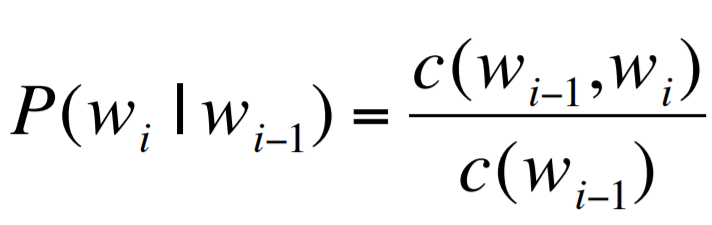
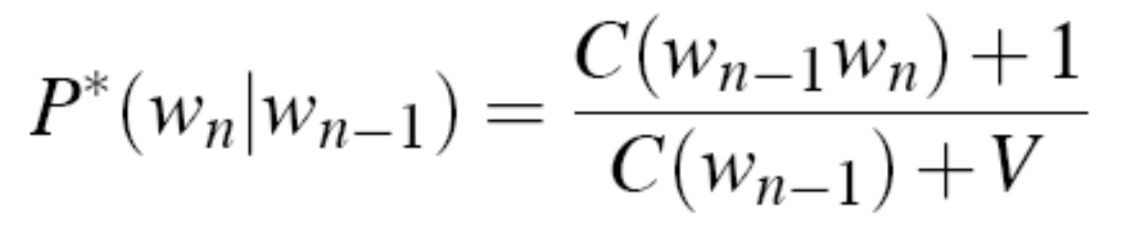
# 

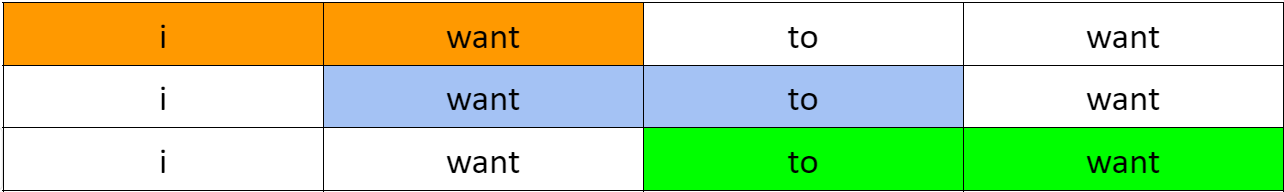
# Given the raw count tables below, compute the probability of sentences **a** and **b**,with and without Add-one (Laplace) smoothing:

* 1. “i want to want” P(a) = ?, P’(a)=?
  2. “i want to spend” P(b) =?, P’(b)=?

### **Raw unigram counts:**

Hint1: You should infer the value for *V* from the above table. 

### **Raw bigram counts:**



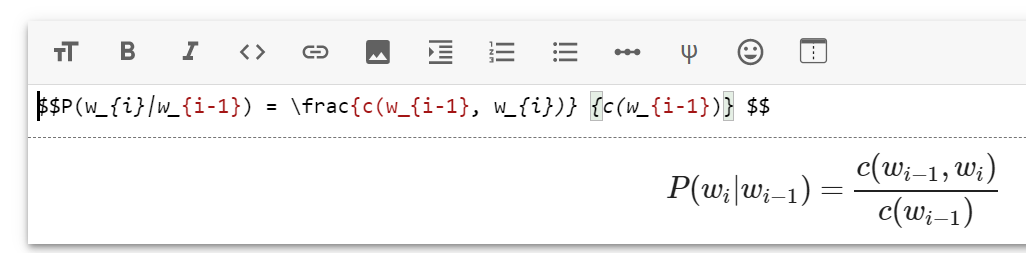
Hint2: You do not need to reconstitute counts in the above tables.

**show your work using LaTeX:** This is a good opportunity to learn and practice [LaTeX](https://en.wikipedia.org/wiki/LaTeX).

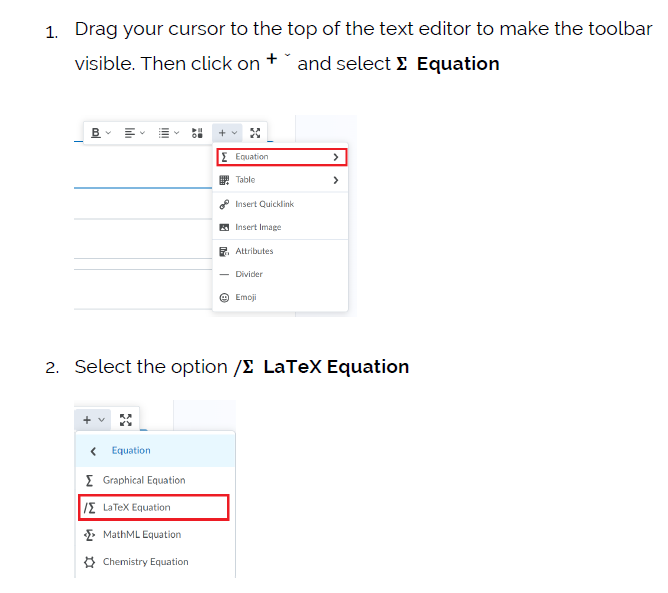
LaTeX is a useful skill to have during your studies and beyond.

* [Intro to LaTeX: Learn to write beautiful math equations](https://youtu.be/Jp0lPj2-DQA)
* [**Adding Latex to your notebook**](https://colab.research.google.com/github/bebi103a/bebi103a.github.io/blob/master/lessons/00/intro_to_latex.ipynb)

$$P(w*\_{i}|w\_*{i-1}) = \frac{c(w\_{i-1}, w*\_{i})} {c(w\_*{i-1})} $$



# [Guide for LaTeX in Brightspace](https://smustudio.squarespace.com/s/Guide-for-LaTeX-in-Brightspace.pdf)



# How to avoid underflow?

What’s the smallest/largest number in Python?

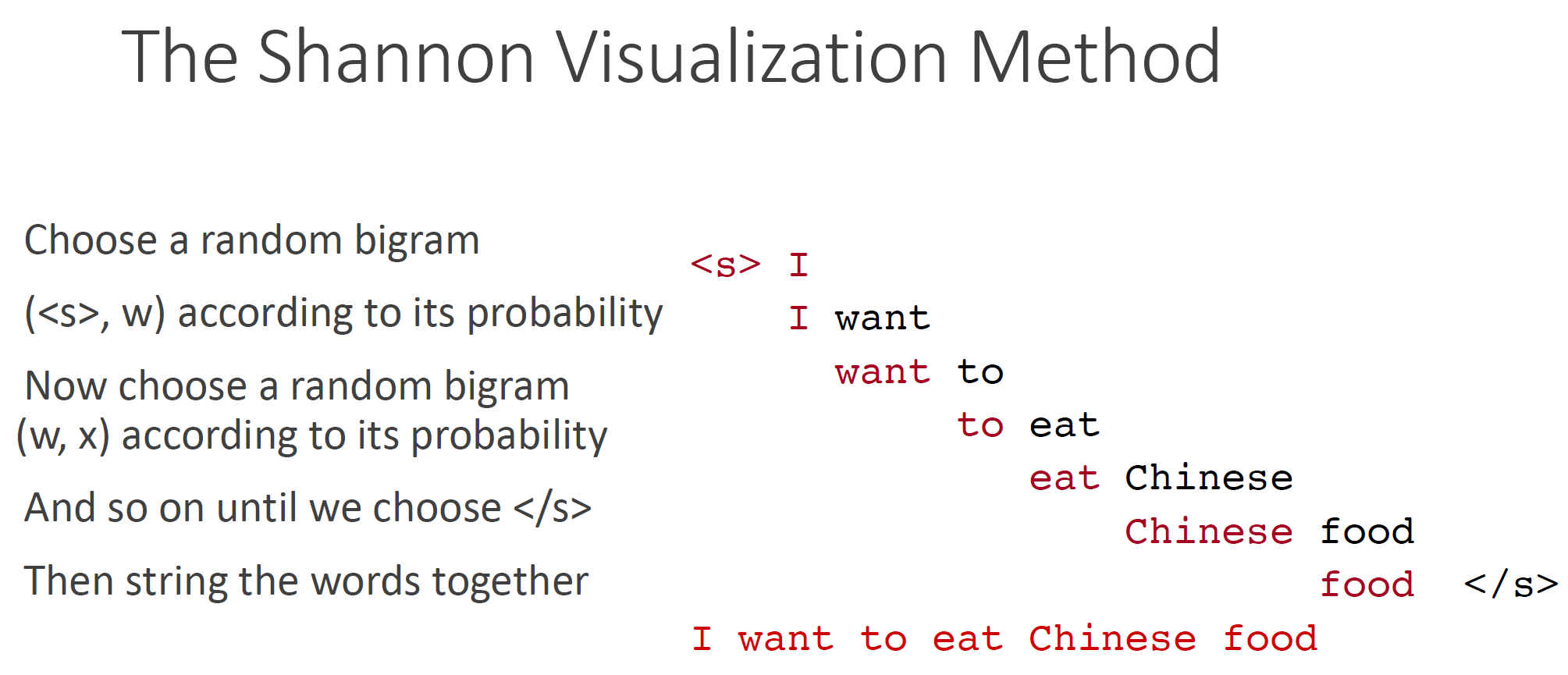
import sys

min = sys.float\_info.min

What happens if we multiply it by itself?

**Sampling sentences from a Bigram language model:**

[Chapter 3, section 3.3](http://web.stanford.edu/~jurafsky/slp3/3.pdf) gives a detailed description of the Shannon Visualisation method



# Consider building an ***N***-gram language model guaranteed to be capable of predicting the correct word (**crashed**) at the end of this sentence:

# “the **computer** which I had just put into the machine room on the fifth floor **crashed**”

# , what would be the minimum value of ***N*** required to achieve this?

# Explain why unigrams (1-gram) are bad at the Shannon Game