

# README

## Overview

The codes in this replication package produce all figures in the paper, except for Figures 2 and 14. All codes are Julia scripts and are arranged in distinct folders. Each folder contains one code, consisting of several modules. The main program is always called *main.jl*.

## Data Availability and Provenance Statements

This paper does not involve analysis of external data (i.e., no data are used or the only data are generated by the authors via simulation in their code).

## Computational and software requirements

A replicator should be able to run all codes on any platform (Linux, Windows, macOS) after modifying the first line of the file `main.jl`, which tells the compiler where to find the source. The only requirements are the installation of the Julia compiler and of a few Julia packages (see below). We last ran the codes with the operating system Ubuntu 22.04.5 LTS.

A replicator that is new to the Julia environment is advised to install the integrated development environment Visual Studio Code, developed by Microsoft for Windows, Linux, and macOS operating systems. The binaries are available for download at no charge from <https://code.visualstudio.com>. She should then install the extension *julia*, by selecting *File*, *Preferences* and *Extensions* in the dropdown menu and digiting *julia* in the search box. The codes were compiled using Julia version 1.11.5.

The scripts make use of a number of Julia packages, listed below. Each folder contains a subfolder called *myproject*, which includes the project environment, in turn consisting of the files *Manifest.toml* and *Project.toml*. *Project.toml* lists the direct dependencies of each project. *Manifest.toml* is a complete record of the dependency tree of each project and provides the specific versions of all packages.

- List of Julia packages
  - Distributions
  - FileIO
  - JLD2
  - LaTeXStrings
  - Measures
  - NLSolve

- Plots
- QuadGK
- Revise
- In the preamble to every *main.jl* file are three sets of instructions
  1. `cd("/home/clem/Dropbox/Alberto_Piero/replication/Scripts/folder_name")`
  2. `if length(LOAD_PATH)==3`  
    `push!(LOAD_PATH,pwd())`  
    `end`
  3. `using Pkg`  
    `Pkg.activate("myproject")`

The first set tells the compiler where to find the code. The second modifies the load path by telling the compiler that modules to be loaded are in the same directory. The third set activates the project environment.

### Controlled Randomness

No Pseudo random generator is used in the analysis described here.

### Memory, Runtime, Storage Requirements

The code was last run on a **72-core Dell Precision 7960 Tower with 128 GB of RAM and 1 TB of local storage**, running Ubuntu 22.04.5 LTS. The approximate storage space requirements is 5 MB. The time needed to run each code is reported below.

### Description of programs (including runtime)

Below is a complete list of the codes needed to reproduce the figures in the paper. It is important to execute the codes in order.

- Folder **Scripts/01\_mixed**.  
 The program solves for mixed strategy equilibria. It produces a data file called **Data/statistiche\_lowchi.jld2**. The code is parallelized. The approximate run time is 2 hours and 23 minutes.
- Folder **Scripts/02\_incomplete\_market**.  
 It requires in input the data file **Data/statistiche\_lowchi.jld2**. Output consists of the data files **Data/statistiche\_incomplete.jld2** **Data/statistiche\_nonseg.jld2**, as well as **Figures/mixed.pdf** (Figure 2 in the paper). Approximate runtime is 47 minutes.
- Folder **Scripts/03\_complete\_market**.  
 The program solves for complete market equilibria. It takes in input the data file **Data/statistiche\_incomplete.jld2**. It produces in output the

data file `Data/statistische_complete.jld2`, to be used below. Approximate runtime is less than 5 seconds.

- Folder `Scripts/04_equity_only`.  
The program solves for equilibria in the case in which equity is the only asset. It produces a data file called `Data/statistische_equity.jld2`, to be used below. Approximate runtime is 3 minutes and 30 seconds.
- Folder `Scripts/05_mainfigs`.  
The code produces `Figures/firm_decisions.pdf` (figure 3 in the paper), `Figures/consumption.pdf` (figure 4), and `Figures/asset_returns.pdf` (figure 5). It requires in input the datafiles `Data/statistische_incomplete.jld2`, `Data/statistische_complete.jld2`, and `Data/statistische_equity.jld2`. Approximate runtime is less than 1 second.
- Folder `Scripts/06_variance`.  
The code produces the data files `Data/statistische_sigma_chilow.jld2` and `Data/statistische_sigma_chihigh.jld2`. It produces `Figures/csvar_2.pdf` (figures 6) and `Figures/csvar_3.pdf` (figure 7). Approximate runtime is 12 minutes.
- Folder `Scripts/07_gov`.  
The program solves for equilibria in the case in which there is exogenous provision of risk-free debt. The code produces the data file `Data/statistische_gov.jld2`, `Figures/csgov_2.pdf` (figure 8) and `Figures/csgov_3.pdf` (figure 9). Approximate runtime is 16 minutes.
- Folder `Scripts/08_derivative`.  
The code produces the data file called `Data/statistische_der.jld2` and `Figures/intermediation.pdf` (figure 10). Approximate runtime is 20 seconds.
- Folder `Scripts/09_discrete`.  
The code produces data files `Data/statistische_disc.jld2` and `Data/statistische_bench.jld2`, as well as `Figures/specialization_1.pdf` (figure 11), `Figures/specialization_2.pdf` (figure 12), and `Figures/specialization_3.pdf` (figure 13). Approximate runtime is 17 minutes.
- Folder `Scripts/10_agency`.  
The code produces data files `Data/statistische_agency.jld2`, `Data/statistische_noag.jld2`, and `Data/statistische_pe.jld2`, as well as `Figures/agency_mrs.pdf` (figure 15) and `Figures/agency.pdf` (figure 16). Approximate runtime is 2 minutes and 20 seconds.

## List of tables and programs

The codes reproduce all figures appearing on the paper, as detailed in Table 1, except figures 1 and 14, respectively, that are qualitative illustrations.

Figure	Program	Output
Figure 2	\Scripts\02_incomplete_market\main.jl	mixed.pdf
Figure 3	\Scripts\05_mainfigs\main.jl	firm_decisions.pdf
Figure 4	\Scripts\05_mainfigs\main.jl	consumption.pdf
Figure 5	\Scripts\05_mainfigs\main.jl	asset_returns.pdf
Figure 6	\Scripts\06_variance\main.jl	csvar_2.pdf
Figure 7	\Scripts\06_variance\main.jl	csvar_3.pdf
Figure 8	\Scripts\07_gov\main.jl	csgov_2.pdf
Figure 9	\Scripts\07_gov\main.jl	csgov_3.pdf
Figure 10	\Scripts\08_derivative\main.jl	intermediation.pdf
Figure 11	\Scripts\09_discrete\main.jl	specialization_1.pdf
Figure 12	\Scripts\09_discrete\main.jl	specialization_2.pdf
Figure 13	\Scripts\09_discrete\main.jl	specialization_3.pdf
Figure 15	\Scripts\10_agency\main.jl	agency_mrs.pdf
Figure 16	\Scripts\10_agency\main.jl	agency.pdf

Table 1: Figures reproduced by the codes