RELAZIONE TEXT MINING

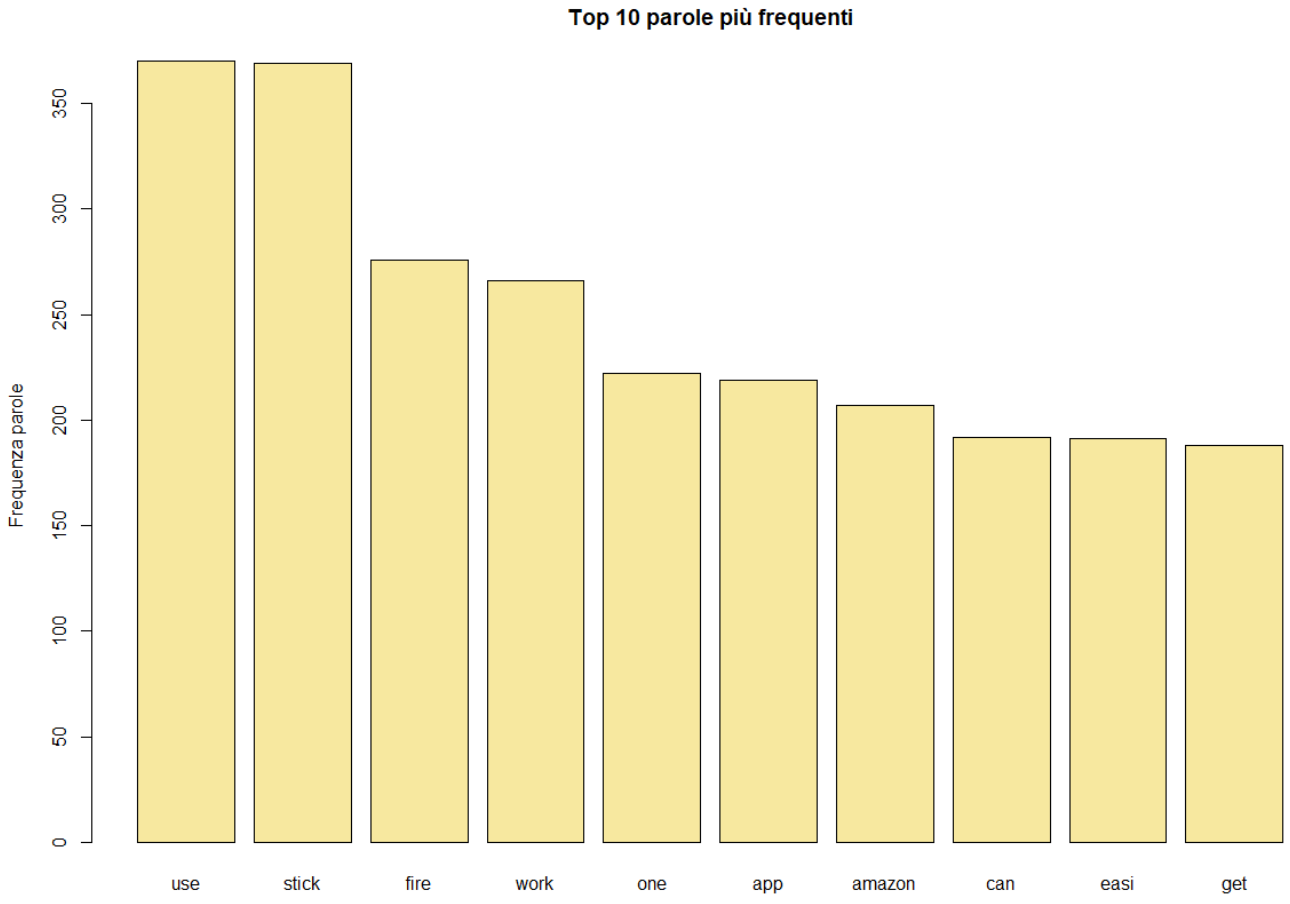
Per lo sviluppo del progetto ho deciso di lavorare su un dataset di recensioni di 500 righe scaricate da Amazon, il prodotto da cui queste ultime sono state ricavate è Alexa Fire Stick TV. Ho utilizzato R per la creazione dei vari grafici e per le varie operazioni sul testo.

1. ANALISI STATISTICHE

Come primo passo ho effettuato delle operazioni di pulizia del testo, attraverso la libreria ‘tm’; ho trasformato il vettore delle recensioni in un corpus rimuovendo le stopwords, la punteggiatura, gli spazi bianchi, i numeri e trasformando tutto il testo in minuscolo, ho poi ridotto attraverso lo stemming le parole dalla loro forma flessa alla forma radice e infine ho trasformato il corpus in una matrice e l’ho ordinata in ordine decrescente per mostrare le parole più utilizzate.

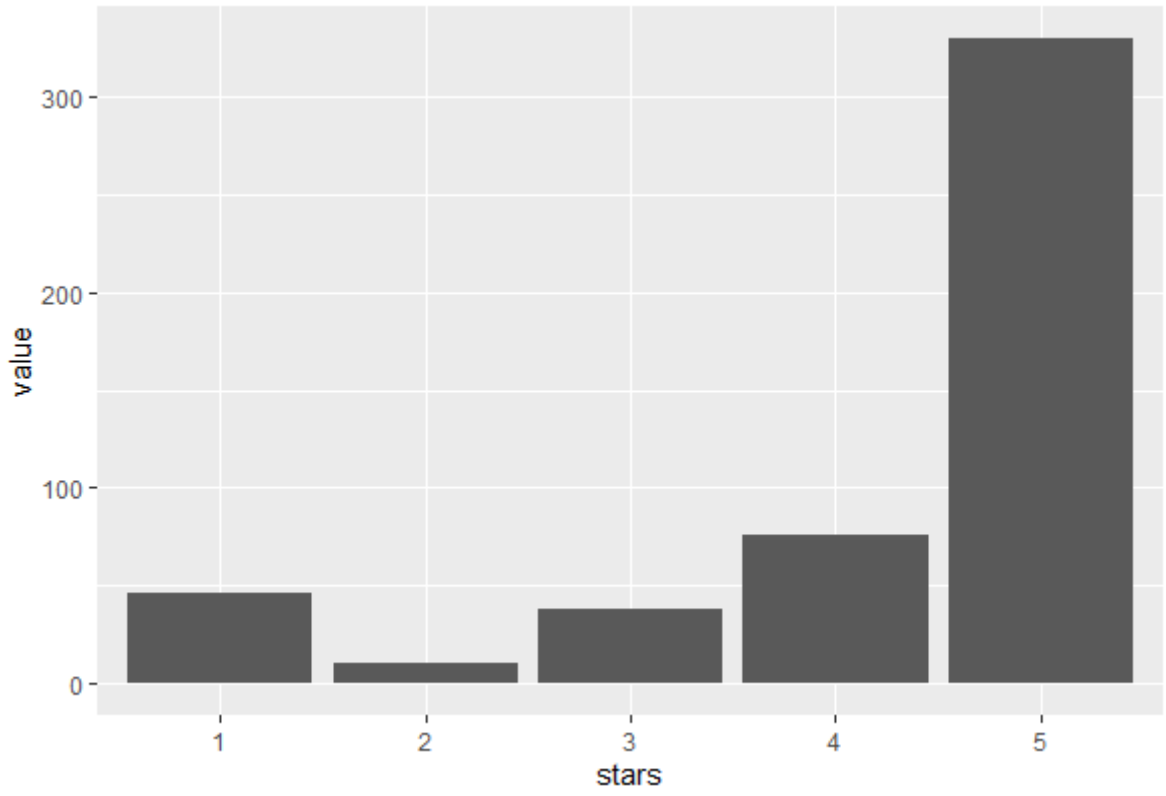
Ho creato due grafici attraverso la libreria ‘ggplot2’ per una rappresentazione visiva e meglio leggibile delle parole più frequenti.

Un grafico a barre che mostra le 10 parole più frequenti.



Una world cloud, attraverso la libreria ‘worldcloud’ e la libreria ‘RColorBrewer’, per visualizzare oltre alle 10 più frequenti anche le altre molto utilizzate.

Dopo la pulizia del testo con relativi grafici sulle frequenze, mi sono incentrato sullo studio del rating e quindi delle stelle, rimuovendo anche qui delle parti del testo non necessarie utilizzando il comando ‘gsub’ e creando un grafico a barre.



Come possiamo notare dal grafico, nella maggior parte delle recensioni le persone hanno assegnato 5 stelle su 5, questo dato ci dice che i vari clienti sono rimasti molto soddisfatti, dobbiamo però verificare che il rating assegnato corrisponda alle recensioni rilasciate dai vari clienti.

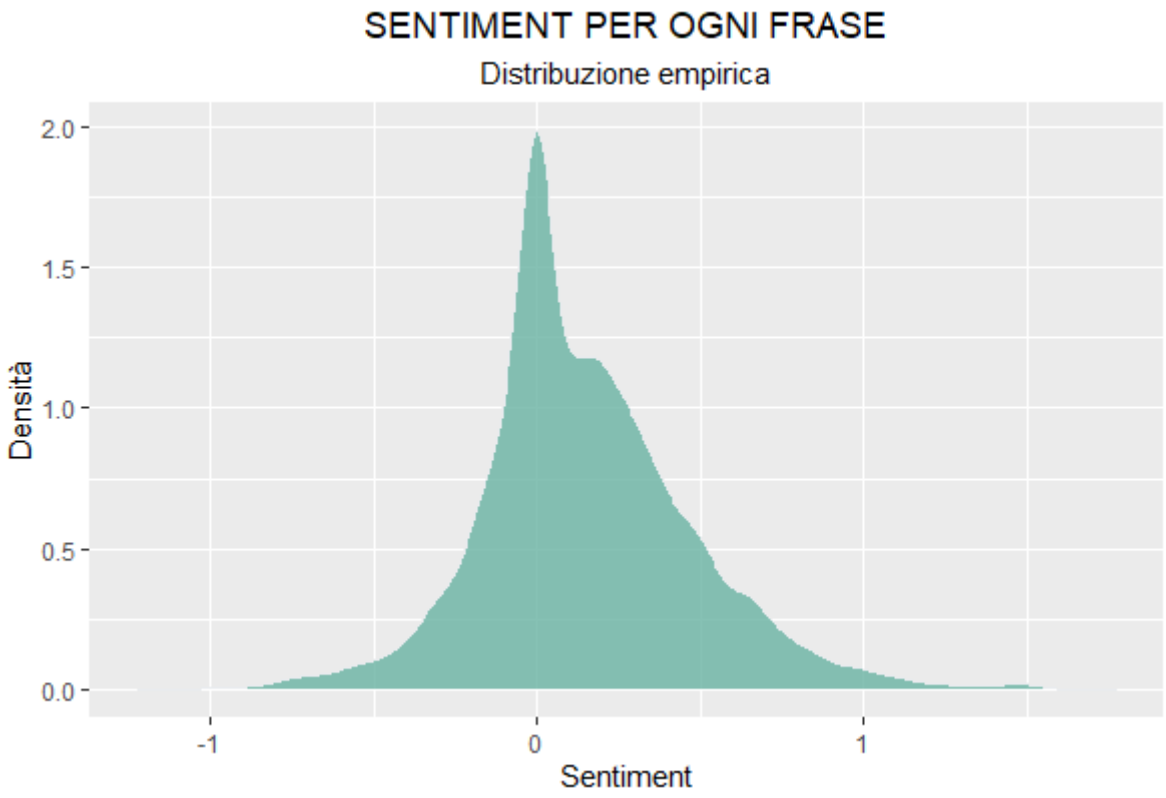
2. SENTIMENT ANALYSIS

La sentiment analysis serve per identificare ed estrarre le opinioni dal testo che possono essere negative o positive.

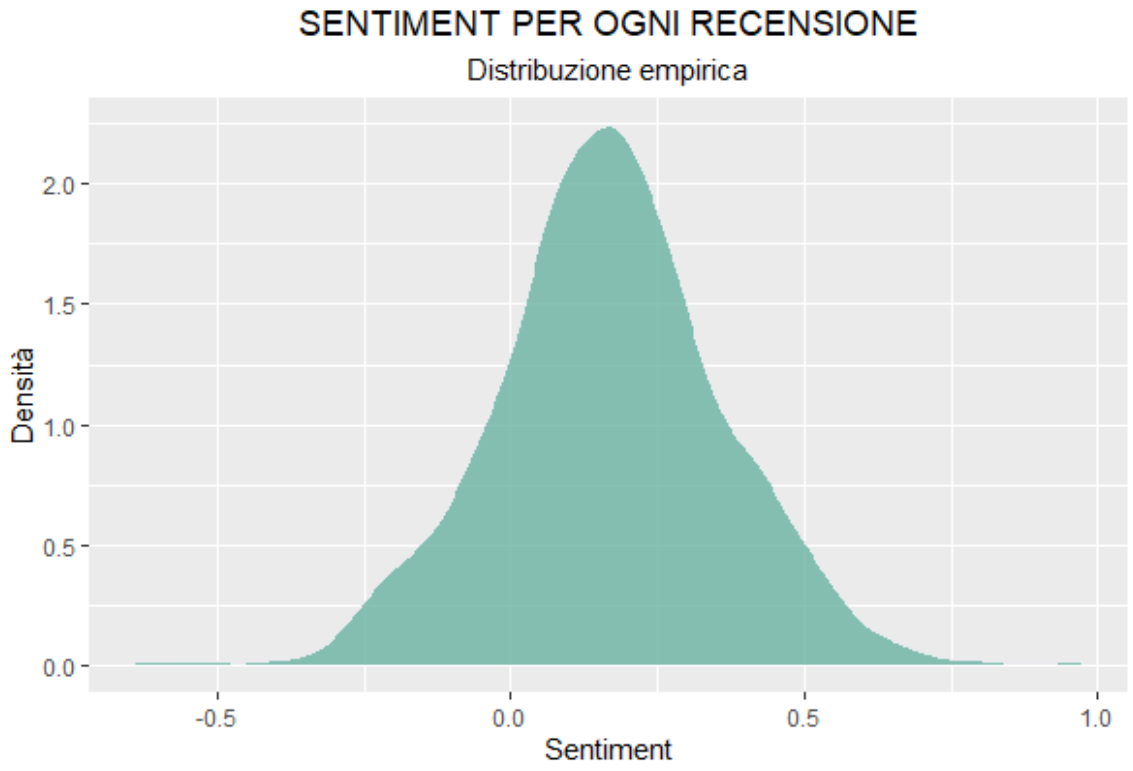
Per verificare che le stelle assegnate al prodotto corrispondano al sentiment delle recensioni ho analizzato il sentiment per ogni singola frase e per ogni recensione completa.

Ho utilizzato la libreria ‘sentimentr’ che scompone ogni frase in un insieme ordinato di parole, rimuove la punteggiatura tranne le virgole, i due punti e i punti e virgola, individua le parole polarizzate e gli assegna un valore.

Anche qui ho creato 2 grafici utilizzando ‘ggplot2’, che mostrano la distribuzione empirica del sentiment in entrambi i casi.



Nel grafico che rappresenta la distribuzione empirica del sentiment per ogni frase possiamo notare una distribuzione quasi simmetrica, con un picco di valori attorno allo 0 da cui poi inizia una discesa che va quasi a stabilizzarsi intorno allo 0.5. Possiamo dire che, basandoci solamente sul sentiment delle singole frasi il nostro prodotto dovrebbe aggirarsi intorno alle 3 stelle evidenziando un sentiment abbastanza neutro, mostrando così una discrepanza dai valori riportati nel grafico del rating.



Nel grafico del sentiment per ogni recensione la curva si comporta quasi come una distribuzione normale, evidenziando un picco di valori tra lo 0 e lo 0.5, mentre i valori presenti nelle code non sono né fortemente negativi né fortemente positivi. Anche in questo caso notiamo una discrepanza con il grafico delle stelle, che se rispecchiasse i valori ricavati da entrambi i sentiment dovrebbe avere un picco di valori attorno alle 3 stelle, mentre il picco è presente attorno alle 5 stelle

Possiamo dire quindi che in questo caso i valori delle stelle non rispecchiano completamente il sentiment delle recensioni, mentre nelle stelle si tende ad incrementare il valore del prodotto e quindi a dare un rate più alto, nelle recensioni nella maggior parte dei casi si tende a rimanere neutri dando valutazioni non particolarmente polarizzate.

3. POS-TAGGING

È opportuno all'interno di un testo vedere il ruolo delle parole oltre alla loro frequenza, per questo una volta calcolato il sentiment ho effettuato un lavoro di etichettatura del testo, attribuendo ad ogni parola il corretto ruolo grammaticale. Ho effettuato questa etichettatura attraverso la libreria 'udpipe', scaricando un modello già esistente che ho inserito in una variabile e facendogli annotare tutto il testo.

	token_id	token	lemma	upos	xpos
1	1	I	I	PRON	PRP
2	2	have	have	AUX	VBP
3	3	n't	not	PART	RB
4	4	given	give	VERB	VBN
5	5	any	any	DET	DT
6	6	stars	star	NOUN	NNS
7	7	for	for	ADP	IN
8	8	Alexa	Alexa	PROPN	NNP
9	9	integration	integration	NOUN	NN
10	10	,	,	PUNCT	,
11	11	for	for	ADP	IN
12	12	the	the	DET	DT
13	13	simple	simple	ADJ	JJ
14	14	reason	reason	NOUN	NN

Questa annotazione ci restituisce 17 diverse variabili, tra queste: token_id che è il numero progressivo della parola, token che è la parola, lemma che è appunto il lemma della parola e upos e xpos che sono i due pos, upos è il pos nella classificazione universale (trasversale a tutte le lingue) mentre xpos è specifico per la lingua.

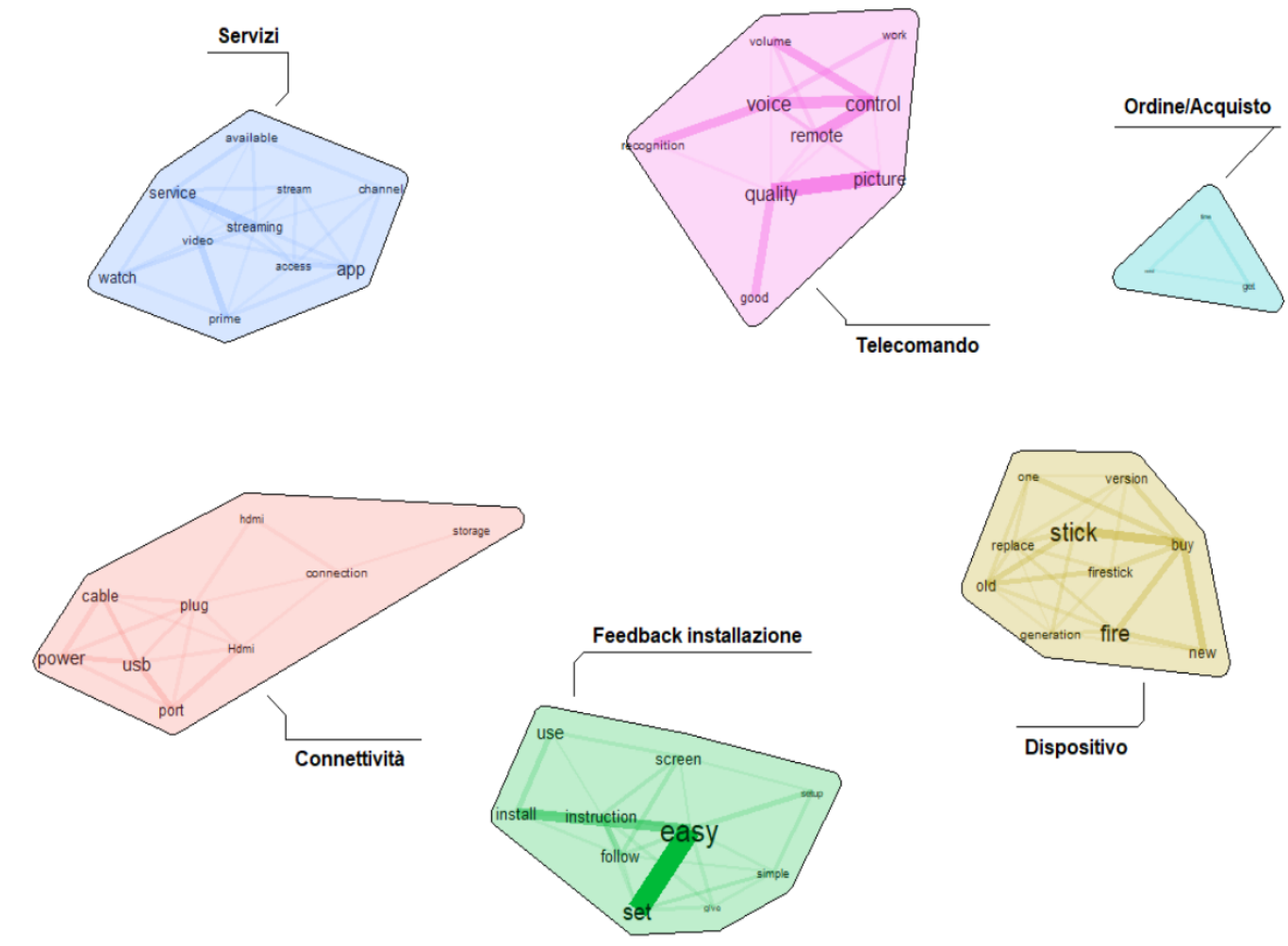
4. TOPIC MODELLING

Dopo aver etichettato tutte le parole presenti all'interno del testo, attraverso il topic modelling ho individuato i vari argomenti (topic) di cui il testo tratta. Il topic modelling è una tecnica di machine learning capace di riconoscere alcune parole che esprimono concetti simili e cerca di raggruppare quest'ultime sotto dei topic, dando la possibilità di poter selezionare tutte le recensioni che trattano di un determinato argomento e di fare un'analisi specifica di un determinato topic. Per l'individuazione dei vari topic all'interno delle recensioni da me scaricate ho utilizzato il modello BTM (Biterm Topic Modelling) che considera occorrenze di bigrammi come caratteristici del topic ed è indicato per documenti non troppo lunghi.

Per trovare i vari topic ho utilizzato la libreria chiamata appunto 'BTM', innanzitutto ho convertito il testo annotato in un data table attraverso la libreria 'data.table', ho calcolato le co-occorrenze (parole che appaiono insieme nello stesso testo) e successivamente ho addestrato il modello, calcolando 6 topic e dando a loro un nome in base alle parole in essi contenute e all'argomento da esse trattato, e infine li ho graficati attraverso le librerie 'textplot', 'concaveman', 'ggraph' .

Ho individuato così 6 differenti topic: Servizi; Connettività; Feedback installazione; Telecomando; Dispositivo; Ordine/acquisto

BTM model

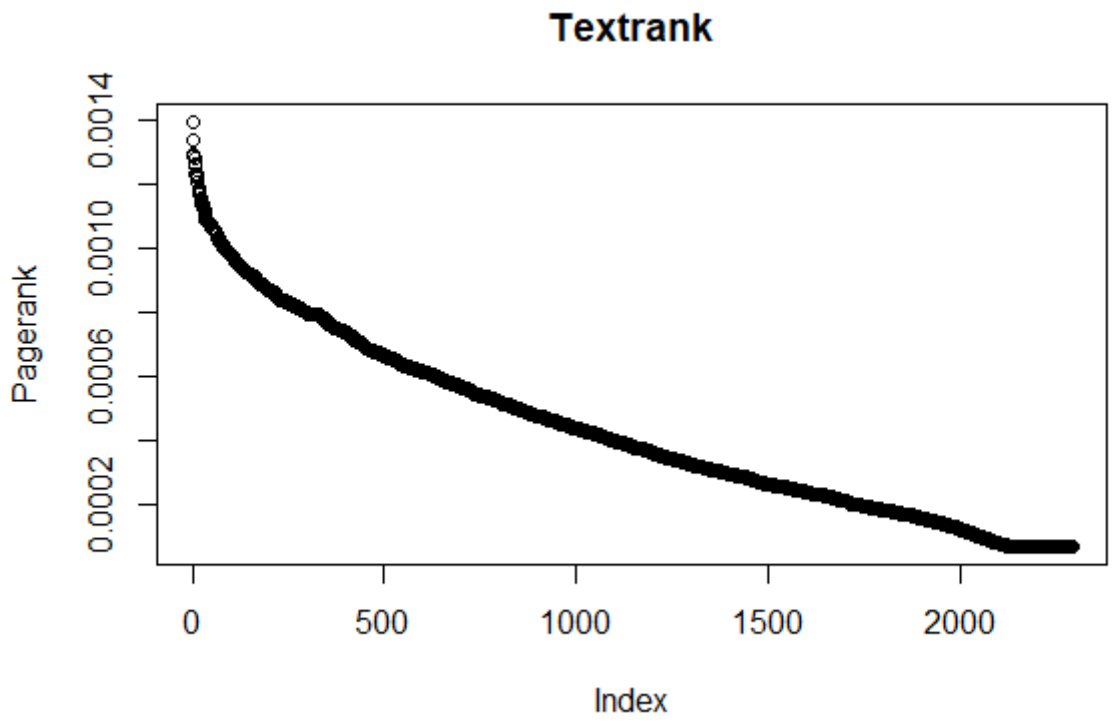


5. SUMMARIZATION

Una volta eseguite tutte le varie operazioni statistiche sulle recensioni, analizzato il sentiment, annotato il testo e trovati i vari topic abbiamo bisogno di riassumere sia le opinioni sia il testo, questo lo facciamo attraverso la summarization.

Per effettuare questa operazione ho utilizzato il metodo TextRank, che è basato sulla costruzione di un grafo dove i nodi del grafo sono le frasi, e il peso degli archi è l'indice di similarità tra due nodi del grafo. Le frasi più importanti sono quelle con maggiore similarità alle altre, ogni frase viene rappresentata attraverso un vettore.

Usando la libreria 'textrank' ho assegnato un punteggio di textrank ad ogni frase, le ho poi ordinate in base al punteggio dal più grande al più piccolo e con 'plot' ho creato un grafico rank size.



Questo tipo di grafico in un insieme di frasi ci serve per capire se la distribuzione dell'intensità è una distribuzione più o meno uniforme, nel caso in cui abbiamo molti dati e la loro intensità cala velocemente possiamo considerare solamente i dati con intensità maggiore.

In questo caso abbiamo un calo ben costante fino ad arrivare ad un valore nullo, quindi potremo decidere che da un certo ramo in poi le restanti frasi saranno superflue, e quindi lavorare esclusivamente su quelle di maggiore intensità e peso.