The Six Dimensions of Data Quality — and how to deal with them

Building your models and analysis on solid foundations

Richard Farnworth Jun 28, 2020 · 9 min read ★

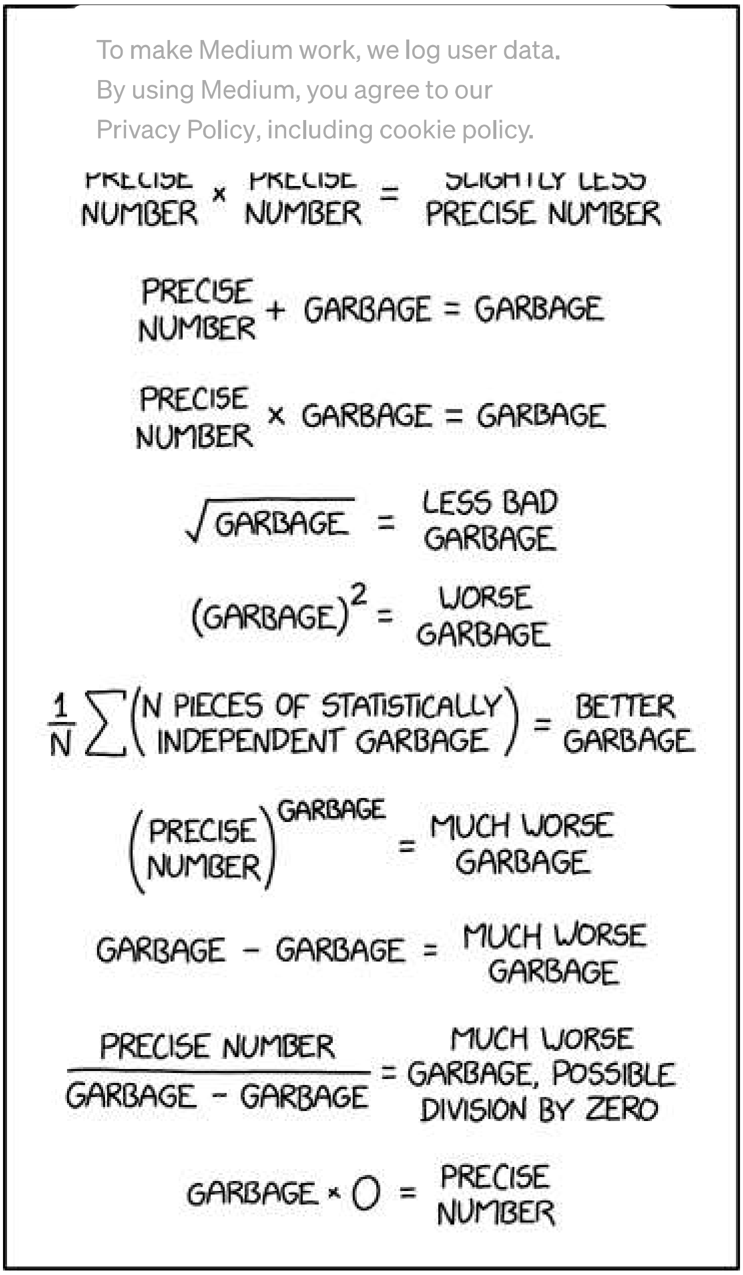

PRECISE NUMBER × PRECISE NUMBER = SLIGHTLY LESS PRECISE NUMBER
PRECISE NUMBER + GARBAGE = GARBAGE
PRECISE NUMBER × GARBAGE = GARBAGE
√GARBAGE = LESS BAD GARBAGE
(GARBAGE)² = WORSE GARBAGE
1/N Σ (N PIECES OF STATISTICALLY INDEPENDENT GARBAGE) = BETTER GARBAGE
(PRECISE NUMBER)^GARBAGE = MUCH WORSE GARBAGE
GARBAGE − GARBAGE = MUCH WORSE GARBAGE
PRECISE NUMBER / GARBAGE − GARBAGE = MUCH WORSE GARBAGE, POSSIBLE DIVISION BY ZERO
GARBAGE × 0 = PRECISE NUMBER

Image by XKCD

Garbage in, garbage out. So goes the familiar phrase, born in the early days of Computer Science, pressing the importance of validating your inputs.

You can have the most ingenious, elegant, well-tested function, model or application — but what comes out is only as good as what goes in.

Whenever we develop code, we make assumptions ahead of time about the nature of the data it will process; A simple arithmetic function might expect a single floating point number. A demand forecasting model for an refreshments kiosk could expect the last five years of sales figures in a particular tabular form. And a self-driving car controller would take in different streams of data from many sensors around the vehicle.

If these assumptions are happen.

- The code checks the es plan B. This could be to warn the user of the problem with the data and gracefully stop.
- The code hits a runtime error, that crashes the program.
- The code continues on, oblivious to the erroneous inputs, and produces a potentially plausible, but incorrect output.

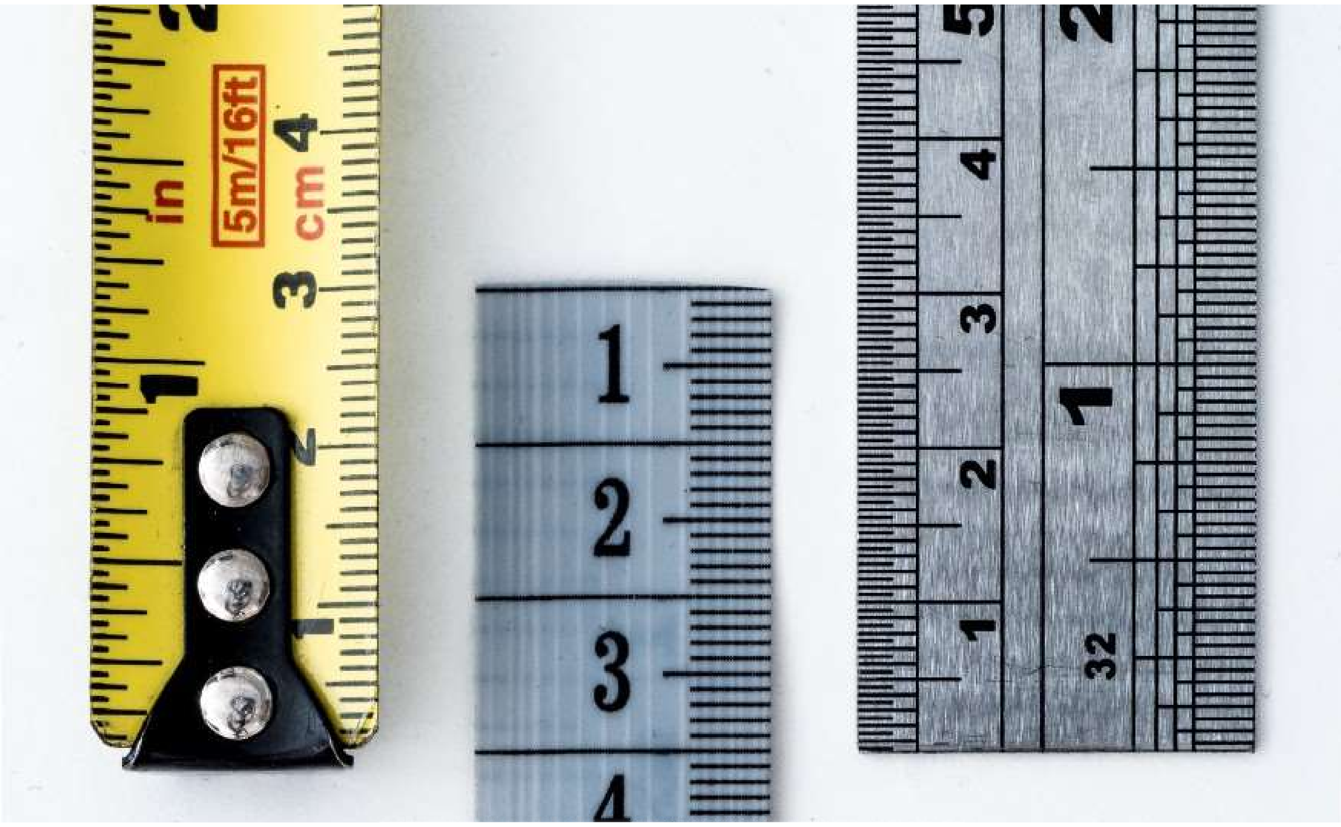
The first scenario gives you a parachute, the second gives you a headache and the third gives you a multi-car pileup in a puddle of melted Cornetto.

Bad Data => Bad Decisions

As organisations becomes more data-mature, important business decisions are more frequently reliant on data analysis and modelling. If the data on which those decisions are made is not up to scratch, then the reasoning you base on that data will be flawed, with potentially very expensive consequences.

This is why understanding Data Quality and being aware of the many ways the data you're using could fall short of your requirements is so important.

Accuracy



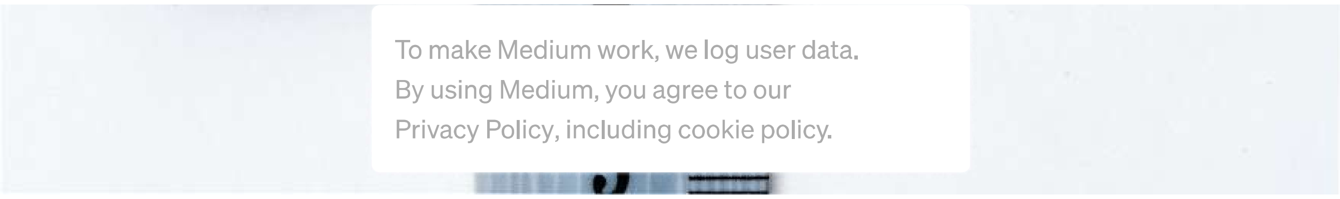


Photo by William Warby on Unsplash

Every piece of data ever created, originated as an event or measurement in the real world. This could be the output of a temperature sensor, the logging of a financial transaction or someone typing their name into a web form. Accuracy describes the "degree to which the data correctly describes the 'real world' objects being described."

In order for this to be achieved, each step on the journey from real-world to data-set has to correctly preserve the essence of the original.

A likely place for errors to occur is right at the start, during the measurement or recording of the event/object. In May 2020, the Australian government overestimated its spending commitments for a COVID 19 wage subsidy scheme by AUD $60 Billion (USD $39 Bn), due to mistakes made filling in a confusing application form. Employers were asked to state the number of employees they were enrolling in the scheme. However, in 0.1% of cases, they instead submitted the dollar value of the subsidies they required — *1,500 times* the correct amount. These errors were missed and their aggregated value flowed into a bill passed by parliament. A few weeks later the government announced its mistake, red faced, but probably not too displeased for finding $60 Bn down the back of the sofa.

In the above example, simply listing the top 100 or so claimants would have likely shed light on the issue. You'd expect to find large fast-food and retail brands, hotel chains etc. but when you come across a local restaurant or small tour company claiming for thousands of employees, you know that there is a problem.

This highlights the importance of basic analysis and profiling to understand your dataset. Before you do any reporting or modelling, you need to be looking closely at each field to see if its values make sense, with no strange surprises.

Accuracy has a closely related cousin: precision. Stage times in the Tour de France are recorded in hours and seconds, but this wouldn't work in the 100m final at the Olympics. Precision can be lost during data type conversion or due to the sensitivity of the instrument used to take the initial measurement and can result in lower variance being available to your model.

Completeness

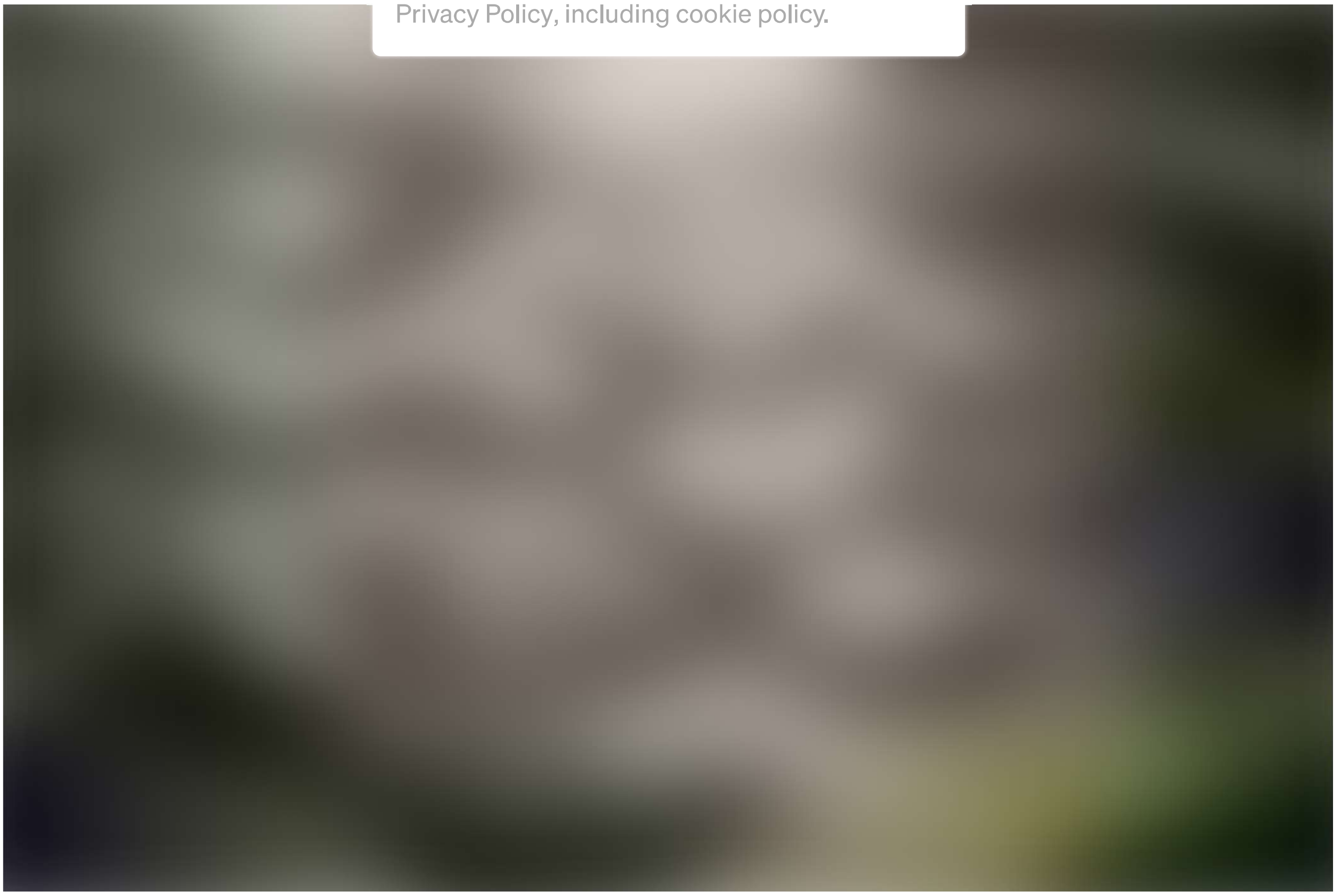


Photo by Gabriel Crismariu on Unsplash

Data completeness denotes the “degree to which required data are in the dataset.” Any dataset may have gaps and missing data, but does that missing data impact on your ability to answer the questions you have. The key thing to understand is whether a bias has been introduced which will affect your results.

In 1936 *Literary Digest* produced a poll asking whether respondents would vote for Republican Alfred Landon, or incumbent Democrat Franklin D. Roosevelt. The mailing list however, was chosen mainly from a telephone directory. Now, in 1936, telephones were far from universal and were considered a luxury item. Therefore, the mailing list ended up being biased towards upper and middle class voters, by its omission of those who could not afford a telephone. Once the responses were in, *Literary Digest* correctly predicted a landslide. Unfortunately however, a landslide for Landon, rather than Roosevelt, who ended up securing 46 of 48 states in one of the most one-sided elections in US history. By using a more complete dataset such as the electoral roll, or at least by understanding and adjusting for the bias created by their missing data, the polling figures could have been nearer the mark.

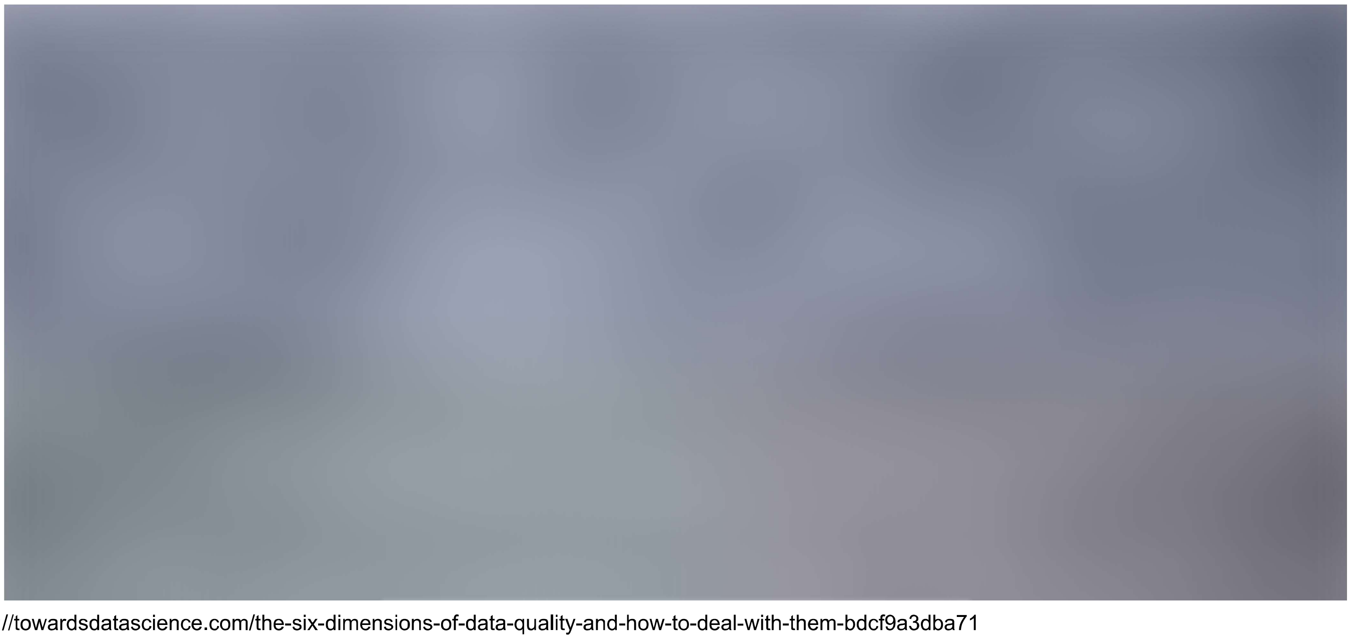
Completeness issues can re you are missing entire rows, but can also uld be blank 80% of the time. This can trip up ma an again introduce biases if the missing values are not uniformly distributed. To mitigate this issue, there are two approaches;

- Throwing away the incomplete column
- Throwing away the rows containing missing data
- Imputing the missing data (a.k.a. filling in the blanks)

A thorough run-through of the above approaches is contained in Yoghita Kinha’s excellent article, How to Deal with Missing Values in your Dataset.

A sneakier and more insidious completeness issue is where a default value gives the illusion of a complete field, despite the real response rate being much lower. This once happened to me when I was building a customer lifetime value model for a retail fashion brand. On each customer profile was a “Gender” flag — M for male, F for female. The field had a high level of completeness in the dataset, but suspicions arose when some basic analysis revealed a sharp swing towards female customers about 6 months prior. It turned out that in a recent redesign of the sign up form, the gender field had changed from a required drop-down box with no default value, to a drop-down with default = “Female”. This small change meant that customers who had ignored that field were now recorded as female, rather than being sent back up the form to fill it in.

Consistency



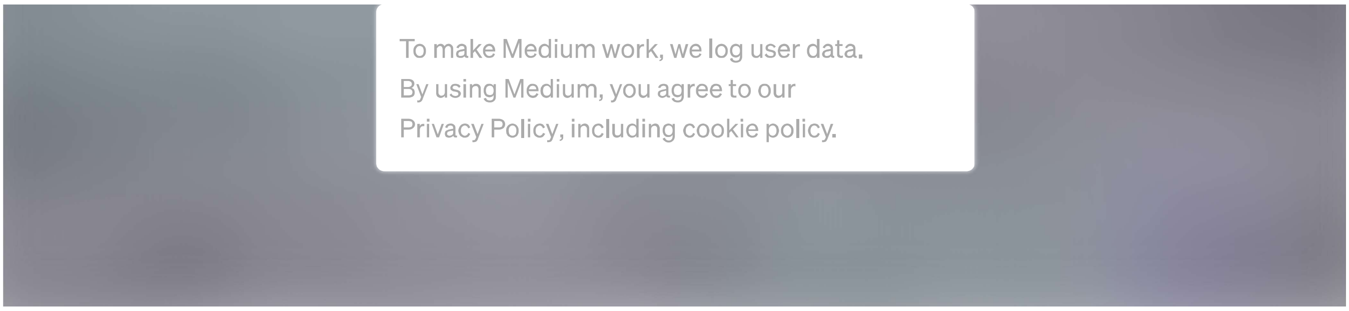


Photo by Ben Coleman on Unsplash

If data is replicated in multiple places, it needs to be consistent across all instances. For a department store, you might hold data on a particular customer through a loyalty program, mailing list, online accounts payment system and order fulfilment system. In that tangled mess of systems there may be misspelled names, old addresses and conflicting status flags. This could cause problems in processes that read data from only one instance of a data point, for example, if a customer has unsubscribed from marketing emails, but this isn't reflected for all representations of that customer, they might continue to receive communications, much to their frustration. Inconsistent contact information can also lead to money wasted in making phone calls or sending letters into the abyss.

In the above example, having a well integrated customer data platform such as Segment or Omneo can help pull together a single view of the customer and ensure that issues around consistency are resolved.

Timeliness

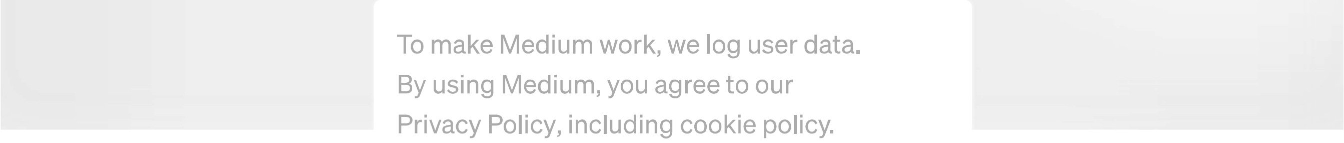


Photo by Insung yoon on Unsplash

Are your datasets sufficiently up to date? What lag is there between an event happening and it appearing in your data. Much data analysis and modelling will be built on historical snapshots, and so running them right up to the present day may not be necessary. However, real-time decision making requires real-time data. If the data from a radar system could only be downloaded in batches, once a day, it wouldn't be of much help to air-traffic controllers. And if Sunday's sales data is delayed for certain stores due to connectivity issues, your figures for Monday's management meeting are going to be off.

The timeliness of your dataset is likely to be dependent on the Data Integration pipeline that led to its creation. This could be real-time, making data available very soon after the event it describes, or processed in a batch, meaning the data is "frozen" until the next refresh. Changes to this pipeline may allow you access to more up to date data and to be more responsive to recent events.

Uniqueness

Photo by Ricardo Gomez Angel on Unsplash

Each real world object o particular dataset once. I.e. if there is a customer ohnny Doe, despite them actually being the

Any metrics involving customers (number of customers, spend per customer, frequency of purchase), would therefore be thrown off by including duplicate representations of a person.

Uncovering this issue means identifying an appropriate primary key. In the John Doe, Johnny Doe example, they could have different names and Customer IDs, but matching email addresses, which is a strong clue that they are the same person. This means an additional step of data wrangling to consolidate these customer records is needed before any analysis or modelling should be done.

Validity

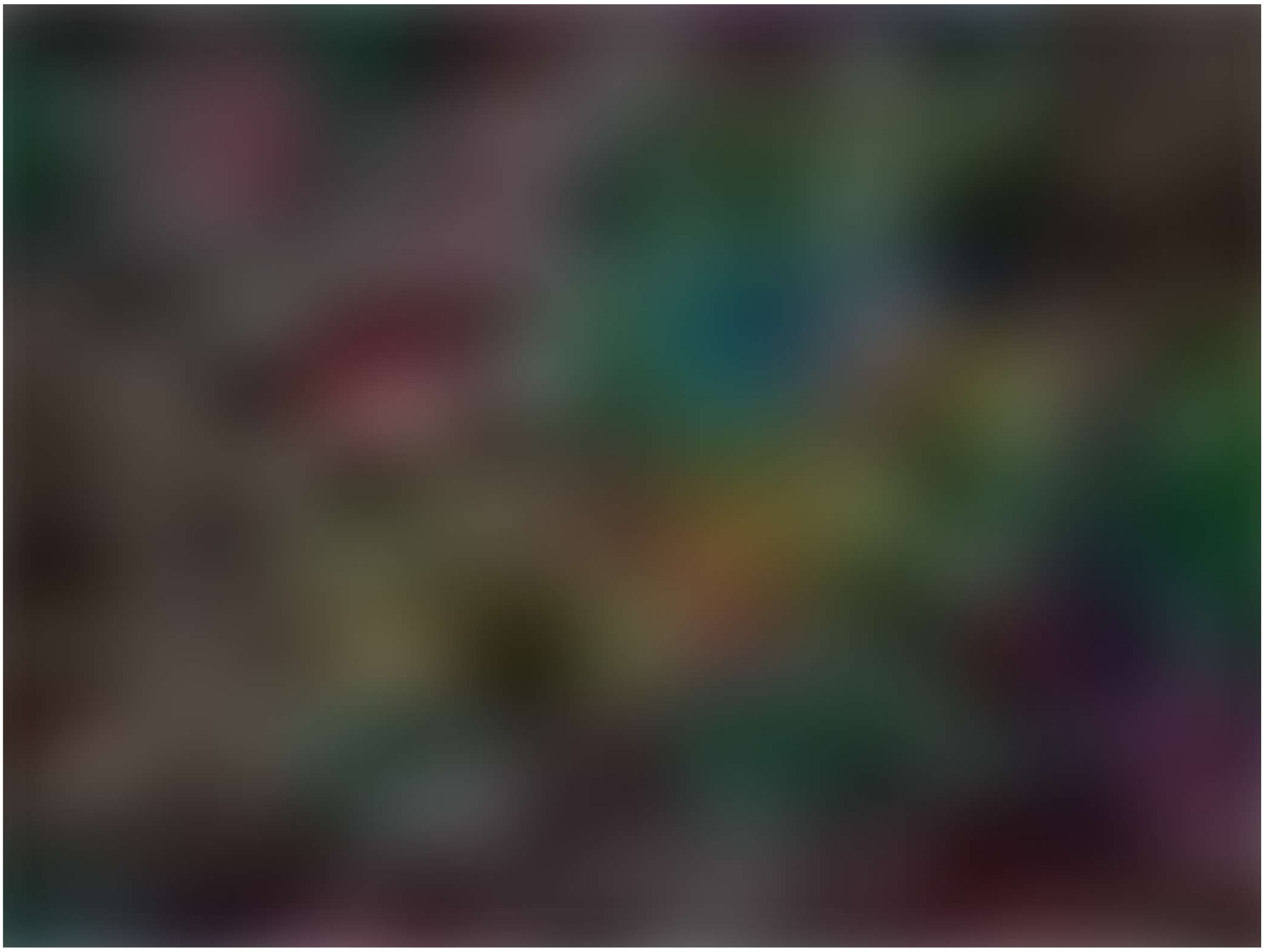


Photo by Max Chen on Unsplash

A field in a dataset may have conditions which it needs to satisfy to be considered valid. An email address must have an '@' symbol, a phone number must be a sequence of numbers and a members "Silver" or "Bronze".

Checking for validity car ; regular expressions. There are databases online such as regexlib.com which contain regular expressions for thousands of common data types. For discrete data types, such as the membership tier example above, simple frequency statistics can tell you whether you have a validity issue. If you have a large number of values other than "Gold", "Silver" or "Bronze", then something is going wrong.

Once invalid data has been identified, it effectively becomes a completeness problem, which can be dealt with using the approaches described previously.

In summary

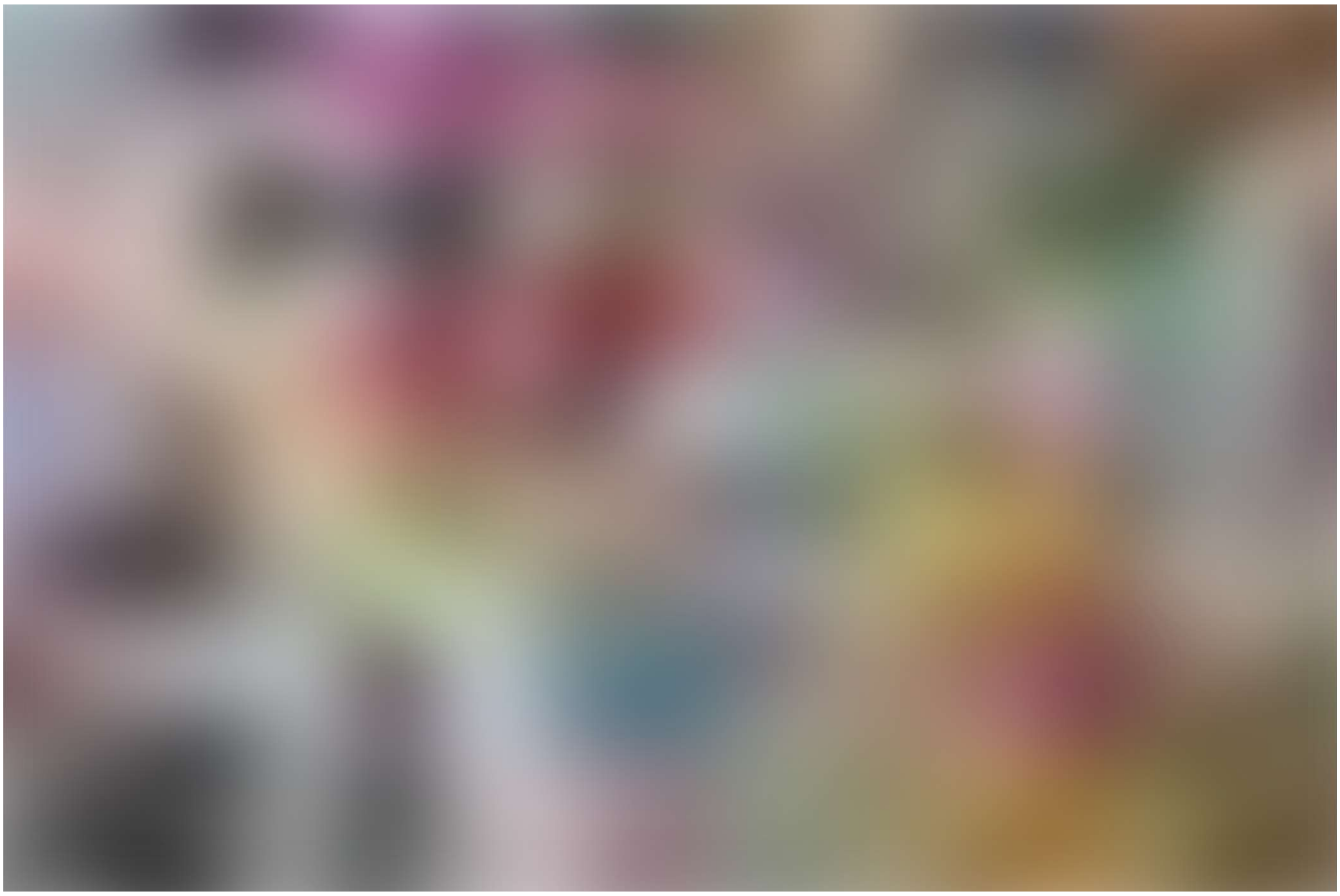


Photo by John Barkiple on Unsplash

At the beginning of any Data Science project it's important to get a clear understanding of your data, and the path it follows from source to dataset. Whilst it can be tempting to rush through towards implementing the sexier and more complex parts of your project, if it's built on shaky foundations, that effort will be wasted. Only by doing the grunt work of asking questions, testing assumptions, profiling and understanding your data, will you be truly confident in the quality of your analysis.

Data Science Data Quality Data Data Analysis Testing