STYLOmetry

The art of measuring style
Agenda

- Software installation
- Why to analyze text?
- History of stylometry?
- What is STYLO?
- STYLO()
- CLASSIFY()
- ROLLING.CLASSIFY()
- Way Forward
About me

• Data science with math background
• Hiking
• Psychology
• People
• yoga

LinkedIn: https://www.linkedin.com/in/janidziak/
Count number of F’s

FINISHED FILES ARE THE RESULT OF YEARS OF SCIENTIFIC STUDY COMBINED WITH THE EXPERIENCE OF YEARS.
Count number of F’s

FINISHED FILES ARE THE RESULT OF YEARS OF SCIENTIFIC STUDY COMBINED WITH THE EXPERIENCE OF YEARS.
Environment preparation
Install R

https://cran.r-project.org/bin/windows/base
Install RStudio

www.rstudio.com/products/rstudio/download
RStudio

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.

Type 'q()' to quit R.
Create project directory

File >> New Project… >> New Directory >> Empty Project
Project Name

Stylo_workshop
Create script

File >> New File >> R Script
File >> Save >> Package_installation
Install packages

install.packages("stylo")
install.packages("ggplot2")
install.packages("Rtsne")
install.packages("networkD3")
install.packages("plotly")
install.packages("dplyr")
install.packages("FNN")
SHORT HISTORY OF STYLOMERTY
Lorenzo Valla (c. 1407–1457)
Constantini Donatione declamatio

NOT AUTHENTIC ?
Constantini Donatione declamatio

- A forged decree where the emperor Constantine I transfers authority of the Roman Empire to the Pope
- Written probably in the 8th century, claimed to be authentic for many centuries
- The first instance of scholarly-based investigation of style:
- Some grammatical forms could not have been used in the 4th century
Wincenty Lutosławski (1863–1954)
Wincenty Lutosławski

• 1890: chronology of Plato’s dialogues
• The basics of stylometry were set out by Polish philosopher Wincenty Lutosławski in *Principes de stylométrie* (1890)

• Pionier of yoga in Poland
Burrows DELTA

Frequencies of 100 – 5,000 most frequent words (MFW) form a “fingerprint” of an author’s style
Standardized to z-scores to give each word equal weight

\[
z(\text{Madding Crowd}) = (0.53, -0.23, -0.32, 0.20, 1.66, -0.37, 1.04, 0.52, -0.44, -0.92, 0.03, \ldots)
\]
\[
z(\text{Tess of the d’U.}) = (0.75, -0.48, -0.08, 0.51, -0.24, -0.87, 0.60, 0.41, -0.14, -0.47, 1.39, \ldots)
\]
\[
z(\text{Oliver Twist}) = (1.05, 0.15, -0.71, -0.56, 0.37, -1.01, -0.06, -0.74, -0.28, 0.48, -0.94, \ldots)
\]
Z Score

- $X$ - frequency of term
- $\text{mean}(X)$ - mean frequency of term
- $S$ - standard deviation

$$Z = \frac{X - \bar{X}}{S}$$
DELTA measures

Burrows’s Delta = Manhattan distance (Burrows 2002)

\[ \Delta_B(D, D') = \|z(D) - z(D')\|_1 = \sum_{i=1}^{n_w} |z_i(D) - z_i(D')| \]

Quadratic Delta = Euclidean distance (Argamon 2008)

\[ \Delta_Q(D, D') = \|z(D) - z(D')\|_2^2 = \sum_{i=1}^{n_w} (z_i(D) - z_i(D'))^2 \]
Distances

- Time
- Manhattan
- Euclidean
Stylo R software
Computational Stylistics Group

A cross-institutional research team focused on computer-assisted text analysis.

Computational Stylistics Group is a cross-institutional research team focused on computer-assisted text analysis, stylometry, authorship attribution, sentiment analysis, and the like stuff. The research projects conducted by the team members could be described as an intersection of linguistics, literary criticism, and computer sciences – however the best name here would be “Digital Humanities”. The group is based mostly in Kraków, at the Institute of Polish Language (Polish Academy of Sciences), but also at the Jagiellonian University and the University of Antwerp.

Even if the Group has been involved in several research projects (some of them are listed on this website, on the Projects subpage), it is probably known – at the first place – for the R package stylo, which is a comprehensive collection of functions written in the programming language R, for performing a variety of experiments in computational stylistics. More information about the package can be found here. Also, please check the discussion list dedicated to various issues in stylometry and beyond.
Why to analyze text?
Applications of stylometry

Digital Humanities
➢ Author attribution identification of unknown authors
➢ Genre classification
➢ Historical study of language change

Other applications
➢ Anonymity
➢ Plagiarism
➢ Criminal civil security- (detection if someone's suicide note is real)
Agatha Christie

5 September 1890 – 12 January 1976

66 detective novels and 14 short story collections from 1971 to 1974, Christie’s health began to fail...

...although she continued to write.
Four indefinite nouns

- thing(s),
- something,
- anything,
- nothing
Number of indefinite noun occurrences

[Graph showing the relationship between age and the number of indefinite noun occurrences for different authors, including Murdoch, Christie, "Frankfurt", and James.]
High-frequency verbs of low specificity

be, come, do, get, give, go, have, know, look, make, see, tell, think, want, ask, feel, find, forget, happen, hear, like, live, mean, meet, put, remember, run, say, seem, speak, suppose, take, use, walk, wonder
Proportion of 35 high-frequency verbs
Who is Robert Galbraith?

“The Cuckoo’s Calling reminds me why I fell in love with crime fiction in the first place”

Val McDermid

ROBERT GALBRAITH

THE CUCKOO’S CALLING
J. K. Rowling . . .
SIRI

What can I help you with?
How to use stylo?

```r
library(stylo)
classify()
```
FUNCTIONS PROVIDED

• **stylo()** - an all-in-one tool for a variety of experiments in computational stylistics.

• **classify()** - Function that performs a number of machine-learning methods for classification used in computational stylistics.

• **rolling.classify()** - Function that splits a text into equal-sized consecutive blocks (slices) and performs a supervised classification of these blocks against a training
Analysis Workflow

1. Data preparation
   a. Naming Convention
   b. Directory
2. Language and Input Formatting
3. Text Characteristics (Features)
4. Summary Type (Statistics)
5. Output
Computational Stylistics Group

A cross-institutional research team focused on computer-assisted text analysis.

Computational Stylistics Group is a cross-institutional research team focused on computer-assisted text analysis, stylometry, authorship attribution, sentiment analysis, and the like stuff. The research projects conducted by the team members could be described as an intersection of linguistics, literary criticism, and computer sciences – however the best name here would be “Digital Humanities”. The group is based mostly in Kraków, at the Institute of Polish Language (Polish Academy of Sciences), but also at the Jagiellonian University and the University of Antwerp.

Even if the Group has been involved in several research projects (some of them are listed on this website, on the Projects subpage), it is probably known – at the first place – for the R package stylo, which is a comprehensive collection of functions written in the programming language R, for performing a variety of experiments in computational stylistics. More information about the package can be found here. Also, please check the discussion list dedicated to various issues in stylometry and beyond.
Data preparation

Resources

Materials prepared by the Group. (More to be added soon...).

Corpora

The following selection of links is but a tip of an iceberg when it comes to the corpora (text collections) suitable for text analysis. The corpora listed below, however, are compiled by the members of CSG, and checked for compatibility with commonly known stylometric software.

- A Small Collection of British Fiction
- 100 Polish Novels
- 100 English Novels
- 68 German Novels
Data preparation

A selection of 28 classic British novels from the 19th century (including a few late 18th-century items). Full text versions, in plain text format, harvested from trustworthy public domain sites.
Data preparation

- https://github.com/computationalstylistics/100_english_novels
- https://github.com/computationalstylistics/A_Small_Collection_of_British_Fiction
Exercise - project preparation

- Create new project for the Exercises
- Name the project Stylometry_Exercise
- Create script with named exercise.R
library(stylo)

stylo()
What is Stylo?

Stylo: Comprehensive collection of functions written in the programming language R, for performing variety of experiments in computational stylistics.
Data preparation for Stylo()
Corpus preparation - labels

Colors on graphs are assigned according to filenames: the sequence of letters before “ - ” (underscore) is assumed to be the label of the author (genre, etc.).

- ABronte_Agnes.txt
- Austen_Emma.txt
- Austen_Pride.txt
- Conrad_Lord.txt
- Dickens_Pickwick.txt
- ...

...
LIVE DEMO

```r
library(stylo)
stylo()
```
INPUT & LANGUAGE

[Image of a software interface for input and language settings]

- **Input:**
  - plain text
  - xml
  - xml (plays)
  - xml (no titles)
  - html

- **Language:**
  - English
  - English (contr.)
  - English (ALL)
  - Latin
  - Latin (u/v > u)
  - Polish
  - Hungarian
  - French
  - Italian
  - Spanish
  - Dutch
  - German
  - CJK
  - Other
  - UTF-8

[OK button]
### FEATURES

**FEATURES:**
- words: off
- chars: on
- ngram size: 4
- preserve case: off

**MFW SETTINGS:**
- Minimum: 100
- Maximum: 1000
- Increment: 100
- Start at freq. rank: 1

**CULLING:**
- Minimum: 0
- Maximum: 20
- Increment: 5000
- List Cutoff: 5000
- Delete pronouns: off

**VARIOUS:**
- Existing frequencies: off
- Existing wordlist: off
- Select files manually: off
- List of files: off

**OK**
FEATURES

- words: words are used as the unit.
- characters: characters are used as the unit.
- $n$-gram size: this is where you can specify the value of $n$ for your $n$-grams
- preserve case: normally, all the words from the input texts are turned into lowercase.
## FEATURES

<table>
<thead>
<tr>
<th>Book Number</th>
<th>The</th>
<th>Big-Data</th>
<th>Analytics</th>
<th>Tree</th>
<th>newbie</th>
<th>book</th>
<th>for</th>
<th>Girl</th>
<th>honest</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>120</td>
<td>80</td>
<td>60</td>
<td>20</td>
<td>1</td>
<td>5</td>
<td>120</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>110</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>10</td>
<td>20</td>
<td>100</td>
<td>40</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>130</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>11</td>
<td>30</td>
<td>110</td>
<td>20</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>20</td>
<td>40</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>5</td>
<td>90</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>30</td>
<td>20</td>
<td>100</td>
<td>100</td>
<td>40</td>
</tr>
</tbody>
</table>
WHAT IS N-GRAM

N = 1 : This is a sentence
unigrams:
- this,
- is,
- a,
- sentence

N = 2 : This is a sentence
bigrams:
- this is,
- is a,
- a sentence

N = 3 : This is a sentence
trigrams:
- this is a,
MFV (most-frequent-word) settings

• Minimum: this setting determines how many words (or features) from the top of the frequency list will be used
• Maximum: this setting determines how many words from the top of the word frequency list for the entire corpus will be used
• Increment: defines the value by which the value of Minimum will be increased at each subsequent run of your analysis
• Start at freq. Rank: how many words from the top overall frequency rank list to be skipped
Culling

• The culling values specify the degree to which words that do not appear in all the texts of your corpus will be removed. Thus, a culling value of 20 indicates that words that appear in at least 20% of the texts in the corpus will be considered in the analysis. A culling setting of 0 means that no words will be removed.
Culling

<table>
<thead>
<tr>
<th>Book Number</th>
<th>Book Number</th>
<th>The</th>
<th>Big-Data</th>
<th>Analytics</th>
<th>Tree</th>
<th>newbie</th>
<th>book</th>
<th>for</th>
<th>Girl</th>
<th>honest</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>120</td>
<td>30</td>
<td>60</td>
<td>20</td>
<td>1</td>
<td>5</td>
<td>120</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>110</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>10</td>
<td>20</td>
<td>100</td>
<td>40</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>130</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>11</td>
<td>30</td>
<td>110</td>
<td>20</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>20</td>
<td>40</td>
<td>100</td>
<td>10</td>
<td>100</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>90</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>30</td>
<td>20</td>
<td>100</td>
<td>100</td>
<td>40</td>
</tr>
</tbody>
</table>
STATISTICS

Stylometry with R | stylo | set parameters

STATISTICS: Cluster Analysis
MDS
PCA (cov.)
PCA (corr.)
tSNE
Consensus Tree
Consensus strength: 0.5
DISTANCES: Classic Delta
Argamon's Delta
Eder's Delta
Eder's Simple
Manhattan
Canberra
Euclidean
Cosine
OK
CLUSTER ANALYSIS

Stylo workshop
Cluster Analysis
Cluster Analysis

Builds “tree” based on the most similar texts based on the MFV. It is not robust on the changes of the parameters.
Consensus tree

Stylo workshop
Bootstrap Consensus Tree

100-1000 MFC 4-grams Culled @ 0%
Classic Delta distance Consensus 0.5
Consensus Tree

Uses many trees to discover unchanged patterns for different parameters

It is more robust but harder to interpret.
PCA (principal component analysis)
MDS (multidimensional scaling)
Dimension reduction

Methods to reduce high dimensions into easy to interpret 2D or 3D space.

It is easy to interpret and visualization gives good intuitions on the similarity between texts.
SAMPLING
OUTPUT
Graphs

- Onscreen: the graph on the screen
- PDF: a PDF file with your graph
- JPG: graph in JPEG format
- SVG: a SVG vector file
- PNG: graph in PNG format
Lets get it to work

> library(stylo)
> stylo()
using current directory...
Performing no sampling (using entire text as sample)
loading ABronte_Agnes.txt ... 
loading ABronte_Tenant.txt ... 
loading Austen_Emma.txt ... 
loading Austen_Pride.txt ... 
loading Austen_Sense.txt ... 
loading CBronte_Jane.txt ... 
loading CBronte_Professor.txt ... 
loading CBronte_Villette.txt ... 
loading Dickens_Bleak.txt ... 
loading Dickens_David.txt ... 
loading Dickens_Hard.txt ... 
loading EBronte_Wuthering.txt ... 
loading Eliot_Adam.txt ... 
loading Eliot_Middlemarch.txt ... 
loading Eliot_Mill.txt ... 
loading Fielding_Joseph.txt ... 
loading Fielding_Tom.txt ... 
loading Richardson_Clarissa.txt ...
Stylo - Exercise

- Use 100 English Novels for stylo() analysis
- Use Eder’s and Manhattan Distance
- Cluster analysis, PCA, Consensus tree
- Use different values for culling and MFW
Intuitions how it is done

how planes fly

air
magic
air
very, important
magic

©2010 lefthandedtoons.com
Zpif Law I

If "The" Were Doubled

![Graph showing occurrences of words compared to expected frequency according to Zipf's Law. The graph highlights the word 'the' with significantly higher occurrences than expected.]
<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>i</td>
<td>15.</td>
<td>po</td>
</tr>
<tr>
<td>2.</td>
<td>się</td>
<td>16.</td>
<td>za</td>
</tr>
<tr>
<td>3.</td>
<td>w</td>
<td>17.</td>
<td>tak</td>
</tr>
<tr>
<td>4.</td>
<td>nie</td>
<td>18.</td>
<td>już</td>
</tr>
<tr>
<td>5.</td>
<td>na</td>
<td>19.</td>
<td>jej</td>
</tr>
<tr>
<td>6.</td>
<td>z</td>
<td>20.</td>
<td>od</td>
</tr>
<tr>
<td>7.</td>
<td>do</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>8.</td>
<td>to</td>
<td>3593.</td>
<td>ludzki</td>
</tr>
<tr>
<td>9.</td>
<td>a</td>
<td>3594.</td>
<td>mgnienie</td>
</tr>
<tr>
<td>10.</td>
<td>że</td>
<td>3595.</td>
<td>pacierz</td>
</tr>
<tr>
<td>11.</td>
<td>ale</td>
<td>3596.</td>
<td>pióra</td>
</tr>
<tr>
<td>12.</td>
<td>jak</td>
<td>3597.</td>
<td>podała</td>
</tr>
<tr>
<td>13.</td>
<td>o</td>
<td>3598.</td>
<td>postawie</td>
</tr>
<tr>
<td>14.</td>
<td>co</td>
<td>3599.</td>
<td>pragnęła</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3600.</td>
<td>przystani</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3601.</td>
<td>racji</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3602.</td>
<td>twardo</td>
</tr>
</tbody>
</table>
|   |   | ... | ...
|   |   | 80725. | dzieciobójstwo |
|   |   | 80726. | dzieciaskami |
|   |   | 80727. | dzieciasków |
|   |   | 80728. | dzieciński |
|   |   | 80729. | dziedzicami |
|   |   | 80730. | dziedzicowe |
|   |   | 80731. | dziedzicowym |
|   |   | 80732. | dziedzictwem |
|   |   | 80733. | dziedziczko |
|   |   | 80734. | dziedzicznej |
Zpif Law III

The more frequent word the more meanings it has
Term document frequency - novel into numbers

<table>
<thead>
<tr>
<th>Book Number</th>
<th>The</th>
<th>Big-Data</th>
<th>Analytics</th>
<th>Tree</th>
<th>newbie</th>
<th>book</th>
<th>for</th>
<th>Girl</th>
<th>honest</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>120</td>
<td>80</td>
<td>60</td>
<td>20</td>
<td>1</td>
<td>5</td>
<td>120</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>110</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>10</td>
<td>20</td>
<td>100</td>
<td>40</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>130</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>11</td>
<td>30</td>
<td>110</td>
<td>20</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>20</td>
<td>40</td>
<td>100</td>
<td>10</td>
<td>100</td>
</tr>
<tr>
<td>5</td>
<td>90</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>30</td>
<td>20</td>
<td>100</td>
<td>100</td>
<td>40</td>
</tr>
</tbody>
</table>
Z Score

- $X$ - frequency of term
- $\text{mean}(X)$ - mean frequency of term
- $S$ - standard deviation

$$Z = \frac{X - \bar{X}}{S}$$
Z Score WHY SO MUCH MATH?

- X - frequency of term
- mean(X) -
- S - standard deviation

\[ Z = \frac{X - \text{mean}(X)}{S} \]
Term document frequency - novel into numbers

<table>
<thead>
<tr>
<th>Term</th>
<th>Document Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;EBronte_Wuthering&quot; &quot;Eliot_Mill_rep&quot;</td>
<td>3.9190127861008 4.19596902207898</td>
</tr>
<tr>
<td>&quot;the&quot;</td>
<td>4.0682268396635  2.97705517340901</td>
</tr>
<tr>
<td>&quot;and&quot;</td>
<td>2.98771127931888  2.92943383520131</td>
</tr>
<tr>
<td>&quot;to&quot;</td>
<td>1.96548061503632  2.43013131944778</td>
</tr>
<tr>
<td>&quot;of&quot;</td>
<td>3.0775424273868   1.32810621001491</td>
</tr>
<tr>
<td>&quot;a&quot;</td>
<td>2.43013131944778  1.32810621001491</td>
</tr>
<tr>
<td>&quot;in&quot;</td>
<td>1.25631372683538  1.49405935831449</td>
</tr>
<tr>
<td>&quot;that&quot;</td>
<td>1.03249264649133  1.41805762662947</td>
</tr>
<tr>
<td>&quot;he&quot;</td>
<td>1.67479911843651  1.1016306219635</td>
</tr>
<tr>
<td>&quot;it&quot;</td>
<td>1.12425071391207  1.15493770744144</td>
</tr>
<tr>
<td>&quot;you&quot;</td>
<td>1.48956787953109  1.16792534513445</td>
</tr>
<tr>
<td>&quot;was&quot;</td>
<td>0.966461140029671 1.46471691760065</td>
</tr>
<tr>
<td>&quot;her&quot;</td>
<td>1.33006320158476  1.38390494973303</td>
</tr>
<tr>
<td>&quot;his&quot;</td>
<td>1.22115409352462  0.948097551589783</td>
</tr>
<tr>
<td>&quot;as&quot;</td>
<td>0.82668701923489  0.968781567174948</td>
</tr>
<tr>
<td>&quot;my&quot;</td>
<td>0.948452547358311 0.372312280532974</td>
</tr>
<tr>
<td>&quot;for&quot;</td>
<td>0.722058810918352 0.856222040502189</td>
</tr>
<tr>
<td>&quot;with&quot;</td>
<td>0.691186937767449 1.0063014093992</td>
</tr>
<tr>
<td>&quot;not&quot;</td>
<td>0.803526253955459  0.671027947472221</td>
</tr>
<tr>
<td>&quot;be&quot;</td>
<td>0.618295015050038  0.683053537928712</td>
</tr>
<tr>
<td>&quot;had&quot;</td>
<td>0.590853350027013  1.0014911732166</td>
</tr>
<tr>
<td>&quot;she&quot;</td>
<td>1.10023925701692  0.946654480735004</td>
</tr>
<tr>
<td>&quot;have&quot;</td>
<td>0.542839436236719  0.538265428832556</td>
</tr>
<tr>
<td>&quot;me&quot;</td>
<td>0.91243536201559  0.417047477031122</td>
</tr>
<tr>
<td>&quot;but&quot;</td>
<td>0.589138245963074  0.77108086070229</td>
</tr>
<tr>
<td>&quot;is&quot;</td>
<td>0.560839028908079  0.427148973014575</td>
</tr>
<tr>
<td>&quot;at&quot;</td>
<td>0.672320793064119  0.682091490692193</td>
</tr>
<tr>
<td>&quot;him&quot;</td>
<td>0.794093181603794  0.513733224301313</td>
</tr>
<tr>
<td>&quot;so&quot;</td>
<td>0.307861179477065  0.347780076001732</td>
</tr>
<tr>
<td>&quot;this&quot;</td>
<td>0.253835401462984  0.381932752898167</td>
</tr>
</tbody>
</table>
Distances

- Time
- Manhattan
- Euclidean
Distance

- dictionary (the, dog, cat, to be, sun)
- text 1  (12, 4, 4, 5, 1)
- text 2  (21, 1, 7, 5, 2)
- text 3  (17, 3, 6, 3, 2)

$d(text1, text2) = |12-21| + |4-1| + |4-7| + |5-5| + |1-2|$

= 9 + 3 + 3 + 0 + 1 = 16
Distance - Exercise

- dictionary (the, dog, cat, to be, sun)

- text 1 (12, 4, 4, 5, 1)

- text 2 (21, 1, 7, 5, 2)

- text 3 (17, 3, 6, 3, 2)

Exercise:
Calculate d(text2, text3).
Distances

- Non-standard words weights adjusted
- More importance to MFV
- Modification of Manhattan distance
- All words equal weights

- Classic Delta:
  \[ \delta_{(AB)} = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{f_i(A) - f_i(B)}{\sigma_i} \right| \]

- Argamon’s Delta:
  \[ \tilde{\delta}_{(AB)} = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{\sqrt{f_i(A)^2} - f_i(B)^2}{\sigma_i} \right| \]

- Eder’s Delta:
  \[ \delta_{(AB)} = \frac{1}{n} \sum_{i=1}^{n} \left( \left| \frac{f_i(A) - f_i(B)}{\sigma_i} \right| \times \frac{n - n_i + 1}{n} \right) \]

- Eder’s Simple:
  \[ \delta_{(AB)} = \sum_{i=1}^{n} \left| \sqrt{f_i(A)} - \sqrt{f_i(B)} \right| \]

- Manhattan:
  \[ \tilde{\delta}_{(AB)} = \sum_{i=1}^{n} \left| f_i(A) - f_i(B) \right| \]
library(stylo)
classify()
What is classification?
CORPUS PREPARATION FOR
CLASSIFY
INPUT & LANGUAGE
FEATURES

Stylometry with R: enter analysis parameters

- FEATURES:
  - words
  - chars
  - ngram size: 1
  - preserve case

- MFW SETTINGS:
  - Minimum: 100
  - Maximum: 100
  - Increment: 100
  - Start at freq. rank: 1

- CULLING:
  - Minimum: 0
  - Maximum: 0
  - Increment: 20
  - List Cutoff: 5000
  - Delete pronouns

- VARIOUS:
  - Existing frequencies
  - Existing wordlist
  - Select files manually
  - List of files

OK
STATISTICS

A screenshot of a software interface showing analysis parameters for Stylometry with R. The interface includes options for statistics such as Delta, k-NN, SVM, NaiveBayes, NSC, and general settings for culling, wordlists, and z-scores. There are also options for delta distance methods like Classic Delta, Argamon's Delta, Eder's Delta, and Eder's Simple, along with distance measures such as Manhattan, Canberra, Euclidean, and Cosine. SVM options include linear, polynomial, and radial with degree and cost settings. k-NN options include k value and l value.
SAMPLING
OUTPUT
RESULTS FOR CLASSIFY

Dickens_Bleak  -->  Eliot
Dickens_David  -->  Eliot
Dickens_Hard   -->  Eliot
Fielding_Joseph -->  Eliot
Fielding_Tom   -->  Austen
Richardson_Clarissa -->  ABronte
Richardson_Pamela  -->  ABronte
Sterne_Sentimental -->  CBronte
Sterne_Tristrum   -->  CBronte
Thackeray_Barry   -->  CBronte
Thackeray_Pendennis -->  CBronte
Thackeray_Vanity  -->  CBronte
Trollope_Barchester -->  Eliot
Trollope_Phineas  -->  Eliot
Trollope_Prime    -->  Eliot

100 MFW, culled @ 0%, 0 of 0 (NaN%)

General attributive success: 0 of 0 (NaN%)

MFWs from 100 to 100 @ increment 100
library(style)
  # Load corpus
  raw.corpus <- load.corpus(files = 'all', corpus.dir = 'corpus', encoding = 'UTF-8')
  # Summary of corpus files
  summary(raw.corpus)
  # Tokenization
  tokenized.corpus <- txt.to.words.ext(raw.corpus, language = 'English.all', preserve.case = FALSE)
  # Austin's Pride and Prejudice famous sentence
  tokenized.corpus[Austen_Pride.txt][8:38]
  # English pronouns
  stylo.pronouns(language = 'English')
  # Deleting useless words
  corpus.no.pronouns <- delete.stop.words(tokenized.corpus, stop.words = stylo.pronouns(language = 'English'))
  # Counting frequent features
  frequent.features <- make.frequency.list(corpus.no.pronouns, head = 3000)
  freqs <- make.table.of.frequencies(corpus.no.pronouns, features = frequent.features, relative = FALSE)
  freqs <- make.table.of.frequencies(corpus.no.pronouns, features = frequent.features)
  # Culling (words in at least 85% of samples)
  culled.freqs <- perform.culling(freqs, culling.level = 85)
  # Simple cluster analysis with stylo
  stylo(frequencies = culled.freqs, gul = FALSE)
  # More options in stylo
  stylo(frequencies = culled.freqs, gul = FALSE, analysis.type = 'PCA', custom.graph.title = "PCA analysis for novels")
More intuitions on classification
Naive Bayes

What is the probability that this is ABronte given there are such words used?

\[
P(c \mid x) = \frac{P(x \mid c)P(c)}{P(x)}
\]

\[
P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)
\]
K-nn

What are the closest observations to ABronte?
Classify - Exercise

- Remember about correct data set up
- Use 100 English Novels for classify() analysis
- KNN (for different k values), Naive Bayes,
- Use different values for culling and MFW
Corpus prep for Rolling.classify
How to use it

rolling.classify(slice.size = 1500, mfw = 100, slice.overlap = 500, plot.legend = T, classification.method = "svm")
Rolling.classify

> rolling.classify(slice.size = 500, mfw = 100, slice.overlap = 100, plot.legend = T)
using current directory...
The subcorpora will be loaded from text files...
loading Bronte_Wuthering.txt ...
slicing input text into tokens...
turning words into features, e.g. char n-grams (if applicable)...
loading Bronte_Wuthering.txt ...
slicing input text into tokens...
Bronte_Wuthering
  - text length (in words): 116612
  - nr. of samples: 291
  - nr. of words dropped at the end of the text: 212
turning words into features, e.g. char n-grams (if applicable)...
loading Bronte_Agnes.txt ...
loading Bronte_Jane.txt ...
loading Bronte_Professor.txt ...
loading Bronte_Tenant.txt ...
loading Bronte_Villette.txt ...
loading CBronte_Jane.txt ...
loading CBronte_Professor.txt ...
loading CBronte_Villette.txt ...
loading EBronte_Wuthering.txt ...
slicing input text into tokens...
turning words into features, e.g. char n-grams (if applicable)...

Results of rolling classify
Rolling.classify - Exercise

- Remember about correct data set up
- Use 100 English Novels for rolling.classify()
- Choose 2 authors for reference set and only one novel
- Change values slice.size, mfw, and slice.overlap
Way Forward

- Workflow using code
- Word clouds
- Graph data visualizations
- ...

...
Visualizations
Download R files

https://github.com/jandziak/Stylo_workshop
Workflow using code

- # Load corpus
- # Tokenization
- # Deleting useless words
- # Creating features
- # Culling
- # Simple cluster analysis with stylo
- # More options in stylo
# Load corpus

# Load corpus
raw.corpus <- load.corpus(files = 'all', corpus.dir = 'corpus', encoding = 'UTF-8')

# Summary of corpus files
summary(raw.corpus)
# Tokenization

tokenized.corpus <-
txt.to.words.ext(raw.corpus, language = 'English.all', preserve.case = FALSE)

# Austin's Pride and Prejudice famous sentence

tokenized.corpus$Austen_Pride.txt[8:30]
# Deleting useless words

# English pronouns
stylo.pronouns(language = 'English')

# Deleting useless words
 corpus.no.pronouns <-
 delete.stop.words(tokenized.corpus,
 stop.words = stylo.pronouns(language = 'English'))
# Creating features

# Counting frequent features
frequent.features <- make.frequency.list(corpus.no.pronouns, head = 3000)
freqs <- make.table.of.frequencies(corpus.no.pronouns, features = frequent.features, relative = FALSE)
freqs <- make.table.of.frequencies(corpus.no.pronouns, features = frequent.features)
# Culling

# Culling (words in at least 80% of samples)
culled.freqs <- perform.culling(freqs, culling.level = 80)
# Simple cluster analysis with stylo

```r
# Simple cluster analysis with stylo
stylo(frequencies = culled.freqs, gui = FALSE)
```
# More options in stylo

```r
stylo(frequencies = culled.freqs, gui = FALSE, analysis.type = 'PCR', custom.graph.title = "PCA analysis for novels")
```
Word clouds
Workflow using code

- # Prepare environment
- # Plot Word cloud
# Prepare Environment

# Install missing packages
install.packages("wordcloud") # word-cloud generator
install.packages("RColorBrewer") # color palettes
library("wordcloud")
library("RColorBrewer")
# Plot Word cloud

# Ensure reproducibility
set.seed(1234)

wordcloud(words = names(df), freq = as.numeric(df[1,1:2543]), min.freq = 0.01, max.words=200, random.order=FALSE, rot.per=0.35, colors=brewer.pal(8, "Dark2"))
Graph data visualizations
Just go to the code
Way forward - Exercise

- Apply stylo code analysis on new data
- Use 100 English Novels
- Produce new word cloud using different novel
- Create graph for novels