

README for Todor Parushev's Chilean data

Copyright (C) 2012 Joel H. Levine

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files and illustrative data (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Assumption on language present in the user's environment 8/22/12

The package assumes that Python 2.6 and Numpy for Python 2.6 are already installed. These are available as free downloads. Google them, probably from either SourceForge or Python.org.

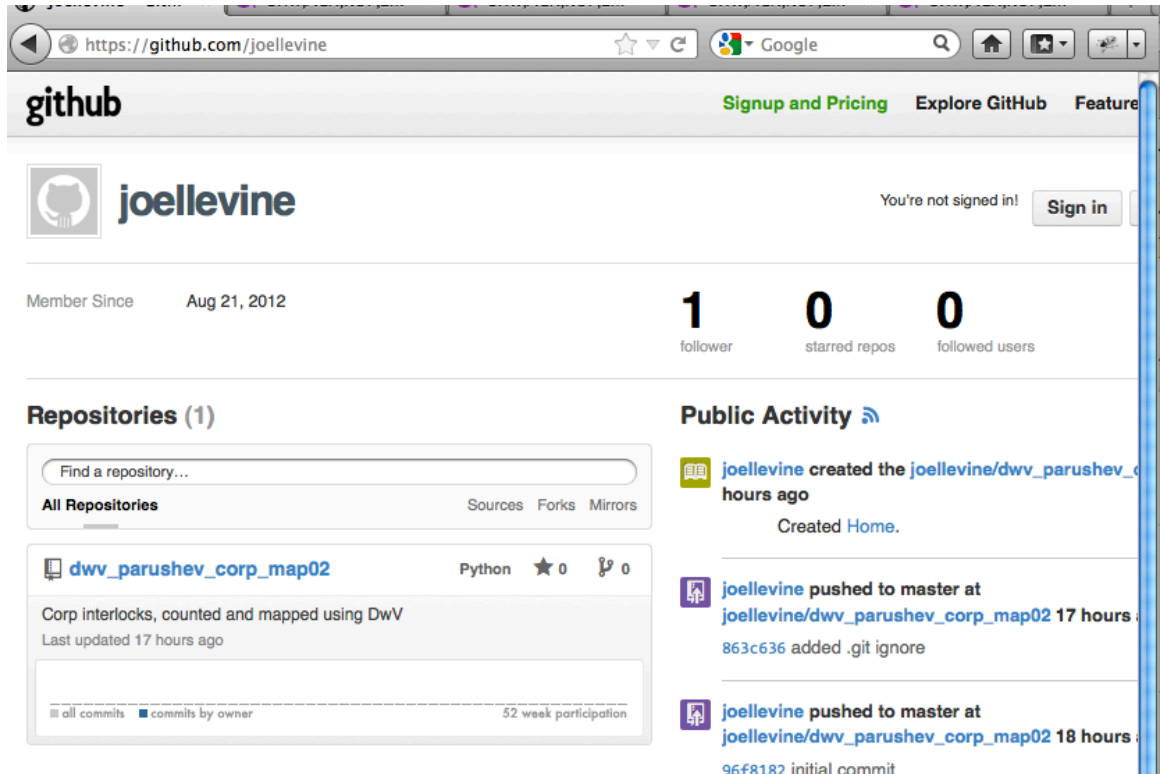
Be careful. Python is increasingly incompatible with Python: So it is easy to have a mismatch between your Python version and your Numpy version, and the automatic installers do not catch these things. Try using "python2.6" as a command where you would ordinarily use "python" as a command. When there is a mismatch the "bugs" can mislead you – directing you to specific lines in the code and asking you to fix them. These lines are (usually) not the guilty party.

I hear no reports of problems when my code is run with Python 2.7 and the Numpy for Python 2.7. I stayed with the Python that was working for me.

The code will not run with Python 3.x

Downloading

Set your browser to accept cookies. Direct the browser to github.com/joellevine



On that web page click/select the repository (known as a “repo”) that you want.

A new window will open. In the new window click/select “ZIP”

github [Signup and Pricing](#) [Explore GitHub](#) [Features](#)

joellevine / dwv_parushev_corp_map02 [★ Star](#)

Code Network Pull Requests 0 Issues 0 Wiki

Corp interlocks, counted and mapped using DwV — [Read more](#)

[Clone in Mac](#) [ZIP](#) [HTTP](#) [Git Read-Only](#) https://github.com/joellevine/dwv_parushev_corp_map02 [Read-Only access](#)

branch: master **Files** Commits Branches 1 Tag

Latest commit to the master branch

added .git ignore

joellevine authored 17 hours ago

dwv_parushev_corp_map02 /

name	age	message
data_prep	18 hours ago	initial commit [joellevine]
example_Parushev01_2dim_p	18 hours ago	initial commit [joellevine]
.gitignore	17 hours ago	added .git ignore [joellevine]
README.md	18 hours ago	Initial commit [joellevine]

The “zip file” will download to your machine. (If it is not immediately visible, look for a folder named “Downloads” on your machine)

Create a folder somewhere, where you want to work with this. Move the zip folder to this work folder. On a Mac, double click the zip file. It will spill its contents into this work folder, in usable form.

The top most folder for what you have downloaded is dwv_parushev_corp_map02. Inside that folder are three folders, a README folder (that you are presently reading), a **data_prep** folder, and an **example_Parushev01_2dim** folder.

Noted below, Parushev collected his data in an Excel spreadsheet, using ID numbers to tell you that “J.J.Jones” and “J.J.Jones, Jr.” are the same person (if they are the same person). His edited spreadsheet was then saved (by Excel) using SAVE AS and specifying the .csv format (comma separated variables).

The main software needs a transformed and edited copy of these data, transformed into counts. The job of the program in the **data_prep** folder is to make this transformation.

Put a copy of this file into the **example_Parushev01_2dim** folder. The job of the program in the **example_Parushev01_2dim** folder is to estimate a map that fits these data.

For now, you also need to hand-edit the “cc” file of counts that is prepared by this program. Open the cc counts file with Excel. Delete rows and columns to reduce it to the data for a single connected set of corporations. Save it with a new name, saving it as a .csv file.

Inside the data_prep folder

Data Preparation

Original Data

Todor has collected his data in file **ChileNetwork no commas.csv**

This is a .csv format file (“comma separated variables”) (Except for the comma formatting it is an ordinary text file.)

Each line of the file records the connection of a person (director) to a corporation (or some identity declared to be a corporation for purposes of this program).

The line includes id numbers as well as textual names for everything. This number is what identifies a unique person or corporation. Your file may show the same person (or corporation) with slightly different spellings. The program knows they are the same person (or corporation) because they have the same id numbers. The process of editing and cleaning the data is a matter of getting these id numbers right.

Pre-processing the data

First a note on “.csv files”. .csv files use the suffix .csv. They can be opened by Excel and edited in Excel. Then Excel can save the file as a .csv file (using “SAVE AS” – unless Excel automatically uses a .csv format for you.)

Data preparation specifically for the **dwv** program requires that these data be reorganized (or reshaped) in several forms. The mapping (later) requires a corporation-to-corporation table. It is a table of counts. And it is saved as a .csv file.

Caution: In your data, do not use any commas or less-than signs, or greater-than signs within the text of your original data file. There should be no visible commas when your file is viewed in Excel. That's why it has "no commas" in its name. If a file has such things in it, edit-replace them in Excel. Replace them with something else.)

Program **Network_Data_Reshape05.py** reorganizes the data into several files including the corporation-to-corporation table that you need. Run the program. It is pre-programmed to seek data in Todor's original data file, using the name for that file that I gave that file.

The several files that the program writes includes two files for which there is no immediate use and a "...cc..." file. The program also puts valuable information on the screen: For each connected set within the full data set, it prints the names of the entities in that connected set (both person names and corporation names).

At the present time the ...cc... file is a corporation to corporation table for *all* the corporations in the original data file.

Before you proceed any further, it is wise to hand-edit this file: Usually there is one large connected set of corporations. Hand-edit the file (in Excel) to remove corporations that are not in this connected set. Here this reduced version of the cc-file is named **Chile.csv**

Inside the *Example Parushev01_2dim_p* Folder

Processing the data with DwV

Now you are prepared for **Example_Parushev01_2dim_p**.

The **Example_Parushev01_2dim_p** folder includes files that you use directly as well as files that the program handles by itself.

In the name of this folder, I have reminded myself that the visualization will be in two dimensions and will use equation "p" (the *power-normal*).

Inside the folder:

Chile.csv is a copy of the file created in the data preparation phase of the work.

Chile_SETUP_2dimscanDwV.txt is a setup file containing commands that will be received by the program. It is "plain text". It is not a **Word** file, although **word**, if asked politely can SAVE AS a .txt file.

generic_csv_pscriptspan14.py is the program you need to run

Starting the Program

If you are going to run this Python code from Python's **IDLE** , then simply double click **generic_csv_pscriptspan14.py** and it will ask you questions.

If you are going to run from a "Terminal" (for Mac OSX or Ubuntu unix) or from a "Command line" (PC), it will be considerably faster.

For Mac, open a "Terminal".

For Mac and Unix, Navigate to **Example_Parushev01_2dim_p**

For Mac and unix, type **python2.6 generic_csv_pscriptspan14.py**

Whichever path you have used to run the program, the program should start up and ask you a question and then request that you navigate to certain files .

First: A question "Continuation or Startup?"

First answer (always): Startup

Second: Navigate to the SETUP file (named **Chile_SETUP_2dimscanDwV.txt**)

Second answer: Occasionally instead of asking you to navigate, it just sits there and stares at you. This usually means that it has opened a navigation window, but left that window "under" some other file that is open on your desktop. Look around for it, closing unnecessary windows. Then navigate to this SETUP file inside this **Example_Parushev01_2dim_p** folder. Double click on it.

Third: Navigate to the data file (named **Chile.csv**)

Running the Program

Sit back. Enjoy the pictures pictures — somewhere on your desktop. This is going to take several hours. It is working. Give it 3 or 4 hours (at least).

On the Mac, command-option-8 will give you a closeup. (If it doesn't, then go into Mac Icon that leads to System Preferences and deal with the Universal Access Preferences.)

Output from the Program