The Journal of Open Source Software

# Cuallee: A Python package for data quality across multiple DataFrame APIs

**Herminio Vazquez** [1*] **and Virginie Grosboillot** [2*]

**1** Independent Researcher, Mexico **2** Swiss Federal Institute of Technology (ETH) * These authors contributed equally.

**DOI:** 10.xxxxxx/draft

**Software**

- Review
- Repository
- Archive

**Editor:** Open Journals

**Reviewers:**

- @openjournals

**Submitted:** 01 January 1970
**Published:** unpublished



## Summary

In today's world, where vast amounts of data are generated and collected daily, and where data heavily influence business, political, and societal decisions, it's crucial to evaluate the quality of the data used for analysis, decision-making, and reporting. This involves understanding how reliable and trustworthy the data are. To address this need, we've created `cuallee`, a Python package for assessing data quality. `cuallee` is designed to be dataframe-agnostic, offering an intuitive and user-friendly API for describing checks across the most popular dataframe implementations such as PySpark, Pandas, Snowpark, Polars, DuckDB and BigQuery. Currently, `cuallee` offers over 50 checks to help users evaluate the quality of their data.

## Statement of need

For data engineers and data scientists, maintaining a consistent workflow involves operating in hybrid environments, where they develop locally before transitioning data pipelines and analysis to cloud-based environments. Whilst working in local environments typically allows them to fit data sets in memory, moving workloads to cloud environments involve operating with full scale data that requires a different computing framework, i.e. distributed computing, parallelization, and horizontal scaling.

This shift in computing frameworks requires the adoption of testing strategies that can accommodate testing activities in both local and remote environments, without the need to rewrite test scenarios or employ different testing approaches for assessing various quality dimensions of the data.

An additional argument is related to the rapid evolution of the data ecosystem. Organizations and data teams are constantly seeking ways to improve, whether through cost-effective solutions or by integrating new capabilities into their data operations. However, this pursuit presents new challenges when migrating workloads from one technology to another. As information technology and data strategies become more resilient against vendor lock-ins, they turn to technologies that enable seamless operation across platforms, avoiding the chaos of fully re-implementing data products. In essence, no data testing strategy needs to be rewritten or reformulated due to platform changes.

A last argument is the need for such a quality tool, is the desire of moving quality procedures to the earliest phases of the data product development life-cycle. Whether in industry or academia, the reduction of time allocated for quality activities is unfortunately like a norm, due to the predominant focus on functional aspects. Enabling a declarative, intuitive and flexible programming interface to data quality, allows teams to embed quality into their development, adopting a mindset of building quality in, as opposed to testing quality out.

## Checks

(Binney & Tremaine, 2008) is the reference for the checks

## Controls

This are the controls

## References

Binney, J., & Tremaine, S. (2008). *Galactic Dynamics: Second Edition*. Princeton University Press. http://adsabs.harvard.edu/abs/2008gady.book.....B