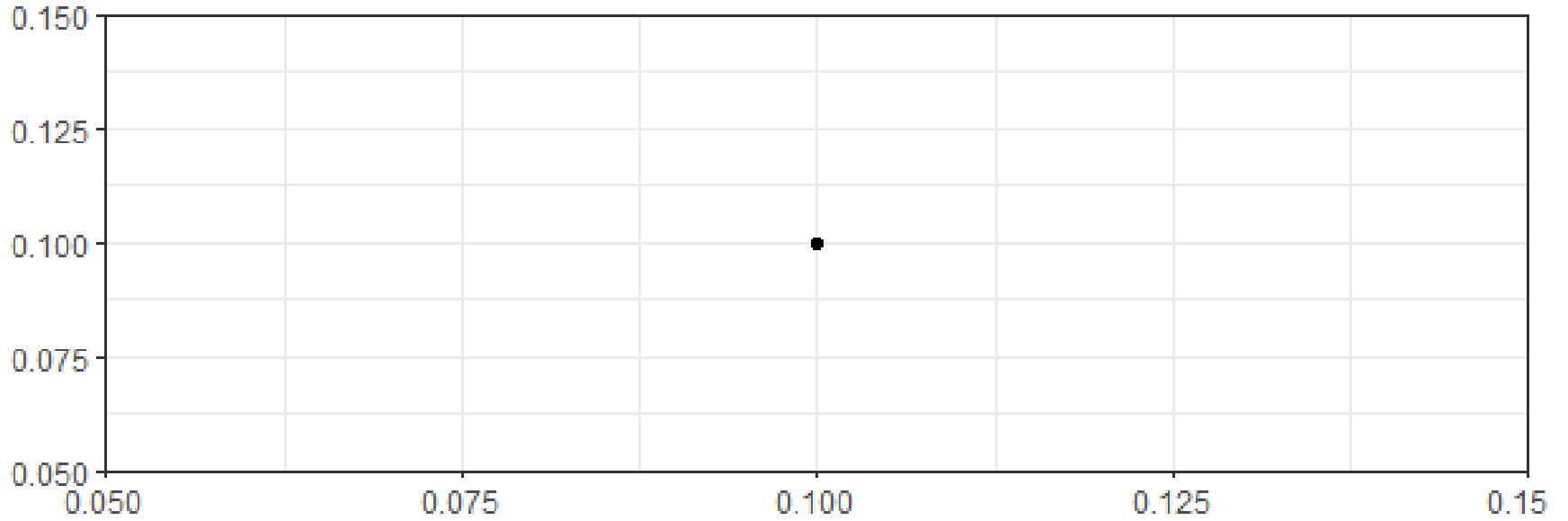
Automatic renewal of anomaly (step 3)



Automatic renewal of anomaly (step 4)



Automatic renewal of anomaly (step n)



Automatic renewal of anomaly (step 9)

In the early stages of anomaly detection, we use **simple data analyses**:

- descriptive statistics,
- 1– and 2–way tables, and
- traditional visualizations.

The goal is to **help identify anomalous observations** and to **obtain insights about the data**.

This leads to more sophisticated anomaly detection methods and could also eventually lead to modifications of the analysis plan.

THIS IS NEVER AN UNWELCOME DEVELOPMENT!

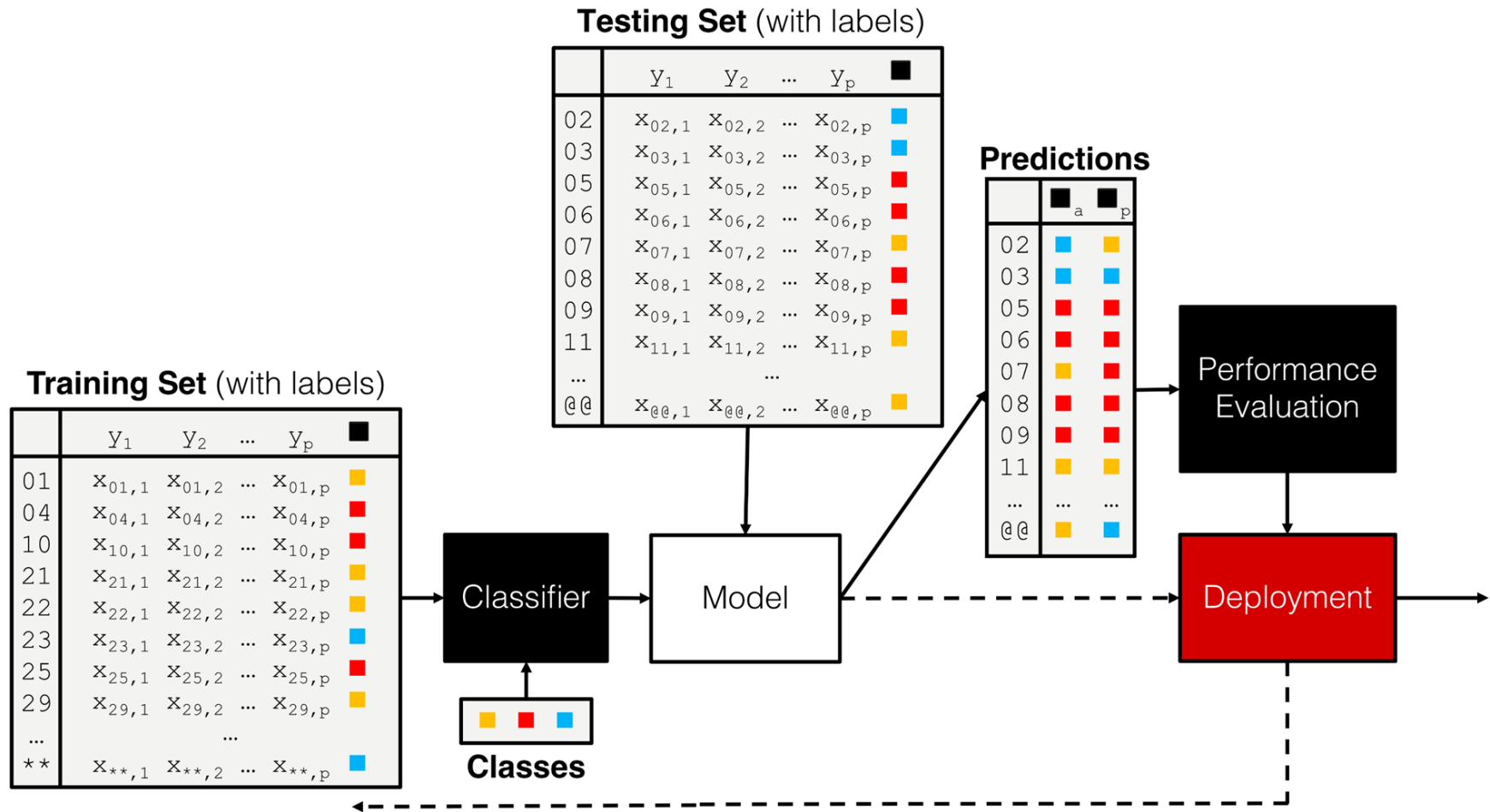
Learning Framework

How are outliers detected, in practice?

Methods come in two flavours:

- **supervised**, and
- **unsupervised**.

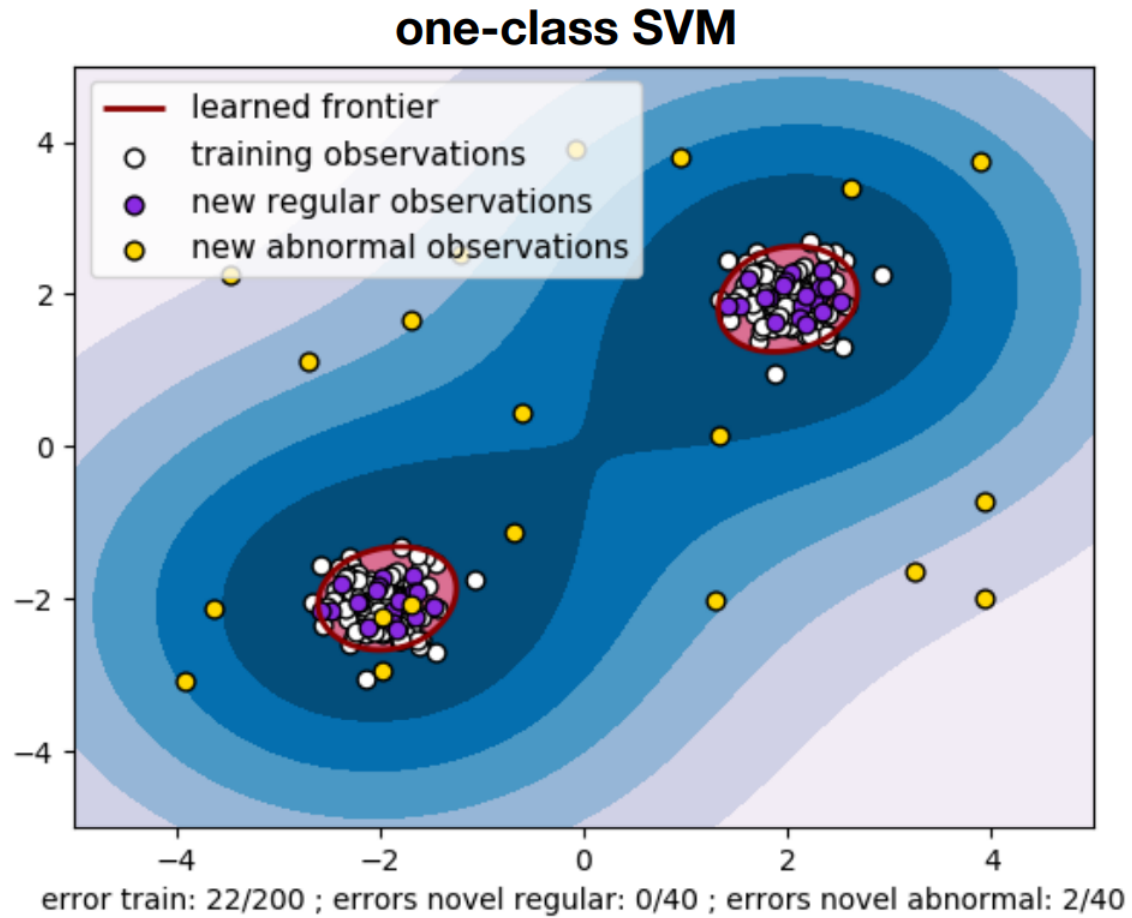
Supervised methods (SL) use a historical record of **previously identified anomalous observations** to build a **predictive classification or regression model** which estimates the probability that a unit is anomalous.



SL Challenges:

- domain expertise and resources are required to tag the data;
- since anomalies are typically **infrequent**, these models often also have to accommodate the **rare occurrence** (or class imbalance) problem, and
- SL methods need to minimize a **loss function** (cost of making a mistake) which is usually symmetrical (in the anomaly detection context, this is not usually a valid assumption).

Even more than in traditional analysis settings, anomaly detection can lead to **technically correct but ultimately useless** (non-actionable) **results**.



Learning an **anomaly frontier** [Baron].

Example: The vast majority ($99.999+\%$) of air passengers **do not** bring weapons with them on flights.

A model that predicts that no passenger is ever attempting to smuggle a weapon on board a flight would be $99.999+\%$ accurate.

But it would miss the point **completely**. For the **security agency**, the cost of wrongly thinking that a passenger is:

- smuggling a weapon \implies cost of a single search;
- NOT smuggling a weapon \implies catastrophe (potentially).

The wrongly targeted individuals may have a ... somewhat different take on this, from a societal and personal perspective.

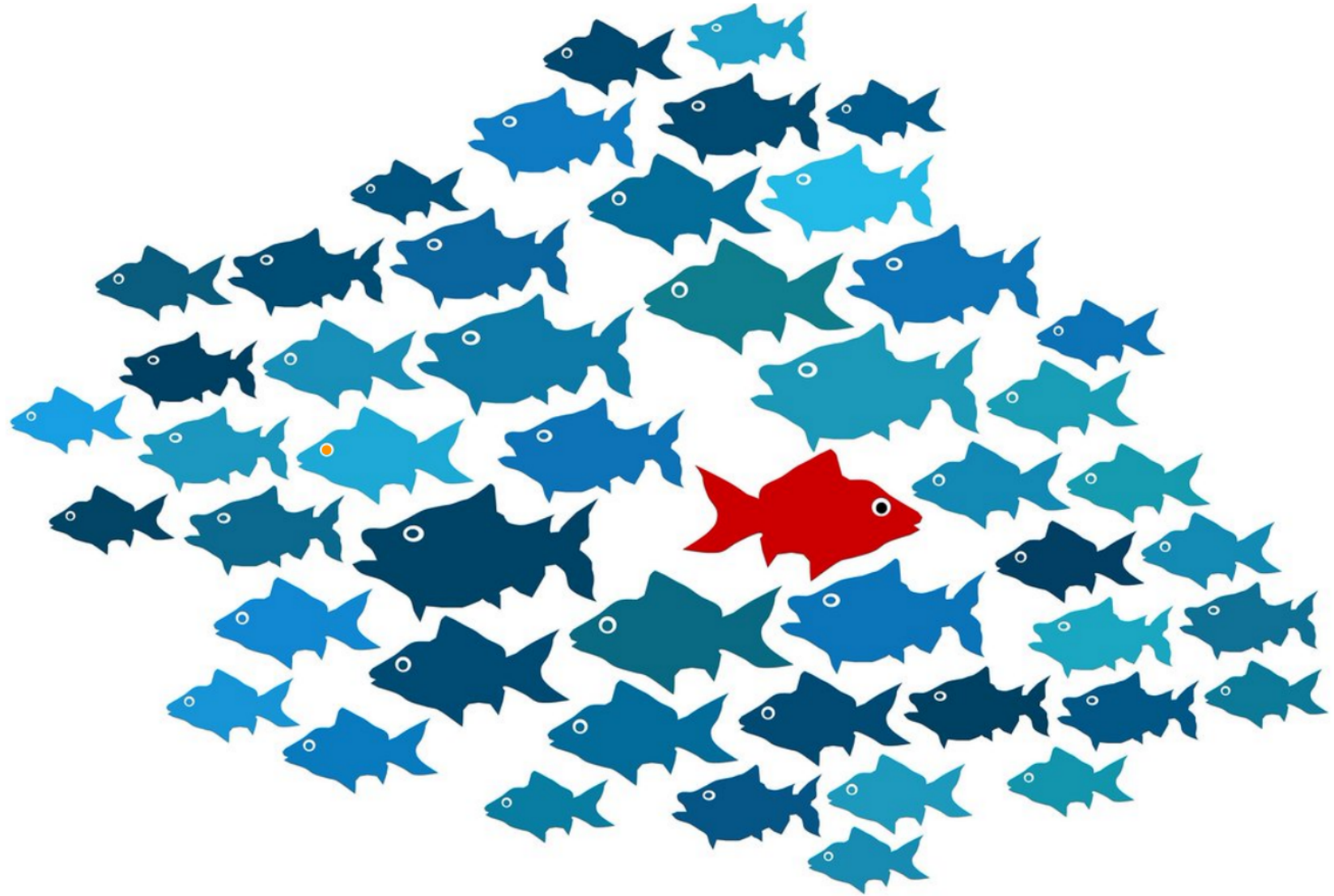
Unsupervised methods (UL)

- use no previously labeled (anomalous/non-anomalous) data, and
- try to determine if an observation anomalous solely by comparing its behaviour to that of the other observations.

Example: if all workshop participants except for one can view the video conference lectures, then the one individual/internet connection/computer is **anomalous** – it behaves in a manner which is different from the others.

VERY IMPORTANT NOTE: this **DOES NOT** mean that the different behaviour is the one we are actually interested in/searching for!

Be weary: this is true of anomaly detection in data and in real-life.



Traditional Outlier Detection Tests

The most commonly-used test is **Tukey's** (univariate) **boxplot test**. Let Q_1 and Q_3 represent an observed feature's 1st and 3rd quartile, respectively.

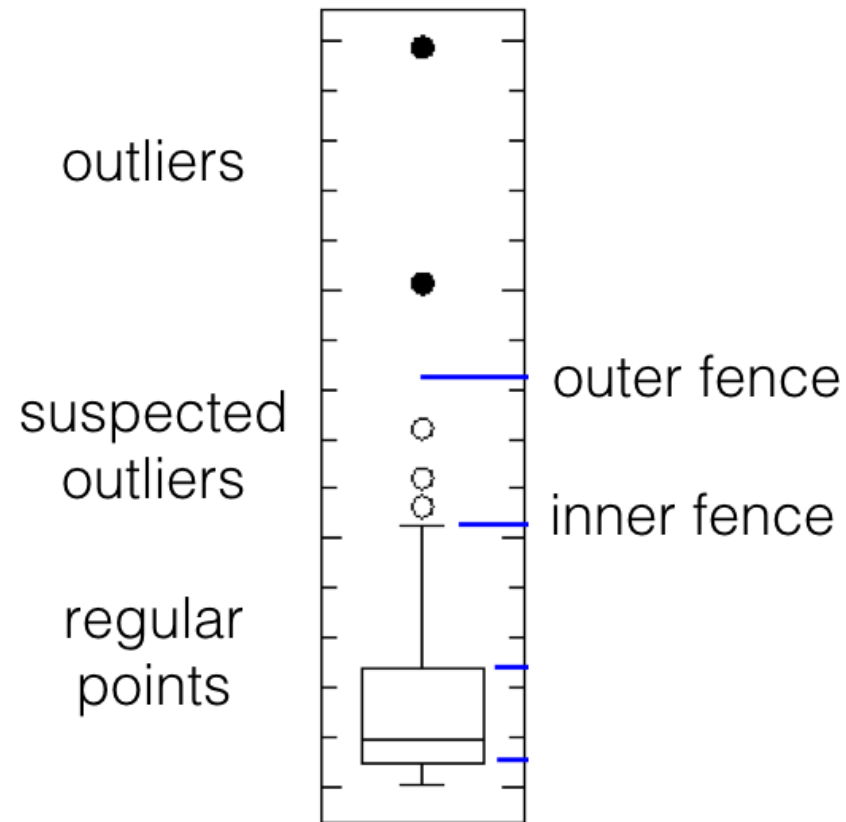
For **normally distributed** measurements, regular observations typically lie between the **inner fences**

$$Q_1 - 1.5(Q_3 - Q_1) \quad \text{and} \quad Q_3 + 1.5(Q_3 - Q_1).$$

Suspected outliers lie between the inner fences and their **outer fences**

$$Q_1 - 3(Q_3 - Q_1) \quad \text{and} \quad Q_3 + 3(Q_3 - Q_1).$$

Points beyond the outer fences are identified as **outliers**.



Suspected outliers are marked by white disks, outliers by black disks. Use `boxplot()` and `boxplot.stats()` in R to plot and identify outliers in normally distributed data.

The **Grubbs test** is another univariate test:

H_0 : no outlier in the data vs. H_1 : **exactly one** outlier in the data.

- let x_i be the value of feature X for the i^{th} unit, $1 \leq i \leq N$,
- let (\bar{x}, s_x) be the mean and standard deviation of feature X ,
- let α be the desired significance level, and
- let $T(\alpha/2N; N)$ be the critical value of the Student t -distribution.

The test statistic is

$$G = \frac{\max_i \{|x_i - \bar{x}|\}}{s_x} = \frac{|x_{i^*} - \bar{x}|}{s_x}.$$

Under H_0 , G follows a special distribution with critical value

$$\ell(\alpha; N) = \frac{N-1}{\sqrt{N}} \sqrt{\frac{T^2(\alpha/2N, N)}{N-2+T^2(\alpha/2N, N)}}.$$

At significance level α , we reject the null hypothesis in favour of the alternative (i.e. x_{i^*} is the outlier) if $G \geq \ell(\alpha; N)$.

If looking for more than one outlier, it can be tempting to classify every observation i for which

$$\frac{|x_i - \bar{x}|}{s_x} \geq \ell(\alpha; N)$$

as an outlier, but this is **NOT RECOMMENDED**.

Other generalizations are also problematic (cf. outlier sequence).

Other common tests include:

- the **Mahalanobis distance**, which is linked to the leverage of an observation (a measure of influence), can also be used to find multi-dimensional outliers, when all relationships are linear (or nearly linear);
- the **Tietjen-Moore** test, which is used to find a specific number of outliers (this is similar to Grubbs' test, replacing H_1 by H_k);
- the **generalized extreme studentized deviate** test, the preferred extension to Grubbs' test if the number of outliers is unknown;
- the **chi-square** test, when outliers affect the goodness-of-fit;
- DBSCAN and other clustering-based outlier detection methods.

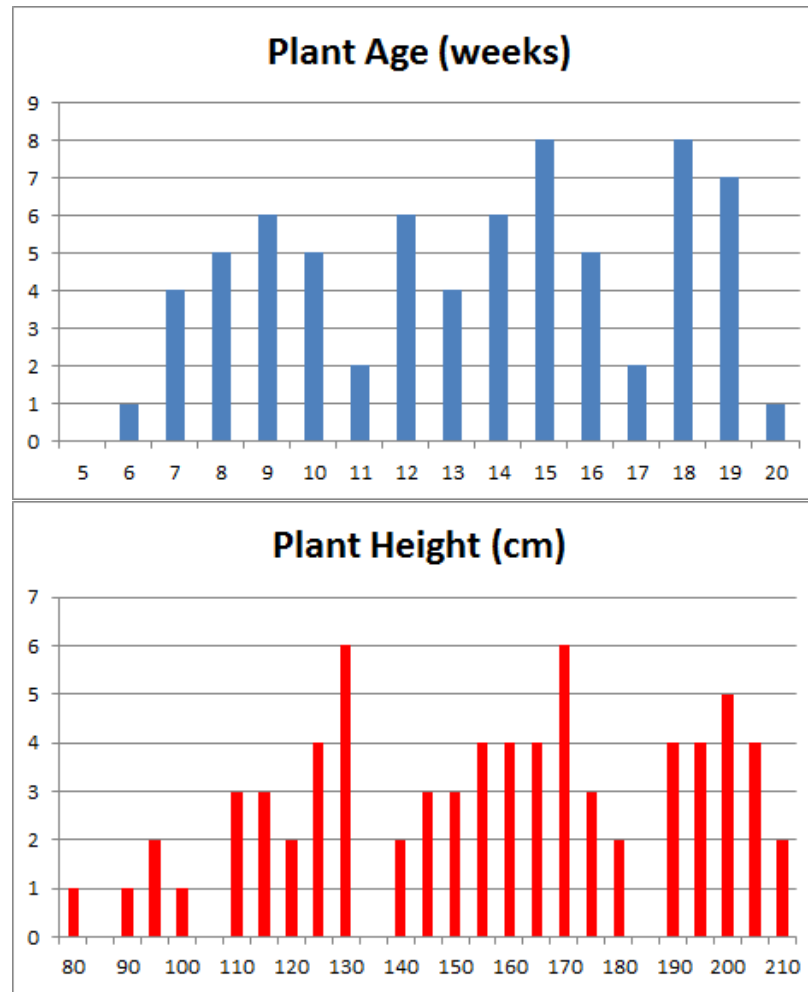
Visual Outlier Detection

The following simple examples illustrate the principles underlying **visual outlier and anomaly detection**.

Example 1: on a specific day, the **height** of several plants in a nursery are measured. The records also show each plant's **age** (the number of weeks since the seed has been planted).

Very little can be said about the data at that stage:

- the age of the plants (controlled by the nursery staff) seems to be somewhat haphazard,
- as does the response variable (height).



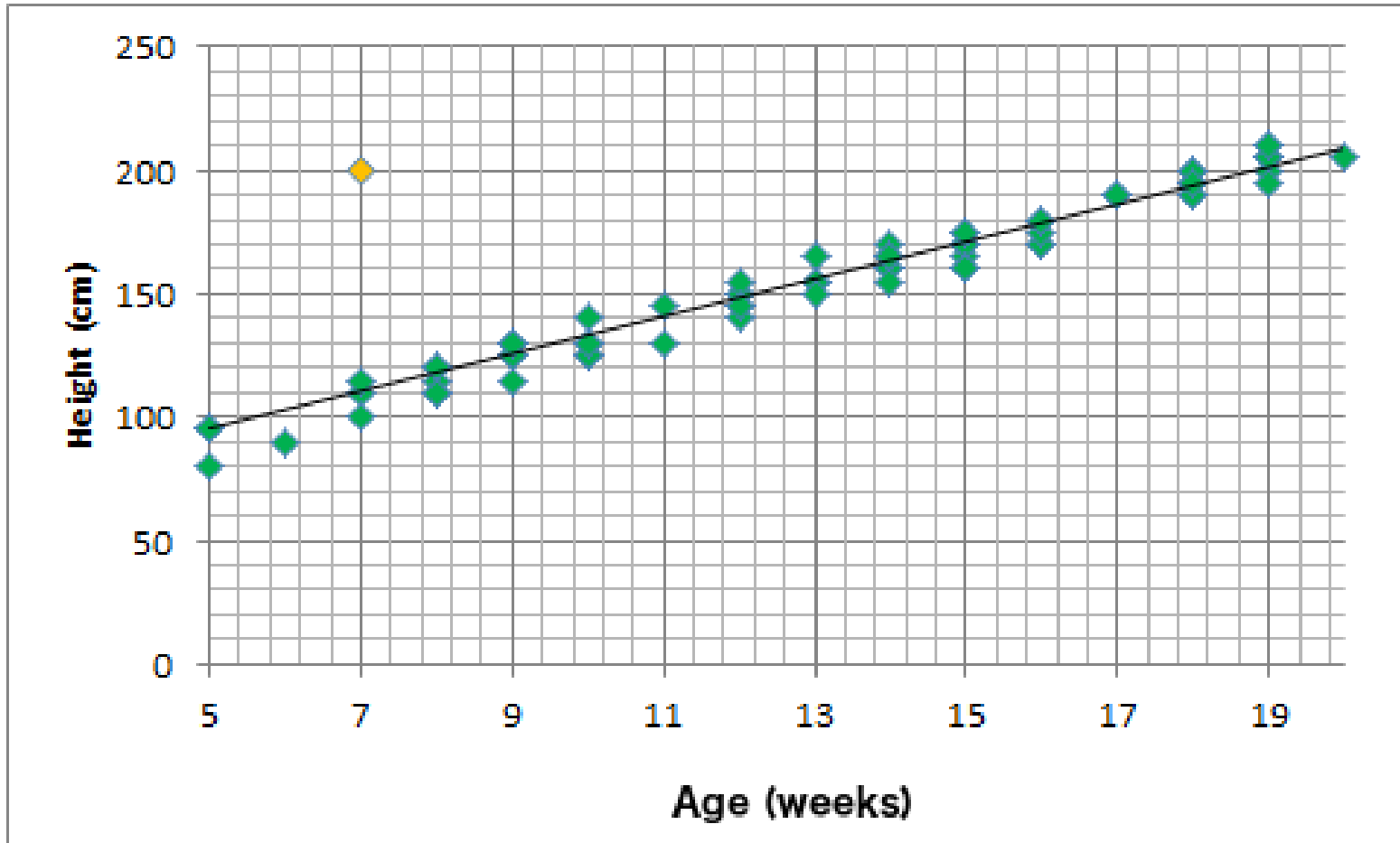
A scatter plot of the data reveals that **growth is strongly correlated with age** for the observations in the dataset; points clutter around a linear trend.

One point (in yellow) is easily identified as an **outlier**.

There are (at least) two possibilities:

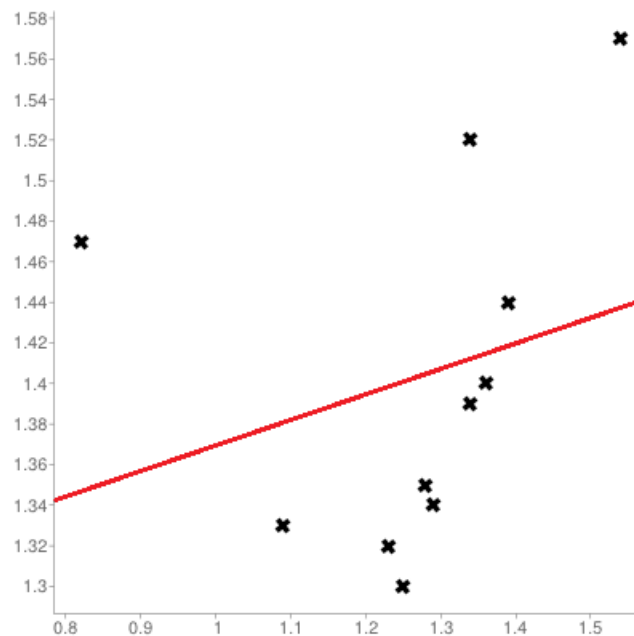
- either that measurement was botched or mis-entered in the database (representing an invalid entry), or
- that one specimen has experienced unusual growth (outlier).

Either way, the analyst has to investigate further.

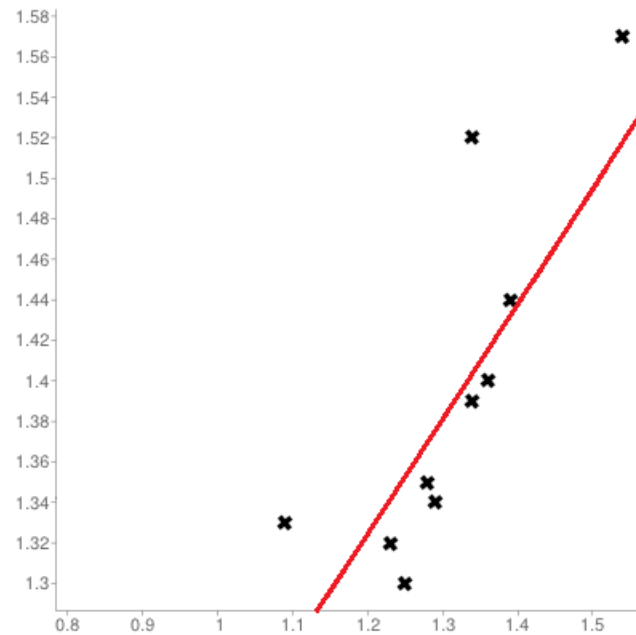


Example 2: a government department has 11 service points. The monthly average arrival and service rates per teller for each service point are available.

The scatter plot of the service rate per teller (y axis) against the arrival rate per teller (x axis), with linear regression trend, is shown below.



A similar chart, but with the left-most point removed from consideration, is shown below.



The trend still slopes upward, but the fit is significantly improved.

This suggests that the removed observation is unduly **influential** (or anomalous) – a better understanding of the relationship between arrivals and services is afforded if it is set aside.

Any attempt to fit that data point into the model must take this information into consideration.

The status of an influential observations **depends on the analysis that is ultimately conducted** – a point may be influential for one analysis, but not for another.

Note that setting aside an influential observation does not mean that the observation is removed from the dataset – only that it will not be used in a specific analysis.

Example 3: Measurements of the length of the appendage of a certain species of insect have been made on 71 individuals. Descriptive statistics have been computed; the results are shown below.

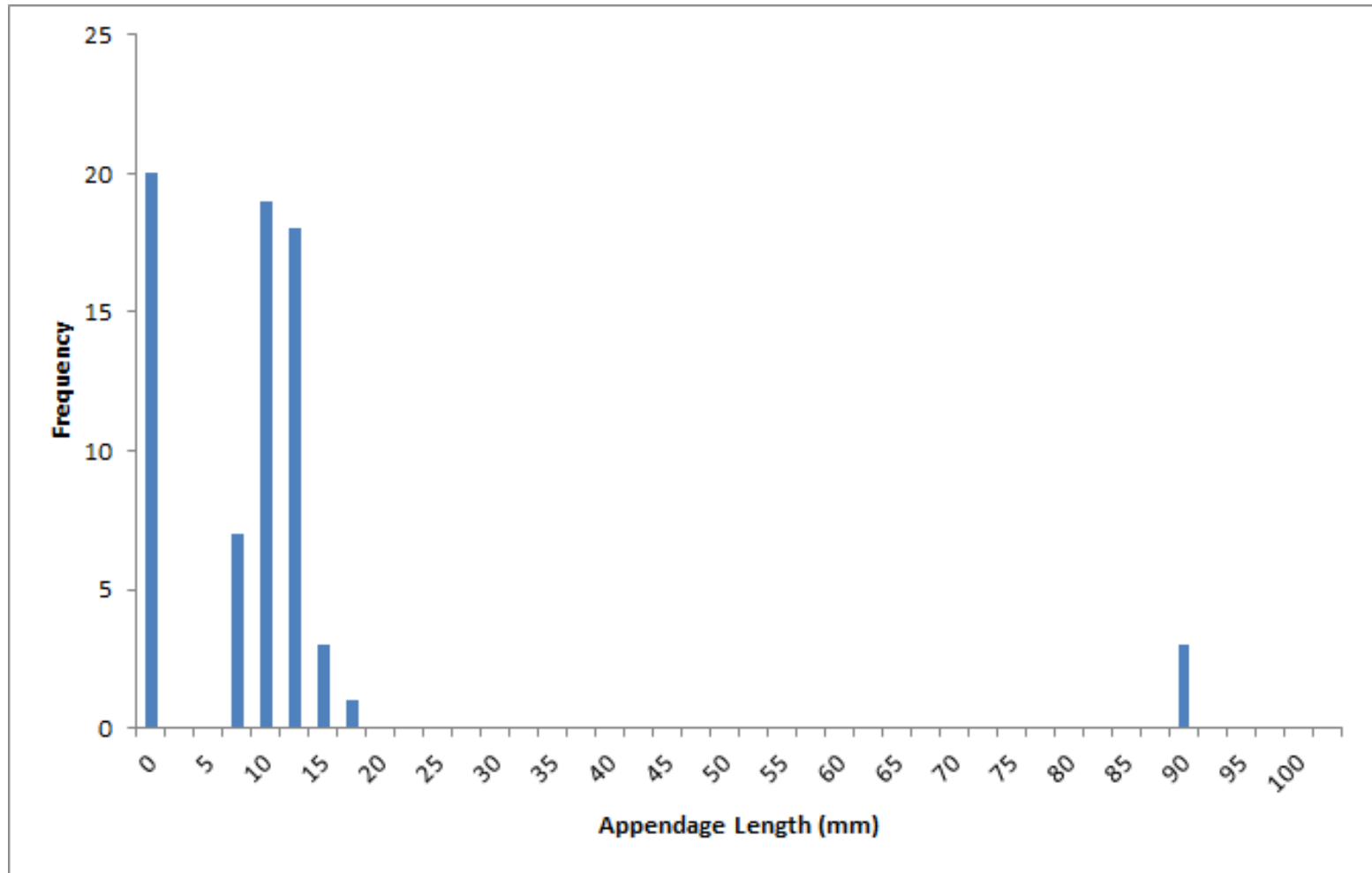
<i>Appendage length (mm)</i>	
Mean	10.35
Standard Deviation	16.98
Kurtosis	16.78
Skewness	4.07
Minimum	0
First Quartile	0
Median	8.77
Third Quartile	10.58
Maximum	88
Range	88
Interquartile Range	10.58
Mode	0
Count	71

The descriptive statistics might help the analyst recognize the tell-tale signs that the distribution of appendage lengths is likely to:

- be **asymmetrical** (since skewness is non-negligible), and
- have a **“fat” tail** (since large kurtosis, $\text{range} \gg \text{interquartile range}$, and $\text{max} \gg Q_3$)

The mode, min, and Q_1 belong to individuals without appendages \implies at least two sub-groups in the population (perhaps split along the lines of juveniles/adults, or males/females).

Since $\text{max} \gg$ other observations, might it belong to an **outlier**? The histogram of the measurements shows 3 individuals with long appendages.



It is plausible that these individuals belong to another species who were **erroneously added** to the dataset.

On its own, the chart does not constitute a proof of such an error, but it **raises the possibility of an error**, which is often the best that an analyst can do in the absence of subject matter expertise.

This traditional approach to anomaly detection is difficult to apply to high-dimensional datasets because it is nearly impossible to visualize them directly \implies fundamentally different approaches are needed.

Dimension reduction methods can be used to provide a **low-dimensional representation** of the data on which to apply visual detection (see autoencoder example), but some information always gets lost in the process.

5.1.1 – Anomaly Detection as Statistical Learning

Fraudulent behaviour is not always easily identifiable, even after the fact.

Example: credit card fraudsters try to disguise their transactions as regular and banal, and try to avoid outlandish behaviour.

Their goal: fool human observers into confusing **plausible** (or possible) with **probable** (or at least, **not improbable**).

It is plausible that a generic 40-something father of 3 might purchase a new TV; is it probable that THIS particular father of 3 would do so?

But it's unlikely that a generic father of 3 who resides in North America would purchase a round of drinks at a dance club in Kiev.

Anomaly detection is really a problem in **applied probability**. Let I be what is known about the dataset/situation:

- behaviour of individual observations,
- behaviour of observations as a whole,
- anomalous/normal verdict for a number of similar observations, etc.

Main Question: is $P(\text{obs. is anomalous} \mid I) > P(\text{obs. is normal} \mid I)$?

Anomaly detection models assume **stationarity of regular observations**: that the underlying mechanism that generates regular data does not change much over time.

For time series data, this means that it may be necessary to first perform **trend and seasonality extraction**.

Example: supply chains play a crucial role in the transportation of goods from one part of the world to another – as the saying goes, “a given chain is only as strong as its weakest link.”

Say that marine cargo departing Shanghai in Feb’13 took two more days, on average, to arrive in Vancouver than those departing in Jul’17.

- Has the shipping process improved in the intervening years?
- Do departures in Feb usually take longer to reach Vancouver?
- Are either the Feb’13 or the Jul’17 performance anomalous?

Seasonal variability is relevant to supply chain monitoring: quantifying and accounting for impact severity is of great interest.

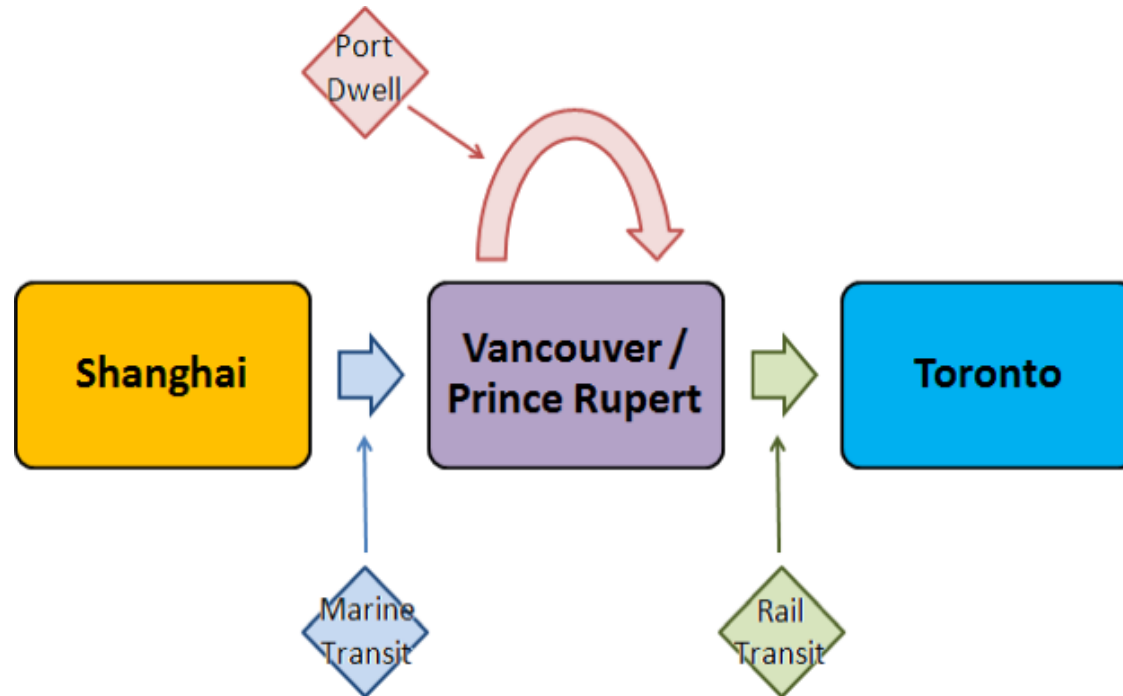
Potential Solution: create an **index** to track container transit times.

This index should depict

- **reliability** and
- **variability** of transit times,
- and allow for performance comparison between differing time periods.

Consider the scenario where we want to compare the monthly performance, irrespective of the transit season, of the corridor

Shanghai → Port Metro Vancouver/Prince Rupert → Toronto.



For each of the three segments (Marine Transit, Port Dwell, Rail Transit), the data consists of:

- monthly empirical distribution of transit/dwell times
- built from sub-samples (assumed to be randomly selected and fully representative) of all containers entering the appropriate segment.

Specific containers are not followed from Shanghai to Toronto: no covariance information about the various transit/dwell times is available.

Each segment's performance is measured using **fluidity indicators**, which are computed using various statistics of the transit/dwell time distributions for each of the supply chain segments.

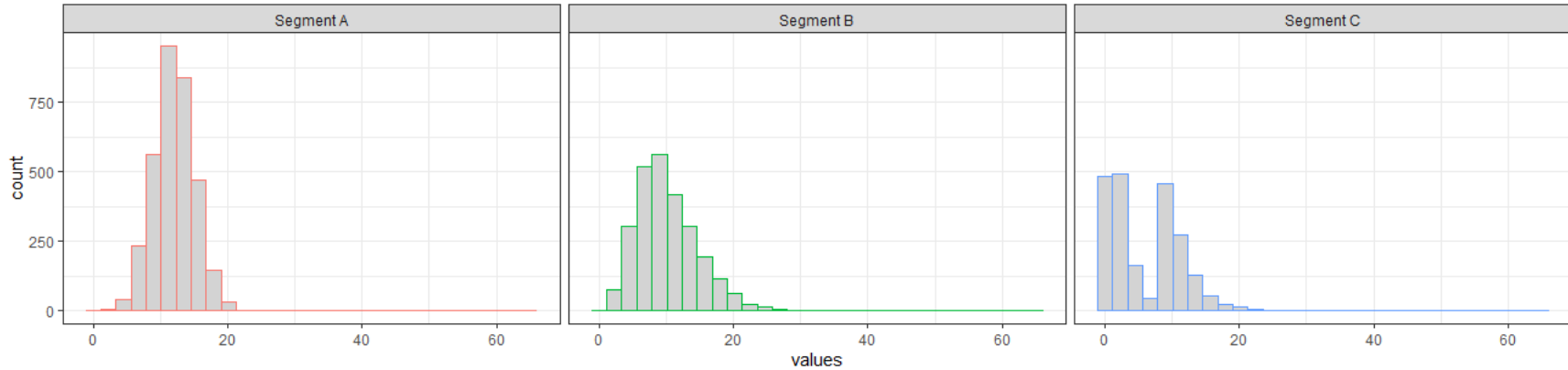
Reliability Indicator (RI) – the ratio of the 95th percentile to the 5th percentile of transit/dwell times.

A high RI indicates high volatility, whereas a low RI (≈ 1) indicates a reliable corridor.

Buffer Index (BI) – the ratio of the positive difference between the 95th percentile and the mean, to the mean.

A small BI (≈ 0) indicates only slight variability in the upper (longer) transit/dwell times; a large BI indicates that the variability of the longer transit/dwell times is high, and that outliers might be found there;

Coefficient of Variation (CV) – the ratio of the standard deviation of transit/dwell times to the mean transit/dwell time.

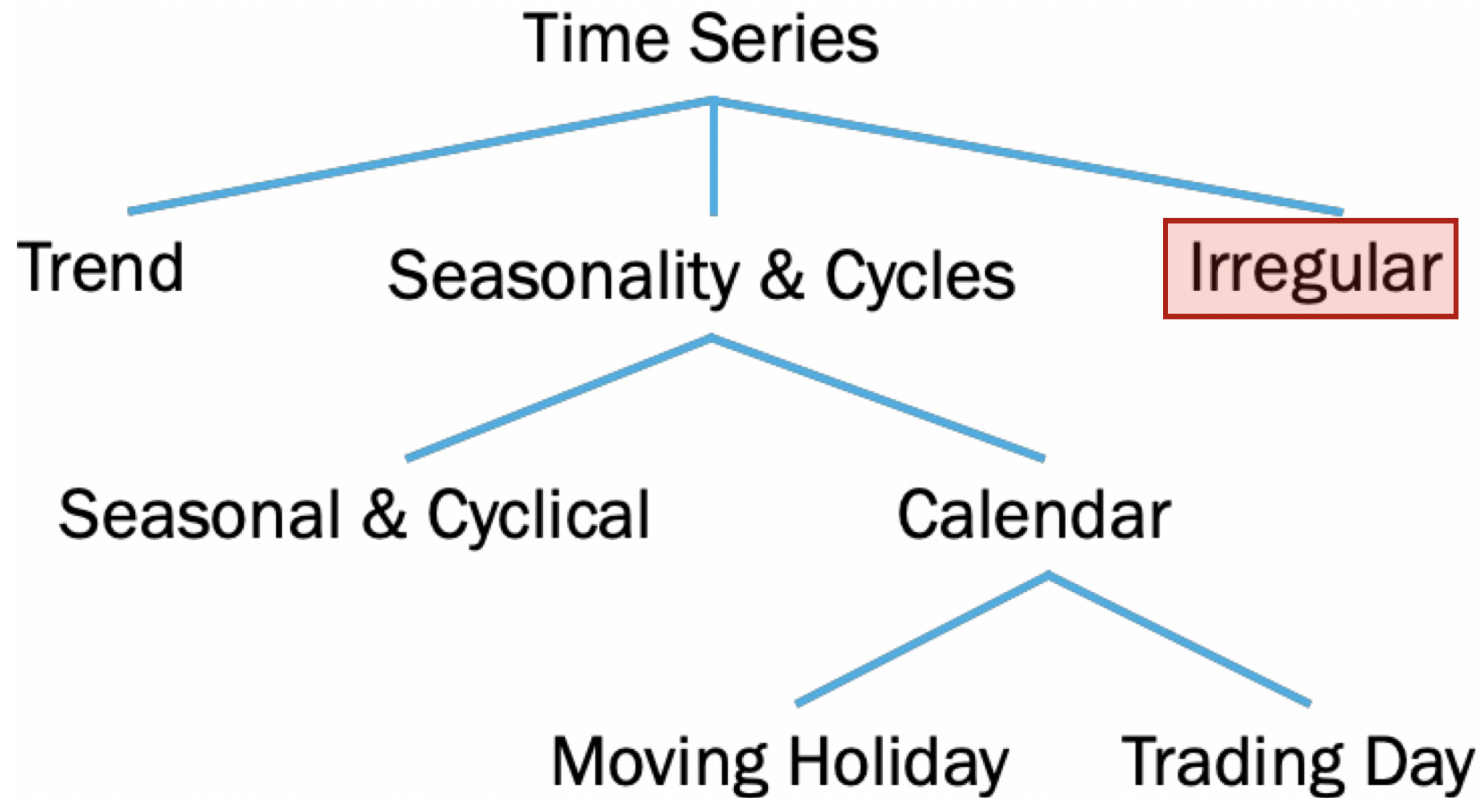


Segmt	Freq	Mean	SD	C05	C95	RI	BI	CV
<i>A</i>	3286	12.10	3.33	7.06	17.00	2.41	0.41	0.27
<i>B</i>	2594	10.09	4.43	3.88	18.20	4.69	0.80	0.44
<i>C</i>	2142	5.96	5.08	0.19	14.40	77.12	1.41	0.85

The time series of monthly fluidity indicators are then **decomposed** into:

- trend \implies **expected behaviour**
- seasonal component (seasonality, trading-day, moving-holiday) \implies **expected behaviour**
- structural breaks, \implies **explained unexpected behaviour** and
- irregular component \implies chain **volatility**

A high irregular component at a given time indicates a poor performance against expectations \implies an **anomalous observation**.



Conceptual time series decomposition (after structural breaks are removed); potential anomalous behaviour should be searched for in the **irregular component**.

In general, the decomposition follows a model which is

- multiplicative;
- additive, or
- pseudo-additive.

The choice of a model is driven by data behaviour and choice of assumptions; the X12 model automates some of the aspects of the decomposition, but manual intervention and diagnostics are still required.

IMPORTANT NOTE: anomaly detection often requires modeling choices/assumptions.

The **additive model**, for instance, assumes that:

1. the seasonal component S_t and the irregular component I_t are independent of the trend T_t ;
2. the seasonal component S_t remains stable from year to year; and
3. there is no seasonal fluctuation: $\sum_{j=1}^{12} S_{t+j} = 0$.

Mathematically, the model is expressed as:

$$O_t = T_t + S_t + I_t$$

All components share the same dimensions and units.

After seasonality adjustment, the seasonality adjusted series is:

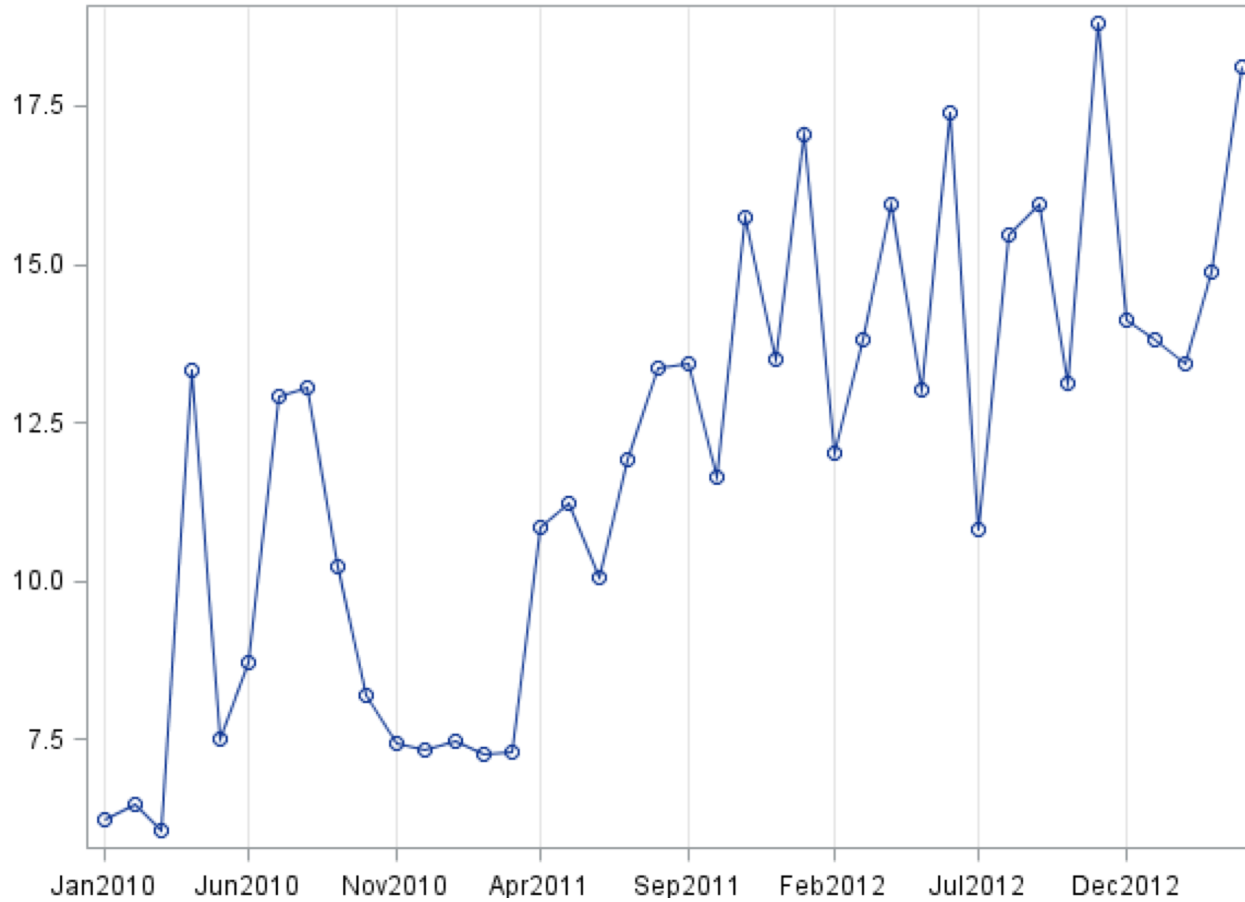
$$SA_t = O_t - S_t = T_t + I_t$$

The multiplicative and pseudo-additive models are defined in similar ways:

- if the size of S_t increases/decreases over time, use a multiplicative model;
- otherwise, use an additive model.

The data decomposition/preparation process is illustrated with the 40-month time series of marine transit CVs from 2010-2013.

The size of the peaks and troughs seems fairly constant with respect to the changing trend \implies use the additive model.

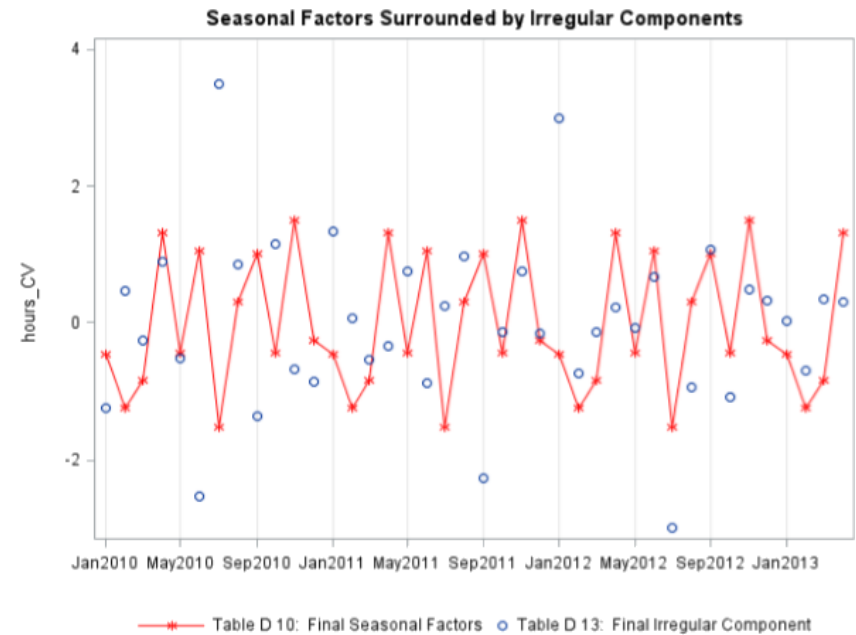


Estimation Summary	
For Variable hours_CV	
Number of Observations	40
Number of Residuals	27
Number of Parameters Estimated	3
Variance Estimate	5.6E-02
Standard Error Estimate	2.4E-01
Standard Error of Variance	1.5E-02
Log likelihood	0.4658
Transformation Adjustment	-69.5685
Adjusted Log likelihood	-69.1027
AIC	144.2053
AICC (F-corrected-AIC)	145.2488
Hannan Quinn	145.3613
BIC	148.0928

Results of Automatic Transformation Selection	
For Variable hours_CV	
AICC (with aicdiff=-2.00) prefers	No transformation
Adjustment will be	Additive

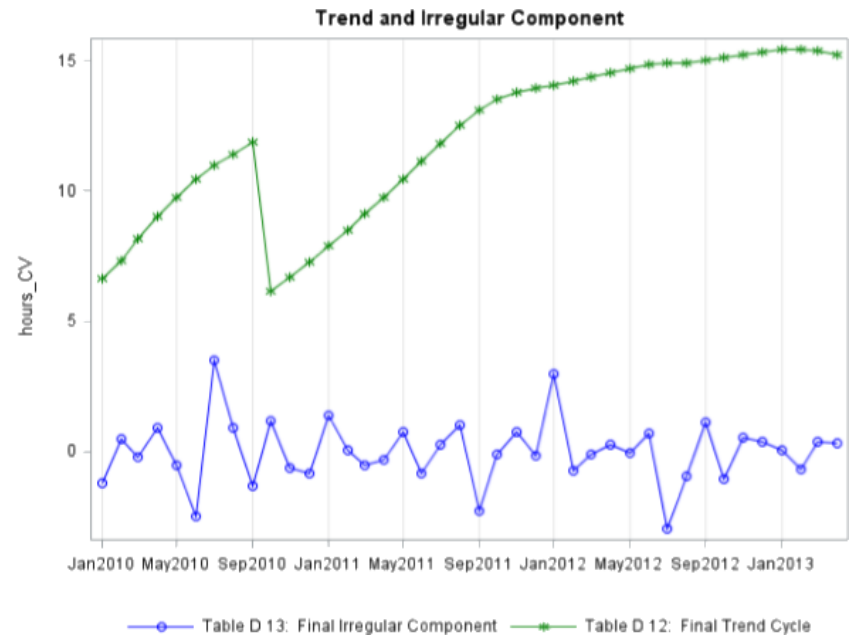
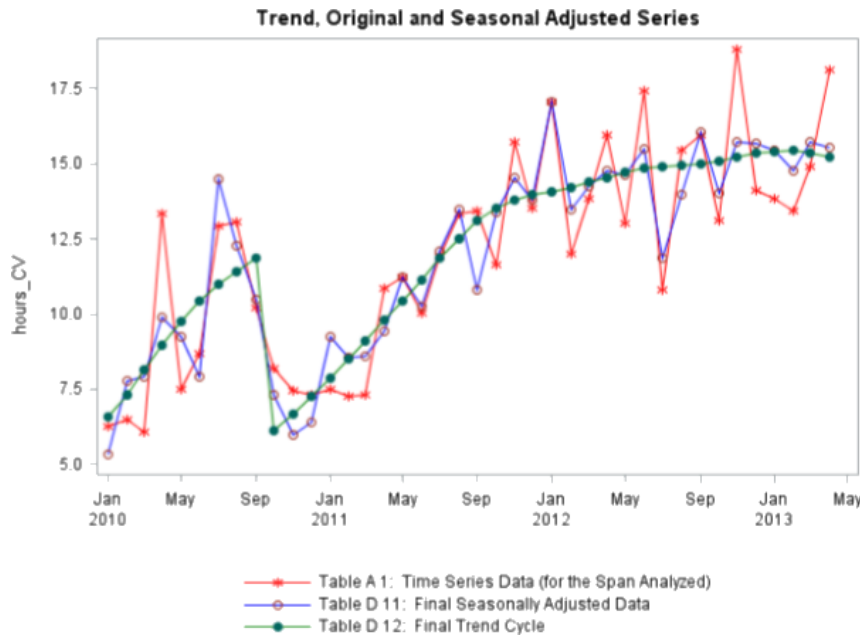
A structural break (trend level shift) is identified at OCT2010.

The SI (Seasonal Irregular) chart shows that there are more than one irregular component which exhibits volatility.



The adjusted series is shown below; the trend and irregular components are also shown separately for readability.

It is on the irregular component that detection anomaly would be conducted.



Given that the vast majority of observations in a general problem are typically “normal”, another conceptually important approach is to view anomaly detection as a:

- **rare occurrence learning** classification problem, or
- **novelty detection** data stream problem.

While there a number of strategies that use regular classification/clustering algorithms for anomaly detection, they are rarely successful unless they are **adapted** or **modified for the anomaly detection context**.

MORAL OF THE STORY: anomaly detection is a difficult problem.

Basic Concepts

Generic systems (think of the monthly transit/dwell times from supply chains) may be realized in

- **normal** states, or
- **abnormal** states.

Normality is not confined to finding the most likely state – infrequently occurring states could still be normal or plausible under some interpretation of the system.

A system's states are the results of processes or behaviours that follow certain **natural rules** and **broad principles**; the observations are a manifestation of these states.

Data allows for inferences to be made about the underlying processes, which can be tested or invalidated by the collection of additional data.

When the inputs are perturbed, the corresponding outputs are likely to be perturbed as well.

If anomalies arise from perturbed processes, the **anomaly detection problem** could be helped along by being able to identify when the underlying process is abnormal.

Supervised anomaly detection algorithms require a

1. **training set of historical labeled data** on which to build the prediction model (usually costly to obtain), and
2. testing set on which to evaluate the model's performance in terms of
 - **True Positives (TP)** – detected anomalies that actually arise from process abnormalities;
 - **True Negatives (TN)** – predicted normal observations that indeed arise from normal processes;
 - **False Positives (FP)** – detected anomalies corresponding to regular processes, and
 - **False Negatives (FN)** – predicted normal observations that are in fact the product of an abnormal process.

This is often summarized in a **confusion matrix**:

		Predicted Class	
		Normal	Anomaly
Actual Class	Normal	<i>TN</i>	<i>FP</i>
	Anomaly	<i>FN</i>	<i>TP</i>

Naïvely, one might look for an algorithm which maximizes the **accuracy**

$$a = \frac{TN + TP}{TN + TP + FN + FP}.$$

For rare occurrences, this is a losing strategy (see weapon smuggling ex.).

Better approach: try to minimize the FP rate and the FN rate under the assumption that the **cost of making a false negative error could be substantially higher than the cost of making a false positive error.**

For a testing set with $d = \text{FN} + \text{TP}$ **true outliers**, assume that an anomaly detection algorithm identifies $m = \text{FP} + \text{TP}$ **suspicious observations**, of which $n = \text{TP}$ are **known** to be true outliers.

How well did the algorithm **perform**?

Precision: proportion of true outliers among suspicious observations

$$p = \frac{n}{m} = \frac{\text{TP}}{\text{FP} + \text{TP}};$$

if most of the suspicious points are true outliers, $p \approx 1$;

Recall: proportion of true outliers detected by the algorithm

$$r = \frac{n}{d} = \frac{TP}{FN + TP};$$

if most of the true outliers are identified by the algorithm, $r \approx 1$;

F_1 –**Score:** harmonic mean of the algorithm's precision and its recall on the testing set

$$F_1 = \frac{2pr}{p + r} = \frac{2TP}{2TP + FP + FN}.$$

Question: precision, recall, and F_1 –score do not incorporate TN in the evaluation process. Is this likely to be a problem?

Example: consider a test dataset with 5000 observations, 100 of which are anomalous.

An algorithm that predicts all observations to be anomalous yields

		Predicted Class		Total	
		Normal	Anomaly		
Actual Class	Normal	0	4900	4900	Accuracy 0.02 Precision 0.02 Recall 1.00 F1-Score 0.04
	Anomaly	0	100	100	
Total		0	5000	5000	

An algorithm that only detects 10 of the true outliers and x of the normal observations yields

		Predicted Class		Total
		Normal	Anomaly	
Actual Class	Normal	x	$4900 - x$	4900
	Anomaly	90	10	100
Total		$90 + x$	$4910 - x$	5000

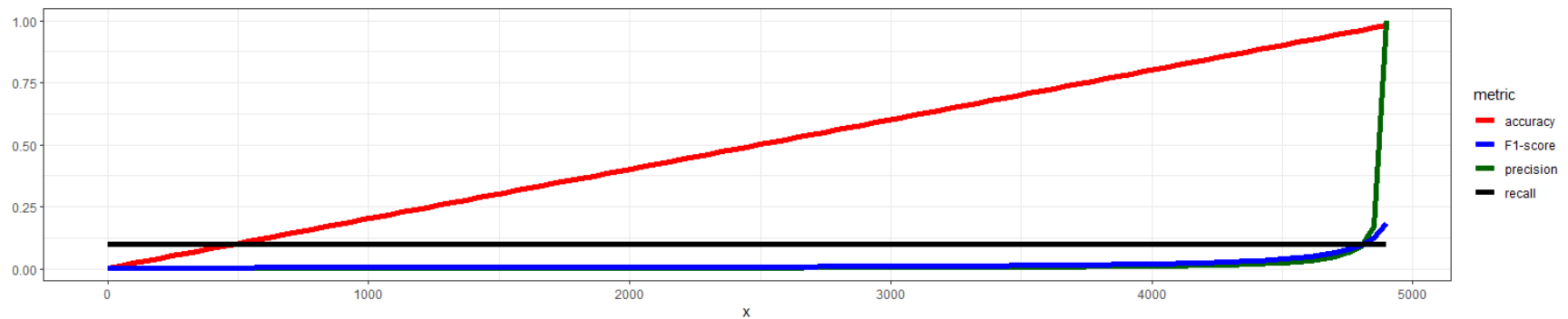
$0 \leq x \leq 4900$

Accuracy $(x + 10)/5000$

Precision $10/(4910 - x)$

Recall $1/10$

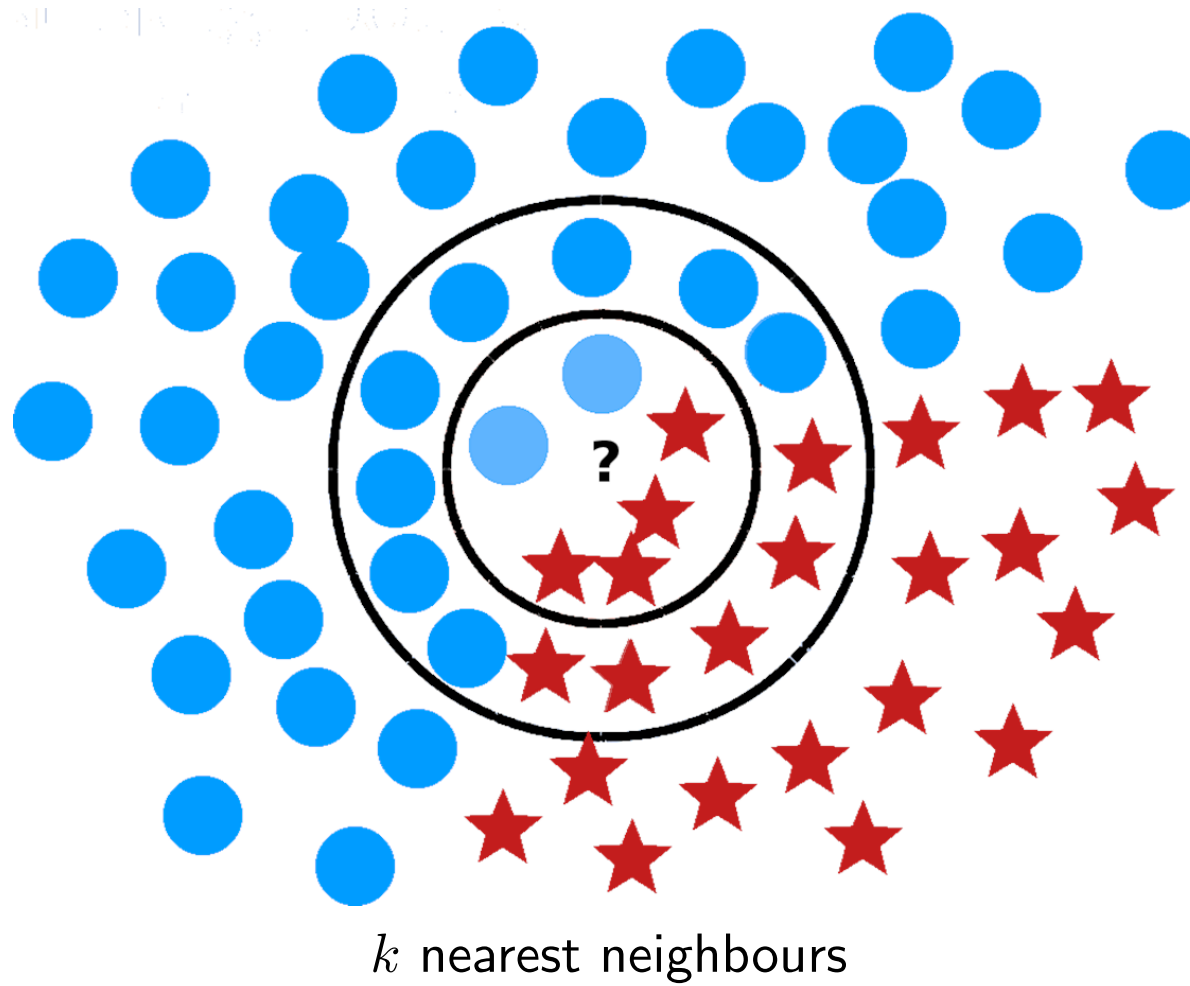
F1-Score $20/(5010 - x)$

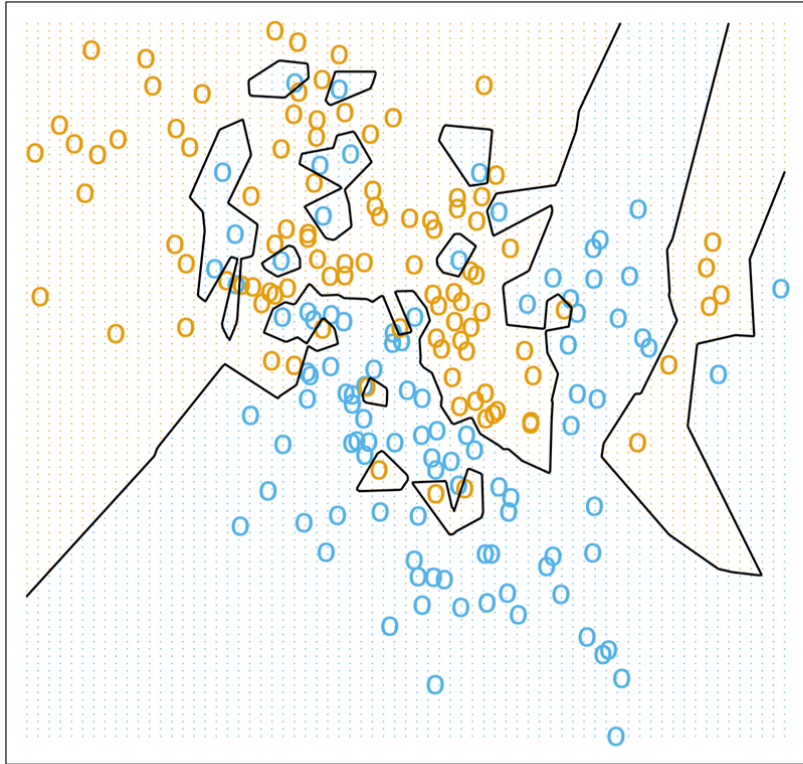


SL algorithms include:

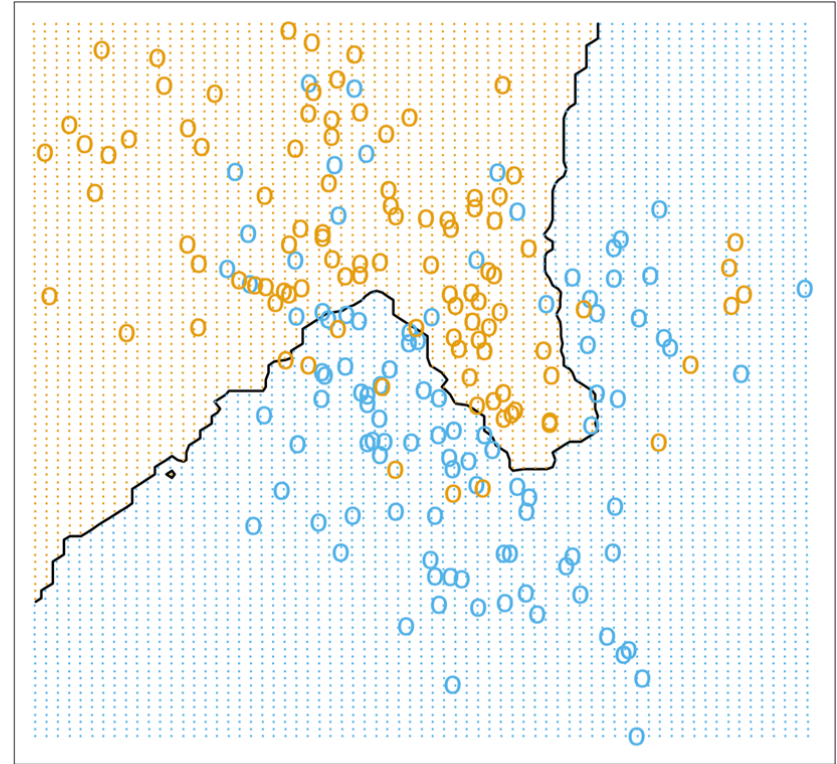
- logistic regression
- naïve or optimal Bayes classifiers
- support vector machines
- neural networks (deep learning)
- decision trees, etc.

Such algorithms incorporate (and help us learn) historical patterns and rules.



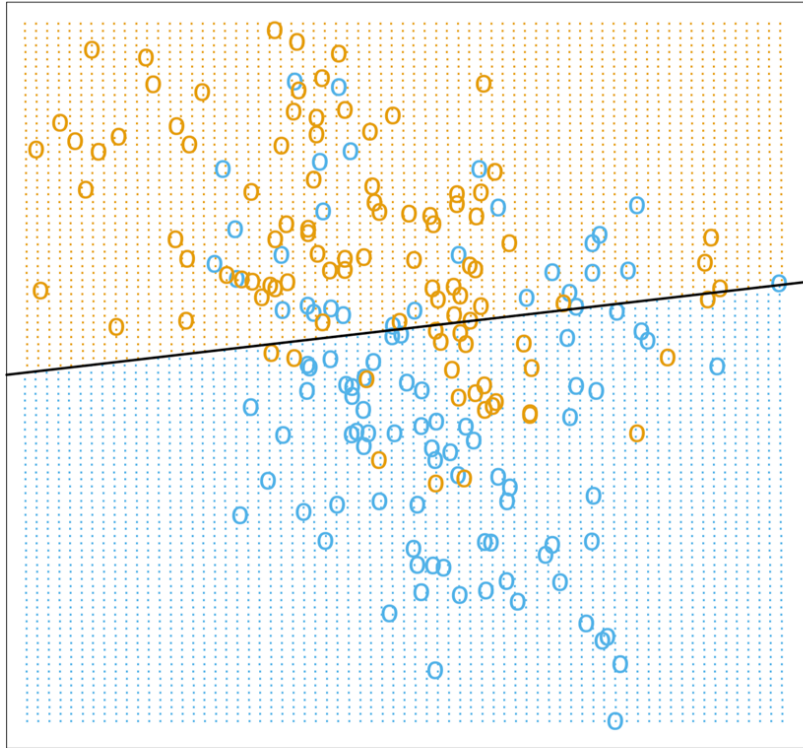


1NN Classifier

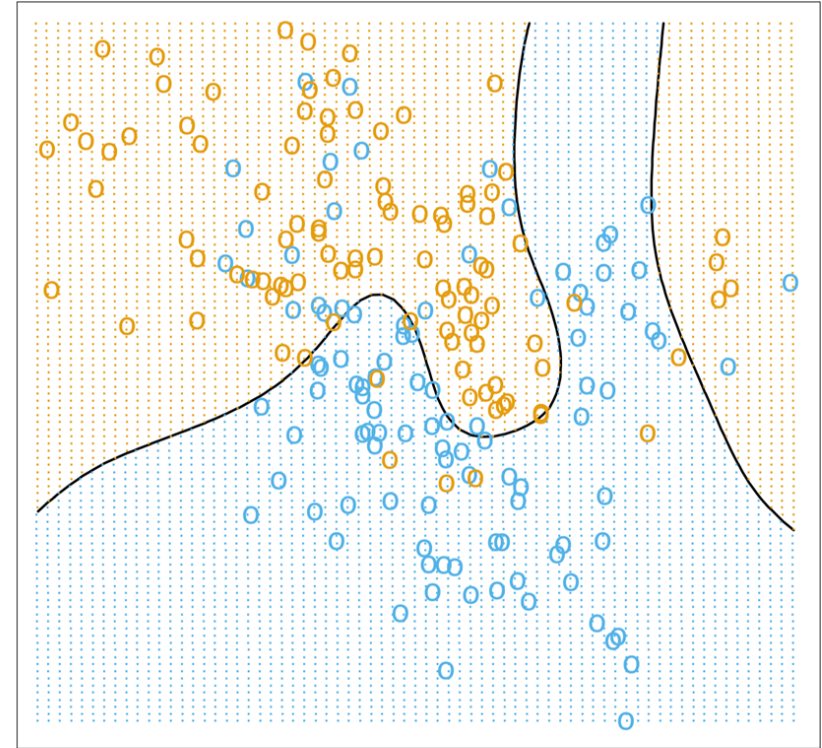


15NN Classifier

[Tibshirani, Hastie, Friedman]

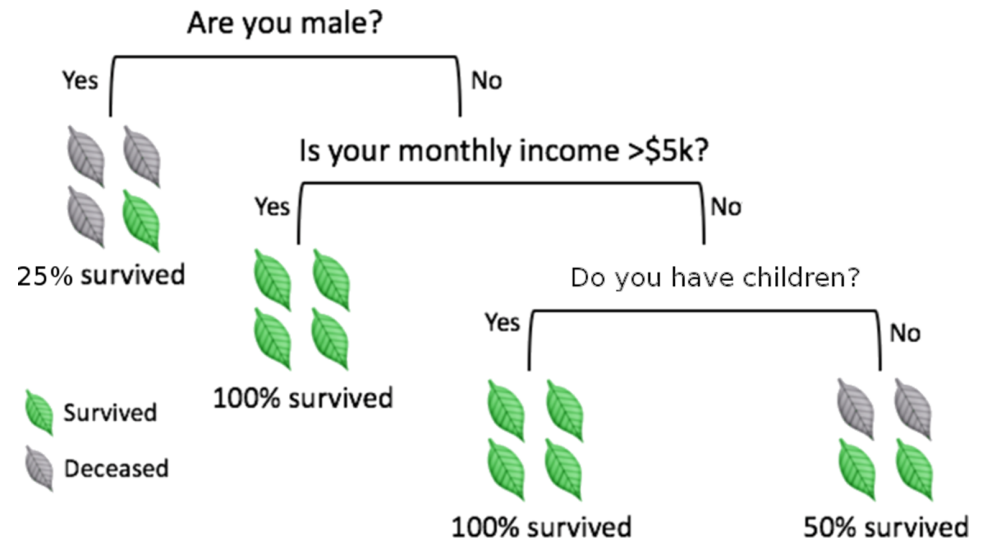
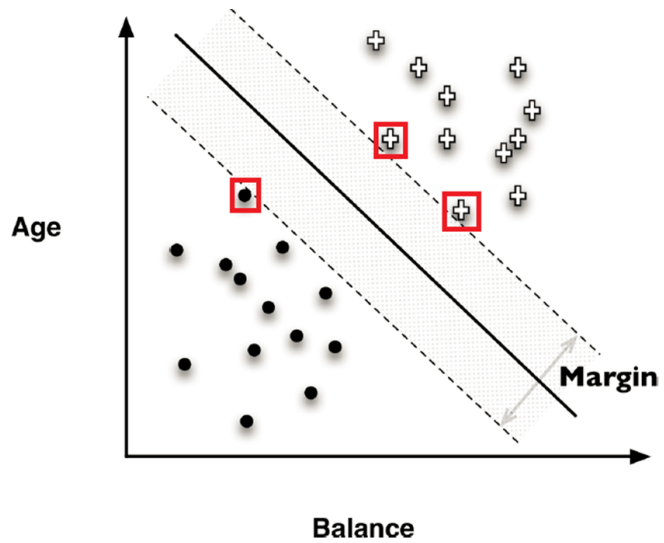


Linear Regression Classifier



Optimal Bayes Classifier

[Tibshirani, Hastie, Friedman]



Support vector machines (left), decision tree (right)
 [Foster and Provost, Ng and Soo]

Another SL approach: estimate the **relative abnormality** of observations.

Estimating the probability that an observation \mathbf{x}_1 is anomalous \implies difficult;
 Determining it is more likely to be anomalous than another $\mathbf{x}_2 \implies$ easier.
 (We write $\mathbf{x}_1 \succeq \mathbf{x}_2$.)

Let $k_i \in \{1, \dots, m\}$ be the rank of the i^{th} **true outlier**, $i \in \{1, \dots, n\}$, in the sorted list of suspicious observations

$$\mathbf{x}_1 \succeq \mathbf{x}_{k_1} \succeq \cdots \succeq \mathbf{x}_{k_i} \succeq \cdots \succeq \mathbf{x}_{k_n} \succeq \mathbf{x}_m, \quad n \leq m;$$

the **rank power** of the algorithm is

$$\text{RP} = \frac{n(n+1)}{2 \sum_{i=1}^n k_i}.$$

When the n actual anomalies are ranked in (or near) the top n suspicious observations, we have

$$\sum_{i=1}^n k_i \approx \sum_{i=1}^n i = \frac{n(n+1)}{2} \implies \text{RP} \approx 1.$$

As with most performance evaluation metrics, a single raw number is meaningless – it needs to be compared to other algorithms.

Other SL performance evaluation metrics include:

- **AUC** – the probability of ranking a randomly chosen anomaly higher than a randomly chosen normal observation (higher is better);
- **probabilistic AUC** – a calibrated version of AUC.

The **rare occurrence** problem can be tackled by using:

- a **manipulated training set** (oversampling, undersampling, generating artificial instances);
- **specific SL AD algorithms** (CREDOS, PN, SHRINK);
- **boosting algorithms** (SMOTEBoost, RareBoost);
- **cost-sensitive classifiers** (MetaCost, AdaCost, CSB, SSTBoost),
- etc.

See A.Lazarević et al. [2004], *Data Mining for Analysis of Rare Events: A Case Study in Security, Financial and Medical Applications* for more information.

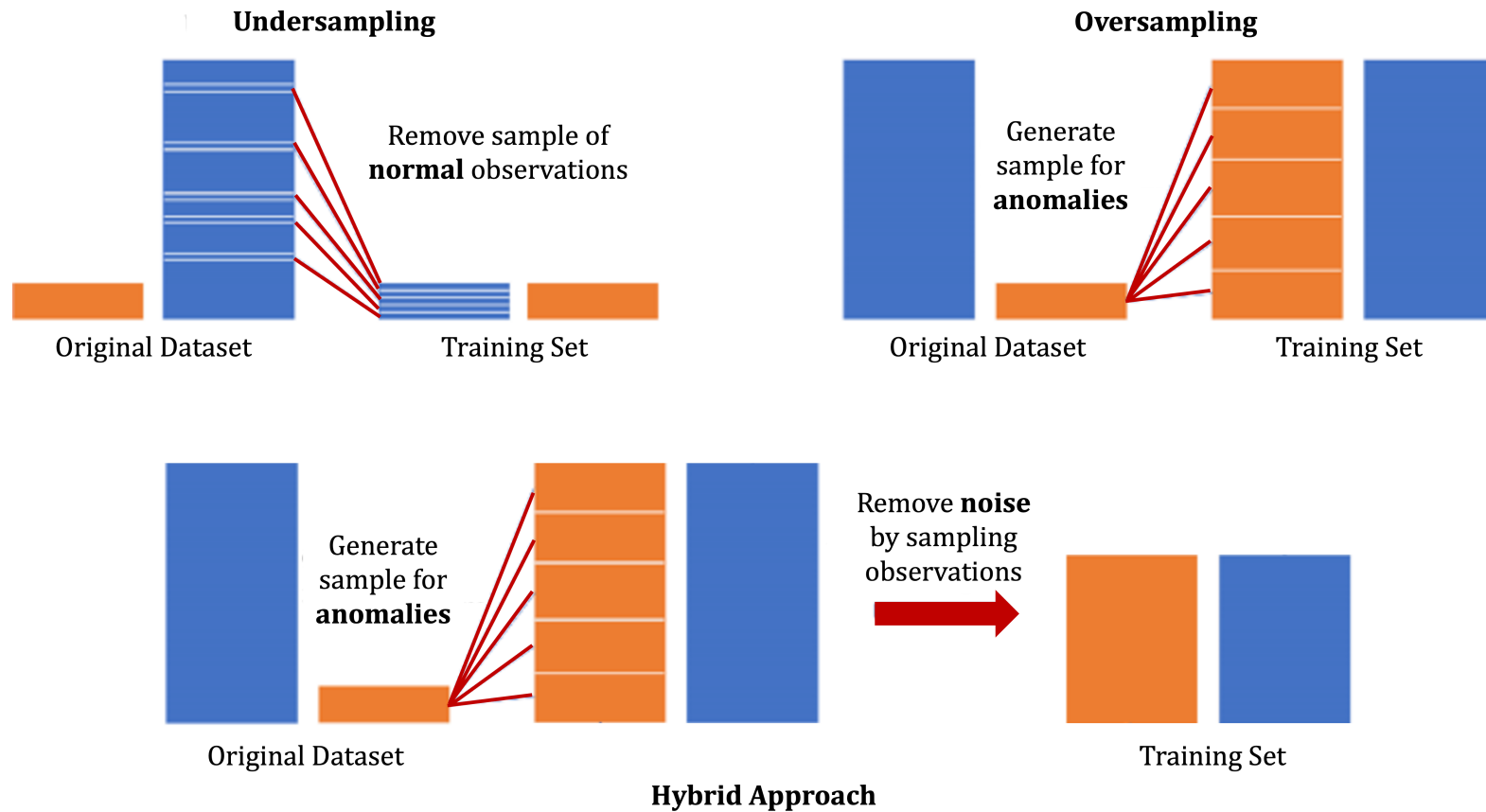
The rare (anomalous) class can be **oversampled** by duplicating the rare events until the data set is **balanced** (roughly the same number of anomalies and normal observations).

This does not increase the overall level of information, but it will increase the mis-classification cost.

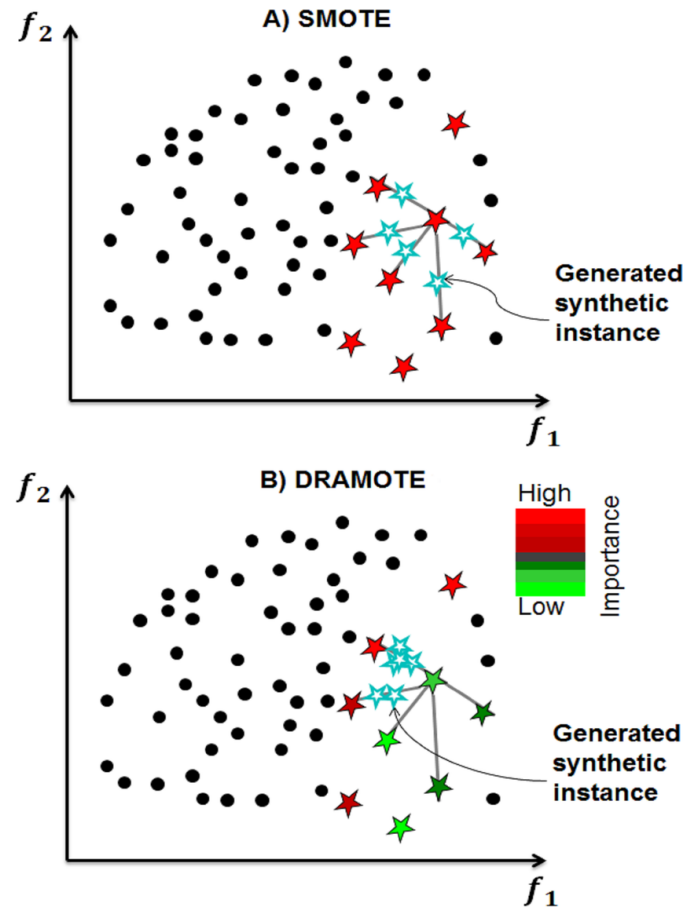
The majority class (normal observations) can also be **undersampled** by randomly removing:

- “near miss” observations or
- or observations far from anomalous observations.

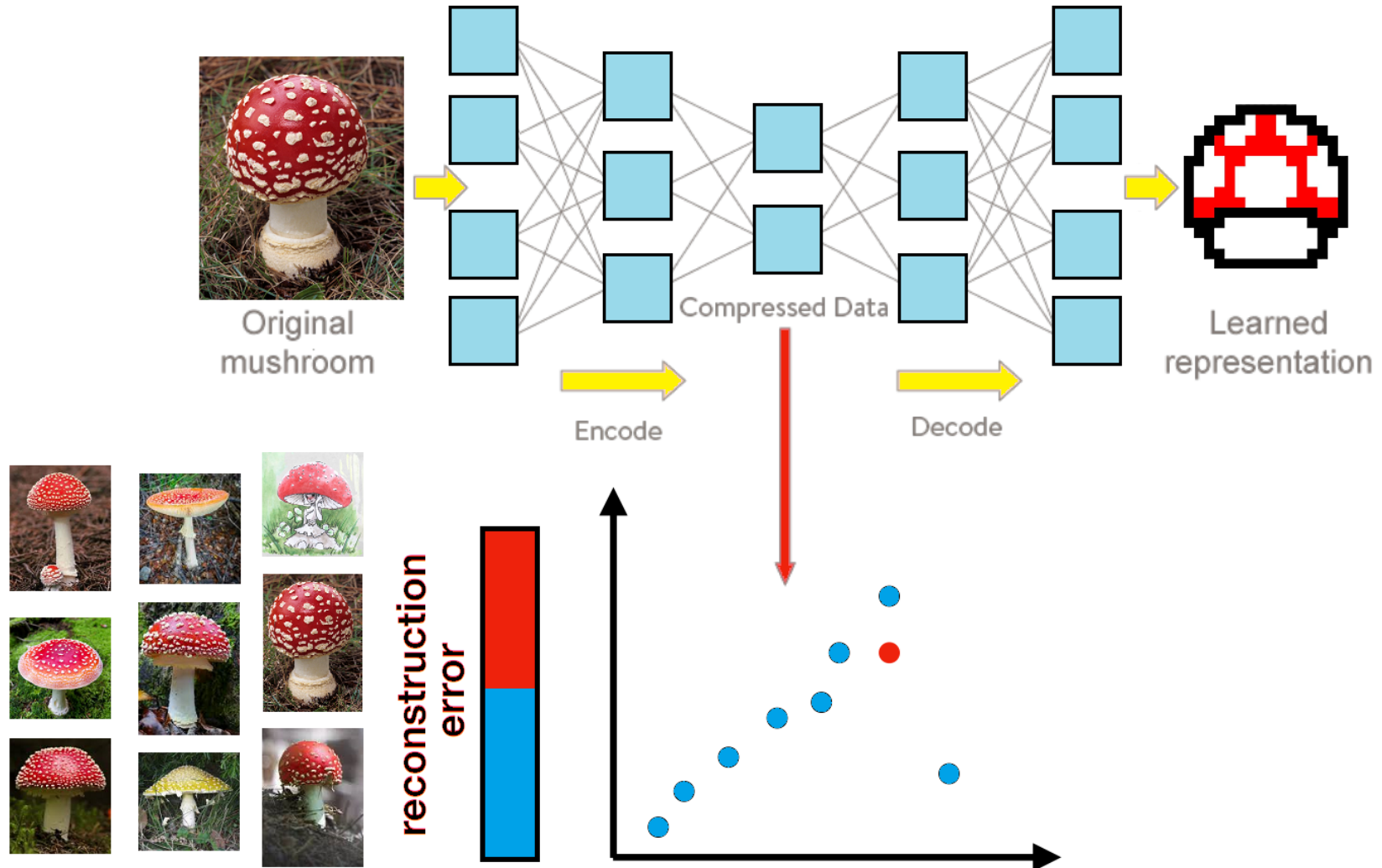
Some loss of information has to be expected (and overly general rules).



[Le et al., A Hybrid Approach Using Oversampling Technique and Cost-Sensitive Learning for Bankruptcy Prediction]



Generating artificial cases with SMOTE and DRAMOTE [Soufan, et al.]



Autoencoders learn a compressed representation of the data (dimension reduction).

The **reconstruction error** measures (in a sense) how much information is lost in the compression.

Anomaly detection algorithms can be applied to the compressed data:

- look for anomalous patterns, and/or
- anomalous reconstruction errors.

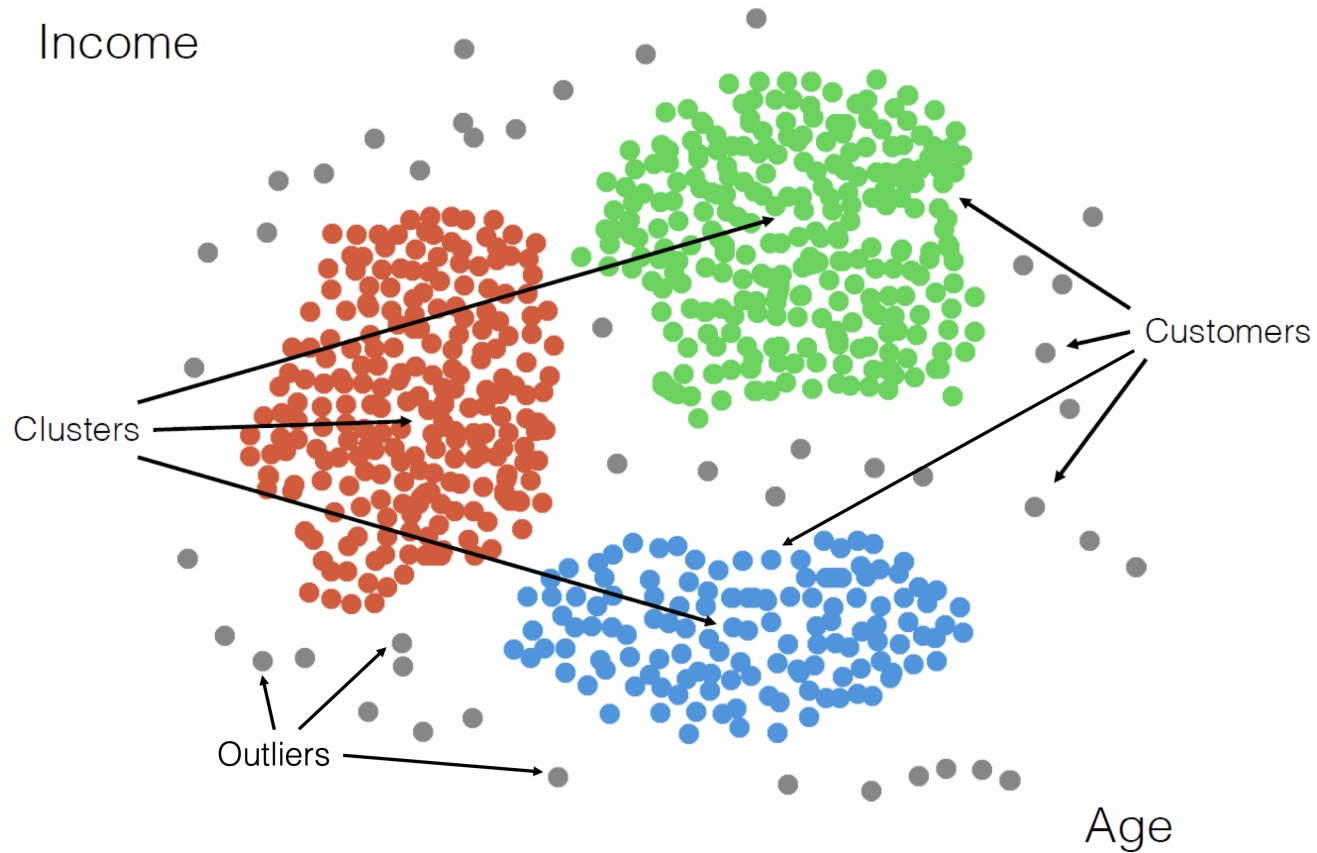
[Previous slide adapted from Baron].

On the **unsupervised** front, anomalous/normal labels are not used:

- anomalies are those observations that are dissimilar to other observations;
- **clusters** are groupings of similar observations, so
- observations without a natural cluster fit are potential anomalies.

Challenges:

- most clustering algorithms do not recognize potential outliers (DBSCAN and variants are exceptions), and
- finding an appropriate measure of similarity/dissimilarity of observations is difficult (different measures often lead to different cluster assignments).



Clusters of regular customers (red, green, blue) and potential anomalies/outliers (grey) in an artificial dataset.

5.2 – Quantitative Methods of Anomaly Detection

Cluster-based methods are not the only types of UL anomaly detection methods.

- **Distance-based methods:** distance to all points, distance to k nearest neighbours (k NN), average distance to k NN, median distance to k NN, etc.
- **Density-based methods:** local outlier factor (LOF), isolation forest, HDBSCAN, etc.

5.2.1 – Distance-Based Methods

We find anomalous observations by comparing them to other observations (**anomalies are relative, not absolute**).

In the **distance-based context**, the natural way to compare observations is to consider their **distance from a subset of observations**: increasing distance being increasingly **suggestive** of anomalous status.

Requirement: a **distance function** or a **pre-computed table of pair-wise distances** (in discrete case).

The choice of subsets and distance functions distinguish the different distance-based algorithms.

Notation

- $D \subset \mathbb{R}^n$ is an n -dimensional (numerical) data set
- $\mathbf{p}, \mathbf{q} \in D$ are specific observations in D
- $P \subset D$ is a subset of D
- $d : D \times D \rightarrow \mathbb{R}$ is a distance function on $D \subset \mathbb{R}^n$
- the distance between \mathbf{p} and \mathbf{q} is written $d(\mathbf{p}, \mathbf{q})$

- the output of an anomaly detection algorithm is a function $a : D \rightarrow \mathbb{R}$
- $a(\mathbf{p})$ is a number that describes how anomalous \mathbf{p} is
- if $a(\mathbf{p}) < a(\mathbf{q})$ for $\mathbf{p}, \mathbf{q} \in D$, then \mathbf{p} is **less anomalous** than \mathbf{q}
- $\alpha \in \mathbb{R}$ is the **absolute anomaly threshold**
- any $\mathbf{p} \in D$ for which $a(\mathbf{p}) > \alpha$ is **absolutely anomalous**

Similarity Measures

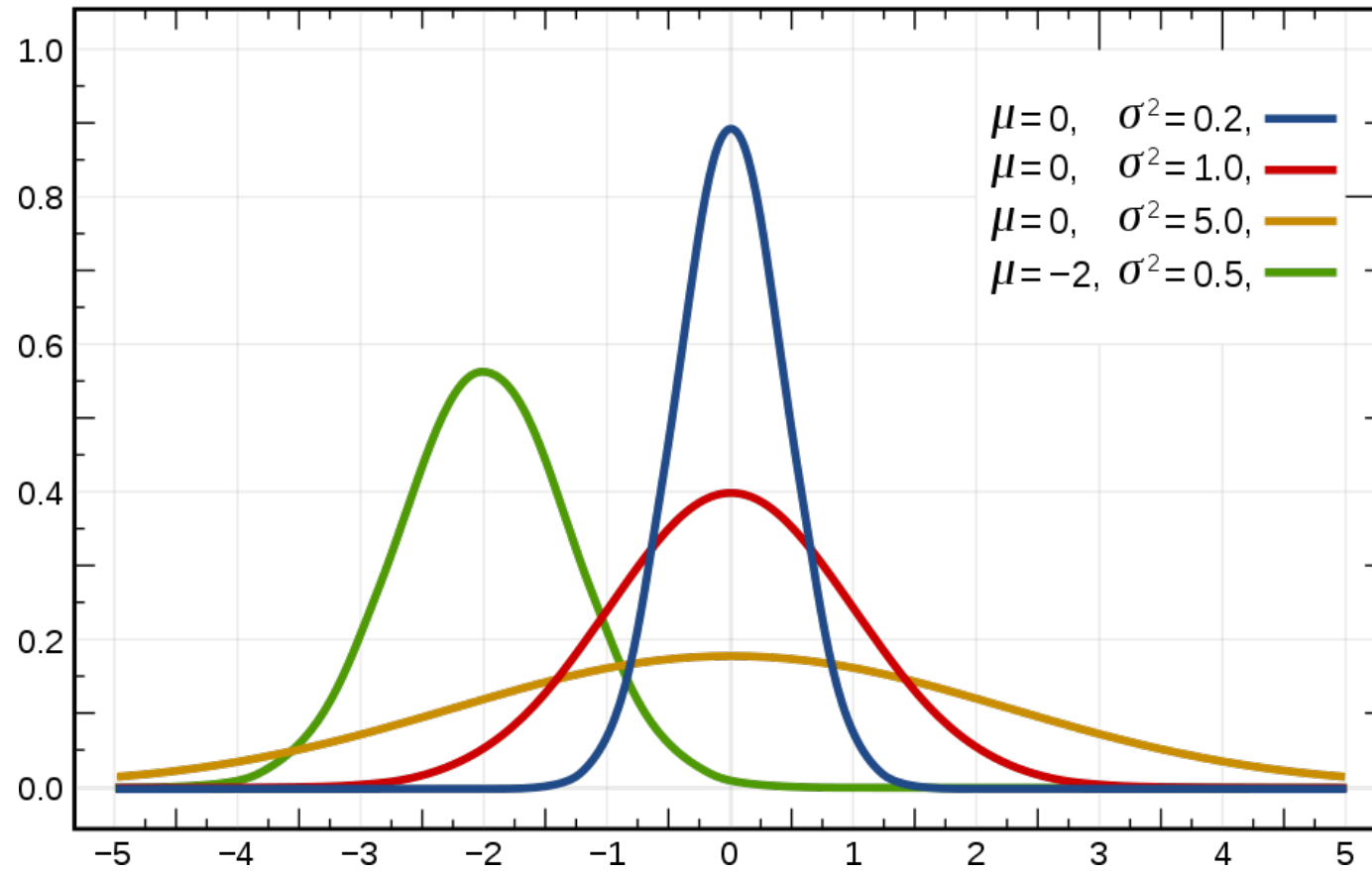
A **similarity measure** is a real-valued function that describes the **similarity between two objects**.

A common construction for the similarity w between two points \mathbf{p}, \mathbf{q} :

$$w(\mathbf{p}, \mathbf{q}) = \frac{1}{1 + d(\mathbf{p}, \mathbf{q})}, \quad \text{for some distance } d.$$

Note: $w \rightarrow 1$ as $d \rightarrow 0$, and $w \rightarrow 0$ as $d \rightarrow \infty$.

Similarity measures can also be constructed between **probability distributions** (see Hellinger distance).



We can think of a single point \mathbf{p} as a probability distribution (with 0% chance of drawing another point).

The distance between that point and any other distribution with mean μ and covariance matrix Σ can be given using the **Mahalanobis framework**:

$$M(\mathbf{p}) = \sqrt{(\mathbf{p} - \mu)^\top \Sigma^{-1} (\mathbf{p} - \mu)} \quad (\text{BACON}).$$

Alternatively, if \mathbf{p} and \mathbf{q} are drawn from the same distribution with covariance Σ , then the Mahalanobis distance is a dissimilarity measure between \mathbf{p} and \mathbf{q} :

$$d_M(\mathbf{p}, \mathbf{q}) = \sqrt{(\mathbf{p} - \mathbf{q})^\top \Sigma^{-1} (\mathbf{p} - \mathbf{q})}.$$

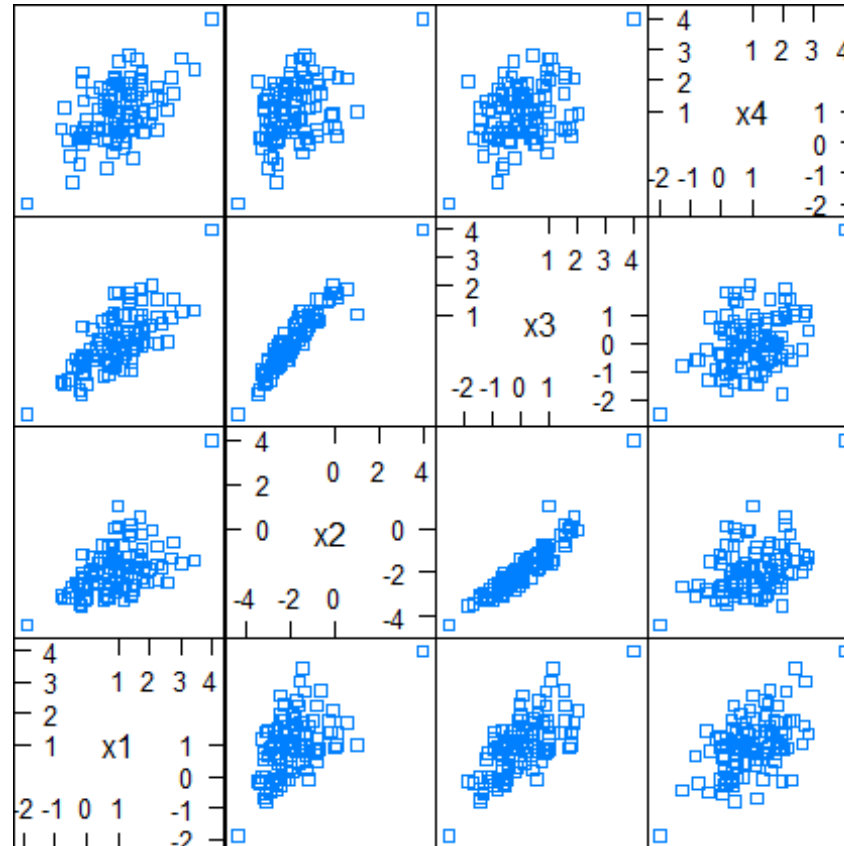
Example: consider a 4D-dataset drawn from a multivariate $\mathcal{N}(\mu, \Sigma)$ with

$$\mu = (1, -2, 0, 1), \quad \Sigma = \begin{pmatrix} 1 & 0.5 & 0.7 & 0.5 \\ 0.5 & 1 & 0.95 & 0.3 \\ 0.7 & 0.95 & 1 & 0.3 \\ 0.5 & 0.3 & 0.3 & 1 \end{pmatrix}.$$

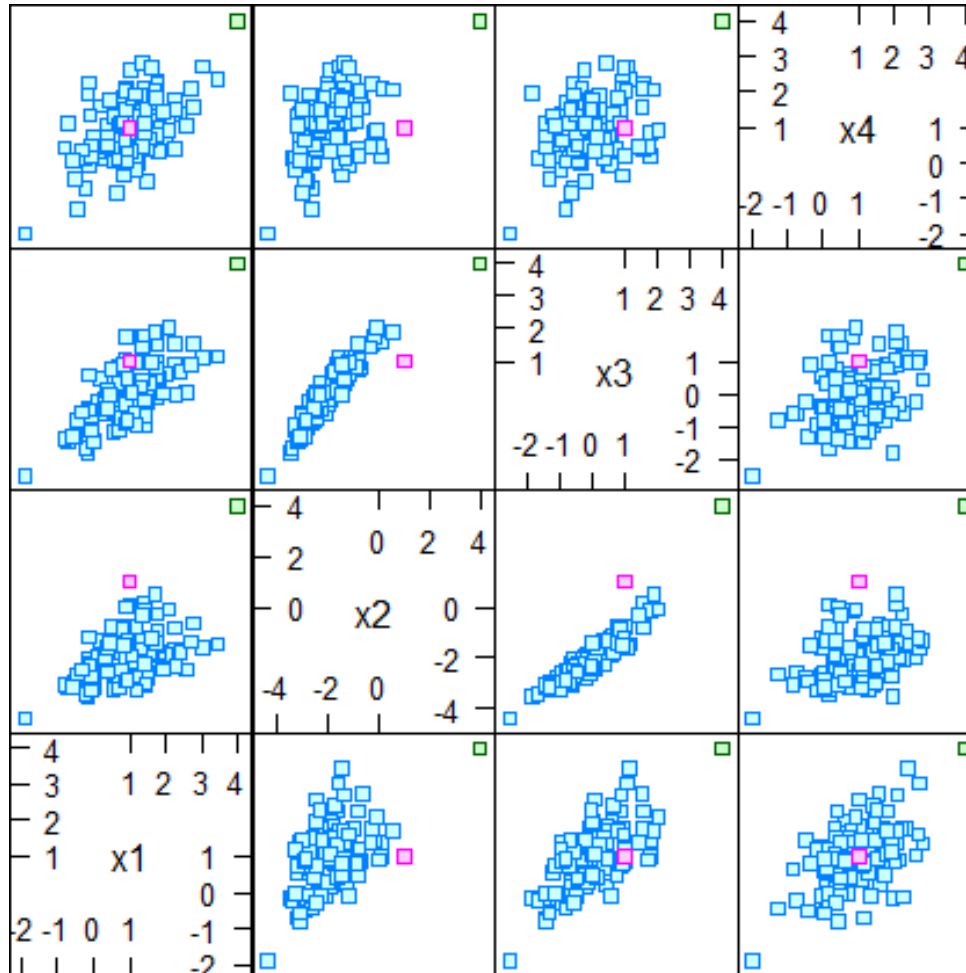
100 observations \mathbf{p}_1 to \mathbf{p}_{100} are “normal”:

stat	x_1	x_2	x_3	x_4
min	-1.9049	-4.4113	-2.5324	-1.9949
Q_1	0.3812	-2.6464	-0.6190	0.3361
med	0.9273	-2.0220	-0.0506	0.9381
avg	0.9374	-1.9788	0.0071	0.9438
Q_3	1.4615	-1.4002	0.6296	1.5906
max	3.4414	0.5223	2.0265	2.8073

2 observations are “anomalous”: $\mathbf{z}_1 = (1, 1, 1, 1)$, $\mathbf{z}_4 = (4, 4, 4, 4)$.



Visually, it seems there might be 3 outliers.



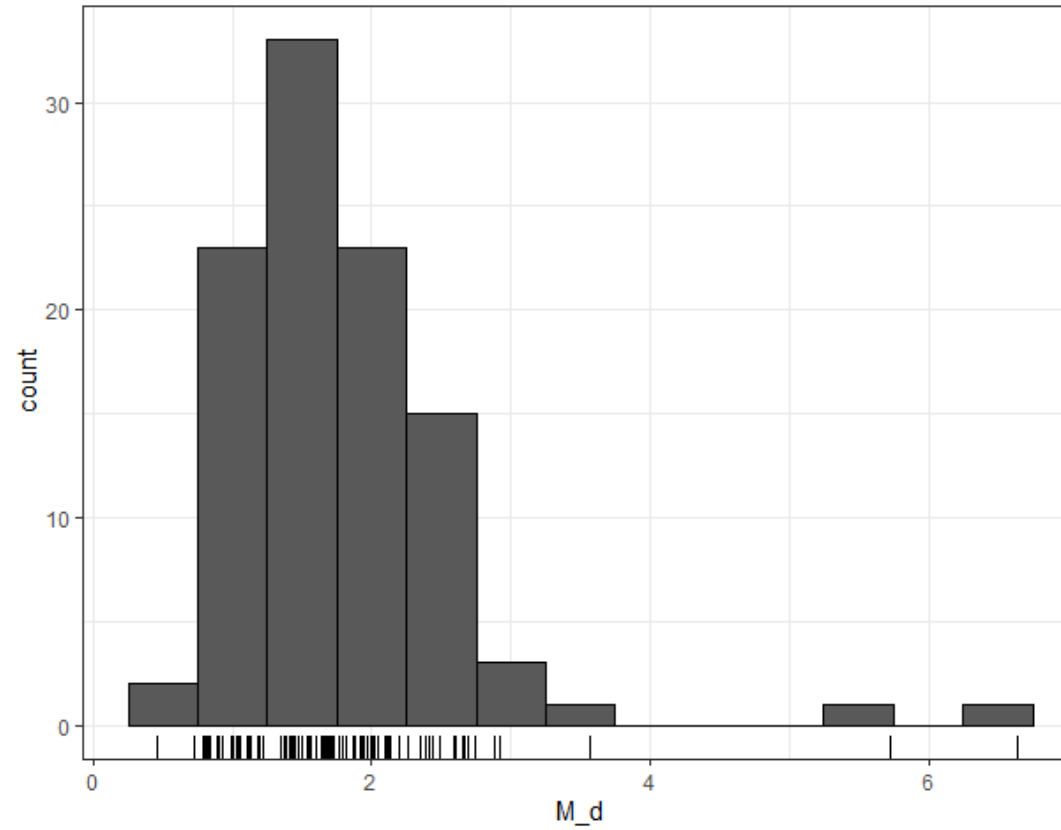
In general, the mean vector and the covariance structure must be estimated from the data:

$$\hat{\mu} = (0.968, -1.891, 0.056, 0.974), \quad \hat{\Sigma} = \begin{pmatrix} 0.900 & 0.569 & 0.665 & 0.503 \\ 0.569 & 1.312 & 1.069 & 0.469 \\ 0.665 & 1.069 & 0.992 & 0.397 \\ 0.503 & 0.469 & 0.397 & 0.904 \end{pmatrix}.$$

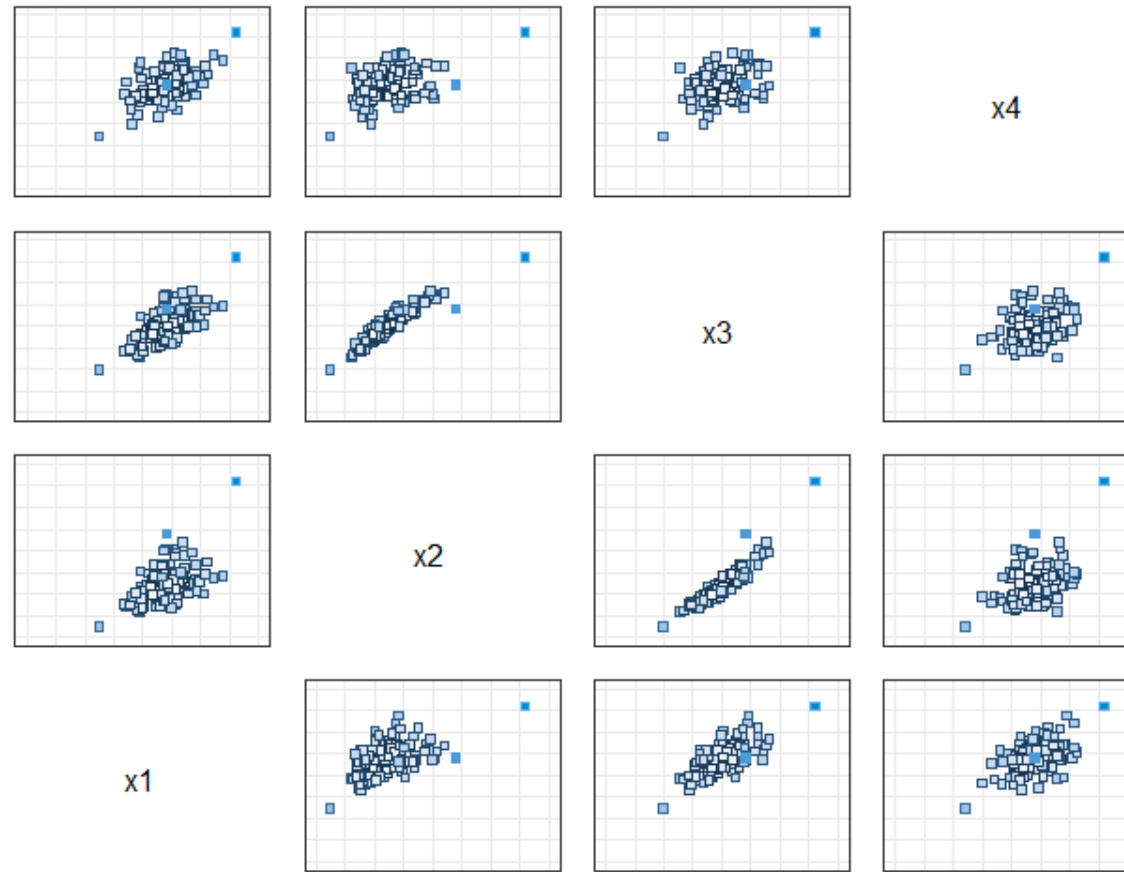
These are distinct from μ and Σ , but close enough to be explained by

- sampling variation
- $\mathbf{z}_1, \mathbf{z}_4 \not\sim \mathcal{N}(\mu, \Sigma)$

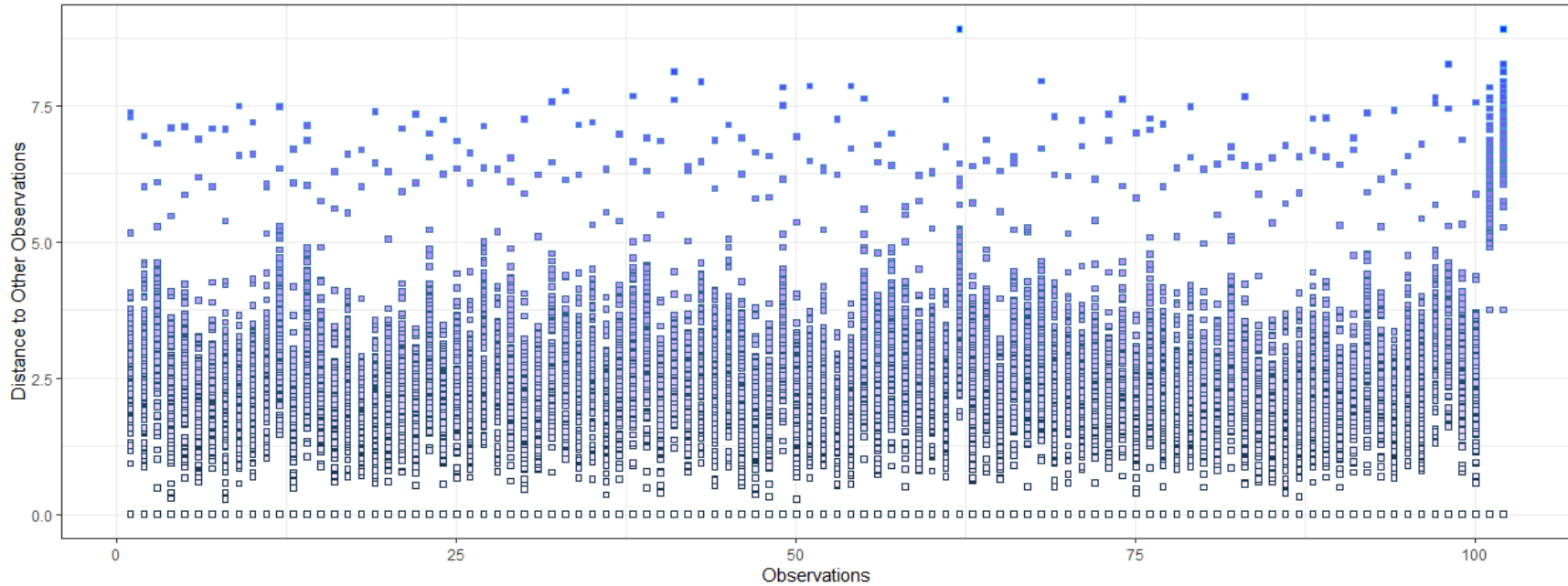
To identify anomalous observations, compute the Mahalanobis distance from all points to the empirical distribution, and between all pairs.



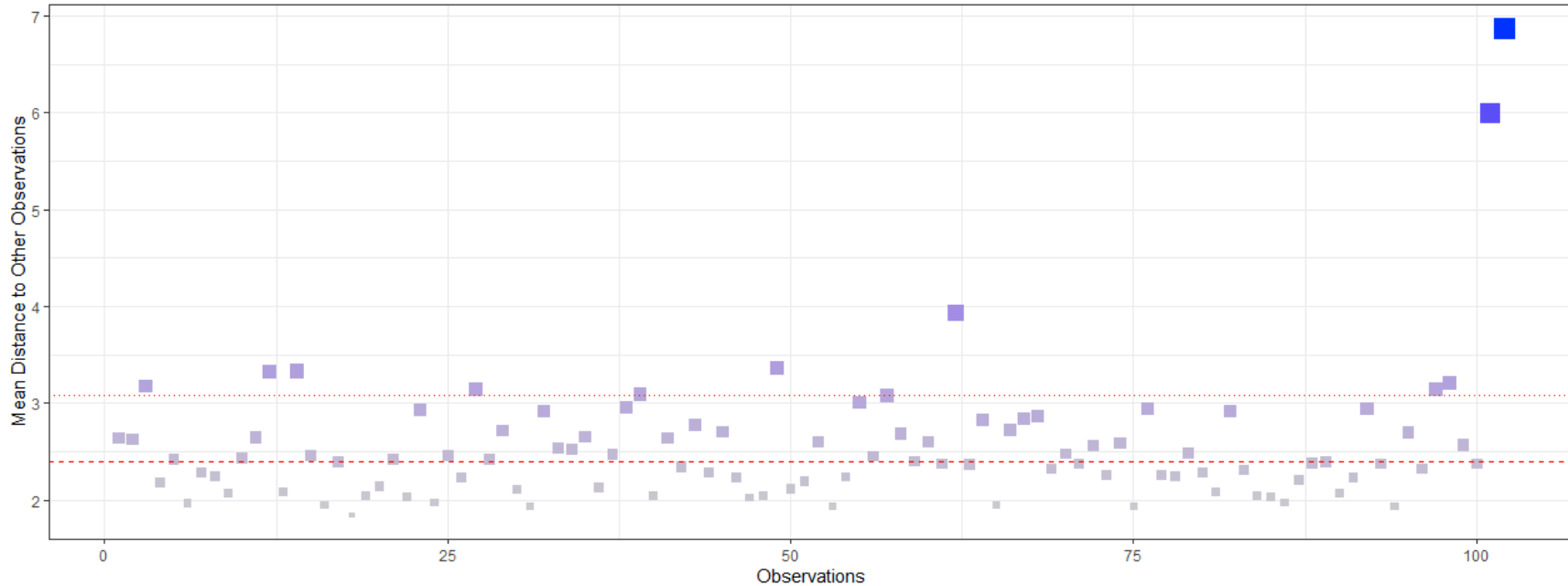
Histogram of Mahalanobis distances to empirical distribution.



Scatter plot of Mahalanobis distances to empirical distribution.



Mahalanobis distances between each pair (empirical distribution). Notice observations 101 and 102, as well as the diffuse cloud of points above the value 5.0.



Mean Mahalanobis distances between each pair (empirical distribution). Notice observations 101 and 102 again. The red lines represent the median mean distance, and 1 standard deviation the median mean distance. The Mahalanobis framework seems to identify 2 outliers.

If Σ is diagonal, then

$$d_M(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^n \frac{(p_i - q_i)^2}{\sigma_i^2}},$$

where σ_i^2 is the variance along the i -th dimension.

If Σ is the identity matrix, then we recover the **Euclidean distance**

$$d_2(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}.$$

In an anomaly detection context, a **linear normalization** is usually applied to each dimension so that each entry lies in the hypercube $[-1, 1]^n$.

The **Minkowski distance** of order p is a generalization of the Euclidean distance:

$$d_p(\mathbf{p}, \mathbf{q}) = \left(\sum_{i=1}^n |p_i - q_i|^p \right)^{1/p}.$$

- For $p = 1$, we recover the **Manhattan distance**:

$$d_1(\mathbf{p}, \mathbf{q}) = \sum_{i=1}^n |p_i - q_i|;$$

- for $p = \infty$, we recover the **supremum (Chebychev) distance**

$$d_\infty(\mathbf{p}, \mathbf{q}) = \max_{i=1}^n \{|p_i - q_i|\}.$$

The Minkowski distance d_p is only an actual distance function (a **metric**) when $p \geq 1$, but an exception is made for

$$d_{-\infty}(\mathbf{p}, \mathbf{q}) = \min_{i=1}^n \{|p_i - q_i|\}.$$

When working with categorical data (such as in one-hot encoding of text), it can be useful to use distances for binary vectors.

Let $\mathbf{p}, \mathbf{q} \in \{0, 1\}^n$.

The **Hamming distance** between \mathbf{p} and \mathbf{q} counts the number of positions where they differ:

$$d_H(\mathbf{p}, \mathbf{q}) = \sum_{i=1}^n |p_i - q_i|.$$

The **Jaccard similarity** of two datasets P and Q , is defined as the size of their intersection divided by the size of their union

$$J(P, Q) = \frac{|P \cap Q|}{|P \cup Q|} = \frac{|P \cap Q|}{|P| + |Q| - |P \cap Q|}$$

Their **Jaccard distance** is $d_J(P, Q) = 1 - J(P, Q)$. This can be extended to binary vectors \mathbf{p} and \mathbf{q} .

Consider an arbitrary set $D = \{x_1, x_2, \dots, x_n\}$. We build P as follows: if $p_i = 1$ then $x_i \in P$; otherwise $x_i \notin P$. Similarly for Q .

Then $|P| = \sum p_i$, $|Q| = \sum q_i$, $|P \cap Q| = \sum p_i q_i = \mathbf{p} \cdot \mathbf{q}$ and

$$d_J(\mathbf{p}, \mathbf{q}) = d_J(P, Q) = 1 - J(P, Q) = 1 - \frac{\mathbf{p} \cdot \mathbf{q}}{\sum (p_i + q_i) - \mathbf{p} \cdot \mathbf{q}}.$$

Finally, let $\mathbf{p}, \mathbf{q} \neq \mathbf{0}$. Recall that $\mathbf{p} \cdot \mathbf{q} = \|\mathbf{p}\| \|\mathbf{q}\| \cos \theta$, where θ is the angle between \mathbf{p} and \mathbf{q} .

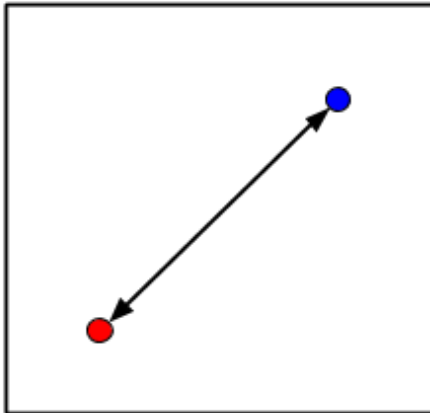
The **cosine similarity** between \mathbf{p} and \mathbf{q} is

$$\cos \theta = \frac{\mathbf{p} \cdot \mathbf{q}}{\|\mathbf{p}\| \|\mathbf{q}\|} = \frac{\sum_{i=1}^n p_i q_i}{\sqrt{\sum_{i=1}^n p_i^2} \sqrt{\sum_{i=1}^n q_i^2}}.$$

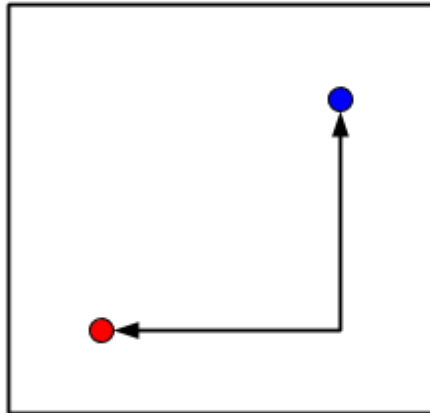
This also holds \mathbf{p}, \mathbf{q} are non-binary. The value ranges between 1 and -1 :

- $\cos \theta = 1$ when $\mathbf{p} = \mathbf{q}$;
- $\cos \theta = -1$ when $\mathbf{p} = -\mathbf{q}$, and
- $\cos \theta = 0$ when \mathbf{p} and \mathbf{q} are perpendicular.

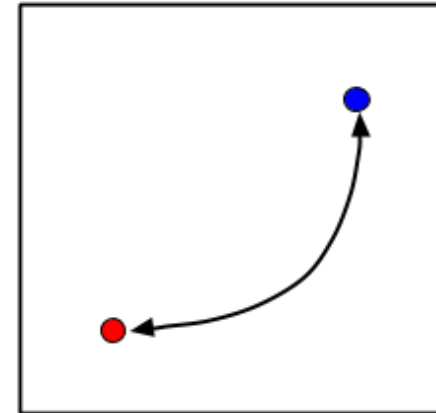
Euclidean



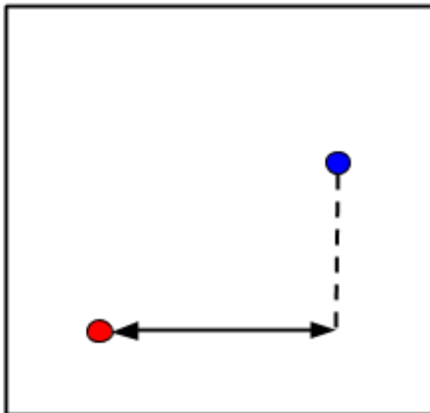
Manhattan



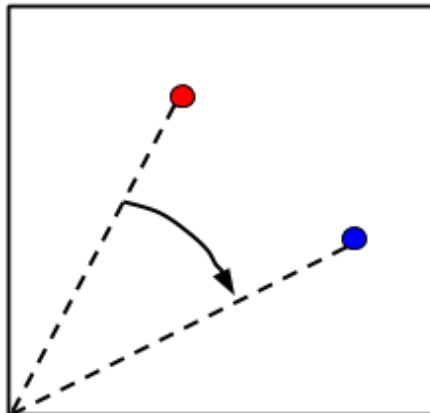
Minkowski



Chebychev



Cosine Similarity



Hamming



Distance-Based Approaches

Finding the right distance function to use for anomaly detection is **NOT AN EASY TASK** – contextual understanding and domain expertise are required.

Any such distance function can be used as the basis for anomaly detection algorithms (the ideas can also be extended to more complex algorithms).

Given some distance function d , dataset D , and integers $k, \nu \leq |D|$, the **distance to all points** (DTAP) anomaly detection algorithm considers each point \mathbf{p} in D and adds the distance from \mathbf{p} to every other point in D :

$$a(\mathbf{p}) = \sum_{\mathbf{q} \neq \mathbf{p} \in D} d(\mathbf{q}, \mathbf{p}).$$

The ν points with largest values for a are then said to be **anomalous according to a** .

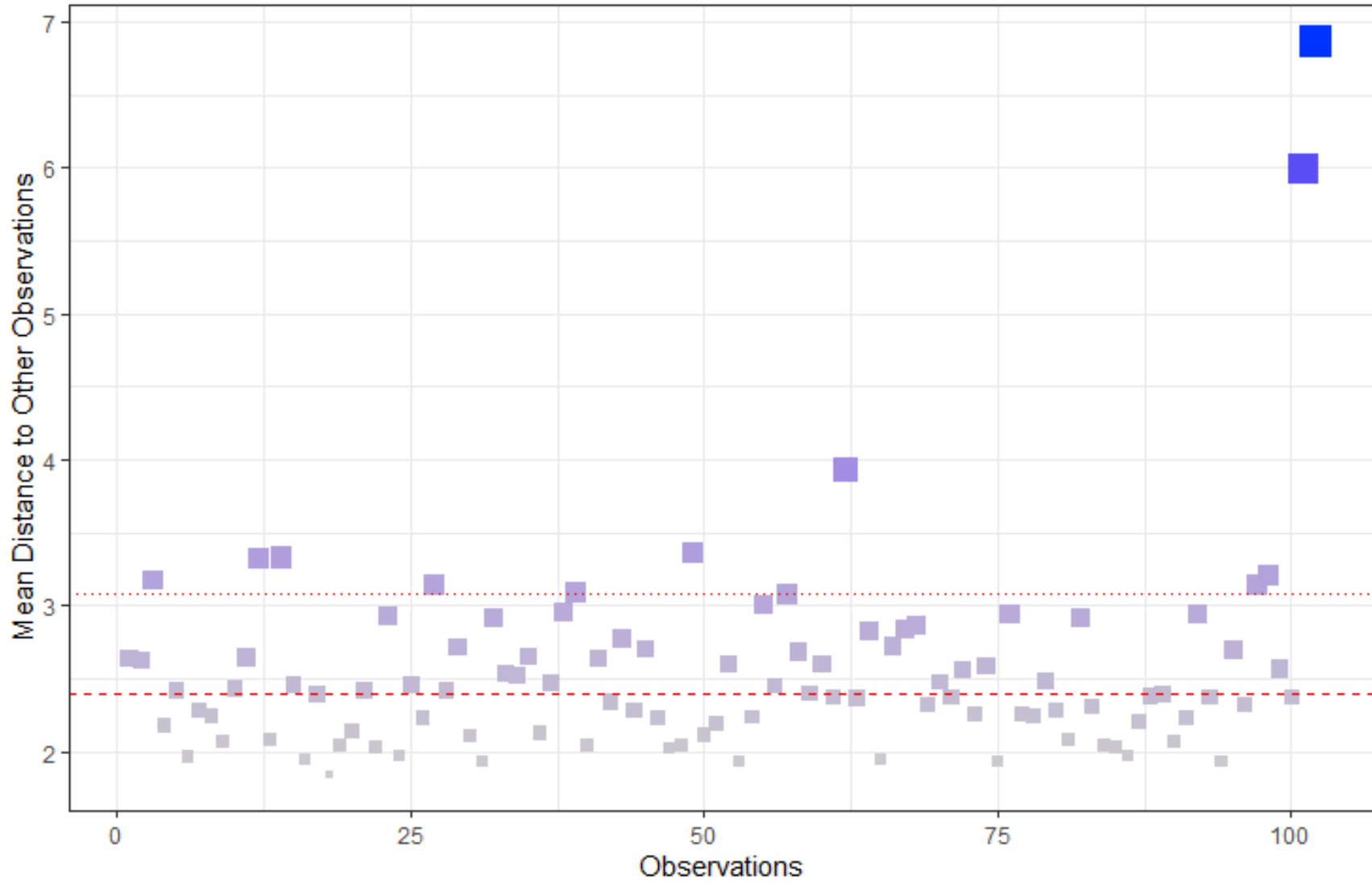
This approach often selects the most extreme observations as anomalous, which may be of limited use in practice.

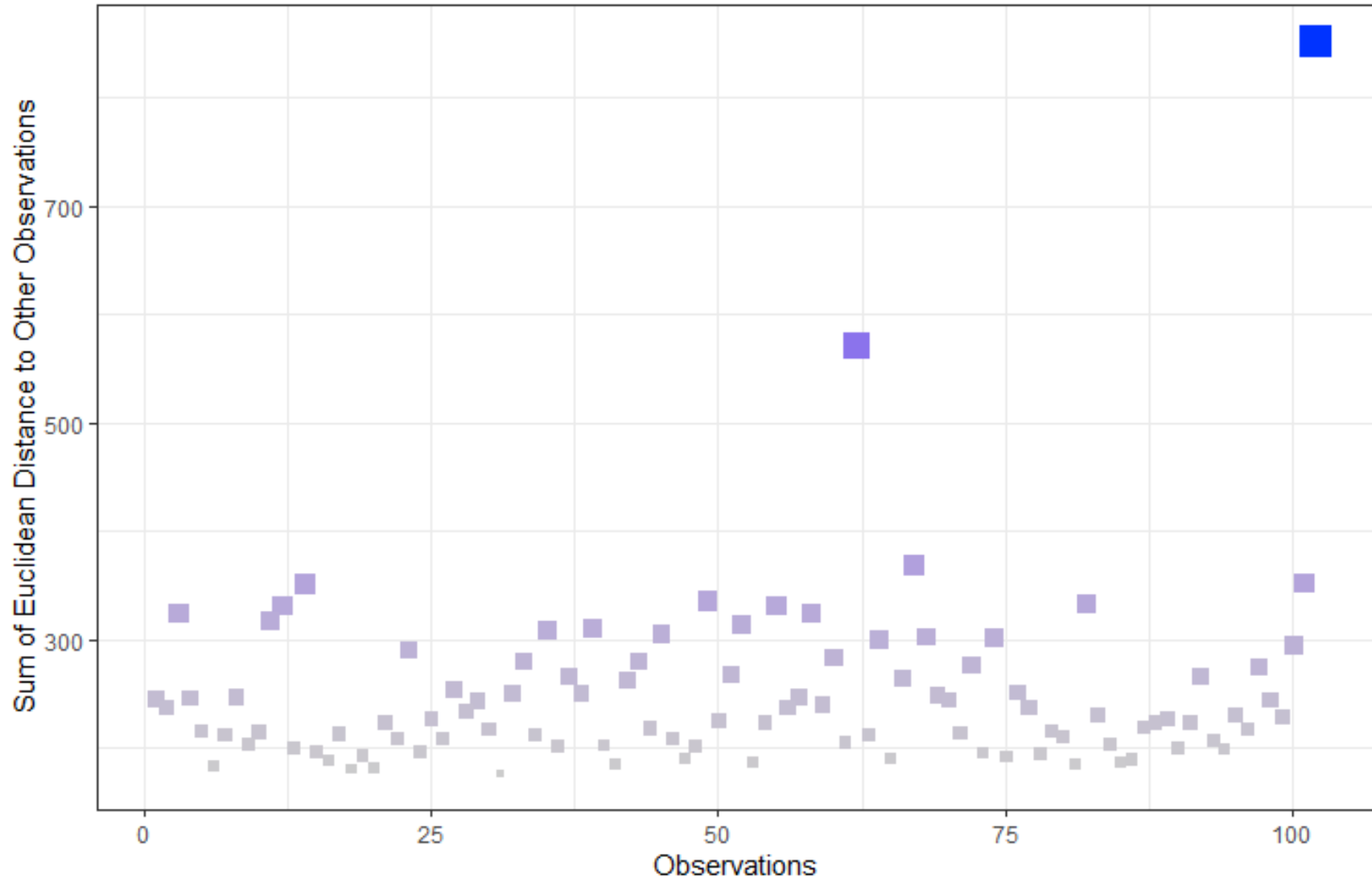
The **distance to nearest neighbour** (DTNN) algorithm uses

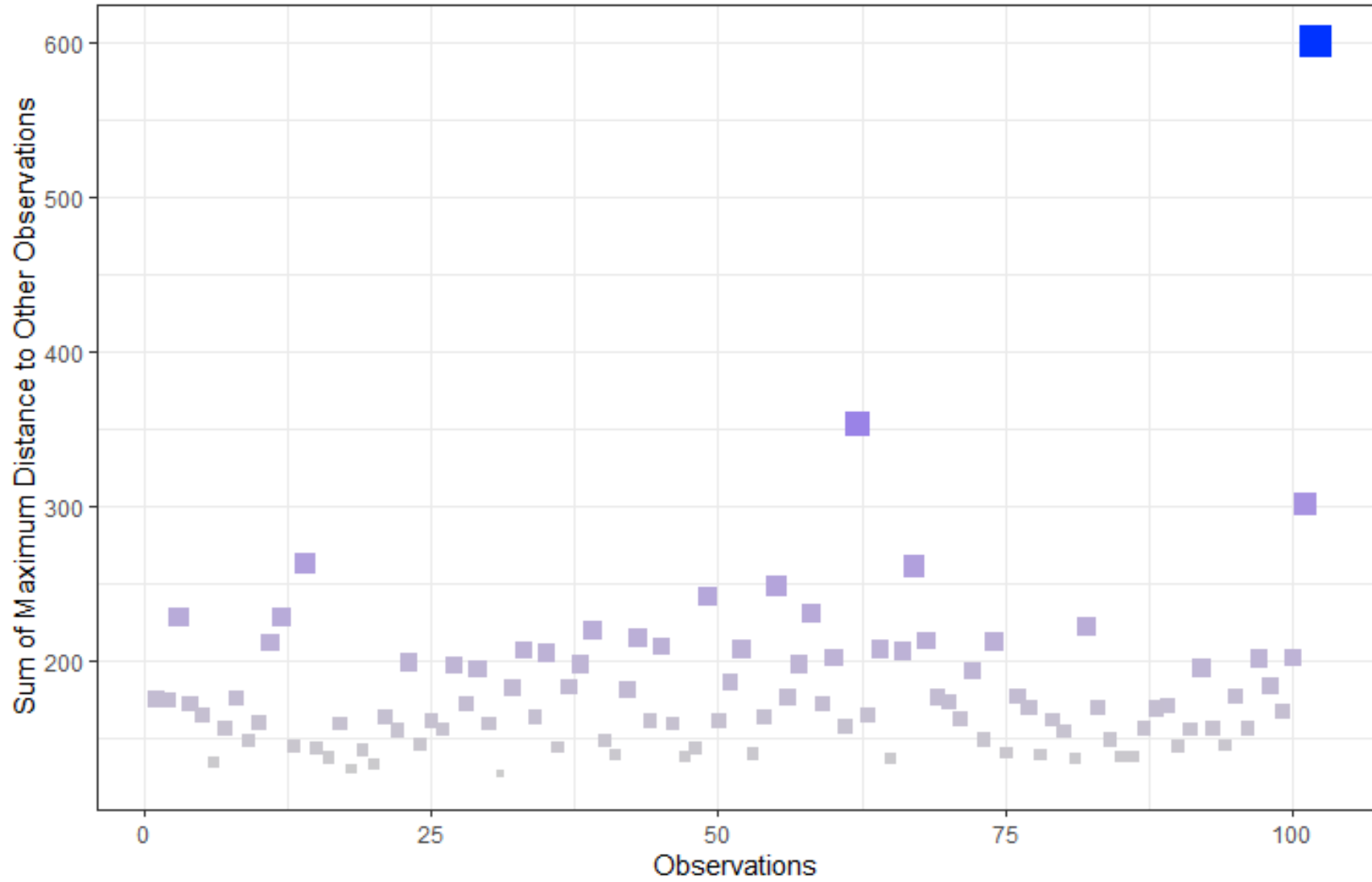
$$a(\mathbf{p}) = \min_{\mathbf{q} \neq \mathbf{p} \in D} \{d(\mathbf{q}, \mathbf{p})\},$$

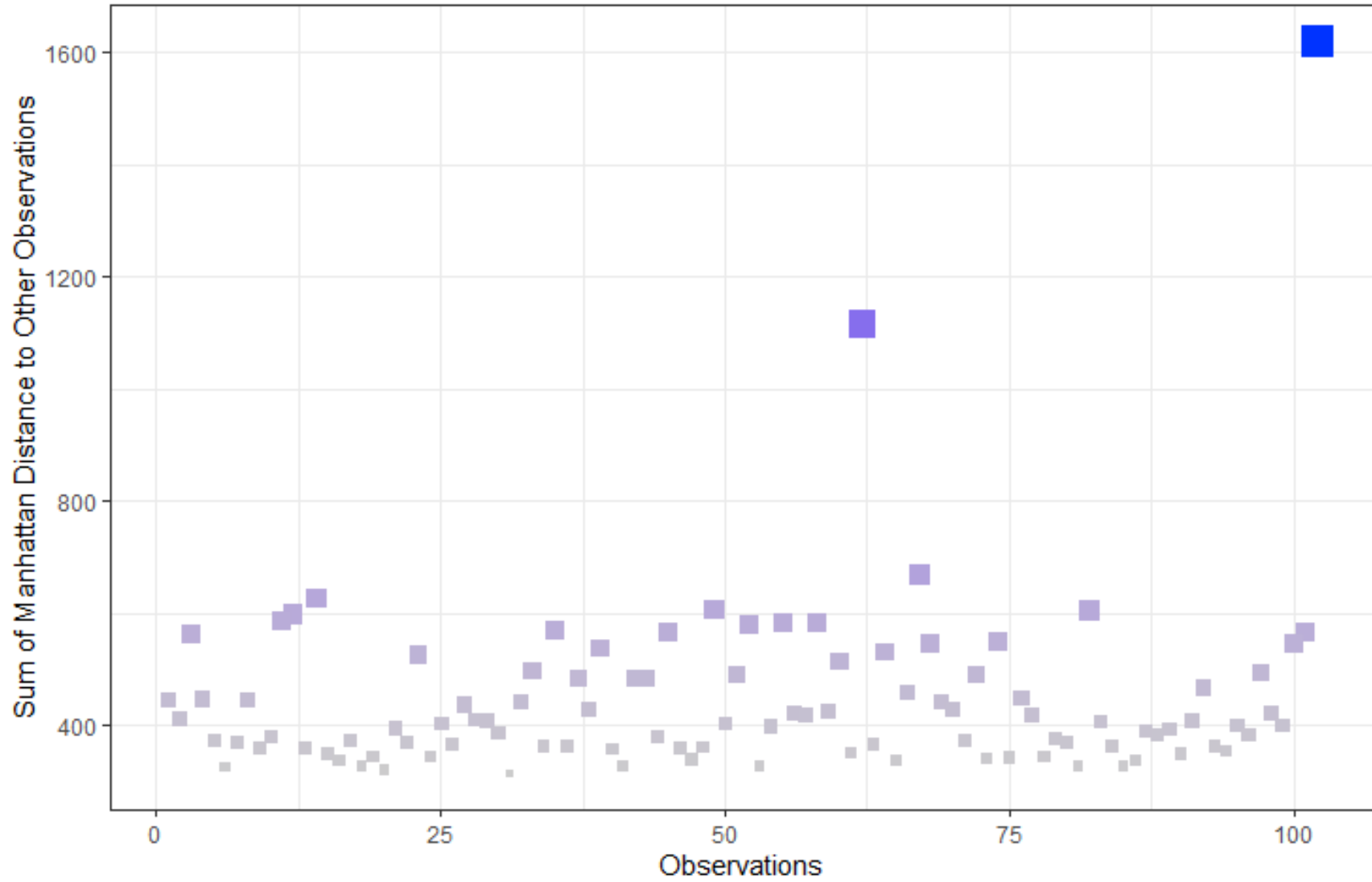
with a similar definition for the ν anomalous points.

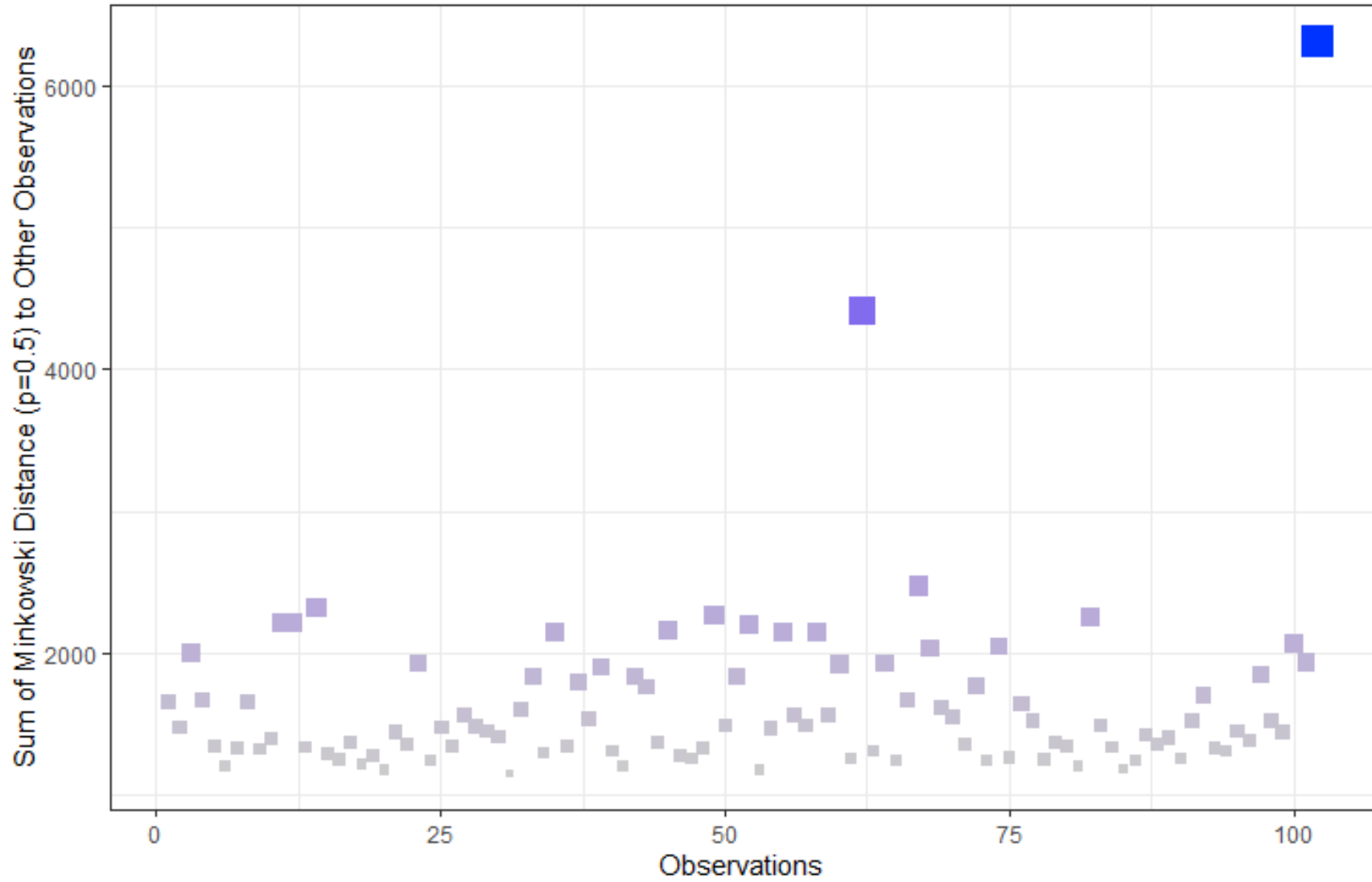
The **average distance to k nearest neighbours** and **median distance to k nearest neighbours** are defined similarly.

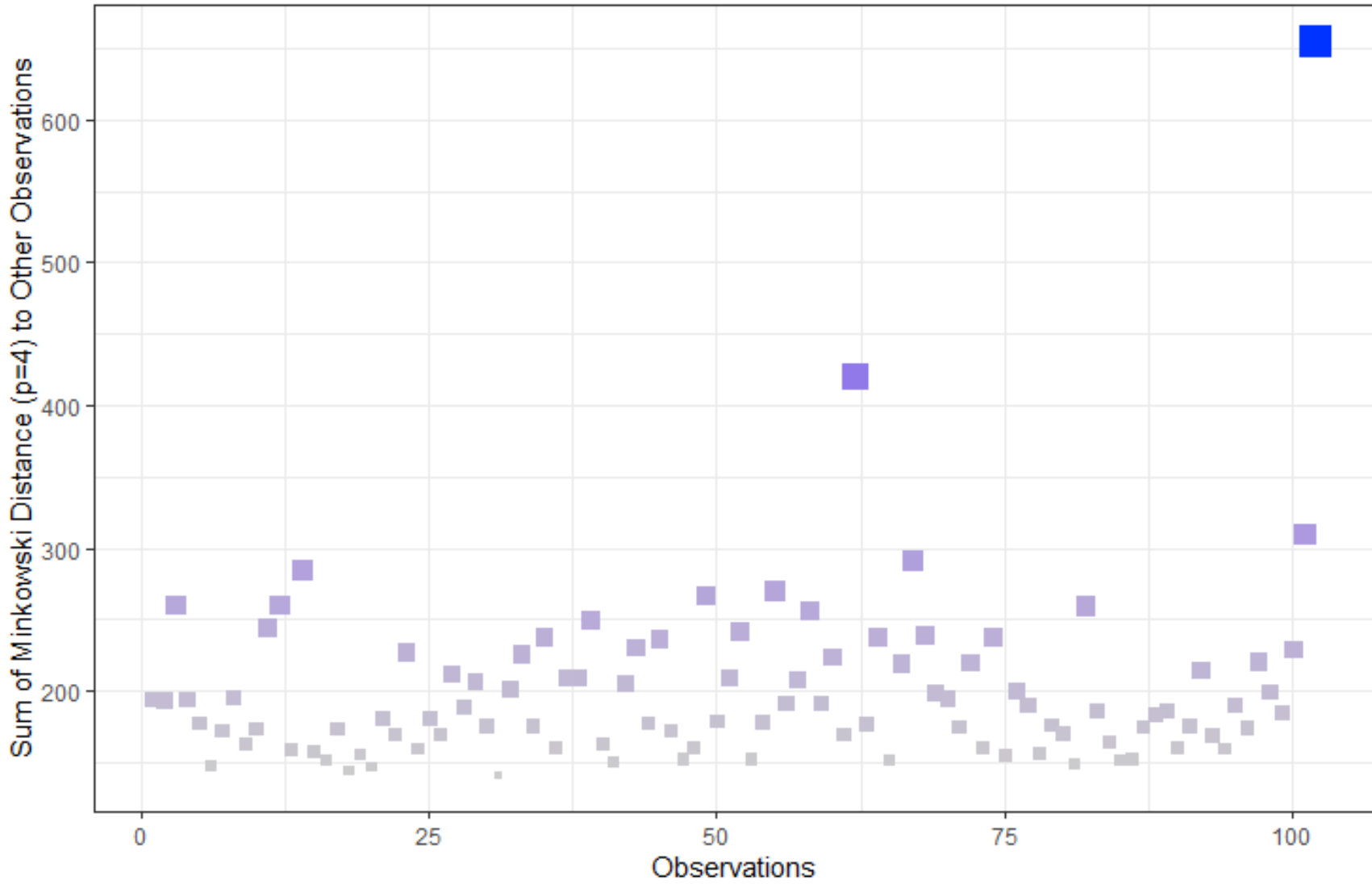












Mahalanobis	Euclidean	Supremum
102	102	102
101	62	62
67	67	101
14	101	14
12	14	67

Manhattan	Minkowski ($p = 0.5$)	Minkowski ($p = 4$)
102	102	102
62	62	62
67	67	101
14	14	67
49	49	14

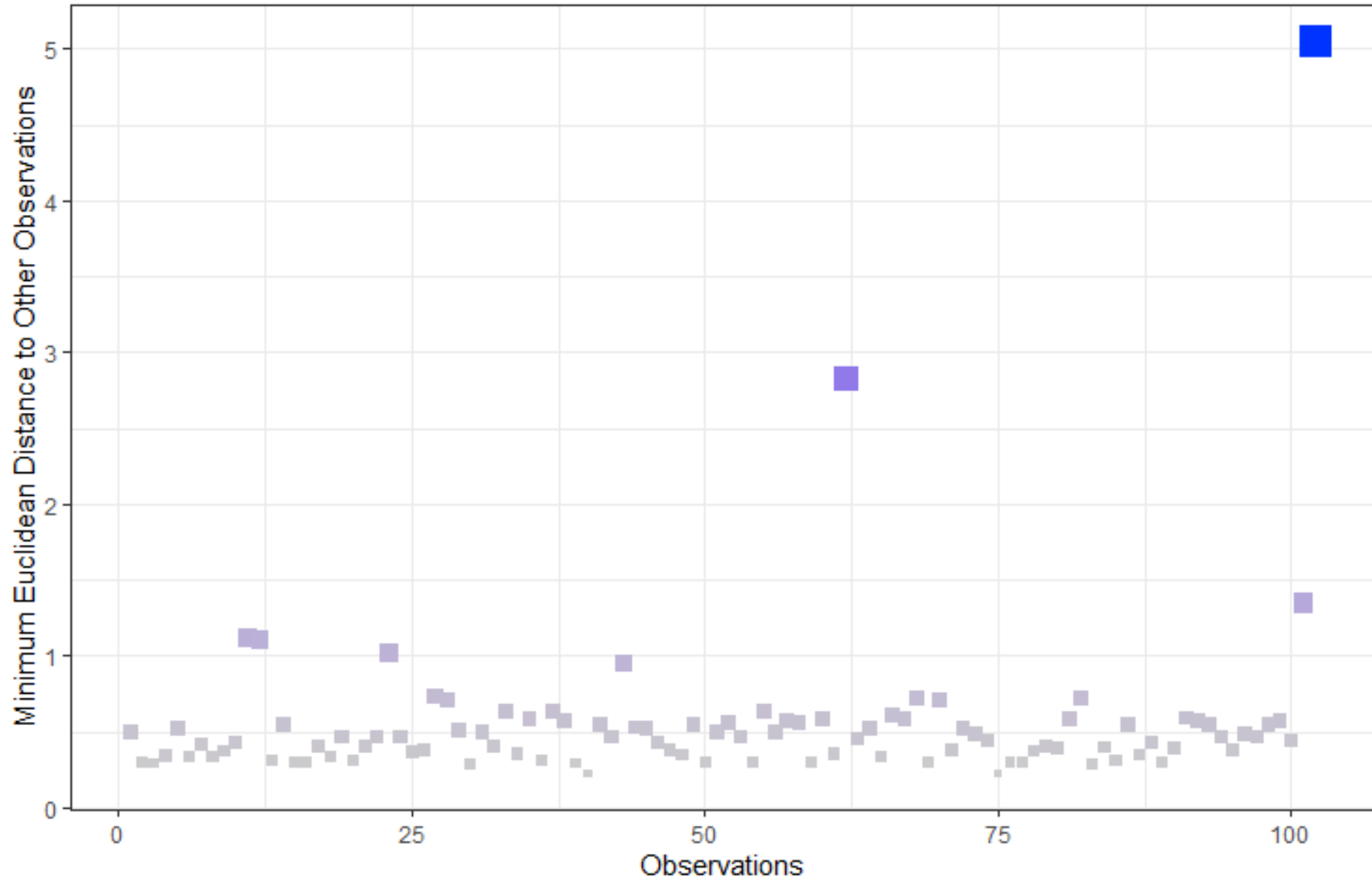
DTAP anomalies ($\nu = 5$), for various distances; unscaled data.

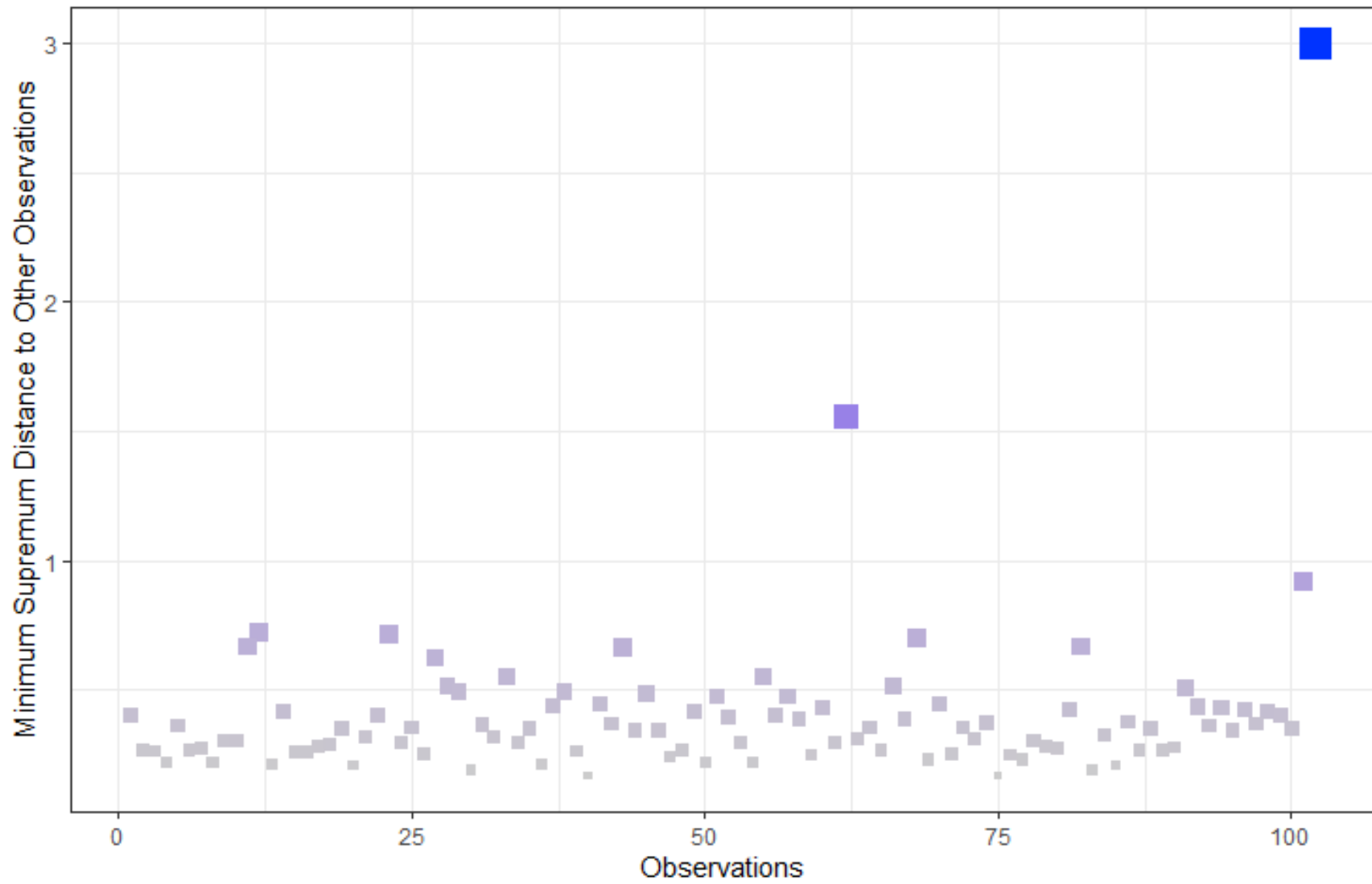
Euclidean	Supremum	Manhattan
102	102	102
61	62	62
67	14	14
101	55	67
14	49	49

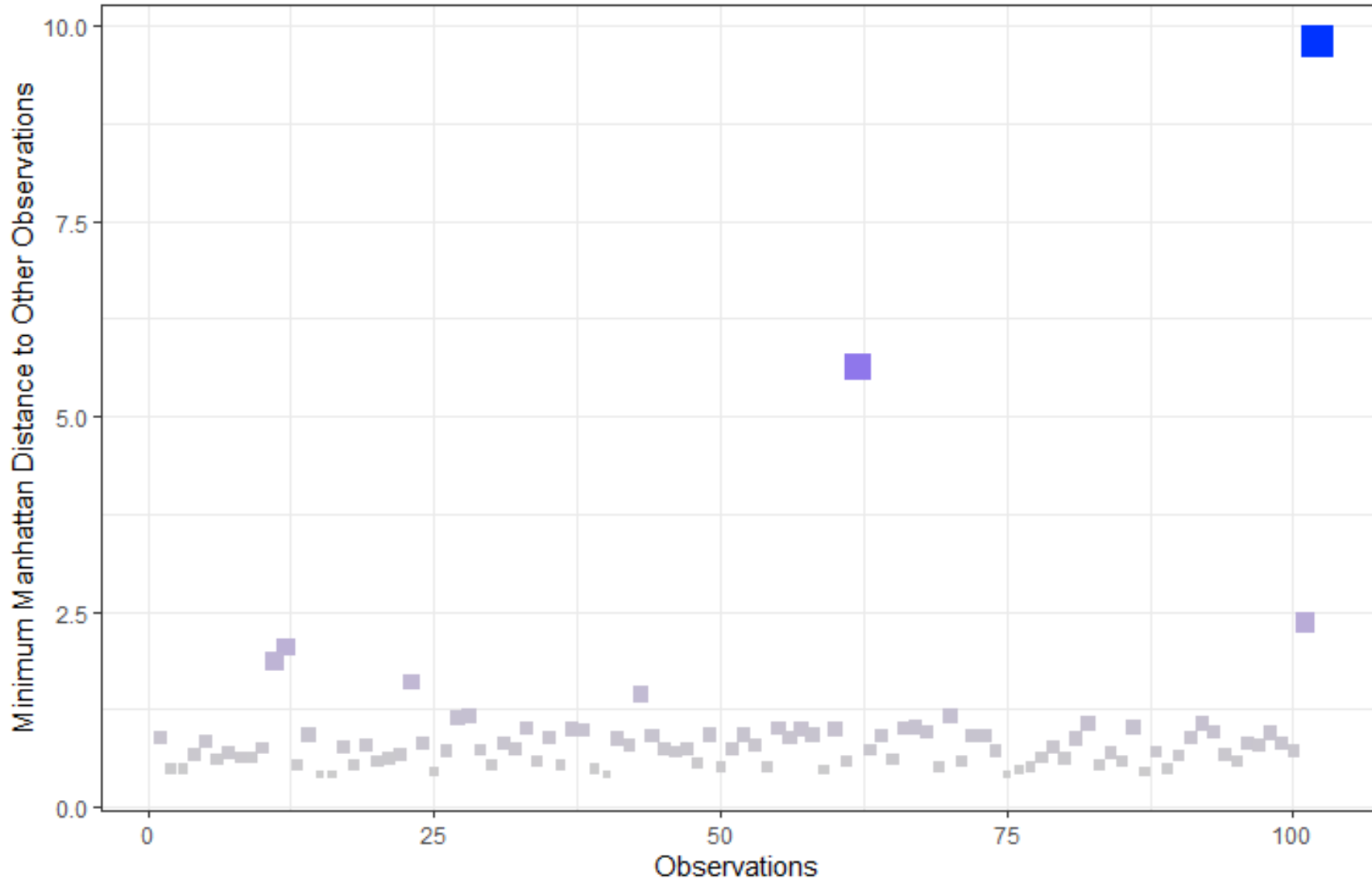
DTAP anomalies ($\nu = 5$), for various distances; scaled data.

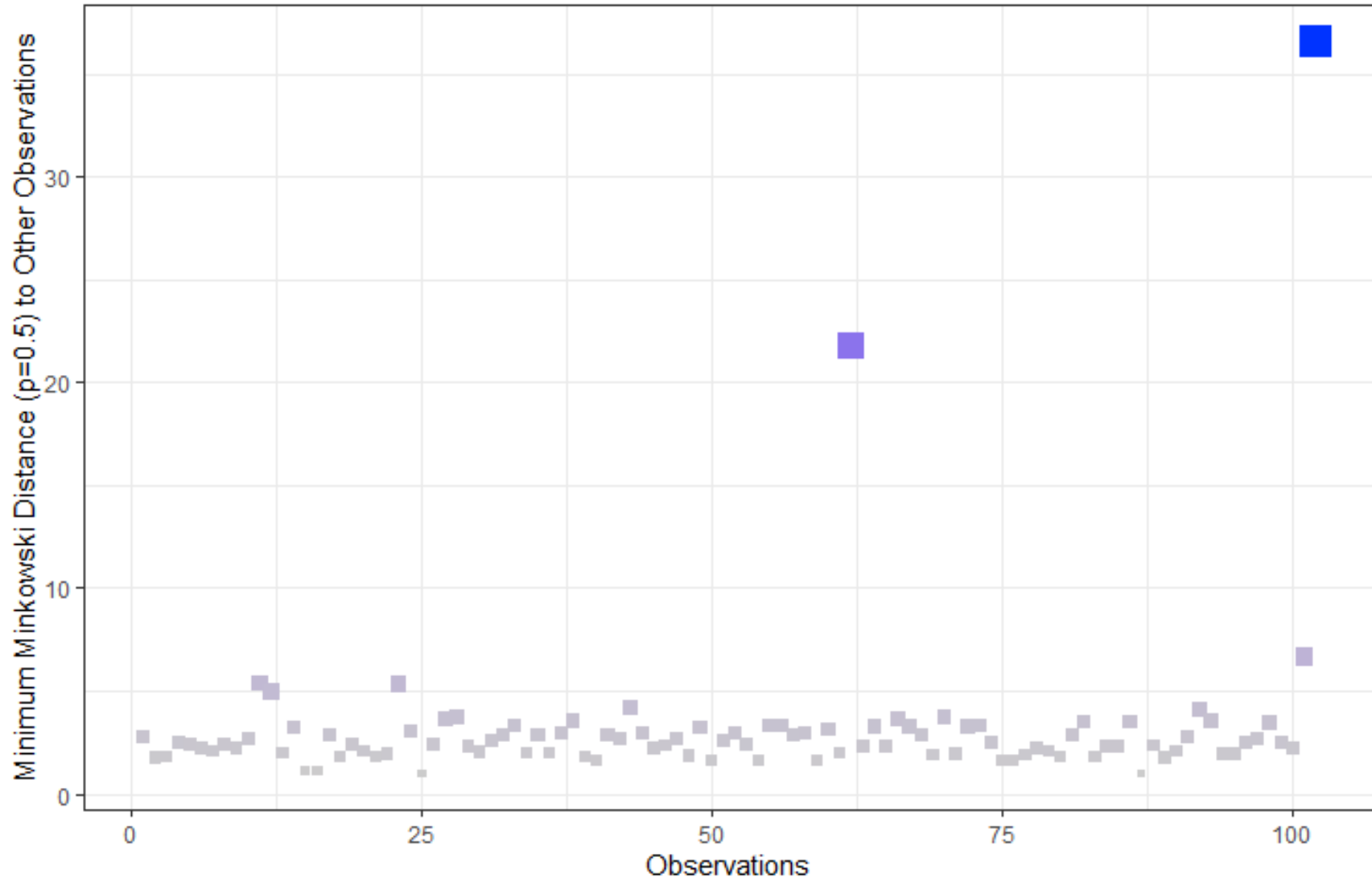
The rankings change according to the selected distance function, the data scaling, and the choice of algorithm (see following slides).

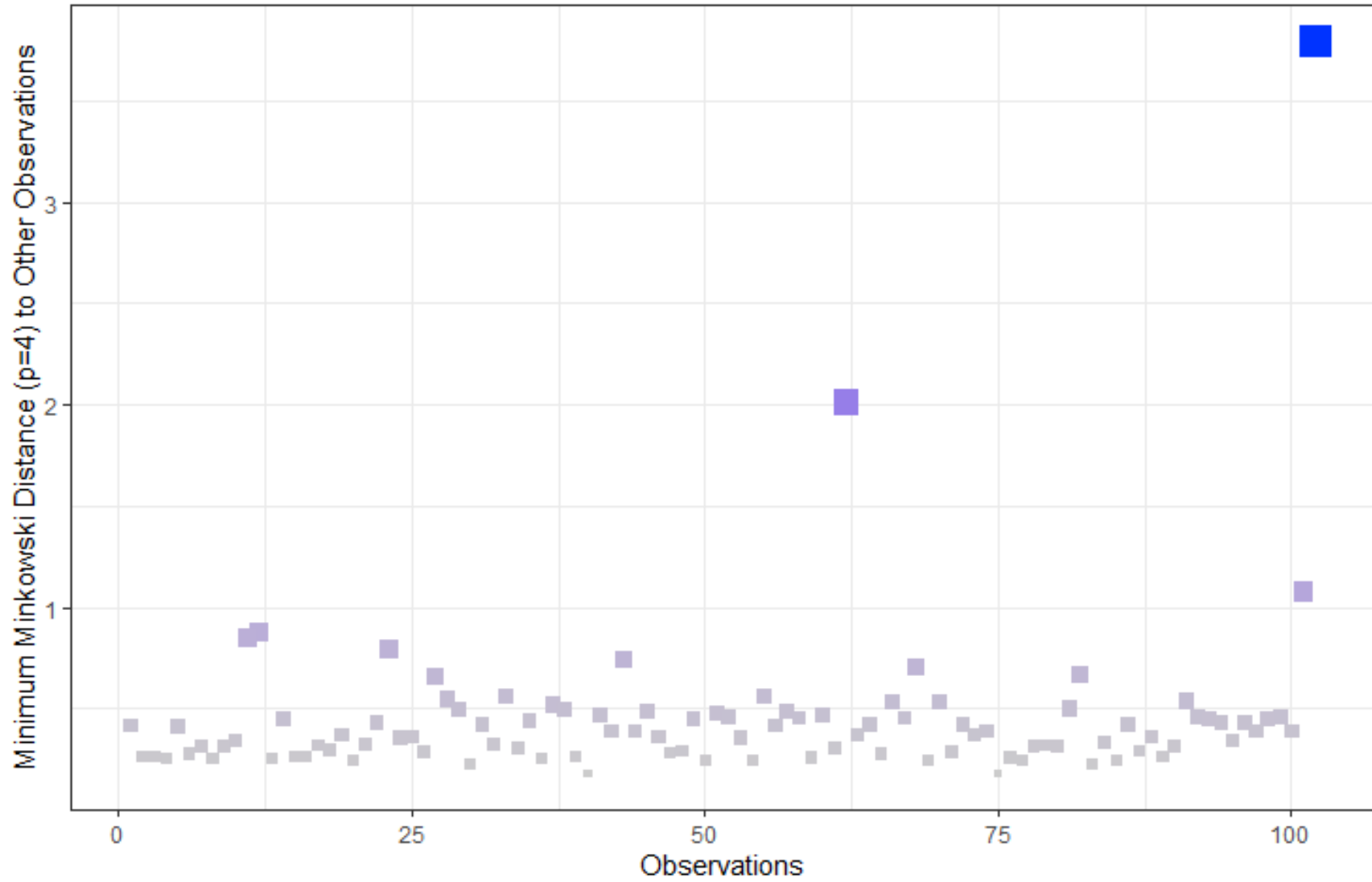
How do we make these decisions, then?











Mahalanobis	Euclidean	Supremum
102	102	102
101	62	62
67	101	101
14	11	12
12	12	23

Manhattan	Minkowski ($p = 0.5$)	Minkowski ($p = 4$)
102	102	102
62	62	62
101	101	101
12	11	12
11	23	11

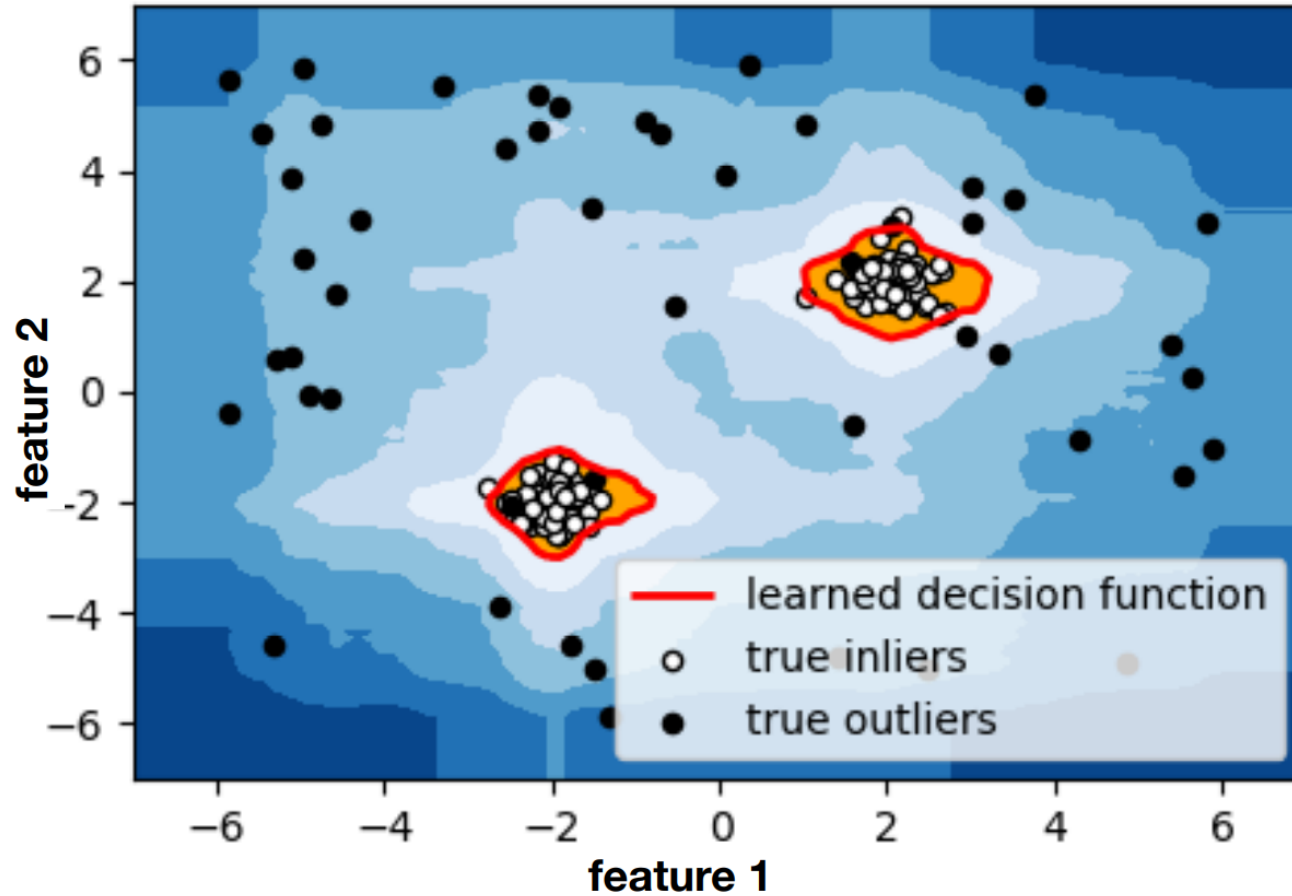
DTNN anomalies ($\nu = 5$), for various distances; unscaled data.

5.2.2 – Density-Based Methods

The flexibility in the choice of distance functions, scaling, and distance-based anomaly detection algorithm gives rise to different **anomaly rankings**. This is par for the course in the anomaly detection context.

Density-based approaches view points as anomalous if they occur in **low density regions**. Methods include:

- **local outlier factors;**
- **DBSCAN,** and
- **isolation forests.**



Low-density areas as outlier nurseries [Baron].

Local Outlier Factor

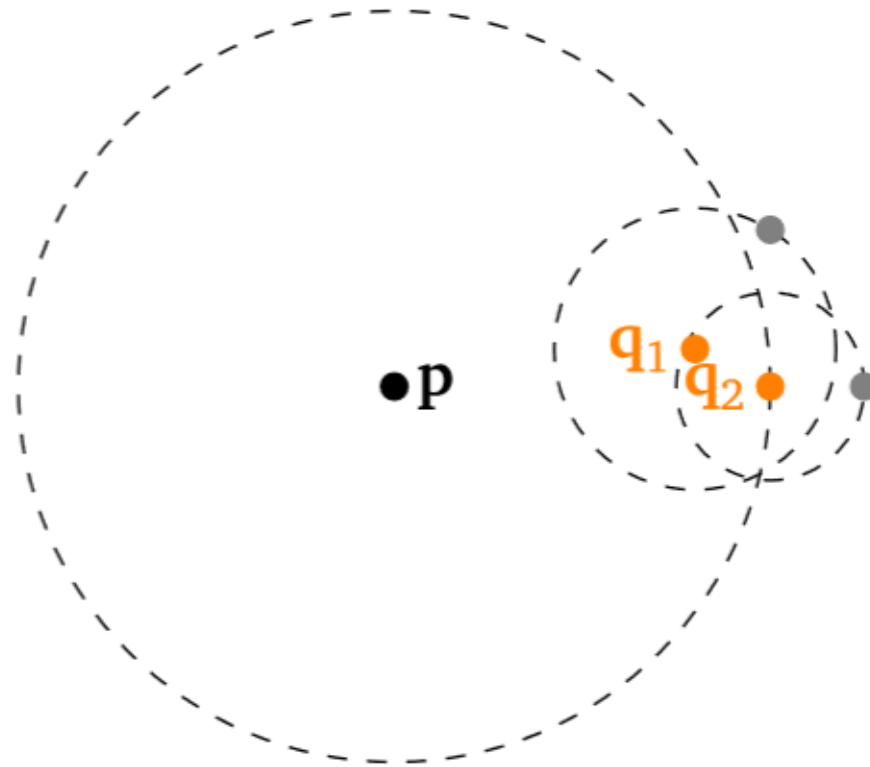
The **Local Outlier Factor** (LOF) algorithm works by measuring the **local deviation** of each observation **from its k nearest neighbours**.

An observation is said to be anomalous if the **deviation is large**.

A **local k -region** around a point \mathbf{p} is simply the set $N_k(\mathbf{p})$ of the k nearest neighbours of $\mathbf{p} \implies$ need to select a distance measure.

Local k -regions have different extent from one observation to the next \implies each \mathbf{p} has a **local density**.

Observations with anomalously small local density compared to its k -neighbours are identified as **outliers**.



For $k = 2$, \mathbf{p} has lower density than its k -neighbours $\mathbf{q}_1, \mathbf{q}_2$. The formal procedure is provided in Algorithm 1.

Algorithm 1: Local Outlier Factor (LOF)

```
1 Input: dataset  $D$ , point  $\mathbf{p} \in D$ , integer  $k$  for  
   number of nearest neighbours to consider,  
   distance function  $d$   
2 Compute the distance between all points in  $D$   
3 for  $\mathbf{p} \in D$  do  
4   | for  $\mathbf{q} \in D \setminus \{\mathbf{p}\}$  do  
5   |   | Compute  $d(\mathbf{p}, \mathbf{q})$   
6   | end  
7   | Order  $D$  by increasing distance from  $\mathbf{p}$   
8   | Set  $d_k(\mathbf{p}) = d(\mathbf{p}, \mathbf{q}_k)$   
9 end
```

- 10 Find the k nearest neighbours of \mathbf{p}
 11 Set $N_k(\mathbf{p}) = \{\mathbf{q} \in D \setminus \{\mathbf{p}\} : d(\mathbf{p}, \mathbf{q}) \leq d_k(\mathbf{p})\}$

- 12 Define the reachability distance

$$d_{\text{reach}}(\mathbf{p}, \mathbf{q}) = \max\{d_k(\mathbf{q}), d(\mathbf{p}, \mathbf{q})\}$$

- 13 Define the average reachability distance

$$\overline{d_{\text{reach}}}(\mathbf{p}) = \frac{\sum_{\mathbf{q} \in N_k(\mathbf{p})} d_{\text{reach}}(\mathbf{p}, \mathbf{q})}{|N_k(\mathbf{p})|}$$

- 14 Define the local reachability density

$$\ell_k(\mathbf{p}) = \left(\overline{d_{\text{reach}}}(\mathbf{p})\right)^{-1}$$

- 15 Compute the local outlier factor $a_k(\mathbf{p}) = \frac{\sum_{\mathbf{q} \in N_k(\mathbf{p})} \frac{\ell_k(\mathbf{q})}{\ell_k(\mathbf{p})}}{|N_k(\mathbf{p})|}$

- 16 **Output:** LOF $a_k(\mathbf{p})$
-

Any point with a local outlier factor $a_k(\mathbf{p})$ above some threshold τ is a **local outliers**.

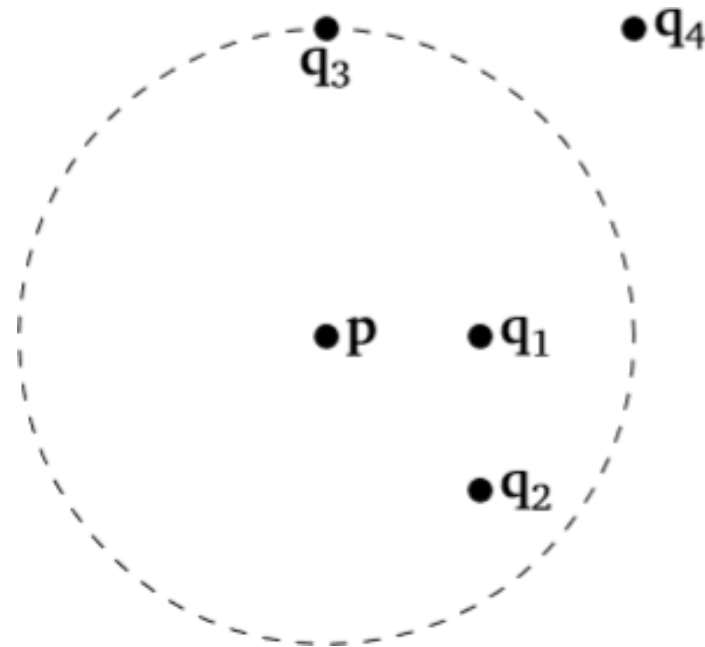
 **Selecting appropriate k and threshold τ is not simple.**

Using a derived **reachability distance** improves the stability of the algorithm results: within $N_k(\mathbf{p})$,

$$d_{\text{reach}}(\mathbf{p}, \mathbf{q}) = \max_{\ell} \{d(\mathbf{p}, \mathbf{q}_{\ell}); \mathbf{q}_{\ell} \in N_k(\mathbf{p})\};$$

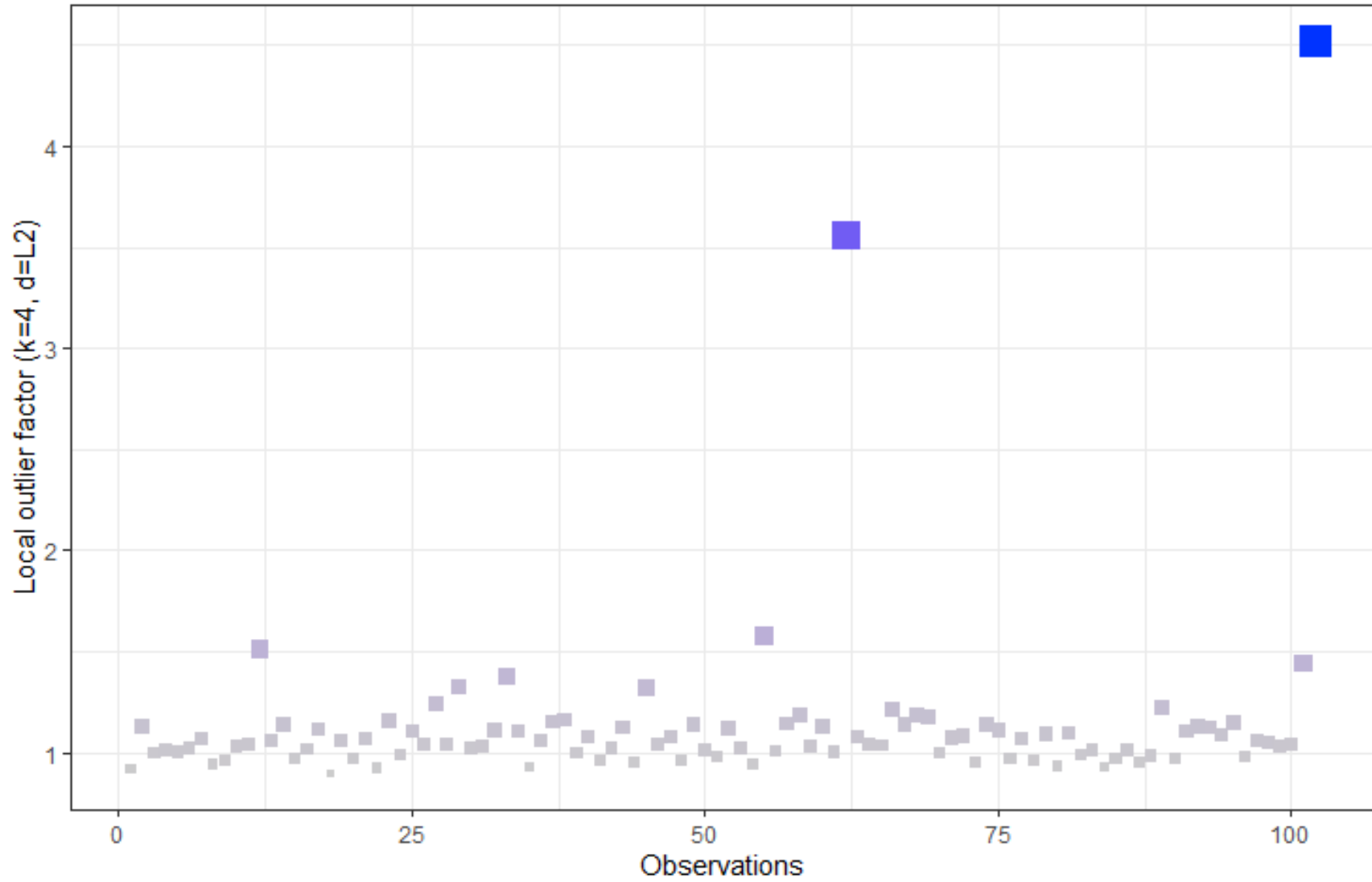
outside of $N_k(\mathbf{p})$,

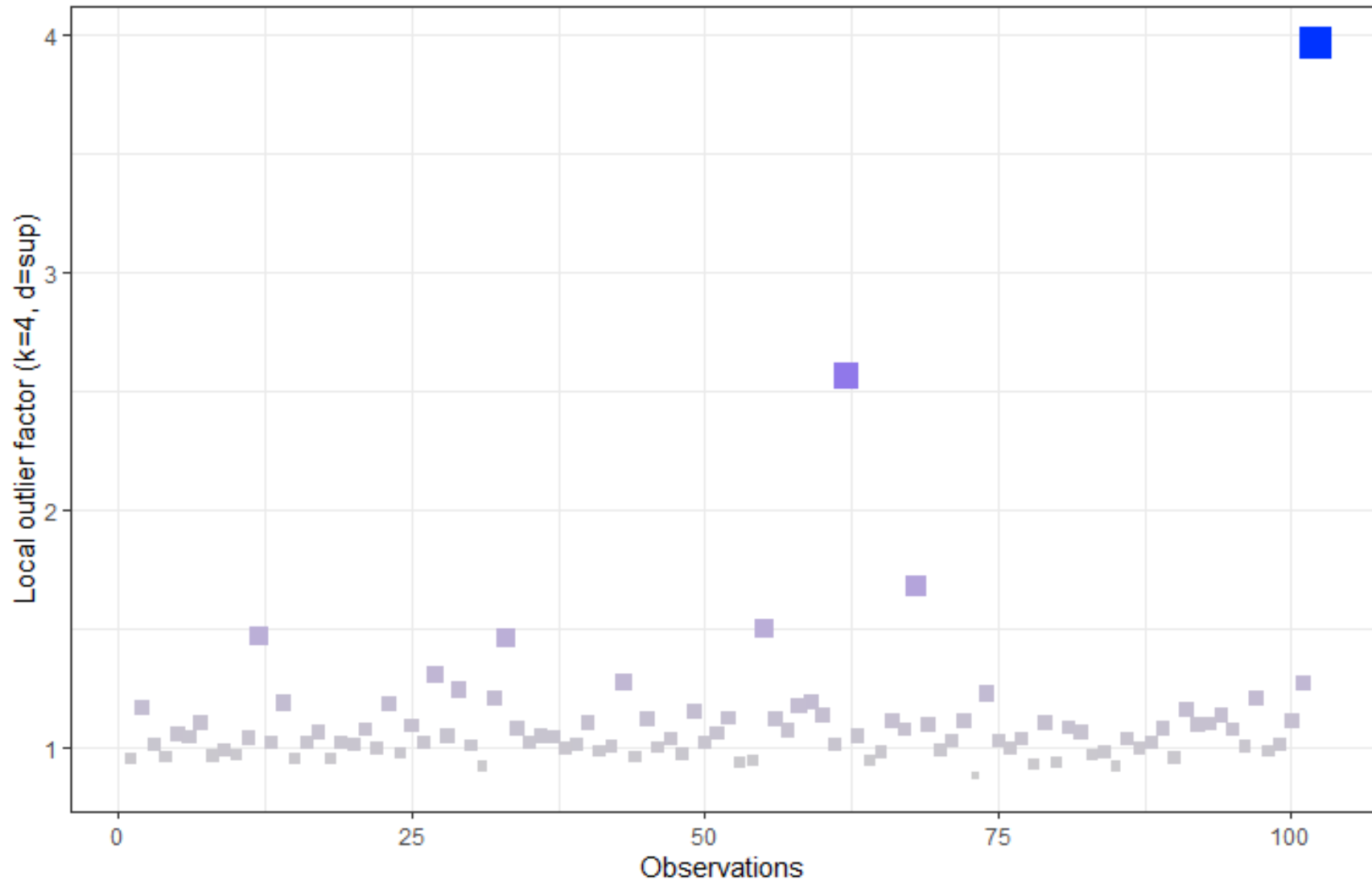
$$d_{\text{reach}}(\mathbf{p}, \mathbf{q}) = d(\mathbf{p}, \mathbf{q}).$$



The region of uniform reachability distance around \mathbf{p} for $k = 3$:

$$d_{\text{reach}}(\mathbf{p}, \mathbf{q}_1) = d_{\text{reach}}(\mathbf{p}, \mathbf{q}_2) = d_{\text{reach}}(\mathbf{p}, \mathbf{q}_3) = d(\mathbf{p}, \mathbf{q}_3); \quad d_{\text{reach}}(\mathbf{p}, \mathbf{q}_4) = d(\mathbf{p}, \mathbf{q}_4).$$






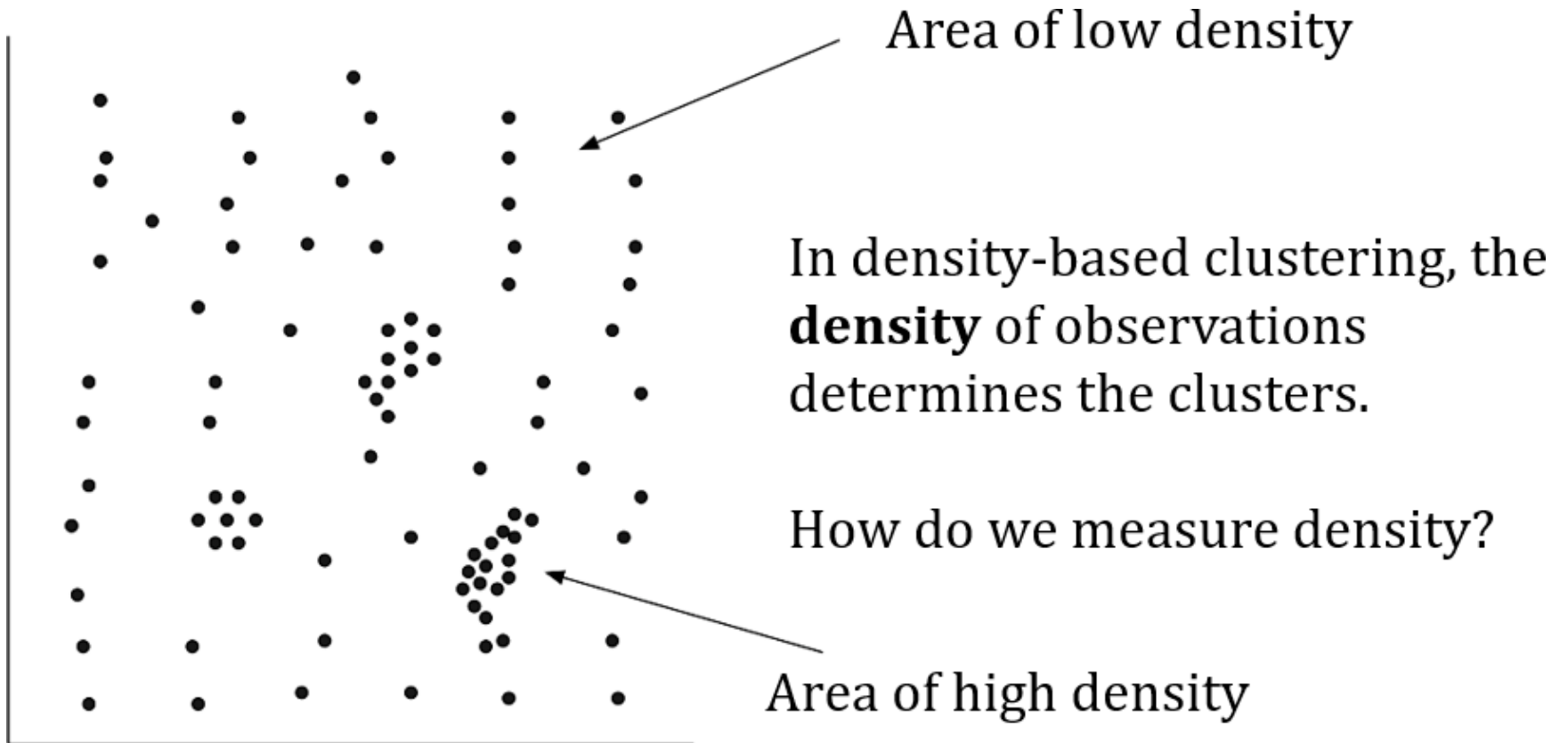
DBSCAN

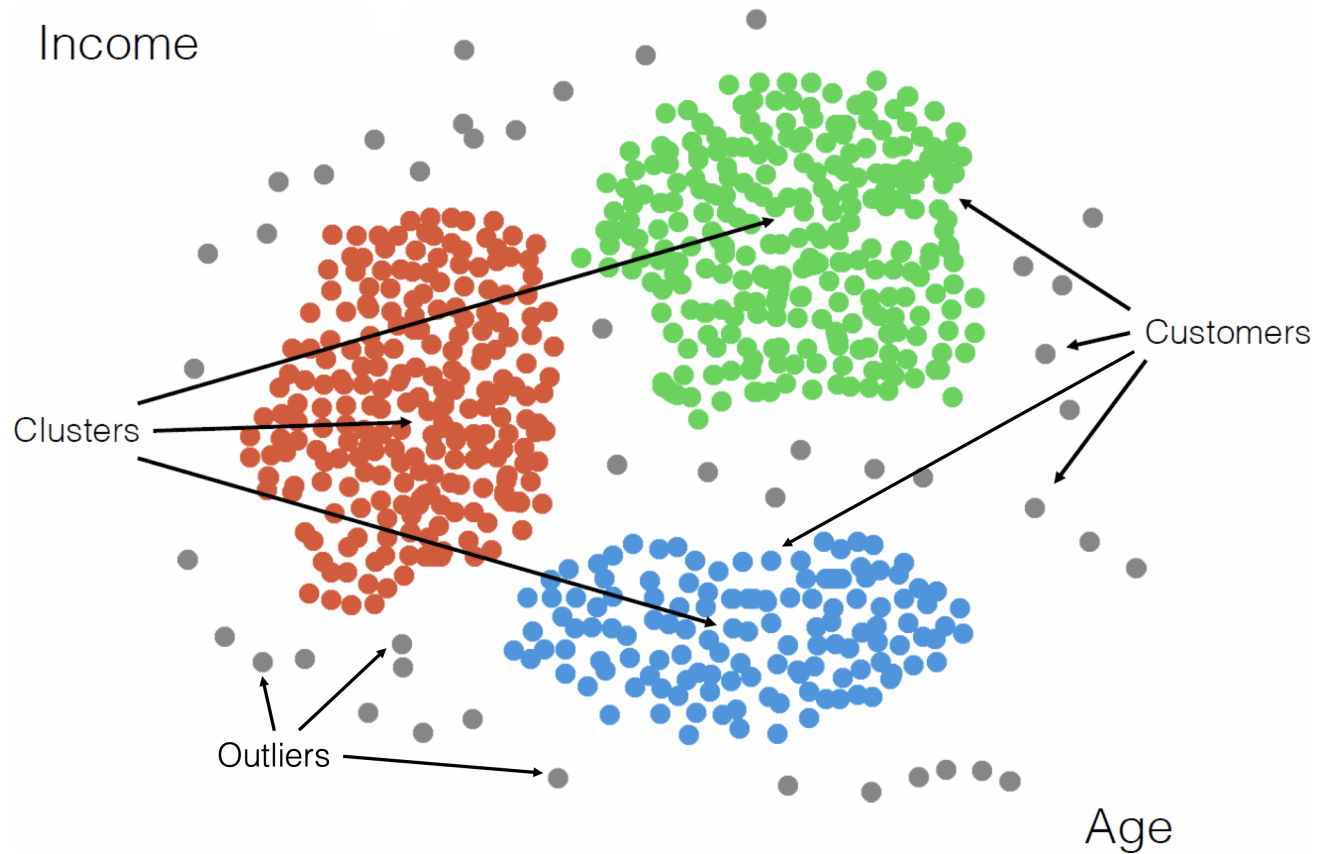
Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a density-based clustering algorithm that groups nearby points together.

Points that do not fall in the clusters are labeled as (potential) **anomalies**.

Hierarchical DBSCAN (HDBSCAN), which removes the problem of choosing one of DBSCAN's parameters (the radius of neighbourhoods).

 We won't be talking about **clustering** in a general sense – it could form the basis of 2+ courses – but it would be a good idea to take some time to read up on the topic if you are not familiar with it.

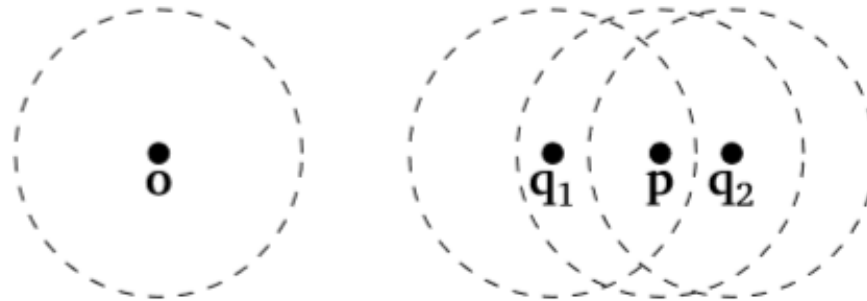




Clusters of regular customers (red, green, blue) and potential anomalies/outliers (grey) in an artificial dataset.

In DBSCAN,

- \mathbf{p} is a **core point** if there are $m+$ points within distance r of \mathbf{p} ;
- \mathbf{q} (non-core) is a **border point** if it is within distance r of a core point;
- \mathbf{o} is an **outlier** if it is neither a core nor a border point.



For minimum neighbourhood size $m = 2$ and the fixed radius r above, \mathbf{o} is an outlier, \mathbf{p} is a core point, and $\mathbf{q}_1, \mathbf{q}_2$ are border points.

DBSCAN considers each point in the dataset individually:

- if a point is not a core point or a border point, it is considered an outlier;
- if it is a border point, it is not considered an outlier, but it does not form the basis of a new cluster of regular observations;
- if it is a core point, then its r -neighbourhood forms the beginning of a new cluster;
- each point in this r -neighbourhood is then considered in turn, with the r -neighbourhoods of other core points contained in the neighbourhood being added to the cluster (regular observations).

This expansion repeats until all points have been examined.

During the expansion, points that were previously labelled as outliers may be updated as they become border points in a new cluster.

This process continues until every point has either been assigned to a cluster or labelled as an outlier.

DBSCAN's dual use as a clustering algorithm may seem irrelevant in the outlier detection setting, but its ability to successfully identify clusters is crucial (the remaining points are outliers).

The formal procedure is provided in Algorithm 2.

(But why use DBSCAN instead of any other of the 50+ cluster algorithms?)

Algorithm 2: DBSCAN

```
1 Input: dataset  $D$ , distance function  $d$ ,  
   neighbourhood radius  $r > 0$ , minimum number of  
   points to be considered a cluster  $m \in \mathbb{N}$   
2  $Clusters = \{\}$   
3  $Outliers = \{\}$   
4 for  $\mathbf{p} \in D$  do  
5   if  $\mathbf{p} \in Outliers \cup (\cup_{C \in Clusters} C)$  then  
6     continue  
7   end  
8   Set  $N(\mathbf{p}) = \{\mathbf{q} \in D : d(\mathbf{p}, \mathbf{q}) \leq r\}$   
9   if  $|N(\mathbf{p})| < m$  then  
10    Add  $\mathbf{p}$  to  $Outliers$   
11    continue  
12  end  
13  else  
14     $Cluster = N(\mathbf{p})$ 
```

```
15   |   |   for  $q \in Cluster \setminus \{p\}$  do
16   |   |   |   if  $q \in Outliers$  then
17   |   |   |   |   Remove  $q$  from Outliers
18   |   |   |   end
19   |   |   |   else if  $q \in \cup_{C \in Clusters} C$  then
20   |   |   |   |   continue
21   |   |   |   end
22   |   |   |   Set  $N(q) = \{q' \in D : d(q, q') \leq r\}$ 
23   |   |   |   if  $|N(q)| \geq m$  then
24   |   |   |   |    $Cluster = Cluster \cup N(q)$ 
25   |   |   |   end
26   |   |   end
27   |   end
28   |   Add Cluster to Clusters
29 end
30 return Outliers
31 Output: a list of outliers
```

ϵ : _____
minPts: 3

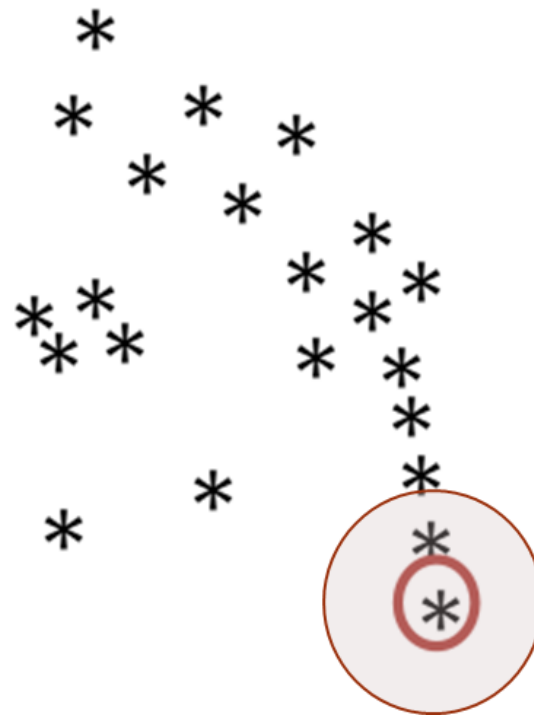


ϵ : _____
minPts: 3



Point picked at random

ϵ : _____
minPts: 3



Point identified as non-core point

ϵ : _____
minPts: 3



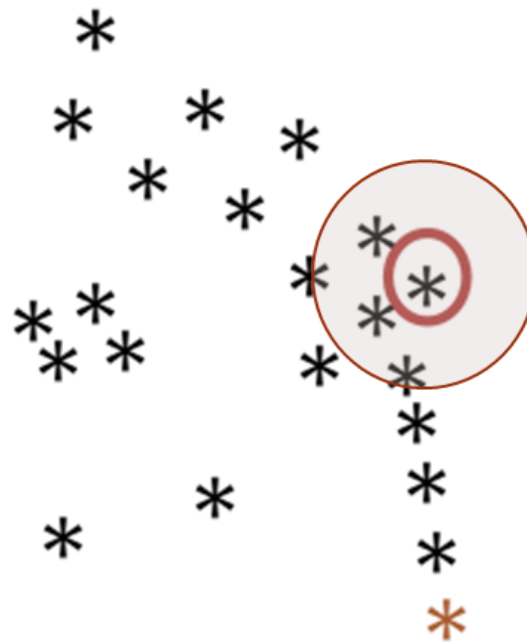
Point identified as non-core point

$\epsilon:$
minPts: 3



Another point picked at random

$\epsilon:$
minPts: 3



Point identified as a core point

ϵ :
minPts: 3



Points in the ϵ -neighbourhood

ϵ : _____
minPts: 3

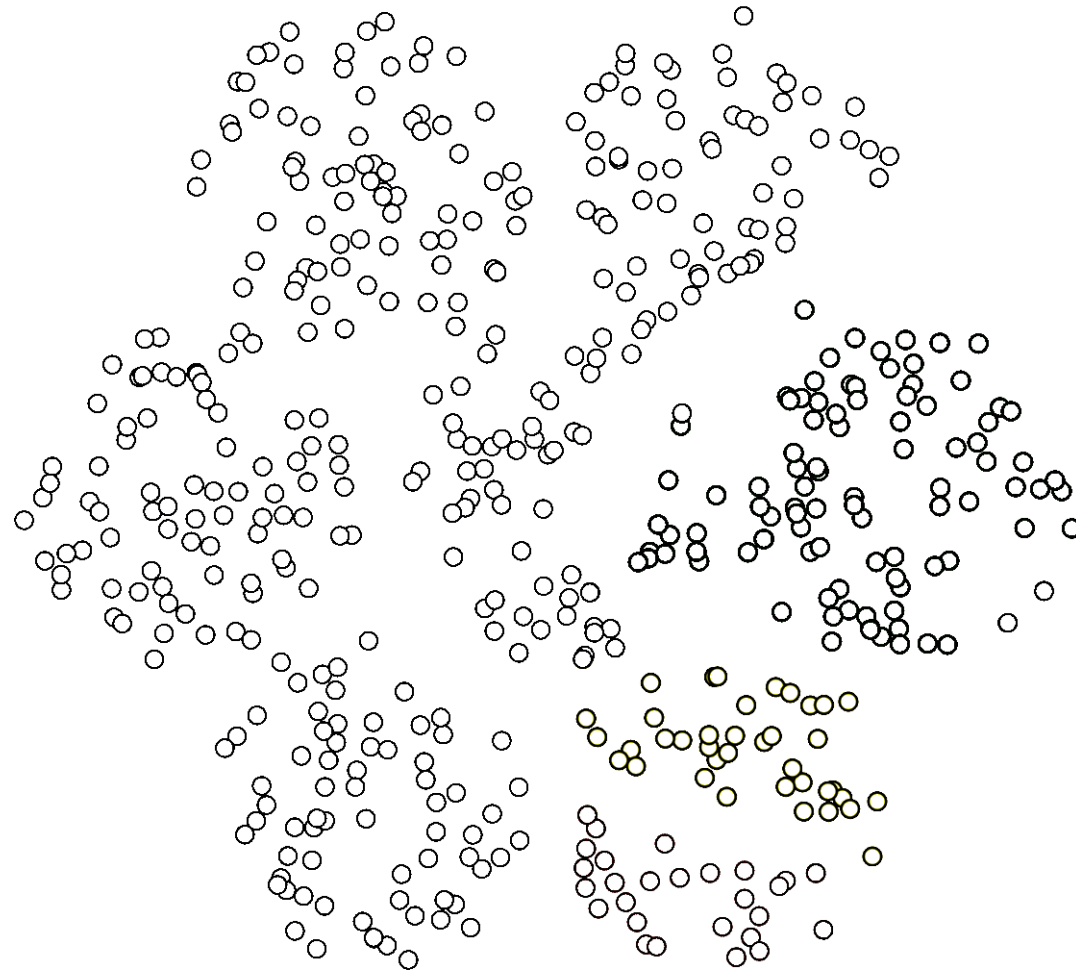


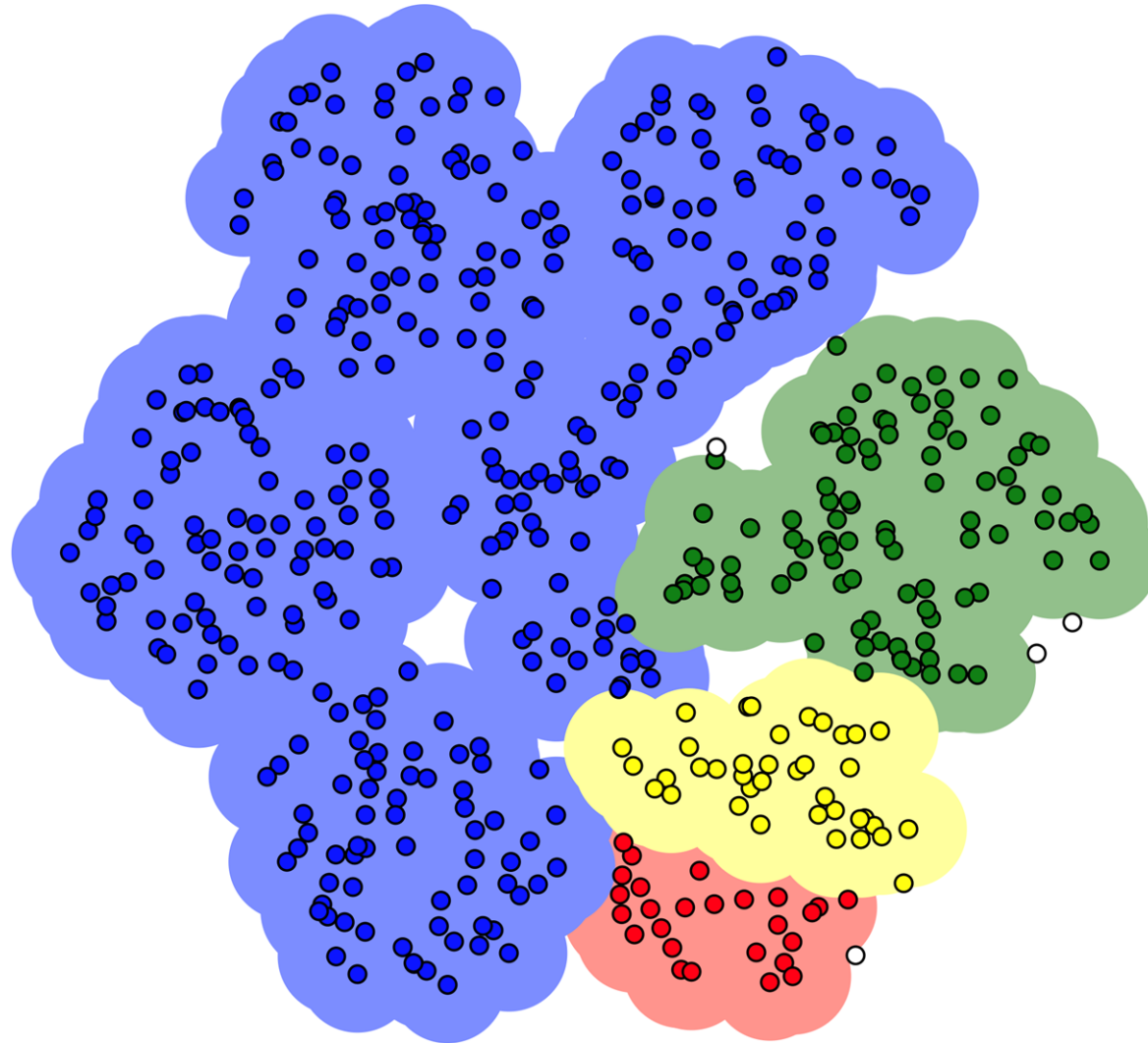
Resulting cluster

ϵ : _____
minPts: 3



Clusters and outliers





Strengths:

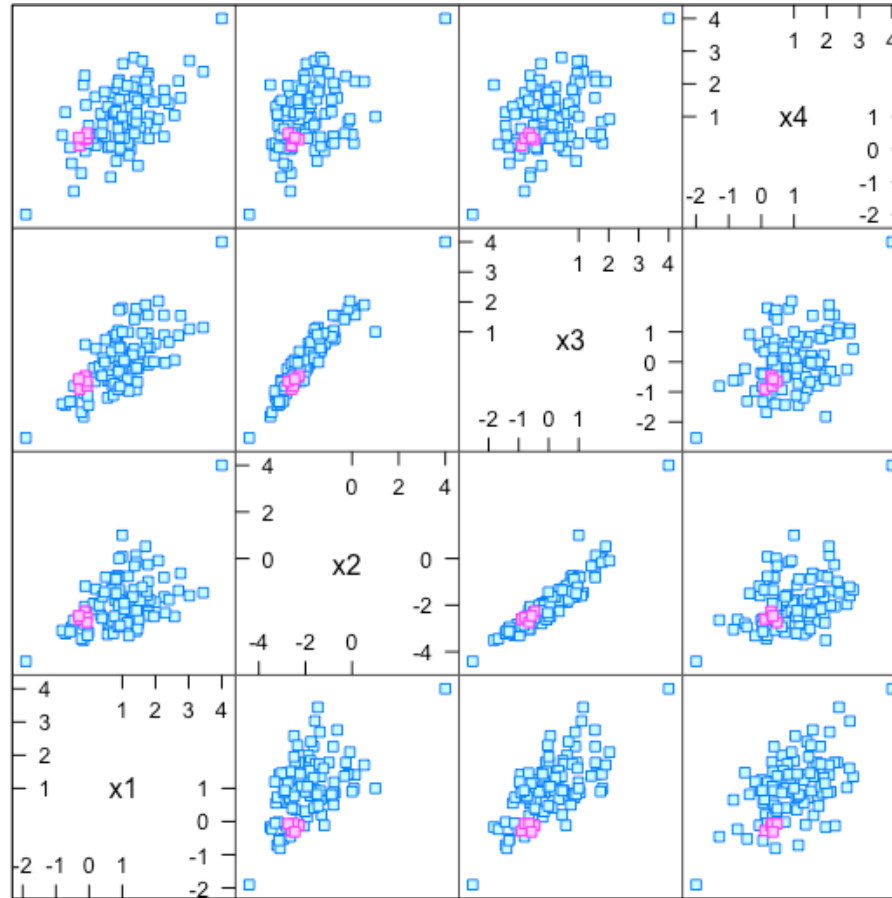
- the number of clusters does not need to be known beforehand;
- clusters of arbitrary shape can be detected;
- with HDBSCAN, only the parameter for the minimum cluster size $m \geq n + 1$ is required (larger values of m allow for better noise identification).

Limitations:

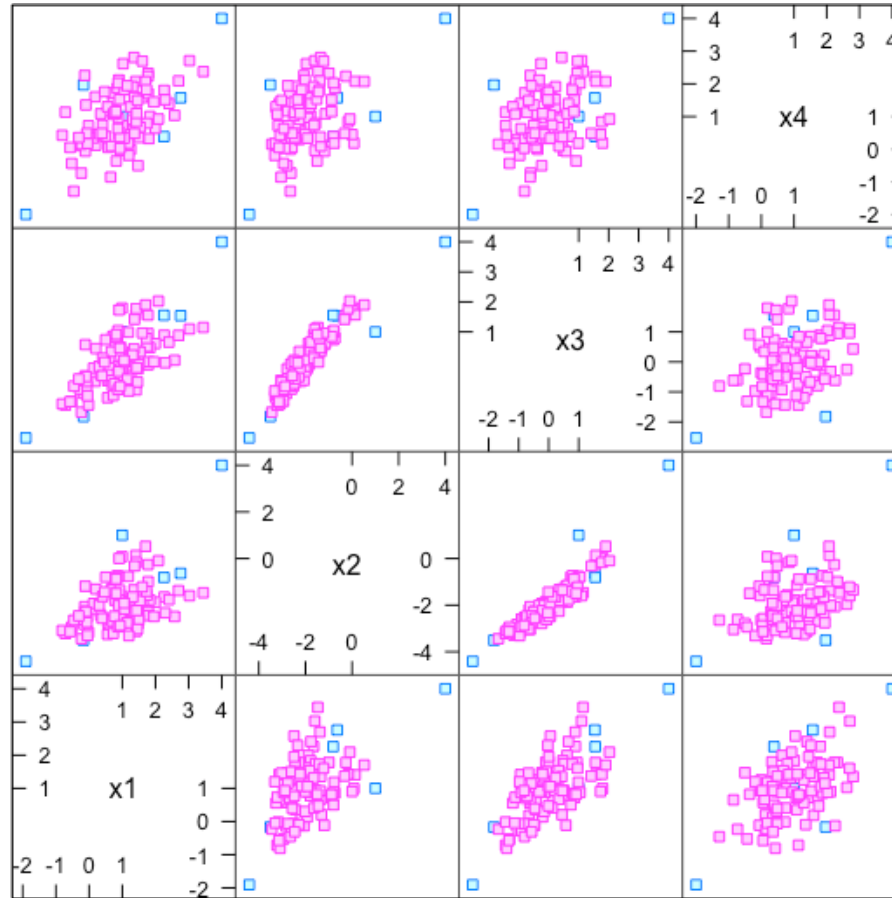
- not entirely deterministic, as border points can be assigned to different clusters depending on the order in which core points are considered (does not affect its use as an anomaly detection algorithm).

- suffers from the **Curse of Dimensionality** – in high-dimensional spaces, Euclidean-based distances have a difficult time distinguish **near** observations from **distant** ones;
- cannot handle differences in local densities when the radius r of a neighbourhood is fixed \implies sparser clusters could be labelled as outliers, or outliers surrounding a denser cluster could be included in the cluster.

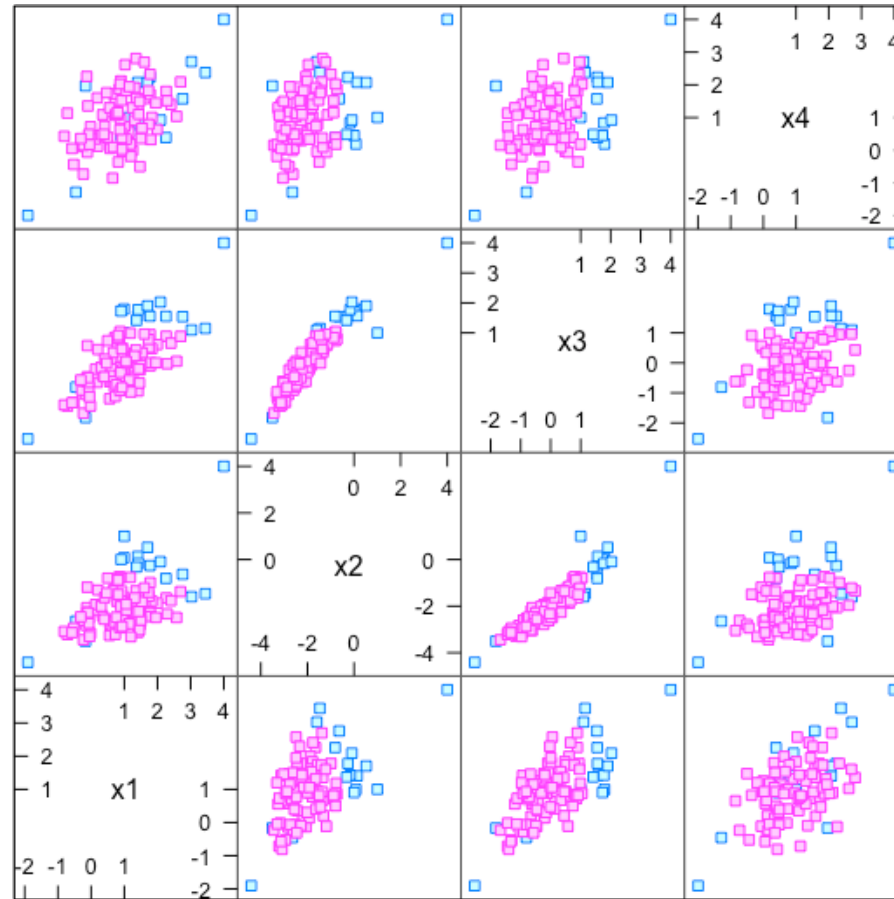




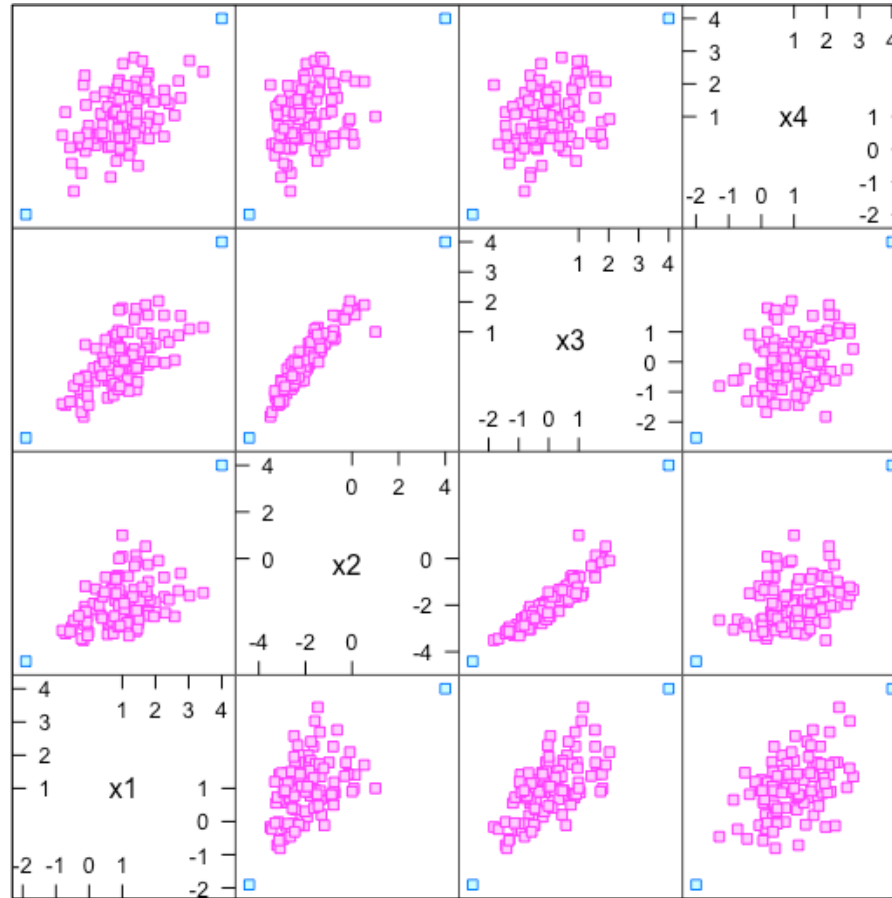
DBSCAN, d_2 , unscaled data, $\varepsilon = 0.4$, $\text{minPts} = 4$



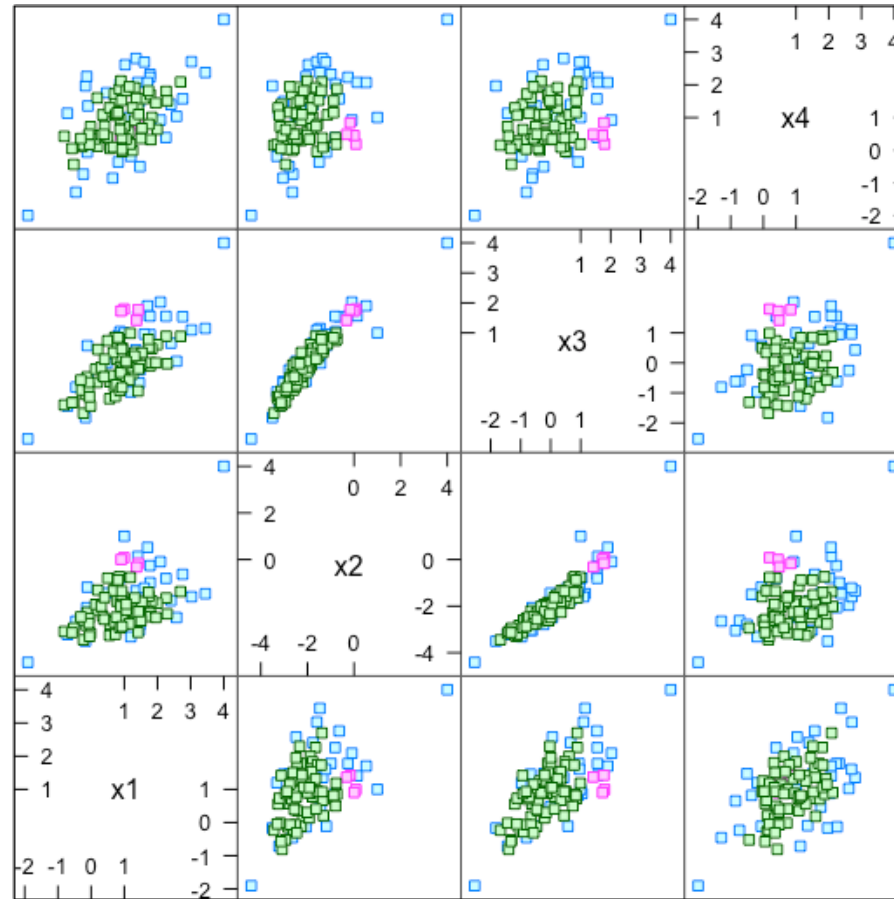
DBSCAN, d_2 , unscaled data, $\varepsilon = 1$, minPts = 4



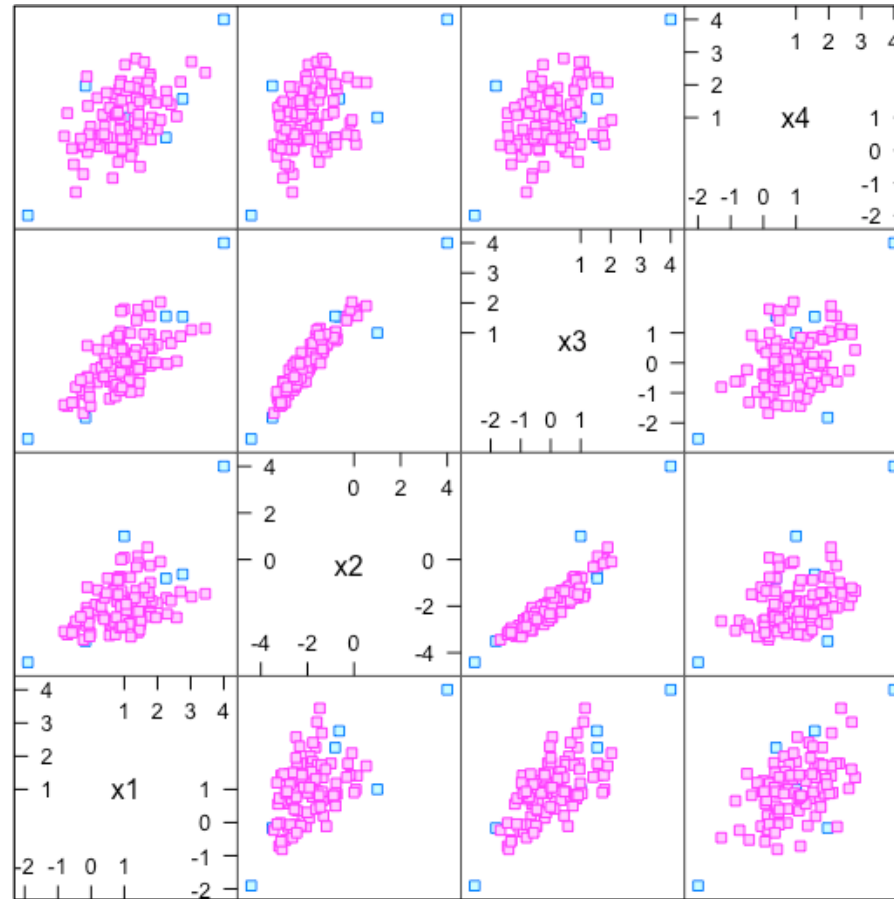
DBSCAN, d_2 , unscaled data, $\varepsilon = 1$, minPts = 8



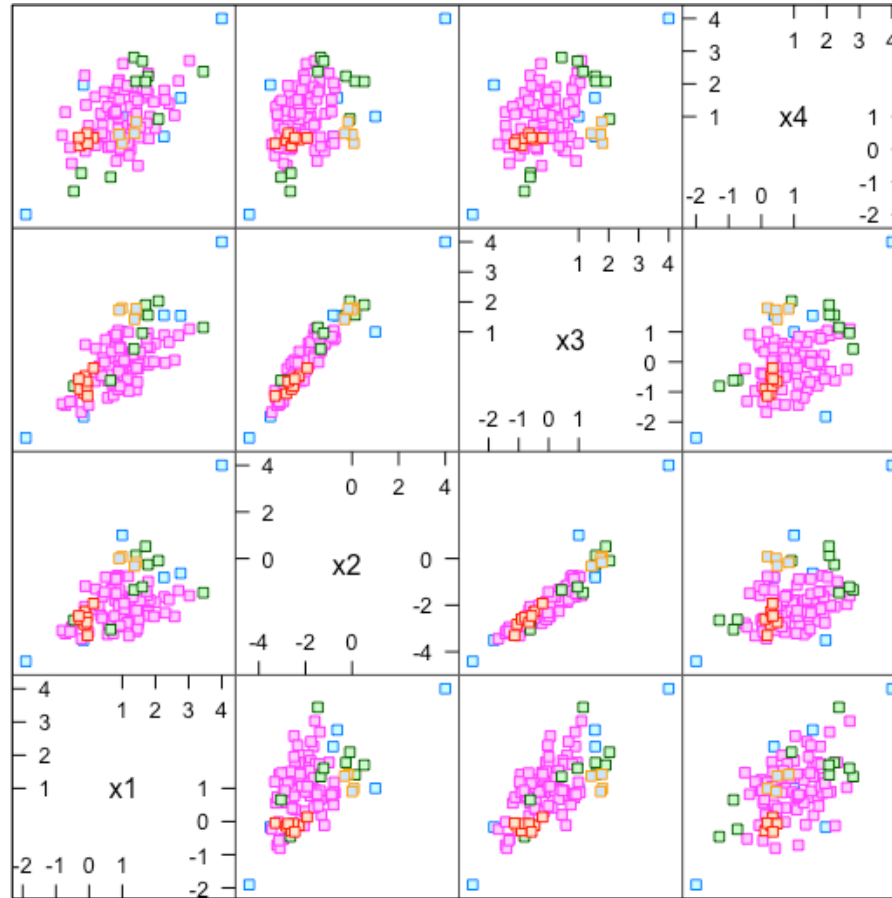
DBSCAN, d_2 , unscaled data, $\epsilon = 2$, $\text{minPts} = 8$



HDBSCAN, d_2 , unscaled data, minPts = 4



OPTICS, d_2 , unscalled data, $\varepsilon = 1$, $\text{minPts} = 4$, $\varepsilon_{cl} = 1$



OPTICS, d_2 , unscaled data, $\varepsilon = 1$, $\text{minPts} = 4$, $\varepsilon_{cl} = 1$, $\xi = 0.05$

Isolation Forest

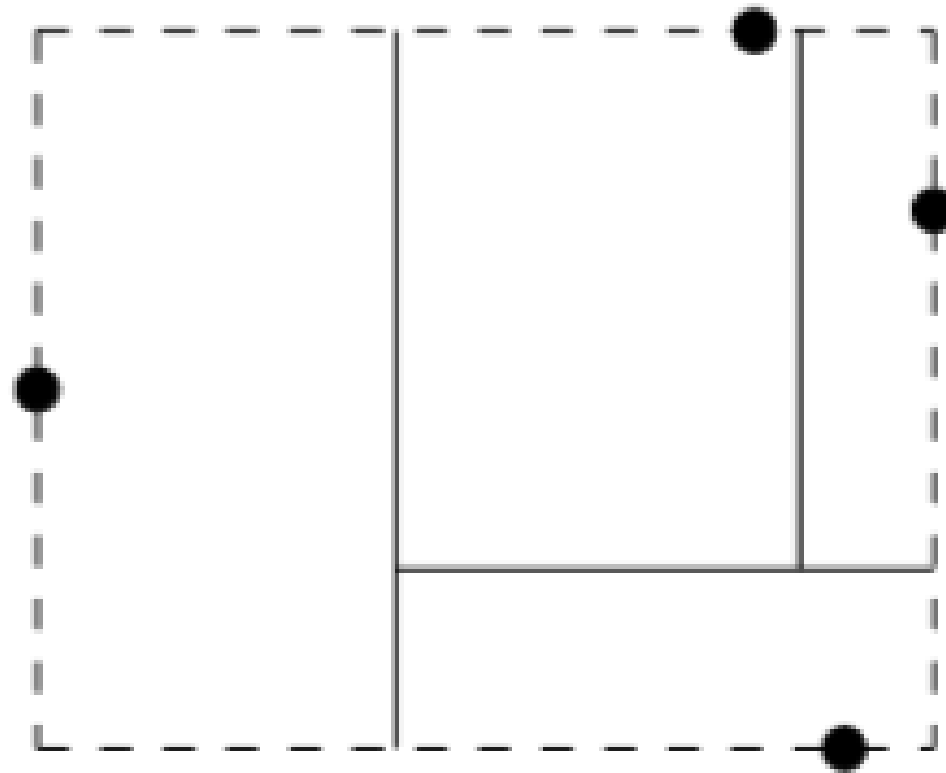
Both the LOF and the DBSCAN approach first construct models of what normal points look like, and then identify points that do not fit this model.

The **isolation forest** (IsoForest) algorithm tries instead to explicitly identify outliers under the assumptions that:

- there are **few** outliers, and
- that these outliers have **very different attributes** compared to normal observations.

IsoForest uses sampling techniques that increase algorithmic speed while decreasing memory requirements.

IsoForest tries to **isolate** anomalous points by **randomly** selecting an attribute and a split value between that attribute's min/max values, continuing until every point is alone in its component.



This recursive partitioning yields an **Isolation Tree** (IsoTree):

- the **root** of this tree is the entire dataset;
- each **node** is a subset of the observations;
- each **branch** corresponds to one of the generated partitions, and
- the **leaves** are sets containing a single isolated point.

Each point is then assigned a score derived from **how deep in the tree** its singleton partition appears.

Points that are **shallower** in the tree are easier to separate from the rest \implies likely **outliers**?

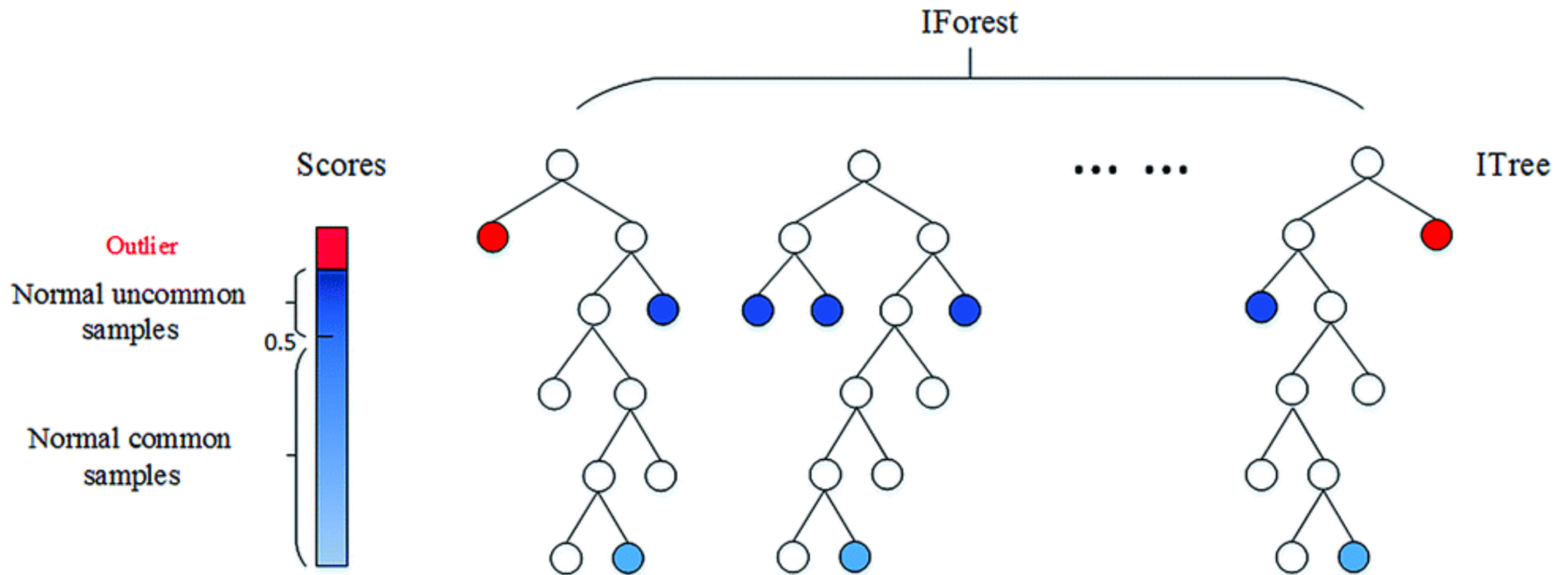
The points of interest are shallow: once the height of the tree has reached a **given threshold** (expected height of a random binary tree?), stop growing the tree (reduces computational cost).

IsoTrees can be constructed from subsets: the location of any point within this smaller tree can be estimated (reduces computational cost).

A collection of IsoTrees forms an **Isolation Forest**.

An IsoForest **score** can be computed for each point: search each tree for the point's location and record the path length required to reach it. The score is simply the average path length (it can be **normalized** to make it independent of the dataset's size); low scores \implies outliers.

The formal procedure is provided in Algorithms 3 and 4.



Isolation Forest schematics [Baron].

Algorithm 3: Recursive Isolation Tree Construction: $iTree(D)$

```

1 Input: dataset  $D$ 
2 if  $|D| \leq 1$  then
3   | return  $\{\}$ 
4 end
5 else
6   | Let  $\bar{A}$  be a list of attributes in  $D$ 
7   | Randomly select an attribute  $A \in \bar{A}$ 
8   | Randomly sample a point  $s$  from
   |  $[\min_{\mathbf{q} \in D} A(\mathbf{q}), \max_{\mathbf{q} \in D} A(\mathbf{q})]$ 
9   | Return
   | Node  $\begin{cases} \text{LeftChild} & = iTree(\{\mathbf{q} \in D : A(\mathbf{q}) \leq s\}) \\ \text{RightChild} & = iTree(\{\mathbf{q} \in D : A(\mathbf{q}) > s\}) \\ \text{NodeValue} & = D \end{cases}$ 
10 end
11 Output: Binary tree with node values that are
    subsets of  $D$ 

```

Algorithm 4: Isolation Forest

```
1 Input: dataset  $D$ , integer  $t$  number of Isolation  
   Trees  
2  $Forest = \{\}$   
3 for  $i = 1$  to  $t$  do  
4   |  $Tree = iTree(D)$   
5   | Add  $Tree$  to  $Forest$   
6 end  
7 for  $p \in D$  do  
8   |  $PathLengths = \{\}$   
9   | for  $Tree$  in  $Forest$  do
```

```
10   |   Find the path length  $\ell$  from the root of Tree  
    |   to node  $\{\mathbf{p}\}$   
11   |   Add  $\ell$  to PathLengths  
12   |   end  
13   |    $AveragePathLength = \frac{\sum_{\ell \in PathLengths} \ell}{t}$   
14   |   Set  $a(\mathbf{p}) = 2^{-\frac{AveragePathLength}{c(|D|)}}$   
15   |   end  
16   |   Output: Anomaly score  $a(\mathbf{p}) \in [0, 1]$  for each  
    |    $\mathbf{p} \in D$ 
```

With $|D| = n$, it can be shown (not obvious!) that the expected length to a random point in an IsoTree is

$$c(n) = 2H(n - 1) - \frac{2(n - 1)}{n},$$

where $H(n)$ is the $(n)^{\text{th}}$ harmonic number: $H(n) \approx \ln n + 0.577$.

The normalized anomaly score of \mathbf{p} in the IsoForest, $a(\mathbf{p})$, is

$$\log_2 a(\mathbf{p}) = - \frac{\text{average path length to } \mathbf{p} \text{ in the Isolation Trees}}{c(n)}.$$

If $a(\mathbf{p}) \approx 1$, we label \mathbf{p} an **anomaly**; if $a(\mathbf{p}) \leq 0.5$, a **regular observation**.
If every point receives a score around 0.5, there are no outright outlier.

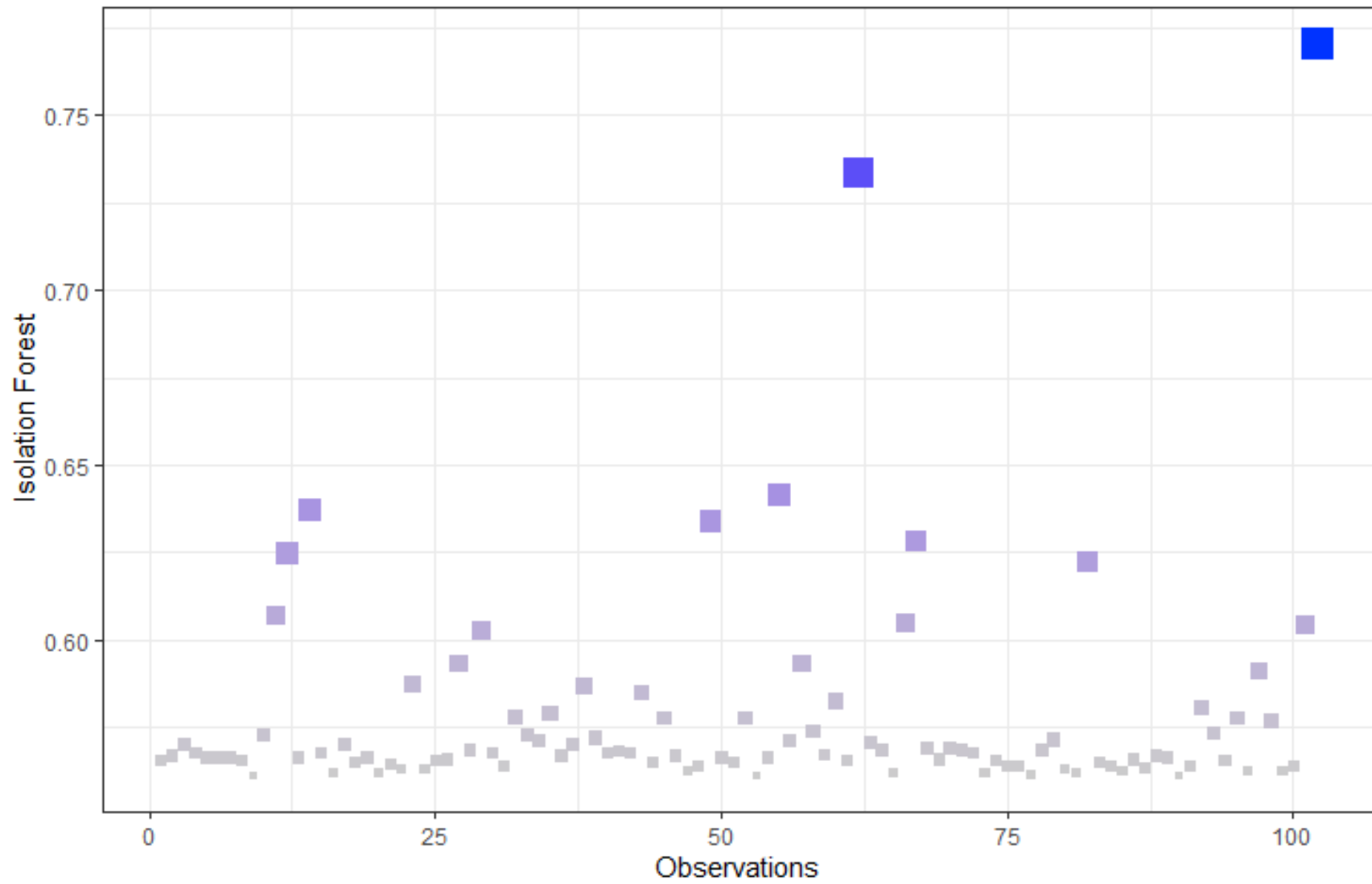
Strengths:

- small time and memory requirements;
- can handle high dimensional data;
- do not need labeled anomalies in the training set.

Main Limitation:

- anomaly score can have high variance over multiple runs.

In general, density-based schemes are more powerful than distance-based schemes when a dataset contains patterns with diverse characteristics, but less effective when the patterns are of comparable densities with the outliers.



5.3 – Qualitative Methods

Categorical (or qualitative) variables are one whose levels are measured on a nominal scale.

Examples include:

- the mother tongue of an individual,
- her hair colour,
- her age,
- and so forth.

The **central tendency** of the values of such a variable is its **mode**.

Measures of **spread** are much more difficult to define in a consistent manner. One possibility: use the proportion of levels with more than a certain percentage of the observations above a given threshold.

Example: consider a dataset with $n = 517$ individuals and $p = 3$ features:

- **age** (25–, 25 – 44, 45 – 64, 65+);
- **mother tongue** (French, English, Mandarin, Arabic, Other), and
- **hair colour** (black, brown, blond, red).

Their respective modes are 25 – 44, English, and brown. And their spread?

Age	Mother Tongue	Hair Colour				Mother Tongue	Hair Colour				Hair Colour						
		Black	Brown	Blond	Red		Black	Brown	Blond	Red	Black	Brown	Blond	Red			
24-	French	11	24	12	2	French	34	70	32	5	187	217	79	34			
	English	12	44	3	6	English	40	111	13	18	36.2%	42.0%	15.3%	6.6%			
	Mandarin	16	2	1	0	Mandarin	46	6	2	0							
	Arabic	9	1	0	0	Arabic	30	4	0	2							
	Other	11	7	13	1	Other	37	26	32	9							
25-44	French	15	32	17	2	Age				Mother Tongue							
	English	21	47	8	7	Hair Colour				French	English	Mandarin	Arabic	Other			
	Mandarin	23	3	1	0	24-	59	78	29	9	141	182	54	36	104		
	Arabic	15	2	0	2	25-44	89	100	38	17	27.3%	35.2%	10.4%	7.0%	20.1%		
	Other	15	16	12	6	45-64	23	33	5	5							
45-64	French	7	12	2	1	65+	16	6	7	3	Age						
	English	4	17	2	3					24-	25-44	45-64	65+				
	Mandarin	3	1	0	0					175	244	66	32				
	Arabic	3	1	0	0					33.8%	47.2%	12.8%	6.2%				
	Other	6	2	1	1					Total Number of Observations:		517					
65+	French	1	2	1	0	Mother Tongue				Age				Percentage of Levels Above:			
	English	3	3	0	2	French	49	66	22	4					15%	25%	
	Mandarin	4	0	0	0	English	65	83	26	8					Hair Colour	75%	50%
	Arabic	3	0	0	0	Mandarin	19	27	4	4					Mother Tongue	60%	60%
	Other	5	1	6	1	Arabic	10	19	4	3					Age	50%	50%
						Other	32	49	10	13							

Qualitative features are often associated to numerical values: in R, for instance, there is a difference between factor **levels** and factor **labels**.

Categorical variables with numerical levels are treated as **ordinal** variables.

 **These should not be interpreted as numerals!**

If we use the code “red” = 1, “blond” = 2, “brown” = 3, and “black” = 4 to represent hair colour, we **cannot conclude** that “blond” > “red”, even though $2 > 1$, or that “black” – “brown” = “red”, even though $4 - 3 = 1$.

A categorical variable that has exactly two levels is a **dichotomous** (binary) variable; a variable with more than two levels is **polytomous**.

Regression on categorical variables \implies **multinomial logistic regression**.

Distances (apart from the 0 – 1 distance and the related Hamming distance) require numerical inputs.

But representing categorical variables with numerical features can lead to traps (see previous slide).

Anomaly detection methods based on distance or on density are not recommended in the qualitative context (unless the distance function has been modified appropriately).

Another option is to look at combinations of feature levels, but this can be computationally expensive.

5.3.1 – AVF Algorithm

The **Attribute Value Frequency** (AVF) algorithm is a fast and simple way to detect outlying observations in categorical data.

It can be done without having to create or search through various combinations feature levels (which increase the search time).

Intuitively, outlying observations are points which occur relatively infrequently in the (categorical) dataset; an “ideal” anomalous point is one for which **each feature value is extremely anomalous** (or relatively infrequent).

The **rarity** of an attribute level can be measured by summing the number of times the corresponding feature takes that value in the dataset.

Let's say that there are n observations in the dataset: $\{\mathbf{x}_i\}$, $i = 1, \dots, n$, and that each observation is a collection of m features.

We write

$$\mathbf{x}_i = (x_{i,1}, \dots, x_{i,\ell}, \dots, x_{i,m}),$$

where $x_{i,\ell}$ is \mathbf{x}_i 's ℓ th feature's level.

Example: in the previous example, we may have

$$\mathbf{x}_1 = (x_{1,1}, x_{2,1}, x_{3,1}) = (24-, \text{French}, \text{blond})$$

⋮

$$\mathbf{x}_{517} = (x_{517,1}, x_{517,1}, x_{517,1}) = (24-, \text{Mandarin}, \text{blond}).$$

The **AVF score** of an observation \mathbf{x}_i is

$$\text{AVFscore}(\mathbf{x}_i) = \frac{1}{m} \sum_{\ell=1}^m f(x_{i,\ell}),$$

where $f(x_{i,\ell})$ is the number of dataset observations \mathbf{x} for which the ℓ th feature takes on the level $x_{i,\ell}$.

A **low** AVF score indicates that the observation is likely to be an **outlier**.

An “ideal” anomalous observation minimizes the AVF score – reached when the observation’s features’ levels occurs only once in the dataset.

For an integer k , the suggested outliers are the k observations with smallest AVF scores. The formal procedure is provided in Algorithm 5.

Algorithm 5: AVF

```
1 Inputs: dataset  $D$  ( $n$  observations,  $m$  features),  
   number of anomalous observations  $k$   
2 while  $i \leq n$  do  
3    $j = 1$   
4   AVFscore( $\mathbf{x}_i$ ) =  $f(x_{i,j})$   
5   while  $j \leq m$  do  
6     AVFscore( $\mathbf{x}_i$ ) = AVFscore( $\mathbf{x}_i$ ) +  $f(x_{i,j})$ ;  
7      $j = j + 1$   
8   end  
9   AVFscore( $\mathbf{x}_i$ ) = Mean(AVFscore( $\mathbf{x}_i$ ))  
10   $i = i + 1$   
11 end  
Outputs:  $k$  observations with smallest AVF scores
```

Age	Mother Tongue	Hair Colour				Age	Mother Tongue	Hair Colour			
		Black	Brown	Blond	Red			Black	Brown	Blond	Red
24-	French	167.7	177.7	131.7	116.7	45-64	French	131.3	141.3	95.3	80.3
	English	181.3	191.3	145.3	130.3		English	145.0	155.0	109.0	94.0
	Mandarin	138.7	148.7	102.7	87.7		Mandarin	102.3	112.3	66.3	51.3
	Arabic	132.7	142.7	96.7	81.7		Arabic	96.3	106.3	60.3	45.3
	Other	155.3	165.3	119.3	104.3		Other	119.0	129.0	83.0	68.0
25-44	French	190.7	200.7	154.7	139.7	65+	French	120.0	130.0	84.0	69.0
	English	204.3	214.3	168.3	153.3		English	133.7	143.7	97.7	82.7
	Mandarin	161.7	171.7	125.7	110.7		Mandarin	91.0	101.0	55.0	40.0
	Arabic	155.7	165.7	119.7	104.7		Arabic	85.0	95.0	49.0	34.0
	Other	178.3	188.3	142.3	127.3		Other	107.7	117.7	71.7	56.7

AVF scores; 10 lowest scores highlighted (in red).

$$\begin{aligned}
 \text{AVFscore}(24-, \text{French}, \text{blond}) &= \frac{1}{3}(f(24-) + f(\text{French}) + f(\text{blond})) \\
 &= \frac{1}{3}(175 + 141 + 79) = 131.7
 \end{aligned}$$

5.3.2 – Greedy Algorithm

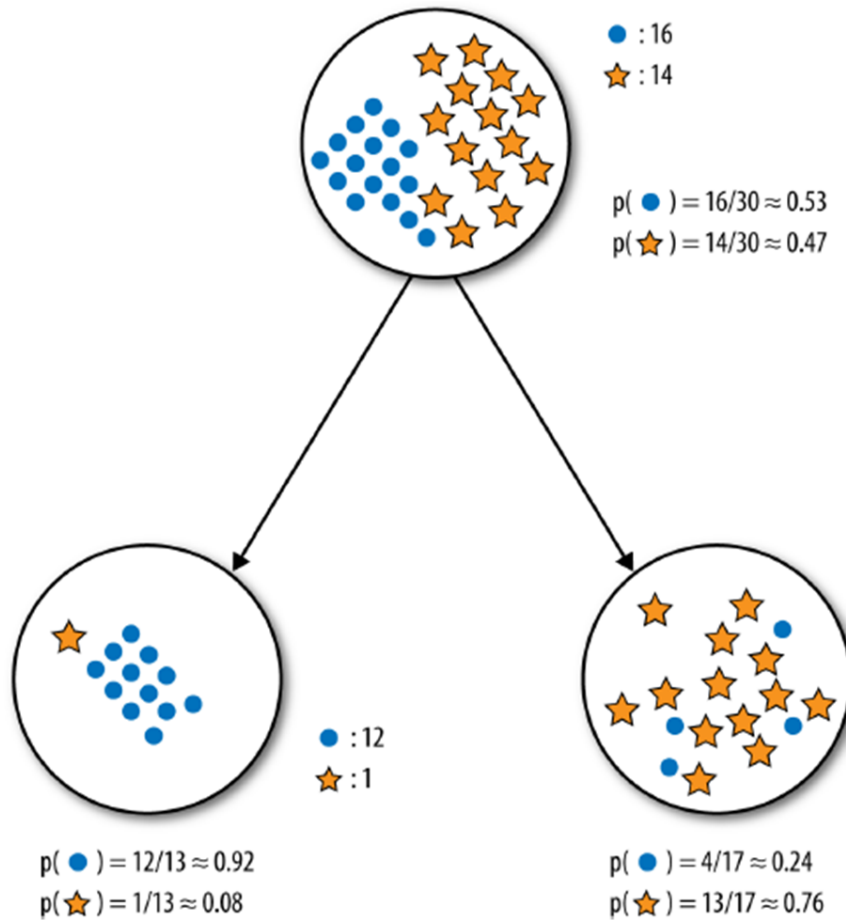
The **greedy** algorithm identifies a set OS of **candidate** anomalous observations in an efficient manner.

The **entropy of a set** $\Sigma \subseteq D$ is a measure of the **disorder** in Σ . Let X be a feature of D ; the set of levels that X takes on Σ is denoted by

$$S(X; \Sigma) = \{z | X = z, \mathbf{x} \in \Sigma\}.$$

Let $p_X(z)$ be the % of observations in Σ for which $X = z$. The **entropy of a feature** X on Σ is

$$H(X; \Sigma) = - \sum_{z \in S(X; \Sigma)} p_X(z) \log p_X(z).$$



$$E(S) = -p_o \log p_o - p_* \log p_*$$

$$= -\frac{16}{30} \log \frac{16}{30} - \frac{14}{30} \log \frac{14}{30} \approx 0.99$$

$$E(L) = -p_o \log p_o - p_* \log p_*$$

$$= -\frac{12}{13} \log \frac{12}{13} - \frac{1}{13} \log \frac{1}{13} \approx 0.39$$

$$E(R) = -p_o \log p_o - p_* \log p_*$$

$$= -\frac{4}{17} \log \frac{4}{17} - \frac{13}{17} \log \frac{13}{17} \approx 0.79$$

[Foster, Provost]

The mathematical formulation of the problem is simple: in order to find k anomalous observations in a dataset D , solve the optimization problem

$$OS = \arg \min_{O \subseteq D} \{H(D \setminus O)\}, \quad \text{subject to } |O| = k,$$

where the **entropy** $H(D \setminus O)$ is the sum of the entropy of each feature:

$$H(D \setminus O) = H(X_1; D \setminus O) + \cdots + H(X_m; D \setminus O)$$
$$H(X_\ell; D \setminus O) = - \sum_{z_\ell \in S(X_\ell; D \setminus O)} p(z_\ell) \log p(z_\ell),$$

where $S(X_\ell; D \setminus O)$ is the set of levels that the ℓ th feature takes in $D \setminus O$.

The greedy algorithm solves the optimization problem as follows:

1. The set of outlying and/or anomalous observations OS is initially set to be empty, and all observations of $D \setminus OS$ are identified as normal.
2. Compute $H(D \setminus OS)$.
3. Every normal observation \mathbf{x} is temporarily taken out of $D \setminus OS$ to create a subset D'_x , whose entropy $H(D'_x)$ is also computed.
4. The \mathbf{y} which provides the **maximal entropy impact** is added to OS:

$$\mathbf{y} = \arg \min_{\mathbf{x} \in D \setminus OS} \{H(D \setminus OS) - H(D'_x)\}.$$

5. Repeat steps 2-4 another $k - 1$ times to obtain a set OS of k candidate anomalous observations.

5.4 – Anomalies in High-Dimensional Datasets

In recent times, datasets have become quite **large** – they may contain **hundreds or thousands of features** (or more).

Conventional **proximity-based** anomaly detection methods can only be expected to work reasonably well when the sample size n is larger than the dimension p ($n > p$).

In **high-dimensional data** ($n < p$), the main problem is an off-shoot of the **curse of dimensionality** (CoD): observations are often **isolated** and **scattered** (or sparse); the notion of proximity fails to maintain its relevance.

High-dimensional anomaly detection methods are linked with dimension reduction and feature selection methods.

5.4.1 – Definitions and Challenges

The challenges of anomaly and outlier detection in **high-dimensional data** (HDD) are due to:

- the notion of distance failing to retain its **relevance** due to the CoD (“the problem of detecting outliers is like finding a needle in a haystack”);
- all points in HDDs **tend to be outliers**, and
- datasets become more **sparse** as the dimension of the feature space increases.

Good HDD anomaly detection methods should:

- allow for **effective management** of sparse data issues;
- provide **interpretability** of the discrepancies (i.e. how is the behaviour of such observations different than the behaviour from regular ones?);
- allow anomaly measurements to be **compared** (“apples-to-apples”), and
- consider the **local data behaviour** to determine whether an observation is abnormal or not.

5.4.2 – Projection-Based Methods

One approach to mitigate the effects of the CoD on conventional anomaly/outlier detection methods in **high dimension, low sample size** (HDLSS) datasets is to **reduce the dimensionality** of the dataset while preserving its essential characteristics.

Such projection-based methods include:

- **principal component analysis,**
- **independent component analysis,**
- **feature selection,** etc. – see *Feature Selection and Dimension Reduction* module/report for more examples.

Principal Components Analysis

Principal component analysis (PCA) can be used to find the combinations of variables along which the data points are **most spread out**.

Geometrically, the procedure fits the “best” p -**ellipsoid** to a centered representation of the data.

The ellipsoid axes are the **principal components** of the data.

Small axes are components along which the variance is “small”; removing these components can lead to a “small” loss of information.

There are scenarios where it could be those “small” axes that are more interesting – such as the “pancake stack” problem.

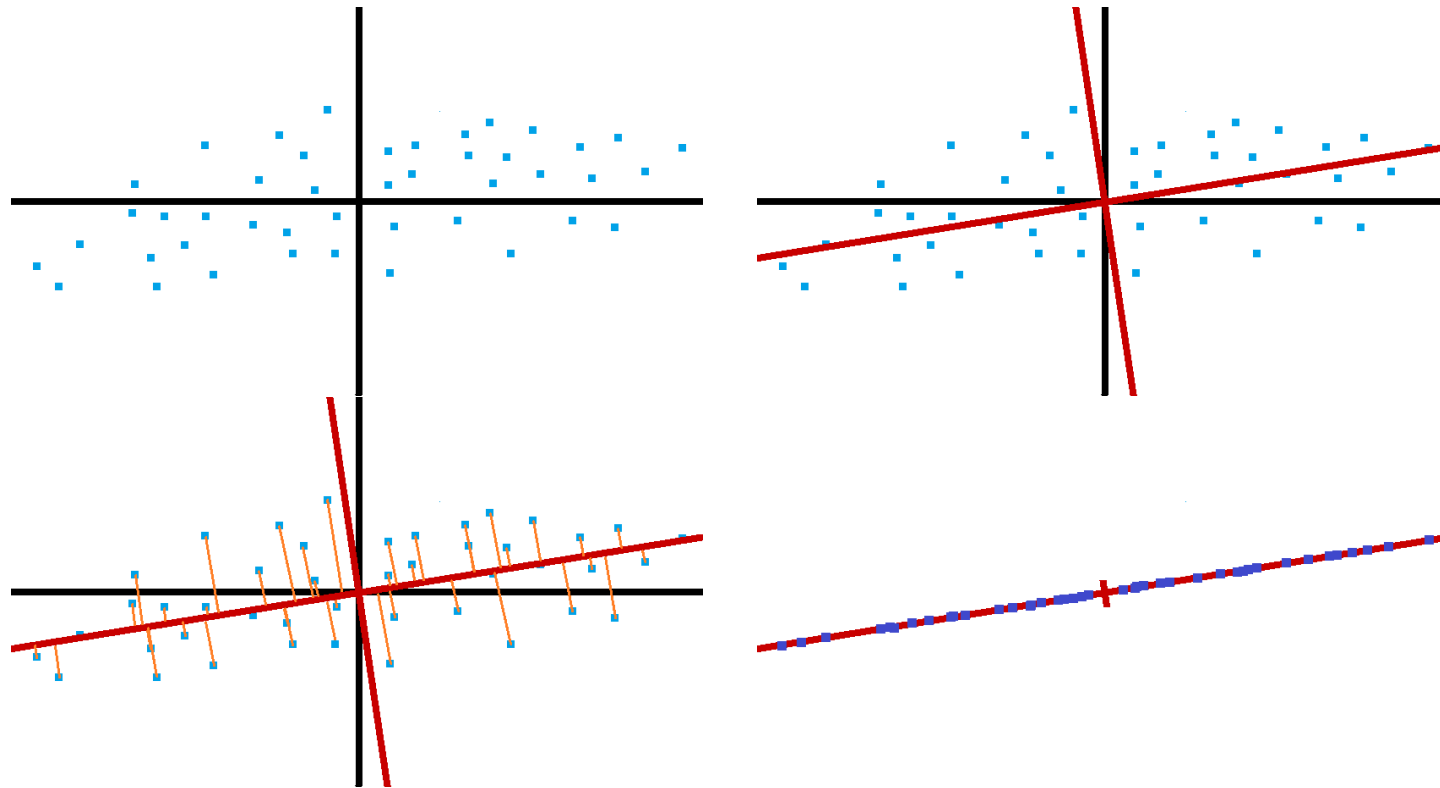


Illustration of PCA on an artificial 2D dataset. The red axes provide the best elliptic fit. Removing the minor axis by projecting the points on the major axis leads to dimension reduction and a (small) loss of information.

PCA Procedure:

1. centre and “scale” the data to obtain a matrix \mathbf{X} ;
2. compute the data’s “covariance matrix” $\mathbf{K} = \mathbf{X}^T \mathbf{X}$;
3. compute \mathbf{K} ’s eigenvalues, $\mathbf{\Lambda}$ (ordered diagonal matrix), and its orthonormal eigenvectors matrix \mathbf{W} ;
4. each eigenvector \mathbf{w} (also known as **loading**) represents an axis, whose variance is given by the associated eigenvalue λ .

Note that $\mathbf{K} \geq 0 \implies \mathbf{\Lambda} \geq 0$.

The **first principal component** PC_1 is the eigenvector \mathbf{w}_1 of \mathbf{K} associated to its largest eigenvalue λ_1 , and the variance of the data along \mathbf{w}_1 is proportional to λ_1 .

The **second principal component** PC_2 is the eigenvector \mathbf{w}_2 of \mathbf{K} associated to its second largest eigenvalue $\lambda_2 \leq \lambda_1$, and the variance of the data along \mathbf{w}_1 is proportional to λ_2 , and so on.

Final Result: $r = \text{rank}(\mathbf{X})$ **orthonormal** principal components

$$PC_1, \dots, PC_r.$$

If some of the eigenvalues are 0, $r < p$, and *vice-versa* \implies data is embedded in a r -dimensional subspace in the first place.

PCA can provide an avenue for dimension reduction by “removing” components with small eigenvalues.

The **proportion of the spread in the data** which can be explained by each PC can be placed in a **scree plot** (eigenvalues against ordered component indices), and retain the ordered PCs:

- for which the eigenvalue is above some threshold (say, 25%);
- for which the cumulative proportion of the spread falls below some threshold (say 95%), or
- prior to a **kink** in the scree plot.

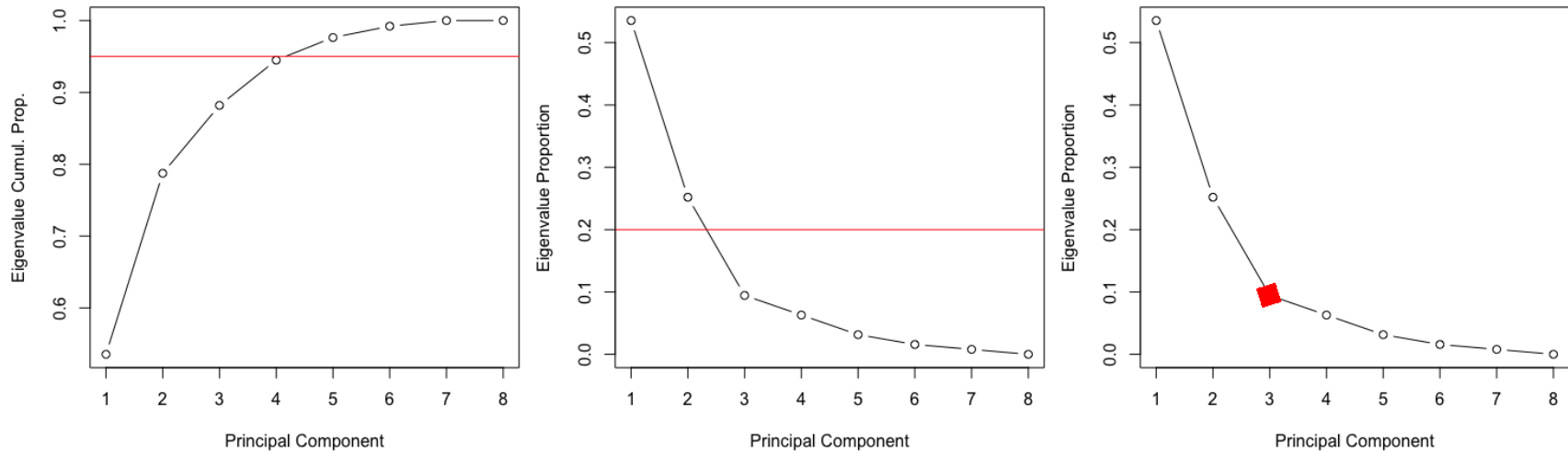
Example: consider an $8D$ dataset for which the ordered PCA eigenvalues are

PC	1	2	3	4	5	6	7	8
Var	17	8	3	2	1	0.5	0.25	0
Prop	54	25	9	6	3	2	1	0
Cumul	54	79	88	94	98	99	100	100

If only the PCs that explain up to 95% of the **cumulative variance** are retained, the original dataset reduces to a $4D$ subset.

If only the PCs that **individually explain** more than 25% of the variance are retained, the original dataset reduces to a $2D$ subset.

If only the PCs that lead into the **first kink** in the scree plot are retained, the original dataset reduces to a $3D$ subset.

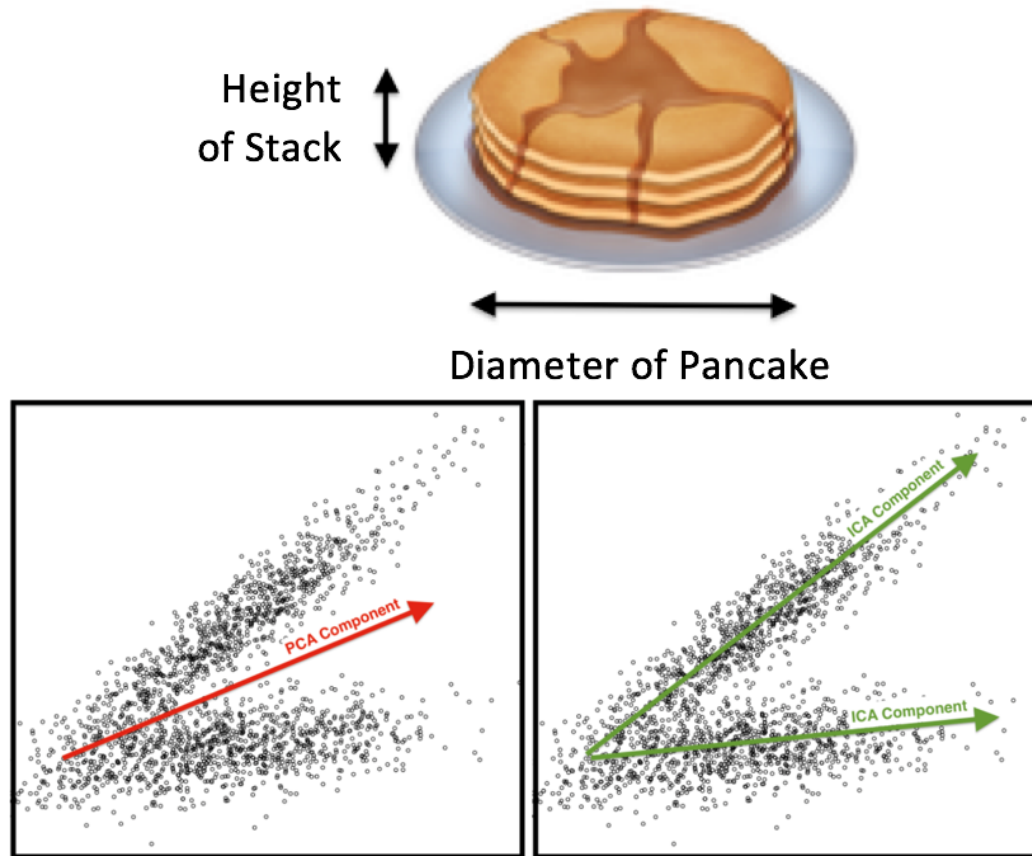


The proportion of the variance explained by each (ordered) component is shown in the first 3 charts; the cumulative proportion is shown in the last chart.

The cumulative proportion method is shown in the first image, the individual threshold method in the second, and the kink method in the third.

PCA Limitations:

- dependent on scaling, and so not unique;
- interpreting the PCs require domain expertise;
- (quite) sensitive to outliers;
- analysis goals not always aligned with the PCs, and
- data assumptions not always met – does it always make sense that important data structures and data spread be linked (see **counting pancakes** problem), or that the PCs be **orthogonal**?



(algobeans.com)

PCA is used in various contexts:

- as a dimension reduction method used during data pre-processing;
- as a data visualization aid, and
- as an anomaly and outlier detection method.

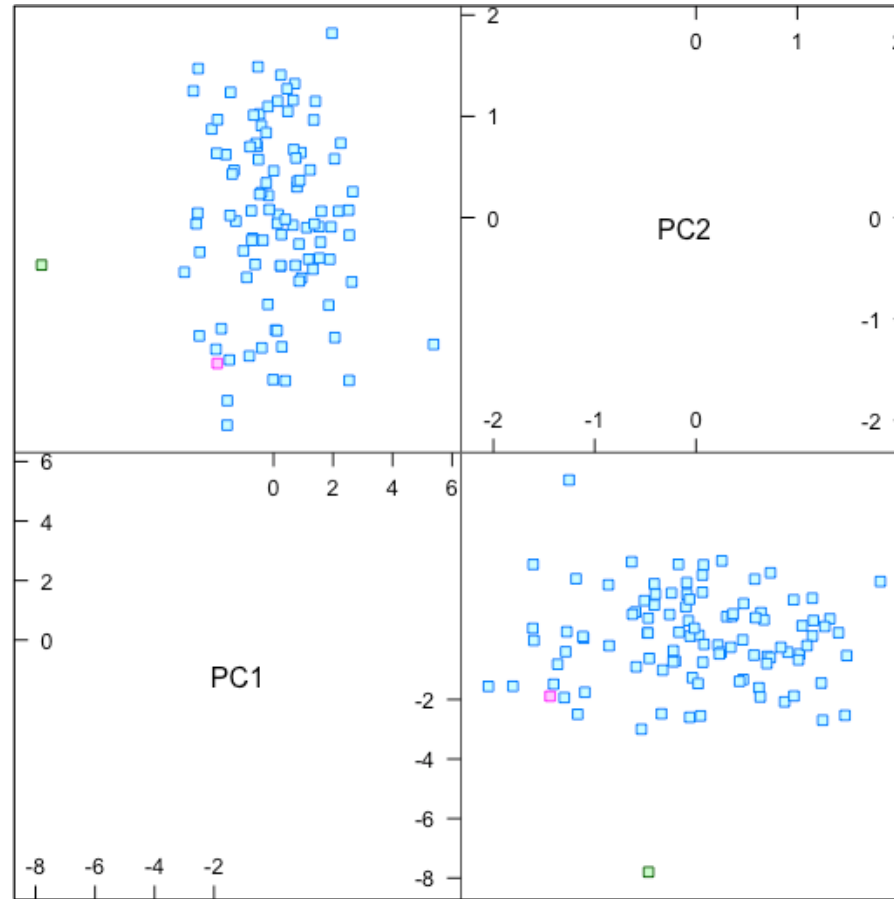
The **quality** of the PCA results is strongly dependent on the **number of retained principal components** (= the dimension k of the subspace on which the observations are projected).

For anomaly detection purposes, it is not obvious that the methods shown prior to find an optimal k are appropriate \implies a good k is one which allows for good anomaly detection.

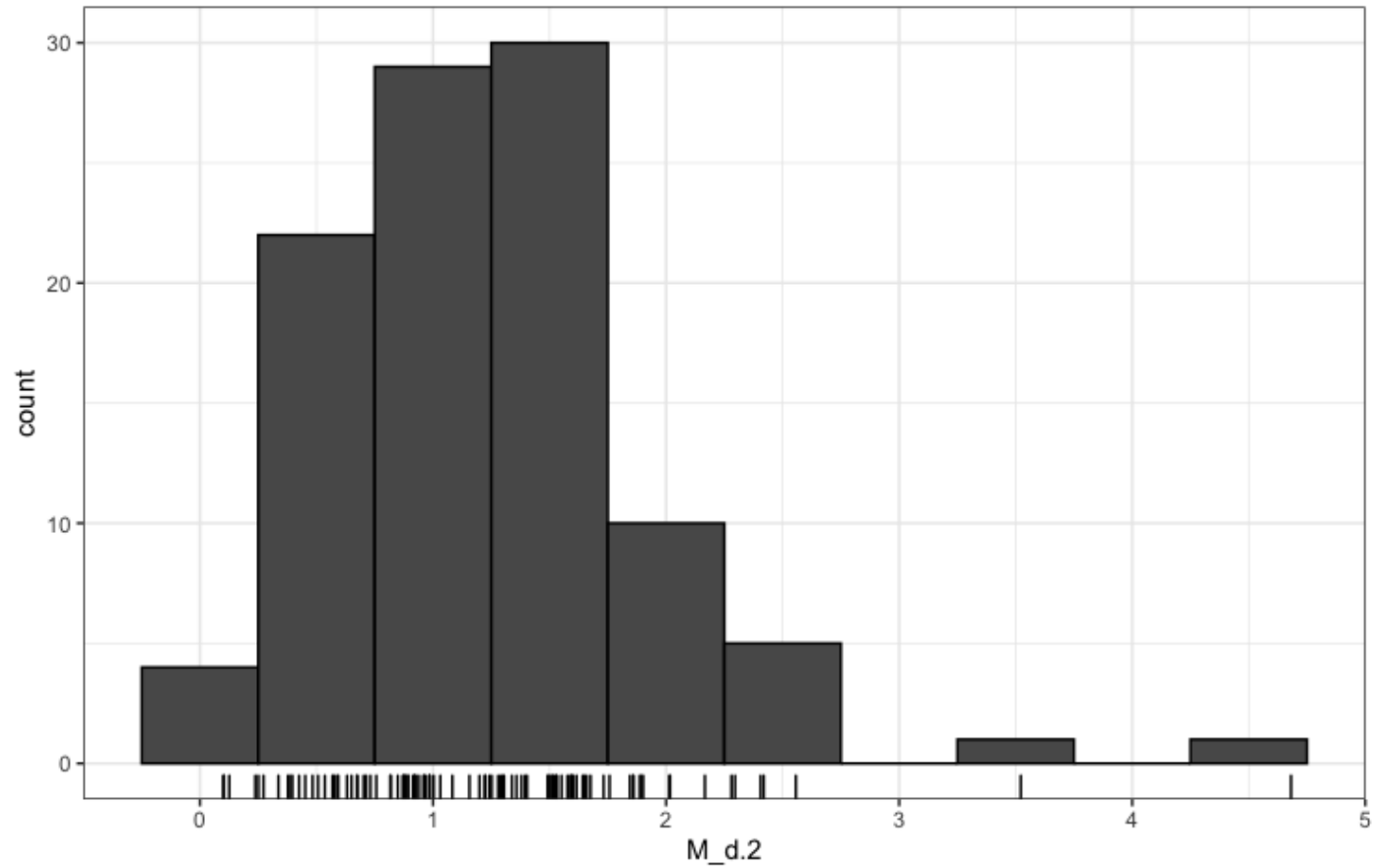
There are other PCA-associated dimension reduction methods: ICA, singular value decomposition, kernel PCA, etc.

What is the link with anomaly and/or outlier detection?

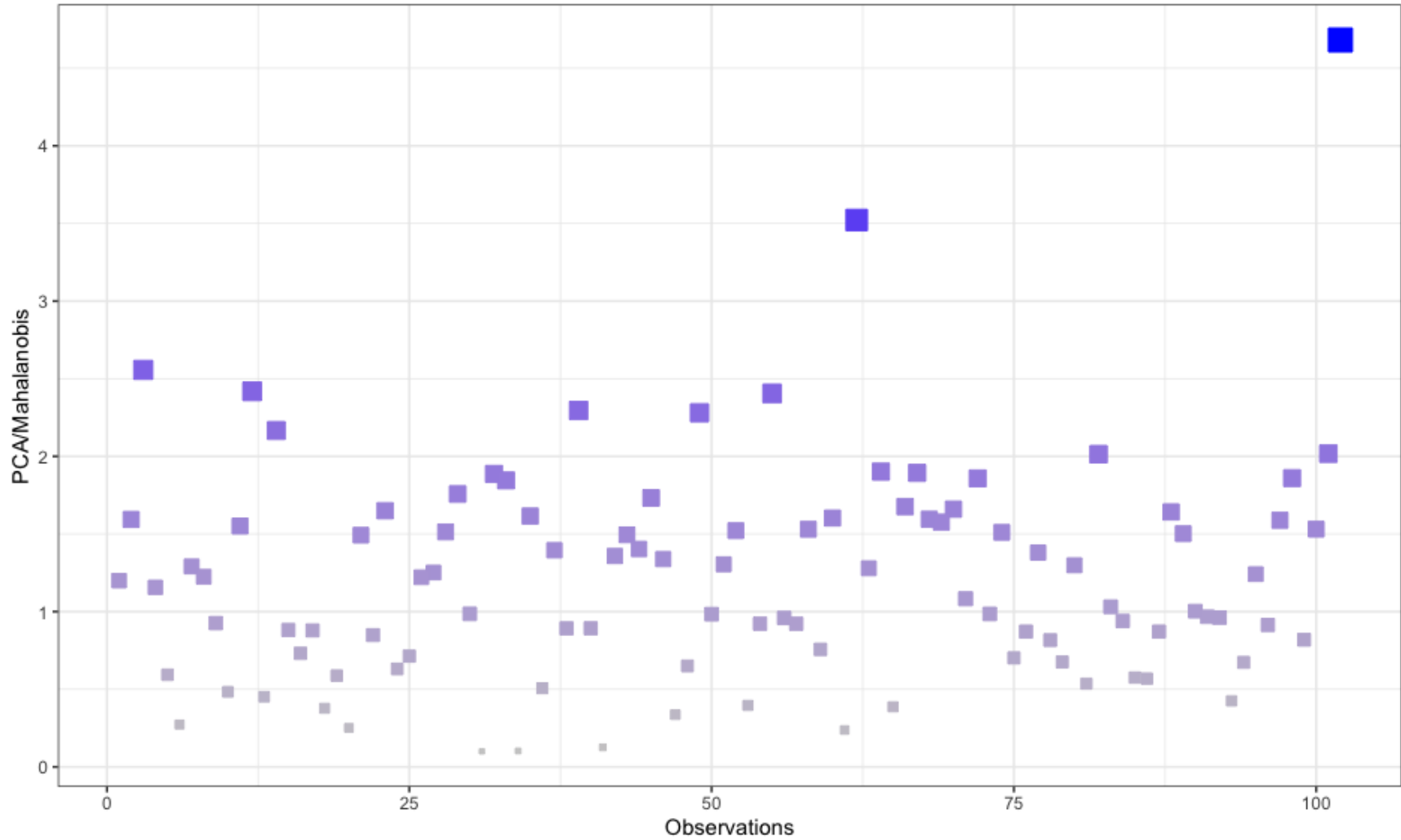
- Once the dataset has been projected on a lower-dimensional subspace, the curse of dimensionality is **mitigated** – traditional methods are applied to the **projected data**.
- Dimension reduction usually leads to a loss of information, which can affect the accuracy of the detection procedure – especially if the presence/absence of anomalies is **not aligned** with the dataset's principal components.



2D PCA projection.



Histogram of 2D PCA Mahalanobis scores.



Distance-Based Outlier Basis Using Neighbours

Main problem with using PCA-type projections for anomaly detection: there may not be a correlation between the axes of heightened variance and the presence or absence of anomalies.

The **distance-based outlier basis using neighbours** algorithm (DOBIN) builds a basis which is better suited for the eventual detection of outlying observations. DOBIN's main idea is to search for nearest neighbours that are in fact relatively distant from one another:

1. start by building a space $\mathbf{Y} = \{\mathbf{y}_\ell\}$ which contains $M \ll n(n+1)/2$ vectors of the form

$$\mathbf{y}_\ell = (\mathbf{x}_i - \mathbf{x}_j) \odot (\mathbf{x}_i - \mathbf{x}_j),$$

where \odot is the element-by-element Hadamard multiplication, and for which the 1–norm

$$\|\mathbf{y}_\ell\|_1 = (x_{1,1} - x_{2,1})^2 + \cdots + (x_{1,p} - x_{2,p})^2$$

is the square of the distance between $\mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}$;

2. the selection of each of the M observation pairs is made according to a complex procedure which only picks \mathbf{x}_i and \mathbf{x}_j if they are part of one another's k –neighbourhood, for $k \in \{k_1, \dots, k_2\}$;
3. the set \mathbf{Y} thus contains points for which $\|\mathbf{y}_\ell\|_1$ is relatively large, which is to say that the observations \mathbf{x}_i are \mathbf{x}_j fairly distant from one another even if they are k –neighbours of each other;

4. next, a basis $\{\eta_1, \dots, \eta_p\} \subset \mathbb{R}^p$ is built where each η_i is a unit vector given by a particular linear combination of points in \mathbf{Y} ; they can be found using a Gram-Schmidt-like procedure:

$$\begin{aligned}\mathbf{y}_{\ell_0} &= \mathbf{y}_\ell, \quad \ell = 1, \dots, M \\ \mathbf{y}_{\ell_{b-1}} &= \mathbf{y}_{\ell_{b-2}} - \langle \eta_{b-1} \mid \mathbf{y}_{\ell_{b-2}} \rangle, \quad \ell = 1, \dots, M \\ \eta_b &= \frac{\sum_{\ell=1}^M \mathbf{y}_{\ell_{b-1}}}{\left\| \sum_{\ell=1}^M \mathbf{y}_{\ell_{b-1}} \right\|_2},\end{aligned}$$

for $b = 1, \dots, p$;

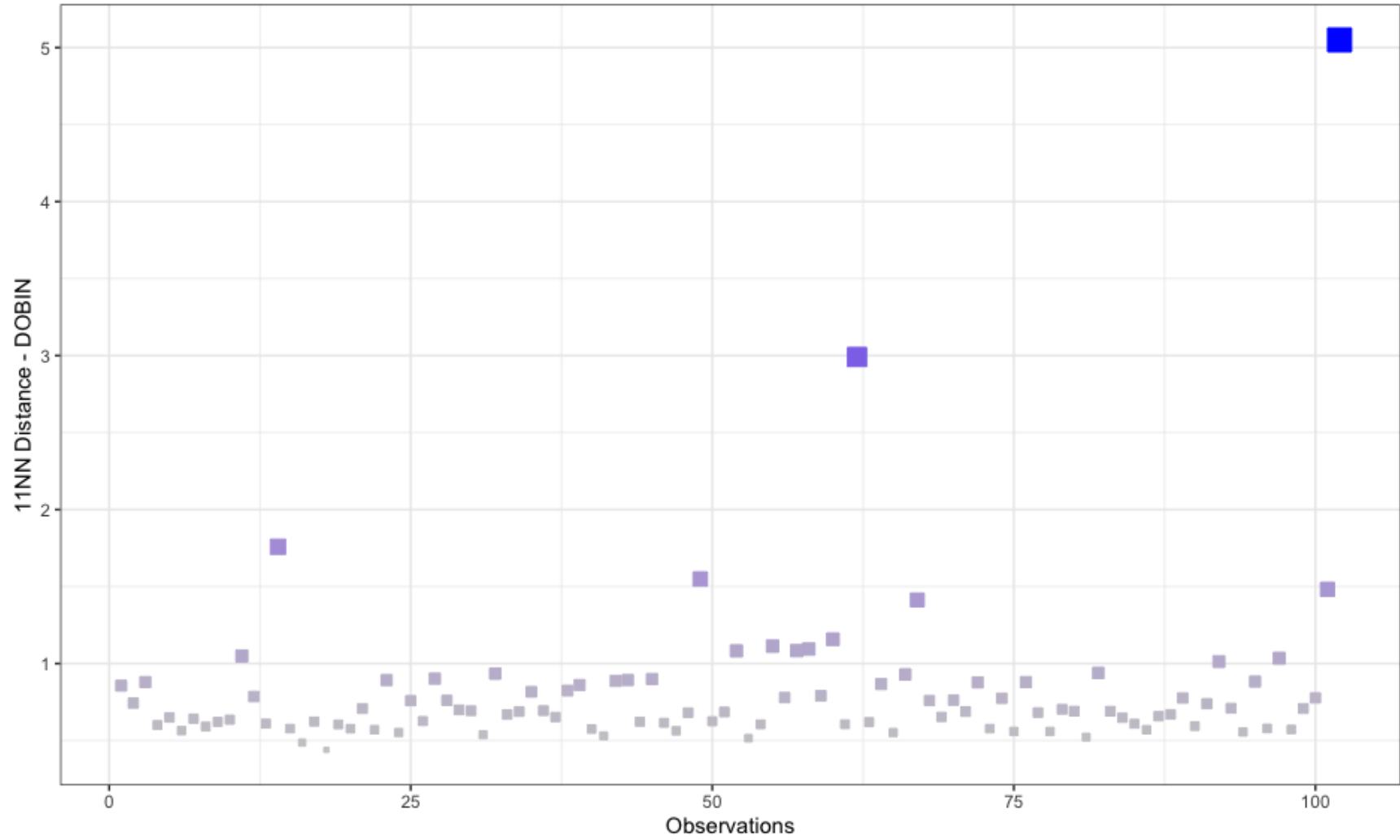
5. finally, transform the original dataset \mathbf{X} according to $\hat{\mathbf{X}} = \mathcal{T}(\mathbf{X})\Theta$, where $\mathcal{T}(\mathbf{X})$ normalizes each feature of \mathbf{X} according to a problem-specific scheme (Min-Max or Median-IQR, say), and

$$\Theta = [\eta_1 \mid \cdots \mid \eta_p]$$

is an orthogonal $p \times p$ matrix.

$\hat{\mathbf{X}}$ plays an analogous role to the subspace projection of \mathbf{X} in PCA – this is the object on which outlier and anomaly detection algorithms are applied.

In a nutshell, the first component provides the direction of largest k NN distance, instead of the direction of largest variance, and so forth. The paper by Kandanaarachchi and Hyndman provide more details.



5.4.3 – Subspace Methods

Subspace methods have been used effectively for anomaly and outlier detection in high-dimensional datasets.

In this context, a **subspace** is obtained by projecting the original dataset D on some collection of its features (looking at D only along some of its axes).

Strengths:

- eliminates **additive noise effects** of HDD;
- leads to more robust outliers (identified as such even when using different methods).

Limitation:

- difficult to solve effectively and efficiently \implies the potential # of subspace projections is exponential in # of features.

Feature bagging (FB) combines the results of the anomaly detection algorithm applied to **various subspaces** of the original data.

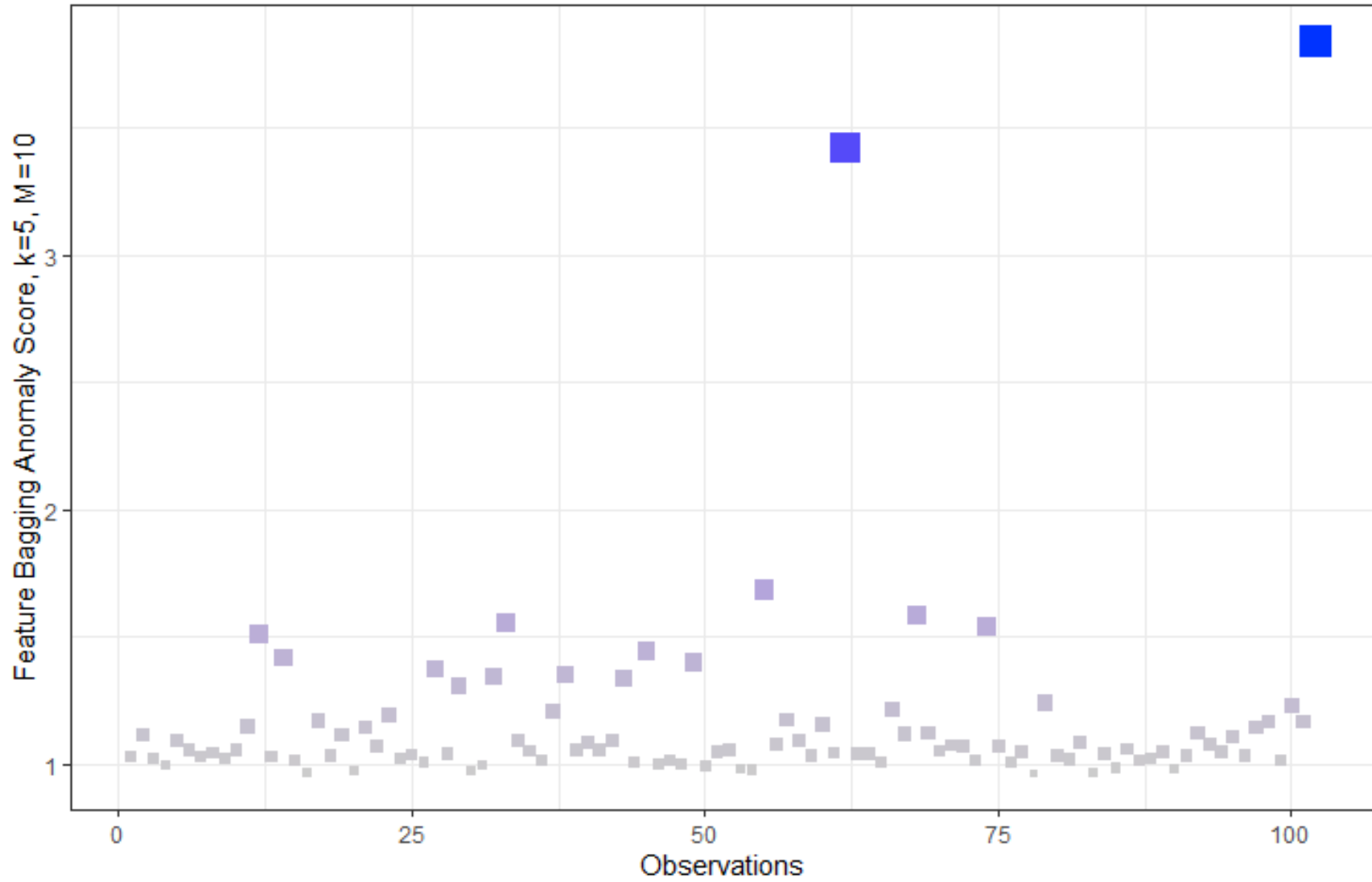
Officially, FB uses the Local Outlying Factor algorithm (LOF), but any fast anomaly detection algorithm can be used instead.

The anomaly scores and rankings from each run are aggregated as they are in the Independent Ensemble approach (cf. last Section of the slides).

The formal procedure is provided in Algorithm 8.


Algorithm 8: FeatureBagging

- 1 **Input:** dataset D
 - 2 $j = 1$;
 - 3 **while** *stopping criteria are not met* **do**
 - 4 Sample an integer r between $p/2$ et $p - 1$;
 - 5 Randomly select r features (variables) of D in order to create a projected dataset \tilde{D}_r in the corresponding r -dimensional sub-space;
 - 6 Compute the LOF result for each observation in the projected \tilde{D}_r ;
 - 7 $j = j + 1$;
 - 8 **end**
 - 9 **Output:** anomaly scores given by the independent ensemble method (average, minimal rank, etc.).
-



Other, more sophisticated, subspace anomaly detection methods include:

- **High-dimensional Outlying Subspaces (HOS);**
- **Subspace Outlier Degree (SOD)**, implemented in `HighDimOut`;
- **Projected Clustering Ensembles (OutRank);**
- **Local Selection of Subspace Projections (OUTRES)**, etc.

 **“No Free Lunch” Theorem:** there is no magic method – all methods have strengths and limitations, and the results depend heavily on the data.

Another list of algorithms is found at <https://pyod.readthedocs.io/en/latest/> it is far from complete.

5.5 – Advanced Topics

Anomaly detection and outlier analysis is still a young field, with a very active research community.

The challenges are numerous (we have highlighted some of them), and new algorithms come out nearly monthly.

An application to time series (using the S&P 500) is provided in the accompanying report, as are suggested exercises and projects (Airline Data, Distracted Driving Fatalities Data, Houseprice Data, etc.).

We wrap up this module with a discussion of outlier ensembles and of anomalies in text data.

5.5.1 – Outlier Ensembles

We have looked at various anomaly detection algorithms whose relative performance varies with the type of data being considered.

The **No Free Lunch theorem** reminds us that there is no specific algorithm that is guaranteed to outperform every other algorithm for all datasets.

The impact of algorithmic mismatch can be mitigated by using **ensemble methods**, where the results of several algorithms are considered before making a final decision.

We discuss two types of ensemble methods: **sequential ensembles** (boosting) and **independent ensembles**.

Sequential Ensembles

In **sequential ensembles**, a baseline algorithm is applied to a dataset in a sequential manner.

At each step, the weight associated with each observation is modified according to the preceding results using some “boosting” method (such as AdaBoost or XGBoost, for instance).

The final result is either some weighted combination of all preceding results, or simply the outputs of the last step in the sequence (see *Boosting with AdaBoost and Gradient Boosting*, on the Data Action Lab blog).

The formal procedure is provided in Algorithm 6.

Algorithm 6: SequentialEnsemble

- 1 **Inputs:** dataset D , base algorithms A_1, \dots, A_r
 - 2 $j = 1$;
 - 3 **while** *stopping criteria are not met* **do**
 - 4 Select an algorithm A_j based on the results from the preceding steps;
 - 5 Create a new dataset D_j from D by modifying the weight of each observation based on the results from the preceding steps;
 - 6 Apply A_j to D_j ;
 - 7 $j = j + 1$;
 - 8 **end**
 - 9 **Output:** anomalous observations obtained by weighing the results of all previous steps
-

Independent Ensembles

In **independent ensembles**, different algorithms (or different instantiations of one algorithm) to the dataset (or some **resampled** dataset).

Choices made at the data and algorithm level are **independent** of preceding runs results (in comparison with sequential ensembles). The results are then combined to obtain more robust outliers

Every base anomaly detection algorithm provides an **anomaly score** (or an abnormal/regular classification) for each observation in D ; observations with higher scores are considered more anomalous than observations with lower scores.

The formal procedure is provided in Algorithm 7.

Algorithm 7: IndependantEnsemble

```
1 Inputs: dataset  $D$ , base algorithms  $A_1, \dots, A_r$ 
2  $j = 1$ ;
3 while stopping criteria are not met do
4   |   Select an algorithm  $A_j$ ;
5   |   Create a new dataset  $D_j$  from  $D$  by (potential)
   |   re-sampling, but independently of the
   |   preceding steps' results;
6   |   Apply  $A_j$  to  $D_j$ ;
7   |    $j = j + 1$ ;
8 end
9 Output: anomalous observations obtained by
   combining the results of all previous steps
```

Many combination techniques are used in practice:

- **majority vote,**
- **average,**
- **minimal rank, etc.**

Let $\alpha_i(\mathbf{p})$ and $r_i(\mathbf{p})$ represent the (normalized) **anomaly score** and the **anomaly rank** of $\mathbf{p} \in D$ according to algorithm A_i , respectively. The smaller the anomaly score, the smaller the anomaly rank, and *vice-versa*.

Anomaly scores lie between 0 (unlikely to be an anomaly) to 1 (likely to be an anomaly); ranks range from 1 to n (the number of observations, with ties allowed).

If the base detection algorithms are A_1, \dots, A_m , the average anomaly score and the minimal rank of an observation $\mathbf{p} \in D$ according to the independent ensemble method, say, are respectively

$$\alpha(\mathbf{p}) = \frac{1}{m} \sum_{i=1}^m \alpha_i(\mathbf{p}) \quad \text{and} \quad r(\mathbf{p}) = \min_{1 \leq i \leq m} \{r_i(\mathbf{p})\}.$$

If $n = m = 3$, for instance, we could end up with

$$\alpha_1(\mathbf{p}_1) = 1.0, \alpha_1(\mathbf{p}_2) = 0.9, \alpha_1(\mathbf{p}_3) = 0.0;$$

$$\alpha_2(\mathbf{p}_1) = 1.0, \alpha_2(\mathbf{p}_2) = 0.8, \alpha_2(\mathbf{p}_3) = 0.0;$$

$$\alpha_3(\mathbf{p}_1) = 0.1, \alpha_3(\mathbf{p}_2) = 1.0, \alpha_3(\mathbf{p}_3) = 0.0.$$

Using the mean as the combination techniques, we obtain

$$\alpha(\mathbf{p}_1) = 0.7, \alpha(\mathbf{p}_2) = 0.9, \alpha(\mathbf{p}_3) = 0.0, \implies \mathbf{p}_2 \succeq \mathbf{p}_1 \succeq \mathbf{p}_3 :$$

\mathbf{p}_2 is more anomalous than \mathbf{p}_1 , which is itself more anomalous than \mathbf{p}_3 .

Consequently,


$$r_1(\mathbf{p}_1) = 1, r_1(\mathbf{p}_2) = 2, r_1(\mathbf{p}_3) = 3;$$

$$r_2(\mathbf{p}_1) = 1, r_2(\mathbf{p}_2) = 2, r_2(\mathbf{p}_3) = 3;$$

$$r_3(\mathbf{p}_1) = 2, r_3(\mathbf{p}_2) = 1, r_3(\mathbf{p}_3) = 3,$$

and under the minimal rank method, we obtain

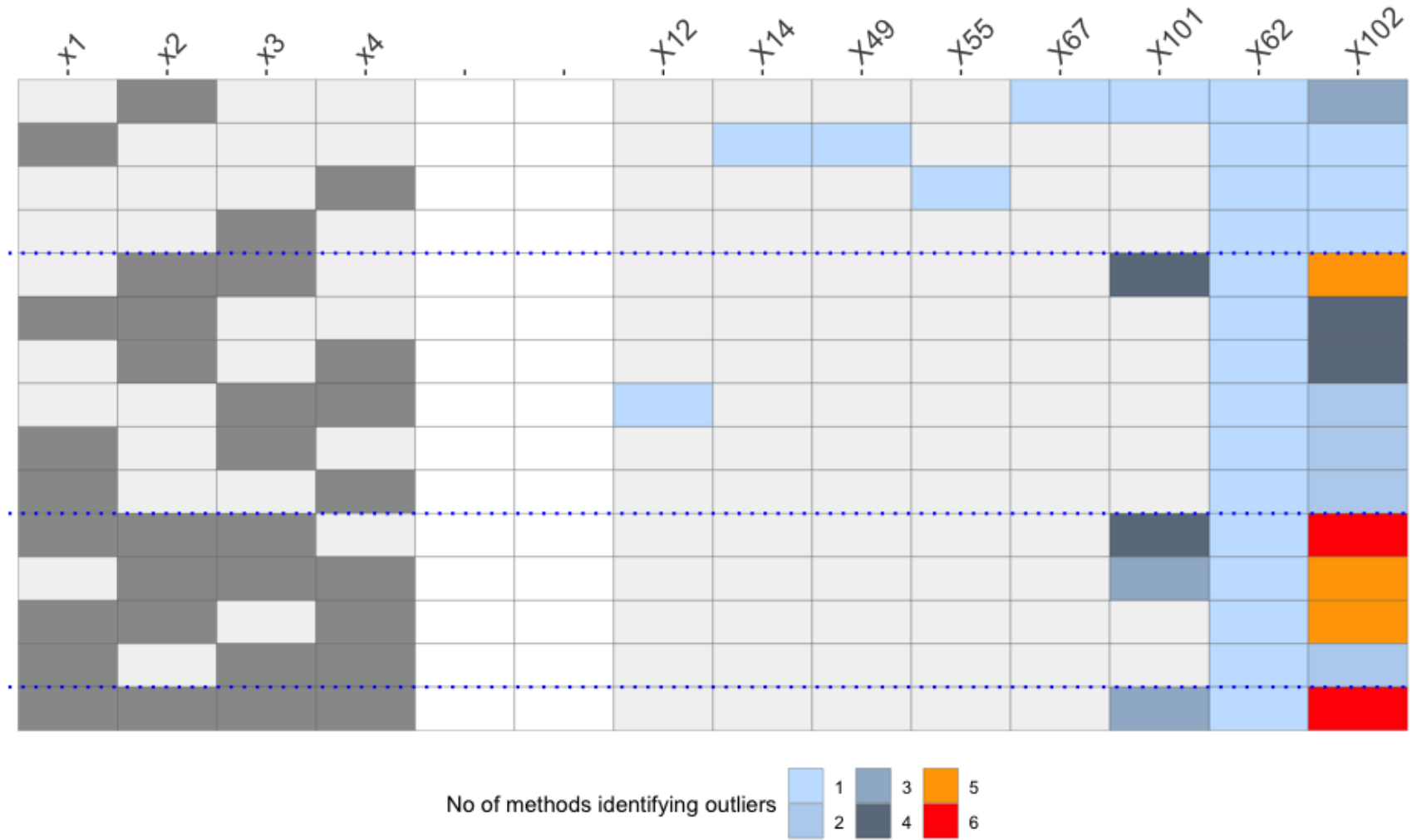
$$r(\mathbf{p}_1) = r(\mathbf{p}_2) = 1, r(\mathbf{p}_3) = 3, \implies \mathbf{p}_1 \succeq \mathbf{p}_3 \text{ and } \mathbf{p}_2 \succeq \mathbf{p}_3.$$

 In general, the results not only depend on the dataset under consideration and on the base algorithms that are used in the ensemble, **but also on how they are combined.**

For HDLSS data, ensemble methods can sometimes allow the analyst to mitigate some of the effects of the curse of dimensionality by selecting **fast** baseline algorithms (which can be run efficiently multiple times) and focusing on building robust relative anomaly scores through combination.

Another suggestion: use a **different sub-collection** of the original dataset's features at each step, in order to **de-correlate** the base detection models.

Even without combining the results, it may be useful to run multiple algorithms on different subspaces to produce an **Overview of Outliers (O3)**.



The columns on the left indicate the **subspace variables** (see row colouring).

The columns on the right indicate which **observations were identified as an outlier** by at least 1 method (HDoutliers, FastPCS, mvBACON, adjOutlyingness, DectectDeviatingCells, covMCD) in at least 1 subspace.

The **colours** depict the number of methods that identify each observation in each subspace as an outlier.

Observation 102 is identified as an outlier by 6 methods in 2 subspaces, 5 methods in 3 subspaces, 4 methods in 2 subspaces, 3 methods in 1 subspace, 2 methods in 4 subspaces, and 1 method in 3 subspaces – it is clearly the **most anomalous** observation in the dataset.

Observations 62 and 101 are also commonly identified as outliers.

5.5.2 – Anomalies in Text Datasets

In the **one-hot encoding** framework, text data is typically

- **sparse;**
- **high-dimensional,** and
- **non-negative.**

Word vector representations help with the first two of these, but at the cost of removing the third (and reduced interpretability).

The anomaly detection models that have been discussed previously can be extended to text data, but with subtle differences.

Text Processing

Proximity-based models require either a sound distance function or a sound similarity function.

The **term frequency-inverse document frequency** representation is commonly-used for text processing: the term frequency for each word (i.e. how often it shows up in a document) is **normalized** by the inverse document frequency (i.e. in how many documents it appears).

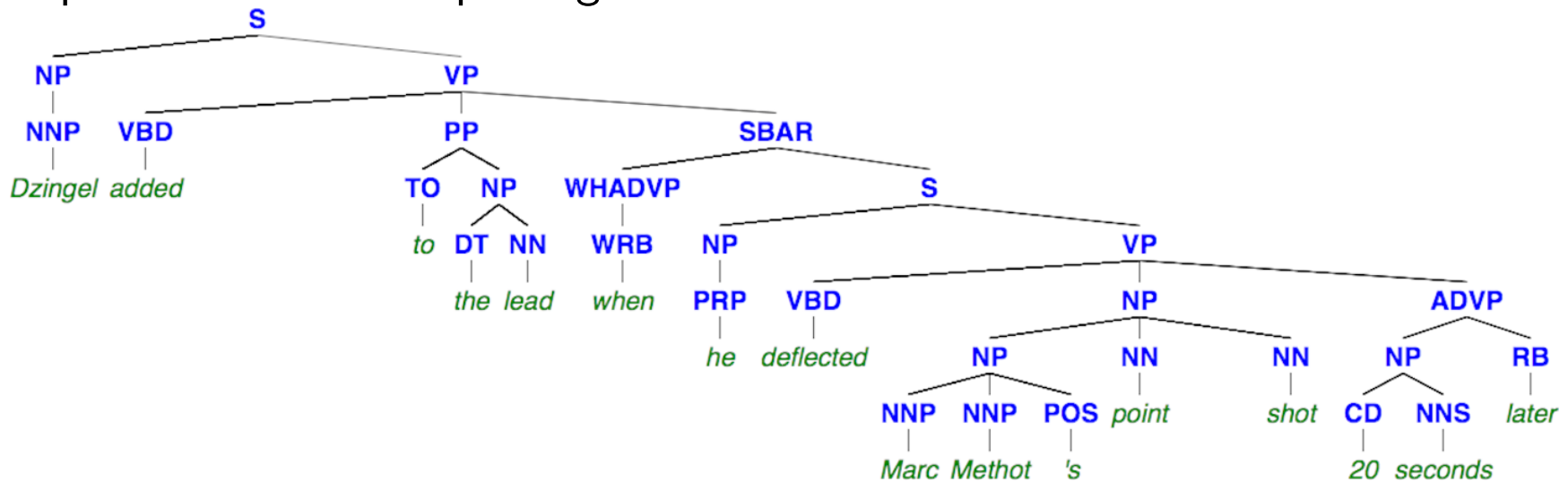
Semantic parsing is the process of converting a sentence in a natural language to a formal meaning representation.

The word **order** and its **type/role** provide the word's **attributes**.

Consider the sentence:

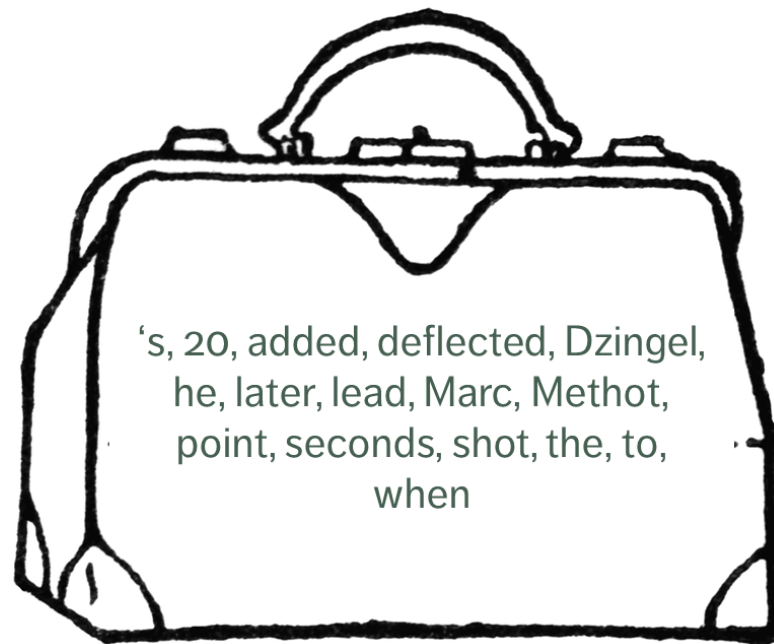
“Dzingel added to the lead when he deflected Marc Methot’s point shot 20 seconds later.” (AP game recap, Ottawa Senators vs. Toronto Maple Leafs, 18-2-2017)

A potential semantic parsing tree is shown below:




By contrast, in the **Bag-of-Words** (BoW) framework, only the **presence** (or **absence**) of “words” (stems, n -grams, sentences, etc.) is important.

Relative frequencies provide information (intent, theme, feeling, etc.) about the corpus; the **words themselves** are attributes of the document.



In general, text data requires **extensive cleaning and processing**. But there are a number of challenges due to the nature of the data:

- what is an anomaly in the text?
- what is an outlier?
- are these concepts even definable?
- how do we deal with encoding errors?

 Spelling mistakes and typographical errors are difficult to catch in large documents, even with spell-checkers; in the anomaly detection context, perhaps the text should not be cleaned up too thoroughly?

The process can be simplified to some extent with the help of **regular expressions** and **text pre-processing functions**.

Specific pre-processing steps vary depending on the problem:

- tweetish uses a different vocabulary than legalese
- same for a child who's learning to speak and a Ph.D. candidate defending her thesis

As is almost everything else related to text mining, the cleaning process is **strongly context-dependent**, and the order of pre-processing tasks can affect results.

Text Processing Options:

- convert all letters to **lower case** (avoid when seeking names)
- remove all **punctuation marks** (avoid if seeking emojis)
- remove all **numerals** (avoid when mining for quantities)
- remove all extraneous **white space**
- remove characters within **brackets** (avoid if seeking tags)
- replace **numerals with words**
- replace **abbreviations**

- replace **contractions** (avoid if seeking non-formal speech)
- replace all **symbols with words**
- remove **stop words** and **uninformative words** (language-, era- and context-dependent)
- **stem words** and **complete stems** to remove empty variations
 - “sleepiness”, “sleeping”, “sleeps”, “slept” \implies “sleep”
 - the stem “operati” has different meanings in “operations research”, “operating systems” and “operative dentistry”

Text Processing Challenges:

- **phonetic accent representation:** “ya new cah’s wicked pissa!”
- **neologisms and portmanteaus:** “I’m planning prevenge”
- **poor translations/foreign words, puns and play-on-words**
- **mark-up, tags, and uninformative text:** ; \verb; ISBN blurb
- **specialized vocabulary:** “clopen”; “poset”; “retro-encabulator”
- **fictional names and places:** “Alderaan”; “Kilgore Trout”
- **slang and curses:** “skengfire”; #\$&#!

Text must be stored to data structures with right properties:

- a **string** or **vector of characters**, with language-specific encoding;
- a **corpus** (collection) of text documents (with meta information)
- a **document-term matrix** (DTM) where the rows are documents, the columns are terms, and the entries are an appropriate text statistic (or the transposed term-document matrix (TDM))
- a **tidy text dataset** with one token (single word, n -gram, sentence, paragraph) per row

No magic recipe: best format depends on the problem at hand. But this step is **crucial**, both for semantic analysis and BoW.

	Document 1	Document 2	Document 3	...	Document N	
Token 1	0	0	1	62	3	66
Token 2	0	1	0	61	2	64
Token 3	1	0	3	101	0	105
...	112	24	38	84	0	258
Token M	2	2	0	12	3	19
Sum	115	27	42	320	8	

Document-Term Matrix

Consider a corpus $\mathcal{C} = \{d_1, \dots, d_N\}$ consisting of N **documents** and M **BoW terms** $\mathcal{T} = \{t_1, \dots, t_M\}$. Each document d contains M_d terms.

For instance, if

$$\mathcal{C} = \{ \text{“the dogs who have been left out”}, \text{“who did that”}, \\ \text{“my dogs breath smells like dog food”} \},$$

then $N = 3$, $M = 14$, and

$$\mathcal{T} = \{ \text{“been”}, \text{“breath”}, \text{“did”}, \text{“dogs”}, \text{“food”}, \text{“have”}, \text{“let”}, \\ \text{“like”}, \text{“my”}, \text{“out”}, \text{“smells”}, \text{“that”}, \text{“the”}, \text{“who”} \}.$$

The **relative term frequency** of term t in document d is

$$tf_{t,d}^* = \frac{\# \text{ of times term } t \text{ occurs in document } d}{M_d}.$$

$tf_{t,d}^*$		t													
		1 been	2 breath	3 did	4 dogs	5 food	6 have	7 let	8 like	9 my	10 out	11 smells	12 that	13 the	14 who
d	1	1/7	0	0	1/7	0	1/7	1/7	0	0	1/7	0	0	1/7	1/7
	2	0	0	1/3	0	0	0	0	0	0	0	0	1/3	0	1/3
	3	0	1/7	0	2/7	1/7	0	0	1/7	1/7	0	1/7	0	0	0

For instance, $1/3$ of the terms in document d_2 are “did;” $2/7$ of the terms in document d_3 are “dogs”.

The **relative document frequency** of t is

$$df_t^* = \frac{\# \text{ of documents in which term } t \text{ occurs}}{N} = \frac{\sum_d \text{sign}(tf_{t,d}^*)}{N}.$$

In this example, all the terms occur in exactly one of the documents (not all the same), except for “dogs” and “who”, which appear in two documents.

	t													
df_t^*	1 been	2 breath	3 did	4 dogs	5 food	6 have	7 let	8 like	9 my	10 out	11 smells	12 that	13 the	14 who
	1/3	1/3	1/3	2/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	1/3	2/3

None of the $df_{t,d}^*$ entries can be 0 (otherwise the corresponding document would be empty).

The **term frequency-inverse document frequency** of t in d is

$$tf-idf_t^* = -tf_{t,d}^* \times \ln(df_t^*).$$

$tf-idf_t^*$		t													
		1 been	2 breath	3 did	4 dogs	5 food	6 have	7 let	8 like	9 my	10 out	11 smells	12 that	13 the	14 who
d	1	0.16	0	0	0.06	0	0.16	0.16	0	0	0.16	0	0	0.16	0.06
	2	0	0	0.37	0	0	0	0	0	0	0	0	0.37	0	0.14
	3	0	0.16	0	0.12	0.16	0	0	0.16	0.16	0	0.16	0	0	0

Note that the entries for which $tf-idf_{t,d}^* = 0$ are also those for which the relative term frequency is $tf_{t,d}^* = 0$.

If all the documents contain the term t , then $df_t^* = 1$ and

$$tf-idf_{t,d}^* = -tf_{t,d}^* \times \ln(1) = 0,$$

and that specific term does not provide information.

If a term t rarely occurs in a document d , then $tf_{t,d}^* \approx 0$ and


$$tf-idf_{t,d}^* = 0 \times \ln(df_t^*) \approx 0.$$

Terms that appear relatively often only in a small subset of documents are crucial to understanding those documents **in the general context** of the corpus.

At the analysis stage, it is easy to forget where the data comes from and what it really applies to.

Text comes unstructured and unorganized. After processing, text is clean, but still unstructured. Bag of Words provides a framework for a structured numerical representation of text.

How does this affect the choice of text statistic in the DTM/TDM?

 There is no sound mathematical justification to use the term frequency-inverse document frequency statistic. It may may not always be the ideal choice, but it is often used nonetheless.

Word vector representations change the nature of DTM.

Anomaly Detection Models

However the documents have been **normalized**, the corpus is now represented by a matrix

$$\mathbf{X} = \begin{bmatrix} \mathbf{D}_1 \\ \vdots \\ \mathbf{D}_N \end{bmatrix}.$$

For proximity-based methods, the similarity between documents is computed using the **cosine similarity**:

$$w(\mathbf{D}_i, \mathbf{D}_j) = \cos(\mathbf{D}_i, \mathbf{D}_j) = \frac{\mathbf{D}_i \cdot \mathbf{D}_j}{\|\mathbf{D}_i\| \|\mathbf{D}_j\|}.$$

Proximity-based algorithms can then be applied to this dataset, as they would to any numerical dataset. So can supervised algorithms, if data labels are available. The results, however, are not always ... satisfactory.