



*E*conometrics of *H*uman *C*apital

Philipp Eisenhauer

Material available on



Visit us!



UNIVERSITÄT **BONN**

grmpy Tutorial

Benedikt Kauf

Introduction

***grmpy* is ...**

- ▶ ... an open-source Python Package for the simulation and estimation of the generalized Roy model.
- ▶ ... intended as a useful device to support and improve the understanding of the framework by providing the opportunity to experience the effect of particular specifications directly.

Setup

Setup

- ▶ Normal linear-in-parameters version of the generalized Roy model.

Potential Outcomes

$$Y_1 = \beta_1 X + U_1$$

$$Y_0 = \beta_0 X + U_0$$

Cost

$$C = \gamma Z + U_C$$

Observed Outcomes

$$Y = DY_1 + (1 - D)Y_0$$

Choice

$$D = \mathbb{I}[D^* > 0]$$

$$D^* = \mu_D(X, Z) - V$$

$$U_D = F_V(V)$$

Features

Features

- ▶ *grmpy* is currently capable of the following features:
 - ▶ Simulating a dataset based on your own specifications.
 - ▶ Providing some useful information about the simulated dataset for instance:
 - ▶ Distributional outcome characteristics
 - ▶ B^{ATE}, B^{TT}, B^{TUT}
 - ▶ MTE by ventile
 - ▶ Estimating the coefficients of interest given a dataset (of a specific form).

Install the package

- ▶ OS, Linux : Use the pip install manager (*pip install grmpy*) or download the package via [GitHub](#) and install it manually.
- ▶ Windows: The same procedure as for Linux, OS but you have to verify that the *numpy* package is already installed on your machine.

Initialization file

- ▶ The initialization file provides the user with the opportunity to specify all parameters of his/her model, for instance:
 - ▶ Simulation parameters (number of observations, name of the output files)
 - ▶ Estimation parameters (optimization algorithm, start values)
 - ▶ Optimizer specifications
 - ▶ Coefficients and covariance parameters, dummy variables ...

Initialization file

- ▶ [Example](#)
- ▶ for a detailed explanation see: [grmpy-documentation](#)

Simulation

- ▶ *grmpy.simulate()*:
 - ▶ Input: path of the initialization file.
 - ▶ The function returns a data frame based on your specifications and different output files.
 - ▶ The data set as a pickle and a txt file.
 - ▶ An [Info file](#) that provides the distributional characteristics of the data as well as information about the different treatment effects.

Estimation

- ▶ *grmpy.fit()*:
 - ▶ Input: path of the initialization file.
 - ▶ At the moment the estimation process is only capable of two different optimization algorithms:
 - ▶ Broyden Fletcher Goldfarb Shanno (BFGS) algorithm
 - ▶ Powell's conjugate direction method

- ▶ There are two different options for the start values that could be set in the initialization file:
 - ▶ *init*: The estimation process uses the coefficient values specified in the initialization file as the start values for the estimation process.
 - ▶ *auto*: The start values are determined via a simple OLS followed by a Probit regression for the choice indicator.
- ▶ The estimation results are printed to an **output file**.

Test battery

- ▶ We also provide a test battery that includes several tests to ensure that the processes perform as intended.
 - ▶ Property-based testing
 - ▶ Reliability testing
 - ▶ Regression testing

What's new?

Figure: Replication Carneiro (2011)

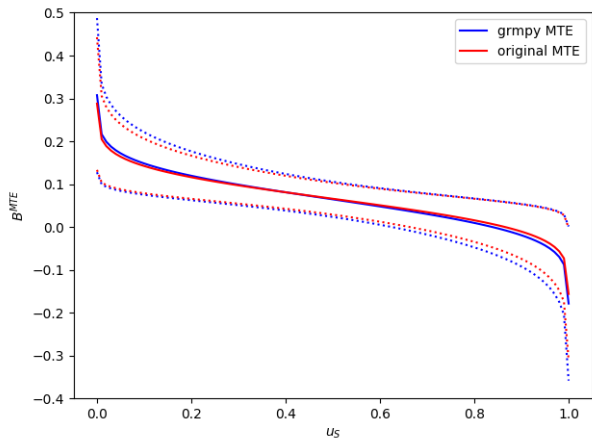
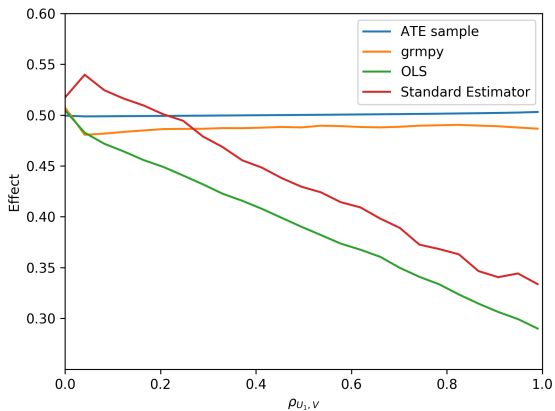


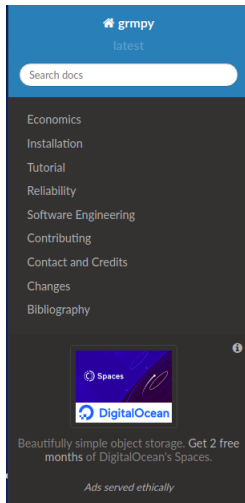
Figure: Performance comparison



Implementation of standard errors and adjustments on the output files

See: [Example](#)

Online documentation



The sidebar of the grumpy documentation website. It features a blue header with the 'grumpy' logo and 'latest' version indicator. Below is a search bar labeled 'Search docs'. A dark grey menu lists various sections: Economics, Installation, Tutorial, Reliability, Software Engineering, Contributing, Contact and Credits, Changes, and Bibliography. At the bottom, there is a DigitalOcean advertisement for Spaces, which includes the text 'Beautifully simple object storage. Get 2 free months of DigitalOcean's Spaces.' and 'Ads served ethically'.

[Docs](#) » Welcome to grumpy's documentation!

[Edit on GitHub](#)

Welcome to grumpy's documentation!

[PyPI](#) | [GitHub](#) | [Issues](#)

`grumpy` is an open-source Python package for the simulation and estimation of generalized Roy Model (Heckman & Vytlacil, 2005 [11]). It's main purpose is to serve as a teaching tool to promote the conceptual framework provided by the generalized Roy model which allows to illustrate a variety of issues in the econometrics of policy evaluation.

We build on the following main references:

James J. Heckman and Edward J. Vytlacil. [Chapter 70 Econometric Evaluation of Social Programs, Part I: Causal Models, Structural Models and Econometric Policy Evaluation](#). Handbook of Econometrics, 6, 4779 – 4874, 2007.

James J. Heckman and Edward J. Vytlacil. [Chapter 71 Econometric Evaluation of Social Programs, Part II: Using the Marginal Treatment Effect to Organize Alternative Econometric Estimators to Evaluate Social Programs, and to Forecast their Effects in New Environments](#). Handbook of Econometrics, 6, 4875 – 5143, 2007.

license [MIT](#)

Appendix

References

Abbring, J. H., & Heckman, J. J. (2007). Econometric evaluation of social programs, part III: Distributional treatment effects, dynamic treatment effects, dynamic discrete choice, and general equilibrium policy evaluation. In J. J. Heckman & E. E. Leamer (Eds.), *Handbook of econometrics* (Vol. 6, pp. 5145–5303). Amsterdam, Netherlands: Elsevier Science.

- Heckman, J. J., & Vytlačil, E. J. (2007a). Econometric evaluation of social programs, part I: Causal effects, structural models and econometric policy evaluation. In J. J. Heckman & E. E. Leamer (Eds.), *Handbook of econometrics* (Vol. 6B, pp. 4779–4874). Amsterdam, Netherlands: Elsevier Science.
- Heckman, J. J., & Vytlačil, E. J. (2007b). Econometric evaluation of social programs, part II: Using the marginal treatment effect to organize alternative economic estimators to evaluate social programs and to forecast their effects in new environments. In J. J. Heckman & E. E. Leamer (Eds.), *Handbook*

of econometrics (Vol. 6B, pp. 4875–5144). Amsterdam, Netherlands: Elsevier Science.