An Abstract Interpretation Framework for Input Data Usage

Caterina Urban and Peter Müller

Department of Computer Science
ETH Zürich, Switzerland
Data is Revolutionizing Industries
Data Science Pipeline

- data sources
- stored data
- pre-processing
- data analysis
- results
For Big-Data Scientists, ‘Janitor Work’ Is Key Hurdle to Insights

By STEVE LOHR AUG. 17, 2014

Technology revolutions come in measured, sometimes foot-dragging steps. The lab science and marketing enthusiasm tend to underestimate the bottlenecks to progress that must be overcome with hard work and practical engineering.

The field known as “big data” offers a contemporary case study. The catchphrase stands for the modern abundance of digital data from many sources — the web, sensors, smartphones and corporate databases — that can be mined with clever software for discoveries and insights. Its promise is smarter, data-driven decision-making in every field. That is why data scientist is the economy’s hot new job.

Yet far too much handcrafted work — what data scientists call “data wrangling,” “data munging” and “data janitor work” — is still required. Data scientists, according to interviews and expert estimates, spend from 50 percent to 80 percent of their time mired in this more mundane labor of collecting and preparing unruly digital data, before it can be explored for useful nuggets.
The Reinhart-Rogoff Paper

American Economic Review: Papers & Proceedings 100 (May 2010): 573–578
http://www.aeaweb.org/articles.php?doi=10.1257/100.2.573

Growth in a Time of Debt

By Carmen M. Reinhart and Kenneth S. Rogoff

In this paper, we exploit a new multi-country historical dataset on public and external debt to search for a systemic relationship between high public debt levels, growth and inflation. Our main result is that whereas the link between growth and debt seems relatively weak at “normal” debt levels, median growth rates for countries with public debt over roughly 90 percent of GDP are about one percent lower than otherwise; average (mean) growth rates are several percent lower. Surprisingly, the relationship between public debt and growth is remarkably similar across emerging markets and advanced economies. This is not the case for inflation. We find no systematic relationship between high debt levels and inflation for advanced economies as a group (albeit with individual country exceptions including the United States). By contrast, in emerging market countries, high public debt levels coincide with higher inflation. Our topic would seem to be a timely one.

Public debt has been soaring in the wake of the recent global financial maelstrom, especially in the epicenter countries. This should not be surprising, given the experience of earlier severe financial crises. Outsize deficits and epic bank bailouts may be useful in fighting a downturn, but what is the long-run macroeconomic impact, especially against the backdrop of graying populations and entitlement insurance costs? Are sharply elevated public debts ultimately a manageable policy challenge?

Our approach here is decidedly empirical, taking advantage of a broad new historical dataset on public debt (in particular, central government debt) first presented in Carmen M. Reinhart and Kenneth S. Rogoff (2008, 2009b). Prior to this dataset, it was exceedingly difficult to get more than two or three decades of public debt data even for many rich countries, and virtually impossible for most emerging markets. Our results incorporate data on 44 countries spanning about 200 years. Taken together, the data incorporate over 3,700 annual observations covering a wide range of political systems, institutions, exchange rate and monetary arrangements, and historic circumstances.

We also employ more recent data on external debt, including debt owed both by governments and by private entities. For emerging markets, we find that there exists a significantly more severe threshold for total gross external debt (public and private)—which is almost exclusively denominated in a foreign currency—than for total public debt (the domestically issued component of which is largely denominated in home currency). When gross external debt reaches 60 percent of GDP, annual growth declines by about two percent; for levels of external debt in excess of 90 percent of GDP, growth rates are roughly cut in half. We are not in a position to calculate separate total external debt thresholds (as opposed to public debt thresholds) for advanced countries. The available time-series is too recent, beginning only in 2000. We do note, however, that external debt levels in advanced countries now average nearly 200 percent of GDP, with external debt levels being particularly high across Europe.

The focus of this paper is on the longer term macroeconomic implications of much higher public and external debt. The final section, however, searches for a systemic relationship between high external debt and growth for advanced economies. Our results incorporate data on 26 countries spanning about 200 years.
The Reinhart-Rogoff Paper

FAQ: Reinhart, Rogoff, and the Excel Error That Changed History

By Peter Coy  |  April 18, 2013

The Excel Depression

In this age of information, math errors can lead to disaster. NASA’s Mars Orbiter crashed because engineers forgot to convert to metric measurements; JPMorgan Chase’s “London Whale” venture went bad in part because modelers divided by a sum instead of an average. So, did an Excel coding error destroy the economies of the Western world?

The story so far: At the beginning of 2010, two Harvard economists, Carmen Reinhart and Kenneth Rogoff, circulated a paper, “Growth in a Time of Debt,” that purported to identify a critical “threshold,” a tipping point, for government indebtedness. Once debt exceeds 90 percent of gross domestic product, they claimed, economic growth drops off sharply.

Ms. Reinhart and Mr. Rogoff had credibility thanks to a widely admired earlier book on the history of financial

---

| Country | Coverage | 30 or less | 30 to 60 | 60 to 90 | 90 or above 30 or less |
|---------|----------|------------|----------|----------|------------------------|
|         |          | 3.7        | 3.0      | 3.5      | 1.7                    |
|         |          | 1.6        | 0.3      | 1.3      | -1.8                   |
|         |          | 3.6        |          |          | 13.3                   |
|         |          | -2.0       |          |          | n.a.                   |
|         |          | 2.4        |          |          | n.a.                   |
|         |          | n.a.       |          |          | 6.3                    |
|         |          | n.a.       |          |          | 9.9                    |
|         |          | n.a.       |          |          | 7.9                    |

---

[Table data: Real GDP growth and debt to GDP ratios for different countries and debt thresholds.]
Static Analysis Recipe

**practical tools**
targeting specific programs

**algorithmic approaches**
to decide program properties

**mathematical models**
of the program behavior
Static Analysis Recipe

- **practical tools**
  targeting specific programs

- **algorithmic approaches**
  to decide program properties

- **mathematical models**
  of the program behavior
Input Data Usage is a Hyperproperty

Unused Input Data

the outcome of the program does not depend on it

\[ P \models \mathcal{N}_K \iff [P] \in \mathcal{N}_K \]
Sound Input Data Usage Validation

\[ [P] \subset [P]^\triangledown \]

\[ [P]^\triangledown \subset T \]

\[ [P] \subset T \]

\[ [P]^\triangledown \in \mathcal{H} \]

\[ [P] \in \mathcal{H} \]

Trace Properties

Hyperproperties
Outcome Semantics

Partition executions based on their outcome

Input Data Usage
Outcome Semantics

Partition executions based on their outcome

Input Data Usage

Any possible value of the unused data

same outcome

\[ P \models \mathcal{N}_K \iff [P] \subseteq \mathcal{N}_K \]
Unused Input Data Analysis

- **practical tools**
  targeting specific programs

- **algorithmic approaches**
  to decide program properties

- **mathematical models**
  of the program behavior
Piecewise Unused Input Data Analysis

```python
grades = list(map(int, input().split()))

count = 0

i = 1

while i < len(grades):
    if grades[i] < 4:
        count = count + 1
        i = i + 1

if 2 * count < len(grades):
    passing = True
else:
    passing = False

print(passing)
```

ERROR: 1 SHOULD BE 0

OUTPUT VARIABLE
grades = list(map(int, input().split()))

count = 0

i = 1

while i < len(grades):
    if grades[i] < 4:
        count = count + 1
        i = i + 1

if 2 * count < len(grades):
    passing = True
else:
    passing = False

print(passing)
grades = list(map(int, input().split()))

count = 0

i = 1

while i < len(grades):
    if grades[i] < 4:
        count = count + 1
        i = i + 1

if 2 * count < len(grades):
    passing = True
else:
    passing = False

print(passing)
Piecewise Unused Input Data Analysis

```
grades = list(map(int, input().split()))  # INPUT VARIABLE

count = 0

i = 1  # ERROR: 1 SHOULD BE 0

while i < len(grades):
    if grades[i] < 4:
        count = count + 1
    i = i + 1

if 2 * count < len(grades):
    passing = True
else:
    passing = False

print(passing)  # OUTPUT VARIABLE
```
```python
grades = list(map(int, input().split()))

count = 0

i = 1

while i < len(grades):
    if grades[i] < 4:
        count = count + 1
    i = i + 1

if 2 * count < len(grades):
    passing = True
else:
    passing = False

print(passing)
```

**Piecewise Unused Input Data Analysis**

- **U**: used
- **N**: not used

- **ERROR:** 1 SHOULD BE 0
grades = list(map(int, input().split()))

count = 0

i = 1

while i < len(grades):
    if grades[i] < 4:
        count = count + 1
    i = i + 1

if 2 * count < len(grades):
    passing = True
else:
    passing = False

print(passing)
grades = list(map(int, input().split()))

count = 0

i = 1

while i < len(grades):
    if grades[i] < 4:
        count = count + 1
        i = i + 1
    else:
        passing = True
else:
    passing = False

print(passing)
grades = list(map(int, input().split()))
count = 0
i = 1
while i < len(grades):
    if grades[i] < 4:
        count = count + 1
        i = i + 1
if 2 * count < len(grades):
    passing = True
else:
    passing = False
print(passing)
grades = list(map(int, input().split()))
count = 0

while i < len(grades):
    if grades[i] < 4:
        count = count + 1
        i = i + 1
    else:
        i = i + 1

if 2 * count < len(grades):
    passing = True
else:
    passing = False

print(passing)
grades = list(map(int, input().split()))

count = 0

i = 1

while i < len(grades):
    if grades[i] < 4:
        count = count + 1
        i = i + 1

    if 2 * count < len(grades):
        passing = True
    else:
        passing = False

print(passing)
Piecewise Unused Input Data Analysis

grades = list(map(int, input().split()))

count = 0

i = 1

while i < len(grades):
    if grades[i] < 4:
        count = count + 1
        i = i + 1

if 2 * count < len(grades):
    passing = True
else:
    passing = False

print(passing)
grades = \textbf{list(map(int, input().split()))}

count = 0

\textbf{input variable}

\textbf{error: 1 should be 0}

\textbf{while } i < \textbf{len(grades)}:

\hspace{1em} \textbf{if} grades[i] < 4:

\hspace{2em} count = count + 1

\hspace{1em} \textbf{else:}

\hspace{2em} \textbf{if} 2 \times \text{count} < \text{len}(\text{grades}):

\hspace{3em} \textbf{passing} = \text{True}

\hspace{1em} \textbf{else:}

\hspace{2em} \textbf{passing} = \text{False}

\textbf{output variable}

\textbf{print}(\text{passing})
grades = list(map(int, input().split()))

count = 0

i = 1

while i < len(grades):
    if grades[i] < 4:
        count = count + 1
    i = i + 1

if 2 * count < len(grades):
    passing = True
else:
    passing = False

print(passing)
grades = list(map(int, input().split()))

count = 0

i = 1

while i < len(grades):
    if grades[i] < 4:
        count = count + 1
        i = i + 1

if 2 * count < len(grades):
    passing = True
else:
    passing = False

print(passing)
Unifying Framework

practical tools
targeting specific programs

algorithmic approaches
to decide program properties

mathematical models
of the program behavior

secure information flow
program slicing
strongly-live variable analysis
Lyra

practical tools
targeting specific programs

algorithmic approaches
to decide program properties

mathematical models
of the program behavior
Possible Applications

Data Provenance

Algorithmic Bias

Data Privacy
practical tools
targeting specific programs

algorithmic approaches
to decide program properties

mathematical models
of the program behavior

secure information flow

program slicing

strongly-live variable analysis

QUESTIONS?