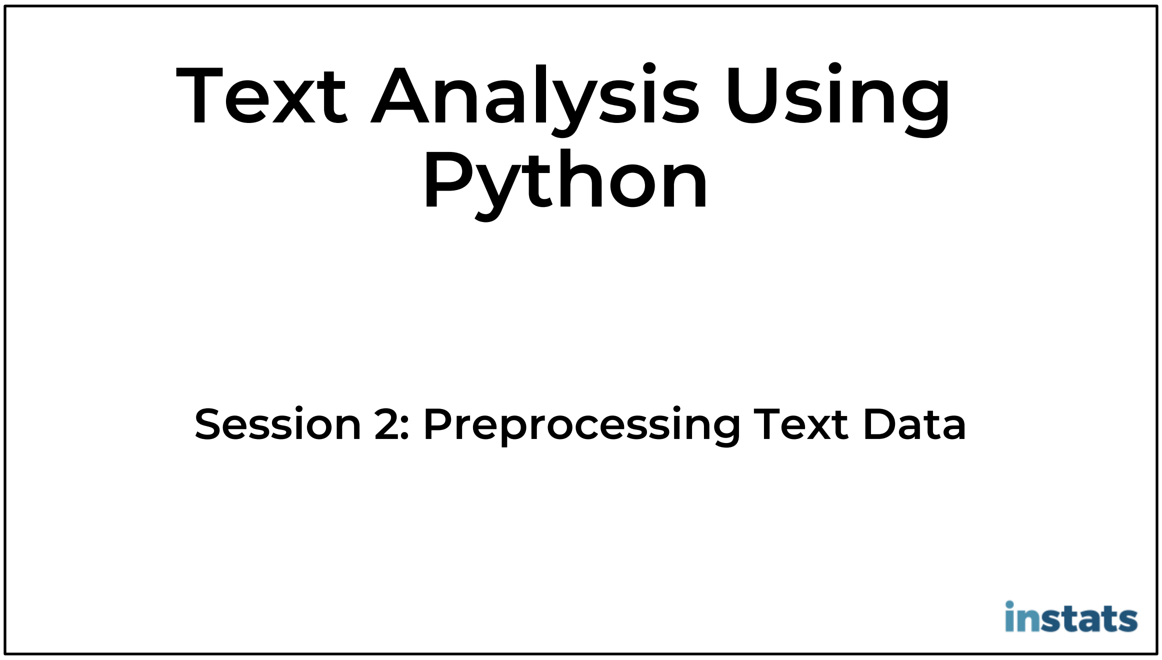
Welcome to instats

The Session Will Begin Shortly

START



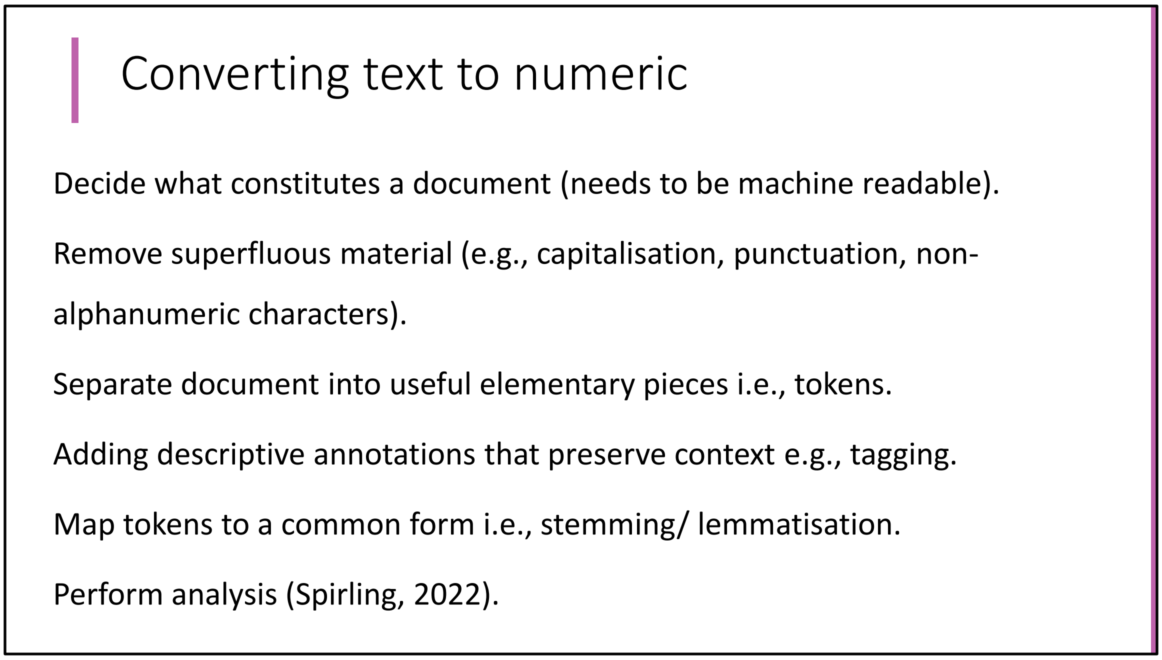
Text Analysis Using Python

Session 2: Preprocessing Text Data

instats

Outline

1. Converting text to numeric:
    1. Workflow
    2. Pre-processing
2. Simple representations of text:
    1. Bag of words
    2. DFM / DTM
3. Other considerations:
    1. Weighting
    2. N-grams



Converting text to numeric

Decide what constitutes a document (needs to be machine readable).

Remove superfluous material (e.g., capitalisation, punctuation, non-alphanumeric characters).

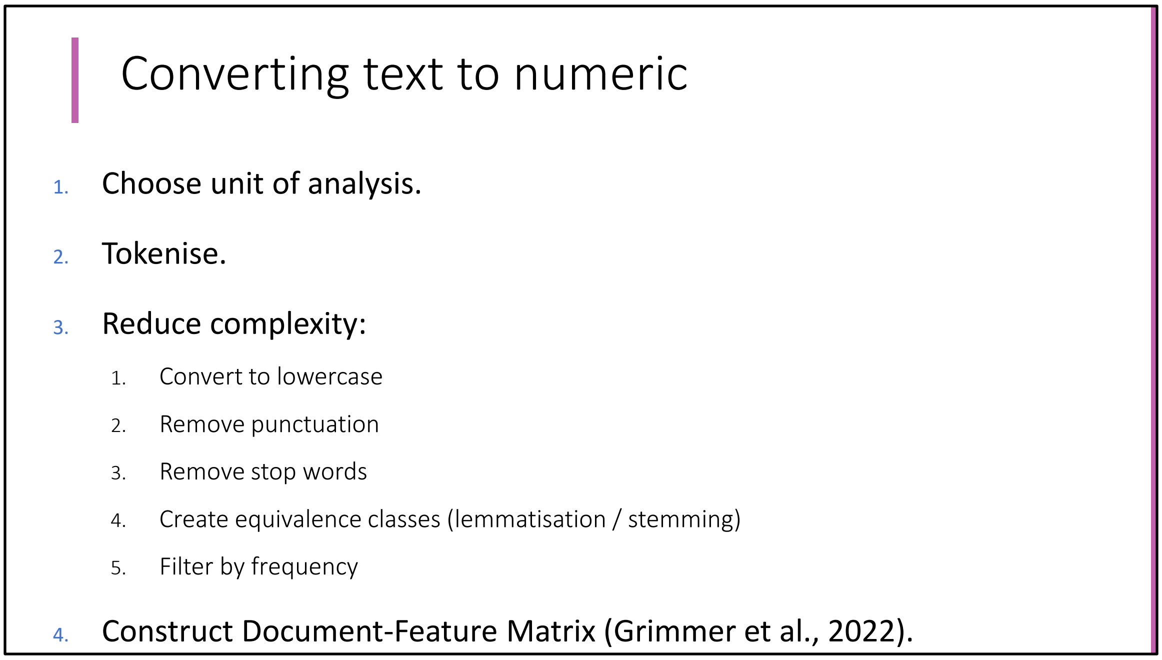
Separate document into useful elementary pieces i.e., tokens.

Adding descriptive annotations that preserve context e.g., tagging.

Map tokens to a common form i.e., stemming/ lemmatisation.

Perform analysis (Spirling, 2022).

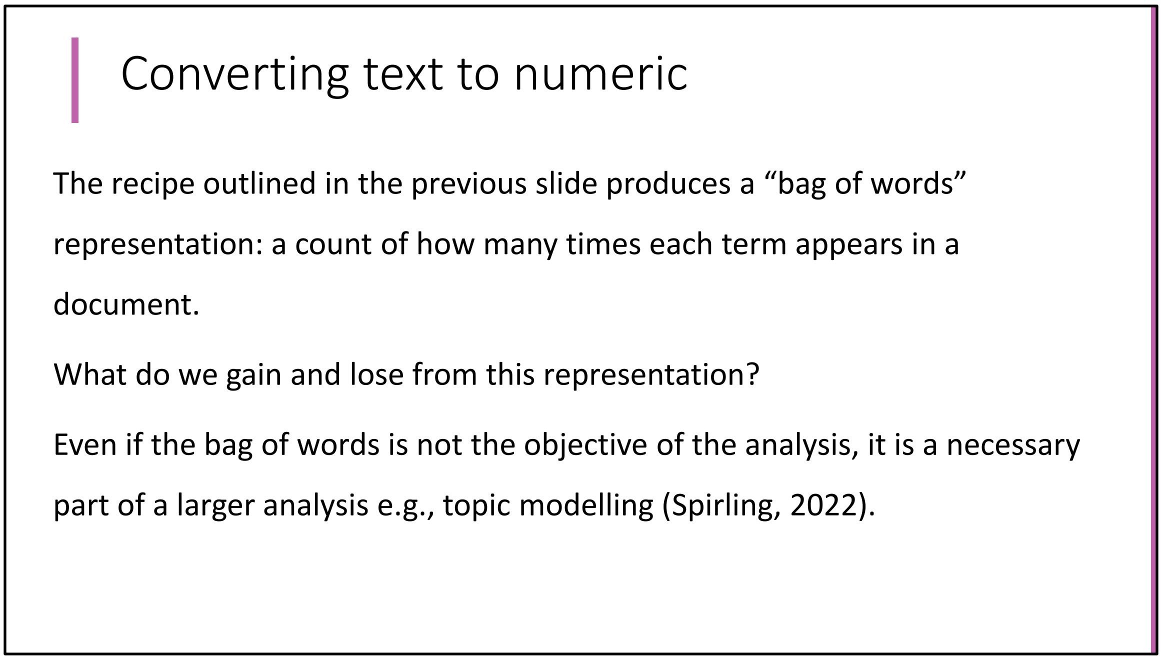
Not all of these steps are necessary for every use of text analysis, and what you do within each step may vary also (e.g., stemming vs lemmatisation). In addition, the order of these steps is your decision!



Converting text to numeric

1. Choose unit of analysis.
2. Tokenise.
3. Reduce complexity:
   1. Convert to lowercase
   2. Remove punctuation
   3. Remove stop words
   4. Create equivalence classes (lemmatisation / stemming)
   5. Filter by frequency
4. Construct Document-Feature Matrix (Grimmer et al., 2022).

This workflow or recipe is a more technical implementation of the general workflow proposed by Spirling (2022). The order also matters, as we will see in the practical. We will look at each of these steps in turn during the practical – for now let's focus on what this recipe produces.



Converting text to numeric

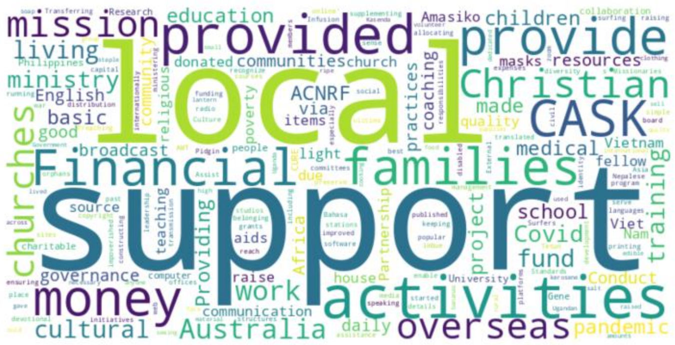
The recipe outlined in the previous slide produces a “bag of words” representation: a count of how many times each term appears in a document.

What do we gain and lose from this representation?

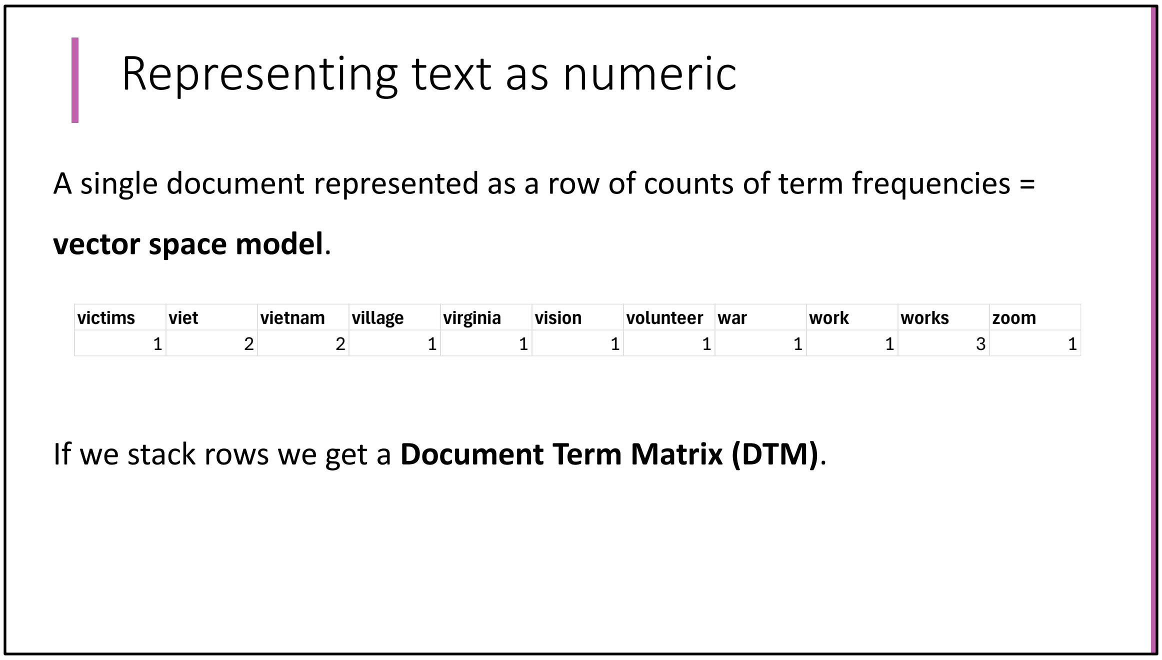
Even if the bag of words is not the objective of the analysis, it is a necessary part of a larger analysis e.g., topic modelling (Spirling, 2022).

Using the bag of words recipe means we are not interested in (or able to analyse) word order.

Representing text as numeric



Here is a word cloud visualisation of the bag of words representation of the activities of a sample of Australian charities operating overseas. What are your thoughts on the interpretation and insights of this approach?



Representing text as numeric

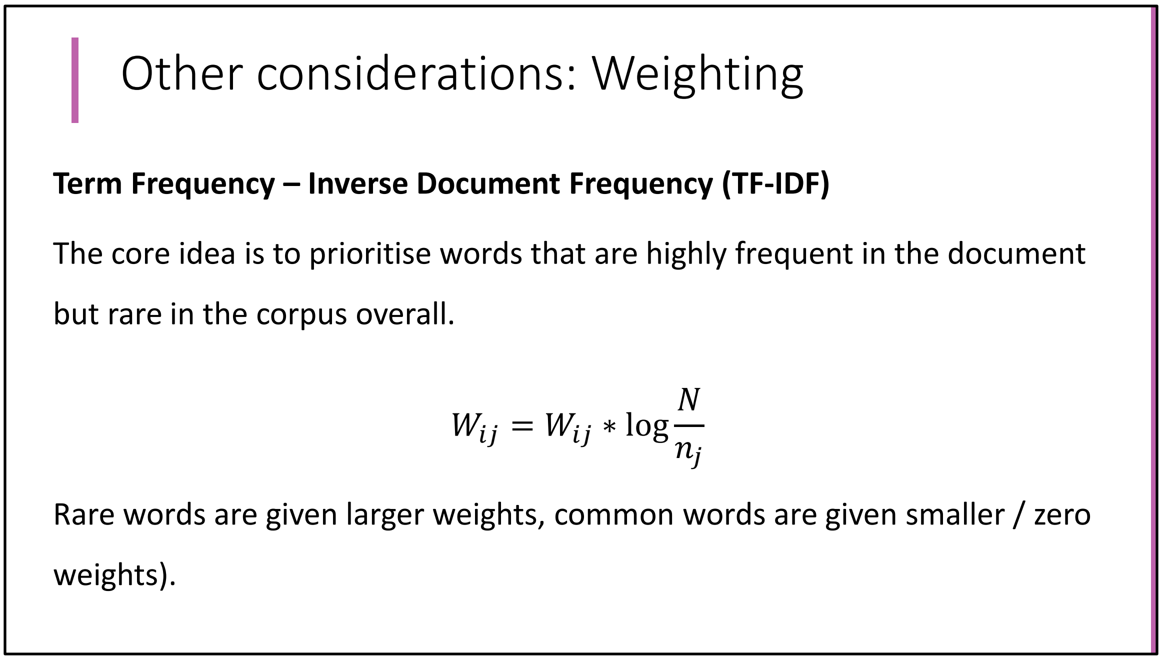
A single document represented as a row of counts of term frequencies = **vector space model**.

victims	viet	vietnam	village	virginia	vision	volunteer	war	work	works	zoom
1	2	2	1	1	1	1	1	1	3	1

If we stack rows we get a **Document Term Matrix (DTM)**.

A vector space model represents documents as a sequence of term counts. This representation is useful as we can not only compare the raw counts but also the sequence.

Word clouds and other visualisations are nice but the actual underpinning data structure / format is what's called a Document Term Matrix / Document Feature Matrix. The features of the document are the terms found in it and the cells capture counts of those terms in the document.



Other considerations: Weighting

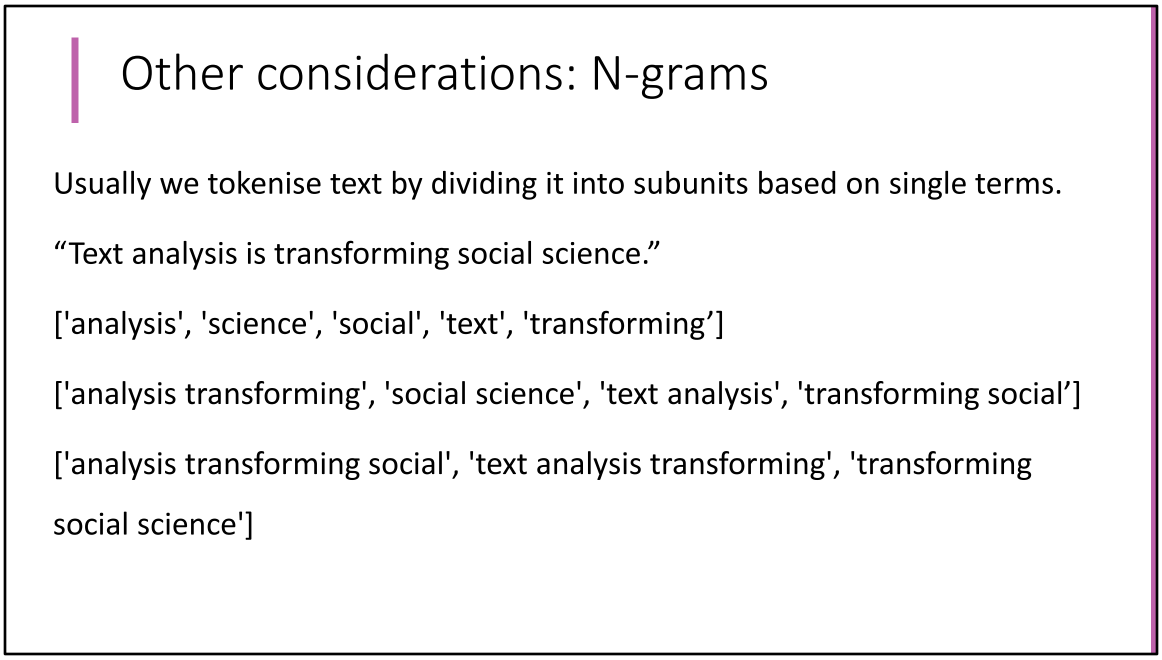
Term Frequency – Inverse Document Frequency (TF-IDF)

The core idea is to prioritise words that are highly frequent in the document but rare in the corpus overall.

$$W_{ij} = W_{ij} * \log \frac{N}{n_j}$$

Rare words are given larger weights, common words are given smaller / zero weights).

You may wonder why we adjust the raw counts of terms in the corpus? The idea is to aid our analysis by a) ensuring documents that use the same terms but in very different frequencies are still considered similar and b) improve our ability to discriminate between documents i.e., easier to identify documents that use particular words that are rare overall – good for information retrieval tasks.



Other considerations: N-grams

Usually we tokenise text by dividing it into subunits based on single terms.

“Text analysis is transforming social science.”

['analysis', 'science', 'social', 'text', 'transforming’]

['analysis transforming', 'social science', 'text analysis', 'transforming social’]

['analysis transforming social', 'text analysis transforming', 'transforming social science']

By tokenising text we create unigrams: each term is treated separately.

Bigrams and trigrams can often be often important depending on your analysis e.g., “social” and “science” will not be linked through a stem or lemma but they are substantively linked. The problem with bigrams and trigrams (and higher) is they can be computationally intensive to process.

When might you use bigrams and trigrams in your analysis? If you want to preserve word order or multiword terms.

STOP