The Truel Problem

Imagine that you have a three-way duel (or "truel"). The three duelists are named A, B, and C and have accuracy probabilities of a, b, and c with $0 < a < b < c < 1$. The rules of the duel are that the duelists fire *sequentially* in the starting order A, B, C (so the least accurate person fires first, etc., and continuing to fire in that order) and the last person standing wins the duel. If you are duelist A, what strategy should you use to maximize your chance of winning?

Common Strategy Choices

If you ask most people what A's strategy should be when all 3 duelists are alive, there are two common answers:

1) Always fire at the most accurate duelist (C).
2) Fire randomly (since you are the least accurate it doesn't matter).

To this we'll add what seems like a bad option:

3) Fire your shot into the ground, wasting it.

To analyze the effectiveness of these strategies we will use a technique from probability theory known as Markov chains.

What is a Markov Chain?

A Markov chain is a way to model a system in which:

1) The system consists of a number of states, and the system can only be in one state at any time. This is often viewed as the system moving in discrete steps from one state to another.

3) The probability of moving from a state i to a state j doesn't depend on how the system got to state i - Markov chains have no "long term memory".

3) The probability that the system will move between any two given states is known. These probabilities are usually given as a matrix, where row i of the matrix is the list of probabilities of moving from state i to any other state.

Once the matrix for a Markov chain is known it can be used to find all sort of probabilities, including the probability that the system will be in any given state after a given number of steps, the number of steps it will take on average to move from one given state to another, and the probability that the system will end up in a given state in the long run.

A simple example of a Markov chain is a coin flipping game. Let's say we have a coin which has a 45% chance of coming up Heads and a 55% chance of coming up tails. I win the game if the coin comes up Heads twice in a row and you will win if it comes up Tails twice in a row. In terms of playing the game since we are only interested in getting two heads/tails we only need to track what the coin was on the last flip. This means the game has

5 states:

State 1: The start (we haven't flipped the coin yet)
State 2: Heads was the previous flip.
State 3: Tails was the previous flip.
State 4: I won the game.
State 5: You won the game.

The probability table for this process looks like this:

	Start	Heads	Tails	I win	You win
Start	0	0.45	0.55	0	0
Heads	0	0	0.55	0.45	0
Tails	0	0.45	0	0	0.55
I win	0	0	0	1	0
You win	0	0	0	0	1

Note that the "I win"/ "You Win" rows are all 0's except for the single 1 - once the game enters one of those states it is essentially over and can't move to another state.

Entering this as a matrix we have:

$$\textbf{flipmatrix} = \begin{pmatrix} 0 & 0.45 & 0.55 & 0 & 0 \\ 0 & 0 & 0.55 & 0.45 & 0 \\ 0 & 0.45 & 0 & 0 & 0.55 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix};$$

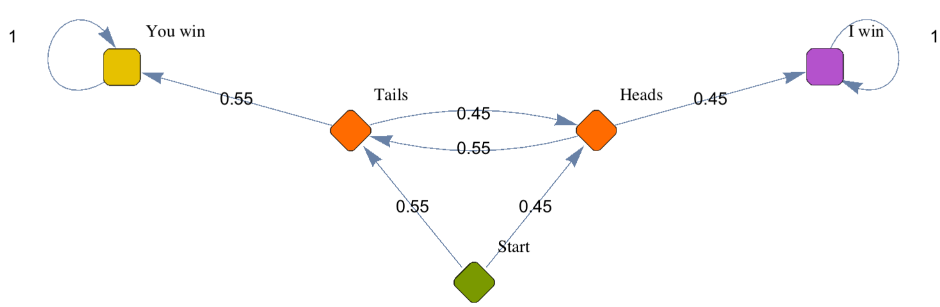
In *Mathematica* we can now describe the game using DiscreteMarkovProcess, letting the computer know the game starts in state 1:

```
coinprocess = DiscreteMarkovProcess[1, flipmatrix]
```

```
DiscreteMarkovProcess[1, {{0, 0.45, 0.55, 0, 0},
  {0, 0, 0.55, 0.45, 0}, {0, 0.45, 0, 0, 0.55}, {0, 0, 0, 1, 0}, {0, 0, 0, 0, 1}}]
```

Before we do any calculations we can have *Mathematica* show us the graph of how the system moves from state to state:

```
Graph[ coinprocess, VertexLabels → Thread[Range[5] → coinstates],
 ImageSize → 500, EdgeLabels → Getedges[flipmatrix], VertexSize → Medium]
```



Now we can see the chance that I will win (which is state 4) in 3 flips or less:

```
PDF[coinprocess[3], 4]
```

```
0.313875
```

The chance that I will win in any number of flips:

```
PDF[coinprocess[∞], 4]
```

```
0.41711
```

We could even do this for an arbitrary “Heads” probability. In this case the probability matrix looks like this:

```
Clear[p]
```

$$\texttt{flipmatrix2} = \begin{pmatrix} 0 & p & 1-p & 0 & 0 \\ 0 & 0 & 1-p & p & 0 \\ 0 & p & 0 & 0 & 1-p \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix};$$

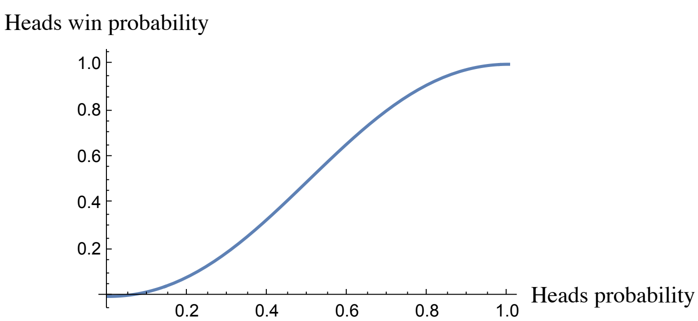
We can now get an exact formula for what my chances of winning are:

```
newcoinprocess = DiscreteMarkovProcess[1, flipmatrix2];
headswins = Simplify[PDF[ newcoinprocess[∞], 4]]
```

$$-\frac{(-2+p)\,p^2}{1-p+p^2}$$

Graphing this as the probability p goes from 0 to 1 looks like:

```
Plot[ headswins, {p, 0, 1}, AxesLabel → (Style[Text[#], FontSize → 12] & /@
    {"Heads probability", "Heads win probability"}), LabelStyle → Black]
```



If we want to figure out how I to need to bias the coin so I win 3 times out of 4 we can use Reduce:

```
Reduce[ headswins == 3 / 4 ∧ 0 ≤ p ≤ 1, p, Reals]
```

```
p == Root[3 - 3 #1 - 5 #1^2 + 4 #1^3 &, 2]
```

```
N[%, 10]
```

```
p == 0.6581690825
```

So if I can bias the coin to come up heads 65.82% of the time I’ll have more than 75% chance of winning the game.

The Truel Problem as a Markov Chain

We can set up the Truel problem as a Markov chain in such a way as to allow us to investigate the three strategies for duelist A. To define the Markov chain we need to define the states that the duel can be in and the chance moving from one state to any other.

The states are simply the different combinations of who is still alive and whose turn it is to fire:

State 1: ABC alive, A's turn to fire.
State 2: ABC alive, B's turn to fire.
State 3: ABC alive, C's turn to fire.
State 4: AB alive, A's turn to fire.
State 5: AB alive, B's turn to fire.
State 6: AC alive, A's turn to fire.
State 7: AC alive, C's turn to fire.
State 8: BC alive, B's turn to fire.
State 9: BC alive, C's turn to fire.
State 10: A wins
State 11: B wins
State 12: C wins

Although this system has a lot of states, fortunately most of the states aren't directly connected (that is it is not possible to move from some states to other ones) so it is not quite as complicated as it might seem. For example none of the states 1-3 directly connect to the states 10-12 as you can't go from all 3 people alive to just 1 person alive in a single step.

Next we have to find the probabilities of moving from one state to another. We know the accuracy probabilities for the shooters are a, b, and c (with $0 < a < b < c < 1$). Define pab to be the chance that A shoots at B when 3 people are alive, pac the chance A shoots at C when 3 people are alive, and pag the chance that A shoots at the ground when 3 people are alive (so $0 \leq pab, pac, pag \leq 1$ and $pab+pac+pag = 1$). Let pba, pbc, pca, and pcb be defined the same way (we will assume that as B and C are not the worst shots they won't waste their shots, so there is no need to consider terms like pbg and pcg). As these are also probabilities we know $0 \leq pba, pbc, pca, pcb \leq 1$, $pba + pbc = 1$, and $pca + pcb = 1$. It may seem like extra work to use general probabilities rather than start with each strategy, but this will let us find the matrix for the Markov chain just once and then treat each strategy as a special case of the general matrix.

Working with the probabilities to move from each given state, we have:

State 1 only directly connects to states 2, 5, and 7 (all other probabiltities must be 0)

The chance from moving from state 1 to state 2 is:

$$p(1 \to 2) = p(\text{ground shot}) + p(A \text{ chose } B \text{ and missed}) + p(A \text{ chose } C \text{ and missed}) =$$
$$pag + (1 - a)\, pab + (1 - a)\, pac$$

The chance from moving from state 1 to state 5 is:

$$p(1 \to 5) = p(A \text{ chose } C \text{ and hit}) = pac \times a = a\, pac$$

The chance of moving from state 1 to state 7 is

$$p(1 \to 7) = p(A \text{ chose } B \text{ and hit}) = pab \times a = a\, pab$$

State 2 only directly connects to states 3, 4, and 9 (all other probabiltities must be 0); those probabilities are

$$p(2 \to 3) = p(B \text{ chose } A \text{ and missed}) + p(B \text{ chose } C \text{ and missed}) = (1-b)\, pba + (1-b)\, pbc$$
$$p(2 \to 4) = p(B \text{ chose } C \text{ and hit}) = b\, pbc$$
$$p(2 \to 9) = p(B \text{ chose } A \text{ and hit}) = b\, pba$$

State 3 only directly connects to states 1, 6, and 8 (all other probabiltities must be 0); those probabilities are

$$p(3 \to 1) = p(C \text{ chose } A \text{ and missed}) + p(C \text{ chose } B \text{ and missed}) = (1-c)\, pca + (1-c)\, pcb$$
$$p(3 \to 6) = p(C \text{ chose } B \text{ and hit}) = c\, pcb$$
$$p(3 \to 8) = p(C \text{ chose } A \text{ and hit}) = c\, pca$$

State 4 only directly connects to states 5 and 10 (all other probabiltities must be 0). As this state is essentially a 2 person duel there is no advantage to A shooting the ground and so no real choice for what to shoot at, and the two probabilities are

$$p(4 \to 5) = p(A \text{ shot at } B \text{ and missed}) = 1-a$$
$$p(4 \to 10) = p\,(A \text{ shot at } B \text{ and hit}) = a$$

Likewise state 5 only directly connects to states 4 and 11 (all other probabilities must be 0), and these probabilities are

$$p(5 \to 4) = p(B \text{ shot at } A \text{ and missed}) = 1-b$$
$$p(5 \to 11) = p(B \text{ shot at } A \text{ and hit}) = b$$

Similarly state 6 only directly connects to states 7 and 10 (all other probabilities must be 0), and these probabilities are

$$p(6 \to 7) = p(A \text{ shot at } C \text{ and missed}) = 1-a$$
$$p(6 \to 10) = p\,(A \text{ shot at } C \text{ and hit}) = a$$

Likewise state 7 only directly connects to states 6 and 12 (all other probabilities must be 0), and these probabilities are

$$p(7 \to 6) = p(C \text{ shot at } A \text{ and missed}) = 1-c$$
$$p(7 \to 12) = p(C \text{ shot at } A \text{ and hit}) = c$$

Similarly state 8 only directly connects to states 9 and 11 (all other probabilities must be 0), and these probabilities are

$$p(8 \to 9) = p(B \text{ shot at } C \text{ and missed}) = 1-b$$
$$p(8 \to 11) = p(B \text{ shot at } C \text{ and hit}) = b$$

Likewise state 9 only directly connects to states 8 and 12 (all other probabilities must be 0), and these probabilities are

$$p(9 \to 8) = p(C \text{ shot at } B \text{ and missed}) = 1-c$$
$$p(9 \to 12) = p(C \text{ shot at ABand hit}) = c$$

States 10, 11, and 12 are the “win” states, so they each directly connect only to themselves with probability 1 as they are possible ends for the duel (all other probabilities are 0).

As most of the probabilties are 0 it is easier to define the matrix using a command called SparseArray (in which you only have to say which entries are not 0):

```
truelmatrix =
  Normal[SparseArray[ { {1, 2} → pag + (1 - a) pab + (1 - a) pac, {1, 5} → a pac,
     {1, 7} → a pab, {2, 3} → (1 - b) pba + (1 - b) pbc, {2, 4} → b pbc,
     {2, 9} → b pba, {3, 1} → (1 - c) pca + (1 - c) pcb, {3, 6} → c pcb, {3, 8} → c pca,
     {4, 5} → 1 - a, {4, 10} → a, {5, 4} → 1 - b, {5, 11} → b, {6, 7} → 1 - a,
     {6, 10} → a, {7, 6} → 1 - c, {7, 12} → c, {8, 9} → 1 - b, {8, 11} → b,
     {9, 8} → 1 - c, {9, 12} → c, {10, 10} → 1, {11, 11} → 1, {12, 12} → 1}]];
MatrixForm[truelmatrix]
```

0	(1 − a) pab + (1 − a) pac + pag	0	0	a pac	(
0	0	(1 − b) pba + (1 − b) pbc	b pbc	0	(
(1 − c) pca + (1 − c) pcb	0	0	0	0	c p
0	0	0	0	1 − a	(
0	0	0	1 − b	0	(
0	0	0	0	0	(
0	0	0	0	0	1 -
0	0	0	0	0	(
0	0	0	0	0	(
0	0	0	0	0	(
0	0	0	0	0	(
0	0	0	0	0	(

Since A shoots first the initial state for the Markov chain will be state 1. Once we implement the different strategies by choosing values for the probabilites *pab*, etc. we can find the chance that A (or anyone else) will win the duel by looking at the probability we end up in state 10 (or state 11 for B to win, or state 12 for C to win).

Implementing the Three Strategies

The three stratgies for duelist A are to shoot the best duelist (C), shoot B or C at random, or waste the shot in the ground. Assuming that B and C will always shoot at each other when all three people are alive lets us set specific values for many of the probabilities:

The "shoot the best duelist" strategy corresponds to
$pac = 1,\ pab = 0,\ pag = 0,\ pba = 0,\ pbc = 1,\ pcb = 1,\ pca = 0$:

```
shootatbest = {pac → 1, pab → 0, pag → 0, pba → 0, pbc → 1, pcb → 1, pca → 0};
shootatbestprocess = DiscreteMarkovProcess[1, truelmatrix /. shootatbest];
```

The "shoot randomly" strategy for A corresponds to
$pac = \frac{1}{2},\ pab = \frac{1}{2},\ pag = 0,\ pba = 0,\ pbc = 1,\ pcb = 1,\ pca = 0$:

```
shootatrandom =
  {pac → 1 / 2, pab → 1 / 2, pag → 0, pba → 0, pbc → 1, pcb → 1, pca → 0};
shootatrandomprocess = DiscreteMarkovProcess[1, truelmatrix /. shootatrandom];
```

The "waste your shot" strategy corresponds to $pab = pac = 0,\ pag = 1,\ pba = 0,\ pbc = 1,\ pcb = 1,\ pca = 0$:

```
shootatground = { pac → 0, pab → 0, pag → 1, pba → 0, pbc → 1, pcb → 1, pca → 0};
shootatgroundprocess = DiscreteMarkovProcess[1, truelmatrix /. shootatground];
```

Assuming that A uses the "shoot at the best duelist" strategy his chance of winning is given by

```
bestawin = Together[ PDF[shootatbestprocess[∞], 10]]
```

$$-\left(\left(a\left(-a^2-a\,b+2\,a^2\,b-2\,a\,c+2\,a^2\,c-2\,b\,c+6\,a\,b\,c-4\,a^2\,b\,c+b^2\,c-2\,a\,b^2\,c+a^2\,b^2\,c\right)\right)\Big/\left((-a-b+a\,b)\,(-a-c+a\,c)\,(a+b-a\,b+c-a\,c-b\,c+a\,b\,c)\right)\right)$$

In the case A is using rhe "shoot at random" strategy the chance he wins is

```
randomawin = Together[ PDF[shootatrandomprocess[∞], 10]]
```

$$\left(2 a^3 + 3 a^2 b - 4 a^3 b + 3 a^2 c - 4 a^3 c + 4 a b c - 12 a^2 b c + 8 a^3 b c - 2 a b^2 c + 4 a^2 b^2 c - 2 a^3 b^2 c\right) / (2 (-a - b + a b) (-a - c + a c) (a + b - a b + c - a c - b c + a b c))$$

If A is using the "waste your shot"/ "shoot the ground" strategy his chance of winning is

```
groundawin = Together[ PDF[shootatgroundprocess[∞], 10]]
```

$$-\left(\left(a \left(a b + a c + 2 b c - 3 a b c - b^2 c + a b^2 c\right)\right) / ((-a - b + a b) (-a - c + a c) (-b - c + b c))\right)$$

So the question of which strategy to use is which of these 3 quantities is largest - for certain values of a, b, and c different strategies might work better. For example if $a = .1$, $b = .2$, and $c = .9$ these numbers are

```
{bestawin, randomawin, groundawin} /. {a → .1, b → .2, c → .9}
{0.176795, 0.161993, 0.163641}
```

and therefore it would better for A to use the "shoot at best" strategy. But if $a = .4$, $b = .8$, $c = .9$ we get

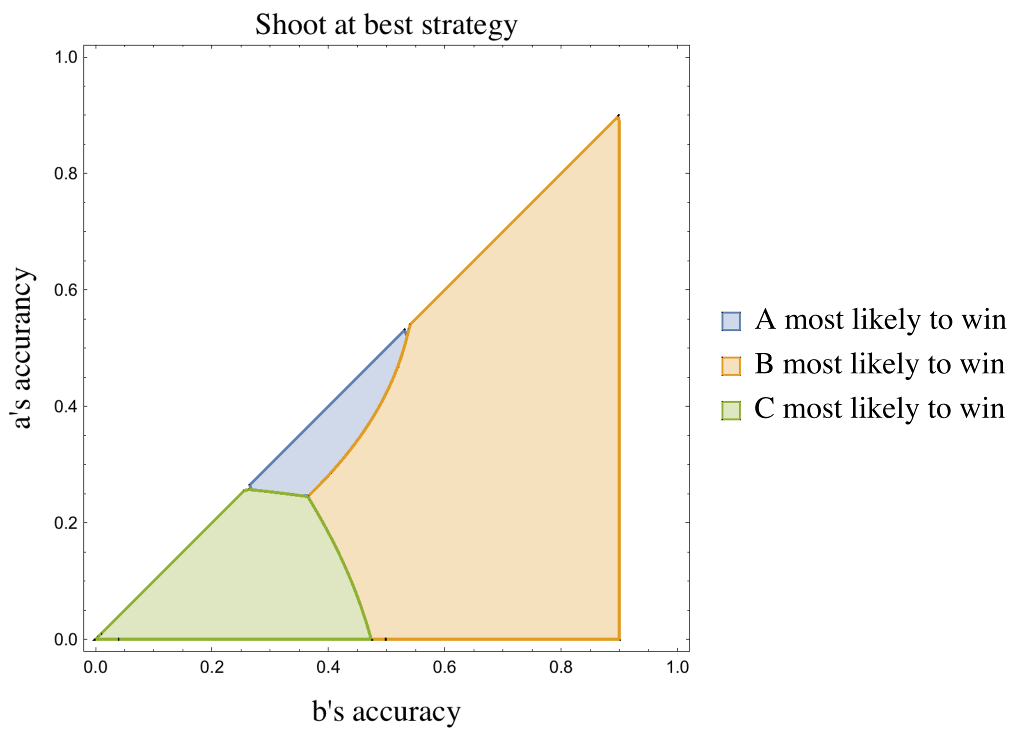
```
{bestawin, randomawin, groundawin} /. {a → .4, b → .8, c → .9}
{0.304153, 0.294364, 0.449216}
```

and it is best for A to shoot at the ground and waste the shot.

So if you know the values for a, b, and c this lets you pick the best strategy. But what if you only know some of the accuracy numbers (or even none of them), subject to $0 < a < b < c < 1$? For example what if you know $c = .9$ but not the specific values for a and b?

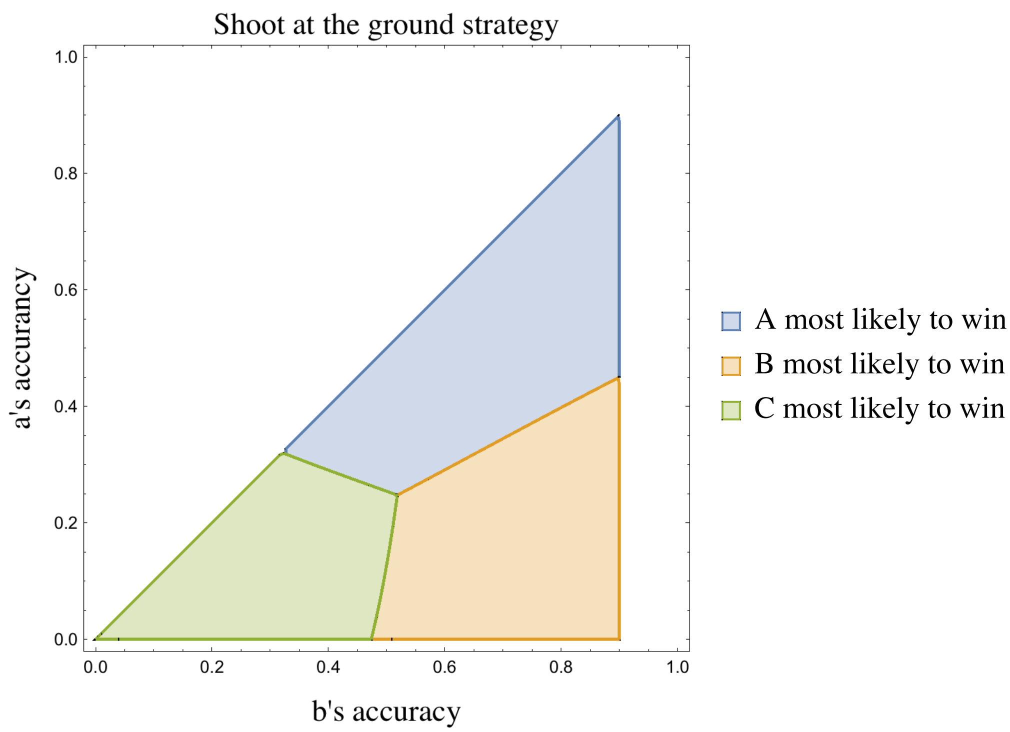
In this case we can use *Mathematica* to find ranges of values for a and b where each of the three players is most likely to win and see which range is the largest. For example if we use the "shoot at best" strategy with $c = .9$ we could set up and graph the regions of values for a and b where each player is most likely to win:

```
alikelytowin = PDF[ shootatbestprocess[∞], 10] /. c → .9;
blikelytowin = PDF[shootatbestprocess[∞], 11] /. c → .9;
clikelytowin = PDF[shootatbestprocess[∞], 12] /. c → .9;
bestplot = RegionPlot[
  {alikelytowin ≥ blikelytowin ⋀ alikelytowin ≥ clikelytowin ⋀ 0 ≤ a < b < .9,
   blikelytowin ≥ alikelytowin ⋀ blikelytowin ≥ clikelytowin ⋀ 0 ≤ a < b < .9,
   clikelytowin ≥ alikelytowin ⋀ clikelytowin ≥ blikelytowin ⋀ 0 ≤ a < b < .9},
  {b, 0, 1}, {a, 0, 1}, FrameLabel → (Style[Text[#], FontSize → 16] &) /@
    {"b's accuracy", "a's accurancy"}, PlotPoints → 50,
  PlotLegends → (Style[Text[#], FontSize → 16] & /@ {"A most likely to win",
      "B most likely to win", "C most likely to win"}), LabelStyle → Black,
  PlotLabel → Style[ Text["Shoot at best strategy"], FontSize → 16]]
```



The upper portion of this is empty as we know $a < b$. Clearly B has the best chance to win in this case as he has the largest area. Repeating this process using the "Shoot at the ground" strategy gives us this graph:

```
alikelytowin = PDF[ shootatgroundprocess[∞], 10] /. c → .9;
blikelytowin = PDF[shootatgroundprocess[∞], 11] /. c → .9;
clikelytowin = PDF[shootatgroundprocess[∞], 12] /. c → .9;
bestplot = RegionPlot[
  {alikelytowin ≥ blikelytowin ∧ alikelytowin ≥ clikelytowin ∧ 0 ≤ a < b < .9,
   blikelytowin ≥ alikelytowin ∧ blikelytowin ≥ clikelytowin ∧ 0 ≤ a < b < .9,
   clikelytowin ≥ alikelytowin ∧ clikelytowin ≥ blikelytowin ∧ 0 ≤ a < b < .9},
  {b, 0, 1}, {a, 0, 1}, FrameLabel → (Style[Text[#], FontSize → 16] &) /@
    {"b's accuracy", "a's accurancy"}, PlotPoints → 50,
  PlotLegends → (Style[Text[#], FontSize → 16] & /@ {"A most likely to win",
      "B most likely to win", "C most likely to win"}), LabelStyle → Black,
  PlotLabel → Style[ Text["Shoot at the ground strategy"], FontSize → 16]]
```



Comparing the areas in this graph shows that A has the largest range over which he is likely to win. So for $c = .9$ it looks like the "shoot the ground" strategy works better for duelist A than the "shoot the best" strategy. We could make this more specific by computing the areas to several decimal places and then scaling them up by the same factor so the total area of all 3 would be 1. That would give us the following computations:

For the "shoot at best" strategy:

```
alikelytowin = PDF[ shootatbestprocess[∞], 10] /. c → .9;
blikelytowin = PDF[shootatbestprocess[∞], 11] /. c → .9;
clikelytowin = PDF[shootatbestprocess[∞], 12] /. c → .9;
aifbest = NIntegrate[ 1, {a, b} ∈ ImplicitRegion[alikelytowin ≥ blikelytowin ∧
       alikelytowin ≥ clikelytowin ∧ 0 ≤ a < b < .9, {a, b}]] /
  NIntegrate[ 1, {a, b} ∈ ImplicitRegion[ 0 ≤ a ≤ b ≤ .9, {a, b}]]
0.0565597
```

versus the choice of the "shoot at the ground" strategy:

```
alikelytowin = PDF[ shootatgroundprocess[∞], 10] /. c → .9;
blikelytowin = PDF[shootatgroundprocess[∞], 11] /. c → .9;
clikelytowin = PDF[shootatgroundprocess[∞], 12] /. c → .9;
aifground = NIntegrate[ 1, {a, b} ∈ ImplicitRegion[alikelytowin ≥ blikelytowin ⋀
      alikelytowin ≥ clikelytowin ⋀ 0 ≤ a < b < .9, {a, b}]] /
  NIntegrate[ 1, {a, b} ∈ ImplicitRegion[ 0 ≤ a ≤ b ≤ .9, {a, b}]]
```

```
0.405542
```

So if $c = .9$ the switch in strategy from "shoot at best" to "shoot at the ground" increases A's chance of winning almost sevenfold. If we look at the "shoot at random" strategy we get the computations

```
alikelytowin = PDF[ shootatrandomprocess[∞], 10] /. c → .9;
blikelytowin = PDF[shootatrandomprocess[∞], 11] /. c → .9;
clikelytowin = PDF[shootatrandomprocess[∞], 12] /. c → .9;
aifrandom = NIntegrate[ 1, {a, b} ∈ ImplicitRegion[alikelytowin ≥ blikelytowin ⋀
      alikelytowin ≥ clikelytowin ⋀ 0 ≤ a < b < .9, {a, b}]] /
  NIntegrate[ 1, {a, b} ∈ ImplicitRegion[ 0 ≤ a ≤ b ≤ .9, {a, b}]]
```

```
0.102702
```

So it looks like if you know that you are the worst shooter and the best shooter is 90% accurate, your best chance is to go with the "shoot at the ground" strategy.

But what if you don't know the value for c? Can we come up with a statement that tells us under what conditions it is best for A to waste his shot? To do this can define the chance for A under each strategy and require various strategies be the best (it will be important to simplify and use the assumption that $0 < a < b < c < 1$):

```
chanceforaunderbest = PDF[ shootatbestprocess[∞], 10];
chanceforaunderrandom = PDF[ shootatrandomprocess[∞], 10];
chanceforaunderground = PDF[ shootatgroundprocess[∞], 10];
```

The conditions under which shooting at the best duelist is the best strategy are:

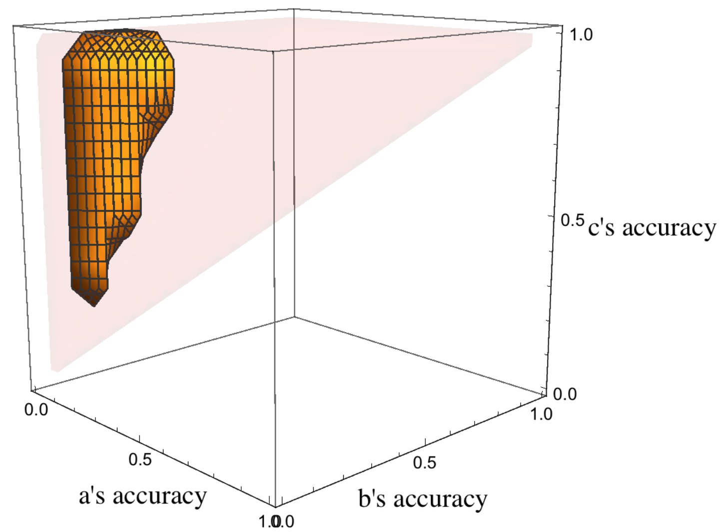
```
bestisbest =
 FullSimplify[Reduce[ chanceforaunderbest > chanceforaunderrandom ⋀
    chanceforaunderbest > chanceforaunderground ⋀ 0 < a < b < c < 1]]
```

$$\Bigg(0 < a < b \,\&\&\, c < 1 \,\&\& \Bigg(\Bigg(b + \text{Root}\left[1 + 8\,\#1 + 12\,\#1^2 + 4\,\#1^3\,\&,\, 3\right] < 0 \,\&\&\, \frac{b + b^2 + \sqrt{b^2 + 6\,b^3 - 3\,b^4}}{2\,(-1+b)^2} < c\Bigg) \;||\; \Bigg(b + \text{Root}\left[1 + 4\,\#1 + 3\,\#1^2 + \#1^3\,\&,\, 1\right] < 0 \,\&\&\, \frac{b + b^2 + \sqrt{b^2 + 6\,b^3 - 3\,b^4}}{2\,(-1+b)^2} \le c \,\&\&\, b + \text{Root}\left[1 + 8\,\#1 + 12\,\#1^2 + 4\,\#1^3\,\&,\, 3\right] \ge 0\Bigg)\Bigg)\Bigg) \;||$$

$$\Bigg(0 < a < \frac{c\left(-b + (-1+b)^2\,c\right)}{b^2 - b^2\,c + (-1+b)^2\,c^2} \,\&\&\, \frac{b}{(-1+b)^2} < c \,\&\& \Bigg(\Bigg(b > 0 \,\&\&\, b + \text{Root}\left[1 + 8\,\#1 + 12\,\#1^2 + 4\,\#1^3\,\&,\, 3\right] < 0 \,\&\&\, c \le \frac{1}{2}\left(\frac{b\,(1+b)}{(-1+b)^2} + \sqrt{\frac{b^2 + 6\,b^3 - 3\,b^4}{(-1+b)^4}}\right)\Bigg) \;||\; \Bigg(b + \text{Root}\left[1 + 4\,\#1 + 3\,\#1^2 + \#1^3\,\&,\, 1\right] < 0 \,\&\&\, c < \frac{1}{2}\left(\frac{b\,(1+b)}{(-1+b)^2} + \sqrt{\frac{b^2 + 6\,b^3 - 3\,b^4}{(-1+b)^4}}\right) \,\&\&\, b + \text{Root}\left[1 + 8\,\#1 + 12\,\#1^2 + 4\,\#1^3\,\&,\, 3\right] \ge 0\Bigg) \;||\; \left(b + \text{Root}\left[1 + 4\,\#1 + 3\,\#1^2 + \#1^3\,\&,\, 1\right] \ge 0 \,\&\&\, \sqrt{5} + 2\,b < 3 \,\&\&\, c < 1\right)\Bigg)\Bigg)$$

These complicated inequalities define a portion of 3D space which you can see below. The red shading is the portion of the 3D cube were $0 < a < b < c < 1$ (just as in the 2D pictures above the inequality $0 < a < b < .9$ forced the regions into half of a square, the inequality $0 < a < b < c < 1$ forces our regions into half a cube).

```
Show[RegionPlot3D[ bestisbest ∧ 0 < a < b < c < 1, {a, 0, 1},
    {b, 0, 1}, {c, 0, 1}, AxesLabel → (Style[ Text[#], FontSize → 14] & /@
      {"a's accuracy", "b's accuracy", "c's accuracy"}), LabelStyle → Black],
  RegionPlot3D[ 0 < a ≤ b ≤ c < 1, {a, 0, 1}, {b, 0, 1}, {c, 0, 1}, Mesh → None,
   PlotPoints → 30, PlotStyle → Directive[ Red, Opacity[.05]] ] ] // Quiet
```



The probability that the “shoot at the best” strategy is the best for A is the ratio of the solid volume divided by the volume of the lightly shaded portion of the cube.

```
NIntegrate[ 1, {a, b, c} ∈ ImplicitRegion[bestisbest && 0 < a < b < c < 1, {a, b, c}] ] /
 NIntegrate[1, {a, b, c} ∈ ImplicitRegion[0 < a < b < c < 1, {a, b, c}] ]
```

```
0.177789
```

Next we find those ranges for *a*, *b*, and *c* where the “shooting randomly” is best for A:

```
randomisbest = Reduce[ chanceforaunderbest <  chanceforaunderrandom ∧
   chanceforaunderrandom > chanceforaunderground ∧ 0 <  a < b < c <  1]
```

```
False
```

So shooting randomly turns out to never be the best strategy of the three.

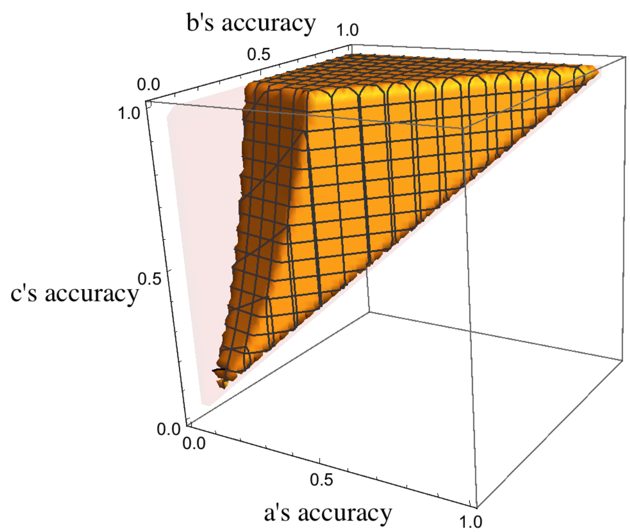
Checking the conditions under which shooting at the ground is best gives us this computation:

```
groundisbest =
 FullSimplify[Reduce[ chanceforaunderrandom ≤ chanceforaunderground ∧
    chanceforaunderbest ≤ chanceforaunderground ∧ 0 < a < b < c < 1]]
```

$$\left(0<a<b \,\&\&\, b<c \,\&\&\, \left(\left(\sqrt{5}+2b\geq 3 \,\&\&\, b<1 \,\&\&\, c<1\right) \,||\, \left(\sqrt{5}+2b<3 \,\&\&\, c\leq \frac{b}{(-1+b)^2}\right)\right)\right) \,||\, \left(\frac{c\left(-b+(-1+b)^2 c\right)}{b^2-b^2c+(-1+b)^2c^2}\leq a<b \,\&\&\, \frac{b}{(-1+b)^2}<c \,\&\&\, \left(\left(b>0 \,\&\&\, c<\frac{1}{2}\left(\frac{b(1+b)}{(-1+b)^2}+\sqrt{\frac{b^2+6b^3-3b^4}{(-1+b)^4}}\right) \,\&\&\, b+\text{Root}\left[1+4\#1+3\#1^2+\#1^3\,\&,1\right]\leq 0\right) \,||\, \left(b+\text{Root}\left[1+4\#1+3\#1^2+\#1^3\,\&,1\right]>0 \,\&\&\, \sqrt{5}+2b<3 \,\&\&\, c<1\right)\right)\right)$$

Seeing what portion of the 3D cube these inequalities define looks like

```
Show[RegionPlot3D[ groundisbest ∧ 0 < a < b < c < 1, {a, 0, 1}, {b, 0, 1},
   {c, 0, 1}, PlotPoints → 50, AxesLabel → (Style[ Text[#], FontSize → 14] & /@
      {"a's accuracy", "b's accuracy", "c's accuracy"}), LabelStyle → Black],
  RegionPlot3D[ 0 < a ≤ b ≤ c < 1, {a, 0, 1}, {b, 0, 1}, {c, 0, 1}, Mesh → None,
   PlotPoints → 30, PlotStyle → Directive[ Red, Opacity[.05]] ] ] // Quiet
```



And computing the probability that the “shoot the ground” strategy is the best for A is again the ratio of the two volumes and is given by:

```
NIntegrate[ 1, {a, b, c} ∈ ImplicitRegion[groundisbest && 0 < a < b < c < 1, {a, b, c}] ] /
 NIntegrate[1, {a, b, c} ∈ ImplicitRegion[0 < a < b < c < 1, {a, b, c}] ]
0.822211
```

Of course the approximate probabilities 0.177789, 0 (for False), and 0.822211 add up to 1 as you would expect.

So suprisingly if all you know is that you are the worst duelist of the three, there is about an 82.2% chance that wasting your shot is a better strategy for you than either shooting randomly or at the best duelist!

Now this does not necessarily mean that you should pick the “shoot at ground” strategy if you don’t have the specific numbers for *a*, *b*, and *c*. While “shoot the ground” is the best strategy about 82.2% of the time, it might be the case that one of the other strategies is just a tiny bit worse that 82.2% of the time but a lot better the other 17.8% of the time - in which case the other strategy might give you the best chance overall. We can compute A’s survival chances under each strategy using the same kind of ratios as we used before.

Using the “shoot at ground” strategy, A’s chance of winning the truel is:

```
NIntegrate[chanceforaunderground,
   {a, b, c} ∈ ImplicitRegion[0 < a < b < c < 1, {a, b, c}]] /
 NIntegrate[1, {a, b, c} ∈ ImplicitRegion[0 < a < b < c < 1, {a, b, c}]]
```

```
0.334523
```

Using the “shoot at best” strategy, A’s chance of winning the truel is:

```
NIntegrate[chanceforaunderbest,
   {a, b, c} ∈ ImplicitRegion[0 < a < b < c < 1, {a, b, c}]] /
 NIntegrate[1, {a, b, c} ∈ ImplicitRegion[0 < a < b < c < 1, {a, b, c}]]
```

```
0.259532
```

Using the “shoot at random” strategy, A’s chance of winning the truel is:

```
NIntegrate[chanceforaunderrandom,
   {a, b, c} ∈ ImplicitRegion[0 < a < b < c < 1, {a, b, c}]] /
 NIntegrate[1, {a, b, c} ∈ ImplicitRegion[0 < a < b < c < 1, {a, b, c}]]
```

```
0.243959
```

So not only is the “shoot at the ground” strategy the best 82.2% of the time, it is overall A’s best strategy for winning the truel.