Calibration of P-values for calibration and for deviation of a subpopulation from the full population

Mark Tygert

March 24, 2022

Abstract

The author's recent research papers, "Cumulative deviation of a subpopulation from the full population" and "A graphical method of cumulative differences between two subpopulations" (both published in volume 8 of Springer's open-access *Journal of Big Data* during 2021), propose graphical methods and summary statistics, without extensively calibrating formal significance tests. The summary metrics and methods can measure the calibration of probabilistic predictions and can assess differences in responses between a subpopulation and the full population while controlling for a covariate or score via conditioning on it. These recently published papers construct significance tests based on the scalar summary statistics, but only sketch how to calibrate the attained significance levels (also known as "P-values") for the tests. The present article reviews and synthesizes work spanning many decades in order to detail how to calibrate the P-values. The present paper presents computationally efficient, easily implemented numerical methods for evaluating properly calibrated P-values, together with rigorous mathematical proofs guaranteeing their accuracy, and illustrates and validates the methods with open-source software and numerical examples.

1 Introduction

Two basic problems in statistics are (1) checking calibration of probabilistic predictions such that any event predicted to happen, say, x percent of the time actually occurs x percent of the time and (2) assessing the deviation of a subpopulation from the full population while conditioning on a specified covariate or score ("conditioning on" is also known as "controlling for," and involves comparing only individuals whose values for the covariate or score are similar or otherwise match up). In fact, the first problem can be viewed as a special case of the second problem by requiring the expected response of the full population to be equal to the predicted probability, so that the deviation of the subpopulation from the full population is simply the deviation from the probabilities. In all cases, the data consists of observations of responses paired with scores (and weights, in the case of weighted samples). In the first case, the scores are the predicted probabilities; in the second case, the scores are the values of the specified covariate (which could be probabilistic predictions, too). In the social and biomedical sciences, controlling for income or age is common.

Recent work of [11] and [12] proposes metrics for (inter alia) measuring miscalibration or deviation of a subpopulation from the full population, reviewed in Subsection 2.2 below. The present paper develops methods for converting the values of such metrics into properly calibrated attained significance levels (also known as "P-values"), deriving the cumulative distribution functions for the metrics under the null hypothesis of no deviation between the subpopulation and the full population (or of perfect calibration in the underlying subpopulation). Technically speaking, the calibration of P-values given below is conservative, yielding estimates distributed at slightly greater values than the actual P-values, but the works of [3], [4], and [10] which form the basis of the estimates also prove that the conservative estimates converge reasonably rapidly to the exact values in most settings encountered in practice, as confirmed in the numerical experiments presented below. The metrics discussed in the present paper are very similar to those of Kuiper from [7] and of Kolmogorov and Smirnov from [6] and [9], and the estimates are conservative in precisely the same sense that the classical, widely employed asymptotic estimates for P-values of Kuiper or Kolmogorov-Smirnov statistics are conservative relative to the exact, finite-sample P-values. Figures 7 and 8 below illustrate the mild conservatism and rapid convergence.

For the most part, the present paper amounts to little more than an exposition of widely deployed prior work, elaborating details that earlier publications omitted. The elaboration is for the convenience and reference of the reader; the reader undoubtedly could derive all results presented below, but is welcome to

spare the effort required by instead leveraging the present paper and the associated open-source software. The presentation below provides full proofs that earlier publications omitted, and also summarizes everything required to solve the problems posed here, rather than requiring the reader to traverse literature that spans many decades and disciplines. The novelty of the results presented below is not extremely high; the present paper is merely a response to many requests for pulling together everything into a comprehensive, convenient, reasonably elementary exposition. In particular, Subsection 2.2 below briefly reviews the cumulative methods of [11] for assessing the deviation of a subpopulation from the full population; readers unfamiliar with that approach may wish to start with the full paper of [11] or the summary in Subsection 2.2 below.

The remainder of the paper has the following structure: Section 2 presents the main methods, Section 3 validates and illustrates the methods via numerical examples, Section 4 briefly discusses the results and draws conclusions, and the appendix lists code in Python 3 that implements the methods and automatically reproduces the numerical results (including the figures) of Section 3.[1]

2 Methods

The present section details the methodology of the present paper. Subsection 2.1 provides computationally efficient formulae for evaluating the cumulative distribution functions of the range and of the maximum absolute value of the standard Brownian motion over the unit interval $[0, 1]$, based on the works of [5] and [2]. Subsection 2.2 reviews the methods of [11] for assessing deviation of a subpopulation from the full population and for assessing calibration of probabilistic predictions, introducing a graphical method along with two statistics which summarize the graph as scalars. Finally, Subsection 2.3 shows how to use the numerical methods of Subsection 2.1 to calculate attained significance levels (P-values) for the scalar summary statistics introduced in Subsection 2.2, based on the works of [3], [4], and [10]. Readers unfamiliar with the work of [11] or [12] might want to skip to Subsection 2.2 at first; however, readers interested mainly in the numerical methods might want to start instead with Subsection 2.1.

2.1 Distributions of the range and maximum absolute value of Brownian motion

This subsection presents series for the cumulative distribution functions of the range and maximum absolute value of the standard Brownian motion over the unit interval $[0, 1]$. The terms in the series consist entirely of elementary functions that are easy to program (as implemented in the codes mentioned in Section 3 and listed in the appendix). The series converge rapidly and the present subsection proves rigorous bounds on the numbers of terms required to attain a specified accuracy. Subsubsection 2.1.1 gives the results for the range of the standard Brownian motion — see especially Theorems 3 and 4; Subsubsection 2.1.2 gives the results for the maximum absolute value — see Theorems 5 and 6.

2.1.1 Range of the standard Brownian motion

This subsubsection presents Theorems 3 and 4, enabling easy, rapid computation of the cumulative distribution function for the range (the maximum minus the minimum) of the standard Brownian motion over the unit interval $[0, 1]$.

We define the series

$$F(x) = \sum_{k=1}^{\infty} \left(\frac{8}{x^2} + \frac{8}{(2k-1)^2 \pi^2} \right) \exp\left(-\frac{(2k-1)^2 \pi^2}{2x^2} \right) \tag{1}$$

for any positive real number x. The following theorem exhibits F to be the cumulative distribution function associated with the probability density function of Formulae 3.6–3.8 of [5] (Theorem 2 below reviews those formulae).

Theorem 1. *Suppose that F is the series defined in (1). Then,*

$$F(x) = \int_0^x f(y)\, dy \tag{2}$$

[1] The software is also available for download at `http://tygert.com/dists.py`.

for any positive real number x, where

$$f(x) = \sqrt{\frac{2}{\pi x^2}} \cdot \frac{\partial G}{\partial x}\left(\frac{x}{2}\right), \tag{3}$$

with

$$G(x) = \frac{\sqrt{2\pi}}{x} \sum_{k=1}^{\infty} \exp\left(-\frac{(2k-1)^2\pi^2}{8x^2}\right). \tag{4}$$

Proof. Clearly $\lim_{x\to 0^+} F(x) = 0 = \lim_{x\to 0^+} \int_0^x f(y)\,dy$, so we need only show that $\frac{\partial F}{\partial x} = f$.

Differentiating (4) yields

$$\sqrt{\frac{2}{\pi}} \cdot \frac{\partial G}{\partial x} = \sum_{k=1}^{\infty} \left(\frac{2}{x}\frac{(2k-1)^2\pi^2}{4x^3} - \frac{2}{x^2}\right) \exp\left(-\frac{(2k-1)^2\pi^2}{8x^2}\right), \tag{5}$$

which when combined with (3) yields

$$f(x) = \sum_{k=1}^{\infty} \left(\frac{8(2k-1)^2\pi^2}{x^5} - \frac{8}{x^3}\right) \exp\left(-\frac{(2k-1)^2\pi^2}{2x^2}\right). \tag{6}$$

Differentiating (1) yields

$$\frac{\partial F}{\partial x} = \sum_{k=1}^{\infty} \left[\left(\frac{8}{x^2} + \frac{8}{(2k-1)^2\pi^2}\right)\left(\frac{(2k-1)^2\pi^2}{x^3}\right) - \frac{16}{x^3}\right] \exp\left(-\frac{(2k-1)^2\pi^2}{2x^2}\right) \tag{7}$$

The right-hand sides of (6) and (7) are equal, completing the proof. □

Formulae 3.6–3.8 of [5] state the following theorem (though Formula 3.6 of [5] is missing a factor of $1/\sqrt{t}$).

Theorem 2. *The probability density function for the range (the maximum minus the minimum) of the standard Brownian motion over the unit interval $[0,1]$ is given by Formula (3).*

Combining Theorems 1 and 2 yields the following theorem.

Theorem 3. *The cumulative distribution function for the range (the maximum minus the minimum) of the standard Brownian motion over the unit interval $[0,1]$ is given by Formula (1).*

The following theorem upper-bounds the tail of the series for F defined in (1).

Theorem 4. *Suppose that n is a positive integer. Then, the tail of the series for F defined in (1) satisfies*

$$\sum_{k=n+1}^{\infty} \left(\frac{8}{x^2} + \frac{8}{(2k-1)^2\pi^2}\right) \exp\left(-\frac{(2k-1)^2\pi^2}{2x^2}\right) < \frac{4}{\sqrt{2\pi}}\left(\frac{1}{x} + \frac{x}{\pi^2}\right) \exp\left(-\frac{(2n-1)^2\pi^2}{2x^2}\right) \tag{8}$$

for any positive real number x. If ε is a positive real number less than 1 and

$$n \geq \frac{1}{2} + \frac{x}{\pi\sqrt{2}} \sqrt{\ln\left(\frac{4}{\varepsilon\sqrt{2\pi}}\left(\frac{1}{x} + \frac{x}{\pi^2}\right)\right)}, \tag{9}$$

then the right-hand side of (8) is at most ε.

Proof. Clearly,

$$\sum_{k=n+1}^{\infty} \left(\frac{8}{x^2} + \frac{8}{(2k-1)^2\pi^2}\right) \exp\left(-\frac{(2k-1)^2\pi^2}{2x^2}\right) < \left(\frac{8}{x^2} + \frac{8}{\pi^2}\right) \sum_{k=n+1}^{\infty} \exp\left(-\frac{(2k-1)^2\pi^2}{2x^2}\right), \tag{10}$$

$$\sum_{k=n+1}^{\infty} \exp\left(-\frac{(2k-1)^2\pi^2}{2x^2}\right) < \int_n^{\infty} \exp\left(-\frac{(2t-1)^2\pi^2}{2x^2}\right) dt, \tag{11}$$

$$\int_n^{\infty} \exp\left(-\frac{(2t-1)^2\pi^2}{2x^2}\right) dt = \frac{x}{\pi\sqrt{2}} \int_{((2n-1)\pi)/(x\sqrt{2})}^{\infty} \exp(-u^2)\, du, \tag{12}$$

and

$$\int_{((2n-1)\pi)/(x\sqrt{2})}^{\infty} \exp(-u^2)\, du \le \exp\left(-\frac{(2n-1)^2\pi^2}{2x^2}\right) \int_0^{\infty} \exp(-u^2)\, du = \frac{\sqrt{\pi}}{2} \exp\left(-\frac{(2n-1)^2\pi^2}{2x^2}\right). \tag{13}$$

Combining (10)–(13) yields (8). □

2.1.2 Maximum absolute value of the standard Brownian motion

This subsubsection presents Theorems 5 and 6, enabling easy, rapid computation of the cumulative distribution function for the maximum of the absolute value of the standard Brownian motion over the unit interval $[0, 1]$.

The following theorem states Formulae 3.8 and 5.2 of [2]; see also the displayed formula immediately before Formula 5.2 of [2] (or Formula 2.22 of [1] and the sentence of [1] immediately following).

Theorem 5. *The cumulative distribution function for the maximum of the absolute value of the standard Brownian motion over the unit interval* $[0, 1]$ *is*

$$D(x) = \frac{4}{\pi} \sum_{k=1}^{\infty} \frac{(-1)^{k-1}}{2k-1} \exp\left(-\frac{(2k-1)^2\pi^2}{8x^2}\right) \tag{14}$$

for any positive real number x.

The following theorem follows from the Leibniz bound on the tail of an alternating series for which the absolute values of the terms in the series decrease monotonically to zero (namely, the absolute value of the leading term of the tail is an upper bound on the absolute value of the tail; the bound in Theorem 6 would also be valid if the summation started from n rather than $n + 1$).

Theorem 6. *Suppose that* n *is a positive integer. Then, the tail of the series for* D *defined in (14) satisfies*

$$\left| \frac{4}{\pi} \sum_{k=n+1}^{\infty} \frac{(-1)^{k-1}}{2k-1} \exp\left(-\frac{(2k-1)^2\pi^2}{8x^2}\right) \right| < \frac{4}{\pi} \exp\left(-\frac{(2n-1)^2\pi^2}{8x^2}\right) \tag{15}$$

for any positive real number x. *If* ε *is a positive real number less than 1 and*

$$n \ge \frac{1}{2} + \frac{x\sqrt{2}}{\pi} \sqrt{\ln\left(\frac{4}{\pi\varepsilon}\right)}, \tag{16}$$

then the right-hand side of (15) is at most ε.

2.2 Calibration and deviation of a subpopulation from the full population

This subsection summarizes methods of [11] for assessing deviation of a subpopulation from the full population and for assessing the calibration of probabilistic predictions. The primary goal of this subsection is to introduce the Kolmogorov-Smirnov and Kuiper metrics, as well as factors suitable for normalizing them so as to facilitate evaluation of attained significance levels (P-values).

We consider n real numbers $S_1, S_2, \ldots, S_n$ known as "scores" (or sometimes as "predicted probabilities" when calibrating probabilistic predictions), each paired with a real-valued "response," $R_1, R_2, \ldots, R_n$, as well as a positive "weight," $W_1, W_2, \ldots, W_n$; we view the scores $S_1, S_2, \ldots, S_n$ and weights $W_1, W_2, \ldots, W_n$ as given, not random, while we view the responses $R_1, R_2, \ldots, R_n$ as random. We assume throughout

that all responses are stochastically independent (allowing dependence among the responses would be far beyond the scope of the present paper). Without loss of generality, we assume that $S_1 < S_2 < \cdots < S_n$ (perturbing the original scores slightly in order to ensure their uniqueness, if necessary). We consider also a given function r which returns the expected response averaged over the full population at any specified score s; that is, $r(s)$ is the expected value of the response for all members of the full population whose score is s. When assessing the calibration of probabilistic predictions, the score s is a predicted probability and the expected response $r(s)$ is supposed to match the prediction, s; hence, $r(s) = s$ when assessing calibration.

In order to gauge deviation of the observed responses R_1, R_2, ..., R_n from the given function r, we construct the sequence of cumulative differences

$$B_j = \frac{\sum_{k=1}^{j} (R_k - r(S_k))\, W_k}{\sum_{k=1}^{n} W_k} \tag{17}$$

for $j = 1, 2, \ldots, n$. We also construct the sequence of cumulative weights

$$A_j = \frac{\sum_{k=1}^{j} W_k}{\sum_{k=1}^{n} W_k} \tag{18}$$

for $j = 1, 2, \ldots, n$. Figures 1, 2, and 3 duplicate plots from [11] of B_1, B_2, ..., B_n versus A_1, A_2, ..., A_n for some numerical examples. A simple calculation of [11] shows that the expected slope of the line graph of B_1, B_2, ..., B_n versus A_1, A_2, ..., A_n from A_{j-1} to A_j is $\mathbb{E}[R_j] - r(S_j)$; that is, the expected slope is simply the deviation of the expected response from the full population's, in a graph for which A_1, A_2, ..., A_n are the abscissae (the horizontal coordinates) and B_1, B_2, ..., B_n are the ordinates (the vertical coordinates). Thus, a long range of significant deviation indicates significant average deviation over that range.

In particular, absence of significant deviation results in a flat graph that is nearly horizontal. Two metrics which measure deviations away from 0 (thus characterizing "goodness-of-fit") are the maximum absolute value

$$G = \max_{1 \le k \le n} |B_k| \tag{19}$$

and the range (the maximum value minus the minimum value)

$$H = \max_{0 \le k \le n} B_k - \min_{0 \le k \le n} B_k, \tag{20}$$

where $B_0 = 0$ (Remark 1 of [11] gives a compelling reason to include $B_0 = 0$, a reason analogous to why [7] introduced an analogous statistic decades earlier in a related context). The statistic G is due to Kolmogorov and Smirnov in the similar settings discussed by [6] and [9]; the statistic H is due to Kuiper in the similar setting discussed by [7].

Under the null hypothesis that the response at every score s is an independent Bernoulli variate taking the value 1 with probability $r(s)$ and the value 0 with probability $1 - r(s)$, calibrating attained significance levels (P-values) for these statistics involves normalization by the quantity

$$\sigma = \frac{\sqrt{\sum_{k=1}^{n} r(S_k) \cdot (1 - r(S_k)) \cdot (W_k)^2}}{\sum_{k=1}^{n} W_k}; \tag{21}$$

needless to say, such a null hypothesis can be appropriate only when each R_k is either 0 or 1, for each $k = 1, 2, \ldots, n$. More generally, under a null hypothesis for which the response at score s is expected to have a variance $v(s)$ around $r(s)$, the normalization would be by the quantity

$$\sigma = \frac{\sqrt{\sum_{k=1}^{n} v(S_k) \cdot (W_k)^2}}{\sum_{k=1}^{n} W_k}; \tag{22}$$

needless to say, $v(s) = r(s) \cdot (1 - r(s))$ for a Bernoulli variate taking the value 1 with probability $r(s)$ and the value 0 with probability $1 - r(s)$, consistent with (21). "Normalization" means considering the ratios G/σ and H/σ rather than the unnormalized G and H from (19) and (20).

In many practical applications, such a large sample of the full population is available that r and v can be estimated to high accuracy from the data; [11] elaborates some methods for such estimation.

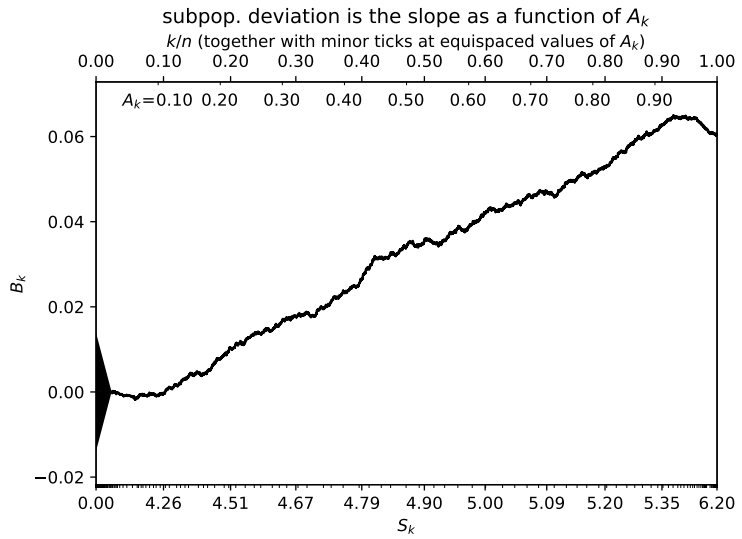

Figure 1: (duplicated from [11]) Difference in the number of people in a household between the county of Los Angeles and the entire state of California (the county is the subpopulation, while the state is the full population). The scores indicated along the lower horizontal axis are $\log_{10}$ of the adjusted household income, randomly perturbed by about one part in a hundred million to ensure their uniqueness, and there are $n = 35{,}364$ households representing Los Angeles in the (weighted) sample from the 2019 American Community Survey of the United States Census Bureau. Kuiper's statistic $H = 0.06674$, while $H/\sigma = 9.605$; Kolmogorov's and Smirnov's $G = 0.06495$, while $G/\sigma = 9.347$. P-values for both statistics are 0 to full double-precision accuracy. These P-values reflect the observed difference of many standard deviations beyond the expected means. Deviation of the subpopulation from the full population is the slope as graphed here.

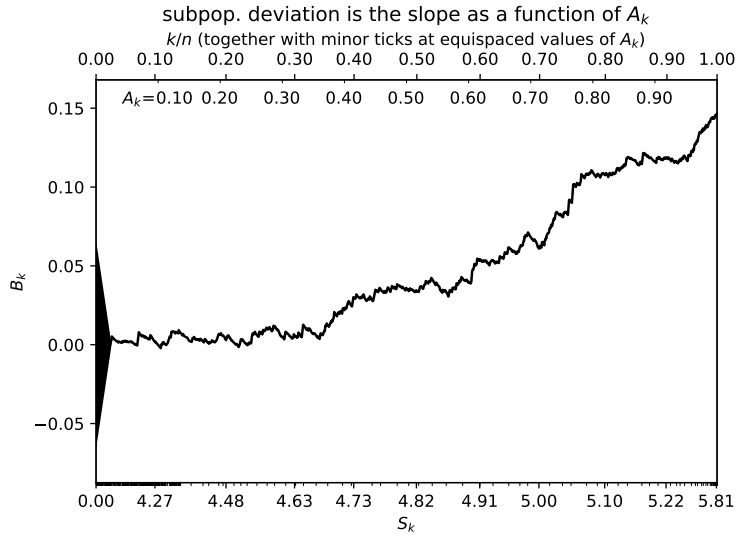

Figure 2: (duplicated from [11]) Difference in the number of children related to the head of household between the county of Stanislaus and the entire state of California (the county is the subpopulation, while the state is the full population). The scores indicated along the lower horizontal axis are $\log_{10}$ of the adjusted household income, randomly perturbed by about one part in a hundred million to guarantee their uniqueness, and there are $n = 1{,}624$ households representing Stanislaus in the (weighted) sample from the 2019 American Community Survey of the United States Census Bureau. Kuiper's statistic $H = 0.1489$, while $H/\sigma = 4.500$; Kolmogorov's and Smirnov's $G = 0.1467$, while $G/\sigma = 4.433$. The P-value for Kuiper's statistic is 0.00002713; the P-value for Kolmogorov's and Smirnov's is 0.00001859. These P-values reflect the observed difference of several standard deviations beyond the expected means. Deviation of the subpopulation's response (the number of children) from the full population's is the slope exactly as displayed in this plot.

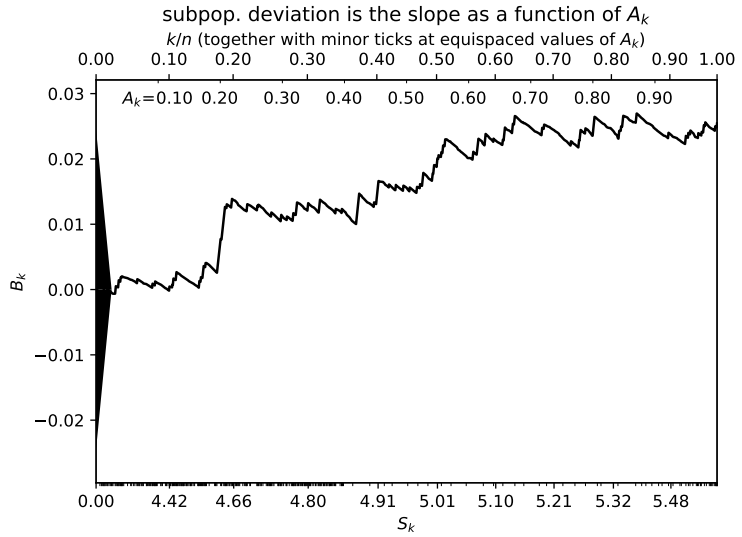

Figure 3: (duplicated from [11]) Difference in whether a household has internet access via satellite between the county of Napa and the entire state of California (the county is the subpopulation, while the state is the full population). The scores indicated along the lower horizontal axis are $\log_{10}$ of the adjusted household income, randomly perturbed by about one part in a hundred million to ensure their uniqueness, and there are $n = 679$ households representing Napa in the (weighted) sample from the 2019 American Community Survey of the United States Census Bureau. Kuiper's statistic $H = 0.02761$, while $H/\sigma = 2.259$; Kolmogorov's and Smirnov's $G = 0.02695$, while $G/\sigma = 2.205$. The P-value for Kuiper's statistic is 0.09559; the P-value for Kolmogorov's and Smirnov's is 0.05492. The P-values reflect the observed difference of not even a couple standard deviations beyond the expected means. Deviation of the subpopulation is the slope as displayed.

2.3 Calibration of P-values for the Kolmogorov-Smirnov and Kuiper statistics

This subsection derives Corollary 9, providing a method for the calculation of attained significance levels (P-values) for the Kolmogorov-Smirnov and Kuiper metrics introduced in the previous subsection.

Propositions 1–4 of [4] prove the following theorem. (Technically, [4] provides much stronger and more general results, characterizing not only convergence but also the convergence rates. See also closely related results of [10].) Earlier results of [3] motivated the work of [4] and [10] (among others), and are also closely related to the metrics of [12]. The proofs of [3] are in some ways simpler and easier to grasp, despite being restricted to a somewhat more special case, and are a superb starting point in addition to being of substantial independent importance, both practically and theoretically.

Theorem 7. *Assume the null hypothesis that the subpopulation has no expected deviation from the full population (that is, $\mathbb{E}[R_k] = r(S_k)$ for $k = 1, 2, \ldots, n$). Suppose also that $\max_{1 \le k \le n} v(S_k) \cdot (W_k)^2 / \sum_{j=1}^{n} v(S_j) \cdot (W_j)^2$ converges to 0 in the limit as n becomes large. Then, with G defined in (19) and σ defined in (22), every accumulation point (in the weak topology) of the normalized Kolmogorov-Smirnov statistic G/σ for measuring deviation of a subpopulation from the full population is distributed as the maximum over a subset of the unit interval $[0, 1]$ of the absolute value of the standard Brownian motion (if the scores $S_1, S_2, \ldots, S_n$ are all distinct for each n, then the subset is in fact always the same — the entire unit interval $[0, 1]$). This is stochastically dominated by the maximum of the absolute value of the standard Brownian motion over the whole unit interval $[0, 1]$. The same statements hold for the normalized Kolmogorov-Smirnov statistic G/σ for measuring calibration, taking the expected response at each score to be equal to the score, that is, $r(s) = s$ for every score s.*

The theorems of [4] similarly yield the analogous theorem for the Kuiper statistic:

Theorem 8. *Assume the null hypothesis that the subpopulation has no expected deviation from the full population (that is, $\mathbb{E}[R_k] = r(S_k)$ for $k = 1, 2, \ldots, n$). Suppose also that $\max_{1 \le k \le n} v(S_k) \cdot (W_k)^2 / \sum_{j=1}^{n} v(S_j) \cdot (W_j)^2$ converges to 0 in the limit as n becomes large. Then, with H defined in (20) and σ defined in (22),*

every accumulation point (in the weak topology) of the normalized Kuiper statistic H/σ for measuring deviation of a subpopulation from the full population is distributed as the range over a subset of the unit interval $[0,1]$ of the standard Brownian motion (if the scores S_1, S_2, ..., S_n are all distinct for each n, then the subset is in fact always the same — the entire unit interval $[0,1]$). This is stochastically dominated by the range of the standard Brownian motion over the whole unit interval $[0,1]$. (The range is the maximum minus the minimum.) The same statements hold for the normalized Kuiper statistic H/σ for measuring calibration, taking the expected response at each score to be equal to the score, that is, $r(s) = s$ for every score s.

Putting everything together yields the following.

Corollary 9. *Evaluating 1 minus the function D from (14) applied to the normalized Kolmogorov-Smirnov statistic G/σ yields rigorously conservative P-values (due to Theorems 5 and 7) — this is $1 - D(G/\sigma)$. Taking 1 minus the function F from (1) applied to the normalized Kuiper statistic H/σ yields rigorously conservative P-values (due to Theorems 3 and 8) — this is $1 - F(H/\sigma)$. The Kolmogorov-Smirnov metric G is defined in (19), the Kuiper metric H is defined in (20), and the normalizing factor σ is defined in (22), with (22) reducing to (21) in the case that the responses are Bernoulli variates. This calibration of P-values is perfect (no excess conservatism in the limit of large n) if the subsets from Theorems 7 and 8 are dense in the unit interval $[0,1]$.*

3 Results

The present section illustrates (via examples and plots) the numerical methods of the preceding section. The appendix lists the code for an implementation in Python 3; running the code as a script automatically reproduces all results and figures reported in the present section.[2]

Figures 4 and 5 plot $1 - F(x)$ versus x and $1 - D(x)$ versus x, respectively, where F is defined in (1) and D is defined in (14). The calculation for F truncates the series in (1) after n terms, where $n = n(x)$ is the least integer such that (9) guarantees full double-precision accuracy (with $\varepsilon \approx 2.2\text{E}{-16}$). Similarly, the calculation for D truncates the series in (14) after n terms, where $n = n(x)$ is the least integer such that (16) guarantees full double-precision accuracy (again with $\varepsilon \approx 2.2\text{E}{-16}$). To give an indication of how another sub-Gaussian distribution decays, Figure 6 plots $1 - \Phi(x)$ versus x, where Φ is the cumulative distribution function for the standard normal distribution.

Formula 1.4 of [5] and Formula 46 of [8] give the means of the distributions associated with the cumulative distribution functions F and D defined in (1) and (14), as $2\sqrt{2/\pi} \approx 1.5958$ and $\sqrt{\pi/2} \approx 1.2533$, respectively. The horizontal positions of the vertical dotted lines labeled "mean" in Figures 4 and 5 are at these mean values. A unit test of the implementations of the cumulative distribution functions is to numerically evaluate the means. Using a Gauss-Chebyshev quadrature of order 100,000 to integrate $1 - F(x)$ and $1 - D(x)$ from $x = 1\text{E}{-8}$ to $x = 8$ yields the correct means to better than 8-digit relative accuracy in the implemented codes, thus passing this unit test.

Figures 7 and 8 plot the calibration curves for the Kuiper and Kolmogorov-Smirnov statistics, respectively. The calibration curves are the empirical cumulative distribution functions of the P-values for calibration calculated for 100,000 data sets generated by drawing independent Bernoulli responses at the scores, with the probability of success in the Bernoulli distribution being exactly equal to the score (so that the data is perfectly calibrated, by construction). Perfectly calibrated P-values would follow the uniform distribution over the unit interval $[0,1]$ under the null hypothesis, and so ideally the plotted empirical cumulative distribution functions should approach the cumulative distribution function for the uniform distribution as the sample size increases. The cumulative distribution function for the uniform distribution over the unit interval $[0,1]$ is the line connecting the origin $(0,0)$ to the point $(1,1)$; each plot displays a dashed line to indicate the ideal calibration curve. The other curves are the empirical cumulative distribution functions of the P-values for data sets with sample sizes $n = 100$, 1,000, 10,000; as expected, the curve closest to the diagonal dashed line in each plot is that for $n = 10{,}000$, the next closest is for $n = 1{,}000$, and the farthest is for $n = 100$. The weights in these synthetically generated data sets are uniform (all equal), just for simplicity.

Figures 7 and 8 illustrate Corollary 9 — notice that the empirical curves all lie entirely below the diagonal dashed line, in accordance with the calculated P-values being conservatively calibrated (the P-values are not

[2]The software is also available for download at `http://tygert.com/dists.py`.

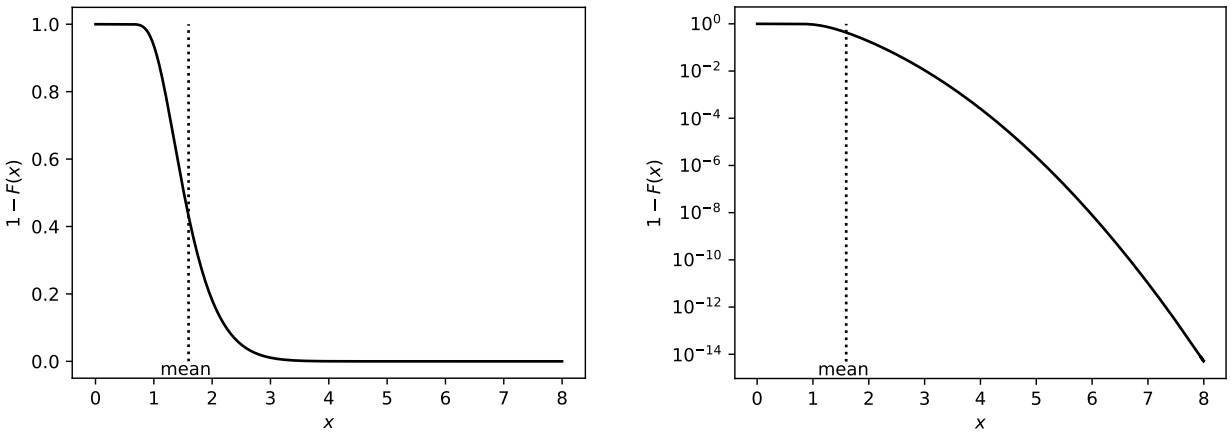

Figure 4: Both plots graph $1 - F(x)$ versus x, where F is defined in (1) and is central to Corollary 9. The plot on the right uses a logarithmic scale for the vertical axis, unlike the plot on the left. The vertical dotted line indicates the value of x corresponding to the mean of the distribution for which F is the cumulative distribution function.

smaller on average than expected). The figures also illustrate the convergence to ideal calibration that [3], [4], and [10] prove as the scores become dense in the unit interval $[0, 1]$ (the scores are quite dense already with $n = 10{,}000$, for example).

The ends of the captions of Figures 1, 2, and 3 report P-values evaluated using Corollary 9. Attained significance levels (P-values) for all methods of [11] can also be calibrated and calculated directly using Corollary 9, under the assumption that, for each of the scores from the subpopulation, the full population contains many members whose scores are closer to the score from the subpopulation than to other scores from the subpopulation; if this assumption is invalid, then the statistics fed into the cumulative distribution functions require adjustment to account for the additional stochasticity (as described by [11] and [12]).

4 Discussion and conclusion

As shown above, the combination of [5], [2], [3], [4], [10], and others trivially yields computationally efficient and convenient calibration of P-values for the metrics of [11], metrics very similar to those of [6] and [9] and of [7] (whose work directly stimulated all the others', including that of the author of the present paper). The results of [3], [4], and [10] reduce the problem of calibration to the calculation of the distributions of the range and of the maximum absolute value of the standard Brownian motion over the unit interval $[0, 1]$; the results of [5] and [2] completely characterize those distributions. Simple, straightforward manipulation of the resulting formulae then yields the cumulative distribution functions required for calibrating P-values, as detailed in Section 2 above. Implementation is easy and Section 3 demonstrates the corresponding Python module and unit tests listed in the appendix; Section 3 validates the numerical methods and implementation via plots of the cumulative distribution functions of the metrics and of the associated P-values, as well as via checks against analytic, closed-form expressions, illustrating use of the codes both on their own and as applied to both real and synthetic data sets. The software of the appendix is ready for widespread use under the permissive MIT copyright license.

Acknowledgements

We would like to thank Kamalika Chaudhuri, Imanol Arrieta Ibarra, Mike Rabbat, Jonathan Tannen, and Susan Zhang.

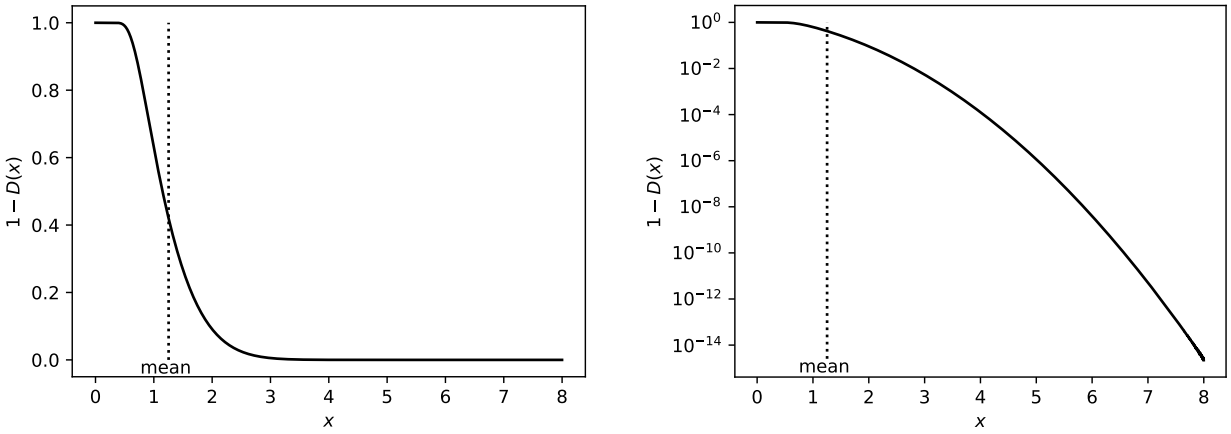

Figure 5: Both plots graph $1 - D(x)$ versus x, where D is defined in (14) and is central to Corollary 9. The plot on the right uses a logarithmic scale for the vertical axis, unlike the plot on the left. The vertical dotted line indicates the value of x corresponding to the mean of the distribution for which D is the cumulative distribution function.

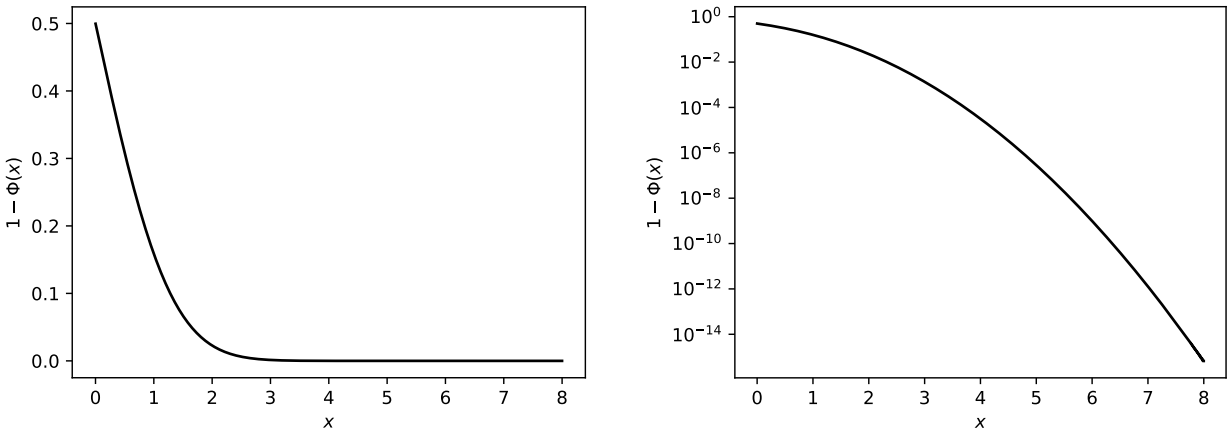

Figure 6: Both plots graph $1-\Phi(x)$ versus x, where Φ is the cumulative distribution function for the standard normal distribution; $\Phi(x) = \int_{-\infty}^{x} \exp(-y^2/2)\,dy\,/\,\sqrt{2\pi}$. The plot on the right uses a logarithmic scale for the vertical axis, unlike the plot on the left.

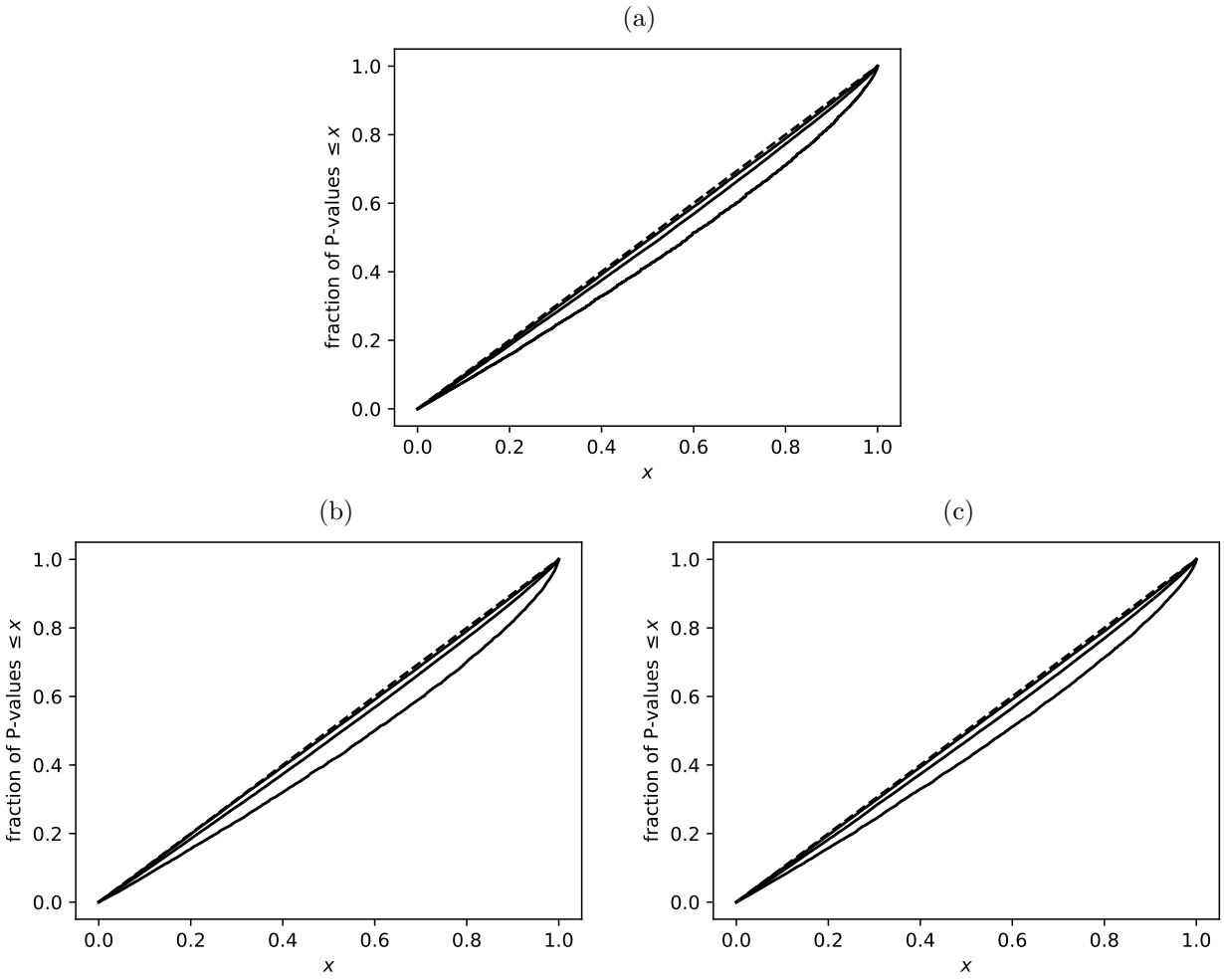

Figure 7: Calibration curves (empirical cumulative distribution functions under the null hypothesis of perfectly calibrated data) of the Kuiper P-value for calibration for sample sizes $n = 100$, 1,000, 10,000; in each plot, the dashed line connects the origin $(0, 0)$ to the point $(1, 1)$ and illustrates perfect calibration, while the curve for $n = 10{,}000$ is closest to perfect, $n = 1{,}000$ is next closest, and $n = 100$ is the farthest. Subfigure (a) uses scores equispaced on the unit interval $[0, 1]$, (b) squares each of the initially equispaced scores, and (c) takes the square root of each of the initially equispaced scores. The score s is the predicted probability, with the expected response $r(s) = s$ to assess calibration. Each empirical cumulative distribution function plotted arises from 100,000 data sets generated independently while assuming the null hypothesis.

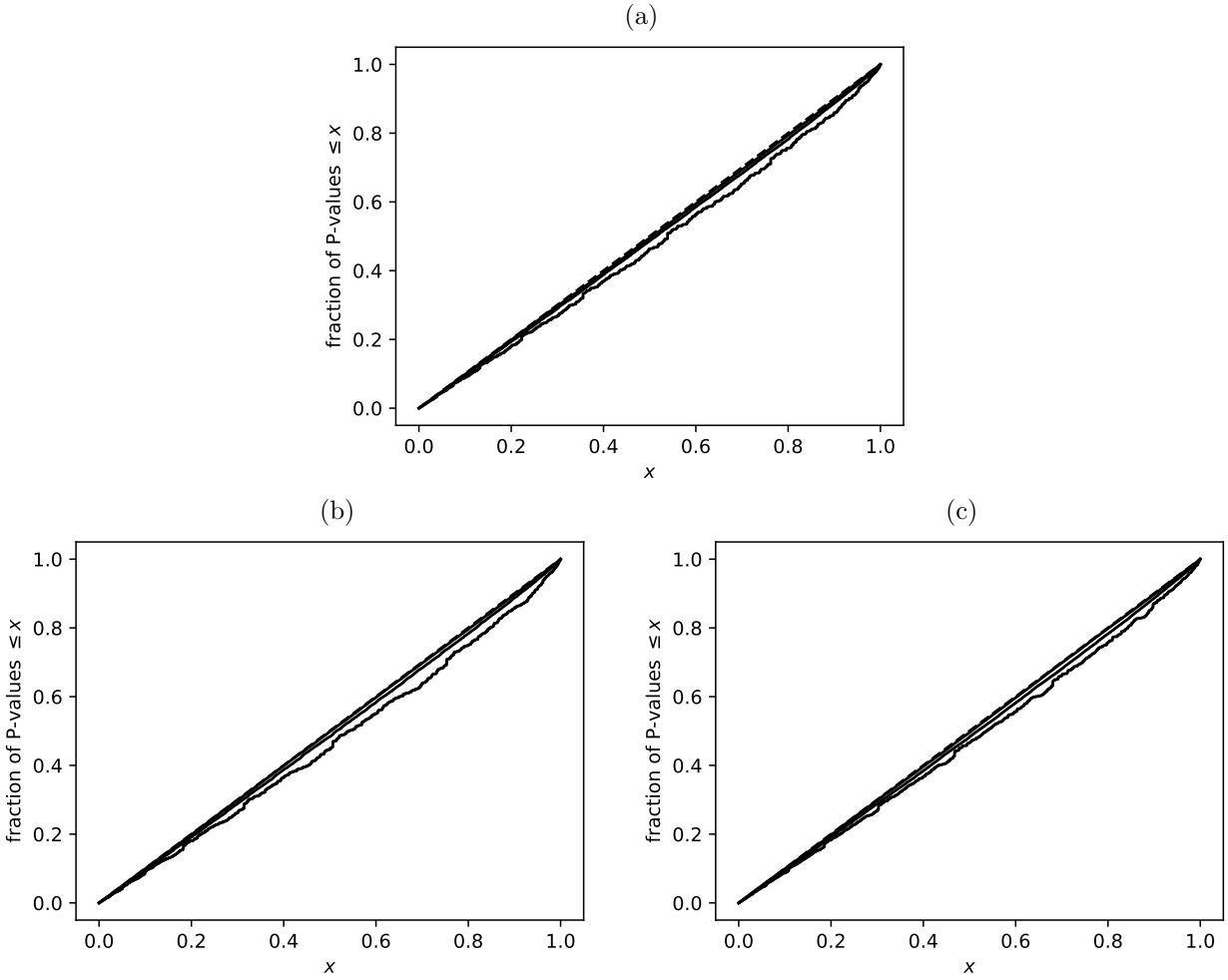

Figure 8: Calibration curves (empirical cumulative distribution functions under the null hypothesis of perfectly calibrated data) of the Kolmogorov-Smirnov P-value for calibration for sample sizes $n = 100$, 1,000, 10,000; in each plot, the dashed line connects the origin $(0, 0)$ to the point $(1, 1)$ and illustrates perfect calibration, while the curve for $n = 10{,}000$ is closest to perfect, $n = 1{,}000$ is next closest, and $n = 100$ is the farthest. Subfigure (a) uses scores equispaced on the unit interval $[0, 1]$, (b) squares each of the initially equispaced scores, and (c) takes the square root of each of the initially equispaced scores. The score s is the predicted probability, with the expected response $r(s) = s$ to assess calibration. Each empirical cumulative distribution function plotted arises from 100,000 data sets generated independently while assuming the null hypothesis.

A Python implementation

This appendix lists an implementation in Python 3 (the software can be used as a freestanding module or run as a script to exercise the unit tests and reproduce the figures of Section 3 above):[3]

```
#!/usr/bin/env python3

"""
Copyright (c) Meta Platforms, Inc. and affiliates.

Calculate cumulative distribution functions for standard Brownian motions.

Running as a script tests assertions that closed-form, analytical expressions
for the means match numerical evaluations of the means for the cumulative
distribution functions, prints values of the cumulative distribution functions
at some interesting values for their arguments, saves to disk plots in pdf
of the complementary cumulative distribution functions, and saves to disk plots
in both pdf and jpg of calibration curves for synthetic data sets drawn from
perfectly calibrated distributions. The script saves plots in the current
directory, in files named "gauss.pdf", "gauss_log.pdf", "kuiper.pdf",
"kuiper_log.pdf", "kolmogorov_smirnov.pdf", and "kolmogorov_smirnov_log.pdf".
The files whose names end with "_log.pdf" use log scales for the vertical axes.
The plots for the metrics of Kuiper and of Kolmogorov and Smirnov include
vertical dotted lines at the means associated with the corresponding
distribution. The script saves twelve other plots in the current directory,
too, as detailed in the docstring for function plotnull below.

An article detailing the functions named after mathematicians and statisticians
(Kolmogorov, Smirnov, Kuiper, Gauss, and Chebyshev) is Mark Tygert's
"Calibration of P-values for calibration and for deviation of a subpopulation
from the full population."

Functions
---------
kolmogorov_smirnov
    Evaluates the cumulative distribution function for the maximum
    of the absolute value of the standard Brownian motion on [0, 1]
kuiper
    Evaluates the cumulative distribution function for the range
    (maximum minus minimum) of the standard Brownian motion on [0, 1]
gauss
    Evaluates the cumulative distribution function for the distribution N(0, 1)
    (the standard normal distribution, involving a Gaussian)
chebyshev
    Integrates the function f(x) from x=a to x=b using n Chebyshev nodes
testmeans
    Verifies that the means of the cumulative distribution functions are right
printvals
    Evaluates the cumulative distribution functions at some points of interest
    and prints them
saveplots
    Plots and saves to disk the complementary cumulative distribution functions
plotnull
    Plots the P-values for data generated from a perfectly calibrated model

This source code is licensed under the MIT license found in the LICENSE file in
the root directory of this source tree.
"""
```

[3]The code is also available for download at `http://tygert.com/dists.py`.

```
54
55
56  import math
57  import numpy as np
58  from numpy.random import default_rng
59  import subprocess
60  import matplotlib
61  matplotlib.use('agg')
62  import matplotlib.pyplot as plt
63
64
65  def kolmogorov_smirnov(x):
66      """
67      Evaluates the cumulative distribution function for the maximum
68      of the absolute value of the standard Brownian motion on [0, 1]
69
70      Parameters
71      ----------
72      x : float
73          argument at which to evaluate the cumulative distribution function
74          (must be positive)
75
76      Returns
77      -------
78      float
79          cumulative distribution function evaluated at x
80      """
81      assert x > 0
82      # Compute the machine precision assuming binary numerical representations.
83      eps = 7 / 3 - 4 / 3 - 1
84      # Determine how many terms to use to attain accuracy eps.
85      fact = 4 / math.pi
86      kmax = math.ceil(
87          1 / 2 + x * math.sqrt(2) / math.pi * math.sqrt(math.log(fact / eps)))
88      # Sum the series.
89      c = 0
90      for k in range(kmax):
91          kplus = k + 1 / 2
92          c += (-1)**k / kplus * math.exp(-kplus**2 * math.pi**2 / (2 * x**2))
93      c *= 2 / math.pi
94      return c
95
96
97  def kuiper(x):
98      """
99      Evaluates the cumulative distribution function for the range
100     (maximum minus minimum) of the standard Brownian motion on [0, 1]
101
102     Parameters
103     ----------
104     x : float
105         argument at which to evaluate the cumulative distribution function
106         (must be positive)
107
108     Returns
109     -------
110     float
111         cumulative distribution function evaluated at x
112     """
```

```
    assert x > 0
    # Compute the machine precision assuming binary numerical representations.
    eps = 7 / 3 - 4 / 3 - 1
    # Determine how many terms to use to attain accuracy eps.
    fact = 4 / math.sqrt(2 * math.pi) * (1 / x + x / math.pi**2)
    kmax = math.ceil(
        1 / 2 + x / math.pi / math.sqrt(2) * math.sqrt(math.log(fact / eps)))
    # Sum the series.
    c = 0
    for k in range(kmax):
        kplus = k + 1 / 2
        c += (8 / x**2 + 2 / kplus**2 / math.pi**2) * math.exp(
            -2 * kplus**2 * math.pi**2 / x**2)
    return c


def gauss(x):
    """
    Evaluates the cumulative distribution function for the distribution N(0, 1)
    (the standard normal distribution, involving a Gaussian)

    Parameters
    ----------
    x : float
        argument at which to evaluate the cumulative distribution function

    Returns
    -------
    float
        cumulative distribution function evaluated at x
    """
    return (1 + math.erf(x / math.sqrt(2))) / 2


def chebyshev(a, b, n, f):
    """
    Integrates the function f(x) from x=a to x=b using n Chebyshev nodes

    Parameters
    ----------
    a : float
        lower limit of integration
    b : float
        upper limit of integration
    n : int
        number of Chebyshev nodes in the Gauss-Chebyshev quadrature
    f : callable
        real-valued function of a real argument to be integrated

    Returns
    -------
    float
        integral from x=a to x=b of f(x) (dx)
    """
    sum = 0
    for k in range(n):
        c = math.cos((2 * k + 1) * math.pi / (2 * n))
        x = a + (b - a) * (1 + c) / 2
        sum += f(x) * math.sqrt(1 - c**2)
```

```
    sum *= (b - a) * math.pi / (2 * n)
    return sum


def testmeans():
    """
    Verifies that the means of the cumulative distribution functions are right

    Returns
    -------
    float
        mean of the Kolmogorov-Smirnov statistic under the null hypothesis
        that the subpopulation arises from the full population's distribution
        (and that the scores are dense in their domain)
    float
        mean of the Kuiper statistic under the null hypothesis
        that the subpopulation arises from the full population's distribution
        (and that the scores are dense in their domain)

    References
    ----------
    William Feller, "The asymptotic distribution of the range of sums of
        independent random variables," Ann. Math. Statist., 22 (1951): 427-432.
    Jaume Masoliver, "Extreme values and the level-crossing problem: an
        application to the Feller process," Phys. Rev. E., 89 (2014): 042106.
    """
    # Compute the means of the Kolmogorov-Smirnov and Kuiper statistics
    # using closed-form analytic expressions (see Formula 1.4 of the reference
    # to Feller given in the docstring, as well as Formula 46 of the reference
    # to Masoliver).
    ks_mean = math.sqrt(math.pi / 2)
    ku_mean = 2 * math.sqrt(2 / math.pi)
    # Compute the means from the associated cumulative distribution functions
    # evaluated numerically.
    ks_mean2 = chebyshev(1e-8, 8, 100000, lambda x: 1 - kolmogorov_smirnov(x))
    ku_mean2 = chebyshev(1e-8, 8, 100000, lambda x: 1 - kuiper(x))
    # Check that the calculated values agree with each other.
    tolerance = 1e-8
    assert (ks_mean - ks_mean2) / ks_mean < tolerance
    assert (ku_mean - ku_mean2) / ku_mean < tolerance
    return ks_mean, ku_mean


def printvals(ks_mean, ku_mean):
    """
    Evaluates the cumulative distribution functions at some points of interest
    and prints them

    Parameters
    ----------
    ks_mean : float
        mean of the Kolmogorov-Smirnov statistic under the null hypothesis
        that the subpopulation arises from the full population's distribution
        (and that the scores are dense in their domain)
    ku_mean : float
        mean of the Kuiper statistic under the null hypothesis
        that the subpopulation arises from the full population's distribution
        (and that the scores are dense in their domain)
    """
```

```
    print(f'1 - kolmogorov_smirnov(0.001) = {1 - kolmogorov_smirnov(0.001)}')
    print('1 - kolmogorov_smirnov(ks_mean) = {}'
          .format(1 - kolmogorov_smirnov(ks_mean)))
    print('1 - kolmogorov_smirnov(9.347180056695407) = {}'
          .format(1 - kolmogorov_smirnov(9.347180056695407)))
    print('1 - kolmogorov_smirnov(4.433008036126233) = {}'
          .format(1 - kolmogorov_smirnov(4.433008036126233)))
    print('1 - kolmogorov_smirnov(2.2049236860640984) = {}'
          .format(1 - kolmogorov_smirnov(2.2049236860640984)))
    print(f'1 - kolmogorov_smirnov(1000) = {1 - kolmogorov_smirnov(1000)}')

    print()

    print(f'1 - kuiper(0.001) = {1 - kuiper(0.001)}')
    print(f'1 - kuiper(ku_mean) = {1 - kuiper(ku_mean)}')
    print(f'1 - kuiper(9.604586718869454) = {1 - kuiper(9.604586718869454)}')
    print(f'1 - kuiper(4.500374236241608) = {1 - kuiper(4.500374236241608)}')
    print(f'1 - kuiper(2.2585549672545224) = {1 - kuiper(2.2585549672545224)}')
    print(f'1 - kuiper(1000) = {1 - kuiper(1000)}')

    print()

    print('switch the mean values and see that the P-values deviate '
          + 'far from 0.5:')
    print('1 - kolmogorov_smirnov(ku_mean) = {}'
          .format(1 - kolmogorov_smirnov(ku_mean)))
    print(f'1 - kuiper(ks_mean) = {1 - kuiper(ks_mean)}')


def saveplots(ks_mean, ku_mean):
    """
    Plots and saves to disk the complementary cumulative distribution functions

    The plots, saved in the current directory, are "gauss.pdf",
    "gauss_log.pdf", "kuiper.pdf", "kuiper_log.pdf", "kolmogorov_smirnov.pdf",
    and "kolmogorov_smirnov_log.pdf". The files whose names end with "_log.pdf"
    use logarithmic scales for the vertical axes. The plots for Kuiper
    and for Kolmogorov and Smirnov include vertical dotted lines at the means
    of the corresponding distribution, assuming that the input parameters
    are correct.

    Parameters
    ----------
    ks_mean : float
        mean of the Kolmogorov-Smirnov statistic under the null hypothesis
        that the subpopulation arises from the full population's distribution
        (and that the scores are dense in their domain)
    ku_mean : float
        mean of the Kuiper statistic under the null hypothesis
        that the subpopulation arises from the full population's distribution
        (and that the scores are dense in their domain)
    """
    for func in ['gauss', 'kuiper', 'kolmogorov_smirnov']:
        for logscale in [True, False]:
            # Create a plot.
            plt.figure(figsize=[4.8, 3.6])
            # Create abscissae and ordinates.
            xmax = 8
            x = np.arange(1e-3, xmax, 1e-3)
```

```
            y = 1 - np.vectorize(globals()[func])(x)
            # Plot y versus x.
            plt.plot(x, y, 'k')
            # Plot a vertical line at the mean.
            if func == 'kuiper':
                mean = ku_mean
            elif func == 'kolmogorov_smirnov':
                mean = ks_mean
            else:
                mean = 0
            if mean > 0:
                plt.vlines(mean, 1 - globals()[func](xmax), 1, 'k', 'dotted')
                plt.text(
                    mean, 1 - globals()[func](xmax), 'mean ',
                    ha='center', va='top')
            # Set the vertical axis to use a logscale if logscale is True.
            if logscale:
                plt.yscale('log')
            # Title the axes.
            plt.xlabel('$x$')
            if func == 'kuiper':
                plt.ylabel('$1 - F(x)$')
            elif func == 'kolmogorov_smirnov':
                plt.ylabel('$1 - D(x)$')
            else:
                plt.ylabel('$1 - \\Phi(x)$')
            # Clean up the whitespace in the plot.
            plt.tight_layout()
            # Save the plot.
            filename = func
            if logscale:
                filename += '_log'
            filename += '.pdf'
            plt.savefig(filename, bbox_inches='tight')
            plt.close()


def plotnull(ns, points, transform=None, suffix=''):
    """
    Plots the P-values for data generated from a perfectly calibrated model

    The plots, saved in the current directory are "kuiper_ecdf[suffix].pdf",
    "kuiper_ecdf[suffix].jpg", "kolmogorov_smirnov_ecdf[suffix].pdf", and
    "kolmogorov_smirnov_ecdf[suffix].jpg". The JPEG versions are conversions
    from the PDF at a resolution of 1200 pixels per inch. The plots display
    the empirical cumulative distribution functions of the P-values associated
    with the Kuiper and Kolmogorov-Smirnov statistics, for points data sets,
    each with the number of scores and corresponding Bernoulli responses
    given by the given entry in ns (running everything again for each entry
    in ns). The Bernoulli responses are independent, and the probability
    of success for each is equal to the corresponding score (ensuring perfect
    calibration of the underlying data distribution). The transform gets
    applied to each score, with the scores being equispaced before application
    of transform (and remaining equispaced if transform is None).

    Parameters
    ----------
    ns : list of int
        sample sizes for each generated data set
```

```
    points : int
        number of data sets to generate per calibration curve (that is,
        per empirical cumulative distribution function of P-values)
    transform : callable, optional
        numpy function to apply to the otherwise equispaced scores
        (set to None -- the default -- to use the original equispaced scores)
    suffix : string, optional
        suffix to append to the filename (defaults to the empty string)
    """
    # Store processes for converting from pdf to jpeg in procs.
    procs = []
    # Store the calibration curves for both Kolmogorov-Smirnov and Kuiper
    # statistics (these are empirical cumulative distribution functions),
    # in ksc and kuc, respectively.
    ksc = np.zeros((len(ns), points))
    kuc = np.zeros((len(ns), points))
    for j, n in enumerate(ns):
        rng = default_rng(seed=543216789)
        # Run simulations points times.
        pks = np.zeros((points))
        pku = np.zeros((points))
        for k in range(points):
            # Generate predicted probabilities (the "scores").
            s = np.arange(0, 1, 1 / n)[:n]
            if transform is not None:
                s = transform(s)
            # Generate a sample of classifications (the "responses")
            # into two classes, correct (class 1) and incorrect (class 0),
            # avoiding numpy's random number generators that are based
            # on random bits -- they yield strange results for many seeds.
            uniform = rng.uniform(size=(n))
            r = (uniform <= s).astype(float)
            # Calculate the cumulative differences.
            c = (np.cumsum(r) - np.cumsum(s)) / n
            # Calculate the estimate of sigma.
            sigma = np.sqrt(np.sum(s * (1 - s))) / n
            # Compute the normalized Kolmogorov-Smirnov and Kuiper statistics.
            ks = np.abs(c).max() / sigma
            c = np.insert(c, 0, [0])
            ku = (c.max() - c.min()) / sigma
            # Evaluate the P-values.
            pks[k] = 1 - kolmogorov_smirnov(ks)
            pku[k] = 1 - kuiper(ku)
        # Calculate the empirical cumulative distributions of the P-values.
        ksc[j, :] = np.sort(pks)
        kuc[j, :] = np.sort(pku)
    for stat in ['kolmogorov_smirnov', 'kuiper']:
        # Create a plot.
        plt.figure(figsize=[4.8, 3.6])
        # Title the axes.
        plt.xlabel('$x$')
        plt.ylabel('fraction of P-values $\\leq x$')
        # Plot the empirical cumulative distribution functions.
        frac = np.arange(1 / points, 1 + 1 / points, 1 / points)[:points]
        for j in range(len(ns)):
            if stat == 'kolmogorov_smirnov':
                plt.plot(ksc[j, :], frac, color='k')
            elif stat == 'kuiper':
                plt.plot(kuc[j, :], frac, color='k')
```

```
        # Add a diagonal line from (0, 0) to (1, 1).
        zeroone = np.asarray((0, 1))
        plt.plot(zeroone, zeroone, 'k', linestyle='dashed')
        # Save the plot.
        filepdf = stat + '_ecdf' + suffix + '.pdf'
        plt.savefig(filepdf, bbox_inches='tight')
        plt.close()
        # Convert the pdf to jpg.
        filejpg = filepdf[:-4] + '.jpg'
        args = ['convert', '-density', '1200', filepdf, filejpg]
        procs.append(subprocess.Popen(args))
    print('waiting for conversion from pdf to jpg to finish....')
    for iproc, proc in enumerate(procs):
        proc.wait()
        print(f'{iproc + 1} of {len(procs)} conversions are done....')


if __name__ == '__main__':
    # Test if the cumulative distribution functions yield the known values
    # for their means.
    print('testing means...')
    ks_mean, ku_mean = testmeans()
    # Print values of the cumulative distribution functions at some interesting
    # values for their arguments.
    print()
    print('evaluating for particular values of the arguments...')
    print()
    printvals(ks_mean, ku_mean)
    # Save plots of the complementary cumulative distribution functions.
    print()
    print('plotting the complementary cumulative distribution functions...')
    saveplots(ks_mean, ku_mean)
    # Plot the calibration curves ("calibration curves" are the empirical
    # cumulative distribution functions of P-values under the null hypothesis
    # of perfect calibration).
    ns = [100, 1000, 10000]
    points = 100000
    print('plotting calibration with equispaced scores...')
    plotnull(ns, points)
    print('plotting calibration with square-rooted scores...')
    plotnull(ns, points, np.sqrt, suffix='_sqrt')
    print('plotting calibration with squared scores...')
    plotnull(ns, points, np.square, suffix='_square')
```

References

[1] Z. Ciesielski and S. J. Taylor. First passage times and sojourn times for Brownian motion in space and the exact Hausdorff measure of the sample path. *Trans. Amer. Math. Soc.*, 103(3):434–450, 1962.

[2] D. A. Darling and A. J. F. Siegert. The first passage problem for a continuous Markov process. *Ann. Math. Statist.*, 24(4):624–639, 1953.

[3] M. A. Delgado. Testing the equality of nonparametric regression curves. *Statist. Probab. Lett.*, 17(3):199–204, 1993.

[4] J. Diebolt. A nonparametric test for the regression function: asymptotic theory. *J. Statist. Plann. Inference*, 44(1):1–17, 1995.

[5] W. Feller. The asymptotic distribution of the range of sums of independent random variables. *Ann. Math. Statist.*, 22(3):427–432, 1951.

[6] A. N. Kolmogorov. Sulla determinazione empirica di una legge di distribuzione (On the empirical determination of a distribution function). *Giorn. Ist. Ital. Attuar.*, 4:83–91, 1933.

[7] N. H. Kuiper. Tests concerning random points on a circle. *Proc. Koninklijke Nederlandse Akademie van Wetenschappen Series A*, 63:38–47, 1962.

[8] J. Masoliver. Extreme values and the level-crossing problem: an application to the Feller process. *Phys. Rev. E*, 89(4), 2014. No. 042106.

[9] N. Smirnov. On the estimation of the discrepancy between empirical curves of distribution for two independent samples. *Bulletin Mathématique de l'Université de Moscou*, 2(2):3–11, 1939.

[10] W. Stute. Nonparametric model checks for regression. *Ann. Statist.*, 25(2):613–641, 1997.

[11] M. Tygert. Cumulative deviation of a subpopulation from the full population. *J. Big Data*, 8(117):1–60, 2021.

[12] M. Tygert. A graphical method of cumulative differences between two subpopulations. *J. Big Data*, 8(158):1–29, 2021.