25 January 2021

README for "COVID-19 Shifted Patent Applications toward Technologies that Support Working from Home" by Nicholas Bloom, Steven J. Davis and Yulia Zhestkova

Our full replication package is at
https://www.dropbox.com/s/rhqdhxmijn4t7il/Replication%20package.zip?dl=0**.**

1. Accessing raw patent application data

Raw XML files of patent applications are available at https://bulkdata.uspto.gov, USPTO Bulk Data Storage System (BDSS) HTTPS environment for no charge. New patent applications are released weekly on Thursday and stored as a separate file. There is no one aggregate data file with all accumulated historic patent application. Hence, researchers interested in the analysis of patent application dynamics need to manually download weekly data files and combine them together.

We used two types of data products provided by USPTO BDSS:

- Patent Application Bibliographic (Front Page) Data
- Patent Application Full Text Data (No Images)

For both data products, we used raw patent application files published from January 7th 2010 till December 24th 2020. All plots (Figure 1 and Figure 2) were constructed using counts derived from the full patent text data. In particular, we used the identification of Working From Home technologies in the underlying data that is based on the analysis of full description of a patent.

2. Required software

- Stata 14 or higher
- Python 3.7.4, including such libraries as:
    - pandas
    - numpy
    - re
    - os
- Microsoft Excel (optional)

3. Downloading, parsing and processing Patent Application Bibliographic data

This section describes how we build a dataset of patent applications with flags that indicate Working From Home (WFH) technologies based on the content of their title and abstract. The final dataset includes such variables as patent application number, date of patent application filing, date of patent application publishing, number of terms from WFH dictionary in the title and abstract of a patent, and patent assignee. The code generating this dataset from raw patent

files is written and run in Python and is included in the file "*wfh_patents_bibliographic_data_AER.ipynb*".

3.1. Downloading and storing raw files before running the code

For each of the two data products (Patent Application Bibliographic Data and Patent Application Full Text Data), we created a core home directory. In the code, the depository for Patent Application Bibliographic Data is called "*patent parsing biblio*". It is one of the two core folders where raw data files should be stored after being downloaded. For each year from 2010 to 2020, one needs to create a separate folder in this core directory (eleven empty files total) and name them accordingly, i.e. "2010", "2011", etc.

We downloaded raw patent application files contained in section **Patent Application Bibliographic (Front Page) Data** and saved them in the folder that corresponds to the year when the weekly patent releases were published (for instance, raw files from 2010 should be save in folder "*patent parsing biblio/2010*"). Note that the last release included in our sample is from December 24th, 2020. Every .zip archive of weekly patent application bibliographic data releases contains three files: XML, TXT and HTML. One wants to **keep only XML file** for each weekly application – this is the file containing the data. Every year-folder should have from 52 to 53 raw XML files. One needs to change the format of the XML files to TXT – we did this manually.

Once the raw zip files for each patent application bibliographic weekly releases from January 7th 2010 to December 24th 2020 are downloaded, saved in the directory that corresponds to the year of publishing, only XML files are kept and turned into TXT format, the data is ready to be processed. Code "*wfh_patents_bibliographic_data_AER.ipynb*" parses the data and generates the final dataset *"wfhflag_biblio.csv"* that can be found in the replication package.

4. Downloading, parsing and processing Patent Application Full Text data

This section described a construction of the dataset of patent applications with flags that indicate WFH technologies based on the content of their title and full text of patent description (in contrast to only abstracts that were used in the Bibliographic data discussed above). The final dataset includes such variables as patent application number, date of patent application filing, and number of terms from WFH dictionary in the title and patent description. The code generating this dataset from raw patent files is written in Python and is attached in the file "*wfh_patents_full_text_data_AER.ipynb*".

4.1. Downloading and storing raw files before running the code

The process of downloading and sorting raw files for this dataset is very similar to the one described above. We start with creating a core home directory – in the code, the depository for Patent Application Full Text Data is called "*patent parsing full text*". For each year from 2010

till 2020, one needs to create a separate folder in this core directory (eleven empty files total) and name them accordingly, i.e. “2010”, “2011”, etc.

We downloaded raw patent application files contained in section **Patent Application Full Text Data (No Images)** and saved them in the folder that corresponds to the year when the weekly patent releases were published (for instance, raw files from 2010 should be save in folder “*patent parsing full text/2010*”). Note that the last release included in our sample is from December 24th, 2020. Every .zip archive of weekly patent application full text data releases **contains only one XML** file (note the difference with patent application bibliographic data releases). Every year-folder should have from 52 to 53 raw files. One want to change the format of the XML files to TXT – we did this manually.

Once the raw zip files for each patent application full text weekly releases from January 7th 2010 till December 24th 2020 are downloaded, saved in the directory that corresponds to the year of publishing and XML files are turned into TXT format, the data is ready to be processed. Code “*wfh_patents_full_text_data_AER.ipynb*” parses the data and generates the final dataset *“wfhflag_full_text.csv”* that can be found in the replication package.

5. Codes that create Figure 1 and Figure 2

We used Python to build our initial datasets with WFH technology flags. Once these CSV datasets were created, we used Stata to merge the datasets and create aggregate counts and shares for plots. One can use either Stata or Microsoft Excel to generate Figure 1 and Figure 2.

Stata codes that generate Figure 1 and Figure 2 can be found in the Stata do-file “*Figures generating code.do*”. Comments in the code can guide a researcher on what part of the code creates a merged dataset and format variables used to build the figures (lines 1 - 46), what part generates Figure 1 (lines 50 - 78), Figure 2 (lines 83 - 93), and what output is needed to replicate these figures in Excel. In particular, Excel files “*Figure 1.xlsx*” and “*Figure 2.xlsx*” include the Excel version of the graphs used in the paper.