Overview

The data and code in this replication package reproduce all tables and figures in Hengel (2022). Raw data are in the `0-data/fixed` directory. Transformed data, estimation results and figures are saved in the `0-data/generated`, `0-tex/genderated` and `0-images/generated` directories, respectively. The replication code, described in detail below, will take about 10–11 hours to run.

The data in this replication package are publicly available and licensed under a Creative Commons Attribution 4.0 International License. See LICENSE.txt for details.

Data

Main dataset: `read.db`

Almost all figures and tables in Hengel (2022) were generated using the raw data in `0-data/fixed/read.db`. `read.db` is an SQLite database of biblio- and biographic data on articles and authors published in the *American Economic Review* (*AER*), *Econometrica*, *Journal of Political Economy* (*JPE*), *Quarterly Journal of Economics* (*QJE*) and the *Review of Economic Studies* (*REStud*). `read.db` contains 12 tables. Their contents and provenance are described below. Figure 1 displays `read.db`'s entity-relationship diagram; Table 1 describes each column. Please see Hengel (2022), Section 2 and Appendices C and D (online appendix) for additional details on data provenance and variable construction.

- **Article**. The Article table contains bibliographic information from every English-language article published with an abstract in the *AER*, *Econometrica*, *JPE* and *QJE* between January 1950 and December 2015 (inclusive). It also contains data on *REStud* articles published with submission and acceptance dates during that same period.

  All data were collected from publicly available sources (*e.g.*, publishers' websites and JSTOR). The exception is citations which were obtained from Web of Science in September 2017 and January 2018. (***Citation data are proprietary to Web of Science and are included here for replication purposes only; please do not distribute these data or publish online.***) Data on submit-accept times and institutions were collected from journals' online archives, extracted from digitised articles using the open source command utility `pdftotext` or entered manually by me or a research assistant.

  The column `Abstract` contains unicode textual data of articles' abstracts. When using the data, please ensure it is imported with the proper encoding (*e.g.*, `encoding("utf-8")` in Stata (14+) or `iconv(, from="macintosh", to="UTF-8")` in R).

- **Author**. The Author table contains biographic details on authors. Gender was initially assigned using GenderChecker.com's database of male and female names. Three separate Mechanical Turk workers, a research assistant or I then manually verified them based on photos and other information found on faculty websites, Wikipedia articles, *etc.* In situations where the author could not be found, I emailed or telephoned colleagues and associated institutions.

  Authors were assumed to be native-English speakers if one or more of the following criteria were satisfied: (i) they were raised in an English-speaking country; (ii) they obtained all post-secondary education from English speaking institutions; or (iii) they spoke with no discernible non-native accent. This information was almost always found on authors' CVs, websites, Wikipedia articles, faculty bios or obituaries. In a small number of cases the criteria were ambiguously satisfied or not available; in these instances I asked friends and colleagues of the author or inferred English fluency from the author's first name, country of residence or surname (in that order). If one co-author on a paper was found to be a native English speaker, I did not necessarily check whether any of the other co-authors were also native English speakers.

- **AuthorCorr**. The AuthorCorr table maps `AuthorID` in Author to `ArticleID` in Article.

- **Children**. The Children table contains data on the year female authors with at least one exclusively female-authored paper published in *Econometrica* gave birth. This information was obtained from authors' published profiles, CVs, acknowledgements, Wikipedia, personal websites, Facebook pages, background checks and by consulting local school district/popular extra-curricular activity websites. Exact years were recorded whenever found; otherwise, they were approximated by subtracting a
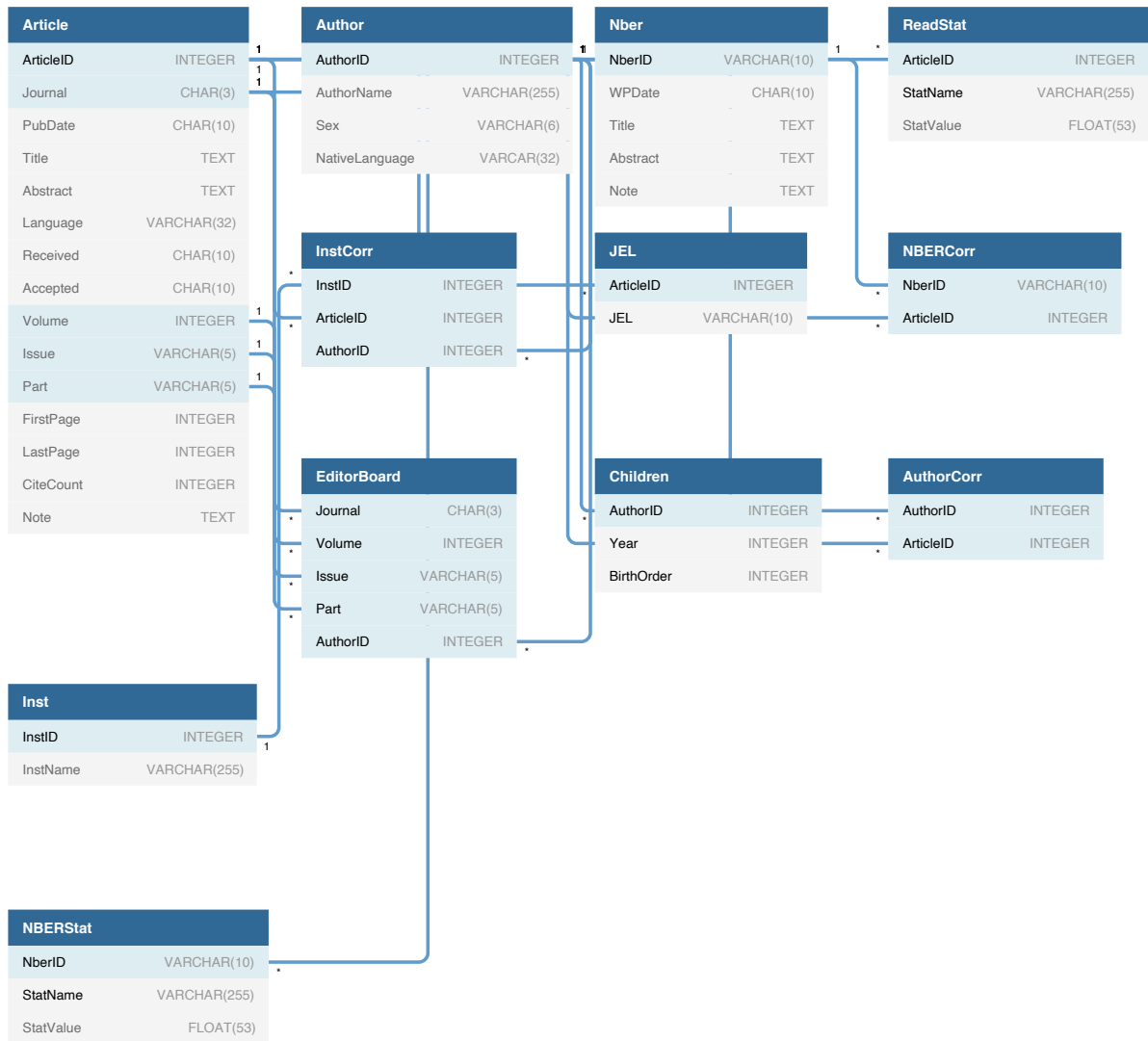
Article

ArticleID	INTEGER
Journal	CHAR(3)
PubDate	CHAR(10)
Title	TEXT
Abstract	TEXT
Language	VARCHAR(32)
Received	CHAR(10)
Accepted	CHAR(10)
Volume	INTEGER
Issue	VARCHAR(5)
Part	VARCHAR(5)
FirstPage	INTEGER
LastPage	INTEGER
CiteCount	INTEGER
Note	TEXT

Author

AuthorID	INTEGER
AuthorName	VARCHAR(255)
Sex	VARCHAR(6)
NativeLanguage	VARCAR(32)

Nber

NberID	VARCHAR(10)
WPDate	CHAR(10)
Title	TEXT
Abstract	TEXT
Note	TEXT

ReadStat

ArticleID	INTEGER
StatName	VARCHAR(255)
StatValue	FLOAT(53)

InstCorr

InstID	INTEGER
ArticleID	INTEGER
AuthorID	INTEGER

JEL

ArticleID	INTEGER
JEL	VARCHAR(10)

NBERCorr

NberID	VARCHAR(10)
ArticleID	INTEGER

EditorBoard

Journal	CHAR(3)
Volume	INTEGER
Issue	VARCHAR(5)
Part	VARCHAR(5)
AuthorID	INTEGER

Children

AuthorID	INTEGER
Year	INTEGER
BirthOrder	INTEGER

AuthorCorr

AuthorID	INTEGER
ArticleID	INTEGER

Inst

InstID	INTEGER
InstName	VARCHAR(255)

NBERStat

NberID	VARCHAR(10)
StatName	VARCHAR(255)
StatValue	FLOAT(53)

Figure 1: Entity-relationship diagram for `read.db`

child's actual or estimated age from the date the source material was posted online. In several instances, I obtained or verified this information from acquaintances, friends and colleagues or by asking the woman directly. If an exhaustive search turned up no reference to children, I assumed the woman in question did not have any.

Please note that data on children's birth years were only systematically collected for children potentially born during the time a woman had an exclusively female-authored paper under review at *Econometrica.*

- **EditorBoard**. The EditorBoard table contains the `AuthorID` for every editor and each issue of a journal. Editors were identified from issue mastheads.

- **Inst**. The Inst table maps each unique `InstID` to an institution name.

- **InstCorr**. The InstCorr table maps each (`ArticleID`, `AuthorID`) combination in AuthorCorr to at least one `InstID` in Inst.

- **JEL**. The JEL table maps *JEL* codes to each `ArticleID` in Article. The *JEL* system was significantly revised in the 1990s; because exact mapping from one system to another is not possible, I collected these data only for articles published post-reform. Codes were recorded whenever found in the text of an article or on the websites where bibliographic information was scraped. Remaining articles were classified using data from the American Economic Association's Econlit database.

- **NBER**. The NBER table contains basic bibliographic data on the NBER working papers that were eventually published in a top-four journal. Data were scraped from the NBER website or extracted from digitised working papers.

- **NBERCorr**. The NBERCorr table maps each `NberID` in NBER to at least one `ArticleID` in Article. (The mapping is not one-for-one because a small number of working papers were eventually published as multiple articles or combined into one.) Matches were identified using citation data from RePEc and by searching NBER's database directly for unmatched papers authored by NBER family members. The column `Note` contains notes in situations where the matching process involved some degree of ambiguity (*e.g.*, because of title changes between draft and final versions of the paper).

- **ReadStat**. The ReadStat table contains readability statistics for every article with an abstract in Article. Readability scores were generated with the Python module `Textatistic` using the text in the `Abstract` column of the Article table. `Textatistic`'s code and documentation are available on GitHub; a brief description is provided in Hengel (2022), Appendix D.3.

- **NberStat**. The NberStat table contains readability statistics calculated from the `Abstract` text of every working paper in the NBER table. Statistics were generated using the Python module `Textatistic`.

Table 1: Description of variables in `read.db`

Table	Column name	Description
Article	`ArticleID`	Unique ID for each article
Article	`Journal`	Journal (*AER*, *ECA*, *JPE*, *QJE*, *RES*, *P&P*)
Article	`PubDate`	Date of publication (YYYY-MM-DD)
Article	`Title`	Title
Article	`Abstract`	Abstract
Article	`Language`	Language (*e.g.*, English or French)
Article	`Received`	Date (YYYY-MM-01) manuscript first submitted
Article	`Accepted`	Date (YYYY-MM-01) manuscript finally accepted
Article	`Volume`	Volume
Article	`Issue`	Issue
Article	`Part`	Part
Article	`FirstPage`	First page number
Article	`LastPage`	Last page number
Article	`CiteCount`	Citation count (Web of Science)
Article	`Note`	Notes on observation

Table	Column name	Description
Author	AuthorID	Unique ID for each author
Author	AuthorName	Author name
Author	Sex	Author gender
Author	NativeLanguage	Native-English speaking indicator (English, Non-English, Unknown)
AuthorCorr	AuthorID	Unique ID for each author (maps to Author table)
AuthorCorr	ArticleID	Unique ID for each article (maps to Article table)
Children	AuthorID	Unique ID for each author (maps to Author table)
Children	Year	Year a child was born
Children	BirthOrder	Order of birth if two children are born in the same year
EditorBoard	AuthorID	Unique ID for each editor (maps to Author table)
EditorBoard	Journal	Journal
EditorBoard	Part	Part
EditorBoard	Volume	Volume
EditorBoard	Issue	Issue
Inst	InstID	Unique ID for each institution
Inst	InstName	Institution name
InstCorr	InstID	Unique ID for each institution (maps to Inst table)
InstCorr	ArticleID	Unique ID for each article (maps to Article table)
InstCorr	AuthorID	Unique ID for each author (maps to Author table)
JEL	ArticleID	Unique ID for each article (maps to Article table)
JEL	JEL	Tertiary *JEL* code
NBER	NberID	Unique ID for each NBER working paper
NBER	WPDate	Date (YYYY-MM-DD) manuscript was released as a working paper
NBER	Title	Working paper title
NBER	Abstract	Working paper abstract
NBER	Note	Notes on observation and/or matching process
NBERCorr	NberID	Unique ID for each NBER working paper (maps to NBER table)
NBERCorr	ArticleID	Unique ID for each article (maps to Article table)
ReadStat	ArticleID	Unique ID for each article (maps to Article table)
ReadStat	StatName	Name of statistic (*e.g.*, flesch)
ReadStat	StatValue	Value of statistic
NBERStat	NberID	Unique ID for each NBER working paper (maps to NBER table)
NBERStat	StatName	Name of statistic
NBERStat	StatValue	Value of statistic

CSV files

To facilitate future third-party research, all tables and data in `read.db`—with the exception of citation counts, which are proprietary to Web of Science—have also been saved as CSV files in the `0-data/fixed/read_excluding_confidential-csv` directory.

Other datasets

A small number of figures and tables in Hengel (2022) are generated from data contained in `introduction_text.txt`, `readability_corr.txt` and `JEL.csv`. Their contents and provenance are described below and in Table 2.

- **`introduction_text.txt`**. This file contains the first paragraph of text to come after a heading explicitly titled "Introduction" in NBER working papers eventually published in top-four journals. Data are used to generate Figure D.2 in Appendix D.2 in Hengel (2022). Textual data kindly provided by Henrik Kleven and Dana Scott.
- **`readability_corr.txt`**. This file contains coefficients of correlations between the five readability scores used in Hengel (2022) and alternative measures of text difficulty. These figures are from the studies listed in Appendix D.4. They are used to produce the top graphic of Figure D.1 in Appendix D.1.
- **`JEL.csv`**. The file `JEL.csv` categorises all tertiary *JEL* codes as either theory/methodology, empirical or other. Categorisation was done manually by me. Data in `JEL.csv` are used to generate Table

C.1 and construct the theory/methodology, empirical and other dummies described in Appendix C.

Table 2: Description of variables in other datasets

File name	Column name	Description
introduction_text.txt	NberID	Unique ID for each NBER working paper
introduction_text.txt	Text	First paragraph of text
readability_corr.txt	StatName	Name of readability statistic
readability_corr.txt	Correlation	Coefficient of correlation
readability_corr.txt	Test	Name of alternative measure of text difficulty
readability_corr.txt	TestType	Type of alternative measure
readability_corr.txt	Source	BibTeX label for source study (*e.g.*, SurnameYYYY)
readability_corr.txt	Note	Notes on calculations, *etc.*
JEL.csv	JEL	Tertiary *JEL* code
JEL.csv	Description	Long name of *JEL* code
JEL.csv	Type	Classification (empirical, theory or other)

Code

Software requirements

- Python 3.9.0
  - `pandas` (1.2.3)
  - `Textatistic` (0.0.1)
  - Please see below for instructions on how to install these packages.
- R 4.1.2
  - `RSQLite` (2.2.9)
  - `tidyverse` (1.3.1)
  - `haven` (2.4.3)
  - `remotes` (2.4.2)
  - `lexicon` (1.3.1, from https://github.com/trinker/readability)
  - `textclean` (0.9.7, from https://github.com/trinker/readability)
  - `textshape` (1.7.4, from https://github.com/trinker/readability)
  - `syllable` (0.2.0, from https://github.com/trinker/readability)
  - `readability` (0.1.2, from https://github.com/trinker/readability)
  - The file `2-update-readability.R` installs the latest version of all dependencies.
- Stata 17.0
  - `ftools` (2.37.0)
  - `estout` (3.24)
  - `psmatch2` (4.0.12)
  - `xtabond2` (3.7.0)
  - `listtex` (25 September 2009)
  - `reghdfe` (5.7.3)
  - `binscatter` (7.02)
  - `distinct` (1.2.1)
  - `labutil` (1.0.0)
  - `coefplot` (1.8.5)
  - `wordwrap` (0.2, from https://mloeffler.github.io/stata/wordwrap)
  - The file `4-master.do` locally installs the latest version of all dependencies.
- Mathematica 12.3.1

Optional portions of the code use bash scripting and WolframScript; instructions for installing the latter are provided below.

Instructions to replicators

To generate all figures and tables in Hengel (2022), first navigate to the project's root directory and then copy `4-confidential-data-not-for-publication/read.db` to `data/fixed/read.db`.[1] Next, execute the following five steps.

1. Run `1-update-textatistic.py` in Python.
2. Run `2-update-readability.R` in R.
3. Run `3-export-raw-data.py` in Python.
4. Run `4-master.do` in Stata.
5. Execute `Figure-3.nb` and `Figure-G.2.nb` (both in the `0-code/output` directory) in Mathematica.

Each step can be executed individually by following the steps outlined below. Alternatively, the Bash script `5-master.sh` completes all five steps automatically. To run `5-master.sh`, install the latest version of WolframScript and follow the instructions under the `1-update-textatistic.py` and `4-master.do` headings for installing `Textatistic` and an SQLite driver, respectively. Then, navigate to the project's root directory and issue the following command in a Bash shell:

```
sh 5-master.sh
```

`5-master.sh` was last run on 12 March 2022 on a 4-core Intel-based iMac running MacOS version 11.6.5. Computation took 10 hours, 12 minutes and 20 seconds.

`1-update-textatistic.py`

The Python script `1-update-textatistic.py` calculates readability scores for every article and NBER working paper with an abstract in `read.db` and updates its ReadStat and NBERStat tables with the results. More details on the `Textatistic` program are available on GitHub. Documentation on how it calculates readability scores are available at erinhengel.com.

For `1-update-textatistic.py` to work, you must first install the Python package `Textatistic`.[2] If you're lucky, this can be done in a single step by issuing the following command in your terminal application:

```
pip install textatistic
```

But you're probably not lucky. The problem is the `PyHyphen` dependency; for reasons I do not fully understand, `pip` does not always properly install it before trying to install `Textatistic`.

If you encounter this error, you will need to install both `PyHyphen` and `Textatistic` from source. But don't panic; it's not that hard. Just navigate to the project's root directory and issue the following sequence of commands in your terminal application.[3]

```
cd "0-code/programs/Textatistic/required_packages/PyHyphen-2.0.5/"
sudo python setup.py install
cd ../..
sudo python setup.py install
```

Once `Textatistic` has been properly installed, navigate back to the project's root directory and run the following command in the terminal application.

```
python 1-update-textatistic.py
```

You will be alerted when the ReadStat and NBERStat tables in `read.db` have been successfully updated with the newly calculated readability statistics.

`1-update-textatistic.py` was last run on 16 January 2022 on a 4-core Intel-based iMac running MacOS version 11.6.3. Computation took 2 minutes and 7 seconds.

---

[1] `CiteCount` *is proprietary to Web of Science and are included here for replication purposes only; please do not distribute these data or publish online.*

[2] `1-update-textatistic.py` also requires the `re`, `sys` and `sqlite3` (3.32.3) libraries, which are automatically installed with Python 3.9.0.

[3] `PyHyphen` may need to be installed using Python 3.7 or earlier.

`2-update-readability.R`

The R script `2-update-readability.R` (R version 4.1.2) calculates readability scores using the R [readability package](). To run it, open R, set the current working directory as the project directory and issue the following command:

```
source("2-update-readability.R")
```

`2-update-readability.R` first installs the latest version of `RSQLite`, `tidyverse`, `haven` and `remotes` from CRAN. It then installs `readability`, `syllable`, `textshape`, `textclean` and `lexicon` from [GitHub](). Once these packages are installed and loaded, it connects to the `read.db` SQLite database, fetches published article and NBER abstracts, calculates readability scores and exports the results to `readstat.dta` and `nberstat.dta` in the `0-data/generated` directory. Finally, it reads in `introduction_text.txt`, calculates readability scores and exports the result to `articlestat.dta`, also in the `0-data/generated` directory.

`2-update-readability.R` was last run on 7 February 2022 on a 4-core Intel-based iMac running MacOS version 11.6.4 Computation took 2 minutes and 14 seconds to run.

`3-export-raw-data.py`

The Python script `3-export-raw-data.py` (Python version 3.9.0) exports raw data from `read.db` as csv files for futher processing in Stata. To run it, you will need to install the `pandas` module (version 1.2.3). Usually, this can be done with `pip` by issuing the following command in your terminal application.[4]

```
pip install pandas
```

Once `pandas` has been properly installed, navigate to the project's root directory and run the following command in the terminal application.

```
python 3-export-raw-data.py
```

The raw data generated by `3-export-raw-data.py` are saved in the `0-data/generated` directory.

`3-export-raw-data.py` was last run on 10 March 2022 on a 4-core Intel-based iMac running MacOS version 11.6.5. Computation took less than a minute.

`4-master.do`

The Stata script `master.do` (Stata version 17.0) generates all figures and tables in Hengel (2022) with the exception of Figure 3 and Figure G.2 (see below).

To run `4-master.do`, open a Stata terminal, navigate to the project's root directory and issue the following command:

```
do 4-master.do
```

`4-master.do` first installs several third-party packages from SSC (`ftools`, `estout`, `psmatch2`, `xtabond2`, `listtex`, `reghdfe`, `binscatter`, `distinct`, `labutil` and `coefplot`) and `wordwrap` from [GitHub](). It then copies the ado, scheme, colors and `estout` definition files in the `0-code/programs/stata` directory into your Stata personal ado directory. (Alternatively, manually load these files into Stata before running `4-master.do` and comment out lines 25–29.) It then transforms the raw data (results are saved in `0-data/generated`) and executes the Stata do files in the `0-code/output` directory. Estimation results are either saved as LaTeX output in the `0-tex/generated` directory or as image files in the `0-images/generated` directory. A log of all output is saved in the `0-log` directory as `YYYY-MM-DD-HH-MM-SS.smcl`.

`4-master.do` was last run on 11 March 2022 on a 4-core Intel-based iMac running MacOS version 11.6.5 Computation took 11 hours 13 minutes and 16 seconds.

Create Mathematica graphs

Figures 3 and G.2 in Hengel (2022) were created using Mathematica (version 12.3.1). To generate them, follow the three steps below:

---

[4] `3-export-raw-data.py` also requires the `sqlite3` (3.32.3) library, which is automatically installed with Python 3.9.0.

1. Navigate to the `0-code/output` directory and open the files `Figure-3.nb` and `Figure-G.2.nb` in Mathematica.
2. Select "Evaluate Notebook" from the Evaluation dropdown menu.
3. Click "Yes" to run the initialisation cells.

`Figure-3.nb` generates `Figure-3.png`; `Figure-G.2.nb` generates `Figure-G.2.png`. Both files are saved in the `0-images/generated` directory.

`Figure-3.nb` and `Figure-G.2.nb` were last run on 16 January 2022 on a 4-core Intel-based iMac running MacOS version 11.6.3. Combined computation took less than a minute.

References

American Economic Association (2022). *EconLit [database].* American Economic Association, Nashville Tennessee. https://www.aeaweb.org/econlit/ (last accessed January 2019).

Clarivate (2022). *Web of Science [database].* Clarivate, London https://www.webofknowledge.com/ (last accessed January 2018).

Hengel, E. (2022). "Publishing while female: Are women held to higher standards? Evidence from peer review." Mimeo.

Kleven, H. (2018). "Language trends in public economics". https://www.henrikkleven.com/uploads/3/7/3/1/37310663/languagetrends_slides_kleven.pdf (last accessed 2 December 2018).

Kleven, H. and D. Scott (2018). *Language trends in public economics [database].*