Using Automatic Detection and Characterization to Measure Educational Impact of nanoHUB

Nathan Denny, George Bunch Adams III, Gerhard Klimeck,
Krishna Madhavan, Swaroop Samek, Michael Zentner

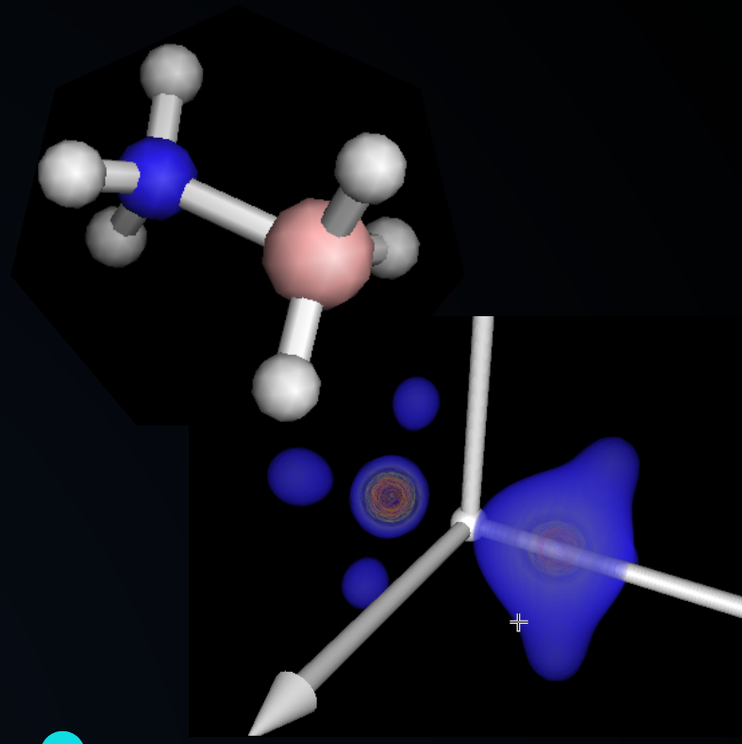
PURDUE
UNIVERSITY

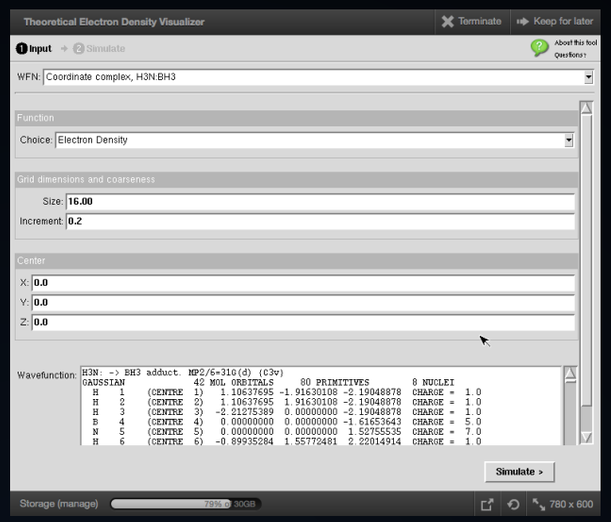
1998

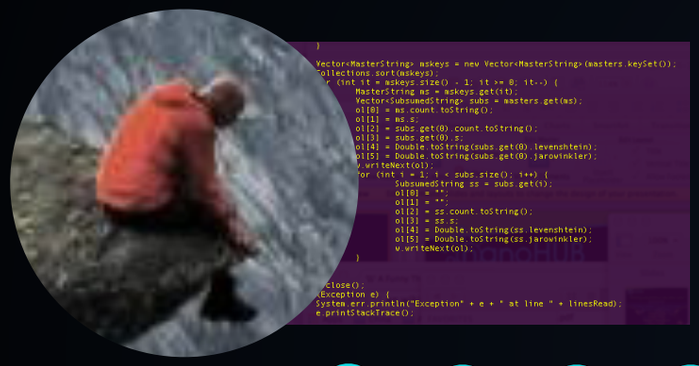
```
                }

                Vector<MasterString> mskeys = new Vector<MasterString>(masters.keySet());
                Collections.sort(mskeys);
                for (int it = mskeys.size() - 1; it >= 0; it--) {
                        MasterString ms = mskeys.get(it);
                        Vector<SubsumedString> subs = masters.get(ms);
                        ol[0] = ms.count.toString();
                        ol[1] = ms.s;
                        ol[2] = subs.get(0).count.toString();
                        ol[3] = subs.get(0).s;
                        ol[4] = Double.toString(subs.get(0).levenshtein);
                        ol[5] = Double.toString(subs.get(0).jarowinkler);
                        w.writeNext(ol);
                        for (int i = 1; i < subs.size(); i++) {
                                SubsumedString ss = subs.get(i);
                                ol[0] = "";
                                ol[1] = "";
                                ol[2] = ss.count.toString();
                                ol[3] = ss.s;
                                ol[4] = Double.toString(ss.levenshtein);
                                ol[5] = Double.toString(ss.jarowinkler);
                                w.writeNext(ol);
                        }
                }

                w.close();
        } catch (Exception e) {
                System.err.println("Exception" + e + " at line " + linesRead);
                e.printStackTrace();
        }
```



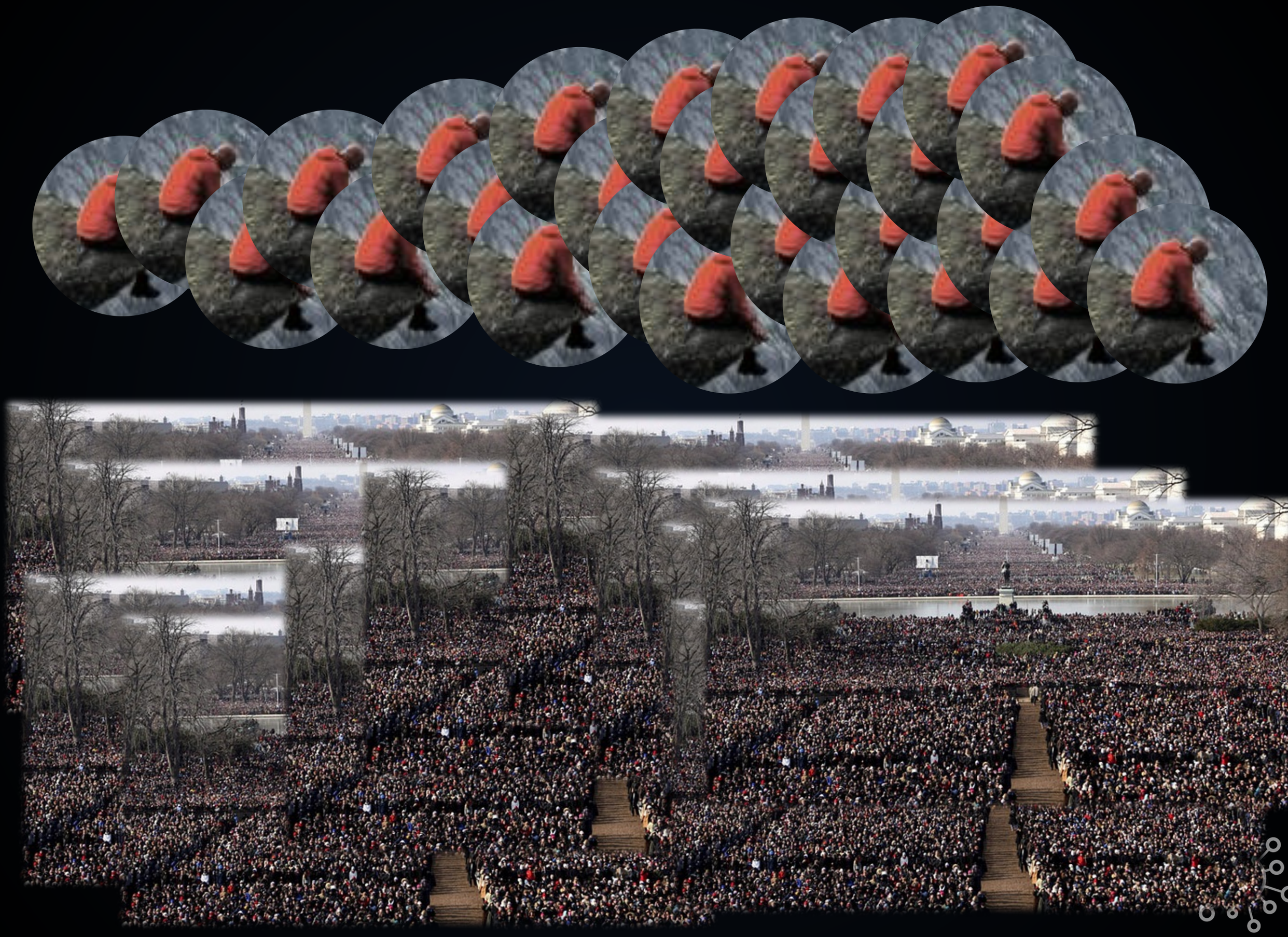


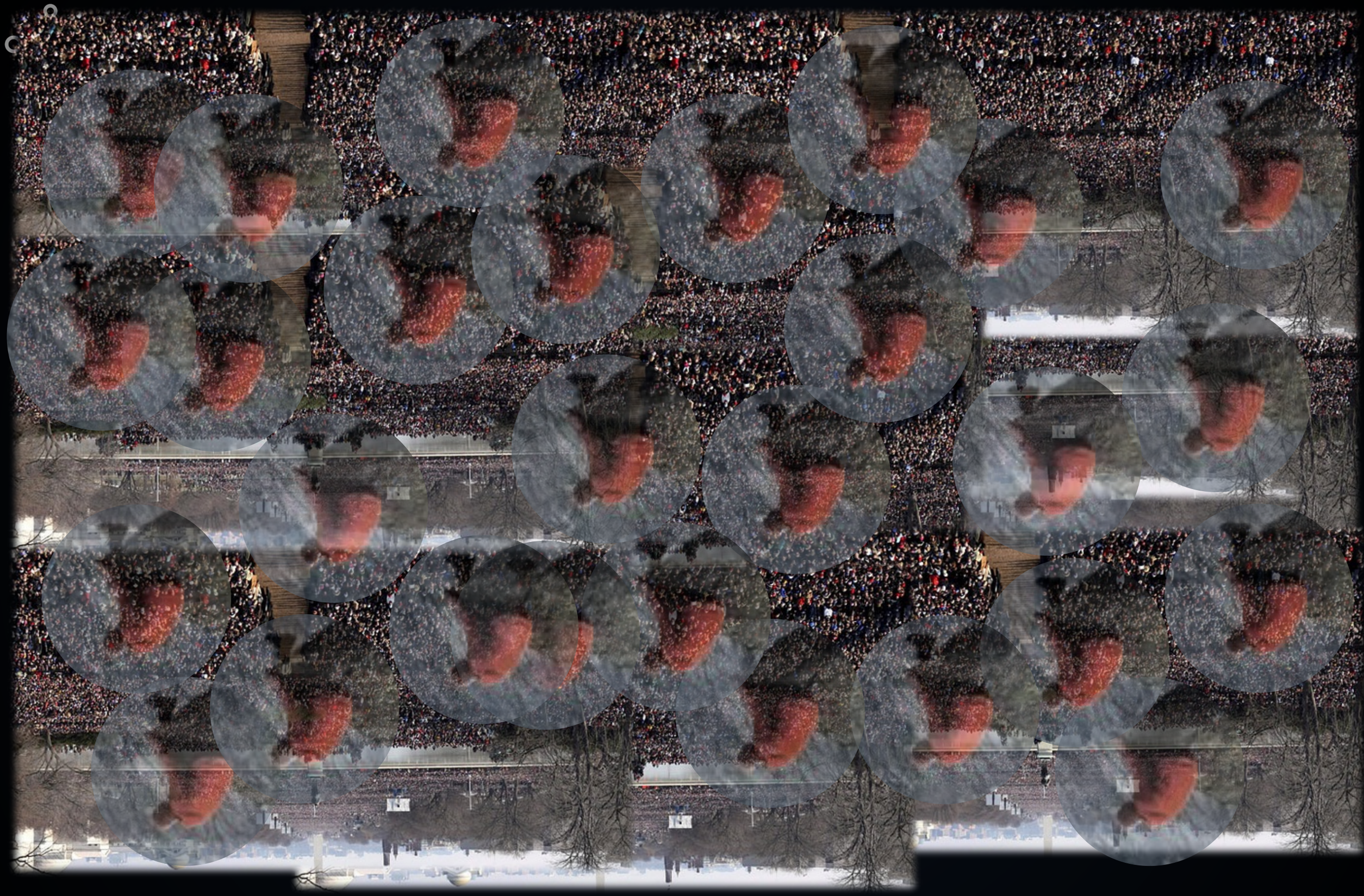


```
}

Vector<MasterString> mskeys = new Vector<MasterString>(masters.keySet());
Collections.sort(mskeys);
for (int it = mskeys.size() - 1; it >= 0; it--) {
        MasterString ms = mskeys.get(it);
        Vector<SubsumedString> subs = masters.get(ms);
        ol[0] = ms.count.toString();
        ol[1] = ms.s;
        ol[2] = subs.get(0).count.toString();
        ol[3] = subs.get(0).s;
        ol[4] = Double.toString(subs.get(0).levenshtein);
        ol[5] = Double.toString(subs.get(0).jarowinkler);
        w.writeNext(ol);
        for (int i = 1; i < subs.size(); i++) {
                SubsumedString ss = subs.get(i);
                ol[0] = "";
                ol[1] = "";
                ol[2] = ss.count.toString();
                ol[3] = ss.s;
                ol[4] = Double.toString(ss.levenshtein);
                ol[5] = Double.toString(ss.jarowinkler);
                w.writeNext(ol);
        }

        close();
} catch (Exception e) {
System.err.println("Exception" + e + " at line " + linesRead);
e.printStackTrace();
```



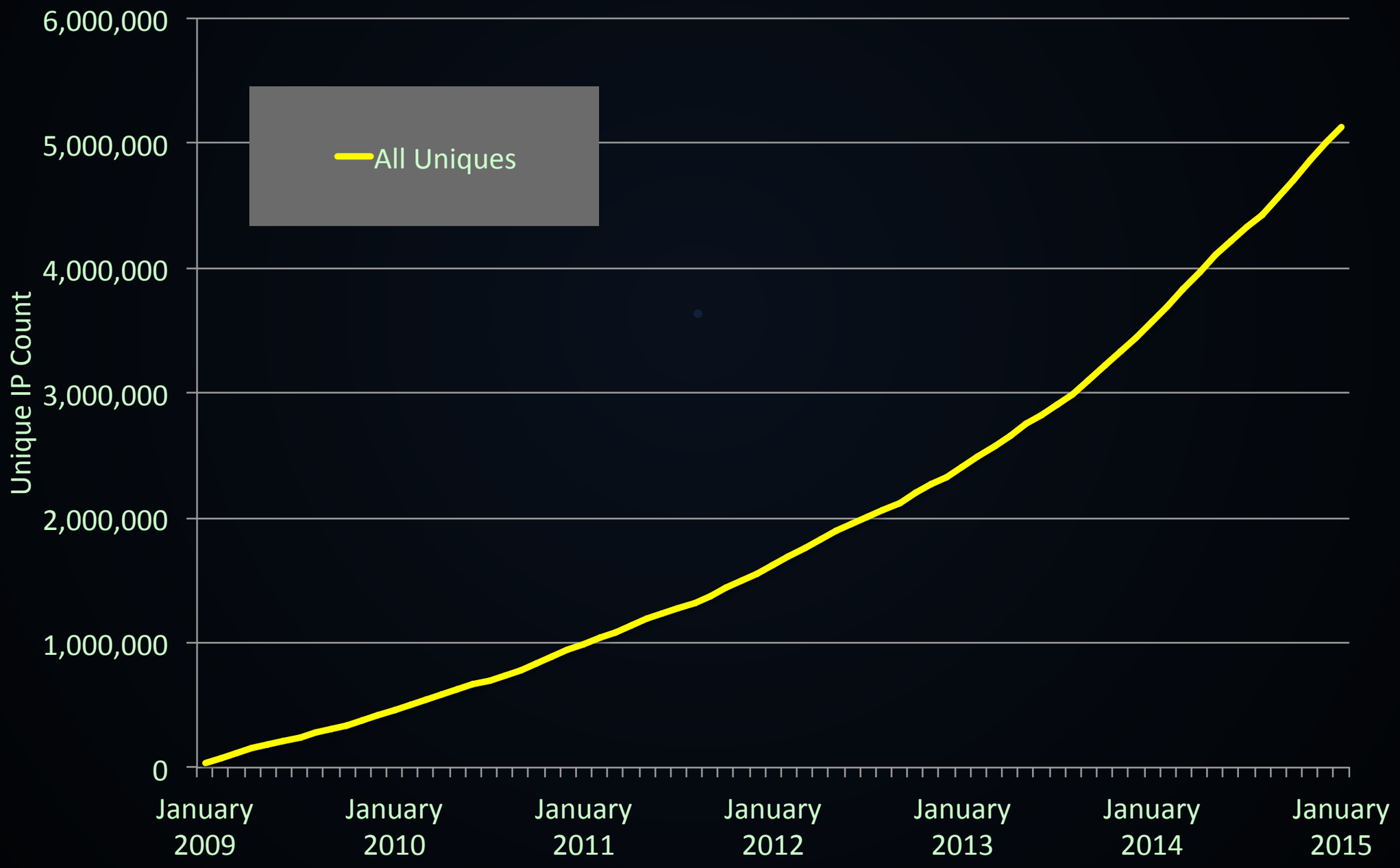


NANOHUB

500+ Simulation Tools
6100+ Other Resources
3900+ Online Presentations
700+ Teaching Materials
35,000+ Students Reached
1,400,000 Unique (re)Visitors Annually
800,000 Simulations Annually

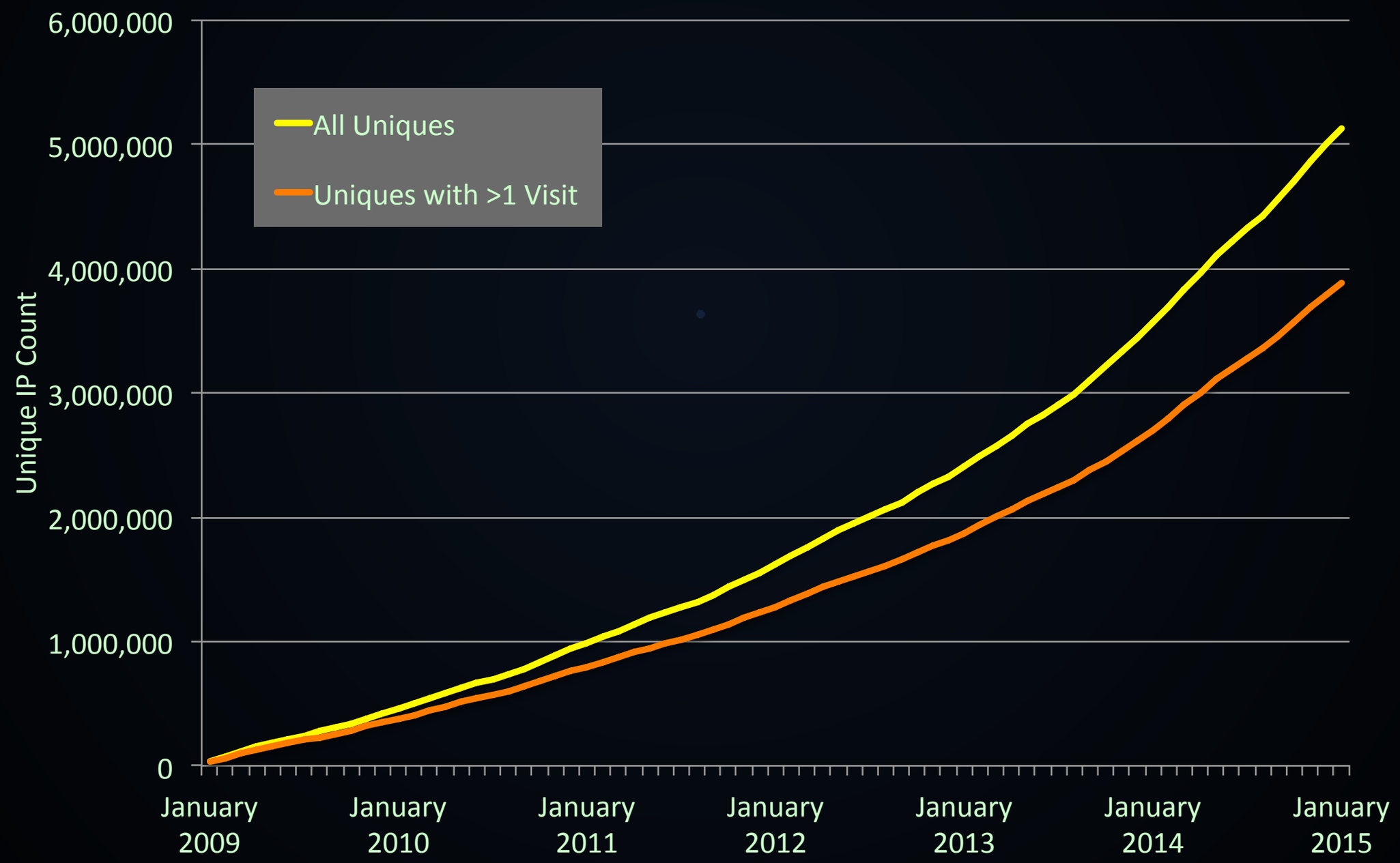
Credit: Nathan Denny

We knew a lot of people came



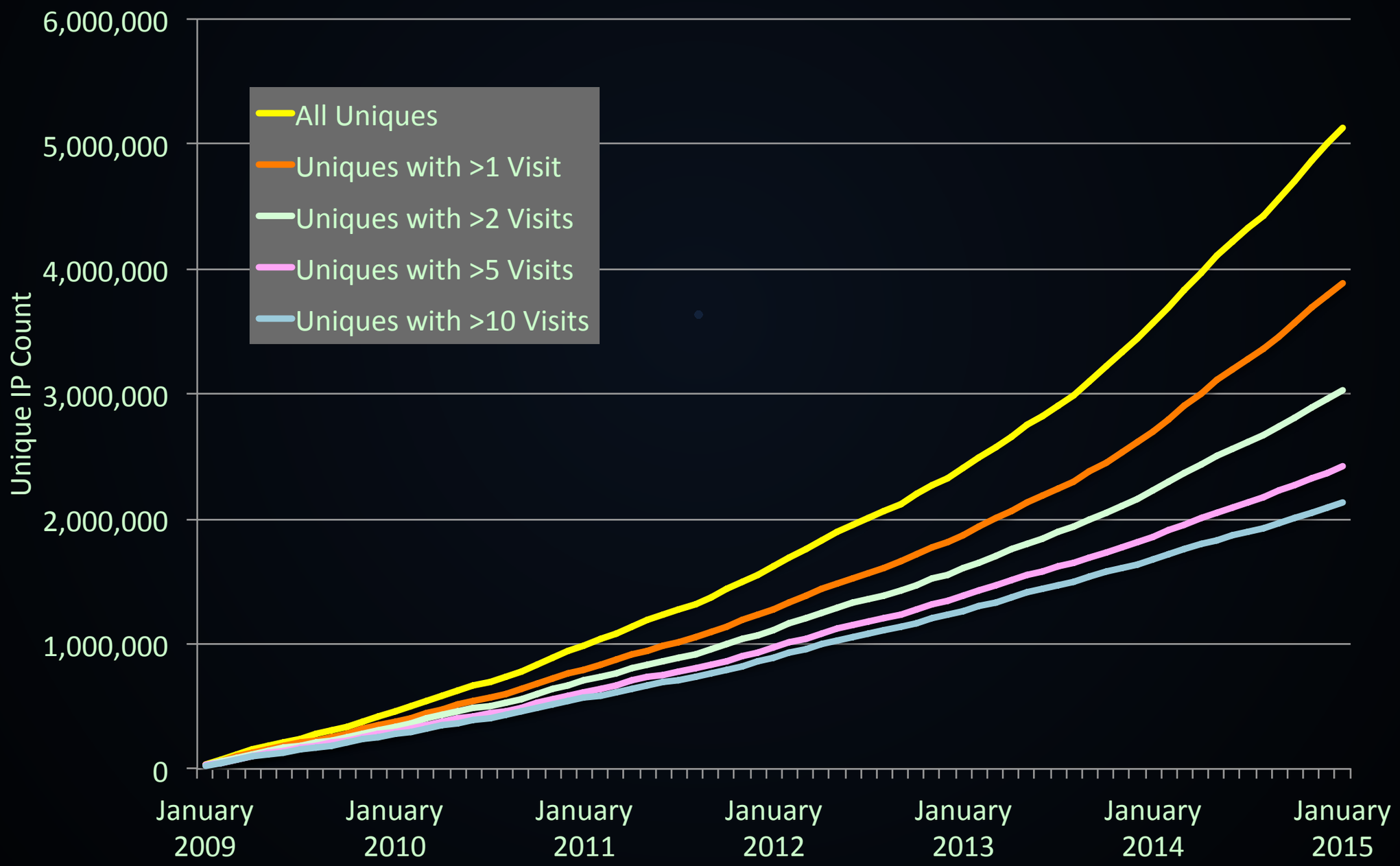
nanoHUB

We knew a lot of people came back



nanoHUB

And a lot of people came back pretty much



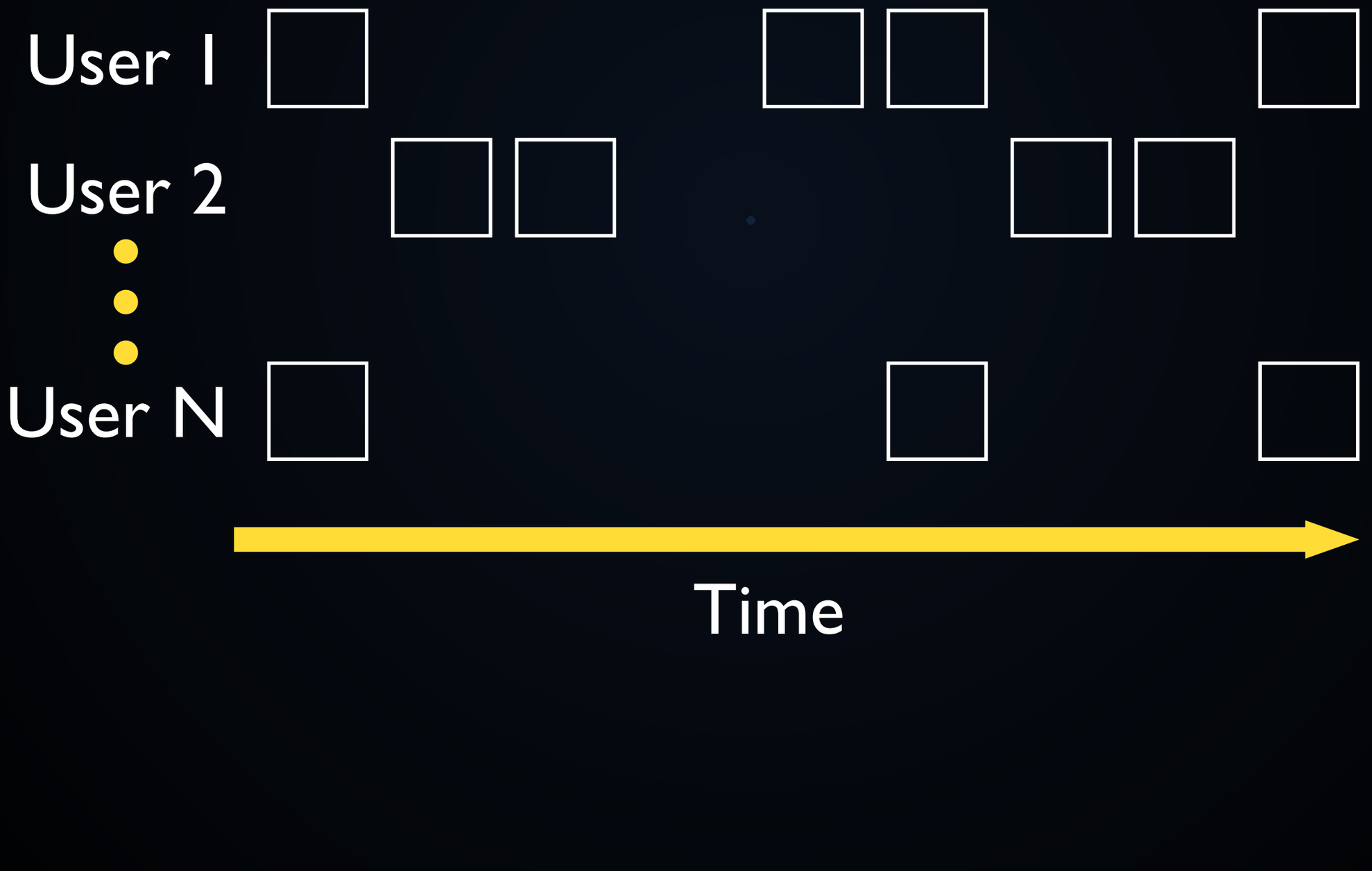
nanoHUB

The Savannah of nanoHUB

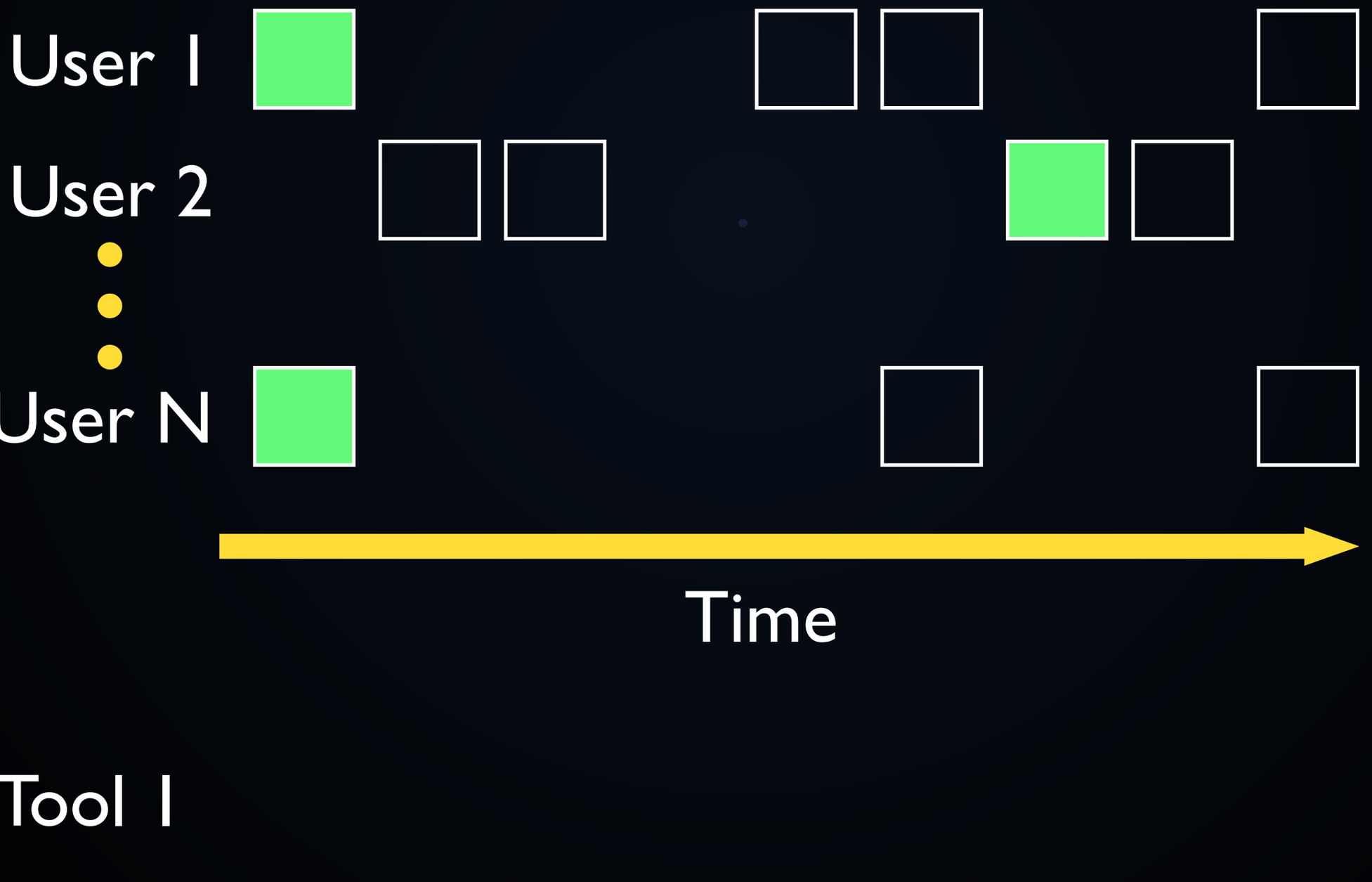
nanoHUB

Credit: Dwight McKay

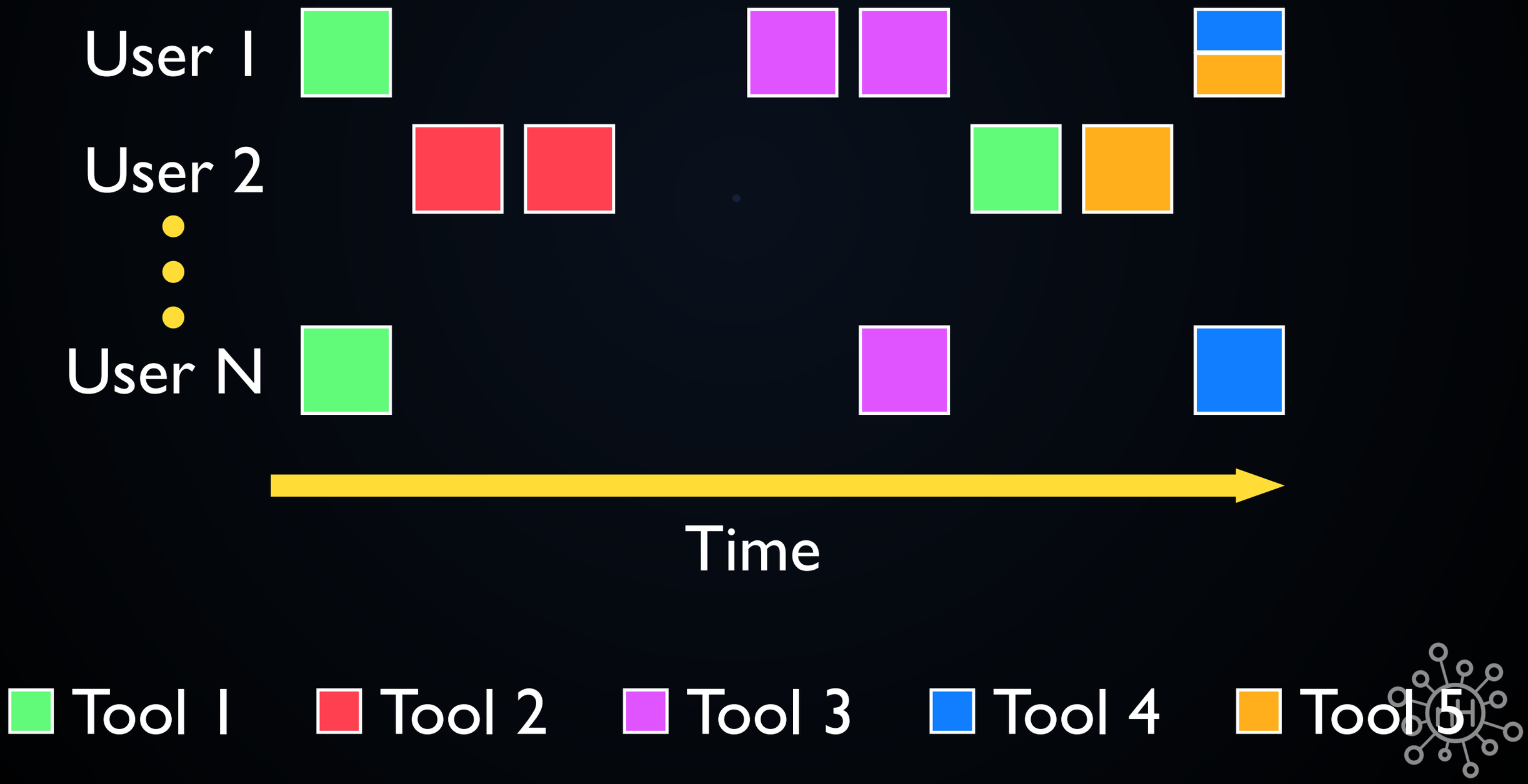
Simulation Visualization



Simulation Visualization



Simulation Visualization



Simulation Visualization

Pattern 1

Pattern 2

Clustering

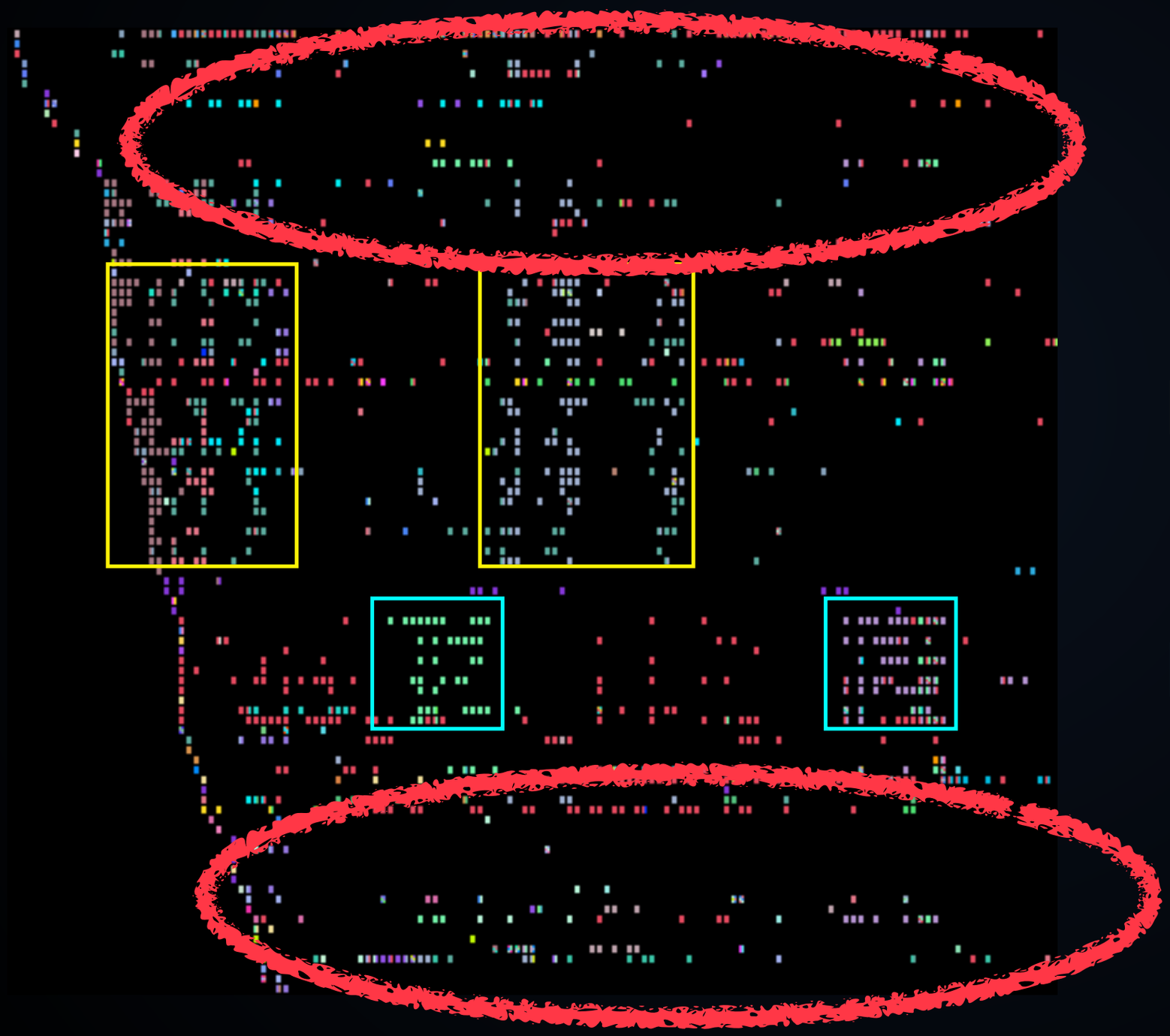
Clustering

Clustering

Clustering

Clustering

Simulation Visualization



Pattern 1

Pattern 2

Clustering

Clustering

Clustering

Clustering

How many do we need?

What do we do with the stuff that doesn’t fit our differentiation characteristics?

Clustering

$$S_{xdt} = p_s(n\text{-}d) \tag{1}$$

$$n > d \wedge t \in U_{xn} \wedge \neg\exists m: (m > d \wedge t \in U_{xm} \wedge m < n) \tag{2}$$

$$N_{xdt} = p_N|n\text{-}d| \tag{3}$$

$$\mathrm{n}: \mathrm{t} \in \mathrm{U_{xn}} \neg\exists \mathrm{m}: (\mathrm{m} \neq \mathrm{d} \wedge \mathrm{t} \in \mathrm{U_{xm}} \wedge |\mathrm{m} - \mathrm{d}| < |\mathrm{n} - \mathrm{d}|) \tag{4}$$

$$I_{xdt} = \frac{p_I\left(D_{\max} - D_{\min}\right)}{\left|\bigcup_d U_{xd}\right|} \tag{5}$$

$$I_{xdt} = \frac{p_I\left(D_{\max} - D_{\min}\right)}{b\min\left(T, \left|\bigcup_d U_{xd}\right|\right) + c\max\left(0, \left|\bigcup_d U_{xd}\right| - T\right)} \tag{6}$$

$$E_{ij} = \sum_{d=D_{\min}}^{D_{\max}} \left(\sum_{t \in U_{jd} - U_{id}} M_{idt} + \sum_{t \in U_{id} - U_{jd}} M_{jdt} \right) \tag{7}$$

$$M_{xdt} = \min(S_{xdt}, N_{xdt}, I_{xdt}) \tag{8}$$

$$C_i = \{P_i\} \bigcup_{x:E_{ix}<H} \{P_x\} \tag{9}$$

$$C_i = C_i \cup C_j \tag{10}$$

$$C = C\text{-}C_j \tag{11}$$

$$i, j: |C_i \cap C_j| \geq q|C_j| \tag{12a}$$

$$\wedge(\neg\exists m: m \neq i \wedge |C_m| > |C_j| \wedge |C_i \cap C_m| \geq q|C_m|) \tag{12b}$$

$$\wedge(\neg\exists k, l: k \neq l \wedge k \neq i \wedge |C_k| > |C_i| \wedge |C_k \cap C_l| \geq q|C_l|) \tag{12c}$$

Similarity

1: add orange
2: add purple

Penalties Assessed

|| = 3.04

Similarity

User 1

User 2

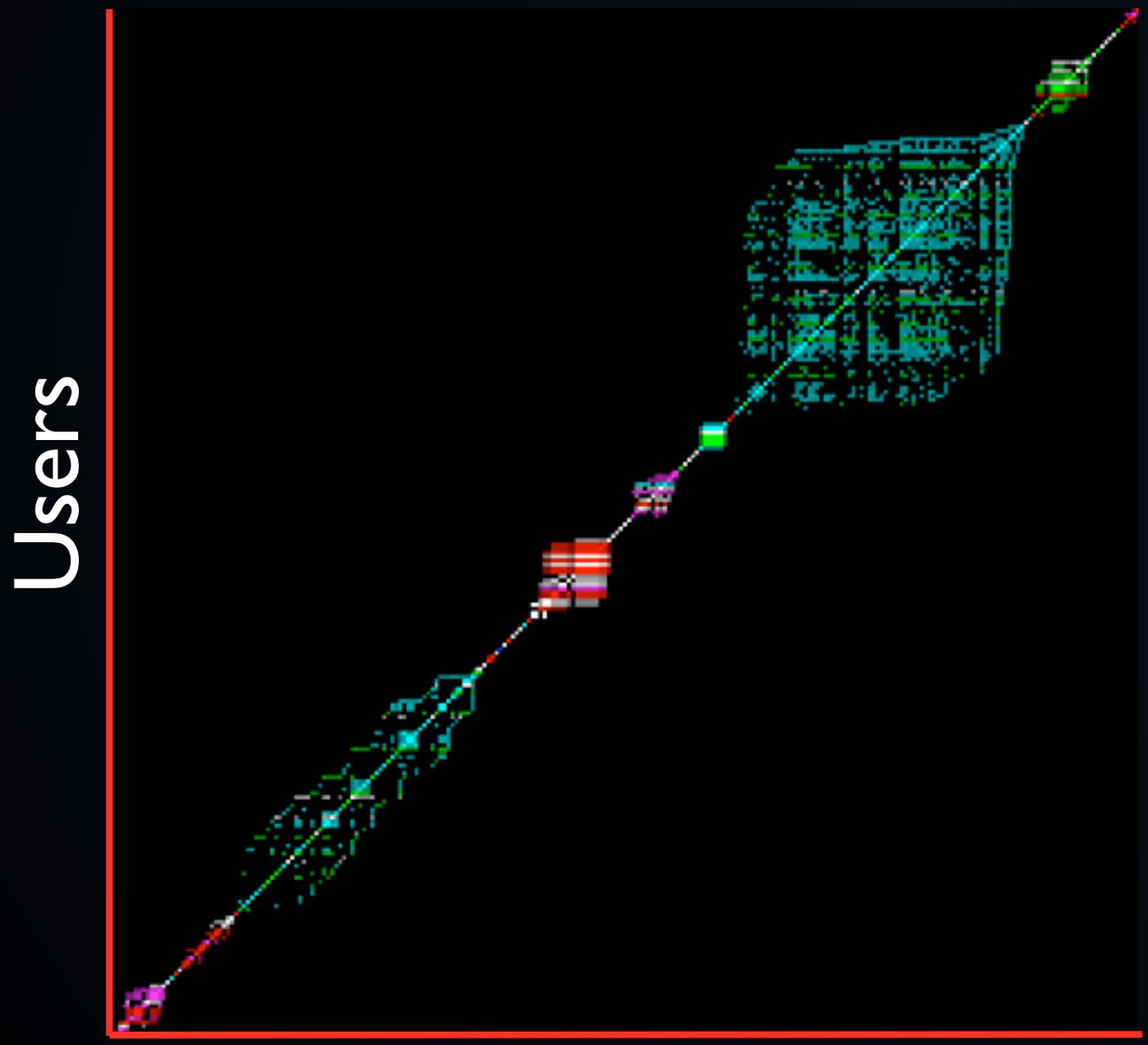
1: move orange
2: move green
3: add red
4: add red
5: add purple
6: add purple
7: add blue

Penalties Assessed

~~IIII~~ II = 25.6

nanoHUB

Clustering by Growth

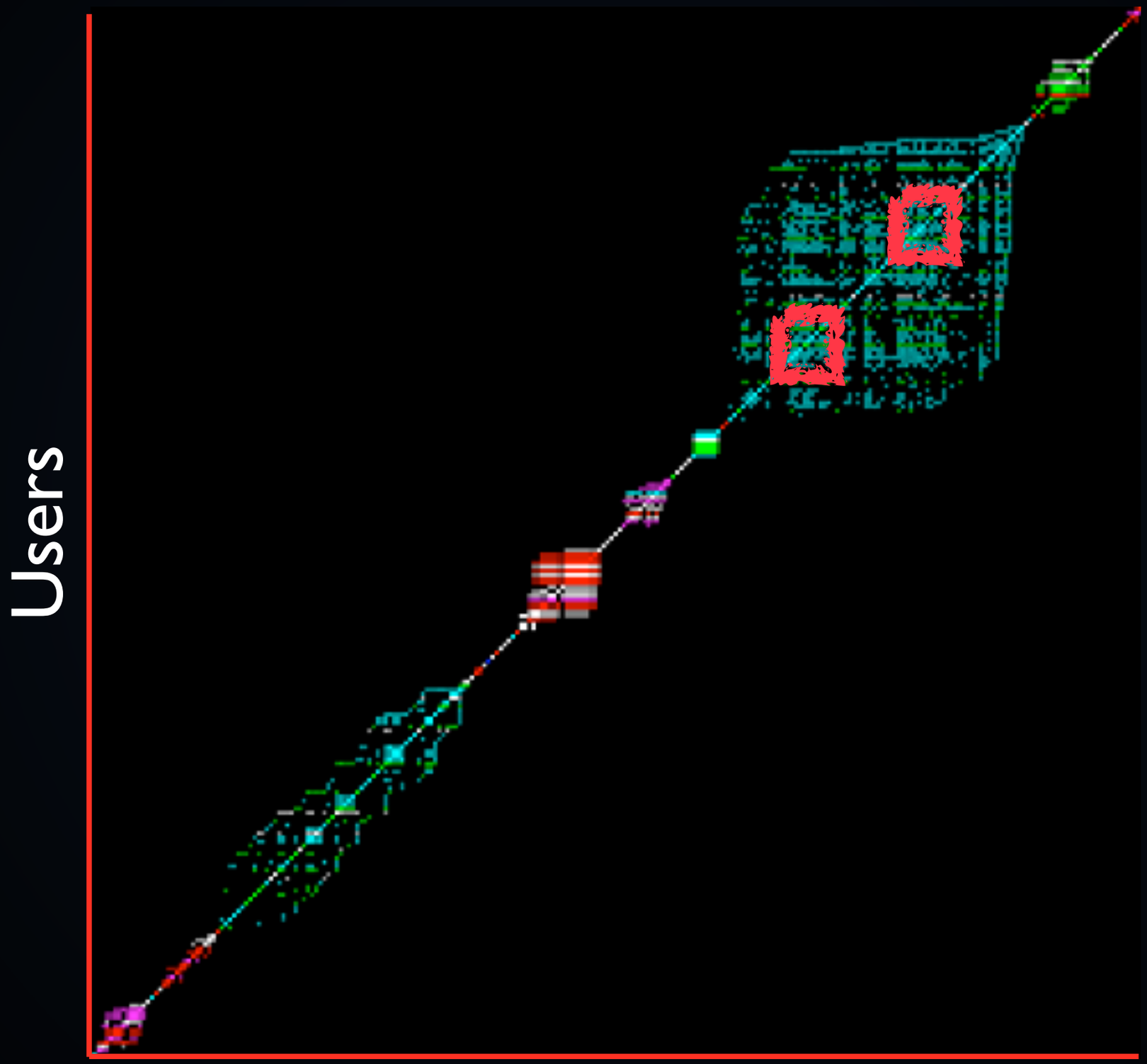


Dot intensity is the strength of User-User similarity

Existence of clusters becomes evident

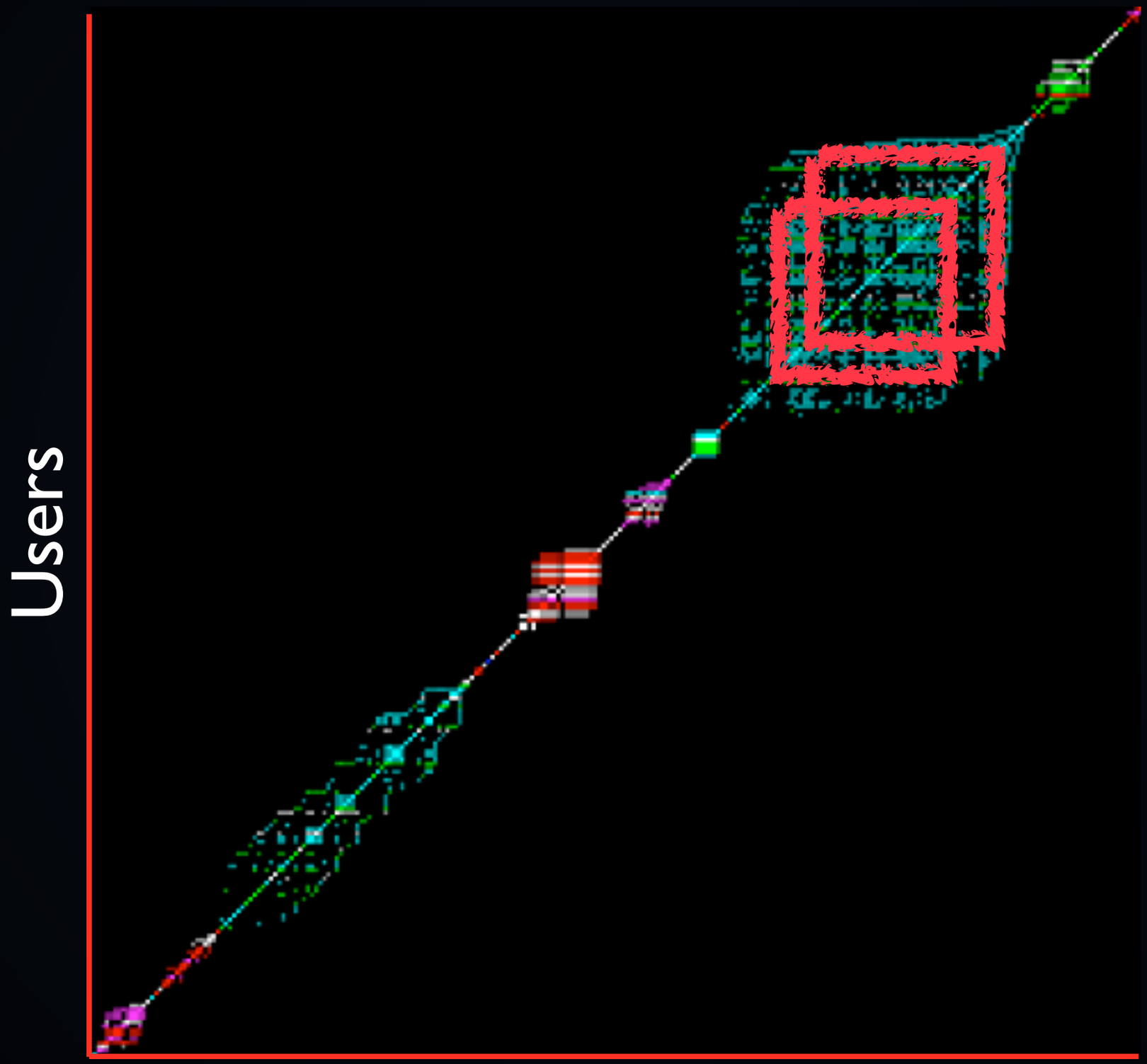
nanoHUB

Clustering by Growth



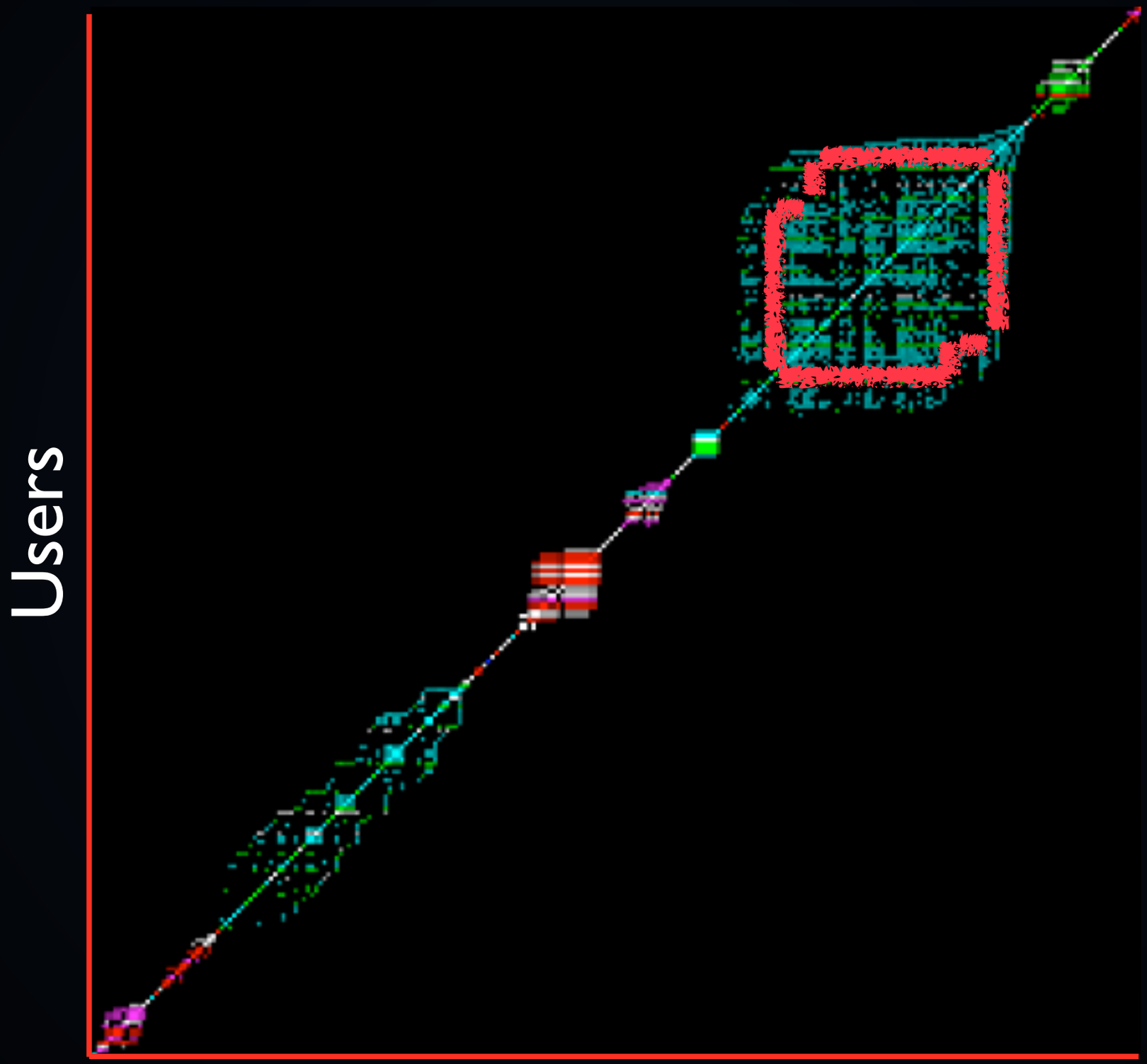
nanoHUB

Clustering by Growth



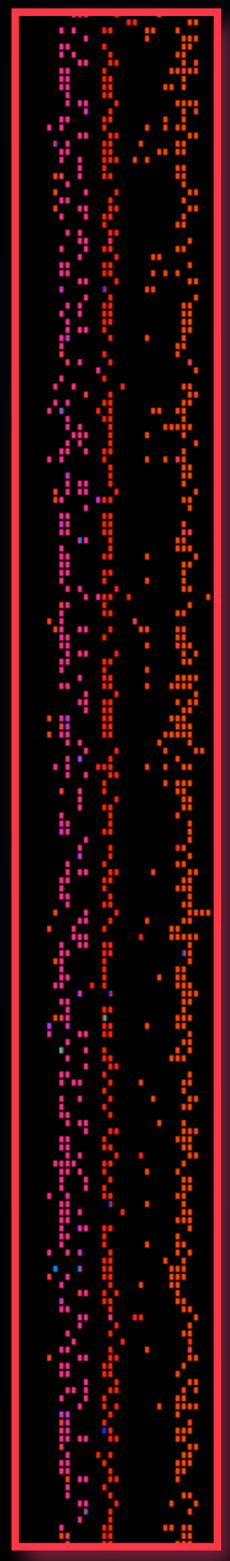
nanoHUB

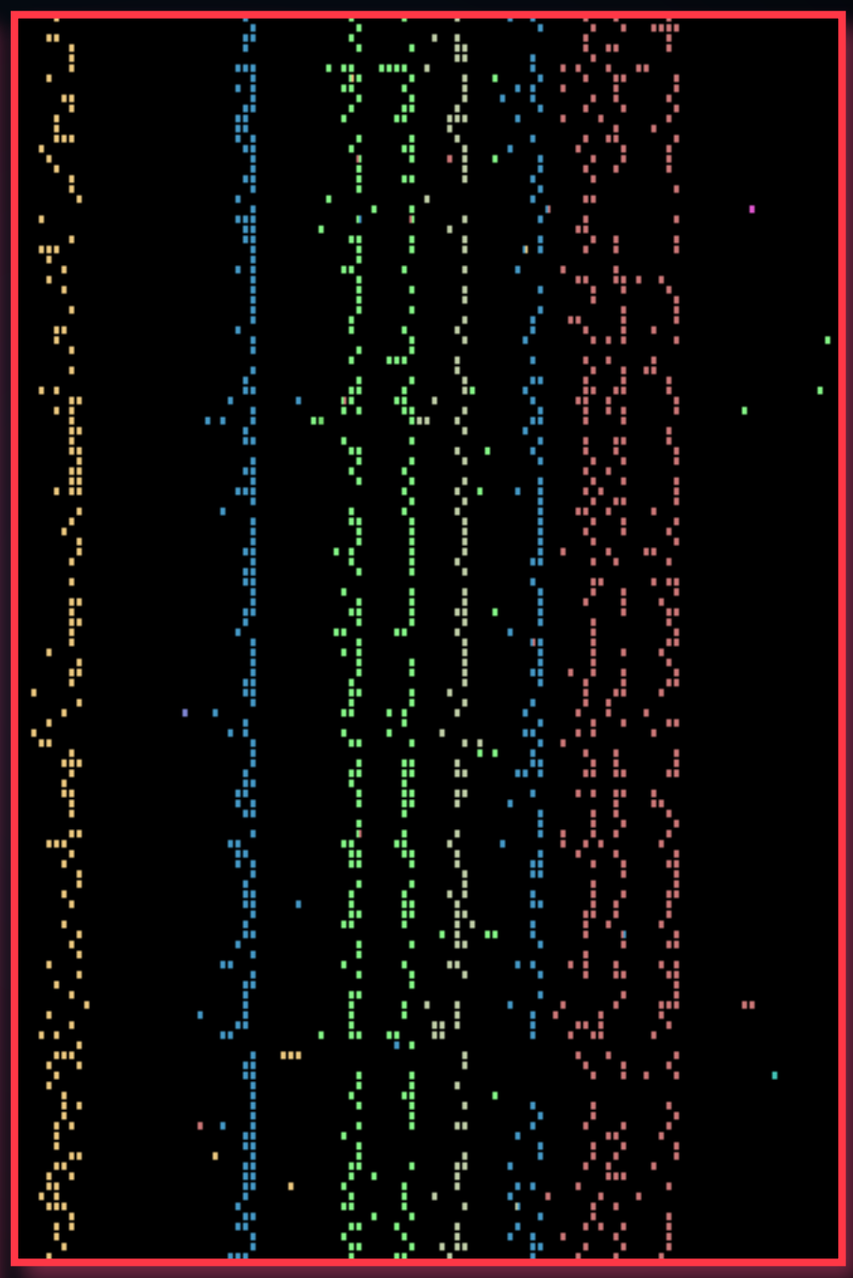
Clustering by Growth

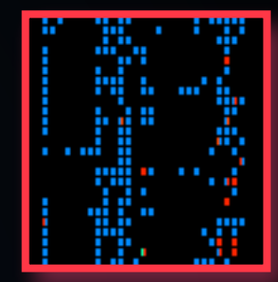


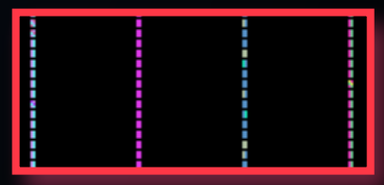
nanoHUB

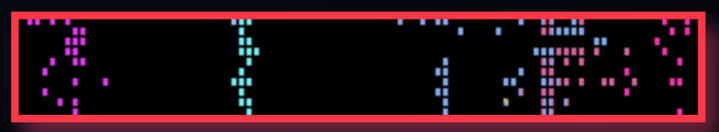
Classroom Clusters



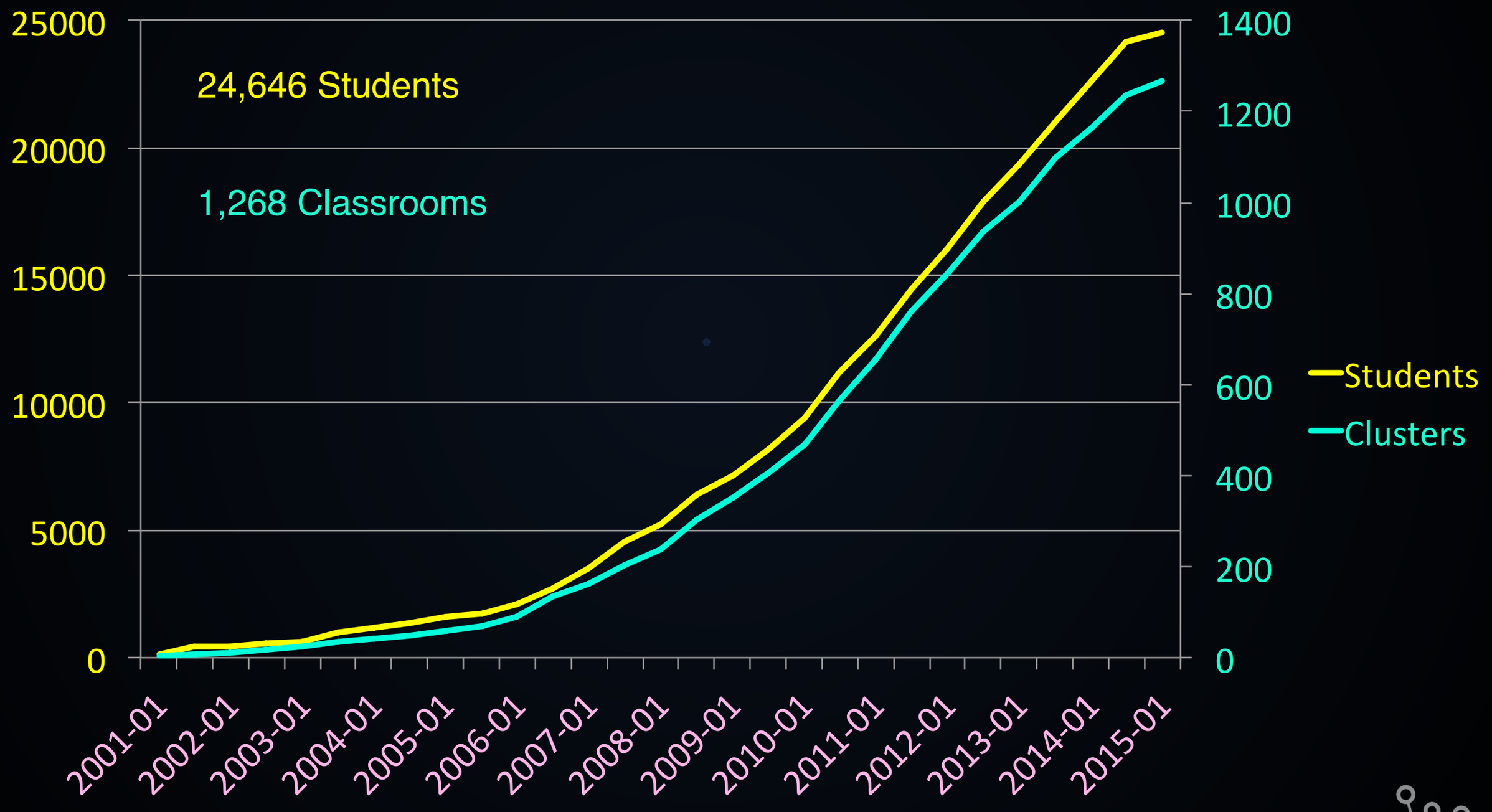






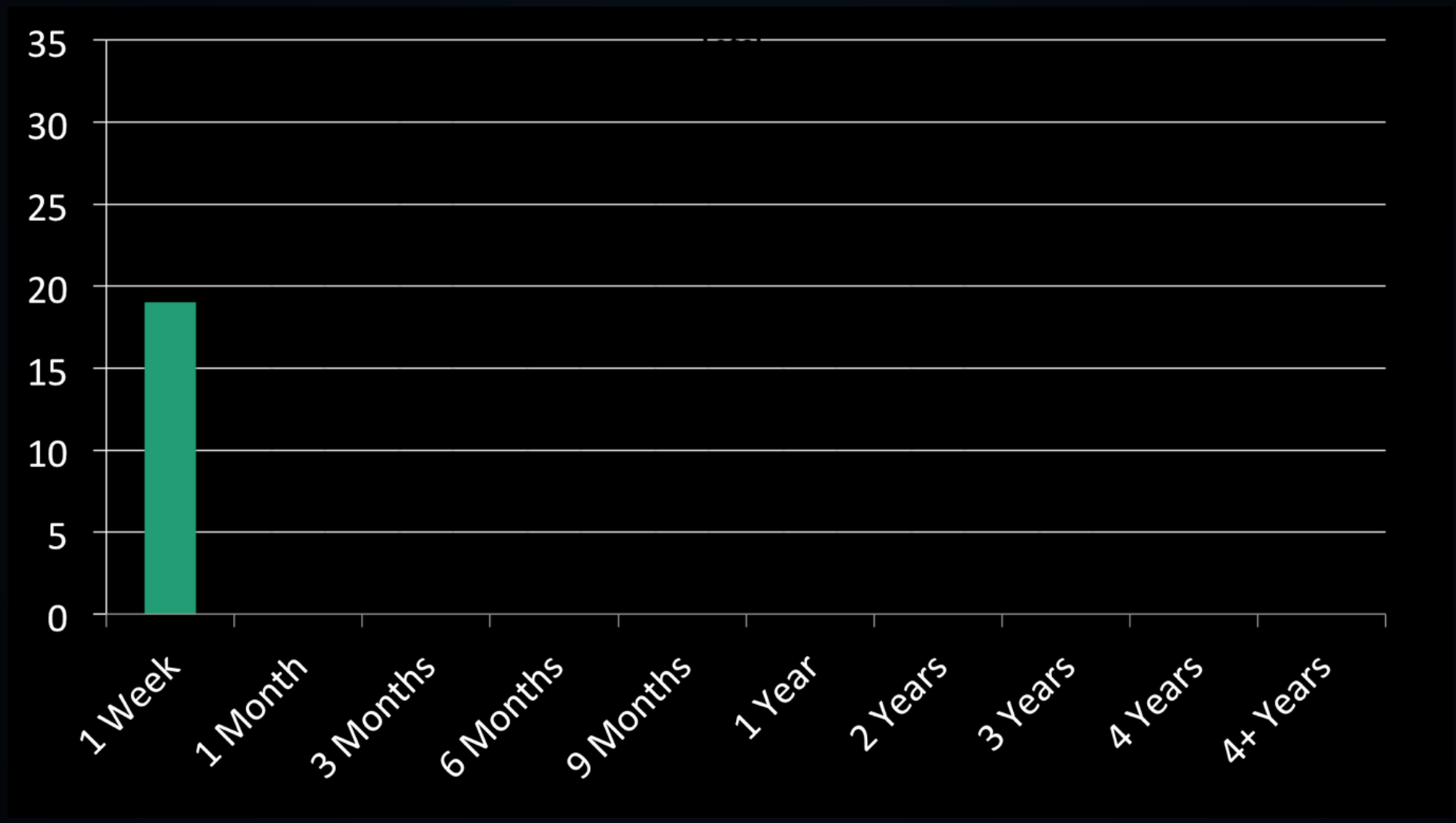


Classroom Impact

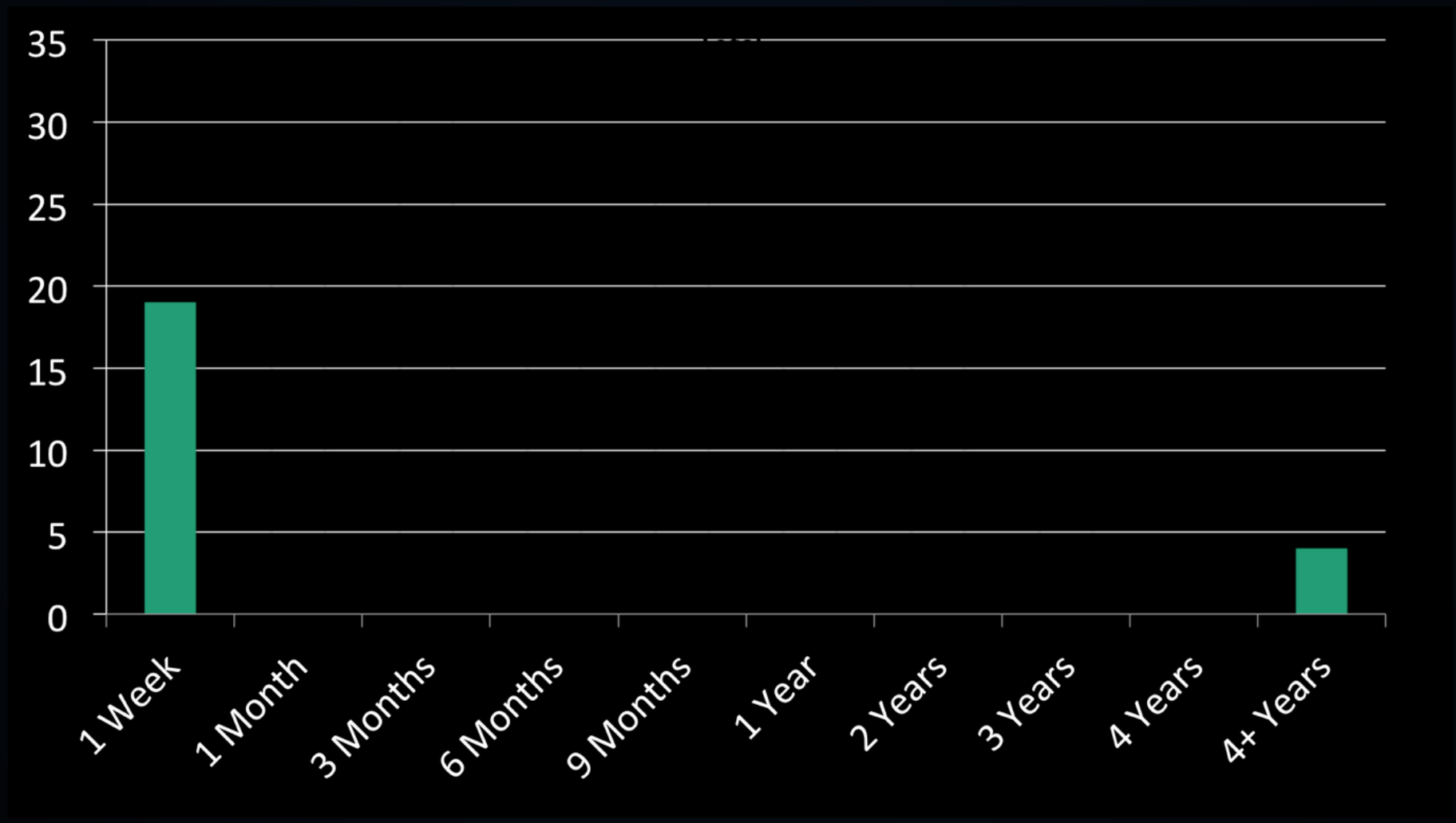


Today > 30,000 students in > 2,000 classrooms

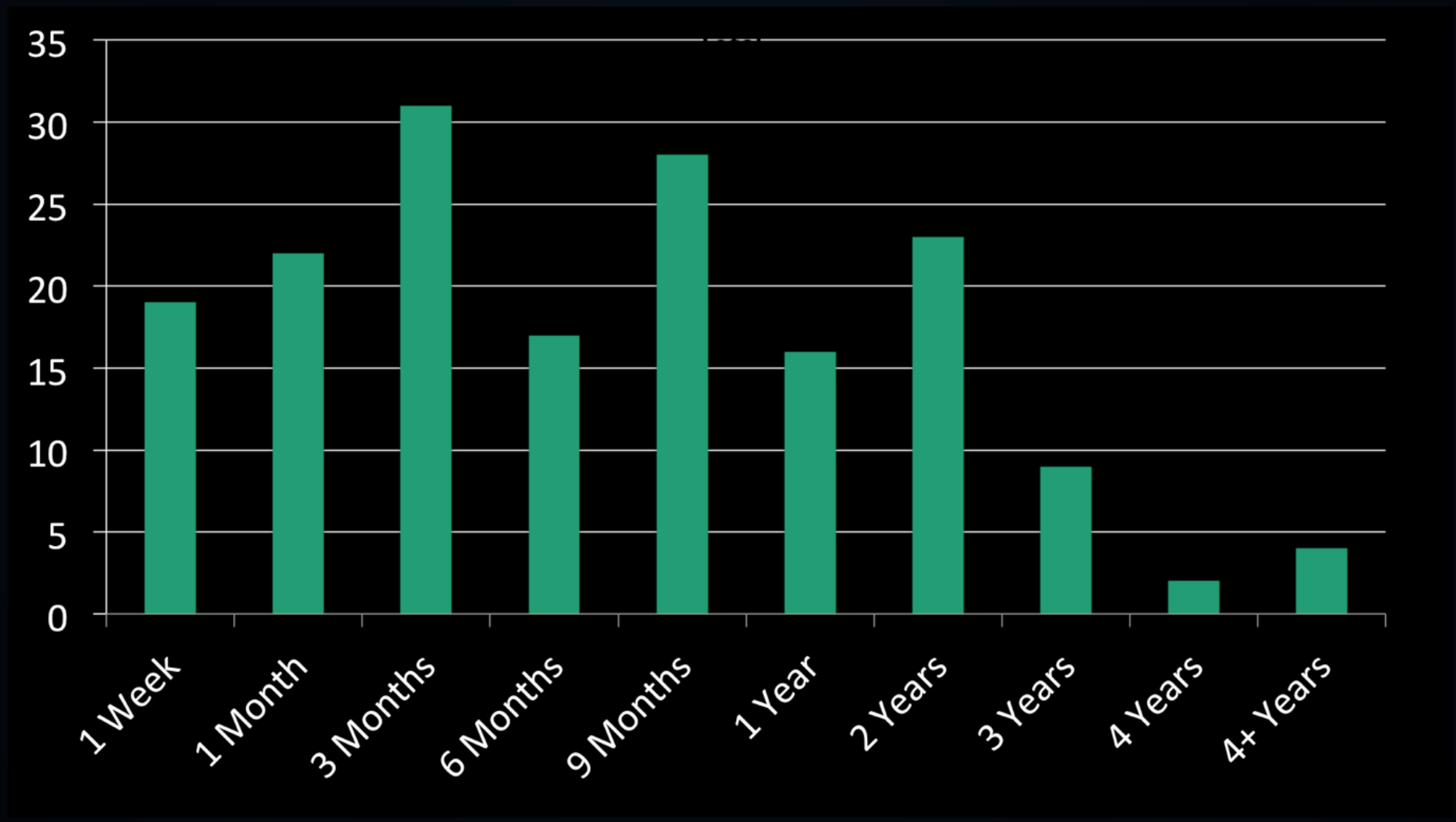
Impact - Adoption



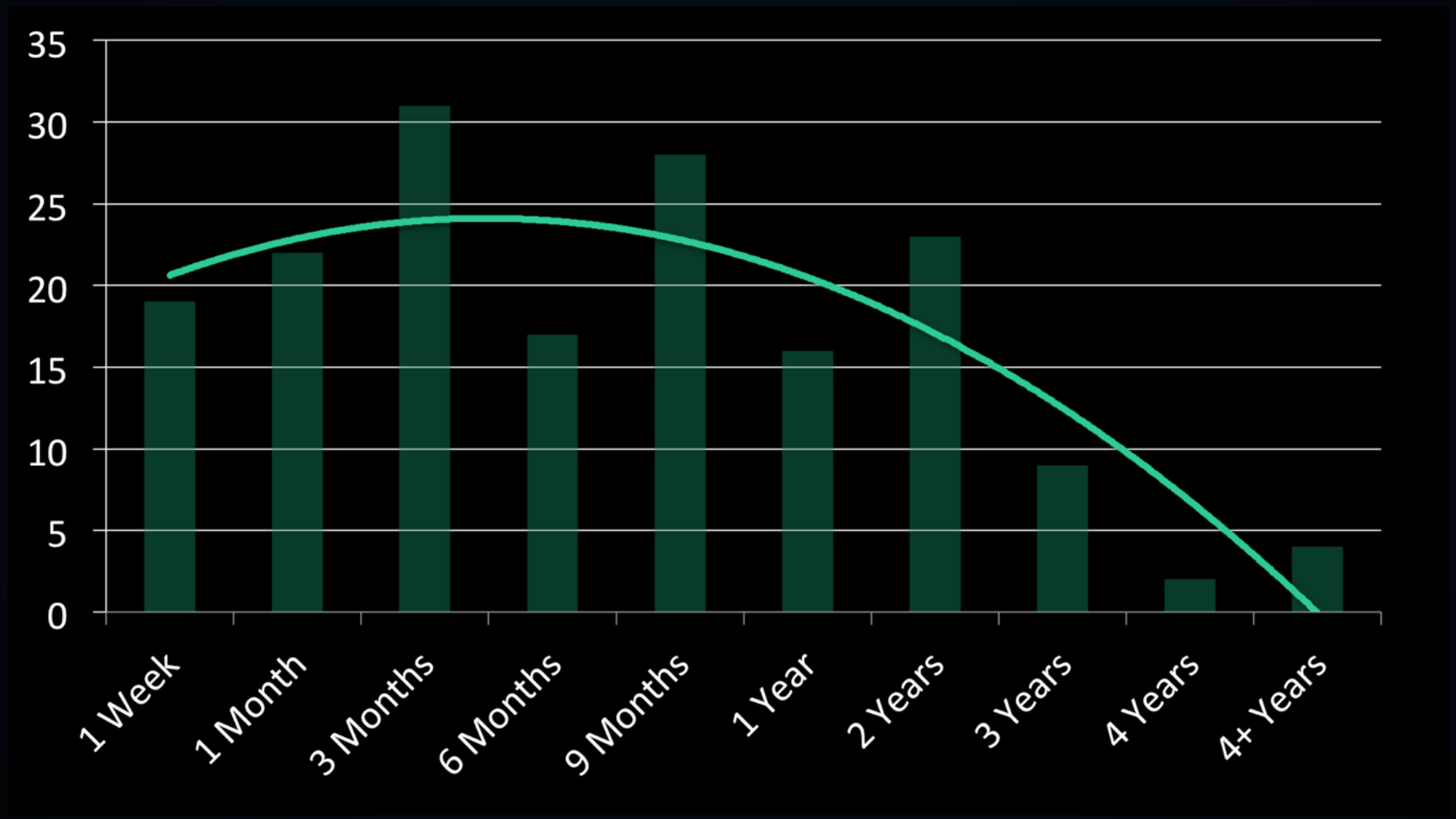
Impact - Adoption



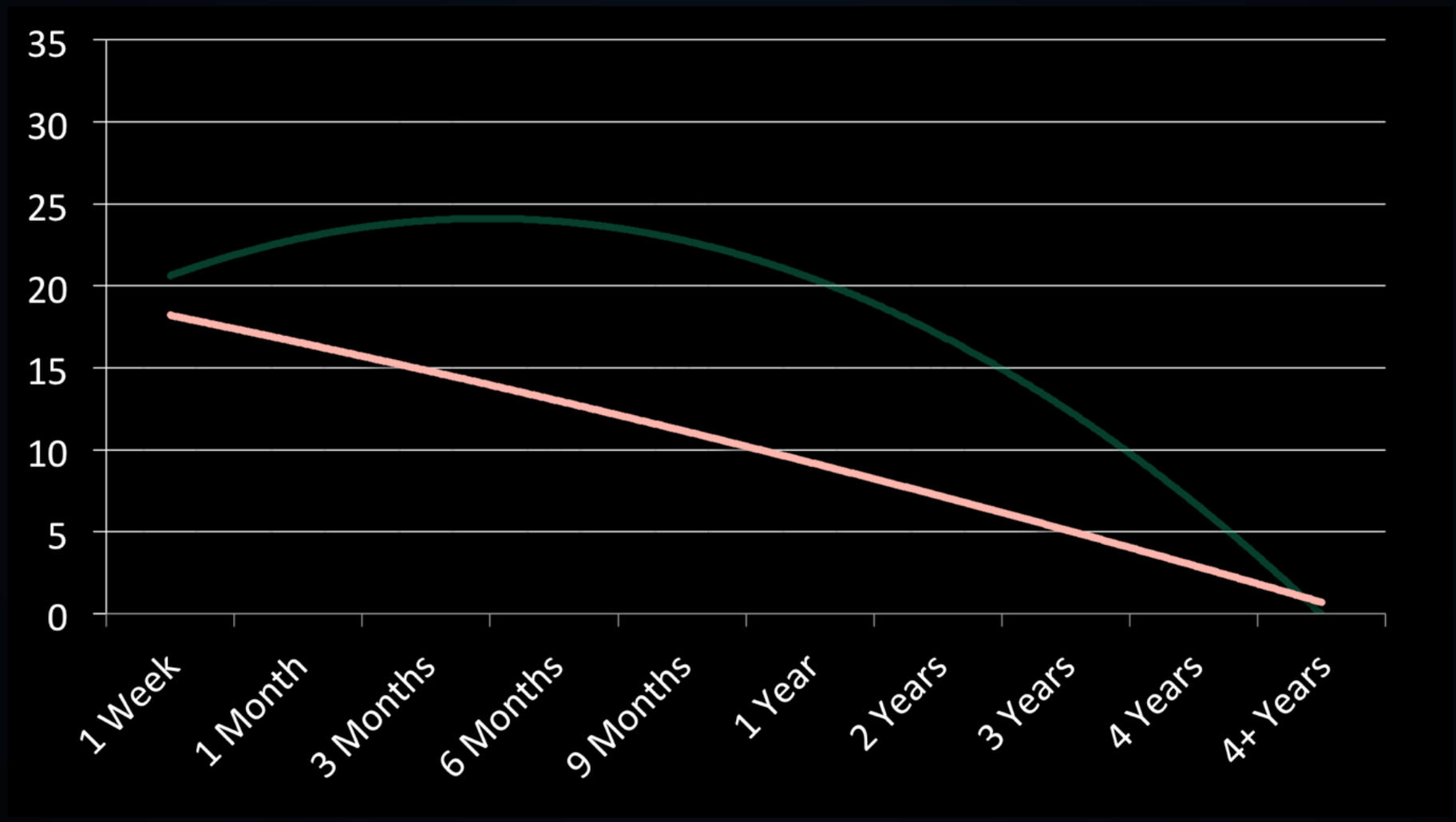
Impact - Adoption



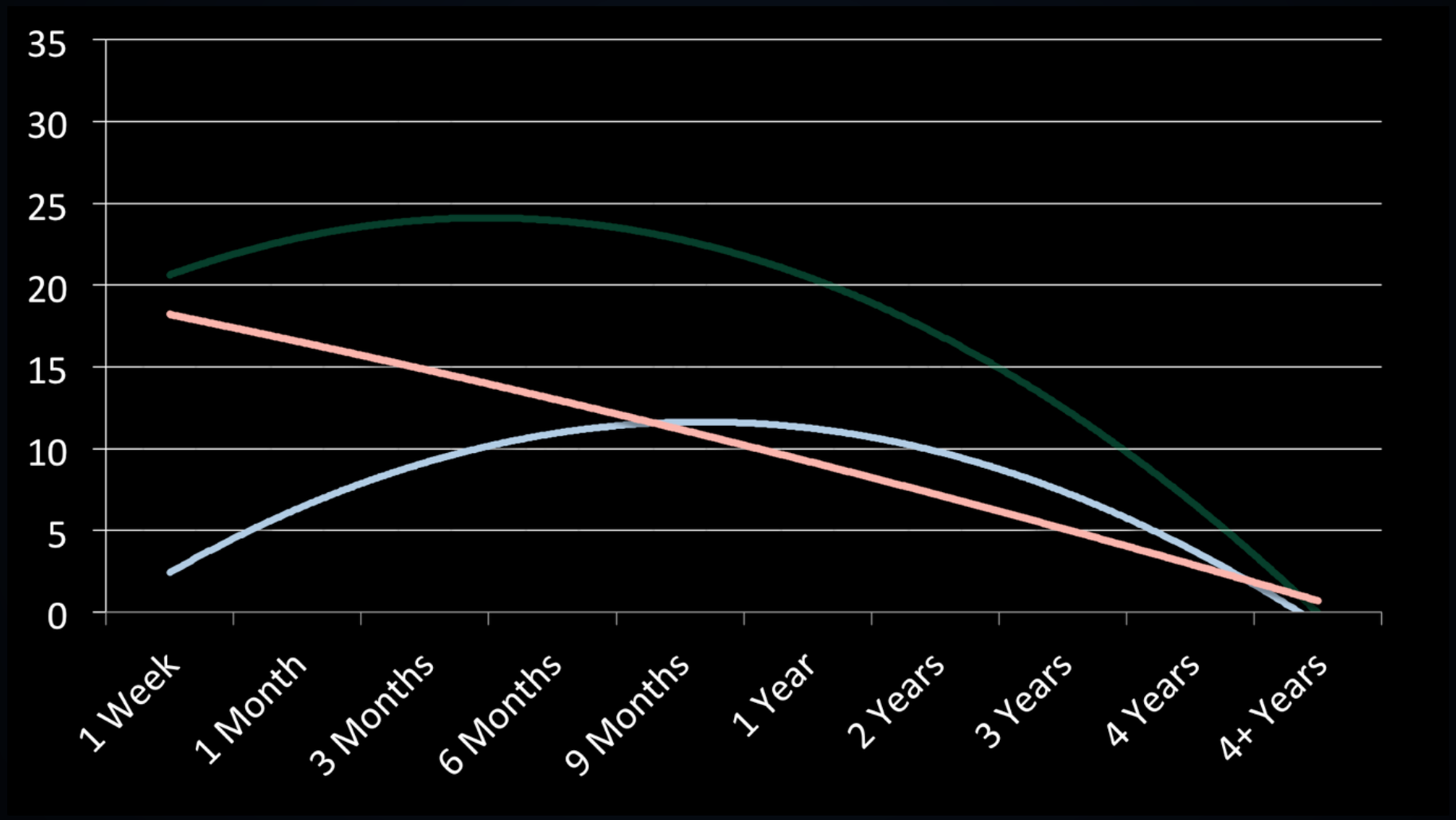
Impact - Adoption



Impact - Adoption

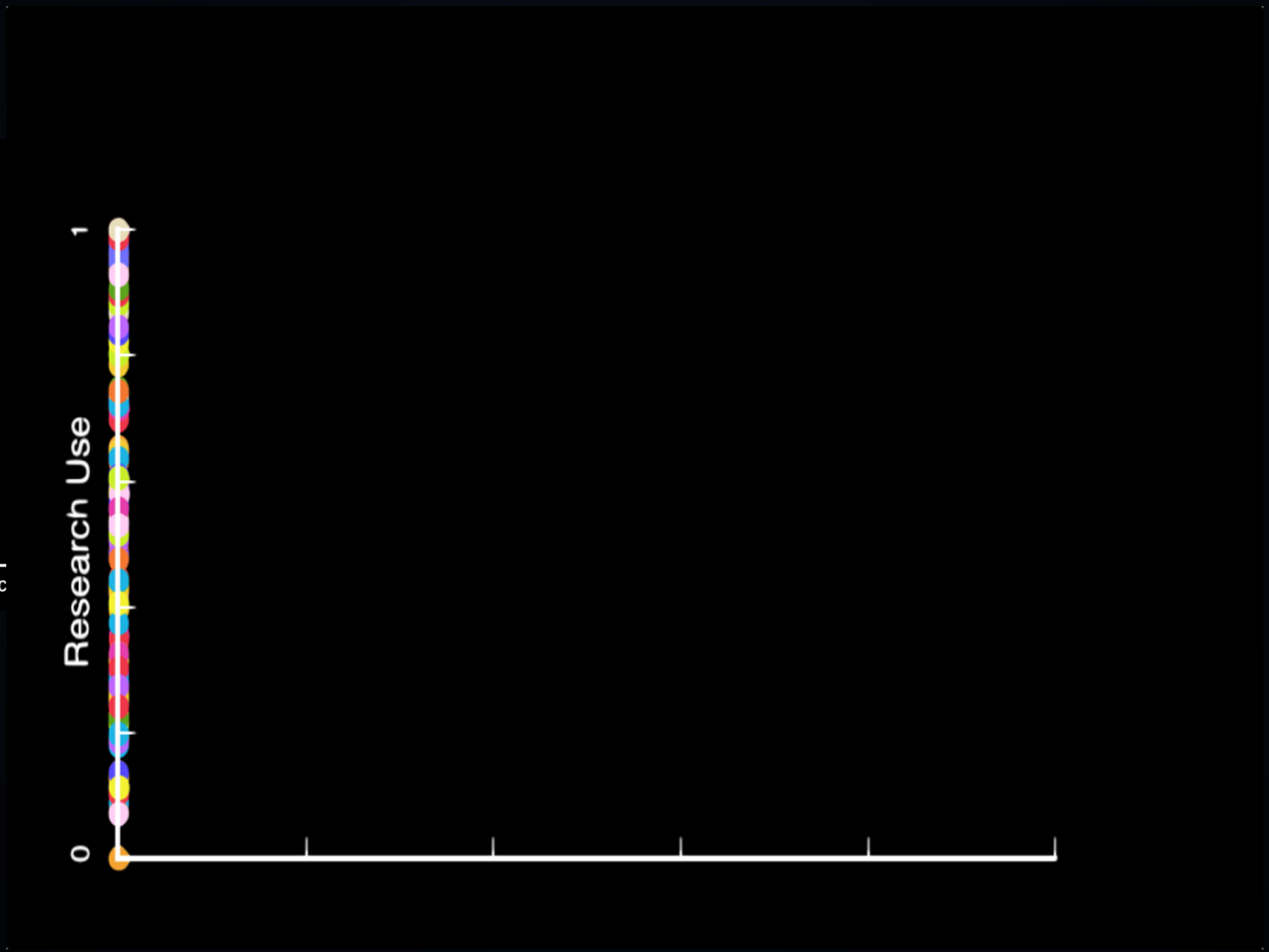


Impact - Adoption

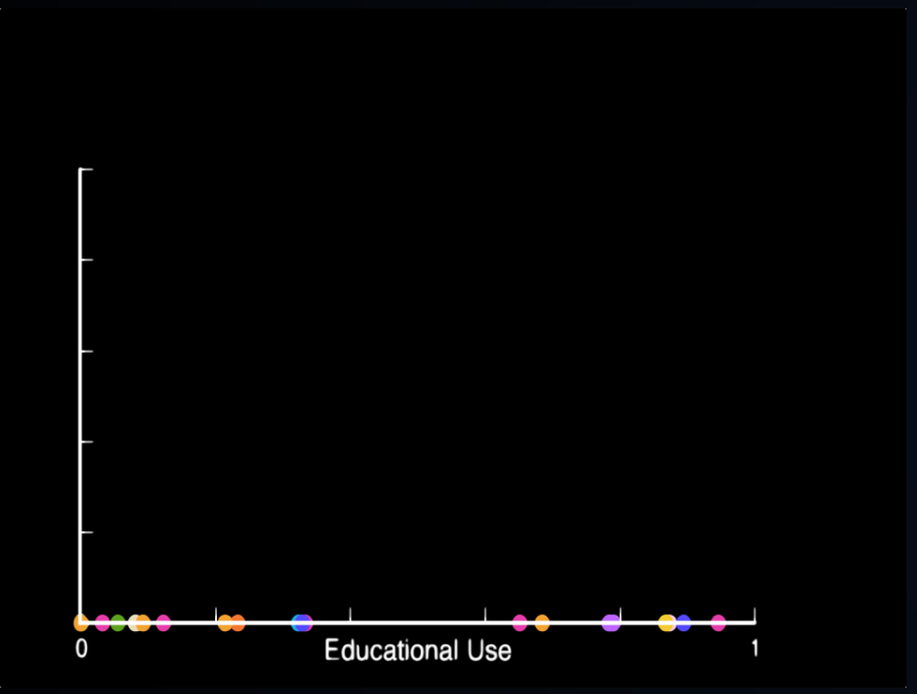


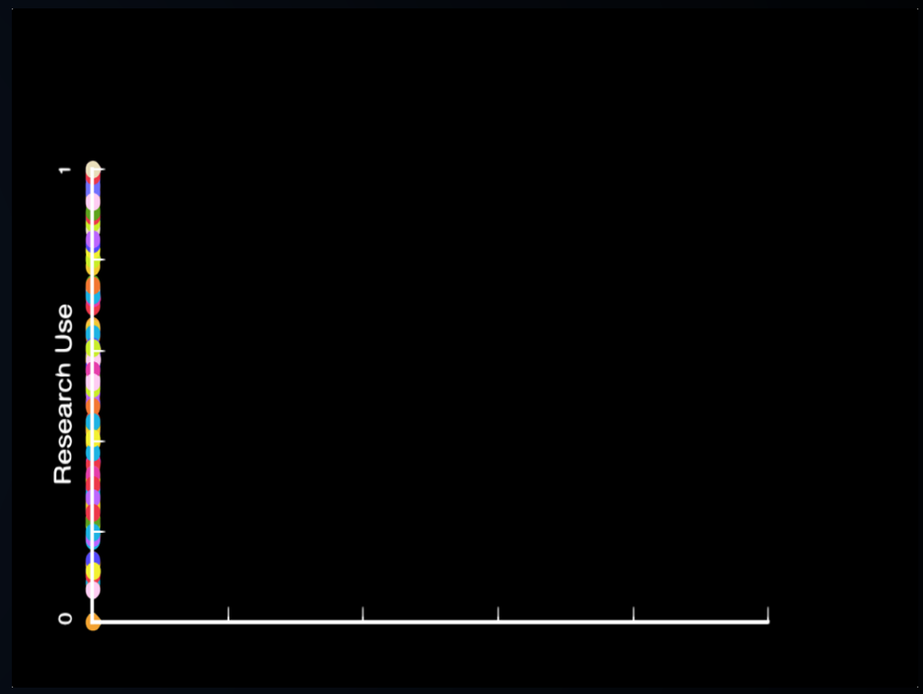
Impact - Migration

Impact - Migration

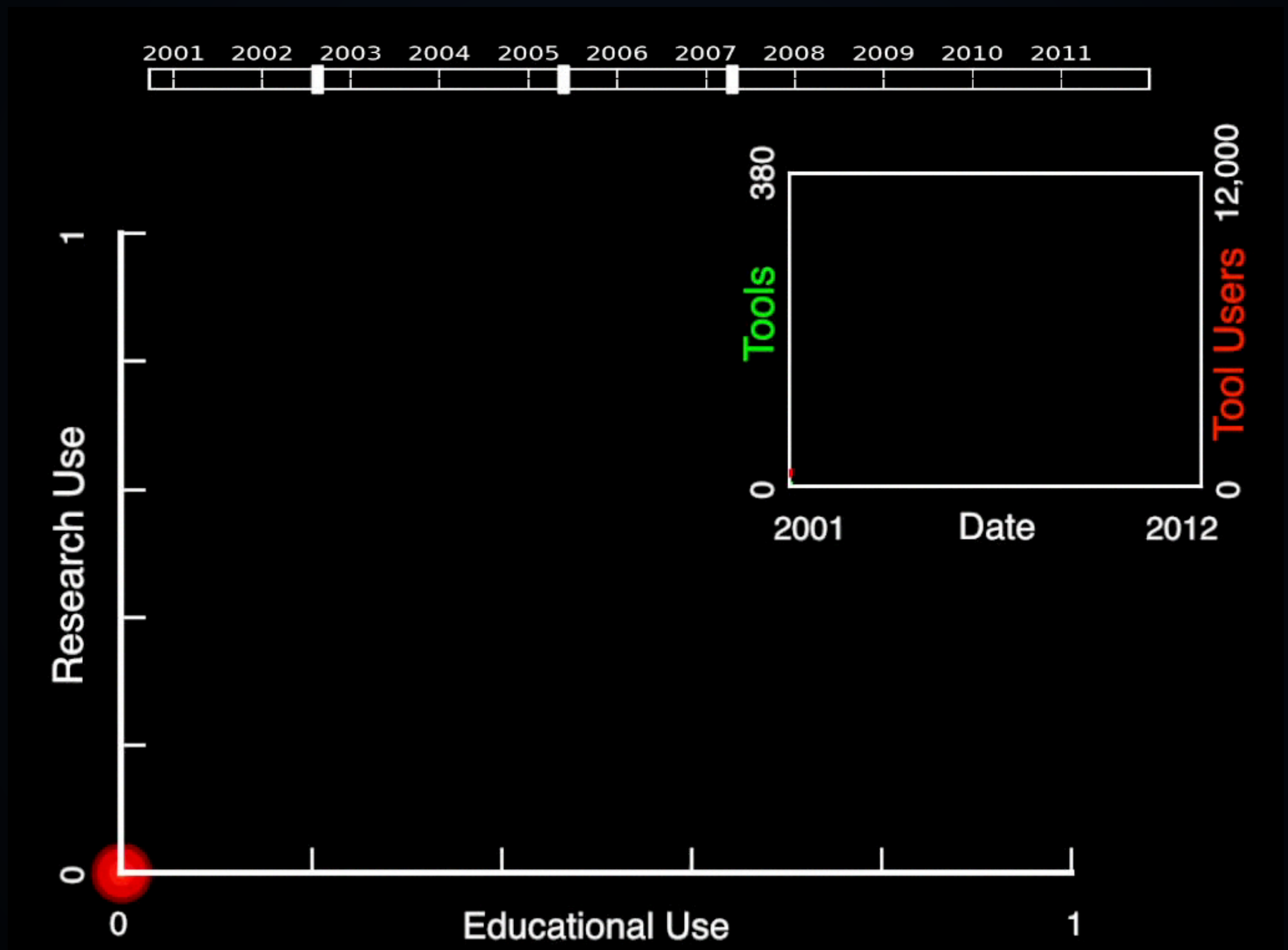


Impact - Migration





Impact - Migration

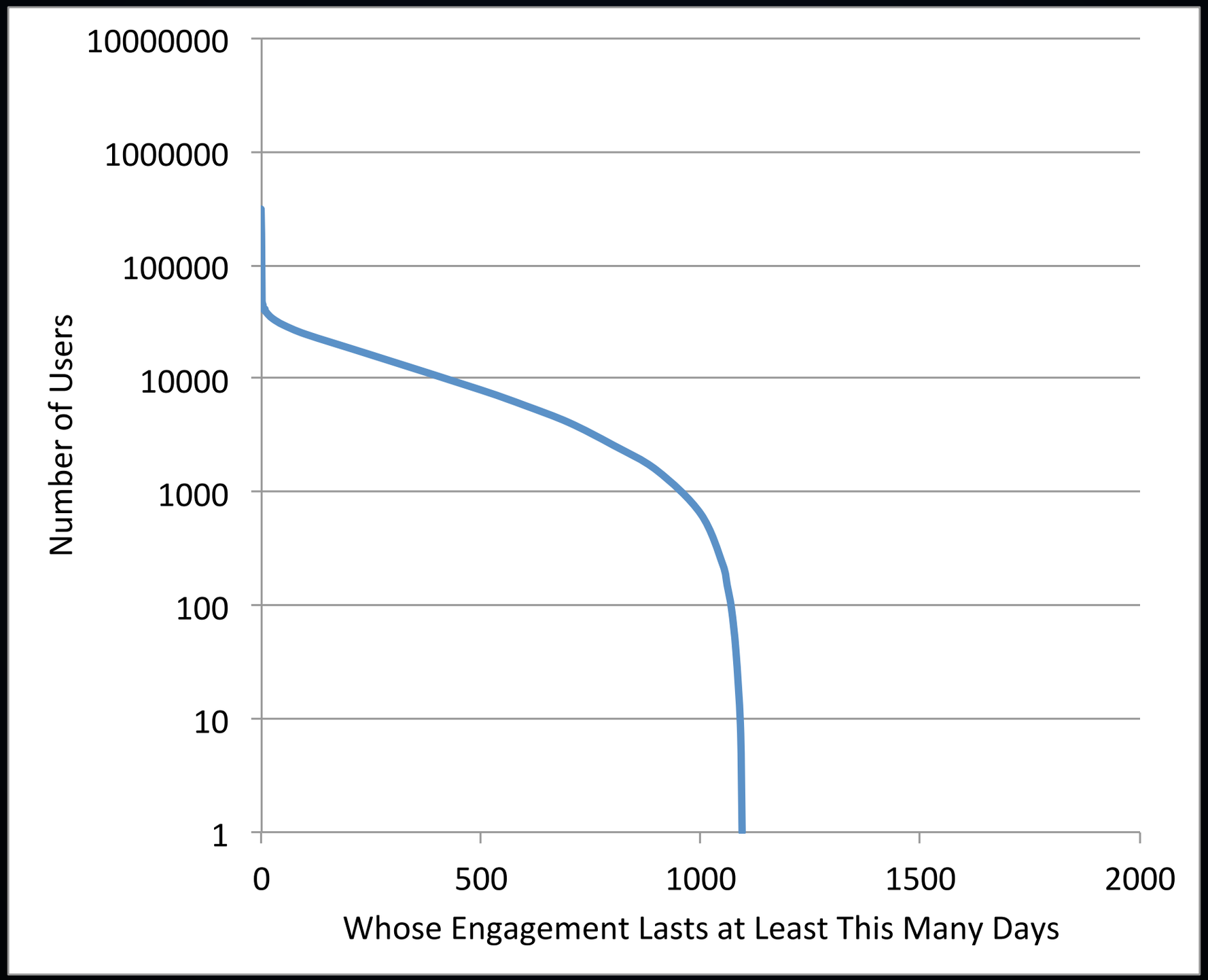


Credit: Dwight McKay

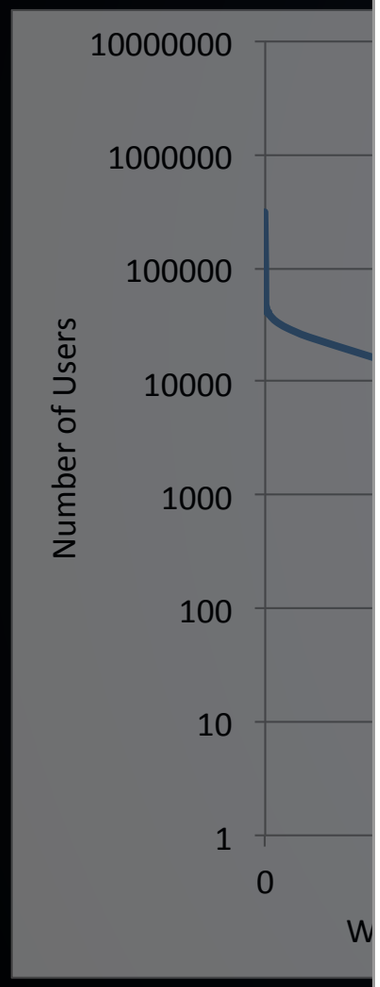
Who Cares?

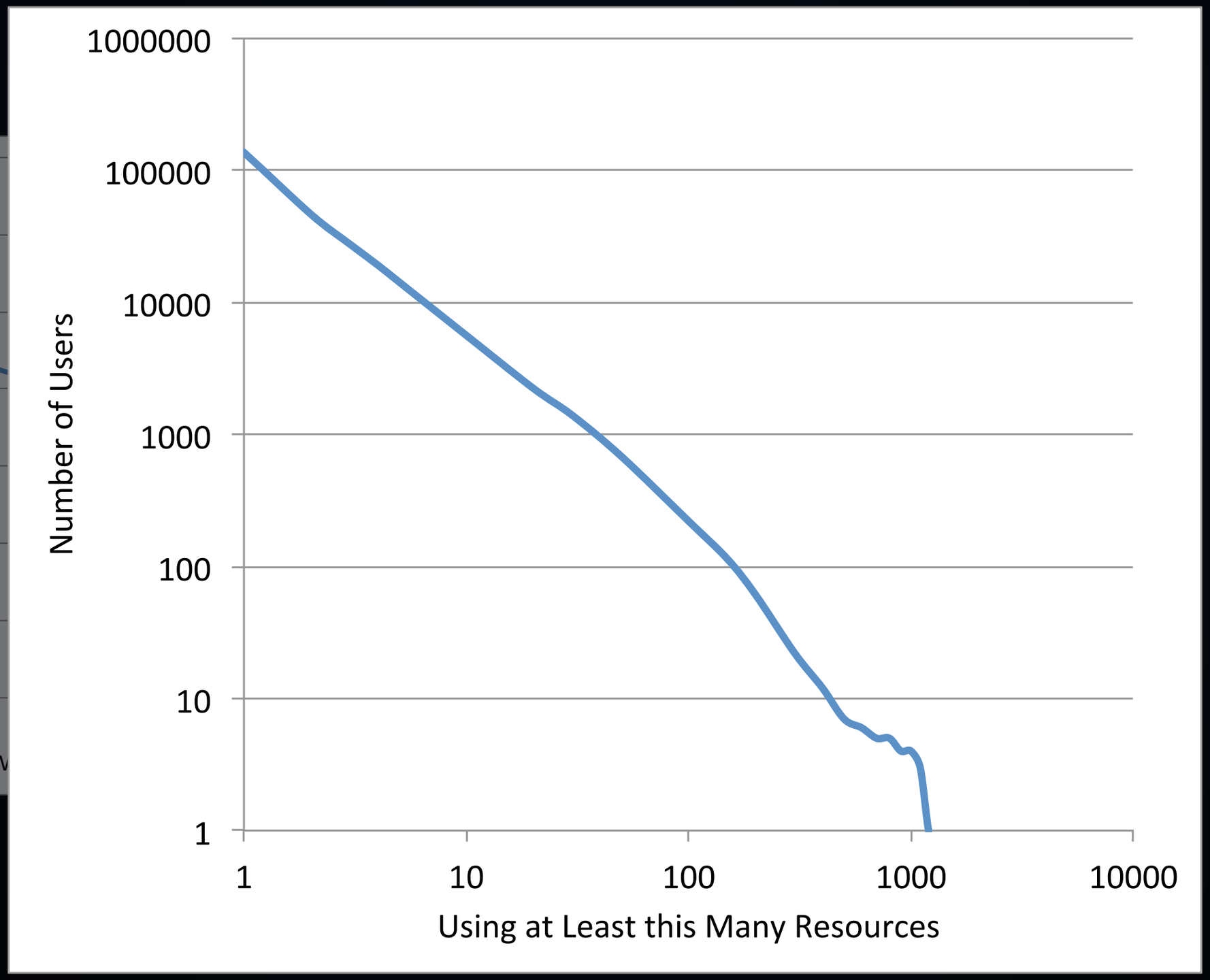

nH

Strategy Driving Curves

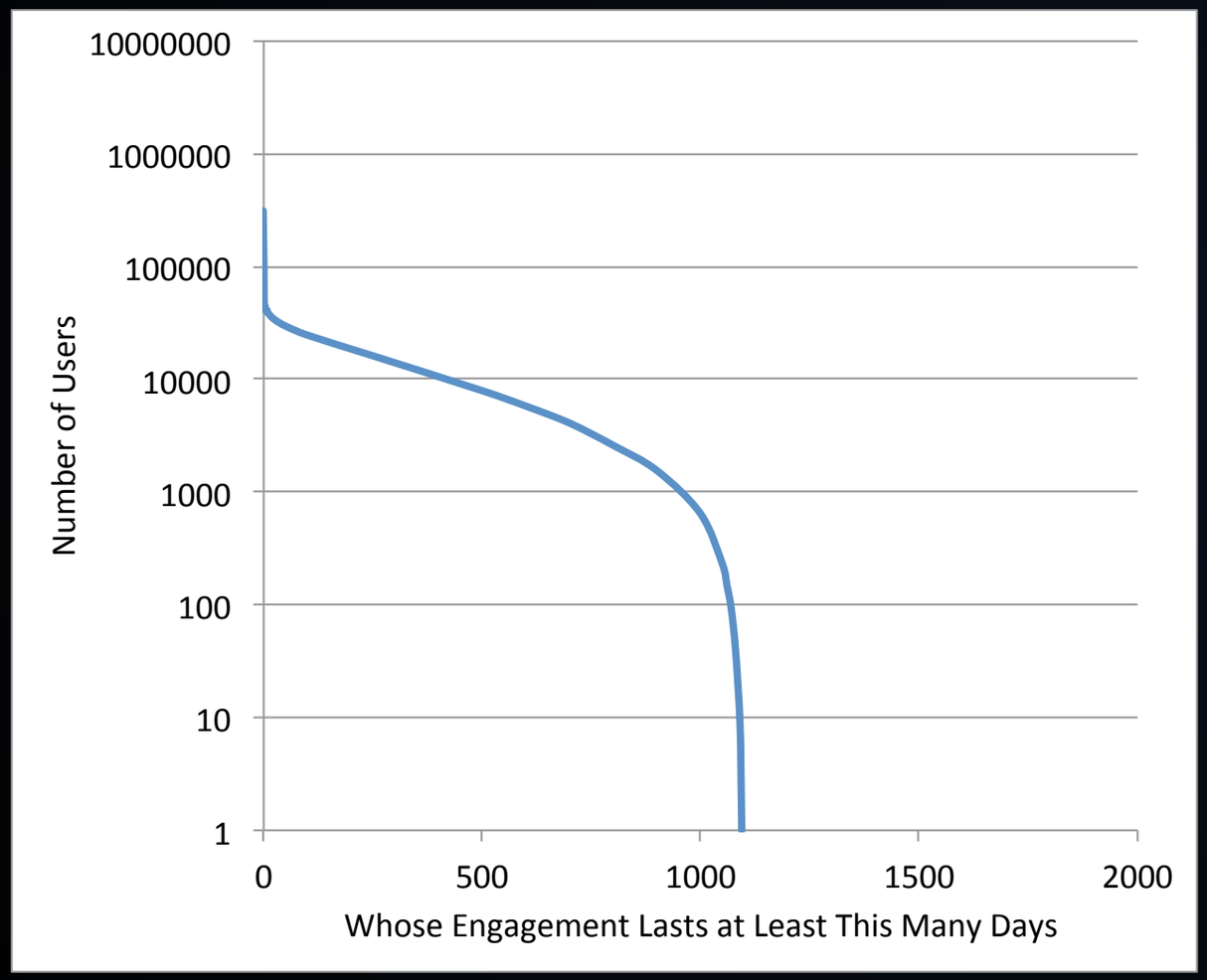


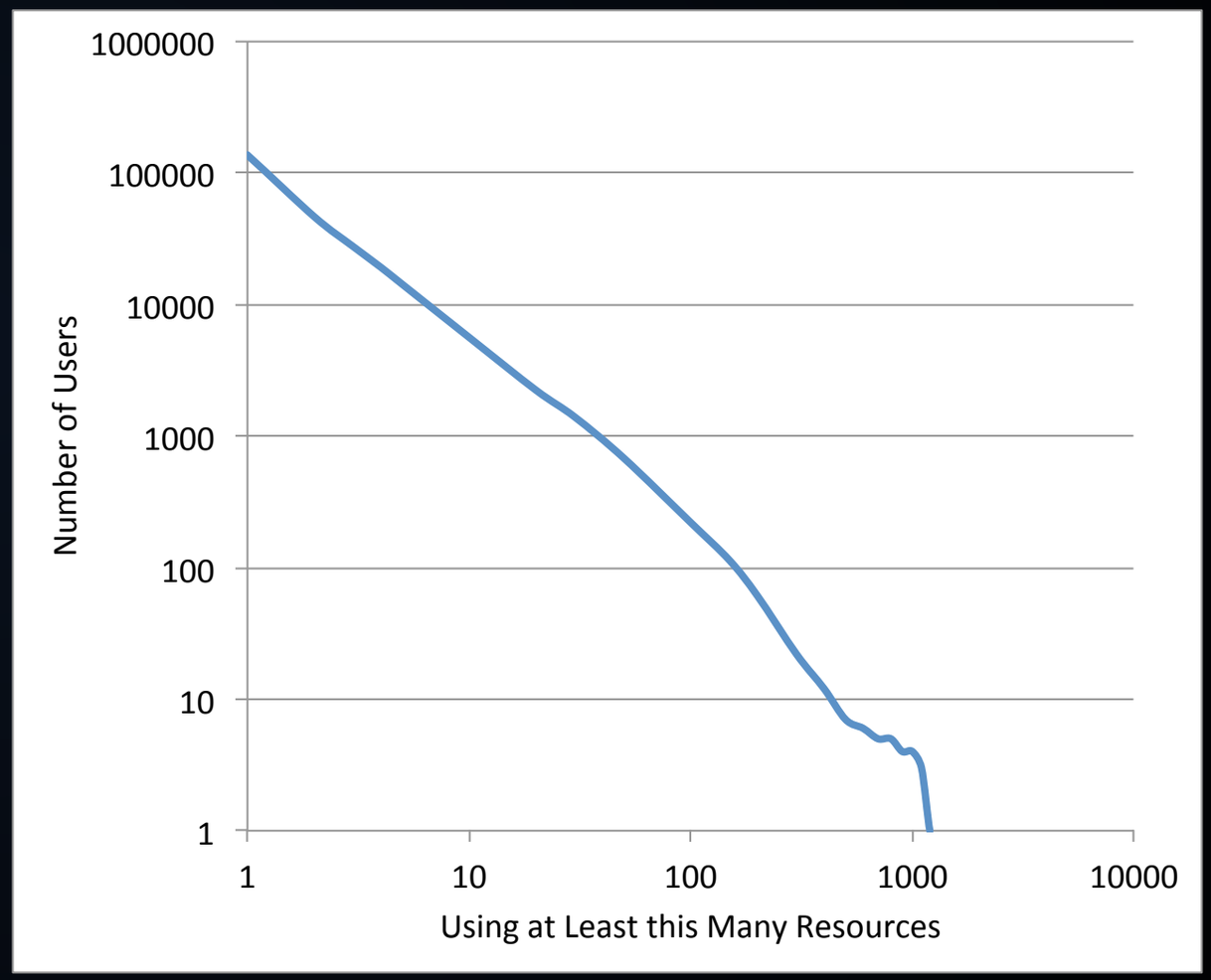
Strategy Driving Curves





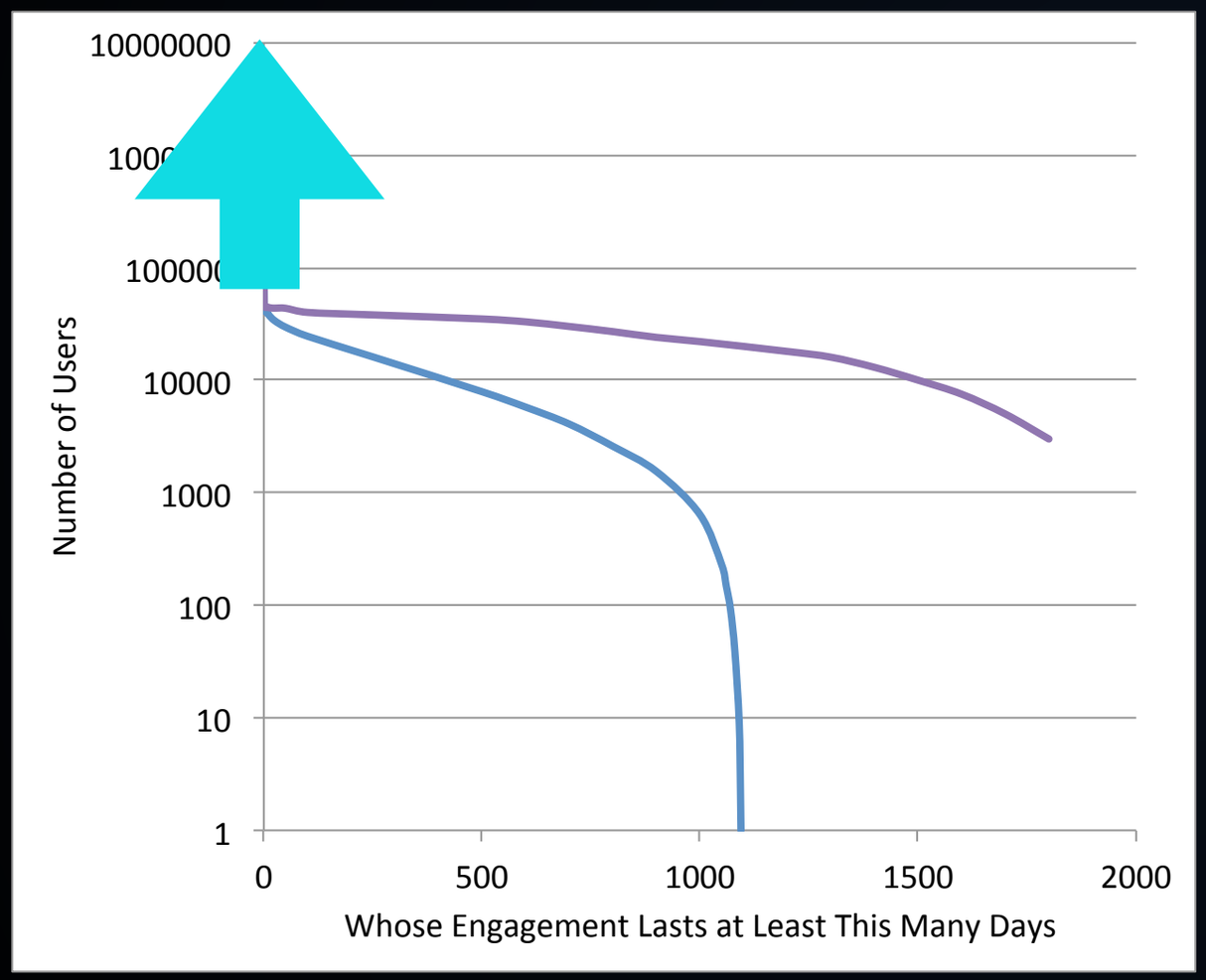
Strategy Driving Curves



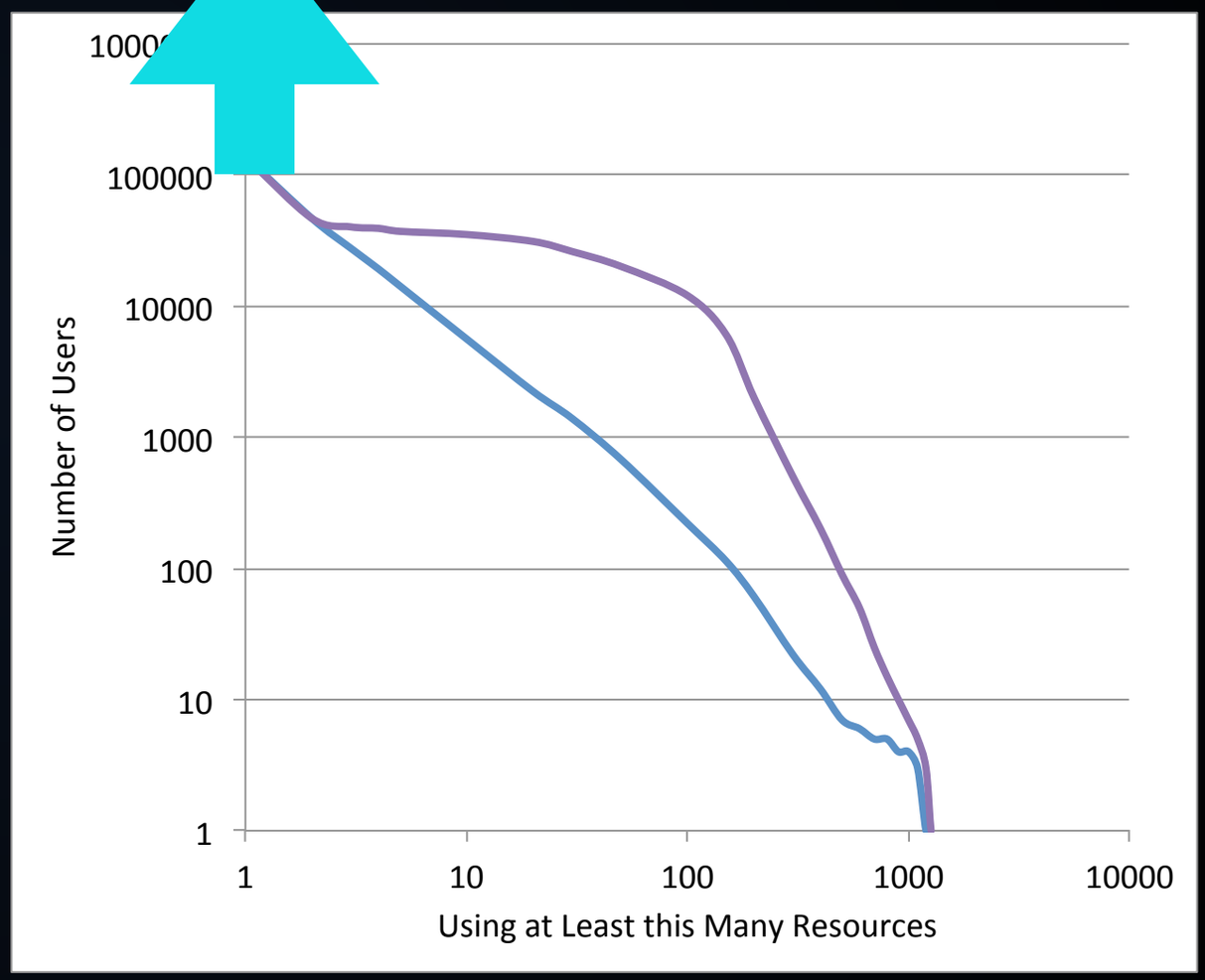


Strategy Driving Curves

To Increase User Engagement…



Lengthening



Deepening

Growing

Potential Data Driven Sustainability Interventions

- **Lengthening**
  - Recommend new simulations based on the behavior of others
  - Show instructors how their tools may be used more comprehensively in other classroom settings
- **Deepening**
  - Recommend related simulation tools to those a user has accessed
  - Show instructors tools related to those used in their classroom settings
- **Growing**
  - Data driven identification of educational domain champions
  - Producing “packaged” instruction kits to new instructors