WARREN D. MACEVOY

LUCK

Contents

List of Figures

List of Tables

Introduction

The point of these notes is to introduce an idea of "luck" that connects mathematical probability with the everyday notion.

As a motivating problem, imagine walking along the beach and asking a random person to toss a tennis ball so that it lands in the sand. The probability that it lands at some point would depend on the habits of the thrower and the details of the beach, but we can summarize this as some probability distribution, $p(x) = \rho(x)dx$, where $x \in \mathbb{R}^2$ is a suitable coordinate system for the beach in question. It would almost certainly not be a uniform distribution, and it would almost certainly not be particularly concentrated.

Traditional probability feels uncomfortable here. The chances of the ball landing at a given point is zero, and so miraculous. Yet anyone watching this process would only occasionaly be surprized by the outcome.

For a continuous probablity distribution such as this, the chance of the ball landing in some small area dx near x is $p(x) = \rho(x)dx$. But the ball lands at a point, so dx is zero, so the probablity $p(x) = \rho(x)dx$ is zero.

As common (and mundane, not miraculous) such situations are, the language of statistics seems to have difficulty with the notion. Nor is it limited to continuous cases, just when there are a lot of possible outcomes. Such examples lead to non-zero but very small probabilities.

The real motivation of this came from the space of passwords a person might choose from, which is an effectively infinite discrete space.

To distinguish from the more general notion of luck, note that that there is no extrinsic value on an outcome. To say something is "lucky" often means there is some value (different from the probability) associated with outcomes. However, outcomes that are the most valuable are often the least probable, and outcomes of equal probability ought to be equally lucky. In the most extreme case of all equally probable outcomes (uniform probability), every outcome should have a luck of $\frac{1}{2}$.

These observations lead to the following definition of luck:

> The luck $L(x)$ of an outcome x is the probability of getting any outcome y that is more probable than x, plus one-half the probability of getting any outcome y that is equally probable to x.

From the perspective of discussing luck, it it convenient to have few sets: $\Omega(x)$, the outcomes more likely than x, and $\omega(x)$, the outcomes equally likely to x.

Definition 1. Omega. $\Omega(x)$ is set of outcomes more likely than x:

$$\Omega(x) = \{y \mid p(y) > p(x)\} \,. \tag{1}$$

We define $|\Omega(x)|$ as the probability an outcome is in $\Omega(x)$, $|\Omega(x)| = P(y \in \Omega(x))$. In the discrete case, this is

$$|\Omega(x)| = \sum_{y \in \Omega(x)} p(y), \tag{2}$$

and, in the continuous case,

$$|\Omega(x)| = \int_{\Omega(x)} \rho(y)\, dy \,. \tag{3}$$

Definition 2. omega. $\omega(x)$ is the set of outcomes equally likley to x:

$$\omega(x) = \{y \mid p(y) = p(x)\} \,. \tag{4}$$

For the tyical case of many outcomes with different probabilities, $|\omega(x)|$ is small. For example, $|\omega(x)| = 0$ for any multivariate normal distribution.

Similar to $\Omega(x)$, we define $|\omega(x)|$ as the probability an outcome is in $\omega(x)$, $|\omega(x)| = P(y \in \omega(x))$.

In the discrete case, this is

$$|\omega(x)| = \sum_{y \in \omega(x)} p(y), \tag{5}$$

and, in the continuous case,

$$|\omega(x)| = \int_{\omega(x)} \rho(y)\, dy \,. \tag{6}$$

With these definitions in place, we define luck mathematically as follows:

Definition 3. Luck. The luck of an outcome is the probability getting any more likely outcome, plus half the probability of getting any equally likely outcome:

$$L(x) = |\Omega(x)| + \frac{1}{2}|\omega(x)| \,. \tag{7}$$

Properties of Luck.

- $0 \leq L(x) \leq 1$. This ranges from no luck to perfect luck.

- If $L(x)$ is close to 1, then $p(x)$ is comparatively small, and most outcomes would have a higher probability (you are lucky).

- If $L(x)$ is close to 0, then $p(x)$ is comparatively large, and most outcomes would have a lower probability (you are unlucky).

- $E(L) = \frac{1}{2}$. On average, luck is always 50:50.

We are interested in cases which have many possible outcomes with low but somewhat different probabilities (like the tennis ball on the beach). If the space is well divided (so $\max |\omega| = \max_x |\omega(x)|$ is small), then there are other interesting properties of luck:

- $E(f(L)) = \int_0^1 f(L)dL + \varepsilon$, where $|\varepsilon| \leq \max |f''| \cdot \max |\omega|^2/12$.

- For $p \geq 1$, $E(L^p) = 1/(p+1) - \varepsilon$, $0 \leq \varepsilon \leq p \cdot (p-1) \max |\omega|^2/12$.

- For $0 \leq a \leq b \leq 1$, $E(L \in [a,b]) = b - a + \varepsilon$, $|\varepsilon| \leq \max |\omega|$.

In particular $|\omega(x)| = 0$ for the normal, exponential, beta, and gamma distributions. As a worst-case counterexample, the flattest distribution is the uniform distribution, for which $|\omega(x)| = 1$.

Example 1. Coins. Suppose we toss 8 fair coins. The probability of getting exactly x heads out of 8 tosses is given by the binomial distribution

$$p(x) = \frac{8!}{x!(8-x)!} \left(\frac{1}{2}\right)^8 \qquad (8)$$

What is the luck associated with this distribution?

x	$p(x)$	$\Omega(x)$	$|\Omega(x)|$	$\omega(x)$	$|\omega(x)|$	$L(x)$
4	0.2734	{}	0.0000	{4}	0.2734	0.1367
3 or 5	0.2188	{4}	0.2734	{3,5}	0.4375	0.4922
2 or 6	0.1094	{3,4,5}	0.7109	{2,6}	0.2188	0.8203
1 or 7	0.0313	{2,3,4,5,6}	0.9297	{1,7}	0.0625	0.9609
0 or 8	0.0039	{1,2,3,4,5,6,7}	0.9922	{0,8}	0.0078	0.9961

Table 1: Table is arranged in increasing luck (which is decreasing probability). Getting exactly $x = 4$ heads is unlucky, requiring only $L = 14\%$ luck, while getting $x = 0$ or $x = 8$ heads is almost 100% luck.

$p = 0.27$ $p = 0.22$ $p = 0.22$ $p = 0.11$ $p = 0.11$

$L(x = 4) = 0.14$ $L(x = 3 \text{ or } 5) = 0.49$ $L(x = 6 \text{ or } 2) = 0.82$ $\ldots$
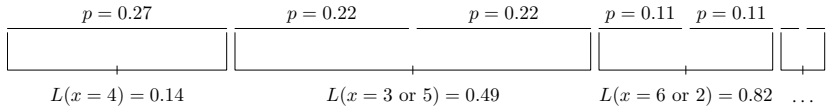
Figure 1: Luck to get x heads on 8 fair coins. After the probabilities are arranged in decreasing order on the unit interval, $L(x)$ is the center point of equally probable outcomes.

Luck on average is $\frac{1}{2}$:

$$E(L) = \sum_{x=0}^{8} p(x) \cdot L(x) = \frac{1}{2}. \qquad (9)$$

For any distribution, $E(L) = \frac{1}{2}$.

The second moment should be close to $\frac{1}{3}$:

$$E(L^2) = \sum_{x=0}^{8} p(x) \cdot L(x)^2 = \frac{1}{3} - 0.0096 \qquad (10)$$

For any distribution, $E(L^2) = \frac{1}{3} - \varepsilon$, with $0 \leq \varepsilon \leq \max |\omega|^2/6$.
For the coin distribution, the bound is $0 \leq \varepsilon \leq 0.032$, and the actual error $\varepsilon = 0.0096$.

The probability luck is in the middle half is about $\frac{1}{2}$:

$$E(L \in [\frac{1}{4}, \frac{3}{4}]) = \sum_{x=0}^{8} p(x) \cdot \left\{ \begin{array}{ll} 1 & \text{if } L(x) \in [\frac{1}{4}, \frac{3}{4}] \\ 0 & \text{otherwise} \end{array} \right\}$$
$$= \frac{1}{2} - 0.0625. \qquad (11)$$

For any distribution, $E(L \in [a,b]) = b - a + \varepsilon$, with $|\varepsilon| \leq \max |\omega|$.
For the coin distribution and $[a,b] = [\frac{1}{4}, \frac{3}{4}]$, $E(L \in [a,b]) = \frac{1}{2} + \varepsilon$ with $|\varepsilon| \leq 0.4375$ as the (poor) error bound, and the actual error of $\varepsilon = 0.0625$

Normal Distribution

Suppose we are in a probability space well approximated by the multivariate normal (Gaussian) distribution of a random variable $x \in \mathbb{R}^n$ with mean μ and non-singular covariance σ:

$$P_{\text{normal}}(x; \mu, \sigma) = \frac{e^{-\frac{1}{2}(x-\mu)^T \sigma^{-1}(x-\mu)}}{\sqrt{(2\pi)^n \det \sigma}}, \tag{12}$$

where

$$\mu_i = E(x_i), \tag{13}$$

and

$$\sigma_{ij} = E((x_i - \mu_i)(x_j - \mu_j)). \tag{14}$$

How lucky is some outcome x? From the definition:

$$L(x) = |\Omega(x)| + \frac{1}{2}|\omega(x)|, \tag{15}$$

where

$$\Omega(x) = \{y \in \mathbb{R}^n \,|\, P_{\text{normal}}(y) > P_{\text{normal}}(x)\} \tag{16}$$

$$= \left\{ y \in \mathbb{R}^n \,\middle|\, |\sqrt{\sigma^{-1}}(y - \mu)| < |\sqrt{\sigma^{-1}}(x - \mu)| \right\} \tag{17}$$

and

$$\omega(x) = \left\{ y \in \mathbb{R}^n \,\middle|\, |\sqrt{\sigma^{-1}}(y - \mu)| = |\sqrt{\sigma^{-1}}(x - \mu)| \right\}. \tag{18}$$

Because $\omega(x)$ has no volume in $\mathbb{R}^n$,

$$|\omega(x)| = \int_{\omega(x)} P_{\text{normal}}(y; \mu, \sigma) \, dy = 0. \tag{19}$$

So

$$L(x) = |\Omega(x)| \tag{20}$$

$$= \int_{\Omega(x)} P_{\text{normal}}(y; \mu, \sigma) \, dy. \tag{21}$$

By changing variables to $z = \sqrt{\sigma^{-1}}(x - \mu)$,

$$L(x) = \int_{|z|<R} P_{\text{normal}}(z; 0, I)\, dz\,, \tag{22}$$

where $R = |\sqrt{\sigma^{-1}}(x - \mu)|$.

This can be evaluated in spherical coordinates:

$$L(x) = \frac{1}{\sqrt{(2\pi)^n}} \int_0^R \frac{n\pi^{n/2}}{\Gamma(\frac{n}{2} + 1)} r^{n-1} e^{-\frac{1}{2}r^2}\, dr \tag{23}$$

$$= \frac{\gamma(n/2, R^2/2)}{\Gamma(n/2)} \tag{24}$$

The last form uses the lower incomplete gamma function, defined to be

$$\gamma(s, x) = \int_0^x t^{s-1} e^{-t}\, dt \tag{25}$$

For any value of n, but particularly for large values, we find the following approximation to be very good:

$$L(x) \approx \frac{1}{2}\left[1 + \text{erf}(|\sqrt{\sigma^{-1}}(x - \mu)| - \sqrt{n - 1/2})\right]. \tag{26}$$

This comes from a taylor expansion of the log of the integrand in (23)

Not only is this result pretty, it is very useful. Suppose we have distribution parameters μ and σ, and would like to know if they fit actual observations. A traditional approach requires a large sample to estimate μ and σ, but we don't need this, nor do we need to assume that the distribution is normal. We just need to ask if the observations are surprising (lucky or unlucky). In large dimensions, numerical experiments suggest one sample is in most cases sufficient to establish practical certainty (probability of error less than 10^{-15}).

$L^{(x)}(x)$	$L^{(y)}(x)$	$L^{(x)}(y)$	$L^{(y)}(y)$
0.501 417 202 0	1.000 000 000 0	1.000 000 000 0	0.838 180 264 1
0.731 421 266 5	1.000 000 000 0	1.000 000 000 0	0.239 258 143 2
0.982 563 033 9	1.000 000 000 0	1.000 000 000 0	0.271 695 512 7
0.033 455 080 7	1.000 000 000 0	1.000 000 000 0	0.421 320 625 9
0.689 429 934 0	1.000 000 000 0	1.000 000 000 0	0.074 461 655 7
0.736 397 593 7	1.000 000 000 0	1.000 000 000 0	0.294 050 728 4
0.304 521 296 7	1.000 000 000 0	1.000 000 000 0	0.707 849 014 7
0.231 111 574 4	1.000 000 000 0	1.000 000 000 0	0.290 313 093 2
0.585 247 719 9	1.000 000 000 0	1.000 000 000 0	0.636 902 202 8
0.214 552 926 1	1.000 000 000 0	1.000 000 000 0	0.268 989 787 4

Table 2: Luck from two randomly generated distributions $\mu^{(x)}$ and $\mu^{(y)}$ uniformly chosen in $[0, 1]^{100}$, and $\sigma^{(x)}, \sigma^{(y)}$ are transposed squares of random 100×100 matrices. In each row, x is a sample from the $\mu^{(x)}, \sigma^{(x)}$, normal distribution, and y is from the $\mu^{(y)}, \sigma^{(y)}$ distribution. The actual values of x and y are not given, since they are very large (100 numbers each) and uninteresting.
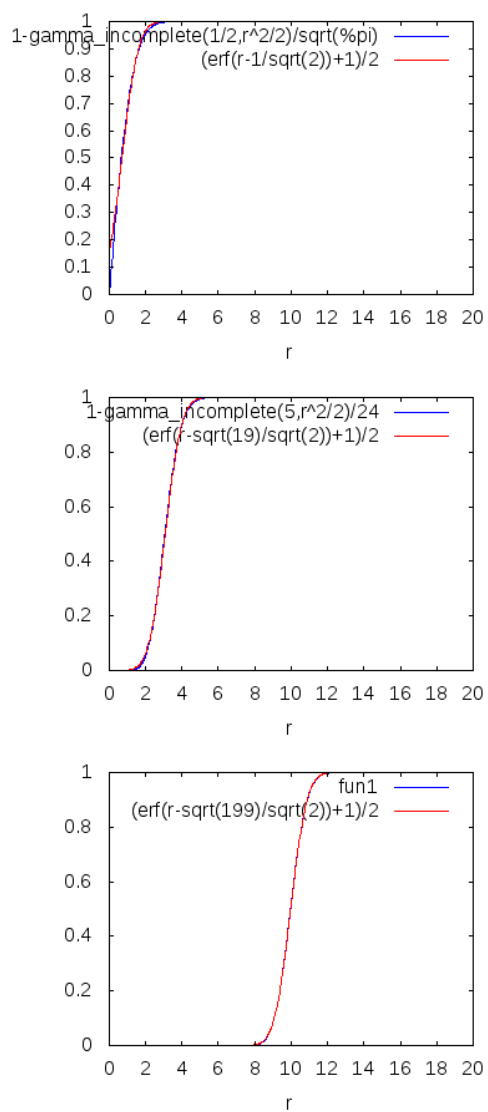
Figure 2: Exact (blue) vs approximate (red) luck for normal distribution for $n = 1$, 10, and 100.

Multinomial Distribution

Selecting independent samples (with repetition) from a discrete set of outcomes follows the multinomial distribution.

Specifically, select N independent samples (with repetition) from a discrete set of possible outcomes $\{x_i \in \mathbb{Z}_{0+}\}_{i=1}^n$ with probabilities p_i, $\sum_{i=1}^n p_i = 1$. Not considering order, this will result in $x_i \in \mathbb{Z}_{0+}$ counts of outcome w_i, $\sum_{i=1}^{N_p} x_i = N$. The probability of such a sample is given by the multinomial distribution:

$$P_{\text{multinomial}}(x; p) = N! \prod_{i=1}^n \frac{p_i^{x_i}}{x_i!} . \tag{27}$$

The counts x_i are not independent (they must sum to N), but if the probabilities are small, then the multinomial distribution can be treated as a product of independent Poisson distributions:

$$P_{\text{poisson}}(x; p) = \prod_{i=1}^{N_p} \frac{e^{-p_i} p_i^{x_i}}{x_i!} . \tag{28}$$

Approximating the Multinomial as a Normal Distribution

The multinomial distribution can be approximated by a normal distribution,

$$\mu_i = N p_i , \tag{29}$$

$$\Sigma_{ij} = \begin{cases} N p_i (1 - p_i) & \text{if } i = j , \\ -N p_i p_j & \text{if } i \neq j . \end{cases} \tag{30}$$

But the covarience matrix Σ is singular. Because of this, instead of the general normal distribution, the multinomial distribution is approximated by the continuous distribution,

$$P(x) = \frac{\exp\left(-\frac{1}{2} \sum_{i=1}^n \frac{(x_i - N p_i)^2}{N p_i}\right)}{\sqrt{(2\pi)^{(n-1)} n \prod_{i=1}^n p_i}} . \tag{31}$$

subject to the constraint,

$$\sum_{i=1}^n x_i = N . \tag{32}$$

Changing coordinates to

$$z_i = \frac{x_i - N p_i}{\sqrt{N p_i}} \tag{33}$$

leads to an integration on the $\mathbb{R}^{n-1}$ hypersurface $n \cdot z = 0$, where

$$n_i = \sqrt{p_i} \tag{34}$$

This has the same form as the multinomial gaussian, except it is one lower dimension, yeilding,

$$L(x) = \frac{\gamma((n-1)/2, |z|^2/2)}{\Gamma((n-1)/2)}.$$

(35)

Computation

Unfortunately, the explicit computations for luck may be impractical. However, if n independent samples are taken $S = (x_1, \ldots, x_n)$, then it can be estimated as:

$$\ell(x) = \frac{1}{n} \{\# \text{ of outcomes in S more probable than } x\}$$

$$+ \frac{1}{2n} \{\# \text{ of outcomes in S equally probable to } x\}$$

It is a reasonably straightforward calculation to show that

$$E(\ell(x)) = L(x), \tag{36}$$

and

$$E((\ell(x) - L(x))^2) \leq \frac{1}{n} L(x) \cdot (1 - L(x)). \tag{37}$$

Example 2. Multinomial. We are aware of no computationally efficient way to exactly compute the luck for a multinomial distribution other than the explicit sum. Suppose there is a questionaire with 4 possible answers with category probabilities 0.1, 0.2, 0.3, and 0.4. How lucky would it be to get 13, 15, 27 and 45 responses of each respective answer in 100 samples? This example is small enough to compute the exact luck of 0.6287498, indicating this is a fairly typical result.

The probability in this case is the multinomial probability (see figure 3):

$$y = \text{mulprobln(x,p)} = \log \left[\left(\sum_{i=1}^{n_p} x_i \right)! \prod_{i=1}^{n_p} \frac{p_i^{x_i}}{x_i!} \right]. \tag{38}$$

Numerical estimates of luck requires samples from the given probability distribution. Figure 4 uses a built-in function in scilab to generate such a sample set.

For smaller spaces, luck for a multinomial distribution can be computed explicitly. Figure 5 computes this as an inefficient testing reference.

```
function y=mulprobln(x,p)
  [nprobs,nsamps]=size(x);
  pln=log(p);
  y=zeros(1,nsamps);
  for i=1:nsamps
    xi=x(:,i);
    ni=sum(xi);
    y(i)=gammaln(ni+1)+sum(xi.*pln-gammaln(xi+1));
  end
endfunction
```

Figure 3: Scilab listing to compute the natural log of the multinomial probability y of outcome x given category probabilities p. p is a $n_{\text{probs}} \times 1$ column vector of category probabilities, x is a $n_{\text{probs}} \times n_{\text{samps}}$ of outcomes, and the result y is a $1 \times n_{\text{samps}}$ row vector of the natural log of multinomial probabilites for these outcomes.

```
function x=mulsamp(nsamps,ntrials,p)
  x=grand(nsamps,"mul",ntrials,p(1:(length(p)-1)));
endfunction
```

Figure 4: Scilab listing to get a sample of outcomes from a multinomial distribution. n_{samps} is the number of desired samples, n_{trials} is the number of trials in each sample, and p is a $n_{\text{probs}} \times 1$ column vector of category probabilities. The result x is a $n_{\text{probs}} \times n_{\text{samps}}$ matrix of sample outcomes, where $\sum_{j=1}^{n_{\text{probs}}} x(j,i) = n_{\text{trials}}$.

```
function el=mulluckrec(nb,mb,lpx,k,p,y,eps)
  [n,m]=size(lpx);
  el=zeros(n,m);
  for yk=0:nb-mb
    y(k)=yk;
    if k < length(p)-1 then
      el=el+mulluckrec(nb,mb+yk,lpx,k+1,p,y,eps);
    else
      y(length(p))=nb-mb-yk;
      lpy=mulprobln(y,p);
      c=0.5*bool2s(lpy > lpx-eps)+0.5*bool2s(lpy > lpx+eps);
      el = el + c .* exp(lpy);
    end
  end
endfunction


function lucks=mulluck(x,p,eps)
  if ~exists("eps","local") then
    eps=sqrt(%eps);
  end
  [nprobs,nsamps]=size(x);
  ntrials=sum(x,'r');
  min_ntrials=min(ntrials);
  max_ntrials=max(ntrials);
  assert_checkequal(min_ntrials,max_ntrials);
  lucks=mulluckrec(min_ntrials,0,mulprobln(x,p),1,p,zeros(nprobs,1),eps)
endfunction
```

Figure 5: Recursively compute luck of multinomial exactly using exaustive sum. x is a $n_{probs} \times n_{samps}$ matrix of outcomes, and p is a $n_{probs} \times 1$ column vector of category probabilities. The result is a $1 \times n_{samps}$ of luck values.

Figure 6: Given the log of the probabilites of a set of sample data, return a table used for quickly estimating luck. `problns` is a $1 \times n_{nsamps}$ row vector of logs of probabilities, and eps is an optional parameter giving the absolute error (in log space) for considering two probabilities to be equal. Returns `setup`, a $3 \times N$ matrix giving estimates for $(L(x), |\Omega(x)|, |\omega(x)|)$ for each unique probability in the sample. This function is useful generally (not just for multinomial distributions).

```
function setup=numlucksetup(problns,eps)
  if ~exists("eps","local") then
    eps=sqrt(%eps);
  end

  n=length(problns);
  problns=gsort(problns);

  i=1;
  count=0;
  while i<=n
    j=i;
    while (j <= n-1 & abs(problns(j+1)-problns(i)) < eps)
      j=j+1;
    end
    i=j+1;
    count=count+1;
  end

  setup=zeros(3,count);

  i=1;
  count=0;
  while i<=n
    j=i;
    while (j <= n-1 & abs(problns(j+1)-problns(i)) < eps)
      j=j+1;
    end

    count=count+1;
    setup(1,count)=0.5*(problns(i)+problns(j));
    setup(2,count)=(i-1)/n;
    setup(3,count)=(j-i+1)/n;
    i=j+1;
  end
endfunction
```

Figure 7: Estimate luck given log of probabilities and setup. `problns` is a row vector of log-probabilities, `setup` is the setup from a (possibly different) sample, and `eps` is the same optional parameter as in `numlucksetup`.

```
function lucks=numluck(problns,setup,eps)
  if ~exists("eps","local") then
    eps=sqrt(%eps);
  end

  [n,m]=size(problns);
  [three,nsetup]=size(setup);

  abs_Omega=zeros(n,m);
  abs_omega=zeros(n,m);

  for i=1:n
    for j=1:m
      lo=1;
      hi=nsetup;
      while %T
        mid=floor((hi+lo)/2)
        if setup(1,mid) > problns(i,j)+eps then
          if mid == lo then
            break
          end
          lo=mid;
        else
          if mid == hi then
            break
          end
          hi=mid;
        end
      end

      if abs(problns(i,j)-setup(1,hi))<eps then
        abs_Omega(i,j)=setup(2,hi);
        abs_omega(i,j)=setup(3,hi);
      elseif abs(problns(i,j)-setup(1,lo))<eps then
        abs_Omega(i,j)=setup(2,lo);
        abs_omega(i,j)=setup(3,lo);
      else
        abs_Omega(i,j)=setup(2,lo);
        abs_omega(i,j)=0.5*exp(problns(i,j));
      end
    end
  end
  lucks=abs_Omega+0.5*abs_omega;
endfunction
```

```
// get sample set
nsamps=10000;
ntrials=100;
p=[0.1;0.2;0.3;0.4];
x=mulsamp(nsamps,ntrials,p);
problns=mulprobln(x,p);

// setup for luck estimates
setup=numlucksetup(problns);

// estimate luck numerically
xo=[13; 15; 27; 45];
problnso=mulprobln(xo,p);
nluck=numluck(problnso,setup);
nsd=sqrt(nluck .* (1-nluck) ./ nsamps);

// optional exact luck
luck=mulluck(xo,p);
sd=sqrt(luck .* (1-luck) ./ nsamps);

// z is approximately normally disributed
z=(nluck-luck) ./ sd;
```

Figure 8: Use the above functions for estimating luck numerically (nlucks) and estimating the error in luck (numerical standard deviation). The last lines optionally compute the exact values of luck (lucks) and standard deviations to compare with.