MAKE
SCHOOL

# PROBABILITY

*You've almost surely encountered this idea before*

# WHAT IS PROBABILITY?

Pragmatic answer

- *A measure of the likelihood of an event*

Theoretical answer

- *A formal system to quantify uncertainty*

# APPLICATIONS

Everyday real-world problems deal with uncertain information and/or outcomes

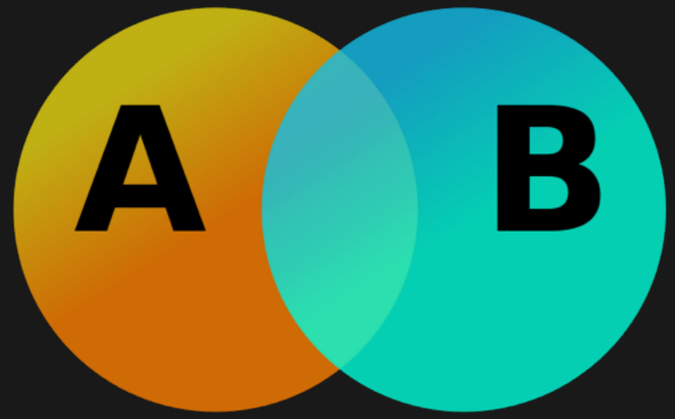
- Diagnosis – predict cause given symptoms (e.g., medical treatment, mechanical repairs)
- Risk assessment (e.g., financial, environmental)
- Product reliability (e.g., electronics, vehicles)

# RULES OF PROBABILITY

A and B are events of uncertain occurrence

Probability theory assumes these *axioms*:

- $0 \leq P(A) \leq 1$
- $P(True) = 1$ and $P(False) = 0$
- $P(A \text{ or } B) = P(A) + P(B) - P(A \text{ and } B)$



# DISCRETE PROBABILITY

Deals with events that occur in countable sample spaces

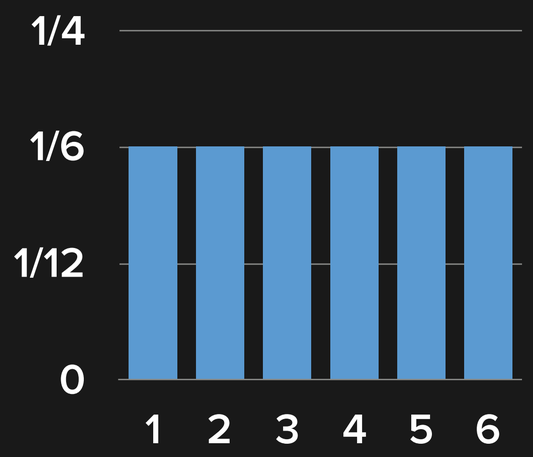
Examples: coins, dice, cards, random walks

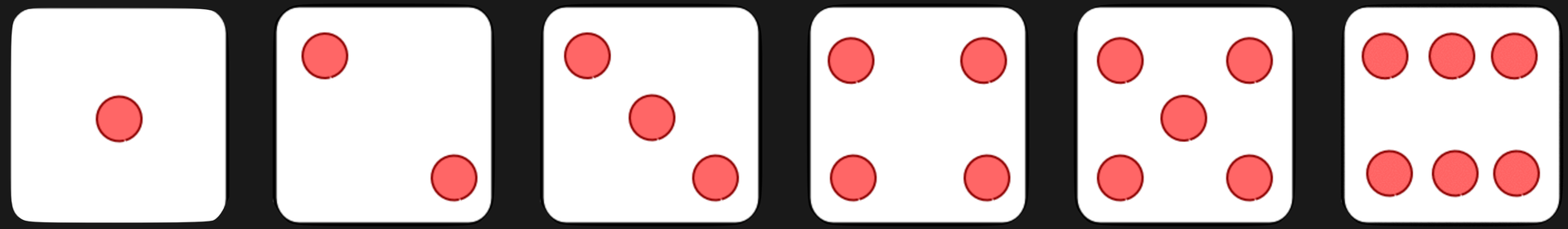
# UNIFORM DISTRIBUTION

A known finite number of outcomes are equally likely to occur

Each of *n* values has probability 1/*n*

Example: Rolling a fair die



MAKE SCHOOL

# SAMPLING WORDS

What distribution does this function use when sampling words?

```
colors = ['red', 'yellow', 'green', 'blue']

def sample(words):
    index = random.randint(0, len(words)-1)
    return words[index]

print(sample(colors))
```

Each word has *equal probability* of being selected

Therefore, words are sampled using *uniform distribution*

# WORD FREQUENCIES

How many distinct words are in a text sample?

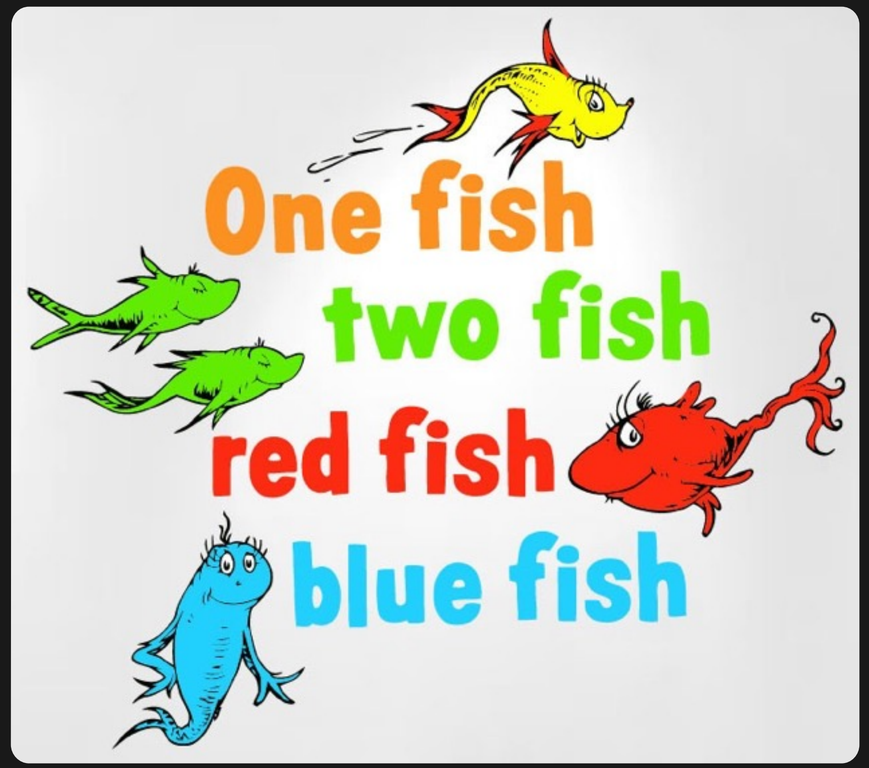
What are the frequencies of individual words?

We distinguish between *tokens* and *types*:

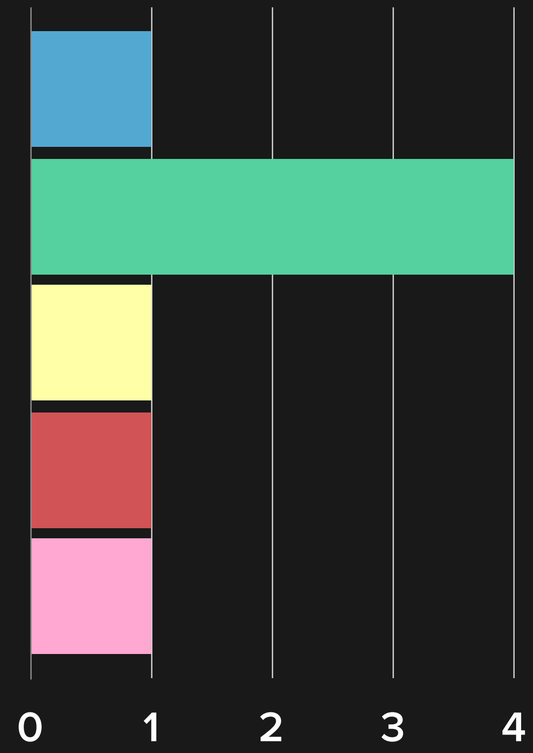
- **Tokens** – occurrences of words
- **Types** – distinct words

# FREQUENCY DISTRIBUTION

We need to count tokens in a word histogram




| word | count |
|---|---|
| blue | 1 |
| fish | 4 |
| one | 1 |
| red | 1 |
| two | 1 |




MAKE SCHOOL

# SAMPLING DISTRIBUTIONS

Observed word *tokens* have non-uniform distribution:

```
text = 'one fish two fish red fish blue fish'

word_counts = {'one': 1, 'fish': 4, 'two': 1,
               'red': 1, 'blue': 1}

def sample_by_frequency(histogram):
    # TODO: select a word based on frequency
    return word
```

How can you sample words using their *observed frequencies*?

# SAMPLING DISTRIBUTIONS

Ideas for how to sample using word frequencies:

- Duplicate words in the list by their multiplicity, then sample that list with uniform distribution
- Accumulate word counts through the list, then find where a uniform random number splits it
- Any other ideas? There are several ways...

# FUTURE DIRECTIONS

Collocations and *n*-grams

Conditional probability

Markov models and chains

Text generation and classification

Smoothing and back-off

Today's milestone – Tweet Generator tutorial page 4: "Stochastic Sampling": www.makeschool.com/academy

MAKE SCHOOL