Stima del volume totale di ricerche su Google

Giovanni Ivan Indiveri Lorenzo Mannocci Jacopo Rescigno

Introduzione

Questo report riguarda l'analisi delle queries totali relative alle ricerche inerenti la *"Premier League"* eseguite negli ultimi 12 mesi nel Regno Unito per poterne stimare il volume totale.
La fase iniziale è consistita nella creazione delle keywords (ottenute sia concatenando termini relativi al calcio, squadre, allenatori e giocatori del campionato inglese sia collezionando i più recenti tweets dell'account ufficiale *"@premierleague"* e raccogliendo i più frequenti bigrammi e trigrammi). Successivamente, abbiamo effettuato la raccolta della loro mole di ricerca su Google principalmente tramite il sito *SearchVolume*. In seguito, il dataset creato è stato utilizzato anche su *Google Trends* per ottenerne i risultati relativi al volume di una singola ricerca di base.
La differenza sostanziale tra le due stime è che il primo fornisce risultati discreti categorizzati in range, mentre il secondo valori continui relativi a una query di riferimento pari ad 1.00.
Per l'analisi abbiamo utilizzato prevalentemente il software di statistica R.

Analisi preliminare dei dati raccolti

Una volta raccolti i dati, abbiamo deciso di usare il dataset di "Google Trends" per procedere all'analisi di quest'ultimi.
Il dataset ottenuto è stato rappresentato in modo tale da poterne valutare la distribuzione e ipotizzare un modello che fosse congruente con l'andamento empirico dei dati.
Per prima cosa abbiamo rappresentato il *Q-Q plot*, il quale ci ha permesso di dedurre che l'andamento non derivasse da una distribuzione normale.

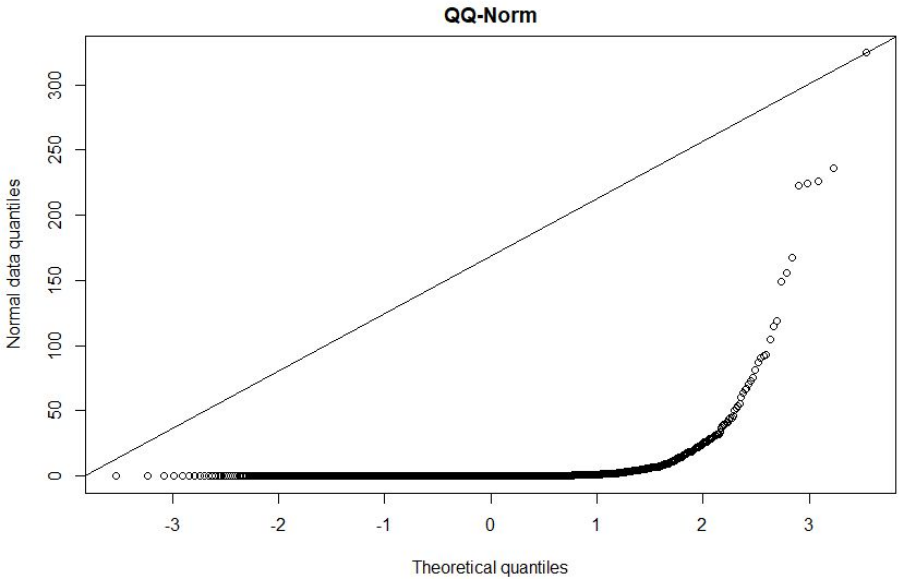

Figura 1 - Q-Q plot per verificare se il dataset è distribuito normalmente

Una seconda conferma ci è pervenuta dall'utilizzo dello *Shapiro-Wilk test*[1], il quale ci ha fornito un valore molto piccolo di p-value pari a 2.2e-16, rigettando l'ipotesi di base.
Un'ultima conferma ci è stata data dal pacchetto **ptsuite**[2] il quale fornisce due interessanti funzioni: *pareto_qq_norm()* e *pareto_test() dove* la prima, simile al qq-plot, verifica graficamente se la distribuzione è o meno una power-law, la seconda ci ha restituito un p-value $0 < 0.05$, il quale ci ha fatto rigettare l'ipotesi H_0 in merito all'omonima distribuzione.

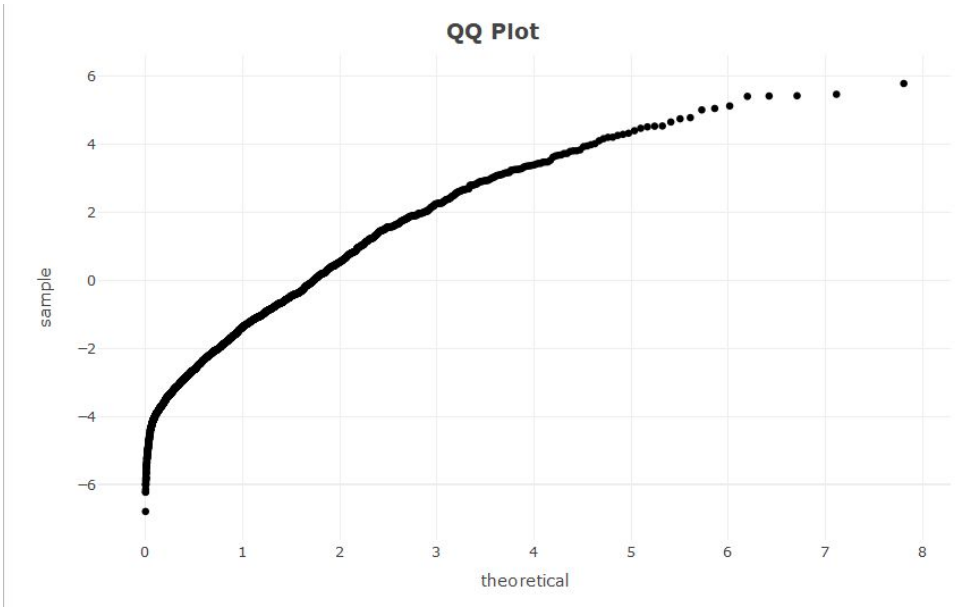

Figura 2 - Q-Q plot per verificare se il dataset è distribuito come una power-law

Queste prime analisi ci hanno condotto a scartare l'ipotesi di normalità in favore di altre possibili distribuzioni. Giacché in letteratura la power-law sembrerebbe essere la più indicata a seguire l'andamento di questo tipo di dati, abbiamo condotto un approfondimento ipotizzando principalmente questa particolare distribuzione statistica ed altre simili.

Power-law, log-normal o exponential?

Utilizzando il package **poweRlaw**[3], abbiamo stimato i seguenti valori, i quali sono ricavati dal package attraverso la funzione `estimate_xmin()`, seguendo il metodo di Newman[4] che ottimizza la log-Maximum Likelihood Estimation (MLE):

$$\hat{\alpha} \simeq 1 + n \left[\sum_{i=1}^{n} log \left(\frac{x_i}{x_{min} - 0.5} \right) \right] \qquad (1)$$

e restituisce:

- $\hat{\alpha}$ per la power-law distribution.

[1] Sam S. Shapiro, Martin Bradbury Wilk (1965). "An analysis of variance test for normality (complete samples)", *Biometrika*, 52, 3 e 4, pagine 591-611.
[2] Package: 'ptsuite': https://cran.r-project.org/web/packages/ptsuite/ptsuite.pdf.
[3] The poweRlaw package: Example, Colin S.Gillespie, 2019.
[4] Power laws, Pareto distributions and Zipf's law, M. E. J. Newman, 2005

Mentre, per un confronto con altre possibili distribuzioni, abbiamo stimato anche i seguenti parametri:

- *log-*$\widehat{\mu}$, *log-*$\widehat{\sigma}^2$ per la distribuzione log-normal;
- $\widehat{\lambda}$, per la distribuzione exponential.

oltre ai valori di $\widehat{x}_{min}$ per le varie distribuzioni stimate.

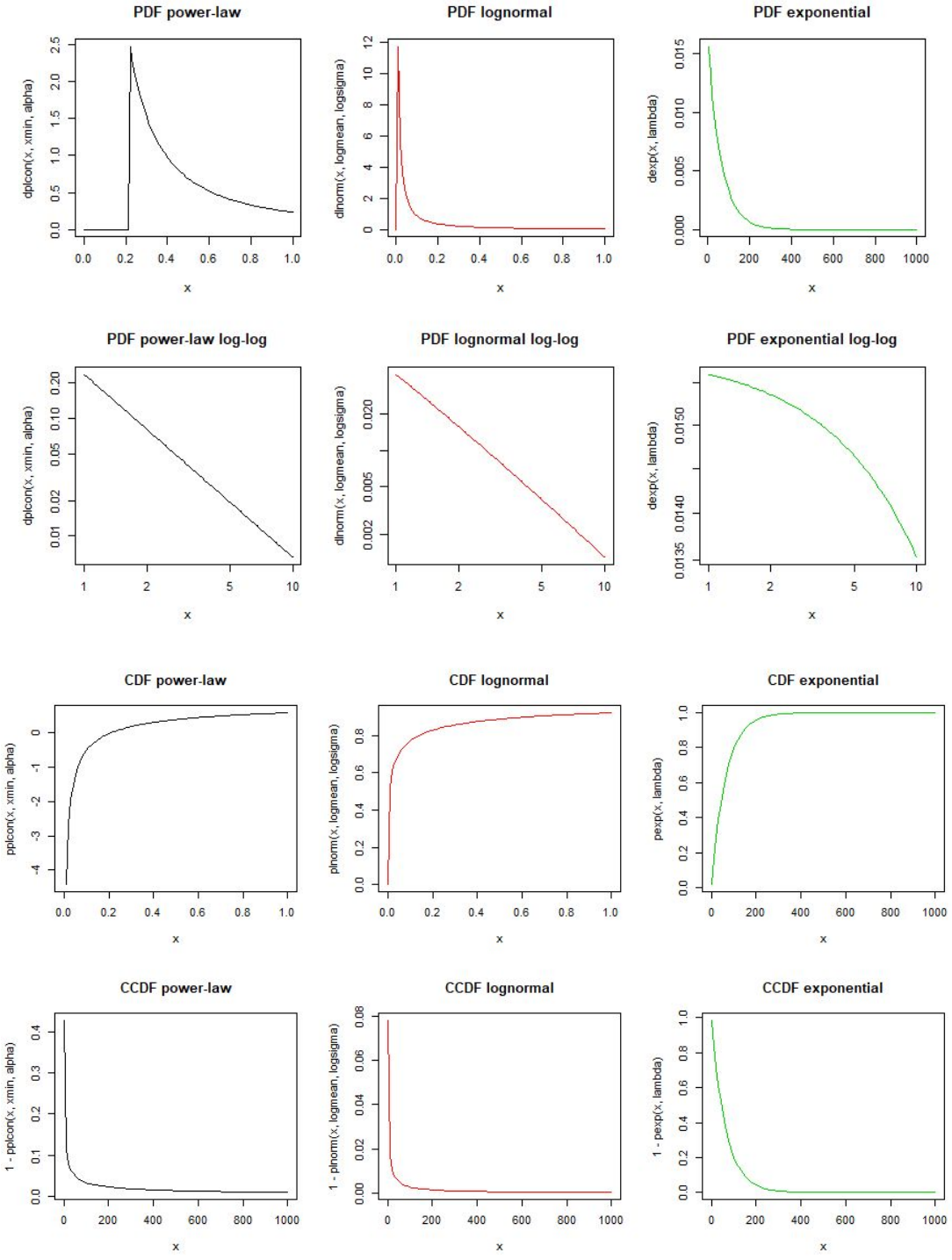

Figura 3 - PDF, PDF log-log e CCDF per power-law, log-normal ed exponential

Nel fare questo abbiamo ricavato la *CCDF (complementary cumulative distribution function)* di queste tre distribuzioni per poter confrontare quale tra queste approssimasse meglio il dataset iniziale e fin da subito abbiamo ipotizzato che una distribuzione log-normal fosse quella più appropriata giacché con la funzione `get_distance_statistic()` calcolata utilizzando la *Kolmogorov-Smirnov distance*[5] sia sulla power-law che sulla log-normal ci ha fornito un valore inferiore per la seconda.

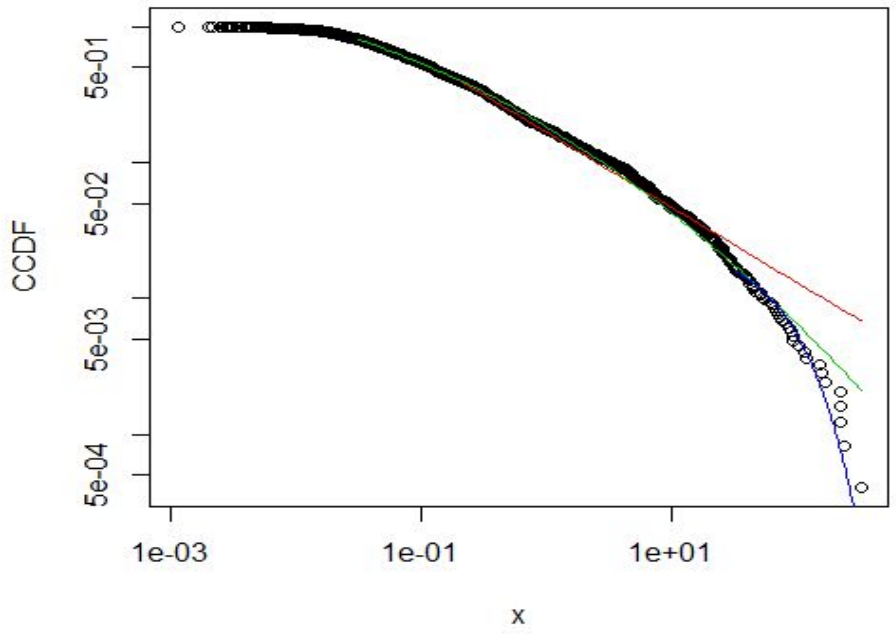

Figura 4 - CCDF di confronto per power-law, log-normal ed exponential

Come si evince dal grafico, la distribuzione log-normal (in verde) sembrerebbe quella che si adatta meglio ai dati rispetto alla power-law (in rosso) e alla exponential (in blu). Anche il valore di x_{min} per la prima possibilità risulta essere il più piccolo.

I parametri stimati per le tre distribuzioni sono elencati nella seguente tabella:

Distribuzione stimata	**Parametri stimati**		**x_{min} stimati**
continuous power-law	$\hat{\alpha} = 1.55$		$x_{min} = 0.2139041$
continuous log-normal	$log-\hat{\mu} = -4.828037$	$log-\hat{\sigma}^2 = 3.402472$	$x_{min} = 0.03175482$
continuous exponential	$\hat{\lambda} = 0.01585425$		$x_{min} = 32.36799$

Tabella 1 - Stime dei parametri con il package poweRlaw

[5] Power-law distribution in empirical data, Aaron Clauset et al., 2009.

I valori restituiti dalla funzione `get_distance_statistic(distance = "ks")` sono rispettivamente:

Kolmogorov-Smirnov GOF distance	
continuous power-law	0.036752
continuous log-normal	0.020094
continuous exponential	0.086435

Tabella 2 - Godness-of-fit con il package poweRlaw

Come possiamo vedere dalla Tabella 2, anche analiticamente la log-normal risulta essere la distribuzione con KS distance più piccolo, e quella dunque che approssima meglio il dataset.

Il confronto è stato fatto con il package **fitdistrplus**[6] che pone a confronto il dataset con le possibili distribuzioni più pertinenti. Nel nostro caso abbiamo ipotizzato cinque possibili distribuzioni: normal, log-normal, gamma, pareto ed exponential.

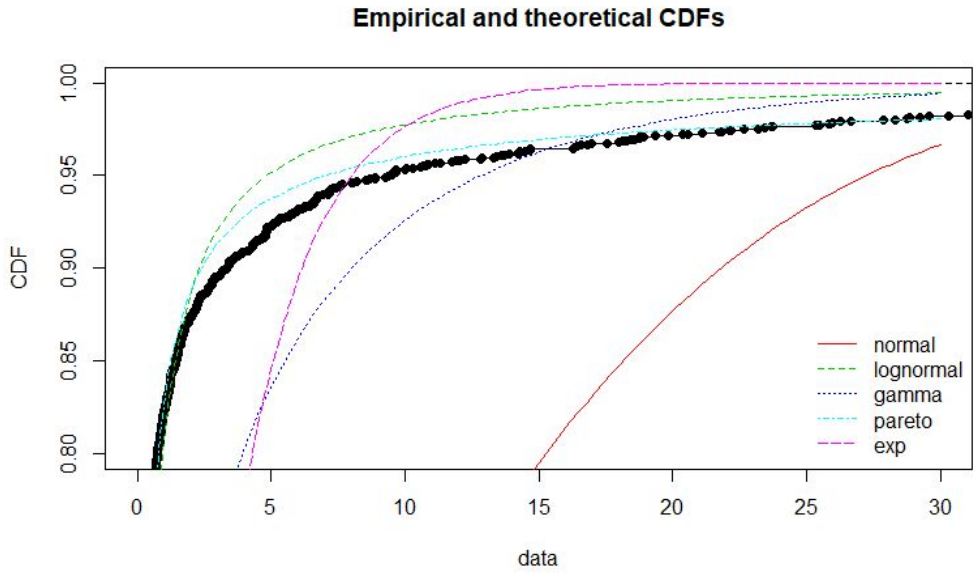

Figura 5 - Le varie distribuzioni stimate con il package fitdistrplus

La CDF del dataset (in nero) visivamente sembrerebbe essere più vicino alla pareto che alla log-normal distribution ed anche in base alla KS-distance (vedi tabella sottostante), la power-law sembrerebbe approssimare meglio rispetto alla log-normal, in controtendenza al package poweRlaw.

[6] Marie Laure Delignette-Muller, Christophe Dutang (2015). fitdistrplus: An R Package for Fitting Distributions. Journal of Statistical Software, 64(4), 1-34. URL http://www.jstatsoft.org/v64/i04/.

Kolmogorov-Smirnov GOF distance	
continuous power-law	0.0498184
continuous log-normal	0.07721355
continuous exponential	0.5791235
continuous gamma	0.251465
continuous normal	0.4291443

Tabella 3 - Godness-of-fit con il package fitdistrplus

Quindi secondo il package fitdistrplus, la log-normal approssima meglio il dataset.

Visualizziamo infine il KS-distance per le 2 distribuzioni più prossime alla power-law utilizzando la funzione ks.dist() implementata, e poi avvalendoci dello script basato sul package ggplot[7].

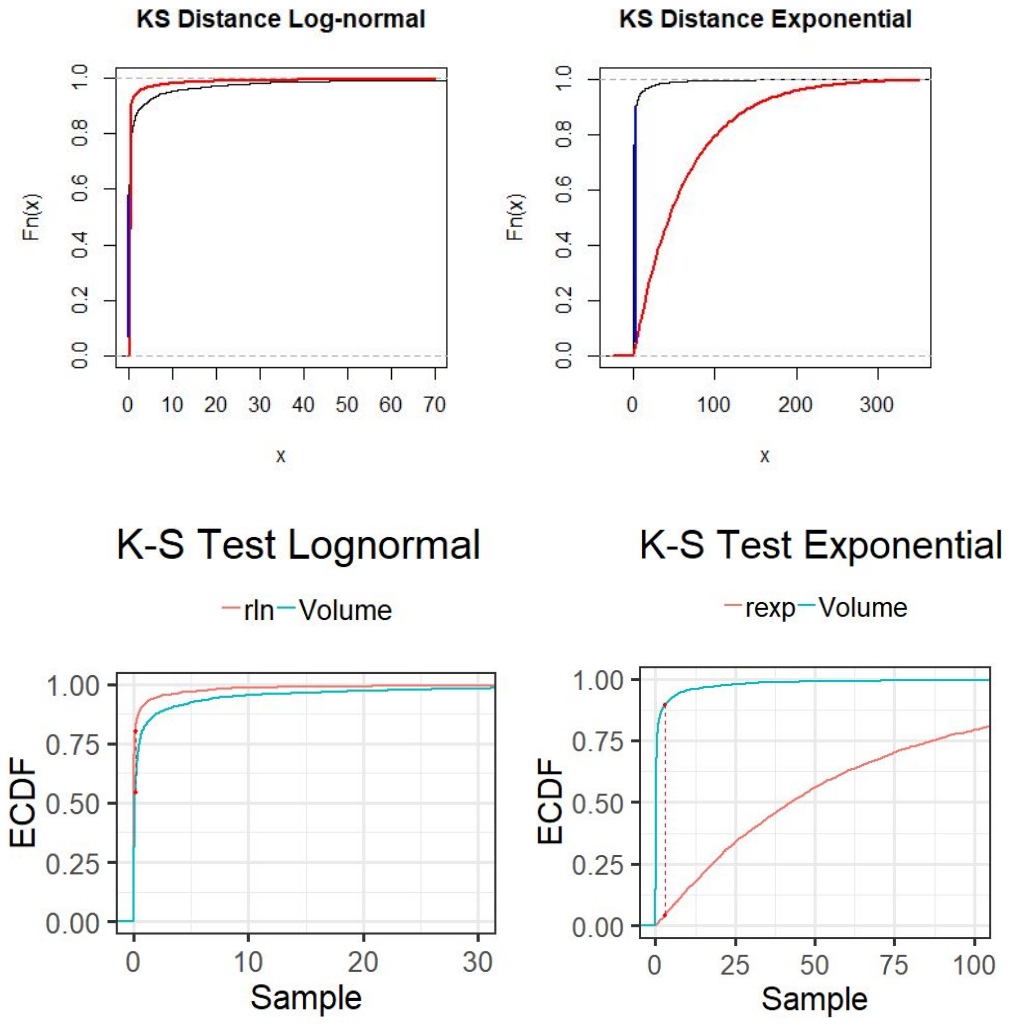

Figura 6 - Visualizzazione della distanza di Kolmogorov-Smirnov

[7] Abbiamo utilizzato la libreria **tidyverse** come confronto al codice fornito. In allegato al progetto: “final_script.R”

Quindi secondo il package poweRlaw la log-normal approssimerebbe meglio il dataset, mentre secondo il package fitdistrplus è la pareto ad approssimare meglio; è possibile dunque che le procedure dei due pacchetti siano leggermente diverse e/o che gli intervalli di confidenza delle stime si sovrappongono.
Dato che il package poweRlaw si rifà esplicitamente ai metodi della letteratura, abbiamo optato di seguire i risultati del suddetto package e quindi ne risulta che la distribuzione che corrisponde meglio al dataset è effettivamente una log-normal che ricordiamo avere la seguente PDF:

$$f(x) = \frac{1}{x\sigma\sqrt{2\pi}} exp\left(\frac{-(ln\, x - \mu)^2}{2\sigma^2}\right) \quad .$$

Troncamento ricorsivo della distribuzione

Tenendo in considerazione la metodologia proposta da Clauset et al. (2009) e il fatto che la power-law è generalmente visibile lungo la distribuzione per $x \geq x_{min}$, "*gli autori suggeriscono di testare l'uguaglianza tra le funzione di densità teorica ed empirica usando i test di Kolmogorov-Smirnov 'ricorsivamente' sulla distribuzione troncata. La nostra stima di* x_{min} *è quindi il valore di* x *che minimizza la Kolmogorov-Smirnov,* D*:*

$$D = \sup_{x \geq x_{min}} |\Phi_x - \Phi(x)| \qquad (1)$$

dove $\Phi_x(x)$ *è la funzione di densità cumulativa empirica per le osservazioni di* x *i.i.d e* $\Phi(x)$ *è la funzione di densità cumulativa teorica"* [8]*.*

Grazie a tale metodo abbiamo utilizzato un algoritmo per rimuovere in maniera ricorsiva i valori precedenti all' x_{min} dal dataset originale fino al punto in cui la KS-distance non sia risultata minore per la distribuzione power-law. Così facendo il numero di istanze è diminuito da n = 2453 ad n = 78. Conseguentemente a questo taglio, abbiamo ricavato una distribuzione power-law troncata che siamo andati a stimare attraverso diversi test di approssimazione:

- Da prima utilizzando nuovamente il package poweRlaw e confrontando le due principali distribuzioni stimate attraverso la funzione `bootstrap_p()`. In questa maniera abbiamo ricavato un p-value di 0.833 a favore della continuous power-law e 0.612 per la continuous log-normal. Da questi risultati è evidente come in entrambi casi entrambe le distribuzioni risultino plausibili perché le ipotesi H_0 non possono essere scartate, ma

[8] Pareto or log-normal? A recursive-truncation approach to the distribution of (all) cities, Giorgio Fazio, Marco Modica, 2012, pag.11.

l'esiguo scarto ci ha orientato ad approssimare il nostro dataset ad una power-law teorica con i valori stimati di $x_{min} = 18.15$ e un valore di $\hat{\alpha} = 2.17$;

- Con la funzione `get_distance_statistic()` ottenendo per la log-normal una distanza pari a 0.057 e per la power-law di 0.054;
- Infine utilizzando la funzione `pareto_test()` del package ptsuite, che ci ha restituito un p-value pari a 0.471.

Ottenuto questo nuovo dataset con questi parametri stimati, il passo successivo è stato quello di ordinare i valori del volume ed estrarre il valore di $\hat{\beta} = 1/(\hat{\alpha} - 1)$ della Zipf-law attraverso la relazione:

$$\beta = \frac{1}{(\alpha - 1)} \quad . \qquad (2)$$

In questa maniera abbiamo ottenuto il seguente grafico in scala logaritmica, con la relativa Zipf stimata dal nostro valore di $\hat{\alpha}$ dove il rank nell'asse delle x risulta essere ordinato con $V_{r_1} = 324.41$ (il volume della ricerca corrispondente a "football") e conseguentemente $V_{r_2} < V_{r_3} < \ldots < V_{r_n}$:

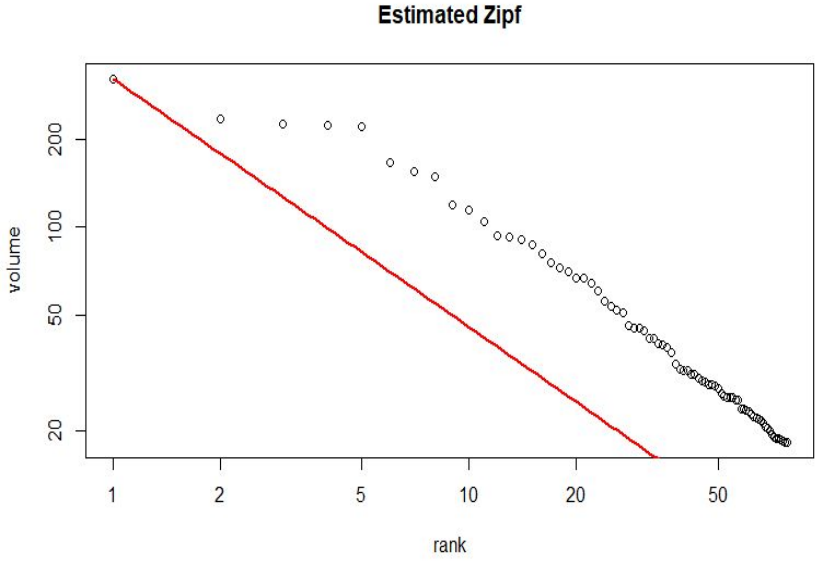

Figura 7 - Distribuzione stimata di Zipf per il nostro dataset

Notiamo subito come lo scostamento tra la retta ed i valori del dataset analizzato sia minore per le ricerche con volume maggiore e tenda ad aumentare per quelle con rank inferiore. Chiaramente, essendo il numero n = 78 non eccessivamente alto, i risultati su un dataset con più valori probabilmente ne gioverebbe a livello di approssimazione generale.

Stima del numero e del volume di ricerche

Stima del numero delle ricerche

Ricavare la Zipf della nostra distribuzione è servito principalmente per determinare la stima del volume totale e del numero delle nostre ricerche inerenti la Premier League. Innanzitutto abbiamo ricavato la costante moltiplicativa g del volume esaminando il valore della query di riferimento con valore 1.00 ("season games") su Google Trends con il volume mensile restituito da un SEO tool (SEMrush). Abbiamo così verificato che g è uguale a 480 nel nostro caso.
Di conseguenza la query relativa a "season games" avrà un volume mensile assoluto $V = V_{GT}\, g$, dove V_{GT} è il volume stimato da Google Trends.

Partendo quindi dal fatto che la Zipf-law, nel nostro caso, risulta essere:

$$f(i) = \frac{c}{r(i)^{\beta}} \quad , \qquad (1)$$

dove $f(i)$ rappresenta il volume della query i, c la nostra costante pari al volume della prima ricerca (rank max), $r(i)$ il rank della $i-$esima ricerca e β l'esponente ricavato dalla power-law. Come già detto, seguendo il metodo di Clauset et al. e Newman, abbiamo stimato $\widehat{c} = 324.41$ e $\widehat{\beta} = 0.854$. Dall'uguaglianza di sopra si ha ovviamente che $f(1) = V_1 = c$, ossia il volume della prima ricerca è pari al volume della prima query.
Dunque riprendendo il metodo di stima di "*Estimating the total volume of queries to Google"* [9], possiamo scrivere l'equazione precedente come:

$$V_i = \frac{c}{i^{\beta}} \qquad (2)$$

per cui, attraverso semplici passaggi algebrici, possiamo calcolarci il volume V_n, cioè quello della query con volume più piccolo:

[9] Estimating the total volume of queries to Google, F.Lillo, S.Ruggieri, 2019

$$V_n = \frac{c}{N^{\beta}} \quad , \tag{3}$$

dal quale risulta

$$N^{\beta} = \frac{c}{V_n} \quad ,$$

$$N = \left(\frac{c}{V_n}\right)^{\frac{1}{\beta}} \tag{4}$$

che può essere utilizzato come stimatore $\widehat{N}$ del numero di ricerche per il nostro dataset:

$$\widehat{N} = \left(\frac{c}{V_n}\right)^{\frac{1}{\beta}} \quad ,$$

da cui per calcolarci il numero di query con un volume superiore a *v* ricerche:

$$\widehat{N}_v = \left(\frac{c}{v}\right)^{\frac{1}{\beta}} \quad . \tag{5}$$

Stima volume delle ricerche

Dunque abbiamo che il volume totale è dato da:

$$V = \sum_{n=1}^{N} \frac{c}{i^{\beta}} \tag{4}$$

con N = 78, è possibile esplicitare la (4) come:

$$V = c\left(\sum_{n=1}^{\infty} \frac{c}{i^{\beta}} - \sum_{n=N+1}^{\infty} \frac{c}{i^{\beta}}\right) \quad . \tag{5}$$

La (5) può essere quindi vista come la differenza fra la funzione zeta di Riemann e la funzione zeta di Hurwitz.
Ricordiamo la definizione della funzione zeta di Riemann:

$$\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s}$$

e la più generica funzione zeta di Hurwitz:

$$\zeta(s,q) = \sum_{n=1}^{\infty} \frac{1}{(q+n)^s} .$$

Per cui si ha che (5) può essere riscritta:

$$V = c\,(\,\zeta(\beta) - \zeta(\beta,\, N+1)\,)\ , \qquad (6)$$

che può essere utilizzato come stimatore $\widehat{V}$ del volume, utilizzando $\widehat{\beta}$ e $\widehat{c}$:

$$\widehat{V} = \widehat{c}\,(\,\zeta(\widehat{\beta}) - \zeta(\widehat{\beta}, \widehat{N}+1)\,) \qquad (7)$$

$$\widehat{V} = \widehat{c}\,(\,\zeta(\widehat{\beta}) - \zeta(\widehat{\beta}, (\tfrac{\widehat{c}}{V_n})^{\frac{1}{\widehat{\beta}}}+1)\,) \qquad (8)$$

che possiamo generalizzare nello stimatore del volume totale delle query con almeno *v* ricerche:

$$\widehat{V}_v = \widehat{c}\,(\,\zeta(\widehat{\beta}) - \zeta(\widehat{\beta}, (\tfrac{\widehat{c}}{v})^{\frac{1}{\widehat{\beta}}}+1)\,)\ . \qquad (9)$$

Utilizzando questa metodologia nello script di R con l'ausilio della libreria **reticulate**[10] siamo stati in grado di calcolare il volume interagendo con uno script in Python ed ottenendo quindi i seguenti risultati date *v* ricerche:

Valori stimati di $\widehat{N_v}$ e $\widehat{V_v}$ per query con almeno *v* ricerche			
v	***v*/12**	$\widehat{N_v}$	$\widehat{V_v}$
12	1	1,194,090	86,650,188
120	10	80,659	54,708,984
1,200	100	5,444	33,133,733
12,000	1,000	368	18,565,040
120,000	10,000	25	8,774,143
600,000	50,000	4	4,088,594

Tabella 4 - Valori di $\widehat{N_v}$ e $\widehat{V_v}$ relativi al dataset della Premier League

Test sperimentale degli stimatori

Per valutare la robustezza del modello assunto è stato deciso di fare una stima dell'errore statistico σ relativo ad $\widehat{\alpha}$. Tale valore, è stato ricavato attraverso la seguente formula presente sia nel paper di Clauset et al. sia nel Newman[11]:

$$\sigma = \frac{\widehat{\alpha} - 1}{\sqrt{n}} \qquad (9)$$

Il valore ottenuto dalla (9) è stato

$$\sigma = 0,13$$

perciò il valore di $\widehat{\alpha}$ è

$$\widehat{\alpha} = 2,17 \pm 0,13$$

[10] https://cran.r-project.org/web/packages/reticulate/reticulate.pdf , questo package permette di eseguire uno script Python da R. Questo è stato necessario per poter calcolare la funzione zeta di Hurwitz, che non è implementata correttamente in R. Mentre è implementata la funzione zeta di Riemann: abbiamo a quel punto deciso per comodità di scrivere in python la funzione che stima il volume delle ricerche. Il package reticulate ci permette di visualizzare i risultati direttamente in R.

[11] Power laws, Pareto distributions and Zipf's law, M. E. J. Newman, 2005, pag. 5; Power-law distribution in empirical data, Aaron Clauset et al., 2009, pag. 5

Determinato questo intervallo, abbiamo condotto dei test al variare di $\widehat{\alpha}$ (e quindi anche di $\widehat{\beta}$) e calcolato i valori relativi al numero di query e volume sia sul dataset (troncato) riguardante la Premier League, sia sui dati generati attraverso una random power-law avente lo stesso numero di elementi.

Definiamo le seguenti simbologie:

- ➢ f^{+}_{rpl} è la realizzazione della random power-law con $\alpha = \widehat{\alpha} + \sigma$
- ➢ f^{-}_{rpl} è la realizzazione della random power-law con $\alpha = \widehat{\alpha} - \sigma$
- ➢ f_{rpl} è la realizzazione della random power-law con $\alpha = \widehat{\alpha}$

Nello specifico, i test condotti sono stati caratterizzati dalle seguenti condizioni :

- calcolo del numero di query per $\widehat{\alpha} = 2,17 \pm 0,13$ sul dataset della Premier League;

Stima numero di ricerche su dataset, con errore su $\widehat{\alpha}$				
v	***v/12***	**N_v ($\widehat{\alpha}$ + σ)**	**N_v ($\widehat{\alpha}$)**	**N_v ($\widehat{\alpha}$ - σ)**
12	1	5,822,720	1,194,090	244,877
120	10	289,882	80,659	22,443
1,200	100	14,124	5,444	2,022
12,000	1,000	719	368	189
120,000	10,000	36	25	17
600,000	50,000	4	4	3

Tabella 5 - Valori di N_v al variare di $\widehat{\alpha}$ relativi al dataset della Premier League

- calcolo del numero di ricerche per il dataset generato dalla random power-law con $\widehat{\alpha} = 2,17 \pm 0,13$ ed avente un valore di $\widehat{c}_{rpl}^{\pm} = f_{rpl}^{\pm}[1]$, ovvero corrispondente al volume della query con rank = 1 della relativa Zipf's Law;

Stima numero di ricerche su random power law, con errore su $\widehat{\alpha}$ e $\widehat{c}$				
v	***v/12***	**N_v ($\widehat{\alpha}$ + σ) , $\widehat{c}_{rpl}^{+}$**	**N_v ($\widehat{\alpha}$), $\widehat{c}$ (dataset)**	**N_v ($\widehat{\alpha}$ - σ), $\widehat{c}_{rpl}^{-}$**
12	1	12,245,394	1,194,090	328,509
120	10	609,632	80,659	30,108
1,200	100	29,704	5,444	2,712
12,000	1,000	1,511	368	252
120,000	10,000	75	25	23
600,000	50,000	9	4	4

Tabella 6 - Confronto tra i valori di N_v al variare di $\widehat{\alpha}$ ottenuti dal dataset della Premier League e da una rpl

- calcolo del numero di ricerche per il dataset generato dalla random power-law con $\widehat{\alpha} = 2,17$ ed avente un valore di $\widehat{c}_{rpl} = f_{rpl}[1]$, ovvero corrispondente alla query con rank massimo della relativa Zipf's Law;

Stima numero di ricerche su random power law, con errore su $\widehat{c}$			
v	***v/12***	**N_v ($\widehat{\alpha}$) , $\widehat{c}_{rpl}$**	**N_v ($\widehat{\alpha}$), $\widehat{c}$ (dataset)**
12	1	1,266,844	1,194,090
120	10	85,573	80,659
1,200	100	5,670	5,444
12,000	1,000	390	368
120,000	10,000	26	25
600,000	50,000	4	4

Tabella 7 - Confronto tra i valori di N_v ottenuti dal dataset della Premier League e da una rpl

- stima del volume per $\widehat{\alpha} = 2,17 \pm 0,13$ sul dataset della Premier League;

Stima volume su dataset, con errore su $\widehat{\alpha}$				
v	***v/12***	**V_v ($\widehat{\alpha}$ + σ)**	**V_v ($\widehat{\alpha}$)**	**V_v ($\widehat{\alpha}$ - σ)**
12	1	293,556,450	86,650,188	30,399,207
120	10	142,635,240	54,708,984	23,674,287
1,200	100	67,128,265	33,133,733	17,469,032
12,000	1,000	30,099,804	18,565,040	11,867,895
120,000	10,000	11,531,236	8,774,143	6,744,139
600,000	50,000	4,639,070	4,088,594	3,610,178

Tabella 8 - Valori di V_v al variare di $\widehat{\alpha}$ relativi al dataset della Premier League

- stima del volume delle ricerche per il dataset generato dalla random power-law con $\widehat{\alpha} = 2,17 \pm 0,13$ ed avente un valore di $\widehat{c}^{\pm}_{rpl} = f^{\pm}_{rpl}[1]$, ovvero corrispondente alla query con rank massimo della relativa Zipf's Law;

Stima volume di ricerche su random power law, con errore su $\widehat{\alpha}$ e $\widehat{C}$				
v	**v/12**	**V_v ($\widehat{\alpha}$ + σ) , $\widehat{c}^{+}_{rpl}$**	**V_v ($\widehat{\alpha}$), $\widehat{C}$ (dataset)**	**V_v ($\widehat{\alpha}$ - σ), $\widehat{c}^{+}_{rpl}$**
12	1	619,693,427	86,650,188	41,498,609
120	10	302,300,562	54,708,984	32,476,923
1,200	100	143,506,036	33,133,733	24,152,215
12,000	1,000	65,627,888	18,565,040	16,636,295
120,000	10,000	26,518,296	8,774,143	9,744,522
600,000	50,000	11,772,887	4,088,594	5,466,641

Tabella 9 - Confronto tra i valori di V_v al variare di $\widehat{\alpha}$ ottenuti dal dataset della Premier League e da una rpl

- stima del volume per il dataset generato dalla random power-law con $\widehat{\alpha} = 2,17$ ed avente un valore di $\widehat{c}_{rpl} = f_{rpl}[1]$, ovvero corrispondente alla query con rank massimo della relativa Zipf's Law.

Stima numero di ricerche su random power law, con errore su $\widehat{c}$			
v	***v/12***	$\mathbf{V_v}(\widehat{\alpha})$, $\widehat{c}_{rpl}$	$\mathbf{V_v}(\widehat{\alpha})$, $\widehat{c}$ **(dataset)**
12	1	92,036,787	86,650,188
120	10	58,149,447	54,708,984
1,200	100	35,125,621	33,133,733
12,000	1,000	19,802,940	18,565,040
120,000	10,000	9,412,245	8,774,143
600,000	50,000	4,427,871	4,088,594

Tabella 10 - Confronto tra i valori di V_v ottenuti dal dataset della Premier League e da una rpl

Conclusioni

Dall'osservazione dei risultati riportati nelle tabelle di cui sopra, si evince che il modello da noi prodotto ottiene dei buoni risultati se comparato con distribuzioni generate in maniera casuale, ma con parametri simili.
Nello specifico i risultati ottenuti nei test sperimentali degli errori nel caso della stima del numero delle ricerche e della stima del volume totale seguono un trend simile.
Nel caso della sola variazione di $\widehat{\alpha}$ si ha uno variazione piuttosto ampia dai valori stimati.
Al variare di $\widehat{\alpha}$ e di $\widehat{c}$, si ha una variazione ancora maggiore, ma non eccessiva; mentre possiamo notare come nel caso del solo errore nella $\widehat{c}$, i valori stimati non variano particolarmente.
Dunque la stima di $\widehat{c}$ è valida e con un'esigua variazione, mentre la maggior parte dell'errore lo si ritrova nella stima di $\widehat{\alpha}$. Questo è chiaramente dovuto al fatto che il valore di α è il parametro fondamentale nelle distribuzioni power-law (pareto) ed influisce sui risultati in maniera decisiva.

Un ulteriore aspetto osservato, è il diminuire della variazione all'aumentare di *v,* numero di ricerche minimo delle queries prese in considerazione.

Questo ci porta a concludere che il modello è buono nel caso di ricerche con volume elevato, ovvero le query di maggior interesse.

Bibliografia e sitografia

- An analysis of variance test for normality, Sam S. Shapiro, Martin Bradbury Wilk, 1965;
- The poweRlaw package: Example, Colin S.Gillespie, 2019;
- Power laws, Pareto distributions and Zipf's law, M. E. J. Newman, 2005;
- Power-law distribution in empirical data, Aaron Clauset et al., 2009;
- Marie Laure Delignette-Muller, Christophe Dutang. fitdistrplus: An R Package for Fitting Distributions. Journal of Statistical Software, 2015;
- Pareto or log-normal? A recursive-truncation approach to the distribution of (all) cities, Giorgio Fazio, Marco Modica, 2012;
- Estimating the total volume of queries to Google, F.Lillo, S.Ruggieri, 2019;
- http://www.jstatsoft.org/v64/i04/;
- https://cran.r-project.org/web/packages/reticulate/reticulate.pdf ;