In [1]:
```
from datascience import *
import numpy as np
```

Monty Hall Problem

In [2]:
```
hidden_behind_doors = ['first goat', 'second goat', 'car']

goats = ['first goat', 'second goat']
```

In [3]:
```
def other_goat(goat):
    if goat == 'first goat':
        return 'second goat'
    elif goat == 'second goat':
        return 'first goat'
```

In [4]:
```
other_goat('first goat')
```

Out[4]:
```
'second goat'
```

In [5]:
```
other_goat('second goat')
```

Out[5]:
```
'first goat'
```

In [6]:
```
other_goat('apple')
```

In [7]:
```
def play_monty_hall():
    """
    Returns a list of 3 items:
    [contestant's first guess, what Monty reveals, what's behind other door]
    """
    first_guess = np.random.choice(hidden_behind_doors)

    if first_guess == 'first goat':
        return [first_guess, 'second goat', 'car']

    elif first_guess == 'second goat':
        return [first_guess, 'first goat', 'car']

    elif first_guess == 'car':
        reveal = np.random.choice(goats)
        return ['car', reveal, other_goat(reveal)]
```

PROBABILITY 1/3, 1/3, 1/3

PROBABILITY 1/2, 1/2

In [8]:
```
play_monty_hall()
```

Out[8]:
```
['first goat', 'second goat', 'car']
```

In [9]:
```
results = Table(['Guess', 'Revealed', 'Remaining'])
```

In [10]:
```
results
```

Out[10]:

Guess	Revealed	Remaining

In [11]:
```
results.append(play_monty_hall())
```

Out[11]:

Guess	Revealed	Remaining
car	second goat	first goat

e.g. IF CONTESTANT (KNOWING NOTHING) CHOOSES DOOR W/ CAR (PROBABILITY = 1/3)
AND HOST (KNOWING EVERYTHING) REVEALS DOOR W/ 2ND GOAT (BY "RULES")
THEN FIRST GOAT REMAINS BEHIND OTHER DOOR

In [12]:
```
results
```

Out[12]:

Guess	Revealed	Remaining
car	second goat	first goat

In [13]:
```
trials = Table(['Guess', 'Revealed', 'Remaining'])

for i in np.arange(10000):
    trials.append(play_monty_hall())
```

REPEAT 10000 TIMES

In [14]:
```
trials
```

Out[14]:

Guess	Revealed	Remaining
second goat	first goat	car
car	first goat	second goat
second goat	first goat	car
second goat	first goat	car
first goat	second goat	car
second goat	first goat	car
second goat	first goat	car
first goat	second goat	car
first goat	second goat	car
car	first goat	second goat

... (9990 rows omitted)

In [15]:
```
trials.group('Guess')
```

Out[15]:

Guess	count
car	3379
first goat	3289
second goat	3332

In [16]:
```
trials.group('Remaining')
```

Out[16]:

Remaining	count
car	6621
first goat	1688
second goat	1691

2/3 OF TIME CAR IS BEHIND REMAINING DOOR.

1/3 OF TIME A GOAT IS BEHIND REMAINING DOOR
(AND CAR IS BEHIND ORIGINAL CONTESTANT GUESS)