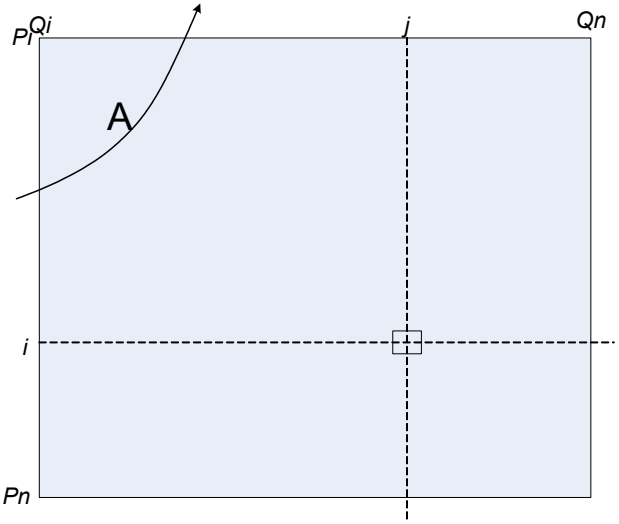
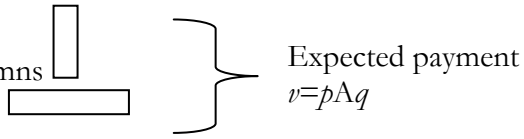
# Last time:

We went over two-player, zero-sum games, where each players takes turns choosing a row or column. A row player chooses a row, while the column players then chooses a column




Row player- chooses row
Column player – chooses column
Row player gives column player position $a_{ij}$ which is how much the row player has to pay the column player

The rows and columns each have a probability distribution of values which mean there is a randomized Nash



Probability distribution q on columns
Probability distribution p on rows

Expected payment
$v = pAq$

Now consider a Nash configuration from the two player's perspective:

-row $\underbrace{Aq}_{\text{expected loss on rows}}$ $p_i > 0 \rightarrow$ $(Aq)$ min values in entry $i$

-column $\underbrace{pA}_{\text{expected win on columns}}$ $q_i > 0 \rightarrow$ $(pA)$ max value in entry $j$

Expected payment row → column = $v$

Here the products $Aq$ and $pA$ represent the distribution of outcomes from the row-player's and column-players' perspective.

## *Row-player's minimum guarantee-able loss*

Now we look at the minimum guaranteed loss for a row-player, given that the row-player publishes their strategy.

$p$ probability dist. $\geq 0$ & $1 \times p = 1$ Also, this should have the property where $p$ is published; thus, the column player can examine it and choose a maximal strategy.

Therefore, the row-player makes max. entry $pA$ as small as possible.

$pA \leq v_R e$ & and the goal is to minimize $v_R$ $\quad e = (1,\cdots,1)$

## *Column-player's guarantee-able win*

This is an opposite inequality system, which is the maximum guaranteed win for a column, given a published strategy.

$q \geq 0 \mid q=1$

We make the min. entry in A$q$ as high as possible.

A$q \geq v_C e$ *(where e is a column vector here)* & the column-player tries to maximize $v^R$

---

From the above guarantees above, we see a relationship between the row-player's minimized loss and the column-player's maximized win and that overall $v_R \geq v_C$. To see why, assume that they both play then own strategies with best guaranteed value, then the value $x$ of this play must satisfy overall $v_R \geq x \geq v_C$ by the two guarantees, and hence $v_R \geq v_C$.

**Claim:** $v_R = v_C$ on all zero-sum games

**Proof:** $v$=Nash value. Note that Nash is a publishable strategy: the definition is that "even if the other player knows my strategy, he has no reason to change from his current strategy". This means that there is no harm in publishing the Nash strategy. The Nash strategy guarantees the Nash value. But the best publishable strategy can maybe be better?

Column-player's perspective: Nash value guaranteed ➔ $v_C \geq v$ *($v_C$ is better win)*

Row-player's perspective: Nash value guaranteed ➔ $v_R \leq v$ *($v_R$ is less loss)*

But from the above we know that $v_R \geq v_C$. putting these together, we get that $v_R = v_C = v$. Note that this also implies that there is only one possible Nash value $v$ (though there can be many Nash different equilibria): since the max. $v_C$ and the min $v_R$ are unique, so must plain $v$.

**Corollary:** Nash value $v$ is the same on all Nash equilibria

Note that there has been heavy linear programming (LP) usage, where you have a lot of inequalities and you look for solutions $v_R$ & $v_C$ that satisfy them all – this is basically an LP.

So overall, an important question to ask is: How do we find these Nashes? In the zero-sum game there are two good answers: Linear Programming and another which we will spend two weeks on.

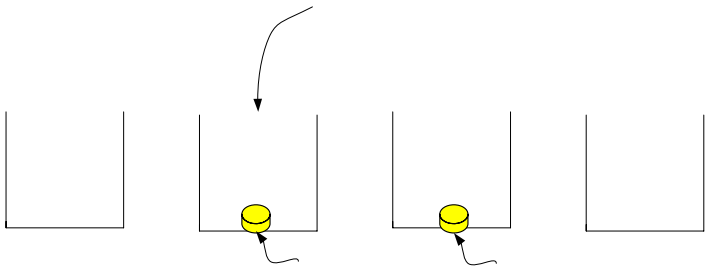
In linear programming we write a bunch of inequalities and throw them into a solver, and claim as a Nash whatever comes out. Specifically in this case we solve for max $v_C$ & min $v_R$ given guaranteed strategies $p$ & $q$. The resulting $p$ & $q$ is Nash because $v_R = v_C$. In more general cases, where it's harder to write inequalities, LP breaks down and we can't use this method.

Switching topics, we look at another strategy using machine learning

## Machine Learning

This is a simple version of this that is useful for determining Nash. Machine learning is roughly divided into two camps: Empirical machine learning, and more theoretical computational learning, where there is a desire for worst case guarantees.

We have $n$ machine slots, where a penny is randomly put into a slot. Additionally, Eva goes into a slot looking for a penny to be dropped. If she's in a slot when a penny is dropped then that's cool.



This is a simple zero-sum game where Eva chooses where to stand and then the other player chooses where to drop the penny.

Given a published strategy for Eva of only picking a slot randomly, how much can we expect to win?

P$i = 1/n$

Reasonably, we expect Eva to win $1/n$ coins. Now what if the pennies always go into the same bin, yet Eva always goes to a random slot. She really should learn the game that the penny-player is playing.

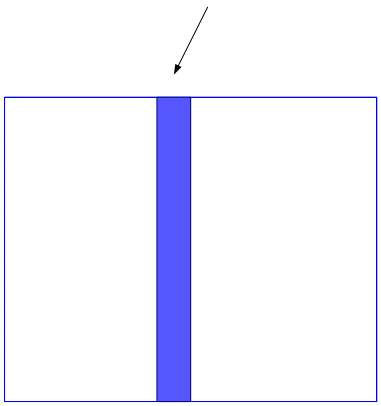
The standard of comparison will be the best bin which we could have stood in all along, with hindsight. We'll see which bin was the best overall and specifically, we'll look at the max number of pennies for a single bin.

This type of comparison was originally invented for finance. This is like knowing that Google would do well, and with hindsight, make sure that you own only Google stock.

This standard will be $d_{max}$. Note that $d_{max}$ is not the most you could earn. It would be better to predict each move and collect every penny, instead of waiting in the best slot.

**Theorem:** There is algorithm in expectation that gets $\geq d_{\max} - \sqrt{2d_{\max} \ln n}$. If $d_{max}$ is big enough, then this algorithm is amazingly good.

So why should be care about this – doesn't this seem weird – looking a bins with hindsight?



We're going to watch what the other player does and then input that into the probability distribution. Therefore the empirical distribution is:

P$i$ = percent of time row-players choose row $i$

Given this notation, what is the meaning of this standard? Again, what does this mean, the single best column in hind sight?

For next time:….

**Claim:** Max income in single bin (in hindsight)=max entry in $p$A

This is similar to the situation before – the best entry is based on what other player did first.