Assessing the Predictive Validity of Qualitative Sensitivity Measures

Master Thesis Presented to the

Department of Economics at the

Rheinische Friedrich-Wilhelms-Universität Bonn

in Partial Fulfillment of the Requirements for the Degree of

Master of Science (M.Sc.)

Supervisor: Prof. Dr. Philipp Eisenhauer

Submitted in May 2021 by:

Benedikt Heinrich Müller

Matriculation Number: 3219976

1 Introduction

Economics is thinking in models that abstract from the actually existing economic systems by imposing assumptions on economic relations. Those assumptions are defendable as long as they are sufficiently realistic for the research question at hand (Friedman, 1953), what prompts the question, what sufficiently realistic means. Often, assumptions are decided upon by mathematical convenience: do the assumptions grant that the model is solvable, i.e. whether a solution exists in closed form and whether this solution is unique.

Due to the advancements in computational tractability one is less coerced to impose insufficiently realistic assumptions if numerical methods can be applied. That is, since even if the model at hand does not have an analytical solution, one can still obtain an approximate solution by applying numerical methods (Miranda and Fackler, 2004).

[Example with Rust model and why no analytical solution exists.]
Consider the following model

$$Y = f(X),$$

where Y is the response variable that depends on the values of some independent variables X. Let X denote the vector $X = (X_1, \ldots, X_k)' \in \mathbb{R}^k$, where k is the number of independent variables. Let x be the vector of specific values assigned to X, i.e. one realisation of X. Let $f(\cdot)$ denote a function that describes in which way Y depends on X. This function may be some complex function (e.g. computer code) or a model that can be solved by numerical methods only. Thus, f is generally not available in closed form and, hence, the relationship between Y and X is considered a black box.

In the remainder of this work, I name Y the (model) output and the vector X the (model) inputs. In this work I will consider X as being stochastic, that is, the X_i, $i = 1, \ldots, k$, follow a joint cumulative distribution function $G(X)$. Thus, although f is assumed to be a deterministic function of X, the output Y is also stochastic due to the uncertainty in X (Song et al., 2016).

In order to understand the model behaviour, sa is a way of learning about the input-output relationship of the model at hand (Borgonovo and Plischke, 2016). One can generally structure sa into local and global methods of sa (Borgonovo and Plischke, 2016). Local methods conduct sa around a certain point, or base case, x_0, in a deterministic framework, i.e. no probability distribution is assigned to X (Borgonovo and Plischke, 2016). In contrast, when performing global sa, we find ourselves in a stochastic context, which requires knowledge of the distribution of X, be it joint or marginal with or without dependence between the inputs (Saltelli and Tarantola, 2002).

The settings for which sa methods can yield useful information are further discussed in section 3.2. The result of performing sa is some sm, that depends on the sa method we apply to the context we find ourselves in. In this work I conduct variance-based sa, a method that evaluates the importance of an input by assigning to it the expected reduction

in output variance if this input was known with certainty (Borgonovo and Plischke, 2016). So, we have one instance of the sm for each model input.

One set of common variance-based sm are the total and first-order effects, or Sobol' indices, introduced by Sobol' (1993) for independently distributed input variables, which are based on an Analysis of Variances (henceforth ANOVA) decomposition. One obvious drawback of the Sobol' indices in the original definition is that they rely on the assumption of inputs independence. Even if the ANOVA decomposition is adapted to handle dependent inputs, they suffer from problems (Owen and Prieur, 2017) which are discussed in section 3.2. Owen and Prieur (2017) suggest using Shapley values instead, which were suggested in the context of quantitative sm by Owen (2014). Shapley values are a concept from game theory introduced by Shapley (1953). The main advantages of Shapley values for sa are that they are as easily applied to dependent inputs as they are interpreted. Furthermore, Shapley values satisfy a range of desirable properties, the most important being efficiency and the null-player property Shapley (1953) and Owen (2014). In the context of sa, Song et al. (2016) use the term Shapley effects. Despite of these appealing features, estimation of Shapley effects can be computationally demanding (Song et al., 2016).

In this work I estimate Shapley effects for a classical structural econometric model, the single-agent dynamic stochastic model of discrete choice introduced by Rust (1987).

2 The Rust Model

This section sets up the model whose input-output relationship I want to shed light on. I consider the single-agent dynamic discrete choice model of Rust (1987). Every period an agent needs to decide whether to replace the engine of a bus in the carpark or to conduct standard maintenance operations. The Rust model assumes optimal behaviour of the agent and asks what the implied deep economic parameters are, given this optimal behaviour.

The model output I consider is implied annual demand for engine replacements. Model inputs are engine replacement costs and a cost parameter that is part of the agent's utility function. In the Rust model these variables are estimated but I consider them to be inputs for the implied annual demand. Computing Shapley effects requires the simulation of the input variables. To that end, information about the distribution of the inputs are needed.

2.1 Bus-Engine Replacement Model

Information about the model is required for calculating the variance-covariance matrix and mean vector of the inputs.

In Rust (1987) an agent needs to manage a pool of buses for public transportation. Each period a bus exhibits an accumulated mileage state (odometer state) x and other state variables ε, that are unobserved by the econometrician. The agent has to decide whether to perform standard maintenance or an expensive engine replacement. The agent's

decision in period t is

$$i_t = \begin{cases} 0, \ \textit{standard maintenance} \\ 1, \ \textit{replace engine.} \end{cases} \tag{1}$$

If replacement is chosen, the accumulated mileage state x is set to zero. Otherwise, the bus is used for another period with standard maintenance and x increases until the next period. Note that x needs to me discretised for the estimation. I follow Rust (1987) and discretise the state space into 175 grids with a size of each bin of 2,571 miles.

The agent's utility function given state (x, ε) is described by

$$u(x, \ i, \ \varepsilon; \ \theta_1, \ RC) = v(x, \ i; \ \theta_1, \ RC) + \varepsilon(i), \tag{2}$$

where

$$v(x, \ i; \ \theta_1, \ RC) = \begin{cases} -c(x; \theta_1), \ if \ d = 0, \\ -RC - c(0; \theta_1), \ if \ d = 1, \end{cases}$$

and $\varepsilon(i)$ denotes utility shocks. In the above expression, RC represents the expected replacement costs of an engine replacement, that is, net the possible payoff of selling the old engine. $c(x; \theta_1)$ is a cost function of operating a bus at state x. θ_1 is a vector of structural parameters. I consider the linear cost function $c(x; \theta_{11}) = 0.001\theta_{11}x$.

[Give intuition for what that means: utility depends on what? Give examples.]

How the state (x, ε) changes over time, depends on the choice i_t and is described by the Markovian transition density $p(x_{t+1}, \varepsilon_{t+1} \mid x_t, \varepsilon_t, i_t; \theta_2, \theta_3)$, where θ_2 is assumed to be Euler's constant and θ_3 is a vector of transition probabilities. Markovian means that p depends on (x_t, ε_t) only and not on other past states.

The discount factor is denoted $\beta \in (0, 1)$. Given state $(x_t, \ \varepsilon_t)$, the agent maximises, choosing a sequence of decisions

$$\max_{\{i_t, \ i_{t+1}, \ i_{t+2}, \ ...\}} E\left[\sum_{\tau=1}^{\infty} \ \beta^{\tau - t} \ u(x_\tau, \ i_\tau, \ \varepsilon_\tau; \ \theta_1, \ RC)\right]. \tag{3}$$

By p being Markovian, the optimal decision is time-invariant. Thus, drop time indices. The decision problem is now defined by the solution to the Bellman equation

$$V(x, \ \varepsilon) = \max_i \{v(x, \ i; \ \theta_1, \ RC) + \varepsilon(i) + \beta \int_{x'} \int_{\varepsilon'} V(x', \ \varepsilon') p(x', \ \varepsilon' \mid x, \ \varepsilon, \ i; \ \theta_2, \theta_3) dx' d\varepsilon'\}. \tag{4}$$

The above problem is simplified by imposing some further assumptions to ease the estimation process. The model is solved by use of the so called NFXP (Rust, 1987). The parameters to be estimated are combined in a vector $\theta = (\theta_3, \ RC, \ \theta_{11})$. The estimation procedure via Maximum Likelihood is conducted by first estimating the transition proba-

Table 1: Variance-Covariance Matrix of the Input Variables

	RC	θ_{11}
RC	1.604736	0.605903
θ_{11}	0.605903	0.273094

Notes: Estimated variance-covariance matrix based on $10,000$ samples.

bilities θ_3. In a second step, the parameters RC and θ_{11} are estimated. I do not go into further details of how this model is solved. For details see Rust (1987).

2.2 Distribution of the Input Variables

The Maximum Likelihood estimator is asymptotically normally distributed (Rao, 1973). Thus, I estimate the variance-covariance matrix and the mean vector of θ_{11} and RC by an MC simulation. Note that this general setup is used in section 3 as well. For solving the model, I use the simulation and estimation capabilities of the Python *ruspy* package (Blesch, 2019). I chose *scipy_L-BFGS-B* as the solver for the NFXP algorithm (Virtanen et al., 2020). I proceed as follows.

First, I set the true parameters in θ. I chose to consider three transition probabilities, i.e. the mileage state x can increase from the current period to the next by two grids at most. I set

$$\begin{aligned} \theta_3 &= (\theta_{30},\ \theta_{31},\ \theta_{32}) = (0.39189182,\ 0.59529371,\ 0.01281447), \\ RC &= 10.07780762, \\ \theta_{11} &= 2.29417622. \end{aligned}$$

Note that RC is scaled by 0.001 for estimation purposes. I consider a total of 50 buses for 120 periods (months, i.e. in total ten years).

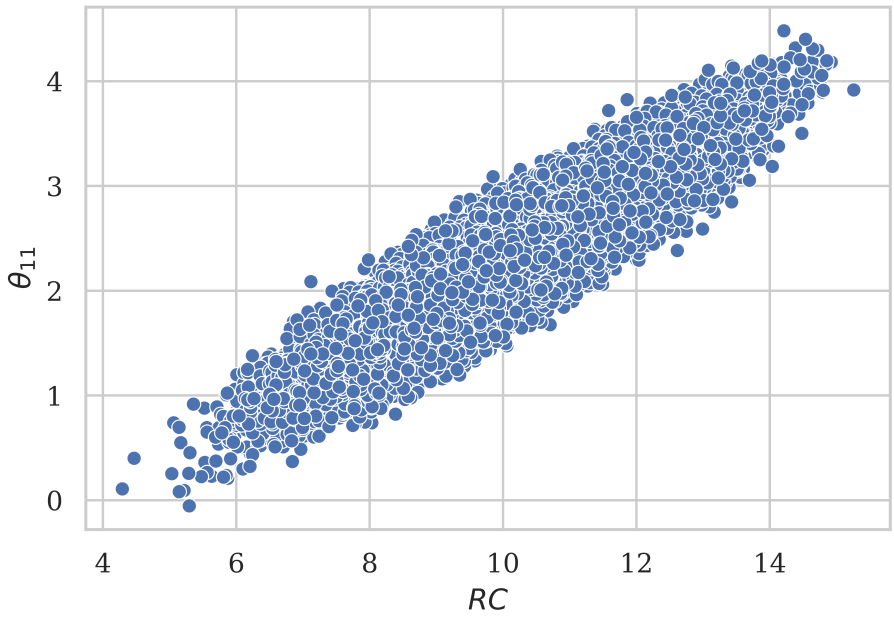
Second, I simulate $10,000$ data sets from the above true parameters and estimate $\hat{\theta}, \hat{\theta}$ being the vector of estimated structural parameters.

Third, from the $10,000$ parameter estimates of $\widehat{RC}$ and $\hat{\theta}_{11}$, I calculate the sample variance-covariance matrix and the mean vector. Table 1 shows the estimated variance-covariance matrix for $\widehat{RC}$ and $\hat{\theta}_{11}$. Table 1 and fig. 1 show that $\widehat{RC}$ and $\hat{\theta}_{11}$ correlate, hinting that inputs in the Rust model are indeed dependent.

2.3 Derivation of Implied Annual Demand

Rust (1987) shows how one can derive further quantities from the parameter estimates. Rust (1987) derives the implied annual demand (henceforth, demand) for bus-engine

Figure 1: Correlation Between the Input Variables



replacements. Rust (1987) computes the demand for bus engines by

$$\tilde{d}(RC) = \sum_{t=1}^{12} \sum_{i=1}^{M} \tilde{i}_t^m, \tag{5}$$

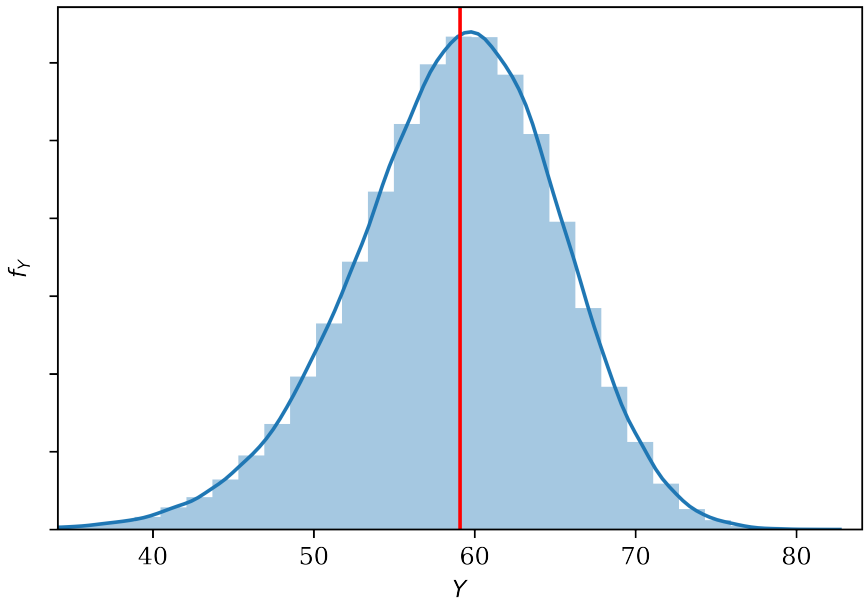
where M is the number of buses and $\tilde{i}_t^m$ is a realisation of the process $(i_t^m,\ x_t^m)$, with m being the bus index. Here, $M = 50$. Recall that a period is represented by one month, so we end up with the annual demand for 50 buses. $\tilde{d}(RC)$ is stochastic, since $\tilde{i}_t^m$ is. Let $\pi_m(x_0^m,\ i_0^m)$ denote the distribution of the initial state for each bus m. Given this, one can compute the probability distribution of $\tilde{d}(RC)$ by the transition probability $P(i_t \mid x_t,\ \theta)p(x_t \mid x_{t-1}, i_{t-1},\ \theta_3)$. Note that $P(\cdot)$ depends on θ, i.e. it depends on RC as well. We can compute demand for a range of values for RC. In my analysis I compute demand for a single value of RC only; $RC^{demand} = 11.5$, i.e. 11,500 money units. Rust (1987) considers the expected demand, i.e. $d(RC) = E[\tilde{d}(RC)]$. Assuming that the initial state distribution π is the long-run equilibrium distribution of $(i_t,\ x_t)$, π is the solution to

$$\pi(x,\ i) = \int_y \int_j P(i \ \mid \ x,\ \theta)\ p(x \ \mid \ y,\ \ j,\ \theta_3)\ \pi(dy,\ dj). \tag{6}$$

Note, that π is a function of θ. Assuming further that the processes (i_t^m, x_t^m) and (i_t^k, x_t^k) are independent for $k \neq m$, i.e. the process of a bus is independent from the process of any other bus, one can write

$$d(RC) = 12 \cdot M \cdot \int_0^{\infty} \pi(dx,\ 1). \tag{7}$$

Figure 2: Uncertainty Propagation



To show how the uncertainty in the inputs (RC, θ_{11}) propagate to the output, demand, see figure xy. I again use *ruspy* for the computation of the demand at RC^{demand}. I compute demand for $100,000$ simulated data sets. The output is normally distributed as can be seen in figure xy with mean xx and variance xx.

Now, that the model, i.e. how the output depends on the inputs, has been clarified, one is ready to consider the computation of Shapley effects.

3 The Benchmark: Computation of Shapley Effects

This section introduces Shapley effects for sa. Shapley effects will serve as a benchmark for evaluating the performance of the sm derived by the Morris method.

3.1 Shapley Values in Economics

The Shapley value is a concept from cooperative game theory, introduced by Shapley (1953). It is a value that informs about how to fairly distribute the profit of a product that was created by a team effort. Fairly here means, that any player receives as much of the gain as he or she contributed to the team effort. As individual contributions are not directly observed, Shapley (1953) proposes the following value. The Shapley value ensures that every individual receives at least as much as if he or she had if acted independently (Iooss and Prieur, 2019).

Consider a game with k players, where K denotes the set of all players, the grand coalition. A coalition is a subset $J \subset K$. Let $-J$ denote the complement set of J, i.e. $-J = K \setminus J$. The corresponding game is defined by a coalition value function that maps

each coalition J to the value attained by this coalition, i.e. $val : 2^K \to \mathbb{R}_{\geq 0}$, where 2^K denotes the power set of K (set of all subsets of K) (Song et al., 2016). It is generally assumed that $val(\emptyset) = 0$. The following definition is strongly inspired by the one found in Plischke et al. (2020):

Definition 1. *Given a coalition worth function val, the marginal contribution of player i joining coalition J is $mar(J,\ i) = val(J \cup \{i\}) - val(J)$. The Shapley value is then defined by*

$$\begin{aligned} \phi_i(val) &= \sum_{J|i \notin J} \frac{|J|!(k - |J| - 1)!}{k!}(val(J \cup \{i\}) - val(J)) \\ &= \frac{1}{k} \sum_{J|i \notin J} \binom{k-1}{|J|}^{-1} mar(J,\ i). \end{aligned} \tag{8}$$

In words, one considers every coalition of players and evaluates by how much the inclusion of player i increases the overall profit of the coalition. This is achieved by subtracting the value generated by the members of coalition J from the value created by the larger coalition that includes coalition J and player i. In the context of a game, the marginal value for each coalition is divided among all members of this coalition, e.g. if the coalition has one member only, the entire marginal value is gained by this player. One can also interpret the Shapley value as the expected payoff from joining a coalition or from leaving the complementary coalition (Plischke et al., 2020).

Shapley (1953) shows that ϕ_i satisfies the following four axioms and that the Shapley value is the only value that satisfies all four axioms at the same time. Note that Shapley values normalised by total profit are considered.

- Pareto efficiency: $\sum_{i=1}^{k} \phi_i = 1$.
- Symmetry: If $val(J \cup i) = val(J \cup j)\ \forall\ J \subseteq K \setminus \{i,\ j\}$, then $\phi_i = \phi_j$.
- Linearity: $\phi_i(val_1 + val_2) = \phi_i(val_1) + \phi_i(val_2)$.
- Null-player: If $\forall J,\ mar(J,\ i) = 0$ holds then $\phi_i(val) = 0$.

Pareto efficiency assures that nothing of the profit gained by the team is wasted. Normalised by the total profit, the Shapley value gives the share attained by each individual in the game. Symmetry results into all players who contributed the same to the common product, receiving the same payoff, reflecting the meritocratic principle. The null-player axiom ensures that players that do not contribute to the team effort at all, do not receive any share in the profits. Especially axioms 1 and 4 are desirable in the context of the Shapley value as a variance-based sm.

3.2 Variance-based Sensitivity Analysis

There are several purposes for which one can apply sa methods in general. Note, that in sa one refers to input variables by the term factors (Razavi et al., 2021). In their textbook, Saltelli, Tarantola, et al. (2002) summarise the following four possible objectives of sa.

Factors Prioritisation (FP): In the FP setting, one determines on which inputs applied, uncertainty reduction results into the largest reduction in output uncertainty. SA determines the importance of an input variable. Applied to all inputs, one can derive a ranking of all inputs in order of importance. FP can guide research by prioritising inputs. Inputs are identified for which better experimental measurement can reduce output uncertainty the most, supposing that additional information costs the same for alle inputs.

Factors Fixing (FF): The FF setting allows one to determine the least influential inputs. These then can be fixed at a specific value without losing information in the model output. FF can also be seen as inputs screening. One would want to fix the least influential inputs to reduce dimensionality of the model and thus the complexity and computational burden.

Variance Cutting (VC): VC informs about which inputs to fix to arrive at a certain desired value of the output variance, under the condition that the smallest number of inputs are fixed. VC is most useful for risk assessment.

Factors Mapping (FM): If one cares about the input importance in certain regions of output values only, one can apply FM. First, one classifies output values into groups and only then employs the importance exercise. So, the inputs are determined that contribute the most to producing output values in a target region. E.g. one could be interested in a certain percentile of the output range. The term mapping stems from mapping the importance of the inputs to the categories of Y.

SA does address more fundamental objectives of mathematical modelling and the analysis of systems that can be achieved directly from the above settings and are summarised in Razavi et al. (2021). By sa, one can achieve or conduct a. scientific discovery, i.e. identification of causal relationships, b. dimensionality reduction, i.e. determination of the least important inputs, which is achieved by FF, c. data worth assessment, which relates to FP, and d. decision support (Razavi et al., 2021).

In global sa, where, as one may recall, inputs are probabilistic in nature, one distinguishes six classes of sa methods (Borgonovo and Plischke, 2016). The sensitivity indices discussed and applied in this work belong to the area of variance-based sensitivity measures. Variance-based sensitivity measures are obtained by determining the expected reduction in output variance due to knowing input i with certainty (Borgonovo and Plischke, 2016). These methods are based on the classical formula for the law of total variance

$$V[Y] = V[E[Y \mid X_J]] + E[V[Y \mid X_J]] \tag{9}$$

assuming that $f(X) < \infty$, where the V operator stands for the unconditional variance of $Y, V[\cdot \mid \cdot]$ for the conditional variance, and $E[\cdot \mid \cdot]$ denotes the conditional expectation.

The above expression decomposes total variance into the explained and the unexplained component (citep!).

A variance-based sensitivity measure determines much output variance is attributable to each input i (Borgonovo and Plischke, 2016). Variance-based global sa is not applicable to decision variables, since global sa attaches a distribution to input variables, implying that inputs are uncertain (Song et al., 2016). Recall that due to this uncertainty in the inputs X_K the output Y is uncertain as well. $V[Y]$ measures this uncertainty in the output, where $V[Y]$ is taken according to the joint distribution of G_K. In the variance of the model output, there are three parts of variances; the one caused by every input in isolation, the one that is caused by interaction effects among inputs, and one that is due to input dependence.

Sobol' (1993) introduced often used variance-based sensitivity indices that attribute the variance reduction to each subset $J \subset K$ using an ANOVA decomposition. To ensure uniqueness of the ANOVA decomposition employed by Shapley (1953), one needs to assume independence of model inputs (Ge and Menendez, 2017). Sobol' indices were introduced as a subset importance measure (Song et al., 2016). For the purposes of sa we let the subsets be the single inputs, i.e. the subsets under considerations are singletons. The Sobol' indices are then defined as

$$S_i = V[Y]^{-1} V[E[Y \mid X_i]] \tag{10}$$

$$S_i^T = V[Y]^{-1} (V[Y] - V[E[Y \mid X_{-i}]]) \tag{11}$$

where X_{-i} is the subset of input variables without input i. What follows does apply to models with independent inputs only. The sensitivity measure S_i is called the first-order sensitivity index. S_i represents the share of output variance reduced by the isolated effect of input i, excluding contributions in variance reduction by interactions between input i and the remaining $(K-1)$ inputs. The subtrahend in the nominator of eq. (11) can be seen as the expected variance reduction when X_i is fixed at a certain value, i.e. if we know X_i with certainty (Song et al., 2016). S_i^T is called the total sensitivity index. S_i^T complements S_i in the sense that it measures the total effect of X_i in the output variance, including interaction effects (Song et al., 2016). It can be considered the expected remaining output variance, when all values of the inputs are known, except for the value attached to X_i (Song et al., 2016). By normalising both measures by $V[Y]$, it is clear that the value s of S_i and S_i^T are in the interval $[0, 1]$, since the numerators in equations xy and yz are always smaller than the total variance (Ge and Menendez, 2017). For the relationship between S_i and S_i^T the weak inequality $S_i \leq S_i^T$ holds true, while equality only holds when there are no interaction effects between X_i and X_{-i}. Note that the terms indices and effects are used interchangeably.

Under input independence, the Sobol' indices have a clear interpretation. If S_i is large, then the corresponding input X_i is an influential input with respect to the output variance reduction (Ge and Menendez, 2017). In contrast, a small first-order effect S_i does not

imply that X_i is an uninfluential input, if strong interaction effects are present (Ge and Menendez, 2017). As S_i^T measures the total variance contribution by X_i, one can infer from a small S_i^T that X_i is indeed not uninfluential (Ge and Menendez, 2017). If so, X_i could be fixed at a certain value without causing changes in the model output variance (Ge and Menendez, 2017).

Evaluating S_i and S_i^T one can also learn something about the structure of the model one is analysing, always assuming that inputs are independent. The model structure can be determined by the sums $\sum_{i \in K} S_i$ and $\sum_{i \in K} S_i^T$ (Ge and Menendez, 2017).If both sums are equal to one, interaction effects are negligible , since the model is additive in nature. The model is non-additive if $\sum_{i \in K} S_i < 1$ and $\sum_{i \in K} S_i^T > 1$ (Ge and Menendez, 2017). Thus, it can be inferred that interaction effects cannot be ignored and play a role in the system under consideration.

Recall that the statements about the model structure and input importance do not hold if inputs are dependent. In the case of dependence, one cannot simply apply the ANOVA decomposition since it is no longer unique (Owen, 2014). In the Rust model, there are two input variables, RC and θ_{11}. As one can see in section 2.2, there exists significant dependence between the inputs. Thus, Sobol' indices should not be used. Another variance-based sensitivity measure are the now popular Shapley effects, which can be used also in the context of dependence. They are derived in the following section.

3.3 Shapley Effects for Sensitivity Analysis

In the context of sa, Song et al. (2016) named Shapley values Shapley effects, which were first suggested as a variance-based sensitivity measure by Owen (2014). According to Plischke et al. (2020) the Shapley effect is becoming more and more popular in sa.

When applied as a sensitivity measure, the interpretation of the Shapley value changes. X_i is now interpreted as a model input instead of a player in a game. A coalition is now a subset of model inputs. Owen (2014) defines the function $\widetilde{val}$ as a function that assigns the explanatory power of this subset of inputs to this subset. That is, the function $\widetilde{val}$ assigns the conditional variance to a subset of inputs, $\widetilde{val}(J) = V[Y]^{-1}V[E[Y \mid X_J]]$ and $\widetilde{val}$ measures the reduction in $V[Y]$ due to the inputs in J (Song et al., 2016). (Song et al., 2016) show that the following alternative formulation of the value function results into the same Shapley effect.

$$val(J) = E[V[Y \mid X_{-J}]] \tag{12}$$

$val(J)$ can be interpreted as the remaining variance of Y, given the values of the inputs in $-J$ are known. Both formulations satisfy the following two requirements

$$val(\emptyset) = 0 \tag{13}$$

$$val(K) = V[Y]. \tag{14}$$

In words, the value of the empty input subset should be zero and the value of the set of all inputs should equal the entire output variance. In this work I consider the value functions normalised by $val(K) = V[Y]$ to get Shapley effects normalised to the interval $[0,\ 1]$. The most appealing properties of Shapley effects are that they satisfy the following two conditions (compare to axioms 1 and 4).

$$\sum_{i=1}^{k} \phi_i = 1 \tag{15}$$

$$\phi_i \geq 0,\ \forall\ i = 1, ..., k. \tag{16}$$

So, the Shapley effects calculated for a model sum up to one and each Shapley effect is weakly larger than zero. Hence, the Shapley effects are input importance measures in terms of the expected output variance reduction induced by X_i. The non-negativity condition ensures that Shapley effects are always clearly interpretable. Note that due to the uncertainty in the computation process there exists the possibility of Shapley effects turning out to be below zero. By increasing sample sizes, this phenomenon should be mitigated.

Shapley effects can be used to compare input importance, that is, inputs can be ranked according to their contribution to output variance reduction. Furthermore, the differences between values of Shapley effects can be interpreted.

Recall the three parts of model output variance as discussed in section 3.2: variance due to the isolated effect of an input (i.e. the main variance), due to interaction effects, and due to dependence among inputs. Shapley effects take all three into account. In this regard they differ from first-order and total Sobol' indices as defined in section 3.2 (Owen, 2014).

In comparison to Shapley effects one has to acknowledge that Sobol' indices can inform about the model structure as discussed in section 3.2. In addition, Sobol' indices address more sa settings than Shapley effects, if inputs are independent. Shapley effects can be applied to FF only, since they distribute the effect of interactions between inputs equally across all inputs contained in the current subset (Iooss and Prieur, 2019). FP cannot be precisely conducted by using Shapley effects, since one cannot distinguish between contributions of main variance and variance contributions due to due to interactions (Iooss and Prieur, 2019).

Shapley effects yield a single value for each input that serves as the sensitivity measure, as opposed to Sobol' indices, which yield the first-order and total effects., i.e. two measures per input. Especially when computing sensitivity indices for studies that evaluate scientific phenomena, having a single value per input that informs about the variance contribution is very useful (Song et al., 2016). Furthermore, when this measure considers main and interaction effects and can also handle input dependence, one has a very useful and versatile sensitivity measure.

Sobol' indices for dependent inputs have been proposed (Mara et al., 2015). This strategy is based on the estimation of four sensitivity indices, instead of the two measures in the case of input independence, elevating the practical usefulness of Shapley effects even more. In case of input dependence, Sobol' indices require a complicated ANOVA decomposition, whereas Shapley effects do not rely on such variance decompositions (Iooss and Prieur, 2019). When applying Shapley effects, one would not even need to know whether input dependence or independence prevails.

Several algorithms exist for estimating Shapley effects. See the following section.

3.4 Algorithm

The algorithm for the computation of Shapley effects needs to allow for conditional sampling of dependent inputs. Otherwise, some of the advantages of Shapley effects cannot be exploited. I use algorithm 1 from Song et al. (2016), see algorithm 1 in this paper.

Shapley effects consider all input subsets. To iterate over all subsets, Song et al. (2016) restate the equation for Shapley effects to iterate over all permutations of model inputs. For instance, let $k = 5$. Then, $K = \{1,\ 2,\ 3,\ 4,\ 5\}$. There exist 5! permutations, where one is $\pi' = (2,\ 4,\ 3,\ 5,\ 1)$. Let the set of all permutations of K be denoted by $\Pi(K)$. Further, let $P_i(\pi)$ denote all inputs in permutation π, that come before input i. In the above example, if $i = 3$, then $P_3(\pi') = \{2,\ 4\}$. Using the permutation representation, Song et al. (2016) state the marginal contribution due to input i as $val(P_i(\pi) \cup \{i\}) - val(P_i(\pi))$. By considering all permutations of K, we can restate the equation for the Shapley effects for input i as

$$\phi_i = \sum_{\pi \in \Pi(K)} (k!)^{-1}(val(P_i(\pi) \cup \{i\}) - val(P_i(\pi))). \tag{17}$$

The algorithm of Song et al. (2016) exploits the fact that their permutation-based algorithm evades redundant model evaluations by going through the permutations beginning with the smallest subset. The contribution of the preceding subset is subtracted from the contribution of the current subset. The marginal contribution is then computed by writing

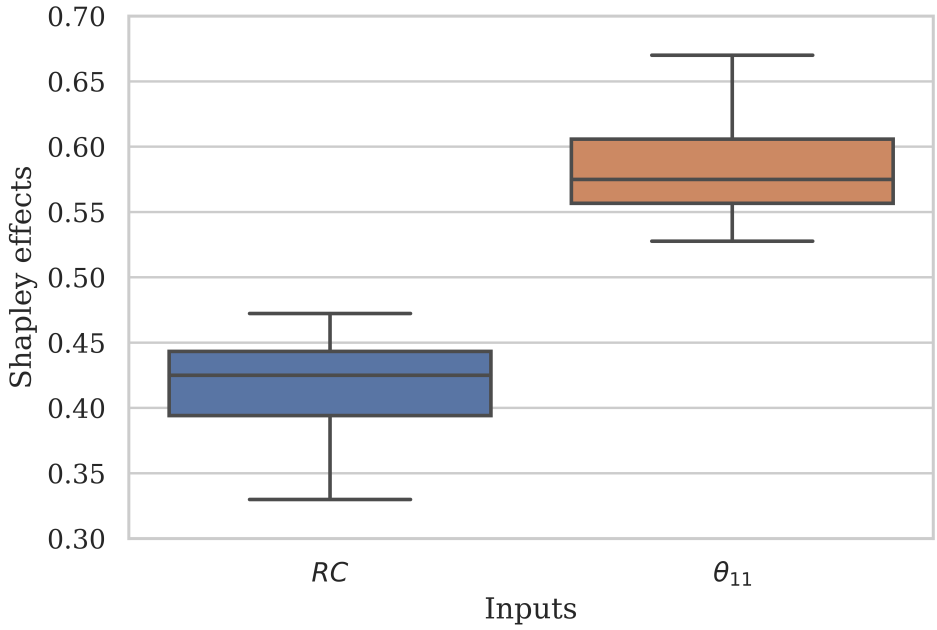
$$val(P_{\pi(j)}(\pi) \cup \{\pi(j)\}) - val(P_{\pi(j)}(\pi)), \tag{18}$$

where $\pi(j)$ denotes the input at position j in permutation π. In the above permutation π', at position 2 we have input 4, i.e. $\pi'(2) = 4$.

To see for which subsets the algorithm performs evaluations of val, consider the below example. For instance, following Song et al. (2016), if $k = 3$ and $\pi = (1, 3, 2)$, the algorithm computes

$$\begin{aligned}
&\Delta_1 = val(\{1\}) - val(\emptyset),\ \ set\ prevC = 0,\\
&\Delta_2 = val(\{1,\ 3\}) - prevC,\ \ set\ prevC = val(\{1,\ 3\}),\\
&\Delta_3 = val(\{1,\ 3,\ 2\}) - prevC,\ \ set\ prevC = val(\{1,\ 3,\ 2\}).
\end{aligned}$$

Figure 3: Shapley Effects for the Rust Model - 100 Replicates



Algorithm 1 implements three MC simulations. There are N_V MC samples of model inputs which are evaluated to get an estimate of $V[Y]$. Since the algorithm needs to handle dependent inputs, Song et al. (2016) implement dependent sampling by an inner and an outer MC simulation. To see why both MC simulations are needed, fix a permutation π. Then, sample N_O outer samples, that are unconditionally drawn. Given this set of unconditionally drawn outer samples, draw N_I inner samples conditionally on the outer samples. That is, for each outer sample, one has N_I inner samples. Thus, for algorithm 1, the computational cost in terms of number of model evaluations is given by $N_V + m \cdot N_I \cdot N_O \cdot (k-1)$, where m is the number of permutations considered. In the exact setting of algorithm 1, $m = |\Pi(K)| = k!$. If only a random subset of permutations should be considered, one sets $m < k!$. This could be useful to reduce the computational burden if the number of inputs is large.

3.5 Shapley Effects for the Rust Model

Song et al. (2016) recommend setting $m = |\Pi(K)|$, if computationally feasible. In order to reduce variance of the estimates of Shapley effects, Song et al. (2016) recommend choosing $N_I = 3$, while setting N_O as large as possible, given the constraints on the computational budget. Since the Rust model has two inputs only, the number of permutations to be consiedered is two and thus, very small. Hence, I set $m = 2! = 2$, $N_I = 3$, $N_V = 100$ and $N_O = 10$. Thus, the computational cost of my choice of MC runs imply that 160 model evaluations are needed to estimate the Shapley effects. On my machine (Windows 10, i5 processor), one estimation run takes approximately 55 seconds. I run the estimation by using the implementation of the package *econsa*, a Python package for sa (econsa\dev, 2021).

Table 2: Descriptive Statistics Shapley Effects - 100 Replicates

Shapley Effect	Mean	Std. errors	CI lower bound	CI upper bound
RC	0.416256	0.004317	0.407794	0.424718
θ_{11}	0.583744	0.004317	0.575282	0.592206

Notes: Descriptive statistics for 100 replicates of the Shapley effects.

I estimate 200 Shapley effects for the inputs of the Rust model, RC and θ by the setup as described in the preceding section. The distribution of the 200 replicates is visualised by the boxplots in figure xy. For further details on the estimated replicates, see table xy. The mean of ϕ_{RC} is 0.415624. For $\phi_{\theta_{11}}$ the mean value is 0.584376. The variance of the Shapley effects is 0.00135. Both, the boxplots, and the mean values show that θ_{11} is the more important input in terms of contributions to output variance. The confidence intervals at the 95-percent level, are quite large, but they do not overlap, indicating that the implied input importance ranking is robust. The lower value of ϕ_{RC} shows that RC has less an impact on the output variance than θ_{11}. Since both ϕ_i are far from zero, it is not recommended to fix any one of them, since both influence the output variance significantly.

4 Qualitative Sensitivity Measures: the Morris Method

This section introduces the Morris method for input screening.

4.1 Input Screening

In this section the applications of input screening is investigated.

4.2 Morris Method for Independent Inputs

The Morris method was introduced by Morris (1991) to identify the "important" input variables of a model, especially in cases where there are many inputs and/or the evaluation of a model is time-consuming.

Consider the same setup as employed in the preceding sections. Let $x = \{x_1, \ldots, x_k\}$ denote a sample of values assigned to the X_i's. $f(x)$ is then the model output obtained for the values in x. Now consider a second sample $x_{\Delta_i} = \{x_1, \ldots, x_{i-1}, x_i + \Delta, x_{i+1}, \ldots, x_k\}$ that is identical to x up to input x_i which is varied by Δ. Then, one elementary effect for input i is derived by

$$EE_i = \frac{f(x_{\Delta_i}) - f(x)}{\Delta}. \tag{19}$$

The above elementary effect is computed N times, each for a varying Δ (Ge and Menendez, 2017). The actual sm resulting from the Morris method are the mean, denoted

by μ_i, and the standard deviation, denoted by σ_i, taken from all N different elementary effects per input i.

$$\mu_i^* = \frac{1}{N}\sum_{r=1}^{N}|EE_{i,\,r}|, \tag{20}$$

$$\sigma_i = \sqrt{\frac{1}{N-1}\sum_{r=1}^{N}(EE_{i,\,r} - \mu_i)^2}, \tag{21}$$

with $EE_{i,\,r}$ denoting the r-th elementary effect of input i, $r = 1, \ldots, N$, and $|\cdot|$ the absolute value. Note that in Morris (1991) the absolute value was absent and elementary effects could potentially cancel each other out (Campolongo, Cariboni, et al., 2007). Therefore, Campolongo, Cariboni, et al. (2007) proposed the version presented above, thus making the screening method more robust. A total of $2 \cdot k \cdot N$ model evaluations is needed to compute the full set of sm using the Morris method.

μ_i^* and σ_i can now be used to identify non-influential model inputs. Uninfluential inputs exhibit a μ_i^* close to zero. If μ_i^* is large, it depends on σ_i whether there exist substantial non-linear or interaction effects. A low σ_i indicates that non-linear effects are non-existent, whereas a high σ_i suggests large interaction or non-linear effects (Ge and Menendez, 2017).

The Morris method exhibits some drawbacks. Firstly, as they stand, the sensitivity indices derived by the Morris method are not suited for screening inputs under dependence. To see why consider two inputs X_i and X_j which are dependent, i.e. $G(x_i,\ x_j) \neq G(x_i)G(x_j)$, where $G(\cdot)$ again denotes the cumulative distribution function. If x_i changes, x_j should change as well due to the dependence between the two inputs. The sensitivity indices presented above are derived using a One-At-a-Time approach that does not allow for the screening of dependent inputs (Ge and Menendez, 2017).

Secondly, similar to the Sobol' indices, the person conducting sa has to take two indices per input into account. Compare to the arguments made in section 3.2.

Thirdly, there exists no clear interpretation of the absolute values of the sensitivity indices. They only provide a ranking of inputs and give a hint of which inputs are the least influential ones (Ge and Menendez, 2017).

On the advantages, Morris indices are easily computed, with a much lower computational burden than the Shapley effects as presented in section 3. Recall that Shapley effects as computed by use of the algorithm in Song et al. (2016) came at a cost of $N_V + m \cdot N_I \cdot N_O \cdot (k-1)$ model evaluations. Even the more efficient approach by Plischke et al. (2020) needed 2^k model runs. See section 5 for a discussion of the respective computational costs.

Considering the interpretation of Morris indices, μ_i^* and σ_i, it is apparent that not only an input ranking is feasible but we can also learn something about the underlying model structure, i.e. whether interaction or non-linear effects are present.

Morris (1991) points out that his method does not rely on simplifying assumptions, e.g. monotonicity of the model or input sparsity. He argues that if those assumptions hold, one could apply other, more effective and economical procedures, e.g. based on Latin hypercube designs. However, the Morris method does not rely on such assumptions and will work well if these assumptions are justifiable or not (Morris, 1991).

Borgonovo and Plischke (2016) group the Morris method to the family of local sm. However, while the elementary effects themselves consider only local changes, the actual measures for input importance, μ_i^* and σ_i, average over these N elementary effects. Thus, they take N local changes per input i into account (Morris, 1991). Indeed, Campolongo, Saltelli, et al. (2011) make a case for the Morris method to be seen a global sm. Borgonovo and Plischke (2016) acknowledge that screening methods like the Morris sm stand apart from other local sm.

4.3 Algorithm for Extended Morris Method

In this section I introduce the extended Morris method for dependent samples as proposed by Ge and Menendez (2017).

To grasp the computation procedure of the extended elementary effects, some more notation is needed. Following Ge and Menendez (2017), let $X' = \{X_1',\ X_2',\ \ldots,\ X_k'\}$ be k dependent random inputs, following the joint pdf $g(X)$. Thus, X' is just a set of inputs independently drawn from the set X. Input subsets denoted by $\bar{X}$ are conditionally drawn inputs. Hence, let $\bar{X_{-i}'}$ follow the conditional pdf $g(\bar{X_{-i}'} \mid X_{-i})$. That is $\bar{X_{-i}'}$ is drawn conditionally on the inputs in the first set X_{-i}. The input denoted by $\bar{X_{-i}}$ is conditionally drawn following the pdf $g(\bar{X_{-i}} \mid X_i')$.

Analogously to the independent and full Sobol' indices (Mara et al., 2015), Ge and Menendez (2017) developed the following elementary effects for dependent inputs.

$$EE_i^{ind} = \frac{f(\bar{x_i}',\ x_{-i}) - f(x_i,\ x_{-i})}{\Delta}, \tag{22}$$

$$EE_i^{full} = \frac{f(x_i',\ \bar{x_{-i}}) - f(x_i,\ x_{-i})}{\Delta}, \tag{23}$$

where

- EE_i^{ind} denotes *independent* elementary effects for input i, effects that exclude the contributions attributable to the dependence between input X_i and X_j for $i \neq j$, and
- EE_i^{full} denotes *full* elementary effects for input i, that include the effects due to correlation with other inputs.

As in the case of the classic Morris method, the sm for input i is derived by considering N random samples yielding $2 \cdot N$ elementary effects, once for the independent and once for the full elementary effects. Ge and Menendez (2017) compute the corresponding sm as

shown in eq. (20) and eq. (21). Since two sets of elementary effects are computed, they end up with a set of four sm, (μ_i^{*ind}, σ_i^{ind}) and (μ_i^{*full}, σ_i^{full}).

Interpretation-wise, X_i is an unimportant input if all sm are essentially zero. A μ_i^{*ind} strongly larger than zero and σ_i^{ind} is close to zero, input X_i is an important input by itself. When all sm except μ_i^{*full} are close to zero, X_i's contribution is due to the dependence with other important inputs. Strong interaction effects are present if either or both of the two σ's is larger than zero (Ge and Menendez, 2017).

The computation of the extended Morris indices requires the generation of dependent samples. According to Ge and Menendez (2017), the computation involves the following steps:

1. Create independent, uniformly distributed samples.
2. Transform these uniformly distributed samples into dependent samples of a target distribution.
3. Use the above methods for the computation of the extended elementary effects.
4. As in the case of the classic Morris method, average over all N elementary effects per input i and compute their standard deviation. Do so for the *independent* and *full* elementary effects.

In what follows I stick to the version of the algorithm as implemented in the *econsa* Python-package (econsa\dev, 2021). There, the radial design is used for obtaining independent samples. To derive dependent samples, the inverse Nataf transformation is applied. The total computational cost amounts to $3kN$ model runs.

4.4 Morris Indices for the Rust Model

5 Comparing Shapley Effects and Morris Indices

6 Conclusion

Figure 4: Convergence of Morris Indices

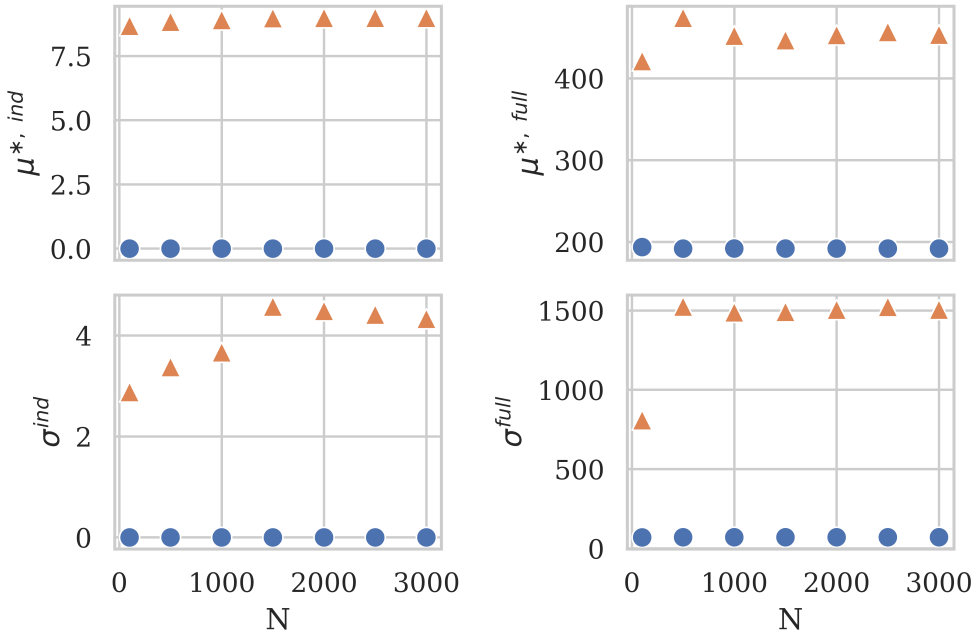


Figure 5: Uncertainty in Morris Indices - 100 Replicates