EFive: An Ontology Design Pattern for Representing Statistical Variation

David K. Kedrowski[1], Torsten Hahmann[1]

[1]School of Computing and Information Science, University of Maine, Orono, ME 04469, USA

Abstract

Our world is replete with variation; however, formal logic is not well-suited to representing and reasoning about statistical summaries and variation. The EFive ontology design pattern (ODP) addresses this gap by providing a formal structure for representing five-number summaries from exploratory data analysis, along with two key measures of variation: the interquartile range (IQR) and the values beyond which outliers lie. To enable querying and reasoning over these summaries, EFive is encoded entirely in OWL2.

The EFive ODP is built around the concept of measurement scales (nominal, ordinal, interval, and ratio) and extensively reuses the STAD ODP. The result is a flexible framework that supports multiple goals: representing statistical summaries—including variation—within datasets and their partitions; maximizing reuse of existing ontologies and ODPs; offering interoperability with existing OWL2 domain ontologies; and supporting the detailed specification of provenance.

Keywords

ontology, ontology design pattern, knowledge graph, statistical summarization, variation, five-number summary

1. Introduction

Most everything in the world around us exhibits some level of *variability*. That is, things are rarely identically the same - they *vary*, often across space and/or time. *Variation* refers to, for example, the differences we see in the definitions of concepts, among the members of classes, or when measuring attributes that exhibit variability. This distinction between *variability* and *variation* is from Reading and Shaugnessy [1]. *Probability* can be used to describe events that exhibit variability, like in games of chance where there is variation in the results.

Much of statistics involves working with variation – according to [2], it "is at the heart of statistics." When we have a dataset, it is helpful to know how much variation is present (think of the example in nearly every introductory statistics textbook where a measure of variation is used to show a distinction between two datasets with identical measures of central tendency). Variation makes modeling hard because data rarely (if ever) matches any function exactly [3].

Variation can be leveraged to do much more. Hahmann and McIlraith [4] provide six areas where knowledge of variation can be useful: object classification tasks, purely logical reasoning, statistical querying, object retrieval tasks, inductive reasoning, and clustering. These tasks are aided by knowledge of how things within a class or between classes are similar (low variation) and how they differ (high variation). Many of these tasks are related to work in machine learning (ML) and artificial intelligence (AI) where variation plays a crucial role in, e.g., evaluation, classification, and clustering.

Overall Objective Ontological languages based on first-order logic (FOL) or description logic (DL) subsets, which form the basis of OWL2, are not well-suited for representing variation. Our goal is to develop EFive as an ontology design pattern (ODP) [5] that (1) can succinctly represent variation in the spirit of five-number summaries, (2) can be implemented in OWL2 to represent variation, and (3) can facilitate basic reasoning about/with variation. The EFive ODP is not intended as a domain ontology

Proceedings of FOIS 2025 Satellite events co-located with the 15th International Conference on Formal Ontology in Information Systems (FOIS 2025), September 10-12, 2025, Catania, Italy

david.kedrowski@maine.edu (D. K. Kedrowski); torsten.hahmann@maine.edu (T. Hahmann)

0000-0002-9070-3169 (D. K. Kedrowski); 0000-0002-5331-5052 (T. Hahmann)

about statistics; rather, it is meant to be used with existing ontologies in any domain that considers variation to facilitate the explicit representation of variation therein. However, this leaves open an important question: what do we mean by *variation*?

To answer that question, we first note that nearly all of the efforts to explicitly capture uncertainty—a concept that is distinct from but closely related to variability—in ontological representations take a Bayesian probabilistic approach. A hallmark of this approach is the focus on subjective probability and "degrees of belief." Work with uncertainty typically attempts to quantify the likelihood a given statement is true (see Section 2.1), while variation is about how the values in a dataset are dispersed. Therefore, for specifically representing variation we choose to take a frequentist, or empirical, approach. Empiricism, because it is rooted in data, is the natural approach for this work but presumes that the developed ODP will be used in practice with a domain ontology that encompasses both a TBox (terminology and axioms) and an ABox (i.e. data) in order to represent variation in the data and reason with it.

2. Related Work

Two distinct bodies of knowledge inform our work on the EFive ODP: one focusing on unifying logic and probability with the goal of representing and reasoning about uncertainty in knowledge more generally and the second aiming to capture statistical knowledge in some fashion within OWL2 ontologies and knowledge graphs (KGs) that incorporate them. We also briefly consider datacubes.

2.1. Unifying Logical and Statistical Representations of Knowledge

Efforts to represent and reason with variation appear to be absent from the literature on logic-based representations of knowledge. The closest efforts, seeking to unify first-order logic or description logics with probability-based representations of uncertainty,[1] date back to at least the 1950s. The emphasis for much of the work in this area is on representing and reasoning about uncertainty using subjective probability. A common example in the literature involves the statement "This bird can fly." While a statement like "All birds can fly" is logically false, "This bird can fly" will be false only some of the time and is dependent on the specific bird in question. A (subjective) probability can be applied to this statement, essentially assigning it a truth value somewhere in the interval $[0, 1]$.

There are many extensions of Nilsson's [7] influential framework for embedding probability within logic, including work with nonmeasurable probability events [8], reference classes [9, 10], and Bayesian [11] and multi-entity Bayesian networks [12]. Other work has focused on probabilistic extensions of DLs [13, 14, 15]. We found only a single work [16] that has considered empirical probabilities instead of subjective probabilities; however, we note that its focus was still on reasoning with probabilities and did not extend to other statistical concepts such as variation.

There have been a variety of efforts to extend OWL and OWL2 to work with probability [17, 18, 19, 20, 21] – again for the purpose of reasoning with and about uncertainty. They mostly differ in how probabilities are represented in OWL, whether or not a Bayesian network is stored in OWL, and the semantics that can be stored within a Bayesian network. In general, much of the reasoning needs to be done outside purely logical reasoners as opposed to our desire to reason with and about variation within the confines of a logical reasoner.

2.2. Ontologies for Representing Statistical Knowledge

A variety of ontologies and ontology design patterns (ODPs) incorporate statistical concepts in some fashion. They can generally be classified as defining statistical concepts and procedures, dealing with values and units, or defining data structures that facilitate data analysis (see Section 2.3).

Statistical concepts include measures of variation, measures of central tendency, measures of position, and many more. Procedures are algorithms for calculating values that correspond to these concepts.

[1] Though related, variation and uncertainty are different concepts [6].

Ontologies like GovStat [22], the Ontology for Biomedical Investigations (OBI) [23], The Information Artifact Ontology (IAO) [24], the Ontology of Biological and Clinical Studies (OBCS) [25], and the Statistical Methods Ontology (STATO) [26] all represent many statistical concepts hierarchically and generally capture both concepts and procedures. They tend to suffer from incompleteness (e.g., GovStat focused only on concepts found on US government websites), atypical naming conventions (e.g., OBI uses the ambiguous names *center value* and *average value* for the specific concepts of median and arithmetic mean, respectively), and problematic hierarchies (e.g., OBCS classifies the descriptive statistics *percentile* and *interquartile range* as inferential statistics). These and similar issues, along with their focus on concepts and procedures as opposed to the use of aggregation results, make them unsuitable for reuse in this work.

Statistical calculations result in values which often have an accompanying unit. The ontologies mentioned so far have limited capacity for representing values, units, and related concepts. For that we consider ontologies like the Ontology of units of Measure (OM) 1.8 [27], the Extensible Observation Ontology (OBOE) [28], and the Quantities, Units, Dimensions and Types (QUDT) [29] ontology, as well as the Spatial and Temporal Aggregate Data (STAD) [30, 31] ODP. A common feature is a focus on quantitative measurements (especially in OM and OBOE). While QUDT allows for qualitative values, it is still focused on quantities as evidenced by its focus on *quantity* kinds. We find STAD most useful for two primary reasons: it extends the notion of quantity kind to the more general quality kind (which encompasses both qualitative and quantitative kinds) and it differentiates between observed data values and calculated (aggregate) data values, the latter of which include statistical values. STAD also allows for the representation of underlying datasets and their descriptions as well as algorithms (and their implementations and executions) used for producing aggregates. These concepts are crucial to the appropriate reuse of aggregated values.

2.3. Datacubes

Datacubes are an altogether different approach for structuring data storage to support statistical queries of datasets. The general notion of a datacube does not include semantics; however, The RDF Data Cube Vocabulary [32] (QB) provides a pattern for representing multi-dimensional datasets within an OWL2-based KG where it is also possible to capture semantics. QB4OLAP [33] is an extension of QB for integrating data cubes with the online analytical processing (OLAP) [34] model. However, it takes considerable planning to implement a data cube structure, so it is better undertaken when first creating a KG rather than attempting to adapt a KG later. We see this as a disadvantage to this approach.

3. Example Use Cases

To guide the development of and to evaluate the EFive ODP, we rely on two use cases. The first use case involves a synthetic dataset simulating a fictitious medical practice with approximately 100 patients. It includes attributes such as age, height, weight, gender, educational attainment, body temperature, eye color, and patient addresses. This dataset has proven useful for testing EFive's core ideas on statistics across the entire dataset and within subsets defined by individual attributes (e.g., gender, patient type, or zip code) or combinations thereof.

The second use case draws from the Safe Agricultural Products and Water Graph (SAWGraph) [35, 36], an NSF Proto-OKN project. SAWGraph currently contains about 1.5 billion triples, with ongoing expansion. Its data includes extensive records about PFAS environmental testing, including observations and measurements represented using the Contaminant Observations and Samples Ontology (ContaminOSO) [37]. Some of the capabilities of EFive have been motivated by anticipated needs within SAWGraph. For example, stakeholders from the EPA and other government agencies have highlighted the challenges posed by so-called "non-detects"—measurements that fall below detection or quality assurance thresholds but do not necessarily indicate the complete absence of the tested contaminant. These common cases complicate data analysis and prompted the recommendation of non-parametric methods, such as the five-number summary, for more robust statistical summarization. Although EFive

has not yet been applied to SAWGraph data, we expect to use it for future testing and evaluation of the pattern at scale.

Examples of specific questions relevant to the two use cases help illustrate the kinds of knowledge the pattern should represent and the types of questions it should help answer. They suggest the following broader categories of questions:

- How does the variation in data for A compare with the variation in B?
  - How does the IQR for adult male patient height compare with the IQR for adult females in a medical practice? for adult males vs. females living in the 04411 zip code?
  - Do private wells in Illinois show more/less variation in PFOA levels than in Ohio? in 2024?
- Is a specific data value an *outlier*?
  - In a medical practice, is a 57 inch tall male adult an outlier? among men who are overweight (BMI > 25)?
  - In the state of Maine, is a public water supply PFOS level of 500 ng/g an outlier? in Hancock County? for public water supplies within one mile of a known PFOS source?
- Is a specific data value *typical*? (Define typical as in the middle 50%.)
  - In a medical practice, is 52 in. a typical height for a 10 year old male? in the 04411 zip code?
  - In Kansas, is an aquifer measurement of 25 ng/g PFOS typical? for the Ozark Aquifer?
- Into which *quartile* does a given data value fall?[2]
  - In a medical practice, in which quartile does a 100 kg male fall? a 100 kg male among males aged 40-50?
  - In the state of Maine, in which quartile does a 25 ng/g PFOS level for a private well fall? a 25 ng/g PFOS level among shallow (less than 100 ft deep) wells?

More broadly, the EFive ODP is expected to be applicable across a wide range of applications, including business intelligence (e.g. comparison of different classes of customers, products or locations), environmental AI (e.g. identification of locations with similar ecological or climate patterns), census data (e.g. aggregation and comparison of locations by demographics), and health and biomedical applications (e.g. comparison of patient populations and responses to treatment). Once information from any of these areas is represented via a domain ontology, the inclusion of EFive would add a way to store precomputed measures of variation with the ontology or a knowledge graph (KG) based on it. This added statistical knowledge can then be accessed like any other data from the graph, facilitating queries that, for example, compare the variability of disjoint subsets of a dataset against each other (e.g., county-by-county or zip code-by-zip code within a state). And once something is explicitly represented in an ontology or a KG it can be reasoned over as well.

The work with these use cases has helped define requirements that inform our choice of approach and the five-number summary. Most importantly, the pattern should represent a non-parametric measure of statistical variation. It should explicitly represent datapoints as statistical aggregates. As shown in the questions above, the pattern should support questions that compare variation across and within classes, allow for the classification of particular datapoints as outliers or as typical values, and indicate the approximate position of a datapoint within a distribution. Further, the pattern should be reasonably simple to apply to existing ontologies and KGs and not require changes to underlying ontologies or datasets.

[2] A five-number summary creates four intervals: $[\min, Q_1]$, $[Q_1, \text{median}]$, $[\text{median}, Q_3]$, and $[Q_3, \max]$ which are often referred to as the first, second, third, and fourth quartiles, respectively.

4. Approach

Ultimately, the EFive ODP is intended to be implemented in OWL2-based KGs where querying and reasoning over aggregate statistics—focusing on variation—is desired. Measures of variation are generally expected to be calculated over a given class or over disjoint partitions of a class (e.g., private water wells partitioned by counties within a given state or by depth). They can be used for comparisons across classes as well as within classes. Our approach centers on three key components: the adoption of the five-number summary as a statistical foundation, the reuse of existing ontology design patterns to represent aggregates, and an emphasis on interoperability with domain ontologies.

Five-Number Summary Typical introductory statistics courses take the frequentist approach and include the range, variance, standard deviation, and interquartile range (IQR) as *measurements of variation*. Motivated by the use cases and because we purposely seek a representation that is broadly applicable and agnostic toward the type of distribution a dataset may have, we choose to work with the five-number summary of exploratory data analysis [38] as a simple statistical summary of data. A five-number summary consists of the minimum, first quartile (Q_1), median, third quartile (Q_3), and maximum values for a dataset. As such, it provides a simple yet powerful non-parametric statistical summary that includes not only a measure of central tendency (the median) but also two simple measures of variation: the range (range = max − min) and the interquartile range (IQR = $Q_3 - Q_1$).

Reuse of STAD and QUDT Another critical element of our approach is the reuse of the Spatial and Temporal Aggregate [30] ontology design pattern (STAD) as well as the Quantities, Units, Dimensions and Types [29] ontology (QUDT). A five-number summary is a set of aggregate statistics about a dataset and STAD extends QUDT for the express purpose of better representing aggregate statistics. As we discuss in more detail in Sections 5.2, 5.3, and 5.5, STAD provides useful concepts for representing not only statistical aggregate values but also their base datasets and the algorithm(s) involved, which facilitate semantic integration and comparison of data across datasets.

Interoperability with Domain Ontologies As discussed in Section 2.3, datacubes are an example of an existing approach for representing data that supports a wide-variety of statistical analyses. However, datacubes require the data to be formatted in specific ways to define the structure of the cube, which can make adapting existing ontologies and datasets difficult and time-consuming. Conversely, the EFive ODP has been designed to co-exist with existing ontologies and their associated KGs. While it is important to know the structure of an ontology to use its associated data with EFive, it is not necessary to change the structure of or design the ontology in any particular way.

5. The EFive ODP

We now present how we model ontologically five-number summaries by the Extended Five-Number Summary (EFive) ontology design pattern (ODP). We first provide a more in-depth review of the five-number summary in Section 5.1. We introduce its conceptualization as an ODP in Section 5.2, and then generalize and extend the ODP—thus the "Extended" in its name—in Sections 5.3 and 5.4, respectively, before discussing more tangential aspects arising from its integration with the STAD ODP.

5.1. The Five-Number Summary

The five-number summary, introduced by Tukey in 1977 [38], is a widely accepted summarization method within descriptive statistics. The measures that constitute a five-number summary are useful on their own or for other statistical analyses like comparison of central tendency and spread, outlier detection ("outside" and "far out" per Tukey), and spatial techniques such as median polish [39].

The five-number summary of a dataset includes the minimum, first quartile (Q_1), median (second quartile or Q_2), third quartile (Q_3), and maximum values from the dataset. The median, a measure of

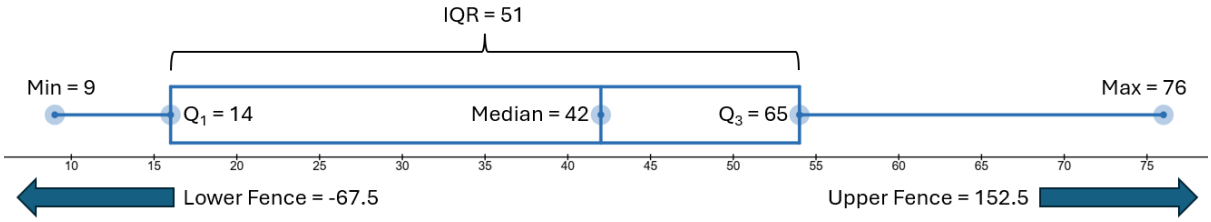


Figure 1: An example of a five-number summary presented as a box-and-whisker plot and annotated with the interquartile range (IQR) and fences.

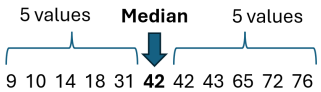
central tendency, is determined by putting the data values in ascending order and selecting the middle value. The median therefore bisects the data into two smaller datasets of equal size. We get Q_1 if we repeat this process on the dataset values to the left of the median and Q_3 if repeated on the values to the right of the median. The original dataset is now subdivided into four parts, each containing essentially the same number of values (within 1), where min $\leq Q_1 \leq$ median $\leq Q_3 \leq$ max.

For example, consider the ordered dataset $[9, 10, 14, 18, 31, 42, 42, 43, 65, 72, 76]$. Its five-number summary is shown in Figure 1 in the form of a box-and-whisker plot. It has a minimum of 9 and maximum of 76. As visualized in Figure 2, the median value is 42 with 5 values smaller and larger, while the median of the first half is 14 and that of the second half is 65, which are the values for Q_1 and Q_3, respectively.[3]

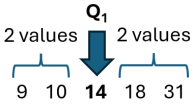
These five values allow us to capture variation in at least three different ways: range, interquartile range (IQR), and outliers. The range is simply the difference between the maximum value and the minimum value (max − min). Since it is generally considered to be of limited use, particularly because it is strongly influenced by outliers [40], we choose not to include it explicitly in this work. Instead, we rely on the IQR as our principle measure of variation. It is the difference between the third and first quartiles: IQR $= Q_3 - Q_1$. Outliers have little to no influence on this value, making it a much more useful and robust comparator across datasets. It provides a sense of the spread of the data values as well as a glimpse into their distribution. The IQR also supports a standard method for detecting outliers that labels any data value less than $Q_1 - 1.5 \cdot$ IQR or greater than $Q_3 + 1.5 \cdot$ IQR as an outlier. Identifying outliers this way can enhance our understanding of variation and can reveal important characteristics, such as skewness or long tails, in the distribution of the dataset. For example, consider Figure 1. Recall that each of the four segments of the plot contains ~25% of the dataset and note how values "clump" in the first and third segments, while the second and fourth segments show greater spread, indicating an uneven distribution across the entire dataset. As an historical aside, Tukey referred to the values at $Q_1 - 1.5 \cdot$ IQR and $Q_3 + 1.5 \cdot$ IQR as inner fences. We simply refer to them as *fences* as we do not model outer fences in the ODP.

We have chosen not to use standard deviation (and, by extension, variance) as a measure of variation. We do so for three reasons: IQR is simpler to calculate and understand, it does not imply any specific distribution (e.g., normal), and it provides useful thresholds for outlier detection. Using IQR to identify

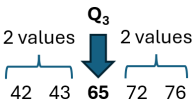
[3]This is a simplistic example always with odd numbers of values so there is an obvious middle value. If there are an even number of values, the median is defined as the mean of the middle two numbers.



(a) The median of an ordered dataset.



(b) The first quartile (Q_1) of the dataset.



(c) The third quartile (Q_3) of the dataset

Figure 2: A visual example of determining the median, Q_1, and Q_3 of a dataset.

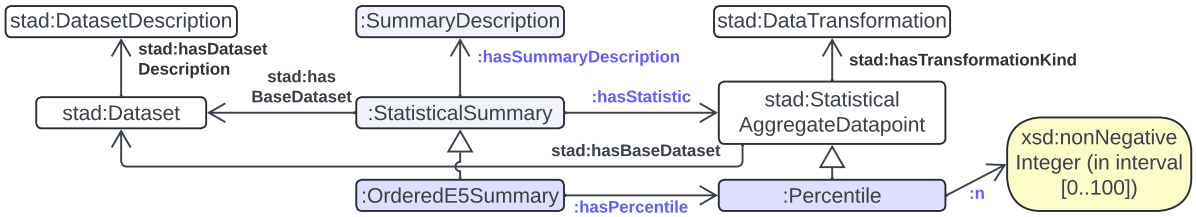


Figure 3: The core structure of the EFive ODP, organized around the classes `:Percentile` and `:OrderedE5Summary` (represented in a slightly darker shade) and their generalizations to `stad:StatisticalAggregateDatapoint` and `:StatisticalSummary`.

outliers is similar to the $\pm 3\sigma$ threshold for normally distributed data (for normal data it is approximately equivalent to $\pm 2.7\sigma$ which classifies 0.70% of normally distributed data as outliers as opposed to 0.27% at $\pm 3\sigma$), but works well regardless of statistical distribution.

5.2. The Core Pattern: `:Percentile` and `:OrderedE5Summary`

Figure 3 shows the high-level conceptual structure of the EFive ODP.[4] At its heart are the classes `:Percentile` and `:OrderedE5Summary`. An `:OrderedE5Summary` represents a five-number summary, which consists of a collection of five statistical values, each linked to it via the `:hasPercentile` object property. Each of them can be thought of as a specific percentile of the form P_n where n is any integer in the closed interval $[0, 100]$. This is captured by the `:Percentile` class and its datatype property `:n` to represent the percentile number n, restricted to the interval $[0, 100]$ in OWL2 using a datatype restriction involving `xsd:minInclusive` and `xsd:maxInclusive`. A cardinality restriction is placed on `:Percentile` requiring instances to have exactly one `:n` property. The `:Percentile` class is modeled as a subclass of `stad:StatisticalAggregateDatapoint`, which encompasses statistical aggregates as opposed to specific observations or predictions from a model. The algorithm used to calculate a `:Percentile` is represented by the associated `stad:DataTransformation`.

We use the shortcut notation `:P<n>` to denote named subclasses of `:Percentile` with a fixed n. The statistical values `:Minimum`, `:Q1`, `:Median`, `:Q3`, and `:Maximum` can then be represented as `:P0`, `:P25`, `:P50`, `:P75`, and `:P100`, respectively. The axiomatic definitions of these `:P<n>` in OWL2 are exemplified by the following definition of `:P50`:

```
:P50 owl:equivalentClass [ owl:intersectionOf ( :Percentile
                                                [ rdf:type owl:Restriction ;
                                                  owl:onProperty :n ;
                                                  owl:hasValue "50"^^xsd:integer
                                                ] ) ] .
```

All these defined classes are subclasses of `:Percentile`; the hierarchy is shown in Figure 4. They are associated with an `:OrderedE5Summary` by specific properties :hasMin, `:hasQ1`, etc., which are all subproperties of the generic `:hasPercentile` object property. They are shown as attributes inside the `:OrderedE5Summary` class in Figure 5.

Only data that can be *ordered* can have a five-number summary, thus we name the appropriate class `:OrderedE5Summary`. It is further described by an associated `:SummaryDescription`; the dataset it summarizes can be specified and described further by the associated `stad:Dataset` and `stad:DatasetDescription` classes that we reuse from STAD.

5.3. Generalizing the `:OrderedE5Summary`

We have focused thus far on the `:OrderedE5Summary` which is modeled for use with ordinal scale data. However, it does not apply to nominal scale data and is inappropriate for measures like IQR which require interval or ratio scale data.

[4]The EFive ODP and supplementary materials, including sample queries, are available at https://github.com/theSKAILab/EFive.

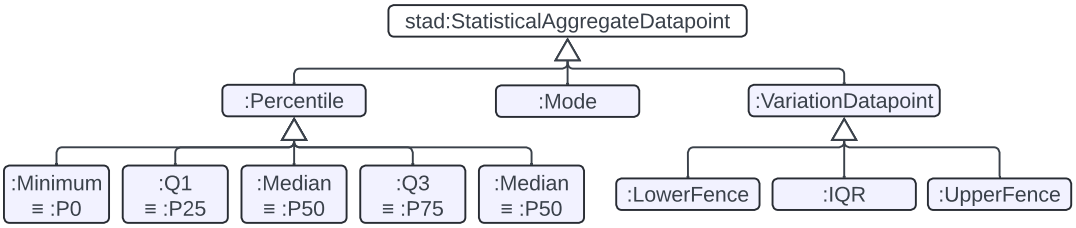


Figure 4: The class hierarchy for statistical aggregate datapoints in EFive.

As a brief refresher on measurement scales [41], nominal scale data has no natural order (e.g., gender, eye color), ordinal scale data can be ordered but lacks consistent differences (e.g., educational attainment), interval scale data can be ordered and has consistent differences but lacks meaningful ratios (e.g., body temperature), and ratio scale data can be ordered, has consistent differences, and has meaningful ratios (e.g., height, weight). We note that, typically, nominal and ordinal scale data are qualitative while interval and ratio level data are quantitative. As shown in Figure 5, we extend the `:AggregateVariable` class to fully capture the measurement scale hierarchy (see also Section 5.3.4). An `:OrderedE5Summary` can summarize an `:AggregateOrdinalVariable` and, likewise, we restrict instances of `:QuantitativeE5Summary` in Section 5.4 to summarizing an `:AggregateIntervalVariable` or an `:AggregateRatioVariable`. While, in everyday language, we talk about measurement scales as disjoint, their definition clearly reflects a nested hierarchy.

5.3.1. The `:StatisticalSummary` Class

The hierarchy of measurement scales for variables suggests an analogous class hierarchy for the statistical summaries like `:OrderedE5Summary`. As shown in Figure 5, we introduce at the top of the hierarchy the `:StatisticalSummary` class that can be used with all kinds of data, including nominal scale data. With none of the five values from a five-number summary being applicable to nominal scale data, we have added the `:Mode` as another subclass of `stad:StatisticalAggregateDatapoint`, which is inherited by all of the other summary classes. `:OrderedE5Summary` is then a subclass of `:StatisticalSummary`.

We introduce `:hasStatistic` as a generalization of `:hasPercentile` and `:hasMode`. Therefore, all object properties that link a `:StatisticalSummary` to a `stad:StatisticalAggregateDatapoint` are subproperties of `:hasStatistic`. This creates an object proprety hierarchy that mirrors the class hierarchy of Figure 4.

5.3.2. Extending STAD's Dataset Class

A `:StatisticalSummary` summarizes a variable over a dataset. That dataset is a class or a subset of a class. Subsets of classes are created using attribute filters; for example, the set of all patients (class) who are male (filter) and live in the 04411 zip code (filter). These datasets reside in the ABox of some ontology or KG and EFive uses the `stad:Dataset` class to represent them (see Figure 6).

There are two links via the `stad:hasBaseDataset` property to a dataset, one from the `:StatisticalSummary` and one from each `stad:StatisticalAggregateDatapoint`. The latter one was already provided by STAD while the former is an addition. Its introduction allows us to explicitly express that a `:StatisticalSummary`—which may consist of multiple aggregate datapoints, e.g., five aggregate datapoints for a five-number summary—is tied to a single dataset and each `stad:StatisticalAggregateDatapoint` that is part of that summary must be calculated over that *same* dataset. We enforce this axiomatically with a combination of restrictions that, collectively, have the desired effect: the composition of `stad:hasBaseDataset` with `:hasStatistic` is defined as a subproperty of `stad:hasBaseDataset` while cardinality restrictions require each `:StatisticalSummary` and `stad:StatisticalAggregateDatapoint` to be related to exactly one `stad:Dataset`.

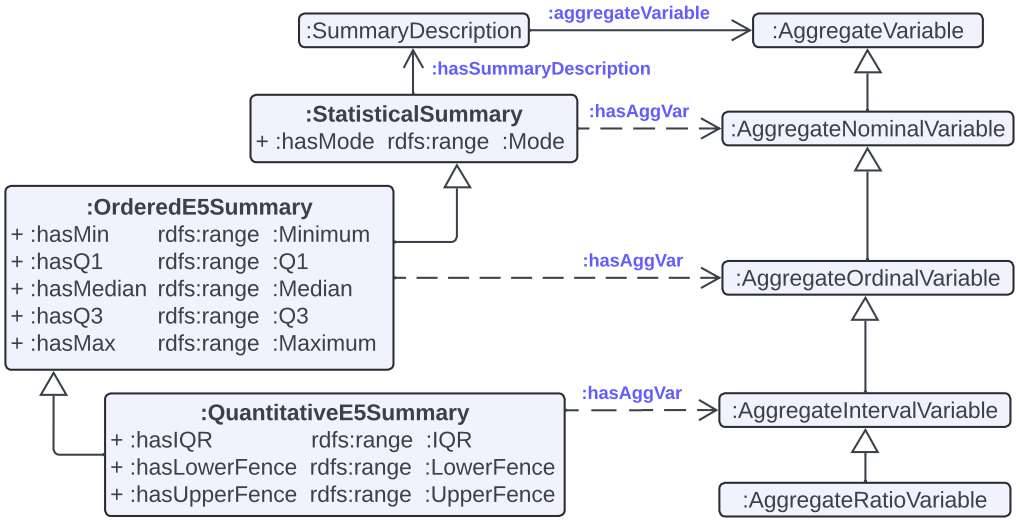


Figure 5: Each instance of one of the three summary classes represents a collection of statistical values that together from a statistical summarization of some aggregate variable over some dataset (statistical object properties are included in UML-style as attributes). Each summary has a `:SummaryDescription` that must specify an `:AggregateVariable`. Both the summary and aggregate variable classes align with the measurement scales such that a `:StatisticalSummary` can summarize any type of aggregate variable, but a `:QuantitativeE5Summary` can only summarize an `:AggregateIntervalVariable` or an `:AggregateRatioVariable`. The dashed arrows indicate the inferred composite object property `:hasAggVar`.

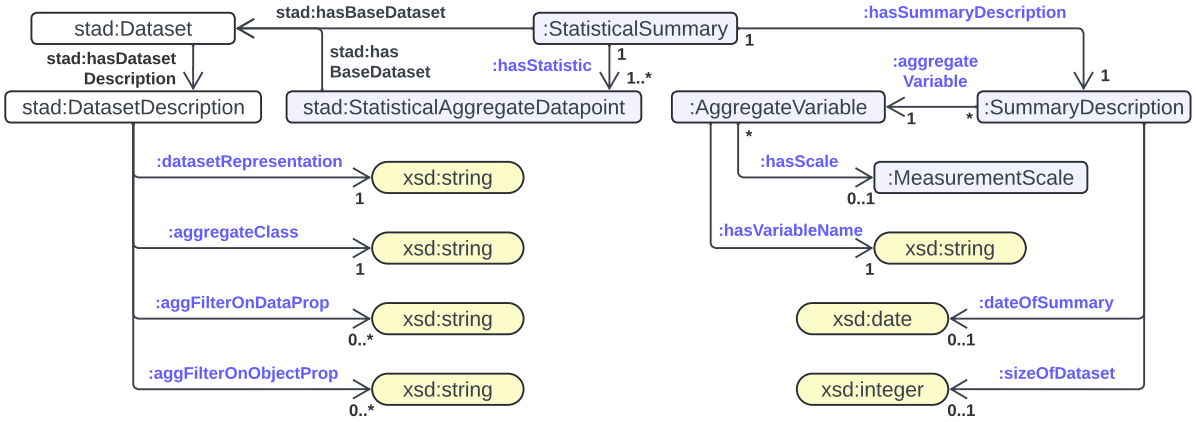


Figure 6: The dataset/dataset description pattern from EFive is shown on the left and the summary description pattern is shown on the right. While datasets and their descriptions may be reused across multiple summaries, each summary description is unique to its summary. See Sections 5.3.2, 5.3.3, and 5.3.4 for more details.

5.3.3. Extending STAD's DatasetDescription Class

The `stad:DatasetDescription` class provides a detailed description of a dataset to support user understanding and to enhance the reusability of `:Dataset` instances across multiple EFive summaries. EFive extends it by adding the data property `:datasetRepresentation` to allow describing the related dataset in more detail using OWL2 class expressions. It is a string representation of either the name of a class or, if filters are used, a class expression that uses OWL2 property restrictions. As an example, consider an `:OrderedE5Summary` that represents the median educational attainment of male patients with addresses in the 04411 zip code. The corresponding dataset can be described—as a string—as the intersection of the patient class, a property restriction on gender with value *male*, and a property restriction on zip code with value *04411*. This representation provides a mechanism to quickly find the slice of data from which a descriptive statistic was calculated. Moreover, it simplifies updating a summary as the ABox changes and instances and attributes of this slice may change.

The annotation properties :aggregateFilterOnDataProp, :aggregateFilterOnObjectProp, and :aggregateClass capture some of those details of the dataset definition in more granular form to faciliate the automated generation of instances of :StatisticalSummary from an ABox. The properties :aggregateFilterOnDataProp and :aggregateFilterOnObjectProp are only needed when the :aggregateClass is filtered by some attribute(s). Each can be used with individual properties or with property paths.[5] There can be as many of each of these as needed.

5.3.4. The :SummaryDescription Class

The instances in a dataset are typically related to multiple attributes, so it is possible to have different instances of :StatisticalSummary that aggregate over different variables from the same dataset. For example, we can consider the mode of eye color or the median of weight for the same set of male patients. Therefore, the variable of interest is specific to a summary, rather than being a characteristic of the underlying dataset, as are the date and the size of the dataset when the summary was created.

EFive uses the :SummaryDescription class (see Figure 6) to represent summary-specific details. The data properties :sizeOfDataset and :dateOfSummary capture when the summary was created as well as the size of the underlying dataset at that time. The object property :aggregateVariable and the class :AggregateVariable as its range represent the variable of interest. :AggregateVariable has the data property :hasVariableName and the object property :hasScale. The latter more explicitly captures the variable's measurement scale—also captured by the summary and aggregate variable hierarchies—using a controlled vocabulary of instances of :MeasurementScale: :NominalScale, :OrdinalScale, :IntervalScale, and :RatioScale. Each of them are connected to related QUDT classes using skos:closeMatch.

5.4. Specializing the :OrderedE5Summary to Capture Variation

Neither the original :OrderedE5Summary nor its :StatisticalSummary generalization address yet how to capture variation. To do so requires working with interval or ratio scale data because we need consistent differences among values for measures like IQR and fences. We therefore introduce :QuantitativeE5Summary as a specialization of :OrderedE5Summary equipped with three additional properties and associated classes to describe variation within a dataset. The classes for representing variation are :IQR, :LowerFence, and :UpperFence, all of which are modeled as subclasses of the :VariationDatapoint subclass of stad:StatisticalAggregateDatapoint as shown in Figure 4. Because they are intended for use only with interval or ratio scale data, the domains of the associated object properties—:hasIQR, :hasLowerFence, and :hasUpperFence—are restricted to the :QuantitativeE5Summary class as shown in Figure 5. To continue mirroring the class structure of Figure 4, they are generalized by :hasVariationDatapoint which is a subproperty of :hasStatistic.

We follow common practice and define $:\mathtt{IQR} = :\mathtt{Q3} - :\mathtt{Q1}$. The lower and upper fence are used as thresholds beyond which (less than the lower fence or greater than the upper fence) data values may be classified as outliers. They can then be defined in terms of the IQR as follows:

Definition 5.1 (Lower Fence and Upper Fence)**.** The :LowerFence is a value 1.5 times the :IQR below :P25 and the :UpperFence is a value 1.5 times the :IQR above :P75:

$$:\mathtt{LowerFence} = :\mathtt{P25} - 1.5 \cdot :\mathtt{IQR} \qquad :\mathtt{UpperFence} = :\mathtt{P75} + 1.5 \cdot :\mathtt{IQR}$$

5.5. Reusing Other Features of STAD's StatisticalAggregateDatapoint

Recall that each instance of :StatisticalSummary (or its subclasses) may be linked to multiple instances of stad:StatisticalAggregateDatapoint or any of its subclasses that are shown in Figure 4.

In addition to the classification from Figure 4, each stad:StatisticalAggregateDatapoint must also be an instance of either stad:QualitativeDatapoint or stad:QuantitativeDatapoint. Each

[5]The last property in a path determines whether the filter is an object or data property type.

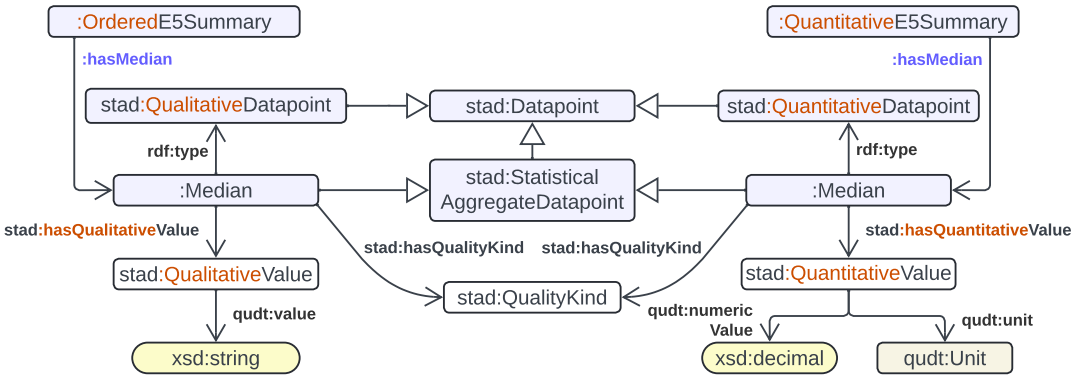


Figure 7: The general pattern for representing statistical values in EFive (here depicting the median) leverages STAD concepts for datapoints, quality kinds, and values, and uses QUDT concepts for the actual value and unit (when applicable). The left pattern is for qualitative data and the right pattern is for quantitative data (differences are highlighted in amber). Note that the `:Median` class is a subclass of `stad:StatisticalAggregateDatapoint` but a subclass of neither `stad:QualitativeDatapoint` nor `stad:QuantitativeDatapoint`. However, all instances of `:Median` in the qualitative case on the left will also be instances of `stad:QualitativeDatapoint` while all instances in the quantitative case on the right will also be instances of `stad:QuantitativeDatapoint`.

datapoint is related to a data value, a quality kind, and a data transformation. These features can be specified by reusing concepts from STAD [30] and QUDT as illustrated in Figure 7 using the example of a `:Median`. They are summarized in the remainder of this section.

Data Values Semantically, a datapoint (`stad:Datapoint`) and its value (`stad:DataValue`) are different things. In the conceptualization we use here (based on STAD and QUDT), a data value includes a value and, when applicable, a unit. A datapoint includes a data value (as just described), a quality kind, and, since we are working with aggregate statistical values, a data transformation.

STAD provides `stad:DataValue` as an extension of the `qudt:QuantityValue` that includes subclasses for both qualitative values (`stad:QualitativeValue`, associated with `stad:QualitativeDatapoint` instances) and quantitative values (`stad:QuantitativeValue`, associated with `stad:QuantitativeDatapoint` instances). A value along with an optional unit are attached to each `stad:DataValue` instance. Figure 7 shows the use of `stad:QualitativeValue` on the left and `stad:QuantitativeValue` on the right. The property `qudt:value` is used for qualitative values, while `qudt:numericValue` is used for quantitative ones. In either case, units are optional and can be added via the `qudt:unit` property.

QualityKind The notion of quality kind goes beyond the value (and unit) of a datapoint and answers the question "What was measured and aggregated?" Examples include length, color, and age. Quality kinds can be adapted to statistical results, like MedianLength using the `stad:StatisticalQualityKind` subclass of `stad:QualityKind`. While the unit associated with a data value is helpful, the quality kind adds additional semantics; for example, QUDT's *quantitykind* ontology[6] includes 57 `qudt:QualityKind` instances that have the unit `qudt:M` (meter), including length, depth, or diameter.

DataTransformation EFive uses the `stad:DataTransformation` class to represent the specific algorithm used to calculate a descriptive statistic. This can be important for data reuse; for example, there are at least 15 different methods to calculate percentiles [42]. The object property `stad:hasTransformationKind` is used to link a `stad:StatisticalAggregateDatapoint` to the specific algorithm used in calculating the statistical value. As outlined in Figure 8, STAD includes a custom version of the Algorithm, Implementation, and Execution ODP [43] using the prefix `stad-mls:`. An algorithm is an instance of `stad:DataTransformation`. Different implementations of an

[6] https://www.qudt.org/doc/DOC_VOCAB-QUANTITY-KINDS.html

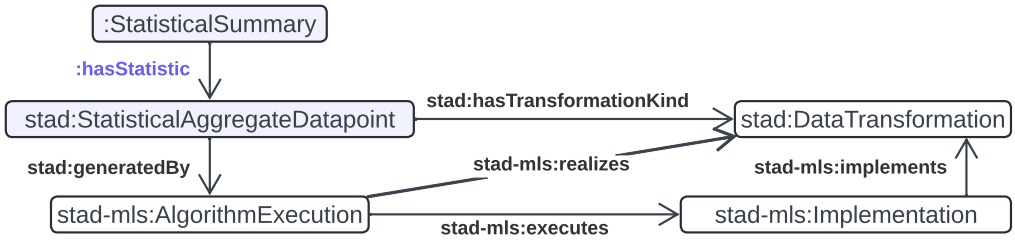


Figure 8: The data transformation pattern from EFive.

algorithm can be captured as instances of `stad-mls:Implementation`. The execution of an algorithm's implementation that generates a specific result can then be represented as an instance of `stad-mls:AlgorithmExecution`. Data transformations from the Ontology for Biomedical Investigations (OBI) [23], STATO, and other ontologies discussed in Section 2.2 can be reused here as well.

6. Summary and Future Work

To the best of our knowledge, no prior work has attempted to represent empirical variation and its semantics using only the expressivity offered by the OWL2 language. This paper spends considerable time developing the EFive ODP as a semantic framework for modeling five-number summaries in OWL2 in a way that enables integration with instance data grounded in other OWL2 domain ontologies. Central to this pattern are the interquartile range (IQR) and the notion of fences, which explicitly capture variation. These values allow users to quantify and compare variation within and across datasets; help identify outliers (values that show excessive variation); and determine whether specific data points fall within a typical range based on some percentile-based distance from the median.

This work is incomplete and ongoing. This paper presents a complete working draft of the ODP that is now readied for more comprehensive evaluation. Considerable work remains to be done:

- More in-depth evaluation, including of the consistency and completeness, sound ontological structure, and validation via implementation, querying, and reasoning with the ODP.
- Extending the pattern to allow for different approaches to variation by defining Q_1 and Q_3 as "hinges" (a term from Tukey [38]) and then expanding to offer additional options, such as the 20^{th} and 80^{th} percentiles, the 10^{th} and 90^{th} percentiles, and the 5^{th} and 95^{th} percentiles.
- Extending the notion of an interquartile range (IQR) to the generalized interquantile range (GIQR) as a measure of variation for different sets of hinges.
- Developing the class `:PositionSet`, instances of which will connect to all quartiles, quintiles, deciles, or twentiles. Users can use position sets to determine where in a distribution specific values lie, get a general sense of what a distribution looks like, and more.
- Making sure summaries can be added to an ontology or KG and subsequently queried and reasoned over, even without access to their underlying data.
- Possibly adding empirical frequency distributions of some form. While we do not have current plans to pursue this line of work, we see it as a useful long term expansion of the ODP.
- Evaluate how the ODP can be used to analyze data that includes "non-detect" values. A result of "non-detect" is different from a result of zero because, while the actual value could be zero, it could also be any value up to some minimum detection limit. Non-parametric methods, like five-number summaries and IQR, may prove helpful in analyzing data that includes these values.

Acknowledgments

This material is based in part upon work supported by the National Science Foundation under Grant Number ITE-2333782.

Declaration on Generative AI

The authors have employed ChatGPT for improving the clarity of the text. The authors reviewed and edited the content afterwards as needed and take full responsibility for the publication's content.

References

[1] C. Reading, Reasoning about variation, The challenge of developing statistical literacy, reasoning, and thinking/Kluwer Academic Publishers (2004).

[2] J. Garfield, D. Ben-Zvi, A framework for teaching and assessing reasoning about variability, Statistics Education Research Journal 4 (2005) 92–99.

[3] C. J. Wild, M. Pfannkuch, Statistical thinking in empirical enquiry, International statistical review 67 (1999) 223–248.

[4] T. Hahmann, S. A. McIlraith, Towards ontologies in variation, in: 2015 AAAI Spring Symposium Series, 2015.

[5] V. Presutti, E. Blomqvist, E. Daga, A. Gangemi, Pattern-based ontology design, in: Ontology Engineering in a Networked World, Springer, 2011, pp. 35–64.

[6] S. H. Begg, M. B. Welsh, R. B. Bratvold, Uncertainty vs. variability: What's the difference and why is it important?, in: SPE hydrocarbon economics and evaluation symposium, SPE, 2014, p. D011S003R002.

[7] N. J. Nilsson, Probabilistic logic, Artificial intelligence 28 (1986) 71–87.

[8] R. Fagin, J. Y. Halpern, N. Megiddo, A logic for reasoning about probabilities, Information and computation 87 (1990) 78–128.

[9] H. E. Kyburg, The reference class, Philosophy of science 50 (1983) 374–397.

[10] F. Bacchus, A. J. Grove, J. Y. Halpern, D. Koller, A response to "believing on the basis of the evidence", Computational Intelligence 10 (1994) 21–25.

[11] D. Koller, A. Levy, A. Pfeffer, P-classic: A tractable probablistic description logic, AAAI/IAAI 1997 (1997) 14.

[12] K. B. Laskey, Mebn: A language for first-order bayesian knowledge bases, Artificial intelligence 172 (2008) 140–178.

[13] R. Giugno, T. Lukasiewicz, P-(d): a probabilistic extension of (d) for probabilistic ontologies in the semantic web, in: European Workshop on Logics in Artificial Intelligence, Springer, 2002, pp. 86–97.

[14] T. Lukasiewicz, Expressive probabilistic description logics, Artificial Intelligence 172 (2008) 852–883.

[15] I. Horrocks, P. F. Patel-Schneider, F. Van Harmelen, From SHIQ and RDF to OWL: The making of a web ontology language, Journal of Web Semantics 1 (2003) 7–26.

[16] F. Bacchus, Lp, a logic for representing and reasoning with statistical knowledge, Computational Intelligence 6 (1990) 209–231.

[17] L. Demey, B. Kooi, Logic and probabilistic update, Johan van Benthem on Logic and Information Dynamics (2014) 381–404.

[18] Z. Ding, Y. Peng, A probabilistic extension to ontology language OWL, in: 37th Hawaii Intern. Conf. on System Sciences, IEEE, 2004.

[19] P. C. G. Da Costa, K. B. Laskey, K. J. Laskey, PR-OWL: a bayesian ontology language for the semantic web, in: Intern. Workshop on Uncertainty Reasoning for the Semantic Web, Springer, 2005, pp. 88–107.

[20] S. Fenz, A. M. Tjoa, M. Hudec, Ontology-based generation of bayesian networks, in: 2009 Intern. Conf. on Complex, Intelligent and Software Intensive Systems, IEEE, 2009, pp. 712–717.

[21] Y. Yang, J. Calmet, Ontobayes: An ontology-driven uncertainty model, in: Intern. Conf. on Computational Intelligence for Modelling, Control and Automation and on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06), volume 1, IEEE, 2005, pp. 457–463.

[22] M. C. Pattuelli, The GovStat Ontology: Technical Report, Technical Report, University of North Carolina School of Information and Library Science, 2004. Accessed on March 27, 2025.
[23] A. Bandrowski, R. Brinkman, M. Brochhausen, M. H. Brush, B. Bug, M. C. Chibucos, K. Clancy, M. Courtot, D. Derom, M. Dumontier, et al., The ontology for biomedical investigations, PloS one 11 (2016) e0154556.
[24] W. Ceusters, An information artifact ontology perspective on data collections and associated representational artifacts, Stud Health Technol Inform 180 (2012) 68–72.
[25] J. Zheng, M. R. Harris, A. M. Masci, Y. Lin, A. Hero, B. Smith, Y. He, The ontology of biological and clinical statistics (obcs) for standardized and reproducible statistical analysis, Journal of Biomedical Semantics 7 (2016) 1–13.
[26] A. Gonzalez-Bertran, P. Rocca-Serra, O. Burke, S.-A. Sansone, Statistics ontology (stato), http://static-ontology.org/, 2012. Accessed on March 27, 2025.
[27] C. Keßler, M. d'Aquin, S. Dietze, H. Rijgersberg, M. van Assem, J. Top, Ontology of units of measure and related concepts, Semantic Web 4 (2013) 3–13.
[28] J. Madin, S. Bowers, M. Schildhauer, S. Krivov, D. Pennington, F. Villa, An ontology for describing and synthesizing ecological observation data, Ecological informatics 2 (2007) 279–296.
[29] R. Hodgson, P. J. Keller, J. Hodges, J. Spivak, QUDT: Quantities, units, dimensions and types, https://qudt.org/, 2022.
[30] K. Wiafe-Kwakye, T. Hahmann, K. Beard, An ontology design pattern for spatial and temporal aggregate data (stad), in: 13th Workshop on Ontology Design and Patterns (WOP 2022), 2022.
[31] K. Wiafe-Kwakye, T. Hahmann, K. Beard, STAD: An ontology design pattern and ontology for the semantic representation of aggregate spatial and temporal data, submitted for review to Semantic Web Journal (2025).
[32] Government Linked Data Working Group, The RDF Data Cube Vocabulary, Technical Report, World Wide Web Consortium (W3C), 2014. Recommendation.
[33] L. Etcheverry, A. A. Vaisman, QB4OLAP: a new vocabulary for olap cubes on the semantic web, in: 3rd Intern. Conf. on Consuming Linked Data, volume 905, CEUR-WS.org, 2012, pp. 27–38.
[34] S. Chaudhuri, U. Dayal, An overview of data warehousing and olap technology, ACM Sigmod record 26 (1997) 65–74.
[35] T. Hahmann, P. Hitzler, H. K. McGinty, G. Hettiarachchi, O. Apul, et al., Safe Agricultural Products and Water Graph (SAWGraph): An Open Knowledge Network to Monitor and Trace PFAS and Other Contaminants in the Nation's Food and Water Systems, https://sawgraph.github.io/, 2024.
[36] K. Schweikert, D. Kedrowski, S. Stephen, T. Hahmann, Precomputed topological relations for integrated geospatial analysis across knowledge graphs, in: 13th Intern. Conf. on Geographic Information Science (GIScience 2025), LIPIcs 346, 2025 (to appear), pp. 4:1–21.
[37] T. Hahmann, K. Schweikert, S. Stephen, D. Kedrowski, ContaminOSO: Ontological foundations and key design choices for an ontology for environmental contaminant data, in: 25th International Conference on Formal Ontology in Inf. Systems (FOIS-25), IOS Press, 2025 (to appear).
[38] J. W. Tukey, Exploratory Data Analysis, Addison-Wesley, 1977.
[39] G.-H. Strand, Large-scale variations in radial tree growth in norway: an application of median polish for spatial trend detection, Applied Geography 18 (1998) 153–168.
[40] R. W. Cooksey, Descriptive statistics for summarising data, Illustrating statistical procedures: Finding meaning in quantitative data (2020) 61–139.
[41] S. S. Stevens, On the theory of scales of measurement, Science 103 (1946) 677–680.
[42] E. Langford, Quartiles in elementary statistics, Journal of Statistics Education 14 (2006).
[43] A. Ławrynowicz, D. Esteves, P. Panov, T. Soru, S. Džeroski, J. Vanschoren, An algorithm, implementation and execution ontology design pattern, in: Advances in Ontology Design and Patterns, IOS Press, 2017, pp. 55–68.