HDSR
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. 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. An example is provided.

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

Media Summary

The Media Summary should be written in plain language to highlight the key messages of the article, in ways that can be understood by the general public and cited by the media directly and accurately. It therefore should avoid technical terms or language designed for academic communications. It should not exceed 400 words, and more succinct, the better.

1. A short history of synthetic data, verification servers, and improving access to confidential data

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 (Daily, 2022).

On the other hand, 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

[*]lars.vilhuber@cornell.edu

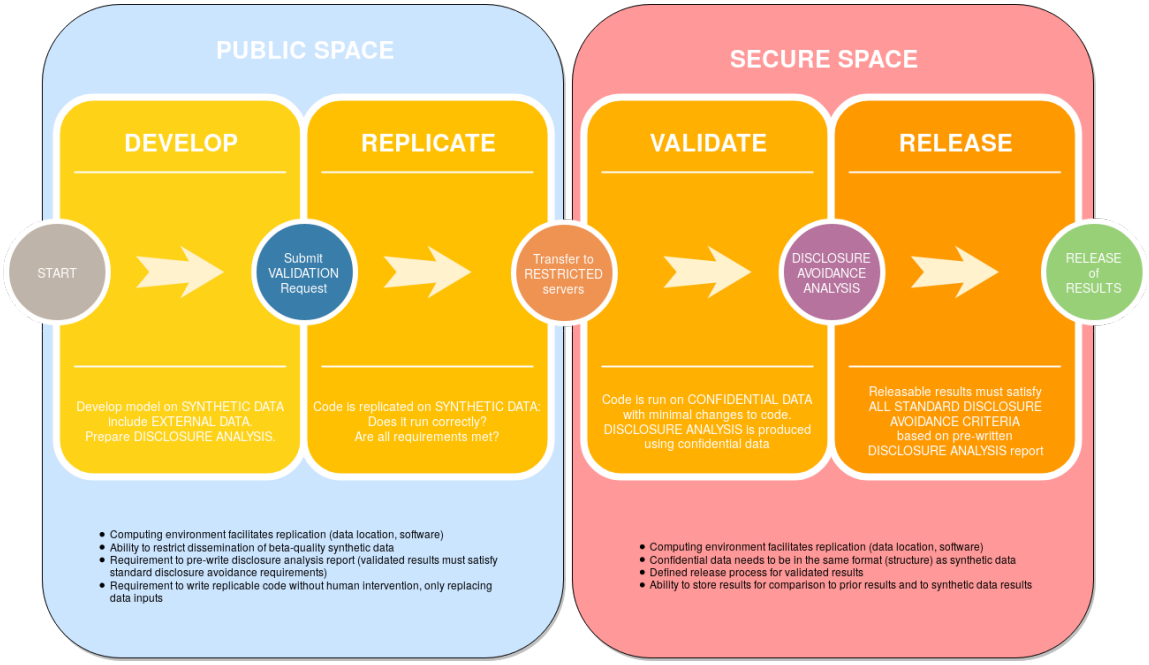

Figure 1. Processing principles of the Synthetic Data Server

2022 (Vilhuber & Abowd, 2022).[1] These pilot projects made use of a publicly accessible analysis server (the "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. Researchers obtained (fast) access to the SDS, where they interactively developed their code, using the releasable synthetic data, in an environment specifically designed to mimic the Census Bureau's environment. Requests to validate the results obtained from prepared code were sent via email to designated Census Bureau staff, who then ran the code (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.

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), 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.

[1]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.

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

2. Lessons learned from the SDS mechanism

Several lessons emerged from the SDS mechanism. While many researchers used the data to write papers, [2] and even organized conference sessions specifically around the use of the data, [3] even more researchers only "tried out" the data. While over 100 researchers were granted access to the server to access the 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). 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.

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 was 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, authors used the synthetic data to prepare

[2] 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.

[3] 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).

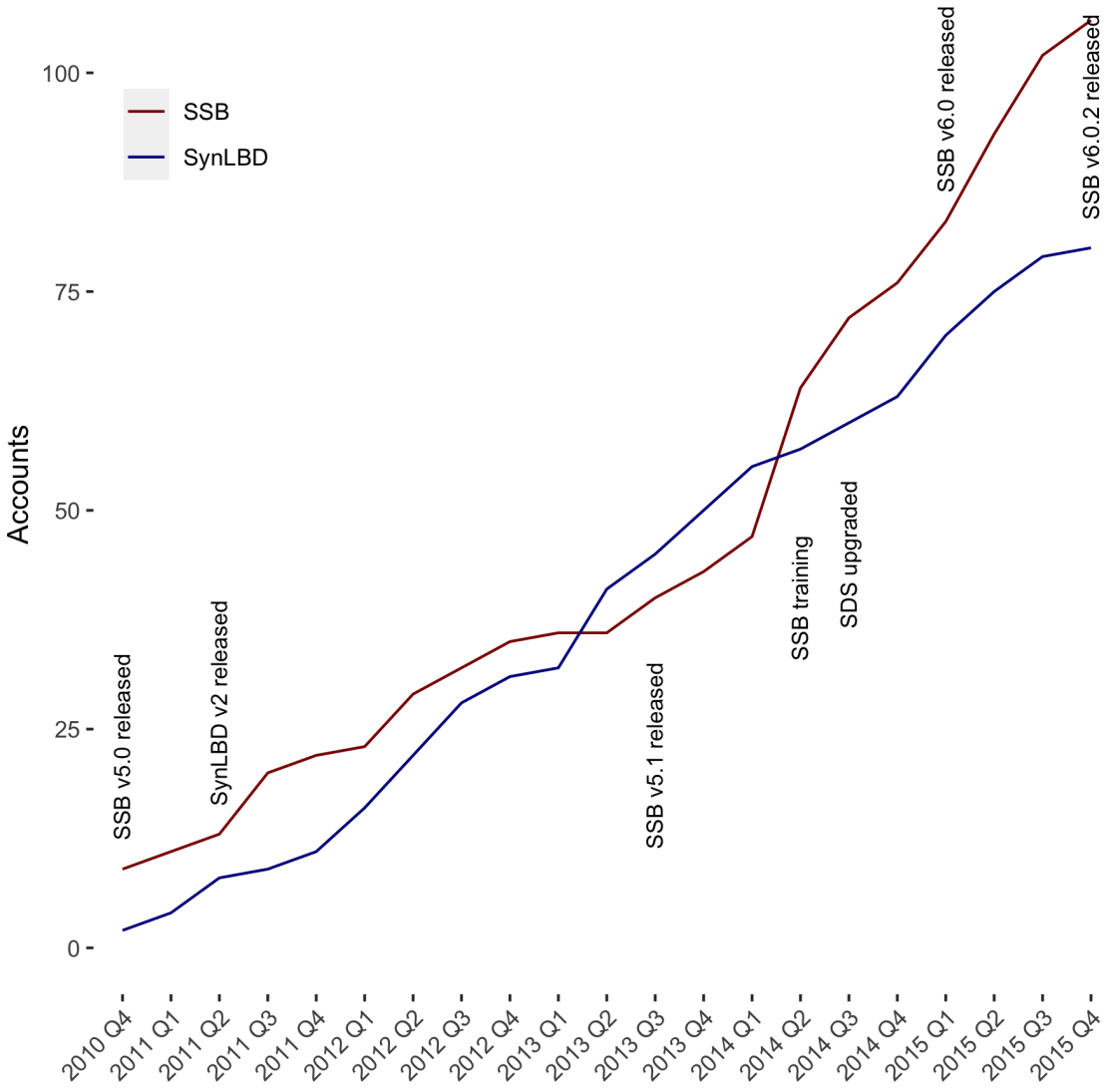

Figure 2. Computer accounts on the SDS over time

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

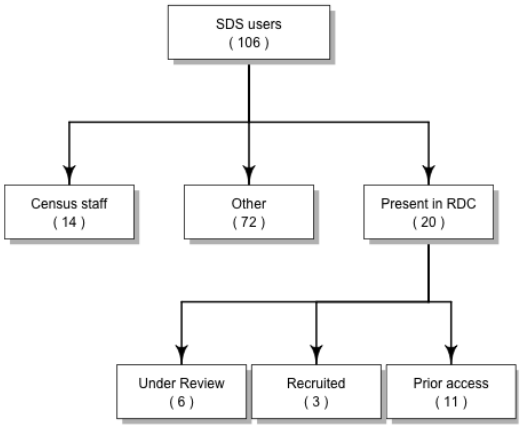

Figure 3. SDS users and access to FSRDC

the data accessed in this fashion. In general, based on conversation of the author 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 had 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 analysis that is 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. In a sample of over 8,000 replication packages associated with high-profile economics articles, only 30% had some sort of master script.[4] 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

[4]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.

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:

(1) the mechanism must support arbitrary modeling approaches and ideally a large number of programming languages
(2) the mechanism must allow for development of models by researchers that are close to their "normal" method of developing models
(3) the mechanism must be low-cost for the data provider, scaling at best sub-linearly with the number of users of those datasets
(4) the mechanism must be low-cost for the data user, imposing at best marginal costs on their existing research infrastructure (software, computers)
(5) the privacy-protected data provided as part of such mechanisms must be good enough to allow for complex modeling
(6) 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.[5]

5. A Proposal using Containers

I demonstrate a simple scenario that satisfies most of the desiderata, using containers. Containers, often referred to using the name of a particular implementation by a commercial provider (Docker), are technology most often, but not exclusively associated with Linux, 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 file, which can then be run on any computer without (much) further ado. Containers can be hosted on a cloud platform, but can also run individually on researcher compute platforms (laptops). The purpose of using containers is to provide users with access to synthetic data and coding resources such that their analysis is easily portable, and verifiably reproducible. Containers can easily be validated for reproducibility before they are then forwarded to the confidential computing environment. Once determined to be reproducible, they can then extend the analysis to use confidential data, and enable a wide spectrum of plug-in disclosure avoidance measures as well. 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.

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

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

out-of-box when using synthetic data, as well as streamlines validation against the confidential data (which is in essence a replication of the analysis on the synthetic data). Furthermore, containers enable scalability.

Codeocean is a commercial service facilitating that process by making the resources available through a web browser, though the basic functionality can be achieved on any container system. Other services in this space include Options include Wholetale, Onyxia, and many others.[6] 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.

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 economics community is not great, so far. In the same 8,000 replication packages mentioned earlier, only 0.13% had used containers.

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. 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.

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 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 recipe to build such a container.

[6]An earlier version of this presentation mentioned Gigantum. As is not unusual in this space, Gigantum no longer functions as a company. However, with the right setup, the open-source specification of most current container technology do not depend on any single provider.

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 "SIPP Synthetic Beta file" (SSB) (Benedetto et al., 2013). The container host in the public domain is Codeocean, which provides cloud-based access to Docker-based containers (called "compute capsules"). Users log in via a simple web browser, with no install required. Crucially, Codeocean provides access to Stata and Matlab compute capsules, covering 95% of economists' software needs. 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 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.

In this configuration, the Census Bureau can use 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 (Codeocean also provides a limited amount of storage space and computing time for free), or they may download the compute capsule to their local machine and use their own computing resources. When the user requests validation, the Census Bureau can download the user's compute capsule, securely rebuild the container (if necessary), and make minimal changes to excecute 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 ensures 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, it also ensures 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.**

This repository itself is the code. The code is written in Stata, and is confirmed to run on in this compute capsule.

Basic structure[7] imposes a strict separation between code, immutable data, and reproducible results:

- all code is under '/code'
- all data are under '/data'
- all outputs [MUST be written to '/results'](https://help.Codeocean.com/getting-started/uploading-code-and-data/saving-files), 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, the template code includes a `config.do` file, which applies best programming practices by defining some global variables for file paths that are used elsewhere in the code. It also instantiates logfiles, which are also written to '/results'.

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

Stata, R, and many other programming languages use external packages of code 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 2. Packages can be installed by a programming-language specific install script, and stored locally. 3. Packages can be incorporated into the container image itself.

Codeocean has the ability to do the third method, via a "post-install" script and environment setup (see [here](https://help.Codeocean.com/getting-started/the-computational-environment/using-the-postinstall-script-for-further-customization) and [here](https://help.Codeocean.com/tips-and-tricks/language-specific-issues/using-stata-on-code-ocean)). It can also support the first method during runtime (while connected to the internet). Because each run of Stata is transitory, there is no easy way to accomodate the second method.

I note that while hosted on Codeocean, the same image, once built, is re-used, and packages 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.** In its base configuration, Codeocean signals to researchers the successful completion of a run of the controller script ‘run‘ in the right pane of the user interface, indicating to the custodian of the confidential data that the code is verified to execute on the synthetic data. This is important for scalability and efficiency, as it reduces the need for extensive debugging.

6.3. **Concrete example: Estimating economic returns to education in the SIPP.** This example runs a Mincer equation on the SIPP Synthetic Beta data (need cite). The code is split into 4 pieces, tied together by a script. The environment is specified through a Dockerfile.

6.3.1. *Dockerfile.* The Dockerfile in this case specifies the use of a Codeocean-specific pre-built Stata container, and handles installation of any Stata packages:

Listing 1. 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
```

6.3.2. *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.

6.3.3. *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 2. Building the container

```
cd /path/to/downloaded/capsule/environment
VERSION=16
TAG=$(date +%F)
MYHUBID=larsvilhuber
MYIMG=ssb-demo
DOCKER_BUILDKIT=1 docker build  . -t $MYHUBID/${MYIMG}:$TAG
[+] Building 5.9s (8/8) FINISHED
 => [internal] load build definition from Dockerfile
    ↪ 0.0s
 => => transferring dockerfile: 365B
    ↪ 0.0s
 => [internal] load .dockerignore
    ↪ 0.0s
 => => transferring context: 2B
    ↪ 0.0s
 => [internal] load metadata for registry.codeocean.com/codeocean/stata:1
    ↪ 0.0s
 => [internal] load build context
    ↪ 0.0s
 => => transferring context: 133B
    ↪ 0.0s
 => [1/3] FROM registry.codeocean.com/codeocean/stata:16.0-ubuntu18.04
    ↪ 0.0s
 => CACHED [2/3] COPY stata.lic /usr/local/stata/stata.lic
    ↪ 0.0s
 => [3/3] RUN stata 'ssc install estout'     && stata 'ssc install outreg
    ↪ 5.8s
 => exporting to image
    ↪ 0.0s
 => => exporting layers
    ↪ 0.0s
 => => writing image sha256:c76d3d1981c510f744cdd65e3f0c2321bc0b7a99e5285
    ↪ 0.0s
 => => naming to docker.io/larsvilhuber/ssb-demo:2022-09-11
    ↪ 0.0s
```

Note that the image built has been posted publicly at https://hub.docker.com/r/larsvilhuber/ssb-demo.

6.3.4. *Validating researcher-provided code.* Should the data custodian have doubts about the verified run of the capsule, or the capsule was not validated on Codeocean (because run on private infrastructure), the replicator can run the container again, using the synthetic data. This can happen in an unsecure environment, outside of the confidential data environment, since no additional data requirements need to be satisfied.

Listing 3. 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
```

which runs for about 3 minutes on a 2021-vintage Linux workstation.

6.3.5. *Porting to confidential compute server.* In order to conduct a validation exercise, the code needs to be re-executed in the secure data environment. The compute capsule is exported (via "Capsule -> Export"), which provides a full package. Since exporting the package is done here by the data owner, exporting the data is not necessary, making for a light package. Alternatively, the code can also be downloaded via 'git clone' from the default CodeOcean git repository, or from a researcher's git repository. Note that it is not necessary to publish the CodeOcean capsule or to make a git repository publicly viewable, as long as it is shared with replicator.

6.3.6. *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 4).

Listing 4. A dynamic configuration file

```
global confidential no

/* SSB parameters */
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'.

6.4. **Modifying the container base image.** The CodeOcean capsule uses a CodeOcean-specific prebuilt container used to execute the code (in the case of this capsule, registry.codeocean.com/codeocean/stata:16.0-ubuntu18.04). This image will need to be replaced with a container that satisfies the data owner'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.[8] At scale, this would simply use a similar, security-vetted, pre-built container, e.g., registry.census.gov/codeocean/stata:16.0-ubuntu18.04-secure.[9]

The key feature here is that no binary code needs to be transferred into the secure environment, eliminating a security risk. The execution environment is completely known to the IT personnel of

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

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

the data provider. Only the user-provided Stata code is needed for the validation. Since execution is in a controlled environment, and can be trivially separated from othe sensitive areas (code cannot "break out" of the container), security is substantially enhanced. Because all code should be basic ASCII or UTF-8 code, malware or more enhanced code scanners should have no problem verifying the safety of the code. I discuss additional security considerations later.

6.4.1. *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:

```
/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 provider's disclosure avoidance procedures.

Listing 5. Running the container with confidential data

```
VERSION=16
TAG=16.0-ubuntu18.04-secure
MYHUBID=censusbureau
MYIMG=stata
STATALIC=/path/to/stata/licenses
CONFDATA=/path/to/confidential/data
REQUEST=12345
cd /path/to/requests
docker run -it --rm \
  -v ${STATALIC}/stata.lic.${VERSION}:/usr/local/stata/stata.lic \
  -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
```

6.4.2. *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.

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. Howver, the challenge of creating automated and reliable disclosure avoidance procedures is not unique to the validation process described here.

6.5. **Other considerations, including additional security considerations.** For Stata (and/or R code), the security implications are no worse than those currently faced by SSB Validation using the Cornell Synthetic Data Server. They are similar to those faced by other systems, such as the German IAB (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 (reference), `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`. Singularity is also an option, used for instance in the RDC environment of the Bank of Portugal (Guimarães, 2023). Data curators administrating a validation system should choose the one that is authorized within their IT environment.

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.

Statistical agencies should always rebuild the containers. Containers 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 containers 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 maintained by one of the authors 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
```

6.5.1. *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.

6.5.2. *Data licensing.* One key condition for such a system is the ability to post SSB 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.[10]

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. The benefit would be improved user input on new and novel methods and data.

7. Conclusion

The use of containers ensures reproducibility, reliable portability, and enables scalability. The use of cloud-based commercial services requires no infrastructure or software maintenance by either data provider or users, but is not a necessary condition, as users can easily provide their own infrastructure. With very little effort, automation is possible (potentially through web forms), and the only likely constraint to full automation is the absence of automated output vetting algorithms.

Thus, containers satisfy most of the desiderate outlined earlier, but still rely on high-quality synthetic data, and a privacy-protection mechanism that can scale. If such a privacy-protection mechanism 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.

Disclosure Statement. The author have 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.

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

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

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

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

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

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

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

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

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

Moreau, D., Wiebels, K., & Boettiger, C. (2023). Containers for computational reproducibility [Number: 1 Publisher: Nature Publishing Group OA version: https://minio.carlboettiger.info/shared-data/Containers.pdf]. *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

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., 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

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

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., & 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/20230602202220/https://web.archive.org/web/20221130032540/https://www2.vrdc.cornell.edu/news/