# Stylometry with R
# aka
# Stylo in Galway

Computational Stylistics Group

Distant Reading

IJP PAN Institute of Polish Language, Polish Academy of Sciences

# About me

joanna.byszuk@ijp.pan.pl

Twitter: @jbyszuk

Notes from our tutorial: https://github.com/JoannaBy/Stylo-in-Galway

Computational Stylistics Group: https://computationalstylistics.github.io/

# Tell me about yourselves.

What’s your name?

Are you (why?) interested in stylometry (a specific problem)?

# What is stylometry?

Stylometry =

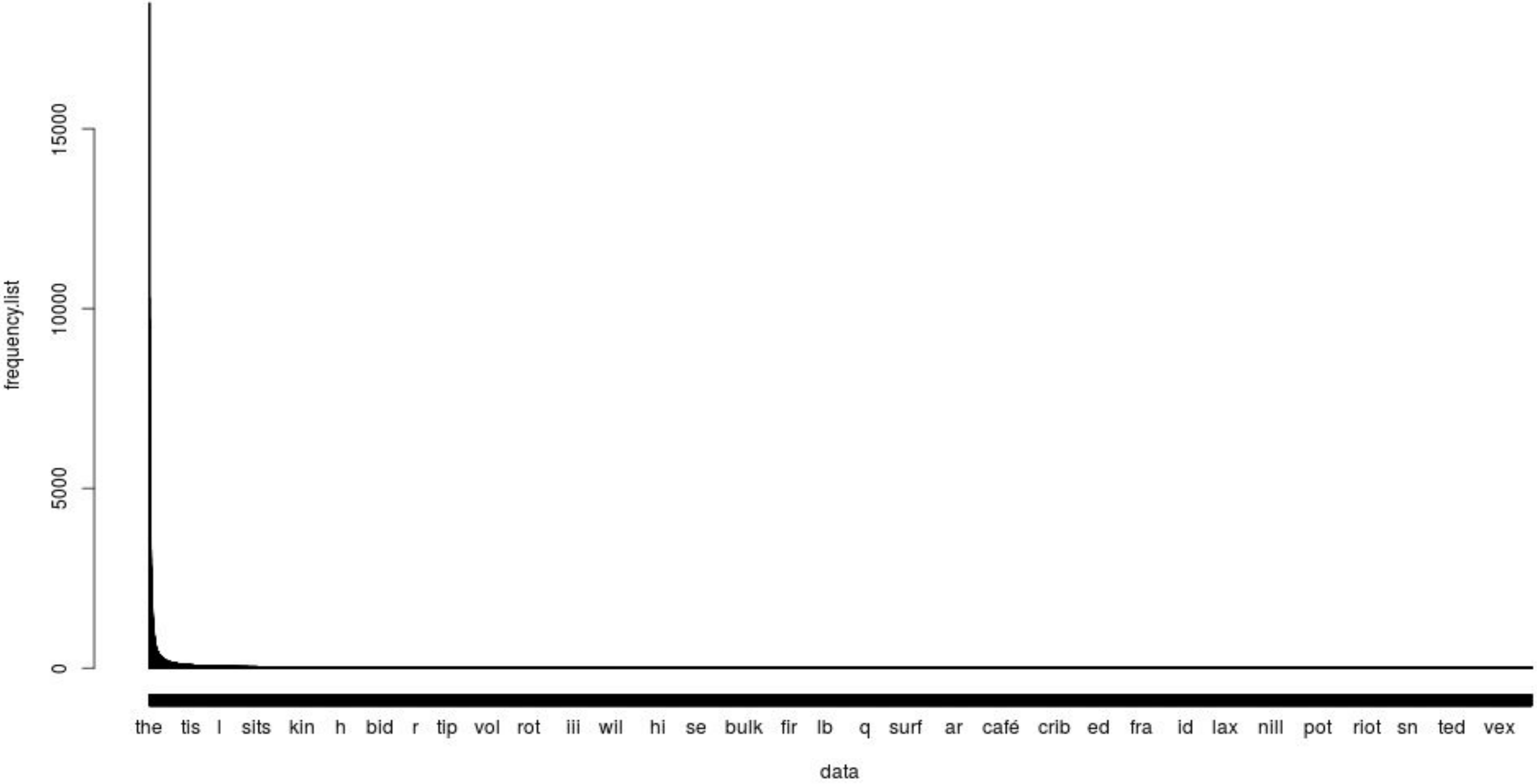
use of **quantitative methods**
to examine **similarities** and **differences**
within **a group of texts**

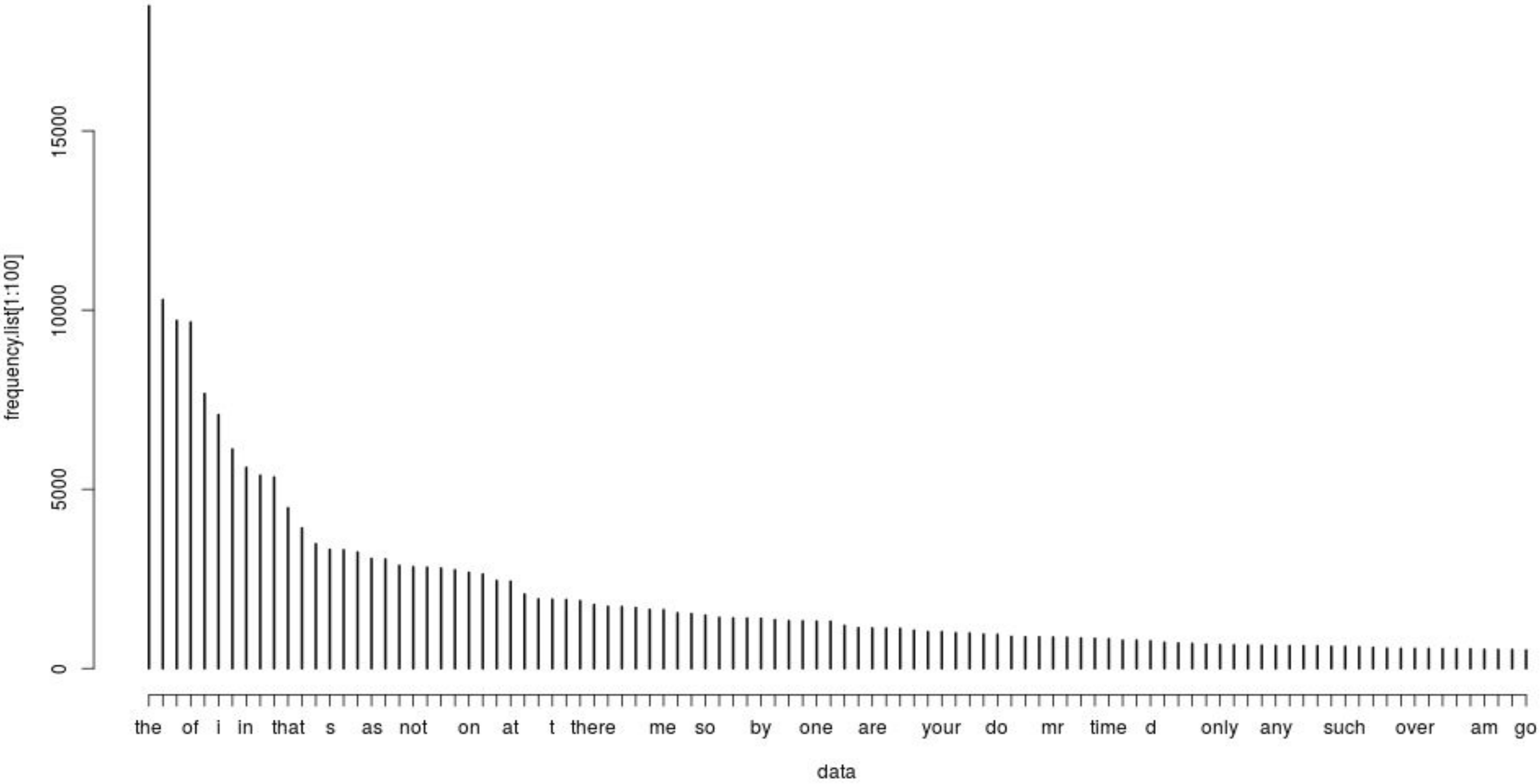
# What do we need stylometry for?

It is a truth NOT generally acknowledged that, **in most discussions of works of English fiction, we proceed as if a third, two-fifths, or half of our material were not really there...** That third, two-fifths, or half comprises the twenty, thirty or fifty most common words of [the] literary vocabulary. The identity of these words scarcely changes from novel to novel over the twenty years of her mature career. **Eight personal pronouns, six auxiliary verb forms, five prepositions, three conjunctions, two adverbs, the definite and indefinite articles, and four other words ('to', 'that', 'for', and 'all'),** each of which serves more than a single main grammatical function, almost always **find their place** - and usually about the same place - **among the thirty most common words of each novel.**

John Burrows, Computation into Criticism: A study of Jane Austen's Novels and an Experiment in Method, 1987

# A book or collection of them


frequency.list
0
5000
10000
15000
the
tis
l
sits
kin
h
bid
r
tip
vol
rot
iii
wil
hi
se
bulk
fir
lb
q
surf
ar
café
crib
ed
fra
id
lax
nill
pot
riot
sn
ted
vex
data


frequency.list[1:100]
0
5000
10000
15000
the
of
i
in
that
s
as
not
on
at
t
there
me
so
by
one
are
your
do
mr
time
d
only
any
such
over
am
go
data

**The possibility of using frequency patterns of very common words rests upon the fact that words do not function as discrete entities.** Since they gain their full meaning through the different sorts of relationship they form with each other, **they can be seen as markers** of those relationships and, accordingly, of everything that those relationships entail.

in: Wayne McKenna, John Burrows, Alexis Antonia (1999) Beckett's "Molloy": Computational Stylistics and the Meaning of Translation. Variété: Perspectives in French Literature, Society and Culture. Studies in Honour of Kenneth Raymond Dutton. Ed. Marie Ramsland. Peter Lang, Frankfurt, 79-92.

# What is used for?

Let’s see the experts present their work

# How does it work?

# What do we need?

corpus of texts

+

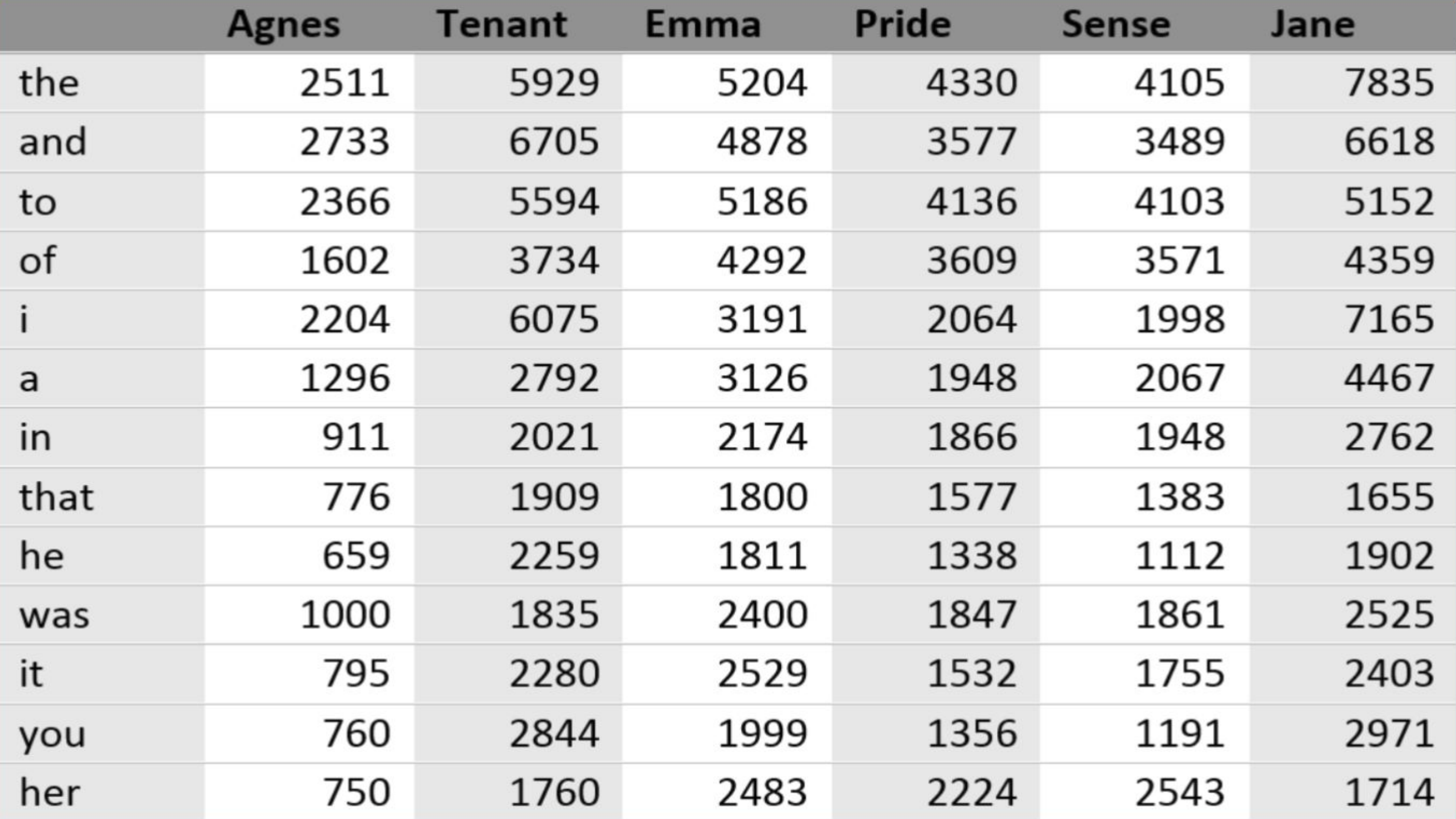
distance measure

+

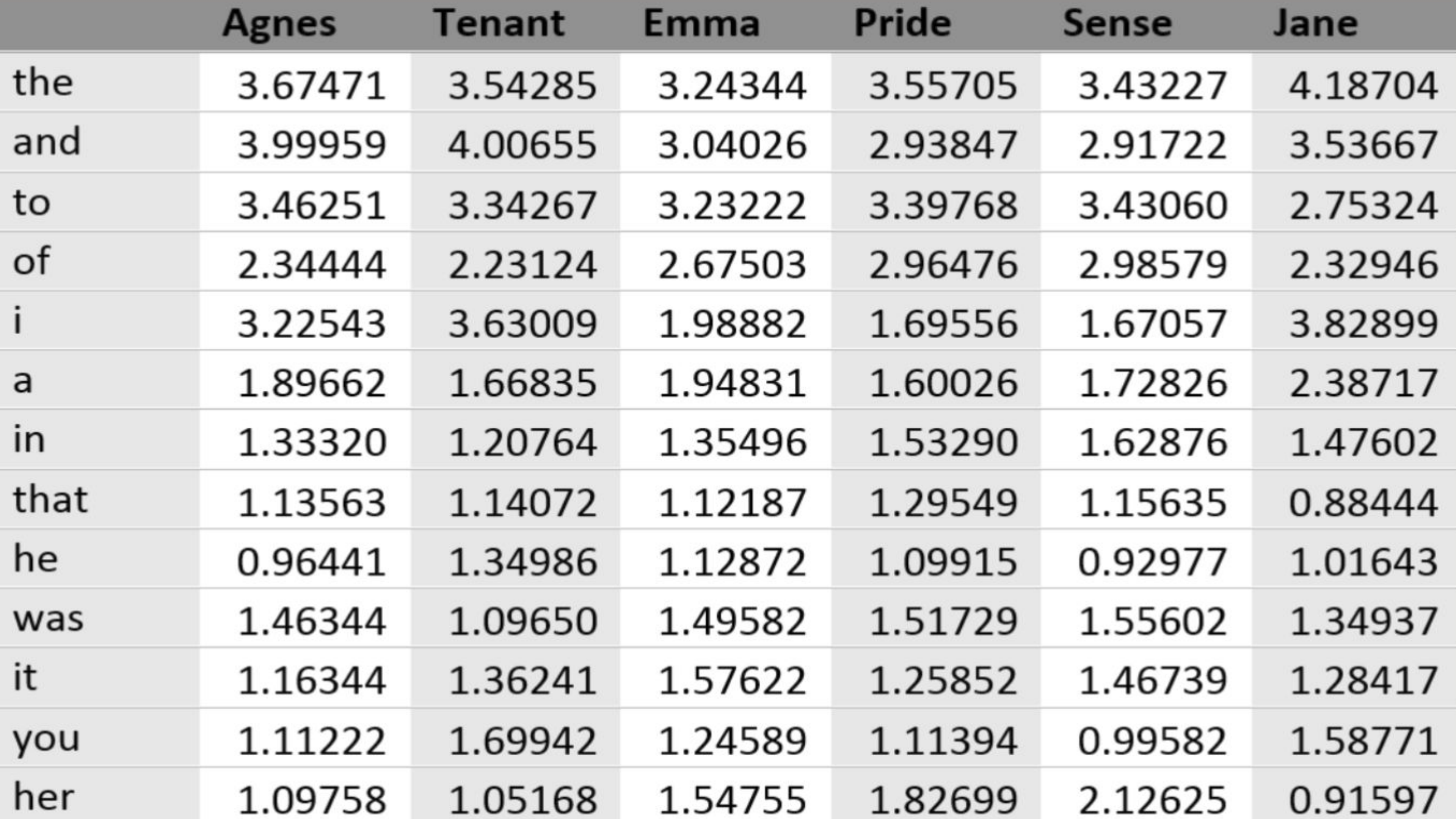
classification algorithm

+

(visualisation)



| | Agnes | Tenant | Emma | Pride | Sense | Jane |
|---|---|---|---|---|---|---|
| the | 2511 | 5929 | 5204 | 4330 | 4105 | 7835 |
| and | 2733 | 6705 | 4878 | 3577 | 3489 | 6618 |
| to | 2366 | 5594 | 5186 | 4136 | 4103 | 5152 |
| of | 1602 | 3734 | 4292 | 3609 | 3571 | 4359 |
| i | 2204 | 6075 | 3191 | 2064 | 1998 | 7165 |
| a | 1296 | 2792 | 3126 | 1948 | 2067 | 4467 |
| in | 911 | 2021 | 2174 | 1866 | 1948 | 2762 |
| that | 776 | 1909 | 1800 | 1577 | 1383 | 1655 |
| he | 659 | 2259 | 1811 | 1338 | 1112 | 1902 |
| was | 1000 | 1835 | 2400 | 1847 | 1861 | 2525 |
| it | 795 | 2280 | 2529 | 1532 | 1755 | 2403 |
| you | 760 | 2844 | 1999 | 1356 | 1191 | 2971 |
| her | 750 | 1760 | 2483 | 2224 | 2543 | 1714 |



| | Agnes | Tenant | Emma | Pride | Sense | Jane |
|---|---|---|---|---|---|---|
| the | 3.67471 | 3.54285 | 3.24344 | 3.55705 | 3.43227 | 4.18704 |
| and | 3.99959 | 4.00655 | 3.04026 | 2.93847 | 2.91722 | 3.53667 |
| to | 3.46251 | 3.34267 | 3.23222 | 3.39768 | 3.43060 | 2.75324 |
| of | 2.34444 | 2.23124 | 2.67503 | 2.96476 | 2.98579 | 2.32946 |
| i | 3.22543 | 3.63009 | 1.98882 | 1.69556 | 1.67057 | 3.82899 |
| a | 1.89662 | 1.66835 | 1.94831 | 1.60026 | 1.72826 | 2.38717 |
| in | 1.33320 | 1.20764 | 1.35496 | 1.53290 | 1.62876 | 1.47602 |
| that | 1.13563 | 1.14072 | 1.12187 | 1.29549 | 1.15635 | 0.88444 |
| he | 0.96441 | 1.34986 | 1.12872 | 1.09915 | 0.92977 | 1.01643 |
| was | 1.46344 | 1.09650 | 1.49582 | 1.51729 | 1.55602 | 1.34937 |
| it | 1.16344 | 1.36241 | 1.57622 | 1.25852 | 1.46739 | 1.28417 |
| you | 1.11222 | 1.69942 | 1.24589 | 1.11394 | 0.99582 | 1.58771 |
| her | 1.09758 | 1.05168 | 1.54755 | 1.82699 | 2.12625 | 0.91597 |

For two texts *T* and *T1*, and for a set of *n* words,

$$\Delta(T,T_1)=\frac{1}{n}\sum_{i=1}^{n}\left|z(f_i(T))-z(f_i(T_1))\right|$$

Where $z(f_x(T))=\dfrac{f_x(T)-\mu_x}{\sigma_x}$;

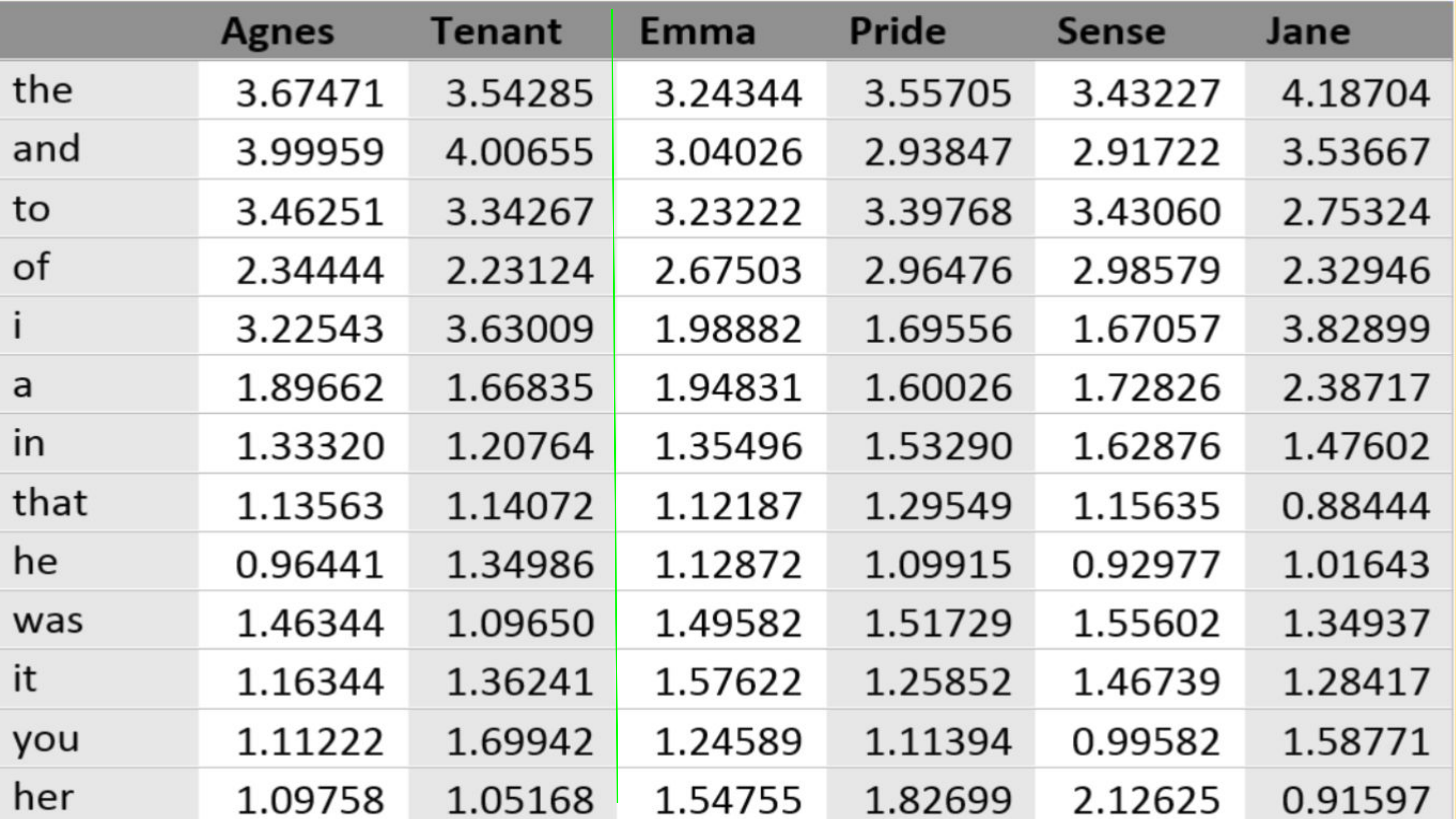
$f_x(T)$ = raw frequency of word *x* in text T;

$\mu_x$ = mean frequency of word *x* in a collection of texts;

$\sigma_x$ = standard deviation of frequency of word *x*.

= what’s the difference in how two texts use a given feature, compared to its average use

E.g. ...

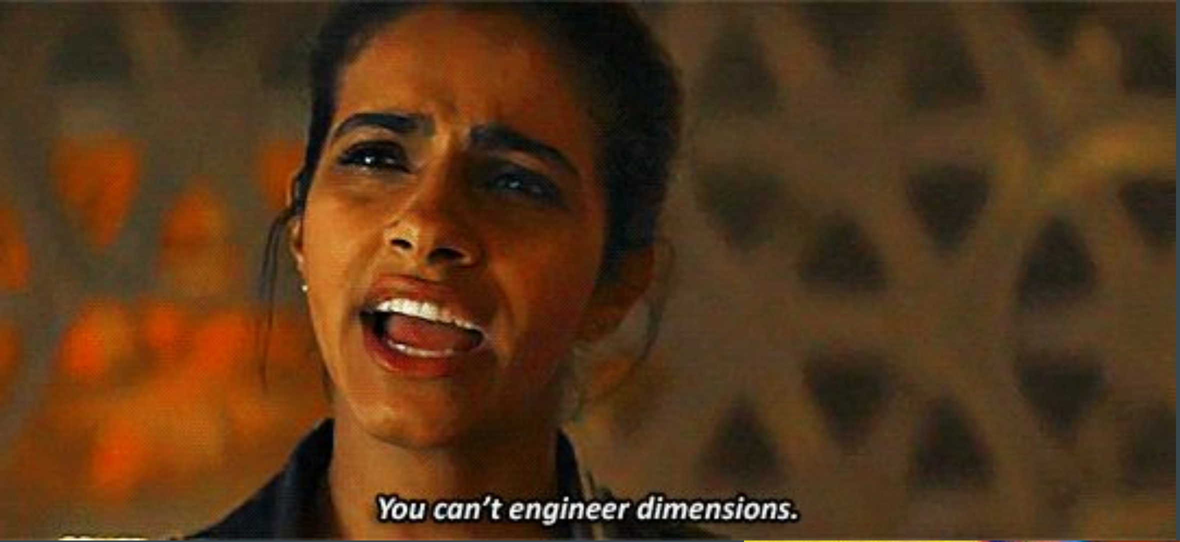


| | Agnes | Tenant | Emma | Pride | Sense | Jane |
|---|---|---|---|---|---|---|
| the | 3.67471 | 3.54285 | 3.24344 | 3.55705 | 3.43227 | 4.18704 |
| and | 3.99959 | 4.00655 | 3.04026 | 2.93847 | 2.91722 | 3.53667 |
| to | 3.46251 | 3.34267 | 3.23222 | 3.39768 | 3.43060 | 2.75324 |
| of | 2.34444 | 2.23124 | 2.67503 | 2.96476 | 2.98579 | 2.32946 |
| i | 3.22543 | 3.63009 | 1.98882 | 1.69556 | 1.67057 | 3.82899 |
| a | 1.89662 | 1.66835 | 1.94831 | 1.60026 | 1.72826 | 2.38717 |
| in | 1.33320 | 1.20764 | 1.35496 | 1.53290 | 1.62876 | 1.47602 |
| that | 1.13563 | 1.14072 | 1.12187 | 1.29549 | 1.15635 | 0.88444 |
| he | 0.96441 | 1.34986 | 1.12872 | 1.09915 | 0.92977 | 1.01643 |
| was | 1.46344 | 1.09650 | 1.49582 | 1.51729 | 1.55602 | 1.34937 |
| it | 1.16344 | 1.36241 | 1.57622 | 1.25852 | 1.46739 | 1.28417 |
| you | 1.11222 | 1.69942 | 1.24589 | 1.11394 | 0.99582 | 1.58771 |
| her | 1.09758 | 1.05168 | 1.54755 | 1.82699 | 2.12625 | 0.91597 |

|  | Agnes | Pride | Jane | David | Mill | Tom | Clarissa |
|---|---|---|---|---|---|---|---|
| Tenant | 0.81 | 1.07 | 0.88 | 0.92 | 0.98 | 1.16 | 1.1 |
| Emma | 1.12 | 0.78 | 1.28 | 1.15 | 1.2 | 1.25 | 1.24 |
| Sense | 1.14 | 0.69 | 1.24 | 1.16 | 1.25 | 1.13 | 1.21 |
| Professor | 1.06 | 1.21 | 0.69 | 0.94 | 1 | 1.27 | 1.3 |
| Villette | 1.07 | 1.26 | 0.65 | 0.91 | 0.96 | 1.28 | 1.3 |
| Bleak | 1.09 | 1.18 | 0.92 | 0.55 | 0.87 | 1.21 | 1.17 |
| Hard | 1.16 | 1.25 | 0.96 | 0.65 | 0.91 | 1.26 | 1.25 |
| Wuthering | 1.06 | 1.31 | 0.81 | 0.94 | 1.01 | 1.32 | 1.27 |
| Adam | 1.13 | 1.37 | 0.95 | 0.9 | 0.66 | 1.42 | 1.32 |
| Middlemarch | 1.01 | 1.1 | 0.99 | 0.87 | 0.65 | 1.17 | 1.12 |
| Joseph | 1.2 | 1.19 | 1.24 | 1.18 | 1.29 | 0.64 | 1.11 |
| Pamela | 1.15 | 1.24 | 1.27 | 1.19 | 1.26 | 1.11 | 0.67 |
| Sentimental | 1.38 | 1.53 | 1.23 | 1.22 | 1.29 | 1.42 | 1.38 |

# Cool, but how to actually analyse this?

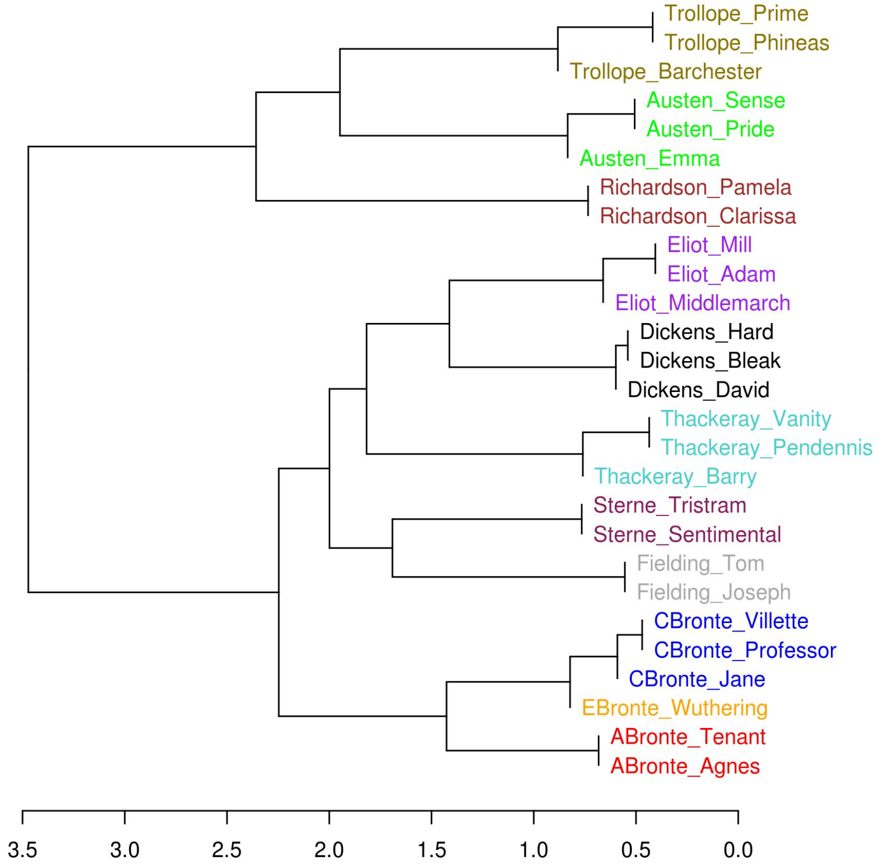
Answer: dimension reduction


You can’t engineer dimensions.

Maybe *you* can’t.

# #1 Cluster analysis

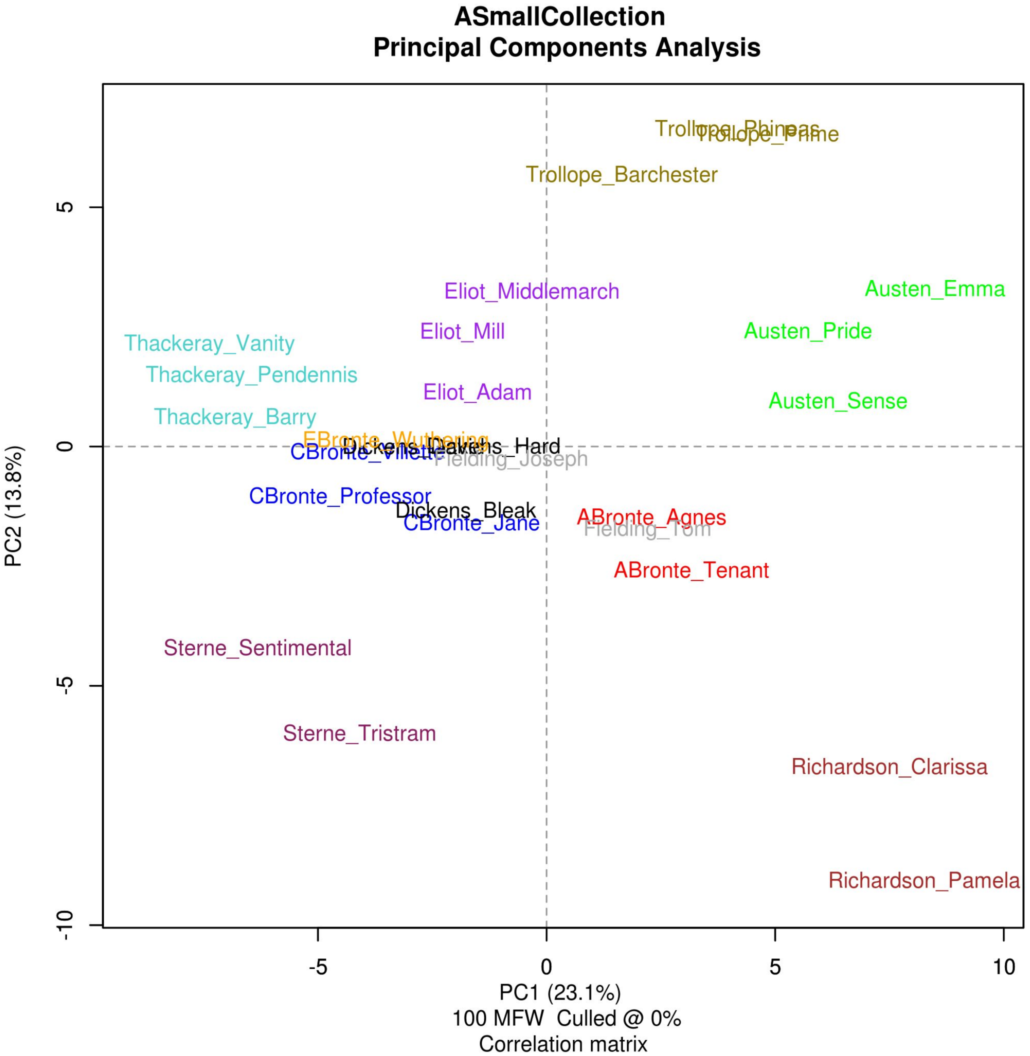
## ASmallCollection
## Cluster Analysis



100 MFW Culled @ 0%
Classic Delta distance

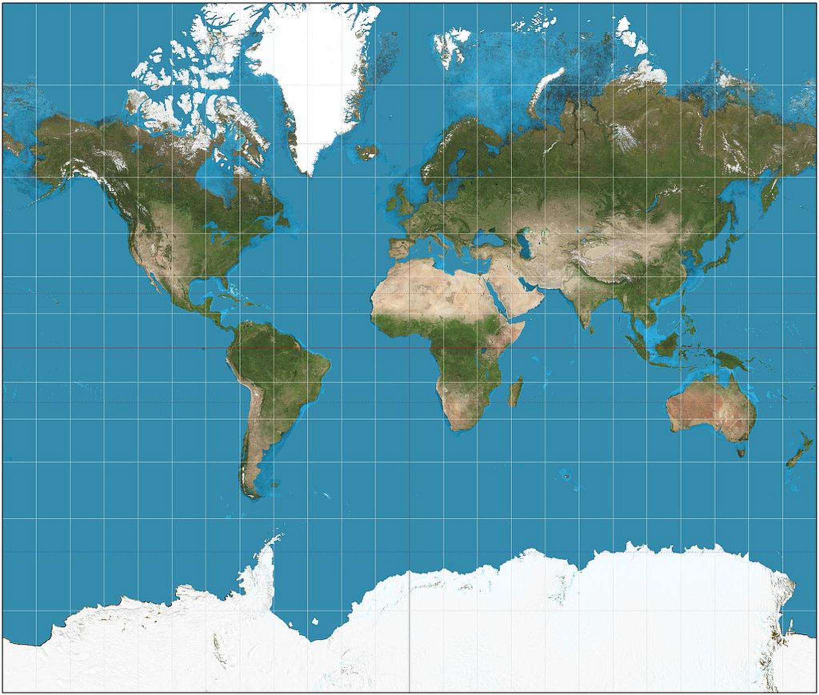
# #2 PCA

# PCA – Principal Component Analysis



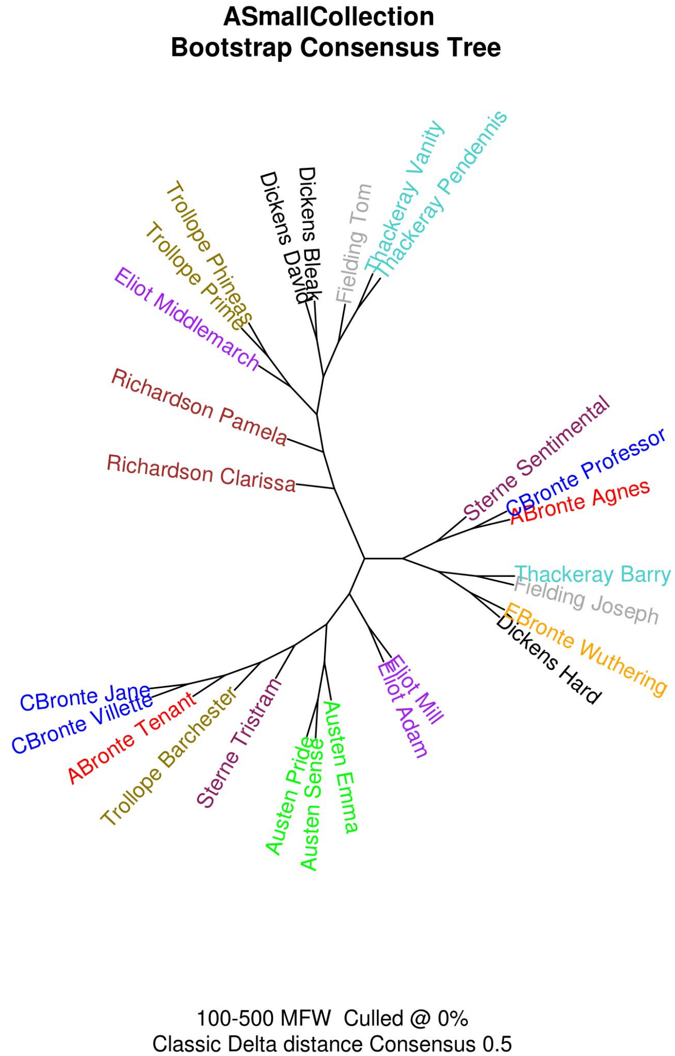
# Problems with dimension reduction

# Information loss

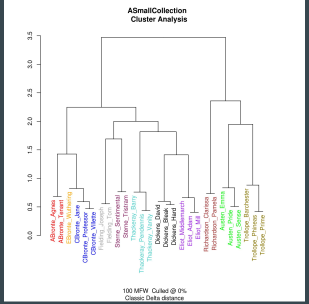


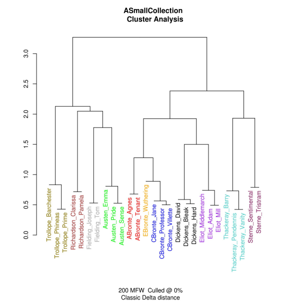
# Gold standard

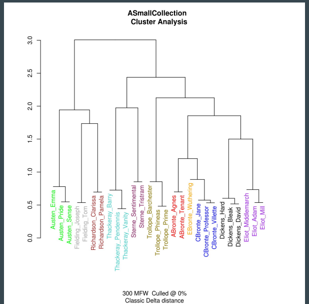
Bootstrap consensus tree

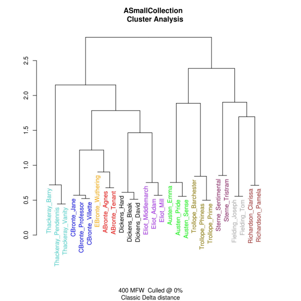


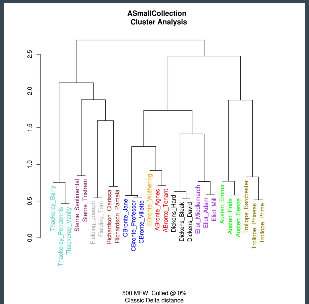
# Bootstrap consensus tree – how

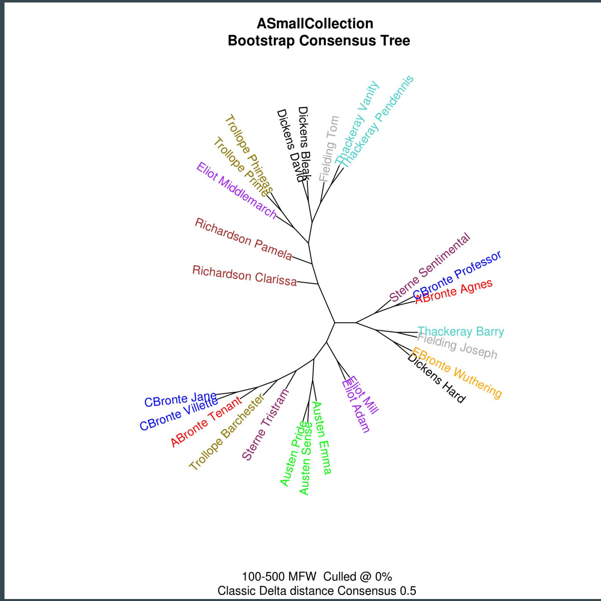












Let’s try to do that!

https://computationalstylistics.github.io/stylo_nutshell/#main-functions-stylo

# Getting started

## Set working directory:

Type: **setwd("the/path/to/my/favourite/folder")**

*OR:*

RStudio users: find your directory in the Files panel, then use ***Menu > More > Set as Working Directory***

Windows users: use ***Menu > File > Change directory***

## NEXT:

Type: **library(stylo)**

And then:

```
stylo()
```

# Stylometry for literary explorations

# Cluster analysis and network examinations

For this part follow the exercises. Basic interpretation tips:

In cluster analysis:

- distance = 0 means the same text in two files, the closer to 0 the texts get on one branch, the more similarities they share
- The distance is relative for the corpus. If you compare texts similar in topic and genre they will get low maximum distance (value on the left), if you throw in programming manual and a cookbook to poetry the distance will skyrocket.

In networks:

- the proximity of the nodes can be manipulated with (e.g. when you try to move overlapping nodes). Trust thickness of the edges. Thicker edges mean more shared similarities between the nodes.

# Authorship attribution

| Authorship attribution | vs | Authorship verification |
|---|---|---|
| We know the author must be one of a few candidates | | We don’t know if we have the author in our dataset |
| Relatively easy | | Still quite difficult |
| e.g. classify(), rolling.classify() | | e.g. imposters() |

That’s cool, but I don’t study English…

# More complex, but not a problem

- Features – words vs character n-grams – the second better for languages not using Latin or Cyrillic characters
- To lemmatize or not? – there is no definite answer
- Might need more data!
- Take care of your diacritics and encoding – your files should be encoded in UTF-8 which guarantees they are represented properly.