Università degli studi di Padova
Dipartimento di Scienze Statistiche
Corso di Laurea Triennale in

Statistica per le Tecnologie e le Scienze

Relazione Finale

Studio di tre opere di Lisia

Text mining in greco antico

Relatore: Prof. Bruno Scarpa
Dipartimento di Scienze Statistiche

Laureanda: Sara Bonacina
Matricola N. 1198027

Anno Accademico 2020/2021

Indice

Introduzione

Lisia è stato, insieme a Demostene e a Isocrate, uno dei maggiori esponenti dell'arte oratoria in Grecia a cavallo tra il V e il IV secolo a.C. Già per Platone nel *Fedro* egli è il rappresentante riconosciuto dell'eloquenza sofistica, mentre per Cicerone nel *Brutus* e nell'*Orator* è soprattutto un oratore giudiziario. Fin dall'antichità Lisia ebbe, dunque, grande fortuna e il suo stile attico e classico venne preso a modello. La tradizione antica attribuì a Lisia 425 orazioni, delle quali secondo Dionigi di Alicarnasso solo 233 erano autentiche. A noi ne sono giunte 34, tutte di genere giudiziario eccetto due, l'*Olimpico* e l'*Epitafio*, di genere, invece, epidittico. Di queste una sola fu pronunciata in causa propria, la *Contro Eratostene*, mentre le restanti furono scritte per altri e non furono esposte dall'autore in persona.

L'obiettivo di questo elaborato è quello di fornire un quadro descrittivo di ciò che ci resta della produzione di Lisia attraverso l'elaborazione statistica di tre sue opere il più possibile rappresentative della varietà degli argomenti, dei generi e dello stile propri dell'autore. Per motivi che chiariemo in seguito, abbiamo deciso di considerare le già citate *Contro Eratostene* ed *Epitafio* e inoltre la celebre orazione *Per l'uccisione di Eratostene*. Tuttavia, non ci siamo limitati solo a studiare i contenuti e trovare somiglianze e differenze tra queste opere, ma abbiamo affrontato anche la spinosa questione dell'autenticità di una di esse, l'*Epitafio* appunto, argomento al centro di dibattiti tra gli studiosi ormai dagli anni Ottanta dell'Ottocento. Per fare ciò siamo ricorsi a un insieme di tecniche statistiche che prende il nome di *text mining*, ossia un processo semi-automatizzato che consente di estrarre e classificare informazioni da testi. Ovviamente le orazioni prese in considerazione sono state elaborate in lingua originale, cioè in greco antico. L'analisi testuale è

stata implementata tramite il *software* R (R Core Team 2020).

Prima di addentrarci in questa relazione, diamo una breve descrizione dei capitoli che la compongono. Nel Capitolo 1 forniremo gli strumenti necessari alla conoscenza e alla comprensione delle opere considerate, presentando in primo luogo l'autore, il suo contesto storico-sociale e gli aspetti principali della sua produzione, e in seconda battuta i contenuti e le peculiarità delle tre orazioni analizzate. Infine, riassumeremo le principali posizioni e le relative motivazioni dei classicisti circa la paternità dell'*Epitafio*. Successivamente, nel Capitolo 2 ci occuperemo della vera e propria analisi testuale, al fine di indagare e valutare anche con un approccio statistico quanto presentato nel Capitolo 1. In particolare, dopo una fase iniziale di elaborazione dei testi, forniremo un quadro descrittivo dei contenuti delle opere, per poi procedere con una serie di analisi supervisionate (regressione lasso, ridge ed *elastic net*) e non supervisionate (Latent Dirichlet Allocation e *clustering* gerarchico), con l'intento di classificare e raggruppare i testi provenienti dalle tre orazioni e studiare le possibili relazioni che intercorrono fra di esse.

Capitolo 1

Presentazione delle opere

1.1 Cenni su Lisia

Lisia (Todd et al. 2007), nato ad Atene probabilmente poco dopo il 440 a.C., era figlio di un meteco, Cefalo, ricco siracusano trasferitosi ad Atene ai tempi di Pericle per stabilirvi una fiorente fabbrica di scudi. La sua formazione retorica avvenne tra il 430 e il 418 a.C. circa nella colonia magnogreca di Turii, in Italia meridionale. Tornato in patria, Lisia assisté all'ultima fase della Guerra del Peloponneso, che si concluse con la sconfitta di Atene e con la caduta della sua democrazia (404 a.C.), in seguito alla quale si instaurò il regime oligarchico dei Trenta Tiranni. Essi, che già non vedevano di buon occhio le simpatie democratiche di Lisia e della sua famiglia, erano anche desiderosi di impossessarsi del suo ingente patrimonio, pertanto non esitarono a catturare e a uccidere il fratello dell'autore, Polemarco. Lisia si salvò fuggendo a Megara, ma perdette tutti i suoi beni. Da qui in poi egli ci appare legato ai democratici ateniesi, che aiutò nella reastaurazione della democrazia. Il nuovo capo di questa, Trasibulo, ottenne per lui la piena cittadinanza, ma il decreto fu reso vano da un'eccezione giuridica. Ridotto in povertà e privo della cittadinanza, che sola dà accesso alla politica, Lisia fu costretto a praticare la poco onorata ma redditizia professione di logografo, in cui eccelse, lavorando come maestro di retorica e scrivendo orazioni giudiziarie per altri. Non si hanno ulteriori informazioni precise sulla sua vita, ma sulla base

dei testi pervenutici Lisia non visse oltre il 361 a.C.

Come anticipato nell'Introduzione, l'opera di Lisia riscosse fin da subito uno straordinario successo, tanto che nel IV sec. a.C. circolavano sul mercato librario ateniese orazioni giudiziarie attribuite all'autore ad uso di quanti, non potendo permettersi di pagare un logografo, cercavano un testo da poter adattare alle proprie esigenze. Molte di queste erano addirittura false, mentre altre risultavano scritte "a quattro mani" con il cliente. Di qui la difficoltà di definire quali delle 34 orazioni lisiane a noi giunte siano veramente opere autentiche. La maggior parte di esse appartiene al genere giudiziario, cioè sono scritte su incarico di un committente che, secondo la prassi giudiziaria del tempo, doveva poi recitarle di persona in tribunale. Si tratta per lo più di cause private. Le orazioni affrontano temi svariati, come appare evidente anche dai titoli (*Per l'invalido, Contro i mercanti di grano, Per il soldato, Per l'olivo sacro*, etc.), in relazione alla varietà delle cause: peculato, tradimento, corruzione, inadempienza agli obblighi militari, sacrilegio, diffamazione. All'interno del *corpus* lisiano si trovano, tuttavia, anche due orazioni di interesse politico (legate, quindi, a cause pubbliche): la *Contro Eratostene*, di cui parleremo nella prossima sezione, e la *Contro Agorato*, in cui l'autore attacca un emissario degli oligarchi, che aveva provocato la morte di alcuni esponenti del partito democratico.

Le opere di Lisia si distinguono per il rigore della documentazione e la chiarezza espositiva, in quanto esse seguono sempre la seguente struttura: prefazione, esposizione del fatto, presentazione delle testimonianze ed epilogo. Lisia si preoccupava di valorizzare le ragioni del suo committente e di far coincidere lo stile dell'argomentazione con la personalità del cliente e il suo livello culturale e sociale secondo il principio greco dell'ηθοποιία (etopea, "rappresentazione del carattere"). La lingua usata da Lisia è un dialetto attico puro, in cui sono accuratamente evitati vocaboli o espressioni poetiche e figure retoriche, che sono concentrati, semmai, nel proemio e nella conclusione. La narrazione dell'evento procede, invece, per periodi brevi e slegati fra loro, in quanto l'obiettivo dell'autore è quello di apparire semplice e chiaro. Per tale ragione Lisia ha costituito un punto di riferimento essenziale per tutta la prosa successiva, in modo particolare per la corrente ellenistica dell'atticismo.

1.2 Le opere considerate

Presentiamo ora brevemente le orazioni che analizzeremo in questo elaborato e che abbiamo selezionato sulla base della loro rappresentatività nella produzione dell'autore per quanto riguarda gli argomenti trattati e lo stile adottato, in modo da dare un'idea dei contenuti e del genere delle tre opere considerate.

Uno dei più celebri e apprezzati discorsi di Lisia è sicuramente *Contro Eratostene*, pronunciato nel 403 a.C. ad Atene in un clima di distensione fra gli oligarchici e i democratici di Trasibulo. Lisia cerca di riappropriarsi del proprio patrimonio e di vendicarsi delle ingiustizie subite da parte di Eratostene, uno dei Trenta Tiranni, responsabile, tra l'altro, dell'arresto dell'autore e della condanna a morte del fratello Polemarco. Eratostene, tuttavia, venne probabilmente assolto, anche perché i democratici volevano mantenere la pace, evitando di creare nuovi attriti con la parte avversaria. L'opera si apre con un breve esordio, a cui fa seguito una narrazione chiara dei fatti precedentemente descritti. A questo punto l'autore incalza Eratostene con un interrogatorio. Infine, il discorso si conclude con una perorazione in cui Lisia mette a confronto il governo dei Trenta Tiranni e la democrazia, auspicando una riappacificazione lontana da intrighi e rancori, affinché si faccia giustizia e si condannino i colpevoli dei passati misfatti. Abbiamo deciso di prendere in considerazione quest'opera anche perché si tratta di una delle poche orazioni per cause pubbliche, che fu addirittura pronunciata da Lisia in persona (fatto unico nella storia dell'autore).

Un altro discorso che abbiamo analizzato è *Per l'uccisione di Eratostene*, anch'esso molto celebre e appartenente al genere giudiziario. In questo caso abbiamo a che fare con una causa privata, in cui Lisia deve cercare di difendere un cittadino ateniese, Eufileto, dall'accusa di omicidio premeditato da parte dei parenti dell'ucciso, Eratostene (si tratta di una persona diversa dal tiranno Eratostene, di cui abbiamo parlato in precedenza). L'oratore si propone di dimostrare che l'omicidio è legittimo e rientra nel cosiddetto φόνος δίκαιος ("delitto d'onore"), previsto dalla legge di Dracone, sostenendo che Eufileto abbia compiuto tale gesto a causa della relazione adulterina tra sua moglie ed Eratostene. Il contenuto dell'opera consiste principalmente nella

narrazione dettagliata degli avvenimenti successivi al matrimonio tra Eufileto e la moglie, dalla nascita del figlio, ai comportamenti insoliti della donna, fino alla delazione della serva, che il marito tradito usa come complice per organizzare un tranello ai danni degli amanti. L'orazione è molto enfatica, in quanto Lisia contrappone la figura dell'onesto cittadino Eufileto a quella dell'infida moglie, ponendo sotto una luce negativa Eratostene, reo di aver commesso un atto illegale, l'adulterio.

Infine, come orazione rappresentativa della produzione epidittica dell'autore abbiamo considerato l'*Epitafio*, sebbene si tratti di un testo assai problematico, come discuteremo nel prossimo paragrafo. L'opera è un discorso celebrativo degli uomini caduti in una battaglia (probabilmente nella battaglia di Cnido del 394 a.C.) durante la guerra di Corinto (395-387 a.C.), un conflitto che vide contrapporsi Sparta con parte della Lega Peloponnesiaca e Atene, sostenuta non solo da altre città greche ma anche dalla Persia, almeno in un primo momento. L'episodio in questione riguarda in particolare la spedizione di un contingente ateniese in soccorso ai corinzi, tuttavia non si hanno riferimenti precisi a tale fatto, in quanto Lisia riserva uno spazio molto ridotto all'occasione concreta del discorso. L'orazione si diffonde invece su episodi mitici della storia ateniese e delle sue imprese passate. Seguono poi un elogio dei caduti, del loro sacrificio e della loro devozione alla democrazia e un epilogo, che costituisce una consolazione e un incoraggiamento per le famiglie dei morti in battaglia.

1.3 La questione dell'autenticità dell'*Epitafio*

Il problema, più che secolare, riguardante la paternità dell'*Epitafio* è assai rilevante per la valutazione globale della figura di Lisia, in quanto questo discorso, se autentico, fa dell'oratore il protagonista di un'occasione pubblica, l'elogio dei caduti in guerra, nella quale la democrazia ateniese costruisce un'immagine di sé fondata sui valori di libertà, concordia e aiuto dei deboli contro ogni forma di oppressione. Gli studiosi sono stati, dunque, portati a valutare la compatibilità di un simile intervento pubblico sia con lo status sociale dell'autore sia con quanto si può ricavare dal resto della sua produzione circa le sue idee politiche. Una trattazione più dettagliata della questione,

a partire dalla seconda metà dell'Ottocento a oggi, si può trovare in Medda 2016. Di seguito riportiamo una sintesi dei punti più significativi.

I principali argomenti addotti contro la paternità lisiana dell'*Epitafio* riguardano controverse questioni di datazione e composizione dell'opera, dovute soprattutto agli scarsi riferimenti ai fatti storici narrati, come già abbiamo spiegato in precedenza, e all'ambiguità di certi passaggi. Tuttavia, l'aspetto che più considereremo anche nelle nostre analisi, è quello riguardante lo stile dell'*Epitafio*, che appare completamente diverso da quello del Lisia delle orazioni giudiziarie. Si tratta, infatti, di uno stile enfatico, ricco di figure di ripetizione, con una sintassi carica di subordinate e talvolta ai limiti della comprensibilità. Inoltre, alcuni studiosi hanno evidenziato una serie di somiglianze tra l'*Epitafio* e il *Panegirico* di Isocrate: si è dunque pensato che il primo fosse una sommaria imitazione del secondo ad opera di un autore meno dotato di Lisia.

D'altra parte, queste argomentazioni sono state soggette a diverse critiche da parte dei sostenitori dell'autenticità dell'*Epitafio*. In particolare, per quanto riguarda le questioni stilistiche bisogna considerare che le orazioni epidittiche di Lisia sono andate quasi completamente perdute, pertanto non è possibile valutare correttamente quanto il suo stile in questo genere di discorsi si discostasse da quello ben noto dei discorsi giudiziari. Si deve, inoltre, tener conto del fatto che negli epitafi era inevitabile un'alternanza tra luoghi comuni ed espressioni elevate e poetiche. Alcuni studiosi hanno, pertanto, sostenuto che fosse possibile individuare nell'*Epitafio* tratti dello stile di Lisia riscontrabili anche nelle orazioni giudiziarie e che gli altri elementi più audaci fossero attribuibili alla peculiarità del genere epidittico.

Capitolo 2

Analisi testuale

L'analisi testuale (Meyer, Hornik e Feinerer 2008), anche detta *text analysis* o *text mining*, consiste nel processo di derivazione di informazioni rilevanti da testi, che rappresentano un tipo di dato non strutturato, dopo averli appositamente riorganizzati in dati strutturati, al fine di esplorarne i contenuti, identificarne gli elementi o gli aspetti rilevanti e interessanti, classificarli o raggrupparli e rispondere a specifiche domande.

L'analisi testuale prevede una fase iniziale di analisi preliminare, in cui i dati in forma di testo vengono letti, caricati e sottoposti a un'operazione di pulizia per poter poi essere utilizzati nella creazione della matrice termini-documenti (*document-term matrix*). Si tratta di una struttura dati le cui righe corrispondono ai documenti del *corpus* e le colonne ai termini rilevanti contenuti in tali testi. Essa pertanto descrive la frequenza dei termini nei documenti presi in analisi. Durante questa fase di elaborazione dei documenti i testi vengono normalizzati, le parole che non forniscono informazioni utili alle analisi vengono rimosse e si cerca di raggruppare in un unico termine parole che esprimono il medesimo concetto. Nella prossima sezione illustreremo in modo dettagliato queste operazioni, in quanto si tratta di un processo non standardizzato che dipende da vari aspetti, quali il tipo di testo, la sua struttura, la lingua in cui è scritto, le tematiche trattate, etc.

Una volta ottenuta la matrice termini-documenti, segue una fase di analisi statistiche, che nel nostro caso si basano sulla classificazione e sul raggruppamento di testi attraverso metodi supervisionati e non supervisionati. Il nostro

obiettivo infatti è quello di studiare i contenuti e lo stile delle tre orazioni presentate nel Capitolo 1, al fine di verificare se una delle opere considerate, l'*Epitafio*, sebbene sia storicamente attribuita a Lisia, presenti tuttavia differenze rispetto alle altre opere autentiche dell'autore tali da poterci portare a ritenerla spuria.

2.1 Elaborazione dei documenti

Nel seguito illustriamo nel dettaglio i passaggi della fase di analisi preliminare delle opere di Lisia prese in considerazione nelle nostre analisi: *Contro Eratostene*, *Per l'uccisione di Eratostene* e l'*Epitafio*. A tale scopo è stato utilizzato l'ambiente statistico `R`. I testi delle opere sono state reperite mediante la libreria `rperseus` (Ranzolin 2021), nella quale è disponibile la funzione `get_perseus_text()`, che permette di ottenere un testo dalla *Perseus Digital Library* (`http://www.perseus.tufts.edu/hopper/`) in formato *tibble* a partire da un catalogo disponibile nel *dataframe* `perseus_catalog`. Una volta caricate le tre opere, queste sono state suddivise sulla base dei paragrafi (*excerpts*) che le compongono. In questo contesto, per paragrafi si intendono gruppi di due o tre frasi. Abbiamo quindi ottenuto 231 documenti, che costituiranno le righe del nostro *dataframe*. Infine, per identificare l'opera a cui appartiene ciascun paragrafo, abbiamo definito una variabile `opera`.

Per le operazioni di normalizzazione dei testi, rimozione delle *stopwords* e *stemming* è stato necessario ricorrere a diverse librerie, quali `quanteda` (Benoit et al. 2018), `tm` (Feinerer e Hornik 2020) e `tidytext` (Silge e Robinson 2016), in quanto l'analisi testuale in greco antico presenta diverse criticità dovute innanzitutto all'alfabeto e alla sua codifica e in secondo luogo alla mancanza o all'incompletezza di funzioni che implementino le operazioni sopra citate.

2.1.1 Pulizia del testo

Inizialmente i testi sono stati sottoposti al processo di normalizzazione, che consiste nella rimozione di punteggiatura, numeri e spazi vuoti. Per fare

ciò è stata utilizzata la funzione `tokens()` della libreria `quanteda`, specificando gli argomenti `remove_punct`, `remove_numbers` e `remove_separators`.

Successivamente abbiamo proceduto con l'eliminazione delle cosiddette *stopwords*, cioè parole poco o per niente specifiche che non forniscono informazione, come ad esempio articoli, congiunzioni, preposizioni, avverbi, interiezioni. Nel caso del greco antico risulta fondamentale rimuovere anche le particelle, ossia parole utilizzate con una certa frequenza come intercalare e spesso non traducibili in un unico modo o in un'unica parola in italiano, come per esempio μέν ... δέ ("mentre" ... "invece", "da una parte" ... "dall'altra", etc.). Infine è necessario eliminare anche le abbreviazioni critiche, tipiche nei testi antichi. La libreria `quanteda` fornisce una *stoplist* molto esaustiva in questo senso per il greco antico (contiene 6489 *stopwords*) attraverso la funzione `stopwords()`, specificando `language = "grc"` e `source = "ancient"`. Una volta costruita la *stoplist*, è possibile rimuovere le *stopwords* grazie alla funzione `tokens_remove()` della medesima libreria.

2.1.2 Stemming

A questo punto abbiamo valutato se applicare lo *stemming* ai termini presenti nel vocabolario del nostro *corpus* al fine di ridurne il numero, raggruppando parole con lo stesso significato in un solo termine. Lo *stemming* (Lovins 1968) infatti è il processo di riduzione delle parole alla loro radice fondamentale, detta tema, la quale però non corrisponde necessariamente alla radice morfologica della parola, chiamata invece lemma. Tuttavia è sufficiente che parole tra loro legate appartengano allo stesso tema, anche se quest'ultimo non è una valida radice per la parola.

Vista la complessità della lingua greca e le sue innumerevoli eccezioni, alcuni (ad es. Berra 2020) sostengono che fare *stemming* in lingue antiche, come il greco, sia per lo più inutile, in quanto, per esempio, al variare del caso (per sostantivi e aggettivi) oppure al variare della persona, del modo o del tempo (per i verbi) le vocali cambiano, diventando lunghe o brevi (da ε a η oppure da ο a ω, e viceversa), e inoltre anche gli accenti si spostano da una sillaba all'altra e cambiano in acuto, grave o circonflesso. Tuttavia abbiamo ritenuto che questa non fosse una motivazione sufficiente per giustificare la

mancata applicazione dello *stemming* a testi in greco antico e che ne indichi piuttosto la difficoltà.

I problemi evidenziati in precedenza, infatti, più che altro mettono in evidenza i limiti del modo tradizionale di fare *stemming*, cioè attraverso metodi che sostanzialmente troncano le parole. In questo caso potrebbe essere opportuno procedere con un'altra tecnica, più complessa dello *stemming*, ossia la lemmatizzatione, che prevede l'utilizzo del lemma (radice morfologica) della parola anziché il suo stilema (radice fondamentale). Il lemma è il termine che per convenzione rappresenta tutte le flessioni e coincide con la forma presente nel vocabolario. Un algoritmo per la lemmatizzazione in greco antico è disponibile in Python nella libreria `ctlk` (*Classical Language Toolkit*) [`https://legacy.cltk.org/en/latest/`]. Tuttavia, vista la natura complicata di questa operazione e l'eccessivo costo computazionale derivante dalla quantità di parole presenti nel *corpus*, abbiamo preferito un'altra soluzione.

Abbiamo dunque ricercato uno *stemmer* per il greco antico. Al momento però non sembra disponibile alcuno strumento di qualità, se non un algoritmo di *stemming*, simile a quello di Porter (Porter 1980), uno degli algoritmi di *stemming* più utilizzati, che tuttavia funziona solo per il greco moderno e per testi scritti in caratteri maiuscoli senza spiriti e accenti (Ntais 2006). La libreria `tm` invece fornisce la funzione `stemDocument()`, che utilizza lo *stemmer* di Snowball (Porter 2001). Si tratta di un linguaggio di elaborazione di stringhe di piccole dimensioni progettato per la creazione di algoritmi di derivazione da utilizzare nel recupero delle informazioni. Specificando l'opzione `language = "greek"` è possibile fare *stemming* automatico di documenti in lingua greca. L'algoritmo funziona sia per il greco moderno sia per il greco antico, in quanto implementa uno "*stemming* leggero" che permette di soprassedere sulle differenze tra le due lingue, che sono comunque ridotte. Abbiamo appurato che i risultati ottenuti sono soddisfacenti e portano a una riduzione dei termini del vocabolario del nostro *corpus*, che passano da 2854 a 2522.

2.2 Quadro descrittivo

Prima di procedere con le analisi statistiche può essere utile fornire un

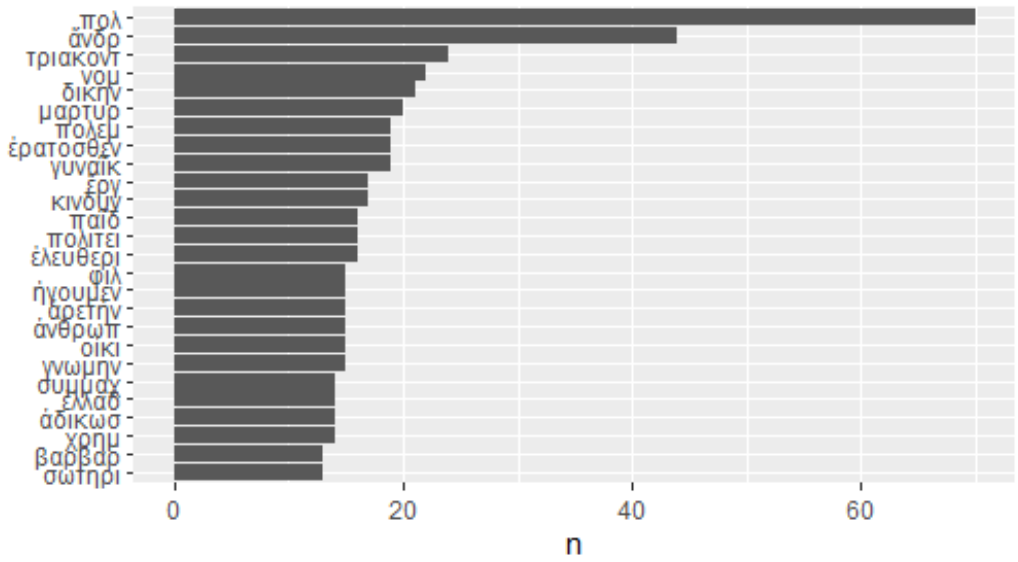

Figura 2.1: Distribuzione di frequenza delle parole più utilizzate

quadro descrittivo dei contenuti delle opere presentate nel Capitolo 1 attraverso il calcolo delle frequenze dei termini. Ricordando che abbiamo eliminato le *stopwords*, andiamo ad osservare quali sono le parole più frequenti all'interno del nostro *corpus*. Per visualizzare ciò utilizziamo un grafico a barre (Figura 2.1) con le 25 parole più utilizzate. Fra esse si possono notare due parole molto più frequenti delle altre (πόλις e ανήρ), ma si tratta di parole piuttosto comuni appartenenti al lessico di base (significano, rispettivamente, "città" e "uomo"). Gli altri termini sembrano, invece, utili per identificare il contenuto di ciascuna opera. Procediamo pertanto al calcolo di altri indici che possono servirci in questo contesto.

Uno di questi è il *term frequency* (TF), che descrive la frequenza dei termini, cioè quante volte una parola compare in un documento (nel nostro caso, in uno dei paragrafi che costituiscono le opere considerate). Indicando con n_{ij} il numero di volte che il termine t_i compare nel paragrafo d_j e con n_j il numero di termini del paragrafo d_j, l'indice TF si calcola come:

$$TF(i,j) = \frac{n_{ij}}{n_j}$$

Un altro indice utile è l'*inverse document frequency* (IDF), che dà un peso minore alle parole di uso comune e un peso maggiore alle parole che non

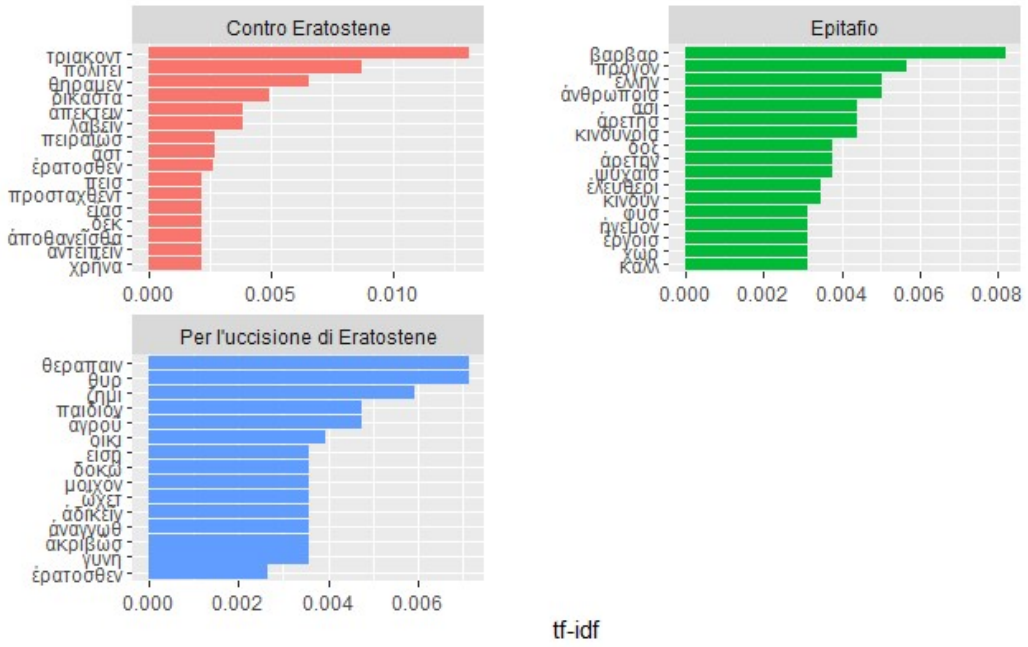

Figura 2.2: Distribuzione di frequenza dei termini per ciascuna opera

sono molto utilizzate in un insieme di documenti. Sia D il numero totale di documenti (paragrafi) presenti nel *corpus* e d_i il numero di paragrafi che contengono il termine i, è possibile ottenere IDF tramite la formula:

$$IDF(i) = \log\left(\frac{D}{d_i}\right)$$

Infine, moltiplicando i due indici sopra descritti, si ottiene il TF-IDF, un indice numerico che intende riflettere l'importanza di una parola per un documento (paragrafo) in un *corpus*:

$$TF - IDF(i,j) = TF(i,j) \times IDF(i)$$

La libreria `tidytext` permette di calcolare in maniera semplice e immediata TF, IDF e TF-IDF di ciascun termine usando il comando `bind_tf_idf()`.

In Figura 2.2 sono riportate le parole con TF-IDF più elevato per ciascuna delle tre opere. Si tratta soprattutto di parole che descrivono l'argomento e il contenuto delle opere considerate. Infatti, i termini più importanti nell'orazione *Contro Eratostene*, per esempio, riguardano il regime dei Trenta

Tiranni (τριάκοντα), i cittadini (πολίτες), la giustizia (δίκη), l'omicidio e la morte di Polemarco (αποκτείνω, αποθνήσκω) e rispecchiano effettivamente il carattere politico dell'opera. Nell'*Epitafio* emergono, invece, parole relative alla guerra tra i barbari (βάρβαρος) e la Grecia (Ἑλλας) per la libertà (ελευθερία) di quest'ultima, i pericoli (κινδύνος) della battaglia, l'eroicità (αρετή) dei combattenti, sottolineando quindi l'argomento bellico e lo scopo celebrativo della composizione. Infine, nell'opera *Per l'uccisione di Eratostene* i termini più rappresentativi sono θεράπαινα ("serva"), θύρα ("porta"), πάις ("bambino"), οικία ("casa"), γυνή ("donna"), che sono collegati alla vicenda giudiziaria raccontata nell'orazione.

Al termine di queste analisi descrittive, prima di procedere con le analisi statisctiche, abbiamo deciso di rimuovere il termine τριακοντ, in quanto presentava un valore di TF-IDF nettamente superiore rispetto agli altri termini (pari a circa 0.012) ed era specifico di una delle opere considerate (è infatti legato al tema della *Contro Eratostene*), e ερατοσθεν, poiché è il nome, in un caso, del tiranno accusato da Lisia e, nell'altro, dell'amante ucciso, e rappresenta quindi due entità diverse, entrambe caratteristiche delle due orazioni.

2.3 Analisi statistica

Come accennato ad inizio capitolo, la matrice termini-documenti è una forma di dati strutturati in grado di rappresentare il contenuto dei testi del nostro *corpus*. Una volta ottenuta, è possibile procedere con le analisi statistiche al fine di classificare e raggruppare i testi in questione (cioè i paragrafi delle tre opere, che costituiscono le righe del nostro *dataframe*). In particolare, ricercheremo somiglianze e differenze tra le orazioni non solo a livello di contenuti (aspetto, tra l'altro, già emerso in fase di analisi descrittiva), ma anche soprattutto a livello stilistico, in modo da poter valutare l'autenticità dell'*Epitafio* rispetto alla *Contro Eratostene* e alla *Per l'uccisione di Eratostene*.

Per fare ciò utilizzeremo metodi di classificazione supervisionata, ricorrendo alle regressioni lasso e ridge per una distribuzione multinomiale. Successivamente valuteremo tecniche di *cluster analysis* non supervisionate, conside-

rando prima un approccio di tipo modellistico attraverso la *Latent Dirichlet Allocation* e poi strumenti classici, come il *clustering* gerarchico.

2.3.1 Regressione lasso e ridge

Come prima analisi abbiamo provato a classificare attraverso un modello di regressione i nostri testi, ossia i paragrafi delle orazioni, avendo come variabile risposta una variabile categoriale a tre livelli, che identifica le tre opere, e come variabili esplicative i termini risultanti dalla precedente fase di elaborazione dei documenti. Dato l'elevato numero di parametri p, abbiamo deciso di controllare la complessità del modello attraverso metodi di regolarizzazione (*shrinkage*), che contraggono i coefficienti di regressione β verso zero. Per fare ciò abbiamo utilizzato la libreria `glmnet` (Friedman, Hastie e Tibshirani 2010) di `R`, che permette di stimare un modello lineare generalizzato con la penalizzazione *elastic net*.

Supponiamo che la variabile risposta abbia K livelli $G = \{1, 2, ..., K\}$, il modello multinomiale risulta:

$$Pr(G = k|X = x) = \frac{\mathrm{e}^{\beta_{0k}+\beta_k^T x}}{\sum_{l=1}^{K} \mathrm{e}^{\beta_{0l}+\beta_l^T x}}$$

Questo significa che c'è un predittore lineare per ogni classe, ossia, nel nostro caso, per ognuna delle tre opere. Siano Y la matrice $N \times K$ della risposta, con elementi $y_{il} = I(g_i = l)$, e β la matrice $p \times K$ dei coefficienti, dove β_k si riferisce alla k-esima colonna (livello k della risposta) e β_j alla j-esima riga (vettore di K coefficienti per la variabile j). Allora la funzione di log-verosimiglianza negativa penalizzata con *elastic net* diventa:

$$l(\{\beta_{0k}, \beta_k\}_1^K) = -\left[\frac{1}{N}\sum_{i=1}^{N}\left(\sum_{k=1}^{K} y_{il}(\beta_{0k} + x_i^T\beta_k) - \log\left(\sum_{l=1}^{K}\mathrm{e}^{\beta_{0l}+x_i^T\beta_l}\right)\right)\right] +$$

$$\lambda\left[(1-\alpha)||\beta||^2/2 + \alpha\sum_{j=1}^{p}||\beta_j||\right]$$

dove $\lambda \geq 0$ è il parametro che regola la complessità del modello e $0 \leq \alpha \leq 1$

rappresenta un compromesso tra la regressione ridge ($\alpha = 0$) e la regressione lasso ($\alpha = 1$). Si tratta, quindi, di minimizzare tale funzione obiettivo rispetto a (β_{0k}, β_k).

Consideriamo inizialmente la regressione lasso, che utilizza una contrazione (*shrinking*) in valore assoluto. Prima di procedere con la stima del modello sui nostri dati, abbiamo suddiviso casualmente le osservazioni in un insieme di stima (di dimensione pari a 197 osservazioni) e in uno di verifica (di dimensione pari a 34 osservazioni), con l'intento di valutare successivamente la bontà di adattamento del modello evitando il sovradattamento. Inoltre, al fine di scegliere il valore ottimo per il parametro λ, visto il numero esiguo di osservazioni presenti nel nostro insieme di stima (197 righe, corrispondenti ai paragrafi delle opere) rispetto al numero di variabili (2520 termini), il modello è stato stimato ricorrendo alla convalida incrociata a dieci gruppi. Abbiamo dunque utilizzato la funzione `cv.glmnet()`, specificando l'argomento `family = "multinomial"` e `alpha = "1"`. Il numero di gruppi (*folds*), `nfolds`, per la convalida incrociata è stato scelto pari a 10 (*default*), mentre come funzione di costo da minimizzare sono state considerate sia la logverosimiglianza (`type.measure = "deviance"`) sia il tasso di errata classificazione (`type.measure = "class"`). I grafici in Figura 2.3 riportano rispettivamente il valore della logversimiglianza multinomiale e del tasso di errata classificazione (punti rossi) e le loro deviazioni standard (barre di errore grigie) al variare di $\log(\lambda)$. Le linee verticali tratteggiate indicano due particolari valori di λ: quello a sinistra è il valore di λ che fornisce il minimo errore di convalida incrociata, l'altro, più a destra, corrisponde al valore di λ tale per cui l'errore di convalida incrociata è pari al minimo più il suo errore standard. A questo punto abbiamo calcolato le previsioni in corrispondenza di entrambe queste quantità sull'insieme di verifica. In ogni caso, valutando le tabelle di errata classificazione, si perviene a risultati analoghi: i testi provenienti dalla *Contro Eratostene* vengono tutti classificati correttamente, quelli della *Per l'uccisione di Eratostene* vengono in parte assimilati alla precedente orazione (ma ciò, come spiegheremo poco più avanti, non rappresenta per forza un problema), mentre i paragrafi dell'*Epitafio* sono classificati correttamente solo al 50% (l'altra metà dei testi rientra nella *Contro Eratostene*). Viene riportata la tabella con il tasso di corretta classificazione leggermente miglio-

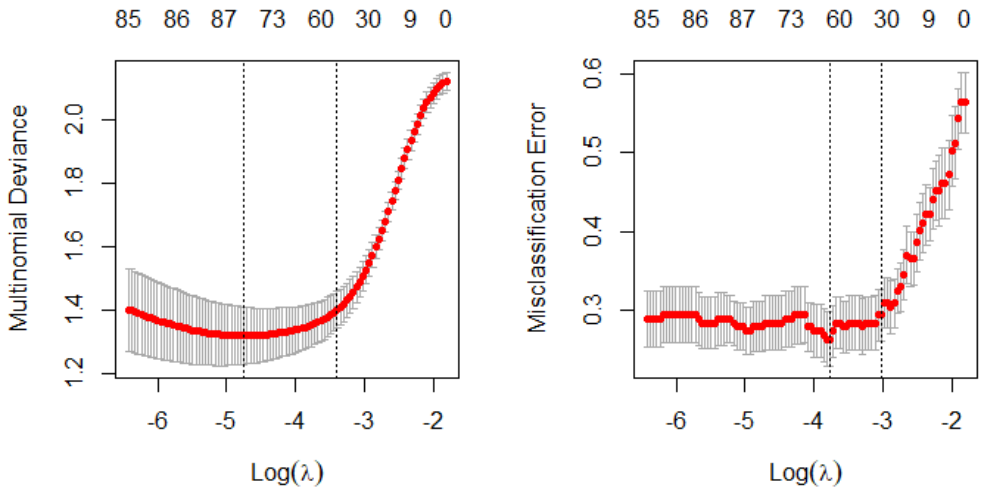

Figura 2.3: Curve delle funzioni di costo considerate per la regressione lasso. A sinistra, logverosimiglianza multinomiale e, a destra, tasso di errata classificazione

re (pari a circa 76.5%), che corrisponde alla previsione basata sul modello stimato utilizzando come funzione di costo per la convalida incrociata il tasso di errata classificazione e con λ pari a `lambda.1se` (Tabella 2.1). Da notare il fatto che né i testi della *Contro Eratostene* né quelli della *Per l'uccisione di Eratostene* vengono mai classificati come provenienti dall'*Epitafio*.

Previsione	Risposta effettiva		
	Contro E.	*Per l'uccisione di E.*	*Epitafio*
Contro E.	14	3	5
Per l'uccisione di E.	0	7	0
Epitafio	0	0	5

Tabella 2.1: Tabella di errata classificazione per la regressione lasso

Consideriamo ora, invece, la regressione ridge, che utilizza una contrazione (*shrinking*) quadratica, e procediamo nello stesso modo descritto precedentemente per la regressione lasso, con l'unica differenza che specifichiamo `alpha = 0` nel comando `cv.glmnet()` (Figura 2.4). Le tabelle di errata classi-

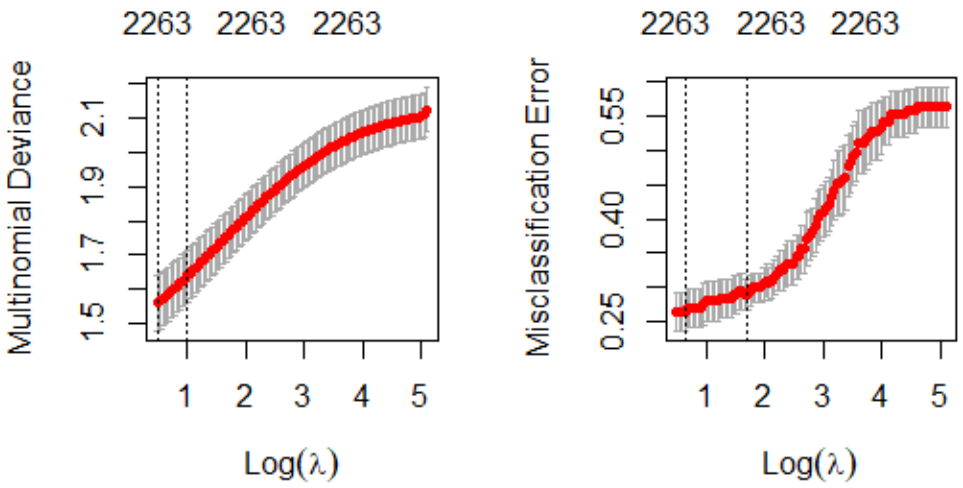

Figura 2.4: Curve delle funzioni di costo considerate per la regressione ridge. A sinistra, logverosimiglianza multinomiale e, a destra, tasso di errata classificazione

ficazione sono identiche utilizzando la logverosimiglianza o il tasso di errata classificazione usando entrambi i criteri di scelta per λ (Tabella 2.2). Il tasso di corretta classificazione, pari al 70%, è peggiore rispetto a quello ottenuto con la lasso e soprattutto potremmo essere portati a valutare negativamente il fatto che il modello non sia minimamente in grado di riconoscere i testi provenienti dalla *Per l'uccisione di Eratostene*, in quanto li classifica nella *Contro Eratostene*, nonostante le due opere trattino argomenti molto diversi tra loro. In realtà, però, ricordando che buona parte della critica ritiene che l'*Epitafio* sia spurio, questo risultato, unitamente a quanto detto in conclusione sulla lasso, potrebbe suggerirci che in effetti *Per l'uccisione di Eratostene* sia assimilabile dal punto di vista della lingua e dello stile alla *Contro Eratostene*, in quanto prodotte dallo stesso autore, Lisia. Pertanto queste due orazioni rientrano nella medesima categoria, mentre l'*Epitafio*, pur trattando tematiche simili alla *Contro Eratostene*, costituisce una categoria a sé, essendo profondamente diverso nella forma dalle altre opere.

Infine, abbiamo valutato le previsioni fornite dal modello al variare del

Previsione	Risposta effettiva		
	Contro E.	*Per l'uccisione di E.*	*Epitafio*
Contro E.	14	10	0
Per l'uccisione di E.	0	0	0
Epitafio	0	0	10

Tabella 2.2: Tabella di errata classificazione per la regressione ridge

parametro di penalità di *elastic net*, provando valori di α compresi tra 0 (regressione ridge) e 1 (regressione lasso), in particolare ponendo α pari a 0.1, 0.3, 0.5, 0.7 e 0.9. Il valore di α per cui si ha la classificazione migliore è 0.1 con un tasso di corretta classificazione pari all'85.3%. In particolare, vengono classificate correttamente tutte le osservazioni provenienti dalla *Contro Eratostene* e dall'*Epitafio*, mentre i paragrafi della *Per l'uccisione di Eratostene* per metà sono classificati correttamente e per la restante metà vengono assimilati all'altra orazione certamente autentica, la *Contro Eratostene*. Per gli altri valori di α si hanno risultati analoghi a quelli ottenuti con la regressione ridge.

2.3.2 Latent Dirichlet Allocation

Un modello statistico molto utilizzato nell'analisi testuale è il Latent Dirichlet Allocation, introdotto da Blei 2003. Si tratta di un tipo di analisi di raggruppamento basata su modelli, il cui scopo è quello di individuare gli argomenti principali, definiti *topics*, che costituiscono un documento.

Supponiamo di avere un *corpus* costituito da n documenti e supponiamo un numero di K *topics* per tale *corpus*. Ogni documento è costituito da n_i parole e ogni parola è indicata con w_{ij}. Ad ogni parola è associata una variabile indicatrice z_{ij}, tale che $z_{ij} = k$ indica che la parola w_{ij} appartiene al *topic* k. I *topics* hanno una distribuzione a priori multinomiale di parametro θ_i, mentre ogni parola w_{ij} ha una distribuzione a priori $F(\phi_{z_{ij}})$. Solitamente $F(\phi_{z_{ij}}) \sim Mult(\phi_k)$, dato che definisce la distribuzione delle parole nel *topic* indicato da $z_{ij} = k$. A loro volta si assume che i parametri θ_i e ϕ_k abbiano una distribuzione a priori di Dirichlet con rispettivi iperparametri $\alpha = \alpha_1, ..., \alpha_K$ e $\beta = \beta_1, ..., \beta_V$, dove V è il numero di parole del vocabolario.

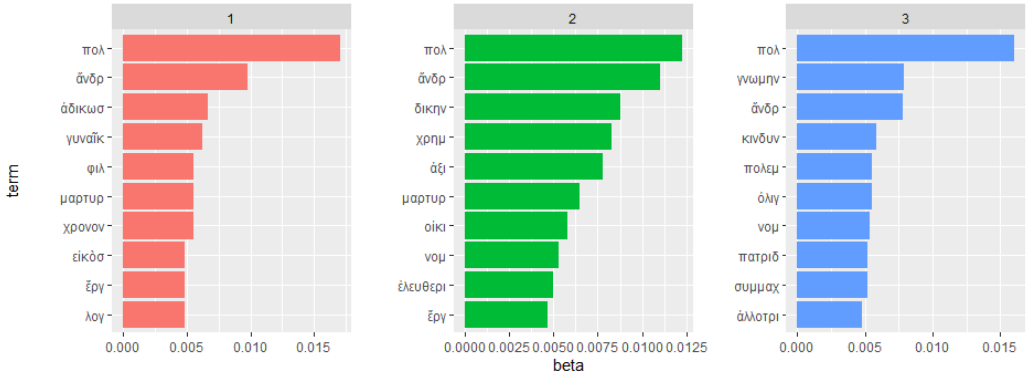

Figura 2.5: Confronto dei termini nei tre *topics*. Sull'asse delle ascisse ritroviamo i termini, mentre sull'asse delle ordinate vengono rappresentate le stime della probabilità di appartenenza ad uno dei 3 argomenti.

Il modello LDA assume il seguente processo generativo per ogni documento i in un *corpus*:

- Si estrae $\theta_i \sim Dir(\alpha)$, dove $i \in 1, ..., n$ e $Dir(\alpha)$ è la distribuzione di Dirichlet per il parametro α
- Si estrae $\phi_k \sim Dir(\beta)$, dove $k \in 1, ..., K$ e $Dir(\beta)$ è la distribuzione di Dirichlet per il parametro β
- Per ogni valore i, j, dove $i \in 1, ..., n$ e $j \in 1, ..., n_i$:
  - Si estrae un *topic* da $z_{ij} \sim Mult(\theta_i)$
  - Si estrae una parola $w_{ij} \sim Mult(\phi_{z_{ij}})$

Per calcolare la probabilità che un termine sia generato da un determinato argomento (β) e la probabilità di ogni argomento all'interno dei documenti (α) abbiamo utilizzato la libreria `topicmodels` (Grün e Hornik 2011), che permette di stimare il modello tramite il comando `LDA()`. Abbiamo assunto la presenza di 2 e 3 *topics*. Nel primo caso, tuttavia, ciascun documento risultava attribuito a uno dei due argomenti con un livello di probabilità piuttosto basso, inferiore al 57%.

Nel secondo caso, invece, i 10 termini che maggiormente influiscono sui tre argomenti sono riportati in Figura 2.5. Notiamo che i termini presenti in

Topic	Paragrafi
Topic 1	32, 5, 46, 116, 4, 61, 195, 7, 74, 108, 225, 211, 150, 180, 34, 138, 145, 2, 19, 38, 149
Topic 2	101, 15, 213, 177, 100, 117, 204, 199, 118, 167, 175, 14, 129, 139, 29, 33, 83, 210, 71, 93, 165, 222
Topic 3	21, 159, 6, 10, 23, 27, 107, 196, 1, 133, 151, 40, 64, 67, 221, 62, 97, 205, 16, 44, 125, 173, 178, 198

Tabella 2.3: Paragrafi delle opere ripartiti nei vari *topics*. I numeri da 1 a 81 si riferiscono ai paragrafi dell'*Epitafio*, quelli da 82 a 181 alla *Contro Eratostene* e quelli da 182 a 231 alla *Per l'uccisione di Eratostene*

tutti e tre i *topics* sono πόλις ("città") e ανήρ ("uomo"), che, come abbiamo già detto, sono parole piuttosto comuni. La prima, infatti, rappresenta un'istituzione molto importante per la realtà ateniese del tempo, a cui pertanto si faceva spesso riferimento in occasioni pubbliche, come quelle in cui venivano pronunciate le orazioni. La seconda, invece, poteva essere utilizzata sia per appellarsi direttamente ai giudici, che ascoltavano le orazioni di genere giudiziario, sia per indicare gli uomini valorosi caduti in battaglia (nel caso dell'*Epitafio*). Nel primo *topic* sono poi presenti termini quali εργόν ("opera") e λόγος ("discorso"), che potrebbero genericamente riferirsi all'*Epitafio*, e parole come γυνή ("donna") e μάρτυρος ("testimone"), che sembrano invece indicare la *Per l'uccisione di Eratostene*. Nel secondo *topic* troviamo termini come δίκη ("giustizia") e χρήμα ("beni", "soldi", "patrimonio"), che sembrano identificare la *Contro Eratostene*. Nel terzo *topic* compaiono parole come γνώμη ("opinione", spesso intesa come "buona opinione", "fama"), κινδύνος ("pericolo"), πολεμός ("guerra") e συμμάχομαι ("combattere"), che indicano abbastanza chiaramente l'*Epitafio*.

Abbiamo infine calcolato per ogni paragrafo delle opere analizzate ($i = 1, ..., 231$) i valori delle probabilità che l'i-esimo paragrafo fosse associato al k-esimo argomento ($k = 1, 2, 3$) e abbiamo considerato la probabilità più alta che superasse una soglia pari a 0.80. Il raggruppamento risultante (Tabella 2.3) mostra che il primo argomento è rappresentato maggiormente da paragrafi provenienti dall'*Epitafio*, così come il terzo, mentre nel secondo

argomento vengono inseriti principalmente i paragrafi della *Contro Eratostene*. In nessuno dei *topics* sembra invece emergere chiaramente l'orazione *Per l'uccisione di Eratostene*.

2.3.3 Clustering gerarchico

Come ultima analisi abbiamo cercato di raggruppare i paragrafi tratti dalle opere considerate, effettuando un *clustering* gerarchico. Esso, a differenza di quello non gerarchico, si basa su un algoritmo di raggruppamento che calcola una matrice di distanza tra le osservazioni, ossia tra le righe del nostro *dataframe*. In particolare abbiamo eseguito un *clustering* gerarchico agglomerativo: si tratta di un approccio *bottom-up* (dal basso verso l'alto), in cui inizialmente ci sono n singoli gruppi formati da un solo elemento che vengono sequenzialmente raggrupati in gruppi via via più grandi.

Il primo passaggio importante consiste nella scelta della metrica da utilizzare per il calcolo della matrice delle distanze. La letteratura si divide sulla scelta di un'appropriata metrica per l'analisi testuale: alcuni sostengono che sia giusto utilizzare la distanza euclidea (Ordonez 2003), mentre altri lo sconsigliano, in quanto le variabili considerate (cioè le frequenze dei termini) non sono continue.

Abbiamo, dunque, valutato anche la distanza di Jaccard, che misura la dissimilarità tra due osservazioni nel caso in cui le variabili siano binarie (presenza/assenza dei termini). Essa è complementare all'omonimo indice di similarità, nel senso che si ottiene sottraendo a 1 il coefficiente di similarità di Jaccard. In riferimento alla Tabella di contingenza 2.4, questo è pari a:

$$S_{ij} = \frac{a}{a+b+c}$$

Infine, abbiamo considerato anche un'altra misura di dissimilarità, il coefficiente di Gower:

$$d_{ij} = \frac{\sum_{k=1}^{p} w_k \delta_{ij;k} d_{ij;k}}{\sum_{k=1}^{p} w_k \delta_{ij;k}}$$

Osservazione j	Osservazione i	
	1	0
1	a	b
0	c	d

Tabella 2.4: Tabella di contingenza

dove d_k è il peso della k-esima variabile; $\delta_{ij;k}$ vale 0 se la k-esima variabile manca nell'i-esima o nella j-esima o in entrambe le osservazioni, 1 altrimenti; $d_{ij;k}$ è il contributo della k-esima variabile alla dissimilarità totale. In caso di variabili binarie, esso vale 0 se la k-esima variabile vale 1 sia nell'i-esima che nella j-esima osservazione, 0 altrimenti; in caso di variabili continue, esso è pari alla differenza in valore assoluto tra il valore della variabile nelle due osservazioni, diviso per il campo di variazione della variabile.

Una volta ottenuta la matrice delle distanze tra le osservazioni, bisogna stabilire il modo in cui calcolare le distanze tra i gruppi che vengono a formarsi durante la costruzione agglomerativa del dendrogramma. Abbiamo deciso di utilizzare il metodo di Ward (Ward Jr 1963). Siano n il numero di osservazioni, p il numero di variabili e g il numero di gruppi o *cluster*, per la devianza vale la seguente formula di scomposizione:

$$Dev(totale) = Dev(tra) + Dev(dentro)$$

$$\sum_{k=1}^{p}\sum_{i=1}^{n}(x_{ik} - \bar{x}_k)^2 = \sum_{k=1}^{p}\sum_{j=1}^{g}(\bar{x}_{kj} - \bar{x}_k)^2 n_j + \sum_{j=1}^{g}\sum_{k=1}^{p}\sum_{i=1}^{n}(x_{ik} - \bar{x}_{kj})^2$$

Secondo il metodo di Ward ad ogni passo si aggregano tra loro quei gruppi per cui vi è il minor incremento della devianza dentro i gruppi o, analogamente, il maggior decremento della devianza tra i gruppi. Abbiamo applicato tale metodo sia alla matrice di distanza euclidea sia, basandoci su quanto riportato in Akay e Yüksel 2018, a quella di Jaccard e di Gower, sebbene molti non ritengano corretto calcolare la devianza (necessaria per il metodo di Ward) su misure di similarità o dissimilarità.

In `R` è possibile ottenere la matrice delle distanze per i nostri dati attraverso il comando `dist()`, specificando l'argomento `method = "euclidean"` oppure `method = "binary"`, che corrisponde alla distanza di Jaccard. La distan-

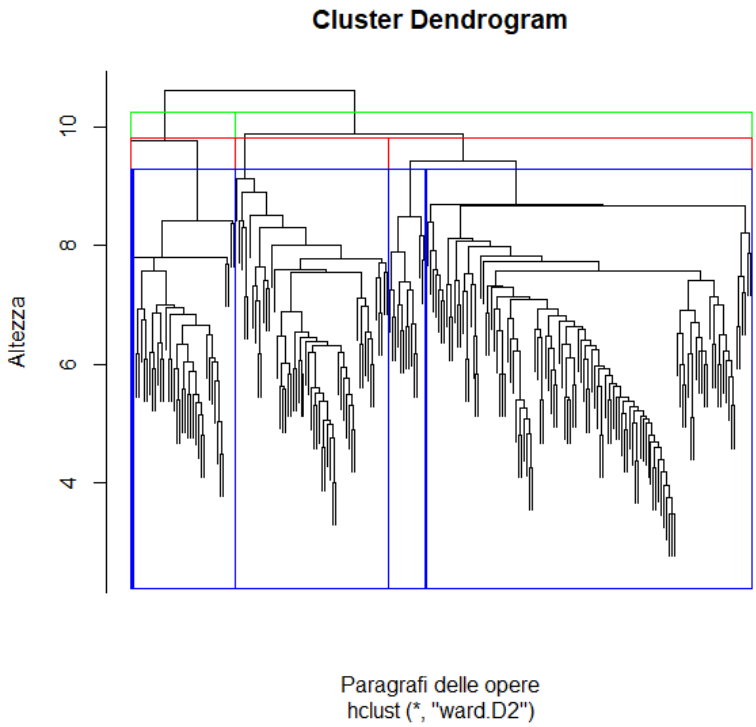

Figura 2.6: Dendrogramma con distanza euclidea e metodo di Ward

za di Gower è, invece, calcolabile tramite la funzione `daisy()` della libreria `cluster` (Maechler et al. 2019), specificando `metric = "gower"`. Si effettua, poi, il *clustering* gerarchico agglomerativo attraverso la funzione `hclust()`, specificando la matrice di distanza e `method = "ward.D2"`. Il risultato finale non fornisce una singola partizione delle n unità, ma una serie di partizioni nidificate che possono essere rappresentate graficamente attraverso un *dendrogramma* (diagramma ad albero), nel quale sull'asse delle ordinate viene riportato il livello di distanza, mentre sull'asse delle ascisse vengono riportate le singole unità.

Consideriamo in prima battuta i risultati ottenuti utilizzando la distanza euclidea e il metodo di Ward. Il dentrogramma in Figura 2.6 permette di visualizzare dei possibili raggruppamenti delle osservazioni (i paragrafi delle opere considerate) in due (linee verdi), tre (linee rosse) o cinque gruppi (linee blu). Nelle tabelle seguenti è possibile confrontare i gruppi di paragrafi derivanti da tale *clustering* con le tre orazioni in analisi. Nella Tabella 2.5

Opera	Cluster	
	1	2
Contro E.	68	32
Per l'uccis. di E.	48	2
Epitafio	76	5

Tabella 2.5: Due gruppi, distanza euclidea

Opera	Cluster		
	1	2	3
Contro E.	45	23	32
Per l'uccis. di E.	20	28	2
Epitafio	70	6	5

Tabella 2.6: Tre gruppi, distanza euclidea

Opera	Cluster				
	1	2	3	4	5
Contro E.	45	23	31	0	1
Per l'uccisione di E.	20	28	2	0	0
Epitafio	56	6	5	14	0

Tabella 2.7: Cinque gruppi, distanza euclidea

notiamo che il gruppo 1 è costituito dalla maggior parte dei paragrafi provenienti da tutte le opere, mentre il gruppo 2 contiene principalmente alcune osservazioni della *Contro Eratostene*. Nella Tabella 2.6 viene a crearsi un nuovo gruppo, oltre ai due precedentemente descritti, formato da un insieme di paragrafi sia della *Contro Eratostene* sia della *Per l'uccisione di Eratostene*, che sono le orazioni di Lisia ritenute certamente autentiche. Considerando quattro gruppi si ottiene un singoletto, tuttavia, se ne consideriamo cinque, nella Tabella 2.7 possiamo visualizzare un gruppo formato unicamente da paragrafi dell'*Epitafio*, l'opera probabilmente spuria.

Delle conclusioni simili si possono trarre anche dal *clustering* effettuato utilizzando la distanza di Jaccard e il metodo di Ward, come si può vedere dal dendrogramma in Figura 2.7. Infatti, dalla Tabella 2.8 emergono nuovamente due gruppi, uno dato da osservazioni provenienti da tutte le opere, un altro costituito per lo più da paragrafi delle due orazioni autentiche. Prendendo poi in considerazione tre gruppi, nella Tabella 2.9 si nota ancora il gruppo contenente osservazioni dalla *Contro Eratostene*. Infine, con quattro gruppi nella Tabella 2.10 possiamo cogliere un gruppo formato quasi esclusivamente da paragrafi dell'*Epitafio*.

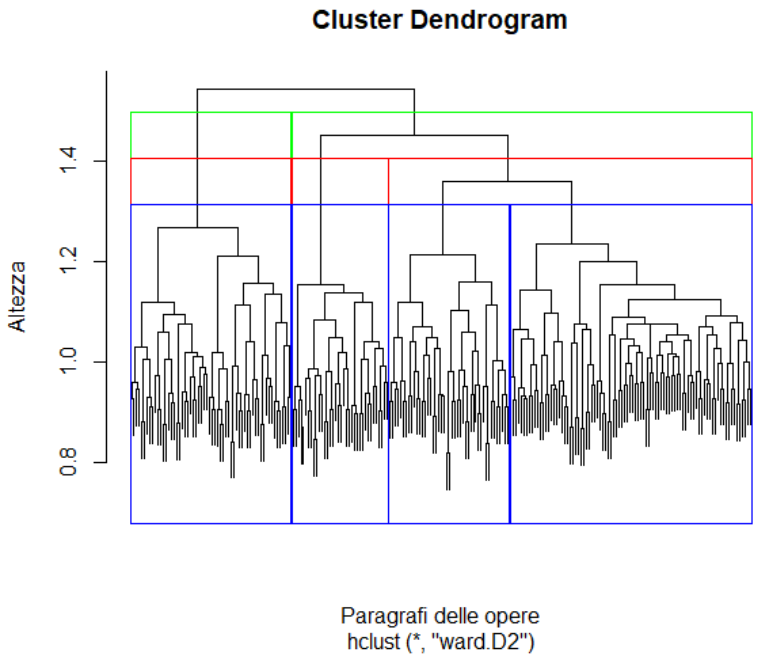

Figura 2.7: Dendrogramma con distanza di Jaccard e metodo di Ward

Opera	Cluster	
	1	2
Contro E.	74	26
Per l'uccis. di E.	20	30
Epitafio	77	4

Tabella 2.8: Due gruppi, distanza di Jaccard

Opera	Cluster		
	1	2	3
Contro E.	47	26	27
Per l'uccis. di E.	18	30	2
Epitafio	70	4	7

Tabella 2.9: Tre gruppi, distanza di Jaccard

Opera	Cluster			
	1	2	3	4
Contro E.	45	26	2	27
Per l'uccisione di E.	17	30	1	2
Epitafio	28	4	42	7

Tabella 2.10: Quattro gruppi, distanza di Jaccard

Risultati leggermente diversi si ottengono, invece, considerando il *clustering* con la distanza di Gower e il metodo di Ward. Dal dendrogramma in Figura 2.8 emergono abbastanza chiaramente due gruppi, uno molto più grande dell'altro. Dalla Tabella 2.11 si evince che il primo è, come sempre, costituito dalla maggior parte delle osservazioni provenienti da tutte le opere, mentre il secondo è dato per lo più da paragrafi dell'*Epitafio*. Da notare il fatto che, aumentando il numero di *cluster*, in questo caso vengono a formarsi singoletti, e non altri gruppi. Questa analisi evidenzia, quindi, la presenza di un gruppo di testi nettamente differenti dalla restante maggioranza, risultato che sembra supportare l'ipotesi che l'*Epitafio*, sebbene presenti somiglianze con le altre orazioni di Lisia, non sia tuttavia un'opera autentica dell'autore.

In conclusione, a seconda della distanza utilizzata il *clustering* gerarchico ha messo in evidenza l'esistenza di principalmente tre gruppi: uno costituito da osservazioni provenienti da tutte le opere considerate, indice del fatto che esse sono effettivamente simili tra loro (ma d'altronde, se non lo fossero state, non ci saremmo nemmeno dovuti interrogare sulla paternità dell'*Epitafio*), un altro contenente testi dalla *Contro Eratostene* e dalla *Per l'uccisione di Eratostene*, rappresentativo, dunque, delle orazioni autentiche, e un ultimo formato da paragrafi dell'*Epitafio*, che a questo punto potremmo essere portati a considerare spurio.

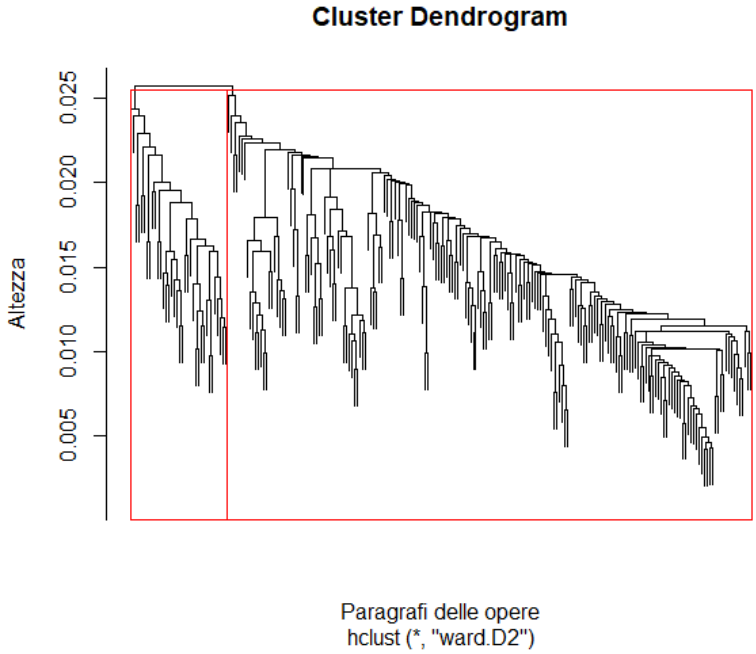

Figura 2.8: Dendrogramma con distanza di Gower e metodo di Ward

Opera	Cluster	
	1	2
Contro E.	100	0
Per l'uccisione di E.	49	1
Epitafio	46	35

Tabella 2.11: Due gruppi, distanza di gower

Conclusione

In questo elaborato abbiamo analizzato attraverso una serie di tecniche statistiche tre opere dell'oratore greco Lisia fra loro molto diverse sia negli argomenti trattati sia nel genere. Lo scopo principale delle nostre analisi era quello di indagare le somiglianze e le differenze non solo contenutistiche ma anche stilistiche tra i discorsi *Contro Eratostene*, *Per l'uccisione di Eratostene* e *Epitafio* e, in particolare, abbiamo cercato di affrontare la questione dell'autenticità di quest'ultimo. Per fare ciò siamo ricorsi a metodi di classificazione e di raggruppamento di testi.

In primo luogo abbiamo effettuato un'analisi di classificazione attraverso la regressione penalizzata con *elastic net* e abbiamo notato che con un α pari a 0.1 è possibile distinguere piuttosto chiaramente le tre opere (a eccezione di alcuni paragrafi della *Per l'uccisione di Eratostene* che vengono assimilati alla *Contro Eratostene*). Considerando invece una contrazione quadratica (ridge) si nota che i paragrafi provenienti dalla *Contro Eratostene* e dalla *Per l'uccisione di Eratostene* vengono classificati nella medesima categoria, mentre l'*Epitafio* costituisce una categoria a sé stante.

Successivamente abbiamo valutato una tecnica di raggruppamento basata su modelli, la Latent Dirichlet Allocation. Tuttavia in questo caso, ipotizzando la presenza di tre *topics*, siamo riusciti a distinguere solo la *Contro Eratostene* e l'*Epitafio*, mentre la *Per l'uccisione di Eratostene* non sembra rappresentare alcun argomento in particolare.

Infine abbiamo effettuato un *clustering* gerarchico agglomerativo. A seconda della matrice di distanze utilizzata e del numero di gruppi fissato, abbiamo ottenuto diversi possibili raggruppamenti dei testi. In particolare, è interessante notare che tendenzialmente un gruppo è costituito da paragra-

fi provenienti dalla *Contro Eratostene* e dalla *Per l'uccisione di Eratostene*, mentre un altro gruppo è dato da paragrafi dell'*Epitafio*.

I risultati delle diverse analisi potrebbero pertanto sembrare contraddittori, specialmente se confrontiamo le conclusioni tratte dalla Latent Dirichlet Allocation con quelle tratte dalla regressione multinomiale e dal *clustering* gerarchico. Tuttavia è bene ricordare che la Latent Dirichlet Allocation è una tecnica che raggruppa i testi sulla base degli argomenti (*topics*) in essi presenti. La *Contro Eratostene* e l'*Epitafio* sono opere che affrontano chiaramente determinate tematiche (giustizia e democrazia la prima, valore militare e guerra la seconda), mettendole al centro del discorso, mentre la *Per l'uccisione di Eratostene* si presenta semplicemente come un'orazione difensiva, in cui vengono ricostruiti i fatti precedenti il delitto anche attraverso descrizioni della vita quotidiana dei soggetti coinvolti. Quest'ultima quindi non ruota attorno a un vero e proprio tema a differenza delle altre due e ciò potrebbe spiegare la difficoltà riscontrata nell'individuare un *topic* ad essa relativo attraverso la Latent Dirichlet Allocation.

Di conseguenza, sulla base di queste motivazioni, per valutare l'autenticità dell'*Epitafio* sembra più opportuno considerare i risultati ottenuti dalla regressione multinomiale e dal *clustering* gerarchico. Un altro motivo è dato dal fatto che, come abbiamo spiegato nel Capitolo 1, il dubbio circa la paternità lisiana dell'opera nasce soprattutto in relazione a questioni stilistiche e linguistiche, e non per via di differenze tra l'*Epitafio* e le altre opere nelle tematiche trattate, in quanto sappiamo che anche le orazioni certamente autentiche di Lisia affrontano svariati temi molto diversi fra loro. Pertanto, vista la classificazione delle opere prodotta dalla regressione e visti i gruppi risultanti dall'analisi di raggruppamento, le nostre analisi sembrano suggerire che in effetti l'*Epitafio*, nonostante presenti somiglianze con la *Contro Eratostene* e la *Per l'uccisione di Eratostene*, sia un'opera spuria, in quanto diversa nella lingua e nello stile dalle altre orazioni lisiane.

Codice R

```
library(rperseus)
library(quanteda)
library(tidyverse)
library(tidytext)
library(tm)
library(ggplot2)
library(forcats)



###### caricamento dei dati
FuneralOration <- perseus_catalog %>% filter(group_name == "Lysias",
  language == "grc", label == "Funeral Oration") %>%
  pull(urn) %>% map_df(get_perseus_text)
FuneralOration <- FuneralOration %>% select(-urn) %>%
  select(-group_name) %>% select(-description) %>%
  select(-language) %>% select(-section)

AgainstErat <- perseus_catalog %>% filter(group_name == "Lysias",
  language == "grc", label == "Against Eratosthenes") %>%
  pull(urn) %>% map_df(get_perseus_text)
AgainstErat <- AgainstErat %>% select(-urn) %>% select(-group_name) %>%
  select(-description) %>% select(-language) %>% select(-section)

MurderErat <- perseus_catalog %>% filter(group_name == "Lysias",
  language == "grc", label == "On the Murder of Eratosthenes") %>%
  pull(urn) %>% map_df(get_perseus_text)
MurderErat <- MurderErat %>% select(-urn) %>% select(-group_name) %>%
```

```
  select(-description) %>% select(-language) %>% select(-section)

id <- c(1:231)
lys_tibble <- bind_rows(FuneralOration, AgainstErat, MurderErat) %>%
  add_column(id)


###### tokenizzazione e rimozione delle stopwords
sw <- stopwords::stopwords("grc", source = "ancient")
sw <- c(sw, "bekker", "canter", "cobet", "dobree", "et", "reiske", "sauppe
    ", "taylor", "baiter", "contius", "frohberger", "fuhr", "gebauer", "
    hertlein", "kayser", "scheibe", "auger", "bake", "bizor", "duo", "
    franz", "fritzsche", "fronhberger", "gernet", "hude", "jacobs", "
    lipsius", "madvig", "marklans", "maussac", "plerique", "scaliger", "
    schott", "sluiter", "stephanus", "swzonta", "markland")

lys_words <- tokens(lys_tibble$text, remove_numbers = T,
  remove_punct = T, remove_separators = T) %>% tokens_remove(sw)


###### stemming
lys_words <- as.list(lys_words)
for(i in (1:231)){
lys_words[[i]] <- stemDocument(as.character(lys_words[[i]]), language = "
    greek")
}
lys_words <- as.tokens(lys_words)


###### document term matrix
lys_dfm <- dfm(lys_words)


###### dataframe
opera <- as.factor(c(rep("FuneralOration", 81), rep("AgainstErat", 100),
  rep("MurderErat", 50)))
```

```
lys_dataframe <- convert(lys_dfm, to = "data.frame") %>%
  add_column(opera, .after = "doc_id")


###### tidytext
sw_df <- as.data.frame(sw)
lys_tidy <- lys_tibble %>% unnest_tokens(word, text) %>%
  anti_join(sw_df, by = c("word" = "sw" ), copy = T)

stem_word <- stemDocument(lys_tidy$word, language = "greek")
lys_stemmed <- lys_tidy %>% mutate(word = stem_word)


###### rimozione di parole con apostrofo
countord <- lys_stemmed %>% count(word, sort = T)
lys_dataframe <- lys_dataframe %>% select(-(countord$word[c(1, 2, 13, 22,
    25, 44, 51, 63, 68, 96, 154, 185, 212, 216, 217, 260, 322, 400, 631,
    794, 795, 843, 1093, 1094, 1501, 1640, 1663, 1808, 2070, 2200, 2520,
    2522, 2559, 2560, 694, 2317)]))

temp <- c(sw, countord$word[c(1, 2, 13, 22, 25, 44, 51, 63, 68, 96,
    154,185, 212, 216, 217, 260, 322, 400, 631, 794, 795, 843, 1093, 1094,
     1501, 1640, 1663, 1808, 2070, 2200, 2520, 2522, 2559, 2560, 694,
    2317)])
temp <- as.data.frame(temp)
lys_tidy <- lys_tidy %>% anti_join(temp, by = c("word" = "temp"),
  copy = T)
lys_stemmed <- lys_stemmed %>% anti_join(temp, by = c("word" = "temp"),
  copy = T)


###### stemming pol-polin e politei-politon
countord2 <- lys_stemmed %>% count(word, sort = T)
for (i in (1:4682)){
  if (lys_stemmed$word[i] == countord2$word[3]){
    lys_stemmed$word[i] = countord2$word[2]
```

```
  }
}

for (i in (1:4682)){
  if (lys_stemmed$word[i] == countord2$word[54]){
    lys_stemmed$word[i] = countord2$word[88]
  }
}

which(names(lys_dataframe) == countord2$word[3])
# 114
which(names(lys_dataframe) == countord2$word[2])
#140
lys_dataframe[, 140] <- lys_dataframe[, 140] + lys_dataframe[, 114]
lys_dataframe <- lys_dataframe[, -114]

which(names(lys_dataframe) == countord2$word[54])
# 1349
which(names(lys_dataframe) == countord2$word[88])
#1144
lys_dataframe[, 1144] <- lys_dataframe[, 1144] + lys_dataframe[, 1349]
lys_dataframe <- lys_dataframe[, -1349]



###### tf-idf con e senza stemming
words_count <- lys_stemmed %>% count(label, word, sort = T)
words_total <- words_count %>% group_by(label) %>%
  summarize(total = sum(n))
words_count <- left_join(words_count, words_total)
tf_idf <- words_count %>% bind_tf_idf(word, label, n)

words_count2 <- lys_tidy %>% count(label, word, sort = T)
words_total2 <- words_count2 %>% group_by(label) %>%
  summarize(total = sum(n))
words_count2 <- left_join(words_count2, words_total2)
tf_idf2 <- words_count2 %>% bind_tf_idf(word, label, n)
```

```
###### frequenza dei termini
lys_tidy %>% count(word, sort = TRUE) %>% filter(n > 10) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(n, word)) + geom_col() + labs(y = NULL)

lys_stemmed %>% count(word, sort = TRUE) %>% filter(n > 12) %>%
  filter(n != 31) %>% mutate(word = reorder(word, n)) %>%
  ggplot(aes(n, word)) + geom_col() + labs(y = NULL)

tf_idf %>% group_by(label) %>% slice_max(tf_idf, n = 15) %>% ungroup() %>%
  ggplot(aes(tf_idf, fct_reorder(word, tf_idf), fill = label)) +
  geom_col(show.legend = FALSE) + facet_wrap(~label, ncol = 2,
  scales = "free") + labs(x = "tf-idf", y = NULL)

tf_idf2 %>% group_by(label) %>% slice_max(tf_idf, n = 15) %>%
  ungroup() %>% ggplot(aes(tf_idf, fct_reorder(word, tf_idf),
  fill = label)) + geom_col(show.legend = FALSE) +
  facet_wrap(~label, ncol = 2,
  scales = "free") + labs(x = "tf-idf", y = NULL)


###### rimuovo eratostene e triakonta
lys_dataframe <- lys_dataframe %>% select(-(countord2$word[c(9)]))
lys_dataframe <- lys_dataframe %>% select(-(countord2$word[c(4)]))

temp2 <- data.frame(countord2$word[c(4,9)])
lys_stemmed <- lys_stemmed %>% anti_join(temp2,
by = c("word" = "countord2.word.c.4..9.."), copy = T)


library(tidymodels)
set.seed(123)
split <- initial_split(lys_dataframe, prop = 85/100)
train <- training(split)
```

```
test <- testing(split)



###### lasso
library(glmnet)

# dev e lambda min
cvfit1 <- cv.glmnet(y = train %>% pull(opera), x = train %>%
  select(!doc_id) %>% select(!opera) %>% as.matrix(),
  family = "multinomial")
plot(cvfit1)
prev1 <- predict(cvfit1, newx = test %>% select(!doc_id) %>%
  select(!opera) %>% as.matrix(), s = "lambda.min", type = "class")
table(prev1, test$opera)

# class e lambda min
cvfit1.1 <- cv.glmnet(y = train %>% pull(opera), x = train %>%
  select(!doc_id) %>% select(!opera) %>% as.matrix(),
  family = "multinomial", type.measure = "class")
plot(cvfit1.1)
prev1.1 <- predict(cvfit1.1, newx = test %>% select(!doc_id) %>%
  select(!opera) %>% as.matrix(), s = "lambda.min", type = "class")
table(prev1.1, test$opera)

# dev e lambda 1se
prev12 <- predict(cvfit1, newx = test %>% select(!doc_id) %>%
  select(!opera) %>% as.matrix(), s = "lambda.1se", type = "class")
table(prev12, test$opera)

# class e lambda 1se
prev1.12 <- predict(cvfit1.1, newx = test %>% select(!doc_id) %>%
  select(!opera) %>% as.matrix(), s = "lambda.1se", type = "class")
table(prev1.12, test$opera)



###### ridge
```

```
# dev e lambda min
cvfit0 <- cv.glmnet(y = train %>% pull(opera), x = train %>%
  select(!doc_id) %>% select(!opera) %>% as.matrix(),
  family = "multinomial", alpha = 0)
plot(cvfit0)
prev0 <- predict(cvfit0, newx = test %>% select(!doc_id) %>%
  select(!opera) %>% as.matrix(), s = "lambda.min", type = "class")
table(prev0, test$opera)

# class e lambda min
cvfit0.1 <- cv.glmnet(y = train %>% pull(opera), x = train %>%
  select(!doc_id) %>% select(!opera) %>% as.matrix(),
  family = "multinomial", alpha = 0, type.measure = "class")
plot(cvfit0.1)
prev0.1 <- predict(cvfit0.1, newx = test %>% select(!doc_id) %>%
  select(!opera) %>% as.matrix(), s = "lambda.min", type = "class")
table(prev0.1, test$opera)

# dev e lambda 1se
prev02 <- predict(cvfit0, newx = test %>% select(!doc_id) %>%
  select(!opera) %>% as.matrix(), s = "lambda.1se", type = "class")
table(prev02, test$opera)

# class e lambda 1se
prev0.12 <- predict(cvfit0.1, newx = test %>% select(!doc_id) %>%
  select(!opera) %>% as.matrix(), s = "lambda.1se", type = "class")
table(prev0.12, test$opera)



###### elastic net
# provati alpha = 0.1, 0.3, 0.5, 0.7, 0.9
cvfit0.5 <- cv.glmnet(y = train %>% pull(opera), x = train %>%
  select(!doc_id) %>% select(!opera) %>% as.matrix(),
  family = "multinomial", alpha = 0.1)
plot(cvfit0.5)
prev0.5 <- predict(cvfit0.5, newx = test %>% select(!doc_id) %>%
```

```
  select(!opera) %>% as.matrix(), s = "lambda.min", type = "class")
table(prev0.5, test$opera)



###### LDA
library(topicmodels)
matr <- lys_stemmed %>% count(id, word) %>% cast_dtm(id, word, n)
lys_lda <- LDA(matr, k = 3, control = list(seed=1234))

topics_lys <- tidy(lys_lda, matrix = "beta")
lys_top_terms <- topics_lys %>% group_by(topic) %>%
  slice_max(beta, n = 10) %>% ungroup() %>%
  arrange(topic, -beta)
lys_top_terms %>%
  mutate(term = reorder_within(term, beta, topic)) %>%
  ggplot(aes(beta, term, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  scale_y_reordered()

docs_lys <- tidy(lys_lda, matrix = "gamma")
groups_docs <- docs_lys %>% filter(gamma > 0.8) %>% group_by(topic) %>%
  slice_max(gamma, n = 20) %>% ungroup() %>% arrange(topic, -gamma)
groups_docs$document[groups_docs$topic==1]
groups_docs$document[groups_docs$topic==2]
groups_docs$document[groups_docs$topic==3]



###### clustering gerarchico
eu.dist <- dist(lys_dataframe[,-c(1,2)], method="euclidean")
hcw.eu <- hclust(eu.dist, method="ward.D2")
plot(hcw.eu, labels = F, xlab = "Paragrafi delle opere", ylab = "Altezza")
rect.hclust(hcw.eu, k=2, border="green")
rect.hclust(hcw.eu, k=3, border="red")
rect.hclust(hcw.eu, k=5, border="blue")
table(lys_dataframe$opera, cutree(hcw.eu, k=2))
```

```
table(lys_dataframe$opera, cutree(hcw.eu, k=3))
table(lys_dataframe$opera, cutree(hcw.eu, k=4))
table(lys_dataframe$opera, cutree(hcw.eu, k=5))

bi.dist <- dist(lys_dataframe[,-c(1,2)], method="binary")
hcw.bi <- hclust(bi.dist, method="ward.D2")
plot(hcw.bi, labels = F, xlab = "Paragrafi delle opere", ylab = "Altezza")
rect.hclust(hcw.bi, k=2, border="green")
rect.hclust(hcw.bi, k=3, border="red")
rect.hclust(hcw.bi, k=4, border="blue")
table(lys_dataframe$opera, cutree(hcw.bi, k=2))
table(lys_dataframe$opera, cutree(hcw.bi, k=3))
table(lys_dataframe$opera, cutree(hcw.bi, k=4))

library(cluster)
gower.dist <- daisy(lys_dataframe[, -c(1,2)], metric = "gower")
hcw.gower <- hclust(gower.dist, method = "ward.D2")
plot(hcw.gower, labels = F, xlab = "Paragrafi delle opere",
  ylab = "Altezza")
rect.hclust(hcw.gower, k=2, border = "red")
table(lys_dataframe$opera, cutree(hcw.gower, k=2))

agg.go <- agnes(gower.dist, method = "ward")
table(lys_dataframe$opera, cutree(agg.go, k=2))
```

Bibliografia

Akay, Özlem e Güzin Yüksel (2018). «Clustering the mixed panel dataset using Gower's distance and k-prototypes algorithms». In: *Communications in Statistics-Simulation and Computation* 47.10, pp. 3031–3041.

Benoit, Kenneth et al. (2018). «quanteda: An R package for the quantitative analysis of textual data». In: *Journal of Open Source Software* 3.30, p. 774. DOI: 10.21105/joss.00774. URL: https://quanteda.io.

Berra, Aurélien (2020). *Update ancient Greek and Latin stopwords*. URL: https://github.com/quanteda/stopwords/issues/19.

Blei Ng, Jordan (2003). «Latent dirichlet allocation». In: *the Journal of machine Learning research* 3, pp. 993–1022.

Feinerer, Ingo e Kurt Hornik (2020). *tm: Text Mining Package*. R package version 0.7-8. URL: https://CRAN.R-project.org/package=tm.

Friedman, Jerome, Trevor Hastie e Robert Tibshirani (2010). «Regularization Paths for Generalized Linear Models via Coordinate Descent». In: *Journal of Statistical Software* 33.1, pp. 1–22. URL: https://www.jstatsoft.org/v33/i01/.

Grün, Bettina e Kurt Hornik (2011). «topicmodels: An R Package for Fitting Topic Models». In: *Journal of Statistical Software* 40.13, pp. 1–30. DOI: 10.18637/jss.v040.i13.

Lovins, Julie Beth (1968). «Development of a stemming algorithm.» In: *Mech. Transl. Comput. Linguistics* 11.1-2, pp. 22–31.

Maechler, Martin et al. (2019). *cluster: Cluster Analysis Basics and Extensions*. R package version 2.1.0 — For new features, see the 'Changelog' file (in the package source).

Medda, Enrico (2016). «L'Epitafio e gli ideali democratici di Lisia nella lettura di Giuseppe Schiassi». In: *Paradeigmata* 37, pp. 75–92.

Meyer, David, Kurt Hornik e Ingo Feinerer (2008). «Text mining infrastructure in R». In: *Journal of statistical software* 25.5, pp. 1–54.

Ntais, Georgios (2006). «Development of a Stemmer for the Greek Language». In: *Department of Computer and Systems SciencesMaster Thesis at Stockholm University/Royal Institute of Technology*, pp. 1–40.

Ordonez, Carlos (2003). «Clustering binary data streams with k-means». In: *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, pp. 12–19.

Porter, Martin F (1980). «An algorithm for suffix stripping». In: *Program*.

— (2001). *Snowball: A language for stemming algorithms*.

R Core Team (2020). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria. URL: https://www.R-project.org/.

Ranzolin, David (2021). *rperseus: Get Texts from the Perseus Digital Library*. https://docs.ropensci.org/rperseus, https://github.com/ropensci/rperseus.

Silge, Julia e David Robinson (2016). «tidytext: Text Mining and Analysis Using Tidy Data Principles in R». In: *JOSS* 1.3. DOI: 10.21105/joss.00037. URL: http://dx.doi.org/10.21105/joss.00037.

Todd, Stephen Charles et al. (2007). *A commentary on Lysias, speeches 1-11*. Oxford University Press on Demand, pp. 1–210.

Ward Jr, Joe H (1963). «Hierarchical grouping to optimize an objective function». In: *Journal of the American statistical association* 58.301, pp. 236–244.