

OpenQuestion: A survey building and reporting platform written in Python

Allan Campopiano¹ and Stu Cork²

1 Halton Catholic District School Board 2 Anvil

DOI: [10.21105/jose.0XXXX](https://doi.org/10.21105/jose.0XXXX)

Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Submitted: 01 January XXXX

Published: 01 January XXXX

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

Summary

[OpenQuestion](#) is a survey building and reporting platform for web-based data collection. Surveys can be developed using a GUI or programmatically by writing Python code. To report on survey results, OpenQuestion provides a Jupyter-like environment (Kluyver et al., 2016) for data visualization and document creation.¹

Statement of need

OpenQuestion's GUI and general functionality will be familiar to researchers who have used commercial survey platforms. However, OpenQuestion is unique in comparison since survey and report development can be controlled by Python² and JSON. The following sections describes a few of the reasons why OpenQuestion can be useful for data acquisition, analysis, and reporting.

Surveys and report designs are stored as a Python dict/JSON

Since the format of both surveys and report designs are JSON, researchers can easily share, archive, reformat, and “batch” create surveys and reports. For example, suppose a school board uses OpenQuestion to create a form for a student census. If another school board would like to use the same overall census design, they would simply need to request a copy of the underlying JSON representation of the survey. Once the JSON is uploaded to OpenQuestion, their web form would be fully operational, obviating the need for manual development using the GUI.

Analysis of survey data

OpenQuestion includes a reporting module that functions like a Jupyter-like environment in the sense that rich text and interactive visualizations can be arranged on a page to form a narrative based on the survey data. Specifically, OpenQuestion uses markdown widgets for rich text and chart widgets that use vega-lite (Satyanarayan et al., 2017), a high-level, interactive visualization library. While reports are natively “connected” to their

¹OpenQuestion is built with the open source Anvil App Server, a runtime engine for writing full-stack web applications in Python (Luff & Davies, 2020). Note that an Anvil account is not required to use OpenQuestion.

²Those without Python knowledge can simply use OpenQuestion's GUI as a free alternative to other commonly used survey platforms.

associated survey's dataset, OpenQuestion also allows additional datasets to be loaded into any report. This means that, unlike many other survey platforms, charting can occur across different data sets in the same report/environment.

Automatic chart generation and templating

OpenQuestion's reporting module is able to automatically generate interactive charts (as well as their related vega-lite JSON representations). Automatic chart generation works in the following way:

- A user selects one or more columns in their dataset using a GUI
- OpenQuestion attempts to match those columns to one or more templates that are stored internally. Templates are simply vega-lite JSON specifications that have placeholders for a certain number of fields with given data types
- If one or more compatible templates are found, they are populated with the appropriate data, and displayed in the report

In addition, once a user has a chart design that they would like to generally reuse (possibly with a different dataset), they can save the chart as a template. The next time automatic chart generation is used, this new template will be considered in the matching process. Automatic chart generation and templating is depicted in the figure below:

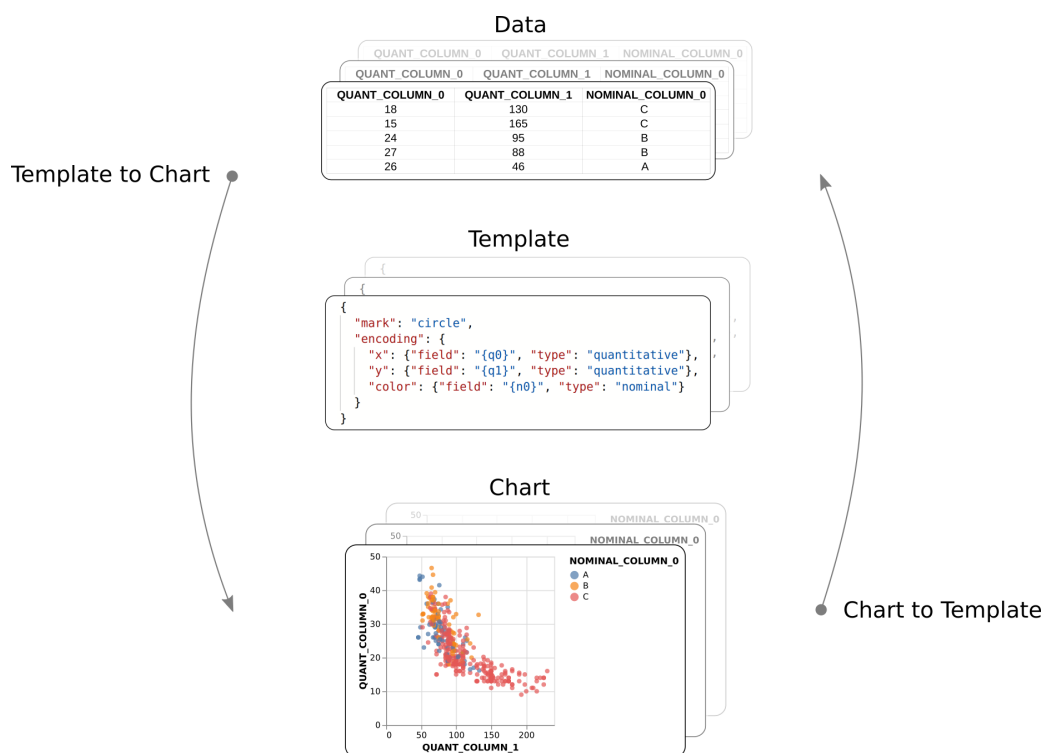


Figure 1: OpenQuestion's automatic chart creation and templating process. Users indicate the columns they would like to visualize. The number of selected columns and their data types are compared to underlying JSON templates. If a match is found, the compatible templates are populated with data and the resulting charts are displayed. To go the other direction, OpenQuestion also allows any chart to be converted to a template to be used in future automatic chart creation.

Interactive “code-free” HTML reports

OpenQuestion allows reports (which include interactive charts and rich text) to be exported to HTML files. All code blocks are removed and the interactivity of the charts is maintained. This is useful for preparing manuscripts that are to be shared with either a non-technical audience (since there are no code blocks), or with colleagues and/or supervisors who would prefer to only see the narrative aspect of the report (i.e., charts and text).

Comparison to other survey development platforms

OpenQuestion’s graphical interfaces (for developing and managing surveys), installation, and documentation are intended to create a low barrier to entry for researchers. This contrasts with platforms such as REDcap (Harris et al., 2009) (not open source; written in PHP) and LimeSurvey (LimeSurvey Project Team / Carsten Schmitz, 2012) (open source; mostly written in PHP) which have a potentially higher barrier to entry for many researchers (due to licensing, installation, complex interfaces, etc.). Although, it is also true that these platforms are more mature and therefore have more features.³

One notable Python-based survey platform is Docassemble (Pyle, 2017). In contrast to OpenQuestion, Docassemble focuses on document generation, and in terms of development, web forms are created by writing YAML, Markdown, and Python. OpenQuestion on the other hand, is designed for surveys that save submissions to a database and survey development can be accomplished using the GUI or by writing Python code. Docassemble is a mature project with a thriving community and should be considered by researchers looking to automate document generation via web-based forms.

Acknowledgements

The authors would like to thank James Desjardins, Stefon van Noordt, Meredydd Luff, Ian Davies, Shaun Taylor-Morgan Phil Colbert, Grant Bryer, Stephanie Spicer, Trevor Dixon, Lisa Collimore, Jennifer MacDonald, Zoe Walters, Whedon, the Journal of Open Source Software, the Halton Catholic District School Board, and the Anvil community for their support of this project.

References

- Harris, P. A., Taylor, R., Thielke, R., Payne, J., Gonzalez, N., & Conde, J. G. (2009). Research electronic data capture (REDCap)—a metadata-driven methodology and workflow process for providing translational research informatics support. *Journal of Biomedical Informatics*, 42(2), 377–381.
- Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., & Willing, C. (2016). *Jupyter notebooks – a publishing format for reproducible computational workflows* (F. Loizides & B. Schmidt, Eds.; pp. 87–90). IOS Press.
- LimeSurvey Project Team / Carsten Schmitz. (2012). *LimeSurvey: An open source survey tool*. LimeSurvey Project. <http://www.limesurvey.org>

³Note that neither REDcap nor LimeSurvey are written in Python. This is only to say that perhaps there is an argument for OpenQuestion being an easier platform to contribute code to for many researchers.

- Luff, M., & Davies, I. (2020). Anvil app server. In *GitHub repository*. GitHub. <https://github.com/anvil-works/anvil-runtime>
- Pyle, J. (2017). Docassemble. In *GitHub repository*. GitHub. <https://github.com/jhpyle/docassemble>
- Satyanarayan, A., Moritz, D., Wongsuphasawat, K., & Heer, J. (2017). Vega-lite: A grammar of interactive graphics. *IEEE Transactions on Visualization and Computer Graphics*, 23(1), 341–350. <https://doi.org/10.1109/TVCG.2016.2599030>