

The Impact of Coverage on Bug Density in a Large Industrial Software Project

The 11st International Symposium on
Empirical **S**oftware **E**ngineering and **M**easurement (ESEM2017)



Thomas Bach



Artur Andrzejak



Ralf Panneman



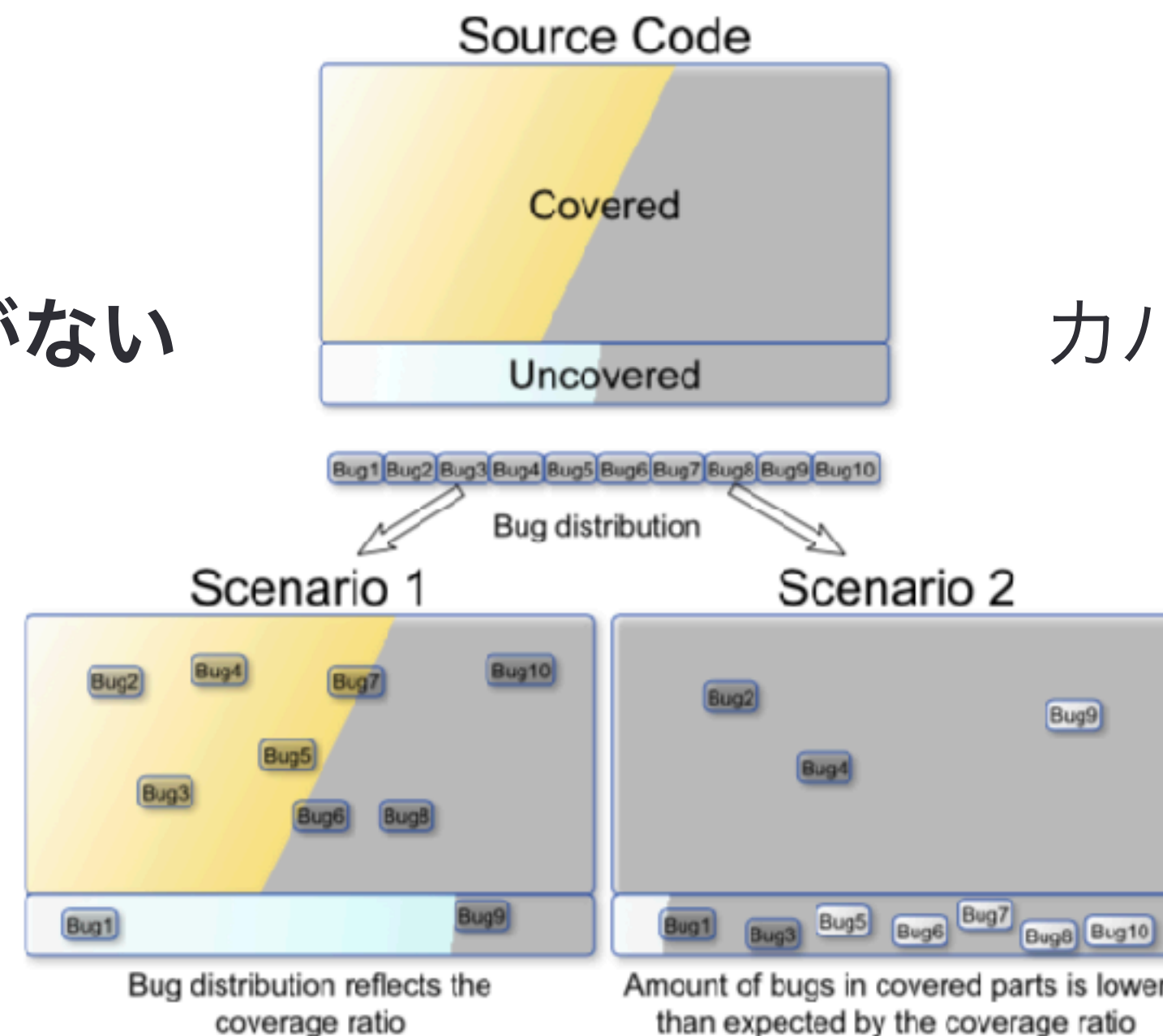
David Lo

背景

- ソフトウェアテストは品質保証に欠かせないツールである
 - テストの質は、カバレッジで測られることが多い
 - しかしながら、**カバレッジの効果は疑問視されている**
⇒カバレッジ率と不具合発見の関係を明らかにする

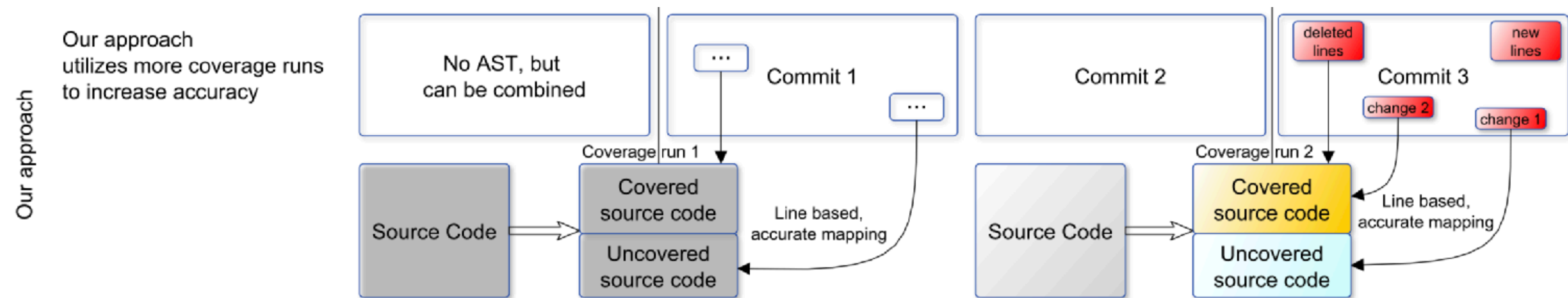
カバレッジ率と不具合出現率に**相関がない**
⇒カバレッジは**効果なし**

カバレッジ率と不具合出現率に**相関がある**
⇒カバレッジは**効果あり**



1. ソースコードがテストされているか判定

- コミットの変更行が1行でもテストがされていたら、『Covered source code』として扱う



2. Covered/Uncovered source codeの不具合数を比較

- カバレッジ計測時（計72回）ごとに比較・検定

- SAP HANAの開発データ
 - データ期間：May 2016 ～ April 2017
 - カバレッジ計測回数：72回
 - 不具合修正Commit 件数：16,215
 - 不具合件数：???
 - ソースコード数：
 - » LOC：数百万
 - » セクション数：76,979
 - » チャンク数：239,119



Table II
NUMBERS OF BUG-FIXING COMMITS (BUG-FIXES) IN OUR DATA SET BY
CATEGORIES DEFINED IN SECTION II-D.

Metric	Number	Percentage
Total number of bug-fixes	16 215	100%
Bug-fixes in covered source code	8 348	51.48 %
Bug-fixes in uncovered source code	6 171	38.06 %
Bug-fixes with undecided coverage	1 696	10.46 %

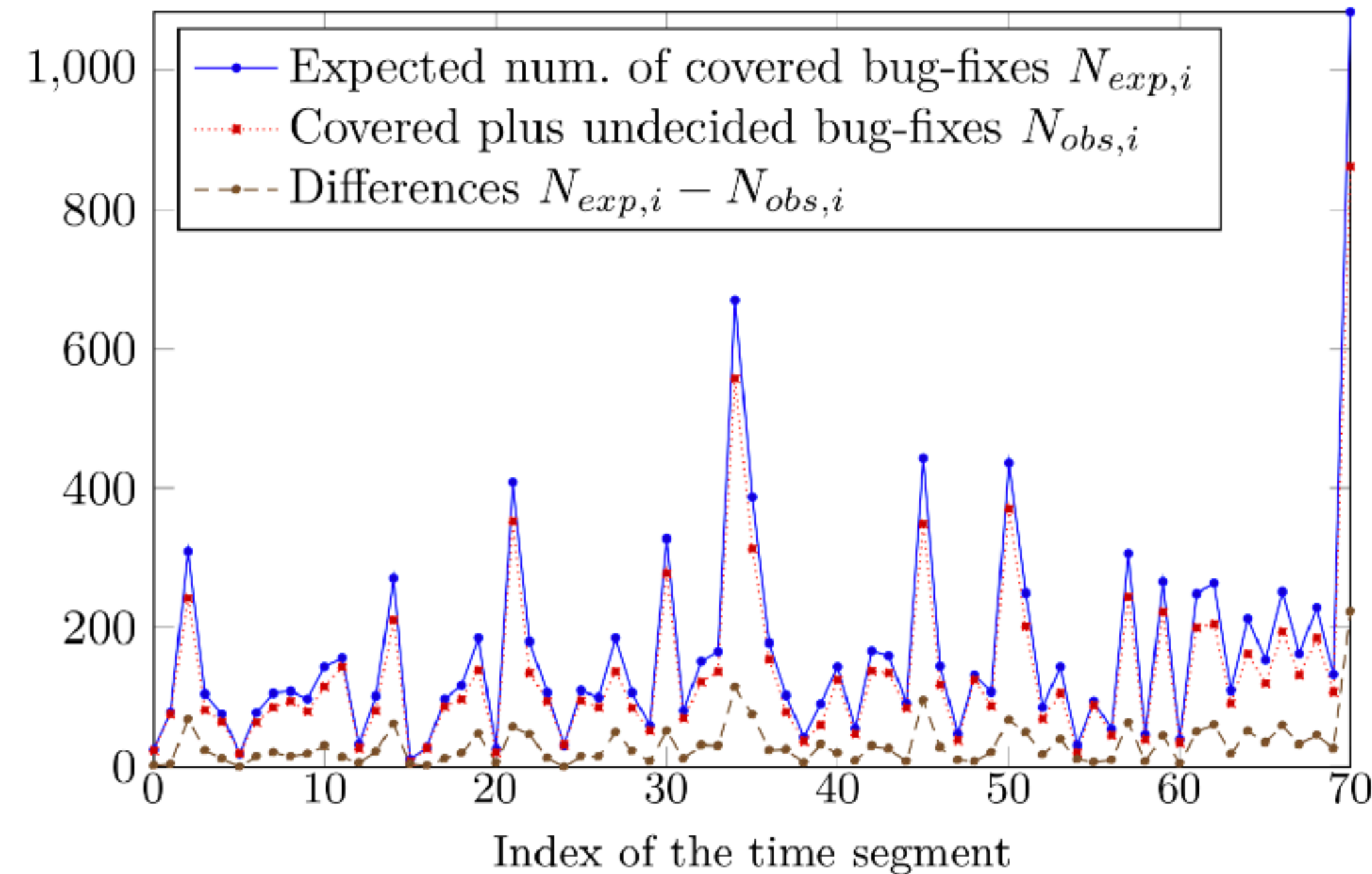
評価方法

注意：Covered/Uncovered部分を単純比較はしない

1. カバレッジが効果がないと想定される値 (N_{exp}) を算出
 - 帰無仮説であり、閾値になる
 - 多分、Uncovered部分から（論文に記載なし）
2. カバレッジ部分で出現した値 (N_{obs}) と N_{exp} を比較
3. ウィルコクソンの符号順位検定を実施

評価結果

- 72回のカバレッジ計測中, $N_{obs} < N_{exp}$ が70箇所

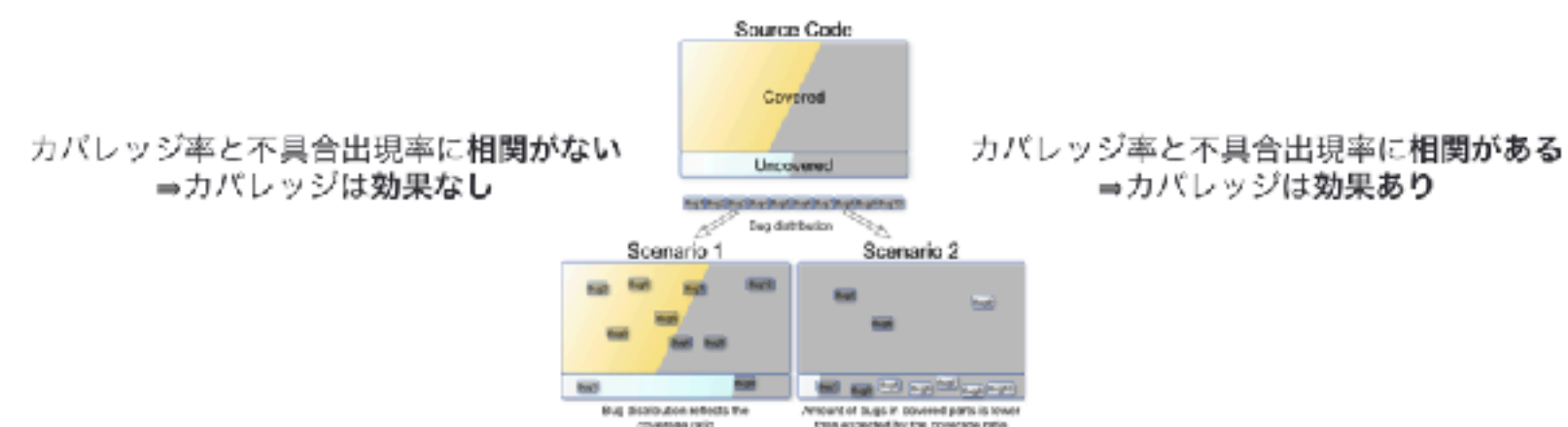


- 検定の結果, p値は $2.2e-16$, effect-sizeは $0.612(>0.5)$ で有意差あり
 - ちなみに, ウィルコクソン・マンホイットニー検定では, p値が 0.06 で有意差なし (ただし, effect-sizeは 0.13)

背景

2

- ソフトウェアテストは品質保証に欠かせないツールである
 - テストの質は、カバレッジで測られることが多い
 - しかしながら、**カバレッジの効果は疑問視されている**
⇒カバレッジ率と不具合発見の関係を明らかにする



アプローチ

3

- ソースコードがテストされているか判定
 - コミットの変更行が1行でもテストがされていたら、『Covered source code』として扱う

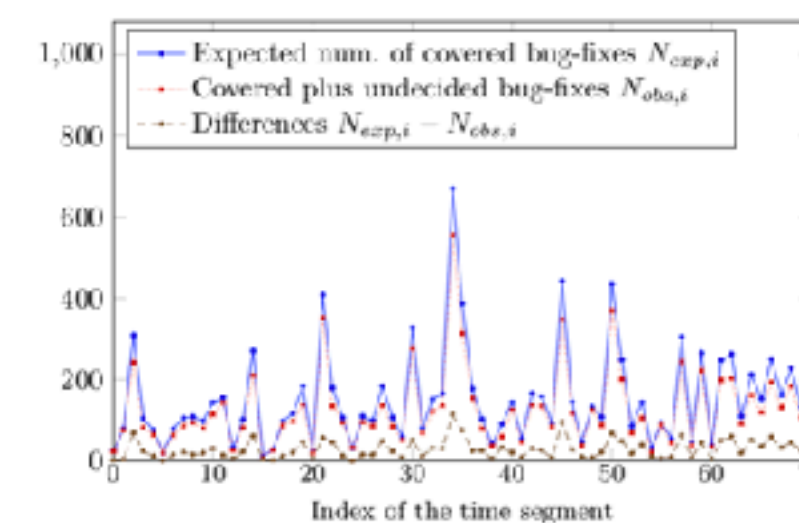


- Covered/Uncovered source codeの不具合数を比較
 - カバレッジ計測時（計72回）ごとに比較・検定

評価結果

6

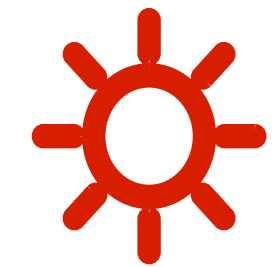
- 72回のカバレッジ計測中、 $N_{obs} < N_{exp}$ が70箇所



効果あり

- 検定の結果、p値は $2.2e-16$, effect-sizeは $0.612(>0.5)$ で有意差あり
 - ちなみに、ウィルコクソン・マンホイットニー検定では、p値が 0.06 で有意差なし（ただし、effect-sizeは 0.13 ）

所感



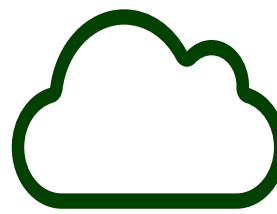
論文の書き方が興味を誘うような書き方

- 2つのシナリオが用意されている
- うだうだ書かず、2章にアプローチ（最近多い気がする．．．）
- 図が多い（少しわかりにくいですが、印象が良い）



中身が薄い

- カバレッジ率と不具合数の関係もよくわからない
- 数値的な議論だけで、出てきた不具合がどんなものかもわからない



そもそもカバレッジが低いところは、テストが要らないところでは？