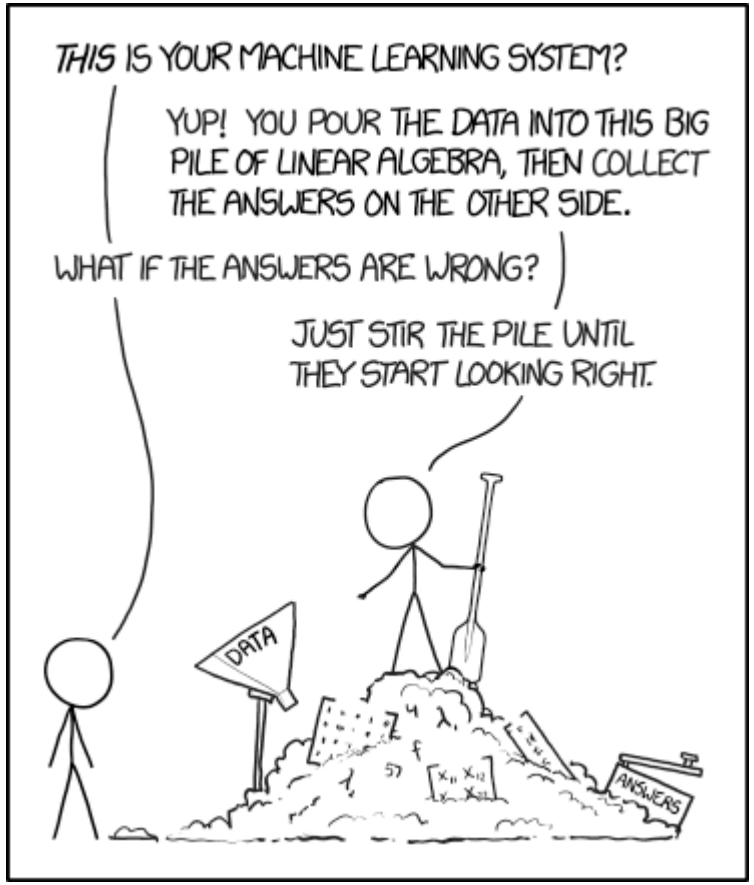
How I learned to stop worrying and model the diffusion

Pavan

https://github.com/pavanGurazada

Who am I?

- Sales guy
- Tinker with data for fun
- Right of the middle in $\boldsymbol{y} = f(\boldsymbol{X})$



Go grab a coffee, if this code seems obvious

```
#define ARMA_64BIT_WORD
#include <RcppArmadillo.h>
//[[Rcpp::depends(RcppArmadillo)]]
using namespace Rcpp;

// [[Rcpp::export]]
arma::vec mat_mult(arma::sp_mat& A,
                   arma::vec& n) {
return A * n;
}
```

```
mat.mult <- function(A, n) {
              return (A %*% n)
            }
```

```
benchmark(mat_mult(A, n), mat.mult(A, n))
        test       replications  elapsed  relative
2 mat.mult(A, n)          2000   243.92     1.000
1 mat_mult(A, n)          2000   462.14     1.895
```

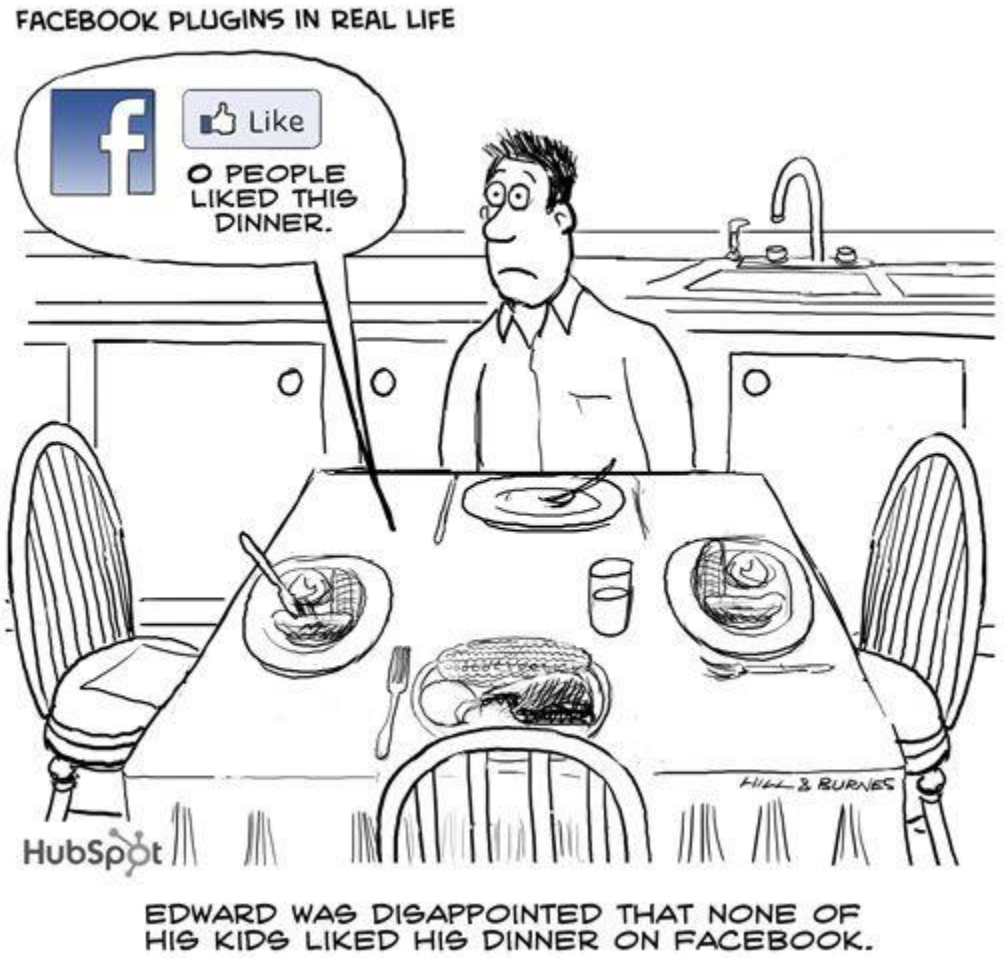
Go grab a coffee, if this code seems obvious

```
#define ARMA_64BIT_WORD
#include <RcppArmadillo.h>
//[[Rcpp::depends(RcppArmadillo)]]
using namespace Rcpp;


// [[Rcpp::export]]
arma::vec mat_mult(arma::sp_mat& A,
                   arma::vec& n) {

return A * n;
}
```

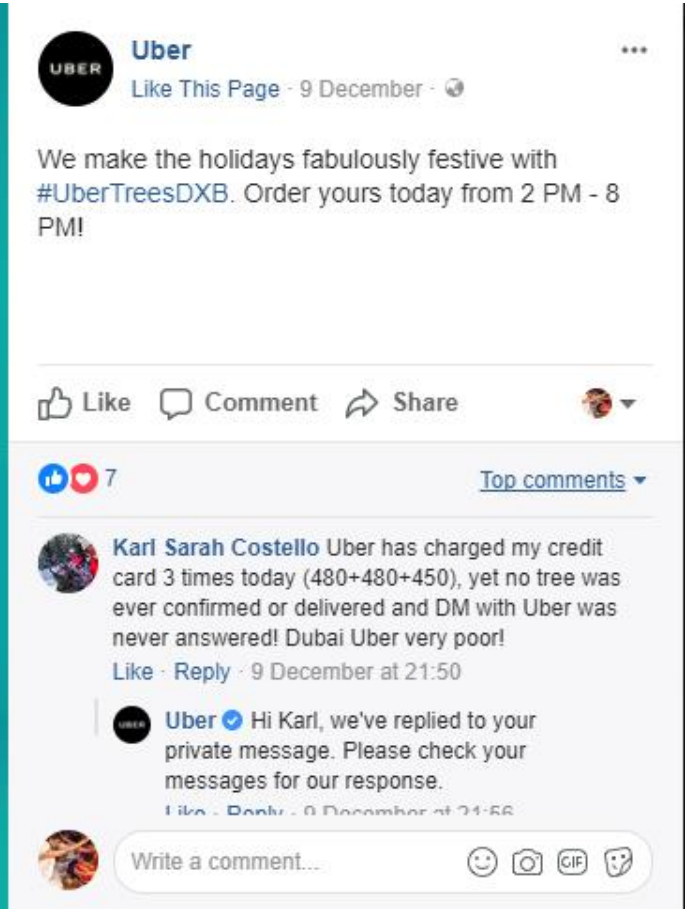
```
mat.mult <- function(A, n) {
              return (A %*% n)
            }
```

```
benchmark(mat_mult(A, n), mat.mult(A, n))
       test       replications  elapsed  relative
2 mat.mult(A, n)          2000   243.92     1.000
1 mat_mult(A, n)          2000   462.14     1.895
```

What was my problem?



Why should you bother?

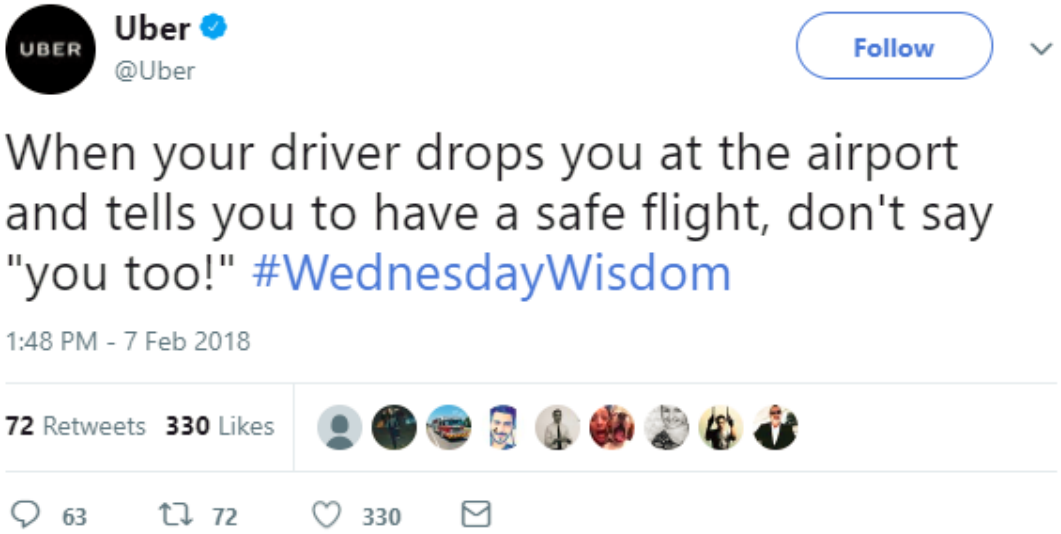


So, does anyone read anything we post on social media ever?

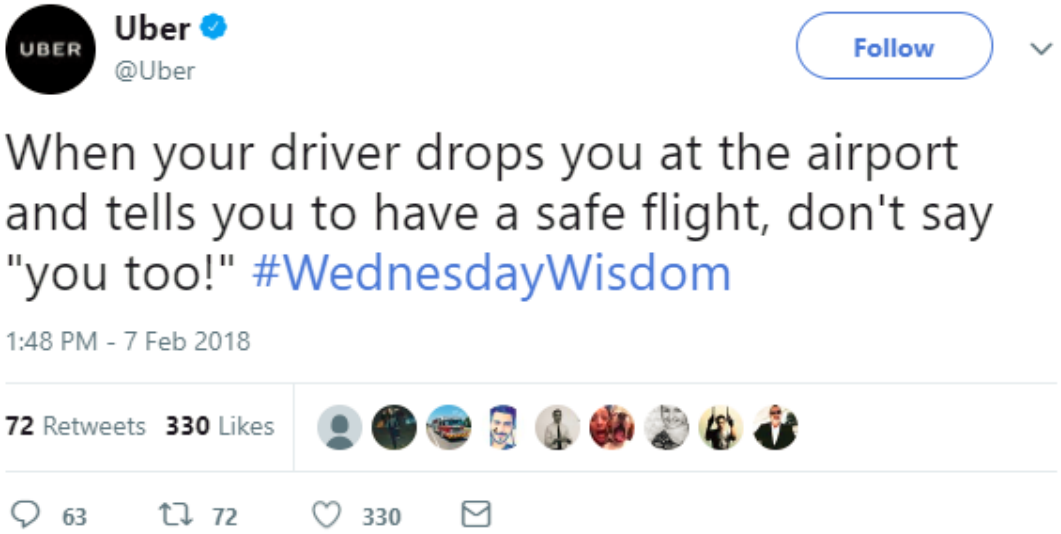
Poll

Do you find your social media newsfeed interesting?

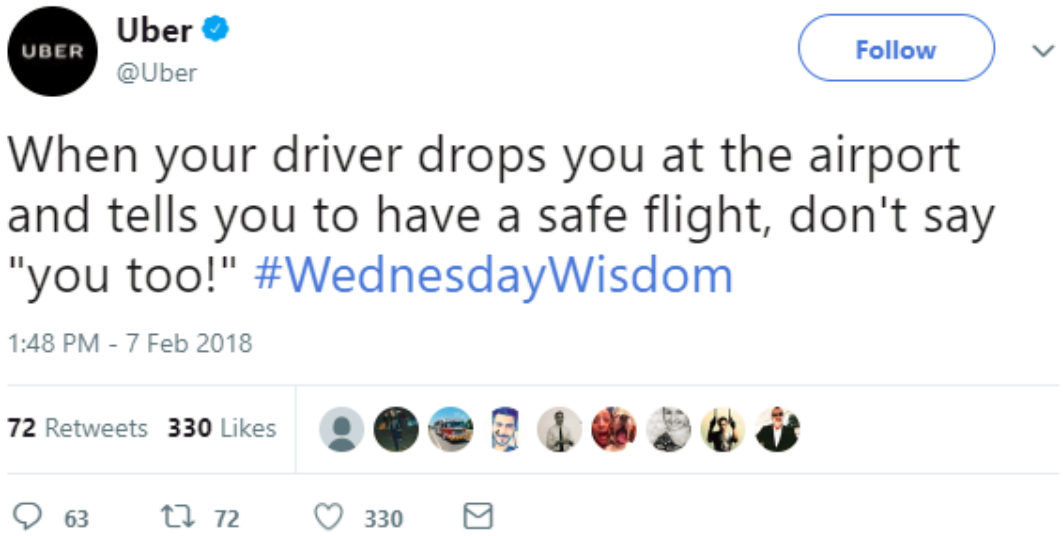
Okay, but why should you bother?



Okay, but why should you bother?



Okay, but why should you bother?



Cost-to-marketer: $0

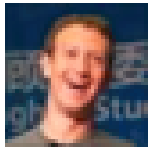
Cost-to-marketer: $10,000

As a business, should I:

1. Place ads, or

2. Co-create content

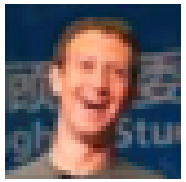
It turns out, even Facebook had the same problem!



Mark Zuckerberg

about 2 weeks ago

We just announced our quarterly results and community update. Our focus in 2018 is making sure Facebook isn't just fun, but also good for people's well-being and for society. We're doing this by encouraging meaningful connections between people rather than passive consumption of content. Already last quarter, we made changes to show fewer viral videos to make sure people's time is well spent. In total, we made changes that reduced time spent on Facebook by roughly 50 million hours every day. By focusing on meaningful connections, our community and business will be stronger over the long term.

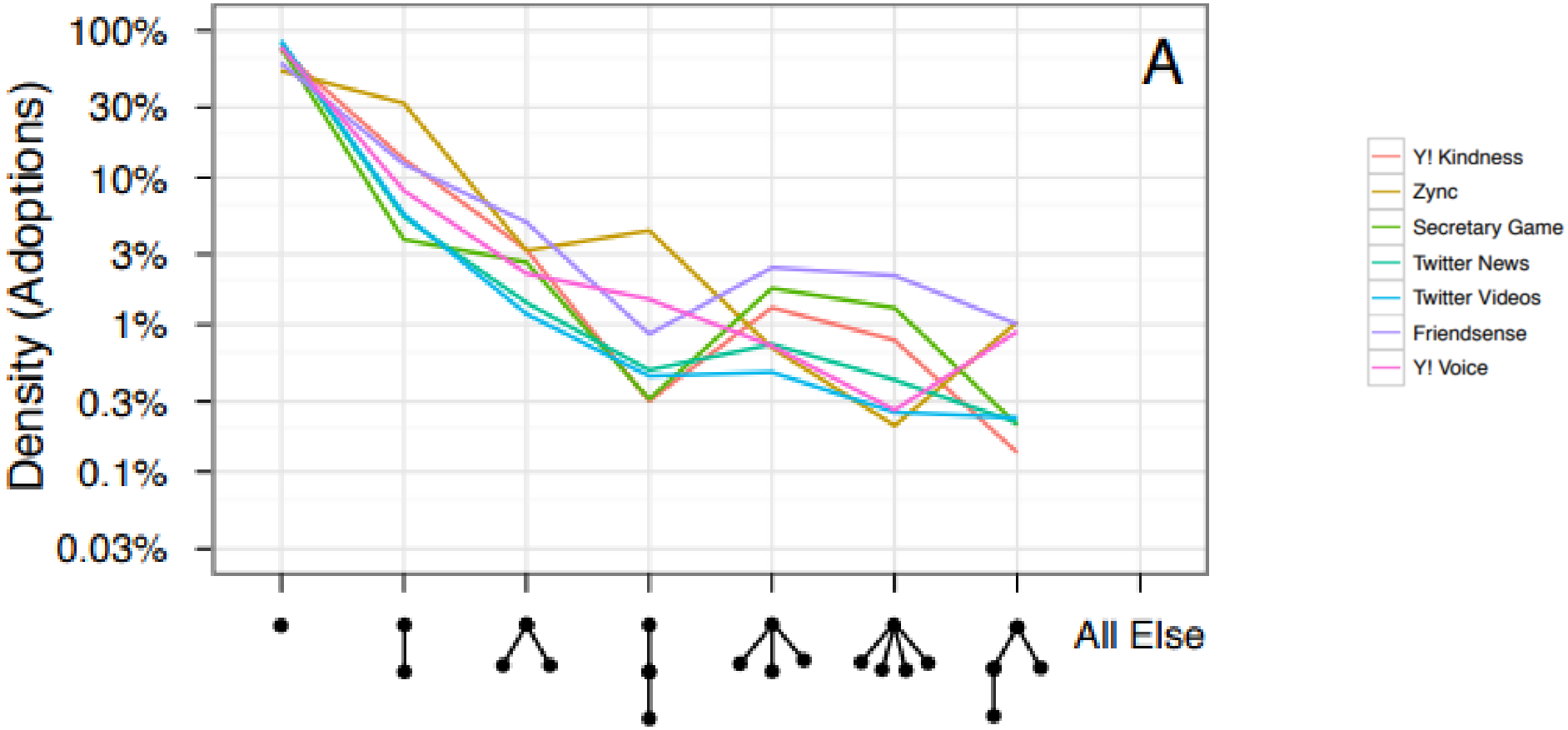


Mark Zuckerberg
about 2 weeks ago

rather than passive consumption of content. Already last quarter, we made changes to show fewer viral videos to make sure people's time is well spent. In total, we made changes that reduced time spent on Facebook by roughly 50 million hours every day.

What do scholars say about social media?

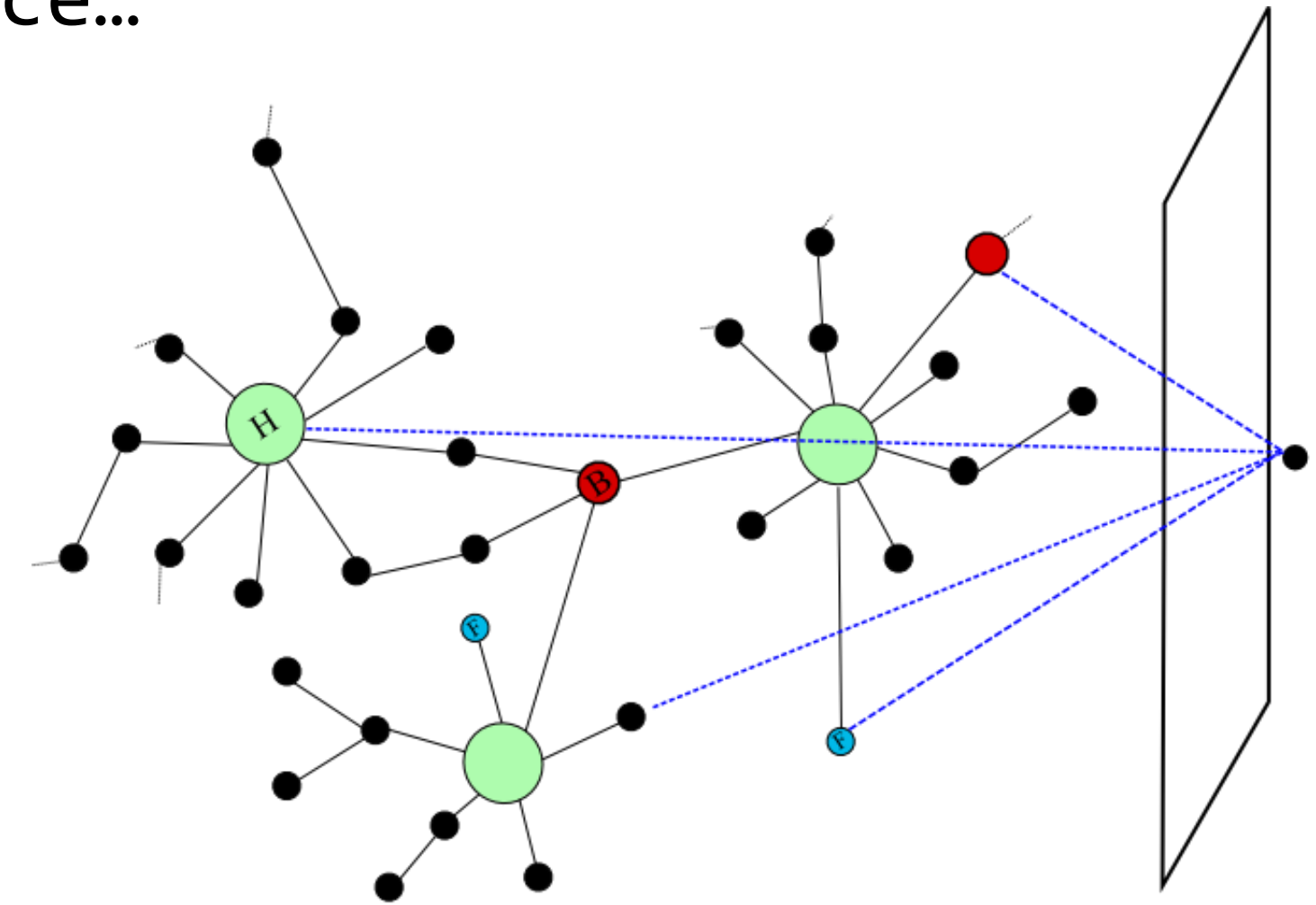
The law of social media is…



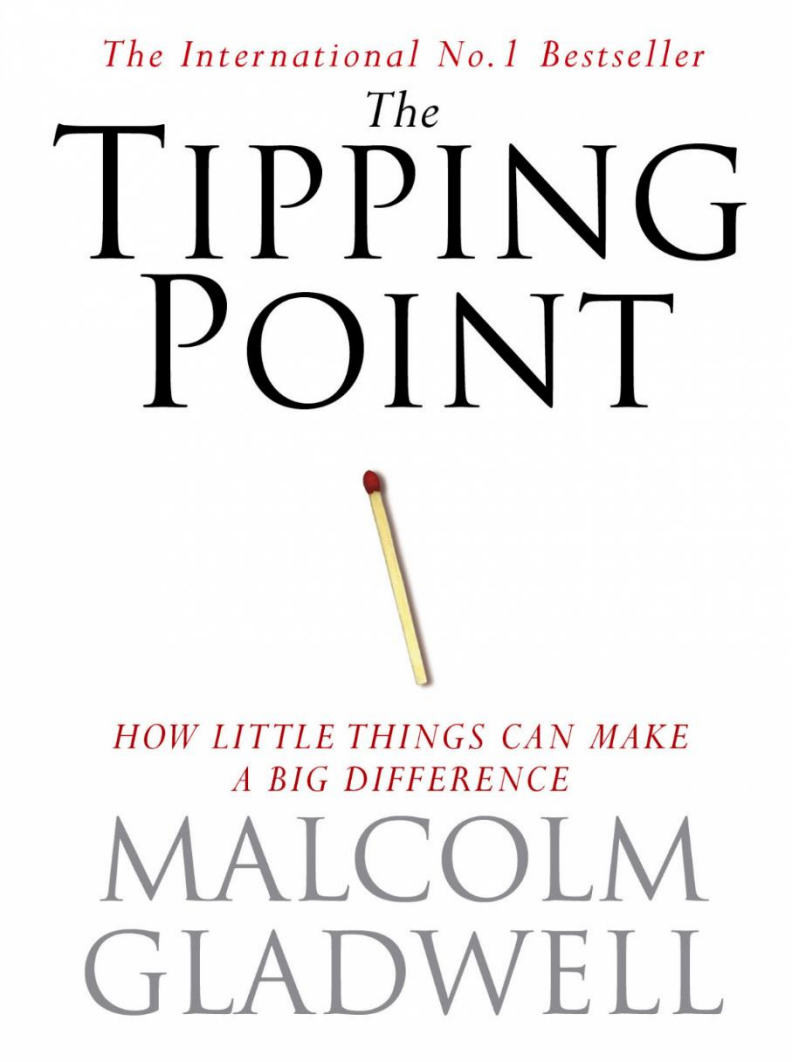
Nothing happens most of the time!

(Goel et al., 2012)

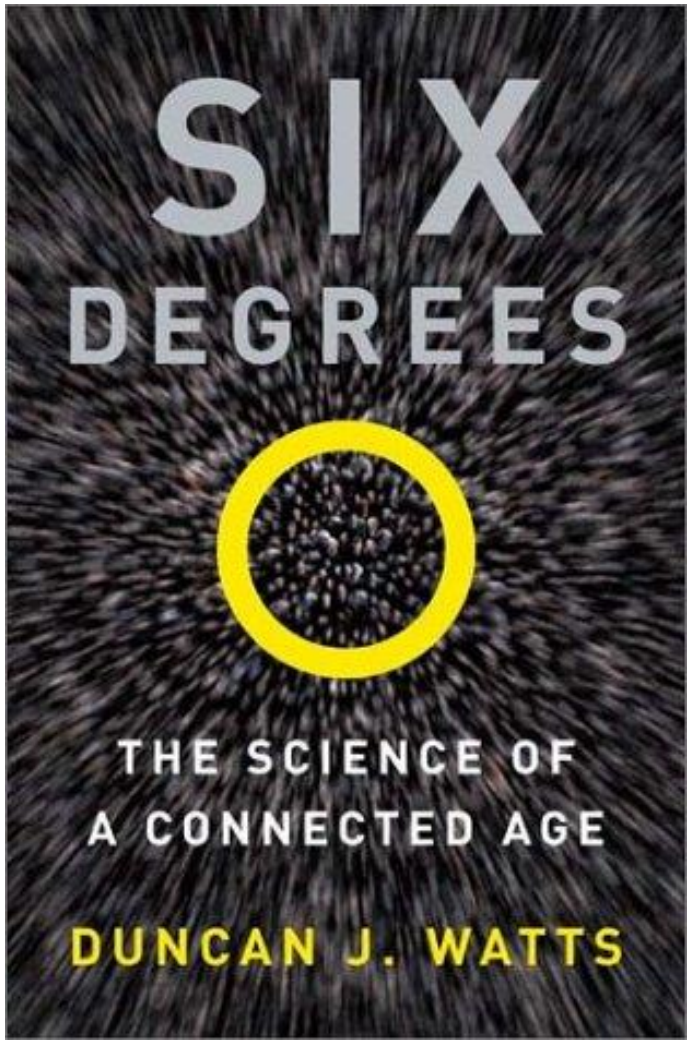
But there is a lot cooking beneath the surface…



Co-create content (or) Place ads?



(or)



Can we explore this?

Recipe

- Cook up a large social network
- Ask specifically chosen seeds to post content
- Track diffusion

$$\boldsymbol{P}(\text{I respond}) = 1, \text{if } \emptyset \text{ of my friends respond},$$
$$\emptyset \in (0, 1)$$

Problem

Explicit time dependence ⇒ Fast simulations are difficult to execute

What did I do?

First, I rubbed R the wrong way

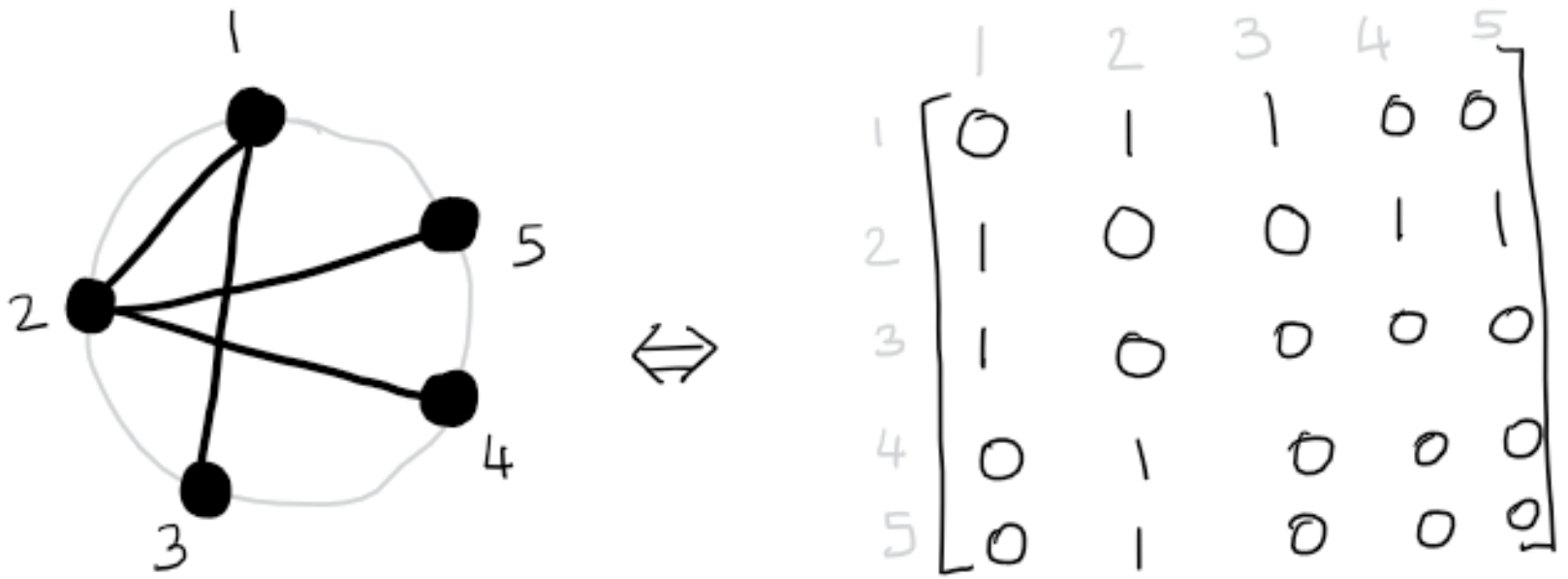
```
for (person in network) {

  if (engagement_status[person] != 1) {

    n_engaged_nbrs <- 0

    for (nbr in neighbors(g, person)) {
      if (engagement_status[nbr] == 1) n_engaged_nbrs <- n_engaged_nbrs + 1
    }

    if (n_engaged_nbrs > threshold[person]) engagement_status[person] = 1
  }
}
```

is a programming language

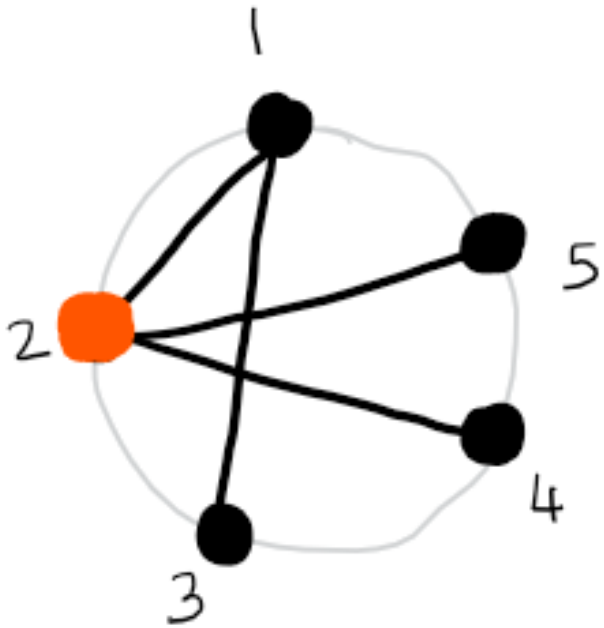
has a

~~is a programming~~ language

So, I vectorized…



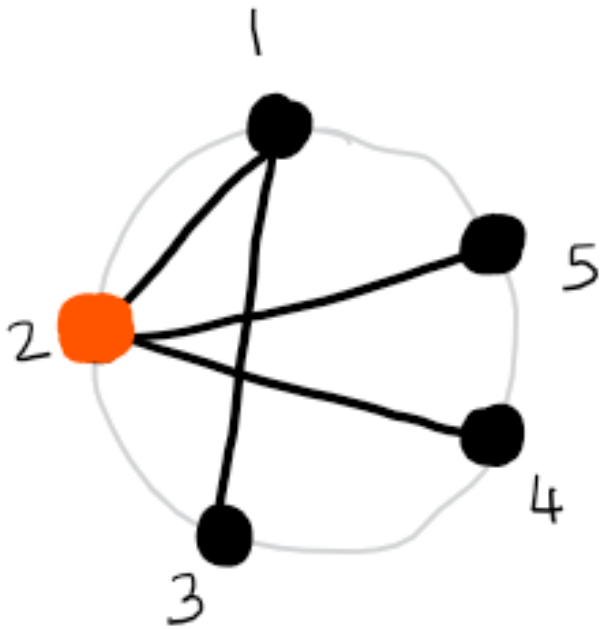
How many neighbors for node 2?



⇔

	1	2	3	4	5
1	0	1	1	0	0
2	1	0	0	1	1
3	1	0	0	0	0
4	0	1	0	0	0
5	0	1	0	0	0

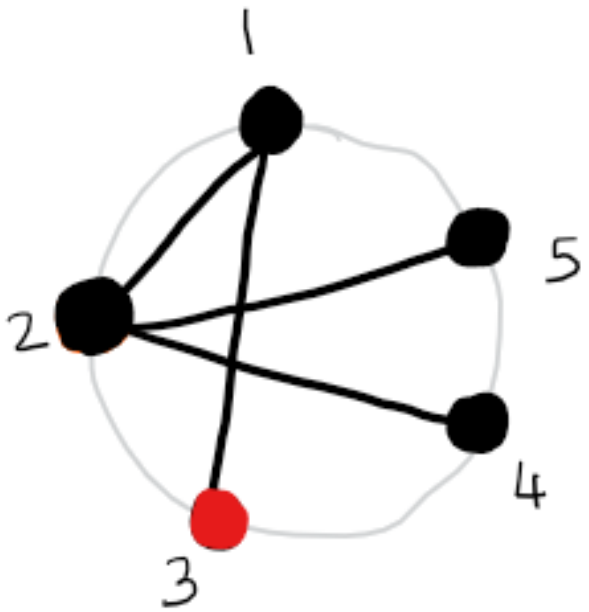
How many neighbors for node 2?



$\Leftrightarrow$

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \end{array} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

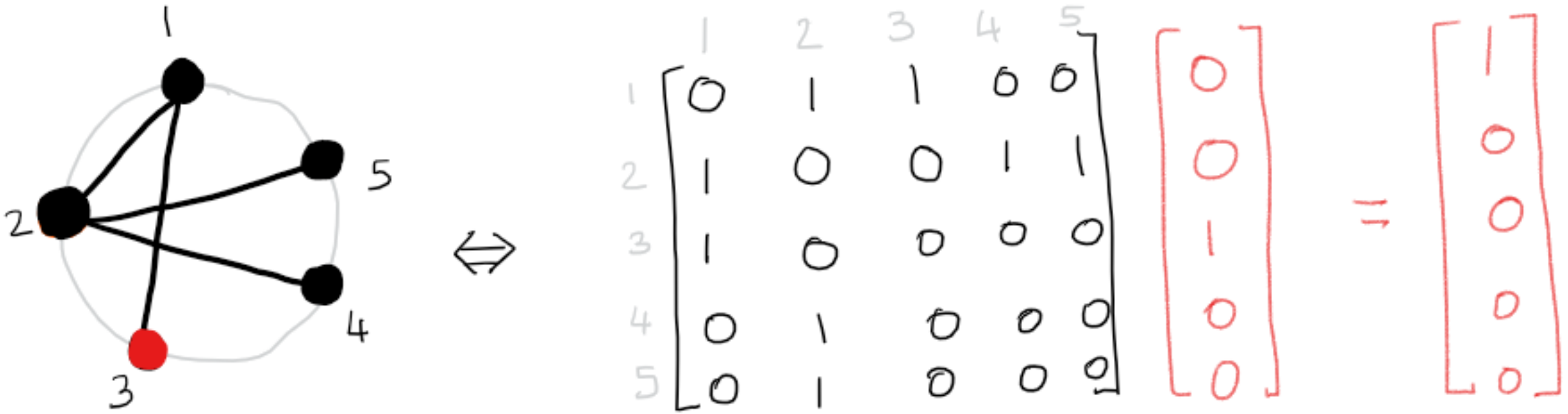
If only node 3 engaged, how many engaged neighbors does each node have?



⇔

	1	2	3	4	5
1	0	1	1	0	0
2	1	0	0	1	1
3	1	0	0	0	0
4	0	1	0	0	0
5	0	1	0	0	0

If only node 3 engaged, how many engaged neighbors does each node have?



So, this becomes…

```
for (person in network) {

  if (engagement_status[person] != 1) {

    n_engaged_nbrs <- 0

    for (nbr in neighbors(g, person)) {
      if (engagement_status[nbr] == 1) n_engaged_nbrs <- n_engaged_nbrs + 1
    }

    if (n_engaged_nbrs > threshold[person]) engagement_status[person] = 1
  }
}
```

…an order of magnitude faster…

```
n_engaged_nbrs <- as.vector(A %*% engagement_status)

# Which nodes are ready to engage?

vuln_persons <- n_engaged_nbrs > threshold

# engage these persons

engagement_status[vuln_persons] <- 1
```

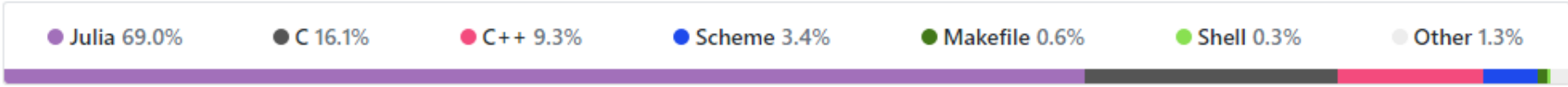
How to write fast R?

1. Solve problems on vectors

2. Follow 1, even when you don’t need to

But, vectorization is unnatural, aka, ‘my head hurts’

Enter julia



Pythonic syntax + speed = Zen

```
function update_person_status!(g, engagement_status::BitArray, threshold::Vector{Int})

    for person in vertices(g)
        if engagement_status[person] == false
            num_engaged_friends = sum([engagement_status[friend] for friend in neighbors(g, person)])
        end

        if num_engaged_friends > threshold[person]
            engagement_status[person] = true
        end
    end

    return nothing

end
```

Enough pseudo-code, show me the real stuff!

What did I learn?

- Social media users have thresholds to act
- Never pay to push posts :P

- Don’t write loops in R, even when it seems impossible ☺
- Julia is a productive C, keep an eye

Questions?

Pavan

https://github.com/pavanGurazada