Privacy-preserving impact evaluation using Differences-in-Differences

Manuel Huth [1,2], Carolina Alvarez Garavito [2], Laia Cirera [3], Elisa Sicuri [3,4], and Jan Hasenauer [1,2]

[1] Institute for Computational Biology, Helmholtz Munich - German Research Center for Environmental Health, Munich, Germany.
[2] Faculty of Mathematics and Natural Sciences, University of Bonn, Bonn, Germany.
[3] ISGlobal, Hospital Clínic - Universitat de Barcelona, Barcelona, Spain.
[4] LSE Health - Department of Health Policy, London School of Economics and Political Science, London, UK.
* Correspondence: jan.hasenhauer@uni-bonn.de

Abstract

Difference-in-Differences is a widely used tool for causal impact evaluation. However, its application to data containing sensitive personal information, such as individual-level health care data, is limited due to strict data privacy regulations. Obtaining consent from subjects often leads to smaller sample sizes or no individuals from treated/untreated groups, reducing statistical power or making impact estimation impossible. Federated Learning, a distributed learning approach that ensures privacy by sharing aggregated statistics, can address data protection concerns. However, software packages for advanced Difference-in-Differences algorithms are yet unavailable. We have developed a federated version of the Difference-in-Differences with multiple time periods, which we have implemented. Our federated package ensures data privacy and is able to reproduce outputs of the non-federated implementation. By leveraging federated estimates, we are able to increase effective sample sizes, reduce estimation uncertainty, and make estimation possible in cases where data owners only have access to the treated or untreated group.

Background

Difference-in-Differences (DID) is a widely-used tool for evaluating treatment effects across many fields, including clinical research [Galiani et al., 2005, Miller and Wherry, 2017], Epidemiology [Goodman-Bacon and Marcus, 2020, Groeniger et al., 2021], and Economics [Colchero et al., 2016, Wen et al., 2015]. Early DID estimates have focused on the two-period canonical DID where data from two distinct time periods is used to estimate the treatment effect [Card and Krueger, 1993]. Two-way fixed effect estimators have extended the framework allowing estimators with variation

in treatment timing. Latest advances that have gained large attention in the field have focused on continuous treatment effects [Callaway et al., 2021], variation in treatment timing [Goodman-Bacon, 2021, Callaway and Sant'Anna, 2021] and varying effects across multiple time periods [Callaway and Sant'Anna, 2021]. The approach of Callaway and Sant'Anna [Callaway and Sant'Anna, 2021] can account for a varying impact strength over different time periods, variations in the timing of treatments, and when the assumption of parallel trends holds true only after controlling for covariates. The method is available as a software package for the statistical programming languages R [Callaway and Sant'Anna, 2021] and Stata [Rios-Avila et al., 2023]. In this paper, we follow the naming convention used in Stata and refer to the approach as CSDID (Callaway and Sant'Anna Difference-in-Differences) [Rios-Avila et al., 2023]. As a policy evaluation tool, CSDID has been applied, inter alia, to interventions on teen births [Mark and Wu, 2022] and the adoption of electric vehicles as a response to an improved charging infrastructure [Schulz and Rode, 2022]. Yet, the application to data that involves sensitive individual-level outcomes, e.g. patient data or test performances, has been limited.

In the case of data that involves sensitive individual-level outcomes, the sharing of data is restricted by legal frameworks like the General Data Protection Regulation (GDPR) [Hansen et al., 2021]. Obtaining consent from all potential participants can therefore be challenging. As policies are often subject to regional implementations, some entities, e.g. schools or regional governments, might either have data for only treated or only untreated individuals, making the use of the CSDID impossible without the use of data aggregation. Moreover, worse performing individuals, e.g. students, may choose not to share their performances due to data-privacy concerns. This non-randomized selection of participants can introduce bias into the CSDID estimates, thereby limiting its utility in these contexts. This limitation can be resolved by applying Federated Learning [McMahan et al., 2017], which enables collaboration among multiple data owners who only share summary statistics, resulting in joint model training with larger sample sizes, all while preserving individual data privacy (Figure 1). As a desirable feature, Federated Learning can produce parameter estimates with convergence properties identical to central learning (Figure 1A), or in many cases even lead to the same parameter estimates. In the context of the CSDID, this can help to enable the analysis or to overcome the selection bias, while increasing sample sizes leading to stronger statistical power. Due to these desirable properties, Federated Learning has been applied to a variety of biomedical studies [Dayan et al., 2021, Harrison et al., 2020, Jannasch et al., 2022, Li et al., 2020].

Amongst the available Federated Learning platforms, DataSHIELD [Marcon et al., 2021] is an established toolbox implemented in the statistical computing language R [R Core Team, 2022]. Within the biomedical community, DataSHIELD is, among others, used by the European Union's Horizon 2020 projects ORCHESTRA and unCover [Tacconelli et al., 2022]. Many tools for biomedical analysis have already been implemented in DataSHIELD, e.g. Survival Analysis [Banerjee et al., 2022] and deep Boltzmann Machines [Lenz et al., 2021]. Yet a tool that allows for the implementation of causal impact evaluation is currently missing.

We developed a Federated Learning package in DataSHIELD with a distributed version of the CSDID approach [Callaway and Sant'Anna, 2021]. We are able to obtain exact federated treatment

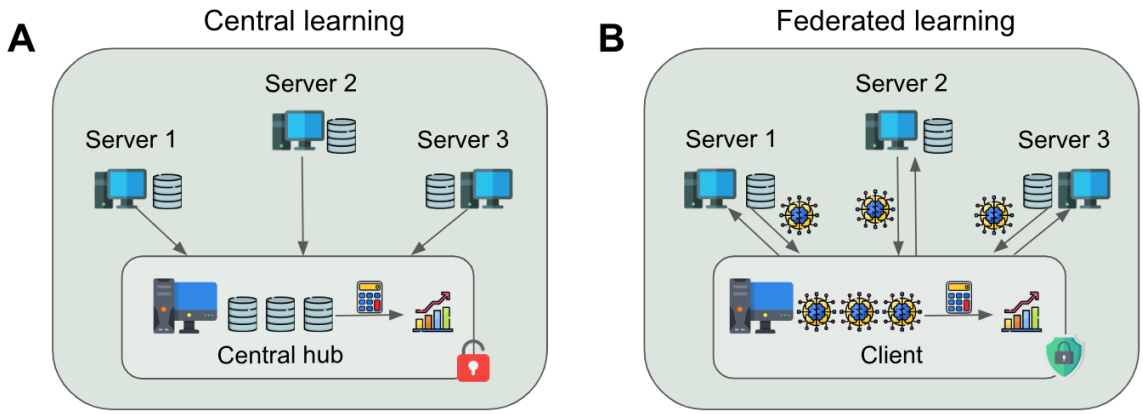

Figure 1: Learning paradigma. **A** Explanation of Central Learning. In Central Learning, models are trained at one central hub (analyst). All servers (data owners) send their data to the central hub, granting the analyst full access to individual-level information without preserving data privacy. **B** Explanation of Federated Learning. In Federated Learning, model updates are computed locally at the servers, and only aggregated information is sent to the analyst (client). The locally aggregated information is then further aggregated to compute overall model parameter updates, which are sent back to the servers. This iterative process continues until a convergence criterion is met, ensuring efficient and privacy-preserving collaboration between servers and the analyst. This figure has been designed using resources from Flaticon.com

effects and asymptotic standard errors, as well as distributional equivalent bootstrapped standard errors compared to the package developed for the non-federated implementation [Callaway and Sant'Anna, 2021]. It's worth mentioning that the ordinary regression DID can be implemented in principle within DataSHIELD using the `ds.glm` function for (generalized) linear models by creating appropriate dummy variables. However, due to less flexibility of treatment effects and the additional unconditional parallel trend assumption we concluded that there is a need for the more sophisticated CSDID. Our software incorporates the standard DataSHIELD security measures such as validity checks on the minimum non-zero counts of observational units for, e.g., mean calculations, and the maximum number of parameters in a regression. In order to prevent attacks as described in Huth et al. [2023], we additionally include further security measures tailored to the needs of the CSDID method. The implementation of these measures guarantees that the computations are carried out in a secure and confidential manner, protecting sensitive data from unauthorized access or use.

In the remainder of this paper, we state the basics of the federated CSDID [Callaway and Sant'Anna, 2021], explain the additional security measures implemented in the software, and showcase its use with a simulation study.

Methods

DataSHIELD

DataSHIELD [Gaye et al., 2014, Wilson et al., 2017] is an advanced distributed learning ecosystem that leverages Federated Learning and federated meta-analysis to facilitate secure and collaborative data analysis across multiple data owners without sharing individual-level data (Figure 1B). This approach ensures that high statistical power is maintained by using larger sample sizes enabled

by data privacy. Federated algorithms are implemented in DataSHIELD through a client-server structure, where client-side packages are called on the analyst's machine and server-side packages are installed on the data owner's machines. Server-side functions aggregate the data locally, detect disclosure risks such as small sample sizes, and send non-disclosive information to the analyst to update parameters or compute overall-aggregated statistics.

The dsBase package [Developers, 2022] is the core of the DataSHIELD package system and provides a wide range of capabilities, including the computation of summary statistics and generalized linear models. Additionally, user-written packages extend DataSHIELD's capabilities, such as the implementation of restricted Boltzmann machines [Lenz et al., 2021], Omics analysis [Gonzalez et al., 2021], and mediation methods [Avraam and Wheater, 2021]. Our software implementation in DataSHIELD supplements the framework by providing a rigorous tool for evaluating causal impact and treatment effects via the federated version of the CSDID.

Difference-in-Differences with multiple time periods

The CSDID is employed to estimate treatment effects with varying treatment timing. Specifically, for each treatment period g and evaluation period t, individuals are assigned to either a treatment or a control group (Figure 2), and treatment effects are estimated based on these individuals using a doubly robust (DR), inverse-probability weighting (IPW), or a regression outcome (RO) approach. Since IPW and RO are nested within DR, we use the notation for DR estimate in the remainder of this section. Equivalent formulations for the IPW and RO approaches are provided in the work of Callaway and Sant'Anna [Callaway and Sant'Anna, 2021]. The treatment group comprises all individuals who received treatment in period g, while the control group can consist of individuals who have never been treated or have not yet been treated in period g. In the remainder, we use the notation for never-treated individuals as control group in order to assist readers. Subsequently, we adhere closely to the notation used in the non-federated work of Callaway and Sant'Anna [Callaway and Sant'Anna, 2021].

Based on the treatment and the control group, the DR average treatment effect (ATT) [Callaway and Sant'Anna, 2021] is represented by

$$ATT(g,t;\delta) = E\left[\left(\frac{G_g}{E[G_g]} - \frac{\frac{p_g(X)C}{1-p_g(X)}}{E\left[\frac{p_g(X)C}{1-p_g(X)}\right]}\right)(Y_t - Y_{g-\delta-1} - m_{g,t,\delta}(X))\right], \tag{1}$$

where δ is the number of periods during which individuals have knowledge of the treatment in advance, Y_t is the outcome of interest in period t, and X are pre-treatment covariates. The binary variable G_g indicates whether an individual was treated in period g, and C is a binary variable indicating whether an individual has never been treated. The probability of receiving treatment in period g, given pre-treatment covariates and treatment status, is denoted by $p_g(X) = E\left[G_g|X, G_g + C = 1\right]$. And the expected outcome difference of the control group is denoted by $m_{g,t,\delta}(X) = E\left[Y_t - Y_{g-\delta-1}|X, C = 1\right]$. To calculate the estimate of the average treatment effect, $\widehat{ATT}(g,t;\delta)$, the quantities in (1) are replaced by their sample analogues.

To quantify the uncertainty of the treatment effect estimate, the CSDID approach offers two options: (i) asymptotic and (ii) bootstrapped standard errors (Figure 2) [Callaway and Sant'Anna, 2021, Callaway and Sant'Anna, 2021]. (i) Asymptotic standard errors are computed by estimating the variance-covariance matrix of the limiting distribution of the average treatment effect estimator

$$\sqrt{n}\left(\widehat{ATT}_{t\geq g-\delta} - ATT_{t\geq g-\delta}\right) \overset{d}{\sim} \mathcal{N}\left(0, E\left[\Psi_{t\geq g-\delta}(W)\Psi_{t\geq g-\delta}(W)^T\right]\right), \tag{2}$$

where $ATT_{t\geq g-\delta}$ is the collection of $ATT(g,t;\delta)$ with $t \geq g-\delta$ and $\Psi_{t\geq g-\delta}(W)$ is the collection of relevant influence functions $\psi_{g,t,\delta}$. Details on the influence functions are given in the supplementary material and in the work of Callaway and Sant'Anna [Callaway and Sant'Anna, 2021]. (ii) The multiplier bootstrap [Callaway and Sant'Anna, 2021] involves drawing B independent and identically distributed random samples from the distribution of the average treatment effect

$$\widehat{ATT}^{(b)}_{t\geq g-\delta} = \widehat{ATT}_{t\geq g-\delta} + E_n\left[V \cdot \hat{\Psi}_{t\geq g-\delta}(W)\right], \tag{3}$$

where $\hat{\Psi}_{t\geq g-\delta}(W)$ is the sample analogue of $\Psi_{t\geq g-\delta}(W)$ and $E_n(a)$ is the expectation over the components of a vector a. V is a random variable with $P(V = 1 - \frac{\sqrt{5}+1}{2}) = \frac{\sqrt{5}+1}{2\sqrt{5}}$ and $P(V = \frac{\sqrt{5}+1}{2}) = 1 - \frac{\sqrt{5}+1}{2\sqrt{5}}$. Subsequently, the variance-covariance matrix can be estimated by the empirical distribution of the B bootstrap draws.

Federated implementation in DataSHIELD

The package aims to minimize information transfer between analysts and data owners in order to reduce data leakage risks and communication overhead while obtaining estimates equivalent to pooled estimation. As for every DataSHIELD analysis, users require login credentials that authenticate them as analysts to perform the federated analysis, providing an added layer of privacy and security for the individuals involved. Our software package is fully open-source and consists of both a client-side [Huth, 2023a] and server-side [Huth, 2023b] R package. The server-side package can be accessed through Opal [OBiBa, 2022] from the client side. All necessary client-side packages are installed as a dependency when the client-side package (dsDidClient) is installed, streamlining the setup process for users. To run the package, the server-side package (dsDid) [Huth, 2023b] and a version of the DataSHIELD base package [Developers, 2022] must be installed on the servers, since they are called in order to conduct the local privacy-safe computations and aggregations (Figure 3) on the servers.

The core function of the client-side package is `ds.did`, which returns average treatment effects (1) and asymptotic (bootstrapped) standard errors ((11) and (12)). As in the non-federated package of Callaway and Sant'Anna [Callaway and Sant'Anna, 2021], the user can choose between the DR, OR and IPW estimates to compute treatment effects. The control group can be chosen to be all never-treated individuals or all individuals that have not been treated in the period of interest.

In some scenarios, treated and untreated individuals might be separately stored on different servers. During the estimation of the average treatment effect (1) one needs to estimate the expected outcome difference of the control group by $m_{g,t,\delta}(X) = E\left[Y_t - Y_{g-\delta-1}|X, C = 1\right]$. Inference for this is

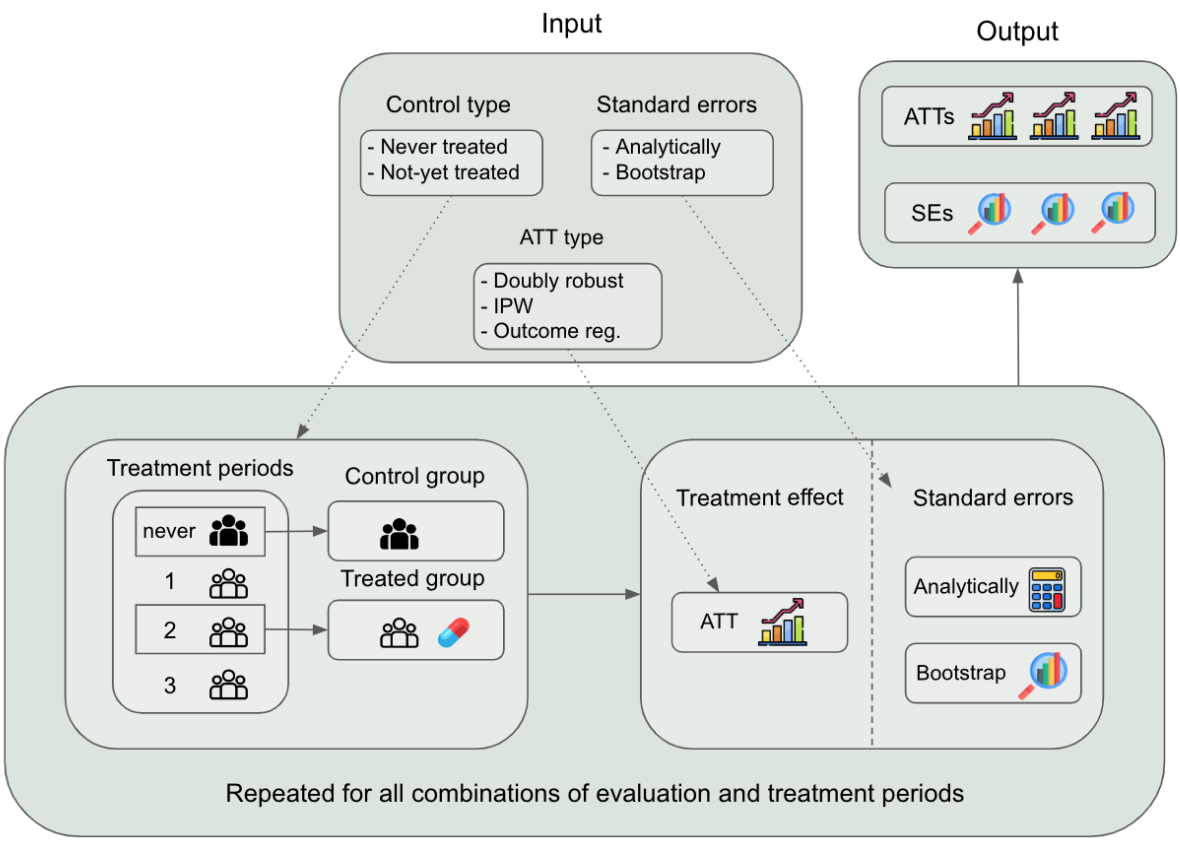

Figure 2: Functionality of CSDID. Visualization of the Callaway and Sant'Anna Difference-in-Differences estimator [Callaway and Sant'Anna, 2021, Callaway and Sant'Anna, 2021]. The Inputs specify the different types of estimators, standard errors, and control groups that can be specified. After the inputs are chosen, the estimator is computed for each combination of treatment periods and evaluation periods. This figure has been designed using resources from Flaticon.com.

implemented in the R package of Callaway and Sant'Anna with a linear regression on the subset of untreated individuals.

CSDID subsets individuals into treatment and control groups for a fixed treatment period g and a fixed evaluation period t (Figure 2). However, in the federated analysis this may result in some servers not having enough individuals to pass standard DataSHIELD security sample size requirements that are set by data owners, especially when the number of overall observations is low. To address this issue, our package automatically checks in advance which servers have enough observations to compute treatment effects for a specific combination of g and t and only excludes these servers for these computations. This approach ensures that only servers with insufficient data are dropped, enabling larger sample sizes for treatment effect estimations to be computed from the remaining servers. This improves the accuracy and efficiency of the federated analyses while maintaining data security and privacy.

`ds.did` calls client-side functions from dsBaseClient (Table 1 in the supplementay material) and modularly-developed functions from dsDidClient. These functions, in turn, call the respective server-side packages [Huth, 2023b, Developers, 2022]. On the server, the packages utilize the stored data to compute the influence function for each individual. During this process, the standard

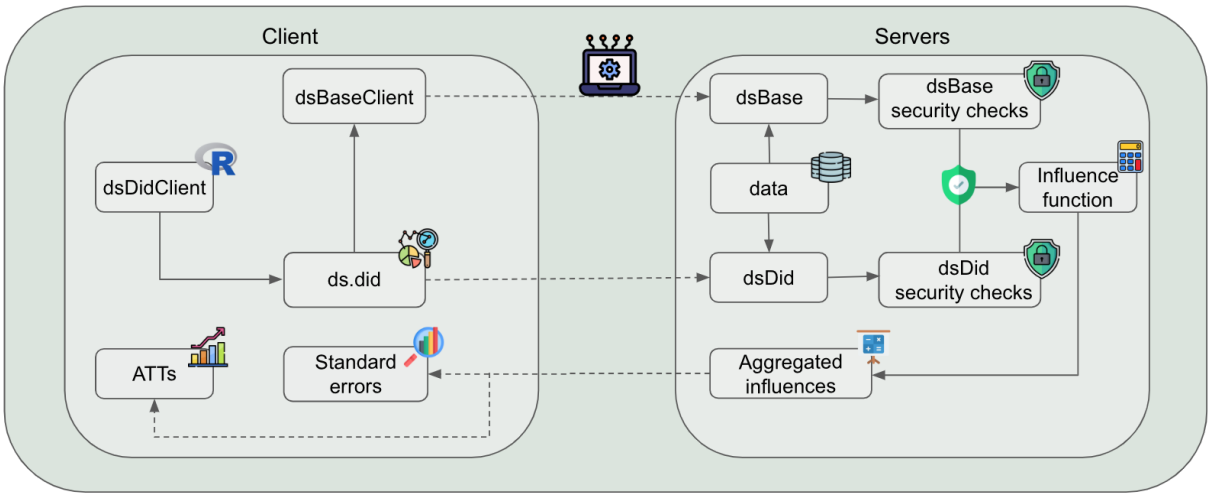

Figure 3: Implementation of CSDID in DataSHIELD. Our Federated DataSHIELD implementation of CSDID consists of a client-side package installed on the analyst's machine, and a server-side package installed on the data owner's machines. The client-side package utilizes the DataSHIELD base package functions and security layers. Additionally, more security check are conducted on the server side by dsDid prior to the return of the aggregated information to the client. This figure has been designed using resources from Flaticon.com.

DataSHIELD security checks are applied, including checking the sufficiency of sample sizes for computed subsets or aggregations of the data. Additionally, the DsDid functions are implemented with further security measures tailored to their functionality to ensure data privacy. This comprises random shuffling of data frames and limiting the type of information transmitted to the client. Detailed information on these measures is provided in the additional security measures section.

The output of `ds.did` utilizes the summary function from the non-federated package [Callaway and Sant'Anna, 2021], allowing users familiar with the non-federated package [Callaway and Sant'Anna, 2021] an the use of model diagnostic tools that do not involve individual-level data.

Additional security measures

Our package adheres to the data security standards of DataSHIELD and highest security standards following a recent assessments [Huth et al., 2023], ensuring that all function outputs are in compliance with the disclosure settings that can be configured in Opal [OBiBa, 2022]. This includes measures such as preventing subsetting of data frames if the subset does not contain the minimum number of rows and prohibiting aggregations with fewer objects than the specified threshold.

Additionally, when performing federated estimation of the influence functions, it is necessary to append columns to a matrix (`ds.AppendInfluence`). However, this process enables malicious clients to append arbitrary columns to a matrix. The creation of known linearly independent matrices can cause data leakage of all data points [Huth et al., 2023]. To mitigate this risk, we implement a strategy of row-wise shuffling of data frames and matrices after appending columns. This ensures that the original results can still be obtained through matching with respect to an ID column, while also providing a level of protection against the aforementioned attack.

To use globally summarized quantities, such as global means, on the server side, data transmission from the client to the server is required. However, unrestricted data transmission from the client to the server can again enable the execution of data leakage attacks. As a countermeasure, we process everything that is send to the server immediately, such that it is not stored on the server side. For information that must be stored on the server side, the client is limited to sending a single number at a time to the servers to prevent attacks. Sending known numbers to the servers raises the potential threat of data leakage if the client is able to concatenate multiple known single numbers into different linearly independent vectors Huth et al. [2023]. In DataSHIELD, this would be possible by sending multiple numbers to the servers using `ds.sendToServer` and concatenating them using `ds.cbind` and/or `ds.rbind`. It is therefore not recommended to use this package together with `ds.cbind` and/or `ds.rbind` from the base package of DataSHIELD [Developers, 2022], which can be disabled in Opal, in order to enhance data privacy.

The supplementary material provides further information on the specific functions and the security measures implemented to protect the data privacy.

Verifying the validity of the federated multiplier bootstrap

Our package allows to run a federated version of the multiplier bootstrap in order to obtain standard errors. To evaluate the empirical validity of the federated bootstrap method used in our package, we need to determine if the standard errors obtained from the federated bootstrap are equivalent to those obtained from the pooled bootstrap. However, directly comparing these two sets of standard errors is challenging due to the inherent randomness of the bootstrap process. To overcome this limitation, we employ Monte Carlo Simulations to generate empirical distributions for both the federated and pooled bootstrapped standard errors.

The aim is to examine whether the distributions of federated and pooled standard errors are drawn from the same underlying distribution. In a statistical testing framework, this would involve testing the null hypothesis that the two distributions are different. Unfortunately, existing tests only allow for testing the null hypothesis that the two distributions are equal, such as the Kolmogorov-Smirnov test. Consequently, we rely on visual inspection to evaluate the validity of the federated multiplier bootstrap. To perform this evaluation, we repeated the bootstrap computation 500 times using both the pooled and federated algorithms. This generated one empirical distribution of bootstrapped standard errors for each method. Additionally, we repeated this process one more time for the pooled implementation of the Callaway and Sant'Anna package, creating a reference distribution. Since both pooled distributions are guaranteed to originate from the same distribution, we used the difference between these distributions as a benchmark for visual inspection.

Results

Federated estimates preserve privacy and equal estimates with pooled data

This manuscript provides the DataSHIELD package dsDid enabling causal impact evaluation for CSDID [Callaway and Sant'Anna, 2021, Callaway and Sant'Anna, 2021]. To validate this package, we utilized a simulated dataset generated with the package of Callaway and Sant'Anna [Callaway and Sant'Anna, 2021] and compared our federated implementation against the pooled estimation approach. Specifically, we compared the estimation results in terms of point estimates of the average treatment effect, asymptotic standard errors, and bootstrapped standard errors. Additionally, we evaluated the reduction of uncertainty achieved by the federated model in comparison to the study-level analysis, in which each data owner only analyzes their local dataset.

We simulated a dataset consisting of 801 individuals with observations at four time periods, and individuals randomly allocated across six servers (Figure 4). Three servers contained 134 (536) individuals (observations), and three servers contained 133 (532) individuals (observations). Individuals were either never treated or treated in period two or three. All observations of one individual were stored on one server, reflecting the plausible assumption that two data providers, such as hospitals/schools, will not have data on the same individual. Our main results utilized the DR estimator and not-yet-treated individuals as the control group. Qualitatively equivalent results for the IPW and OR methods and other control groups are presented in the supplementary material.

The federated package successfully reproduced the average treatment effect (Figure 4A) and asymptotic standard errors (Figure 4B) up to numerical precision. For the federated bootstrapped standard errors, the violin plots and boxplots (Figure 4C) indicated that the quantiles of all distributions are visually almost identical and that the densities show the same shapes providing strong evidence that the distributions of the federated and the pooled bootstrapped standard errors are the same.

We also examined the impact of the federated implementation on the estimation of the overall treatment effect compared to study-level analysis. By observing the point estimates and 95% confidence intervals, with standard errors obtained from the multiplier bootstrap (Figure 4E), we found that (i) the study-level point estimates fluctuated around the federated point estimate and (ii) there was increased uncertainty in the study-level DR estimates, reflected in wider confidence intervals. (i) is expected since the pooled simulated data is independent and identically distributed (*i.i.d.*) and we allocated individuals randomly to the servers, so the server data is also *i.i.d*, and thus, the point estimates of the study-level analysis are not biased and fluctuate around the federated estimate. Note, this *i.i.d* case is the most favorable scenario for the study-level analysis and point estimates will deviate more from the federated estimate in non-*i.i.d.* settings. With respect to (ii), the increased size of the confidence intervals reflects the reduced sample sizes, which results in more uncertainty and may lead to the null hypothesis of no treatment effect not being rejected in some study-level cases where it could be rejected in the federated case. The federated setup therefore yield estimates with lower uncertainty enhancing the detection of true treatment effects.

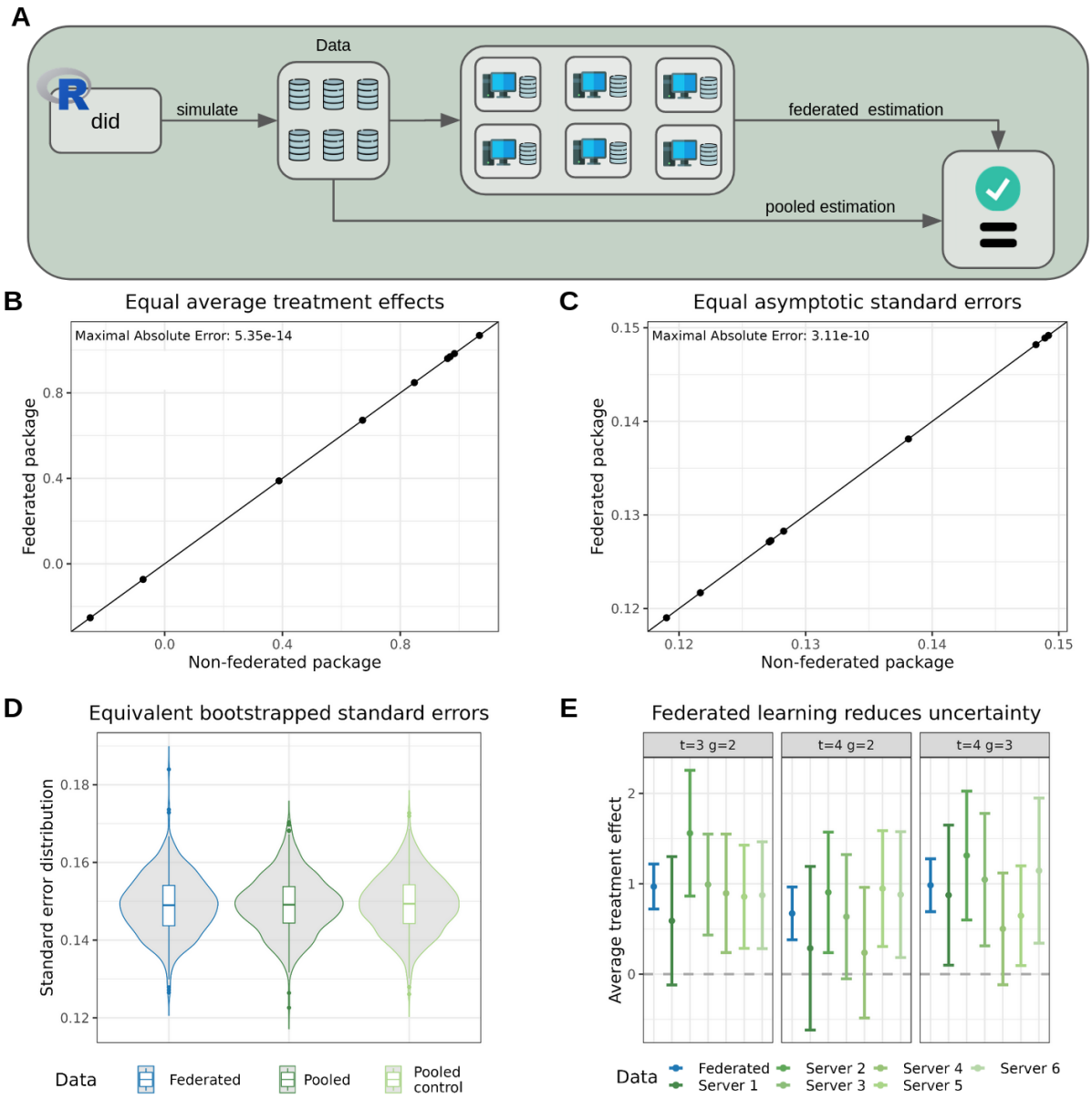

Figure 4: Similarity of federated and pooled estimation for the DR estimate and not-yet treated individuals as control group. **A** Simulation set-up. The federated set-up consisted of 6 servers. Three servers contained 134 (536) individuals (observations), and three servers contained 133 (532) individuals (observations). Individuals were either never treated or treated in period two or three. All observations of one individual were within one server. **B** Depicts the equality of pooled and federated point estimates. The x-axis represents the point estimates of the pooled (non-federated) estimator [Callaway and Sant'Anna, 2021], while the y-axis represents the estimates obtained from our federated approach. The diagonal line depicts the 45° line, indicating that the federated estimate yields results equivalent to the non-federated estimate when they align along this line. **C** Depicts the equality of pooled and federated asymptotic standard errors. The x-axis represents the asymptotic standard errors of the pooled (non-federated) estimator, while the y-axis represents the asymptotic standard errors obtained from our federated approach. The diagonal line again depicts the 45° line. **D** The subplot displays a comparison of the distribution of bootstrapped standard errors using boxplots and densities. The distribution of the federated bootstrapped standard errors is shown in blue, while the distribution of the pooled estimation is shown in green (dark and light). Two pooled distributions were computed to establish a plausible reference difference between two equal distributions. A total of 500 bootstrapped standard errors were computed to obtain the distributions for analysis. **E** The subplot presents point estimates (depicted as dots) and 95% confidence intervals of the estimated treatment effects for the periods following the treatment. The estimates obtained with our federated package are shown in blue, while estimates based on data from only one server are depicted in green. This figure has been designed using resources from Flaticon.com

Discussion

The use of Federated Learning has been recognized as a promising approach to increase sample sizes when dealing with sensitive data. However, the availability of tools for causal analysis within the context of Federated Learning has been limited. To address this gap and make such methods accessible to the research community, we have developed a federated version of the Difference-in-Differences (DID) estimator proposed by Callaway and Sant'Anna (CSDID) [Callaway and Sant'Anna, 2021, Callaway and Sant'Anna, 2021, Rios-Avila et al., 2023] and implemented it in the Federated Learning platform DataSHIELD using the statistical programming language R. Our federated package covers most functionalities of the Callaway and Sant'Anna package [Callaway and Sant'Anna, 2021] and additionally allows for the estimation of treatment effects when treatment and control groups are only mutually exclusively available for single data owners.

We specifically chose the Callaway and Sant'Anna estimator since it gained high relevance after its publication and its flexibility in different settings. This estimator allows for multiple time periods of treatments and treatment evaluations, incorporation of different types of control-groups, and offers various types of estimators and simultaneous confidence bands via bootstrapped standard errors. Moreover, the parallel trend assumption must only hold after conditioning on covariates. This relaxes an assumption of the classical regression DID for which the parallel trend assumption must hold unconditionally as well.

Further work on federated causal impact analysis could include the possibility of estimating treatment effects from repeated cross sections in the context of the CSDID, which is implemented in the package of Callaway and Sant'Anna [Callaway and Sant'Anna, 2021], or synthetic control methods that have gained attraction in recent years [Abadie et al., 2010, 2015].

Conclusions

Our results demonstrate that the proposed federated version of the CSDID estimator can be implemented while preserving the original estimates, asymptotic standard errors, and distributional equivalent bootstrapped standard errors. Additionally, Federated Learning can reduce uncertainty and allow estimation in scenarios in which treated and untreated individuals are separated across data owners. We have ensured that standard DataSHIELD security measures are in place to protect data privacy, and have further enhanced the security level with additional data security measures. In summary, our federated version of the DID estimator based on Callaway and Sant'Anna's methodology fills a gap in the availability of tools for causal analysis in Federated Learning. Our results highlight the potential of using Federated Learning for privacy-preserving causal analysis in settings with sensitive data, while maintaining high levels of data security through standard and additional security measures implemented within DataSHIELD.

Acknowledgements

Not applicable.

Funding

This study was funded by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG) under Germany's Excellence Strategy (EXC 2047 - 390873048 & EXC 2151 - 390685813), the German Ministry for Education and Research (Deutches Bundesminsterium für Bildung und Forschung, BMBF) under the CompLS program (grant agreement No 031L0293C), the University of Bonn (via the Schlegel Professorship of JH), the Helmholtz Association - Munich School for Data Science (MUDS), and the ORCHESTRA project. The ORCHESTRA project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 101016167. The views expressed in this paper are the sole responsibility of the authors and the Commission is not responsible for any use that may be made of the information it contains. The funders had no role in the study design, data collection, data analyses, data interpretation, writing, or submission of this manuscript.

Abbreviations

ATT Average treatment effect of the treated
CSDID Callaway and Sant'Anna Difference-in-Differences
DID Difference-in-Differences
DR Doubly Robust
IPW Inverse probability weighting
RO Regression Outcome
SE Standard error

Availability of data and materials

All code is available at the client-side (https://github.com/manuhuth/dsDidClient) as well as the server-side package (https://github.com/manuhuth/dsDid)

Ethics approval and consent to participate

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

M.H. developed the federation of the algorithms. M.H. and C.G, implemented the software packages in R. M.H. and C.G visualized the results. J.H and E.S. conceptualized the study. M.H., E.S. and J.H. wrote the manuscript. All authors read and approved the final manuscript.

Authors' information

Helmholtz Zentrum München - German Research Center for Environmental Health, Institute of Computational Biology, Neuherberg, Germany
Manuel Huth, Jan Hasenauer

University of Bonn, Life and Medical Sciences Institute, Bonn, Germany
Manuel Huth, Carolina Alvarez Garavito, Jan Hasenauer

ISGlobal, Hospital Clínic - Universitat de Barcelona, Barcelona, Spain
Elisa Sicuri, Friederike Röder, Laia Cirera

LSE Health, Department of Health Policy, London School of Economics and Political Science, London, UK
Elisa Sicuri

References

A. Abadie, A. Diamond, and J. Hainmueller. Synthetic control methods for comparative case studies: Estimating the effect of california's tobacco control program. *Journal of the American statistical Association*, 105(490):493–505, 2010.

A. Abadie, A. Diamond, and J. Hainmueller. Comparative politics and the synthetic control method. *American Journal of Political Science*, 59(2):495–510, 2015.

D. Avraam and S. Wheater. *dsMediation: Methods to apply causal mediation analysis*, 2021. URL `https://github.com/datashield/dsMediation`. R package version 0.0.3.

S. Banerjee, G. N. Sofack, T. Papakonstantinou, D. Avraam, P. Burton, D. Zöller, and T. R. Bishop. dssurvival: Privacy preserving survival models for federated individual patient meta-analysis in DataSHIELD. *BMC Research Notes*, 15(1):197, 2022.

B. Callaway and P. H. Sant'Anna. did: Difference in differences, 2021. URL `https://bcallaway11.github.io/did/`. R package version 2.1.2.

B. Callaway and P. H. Sant'Anna. Difference-in-differences with multiple time periods. *Journal of Econometrics*, 225(2):200–230, 2021.

B. Callaway, A. Goodman-Bacon, and P. H. Sant'Anna. Difference-in-differences with a continuous treatment. *arXiv preprint arXiv:2107.02637*, 2021.

D. Card and A. B. Krueger. Minimum wages and employment: A case study of the fast food industry in New Jersey and Pennsylvania, 1993.

M. A. Colchero, B. M. Popkin, J. A. Rivera, and S. W. Ng. Beverage purchases from stores in mexico under the excise tax on sugar sweetened beverages: observational study. *bmj*, 352, 2016.

I. Dayan, H. R. Roth, A. Zhong, A. Harouni, A. Gentili, A. Z. Abidin, A. Liu, A. B. Costa, B. J. Wood, C.-S. Tsai, et al. Federated learning for predicting clinical outcomes in patients with COVID-19. *Nature Medicine*, 27(10):1735–1743, 2021.

D. Developers. dsbase, 2022. URL `https://github.com/datashield/dsBase.git`. R package.

S. Galiani, P. Gertler, and E. Schargrodsky. Water for life: The impact of the privatization of water services on child mortality. *Journal of political economy*, 113(1):83–120, 2005.

A. Gaye, Y. Marcon, J. Isaeva, P. LaFlamme, A. Turner, E. M. Jones, J. Minion, A. W. Boyd, C. J. Newby, M.-L. Nuotio, et al. Datashield: taking the analysis to the data, not the data to the analysis. *International journal of epidemiology*, 43(6):1929–1944, 2014.

J. R. Gonzalez, Y. Marcon, and X. Esriba-Montagut. *dsOmics: DataSHIELD Omic functions*, 2021. URL `https://github.com/isglobal-brge/dsOmics`. R package version 1.0.18.

A. Goodman-Bacon. Difference-in-differences with variation in treatment timing. *Journal of Econometrics*, 225(2):254–277, 2021.

A. Goodman-Bacon and J. Marcus. Using difference-in-differences to identify causal effects of covid-19 policies. 2020.

J. O. Groeniger, K. Noordzij, J. Van Der Waal, and W. De Koster. Dutch covid-19 lockdown measures increased trust in government and trust in science: A difference-in-differences analysis. *Social Science & Medicine*, 275:113819, 2021.

J. Hansen, P. Wilson, E. Verhoeven, M. Kroneman, M. Kirwan, R. Verheij, and E.-B. van Veen. Assessment of the EU member states' rules on health data in the light of GDPR, 2021. URL `doi:10.2818/546193`.

S. L. Harrison, E. Fazio-Eynullayeva, D. A. Lane, P. Underhill, and G. Y. Lip. Comorbidities associated with mortality in 31,461 adults with covid-19 in the united states: A federated electronic medical record analysis. *PLoS Medicine*, 17(9):e1003321, 2020.

M. Huth. dsdidclient, 2023a. URL `https://github.com/manuhuth/dsDidClient.git`. R package.

M. Huth. dsdid, 2023b. URL `https://github.com/manuhuth/dsDid.git`. R package.

M. Huth, J. Arruda, R. Gusinow, L. Contento, E. Tacconelli, and J. Hasenauer. Accessibility of covariance information creates vulnerability in federated learning frameworks. *Bioinformatics*, 2023.

F. Jannasch, S. Dietrich, T. R. Bishop, M. Pearce, A. Fanidi, G. O'Donoghue, D. O'Gorman, P. Marques-Vidal, P. Vollenweider, M. Bes-Rastrollo, et al. Associations between exploratory dietary patterns and incident type 2 diabetes: A federated meta-analysis of individual participant data from 25 cohort studies. *European Journal of Nutrition*, pages 1–19, 2022.

S. Lenz, M. Hess, and H. Binder. Deep generative models in datashield. *BMC Medical Research Methodology*, 21(1):1–16, 2021.

X. Li, Y. Gu, N. Dvornek, L. H. Staib, P. Ventola, and J. S. Duncan. Multi-site fmri analysis using privacy-preserving federated learning and domain adaptation: Abide results. *Medical Image Analysis*, 65:101765, 2020.

Y. Marcon, T. Bishop, D. Avraam, X. Escriba-Montagut, P. Ryser-Welch, S. Wheater, P. Burton, and J. R. González. Orchestrating privacy-protected big data analyses of data from different resources with R and DataSHIELD. *PLoS Computational Biology*, 17(3):e1008880, 2021.

N. D. Mark and L. L. Wu. More comprehensive sex education reduced teen births: Quasi-experimental evidence. *Proceedings of the National Academy of Sciences*, 119(8):e2113144119, 2022.

B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017.

S. Miller and L. R. Wherry. Health and access to care during the first 2 years of the aca medicaid expansions. *New England Journal of Medicine*, 376(10):947–956, 2017.

OBiBa. Opal, 2022. URL `https://opaldoc.obiba.org/en/latest/`.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2022. URL `https://www.R-project.org/`.

F. Rios-Avila, P. Sant'Anna, and B. Callaway. Csdid: Stata module for the estimation of difference-in-difference models with multiple time periods, 2023. URL `https://EconPapers.repec.org/RePEc:boc:bocode:s458976`.

F. Schulz and J. Rode. Public charging infrastructure and electric vehicles in norway. *Energy Policy*, 160:112660, 2022.

E. Tacconelli, A. Gorska, E. Carrara, R. J. Davis, M. Bonten, A. W. Friedrich, C. Glasner, H. Goossens, J. Hasenauer, J. M. H. Abad, et al. Challenges of data sharing in European COVID-19 projects: A learning opportunity for advancing pandemic preparedness and response. *The Lancet Regional Health-Europe*, 21:100467, 2022.

H. Wen, J. M. Hockenberry, and J. R. Cummings. The effect of medical marijuana laws on adolescent and adult use of marijuana, alcohol, and other substances. *Journal of health economics*, 42:64–80, 2015.

R. Wilson, O. Butters, D. Avraam, J. Baker, J. Tedds, A. Turner, M. Murtagh, and P. Burton. Datashield–new directions and dimensions. *Data Science Journal*, 16, 2017.

Additional information 1

List of DataSHIELD client-side functions

We list all functions that are used within the DataSHIELD framework for the client-side [Huth, 2023a] and explain their functionalities. The main function is `ds.did` which computes treatment effects and standard errors and calls the other client-side functions while being executed.

Federated point estimates

In this section, we present the methodology for computing the point estimates of the average treatment effect of the treated (ATT). To enhance readability, we adopt the notation of Callaway and Sant'Anna [2021]. The choice of control group, either never-treated or not-yet-treated individuals, does not affect the overall methodology and only affects the individuals used to compute the counterfactual. Therefore, for illustration purposes, we showcase the DR algorithm using never-treated individuals as the control group. However, it is important to note that the federation of the outcome

Table 1: New DataSHIELD functions implemented in our software package. Functions starting with *ds.* are client-side functions and can be called via the dsDidClient [Huth, 2023a] package.

Client side function	**Description**
`ds.did`	Calculates the average treatment effect of the treated, standard errors, and pre-tests for the parallel trend assumption.
`ds.addColumnOnes`	Adds a column of ones to a dataframe on the server side.
`ds.appendInfluence`	Appends influence to a specified column for given ids and periods in a dataframe on the server side.
`ds.computeMatrixCrossproduct`	Computes cross product of a matrix on the server side.
`ds.computeOdds`	Computes odds using propensity scores for a given dataset and outcome variable.
`ds.createEmptyIdMatrix`	Creates empty matrix with unique identifiers from a dataframe on the server side.
`ds.enoughIndividuals`	Checks if enough individuals in a dataframe have a specified value in a specified column on the server side.
`ds.generateNotYetTreated`	Generates a dataframe of untreated individuals based on given variables on the server side.
`ds.multiplierBootstrap`	Performs multiplier bootstrap on a given influence matrix on the server side.
`ds.multiplyMatrixMatrix`	Multiplies two matrices and assigns result to a new object on the server side.
`ds.multiplyMatrixScalar`	Multiplies a matrix and a scalar and assigns result to a new object on the server side.
`ds.recode`	Creates a new object and replaces entries of a vector.
`ds.sendToServer`	Sends a single number to the server side.
`ds.subsetDf`	Subsets a dataframe based on a specified variable and value.

regression or the inverse probability weighting are analogous and the algorithms work equivalently for the not-yet-treated individuals as control groups, see Callaway and Sant'Anna [2021] for further details on these methods.

The DR non-federated ATT is given, as described in Callaway and Sant'Anna [2021], by

$$ATT(g,t;\delta) = E^{(1)}\left[\left(\frac{G_g}{E^{(2)}[G_g]} - \frac{\frac{p_g(X)C}{1-p_g(X)}}{E^{(3)}\left[\frac{p_g(X)C}{1-p_g(X)}\right]}\right)(Y_t - Y_{g-\delta-1} - m_{g,t,\delta}(X))\right]. \tag{4}$$

It is defined as the expectation of the difference in treatment effect between the treated and control group, weighted by the probability of receiving treatment. The variables in the equation are defined as follows: g is the time of treatment, t is the time at which the treatment effect is evaluated, and δ is the number of anticipation periods in which individuals know about the upcoming treatment. G_g is a binary variable indicating treatment status in period g, C is a binary variable indicating membership in the never-treated control group, Y_t is the outcome variable affected by the treatment in period t, and X is a matrix of pre-treatment covariates. The term $m_{g,t,\delta}(X) = E\left[Y_t - Y_{g-\delta-1}|X, C=1\right]$ is the expected difference in outcome of the control group and $p_g(X) = E\left[G_g|X, G_g + C = 1\right]$ is the probability of receiving treatment in period g given pre-treatment covariates and treatment status. This equation is shown for fixed values of g, t, and δ for ease of reference. In order to compute the estimate of $ATT(g,t;\delta)$, the expectations are replaced by their federated sample analogues which are denoted by hat variables. To compute the estimate of the average treatment effect (ATT), the expectations in (4) are replaced by their corresponding sample estimates, denoted by hat variables. In order to compute $\widehat{ATT}_{(g,t;\delta)}$, the client must compute $\hat{E}^{(1)}(\cdot)$. This can be done by using a federated mean if all components of the expectation are known on the servers. Initially, the servers have access to individual data such as G_g, C, Y_t, and $Y_{g-\delta-1}$. The unknown quantities $\hat{E}^{(2)}, \hat{E}^{(3)}, \hat{p}_g(X)$, and $\hat{m}_{g,t,\delta}(X)$ can be computed by the client by using federated means and federated generalized linear models (GLMs) in order to estimate the ATT.

In more detail, the computations are

1. $\hat{E}^{(2)}(\cdot)$: The client computes the sample mean of the variable G_g indicating if an individual has been treated in period g using a federated mean (`ds.mean`). The aggregate mean is then sent back to the servers (`ds.SendToServer`) for further analysis and processing.

2. $\hat{p}_g(x)$: The client computes the parameter estimates, denoted by $\hat{\pi}$, of a logistic regression model using a federated generalized linear model (`ds.mean`) with the individuals that have been treated in period g or are never treated

$$p_g(x) = \frac{1}{1 + \exp\left(-\pi_0 - x^T\pi\right)} \tag{5}$$

   The estimated parameters, $\hat{\pi}$, are then sent to the servers where they are immediately processed to compute the estimated probability of treatment, $\hat{p}_g(X)$, using (5) (`ds.genProp`). If no covariates are used, the probability of treatment can be directly computed using a federated mean of the variable indicating treatment G_g (`ds.mean`).

3. $\hat{E}^{(3)}(\cdot)$: With the knowledge of the estimated probability of treatment, $\hat{p}_g(X)$, from step 2 and the binary variable indicating if the individual is part of the never-treated group, C, the mean of the expected value, $\hat{E}^{(3)}(\cdot)$, can be computed using a federated mean (`ds.mean`).

4. $\hat{m}_{g,t,\delta}$: The client computes the parameter estimates $\hat{\beta}$ of a linear regression model using the individuals that have never been treated and the ones that are treated in g. The linear regression model is represented by

$$m_{g,t,\delta}(x) = \beta_0 + x^T\beta, \tag{6}$$

which relates the expected difference in the outcome of the control group and the covariates X. Subsequently, the client sends the estimated parameters $\hat{\beta}$ to the servers, where they are immediately processed to compute $\hat{m}_{g,t,\delta}$ by (6) (`ds.multiplyMatrixMatrix`). If no covariates are used, the client can directly compute $\hat{m}_{g,t,\delta}$ by taking the federated mean of $Y_t - Y_{g-\delta-1}$ (`ds.mean`).

5. $\hat{E}^{(1)}(\cdot)$: Finally, the client computes the expectation $\hat{E}^{(1)}(\cdot)$ by using a federated mean (`ds.mean`) of all the known quantities and parameters. This expectation represents the point estimate of the average treatment effect of the treated $\widehat{ATT}(g,t;\delta)$.

Federated computations of influence functions

The influence function, denoted as $\psi_{g,t,\delta}(W_i, \beta_{g,t,\delta})$, measures the impact that each individual has on the estimated treatment effect $\widehat{ATT}_{t\geq g-\delta}$. The collection of these influence functions $\widehat{ATT}_{t\geq g-\delta}$ is then used to compute (approximate) the variance of $\widehat{ATT}_{t\geq g-\delta}$ through the multiplier bootstrap method or asymptotic standard errors. In this section, we demonstrate the federation of the influence function for the DR estimate using the never-treated individuals as the control group. To maintain consistency with the notation used in Callaway and Sant'Anna [2021], we adhere closely to the notation used in that paper throughout this section. It is also worth noting that, following Callaway and Sant'Anna [2021], the parameters β, π used to estimate $m_{g,t,\delta}(X), p_g(X)$ are included in this section.

The theoretical influence function $\psi_{g,t,\delta}(W_i, \beta_{g,t,\delta})$, as defined in Callaway and Sant'Anna [2021],

$$\psi_{g,t,\delta}(W_i, \beta_{g,t,\delta}) = \psi^{treat}_{g,t,\delta}(W_i, \beta_{g,t,\delta}) - \psi^{comp}_{g,t,\delta}(W_i, \beta_{g,t,\delta}, \pi_g) - \psi^{est}_{g,t,\delta}(W_i, \beta_{g,t,\delta}, \pi_g). \tag{7}$$

quantifies the effect of each individual on the treatment estimate $\widehat{ATT}_{t\geq g-\delta}$. We will show in the following how this influence function can be estimated on the server sides in a federated manner using sample analogues, denoted by hat variables.

The components of the estimated influence function are given by Callaway and Sant'Anna [2021]

$$
\begin{aligned}
\widehat{\psi}^{treat}_{g,t,\delta}(W_i, \beta_{g,t,\delta}) &= \widehat{w_g}^{treat}(\beta_{g,t,\delta})(Y_t - Y_{g-\delta-1} - \hat{m}_{g,t,\delta}(X; \beta_{g,t,\delta})) \\
&\quad - \widehat{w_g}^{treat}(\beta_{g,t,\delta})\hat{E}\left(\widehat{w_g}^{treat}(\beta_{g,t,\delta})(Y_t - Y_{g-\delta-1} - \hat{m}_{g,t,\delta}(X; \beta_{g,t,\delta}))\right) \\
\widehat{\psi}^{comp}_{g,t,\delta}(W_i, \beta_{g,t,\delta}, \pi_g) &= \widehat{w_g}^{comp}(\pi_g)\left[(Y_t - Y_{g-\delta-1} - \hat{m}_{g,t,\delta}(X; \beta_{g,t,\delta})\right) \\
&\quad - \widehat{w_g}^{comp}(\pi_g)\hat{E}(\widehat{w_g}^{comp}(\pi_g)(Y_t - Y_{g-\delta-1} - \hat{m}_{g,t,\delta}(X; \beta_{g,t,\delta}))) \\
\widehat{\psi}^{est}_{g,t}(W_i, \beta_{g,t,\delta}) &= \hat{l}^{or}_{g,t}(\beta_{g,t,\delta})^T\ \hat{M}^1_{g,\delta} + \hat{l}^{ps}_g(\beta_{g,t,\delta})^T\ \hat{M}^2_{g,\delta},
\end{aligned}
\tag{8}
$$

where W_i is the collection of all individual information about Y, X, and C. The weights are defined as $\widehat{w_g}^{treat}(\beta_{g,t,\delta}) = \frac{G_g}{\hat{E}(G_g)}$ and $\widehat{w_g}^{comp}(\pi_g) = \frac{p_g(X;\pi_g)C}{1-p_g(X;\pi_g)} / \hat{E}\left(\frac{p_g(X;\pi_g)C}{1-p_g(X;\pi_g)}\right)$ and the asymptotic linear representations of the regression estimator for the outcome evaluation of the comparison group and the generalised propensity score are

$$
\begin{aligned}
\hat{l}^{or}_{g,t} &= \left[(Y_t - Y_{g-\delta-1} - \hat{m}_{g,t,\delta}(X; \beta_{g,t,\delta})) \circ X\right](X^T X)^{-1} \\
\hat{l}^{ps}_g &= \left[(G_g - p_g(X; \pi_g)) \circ X\right] H^{ps}_g,
\end{aligned}
\tag{9}
$$

where $a \circ B$ defines a columnwise Hadarmad product between the vector $a \in R^n$ and the matrix $B \in R^{n \times k}$. H^{ps}_g is the hessian obtained from the regression on $p_g(X; \pi_g)$.

$$
\begin{aligned}
\hat{M}^1_{g,\delta} &= \hat{E}\left[(\widehat{w_g}^{treat}(\beta_{g,t,\delta}) - \widehat{w_g}^{comp}(\pi_g)) \circ X\right] \\
\hat{M}^2_{g,\delta} &= \hat{E}\left[\widehat{w_g}^{comp}(\pi_g)\,(Y_t - Y_{g-\delta-1} - \hat{m}_{g,t,\delta}(X; \beta_{g,t,\delta})) \circ X\right] \\
&\quad - \hat{E}\left[(\widehat{w_g}^{comp}(\pi_g))^2\,(Y_t - Y_{g-\delta-1} - \hat{m}_{g,t,\delta}(X; \beta_{g,t,\delta})) \circ X\right]
\end{aligned}
\tag{10}
$$

where π_g is the analogue of π in equation (5) and $\beta_{g,t,\delta}$ is the analogue of β in equation (6).

To compute the influence function in a federated manner, the sample analogues of the theoretical influence function must be computed on the client side. Specifically, the client must compute $\widehat{\psi}^{treat}_{g,t,\delta}(W_i, \beta_{g,t,\delta})$, $\widehat{\psi}^{comp}_{g,t,\delta}(W_i, \beta g, t, \delta, \pi_g)$, and $\widehat{\psi}^{est}_{g,t,\delta}(W_i, \beta_{g,t,\delta}, \pi_g)$ using the individual data available on the servers. These sample analogues will be used to estimate the noise-corrupted influence function $\psi^*_{g,t,\delta}(W_i, \beta_{g,t,\delta})$.

1. $\widehat{\psi}^{treat}_{g,t,\delta}(W_i, \beta_{g,t,\delta})$: $Y_t, Y_{g-\delta-1}, \widehat{w_g}^{treat}(\beta_{g,t,\delta})$, and $\hat{m}_{g,t,\delta}(X; \beta_{g,t,\delta})$ are known from the federated point estimate computations on the server side.
   $\widehat{w_g}^{treat}(\beta_{g,t,\delta})(Y_t - Y_{g-\delta-1} - \hat{m}_{g,t,\delta}(X; \beta_{g,t,\delta}))$ can therefore be computed on the client side (`ds.make`). Subsequently, the federated expectation can be returned to the client side (`ds.mean`) and be send back to the server side (`ds.SendToServer`). Finally, $\widehat{\psi}^{treat}_{g,t,\delta}(W_i, \beta_{g,t,\delta})$ can be computed on the server side (`ds.make`).

2. $\widehat{\psi}^{comp}_{g,t,\delta}(W_i, \beta_{g,t,\delta}, \pi_g)$: Analogously to $\widehat{\psi}^{treat}_{g,t,\delta}(W_i, \beta_{g,t,\delta})$ using $\widehat{w_g}^{comp}(\pi_g)$ instead of $\widehat{w_g}^{treat}(\beta_{g,t,\delta})$.

3. $\widehat{\psi}^{est}_{g,t}(W_i, \beta_{g,t,\delta})$: $\hat{l}^{or}_{g,t}, \hat{l}^{ps}_g, \hat{M}^1_{g,\delta}$, and $\hat{M}^2_{g,\delta}$ are unknown to the server side after the federated point estimate computations.

   (a) $\hat{l}^{or}_{g,t}$: $Y_t, Y_{g-\delta-1}, \hat{m}_{g,t,\delta}(X; \beta_{g,t,\delta})$, and X are known from the federated point estimate computations on the server side. $(X^T X)$ can be computed via a federated aggregation

(`ds.computeMatrixCrossproduct`) and be send to the client side, where its inverse $(X^T X)^{-1}$ is computed and send to the servers. Subsequently, $\hat{l}^{or}_{g,t}$ can be computed on the server side (`ds.multiplyMatrixMatrix`).

(b) $\hat{M}^1_{g,\delta}$: $\widehat{w_g}^{treat}(\beta_{g,t,\delta}), \beta_{g,t,\delta}, X$ and $\widehat{w_g}^{comp}(\pi_g))$, are known to the server side from the federated point estimate computations.
Such that $(\widehat{w_g}^{treat}(\beta_{g,t,\delta}) - \widehat{w_g}^{comp}(\pi_g)) \circ X)$ can be computed on the server side (`ds.make`) and the expectation can be estimated on the client side using a federated mean (`ds.mean`). The federated mean is subsequently send back to the servers (`ds.sendToServer`).

(c) $\hat{l}^{ps}_g$: $Y_t, Y_{g-\delta-1}, \hat{m}_{g,t,\delta}(X; \beta_{g,t,\delta})$, and X are known from the federated point estimate computations on the server side such that $G_g - (p_g(X; \pi_g)) \circ X$ can be computed on the server side (`ds.make`). (H^{ps}_g) can be send to the server side where it is immediately multiplied to compute $\hat{l}^{ps}_g$ on the server side (`ds.multiplyMatrixMatrix`).

(d) $\hat{M}^2_{g,\delta}$: $\widehat{w_g}^{comp}(\pi_g), Y_t, Y_{g-\delta-1}, \hat{m}_{g,t,\delta}(X; \beta_{g,t,\delta})$ and X, are known to the server side from the federated point estimate computations.
$\widehat{w_g}^{comp}(\pi_g)\,(Y_t - Y_{g-\delta-1} - \hat{m}_{g,t,\delta}(X; \beta_{g,t,\delta})) \circ X$ and
$(\widehat{w_g}^{comp}(\pi_g))^2\,(Y_t - Y_{g-\delta-1} - \hat{m}_{g,t,\delta}(X; \beta_{g,t,\delta})) \circ X$ can therefore be computed on the server side (`ds.make`). The expectations can be returned to the client side using a federated mean (`ds.mean`). The means are subsequently send back to the servers (`ds.sendToServer`).

$\hat{l}^{or}_{g,t}, \hat{l}^{ps}_g, \hat{M}^1_{g,\delta}$ and $\hat{M}^2_{g,\delta}$ are now known to the server side. $\hat{l}^{or}_{g,t}(\beta_{g,t,\delta})^T\ \hat{M}^1_{g,\delta}$ and $\hat{l}^{ps}_g(\beta_{g,t,\delta})^T\ \hat{M}^2_{g,\delta}$ can therefore be computed on the servers
(`ds.multiplyMatrixMatrix`). Subsequently, $\widehat{\psi}^{est}_{g,t}(W_i, \beta_{g,t,\delta})$ can be computed on the server sides (`ds.make`).

Knowing $\widehat{\psi}^{treat}_{g,t,\delta}(W_i, \beta_{g,t,\delta}), \widehat{\psi}^{comp}_{g,t,\delta}(W_i, \beta_{g,t,\delta}, \pi_g)$ and $\widehat{\psi}^{est}_{g,t,\delta}(W_i, \beta_{g,t,\delta}, \pi_g)$ on the server sides, $\widehat{\psi}^{*}_{g,t,\delta}$ can be computed on the server sides (`ds.make`).

Federated asymptotic variance-covariance matrix

In this subsection, we will demonstrate the estimation of the asymptotic variance-covariance matrix of the average treatment effect's estimates in a federated setting.

The asymptotic distribution of the average treatment effects [Callaway and Sant'Anna, 2021] is defined in Callaway and Sant'Anna [2021] as

$$\sqrt{n}\left(\widehat{ATT}_{t\geq g-\delta} - ATT_{t\geq g-\delta}\right) \overset{d}{\sim} \mathcal{N}\left(0, E\left[\Psi_{t\geq g-\delta}(W)\Psi_{t\geq g-\delta}(W)^T\right]\right), \tag{11}$$

where $ATT_{t\geq g-\delta}$ is the collection of $ATT(g,t;\delta)$ with $t \geq g-\delta$ and $\Psi_{t\geq g-\delta}(W)$ is the collection of relevant influence functions $\psi_{g,t,\delta}$.

In order to estimate $E\left[\Psi_{t\geq g-\delta}(W)\Psi_{t\geq g-\delta}(W)^T\right]$, we use the sample analogue $\hat{\Psi}_{t\geq g-\delta}(W)^T\hat{\Psi}_{t\geq g-\delta}(W)$, where the number of rows in the matrix $\hat{\Psi}_{t\geq g-\delta}(W)$ corresponds to the total number of individuals and each column represents a combination of treatment periods g and treatment evaluation periods

t. The federated matrix product is computed using (`ds.computeMatrixCrossproduct`), which we explain more detail within the section Federated matrix product with its transpose.

Federated multiplier bootstrap

The multiplier bootstrap allows computations of (clustered) standard errors. They define one draw of the multiplier bootstrap as

$$\widehat{ATT}^{(b)}_{t\geq g-\delta} = \widehat{ATT}_{t\geq g-\delta} + E_n\left[V \cdot \hat{\Psi}_{t\geq g-\delta}(W)\right], \tag{12}$$

where $\hat{\Psi}_{t\geq g-\delta}(W)$ is the sample analogue of $\Psi_{t\geq g-\delta}(W)$ and $E_n(a)$ is the expectation over the components of a vector a. V is a random variable with $P(V = 1 - \frac{\sqrt{5}+1}{2}) = \frac{\sqrt{5}+1}{2\sqrt{5}}$ and $P(V = \frac{\sqrt{5}+1}{2}) = 1 - \frac{\sqrt{5}+1}{2\sqrt{5}}$. We subsequently show that the multiplier bootstrap can be federated using an example with s servers. By running the computations on each server (`ds.multiplierBootstrap`) and aggregating the results via a federated mean (`ds.mean`), the client can, due to the randomness of V, not exactly reproduce the results but obtain the same results qualitatively. Let $\hat{\Psi}_{t\geq g-\delta}(W)^{(k)}$ be the estimated influence matrix and $v^{(k)}$ be the vector or observations drawn from V on server k, such that

$$\hat{E}_n\left[V \cdot \hat{\Psi}_{t\geq g-\delta}(W)\right] = \hat{E}_n \begin{pmatrix} v^{(1)} \cdot \hat{\Psi}_{t\geq g-\delta}(W)^{(1)} \\ \vdots \\ v^{(s)} \cdot \hat{\Psi}_{t\geq g-\delta}(W)^{(s)} \end{pmatrix} \tag{13}$$

The multiplier bootstrap can be run on each server (`ds.multiplierBootstrap`) and multiplied by the server side sample size in order to compute each row of the matrix on the right hand-side of (13) on the respective servers. The client subsequently returns an estimate of the expectation using (`ds.mean`). The variance-covariance matrix of the average treatment effect's estimate can finally be estimated with B bootstrap draws $\frac{1}{B-1}\sum_{b=1} \widehat{ATT}^{(b)}_{t\geq g-\delta} \left(\widehat{ATT}^{(b)}{}_{t\geq g-\delta}\right)^T$.

Federated matrix product with its transpose

To compute asymptotic standard errors, the matrix product of the transposed data and itself must be returned to the client side. This allows for the calculation of the asymptotic linear representation, denoted $\hat{l}^{or}_{g,t}$, of the outcome regression in (10). In our package, this is done via the function `ds.computeMatrixCrossproduct`. Let $Z^{(i)} = \begin{pmatrix} z_1^{(i)} & \dots & z_p^{(i)} \end{pmatrix} \in R^{n\times p}$, for $i = 1, 2, \dots, s$, be data matrices of s different servers such that the full data Z is given, without loss of generality, by

$$Z = \begin{pmatrix} z_1^{(1)} & \dots & z_p^{(1)} \\ z_1^{(2)} & \dots & z_p^{(2)} \\ \vdots & \vdots & \vdots \\ z_1^{(s)} & \dots & z_p^{(s)} \end{pmatrix} = \begin{pmatrix} Z^{(1)} \\ Z^{(2)} \\ \vdots \\ Z^{(s)} \end{pmatrix} \tag{14}$$

The matrix product of it's transpose is therefore given by

$$
\begin{aligned}
Z^T Z &= \begin{pmatrix} Z^{(1)} \\ Z^{(2)} \\ \vdots \\ Z^{(s)} \end{pmatrix}^T \begin{pmatrix} Z^{(1)} \\ Z^{(2)} \\ \vdots \\ Z^{(s)} \end{pmatrix} = \begin{pmatrix} Z^{(1)T} Z^{(2)T} & \dots & Z^{(s)T} \end{pmatrix} \begin{pmatrix} Z^{(1)} \\ Z^{(2)} \\ \vdots \\ Z^{(s)} \end{pmatrix} \\
&= \sum_{i=1}^{s} Z^{(i)T} Z^{(i)}.
\end{aligned} \tag{15}
$$

The multiplication of the server-side data with its transposed $Z^{(i)T} Z^{(i)}$ can be computed on each server and returned to the client side if it contains at least as enough different entries as allowed by the data providers via RDataSHIELDS disclosure settings (default is 5). Knowing all $Z^{(i)T} Z^{(i)}$ on the client side, the sum can subsequently be computed on the client side to obtain $Z^T Z$.

Federated column appending of influence functions

While iterating over the treatment periods g and the treatment impact periods t, the data frames are subsetted with respect to the individuals who are either treated in g or in the control group. On the client side, the influence function is computed for these subsetted individuals. To apply the clustered multiplier bootstrap, however, the influence function must be sent back to the server side after each iteration of g and t. Since the individuals change every iteration, simply appending the influences column-wise is not possible. Therefore, we create a data frame filled with zeros and one column indicating the ID of the individuals, which is known to the server. However, appending a full vector, even though the vector is stored on ther server side, poses a data security problem as it allows for the Covariance-Based Attack Algorithm described in Huth et al. [2023]. To address this, the rows of the data frame are randomly shuffled after each iteration.

To enhance understanding, we present a simplified example utilizing 4 individuals, two treatment periods $g = 1, 2$, and two treatment impact periods $t = 3, 4$. We assume that individual id_1 is treated in $g = 1$, individual id_3 is treated in $g = 2$, and individuals id_2 and id_4 are never treated. The initialized data frame $\mathcal{F}_0$ would be given by

$$
\mathcal{F}_0 = \begin{array}{c} \begin{array}{ccccc} \text{id} & \hat{\psi}_{1,3} & \hat{\psi}_{2,3} & \hat{\psi}_{1,4} & \hat{\psi}_{2,4} \end{array} \\ \begin{pmatrix} \text{id}_1 & 0 & 0 & 0 & 0 \\ \text{id}_2 & 0 & 0 & 0 & 0 \\ \text{id}_3 & 0 & 0 & 0 & 0 \\ \text{id}_4 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}. \tag{16}
$$

At the first iteration, the server stores the ID vector $v_{1,3} = \begin{pmatrix} \text{id1} & \text{id2} & \text{id4} \end{pmatrix}^T$ of the individuals used in this run. The server computes the influence vector $w_{1,3} = \begin{pmatrix} 3 & -1 & 2 \end{pmatrix}^T$. On the server, the ID vector and the influence vector are matched component-wise to create the matrix $\begin{pmatrix} v_{1,3} & w_{1,3} \end{pmatrix}$. In

the next step, the relevant IDs in the first column of $\mathcal{F}_0$ are replaced with the corresponding values from $w_{1,3}$, and the matrix is subsequently shuffled row-wise

$$\mathcal{F}_1 = \begin{array}{c} \begin{array}{ccccc} \text{id} & \hat{\psi}_{1,3} & \hat{\psi}_{2,3} & \hat{\psi}_{1,4} & \hat{\psi}_{2,4} \end{array} \\ \begin{pmatrix} \text{id}_4 & 2 & 0 & 0 & 0 \\ \text{id}_2 & -1 & 0 & 0 & 0 \\ \text{id}_3 & 0 & 0 & 0 & 0 \\ \text{id}_1 & 3 & 0 & 0 & 0 \end{pmatrix} \end{array}. \tag{17}$$

This procedure is repeated for all combinations of t and g such that the final data frame is, depending on the random shuffling, given by

$$\mathcal{F}_4 = \begin{array}{c} \begin{array}{ccccc} \text{id} & \hat{\psi}_{1,3} & \hat{\psi}_{2,3} & \hat{\psi}_{1,4} & \hat{\psi}_{2,4} \end{array} \\ \begin{pmatrix} \text{id}_2 & -1 & 1 & 3 & -2 \\ \text{id}_4 & 2 & 4 & 1 & 2 \\ \text{id}_1 & 3 & 0 & 2 & 0 \\ \text{id}_3 & 0 & 2 & 0 & -1 \end{pmatrix} \end{array}. \tag{18}$$

By implementing this method, it is possible to append the influence functions in a privacy-preserving manner.

Additional information 2

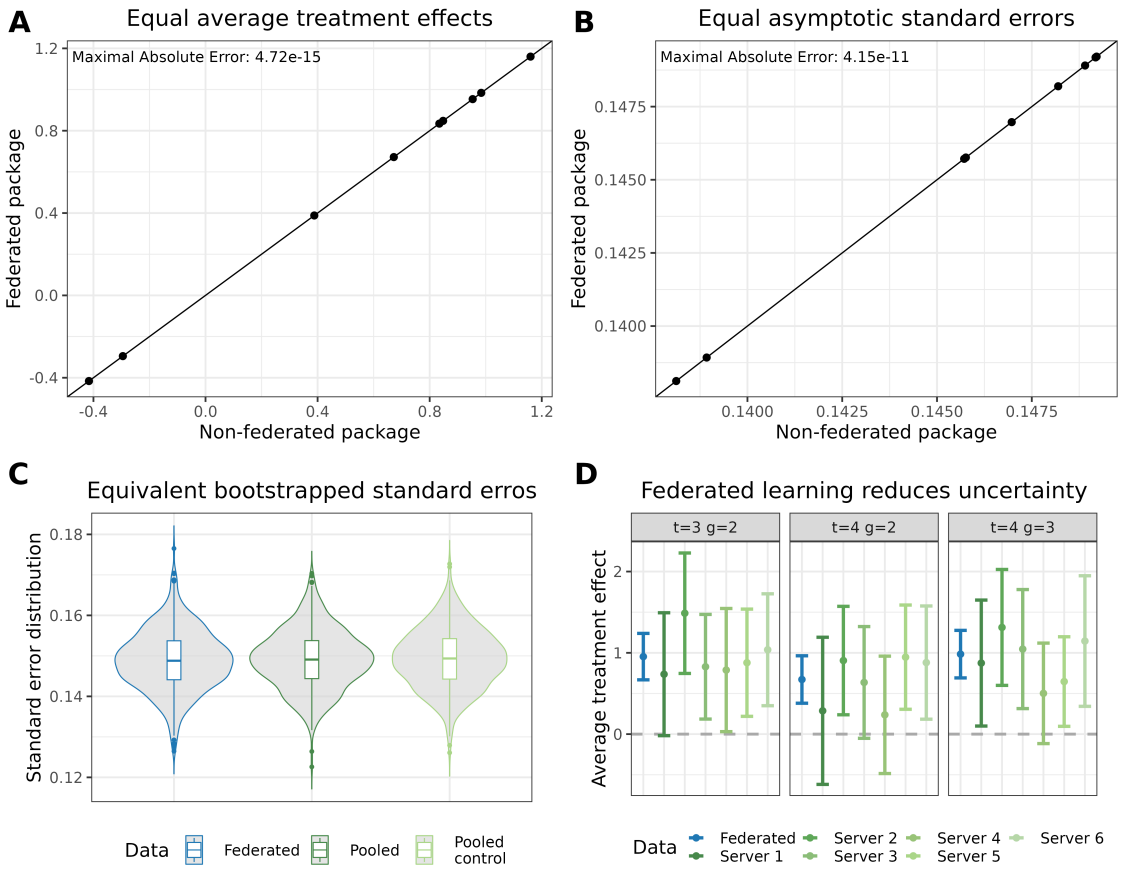

Figure 5: DR estimate and never treated individuals as control group. The federated set-up consisted of 6 servers. Three servers contained 134 (536) individuals (observations), and three servers contained 133 (532) individuals (observations). Individuals were either never treated or treated in period two or three. All observations of one individual were within one server. **A** Depicts the equality of pooled and federated point estimates. The x-axis represents the point estimates of the pooled (non-federated) estimator [Callaway and Sant'Anna, 2021], while the y-axis represents the estimates obtained from our federated approach. The diagonal line depicts the 45° line, indicating that the federated estimate yields results equivalent to the non-federated estimate when they align along this line. **B** Depicts the equality of pooled and federated asymptotic standard errors. The x-axis represents the asymptotic standard errors of the pooled (non-federated) estimator, while the y-axis represents the asymptotic standard errors obtained from our federated approach. The diagonal line again depicts the 45° line. **C** The subplot displays a comparison of the distribution of bootstrapped standard errors using boxplots and densities. The distribution of the federated bootstrapped standard errors is shown in blue, while the distribution of the pooled estimation is shown in green (dark and light). Two pooled distributions were computed to establish a plausible reference difference between two equal distributions. A total of 500 bootstrapped standard errors were computed to obtain the distributions for analysis. **D** The subplot presents point estimates (depicted as dots) and 95% confidence intervals of the estimated treatment effects for the periods following the treatment. The estimates obtained with our federated package are shown in blue, while estimates based on data from only one server are depicted in green. This figure has been designed using resources from Flaticon.com

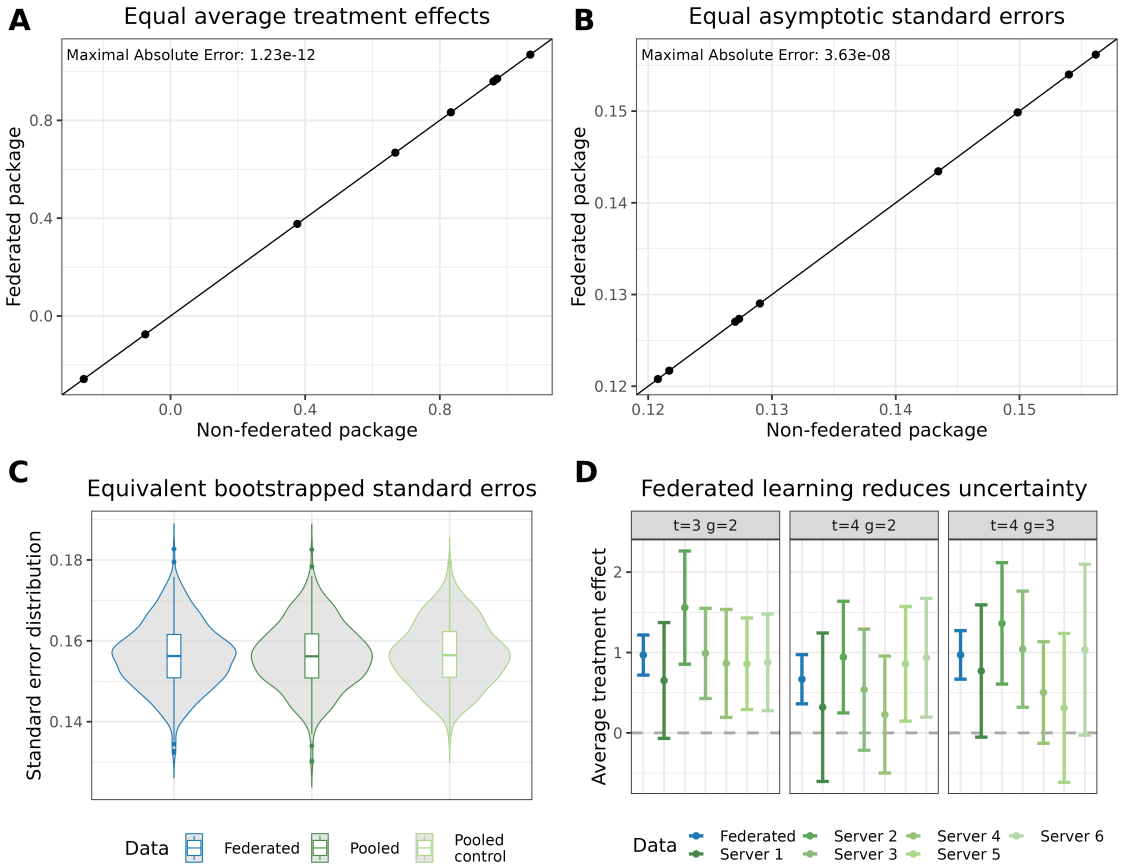

Figure 6: Inverse probability weighted estimate and not-yet treated individuals as control group. The federated set-up consisted of 6 servers. Three servers contained 134 (536) individuals (observations), and three servers contained 133 (532) individuals (observations). Individuals were either never treated or treated in period two or three. All observations of one individual were within one server. **A** Depicts the equality of pooled and federated point estimates. The x-axis represents the point estimates of the pooled (non-federated) estimator [Callaway and Sant'Anna, 2021], while the y-axis represents the estimates obtained from our federated approach. The diagonal line depicts the 45° line, indicating that the federated estimate yields results equivalent to the non-federated estimate when they align along this line. **B** Depicts the equality of pooled and federated asymptotic standard errors. The x-axis represents the asymptotic standard errors of the pooled (non-federated) estimator, while the y-axis represents the asymptotic standard errors obtained from our federated approach. The diagonal line again depicts the 45° line. **C** The subplot displays a comparison of the distribution of bootstrapped standard errors using boxplots and densities. The distribution of the federated bootstrapped standard errors is shown in blue, while the distribution of the pooled estimation is shown in green (dark and light). Two pooled distributions were computed to establish a plausible reference difference between two equal distributions. A total of 500 bootstrapped standard errors were computed to obtain the distributions for analysis. **D** The subplot presents point estimates (depicted as dots) and 95% confidence intervals of the estimated treatment effects for the periods following the treatment. The estimates obtained with our federated package are shown in blue, while estimates based on data from only one server are depicted in green. This figure has been designed using resources from Flaticon.com

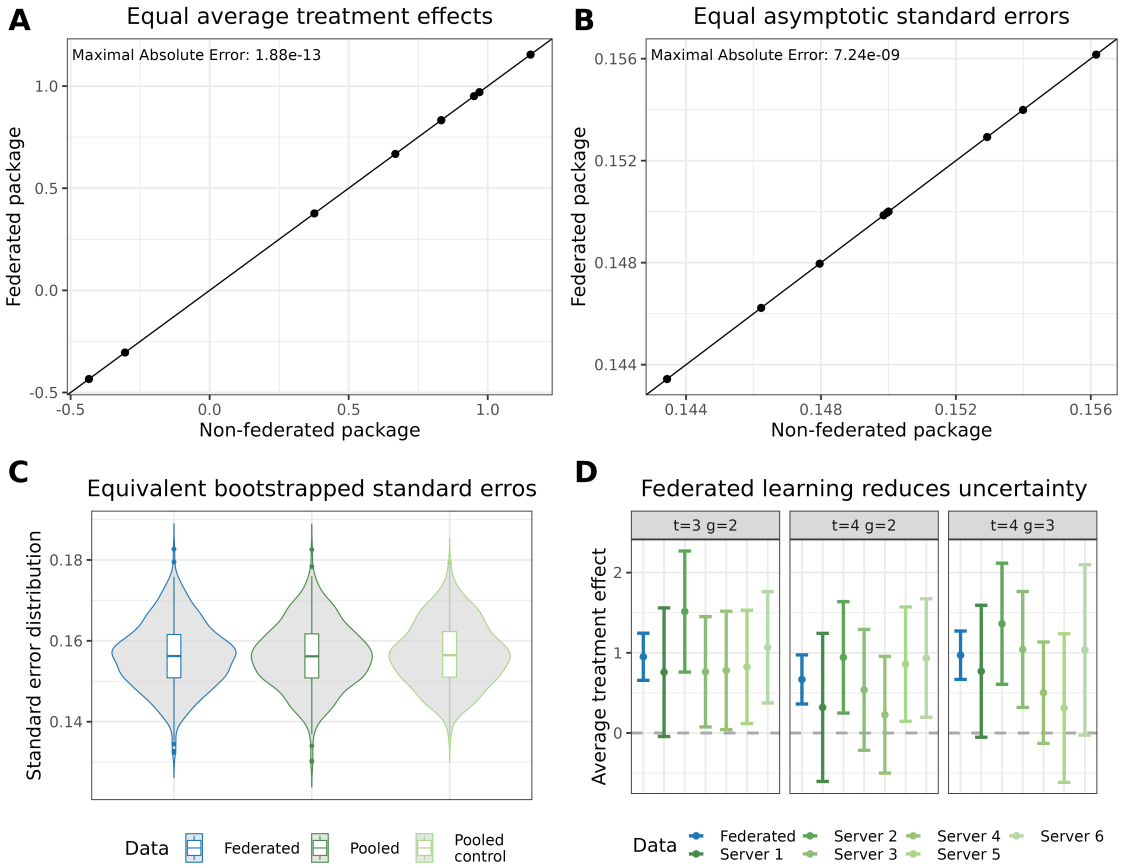

Figure 7: Inverse probability weighted estimate and never treated individuals as control group. The federated set-up consisted of 6 servers. Three servers contained 134 (536) individuals (observations), and three servers contained 133 (532) individuals (observations). Individuals were either never treated or treated in period two or three. All observations of one individual were within one server. **A** Depicts the equality of pooled and federated point estimates. The x-axis represents the point estimates of the pooled (non-federated) estimator [Callaway and Sant'Anna, 2021], while the y-axis represents the estimates obtained from our federated approach. The diagonal line depicts the 45° line, indicating that the federated estimate yields results equivalent to the non-federated estimate when they align along this line. **B** Depicts the equality of pooled and federated asymptotic standard errors. The x-axis represents the asymptotic standard errors of the pooled (non-federated) estimator, while the y-axis represents the asymptotic standard errors obtained from our federated approach. The diagonal line again depicts the 45° line. **C** The subplot displays a comparison of the distribution of bootstrapped standard errors using boxplots and densities. The distribution of the federated bootstrapped standard errors is shown in blue, while the distribution of the pooled estimation is shown in green (dark and light). Two pooled distributions were computed to establish a plausible reference difference between two equal distributions. A total of 500 bootstrapped standard errors were computed to obtain the distributions for analysis. **D** The subplot presents point estimates (depicted as dots) and 95% confidence intervals of the estimated treatment effects for the periods following the treatment. The estimates obtained with our federated package are shown in blue, while estimates based on data from only one server are depicted in green. This figure has been designed using resources from Flaticon.com

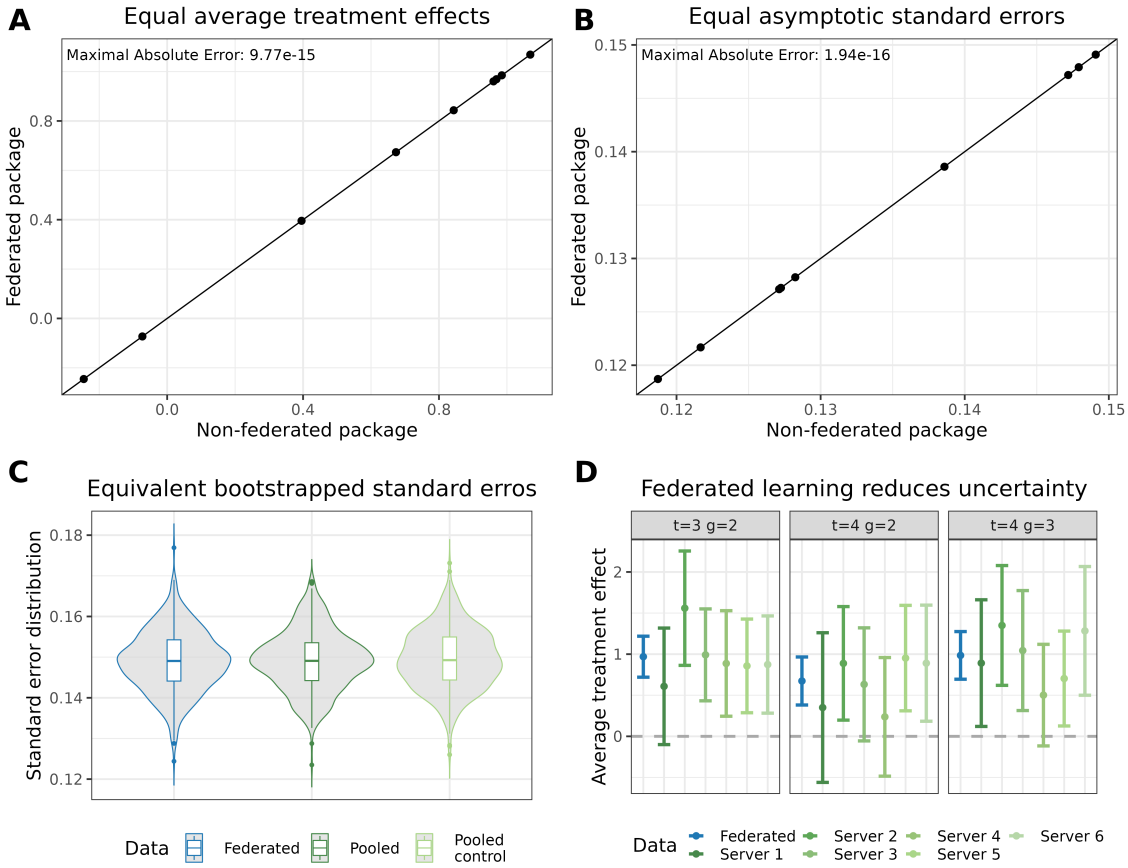

Figure 8: Outcome regression estimate and not-yet treated individuals as control group. The federated set-up consisted of 6 servers. Three servers contained 134 (536) individuals (observations), and three servers contained 133 (532) individuals (observations). Individuals were either never treated or treated in period two or three. All observations of one individual were within one server. **A** Depicts the equality of pooled and federated point estimates. The x-axis represents the point estimates of the pooled (non-federated) estimator [Callaway and Sant'Anna, 2021], while the y-axis represents the estimates obtained from our federated approach. The diagonal line depicts the 45° line, indicating that the federated estimate yields results equivalent to the non-federated estimate when they align along this line. **B** Depicts the equality of pooled and federated asymptotic standard errors. The x-axis represents the asymptotic standard errors of the pooled (non-federated) estimator, while the y-axis represents the asymptotic standard errors obtained from our federated approach. The diagonal line again depicts the 45° line. **C** The subplot displays a comparison of the distribution of bootstrapped standard errors using boxplots and densities. The distribution of the federated bootstrapped standard errors is shown in blue, while the distribution of the pooled estimation is shown in green (dark and light). Two pooled distributions were computed to establish a plausible reference difference between two equal distributions. A total of 500 bootstrapped standard errors were computed to obtain the distributions for analysis. **D** The subplot presents point estimates (depicted as dots) and 95% confidence intervals of the estimated treatment effects for the periods following the treatment. The estimates obtained with our federated package are shown in blue, while estimates based on data from only one server are depicted in green. This figure has been designed using resources from Flaticon.com

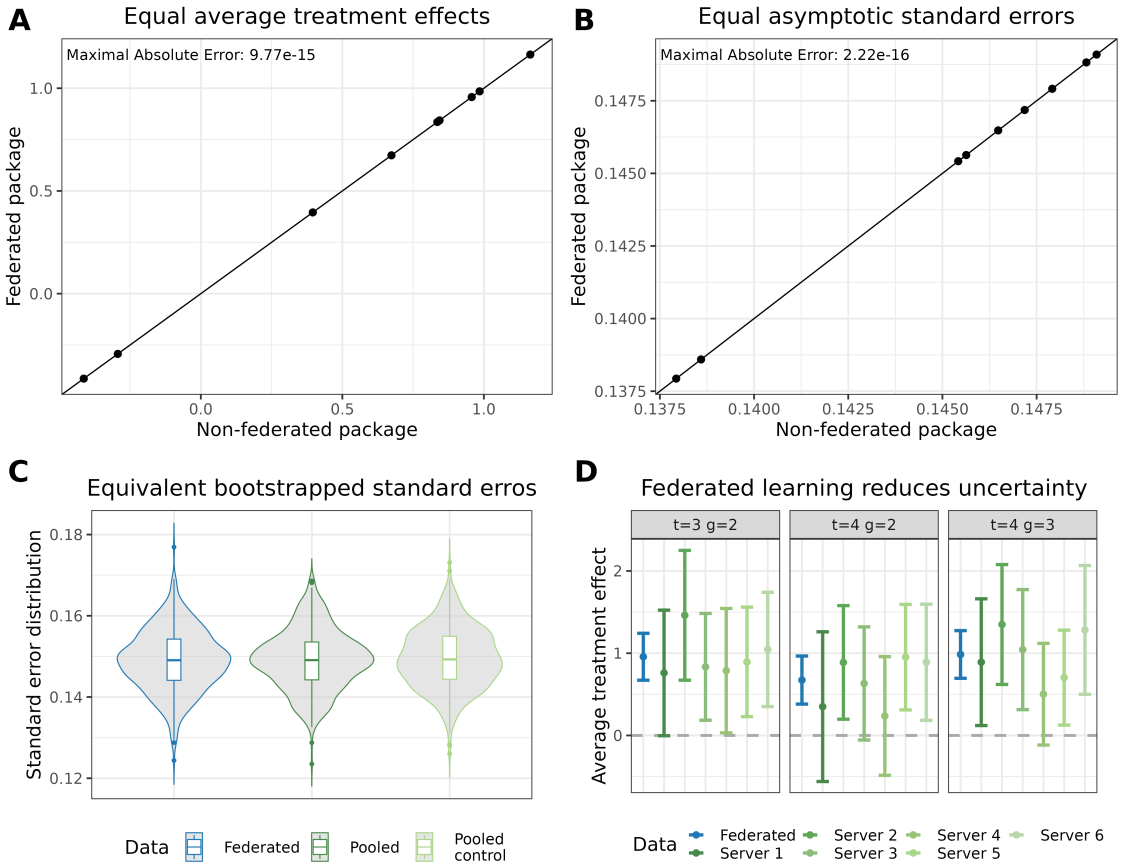

Figure 9: Outcome regression estimate and never treated individuals as control group. The federated set-up consisted of 6 servers. Three servers contained 134 (536) individuals (observations), and three servers contained 133 (532) individuals (observations). Individuals were either never treated or treated in period two or three. All observations of one individual were within one server. **A** Depicts the equality of pooled and federated point estimates. The x-axis represents the point estimates of the pooled (non-federated) estimator [Callaway and Sant'Anna, 2021], while the y-axis represents the estimates obtained from our federated approach. The diagonal line depicts the 45° line, indicating that the federated estimate yields results equivalent to the non-federated estimate when they align along this line. **B** Depicts the equality of pooled and federated asymptotic standard errors. The x-axis represents the asymptotic standard errors of the pooled (non-federated) estimator, while the y-axis represents the asymptotic standard errors obtained from our federated approach. The diagonal line again depicts the 45° line. **C** The subplot displays a comparison of the distribution of bootstrapped standard errors using boxplots and densities. The distribution of the federated bootstrapped standard errors is shown in blue, while the distribution of the pooled estimation is shown in green (dark and light). Two pooled distributions were computed to establish a plausible reference difference between two equal distributions. A total of 500 bootstrapped standard errors were computed to obtain the distributions for analysis. **D** The subplot presents point estimates (depicted as dots) and 95% confidence intervals of the estimated treatment effects for the periods following the treatment. The estimates obtained with our federated package are shown in blue, while estimates based on data from only one server are depicted in green. This figure has been designed using resources from Flaticon.com