

Usage guide Code

Larson Beemster

11 July 2025

1 Introduction

This document contains a short description of the code used to generate the results for the thesis. If anything remains unclear after reading this description, feel free to write an e-mail to `l.m.j.beemster@student.vu.nl`.

2 Data structuring

To run the code, the dataset `crop_yield.csv` is required. First, a Jupyter notebook `data_analysis.ipynb` contains the exploratory data analysis on this dataset, which shows the results of the data section of the survey. Some summary statistics were instead obtained from the code provided in the model, as there were some instances filtered, which are included here. Second, the `cov_calculate.py` file can be used to compute a required covariance matrices yourself directly from the provided data. These are also provided, but can be calculated manually.

The price-yield covariance and correlation matrix are obtained by setting first both `COVARIANCE_PRICE_YIELD` and `NEGATIVE` to `False` and saving the matrix under `PQ`. Second, the covariance matrix of price-yield against price is obtained by setting `COVARIANCE_PRICE_YIELD` to `true`, but keeping `NEGATIVE` `false`, and this should be saved with the suffix `_PQ_P`. Finally, setting both to `true` yields the price yield against the negative of the price, all needed to run the model. Running this may take a while, so it is advised to simply use the provided matrices.

3 Running the models

The single product model, which code is given in `single_product.py`, can be run for each product individually, by changing the `crop_type` variable to the desired crop type. To save the figures, a folder is required named `Figures`. Note that for the organic implementation is not included here. Running this for different crop types will produce all results that were used in the thesis for the

single product model.

For the multi-product model, all results can simply be obtained by running the `multi_product.py` file. Similarly, all figures will be saved in the Figures folder. Lastly, the results from the credibility model can be obtained by running the `organic_implementation.py` file. Together, this is all the code used to produce the results.

Have fun running the code!