Data Preparation Techniques - Data Cleaning

COMP 3400/6981

MEMORIAL UNIVERSITY

A working definition of perfect data

Perfect (tabular) data corresponds to a rectangular array or grid of values (readings, observations, etc.), where

1. each row describes a **unique** instance (sequence of values), and
2. each column represents a **single** variable, and
3. each value should be **complete** (meaning values are recorded for all variables), **valid** (satisfies certain assumptions from domain knowledge), and **correct** (meaning the value is an accurate snapshot of what the data is supposed to represent).

Data which is not perfect is **imperfect**. Imperfectness in data can **effect** virtually all the data-oriented problems. That is either

1. The results may differ in the presence of imperfect data.
2. The problem cannot be solved in the presence of imperfect data.

Identifying data cleaning topics

Data Cleaning is the process of **reducing the imperfections** of imperfect data.

Note. A flawless perfection of imperfect data may not be possible or even desirable in each and every case.

Based on the definition of imperfect data we presented, one can identify the following topics in Data Cleaning:

1. Duplicate instances
2. Compound variables
3. Missing data
4. Data validation
5. Data correctness

Duplicate instances

Duplicates alter the distribution of variables. Yet duplicates are not necessarily harmful. The impact of duplicate instances depends on three factors:

1. Which records are duplicated
2. How frequently they are duplicated
3. The task at hand

Duplicate instances in a dataset, can be **identified**, and **removed**.

Duplicate instances

Note 1. Sometimes the duplicity is subtle. For example:

- If the information comes from different sources, the systems of measurement may be different as well, resulting in some instances being actually the same, but not identified like that. Their values can be represented using the metric system and the imperial system in different sources, resulting in a not-so-obvious duplication.
- Depending on the application, over-the-extreme or below-the-extreme records might be regarded as the same.

Note 2. In some datasets there might be an identifier variable. It is possible that duplicates with the different identifiers may emerge. The process of duplicate identification in such cases is less straightforward.

Note 3. The definition of duplicity can be extended. For example in some applications, practitioners regard instances in a close proximity as duplicates. In which case, a proper (in accordance with the measurement level) distance must be defined between the data instances.

Compound variables

A **compound variable** is a variable which consists of two or more variables. A compound variable either

1. Presents an incohesive attribute. For example, due to some deficiency in the data compilation process, two variables might emerge as one through some erroneous string concatenation.
2. Presents a cohesive attribute. For example *date* is a cohesive attribute. Yet, *year*, *month*, and *day* are variables themselves too.

Not only is decoupling of variables helpful in the 1st case, but also in the 2nd case (where although the data might be regarded as clean, one may make it cleaner depending on the application).

Compound variables

The practitioner can consult the following sources to identify **in-cohesive compound variables:** 1- Observation, 2- Domain knowledge, 3- Metadata, and 4- Experts

The following means help the practitioner consider splitting of **cohesive compound variables:**

1. The nature of the task hand

   Example: If the data is to be analyzed for monthly patterns, you need to extract month from the data.

2. Feedback from doing the task

   Example: If the predictive model has not performed well with the given compound variable you may test different split variables.

Important. Usually, practitioners transform the compound variable to `string` types. The reasons is that the rich string manipulation capabilities (such as regular expressions), tremendously help with the decoupling process.

Missing data
Representation

We firs study the **characteristics** of missingness and then its **treatment**.

There is no **universal representation** of missing data. Causes:

1. Different default representations adopted by different software environments.
2. Avoiding logical complications:
   Let's assume we want to extract all patient records from a dataset where body mass index (BMI) is greater than 35. Now what do we do about those patients whose BMI value is missing? Here a number (0 perhaps) for missingness would not break the operation.
3. The unfortunate practice of using numerical codes for missing data.
4. Representations for missing text data is almost unlimited.
   Missing values can be represented by blanks (one or more), empty character strings (distinct from blanks), symbols like "?" or "???", words like "UNKNOWN" (in uppercase, lowercase, or mixed case), or abbreviations like "UNK"
5. Distinguishing between different classes of missing data.
   For example "don't know" or "refuse to answer" in a survey.

Missing data
Representation

Variety of representations poses significant **complications:**

1. Since the default missing data representation in one **environment** may not be **translated** correctly into the corresponding default representation in another.
2. Some software environments support **multiple representations** for missing data.
3. It is possible that the missing data codes used by those who collect and aggregate the data are not recognized as such by those who analyze the data, leading to the problem of **disguised missing data**.

Therefore, for the treatment of the missing data, the **practitioner should:**

- Identify the representation(s) of missing data in a **dataset**.
- Know the representation(s) of missing data in the **target environment**.
- Make sure of the integrity of the **treatment process**.
  For example does a piece of code take to account all the present representation of missing values?

Missing data
Detecting missing data and its severity and patterns

Practitioner can become aware of missing data, either through **textual** queries (mostly numbers), or **visualization**.

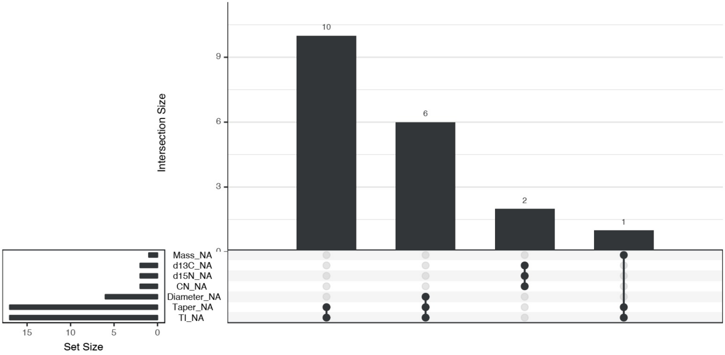
Textual, for example:

1. The number of missing values.
2. The number of data instances with one or more missing values.
3. The number of missing values for each variable.
4. Row numbers, IDs, or index of the of data instances with missing values (for at least one or all, or a particular or a combination of variables).

Detection of **patterns** in missing data can be done through **visualization of missingness.**

Co-occurrence plot

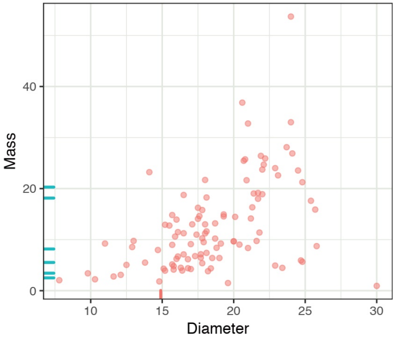
1. Mainly consist of **histograms** for illustrating the **distribution** of missing values in variables.
2. Displays the **frequencies** of the most common variable **combinations**.



Missing data
Detecting missing data and its severity and patterns

Annotated scatter plot

1. For exploring the variable **relationships** as to the missing values.
2. The **ticks** on both axes show **where** the missing values belong.
3. Annotated scatter plot can have different usages. For example identifying the **regions** of imperfection in the data.



A missing value can be either **legitimate** or **illegitimate**.

Examples of **legitimate** missing data:

- If you are allowed to leave a field unanswered in a survey.
- In a dataset of employees with annual salaries as variables, there can be missing values for retirees.
- There might be missing values in a dataset for privacy reasons.

Examples of **illegitimate** missing data:

- Missing values caused by a sensor's failure/miscalibration.
- Skipped required fields of a survey.
- As a side-effect of some data preparation operations.

Missing data
Mechanisms of missingness

The **mechanism** behind missingness can **inform the treatment process.** There exist three mechanisms behind a missing value:

1. **Structural deficiencies** in the data
   - A **missingness** which is **defined** as a value of an attribute.
   - **Example:** The value of the Alley attribute in a housing dataset is "gravel" or "paved", or **missing**.
2. **Random occurrences** falling under any of the following two classes:
   - **Missing completely at random** (MCAR): The **likelihood** of a missing result is **equal** for all data points (observed or unobserved).
   - **Missing at random** (MAR): The **likelihood** of a missing results is **not equal** for all data points.
   - It can be **difficult** or impossible to **distinguish** MCAR from MAR.

There exist three **mechanisms** behind a missing value:

3. **Specific causes**
   - Also known as **not missing at random** (NMAR)
   - **Example:** Consider a clinical study where patients are measured over time. A patient may drop out of a study due to an adverse side effect. For this patient, no measurements will be recorded after the time of drop-out.
   - **Pre-tailored** treatment approaches do not usually work for data under **MNAR**.
   - The practitioner is usually required to **model** the missingness **explicitly**, and devise a **custom** treatment approach.

Missing data
Considerations prior to treatment of missing values

Important:

1. The **target problem** is key to choosing the **right treatment.**
2. The **severity** of missing data is also important.
3. **Legitimacy** or lack-there-of can inform the treatment of missing data.
4. The **mechanism** behind missing data can also inform the treatment.

Ignoring the missing values as a technique, might be *harmful*, *conceivable*, *advisable*, or even *necessary*:

- **Necessary:** Descriptive analysis of data with legitimate missing values. Or pattern mining on data with legitimate missing values (missingness here is meaningful).
- **Advisable:** Pattern mining on data with small number of legitimate missing values.
- **Conceivable:** Predictive modeling with illegitimate missing values (target model must be able to handle missing values).
- **Harmful:** Including a variable with severe missingness (large number of missing values) in a predictive model.

Missing data
Treatment: Deletion of missing values

Deletion: Discarding the values where key variables are missing .

- **Listwise deletion** (complete-case analysis): Deleting data instances with one or more missing values.
- **Variable deletion:** In cases where missingness is severe solely for a set of variables, one can also remove the entire set of variables to maintain most of the data instances.

Rule of thumb: In practice, more often than not, data instances are more critical than variables and a higher priority should be placed on keeping as many as possible.

Missing data
Treatment: Deletion of missing values

Advantages:

- Simple evaluation of the results
- Computational lightness

Disadvantages:

- Deleted legitimate missing values might entail meaningful information.
- When data are MCAR, missingness ia not biased; however this is not the case with the more common MAR or MNAR missing data. Where deletion of only some instances might be justified, not all.
- In cases where the number of instances is small, deletion is harmful.
- In case of severe missingness, deletion might result in poor results due to substantial decrease of data instances (more substantially in case of MAR and MNAR where complete instances may not be able to represent the sample at all).

Imputation uses the relationships among the non-missing values and/or other inputs, to provide an estimate to fill in the missing values.

Usage: Imputation in general, is usually used with illegitimate missing data. It can be used under MCAR, MAR, and rarely NMAR. However, the degree of applicability to each of the mechanisms varies across the imputation techniques (case by case investigation).

There exist two **classes** of imputation:

Parameterized. Meaning the imputation technique tries to guess the underlying distribution of a variable and impute accordingly. Examples:

- Mean substitution (univariate: based on a single variable)
- Linear imputations (multivariate: based on multiple variables)
- Maximum likelihood (multivariate)
- Expectation-Maximization or EM (multivariate)
- Multiple Imputation (multivariate)

Non-parameterized. Meaning the imputation technique impute the missing values without any explicit assumption of the underlying distributions. Examples:

- Non-parametric Multiple Imputation (multivariate)
- Various techniques based on non-parametric Machine Learning (multivariate)

Missing data
Treatment: Imputation - Mean substitution

Mean substitution is a univariate missing data imputation strategy that replaces all missing values with the mean of the observed data values for a variable.

One can **extend** this idea to other measures of central tendencies, such as **median**, which can be applied to ordinal variables and is also less sensitive to outliers. Or **mode** for both nominal and ordinal variables.

Advantages:

1. Computationally light
2. Can be an acceptable approach when there exist only a few missing values

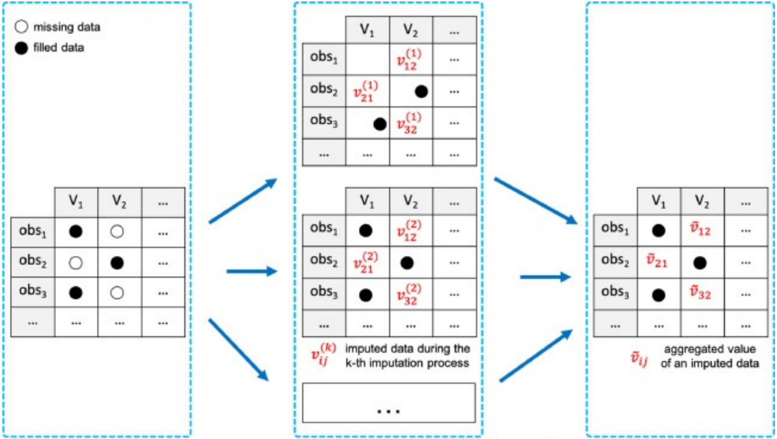
Disadvantages:

1. Is inherently biased (oversimplification of the underlying distribution).
   Bias: The difference between an estimate and the true value
2. Can perform even worse in the presence of high variance.
3. Imputes with inaccurate mean when data is not MCAR. Usually even worse in case of MNAR.

Multiple Imputation or **MI**, is one of the modern imputation techniques:

1. **Main idea:** Minimize the bias.
2. It generates **several** imputed datasets based on the input dataset with missing values.
3. This repeated imputation can be done thanks to the use of **Markov Chains** and **Monte Carlo** methods.
4. Therefore, the imputations are not solely based on the data, but also an underlying **random process**.
5. MI then **combines** the multiple imputed versions of the given dataset, into the **final result.**

Missing data
Treatment: Imputation - Multiple Imputation



Multiple Imputation

Advantages of MI:

1. Produces results which are **superior** to most of the other techniques under **both MCAR** and under **MAR**.
2. Some scholars have reported that MI in some cases have produced acceptable results under **MNAR**.
3. MI can produce **confidence intervals** for the imputed dataset, hence providing the practitioner with an **estimate of the bias** of the imputed datasets.

Disadvanteges of MI:

1. Computationally **heavy**.
2. Choosing the target **number** of imputed datasets can be challenging.
3. **Combining** the results can be challenging.

Machine Learning Imputation or **MLI**:

1. **The main idea:** Machine Learning is mainly used for predication of a target variable. The same process can be used to **predict the missing values**.
2. Unlike parametric imputations, (many) MLI techniques **do not require** or search for an underlying **distribution**.
3. **Almost any** machine learning method, is also potentially an **imputer**.
4. As an example we take a look at an MLI technique based on kNN (K-Nearest Neighbors), better known as ***k*NNI** (kNN Imputer).

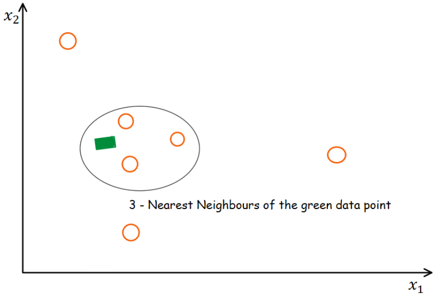
The idea behind ***k*NNI**:

1. When imputing a value for the variable V, kNNI finds the ***k* most similar data instances**, to the instance which is hosting the missing value. All the ***k* instances** must be **complete**.
2. Then, kNNI imputes the missing value in V, with the **central tendency** of the values of V over the k most similar data instances.
3. Since kNNI works based on similarity, it requires a **measure of distance** between the data instances.

Assume:

1. A dataset which consists of three variables x_1, x_2, and x_3.
2. A data instance, which we refer to as the **green** instance misses the value of x_3.
3. The **red** instances are **complete.**
4. The **measure of similarity** is Eculidean distance.

3-NNI ($k = 3$), finds the three nearest (with reference to the values of x_1 and x_2) red instances to the green instance through 6 **pairwise computations** of distances between the green instance and each of the red instances. Then, the algorithm takes the average value of the three nearest neighbors for x_3, and impute the green instance with the average value.



Advantages of kNNI:

1. Computationally light for smaller datasets with a small or moderate level of missingness, and smaller k's.
2. Can be used both under MCAR and MAR (in case of MAR you should expect poorer imputation).

Disadvantages of kNNI:

1. Can be computationally heavy (depending on the size of the dataset, missingness, and the value of k).
2. It relies too much on the complete instances.
3. Choosing K and the distance measure can be challenging.
4. Is sensitive to outliers (specifically when k is small).

Data validation

Reminder: Data validation is the process of ensuring that the data satisfies certain assumptions from the domain knowledge (including general knowledge).

The main **tool of data validation** is validation rules: a set of short statements rooted in domain/general knowledge that express the assumptions about the variables.

Data validation

Some examples of **validation rules:** for further **inspiration:**

- Yield per area (for a certain crop) must be between 40 and 60 metric tons/ha.
- The variable "type of ownership" (for buildings) may not be empty.
- The submitted "regional code" must occur in the official code list.
- The sum of reported profits and costs must add up to the total revenue.
- The persons in a married couple must have the same year of marriage.
- If a person is a child of a reference person, then the code of the person's father must be the reference person's code.
- The number of employees must be equal to or greater than zero.
- Date of birth must be larger than December 30, 2012 (for a farm animal).
- Married persons must be at least 18 years old.
- If the number of employees is positive, the amount of salary paid must be positive.
- The current average price divided by last period's average price must lie between 0.9 and 1.1.

Data validation

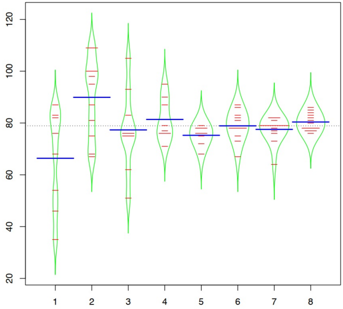
Therefore:

1. The examples include rules where values are compared with constants, past values, values of other variables, (complex) aggregates, and even values coming from different domains or datasets.
2. The set of knowledge-based validation rules for a dataset can thus be varied and, depending on the number of variables and known relationships between them, may be large.
3. Since any single variable may occur in several rules, validation rules are often interconnected and may therefore give rise to redundancies or contradictions.
4. A systematic way of defining rules, confronting data with them, and maintaining and analyzing rule sets is desirable.

Data correctness

Reminder: Data correctness means that a value is an accurate characterization of what the data is supposed to represent.

The following graph shows the distributions of the same Gaussian variable (whose values we know apriori) across different samples of the same population:



Problem. In data-oriented problems, we usually do not know what the data is supposed to represent independent of the data.

Data correctness

Correctness of data: Correct values of data are reasonably possible with reference to the character of the data.

The **character of data** is expressed by the **recorded instances** (in the absence of prior knowledge of the distribution underlying the population).

Therefore, a data instance which does not conform or comply with the rest of the instances is called a **nonconforming instance** or an **extreme instance** or an **anomaly**, in the sense that the instance does not conform to the character of the data.

Data correctness

Depending on the application a **nonconforming instance** may be deemed as an **incorrect instance**. Therefore, finding incorrect data in the sense we explained earlier, is mainly done through **identifying** nonconforming instances.

The process of identifying nonconforming instances in the data is usually called **outlier detection**.

We call an **outlier** in a given application **noise** when the outlier is **weak** or **less extreme**.

In some other texts-books where it is supposed that outliers are harmful, the term **noise** and terms such **noise removal** is regularly used instead of the term outlier and outlier detection.

Data correctness

Examples of cases where outliers might be **harmful** due to deforming the underlying distribution of a variable (application in the parenthesis):

- They generally increase error variance in data analysis.
  Since we have a set of observations, we have a set of errors and therefore we can compute its variance.
- Reduce the power of statistical (inferential) tests in data analysis.
  Statistical tests evaluate a hypothesis on the whole population based on a sample data.
- They can reduce the generalization power of a model in predictive modeling.
- It may introduce unrealistic patterns in the data in pattern mining.

Data correctness

Examples of cases where outliers might be **useful**:

- Fraud detection in banking sector
- Finding cure for diseases
  Example: Researchers in Africa discovered that some women were living with HIV for many years longer than expected despite being untreated (Rowland-Jones et al., 1995).
- In case where data is scarce and valuable
- Activity monitoring
- Quality control
- Intrusion detection systems: detecting unusual and malicious activities in computer systems or network systems, based on collected data such as operating system calls and network traffic.
- Anomaly detection in urban traffic flow: identifying unexpected and deviant flow values that could be caused by traffic congestions, traffic accidents, and so on.

Data correctness

Some possible **sources** of outliers:

1. Collection errors or misreporting (human or device)
2. External (to the collection process) factors (such as internet bots show up in the data when the data is supposed to reflect humans)
3. Improper sampling of a population (for example in election polls)
4. Data transformation
5. Unexplained/unknown/unnoticed phenomena, patterns, etc.

Outliers can be studied from **several standpoints**.

Based on the **number of data instances** involved to comprise a deviant pattern:

- There are (1) **point outliers**, (2) **collective outliers**.
- A **point outlier** is an **individual data instance** that deviates largely from the rest of the dataset. This is the simplest type of outlier to identify and is the major focus of the research on outlier detection
- **Collective outliers** are a **collection of data instances** that appear anomalous with respect to the rest of the entire dataset. However, each instance within the collection may not constitute an outlier individually. An example of collective outliers is a specific sequence of considerable withdrawal of a bank account. Collective outliers are common in sequential data form (such as time series data).

Based on the **context**,

1. An outlier can be **contextual** or not: A data point is considered a contextual outlier if its value significantly deviates from the rest the data points in the **same context.** Contextual outliers are common in sequential data form (such as time series data).
2. **Example.** A sudden surge in order volume at an ecommerce company, as seen in that company's hourly total orders for example, could be a contextual outlier if this high volume occurs **outside** of a known promotional discount or high volume period like Black Friday.

Data correctness
The categorization of outliers

Based on the **scope of comparison:**

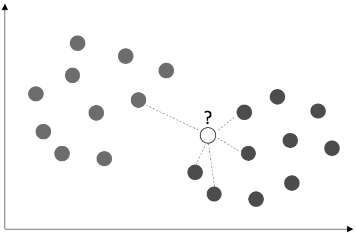
- Point outliers can further be classified into (1) **local** outliers and (2) **global** outliers.
- The detection of **local** outliers relies on the characteristic differences (e.g., the difference in neighborhood density) between the outlier and its **nearest neighbors.**
- **Global** outliers address the difference with the **entire dataset.**

Based on the number of **variables** under study:

- **Univariate** (for example to detect noise in one single vairable)
- **Multivariate**

Nearest neighbor:

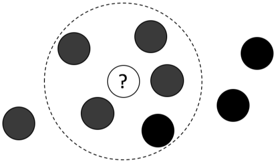
1. Nearest-neighbor-based outlier detection approaches robustly measure the **degree** of **extremeness** of an outlier (in outlier detection literature, also known as measuring the **granularity**), on the basis of a data point's **distance** to its **nearest neighbors.**
2. The underlying assumption is that **normal data instances** are **closer to their neighbors**, thus forming a dense neighborhood, whereas **outliers** are **far** from their neighbors.
3. There are two main ways to define the neighborhood:
   i. ***k* nearest neighbors** (kNN): An instance is labeled as outlier based on a function of distance between the k nearest neighbors to the instance. Depending on the definition of the function, the result might be different.
   ii. **The neighborhood within a pre-specified radius:** An instance is considered an outlier if its neighborhood does not have enough other points.



***k*NN:**

1. This is an example of $5NN$.
2. The data instance will be labeled as an **outlier**, if a **function** of the distance measurements to its closest neighbor violates a **threshold**.
3. An example for such a function: if the **average/sum** of the distance measurements **surpasses a fixed value.**

Data correctness
Approaches for outlier detection: pre-specified radius

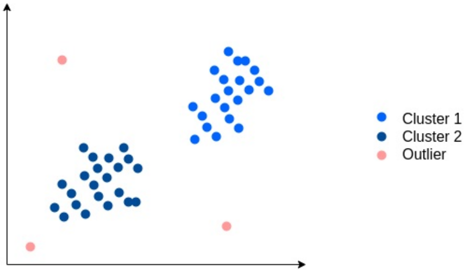


The neighborhood within a pre-specified radius:

1. A **radius** has be specified.
2. If the **number** of the data instances within the radius falls **below** a **specified number**, the instance indicated by a question mark will be labeled as an **outlier**.

Data correctness
Approaches for outlier detection: Clustering-based

Clustering-based outlier detection methods assume that the **normal data** objects belong to **large** and **dense clusters**, whereas **outliers** belong to either **small** or **sparse clusters**, or **do not belong** to any clusters.



Notes:

1. In this example, the outliers **do not belong** to any clusters.
2. Note that **sparse** (spaced out) and **small** (consisting of few data instances) clusters can be identified as **collective** outliers.
3. Note the **global scope** of this method (no use of neighborhoods).

Data correctness
Handling of harmful outliers

There exist **different approaches** of dealing with **harmful outliers** (afer their identification):

Outlier **removal**:

1. You simply remove a **data instance** which is detected as an outlier.
2. Outlier removal **can** improve the result of data-oriented solutions. **For example**, removing noise can enhance the generalization power of a predictive model.
3. But it can also reduce the sample size and introduce **bias** or **information loss** (specifically when the data is scarce and valuable). This could ultimately **skew the results** of data analyses and **damage model performance** for example.

Outlier resistant/robust data-oriented solutions:

1. In some data-oriented problems, there exist **out-of-the-box** solutions which are r**obust in the presence of outliers.**
2. Example 1: In classification **Random Forest** is not overly sensitive to outliers, while **Linear Discriminant Analysis** is.
3. Example 2: In clustering, **Hierarchical** schemes are more robust in the the presence of outliers than algorithms such as **K-Means**.
4. Example 3: In data analysis, you may use **median** (instead mean) which is less sensitive to outliers.

Data correctness
Handling of harmful outliers

Correction of an outlier:

1. Depending on the distribution of a variable, certain variable **transformations** can fold in the outliers as normal instances.
   i. For example **logarithmic transformation** can **de-emphasize** outliers by **compressing** the data's range and bringing extreme values closer to the mean.
   ii. Success is **not guaranteed however**.
   iii. Furthermore, a transformed variable is **harder to interpret** with its new values.
2. Using expert knowledge to **modify the unrealistic values** (direct manipulation) of variables in an outlier instance or a group of them. While useful in some applications, this approach may introduce **bias** in the data.