https://stochasticsolutions.com/pdf/toronto2021-reproducibility-in-safe-haven-slides.pdf

REPRODUCIBILITY IN A SAFE HAVEN

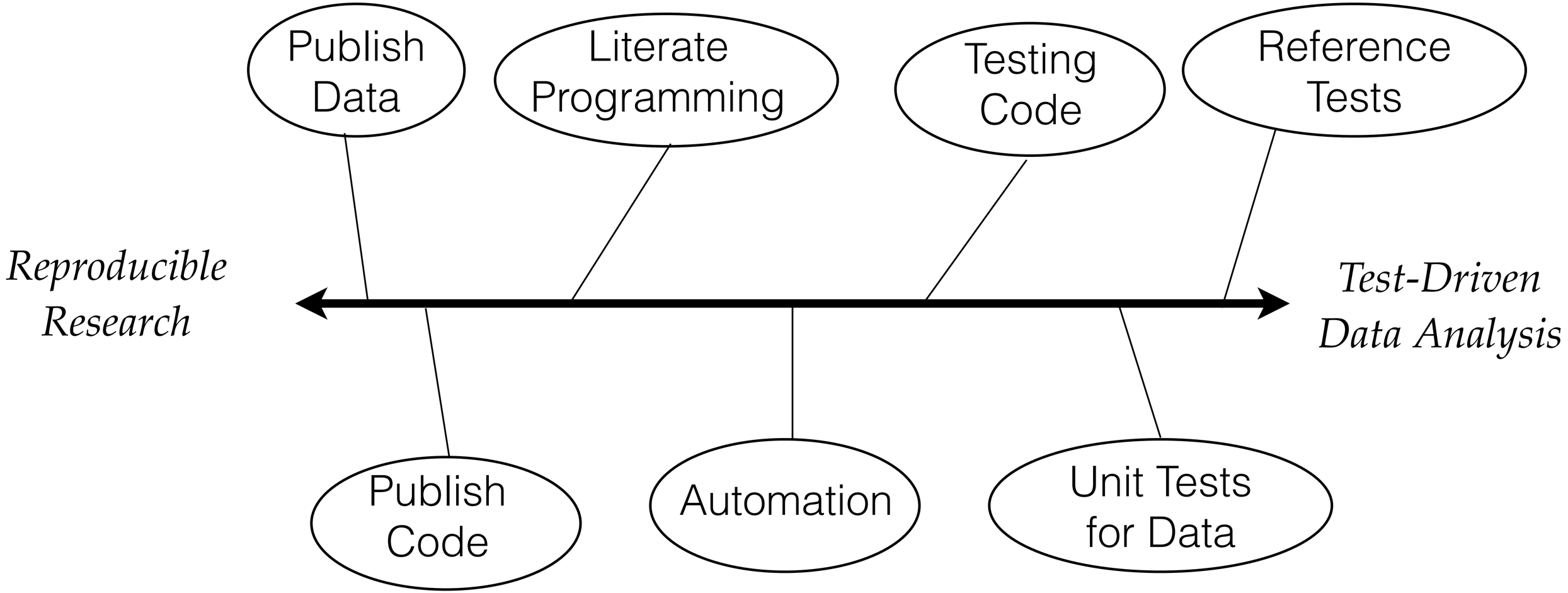
GLOBAL OPEN FINANCE
CENTRE OF EXCELLENCE

Stochastic Solutions

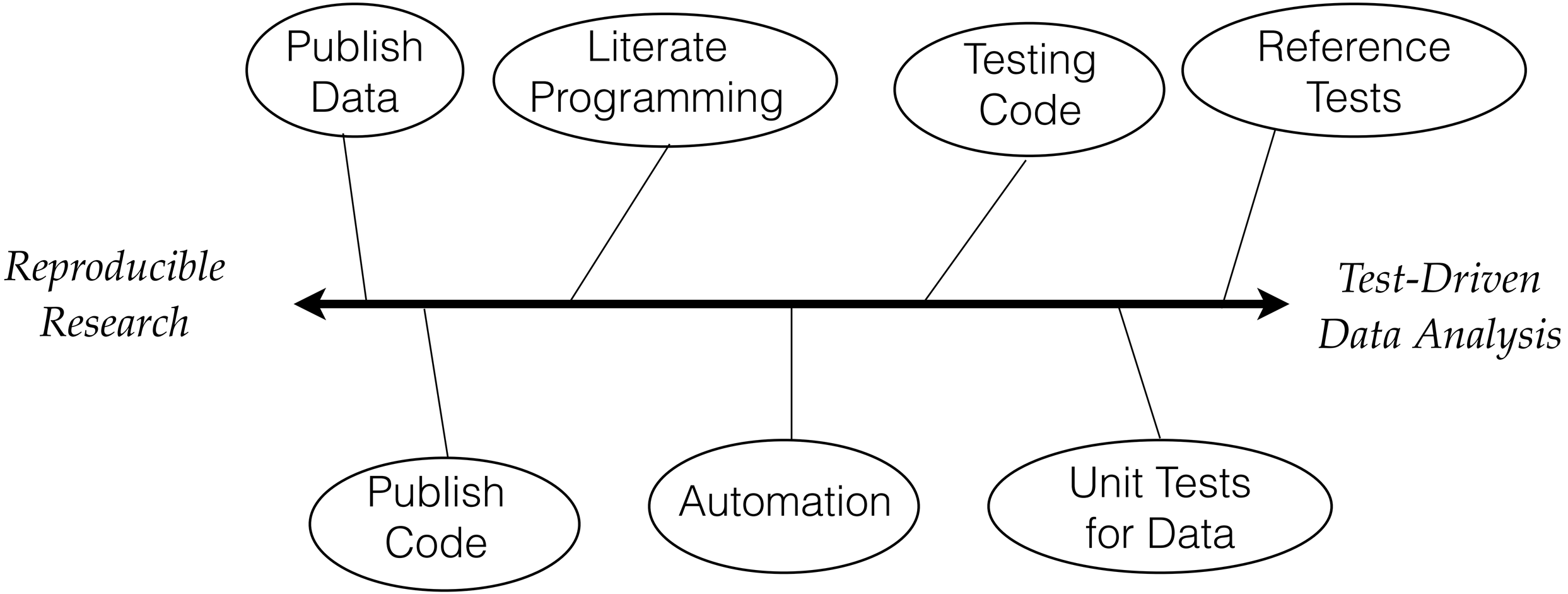
Nicholas J. Radcliffe, Ph.D.
The Global Open Finance Centre of Excellence
& Department of Mathematics, University of Edinburgh
& Stochastic Solutions Limited

Pei Shan Yu, Ph.D.
The Global Open Finance Centre of Excellence
University of Edinburgh

REPRODUCIBILITY



REPRODUCIBILITY



R Python

REPRODUCIBLE
RESEARCH

actually
REPRODUCED
RESULTS

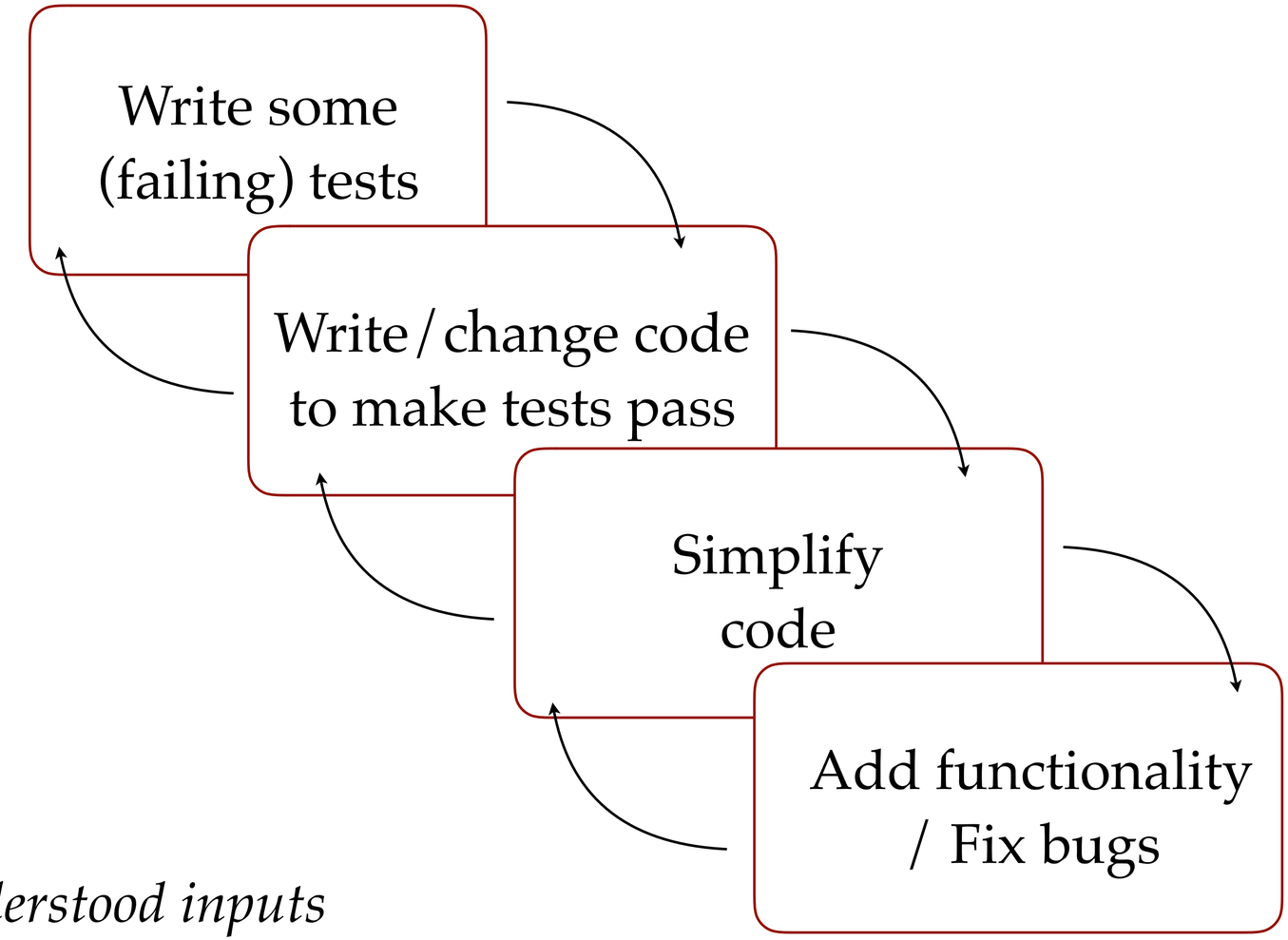
(If a tree falls in a forest, and no one is around to hear it, does it make a sound?)

LINUS'S LAW

*"Given enough eyeballs, all bugs are shallow"**

* **Just a claim; not actually true**

TEST-DRIVEN DEVELOPMENT (TDD)



Often:

- *Well-understood inputs*
- *Well-understood goal*
- *Many kinds of errors/failures are unmistakable*

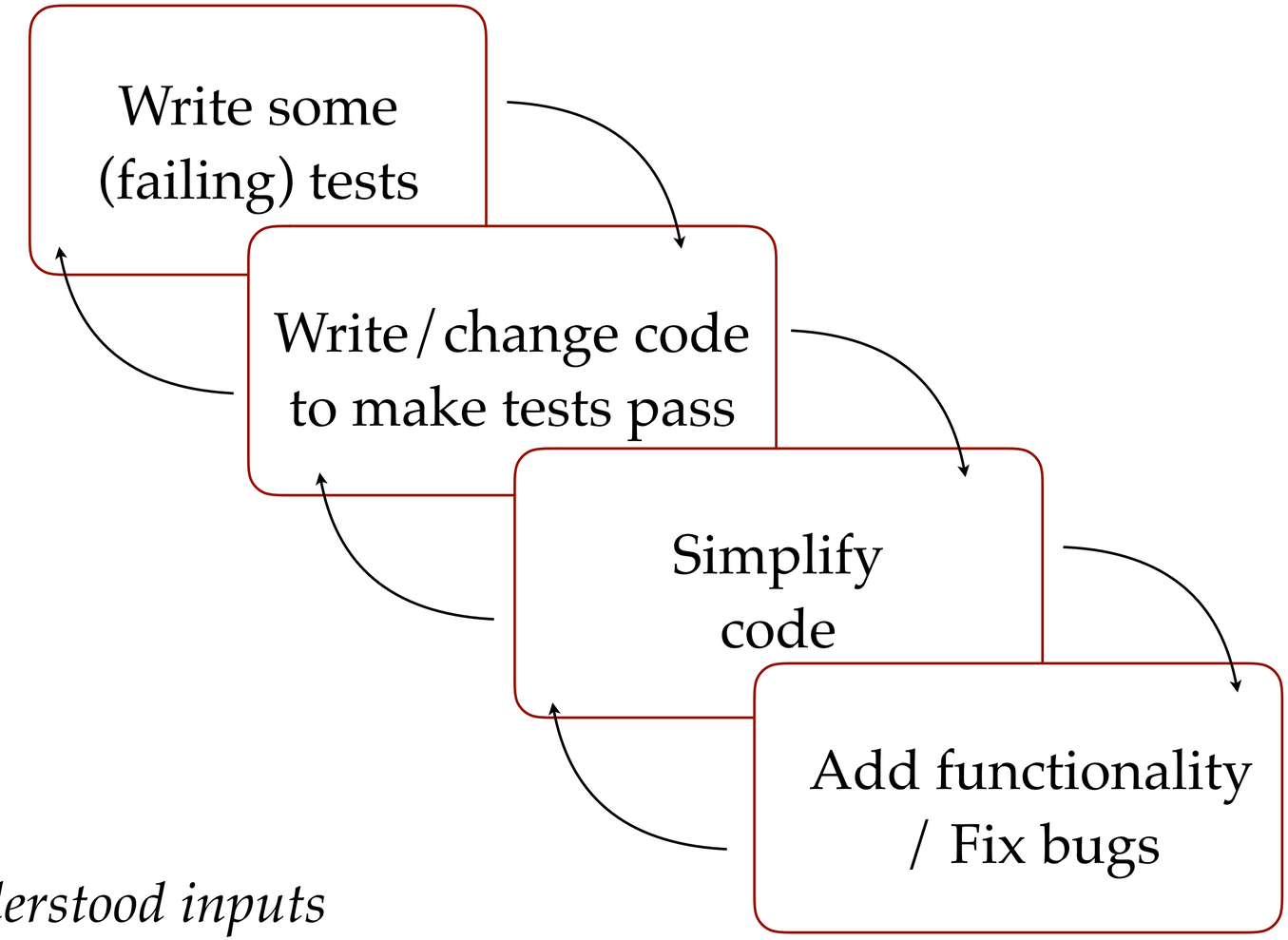
TESTS (ADDITION)

$$2 + 2 = 4$$

$$-1 + 1 = 0$$

$$10^{100} + 10^{100} = 2 \times 10^{100}$$

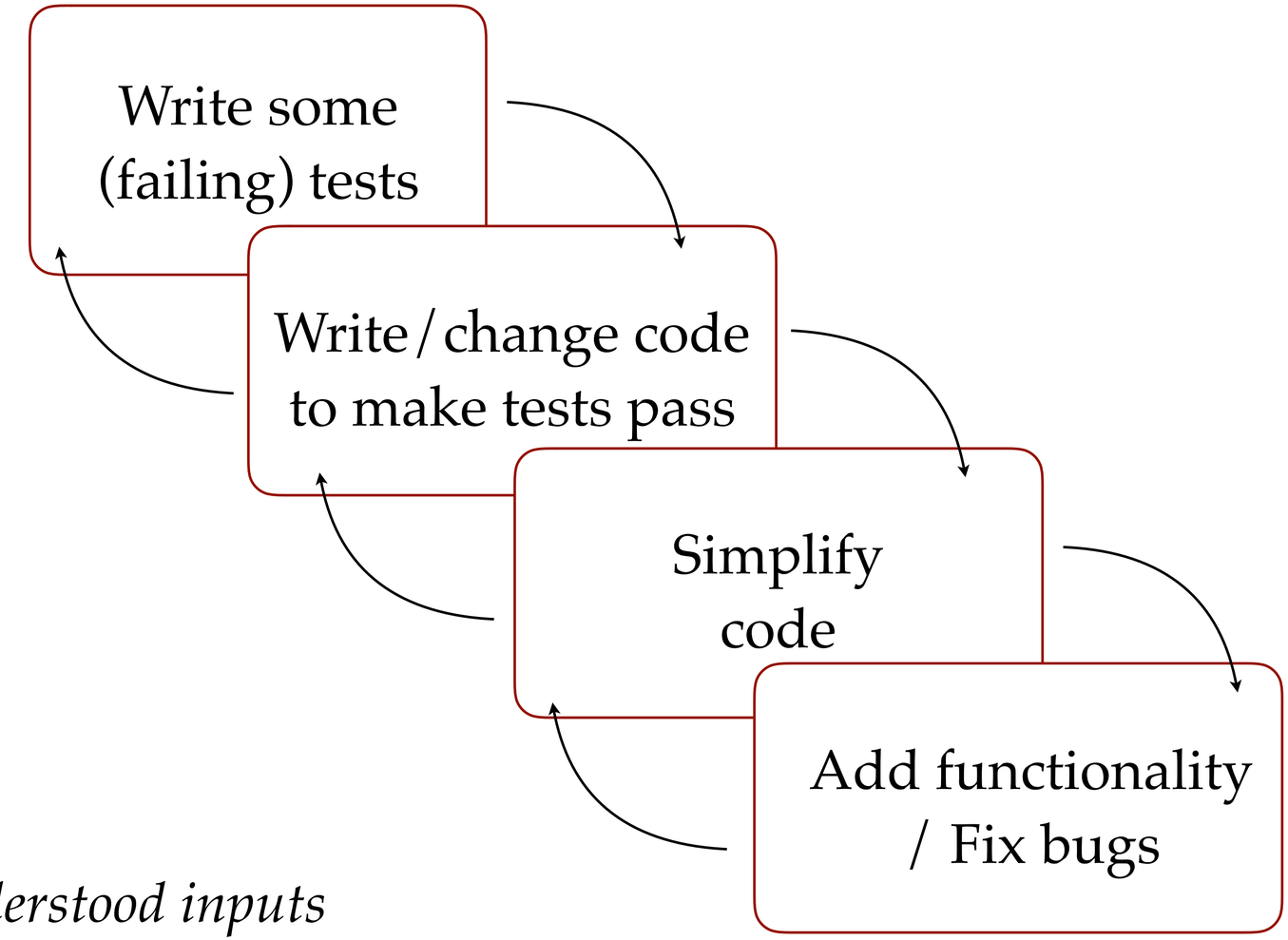
TEST-DRIVEN DEVELOPMENT (TDD)



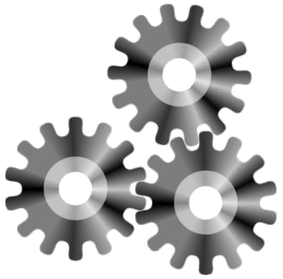
Often:

- *Well-understood inputs*
- *Well-understood goal*
- *Many kinds of errors/failures are unmistakable*

TEST-DRIVEN DEVELOPMENT (TDD)



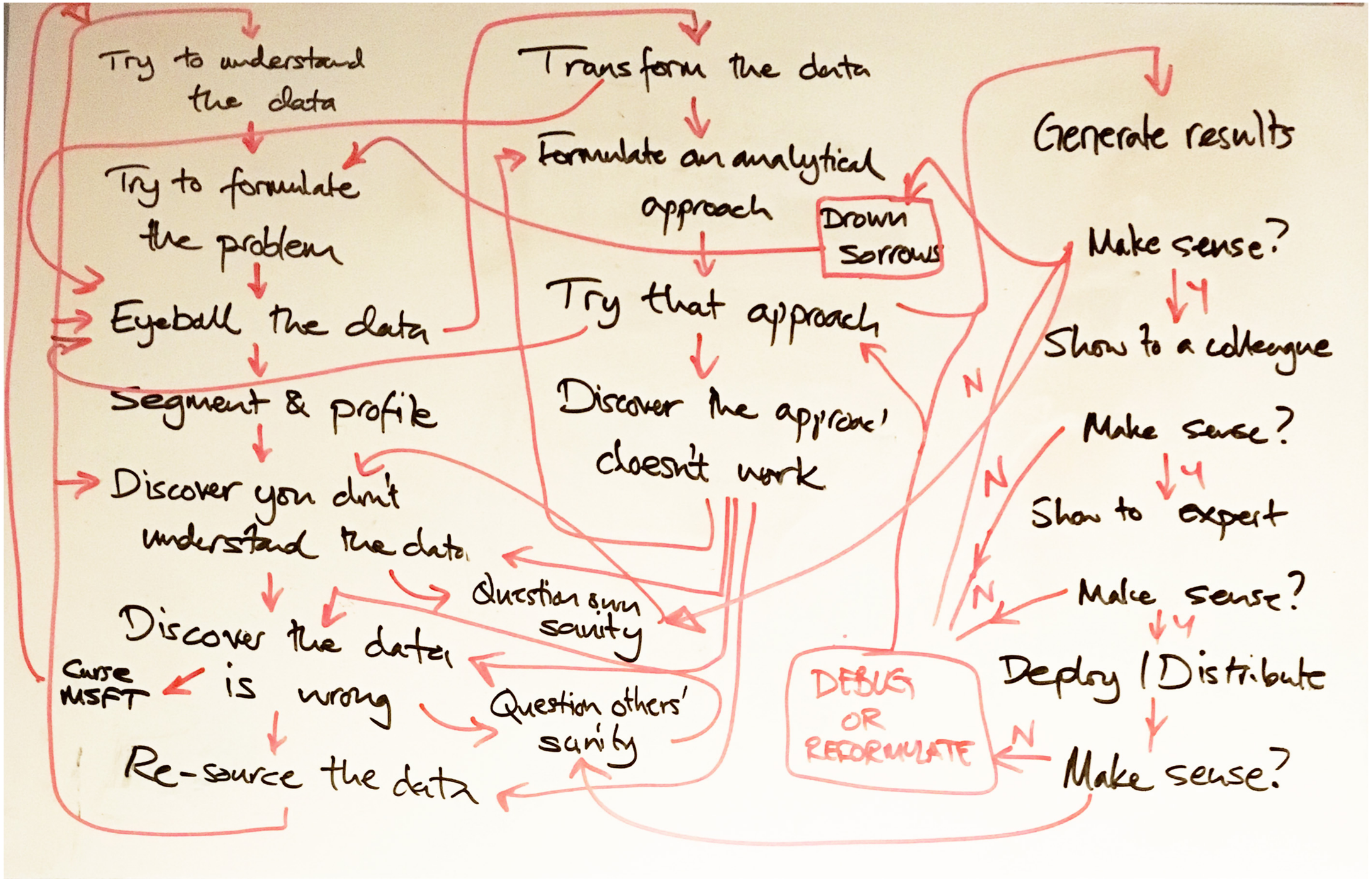
Constantly run tests with CI?



CONTINUOUS INTEGRATION SYSTEM

Often:

- *Well-understood inputs*
- *Well-understood goal*
- *Many kinds of errors/failures are unmistakable*


Try to understand the data
Try to formulate the problem
Eyeball the data
Segment & profile
Discover you don't understand the data
Discover the data is wrong
Curse MSFT
Re-source the data
Transform the data
Formulate an analytical approach
Drown Sorrows
Try that approach
Discover the approach doesn't work
Question own sanity
Question others' sanity
DEBUG OR REFORMULATE
Generate results
Make sense?
Y
Show to a colleague
Make sense?
Y
Show to expert
Make sense?
Y
Deploy / Distribute
Make sense?
N
N
N
N

“Shouldn’t we do test-driven data analysis?”

— Patrick Surry, c. 2010

TDD ↦ TDDA

We need to extend TDD's idea of testing for

software correctness

with the idea of testing for

meaningfulness of analysis,

correctness and validity of input and output data,

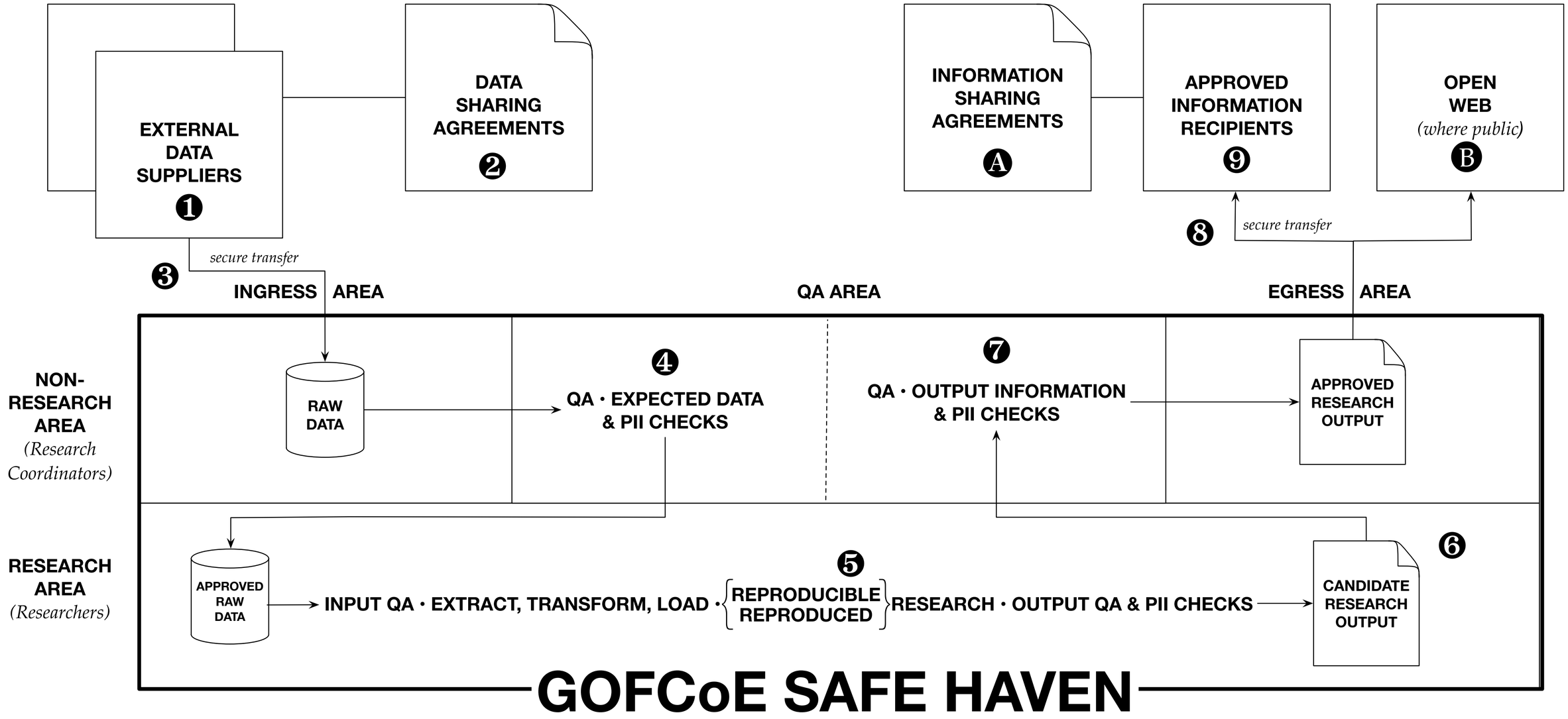
& *correctness of interpretation*.

"test-driven data analysis"

GLOBAL OPEN FINANCE

CENTRE OF EXCELLENCE

- Non-profit established as global economic observatory to study the way people and businesses earn, spend and save with a view to improving financial outcomes, equity and sustainability
- Incubated in the Edinburgh Futures Institute at University of Edinburgh
- Currently looking at the financial impact of COVID-19 on citizens and businesses in UK
- Outputs to national, regional and local governments to guide policy development and assessment



4-layer access controls for approved researchers only
No external internet access from/to the Safe Haven
No transfers (file, copy & paste etc.) to/from the user's computer to the Safe Haven
Data ingress and egress only though Research Coordinators (information governance)
via secure, encrypted, managed file transfer processes

To date, every output shared with government has been reproduced by two separate software systems / teams in the Safe Haven

CORE ASPECTS

1. *Data ingestion and verification*
    - *Checks for personal/identifying data and data matches specs*
2. *Replicating outputs and testing analytical code and processes*
3. *Output disclosure control*
    - *Vetting of outputs, audit trails, data provenance & lineage*

REPRODUCIBILITY & TDDA

1. *Data ingestion and verification*
   - ***Checks for personal data*** *and* ***data matching specs***
2. ***Replicating outputs*** *and* ***testing analytical code and processes***
3. *Output disclosure control*
   - ***Vetting of outputs****, audit trails, data provenance & lineage*

KEY

Reproducible Research

TDDA

CONSTRAINT GENERATION, DATA VERIFICATION & ANOMALY DETECTION

TDDA LIBRARY

Install from PyPI (recommended)

```
pip install tdda
```

or from Github (source)

```
git clone https://github.com/tdda/tdda.git
python setup.py install
```

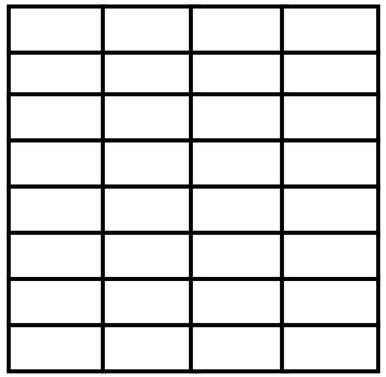
TDDA LIBRARY

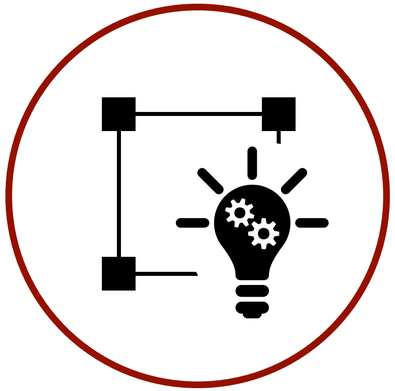
- Runs on Python 2* & Python 3, Mac, Linux & Windows, under `unittest` and `pytest`
- MIT Licensed
- Documentation:
  - Sphinx source in `doc` subdirectory
  - Built copy at `http://tdda.readthedocs.io`
- Quick reference:

  `http://www.tdda.info/pdf/tdda-quickref.pdf`

* But no one should still be using Python 2!

AUTOMATIC CONSTRAINT GENERATION





```
{
 C1: Age ≥ 0
 C2: ID is not null
 C3: CardNumber ~
      DDDD DDDD DDDD DDDD
              ⋮
}
```

TRAINING DATA

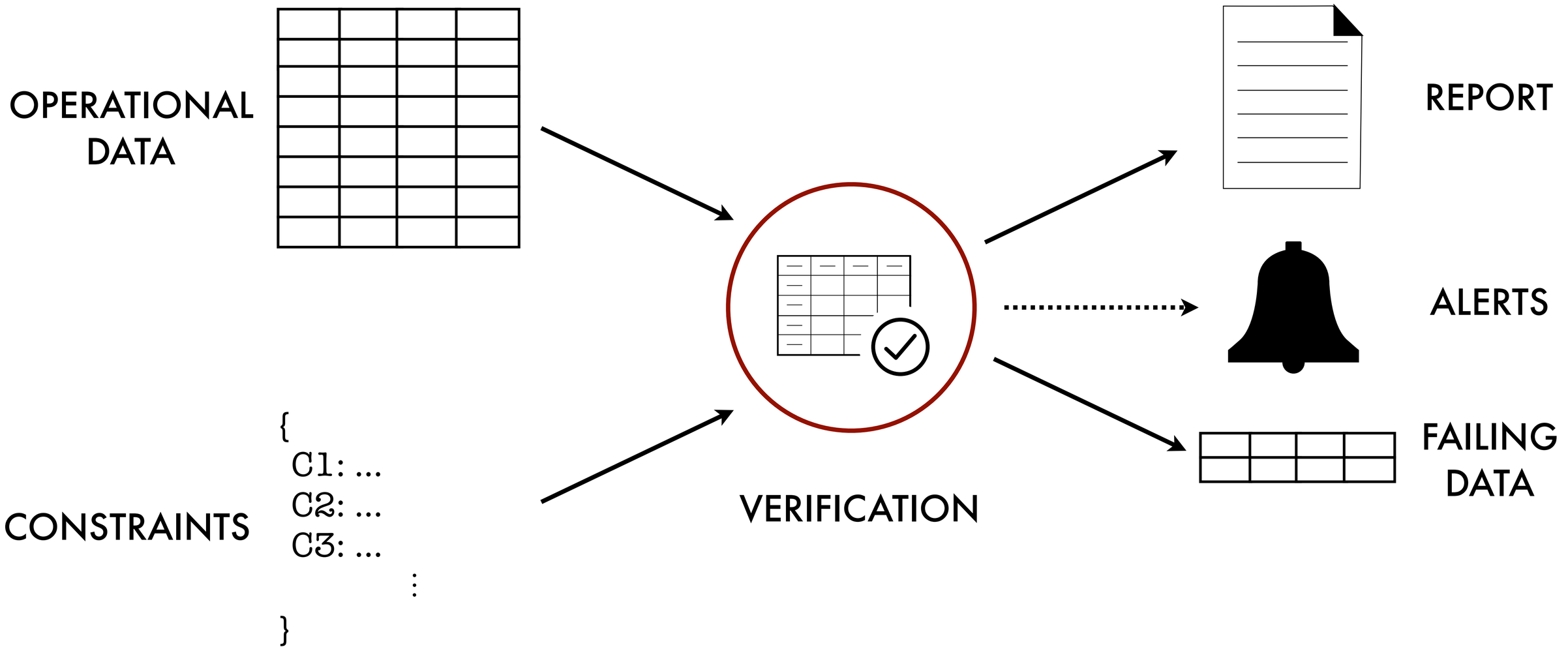
(believed to be "good")

AUTOMATIC DISCOVERY OF CONSTRAINTS

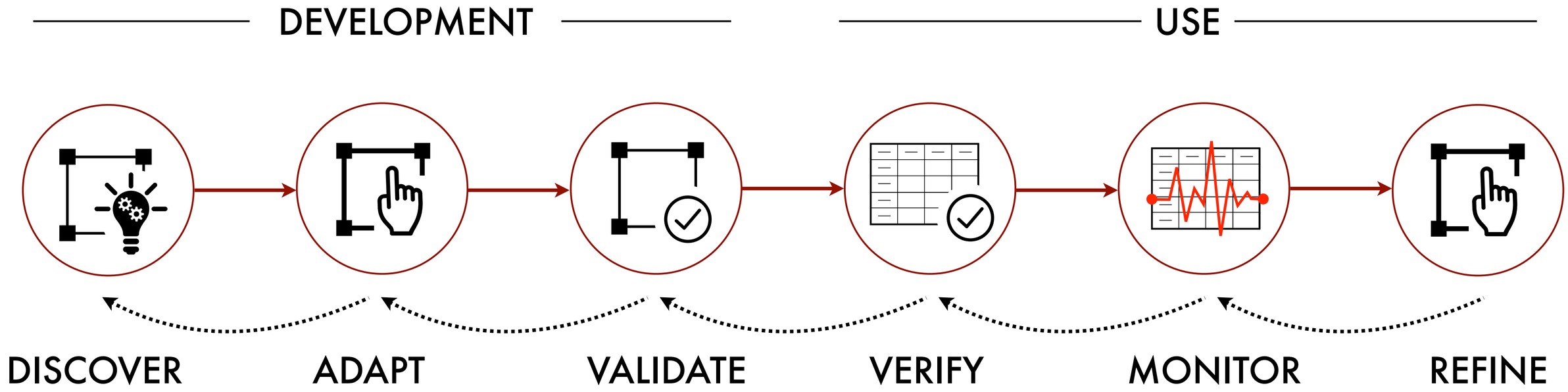
DISCOVERED CONSTRAINTS

```
{
  "creation_metadata": {
    "as_at": "2021-01-20 13:51:08",
    "local_time": "2021-01-20 13:51:08",
    "utc_time": "2021-01-20 13:51:08",
    "creator": "Miro 3.1.18",
    "creation_command": "discover -o /tmp/tbank",
    “creation_session_log":
        “~/miro/log/2021/01/20/session073.html",
    "source": "banksc-raw",
    "host": "gofcoe01",
    "user": "njr",
    "dataset": banksc-raw",
    "n_records": 205723986,
    "n_selected": 205723986,
    "tddafile": "~/citizendata/qa/banksc.tdda",
    "last_edit": "Nick Radcliffe",
    "version": "1.1",
    "last_edit_date": "2021-02-25 16:00:00"
  },
  "fields": {
    "pseudoid": {
      "type": "string",
      "min_length": 40,
      "max_length": 40,
      "max_nulls": 0,
      "rex": [
        "^([0-9a-z]{40})$",
      ]
    },
    "end_of_previous_period": {
      "type": "date",
      "min": "2018-12-30",
      "max": "2022-01-01",
      "max_nulls": 0
    },
    "end_of_this_period": {
      "type": "date",
      "min": "2019-01-07",
      "max": "2022-01-01",
      "max_nulls": 0
    },
    "end_of_this_period:time-before-now": {
      "min": "6 days",
       "transform": "time-before-now"
    },
    "end_of_this_period:time-before-now": {
      "min": "0 days",
      "transform": "time-before-now"
    },
    "postal_district": {
      "type": "string",
      "min_length": 2,
      "max_length": 5,
      "max_nulls": 0,
      "rex": [
         "^([A-Z]{1,2})([0-9]{1,2})([A-Z]?)$"
      ]
    },
    "sex": {
      "type": "string",
      "min_length": 1,
      "max_length": 1,
      "max_nulls": 0,
      "allowed_values": [
        "F",
        "M"
      ]
    },
    "total_credits": {
      "type": "real",
      "min": 0,
      "max": 10000000,
      "sign": "non-negative",
      "max_nulls": 0
    },
    "total_debits": {
      "type": "real",
      "min": -10000000,
      "max": 0,
      "sign": "non-positive",
      "max_nulls": 0
    },
    "final_balance": {
      "type": "real",
      "min": -10000000,
      "max": 10000000,
      "max_nulls": 0
    }
  },
  "dataset": {
    "required_fields": [
      "pseudoid",
      "end_of_previous_period",
      "end_of_this_period",
      "postal_district",
      "sex",
      "total_credits",
      "total_debits",
      "final_balance"
    ],
    "allowed_fields": [
      "pseudoid",
      "end_of_previous_period",
      "end_of_this_period",
      "postal_district",
      "sex",
      "total_credits",
      "total_debits",
      "final_balance"
    ],
    "allowed_all_null_fields": []
  },
  "field_groups": {
    “end_of_previous_period,end_of_this_period": {
      "lt": true
    }
  }
}
```

DATA VERIFICATION



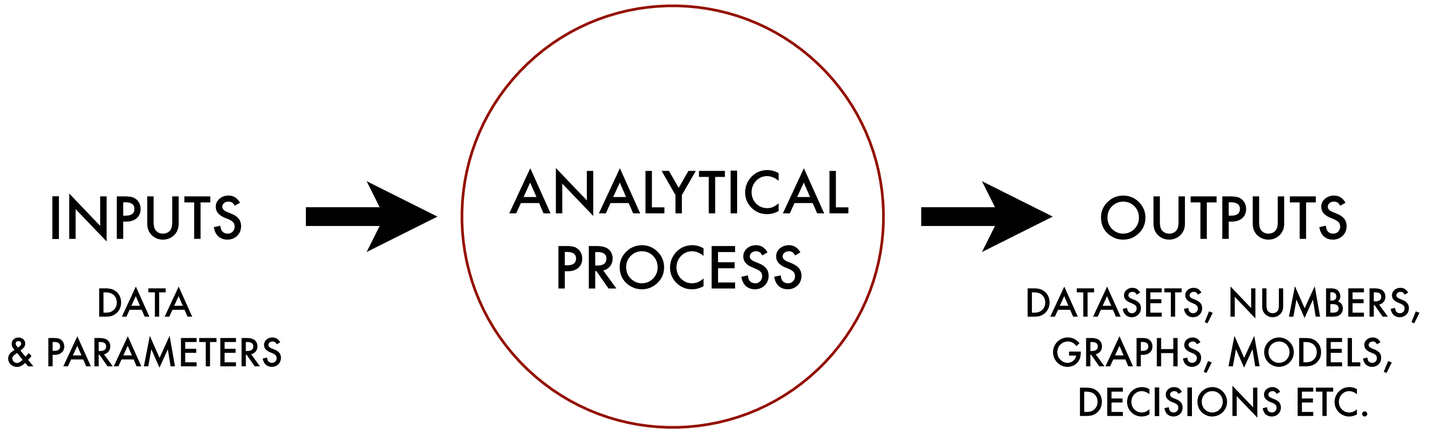
GENERATING CONSTRAINTS & VERIFYING DATA



training data

operational data

REFERENCE TESTS



Develop a verification procedure (`diff`) *and periodically rerun:*
do the same inputs (still) produce the same (or equivalent) outputs?

TDDA LIBRARY SUPPORT FOR REFERENCE TESTS

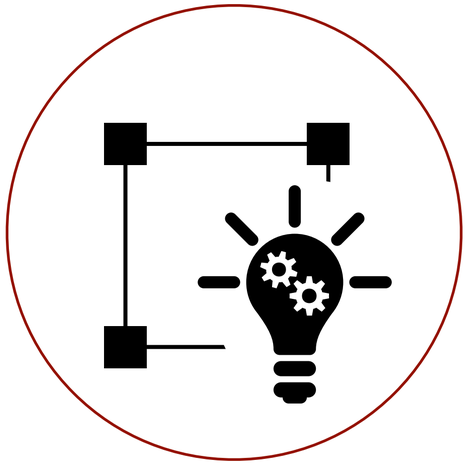
- With Data Science, the outputs are often too complex to produce by hand

- Output artefacts can be large and complex vary *even when semantic results don't vary*

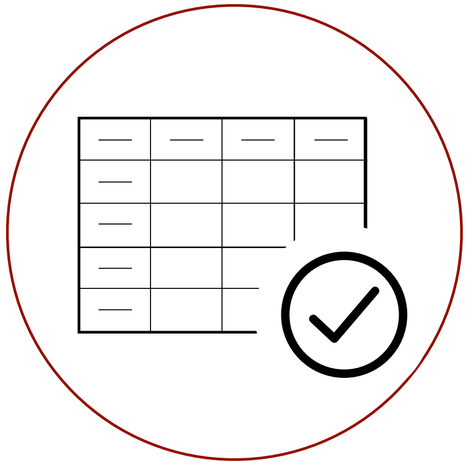
  - Output artefacts include random IDs, dates, run numbers, version numbers, environment information etc.

- TDDA library provides support for testing that everything that should remain consistent does, while allowing variation in the artefacts themselves, and regeneration when fixes mean that reference (target) results change.

TDDA LIBRARY FEATURES



GENERATE
CONSTRAINTS
FROM DATA



VERIFY
DATA WITH
CONSTRAINTS

TEST SUPPORT FOR
CODE WITH COMPLEX,
NOT-ALWAYS-IDENTICAL
OUTPUTS

CLASSES OF ERRORS IDENTIFIED

(more detail available in written transcript)

- 🙈 Rolling values with missing weeks for some customers
- 🤦 Inconsistent handling of nulls (missing values)
- 🤦‍♀️ Inconsistent handling of invalid values
- 😡 CSV precision (weird behaviour in R library **write.csv** in **utils** package in R)

CLASSES OF ERRORS IDENTIFIED

(more detail available in written transcript)

🤪 Numerical accuracy

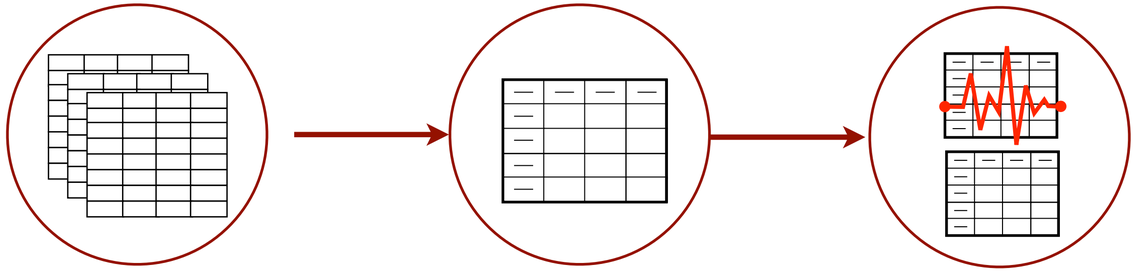
- Comparisons to nearest half penny
- Counts above thresholds — Expressing in whole pennies and rounding to ints

😭 Bin allocation inconsistent because of numerical (in)accuracies

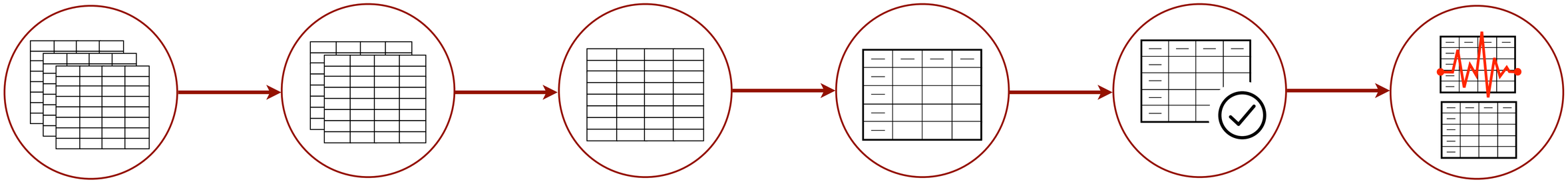
🥵 Incorrect Loop Unrolling (copy/paste error)

🤬 Inconsistent sort ordering (and mode tiebreaks)

ORIGINAL ANALYSIS PIPELINE



DECOMPOSED ANALYSIS PIPELINE



RAW DATA

EXTRACT, TRANSFORM LOAD

MEASURE

ANALYSE

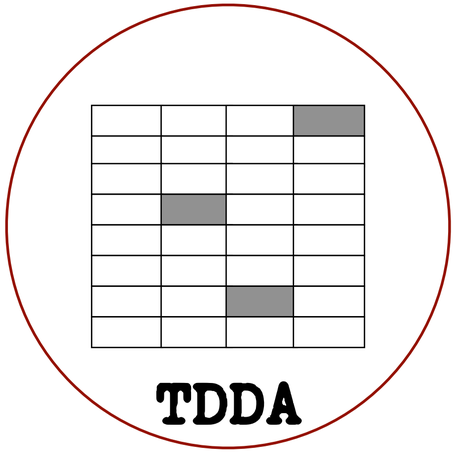
APPLY DISCLOSURE CONTROLS

GRAPHING, MAPPING REPORT CONTRUCTION

OUTPUT VERIFICATION

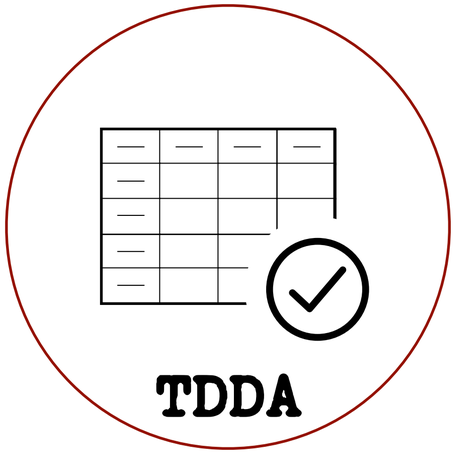
CORRECTNESS:
TESTS
REPRODUCED

✔



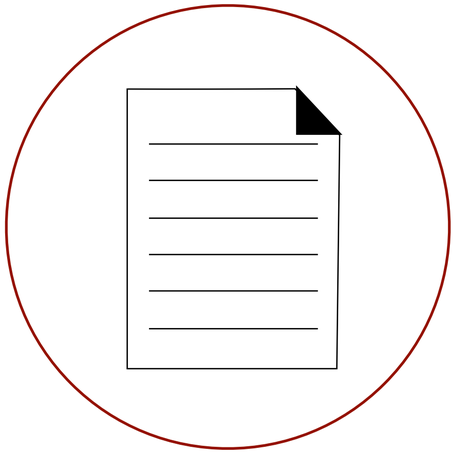
MINIMUM
CELL
SIZES

✔



CHECKS FOR
PERSONAL DATA
(“PII”)

✔



AUDIT TRAIL
& DATA LINEAGE
CHECKS

✔

AUDIT TRAIL & DATA LINEAGE

```
/.../audit/miro/xxxxxx-by-pa-enhanced.csv
(hash 2a4cf239fed057281c3a513888492496)
compared to
/.../audit/r/xxxxxx-by-pa-enhanced.csv
(hash 59e5540d687daff81f066cc7884efb9c)
validated as equivalent on 2021-01-05T10:34:51Z.
    ⋮
All files produce for report validated successfully.
```

Closing thoughts

Why is this lying bastard lying to me?

— Louis Heren
(Journalist, The Times)

The Data Scientist's Version:

How is this misleading data misleading me?

ABC (Blake's Dictum)*

Always Be Closing

– Blake (Alec Baldwin)
Glengarry Glen Ross

The Data Scientist's Version:

Always Be Checking

*not William Blake!


globalopenfinance.com • stochasticsolutions.com


http://tdda.info


https://github.com/tdda


njr@StochasticSolutions.com Pei-Shan.Yu@ed.ac.uk


https://linkedin.com/in/njradcliffe


#tdda*

* *tweet (DM) us your email address for invitation. Or email me.*


@tdda0 @njr0 @StochasticSolns @GOFCoE

Correct interpretation: Zero (Error of interpretation: Letter "Oh")

https://stochasticsolutions.com/pdf/toronto2021-reproducibility-in-safe-haven-transcript.pdf

https://stochasticsolutions.com/pdf/toronto2021-reproducibility-in-safe-haven-slides.pdf