Lecture 1: Random and Quasirandom Simulation

The "Bartholomew Paradox" (a warm-up)

Given a family with two children, what's the chance that both are boys?

With no prior information, the answer is about 1/4.

What if we know that at least one of the children is a boy?

Then the chance (the conditional probability) that both are boys is about 1/3.

What if we know that at least one of the children is a boy named Bartholomew?

Then the chance that both are boys

is about 1/2.

? ! ?

To see what's going on here, let's look at an analogous problem about numbers.

If I am assigned a two-digit PIN (00-99), the probability that both digits are odd is exactly 1/4.

If I know that at least one of the digits is odd, then the probability that both digits are odd is exactly 25/75, or 1/3.

If I know that at least one of the digits is a 7, then the probability that both digits are odd is exactly 9/19, which is close to 1/2.

Check: the PIN is equally likely to be any of the nineteen combinations

70,<u>71</u>,72,<u>73</u>,74,<u>75</u>,76,<u>77</u>,78,<u>79</u>,
07,<u>17</u>,27,<u>37</u>,47,<u>57</u>,67,87, or <u>97</u>,
of which exactly nine (the ones underlined) have both digits odd.

Let's see how we could have used *Mathematica* to work this out for us.

```
In[1]:= Range[0, 9]

Out[1]= {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

In[2]:= Tuples[Range[0, 9], 2]

Out[2]= {{0, 0}, {0, 1}, {0, 2}, {0, 3}, {0, 4}, {0, 5}, {0, 6}, {0, 7}, {0, 8}, {0, 9}, {1, 0}, {1, 1},
  {1, 2}, {1, 3}, {1, 4}, {1, 5}, {1, 6}, {1, 7}, {1, 8}, {1, 9}, {2, 0}, {2, 1}, {2, 2},
  {2, 3}, {2, 4}, {2, 5}, {2, 6}, {2, 7}, {2, 8}, {2, 9}, {3, 0}, {3, 1}, {3, 2}, {3, 3},
  {3, 4}, {3, 5}, {3, 6}, {3, 7}, {3, 8}, {3, 9}, {4, 0}, {4, 1}, {4, 2}, {4, 3}, {4, 4},
  {4, 5}, {4, 6}, {4, 7}, {4, 8}, {4, 9}, {5, 0}, {5, 1}, {5, 2}, {5, 3}, {5, 4}, {5, 5},
  {5, 6}, {5, 7}, {5, 8}, {5, 9}, {6, 0}, {6, 1}, {6, 2}, {6, 3}, {6, 4}, {6, 5}, {6, 6},
  {6, 7}, {6, 8}, {6, 9}, {7, 0}, {7, 1}, {7, 2}, {7, 3}, {7, 4}, {7, 5}, {7, 6}, {7, 7},
  {7, 8}, {7, 9}, {8, 0}, {8, 1}, {8, 2}, {8, 3}, {8, 4}, {8, 5}, {8, 6}, {8, 7}, {8, 8},
  {8, 9}, {9, 0}, {9, 1}, {9, 2}, {9, 3}, {9, 4}, {9, 5}, {9, 6}, {9, 7}, {9, 8}, {9, 9}}

In[3]:= AllPairs = Tuples[Range[0, 9], 2]

Out[3]= {{0, 0}, {0, 1}, {0, 2}, {0, 3}, {0, 4}, {0, 5}, {0, 6}, {0, 7}, {0, 8}, {0, 9}, {1, 0}, {1, 1},
  {1, 2}, {1, 3}, {1, 4}, {1, 5}, {1, 6}, {1, 7}, {1, 8}, {1, 9}, {2, 0}, {2, 1}, {2, 2},
  {2, 3}, {2, 4}, {2, 5}, {2, 6}, {2, 7}, {2, 8}, {2, 9}, {3, 0}, {3, 1}, {3, 2}, {3, 3},
  {3, 4}, {3, 5}, {3, 6}, {3, 7}, {3, 8}, {3, 9}, {4, 0}, {4, 1}, {4, 2}, {4, 3}, {4, 4},
  {4, 5}, {4, 6}, {4, 7}, {4, 8}, {4, 9}, {5, 0}, {5, 1}, {5, 2}, {5, 3}, {5, 4}, {5, 5},
  {5, 6}, {5, 7}, {5, 8}, {5, 9}, {6, 0}, {6, 1}, {6, 2}, {6, 3}, {6, 4}, {6, 5}, {6, 6},
  {6, 7}, {6, 8}, {6, 9}, {7, 0}, {7, 1}, {7, 2}, {7, 3}, {7, 4}, {7, 5}, {7, 6}, {7, 7},
  {7, 8}, {7, 9}, {8, 0}, {8, 1}, {8, 2}, {8, 3}, {8, 4}, {8, 5}, {8, 6}, {8, 7}, {8, 8},
  {8, 9}, {9, 0}, {9, 1}, {9, 2}, {9, 3}, {9, 4}, {9, 5}, {9, 6}, {9, 7}, {9, 8}, {9, 9}}
```

```
In[4]:= Length[%]

Out[4]= 100

In[5]:= AllPairs[[99]]

Out[5]= {9, 8}

In[6]:= AtLeastOneSeven =
         Select[AllPairs, MemberQ[#, 7] &]

Out[6]= {{0, 7}, {1, 7}, {2, 7}, {3, 7}, {4, 7}, {5, 7}, {6, 7}, {7, 0}, {7, 1},
         {7, 2}, {7, 3}, {7, 4}, {7, 5}, {7, 6}, {7, 7}, {7, 8}, {7, 9}, {8, 7}, {9, 7}}

In[7]:= Length[%]

Out[7]= 19

In[8]:= Select[AtLeastOneSeven, OddQ[#[[1]]] && OddQ[#[[2]]] &]

Out[8]= {{1, 7}, {3, 7}, {5, 7}, {7, 1}, {7, 3}, {7, 5}, {7, 7}, {7, 9}, {9, 7}}

In[9]:= Length[%]

Out[9]= 9
```

If we limit ourselves to PINs in which the two digits are distinct (analogous to the assumption that the children in a two-child family are given different names), that changes the answer slightly: the probability is 8/18, which is still quite close to 1/2.

I won't discuss this paradox further in class, except to ask what I hope is a clarifying question: If you ring the doorbell and a boy answers, is that the same as learning that at least one child in the family is a boy?

I got the Bartholomew problem wrong the first time I heard it, years after I got my Ph.D.

MORAL: No matter how much probability theory you know, you're never immune to being led astray by your intuition!

Gambler's ruin

A gambler starts with \$1 in his pocket and makes a sequence of fair \$1 wagers at a casino, each time either gaining \$1 or losing a \$1, until he either goes down to \$0 (and is forced to stop) or gets up to \$3 (and leaves with the \$3).

Question 1: How likely is the gambler to leave with \$3? (Call this probability P.)

Question 2: How many wagers on average can the gambler expect to make before he leaves? (Call this average A.)

Note: We also call this a random walk problem; imagine a drunkard on the number line who starts at 1 and randomly staggers to the left or to the right repeatedly until he arrives either at 0 or at 3.

P = 1/4 (win, win)

+ 1/16 (win, lose, win, win)

+ 1/64 (win after 6 wagers)

+ ...

$= \frac{1/4}{1-1/4} = 1/3.$

```
In[10]:= Sum[1 / 4^n, {n, Infinity}]
```

Out[10]= $\frac{1}{3}$

$A = (1)(1/2)+(2)(1/4)+(3)(1/8)+$

$+ (4)(1/16)+\ldots$

$= ?$

$A = A(1/2)$ with

$A(x) = 1x+2x^2+3x^3+4x^4+\ldots$

$$= (x + x^2 + x^3 + x^4 + \ldots)$$
$$+ (x^2 + x^3 + x^4 + \ldots)$$
$$+ (x^3 + x^4 + \ldots)$$
$$+ (x^4 + \ldots)$$
$$+ \ldots$$
$$= \frac{x}{1-x} + \frac{x^2}{1-x} + \frac{x^3}{1-x} + \frac{x^4}{1-x} + \ldots$$
$$= (x + x^2 + x^3 + x^4 + \ldots) \,/\, (1-x)$$
$$= \left(\frac{x}{1-x}\right)/(1-\mathrm{x})$$
$$= \frac{x}{(1-x)^2}, \text{ so}$$

$$A = A(1/2) = \frac{1/2}{1/4} = 2.$$

```
Sum[n / 2^n, {n, Infinity}]
```

Out[11]= 2

In[12]:=
```
Sum[n / 2^n, {n, N}]
```

Out[12]= $2^{-N}\left(-2 + 2^{1+N} - N\right)$

In[13]:=
```
Limit[%, N → Infinity]
```

Out[13]= $\frac{\text{Log}[4]}{\text{Log}[2]}$

In[14]:= `Simplify[%]`

Out[14]= $\frac{\text{Log}[4]}{\text{Log}[2]}$

In[15]:= `FullSimplify[%]`

Out[15]= 2

Another way to calculate A is to recognize it as the average value of a geometric random variable with parameter 1/2.

(Regardless of whether the gambler has \$1 or \$2, his chance of leaving the game after his next wager is 1/2.)

So A is also the expected number of times you have to toss a fair coin until it comes up Heads.

If you don't remember why this should be 2, here's an argument:
If you toss the coin until it comes up Heads, then with probability 1/2 the first toss comes up Heads and the total number of tosses required is 1, while with probability 1/2 the

first toss comes up Tails and the total number of tosses required is 1+A on average. So A = (1/2)(1) + (1/2)(1+A).

```
In[16]:= Solve[A == 1 / 2 + (1 + A) / 2, A]
Out[16]= {{A → 2}}
```

(Note that strictly speaking this argument does not rule out the possibility that A = ∞; for more on this, see below.)

Harmonic functions

We can also use linear equations to solve for P and A.

Recall that P = the probability of ending up at 3 if you start at 1.
Let Q = the probability of ending up at 3 if you start at 2.
(Note:
the probability of ending up at 3 if you start at 0 is 0, and

the probability of ending up at 3 if you start at 3 is 1.)

I claim that

(*) $P = \frac{0+Q}{2}$ and $Q = \frac{P+1}{2}$.

That is, I claim that the function $h(x)$ defined by

$$h(0) = 0,$$
$$h(1) = P,$$
$$h(2) = Q,$$
$$h(3) = 1$$

satisfies

$$h(x) = \frac{h(x-1)+h(x+1)}{2} \text{ for } x = 1,2.$$

Such a function is called harmonic (we'll see the definition of this term in a future lecture).

```
In[17]:= Solve[{P == (0 + Q) / 2, Q == (P + 1) / 2}, {P, Q}]
```

Out[17]= $\left\{\left\{P \to \frac{1}{3}, Q \to \frac{2}{3}\right\}\right\}$

Likewise, recalling that A = the expected number of steps the walker takes starting from 1, let B = the expected number of steps the walker takes starting from 2.

Then

(**) $\quad A = 1 + \frac{0+B}{2}$ and $B = 1 + \frac{A+0}{2}$.

```
In[18]:= Solve[{A == 1 + (0 + B) / 2, B == 1 + (A + 0) / 2}, {A, B}]
```

Out[18]= $\{\{A \to 2, B \to 2\}\}$

Caveat: This argument only shows that if A and B are finite, then they must both equal 2; it does not rule out the possibility that the expected number of steps the walker takes is infinite starting from both 1 and 2. Indeed, we will see later that for random walk on {0,1,2,...}, a walker who starts at 1 will eventually hit 0 with probability 1, but that the time it takes for this to happen has infinite expected value.

(If you've never seen a random variable with infinite expected value, consider 2^X, where X is a geometric random variable with parameter $\frac{1}{2}$:

$$\text{Exp}\,(2^X) = (2)(\tfrac{1}{2}) + (4)(\tfrac{1}{4}) + (8)(\tfrac{1}{8}) + (16)(\tfrac{1}{16}) + \ldots$$

which is ∞.)

Fortunately for us, it's not hard to

show that A is finite.

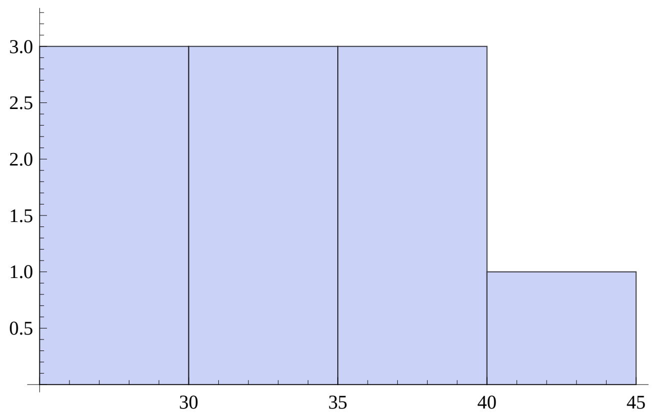
Random simulation

One way to see what a random system does is to simulate it on a computer using a pseudorandom number generator; this is an algorithm designed to produce output that passes as many statistical tests for randomness as possible.

```
In[21]:= RandomInteger[]
         (* generate a random bit *)

Out[21]= 0

In[25]:= Random[Integer] (* generate a random bit *)

Out[25]= 1

In[26]:= Table[Random[Integer], {n, 10}]

Out[26]= {1, 1, 0, 0, 1, 0, 1, 1, 1, 0}

In[28]:= Wager[n_] := If[Random[Integer] == 0, n - 1, n + 1]

In[32]:= Wager[1]

Out[32]= 0
```

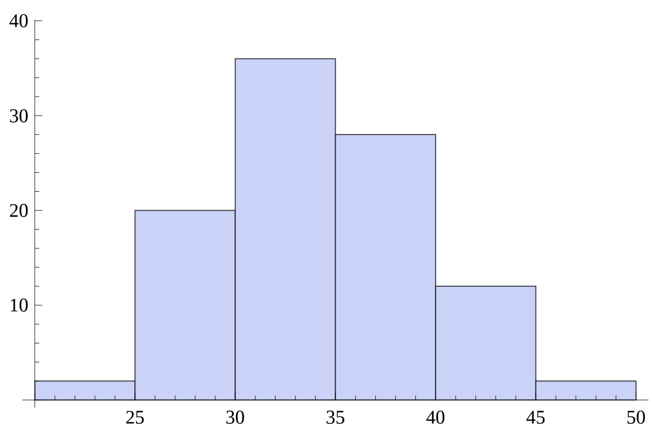
```
Wager[n_] := n + (-1)^Random[Integer]
(* another implementation *)

In[33]:= Table[Wager[1], {k, 10}]

Out[33]= {0, 2, 0, 2, 0, 2, 0, 2, 0, 0}

In[34]:= Ruin[n_] := If[(n ≤ 0 || n ≥ 3), n, Ruin[Wager[n]]]

In[35]:= Table[Ruin[1], {n, 20}]

Out[35]= {0, 3, 0, 3, 0, 0, 0, 3, 0, 3, 0, 0, 0, 0, 0, 3, 0, 0, 0, 3}

In[37]:= ManyRuins[n_] := Sum[Ruin[1] / 3, {k, n}]

In[41]:= ManyRuins[100]

Out[41]= 43

In[42]:= Table[ManyRuins[100], {n, 10}]

Out[42]= {31, 40, 28, 34, 35, 29, 36, 36, 32, 29}

In[43]:= Histogram[%]
```

Out[43]=



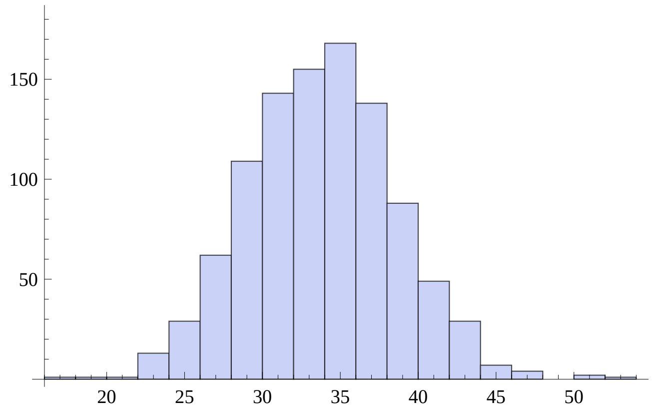

```
In[44]:= Histogram[Table[ManyRuins[100], {n, 100}]]
```

Out[44]=



In[45]:=
```
Histogram[Table[ManyRuins[100], {n, 1000}]]
```

Out[45]=



Let's manually re-enter the code.

```
Histogram[Table[ManyRuins[100], {n, 1000}]]
```

Note how *Mathematica* helpfully matches brackets as you type.

In[46]:=
```
HundredValues = Table[ManyRuins[100] / 100, {n, 100}];
```

In[47]:=
```
Mean[HundredValues]
```

Out[47]= $\frac{1673}{5000}$

In[48]:=
```
N[%]
```

Out[48]= 0.3346

In[49]:=
```
StandardDeviation[HundredValues]
```

Out[49]= $\frac{\sqrt{\frac{54121}{11}}}{1500}$

In[50]:=
```
N[%]
```

Out[50]= 0.0467622

Drawing a sample of size 100 will usually lead us to estimate P as lying in some interval whose endpoints both lie between .29 and .37, but it does not suffice to give us an estimate of P with two significant figures.

To see why, note that the output of ManyRuins[100] is governed by the distribution Binomial(100,P).

Recall that a Binomial(n,P) random variable is a sum of n independent Bernoulli(P) random variables, each of which has variance P(1-P), for a total variance of nP(1-P), and standard deviation of $\sqrt{n\,P\,(1-P)}$. So Binomial(n,P)/n has standard deviation $\sqrt{n\,P(1-P)\,/\,n} = \sqrt{P(1-P)/n}$.

So, the error for this kind of simulation-based estimation goes like $C/\sqrt{n}$, where C is some constant and n is the size of our sample.

In our case:

In[51]:=
```
N[Sqrt[(1 / 3) (1 - 1 / 3) / 100]]
```

Out[51]=
```
0.0471405
```

That's good enough to estimate P to one digit, but not to two digits.

To decrease the error by a factor of 10, we need to increase n by a factor of 100.

E.g., to estimate P to 3 significant figures, we would need $n \sim 10^6$, and to estimate P to 6 significant figures, we would need $n \sim 10^{12}$ (which would be computationally infeasible in most cases).

We could also estimate A by simulation. Here again, we need about 10^{2k} trials to estimate A to k significant figures.

Probabilities, expected values, and integrals

If U is a U(0,1) random variable (a real number uniformly distributed in the interval [0,1]), the bits of its binary expansion can be used as a sequence of independent, identically distributed ("i.i.d.") bits: the first bit is equally likely to be a 0 or a 1, the next bit is equally likely to be a 0 or a 1 regardless of what the first bit was, etc.

Take 0 = Heads = Left,

1 = Tails = Right.

If $U = .0...$ (that is, if the first bit is 0), the walk ends at 0.

If $U = .11...$, the walk ends at 3.

If $U = .100...$, the walk ends at 0.

If $U = .1011...$, the walk ends at 3.

If U=.10100..., the walk ends at 0.
I.e., the walk ends at 0 if $U < \frac{2}{3}$ and the walk ends at 3 if $U > \frac{2}{3}$.
("What if $U = \frac{2}{3}$?"
The probability of that happening is 0.
This is an aspect of probability theory that gives novices trouble: you have to learn how to pay the right sort of attention to events of probability 0.

While we're speaking of such things, note that there's a slight mis-match between the discrete world of sequences of H's and T's and the continuous world of real numbers; e.g., the bit-strings THHH... (infinitely many H's) and HTTT... (infinitely many T's) both correspond to the fraction $\frac{1}{2}$ (though only the former is the standard binary representation of $\frac{1}{2}$).

But this problem only affects rational numbers of the form $\frac{k}{2^n}$, of which there are only countably many in [0,1], so this won't cause problems for us. I'll say more about this in the next lecture.)

So P (the probability that the walker reaches 3 before reaching 0) is equal to the integral $\int_0^1 1_{[2/3,1]}(x)\, d\,x$, where $1_{[2/3,1]}(x) = 1$

if x is in [2/3, 1] and = 0 otherwise. This equals 1/3.

What about A? Write $A = \text{Exp}(W)$, where the random variable W is the number of steps the walker takes before reaching 0 or 3.

If we use U to generate coin-flips as before, then U determines W:

If $0 < U < \frac{1}{2}$, then $W = 1$;

if $\frac{3}{4} < U < 1$, then $W = 2$;

if $\frac{1}{2} < U < \frac{5}{8}$, then $W = 3$;

if $\frac{11}{16} < U < \frac{3}{4}$, then $W = 4$;

if $\frac{5}{8} < U < \frac{21}{32}$, then $W = 5$; etc.

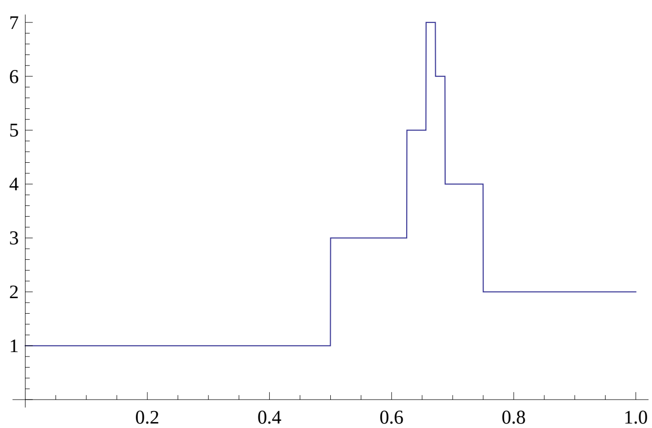
That is, $W = f(U)$, where

$f(x)$ = the first position at which the binary expansion of x differs from the binary expansion of $\frac{2}{3}$.

So $A = \text{Exp}(W) = \int_0^1 f(x)\, dx$.

In[52]:=
```
WPlot = Plot[Piecewise[{
    {1, 0 < x && x < 1 / 2},
    {2, 3 / 4 < x && x < 1},
    {3, 1 / 2 < x && x < 5 / 8},
    {4, 11 / 16 < x && x < 3 / 4},
    {5, 5 / 8 < x && x < 21 / 32},
    {6, 43 / 64 < x && x < 11 / 16},
    {7, 21 / 32 < x && x < 43 / 64}}], {x, 0, 1}, Exclusions → None, AxesOrigin → {0, 0}]
```

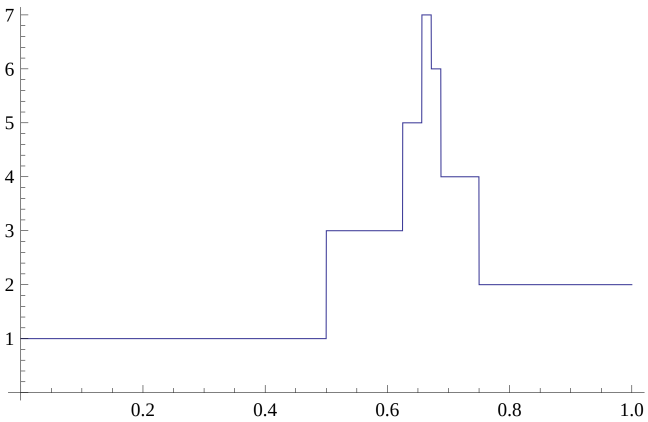
Out[52]=



(Of course this is just an approximation to the graph of $f(x)$, since $f(x) \to \infty$ as $x \to \frac{2}{3}$.)
Note that the x- and y-axes are not drawn to the same scale.
If we compute the area under the curve in the usual way (splitting the region along vertical lines into rectangles that touch the x-axis), we get $(\frac{1}{2})(1)+(\frac{1}{4})(2)+(\frac{1}{8})(3)+\ldots = 2$ as before.

In[53]:= `WPlot`

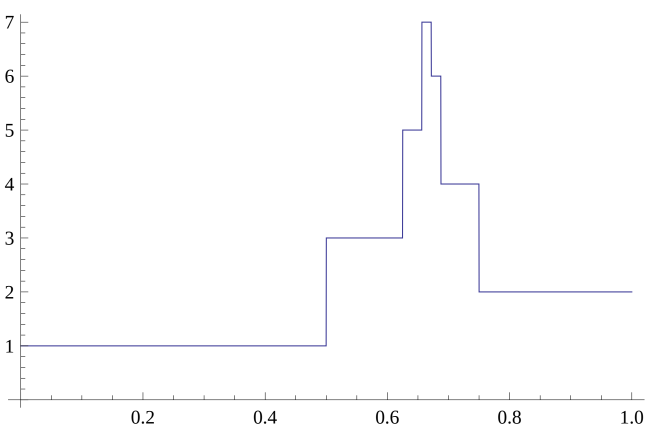
Out[53]=



But if we split the region along horizontal lines into maximally wide rectangles, we get $(1)(1)+(\frac{1}{2})(1)+(\frac{1}{4})(1)+\ldots = 2$.

In[54]:= `WPlot`

Out[54]=



This second calculation of the area is a geometrical version of one of the slickest tricks I know for computing the expected value of a random variable that takes on only nonnegative integer values:

$$\mathrm{Exp}(X) = \sum_{n=1}^{\infty} \mathrm{P}(X \geq n)$$

Proof: $\mathrm{Exp}(X) =$

$$1\mathrm{P}(X=1)+2\mathrm{P}(X=2)+3\mathrm{P}(X=3)+\ldots$$

$$\begin{aligned} = \quad & \mathrm{P}(X=1)+\mathrm{P}(X=2)+\mathrm{P}(X=3)+\ldots \\ & \qquad\qquad +\mathrm{P}(X=2)+\mathrm{P}(X=3)+\ldots \\ & \qquad\qquad\qquad\qquad +\mathrm{P}(X=3)+\ldots \end{aligned}$$

$+ \ldots$

$= P(X \geq 1) + P(X \geq 2) + P(X \geq 3) + \ldots$

Reduction of variance and quasirandomness

Recall that $A = \int_0^1 f(x)\, dx$ for a certain piecewise-constant, unbounded function $f(x)$ on [0,1], namely $f(x)$ = the 1st position at which the binary representations of x and 2/3 = .101010... disagree.

Likewise $P = \int_0^1 1_{[2/3,1]}(x)\, d\, x$, where $1_{[2/3,1]}(x)$ is 1 if x ϵ [2/3, 1] and is 0 otherwise.

Our simulation-based methods of estimating P and A are tantamount to estimating these integrals by sampling the functions $1_{[2/3,1]}(x)$ and $f(x)$ at n randomly chosen points X_1, X_2, ..., X_n in [0,1] and taking the average of the n values:

$$\int_0^1 f(x)\, dx \sim \frac{1}{n}[f(X_1)+f(X_2)+\ldots+f(X_n)].$$

This technique is called Monte Carlo integration, or MC integration for short. (Its serious applications come in higher-dimensional problems.)

Intuition: If X is chosen uniformly from [0,1], then the expected value of $f(X)$ is $\int_0^1 f(x)\, dx$. The same is true for each of X_1, X_2, ..., X_n, and if we average these estimates of the integral, then by the law of large numbers, our average will be a better and better estimate of the true integral as n goes to infinity.

If σ is the standard deviation of the random variable $f(X)$ (where X is uniform on [0,1]), then the average

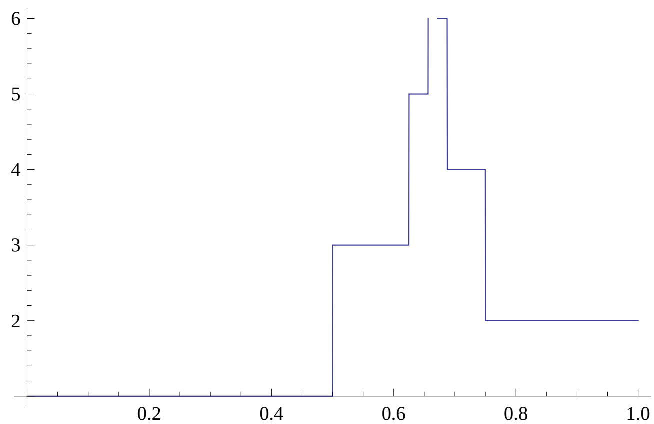
(1) $\quad \frac{1}{n}[f(X_1)+f(X_2)+...+f(X_n)]$

has standard deviation $\sigma/\sqrt{n}$.

This is not so good: e.g., to increase the accuracy of your estimate of the integral $\int_0^1 f(x)\,dx$ by a factor of 10, you have to increase the number of sample-points 100-fold.

```
In[55]:= f[x_] := Which[x < 1 / 2, 1, x > 3 / 4, 2, True, 2 + f[4 x - 2]]

In[56]:= Plot[f[x], {x, 0, 1}]
```

Out[56]=




```
In[57]:= Integrate[f[x], {x, 0, 1}]
```

Out[57]= $\int_0^1 \text{Which}\left[x < \frac{1}{2}, 1, x > \frac{3}{4}, 2, \text{True}, 2 + f[4x - 2]\right] dx$

```
In[59]:= Table[f[Random[]], {n, 10}]

Out[59]= {2, 1, 1, 1, 2, 1, 2, 3, 1, 2}

In[60]:= Mean[%]
```

Out[60]= $\frac{8}{5}$

```
In[61]:= MC[n_] := Mean[Table[f[Random[]], {k, n}]]

In[62]:= N[MC[100]]

Out[62]= 2.07
```

```
In[63]:= N[MC[10^4]]
Out[63]= 1.9932
In[64]:= N[MC[10^6]]
Out[64]= 2.00207
```

A much better way to estimate an integral in one dimension is to choose *n* evenly-spaced random points in [0,1].

"How can they be evenly-spaced if they're random?"

If you choose a random number *U* in [0,1] (under the uniform distribution) and take

$U_1 = U$,

$U_2 = U + 1/n \pmod 1$,

$U_3 = U + 2/n \pmod 1$,

...

$U_n = U + (n\text{-}1)/n \pmod 1$

then all of the random variables U_1, U_2, ..., U_n are (individually) uniformly distributed on [0,1], so the derived random variables $f(U_1)$, $f(U_2)$, ..., $f(U_n)$ all have expected value equal to $\int_0^1 f(x)\, dx$, and so the average

(2) $\frac{1}{n}[f(U_1)+f(U_2)+...+f(U_n)]$

is also a random variable with expected value $\int_0^1 f(x)\, dx$, but this estimate of the integral will usually have much less error than (1).

A consequence of this is that if you choose V uniformly at random in $[0, 1/n]$ (not $[0, 1]$) and then take

$$V_1 = V,$$

$$V_2 = V + 1/n\,,$$

$$V_3 = V + 2/n,$$

...

$$V_n = V + (n-1)/n$$

then

(3) $\frac{1}{n}[f(V_1)+f(V_2)+\ldots+f(V_n)]$

has expected value $\int_0^1 f(x)\,dx$, and because the variance of (3) is low, (3) tends to be very close to the integral (compare with Riemann integration).

```
In[65]:= Range[10]
Out[65]= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

In[66]:= Random[]
Out[66]= 0.969234

In[67]:= (Range[10] - Random[]) / 10
Out[67]= {0.0359418, 0.135942, 0.235942, 0.335942,
          0.435942, 0.535942, 0.635942, 0.735942, 0.835942, 0.935942}

In[68]:= Map[f, (Range[10] - Random[]) / 10]
Out[68]= {1, 1, 1, 1, 1, 3, 5, 2, 2, 2}
```

```
In[69]:= Mean[Map[f, (Range[10] - Random[]) / 10]]
```

Out[69]= $\frac{9}{5}$

```
In[70]:= QMC[n_] := Mean[Map[f, (Range[n] - Random[]) / n]]

In[71]:= N[QMC[10^2]]

Out[71]= 1.97

In[72]:= N[QMC[10^4]]

Out[72]= 2.0001

In[73]:= N[QMC[10^6]]

Out[73]= 2.

In[74]:= % - 2
```

Out[74]= $-3. \times 10^{-6}$

This is an example of quasi-Monte Carlo integration. (Actually, the term "QMC" is sometimes applied to only fully deterministic analogues of MC; e.g., taking $V_i = i/n$ for $i = 1,2,\ldots,n$. A scheme like the above, which still uses some randomness, would be called randomized quasi-Monte Carlo, or RQMC integration.)

A crude slogan we might consider is "To reduce variance, reduce randomness." But how true is this? And how much randomness can we take away from a stochastic process before we destroy the things we're trying to measure, like A and P?

Can we take away ALL of the randomness and construct a non-stochastic process that nonetheless gives useful information about the original stochastic process?

These are major themes of my research, and they're themes that I'll touch upon in the lectures and the homework, and that you may wish to explore in your final project.

At the same time, I want you to end up with a basic knowledge of some of the sorts of stochastic processes I and others study: Markov chains, Poisson processes, Brownian motion, etc.

Homework for next Monday

1. Use *Mathematica* to find the 100th digit of π (counting the initial 3 as a digit) and the 100th digit of π after the decimal point. (To get a free copy of *Mathematica* for your PC, contact me.)

2. Suppose X and Y are Bernoulli random variables with $P(X=1) = P(Y=1) = .4$. Note that this description does not specify the joint distribution of X and Y. (Let $s = P(X=1,Y=1)$, $t = P(X=1,Y=0)$, $u = P(X=0,Y=1)$, and $v = P(X=0, Y=0)$; all we know is that $s + t + u + v = 1$ with $s + t = P(X=1) = .4$ and $s + u = P(Y=1) = .4$.) How big might the variance of $X+Y$ be? How small might the variance of $X+Y$ be? What would the variance be if X and

Y are independent? (As a warm-up, first do the simplified version of the problem in which .4 is replaced by .5.)