arXiv:2611.30021v1 [stat.CO] 23 Nov 2026

StatClaim：将研究设计视为类型、将方案视为程序、将统计主张视为证明义务

Supun Manathunga麦
吉尔大学

摘要

统计保证取决于研究模型、假设、抽样计划与分析之间的兼容性。StatClaim 是一种声明式领域特定语言，用于将研究方案和统计分析计划编码为程序；这些程序由若干对象构成，其类型编码了它们在研究设计中的统计角色。编译器会检查类型兼容性，并应用契约来检查这些对象的值及其关系是否满足其他要求。请求验证时，StatClaim 的后端会在给定假设下，使用 Lean 构造并检查对受支持统计主张的证明。这些主张涉及第一类错误、家族错误率、检验效能和区间覆盖率，也包括分析方法与程序同预设停止规则的受支持组合。报告会区分形式化证明的、有条件的、引用的和作者声明的保证，并为已验证的主张提供可重放证书。同一规范还支持使用精确计算、解析近似或可复现模拟进行样本量搜索、功效计算和区间精度评估。命令行界面、编辑器集成和图形化块编辑器共用一个编译器和执行层。验证工作重现了 Simon 发表的表格中的 102 种最优和极小极大两阶段设计，并在 Lean 中验证了它们的错误率和检验效能界限。一项临床试验案例研究展示了如何使用引用的方法和有条件的证明，对一份已发表的分析计划进行部分编码。该开源实现包含可运行的教程和示例，用于在收集结局数据前指定并检查研究设计。

关键词：研究设计、领域特定语言、类型系统、形式化验证、样本量规划、Lean。

1. 引言

统计分析可以被正确计算，但将其用于一项研究却未必合理。检验的某些性质（例如其I类错误保证）取决于它所适用的其他要素，例如零假设、抽样机制和决策规则。例如，在5%的显著性水平下分别检验多个终点，并不能仅凭此确立5%的族系错误界。同样，对于固定样本量下的单次分析所作的保证，

也不会自动延伸到一种反复检查累积数据并在达到显著性时停止的程序。在方案制定期间识别并处理这些问题，能够让研究人员在数据收集开始前修改研究设计。

StatCla im 是一种用于表达统计组成部分的领域特定语言（DSL） f

将研究方案表示为类型化对象，包括模型、抽样计划、假设、分析和主张（Manathunga2026）。领域专用类型用于区分这些概念，而声明则构造值，其关系由兼容性契约检查。例如，分析必须支持已声明的零假设，并使用该零假设所属的模型。编译器会报告不兼容的声明和未解决的要求，而不会执行统计计算。仅靠编译并不能确立统计保证。请求验证时，StatClaim 的后端会使用Lean(de Moura and Ullrich 2021）构造并检查受支持统计主张的证明。

可执行设计规范和统计程序的形式化检查有着重要的先例。**DeclareDesign**通过模型、研究问题、数据策略和答案策略来表示研究设计，并通过模拟诊断检验效能、偏倚和覆盖率等特性（Blair、Cooper、Coppock 和 Humphreys 2019；Coppock、Blair、Cooper、Humphreys 和 Fultz 2026）。**StatWhy**通过 Why3 平台验证带注释的假设检验程序，检查使用统计方法的要求（Kawamoto、Kobayashi 和 Suenaga 2025）。StatClaim 将侧重研究设计的语言与明确的兼容性契约、分级证据，以及对受支持方法的统计保证本身进行的形式证明结合起来。

StatClaim 主要侧重于设计阶段，旨在使方案的统计论证依据明确、可审查，并在支持的情况下，于观测结局之前进行形式化验证。主张涉及第一类错误、家族错误率、检验效能或区间覆盖率。组合支持涉及多个终点或计划中的期中分析的程序，并对整个已声明的程序提出要求。尚未解决的要求意味着缺少论证依据，但不一定表明所提出的统计主张是错误的。成功的检查会解释兼容性契约为何成立，并报告主张所依赖的假设。这些解释不同于对统计保证的证明。证据等级区分了在 Lean 中证明的保证、以成员保证为假设而有条件证明的组合、引用自文献的结果，以及作者声明的陈述。

同一规范支持通过功效评估、区间精度评估和样本量搜索开展研究规划。根据所采用的方法，这些功能使用精确计算、解析近似，或带有已报告蒙特卡洛不确定性的可复现模拟。受支持的精确规划主张可以证明某项设计满足其错误率和功效目标，并在适用时证明在已声明的搜索范围内不存在更小且足够的候选方案。功效是在指定的规划备择假设下进行评估的。目前，精度计算和模拟估计仅提供数值评估，尚无经过 Lean 验证的保证。

编译使研究报告可供审阅。成功的形式化验证会生成可重放的证明证书。研究人员可以使用文本编辑器将方案编写为 `.stat`源文件，并通过编辑器集成获得语言提示；也可以使用 StatClaim Studio——一个由同一编译器和统计契约支持的图形化块编辑器。

作为一个简单的示例，以下程序声明两个组样本量固定的独立二项样本，指定比例相等的零假设，并请求验证双侧费舍尔检验的第一类错误界。该程序不需要观测数据，其主张涉及零假设下每个共同成功概率对应的程序重复抽样行为。

```
use models.two_arm_binomial
use sampling
use tests.fisher

let study = two_arm_binomial(n_control=50, n_treatment=50)
let design = fixed(model=study)
let null = equal_proportions(model=study)
let analysis = fisher(design=design, null=null, alpha=1/20)

let claim = typeI(analysis)
let verification = verify(claim)
```

在 StatClaim Studio 中打开此示例。 执行环境在第6.2节中进行了描述。运行 `statclaim check study.stat` 会报告已确立的保证及其假设。以下输出为摘录，并已换行：

```
verified [verification]: typeI(analysis)
Evidence: proved in Lean; guarantee: finite-sample bound
The specified two-sided Fisher analysis has Type I error at most
1/20 under the null, for every common success probability in [0, 1].
Assumptions:The samples follow two independent binomial distributions.
Positive arm sizes were fixed before outcomes were observed.
The error bound is under the null of equal success probabilities,
for every p in [0, 1].Program status: verified
```

仅将抽样计划更改为未指定的依赖数据的停止，

```
let design = adaptive(model=study)
```

会使该保证仍未得到解决。检查器报告了以下诊断信息（省略源位置并已换行）：

```
obligation [sampling_scope]: analysis: data-dependent stopping needs
a justified sampling law before the fixed-sample guarantee can be used.
This is not evidence of inflated error.
```

该诊断信息指出缺少论证依据。对于依赖数据的停止，其保证需要规定完整的程序以及适用的定理。完整变体也可以在 StatClaim Studio 中打开。

当前的程序库包含适用于二项分布和特定观察性设计的精确方法、随机化检验和两阶段单臂设计，也包含适用于引用方法的契约，例如 t 检验、对数秩检验和回归分析。

StatClaim 的贡献在于整合带类型的研究规范、可解释的兼容性检查、规划操作，以及针对统计主张的明确证据。第 2 节介绍统计对象、主张和兼容性契约。第 3 节描述语言和编译器架构。第 4 节阐述契约推导、形式化验证和证书重放。第 5 节介绍检验效能、样本量和精度

规划，包括基于模拟的评估。第6节介绍创作界面和方法扩展工作流。第 7节检查已发表的两阶段设计，并介绍慢性阻塞性肺疾病分析计划示例。第 8 节讨论范围、局限性和未来发展。附录 A 总结教程和示例的覆盖范围。

2. 统计主张与契约

2.1. 对象及其统计含义

StatClaim 将描述一项研究所需的对象与论证相关主张所需的证据区分开来。表1 概述了各对象的主要作用。模型规定一族概率律和一个可接受的结果域。抽样计划描述如何获取观测值，包括停止机制。分析将一项设计、一个零假设和一个带有相应设置的程序结合起来。对于引用的方法，模型声明和列出的假设不必构成完全形式化的概率律。

对象	Role
模型	参数域、可能的观测值及抽样假设。
抽样计划	抽样如何进行，以及何时结束。
假设	评估保证所依据的一项限制条件。
分析	一项设计、支持的假设、程序和设置。
观测	根据已声明的域检查所提供的数据。
主张	分析、结果或规划目标的一项指定性质。
执行结果	计算、模拟或验证请求的结果。
证据	已确立陈述的依据，包括假设和来源。

表1：主要对象角色。特定声明会构造一个值，而其类型决定了该值在后续声明中可以发挥的作用。

对于固定样本模型，设 Θ 为参数空间， P_θ 为结果空间 $\mathcal{X}$ 上的概率律。分析指定一个零假设集合 $H \subseteq \Theta$ 和一个可测的拒绝事件$R \subseteq \mathcal{X}$。其显著性水平为 α 的第一类错误主张是

$$\forall \theta \in H, \qquad P_\theta(R) \le \alpha. \tag{1}$$

抽样计划之所以重要，是因为赋予其观测值的抽样律必须与定理所使用的抽样律一致。在共享的 Lean 核心中， `Compatibility` 明确提供了这种一致性。可以使用模型的抽样律构造固定计划，但未指定的自适应计划不会继承该抽样律。因此，对于一个有依据的抽样律，仍存在一项尚未解决的要求。对于区间估计，令 $\tau(\theta)$ 表示所关注的量，例如响应率或两个响应率之差。令 X 表示服从抽样律 P_θ的随机观测，并令 $C(x)$ 表示程序针对可能的观测 x返回的区间。在置信水平 $1-\alpha$ 下，覆盖率声明指出：

$$\forall \theta \in \Theta, \qquad P_\theta\{\tau(\theta) \in C(X)\} \ge 1-\alpha. \tag{2}$$

对于每个固定的参数值，从模型中重复抽取样本时，该区间以至少 $1-\alpha$ 的概率包含所关注的量。

当抽样可能提前停止时，记录的观测值也取决于停止规则。例如，两阶段二项研究首先在 n_1 名参与者中观测到响应计数 X_1。如果 X_1 不大于预先设定的阈值 r_1，则因无效性而停止。否则，将在另外 n_2 名参与者中观测第二个计数 X_2，并应用最终拒绝规则。记录中要么包含第一个计数和提前停止指示变量，要么包含两个计数和完成指示变量。

形式化核心将这一机制表示为抽样策略。令 Z 表示观测所有计划阶段时的潜在数据，其可能值属于空间$\mathcal{Z}$，概率律为 L_θ。在两阶段示例中，$Z=(X_1,X_2)$具有已声明的乘积二项分布律。可测规则 $g\colon \mathcal{Z}\to\mathcal{Y}$ 将潜在数据映射为记录 $Y=g(Z)$，其可能值属于$\mathcal{Y}$。对于记录的任意可测集合 $B\subseteq\mathcal{Y}$，其概率为

$$Q_\theta(B)=L_\theta\{z\in\mathcal{Z}\mid g(z)\in B\}. \tag{3}$$

因此，Q_θ 是应用停止规则后记录数据的分布。第一类错误和检验效能根据该分布以及针对记录定义的拒绝规则进行评估。已实现的两阶段模块规定了这些组成部分。

StatClaim 通过完整的设计构造器支持显式停止规则，包括两阶段二项设计。通用的 `adaptive(model=study)` 声明则标示一种未指定的依赖数据的机制。要获得受支持的保证，必须通过适当的设计构造器指定该机制。

2.2. 兼容性、假设与开放要求

类型兼容声明还需对其值及彼此之间的关系进行额外检查。这在概念上与精化类型有关，精化类型用于在现有类型中表达更精确的限制（Freeman 和 Pfenning（1991）；Rondon、Kawaguchi 和 Jhala（2008））。例如，分析构造器要求某个假设属于设计的模型，且所选方法支持该假设。StatClaim 通过在 Elaboration 期间评估已注册的兼容性契约来实现这些限制。目前，它不提供通用的细化类型推断或基于谓词的子类型。

针对二项计数的检验无法使用连续结局模型。对于双侧 Fisher 分析，支持比例相等，而差值非零的相等关系则属于不同的统计问题。精确零差值通过形式化库中确立的一种等价关系，规范化为比例相等。检查器不会尝试证明假设之间任意的等价关系。

StatClaim 会拒绝组合了不兼容对象的声明。如果声明有效，但所请求的保证缺少必要的论证依据，那么该主张仍未得到解决。研究人员必须评估，为有效程序显现的假设是否适用于实际研究。

设计声明可以揭示与方法假设之间的矛盾。声明每位参与者有多次测量值，就不能再用普通双样本 t 检验将这些测量值视为相互独立的参与者。声明采用集群分配，就不能用参与者人数替代独立集群的数量。结局方向也可以

证据等级	结果状态	该陈述的依据
已证明	`verified`	经 Lean 检查的证明。
有条件的	`conditional`	一种经过核查的组合，假定其成员保证——条件。
引用的	`cited`	其支持的契约符合该声明的文献结果。
已声明	`declared`	没有经过检验保证的作者主张。

表2：StatClaim 中陈述的证据等级。

明确说明，并与单侧零假设进行核对。如果省略这些事实，相关假设仍然是显而易见的。

2.3. 证据与范围

表2 区分了陈述的依据与其执行状态。引用的方法可以通过其兼容性检查，而无需对其统计保证提供形式化证明。有条件的结果建立了一个组合定理，其中的成员界仍属于前提。已声明的方法记录作者的主张，并检查所支持的设计要求。它不为该主张提供数学论证。

证据的范围单独记录。有限样本保证适用于已声明的样本量，而渐近陈述描述极限行为。有条件的组合可能继承渐近成员的局限性。每个结果都标明其证据中最薄弱的成员，文件摘要则报告其具有证据的结果中最薄弱的部分。

2.4. 程序组合

组合将单项分析的保证汇总为涉及多个检验的程序所提供的保证。例如，一项研究可能检验两个终点，并要求任何错误拒绝的概率至多为 5%。Bonferroni程序可以为每项检验分配 2.5%。如果每项检验都将自身的第一类错误控制在该显著性水平，那么拒绝任何真零假设的概率至多为所分配显著性水平之和，即 5%。这些终点不必相互独立。

更一般地，考虑 m 项检验。对于检验 i，令 H_i 为其零假设，R_i 为其拒绝事件，α_i 为该零假设为真时的误差界。在参数值 θ处，记 $I_0(\theta)$ 为真零假设的指标集合。家族错误率是至少拒绝其中一个假设的概率。并集界给出

$$P_\theta\left(\bigcup_{i\in I_0(\theta)} R_i\right) \le \sum_{i\in I_0(\theta)} \alpha_i \le \sum_{i=1}^{m} \alpha_i \le \alpha. \tag{4}$$

只要单项错误界限成立，该界限就适用于真零假设和错误零假设的每一种组合。StatClaim 会检查已声明的显著性水平之和是否至多为所请求的检验族水平 α。已证明的 Fisher族适配器还会根据其声明的抽样假设确立单项界限。通用 Bonferroni适配器使用 Lean 检查组合，而单项界限仍作为假设，从而提供条件性证据。

固定序列检验使用不同的规则。终点按照预先指定的顺序进行检验，每个终点均使用完整的显著性水平 α；一旦某个终点未被拒绝，检验就会停止。任何错误拒绝都必须意味着按该顺序排列的第一个真零假设被拒绝，而该零假设自身的错误概率至多为α。Lean 会在单项检验界限这一条件下检查此论证。

预先规划的期中分析会在多个已声明的样本量下重复检验同一个问题。每次检验分配一部分错误预算，并在首次拒绝时终止研究。StatClaim 会检查分配的显著性水平之和是否不超过 α ，并对各次检验应用并集界。由此得到的主张涉及该单一问题的第一类错误。当前构造假设每次检验都满足其界限，且不会利用各次检验之间的相关性来获得较不保守的阈值。

图形化程序允许根据已声明的图，将被拒绝终点所分配的显著性水平转移给其他终点。StatClaim 会检查初始分配和转移规则，而程序的一般保证则通过引用给出。检验族或序列也可以将一个完整的受支持程序作为单个条目。由此得到的证据会保留其所依赖部分的假设和等级。

3. 语言与架构

3.1. 声明与精确输入

一个 `.stat` 文件是由一系列模块导入项、不可变绑定和显式操作请求组成的。以下示意语法概述了其声明结构。构造器签名提供特定领域的参数约束。

```
statement  = "use" module
           | "let" name "=" expression
           | operation
expression = literal | name | call | list | expression "." name
operation  = verify(claim)
           | calculate(claim)
           | simulate(claim, seed=integer, ...)
```

省略号表示可选的模拟设置。导入项从已安装的注册表中选择构造器。它们不会加载任意 Python 代码或其他 `.stat` 文件。名称必须先定义后使用，且不能重新绑定。调用时可以先使用位置参数，再使用命名参数。注释以 `#` 开头。当前语言版本（v0.41.0）没有通用算术表达式、循环、条件语句或嵌入式宿主语言求值。更复杂的统计程序通过具有明确语义的已注册构造器提供。

整数、有限小数和分数都是精确输入。例如，0.025 和 `1/40`表示相同的有理数值。分数字面量不允许对任意表达式进行除法。前端会检查正样本量、概率水平和计数界等域，并在需要时施加特定于方法的限制。输入表示精确，并不意味着后端的每项计算都是精确的。解析检验效能近似和模拟使用数值算法，其状态会单独报告。

模型、假设和分析各有不同作用，构造器要求相应的输入类别。具体研究是此类类别中的一个值。其组样本量和停止规则也会参与语义检查。前端不会将每项统计条件都编码为单独的依赖类型。Lean 后端使用按模型和主张索引的对象来表示数学陈述和证据。

3.2. 组合模型并为主张命名

双臂模型可以由一项分配和一个结局构建而成，从而让多个结局共用同一分配。以下自包含示例声明每组 100 名参与者的连续得分：

```
use models.two_arm_continuous

let arms = assignment(n_control=100, n_treatment=100)
let score = continuous_outcome(better="lower")
let study = two_arm(assignment=arms, outcome=score)
```

在 StatClaim Studio 中打开此示例。 共用一项分配会复用其已声明的结构，但并不表示各结局之间相互独立。其他字段可描述随机化、聚类、重复测量或分析人群。可以为结局命名，以便在同一检验族中检测对同一已声明问题的重复检验。但仍必须准确说明这些名称的含义。

主张可以命名并重复使用。在入门程序中， `typeI(analysis)` 会先构建一个主张，然后才请求验证。因此，在任何后端运行之前，都可以对其进行检查或重复使用。观测结果主张则单独构建。继续前面的入门示例，

```
let data = observe(analysis=analysis,control_successes=12,
 treatment_successes=24)let observed = result(data)
let computation = calculate(observed)
let checked_result = verify(observed)let saved = checked result_
let p_value = computation.calculation.p_value
let certificate = checked result.evidence.certificate.file_
```

在 StatClaim Studio 中打开完整程序。此处 `calculate` 计算一个拟议结果和决策，而 `verify` 请求对受支持的结果属性进行独立的形式化检查。别名 `checked_result` 不会再次运行验证。观测计数及其 p值都不能证明零假设为真。

3.3. 公共字段与执行阶段

只能访问公共模式中的字段。分析会公开其设计、零假设和显著性水平。主张会公开其主体和范围。执行结果会公开状态、诊断信息和相应的结果载荷。同一模式驱动字段检查、补全和悬停提示

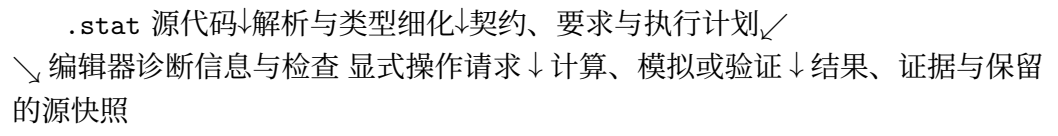

图1：编译由各接口共享，不会执行所请求的操作。形式化验证是一条明确的执行路径。

信息和序列化。字段访问不会提供对底层 Python 属性或方法的任意访问。

有些投影在编译时已知。例如， `analysis.alpha` 可以配置另一个兼容的构造器。其他投影则取决于执行。编译器会为 `computation.calculation.p_value`保留一个带类型的延迟投影，并在执行期间解析它。这样的值可以命名、设置别名或收集到列表中，但不能用于配置新的统计构造器。该构造器的输入必须在编译期间已知。这样可以防止延迟结果悄然改变正在检查其主张的研究。

证据具有可选类型。其投影在执行前处于待定状态，在验证尝试未成功后则不存在。这是两种不同的状态。通过不存在的可选值进行访问会传播不存在状态，因此，不能仅仅因为其字段表达式通过了类型检查，就将证书路径视为存在。引用证据可以存在，而不具备形式化证书。可选值会保留在结果模式中，不会被隐式转换为证明对象。

3.4. 编译器与执行流水线

图 1 展示了共享架构。解析器会创建带有源代码跨度的语法树。Elaboration 会解析导入项和名称，检查参数类别与构造器契约，并构建不可变类型化对象。需求检查会为所请求的主张添加已知义务。绑定会保留参数位置和命名依赖项，因此诊断信息可以指出出错的使用位置及其所引用的声明。

编译器由 Python 实现。它会在内存中构造对象。 `statclaim compilestudy.stat` 执行快速的无后端检查。 `statclaim check study.stat`会编译整个文件，然后按声明顺序执行符合条件的请求。无效文件不会启动统计后端。有效文件可以包含一个未解析的请求，同时还包含其他可执行的请求；报告会记录每个请求的状态以及整体未解析状态。显示编译诊断不需要 Lean 证明、数值计算或证书写入。

执行通过请求标识符解析类型化结果槽。该运行会保留源文件、规范化对象和结果报告。此外，形式化请求还可以添加证书包-

。如果源文件在执行期间发生变化，结果将引用所保留的快照，并将当前文件标记为过时。源文件与结果之间的这种绑定至关重要，因为修改输入可能会同时改变计算以及需要证明的命题。

4. 契约推导与形式化验证

4.1. 解释和检查契约推导

对于受支持的声明，编译器会保留一份推导，说明契约为何成立。推导会记录主体、带版本规则、各规则使用的事实、步骤之间的依赖关系、假设和开放义务。对于双侧 Fisher 分析，编译器会检查设计与零假设是否引用同一模型、方法是否支持已声明的比例相等零假设，以及显著性水平是否处于允许范围内。推导还会记录已声明的抽样计划是固定的，这是 Fisher 第一类错误保证的要求。收到验证请求后，后端会将相应定理应用于已声明的设计和显著性水平，并由 Lean 检查所得证明。

命令 `statclaim explain study.stat` 会根据请求展示这些步骤。独立检查器会根据序列化主体重新评估前提，并检查推导的结构、结论、假设和开放要求。它与推导构建器共用规则陈述，但不会调用构建器或构造器的契约检查函数。这为记录的解释提供了独立检查。更改规则名称或成功状态标志并不足以通过检查。这并非对前端或规则定义的统计正确性的形式化证明。

`statclaim report study.stat` 汇总主张描述、证据预期和假设登记表，便于检查统计分析计划。生成此报告并不会执行验证。请求仍处于待处理状态时，报告可以说明请求成功后将确立什么。 `statclaim impact before.stat after.stat`比较声明及其依赖项，以识别受影响的契约，以及需要重新检查的主张。

4.2. 形式化对象与定理应用

共享形式化库使用Lean和**mathlib** （de Moura and Ullrich 2021；The mathlibCommunity 2020）。其模型对象包含每个参数对应的概率律、一个允许的观测集合，以及可测性和支撑集方面的证据。分析包含其设计、假设和程序。主张包含一个命题，逻辑证明证书则包含该命题的证明。这些逻辑证明证书有别于下文所述的序列化工件包。标签或假定的评估结果不能转化为证明。

在入门示例中，可复用的 Fisher 定理给出了使用概率排序的双侧检验的拒绝概率界：在两个具有共同成功概率的独立二项样本下，拒绝概率至多为 α 。该定理对样本量以及每个实数 $p \in [0,1]$都成立。其证明首先确立有限分布 p 值界，构造归一化超几何概率，并按合并总数分解联合二项概率。对条件界求和即可得到无条件界。与 mathlib 的概率测度建立的形式化联系，使这成为关于以下内容的概率陈述：

已声明模型。该论证还涵盖了 $p = 0$ 和 $p = 1$。
Fisher 提供器要求满足抽样兼容性，并针对所声明的设计和显著性水平实例化该定理。因此，设计级请求不需要观测值。其他模块则提供不同的论述。单侧检验和两阶段设计可以利用单调性，将经过核验的边界计算扩展到复合原假设；区间模块则证明覆盖率。新构造器不会仅仅因为返回了一个接口相似的对象，就自动获得这些定理。

4.3. 核验具体计算

程序级定理与观测计算的正确性是两个不同的命题。如果 T 是形式化程序， x是可接受观测值，且 v 是拟议输出，那么后一个主张是

$$v = T(x). \tag{5}$$

对于 Fisher 检验，Python 使用整数权重递推给出一个精确有理数 p 值。形式化库提供了一个阶乘公式，并已证明该公式等于数学定义。图 2 展示了这两个表达式如何关联。由于分母为正，分数相等可归结为自然数相等，并由 `decide +kernel` 通过Lean内核进行核验。

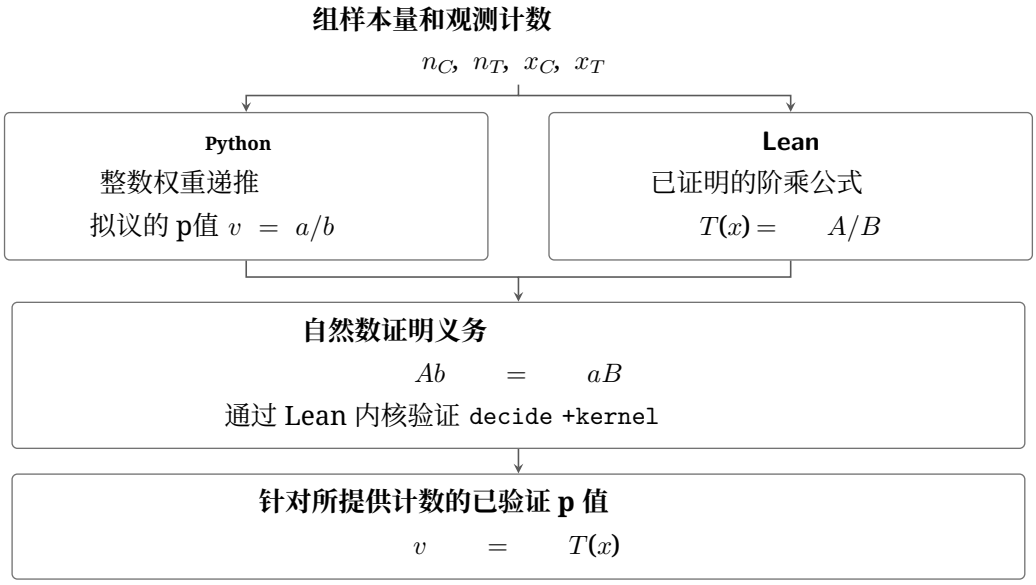

图 2：对所提供计数的 Fisher p 值进行验证。两组的组样本量为 n_C， n_T，观测到的成功次数为 x_C， x_T。分子 a， A和分母 b， B均为自然数，且满足 b， $B > 0$。证书还会检查计数界以及在已声明显著性水平 α下的拒绝决策。第一类错误界由单独的程序定理推出。

这会根据指定的数学程序核验一项特定计算。它并不能证明Python 实现的每次执行都正确，也不能证明所提供的计数是通过已声明的抽样机制得出的。反过来，单靠数值计算也无法为方程 1中的一致误差声明提供证据。

制品	用途
`inputs.json`	规范数学输入和拟议结果（如适用）。
`Proof.lean`	针对该请求生成的陈述和证明。
`Proof.olean`	已编译并检查过的证明制品。
`transcript.log`	后端命令、环境检查和证明输出。
`certificate.json`	主张范围、假设、版本和制品哈希。

表 3：形式化证书包的内容。程序的父级运行报告记录了来源，但并非额外的数学证书。

4.4. 证书签发与重放

在颁发形式化证书之前，后端会构建项目，将正在运行的 Lean 版本与固定工具链进行核对，并检查依赖项修订版本。它会将请求绑定到规范输入，并记录相关应用程序和形式化源代码的哈希值。运行生成的证明后，它会检查已绑定的源代码和制品是否保持稳定。意外的依赖项变更、证明失败或超时都会使主张悬而未决，并且不会颁发证书。

每个必需定理都要接受传递公理审计。允许的逻辑依赖项为 `propext`, `Classical.choice` and `Quot.sound`。意外假设会阻止证书签发。证明还必须针对输入摘要输出特定的经审计的完成标记。仅进程成功退出并不足够。源代码防护检查还会单独检查项目库中是否存在证明占位符。

`statclaim check-certificate bundle` 验证该捆绑包，重建派生字段，并针对其固定范围重新生成预期的证明陈述。重放需要绑定的来源版本和依赖项版本，并重新运行证明和审计。更改元数据不能将设计证书变成对观测到的 p值或更强误差界的证明。旧记录和制品哈希属于溯源记录。重放会根据绑定的定义重新检查保留的证明，从而获得数学证据。

信任边界包括 Leankernel 和导入制品、定义的统计含义，以及将声明转换为陈述并将其绑定到报告的应用代码。哈希可以检测内容变更，但无法验证作者身份，而且捆绑包没有经过数字签名。独立推导检查器、回归测试和重放检查会检查该边界的部分内容，但并不构成对整个软件栈的形式化验证。

5. 规划与运行特性

5.1. 目标与候选设计

规划规定了一项分析、参数值和一个目标。样本量搜索还规定了一组$\mathcal{D}$ 候选设计。每个候选方案 $d \in \mathcal{D}$ 都有一个抽样计划和与之关联的检验。设 $P_{\theta,d}$ 表示候选方案d 在参数值 θ下记录数据的概率律，并设 R_d 表示其检验拒绝零假设的事件。规划

点 θ_1 规定了一个备择效应以及计算拒绝概率所需的其他参数，例如基线响应率或结果方差。在该点的检验效能为

$$\pi(d, \theta_1) = P_{\theta_1, d}(R_d). \tag{6}$$

效能目标规定了最低检验效能 $1 - \beta$ 和第一类错误的最大值 α_0，其中$0 < \beta < 1$ 且 $0 \leq \alpha_0 \leq 1$。此处的 β 是在规划点未能拒绝零假设的最大概率。候选方案必须满足 $\pi(d, \theta_1) \geq 1 - \beta$，并且在其已声明的零假设下满足所请求的误差界。零假设和规划备择假设分别指定。当需要备择假设时，StatClaim 会拒绝零假设内的点。检验效能主张适用于指定的规划点。

导入 `planning`后，可以通过样本量搜索扩展入门 Fisher 分析。

```
let alternative = binomial_rates(p_control=1/10, p_treatment=2/5)
let target = power_target(at=alternative, minimum=4/5,
maximum_typeI=1/20)
let candidates = equal_arm_sizes(analysis=analysis,min_per_arm=2,
 max_per_arm=40)
let sizing = smallest_sample_size(candidates=candidates,
target=target)let recommendation = calculate(sizing)
let sizing_verification = verify(sizing)
```

在 StatClaim Studio 中打开完整程序。规划点将对照组和治疗组的响应率分别设为 0.1 和 0.4。目标要求检验效能至少达到 80%，且第一类错误界不超过 5%。在已声明的范围内，每组 36 人是满足要求的最小样本量。计算选择了这一样本量。验证会检查该样本量是否达到目标，并确认范围内所有更小的候选方案都未达到目标。最小性适用于每组 2 至 40 名参与者的等比例分配。由于离散检验的检验效能不一定随样本量增加而提高，因此会逐一检查每个更小的候选方案。

`sample_size` 请求一个满足要求的候选方案。 `smallest_sample_size` 还请求在适配器支持时验证最小性。根据设计不同，样本量可以按参与者、簇、匹配对或事件计。若没有候选方案满足目标，搜索会报告在已声明的范围内未找到满足要求的样本量。分析水平和目标的误差限值α_0 会分别检查。当错误定理将第一类错误界定为不超过分析水平时，若要依据该界限满足目标，分析水平必须不高于 α_0。

5.2. 精确、近似和自适应规划

精确二项规划会在有理规划点处对拒绝概率求和。受支持的形式化适配器会在 Lean 中检验相应的检验效能和错误主张。两阶段单臂设计会规定中期和最终规则，并评估在整个策略下拒绝的概率。计数规则会将响应计数与已声明的截断值进行比较。后验规则会指定一个参数为正整数的贝塔先验，以及一个参考响应率。当后验概率达到已声明的阈值，表明响应率超过该参考响应率时，规则会继续或拒绝。在抽样模型下，根据可能的数据和停止决策评估第一类错误和检验效能

。形式化后验规则使用有限二项和来定义这一概率。该库并未证明它与相应的贝塔分布积分相等。

对 Simon 型两阶段设计的搜索（Simon1989）可以在零假设下使期望样本量最小，或在已声明的候选方案中使可能的最大样本量最小。当前证书会检查所选设计是否满足其错误率和功效目标。系统会计算候选方案按样本量的排序，但不会提供正式的最优性证书。

引用的方法使用规划公式，例如正态近似或利用非中心 t 分布进行计算。其效能估计会在规划方法的假设下报告为 `calculated`。StatClaim 不会正式验证这些近似的准确性。不受支持的规划操作会生成诊断信息。

5.3. 模拟及其不确定性

模拟会在指定的数据分布下生成独立研究，并对每项研究应用已声明的检验。拒绝的研究所占比例会估计该情景下的拒绝概率。蒙特卡洛不确定性描述了由模拟研究数量有限所导致的不确定性（Morris、White 和 Crowther 2019）。下面的自包含程序将效能公式与合并方差双样本 t 检验的模拟估计进行比较。

```
use models.two_arm_continuous
use sampling
use tests.t
use planning

let study = two_arm_continuous(n_control=40, n_treatment=40)
let analysis = t_test(design=fixed(model=study),
null=equal_means(model=study), alpha=1/20)
let target = power_target(at=mean_difference(value=5, sd=10),minimum=4/5,
 maximum_typeI=1/20)
let formula = calculate(meets(analysis=analysis, target=target))
let estimate = simulate(meets(analysis=analysis, target=target),seed=2026,
 runs=10000)
```

在 StatClaim Studio 中打开此示例。此情景的结局服从正态分布，均值差为 5，共同标准差为 10。每个组有 40 名参与者，模拟使用 10,000 项研究。拒绝比例以 95% Clopper–Pearson 区间（Clopper 和 Pearson 1934）进行报告。当区间的下端点至少达到所请求的最低值时，目标中的检验效能部分通过。当上端点低于该最低值时，则不通过。否则，功效评估仍无法确定。

受支持的样本量搜索会检查候选方案，并使用独立随机流确认通过评估的候选方案。结果会报告通过此项评估的样本量，但无法证明该样本量是达到足够真实效能的最小样本量。系统会保留种子、运行次数、情景详情和模拟引擎版本，以便复现。

受支持的情景包括方差不等、结局分布偏斜、集群大小不等，以及生存研究中治疗效应延迟出现的情况。对于计划进行期中分析的设计，将按阶段逐一模拟。报告将涵盖完整程序的停止概率和期望样本量。

盲态样本量重新估计使用合并治疗组的期中方差估计更新计划样本量。当前适配器在明确提供的零假设情景下评估其错误率。通过模拟评估第一类错误时，对于目标的该组成部分，只有在每个提供的零假设点上，其区间的上端点均不超过 α_0，才能通过评估。评估仅适用于这些点。

家族模拟可以估计至少拒绝一个终点或拒绝所有终点的概率。它们还可以估计全局零假设下的家族错误率，此时每个终点的零假设都为真。强家族错误率控制要求对真、假零假设的每一种配置都设定上限。

5.4. 精度目标

精度规划需指定区间程序、置信水平、规划点和宽度标准。对于区间 $[L, U]$，其中 L 和 U 分别为下端点和上端点，绝对误差界为 $(U - L)/2$。对于比例和差值，相对误差界为 $(U - L)/(2|q|)$，其中 $q \neq 0$ 是所估计数量的规划值。对于比值，相对误差界为 $1 - \sqrt{L/U}$，根据区间在对数尺度上的宽度计算。该标准要求端点有限且满足 $0 < L \leq U$。下端点非正或上端点为无穷大的比值区间，无法达到有限的相对精度目标。

可选的保证概率设定了一个最低概率，即在规划点下，最终区间满足宽度标准。置信水平涉及未知参数的覆盖率，如公式 2中所定义。

以下程序规定了单臂二项响应率的精度。

```
use tests.binomial
use planning

let arm = one_arm_binomial(n=60)
let interval = planned_interval(model=arm, measure="response_rate")
let target = precision_target(at=response_rate(value=3/10), level=19/20,
 margin_of_error=1/10)let sizes = interval_sizes(interval=interval,
min_size=10, max_size=2000)
let needed = calculate(smallest_sample_size(candidates=sizes,
target=target))
```

在 StatClaim Studio 中打开此示例。目标采用 0.3 的规划响应率、95% 的置信水平，以及不超过 0.1 的绝对误差界。搜索范围为 10 至 2,000 名参与者。不考虑保证概率时，会根据规划点下的预期计数评估精度。考虑保证概率时，会枚举受支持的计数分布，以计算满足宽度标准的概率。对于仅固定总样本量的设计，若有相关记录，此计算会采用预期的组分配。

结果会指出区间方法是精确方法还是采用正态近似。当前所有精度搜索都采用数值计算，包括基于精确区间方法的搜索。区间程序具有正式的覆盖率保证，并不能验证数值精度搜索。

6. 创作界面与扩展

6.1. 命令行与编辑器集成

命令行支持编译、解释、报告和执行。 `compile` 检查声明， `explain`显示契约推导。 `report`汇总主张、假设和要求。`check` 执行所请求的计算、模拟和验证。结构化 JSON 输出支持程序化检查。以下命令演示了针对名为 `study.stat` 的文件的工作流程。

```
statclaim compile study.stat
statclaim explain study.stat
statclaim report study.stat
statclaim check study.stat --output run
statclaim check-certificate run/request-001
```

执行前，输出目录 `run` 必须不存在。如果第一个请求签发了形式化证书，最后一条命令会在 `run/request-001` 中重新检查其文件包。每个结果都会说明其范围，并列出该请求生成的文件。

语言服务器使用语言服务器协议（LSP），提供诊断信息、补全、悬停提示和源代码导航功能，适用于**Visual StudioCode**(微软 2026)。补全功能会使用导入的构造器、它们接受的参数以及语言对象的公共字段。诊断信息会指出有问题的参数，并在可用时指出提供其值的声明。对于不相容的零假设，它可以同时指出假设声明和使用该假设的分析。

扩展的“主张”视图会列出已声明的主张、开放要求和执行结果。结果会显示其假设、证据等级和可用证书。每项执行结果都关联到文档版本，以及本次运行所用源文本的哈希值。编辑文件或重启服务器后，先前的结果会被标记为过时。这些结果仍可作为历史记录显示，并保留其原始源版本。

6.2. StatClaim Studio

StatClaim Studio 是一款用于同一 `.stat` 语言的图形化块编辑器。引导式的新建研究工作流会要求研究者选择一种设计并提供其输入项。它会先检查生成的规范能否编译，再创建研究。在画布上，块表示构造器调用和对命名值的引用。连接线显示各输入项所接受的类型。参数可以直接编辑，文本源代码也仍可供检查。保存后会生成一个 `.stat` 文件，也可以从命令行运行。

该网页 Studio会打开本文中链接的完整程序。它可在浏览器中运行编译、计算、模拟和引用的方法检查。形式化证明

生成需要使用Lean ，无法在浏览器中进行。若要验证相同的程序，请在项目的 GitHub Codespace 中使用完整的 Studio，或使用本地安装，并配置计算细节中所述的形式化依赖。表9 列出了本文中的程序及其 Studio 链接。

服务器使用语言对象和构造器定义的共享注册表，获取参数类型、可用选项和公共字段。它会将选定的值作为输入，并为其余参数提供有代表性的值进行尝试，从而生成兼容性提示。例如，连续结局研究可以推荐 t 检验。Fisher 二项分析与该结局类型不兼容。不受支持的组合会显示编译器诊断。建议的替代检验使用相同的已声明零假设。

解析器会记录声明和参数的源代码范围。Studio 使用这些范围将代码块编辑转换为源代码编辑。通用编译器会检查每项编辑。运行控件使用与命令行相同的执行层，结果则显示假设、计算、证据等级和可用的证书。更改值会将先前的结果标记为过时。直接进行参数探索时，可以自动重新运行计算和模拟。形式证明会继续附属于先前的源代码，直到再次请求验证。

发行包包含一套引导式教程，共 45 节课程，并配有相应的 `.stat` 文件。课程涵盖声明和诊断信息、精确方法与引用的方法、组合、规划和模拟。教程可在浏览器中阅读，其源文件可通过命令行、编辑器或 Studio 使用。有些示例会有意触发诊断信息，或使要求保持未解决状态。附录A 总结了相关主题、代表性结果和软件要求。此外， `examples/`中还提供了可编辑设计。

6.3. 添加方法并保持一致性

方法通过在代码仓库中注册来添加。目前不支持外部插件加载。方法条目提供类型化构造器、支持的结局和零形式、假设与保证。在可用的情况下，还会注册用于计算和模拟的函数。该注册表还提供接口所使用的文档和说明。分类目录按角色对公共对象和构造器进行分组。

形式化验证支持需要数学定义和证明。适配器将受支持语言中的声明转换为使用这些定义的陈述。证书签发和重放必须针对请求的主张检查这些陈述。新方法的检验应涵盖兼容和不兼容的声明、未满足的要求、源位置以及具体计算。具有证明支持的方法还需要进行证明检查和拒绝篡改后的证书的检验。审查必须检查形式化定义和假设是否描述了预期的研究，以及定理是否确立了所请求的主张。

7. 验证与应用

复现材料包括对已发表的两阶段设计进行核查，以及对临床试验统计分析计划中选定部分进行编码。前者检验数值是否一致以及是否具备形式化保证。后者考察包含多个假设的方案如何表示，以及有哪些证据可用于支持其主张。

7.1. 重新核验 Simon 的两阶段设计

Simon (1989年提出了针对二元响应终点的单臂 II 期试验设计。当治疗显示活性不足时，这些设计允许提前停止，同时满足规定的第一类错误率和检验效能目标。我们利用已发表的设计，评估 StatClaim 对完整两阶段策略的计算和形式化验证。

论文的表 1 和表 2 共包含 102 项，涵盖 51 种参数设定下的最优设计和极小极大设计。每种设计都指定了第一阶段样本量 n_1、最大总样本量 n、中期截断值 r_1 和最终截断值 r。当第一阶段响应数不超过 r_1时，研究因无效性而停止。否则，研究继续纳入 n 名参与者，并在总响应数超过 r时拒绝零假设。原假设边界 p_0 和规划备择假设 p_1 均为响应概率，其中 $p_1 > p_0$。请求的第一类错误上限为 α，且在p_1 处的最低检验效能为 $1-\beta$。最优设计使 p_0 处的期望样本量最小。极小极大设计则使最大总样本量最小。

对于每一项，我们都对停止规则进行编码，重新计算其运行特性，并获得其错误率和检验效能界限的 Lean 证明。我们还核查数值设计搜索能否复现制表设计，并将计算结果与独立的 R 实现进行比较。这些证明涵盖了每个选定设计的运行界限。最优性则通过数值搜索进行评估。

第一个列表中的最优设计为 $n_1 = 9$, $n = 24$, $r_1 = 0$ 和 $r = 2$，其中 $p_0 = 0.05$,$p_1 = 0.25$ 和 $\alpha = \beta = 0.10$。其 StatClaim 规格说明为

```
use designs.two_stage
use planning

let stages = two_stage_binomial(n_stage1=9, n_stage2=15)
let rule = threshold_rule(continue_above=0, reject_above=2)
let analysis = two_stage_test(design=stages, rule=rule,null_rate=1/20)
let target = power_target(at=response_rate(value=1/4),minimum=9/10,
 maximum_typeI=1/10)
let checked = verify(meets(analysis=analysis, target=target))
```

在 StatClaim Studio 中打开此示例。 第二阶段样本量为 15，因为总样本量为 24。精确计算得出，在 p_0 时，拒绝概率为 0.09313；在p_1时，检验效能为 0.90284。 在 p_0 时，期望样本量为 14.546。验证证实，对于每个响应概率 $p \in [0, p_0]$，误差界均成立；并且在 p_1时，检验效能界成立。

StatClaim 还可以根据规划目标选择样本量和停止截断值。在上述规划的基础上，以下搜索将比较最大总样本量为 120 名参与者的设计。

```
let optimal_candidates = two_stage_designs(null_rate=1/20,max_total=120,
 criterion="optimal")
let optimal = calculate(sample_size(candidates=optimal_candidates,
target=target))let minimax_candidates = two_stage_designs(null_rate=1/20,
max_total=120, criterion="minimax")
let minimax = calculate(sample_size(candidates=minimax_candidates,
target=target))
```

在 StatClaim Studio 中打开完整的搜索程序。选定的设计为

标准	n_1	n	r_1	r	预期规模为 p_0
最优	9	24	0	2	14.546
极小极大	13	20	0	2	16.407

最优设计在零假设下的期望样本量较小。极小极大设计的最大总样本量较小。两种搜索均返回 `calculated` 结果。将`calculate` 替换为 `verify`，即可在 Lean 中检查所选设计的错误率和检验效能界限。按所选标准进行的排序仍是数值计算所得。

表4 汇总了所有条目的检查结果。精确计算满足所有给定的错误率和功效目标。对最多包含 120 名参与者的计数阈值设计进行数值搜索，找出了所有制表设计。形式化证书证实了所选设计的运行目标。搜索中候选方案的排序由数值计算得出，且不在最优性证明的涵盖范围内。

检查	条目	依据
错误率和检验效能目标均已达到	102/102	精确计算
在 Lean 中检查错误率和检验效能界限	102/102	形式化证明
有界搜索返回制表设计	102/102	计算
clinfun 返回制表设计	102/102	比较
打印的提前停止概率一致	98/102	比较

表4：对Simon表中102项内容的检查。每项内容都包含一个错误主张和一个检验效能主张。形式化证明针对完整停止策略下的这些主张。搜索和比较结果均为数值结果。

比较脚本还会运行 `ph2simon` 来自R 软件包**clinfun** (Seshan和Whiting 2026)，并采用相同的120例最大样本量。所有情况下，它都返回相同的设计。期望样本量的差异在0.005以内，提前停止概率的差异在0.00005以内。所有打印出的期望样本量均与重新计算结果一致，精度与复现中采用的精度相同。有四个打印出的提前停止概率与重新计算结果的差异超过0.005。重新计算出的概率与**clinfun**一致。

7.2. 慢性阻塞性肺疾病试验分析计划

第二个示例对研究 NCT03162055（阿斯利康2018）的 AERISTO统计分析计划 2.0版（日期为2018年6月8日）中的选定组成部分进行了编码。这是一项针对慢性阻塞性肺疾病（慢性阻塞性肺疾病）的试验。该计划规定每个

组纳入500名参与者，并预计在其符合方案分析中每组约有440名参与者。该编码涵盖主要检验层级和四项规划陈述。
主要检验族包含两个非劣效假设和一个优效假设。其单侧错误预算为 2.5%，最初平均分配给两个非劣效检验。拒绝假设后，分配的显著性水平可沿已声明的图传递。如果三个假设均被拒绝，则按固定顺序以 2.5% 的显著性水平继续检验四个次要终点。

该文件 `replication/copd/design_cited.stat`可在 StatClaimStudio 中获取，采用引用的连续结局混合模型契约，以及引用的二元结局逻辑回归契约。A `graphical` 声明代表主要检验族。A `fixed_sequence`声明将该检验族排在四项次要分析之前。表 5 记录了由此产生的证据。主要检验族的保证是引用的，且适用范围为大样本。该检验序列具有一个组合定理；在其各成员的保证成立这一条件下，该定理由 Lean 检验，且适用范围同样继承为大样本。文件中最弱的证据是引用的。第二种编码 采用作者声明的检验，并报告已声明的证据。

组成部分	表示	结果
单项检验	混合模型与逻辑回归契约	引用的
主要检验族	图形化分配与再循环	引用的
完整层级	从主要检验族开始的固定序列检验族	有条件的
规划陈述	数值功效计算	已计算

表5：慢性阻塞性肺疾病分析计划中编码组件的证据。完整层级的结果以其成员的保证为前提。

文件 `replication/copd/power.stat`也可在StatClaimStudio中获取，使用其提供的差值和标准差，对计划第1.3节中的四项陈述进行评估。第一秒用力呼气容积记为 FEV_1，，TDI 表示呼吸困难转换指数。对于 FEV_1，，计划给出了基线至 24周变化的有效标准差。表 6展示了计划中的功效陈述及计算值。最后一行采用计划中额外的规划情景，边际为 -0.75 个 TDI 单位。其单侧显著性水平取 2.5%，即次级层级规定的显著性水平。

陈述	计划中的陈述	计算出的检验效能
谷浓度 FEV_1 非劣效性，2.5%	约 94%	94.44%
谷值 FEV_1 非劣效性，1.25%	90%	90.51%
峰值 FEV_1 优效性，1.25%	至少 97%	97.52%
TDI 非劣效性，界值 -0.75	至少 90%	95.98%

表 6：COPD 计划的单项检验效能计算。这些是基于所提供规划输入值的正态近似计算。它们按各项检验所注明的显著性水平进行评估。

计算得出的检验效能与全部四项陈述一致。对于 1.25% 水平下的谷值 FEV_1，计算得出的拒绝阈值为 -24.76 mL。计划中提到的是 -24.5 mL，两者相差约 0.26 mL。这些计算针对单项检验。此处未评估完整检验层级的联合效能。标准差属于规划输入值，报告保留了引用的方法所采用的模型和缺失数据假设。

分数　这种编码仅限于已声明的组成部分。详细的协变量、 -

方差设定、备用分析以及分析集成员资格规则均未编码。其他不决定次要层级推进的优效性检验也未纳入。慢性阻塞性肺疾病晨间症状量表（EMSCI）严重程度评分的界值 -0.1 按原文编码，因此将较高分数视为更好。方案和分析计划均未说明这一方向，报告则将其列为一项假设。如果较高分数表示症状更严重，那么非劣效性零假设应改为治疗差异至少为 0.1。家族错误率声明涉及的是已编码的序列。我们未核查方案与分析计划是否一致，也未分析任何观测到的试验结局。

8. 总结与讨论

StatClaim 是一种声明式领域特定语言，用于将研究规范表述为程序。模型、抽样计划、假设和分析均声明为类型化对象，契约则用于检查它们的兼容性。对于受支持的验证请求，StatClaim 会生成证明义务，并调用 Lean 来构造和检查证明。

研究人员可以在方案制定过程中使用 StatClaim，计算拟议设计的样本量、检验效能和精度。这些规划请求引用已声明的模型和分析，因此其输入和假设仍是研究规范的一部分。修订设计后，重新编译会检查更新后的声明，后续执行则会评估修订后的主张。因此，程序及其结果可以作为指定和检查内容的记录，随方案或统计分析计划一并提交。审稿人可以通过命令行、编辑器或 Studio 检查这份记录，并根据记录的输入和依赖项重放形式化证书。

研究规范可能需要一些尚无形式证明的方法。StatClaim 通过证据等级标明每项保证的依据，从而支持这些方法。例如，条件组合证明可以在假设各组成检验的保证成立的前提下，确立一个检验族保证。这些组成检验的保证本身可以经过形式化证明、引用自文献，或由作者声明。

第7节中的示例展示了这两种用途。Simon复现选择了两阶段样本量和停止规则，并验证了它们的错误率和检验效能界限。慢性阻塞性肺疾病示例使用引用的方法和条件组合证明，对已发表的分析计划进行了部分编码。

这些结果的解释取决于编码的研究与实际开展的研究之间是否对应。StatClaim 无法确定已声明的抽样模型是否描述了招募人群，也无法确定终点是否在检查结局之前选定。研究人员必须说明这些选择的依据，并评估记录的假设是否成立。形式化验证还取决于这些定义是否代表预期的统计程序。从声明到证明陈述的转换以及验证工具链，仍属于第4节所述的信任边界。

在此范围内，当前实现的覆盖率和计算能力有限。部分程序支持形式化验证，而精确枚举和内核计算限制了某些适配器可处理的问题规模。StatClaim 还支持近似规划、模拟和精度搜索，其数值结果的适用范围如第5节所述。

要扩大这一覆盖范围，需要为区间程序、精度搜索和自适应采样策略提供新的证明。每项扩展都需要明确的观测定律，并精确陈述要建立的保证。连续型方法和大样本方法还需要进一步的分布论证和渐近论证。还需要对新模块进行统计和形式化审查，以确认这些定义和证明能够支撑其所宣称的主张。

随着方法库不断发展，可复用的程序组件和版本化扩展接口将使方法的添加与维护更加便捷。未来的工作将探索精化类型和依赖类型，以便在 DSL 的契约层中更精确地表达模型标识和抽样策略等共享要求。形式化后端已经使用 Lean 的依赖类型系统来表示陈述并检查证明。计划中的开发工作还包括一个 R 包，用于直接从 R 脚本调用 StatClaim 规格说明，使研究人员能够将设计检查和规划计算纳入现有工作流程。

StatClaim 是一个依据 MIT 许可证发布的开源项目，旨在通过社区贡献不断发展。其模块化的方法定义、共享注册表和独立的形式化适配器经过结构化设计，可支持对语言、统计方法和创作工具进行扩展。贡献指南记录了如何添加和测试这些扩展，为统计学家、形式化方法研究人员和软件开发人员提供了共同的起点。

计算细节

可执行示例已使用 Python 3.12.15、statclaim 0.1.0（语言版本 0.41.0）和 Lean 4.35.0 进行检查。Python 依赖版本锁定记录在 `uv.lock` 中。形式化工具链和库版本锁定见 `lean-toolchain` 和 `lake-manifest.json`。形式化验证和证书重放需要项目检出副本及其形式化依赖。从该检出副本中， `make setup` 会准备受支持的开发环境。随后，可以使用前缀 `uv run --locked` 运行正文中的命令。运行时报告会保留源代码摘要和后端版本，因为仅凭软件包版本无法标识特定的开发快照。配套脚本会运行所展示的示例并检查诊断变体。使用 `--lean` 后，它还会验证形式化请求、比较打印输出，并重放一个观测结果证书：

```
uv run --locked python \
  statClaim_paper/manuscript/check_examples.py --lean
```

第7 节中的验证结果通过以下方式复现： `make replicate`用于计算和存储输出检查，并通过 `make replicate-lean`生成新的形式化证明。与 **clinfun** 的 Simon 比较使用单独的 R 脚本 `replication/comparison/simon_clinfun.R`。该脚本的软件包要求和记录的输出均存放在同一目录中。手稿中的验证表和数值由以下工具生成：

```
uv run --locked python \
  statClaim_paper/manuscript/make_tables.py --validation
```

参考文献

阿斯利康（2018）。“统计分析计划，研究 D5970C00002（AERISTO）：一项随机、双盲、双模拟、多中心、平行组研究，旨在评估格隆溴铵/富马酸福莫特罗固定剂量复方制剂相较于乌美溴铵/维兰特罗固定剂量复方制剂，在中度至极重度慢性阻塞性肺疾病患者中为期24周的疗效和安全性。”第2.0版，2018年6月8日。随 ClinicalTrials.gov 记录 NCT03162055 发布，网址 https://cdn.clinicaltrials.gov/large-docs/55/NCT03162055/SAP_001.pdf。Blair G、Cooper J、Coppock A、Humphreys M（2019）。“声明并诊断研究设计。”《美国政治科学评论》，113(3)，838–859。doi:10.1017/S0003055419000194。Clopper CJ、Pearson ES（1934）。“以二项分布为例说明置信限或枢轴限的应用。”《生物计量学》，26(4)，404–413。 doi:10.1093/biomet/26.4.404。Coppock A、Blair G、Cooper J、Humphreys M、Fultz N（2026）。DeclareDesign：基于 Tidyverse 的研究设计声明与诊断工具。R 包版本 2.0.0，网址https://declaredesign.org/r/declaredesign/。de Moura L、Ullrich S（2021）。“Lean 4 定理证明器与编程语言。”载于《自动推理——CADE 28》，第625–635页。施普林格出版社。doi:10.1007/978-3-030-79876-5_37。Freeman T、Pfenning F（1991）。“ML 的精化类型。”载于《ACM SIGPLAN 1991 年编程语言设计与实现会议论文集》，第268–277页。ACM。 doi:10.1145/113445.113468。Kawamoto Y、Kobayashi K、Suenaga K（2025）。“StatWhy：用于统计假设检验程序的形式化验证工具。”载于《计算机辅助验证——CAV 2025》，《计算机科学讲义》第15932卷，第216–230页。施普林格出版社。 doi:10.1007/978-3-031-98679-6_10。Manathunga S（2026）。statclaim：具有分级证据的带类型统计主张。Python 软件包版本 0.1.0，网址 https://github.com/ssm123ssm/statClaim。微软（2026）。语言服务器扩展指南。Visual Studio Code 文档，网址 https://code.visualstudio.com/api/language-extensions/language-server-extension-guide。Morris TP、White IR、Crowther MJ（2019）。“使用模拟研究评估统计方法。”《医学统计学》，38(11)，2074–2102。 doi:10.1002/sim.8086。Rondon PM、Kawaguchi M、Jhala R（2008）。“液体类型。”载于《ACM SIGPLAN 2008 年编程语言设计与实现会议论文集》，第159–169页。ACM。 doi:10.1145/1375581.1375602。Seshan V、Whiting K（2026）。clinfun：临床试验设计与数据分析函数。doi:10.32614/CRAN.package.clinfun。R 包版本 1.1.6。

Simon R（1989年）。“II期临床试验的最优两阶段设计。”《受控临床试验》，**10**（1），1–10。doi:10.1016/0197-2456(89)90015-9。

mathlib 社区（2020年）。“Lean 数学库。”载于第九届*ACM SIGPLAN*认证程序与证明国际会议论文集。ACM。doi:10.1145/3372885.3373824。

A. 教程与示例覆盖率

软件存档中包含一份引导式教程，位于 `docs/tutorial.html` ，可运行源代码位于 `examples/tutorial/`。共有45节编号课程和46个源文件，因为第3课针对不相容的零假设和未指定的自适应停止分别提供了示例。表7按主题对课程进行了分组。

课程	主题
1–8	声明、主张、结果字段、诊断信息、观测数据、多重性，样本量搜索和非劣效性。
9–18	单臂设计和两阶段设计、后验停止规则、引用的方法，期中分析、组合和近似效能。
19–27	共享分配、簇、重复测量值、随机化，分析人群和观察性设计。
28–36	模拟效能、观察性研究与试验样本量、回归、调查规划与区间精度。
37–45	不等集群、偏度和延迟效应的模拟、期中分析、样本量重新估计、家族效能和精确规划。

表7：引导式教程的主题覆盖率。

表 8 列出了选定的请求以及需要检查的结果。这些结果对应于所提供的文件。某些课程会有意生成诊断信息或未解决的要求。执行环境见第 6.2节。

源文件	选定的结果	Lean
`01_first_claim.stat`	已验证的第一类错误界	Yes
`03_wrong_null.stat`	不兼容的假设诊断信息	No
`03_optional_stopping.stat`	未解决的抽样要求	No
`07_sample_size.stat`	已验证的最小样本量	Yes
`12_continuous_outcome.stat`	引用的检验保证	No
`13_mixed_endpoints.stat`	条件族保证	Yes
`28_simulated_power.stat`	计算的和模拟的检验效能	No
`36_precision.stat`	计算区间精度	No

表 8：具有代表性的教程请求及其计算需求。文件名称相对于 `examples/tutorial/`。第13课中的有条件的检验族，其成员中包含引用证据。

这些文件使用第6节中所述的编译和执行命令。它们也可以在编辑器中打开，或导入 Studio。生成的参考文档 `docs/catalogue.md` 记录了分发软件版本的构造函数可用性、参数和限制。

表 9 列出了正文中使用的全部程序。每个链接都会在网页 Studio 中打开相应的源文件。复现材料中也包含这些源文件。

示例	源文件	Studio
Fisher 设计保证	`examples/paper/fisher_design.stat`	Open
未指定停止	`examples/paper/fisher_adaptive.stat`	Open
模型组合	`examples/paper/composed_model.stat`	Open
观测 Fisher 结果	`examples/paper/fisher_observed.stat`	Open
Fisher样本量	`examples/paper/fisher_sample_size.stat`	Open
模拟效能	`examples/paper/t_test_simulation.stat`	Open
区间精度	`examples/paper/response_rate_precision.stat`	Open
Simon设计保证	`examples/paper/simon_design.stat`	Open
Simon设计选择	`examples/paper/simon_search.stat`	Open
慢性阻塞性肺疾病引用方法	`replication/copd/design_cited.stat`	Open
慢性阻塞性肺疾病声明方法	`replication/copd/design_declared.stat`	Open
慢性阻塞性肺疾病检验效能	`replication/copd/power.stat`	Open

T表9：本文演示和案例研究的源文件及 Studio 链接 .

所属机构：
Supun Manathunga 实验医学部 麦吉尔大学 蒙特利尔，魁北克省 加拿大 电子邮箱：
supun.manathunga@mail.mcgill.ca