Fake it ‘till you make it

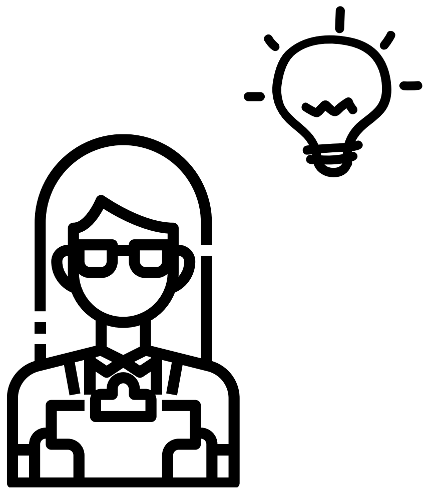
Generating synthetic data

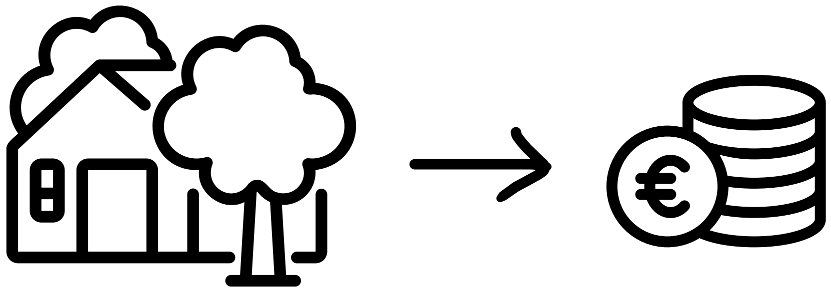
Erik-Jan van Kesteren
Thom Volker

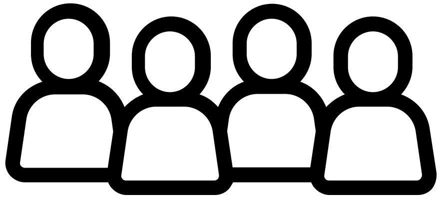
Utrecht University
ODISSEI Social Data Science team

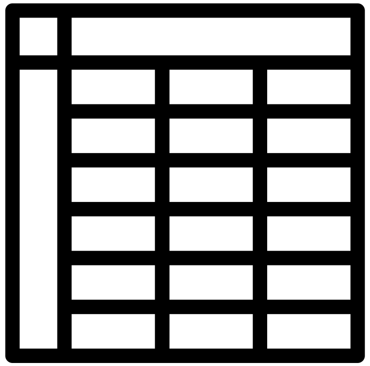
thomvolker.github.io/OSWS_Synthetic

Imagine..

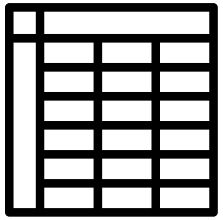
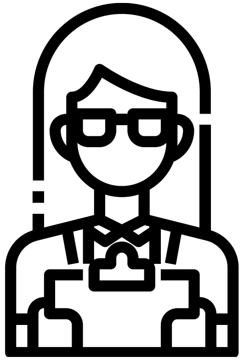
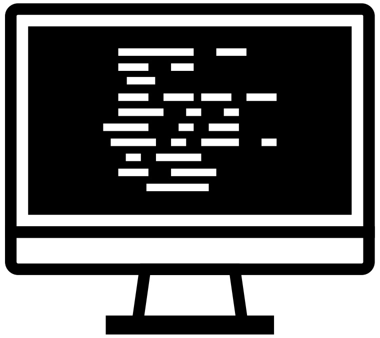








- Where do you live?
- How long have you lived there?
- What do you earn?
- How much do you spend on gifts for your friends?

“More generous gifting behaviour in greener neighbourhoods”


OPEN MATERIALS

```
my_data <- read_csv("super_private_data_file.csv")
```

Open data not allowed, options:

- Data just not available, good luck
- “Data available upon reasonable request”
- Data is part of a large project with data access procedures

I just want to check out the script to learn from the cool analysis!

Solution: publish open synthetic data with your open materials

What will we do in this tutorial?

Lecture

A primer on synthetic data

Practical

Creating synthetic data in R

Lecture

Privacy & utility for synthetic data

Practical

Assessing utility & disclosure control in R

A primer on synthetic data

Synthetic data (EJ’s definition)

Synthetic data is generated from a model
As opposed to real, natural, collected data

fake data
generated data
simulated data
digital twin
public use file

To create synthetic data, you need a **generative model**

Generative model

$$p(\boldsymbol{X}|\theta)$$

- A model for data $\boldsymbol{X}$
- Has parameters (θ)
- You can fit / estimate / learn θ based on real data

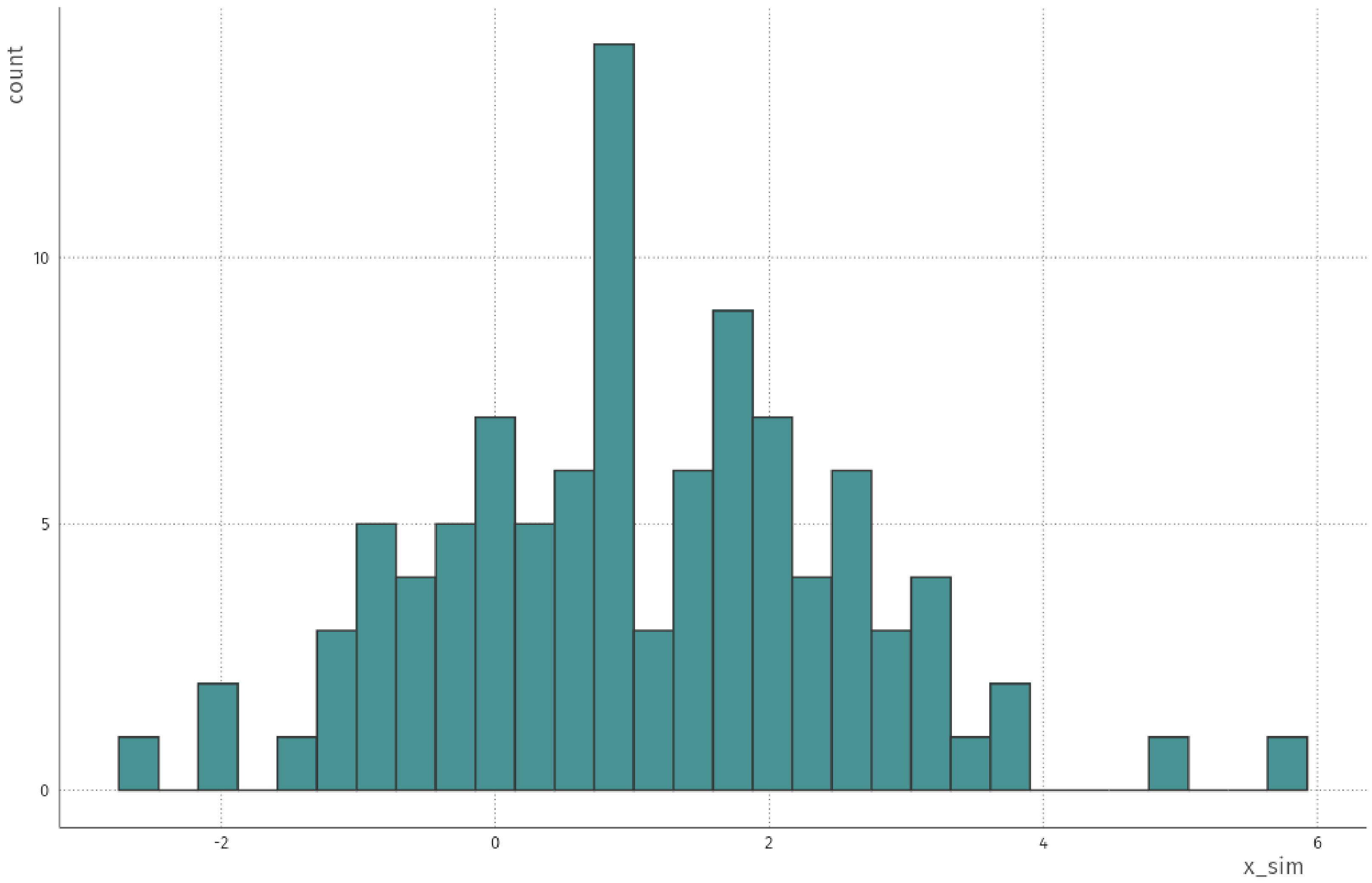
- Examples:
  - A normal distribution with parameters $\theta = \mu, \sigma$
  - A histogram with bins and proportions
  - A generative adversarial network with a million parameters

Generative model

In R code:

```r
# parameters
mu <- 1.0
sigma <- 1.5

# generate data
x_sim <- rnorm(100, mean = mu, sd = sigma)
```


count
10
5
0
-2
0
2
4
6
x_sim

Generative model

Today we will fit two types of generative models:

Parametric: Assume that variables (conditionally) follow a certain distributions (e.g., Bernoulli, Normal, Exponential, ...)

Non-parametric: Do not assume certain distributions, use a *machine learning*® method

There are infinitely many more generative models. This is an active field of research

Software

There are many ways creating generative models & synthetic data

- Manually creating a csv file ☺
- Metasynth (https://github.com/sodascience/metasynth)
- Synthpop (https://synthpop.org.uk/)
- MICE (https://amices.org/mice/)
- Synthetic Data Vault (https://sdv.dev/)
- DataSynthesizer (https://github.com/DataResponsibly/DataSynthesizer)

Software

There are many ways creating generative models & synthetic data

- Manually creating a csv file ☺
- Metasynth (https://github.com/sodascience/metasynth)
- Synthpop (https://synthpop.org.uk/)
- **MICE (https://amices.org/mice/)**
- Synthetic Data Vault (https://sdv.dev/)
- DataSynthesizer (https://github.com/DataResponsibly/DataSynthesizer)

The MICE generative model

The MICE generative model

- MICE: Multiple Imputation by **Chained Equations**
- Trick to create multivariate generative model through univariate prediction models

$$p(X_1, X_2, X_3) = p(X_1|X_2, X_3)p(X_2|X_1, X_3)p(X_3|X_1, X_2)$$

- Focus on making good univariate predictions

The MICE generative model

Many prediction methods available in MICE, we will use two types:

Parametric

Linear & Logistic regression

Nonparametric

Classification and regression trees

Built-in univariate imputation methods are:

`pmm`	any	Predictive mean matching
`midastouch`	any	Weighted predictive mean matching
`sample`	any	Random sample from observed values
`cart`	any	Classification and regression trees
`rf`	any	Random forest imputations
`mean`	numeric	Unconditional mean imputation
`norm`	numeric	Bayesian linear regression
`norm.nob`	numeric	Linear regression ignoring model error
`norm.boot`	numeric	Linear regression using bootstrap
`norm.predict`	numeric	Linear regression, predicted values
`lasso.norm`	numeric	Lasso linear regression
`lasso.select.norm`	numeric	Lasso select + linear regression
`quadratic`	numeric	Imputation of quadratic terms
`ri`	numeric	Random indicator for nonignorable data
`logreg`	binary	Logistic regression
`logreg.boot`	binary	Logistic regression with bootstrap
`lasso.logreg`	binary	Lasso logistic regression
`lasso.select.logreg`	binary	Lasso select + logistic regression
`polr`	ordered	Proportional odds model
`polyreg`	unordered	Polytomous logistic regression
`lda`	unordered	Linear discriminant analysis
`2l.norm`	numeric	Level-1 normal heteroscedastic
`2l.lmer`	numeric	Level-1 normal homoscedastic, lmer
`2l.pan`	numeric	Level-1 normal homoscedastic, pan
`2l.bin`	binary	Level-1 logistic, glmer
`2lonly.mean`	numeric	Level-2 class mean
`2lonly.norm`	numeric	Level-2 class normal
`2lonly.pmm`	any	Level-2 class predictive mean matching

Let’s get started!

Icons from the noun project

Scientist by Justicon
Idea by Icon
house tree by LUTFI GANI AL ACHMAD
Euro by Larea
people by Alice Design
Table by Alex Burte
Hacking by Alfredo
Paper by Egi Maulana
Scientist 2 by Justicon
Question by Anggara Putra

https://thenounproject.com/