Amsterdam School of Economics
Faculty of Economics and Business

de volksbank

MSc Financial Econometrics Thesis

Application Of Generalizability Theory to Construct a Reliability Framework for Machine Learning: An Application to Random Forests

Author: J.W. (Jan Willem) Nijenhuis, BSc
Student ID: 11862386
Supervisor: Dr. Y. (Yi) He
Second reader: Dr. M.P. (Mario) Rothfelder

August 14, 2021

Abstract

The popularity of machine learning in the financial sector is rising. Subtle improvements to disruptive technologies; the range of potential applications is vast. Not much is known however, on the robustness of predictions made by these models. Changes in the parameter configuration can result in models performing equally well, but producing a materially different output. Large financial institutions emphasize the importance of robustness in such 'black box' algorithms, to increase confidence of both its users and targets. This paper introduces a framework to assess model robustness. It does so by providing a method to decompose variance into components that can be attributed to specific parts of the parameter configuration. These components can be utilized to assess which parameters have high impact on predictions, and if this is desirable. A measure of robustness can be constructed from these components, called the generalizability coefficient. This coefficient indicates the proportion of total variance that is due to prediction effects. I show evidence that the framework can produce unbiased and consistent estimates of the variance components, and that the application to random forests is adequate. Since the framework only uses realized predictions as an input, it is not limited to a specific type of algorithm. Hence, it has a wide range of applications. This is to the extent of my knowledge the first application of generalizability theory to machine learning in the literature.

Acknowledgements

I am extremely grateful to Maarten Terpstra for guiding me on this project within de Volksbank, as his original thinking and practical recommendations greatly improved the work in this paper; to Yi He from the University of Amsterdam for providing critical thinking and comments which greatly improved the quality of my work by helping me formalize the practical ideas and focus on the right points of the research; to Laura Seidel for the interesting discussions; to the entire team of Model Validation who made me express my ideas clearly, and provided great comments on the work; to Hannah Oosterhuis and all in the AICoE team for the assistance on the algorithms; to all at the University of Amsterdam that have lectured me for the past four years; to Sijmen Rijks, Jochem Huismans and Frank Nijenhuis for proofreading my thesis and delivering useful comments; and to all at de Volksbank, for giving me the opportunity to expand my skillset by doing this internship. Even though I mainly worked from home, I had a great experience and truly felt part of the bank.

Statement of originality

This document is written by Jan Willem Nijenhuis who declares to take full responsibility for the contents of this document. I declare that the text and the work presented in this document are original and that no sources other than those mentioned in the text and its references have been used in creating it. The Faculty of Economics and Business is responsible solely for the supervision of completion of the work, not for the contents.

Contents

Glossary

decision study In a decision study a researcher uses estimated variance components from a generalizability study to calculate measures of generalizability (robustness).

facet The word *facet* has its origins from the French *facette*, which means *face, side*. A facet is defined as one dimension along which the researcher wishes to decompose variance.

fixed effects design In a fixed effects design the levels of the facet are bounded by conditions set by the researcher. That is, there is a finite number of values which the effect can attain.

generalizability The generalizability of a model or score indicates how well defined a particular score or observation is. By (slightly) changing the circumstances under which the score is observed the generalizability indicates how sensitive the value of this score is to the applied changes.

generalizability study In a generalizability study a researcher decomposes unexplained variance in multiple components attributable to different facets.

grand mean The grand mean is the mean accross all facet scores in the universe of generalization.

object of measurement The object of measurement is the dependent variable a researcher tries to model or predict.

random effects design There is no bound on the values a facet can attain.

reliability Reliability compares the variance between the true outcomes with the variance of the predicted outcomes.

universe of admissible observations The universe of admissible observations contains the observations due to all possible combinations of facet values. A generalizability study tries to find the variance around a certain point in this universe of admissible observations.

universe of generalization The patricular universe of admissible observations which is subject to study in a generalizability study.

universe-score-variance The variance of the object of measurement.

External files used in this paper

In the development of this paper I mainly used Python for the analyses and to build the framework. All results presented in this research, along with the relevant code can be found in the GitHub repository; `https://github.com/janwillemnijenhuis/ML_variance_decomposition`. This section is intended to give a small summary of all files in this repository.

The folder **all_code** contains the relevant code, and the folders **4_mc1**, **4_mc2** and **5_application**; which are respectively the results of the two Monte Carlo experiments and the application. These contain all the data used in the analyses in these sections, as well as resulting plots and tables.

- **calc_res.py**: calculates the statistics of the data generated in the first stage of Monte Carlo simulations; returns an Excel file with the summarizing statistics.
- **dist_pds.py**: performs the simulations in the second stage of Monte Carlo simulations; returns plots of data distribution and saves summarizing statistics in a pickle file; not used in final analysis.
- **experiment_omh.py**: performs the experiments in the application; returns the variance components and generalizability coefficients; this file requires packages from de Volksbank.
- **init_dist_fns.py**: contains the functions used in the second stage of Monte Carlo simulations; used in final analysis.
- **open_cors.py**: plots the heatmaps in Fig. 5 and 6.
- **open_omh.py**: converts the results of the application into a readable Excel format.
- **plot_graphs.py**: returns the plot used in Fig. 1.
- **sim_g.py**: single simulation of the first stage of Monte Carlo simulations; used in Lisa supercomputer to decrease running time; these scripts ran in parallel.
- **simulation_gtheory.py**: performs simulations of the first stage of Monte Carlo simulations; since this took too much time the file above has been created.
- **test_rf.py**: generates data for the second stage of Monte Carlo simulations and prints results; used to determine suitable data generating process.
- **unpack_results.py**: calculates the overlap between experiments in the application.

- **unpack_results1.py**: calculates the resulting statistics of the second stage of Monte Carlo simulations; calculates summarizing statistics or saves histogram in Appendix 8.2.

1 Introduction

The reliability of predictions has been an underrepresented topic in the field of artificial intelligence. As a large portion of industries is experimenting with artificial intelligence, there is a need for a measure to indicate robustness which is linked to the explainability of these models. Margrethe Verstager (Executive Vice-President for a Europe fit for the Digital Age) also stresses the importance of explainability, robustness and trust in artificial intelligence (European Commission, 2021):

> *"On Artificial Intelligence, trust is a must, not nice to have. With these landmark rules, the EU is spearheading the development of new global norms to make sure AI can be trusted. By setting the standards, we can pave the way to ethical technology worldwide and ensure that the EU remains competitive along the way. Future-proof and innovation-friendly, our rules will intervene where strictly needed: when the safety and fundamental rights of EU citizens are at stake."*

In this paper, I introduce a new statistic for measuring the reliability of artificial intelligence algorithms. This coefficient is called the generalizability coefficient and measures how sensitive a model is to changes in its parameters. It is directly derived from generalizability theory where it is used to determine appropriate tests in psychological research. The method determines different sources of variance in predictions and decomposes the total variance in variance components attributed to these sources. These components are then used to determine by how much the total variation in predictions is affected by different sources. This enables a researcher to investigate how robust a model is. An illustration of the robustness of a random forest algorithm is given in Fig. 1.

The main applications of artificial intelligence in the financial sector are fraud detection, customer interaction and client onboarding (DNB, 2019). De Nederlandsche Bank (2019) foresees increased involvement of artificial intelligence in compliance procedures, insurance and risk assessment. Machine learning approaches are often able to detect underlying factors of importance that remain untraceable to the human eye. In customer retention, machine learning can be used to detect customers that are prone to leave the company. During the screening of new credit clients, these algorithms can observe features that are related to a higher probability of default in the future, which can greatly improve the banks' assessment of new clients. Traditionally, these credit risk models consisted either of asset valuation, an actuarial approach or an assessment of the default probability based on econometric models (Crouhy, Galai & Mark, 2000). It is of interest to banks to improve their modeling techniques involved. Because artificial intelligence offers these great improvements to financial modeling I expect an increase in its use in the banking sector. When these models are to be used for credit scoring, the impact on

the life of customers is large as these algorithms might influence the decision to grant them a loan. Therefore, an appropriate framework should exist which can guarantee the soundness of predictions made by these algorithms.

This paper focuses on random forests as defined by Breiman (2001), because it shows better performance in credit scoring compared to classical logistic regression and other machine learning approaches as is shown in studies by, among others, De Brito Filho & Artes (2018), Kruppa et al. (2013) and Brown & Mues (2012). And as credit scoring is one of the core activities of a bank, I expect an increase in the use of such algorithms. An additional benefit of random forests is its intuitiveness in application as the tree structure is relatively easy to understand. Although the workings of the algorithm and decision-making as a result of this are not clear to the user, the structure of this decision-making can be explained with relative ease, since a decision tree is comprehensible. However, the same cannot be said of a random forest. As a random forest is an ensemble of a large number of decision trees, the decision making quickly becomes opaque.

The use of these models poses a new obstacle. Contrary to traditional models the outcomes of random forests are (as its name suggests) random. Using traditional methods for reliability of estimates will therefore not suffice. This creates a need for a framework to correctly assess the reliability of observations, and investigate where variety in results originates. Minimum requirements for reliability are set by testing institutions such as COTAN[1] in the Netherlands. COTAN (Evers et al. (2009)) distinguishes between different applications of a test. For example, a test concerned with decisions on an individual (customer) level requires a higher level of reliability than on group level (e.g. the creditworthiness of a mortgage portfolio). Such thresholds and guidelines are not yet available for machine learning algorithms, but could help regulators and banks in their job of assessing the soundness of these models.

A common approach in modeling a random forest is to optimize for measures of predictive performance. For example, the model can be 'tuned' in order to get the best accuracy in a classification problem, or decrease the mean squared error in a regression setting. But how do we proceed when different parameter configurations perform equally well? Now, the modeller has to make a decision on which model will be the 'true' one. And, how reliable individual predictions made by this algorithm are. The question should be asked what the reliability of a certain configuration of the model is, and how reliable individual predictions of this model are (Bosnic (2009)).

To clarify this with an example I introduce model A and B. Both are random forests, fitted on the same training data (and split) and have an equal accuracy on the test set. The only difference between the two is the number of trees. A has 500 and B has 600 trees. As long as the predictions made by the two models are not materially different, there is no problem. But if the correlation between the predictions of A and B is low, the

[1] Commissie Testaangelegenheden Nederland (Committee on Test Matters in the Netherlands)

question arises how reliable a single prediction made by this model is. Especially when we are dealing with anomaly detection - as in credit scoring or detecting a tumor within a cancer patient - this reliability issue suddenly becomes highly relevant. On group level you might have good results, but the individual mistakes matter most here. When you have two models giving opposite predictions the question not only arises which model to use, but also how many more parameter configurations that give similar results might be out there. This directly links to the concepts of explainability and robustness as described in the beginning of this section.

Due to the difficulty in understanding the process within a random forest, it is also hard to understand when a prediction is stable enough to be used in the real world. It is therefore important to create a method to assess the robustness of predictions and discuss how variation in them can be interpreted. The purpose of this study is to show how we can use generalizability theory to analyse the relative robustness of an optimized model configuration applied to random forests. The applicability of this framework is expected to be high, as the algorithm only uses predictions. The framework is therefore invariant to the type of algorithm. This advantage enables regulators to set thresholds for robustness for machine learning models in general.

The paper is organized as follows. Section 2 summarizes the existing literature. Section 3 describes the methodology used in this paper. In Section 4, a Monte Carlo study is deployed to show unbiasedness and consistency of estimates, as well as an application to random forests. Section 5 contains an application on real-world data (and models) of de Volksbank. Section 6 summarizes the most important results and concludes. Section 7 gives suggestions for future research and possible improvements to this work.

Fig. 1: Illustration of robustness of random forests

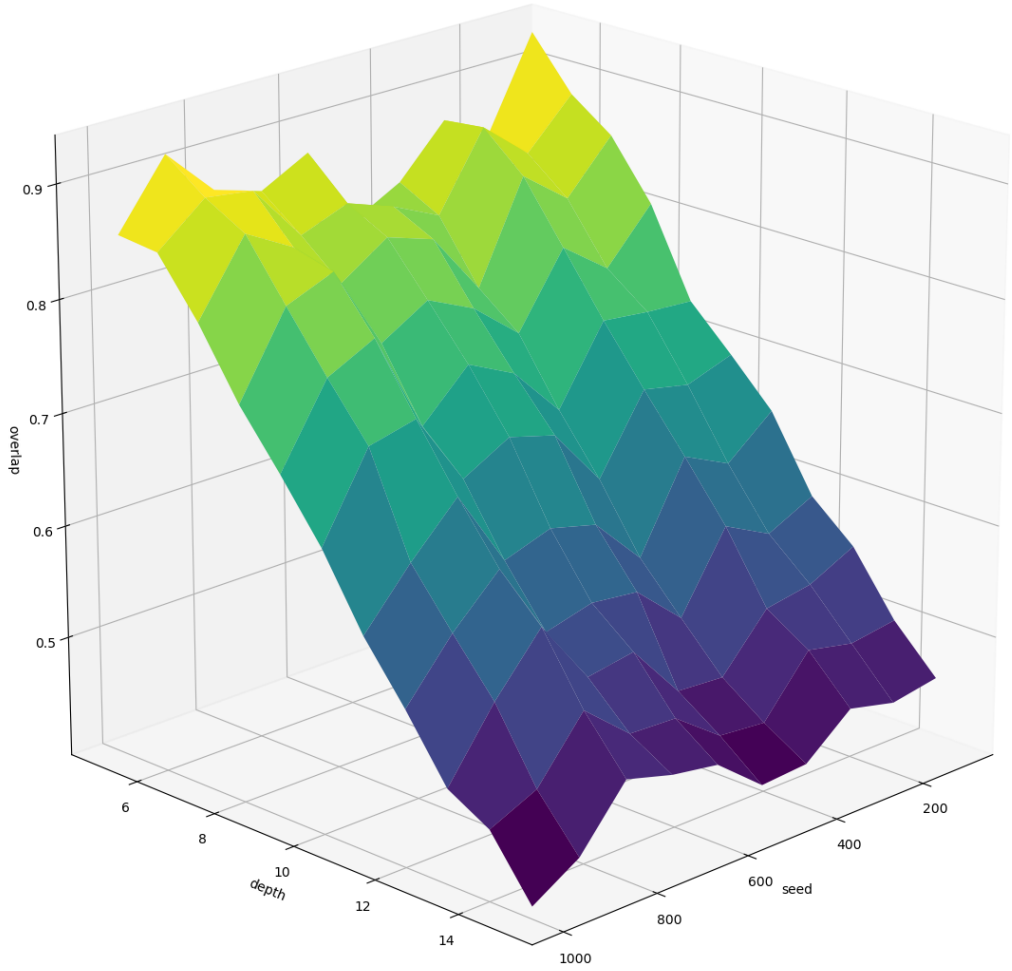


Fig. 1 shows the overlap over the grid of possible combinations of *seeds* and *depths*. The data for this figure is taken from the application in Section 5. The overlap measures what proportion of customers is selected by the model that is also selected by the 'true' model. A high overlap shows high similarity to the 'true' model. The figure shows that the algorithm is relatively robust with respect to the *seed* parameter. Conditional on a value for *depth*, the overlap varies little over different *seeds*. The algorithm is however not robust with respect to the *depth* parameter. The overlap decreases rapidly for an increasing tree depth. A high sensitivity of an algorithm to a particular parameter can be justified by the nature of this parameter. Since the tree depth is 'more fundamental' than the seed, a lower robustness is expected with respect to this parameter.

2 Literature review

This section summarizes the relevant literature of all subdomains that relate to the work in this paper. First, the current state of affairs of artificial intelligence in the financial sector is discussed. This is followed by a description of the existing methods for validating models of this type, and how estimates of reliability can be assessed with developed techniques. The connection is then made to variance decomposition in generalizability theory and how it can benefit the validation process of artificial intelligence algorithms.

2.1 Artificial Intelligence in the financial sector

The application of artificial intelligence algorithms is becoming more mainstream and this is also recognised by major Dutch and European institutions. Standard evaluation metrics for artificial intelligence are not the only requirements when determining the validity of a model; Explainability, simplicity, and reliability of the chosen models are also relevant in the validation of these models (DNB, 2019). The European Commission (2021) defines a risk-based categorization for artificial intelligence applications, including “Unacceptable", “High", “Limited" and “Minimal". They define credit-scoring applications in the “High" risk category, which means that these types of models will be subject to a large range of requirements. Among these requirements are a high level of robustness, security and accuracy. The European Banking Authority (2020) formulated the importance of explainability as follows:

> *“Due to the fact that ML models can quickly become ’black boxes’ - opaque systems for which the internal behaviour cannot be easily understood - which therefore decrease the ability to understand how a certain conclusion is reached. This is directly linked to the opposing concept of explainability.”*

The validation process of artificial intelligence algorithms is also complicated by the fact that the models are subject to continuous improvement. This process (‘tuning’) is a fundamental part of establishing the best model fit. To give an example, the number of variables m selected in each split has a particulary large impact on the predictive ability of a model (Mitchell, 2011). This is commonly referred to as the curse of dimensionality, as it can also cause overfitting. Also, the expected prediction error is usually a monotonically decreasing function of the number of trees T (Probst & Boulesteix, 2018), but they also suggest that more trees are not necessarily better. When such important parameters are tuned, the realisations of the model around the optimum should not give too many opposing predictions, especially when the impact on the life of a customer is large.

Methods exist to evaluate variance in specific algorithms or specific parts of it. For example, the infinitesimal jackknife estimator (Efron, 1992, 2014) is a theoretical mea-

sure estimating variance for a random forest predictor when bagging effects would be mitigated. Wager (2014) produces bias-corrected versions of this method. In his follow-up paper he discussed statistical properties of a random forest, and shows asymptotic normality of predictions made by the algorithm (Wager, 2016). This is a step in establishing the statistical properties of random forests. Still, this approach is strictly limited to the space of random forests, and we wish to define a more uniform framework for assessing reliability in machine learning in general.

2.2 Reliability

When it concerns decisions on the customer level - thus having a large impact on peoples lives - the applications at hand should meet high standards of robustness, reliability and explainability. Thresholds for reliability are advised by COTAN (Evers et al., 2010). There, a distinction is made between what is defined as 'reliable', depending on the application. They state[2]:

> *"High requirements apply to reliability of tests used for important decisions on individual level. The reliability of a certain test is qualified 'inadequate' when the score is lower than 0.80."*

A confidence interval is often constructed as a measure of prediction reliability. Oosterwijk et al. (2019) show that using standard reliability estimates when not appropriate presents us with inconsistent results. To be precise, about 20% of constructed intervals had to be downgraded. That is, the intervals were not as reliable as stated in the research.

Other theories have been proposed, of which Sijtsma (2015) discusses the three methods most widely used in psychometrics with their advantages and shortcomings. These are classical test theory, factor analysis and generalizability theory. Classical test theory decomposes observed variance in true score and error variance. Factor analysis describes variability in observed variance into underlying ('latent') variance components. These components are a linear combination of the variables used in the model. In generalizability theory one looks at variance in different components of the model or research setup. These methods use different definitions in theory, but all look at proportions of true/wanted variance and error/unwanted variance as a measure of how well model predictions generalize to the true scores. In a broader context, reliability can be seen as a study of consistency in predicted outcomes.

[2] Translated from Dutch

2.3 Reliability estimates for artificial intelligence

Scoring of reliability in machine learning predictions has seen a number of proposals. Bosnic & Kononenko (2008a) notice that the accuracy of a model does not directly indicate the reliability of a single prediction. They use a sensitivity analysis to assess the changes in a prediction by small alterations in the input (training) set. Sensitivity analysis for machine learning as introduced by Bousquet & Elliseeff (2001, 2002) aims to understand the dynamics of a model with respect to changes in its parameters. They also provide a bound on the generalization error of the algorithm as a whole for changes in the learning data. The generalization error of a machine learning or statistical learning algorithm is the error when the algorithm is used to predict out of sample data. The measures developed in this work - local variance, local absolute variance and local bias - correlate better with prediction error than common density estimates. A similar method using this transductive approach was already introduced by Kukar & Kononenko (2002), where they use it to construct a measure of reliability on the interval $[0, 1]$.

In their follow-up paper Bosnic & Kononenko (2008b) compare this sensitivity analysis approach with the variance of bagged models, local cross-validation, density estimation, and local modeling. The performance of each approach depends on the algorithm and dataset. A uniform best method cannot be given. They find that a combination of these methods performs best on average. They notice the difficulty of interpretation of reliability estimates, as not much work has been done in this subdomain (Bosnic & Kononenko, 2009). In further research Bosnic & Kononenko (2010) favor the internal cross-validation approach over the previously mentioned methods. The assessment is based on the level of positive correlation with the prediction error. These estimates are useful in determining the reliability of single predictions with respect to alterations in the training set. However, they do not indicate the robustness of the algorithm as a whole.

An entirely different approach is given by Virani, Iyer & Yang (2020) who investigate reliability based on Plato's classical Justified True Belief theory. It uses the location of classifications in output space to see if they are 'grouped together'. They find that the estimate provides good information on the location of training data points. The measure is able to characterize reliability in predictions and point out regions of uncertainty. However, the algorithm has some shortcomings, one of them being its computational complexity. It is also not easy to directly extend this technique to different algorithms.

2.4 Variance decomposition

The main framework used in this paper is generalizability theory for variance decomposition. In this framework, variance components are estimated based on an analysis of variance approach. The notion of generalizability can be traced back to Cronbach et al.

(1963). It builds on the framework of classical test theory (Novick, 1966) and analysis of variance as introduced first by Fisher (1925). Cronbach et al. (1972) developed on the basis of this framework the first measure of 'generalizability'. Here, every observed score is part of a universe, called the universe of generalization and the measure of generalizability covers the ability of this score to generalize to this universe. In other words, by slightly changing the circumstances under which a score is observed, this theory investigates how sensitive the test result or prediction is to these changes. This notion of generalizability replaces the concept of reliability. The theory liberalizes the idea of true and error variance and enables the researcher to decompose the total variance into components that can be attributed to specific changes (effects) of an experiment. These components can be determined by the researcher. Each dimension in which a researcher wishes to decompose is called a *facet*. As an example, in random forests one could consider a tuning parameter a facet, and therefore investigate the sensitivity of predictions to this parameter.

Shavelson & Webb (1981) provide an overview of advancements in the theory over the period 1973-1980. They extend this work (Shavelson & Webb, 1989,1991,2005) by summarizing new developments in the theory. Brennan (1997, 2001) provides an extensive overview of the theory as well. In his later work he discusses the advantages and disadvantages of generalizability theory in comparison with classical test theory (Brennan, 2010). He covers the mathematical background of the algorithm and simulation methods to show its statistical properties and applications in his book (Brennan, 2001). As the theory only deals with outcomes of an experiment it can be extended to other fields with relative ease.

Other methods than the analysis of variance framework have been suggested to decompose variance components in generalizability theory. Nelder (1968) designed a method to decompose variance of treatment effects in blocked designs. He uses maximum likelihood to estimate these components. Patterson & Thompson (1971) extend this work by introducing a restricted maximum likelihood-based method for unequal block sizes. Asymptotic properties for this type of estimators as proven by Miller (1973). He shows that the estimator is consistent, asymptotically efficient and above all asymptotically equivalent to the analysis of variance estimates. Harville (1977) compares the restricted maximum likelihood-based approach with Bayesian methods and analysis of variance. He concludes that all methods provide consistent results, but it should be determined by the type of problem which technology to use. The restricted maximum likelihood-based method is extended by the work of Johnson & Thompson (1994) with an application in Gilmour et al. (1995). By using a Newton-Rhapson method they decrease the computational cost and therefore the time required to run the algorithm. Jennrich & Sampson (1976) construct a method with similar properties based on Newton-Rhapson. Aitkin (1999) makes use of the same techniques in generalized linear models.

3 Methodology

This section summarizes the methodology used to conduct the research in this paper. The mathematical framework regarding generalizability theory is explained in Section 3.1. This contains an application to 2 and 3 *facets*, which are the dimensions along which a researcher wishes to decompose variance. This can be a certain aspect in a test - such as a different test setting - but also a parameter of an algorithm in e.g. random forests. The obtained variance components can be applied in what is called a decision study. This is the second part of generalizability theory, in which the variance components are used to create measures of generalizability. The theoretical framework underlying this is described in Section 3.2. It is called a decision study, as psychologists use this to determine adequate testing requirements (hence making *decisions*). The framework will be applied to random forests. Therefore, Section 3.3 will cover a basic theoretical description of random forests. Section 3.4 gives an interpretation of the coefficients developed in Section 3.3.

In order to keep things simple and concise I will introduce some notation here. In the application in Section 5 I investigate the sensitivity of random forests to different *seeds* and *depths*. *Seeds* here refers to the different values which the random state can attain. We thus look at the effect of different randomness in data and the model on predictions. The *depth* is the number of 'layers' in a decision tree. A large depth usually results in overfitting, whereas a depth which is too low has a bad predictive ability. Hence, instead of speaking about *facet 1* and *facet 2* - which would be more general - we use *seed* and *depth* to refer to the different facets. Note that they can be used interchangeably, but for ease of notation we stick to *seed* and *depth*.

The theory is explained for 2 and 3 facets since this is what is used in the application in Section 5. Appendix 8.1.2 provides a general framework for m facets to give insight in the mathematics for larger *designs*.

3.1 Variance decomposition according to generalizability theory

Generalizability theory has its origins in psychometrics, the field of study concerned with techniques of measurement in psychological research. It is a statistical theory developed to measure reliability of (psychological) tests, based on techniques from classical test theory and analysis of variance. In classical test theory a researcher is concerned with minimizing the error variance, whereas analysis of variance deals with variance between groups. In generalizability theory one is not necessarily concerned with reducing the unexplained (error) variance or comparing groups, but it seeks to combine the two. It provides a framework to decompose the variance into variance components related to different dimensions, called *facets*. The purpose of a generalizability study is to obtain estimates of variance components associated with a particular scoring moment. This can

be different testers or raters as often quoted in behavioural research, but it has a broad range of applications. In the context of machine learning these facets can be taken to be different sets of parameter configurations that one wishes to determine the reliability of model scores on. As mentioned in the beginning of this section we will stick to *seeds* and *depths*.

Consider the true score y which we believe is generated according to the following process:

$$y = f(x) + \varepsilon \tag{1}$$

Hence, there is a function $f(x)$ is the deterministic part of the process which we wish to approximate the behaviour of y with, and some irreducible error ε. We assume that $\mathbb{E}[\varepsilon] = 0$ and $\mathbb{V}[\varepsilon] = \sigma^2(\varepsilon)$. If we have an unbiased and consistent estimator $\hat{f}(x)$ of $f(x)$, we can formulate reliability according to classical test theory as:

$$\rho(y, \hat{f}(x)) = \frac{\sigma^2(\hat{f}(x))}{\sigma^2(y)} = \frac{\sigma^2(\hat{f}(x))}{\sigma^2(\hat{f}(x)) + \sigma^2(\varepsilon)} \tag{2}$$

Which measures the proportion of prediction score variance with respect to the true score variance. The objective of a generalizability study is to investigate the variance component $\sigma^2(\hat{f}(x))$. This component can be split in variance components which are related to different *score effects*. A single observed prediction (*score*) is then the sum of the mean over all predictions, plus its prediction-specific *score effects*. These variance components can subsequently be used to form a measure of generalizability similar to the one in Eq. 2, but now with respect to $\sigma^2(\hat{f}(x))$.

For a particular instance of the model - say with seed (random state) s - the estimate $\hat{f}_s(x)$ is defined as:

$$\hat{f}_s(x) = \mu + \nu_p(x) + \nu_s(s) + \nu_{ps}(x, s) \tag{3}$$

The *grand mean* μ refers to the mean over all predictions and facets of this algorithm. The *prediction effect* $\nu_p(x)$ is the change from this *grand mean* due to a different value for x. The estimator $f(x)$ can be written as:

$$f(x) = \mu + \nu_p(x) \tag{4}$$

since it does not depend on *seeds*. The *seed effect* $\nu_s(s)$ is the effect on the prediction because of a particular seed, and the *interaction effect* $\nu_{ps}(x, s)$ can be seen as the changes in the prediction effect over different seeds. The expectation of these terms over all seeds

is zero, that is[3]:

$$\begin{aligned} \mathbb{E}_s[\nu_s] &= 0 \\ \mathbb{E}_s[\nu_{ps}] &= 0 \end{aligned} \tag{5}$$

Hence, taking the average over all *seeds*:

$$\begin{aligned} \mathbb{E}[\frac{1}{n_s}\sum_s \hat{f}_s(x)] &= \mathbb{E}[\frac{1}{n_s}\sum_s (\mu + \nu_p(x) + \nu_s(s) + \nu_{ps}(x,s))] \\ &= \sum_s (\mu + \nu_p(x)) = f(x) \end{aligned} \tag{6}$$

In this notation a score's usefulness, largely depends on the extent to which it allows us to generalize the realisations $\hat{f}_s(x)$ accurately to the behaviour of the estimator $\hat{f}(x)$. Using observed model scores a generalizability study determines dependence on the facets. Subsequently, a decision study can give perspective on the influence of this dependence on the predictions made by the algorithm.

The following subsections describe an application of generalizability theory to random forests, but note that it could be applied to any set of model scores. First, a 2-facet generalizability study is explained. Then we show a 3-facet design. Additionally, as mentioned in the beginning of this section an extension to a m-facet design is made in 8.1.2 if the reader is interested in an application to more facets. Note here, that for ease of notation the object of measurement is considered a facet as well. Officially, it should be called a 1-facet-1-object-of-measurement design, but as one can see this pollutes the notation quickly and since the aim of this framework is simplicity in the first place I will stick to writing it as a facet. The notation developed in the following subsections is largely based on the theory developed by Cronbach (1972), Shavelson (1989,1991) and Brennan (2001).

3.1.1 Applied generalizability study

Define a random forest configuration and a fixed dataset D_n. Assume we already optimized our parameter specification to standard measures such as accuracy, precision and recall. We are interested in variability in the predictions made by different model configurations (i.e. parameter tuning). Therefore, we construct what we call a $p \times s$ design, consisting of possible values for each facet (*prediction* and *seed*, respectively). The possible values for the seed facet are determined by the researcher, but should be relevant for the problem. For example, if your optimized model consists of 500 decision trees, investigating 5 or 5000 trees is not reasonable, but 600 is.

To expand on this, you would want to explore variance around the optimum of your

[3] See Brennan (2001), p. 23 Eq. 2.6 for more details

model. The objective is to compare variance among predictions (which is the desirable variance) with variance among seeds (which is disliked). For each possible combination of values defined in the $p \times s$ design a random forest is fitted and predictions y_{ps}[4] are made on the same data x. This is necessary, as one otherwise compares different predictions with each other. The results of each combination of facet values are stored in a $n_p \times n_s$ matrix of results y where n_p, n_s are the lengths of the respective facets. In the beginning of Section 3.1 we write a prediction by a random forest as $\hat{f}_s(x)$. Here, we define $y_{ps} := \hat{f}_s(x_p)$. That is, we observe prediction with index p for seed with index s. We can decompose this score using the same methodology as before. In the second line the *score effects* are replaced by their theoretical values (i.e. deviation from the population means[5]:

$$\begin{aligned} y_{ps} &= \mu + \nu_p(x) + \nu_s(s) + \nu_{ps}(x, s) \\ &= \mu + (\mu_p - \mu) + (\mu_s - \mu) + (y_{ps} - \mu_p - \mu_s + \mu) \end{aligned} \tag{7}$$

We have μ_i for the mean over all other facets. That is, $\mu_p = \mathbb{E}_s[y_{ps}]$, $\mu_s = \mathbb{E}_p[y_{ps}]$ and $\mu = \mathbb{E}_{ps}[y_{ps}]$. The indices of $\mathbb{E}$ here refer to over which facet the expectation is taken. The assumption of uncorrelated score effects is justified due to the nature of the facets[6]. The population means can be replaced by their sample equivalents $\bar{y}_p = \frac{1}{n_s}\sum_s^{n_s} y_{ps}$, $\bar{y}_s = \frac{1}{n_p}\sum_p^{n_p} y_{ps}$ and $\bar{y} = \frac{1}{n_s n_p}\sum_p^{n_p}\sum_s^{n_s} y_{ps}$. Then, rewriting Equation 7:

$$y_{ps} - \bar{y} = (\bar{y}_p - \bar{y}) + (\bar{y}_s - \bar{y}) + (y_{ps} - \bar{y}_p - \bar{y}_s + \bar{y}) \tag{8}$$

We can now use this to compute the total sum of squares (TSS):

$$\begin{aligned} TSS = \sum_s^{n_p}\sum_s^{n_s}(y_{ps} - \bar{y})^2 &= \sum_p^{n_p}\sum_s^{n_s}((\bar{y}_p - \bar{y}) + (\bar{y}_s - \bar{y}) + (y_{ps} - \bar{y}_p - \bar{y}_s + \bar{y}))^2 \\ &= n_s \sum_p^{n_p}(\bar{y}_p - \bar{y})^2 + n_p \sum_s^{n_s}(\bar{y}_s - \bar{y})^2 + \sum_p^{n_p}\sum_s^{n_s}(y_{ps} - \bar{y}_p - \bar{y}_s + \bar{y})^2 \\ &= SS(p) + SS(s) + SS(ps) \end{aligned} \tag{9}$$

Note that the crossed terms disappear. An extensive proof of this disappearance of cross terms in Equation 9 is given in Appendix 8.1.1. We then divide the sum of squares by its degrees of freedom. In the notation developed by Brennan (2001) this is called the

[4] Here p, s are the indices of the respective prediction and seed
[5] See Brennan (2001), p. 22-23
[6] see Brennan (2001), p. 66 Eq. 3.7

mean square (MS). This results in:

$$MS(p) = \frac{n_s}{n_p - 1} \sum_{p}^{n_p} (\bar{y}_p - \bar{y})^2 = \frac{SS(p)}{n_p - 1} \tag{10}$$

$$MS(s) = \frac{n_p}{n_s - 1} \sum_{s}^{n_s} (\bar{y}_s - \bar{y})^2 = \frac{SS(s)}{n_s - 1} \tag{11}$$

$$MS(ps) = \frac{1}{(n_p - 1)(n_s - 1)} \sum_{p}^{n_p} \sum_{s}^{n_s} (y_{ps} - \bar{y}_p - \bar{y}_s + \bar{y})^2 = \frac{SS(ps)}{(n_p - 1)(n_s - 1)} \tag{12}$$

We define the variance components in line with the analysis of variance framework as:

$$\sigma^2(p) = \frac{MS(p) - MS(ps)}{n_s} \tag{13}$$

$$\sigma^2(s) = \frac{MS(s) - MS(ps)}{n_p} \tag{14}$$

$$\sigma^2(ps) = MS(ps) \tag{15}$$

They represent the amount of variance over the entire facet. The estimates can be used for the analysis of the influence of different facets and in the computation of measures of generalizability. An expansion on these decision studies is given in 3.2. The following four paragraphs give insight in the formal derivation of variance components, possible negative variance estimates, the computational cost and a generalizability study with 3 facets.

Formal derivation of variance components in a random forest algorithm

This paragraph contains technical details about random forests. It might be beneficial to read Section 3.3 for familiarity first.

Consider a decision tree estimator $m_t(x|D_n, \Theta_t, \psi) = \sum_{X_i \in D_n} \frac{\mathbb{I}_{[X_i \in A_k]} Y_i}{N_k}$ where A_k is a leaf of the decision tree (i.e. a final node) and N_k the number of $X_i \in A_k$. The random forest $M_T(x|D_n, \Theta, \psi) = \frac{1}{T} \sum_{t=1}^{T} m_t(x|D_n, \Theta_i, \psi)$. Here, D_n is the training set and $\Theta = \{\Theta_1, ..., \Theta_T\}$ the parameter application of parameters ψ in tree t. For example, if we have selected a number $m = 5$ variables to use for splits in each tree, Θ_t contains which variables are selected in this particular split. We consider a 2-facet design (hence 3 score effects): a person effect $\nu_p(x)$, a seed effect $\nu_s(s)$ and an interaction term $\nu_{ps}(x, s)$. These effects are explicitly defined in Section 3.1. We have a vector of datapoints x (where $x \notin D$).

Using the notation developed in this section we set set $\hat{f}_s := \hat{f}_s(x, D_n) = M_T(x|D_n, \Theta, \psi(s = s))$. That is, our estimator is the random forest, and depends on the training set D_n and

vector of datapoints x. The relevant facet is in this case the seed s. We define:

$$\begin{aligned} \hat{f} &:= \hat{f}(x, D_n) = \mathbb{E}[\hat{f}_s | D_n] \\ \hat{\varepsilon} &:= \hat{f}_s - \hat{f} \end{aligned} \tag{16}$$

The estimators can be decomposed into score effects as defined in Eq. 3 and 4:

$$\begin{aligned} \hat{f}_s &= \mu + \nu_p(x) + \nu_s(s) + \nu_{ps}(x, s) \\ \hat{f} &= \mu + \nu_p(x) \end{aligned} \tag{17}$$

By the law of total variance:

$$\mathbb{V}[\hat{f}_s] = \mathbb{V}[\mathbb{E}[\hat{f}_s | D_n]] + \mathbb{E}[\mathbb{V}[\hat{f}_s | D_n]] = \mathbb{V}[\hat{f}] + \mathbb{E}[\mathbb{V}[\hat{f}_s | D_n]] \tag{18}$$

Because $Cov(\hat{f}, \hat{\varepsilon}) = 0$ we have:

$$\mathbb{V}[\hat{f}_s | D_n] = \mathbb{V}[\hat{f} + \hat{\varepsilon} | D_n] = \mathbb{V}[\hat{f} | D_n] + \mathbb{V}[\hat{\varepsilon} | D_n] \tag{19}$$

And by the definition of variance:

$$\begin{aligned} \mathbb{V}[\hat{f} | D_n] = \mathbb{V}[\hat{f} | D_n] &= \mathbb{E}[\hat{f}^2 | D_n] - \mathbb{E}[\hat{f} | D_n]^2 \\ &= \hat{f}^2 - \hat{f}^2 = 0 \end{aligned} \tag{20}$$

And:

$$\begin{aligned} \mathbb{V}[\hat{\varepsilon} | D_n] &= \mathbb{V}[\hat{f}_s - \hat{f} | D_n] = \mathbb{V}[\mu + \nu_p(x) + \nu_s(s) + \nu_{ps}(x, s) - \mu - \nu_p(x) | D_n] \\ &= \mathbb{V}[\nu_s(s) + \nu_{ps}(x, s) | D_n] = \mathbb{V}[\nu_s(s) | D_n] + \mathbb{V}[\nu_{ps}(x, s) | D_n] \\ &= \sigma_s^2(s) + \sigma_{ps}^2(x, s) =: \sigma_s^2 + \sigma_{ps}^2 \end{aligned} \tag{21}$$

And:

$$\mathbb{V}[\hat{f}] = \mathbb{V}[\mu + \nu_p(x)] = \mathbb{V}[\mu] + \mathbb{V}[\nu_p(x)] = 0 + \sigma_p^2(x) = \sigma_p^2(x) =: \sigma_p^2 \tag{22}$$

And substituting these results in Eq. 18 gives us:

$$\begin{aligned} \mathbb{V}[\hat{f}_s] &= \mathbb{V}[\hat{f}] + \mathbb{E}[\mathbb{V}[\hat{f}_s | D_n]] = \mathbb{V}[\hat{f}] + \mathbb{E}[\mathbb{V}[\hat{f} | D_n] + \mathbb{V}[\hat{\varepsilon} | D_n]] \\ &= \mathbb{V}[\hat{f}] + \mathbb{E}[\mathbb{V}[\hat{\varepsilon} | D_n]] = \sigma_p^2(x) + \mathbb{E}[\sigma_s^2(\psi) + \sigma_{ps}^2(x, \psi)] \\ &= \sigma_p^2(x) + \sigma_s^2(s) + \sigma_{ps}^2(x, s) =: \sigma_p^2 + \sigma_s^2 + \sigma_{ps}^2 \end{aligned} \tag{23}$$

This can be extended to a 3-facet design by increasing the number of components: $\hat{f}_{ds} = \mu + \nu_p(x) + \nu_s(s) + \nu_d(d) + \nu_{ps}(x, s) + \nu_{pd}(x, d) + \nu_{ds}(d, s) + \nu_{pds}(x, d, s)$.

Negative variance estimates

From the structure of the variance components one can observe that if the mean square of some of the components are large enough, this will result in a negative estimate for one or more variance estimates. The possibility of negative variance estimates did not go unnoticed by the creators of this framework. Brennan (2001) stated:

> *"Estimated variance components are subject to sampling variability, and the smaller the sample sizes, the more likely it is that estimates will vary. (...) One possible consequence of sampling variability of estimated variance components is that one or more estimated variance components may be negative even though, by definition, variance components are non-negative."*

Due to limited sample sizes the possibility of negative estimates cannot be ruled out. Based on the assumption that these components either have a mean at or are not significantly different from zero it is usually proposed to set these estimates to zero for further computations. The practice is to still report the estimates in-between brackets to show their magnitude. Patterson and Thompson (1971) introduced the restricted maximum likelihood estimator as a proposed solution. This method however, has the disadvantage of being computationally and conceptually much more difficult than the analysis of variance framework. In Section 4.2 a Monte Carlo study has been performed to show that negative estimates are a consequence of sampling variability. That is, when the procedure is replicated $B = 1000$ times the mean estimate will be at zero or positive. Hence the motivation for setting these components to zero is well motivated.

Computational cost

It is evident that the computational cost of this algorithm increases fast. The number of score effects in a m-facet design is $\sum_{i=1}^{m} \binom{m}{i}$. In the case of a 3-facet design this is $\binom{3}{1} + \binom{3}{2} + \binom{3}{3} = 7$, but for a 4-facet design it is 15, for a 5-facet design 31, for a 6-facet design 63, etc. It thus more than doubles with each additional facet. This is only for the computation of all components. If a researcher wishes to generate a particular set of predictions from a model configuration the computational burden increases exponentially. So a $n_p \times n_s = 1000 \times 10$ result matrix requires 10 random forests to be trained making 1000 predictions each, whereas a $n_p \times n_s \times n_d = 1000 \times 10 \times 10$ result matrix already requires 100 random forests.

3-facet G-study

For illustratory purposes, computation of the variance components for a 3-facet design is given in the tables below. The facets are: prediction (p), depth (d) and seed (s). As before, the prediction (p) effect is actually the object of measurement, but for simplicity considered a facet here. The variance over predictions is not interesting for interpretation

as it is dependent upon the application. However, these estimates become relevant in the decision study as discussed in Section 3.3. Tab. 1 shows the degrees of freedom, score effects, mean, sum of squares, mean square and variance estimate for the components in a generalizability study for 3 facets.

Tab. 1: Degrees of freedom, score effects, sample means, sum of squares, mean squares and variance estimates for a 3-facet design.

α	$df(\alpha)$	ν_α	$\bar{y}_\alpha$
p	$n_p - 1$	$\mu_p - \mu$	$\frac{1}{n_d n_s} \sum_d^{n_d} \sum_s^{n_s} y_{pds}$
d	$n_d - 1$	$\mu_d - \mu$	$\frac{1}{n_p n_s} \sum_p^{n_p} \sum_s^{n_s} y_{pds}$
s	$n_s - 1$	$\mu_s - \mu$	$\frac{1}{n_p n_d} \sum_p^{n_p} \sum_d^{n_d} y_{pds}$
pd	$(n_p - 1)(n_d - 1)$	$\mu_{pd} - \mu_p - \mu_d + \mu$	$\frac{1}{n_s} \sum_s^{n_s} y_{pds}$
ps	$(n_p - 1)(n_s - 1)$	$\mu_{ps} - \mu_p - \mu_s + \mu$	$\frac{1}{n_d} \sum_d^{n_d} y_{pds}$
ds	$(n_d - 1)(n_s - 1)$	$\mu_{ds} - \mu_d - \mu_s + \mu$	$\frac{1}{n_p} \sum_p^{n_p} y_{pds}$
pds	$(n_p - 1)(n_d - 1)(n_s - 1)$	$\mu_{pds} - \mu_{pd} - \mu_{ps} - \mu_{ds} + \mu_p + \mu_d + \mu_s - \mu$	

α	$SS(\alpha)$	$MS(\alpha)$	$\sigma^2(\alpha)$
p	$n_d n_s \sum_p^{n_p} (\bar{y}_p - \bar{y})^2$	$\frac{SS(p)}{n_p - 1}$	$\frac{MS(p) - MS(pd) - MS(ps) + MS(pds)}{n_d n_s}$
d	$n_p n_s \sum_d^{n_d} (\bar{y}_d - \bar{y})^2$	$\frac{SS(d)}{n_d - 1}$	$\frac{MS(d) - MS(pd) - MS(ds) + MS(pds)}{n_p n_s}$
s	$n_p n_d \sum_p^{n_s} (\bar{y}_s - \bar{y})^2$	$\frac{SS(s)}{n_s - 1}$	$\frac{MS(s) - MS(ps) - MS(ds) + MS(pds)}{n_p n_d}$
pd	$n_s \sum_p^{n_p} \sum_d^{n_d} (\bar{y}_{pd} - \bar{y}_p - \bar{y}_d + \bar{y})^2$	$\frac{SS(pd)}{(n_p - 1)(n_d - 1)}$	$\frac{MS(pd) - MS(pds)}{n_s}$
ps	$n_d \sum_p^{n_p} \sum_s^{n_s} (\bar{y}_{ps} - \bar{y}_p - \bar{y}_d + \bar{y})^2$	$\frac{SS(ps)}{(n_p - 1)(n_s - 1)}$	$\frac{MS(ps) - MS(pds)}{n_d}$
ds	$n_p \sum_d^{n_d} \sum_s^{n_s} (\bar{y}_{ds} - \bar{y}_d - \bar{y}_s + \bar{y})^2$	$\frac{SS(ds)}{(n_d - 1)(n_s - 1)}$	$\frac{MS(ds) - MS(pds)}{n_p}$
pds	$\sum_p^{n_p} \sum_d^{n_d} \sum_s^{n_s} (\bar{y}_{pds} - \bar{y}_{pd} - \bar{y}_{ps} - \bar{y}_{ds} + \bar{y}_p + \bar{y}_d + \bar{y}_s - \bar{y})^2$	$\frac{SS(pds)}{(n_p - 1)(n_d - 1)(n_s - 1)}$	$MS(pds)$

3.2 Measures of generalizability in a decision study

A decision study uses the information obtained in the variance decomposition stage of a generalizability study to create measures of generalizability.

Consider $\mathcal{R}$ the set of indices concerned with score effects indices, and p the object of measurement. So in the case of a 2-facet design we would have $\mathcal{R} = s$. So we have for example $\mu_p = \mathbb{E}_\mathcal{R}[y_{p\mathcal{R}}] (= \mathbb{E}_s[y_{ps}])$. The mean variance of a facet $\sigma^2(\bar{\alpha})$ is defined as:

$$\sigma^2(\bar{\alpha}) = \frac{\sigma^2(\alpha)}{d(\bar{\alpha})} \tag{24}$$

Here, α is the facet, and

$$d(\bar{\alpha}) = \begin{cases} 1 & \text{if } \bar{\alpha} = p \\ \prod_{i \in \mathcal{R}} n_i & \text{elsewhere} \end{cases} \tag{25}$$

So it is simply the product of the sample sizes included in $\mathcal{R}$. The notation τ is used for the object of measurement (p in this case), Δ for absolute error and δ for relative error. Furthermore, we use Ω to indicate the set of all combinations of indices. That is, in a 2-facet design we have $\Omega = \{p, s, ps\}$.

Absolute error variance

We define the absolute error as:

$$\Delta_{p\mathcal{R}} = y_{p\mathcal{R}} - \mu_p = (\mu + \sum_{i \in \Omega} \nu_i) - \mu_p = \sum_{i \in \Omega} \nu_i - (\mu_p - \mu) = \sum_{i \in \Omega} \nu_i - \nu_p = \sum_{i \in \Omega, i \neq p} \nu_i \quad (26)$$

Which has expectation $\mathbb{E}_{\mathcal{R}}[\Delta_{p\mathcal{R}}] = 0$ and thus variance $\sigma^2(\Delta_{p\mathcal{R}}) = \mathbb{E}_p[(\mathbb{E}_{\mathcal{R}}[\Delta_{p\mathcal{R}}])^2]$. This can be written as the sum:

$$\sigma^2(\Delta_{p\mathcal{R}}) = \sum_{i \in \Omega, i \neq p} \sigma^2(\bar{i}) \quad (27)$$

Where each variance component $\sigma^2(\bar{i})$ in this sum is defined as in Eq. 24. For example, in a 2-facet design we obtain:

$$\sigma^2(\Delta_{ps}) = \sigma^2(\bar{s}) + \sigma^2(\bar{ps}) = \frac{\sigma^2(s)}{n_s} + \frac{\sigma^2(ps)}{n_p n_s} \quad (28)$$

Relative error variance

We define the relative error as:

$$\begin{aligned} \delta_{p\mathcal{R}} &= (y_{p\mathcal{R}} - \mathbb{E}_p[y_{p\mathcal{R}}]) - (\mu_p - \mu) = (\mu + \sum_{i \in \Omega} \nu_i - \mu_{\mathcal{R}}) - \nu_p \\ &= (\mu + \sum_{i \in \Omega} \nu_i - \mu + \sum_{j \in \Omega, p \notin j} \nu_j) - \nu_p = \sum_{i \in \Omega, p \in i, i \neq p} \nu_i \end{aligned} \quad (29)$$

It might seem complicated at first hand, but it is simply the sum of score effects that contain p, but not ν_p. Since all score effects have zero expectation, the relative error variances can then be written as $\sigma^2(\delta_{p\mathcal{R}}) = \mathbb{E}_p[(\mathbb{E}_{\mathcal{R}}[\delta_{p\mathcal{R}}])^2]$. This gives:

$$\sigma^2(\delta_{p\mathcal{R}}) = \sum_{i \in \Omega, p \subseteq i, i \neq p} \sigma^2(\bar{i}) \quad (30)$$

For example, in a 2-facet design we obtain:

$$\sigma^2(\delta_{ps}) = \sigma^2(\bar{ps}) = \frac{\sigma^2(ps)}{n_p n_s} \quad (31)$$

Coefficients

There are two coefficients that give an interpretation of the amount of variance being due to the different data characteristics (the desired variance) with respect to the error variances (the disliked variance). First is the generalizability coefficient:

$$\rho^2(p) = \frac{\sigma^2(p)}{\sigma^2(p) + \sigma^2(\delta_p)} = \frac{\sigma^2(p)}{\mathbb{E}_{\mathcal{R}}[\mathbb{E}_p[(y_{p\mathcal{R}} - \mathbb{E}_p[y_{p\mathcal{R}}])^2]]} \tag{32}$$

The generalizability coefficient can be seen as the variance solely due to prediction effects relative to the variance components related to p. Second is the index of dependability:

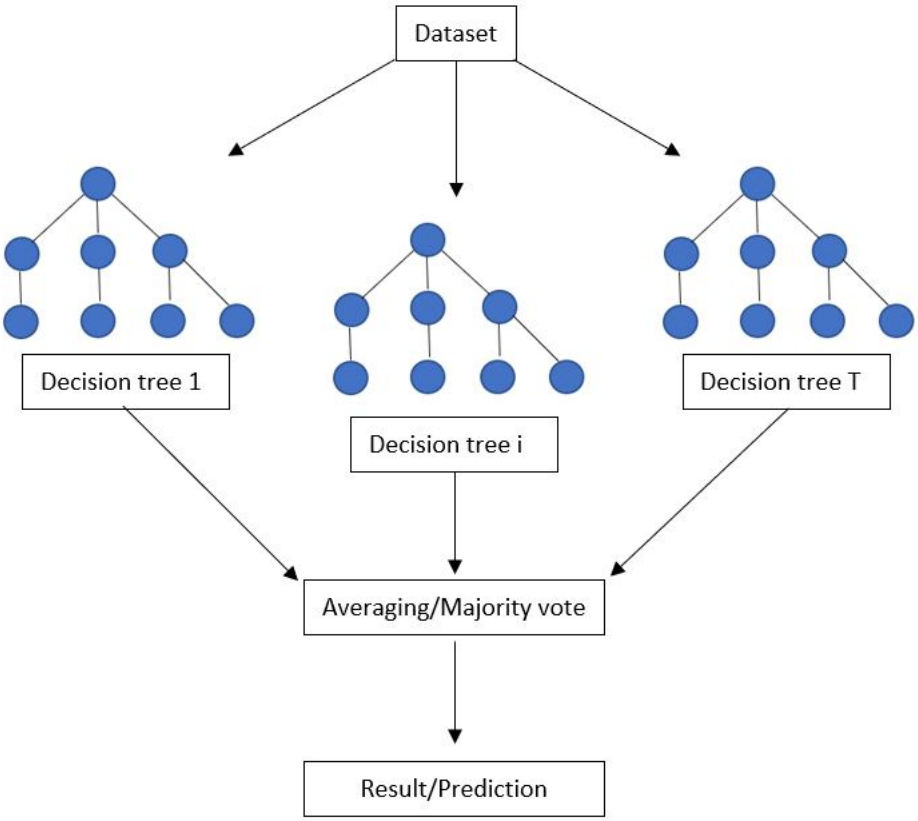
$$\Phi(p) = \frac{\sigma^2(p)}{\sigma^2(p) + \sigma^2(\Delta_p)} = \frac{\sigma^2(p)}{\mathbb{E}_p[(y_{p\mathcal{R}} - \mathbb{E}_{p\mathcal{R}}[y_{p\mathcal{R}}])^2]} \tag{33}$$

The index of dependability is the variance of prediction effects relative to all variance components. Hence we always have: $\Phi(p) \leq \rho^2(p)$.

3.3 Random forests

The origins of random forests can be traced back to Amit and Geman (1997), Ho (1998), and Dietterich (2000). There, the first forms of ensemble learning were applied. Based on their work it was Leo Breiman (2001) who formalized the procedure and showed the statistical properties of the algorithm. He proved the consistency of the algorithm, and showed that as the number of trees in a forest grows large the generalization error converges asymptotically to a limit. These estimators have shown to be reliable predictors of large datasets. A schematic illustration of the workings of a random forest with a total of T decision trees is given in Fig. 2.

Fig. 2: Schematic view of the random forest algorithm



Regression

We define $X_i \in \mathbb{R}^p$ the input data for a single observation, and $Y_i \in \mathbb{R}$ the corresponding true value. The set $D_n = \{(X_1, Y_1), ..., (X_n, Y_n)\}$ constitutes the training data to construct the decision tree $m_t(x|D_n, \Theta_t, \psi)$ from which the random forest estimator $M_T := M_T(x|D_n, \Theta, \psi)$ is constructed, with x the data for prediction outside the sample D_n and T the number of regression trees. We have a vector of parameters ψ and an application of parameters to the algorithm $\Theta = \{\Theta_1, ..., \Theta_T\}$. The latter is the application of the parameters to the forest. For example, if in each tree $m = 5$ variables are selected for splitting the data, then the selected variables for tree t are contained in Θ_t. It is thus the application of ψ on a given decision tree in the algorithm. The cell $A_k := A_k(x|D_n, \Theta_t)$ is the final node (*leaf*) of a decision tree t containing x from all possible nodes $\mathcal{A}$ generated by the tree. $N_k = N_k(x|D_n, \Theta_t)$ is the number of datapoints

X_i in node A_k. The prediction of a single decision tree of the datapoint x if $x \in A_k$:

$$m_t(x|D_n, \Theta_t, \psi) = \sum_{X_i \in D_n} \frac{\mathbb{I}_{[X_i \in A_k]} Y_i}{N_k} \tag{34}$$

The prediction is the average value of all datapoints in the training set D_n contained in A_k. The random forest estimator aggregates over all these decision trees. A prediction $\hat{f}(x)$ by this algorithm can thus be written as:

$$\hat{f}(x) = M_T(x|D_n, \Theta, \psi) = \frac{1}{T} \sum_{t=1}^{T} \omega_t m_t(x|D_n, \Theta_t, \psi) \tag{35}$$

with ω_t a weight function such that $\sum_t^T \omega_t = 1$. It is most common that all trees receive a unit vote. Then, the forest estimate is simply the average over all decision tree estimates.

Classification

In the case of classification trees the prediction depends on the probability assigned by the forest classifier. The dataset is split in subsets which are as homogeneous as possible. If $x \in A_k$, and the label $y = j$ is assigned to the observations $X_i \in A_k$ we have:

$$m_t(x|D_n, \Theta_t, \psi) = j \tag{36}$$

The probability $\mathbb{P}[y = j]$ for the outcome $y = j$ in the random forest classifier is:

$$\mathbb{P}[y = j] = M_T(x|D_n, \Theta_t, \psi) = \frac{1}{T} \sum_{t=1}^{T} \mathbb{I}_{[m_t(x|\Theta_t, D_n) = j)]} \tag{37}$$

with ω_t a weight function such that $\sum_t^T \omega_t = 1$. It is most common that all trees receive a unit vote, and then the forest estimate is simply the average over all decision classifier estimates. The class label is then assigned as $\hat{y} = j$ if $\mathbb{P}[y = j]$ is the highest probability among all classes.

3.4 Interpretation of the decision study coefficients for Machine Learning

Although both the index of dependability and generalizability coefficient are also contained on the interval $[0, 1]$, they cannot be used with the same interpretation as a standard reliability coefficient. With the standard reliability measure as in classical test theory we refer to Eq. 2.

For the coefficients in generalizability theory we use only the variance of the mean over a facet ($\sigma^2(\bar{\alpha})$) as defined in Section 3.2. The question arises how to interpret the value of a coefficient in a decision study. It cannot be handled in the same way as ρ since

it uses different components. The variance component $\sigma^2(\bar{\alpha})$ is the average variance over facets. Since it is not necessarily prediction error variance in the sense of ρ that we are dealing with, the interpretation of the coefficient should be determined with the application at hand. For example, in the case of choosing a different seed we expect there to be little to no influence. However, depth is a more fundamental parameter in the model and the variance due to different choices for depth can thus also be greater.

So what is good generalizability? A study across applications of generalizability theory show that regularly 0.80 is considered 'acceptable' and 0.90 is deemed 'good' (Mushquash & O'Connor, 2006). In machine learning there are two dimensions on which we can base what a good value is. The first being the relative importance of the facet to the model, the second the relative variability of the facet. In our example *seed* has a low importance (it should be around 0 in the ideal scenario), and a low variability as there is an infinite amount of theoretically possible seeds. In the application of Section 5 we will see that *depth* has a high importance, and the amount of variability can be determined by the researcher (and also depends on the model and data).

Consider the following example: We observe $\sigma^2(p) = 5$, $\sigma^2(s) = 2.5$ and $\sigma^2(ps) = 1$ over 10 seeds and 100 observations. We then get $\sigma^2(\bar{s}) = 0.25$ and $\sigma^2(\bar{ps}) = 0.01$ and would then get $\rho^2(p) = \frac{5}{5+0.01} = 1.00$ and $\Phi(p) = \frac{5}{5.26} = 0.95$ as in Eq. 32 and 33. This shows that about 5% of variance is due to *seed*-related effects on mean predictions. In an application with a large amount of data this is undesirable and one could question the appropriateness of random forest in this regard. However, when the same results would be observed for the *depth*-facet the interpretation would be different, as it is expected to have a bigger influence. Still, coefficients below 0.80 are undesirable as this indicates about 20% of variance in mean scores can be directly related to score effects. By comparing results of applications in the literature on generalizability theory we see that researchers are generally satisfied with a generalizability of 0.80 and aim at a value of 0.90 (Arterberry et al. (2014), Graham et al. (2016)).

Instead of only looking at applications of generalizability in psychometrics, a second approach would be to do multiple studies in machine learning and capture relative generalizability in these coefficients. So instead of only looking at the value, compare it with different designs or values. This gives the researcher the option to compare generalizability to a certain facet of the universe with respect to other facets. This indicates the relative importance of a facet. To increase the interpretability of these coefficients, more research should be done on this topic.

4 Monte Carlo study

In order to assess the performance of the framework described in Section 3 a Monte Carlo study is performed in two stages: (i) assessing the adequacy of the variance decomposition framework by estimating fixed variance components for 2 and 3 facets (i.e. 3 and 7 components respectively, as defined in Section 3). I report the mean, mean absolute error, standard deviation and root mean squared error of the estimates; (ii) the framework is applied to a random forest, for three different datapoints. The variance estimates are compared with the variance of the prediction error. I report the mean variance estimates, ratio with respect to prediction error variance, standard deviation, minimum and maximum estimate. In the appendix plots of the (empirical) distribution for stage (ii) are depicted. Study (i) and (ii) are performed for 2 and 3 facets. In the 2 facet design we have a *prediction* (p) and *seed* (s) effect. In the 3 facet design we have a *prediction* (p), *seed* (s) and *depth* (d) effect.

4.1 Adequacy of the framework in estimating variance components

The aim of study (i) is to show that the framework correctly estimates variance components from a matrix of observed scores.

4.1.1 Monte Carlo design

The objective is to estimate components from a matrix of results Y. We have the sets of subscripts $\Omega_2 = \{p, s, ps\}$, $\Omega_3 = \{p, s, d, ps, pd, ds, pds\}$. Variance components σ_j^2 are fixed for $j \in \Omega_i$ for $i \in \{2, 3\}$. For each component a vector of independently distributed variables $z^j \sim \mathcal{N}(0, 1)$ is generated. The size of this vector is equal to the length of this facet. So z^p has shape $(n_p, 1)$, z^s has shape $(1, n_s)$ and z^{ps} has shape (n_p, n_s). This enables us to fix variance along each dimension of the observed scores which will be generated. The fixed grand mean $\mu = 0$ is irrelevant to the components in a correctly designed framework. We now differentiate between continuous outcomes and probabilities, in order to work with both kinds of outputs. For 2 facets the (n_p, n_s) matrix of scores Y is generated as:

$$Y = \mu + \sigma_p z^p + \sigma_s z^s + \sigma_{ps} z^{ps} \tag{38}$$

where $\sigma_p = \sqrt{\sigma_p^2}$ the standard deviation. The (n_p, n_s, n_d) matrix Y for 3 facets is generated as:

$$Y = \mu + \sigma_p z^p + \sigma_s z^s + \sigma_d z^d + \sigma_{ps} z^{ps} + \sigma_{pd} z^{pd} + \sigma_{ds} z^{ds} + \sigma_{pds} z^{pds} \tag{39}$$

An individual score (in the 2 facet design) can be written as: $y_{ps} = \mu + \sigma_p z_p^p + \sigma_s z_s^s + \sigma_{ps} z_{ps}^{ps}$ where the superscript ^{j} refers to the vector and the subscript $_j$ to the index of the variable for $j \in \Omega_2$. For a probability score (in case of classification) a score is mapped on the interval $[0, 1]$ with a logistic transformation:

$$y_{ps} = \frac{1}{1 + \exp(-y_{ps})} \tag{40}$$

We increase the number of predictions n_p, seeds n_s and depths n_d to show that the variance component estimates are estimated consistently with an increasing sample size, $n_i \in \{250, 500, 1000\}$. For each number of observations we regenerate the data and estimate components $B = 1000$ times and report the metrics as defined under Section 4 for 2 and 3 facets. For each iteration the estimated component $\sigma_b^2(j)$ is saved. Note the difference in notation: the true variance is indicated as σ_j^2, whereas the estimate is defined as $\sigma^2(j)$. After $B = 1000$ simulations the mean, mean absolute error (MAE), standard deviation and root mean squared error (RMSE) are computed according to:

$$\begin{aligned}
\bar{\sigma}^2(j) &= \frac{1}{B} \sum_b \sigma_b^2(j) \\
MAE(\sigma^2(j)) &= \frac{1}{B} \sum_b |\sigma_b^2(j) - \sigma_j^2| \\
\sigma(\sigma^2(j)) &= \sqrt{\frac{1}{B-1} \sum_b (\sigma_b^2(j) - \bar{\sigma}^2(j))^2} \\
RMSE(\sigma^2(j)) &= \sqrt{\frac{1}{B} \sum_b (\sigma_b^2(j) - \sigma_j^2)^2}
\end{aligned} \tag{41}$$

The algorithm is summarized in pseudocode in Alg. 1.

Algorithm 1: Adequacy of the framework in estimating variance components

```
Result: Consistency of estimates
for N ∈ {250, 500, 1000} do
    for b ∈ {1, ..., B} do
        generate Y as in Eq. 38 (2 facets) or 39 (3 facets);
        if dichotomous then
            apply transformation in Eq. 40;
        end
        compute σ²_b(j) ∀ j and save the estimates;
    end
    compute the mean, mean absolute error, standard deviation and root mean
     squared error of σ²(j) ∀ j according to Eq. 41;
end
```

In finite samples the empirical variance is defined as $\mathbb{V}[y] = \frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y})^2$ with $\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$. We can see in case of a probability score that we have $\max(\mathbb{V}[y]) = 1$. Therefore, the variance parameters have been normalized in case of Monte Carlo estimation. This has no direct effect on the performance of the algorithm since only the ratio of variance components is relevant in the computation of the generalizability coefficient and index of dependability.

4.1.2 Monte Carlo results

Tab. 2: Results for simulation of variance components for 2 facets, for normal and dichotomous data, $B = 1000$ replications.

Sample size		**Normal**			**Dichotomous**		
		$\sigma^2(p)$	$\sigma^2(s)$	$\sigma^2(ps)$	$\sigma^2(p)$	$\sigma^2(s)$	$\sigma^2(ps)$
	parameters	4.0000	4.0000	2.0000	0.4000	0.4000	0.2000
250		3.9977	4.0018	2.0003	0.3944	0.3953	0.2104
500	MEAN	3.9993	3.9959	1.9998	0.3954	0.3938	0.2108
1000		4.0006	4.0148	2.0000	0.3943	0.3949	0.2108
250		0.2922	0.2950	0.0093	0.0211	0.0215	0.0118
500	MAE	0.2090	0.2044	0.0046	0.0149	0.0153	0.0113
1000		0.1468	0.1459	0.0022	0.0114	0.0114	0.0109
250		0.3643	0.3701	0.0116	0.0258	0.0261	0.0093
500	STD	0.2595	0.2544	0.0058	0.0180	0.0179	0.0070
1000		0.1860	0.1800	0.0028	0.0131	0.0132	0.0047
250		0.3643	0.3701	0.0116	0.0264	0.0266	0.0142
500	RMSE	0.2595	0.2544	0.0058	0.0186	0.0189	0.0131
1000		0.1860	0.1806	0.0028	0.0143	0.0141	0.0118

Tab. 3: Results for simulation of variance components for 3 facets, for normal data, $B = 1000$ replications.

		Normal						
Sample size		$\sigma^2(p)$	$\sigma^2(s)$	$\sigma^2(d)$	$\sigma^2(ps)$	$\sigma^2(pd)$	$\sigma^2(ds)$	$\sigma^2(pds)$
	parameters	4.0000	8.0000	16.0000	5.0000	5.0000	5.0000	2.0000
250		4.0114	8.0034	16.0156	4.9999	4.9994	5.0001	2.0000
500	MEAN	4.0056	8.0048	15.9914	4.9998	5.0004	5.0000	2.0000
1000		3.9912	8.0042	15.9766	5.0003	5.0001	5.0000	2.0000
250		0.2853	0.5881	1.1604	0.0230	0.0222	0.0224	0.0006
500	MAE	0.1984	0.4025	0.8194	0.0115	0.0107	0.0111	0.0002
1000		0.1447	0.2887	0.5830	0.0057	0.0056	0.0057	0.0001
250		0.3608	0.7455	1.4314	0.0289	0.0278	0.0278	0.0007
500	STD	0.2481	0.5060	1.0257	0.0143	0.0135	0.0139	0.0002
1000		0.1803	0.3651	0.7177	0.0072	0.0070	0.0072	0.0001
250		0.3610	0.7455	1.4315	0.0289	0.0278	0.0278	0.0007
500	RMSE	0.2481	0.5060	1.0257	0.0143	0.0135	0.0139	0.0002
1000		0.1805	0.3651	0.7180	0.0072	0.0070	0.0072	0.0001

Tab. 4: Results for simulation of variance components for 3 facets, for dichotomous data, $B = 100$ replications.

		Dichotomous						
Sample size		$\sigma^2(p)$	$\sigma^2(s)$	$\sigma^2(d)$	$\sigma^2(ps)$	$\sigma^2(pd)$	$\sigma^2(ds)$	$\sigma^2(pds)$
	parameters	0.0889	0.1778	0.3556	0.1111	0.1111	0.1111	0.0444
250		0.0877	0.1749	0.3501	0.1102	0.1114	0.1128	0.0529
500	MEAN	0.0875	0.1747	0.3506	0.1102	0.1114	0.1128	0.0529
1000		0.0876	0.1753	0.3501	0.1101	0.1113	0.1127	0.0528
250		0.0064	0.0113	0.0185	0.0034	0.0032	0.0034	0.0084
500	MAE	0.0046	0.0083	0.0125	0.0025	0.0023	0.0026	0.0084
1000		0.0033	0.0060	0.0095	0.0018	0.0016	0.0020	0.0084
250		0.0079	0.0141	0.0223	0.0041	0.0040	0.0039	0.0014
500	STD	0.0055	0.0098	0.0148	0.0030	0.0029	0.0028	0.0010
1000		0.0039	0.0071	0.0104	0.0020	0.0019	0.0019	0.0007
250		0.0080	0.0144	0.0230	0.0042	0.0040	0.0042	0.0086
500	RMSE	0.0057	0.0103	0.0157	0.0031	0.0029	0.0032	0.0085
1000		0.0041	0.0075	0.0117	0.0022	0.0019	0.0025	0.0084

The results can be observed in Tab. 2, 3 and 4. The following is observed: i) the mean estimates are close to the true values, for both the dichotomous data and normal data. For both types of data the difference is smaller than of the order 10^{-2}; ii) the mean absolute error (MAE) decreases proportional to the square root of the increase in sample size; iii) the standard deviation (STD) of estimates decreases proportional to the square root of the increase in sample size; iv) the root mean squared error (RMSE) of estimates decreases proportional to the square root of the increase in sample size; (v) the MAE and RMSE for the term $\sigma^2(pds)$ for dichotomous data do not decrease materially for an increasing sample size, but the standard deviation decreases in the same manner as the other components. The point (i) indicates that the framework produces unbiased estimates of the population variance. The points (ii)-(iv) indicate that for an increase from 250 to 1000 observations for all facets - a 4 times increase - the mean absolute error, standard deviation and root mean squared error decrease by 50% (as $\sqrt{4} = 2$). This is evidence that the framework might estimate variance components consistently. The point (v) indicates that the bias of the residual term does not decrease for an increasing sample size for dichotomous data.

4.2 Adequacy of the framework applied to random forest

The aim of stage (ii) is first to show that with appropriate statistical motivation the negative variance components can be set to 0 and to compare the estimated variance components with the prediction error variance. The Monte Carlo study shows how the

framework can be applied to random forests.

4.2.1 Monte Carlo design

The number of observations $n \in \{250, 500, 1000\}$ and number $B = 1000$ of Monte Carlo simulations are fixed. The process is shown for 2 facets, but is also extended to 3 facets. See Section 3.1 for the decomposition into score effects in this setting. The data generating process is:

$$f(x_i) = 2x_{i1} - x_{i2}^2 + (x_{i3} - 1)^2 - 3x_{i4} + 2 \tag{42}$$

With $x_i = (x_{i1}, x_{i2}, x_{i3}, x_{i4})$. There is some noise in the data. The training data is generated as:

$$y_i = f(x_i) + \eta_i \tag{43}$$

where $x_{i1}, x_{i2}, x_{i3}, x_{i4} \overset{iid}{\sim} 5 * \mathcal{N}(0, 1)$ and the noise $\eta_i \sim \mathcal{N}(0, 1)$. For each simulation the training set $D_n = \{(x_1, y_1), ..., (x_n, y_n)\}$ is generated according to Eq. 42 and 43. The random forest algorithm produces the estimator $\hat{f}_s(x)$ on training data D_n with random state (seed) s. Consider now the data point $x \notin D_n$. Analogous to Eq. 3 and 4 we can write for a particular *seed*:

$$\hat{f}_s(x) = \mu + \nu_p(x) + \nu_s(s) + \nu_{ps}(x, s) = f(x) + \nu_s(s) + \nu_{ps}(x, s) \tag{44}$$

Since there is noise in the data we have for any estimator $\hat{f}(x)$ some prediction error ε such that $\hat{y} = \hat{f}(x) + \varepsilon$. Then, for a particular seed s we can write:

$$\hat{y}_s = \hat{f}_s(x) + \varepsilon = f(x) + \nu_s(s) + \nu_{ps}(x, s) + \varepsilon \tag{45}$$

So we have the prediction error ε since the data does not exactly follow $f(x)$ and the error due to the score effects, in this case *seeds*. We know $f(x)$ as defined in Eq. 42. We also know that the average score effect of ν_s and ν_{ps} over all *seeds* is zero (see Eq. 5). In other words: $\mathbb{E}_s[\nu_s(s)] = \mathbb{E}_s[\nu_{ps}(x, s)] = 0$. In order to gain an estimate of ε we need to eliminate the score effects. To obtain this, we take $\hat{\bar{y}} = \frac{1}{n_s} \sum_s \hat{y}_s$. Rewriting Eq. 45:

$$\hat{\bar{y}} - f(x) = \frac{1}{n_s} \sum_s (\hat{y}_s - f(x)) = \frac{1}{n_s} \sum_s (\nu_s(s) + \nu_{ps}(x, s) + \varepsilon) \tag{46}$$

The score effects have an expectation of 0 over all seeds, as defined in Eq. 5. Hence, $\mathbb{E}[\hat{\bar{y}} - f(x)] = \varepsilon$. For each simulation we obtain the empirical error $\varepsilon_b = \hat{\bar{y}}_b - f(x)$ and

use this to calculate the mean error and error variance:

$$\begin{aligned} \bar{\varepsilon} &= \frac{1}{B}\sum_{b}\varepsilon_b \\ \sigma^2(\varepsilon) &= \frac{1}{B-1}\sum_{b}(\varepsilon_b - \bar{\varepsilon})^2 \end{aligned} \tag{47}$$

For each score effect ν_j the variance is estimated according to the process as defined in Section 3. For each iteration a component $\sigma_b^2(j)$ is saved. After $B = 1000$ simulations the mean ($\bar{\sigma}^2(j)$) and standard deviation ($\sigma(\sigma^2(j))$) of estimates are computed according to:

$$\begin{aligned} \bar{\sigma}^2(j) &= \frac{1}{B}\sum_{b}\sigma_b^2(j) \\ \sigma(\sigma^2(j)) &= \sqrt{\frac{1}{B-1}\sum_{b}(\sigma_b^2(j) - \bar{\sigma}^2(j))^2} \end{aligned} \tag{48}$$

The ratio between all variance components is computed as:

$$\begin{aligned} RATIO(\sigma^2(j)) &= \frac{\sigma^2(j)}{\sigma^2(\varepsilon) + \sum_{j\in\Omega_i}\sigma^2(j)} \\ RATIO(\sigma^2(\varepsilon)) &= \frac{\sigma^2(\varepsilon)}{\sigma^2(\varepsilon) + \sum_{j\in\Omega_i}\sigma^2(j)} \end{aligned} \tag{49}$$

Where $i \in \{2, 3\}$, depending on the number of facets and Ω_i is defined as in Section 4.1.1. The minimum ($\min_b\{\sigma_b^2(j)\}$) and maximum ($\max_b\{\sigma_b^2(j)\}$) of estimates are also reported.

I investigate the robustness of the algorithm around the points δx for $x = (x_1, x_2, x_3, x_4) = (1, 1, 1, 1)$ and $\delta \in \{-1, 0, 1\}$. The algorithm is summarized in pseudocode in Alg. 2. Note that for 2 facets the inner loop for *depth* is skipped.

Algorithm 2: Adequacy of the framework applied to random forest

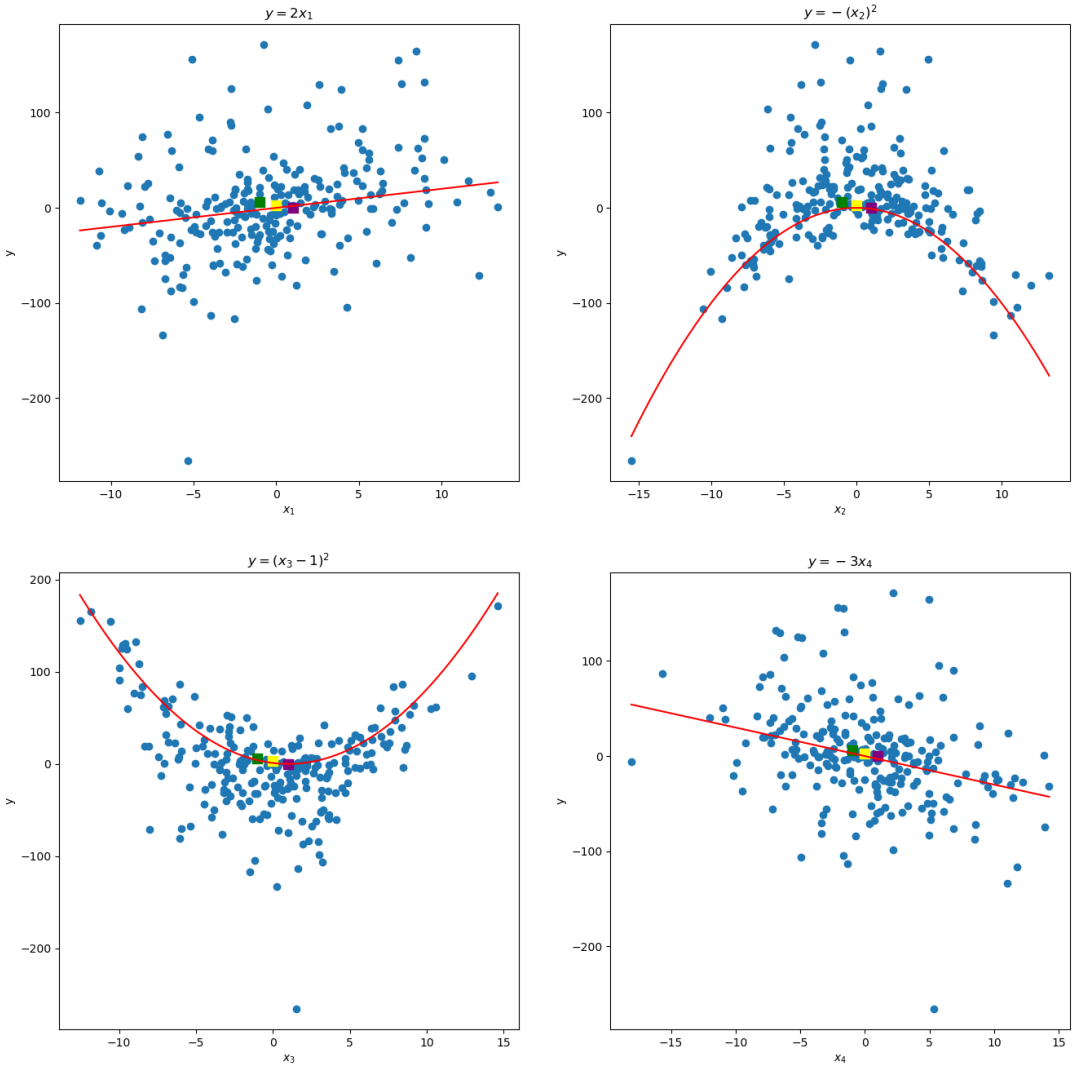
Result: Statistics of variance components and prediction error
for $n \in \{250, 500, 1000\}$ **do**
  for $b \in \{1, ..., B\}$ **do**
    generate $\{x_1, ..., x_n\}$ for $\mathrm{x}_i = (x_{i1}, x_{i2}, x_{i3}, x_{i4}))with\mathrm{x}_i j \overset{iid}{\sim} 5 * \mathcal{N}(0, 1)$;
    generate $\{y_1, ..., y_n\}$ with $y_i = f(x_i) + \eta_i$ with $\eta_i \sim \mathcal{N}(0, 1)$;
    obtain training data $D_n = \{(x_1, y_1), ..., (x_n, y_n)\}$;
    for $s \in \{100, 200, ..., 1000\}$ **do**
      for $d \in \{2, 6, ..., 30\}$ **do**
        train random forest $\hat{f}_{ds}(\cdot)$ on D_n;
        predict $\hat{y}_{ds} = \hat{f}_{ds}(\delta x)$ and save the result;
      end
    end
    compute $\sigma_b^2(j)\ \forall\ j$ and save the estimates;
    calculate $\hat{\bar{y}} = \frac{1}{n_s n_d} \sum_s \sum_d \hat{y}_{ds}$;
    save $\varepsilon_b = \hat{\bar{y}} - f(\delta x)$;
  end
  compute the mean, minimum, maximum and standard deviation of $\sigma^2(j)\ \forall\ j$;
  compute the mean $\bar{\varepsilon}$ and variance $\sigma^2(\varepsilon)$ according to Eq. 47;
end

The algorithm is repeated three times (for all δ) for both 2 (using only *seed*) and 3 (with both *seed* and *depth*) facets. Hence, there are six experiments of which the results are reported in Section 4.2.2.

4.2.2 Monte Carlo results

It should be noted that since we only investigate one prediction at a time, the variance components related to the *prediction* effect are (close to) 0. Therefore, only the components related to either *seed* or *depth* will be reported in the tables below. That is, I report $\sigma^2(s)$ and $\sigma^2(\varepsilon)$ for the 2 facet design, and $\sigma^2(d)$, $\sigma^2(s)$, $\sigma^2(ds)$ and $\sigma^2(\varepsilon)$ for the 3 facet design. The magnitude of the variance in prediction error can be compared with the variance components attributable to different facets. An illustration of the data generating process for a single iteration in the Monte Carlo study is given in Fig. 3.

Fig. 3: Data distribution with respect to all four covariates for $n = 250$ observations



The figure shows the data distribution of a single iteration in the Monte Carlo study. The data is generated according to Eq. 43. The red line gives the data generating process for each covariate. The three squares indicate the points at which the variance components are evaluated; $\delta = -1$ (green), $\delta = 0$ (yellow) and $\delta = 1$ (purple).

For a number of observations $n \in \{250, 500, 1000\}$ $B = 1000$ sets of variance components are estimated. The distribution of these components is examined and compared to the error variance $\sigma^2(\varepsilon)$ from Eq. 47. Resulting statistics are presented in Section 4.2.2, distribution plots are given in Appendix 8.3.1 and 8.3.2. The latter contains plots on all variance estimates, also the components which are (close) to 0.

Tab. 5: Estimates of variance components at $\delta x = \delta(1,1,1,1)$ in random forest, for 2 facets, for $B = 1000$ replications.

		$\delta = -1$	$\delta = 0$	$\delta = 1$
Sample size		$\sigma^2(s)$	$\sigma^2(s)$	$\sigma^2(s)$
250		0.8577	0.7985	0.8522
500	MEAN	0.5686	0.5121	0.5335
1000		0.3620	0.3220	0.3270
250		0.0987	0.0955	0.0959
500	RATIO	0.1134	0.1092	0.1049
1000		0.1265	0.1255	0.1151
250		0.5458	0.5391	0.5780
500	STD	0.3735	0.3265	0.3511
1000		0.2335	0.2046	0.1975
250		0.0601	0.0743	0.0674
500	MIN	0.0534	0.0394	0.0446
1000		0.0353	0.0196	0.0271
250		4.8688	5.3044	4.5664
500	MAX	4.1343	2.4750	3.0289
1000		1.7800	1.4565	1.4013

Tab. 6: Estimates of prediction error variance in random forest, for 2 facets, for $B = 1000$ replications.

		$\delta = -1$	$\delta = 0$	$\delta = 1$
Sample size		$\sigma^2(\varepsilon)$	$\sigma^2(\varepsilon)$	$\sigma^2(\varepsilon)$
250		7.8318	7.5665	8.0355
500	VAR	4.4459	4.1788	4.5503
1000		2.5004	2.2429	2.5134
250		0.9013	0.9045	0.9041
500	RATIO	0.8866	0.8908	0.8951
1000		0.8735	0.8745	0.8849

Tab. 7: Estimates of variance components in random forest, for 3 facets at $\delta x = \delta(1,1,1,1)$, for $\delta = -1$, $B = 1000$ replications.

		$\delta = -1$		
Sample size		$\sigma^2(d)$	$\sigma^2(s)$	$\sigma^2(ds)$
250	MEAN	1.5236	0.6709	0.1988
500		1.5864	0.3966	0.1856
1000		1.7443	0.2324	0.1660
250	RATIO	0.1417	0.0624	0.0185
500		0.2383	0.0596	0.0279
1000		0.3513	0.0468	0.0334
250	STD	1.9010	0.4382	0.0882
500		1.8174	0.2625	0.0757
1000		1.7390	0.1547	0.0632
250	MIN	-0.0365	0.0198	0.0392
500		-0.0208	0.0066	0.0516
1000		-0.0226	0.0076	0.0429
250	MAX	15.2392	2.8695	0.8319
500		11.9468	2.1131	0.7054
1000		12.1011	1.0180	0.5378

Tab. 8: Estimates of variance components in random forest, for 3 facets at $\delta x = \delta(1.1.1.1)$. for $\delta = 0$. $B = 1000$ replications.

		$\delta = -1$		
Sample size		$\sigma^2(d)$	$\sigma^2(s)$	$\sigma^2(ds)$
250	MEAN	1.4215	0.5920	0.1895
500		1.5248	0.3638	0.1762
1000		1.8077	0.2216	0.1654
250	RATIO	0.6453	0.2687	0.0860
500		0.7385	0.1762	0.0853
1000		0.8237	0.1010	0.0753
250	STD	1.7380	0.4206	0.0810
500		1.7813	0.2621	0.0679
1000		1.9177	0.1502	0.0660
250	MIN	-0.0237	0.0392	0.0322
500		-0.0086	0.0136	0.0464
1000		-0.0070	0.0127	0.0439
250	MAX	12.5324	3.3994	0.6476
500		18.6176	2.4143	0.5358
1000		12.6446	1.0886	0.4539

Tab. 9: Estimates of variance components in random forest, for 3 facets at $\delta x = \delta(1,1,1,1)$, for $\delta = 1$, $B = 1000$ replications.

		$\delta = -1$		
Sample size		$\sigma^2(d)$	$\sigma^2(s)$	$\sigma^2(ds)$
250	MEAN	1.4305	0.6291	0.1961
500		1.5441	0.3801	0.1799
1000		1.8864	0.2228	0.1647
250	RATIO	0.1579	0.0694	0.0216
500		0.2529	0.0623	0.0295
1000		0.3746	0.0442	0.0327
250	STD	1.7718	0.4322	0.0835
500		1.6763	0.2470	0.0691
1000		2.0439	0.1557	0.0619
250	MIN	-0.0245	0.0393	0.0502
500		-0.0152	0.0134	0.0407
1000		-0.0122	0.0055	0.0403
250	MAX	15.6588	2.9772	0.6323
500		12.6844	2.1268	0.5008
1000		15.9806	1.3032	0.4510

Tab. 10: Estimates of prediction error variance in random forest, for 3 facets at $\delta x = \delta(1,1,1,1)$, for $B = 1000$ replications.

		$\delta = -1$	$\delta = 0$	$\delta = 1$
Sample size		$\sigma^2(\varepsilon)$	$\sigma^2(\varepsilon)$	$\sigma^2(\varepsilon)$
250	VAR	8.3599	7.2424	6.8041
500		4.4879	4.5771	4.0022
1000		2.8221	2.7158	2.7624
250	RATIO	0.7774	0.7668	0.7510
500		0.6742	0.6891	0.6554
1000		0.5684	0.5531	0.5485

The results of the experiment are given in Tab. 5 7, 8 and 9. Tab. 6 and 10 contain the resulting prediction error variance. I report the mean estimates (MEAN), ratio between with respect to all variance components (RATIO), standard deviation of the estimates (STD), minimum estimate (MIN) and maximum estimate (MAX). The tables on $\sigma^2(\varepsilon)$ contain only the mean value of the variance component (VAR), as there is only one such component for each iteration of the Monte Carlo simulation. The resulting plots of the variance component distributions (including *prediction* effects) are given in Appendix 8.3.1 and 8.3.2.

The following is observed from these results: (i) non-negativity of the variance estimates. Because of sampling variability, variance estimates can by construction become negative. As discussed in Section 3.1 these should be set to 0. As we see from the distribution plots, there are indeed estimates that can turn negative. Their magnitude is of the order 10^{-30} and lower and hence they are not materially different from 0. For components which have some estimates below 0 - *depth*, see Tab. 7 and 8 - we see that the mean is strictly positive and only a small fraction of estimates is negative. This justifies setting (small) negative variance estimates to 0; (ii) the standard deviation of estimates decreases in most cases. This is evidence that the accuracy of the estimates increases with the sample size; (iii) the ratio between the variance components and prediction error variance is larger for the 'more fundamental' facet. That is, the *depth* facet is more important to predictions made by the algorithm, and hence by changing this parameter we indeed observe a higher influence of this component with respect to the prediction error variance; (iv) the magnitude of the prediction error variance $\sigma^2(\varepsilon)$ decreases relative to the other components for an increasing sample size. This can be observed by looking at the ratio in Tab. 6 and 10, which decreases for an increasing sample size; (v) the prediction error variance decreases approximately proportional to the increase in sample size. If the sample size is doubled, the prediction error variance is approximately halved. This is evidence that the random forest is an consistent estimator of $f(x)$.

4.3 Summary of Monte Carlo experiments

The results in Section 4.1 provide evidence that the framework produces unbiased and consistent estimates of the variance components. Section 4.2 shows that the framework can be applied to random forests, and negative components can be fixed to 0. It also indicates that the facet with a larger impact on the algorithm has also substantially greater impact on the total variance.

5 Application

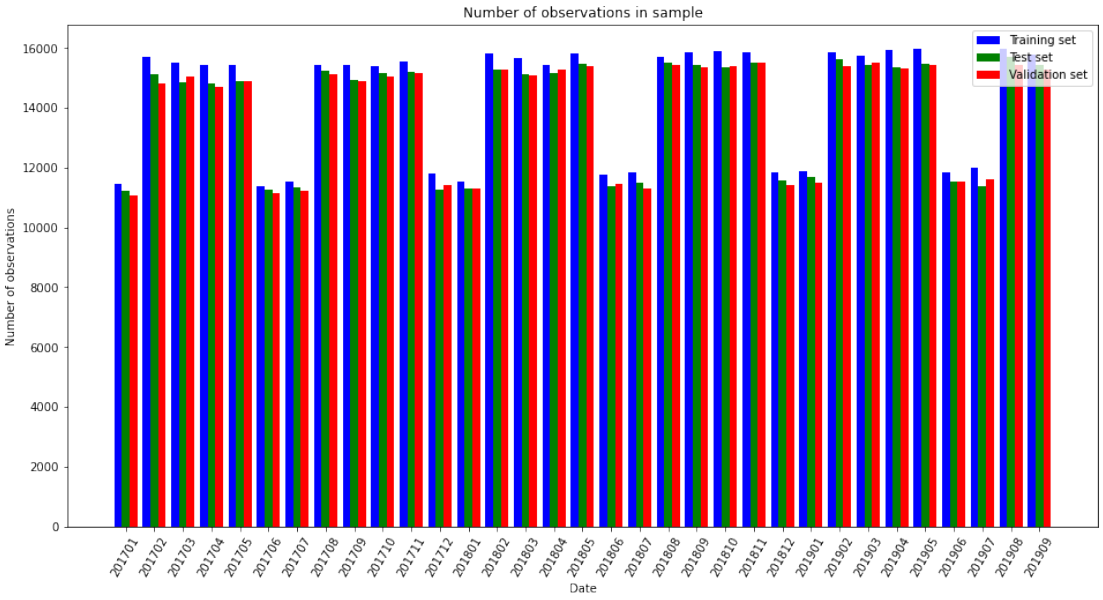
The oversluitmodel hypotheken[7] predicts the probability that a given customer will churn[8]. Every year this occurs for a number of clients, which is obviously undesired by de Volksbank. In an internal study it was found that this number has been steadily increasing over the past decade. In order to reduce this number it can be helpful to contact customers who have a relatively high probability of churning in the foreseeable future. This can enable de Volksbank to resolve issues that might motivate the client to change to another bank.

An analysis of this model, performed by the model validation unit of the bank, shows that using different (fixed) seeds results in materially different predictions. The correlation between predictions of models with a different seed results in 64%-73% correlation (both Pearson and Spearman). This indicates materially different model outcomes and would be a serious shortcoming when this model is used in practice. This is the major motivation for the development of the framework in this paper.

The dataset contains around 500 variables on approximately 1,400,000,000 observations (of which 474,000 were used for training, 460,000 for validation and 461,000 for testing) describing mortgage customers of de Volksbank. The observations are constructed semi-annually over the period January 2017 to September 2019. The original data consisted of monthly observations on each client, but is transformed to remove overlapping windows with the target variable (i.e. multiple 'churn' indicators for one target). This target variable is the churn probability between 3 to 9 months from now. That is, the probability that a given customer will refinance his/her mortgage in 3 to 9 months. This leaves some time to contact the client and try to find out how the bank can improve its services, or offer lower interest rates, as this is often the reason a customer churns. In total 35 variables were selected on their explanatory power to be used in the random forest. The optimized classifier is configured using 500 trees, a tree depth of 5 and a $5:1$ class ratio (i.e. the minority class receives a weight of 5 relative to the majority class). The best split is based on the Gini-coefficient. A graphical representation of the distribution of the training, test and validation set over time is given in Fig. 4. This shows that the three subsets are equally representative of the entire dataset.

[7] English: mortgage refinancing model
[8] Transfer their mortgage to another mortgage provider

Fig. 4: Data distribution of OMH



The model poses a minor financial risk to the bank. The bank might target customers which did not intend to refinance in the first place. This could result in a deterioration of the relationship between the client and the bank. Therefore the model is currently in the pilot phase in which the 'churn' probability is estimated on a monthly basis for all customers in scope of the model. Subsequently, the estimated probabilities are listed in a descending order, from which the top 2000 is selected. Every odd customer (hence 1000 clients) is targeted by the marketing team. The even customers are used as a reference group. This is done to mimic the future application, in which the top 1000 would be targeted. After the pilot phase the effectiveness of the algorithm will be assessed. A list of the variables (with description) is given in Appendix 8.2.1. As part of this pilot phase, the robustness of the model is assessed using the framework developed in this paper.

5.1 Setup of the application

In this experiment three different *designs* are investigated. I refer to the *(person,seed)-design* as *(p,s)-design*; *(person,seed,depth)-design* as *(p,s,d)-design*;*(person,seed,trees)-design* as *(p,s,t)-design*. In order to adequately assess the robustness of the algorithm the framework is applied on two test sets. The first test set is a random subset of the data, to mimic the optimization process. The second test set is a fixed out of time dataset[9]. This imitates the true application of de Volksbank. The out of time dataset

[9] In machine learning one commonly refers to the training-, test- and sometimes validation-set only. I found that it is uncommon to speak about out of time in this setting. However, within the bank

is the last month of the dataset: 2020/05. In the second phase of the application, the facet lengths are altered. This gives an interpretation to the components and coefficients with respect to changes in the *design*. Tab. 11 contains the experiments and their facet lengths. Here n_p is the number of observations in the test set and n_s, n_d and n_t are the respective lenghts of the facets. That is, how many values in the range of the facet are evaluated. Finally, the applications are performed for all customers in the test set as well as the top 1000 customers. The rationale behind this is that the marketing team will only contact the top 1000 customers (i.e. with the highest probability) in the application.

Tab. 11: Type of experiments performed on the oversluitmodel hypotheken.

	Test set	(n_p, n_s, n_d, n_t)
Experiment I	Random	$(575715, 10, 11, 10)$
Experiment II	2020/05	$(47314, 10, 11, 10)$
Experiment III	2020/05	$(47314, 25, 11, 10)$
Experiment IV	2020/05	$(47314, 10, 5, 5)$
Experiment V	2020/05	$(47314, 5, 11, 10)$
Experiment VI	2020/05	$(47314, 5, 5, 5)$

For the *seed* facet, the possible values are selected in the range from 100 to 1000. Hence, in the case of 25 seeds the values are equally distributed in this range. For *depth* it is in the range 5 to 15, as these are the values which were investigated by the artificial intelligence department of de Volksbank when optimizing this model. The *trees* facet contains values in the range from 300-900, as this is deemed a reasonable range around the selected 500 trees.

The rationality of choosing these facets in the application is based on previous research by the bank and the structure of the algorithm. First, the model validation unit (Model Validation de Volksbank, 2020) found a large impact when using different seeds. This initiated the demand for more research into the effects of choosing a different seed on the model outcomes. Second, the artificial intelligence department of de Volksbank developed the model by testing the effect of different depths, and observed a better model performance by choosing a lower tree depth. The impact of the number of estimators was not investigated heavily and is therefore chosen to assess this decision. Furthermore, the number of trees is expected to have a limited impact on the model after a certain threshold is passed (i.e. a certain number of trees). Therefore, the number of trees is expected to have little influence on the robustness of this model in the range as specified above.

models are usually tested on data outside of the time range of the training- and test-set to validate the performance of the model 'out of time'. With the out of time set I thus refer to data which is 'in the future' with respect to the data on which the model is developed. I will stick to this notation as it is used frequently in this manner in the banking sector

Since the relative ranking of customers is important in this framework we first split the dataset in a development set ($0.67N$) and a test set ($0.33N$) with $N = 1744590$ observations. Predictions are made on this test set with the original ('true') model configuration. Then, the development set is split in a training- and cross-validation-set in each *facet* combination, choosing the random state equal to the selected seed s for that iteration. Hence, in each iteration we select a different portion of our data to train our model on, but the test set is fixed. This will thus result in an 34/33/33%-train/validation/test-split. The model is trained on the training set, and estimated on the (fixed) test set. Fixing the test set is a requirement of this analysis, as it enables us to compare variance among predictions, and by changing the test set in each iteration one would compare different customers with each other. The algorithm is applied for: (i) the top 1000 customers selected using the 'true' model (i.e. the model developed by the AICoE team); (ii) comparing how often these top 1000 customers are also in the top 1000 for the models within the algorithm (i.e. the 'overlap'); (iii) all customers in the test set.

(i)-(iii) will be applied to the six experiments (I-VI) in Tab. 11. For each experiment there are three different designs (*(p,s)*, *(p,d,s)*, *(p,t,s)*) and two applications (top 1000 and all customers). The results are reported in the next section.

5.2 Results

The results between experiment I and II are discussed first with the aim of investigating difference between a random test set and an out of time set. It is shown that the latter is more robust. Then, the facets are altered for the out of time test set in experiment III-VI. The measures of generalizability receive an interpretation using the observed overlap in Section 5.3.

Tab. 12: Estimated variance components for oversluitmodel hypotheken for experiment I.

(p,s)-design			**(p,s,d)-design**			**(p,s,t)-design**		
	$\sigma^2(\alpha)$			$\sigma^2(\alpha)$			$\sigma^2(\alpha)$	
α	1000	all	α	1000	all	α	1000	all
p	0.000802	0.000864	*p*	0.001721	0.000817	*p*	0.000790	0.000866
s	0.000031	0.000000	*s*	0.000040	0.000000	*s*	0.000030	0.000000
ps	0.000219	0.000010	*d*	0.000585	0.000014	*t*	0.000000	0.000000
			ps	0.000745	0.000021	*ps*	0.000216	0.000010
			pd	0.000795	0.000027	*pt*	0.000000	0.000000
			ds	0.000006	0.000000	*ts*	0.000000	0.000000
			pds	0.000250	0.000008	*pts*	0.000004	0.000000
total	0.001052	0.000874	*total*	0.004142	0.000887	*total*	0.001040	0.000876

Tab. 13: Estimated variance components for oversluitmodel hypotheken for experiment II.

(p,s)-design			**(p,s,d)-design**			**(p,s,t)-design**		
	$\sigma^2(\alpha)$			$\sigma^2(\alpha)$			$\sigma^2(\alpha)$	
α	1000	all	α	1000	all	α	1000	all
p	0.000996	0.001313	p	0.002574	0.001368	p	0.000998	0.001315
s	0.000018	0.000001	s	0.000008	0.000001	s	0.000018	0.000001
ps	0.000310	0.000017	d	0.000622	0.000019	t	0.000000	0.000000
			ps	0.000465	0.000031	ps	0.000312	0.000017
			pd	0.000454	0.000041	pt	0.000000	0.000000
			ds	0.000004	0.000000	ts	0.000000	0.000000
			pds	0.000161	0.000013	pts	0.000004	0.000000
total	0.001324	0.001331		0.004288	0.001473		0.001332	0.001333

Tab. 14: Generalizability coefficient and index of dependability for oversluitmodel hypotheken for experiment I and II.

			coefficient	
			$\rho^2(p)$	$\Phi(p)$
Experiment I	*(p.s)-design*	1000	0.97	0.97
		all	1.00	1.00
	(p.s.d)-design	1000	0.92	0.89
		all	0.99	0.99
	(p.s.d)-design	1000	0.97	0.97
		all	1.00	1.00
Experiment II	*(p.s)-design*	1000	0.97	0.97
		all	1.00	1.00
	(p.s.d)-design	1000	0.97	0.95
		all	0.99	0.99
	(p.s.d)-design	1000	0.97	0.97
		all	1.00	1.00

The results of experiment I and II are given in Tab. 12 and 13 respectively. The resulting generalizability coefficient and index of dependability are given for both experiments in Tab. 14. From these results it can be observed that the total variance increases from experiment I to II. This change is mainly due to an increase in the $\sigma^2(p)$ component, possibly due to a lower number of observations in the out of time set (in experiment II). The other components either remain approximately the same or increase (slightly) as well. This is also seen in Tab. 14 where the results favor the application according to experiment II, which is also the application in the pilot phase. The major

difference between coefficients is observed in the *(p,s,d)*-design. For the other two designs the difference is marginal.

The results show that: (i) the *(p,s)-design* and *(p,s,t)-design* are not much different from each other comparing the variance components in Tab. 12. The same can be observed in the tables in Tab. 13 for Experiment II. Therefore, the variance in the *trees*-facet is probably of a negligible magnitude. It shows evidence that the choice of 500 trees does not need much further investigation, as the model is robust to changes in this facet. This confirms prior beliefs; (ii) the results of the *(p,s)-design* show that most of the variance related to the *seed*-facet is due to the interaction term. It can be concluded that the change in variance of predictions over seeds is larger than the change in predictions over seeds; (iii) the variance over the *depth*-facet however, has a larger influence. The total variance nearly quadruples for the top 1000 ranked customers; (iv) this difference diminishes when variation accross all predictions is taken into account. The points (i)-(iv) are substantiated by the generalizability coefficients in Tab. 14. The generalizability of this algorithm decreases by adding the depth facet. As discussed before, since *depth* is a fundamental parameter to the algorithm this is expected by design.

The tables 15, 16, 17 and 18 contain the results of experiments III-VI respectively. The difference with the results in Tab. 13 is minor. We observe a slight increase in the variance components $\sigma^2(s)$, $\sigma^2(d)$ and $\sigma^2(t)$ with a decreasing facet size. The same holds for the interaction terms related to an increase in either one of the facets related to these terms. The total variances do not indicate a large shift either.

A higher variance in one facet in combination with a lower facet size causes the denominator to increase, and hence both the generalizability coefficient and index of dependability to decrease. This is also confirmed by the results in Tab. 19, which contains the estimated generalizability coefficient and index of dependability for experiment III-VI. For example, the decrease in the *depth* facet size (experiment IV and VI have $n_d = 5$ whereas experiment III and V have $n_d = 11$) causes a slight drop in both coefficients for the variation in the top 1000 ranked clients. The reported generalizability decreases from 0.98 to 0.90. The *seed* facet shows similar behaviour as the facet size (experiment III has $n_s = 25$, IV has $n_s = 10$ and experiment V and VI have $n_s = 5$) drops. The reported generalizabilty decreases from 0.99 to 0.94.

Tab. 15: Estimated variance components for oversluitmodel hypotheken, experiment III.

(p,s)-design			**(p,s,d)-design**			**(p,s,t)-design**		
	$\sigma^2(\alpha)$			$\sigma^2(\alpha)$			$\sigma^2(\alpha)$	
α	1000	all	α	1000	all	α	1000	all
p	0.001033	0.001307	*p*	0.002458	0.001359	*p*	0.001036	0.001307
s	0.000032	0.000001	*s*	0.000041	0.000001	*s*	0.000033	0.000001
ps	0.000279	0.000017	*d*	0.000593	0.000018	*t*	0.000000	0.000000
			ps	0.000486	0.000340	*ps*	0.000277	0.000017
			pd	0.000438	0.000040	*pt*	0.000000	0.000000
			ds	0.000005	0.000000	*ts*	0.000000	0.000000
			pds	0.000163	0.000013	*pts*	0.000003	0.000000
total	0.001344	0.001325		0.004184	0.001771		0.001349	0.001325

Tab. 16: Estimated variance components for oversluitmodel hypotheken, experiment IV.

(p,s)-design			**(p,s,d)-design**			**(p,s,t)-design**		
	$\sigma^2(\alpha)$			$\sigma^2(\alpha)$			$\sigma^2(\alpha)$	
α	1000	all	α	1000	all	α	1000	all
p	0.000996	0.001313	*p*	0.002222	0.001322	*p*	0.001001	0.001316
s	0.000018	0.000001	*s*	0.000009	0.000001	*s*	0.000001	0.000001
ps	0.000310	0.000017	*d*	0.000764	0.000027	*t*	0.000000	0.000000
			ps	0.000413	0.000028	*ps*	0.000313	0.000017
			pd	0.000632	0.000055	*pt*	0.000000	0.000000
			ds	0.000005	0.000000	*ts*	0.000000	0.000000
			pds	0.000193	0.000016	*pts*	0.000004	0.000000
total	0.001324	0.001331		0.004238	0.001449		0.001319	0.001334

Tab. 17: Estimated variance components for oversluitmodel hypotheken, experiment V.

(p,s)-design			**(p,s,d)-design**			**(p,s,t)-design**		
	$\sigma^2(\alpha)$			$\sigma^2(\alpha)$			$\sigma^2(\alpha)$	
α	1000	all	α	1000	all	α	1000	all
p	0.001170	0.001254	*p*	0.002535	0.001325	*p*	0.001173	0.001258
s	0.000055	0.000002	*s*	0.000092	0.000001	*s*	0.000058	0.000002
ps	0.000347	0.000025	*d*	0.000595	0.000019	*t*	0.000000	0.000000
			ps	0.000570	0.000039	*ps*	0.000342	0.000024
			pd	0.000487	0.000040	*pt*	0.000000	0.000000
			ds	0.000010	0.000000	*ts*	0.000001	0.000000
			pds	0.000168	0.000014	*pts*	0.000003	0.000000
total	0.001572	0.001281		0.004457	0.001438		0.001577	0.001284

Tab. 18: Estimated variance components for oversluitmodel hypotheken, experiment VI.

(p,s)-design			**(p,s,d)-design**			**(p,s,t)-design**		
	$\sigma^2(\alpha)$			$\sigma^2(\alpha)$			$\sigma^2(\alpha)$	
α	1000	all	α	1000	all	α	1000	all
p	0.001170	0.001254	*p*	0.002179	0.001277	*p*	0.001175	0.001258
s	0.000055	0.000002	*s*	0.000088	0.000001	*s*	0.000057	0.000002
ps	0.000347	0.000025	*d*	0.000729	0.000026	*t*	0.000000	0.000000
			ps	0.000516	0.000036	*ps*	0.000338	0.000024
			pd	0.000684	0.000056	*pt*	0.000000	0.000000
			ds	0.000014	0.000000	*ts*	0.000001	0.000000
			pds	0.000202	0.000016	*pts*	0.000004	0.000000
total	0.001572	0.001281		0.004412	0.001412		0.001575	0.001284

Tab. 19: Generalizability coefficient and index of dependability for oversluitmodel hypotheken for experiment III-VI.

			coefficient	
			$\rho^2(p)$	$\Phi(p)$
Experiment III	*(p.s)-design*	1000	0.99	0.99
		all	1.00	1.00
	(p.s.d)-design	1000	0.98	0.96
		all	1.00	1.00
	(p.s.d)-design	1000	0.99	0.99
		all	1.00	1.00
Experiment IV	*(p.s)-design*	1000	0.97	0.97
		all	1.00	1.00
	(p.s.d)-design	1000	0.93	0.87
		all	0.99	0.99
	(p.s.d)-design	1000	0.97	0.97
		all	1.00	1.00
Experiment V	*(p.s)-design*	1000	0.94	0.94
		all	1.00	1.00
	(p.s.d)-design	1000	0.94	0.92
		all	0.99	0.99
	(p.s.d)-design	1000	0.94	0.94
		all	1.00	1.00
Experiment VI	*(p.s)-design*	1000	0.94	0.94
		all	1.00	1.00
	(p.s.d)-design	1000	0.90	0.84
		all	0.99	0.98
	(p.s.d)-design	1000	0.95	0.94
		all	1.00	1.00

5.3 Interpretation of results

In order to give an interpretation of the computed coefficients, the overlap between the top 1000 ranked customers and an instance of the model is used to make a comparison. The average overlap over all experiments is reported in Tab. 20. A heatmap indicating the overlap with respect to different combinations of facet values is given in Fig. 5 and 6. It is observed that the setup of Experiment II (used in the pilot phase) has a much greater overlap than Experiment I. This discredits sampling randomly over time and backs taking fixed windows, as sampling randomly over time changes the output of a model to a large extent. Also, the influence of depth is once again prevalent in these results. We see that even in Experiment II, on average just 2 out of 3 customers in the top 1000 re-emerge in the altered designs. For *seed* and *trees* this fraction is much higher, both 87%. This is also seen in the heatmaps, where a higher depth results in a

materially higher overlap, irrespective of the seed. The overlap for *seed* and *trees* remains stable. This provides additional insight on the findings of the model validation unit of de Volksbank for using different seeds. It now becomes apparent that the differences between models using a different seed are not that large, considering that we use the model from Experiment II. The impact of different tree depths is much larger. Since this is also one of the most defining parameters in the random forest algorithm this it is in my opinion advisable not to discard the model altogether but do take it into account when interpreting its results. The effect of optimizing the algorithm based on the out of time test set is also much larger.

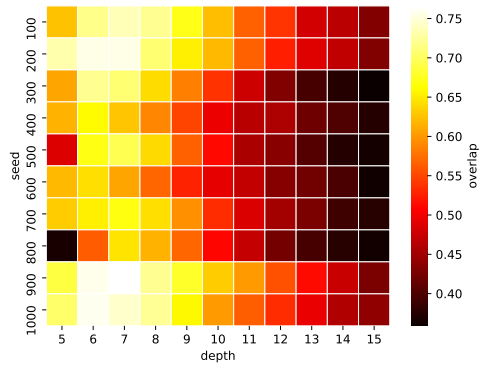
In this paper it has been frequently claimed that the tree depth is a more 'fundamental' parameter to the algorithm. The evidence is in this section. The overlap and generalizability over all designs and experiments have one strong feat in common. They are both materially lower if it contains the *depth*-facet. The decrease in generalizabilty for the *depth*-facet which has been observed in Section 5.2 corresponds with our observations in Tab. 20 and the heatmaps in Fig. 5 and 6. Changes in the generalizability coefficient indicate changes in model performance for different parameter configurations.

Tab. 20: Average overlap between selected target customers.

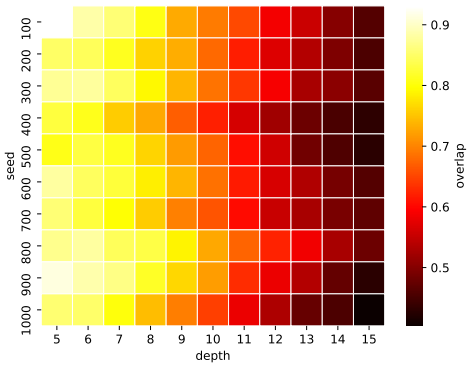
		overlap
Experiment I	*(p.s)-design*	0.61
	(p.s.d)-design	0.55
	(p.s.t)-design	0.61
Experiment II	*(p.s)-design*	0.87
	(p.s.d)-design	0.67
	(p.s.t)-design	0.87
Experiment III	*(p.s)-design*	0.86
	(p.s.d)-design	0.67
	(p.s.t)-design	0.86
Experiment IV	*(p.s)-design*	0.87
	(p.s.d)-design	0.68
	(p.s.t)-design	0.87
Experiment V	*(p.s)-design*	0.89
	(p.s.d)-design	0.70
	(p.s.t)-design	0.89
Experiment VI	*(p.s)-design*	0.89
	(p.s.d)-design	0.70
	(p.s.t)-design	0.89

Fig. 5: Heatmap of overlap for *(p,s,d)-design* for all experiments

(a) Experiment I

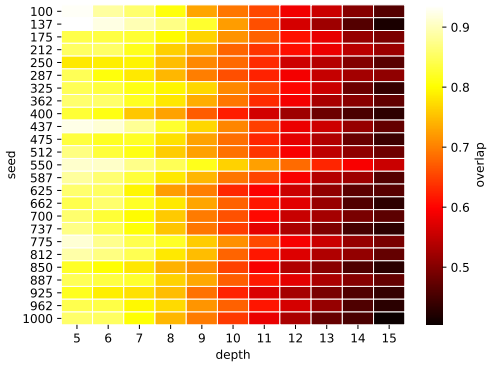


(b) Experiment II

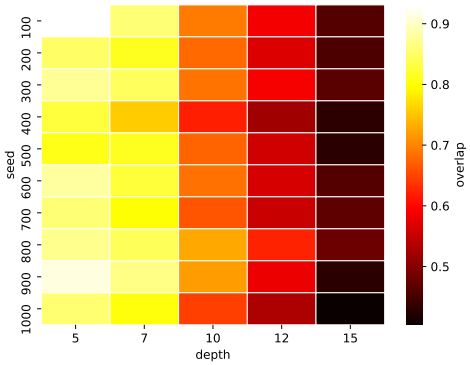


(c) Experiment III

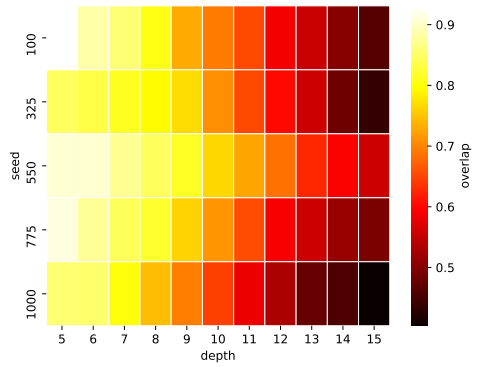
(d) Experiment IV



(e) Experiment V



(f) Experiment VI



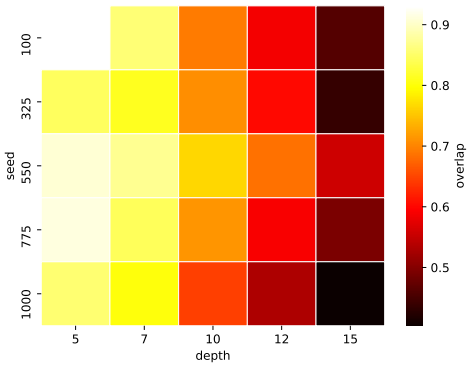
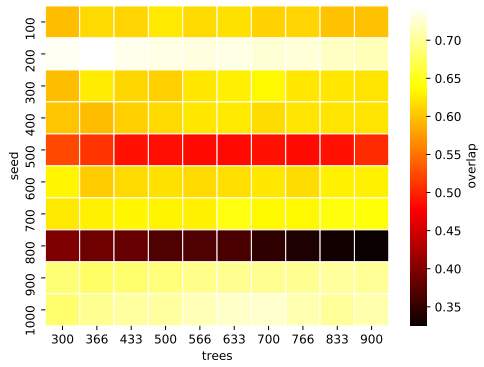
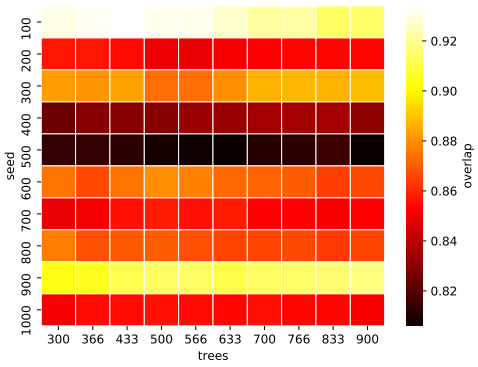


Fig. 6: Heatmap of overlap for *(p,s,t)-design* for all experiments

(a) Experiment I

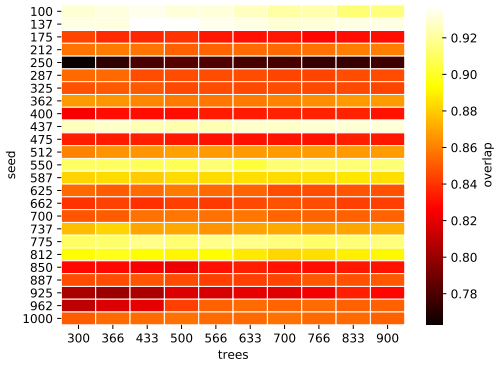


(b) Experiment II

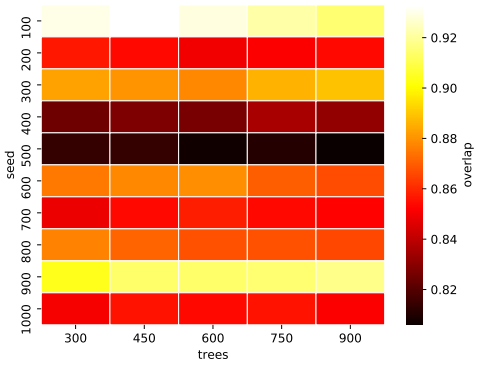


(c) Experiment III

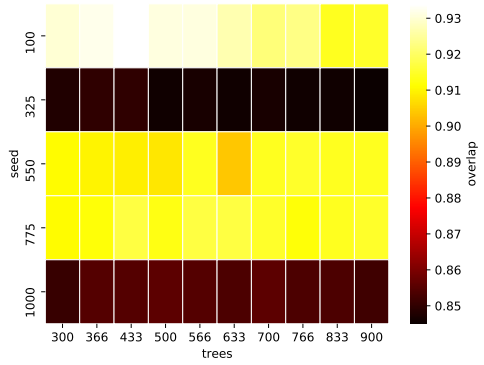
(d) Experiment IV

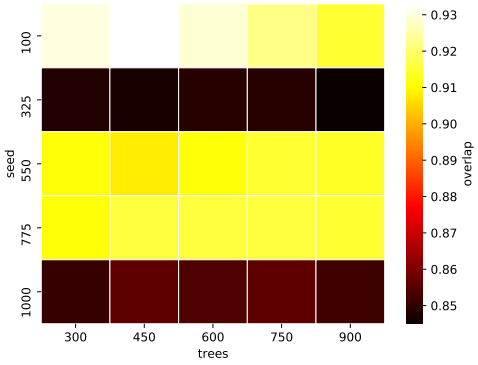


(e) Experiment V



(f) Experiment VI





5.4 Model observations

This section contains the main results based on the analyses performed in Section 5, with some short comments.

Observation I: the influence of using different seeds exists, but is minor

It has been observed that the variation due to the *seed* facet is relatively small. The variance due to using different seeds also increases for assessing fewer seed values. The generalizability decreased from 0.99 to 0.94 for a decrease in seed values from 25 to 5 respectively. However, this study does not confirm the strong dependence on the random state found by the model validation unit.

Observation II: the results from the study suggest a lower tree depth is better

Fig. 5 and 6 show unanimously that there is a higher overlap between models using a lower tree depth. This is an important finding, as it backs up the selection method performed in the modelling phase. For the lowest tree depth of 5 the overlap is generally in-between 0.90 and 1.00. The average overlap reported in Tab. 20 might give a distorted view, as this also takes the higher tree depths into account. It can be seen from the heatmaps that the overlap for a tree depth of 5 is greater than 0.90.

Observation III: the model performs better on an out of time test set

The difference between experiment I and II shows that an out of time test set is preferred when the application is forecasting. This could be taken into account in future iterations of this model or others. When an algorithm is tuned for out of time data, the test set should be constructed in the same way. From Fig. 5 and 6 and Tab. 20 it can also be concluded that the overlap for experiment I is materially lower in comparison with the other experiments.

Observation IV: the results from the study suggest the number of trees is not particularly relevant

The results confirm the prior beliefs of the artificial intelligence unit of the bank that the number of trees has no material impact on the model results.

Observation V: there is materially more variation in the top 1000 compared to the entire test set

For all experiments the generalizability for assessing the entire test set is higher than 0.98. This indicates that the ranking of the top 1000 might not be as stable as the

predictions of the customers with lower scores. It is also confirmed by the overlap which is never higher than 0.90.[10]

[10] 90% of data is on average the same as in the model in the pilot phase.

6 Concluding Remarks

Due to the increased use of artificial intelligence in the financial sector, the demand for this variance decomposition framework is justified. Major European financial regulators have stressed the importance of improving the assessment of explainability, transparency and robustness. This paper aims to contribute to that end. There is an important role for model validators in this respect, as techniques to test these requirements need to be developed, implemented and continuously improved. The framework laid out in this paper contributes to the assessment of the robustness of a random forest algorithm.

The mathematical background – which is relatively simple – is expanded for 2 and 3 facets. From this point the connection is made to the origination of the components from the law of total variance. This gives more insight into the structure of the variance decomposition. Mathematically, it is easily extended to any number of facets. It quickly gets computationally expensive with the computational cost more than doubling for each additional facet with the facet size kept equal. It is therefore preferred to keep the number of facets to a minimum.

Subsequently, the adequacy of the framework is assessed by Monte Carlo simulations. In the first experiment, a matrix of predictions with predetermined variance components is generated. The framework produces unbiased and consistent estimates of these components. The second experiment shows how the framework is able to produce estimates of variance components for individual datapoints in random forests. For the *depth* parameter the influence is shown to be relatively large in comparison to the prediction error variance. This indicates the importance of distinguishing between variance components in a validation of these algorithms, in contrast to just assuming the more standard sources of variance. This second experiment also shows the motivation for setting (possible) negative variance estimates to zero.

In an application on the oversluitmodel hypotheken – which estimates the churn probability of mortgage customers of de Volksbank – the framework provides consistent estimates. The same is concluded when accounting for changes in the facet sizes. To give the coefficient of generalizability an interpretation, a connection is made to the overlap with the 'true' model between the highest probabilities. This shows that the experiments with a higher overlap generally have a higher generalizability coefficient. The heatmaps show how the observed overlap changes over the facet values.

For the oversluitmodel hypotheken good generalizability to the universe of admissible observations for different *seeds*, *trees* and *depths* is observed. The results show that different *depths* have the largest influence on the model outcomes, which is also to be expected. As *depth* is a fundamental parameter for the working of the algorithm, the results of the study confirm the a priori beliefs. It provides a solid ground to state that the robustness of this model is high, and hence does not pose an obstacle for the application of this model.

This study shows how measures of generalizability can be used to indicate the robustness of a machine learning algorithm with respect to its parameter configuration. Applications are not limited to a specific type of algorithm, as the framework only needs model predictions as input. Model validators can use this framework to efficiently assess the (tuned) optimum of a model at hand. It should be noted that more work is needed to provide a better interpretation of these coefficients. In turn, these interpretations can be used to provide thresholds for generalizability in regulation.

7 Discussion for future research

The application of this framework has only been tested to assess the performance of random forests. Due to the relatively simple structure of this framework and since it is only concerned with model scoring it has the ability to be used in a much wider range of applications. It is not even restricted to the domain of artificial intelligence or machine learning but can also be applied to e.g. assess the performance of the newly developed class of switching ordered probit (swopit) estimator as discussed in the paper by Huismans, Nijenhuis & Sirchenko (2021).

The results of the oversluitmodel hypotheken are showing promising results for increased interpretation of generalizability of random forests. This opens the door for using this method in different applications of machine learning. The interpretation of the realized coefficients is difficult at this stage of the research. It has not been investigated yet what actually is defined as 'good' generalizability of a machine learning algorithm. Part of the interpretations from psychology can be used for machine learning. It is however not extensively applied in the assessment of machine learning algorithms and caution should therefore be taken with the interpretation of these coefficients. More applications of the framework (to different models) could provide a base from where an overview of observed coefficients can be created. This can be parsed into thresholds for generalizability in the manner done for reliability in Evers et al. (2010).

The best possible outcome for this framework would be for regulatory institutions to pick up the notion of generalizability in the assessment of model performance. This would probably increase the amount of research invested into the generalizability. This could increase both the use and interpretation of the framework.

References

Amit, Y. Geman, D. (1997). "Shape quantization and recognition with randomized trees." *Neural Computation.*

Arterberry, B. J., Martens, M. P., Cadigan, J. M., Rohrer, D. (2014). "Application of generalizability theory to the big five inventory." *Personality and individual differences*, 69, 98-103.

Bosnić, Z., Kononenko, I. (2008). "Estimation of individual prediction reliability using the local sensitivity analysis." *Applied Intelligence*, 29, 187–203 (2008).

Bosnić, Z., Kononenko, I. (2008). "Comparison of approaches for estimating reliability of individual regression predictions." *Elsevier*, 67, 504-516 (2008).

Bosnić, Z., Kononenko, I. (2009). "An overview of advances in reliability estimation of individual predictions in machine learning." *Intelligent Data Analysis*, 13, 385-401 (2009).

Bosnic, Z., Kononenko, I. (2010). "Automatic selection of reliability estimates for individual regression predictions." *Knowledge Eng. Review.* 25, 27-47 (2010).

Bousquet, O., Elisseeff, A. (2001). Algorithmic stability and generalization performance. *Advances in Neural Information Processing Systems*, 196-202.

Bousquet, O., Elisseeff, A. (2002). Stability and generalization. *The Journal of Machine Learning Research*, 2, 499-526.

Breierova, L. (1996). An introduction to sensitivity analysis. *MIT system dynamics in education project.*

Breiman, L. (2001). "random forests." *Machine Learning*, 45, 5–32.

Brennan, R.L., (2001). "Generalizability Theory - Statistics for social science and public policy" *Springer-Verlag Berlin Heidelberg New York.*

Brennan, R. L. (2010). Generalizability theory and classical test theory. *Applied Measurement in Education*, 24(1), 1-21.

Brito Filho, D. A., Artes, R. (2018). "Application of bayesian additive regression trees in the development of credit scoring models in Brazil." *Production*, 28.

Brown, I., Mues, C. (2012). "An experimental comparison of classification algorithms for imbalanced credit scoring data sets." *Expert Systems with Applications (Elsevier)*, 39(3), 3446-3453.

Cronbach, L.J. (1972). "The dependability of behavioral measurements." *Theory of generalizability for scores and profiles* (1972): 1-33.

Crouhy, M., Galai, D., Mark, R. (2000). "A comparative analysis of current credit risk models." *Journal of Banking Finance* , 24(1-2), 59-117.

Dietterich, T.G. (2000). "An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization." Machine Learning, 2000, 40, 139-157.

De Nederlandsche Bank (DNB) (2019). "General principles for the use of Artificial Intelligence in the financial sector."

European Banking Authority (2020, January). *Final report on big data and advanced analytics* [Press Release]. Retrieved from `https://www.eba.europa.eu/file/609786/`

European Commission (2021, April 21). *Europe fit for the Digital Age: Commission proposes new rules and actions for excellence and trust in Artificial Intelligence* [Press Release]. Retrieved from `https://ec.europa.eu/commission/presscorner/detail/en/IP_21_1682`

Efron, B. (1992). "Bootstrap methods: another look at the jackknife." *Breakthroughs in statistics*. Springer, New York, NY, 1992. 569-593.

Efron, B. (2014). "Estimation and accuracy after model selection." *Journal of the American statistical association*. 109(507),991-1007.

Evers, A., Lucassen, W., Meijer, R. Sijtsma, K. (2010). *Beoordelingssysteem voor de kwaliteit van tests* [Press Release]. Retrieved from `https://www.cotandocumentatie.nl/cotan/beoordelingssysteem/`.

Fisher, R.A. (1925). "Statistical Methods for Research workers." Oliver & Boyd (Edinburgh).

Graham, S., Hebert, M., Paige Sandbank, M., Harris, K. R. (2016). "Assessing the writing achievement of young struggling writers: Application of generalizability theory." *Learning Disability Quarterly*, 39(2), 72-82.

Kruppa, J., Schwarz, A., Arminger, G., Ziegler, A. (2013). Consumer credit risk: Individual probability estimates using machine learning. *Expert Systems with Applications*, 40(13), 5125-5131.

Kukar, M., Kononenko, I. (2002). Reliable classifications with machine learning. *In European Conference on Machine Learning (pp. 219-231)*. Springer, Berlin, Heidelberg.

Ho, T.K. (1998). "The Random Subspace Method for Constructing Decision Forests" *Transactions on pattern analysis and machine intelligence*, 1998, 20(8), 832-844.

Huismans, J., Nijenhuis, J.W. & Sirchenko, A. (2021). "A mixture of ordered probit models with endogenous assignment to two latent classes" *Amsterdam School of Economics Discussion Paper*, 2021. Available at UvA.

Mitchell, M.W. (2011). "Bias of the random forest Out-of-Bag (OOB) Error for Certain Input Parameters." Open Journal of Statistics, 2011, 1, 205-211.

Mushquash, C., O'Connor, B. P. (2006). "SPSS and SAS programs for generalizability theory analyses." *Behavior research methods*, 38(3), 542-547.

Model Validation de Volksbank (2020). "Pilot Validation Report OMH 1.0."

Nelder, J.A. (1968). "The Combination of Information in Generally Balanced Designs." Journal of the Royal Statistical Society. Series B (Methodological), 1968, 30(2), 303-311.

Oosterwijk, P. R., Van der Ark, L. A., Sijtsma, K. (2019). "Using Confidence Intervals for Assessing Reliability of Real Tests." *Assessment*, 26(7), 1207-1216.

Patterson, H.D., Thompson, R. (1971). "Recovery of Inter-Block Information when Block Sizes are Unequal." *Biometrika*, 58 (1971) 545-554.

Probst, P., Boulesteix, A. (2018). "To Tune or Not to Tune the Number of Trees in random forest." *Journal of Machine Learning Research*, 18 (2018) 1-18.

Sijtsma, K., Van der Ark, L. A. (2015). "Conceptions of reliability revisited and practical recommendations." *Nursing research*, 64(2), 128-136.

Shavelson, R. J., Webb, N. M. (1981). Generalizability theory: 1973–1980. *British Journal of Mathematical and Statistical Psychology*, 34(2), 133-166.

Shavelson, R.J., Webb, M.W., Rowley, L.R. (1989). "Generalizability theory" *American Psychologist*, 1989, 44(6), 922-932.

Shavelson, R. J., Webb, N. M. (1991). Generalizability theory: A primer (Vol. 1). sage.

Virani, N., Iyer, N., Yang, Z. (2020). Justification-based reliability in machine learning. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 34, No. 04, pp. 6078-6085).

Wager, S., Hastie, T., Efron, B. (2014). Confidence intervals for random forests: The jackknife and the infinitesimal jackknife. *The Journal of Machine Learning Research*, 15(1), 1625-1651.

Wager, S. (2014). Asymptotic theory for random forests. *arXiv preprint arXiv:1405.0352*.

8 Appendices

8.1 Appendix A - Mathematical derivations

8.1.1 Deomposition of Sum of Squares in 2 facet design

$$
\begin{aligned}
&\mathrm{SS} = \textstyle\sum_d^{n_d}\sum_s^{n_s}(X_{ds}-\bar{X})^2 \overset{1}{=} \sum_d^{n_d}\sum_s^{n_s}((\bar{X}_d-\bar{X})+(\bar{X}_s-\bar{X})+(X_{ds}-\bar{X}_d-\bar{X}_s+\bar{X}))^2\\
&= \textstyle\sum_d^{n_d}\sum_s^{n_s}(\bar{X}_d-\bar{X})^2+\sum_d^{n_d}\sum_s^{n_s}(\bar{X}_s-\bar{X})^2+\sum_d^{n_d}\sum_s^{n_s}(X_{ds}-\bar{X}_d-\bar{X}_s+\bar{X})^2\\
&+2\textstyle\sum_d^{n_d}\sum_s^{n_s}(\bar{X}_d-\bar{X})(\bar{X}_s-\bar{X})+2\sum_d^{n_d}\sum_s^{n_s}(\bar{X}_d-\bar{X})(X_{ds}-\bar{X}_d-\bar{X}_s+\bar{X})\\
&+2\textstyle\sum_d^{n_d}\sum_s^{n_s}(\bar{X}_s-\bar{X})(X_{ds}-\bar{X}_d-\bar{X}_s+\bar{X})\\
&= n_s\textstyle\sum_d^{n_d}(\bar{X}_d-\bar{X})^2+n_d\sum_s^{n_s}(\bar{X}_s-\bar{X})^2+\sum_d^{n_d}\sum_s^{n_s}(X_{ds}-\bar{X}_d-\bar{X}_s+\bar{X})^2\\
&+2\textstyle\sum_d^{n_d}\sum_s^{n_s}(\bar{X}_d-\bar{X})(\bar{X}_s-\bar{X})\\
&-2\textstyle\sum_d^{n_d}\sum_s^{n_s}(\bar{X}_d-\bar{X})(\bar{X}_s-\bar{X})+2\sum_d^{n_d}\sum_s^{n_s}(\bar{X}_d-\bar{X})(X_{ds}-\bar{X}_d)\\
&-2\textstyle\sum_d^{n_d}\sum_s^{n_s}(\bar{X}_d-\bar{X})(\bar{X}_s-\bar{X})+2\sum_d^{n_d}\sum_s^{n_s}(\bar{X}_s-\bar{X})(X_{ds}-\bar{X}_s)\\
&= n_s\textstyle\sum_d^{n_d}(\bar{X}_d-\bar{X})^2+n_d\sum_s^{n_s}(\bar{X}_s-\bar{X})^2+\sum_d^{n_d}\sum_s^{n_s}(X_{ds}-\bar{X}_d-\bar{X}_s+\bar{X})^2\\
&-2\textstyle\sum_d^{n_d}\sum_s^{n_s}(\bar{X}_d-\bar{X})(\bar{X}_s-\bar{X})+2\sum_d^{n_d}\sum_s^{n_s}(\bar{X}_d-\bar{X})(X_{ds}-\bar{X}_d)+2\sum_d^{n_d}\sum_s^{n_s}(\bar{X}_s-\\
&\bar{X})(X_{ds}-\bar{X}_s)\\
&= n_s\textstyle\sum_d^{n_d}(\bar{X}_d-\bar{X})^2+n_d\sum_s^{n_s}(\bar{X}_s-\bar{X})^2+\sum_d^{n_d}\sum_s^{n_s}(X_{ds}-\bar{X}_d-\bar{X}_s+\bar{X})^2\\
&-2\textstyle\sum_d^{n_d}\sum_s^{n_s}\bar{X}_d\bar{X}_s+2\sum_d^{n_d}\sum_s^{n_s}\bar{X}_d\bar{X}+2\sum_d^{n_d}\sum_s^{n_s}\bar{X}_s\bar{X}-2\sum_d^{n_d}\sum_s^{n_s}\bar{X}^2\\
&+2\textstyle\sum_d^{n_d}\sum_s^{n_s}\bar{X}_d\bar{X}_{ds}-2\sum_d^{n_d}\sum_s^{n_s}\bar{X}_d^2+2\sum_d^{n_d}\sum_s^{n_s}\bar{X}\bar{X}_{ds}-2\sum_d^{n_d}\sum_s^{n_s}\bar{X}\bar{X}_d\\
&+2\textstyle\sum_d^{n_d}\sum_s^{n_s}\bar{X}_s\bar{X}_{ds}-2\sum_d^{n_d}\sum_s^{n_s}\bar{X}_s^2+2\sum_d^{n_d}\sum_s^{n_s}\bar{X}\bar{X}_{ds}-2\sum_d^{n_d}\sum_s^{n_s}\bar{X}\bar{X}_{ds}\\
&= n_s\textstyle\sum_d^{n_d}(\bar{X}_d-\bar{X})^2+n_d\sum_s^{n_s}(\bar{X}_s-\bar{X})^2+\sum_d^{n_d}\sum_s^{n_s}(X_{ds}-\bar{X}_d-\bar{X}_s+\bar{X})^2\\
&-2\textstyle\sum_d^{n_d}\sum_s^{n_s}\bar{X}_d\bar{X}_s-2\sum_d^{n_d}\sum_s^{n_s}\bar{X}^2\\
&+2\textstyle\sum_d^{n_d}\sum_s^{n_s}\bar{X}_d\bar{X}_{ds}-2\sum_d^{n_d}\sum_s^{n_s}\bar{X}_d^2+2\sum_d^{n_d}\sum_s^{n_s}\bar{X}\bar{X}_{ds}\\
&+2\textstyle\sum_d^{n_d}\sum_s^{n_s}\bar{X}_s\bar{X}_{ds}-2\sum_d^{n_d}\sum_s^{n_s}\bar{X}_s^2+2\sum_d^{n_d}\sum_s^{n_s}\bar{X}\bar{X}_{ds}\\
&= n_s\textstyle\sum_d^{n_d}(\bar{X}_d-\bar{X})^2+n_d\sum_s^{n_s}(\bar{X}_s-\bar{X})^2+\sum_d^{n_d}\sum_s^{n_s}(X_{ds}-\bar{X}_d-\bar{X}_s+\bar{X})^2\\
&-2\textstyle\sum_d^{n_d}\bar{X}_d\sum_s^{n_s}\bar{X}_s-2\sum_d^{n_d}\sum_s^{n_s}\bar{X}^2\\
&+2\textstyle\sum_d^{n_d}\bar{X}_d\sum_s^{n_s}\bar{X}_{ds}-2\sum_s^{n_s}\sum_d^{n_d}\bar{X}_d^2+2\bar{X}\sum_d^{n_d}\sum_s^{n_s}\bar{X}_{ds}\\
&+2\textstyle\sum_s^{n_s}\bar{X}_s\sum_d^{n_d}\bar{X}_{ds}-2\sum_s^{n_s}\sum_d^{n_d}\bar{X}_s^2+2\bar{X}\sum_d^{n_d}\sum_s^{n_s}\bar{X}_{ds}\\
&= n_s\textstyle\sum_d^{n_d}(\bar{X}_d-\bar{X})^2+n_d\sum_s^{n_s}(\bar{X}_s-\bar{X})^2+\sum_d^{n_d}\sum_s^{n_s}(X_{ds}-\bar{X}_d-\bar{X}_s+\bar{X})^2\\
&-4n_dn_s\bar{X}^2\\
&+2n_s\textstyle\sum_d^{n_d}\bar{X}_d^2-2n_s\sum_d^{n_d}\bar{X}_d^2+2n_dn_s\bar{X}^2\\
&+2\textstyle\sum_s^{n_s}\bar{X}_s^2-2\sum_s^{n_s}\sum_d^{n_d}\bar{X}_s^2+2n_dn_s\bar{X}^2\\
&= n_s\textstyle\sum_d^{n_d}(\bar{X}_d-\bar{X})^2+n_d\sum_s^{n_s}(\bar{X}_s-\bar{X})^2+\sum_d^{n_d}\sum_s^{n_s}(X_{ds}-\bar{X}_d-\bar{X}_s+\bar{X})^2\\
&= \qquad SS(d) \qquad + \qquad SS(s) \qquad + \qquad SS(ds)
\end{aligned}
$$

8.1.2 *m*-facet G-study

In this section we develop the framework for a m-facet G-study, and show an application for 3 facets. The notation in this section is based on theory in Brennan (2001) and modified to fit our application. Define ω the set of all facet indices, and Ω the set of indices for any effect in the design. So in the case of a 2-facet design as defined before we have $\omega = \{d, s\}$ and $\Omega = \{d, s, ds\}$. Thus, Ω contains all combinations of the facet indices. This can be elaborated to nested designs, which are out of scope for this paper. For more information on nested designs consider the literature of Brennnan (2001) and Shavelson (1989). Using the analogy developed in the preceding section, for a m-facet G-study an observed score is written as:

$$X_\omega = \mu + \sum_{i \in \Omega} \nu_i \tag{50}$$

So the observed score is again the sum of the grand mean μ and all the score effects. Now define $\alpha \subseteq \omega$ the set of indices relevant to one component in the design, where α contains m primary indices. Furthermore, define $\dot{\alpha} \subset \omega$ but $\alpha \varnothing \dot{\alpha}$. In other words, $\dot{\alpha}$ contains all elements in ω which are not in α. So in the example of a 2-facet design we would have for the effect d that $\alpha = \{d\}$ and $\dot{\alpha} = \{s\}$. We can then write an individual score effect as:

$$\nu_\alpha = \mu_\alpha + \sum_{i \in \Omega} \mathbb{I}(i)\mu_i \tag{51}$$

where

$$\mathbb{I}(i) = \begin{cases} -1 & \text{if } i \text{ contains } m - j \text{ of the primary indices, for } j \text{ odd, } j > 0 \\ 1 & \text{if } i \text{ contains } m - j \text{ of the primary indices, for } j \text{ even, } j > 0 \\ 0 & \text{elsewhere} \end{cases} \tag{52}$$

This means we add all mean scores of components that contain any of the m minus an even number of indices in α, and subtract the mean scores of components which contain any of the m minus an odd number of indices in α. For example, in the 2-facet design we have for the effect ν_{pd} that:

$$\nu_{pd} = \mu_{pd} + \sum_{i \in \Omega} \mathbb{I}(i)\mu_i = \mu_{pd} - \mu_p - \mu_d + \mu \tag{53}$$

as μ_p, μ_d both contain $2 - 1 = 1$ of the primary indices and μ contains $2 - 2 = 0$ of them. Next, define

$$\pi(\dot{\alpha}) = \prod_{i \in \dot{\alpha}} n_i \tag{54}$$

such that it is the product of all sample sizes which are not in α, and 1 if $\alpha = \omega$. The

mean of an component is then written as:

$$\bar{X}_\alpha = \frac{1}{\pi(\dot{\alpha})} \sum_{\dot{\alpha}} X_\omega \tag{55}$$

For example, $X_d = \frac{1}{n_d} \sum_{s=1}^{n_d} X_{pd}$. By replacing all population mean scores by their sample equivalent we write x_α as the sample equivalent of ν_α. Then, one can compute the individual sum of squares as:

$$SS(\alpha) = \pi(\dot{\alpha}) \sum_{\alpha} x_\alpha^2 \tag{56}$$

Hence, we have $SS(d) = n_d \sum_p (\bar{X}_p - \bar{X})^2$ in the 2-facet example of the preceding section. From there it is straightforward to compute the mean squares

$$MS(\alpha) = \frac{SS(\alpha)}{df(\alpha)} \tag{57}$$

The individual variance components are computed from this as

$$\hat{\sigma}^2(\alpha) = \frac{MS(\alpha) + \sum_{j \in \Omega} \mathbb{I}(j) MS(j)}{\pi(\dot{\alpha})} \tag{58}$$

where

$$\mathbb{I}(j) = \begin{cases} -1 & \text{if } j \text{ contains the } m \text{ indices in } \alpha \text{ and } k \text{ of the indices in } \dot{\alpha}, \text{ for } k \text{ odd}, k > 0 \\ 1 & \text{if } j \text{ contains the } m \text{ indices in } \alpha \text{ and } k \text{ of the indices in } \dot{\alpha}, \text{ for } k \text{ even}, k > 0 \\ 0 & \text{elsewhere} \end{cases} \tag{59}$$

Thus, $\hat{\sigma}^2(p) = \frac{MS(p) - MS(pd)}{n_p}$.

8.1.3 Asymptotic theory of random forests

$$\hat{y} = M_T(x|\psi, \Theta, D_n) := M_T(x) \tag{60}$$

be the estimate of a certain forest. It is proven by Wager (2016) that

$$M_T(x) = \lim_{B \to \infty} M_T^B(x) \tag{61}$$

with B the number of bootstrap samples. From this they deduce that the prediction

$$\hat{y} \overset{a}{\sim} \mathcal{N}(\mathbb{E}[\hat{y}], \sigma^2(\hat{y})) \tag{62}$$

The variance of $\hat{y}$ can be consistently estimated by the infinitesimal Jackknife estimator

$$\hat{V}_{ij} = \sum_{i=1}^{N} Cov(M_T(x), N_i*) \tag{63}$$

where N_i* is the number of times the particular observation Z_i occurs in the training data for base learner $M_T(x)$. Wager (2014) discusses also the properties of of the Jackknife-after-bootstrap estimator, also designed by Efron (1992):

$$\hat{V}_J = \frac{N}{N-1} \sum_{i=1}^{N} (\bar{M}_{T(-i)}(x) - \bar{M}_T(x))^2 \tag{64}$$

which is another estimate used for the variance of $\hat{y}$. Here, $\bar{M}_{T(-i)}(x)$ is the tree estimate excluding training sample i averaged over all bootstrap samples, and $\bar{M}_T(x)$ the average over all bootstrap samples. After simulation studies they (Wager (2014,2016)) conclude that even though this is proven for an infinite number of bootstrap replications, bias corrections are necessary for a finite amount of them. They show consistency of these bias-corrected estimates, which can be used as a consistent estimator for the variance in these models. For this paper the mere result that these variances can be derived is enough to provide a solid ground to extend to empirical variance components.

8.2 Appendix B - Application specifications

8.2.1 Risk drivers of the Oversluitmodel Hypotheken

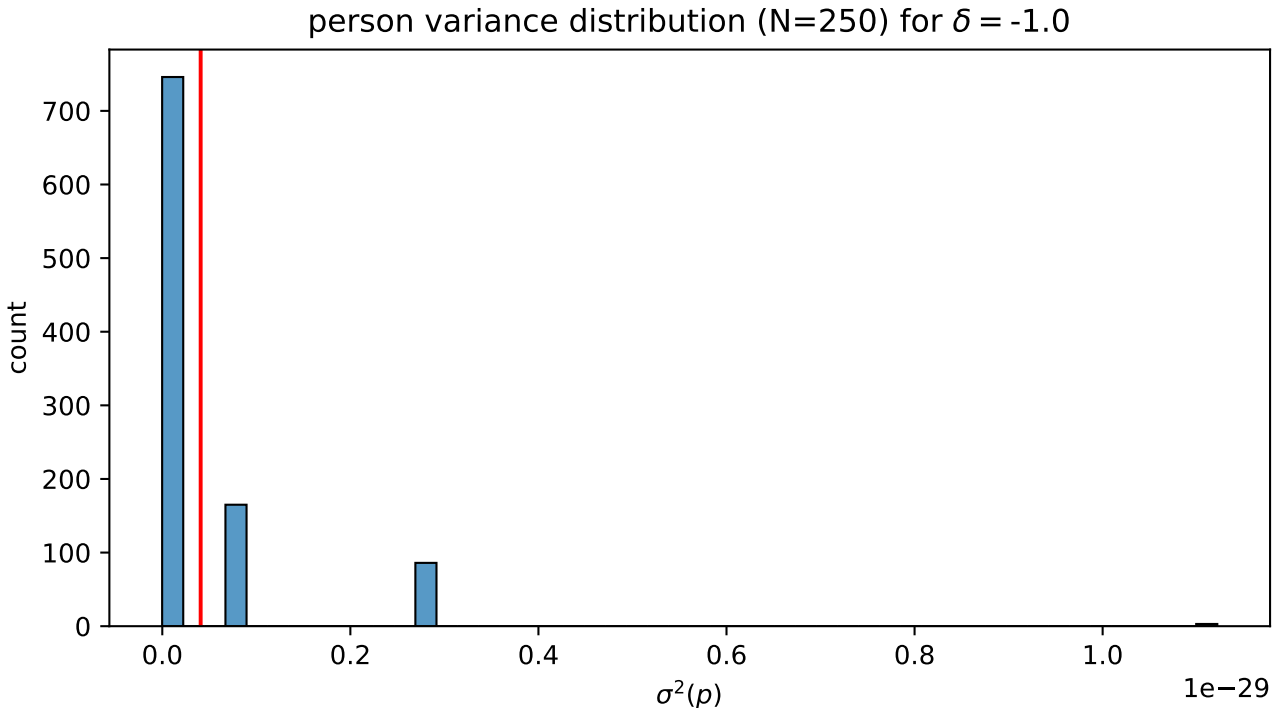
name	description
proxy_rente_incentive_10y',	proxy for 10 year mortgage interest rate decrease a customer could obtain if he/she refinances
'proxy_rente_incentive_10y_pos_part'	proxy for interest
'proxy_rente_incentive_20y',	proxy for 20 year mortgage interest rate decrease a customer could obtain if he/she refinances
'ind_kisg_container_oversluiten_lb3',	index if the customer contacted the bank on refinancing in the last 3 months
'weighted_duur_tot_einde_rvp',	weighted duration until the end of the fixed interest period
'proxy_rente_incentive_20y_pos_part'	proxy for interest
'duur_tot_einde_rvp_min',	minimal duration until end of fixed interest period over all mortgage parts
'ind_kisg_boete_of_aflosnota_lb3',	index if the customer contacted the bank on fines or repayment invoices in the last 3 months
'duur_tot_einde_rvp_max',	maximal duration until end of fixed interest period over all mortgage parts
'ind_kisg_oversluiten_lb3',	index if the customer contacted the bank on refinancing in the last 3 months
'AantalVerkochteWoningen_MA',	moving average of sold houses, proxy for the housing market
'avg_rente_peers_20y',	average 20 year mortgage mortgage interest rate of other banks
'cred_looptijd_verstr_oudst_numc',	the expiration of oldest mortage part of a customer in months
'cred_mnd_hyp_klant_numi',	date at which the customer appears in the dataset for the first time
'hh_loanAge_numi',	difference between current date and loan origination
'cred_looptijd_verstr_numc',	the expiration of this mortage part of a customer in months
'min_rente_peers_20y',	minimum 20 year mortgage mortgage interest rate of other banks
'cred_gem_rente_numc',	average mortgage interest rate over this mortgage (part)

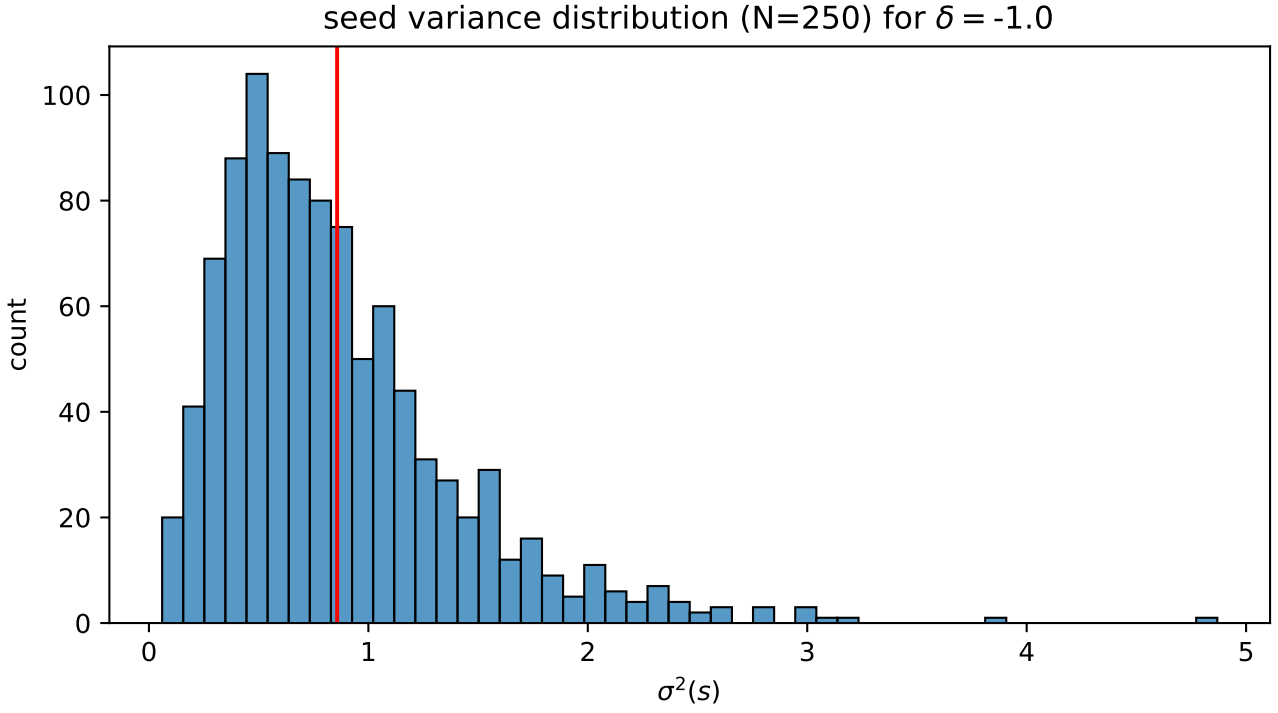
name	description
'weighted_delta5Yklantrente_MA',	weighted average of all differences between the current fixed mortgage interest of de Volksbank and the current mortgage interest of all mortgage parts of the customer
'cred_eam_bg_numc',	exposure at default at current date
'term_gecorr_maandtermijn_nota_bg_numc'	corrected monthly installment
'avg_rente_peers_10y',	average 10 year mortgage mortgage interest rate of other banks
'hh_leeftijd_min_hfd_numi',	minimum age of persons who are head of houshold for this mortgage
'cred_pd_berekend_totaal_numc',	probability of default
'cred_ltv_m_ix_numc',	loan to value of object minus deductible pre-tax
'max_delta5Yklantrente',	maximum change of mortgage interest over past 5 years
'weighted_delta5Yklantrente',	weighted average of changes of mortgage interest over past 5 years
'min_rente_peers_10y',	minimum 10 year mortgage mortgage interest rate of other banks
'sal_gem_saldo_6_maand_numc',	average balance over the past 6 months
'min_delta5Yklantrente'	minimum change of mortgage interest over past 5 years

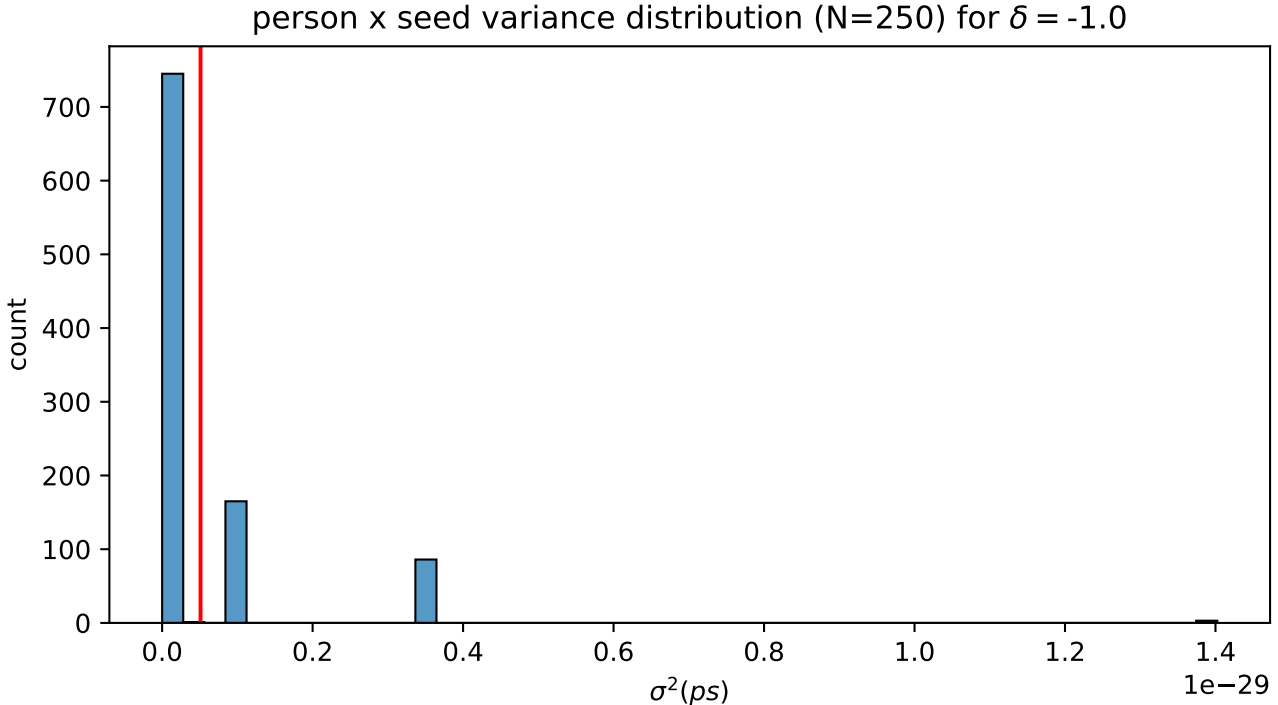
8.3 Appendix C - Figures

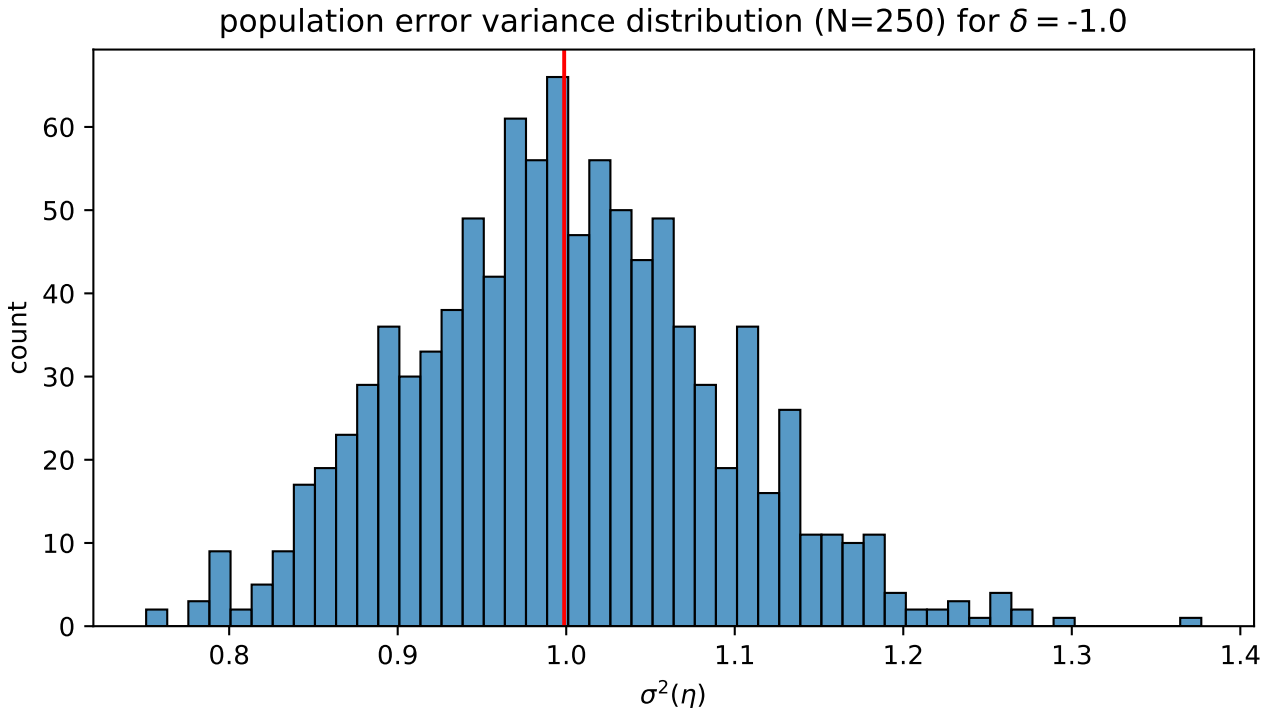
8.3.1 Results of MC study (2 facets)

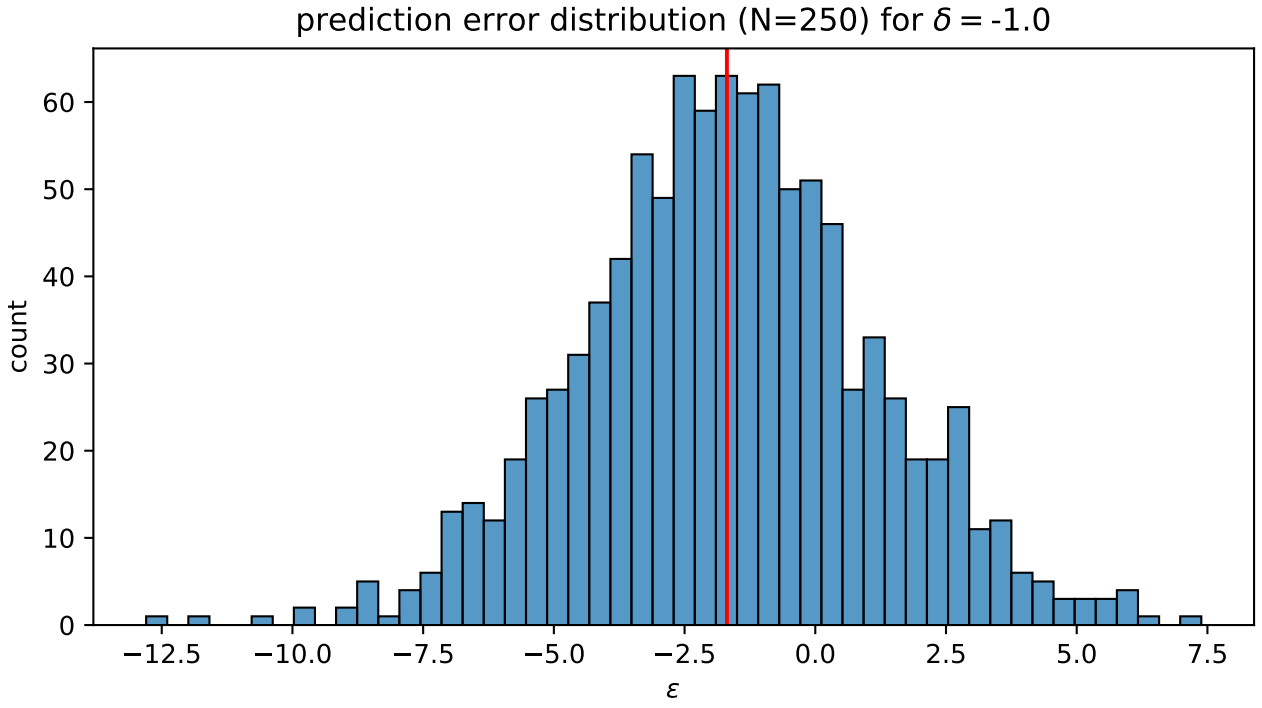
The following pages give an overview of the distribution of variance component estimates from Section 4.2. The horizontal axis shows the value of the estimates, the vertical axis gives the count of each estimate.

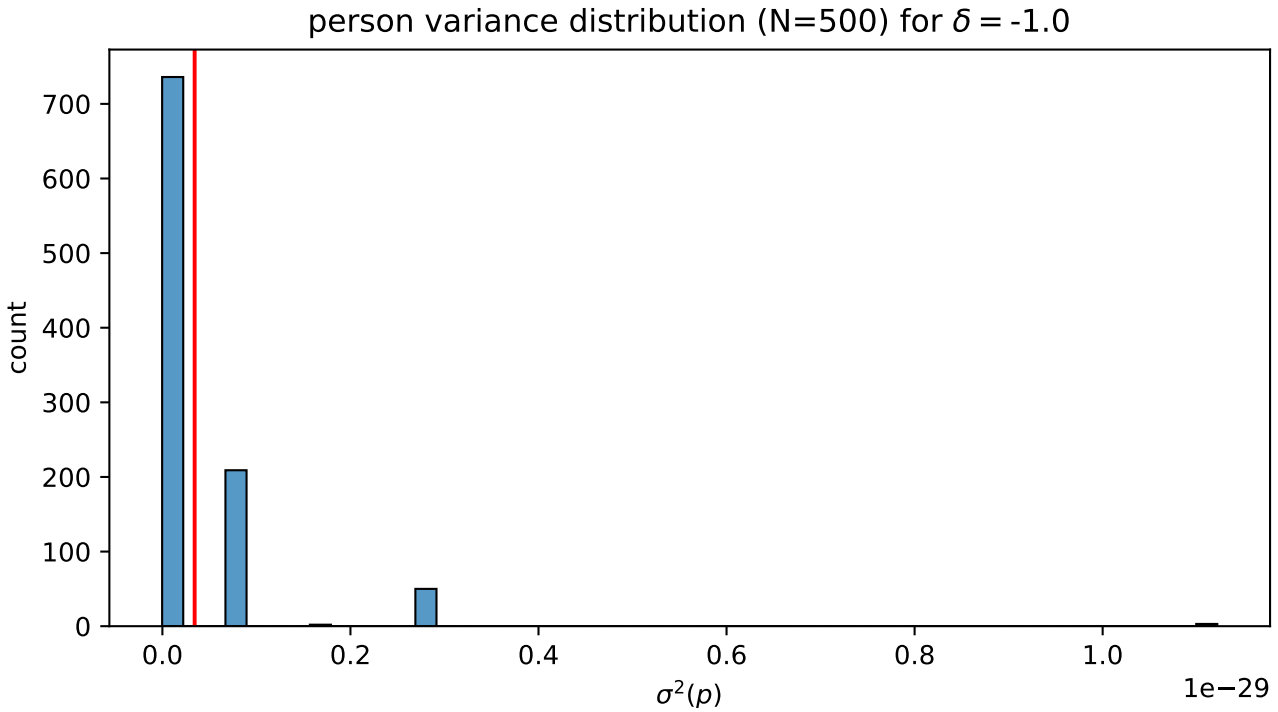

person variance distribution (N=250) for $\delta = -1.0$
count
0
100
200
300
400
500
600
700
0.0
0.2
0.4
0.6
0.8
1.0
$\sigma^2(p)$
1e−29

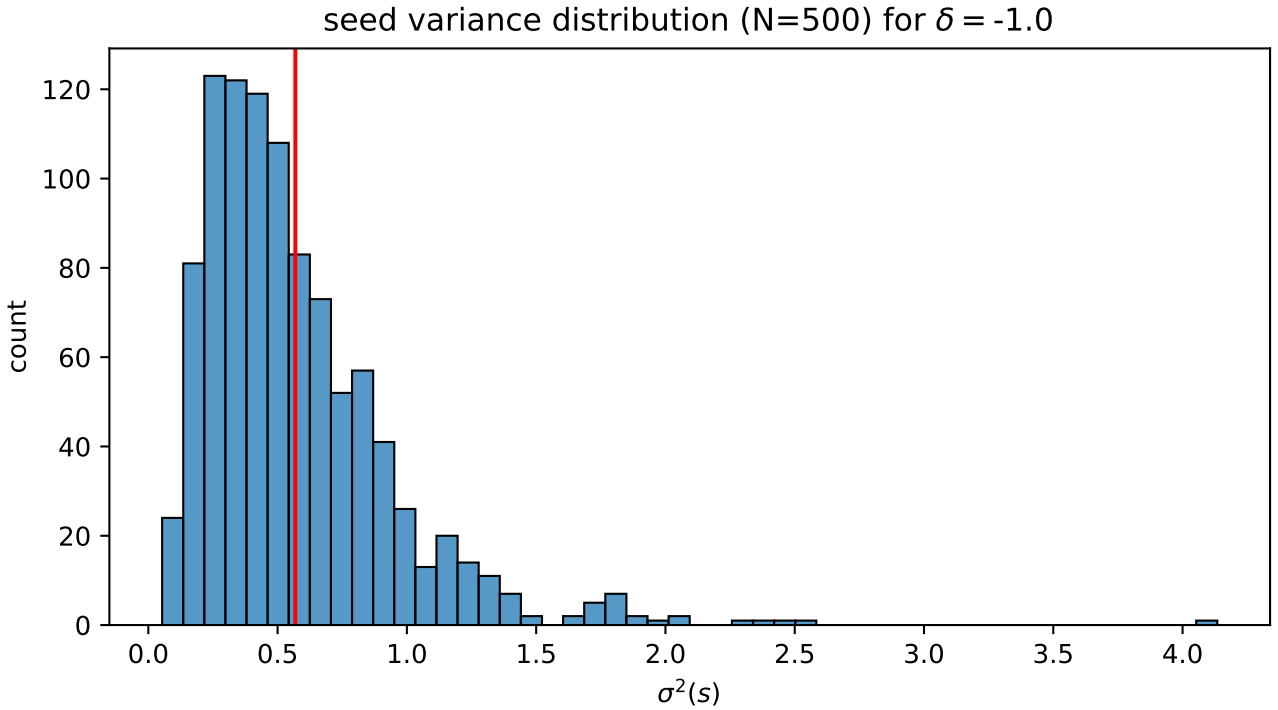

seed variance distribution (N=250) for $\delta = -1.0$
count
0
20
40
60
80
100
0
1
2
3
4
5
$\sigma^2(s)$

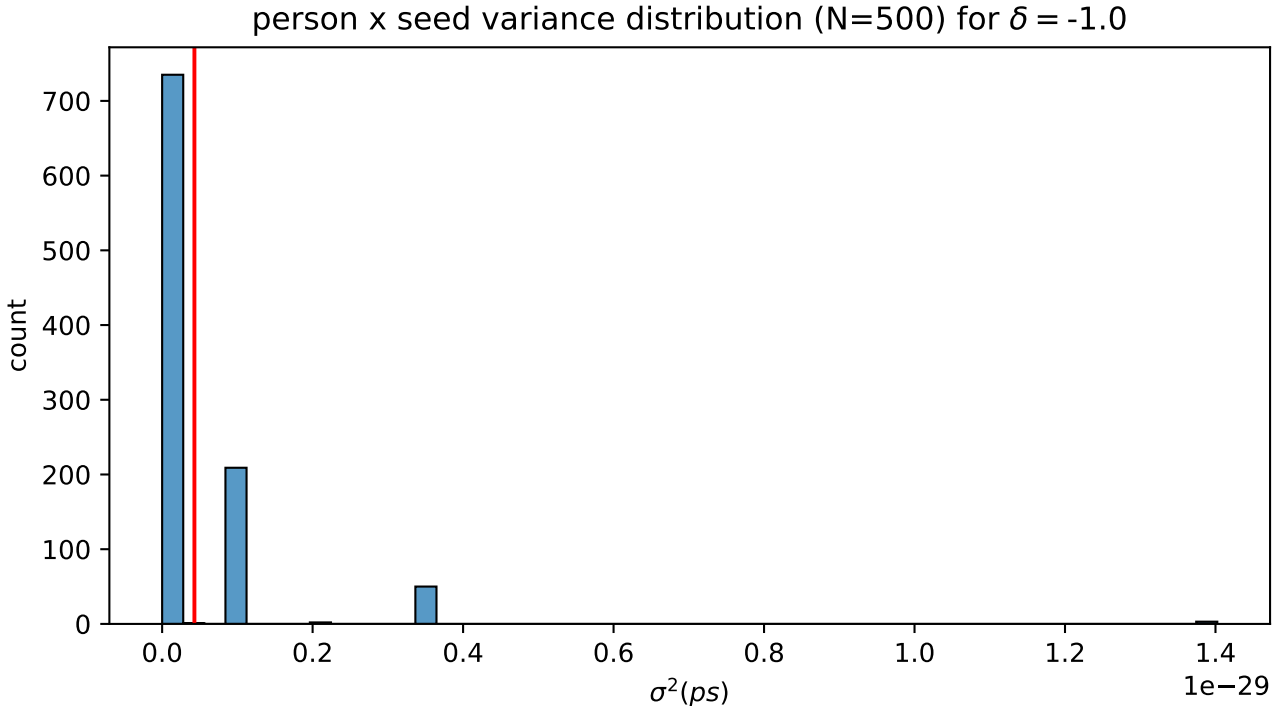

person x seed variance distribution (N=250) for $\delta = -1.0$
count
0
100
200
300
400
500
600
700
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
$\sigma^2(ps)$
1e−29

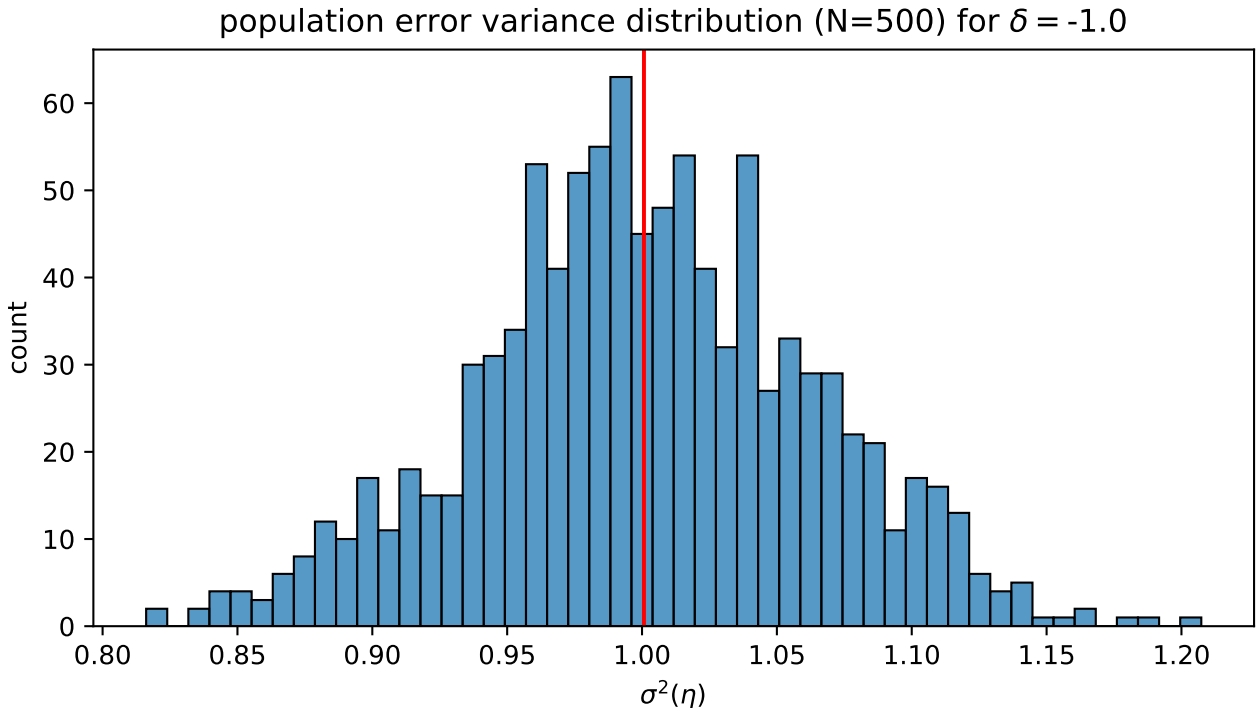

population error variance distribution (N=250) for $\delta = -1.0$
count
0
10
20
30
40
50
60
0.8
0.9
1.0
1.1
1.2
1.3
1.4
$\sigma^2(\eta)$

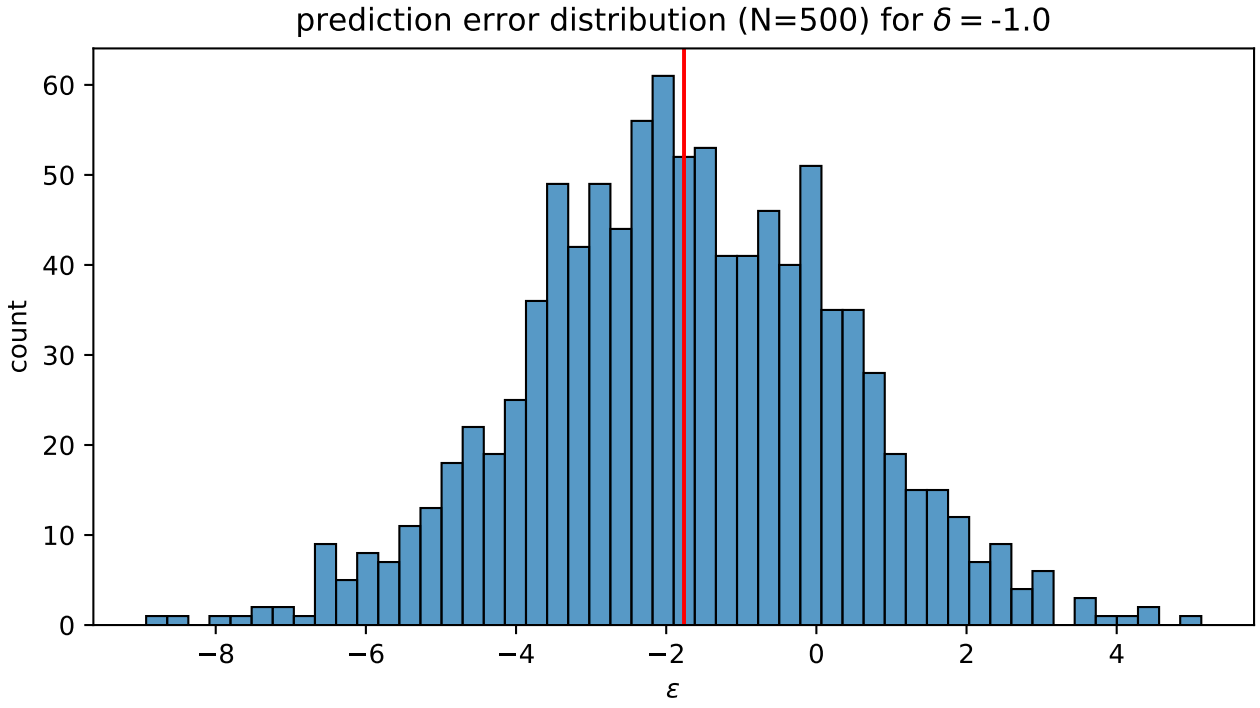

prediction error distribution (N=250) for $\delta = -1.0$
count
0
10
20
30
40
50
60
−12.5
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
ε

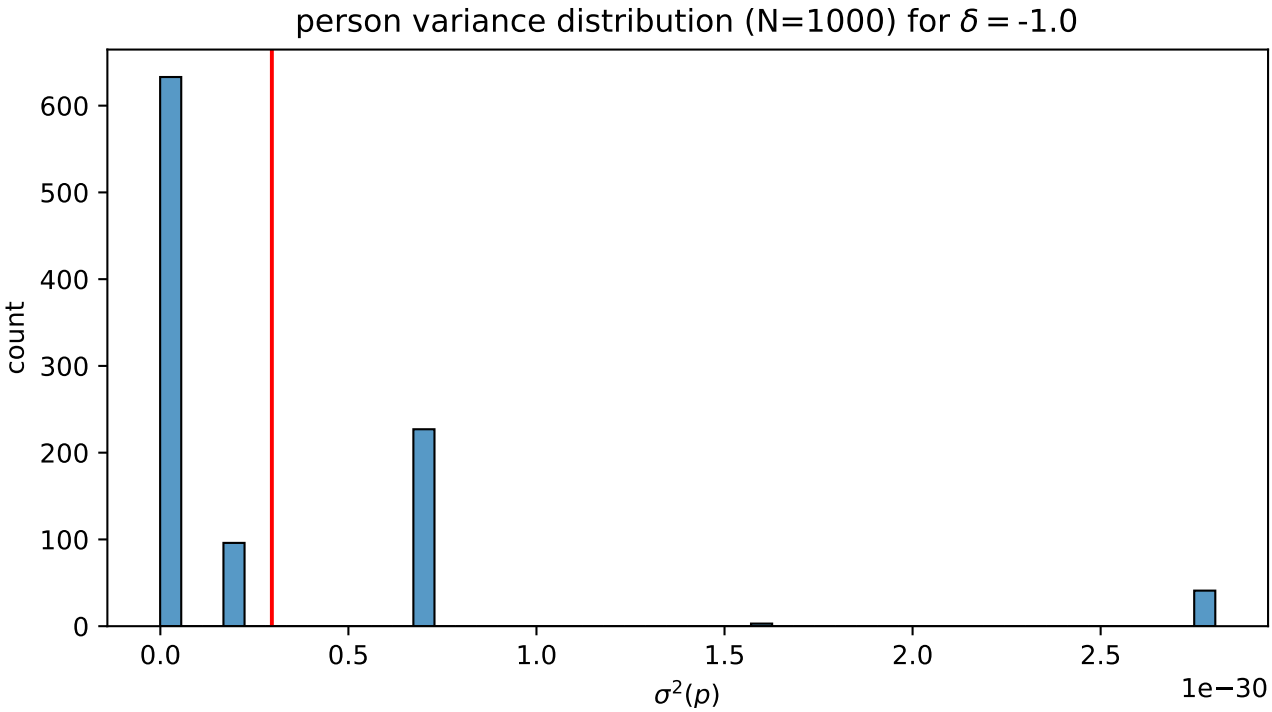

person variance distribution (N=500) for $\delta = -1.0$
count
$\sigma^2(p)$
1e−29

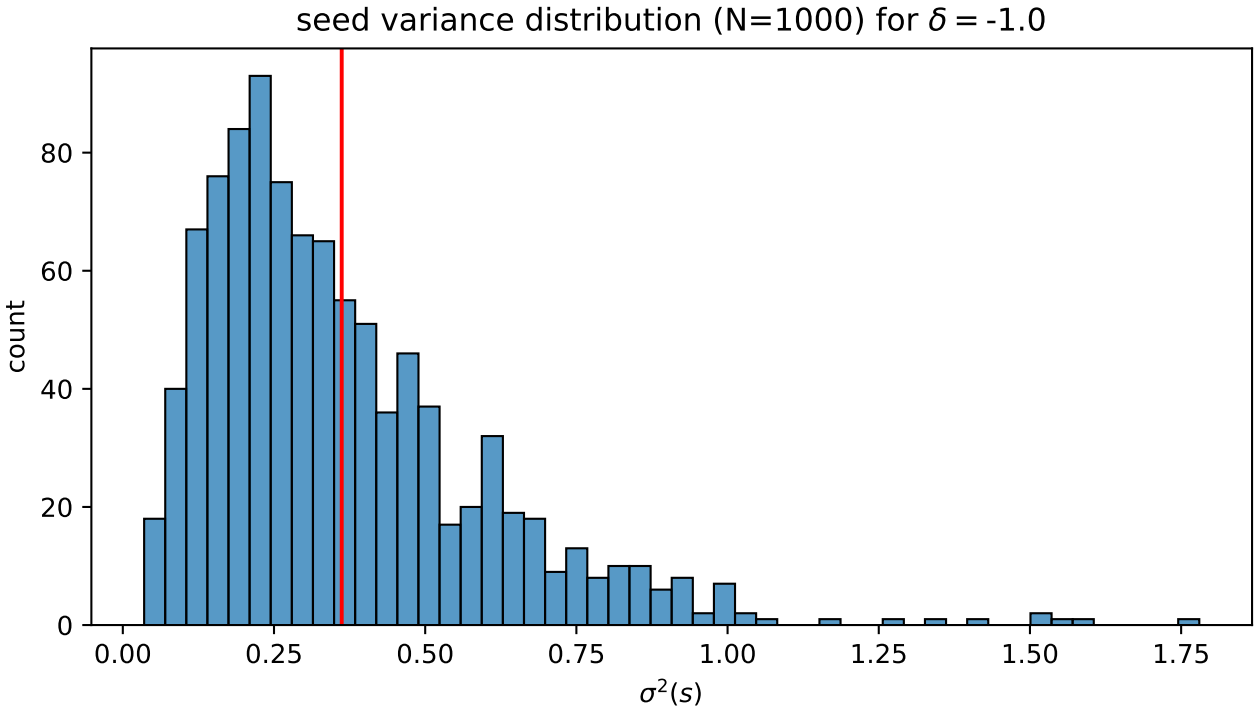

seed variance distribution (N=500) for $\delta = -1.0$
count
$\sigma^2(s)$

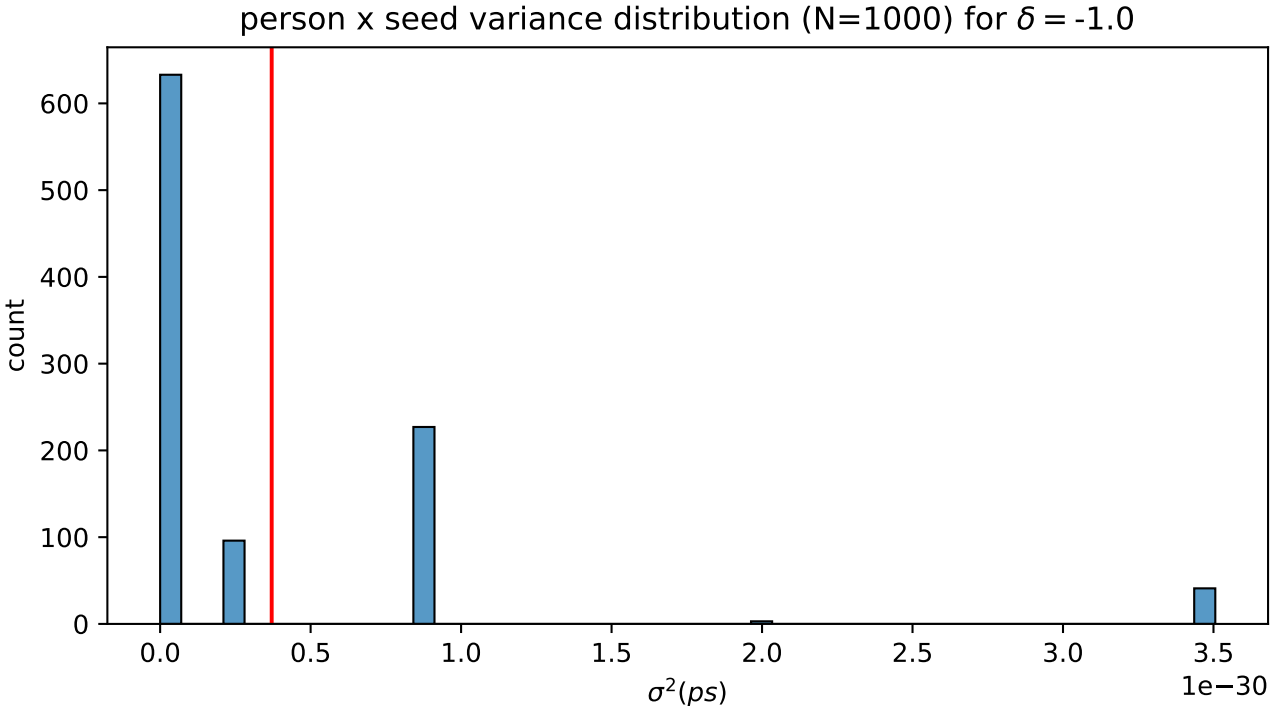

person x seed variance distribution (N=500) for $\delta = -1.0$
count
$\sigma^2(ps)$
1e−29

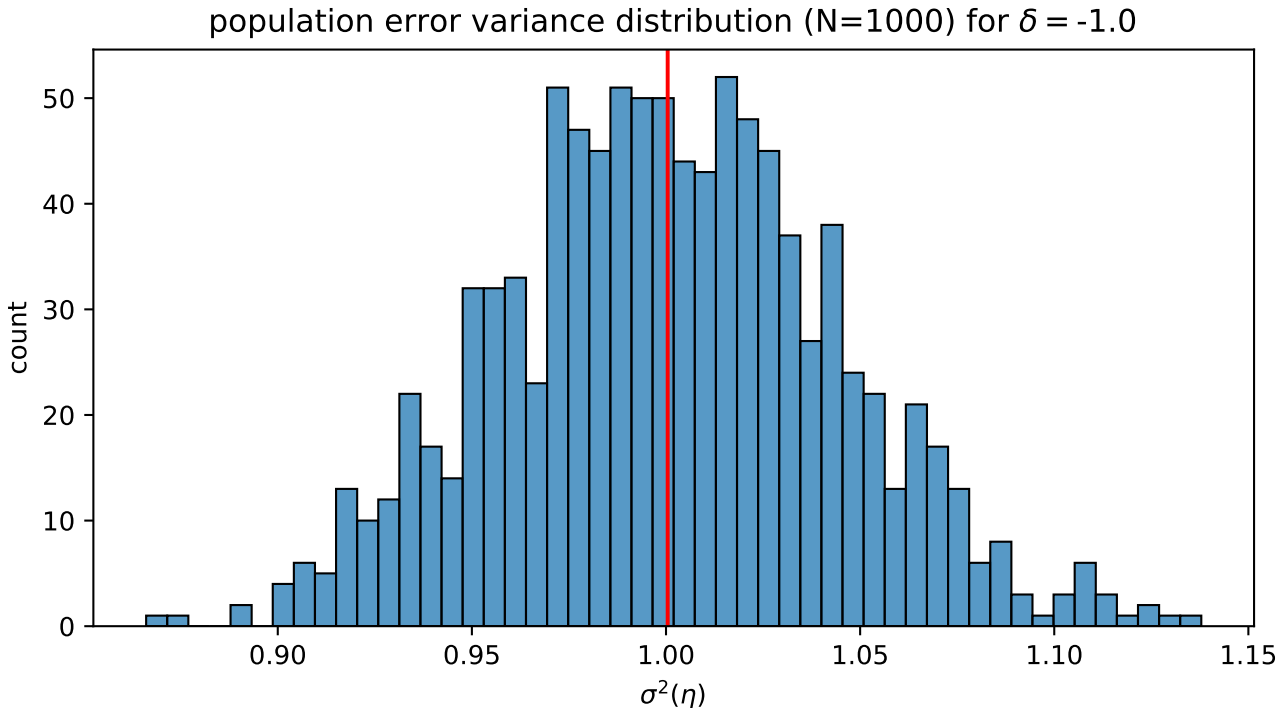

population error variance distribution (N=500) for $\delta = -1.0$
count
$\sigma^2(\eta)$

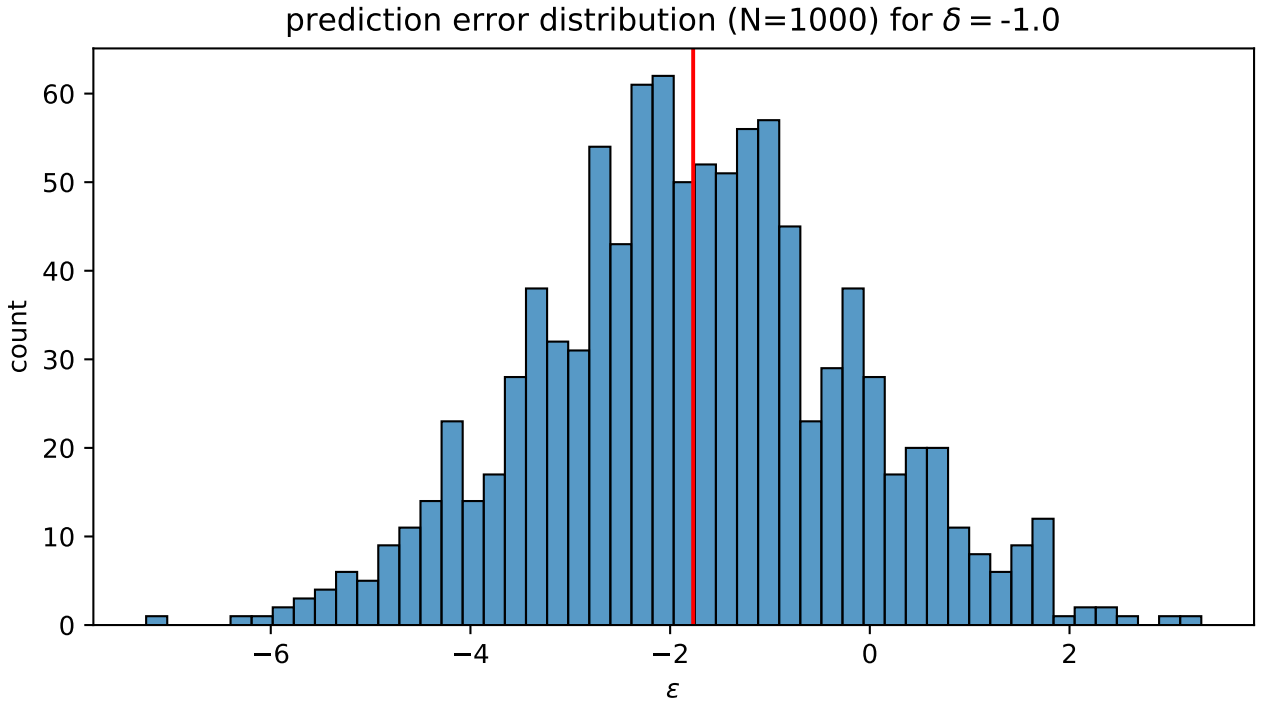

prediction error distribution (N=500) for $\delta = -1.0$
count
ε

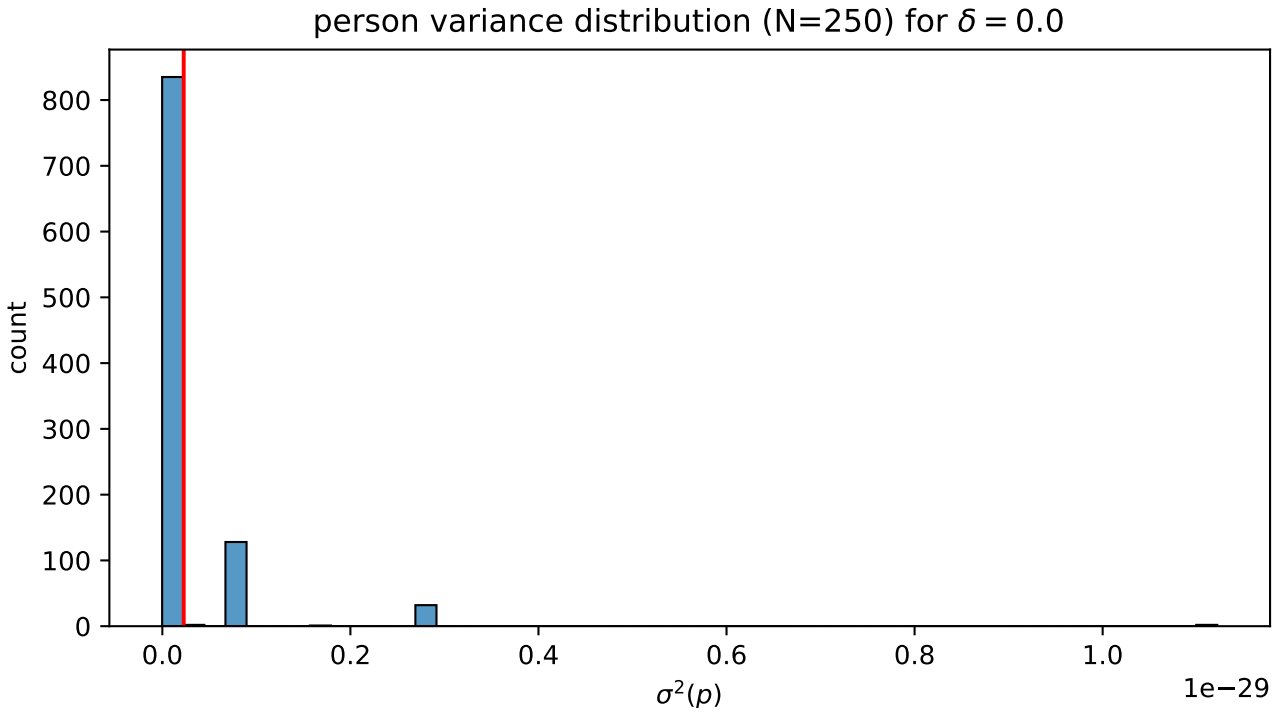

person variance distribution (N=1000) for $\delta = -1.0$
count
0
100
200
300
400
500
600
0.0
0.5
1.0
1.5
2.0
2.5
$\sigma^2(p)$
1e−30

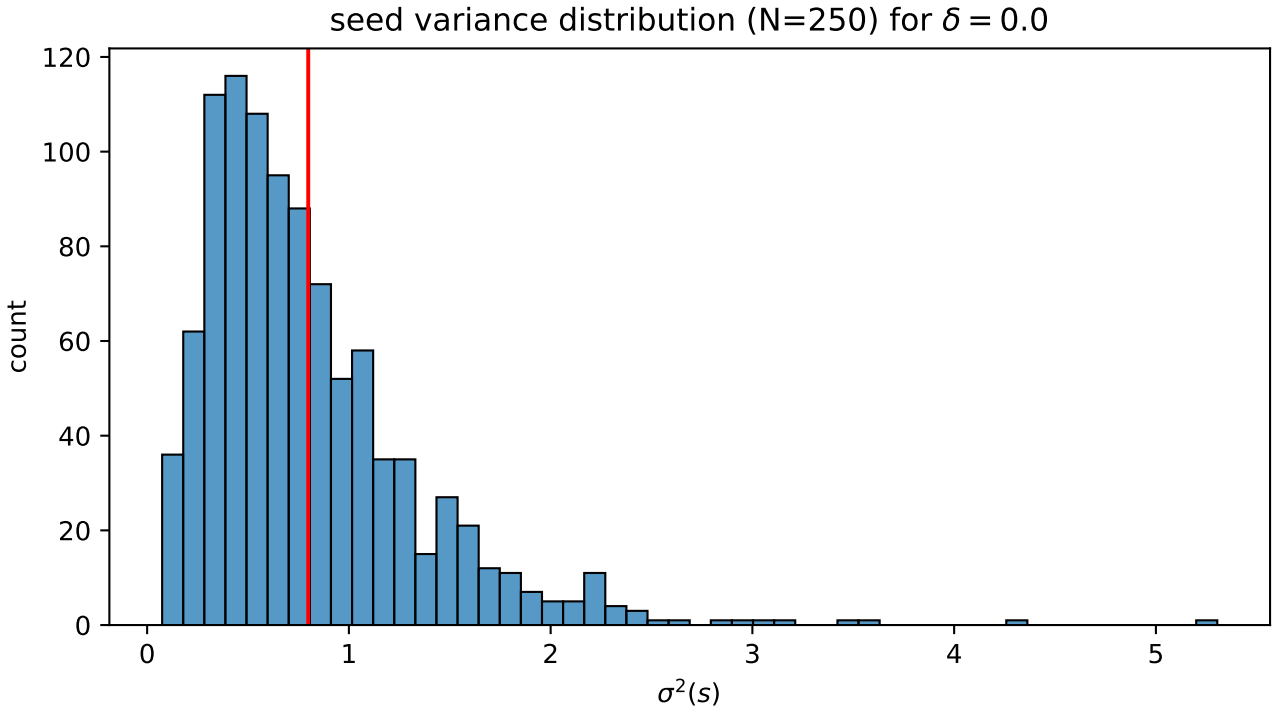

seed variance distribution (N=1000) for $\delta = -1.0$
count
0
20
40
60
80
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
$\sigma^2(s)$

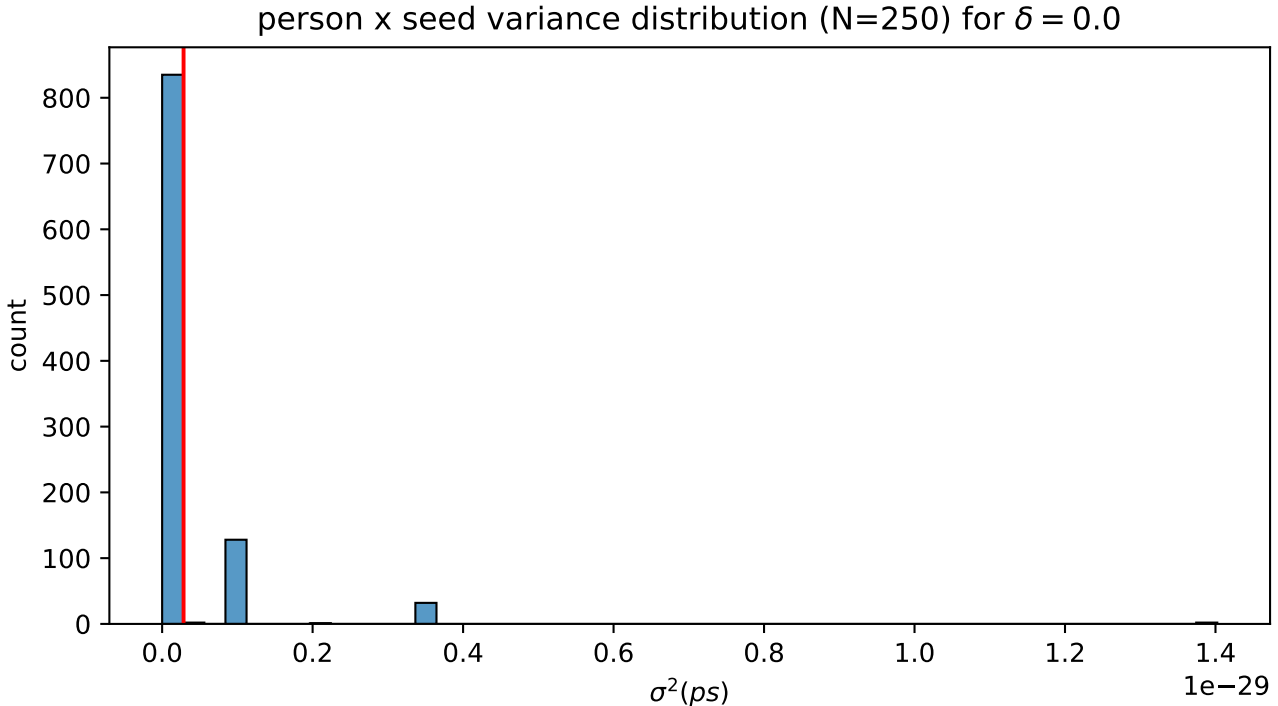

person x seed variance distribution (N=1000) for $\delta = -1.0$
count
0
100
200
300
400
500
600
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
$\sigma^2(ps)$
1e−30

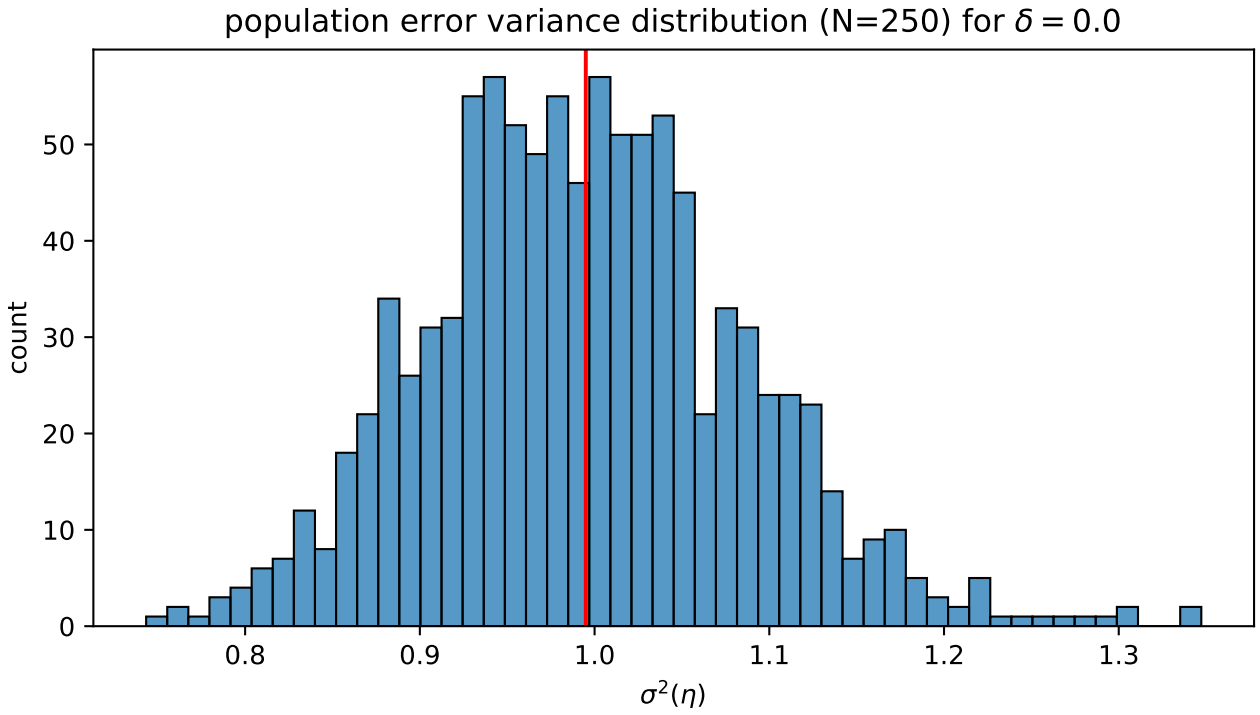

population error variance distribution (N=1000) for $\delta = -1.0$
count
0
10
20
30
40
50
0.90
0.95
1.00
1.05
1.10
1.15
$\sigma^2(\eta)$

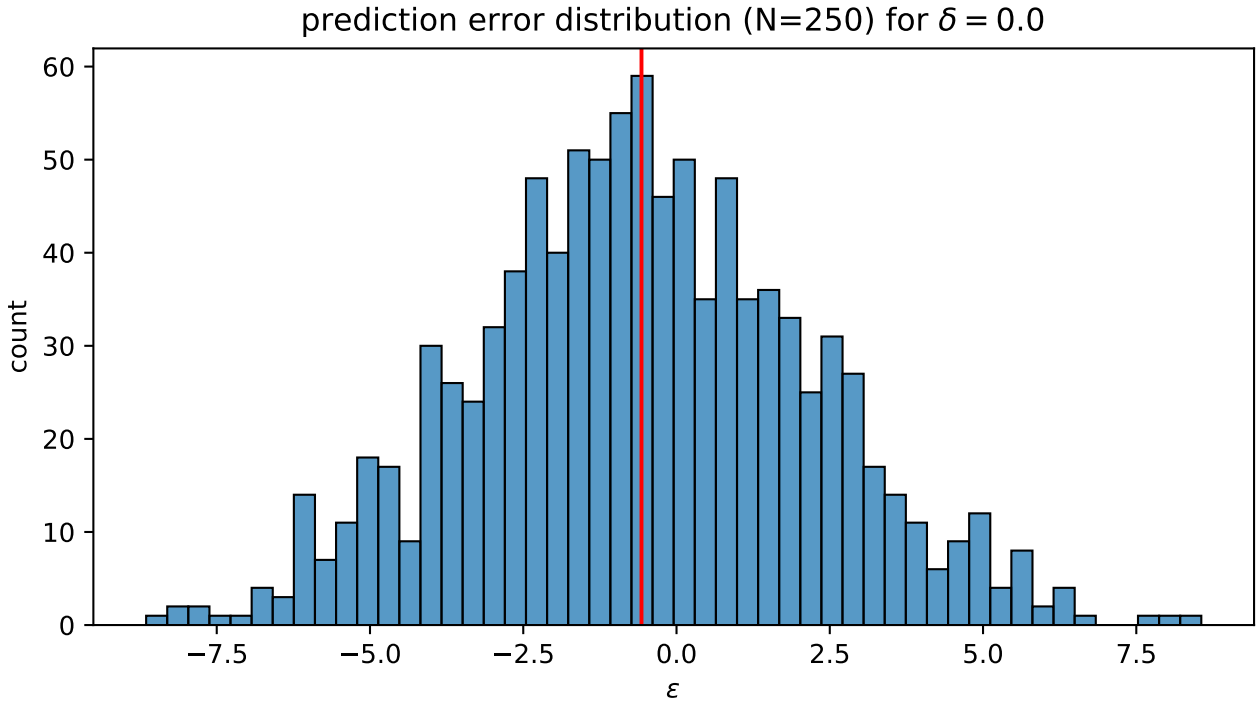

prediction error distribution (N=1000) for $\delta = -1.0$
count
0
10
20
30
40
50
60
−6
−4
−2
0
2
ε

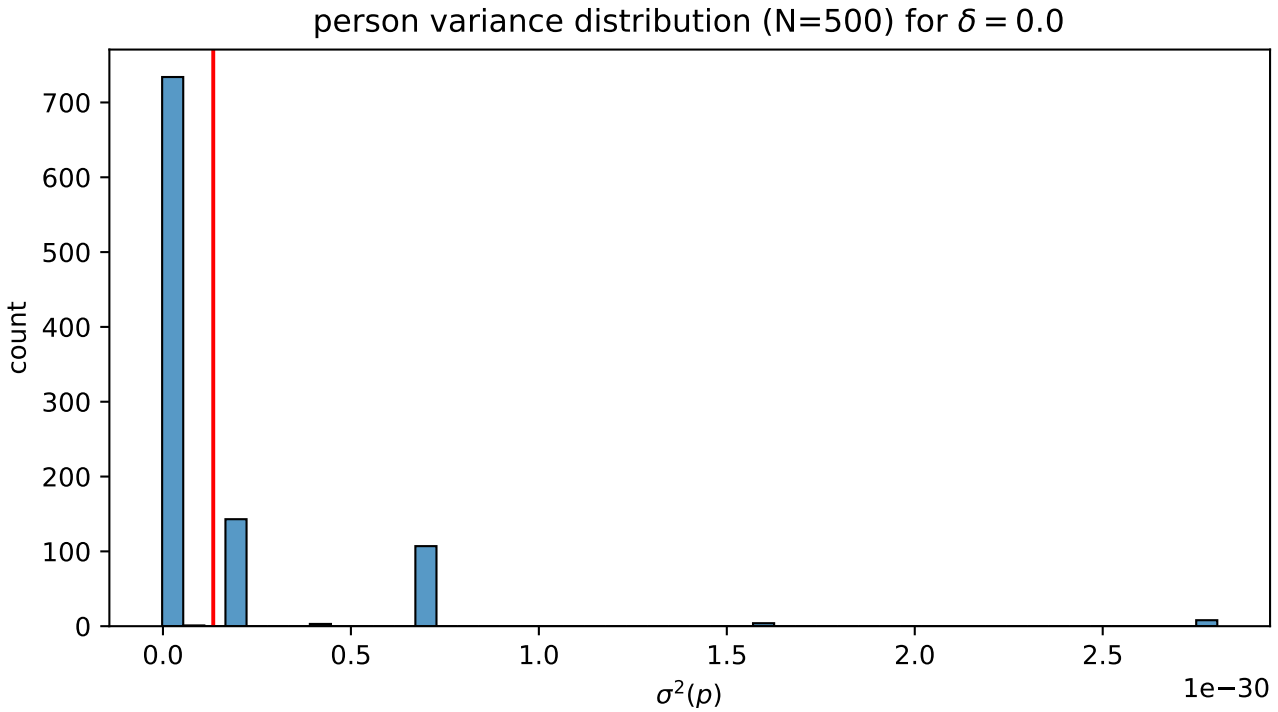

person variance distribution (N=250) for $\delta = 0.0$
count
$\sigma^2(p)$
1e−29

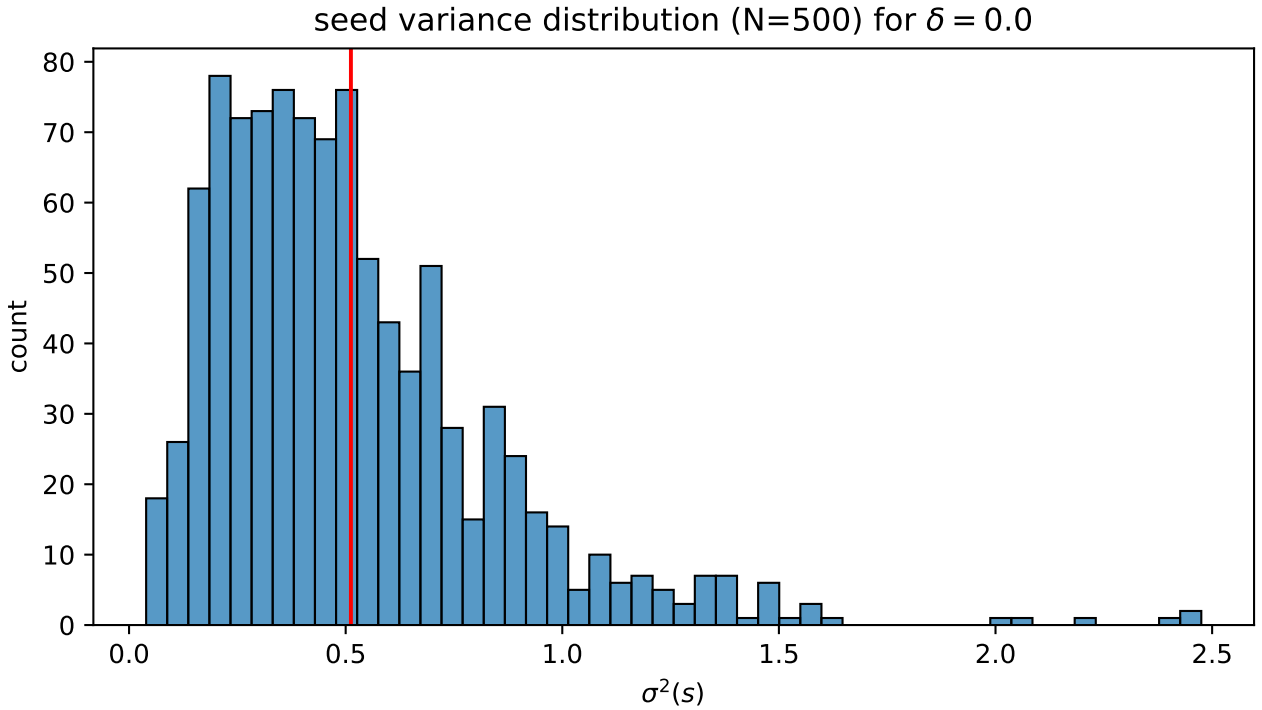

seed variance distribution (N=250) for $\delta = 0.0$
count
$\sigma^2(s)$

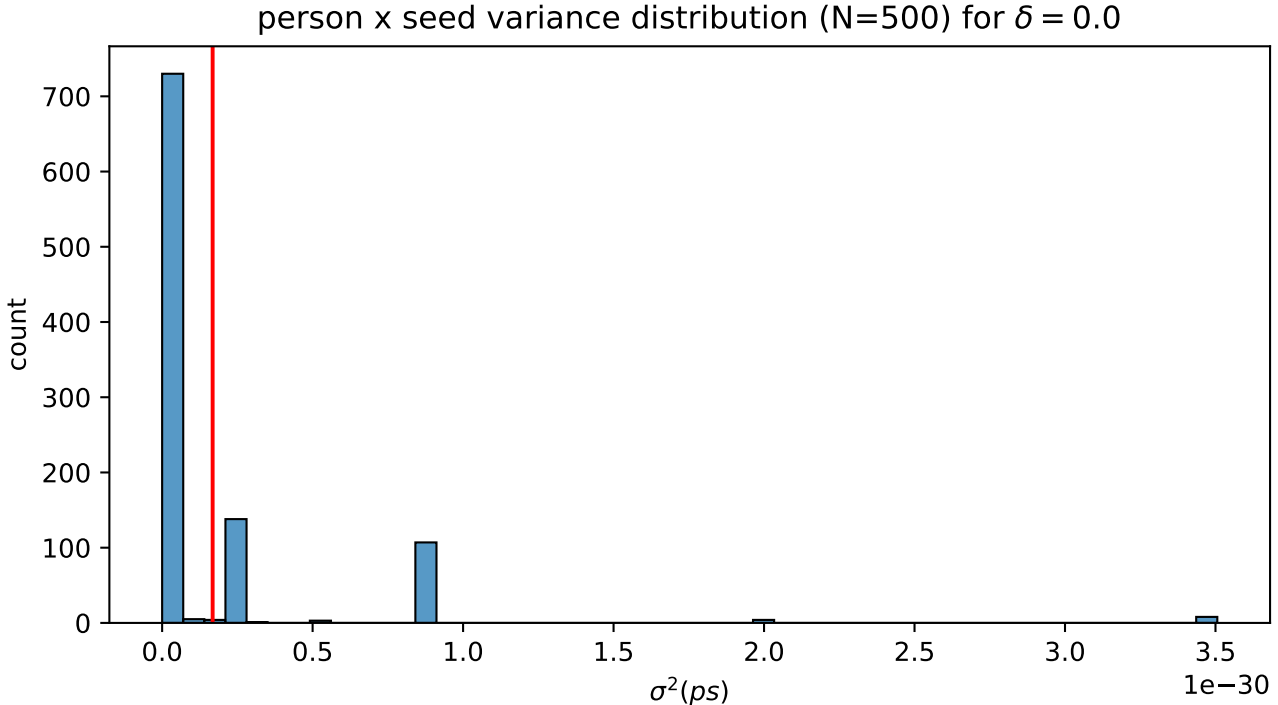

person x seed variance distribution (N=250) for $\delta = 0.0$
count
$\sigma^2(ps)$
1e−29

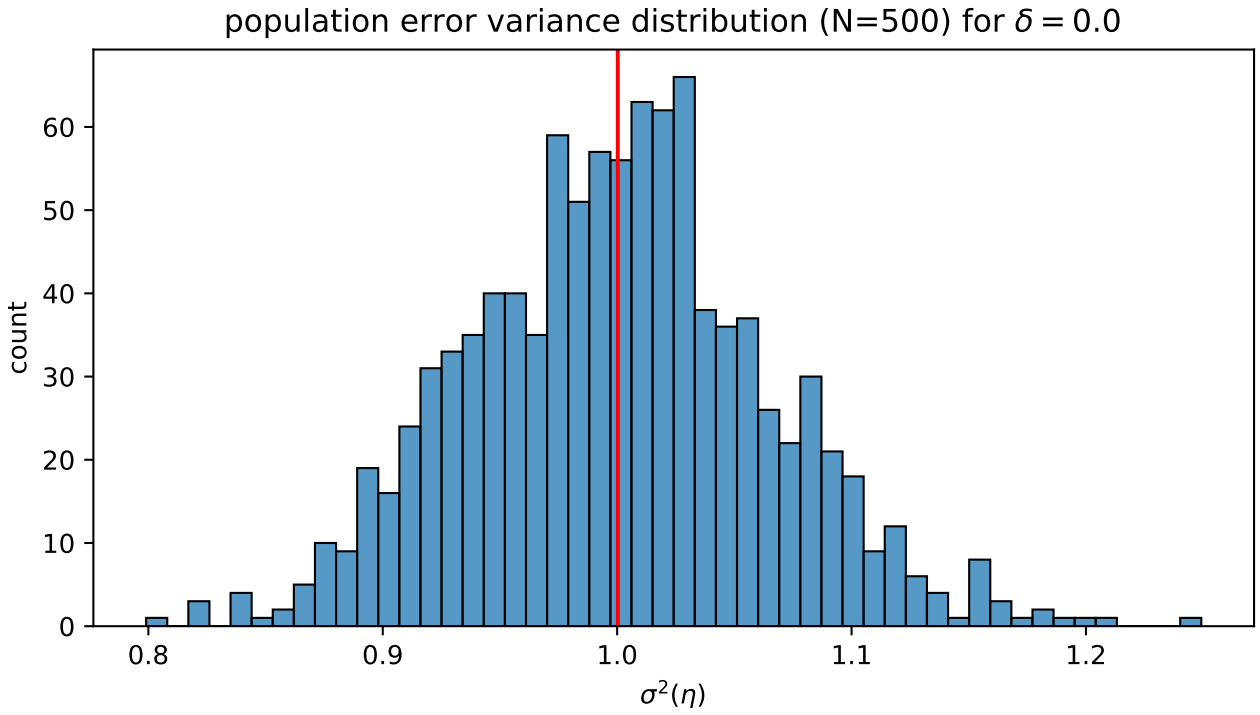

population error variance distribution (N=250) for $\delta = 0.0$
count
0
10
20
30
40
50
0.8
0.9
1.0
1.1
1.2
1.3
$\sigma^2(\eta)$

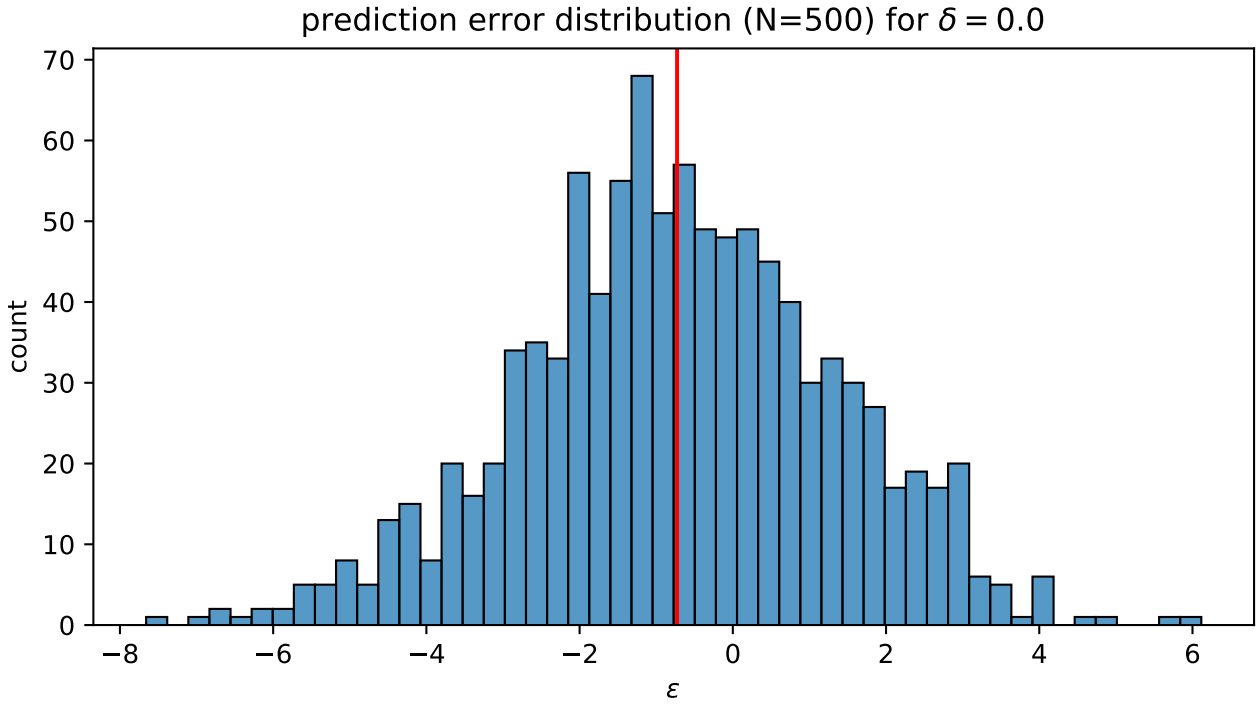

prediction error distribution (N=250) for $\delta = 0.0$
count
0
10
20
30
40
50
60
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
ε

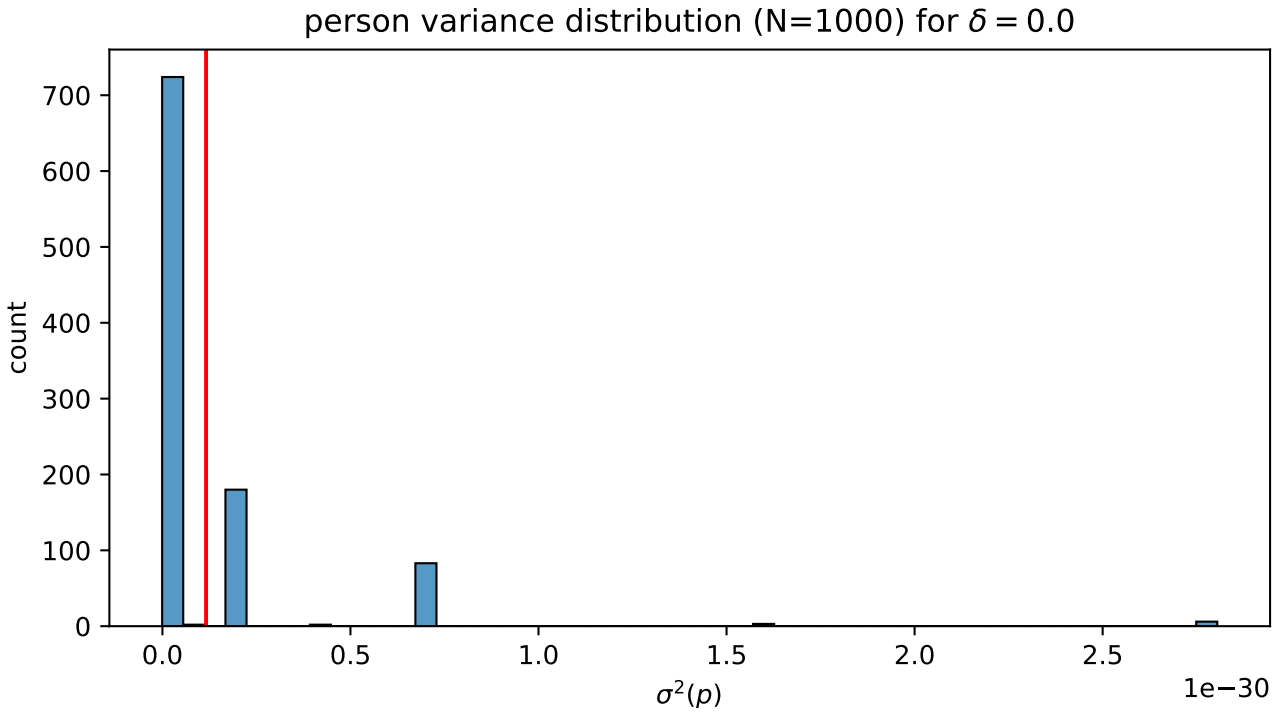

person variance distribution (N=500) for $\delta = 0.0$
count
$\sigma^2(p)$
1e−30

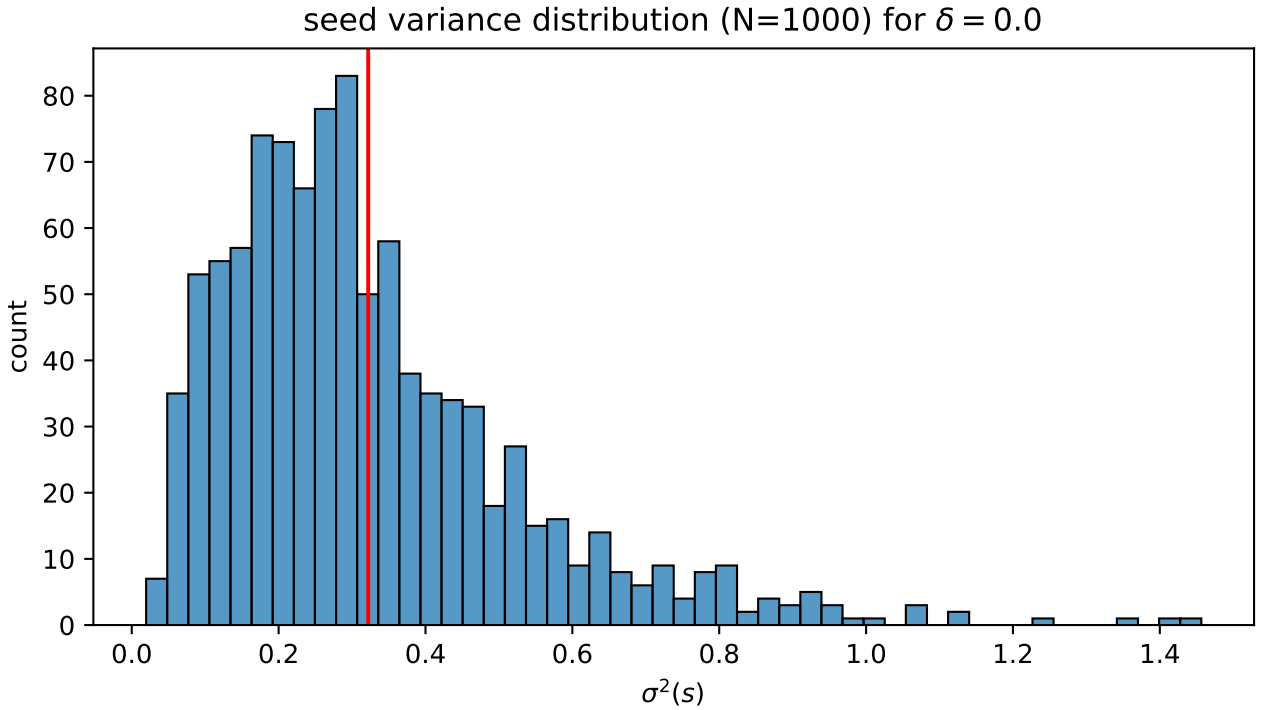

seed variance distribution (N=500) for $\delta = 0.0$
count
$\sigma^2(s)$

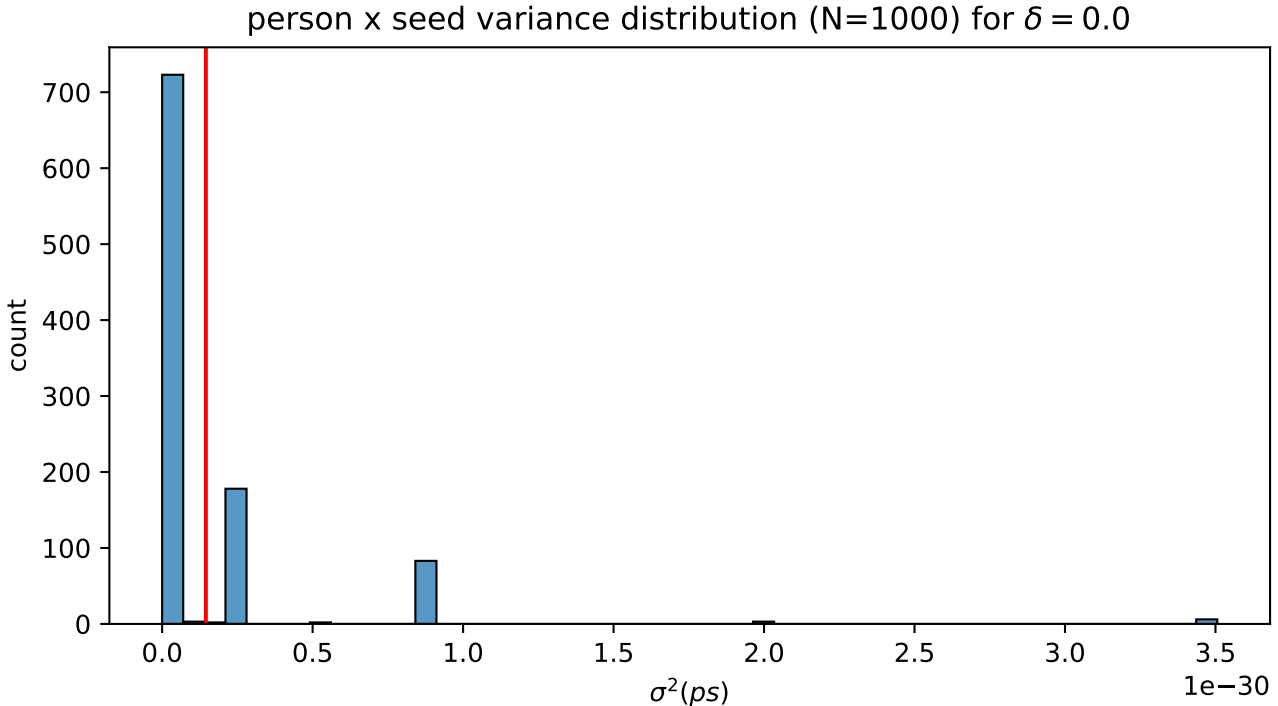

person x seed variance distribution (N=500) for $\delta = 0.0$
count
$\sigma^2(ps)$
1e−30

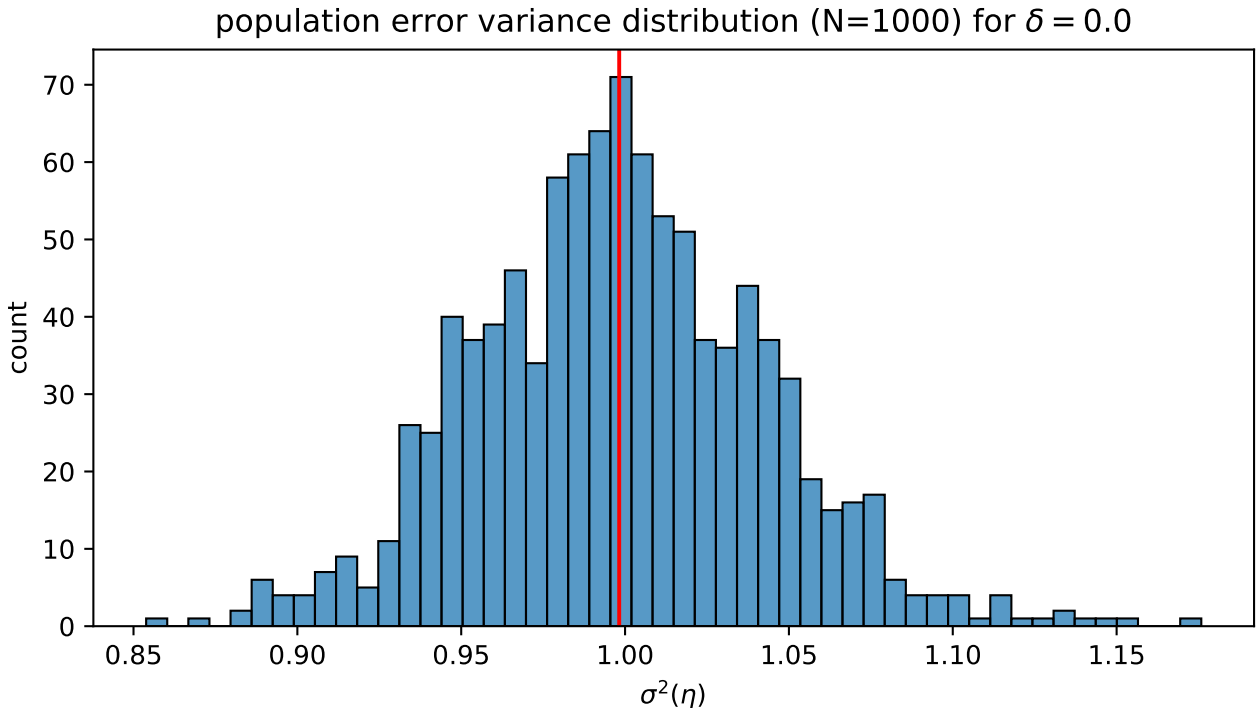

population error variance distribution (N=500) for $\delta = 0.0$
count
0
10
20
30
40
50
60
0.8
0.9
1.0
1.1
1.2
$\sigma^2(\eta)$

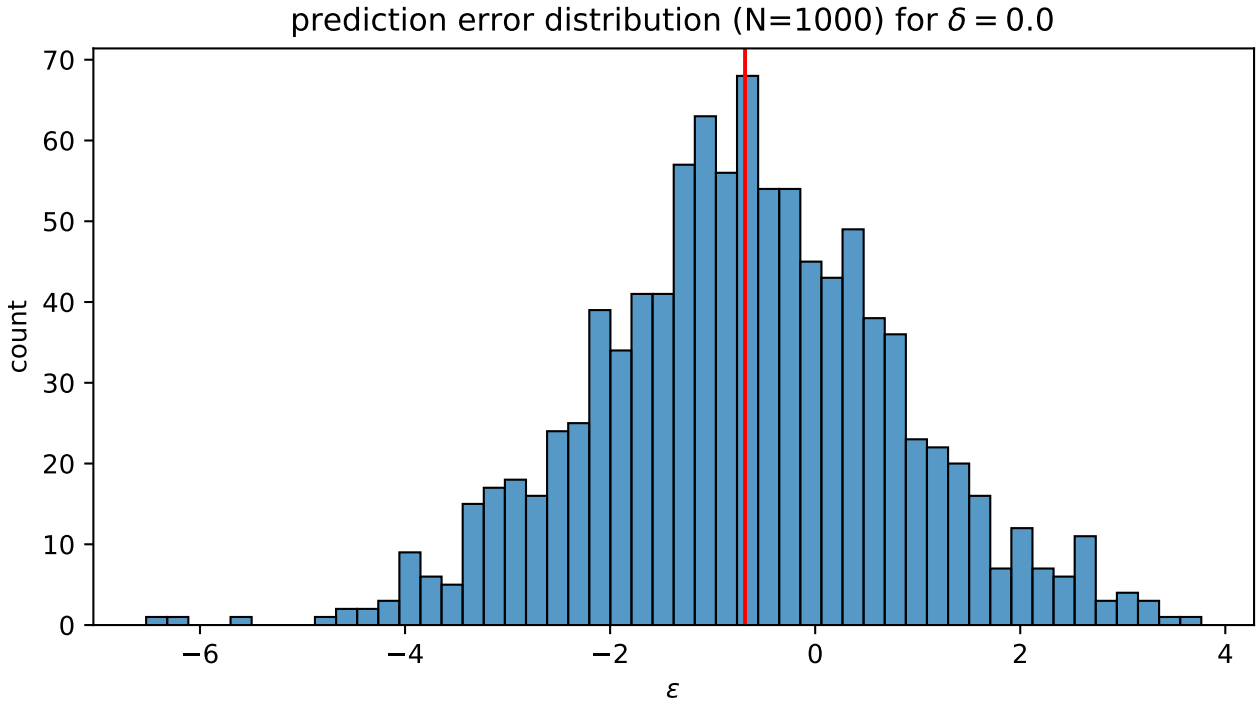

prediction error distribution (N=500) for $\delta = 0.0$
count
0
10
20
30
40
50
60
70
−8
−6
−4
−2
0
2
4
6
ε

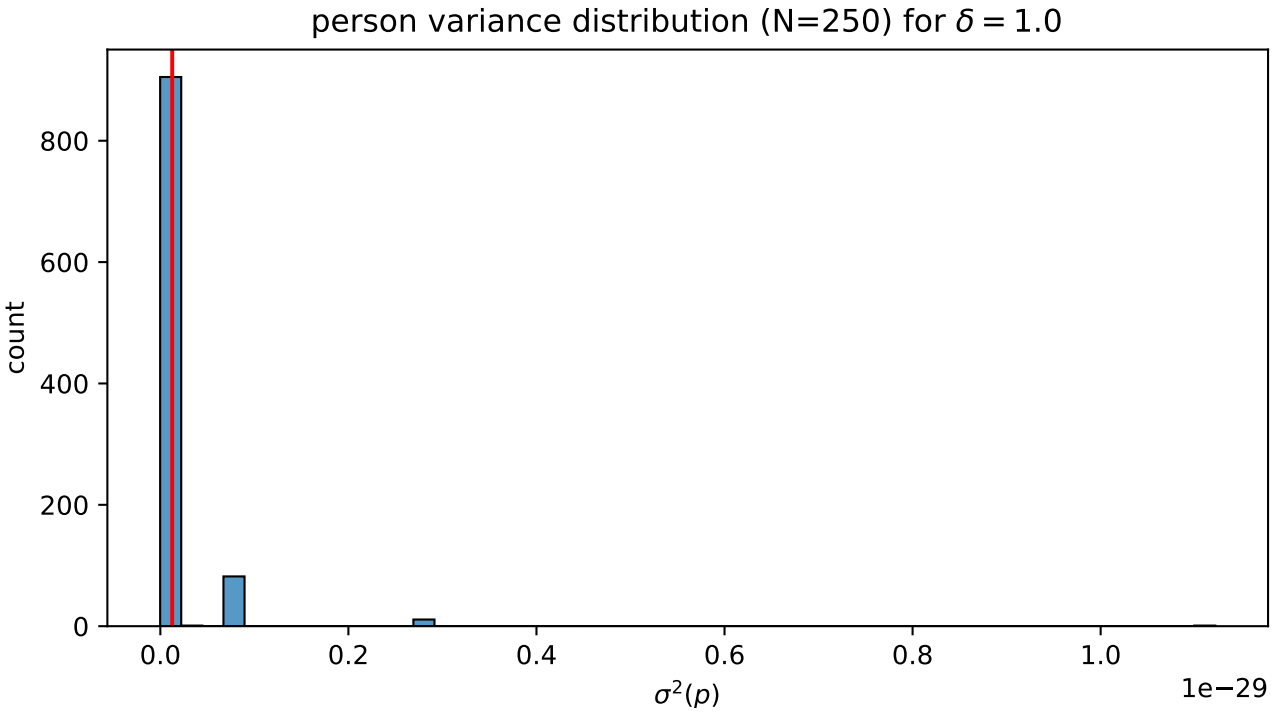

person variance distribution (N=1000) for $\delta = 0.0$
count
$\sigma^2(p)$
1e−30

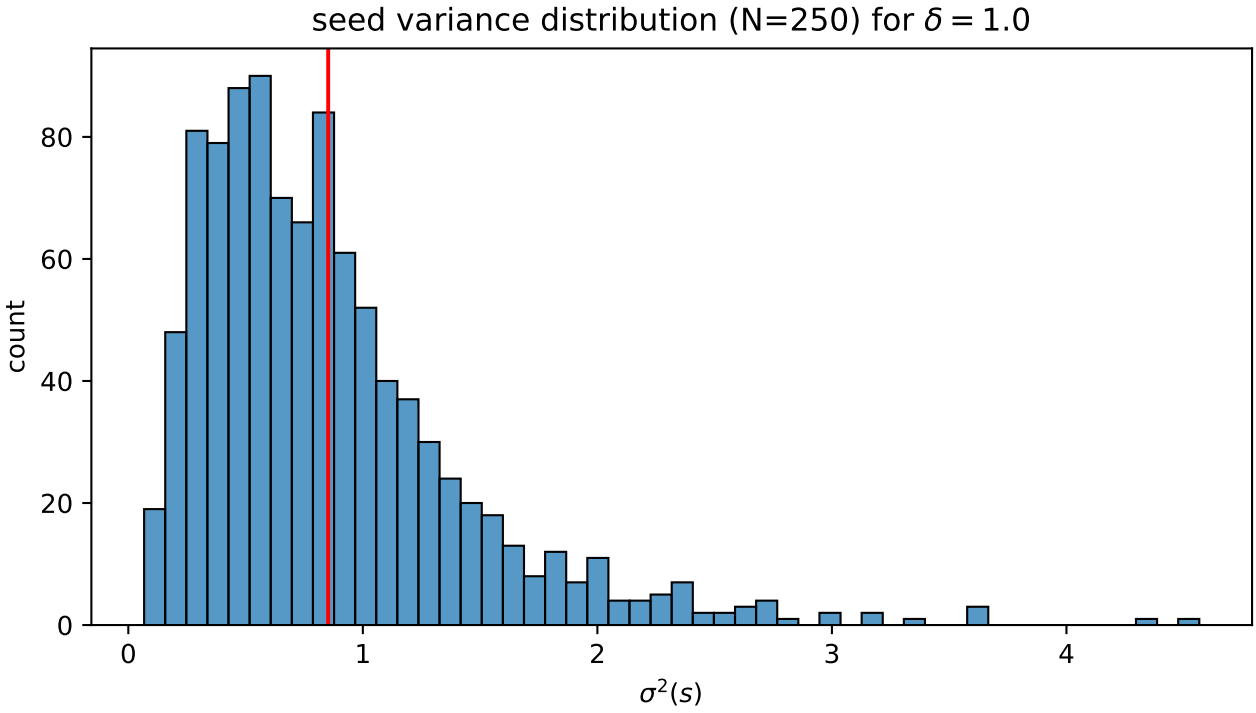

seed variance distribution (N=1000) for $\delta = 0.0$
count
$\sigma^2(s)$

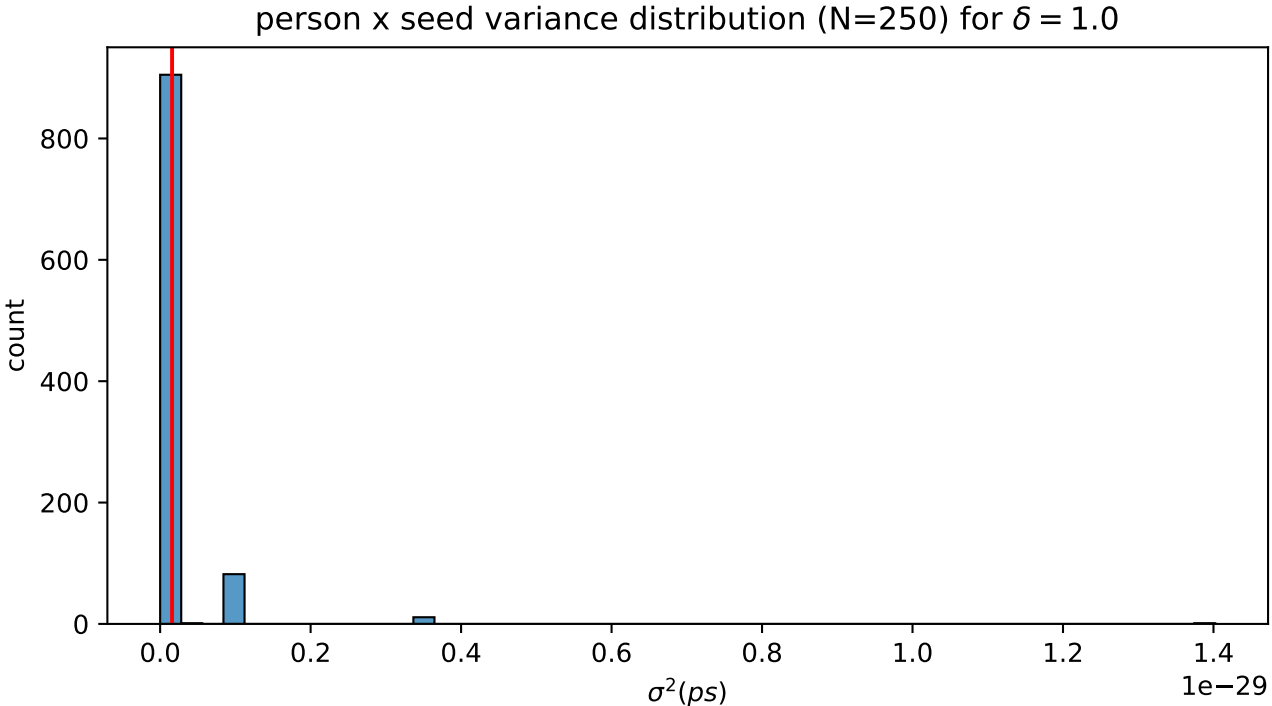

person x seed variance distribution (N=1000) for $\delta = 0.0$
count
$\sigma^2(ps)$
1e−30

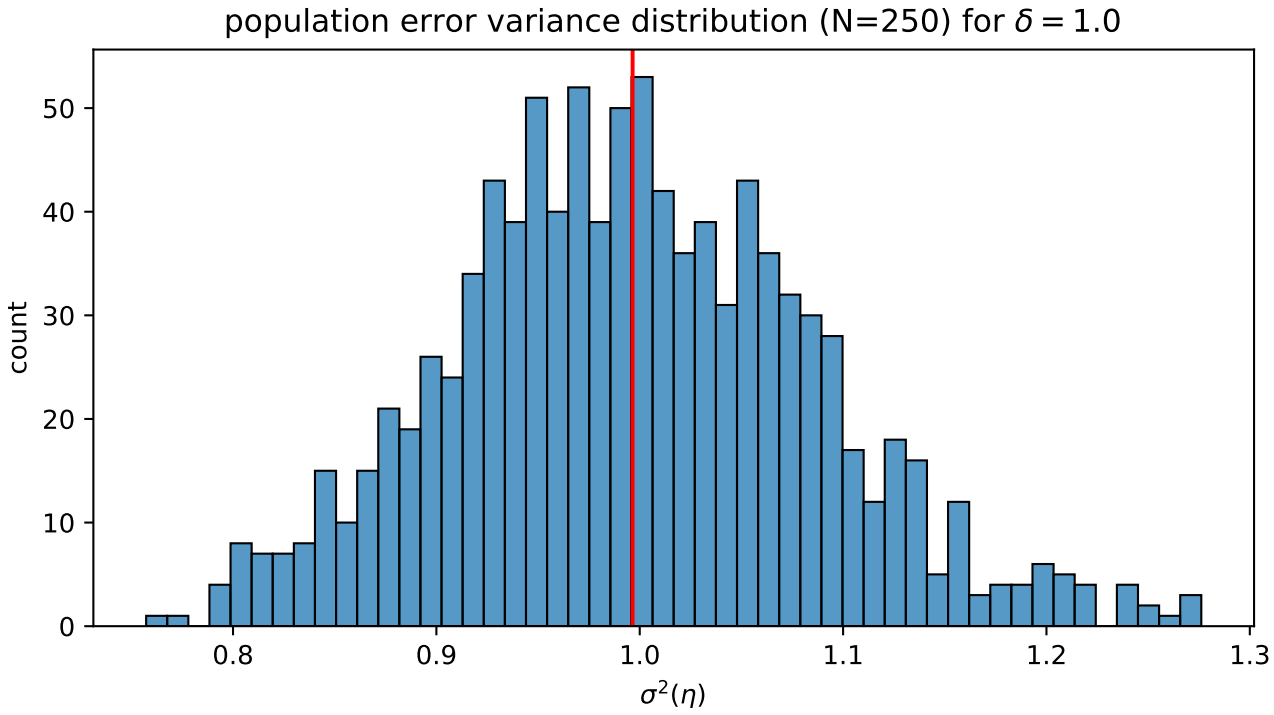

population error variance distribution (N=1000) for $\delta = 0.0$
count
$\sigma^2(\eta)$

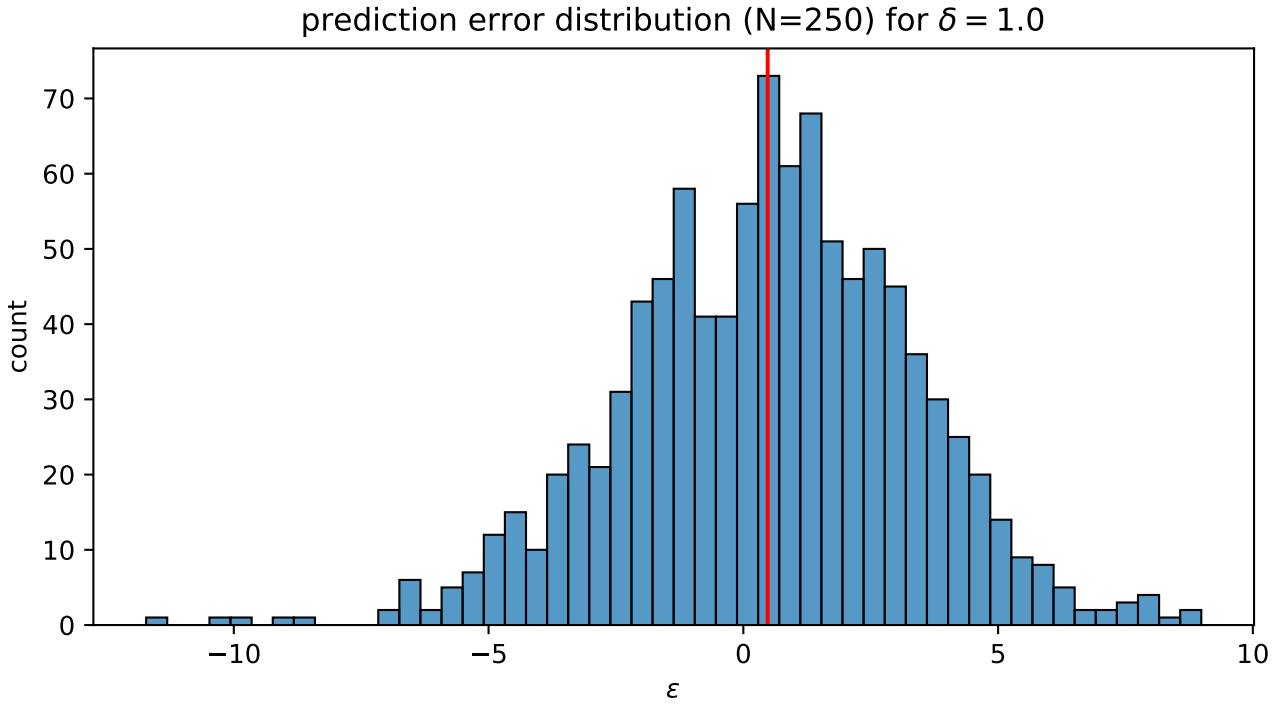

prediction error distribution (N=1000) for $\delta = 0.0$
count
ε

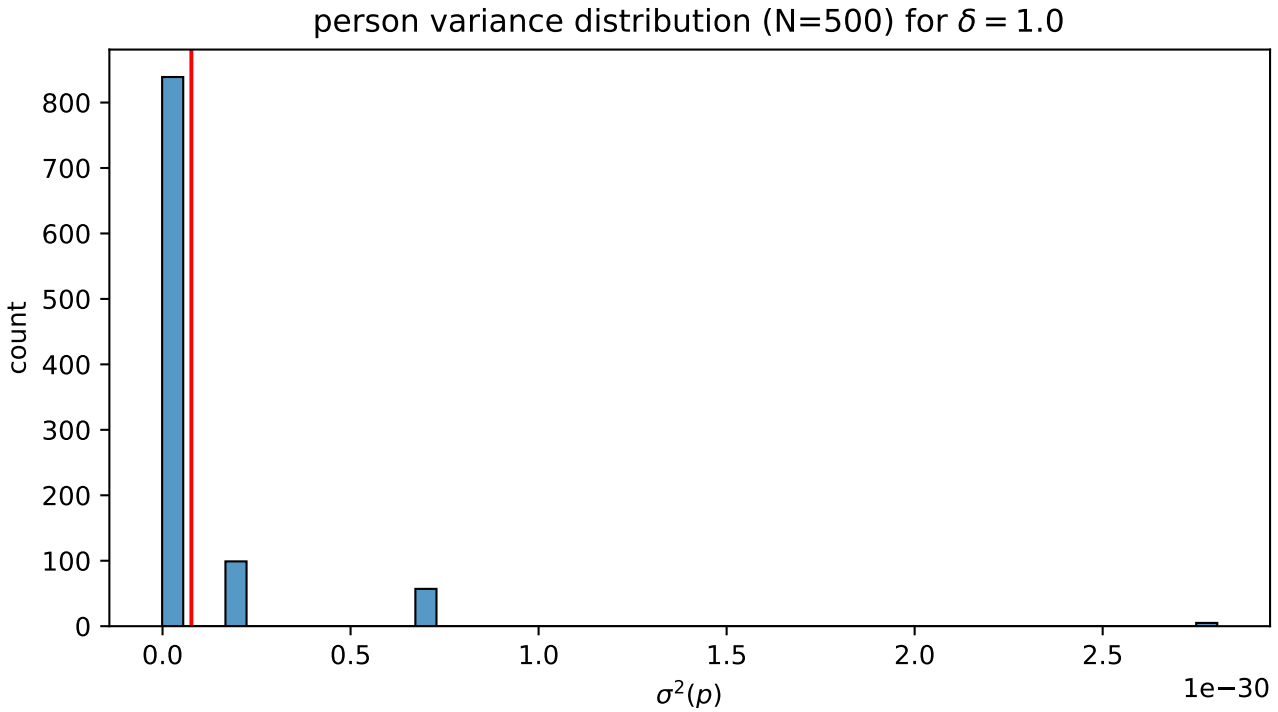

person variance distribution (N=250) for $\delta = 1.0$
count
0
200
400
600
800
0.0
0.2
0.4
0.6
0.8
1.0
$\sigma^2(p)$
1e−29

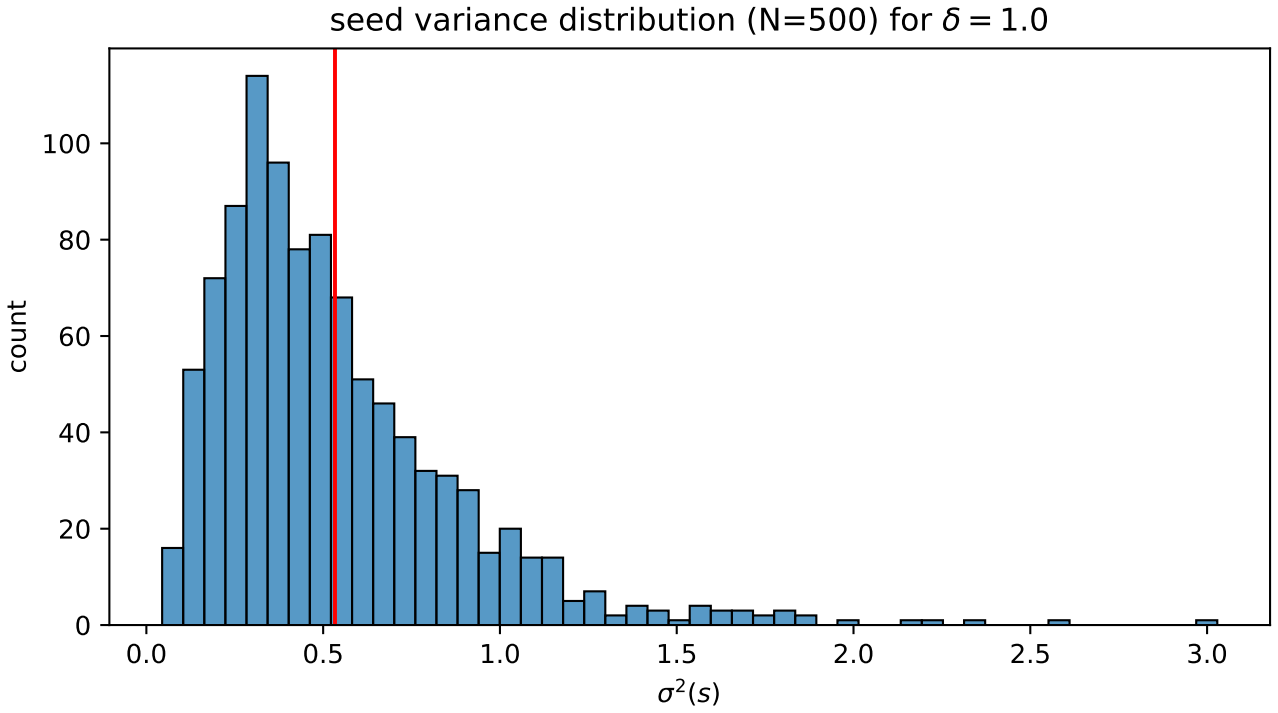

seed variance distribution (N=250) for $\delta = 1.0$
count
0
20
40
60
80
0
1
2
3
4
$\sigma^2(s)$

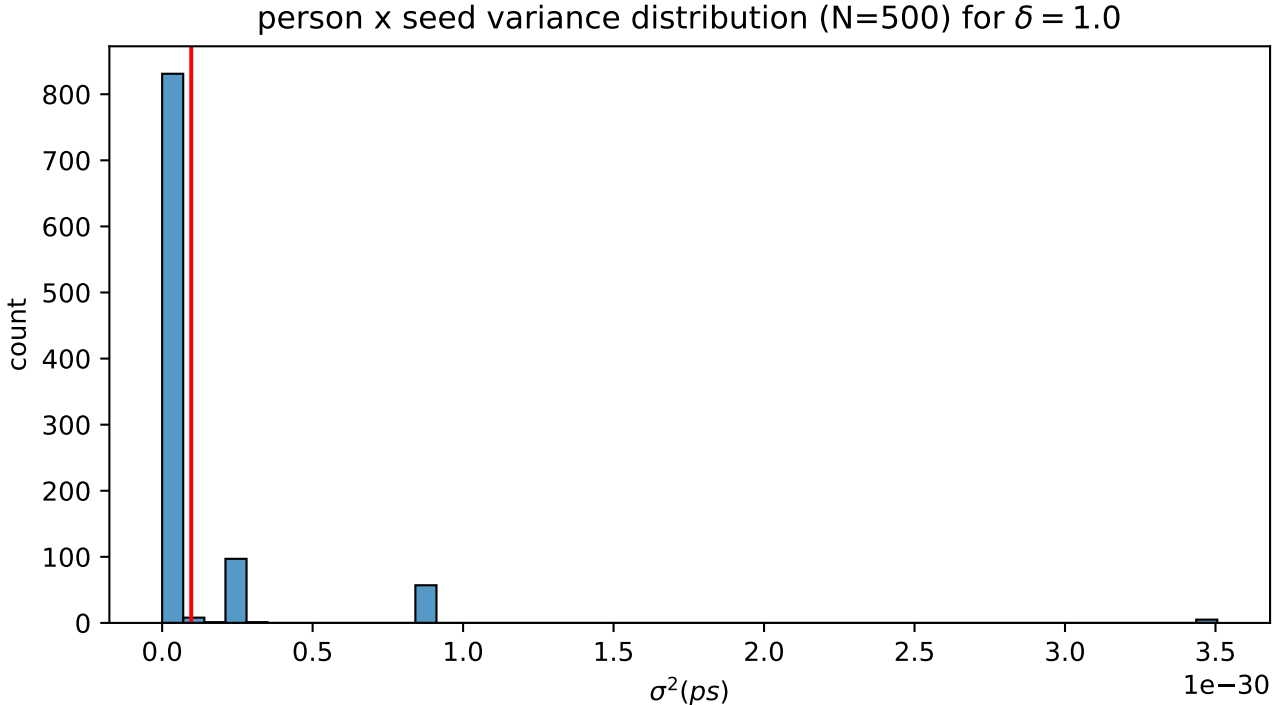

person x seed variance distribution (N=250) for $\delta = 1.0$
count
0
200
400
600
800
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
$\sigma^2(ps)$
1e−29

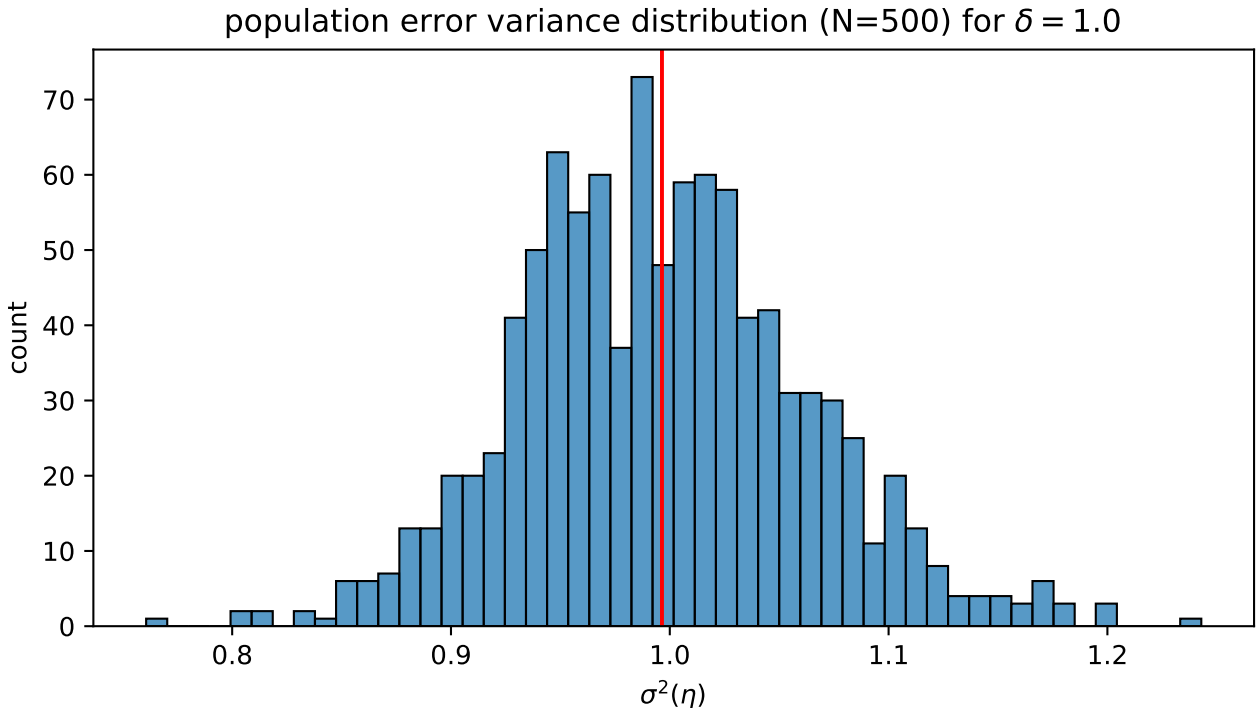

population error variance distribution (N=250) for $\delta = 1.0$
count
0
10
20
30
40
50
0.8
0.9
1.0
1.1
1.2
1.3
$\sigma^2(\eta)$

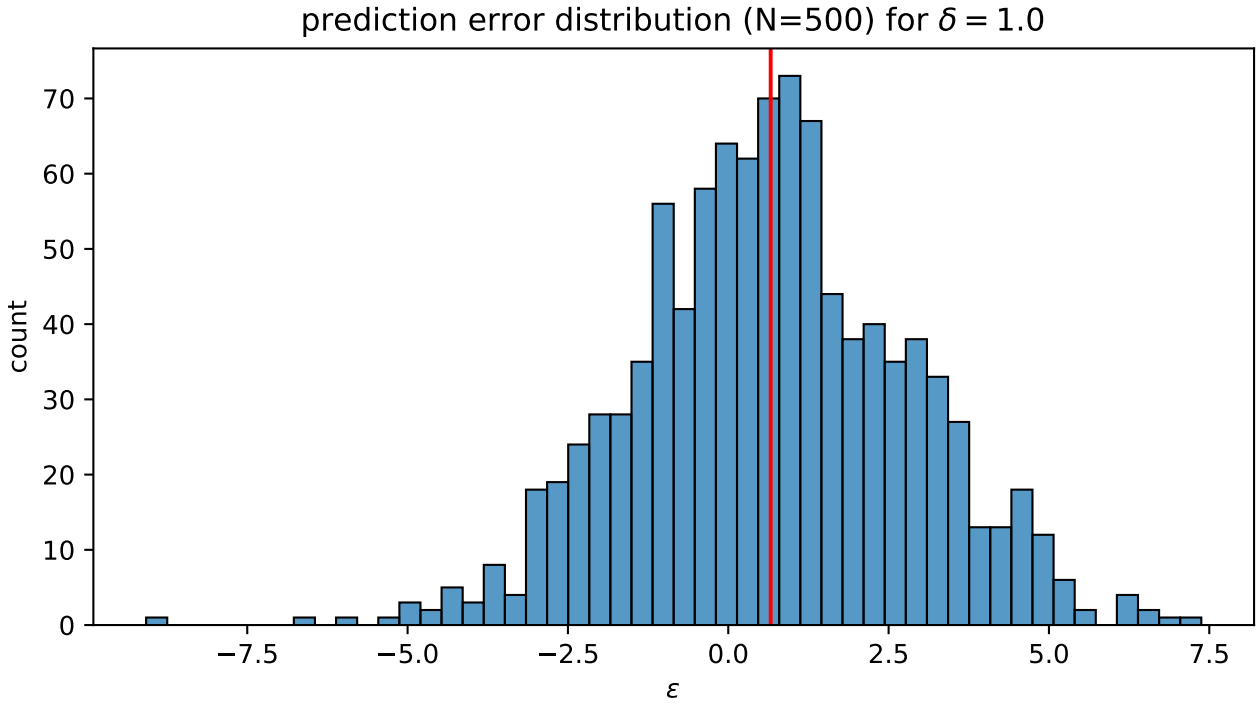

prediction error distribution (N=250) for $\delta = 1.0$
count
0
10
20
30
40
50
60
70
−10
−5
0
5
10
ε

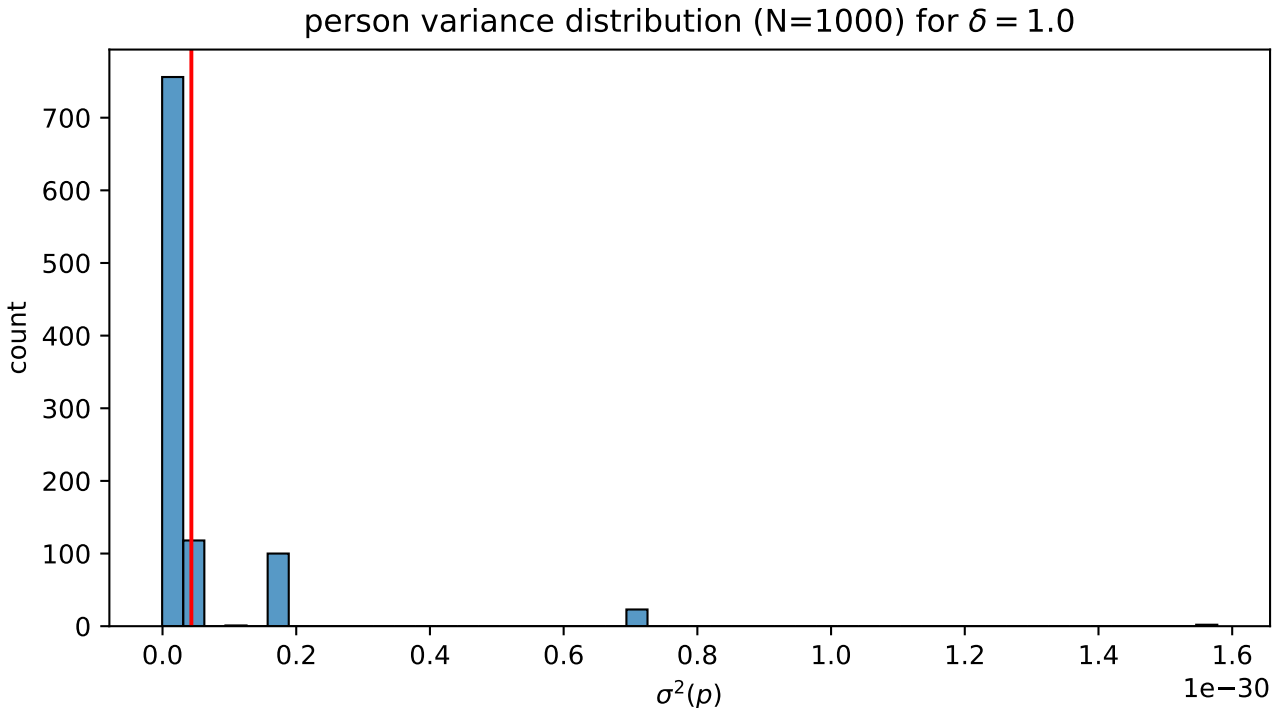

person variance distribution (N=500) for $\delta = 1.0$
count
$\sigma^2(p)$
1e−30

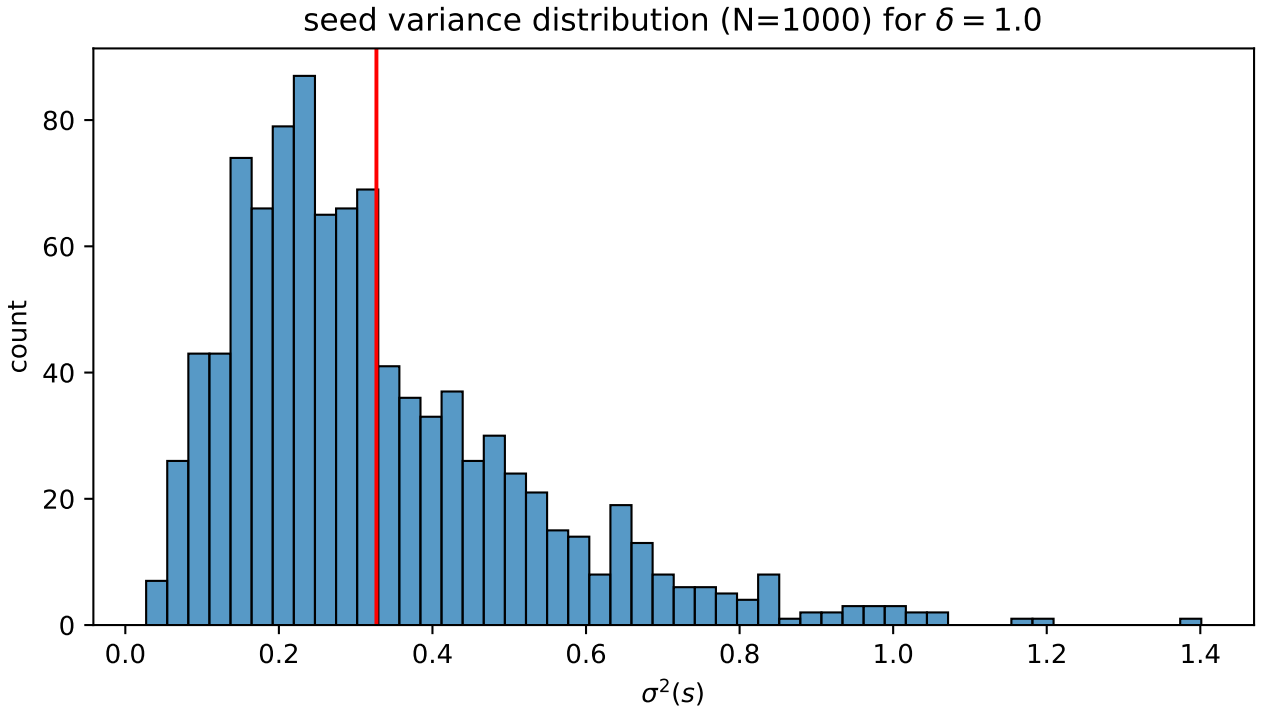

seed variance distribution (N=500) for $\delta = 1.0$
count
$\sigma^2(s)$

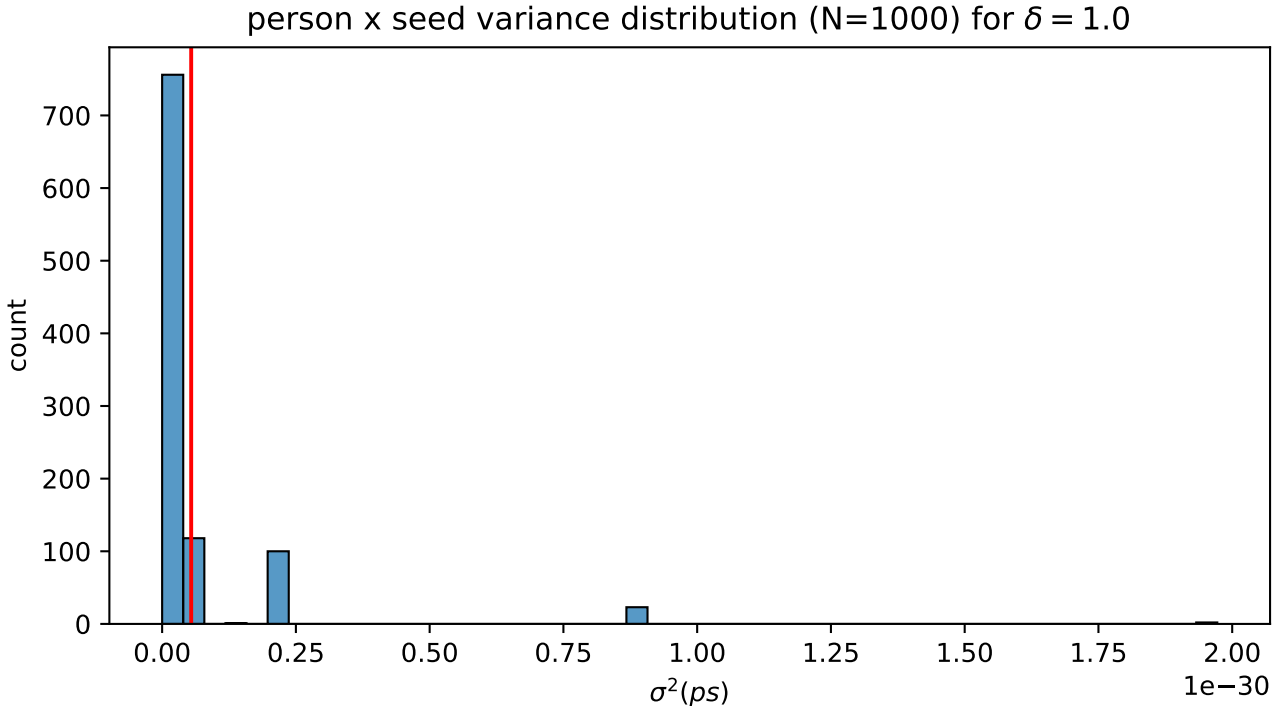

person x seed variance distribution (N=500) for $\delta = 1.0$
count
$\sigma^2(ps)$
1e−30

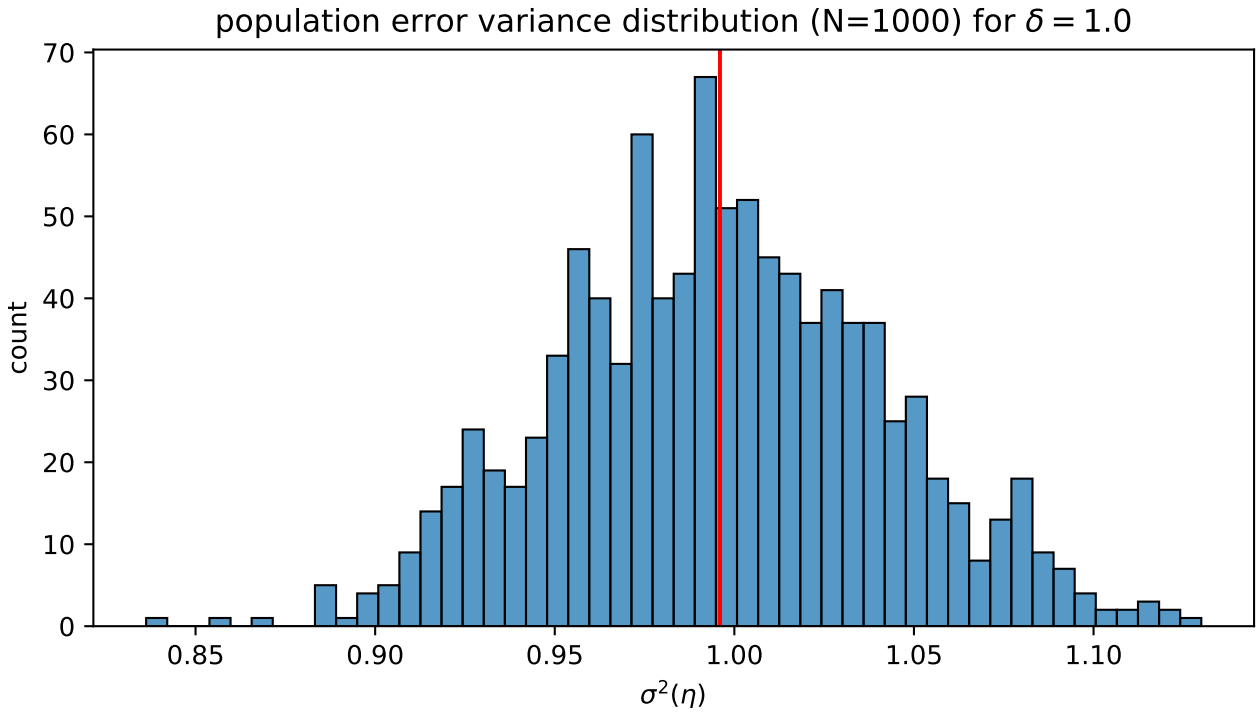

population error variance distribution (N=500) for $\delta = 1.0$
count
$\sigma^2(\eta)$

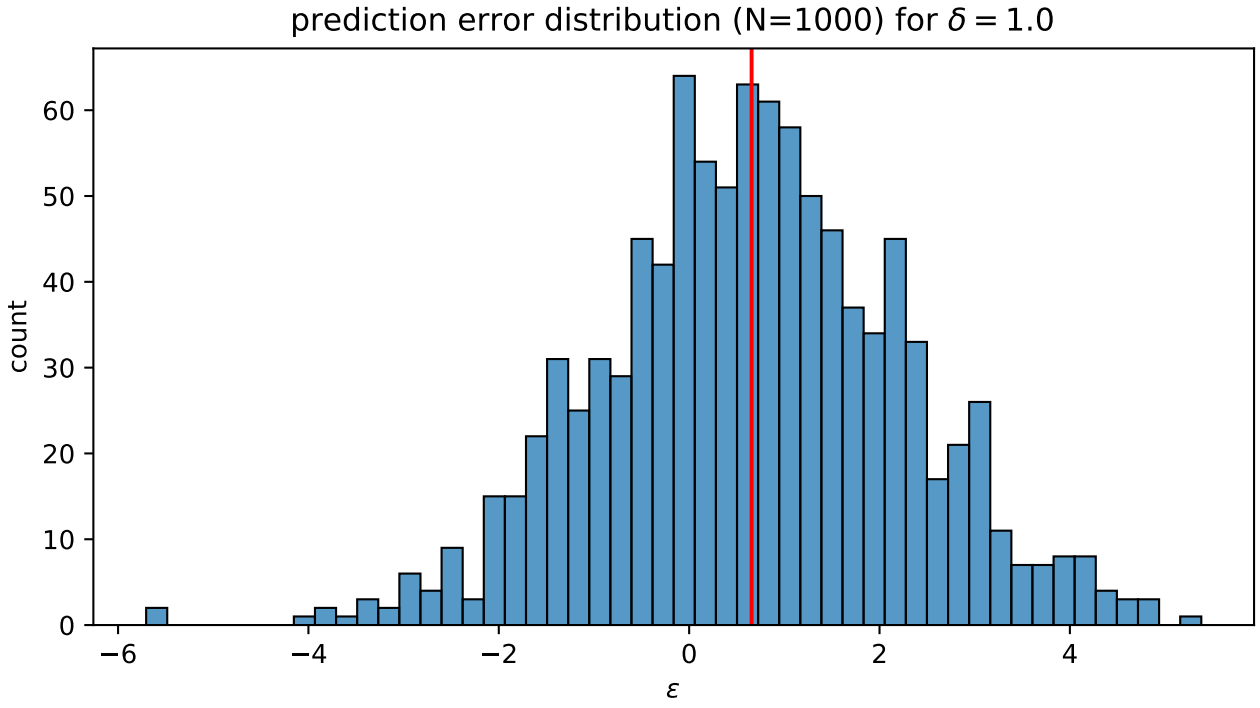

prediction error distribution (N=500) for $\delta = 1.0$
count
ε


person variance distribution (N=1000) for $\delta = 1.0$
count
0
100
200
300
400
500
600
700
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1e−30
$\sigma^2(p)$


seed variance distribution (N=1000) for $\delta = 1.0$
count
0
20
40
60
80
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
$\sigma^2(s)$

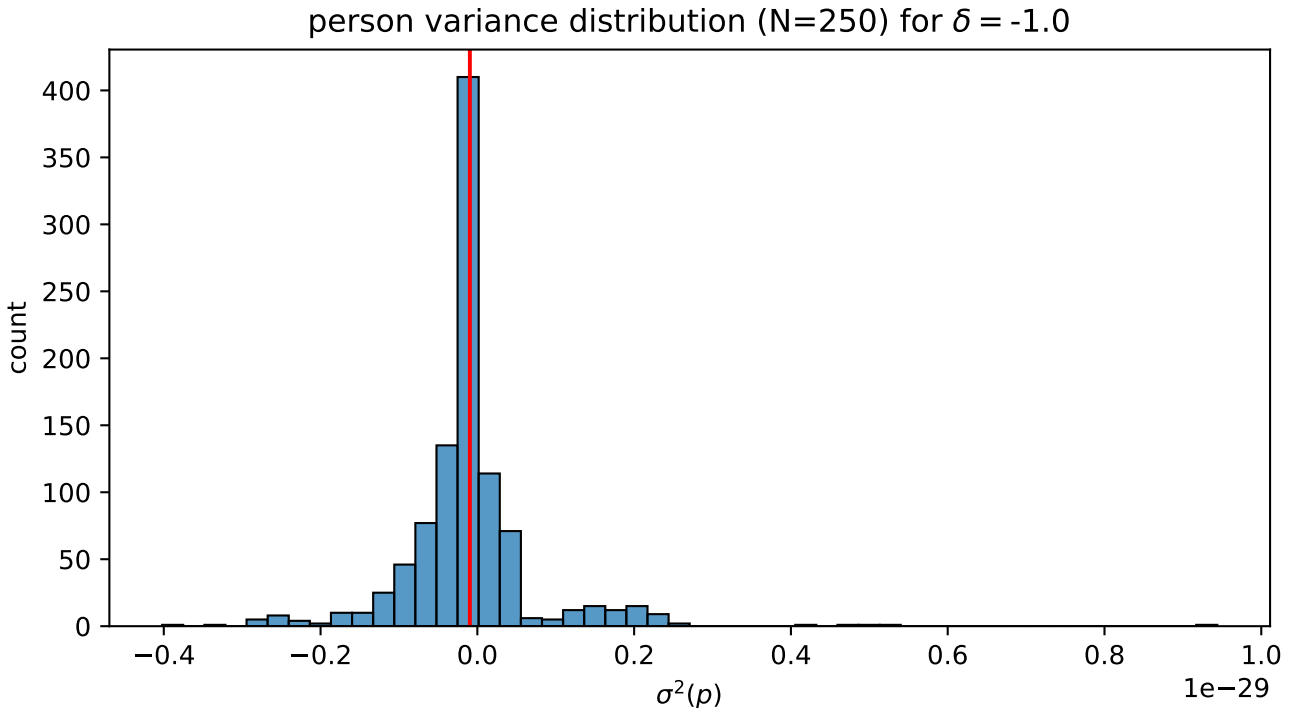

person x seed variance distribution (N=1000) for $\delta = 1.0$
count
0
100
200
300
400
500
600
700
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
1e−30
$\sigma^2(ps)$

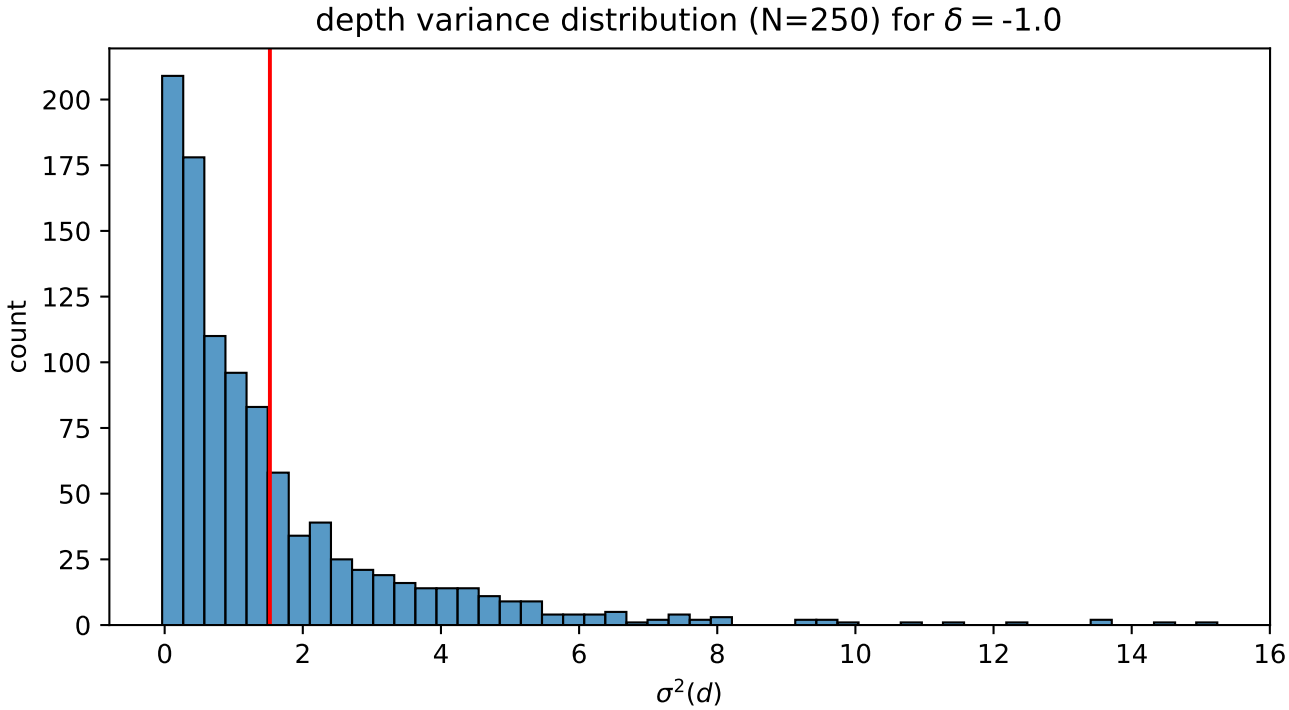

population error variance distribution (N=1000) for $\delta = 1.0$
count
$\sigma^2(\eta)$

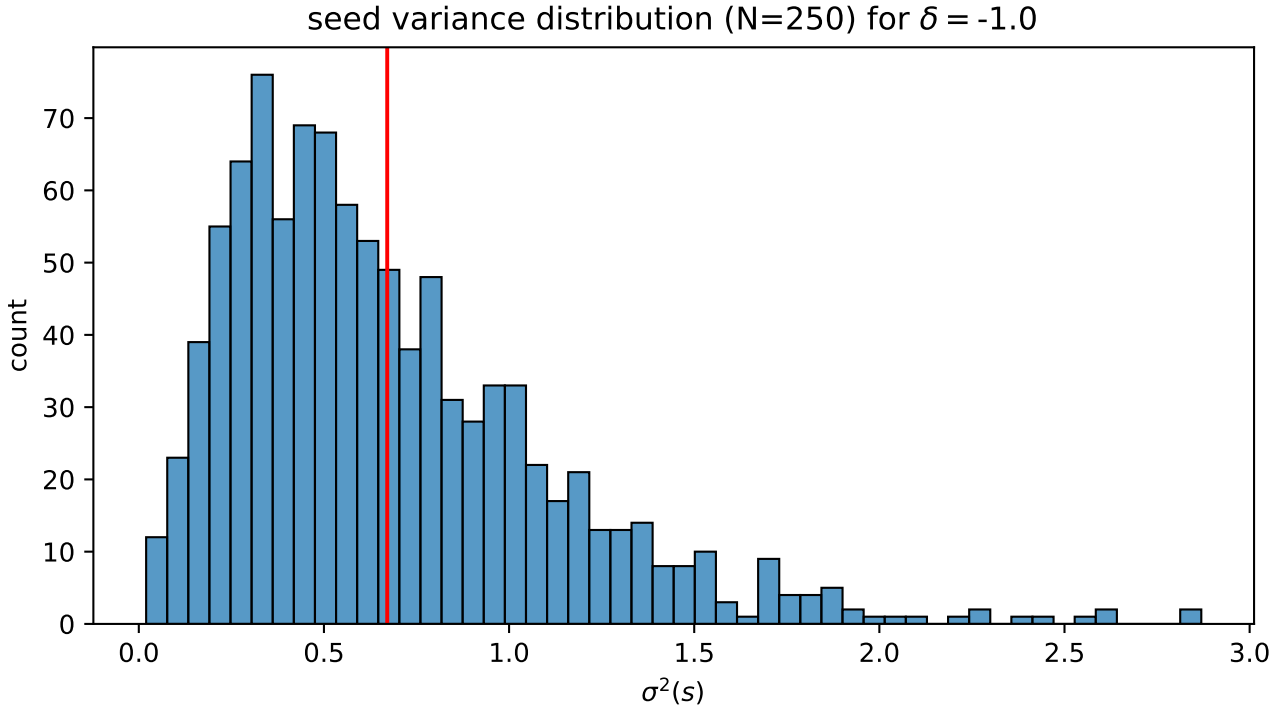

prediction error distribution (N=1000) for $\delta = 1.0$
count
ε

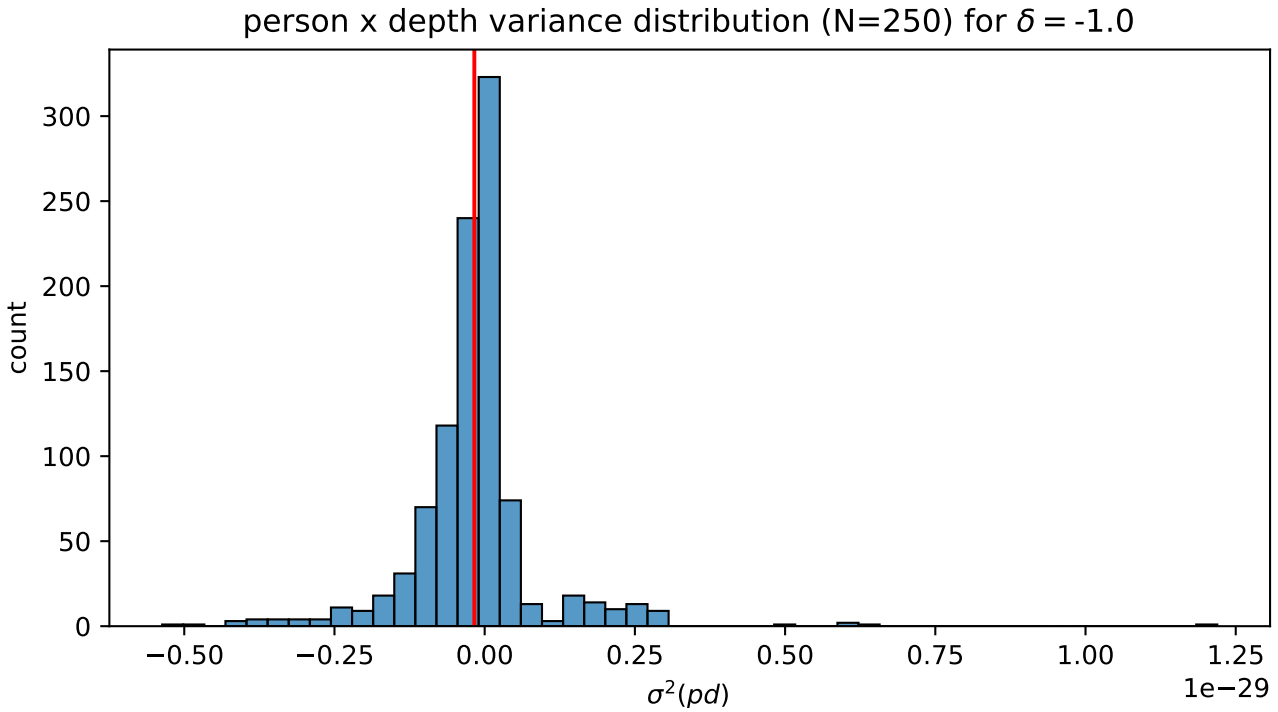
8.3.2 Results of MC study (3 facets)

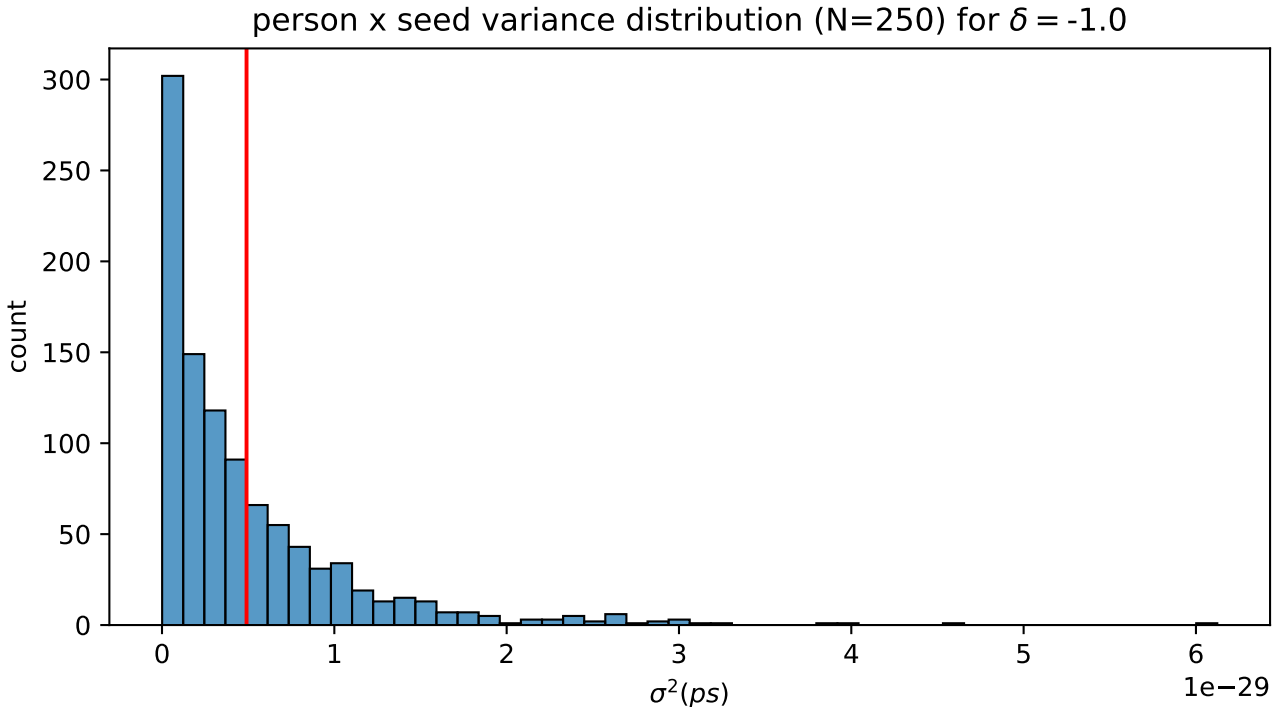
The following pages give an overview of the distribution of variance component estimates from Section 4.2. The horizontal axis shows the value of the estimates, the vertical axis gives the count of each estimate.

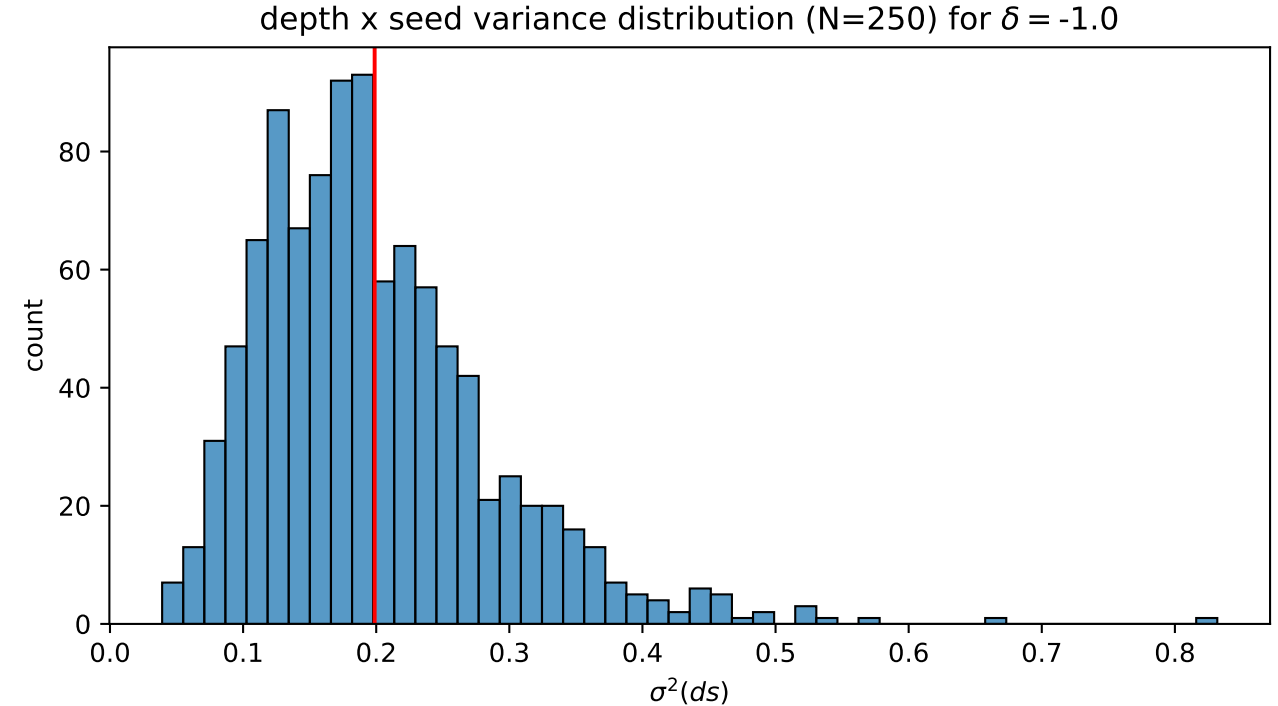

person variance distribution (N=250) for $\delta = -1.0$
count
$\sigma^2(p)$
1e−29

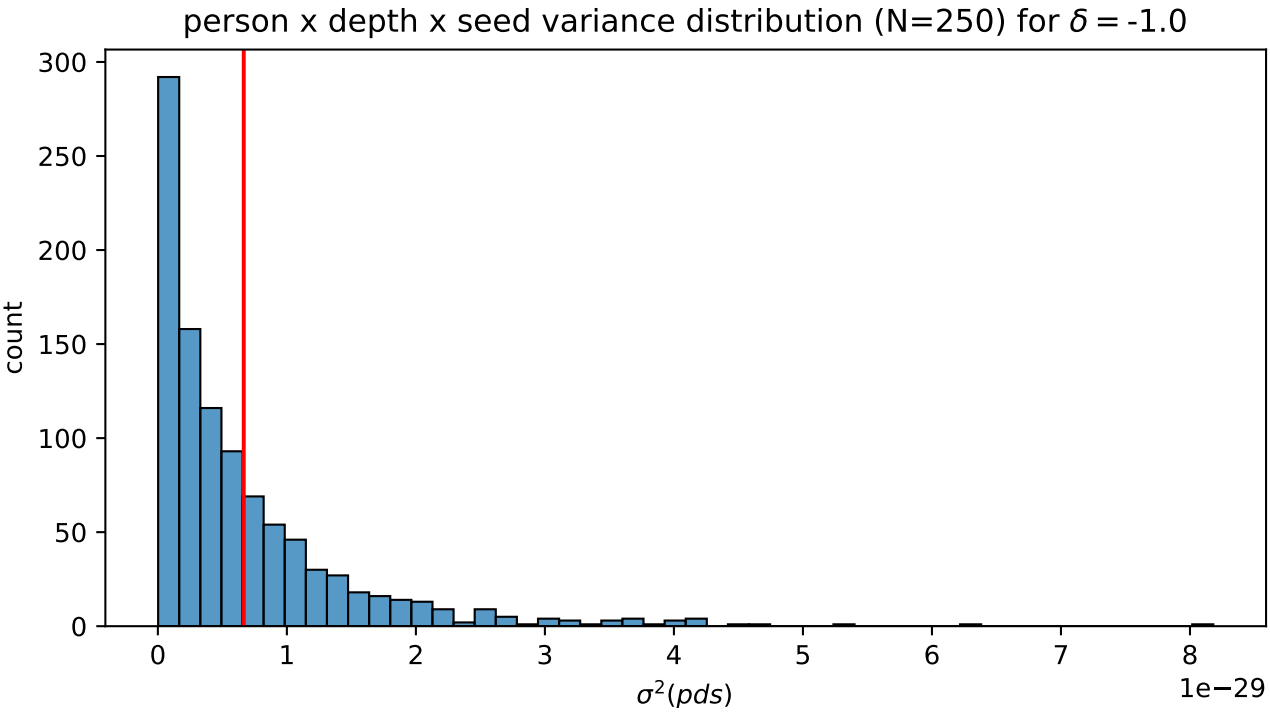

depth variance distribution (N=250) for $\delta = -1.0$
count
$\sigma^2(d)$

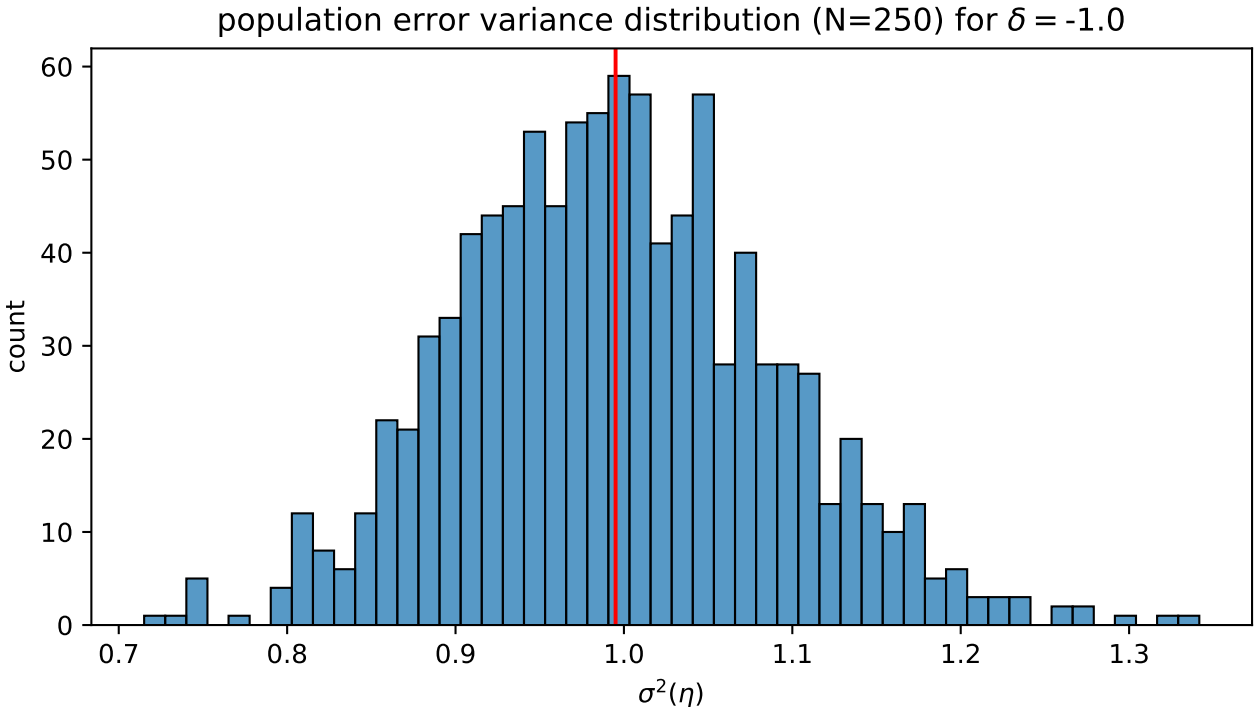

seed variance distribution (N=250) for $\delta = -1.0$
count
$\sigma^2(s)$

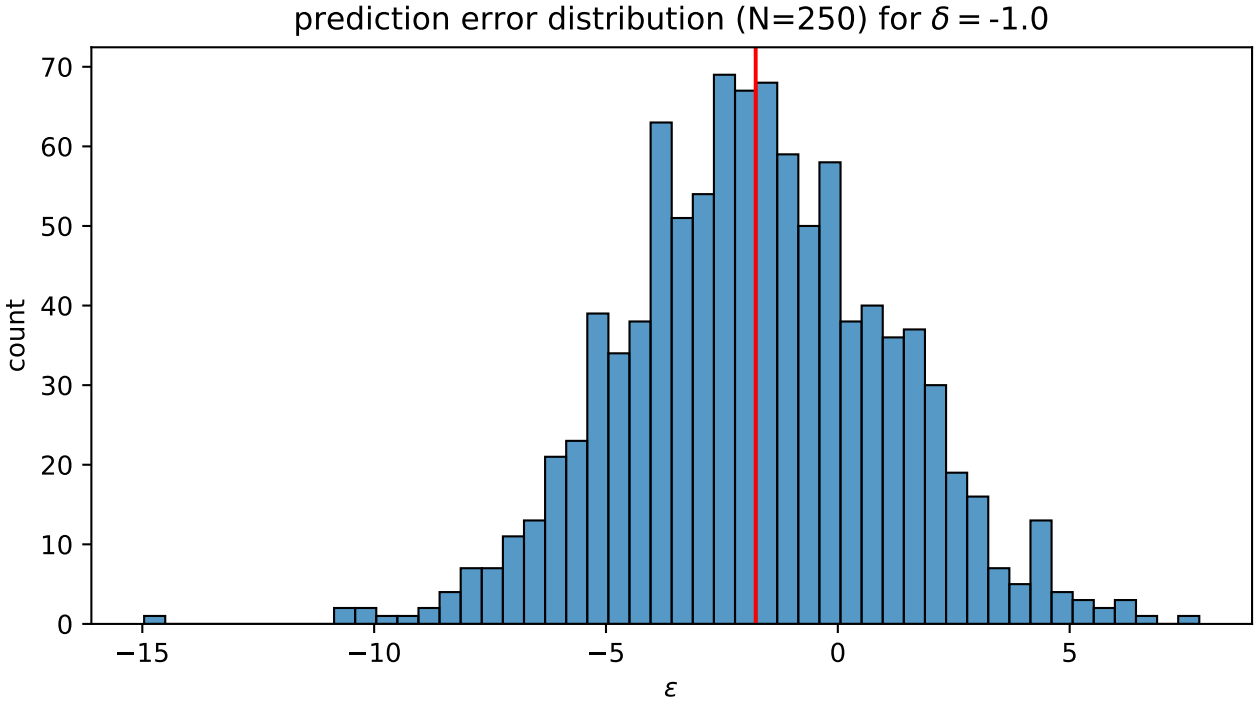

person x depth variance distribution (N=250) for $\delta = -1.0$
count
300
250
200
150
100
50
0
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
1.25
1e−29
$\sigma^2(pd)$

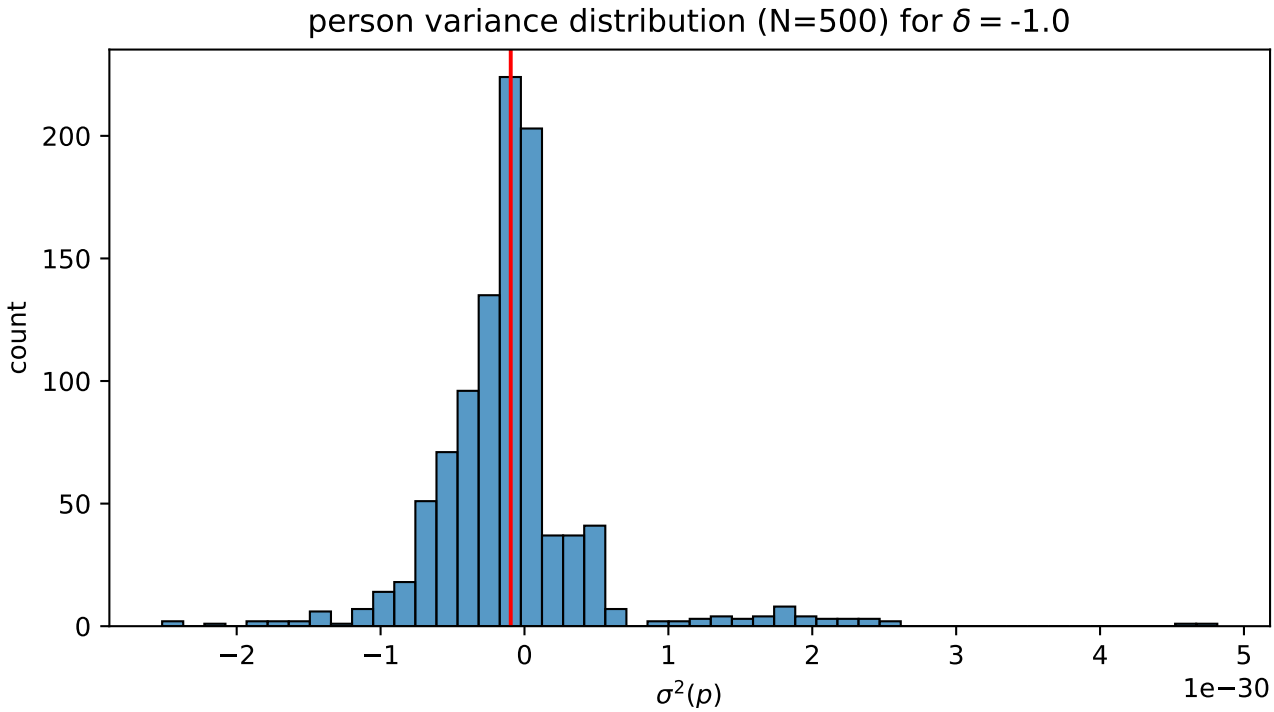

person x seed variance distribution (N=250) for $\delta = -1.0$
count
300
250
200
150
100
50
0
0
1
2
3
4
5
6
1e−29
$\sigma^2(ps)$

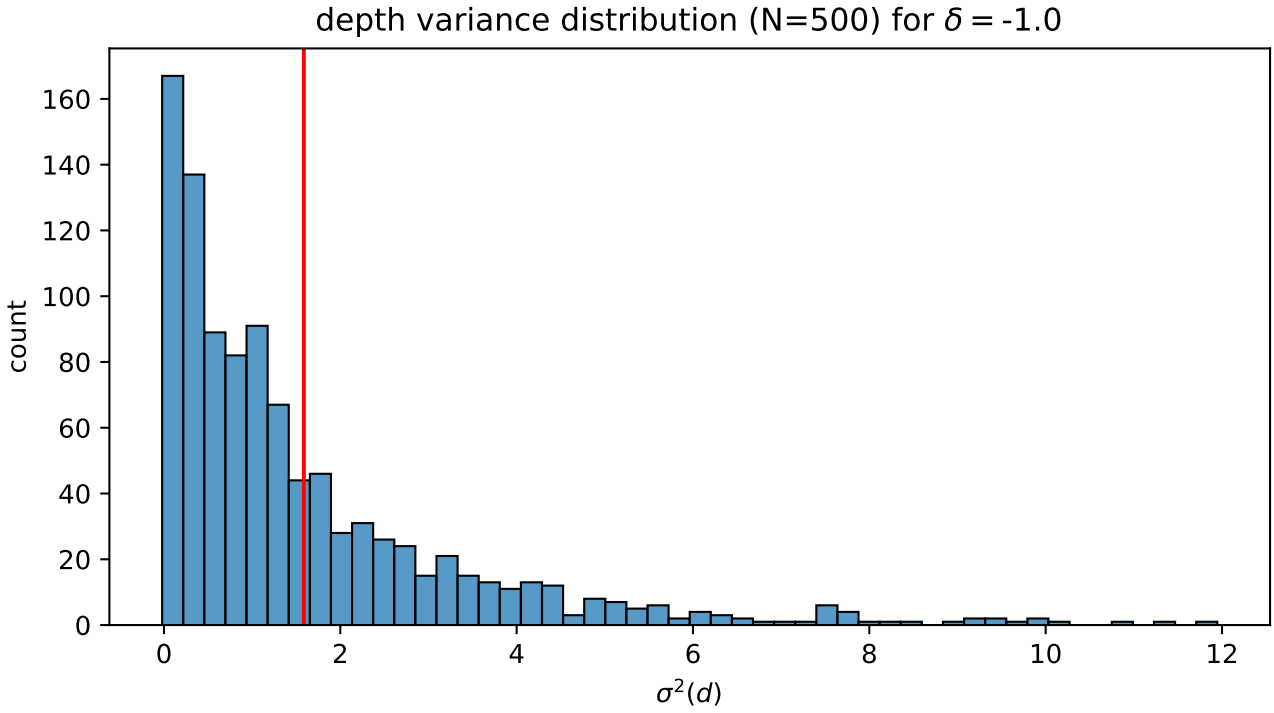

depth x seed variance distribution (N=250) for $\delta = -1.0$
count
80
60
40
20
0
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
$\sigma^2(ds)$

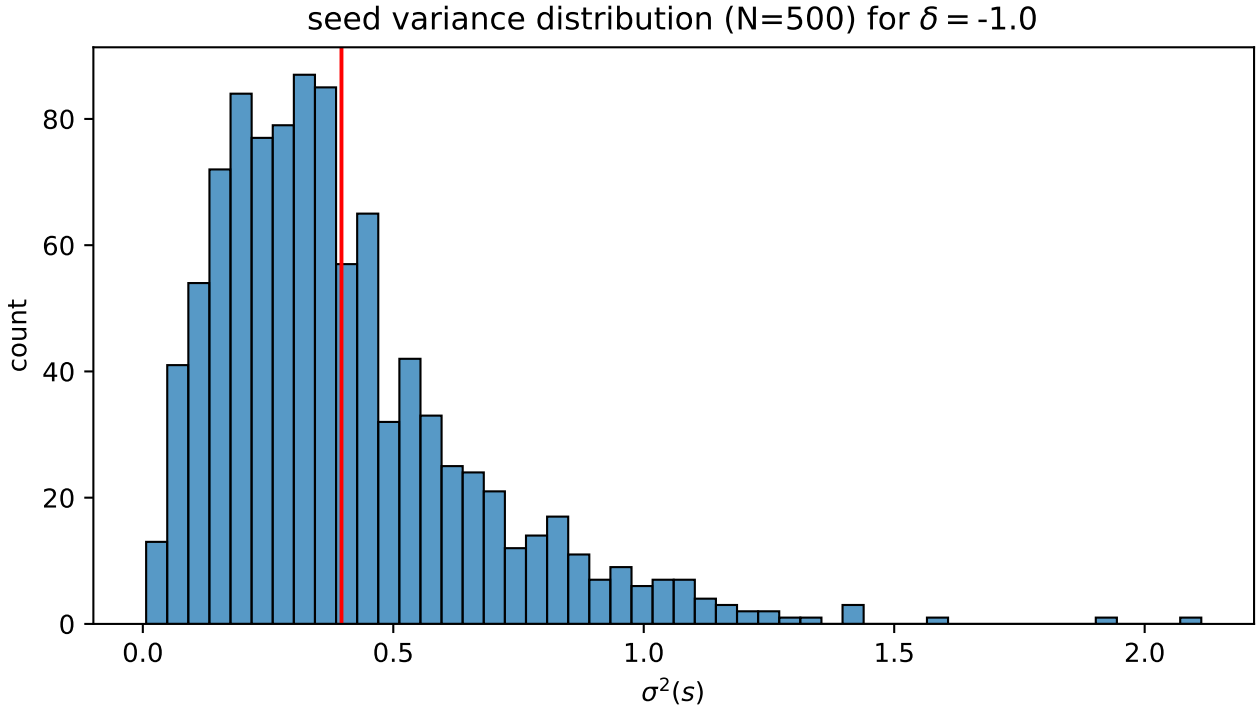

person x depth x seed variance distribution (N=250) for $\delta = -1.0$
count
$\sigma^2(pds)$
1e−29

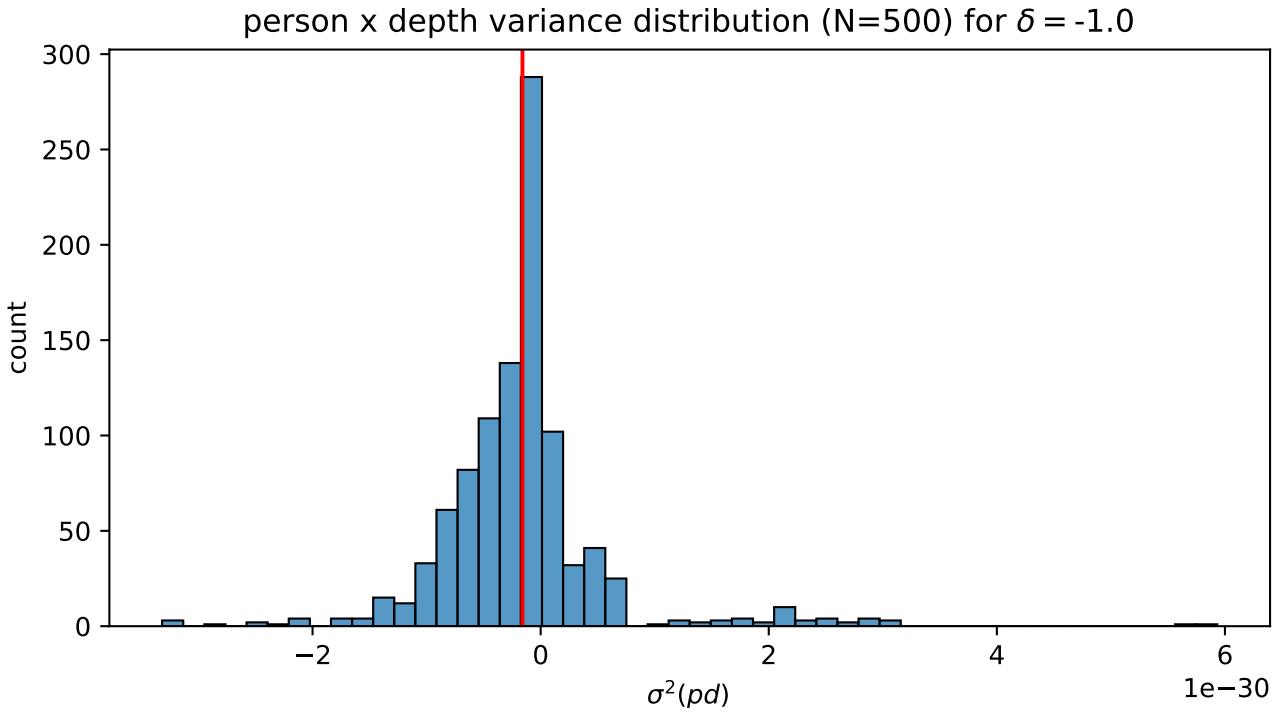

population error variance distribution (N=250) for $\delta = -1.0$
count
$\sigma^2(\eta)$

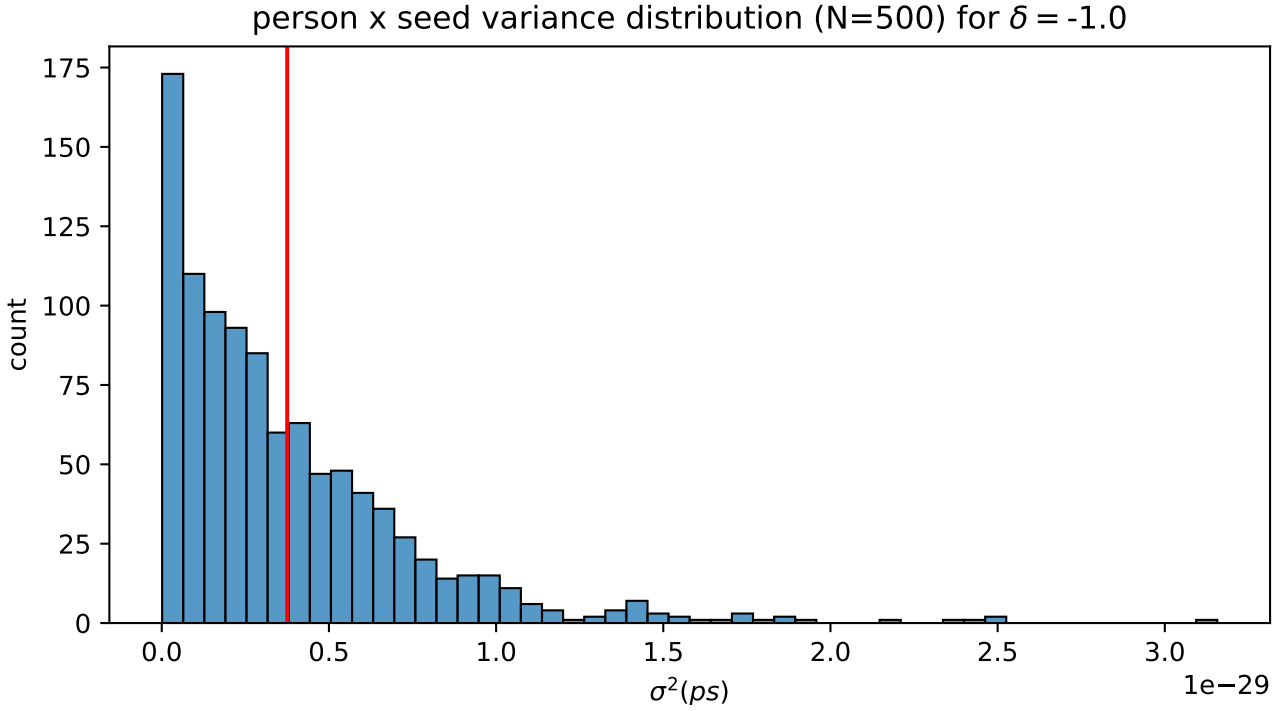

prediction error distribution (N=250) for $\delta = -1.0$
count
ε

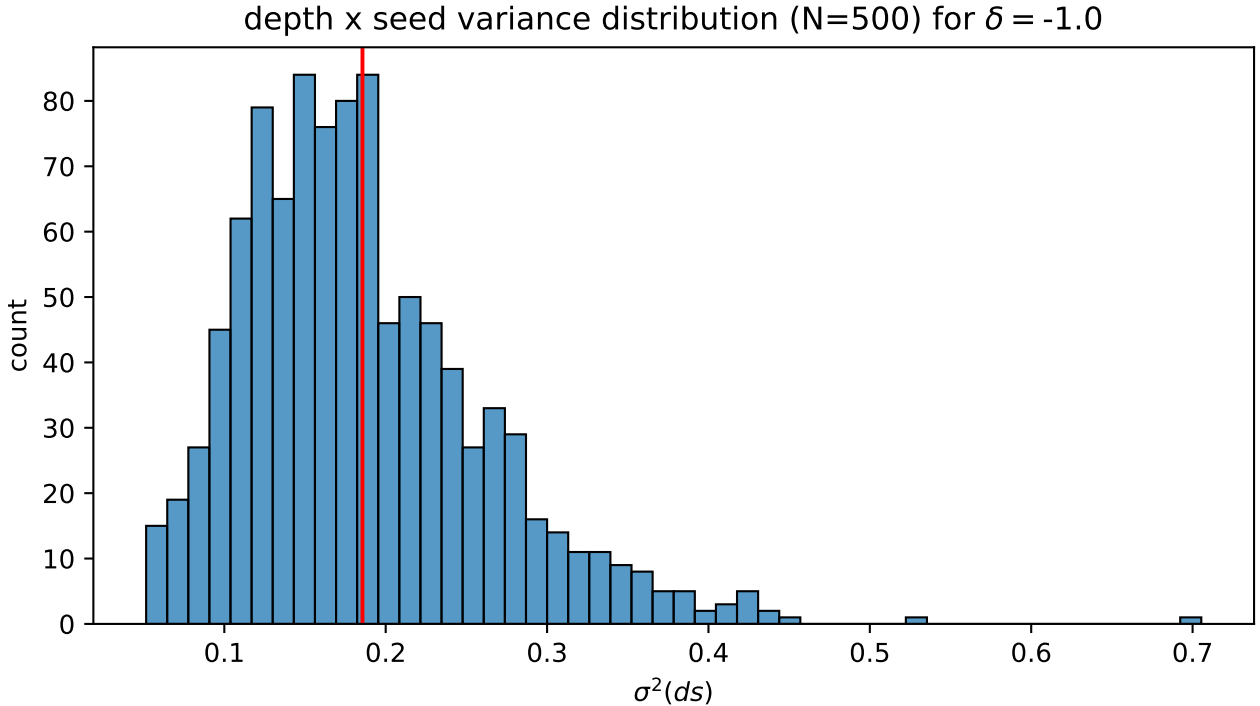

person variance distribution (N=500) for $\delta = -1.0$
count
$\sigma^2(p)$
1e−30

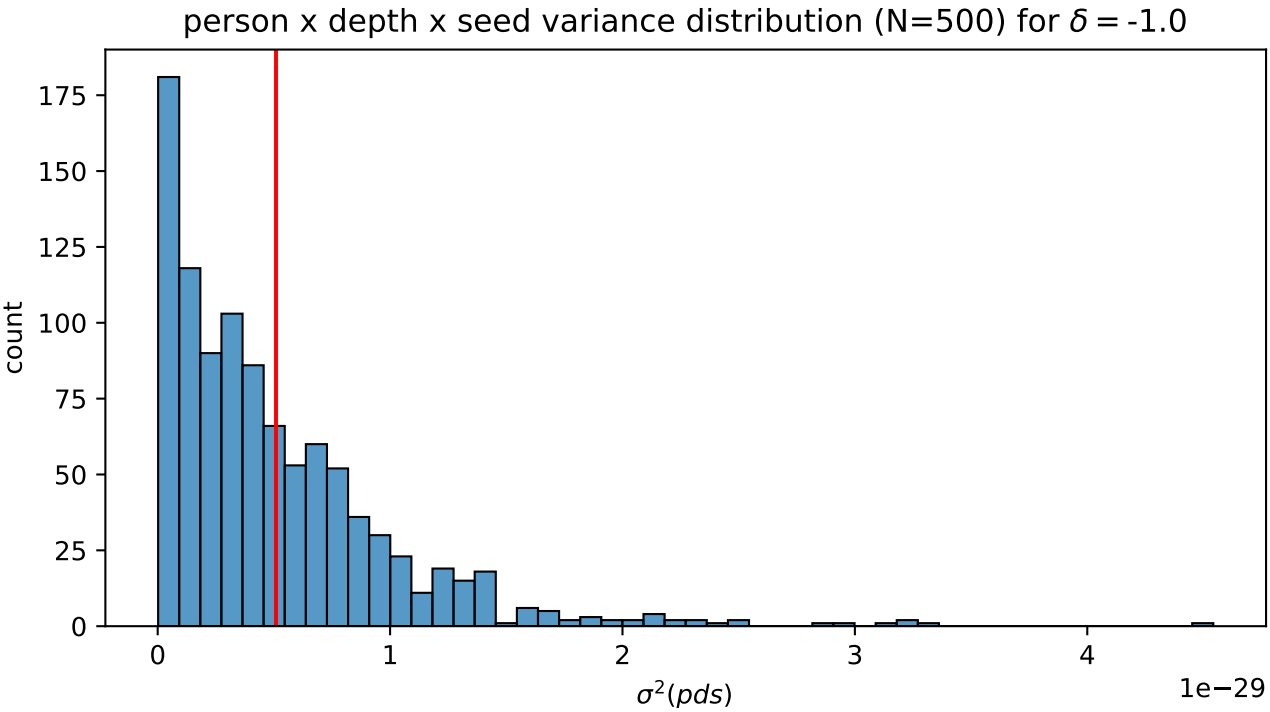

depth variance distribution (N=500) for $\delta = -1.0$
count
$\sigma^2(d)$

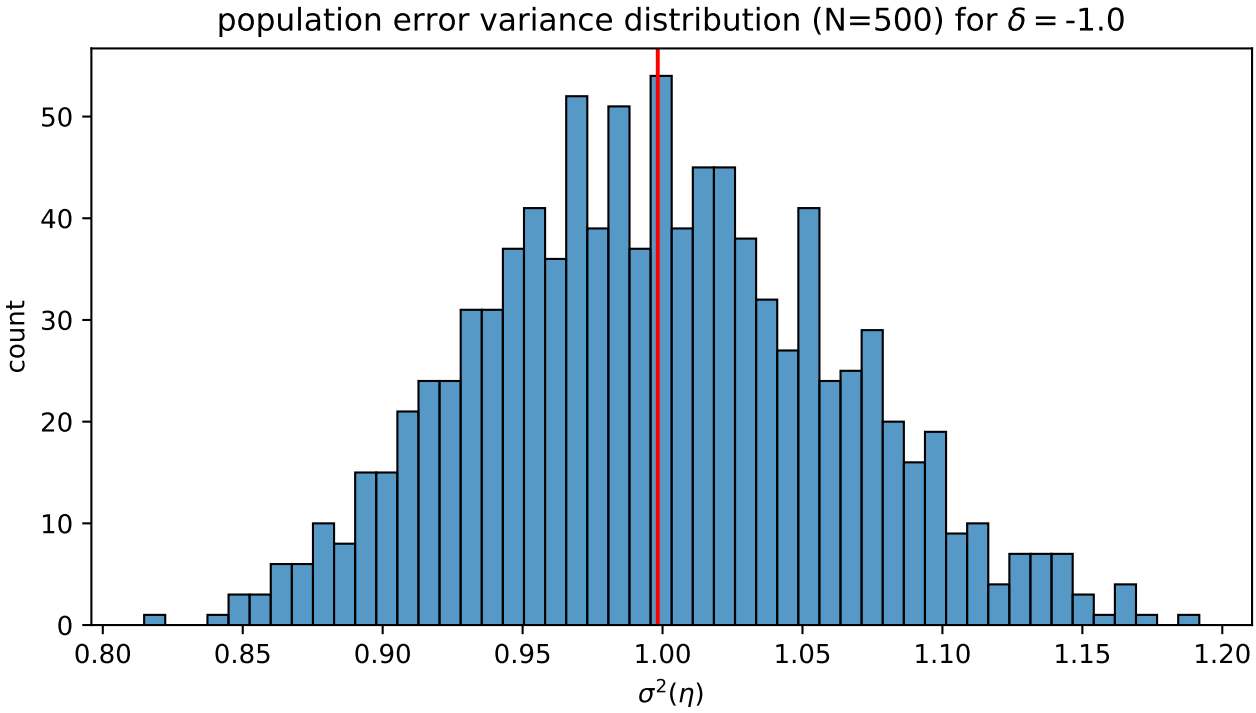

seed variance distribution (N=500) for $\delta = -1.0$
count
$\sigma^2(s)$

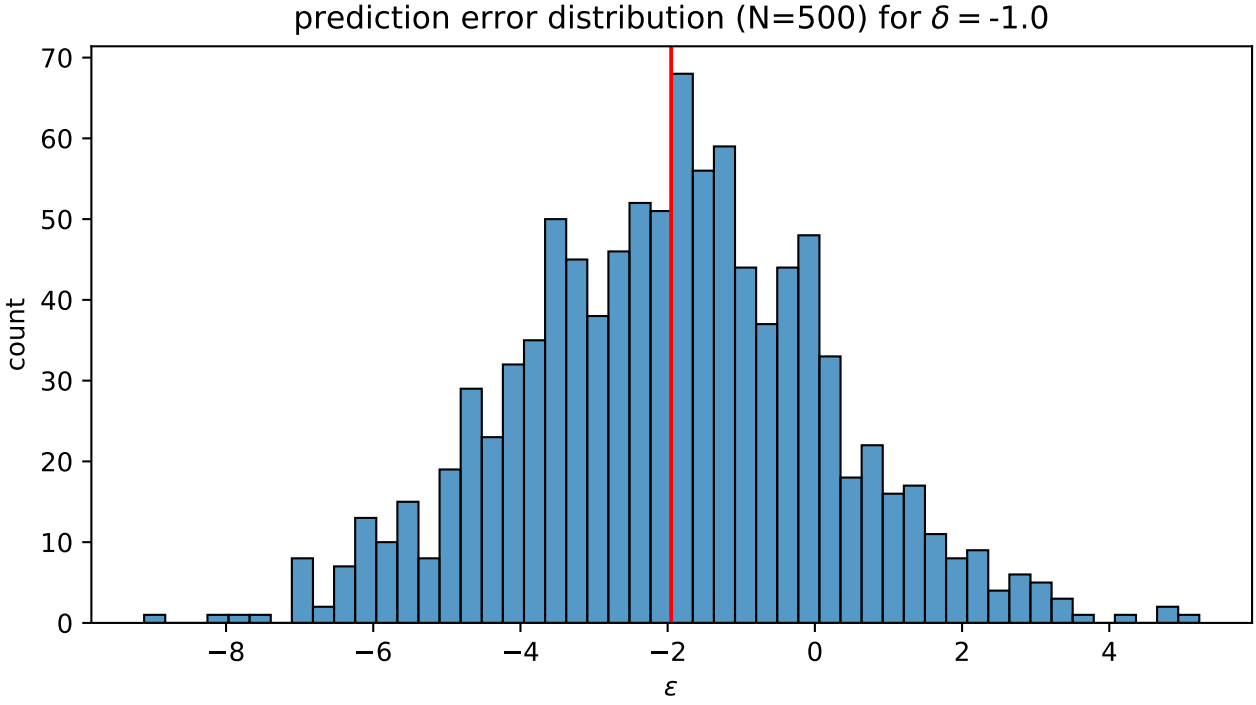

person x depth variance distribution (N=500) for $\delta = -1.0$
count
$\sigma^2(pd)$
1e−30

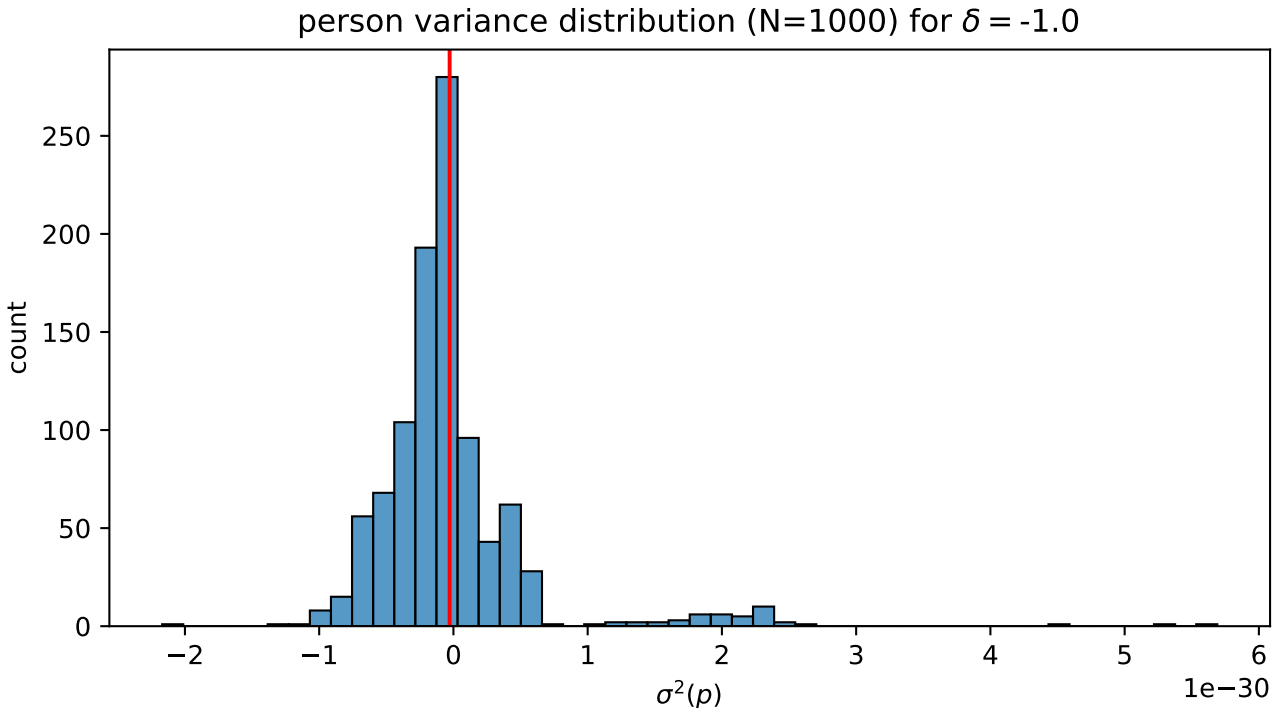

person x seed variance distribution (N=500) for $\delta = -1.0$
count
$\sigma^2(ps)$
1e−29

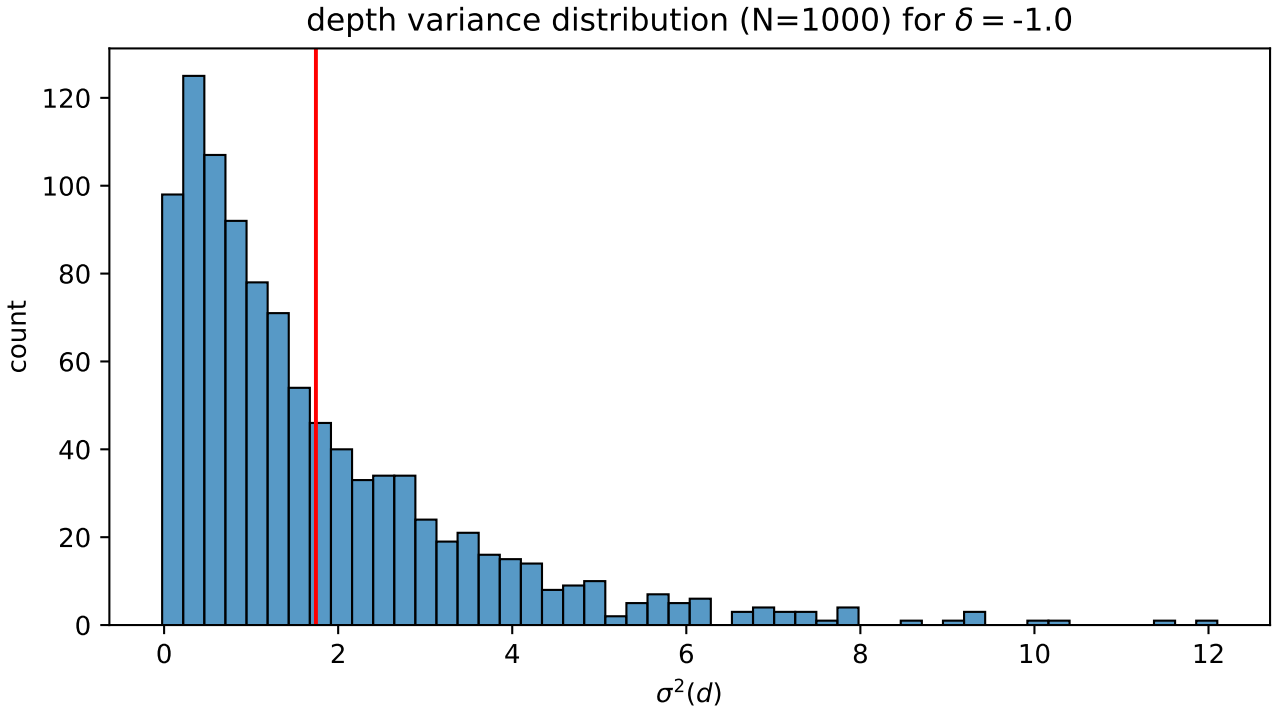

depth x seed variance distribution (N=500) for $\delta = -1.0$
count
$\sigma^2(ds)$

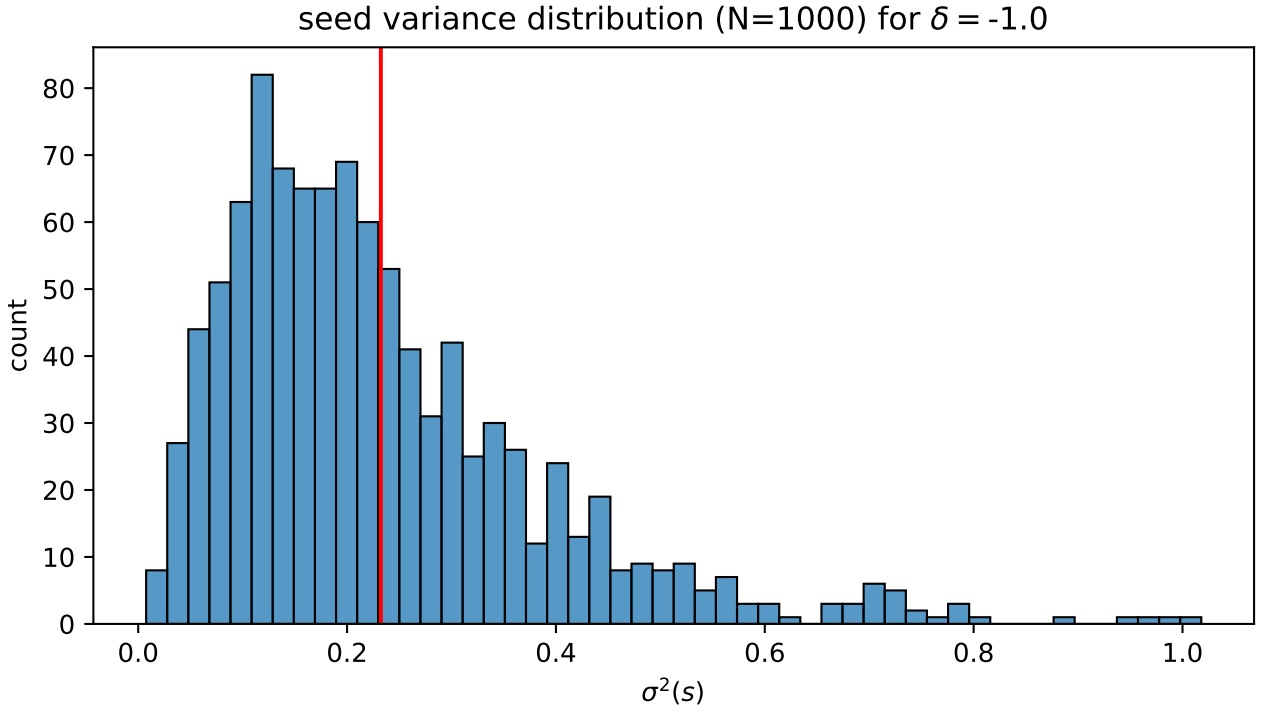

person x depth x seed variance distribution (N=500) for $\delta = -1.0$
count
$\sigma^2(pds)$
1e−29

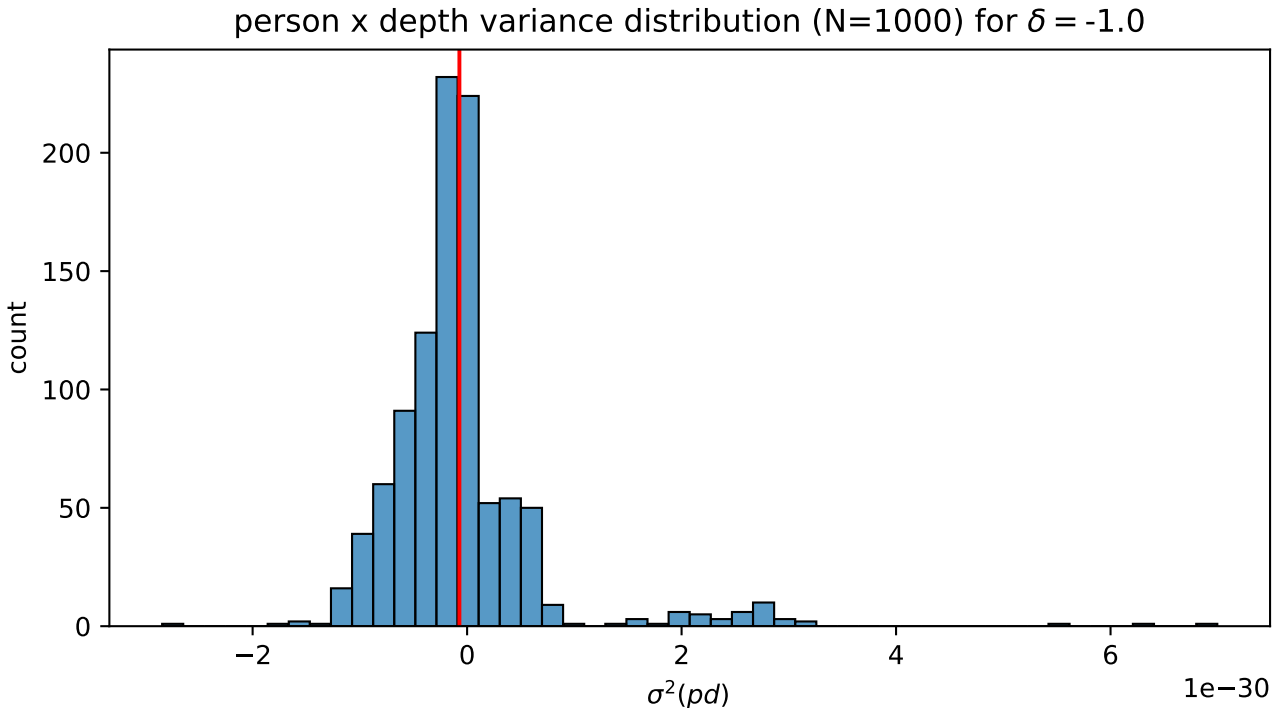

population error variance distribution (N=500) for $\delta = -1.0$
count
$\sigma^2(\eta)$

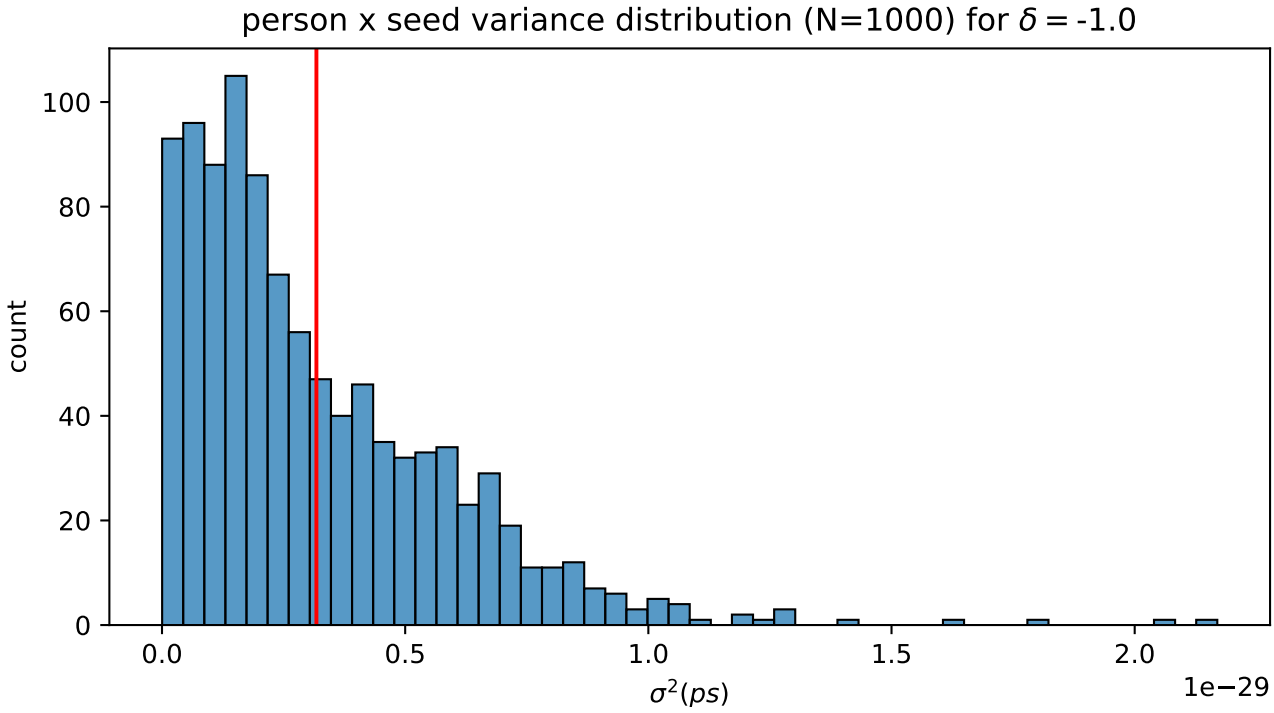

prediction error distribution (N=500) for $\delta = -1.0$
count
ε

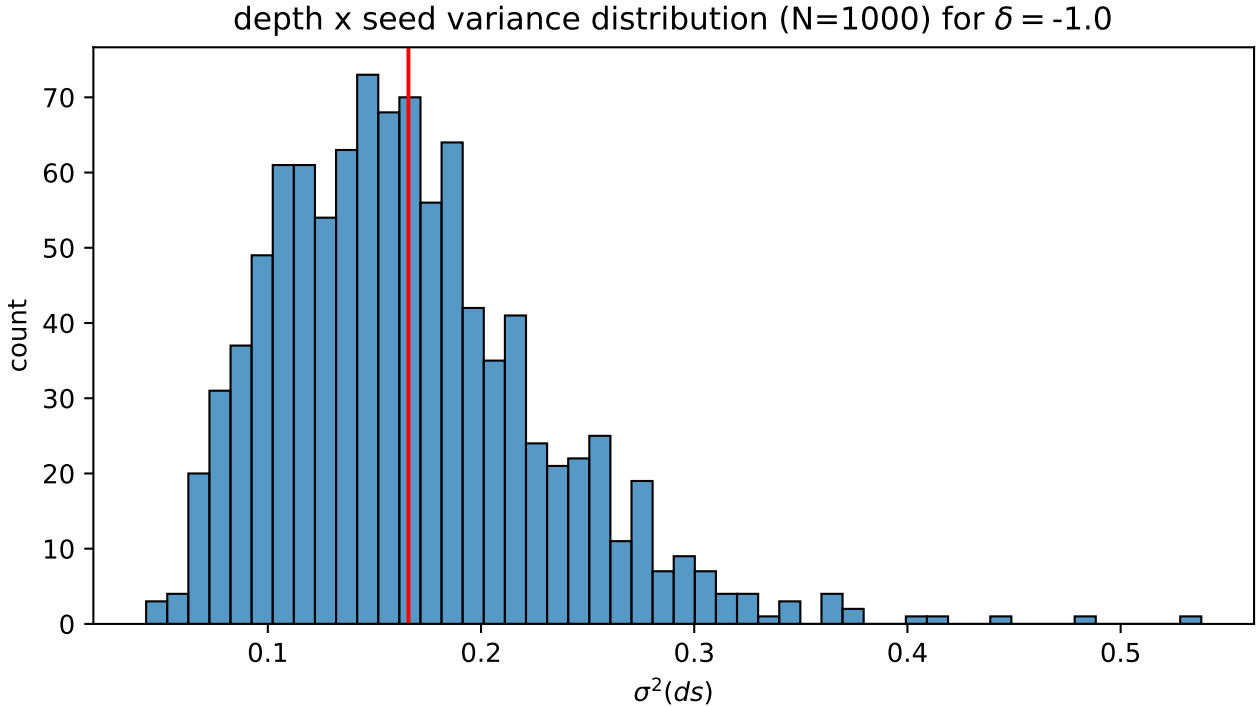

person variance distribution (N=1000) for $\delta = -1.0$
count
$\sigma^2(p)$
1e−30

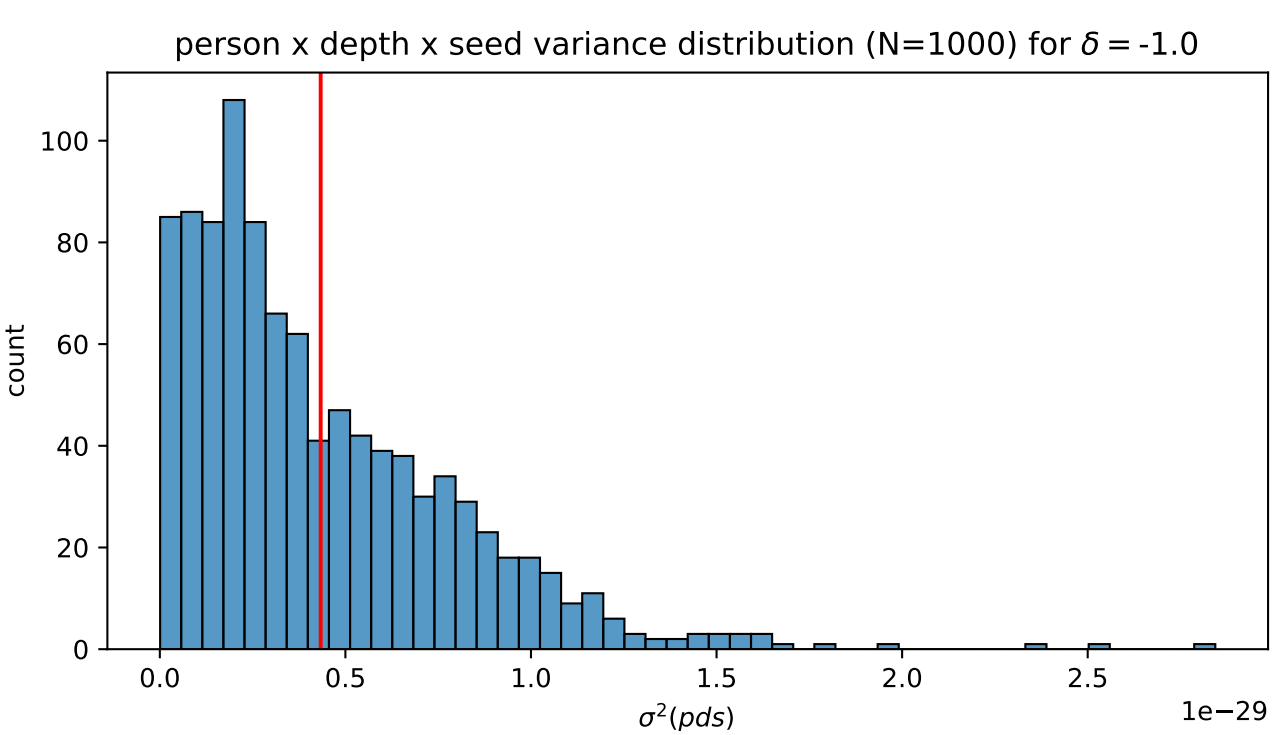

depth variance distribution (N=1000) for $\delta = -1.0$
count
$\sigma^2(d)$

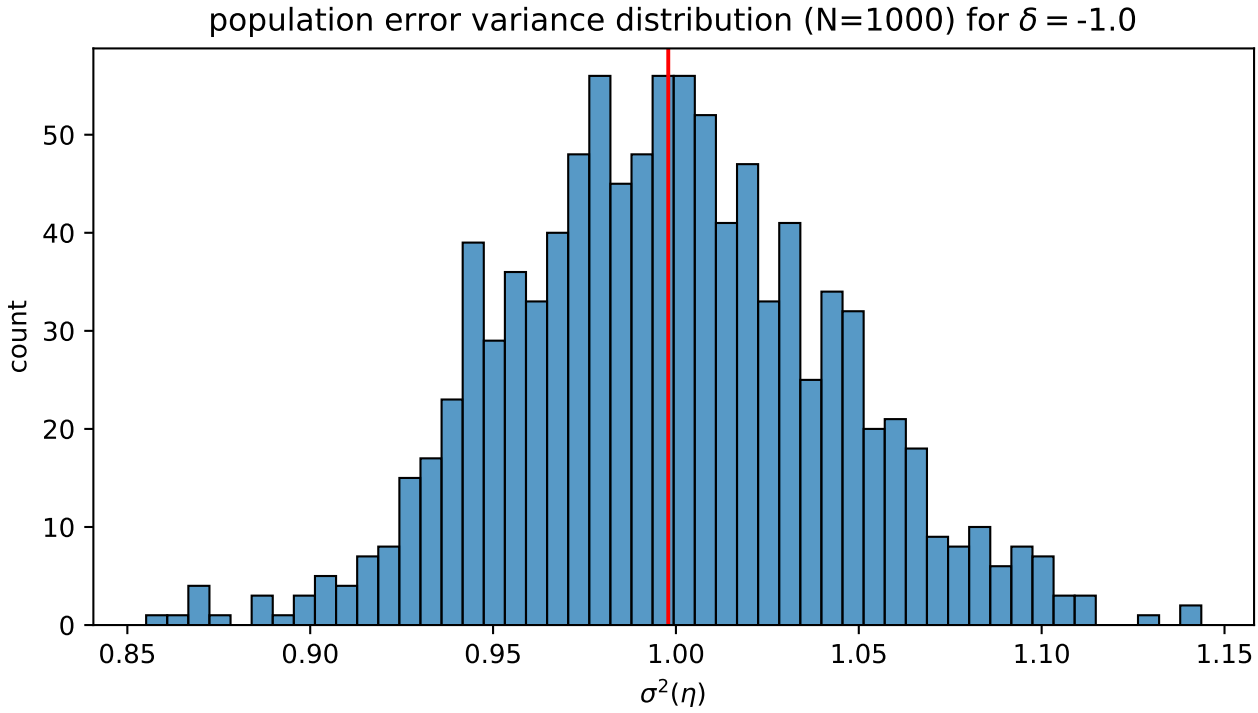

seed variance distribution (N=1000) for $\delta = -1.0$
count
$\sigma^2(s)$

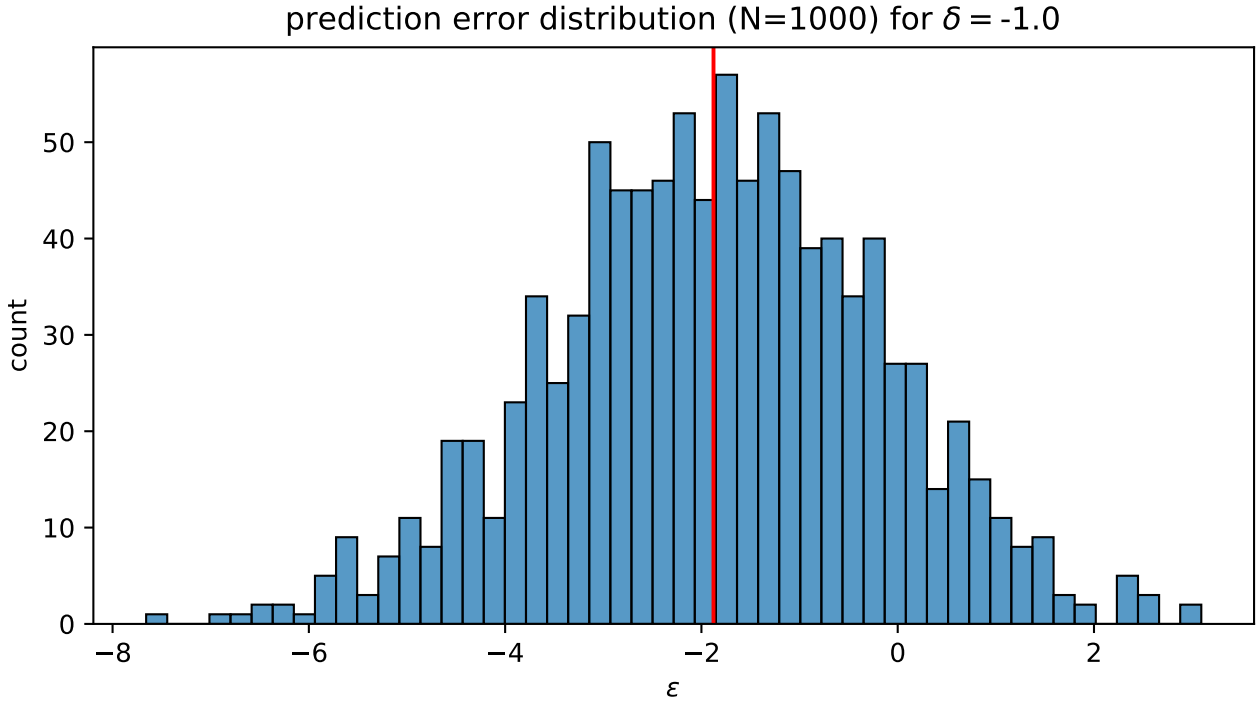

person x depth variance distribution (N=1000) for $\delta = -1.0$
count
200
150
100
50
0
−2
0
2
4
6
$\sigma^2(pd)$
1e−30

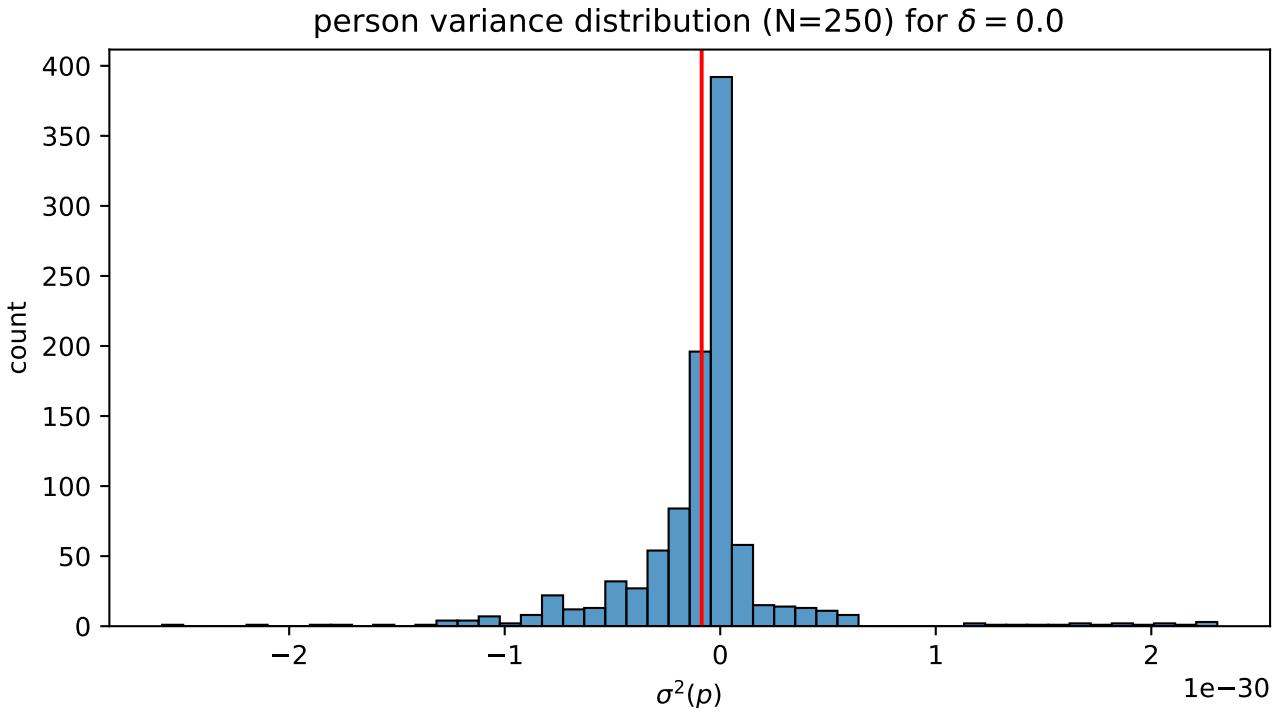

person x seed variance distribution (N=1000) for $\delta = -1.0$
count
100
80
60
40
20
0
0.0
0.5
1.0
1.5
2.0
$\sigma^2(ps)$
1e−29

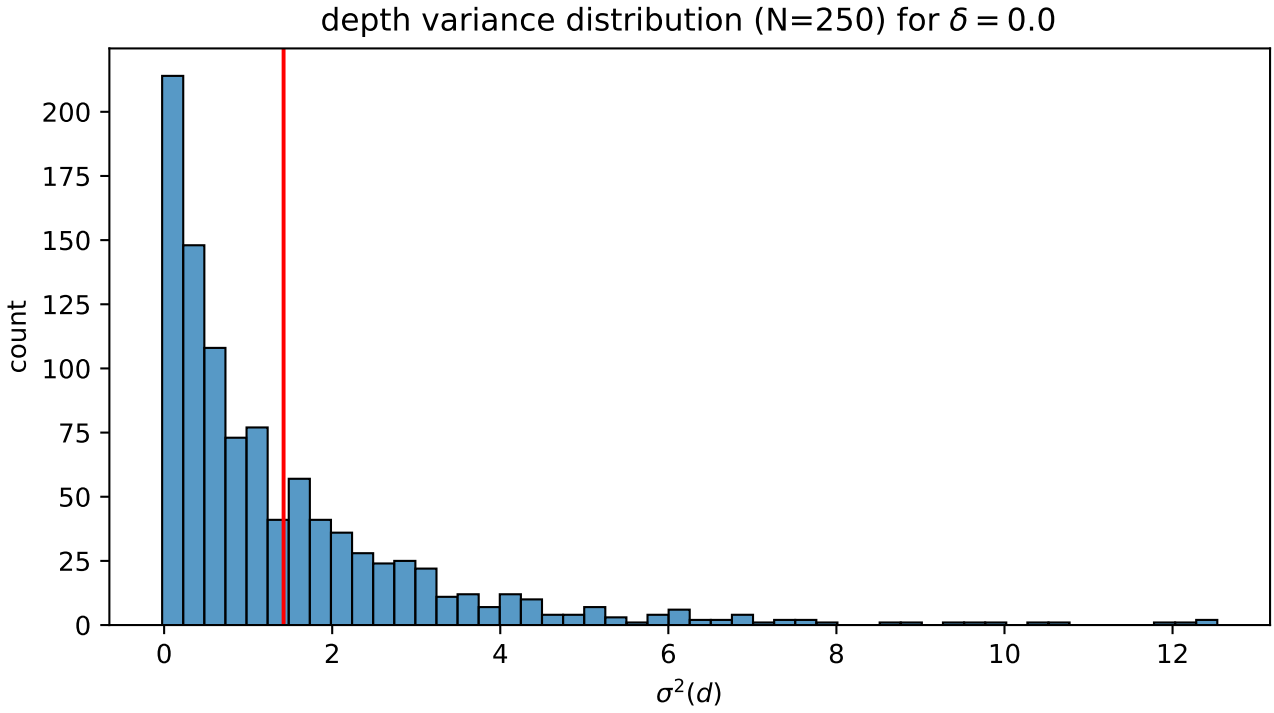

depth x seed variance distribution (N=1000) for $\delta = -1.0$
count
70
60
50
40
30
20
10
0
0.1
0.2
0.3
0.4
0.5
$\sigma^2(ds)$

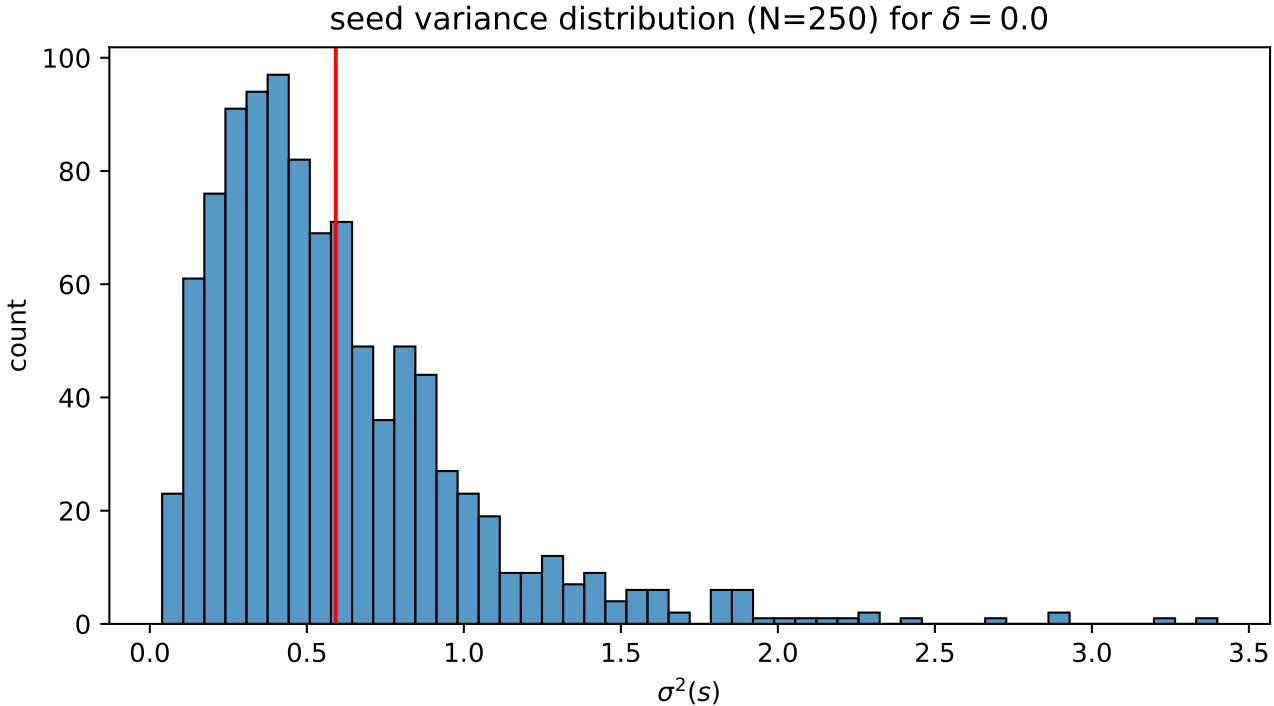

person x depth x seed variance distribution (N=1000) for $\delta = -1.0$
count
0
20
40
60
80
100
0.0
0.5
1.0
1.5
2.0
2.5
$\sigma^2(pds)$
1e−29

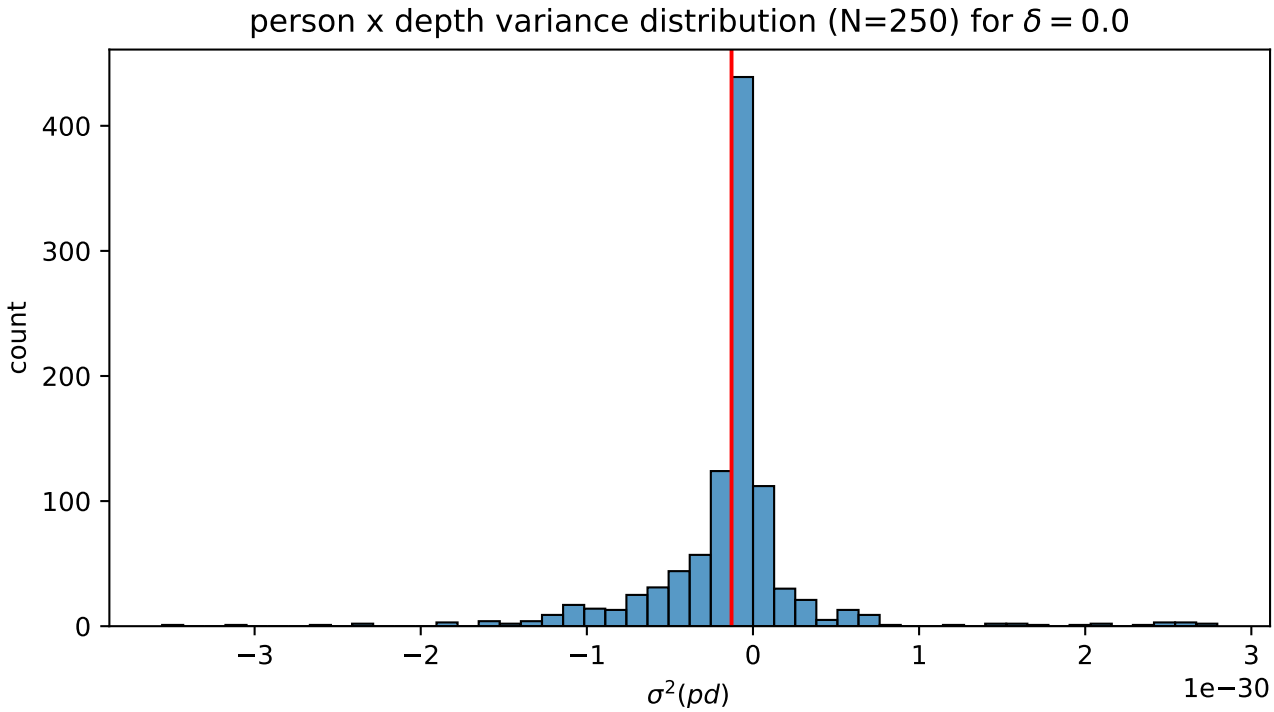

population error variance distribution (N=1000) for $\delta = -1.0$
count
0
10
20
30
40
50
0.85
0.90
0.95
1.00
1.05
1.10
1.15
$\sigma^2(\eta)$

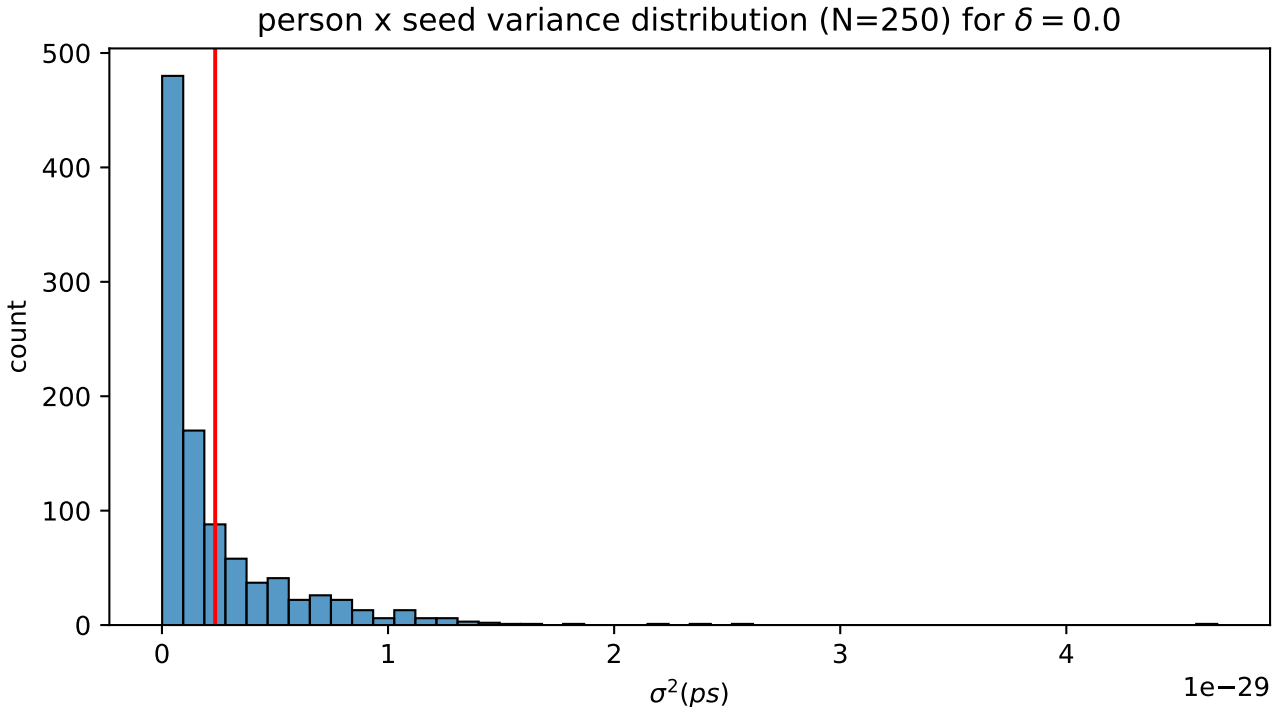

prediction error distribution (N=1000) for $\delta = -1.0$
count
0
10
20
30
40
50
−8
−6
−4
−2
0
2
ε

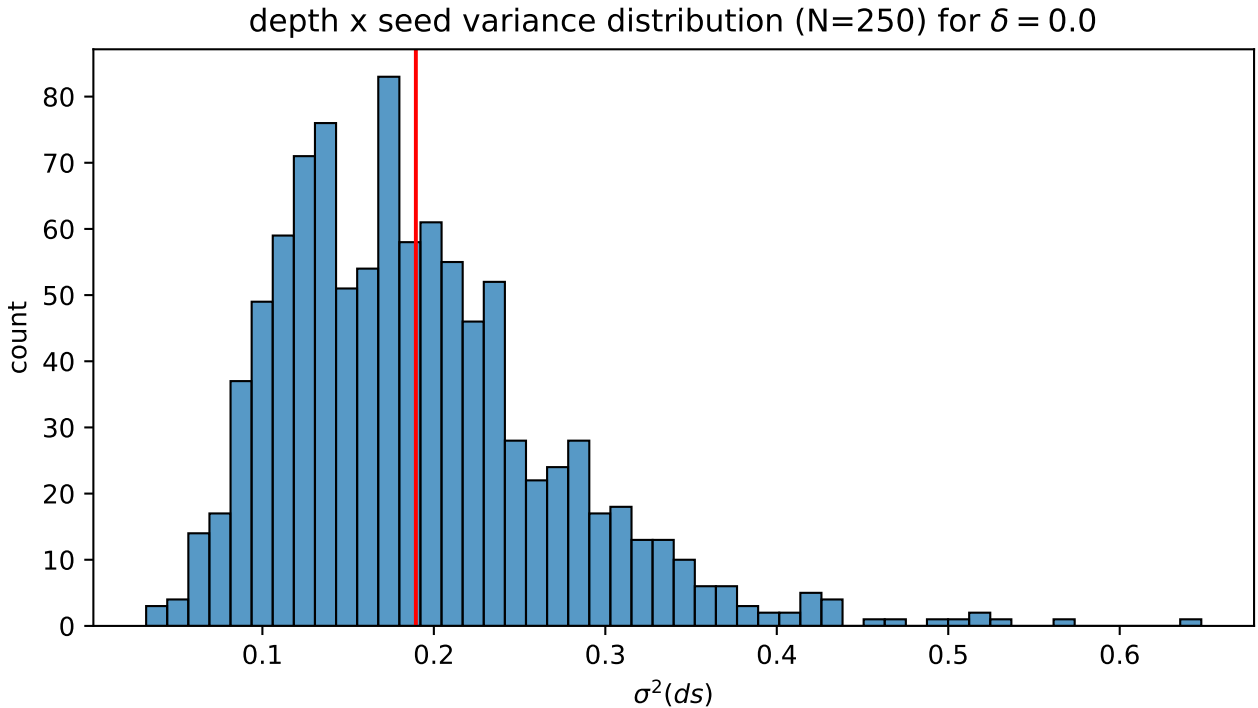

person variance distribution (N=250) for $\delta = 0.0$
count
$\sigma^2(p)$
1e−30

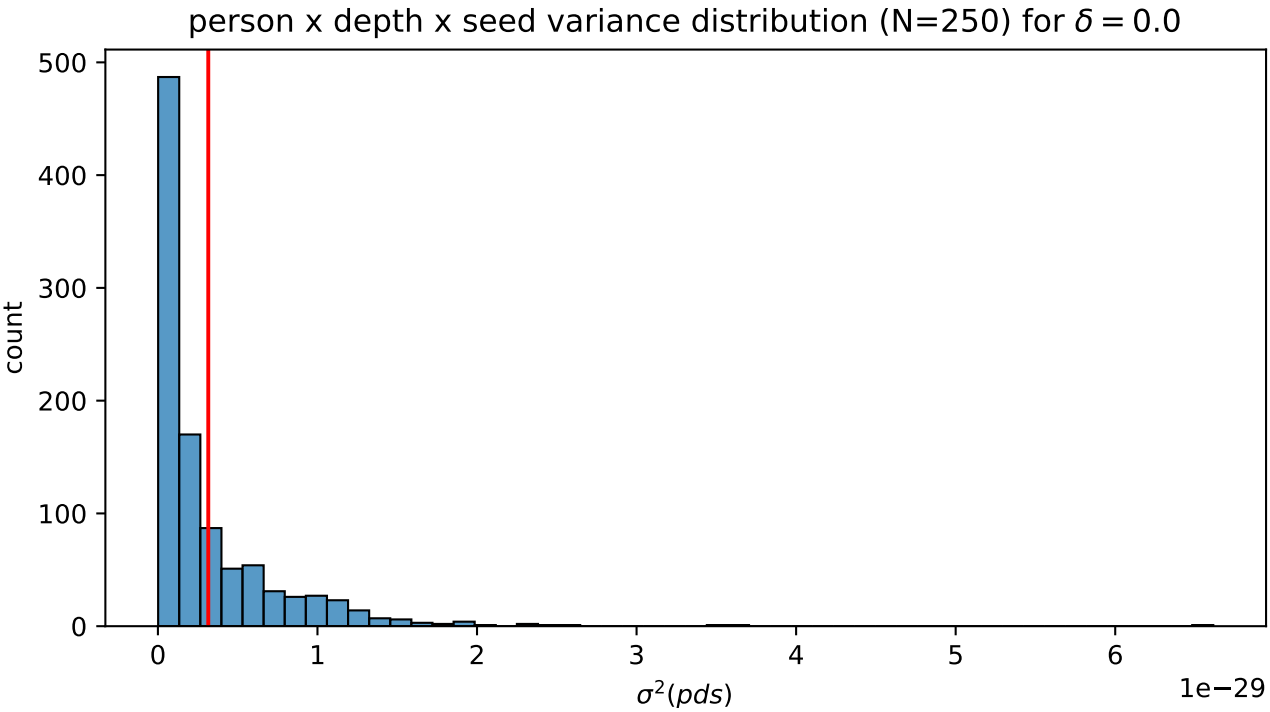

depth variance distribution (N=250) for $\delta = 0.0$
count
$\sigma^2(d)$

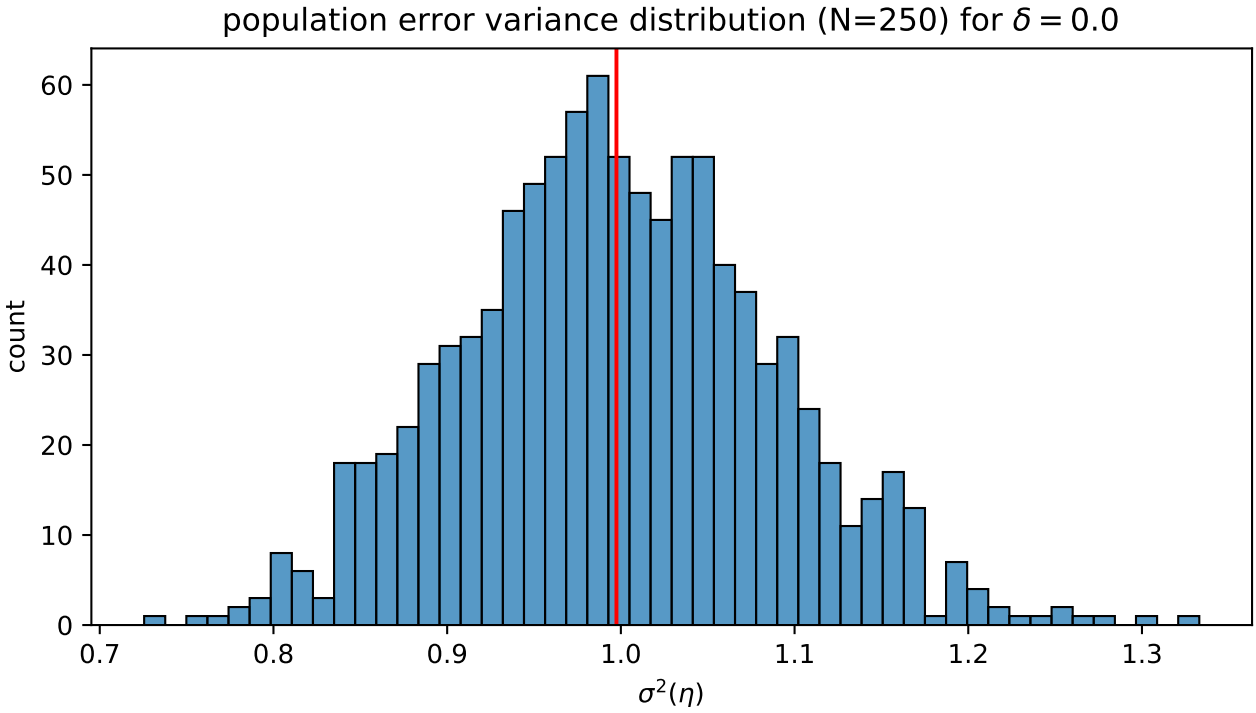

seed variance distribution (N=250) for $\delta = 0.0$
count
$\sigma^2(s)$

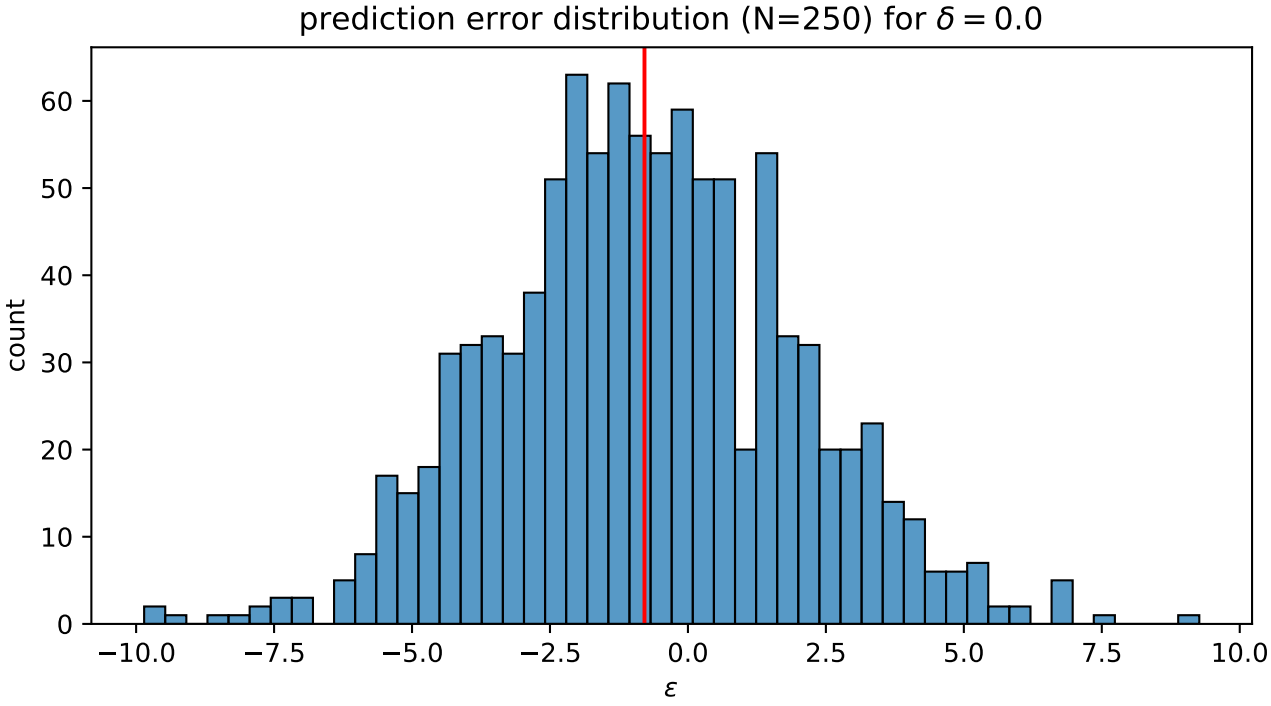

person x depth variance distribution (N=250) for $\delta = 0.0$
count
$\sigma^2(pd)$
1e−30

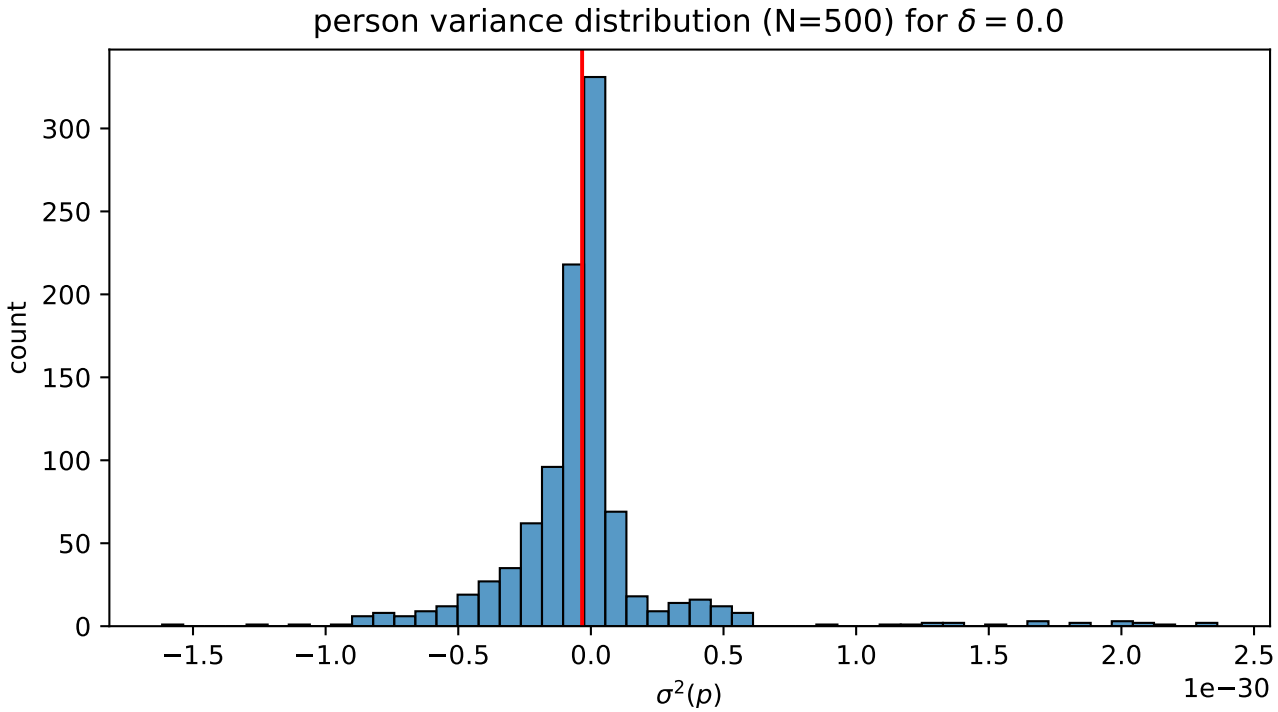

person x seed variance distribution (N=250) for $\delta = 0.0$
count
$\sigma^2(ps)$
1e−29

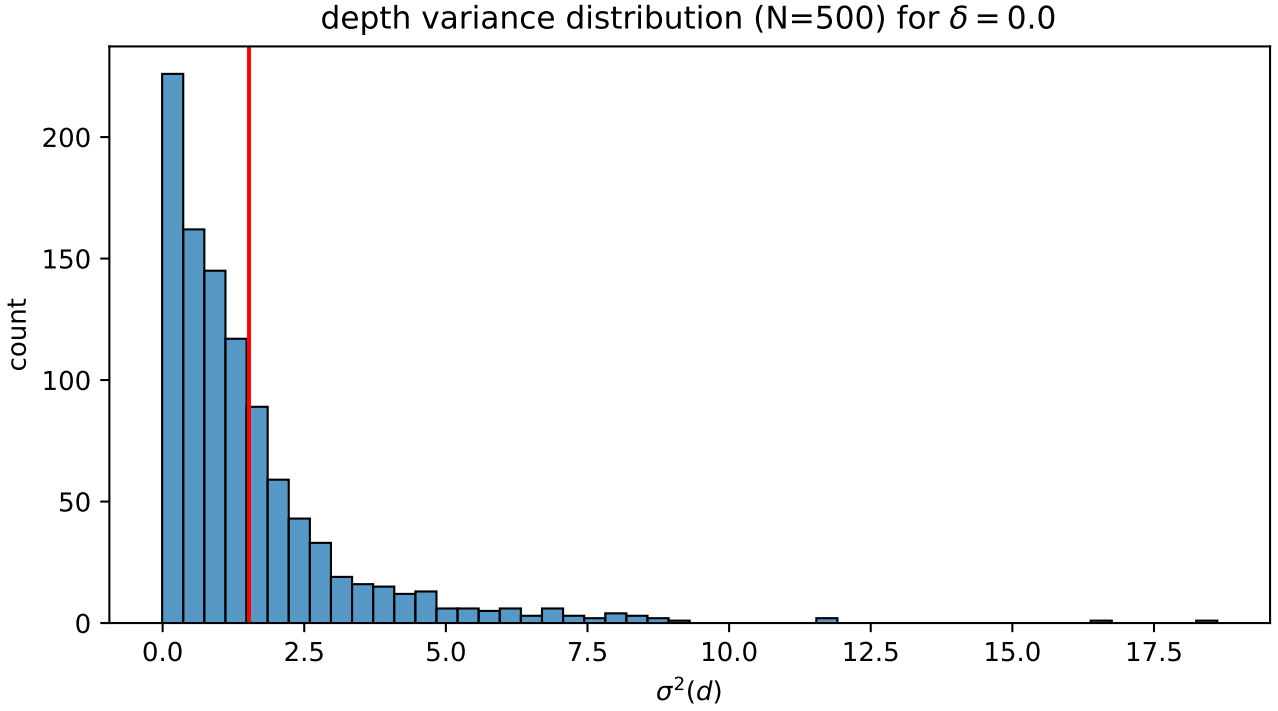

depth x seed variance distribution (N=250) for $\delta = 0.0$
count
$\sigma^2(ds)$

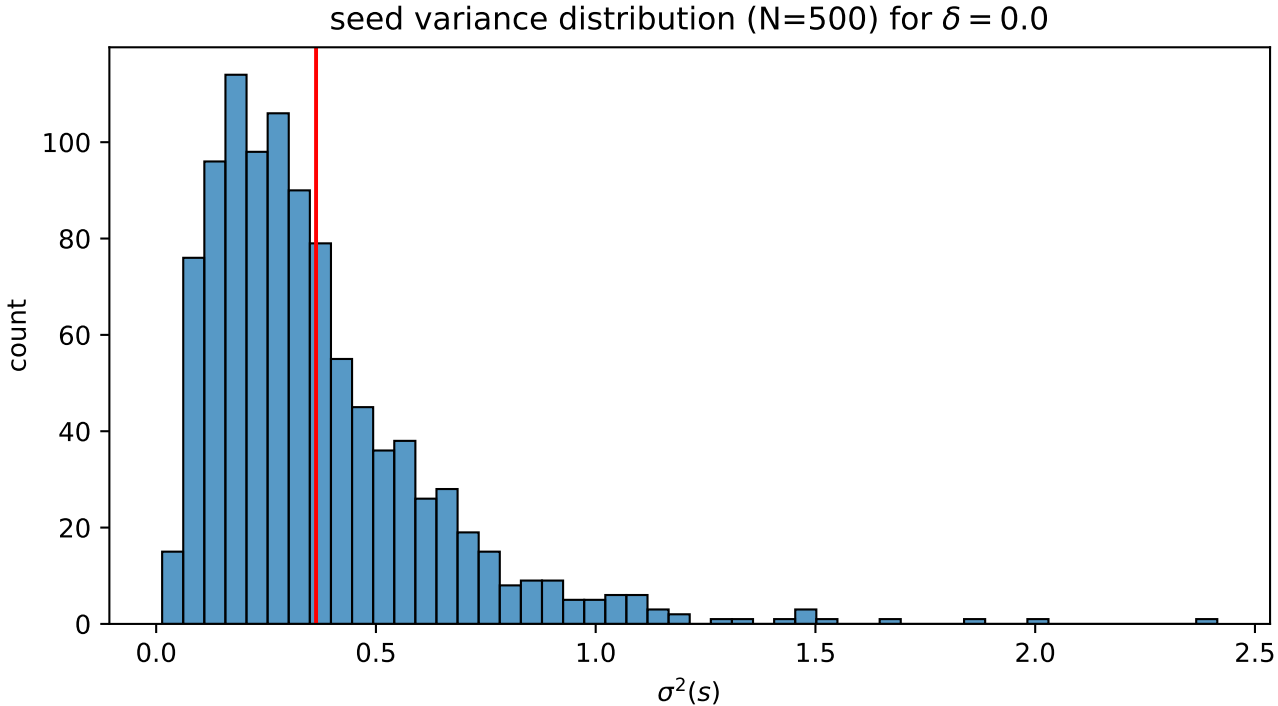

person x depth x seed variance distribution (N=250) for $\delta = 0.0$
count
$\sigma^2(pds)$
1e−29

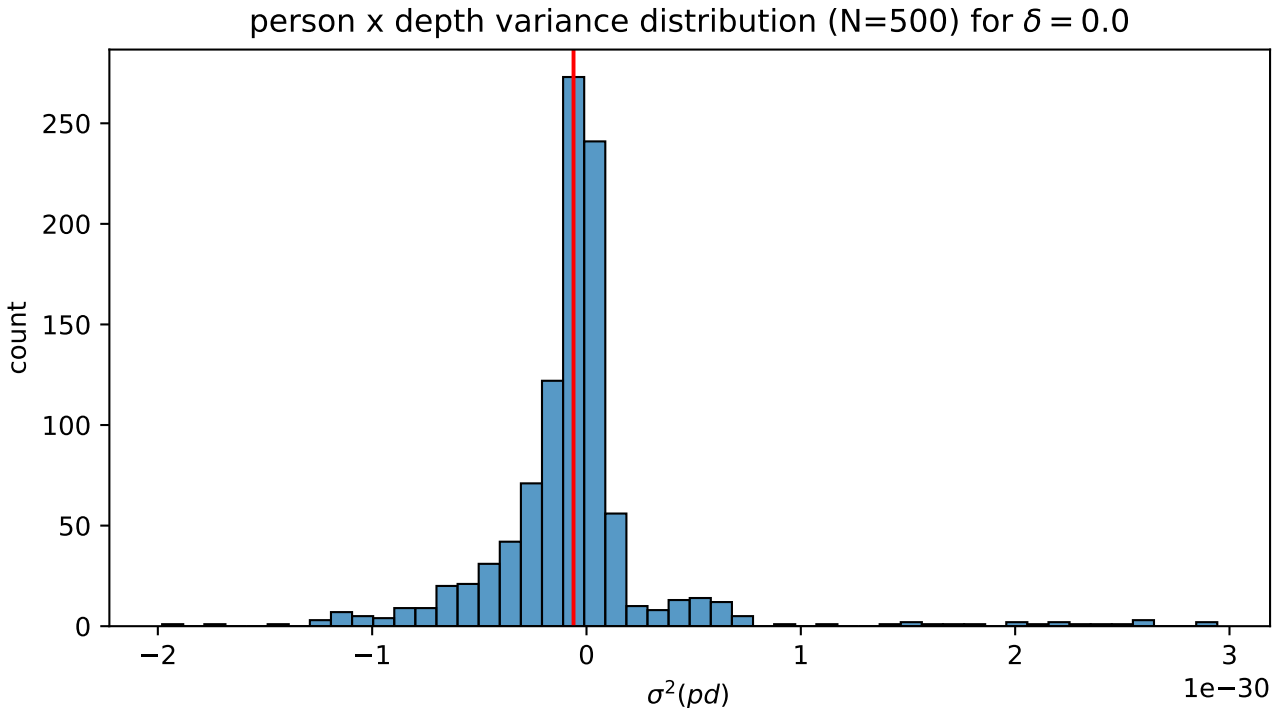

population error variance distribution (N=250) for $\delta = 0.0$
count
$\sigma^2(\eta)$

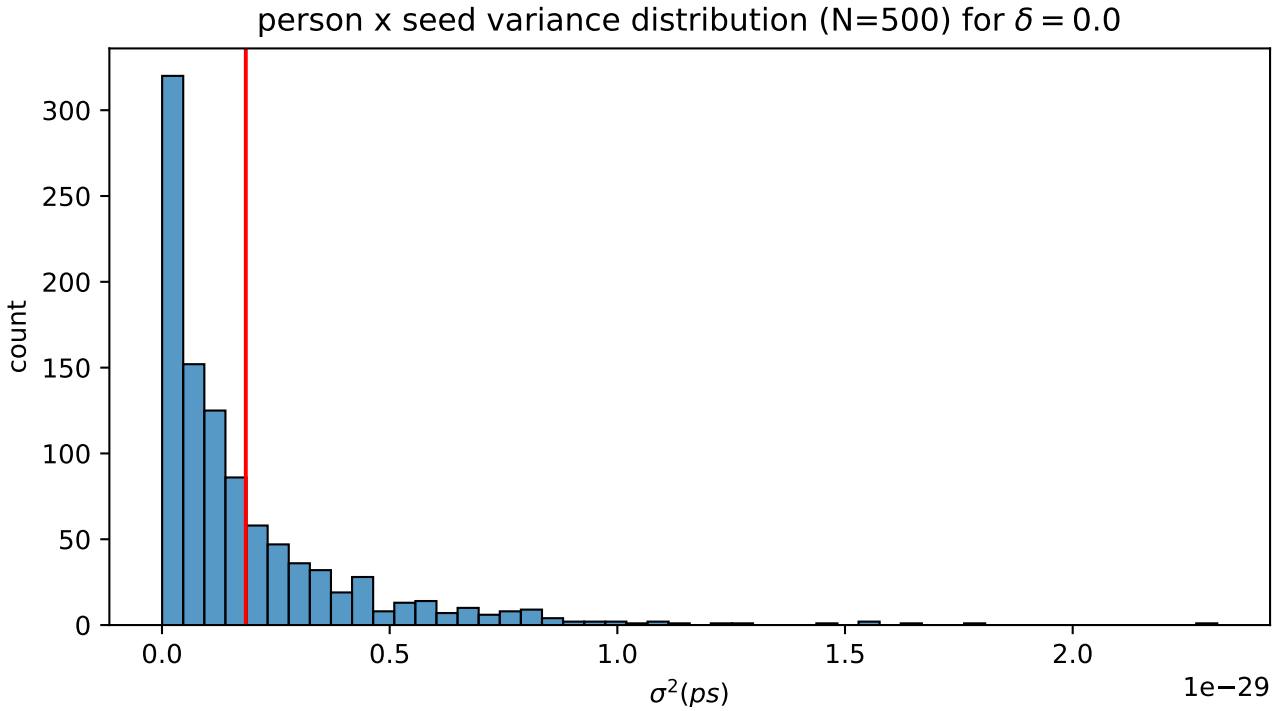

prediction error distribution (N=250) for $\delta = 0.0$
count
ε

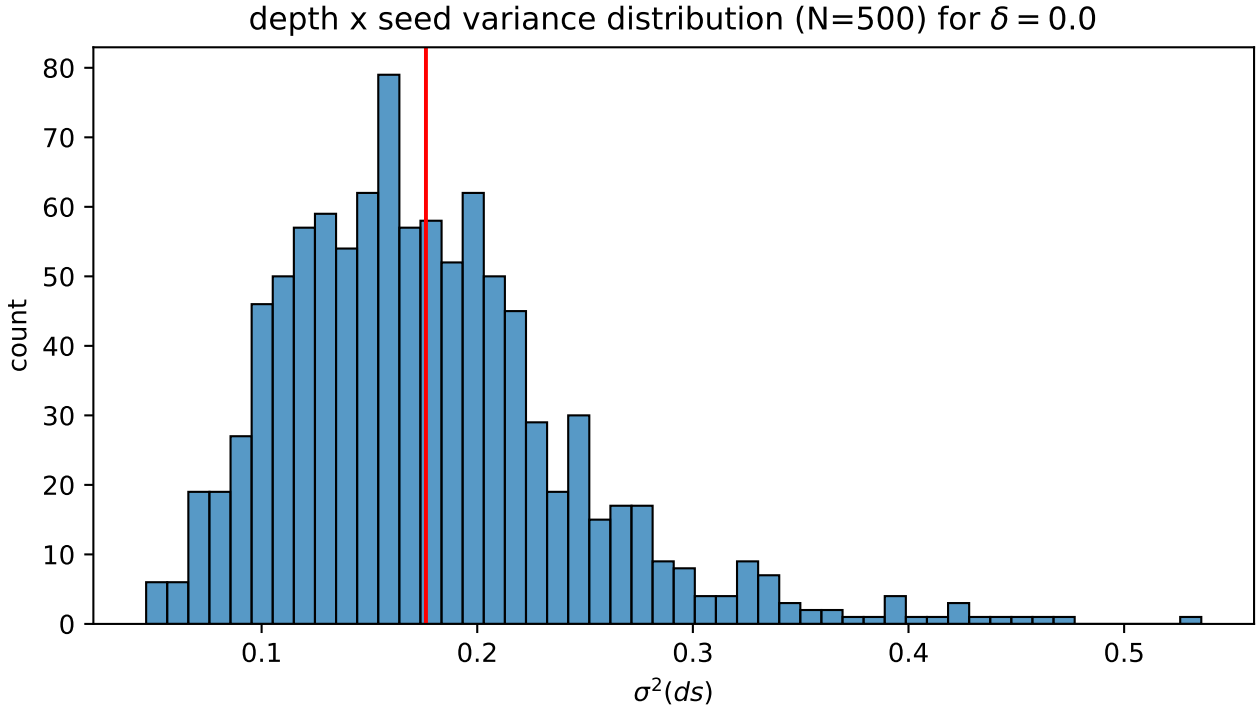

person variance distribution (N=500) for $\delta = 0.0$
count
$\sigma^2(p)$
1e−30

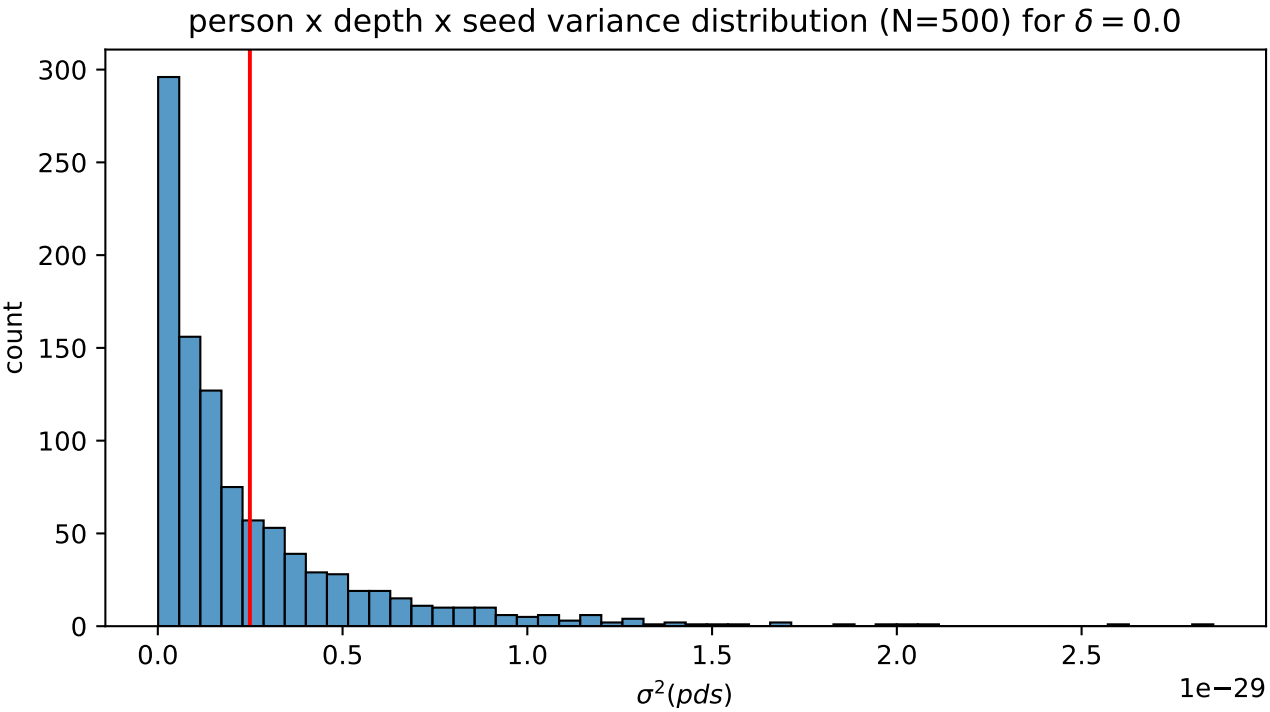

depth variance distribution (N=500) for $\delta = 0.0$
count
$\sigma^2(d)$

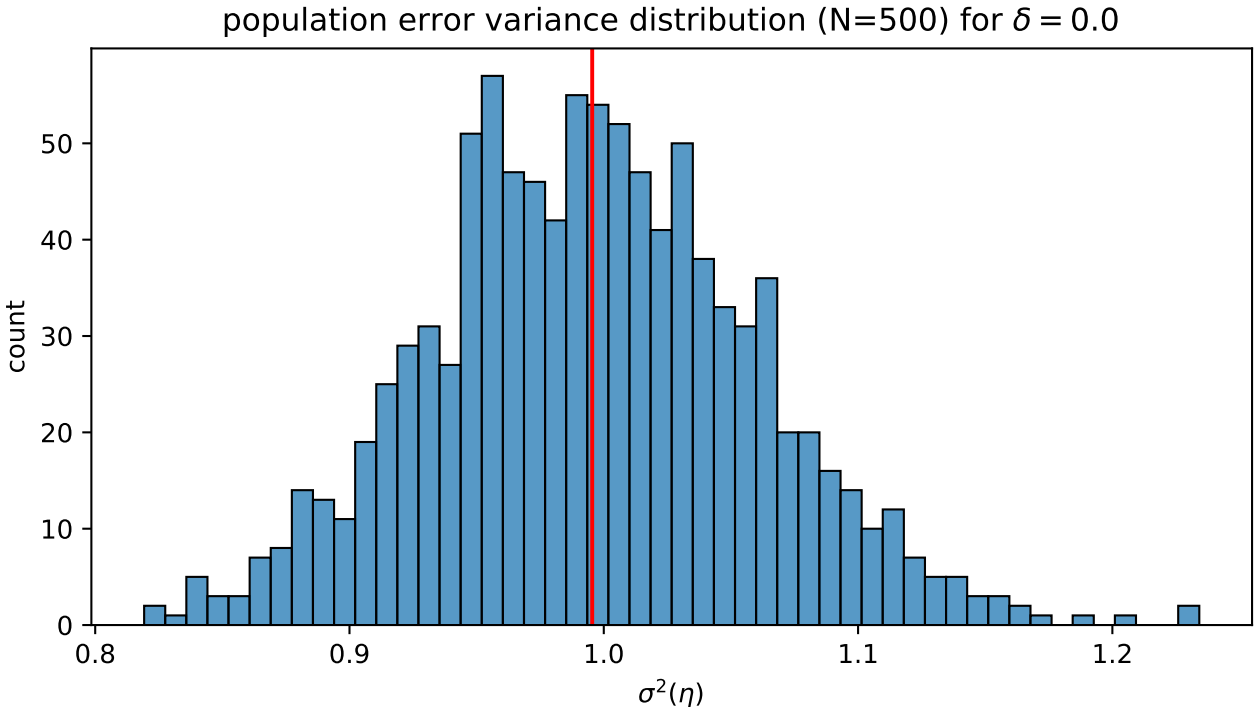

seed variance distribution (N=500) for $\delta = 0.0$
count
$\sigma^2(s)$

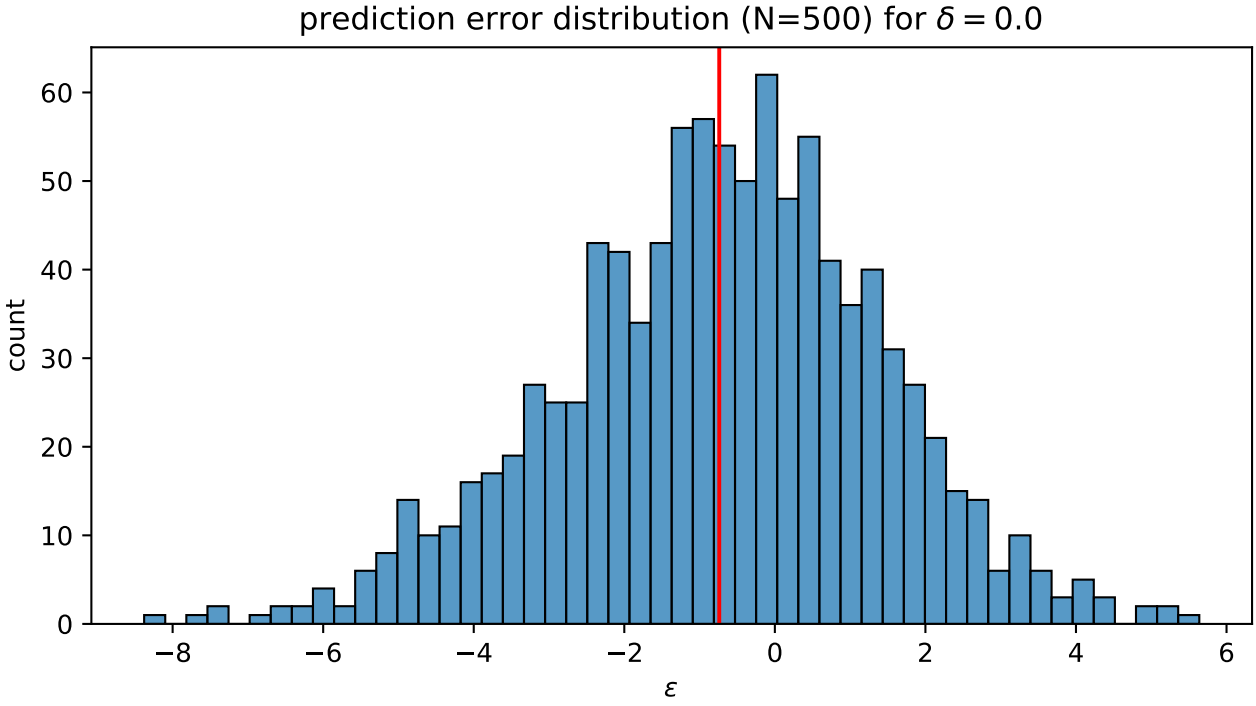

person x depth variance distribution (N=500) for $\delta = 0.0$
count
$\sigma^2(pd)$
1e−30

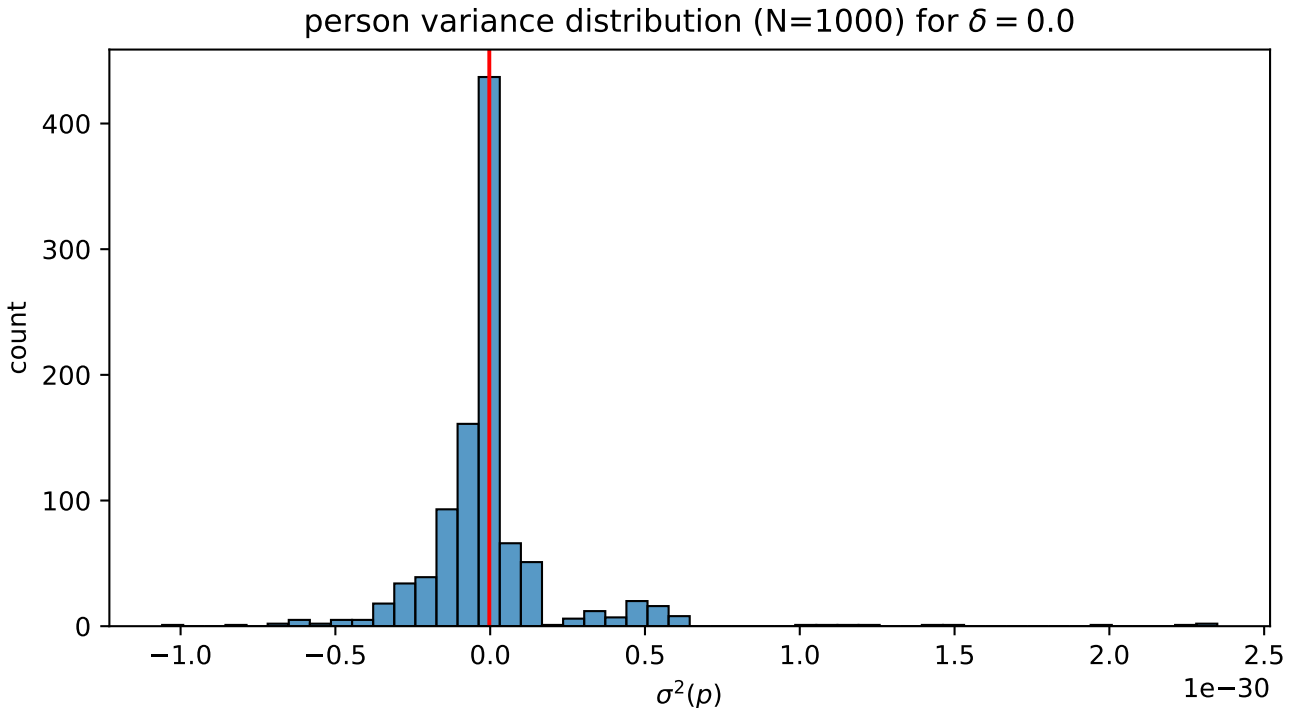

person x seed variance distribution (N=500) for $\delta = 0.0$
count
$\sigma^2(ps)$
1e−29

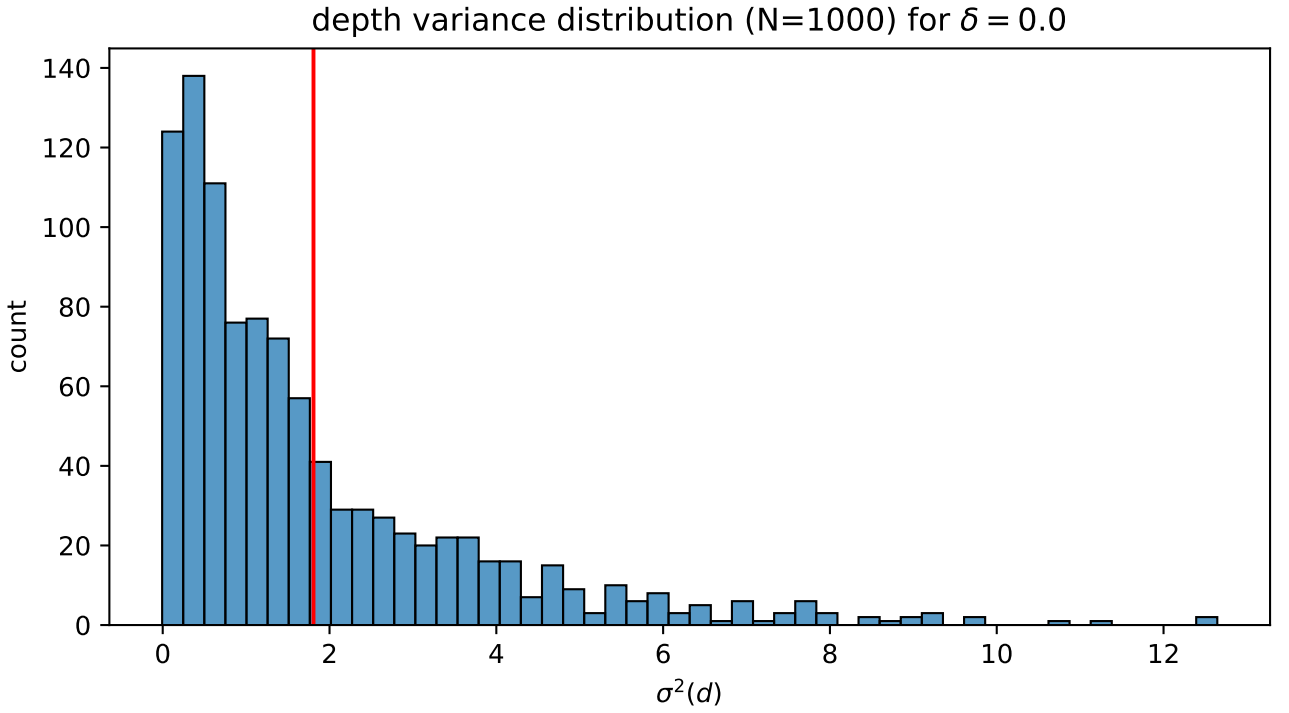

depth x seed variance distribution (N=500) for $\delta = 0.0$
count
$\sigma^2(ds)$

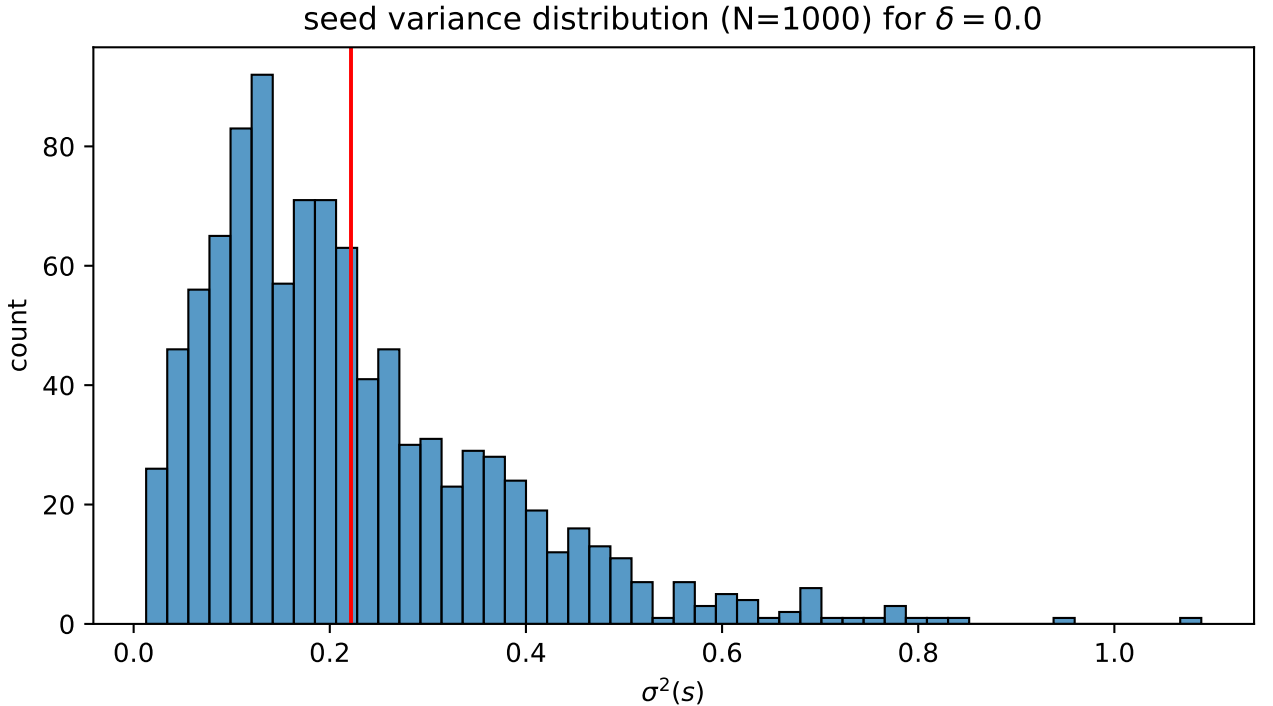

person x depth x seed variance distribution (N=500) for $\delta = 0.0$
count
$\sigma^2(pds)$
1e−29

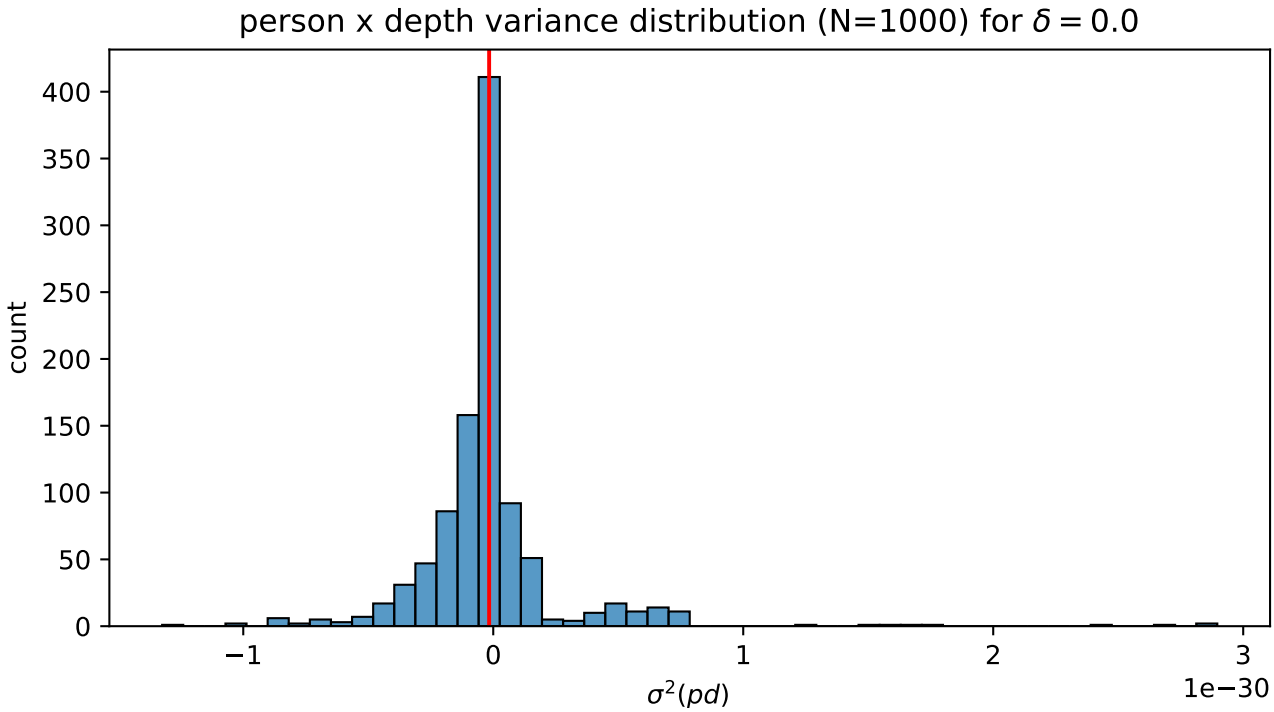

population error variance distribution (N=500) for $\delta = 0.0$
count
$\sigma^2(\eta)$

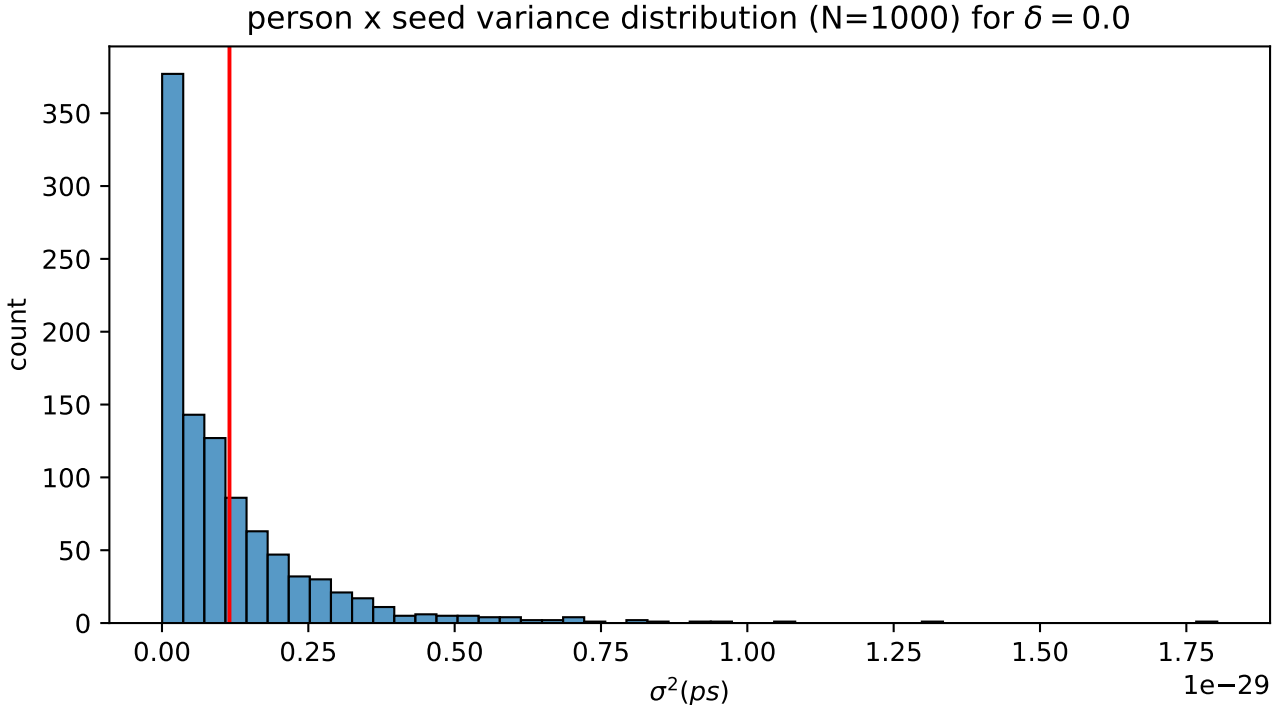

prediction error distribution (N=500) for $\delta = 0.0$
count
ε

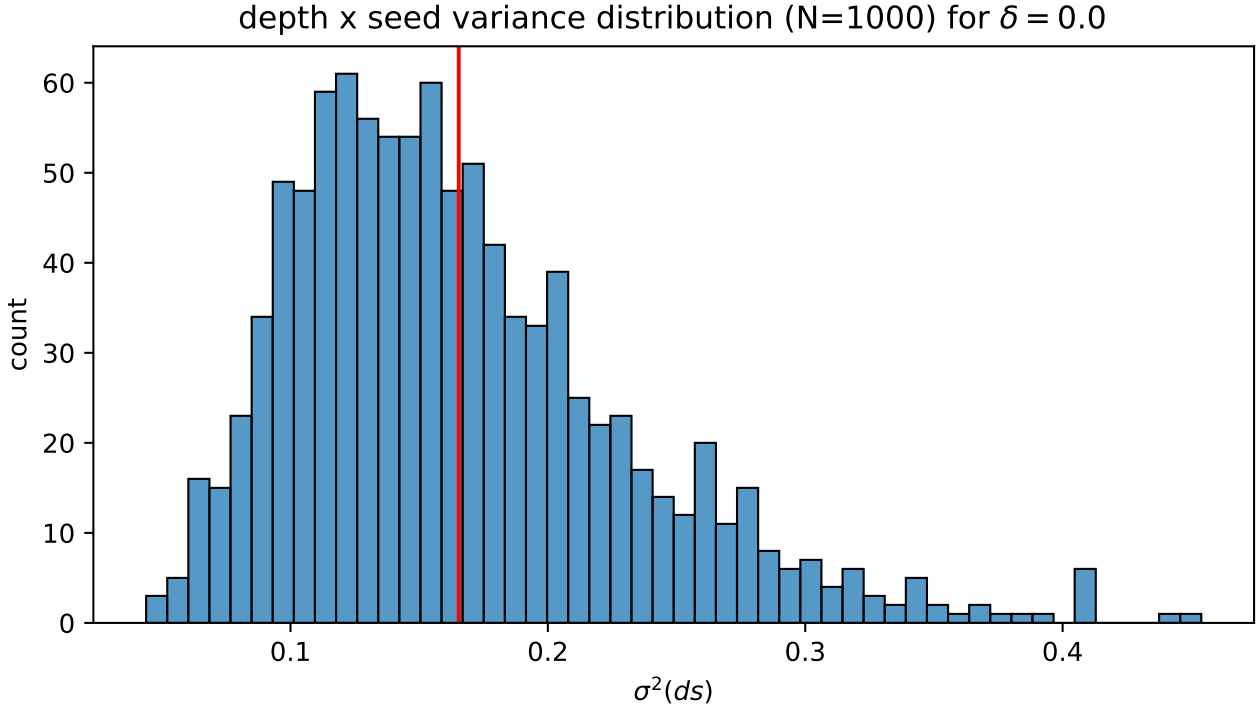

person variance distribution (N=1000) for $\delta = 0.0$
count
$\sigma^2(p)$
1e−30

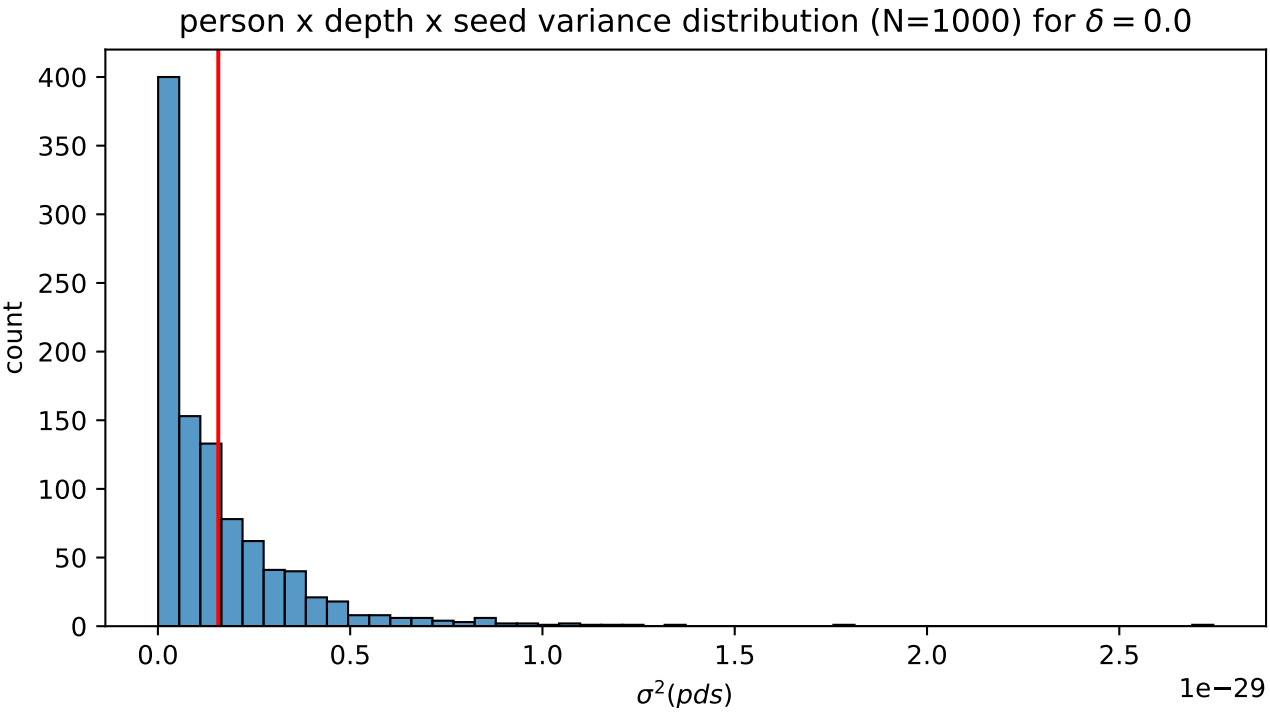

depth variance distribution (N=1000) for $\delta = 0.0$
count
$\sigma^2(d)$

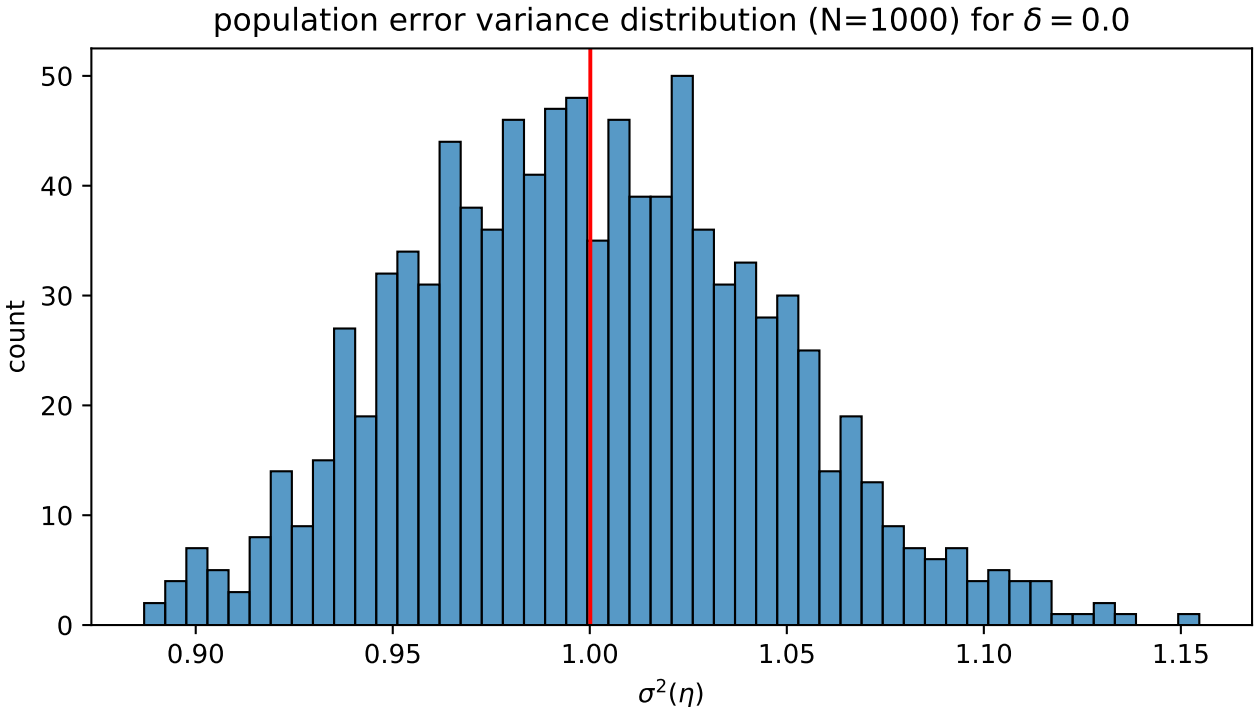

seed variance distribution (N=1000) for $\delta = 0.0$
count
$\sigma^2(s)$

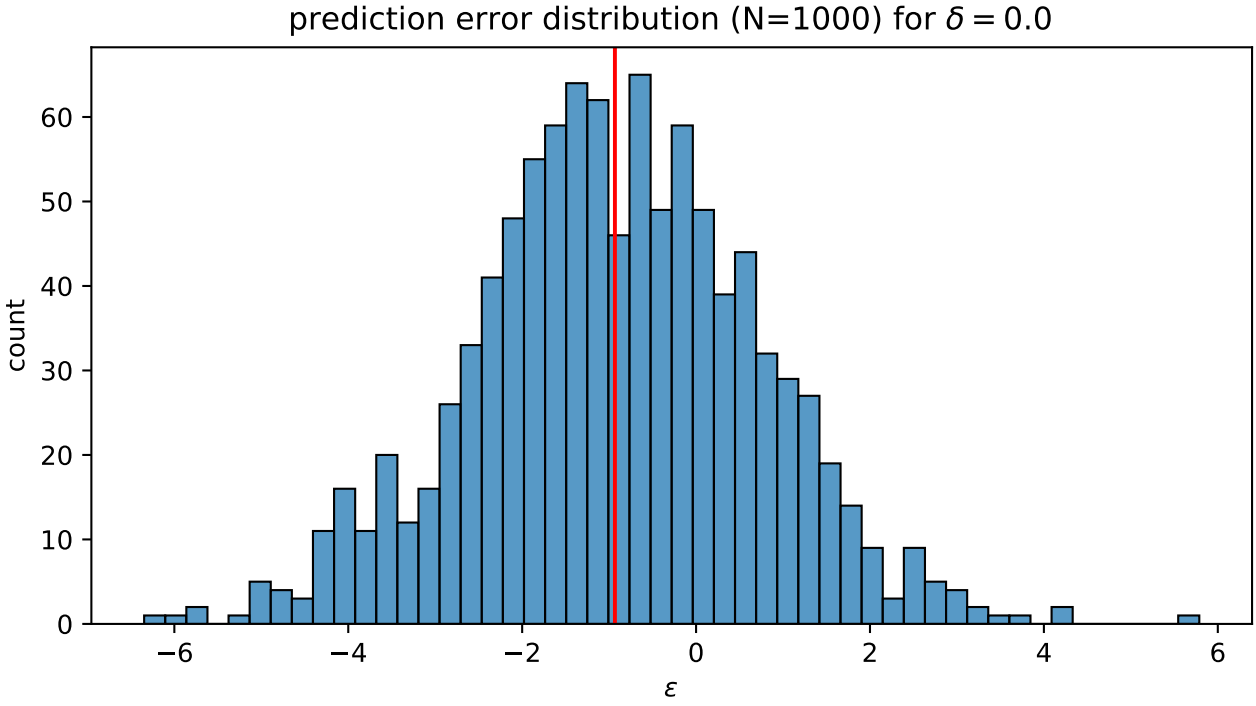

person x depth variance distribution (N=1000) for $\delta = 0.0$
count
$\sigma^2(pd)$
1e−30

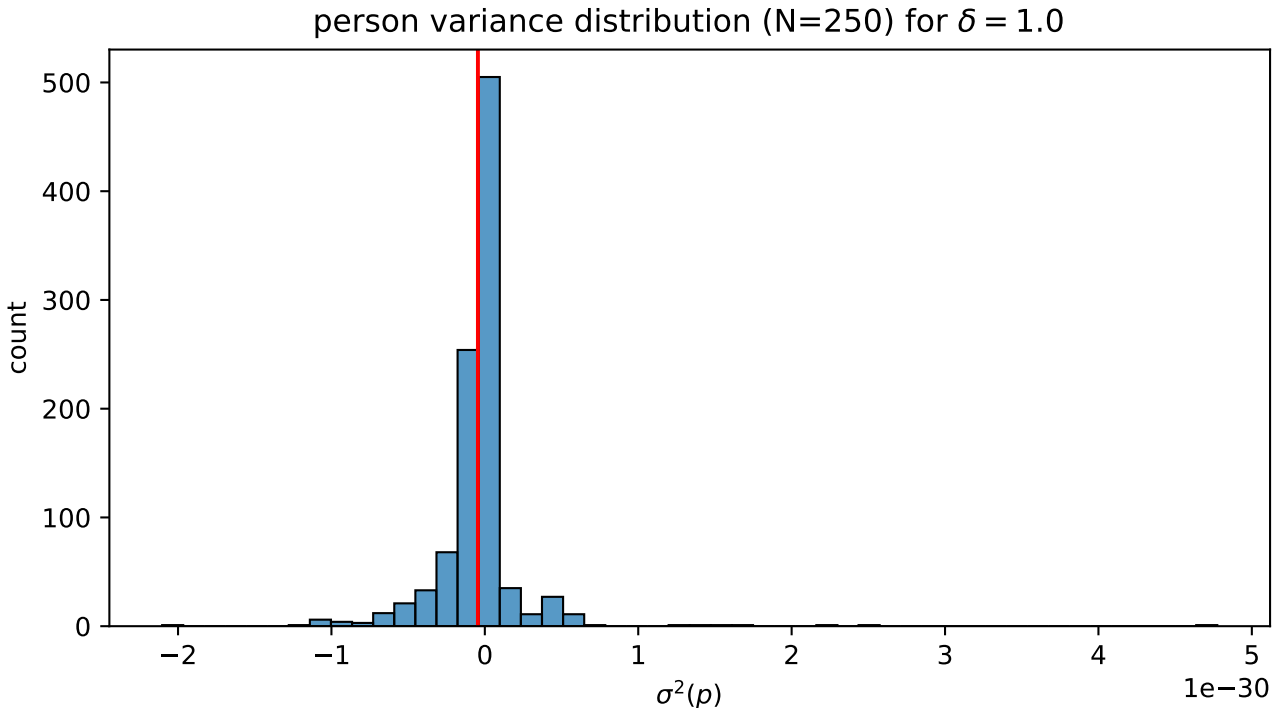

person x seed variance distribution (N=1000) for $\delta = 0.0$
count
$\sigma^2(ps)$
1e−29

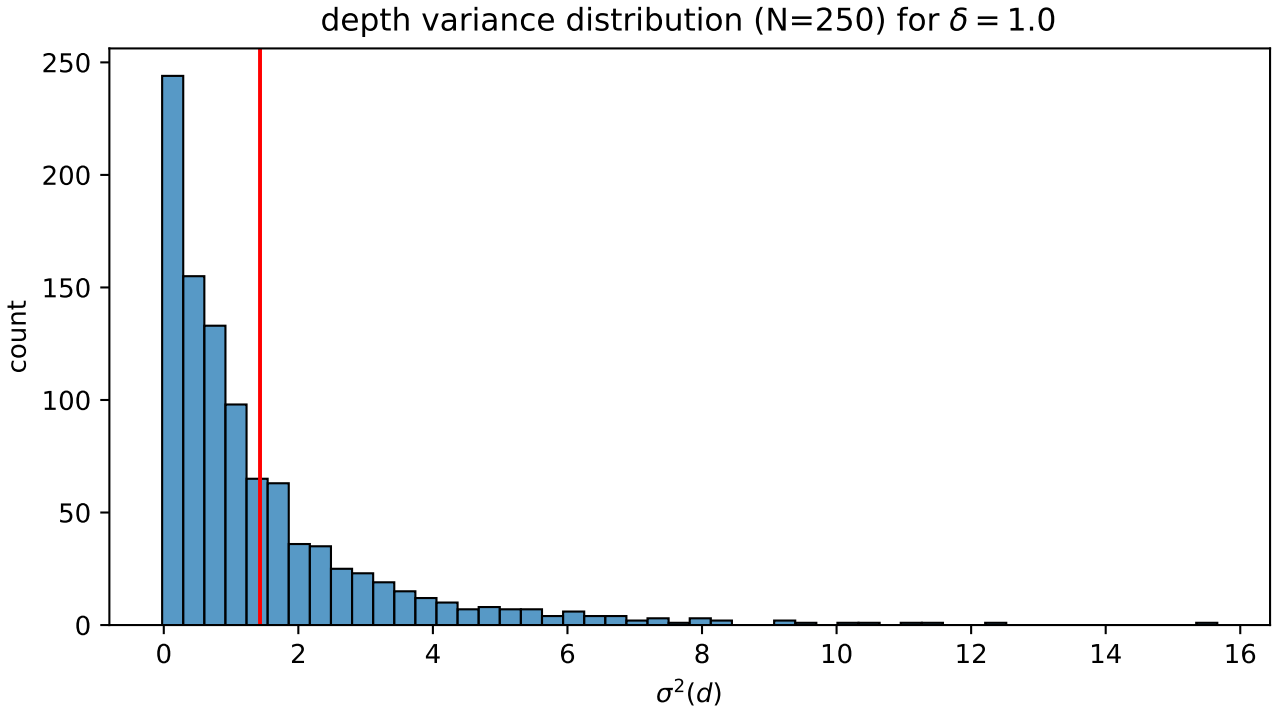

depth x seed variance distribution (N=1000) for $\delta = 0.0$
count
$\sigma^2(ds)$

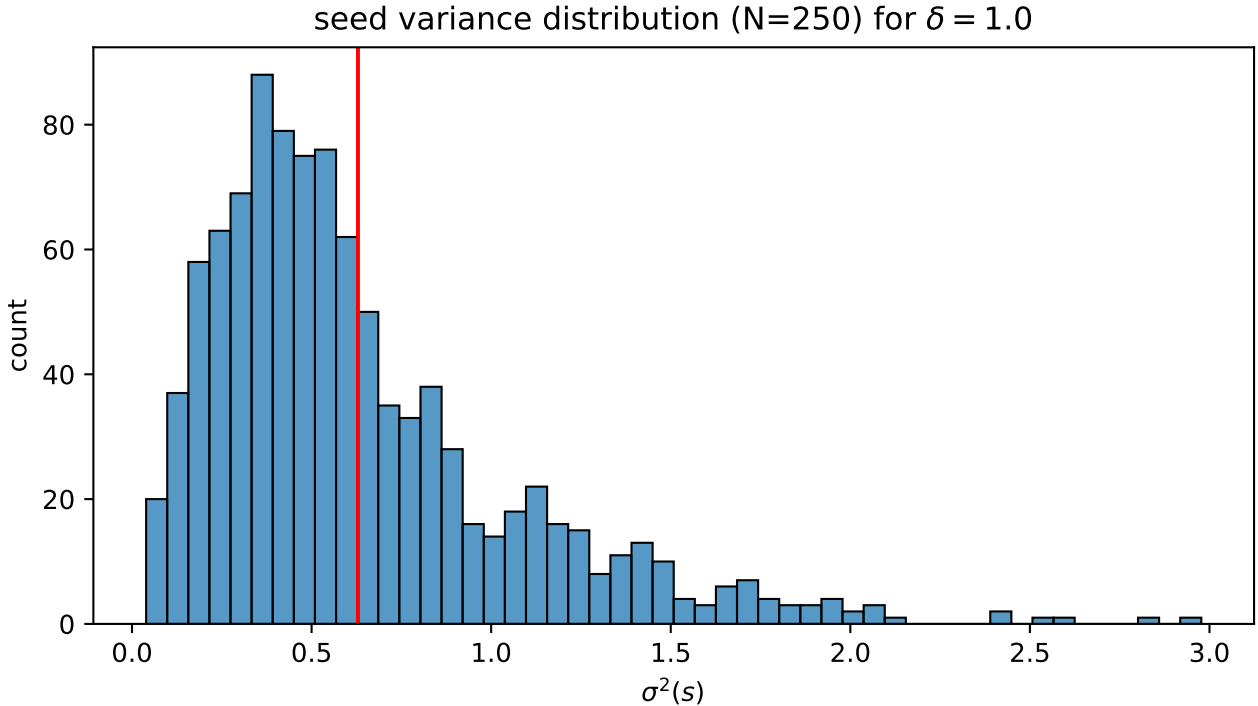

person x depth x seed variance distribution (N=1000) for $\delta = 0.0$
count
$\sigma^2(pds)$
1e−29

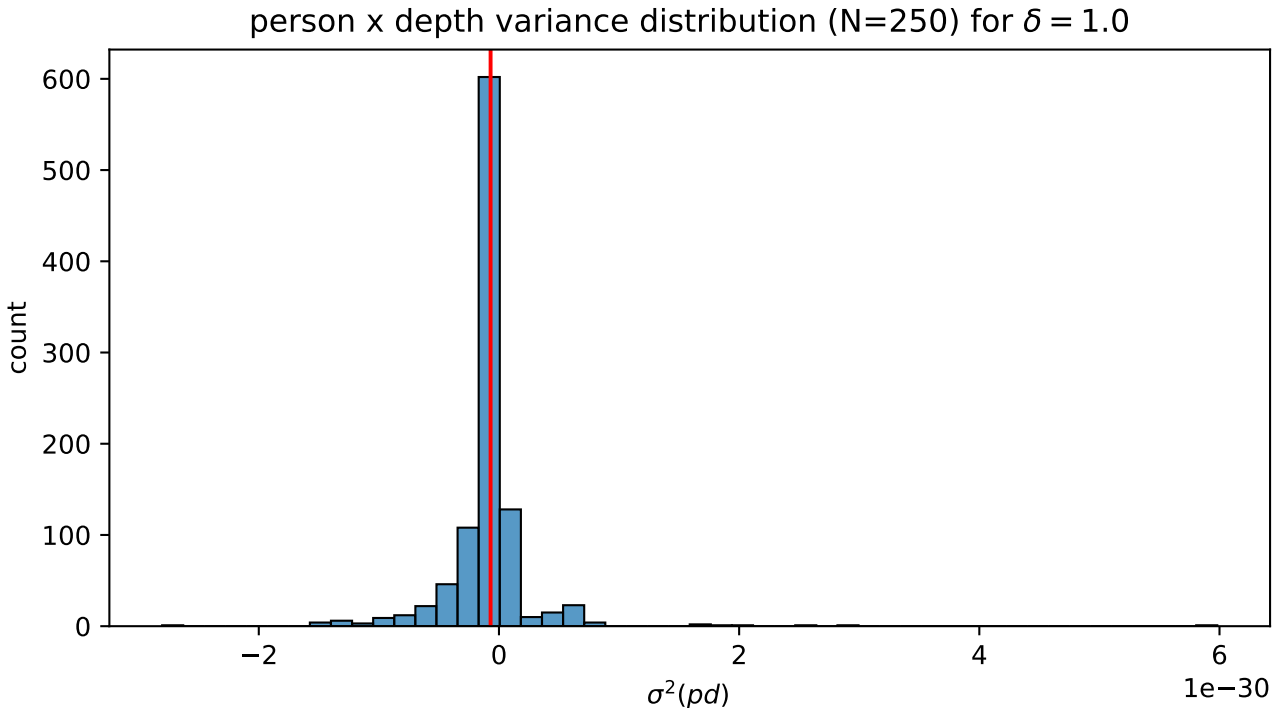

population error variance distribution (N=1000) for $\delta = 0.0$
count
$\sigma^2(\eta)$

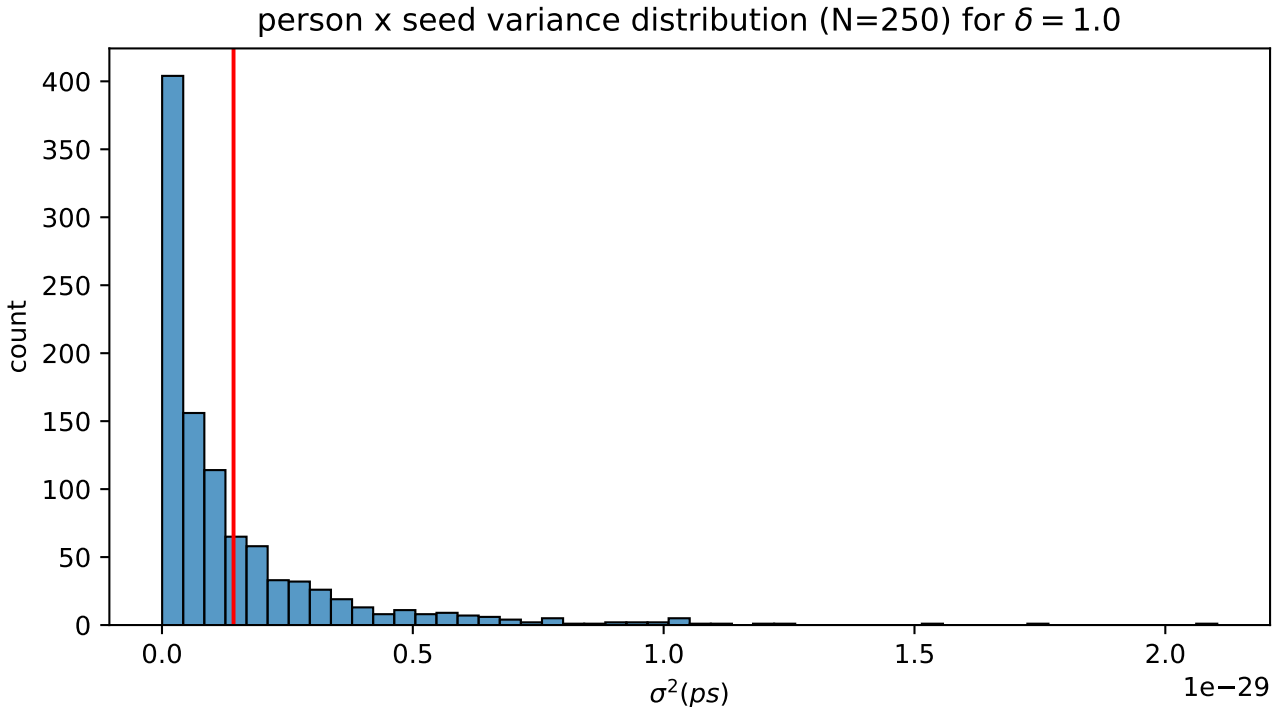

prediction error distribution (N=1000) for $\delta = 0.0$
count
ε

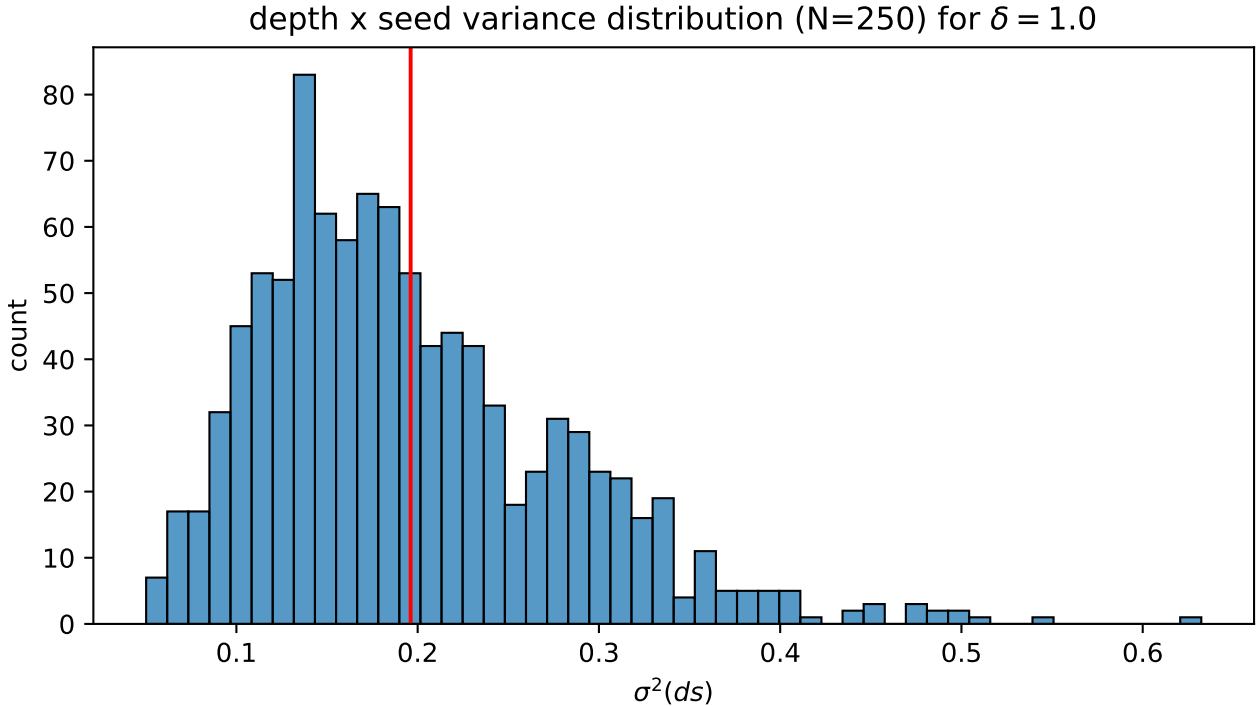

person variance distribution (N=250) for $\delta = 1.0$
count
$\sigma^2(p)$
1e−30

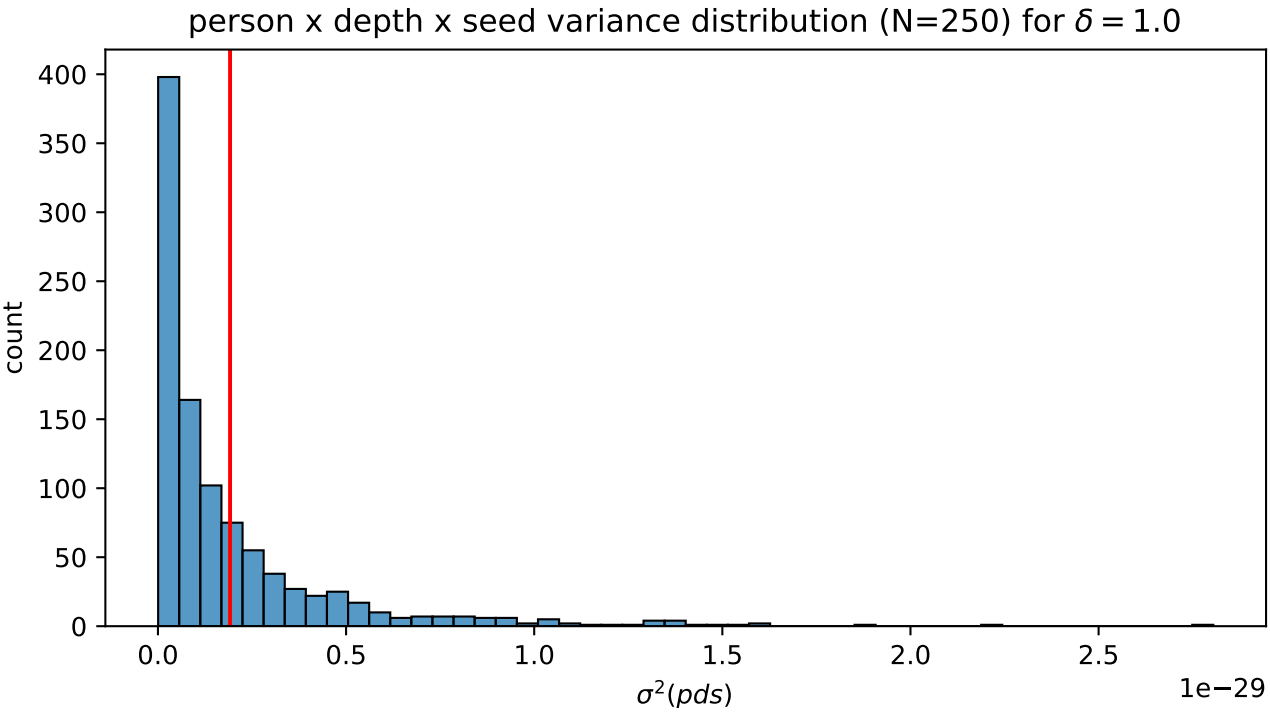

depth variance distribution (N=250) for $\delta = 1.0$
count
$\sigma^2(d)$

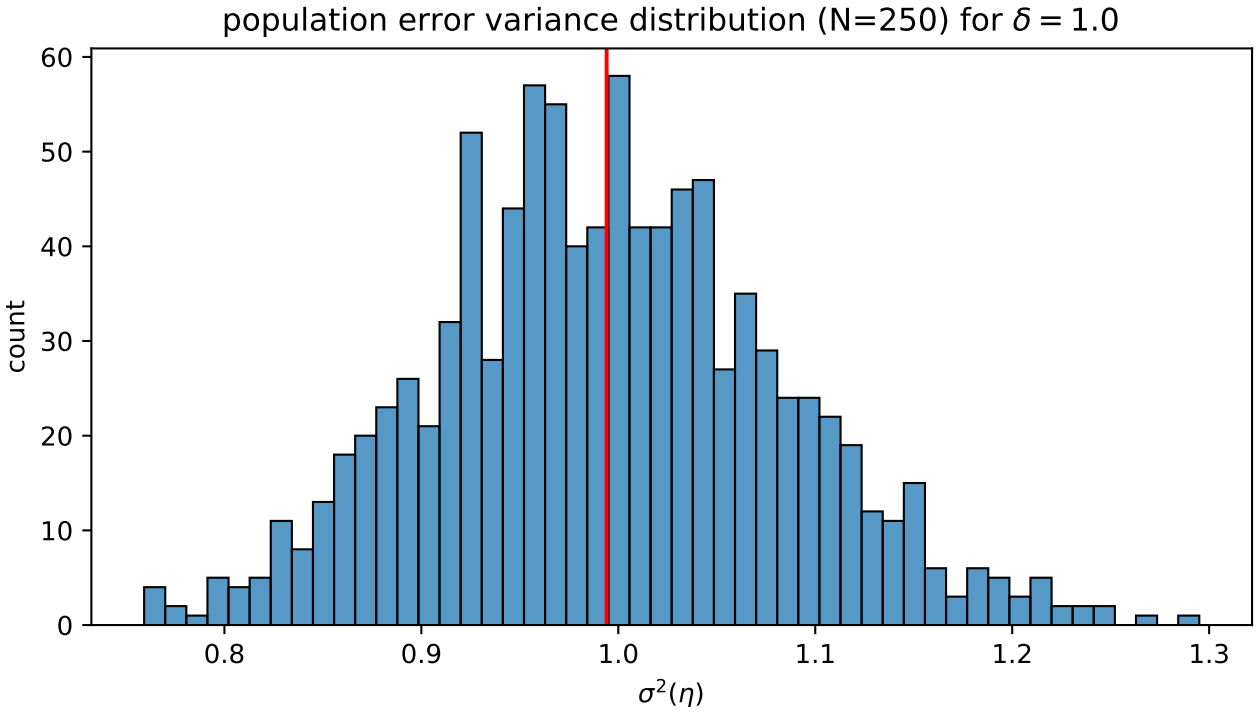

seed variance distribution (N=250) for $\delta = 1.0$
count
$\sigma^2(s)$

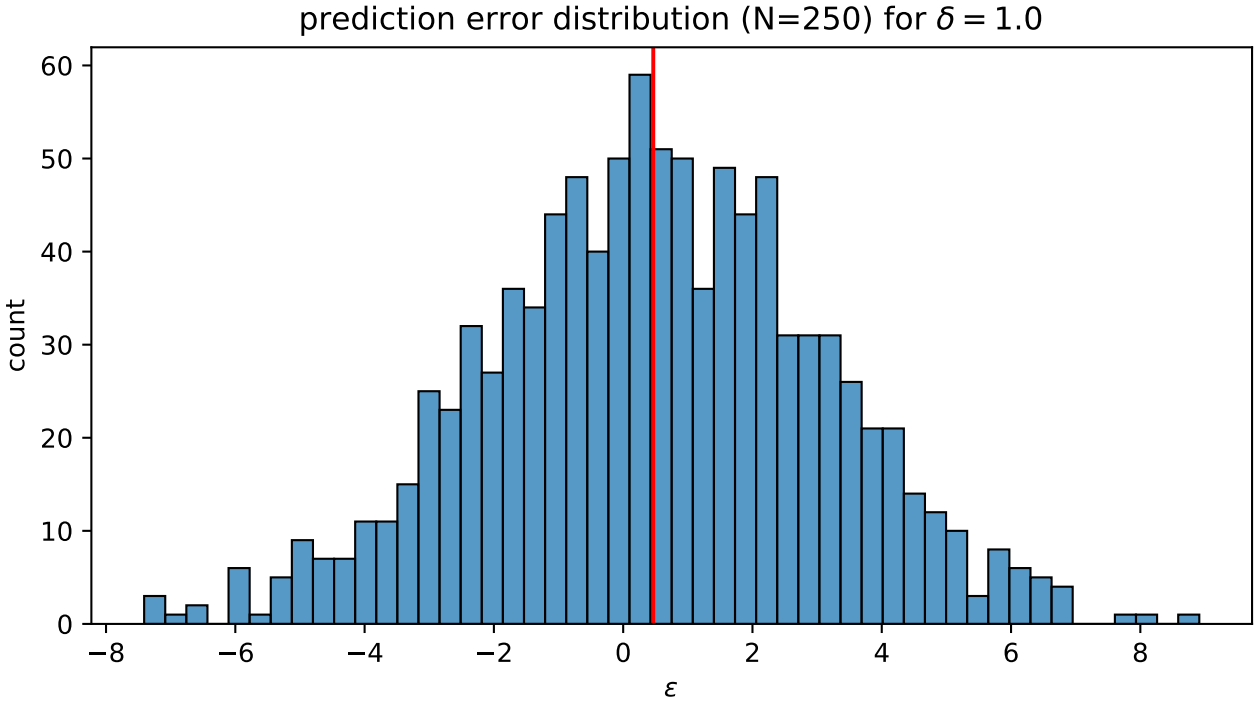

person x depth variance distribution (N=250) for $\delta = 1.0$
count
$\sigma^2(pd)$
1e−30

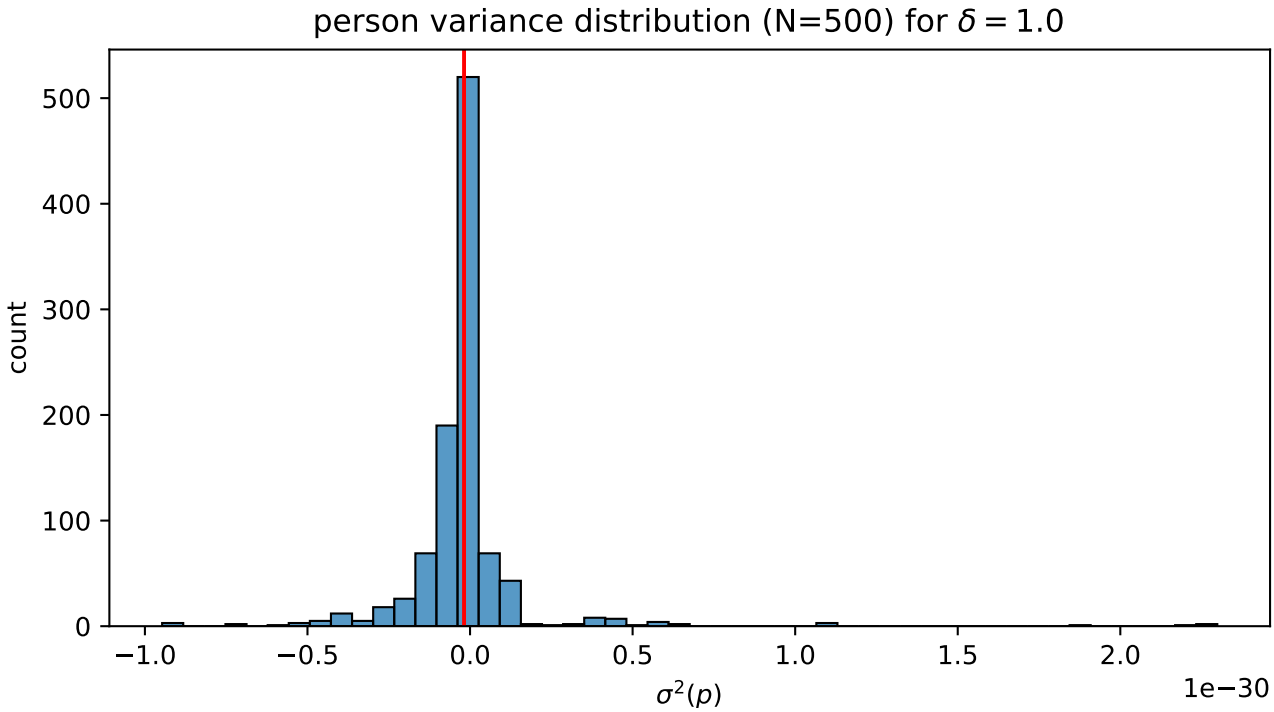

person x seed variance distribution (N=250) for $\delta = 1.0$
count
$\sigma^2(ps)$
1e−29

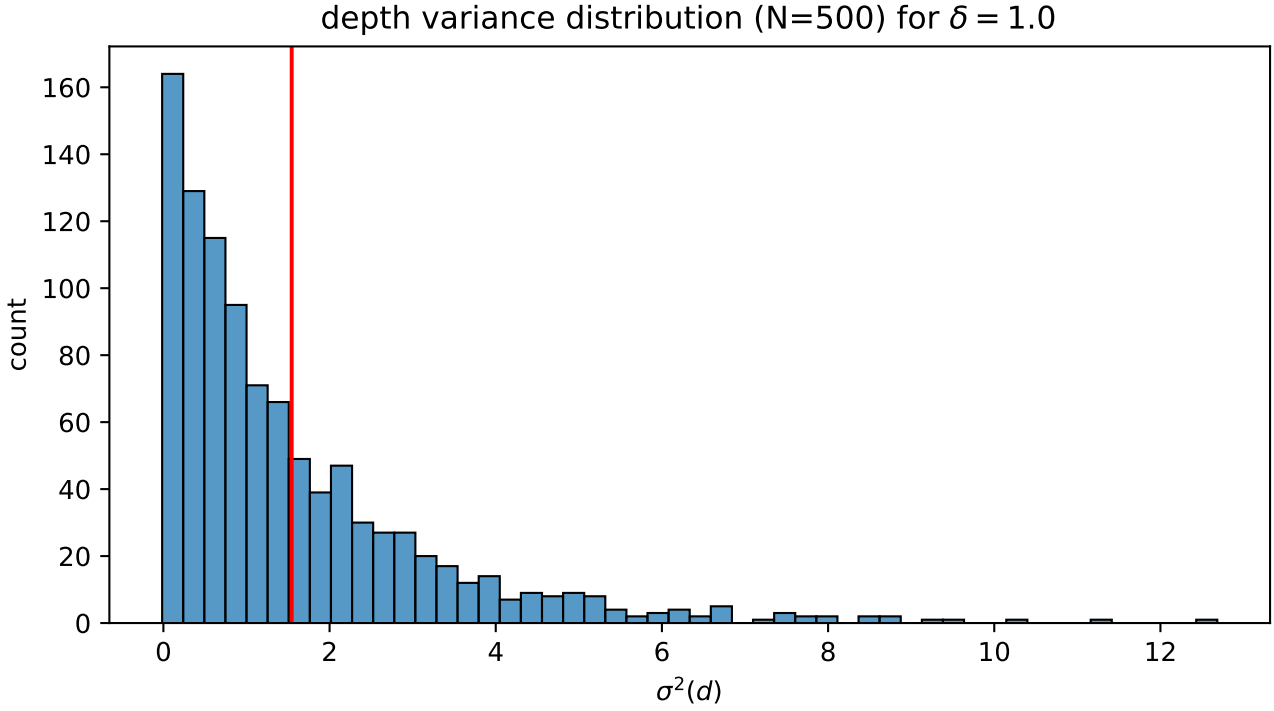

depth x seed variance distribution (N=250) for $\delta = 1.0$
count
$\sigma^2(ds)$

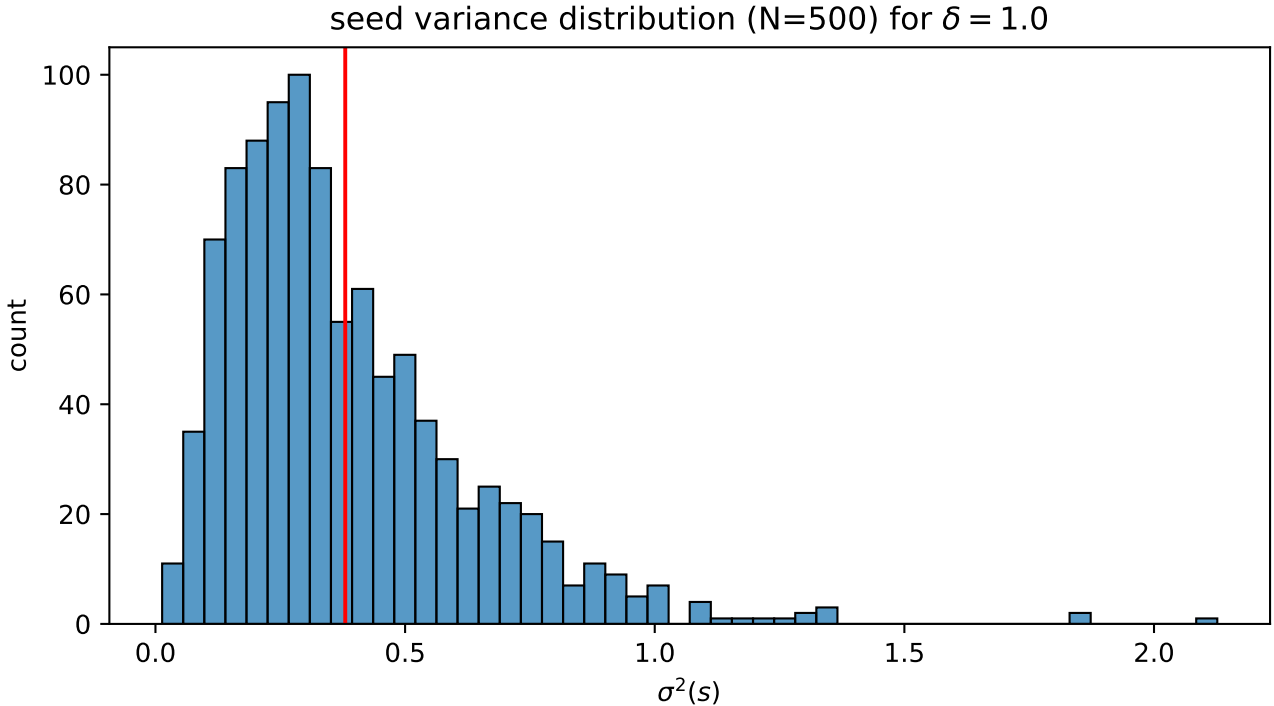

person x depth x seed variance distribution (N=250) for $\delta = 1.0$
count
$\sigma^2(pds)$
1e−29

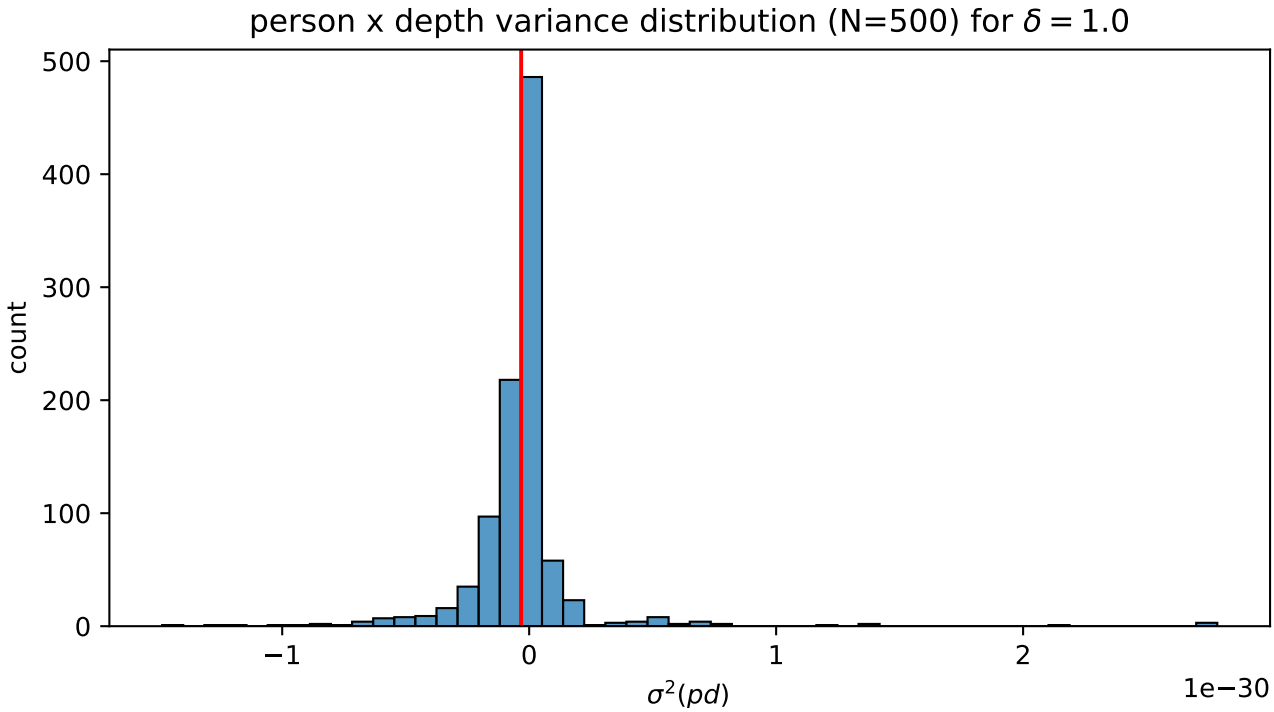

population error variance distribution (N=250) for $\delta = 1.0$
count
$\sigma^2(\eta)$

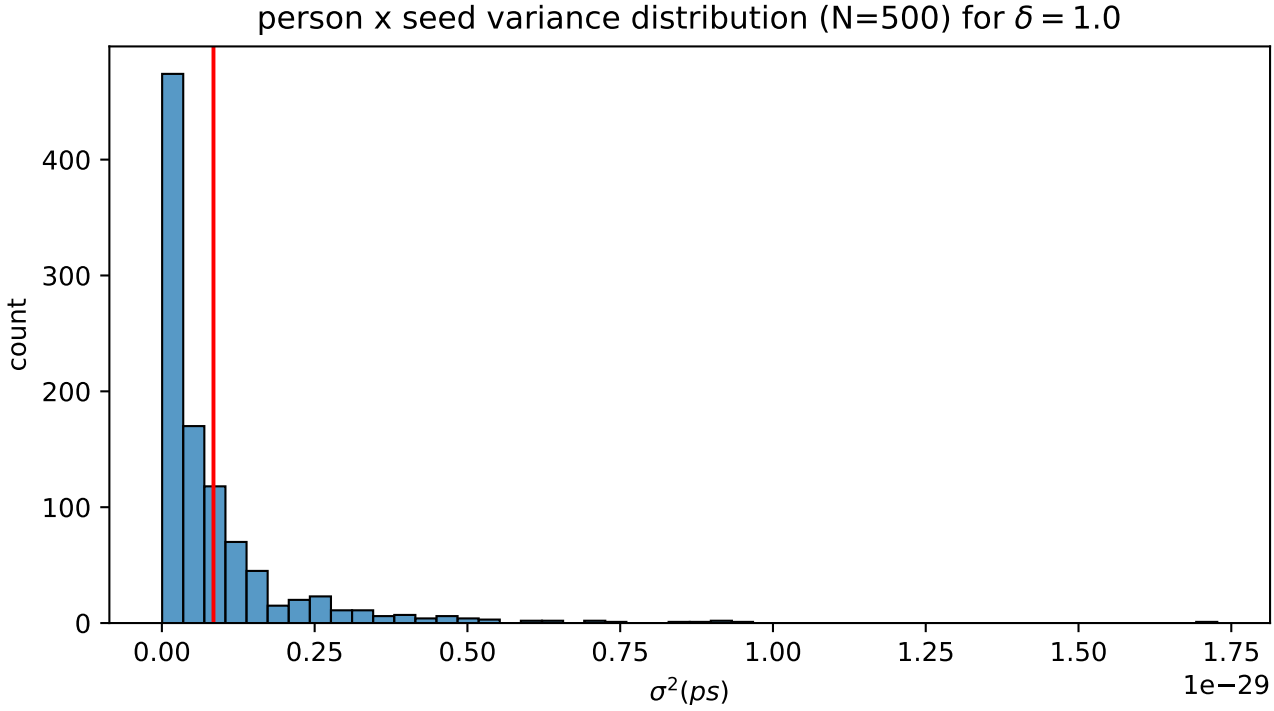

prediction error distribution (N=250) for $\delta = 1.0$
count
ε

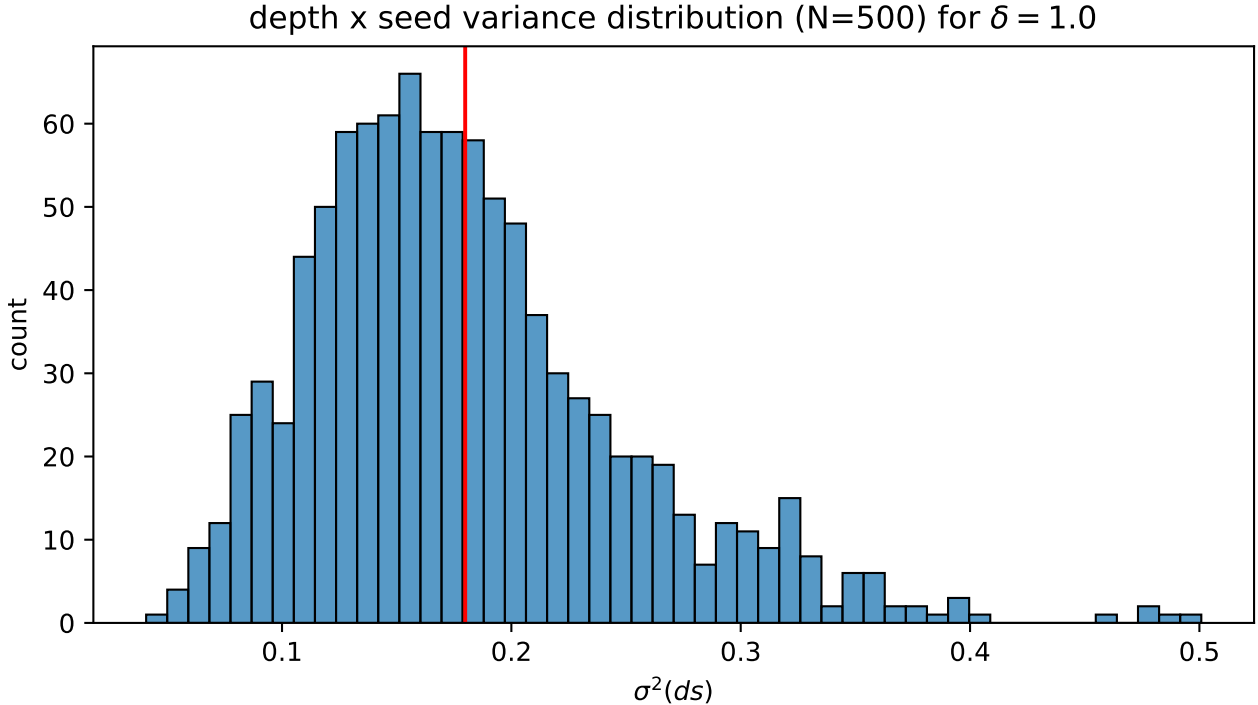

person variance distribution (N=500) for $\delta = 1.0$
count
$\sigma^2(p)$
1e−30

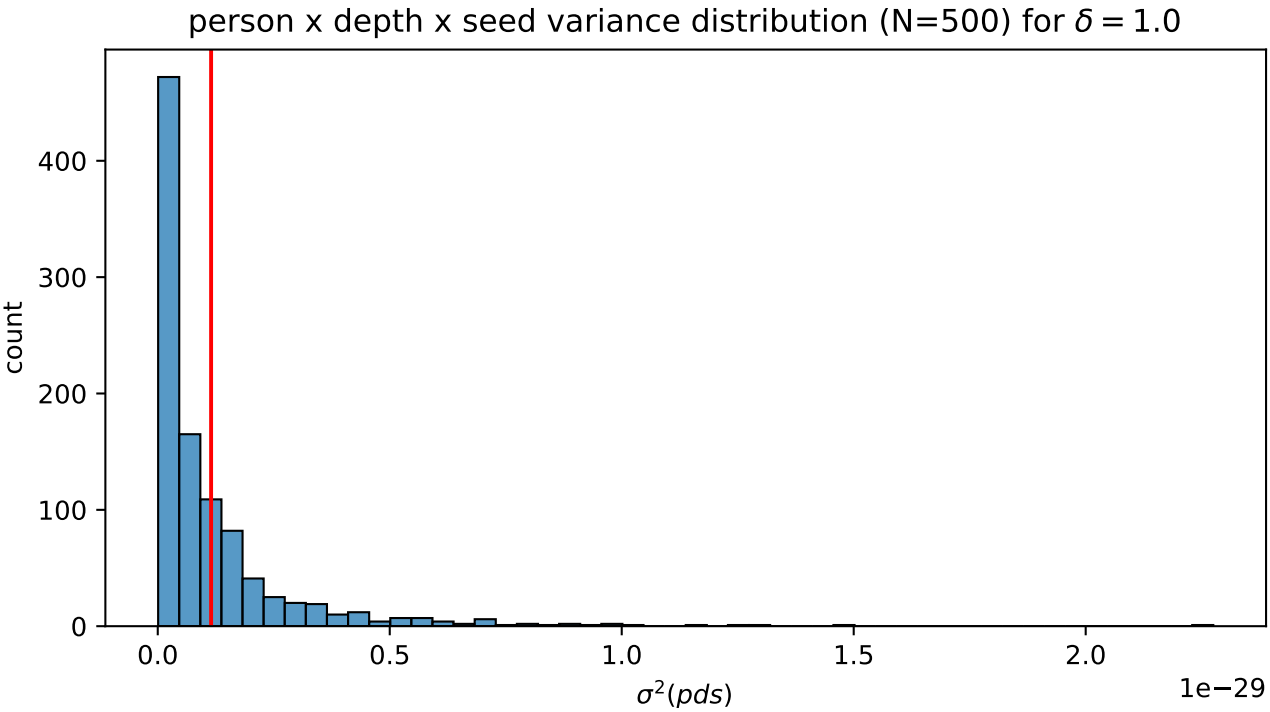

depth variance distribution (N=500) for $\delta = 1.0$
count
$\sigma^2(d)$

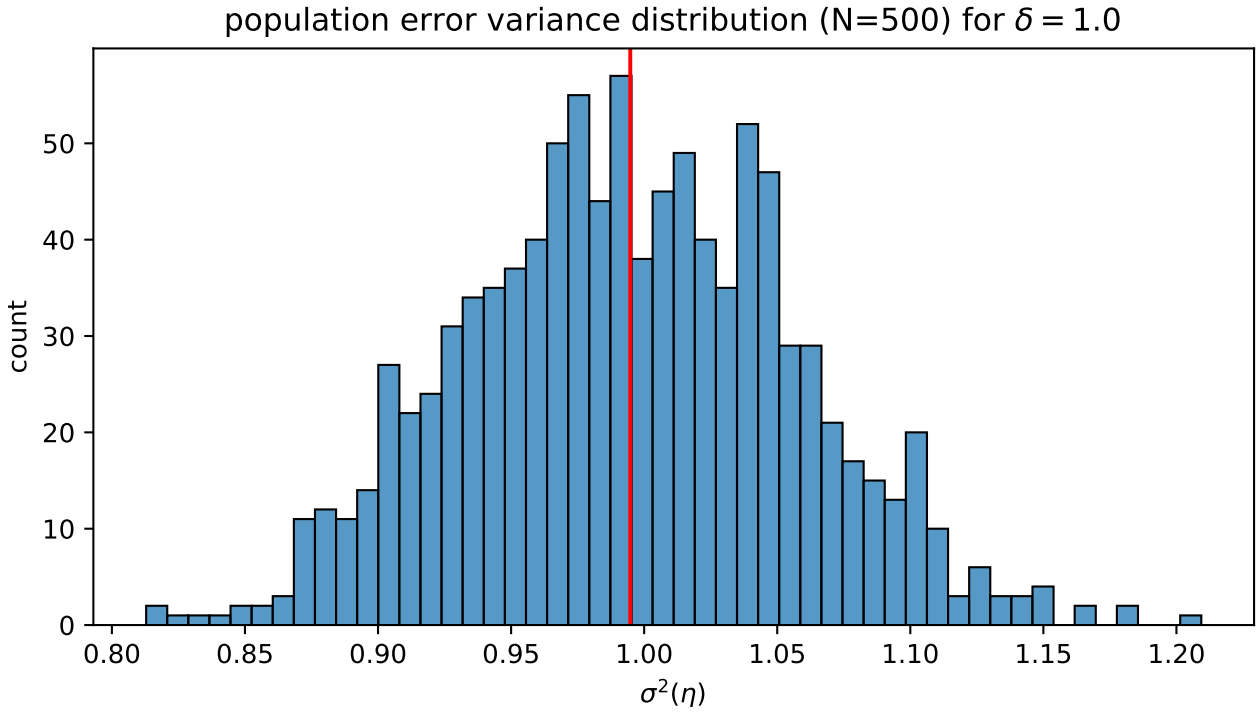

seed variance distribution (N=500) for $\delta = 1.0$
count
$\sigma^2(s)$

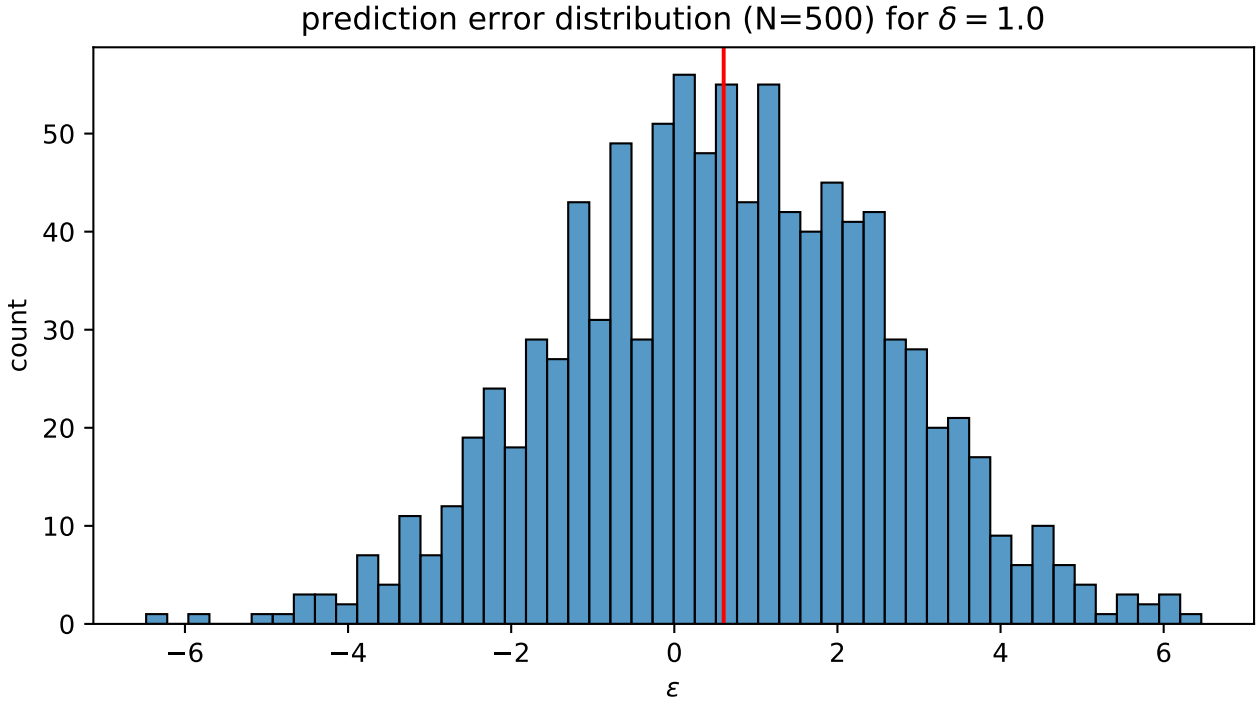

person x depth variance distribution (N=500) for $\delta = 1.0$
count
$\sigma^2(pd)$
1e−30

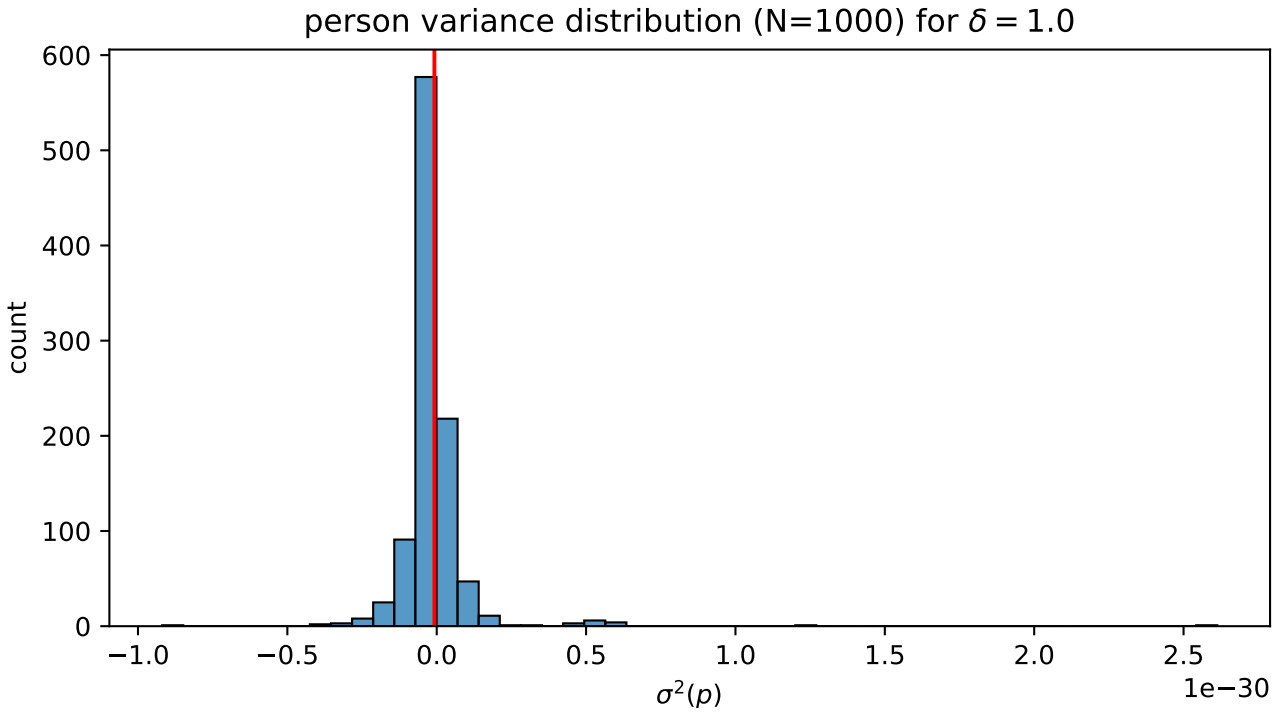

person x seed variance distribution (N=500) for $\delta = 1.0$
count
$\sigma^2(ps)$
1e−29

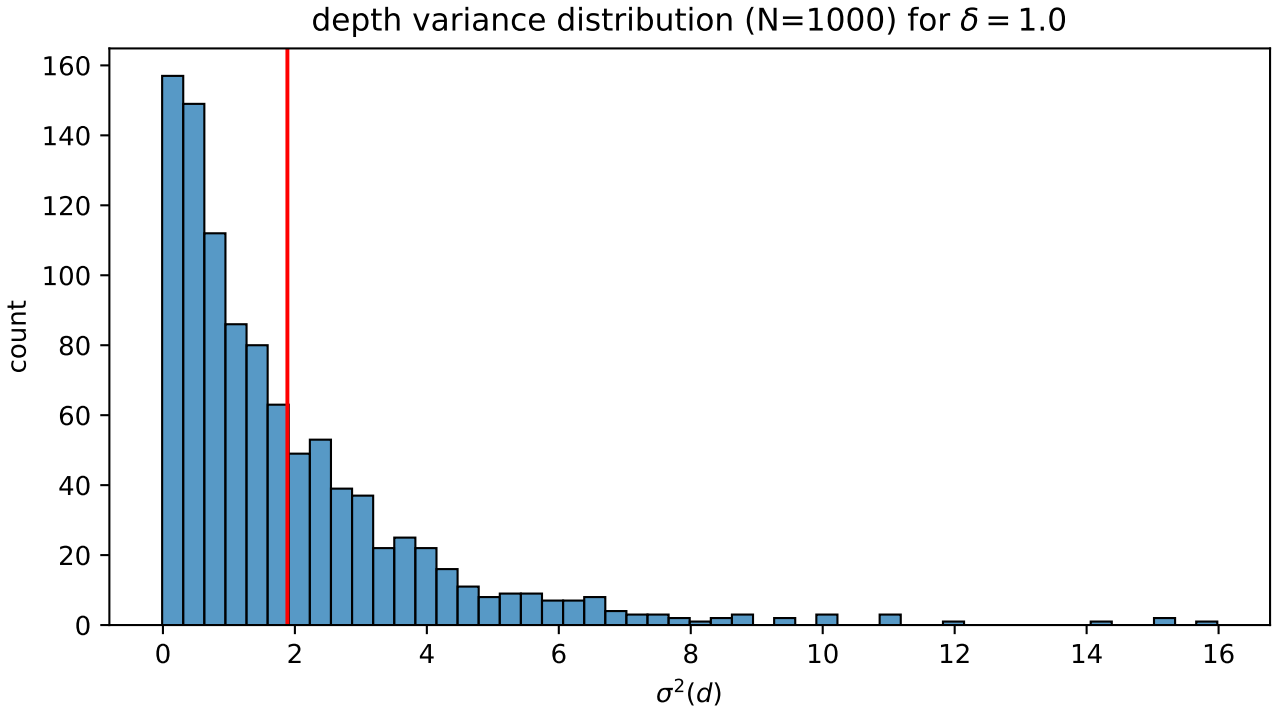

depth x seed variance distribution (N=500) for $\delta = 1.0$
count
$\sigma^2(ds)$

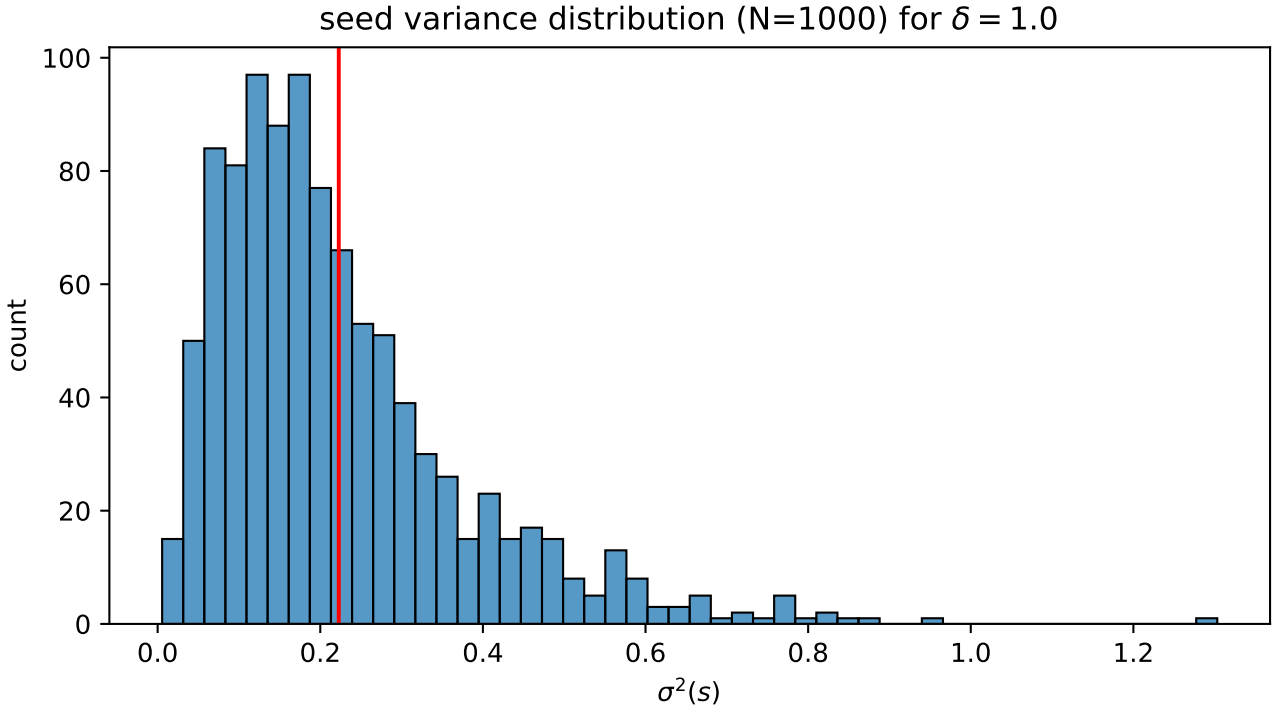

person x depth x seed variance distribution (N=500) for $\delta = 1.0$
count
0
100
200
300
400
0.0
0.5
1.0
1.5
2.0
$\sigma^2(pds)$
1e−29

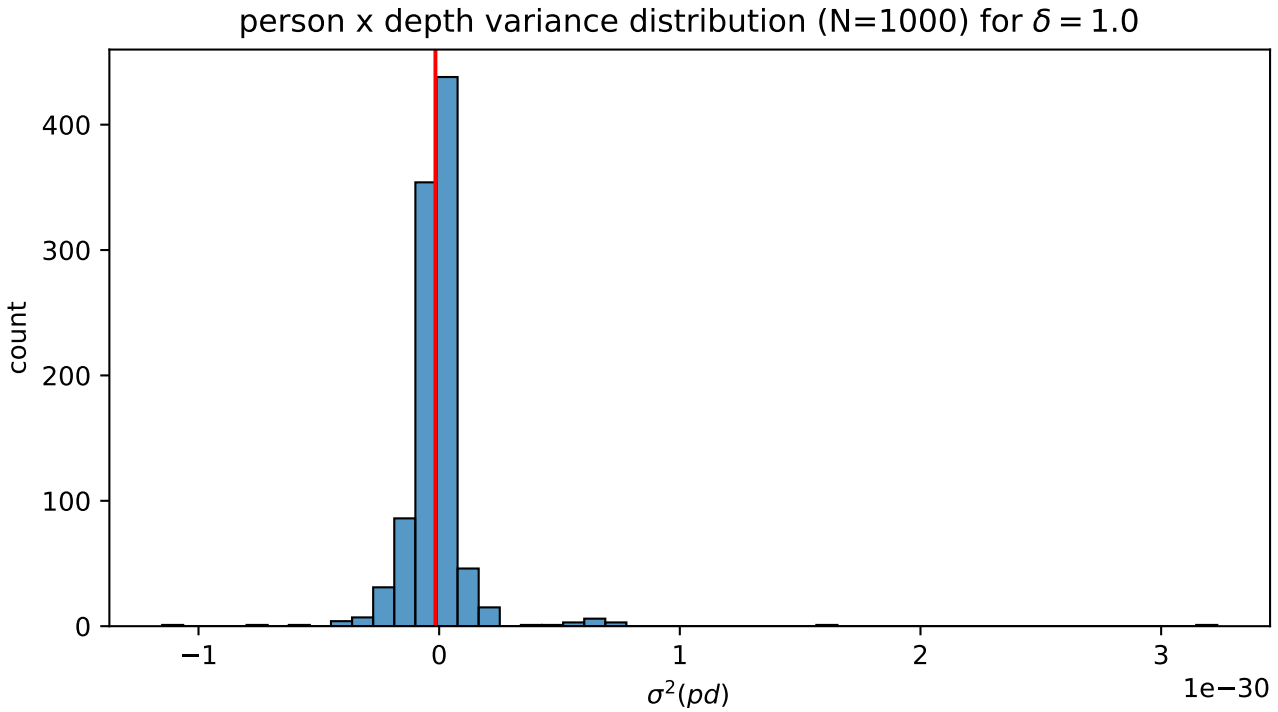

population error variance distribution (N=500) for $\delta = 1.0$
count
0
10
20
30
40
50
0.80
0.85
0.90
0.95
1.00
1.05
1.10
1.15
1.20
$\sigma^2(\eta)$

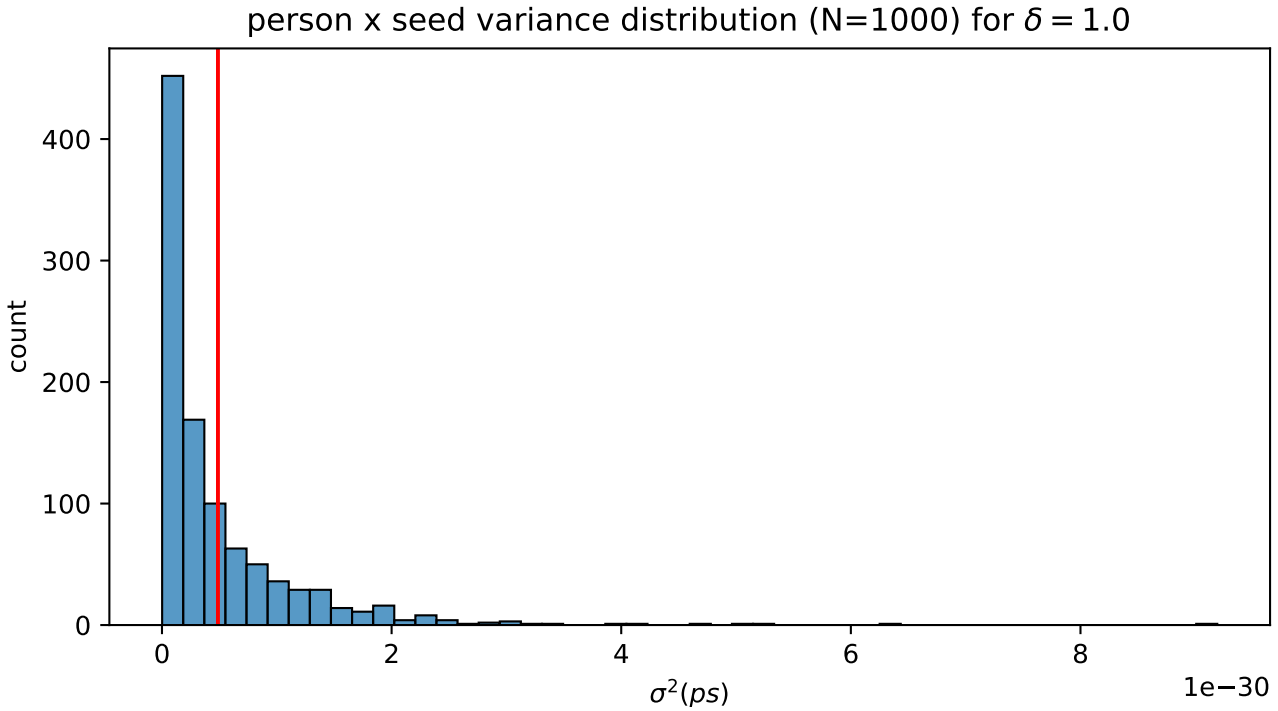

prediction error distribution (N=500) for $\delta = 1.0$
count
0
10
20
30
40
50
−6
−4
−2
0
2
4
6
ε

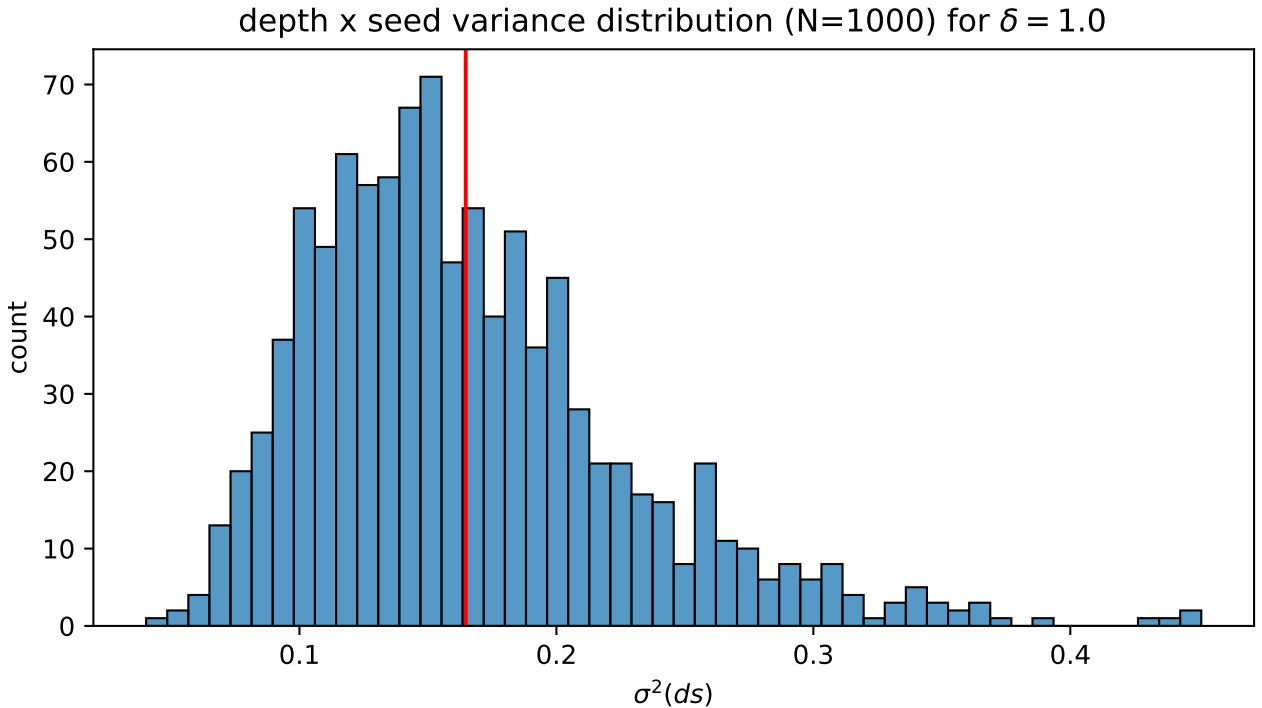

person variance distribution (N=1000) for $\delta = 1.0$
count
$\sigma^2(p)$
1e−30

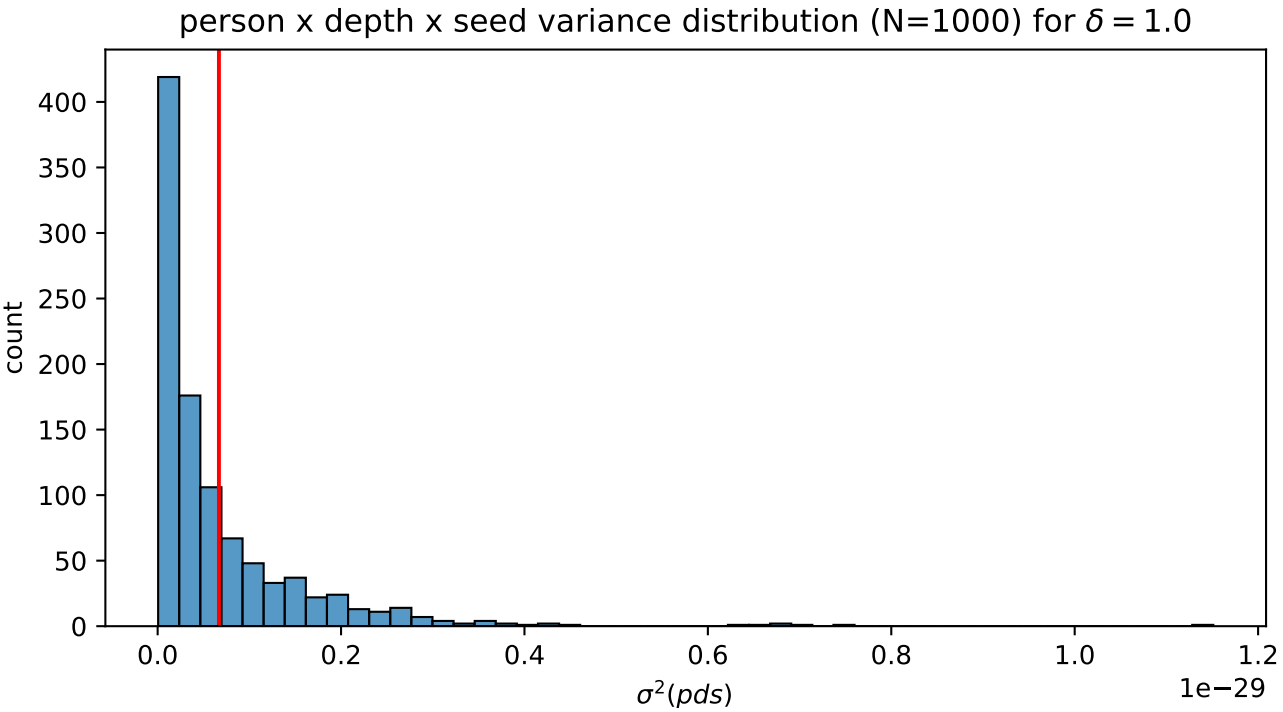

depth variance distribution (N=1000) for $\delta = 1.0$
count
$\sigma^2(d)$

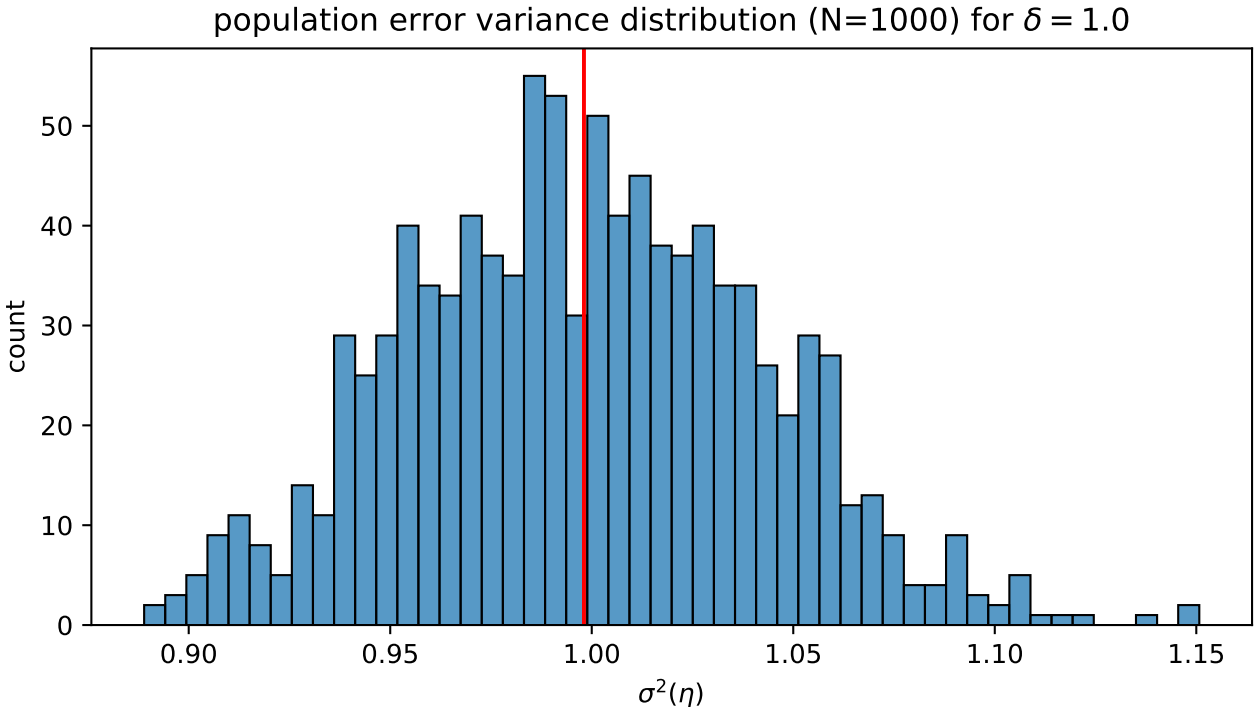

seed variance distribution (N=1000) for $\delta = 1.0$
count
$\sigma^2(s)$

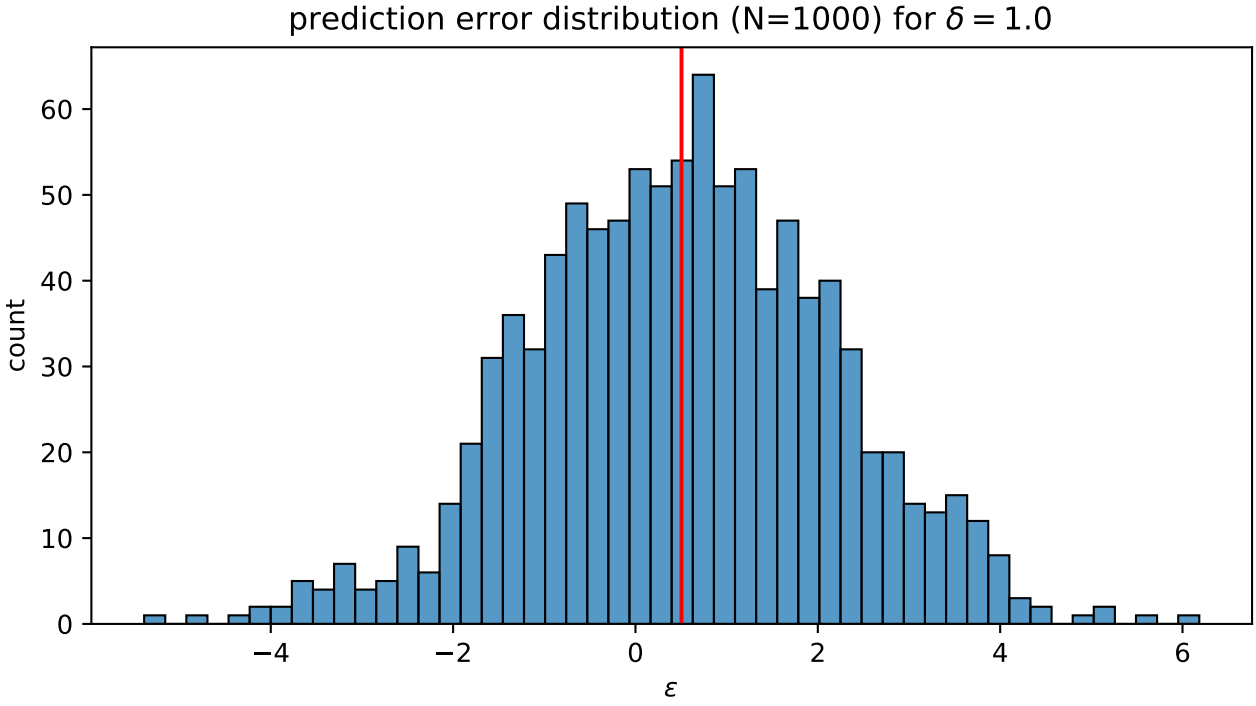

person x depth variance distribution (N=1000) for $\delta = 1.0$
count
$\sigma^2(pd)$
1e−30


person x seed variance distribution (N=1000) for $\delta = 1.0$
count
$\sigma^2(ps)$
1e−30


depth x seed variance distribution (N=1000) for $\delta = 1.0$
count
$\sigma^2(ds)$


person x depth x seed variance distribution (N=1000) for $\delta = 1.0$
count
0
50
100
150
200
250
300
350
400
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1e−29
$\sigma^2(pds)$


population error variance distribution (N=1000) for $\delta = 1.0$
count
0
10
20
30
40
50
0.90
0.95
1.00
1.05
1.10
1.15
$\sigma^2(\eta)$


prediction error distribution (N=1000) for $\delta = 1.0$
count
0
10
20
30
40
50
60
−4
−2
0
2
4
6
ε