

Replication Package ReadMe for “The Credit Line Channel” by Daniel Greenwald, John Krainer, and Pascal Paul

May 2025

Overview

Our replication package contains two components, each with its own directory.

1. **Empirical:** contains the empirical analysis in Sections I through III of the paper. Due to confidentiality, these results can only be reproduced using pseudodata and will not match the results found in the paper.
2. **Model:** contains the structural model results in Section IV of the paper. This component uses no confidential data and should exactly reproduce the results found in the paper.

Full details for running each component now follow in the sections below.

Empirical Analysis

Software: Codes are written in Stata (version 17.0 was used) and in Matlab (version 2020a was used).

Data Sets: For the empirical analysis, data from the following sources was used:

1. **H.1 schedule of FR Y-14 data:** These are confidential data. Only employees within the Federal Reserve System can apply for access to these data. The data was last updated in February 2021. A data dictionary can be found at: https://www.federalreserve.gov/apps/reportingforms/Report/Index/FR_Y-14Q
2. **S&P's Compustat database:** These are proprietary data and cannot be shared publicly. The data was obtained from WRDS and last updated in January 2021. These data can be purchased from S&P: [https://www.marketplace.spglobal.com/en/datasets/compustat-financials-\(8\)](https://www.marketplace.spglobal.com/en/datasets/compustat-financials-(8))
3. **Orbis data:** These are proprietary data and cannot be shared publicly. The data can be purchased from Bureau van Dijk: <https://login.bvdinfo.com/R1/Orbis>

4. **FR Y-9C data:** These data are publicly available at <https://www.ffiec.gov/npw/FinancialReport/FinancialDataDownload>. They were last updated in January 2021.
5. **CPI data** that is publicly available from St. Louis Fed's FRED database. The data was last updated in January 2021 and can be found at: https://fred.stlouisfed.org/series/CPALTT01USQ661S?utm_source=series_page&utm_medium=related_content&utm_term=other_formats&utm_campaign=other_format

Replication of Tables and Figures:

Due to the confidentiality of the data, pseudo data sets are made available to replicate the figures and tables in the empirical section of the paper. The replication codes can be found in the following folders:

1. **Descriptive Evidence:** This folder contains the codes to replicate Tables 3.1 and 3.2, as well as Figures 3.1, 3.2, 3.3., and 4.1. The folder called "Input" contains the pseudo data sets and the folder called "Output" contains the content for the tables and figures.
 - a. The Stata do-file "run_descriptive_table.do" constructs the content for Table 3.2.
 - b. The Stata do-file "run_descriptive_figures.do" constructs the content for Table 3.1 and Figures 3.1, 3.2, 3.3., and 4.1. Note: you will need to set a flag at the top of the do file to specify which table or figure you want to create.
 - c. The figures in the paper are then constructed using Matlab (using the input files from the previous step). To construct the figures, one needs to run the code called "Run_Graphs" in the folder "Matlab Figures".
2. **Regression Evidence:** This folder contains the codes to replicate Tables 4.1, 4.2, and 4.3. To this end, one has to run the file "run_credit_supply_regressions.do". The folder called "Input" contains the pseudo data sets and the folder called "Output" contains the content for the tables. Note: you will need to set a flag at the top of the do file to specify which table or figure you want to create.

Structural Model

Software: The structural model code uses Matlab to generate the model results and Python to create the model figures.

1. **Matlab:** we use Matlab to run the model code using the Dynare package. Our code was last run using Matlab 2024a and Dynare 6.1.
2. **Python:** we use Python to create the model plots, using the following additional packages: matplotlib, numpy, pandas, scipy, tabulate. Our code was last run using Python version 3.13.3, installed using Miniforge. Our conda environment can be recreated using the command

```
conda install matplotlib numpy pandas scipy tabulate
```

For an exact reproduction of our environment using the same package versions, you can directly install from the file `credit_line_channel.yml`, located inside the `Model/python` directory. To do this, use command

```
conda env create -f credit_line_channel.yml
```

which you can then activate using

```
conda activate credit_line_channel
```

Alternatively, you can create the environment using pip from the `credit_line_channel.txt` file in `Model/python` via the command

```
pip install -r requirements.txt
```

Data Sets: our model plots use basic aggregate data obtained via Federal Reserve Economic Data (FRED) and the Financial Accounts of the United States (FAUS).

1. **FRED data.** We obtain several aggregate series from FRED (see Internet Appendix Section IA.A.1 for details). The original sources are the Bureau of Economic analysis and Moody's. We provide the vintage versions of the files that we use in the paper as Python pickle files, which were downloaded in March and October of 2024. Updated versions can be obtained from FRED using the provided series codes.

2. **FAUS data.** We obtain additional series for the US corporate nonfinancial sector from the FAUS (formerly the Flow of Funds). We use Tables B.103 and F.103, and provide the vintage versions that we use in the paper, which date back to June 8, 2022. Updated versions can be obtained from the Federal Reserve Board of Governors (<https://www.federalreserve.gov/releases/z1/>).

Replication of Tables and Figures:

Reproducing our model results involves first solving the model perfect foresight paths in Matlab using the Dynare package, then assembling the aggregate time series data and plotting the results in Python. Step-by-step instructions are as follows:

1. In the Model/dynare directory, open Matlab and run the script `compute_deterministic_paths.m`.

2. In the Model/python directory, run

```
python create_aggregate_data.py
```

to create the aggregate data set used in the model plots.

3. In the Model/python directory, run

```
python perfect foresight_plots.py
```

to create the model plots (Figures 6 and 7 in the paper). These figures will now be saved in the Model/output directory.

4. To collect and rename Figures 6 and 7, you can optionally run the `collect_figures.sh` shell script using

```
bash collect_figures.sh
```

This will create `figures_6.pdf` and `figure_7.pdf` in the directory `Model/figures_for_paper`.