HARVARD DATA SCIENCE REVIEW

Volume 0 Issue 0
DOI: 00.000
ISSN: 2644-2353

Using Containers to Validate Research on Confidential Data at Scale

Lars Vilhuber[†,*]
[†] Cornell University

Abstract. I describe past experience with the validation server process over 10 years and several hundred users, as a means to provide proxy access to confidential data. As a modern replacement, I propose the use of containers — simulated computers that encapsulate entire software and file structures. The use of containers ensures reproducibility, reliable portability, and enables scalability. Infrastructure can be outsourced to commercial providers or users, at little to no cost to data providers. The only likely limitation to full automation is the absence of automated output vetting algorithms at statistical agencies.

Keywords: synthetic data; verification server; confidential data; reproducibility, containers

Media Summary

Vilhuber describes how researchers have used synthetic data as a proxy access to confidential data in the past. Such systems have been long and expensive to develop, because trying to satisfy many different constraints on their suitability imposed by users. Instead, he suggests that modern containers — simulations of entire computer and software systems — could be leveraged as a portable, scalable, and cheaper replacement system. By packaging lower quality synthetic data in prescribed environments, and allowing users and commercial providers to support their use, then providing rapid turnaround on validation against the actual confidential data, monetary and time costs of the system for both providers and end users can be substantially reduced.

Introduction

Concerns about confidentiality in statistical products have increased in the past several years. While the new disclosure avoidance techniques introduced for the 2020 United States Decennial Census (Abowd et al., 2022) garnered much attention, the academic community also expressed concerns about agency plans to apply formal disclosure avoidance techniques to public-use microdata files (PUMFs), and after an initial announcement, the Census Bureau delayed implementation of such methods for the American Community Survey (ACS) (Daily, 2022).

Part of the concern is that the application of confidentiality methods may prevent inferences that were feasibly made in the absence of those confidentiality procedures. Some of those concerns

[*] lars.vilhuber@cornell.edu

may be solved by adopting analysis methods that are congenial (Meng, 1994) to the disclosure avoidance methods, though that is not a recent problem (Abowd & Schmutte, 2015). However, statistical agencies often attempt to provide "one size fits all" datasets for general purpose usage, as in the case of the ACS, with the goal of usability by a wide spectrum of models. One novel approach, at least at the time, was the provision of synthetic data that preserved most of the inferential validity of the underlying confidential data (Little, 1993; Raghunathan, 2021; Reiter, 2023; Rubin, 1993). However, producing such general purpose synthetic data that are reasonably non-disclosive has proven difficult and long.

This article will set out to combine various lessons learned from both the literature and past experiments in a variety of domains to propose an alternative or complementary approach to providing access to confidential data via synthetic data. I draw on lessons learned from the analysis of several thousand replication package in economics (Vilhuber & Cavanagh, 2025), from past efforts to create custom ('bespoke') synthetic data (Nowok et al., 2017), from current technology that allows for reliably reproducible analysis (**containers**, see Boettiger, 2015, for an introduction), and the feedback of hundreds of potential users of synthetic data in a project I co-lead until 2022 (Vilhuber & Abowd, 2022). The proposed mechanism will allow to relax some of the analytical validity of the synthetic data, but in exchange allow for rapid turnaround time for analyses, something that researchers value. A maintained assumption, informed by observing similar usages and from our own experience, is that the cost of such a system is dramatically lower, and can thus be feasibly implemented by budget-constrained federal agencies.

I will start by defining what container technology is, what particular definition of synthetic data I have in mind, and what validation and verification servers are in the social science space. I will then lay out how the fact that containers can create portable and reproducible environments can be leveraged to simplify the validation of code, allowing researchers to develop models on synthetic data, and validate on confidential data, at scale, while maintaining the confidentiality promises that custodians give to their respondents. I then discuss a concrete but to this day theoretical example, and consider advantages and (key) limitations to the mechanism. The combination of containers with a relaxation of the quality of synthetic data can allow data custodians — survey institutes or statistical agencies — to scale up access by researchers at substantially lower cost than currently implemented. In concluding, I discuss some possible extensions, including how to embed stronger privacy-protection methods such as differential privacy into both analysis and release of results.[1]

1. Concepts and Definitions

1.1. **Containers.** Containers, often referred to using the name of a particular implementation by a commercial provider (Docker), are technology which enables computer processes and code libraries to be bundled and constrained. In essence, a container is a bundle of all the dependencies and code an app, or a researcher's statistical analysis, needs, into a single compact bundle or even a single file, which can then be run on any computer without (much) further ado. They can contain the data necessary for such computations, though there is also the ability to inject components — such as researcher-specific code or data — at the time a more general-purpose container is run. Containers can be hosted on a cloud platform, but can also run individually on researcher compute

[1] At the time of this writing, at least one other project, much more ambitious and complex, has been under development for a few years, and is testing a "privacy-preserving validation server" (Burman et al., 2018; Tyagi et al., 2024). I will briefly contrast that project to the present proposal in the discussion.

platforms (personal computers). They have been primarily developed to allow for lightweight but massive scaling of internet applications, which requires the same process be re-executed in a reliable fashion, i.e., in a reproducible fashion. This latter aspect is what has made it attractive for academic researchers as well, and containers have been put forward as a mechanism for reproducible research for nearly a decade at this point (Boettiger, 2015), with various commercial and academic services arising around the use of containers (Brinckman et al., 2018; Chard et al., 2020; Clyburne-Sherin et al., 2019). Alternatives to Docker containers are Apptainer (Contributors to the Apptainer project, 2025), various Docker-compatible Linux implementations (for instance, "Podman", n.d.), or even BSD "jails" (The FreeBSD Project, 2025).[2]

Container usage in the context of confidential data in the social sciences or in statistical agencies is sparse, to the best of my knowledge. Containers themselves are not inherently more or less secure, and need to be managed appropriately to ensure that complete privacy and security are not compromised (see Souppaya et al., 2017), though that is similar to the fundamental IT security considerations all hosts of confidential data must implement.[3] In this article, I do not address container security per se, and in fact, as I will outline in an example later, container usage in the confidential environment is not necessary, albeit possibly convenient.

1.2. Synthetic Data.

The idea for what is now called "synthetic data" was laid out more than three decades ago by Rubin (1993): Using models trained on the confidential data, (multiply) impute some or all the values for a simulated population, and release to the public. The resulting file does not comprise any single person's "real" data, but is expected to be statistically similar to the unreleased confidential data. Since the original proposal, synthetic data in the context of statistical agencies and survey organizations has been much discussed. The primary concerns are two: how protective are synthetic data, and how fit for purpose are they? When statistical agencies aim to use synthetic data as a replacement or enhancement for other methods of disclosure avoidance, both issues are important. However, this article will not contribute to clarifying that, and the interested reader is pointed at Drechsler (2021), Drechsler and Haensch (2024), Raghunathan (2021), and Reiter (2023) and in particular, Raghunathan and Chaney (2023) and Raghunathan and Hotz (2024). I note that synthetic data is also often discussed in the context of machine learning and the training of AI models, often with a different focus than that of statistical analysis. Rather, in that context, synthetic data is used to help develop algorithms, sometimes by explicitly not making the synthetic data representative, for instance to work around inherent biases. While not the primary focus, I will refer back to those domains in one particular aspect that makes synthetic data there useful, namely for the testing of code.

1.3. Verification servers, and improving access to confidential data.

Because of the novelty (in this context) of synthetic data, creators of such data first needed to convince privacy custodians that synthetic data was protective of the underlying respondents. They then need to convince users of the data that the data have analytic validity, and can serve as valid and possibly universal

[2] Containers are often, but not exclusively, associated with Linux. There are container implementations for computers running Windows and macOS, typically running within a virtual machine (a simulated computer) running Linux. It is technically feasible to run a Windows container. Virtual machines are a more complex method of simulating an entire computer, but introduces more complexity in terms of the transparency, a key feature in this article.

[3] Considerations include compliance with General Data Protection Regulation (GDPR) or Health Insurance Portability and Accountability Act (HIPAA).

replacements to more traditional public-use data (Reiter et al., 2009).[4] In this context, the term **verification server** appears to be used when the answer provided to users is a quality measure of some kind, and the term **validation server** when answers derived from the actual confidential data (and possibly modified for confidentiality purposes) are provided to the user (and the public). The latter is more akin to a **remote submission server**, except that the data used to prepare submissions is meant to actually replace the need for submissions, at least in the long-run.

Long-running pilot projects with (non-formal) synthetic microdata products (SynLBD (Kinney et al., 2011; U.S. Census Bureau, 2011; Vilhuber, 2013), SIPP Synthetic Beta (Benedetto et al., 2013; Reeder et al., 2018; U.S. Census Bureau, 2015a)) came to an end in 2022 (Vilhuber & Abowd, 2022). These pilot projects made use of a publicly accessible analysis server (Synthetic Data Server (SDS)) housing synthetic data, authorized for release in this environment by the Census Bureau and the IRS (e.g. U.S. Census Bureau, 2015b), combined with a mechanism to re-run the analysis using confidential data, housed in the confidential computing environments of the U.S. Census Bureau.[5] Researchers obtained (fast) access to the SDS, where they interactively developed their code, using the releasable synthetic data, in a desktop-centric environment specifically designed to mimic the Census Bureau's environment.[6] Researchers were free to develop any code in any kind of structure, but were not allowed to remove the data from the SDS. Analysis code that was to be validated was restricted to the software available at the Census Bureau. Requests to validate the results obtained from prepared code were submitted via email to designated Census Bureau staff, who then manually ran the code on Census Bureau servers, often times debugging it, and handled disclosure avoidance analysis at the Census Bureau on behalf of the researchers. The modeling and analysis code provided by the researchers served to improve subsequent versions of the synthetic data. Figure 1 illustrates the process flow.

These pilot projects had two key goals, and a secondary benefit. The first goal was to develop new, analytically valid public-use data products based on confidential data that could be released to the public, much like the current public-use data products (such as the Current Population Survey, CPS, and the aforementioned ACS), replacing the need for researchers to access confidential data directly. The second goal, in support of the first one, was to allow researchers to contribute to the development of such products, and iteratively improve the data products. The secondary benefit accrued to the participating researchers: As a quid-pro-quo for contributing to the improvement of the data product, they obtained faster but indirect access to the results of analyses run on the confidential data.

[4] This problem, of course, is not unique to synthetic data and applies to any modification applied to released data.

[5] The SDS server was funded in part by NSF grants SES-1042181 and SES-1131848 as well as Alfred P. Sloan Foundation Grant G-2015-13903. The last of the funding through these grants ended in 2018, and availability after that time was funded through John Abowd's Edmund Ezra Day chair at Cornell University, on a shoestring budget.

[6] Technically, the Cornell-based server ran an NX remote Linux desktop. Shell access or remote submission were not allowed.

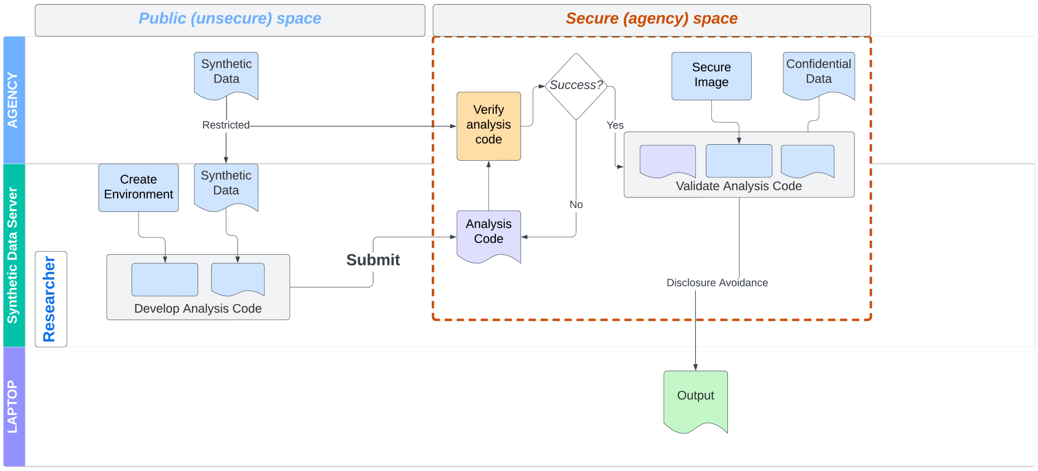


Figure 1. Process flow of validation using the Synthetic Data Server

2. Lessons learned from the SDS mechanism

Several lessons emerged from the SDS mechanism. While many researchers used the data to write papers,[7] and even organized conference sessions specifically around the use of the data,[8] even more researchers only "tried out" the data. While over 100 researchers were granted access to the server to access the SIPP Synthetic Beta (SSB) in the first five years of its availability (Figure 2), far fewer published using the SSB data. Almost none of the published articles actually used the results produced using the synthetic data. Comparison of parameters obtained from synthetic data and from confidential data using confidence interval overlap, a measure of congruence between the synthetic data and the confidential data introduced by Karr et al. (2006), was very heterogeneous even for a given dataset across and within projects (Table 1).[9] While the results in Table 1 do not take into account a distinction between key and nuisance parameters (simply comparing all parameters estimated in researcher-provided models), authors may have simply been very hesitant to use the parameters estimated on the synthetic data.

[7] All publications directly funded by the supporting NSF grant, or using the NSF-funded server, are listed at https://www.zotero.org/groups/5595570/sds-nsf-1042181/library. Some publications were prepared by NSF-funded project personnel and should not be directly included in a publication count of "users." Most publications were included in this list after a bibliographic full-text search for the grant identifiers. Some researchers may not have reported the published article to the project team, or mentioned the support of the grant to the server they used in their acknowledgements.

[8] LERA session "Data Gold! Exploiting the Rich Research Potential of Lifetime Administrative Earnings Data Linked to the Census Bureau's Household SIPP Survey", at the Allied Social Sciences 2016 Annual Meeting (American Economic Association, 2016).

[9] Due to technical issues, the selection of projects is a convenience sample. The analysis was run ex-post, without the collaboration of the original authors, by injecting code into the original authors' analysis code. This often failed, especially for more complex code. The results using the confidential data were obtained by Census Bureau staff, and released to the author of this article after disclosure avoidance, several years ago.

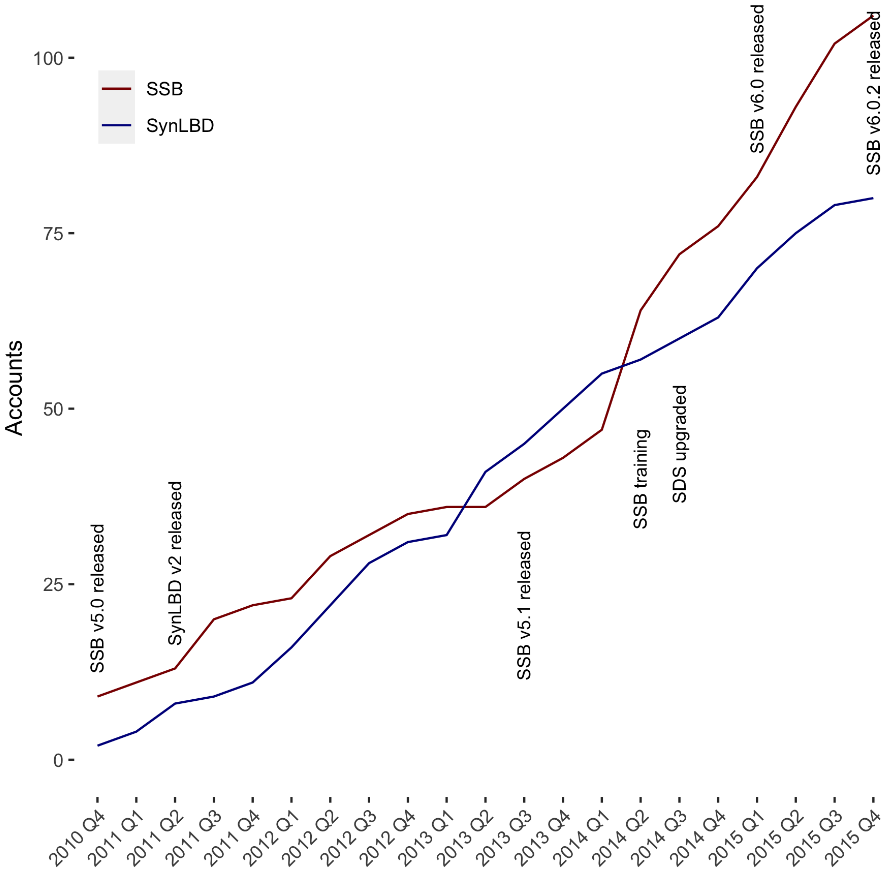


Figure 2. Computer accounts on the SDS over time

Thus, a core goal of the synthetic data — to replace the confidential data in researchers' analyses — was not being met, even when the synthetic data actually w a very good stand-in (note the SynLBD project B with a mean confidence interval overlap of 87%). Nevertheless, the synthetic data were complex enough to allow for development of models without access to the confidential data, what I would call "good enough data."

Anecdotal evidence from both my own and Census staff's attempts to use author-provided computer code to run the analysis on the confidential data demonstrated challenges in reproducibility. Authors might hard-code intermediate findings, rather than letting the data drive the analysis, and would otherwise not fully leverage the similarity between the two computing environments. These lead to time-intensive human debugging, or multiple rounds with authors, neither of which are an efficient and satisfying process.

More interestingly, multiple authors treated the synthetic data access as a gateway process for access to the confidential data. Knowing that the synthetic data did not contain all the features they needed for their analysis, but having to wait for permission to access the more detailed confidential data in the Federal Statistical Research Data Centers (FSRDCs), authors used the synthetic data

User	Request	Mean	75th	90th	Max	Dataset
A	1	0.16	0.25	0.72	0.89	SynLBD
A	2	0.10	0.00	0.52	0.92	SynLBD
B	1	0.87	1.00	1.00	1.00	SynLBD
C	1	0.22	0.51	0.72	0.99	SynLBD
D	1	0.49	0.79	0.87	0.98	SSB
E	1	0.39	0.56	0.63	0.94	SSB

Table 1. Distribution of Parameter-specific Confidence Interval Overlap, for selected projects

The table presents the distribution of overlap in parameter-specific confidence intervals (Karr et al., 2006) across a small number of projects (for selection, see text). The overlap is quantified as the percentage of the length of the smaller interval that is contained within the larger interval, where the pairs of intervals are defined by running the same code on synthetic and confidential data. Each row shows mean and selected percentiles from the distribution of such comparisons across all the parameters of the specific model.

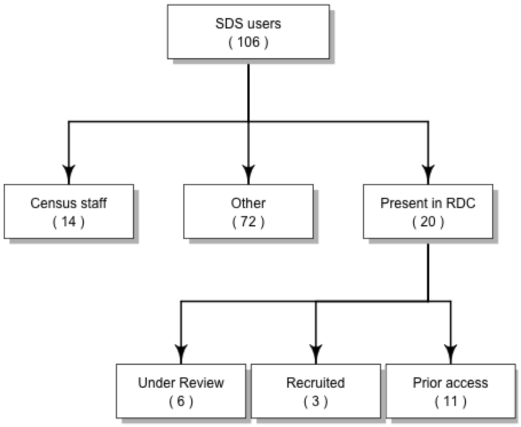


Figure 3. SDS users and access to FSRDC

to prepare analyses and explore the data. Figure 3 shows an early analysis of the first 106 users of the SDS, and subsequent usage of the FSRDC.

Importantly, in the initial phase of the projects, turnaround (submission of validation request and receipt of validated and privacy-protected results) was quite fast - single-digit weeks, rather than the multi-month process of obtaining access to the FSRDC. However, the introduction of new disclosure avoidance procedures at the Census Bureau, and the lack of integration of those procedures into the validation process, greatly increased the time lag in the second half of the projects.

3. Scaling up access to confidential data

If data cannot be made available due to intractable disclosure avoidance issues, yet access should be broadened, what can agencies do? The pilot projects described earlier were not set up to scale, and yet I argue that they demonstrated a need for such a process. The number of researchers gaining access grew continuously (see Figure 2). Special sessions at conferences were organized around the data accessed in this fashion. In general, based on my conversations with various researchers at conferences and by email, researchers were happy with the ability to access such data without having to request a full-blown project in an FSRDC, but were somewhat frustrated by the process.

Statistical agencies and research institutes have explored various ways to scale up access to confidential data. To cite a few examples, Statistics Canada has the Real-time Remote Access (RTRA) process, Norway has the Microdata.no system, the Bank of Portugal uses a two-stage system combining a remote desktop and validation (Guimarães, 2023), and Barrientos et al. (2018) proposed a differentially private verification server.

Many such processes have limitations that limit their utility for researchers. The aforementioned Statistics Canada and microdata.no systems strongly limit the type of analyses that are feasible by restricting the software keywords that can be used (RTRA), by creating a structured new statistical language (microdata.no), or by limiting the types of analysis that can be run and validated (Barrientos et al., 2018). Users of the Bank of Portugal's system still need to use the remote desktop system, similar to the SDS outlined before, because the data hosted there is not authorized as a full public-use product.

The issue is compounded by well-documented problems with the reproducibility of code in the social sciences. Heuristically, many of the problems with the SDS arose because the code failed to reproduce during validation, even though it was run in a very similar environment to the development environment. Researchers in the social sciences appear to rely heavily on interactive computing, with code produced subsequently failing simple reproducibility tests.[10] Examples abound, and can often be gleaned from the fact that many of the reproducibility meta-studies only succeed in reproducing a small number of studies due to limitations in personnel time.[11] In a sample of over 8,000 replication packages associated with high-profile economics articles, only 30% had some sort

[10] I define "interactive computing" as any sequence of computational codes that must be explicitly — through edits — adapted to the environment it is running in, and/or does not have a streamlined workflow that can be triggered from a single file, regardless of workflow technology. Examples of workflow management "systems" range from `make` (1976) (Association for Computing Machinery, 2003) to `Snakemake` (Mölder et al., 2021), to literate programming tools such as `Sweave` (Leisch, 2002) and Quarto (Allaire et al., 2024), to simple concatenated calls to software (canonical `run.sh`).

[11] Stockemer et al. (2018) note "lack of organization in code and data presentation was the main reason that we were unable to replicate some results" in political science articles. Stodden et al. (2018) notes that only 32% of packages in Science required only minor or no effort to reproduce, though some of the reasons listed are unavoidable (GPU setup, custom hardware) and not related to workflow issues. For the economics journal studied in Herbert et al. (2024), only 24% required no change to the code in order to be able to run, even when the ultimate results was fully reproduced.

of master script.[12] This is also my impression from our own efforts at the LDI Replication Lab supporting the AEA Data and Code Availability Policy, though we no longer make a systematic effort to categorize this. In part to cater to this need, remote-access or local secure access in the form of physical or virtual secure data enclaves are still the dominant - but expensive - way to access confidential data. The dominant method of access thus forces researchers to choose between lower quality data in an environment that corresponds to their preferred computing method, and higher quality confidential data in environments that are expensive for researchers, data providers, or both.

4. Desiderata

Drawing on the experience from the SDS pilot projects and other remote access methods used in the past, as well as looking at newer technologies that have emerged in the last decade, I suggest that a new, scalable mechanism to provide access to confidential data should have the following desirable characteristics:

[D1] the mechanism must support arbitrary modeling approaches and ideally a large number of programming languages

[D2] the mechanism must allow for development of models by researchers that are close to their "normal" method of developing models

[D3] the mechanism must be low-cost for the data provider, scaling at best sub-linearly with the number of users of those datasets

[D4] the mechanism must be low-cost for the data user, imposing at best marginal costs on their existing research infrastructure (software, computers)

[D5] the privacy-protected data provided as part of such mechanisms must be good enough to allow for complex modeling

[D6] validation, if necessary, must be fast - on the order of hours

Note that public-use data, as historically provided by statistical agencies, satisfies all of these criteria, except for the last one. Should statistical agencies actually offer validation even for such public use data, as Reiter et al. (2009) have argued? Traditionally, they do not, and leave it up to individual researchers to "self-validate" by requesting access to confidential data in a time-consuming fashion.[13]

5. A Proposal using Containers

I describe a simple scenario that satisfies most of the desiderata, using containers. Containers can easily be validated for reproducibility before they, or the code they used, are then forwarded to the confidential computing environment. Crucially, all validation of reproducibility can be performed prior to validation using the confidential data, on open, possibly commercial platforms. Only once reproducibility is confirmed is the same analysis model ported to the confidential data. Once determined to be reproducible, the analysis is extended to use confidential data, and could use a wide spectrum of plug-in disclosure avoidance measures as well. The concept here is similar in

[12] Code run in November 2023, searching for any filename that contained the strings 'main' or 'master', the most common name used for control code in economics. I am not aware of a similarly comprehensive collection for other social sciences.

[13] See Armour et al. (2016) for one example of such a project, affecting the widely-used Current Population Survey.

principle to a “Common Task Framework” (Liberman, 2010, 2014; Liu & Salganik, 2019), but applied to the access problem around confidential data, not to a validation or holdout dataset (Donoho, 2024; Liu & Salganik, 2019).[14]

The use of cloud providers removes the requirement for users of the synthetic data to install anything locally. The open-source nature of the container technology, on the other hand, allows users to do so, when they want to, or when they have to. The use of containers enforces reproducibility out-of-box when using synthetic data. Furthermore, containers enable scalability.

Finally, the combination of assured reproducibility and easily scalable computation allow for streamlined validation against the confidential data, which is in essence a replication of the analysis on the synthetic data. The streamlining, in turn, allows for low-cost rapid turnaround, quickly providing feedback and possibly results to researchers who otherwise do not have access to the confidential data. And by doing so quickly, the need to have complex and fully analytically valid synthetic data in the first place may no longer be binding. This, in turn, allows data providers to reduce the amount of time and effort devoted to developing complex synthetic data, instead focusing on creating plausibly complex, possibly custom-generated, low fidelity synthetic data, at low or no cost to disclosure avoidance budgets.

The use of containers in this way is novel as a systematic way to provide scalable, potentially high-throughput validation, and differs in usage from previous methods, such as the Cornell Synthetic Data Server. Containers have been used in a small number of well-published instances in the economics literature for precisely this kind of purpose, and are well-understood in the computer science and statistics community (Boettiger, 2015; Moreau et al., 2023). Nevertheless, acceptance in the social science community is not great, so far. In the same 8,000 replication packages in economics mentioned earlier, only 0.13% had used containers. While hard evidence in other social sciences is difficult to come by, none of the studies that analyzed reproducibility in the various social science disciplines mention the use of containers as the primary tool to ensure reproducibility in any of the packages they have analyzed.

5.1. **Details.** Reproducibility is important for synthetic data products when there is a validation or verification process involved. In such a setting, data users will first use the synthetic data to build and test their code for a desired analysis. Once the user is satisfied with the code used to perform their analysis, they can request validation or verification, and their code will be run by the data provider on the confidential data. Output from the analysis on the confidential data will then have to satisfy the data provider’s disclosure avoidance procedures before it can be released to the user. In the case of validation, the results from the second run on confidential data are released to the user, possibly confidentialized (U.S. Census Bureau, 2024). In the case of a verification server, the user only receives a (non-disclosive) message indicating whether her results from the first run can be considered to be inferentially valid or not (Barrientos et al., 2018).

One problem that can arise when maintaining a validation process is that the computing environment on the data user’s end does not exactly match the computing environment for the internal validation. This can lead to validation attempts that fail to run correctly or at all, which increases the wait time for data users and the staff time involved in performing the validation at the data provider. In order for this process to run smoothly, the data user’s code needs to be reproducible. This ensures that the user’s code can be easily transferred and run on the confidential data. This

[14] I thank an anonymous referee for having pointed out this analogy.

can be achieved by using a "container." A container collects all of the necessary libraries, dependencies, and code needed to run an analysis in a single package. This container can then be downloaded to any machine and used with the local system to run the application. Crucially, container systems usually have the capability to either rebuild containers in a trusted environment, or to provide to users pre-configured secure containers that are also authorized to run in the secure area hosting the confidential data. In most cases, the core container itself does not need to be transferred, only a comprehensive, easily parsable recipe to build such a container from known sources, and the user's model-specific code. Both the recipe (called a 'Dockerfile') and the user-provided model code, are plain-text, and adhere to known language specifications, which allows for easier security vetting of the code.

6. A detailed use case when using synthetic data

In this section, we walk through the process for a specific use case. The Census Bureau, data owner (custodian) of the confidential SIPP file merged with administrative data known as the SIPP Gold Standard file, makes a synthetic version of the Gold Standard file available as the SSB (Benedetto et al., 2013). The container host in the public domain in this case is Codeocean, which provides cloud-based access to Docker-based containers (called "compute capsules").[15] We use the example of Codeocean here, because they provide a relatively user-friendly interface for development of code, while retaining the requirement to have a hands-off, non-interactive run as a pre-requisite for publication. Crucially, Codeocean provides access to Stata and MATLAB compute capsules, covering 95% of economists' software needs. Users log in via a simple web browser, with no install required. A Codeocean capsule can, however, also be used by researchers on their own workstation, as long as they have Docker installed (or a compatible container runtime) and have a Stata or MATLAB license. Because of the standards-based approach, it is also straightforward to exchange the configured Codeocean-generated "base image" for a security-vetted base image within the confidential environment, something we return to later.[16] Finally, users who wish to not use such services can also typically provide their own setup for the synthetic data component, at very little additional cost or effort. Many university computing system provide some support on their high-performance computing clusters. For data providers, the tools used (containers) are widely used by numerous cloud providers, are transparent in how they are built, and allow for in-depth security scanning while retaining much of the flexibility that researchers and IT providers seek.

In this configuration, the Census Bureau could allow Codeocean to house both the synthetic dataset as well as setup and configuration code that will assist the data user in creating code that is reproducible. The data user can then build on the setup and configuration code provided by the Census Bureau to perform their desired analysis. To perform the analysis on the synthetic data, users may either run the code directly on Codeocean by paying for access to computing resources[17]

[15] There are many commercial providers able to run Docker containers.

[16] The basic functionality can be achieved on any container system. Other services in this space include or have included Wholetale (Brinckman et al., 2018; Chard et al., 2020), Onyxia, and many others. An earlier version of this article mentioned Gigantum, and linked to the website of Wholetale. As is not unusual in this space, Gigantum no longer functions as a company, and Wholetale, as an academic, externally funded pilot project, is no longer available. However, with the right setup, the open-source specification of most current container technology do not depend on any single provider.

[17] At the time of writing, Codeocean provides a limited amount of storage space and computing time for free to academic users.

or they may download the compute capsule, including the synthetic data, to their local machine and use their own computing resources. When the user requests validation, they provide or point to the compute capsule (containing code and any external data). The Census Bureau can then securely rebuild the container (if necessary), and make minimal (automated) changes to execute the analysis in a secure computing environment with the confidential data. By ensuring that everything can run within the Codeocean compute capsule using the synthetic data, it also makes it very likely that everything can be run within the associated compute capsule's container using the confidential data, even when the confidential data is not housed on Codeocean or any publicly accessible service. In addition to providing a location for the Census Bureau and data users to share access to the synthetic dataset and code, this mechanism also greatly increases the probability that the analysis will run correctly for both the data user and on the confidential data.

6.1. **Development of code on open compute servers using synthetic data.** Basic structure of Codeocean's existing validation process[18] imposes a strict separation between code, immutable data, and reproducible results:[19]

- all code is under `/code`
- all data are under `/data`
- all outputs MUST be written to `/results`, otherwise they are lost.

Outputs are regenerated at each run, and the history of such runs can be found (for the developing user) in the right pane. When the user, at the end of the development process, requests publication of the compute capsule, only the last run is published, together with code and data.

To facilitate this file organization for the user, template code, provided by the agency, can include a configuration file, which applies best programming practices by defining global variables for file paths that are used elsewhere in the code, and which can be modified by runtime environment variables. For instance, the default file path to the data location may be supplied by the operating system in the confidential environment, switching where any user code will look for data. On the open compute server, the same configuration file would seamlessly point to the synthetic data. The template code also instantiates logfiles, which are also written to `/results`, which allow for verification of what happens during runtime.

Stata, R, and many other programming languages use external packages of code (libraries) to augment native capabilities. Initially, these need to be installed over the internet. However, there are various ways to address the issue at subsequent installations: (1) Packages can always be re-installed from source at runtime; (2) packages can be installed by a programming-language specific install script at (first) runtime, and stored locally; and (3) packages can be incorporated into the container image itself (package installation is included in the recipe.

Codeocean has the ability to do the third method, via a "post-install" script that is run when the environment is setup (the project-specific container image is built). This embeds any such libraries into the container image, rendering subsequent internet access unnecessary for the software to run. However, this particular environment also supports the two other methods. This could lead to

[18] See https://help.Codeocean.com/getting-started/uploading-code-and-data/paths

[19] While there are various standards on the publication of research code, this is not meant to be yet another one. This is purely a requirement of the validation process, as implemented by this provider. The data custodian would need to define what their required structure is, which may or may not adhere to a specific code publication standard.

failure in a confidential computing environment that does not allow for internet access, and will need to be specified to users.[20]

I note that while hosted on Codeocean, the same image, once built, is re-used, and packages installed using the third method are not re-installed. However, when exporting a capsule, only the build script is exported, and a replicator would need to rebuild the container, thereby also re-installing any packages. This can lead to version discrepancies when packages cannot be pinned to a particular version (as is the case with Stata).

6.2. **Validating reproducibility.** The Codeocean interface signals to researchers the successful completion of the controller script (called `run`) in the right pane of the user interface. To the custodian of the confidential data, this indicates that the code is verified to execute without error on the synthetic data. The results produced by this specific run of the controller script, and not any previous interactive run, are the results provided in the "results" pane. It is important to highlight here that the Codeocean implementation preserves no outputs from prior runs in the "results" pane and folder, only from the latest run. It is thus not possible for researchers to create output in a prior run, then to comment out code that would then not be reproduced in a subsequent run in the confidential environment. Only fully automated complete runs produce all the expected output. This is important for scalability and efficiency, as it reduces the need for extensive debugging, on the researcher side, and allows for rapid assessment of basic reproducibility by the data custodian.

6.3. **Estimating Economic Returns to Education in the SIPP.** I now provide a concrete, albeit only theoretically possible, example. In this example, code to estimate a standard Mincer earnings equation (Heckman et al., 2003; Mincer, 1984) on the SIPP Synthetic Beta data (U.S. Census Bureau, 2024). The code as provided here is in Stata, the dominant statistical programming language for economists (Vilhuber et al., 2020), though it could just as easily have been produced in any other statistical programming language.[21] The code is split into 4 pieces, tied together by a (system-defined) controller script. Listing 1 provides the code for the main analysis in this example. The fully worked out example can be found at Github.

Listing 1. Example Analysis Code in Stata

```
include "config.do"
use ${inputdata}/${SSBprefix}1.dta, clear
/* data preparation omitted */
**Specify dependent variable, endogeneous education variable, and
    ↪ rest of the specification(s)
local depvar log_total_der_cpi
local educvar educyears1
local spec1 "age age_sq i.birthyear i.race foreign_born hispanic i.
    ↪ state married"
local spec2 "potexp potexp_sq i.birthyear i.race foreign_born
    ↪ hispanic i.state married"
```

[20] The use of software-specific environments in R, Python, Julia, and also Stata might also render subsequent internet access unnecessary when using the second method, but also requires software-specific checks to verify whether this is actually done correctly.

[21] Note that the current SSB access mechanism (U.S. Census Bureau, 2024) requires the use of Stata or SAS.

```
local specs ""‘spec1’" "‘spec2’""

**Loop through regressions, by gender
eststo clear
fvset base 1958 birthyear
foreach g in 1 0 {
foreach spec of local specs {
    eststo: reg ‘depvar’ ‘educvar’ ‘spec’ if male==‘g’
    eststo: ivregress 2sls ‘depvar’ ‘spec’ (‘educvar’ = i.
       ↪ birthquarter#i.birthyear) if male==‘g’
}
}
/* table creation code omitted */
```

6.4. **Dockerfile.** The environment is specified through a Dockerfile. The Dockerfile in this case specifies the use of a Codeocean-specific pre-built Stata container as the source, injects a valid Stata license, and handles installation of any Stata packages (the third method, mentioned earlier). Listing 2 shows the complete recipe, illustrating the simplicity of this approach.

Listing 2. Example Dockerfile

```
FROM registry.codeocean.com/codeocean/stata:16.0-ubuntu18.04
ARG DEBIAN_FRONTEND=noninteractive
COPY stata.lic /usr/local/stata/stata.lic
RUN stata ’ssc install estout’ \
    && stata ’ssc install outreg’ # Original versions: latest latest
```

6.5. **Executing code on public (synthetic) data using commercial infrastructure.** Once the user has developed all the code, they execute a "reproducible run" on Codeocean. This ensures that all code executes without error (note that it does *not* ensure that all necessary code has run - code can be commented out or be non-functional). This particular example, when run on Codeocean infrastructure in 2021, takes about 4 minutes to execute.[22]

6.6. **Executing code on public (synthetic) data using private or academic infrastructure.** Alternatively, the user can export the entire capsule (including data), rebuild the image locally, and execute on their local infrastructure, using an unmodified Dockerfile. Listing 3 illustrates this process when run on an appropriately configured personal computer.[23]
Running the actual code on the user's personal computer is then very straightforward (Listing 4).

[22] Unfortunately, at the time of this writing, Codeocean does not have the ability to remove data from a published compute capsule, and the Census Bureau's current license for the SSB data does not allow to publish the synthetic data. Thus, we cannot actually point to a public instance of the compute capsule.

[23] Note that the image thus created should not be publicly posted, as it contains a valid Stata license file. See for a discussion on how to build Docker images that can be distributed openly, while mounting a license only at run-time.

Listing 3. Building the container

```
cd /path/to/downloaded/capsule/environment
TAG=$(date +%F)
MYHUBID=larsvilhuber
MYIMG=ssb-demo
DOCKER_BUILDKIT=1 docker build  . -t $MYHUBID/${MYIMG}:$TAG
[+] Building 4.5s (8/8) FINISHED
 => [internal] load build definition from Dockerfile                 0.1s
 => => transferring dockerfile: 369B                                 0.0s
 => [internal] load metadata for registry.codeocean.com/codeocean/s  1.1s
 => [internal] load .dockerignore                                    0.1s
 => => transferring context: 2B                                      0.0s
 => [internal] load build context                                    0.1s
 => => transferring context: 145B                                    0.0s
 => [1/3] FROM registry.codeocean.com/codeocean/stata:16.0-ubuntu18  0.3s
 => => resolve registry.codeocean.com/codeocean/stata:16.0-ubuntu18  0.0s
 => => sha256:fd0e31b739fe4aab4c10a61d4d0460143bb09d868fbbda6c69848  0.0s
 => => sha256:4df859214231ed09630bc2a5845a75aebdab67044dff8a2933b2e  0.0s
 => [2/3] COPY stata.lic /usr/local/stata/stata.lic                  0.1s
 => [3/3] RUN stata ’ssc install estout’     && stata ’ssc install   2.6s
 => exporting to image                                               0.1s
 => => exporting layers                                              0.1s
 => => writing image sha256:f703842568b4fac943be530a6170c953f81fc29  0.0s
 => => naming to docker.io/larsvilhuber/ssb-demo:2025-04-07          0.0s
```

Listing 4. Running the container

```
cd /path/to/downloaded/capsule/
docker run -it --rm \
  -v $(pwd)/code:/code \
  -v $(pwd)/data:/data \
  -v $(pwd)/results:/results \
  -w /code \
  $MYHUBID/${MYIMG} ./run
```

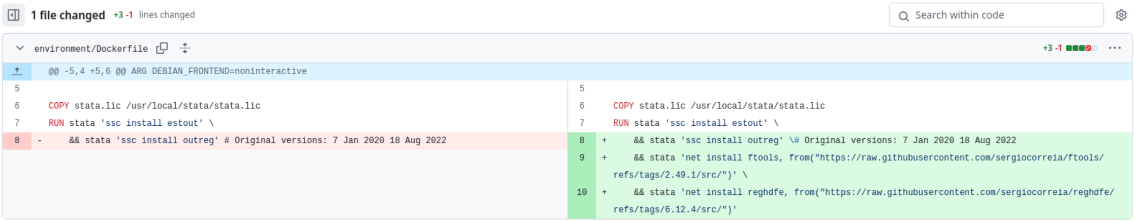
6.7. **Submitting code to the agency.** Once the user is satisfied, they can submit their code to the Census Bureau, much as they are requested to do at present (U.S. Census Bureau, 2024). Notably, they would not submit the Docker **image** built in the previous step, or provided by Codeocean, only, if modified, the Dockerfile (see Listing 5).

Listing 5. Submitted files

```
./code/00_setup.do
./code/01_stats.do
./code/02_mincer.do
./code/LICENSE
./code/README.md
./code/config.do
```

```
./code/run
./environment/Dockerfile
```

The agency can do all the usual checks on the (plaintext) statistical code, and can verify if any modified Dockerfile might cause any issues. This can be done using simple tools like `diff` (Figure 4).



```
@@ -5,4 +5,6 @@ ARG DEBIAN_FRONTEND=noninteractive
```

```

COPY stata.lic /usr/local/stata/stata.lic
RUN stata 'ssc install estout' \
-     && stata 'ssc install outreg' # Original versions: 7 Jan 2020 18 Aug 2022
```

```

COPY stata.lic /usr/local/stata/stata.lic
RUN stata 'ssc install estout' \
+     && stata 'ssc install outreg' \# Original versions: 7 Jan 2020 18 Aug 2022
+     && stata 'net install ftools, from("https://raw.githubusercontent.com/sergiocorreia/ftools/
refs/tags/2.49.1/src/")' \
+     && stata 'net install reghdfe, from("https://raw.githubusercontent.com/sergiocorreia/reghdfe/
refs/tags/6.12.4/src/")'
```

Figure 4. Checking modifications to user-provided container recipe

Importantly, all such content-based checks should be clearly identified to the users, and could be implemented during an upload process, so that efficient and rapid feedback is given to users.

6.8. **Validating researcher-provided code.** While Codeocean or other providers may confirm that the code runs reproducibly, there may be situations where the data custodians wants to separately validate reproducibility. They may have doubts, because some of the code is commented out, or any other indicator that would warrant further inspection. One possible reason is that the code was not actually validated by a commercial provider. The agency can then also run the container again, in an unsecure environment, using the synthetic data, possibly fully automated as part of the submission system. Nothing needs to pass into the secure environment for this to happen.[24]

6.9. **Porting to confidential compute server.** In order to conduct a validation exercise, the code needs to be re-executed in the secure data environment. If the user used Codeocean and transmitted the capsule identifier, the compute capsule can be exported (via "Capsule -> Export") or more likely, pulled via an API. If the user submitted code to a web interface (Listing 5), then the agency already has the code.[25] Since exporting the package is done here by the data owner, exporting the (synthetic) data is not necessary, making for a light package.

6.10. **Dynamic code provided by data custodian.** As configured in the present example, the code requires only minor modifications to work on confidential data. The data custodian can provide a 'config.do' that can handle switching from synthetic to confidential for specific code pieces (Listing 6).[26]

Listing 6. A dynamic configuration file

```
global confidential no

/* SSB parameters */
```

[24] How to do this as part of open infrastructure is demonstrated at https://github.com/labordynamicsinstitute/continuous-integration-stata.

[25] In both cases, pulling the code via 'git' might also be an option.

[26] `sed -i 's/confidential no/confidential yes/' config.do` would do the trick.

```
if ( "$confidential" == "no" ) {
    global SSBtype synthetic
    global inputdata "../data"
}
if ( "$confidential" == "yes" ) {
    global SSBtype confidential
    global inputdata "/confidential/data" // This needs to be
       ↪ mounted when running the capsule!
    // other confidential parameters are stored outside of this file
    include "config-confidential.do"
}
```

Alternate methods exist as well. For instance, one could test for presence of `config-confidential.do` and include it if present, overriding any parameters in the main `config.do`, or rely on environment variables. Equivalent methods exist in all programming languages.

6.11. **Modifying the container base image.** The Codeocean capsule uses a Codeocean-specific prebuilt image as input to the Dockerfile (in the case of this capsule, registry.codeocean.com/codeocean/stata:16.0-ubuntu18.04). This image will need to be replaced with an image that satisfies the data custodian's security requirements, while maintaining full compatibility with the needs of the environment. Because this example uses Stata, which behaves fairly uniformly across various Linux installs, the particular version of the Linux base image is likely not important. Alternatively, the validation exercise can be coordinated with the provider, and the provider can offer a generic security-vetted image that is verified to be functionally equivalent to the image used in the secure environment. Finally, the Docker file underlying the `stata:16.0-ubuntu18.04` image, which builds the container from scratch, can be used to rebuild a container within the secure environment.[27] At scale, this would simply use a similar, security-vetted, pre-built image available within the secure agency environment, e.g., registry.census.gov/codeocean/stata:16.0-ubuntu18.04-secure.[28]

The key feature here is that no binary code needs to be transferred into the secure environment, eliminating a substantial security risk. Only the user-provided Stata code is needed for the validation, and a limited modification of the Dockerfile. The former is already being transferred to the agency for validation activities, and at a much larger scale within secure environments such as the FSRDCs. Only the transfer of the Dockerfile is new. Since execution is in a controlled environment, and can be trivially separated from other sensitive areas (code cannot "break out" of the container), security is substantially enhanced. The execution environment is completely known to the IT personnel of the data provider. Because all code should be basic ASCII or UTF-8 text, malware or more enhanced code scanners should have no problem verifying the safety of the code. I discuss additional security considerations later.

6.12. **Replacing the input data.** Finally, the synthetic input data available in the public-facing environment needs to be replaced by confidential data. In the Stata code, this is already handled, as outlined above. In order to make this actionable, the Docker image can be executed in a particular fashion, provisioning the container with confidential data. Consider a directory with confidential SSB data (`$CONFDATA`) that looks like this:

[27] https://github.com/AEADataEditor/docker-stata/releases/tag/stata16-2021-06-09 for an example.

[28] This is a fake URL, no such registry currently exists.

```
/path/to/confidential/data:
  - ssb_v7_0_confidential1.dta
  - ssb_v7_0_confidential2.dta
  - ssb_v7_0_confidential3.dta
  - ssb_v7_0_confidential4.dta
  - ssb_v7_1_confidential1.dta
  ...
  - config-confidential.do
```

To run the provided capsule on confidential data, the confidential data directory is bind-mounted into the container, as is the configuration file for the confidential data (`config-confidential.do`). Results are stored in a request-specific output area (here referenced by `$REQUEST`). Results are written into a `results-confidential` directory, denoting that they have not yet been vetted by data custodian's disclosure avoidance procedures. All of this is under control of data custodian. Listing 7 shows how this would be implemented as a modification of Listing 4, although it is more likely to be run within an automated environment.

Listing 7. Running the container with confidential data

```
VERSION=16
TAG=16.0-ubuntu18.04-secure
MYHUBID=censusbureau
MYIMG=stata
CONFDATA=/path/to/confidential/data
REQUEST=12345
cd /path/to/requests
docker run -it --rm \
  -v $(pwd)/$REQUEST/code:/code \
  -v $(pwd)/$REQUEST/data:/data \
  -v $(pwd)/$REQUEST/results-confidential:/results \
  -v $CONFDATA/data:/confidential/data \
  -v $CONFDATA/config-confidential.do:/code/config-confidential.do \
  $MYHUBID/${MYIMG} run
```

However, it should be noted that the container does not have to be used for the validation. If the internal computing environment is faithfully reproduced by the container environment, then simply running the code, in combination with the `config-confidential.do`, may be sufficient. Thus, while containers are key to providing a user-facing environment, they are not a necessary condition for use within the existing secure computing environment.

6.13. **Sending results back to user: output vetting.** Once results have been generated, the usual disclosure avoidance workflow at the data provider is triggered. This might entail post-processing of the results, generation of additional supporting statistics (though these should generally be included in the processing), and finally, provision of the results to users.[29]

[29] I call this "usual", since I make no assumptions about changes to existing laws on data confidentiality that an agency is subject to. In order to be implementable, all data access, whether directly or via this new mechanism, must still be compliant with the laws that the agency is subject to, though some re-interpretation of what are permissible uses may be a separate channel to speed up access.

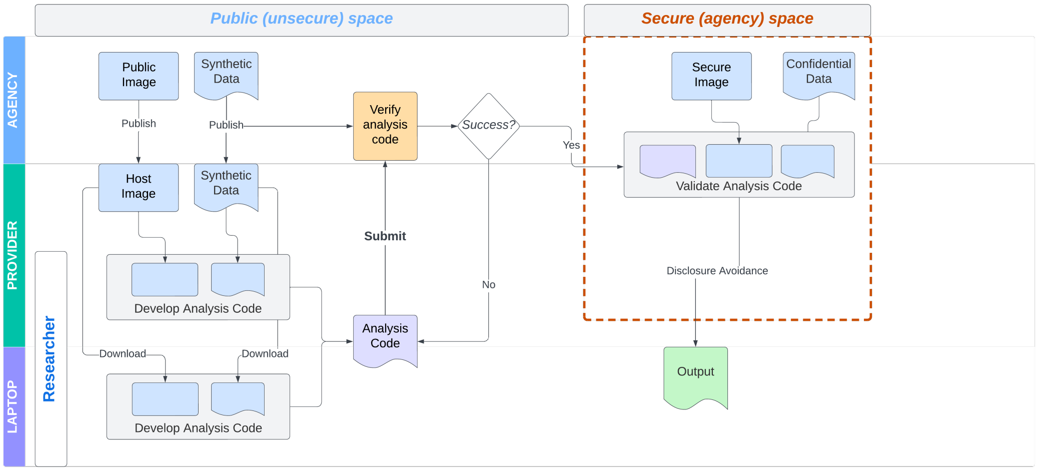


Figure 5. Sketch of a Docker-based workflow for validation of researcher code

Figure 5 provides a simplified process flow diagram, which can be compared to Figure 1. Scalability of a system as described here hinges critically on having streamlined output vetting. Ideally, this part must also be automated. At present, non-automation of output vetting is likely the single most important bottleneck of this system. However, the challenge of creating automated and reliable disclosure avoidance procedures is not unique to the validation process described here.

6.14. **Other considerations, including additional security considerations.** For Stata (and/or R code), the security implications are no worse than those faced by SSB and SynLBD validation using the Cornell Synthetic Data Server (until 2020), the replacement mechanism since 2024 (US Census Bureau, 2023, 2024), or in the present-day FSRDC. They are similar to those faced by other systems, such as the Canadian Research Data Center Network (CRDCN) or the Research Data Centre of the German Federal Employment Agency (FDZ) (Bender & Heining, 2011; Müller & vom Berge, 2021). As noted above, it should be possible to do formal scans for malware and valid statistical code, and properly sand-boxed runs should allow for functional testing.

The example above uses `docker` as a container runtime. Docker is only one of the many container-running software environments. Some statistical agencies use `podman`. By its own documentation ("Podman", n.d.), `podman` is a full "drop-in" replacement for `docker`, including the "`build`" functionality illustrated earlier in this document. `podman` does not require root privileges, one of the key concerns in general with `docker`. Apptainer ("Apptainer - Portable, Reproducible Containers", n.d.) is also an option, used for instance in the secure environment of the Bank of Portugal (Guimarães, 2023), and is used in many academic high-performance computing environments. Data curators administrating a validation system should choose the one that is authorized within their IT environment. The basic principles illustrated above can be ported to any one of the alternate runtimes and image stores.

In principle, we would suggest running all of the various steps (initial check for reproducibility and security issues, final validation against confidential data) in a proper isolated and sandboxed environment. There is no reason the entire process needs to interact with the statistical agency's systems at large, up until the actual validation against the confidential data.

Statistical agencies should always rebuild the base images. Images are layered (on other sites as well), allowing for the use of a properly security vetted container, running on a proper security vetted host. Some of the images demonstrated within this document are built from a Codeocean image:

```
FROM registry.codeocean.com/codeocean/stata:16.0-ubuntu18.04
```

but can just as easily be created from a base image I maintain with full transparency:

```
ARG SRCVERSION=17
ARG SRCTAG=2022-01-17
ARG SRCHUBID=dataeditors
FROM ${SRCHUBID}/stata${SRCVERSION}:${SRCTAG}
```

The use of a (currently nonexistent) **public** Census-sanctioned image might be used for the first step:

```
FROM registry-public.census.gov/validation/stata:17.0-rh-secure-public
```

and would simply be replaced by an equivalent but fully security compliant internal image when rebuilding the image in the confidential environment:

```
FROM registry.census.gov/validation/stata:17.0-rh-secure-internal
```

6.15. **Scalability.** For users to accept the restrictions of the synthetic data, it should scale better. So many of the vetting/building/running parts should (can easily) be streamlined. One key piece missing: standardized/streamlined output vetting.

6.16. **Data licensing.** One key condition for such a system is the ability to post synthetic data publicly, albeit classified and published as "experimental data." In contrast to the Cornell (or any other) Synthetic Data Server, the public component of the system would no longer have control over dissemination of the synthetic data files, but continue to have control over validation.[30]

As an added benefit, by compiling a library of container-based scientific uses of a particular dataset, the data provider can test out new data releases, alternative disclosure avoidance methods, or replacement data sources at scale against prior scientific findings. This is currently not (easily) feasible - most such re-validations are painstakingly manual, and limited in scale, as noted earlier. The benefit would be improved user input on new and novel methods and data.

7. Discussion

7.1. **Returning to the desiderata.** Containers satisfy most of the desiderata outlined earlier: [D1] the Docker-based mechanism supports arbitrary modeling approaches, and in principle a large number of programming languages (modulo what the agency allows); [D2] the Docker-based mechanism allows for development of models by researchers on their personal computer, if necessary, as long as they subsequently test their model within the prescribed Docker-based mechanism themselves; [D3] the mechanism is low-cost (or even at no cost) for the data provider, and can be scaled

[30] I do note that the European concept of a "scientific use file" does allow for controlled dissemination to certified educational institutions.

up effectively; the costs are born by the compute providers (and possibly charged to the user); [D4] the mechanism can be low-cost for the data user: Most current cloud computing solutions that support this are at a trivial cost to the data user, and may be born by their institution (academic or otherwise). Alternatively, development on their own personal computer imposes no additional cost; [D5] current privacy-protected data (SynLBD, SSB) are good enough to allow for complex modeling, and the low-cost nature of the validation process allows for the development of additional synthetic data releases that trade off statistical accuracy for modeling utility, which is likely to be much simpler, and [D6] validation can be technically fast, albeit with a caveat.

7.2. **Some caveats.** Two caveats are in order. First, [D5] is contingent on finding an acceptable mechanism to rapidly generate useful synthetic data, possibly model-dependent. Tools for that exist (Nowok, Raab, & Dibben, 2016; Nowok, Raab, Snoke, & Dibben, 2016), and the disclosure-avoidance analysis is improving (Snoke et al., 2016). However, in order for validation not just to be fast to process, but also fast to release to the user ([D6]), agencies need to accept and deploy privacy-protection mechanism that can scale. Current mechanisms in restricted-access data centers (FSRDC, CRDCN, FDZ) are often manual (Brandt et al., 2010), albeit first attempts at using large language models (LLMs) to speed up the process are being actively developed (Rigaud et al., 2023).

7.3. **Addressing implementation issues.** While much of the technology is straightforward "off-the-rack" technology, the fact that it is not being used by statistical agencies or survey institutes suggests that some training is required and guidelines for deployment needed. (1) Guidelines for agencies on how to prepare container specifications reflective of their internal execution environment, taking into account industry best practices on container specifications, security considerations, and usability are a necessity. NIST has some guidelines (Souppaya et al., 2017), and guidelines for secure use of containers for applications abound, and can be easily adapted to internal use at agencies. What is less available are guidelines on how to publish such containers for public consumption, these will likely need to be developed, possibly in conjunction with the target audience. (2) Users may need some guidance on general use of containers, and how to use containers in conjunction with this mechanism. There are many training documents and class notes available for general use of containers (Boettiger, 2015; Eyers et al., 2020; McDermott, 2023; Nüst, 2017; The Turing Way Community, 2024; Vilhuber, 2021, 2024), but few of them are actually known to most (academic) social scientists. How to use this specifically for the purpose of validation may need some training, though I suspect less so then the use of an idiosyncratic desktop platform that is even less familiar to users, as the SDS was. Financial incentives might be used for a pilot phase.

7.4. **How do we know it works?** The premise of my proposal is that it is easier and cheaper to implement. At present, the creation of highly protective synthetic data, and the supporting infrastructure for validation, is complex and therefore expensive.[31] How would we know if this is cheaper? Implementing agencies could track how much time was spent on submissions that were submitted with manual validation, how much time is spent managing complex data access protocols (when researchers themselves need to access the confidential data), and compare that to

[31] The SDS was supported by Sloan Foundation and National Science Foundation (NSF) grants as well as Census Bureau staff time over more than a decade, and the aforementioned project for synthetic IRS data is supported through grants by the Sloan Foundation, each of which was typically a multi-year project with cumulatively several million dollar support.

automated submissions under the current proposal (ideally in a randomized experiment), allowing an assessment of the first-order impact on costs at the agency. Other cost factors might apply as well, for instance if the agency does not currently use containers, implementation costs of deploying software to manage submissions in such an automated way. IT personnel hopefully is aware of what it means to run containers within a large organization, but may need to adapt it to the particular research environment. Non-pecuniary metrics might include the increase in usage of datasets, as well as in the number of validation requests. Second-order cost increases through expanded use remain hard to measure, and would need to be assessed in a broader cost-benefit analysis. Security and cost considerations might also affect the allowable set of software that can be used in containers, though it is an open question whether restricting users to specific software packages truly is an impediment to data use.[32]

7.5. **Possible extensions to more complex frameworks.** A critical bottleneck is disclosure avoidance, in part because the traditional analysis methods (both of public-use data and of synthetic data) are disclosure-avoidance agnostic, i.e., they do not incorporate prior or subsequent disclosure avoidance measures into the analysis method (Abowd & Schmutte, 2015). A first step is to include in the analysis container methods and metrics that allow users to assess how subsequent disclosure avoidance metrics will impact the releasable tables (Reiter, 2010). For instance, prepared methods to apply traditional disclosure methods to the designated outputs (aggregation, suppression, rounding, see Brandt et al., 2010) can be incorporated into the container. The SDS already allowed to assess inference when combining estimates from multiple synthetic datasets (the SSB provides four implicates).

Applying methods that incorporate diffential privacy directly into the analysis methods (Alabi & Vadhan, 2022; Alabi & Vadhan, 2023; Alabi et al., 2020; Chaudhuri & Monteleoni, 2008; Sheffet, 2019) still provide incomplete coverage of the analysis methods used by social scientists (see Barrientos et al., 2021, for an overview). A promising but still preliminary system at this time is the IRS synthetic data file (Burman et al., 2018; Tyagi et al., 2024). Whether such a system can be implemented with a simple container-based approach as outlined here remains to be seen.

8. In conclusion

The use of containers for validation of analyses that were prepared using synthetic data ensures reproducibility, reliable portability, and enables scalability. The use of cloud-based commercial or university-based services requires no infrastructure or software maintenance by either data provider or users, and often no direct cost. With very little effort, automation is possible (potentially through web forms), and the only likely If privacy-protection mechanisms can be tuned to acceptable protection levels (on par with traditional mechanisms that are applied to unrestricted public-use products), then validation can be made highly automated, and the quality of the synthetic data itself can be decreased, while maintaining high levels of user acceptance due to a fast validation process.

[32] Observationally, the usage of SAS in the broader academic community in the social sciences is approximately zero, yet many replication packages that are associated with FSRDC or CRDCN projects still contain SAS code.

Disclosure Statement

The author has no conflicts of interest to declare. The mention of commercial entities is not meant to endorse any such providers, and the author holds no financial interest in any of the mentioned commercial entities.

Acknowledgments

I have benefited from discussions with many folks, including Gary Benedetto, John Abowd, Rob Sienkiewicz, and from feedback following presentations to the National Academies, Census Bureau, and at the NBER conference on "Data Privacy Protection and the Conduct of Applied Research." The original development of the idea was partially funded by Alfred P. Sloan Foundation Grant G-2015-13903.

Contributions

LV conceived the topic, wrote the text, and prepared the examples. The statistical code used within the container, serving as a stand-in for any statistical analysis, was provided by Evan Totty, U.S. Census Bureau.

References

Abowd, J. M., Ashmead, R., Cumings-Menon, R., Garfinkel, S., Heineck, M., Heiss, C., Johns, R., Kifer, D., Leclerc, P., Machanavajjhala, A., Moran, B., Sexton, W., Spence, M., & Zhuravlev, P. (2022, April). The 2020 Census Disclosure Avoidance System TopDown Algorithm [arXiv:2204.08986 [cs, econ, stat]]. https://doi.org/10.48550/arXiv.2204.08986

Abowd, J. M., & Schmutte, I. (2015). Economic analysis and statistical disclosure limitation [tex.copyright: Copyright © 2015 Brookings Institution Press tex.jstor_articletype: research-article tex.publisher: Brookings Institution Press]. *Brookings Papers on Economic Activity, Fall 2015*. http://www.brookings.edu/about/projects/bpea/papers/2015/economic-analysis-statistical-disclosure-limitation

Alabi, D., McMillan, A., Sarathy, J., Smith, A., & Vadhan, S. (2020, July). Differentially Private Simple Linear Regression [arXiv:2007.05157 [cs, stat]]. Retrieved February 8, 2023, from http://arxiv.org/abs/2007.05157

Alabi, D., & Vadhan, S. (2022). *Hypothesis testing for differentially private linear regression* (arXiv No. 2206.14449) (tex.copyright: arXiv.org perpetual, non-exclusive license). arXiv. https://doi.org/10.48550/ARXIV.2206.14449

Alabi, D., & Vadhan, S. P. (2023). Differentially Private Hypothesis Testing for Linear Regression. *Journal of Machine Learning Research*, *24*, 1–50. https://jmlr.org/papers/v24/23-0045.html

Allaire, J., Teague, C., Scheidegger, C., Xie, Y., Dervieux, C., & Woodhull, G. (2024, November). *Quarto* (Version 1.6). https://doi.org/10.5281/zenodo.5960048

American Economic Association. (2016). *Allied Social Science Associations Program* (Program No. 2016). American Economic Association. San Francisco. Retrieved July 10, 2024, from https://assets.aeaweb.org/asset-server/files/815.pdf

Apptainer - Portable, Reproducible Containers. (n.d.). Retrieved April 3, 2025, from https://apptainer.org/

Armour, P., Burkhauser, R. V., & Larrimore, J. (2016). Using the Pareto distribution to improve estimates of topcoded earnings. *Economic Inquiry*, *54*(2), 1263–1273. https://doi.org/10.1111/ecin.12299

Association for Computing Machinery. (2003). ACM Software System Award: Stuart Feldman. Retrieved April 4, 2025, from https://awards.acm.org/award-recipients/feldman_1240498

Barrientos, A. F., Bolton, A., Balmat, T., Reiter, J. P., de Figueiredo, J. M., Machanavajjhala, A., Chen, Y., Kneifel, C., & DeLong, M. (2018). Providing Access to Confidential Research Data Through Synthesis and Verification: An Application to Data on Employees of the U.S. Federal Government [arXiv: 1705.07872]. *The Annals of Applied Statistics*. Retrieved March 12, 2020, from http://arxiv.org/abs/1705.07872

Barrientos, A. F., Williams, A. R., Snoke, J., & Bowen, C. (2021, November). *Differentially Private Methods for Validation Servers: A Feasibility Study on Administrative Tax Data* (tech. rep.). Urban Institute. Retrieved April 9, 2025, from https://www.urban.org/research/publication/differentially-private-methods-validation-servers

Bender, S., & Heining, J. (2011). *The Research-Data-Centre in Research-Data-Centre approach: A first step towards decentralised international data sharing* (FDZ Methodenreport No. 07/2011 (en)). Retrieved October 5, 2020, from http://doku.iab.de/fdz/reporte/2011/MR_07-11_EN.pdf

Benedetto, G., Stinson, M., & Abowd, J. M. (2013). *The creation and use of the SIPP Synthetic Beta* (tech. rep.) (tex.timestamp: 2015.02.11). US Census Bureau. http://www.census.gov/content/dam/Census/programs-surveys/sipp/methodology/SSBdescribe_nontechnical.pdf

Boettiger, C. (2015). An introduction to Docker for reproducible research. *ACM SIGOPS Operating Systems Review*, *49*(1), 71–79. https://doi.org/10.1145/2723872.2723882

Brandt, M., Franconi, L., Guerke, C., Hundepool, A., Lucarelli, M., Mol, J., Ritchie, F., Seri, G., & Welpton, R. (2010, January). *Guidelines for the checking of output based on microdata research* (tech. rep.). Retrieved April 9, 2025, from https://uwe-repository.worktribe.com/output/983615/guidelines-for-the-checking-of-output-based-on-microdata-research

Brinckman, A., Chard, K., Gaffney, N., Hategan, M., Jones, M. B., Kowalik, K., Kulasekaran, S., Ludäscher, B., Mecum, B. D., Nabrzyski, J., Stodden, V., Taylor, I. J., Turk, M. J., & Turner, K. (2018). Computing environments for reproducibility: Capturing the “Whole Tale”. *Future Generation Computer Systems*. https://doi.org/10.1016/j.future.2017.12.029

Burman, L. E., Engler, A., Khitatrakun, S., Nunns, J. R., Armstrong, S., Iselin, J., MacDonald, G., & Stallworth, P. (2018). *Administrative tax data: Creating a synthetic public use file and a validation server* (document). Tax Policy Center, Urban Institute and Brookings Institution. https://www.urban.org/research/publication/safely-expanding-research-access-administrative-tax-data-creating-synthetic-public-use-file-and-validation-server

Chard, K., Gaffney, N., Hategan, M., Kowalik, K., Ludaescher, B., McPhillips, T., Nabrzyski, J., Stodden, V., Taylor, I., Thelen, T., Turk, M. J., & Willis, C. (2020). Toward Enabling Reproducibility for Data-Intensive Research using the Whole Tale Platform [arXiv: 2005.06087]. *arXiv:2005.06087 [cs]*. https://doi.org/10.3233/APC200107

Chaudhuri, K., & Monteleoni, C. (2008). Privacy-preserving logistic regression. *Advances in Neural Information Processing Systems*, *21*. Retrieved April 9, 2025, from https://proceedings.neurips.cc/paper_files/paper/2008/hash/8065d07da4a77621450aa84fee5656d9-Abstract.html

Clyburne-Sherin, A., Fei, X., & Green, S. A. (2019). Computational Reproducibility via Containers in Psychology. *Meta-Psychology*, *3*. https://doi.org/10.15626/MP.2018.892

Contributors to the Apptainer project. (2025). Apptainer - Portable, Reproducible Containers. Retrieved April 3, 2025, from https://apptainer.org/

Daily, D. (2022, December). *Disclosure Avoidance Protections for the American Community Survey* (Blog post). US Census Bureau. Retrieved July 8, 2024, from https://www.census.gov/newsroom/blogs/random-samplings/2022/12/disclosure-avoidance-protections-acs.html

Donoho, D. (2024). Data Science at the Singularity. *Harvard Data Science Review*, *6*(1). https://doi.org/10.1162/99608f92.b91339ef

Drechsler, J. (2021). Differential Privacy for Government Agencies – Are We There Yet? *arXiv:2102.08847 [cs, stat]*. Retrieved February 24, 2021, from http://arxiv.org/abs/2102.08847

Drechsler, J., & Haensch, A.-C. (2024). 30 Years of Synthetic Data. *Statistical Science*, *39*(2). https://doi.org/10.1214/24-STS927

Eyers, D. M., Stevens, S. L. R., Turner, A., Koch, C., & Cohen, J. (2020). Reproducible Computational Environments Using Containers. Retrieved April 9, 2025, from https://carpentries-incubator.github.io/docker-introduction/

Guimarães, P. (2023). Reproducibility With Confidential Data: The Experience of BPLIM. *Harvard Data Science Review*, *5*(3). https://doi.org/10.1162/99608f92.54a00239

Heckman, J., Lochner, L., & Todd, P. (2003, May). *Fifty Years of Mincer Earnings Regressions* (working paper No. w9732). National Bureau of Economic Research. Cambridge, MA. https://doi.org/10.3386/w9732

Herbert, S., Kingi, H., Stanchi, F., & Vilhuber, L. (2024). Reproduce to validate: A comprehensive study on the reproducibility of economics research. *Canadian Journal of Economics/Revue canadienne d'économique*, caje.12728. https://doi.org/10.1111/caje.12728

Karr, A. F., Kohnen, C. N., Oganian, A., Reiter, J. P., & Sanil, A. P. (2006). A framework for evaluating the utility of data altered to protect confidentiality. *The American Statistician*, *60*(3), 1–9. https://doi.org/10.1198/000313006X124640

Kinney, S. K., Reiter, J. P., Reznek, A. P., Miranda, J., Jarmin, R. S., & Abowd, J. M. (2011). Towards unrestricted public use business microdata: The synthetic longitudinal business database [tex.owner: vilhuber tex.publisher: Blackwell Publishing Ltd tex.timestamp: 2012.09.04]. *International Statistical Review*, *79*(3), 362–384. https://doi.org/10.1111/j.1751-5823.2011.00153.x

Leisch, F. (2002). Sweave: Dynamic generation of statistical reports using literate data analysis. *International Conference on Computational Statistics*. https://link.springer.com/chapter/10.1007/978-3-642-57489-4_89

Liberman, M. (2010). Obituary: Fred Jelinek. *Computational Linguistics*, *36*(4), 595–599. https://doi.org/10.1162/coli_a_00032

Liberman, M. (2014, December). Reproducible Research and the Common Task Method. Retrieved April 3, 2025, from https://www.simonsfoundation.org/event/reproducible-research-and-the-common-task-method/

Little, R. J. (1993). Statistical analysis of masked data. *Journal of Official Statistics*, *9*(2), 407–426.

Liu, D. M., & Salganik, M. J. (2019). Successes and Struggles with Computational Reproducibility: Lessons from the Fragile Families Challenge. *Socius*, *5*, 2378023119849803. https://doi.org/10.1177/2378023119849803

McDermott, G. (2023). Data Science for Economists. Retrieved April 9, 2025, from https://github.com/uo-ec607/lectures

Meng, X.-L. (1994). Multiple-imputation inferences with uncongenial sources of input. *Statistical Science*, *9*(4), 538–558. http://www.jstor.org/stable/2246252

Mincer, J. (1984). *Schooling, experience, and earnings*. Columbia University Press.

Mölder, F., Jablonski, K. P., Letcher, B., Hall, M. B., Tomkins-Tinch, C. H., Sochat, V., Forster, J., Lee, S., Twardziok, S. O., Kanitz, A., Wilm, A., Holtgrewe, M., Rahmann, S., Nahnsen, S., & Köster, J. (2021, January). Sustainable data analysis with Snakemake. https://doi.org/10.12688/f1000research.29032.1

Moreau, D., Wiebels, K., & Boettiger, C. (2023). Containers for computational reproducibility. *Nature Reviews Methods Primers*, *3*(1), 1–16. https://doi.org/10.1038/s43586-023-00236-9

Müller, D., & vom Berge, P. (2021, January). Institute for Employment Research, Germany: International Access to Labor Market Data. In S. Cole, I. Dhaliwal, A. Sautmann, & L. Vilhuber (Eds.), *Handbook on Using Administrative Data for Research and Evidence-based Policy*. Abdul Latif Jameel Poverty Action Lab. https://doi.org/10.31485/admindatahandbook.1.0

Nowok, B., Raab, G. M., & Dibben, C. (2016). Synthpop : Bespoke creation of synthetic data in R [tex.owner: vilhuber tex.timestamp: 2017.07.04]. *Journal of Statistical Software*, *74*, 1–26. https:%20//www.jstatsoft.org/article/view/v074i11

Nowok, B., Raab, G. M., & Dibben, C. (2017). Providing bespoke synthetic data for the UK Longitudinal Studies and other sensitive data with the synthpop package for R1. *Statistical Journal of the IAOS*, *33*(3), 785–796. https://doi.org/10.3233/SJI-150153

Nowok, B., Raab, G. M., Snoke, J., & Dibben, C. (2016). *Synthpop: Generating synthetic versions of sensitive microdata for statistical disclosure control*. https://CRAN.R-project.org/package=synthpop

Nüst, D. (2017, July). Automatically archiving reproducible studies with Docker. https://doi.org/10.5281/zenodo.824007

Podman. (n.d.). Retrieved April 6, 2025, from https://podman.io/

Raghunathan, T. (2021). Synthetic Data. *Annual Review of Statistics and Its Application*, *8*(1), null. https://doi.org/10.1146/annurev-statistics-040720-031848

Raghunathan, T., & Chaney, B. (Eds.). (2023). *A Roadmap for Disclosure Avoidance in the Survey of Income and Program Participation* [Pages: 27169]. National Academies Press. https://doi.org/10.17226/27169

Raghunathan, T., & Hotz, V. J. (2024). *A roadmap for disclosure avoidance in the survey of income and program participation sipp* (Presentation). NBER.

Reeder, L. B., Stanley, J. C., & Vilhuber, L. (2018). *Codebook for the SIPP Synthetic Beta v7.0 [Codebook file]* (DDI-C document). Cornell Institute for Social, Economic Research, and Labor Dynamics Institute [distributor]. Cornell University. Ithaca, NY, USA. http://www2.ncrn.cornell.edu/ced2ar-web/codebooks/ssb/v/v7

Reiter, J. P. (2010). Multiple imputation for disclosure limitation: Future research challenges [tex.owner: vilhuber tex.timestamp: 2017.07.04]. *Journal of Privacy and Confidentiality*, *1*(2). http://repository.cmu.edu/jpc/vol1/iss2/7

Reiter, J. P. (2023). Synthetic Data: A Look Back and A Look Forward. *Transactions on Data Privacy*, *16*(1). http://www.tdp.cat/issues21/tdp.a457a22.pdf

Reiter, J. P., Oganian, A., & Karr, A. F. (2009). Verification servers: Enabling analysts to assess the quality of inferences from public use data. *Computational statistics & data analysis*, *53*(4), 1475–1482. https://doi.org/10.1016/j.csda.2008.10.006

Rigaud, T., Marquier, R., Debonnel, E., & Liu, P. (2023). Checking data outputs from research works: A mixed method with ai and human control. *United Nations Economic Commission for Europe (UNECE) Conference of European Statisticians (CES): Expert meeting on Statistical Data Confidentiality*, 26–28. Retrieved April 9, 2025, from https://scholar.google.com/scholar?cluster=9376666103583018923&hl=en&oi=scholarr

Rubin, D. B. (1993). Discussion: Statistical disclosure limitation. *Journal of Official Statistics*, *9*(2), 461–468. https://www.scb.se/contentassets/ca21efb41fee47d293bbee5bf7be7fb3/discussion-statistical-disclosure-limitation2.pdf

Sheffet, O. (2019). Differentially Private Ordinary Least Squares [Number: 1]. *Journal of Privacy and Confidentiality*, *9*(1). https://doi.org/10.29012/jpc.654

Snoke, J., Raab, G., Nowok, B., Dibben, C., & Slavkovic, A. (2016). General and specific utility measures for synthetic data [arXiv: 1604.06651v2 [stat.AP] tex.__markedentry: [vilhuber:] tex.owner: vilhuber tex.timestamp: 2017.07.04].

Souppaya, M., Morello, J., & Scarfone, K. (2017, September). *Application container security guide* (tech. rep. No. NIST SP 800-190). National Institute of Standards and Technology. Gaithersburg, MD. https://doi.org/10.6028/NIST.SP.800-190

Stockemer, D., Koehler, S., & Lentz, T. (2018). Data Access, Transparency, and Replication: New Insights from the Political Behavior Literature. *PS: Political Science & Politics*, *51*(4), 799–803. https://doi.org/10.1017/S1049096518000926

Stodden, V., Seiler, J., & Ma, Z. (2018). An empirical analysis of journal policy effectiveness for computational reproducibility. *Proceedings of the National Academy of Sciences*, 201708290. https://doi.org/10.1073/pnas.1708290115

The FreeBSD Project. (2025, March). Chapter 17. Jails and Containers. Retrieved April 3, 2025, from https://docs.freebsd.org/en/books/handbook/jails/

The Turing Way Community. (2024). Containers. Retrieved April 9, 2025, from https://book.the-turing-way.org/reproducible-research/renv/renv-containers

Tyagi, E., Taylor, S., MacDonald, G., Tamaroff, D., Miller, J., Williams, A. R., & Bowen, C. (2024, September). *A Privacy-Preserving Validation Server Version 2.0: Technical White Paper for an Automated Validation Server Prototype* (tech. rep.). Urban Institute. Retrieved April 9, 2025, from https://www.urban.org/research/publication/privacy-preserving-validation-server-version-2

U.S. Census Bureau. (2011). *Synthetic LBD Beta Version 2.0* ([Computer file]) (Published: Computer file). Cornell University, Synthetic Data Server [distributor]. Washington,DC, Ithaca, NY, USA. http://www2.vrdc.cornell.edu/news/data/lbd-synthetic-data/

U.S. Census Bureau. (2015a). *SIPP Synthetic Beta Version 7.0* ([Computer file]) (Published: Computer file). Cornell University, Synthetic Data Server [distributor]. Washington,DC, Ithaca, NY, USA. http://www2.vrdc.cornell.edu/news/data/sipp-synthetic-beta-file/

U.S. Census Bureau. (2015b, January). *Disclosure Review Board Memo: Second Request for Release of SIPP Synthetic Beta Version 6.0* (tech. rep.). U.S. Census Bureau. http://hdl.handle.net/1813/42334

US Census Bureau. (2023). Validating Results - Synthetic LBD. Retrieved April 6, 2025, from https://www.census.gov/programs-surveys/ces/data/public-use-data/synthetic-longitudinal-business-database/validating-results.html

U.S. Census Bureau. (2024). *SIPP Synthetic Beta Version 7.0* ([Computer File]). U.S. Census Bureau [distributor]. Washington, DC, USA. https://www.census.gov/programs-surveys/sipp/guidance/sipp-synthetic-beta-data-product.html
Published: Computer file.

Vilhuber, L. (2013). *Codebook for the Synthetic LBD Version 2.0 [Codebook file]* (DDI-C document). Comprehensive Extensible Data Documentation, Access Repository (CED2AR), Cornell Institute for Social, Economic Research, and Labor Dynamics Institute [distributor]. Cornell University. Ithaca, NY, USA. http://www2.ncrn.cornell.edu/ced2ar-web/codebooks/synlbd/v/v2

Vilhuber, L. (2021, November). Use of Docker for Reproducibility in Economics. Retrieved January 3, 2022, from https://aeadataeditor.github.io/posts/2021-11-16-docker

Vilhuber, L. (2024). *Aeadataeditor/docker-stata* (repository). Github. https://github.com/AEADataEditor/docker-stata

Vilhuber, L., & Abowd, J. M. (2022, July). *End of life for the Cornell Synthetic Data Server September 30, 2022* (Blog post). Cornell University. https://web.archive.org/web/20221130032540/https://www2.vrdc.cornell.edu/news/

Vilhuber, L., & Cavanagh, J. (2025). Report by the AEA Data Editor. *AEA Papers and Proceedings*, *115*, xxx–xxx. https://doi.org/10.1257/pandp.115.xxx

Vilhuber, L., Turitto, J., & Welch, K. (2020). Report by the AEA Data Editor. *AEA Papers and Proceedings*, *110*, 764–75. https://doi.org/10.1257/pandp.110.764