Stress Testing a Literature with Declarative Econometrics*

Vanitha Ragunathan, University of Queensland
Jared Stanfield, University of Oklahoma
Robert Tumarkin†, UNSW Sydney

September 01, 2025

Abstract

We create a method for stress testing an economic literature via a novel declarative econometric language. Critically, the language permits generalizable specification manipulation, thereby enabling us to stress test assumptions and methodology at the literature level. Applying this approach to the M&A literature, we find that within-literature variation in how the dependent variable (cumulative abnormal returns) and control variables (leverage and Tobin's Q) are defined creates significant dispersion (and skewness) in the estimated significance of variables of interest. Taken together, our publicly available approach will allow researchers to succinctly demonstrate the robustness of individual papers and stress test economic literatures. This provides a generalizable path to study the mechanisms behind the uncertainty created by researchers in the evidence-generating process.

JEL Classification: G34

Keywords: Replication, Stress testing

* We would like to thank seminar participants at Curtin University, the University of Melbourne, and the University of Oklahoma.

* Corresponding Author. UNSW Business School, UNSW Sydney, Sydney NSW 2052, Australia. Telephone: +(61) 02 9065 1656. E-mail: r.tumarkin@unsw.edu.au.

1. Introduction

To create a lasting impact, research must be resilient. Replicability, which has been a focus of recent studies, is a necessary first step in establishing this resilience — findings should be replicable if they are to be taken as true. Additionally, research needs to withstand methodological challenges. It is not enough to merely state that others can reproduce results in a body of literature. Findings should be collectively robust against different assumptions, including sample selection criteria, specification design, choice of control variables, variable definitions, and estimation method.

Researchers need a way to stress test a literature. For a given research question, empiricists possess a myriad of choices (researcher degrees of freedom) in the data and evidence generating processes. These degrees of freedom introduce uncertainty into reported empirical estimates, creating what Menkveld et al. (2024) identify as “non-standard errors”. Traditionally, to solve this issue, each empirical paper presents a series of robustness tests that are reported in the body of a paper, in untabulated results, or in internet appendices.

Paper-specific robustness sections assure readers that the paper’s results are resilient to a limited set of altered specifications. However, these sections are necessarily idiosyncratic. A paper’s results are, in general, robust to alternatives presented within the paper. But they may not be robust to different empirical choices from the evidence-generating process, including alternative hypotheses analyzed or different variable definitions used in other papers. Literature-level stress testing (i.e., systematically stress testing each paper that comprises a literature) is currently challenging to execute. This limits our understanding as critical assumptions and standard practices that underlie a literature may never be tested.

The current replication technology is insufficient for stress testing an academic literature. Replication research projects typically employ numerous researchers, sometimes as many as several hundred. Each paper is replicated individually from the bottom up, whereby a researcher specifies the steps to be executed and builds a dataset sequentially, resulting in bespoke data generation and estimation code written for each paper. The fact that researchers take this approach when replicating papers is not surprising. They are simply performing what they do on a single paper for many papers. However, systematic stress testing is virtually impracticable when each paper’s unique base-level code must be modified.

To overcome these challenges, we develop and employ a declarative econometric technique, discussed in detail below, to stress test the merger and acquisition (M&A) literature. This literature provides an ideal setting for stress testing and declarative econometrics. It is mature, having been extensively studied over an extended period, and has established results across numerous papers with similar outcome variables. Papers draw information from frequently used databases, making specification modification more appropriate. Furthermore, although the event study methodology is an accepted framework, critical assumptions vary across papers.

In developing a systematic approach to stress testing, an alternative becomes apparent when one reconciles the significant disparity between how research is formulated and how research is executed. Research is motivated from big-picture issues. This top-down perspective stands in stark contrast to how research is implemented. Data manipulation, using languages such as SQL and those embedded in R, SAS, Stata, and Pandas, is typically written as a bottom-up, step-by-step process.

Programming languages exhibit a similar distinction between top-down and bottom-up approaches to coding. Imperative languages, which work from the bottom up, are the most widely known.[1] In these, each program is crafted step-by-step. However, this is not the only option. Declarative languages, most notably those in the ML and Lisp families, operate in a fundamentally different manner. The programmer works from the top down, defining "what" the program does, with the language itself determining the ordering of bottom-up steps.[2]

In the context of econometrics, the "what" is the empirical specification itself, not the data manipulation process. We provide a declarative language and its implementation for empirical econometric analysis, named *Foghorn*. Using it, researchers need only write top-level empirical specifications. And these specifications consist of just two parts: variable definitions and a description of the estimator. The resulting econometric code is succinct, generally more than an order of magnitude shorter than similar bottom-up code. We present an example highlighting the distinction between declarative and imperative approaches in Section 2.

---

[1] C, C++, Java, JavaScript, Go, Python, and Rust, to name a few, are primarily imperative programming languages.

[2] In later sections, we explain how our declarative econometric language determines the bottom-up data manipulation steps from a top-down specification and variable description. A complete discussion is provided in the technical companion paper (Tumarkin 2025).

Declarative econometrics enables researchers to undertake new types of economic analysis, in particular, stress testing an economic literature. It is nearly impossible to systematically modify analyses using a traditional bottom-up approach, particularly if each paper has bespoke code. *Foghorn*, on the other hand, can internally manipulate top-down specifications and variable definitions. Thus, a stress test reflects the desired underlying change in economic reasoning, and *Foghorn* generates the corresponding modified bottom-up data generation and analysis steps automatically.

Consider, for example, a possible stress test in corporate finance. Many financial variables scale with company size. To eliminate the influence of outliers, researchers often normalize, dividing nominal measures by proxy variables for size. Papers use different size proxy variables (e.g., assets, market capitalization, number of employees, revenues, and sales). Some researchers normalize using an end-of-year value, while others use a start-of-year value. Given the variation across papers, a reasonable stress test is to apply the full range of commonly used normalization techniques to all the papers in a literature.

Implementing such a test by modifying replication code written from the bottom up would be difficult. For a single paper, one would first need to manually identify all the normalized variables. Then, any necessary additional data would need to be incorporated, which would be complicated due to the company size proxy variables coming from various sources and having different timing. Finally, the normalized variables must be generated and tests performed. At the literature level, this would be nearly impracticable given that each paper would require unique, tailor-made modifications. Furthermore, it would be very complicated to make tests scalable, for example, by combining individual stress tests into a composite one.

Declaratively implementing a stress test is straightforward since the language only considers variable definitions. For this example, one writes a stress test that changes the denominator of normalized variables by examining the algebraic representation. Given a set of possible firm size proxy variables, $\mathcal{S}$, a stress test considers any case of division, replacing a size proxy, s, in the denominator with an alternative s'. In other words, the stress testing function $f_{s'}(\cdot)$ is defined as:

$$f_{s'}(x/s) = x/s' \text{ if } s \in \mathcal{S}. \tag{1}$$

The logical implementation of such a function in *Foghorn* does not differ materially from this example. Further, unlike a bottom-up approach, stress testing in *Foghorn* is scalable, with a trivial syntax to combine multiple stress tests into a composite and an efficient implementation to perform a collection of tests.[3] It can easily handle variable definitions with multiple component pieces through its internal recursive data structure. The researcher does not need to enumerate the possible ways variables may be constructed. Instead, the researcher focuses on the case (or cases) that require transformation, and *Foghorn* handles the recursive issues.

We begin with a sample of papers in the literature that analyze cumulative abnormal returns around M&A announcements, selected from four journals with arguably the highest impact factors in academic Finance.[4] For each paper, we identify the key result as indicated by the authors. We then declaratively encode the paper's description of the result's corresponding estimator, variable definitions, and sample selection. Our specification is taken verbatim from the paper; we impose no editorial judgment as to what constitutes the "best" methodology (for example, Kolari, Pynnonen, and Tuncez (2021); Hou, Xue, and Zhang (2020)). This serves as a baseline specification for replication and stress testing in *Foghorn*.

In order to stress tests results from the literature, we must first replicate them. Replication itself is not a precisely defined concept. Achieving results identical to those published may be impossible, for example, due to changes in the source databases. We consider a paper replicated when (i) key variables have point estimates and statistical significance comparable to the published results, (ii) the replicated sample size is similar to the published size, and (iii) most control variables have point estimates similar to those published.

Our results show that the papers in our sample are, in general, replicable. The existing literature presents mixed evidence on the replicability of academic studies. Both Ioannidis (2005) and Chang and Li (2022) document a significant lack of replicability in the science and the economic literatures, respectively. Chang and Li (2022) documents that only 49% of papers could be replicated, even after using the author-provided code and with the assistance of the authors of the paper. In contrast, Jensen, Kelly, and Pedersen (2023) find evidence of replicability in the finance asset pricing literature.

---

[3]*Foghorn* logically analyzes a collection of stress tests for efficient implementation. It first applies the stress tests to a baseline specification and generates a master set of all variables that are needed. These are combined into a single data generating process. It then partitions the master into the appropriate sub-datasets required for each stress test.

[4]These are the Journal of Finance, Journal of Financial Economics, Journal of Financial and Quantitative Studies, and the Review of Financial Studies.

Using *Foghorn*, we find that we can replicate approximately 79% of the papers in our sample from the M&A literature.[5] Our experience, is that replicability does not come easily, which may explain the mixed findings in the literature. Given the limited space allocated to describing the data manipulation process in research papers, authors may elide assumptions, particularly those that qualify as standard practices. *Foghorn* overcomes these issues because it can easily perform an exhaustive search over different standard practice assumptions, allowing us to find a combination that yields replication results similar to those published. Declarative econometrics necessarily makes elided assumptions explicit when writing specifications, enabling *bi-directional transparency*, transparency from both thee top-down and the bottom-up transparency.

As discussed earlier, the M&A literature has several features that make it an appropriate laboratory for stress testing and replication. The papers examining cumulative abnormal returns surrounding acquisition announcements use an event study methodology. However, there is material variation in how each paper implements the study. Cumulative abnormal returns (CARs) serves as the dependent variable in our selected studies, capturing the market-based returns of the announcement to shareholders. Yet, the definition of abnormal returns varies, with 80% of the papers in our replicated sample using a unique construction.[6]

We apply our approach and replicate 15 papers on M&A announcement returns drawn from leading finance journals between 2000 and 2023. These studies share a common event-study framework but differ in data treatments, variable definitions, and methodological assumptions. We focus on each paper's principal CAR-based result, encoding the specification directly from the published description. Replication requires judgment in cases where details are omitted—for example, treatments of missing data, construction of composite controls, or choice of estimation window. *Foghorn* makes these assumptions explicit and allows efficient searches across plausible alternatives.

Using qualitative replication criteria—similar coefficient sign, magnitude, and significance, along with comparable sample sizes—we replicate key findings in 15 papers. While exact numer-

[5]Our inability to replicate a result does not prove that it is not replicable. Rather, it suggests that our interpretation of the empirical approach did not yield similar results. Consequently, we do not list those papers whose results we were unable to replicate.

[6]Papers differ on the time window used to compute an abnormal return around the announcement, whether the return is an excess return over a market index or relative to a market model, and the index (or market model) used.

ical matches are often infeasible due to data updates or unreported assumptions, our results demonstrate that many published findings can be reconstructed within a shared data-processing framework. Results in literature can be generated simultaneously without a need to impose paper-specific methodologies and assumptions to reproduce each result. This highlights the potential for literature-level replication to clarify common practices, expose implicit assumptions, and provide bi-directional transparency across studies.

While replication is fundamentally necessary in the sciences, strict replication represents a relatively low bar. Simply, a replicable result demonstrates that no coding errors were made, but remains silent on the sensitivity of results to a myriad of (ad hoc or well-motivated) assumptions and empirical designs found in the literature. Therefore, we now continue our analysis by stress testing the replicated M&A studies. We do so along two dimensions: the construction of the independent variable, cumulative abnormal returns (CARs), and the construction of commonly used control variables (market-to-book/Tobin's Q and leverage). We choose these because they are commonly utilized across the literature, but also demonstrate significant variability in how they are constructed. Thus, we allow the construction of these variables to vary within the literature (how they are defined within the sample of replicated papers).

Varying CAR windows, expected-return models, and estimation periods produces thousands of alternative specifications. Across more than 4,500 such tests, we find substantial dispersion in t-statistics. This provides evidence of the significant impact that researcher degrees of freedom have on the significance of coefficients consistent with Menkveld et al. (2024). We find similar evidence when varying the definitions of commonly used control variables. Quantifying these results among existing published studies that have already undergone the peer-review process will allow the literature to transparently identify best practices and the impact specific definitions may have on significance.

Importantly, we also document significant skewness in favor of published results. Specifically, we find that the t-statistics on the variables of interest from the replicated (published) specifications in our sample are larger than the t-statistics of the stressed alternatives in over 80% of stress tests. Economically, we observe an average decline of nearly 0.8 in the magnitude of the t-statistics as a result of stress tests. Further, over half of the t-statistics of stress tests are at a lower significance level (i.e. at a 10% significance versus a 1% significance) than the relevant published/replicated t-statistic. Finally, we find that results become insignificant in more than one-third of stress tests. We observe similar results in stress tests on a subsample of papers where t-statistics are less than three (and therefore a decline in significance is potentially more meaningful).

To study whether the above skewness is a function of the simple dispersion due to non-standard errors, we perform similar analysis on the t-statistic of replicated variables compared with their stressed alternatives. However, instead of studying variables of interest, we study control variables. We continue to find a significant dispersion in the significance of control variables from stress tests. Importantly, we observe significantly less skewness amongst the significance of control variables. Given the incentive set given by the publication process for significant results, this finding suggests that published results systematically favor particular CAR definitions.

We observe similar patterns when analyzing alternative definitions of market- to-book and leverage on the significance of variables of interest. These tests produce smaller but still meaningful shifts in significance. Given the indirect nature between the definition of a control variable on the significance of a variable of interest, this is unsurprising relative to changes in the dependent variable. Similarly, while the effects are more modest than for CAR construction, they again tend to benefit the published specification. In addition, we do not observe any skewness when we observe the impact on the significance of other control variables by varying these definitions. Together, these findings show that choices in variable definition—whether for outcomes or controls —can materially affect significance and that published M&A results often occupy the more favorable end of this distribution.

We contribute to the literature in several ways. We demonstrate that declarative econometrics can make a significant contribution to academic knowledge, particularly in terms of research efficiency, replication, transparency, and stress testing. It's important to note that declarative econometrics is not intended to replace existing techniques, nor should it. There will always be a benefit in writing bottom-up data processing and estimation code. Doing so helps researchers build invaluable, in-depth knowledge of the nuances and idiosyncrasies present in the data. However, with it's top down focus, declarative econometrics allows researchers to efficiently present transparent, continuously validated results and focus on the high-value, intellectual considerations. It also significantly reduces the time necessary to implement empirical projects.

These benefits create strong value for empirical researches and research outlets. For example, demonstrating a simple histogram as those presented in this paper allows a researcher to succinctly summarize that their results are not dependent on a particular variables definition. Further, this analysis is transparent and can be easily validated by conference discussants, referees, journal editors, and readers. When a meaningful portion of results are insignificant, it also forces the researcher to justify the choice of a specific methodological definition or choice as the most appropriate to the specific situation that is transparent to others.

We also contribute to the literature by finding a strong level of replicability. However, we also document strong variability in the significance of estimates by stress testing seemingly innocuous assumptions for commonly used variables. This demonstrates some of the source for "non-standard errors" highlighted by Menkveld et al. (2024). Further, we find significant skewness in the literature favoring published specifications. Stress testing a literature is fundamental different than performing robustness tests on an individual paper. While a robustness test typically examines a paper's assumptions that differ from common practice, literature-level stress testing examines the assumptions that underlying common practice and how they may co-vary with other innocuous definitions found in the literature.

2. Illustrative Declarative Econometrics

Declarative econometrics takes a novel approach to empirical research. We begin by analyzing a sample statement to introduce the concept before working through a specification. The examples below are intentionally minimal to highlight the differences from a traditional bottom-up approach. While they may not fully showcase the potential of working declaratively or the variety of tools currently available in *Foghorn*, they still offer valuable insights. We remind readers that, although these examples focus on corporate finance, declarative econometrics is a general framework applicable across all areas of the social sciences. We provide a more formal definition of declarative econometrics in Section 3.

2.1. An Illustrative Declarative Econometric Statement

Assume that a researcher wants to compute a market-to-book ratio for research on financial markets. The ratio requires two variables from the Center for Research in Security Prices (CRSP) monthly stock data, shares outstanding (`Crsp.shrout`) and share price (`Crsp.prc`), both of which are indexed by a permno cross-sectional firm identifier and a trading-date time-series identifier.[7] It also requires the book value of total assets variable (`Comp.at`) as of the fiscal year-end from the Compustat Fundamentals Annual data, which is indexed by a gvkey cross-sectional identifier and a datadate fiscal-year-end time-series identifier. Computing a market-to-book ratio in a bottom-up data manipulation language requires writing code to convert gvkey – datadate assets to a corresponding permno – trading-date asset time series. Then, an algebraic step does the actual calculation. The resulting code emphasizes the steps, rather than the algebraic relationship. It would generally be verbose, requiring many lines of precise table joins to transform the underlying data indices before the final algebraic step that computes the market-to-book ratio.

In a declarative language, the index conversion and calculation may be written in a single step. This calculation in *Foghorn* is coded as

$$\texttt{Crsp.shrout} \times \texttt{Crsp.prc} \ / \ \texttt{reindex}(\texttt{Comp.at}). \tag{2}$$

This top-down calculation emphasizes the underlying economic meaning by using a simple reindex function to tag which panel variable should have its indexes converted. The benefits of such succinct coding syntax grow exponentially with variable complexity.

Foghorn can convert this top-level equation into bottom-up data manipulation code because it knows the indices of each variable (i.e., cross-sectional, time-series, or panel) at the language level, a feature known as static typing in programming language design. In the expression above, *Foghorn* identifies that `Crsp.shrout` and `Crsp.prc` are both indexed by permno-trading-date. Thus, their product is also indexed by permno-trading-date. *Foghorn* also knows `Comp.at` is indexed by gvkey-datadate. Therefore, because `Comp.at` divides `Crsp.shrout` $\times$ `Crsp.prc`, *Foghorn* infers that

[7]The term “indexing” refers to index variables, unique identifiers for individual observations within a dataset. These identifiers may be cross-sectional and/or time-series units. The term “reindexing” describes the procedure of aligning identifiers across different datasets, accurately matching an observation from a source dataset with its corresponding observation in a target dataset.

the `reindex` function must convert `Comp.at` from gvkey-datadate to permno-trading-date using a programming language feature known as type-inference.

2.2. An Illustrative Declarative Econometric Specification

Consider a research project to analyze the effect of entrenched corporate boards on stock returns around acquisition announcements. The study will control for both the acquirer's and the target's Tobin's q. To keep this example concise, we will not include additional controls, time-constant or firm-constant fixed effects, or sample selection criteria. We will also not adjust or cluster standard errors. All these features are available in *Foghorn*, which also allows for organizing empirical datasets by any cross-sectional and/or time-series indices.

Figure 1 shows the complete econometric code for implementing this empirical study in *Foghorn*. For brevity, we have omitted boilerplate statements that import *Foghorn* libraries.

Figure 1: Example Declarative Econometric Specification

```
study = estimate
  [regHdFe| car ~ bcfIndex + acquirerTobinsQ + targetTobinsQ |]

-- Cumulative abnormal returns
car = acquirer $ cumulativeAbnormalReturn (-2, 2)
        `overModel` singleIndexModel (-210, -11) CrspEqualWeighted

-- Entrenchment index (Bebchuk, Cohen, and Ferrell (2009))
bcfIndex = acquirer
  (Risk.cboard + Risk.labylw + Risk.lachtr + Risk.gparachute + Risk.supermajor + Risk.ppill)

-- Firm value
acquirerTobinsQ = acquirer tobinsQ
targetTobinsQ = target tobinsQ
tobinsQ = (Funda.at - Funda.ceq + reindex (Msf.shrout * Msf.prc)) / Funda.at
```

The sample code in Figure 1 differs noticeably from traditional bottom-up data management and econometric estimation techniques. Perhaps most strikingly, it is concise and direct. The code comprises only nine lines of statements; the remaining six lines consist of white space and comments. The specification requires only variable definitions, a functional form between dependent and independent variables, and the estimation technique. This code emphasizes the "what" of this empirical study, providing only the econometric definitions required to describe the specification.

Unlike traditional bottom-up research techniques, the declarative specification is reasonably self-evident.[8] An interested third party does not need to review a step-by-step data manipulation process to understand the econometric methodology. Instead, this declarative specification begins with a top-down statement indicating that this study estimates a single regression model. The model's functional form specifies that observed cumulative abnormal returns are related to a hypothesized linear relationship among the acquiring firm's corporate board entrenchment index, the acquiring firm's market-to-book value ratio, and the target firm's market-to-book value ratio.[9]

The code then defines the variables in the model. The acquirer's cumulative abnormal return (CAR) is calculated over a five-day window, starting two days before and ending two days following the announcement. This calculation utilizes the CRSP Equally Weighted Index as a single index market model, with parameters estimated over a window beginning 210 days before and ending 11 days before the announcement. The Entrenchment Index (Bebchuk, Cohen, and Ferrell 2008) for the acquirer is the total of six indices from RiskMetrics (`Risk`): classified boards (`cboard`), limited ability to amend bylaws (`labylw`), limited ability to amend charter (`lachtr`), golden parachutes (`gparachute`), supermajority requirements (`supermajor`), and poison pills (`ppill`). Finally, the acquirer's and target's Tobin's *q* (lines 10 and 11, respectively) are built upon a general definition in line 12. This uses the book values of assets (`at`) and common equity (`ceq`) from Compustat Fundamentals Annual (`Funda`) data as well as stock price (`prc`) and shares outstanding (`shrout`) from the CRSP Monthly Stock File (`Msf`).

The specification's required data sources, including Compustat, CRSP, RiskMetrics, and SDC, are prominent in the specification.[10] Each of these employs different indexing variables to uniquely identify observations. While verbose data manipulation steps that convert variables across indices

[8]We recognize that the language has some notational idiosyncracies (e.g., the dollar signs and back ticks), which arise from our decision to implement *Foghorn* as an embedded domain specific language. More information is provided in Section 4, with further details in the companion technical paper (Tumarkin 2025).

[9]We have borrowed notation from *R* for estimators as we believe the use of algebraic operators makes the functional form clear.

[10]Unlike Compustat, CRSP, and RiskMetrics which have the explicit prefixes `Comp`, `CRSP`, and Risk, respectively, SDC data is referenced implicity by the `cumulativeAbnormalReturn`, `acquirer`, and `target` functions.

often comprise a substantial portion of code in traditional econometric work, they are noticeably concise in Figure 1. The `acquirer` and `target` functions attribute data to the respective parties of an acquisition.

2.3. Declarative Econometrics and the Bottom-Up Process

High-level declarative code, such as in this section, may seem insufficient for implementing an econometric study, particularly because there does not appear to be any bottom-up steps. Recall that in declarative econometrics, the researcher defines the empirical specification, with the language determining the implementation. Thus, an econometric study is well-defined provided the language has sufficient information to convert the specification into a database construction process. *Foghorn* translates top-down specifications into standard database code, a process known as transpiling.[11] *Foghorn* analyzes the specification, determines the order in which to perform algebraic, aggregating, reindexing, time-shifting, and other calculations, and generates all the necessary database code for the researcher. For example, in Equation 2, *Foghorn* checks if this conversion is allowed (user-defined permissible index conversions through modules), and, upon execution, generates the appropriate bottom-up code in SQL or SAS, for instance.[12] Transpilation converts functions in *Foghorn* into their data manipulation equivalents. As will be discussed later, the design decision to provide transpiling has numerous benefits, particularly in terms of transparency.

We anticipate that many readers will understand the main ideas behind writing top-down specifications, but be curious about the mechanics underlying how declarative code operates. How can a declarative econometric language conduct an econometric test without bottom-up data manipulation steps? How can the specification work without explicitly describing how observations are matched between datasets? Does the brevity of declarative code limit its applicability across

[11]Transpiling is a new concept to econometrics, but is an established technique in programming languages to add static type constraints to non-statically typed languages. For example, JavaScript is a weakly typed language, making it relatively easy for programmers to include bugs in web applications. Languages such as TypeScript and PureScript provide type safe languages for web development. These languages transpile to JavaScript, eliminating many classes of errors. *Foghorn* employs transpilation in a similar fashion, eliminating many types of errors in econometric coding. *Foghorn* has a plug-in architecture for transpiling, allowing it target different data manipulation languages. It currently has a SQL and STATA transpiler. SAS transpiler is under development.

[12]We note that, while it would be labor-intesive to convert the the market-to-book ratio into a panel variable indexed by *gvkey-datadate* in a standard data manipulation language, it is trivial to do so in a declarative language. One simply applyies the reindex function to the CRSP variables instead: $\texttt{reindex}\ (\texttt{shrout} \times \texttt{prc}) /\ \texttt{at}$.

the social sciences? We will address these questions and explore how declarative econometrics can enhance replication, transparency, and stress testing in Section 4. Before doing so, we first develop a formal definition of declarative econometrics.

3. A Declarative Language for Econometrics

A declarative econometric language enables researchers to write high-level, top-down specifications and convert them into data manipulation processes and estimation procedures. For our purposes, an econometric specification, $\mathcal{S}$, fully defines an empirical methodology, including variable definitions, the modeled function form, and the estimation technique. Notably, our definition of a specification does not include the steps required to transform raw data into a dataset used for estimation, which we collectively refer to as an empirical procedure, $\mathcal{P}$.

Formally, a declarative econometric language $\mathcal{L}$ consists of two components. First, the language has a notation system $\mathcal{N}$ that provides syntax and semantics for describing high-level econometric specifications. Second, an implementation $\mathcal{I}$ exists that algorithmically converts a specification into an empirical procedure (i.e., the data processing and estimation procedure):

$$\mathcal{I} : \mathcal{S}^{\mathcal{N}} \longrightarrow \mathcal{P}, \tag{3}$$

where $\mathcal{S}^{\mathcal{N}}$ denotes a specification $\mathcal{S}$ written in notation system $\mathcal{N}$. Thus, a declarative econometric language $\mathcal{L}$ is the pair of the notation and the implementation:

$$\mathcal{L} = \{\mathcal{N}, \mathcal{I}\}.$$

For example, consider an ordinary least squares regression where the researcher wishes to examine how cumulative abnormal stock returns relate to the market capitalization of the acquiring firm. The specification $\mathcal{S}$ consists only of the variable definitions, the linear functional form, the choice of OLS as the estimator, the sample selection criteria, and the choice of standard errors. The corresponding procedure $\mathcal{P}$, by contrast, must define the steps that construct cumulative abnormal returns, market capitalization, and other control variables from raw data. The empirical procedure links observations across datasets, combining the data into a final dataset, and estimating the parameter vector on that dataset.

In standard research, the specification consists of the information described in the research paper. The procedure is the complete implementation code from raw data. Due to commercial and proprietary data restrictions, authors may not be able to provide the whole procedure to other researchers, even in online form. Instead, code may start with pre-processed data to satisfy licensing restrictions.

While the specification and procedure are related, they have traditionally been expressed in fundamentally different formats. Specifications are described qualitatively in a paper's text, whereas procedures are implemented in code, leaving no formal link between the two. As a result, errors in the procedure are not visible in the specification. The absence of a programming language for specifications underlies this disconnect. A declarative econometric language resolves the problem by allowing the researcher to state the specification directly, with the implementation translating this high-level description into the complete empirical procedure.

3.1. Declarative econometrics notation

The notation system $\mathcal{N}$ needs to allow researchers to describe a specification's variables, functional form, and estimation technique. It should form a complete algebra whereby different types of operations may be composed transparently through a uniform syntax. Consider, for example, the computation for the acquirer's Tobin's q from Figure 1. We could combine the acquirer function (line 13) with the Tobin's q formula (line 15), yielding:

```
acquirer $ (Funda.at - Funda.ceq + reindex (Msf.shrout * Msf.prc)) / Funda.at.
```

This single statement exemplifies the concept of a complete algebra, where different types of operations are composed through a uniform syntax, and it is valid *Foghorn* code. This statement instructs the language to reindex the CRSP data (prefixed by `Msf`) to work with Compustat data (prefixed by `Funda`), perform algebraic operations with the reindexed CRSP data and Compustat data, and match the result with the acquiring firm in SDC. We could even aggregate Tobin's q across the acquirer's peer firms or compute lagged annual averages, for example, in a single statement.

This complete algebra is a key feature of a declarative econometric language's notation system, providing succinct, top-down semantics to define variables. Operations such as arithmetic, data aggregation, index conversion, data merging, time transformations, and other necessary data manipulation tasks work in harmony. Whereas existing packages let researchers manipulate data through a series of steps, they isolate different types of operations. Index conversion semantics may differ from those used in data aggregation. And, it is generally not possible to chain index operations of various types together. By contrast, a declarative language provides a way to compose such operations, compactly describing every test variable in terms of raw data.

The notation system also provides semantics to describe estimators. Most software packages already use declarative syntax to define. For example, in SAS, Stata, or R, the researcher uses a command to describe the specification but does not instruct the package on how to perform the required computations. Researchers, for example, generally do not concern themselves with the matrix inversion algorithms needed for most econometric models. A declarative econometric language's notation system should be no different. The estimator description parameterizes the estimation method, such as the technique (e.g., OLS, GMM), dependent variables, explanatory variables, and standard error calculations.

While the estimator description in a declarative language is syntactically similar to that in standard econometrics packages, it is functionally more powerful. A declarative language enforces a *whole-specification* approach to writing econometric tests. As detailed in Section 4, declarative econometrics elevates econometric information from the data to the language. The language's compiler can then reason about a specification logically as a whole, rather than evaluating a specification simply as a sequence of programming steps. Compiler reasoning is one factor behind the validity of the code in Figure 1, even though the functional form appears before the variable definitions. A declarative econometric language validates that a specification is internally consistent before execution. A functional form is not valid unless the variables it requires are defined and have identical indexing. The language ensures that all operations that comprise variable definitions are correct, and it can infer desired data transformations (as described in the introduction). The language's ability to ensure specification integrity has clear benefits for writing econometric tests, in general. However, the power of language-level integrity checking is particularly evident in stress testing.

3.2. Empirical Procedure

The declarative language implementation $\mathcal{I}$ converts the empirical specification $\mathcal{S}^{\mathcal{N}}$ into the empirical procedure $\mathcal{P}$. The empirical procedure can be seen as a mapping from the universe of raw data, $\mathcal{D}_0$, to an estimated parameter vector, $\hat{\theta}$:

$$\mathcal{P} : \mathcal{D}_0 \longrightarrow \hat{\theta}.$$

Without loss of generality, we consider the implementation of empirical specification to consist of two broadly defined steps. First, a data manipulation (generation) process, $\mathcal{M}$, transforms raw data into the processed data required for estimation. Second, an estimation procedure, $\mathcal{E}$, computes the coefficient vector, standard errors, and other statistics of interest from the processed data.

Let the data manipulation process $\mathcal{M}$ have n steps. Notate the data after i steps as $\mathcal{D}_i$. An individual step, $\mathcal{M}_i$, in that data process transforms the data from $\mathcal{D}_{i-1}$ to $\mathcal{D}_i$,

$$\mathcal{M}_i : \mathcal{D}_{i-1} \longrightarrow \mathcal{D}_i.$$

The cumulative process $\mathcal{M} : \mathcal{D}_0 \longrightarrow \mathcal{D}_n$, which converts the raw data into the final data needed for estimation, is the sequential composition of the individual steps:

$$\mathcal{M} = \mathcal{M}_n \circ \mathcal{M}_{n-1} \circ \ldots \circ \mathcal{M}_2 \circ \mathcal{M}_1.$$

We note that mathematical composition associates to the right – the rightmost transformation is applied first, followed by those to its left.

Finally, the estimation procedure $\mathcal{E}$ maps the final data into the coefficient vector:

$$\mathcal{E} : \mathcal{D}_n \longrightarrow \hat{\theta}.$$

Thus, we can view the empirical procedure $\mathcal{P}$ as the composition of the estimation procedure and the data manipulation process:

$$\mathcal{P} = \mathcal{E} \circ \mathcal{M}. \tag{4}$$

3.3. Stress testing

Stress testing an economic literature is a principal innovation from declarative econometrics. Stress testing a literature is fundamental different than performing robustness tests on an individual paper. While a robustness test typically examines a paper's assumptions that differ from common practice, literature-level stress testing examines the assumptions that underlying common practice. A stress test $\mathcal{T}$ may be considered a transformation of a base empirical specification $\mathcal{S}$ to a new one, $\mathcal{S}'$:

$$\mathcal{T} : \mathcal{S} \longrightarrow \mathcal{S}'. \tag{5}$$

Researchers have not yet developed a systematic approach for literature-wide stress testing. Such stress tests must be scalable both in the range of tests they can accommodate and in their ability to apply those tests consistently across specifications. Equally important, they must ensure correctness, regardless of stress-test complexity.

3.3.1. *Stress Testing with Existing Technologies*

With existing technologies, researchers cannot manipulate the specification directly as per Equation 5. Instead, they must work with the data manipulation and estimation procedures that together comprise the empirical method.

A manual approach follows traditional economic practice: the researcher applies a stress test transformation to an original procedure $\mathcal{P}$, derives the stressed procedure $\mathcal{P}'$, and constructs the corresponding data-management process $\mathcal{M}'$ and estimation procedure $\mathcal{E}'$. While feasible for a few stress data-generating and estimation processes, this strategy does not scale; the effort required to stress test a literature increases multiplicatively with both the number of tests and the number of projects, and remains vulnerable to human error.

Alternatively, a researcher may pursue a procedural approach, directly modifying the data-manipulation process and estimation procedures:

$$\mathcal{T}^{\mathcal{E},\mathcal{P}} : \mathcal{E} \circ \mathcal{M} \longrightarrow \mathcal{E}' \circ \mathcal{M}'.$$

This strategy will not be effective across empirical procedures. Procedures are typically implemented as bespoke data-manipulation processes and estimation procedures, with coding choices varying across papers. A modification that applies to one implementation is unlikely to translate to another.

Furthermore, this process is unlikely to succeed even when procedural implementations share steps. Consider two procedures $\mathcal{P}_{\mathcal{A}}$ and $\mathcal{P}_{\mathcal{B}}$, with corresponding data manipulation processes $\mathcal{M}_{\mathcal{A}}$ and $\mathcal{M}_{\mathcal{B}}$. Suppose both data manipulation processes contain a common step $\mathcal{R}$ that appears somewhere in the data manipulation pipelines: $R \coloneqq \mathcal{M}_{\mathcal{A},i} = \mathcal{M}_{\mathcal{B},j}$. Function composition, however, suggests that the role of $\mathcal{R}$ within procedure $\mathcal{A}$ need not match its role with procedure $\mathcal{B}$. Consequently, knowing that a stress test $\mathcal{T}^{\mathcal{E},\mathcal{P}}$ correctly transforms $\mathcal{P}_{\mathcal{A}}$ into $\mathcal{P}_{\mathcal{A}'}$ does not imply that it will do the same for $\mathcal{P}_{\mathcal{B}}$, even if the two procedures share steps:

$$\mathcal{T}^{\mathcal{E},\mathcal{P}} : \mathcal{P}_{\mathcal{A}} \to \mathcal{P}'_{\mathcal{A}} \nRightarrow \mathcal{T}^{\mathcal{E},\mathcal{P}} : \mathcal{P}_{\mathcal{B}} \to \mathcal{P}'_{\mathcal{B}}.$$

Like manual stress testing, the procedural approach is neither scalable nor reliable, underscoring the need for a systematic framework.

3.3.2. Declarative Stress Testing

Declarative econometrics offers an alternative, scalable approach to creating correct stress tests. Recall that in declarative econometrics, a researcher is working with specification descriptions, not the data manipulation process and estimation procedure. A declarative language can permit direct specification manipulation, providing a way to write stress tests in terms of modified specifications directly:

$$\mathcal{T}^{\mathcal{S}} : \mathcal{S} \rightarrow \mathcal{S}'.$$

The ability to transform specifications retains the descriptive style of the declarative econometrics, focusing on variable and estimator definitions, and occurring within the standard language without the need to resort to macro-programming or similar techniques.

Recall that the language implementation converts a declarative stress test into our desired modified data manipulation process and estimation procedure. Thus, an implementation can take the specification that arises from a stress test and determine the correct procedure:

$$\mathcal{I} \circ \mathcal{T}^{\mathcal{S}} : \mathcal{S}^{\mathcal{N}} \rightarrow \mathcal{P}' = \mathcal{E}' \circ \mathcal{M}' \quad \text{for all } \mathcal{S}^{\mathcal{N}}.$$

This approach has numerous benefits. First, it is straightforward to conceptualize stress tests based on specifications. The researcher needs only to consider the econometrics of the test, determining the necessary changes to variable definitions and the estimator required for the stress test. By contrast, a stress test that manipulates bottom-up data processes and estimation procedures is more complicated to write, as one needs to consider the role of each step in the data pipeline.

Second, declarative stress testing emphasizes correctness of transformations, avoiding the function composition and bespoke code pitfalls associated with traditional methods. The whole-specification approach embedded within declarative econometrics means that, not only can the language reason about specifications, but it can also reason about stress test transformations. As a consequence, it can ensure that, given any declarative specification, the stress test transformation will return a valid modified version of the specification. Such transformations are called *total* mapping, also known as *total* function, and the language itself checks to ensure that all stress test transformations meet this criterion.[13]

[13]A stress test may be inherently incompatible in certain empirical settings. For example, a stress that lags explanatory variable by a year would be incompatible with a cross-sectional dataset lacking a time dimension. Such

Finally, scalability is a natural consequence of declarative stress testing. Total functions ensure that a declarative stress test can be applied across specifications. The implementation can transform any stressed specification into the corresponding stressed data manipulation process and estimation procedure. The stress tests themselves are scalable. As transformation functions, stress tests can be combined through functional composition. Thus, it is possible to build a complicated stress test from multiple simple tests. The language itself will guarantee that the derived, stressed specification is valid, and it will provide the corresponding bottom-up data processing process and estimation procedure.

4. Implementing a declarative econometric language: How this works

Traditional econometric code embodies a form of dramatic irony; the researcher has information about the data that the programming language does not. For example, a researcher will know that share prices are represented as floating-point numbers, while ticker symbols are represented as text. However, a statement adding the two (i.e., `price + ticker`) is syntactically valid in SQL, SAS, and other data manipulation languages. The language will only recognize the error when the code is executed against the data. This inability of traditional data manipulation languages to reason about the data before execution significantly inhibits their ability to write expressive, succinct econometric specifications.

Declarative econometrics bridges the knowledge gap between the researcher and the computer by enabling the language to reason about a specification before execution, that is, as the researcher develops the specification. The *Foghorn* source code defines all functions and operations at two levels. The "type level" describes rules that the variables involved in a function must satisfy. These rules can embed logic using cross-sectional indices, time-series indices, data types, and other variable-specific information. The "term level" implements the actual function or operation. For instance, *Foghorn* implements algebraic addition similarly to most programming languages at the term level. However, its type-level rule stipulates that the operands must have identical cross-sectional and/or time-series indices and be of numeric data types (i.e., information that can be added). Thus, adding a numeric share price to a textual ticker symbol is forbidden.

---

incompatibility can be handled elegantly by total functions within a declarative language. Details are provided in the companion paper (Tumarkin 2025).

It is the ability to encode logic about the data that makes declarative econometrics powerful, expressive, and succinct. Returning to Figure 1, verbose data manipulation steps to convert variables from one set of indices to another are noticeably absent. However, while some programming languages achieve syntactical succinctness by ignoring information about the data, this declarative econometric code achieves simplicity for the opposite reason: it can reason about the data. *Foghorn* would not let a researcher combine CRSP data with Compustat data to compute Tobin's q without explicit reindexing. Moreover, the `reindex` function does not require detailed instructions, as the language can determine and validate the necessary conversions.

Declarative econometrics offers a general framework applicable across all social science disciplines. At its core, declarative econometrics elevates econometric information from the data to the language. However, the approach does not achieve this by restricting the datasets with which it can work. *Foghorn*'s core is database-agnostic, defining general rules that validate relationships among specifications and variables. Dataset-specific information enters the language through topic modules. For example, the corporate finance module, *Coficat*, provides information about many finance datasets.[14] Users may write modules targeting specific research areas and extend the language's capabilities.

Foghorn will be publicly available for use by econometric researchers. However, we recognize that other researchers may write new modules for *Foghorn* or create other declarative econometric languages. Therefore, when describing how *Foghorn* operates, we emphasize those features that make a declarative econometric approach viable. A complete technical discussion on the design and implementation of the language is beyond the scope of this paper. Details are provided in the companion paper (Tumarkin 2025).

4.1. Embedded domain-specific language

Instead of writing a declarative language from the scratch, we believe it is best to leverage existing technology with an embedded domain-specific language (EDSL). An EDSL operates within a host language, serving as a dialect that facilitates specific tasks. The EDSL allows a programmer to revert to the host language when necessary. Therefore, a declarative econometric EDSL can offer data

[14] *Coficat* is tortured wordplay for a *co*-rporate *fi*-nance research copy-*cat* tool.

manipulation and estimation tools based on a precise top-down specification while preserving the full capabilities of the host general-purpose programming language.

We implement *Foghorn* as an EDSL within Haskell, a language that has strong support for declarative programming. *Foghorn* is designed for social science researchers to use without requiring knowledge of Haskell. As exemplified by the sample code in Figure 1, most specifications can be written in a syntax similar to existing econometric packages with the addition of algebraic constructs to define variables.[15]

Haskell's many unique features make it a desirable host for a declarative econometric EDSL. These include, but are not limited to, lazy evaluation, general algebraic data types, recursive data types, higher-order functions, abstract type classes, and dependent types. However, a technical discussion of the advantages these features provide for a declarative econometric language is deferred to the companion paper (Tumarkin 2025).

Most notably, for this discussion, Haskell code exists at both the term level and the type level. This separation is a key aspect of Haskell's powerful and expressive type system. Most people are familiar with what Haskell considers the term-level, which, in traditional languages, is where the steps of a function are performed. The type-level in Haskell is a unique environment that can be extended to embed logic. Working at the type-level, one can describe intricate relationships among the inputs to and the output from functions. *Foghorn* uses this to analyze, validate, and infer critical aspects of econometric specifications.

4.2. Encoding econometric logic (Database agnostic reasoning in the core language)

Foghorn is designed to be usable across the social sciences. Thus, it contains a "core" language that emphasizes general econometric logic, lacking information about a specific data set or research area. *Foghorn*'s core provides the building blocks of econometric analysis, including specification design, algebraic calculations, data aggregation, data merging, and other common operations. By

[15] *Foghorn* is an open-source programming language that accepts contributions that identify issues, fix bugs, and expand its scope. While *Foghorn* is intended to be used for econometric research without learning Haskell, understanding the source code of the core language requires intermediate to advanced Haskell. This is not uncommon in programming languages where the language's source code is often far more complicated than programs written in the language. For example, many Python users would have trouble working through the source to Python's most popular implementation, CPython, which is not written in Python but C.

tracking index and data type information, the core can to achieve a level of concision that is not possible when a language lacks such information.

A declarative econometric language like *Foghorn* needs to reason about the data. In programming language design, this ability is most powerfully expressed through type-level logic. In *Foghorn*, each variable is declared with type-level information about its indexing variables i (i.e., cross-sectional and/or time-series indices) and data type d (e.g., integer, floating-point number, text). We notate a variable as `Var i d`, where, for example, CRSP stock share price would be declared at the type-level as `Var (Permno, TradingDate) Float`. Variables carry this type-level information around in the code, with the language tracking and reasoning about types throughout the specification.

All operations and functions in *Foghorn* contain "type signatures" that define rules about the relationships among input variables and output at the type level. Type signatures are declared as

```
Constraints => Input 1 -> Input 2 -> … -> Input N -> Output.
```

Functions may include any number of constraints and inputs, with only the output being mandatory. For example, the type signature for addition, subtraction, and multiplication is written as:

$$\underbrace{\texttt{Numeric d}}_{\text{Constraint}} \texttt{=>} \underbrace{\texttt{Var i d}}_{\text{Input 1}} \texttt{->} \underbrace{\texttt{Var i d}}_{\text{Input 2}} \texttt{->} \underbrace{\texttt{Var i d}}_{\text{Output}}.$$

The type constraint, `Numeric d`, is left of the double-arrow and indicates that the data type `d` must be numerical. On the right-hand side, there are two input variables and an output variable, each of type `Var i d`. *Foghorn* can reason that these operations are valid only when the input share indexing identifiers `i` and a numeric data type `d` is used, and the result is a new variable of the same indexing and data type.

Rules in *Foghorn* are often written using type-level variables (a language feature called parametric polymorphism) instead of specific indices or data types.[16] The above example has two type-level variables, `i` and `d`. Type-level variables ensure that rules are as general as possible, and *Foghorn* can apply them to any dataset. *Foghorn* has a rich, extendable list of data types that enable it to precisely reason about econometric specifications. For example, many kinds of economic data are stored as integers. These include standard industry classification (SIC) codes, fiscal years, calendar years,

[16]Type-level variables are distinct from term-level variables. Term-level variables express values, permitting the language to manipulate values. Type-level variables express types, permitting the language to reason about type information.

many identifiers (e.g., CIK, permno, and gvkey), and actual integer numbers. *Foghorn* separates these into separate types. The function to compute Fama-French industry classifications (Fama and French 1992) will only work with SIC codes. Arithmetic operations are only supported for types where the operations make sense (e.g., addition of SIC codes is not permitted).

Type constraints may be of arbitrary complexity due to Haskell's support for advanced programming at the type level. Yet, *Foghorn* does not require that users worry about type-level programming. *Foghorn* inherits Haskell's ability to infer types. For example, consider the reindexing function which has the type signature:

```
Reindexable i i' => Var i d -> Var i' d
```

The type constraint `Reindexable i i'` defines a type class. A source index variable set `i` is paired with a target one `i'` in `Reindexable` only when a variable indexed by observation identifier `i` can be transformed into one indexed by identifier `i'`. Type inference means that *Foghorn* can contextually determine the target indexing variables from reindexing. For example, assume we have variables `a` and `b` indexed by `i` and `i'`, respectively. Type-inference ensures that the expression `reindex a + b` is fully defined. *Foghorn* knows the source indexing for `a` (each variable carries information about index type), and it can infer the target indexing from `b` on the other side of the addition operand. This expression is validated provided that the index type pair `i` and `i'` are an instance of `Reindexable` (and that the underlying data types permit addition).

4.3. Modularity (Database specific information through topic modules)

Foghorn uses a plug-in module system to target different branches of econometric literature. In *Foghorn*, modules must define the data sets available for a literature and the conversions among variable indices .[17] This approach ensures that a declarative econometric language is not tied down to any specific econometric methodology. For example, one module may link firm financial data to stock return data based on what data was publicly known to the market (i.e., the firm's financial data relevant to a specific market observation is from the most recent data published by the company). Another module may use contemporaneous data (i.e., the firm financial data is linked to any return data that occurs during the corresponding fiscal year). A third module may provide

[17]Common data sets used in finance are already in *Foghorn*. Additional types of data sets, both within Finance and in other areas of the social sciences, will be added to the core distribution over time. However, users are not restricted to the core datasets and may implement their own.

both options, requiring the econometrician to specify the linking method explicitly. A fourth module may not allow such index conversions to occur directly.

A module consists of two pieces: a standardized database and its *Foghorn* code. The database is topic-specific, providing an standard against which empirical specifications may be executed. It contains the datasets commonly used in the research area and tables, or other rules, to link those datasets. A module's database serves as a foundation, it does not contain processed data, leaving calculations to the transpiled code generated by *Foghorn*.

The *Foghorn* code serves as the second piece of the module. The code provides information about the database, such as the data sources, variables, index information, and permissible index transformations. This information is used by the core language when analyzing specifications and transpiling. Modules can provide topic specific functionality. *Foghorn* provides a clean way to compose base-level functions into succinct module-level ones. Haskell is a functional programming language, meaning that functions, not objects, are the primary units of programs. Functions are composed simply by using a "dot" operator (i.e., the mathematical definition of composition $h = f \circ g$ is translated literally yielding `h = f . g`. Thus, it is simple to write module-level functions that address common problems in a specific economic literature by composing the building blocks provided by the core.

4.4. Transpiling

Foghorn translates top-down specifications into standard database code, a process known as transpiling. While new to econometrics, transpilation is an established technique in programming languages to add type-safety, enhance abstraction, and add concision to non-statically-typed languages. For example, there are many web-development languages for type-safe, high-level web development that transpile to Javascript, thereby preventing errors and making it easier to write modular code. *Foghorn* similarly employs transpilation, eliminating many types of mistakes in econometric coding.

Transpilation ensures that a declarative econometric language does not constitute a closed system. Having another tool for bottom-up data manipulation would not achieve our objectives for literature-level stress-testing and bi-directional transparency. There are numerous established methods for manipulating data and estimating econometric models. *Foghorn* utilizes a plug-in architecture for transpiling, allowing it to target various data manipulation languages. Users can pick the transpiler needed for their preferred target language. *Foghorn* currently has a SQL and STATA transpiler, with a SAS transpiler under development.

Foghorn uses a simple approach to transpilation. *Foghorn* analyzes a specification and determines the basis set of variables underlying the required estimation panel. This basis set includes final variables used in estimation, intermediate variables needed for calculations, and source variables. It then categorizes the variables into two groups based on whether the variable has been computed. Source variables are placed into the computed group; intermediate variables and final variables are placed into the uncomputed group. For each uncomputed variable, it checks if all its basis variables are in the computed set. If so, that variable is calculated and placed into the computed group. On iteration, all variables are eventually moved from the uncomputed group to the computed group. In summary, *foghorn* analyzes the specification, determines the order in which to perform algebraic, aggregating, reindexing, time-shifting, and other calculations, and generates all the necessary database code for the researcher. In other words, transpilation converts functions in *Foghorn* into their data manipulation equivalents.

Transpilation provides several benefits. The most clear advantage of transpilation is leveraging existing tools. Thus, a declarative approach can incorporate any data process or econometric estimator from any software package. One only needs to define a declarative syntax for a new process or estimator and then write a corresponding transpiler. Declarative econometrics is unique in that it defines the complete estimator at the specification level. By creating a direct link between specification and implementation, and then allowing researchers to manipulate specifications, new types of economic research become possible.

Transpilation greatly improves econometric transparency. Data manipulation languages are generally verbose, requiring hundreds if not thousands of lines of code to implement a standard economic paper. A person reviewing such code may suffer from information overload, failing to identify critical assumptions. A declarative specification, on the other hand, is concise and emphasizes the logic of any empirical study. Key assumptions form the basis of the declarative code, with implementation details provided in the transpiled, step-by-step data manipulation code.

For example, *Foghorn* has an SQL transpiler that exports a specification into the step-by-step data generating-process.

4.5. Whole-specification coding (Benefits and disadvantages)

Declarative econometrics promotes a whole-specification approach to econometrics. As the language can reason about a specification through type-level logic, the estimator definition is intertwined with the variable definitions. There are several benefits of an approach where an empirical researcher can focus on specification design instead of implementation. Declarative econometrics makes key assumptions explicit, improving empirical clarity and transparency. With this,an empirical researcher can focus on specification design instead of implementation

Declarative econometrics is not intended to, nor should it, replace standard techniques. Datasets are idiosyncratic and intimate knowledge is required for convincing empirical research. Thus, there are significant advantages to researchers working directly with data from the bottom-up. However, declarative econometrics makes new types of literature-level analysis possible, improving our critical understanding of standard practices through stress testing and replication.

Finally, we note that declarative econometrics has some disadvantages. *Foghorn* requires modules for specific topics. Creating a module requires building a standard database for the topic and encoding the logic into a *Foghorn* library. Thus, declarative econometrics is best suited to established topics where the upfront cost of writing a module can be amortized across paper replication, stress testing, and new research questions. For a unique empirical question, standard bottom-up techniques are better suited.

Moreover, declarative econometrics is not intrinsically designed to be efficient. *Foghorn* applies sample selection as a last step. As a result, the transpiled code will perform computations over an entire source dataset as it implements a study. It will compute variables that do not ultimately meet sample selection criteria. This inefficiency is by design. Computing variables over an entire dataset guarantees that the transpiled code will execute without crashing and generate the correct final sample for estimation.

Foghorn is just one possible implementation of a declarative econometric language. Other notation systems and implementations are possible. However, we believe the features highlighted above are critical to enabling new types of research, particularly in the areas of stress testing.

Further details about the notation system and implementation are in the technical companion paper (Tumarkin 2025).

5. Replicating the M&A Literature

We employ our declarative econometric technique to stress test the merger and acquisition (M&A) literature. M&A provides an ideal setting for stress testing, given the importance of M&A to corporations. The literature is mature, having been extensively studied over an extended period, and has established results across numerous papers. Papers draw information from a common set of databases, and, although the event study methodology is an accepted framework, critical assumptions vary across papers.

5.1. Sample selection

Our sample selection with papers published in the four journals with arguably the highest impact factors in academic Finance: *The Journal of Finance*, the *Journal of Financial Economics*, *The Review of Financial Studies*, and the *Journal of Financial and Quantitative Analysis*. We identify those papers that analyze returns around M&A announcements, yielding 207 candidates for replication published between 1983 and 2023. The early 1980s start of this period coincides with the nascent academic empirical research into M&A documented by (Mulherin, Netter, and Poulsen 2017). We eliminate studies with purely theoretical contributions and those whose analysis is primarily descriptive.

The inclusion criteria derive from both our replication and stress testing objectives. We first impose a data requirement; we must have access to a paper's data to replicate its results. The candidate paper must use publicly available data, such as that from the Bureau of Labor Statistics and academic websites, or commercially available data to which we have access. As a result, we restrict our sample to papers that use the following commercial data providers: S&P Compustat, Center for Research in Security Prices, Refinitiv Thomson, MSCI RiskMetrics, and SDC Platinum.

Stress testing at the literature-level is the examination of assumptions that underlie common practice. Therefore, we select those papers that exemplify typical empirical specification design and methodology. We limit our sample to those papers where (1) the benchmark analysis, as stated in each paper, utilizes Cumulative Abnormal Returns (CARs) as the dependent variable and (2) the primary estimation technique involves ordinary least squares or panel regressions. As we cannot stress test null results, we examine only those benchmark analyses that find a statistically significant result with a coefficient in the direction hypothesized.

Finally, we have practical considerations that eliminate some papers from our sample. The paper must describe the methodology well enough for us to replicate the key findings. In some cases, the paper may not adequately define a variable, requiring that we make assumptions to replicate the result. Examples include financial variable definitions and industry classification levels. In such cases, we revert to definitions used by other authors in the literature. These replication assumptions are documented in the online replication code.

After applying these criteria, our sample consists of 47 candidate papers. This dataset requirement restricts the candidate studies to those that examine M&A within the United States. It also eliminates those studies that utilize proprietary or hand-collected datasets in their baseline findings. The earliest papers in our sample use hand-collected data from the Wall Street Journal, Grimm's Mergerstat Review, and other sources. Thus, our candidate papers begin with those published after 2000, when data sources became more standardized.

5.2. Defining Replication Success

Academics currently lack consensus on what defines successful replication. We consider replication to be the recreation of published results starting from raw data using a process derived from the description in the published article. This process excludes code provided by the author, as independent verification is necessary to ensure the accuracy of author-provided code.

It is probably not possible to replicate papers exactly. Data providers update information, often restating historic data to improve accuracy (Ljungqvist, Malloy, and Marston 2009). Lyle, Siano, and Yohn (2025) find that Compustat's periodic "standardizations significantly alter key financial figures such as sales and earnings, among many others, leading to material differences in research findings." Researchers also may elide essential steps in their data manipulation process, making precise replication difficult. For example, authors generally do not precisely describe the process to match observations across databases, especially with text-based matching. Authors may also omit their process for dealing with missing financial data, such as research and development expenses. Lyle, Siano, and Yohn (2025) assert that "precise replication of prior studies using common Compustat products is nearly impossible."

Reproduction, as opposed to replication, is the independent execution of a study performed using data and code provided by the authors. Reproduction should be easier than replication, yet researchers have experienced difficulty reproducing results. Chang and Li (2022) attempt to reproduce 67 macroeconomic papers. Defining success as one that produces the "key qualitative results of the paper," they can reproduce 33% of the sample without contacting the authors. Fišar et al. (2024) assess the reproducability of nearly 500 articles in *Management Science*. A "fully reproduced" paper is one in which the reproduction generates the exact numerical results as the published article. A "largely reproduced with minor issues" paper is one in which the reproduction has small differences from the published article. They find that only 45% of articles that voluntarily provided data and code were reproducible before the journal's 2019 institution of a data and code disclosure policy.

We use similar qualitative criteria for replication. We consider a paper replicated when (i) the replicated coefficient on the key economic variable examined in the paper has the same sign with similar magnitude and statistical significance to the published result, and (ii) the replicated and the published samples have a similar number of observations. While many of the control variables are similar to those in published results, we do not concern ourselves with discrepancies due to the issues highlighted above.

5.3. Replication Process

We replicate within *foghorn*, focusing on a single result examining cumulative abnormal returns in the target paper. The specifications selected for replication are typically identified by the author as a key result supporting the paper's hypotheses. We use a single, principal result to keep the scope of our replication manageable and to ensure that stress testing is comparable across papers.

We use the empirical methodology and assumptions as described by the authors to reimplement a paper. In some cases, a paper may not provide sufficient detail. Common issues include the handling of missing data, variable definitions, and sample selection. For example, many well known financial variables are missing from Compustat (e.g., research and development expenses (Koh and Reeb 2015)), and the authors may not indicate if they drop observations or replacing missing values with zero or industry averages. In other cases, paper may not provide the individual variables used to create a composite financial control (e.g., the components used to create a total debt control variable). Papers may also not include the market index or model estimation window used when computing cumulative abnormal returns. Papers may also cite other research as a methodology source without clarifying whether sample selection and other decisions from the source apply.

Fortunately, these elided assumptions generally have precedent in the literature. We use common missing variable treatments, common variable definitions, sample selection, and other criteria as necessary, selecting the assumptions that yield the replication closest to the published result. Declarative econometrics is very helpful in efficiently systematizing this search process. For example, acquirer market capitalization is a common control in the M&A literature and may be sourced from Compustat, CRSP, and SDC in our sample of papers. If not specified, we can quickly switch the source for market capitalization without having to worry about integrating the data into the overall the data management process.[18]

5.4. Results

Table 1 summarizes the results for the papers that were successfully replicated. For each study, it lists the key economic variable of interest and identifies the table and column of the replicated specification. The table reports both the published and replicated coefficients, with significance levels indicated by stars at the 1%, 5%, and 10% levels, and compares the number of observations in the published and replicated tests.

Because we replicate a number of papers, we also provide a separate section, "Replication Tables," located after the standard tables, that presents direct comparisons of all explanatory variables between the published and replicated results. Within each table, coefficients are reported in the same order as tabulated in the original publication, with the primary economic variable highlighted in bold. We also include t-statistics, p-values, or standard errors as appropriate, consistent with the original study.

Finally, many of these papers employ fixed-effect indicator variables, which require omitting one unspecified category to estimate a constant. Consequently, published and replicated constants are not directly comparable, and we do not report constants even when they appear in the original tables.

[18]In *foghorn*, the `acquirer` function links source data to deal observations. Thus, we can simply slot in `acquirer Sdc.market_cap`, `acquirer (Crsp.prc * Crsp.shrout)`, and `acquirer (Comp.prccF * Comp.csho)` and the transpiler creates the correct empirical procedure across database sources.

The findings in Table 1 suggest that key results in published papers are generally replicable. In 16 papers of the 20 papers we have attempted to date, we generate coefficients on key economic variables with comparable economic and statistical magnitudes.[19] These replications suggest that important findings in the literature exist in a common data-processing framework. Thus, the results can be generated simultaneously; researchers do not need to impose paper-specific methodologies and assumptions to reproduce each result.

We do not tabulate papers for which replication was unsuccessful. Failure to replicate does not imply that the original results are not replicable. By applying a declarative approach to economics, we aim to make all assumptions explicit, but the possibility of misinterpreting a paper's methodology remains.

Literature-level replications can significantly enhance transparency. Beyond offering bi-directional transparency for individual papers, generating results within a shared data-processing framework clarifies the empirical practices and assumptions applied across the literature. We now assess the robustness of these results to variations in standard practice through stress testing.

6. Stress Testing the M&A Literature

We now perform two types of within-literature stress tests on the replicated results presented in the previous section. These tests examine the robustness of each replicated paper's coefficient of interest by varying definitions of (1) independent variable definitions and (2) control variables common across most of the papers. This will allow us to understand whether (1) researcher degrees of freedom involved in the choice of an often seemingly innocuous variable definition or sample specification leads to a significant variation in significance of coefficient estimates and (2) whether this variation tends to favor the published specification (i.e. whether reported/replicated results tend to be more significant than their within-literature stress tests).

For all stress tests, we study how the significance (t-statistics) of the variables of interest in each paper vary across these stress tests. We define our primary measure, *tstat_diff* for each stress test as the following:

$$tstat\_diff_{i,j,k} = \begin{cases} base\_tstat_{i,j} - stress\_tstat_{i,j,k} & \text{if} \quad base\_tstat_i > 0 \\ stress\_tstat_{i,j,k} - base\_tstat_{i,j} & \text{if} \quad base\_tstat_i < 0 \end{cases}$$

[19]We continue to work on replication and expand our sample of papers.

where $base\_tstat_{i,j}$ is the replicated t-statistic for variable *i* in paper specification *j*, and $stress\_tstat_{i,j,k}$ is the t-statistic for variable *i* in paper specification *j* and stress *k*. *tstat_diff* will be positive when the magnitude of the replicated t-statistic is greater than the magnitude of the t-statistic of the same variable in the stress test (i.e. the replicated t-statistic is more significant than the stress test's t-statistic). *tstat_diff* less than zero can be thought of as conservative (the alternative definition in the stress test produces a larger, or more significant t-statistic than the replicated t-statistic of the specification found in the published paper) and *tstat_diff* greater than zero can be thought of as aggressive (the replicated t-statistic of the reported specification in the published paper produces a larger, more significant t-statistic than the stress test).

6.1. Dependent variable stress testing

To perform stress testing on independent variable definitions we rely on the chosen sample of papers outlined in the section above and allow the definition and estimation of CARs in robustness tests to vary along these lines This variation comes through three categories. First, we vary the size of the CAR announcement window for each test along the following dimensions relative to the announcement day: [−1, +1], [−2, +2], [−3, +3], [−5, +5], [−1, +5], [−5, +1], [0, +5], and [−5, 0]. Second, we vary how abnormal returns are estimated: subtracting the expected return from the daily raw return utilizing a single-index model (SIM) with equally weighted returns, an SIM with value weighted returns, a Fama-French 3-factor model (FF3), a Fama-French 4-factor model (FF4), and the equally weighted (OEW) and value weighted (OVW) market return (i.e. assuming the beta of the single-index model is equal to one following Brown and Warner (1980)). Third, we vary the expected return estimation period (for tests estimating the market return using the SIM, FF3, and FF4) along the following daily windows relative to the announcement day: [−205, −6], [−210, −11], [−245, −45], [−252, −20], [−272, −20], [−300, −91], and [−370, −253]. These variations generate 238 to 336 stress tests per replicated paper, depending on the specification.

6.1.1. Impact on the significance of variables of interest

We complete the stress tests described in Section 6.1 and compute the *tstat_diff* for each stress test on each paper specification variable of interest. We focus on baseline replicated results from our sample of published papers where the coefficient on the variable of interest was predicted to be significant. From our sample, this yields 15 specifications and 4,543 stress tests for variables of interest. We report the results of these tests in Figure 1 and Table 7.

As can be seen in Figure 1, we document a significant dispersion of t-statistics by varying how the dependent variable (*CAR*) is defined. As a reminder, a positive *tstat_diff* indicates the reported/ replicated t-statistic is larger in magnitude than the corresponding t-statistic for a given stress test. As seen in Table 7, Panel A, the standard deviation of *tstat_diff* is 1.05. This shows that that the definition of an independent variable can introduce significant variation in the significance of coefficients on independent variables of interest. As can be seen in Figure 1, we also find that this distribution is significantly skewed in favor of the published specification. As reported in Panel A, of Table 7, the reported t- statistic is 0.67 larger than the median stress test t-statistic (i.e. this is consistent with the reported t-statistic being a significant 2.20, while the stress test t-statistic is an insignificant 1.53). Specifically, it shows that slight variations (found in the empirical literature) in the definitions of the CARs creates significant variability in significance of variables of interest. This evidence is consistent with the exploratory findings of Menkveld et al. (2024).

As reported in Table 7, Panel C, we find the replicated/reported t-statistic is larger in magnitude than the stress test t-statistic (*tstat_diff* was positive) in 80% of stress tests (3,642 of 4,543). As further reported in the panel, 53% of stress tests result in a drop in significance (i.e. moving from a t-statistic indicating a 1% significance 2.4 to a significance of less than 2.326, or 5% significance of lower) and 37% of stress tests lose significance (a t-statistic of less than 1.65). Finally, only 5% of stress tests increase in significance.

It is possible that these findings are centered in highly significant results where the dispersion will not impact the overall significance of the results. For example, the fact that a t-statistic is 1.0 higher in a reported/replicated result relative to a stress test is less meaningful when each are above 5.0. The evidence reported in Table 7, Panel C does not suggest this is the case, given that we find 53% of stress tests and 37% of stress result in a drop and loss of significance, respectively. However, in Panel B of Figure 1, we restrict our analysis to replicated t-statistics that are less than 3 (where a reduction of 1.05 would reduce the significance of an estimate from 1% significance to 10% significance at best. As seen in the Panel B of the figure, we find similar evidence of dispersion and loss of significance in stress tests where the baseline reported t-statistic is less than 3. Table 7, Panels A and B, reports the summary statistics of *tstat_diff* in this subsample, with a median of 0.56, a mean of 0.60, and standard deviation of 0.76. Given the proximity to cutoffs for significance we see an even greater degree of less significant stress tests in Panel C. While 78% of stress tests are smaller in magnitude than the replicated/reported t-statistic, 64% of stress tests dropped significance, 46% of stress tests loss significance, and still only 6% of stress tests increased in significance. Thus, it does not appear that our findings are only found among highly significant results.

In untabulated robustness tests, we do not find that these results are driven by a specific stress test or specific type of stress test. We find consistent evidence when we drop any specific stress test (i.e. no stress tests have a $[-5, 0]$ announcement window) or stress test type (i.e. do not allow stress test CAR windows to vary but allow the estimation period or method to vary).

6.1.2. Impact on the significance of control variables

In Figure 1 and Table 7, we document that the significance of variables of interest are sensitive to the variation of how CARs are defined within the literature, and that this variation tends to benefit the reported specifications. However, the latter finding may be the result of our methodology. To ensure that this is not the case, we perform the same procedure for control variables. Specifically, using the same stress tests, we calculate *tstat_diff* for control variables rather than variables of interest. We report these results in Figure 2 and Table 10.

As seen in Figure 2 and Panels A and B of Table 10, we still observe a significant amount of variation in the t-statistics of control variables in stress tests relative to the replicated t-statistics. As seen in Table 10, Panel A, the standard deviation of *tstat_diff* continues to be large (1.39) and the skewness (1.35 vs. 1.26) and kurtosis (16.59 vs. 7.80) are even larger. This supports the notion that the choice of definition of the dependent variable can introduce significance variation in the significance of estimated coefficients on all variables). However, we no longer observe the degree of aggressiveness with control variables that we observed with variables of interest. As reported in Panel B of Table 10, the median difference between a replicated/reported t-statistic of a control variable and the corresponding t-statistic from a stress test is 0.22 (relative to 0.67 for variables of interest). This can be seen in greater detail in Panel B. We find that 61% of stress tests of control variables (versus 80% of variables of interest) were lower than the replicated/reported t-statistic. Only 16% of stress tests result in a drop in significance (relative to 53% of stress tests for variables of interest) and 12% of stress tests lose significance (relative to 37% of stress tests for variables of interest). Finally, 11% of stress tests for control variables increase in significance (relative to 5% of stress tests for variables of interest).

A t-test comparing the mean *tstat_diff* between variables of interest (0.775) and control variables (0.336) is highly significant (*t-statistic* of -20.720) and as reported in Table 13, a two-sample Kolmogorov-Smirnov test for the equality of distributions shows that the distribution for *tstat_diff* is significantly smaller for the control variables than for variables of interest. Taken together, these results document evidence of significant sensitivity of variables of interest to within-literature variation in CAR definitions. We document that t-statistics are significantly more skewed towards significance in the published results relative to corresponding stress tests, where incentives for significant results in publication are the strongest. We do not observe this same behavior (in terms of skewness favoring published results) in the significance of control variable estimates, where these publication incentives are less severe.

We have reported stress tests thus far in aggregate across all studies. However, a natural question arises how much variation occurs within each study. To report this variation, we randomly order the individual paper specifications and report histograms of *tstat_diff* for each specification individually in Figure 3. As can be seen by the figure, some paper specifications are significantly skewed right, while others are rather centered around zero. Importantly, we observe significant variation in t-statistics for all specifications. Thus, while the selection of dependent variables tends to favor the published version of specifications more in some specifications than others, all exhibit significant variation and suggests that our evidence is not being driven by one outlying paper-specification. We now turn to studying whether and how the definitions of independent control variables materially impacts the significance of variables of interest.

6.2. Independent (control) variable stress testing

To perform stress testing on independent variable definitions we rely on the chosen sample of papers outlined in the section above and allow the definition and estimation of two controls common to the majority of the published studies in our sample: (1) market-to-book (Tobin's Q) and

(2) leverage. Similar to our analysis in Section 6.1, we vary the definition of these control variables within the literature and study the impact of varying these definitions has on the significance of the variables of interest. We utilize the nine alternative definitions of market-to-book provided, along with references, in Table 5. These are definitions are used within the literature as controls from both acquirers and targets. We also utilize the nine definitions of leverage provided, along with references to the appropriate literature, in Table 6. We allow these variables to vary for those specifications in our sample in which they appear. These variations generate 80 to 96 stress tests per specification, dependent on the specification.

6.2.1. Independent (control) variable stress test results

We perform similar stress tests to those found in Section 6.1, but by allowing the definitions of market-to-book (Tobin's Q) and leverage to vary and study how these variations impact the significance of variables of interest. Unlike varying the dependent variable, these results should not directly influence the coefficient estimate and significance of the variable of interest, and will only do so indirectly through its correlation with the variable of interest. From our sample, 12 of the 15 specifications contain either market-to-book or leverage, yielding 1,136 stress tests for variables of interest. We report the results of these tests in Figure 4 and Table 14. As can be seen in Figure 4, while we continue to find dispersion in *tstat_diff* there is much larger clustering around zero. This can be seen in Table 14, Panels A and B. For stress tests of control variables, the mean, median, and standard deviation of *tstat_diff* is smaller than what we observed in the independent variable stress tests, at 0.13, 0.03, and 0.53, respectively. However, we see even higher skewness and kurtosis, 2.34 and 11.69, respectively.

Importantly, we continue to see a slight skewness favoring the published specifications in Table 14, Panel C. Specifically, 63% of stress tests were lower than the reported/replicated t-statistic. 19% of stress tests resulted in a drop in significance, 11% resulted in a loss in significance, and only 3% resulted in an increase in significance. These results are roughly consistent when we restrict the sample to those reported/replicated results with t-statistics less than 3, as reported in Panel B of Figure 4 and Table 14. In these cases we see roughly similar summary statistics, and slightly higher proportion of stress tests leading to a drop in significance (25%) and a loss in significance (14%).

To fully ascertain whether these results may be an artifact of our methodology, we again compare the results to the variation in t-statistics for control variables. Importantly, since we are varying control market-to-book and leverage, we exclude these variables from this sample when they are varied (to avoid creating a direct effect on the t-statistics). We present these results in Figure 3 and Table 17.

As seen in the figure, the clustering of stress tests with a *tstat_diff* near zero is even higher. Table 17, Panel A shows that the average *tstat_diff* for control variables (excluding market-to-book or leverage when these are allowed to vary in a given stress test) is 0.04 (compared with 0.13). This difference is similarly significant with a t-statistic of -3.45 and a p-value of 0.0006. Further, as reported in Table 20, a two-sample Kolmogorov-Smirnov test for the equality of distributions shows that the distribution for *tstat_diff* is significantly smaller for the control variables than for variables of interest when we vary market-to-book and/or leverage definitions.

Taken together, these findings suggest that, while variation in control variables does not have the same size of impact on the significance of variables of interest, it still generates a meaningful effect. This is unsurprising given its indirect influence on the estimation of the coefficients of interest. However, again we find evidence that the definitions chosen for control variables tend to favor the significance of the variable of interest in published results.

7. Conclusion

This paper develops and applies a declarative econometric framework to the task of stress testing an established literature. By distinguishing between empirical specifications and empirical procedures, and by implementing the *Foghorn* language, we demonstrate that empirical research can be encoded at a high level of abstraction and transpiled into reproducible procedures. This approach yields several benefits. First, it facilitates replication by requiring specifications to be written in a transparent, internally consistent manner. Second, it enables bi-directional transparency, making explicit both the top-down specification and the bottom-up implementation. Third, and most importantly, it allows for systematic stress testing of entire literatures—something that is infeasible with traditional, paper-specific code.

Applying this framework to the mergers and acquisitions literature, we show that most published results are replicable once implicit assumptions are made explicit and that within-literature stress testing reveals significant sensitivity to specification choices. Reported results are often more statistically significant than stressed alternatives, suggesting that researcher discretion plays a meaningful role in shaping published findings. At the same time, the framework highlights that many conclusions remain robust under alternative specifications, reinforcing the value of accumulated knowledge in this field.

More broadly, our results illustrate how declarative econometrics can increase efficiency, replicability, and transparency across empirical research. While not intended to replace traditional coding practices, declarative methods complement them by providing a scalable and systematic way to encode, replicate, and stress test empirical analyses. The promise of this approach extends beyond M&A research: once modules are developed, it can be applied to other areas of finance and the social sciences more generally. In doing so, declarative econometrics provides a path toward more resilient empirical literatures.

References

Bates, Thomas W., and Michael L. Lemmon. 2003. "Breaking up is hard to do? An analysis of termination fee provisions and merger outcomes". *Journal of Financial Economics* 69 (3): 469–504.

Bebchuk, Lucian, Alma Cohen, and Allen Ferrell. 2008. "What Matters in Corporate Governance?". *Review of Financial Studies* 22 (2): 783–827.

Becher, David A, Thomas P Griffin, and Greg Nini. 2021. "Creditor Control of Corporate Acquisitions". *The Review of Financial Studies* 35 (4): 1897–1932.

Brown, Stephen J, and Jerold B Warner. 1980. "Measuring security price performance". *Journal of Financial Economics* 8 (3): 205–58.

Burch, Timothy R. 2001. "Locking out rival bidders: The use of lockup options in corporate mergers". *Journal of Financial Economics* 60 (1): 103–41.

Chang, Andrew C., and Phillip Li. 2022. "Is Economics Research Replicable? Sixty Published Papers From Thirteen Journals Say "Often Not". *Critical Finance Review* 11 (1): 185–206.

Fama, Eugene F, and Kenneth R French. 1992. "The Cross-Section of Expected Stock Returns". *The Journal of Finance* 47 (2): 427.

Fišar, Miloš, Ben Greiner, Christoph Huber, Elena Katok, and Ali I Ozkes. 2024. "Reproducibility in Management Science". *Management Science* 70 (3): 1343–56.

Fuller, Kathleen, Jeffry Netter, and Mike Stegemoller. 2002. "What Do Returns to Acquiring Firms Tell Us? Evidence from Firms That Make Many Acquisitions". *The Journal of Finance* 57 (4): 1763–93.

Golubov, Andrey, Alfred Yawson, and Huizhong Zhang. 2015. "Extraordinary acquirers". *Journal of Financial Economics* 116 (2): 314–30.

Gorton, Gary, Matthias Kahl, and Richard J. Rosen. 2009. "Eat or Be Eaten: A Theory of Mergers and Firm Size". *The Journal of Finance* 64 (3): 1291–1344.

Harford, Jarrad, Mark Humphery-Jenner, and Ronan Powell. 2012. "The sources of value destruction in acquisitions by entrenched managers". *Journal of Financial Economics* 106 (3): 247–61.

Hou, Kewei, Chen Xue, and Lu Zhang. 2020. "Replicating Anomalies". *The Review of Financial Studies* 33 (5): 2019–2133.

Ioannidis, John P. A. 2005. "Why Most Published Research Findings Are False". *Plos Medicine* 2 (8): e124.

Jensen, Theis Ingerslev, Bryan Kelly, and Lasse Heje Pedersen. 2023. “Is There a Replication Crisis in Finance?”. *The Journal of Finance* 78 (5): 2465–2518.

John, Kose, Anzhela Knyazeva, and Diana Knyazeva. 2015. “Employee rights and acquisitions”. *Journal of Financial Economics* 118 (1): 49–69.

Koh, Ping-Sheng, and David M. Reeb. 2015. “Missing R&D”. *Journal of Accounting and Economics* 60 (1): 73–94.

Kolari, James W., Seppo Pynnonen, and Ahmet M. Tuncez. 2021. “Further evidence on long-run abnormal returns after corporate events”. *The Quarterly Review of Economics and Finance* 81:421–39.

Li, Kai, Buhui Qiu, and Rui Shen. 2018. “Organization Capital and Mergers and Acquisitions”. *Journal of Financial and Quantitative Analysis* 53 (4): 1871–1909.

Ljungqvist, Alexander, Christopher Malloy, and Felicia Marston. 2009. “Rewriting History”. *The Journal of Finance* 64 (4): 1935–60.

Lyle, Matthew R., Federico Siano, and Teri Lombardi Yohn. 2025. “Re-Adjusted Financial Statement Data: Challenges in Replicating Research”. *SSRN Electronic Journal*.

Ma, Qingzhong, David A. Whidbee, and Wei Zhang. 2019. “Acquirer reference prices and acquisition performance”. *Journal of Financial Economics* 132 (1): 175–99.

Masulis, Ronald W., Cong Wang, and Fei Xie. 2007. “Corporate Governance and Acquirer Returns”. *The Journal of Finance* 62 (4): 1851–89.

Menkveld, Albert J., Anna Dreber, Felix Holzmeister, Juergen Huber, Magnus Johannesson, Michael Kirchler, Sebastian NeusüSS, Michael Razen, and Utz Weitzel. 2024. “Nonstandard Errors”. *The Journal of Finance* 79 (3): 2339–90.

Moeller, Sara B, Frederik P Schlingemann, and René M Stulz. 2004. “Firm size and the gains from acquisitions”. *Journal of Financial Economics* 73 (2): 201–28.

Mulherin, J. Harold, Jeffry M. Netter, and Annette B. Poulsen. 2017. “The Evidence on Mergers and Acquisitions: A Historical and Modern Report”. Edited by Benjamin E.Hermalin and Michael S. Weisbach. *The Handbook of the Economics of Corporate Governance*.

Nguyen, Nam H., and Hieu V. Phan. 2017. “Policy Uncertainty and Mergers and Acquisitions”. *Journal of Financial and Quantitative Analysis* 52 (2): 613–44.

Roosenboom, Peter, Frederik P Schlingemann, and Manuel Vasconcelos. 2013. “Does Stock Liquidity Affect Incentives to Monitor? Evidence from Corporate Takeovers”. *Review of Financial Studies* 27 (8): 2392–2433.

Tumarkin, Robert. 2025. “Declarative Econometrics: From Data to Beta”. *University of New South Wales Working Paper (Available on Request)*.

Wang, Cong, and Fei Xie. 2009. “Corporate Governance Transfer and Synergistic Gains from Mergers and Acquisitions”. *The Review of Financial Studies* 22 (2): 829–58.

Appendix

A Glossary

This glossary provides definitions of programming and econometric terms as used throughout the paper.

A.1 Programming Terms

Declarative language A programming language in which the programmer specifies what the program should accomplish, rather than detailing how to perform each step. Programs written in declarative languages emphasize logic and relationships, often leading to more concise code, leaving the system to work out the control flow and implementation details needed to accomplish the task. In *foghorn*, researchers write declarative econometric specifications, with the transpiler determining the steps necessary for implementation.

Embedded Domain-Specific Language (EDSL) A specialized programming language built within a general-purpose language, designed to express solutions in a specific domain (e.g., econometrics) more naturally and concisely. EDSLs leverage the host language's syntax and features while providing domain-specific tools and constructs. Developers commonly use EDSLs to express logic specific to a domain, benefiting from enhanced type safety and improved tooling support.Haskell is well-suited for EDSLs because of its strong abstractions and type safety. *foghorn* is an EDSL of Haskell, providing concise, type-safe econometric logic.

Function composition The process of combining two or more functions to create a new function, where the output of one serves as the input of the next. Function composition allows programmers to build complex operations from simple, reusable parts in a concise, declarative manner. In *foghorn*, the composition of econometric functions improves modularity, readability, and code reuse.

Higher-order function A function that takes other functions as arguments, returns a function as its result, or both. Higher-order functions are central to expressing abstract computation succinctly and avoiding repetitive code. *foghorn* employs higher-order functions to improve modularity and reduce boilerplate.

Imperative language A programming language in which programs are written as sequences of explicit instructions that specify how a computer should perform tasks. These languages focus on describing the control flow and individual steps needed to manipulate program state, often using variables, loops, and conditional statements. Imperative programming contrasts with declarative styles, where the focus is on desired outcomes rather than step-by-step procedures. Existing econometric languages are either imperative or used in an imperative style (e.g., SAS, SQL).

Module A self-contained unit of code that groups related functions, types, and definitions under a common namespace. Modules promote code organization, reuse, and abstraction by allowing programmers to separate and isolate components within a program. *foghorn* uses modules for its core language and for working in specific empirical areas of empirical research (e.g., corporate finance).

Static typing A system in which the types of variables and functions are evaluated at compile time rather than at runtime. Static typing allows many errors—such as type mismatches or invalid operations—to be caught early, improving reliability and program safety. *foghorn* variables carry type information, including the cross-sectional and/or time-series indexing dimensions, allowing the language to validate that operations make sense automatically. *foghorn* uses static typing alongside type-level programming and type inference to encode and enforce econometric logic in specifications.

Term-level The "usual" layer of programming and code execution where values are defined, manipulated, and passed to functions. Unlike type-level programming, term-level programming focuses on concrete computations and data transformations. Most everyday programming tasks—such as arithmetic, data processing, and control flow—occur at the term level. In *foghorn*, the term-level defines the actual operations.

Transpiling The process of converting source code written in one programming language into source code of another language, while preserving program behavior. Unlike compiling, which often targets low-level machine code, transpiling typically produces human-readable code in another high-level language. *foghorn* allows econometricians to write specifications with enhanced features and stronger guarantees—such as improved type safety—while producing output in established data and econometric software.

Type A type is an abstract classification that specifies the kind of values a variable, expression, or function can take and the operations that may be performed on them. Types provide a framework for reasoning about program behavior, ensuring consistency and preventing invalid operations. *foghorn* variables carry type information on their index variables (i.e., the cross-sectional and/or time-series dimensions) and information type (e.g., numeric, SIC code, text). This, combined with static typing, type inference, and type-level programming lets the language reason about econometric specifications.

Type checking Type checking is the process by which a compiler or interpreter verifies that program constructs are used consistently with their declared or inferred types. It ensures, for example, that operations are applied to compatible data types and that functions receive valid arguments. *foghorn* uses type checking to ensure that econometric specifications, and the variables within them, are consistent.

Type inference Type inference is the process by which a compiler automatically determines the types of expressions without requiring explicit type annotations from the programmer. This enables concise code while retaining the benefits of static typing, since the compiler can still catch type errors at compile time. *foghorn* leverages Haskell's type inference to achieve strong econometric type safety with reduced syntactic overhead as an econometrician does not need to label a variable's index or informational types.

Type signature A formal declaration of the inputs and output of a function, along with any constraints. *foghorn* uses type signatures to defined econometric logic.

Type-level A layer above the term-level where logic about types is encoded. In strongly typed functional languages like Haskell, type-level programming allows developers to encode invariants, perform compile-time checks, and enforce constraints through types themselves. Type-level programming is the practice of writing programs that use a language's type system to encode logic, perform computations, and write expressions and abstractions that occur within a language's type system rather than at the term-level. This approach can enforce complex invariants, guarantee program properties at compile time, and eliminate certain classes of runtime errors. In *foghorn*, type-level programming define econometric operations, and specifications.

A.2 Econometric & Research Terms

Bi-directional transparency Transparency in both directions: top-down (explicit specifications and assumptions) and bottom-up (generated data-manipulation code). Ensures hidden assumptions are revealed.

Empirical specification A formal, high-level description of the econometric model being estimated, consisting of: the dependent variable, explanatory variables, the functional form of the relationship, and the chosen estimator. In this paper, the specification is written at a high level and does not include step-by-step data manipulation.

Empirical procedure The complete process of estimating a model from raw data. It consists of the complete data manipulation process that transforms raw data into the final dataset and the econometric estimation procedure.

Indexing variables Identifiers that uniquely label dataset observations, which may be cross-sectional or time-series.

Reindexing Reindexing is the process of aligning data across datasets with different indexing systems

Replication Re-implementing a paper's empirical specification to see if results can be reproduced with comparable magnitude, significance, and sample size. In this paper, replication is the process of independently coding an empirical specification as described within a paper. The replication is conducted from raw data, limiting author-provided information to proprietary data. Replication in this context is judged successful when: (i) coefficient estimates and significance are similar, (ii) sample sizes align, and (iii) control variables behave similarly.

Reproduction Obtaining exact results, or results with minor variations, to a published paper's empirical specification using data and code provided by the authors.

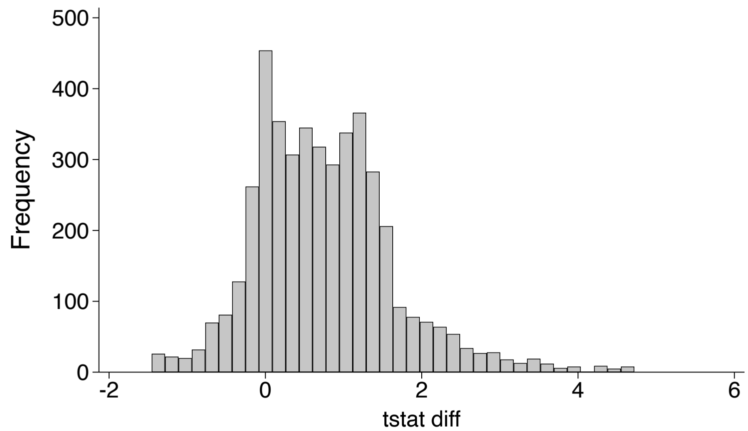
Robustness test A paper-level sensitivity check in which the authors vary aspects of their own methodology to show that results are not overly sensitive to small changes. These tests are typically idiosyncratic, tied to one paper's assumptions

Stress test A literature-level test that systematically varies common assumptions or practices across multiple papers.

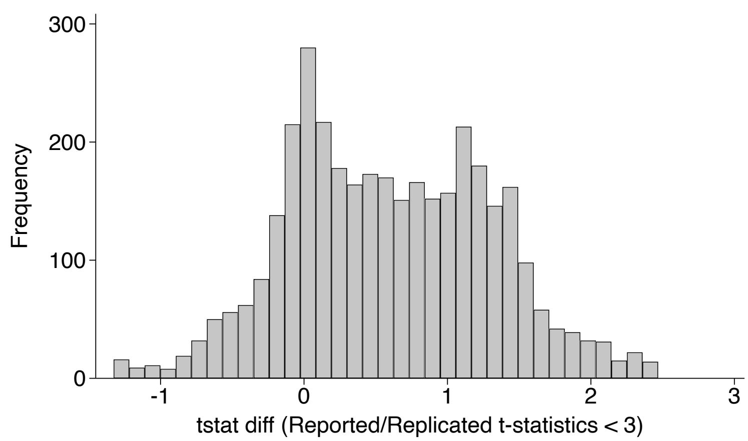
Figure 1

CAR Stress Tests: Histogram of the Change in t-statistics of Variables of Interest

This histogram plots *tstat_diff* for stress tests on the significance of the coefficient on the variable of interest by varying the definition of Cumulative Abnormal Returns (CARs) performed on 15 specifications from replicated papers. Stress tests allow for within-literature variation on (1) the CAR announcement window ([−1, +1], [−2, +2], [−3, +3], [−5, +5], [−1, +5], [−5, +1], [0, +5], and [−5, 0]), (2) abnormal return calculation (equally and value weighted single-index model (SIM), a Fama-French 3-factor (FF3) and 4-factor (FF4) model, and the equally weighted (OEW) and value weighted (OVW) market return). (3) the expected return estimation period for tests estimating the market return using the SIM, FF3, and FF4 ([−205, −6], [−210, −11], [−245, −45], [−252, −20], [−272, −20], [−300, −91], and [−370, −253]). *tstat_diff* defined as the reported/replicated t-statistic less the stress test t-statistic (for reported/replicated t-statistics > 0) and the stress test t-statistic less the reported/replicated t-statistic (for reported/replicated t-statistics < 0). Panel A plots the 1st through 99th percentiles of all stress tests resulting in 4,543 stress tests. Panel B plots the 1st through 99th percentiles of a subsample of 3,633 stress tests where the reported/replicated t-statistic is < 3.



Panel A: All original specifications



Panel B: Original specifications with t-statistic < 3

Figure 2

CAR Stress Tests: Histogram of the Change in t-statistics of Control Variables

This histogram plots *tstat_diff* for stress tests on the significance of the coefficients on control variables by varying the definition of Cumulative Abnormal Returns (CARs) performed on 15 specifications from replicated papers. Stress tests allow for within-literature variation on (1) the CAR announcement window ([−1, +1], [−2, +2], [−3, +3], [−5, +5], [−1, +5], [−5, +1], [0, +5], and [−5, 0]), (2) abnormal return calculation (equally and value weighted single-index model (SIM), a Fama-French 3-factor (FF3) and 4-factor (FF4) model, and the equally weighted (OEW) and value weighted (OVW) market return). (3) the expected return estimation period for tests estimating the market return using the SIM, FF3, and FF4 ([−205, −6], [−210, −11], [−245, −45], [−252, −20], [−272, −20], [−300, −91], and [−370, −253]). *tstat_diff* defined as the reported/replicated t-statistic less the stress test t-statistic (for reported/replicated t-statistics > 0) and the stress test t-statistic less the reported/replicated t-statistic (for reported/replicated t-statistics < 0). The figure plots the 1st through 99th percentiles of all stress tests resulting in 52,234 stress tests (one *tstat_diff* for each control variable-specification-stress test).

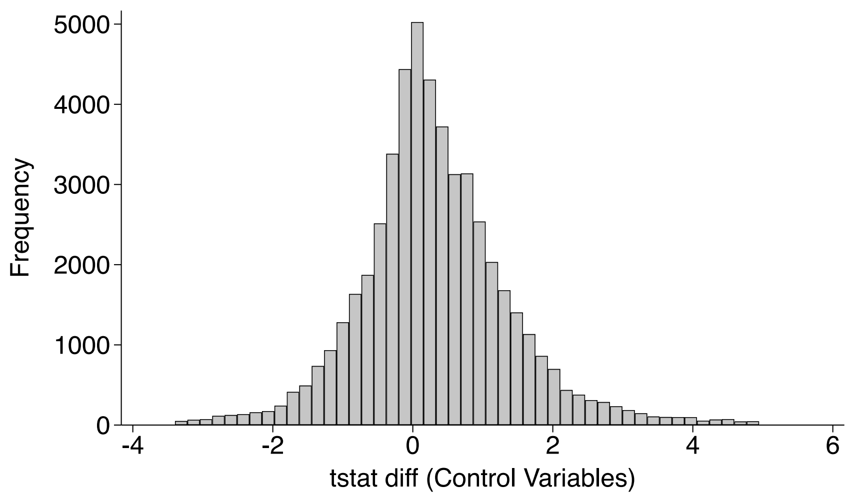


Figure 3

CAR Stress Tests: Histograms Individual Paper Specifications for Variables of Interest

This figure plots the histogram of *tstat_diff* by specification for each of the 15 specifications from replicated papers. Each histogram presents stress tests on the significance of the coefficient on the variable of interest by varying the definition of Cumulative Abnormal Returns (CARs). Stress tests allow for within-literature variation on (1) the CAR announcement window ([−1, +1], [−2, +2], [−3, +3], [−5, +5], [−1, +5], [−5, +1], [0, +5], and [−5, 0]), (2) abnormal return calculation (equally and value weighted single-index model (SIM), a Fama-French 3-factor (FF3) and 4-factor (FF4) model, and the equally weighted (OEW) and value weighted (OVW) market return). (3) the expected return estimation period for tests estimating the market return using the SIM, FF3, and FF4 ([−205, −6], [−210, −11], [−245, −45], [−252, −20], [−272, −20], [−300, −91], and [−370, −253]). *tstat_diff* defined as the reported/replicated t-statistic less the stress test t-statistic (for reported/replicated t-statistics > 0) and the stress test t-statistic less the reported/replicated t-statistic (for reported/replicated t-statistics < 0). Specification numbers are assigned randomly to each specification. Each histogram plots the *tstat_diff* for all stress tests for that specification.

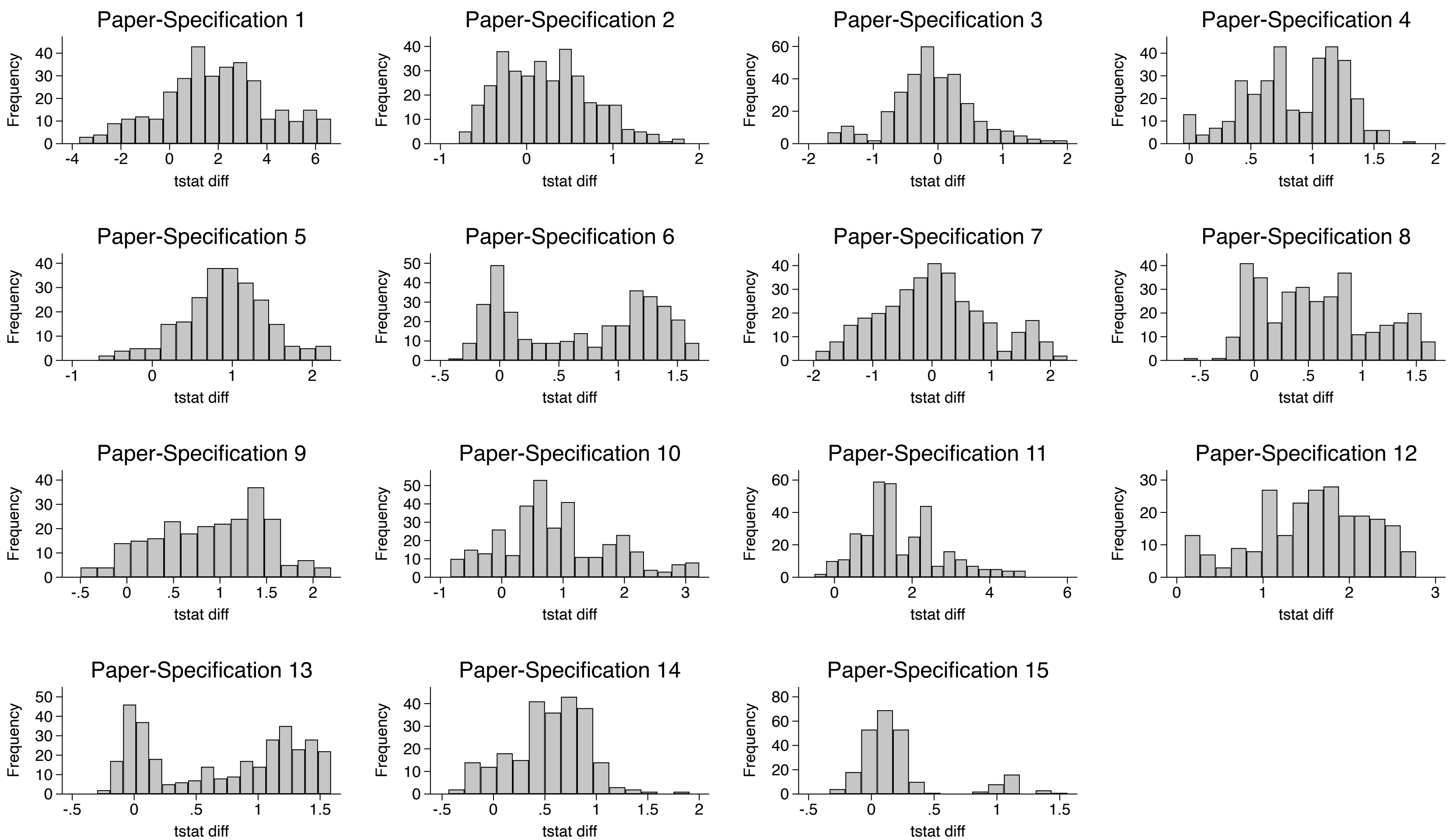
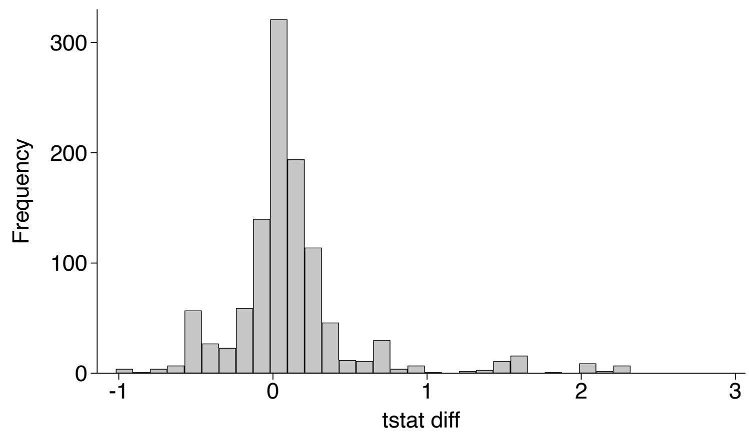

Paper-Specification 1
Paper-Specification 2
Paper-Specification 3
Paper-Specification 4
Paper-Specification 5
Paper-Specification 6
Paper-Specification 7
Paper-Specification 8
Paper-Specification 9
Paper-Specification 10
Paper-Specification 11
Paper-Specification 12
Paper-Specification 13
Paper-Specification 14
Paper-Specification 15
Frequency
tstat diff

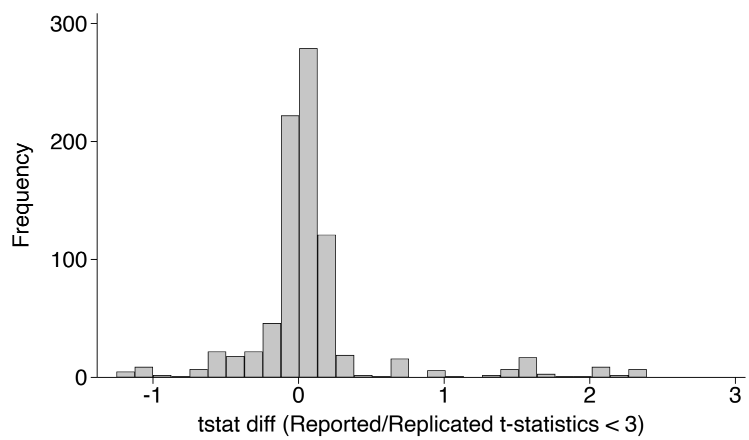
Figure 4

Control Variable Stress Tests: Histogram of the Change in t-statistics of Variables of Interest

This histogram plots *tstat_diff* for stress tests on the significance of the coefficient on the variable of interest by varying the definition of two independent control variables common across the majority of replicated specifications: market-to-book (Tobin's Q) and leverage. These are performed on all specifications that contain these variables (12) from replicated papers. Stress tests allow for within-literature variation based on the definitions presented in Table 2. *tstat_diff* defined as the reported/replicated t-statistic less the stress test t-statistic (for reported/replicated t-statistics > 0) and the stress test t-statistic less the reported/replicated t-statistic (for reported/replicated t-statistics < 0). Panel A plots the 1st through 99th percentiles of all stress tests resulting in 1,136 stress tests. Panel B plots the 1st through 99th percentiles of a subsample of 848 stress tests where the reported/replicated t-statistic is < 3.



Panel A: All original specifications



Panel B: Original specifications with t-statistic < 3

Figure 5

Control Variable Stress Tests: Histogram of the Change in t-statistics of Control Variables

This histogram plots *tstat_diff* for stress tests on the significance of the coefficients on control variables by varying the definition of two independent control variables common across the majority of replicated specifications: market-to-book (Tobin's Q) and leverage. These are performed on all specifications that contain these variables (12) from replicated papers. Stress tests allow for within-literature variation based on the definitions presented in Table 2. *tstat_diff* defined as the reported/replicated t-statistic less the stress test t-statistic (for reported/replicated t-statistics > 0) and the stress test t-statistic less the reported/replicated t-statistic (for reported/replicated t-statistics < 0). The figure plots the 1st through 99th percentiles of all stress tests resulting in 12,374 stress tests (one *tstat_diff* for each control variable-specification-stress test).

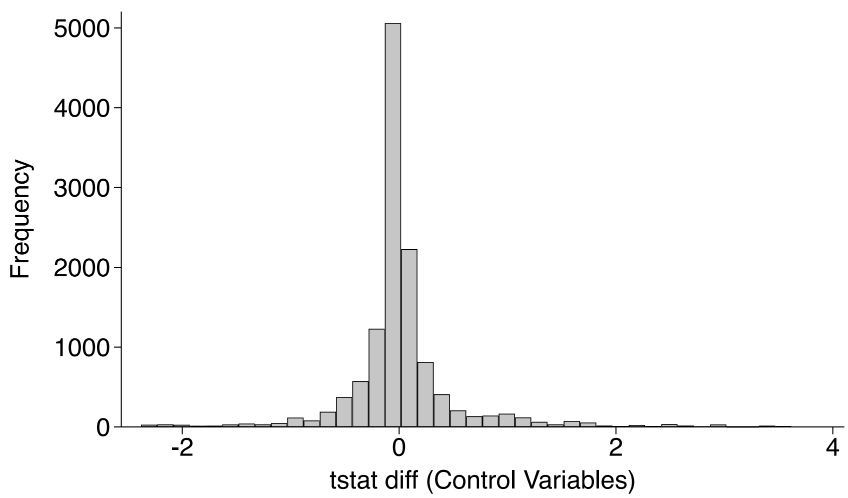


Table 1

Replication summary.

The table reports estimation results from linear regression models examining the relation between the market reaction to acquisition announcements and target lockup options. Replication code written in the foghorn declarative econometric language and the transpiled SQL data manipulation code are available on Github. *t*-statistics are reported in parentheses. Coefficients marked with ***, **, and * are significant at the 1%, 5% and 10% level, respectively.

Paper	Key Economic Variable, Table, and Column	Coefficient		Observations	
		Published	Replicated	Published	Replicated
Bates and Lemmon (2003)	Bidder Termination Fee Indicator Table 8, Column (1)	−0.030**	−0.022**	3037	3203
Becher, Griffin, and Nini (2021)	Financial Covenant Violation Table 6, Column (1)	1.860**	1.093**	7191	7299
Burch (2001)	Lockup(0/1) Table 6, Column (6)	−0.012*	−0.017***	744	766
Fuller, Netter, and Stegemoller (2002)	Dummy = 1, if fifth or higher bid Table 7, Column "Private""	−0.019***	−0.018***	2060	1313
Golubov, Yawson, and Zhang (2015)	Ln(Acquirer Size) Table 1, Column "Full Sample""	−0.004***	−0.003***	12 491	14 863
Gorton, Kahl, and Rosen (2009)	Log 123–456 size ratio Table V, Column "Harford Waves (4)""	0.029***	0.041**	1331	1141
Harford, Humphery-Jenner, and Powell (2012)	Dictator Dummy Table 5, Column (1)	−0.524**	−0.606**	3934	3258
John, Knyazeva, and Knyazeva (2015)	Acquirer - Weak Labor Rights Table 5, Column (3)	0.494***	0.267**	13 838	13 846

Continued

Table 1
Continued.

Paper	Key Economic Variable, Table, and Column	Coefficient		Observations	
		Published	Replicated	Published	Replicated
Li, Qiu, and Shen (2018)	OC *(Organizational Capital)* Table 2, Column 1	0.250***	0.329***	17 910	21 010
Ma, Whidbee, and Zhang (2019)	RPR *(Relative Price Ratio)* Table 3, Column "All"“	−5.487***	−9.397***	19 119	4529
Masulis, Wang, and Xie (2007)	GIM Index Table VI, Column 1	−0.107**	−0.083**	3333	3380
Moeller, Schlingemann, and Stulz (2004)	Small *(Market Capitalization Acquirers)* Table 5, Column 1	0.016***	0.015***	9712	10 796
Nguyen and Phan (2017)	PU_Announcement *(Political Uncertainty)* Table 7, Column 3	0.007**	0.008**	6376	6674
Roosenboom, Schlingemann, and Vasconcelos (2013)	Stock Liquidity Table 2, Column (3)	−0.038***	−0.006***	3815	4189
Wang and Xie (2009)	Shareholder Rights Difference Table 5, Column "TCAR"“	0.836***	0.594***	396	378

Table 2

Cumulative Abnormal Return Definitions in the M&A Literature.

The table presents the definitions of cumulative abnormal returns (CAR) used in the replicated papers. Panel A lists papers that use excess return models, where the daily abnormal return is the stock return less the market index. Panel B lists papers that use abnormal return models, where the daily abnormal return is the stock return less the return predicted by an empirical model. Event windows, *[start, end]*, indicate the start and end days use to sum abnormal returns. Both *start* and *end* are given in trading days relative to the announcement date. The market model include single index models, *SIM*, and the Fama-French 4 factor model, *FF4*. Estimation periods, *[start, end]*, list the start and end days used to estimate the model parameters. Both days are trading days relative to the announcement date.

#	References	Event Window	Market Index/Model	Model Estimation Period
Panel A: Excess Return Models				
1	Bates and Lemmon (2003), Gorton, Kahl, and Rosen (2009), Li, Qiu, and Shen (2018)	[−1, 1]	CRSP Value Weighted	N/A
2	Burch (2001)	[−1, 2]	CRSP Value Weighted	N/A
3	Fuller, Netter, and Stegemoller (2002)	[−2, 2]	CRSP Value Weighted	N/A
Panel B: Abnormal Return Models				
4	Becher, Griffin, and Nini (2021)	[−1, 1]	SIM CRSP Equal Weighted	[−271, −20]
5	Moeller, Schlingemann, and Stulz (2004)	[−1, 1]	SIM CRSP Equal Weighted	[−205, −6]
6	Nguyen and Phan (2017) ????	[−1, 1]	SIM CRSP Value Weighted	[−210, −11]
7	Harford, Humphery-Jenner, and Powell (2012), Masulis, Wang, and Xie (2007)	[−2, 2]	SIM CRSP Equal Weighted	[−210, −11]
8	Roosenboom, Schlingemann, and Vasconcelos (2013)	[−2, 2]	SIM CRSP Value Weighted	[−245, −46]
9	John, Knyazeva, and Knyazeva (2015)	[−2, 2]	SIM CRSP Value Weighted	[−210, −11]

Continued

Table 2
Continued.

#	References	Event Window	Market Index/Model	Model Estimation Period
10	Golubov, Yawson, and Zhang (2015)	[−2, 2]	SIM CRSP Value Weighted	[−300, −91]
11	Ma, Whidbee, and Zhang (2019)	[−5, 1]	SIM CRSP Equal Weighted	[−370, −253]
12	Wang and Xie (2009)	[−5, 5]	SIM CRSP Value Weighted	[−210, −11]

Table 3

Market-to-book ratio (Tobin's q) control variable alternative definitions used for stress testing.

The table presents the various definitions of market-to-book ratio used in the literature. This ratio is also referred to as Tobin's q in the literature. For each definition, the table lists the formula as `datasource.variable`. `Funda` indicates a variable sourced from Compustat Fundamentals Annual Data, and `Fundq` indicates a variable sourced from Compustat Fundamentals Quarterly data. `at` is the total book value of assets, `ceq` is the total book value of common/ordinary equity, `csho` is the total number of shares of common/ordinary equity as of the fiscal year end, `dlc` is the total book value of debt in current liabilities, `dltt` is the total book value of long-term debt, `lt` is the total book value of liabilities, `prccF` is the closed sharing price at the end of the fiscal year, `txdb` is the balance sheet value of deferred taxes, and `txditc` is the value of deferred taxes and investment tax credit. Missing values of `txdb` and `txditc` are treated as zeros. `preferred` is the liquidating value of preferred stock (`Funda. pstkl`) if it is available, otherwise it is the redemption value (`Funda. pstkrv`).

#	Formula References
1	`(Fundq.at - Fundq.ceq + Fundq.csho × Fundq.prccF) / Fundq.at` Becher, Griffin, and Nini (2021)
2	`(Funda.at - Funda.ceq + Funda.csho × Funda.prccF) / Funda.at` Burch (2001), Harford, Humphery-Jenner, and Powell (2012), Masulis, Wang, and Xie (2007), Moeller, Schlingemann, and Stulz (2004)
3	`(Funda.at + (Funda.prccF × Funda.csho) - Funda.ceq - Funda.txdb) / Funda.at` Golubov, Yawson, and Zhang (2015)
4	`Funda.at + (Funda.prccF × Funda.csho) - Funda.ceq - Funda.txdb / Funda.ceq` Gorton, Kahl, and Rosen (2009)
5	`Funda.csho × Funda.prccF / Funda.at` Li, Qiu, and Shen (2018)
6	`ln (Funda.csho × Funda.prccF / Funda.ceq)` Ma, Whidbee, and Zhang (2019)
7	`ln (Funda.csho × Funda.prccF / Funda.ceq)` Nguyen and Phan (2017)
8	`(Funda.lt - txditc + preferred + Funda.csho × Funda.prccF) / Funda.at` Roosenboom, Schlingemann, and Vasconcelos (2013)
9	`(Funda.at - Funda.ceq + Funda.csho × Funda.prccF) / Funda.at` Wang and Xie (2009)

Table 4

Leverage control variable alternative definitions used for stress testing.

The table presents the various definitions of leverage ratios used in the literature. For each definition, the table lists the formula as `datasource.variable`. `Funda` indicates a variable sourced from Compustat Fundamentals Annual Data, and `Fundq` indicates a variable sourced from Compustat Fundamentals Quarterly data. `Funda.at` (`Funda.atq`) is the annual (quarterly) total book value of assets, `Funda.ceq` is the total annual book value of common/ordinary equity, `Funda.csho` is the total annual number of shares of common/ordinary equity as of the fiscal year end, `Funda.dlc` (`Funda.dlcq`) is the total annual (quarterly) book value of debt in current liabilities, `Funda.dltt` (`Funda.dlttq`) is the total annual (quarterly) book value of long-term debt, `Funda.lt` is the total annual book value of liabilities, `Funda.prccF` is the closed sharing price at the end of the fiscal year, `Funda.txditc` is the annual value of deferred taxes and investment tax credit. Missing values of `Funda.txditc` are treated as zeros. `preferred` is the liquidating value of preferred stock (`Funda.pstkl`) if it is available, otherwise it is the redemption value (`Funda.pstkrv`).

#	`Formula` References
1	`(Fundq.dlttq + Fundq.dlcq) / Fundq.atq` Becher, Griffin, and Nini (2021)
2	`(Funda.dlc + Funda.dltt) / Funda.at` Burch (2001), Harford, Humphery-Jenner, and Powell (2012), Nguyen and Phan (2017)
3	`(Funda.dltt + Funda.dlc) / (Funda.lt - txditc + preferred + Funda.csho × Funda.prccF)` Golubov, Yawson, and Zhang (2015), Roosenboom, Schlingemann, and Vasconcelos (2013)
4	`(Funda.at - Funda.ceq) / Funda.at` Ma, Whidbee, and Zhang (2019)
5	`(Funda.dlc + Funda.dltt) / (Funda.at + Funda.csho × Funda.prccF)` Masulis, Wang, and Xie (2007)
6	`(Funda.dltt + Funda.dlc) / (Funda.at - Funda.ceq + Funda.csho × Funda.prccF)` Moeller, Schlingemann, and Stulz (2004)
7	`Funda.dlc + Funda.dltt / (Funda.dlc + Funda.dltt + Funda.csho × Funda.prccF)` Li, Qiu, and Shen (2018)
8	`(Funda.dlc + Funda.dltt) / Funda.at` Wang and Xie (2009)

Table 5

CAR Stress Tests: Variables of Interest

size: 9pt, { par(justify: true, leading: 0.2em,) [This table reports summary statistics on stress tests on the significance of the coefficient on the variable of interest by varying the definition of Cumulative Abnormal Returns (CARs) performed on 15 specifications from replicated papers. Stress tests allow for within-literature variation on (1) the CAR announcement window ([−1, +1], [−2, +2], [−3, +3], [−5, +5], [−1, +5], [−5, +1], [0, +5], and [−5, 0]), (2) abnormal return calculation (equally and value weighted single-index model (SIM), a Fama-French 3-factor (FF3) and 4-factor (FF4) model, and the equally weighted (OEW) and value weighted (OVW) market return). (3) the expected return estimation period for tests estimating the market return using the SIM, FF3, and FF4 ([−205, −6], [−210, −11], [−245, −45], [−252, −20], [−272, −20], [−300, −91], and [−370, −253]). Panel A reports summary statistics for *tstat_diff*. *tstat_diff* is defined as the reported/replicated t-statistic less the stress test t-statistic (for reported/replicated t-statistics > 0) and the stress test t-statistic less the reported/replicated t-statistic (for reported/replicated t-statistics < 0). Panel B reports distribution statistics for *tstat_diff*. Panel C reports summary statistics on the proportion of stress tests resulting in lower significance (*tstat_diff*<0), a reduction in the level of significance (i.e. moving from 1% significance to 5% significance), a loss of significance, and an increase in the level of significance.] },)

	N	Mean	Std. Dev.	Skewness	Kurtosis
Panel A: Summary statistics					
All specifications	4543	0.774	1.052	1.257	7.802
<3	3633	0.597	0.762	0.089	3.156

	Min	1%	5%	10%	25%	50%	75%	90%	95%	99%	Max
Panel B: Distribution											
All specifications	−3.711	−1.475	−0.560	−0.228	0.087	0.674	1.261	1.888	2.512	4.720	6.634
<3	−1.966	−1.327	−0.564	−0.265	0.040	0.560	1.154	1.527	1.830	2.486	3.220

	Increase in tstat_diff	Drop in Significance	Loss of Significance	Increase in Significance
Panel C: Changes in statistical significance thresholds (%)				
All specifications	80.176	52.651	36.897	4.708
<3	77.744	63.603	46.052	5.887

Table 6

CAR Stress Tests: Control Variables

This table reports summary statistics on stress tests on the significance of coefficients on control variables by varying the definition of Cumulative Abnormal Returns (CARs) performed on 15 specifications from replicated papers. Stress tests allow for within-literature variation on (1) the CAR announcement window ([−1, +1], [−2, +2], [−3, +3], [−5, +5], [−1, +5], [−5, +1], [0, +5], and [−5, 0]), (2) abnormal return calculation (equally and value weighted single-index model (SIM), a Fama-French 3-factor (FF3) and 4-factor (FF4) model, and the equally weighted (OEW) and value weighted (OVW) market return). (3) the expected return estimation period for tests estimating the market return using the SIM, FF3, and FF4 ([−205, −6], [−210, −11], [−245, −45], [−252, −20], [−272, −20], [−300, −91], and [−370, −253]). Panel A reports summary statistics for *tstat_diff*. *tstat_diff* is defined as the reported/replicated t-statistic less the stress test t-statistic (for reported/replicated t-statistics > 0) and the stress test t-statistic less the reported/replicated t-statistic (for reported/replicated t-statistics < 0). Panel B reports distribution statistics for *tstat_diff*. Panel C reports summary statistics on the proportion of stress tests resulting in lower significance (*tstat_diff* < 0), a reduction in the level of significance (i.e. moving from 1% significance to 5% significance), a loss of significance, and an increase in the level of significance.

	N	Mean	Std. Dev.	Skewness	Kurtosis
Panel A: Summary statistics					
All specifications	52 234	0.336	1.390	1.347	16.592

	Min	1%	5%	10%	25%	50%	75%	90%	95%	99%	Max
Panel B: Distribution											
All specifications	−14.243	−3.397	−1.430	−0.948	−0.291	0.222	0.893	1.675	2.352	4.946	15.480

	Increase in tstat_diff	Drop in Significance	Loss of Significance	Increase in Significance
Panel C: Changes in statistical significance thresholds (%)				
All specifications	61.418	15.985	11.496	11.250

Table 7

Comparing the Distributions of Variables of Interest and Control Variables in CAR Stress Tests

This table reports a two-sample Kolmogorov-Smirnov test for the equality of distributions between the distribution of *tstat_diff* on variables of interest (corresponding to those described in Table 7) and the distribution of *tstat_diff* on control variables (corresponding to those described in Table 10).

Group	D	p-value
Controls	0.197	<0.001
Variables of Interest	-0.004	0.816
Combined K-S	0.197	<0.001

Table 8

Control Variable Definition Stress Tests: Variables of Interest

This table reports summary statistics on stress tests on the significance of the coefficient on the variable of interest by varying the definition of two independent control variables common across the majority of replicated specifications: market-to-book (Tobin's Q) and leverage. These are performed on all specifications that contain these variables (12) from replicated papers. Stress tests allow for within-literature variation based on the definitions presented in Table 2. Panel A reports summary statistics for *tstat_diff*. *tstat_diff* is defined as the reported/replicated t-statistic less the stress test t-statistic (for reported/replicated t-statistics > 0) and the stress test t-statistic less the reported/replicated t-statistic (for reported/replicated t-statistics < 0). Panel B reports distribution statistics for *tstat_diff*. Panel C reports summary statistics on the proportion of stress tests resulting in lower significance (*tstat_diff*<0), a reduction in the level of significance (i.e. moving from 1% significance to 5% significance), a loss of significance, and an increase in the level of significance.

	N	Mean	Std. Dev.	Skewness	Kurtosis
Panel A: Summary statistics					
All specifications	1136	0.132	0.533	2.336	11.693
<3	848	0.106	0.499	2.068	9.994

	Min	1%	5%	10%	25%	50%	75%	90%	95%	99%	Max
Panel B: Distribution											
All specifications	−1.251	−1.047	−0.501	−0.338	−0.029	0.032	0.207	0.546	1.446	2.320	3.473
<3	−1.251	−1.088	−0.506	−0.251	−0.029	0.025	0.135	0.323	1.502	2.145	2.391

	Increase in tstat_diff	Drop in Significance	Loss of Significance	Increase in Significance
Panel C: Changes in statistical significance thresholds (%)				
All specifications	62.676	18.661	10.651	2.464
<3	61.320	24.764	14.268	3.301

Table 9

Control Variable Definition Stress Tests: Control Variables

This table reports summary statistics on stress tests on the significance of coefficients on control variables by varying the definition of two independent control variables common across the majority of replicated specifications: market-to-book (Tobin's Q) and leverage. These are performed on all specifications that contain these variables (12) from replicated papers. Stress tests allow for within-literature variation based on the definitions presented in Table 2. Panel A reports summary statistics for *tstat_diff*. *tstat_diff* is defined as the reported/replicated t-statistic less the stress test t-statistic (for reported/replicated t-statistics > 0) and the stress test t-statistic less the reported/replicated t-statistic (for reported/replicated t-statistics < 0). Panel B reports distribution statistics for *tstat_diff*. Panel C reports summary statistics on the proportion of stress tests resulting in lower significance (*tstat_diff*<0), a reduction in the level of significance (i.e. moving from 1% significance to 5% significance), a loss of significance, and an increase in the level of significance.

	N	Mean	Std. Dev.	Skewness	Kurtosis
Panel A: Summary statistics					
All specifications	12 784	0.044	0.836	1.021	31.564

	Min	1%	5%	10%	25%	50%	75%	90%	95%	99%	Max
Panel B: Distribution											
All specifications	−9.492	−2.381	−0.696	−0.386	−0.111	−0.001	0.086	0.484	1.097	3.618	11.730

	Increase in tstat_diff	Drop in Significance	Loss of Significance	Increase in Significance
Panel C: Changes in statistical significance thresholds (%)				
All specifications	46.127	4.458	2.933	4.497

Table 10

Comparing the Distributions of Variables of Interest and Control Variables in Control Variable Stress Tests

This table reports a two-sample Kolmogorov-Smirnov test for the equality of distributions between the distribution of *tstat_diff* on variables of interest (corresponding to those described in Table 14) and the distribution of *tstat_diff* on control variables (corresponding to those described in Table 17).

Group	D	p-value
Controls	0.190	<0.001
Variables of Interest	-0.014	0.661
Combined K-S	0.190	<0.001

Stress Testing a Literature with Declarative Econometrics

Vanitha Ragunathan
University of Queensland

Jared Stanfield
University of Oklahoma

Robert Tumarkin[††††]
UNSW Sydney

September 01, 2025

Internet Appendix

* Corresponding Author. UNSW Business School, UNSW Sydney, Sydney NSW 2052, Australia. Telephone: +(61) 02 9065 1656. E-mail: r.tumarkin@unsw.edu.au.

Replication Table IA1

Replication of Bates and Lemmon (2003), Table 8, Column (1).

The table reports estimation results from linear regression models examining the relation between the market reaction to acquisition announcements and target and bidder termination fees. The dependent variable is the acquiring firm's cumulative abnormal return, which is aggregated over the 3-day trading window beginning 1 trading days prior to and ending 1 trading days following the merger announcement. All variables are as defined in Bates and Lemmon (2003). Replication code written in the foghorn declarative econometric language and the transpiled SQL data manipulation code are available on Github. *t*-statistics are reported in parentheses. Coefficients displayed in bold represent the key economic variable of interest in the paper. Coefficients marked with ***, **, and * are significant at the 1%, 5% and 10% level, respectively.

	Column (1)	
	Published	Replicated
Target termination fee indicator	0.010	0.014*
	(0.960)	(1.703)
Bidder termination fee indicator	**-0.030****	**-0.022****
	(-2.340)	**(-2.009)**
Deal includes a lockup of target shares	0.002	0.002
	(0.820)	(0.286)
Deal status (1=completed 0=withdrawn)	0.039***	0.018**
	(3.440)	(1.978)
Prior bidding indicator	-0.063***	-0.057***
	(-6.120)	(-5.959)
Stock offer	-0.025***	-0.034***
	(-2.540)	(-4.145)
Tender offer	0.091***	0.080***
	(7.530)	(7.366)
Bidder toehold	-0.063***	-0.001**
	(-4.320)	(-2.434)
Deal attitude (1=hostile 0=friendly or unsolicited)	0.040**	0.032**
	(2.260)	(2.206)
Log marketvalue of equity	-0.012***	<0.001***
	(-4.600)	(-3.521)
Number of observations	3037	3203
Adjusted-R^2	0.067	0.063

Replication Table IA2

Replication of Becher, Griffin, and Nini (2021), Table 6, Column (1).

The table reports estimation results from linear regression models examining the relation between the market reaction to acquisition announcements and financial covenant violations. The dependent variable is the acquiring firm's cumulative abnormal return, which is aggregated over the 3-day trading window beginning 1 trading days prior to and ending 1 trading days following the merger announcement. All variables are as defined in Becher, Griffin, and Nini (2021). Replication code written in the foghorn declarative econometric language and the transpiled SQL data manipulation code are available on Github. Standard errors are reported in cornered parentheses. Coefficients displayed in bold represent the key economic variable of interest in the paper. Coefficients marked with ***, **, and * are significant at the 1%, 5% and 10% level, respectively.

	Column (1)	
	Published	Replicated
Financial covenant violation	**1.860***	**1.093**
	〔**0.687**〕	〔**0.537**〕
Size	-0.057***	-0.627***
	〔0.007〕	〔0.059〕
Stock price runup	-0.041	-0.003
	〔0.241〕	〔0.002〕
Market-to-book ratio	-0.277**	-0.278**
	〔0.108〕	〔0.118〕
Operating cash flow / assets	-0.711	9.852**
	〔0.933〕	〔3.952〕
Leverage ratio	0.870	0.847
	〔0.556〕	〔0.574〕
Observations	7191	7299
Adjusted R-squared	0.018	0.030

Replication Table IA3

Replication of Burch (2001), Table 6, Column (6).

The table reports estimation results from linear regression models examining the relation between the market reaction to acquisition announcements and target lockup options. The dependent variable is the acquiring firm's cumulative abnormal return, which is aggregated over the 4-day trading window beginning 1 trading days prior to and ending 2 trading days following the merger announcement. All variables are as defined in Burch (2001). Replication code written in the foghorn declarative econometric language and the transpiled SQL data manipulation code are available on Github. *p*-values are reported in brackets. Coefficients displayed in bold represent the key economic variable of interest in the paper. Coefficients marked with ***, **, and * are significant at the 1%, 5% and 10% level, respectively.

	Column (6)	
	Published	Replicated
Lockup (0/1)	**-0.012***	**-0.017*****
	[0.093]	**[0.002]**
Toehold	0.001**	<0.001
	[0.039]	[0.994]
Completed	<0.001	0.002
	[0.945]	[0.820]
Hostile	0.001	0.015
	[0.880]	[0.209]
Free cash flow	-0.002	-0.030
	[0.910]	[0.285]
Instown	-0.023	-0.012
	[0.106]	[0.370]
Litigation	-0.006	-0.002
	[0.427]	[0.804]
Market-to-book	-0.007***	-0.009***
	[0.002]	[<0.001]
Size	-0.002	-0.003*
	[0.289]	[0.086]
Leverage	0.003	0.023*
	[0.833]	[0.060]
Observations	744	776
Adjusted R-squared	0.018	0.043

Replication Table IA4

Replication of Fuller, Netter, and Stegemoller (2002), Table VII, Column "Private".

The table reports estimation results from linear regression models examining the relation between the market reaction to acquisition announcements and bidding firm acquisition frequency. The dependent variable is the acquiring firm's cumulative abnormal return, which is aggregated over the 5-day trading window beginning 2 trading days prior to and ending 2 trading days following the merger announcement. All variables are as defined in Fuller, Netter, and Stegemoller (2002). Replication code written in the foghorn declarative econometric language and the transpiled SQL data manipulation code are available on Github. *p*-values are reported in brackets. Coefficients displayed in bold represent the key economic variable of interest in the paper. Coefficients marked with ***, **, and * are significant at the 1%, 5% and 10% level, respectively.

	Column "Private"	
	Published	Replicated
Dummy = 1 if target is acquired with common stock	0.043***	0.115**
	[0.007]	[0.025]
Dummy = 1 if target is acquired with combo	0.009	0.032
	[0.460]	[0.580]
Dummy = 1 if first bid	-0.003	-0.010
	[0.685]	[0.305]
Dummy = 1 if fifth or higher bid	**-0.019*** **	**-0.018*** **
	[<0.001]	**[0.005]**
Dummy = 1 if target is foreign	-0.012*	-0.006
	[0.062]	[0.474]
Dummy = 1 if bidder or target is a tech firm	-0.004	-0.002
	[0.431]	[0.722]
Dummy = 1 if target and bidder are in same industry	0.004	-0.013**
	[0.358]	[0.035]
Log of relative size	0.007***	0.001
	[0.010]	[0.662]
Log of target size	0.001	0.002
	[0.442]	[0.390]
Interaction variable = relative size × stock	0.011**	0.011**
	[0.012]	[0.020]
Interaction variable = relative size × combo	0.003	0.002
	[0.513]	[0.739]
F-statistic	5.140	6.310
N	2060	1313
Adjusted R2	0.035	0.042

Replication Table IA5

Replication of Golubov, Yawson, and Zhang (2015), Table 1, Column "Full Sample".

The table reports estimation results from linear regression models examining the relation between the market reaction to acquisition announcements and acquirer size. The dependent variable is the acquiring firm's cumulative abnormal return, which is aggregated over the 5-day trading window beginning 2 trading days prior to and ending 2 trading days following the merger announcement. All variables are as defined in Golubov, Yawson, and Zhang (2015). Replication code written in the foghorn declarative econometric language and the transpiled SQL data manipulation code are available on Github. *t*-statistics are reported in parentheses. Coefficients displayed in bold represent the key economic variable of interest in the paper. Coefficients marked with ***, **, and * are significant at the 1%, 5% and 10% level, respectively.

	Column "Full Sample"	
	Published	Replicated
Intercept	0.032***	0.051***
	(2.941)	(3.226)
Ln (acquirer size)	**-0.004*****	**-0.003*****
	(5.496)	**(-2.856)**
Tobin's Q	-0.002***	-0.001*
	(2.966)	(-1.912)
Run-up	-0.013***	-0.005***
	(4.512)	(-3.484)
Free cash flow	-0.012	-0.014*
	(1.331)	(-1.711)
Leverage	0.017**	0.001
	(2.523)	(0.087)
Sigma	0.350**	0.075
	(2.306)	(0.694)
Relative size	0.002	29.263**
	(1.549)	(2.191)
Relatedness	<-0.001	0.001
	(0.160)	(0.683)
Tender offer	0.002	-0.001
	(0.392)	(-0.131)
Hostile	0.007	<0.001
	(0.592)	(-0.010)
Public × All-cash	-0.003	-0.006
	(0.755)	(-1.490)
Public × Stock	-0.032***	-0.042***
	(12.268)	(-9.530)
Private × All-cash	-0.004	-0.002
	(1.520)	(-0.933)
Private × Stock	-0.001	-0.002
	(0.259)	(-0.822)
Subsidiary × All-cash	0.007***	0.005**
	(2.606)	(2.066)
N	12491	14863
R2	0.057	0.082
Adj. R2	0.055	0.080

Replication Table IA6

Replication of Gorton, Kahl, and Rosen (2009), Table V, Column "Harford Waves (4)".

The table reports estimation results from linear regression models examining the relation between the market reaction to acquisition announcements and the distribution of firm sizes in the acquired firm's industry. The dependent variable is the acquiring firm's cumulative abnormal return, which is aggregated over the 3-day trading window beginning 1 trading days prior to and ending 1 trading days following the merger announcement. All variables are as defined in Gorton, Kahl, and Rosen (2009). Replication code written in the foghorn declarative econometric language and the transpiled SQL data manipulation code are available on Github. *p*-values are reported in brackets. Coefficients displayed in bold represent the key economic variable of interest in the paper. Coefficients marked with ***, **, and * are significant at the 1%, 5% and 10% level, respectively.

	Column "Harford Waves (4)"	
	Published	Replicated
Log 123–456 size ratio	**0.029*****	**0.041****
	[0.007]	**[0.023]**
Cash	0.001	0.004
	[0.881]	[0.520]
Ratio	0.003	0.007
	[0.694]	[0.497]
Log value	-0.006***	-0.004**
	[<0.001]	[0.029]
Tar priv	0.015***	0.024***
	[0.001]	[<0.001]
Tar sub	0.022***	0.020***
	[<0.001]	[0.010]
Cross industry	0.007	0.009
	[0.104]	[0.405]
Competing bid	0.006	-0.036**
	[0.720]	[0.014]
Tender offer	0.017*	0.019*
	[0.056]	[0.086]
Stock market return	-0.001	0.003
	[0.861]	[0.558]
Market/book	0.002*	0.001
	[0.080]	[0.114]
Industry Herfindahl	0.036	0.109**
	[0.720]	[0.032]
Observations	1334	1141
R2	0.113	0.075

Replication Table IA7

Replication of Harford, Humphery-Jenner, and Powell (2012), Table 5, Column (1).

The table reports estimation results from linear regression models examining the relation between the market reaction to acquisition announcements and managerial entrenchment. The dependent variable is the acquiring firm's cumulative abnormal return, which is aggregated over the 5-day trading window beginning 2 trading days prior to and ending 2 trading days following the merger announcement. All variables are as defined in Harford, Humphery-Jenner, and Powell (2012). Replication code written in the foghorn declarative econometric language and the transpiled SQL data manipulation code are available on Github. Standard errors are reported in cornered parentheses. Coefficients displayed in bold represent the key economic variable of interest in the paper. Coefficients marked with ***, **, and * are significant at the 1%, 5% and 10% level, respectively.

	Column (1)	
	Published	Replicated
Dictator dummy	**-0.524****	**-0.606****
	〔**-0.210**〕	〔**0.244**〕
Subsidiary	2.059***	2.723***
	〔-0.327〕	〔0.383〕
Private	1.495***	2.298***
	〔-0.278〕	〔0.384〕
All cash	0.313	0.297
	〔-0.310〕	〔0.255〕
All stock	-0.813**	-0.469
	〔-0.345〕	〔0.429〕
Log firm age	-0.016	-0.083
	〔-0.152〕	〔0.165〕
Stock run-up	0.977***	-1.937***
	〔-0.340〕	〔0.366〕
PRIV	-0.085***	
	〔-0.019〕	
Log market value	-0.318***	<0.001
	〔-0.083〕	〔<0.001〕
Tobin's q	0.292**	0.139
	〔-0.117〕	〔0.155〕
Free cash flow	6.625*	-1.906
	〔-3.892〕	〔2.188〕
Leverage	3.187***	-1.108
	〔-1.041〕	〔0.772〕
Industry M&A	-0.156	-2.797**
	〔-6.130〕	〔1.373〕
Relative size	0.146	1.674***
	〔-0.793〕	〔0.614〕

Continued

Replication Table IA7

Continued

	Column (1)	
	Published	Replicated
Tech	0.314	-0.212
	〔-0.248〕	〔0.336〕
Conglomerate	0.038	0.427*
	〔-0.227〕	〔0.238〕
Competed	-0.948	-1.142*
	〔-0.701〕	〔0.658〕
Volume	0.124	-0.334*
	〔-0.098〕	〔0.182〕
Cross-border	2.964**	-0.187
	〔-1.160〕	〔0.294〕
Friendly	-2.786***	-2.652***
	〔-0.870〕	〔0.879〕
Serial_3	0.092	-0.123
	〔-0.274〕	〔0.311〕
Number of observations	3934	3258
F-statistic	8.310***	8.579
Adjusted R2	0.073	0.053

Replication Table IA8

Replication of John, Knyazeva, and Knyazeva (2015), Table V, Column (3).

The table reports estimation results from linear regression models examining the relation between the market reaction to acquisition announcements and state-level labor rights. The dependent variable is the acquiring firm's cumulative abnormal return, which is aggregated over the 5-day trading window beginning 2 trading days prior to and ending 2 trading days following the merger announcement. All variables are as defined in John, Knyazeva, and Knyazeva (2015). Replication code written in the foghorn declarative econometric language and the transpiled SQL data manipulation code are available on Github. *t*-statistics are reported in parentheses. Coefficients displayed in bold represent the key economic variable of interest in the paper. Coefficients marked with ***, **, and * are significant at the 1%, 5% and 10% level, respectively.

	Column (3)	
Acquirer—weak labor rights	**0.494*****	**0.267****
	(5.440)	**(2.122)**
Acquirer size	-0.262***	-0.325***
	(-5.250)	(-6.621)
Relative deal size	-0.882***	2.024***
	(-3.530)	(10.551)
Diversifying acquisition	-0.081	-0.447**
	(-0.930)	(-2.115)
Tech indicator	0.276	-0.939***
	(0.590)	(-3.617)
Number of observations	13838	13846
R2	0.010	0.034

Replication Table IA9

Replication of Li, Qiu, and Shen (2018), Table 2, Column 1.

The table reports estimation results from linear regression models examining the relation between the market reaction to acquisition announcements and organizational capital. The dependent variable is the acquiring firm's cumulative abnormal return, which is aggregated over the 3-day trading window beginning 1 trading days prior to and ending 1 trading days following the merger announcement. All variables are as defined in Li, Qiu, and Shen (2018). Replication code written in the foghorn declarative econometric language and the transpiled SQL data manipulation code are available on Github. Standard errors are reported in cornered parentheses. Coefficients displayed in bold represent the key economic variable of interest in the paper. Coefficients marked with ***, **, and * are significant at the 1%, 5% and 10% level, respectively.

	Column 1	
OC	**0.250***	**0.329***
	〔**0.084**〕	〔**0.116**〕
ROA	-0.007	-0.011
	〔0.008〕	〔0.007〕
M/B	-0.032	-0.114***
	〔0.021〕	〔0.039〕
LEVERAGE	2.921***	3.138***
	〔0.396〕	〔0.904〕
PAST_RETURN	0.002*	0.781***
	〔0.001〕	〔0.170〕
TOP5_INSTITUTIONS	-1.905***	
	〔0.563〕	
FIRM_SIZE	-0.405***	-0.554***
	〔0.040〕	〔0.067〕
ALL_CASH	0.536***	0.410***
	〔0.127〕	〔0.142〕
ALL_STOCK	0.023	0.181
	〔0.196〕	〔0.287〕
DIVERSIFYING	-0.037	-0.066
	〔0.132〕	〔0.165〕
TENDER_OFFER	1.138***	1.241***
	〔0.311〕	〔0.331〕
RELATIVE_SIZE	0.950***	0.181**
	〔0.168〕	〔0.090〕
PRIVATE_TARGET	2.270***	1.949***
	〔0.186〕	〔0.234〕
SUBSIDIARY_TARGET	2.758***	2.729***
	〔0.198〕	〔0.261〕
No. of obs.	17910	21010
Adj. R2	0.053	0.038

Replication Table IA10

Replication of Ma, Whidbee, and Zhang (2019), Table 3, Column "All".

The table reports estimation results from linear regression models examining the relation between the market reaction to acquisition announcements and acquirer stock price relative to its 52-week high. The dependent variable is the acquiring firm's cumulative abnormal return, which is aggregated over the 7-day trading window beginning 5 trading days prior to and ending 1 trading days following the merger announcement. Reference price ratio (RPR) is the ratio of the acquirer's preannouncement stock price to its 52-week high price. All variables are as defined in Ma, Whidbee, and Zhang (2019). Replication code written in the foghorn declarative econometric language and the transpiled SQL data manipulation code are available on Github. *t*-statistics are reported in parentheses. Coefficients displayed in bold represent the key economic variable of interest in the paper. Coefficients marked with ***, **, and * are significant at the 1%, 5% and 10% level, respectively.

	Column "All"	
	Published	Replicated
RPR	**-5.487*****	**-9.397*** **
	(-8.560)	**(-5.676)**
Ln(M/B)	-0.285**	-1.531***
	(-2.140)	(-5.142)
Stock	-1.208***	-1.071**
	(-4.060)	(-1.991)
Cash	0.145	1.130***
	(0.910)	(3.375)
Private	2.321***	0.477
	(10.600)	(1.181)
Stock × Private	2.125***	6.244*
	(5.040)	(1.904)
Rel. size	2.216***	1.132**
	(8.800)	(1.961)
Size	-0.398***	-0.511***
	(-7.690)	(-5.063)
Leverage	-0.081	1.182
	(-0.210)	(1.357)
Dormant > 1 yr	0.628***	2.548***
	(2.880)	(5.948)
Same industry	0.107	0.266
	(0.670)	(0.711)
Tender offer	1.058**	4.050***
	(2.220)	(8.095)
Hostile	-0.759*	-2.155*
	(-1.890)	(-1.874)
Toehold	0.047	-0.914
	(0.100)	(-0.807)
Cross border	-0.187	-0.398
	(-0.770)	(-0.467)
Past return	0.777***	2.039***
	(4.700)	(6.100)
N	19119	4529
Adj. R2	0.064	0.116

Replication Table IA11

Replication of Masulis, Wang, and Xie (2007), Table VI, Column (1).

The table reports estimation results from linear regression models examining the relation between the market reaction to acquisition announcements and corporate governance. The dependent variable is the acquiring firm's cumulative abnormal return, which is aggregated over the 5-day trading window beginning 2 trading days prior to and ending 2 trading days following the merger announcement. All variables are as defined in Masulis, Wang, and Xie (2007). Replication code written in the foghorn declarative econometric language and the transpiled SQL data manipulation code are available on Github. *t*-statistics are reported in parentheses. Coefficients displayed in bold represent the key economic variable of interest in the paper. Coefficients marked with ***, **, and * are significant at the 1%, 5% and 10% level, respectively.

	Column (1)	
	Published	Replicated
Antitakeover Provisions:		
GIM index	**-0.107****	**-0.083****
	(-2.490)	**(-2.072)**
Bidder Characteristics:		
Log(total assets)	-0.301***	-0.328***
	(-3.590)	(-3.634)
Tobin's q	-0.085	-0.033
	(-0.680)	(-0.227)
Free cash flow	1.902	-0.755
	(0.860)	(-0.335)
Leverage	0.678	-0.284
	(0.640)	(-0.239)
Stock price runup	-0.906**	-1.526***
	(-2.540)	(-4.156)
Deal Characteristics:		
Industry M&A	-1.096	-1.637
	(-0.770)	(-1.110)
Relative deal size	0.209	1.486**
	(0.360)	(2.253)
High tech	0.420	-0.197
	(0.920)	(-0.506)
High tech × relative deal size	-6.078***	-0.824
	(-3.150)	(-0.355)
Diversifying acquisition	-0.269	0.165
	(-0.880)	(0.685)
Public target × stock deal	-3.902***	-2.758***
	(-7.290)	(-7.314)
Public target × all-cash deal	-2.082***	-0.942*
	(-3.340)	(-1.891)
Private target × all-cash deal	-1.969***	-0.685
	(-3.530)	(-1.502)
Private target × stock deal	-1.689***	-0.408
	(-3.100)	(-1.280)
Subsidiary target × all-cash deal	-1.472***	-0.069
	(-2.900)	(-0.190)
Number of obs.	3333	3380
Adjusted-R^2	0.062	0.055

Replication Table IA12

Replication of Moeller, Schlingemann, and Stulz (2004), Table 5, Column (1)

The table reports estimation results from linear regression models examining the relation between the market reaction to acquisition announcements and firm size. The dependent variable is the acquiring firm's cumulative abnormal return, which is aggregated over the 3-day trading window beginning 1 trading days prior to and ending 1 trading days following the merger announcement. All variables are as defined in Moeller, Schlingemann, and Stulz (2004). Replication code written in the foghorn declarative econometric language and the transpiled SQL data manipulation code are available on Github. *p*-values are reported in brackets. Coefficients displayed in bold represent the key economic variable of interest in the paper. Coefficients marked with ***, **, and * are significant at the 1%, 5% and 10% level, respectively.

	Column (1)	
	Published	Replicated
Intercept	0.015***	0.004
	[<0.001]	[0.196]
Private	-0.004*	-0.007***
	[0.085]	[0.004]
Public	-0.032***	-0.028***
	[<0.001]	[<0.001]
Small	**0.016*****	**0.015*****
	[<0.001]	**[<0.001]**
Conglomerate	-0.004*	0.001
	[0.051]	[0.736]
Tender offer	0.015***	0.014***
	[0.001]	[0.003]
Hostile	-0.012	0.019*
	[0.195]	[0.065]
Competed	-0.007	-0.010
	[0.299]	[0.234]
All equity	-0.003	0.001
	[0.341]	[0.829]
All cash	-0.004**	0.002
	[0.047]	[0.315]
Relative size	0.012***	<0.001
	[0.001]	[0.102]
Tobin's q	-0.001*	<0.001
	[0.064]	[0.573]
Debt/assets(mkt.)	0.001	-0.012*
	[0.876]	[0.057]
Liquidity index	-0.009***	-0.001
	[0.003]	[0.121]
CF/assets (mkt.)	0.001	0.031***
	[0.811]	[0.002]
n	9712	10796
Adjusted-R2	0.052	0.018

Replication Table IA13

Replication of Nguyen and Phan (2017), Table 7, Column (3).

The table reports estimation results from linear regression models examining the relation between the market reaction to acquisition announcements and government policy uncertainty. The dependent variable is the acquiring firm's cumulative abnormal return, which is aggregated over the 3-day trading window beginning 1 trading days prior to and ending 1 trading days following the merger announcement. All variables are as defined in Nguyen and Phan (2017). Replication code written in the foghorn declarative econometric language and the transpiled SQL data manipulation code are available on Github. *t*-statistics are reported in parentheses. Coefficients displayed in bold represent the key economic variable of interest in the paper. Coefficients marked with ***, **, and * are significant at the 1%, 5% and 10% level, respectively.

	Column (3)	
	Published	Replicated
PU_ANNOUNCEMENT	**0.007**** **(2.040)**	**0.008**** **(2.357)**
SIZE	-0.007*** (8.400)	-0.005*** (-5.656)
MARKET-TO-BOOK_RATIO	-0.003** (2.570)	-0.001** (-2.544)
PAST_12_MONTH_RETURNS	-0.003 (1.060)	0.005*** (4.122)
AVERAGE_SALES_GROWTH	-0.006 (1.040)	<0.001*** (4.615)
BOOK_LEVERAGE	0.022*** (3.940)	0.005 (0.754)
NONCASH_WORKING_CAPITAL	0.016 (1.600)	-0.006 (-0.758)
FIRM_AGE	0.006** (2.720)	<0.001** (2.481)
EXCESS_CASH	0.001*** (4.930)	<0.001 (-0.182)
DEAL_RATIO	0.001 (1.440)	0.024*** (23.766)
STOCK_DUMMY	-0.002 (0.660)	-0.006* (-1.830)
CASH_DUMMY	0.008*** (4.360)	0.013*** (3.917)
HIGH_TECH_DUMMY	-0.007** (2.210)	-0.004 (-1.341)
DIVERSIFYING_DUMMY	-0.001 (0.440)	-0.003 (-1.058)

Continued

Replication Table IA13

Continued

	Column (3)	
	Published	Replicated
HOSTILE_DUMMY	-0.020	-0.021**
	(1.390)	(-2.128)
PUBLIC_TARGET_DUMMY	-0.017***	-0.027***
	(4.720)	(-8.898)
CHALLENGE_DUMMY	0.015**	-0.010
	(2.300)	(-1.358)
TARGET_INDUSTRY_M&A_INTENSITY	0.001	
	(0.570)	
Intercept	0.012	-0.032
	(0.740)	(-0.487)
No. of obs.	6376	6674
Adj. R2	0.030	0.110

Replication Table IA14

Replication of Roosenboom, Schlingemann, and Vasconcelos (2013), Table 2, Column (3).

The table reports estimation results from linear regression models examining the relation between the market reaction to acquisition announcements and acquiring firm stock liquidity. The dependent variable is the acquiring firm's cumulative abnormal return, which is aggregated over the 5-day trading window beginning 2 trading days prior to and ending 2 trading days following the merger announcement. All variables are as defined in Roosenboom, Schlingemann, and Vasconcelos (2013). Replication code written in the foghorn declarative econometric language and the transpiled SQL data manipulation code are available on Github. *p*-values are reported in brackets. Coefficients displayed in bold represent the key economic variable of interest in the paper. Coefficients marked with ***, **, and * are significant at the 1%, 5% and 10% level, respectively.

	Column (3)	
	Published	Replicated
Stock Liquidity	**-0.038***** **[0.006]**	**-0.006***** **[<0.001]**
Payment = Equity	-0.008*** [0.005]	-0.016*** [<0.001]
Target = Public	-0.024*** [<0.001]	-0.026*** [<0.001]
Target = Subsidiary	0.010*** [0.001]	0.006* [0.061]
Relative Size	-0.002 [0.800]	0.005 [0.418]
Total Assets	<0.001 [0.996]	0.001 [0.560]
Leverage	<0.001 [0.997]	0.022* [0.067]
Market to Book	0.001 [0.514]	-0.015 [0.108]
Cash Flow	0.018 [0.401]	-0.003 [0.840]
Observations	3815	4189
Adjusted-R^2	0.057	0.074

Replication Table IA15

Replication of Wang and Xie (2009), Table 5, Column "TCAR".

The table reports estimation results from linear regression models examining the relation between the market reaction to acquisition announcements and shareholder rights. The dependent variable is the acquiring firm's cumulative abnormal return, which is aggregated over the 11-day trading window beginning 5 trading days prior to and ending 5 trading days following the merger announcement. All variables are as defined in Wang and Xie (2009). Replication code written in the foghorn declarative econometric language and the transpiled SQL data manipulation code are available on Github. *t*-statistics are reported in parentheses. Coefficients displayed in bold represent the key economic variable of interest in the paper. Coefficients marked with ***, **, and * are significant at the 1%, 5% and 10% level, respectively.

	Column "TCAR"	
	Published	Replicated
Shareholder-rights difference	**0.836*****	**0.594****
(Target index - bidder index)	**(3.480)**	**(2.170)**
Bidder Characteristics:		
Log(market cap)	2.084	0.582
	(1.250)	(0.207)
Tobin's Q	-0.543	-0.589
	(-0.570)	(-0.539)
Leverage	-10.512*	-13.799***
	(-1.710)	(-2.590)
Return on assets (ROA)	35.827***	38.425***
	(2.660)	(2.866)
Target Characteristics:		
Log(market cap)	-3.985**	-6.677**
	(-2.480)	(-2.194)
Tobin's Q	-0.114	-0.624
	(-0.100)	(-0.649)
Leverage	-1.101	4.273
	(-0.170)	(0.876)
Return on assets (ROA)	-12.927	-9.049
	(-1.200)	(-0.854)
Deal Characteristics:		
Market cap ratio	-5.770	-8.000***
	(-1.590)	(-2.623)
Tender offer	9.859**	5.007*
	(2.550)	(1.833)
Diversifying acquisition	3.873	3.232
	(1.500)	(1.579)
All cash deal	2.022	2.059
	(0.440)	(0.782)
Merger of equals	-9.446***	-0.385
	(-3.990)	(-0.100)
High-tech combination	0.042	-1.284
	(0.010)	(-0.574)
Number of Obs.	396	378
Adjusted R2	0.216	0.190