



Principles of Complex Systems, Vols. 1, 2, & 3D  
CSYS/MATH 6701, 6713, & a pretend number  
University of Vermont, Fall 2023  
Assignment 06

"It's like a thought with another thought's hat on."

**Due:** Sunday, October 15, by 11:59 pm

<https://pdodds.w3.uvm.edu/teaching/courses/2023-2024pocsverse/assignments/06/>

*Some useful reminders:*

**Deliverator:** Prof. Peter Sheridan Dodds (contact through Teams)

**Assistant Deliverator:** Chris O'Neil (contact through Teams)

**Office:** The Ether

**Office hours:** See Teams calendar

**Course website:** <https://pdodds.w3.uvm.edu/teaching/courses/2023-2024pocsverse>

**Overleaf:** LaTeX templates and settings for all assignments are available at  
<https://www.overleaf.com/read/tsxfwwmwdgxj>.

---

All parts are worth 3 points unless marked otherwise. Please show all your workingses clearly and list the names of others with whom you ~~conspired~~ collaborated.

For coding, we recommend you improve your skills with Python, R, and/or Julia. The (evil) Deliverator uses (evil) Matlab.

Graduate students are requested to use  $\LaTeX$  (or related  $\TeX$  variant). If you are new to  $\LaTeX$ , please endeavor to submit at least  $n$  questions per assignment in  $\LaTeX$ , where  $n$  is the assignment number.

**Assignment submission:**

Via Brightspace or other preferred death vortex.

---

**Please submit your project's current draft** in pdf format via Brightspace by the same time specified for this assignment. For teams, please list all team member names clearly at the start.

---

Notes for Baby Name analysis:

- You will have the baby name data sets on hand from the previous assignment.
- Recommended: Use the Allotaxonometer for All created by Jonathan St.-Onge and team.

Here's the app (which should be the easiest to use):

<https://allotaxp.vercel.app/>

Git repo:

<https://github.com/jstonge/allotaxp>

Observable HQ version:

<https://observablehq.com/@jstonge/allotaxonometer-4-all>

- Alternatively, if you love Matlab:

<https://gitlab.com/compstorylab/allotaxonometer>.

Note that you will need to separately install the command `epstopdf` (which should be part of a well stocked unix system with  $\text{\LaTeX}$ ).

1.  $(3 + 3 + 3 + 3 + 3 + 3)$

Generate allotaxonographs comparing the following pairs with the stated values of  $\alpha$  (6 plots):

See alternative below if you cannot get any allotaxonometer code to work on your system.

- (a) Baby girl names in 2001 versus baby girl names in 2002 ( $\alpha = 1/12$ )
- (b) Baby boy names in 2001 versus baby boy names in 2002 ( $\alpha = 1/12$ )
- (c) Baby girl names in 1952 versus baby girl names in 2002 ( $\alpha = \infty$ )
- (d) Baby boy names in 1952 versus baby boy names in 2002 ( $\alpha = \infty$ )
- (e) Baby girl names in 1952 versus baby boy names in 1952 ( $\alpha = \infty$ )
- (f) Baby girl names in 2002 versus baby boy names in 2002 ( $\alpha = \infty$ )

Note that the javascript version does not yet have contour lines which are needed to guide the choice of  $\alpha$ . Comparing baby name distributions in adjacent years (e.g., baby girl names in 2001 and 2002) will require an  $\alpha$  close to 0. Here, we're going with  $1/12$ . But in general, if you can see contour lines, you can choose an  $\alpha$  that fits best.

Online appendices for the main allotaxonomy papers is here:

<http://compstorylab.org/allotaxonomy/>.

More information for the Matlab version:

The gitlab repository:

<https://gitlab.com/compstorylab/allotaxonometer/>

For example baby names code, look through the main script here:

<https://gitlab.com/compstorylab/allotaxonometer/figures/babynames/figures/>

See if you can get this script to run as is.

Contains overview, examples, links to papers, figure-making code, etc.

**Alternative:**

Using rank-turbulence divergence with  $\alpha = 0$  and  $\infty$ , list the top 30 contributing baby names for the four comparisons listed above.

Indicate which year each contributing baby name comes from in parentheses.

For ordering, you do not need to compute RTD in full but rather just the core structure:

$$\left| \frac{1}{[r_{\tau,1}]^\alpha} - \frac{1}{[r_{\tau,2}]^\alpha} \right|. \quad (1)$$

Recall that for  $\alpha = 0$  and  $\alpha = \infty$ , the essential core structure becomes:

$$\left| \ln \frac{r_{\tau,1}}{r_{\tau,2}} \right| \text{ and } \max_{\tau} \left\{ \frac{1}{r_{\tau,1}}, \frac{1}{r_{\tau,2}} \right\}. \quad (2)$$

2. Code up Simon's rich-gets-richer model.

Show Zipf distributions for  $\rho = 0.10, 0.01, \text{ and } 0.001$ . and perform regressions to test  $\alpha = 1 - \rho$ .

Run the simulation for long enough to produce decent scaling laws (recall: three orders of magnitude is good).

Averaging over simulations will produce cleaner results so try 10 and then, if possible, 100.

Note the first mover advantage.

3. (3 + 3 + 3 points) For Herbert Simon's model of what we've called Random Competitive Replication, we found in class that the normalized number of groups in the long time limit,  $n_k$ , satisfies the following difference equation:

$$\frac{n_k}{n_{k-1}} = \frac{(k-1)(1-\rho)}{1+(1-\rho)k} \quad (3)$$

where  $k \geq 2$ . The model parameter  $\rho$  is the probability that a newly arriving node forms a group of its own (or is a novel word, starts a new city, has a unique flavor, etc.). For  $k = 1$ , we have instead

$$n_1 = \rho - (1-\rho)n_1 \quad (4)$$

which directly gives us  $n_1$  in terms of  $\rho$ .

- (a) Derive the exact solution for  $n_k$  in terms of gamma functions and ultimately the beta function.

- (b) From this exact form, determine the large  $k$  behavior for  $n_k$  ( $\sim k^{-\gamma}$ ) and identify the exponent  $\gamma$  in terms of  $\rho$ . You are welcome to use the fact that  $B(x, y) \sim x^{-y}$  for large  $x$  and fixed  $y$  (use Stirling's approximation or possibly Wikipedia).

Note: Simon's own calculation is slightly awry. The end result is good however.

**Hint—Setting up Simon's model:**

<http://www.youtube.com/watch?v=OTzI5J5W1K0>

The hint's output including the bits not in the video:

P.O.C.S 2013-09-23

$$\frac{n_k}{n_{k-1}} = \frac{(k-1)(1-\rho)}{1+(1-\rho)k}$$

$$n_k = \left[ \frac{(k-1)(1-\rho)}{1+(1-\rho)k} \right] \left[ \frac{(k-2)(1-\rho)}{1+(1-\rho)(k-1)} \right] n_{k-2} \dots \left[ \frac{(2)(1-\rho)}{1+(1-\rho)2} \right] n_2 \left[ \frac{(1)(1-\rho)}{1+(1-\rho)1} \right] n_1$$

$\Gamma(k) = (k-1)!$

$$\Gamma(x+1) = x \Gamma(x)$$

$x = n+1 \quad \Gamma(n+1) = n \Gamma(n) = \dots = n! \quad \Gamma(1) = 1$

example  $0 < z < 1$

$$(1+zk)(1+z(k-1)) \dots (1+z1)$$

$$= z^k \left( \frac{1}{z} + k \right) \left( \frac{1}{z} + k-1 \right) \dots \left( \frac{1}{z} + 1 \right) = z^k \frac{\left( \frac{1}{z} + k \right) \left( \frac{1}{z} + k-1 \right) \dots}{\frac{1}{z} \cdot \left( \frac{1}{z} - 1 \right) \left( \frac{1}{z} - 2 \right) \dots}$$

differ by 1

$$= z^k \frac{\Gamma\left(\frac{1}{z} + k + 1\right)}{\Gamma\left(\frac{1}{z} + 1\right)}$$

4. What happens to  $\gamma$  in the limits  $\rho \rightarrow 0$  and  $\rho \rightarrow 1$ ? Explain in a sentence or two what's going on in these cases and how the specific limiting value of  $\gamma$  makes sense.