The National Academies of SCIENCES ENGINEERING MEDICINE

Panel on Reproducibility and Transparency

Transparency of Software and Code

Lars Vilhuber

Cornell University

Benefits of Transparency and Reproducibility

- efficiency,
- innovation and progress,
- trust and confidence,
- and the value from the use of the data products

National Academies of Sciences, Engineering, and Medicine. 2022. Transparency in Statistical Information for the National Center for Science and Engineering Statistics and All Federal Statistical Agencies. Washington, DC: The National Academies Press. https://doi.org/10.17226/26360.

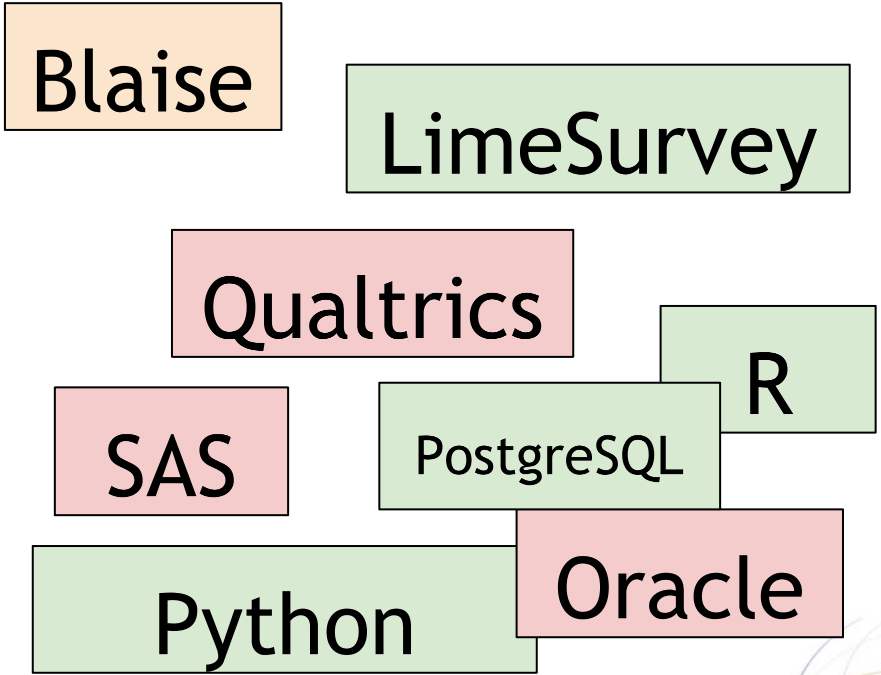
The National Academies of SCIENCES ENGINEERING MEDICINE

Transparency and:

Transparency requires **“the provision of sufficiently detailed documentation”**

Transparency and:

Transparency requires **“the provision of sufficiently detailed documentation”**



Blaise

LimeSurvey

Qualtrics

R

SAS

PostgreSQL

Oracle

Python

Transparency and: *Proprietary Software*

Transparency requires “**the provision of sufficiently detailed documentation**”

Q: What to do when software is proprietary?

Transparency and: ~~Proprietary~~ Software

Transparency requires “**the provision of sufficiently detailed documentation**”

Q: What to do when software is **proprietary**?

A: Clearly describe use of software *(accessibility, price, version)*

Transparency and: Code

Code is
instructions to make software function
(could also be source code for software)

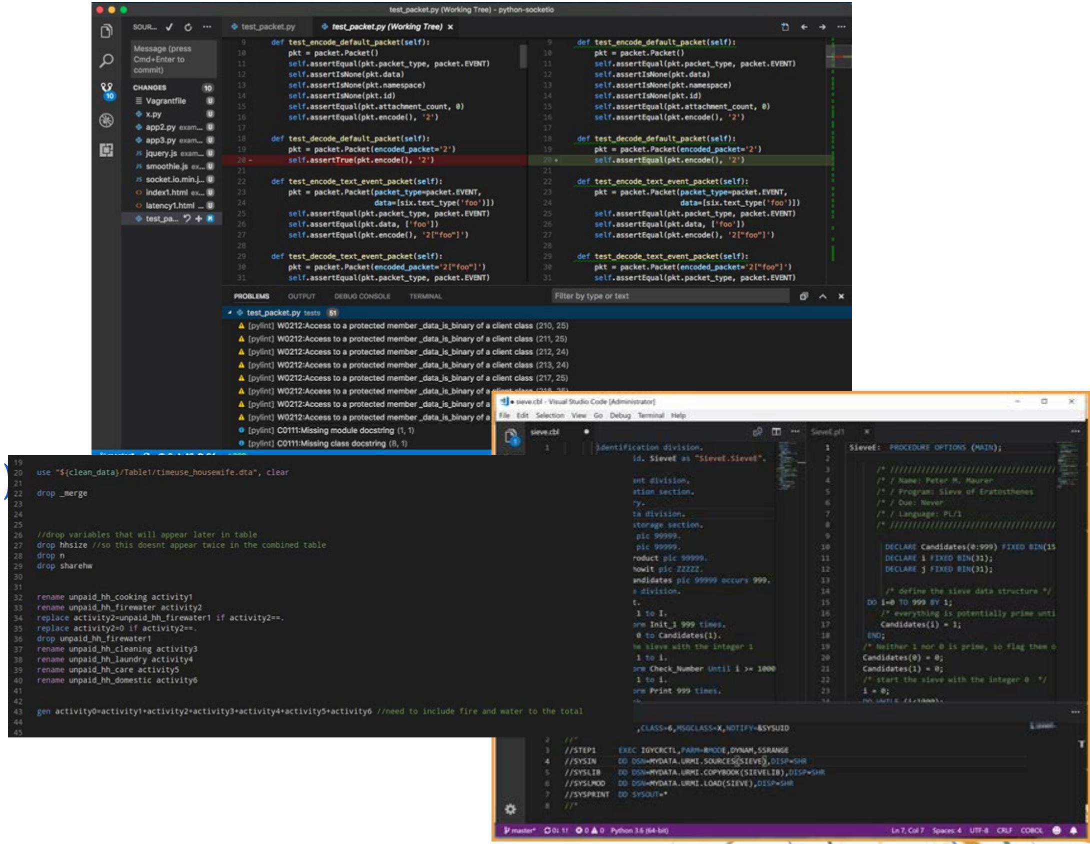
- functionality of code
- development of code (when, who)
- dependencies

Transparency and: *Code*

Code is

instructions to make software function

(could also be source code for software)



```
    def test_encode_default_packet(self):
        pkt = packet.Packet()
        self.assertEqual(pkt.packet_type, packet.EVENT)
        self.assertIsNone(pkt.data)
        self.assertIsNone(pkt.namespace)
        self.assertIsNone(pkt.id)
        self.assertEqual(pkt.attachment_count, 0)
        self.assertEqual(pkt.encode(), '2')

    def test_decode_default_packet(self):
        pkt = packet.Packet(encoded_packet='2')
        self.assertTrue(pkt.encode(), '2')

    def test_encode_text_event_packet(self):
        pkt = packet.Packet(packet_type=packet.EVENT,
                            data=[six.text_type('foo')])
        self.assertEqual(pkt.packet_type, packet.EVENT)
        self.assertEqual(pkt.data, ['foo'])
        self.assertEqual(pkt.encode(), '2["foo"]')

    def test_decode_text_event_packet(self):
        pkt = packet.Packet(encoded_packet='2["foo"]')
        self.assertEqual(pkt.packet_type, packet.EVENT)
```

```
    def test_encode_default_packet(self):
        pkt = packet.Packet()
        self.assertEqual(pkt.packet_type, packet.EVENT)
        self.assertIsNone(pkt.data)
        self.assertIsNone(pkt.namespace)
        self.assertIsNone(pkt.id)
        self.assertEqual(pkt.attachment_count, 0)
        self.assertEqual(pkt.encode(), '2')

    def test_decode_default_packet(self):
        pkt = packet.Packet(encoded_packet='2')
        self.assertEqual(pkt.encode(), '2')

    def test_encode_text_event_packet(self):
        pkt = packet.Packet(packet_type=packet.EVENT,
                            data=[six.text_type('foo')])
        self.assertEqual(pkt.packet_type, packet.EVENT)
        self.assertEqual(pkt.data, ['foo'])
        self.assertEqual(pkt.encode(), '2["foo"]')

    def test_decode_text_event_packet(self):
        pkt = packet.Packet(encoded_packet='2["foo"]')
        self.assertEqual(pkt.packet_type, packet.EVENT)
```

```
[pylint] W0212:Access to a protected member _data_is_binary of a client class (210, 25)
[pylint] W0212:Access to a protected member _data_is_binary of a client class (211, 25)
[pylint] W0212:Access to a protected member _data_is_binary of a client class (212, 24)
[pylint] W0212:Access to a protected member _data_is_binary of a client class (213, 24)
[pylint] W0212:Access to a protected member _data_is_binary of a client class (217, 25)
[pylint] W0212:Access to a protected member _data_is_binary of a
[pylint] W0212:Access to a protected member _data_is_binary of a
[pylint] W0212:Access to a protected member _data_is_binary of a
[pylint] C0111:Missing module docstring (1, 1)
[pylint] C0111:Missing class docstring (8, 1)
```

```
use "${clean_data}/Table1/timeuse_housewife.dta", clear

drop _merge



//drop variables that will appear later in table
drop hhsize //so this doesnt appear twice in the combined table
drop n
drop sharehw


rename unpaid_hh_cooking activity1
rename unpaid_hh_firewater activity2
replace activity2=unpaid_hh_firewater1 if activity2==.
replace activity2=0 if activity2==.
drop unpaid_hh_firewater1
rename unpaid_hh_cleaning activity3
rename unpaid_hh_laundry activity4
rename unpaid_hh_care activity5
rename unpaid_hh_domestic activity6


gen activity0=activity1+activity2+activity3+activity4+activity5+activity6 //need to include fire and water to the total
```

sieve.cbl - Visual Studio Code [Administrator]
File Edit Selection View Go Debug Terminal Help
sieve.cbl
SieveE.pl1
master* 0↓ 1↑ 0 0 Python 3.6 (64-bit) Ln 7, Col 7 Spaces: 4 UTF-8 CRLF COBOL

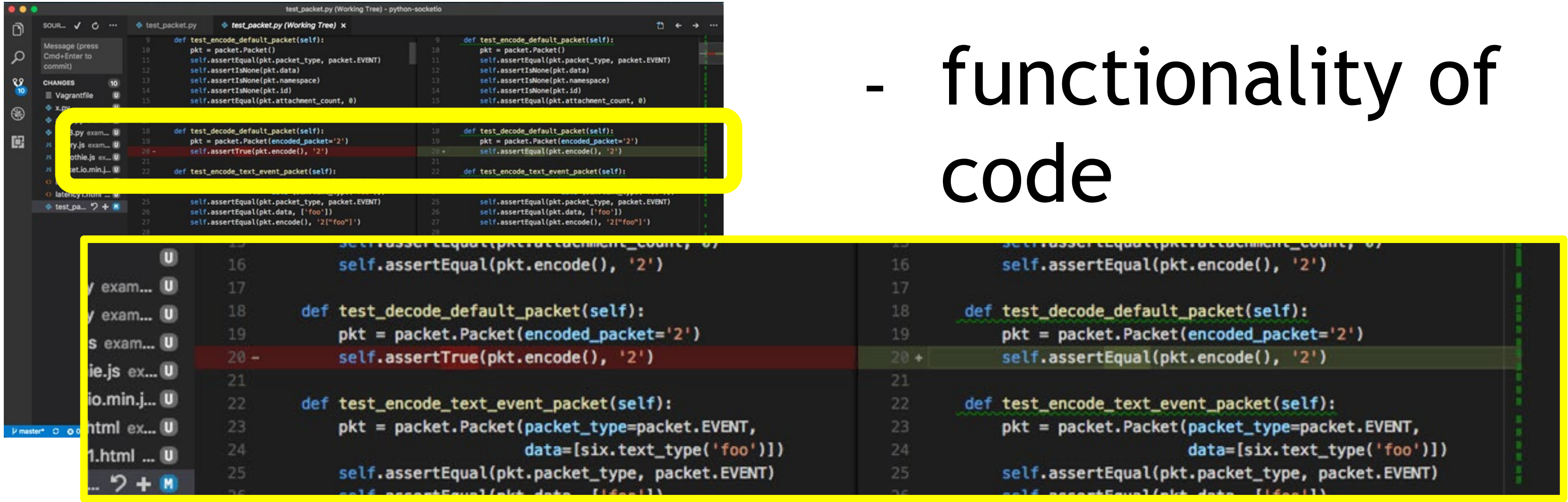
```
identification division.
id. SieveE as "SieveE.SieveE".
ent division.
ation section.
ry.
ta division.
storage section.
 pic 99999.
 pic 99999.
roduct pic 99999.
howit pic ZZZZZ.
andidates pic 99999 occurs 999.
 division.
t.
 1 to I.
rm Init_1 999 times.
 0 to Candidates(1).
he sieve with the integer 1
 1 to i.
rm Check_Number Until i >= 1000
 1 to i.
rm Print 999 times.
```

```
SieveE:  PROCEDURE OPTIONS (MAIN);

    /* ////////////////////////////////
    /* / Name: Peter M. Maurer
    /* / Program: Sieve of Eratosthenes
    /* / Due: Never
    /* / Language: PL/1
    /* ////////////////////////////////

     DECLARE Candidates(0:999) FIXED BIN(15
     DECLARE i FIXED BIN(31);
     DECLARE j FIXED BIN(31);

     /* define the sieve data structure */
  DO i=0 TO 999 BY 1;
    /* everything is potentially prime unti
    Candidates(i) = 1;
  END;
 /* Neither 1 nor 0 is prime, so flag them o
 Candidates(0) = 0;
 Candidates(1) = 0;
 /* start the sieve with the integer 0  */
 i = 0;
```

```
,CLASS=6,MSGCLASS=X,NOTIFY=&SYSUID
//*
//STEP1     EXEC IGYCRCTL,PARM=RMODE,DYNAM,SSRANGE
//SYSIN     DD DSN=MYDATA.URMI.SOURCES(SIEVE),DISP=SHR
//SYSLIB    DD DSN=MYDATA.URMI.COPYBOOK(SIEVELIB),DISP=SHR
//SYSLMOD   DD DSN=MYDATA.URMI.LOAD(SIEVE),DISP=SHR
//SYSPRINT  DD SYSOUT=*
//*
```

The National Academies of SCIENCES ENGINEERING MEDICINE

Transparency and: *Code*

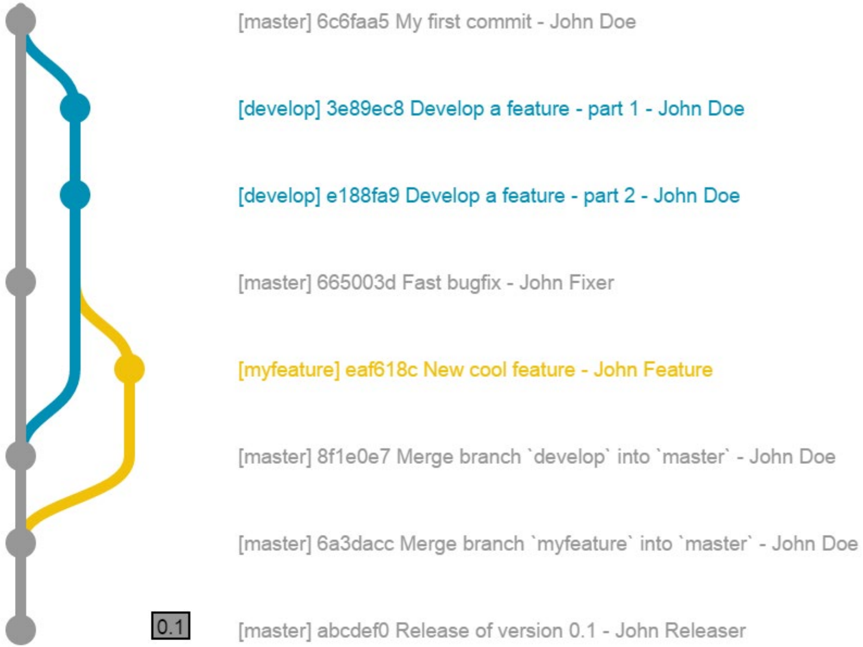


```
        self.assertEqual(pkt.encode(), '2')

    def test_decode_default_packet(self):
        pkt = packet.Packet(encoded_packet='2')
-       self.assertTrue(pkt.encode(), '2')

    def test_encode_text_event_packet(self):
        pkt = packet.Packet(packet_type=packet.EVENT,
                            data=[six.text_type('foo')])
        self.assertEqual(pkt.packet_type, packet.EVENT)
```

```
        self.assertEqual(pkt.encode(), '2')

    def test_decode_default_packet(self):
        pkt = packet.Packet(encoded_packet='2')
+       self.assertEqual(pkt.encode(), '2')

    def test_encode_text_event_packet(self):
        pkt = packet.Packet(packet_type=packet.EVENT,
                            data=[six.text_type('foo')])
        self.assertEqual(pkt.packet_type, packet.EVENT)
```

- functionality of code

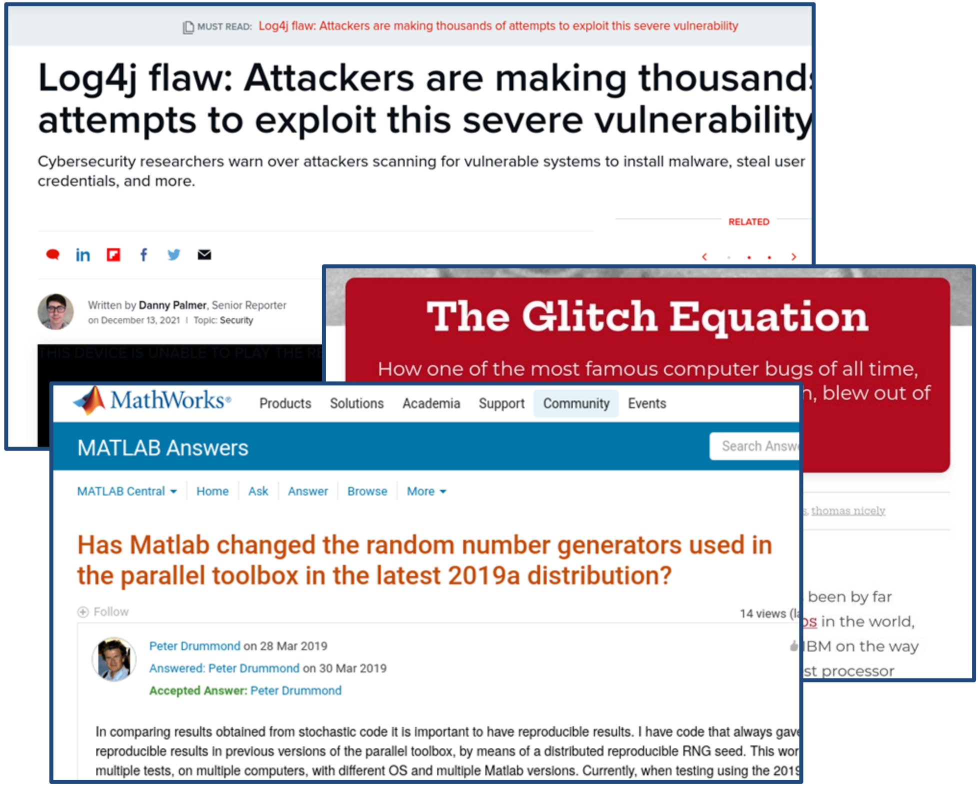
Transparency and: **Code**



```
[master] 6c6faa5 My first commit - John Doe
[develop] 3e89ec8 Develop a feature - part 1 - John Doe
[develop] e188fa9 Develop a feature - part 2 - John Doe
[master] 665003d Fast bugfix - John Fixer
[myfeature] eaf618c New cool feature - John Feature
[master] 8f1e0e7 Merge branch `develop` into `master` - John Doe
[master] 6a3dacc Merge branch `myfeature` into `master` - John Doe
[master] abcdef0 Release of version 0.1 - John Releaser
```

- functionality of code
- development of code (when, who)
  - Includes release policy

Transparency and: *Code*



MUST READ: Log4j flaw: Attackers are making thousands of attempts to exploit this severe vulnerability

Log4j flaw: Attackers are making thousand attempts to exploit this severe vulnerability

Cybersecurity researchers warn over attackers scanning for vulnerable systems to install malware, steal user credentials, and more.

RELATED

Written by Danny Palmer, Senior Reporter on December 13, 2021 | Topic: Security

THIS DEVICE IS UNABLE TO PLAY THE R

The Glitch Equation

How one of the most famous computer bugs of all time, n, blew out of

s, thomas nicely

been by far os in the world, IBM on the way st processor

Has Matlab changed the random number generators used in the parallel toolbox in the latest 2019a distribution?

14 views (la

Peter Drummond on 28 Mar 2019
Answered: Peter Drummond on 30 Mar 2019
Accepted Answer: Peter Drummond

In comparing results obtained from stochastic code it is important to have reproducible results. I have code that always gave reproducible results in previous versions of the parallel toolbox, by means of a distributed reproducible RNG seed. This wor multiple tests, on multiple computers, with different OS and multiple Matlab versions. Currently, when testing using the 2019

- functionality of code
- development of code (when, who)
- dependencies

Transparency and Software

Recommendation 4.1

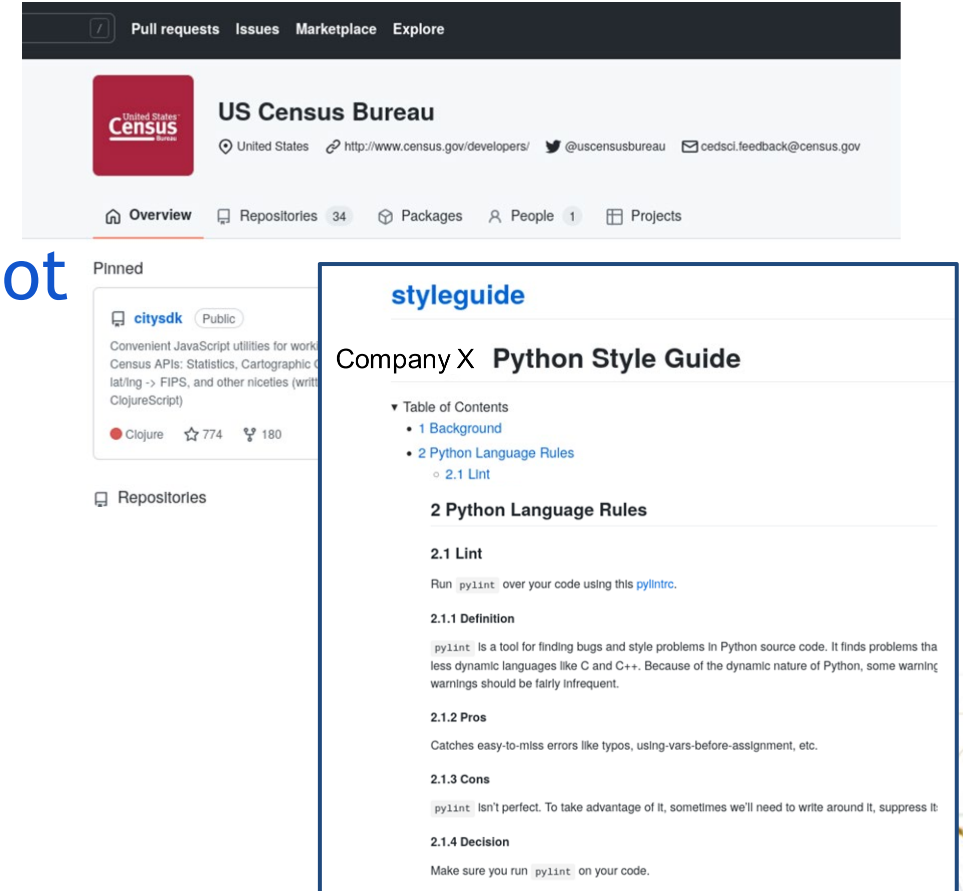
Agencies that produce federal statistics [...]

- should review and make a priority of
- adopting modern information technology tools that assist in
- **collaborative software development and documentation of workflow and methodology.**

Transparency and Software

Recommendation 4.1

- Use versioning systems
    - Use them broadly (not just selectively)
- Create and use style guides
    - Not just for “developers”



US Census Bureau

United States http://www.census.gov/developers/ @uscensusbureau cedsci.feedback@census.gov

Convenient JavaScript utilities for worki
Census APIs: Statistics, Cartographic (
lat/lng -> FIPS, and other niceties (writt
ClojureScript)

Company X Python Style Guide

▾ Table of Contents
- 1 Background
- 2 Python Language Rules
    - 2.1 Lint

2 Python Language Rules

2.1 Lint

Run `pylint` over your code using this pylintrc.

2.1.1 Definition

`pylint` is a tool for finding bugs and style problems in Python source code. It finds problems tha
less dynamic languages like C and C++. Because of the dynamic nature of Python, some warning
warnings should be fairly infrequent.

2.1.2 Pros

Catches easy-to-miss errors like typos, using-vars-before-assignment, etc.

2.1.3 Cons

`pylint` isn’t perfect. To take advantage of it, sometimes we’ll need to write around it, suppress it

2.1.4 Decision

Make sure you run `pylint` on your code.

Some particular notes

Transparency and: *Surveys*

Agencies have exemplary tradition of **publishing questionnaires**

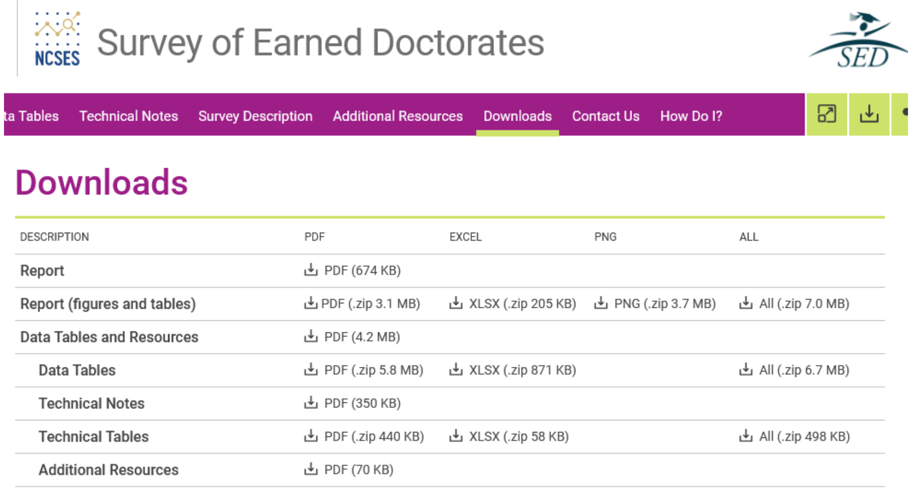
- Most often not in re-usable formats (such as DDI, Blaise)
- Transparency of “code” would allow greater reuse/consistency

The National Academies of SCIENCES ENGINEERING MEDICINE

Example: Survey of Earned Doctorates

"*Data collection*. In 2020, for the first time, the SED data collection did not use the self-administered paper questionnaire. The SED was completed primarily by self-administered Web survey with a small number of nonrespondents contacted to complete computer-assisted telephone interviewing (CATI)."

https://ncses.nsf.gov/pubs/nsf22300/technical-notes#survey-design



Survey of Earned Doctorates

Downloads

DESCRIPTION	PDF	EXCEL	PNG	ALL
Report	PDF (674 KB)			
Report (figures and tables)	PDF (.zip 3.1 MB)	XLSX (.zip 205 KB)	PNG (.zip 3.7 MB)	All (.zip 7.0 MB)
Data Tables and Resources	PDF (4.2 MB)			
Data Tables	PDF (.zip 5.8 MB)	XLSX (.zip 871 KB)		All (.zip 6.7 MB)
Technical Notes	PDF (350 KB)			
Technical Tables	PDF (.zip 440 KB)	XLSX (.zip 58 KB)		All (.zip 498 KB)
Additional Resources	PDF (70 KB)			

Blaise

LimeSurvey

Qualtrics

The National Academies of SCIENCES ENGINEERING MEDICINE

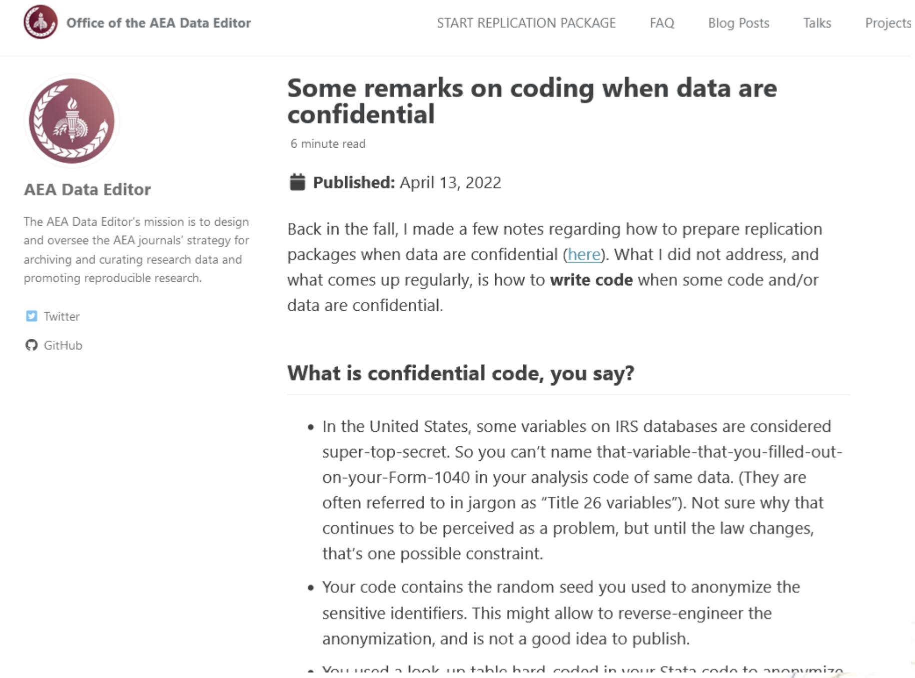
Transparency and: *Processing*

All data cleaning and preparation is *(or should be)* done by code

- Can such code be made available?
- The processed data is probably still confidential (PII)

Analogy: Code for academic articles

All data cleaning and preparation code *(and instructions)* MUST be provided.



Office of the AEA Data Editor

START REPLICATION PACKAGE FAQ Blog Posts Talks Projects

AEA Data Editor

The AEA Data Editor's mission is to design and oversee the AEA journals' strategy for archiving and curating research data and promoting reproducible research.

Twitter

GitHub

Some remarks on coding when data are confidential

6 minute read

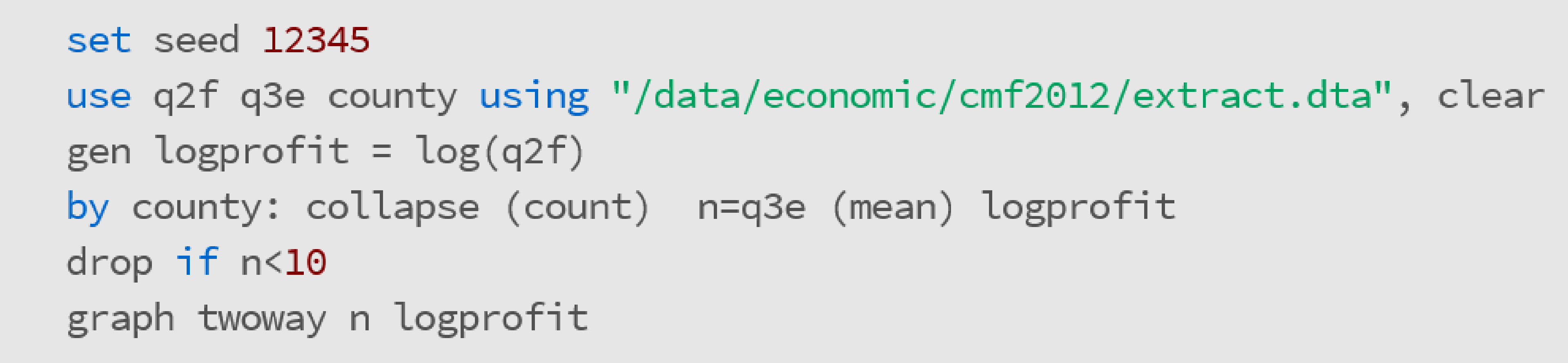
Published: April 13, 2022

Back in the fall, I made a few notes regarding how to prepare replication packages when data are confidential (here). What I did not address, and what comes up regularly, is how to **write code** when some code and/or data are confidential.

What is confidential code, you say?

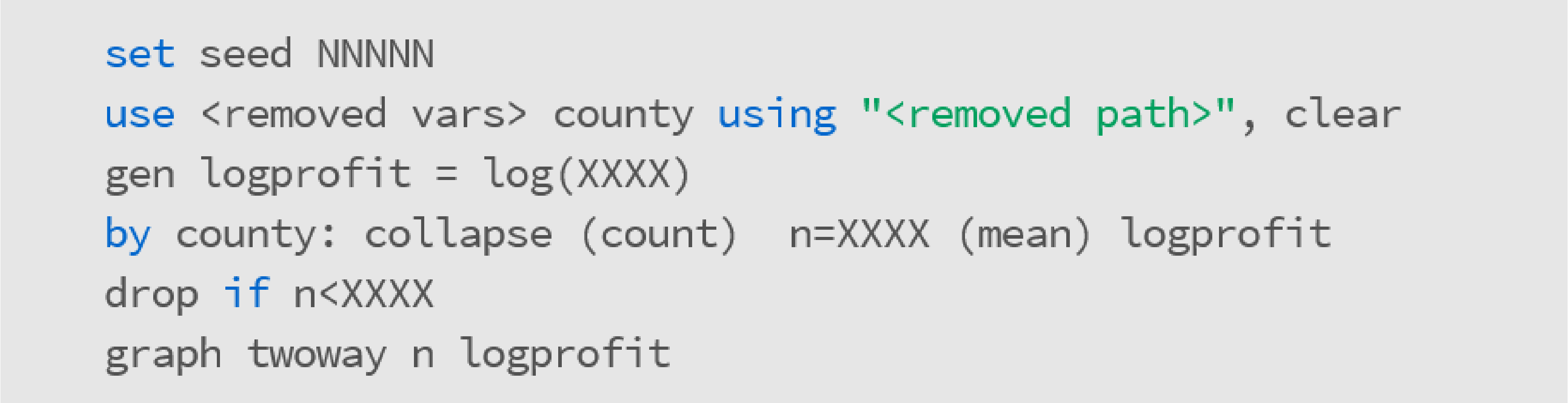
- In the United States, some variables on IRS databases are considered super-top-secret. So you can't name that-variable-that-you-filled-out-on-your-Form-1040 in your analysis code of same data. (They are often referred to in jargon as "Title 26 variables"). Not sure why that continues to be perceived as a problem, but until the law changes, that's one possible constraint.
- Your code contains the random seed you used to anonymize the sensitive identifiers. This might allow to reverse-engineer the anonymization, and is not a good idea to publish.
- You used a look-up table hard-coded in your Stata code to anonymize

Do's and **don't's**



```
set seed 12345
use q2f q3e county using "/data/economic/cmf2012/extract.dta", clear
gen logprofit = log(q2f)
by county: collapse (count)  n=q3e (mean) logprofit
drop if n<10
graph twoway n logprofit
```

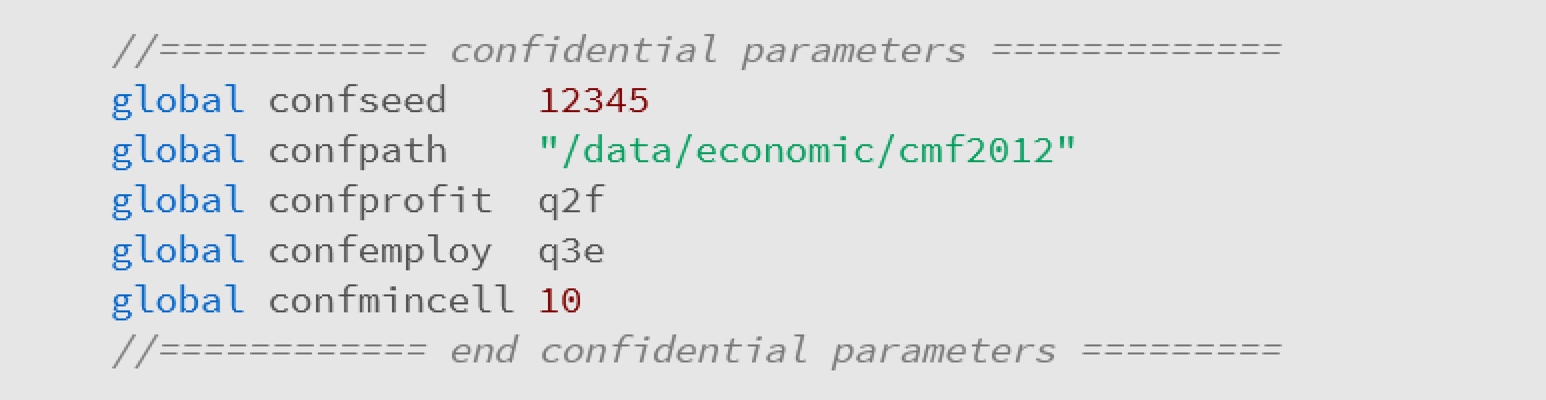
Do’s and **don’t’s**



```
set seed NNNNN
use <removed vars> county using "<removed path>", clear
gen logprofit = log(XXXX)
by county: collapse (count)  n=XXXX (mean) logprofit
drop if n<XXXX
graph twoway n logprofit
```

Do’s and don’t’s

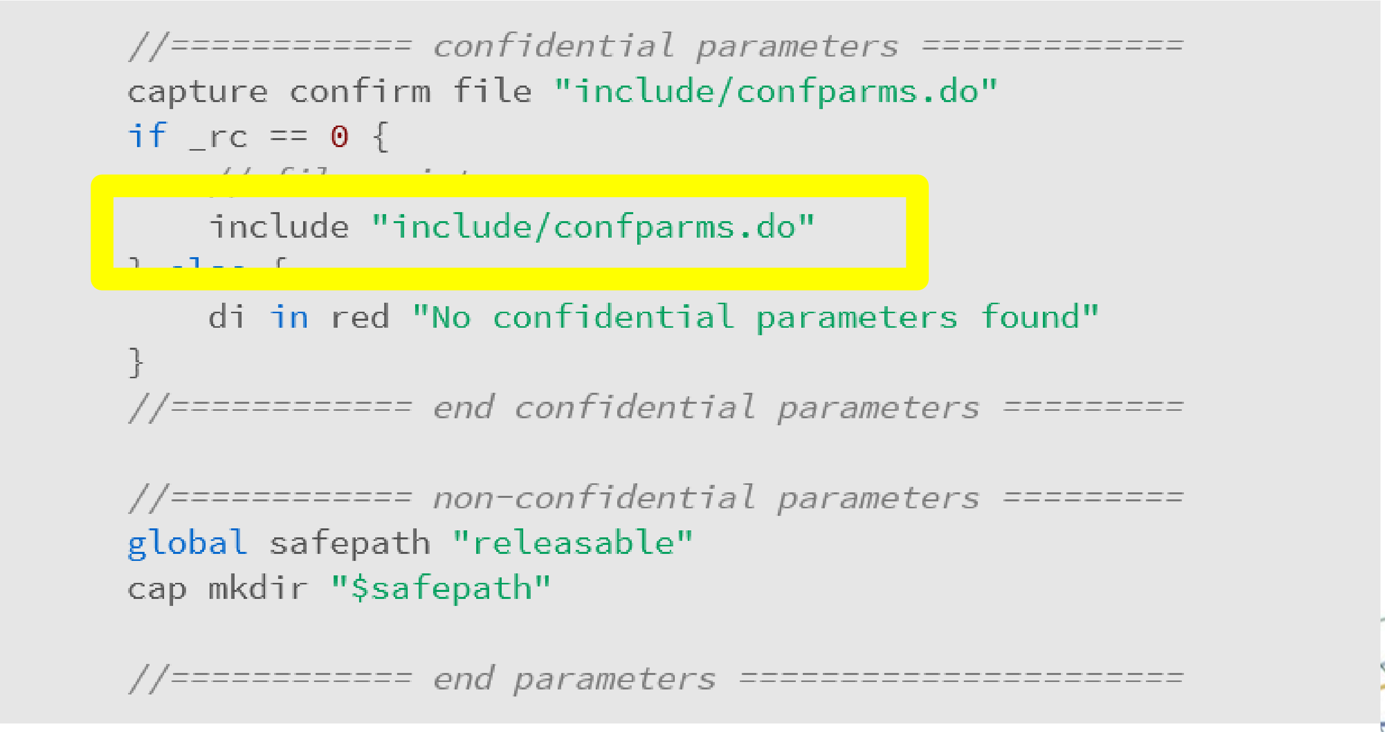
Auxiliary file `include/confparms.do` (not released)



```
//============ confidential parameters =============
global confseed    12345
global confpath    "/data/economic/cmf2012"
global confprofit  q2f
global confemploy  q3e
global confmincell 10
//============ end confidential parameters =========
```

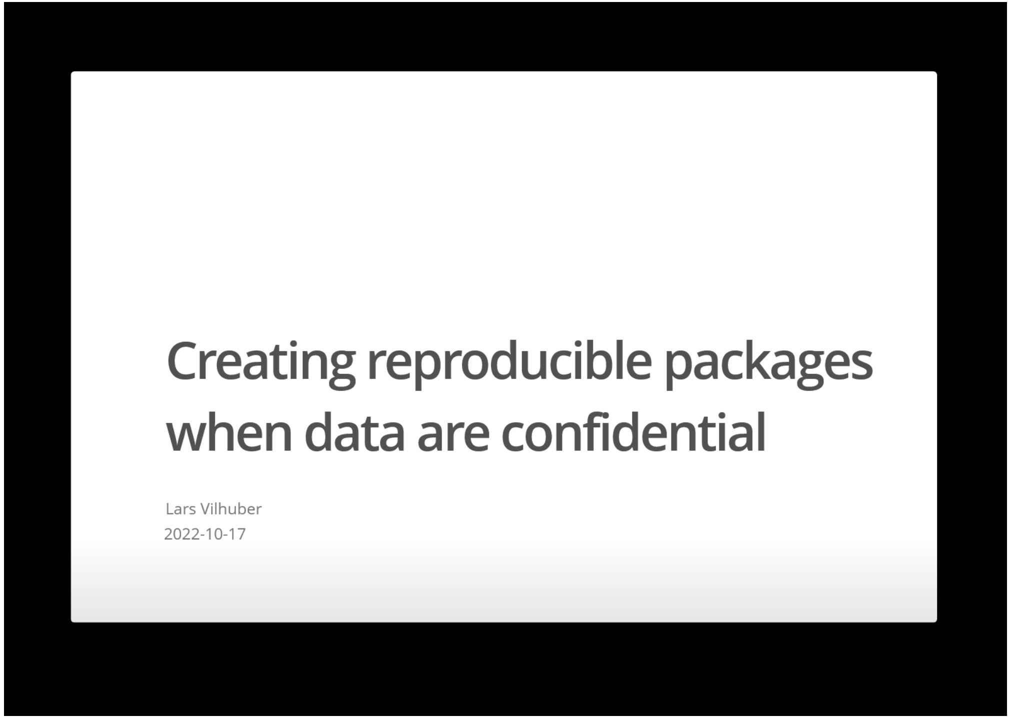
Do's and don't's

Main file `main.do`:



```
//============ confidential parameters =============
capture confirm file "include/confparms.do"
if _rc == 0 {
    // file exists
    include "include/confparms.do"
} else {
    di in red "No confidential parameters found"
}
//============ end confidential parameters =========

//============ non-confidential parameters =========
global safepath "releasable"
cap mkdir "$safepath"

//============ end parameters ======================
```

Do’s and don’t’s



Creating reproducible packages when data are confidential

Lars Vilhuber

2022-10-17

lars.vilhuber.com/p/fsrdc2022/

Transparency and: *Processing*

Tracing of code execution is hard (logging)

- **Already occurs for security purposes**
- May not need to be at the finest level for transparency
- May not need to be publicly available (but auditable)

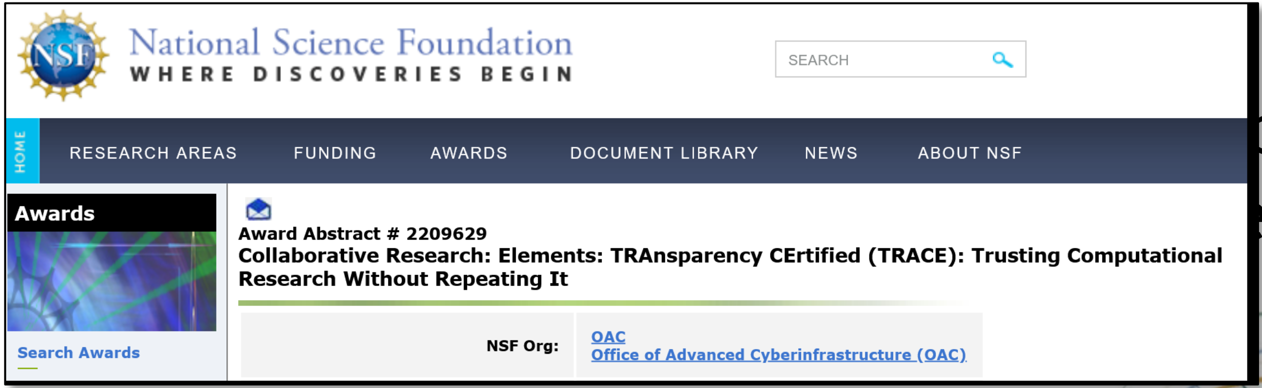
Transparency and: *Processing*

Tracing of code execution is hard (logging)

- **Already occurs for security purposes**
- May not need to be at the finest level for

be

e (but



Award Abstract # 2209629

Collaborative Research: Elements: TRAnsparency CErtified (TRACE): Trusting Computational Research Without Repeating It

NSF Org: OAC Office of Advanced Cyberinfrastructure (OAC)

Transparency and: *Manual steps*

Some manual steps in processing may be **unavoidable**

Transparency implies

- that it be identified
- documented (instructions, training manuals)
- ideally publicly

Transparency and: *Manual steps*

Some manual steps in processing may be **unavoidable**

Can include:

- Rules for manual edits
- Human edits

Transparency and: *Consequences*

Transparency can be hard

- 1,000 of people looking “over your shoulder”
- Errors **<u>will</u>** be found

Transparency and: *Consequences*

Transparency can be hard

… but valuable

- Self-disciplining device
- Possible crowd-sourcing of solutions

Transparency and: *Policies*

Transparency needs frameworks

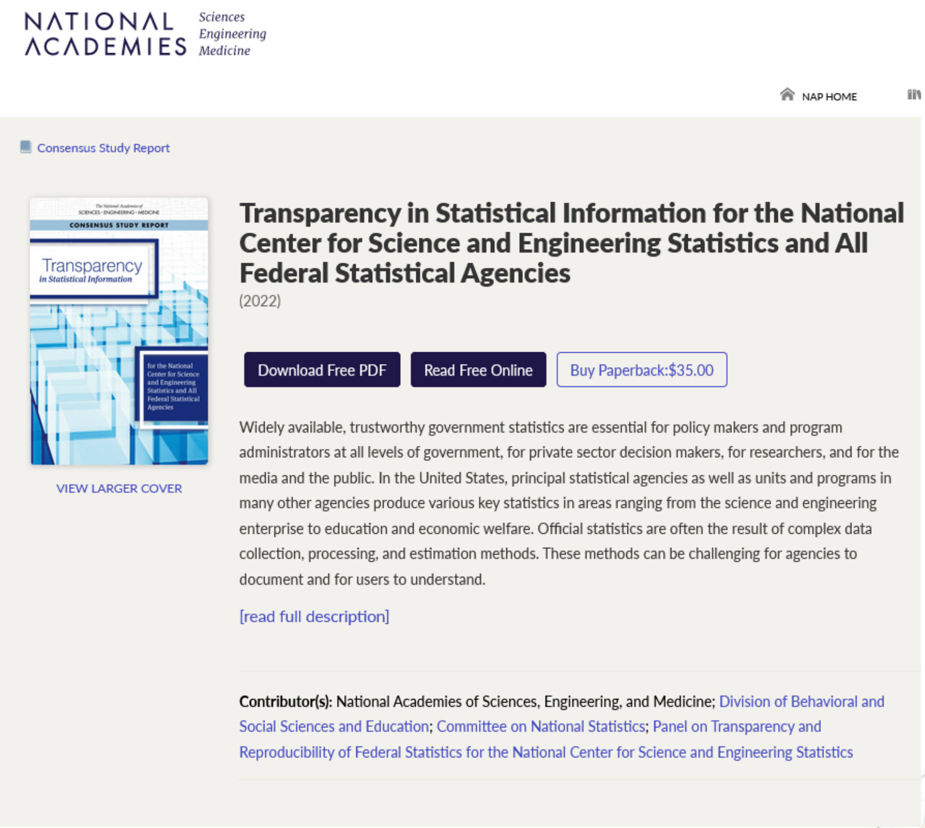
- Internal policies on how to respond to (legitimate) criticism
- Support for the process

Transparency and: *Policies*

Transparency needs frameworks

- Coding guides
- Continuous review process
  - For quality
  - For security
- Training!

doi.org/10.17226/26360



Transparency in Statistical Information for the National Center for Science and Engineering Statistics and All Federal Statistical Agencies

(2022)

Widely available, trustworthy government statistics are essential for policy makers and program administrators at all levels of government, for private sector decision makers, for researchers, and for the media and the public. In the United States, principal statistical agencies as well as units and programs in many other agencies produce various key statistics in areas ranging from the science and engineering enterprise to education and economic welfare. Official statistics are often the result of complex data collection, processing, and estimation methods. These methods can be challenging for agencies to document and for users to understand.

[read full description]

Contributor(s): National Academies of Sciences, Engineering, and Medicine; Division of Behavioral and Social Sciences and Education; Committee on National Statistics; Panel on Transparency and Reproducibility of Federal Statistics for the National Center for Science and Engineering Statistics