Lecture 3: Balls in Bins

You have m balls, and want to randomly distribute them among n different bins. Why? That's a pretty good question: basically, and you'll have to believe me for now, this rather strange scenario (and its many variants) capture a lot of actual interesting or well-motivated problems.

We'll get back to those.

The things we might care about are (1) the *maximum load* of the bins, that is, what's the maximum number of balls any given bin contains once we've distributed them; (2) the *coverage*, that is, how many bins are non-empty; and (3) the *collisions*, that is, how many pairs of distinct balls share the same bin.

The simplest thing we can do is throwing our m into the n independently and uniformly at random. Let's see how that goes.

Collisions

One of the most basic things we can ask is whether the m balls will all fall into their own personal bin, that is, if there's going to be at least one bin containing more than one ball. *What's the probability to get at least one collision?*

Interlude: run the Birthday Paradox experiment in the classroom. Discuss assumptions (uniformity), etc.

Theorem 15 (Birthday Paradox)**.** *If you gather 23 people in a room, then with probability 50% there will be two sharing a birthday.*

To prove that, we'll tackle the more general question, for arbitrary m and n, of finding what the probability $p_{m,n}$ of having at least one collision is: the birthday paradox is for $n = 366$, because, of course, 2024 is a leap year, and asks to check that $p_{23,366} \geq 1/2$. Now, the result has to depend on the relation between m and n: if $m \geq n+1$, then that probability is exactly one, while if $n \gg m$ this should be less likely.

That might change in 2025...

Do you see why? Prove it (Pigeonhole).

Theorem 16. *The probability $p_{m,n}$ to get at least one collision is equal to*

$$p_{m,n} = 1 - \frac{n!}{n^m(n-m)!} = 1 - \frac{m!}{n^m}\binom{n}{m} \tag{19}$$

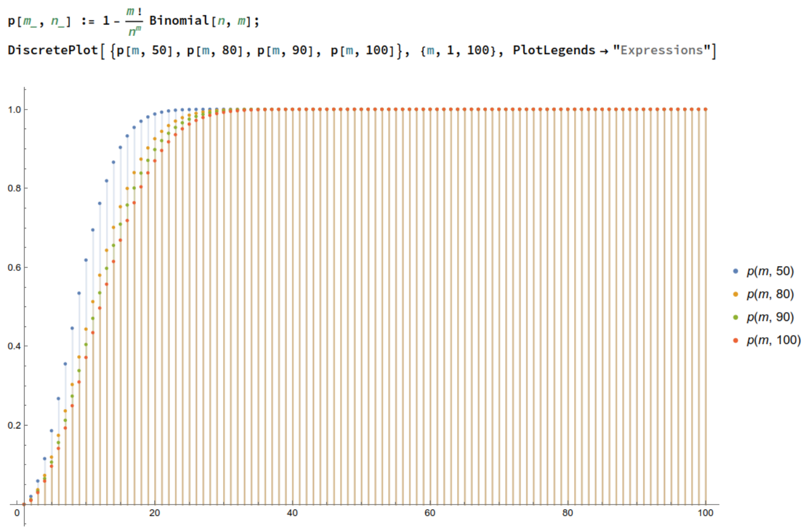
In particular, for $m = 23$ and $n = 366$, this is...?

Check it: $p_{22,366} \approx 0.475$, while $p_{23,366} \approx 0.506$.

Proof.

$$\begin{aligned}1 - p_{m,n} &= \frac{n}{n} \cdot \frac{n-1}{n} \cdot \frac{n-2}{n} \cdots \frac{n-m+1}{n} = \frac{1}{n^m} \prod_{\ell=0}^{m-1} (n-\ell) \\ &= \frac{n!}{n^m (n-m)!}\end{aligned}$$

□



```
p[m_, n_] := 1 - m!/n^m Binomial[n, m];
DiscretePlot[ {p[m, 50], p[m, 80], p[m, 90], p[m, 100]}, {m, 1, 100}, PlotLegends → "Expressions"]
```

Figure 4: The quantity $p_{m,n}$ from Eq. (19), plotted here as a function of m for various choices of n.

Looking at the graph above, it looks like we approach a very high probability of getting a collision *way* before $m = \Theta(n)$. Any guess at what m should be to, say, have probability at least 50% of a collision? Should it be

- $\Theta(\log n)$?
- $\Theta(\sqrt{n})$?
- $\Theta\left(\frac{n}{\log n}\right)$?
- Something else?

And *why*?

The worst approach: rabbit-out-of-a-hat, no intuition given. Take $m = c \cdot \sqrt{n}$ for some fixed constant $c > 0$. Plugging this in the expression

$p_{m,n}$ obtained in Theorem 16, we get

$$\begin{aligned}
1 - p_{m,n} = \frac{n!}{n^m (n-m)!} &\underset{n\to\infty}{\sim} \frac{1}{n^m} \cdot \frac{\sqrt{2\pi n}\left(\frac{n}{e}\right)^n}{\sqrt{2\pi(n-m)}\left(\frac{n-m}{e}\right)^{n-m}} \quad \text{(Stirling)} \\
&= \frac{1}{\sqrt{1-\frac{m}{n}}} \frac{n^{n-m}}{(n-m)^{n-m} e^m} \quad \text{(“Massaging”)} \\
&= \frac{1}{\sqrt{1-\frac{m}{n}}} \frac{1}{\left(1-\frac{m}{n}\right)^{n-m} e^m} \\
&= \frac{1}{\sqrt{1-\frac{m}{n}}} \frac{1}{\left(1-\frac{m}{n}\right)^{n} e^m} \cdot \left(1-\frac{m}{n}\right)^m \\
&= \frac{1}{\sqrt{1-\frac{c}{\sqrt{n}}}} \frac{1}{\left(\left(1-\frac{c}{\sqrt{n}}\right)^{\sqrt{n}} e^c\right)^{\sqrt{n}}} \cdot \left(1-\frac{c}{\sqrt{n}}\right)^{c\sqrt{n}} \\
&\text{(Finally!)}
\end{aligned}$$

the last line using our choice of m. From there, “all” that remains to check is that $\lim_{n\to\infty} \sqrt{1-\frac{c}{\sqrt{n}}} = 1$ (easy), that

$$\lim_{n\to\infty} \left(\left(1-\frac{c}{\sqrt{n}}\right)^{\sqrt{n}} e^c\right)^{\sqrt{n}} = e^{-c^2/2}$$

(less easy), and that

Do it!

$$\lim_{n\to\infty} \left(1-\frac{c}{\sqrt{n}}\right)^{c\sqrt{n}} = e^{-c^2}$$

(not too hard?) to conclude that

$$1 - p_{c\sqrt{n},n} \underset{n\to\infty}{\sim} 1 \cdot \frac{1}{e^{-c^2/2}} \cdot e^{-c^2} = e^{-c^2/2}$$

or, equivalently,

$$p_{c\sqrt{n},n} = 1 - e^{-c^2/2} + o(1)$$

This shows that the probability to get a collision becomes constant for $m = \Theta(\sqrt{n})$. Now, that's nice, but you may ask yourself, *“Well, how did I get here?”*

Let's take a step back. So we throw m balls into n bins. Let's start with the simplest case possible: let's throw *two* balls into n. What's the probability that they end up in the same bin?

Don't look immediately! Think about it first.

The first ball falls into a given bin, say the i-th. Then to get a collision the second ball has to be thrown into the same bin i, which happens with probability $1/n$. So $p_{2,n} = 1/n$. A more verbose way to derive it is as follows: let X_1, X_2 denote the indices of the bins for the first and second ball, respectively. These are independent r.v.'s, uniformly distributed in $[n]$, so

$$\begin{aligned} p_{2,n} = \Pr[\, X_1 = X_2 \,] &= \sum_{k=1}^{n} \Pr[\, X_1 = k, X_2 = k \,] \\ &= \sum_{k=1}^{n} \Pr[\, X_1 = k \,] \cdot \Pr[\, X_2 = k \,] \\ &= \sum_{k=1}^{n} \frac{1}{n} \cdot \frac{1}{n} = \frac{n}{n^2} = \frac{1}{n}\,, \end{aligned} \tag{20}$$

"as foretold."

Back to the general m balls case. The above tells us that for every pair of balls, the probability to get a collision (ignoring all other balls) is $1/n$. How many distinct pairs of balls do we have? Well, $\binom{m}{2}$. So what's the *expected* number of collisions $c(m,n)$?

$$\begin{aligned} c(m,n) &= \mathbb{E}\left[\sum_{(i,j) \text{ pair}} \mathbb{1}_{\{X_i = X_j\}} \right] \\ &= \sum_{(i,j) \text{ pair}} \mathbb{E}\left[\mathbb{1}_{\{X_i = X_j\}} \right] \\ &= \sum_{(i,j) \text{ pair}} \Pr[\, X_i = X_j \,] \\ &= \sum_{(i,j) \text{ pair}} \frac{1}{n} \\ &= \binom{m}{2} \cdot \frac{1}{n} \end{aligned} \tag{21}$$

where we used linearity of expectation, and our previous computation for the $m = 2$ case. This means that the expected number of collisions grows (roughly) as $\frac{m^2}{2n}$. If we believe that the number of collisions does not deviate too pathologically from its expected value, this becomes constant when $m = \Theta(\sqrt{n})$. So we should start expecting collisions when $m = \Theta(\sqrt{n})$, which explains (in hindsight) the result we got before!

But can we easily prove this "intuition"? We have the expectation $c(m,n)$ of the number of collisions, we want to show that number (let's call this random variable C) does not deviate too far from its expectation. The most basic tools we've seen for this are Markov and Chebyshev's inequalities: here, we'll have to use Chebyshev. So we need to compute the variance of our random variable C:

Do you see why?

$$C = \sum_{(i,j) \text{ pair}} \mathbb{1}_{\{X_i = X_j\}}$$

where as before X_i is the index of the bin where the i-th ball lands, and $\mathbb{1}_{\{X_i = X_j\}}$ is the indicator of the event *"ball i and ball j*

collide." We *would like* to write that the variance of the sum is the sum of the variances ("linearity of variance"), something like this

$$\operatorname{Var}[C] = \operatorname{Var}\left[\sum_{(i,j)\text{ pair}} \mathbb{1}_{\{X_i=X_j\}}\right] \overset{?}{=} \sum_{(i,j)\text{ pair}} \operatorname{Var}\left[\mathbb{1}_{\{X_i=X_j\}}\right]$$

which *would* make our life so much easier, since then, using the variance of an indicator random variable (Bernoulli), we'd get

$$\operatorname{Var}[C] = \binom{m}{2}\frac{1}{n}\left(1-\frac{1}{n}\right)$$

Unfortunately, *variance is not linear*: we could write the above $\overset{?}{=}$ equality *if* the indicator variables $\mathbb{1}_{\{X_i=X_j\}}$ were independent (across (i,j)): *and this is not the case here.*

Do you see why they are not independent?

And yet, since we are showing the bin (for each ball) *uniformly* at random, some magic happens, and somehow the above expression is still true.

In the proof below, locate exactly where we use the fact that the bin is chosen uniformly.

Lemma 16.1 (Well, actually. . . (⋆⋆))**.** *We have*

$$\operatorname{Var}[C] = \binom{m}{2}\frac{1}{n}\left(1-\frac{1}{n}\right)$$

Proof. Since $\operatorname{Var}[C] = \mathbb{E}[C^2] - \mathbb{E}[C]^2$ and we already have computed $\mathbb{E}[C]$, we only are missing the first term:

$$\begin{aligned}\mathbb{E}\left[C^2\right] &= \mathbb{E}\left[\left(\sum_{(i,j)\text{ pair}} \mathbb{1}_{\{X_i=X_j\}}\right)^2\right] \\ &= \mathbb{E}\left[\sum_{(i,j)\text{ pair}}\sum_{(k,\ell)\text{ pair}} \mathbb{1}_{\{X_i=X_j\}}\mathbb{1}_{\{X_k=X_\ell\}}\right] \\ &= \sum_{(i,j)\text{ pair}}\sum_{(k,\ell)\text{ pair}} \mathbb{E}\left[\mathbb{1}_{\{X_i=X_j\}}\mathbb{1}_{\{X_k=X_\ell\}}\right]\end{aligned}$$

This is a little intimidating to compute, but we can make the following observations about the summands:

- if the two pairs (i,j), (k,ℓ) are the same,[7] then $\mathbb{1}_{\{X_i=X_j\}}\mathbb{1}_{\{X_k=X_\ell\}} = \mathbb{1}^2_{\{X_i=X_j\}} = \mathbb{1}_{\{X_i=X_j\}}$, so
$$\mathbb{E}\left[\mathbb{1}_{\{X_i=X_j\}}\mathbb{1}_{\{X_k=X_\ell\}}\right] = \mathbb{E}\left[\mathbb{1}_{\{X_i=X_j\}}\right] = \frac{1}{n}.$$
There are exactly $\binom{m}{2}$ such summands.
- if the two pairs (i,j), (k,ℓ) are disjoint, then $\mathbb{1}_{\{X_i=X_j\}}$, $\mathbb{1}_{\{X_k=X_\ell\}}$ are independent, and so
$$\mathbb{E}\left[\mathbb{1}_{\{X_i=X_j\}}\mathbb{1}_{\{X_k=X_\ell\}}\right] = \mathbb{E}\left[\mathbb{1}_{\{X_i=X_j\}}\right]\mathbb{E}\left[\mathbb{1}_{\{X_k=X_\ell\}}\right] = \frac{1}{n^2}.$$
There are exactly $\binom{m}{2}\binom{m-2}{2}$ such summands.

[7] When we consider pairs here, we don't care about ordering, so $(i,j) = (j,i)$.

- else, then the two pairs (i, j), (k, ℓ) are neither disjoint nor equal, then $|\{i, j, k, \ell\}| = 3$. For any such summand, $\mathbb{1}_{\{X_i = X_j\}}\mathbb{1}_{\{X_k = X_\ell\}}$ is of the form $\mathbb{1}_{\{X_i = X_j = X_k\}}$, and so

$$\begin{aligned}\mathbb{E}\left[\mathbb{1}_{\{X_i = X_j\}}\mathbb{1}_{\{X_k = X_\ell\}}\right] &= \mathbb{E}\left[\mathbb{1}_{\{X_i = X_j = X_k\}}\right] \\ &= \sum_{b=1}^{n} \Pr[\,X_i = b, X_j = b, X_k = b\,] \\ &= \frac{n}{n^3} \\ &= \frac{1}{n^2}\,.\end{aligned}$$

There are exactly $2 \cdot \binom{3}{2}\binom{m}{3} = 6\binom{m}{3}$ such summands.

Can you see why? (If that's any consolation, I am terrible at combinatorics.)

As a sanity check, we do have $\binom{m}{2} + \binom{m}{2}\binom{m-2}{2} + 6\binom{m}{3} = \binom{m}{2}^2$, so we did not miss any summand in the above distinction of cases. We can then rewrite

$$\begin{aligned}\mathbb{E}\left[C^2\right] &= \binom{m}{2} \cdot \frac{1}{n} + \binom{m}{2}\binom{m-2}{2} \cdot \frac{1}{n^2} + 6\binom{m}{3} \cdot \frac{1}{n^2} \\ &= \binom{m}{2} \cdot \frac{1}{n} + \binom{m}{2}^2 \cdot \frac{1}{n^2} - \binom{m}{2} \cdot \frac{1}{n^2} && \text{(Magic?)} \\ &= \binom{m}{2} \cdot \frac{1}{n}\left(1 - \frac{1}{n}\right) + \binom{m}{2}^2 \cdot \frac{1}{n^2}\end{aligned}$$

That's really encouraging, since the second term is exactly $\mathbb{E}[C]^2$, and the first is what we were hoping to get for the variance. And, indeed:

$$\begin{aligned}\mathrm{Var}[C] = \mathbb{E}\left[C^2\right] - \mathbb{E}[C]^2 &= \binom{m}{2}\frac{1}{n}\left(1 - \frac{1}{n}\right) + \binom{m}{2}^2\frac{1}{n^2} - \left(\binom{m}{2}\frac{1}{n}\right)^2 \\ &= \binom{m}{2}\frac{1}{n}\left(1 - \frac{1}{n}\right),\end{aligned}$$

concluding the proof. □

Here, we were lucky: somehow in the variance calculation some terms "magically cancel out" and we get the same expression as if things were independent. *This is not usually the case!* But there are some 'ways to handle things nonetheless. For instance:

- If $X_1, \dots, X_n$ are *negatively correlated*, then

$$\mathrm{Var}[\sum_{i=1}^{n} X_i] \leq \sum_{i=1}^{n} \mathrm{Var}[X_i]$$

- Since $\mathrm{Var}[X] = \mathbb{E}\left[X^2\right] - \mathbb{E}[X]^2$, we can always write

$$\mathrm{Var}[X] \leq \mathbb{E}\left[X^2\right].$$

Sometimes, it's good enough!

Now we have the expectation (Eq. (21)), we have the variance (Lemma 16.1), and we have Chebyshev. For any $t > 0$,

$$\Pr[\,|C - c(m,n)| \geq t\,] \leq \frac{\operatorname{Var}[C]}{t^2} \leq \frac{c(m,n)}{t^2}$$

There are also other “fancier” ways, such as the Efron–Stein inequality, but that’s slightly out of scope. Check it out if interested!

Let’s set $m = \lfloor 3\sqrt{n} \rfloor$, so that (using $n \geq 2$)

$$c(m,n) = \binom{\lfloor 3\sqrt{n} \rfloor}{2} \cdot \frac{1}{n} \geq 2$$

(it’s not immediate, but can be checked); and $t := c(m,n)$. We then have

$$\Pr[\,C = 0\,] \leq \Pr[\,|C - c(m,n)| \geq c(m,n)\,] \leq \frac{1}{c(m,n)} \leq \frac{1}{2}$$

showing that *we have at least a 50% chance to get a collision as soon as* $m \geq \lfloor 3\sqrt{n} \rfloor$. And conversely, using this time Markov’s inequality,

$$\Pr[\,C \neq 0\,] = \Pr[\,C \geq 1\,] \leq \mathbb{E}[C] = c(m,n) \leq \frac{m^2}{2n}$$

This inequality, $\Pr[\,X = 0\,] \geq 1 - \mathbb{E}[X]$ for X integer-valued, is sometimes referred to as the *first moment method*.

which is less than 50% for $m \leq \lfloor \sqrt{n} \rfloor$. To sum up, we proved, using Chebyshev’s and Markov’s inequalities:

Theorem 17. *The probability to get* at least one collision *when throwing m independent and uniformly at random in n bins is less than $1/2$ when $m \leq \lfloor \sqrt{n} \rfloor$, and at least $1/2$ as soon as $m \geq \left\lfloor \sqrt{3n} \right\rfloor$.*

confirming the empirical observations and (hopefully) gaining some intuition along the way.

Applications

- Hashing, and hash functions
- Distribution testing (statistics)
- Lower bounds for other problems!

Coverage

Another very natural thing to ask is *when each bin will have received at least one ball*. This is often referred to as the *coupon collector* problem, a term coined a long time ago, when computer scientists were eating cereals for breakfast hoping to collect all of the coupons (cards) of a collection, one cereal box at a time.

Obviously, since we are trying to hit at least each of n bins at least once, we need to throw at least $m \geq n$ balls. But is it enough?

We will stick with the balls-and-bins scenario. But yes, gotta catch’em all!

> What is the expected number of balls $M(n)$ one needs to throw before each of the n bins contains at least one of the m balls?

To figure it out, we can start by trying to simulate the experiment.

This code is definitely *not* optimised!

```python
import numpy as np
import random
def coverage(n):
    (m,ncovered) = (0,0)
    covered = np.zeros(n)
    while ncovered < n:
        draw = random.randint(1, n);
        if covered[draw-1] == 0:
            covered[draw-1] = 1
            ncovered += 1
        m += 1
    return m
```

```python
list_n = np.arange(10, 1001);
experiments_avg = np.zeros(np.size(list_n));
experiments_std = np.zeros(np.size(list_n));
for i in range(len(list_n)):
    coverages_trials = [coverage(list_n[i]) for _ in range(100)];
    experiments_avg[i] = np.mean(coverages_trials);
    experiments_std[i] = np.std(coverages_trials);
```

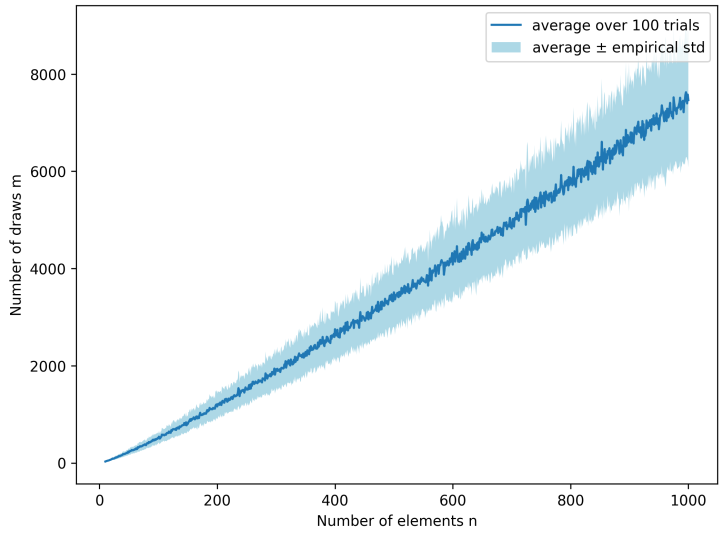


Figure 5: Average (over 100 trials) of the number of balls thrown until all of the n bins contain at least one ball, as a function of n. The range given by one empirical standard deviation is plotted alongside.

Looking at the graph above, we can see how the average number of balls to throw grows with n. But what is it, quantitatively? How does $M(n)$ behaves?

- $\Theta(n)$?
- $\Theta(n \log n)$?
- $\Theta(n^{3/2})$?
- Something else?

And *why*?

Some intuition. Let's look at what happens when $m = n$: how many bins *haven't* been hit by a balls when we have thrown n of them. The probability a fixed bin does not contain any ball is

$$\left(1 - \frac{1}{n}\right)^n \approx e^{-1}$$

and so the expected number of bins with no balls, by linearity of expectation, is

$$\mathbb{E}[\text{empty bins after } n \text{ balls}] = \sum_{i=1}^{n} \Pr\left[\substack{\text{bin } i \text{ empty}\\ \text{after } n \text{ balls}}\right] = n \cdot \left(1 - \frac{1}{n}\right)^{n} \approx \frac{n}{e} \tag{22}$$

This means that after throwing n balls, we still have a constant fraction ($\approx 1/e$) of bins to still hit. Repeating the argument, if we throw n more balls, we expect to still have $\approx 1/e^2$ empty bins; n more balls, and the remaining fraction will be $\approx 1/e^3$; etc. Each time we throw n more balls, we decrease (in expectation) the number of empty bins by a constant factor, so... to bring the expected number of empty bins to < 1, we'll need to repeat that $\Theta(\log n)$ times. This argument tells us that

Generalise Eq. (22) to m bins, to compute directly

$$\mathbb{E}[\text{empty bins after } m \text{ balls}]$$

and solve for m to get this expectation to be less than 1, say $1/2$. Show you retrieve the $\Theta(n \log n)$.

$$M(n) = \Theta(\log n) \cdot n = \Theta(n \log n)$$

sounds reasonable. Can we prove it?

Theorem 18. *We have*

$$M(n) = nH_n \,.$$

where $H_n = \sum_{k=1}^{n} \frac{1}{k}$ is the n-th Harmonic number.

Before proving this, recall the following fact:

Prove the first-order term: $H_n = \Theta(n \log n)$.

Fact 18.1. *The n-th Harmonic number satisfies*

$$H_n = n \ln n + \gamma n + O(1) \,,$$

where $\ln$ *is the natural logarithm and* $\gamma \approx 0.5772$ *is the Euler–Mascheroni constant.*

Proof of Theorem 18. To establish this result, we will introduce some auxiliary random variables, so that we can reduce everything to the one good tool we have – linearity of expectation. For $1 \leq i \leq n$, denote by T_i the number of balls needed, after hitting the $(i-1)$-th distinct bin so far, to hit a new one (the i-th bin). So for instance, $T_1 = 1$ (the first ball we throw by definition hits a new bin, and we had not hit any before), and T_2 is the number of balls we need to throw after that to get a ball in another bin than that first one. It's at least 1, and, if we're unlucky and keep throwing balls into the very same bin, could be much more than that.

The total number of balls to throw before hitting all bins is then, by definition,

$$T_1 + T_2 + \cdots + T_n$$

and so, by linearity of expectation,

$$M(n) = \mathbb{E}[T_1 + T_2 + \cdots + T_n] = \sum_{i=1}^{n} \mathbb{E}[T_i] \,. \tag{23}$$

It remains to get a handle on $\mathbb{E}[T_i]$, for $i \geq 1$. We have seen that $T_1 = 1$ always, so $\mathbb{E}[T_1] = 1$; what about $i \geq 2$? Given that we have hit $i-1$ distinct bins already, the next ball we throw has a probability

$$\frac{n-(i-1)}{n}=\frac{n-i+1}{n}$$

to hit one of the remaining empty $n-(i-1)$ bins, out of n total. We keep throwing balls, each with this probability of success, until we do hit an empty bin: so T_i is a Geometric random variable with parameter $p_i := \frac{n-i+1}{n}$, and so its expectation is

See previous chapter.

$$\mathbb{E}[T_i]=\frac{1}{p_i}=\frac{n}{n-i+1}\,.$$

Plugging this in (23) gives us

$$M(n)=\sum_{i=1}^{n}\frac{n}{n-i+1}=\sum_{j=1}^{n}\frac{n}{j}=nH_n$$

concluding the proof. □

Before going further, let us see how this identity we just proved compared to our empirical average:

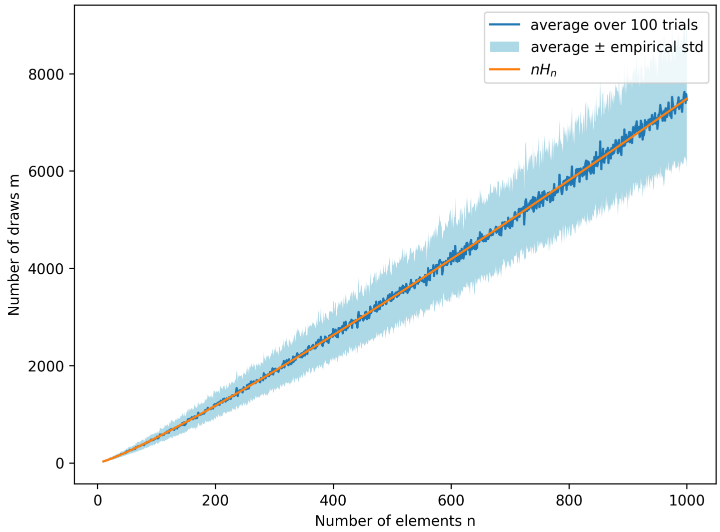


Figure 6: Average (over 100 trials) of the number of balls thrown until all of the n bins contain at least one ball, as a function of n, along with the theoretical value nH_n shown in Theorem 18. The range given by one empirical standard deviation is plotted alongside.

Not bad! That does it for the expectation... but the figure also hints the variance is not too bad, and the number of balls needed to hit all n looks quite concentrated around its expectation. To confirm this (in view of, if we wanted to, applying Chebyshev's inequality), can we also compute the variance?

The amazing thing is that not only we *can*, it's also quite easy.

Theorem 19. *The variance of the number $m(n)$ of bins needed to hit all n bins satisfies*

$$\operatorname{Var}[m(n)]\le\frac{\pi^2}{6}n^2\,.$$

Proof. We start as in the proof of Theorem 18, writing

$$m(n)=T_1+\cdots+T_n$$

The crucial observation is that (suprinsingly?), *the random variables $T_1, \dots, T_n$ are independent.* Intuitively, this is because, once you have hit $i-1$ bins, the number of *new* balls you need to cover the remaining $n-(i-1)$ does not depend on how many balls you already threw: it only depends on n and i, and "doesn't care about the past."

This is called the memorylessness property of the geometric distribution, but try to first convince yourself of this without giving it a label.

This is great, because computing the variance becomes immediate:

$$\mathrm{Var}[m(n)] = \mathrm{Var}[T_1 + \cdots + T_n] = \sum_{i=1}^{n} \mathrm{Var}[T_i]$$

and we already saw that $T_i \sim \mathrm{Geom}(p_i)$ with $p_i = \frac{n-i+1}{n}$, "so" its variance is

$$\mathrm{Var}[T_i] = \frac{1-p_i}{p_i^2} \leq \frac{1}{p_i^2} = \frac{n^2}{(n-i+1)^2}$$

Check it yourself: we don't lose much by ignoring the $-p_i$ term.

and we get

$$\mathrm{Var}[m(n)] \leq n^2 \cdot \sum_{i=1}^{n} \frac{1}{(n-i+1)^2} = n^2 \cdot \sum_{j=1}^{n} \frac{1}{j^2} \leq n^2 \cdot \frac{\pi^2}{6},$$

the last inequality recalling that $\sum_{k=1}^{\infty} \frac{1}{k^2} = \frac{\pi^2}{6}$. □

We won't go through too much here, but for instance, by Chebyshev's inequality, this means that

$$m(n) = nH_n \pm O(n) \tag{24}$$

with probability at least 0.99; combining this with Fact 18.1, similarly,

$$m(n) = n \ln n \pm O(n) \tag{25}$$

with probability at least 0.99 (with a different constant in the $O(\cdot)$).

To conclude this part... how did we do with this variance bound? Well, let us see the empirical simulation again, adding them to the mix:

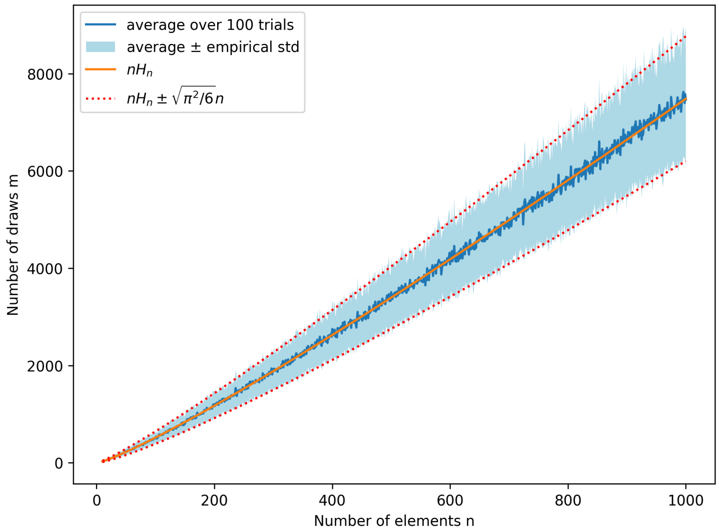


Figure 7: Average (over 100 trials) of the number of balls thrown until all of the n bins contain at least one ball, as a function of n, along with the theoretical value nH_n shown in Theorem 18. The range given by one empirical standard deviation is plotted alongside, as well as the theoretical standard deviation bound obtained.

Quite good!

Load balancing

For the last problem considered in this chapter, let us fix $m = n$, and look at how "balanced" the bin contents are. In particular, we will be interested in the *maximum load* of the bins:

This has applications to, *e.g.*, resource allocations, scheduling, etc.

> We throw n balls into n bins: what the (expected) number $L(n)$ of balls the *fullest* bin will contain?

Let us denote by $L_1, \dots, L_n$ the number of balls contained in each of the n bins. We have, of course, $L_i \leq n$ (number of balls in total) for every i. But that's... quite weak.

It is not hard to see that each bin, separately, follows a Binomial distribution with parameters n and $1/n$ and so bin i will have expected load

$$\mathbb{E}[L_i] = n \cdot \frac{1}{n} = 1$$

Each L_i is a $\mathrm{Bin}(n, 1/n)$ random variable: but $L_1, \dots, L_n$ are *not* independent.

and we also get

$$\mathrm{Var}[L_i] = n \cdot \frac{1}{n}\left(1 - \frac{1}{n}\right) \leq 1$$

This implies, By Chebyshev's inequality, that for each $1 \leq i \leq n$, and setting $t := \sqrt{2n}$,

$$\Pr\left[\, L_i \geq 1 + \sqrt{2n} \,\right] \leq \Pr[\, |L_i - \mathbb{E}[L_i]| \geq t \,] \leq \frac{\mathrm{Var}[L_i]}{t^2} \leq \frac{1}{2n}$$

and so, by a union bound over the n bins,

$$\Pr\left[\max_{1 \leq i \leq n} L_i \geq 1 + \sqrt{2n} \right] \leq n \cdot \frac{1}{2n} = \frac{1}{2}$$

That "simple" application of Chebyshev shows the maximum load $L = \max_{1 \leq i \leq n}$ is $O(\sqrt{n})$ with constant probability. But is it tight? And what does that tell us about $L(n) = \mathbb{E}[L]$?

As in the previous sections, before jumping to conclusions, let's run a simulation.

```
def maxload(n,m):
    loads = np.zeros(n, dtype=int)
    for _ in range(m):
        draw = random.randint(1, n);
        loads[draw-1] += 1
    return np.max(loads)
```

```
list_n = np.arange(10, 1001);
experiments_maxload_avg = np.zeros(np.size(list_n));
experiments_maxload_std = np.zeros(np.size(list_n));
for i in range(len(list_n)):
    maxload_trials = [maxload(list_n[i],list_n[i]) for _ in range(100)];
    experiments_maxload_avg[i] = np.mean(maxload_trials);
    experiments_maxload_std[i] = np.std(maxload_trials);
```

Looking at the graph (Fig. 8), we can see how the average maximum load grows with n. But what is it, quantitatively? How does $L(n)$ behaves?

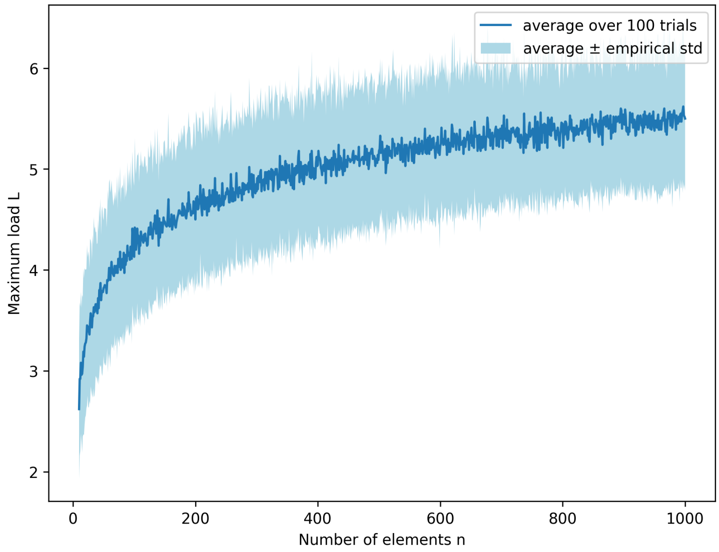


Figure 8: Average (over 100 trials) of the maximum load when throwing n balls into n bins, as a function of n. The range given by one empirical standard deviation is plotted alongside.

- $\Theta(\log n)$?
- $\Theta(\sqrt{n})$?
- $\Theta(n)$?
- Something else?

And *why*?

This one will be tough.

We will show that, oddly, the answer is "something else." Something quite surprising:

Theorem 20. *The expected maximum load $L(n)$ when throwing uniformly and independently n balls into n bins grows as*

$$L(n) = \Theta\left(\frac{\log n}{\log\log n}\right).$$

Proof of the upper bound of Theorem 20. Fix any $1 \leq i \leq n$, and consider the load in bin i. For $0 \leq \ell \leq n$, we will give an upper bound on the probability that at least ℓ balls fall in this bin: namely,

$$\Pr[\, L_i \geq \ell \,] \leq \frac{1}{\ell^\ell} \tag{26}$$

To prove this: $\Pr[\, L_i \geq \ell \,]$ is the probability that there exists a subset $S \subseteq [n]$ of size at least ℓ (a subset of our n balls), and these $|S|$ balls are *exactly* the ones which fell in the i-th bin (not a single other): for a fixed S, this has probability $\left(\frac{1}{n}\right)^{|S|}\left(1-\frac{1}{n}\right)^{n-|S|}$. We have

$$\begin{aligned}
\Pr[\, L_i \geq \ell \,] &= \sum_{\substack{S\subseteq[n]\\ |S|\geq \ell}} \frac{1}{n^{|S|}}\left(1-\frac{1}{n}\right)^{n-|S|} \\
&= \sum_{k=\ell}^{n} \sum_{\substack{S\subseteq[n]\\ |S|=k}} \frac{1}{n^{k}}\left(1-\frac{1}{n}\right)^{n-k} \\
&= \sum_{k=\ell}^{n} \binom{n}{k}\frac{1}{n^{k}}\left(1-\frac{1}{n}\right)^{n-k} \\
&\qquad (\tbinom{n}{k}\text{: number of subsets of size } k) \\
&\leq \binom{n}{\ell}\frac{1}{n^{\ell}}\left(1-\frac{1}{n}\right)^{n-\ell} \\
&\leq \binom{n}{\ell}\frac{1}{n^{\ell}}
\end{aligned}$$

where in the second-to-last inequality, we dropped for convenience some non-negative terms and, in the last step, a factor smaller than one (but still not too small, so it's not too bad to drop it: $e^{-1} \approx \left(1-\frac{1}{n}\right)^{n} \leq \left(1-\frac{1}{n}\right)^{n-\ell} \leq 1$). From here, we will use this *very* convenient and useful inequality on binomial coefficients:

The hope being that the summands decay quick enough with k that the first term, which we kept, "contributes" most of the sum anyways.

Another life saver.

Fact 20.1. *For every* $1 \leq k \leq n$*,*

$$\left(\frac{n}{k}\right)^{k} \leq \binom{n}{k} \leq \left(\frac{en}{k}\right)^{k}.$$

This directly leads to the claimed bound

$$\Pr[\, L_i \geq \ell \,] \leq \left(\frac{en}{\ell}\right)^{\ell}\frac{1}{n^{\ell}} = \frac{e^{\ell}}{\ell^{\ell}}\,.$$

By a union bound over all $1 \leq i \leq n$, we can then conclude that, for every $\ell \geq 1$,

$$\Pr[\, L \geq \ell \,] \leq \sum_{i=1}^{n} \Pr[\, L_i \geq \ell \,] \leq \frac{ne^{\ell}}{\ell^{\ell}} \qquad (27)$$

This is a very good bound for large ℓ, but it is quite useless for small ℓ: for instance, for $\ell = 1$, it gives a vacuous bound! Of course, another bound we have is $\Pr[\, L \geq \ell \,] \leq 1$. We will need that, too. Let $\ell(n)$ be the smallest value such that

$$\ell(n)^{\ell(n)} e^{-\ell(n)} \geq n\,. \qquad (28)$$

This is the value of ℓ starting at which we should switch from using $\Pr[\, L \geq \ell \,] \leq 1$ to using $\Pr[\, L \geq \ell \,] \leq \frac{ne^{\ell}}{\ell^{\ell}}$, as the latter becomes better.

Alright: let us proceed to bounding the expectation of L. We can write, since L is a non-negative integer-valued random variable,

Dividing the summation in two parts and using a different bound for both is a standard, handy trick.

$$\begin{aligned}
L(n) = \mathbb{E}[L] &= \sum_{\ell=1}^{\infty} \Pr[\,L \geq \ell\,] \\
&= \sum_{\ell=1}^{\ell(n)} \Pr[\,L \geq \ell\,] + \sum_{\ell=\ell(n)+1}^{\infty} \Pr[\,L \geq \ell\,] \\
&\leq \sum_{\ell=1}^{\ell(n)} 1 + \sum_{\ell=\ell(n)+1}^{\infty} \frac{ne^{\ell}}{\ell^{\ell}} && \text{(Where the action happens)} \\
&\leq \ell(n) + \sum_{\ell=\ell(n)+1}^{\infty} \frac{ne^{\ell}}{\ell(n)^{\ell}} \\
&= \ell(n) + \frac{ne^{\ell(n)}}{\ell(n)^{\ell(n)}} \sum_{\ell=\ell(n)+1}^{\infty} \frac{e^{\ell-\ell(n)}}{\ell(n)^{\ell-\ell(n)}} \\
&= \ell(n) + \frac{ne^{\ell(n)}}{\ell(n)^{\ell(n)}} \sum_{j=1}^{\infty} \frac{e^{j}}{\ell(n)^{j}} \\
&\leq \ell(n) + \sum_{j=1}^{\infty} \frac{1}{2^{j}} && \left(\text{as } \tfrac{ne^{\ell(n)}}{\ell(n)^{\ell(n)}} \leq 1, \text{ and } \ell(n) \geq 2e\right) \\
&= \ell(n) + 1 && (29)
\end{aligned}$$

so all that remains to do to conclude is to give an upper bound on $\ell(n)$ itself. This part is not too bad: by definition of $\ell(n)$, we know that

$$(\ell(n)-1)^{\ell(n)-1} e^{-(\ell(n)-1))} < n$$

and taking logarithms, we get $(\ell(n)-1)\log(e^{-1}(\ell(n)-1)) < \log n$. One can "easily" show that this implies

Exercise: show it.

$$\ell(n) = \Theta\left(\frac{\log n}{\log\log n}\right)$$

which combined with (29) proves that $\boxed{L(n) = O\left(\frac{\log n}{\log\log n}\right)}$.

What about the lower bound? We will only *sketch* the lower bound in these notes, trying to focus on the key insights. The first insight is that $L_1, \dots, L_n$, which are Binomial r.v.'s with parameters n and $1/n$, are well approximated by a different, "nicer" type of of random variable, *Poisson* random variables with parameter $n \cdot 1/n = 1$. So we will "assume" for convenience that we can instead consider $L'_1, \dots, L'_n \sim \text{Poisson}(1)$. What's more, we will even make the (also not justified! But good for intuition) that these $L'_1, \dots, L'_n$ are independent.

More generally,

$$\boxed{\text{Bin}\left(n, \frac{\lambda}{n}\right) \approx \text{Poisson}(\lambda)}$$

for constant $\lambda > 0$. This is very handy!

This is not an actual proof! But it can be turned into one.

We then can write, since $\mathbb{E}[L] = \sum_{k=1}^{\infty} \Pr[\,L \geq k\,]$, that, for any fixed $\ell \geq 1$ of our choosing,

$$\begin{aligned}
\mathbb{E}[L] \geq \sum_{k=1}^{\ell} \Pr[\,L \geq k\,] &\geq \ell \Pr[\,L \geq \ell\,] = \ell \Pr[\,\exists i,\, L_i \geq \ell\,] \\
&\approx \ell \Pr[\,\exists i,\, L'_i \geq \ell\,] \geq \ell \Pr[\,\exists i,\, L'_i = \ell\,]
\end{aligned}$$

where the $\approx$ is the first "sketchy" Poisson approximation. Using our (unwarranted, sketchy) independence of the L'_i's, we can continue by writing

If $N \sim \text{Poisson}(\lambda)$, then for every non-negative integer k

$$\boxed{\Pr[\,N = k\,] = e^{-\lambda}\frac{\lambda^k}{k!}\,.}$$

$$\begin{aligned}
\Pr[\,\exists i,\, L'_i = \ell\,] &= 1 - \Pr[\,\forall i,\, L'_i \neq \ell\,] \\
&= 1 - \left(1 - \frac{e^{-1}}{\ell!}\right)^n && \text{(Independence)} \\
&\geq 1 - e^{-\frac{n}{\ell!}} && \text{(using } \ln(1-x) \geq -x\text{)} \\
&\geq 1 - e^{-\frac{n}{\ell^\ell}} && \text{(using } \ell! \leq \ell^\ell\text{)}
\end{aligned}$$

Suitably choosing

$$\ell = \Theta\left(\frac{\log n}{\log\log n}\right)$$

we get $e^{-\frac{n}{\ell^\ell}} \leq 1/2$, from which $\Pr[\,\exists i,\, L'_i = \ell\,] \geq 1/2$. So overall (again, modulo the sketchy bits – this is not a full proof), we get

$$\mathbb{E}[L] \geq \ell \cdot \frac{1}{2} = \Omega\left(\frac{\log n}{\log\log n}\right)$$

"showing" the lower bound. □

Alternative (advanced) proof of the upper bound (⋆⋆) Here is a "slick" proof, which seems somewhat magical, but has a couple neat tricks that you will see again or are worth internalizing.

Go over it during the tutorials!

Recall that we want to bound the quantity

$$L(n) = \mathbb{E}\left[\max_{1\leq i\leq n} L_i\right]$$

where the loads $L_1, \dots, L_n$ are *not* independent, but all follow a $\text{Binom}(n, 1/n)$ distribution. One can then give an upper bound on $L(n)$ as follows. First, introduce a free parameter $t > 0$ to be determined later, when we want to optimise the final bound we get.

The idea is to replace the max by a $\sum$ in order to use linearity of expectation – but $\max_i \leq \sum_i$ is too lossy, so first we "exponentiate" the random variables to mitigate that loss, as $\max_i \exp \leq \sum_i \exp$ should be "exponentially less lossy." *But* to exponentiate we write $X = \ln\exp X$, which means we now have a log inside the expectation, and that is not easy to handle: thankfully, ln is concave, so we can use Jensen's inequality to write $\mathbb{E}[\ln] \leq \ln\mathbb{E}[]$. We might also lose something in this step, but that "Jensen gap" is typically small for nice, well-concentrated random variables, so... we can try and hope for the best.

$$\begin{aligned}
L(n) &= \frac{1}{t} \cdot \mathbb{E}\left[\max_{1\leq i\leq n} tL_i\right] \\
&= \frac{1}{t} \cdot \mathbb{E}\left[\ln e^{\max_{1\leq i\leq n} tL_i}\right] \\
&\leq \frac{1}{t} \cdot \ln\mathbb{E}\left[e^{\max_{1\leq i\leq n} tL_i}\right] && \text{(Jensen's)} \\
&= \frac{1}{t} \cdot \ln\mathbb{E}\left[\max_{1\leq i\leq n} e^{tL_i}\right] && (\exp\max_i = \max_i \exp) \\
&\leq \frac{1}{t} \cdot \ln\mathbb{E}\left[\sum_{1\leq i\leq n} e^{tL_i}\right] && (\max_i \leq \textstyle\sum_i) \\
&= \frac{1}{t} \cdot \ln\sum_{1\leq i\leq n} \mathbb{E}\left[e^{tL_i}\right] && \text{(Linearity)} \\
&= \frac{1}{t} \cdot \ln n\mathbb{E}\left[e^{tL_1}\right] && (30)
\end{aligned}$$

where the last step used the fact that $L_1, \dots, L_n$ all have the same distribution. Now, this quantity $\mathbb{E}[e^{tL_1}]$ is called the *moment-generating function* (MGF) of the random variable L_1, and as a function of t it encodes a lot of information about the distribution of L_1. Thankfully, we do *not* have to compute it: it is standard enough that Wikipedia lists the MGFs for most probability distributions of interest, and in particular for a Binomial random variable X with parameters n and p we have

$$\mathbb{E}\left[e^{tX}\right] = (1+(e^t-1)p)^n, \qquad t \in \mathbb{R} \tag{31}$$

In our case, $p = 1/n$, so we get

$$\mathbb{E}\left[e^{tL_1}\right] = \left(1+\frac{e^t-1}{n}\right)^n$$

and, using this along with the standard inequality $\ln(1+x) \leq x$ $(x > -1)$ in (30),

This one is a life saver.

$$\begin{aligned} L(n) &\leq \frac{1}{t}\left(\ln n + \ln \mathbb{E}\left[e^{tL_1}\right]\right) \\ &\leq \frac{1}{t}\left(\ln n + n\ln\left(1+\frac{e^t-1}{n}\right)\right) \\ &\leq \frac{1}{t}(\ln n + e^t - 1) \end{aligned} \tag{32}$$

We're almost there! We still have our free parameter $t > 0$, and we get to choose it however we want in order to get the best upper bound possible (we get a valid upper bound no matter which t we pick). One option to do so would be to differentiate the RHS of (32) to find the minimum: this is unfortunately quite unwieldy. A simpler (and most of the time "good enough" is to observe that

If one does not care too much about the exact constant factors or lower-order terms.

$$\max(a,b) \leq a+b \leq 2\max(a,b), \qquad a,b \geq 0 \tag{33}$$

and so *minimising a sum of two terms is roughly the same as minimising the maximum.* Here we have two terms: $\ln n$ and $e^t - 1$: one way to make sure the maximum is not too bad is to "balance it out", and choose t so that the two terms are equal. In our case, this means choosing

Useful trick: avoids calculus.

$$t := \ln(1+\ln n) = \ln\ln(en) \tag{34}$$

and plugging this choice of t in (32) gives

$$L(n) \leq \frac{2\ln n}{\ln\ln(en)}\,. \tag{35}$$

We're done!

Load balancing: the power of two choices

To conclude, let us mention an even more counter-intuitive result: imagine that instead of throwing n balls uniformly into n bins, each ball instead selects *two* bins uniformly at random, and falls into the *least* full of the two (breaking ties arbitrarily). What becomes the expected maximum load?

This looks strange, but has applications to hashing, task allocation, network broadcasting...

As usual, let us first try to get a sense of what is going on via a simulation:

```python
def maxload2choices(n,m):
    loads = np.zeros(n, dtype=int)
    for _ in range(m):
        draw1 = random.randint(1, n);
        draw2 = random.randint(1, n);
        if loads[draw2-1] > loads[draw1-1]:
            loads[draw1-1] += 1
        else:
            loads[draw2-1] += 1
    return np.max(loads)
```

```python
list_n = np.arange(10, 1001);
experiments_maxload2choices_avg = np.zeros(np.size(list_n));
experiments_maxload2choices_std = np.zeros(np.size(list_n));
for i in range(len(list_n)):
    maxload2choices_trials = [maxload2choices(list_n[i],list_n[i]) for _ in range(100)];
    experiments_maxload2choices_avg[i] = np.mean(maxload2choices_trials);
    experiments_maxload2choices_std[i] = np.std(maxload2choices_trials);
```

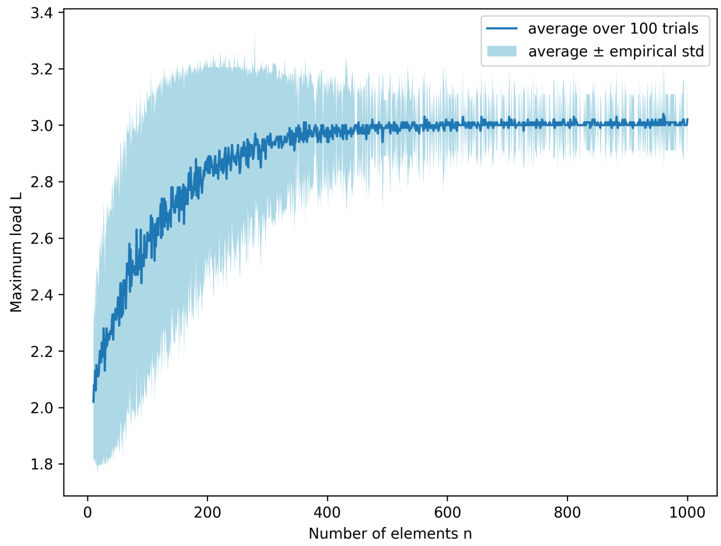


Figure 9: Average (over 100 trials) of the maximum load when throwing n balls into n bins, using the "best of two bins" strategy for each bin, as a function of n. The range given by one empirical standard deviation is plotted alongside.

Looking at the graph (Fig. 9), we can see that with this "best of two choices" the maximum load grows much slower (as a function of n) than in the previous setting (Fig. 8). But how much slower?

- $\Theta(\sqrt{\log n})$?
- $\Theta\left(\frac{\log n}{(\log\log n)^2}\right)$?
- $\Theta(\log\log n)$?
- Something else?

Amazingly, this simple "power of two choices" brings the expected maximum load from $\Theta\left(\frac{\log n}{\log\log n}\right)$ to something *exponentially smaller*:

TODO in class: Visualization of the maximum load as m increases, i.e., as more balls are thrown (same with power of two choices).

Theorem 21. *The expected maximum load $\hat{L}(n)$ when throwing independently n balls into n bins using the "best of two choices" strategy above grows as*

$$\hat{L}(n) = \log\log n + O(1)\,.$$

(We will not prove this theorem in the lecture.)