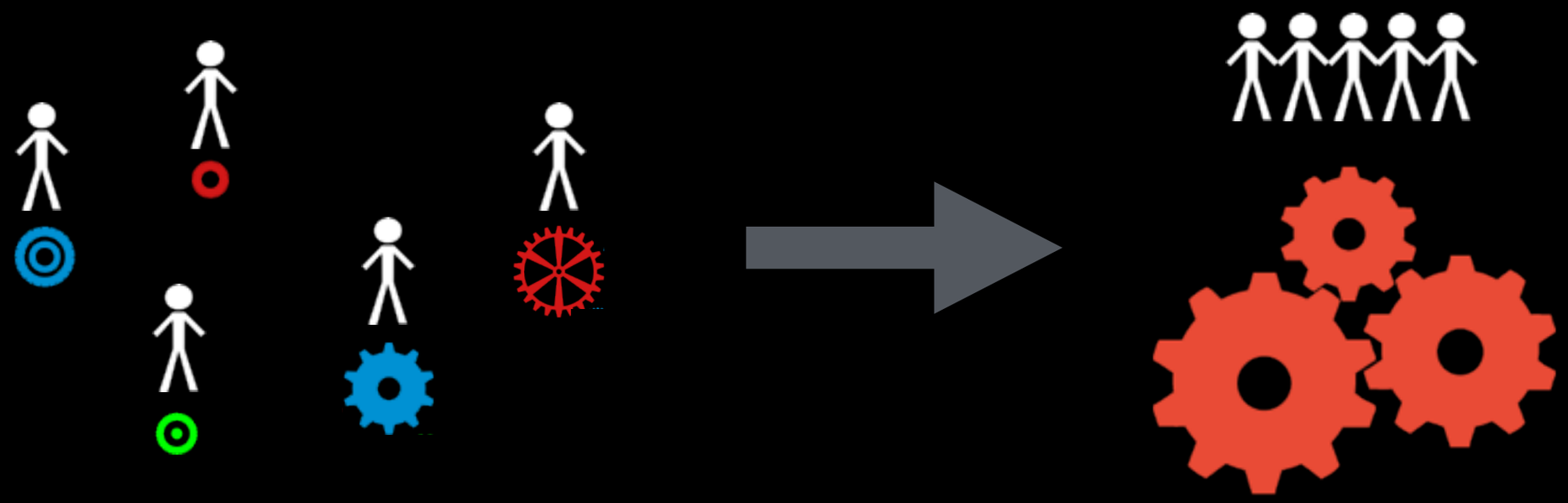
S|S|M

Inference for State Space Models

Modelling complex & dynamic quantities:

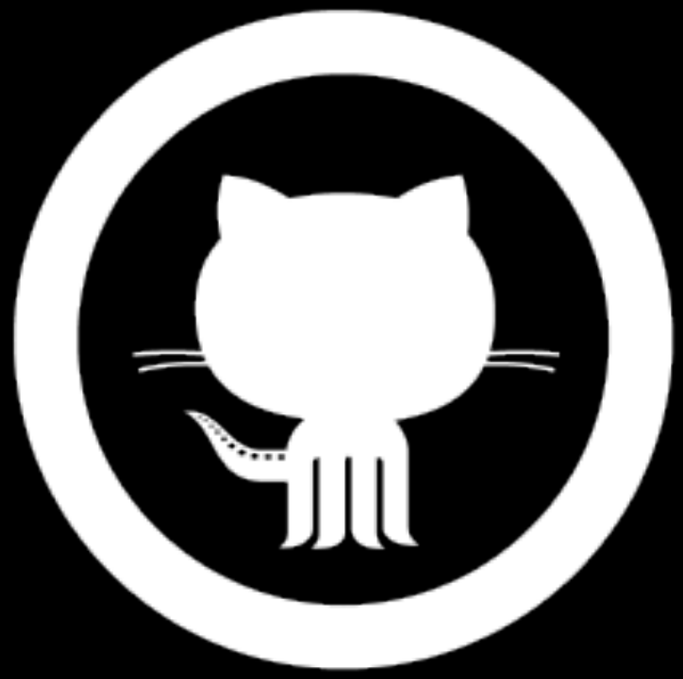
reorganising to meet up with technical challenges.



https://github.com/standard-analytics/ssm

Open Source community

+ you



Sign up for GitHub

https://github.com/

WHY?

Question?
Raise issues

Support!

Keep informed

Objectives 5x3 hours

Build your own compartmental models

Calibrate them against simple and complex data

Study an open problem in epidemiology

Step 0

Is SSM properly installed on your machines?

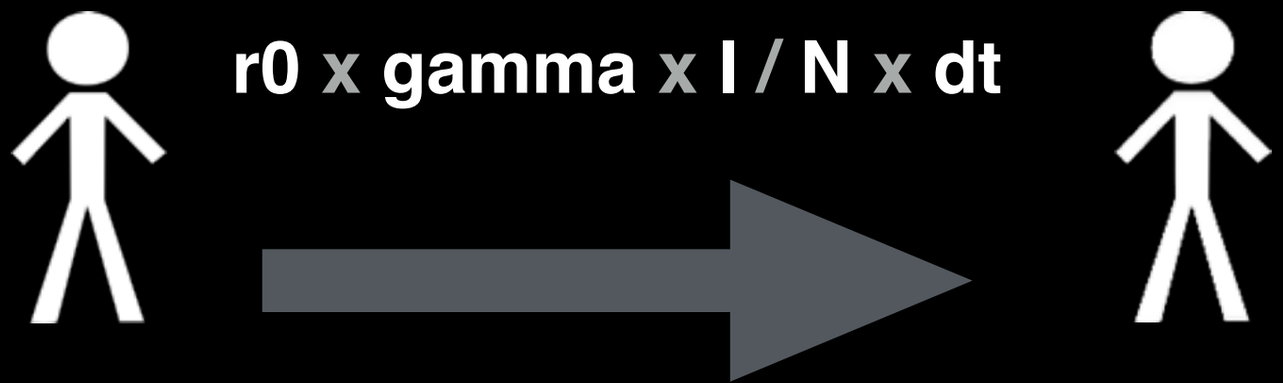
In your terminal: `> ssm -V`

Step 1

Build simple models.

Simulate deterministic and stochastic scenarios from them.

SI Intro



S I

$$dS_t = -r0 \times gamma \times I / N \times S \times dt$$
$$dI_t = r0 \times gamma \times I / N \times S \times dt$$

SIR Intro

$$dS_t = -r0 \times gamma \times I / N \times S \times dt$$
$$dI_t = r0 \times gamma \times I / N \times S \times dt - v \times dt$$
$$dI_t = v \times dt$$

SIR First steps

change directory

```
> cd SIR-city
> ls
```

list content

Saint-Fuscien
Amiens
Paris

package.json???

JSON: JavaScript Object Notation

JSON

double-quotes only!

Lists: `[ ‘‘a’’, 3, … ]`

Objects: `{ ‘‘a’’: ‘‘A’’, ‘‘b’’: 3, …}`

Flexible, readable format.

Popular open standard.

package.json

Used to distribute code & data through the node package manager (npm)

Standard keys:

```
“name”, “description”, “keywords”,
“licenses”, “version”, “resources”
```

SIR Let’s get to it

Closer look at the “model” object.

SIR Simulation

```
> ssm install package.json
> cd bin
> ./simul --help
> cat ../package.json | ./simul --traj
```

→ X_0.csv

SIR Plot X_0.csv

Open TD.R in R

!!! Set TD-STRU to be Working Directory

Plot X_0.csv with ssm.plot.X

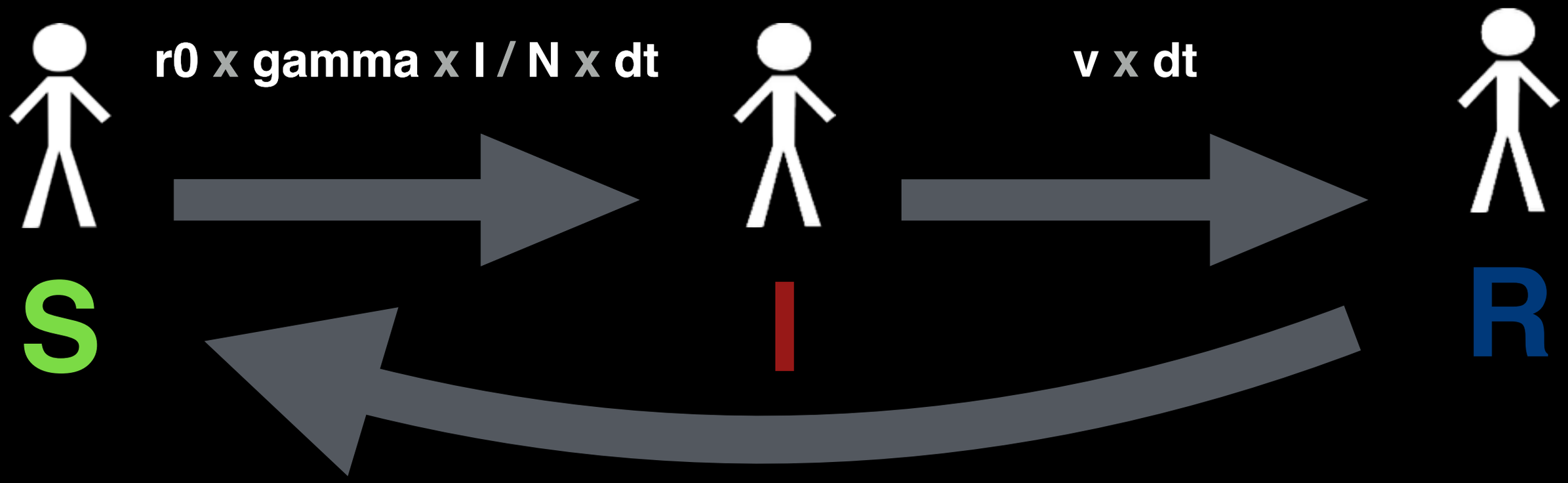
SIR Explore

Explore different values of R0 and d

Under which minimal conditions does an epidemic burst?

According to the SIR model, what will be the number of susceptibles on December 13th, 2012?

SIRS Make your own model



$$dS_t = -r0 \times gamma \times I / N \times S \times dt + k \times dt$$
$$dI_t = r0 \times gamma \times I / N \times S \times dt - v \times dt$$
$$dI_t = v \times dt - k \times dt$$

SIRS Make your own model

What should qualitatively be the impact of immunity loss on the number of susceptibles on December 13th, 2012?

SIRS Make your own model

Duplicate and rename SIR.
Modify it to obtain an SIRS model.

reactions **+**
```
…,
{"from": "R", "to": "S", "rate": "k", "description":"recovery"}
```

inputs **+**
```
…,
{ "name": "k",  "description": “rate of immunity loss",
"data": {"resource":  "pr_k"}, "transformation": "1/pr_k",
"to_resource": "1/k" },
```

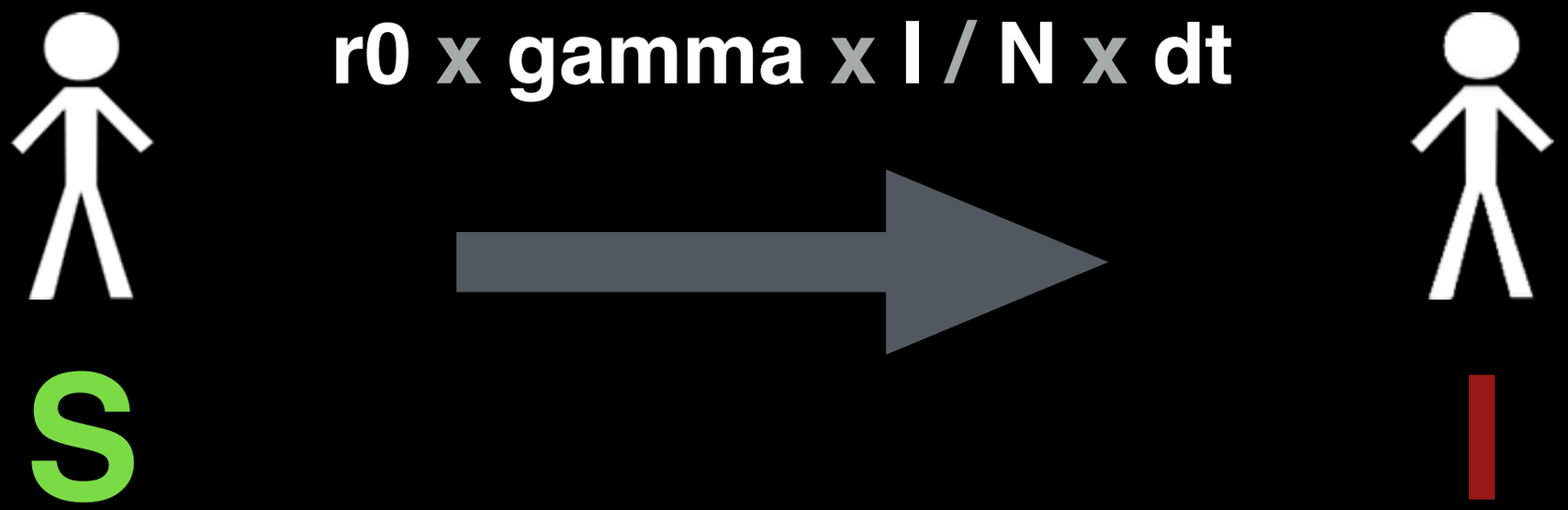
resources **+**
```
{  "name": "pr_k", "description": “duration of immunity",
"data": { "distribution": "fixed", "value": 250.0 } },
```

Ok, but...

Aren’t these deterministic scenarios a bit too simplistic?

Noise Poisson process formalism

For every individual:



infection is a random process

Noise Poisson process formalism

For S_t individuals:

$$\text{p(n infections)} \approx \binom{S_t}{n} \text{r0} \times \text{gamma} \times \text{I} / \text{N} \times \text{dt}$$

→ tractable stochastic model

For more details, see: Breto et al (2009). *Time series analysis for mechanistic models.*

Noise SDE formalism

Going further, following Ethier & Kurtz 1986:

drift

volatility

dSt = - r0 x gamma x I / N x St x dt - sqrt(r0 x gamma x I / N x St) dBt

dIt = r0 x gamma x I / N x St x dt + sqrt(r0 x gamma x I / N x St) dBt

→ ***Diffusion approximation***

For more details, see: Dargatz (2007). *A diffusion approximation for an epidemic model.*

To remember

psr best tractable approximation

sde continuous approximation
classical mathematical object
theory only for large populations

SIR Let's try with noise

```
> cat ../package.json | ./simul psr --traj
```

```
> cat ../package.json | ./simul sde --traj -I 1
```

Run id

Plot and compare X_0.csv and X_1.csv

SIR Let's try with noise

Number of particles

```
> cat ../package.json | ./simul psr --traj -J 3
```

```
> cat ../package.json | ./simul sde --traj -I 1 -J 3
```

Run id

More particles

SIR Let's try with noise

Number of particles

```
> cat ../package.json | ./simul psr --traj -J 1000
```

```
> cat ../package.json | ./simul sde --traj -I 1 -J 1000
```

Run id

Even more particles

SIR Let's try with noise

Number of particles

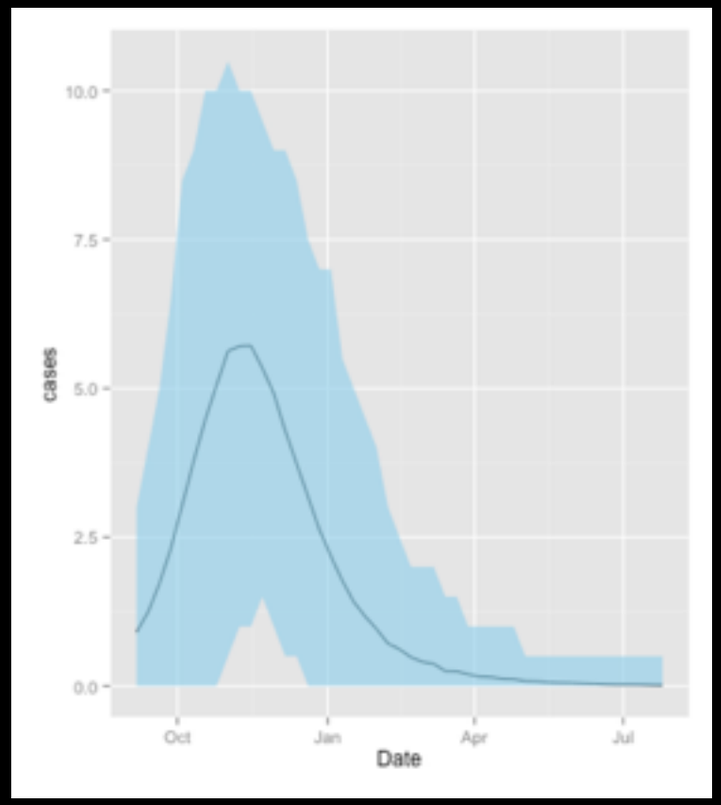
```
> cat ../package.json | ./simul psr --hat -J 1000
```

```
> cat ../package.json | ./simul sde --hat -I 1 -J 1000
```

Run id

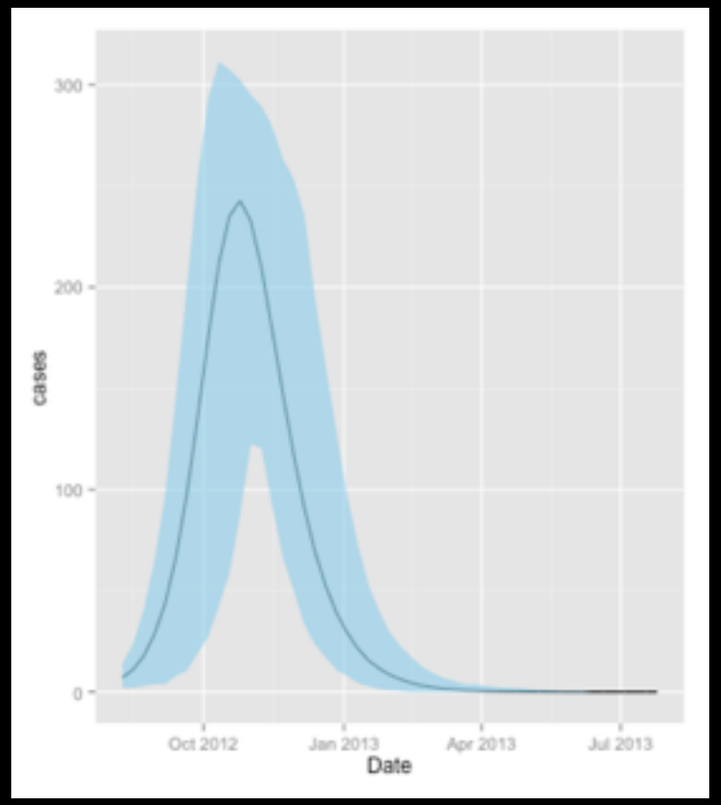
Generate trajectory confidence intervals

→ use ssm.plot.hat



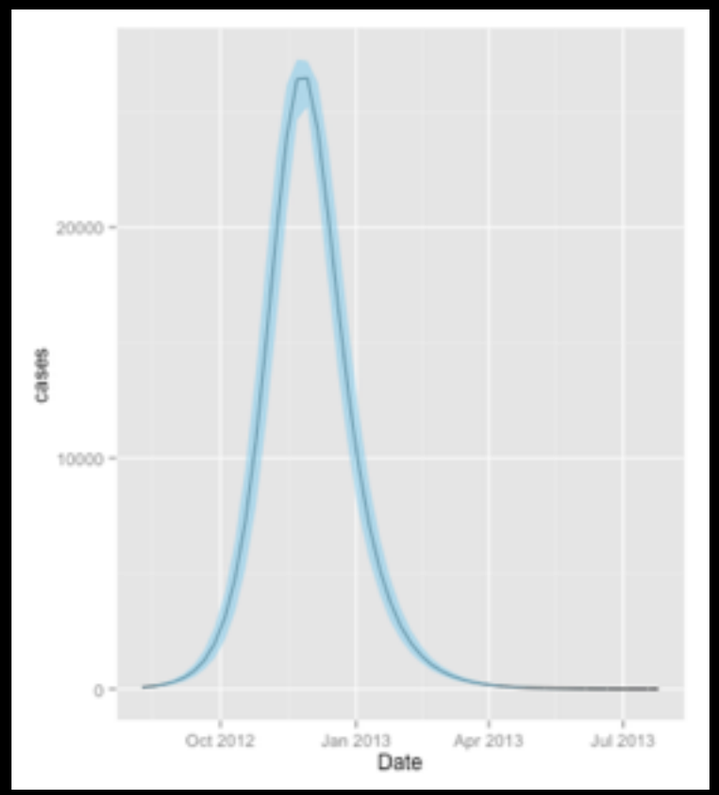
Saint-Fuscien

1’000 inhabs.



Amiens

100’000 inhabs.



Paris

10M inhabs.