29: Simulating Probabilities

numpy
scipy.stats
.poisson.rvs
(λ)

Lisa Yan and Jerry Cain

November 18, 2020

Quick slide reference

random.random()

Since computers are deterministic, true randomness does not exist.

We settle for pseudo-randomness: A sequence that looks random but is actually deterministically generated.

`random.random()`, `np.random.random()`

- returns a float uniformly in [0.0, 1.0) with the Mersenne Twister:
- 53-bit precision floating point, repeats after 2**19937-1 numbers
- Seed number: X_0 used to generate sequence $X_1, X_2, \ldots, X_n, \ldots$

$\cdot 2^{53}$

Initialization [edit]

The state needed for a Mersenne Twister implementation is an array of *n* values of *w* bits each. To initialize the array, a *w*-bit $\boxed{\text{seed value}}$ is used to supply x_0 through x_{n-1} by setting x_0 to the seed value and thereafter setting

$$x_i = f \times (x_{i-1} \oplus (x_{i-1} >> (w-2))) + i$$

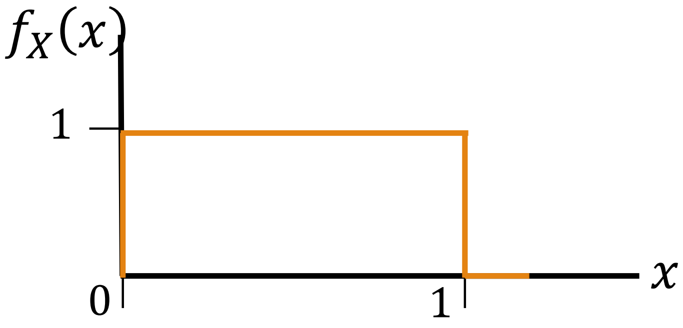
pset1_code — vim cs109_pset1.py — 73×18

```
def q14(seed: int = 37, ntrials: int = 100000) -> float:
    """
    Plays a game described in q14 ntrials times with a predetermined seed
.
    :param seed: seed for the numpy random number generator.
    :param ntrials: the number of trials to run.
    :return: the probability as described in the written pset.
    """
    np.random.seed(seed)
```

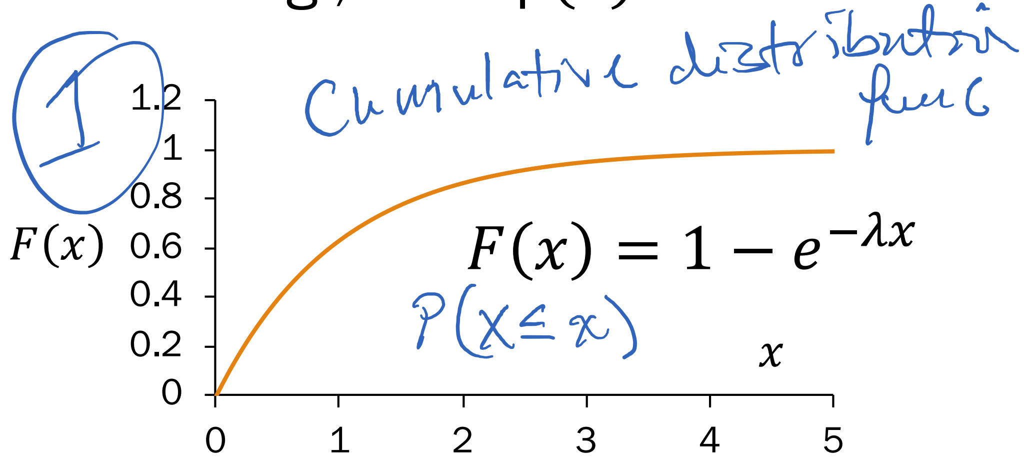
Remember Problem Set 1???

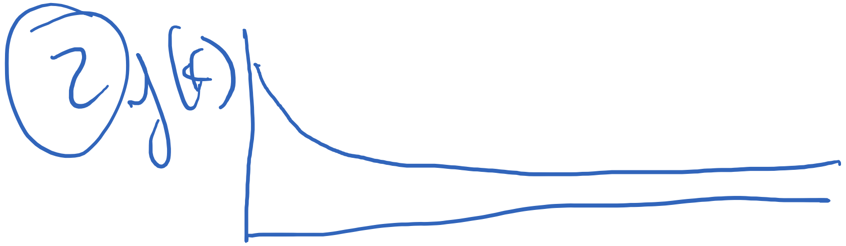
From `random.random()` to everything else

`random.random()`
`np.random.random()`
Generate a random float
in interval [0.0, 1.0)
$U \sim \text{Uni}(0,1)$



Generate a random number
X according to a distribution
e.g., $X \sim \text{Exp}(\lambda)$





Inverse Transform Sampling

Inverse Transform Sampling

$F(x) = P(X \leq x)$

Given the ability to generate numbers $U \sim \text{Uni}(0,1)$, how do we generate another number according to a CDF F?

$F(F^{-1}(a)) = F(b)$

$a = F(b)$

$$\boxed{X = F^{-1}(U)}$$

<u>def</u> F^{-1} the inverse of CDF: $F^{-1}(a) = b \Leftrightarrow F(b) = a$

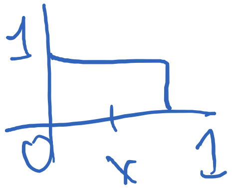
<u>Interpret</u>

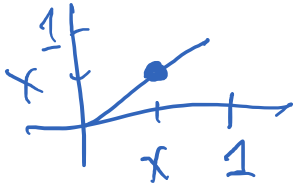
1. Generate $U \sim \text{Uni}(0,1)$
2. Apply inverse F^{-1} to get a RV X.
3. Then X will have CDF F.

Proof: $P(X \leq x) = P(F^{-1}(U) \leq x)$ (our definition of X)

$P(F(F^{-1}(u)) \leq F(x))$

CDF of U u





$= P(U \leq F(x))$ ($\forall x$: $0 \leq F(x) \leq 1$)

$= F(x)$ (CDF $P(U \leq u) = u$ if $0 \leq u \leq 1$)

Inverse Transform Sampling (Continuous)

How do we generate the exponential distribution $X \sim \text{Exp}(\lambda)$?

- CDF: $F(x) = 1 - e^{-\lambda x}$ where $x \geq 0$
- Compute inverse:

$$F^{-1}(u) = -\frac{\log(1-u)}{\lambda}$$

- Note if $U \sim \text{Uni}(0,1)$, then $(1-U) \sim \text{Uni}(0,1)$
- Therefore:

$$F^{-1}(U) = -\frac{\log(U)}{\lambda}$$

- Note: Closed-form inverse may not always exist

$F(x) = 1 - e^{-\lambda x} = u$

$1 - u = e^{-\lambda x}$

$\log(1-u) = -\lambda x$

$x = \frac{-\log(1-u)}{\lambda}$

Check it out!!! (demo)

Inverse Transform Sampling (Discrete)

$X \sim \text{Poi}(\lambda = 3)$ has CDF $F(X = x)$ as shown:

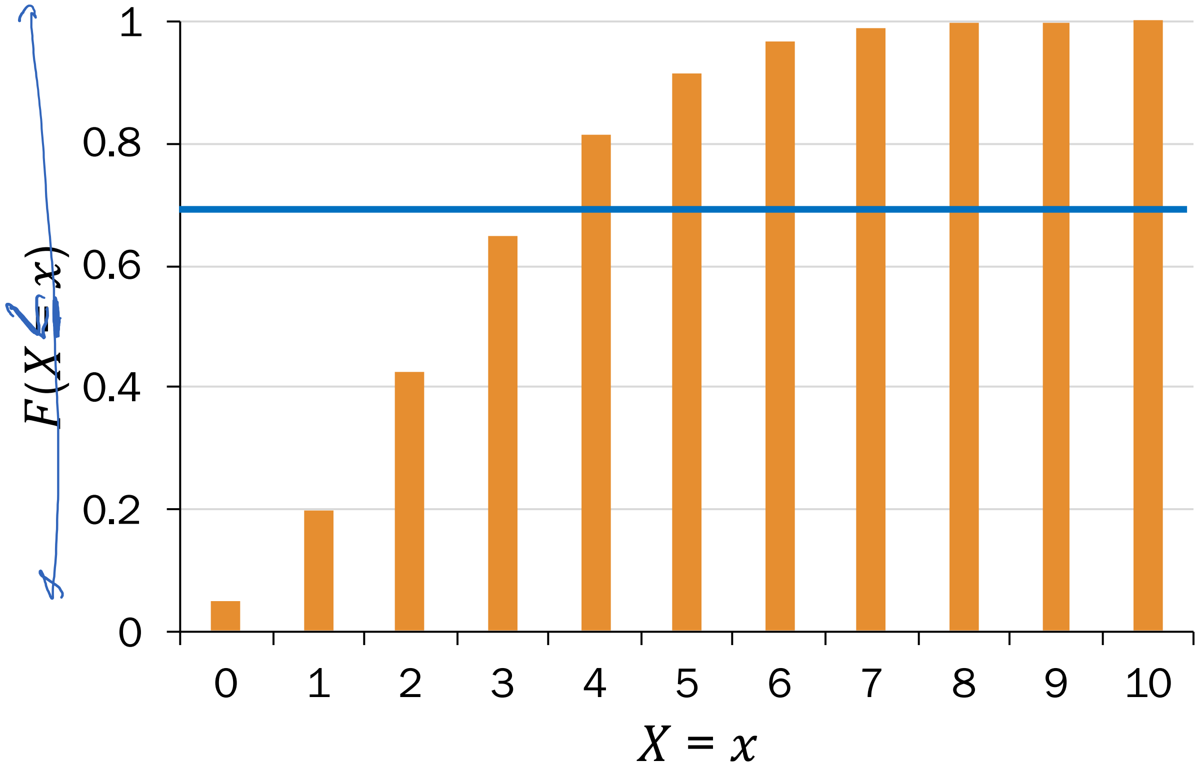
1. Generate $U \sim \text{Uni}(0,1)$

   $u = 0.7$

2. As x increases, determine first $F(x) \geq U$

   $x = 4$

3. Return this value of x



Check it out!!! (demo)

Inverse Transform Sampling of the Normal?

How do we generate $X \sim \mathcal{N}(0,1)$?

Inverse transform sampling:

1. Generate a random probability u from $U \sim \text{Unif}(0,1)$.
2. Find x such that $\Phi(x) = u$. In other words, compute $x = \Phi^{-1}(u)$.

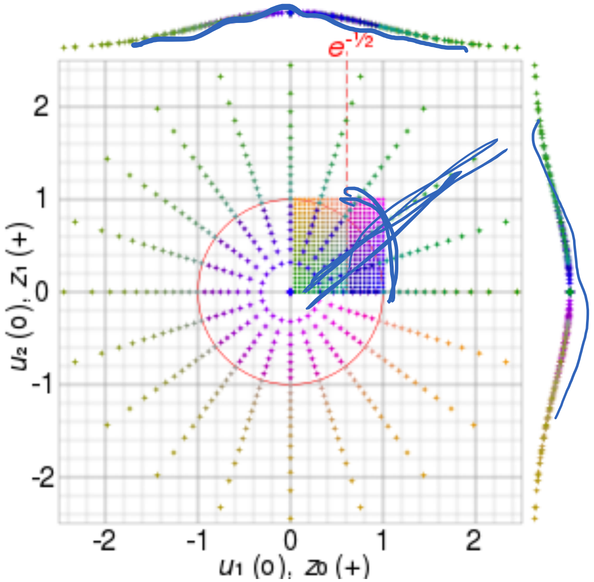
⚠ Φ^{-1} has no analytical solution!

<u>Solution</u> Box-Muller Transform

$R = \sqrt{-2 \log U_1}$

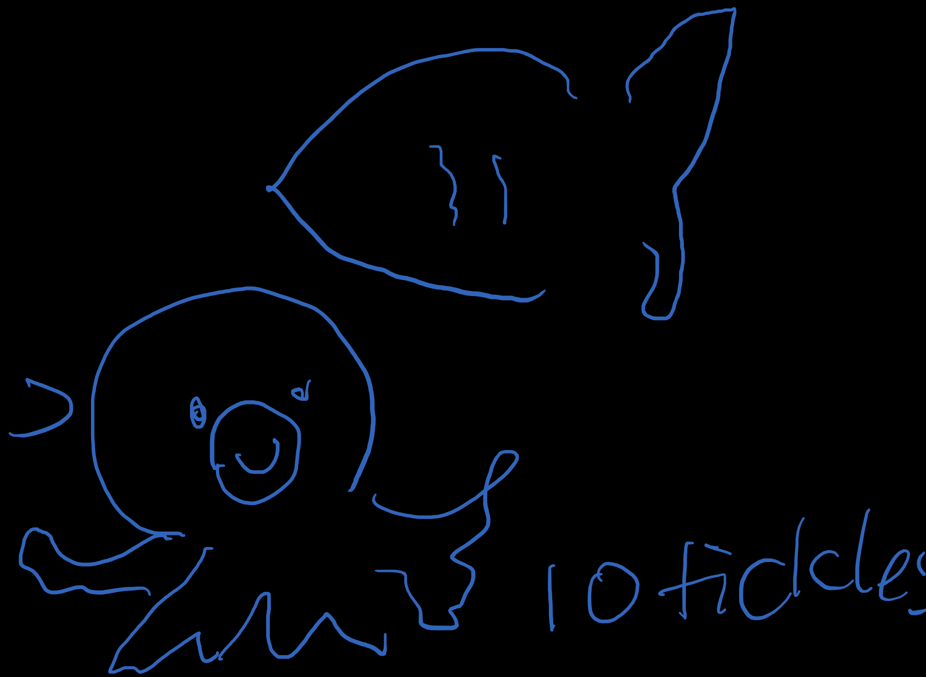
$\Theta = 2\pi U_2$

- Use **two** uniforms U_1 and U_2 to generate polar coordinates R and Θ for a circle inscribed in 2x2 square centered at (0,0)
- Can define $X = R\cos\Theta$, $Y = R\sin\Theta$ such that X and Y are **two** independent unit Normals



https://en.wikipedia.org/wiki/Box%E2%80%93Muller_transform

Haha



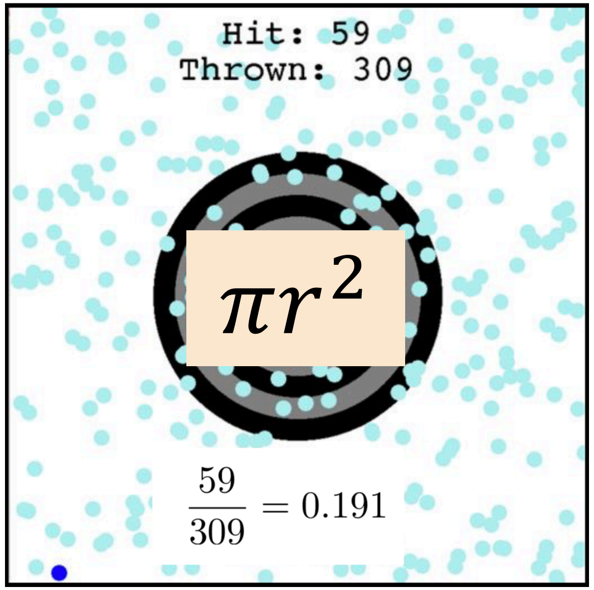
Interlude for jokes/announcements

Monte Carlo Methods

Monte Carlo Integration

Monte Carlo methods: randomly sample repeatedly to obtain a numerical result

- Bootstrap
- Inference in Bayes Nets
- Definite integrals (Monte Carlo integration)



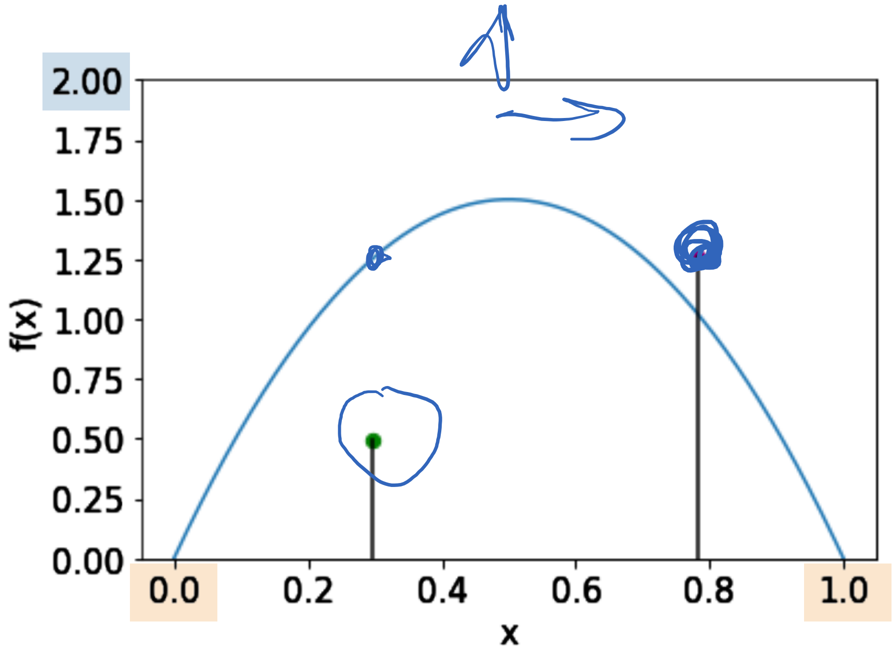
Named after area in Monaco known for its casinos

A Monte Carlo method: Rejection Filtering

Lisa would rename to Acceptance Filtering

Idea for X with PDF $f(x)$:

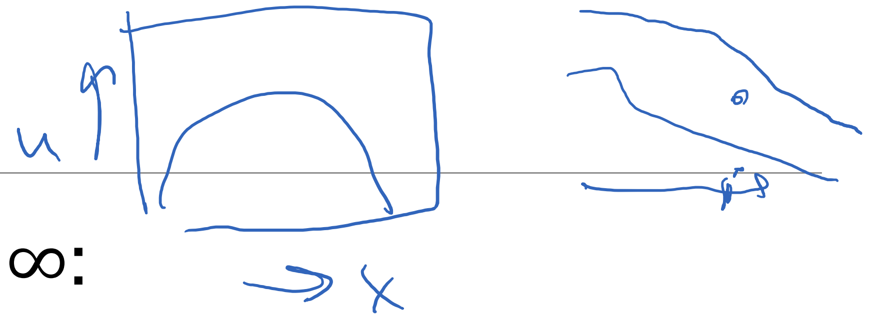
- Throw dart at graph of PDF $f(x)$
- If dart under $f(x)$: return x
- Otherwise, repeat throwing darts until one lands under $f(x)$




```
# random value from distr of X
def random_x():
  while True:
    u = random.random() * HEIGHT
    x = random.random() * WIDTH
    if u <= f(x):
      return x
```

But what if our PDF has infinite support?

Filtering with infinite support



Idea for X with PDF $f(x)$ with support $-\infty < x < \infty$:

- Suppose we can simulate Y with PDF $g(y)$ (where Y has same support as X)
- If we can find a constant c such that $c \geq f(x)/g(x)$ for all x, then

```
def random_x():
  while True:
    u = random.random()  # u ~ Uni(0, 1)
    x = generate_y()     # random value Y = y
    if u <= f(x)/(c * g(x)):
      return x
```

- Number of iterations of loop~Geo$(1/c)$
- Proof of correctness in Ross textbook, 10.2.2

Generating Normal Random Variable

$c \geq \frac{f(x)}{g(x)}$ $\quad \frac{1}{c} \cdot \frac{f(x)}{g(x)}$

Goal: Simulate $Z \sim \mathcal{N}(0,1)$. $-\infty < z < \infty$

$g(y) = e^{-y}$
$0 \leq y < \infty$

- Suppose we can simulate $Y \sim \text{Exp}(1)$ with the inverse transform.
- Let's simulate $X = |Z|$, which has the same support as Y. PDF f:

$$f(x) = \frac{2}{\sqrt{2\pi}} e^{-x^2/2}$$
$$0 \leq x < \infty$$

1. Determine constant $c \geq f(x)/g(x)$ for all $0 \leq x < \infty$:

$\sqrt{\frac{2}{\pi}} e^{-(x-1)^2/2} e^{+1/2}$

$$\frac{f(x)}{g(x)} = \sqrt{\frac{2}{\pi}} e^{-(x^2-2x)/2} = \sqrt{\frac{2}{\pi}} e^{-(x^2-2x+1)/2 + 1/2} = \sqrt{\frac{2e}{\pi}} e^{-(x-1)^2/2} \leq \boxed{\sqrt{\frac{2e}{\pi}}}$$

$\frac{e^{-x^2/2}}{e^{-x}}$

(complete the square)

$(e^{1/2} = \sqrt{e})$

Let this be c

2. Determine $f(x)/(cg(x))$

3. Implement code for $|Z|$ and Z

Generating Normal Random Variable

Goal: Simulate $Z \sim \mathcal{N}(0, 1)$.

- Suppose we can simulate $Y \sim \text{Exp}(1)$ with the inverse transform.
- Let's simulate $X = |Z|$, which has the same support as Y. PDF f:

$g(y) = e^{-y}$
$0 \le y < \infty$

$f(x) = \frac{2}{\sqrt{2\pi}} e^{-x^2/2}$
$0 \le x < \infty$

1. Determine constant $c \ge f(x)/g(x)$ for all $0 \le x < \infty$:

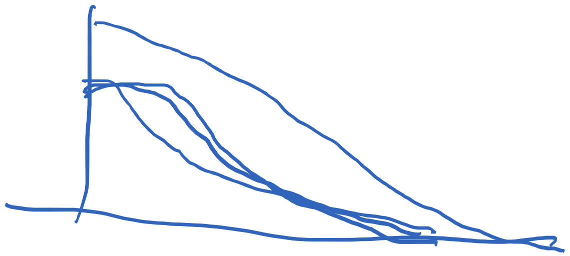
$$\frac{f(x)}{g(x)} = \sqrt{\frac{2}{\pi}} e^{-(x^2-2x)/2} = \sqrt{\frac{2}{\pi}} e^{-(x^2-2x+1)/2 + 1/2} = \sqrt{\frac{2e}{\pi}} e^{-(x-1)^2/2} \le \sqrt{\frac{2e}{\pi}}$$

(complete the square) $(e^{1/2} = \sqrt{e})$

Let this be c

2. Determine $f(x)/(c \cdot g(x))$

$$e^{-(x-1)^2/2}$$



3. Implement code for $|Z|$ and Z

Generating Normal Random Variable

Goal: Simulate $Z \sim \mathcal{N}(0, 1)$.

$$g(y) = e^{-y}$$
$$0 \le y < \infty$$

- Suppose we can simulate $Y \sim \text{Exp}(1)$ with the inverse transform.
- Let's simulate $X = |Z|$, which has the same support as Y. PDF f:

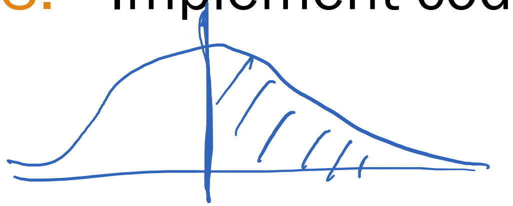
$$f(x) = \frac{2}{\sqrt{2\pi}} e^{-x^2/2}$$
$$0 \le x < \infty$$

3. Implement code for $|Z|$ and Z.

$$\frac{f(x)}{c \cdot g(x)} = e^{-(x-1)^2/2}$$

$$c = \sqrt{2e/\pi} \approx 1.32$$

(from last two slides)



```
# random value from distr of |Z|
def random_abs_z():
  while True:
    u = random.random()  # u ~ Uni(0, 1)
    # inverse transform to get x ~ Exp(1)
    x = -np.log(random.random())
    if u <= np.exp(-(x - 1) ** 2 / 2):
      return x
```

```
# random value from distr of Z
def random_z():
  abs_z = random_abs_z()
  u = random.random()
  if u < 0.5:
    return abs_z
  else:
    return -abs_z
```

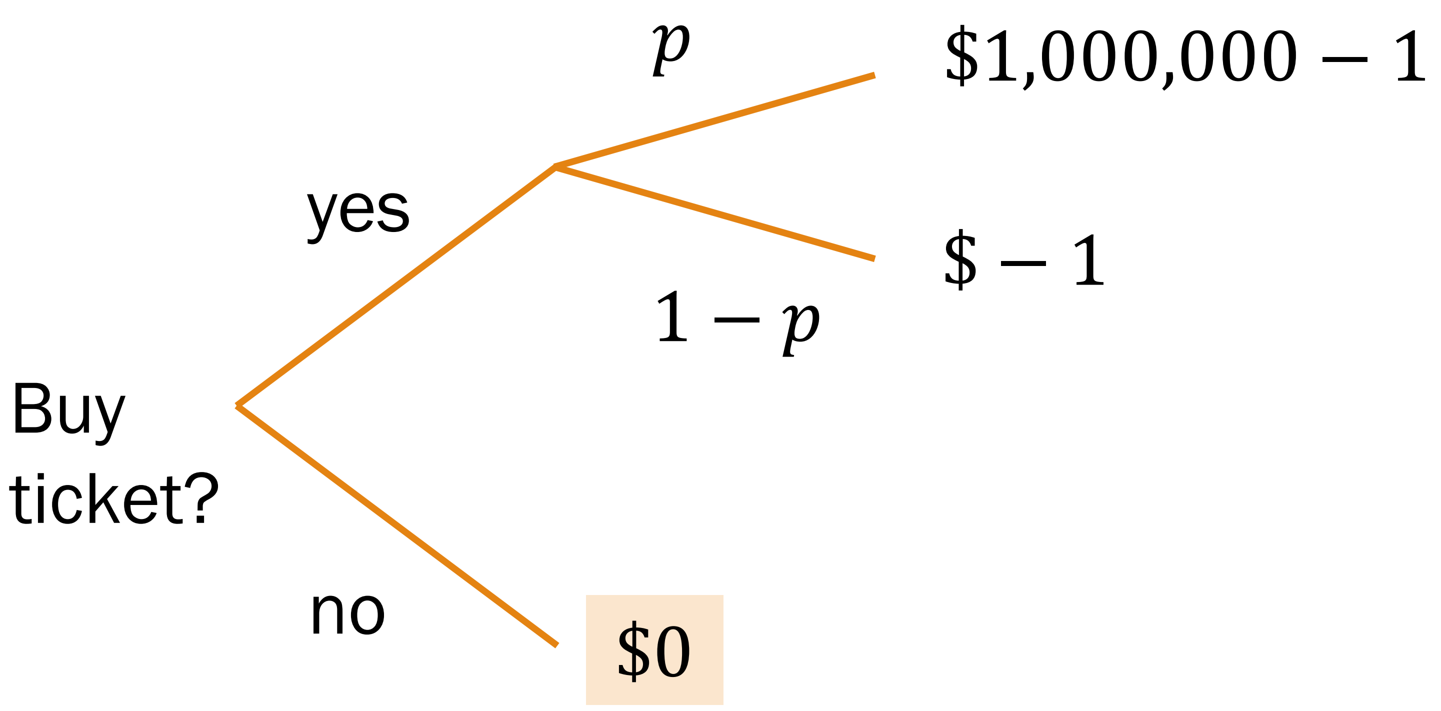
Black magic?

$$P(E) = \lim_{n\to\infty} \frac{n(E)}{n}$$

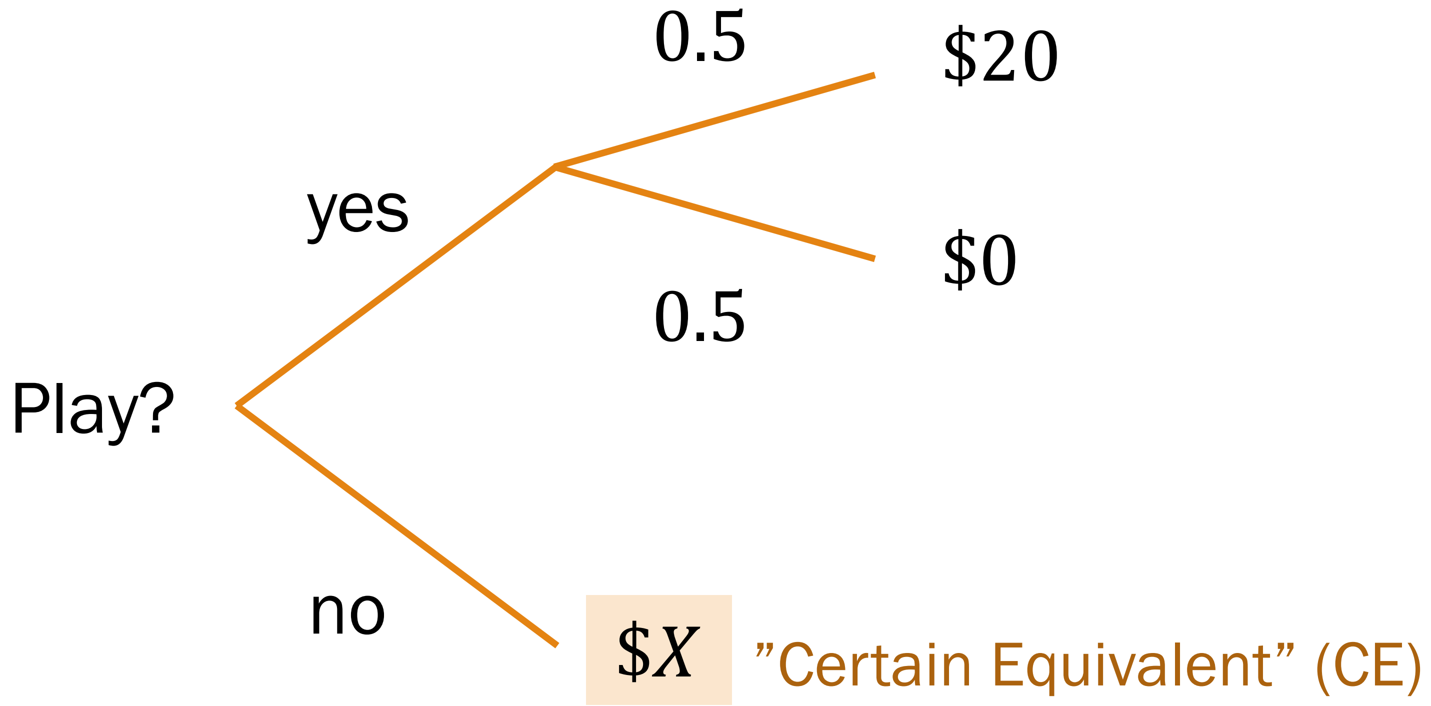
No—it's simulation!

Utility of Money

Recall the probability tree!



Let's play a game. What choice would you make?



For what value of $X are you <u>indifferent</u> to playing?

A. $X = 3$
B. $X = 7$
C. $X = 9$
D. $X = 10$

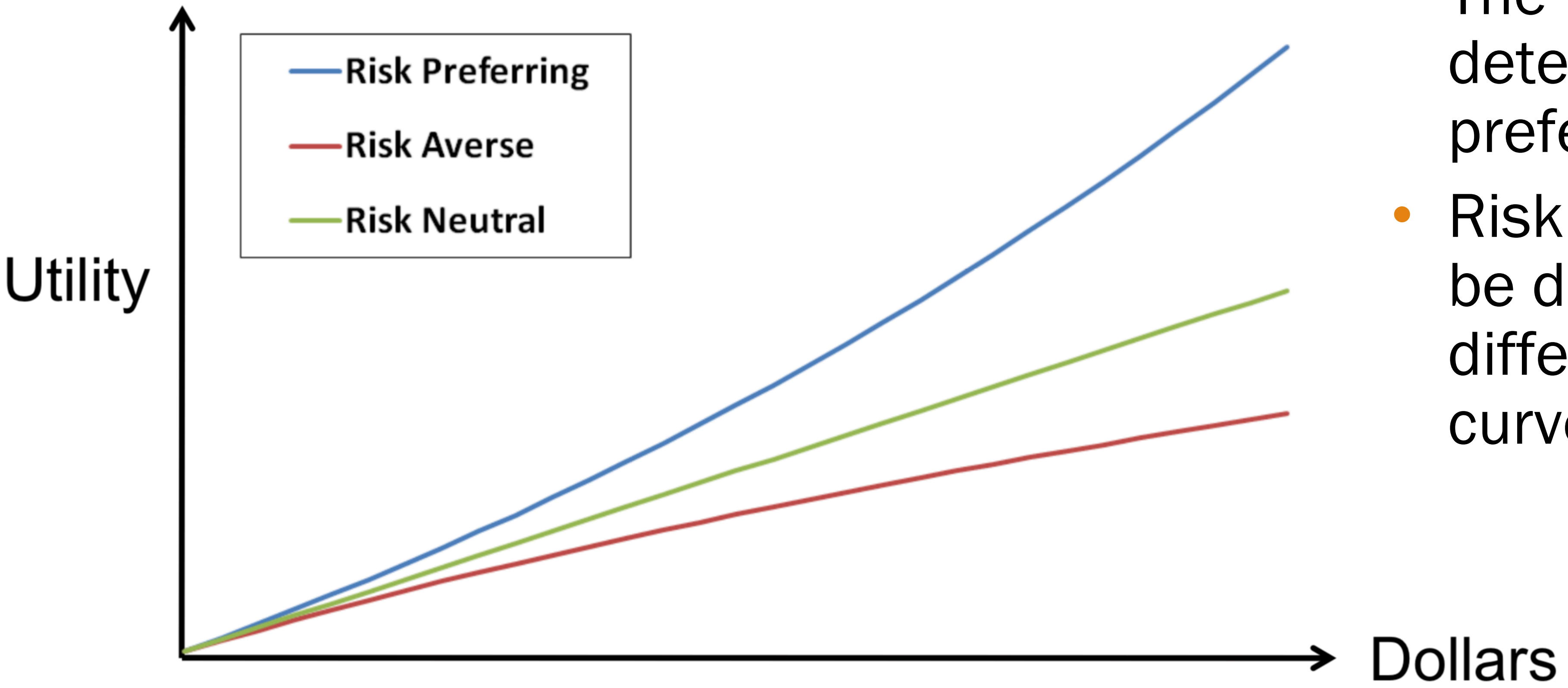
<u>def</u> Certain equivalent: The value of the game to *you* (different for different people)

Utility

def Utility $U(X)$ is the "value" you derive from X

- Can be monetary, but often includes intangibles like quality of life, life expectancy, personal beliefs, etc.

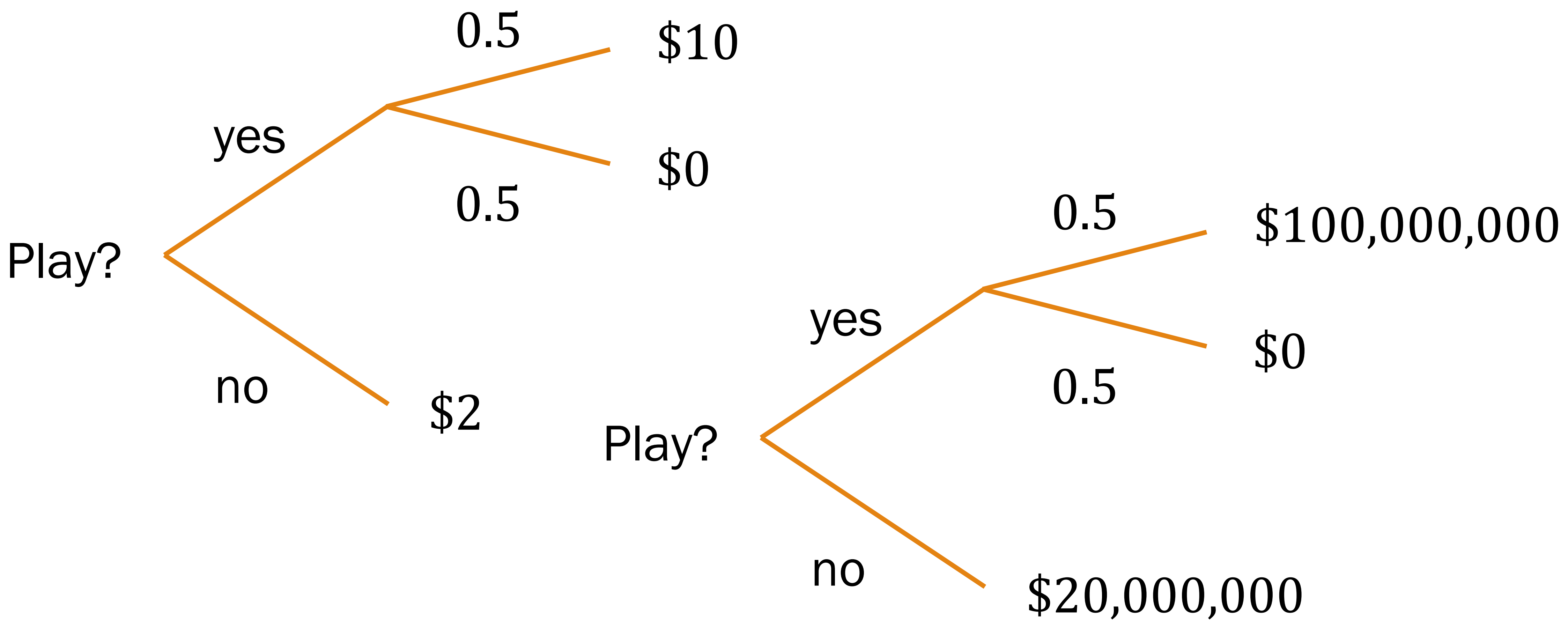
Utility curves



- The utility curve determines your "risk preference."
- Risk preference can be different in different parts of the curve

Non-linearity utility of money

Interestingly, these two choices are different for most people:



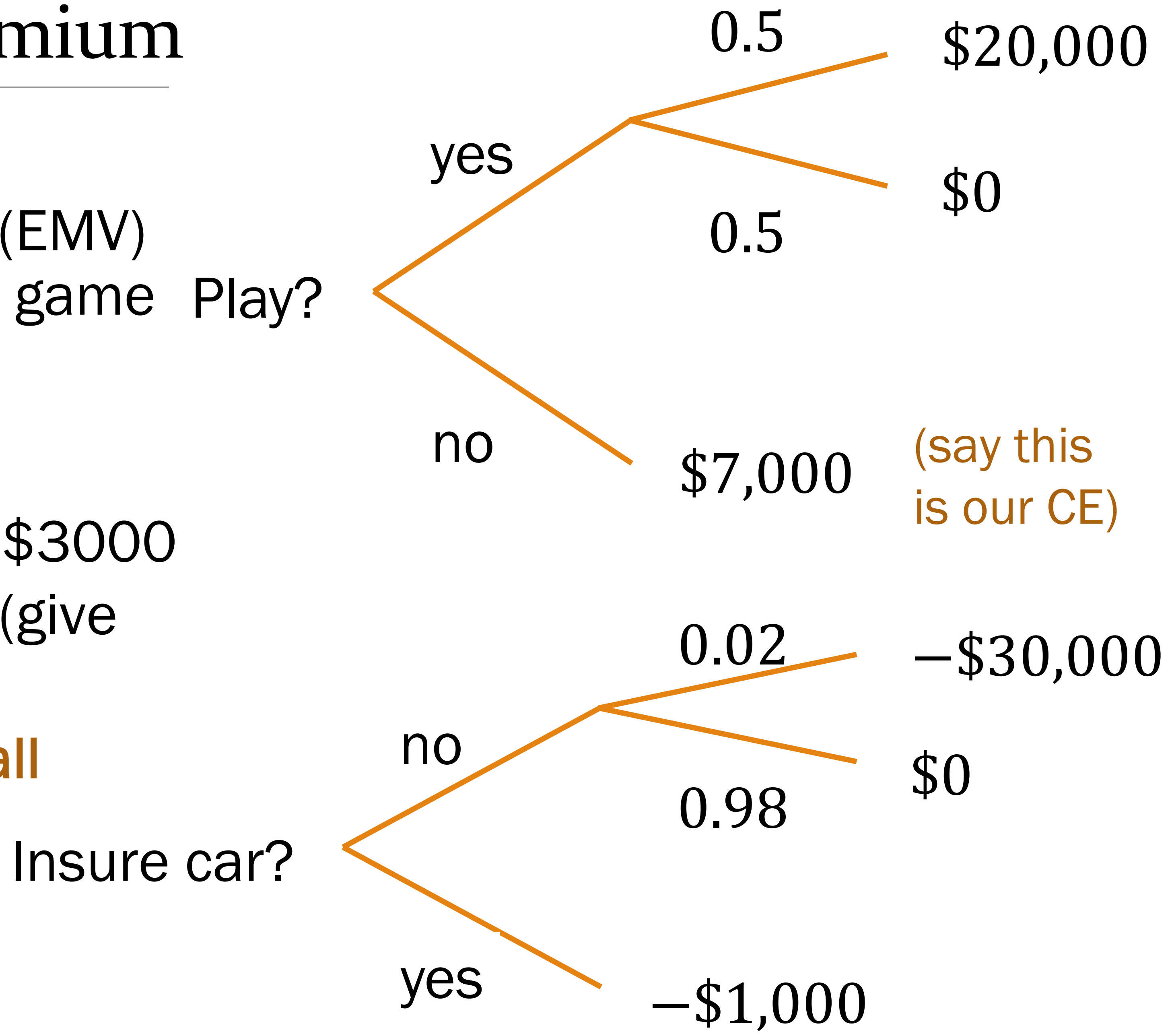
Insurance and risk premium

A slightly different game:

- Expected monetary value (EMV) = expected dollar value of game (here, $10,000)

Risk premium = EMV – CE = $3000

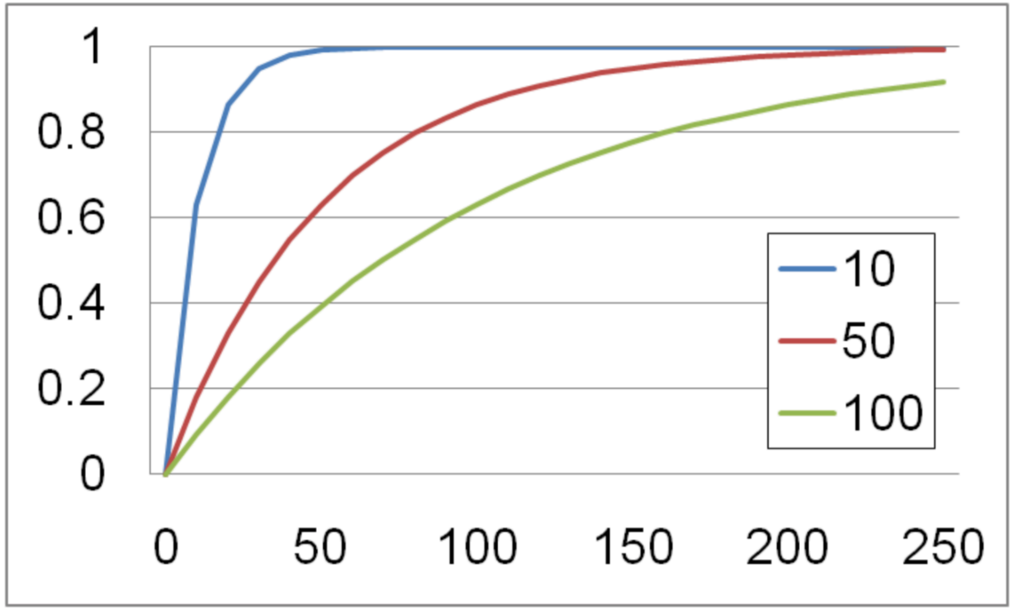
- How much would you pay (give up) to avoid risk?
- **This is what insurance is all about.**



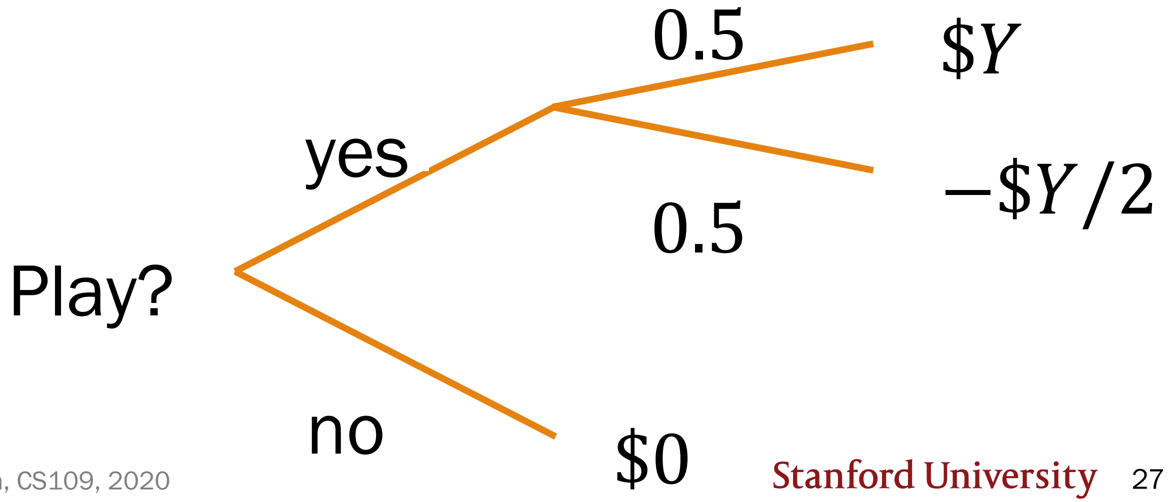
Exponential utility curves

Many people have exponential utility curves:

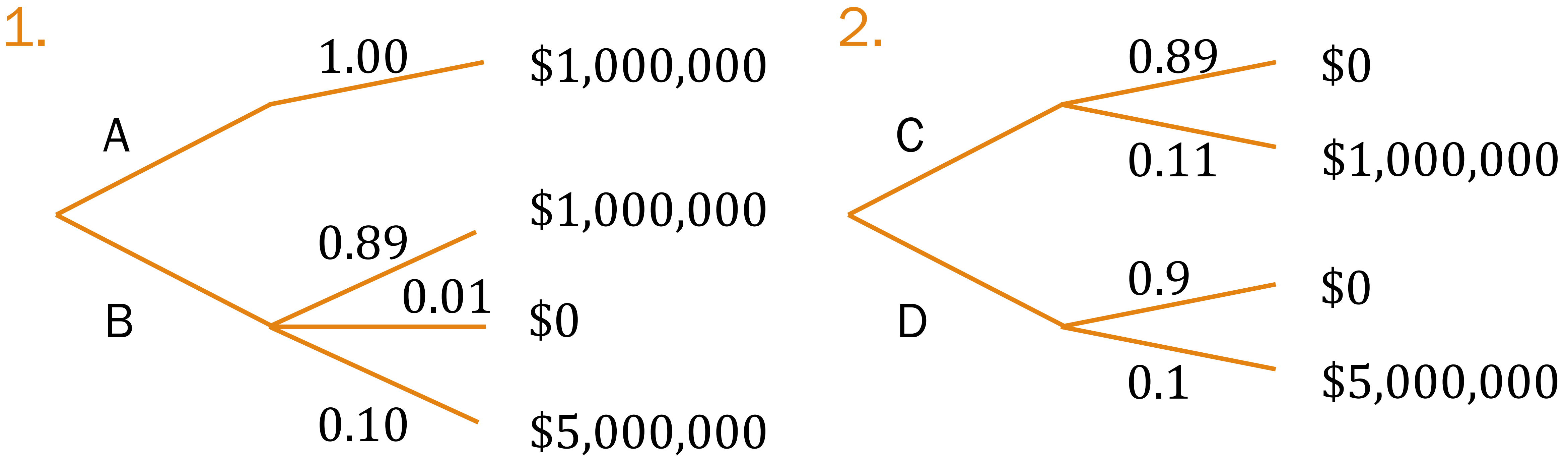
$$U(x) = 1 - e^{-x/R}$$



- R is your "risk tolerance"
- Larger R = less risk aversion. Makes utility function more "linear"
- $R \approx$ highest value of Y for which you would play:



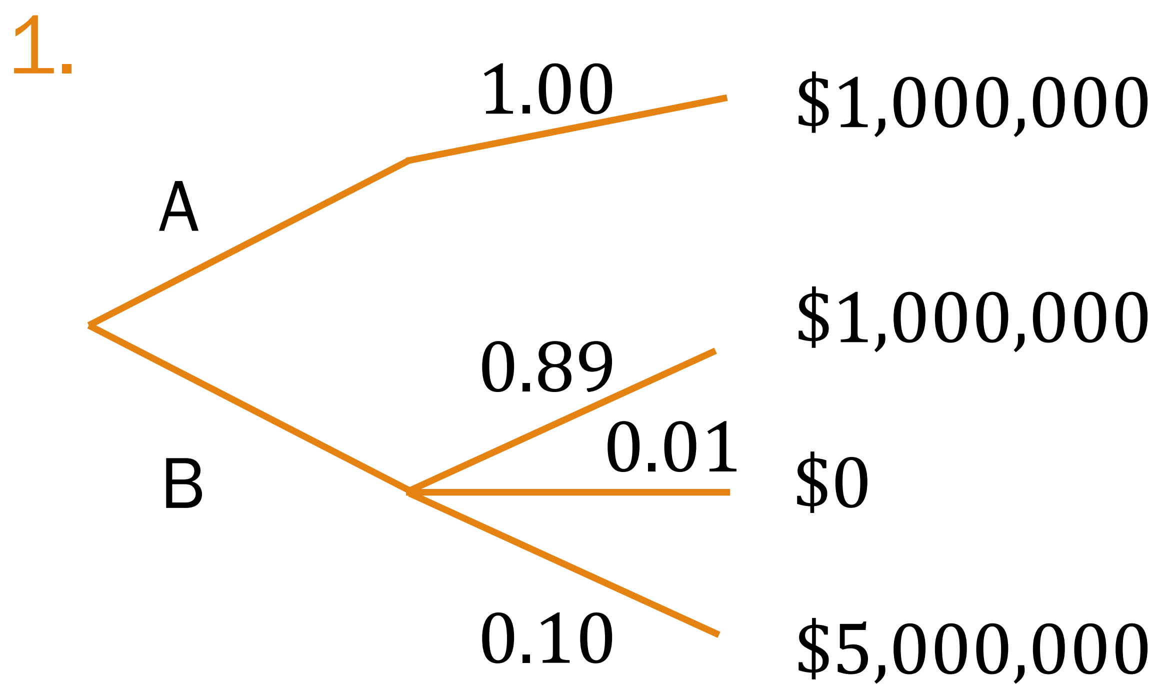
How rational are you?



Which option would you choose in each case?

How many of you chose A over B and D over C?

How rational are you?



Choice A preferred:
$1.00\ U(1{,}000{,}000) >$
$0.89\ U(1{,}000{,}000) + 0.01\ U(0)$
$+0.10\ U(5{,}000{,}000)$

Choice D preferred:
$0.89\ U(0) + 0.11\ U(1{,}000{,}000) <$
$0.90\ U(0) + 0.10\ U(5{,}000{,}000)$

How rational are you?

Choice D preferred:
$0.11\ U(1{,}000{,}000) < $
$0.01\ U(0)$
$+0.10\ U(5{,}000{,}000)$

subtract $0.89\ U(0)$ from both sides

Choice D preferred:
$0.89\ U(0) + 0.11\ U(1{,}000{,}000) <$
$0.90\ U(0) + 0.10\ U(5{,}000{,}000)$

add $0.89\ U(1{,}000{,}000)$ to both sides

Choice D preferred:
$1.00\ U(1{,}000{,}000) <$
$0.89\ U(1{,}000{,}000) +$
$0.01\ U(0) +$
$0.10\ U(5{,}000{,}000)$

Contradiction???

!

Choice A preferred:
$1.00\ U(1{,}000{,}000) >$
$0.89\ U(1{,}000{,}000) + 0.01\ U(0)$
$+0.10\ U(5{,}000{,}000)$

How rational are you?

Choice D preferred:

$1.00\ U(1{,}000{,}000) <$

$0.89\ U(1{,}000{,}000) +$

$0.01\ U(0) +$

$0.10\ U(5{,}000{,}000)$

add

$0.89\ U(1{,}000{,}000)$

to both sides

Choice D preferred:

$0.11\ U(1{,}00$

Con

subtract $0.89\ U(0)$

from both sides

You are inconsistent with utility theory (Allais Paradox)!

⚠️ Human behavior is not always axiomatically consistent ⚠️

Cho

1.00 ,000,000) >

$0.89\ U(1{,}000{,}000) + 0.01\ U(0)$

$+0.10\ U(5{,}000{,}000)$

Choice D preferred:

$0.89\ U(0) + 0.11\ U(1{,}000{,}000) <$

$0.90\ U(0) + 0.10\ U(5{,}000{,}000)$