

“Fake it till you make it” : an introduction to synthetic data

Joachim Ganseman - Smals Research

webinar - 01/12/2022

Smals Research 2022



**Innovation with
new technologies**



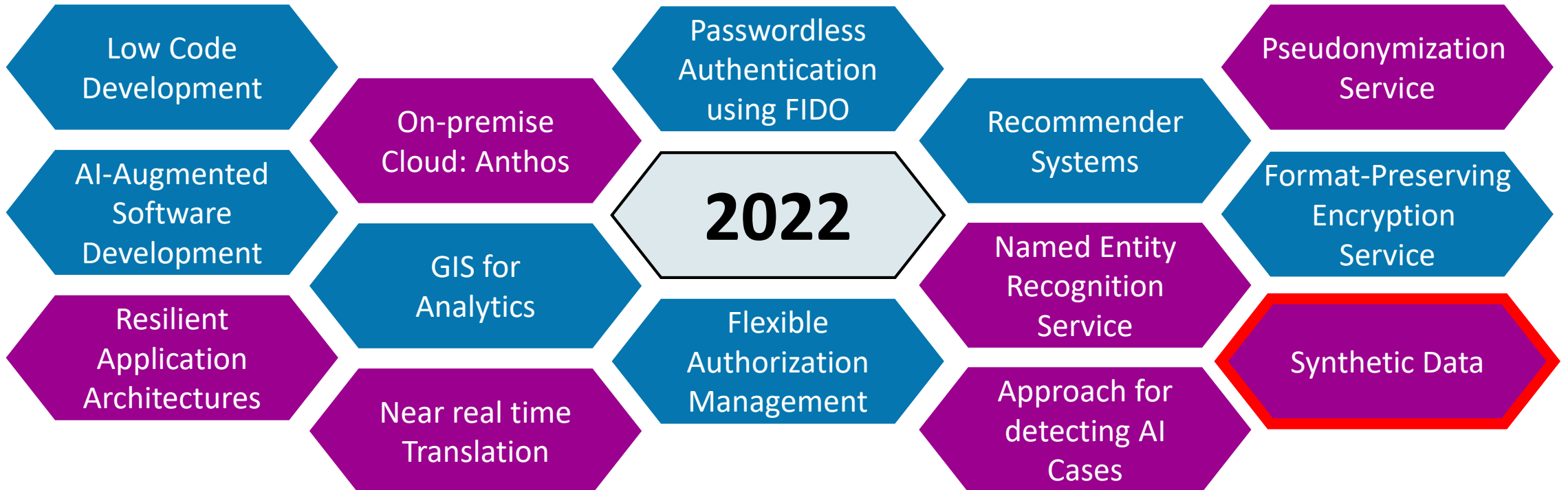
**Consultancy
& expertise**



**Internal & external
knowledge transfer**



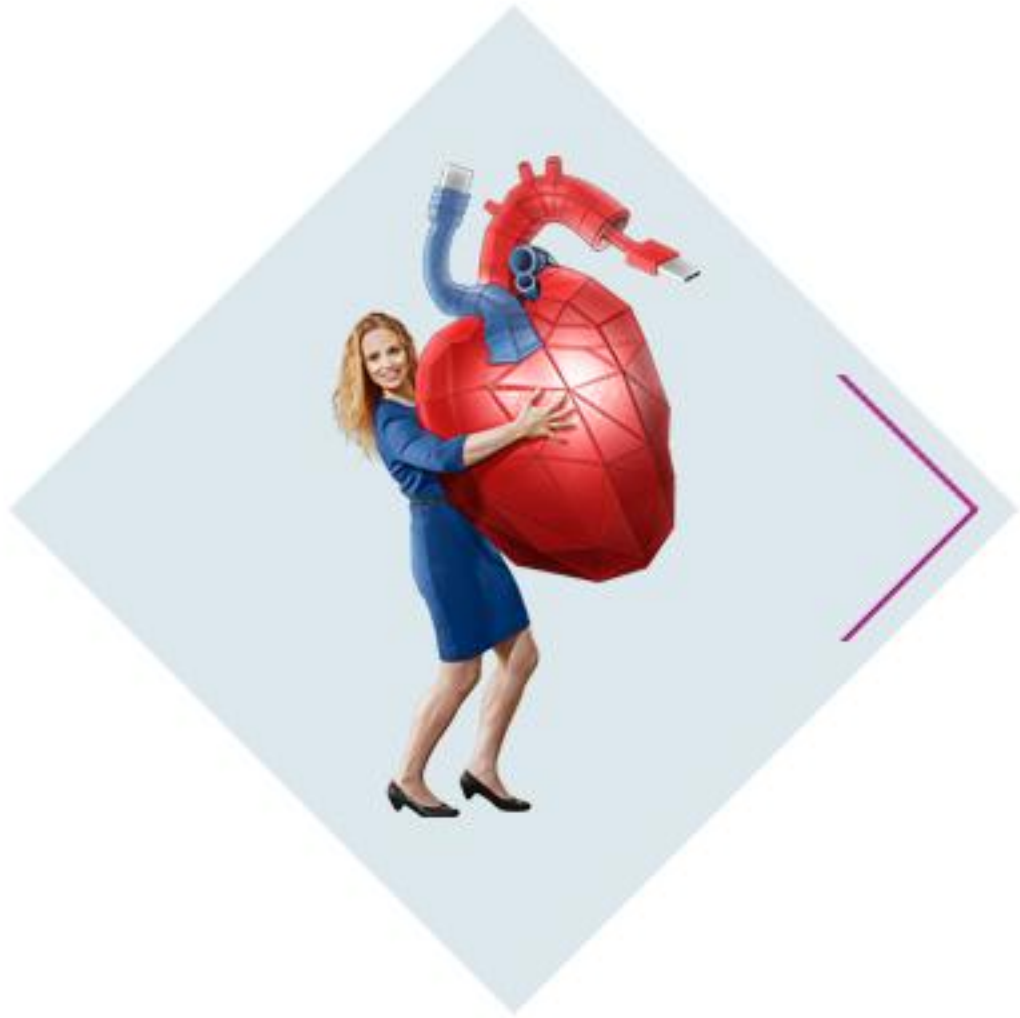
**Support for
going live**



TODAY



- Focus on our sector
- Synthetic data: What, why and how
- Practical considerations, pitfalls and caveats based on a real proof-of-concept experience
- Using and evaluating synthetic data
- Open Source vs. Private Market
- Future directions



Introduction

A synthetically generated picture

“A synthetic dataset consists of fictitious replacement data, that mimics the structure and distribution of the original data.” [as imagined by [DALL-E 2](#)]



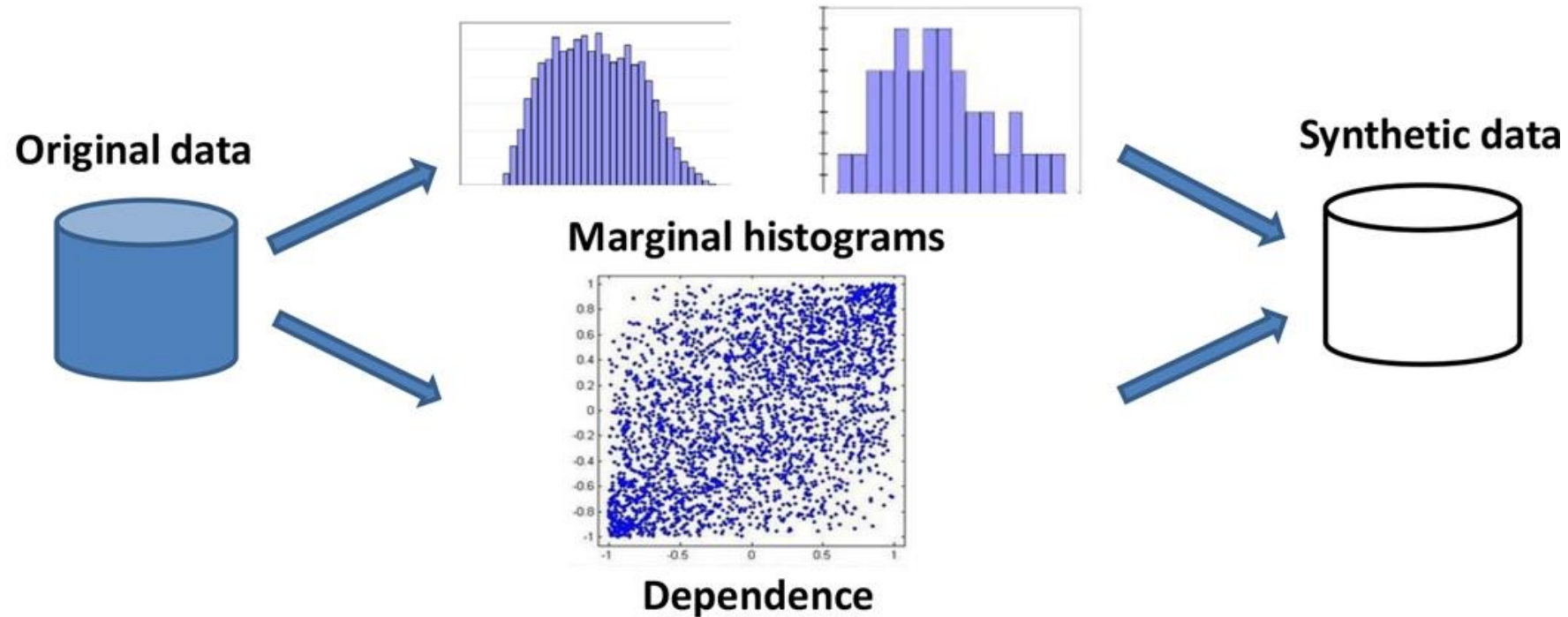
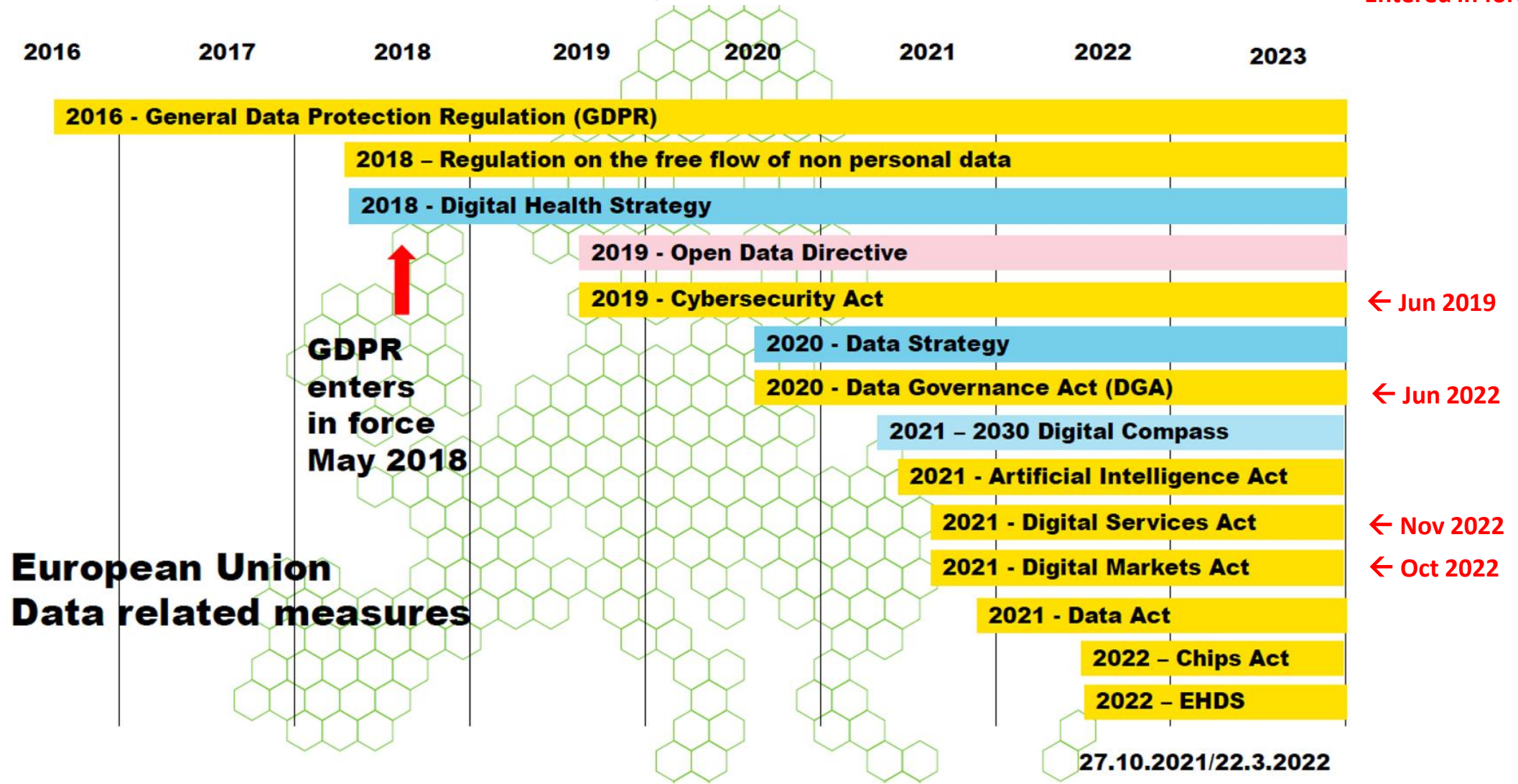


Image © Haoran Li, Li Xiong, Lifan Zhang, and Xiaoqian Jiang,
“DPSynthesizer: Differentially Private Data Synthesizer for Privacy Preserving Data Sharing”

Create a **fictitious dataset** that **mimics** an actual dataset by **learning** its structure and **generating** plausible datapoints.

Regulatory developments impacting data access and processing

Entered in force:



Source: "Towards the European Health Data Space", Markus Kalliola, TEHDAS

- **Regulatory requirements** makes (re-)use of sensitive data a headache
 - “Sufficient / adequate” technical and organizational measures
 - Explicit permission from data subjects
 - Obligations to anonymize / delete data
 - Writing impact assessments, keeping registries, ...
- Improve on **existing bad practices**
 - Production data in test / dev environments
 - Lack of testing due to lack of (realistic) data
 - “Here’s a copy but don’t tell anyone”
- Real data can be **unbalanced, biased** or **expensive** to collect

- Make a **realistic alternative to (sensitive) data** available
 - As a data controller, to external parties for research
 - As a company, to the architects, developers and testers that build your software
 - As a researcher, to the outside world (reproducibility)
 - ...
- Realistic simulations / generate **test data**
- **Data augmentation** for ML applications
- ...

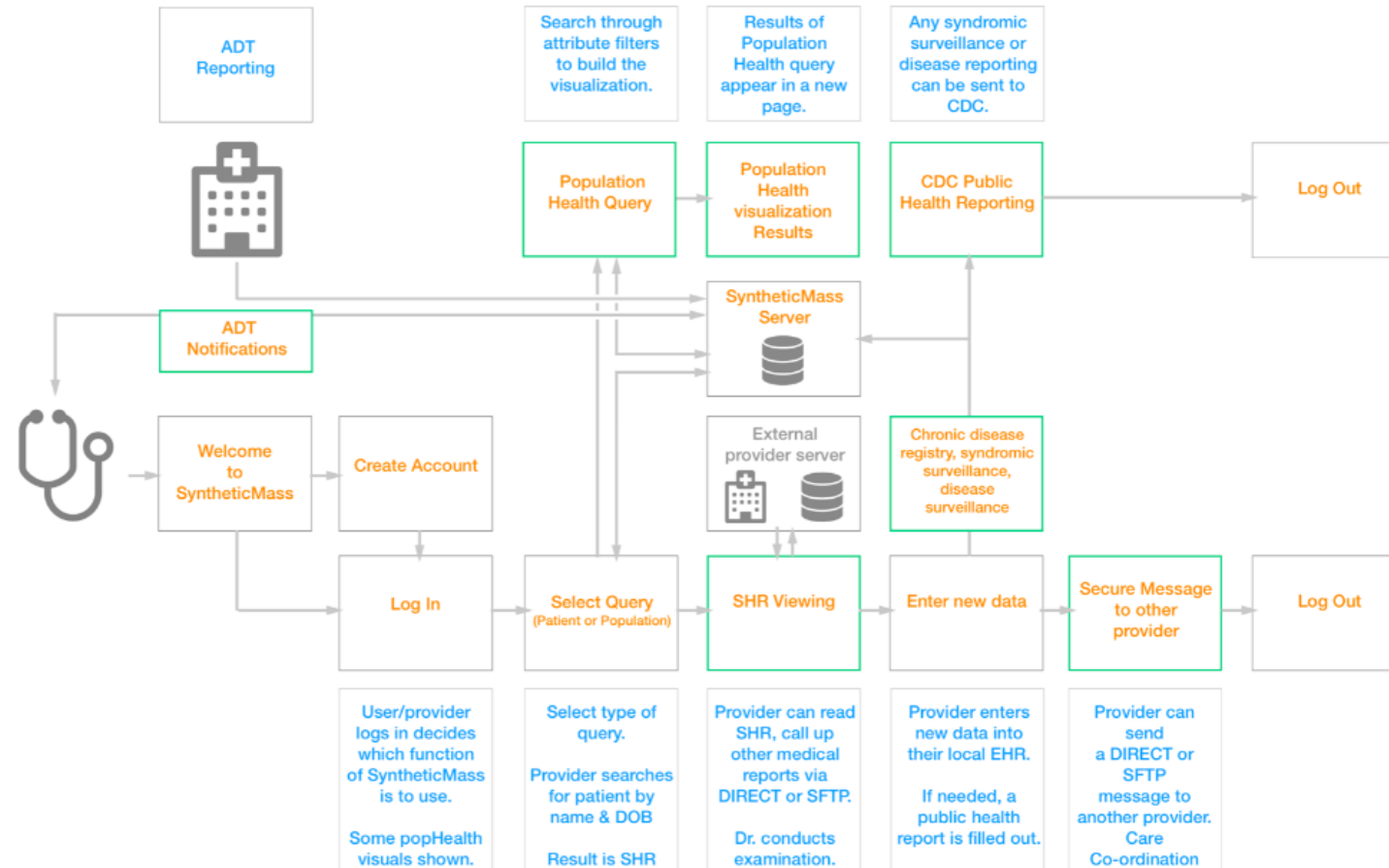
⚠ Synthetic data generation is generally a **one-way pipeline**
→ Can no longer be linked to original/real data
→ Involves randomness: 2 runs = 2 different results

Example: SyntheticMass

- “[SyntheticMass](#) is a model of synthetic residents of the state of Massachusetts, with [statistically plausible] artificial health records for the fictional residents.”
- Tests various aspects of an eHealth-system

SyntheticMass Web Flow Map V 1.0 Provider Journey

- Notes
- Web pages
- External link
- Service



Example: SyntheticMass

Build a map (Select from filters)

Home Synthetic Mass Dashboard

Filters:

Demographics Condition Medication Provider Date Location

Age: Full-time Part-time
 Retired None

RESET BUILD

Birth Sex: Male Female

SAVE

Ethnicity: Pacific Islander

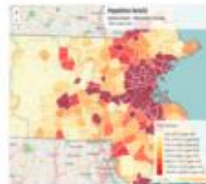
Income: \$0-25 \$25-50K \$75-100K
 \$100-125K \$125-150K \$150-175+

Relationship: Married Single

Employment: Full-time Part-time
 Retired None

Veteran Status: Veteran Non-Veteran

Saved Visualizations:



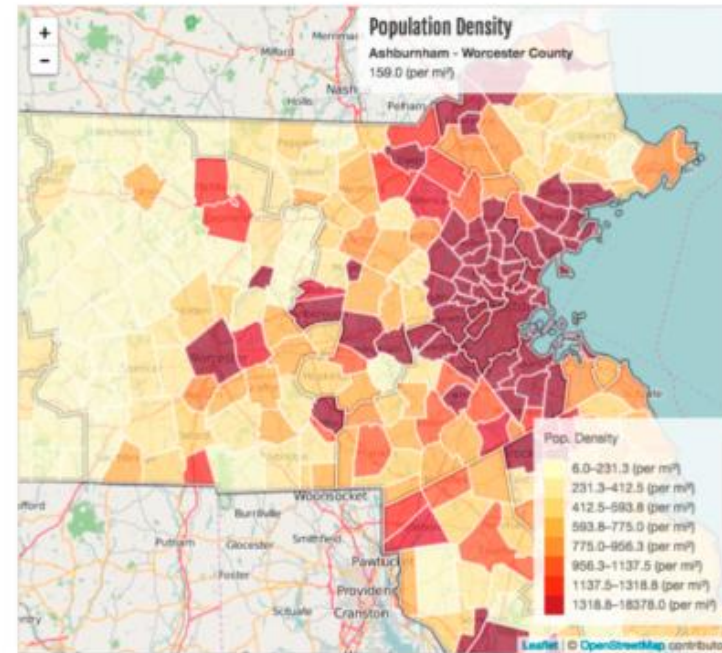
Diabetes by town



Cancer deaths by population



Drug overdose by county



Population Density

Number of residents per sq. mile

Region Minor Civil Division

Type

Data Set Computed from Census ACS Data

Median 547.0 (per sq./mi.)

Max Somerville Minor Civil Division: 183780.0 (per sq./mi.)

Min Gosnold Minor Civil Division: 6.0 (per sq./mi.)



Approaches per data type

1. Independent free text

- Address lines, names, formatted numbers, ...
- Software libraries: [Faker](#) / [Mimesis](#) / [Benerator](#)
- Flexible use in **scripts**
 - Generate new data
 - Shuffle existing data
 - Add your own extensions

```
>>> Faker.seed(0)
>>> for _ in range(5):
...     fake.vat_id()
...
'BE6048764759'
'BE8242194892'
'BE1157815659'
'BE8778408016'
'BE9753513933'
```

```
from mimesis import Generic
from mimesis.locales import Locale
g = Generic(locale=Locale.ES)

g.datetime.month()
# Output: 'Agosto'

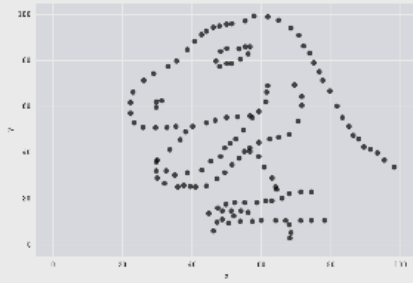
g.code.imei()
# Output: '353918052107063'

g.food.fruit()
# Output: 'Limón'
```

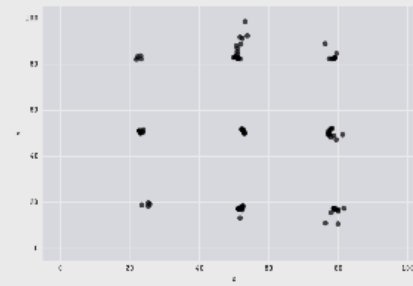
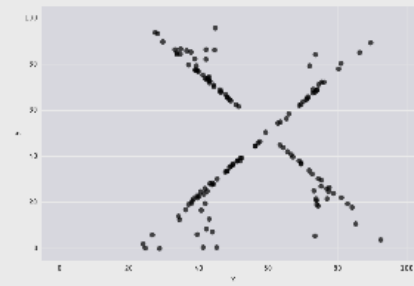
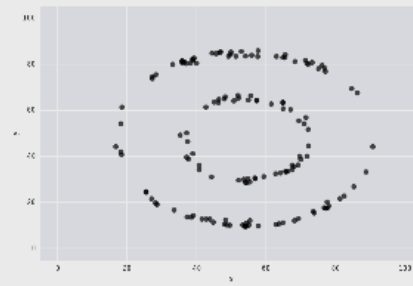
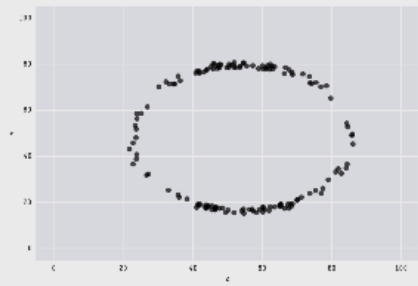
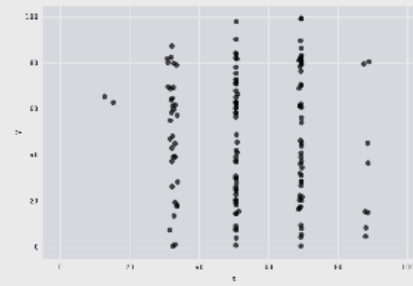
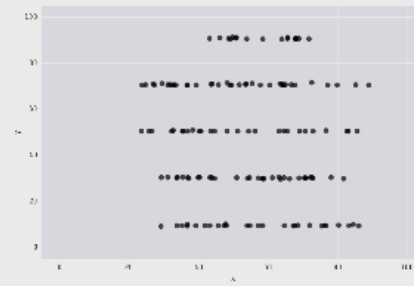
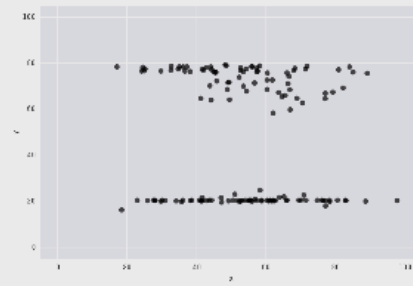
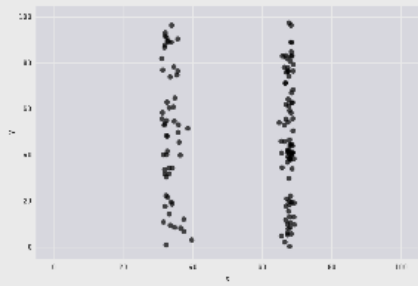
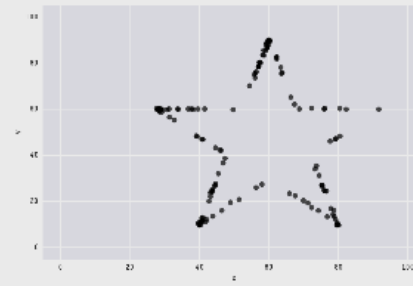
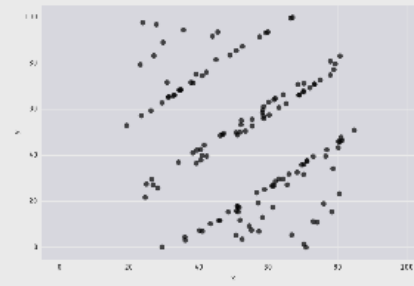
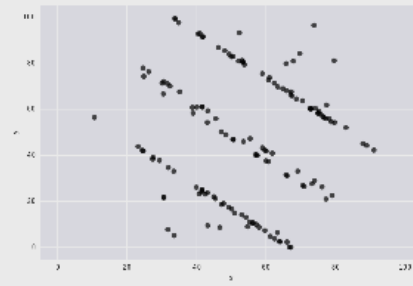
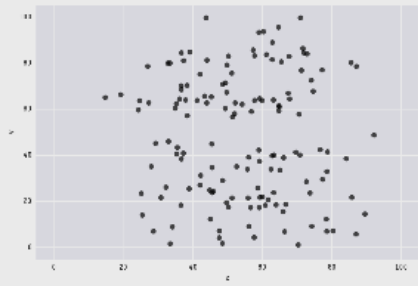
```
from faker import Faker
fake = Faker('it_IT')
for _ in range(10):
    print(fake.name())

# 'Elda Palumbo'
# 'Pacifico Giordano'
# 'Sig. Avide Guerra'
# 'Yago Amato'
# 'Eustachio Messina'
# 'Dott. Violante Lombardo'
# 'Sig. Alighieri Monti'
# 'Costanzo Costa'
# 'Nazzareno Barbieri'
# 'Max Coppola'
```

2. Numerical / Categorical data



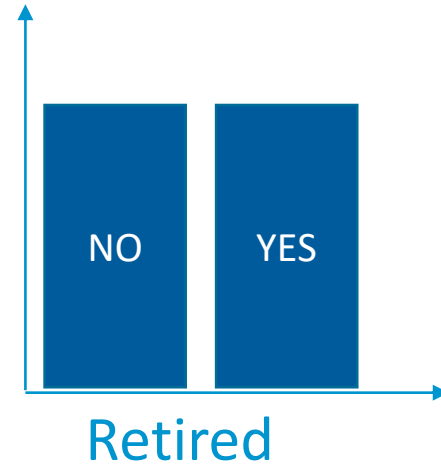
X Mean: 54.26
Y Mean: 47.83
X SD : 16.76
Y SD : 26.93
Corr. : -0.06



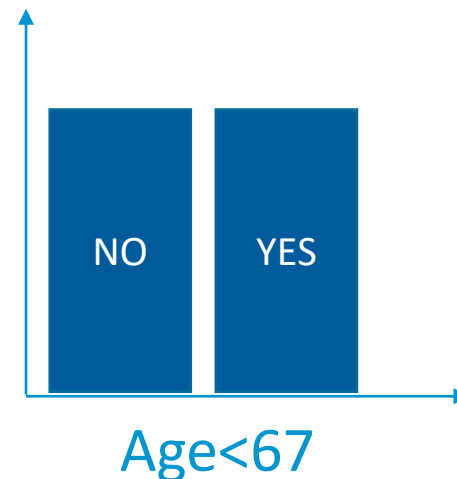
Similar “summary statistics”
≠ good mimicking of
original data

Conservation of distributions \neq conservation of correlations

Age	Retired
15	FALSE
24	FALSE
50	FALSE
68	TRUE
72	TRUE
88	TRUE



Age	Retired
88	FALSE
68	TRUE
50	FALSE
15	TRUE
72	FALSE
24	TRUE



- 1. Learn (joint) distributions from original data → statistical model
- 2. Repeatedly “sample” this model

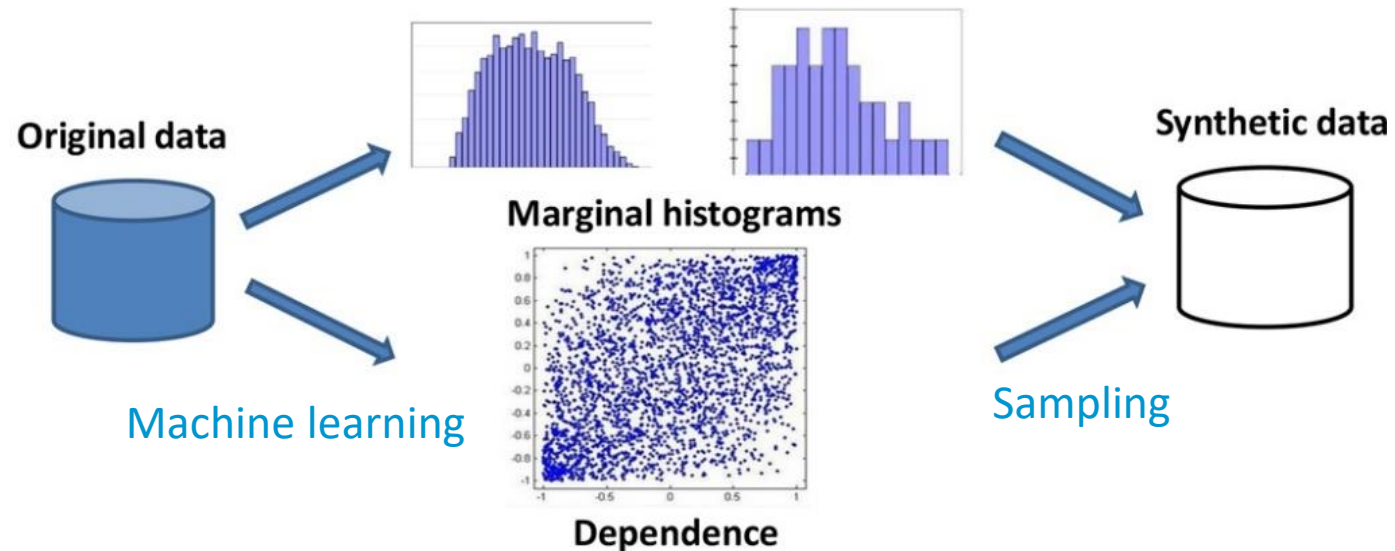
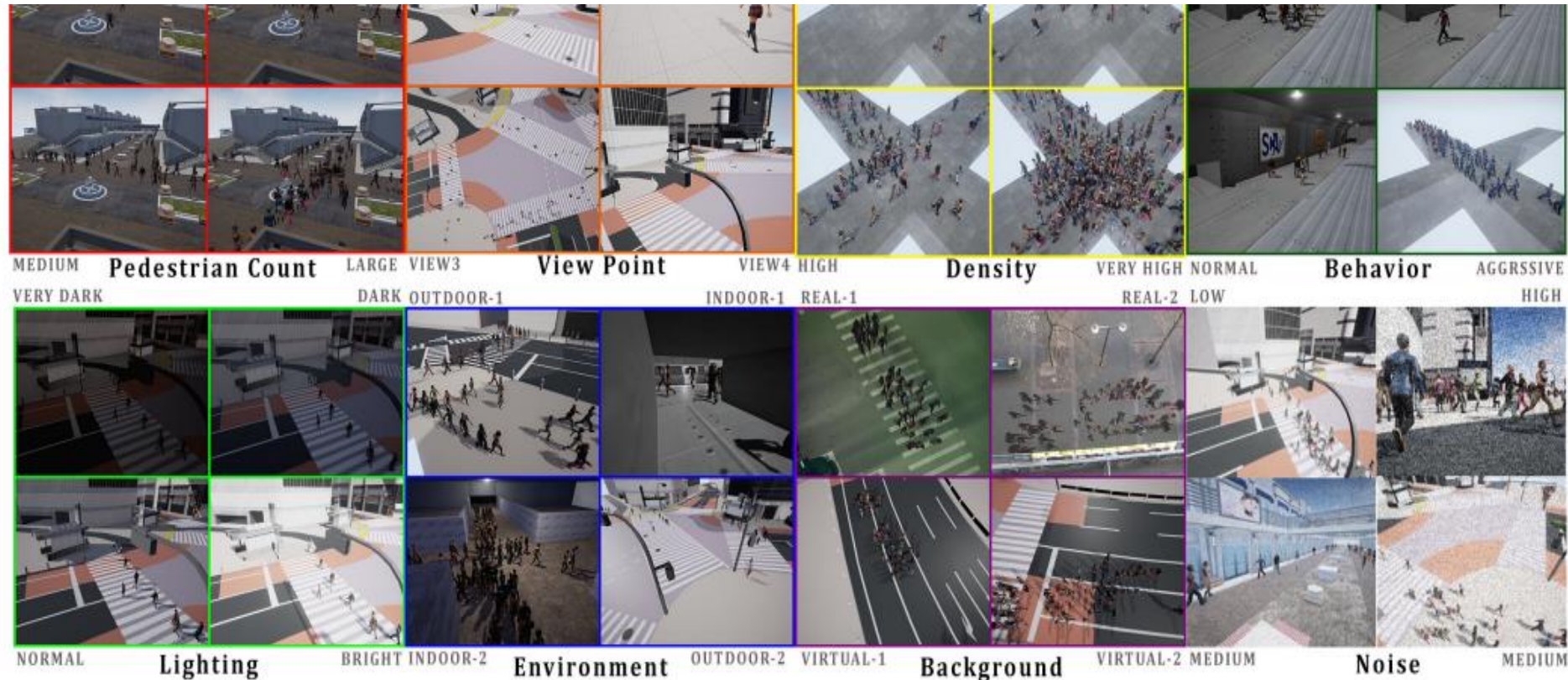


Image © Haoran Li, Li Xiong, Lifan Zhang, and Xiaoqian Jiang,
“DPSynthesizer: Differentially Private Data Synthesizer for Privacy Preserving Data Sharing”

- Sample **conditional distributions** to generate representative subsets

3. Simulation of the generative process

- Generate data for rare or expensive events
- Create annotated datasets for machine learning



3. Simulation of the generative process

Agent-based Modeling

- Complex dynamic systems (e.g. physics/biology simulations)
- Generate interaction data
- Tools: specialized frameworks – Repast (C++), MASON (Java), Mesa (Python), ...

Virtual Environments

- Robotics, VR, self-driving, ...
- Generate many different scenarios
- Tools: 3D engines – Unity3D, GTA, X-Plane, ...

Synthesizers

- Audio, speech, generative art, ...
- Generate multimedia from symbolic representations
- Tools: text-to-speech systems, MIDI, WaveNet, Processing, ...

See also: <https://arxiv.org/abs/1909.11512>



In practice

Let's take a dataset and pick a software library:

	age	workclass	fnlwgt	education	marital-status	occupation	relationship	race	sex	hours-per-week	native-country	capital	income
0	39	State-gov	77516	Bachelors	Never-married	Adm-clerical	Not-in-family	White	Male	40	United-States	2174	<=50K
1	50	Self-emp-not-inc	83311	Bachelors	Married-civ-spouse	Exec-managerial	Husband	White	Male	13	United-States	0	<=50K
2	38	Private	215646	HS-grad	Divorced	Handlers-cleaners	Not-in-family	White	Male	40	United-States	0	<=50K
3	53	Private	234721	11th	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	40	United-States	0	<=50K
4	28	Private	338409	Bachelors	Married-civ-spouse	Prof-specialty	Wife	Black	Female	40	Cuba	0	<=50K

[Source: Kaggle, "Adult Census Income" dataset]



MITRE



BENERATOR
THE SMART WAY
TO HANDLE DATA

```
1 # Display basic statistics about the dataset
2 print("Data description - categoricals:")
3 actual_data.describe(include='object', datetime_is_numeric=True) |
```

Data description - categoricals:

	workclass	education	marital-status	occupation	relationship	race	sex	native-country	income
count	48842	48842	48842	48842	48842	48842	48842	48842	48842
unique	7	16	7	15	6	5	2	42	2
top	Private	HS-grad	Married-civ-spouse	Prof-specialty	Husband	White	Male	United-States	<=50K
freq	33906	15784	22379	6172	19716	41762	32650	43832	37155

native-country
48842
42
United-States
43832

United-States	43832
Mexico	951
?	857
Philippines	295
Germany	206
Puerto-Rico	184
Canada	182
El-Salvador	155
⋮	
Outlying-US(Guam-USVI-etc)	23
Yugoslavia	23
Scotland	21
Honduras	20
Hungary	19
Holand-Netherlands	1
Name: native-country, dtype: int64	

```
1 # Display basic statistics about the dataset
2 print("Data description - integers:")
3 actual_data.describe(datetime_is_numeric=True)
```

Data description - integers:

	age	fnlwgt	hours-per-week	capital
count	48842.000000	4.884200e+04	48842.000000	48842.000000
mean	38.643585	1.896641e+05	40.422382	991.565313
std	13.710510	1.056040e+05	12.391444	7475.549906
min	17.000000	1.228500e+04	1.000000	-4356.000000
25%	28.000000	1.175505e+05	40.000000	0.000000
50%	37.000000	1.781445e+05	40.000000	0.000000
75%	48.000000	2.376420e+05	45.000000	0.000000
max	90.000000	1.490400e+06	99.000000	99999.000000

```
from sdv import load_demo, SDV

# Use pre-loaded demo tables
metadata, tables = load_demo(metadata=True)

sdv = SDV()
sdv.fit(metadata, tables)

synthetic_data = sdv.sample()
print(synthetic_data)
```



Results out-of-the-box (statistical Copula model)

	age	workclass	fnlwgt	education	marital-status	occupation	relationship	race	sex	hours-per-week	native-country	capital	income
0	39	State-gov	77516	Bachelors	Never-married	Adm-clerical	Not-in-family	White	Male	40	United-States	2174	<=50K
1	50	Self-emp-not-inc	83311	Bachelors	Married-civ-spouse	Exec-managerial	Husband	White	Male	13	United-States	0	<=50K
2	38	Private	215646	HS-grad	Divorced	Handlers-cleaners	Not-in-family	White	Male	40	United-States	0	<=50K
3	53	Private	234721	11th	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	40	United-States	0	<=50K
4	28	Private	338409	Bachelors	Married-civ-spouse	Prof-specialty	Wife	Black	Female	40	Cuba	0	<=50K



Generated 48842 synthetic samples. Displaying the first few rows:

	age	workclass	fnlwgt	education	marital-status	occupation	relationship	race	sex	hours-per-week	native-country	capital	income
0	46	Private	129352	Some-college	Married-civ-spouse	Farming-fishing	Not-in-family	Black	Male	52	South	1775	<=50K
1	21	Private	466882	5th-6th	Never-married	Prof-specialty	Not-in-family	White	Male	43	United-States	7510	<=50K
2	52	Local-gov	129500	Some-college	Divorced	Prof-specialty	Husband	White	Male	59	United-States	41618	<=50K
3	37	Self-emp-inc	124908	Some-college	Married-civ-spouse	Tech-support	Not-in-family	White	Female	43	United-States	7586	<=50K
4	38	Federal-gov	149033	Some-college	Married-civ-spouse	Adm-clerical	Wife	White	Male	42	South	1889	<=50K

However ...

```
1 # Display basic statistics about the dataset
2 print("Data description - categoricals:")
3 actual_data.describe(include='object', datetime_is_numeric=True) |
```

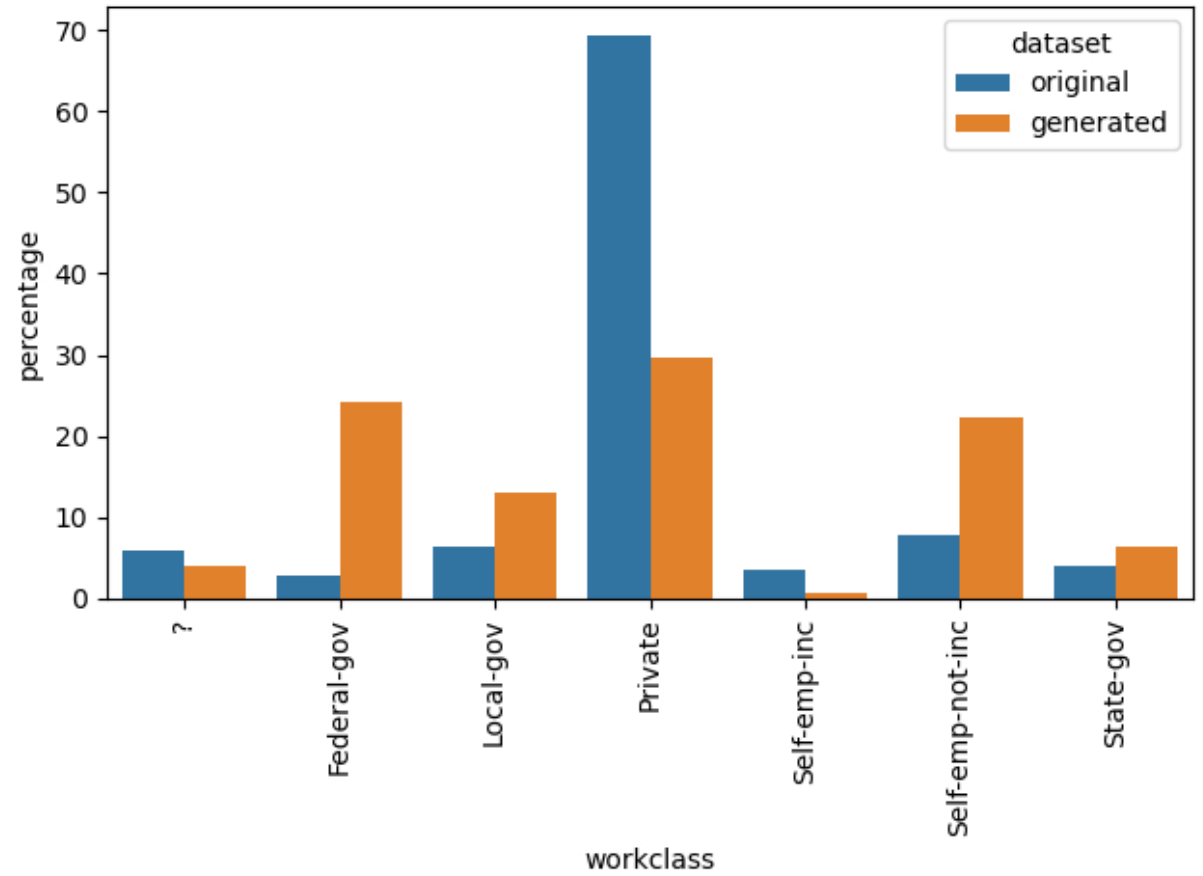
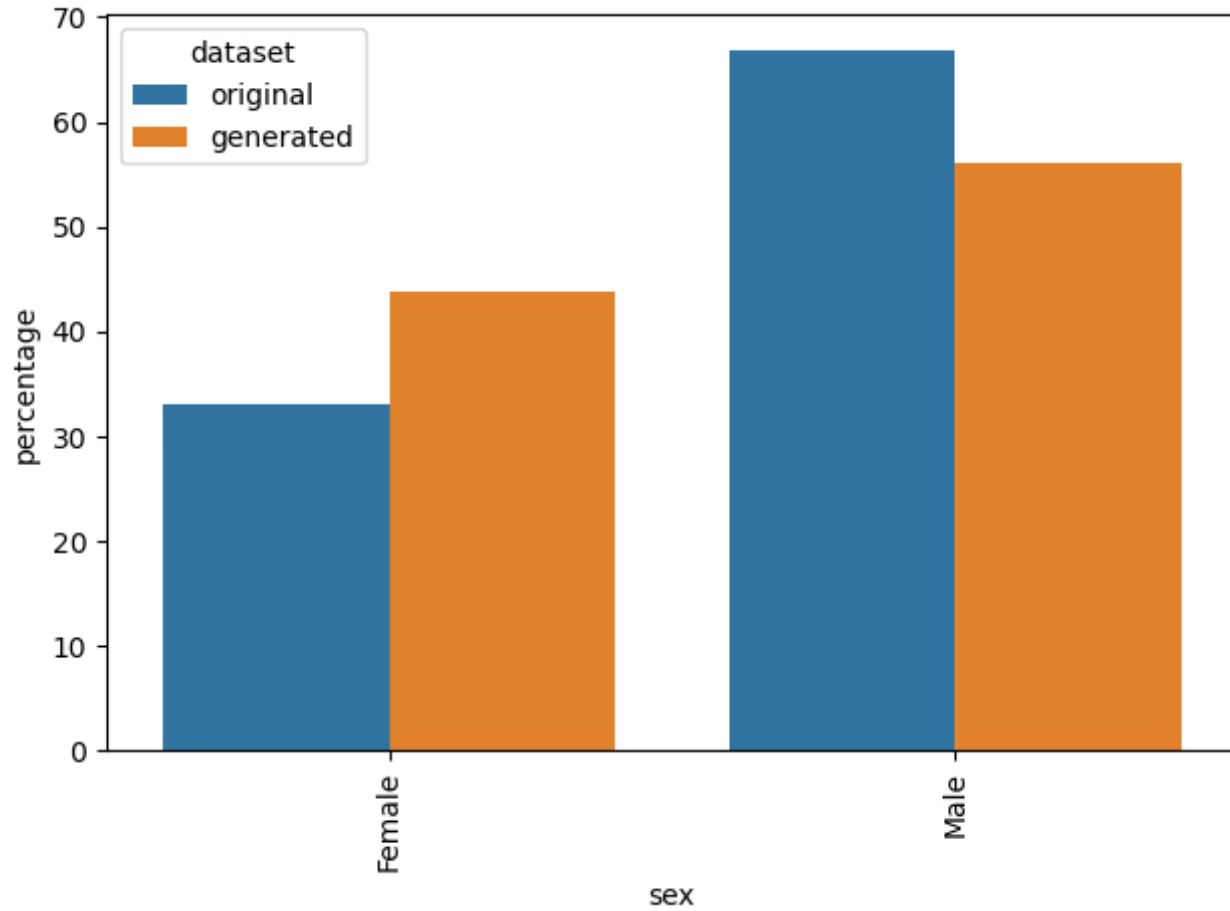
Data description - categoricals:

	workclass	education	marital-status	occupation	relationship	race	sex	native-country	income
count	48842	48842	48842	48842	48842	48842	48842	48842	48842
unique	7	16	7	15	6	5	2	42	2
top	Private	HS-grad	Married-civ-spouse	Prof-specialty	Husband	White	Male	United-States	<=50K
freq	33906	15784	22379	6172	19716	41762	32650	43832	37155

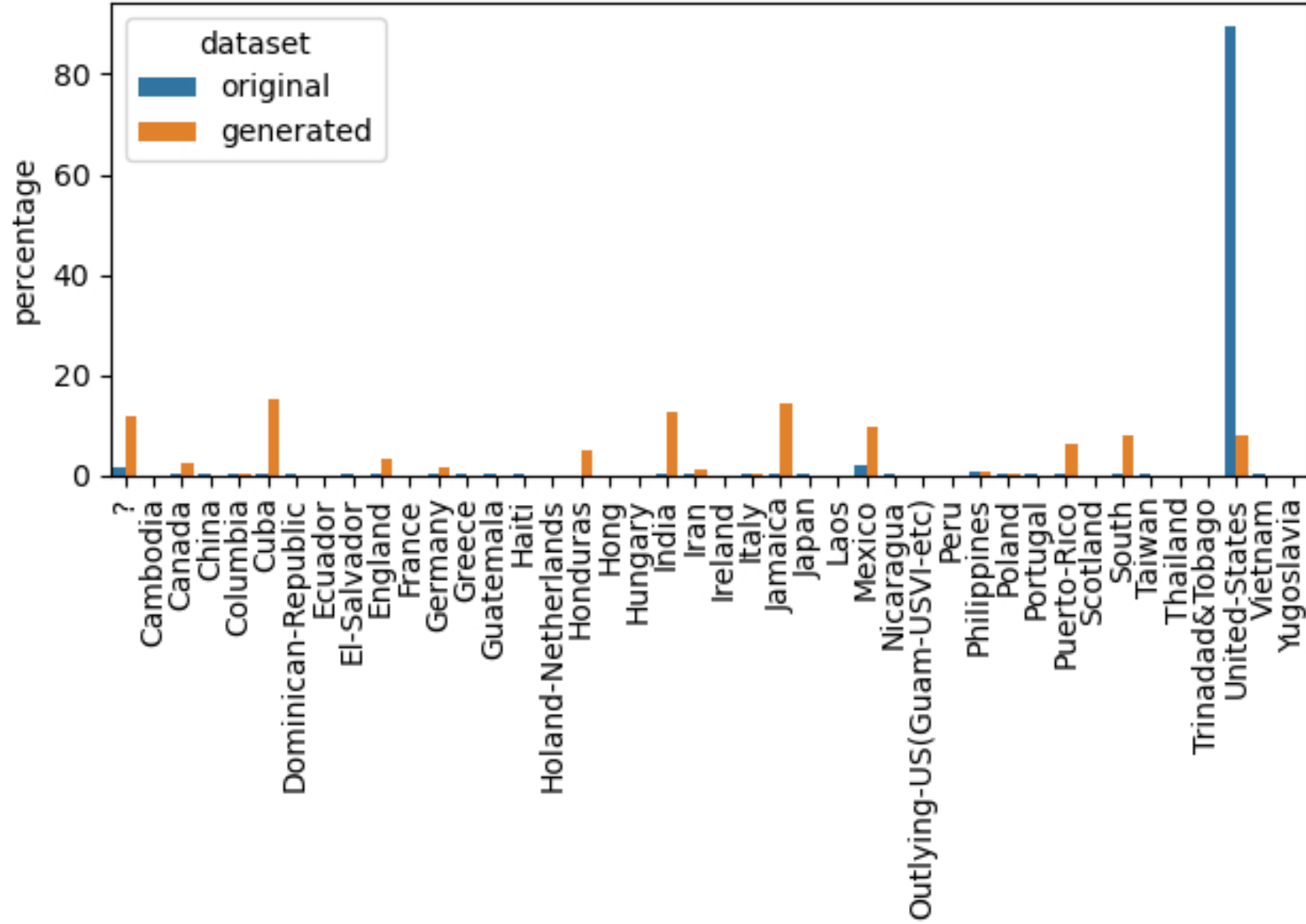


	workclass	education	marital-status	occupation	relationship	race	sex	native-country	income
count	48842	48842	48842	48842	48842	48842	48842	48842	48842
unique	7	16	7	15	6	4	2	24	2
top	Private	Some-college	Married-civ-spouse	Exec-managerial	Husband	White	Male	Cuba	<=50K
freq	14538	15604	19432	6293	15757	26779	27479	7253	31275

... or in graphical form ...



... or in graphical form ...

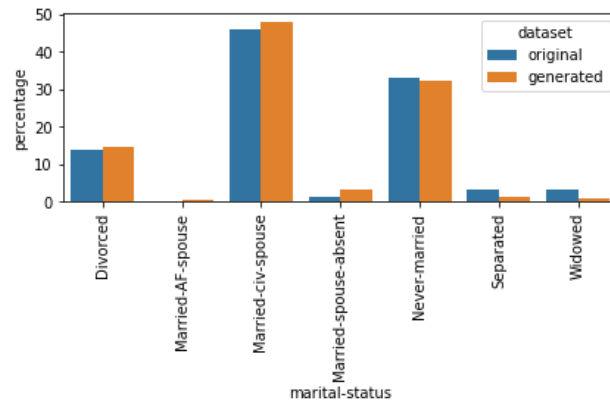
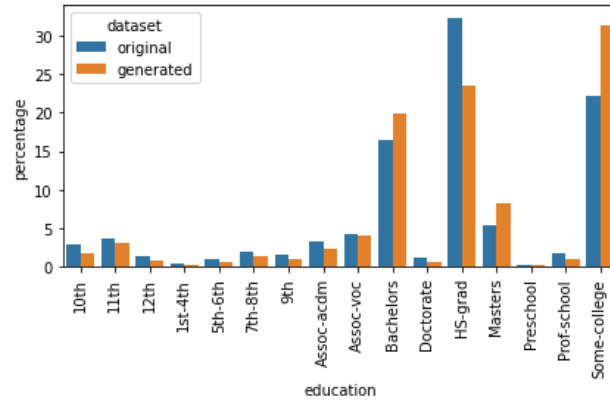
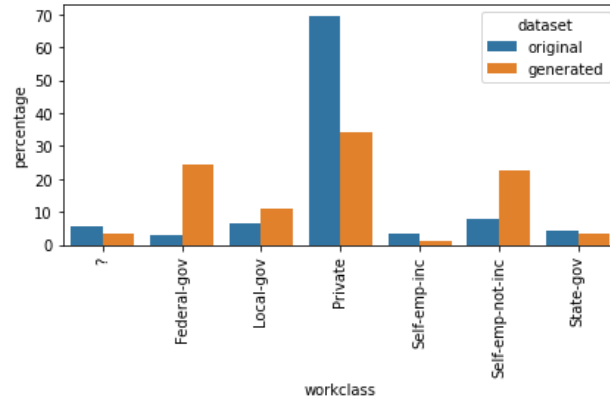


- SDV's default algorithms deal **particularly badly** with:
 - Highly **skewed** or **irregular** distributions
 - Distributions with **long tails**
 - **Outliers** (tend to be ignored)

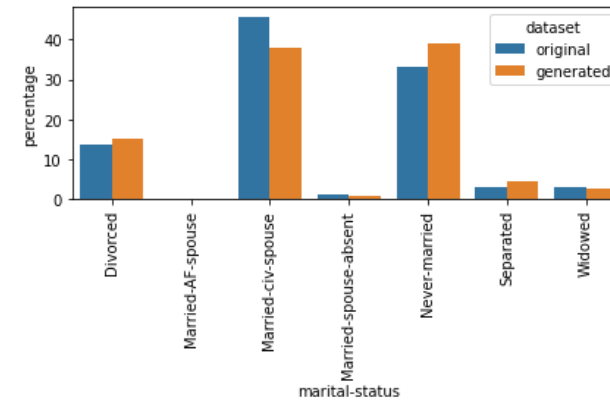
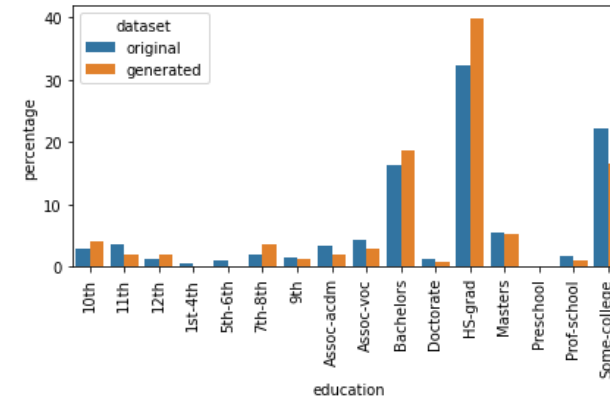
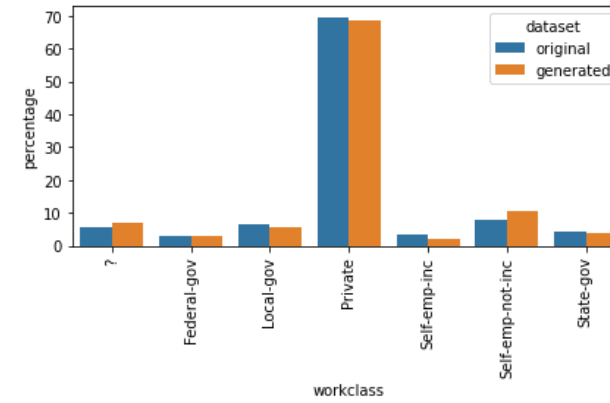
→ but this is all very common in real life datasets!
- There is a **structural limit**:
for rare values, there are not enough datapoints to learn suitable conditional distributions or correlations with other variables

Deep learning: sometimes better, sometimes worse

Copula (stat.)



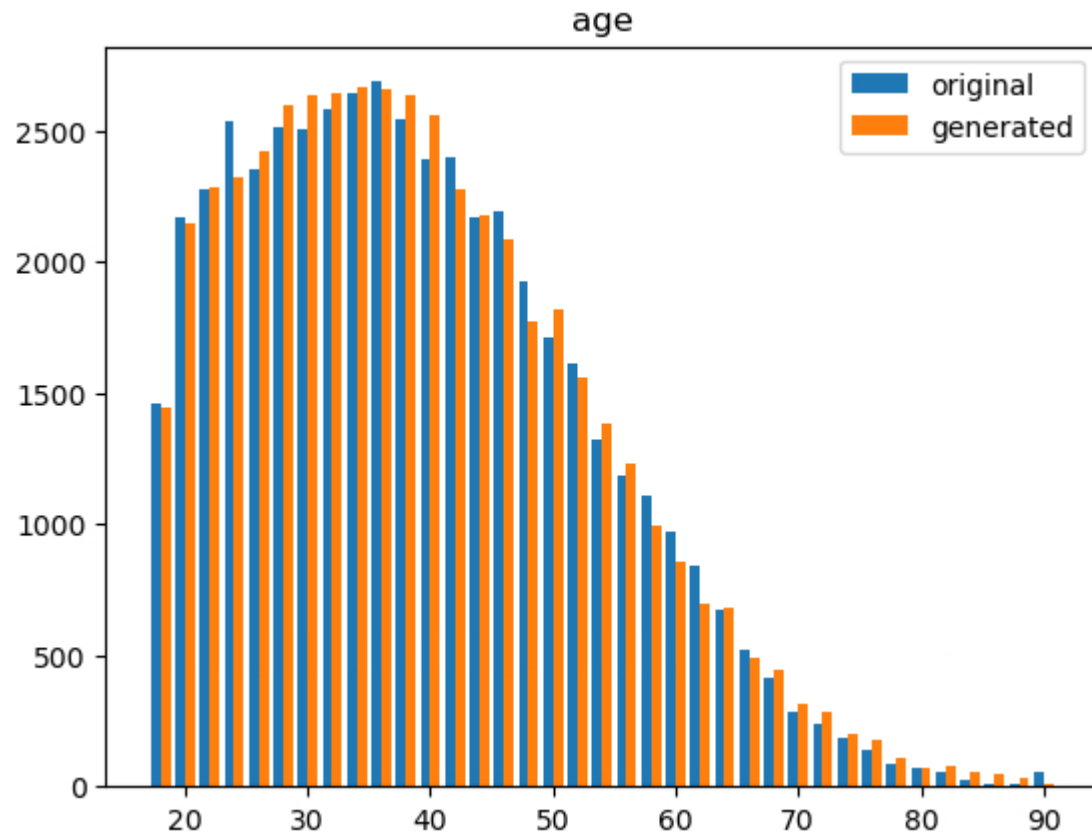
Copula+CTGAN (deep learning)



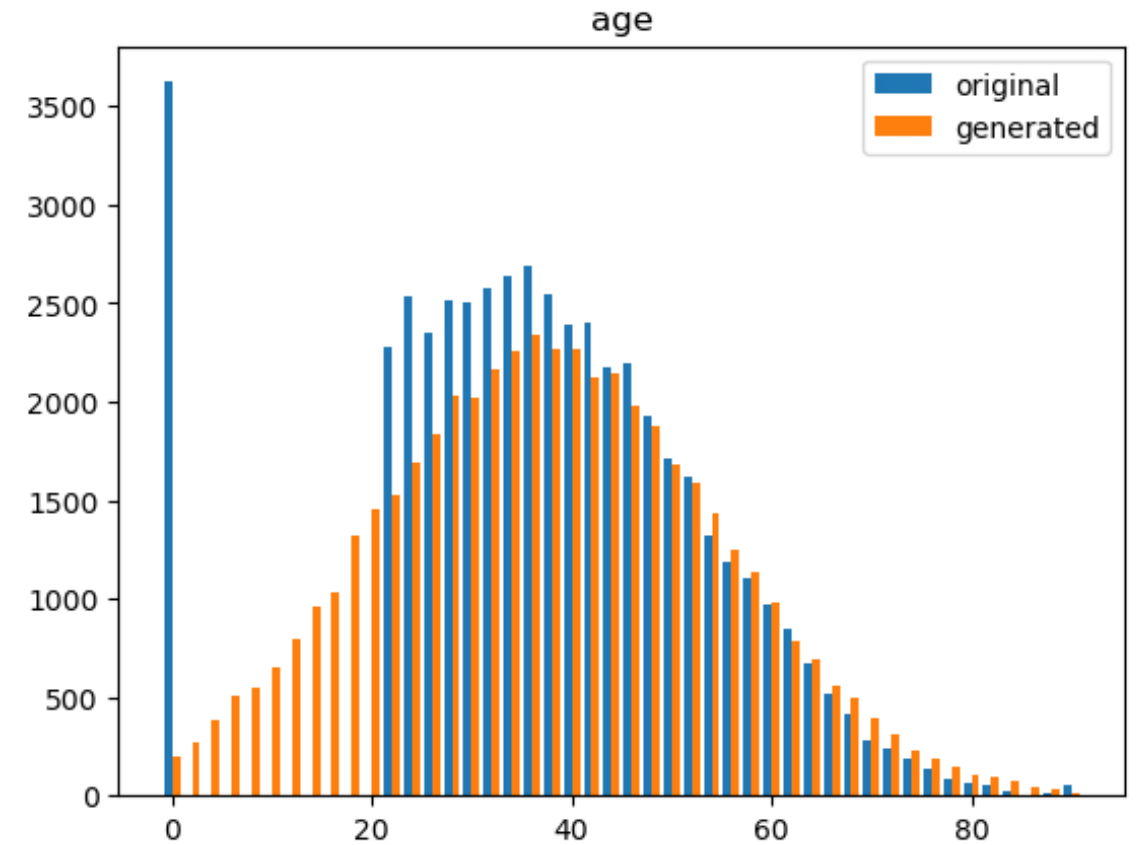


Know your data

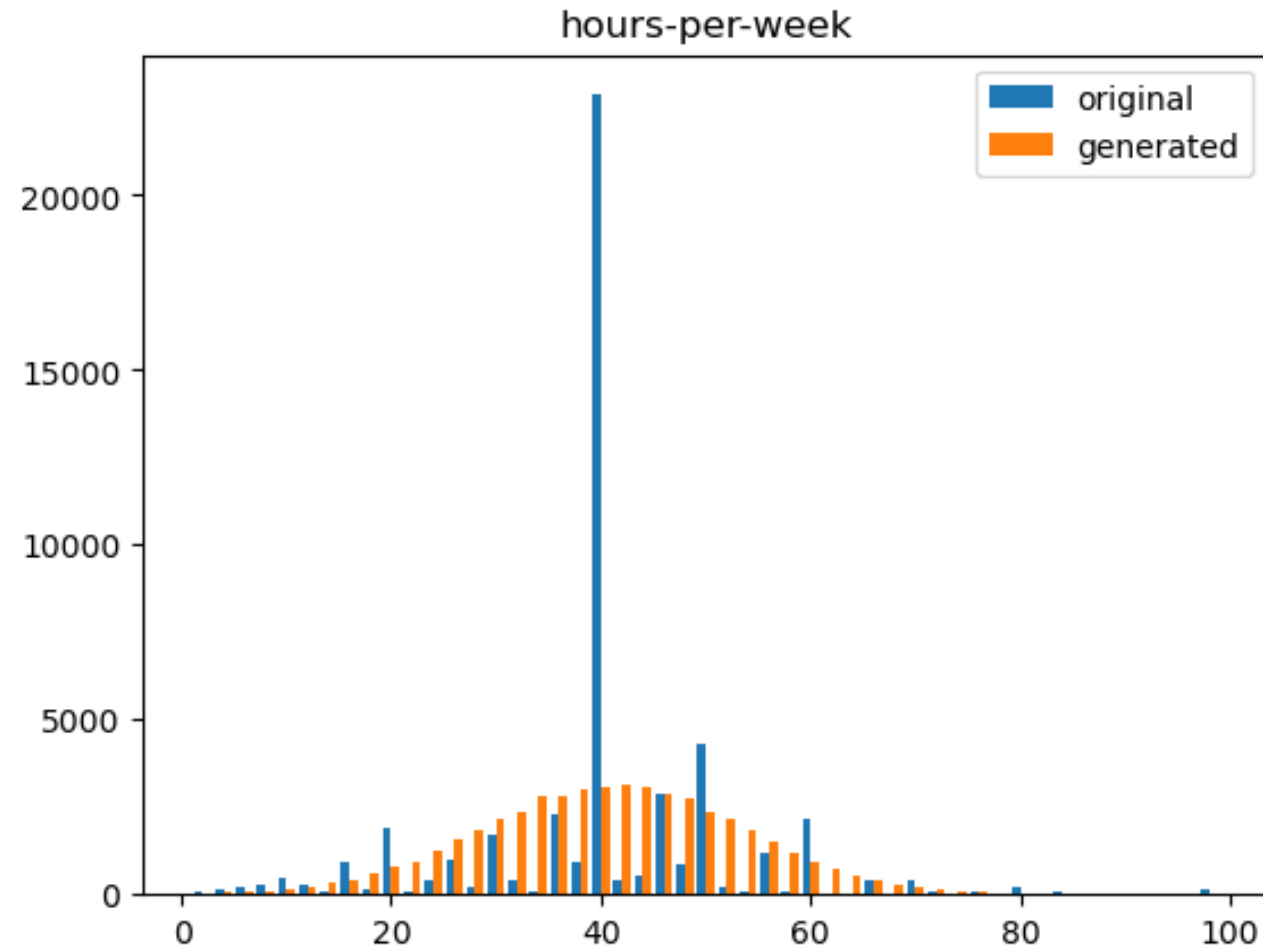
Minors omitted from the dataset:



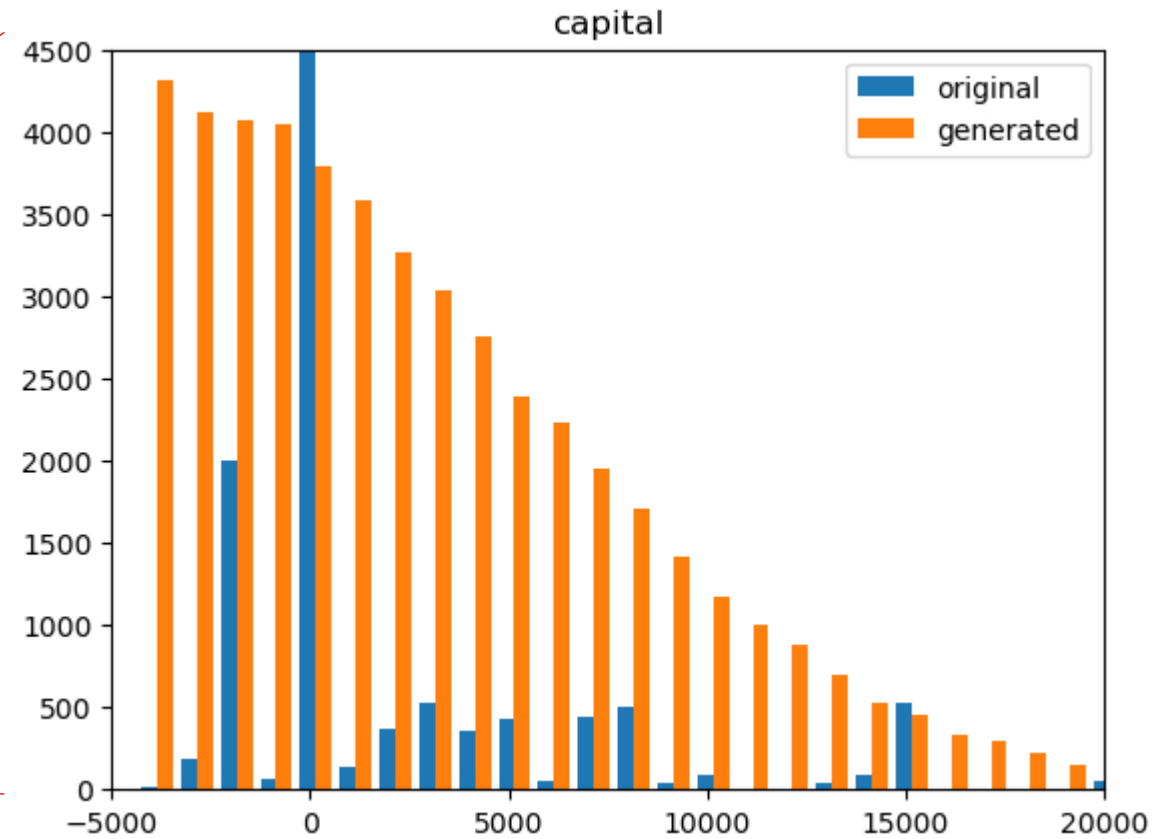
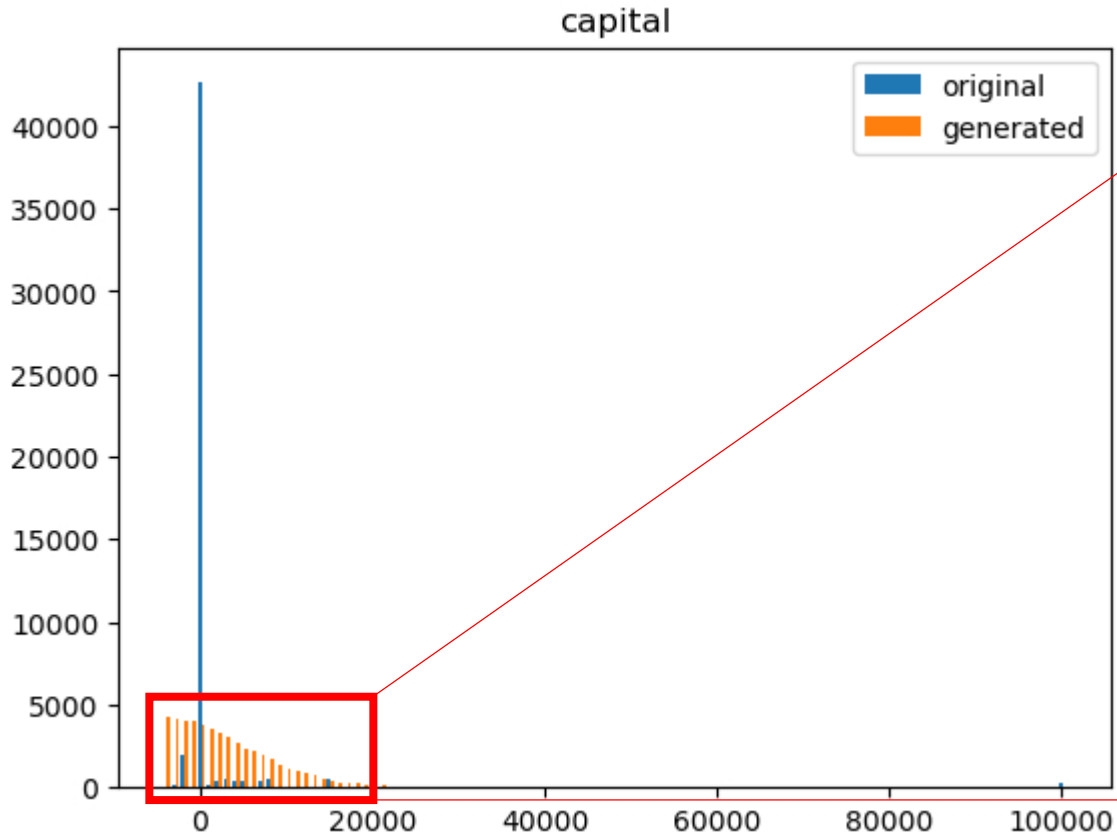
Minors encoded as "-1" in the dataset:



“hours worked per week” is encoded as integer but actually represents a categorical (fulltime, parttime, ...)



Data encoding matters (3)



For the vast majority of datapoints, capital=\$0,
and there is a limited number of other values
→ what does that mean?

```
1 actual_data['capital'].nunique()
```

```
221
```

```
1 new_data['capital'].nunique()
```

```
17428
```

- Assumes **smooth** distributions -> **interpolates** when sampling
- Computer doesn't know **meaning** of data → wrong data type estimates
- What does **omitted data** represent?
 - A missing year of birth doesn't mean a person was born in the year 0
 - Booleans: missing data = FALSE or missing data = third category?



Synthetic data construction
requires many **active decisions**
≠ Load data & press start



Data prep & finetuning

- Some datasets contain **count data**:


Frost	Rain	Sun	# days
No	No	Yes	52
No	Yes	No	43
Yes	No	Yes	1
No	No	No	187
No	Yes	Yes	10

(obviously this table does not say that frost is present in 20% of the datapoints)

- For a correct data model:
 - 1. “unroll” the data (= undo the counting, expand) and delete count variable
 - 2. train the model and generate synthetic data
 - 3. recount / regroup the synthetic data

- There are no guarantees that a particular value will be drawn from the distribution, especially when those values are rare / outliers:

	race	sex	native-country
count	48842	48842	48842
unique	5	2	42
top	White	Male	United-States
freq	41762	32650	43832



	race	sex	native-country
count	48842	48842	48842
unique	4	2	24
top	White	Male	Cuba
freq	26779	27479	7253

- Conditional generation** allows to forcibly generate certain values
- Conditioning on rare values may give repetitive results (because not enough data to properly learn conditional distributions)

- Columns can be entirely computed from others:

X	Y	X+Y	2Y-X
2	4	6	6
8	7	15	6
0	1	1	2
1	0	1	-1

- SDV cannot detect dependencies, only **approximately** learns correlations
- For a correct data model:
 - Remove computed columns
 - Learn model and generate data
 - Re-calculate and re-add the dependent columns

- The **meaning** of the data may imply other dependencies
 - Date of birth < date of death
 - City = Hasselt → Province = Limburg
 - Age < 18 → child_benefits = true
 - Distance > 0
 - \$ORCL = Oracle
 - A 25-year-old in year X, cannot be 36 in year X+1
- Encode these in **constraints** that can be
 - Incorporated in the model
 - Enforced by **fusing columns**
 - Enforced through **rejection sampling**

- **Minimize** the number of columns
 - Resynthesize a column only when necessary
 - → minimizes cumulative error
- **Exploit knowledge** about the data
 - Fuse columns that are strongly correlated (e.g. city and its province)
 - Use constraints to prevent generating nonsensical datapoints
 - Decide what to do with outliers and missing data and why
 - Merging the least-used categories into an “other” category (reduces the “long tail”)
- Watch out for **overfitting**
 - Explore a variety of training parameters



Using & Evaluating

- Possibilities for analytics on synthetic data are **limited!**
 - Structure of the data is **approximately** mimicked
 - **1** variable statistics (min, max, avg, etc) are **mostly** preserved,
 - Links between **2** variables (correlation, ...) are **somewhat** preserved,
 - Links between **more** variables (regressions, ...) are **poorly or not** preserved,
 - The **error margin** on synthetic data is **cumulative** and **increases**
 - More variables (columns)
 - More outliers or distribution imbalance
- Synthetic data **usefulness depends on the use case**

- *SDMetrics* library (under development) provides some toolkit-agnostic evaluation routines
- Commercial solutions often provide well-illustrated analysis reports
- e.g. cross-correlation graphs :

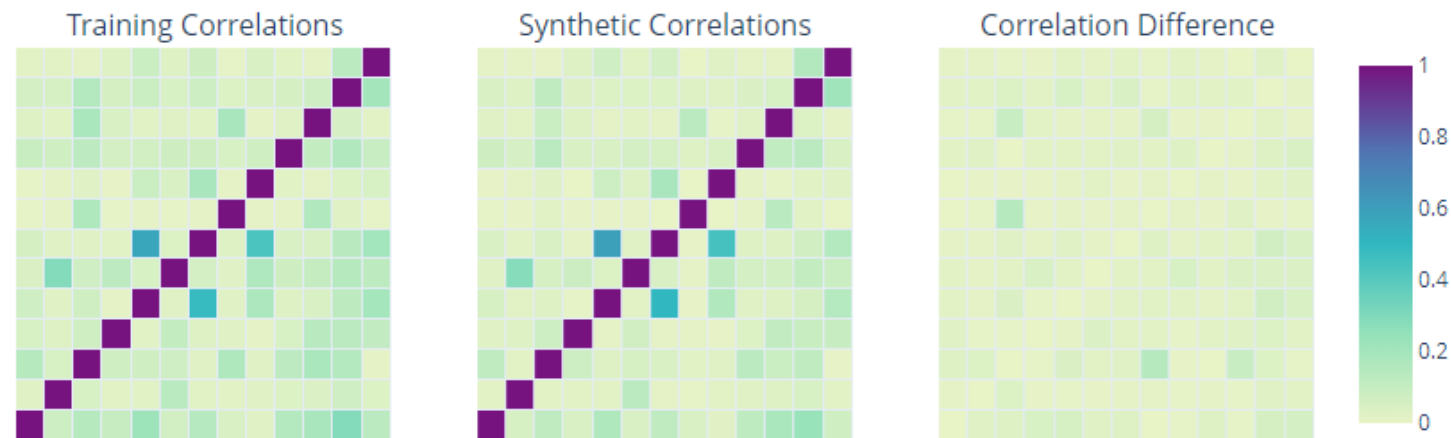


Image source: Gretel.AI

- Evaluation of privacy aspects:





The market

Commercial market is booming!



31M\$



45M\$



67M\$



BENERATOR

THE SMART WAY
TO HANDLE DATA

Configuration: example tonic.ai

The screenshot shows the Tonic AI web interface. The browser address bar displays `app.tonic.ai/bulk`. The main header includes the Tonic logo, the current workspace 'Privacy Scan Demo', and a 'Generate Data' button. A left sidebar contains navigation options: Privacy Hub, Database View (highlighted with a red arrow), Table View, Jobs, Autodetect, Schema Changes, Subsetting, and Collapse. The central panel shows a list of tables under the 'public' schema, including 'customers' (5 rows), 'customers_legacy' (6 rows), 'date', 'employees' (5 rows), 'marketing' (5 rows), 'products', 'retail_sales', 'stores' (1 row), 'vendors' (1 row), 'wholesale_orders', and 'wo_date'. Each table has a menu icon and an external link icon. The right panel, titled 'Every table', shows configuration filters for the selected table. It includes 'All Private Not Private' and 'All Protected Not Protected' filter buttons, a 'Column search' input field, and a list of columns with their respective data types and settings. A red arrow points to the 'Key' label for 'customers.Customer_Key'. Another red arrow points to the 'Passthrough' dropdown for 'customers.Marital_Status'. A third red arrow points to the 'Passthrough' dropdown for 'customers.Number_Of_Children'. A blue chat bubble icon is visible in the bottom right corner of the interface.

Workspace: Privacy Scan Demo

Generate Data

Tables

public

- customers 5
- customers_legacy 6
- date
- employees 5
- marketing 5
- products
- retail_sales
- stores 1
- vendors 1
- wholesale_orders
- wo_date

Every table

Filters

All Private Not Private

All Protected Not Protected

Column search

customers.Customer_Key Key

customers.First_Name Name X Type : First

customers.Last_Name Name X Type : Last

customers.Gender Categorical X

customers.Email Email X Domain : Random

customers.Marital_Status Passthrough

customers.Number_Of_Children Passthrough

∞ Runs

📄 Documentation

👤 User Settings

Tables

us-census-income ▾

Number of Rows	Number of Columns	
48,842	13	
Maximum Training Epochs	Batch Size	Learning Rate
1,000	64	0.001

State ● Training

- 1 Submitted**
The run has been submitted and is in the queue to be processed.
- 2 Provisioning**
Finished provisioning.
- 3 Encoding**
Finished encoding in 24 seconds.
[us-census-income] 13 of 13 columns finished ?
- 4 Training**
Training a generative model for 4 seconds.
- 5 Generating**
Once we have a satisfying model, we will generate the synthetic data.
- 6 Analyzing**
We will analyze the generated data and create a QA report from it.

MOSTLY·AI

🔗 Help

Reports: example Gretel.ai

Review

Results

Generated 5,000 records

```
1 17:24:32 Preparing privacy filters
2 17:24:35 Loaded 2 privacy filters
3 17:24:35 Starting privacy filtering
4 17:24:36 Privacy filtering removed 399 records, generating replacement records – filtered_outliers 17, filtered_si
5 17:25:54 Privacy filtering removed 38 records, generating replacement records – filtered_outliers 0, filtered_simi
6 17:26:04 Privacy filtering removed 1 records, generating replacement records – filtered_outliers 0, filtered_simil
7 17:26:06 Privacy filtering complete
8 17:26:06 Saving model archive
9 17:26:09 Creating synthetic quality report
10 17:26:21 Uploading artifacts to Gretel Cloud
11 17:26:22 Model creation complete!
```

Synthetic Quality Score



Privacy Protection Level



Data summary statistics

Field Correlation Stability



Deep Structure Stability

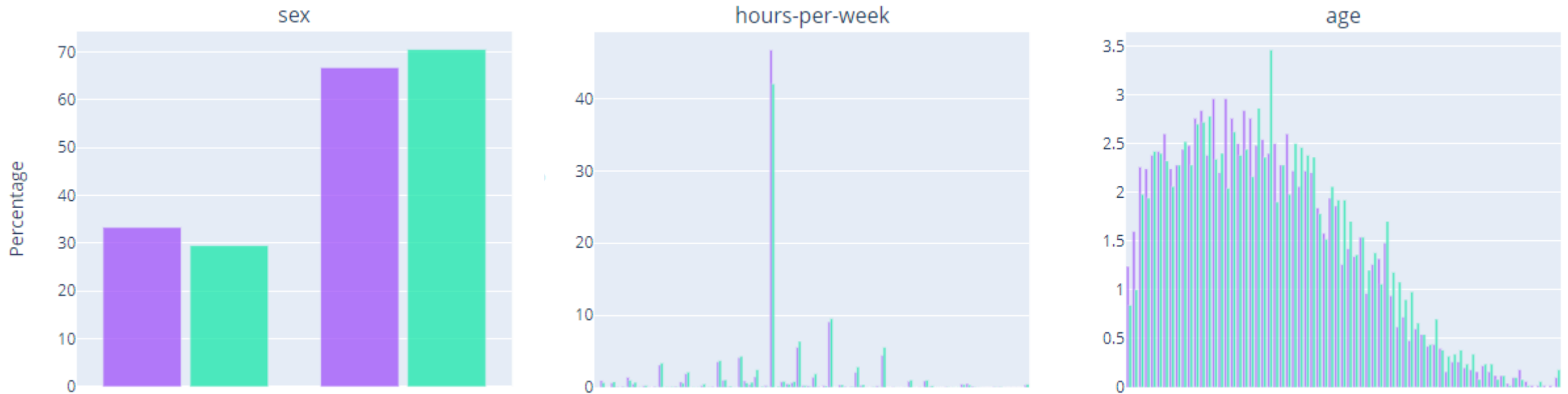


Field Distribution Stability



[Download Synthetic Report](#)

- Today, commercial software performs better than open source
 - Better estimation of data properties and subsequent setting of parameters
 - Seems more up to speed with developments in deep learning
- User-friendly interfaces
- Built-in reports with clean graphics



Don't believe me? Ask these guys!



JRC TECHNICAL REPORT

Multipurpose synthetic population for policy applications

Hradec, J., Craglia, M., Di Leo, M., De
Nigris, S., Ostlaender, N., Nicholson, N.

“Current methods of data synthesis using open source tools are relatively powerful but only for flat tables, with limited number of constraints, low cardinality categorical variables and continuous, without hard breaks.”

“Commercial solutions still beat the available research and open source solutions by a huge margin at the time of writing.”

“The field is evolving very fast and we may expect competitive open source solutions in the near future.”

[DOI 10.2760/50072 – July 2022]



Conclusions

The upsides

Carefully crafted and properly quality-checked synthetic data is **free from many regulatory issues**.

Synthetic data can be used **more flexibly**: put it in the cloud, make it available as Open Data, ...

Create **digital twins** or test environments with synthetic test data, to increase test coverage and improve the development process.

The field is fast evolving while also **steadily maturing**. Multiple vendors offer qualitative solutions today.

Current challenges

Inflated expectations: a synthetic dataset still differs from the original, and is therefore not for every use case a good substitute.

Synthetic data should not be taken at face value. **User discretion** is advised when interpreting results based on a synthetic dataset.

Qualitative synthesis **remains challenging** in some common cases:

- For hierarchical or very complex data
- For small datasets, datasets with many columns, or with many outliers

Creating good synthetic data **requires expert domain knowledge**, careful verification and validation, and a good grasp of statistics.

What's next?



[DALL-E 2 : “surrealist painting of a mirror standing in a barren landscape, reflecting a bustling city”]

Papers on diffusion models for tabular text are starting to appear:

TabDDPM: Modelling Tabular Data with Diffusion Models

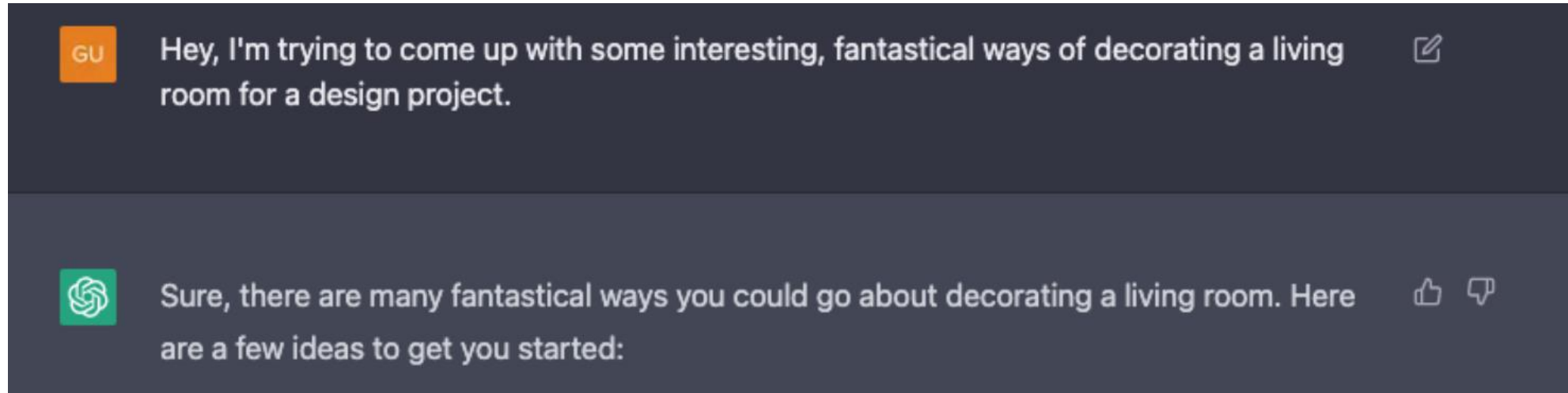
30 Sep 2022 · Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, Artem Babenko · [Edit social preview](#)

[Source: paperswithcode.com]

What's next?


- Continued developments in all fields of “Generative AI”
- Combinations of different technologies and cross-pollination

e.g. ChatGPT (dialog model) + MidJourney (text-to-image generator)



[Source: Twitter / Guy Parsons (@GuyP), 30/11/2022]

What's next?


GU Hey, I'm trying to come up with some interesting, fantastical ways of decorating a living room for a design project. 

1. Create a fairy tale-inspired living room by using pastel colors and adding elements like a magical castle mural on one wall, a chandelier made of branches and twinkling lights, and comfortable furniture with curved, whimsical shapes.



[Source: Twitter / Guy Parsons (@GuyP), 30/11/2022]

What's next?


GU Hey, I'm trying to come up with some interesting, fantastical ways of decorating a living room for a design project. 

2. Transform the living room into an enchanted forest by hanging sheer, green fabric from the ceiling to create a canopy effect, adding potted plants and trees, and using earthy tones and natural materials for the furniture and decor.



[Source: Twitter / Guy Parsons (@GuyP), 30/11/2022]

What's next?

GU Hey, I'm trying to come up with some interesting, fantastical ways of decorating a living room for a design project. 

3. Make the living room into a futuristic space by using sleek, modern furniture and lighting, and adding metallic accents and pops of vibrant color. You could also incorporate unique, space-age elements like a holographic fireplace or a ceiling mural



[Source: Twitter / Guy Parsons (@GuyP), 30/11/2022]

- Related articles on www.smalsresearch.be (NL/FR)
 - Synthetic Data [[link](#)]
 - Data scrambling: synthetische data in de praktijk [[link](#)]
 - Privacybevorderende technologieën voor de publieke sector [K. Verslype, [link](#)]
 - De AI als auteur: een blik op Natural Language Generation [[link](#)]
 - L'intelligence artificielle dans le secteur public [K. Fokou, [link](#)]
- Academic
 - Rodriguez et.al: "In defense of synthetic data" [[link](#)]
 - Gupta et.al.: "Transitioning from Real to Synthetic Data – Quantifying the Bias ..." [[link](#)]
 - Chaudhary et.al.: "FairGen - Fair Synthetic Data Generation" [[link](#)]
 - Arnold et.al.: "Really Useful Synthetic Data – A Framework ..." [[link](#)]
- Other
 - JRC report: "Multipurpose synthetic populations for policy applications" [[link](#)]
 - SyntheticMass [[link](#)]
 - Curated list of various other resources [[link](#)]



Thank you!

Questions?

Joachim Ganseman

joachim.ganseman@smals.be

www.smalsresearch.be

Smals, ICT for society

02 787 57 11

Fonsnylaan 20 / Avenue Fonsny 20

1060 Brussel / 1060 Bruxelles