arXiv:2602.03863v1 [cs.DL] 27 Jan 2026

克服计算可复现性的障碍

Roman Hornung[1,2], L´aszl´o N´emeth[3,4,*], Oleksandr Zadorozhnyi[5], Theresa Ullmann[6], Michael Kammer[6,7], Rebecca Killick[8,9], Christopher J. Paciorek[10], Julien Chiquet[11], Moritz Herrmann[2,12], Lucija Batinovi´c[13,14], Rickard Carlsson[14], Pierre Neuvial[15], Boris Hejblum[16], Julia Wrobel[17], Anne-Laure Boulesteix[12,2], 和 Karsten Tabelow[3]l统计系，慕尼黑大学（LMU），德国慕尼黑[2]慕尼黑机器学习中心（MCML），德国慕尼黑[3]魏尔施特拉斯应用分析与随机研究所，德国柏林[4]马克斯·普朗克人口研究所，德国罗斯托克[5]计算机科学系，慕尼黑工业大学计算、信息与技术学院，慕尼黑工业大学，德国慕尼黑[6]临床生物统计学研究所，医学数据科学中心，维也纳医科大学，奥地利维也纳[7]肾脏病学与透析科，内科三科，维也纳医科大学，奥地利维也纳[8]数学科学学院，兰卡斯特大学，英国兰卡斯特[9]数学与统计科学学院，克莱姆森大学，美国克莱姆森[10]统计系，加州大学伯克利分校，美国[11]巴黎-萨克雷大学数学与计算机应用联合研究组，法国国家农业、食品与环境研究院，巴黎高科农业学院，巴黎-萨克雷大学，法国帕莱索[12]医学信息处理、生物统计与流行病学研究所，医学院，慕尼黑大学（LMU），德国慕尼黑[13]行为科学与学习系，林雪平大学，瑞典林雪平[14]心理学系，林奈大学，瑞典韦克舍[15]图卢兹数学研究所（IMT），法国国家科学研究中心，图卢兹大学，法国图卢兹[16]SISTM，波尔多人口健康研究中心 U1219，波尔多大学 / 法国国家健康与医学研究院 / 法国国家信息与自动化研究所，法国波尔多[17]生物统计与生物信息学系，埃默里大学，美国亚特兰大[*]通讯作者：L´aszl´o N´emeth，nemeth@wias-berlin.de

2026年2月5日

摘要

计算可复现性，即独立研究人员能够精确复现已发表实证结果的可能性，是科学的基础。尽管其重要性不言而喻，但以可复现性为目标的研究论文比例仍然偏低，且在不同学科间分布不均。障碍

包括研究人员和期刊普遍认为缺乏激励、准备可复现材料面临实际困难，以及期刊缺乏协调统一的可复现流程与要求标准。现有指南往往技术性很强，主要面向已经从事可重复研究的人士。本文首先综合了有关可复现性对作者和期刊双方益处的证据。基于我们在多家期刊开展可复现性检查的丰富经验，接着提出简明务实的指南，帮助不同学科创建可复现分析。我们还审视了部分期刊当前的可复现性政策，展示其要求和程序存在的显著差异。鉴于这一情况，我们提出了构建可复现性标准统一多层级体系的概念基础，以支持不同期刊和研究社群开展透明、一致的评估。作为期刊（可复现性）编辑和 MaRDI 倡议的参与者，我们的目标是鼓励更广泛地采用可复现实践，尤其是通过降低作者和期刊面临的实际障碍来实现这一目标。

1 引言

计算可复现性是科学严谨性和可信度的一项基本原则。在本文中，我们将计算可复现性定义为：独立研究人员能够在原作者完全不参与的情况下，准确复现分析最终结果的可能性。需要注意的是，这一领域的术语在不同学科间并不完全一致：在某些学科中，我们所说的计算可复现性被称为可重复性。相比之下，根据美国国家科学、工程与医学院提出的术语， [1],可重复性是指使用新收集的数据开展一项新研究，并获得与先前研究一致的研究发现。

确保计算可复现性不仅需要提供经过分析的数据，还需要提供分析所用的完整编程代码（如果使用点选式软件，则需提供相应的分析文件或工作流程），包括软件环境的所有规格说明。计算可复现性对于各种数据用途都至关重要，包括完全开放的数据和受限数据。越来越多的数据集按照 FAIR 原则——可发现、可访问、可互操作和可重用 [2]——进行整理；这些原则为元数据标准以及如何有效使用和重复使用数据提供了指导。独立验证的实际可能性自然会因数据可访问性的不同程度而异。

本文始终聚焦于分析真实数据、模拟数据或两者兼有的研究中的计算可复现性，不受学科限制。因此，我们关注所有研究领域中结果取决于对真实数据或模拟数据进行计算分析的研究。这不包括涉及实验室、实地或定性研究环境的可复现性形式。

在一项开创性工作中，Claerbout 和 Karrenbach [3] 强调了通过电子文档将发表内容与底层分析联系起来，以确保可复现性的价值。Buckheit 和 Donoho [4]后来指出，呈现基于数据的分析结果的科学论文，不过是底层研究的宣传材料，而代码和数据才构成真正的科学工作。事实上，涉及计算研究的科学论文若没有生成其结果的代码和数据，便无从存在。文章对于向科学界传达结果不可或缺，但研究发现的有效性取决于底层代码和数据，而非手稿本身。

然而，Buckheit 和 Donoho 提出的概念 [4] 本身并未解释为何应公开提供代码和数据。我们可以借鉴纯数学来论证发表代码和数据的合理性：在纯数学中，研究发现以通过证明确立并传达的陈述为形式，而且不向公众公开这些证明是不可想象的。在实证研究中，图或表格等结果也应由可获取的代码和数据作为支撑，从而能够接受验证。科学界有理由认为，也越来越期待已发表研究能够得到验证，且其中的陈述、论断和结果均有效并经过严格核查，以便进一步研究能够建立在已发现的事实和解决方案之上，并

有把握地取得成果。除了作为研究结果的实证基础——或“证明”——这一基本作用外，发表代码和数据还能为研究人员和期刊带来诸多益处，我们将在第2节讨论这些益处。

近几十年来，在一些学科中，旨在通过提供必要材料来实现可复现性的实证研究文章（即基于数据分析的文章）所占比例有所上升，但在另一些学科中则没有；若从整个科学文献来看，这一比例仍然很低 [5–8]。此外，即便提供了可重复性材料，相应的已发表结果往往仍不可复现 [9]。在实证机器学习等某些数据密集型领域，相关实践已更接近 Donoho [10] 所称的“无摩擦可复现性”——将代码、数据和计算环境组织起来，使分析能够以最少的努力得以复现的工作流程（例如，OpenML 等平台 [11]提供了支持）。然而，这类社群仍是例外，而非常态。除了许多情况下基础数据无法公开可用（无论是由于法律、伦理原因，还是符合 FAIR 原则的访问限制）这一事实之外，我们还指出了导致可复现研究占比持续偏低的三个主要因素。

第一个因素是缺乏确保可复现性的激励，这既涉及个体研究人员，也涉及期刊。对研究人员而言，要使分析完全可复现，需要投入额外的时间和精力；而在当前的学术奖励体系中，这项工作往往得不到充分认可 [12]。因此，研究人员通常认为，发表自己的代码和数据几乎没有直接的个人收益，尽管这样做确实有各种利己的好处。许多科学家呼吁期刊强制要求所有实证研究发表代码和数据补充材料 [13–18], ，同时也有人强调，应更系统地评估计算可复现性 [19]。然而，期刊往往未能认识到，要求发表此类材料也有利于自身。因此，只有部分期刊会为补充代码和数据的准备及发表提供指导（见第4.1节）。

第二个因素是，许多研究人员觉得，确保可复现性的实际要求令他们不堪重负。更多地关注这一主题或许能减轻这种感受，但与撰写科学论文相比，目前大多数研究人员很少关注可复现性。尽管现有相关文献总体质量很高，但内容往往十分详尽，对于缺乏背景知识的读者来说也常常难以理解。因此，这些文献主要触及那些已经熟悉或对该主题持积极态度的人。很少接触过可复现性的研究人员，不太可能查阅、更不用说遵循关于编写整洁代码或创建可复现分析的冗长技术建议。我们在不同期刊环境中开展可复现性审查的经验也印证了这一观察：虽然一些期刊收到的投稿中，初始材料越来越多地已符合基本的可复现性标准，但作者仍普遍难以满足可复现性要求，即使期刊提供了结构化指导或检查清单也是如此。

第三个因素是可重复性制度化不足，这涉及期刊及科学出版中其他参与者所发挥的作用。在已经采取措施促进可重复性的期刊中，相关要求和程序差异很大。这种异质性使各期刊难以一致地实施可复现研究实践，也导致相似的作品因发表地点不同而受到截然不同的评价。此外，过于复杂或协调不善的程序会增加期刊进行代码检查的工作量，而含糊不清的指南在实践中则收效甚微。总体而言，系统层面缺乏协调且实用的结构，无法将可重复性确立为出版流程不可或缺的一部分。Agricola 等人 [20]探讨了数学科学中由指标驱动的出版和欺诈性出版等更广泛的问题。还值得一提的是，期刊指定可重复性编辑所作出的积极贡献。

在本文中，我们通过综合现有证据，并结合实践指导与概念性建议，应对上文所述的三个因素，以此作出贡献。我们的目标是降低个体研究人员实现可复现性的障碍，同时概述期刊和

机构如何支持更加一致且务实的实践。本文主要面向作者和期刊编辑，我们力求使建议对计算专业能力水平各异的读者都易于理解，因为即便经验丰富的研究人员，在创建完全可复现分析时也可能遇到挑战。

首先，我们根据文献记载，总结可复现性对个体研究人员和期刊的益处。其次，我们为作者提供简明易懂、便于遵循的说明，以确保可复现性。这些说明源于我们审查数百篇文章的代码和数据补充材料的经验；这一过程揭示了作者常见的陷阱和反复出现的困难。这些经验反映了我们作为多本期刊的（可复现性）编辑所积累的共同参与经历；这些期刊已建立可复现程序，具体包括 Biometrical Journal、Computo、Journal of Statistical Software、《美国统计学会杂志》和 Meta-Psychology。这些说明旨在通过让可复现实践即使对几乎没有相关经验或最初并无确保可复现性动机的人员也易于采用，来推动其更广泛的应用。

第三，我们讨论期刊和机构如何通过更清晰且切实可行的框架条件来促进可复现性。我们总结了多本期刊当前的可复现实践，这凸显出现有指南之间存在显著差异。在此概览的基础上，我们概述了更加一致且结构化的可复现性标准的概念性构想，其中一部分借鉴了我们作为数学研究数据倡议（MaRDI）参与者的经验。这类标准有助于期刊更透明地传达期望，并支持在不同研究和学科之间更加统一地评估可复现性。我们将其作为科学出版未来发展的起点，而非强制性的标准。

2 可重复研究对作者和期刊的益处

研究人员和期刊往往认为，采用可重复实践只能带来有限的自身利益，这可能会降低他们在这一领域投入精力的动力。如果能增强研究人员对激励的认知，那么即使没有严格的期刊政策，可重复结果的比例也可能大幅提高。同样，如果期刊更清楚地认识到实施可重复性政策的优势，就可能会更广泛地采用此类政策，从而进一步促进可重复研究。重要的是，此类政策不一定要包含全面的可重复性检查；其范围十分广泛，既可以是简单的要求，例如要求提供代码，并在可能的情况下提供数据，也可以是更全面的评审程序。即使是轻量级措施，也能提高透明度，并为可重复研究作出实质性贡献。

在以下小节中，我们将介绍对作者和期刊的益处，其中许多益处适用于这两类群体。

2.1 对研究人员自身工作流程和未来工作的益处

对于个别研究人员而言，确保可复现性为他们今后的工作带来多项实际优势，例如更便于重用（更清晰的）代码、更确信结果正确无误，以及减少研究浪费。研究人员通常是自己代码未来的主要使用者，因此最能从代码的可用性和结果的可复现性中受益。公开分享代码的可能性会促使人们更加关注代码质量 [21], ，进而便于日后浏览和修改分析，例如在稿件修订期间或在后续项目中拓展这项工作时 [22]。考虑到从最初分析到发表往往间隔相当长的时间，记录在案的附加软件包版本至关重要。完善的文档可确保日后修改分析时仍能精确复现，因为软件包的功能可能已经发生变化。整洁且文档齐全的代码也有利于研究团队的其他成员 [23]，使他们能够更高效地开展进一步分析，并在同一代码库上协作。始终公开发表项目代码和数据的研究人员，今后也能安全地访问这些内容，无论他们是否更换雇主或电脑 [24]。

公开提供代码和数据也可能减轻一些作者感受到的心理负担 [25]。尽管作者仍须对其工作的完整性承担全部责任，但分享代码可供审稿人检查；一些作者可能会将此视为一种安心和支持 [15, 17]。相反，另一些人可能会担心他人对代码质量的评价，以及潜在错误被发现，因此这种心理影响很可能因人而异。

分享代码和数据表明作者对自己的工作充满信心，并能吸引合作者以透明且互惠互利的方式，在共享材料的基础上开展工作。即使他人独立重新使用这些代码，原作者仍然是相关工作的基础组成部分，通常也会得到致谢，这不仅不会削弱其贡献，反而会进一步凸显其贡献。相比之下，一些研究人员担心他人可能会使用自己的代码和数据，开展分析或开发原本由自己计划的扩展工作。然而，这类担忧在实践中很少成为现实：风险较低，因为原作者最熟悉相关分析，并且对研究背景保有深入而具体的了解 [24]。

2.2 科学影响力与职业收益

可复现性还为科学质量和研究人员的职业生涯带来诸多优势。当作者更加重视代码质量，并在发表前审阅其材料时，在文章发表前发现错误的可能性会大幅提高 [24]。相反，日后因分析错误而发布的撤稿声明可能会扰乱知识构建的迭代过程，并给作者和期刊带来严峻挑战 [15, 17]。可复现性政策有望帮助降低撤稿风险。通过促进代码透明度，以及在某些情况下由期刊开展发表前的可重复性检查，此类政策提高了在发表前发现分析错误的可能性，尽管期刊能够切实提供的可重复性检查程度会因其资源而有所不同。

方法学文章通常会提出复杂的算法。如果没有实现代码，这些算法在实践中很难应用，从而可能大幅降低其影响力和引用数量 [26]。此外，将其纳入比较研究也会变得困难得多，进而进一步限制其可见度。在此背景下，不应要求作者重新实现那些没有可用代码的方法；由于原始工作中的方法学描述不够详细，此类重新实现容易出错，也会给开展比较研究的人员带来不合理的负担。

易于获取的代码会增加其他研究团队在已发表算法的基础上进行开发或扩展的可能性，从而扩大其在方法学界的影响力。此外，相较于仅有理论描述的方法，即使后者颇具前景，清晰整洁的实现也更容易被采用并在其基础上继续开发，尤其是在缺少实现细节或方法较为复杂的情况下 [27]。

主要关注应用所提出方法的人员，往往更加重视实现，而不是论文中的理论描述。这些人员经常就应用方面的问题联系作者。此类咨询能够为实践中的常见困难提供宝贵见解，并促成错误的纠正和方法的改进 [27]。

发布代码和数据能够带来更多引用的一个重要原因是，引用被重复使用的代码已经成为惯例——这不仅适用于计算领域，尤其适用于应用学科；在这些学科中，由于实现底层算法可能颇具挑战，研究人员往往依赖现成的实现。此外，当补充材料上传到分配 DOI 编号的代码仓库时，引用次数往往更高 [15]。总体而言，已有多项研究反复表明，公开提供其代码 [28, 29]和/或数据 [24, 30, 31] 的论文会获得更多引用。H¨offler [32] 还针对经济学期刊表明，在要求可复现性的期刊上发表的文章，其被引用的频率在统计上显著更高；这一结论已考虑期刊之间的差异及总体时间趋势。

将所提出的算法实现为托管在成熟平台（如 GitHub、R 的 CRAN 或 Python 的 PyPI）上的软件包，与仅提供

方法学论文的代码和数据补充材料中的实现相比，这种做法能进一步提升算法的实际适用性。因此，基于软件包的实现也可能对引用次数产生积极影响。在相关背景下，Jalili 等人 [33] 指出，生物医学软件包纳入包管理系统后，其引用次数显著增加。

资助机构正通过要求开放数据和代码、强制制定数据管理计划，并且在某些情况下甚至承担长期数据存储成本，赋予可复现性日益重要的地位 [24, 34]。因此，作者可以通过始终确保其分析具有可复现性，并在资助申请中突出这一点，来提高获得资助的机会。

发表可重复结果也能提高研究人员的可信度和可靠性[35]。这向学术界传达了科学严谨性和透明度，并可能对同行评审流程产生积极影响 [22]。有时，审稿人可能会亲自运行共享代码，并在发现合理问题时形成批判性看法。从科学视角来看，这类情况很有价值，因为它们有助于提升已发表工作的质量。然而，在实践中，这种情况似乎相对少见，不应因此阻止作者分享其材料，尤其是考虑到对于执行良好的工作而言，透明度能带来显著的可信度提升。

2.3 期刊的具体益处

正如 Stodden等 [14]所讨论的，可重复性政策对期刊和更广泛的研究生态系统具有重大益处。首先，这些政策使期刊能够发挥质量标志的作用，表明其致力于科学严谨性，并增强自身声誉。这类政策还可以为作者提供激励，因为作者可能会受益于在公开认可严谨且透明研究实践的期刊上发表论文。事实上，Stodden等 [14]发现，影响因子较高的期刊更有可能采用此类政策，尽管因果关系的方向仍不明确。

这会提升期刊对投稿作者的吸引力，使其更有底气要求实现可复现性——尽管并非所有期刊都意识到了这一潜力。反过来，对质量提出此类要求的期刊在作者眼中可能显得更为严格，从而更具吸引力，即使其影响因子并不特别高。

明确的可复现性政策——即使只要求作者提交代码，并在可能的情况下提交数据，而不进行可重复性检查——也能为期刊带来双重运营效益。首先，它可作为质量过滤器，通过劝退那些不太重视严谨性或透明度的作者来提升投稿质量。其次，明确规定要求可以降低收到格式不合要求的投稿的可能性。

3 制定可复现分析的简明指南

3.1 一般考量

如引言中所述，我们的目标之一是提供简洁、易于遵循的指南，尽量降低实施这些指南所需的投入。对于希望了解更多有关良好编码实践、进阶技术或技术细节的读者，我们建议参阅现有文献（例如 [36], [37], [17]）。 [38] 中列出了一套主要聚焦于数据的最低建议。我们的经验表明，即使是可重复研究中看似显而易见的方面，在实践中也常常被忽视。因此，我们决定在此将这些方面纳入其中，以帮助避免常见问题。还请注意，许多方面并非确保可复现性的硬性要求，但能让尝试复现结果的过程容易得多。虽然其中许多方面都有更复杂的解决方案，但此处提出的建议已为以务实、易于采用的方式创建可复现分析打下了坚实基础。

我们的建议提供了适用于各学科的通用基础。在条件允许时，特定领域的标准（例如心理学领域的 PSYCH-DS）以及地方或国家指南可以提供更多针对具体领域的指导。这些资源还可以帮助作者按照其研究社群的惯例整理材料。

从研究项目一开始就应用以下建议非常有益，因为其中许多建议如果事后再实施，耗时会大幅增加。重要的是，分享不够完善的代码和数据仍然胜过完全不分享，即使并未遵循所有建议。我们还提供了针对 R 和 Python 的具体提示，但这些建议同样可以类似地应用于其他编程语言。

准备代码和数据补充材料时，偶尔从不了解相关研究的人的角度来审视材料可能会很有帮助。这样更容易发现其中可能含糊不清，或在缺乏研究详细知识的情况下难以使用的部分。尽管这些指南主要面向准备代码和数据补充材料的作者，但也可以作为审稿人或可重复性编辑的有益参考。

在详细介绍指南之前，我们在图1 中提供了一份核对清单，简要概括主要建议。请注意，如果不阅读详细指南，核对清单中的所有要点可能都无法完全理解。核对清单主要用于快速初步参考，以帮助读者确定下面详细指南中哪些方面可能值得进一步阅读，因为建议与个别读者的相关性取决于其对可复现研究实践的熟悉程度。我们还在补充材料中提供了核对清单的单页 PDF 版本，该版本采用便于打印的格式。

3.2 结构与README

代码和数据补充材料应包含独立复现相关手稿中所呈现结果所需的全部材料。这通常包括完整的分析代码、相关数据集（在可能的情况下）以及配套的代码簿（用于解释数据集中所有变量的文件）和 README 文件。代码和数据补充材料应采用简单的文件夹结构，并使用清晰的目录名称，例如 data、code 和 results；后者包括 intermediate（用于中间结果，见下文）、figures 和 tables 等子文件夹。为保持文件结构清晰且易于管理，相关功能应整合到单个文件中，而不是分散在多个用途相同的文件中。

README 文件至关重要。它应简要概述代码和数据补充材料中的所有文件，并说明为复现文章和补充材料中呈现的最终结果（图和表）而必须执行脚本的顺序。应避免对分析设计作复杂描述，因为这可能使不熟悉研究细节的读者感到不知所措。相反，只需说明每个脚本生成哪些图和表，并引用其相应编号（例如，图1、表1）。

为了实现长期可复现性，还必须清晰记录软件环境，包括所使用的所有额外软件包的版本。在 R 中，可以通过将所有相关软件包加载到工作区，并将 sessionInfo() 的输出复制到 README 文件中来轻松实现这一点。在 Python 中，session-info 软件包会生成类似的输出，其中包含工作区环境信息。或者，也可以通过提供环境规范文件（例如，在 R 中使用 renv，或在 Python 中使用 conda）进一步支持长期可复现性，从而实现软件环境的自动还原。

包含变量名称、变量描述、变量类型（例如数值型）以及变量可能取值的代码簿，并非计算可复现性所严格要求的内容。不过，强烈建议提供代码簿，因为它能显著提升透明度，促进对数据的理解和复用，并符合既定的开放数据原则。这些信息还有助于检查潜在的错误。

补充材料的 A 节中提供了一个代码和数据补充材料及 README 文件的适当结构示例，并附有更多详细信息。

创建可复现分析的核对清单

1. **结构与 README**

   □ 使用清晰的文件夹结构（例如包含子文件夹的 data/、code/、results/）。□ README 中包含所有文件及准确的执行顺序。□ 在 README 中记录软件环境和软件包版本（例如，R: sessionInfo()；Python：session-info），或者通过环境文件记录（例如 renv、conda；有助于更轻松地实现长期复现）。□ 在 README 中注明哪些脚本用于生成哪些图/表格。□ 将相关功能整合到单个文件中。

2. **将代码与报告结果关联**

   □ 所有图/表均由脚本生成。□ 每个图/表的创建位置在代码注释中标明。□ 输出文件包含在补充材料中（results/figures/、results/tables/）。□ 手稿中的图/表编号体现在输出文件名中。

3. **代码组织与风格**

   □ 结果通过运行脚本无需手动编辑即可生成。□ 冗长的函数定义外置到单独的脚本中，并在需要时引入。□ 仅使用相对路径。□ 使用有意义的文件名；用下划线或驼峰命名代替空格。□ 格式、缩进与空格清晰；平台无关代码。

4. **随机性与并行计算**

   □ 使用确定性（并行安全）的随机数生成方法。□ 每次复现均使用独立的随机数流，以便实现逐次复现。

5. **中间结果与抽查**

   □ 保存分析的中间结果 > 1 小时运行时间。□ 可进行可复现性抽查。□ 使用不同的脚本来生成和评估中间结果。□ 提供分析脚本运行时间的粗略估算。

6. **无法共享真实数据时**

   □ 包含能够模拟真实数据关键属性的合成数据。□ 除基于原始数据的结果外，还提供基于合成数据的结果，以便进行可复现性检查。□ 说明无法共享原始数据的原因。

图1：创建可复现分析的核对清单。

3.3 将代码与报告结果相连接

文章和补充材料中包含实证结果的所有图和表格都应可复现。应使用注释说明各代码部分的用途，尤其应明确指出每个图或表格的生成位置，并注明相应图和表的编号。

代码应生成所有图和表格，这些输出文件也应包含在代码和数据补充材料中。保存的表格结构（行列顺序以及数字的任何舍入方式）应与文章中的对应表格一致，这样更便于将复现的表格与手稿中的表格进行比较。理想情况下，文件名应与文章中使用的名称相对应（例如，figure 1.pdf、table 1.csv）。

3.4 代码组织与风格

从根本上说，运行相应的脚本就应能得到最终结果，而无需手动修改代码。生成结果的分析脚本应避免包含冗长的函数定义，除非项目本身较为简单；对于超出基础分析范围的内容，应将此类函数整理到单独的脚本中，并在分析脚本中调用，以提高代码的清晰度。对于自行编写的关键函数，应简要说明其输入和输出参数。

在文件中，只应使用相对于根文件夹定义的相对路径。所有文件都应使用有意义的名称，避免使用空格，并改用下划线或驼峰命名。

代码应易于阅读，逗号后、运算符两侧应留有空格，并采用规范的缩进。可以使用代码检查工具或其他代码格式化工具自动完成这类格式设置。在 R 和 Python 中，formatR 和 Ruff 等库分别提供了这些功能。

代码应尽可能少地依赖运行环境。尤其是，只要可行，就应确保代码平台无关（除非有充分理由，例如使用基于 fork 的并行化），并且不依赖特定软件（例如，避免依赖特定于集成开发环境的 API，因为这类 API 要求在 RStudio 中执行 R 代码）。

最后，代码应尽可能保持简单，并在可能的情况下只依赖少量广泛使用的软件包。这样可以减少安装依赖项的需要，尽量避免潜在冲突，并促进长期可复现性。

3.5 随机种子与并行计算

对于包含随机元素的分析（例如统计模拟），必须将随机数生成器的种子设为一个任意但固定的值，以确保结果完全可复现。例如，在 R 中可以使用命令 set.seed(1234) 设置种子；在 Python 中则可以使用 numpy.random.default_rng(1234)。

然而，在并行计算中，设置单个全局种子通常并不足够，因为并行工作进程可能会生成各自的随机数流。为了获得可重复结果，建议依赖能够明确为每次复现创建并行安全的确定性随机数流的软件机制。许多现代环境都提供此类功能，例如 R 中的 future 生态系统（通过 future.apply 等软件包中的种子处理选项，或将 foreach 与 doRNG 结合使用），以及 Python 中基于 NumPy 的 SeedSequence 的随机数生成。这些工具旨在降低随机数流重叠的风险，并让用户更轻松地在并行环境中正确使用随机数。

使用此类机制可确保每次重复实验都与其自身确定性的随机数流相关联。因此，结果不会受可用 CPU 核心数量的影响，分析可在不同系统或使用不同并行化后端时复现；如有需要，也可以按顺序重新计算各次重复实验。重要的是，这使得检查特定重复实验的可复现性变得简单，下一小节将对此作出说明。A

补充材料的 B 节提供了一个简单的 R 和 Python 示例，演示如何实现此类并行安全随机数流，以及如何复现各次重复实验。

3.6 中间结果与可复现性抽查

对于运行时间超过约一小时的计算密集型分析，保存中间结果很有帮助。在模拟研究中，这些中间结果包括针对每个模拟数据集获得的原始结果。虽然中间结果并非实现可复现性的严格要求，但它们能使过程更加灵活，也更容易复现结果。应为每个分析脚本提供大致的运行时间，并简要说明所使用的硬件（例如“$\sim$ 5 分钟，在标准笔记本电脑上”）。

保存中间结果主要有两个目的。首先，无需重新运行整个分析，即可根据存储的中间结果快速生成表格和图。这既有利于复现结果的人员，也有利于原始代码作者，例如日后需要调整呈现方式时。其次，通过适当处理随机数生成，尤其是在使用并行计算的分析中，中间结果可以实现所谓的可复现性抽查。在此类抽查中，只需重新计算少量任意选取的部分结果（例如模拟研究中的单个重复实验），并将其与相应存储的结果进行比较。如果这些结果完全一致，则抽查成功，可以据此推断完整分析很可能是可复现的。

确保各个部分结果能够复现的一种直接方法，是为每次重复实验使用可复现的随机数流（见上一小节）。这样，任何复现该分析的人都可以重新运行选定的重复实验，并将新获得的结果与已存储的结果进行比较。除了进行可复现性抽查之外，这种方法还可以更详细地检查个别异常或离群结果，以探究这些异常背后的原因。

生成中间结果的代码应易于识别，并且能够独立运行。这样，在需要时就可以轻松复现各个中间结果。

要使计算密集型分析更易于复现，另一种易于实施的策略是：使用少得多的重复实验次数（例如 10 次，而不是 1,000 次）重新运行分析，并将相应的结果与完整结果一并提供在代码和数据补充材料中。进行可复现性检查的任何人，都可以使用较少的重复实验次数，验证自己是否获得了与补充材料中所列结果相同的结果。至关重要的是，较小规模的运行应使用与大规模运行相同的代码，只是减少重复实验次数。这样，如果结果匹配，就能为完整分析具有可复现性提供有力证据。

为使工作流程更易于处理，应将生成中间结果的代码放在与评估这些结果并生成最终图和表格的代码不同的脚本中。

3.7 无法共享原始数据时的可复现性

在许多情况下，分析所使用的数据无法与代码一同公开，因此无法直接验证基于真实数据的结果的可复现性。为此，可以生成能够模拟真实数据的计算结构、但不一定模拟其统计结构的合成数据（伪数据）。在此情境下，这已经完全足够，因为合成数据仅用于可重复性检查，而非用于实质性推断。应将这些合成数据包含在代码和数据文件夹中。例如，可以在 R 中使用 synthpop 或 simdata 软件包生成此类合成数据。

为了确保可复现性，仍然重要的是，将使用真实数据生成的结果与基于合成数据得出的相应结果一并提供，并且仍使用同一脚本，仅更改数据对象。希望检查可复现性的人随后可以尝试复现使用合成数据得出的结果。如果复现成功，那么也很可能

相应　　基于真实数据得出的结果，若能获取真实数据，便可复现
。

4 现有的可复现性政策与协调统一标准的视角

本节首先概述所选期刊当前的可复现实践。然后，我们讨论协调标准可能带来的益处，并概述可为制定此类标准提供参考的概念维度。这里所说的标准，是指用于跨期刊和学科评估可复现性的共同准则，而非对可实现的可复现性水平提出统一要求。

4.1 采用可重复性政策的部分期刊当前的实践

表格 1 概述了部分期刊的可复现实践。这些期刊被选作成熟可重复性政策的示例。该列表并不全面，但反映了不同领域现有的多种做法。正如引言中所述，我们作者团队的几位成员参与了这些期刊的可复现流程，这为了解其程序提供了额外的实践视角。

尽管这些期刊采取的做法各不相同，但都已实施具体措施来提升计算可复现性。这些措施涵盖了不同程度的要求：从负担较轻、主要要求对分析进行充分描述的政策，到要求更高的程序，即由指定编辑或审稿人主动核查所有已呈现结果的可复现性。

《内科学年鉴》要求对方法进行充分详细的描述，以便实现可复现性，并倡导同时提供注释完善的源代码。

Biometrical Journal 主要面向方法学统计学家，他们大多使用 R。稿件需经过常规同行评审流程，可能被拒稿，也可能有条件录用。有条件录用后，将由一支由四名可重复研究编辑组成的专门团队开展可重复性审查。期刊建议作者遵循一系列指南，以满足期刊的可复现性要求。可复现性分为不同类别（例如，完全可复现以及若干部分可复现类别）。

《英国医学杂志》、《计算与图形统计学杂志》和《技术计量学》均已要求作者提交所有研究论文的分析代码。代码必须作为补充文件提供，最好也存入 GitHub、GitLab、开放科学框架或 Zenodo 等开放存储库。虽然这些期刊不开展正式的可重复性审查，但提交的代码应可供机器读取、注释清晰，并能使用标准统计软件运行。编辑和读者可在需要时检查代码，作者也应记录软件版本及相关软件包。有趣的是，《技术计量学》仅要求一个表格或图具有可复现性。

Computo 要求通过基于可执行笔记本、虚拟环境和持续集成的集成工作流程实现数值可复现性。与传统期刊不同，Computo 的文章直接根据底层代码和环境生成，因此计算工作流本身也是一项重要的贡献组成部分。如果分析的某些部分必须在外部运行，作者必须同时提供中间结果，以及将这些结果导入笔记本所需的代码。生成结果所需的所有代码，包括用于外部运行步骤的代码，都必须提供，并接受同行评审和/或编辑检查。

《统计软件杂志》将完全可复现性作为投稿要求：缺乏完整可复现性的稿件通常会在审稿前退回作者。审稿后还会进行额外的可重复性检查，在满足所有要求之前，稿件将处于有条件接受状态。这包括稿件中报告的图、表以及任何其他输出。该期刊的一个显著特点是专注于软件包；实证分析

表 1：期刊可复现性政策比较概览

期刊	代码要求	数据要求	独立核查	验证通过	时间安排
《内科学年鉴》医学	鼓励	方法描述 only	no	不适用	投稿
Biometrical Journal	yes	开放式或合成的	yes	可复现的研究编辑	最终定稿前录用
《英国医学杂志》	yes	代码仓库首选	no	编辑/读者（审查）	投稿
Computo	是（注：书籍/持续集成）	开放（或 mock 版本，如果受保护）	yes	GitHub Actions	投稿
《计算与图形统计学杂志》图形统计学	yes	代码仓库受到鼓励	no	编辑/审稿人	投稿
Journal of Statistical Software	yes（软件专题）	open	yes	审稿人及复现编辑	首先投稿 & 最终定稿前录用修订
《美国统计学会杂志》	yes（核对清单）	数据信息已提供	yes	副编辑	修订
Meta-Psychology	yes	开放（或有正当理由）	yes	可复现性审核团队（统计编辑）	后-录用
《技术计量学》	是（部分）	not 适用	no	编辑	（有条件的）修订

期刊：正在审查其可重复性政策的学术出版物名称。**代码要求**：表示期刊是否明确要求作者提交研究中使用的分析代码。要求范围从简单的“是”到具体条件，例如要求提供“核对清单”或侧重于“软件”。**数据要求**：关于提供基础数据的政策。要求范围包括要求数据“开放”或至少提供“合成的”数据，或者规定仅需提供数据访问的“方法描述”。**独立核查**：期刊是否由第三方（并非原作者或初始审稿人）主动进行基于执行的审查，以确认结果能否通过计算复现。**验证责任方**：指定负责审查、核查或仔细检查所提交可重复性材料的个人或群体，例如“可重复性研究编辑”、“副编辑”或普通“编辑/审稿人”。**时间安排**：稿件处理流程中提交可重复性材料或进行验证检查的具体阶段，例如“投稿时”、“修订阶段”或“录用后”。

主要仅用于说明。总体指南是，具备足够背景知识的读者应能够在合理的时间内复现结果。软件必须在外部代码仓库中提供，但发表时所使用的版本会与手稿一同保存。

对于《美国统计学会杂志》，如果手稿受邀修订，作者需提供可重复性材料（数据、代码、工作流程）[39]。可重复性由可重复性副编辑进行审查，该期刊目前共有十六名可重复性副编辑。期刊提供了一份作者贡献清单，作者必须完成该清单（可提供 R Markdown 或 Word 文档格式）。此外，模板 GitHub 代码仓库中还提供了一个文件夹结构示例，用于展示研究材料可能的组织方式。该核对清单分为三个部分，必须详细说明：（i）数据相关信息；（ii）软件和代码的详细信息；（iii）有关可重复性工作流程的问题。

Meta-Psychology 要求有条件接收的文章具备完全可复现性。为核查可复现性，可重复性团队十名成员中的一名会开展独立核查；这些成员各自拥有不同领域的专业知识。投稿必须包括开放数据、文档齐全的代码和研究材料，通常通过开放科学框架项目页面提供；只有在有充分理由的情况下才会获准例外。可复现投稿可能会获得开放科学徽章；即使在发表后，作者也应配合复现者或提出方法学问题的读者。对于更复杂的分析，鼓励使用容器来共享计算环境。

上述讨论的期刊仅代表现有举措的一部分。目前，其他已任命专职可重复性编辑的期刊包括 American Antiquity、Information Systems、INFORMS Journal on Data Science、Journal of Archaeological Science、Journal of Economics Surveys、Political Analysis 和 Research Synthesis Methods。这些期刊同样实施了各种政策，以加强实证研究的可复现性。

4.2 迈向协调统一的可重复性标准

如上一小节所示，目前各期刊对作者应如何准备可复现投稿提供的指导差异很大。这些指南从简短、概括性的建议，到详尽的必做事项清单或必须填写的详细模板，不一而足。更完善的可重复性政策有助于开展结构化评估，但也可能让潜在作者望而却步，尤其是在要求耗时较长或政策施加的限制难以满足时（例如，强制要求数据公开）。这种差异使作者难以把握期望，也延长了论文遭拒后改投其他期刊所需的时间。此外，这也使期刊难以向读者和利益相关者传达其可重复性标准。

为应对这一挑战，我们建议考虑制定一套统一的可重复性标准多层级体系。这样的体系将区分可复现性的不同方面和层级。重要的是，这些层级不一定反映所需工作量的差异，而是反映已实现的可复现性方面的差异。分级方案承认可复现性具有多面性，并能在不同研究和学科之间进行一致的评估（例如，Hanselman [40] 用于会议的技术代码审查模板）。在接下来的小节中，我们初步提出若干想法，可作为制定此类可重复性标准多层级体系的基础。其他领域已经存在标准体系，例如用于评估 FAIR 数据的体系（例如，https://www.f-uji.net/index.php?action= 方法）或用于统计软件同行评审的体系（例如，https://stats-devguide.ropensci.org）。

量化可复现性水平可能带来若干益处。目前，科研产出主要通过发表数量、期刊影响因子以及h指数等基于引文的指标来评估。尽管可复现性作为科学价值日益受到认可，但它很少被纳入常用指标。统一标准原则上可为注重可复现性的评估指标或指数奠定基础。此类指标可以纳入

不仅包括不同的可复现性水平，还包括FAIR 原则等既有维度，从而更全面地体现研究透明度。其校准将反映科学界的优先事项，并有助于对研究质量进行更细致的评估。

许多期刊会将文章标记为“可复现”，但这一标记可能代表截然不同的方面，导致读者收到的信号不一致。建立一套清晰且结构化的标准体系，可以明确在特定情境下考量、忽略或能够实现可复现性的哪些方面。这些标准并不意味着各学科必须遵循统一要求，而是提供一个共同框架，以一致的方式描述可复现性。这将有助于期刊更有效、更清晰地传达其可复现性目标和范围，让出版商和编辑委员会受益。更明确的标准还有助于改善与利益相关者的沟通，包括资助机构、数据存储库和研究机构；这些机构日益重视透明度和可复现性。

此外，分级结构可以帮助期刊根据自身可用资源调整可复现性政策。例如，期刊或许可以先明确可用性要求，并对提交材料进行基础检查；之后，在条件允许时，再逐步开展更全面的可复现性审查，并可由专职编辑或审稿人提供支持。

作者也可以从这种标准化且可能可量化的体系中受益。明确的多层级结构能够让作者更快、更准确地评估自己能够合理实现的可复现性水平，并证明其工作达到或超过明确定义的阈值。这既能提升可重复研究在学术生涯中的价值认知，也能提供一个结构化但不具强制性的框架。此外，如果需要将手稿转投其他期刊，统一协调的体系可以减轻负担，避免在向另一家期刊投稿时大幅重构代码和数据补充材料。

总的来说，虽然我们提出的是初步构想，而非最终方案，但我们相信，协调一致的可重复性标准有助于加强沟通、减轻作者和审稿人不必要的负担，并为未来在各科学领域评估可复现性的方式发展奠定基础。

4.3 用于构建可重复性标准的概念维度 s

多层级的可重复性标准体系要求明确哪些方面的可复现性应当进行有意义的区分、评估，并根据其重要性赋予不同权重。虽然我们不提出具体的实现方案，但我们概述了若干概念维度，这些维度原则上可构成此类体系的基础。

我们希望，以下维度能够推动未来围绕如何以透明、系统的方式评估可复现性展开讨论。

- **（A）材料可用性**。这一维度涵盖可复现研究工作流程中的哪些组成部分可供使用。相关要素包括代码、数据、元数据，以及对计算环境的描述。不同级别可以区分例如仅提供代码、提供代码和数据，或在此基础上进一步提供符合 FAIR 原则的结构化元数据。

- **（B）验证范围**。尽管只有对计算可复现性进行明确检查，才能最终确认结果是可复现的，但此类验证可能耗费大量资源，且并非总是可行。不过，更轻量的评估仍可提供有意义的迹象，表明可复现性成立的可能性有多大。因此，这一维度可以区分对代码和数据补充材料所实施验证的深度，其严格程度逐级提高，例如：（i）检查材料的完整性，（ii）评估其质量与易用性，以及（iii）验证部分或全部结果的计算可复现性。

- **（C）验证来源**。根据材料是否经过检查，以及由谁检查，人们对这些检查所体现的可复现性可以给予不同程度的信任。最低程度的信任适用于未进行任何检查的情况。当作者声明他们在投稿前已自行对材料进行了全面审阅时，信任程度更高。这一点还可进一步区分检查是由代码作者还是同一研究团队中的其他人进行的，后者对应更高的信任水平。最高程度的信任适用于材料已在期刊方面接受审查的情况，例如由指定的审稿人或可重复性编辑进行审查。

- **（D）可复现性范围**。不同研究的结果可复现程度各不相同。有些分析能够实现完全的计算可复现性，而另一些则只能实现部分可复现——例如，无法共享原始数据、因而提供合成数据时。不过，结果只能部分可复现还有其他各种原因。例如，如果一项分析的计算成本极高，且没有可复现的中间结果可用于可复现性抽查，那么在缺乏大量计算资源的情况下，这项分析可能无法复现。另一个原因可能是，模拟研究中生成的随机数无法复现（例如，由于随机种子处理方面的问题），此时结果将无法精确复现；不过，如果使用了足够多的重复实验，分析所得的实质性结论仍可能成立。分层系统可以有原则地对这类情况加以区分。

- **（E）代码质量**。代码质量直接影响结果复现的难易程度。我们在第3 节中的说明可为这一维度的指标制定提供依据。重要方面包括代码和数据补充材料的清晰度、注释的详尽程度，以及中间结果是否可用。原则上，可以对这些方面赋予不同权重，并将其汇总为一套细致的评分系统。不过，建立这样的系统需要仔细校准，因此更适合作为长期目标。务实的第一步是定义几个质量等级，以体现有意义的差异，同时保持易于应用。

这些维度并非面面俱到，而是为思考如何在多层级框架中组织可重复性标准提供了概念基础，该框架纳入了可复现性的各个方面和不同程度。我们诚邀学界共同完善并拓展这些理念，使其更具可操作性，从而便于期刊和出版商采用。

5 讨论

5.1 将可重复性标准的多层级体系操作化所面临的挑战

要使我们提出的用于衡量实证研究可复现程度的多层级体系概念切实可行，还需要进行多方面的完善和具体化。例如，必须明确如何呈现不同维度——是采用单一聚合指标，还是以多维度概况呈现——以及如何为不同方面赋予权重。鉴于某些维度相互依存，或并非在所有情况下都可评估，这些选择需要慎重考虑。不过，在我们看来，初步的操作化体系不必做到尽善尽美。即便是一个务实的初版，也能在提高可复现实践的可见度、进而增强人们对其重要性的认识方面发挥重要作用。

正如上一节所述，我们在第3节提出的指导意见，可以作为所提议的第五个可重复性维度——“代码质量”——的初步具体依据。这些指导意见汲取了丰富的代码审查经验，因此非常适合作为制定普遍适用指南的基础。

5.2 期刊有关可复现性检查的政策为何重要

正如我们所提到的，可重复性政策的严格程度存在一个范围。其中最全面、最有效的方法之一，是由期刊自身检查可重复性。然而，这需要审稿人或专门的可重复性研究编辑参与，不仅需要大量协调，也需要付出相当大的努力。对于许多期刊，尤其是资源有限的期刊而言，目前在实践中可能难以实施此类程序。尽管如此，为提高实践中可靠可重复结果的比例，应有更多期刊推行此类政策。对于科学论文，人们普遍认为，没有同行评审的期刊必然会产出质量较低的成果，因此，科学论文的同行评审被视为研究质量保证的必要组成部分。由于代码和数据补充材料是相关科学论文的重要基础，对这些补充材料进行评审同样显得至关重要——这有助于发现分析中的错误，因此将是应对可重复性危机的一项具体且建设性的举措。

如果将可复现性作为一项要求，也有助于更早发现错误。目前，开展可重复性检查的期刊通常会要求作者只有在文章根据其科学内容有条件地被接收后，才提交代码和数据补充材料。这种做法的一个原因是希望将可重复性审查的工作量控制在可管理的范围内。另一个可能的原因是，如果作者尚不知道文章是否会被目标期刊接收，他们可能不太愿意准备能够实现可复现性的整洁代码和数据补充材料。这种意愿下降，主要是因为大多数期刊根本不要求可复现性。

不过，如果更多期刊在初次投稿时就要求提交代码和数据补充材料，期刊提出此类要求可能会变得更容易。一旦向期刊提交代码和数据补充材料成为普遍预期，作者便可以合理地预期，如果论文被拒，他们也需要向下一家期刊提交这些材料。在初次投稿阶段就提交代码和数据补充材料，对期刊和作者都有益，因为这样可以降低论文在发表前不久因代码错误导致关键结果无法复现而被拒稿的风险。据我们的编辑经验，这种情况并不常见，但确实会发生。

区分所使用的数据是否可以公开，还是因法律、伦理或专有权限制而无法公开发布，这一点十分重要。在后一种情况下，科学界无法实现完全可复现性；但在可行的情况下，至少应向审稿人或可重复性编辑提供数据，以便独立验证结果。作者还应讨论，在何种条件下其他研究人员可以访问数据以开展进一步研究。

5.3 审稿人的作用

本文重点关注作者和期刊的视角，因为我们认为，这些科学参与者对可复现实践的传播有最直接的影响。然而，如上所述，审稿人对于确保可复现性也有重要影响，尽管他们通常获得的激励较少。这并不意味着审稿人不必关心所评估的结果是否正确且可复现；忽视这一方面可能导致评估不完整。

即使不进行耗时的可重复性检查，审稿人也能做出有价值的贡献。例如，如果他们指出缺少代码和数据，或者更进一步，说明是否提供了可用于复现论文中所有图和表的代码，这将大有帮助。如果期望审稿人投入更多，明确的指南和易于验证的质量标准将大有裨益。

此外，可以通过公开可见的信息认可审稿人对可复现性的验证，从而加强对审稿人的激励；这在科学论文的同行评审中已是常见做法。例如，Meta-Psychology 已开始在同行评审者之外，也认可可重复性检查人员

的贡献，这或许能为其他期刊提供借鉴。对审稿人而言，一个普遍的好处是，检查同行的代码可能使他们学到新的编码实践，并将其应用于自己的研究中。此外，在某些情况下，审查代码还能让人更深入地了解文章中讨论的方法。

尽管复现报告结果是计算可复现性的核心，但这并不能揭示对分析的描述是否与代码中的实现相符。这一点常常被忽视，而同行评审者未来或许能够在这方面提供宝贵的见解。

5.4 未来展望：教育与自动化

作者、期刊和代码审查者都能从清晰的培训材料中受益，这些材料应展示开展可重复研究的各个方面。材料可以采用交互式教程或短视频的形式。不过，应确保内容清晰简洁，以免令目标用户不堪重负；有关相应培训课程的报告也印证了这一点 [41]。R 社区中此类材料的例子包括 Jennifer Bryan 及其合作者开发的教学资源，例如“What They Forgot to Teach You About R”网站（https://rstats.wtf）以及关于文件命名的简明指南（https://github.com/jennybc/how-to-name-files）。这些资源展示了简洁实用的指导如何帮助作者采用更规范的工作流程，从而支持可复现性。

从长远来看，可重复性检查或许能日益实现自动化。不断改进的容器化技术（例如 Docker 镜像、MaRDI 打包系统 [42]）或环境管理工具（例如用于 R 的 renv 或用于 Python 的 conda），都有助于描述计算环境。基于自然语言的模型可以协助可重复性编辑或审稿人，自动完成工作流程中的部分环节、支持与作者沟通，或在人为评估之前进行初步检查。然而，即使可重复性检查最终实现了完全自动化，仍应重视代码的可访问性和可读性，以确保人类能够理解代码。这一点尤为重要，因为代码和数据补充材料是实证研究的核心组成部分。

6 结论

本文旨在探讨当前阻碍许多作者和期刊更积极地关注基于数据的分析的可复现性的因素，希望以此推动可重复研究得到更广泛的传播。

为此，我们首先总结了可复现性为作者和期刊带来的诸多切实益处，旨在增强这些关键参与者的积极性。随后，我们基于在代码审查方面的丰富经验，提出了务实且易于实施的可复现分析操作指南，以消除人们常有的无所适从感；这种感受往往源于指南过于复杂。最后，我们建议建立一个多层级体系，从多个维度评估可复现性。待该体系进一步细化后，期刊便可采用标准化方式报告已发表实证研究的可复现性水平，这或许能进一步推动可复现性的传播、制度化和标准化，同时使期刊和作者的预期更加明确。

这一体系的核心基础在于认识到可复现性存在于一个连续谱上。因此，作者、期刊和审稿人即便采取微小而务实的步骤，也能切实有助于提升实证研究的质量保证。

致谢

我们衷心感谢引言中所用的类比——即将实证研究中代码与数据的作用比作纯数学中证明的作用——最初由Fabian Scheipl博士在一次无关的先前讨论中提出。

RH、LN 和 OZ 感谢德国研究联合会（DFG）通过国家研究数据基础设施（NFDI），在数学研究数据倡议（MaRDI）框架下提供资助，项目编号为 460135501，NFDI 29/1。RK 感谢英国工程与物理科学研究理事会通过 EP/T021020/1 项目部分资助本研究。ALB 感谢德国研究联合会通过 BO3139/9-1、7-2 和 10-1 项目提供资助。

利益冲突

为确保透明度，现披露以下职务。RH、TU、MK 和 BH 担任《Biometrical Journal》的可重复研究编辑。RK 担任 Journal of Statistical Software 和《计算与图形统计学杂志》的联合主编、《技术计量学》的副编辑，并为 ROpenSci统计软件同行评审标准委员会成员。CJP 和 JW 担任《美国统计学会杂志》的可复现性副编辑。JC 担任 Computo 主编，PN 担任 Computo 副编辑；两人均为该期刊的创办人。MH 担任慕尼黑机器学习中心（MCML）的可复现性与开放科学转移协调员。LB 担任可复现性协调员，RC 担任 Meta-Psychology 主编。

补充材料

本文的补充材料可在 arXiv 页面上获取。

参考文献

[1] 美国国家科学院、工程院和医学院。《科学中的可重复性与可再现性》。美国国家科学院出版社，华盛顿特区，2019年。doi:10.17226/25303。[2] Mark D Wilkinson、Michel Dumontier、IJsbrand Jan Aalbersberg、Gabrielle Appleton、Myles Axton、Arie Baak、Niklas Blomberg、Jan-Willem Boiten、Luiz Bonino da Silva Santos、Philip E Bourne等。科学数据管理与托管的FAIR指导原则。*Scientific Data*，3(1):1–9，2016年。doi:10.1038/sdata.2016.18。[3] Jon F Claerbout与Martin Karrenbach。电子文档赋予可重复研究新的意义。载于*1992*年*SEG*技术会议扩展摘要，第601–604页。勘探地球物理学家协会，1992年。doi:10.1190/1.1822162。[4] Jonathan B Buckheit与David L Donoho。Wavelab与可重复研究。载于《小波与统计》，第55–81页。施普林格，1995年。doi:10.1007/978-1-4612-2544-7 5。[5] Tom E Hardwicke、Robert T Thibault、Beth Clarke、Nicholas Moodie、Sophia Cr¨uwell、Sarah R Schiavone、Sarah A Handcock、Khanh An Nghiem、Fallon Mody、Tuomas Eerola等。心理学中透明研究实践的流行率：一项针对2022年发表实证文章的横断面研究。《心理科学方法与实践进展》，7(4): 25152459241283477，2024年。doi:10.31234/osf.io/t2zs9。

[6] Alfredo S´anchez-T´ojar、Aya Bezine、Marija Purgar与Antica Culina。代码共享政策与生态学研究发现可重复性潜力的提高相关。Peer Community Journal，5，2025年。doi:10.24072/pcjournal.541.[7]Stylianos Serghiou、Despina G Contopoulos-Ioannidis、Kevin W Boyack、Nico Riedel、Joshua D Wallach与John PA Ioannidis。生物医学文献中透明度指标的评估：开放究竟有多开放？PLoS Biology，19(3):e3001107，2021年。doi:10.1371/journal.pbio.3001107.[8] Daniel G Hamilton、Kyungwan Hong、Hannah Fraser、Anisa Rowhani-Farid、Fiona Fidler与Matthew J Page。医学与健康科学中数据与代码共享的流行率及预测因素：基于个体参与者数据的系统综述与荟萃分析。bmj，382，2023年。doi:10.1136/bmj-2023-075767.[9] Sheeba Samuel与Daniel Mietchen。生物医学出版物中Jupyter笔记本的计算可复现性。GigaScience，13:giad113，2024年。doi:10.1093/gigascience/giad113.[10]David Donoho。处于奇点的数据科学。Harvard Data Science Review，6(1)，2024年。doi:10.1162/99608f92.b91339ef.[11] J Vanschoren、J N van Rijn、B Bischl与L Torgo。OpenML：机器学习中的网络化科学。SIGKDD Explor，15(2):49–60，2013年。doi:10.1145/2641190.2641198.[12] Paul E Smaldino与Richard McElreath。不良科学的自然选择。Royal Society Open Science，3(9):160384，2016年。doi:10.1098/rsos.160384.[13] Randall J LeVeque、Ian M Mitchell与Victoria Stodden。面向科学计算的可重复研究：改变文化的工具与策略。Computing in Science & Engineering，14(04):13–17，2012年。doi:10.1109/MCSE.2012.38.[14] Victoria Stodden、Peixuan Guo与Zhaokun Ma。迈向可复现计算研究：期刊数据与代码政策采纳的实证分析。PLOS ONE，8(6):e67111，2013年。doi:10.1371/journal.pone.0067111.[15] Ben Goldacre、Caroline E Morton与Nicholas J DeVito。研究人员为何应共享其分析代码。BMJ，367，2019年。doi:10.1136/bmj.l6365.[16] Jennie DeBlanc、Bradley Kay、Jessica Lehrich、Neil Kamdar、Thomas S Valley、John Z Ayanian与Brahmajee K Nallamothu。综合医学期刊中使用Medicare数据的研究的统计代码可得性。JAMA Internal Medicine，180(6):905–907，2020年。doi:10.1001/jamainternmed.2020.0671.[17] Ricardo Sanchez、Beth Ann Griffin、Joseph Pane与Daniel F McCaffrey。统计计算的最佳实践。Statistics in Medicine，40(27):6057–6068，2021年。doi:10.1002/sim.9169.[18] Ana Trisovic、Matthew K Lau、Thomas Pasquier与Merc`e Crosas。关于研究代码质量与执行的大规模研究。Scientific Data，9(1):60，2022年。doi:10.1038/s41597-022-01143-6.[19] D Stephen Lindsay。致出版期刊的心理学专业学会的呼吁：评估计算可复现性。Meta-Psychology，7，2023年。doi:10.15626/mp.2023.4020.[20] Ilka Agricola、Lynn Heller、Wil Schilders、Moritz Schubotz、Peter Taylor与Luis Vega。数学科学中的欺诈性发表。Notices of the American Mathematical Society，72(9):1038–1049，2025年。doi:10.1090/noti3217.

[21] Melissa Assel 和 Andrew J Vickers。《高影响力专科医学期刊临床研究论文中的统计代码》。《内科学年鉴》，168(11):832，2018年。doi:10.7326/M17-2863。[22] Jesse M Alston 和 Jessica A Rick。《开展可复现研究的初学者指南》。《美国生态学会通报》，102(2):1–14，2021年。doi:10.1002/bes2.1801。[23] Luke V Rasmussen、Eric W Whitley、Leah J Welty 等。《实用型可复现研究：点滴改进从原始数据到结果的研究流程》。《临床研究杂志》，133(16)，2023年。doi:10.1172/JCI173741。[24] Dylan GE Gomes、Patrice Pottier、Robert Crystal-Ornelas、Emma J Hudgins、Vivienne Foroughi- rad、Luna L S´anchez-Reyes、Rachel Turba、Paula Andrea Martinez、David Moreau、Michael G Bertram 等。《我们为什么不共享数据和代码? 公开存档实践中感知到的障碍与益处》。《英国皇家学会学报B辑》，289(1987):20221113，2022年。doi:10.1098/rspb.2022.1113。[25] Anusha M Vable、Scott F Diehl 和 M Maria Glymour。《代码审查：提升可复现性、加快学习并提高团队研究质量的简便方法》。《美国流行病学杂志》，190(10):2172–2177，2021年。doi:10.1093/aje/kwab092。[26] Leo A Celi、Luca Citi、Marzyeh Ghassemi 和 Tom J Pollard。《PLOS ONE健康与生物医学机器学习专题：迈向开放代码和开放数据》。《PLOS ONE》，14(1):e0210232，2019年。doi:10.1371/journal.pone.0210232。[27]Fabien CY Benureau 和 Nicolas P Rougier。《重新运行、重复、再现、再利用、复现：将代码转化为科学贡献》。《神经信息学前沿》，11:69，2018年。doi:10.3389/fninf.2017.00069。[28] Patrick Vandewalle。《代码共享与图像处理领域的研究影响力相关》。《科学与工程中的计算》，14(4):42–47，2012年。doi:10.1109/MCSE.2012.63。[29] Brian Maitner、Paul Efren Santos Andrade、Luna Lei、Jamie Kass、Hannah L Owens、George CG Barbosa、Brad Boyle、Matiss Castorena、Brian J Enquist、Xiao Feng 等。《生态学与进化领域的代码共享能够提高引用率，但仍不普遍》。《生态学与进化》，14 (8):e70030，2024年。doi:10.1002/ece3.70030。[30] Heather A Piwowar、Roger S Day 和 Douglas B Fridsma。《共享详细的研究数据与引用率提高相关》。《PLOS ONE》，2(3):e308，2007年。doi:10.1371/journal.pone.0000308。[31] Giovanni Colavizza、Iain Hrynaszkiewicz、Isla Staden、Kirstie Whitaker 和 Barbara McGillivray。《将出版物关联至研究数据的引用优势》。《PLOS ONE》，15(4):e0230416，2020年。doi:10.1371/journal.pone.0230416。[32] Jan H H¨offler。《复现与经济学期刊政策》。《美国经济评论》，107(5): 52–55，2017年。doi:10.1257/aer.p20171032。[33] Vahid Jalili、Dave Clements、Bj¨orn Gr¨uning、Daniel Blankenberg 和 Jeremy Goecks。《将软件添加到包管理系统可使其引用量提高280%》.bioRxiv，页码2020–11，2020年。doi:10.1101/2020.11.16.385211。[34] Anna Maria Streiber、Sanne JW Hoepel、Elisabet Blok、Frank JA Van Rooij、Julia Neitzel、Jeremy Labrecque、M Kamram Ikram 和 Daniel Bos。《提高医学研究中数据分析与代码的可复现性：入门的5条建议》。《英国医学杂志开放版》，15(10):e104691，2025年。doi:10.1136/bmjopen-2025-104691。

[35] Erin C McKiernan、Philip E Bourne、C Titus Brown、Stuart Buck、Amye Kenall、Jennifer Lin、Damon McDougall、Brian A Nosek、Karthik Ram、Courtney K Soderberg 等。《开放科学如何助力研究人员取得成功》。《eLife》，5:e16800，2016年。doi:10.7554/eLife.16800。[36]Geir Kjetil Sandve、Anton Nekrutenko、James Taylor 和 Eivind Hovig。《可复现计算研究的十条简单规则》。《PLOS计算生物学》，9(10):e1003285，2013年。doi:10.1371/journal.pcbi.1003285。[37] Greg Wilson、Jennifer Bryan、Karen Cranston、Justin Kitzes、Lex Nederbragt 和 Tracy K Teal。《科学计算中的足够好实践》。《PLOS计算生物学》，13(6):e1005510，2017年。doi:10.1371/journal.pcbi.1005510。[38] Gareth B Jenkins、Andrew P Beckerman、C´eline Bellard、Ana Ben´ıtez-L´opez、Aaron M Ellison、Christopher G Foote、Andrew L Hufton、Marcus A Lashley、Christopher J Lortie、Zhaoxue Ma 等。《生态学与进化领域的可复现性：数据和代码的最低标准》。《生态学与进化》，13(5):e9961，2023年。doi:10.1002/ece3.9961。[39] Julia Wrobel、Emily C Hector、Lorin Crawford、Lucy D'Agostino McGowan、Natalia da Silva、Jeff Goldsmith、Stephanie Hicks、Michael Kane、Youjin Lee、Vinicius Mayrink 等。《与作者合作提升JASA的可复现性》。《美国统计学会杂志》，119(546):795–797，2024年。doi:10.1080/01621459.2024.2340557。[40] Jeroen Hanselman。《代码审查模板》。https://github.com/JHanselman/code-review-template，2025年。URL https://github.com/JHanselman/code-review-template。[41] Verena Heise、Constance Holman、Hung Lo、Ekaterini Maria Lyras、Mark Christopher Adkins、Maria Raisa Jessica Aquino、Konstantinos I Bougioukas、Katherine O Bray、Martyna Gajos、Xuanzong Guo 等。《参加培训课程后落实开放且可复现研究实践的十条简单规则》。《PLOS计算生物学》，19(1):e1010750，2023年。doi:10.1371/journal.pcbi.1010750。[42] Aaruni Kaushik。《将预定义的软件环境运行时作为衡量可复现性的指标》。载于《数学软件——ICMS 2024》，页码245–253，2024年。doi:10.1007/978-3-031-64529-7 26。