

Stratified sampling for content analysis: a Python-based solution

Brian Brennan, PhD, *Robert Morris University*, brjbrennan@gmail.com

Abstract

Manual stratified sampling in content analysis is time-intensive and difficult to reproduce. While stratified sampling produces more efficient estimates than simple random sampling, implementation remains predominantly manual, creating reproducibility challenges and substantial researcher burdens. This study develops and validates an automated stratified sampling tool using Python and configuration-driven design. The tool enables researchers to specify sampling parameters through a configuration file without modifying code, supporting adaptation across diverse content analysis contexts. Validation using a set of 1,888 news articles about U.S. Border Patrol coverage (2012-2026) stratified by outlet political bias and presidential administration demonstrates two key findings. First, reproducibility testing across 100 independent runs with identical seeds shows 100% agreement in item selection, confirming exact replicability. Second, implementation verification reveals zero duplicate items, correct stratum allocation, and valid temporal boundaries, demonstrating the elimination of common manual sampling errors. The tool is released as open-source software, extending computational methods to the sampling phase of content analysis research and addressing reproducibility challenges in communication and social science.

Keywords: stratified sampling, automated sampling, content analysis, research methods, Python

Introduction

Manual stratified sampling for content analysis and social science research is time-intensive, error prone, and difficult to reproduce. While content analysis is a commonly used research method in communication and social science research, the sampling process in these studies has been under examined. Sampling is mostly manual where the researcher filters select and verifies the even distribution of the dataset. This manual process may become vulnerable to human error and bias. As research techniques evolve, introduction of new technologies has started to appear. Recent research has explored artificial intelligence and machine learning for the collection and analysis of data for this type of social science research (Felaco et al., 2024). While these methods offer an exciting adaption of these popular technologies, the barrier to entry could be too high for researchers who are not familiar with these tools. These tools also may not sufficiently address the issue of reproducibility in this type of research which historically occurs at an extremely low rate (Open Science Collaboration, 2015).

To address this gap, this research presents and tests a novel approach to sampling for content analysis methodologies using a Python-based program. The program allows for a file of selected news reports to be loaded and randomly sampled directed by inputs into a configuration file. In this study, the program was used for news reports related to the United States southern border, but the flexibility of a configuration file allows for a number of different use cases. This paper makes the following contributions to the body of knowledge. First, the paper presents an open-source stratified sampling tool that is accessible to non-

technical researcher. Second, it validates the reproducibility of the tool through identical seed runs and Monte Carlo simulations. Finally, it provides an architecture that is flexible across content analysis domains by using a configuration file.

Literature Review

Stratified sampling produces results that are more efficient than simply random sampling when the population of content is diverse (Cochran, 1977). This stratified sampling process for content analysis research allows for the data to be split into subpopulations for analysis based on the relationship to the variables being observed (Neuendorf, 2017). Lohr (2022) details how stratified sampling can often produce samples that provide a lower variance for a population. This also can help prevent the researcher from collecting a sample that is not representative a population when the research design is more complex.

A common practice for stratified sampling is using the method to identify a constructed timeframe from a large set of data. A “constructed week or a “typical week” of print or television media allows for the creation of subgroups and prevents sampling of one specific day to overwhelm the sample (Riffe et al., 1996; Krippendorff, 2004). Social media content analysis research, however, may benefit from a simple random sampling due to the differences in the cycle of content creation. (Kim et al., 2018).

Content analysis sampling regularly rely on manual process of researchers but also has the opportunity to benefit from continued collaboration with computational disciplines for sample generation (Pavelko & Grabe, 2017). Reproducibility is a key variable in social science research and continues to be a challenge for researchers in fields with varying data sets between studies. This extends past sampling in the content analysis research process.

Trilling & Jonkman (2018) proposed four criteria that should be used when developing a framework to automate the content analysis process – scalability, open-source, adaptability and accessibility via multiple interfaces. The code used in computational research is key to reproducing studies with large datasets and has been argued that journals should make the code submission a requirement for publication (Peng, 2011) The challenge of reproducibility has been addressed through standardizing sampling methods, such as the STAMP method for systematic literature reviews (Rogge et al., 2024). Computational approaches to sampling and social science research as a whole have provided an opportunity to improve the collection and analysis processes in these research areas.

Sampling and collection models using artificial intelligence have started to be explored in social science research (Felaco et al., 2024), this could cause a high barrier of entry for researchers unfamiliar with these types of tools. This gap is addressed in this research by providing a more accessible solution that provides the same key attributes of consistent sampling and reproducibility. Current tools such as sampling and survey packages in R or scikit require a base knowledge in the programming language or platform. There is currently no tool designed for communication researchers with a limited background in programming.

Methodology

In order to test the efficacy of the Python based stratified sampling tool, a collection of news reports related to the United States southern border and border protection was compiled. This data set needed to be sampled using the strata of political leaning of the news outlet and presidential administration at the time of the report (2012-2026). While this topic was chosen as an example to test the sampling tool, this tool can be

adapted to any topic needing to be sampled in the same way. The goal of the collection process was to sample 60 reports for each presidential administration (20 from left-leaning sources, 20 from right-leaning sources, and 20 center sources) for a total of 240 news reports related to the United States southern border.

The collection of news reports started with a search of the ProQuest Newsstand database. This search initially resulted in over 30,000 results. These results were then filtered by source to those found only on the media bias chart. This chart was created by AdFontes Media and is considered a well-known resource in the study of the media bias (Tavares, 2023). The final result was a selection of 1,888 news reports. To move from the 1,888 reports down to the needed 240 for analysis, a tool needed to be developed to randomize and sample the selection based on the stratified sampling procedure that was used to evenly cover each presidential administration with outlets across the political spectrum.

A tool was developed in Python by using the pandas Python library. According to Gupta & Bagchi (2024), the pandas library allows Python to work directly with spreadsheets and provides multiple data analysis and manipulation tools. In order to run the sampling program, initial settings need to be entered into the configuration file that accompanies the program. These settings include the target sample size (the total number of news reports that need to be sampled), random seed (number set for reproducibility), the input file, and the output file. For this test, the total sample selected was 240 and the random seed was set at 2025.

The configuration file has customization options that include category groups, categories, time periods, and output options. For this process, the program used the category group “source” and the categories of “Left”, “Right”, and “Center”. News sources from the AdFontes media bias chart were entered into the configuration file under the “Left”, “Right”, and “Center” categories to set the source bias. Figure 1 shows the sampling workflow. When the program is run, the input and configuration files are loaded first.

After loading these files, the news reports in the input file are classified into the categories from the configuration file. This included source bias, time period of the news report, and what administration was in power during the time period. Once all 1,888 reports were classified, those that did not meet the category definitions were dropped. The remaining news reports were then divided evenly across a combination of source bias and administration categories then randomly sampled. For example, 20 reports are selected randomly for Obama – Right, 20 for Obama-Center, etc.). If the target size is reached, the news reports are saved to the output file identified in the configuration. If the target size is not reached, the selected reports are trimmed or added to ensure the target is reached.

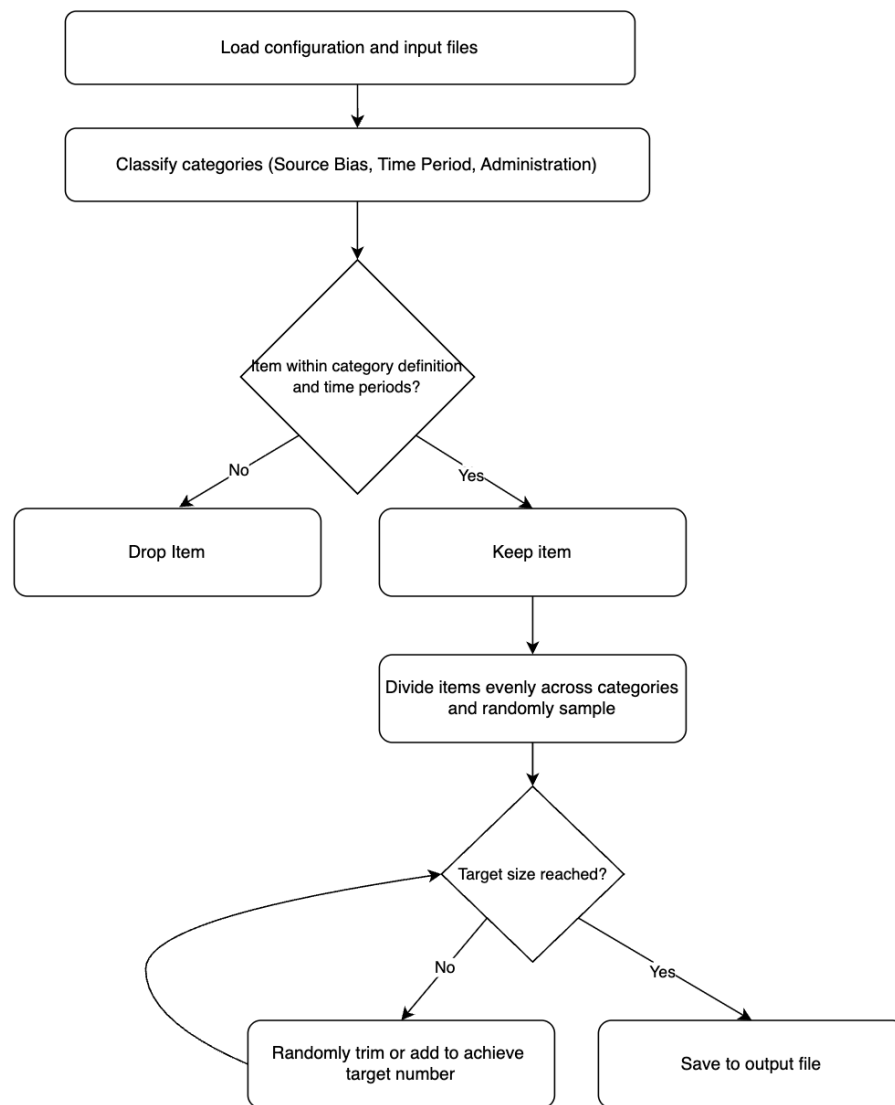


Figure 1. Flowchart of Program Decisions

Results

The results of the sampling test provided the expected target size for each category. The stratification of the sample can be seen in detail in Table 1. The sample tested was successfully stratified based on bias of the news sources in the input file and through each of the four presidential administrations tested. To ensure reproducibility, the script was run 100 times with the same configuration (random seed = 2025). All runs produced identical results with 100% agreement in the news reports selected. This demonstrates that this seeded randomization produces exact reproducibility.

Table 1. News Sample Categories, Category Groups and Stratification

Category Group	Category	n	%
Source Bias	Left	80	33.3%
	Center	80	33.3%
	Right	80	33.3%
Presidential Administration	Obama Administration (2012 – 2016)	60	25%
	Trump 1 Administration (2016 – 2020)	60	25%
	Biden Administration (2020 – 2024)	60	25%
	Trump 2 Administration (2024 – 2026)	60	25%
Publication Year	2012 - 2014	42	17.5%
	2015 - 2017	28	11.7%
	2018 - 2020	50	20.8%
	2021 - 2023	42	17.5%
	2024 - 2026	78	32.5%
Publication	Fox News	63	26.3%
	CNN	30	12.5%
	Wall Street Journal	21	8.8%
	USA Today	21	8.8%
	NPR	20	8.3%
	Los Angeles Times	18	7.5%
	Washington Post	15	6.3%
	Washington Examiner	13	5.4%
	CBS	11	4.6%
	Other Publications	28	11.6%

While this test proved the tool can reproduce the stratified sample from the same set of data, it does not answer how the tool compares to purely random sampling. To answer this, 1,000 Monte Carlo simulations were conducted using the results stratified sampling tool sampling 240 news reports and compared it to a true random sample of 240 news reports from the dataset. A comparison was then made on the distribution balance across all 1,000 iterations. The stratified sampling tool produced an expected 80 reports across each bias category (Left, Right, Center). The simulations conducted using a simple random sample produced a standard deviation of 51.76 news reports from the expected amount.

Table 2. Monte Carlo Simulation Results - Stratified vs. Random Sample

Metric	Stratified Sample	Random Sample	Difference
Target sample size per source category	80	80	0%
Standard Deviation	0	51.76	-51.76
Percent Deviation	0	64.7	-64.7
Range of sample per source category	80-80	25-156	

Discussion

Three key findings can be taken from the results of testing the tool against this dataset. First, the results of the testing show replicability of the stratification based on 100 independent tests with the same random seed. This shows 100% agreement with the selection of news reports on each run confirming that the tool is successful in showing reproducibility in the sample.

The second finding verifies the robustness of the tool. The results show zero duplication, allocation to the correct stratum and time periods based on the settings in the configuration file. Automating the sampling process prevents common manual sampling errors that can significantly impact the validity of a study or lengthen the time a researcher needs to collect a correct sample. The tool takes away any bias or judgement decisions made by the researcher that prevent exact replication even if completed by the same researcher. The Monte Carlo validation confirms the correct implementation of stratified sampling within the tool in over 1,000 simulations and a reduction of sampling error compared to simple random sampling.

The tool was initially developed to sample political news coverage but design to be generalizable across multiple research areas. Any large datasets for social science research can benefit from using this tool, specifically those with multiple stratification variables. Sampling a large corpus of documents would fit directly into the use case of this tool and the tool was designed to be flexible enough to modify the needed configuration file to align with the use case. By using a configuration file instead of hard coding variables, other researchers can use the tool with minimal customization or deep technical knowledge. This provides improvements over current stratification tools in Python (scikit-learn) or in R (sampling) as it relates to the core audience for this tool, the communications researcher.

This tool was built to be more accessible for those with limited or no programming experience. The current status of the tool does require that a user is able to install Python and run a program on the command line. Some comfort with Python is needed to effectively use to tool and benefit from it. Future improvements on this tool could add a simple web-based interface allowing for faster adoption. In this case, the user would be able to input configurations and variables directly into an interface that is much more user friendly.

Limitations

There are some limitations of this study that provide opportunities for future research. First, this tool is implemented for two stratification variables – Outlet Bias and Presidential Administration. If three or more variables were needed, the code would need to be refactored to account for the additional variable. The logic itself would remain however, but the change would require development work. Second, the tool is built to focus on a single set of news articles which limits generalizability outside this type of research.

Third, the equal allocation of the sample size is assumed in the current format of the program. If a different type of allocation is needed, modifications to the code would need to be made.

Future Studies

A future area of study could expand the use case to health communication, social media, or legal research areas. To continue the accessibility of the open-source code to non-technical researchers, an interactive front end that provides easy user input could be developed. As it stands currently, a user would need to be comfortable updating a configuration file and running a Python script. While this is an improvement over fully writing Python or R code, the usability could be improved. A simple webpage or app with fields for user input and file upload could improve the tool significantly.

Conclusions

Manual stratified sampling in content analysis is time-intensive, difficult to reproduce, and is prone to errors in implementation. This study developed and validated a stratified sampling tool that addresses these issues without the need for deep technical knowledge or intense customization of the underlying code. The tool was tested with 100 runs that produce 100% reproducibility without any duplication or errors in the sample. This work aligns with the increasing use of computational models in communications and social science research to bridge the gap between the benefits of automation and sound stratified sampling practices. The key area of improvement provided by this tool is accessibility to non-technical researchers. Applications, AI chatbots, and scripts are continuously being developed in academic research but the barrier for entry may turn off some users from their benefits. This tool is a foundation that can be improved with those types of researchers in mind and allow for more significant research to be done with a reduction in manual processes.

Data and Code Availability

The tool is released as open source to encourage adoption and future adaptation into other fields of study. The program and sample data for this research can be found here:
<https://doi.org/10.5281/zenodo.18970419>

References

- Cochran, W. G. (1977). *Sampling techniques* (3rd ed). Wiley.
- Felaco, C., Amato, F., & Aragona, B. (2024). Digital methods for social sciences. *Mathematical Population Studies*, 31(4), 237–241. <https://doi.org/10.1080/08898480.2024.2414653>
- Gupta, P., & Bagchi, A. (2024). Introduction to Pandas. In P. Gupta & A. Bagchi (Eds.), *Essentials of Python for artificial intelligence and machine learning* (pp. 161–196). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-43725-0_5

- Kim, H., Jang, S. M., Kim, S.-H., & Wan, A. (2018). Evaluating sampling methods for content analysis of Twitter data. *Social Media + Society*, 4(2). <https://doi.org/10.1177/2056305118772836>
- Krippendorff, K. (2004). *Content analysis: An introduction to its methodology* (2nd ed.) Sage.
- Lohr, S. L. (2022). *Sampling: Design and analysis* (3rd ed.). Chapman and Hall/CRC. <https://doi.org/10.1201/9780429298899>
- Neuendorf, K. A. (2017). Message units and sampling. In *The Content Analysis Guidebook* (pp. 70–95). SAGE Publications, Inc. <https://doi.org/10.4135/9781071802878>
- Open Science Collaboration. (2015). Estimating the reproducibility of psychological science. *Science*, 349(6251), aac4716. <https://doi.org/10.1126/science.aac4716>
- Pavelko, R. L., & Grabe, M. E. (2017). Sampling, content analysis. In *The international encyclopedia of communication research methods* (pp. 1–7). John Wiley & Sons. <https://doi.org/10.1002/9781118901731.iecrm0223>
- Peng, R. D. (2011). Reproducible research in computational science. *Science*, 334(6060), 1226–1227. <https://doi.org/10.1126/science.1213847>
- Riffe, D., Lacy, S., Nagovan, J., & Burkum, L. (1996). The effectiveness of simple and stratified random sampling in broadcast news content analysis. *Journalism & Mass Communication Quarterly*, 73(1), 159–168. <https://doi.org/10.1177/107769909607300114>
- R Rogge, A., Anter, L., Kunze, D., Pomsel, K., & Willenbrock, G. (2024). Standardized sampling for systematic literature reviews (STAMP method): Ensuring reproducibility and replicability. *Media and Communication*, 12, Article 7836. <https://doi.org/10.17645/mac.7836>
- Tavares, J. L. S. (2023). *Towards an automated media chart: Framing news articles with natural language processing techniques* [M.I.S., Universidade do Porto (Portugal)]. <https://www.proquest.com/docview/3059392369/abstract/C7FFF6FBAF1F436BPQ/1>
- Trilling, D., & Jonkman, J. G. F. (2018). Scaling up content analysis. *Communication Methods and Measures*, 12(2–3), 158–174. <https://doi.org/10.1080/19312458.2018.1447655>