

# Coffee break!



# What Every Corpus Linguist Should Know About Type-Token Distributions and Zipf's Law

Tutorial Workshop #9, 22 July 2019

Stefan Evert  
FAU Erlangen-Nürnberg

<http://zipfr.r-forge.r-project.org/lrec2018.html>  
Licensed under CC-by-sa version 3.0



**CL 2019**  
International Corpus Linguistics Conference  
Cardiff, Wales, UK  
July 22-26, 2019



# Outline

## Introduction

### Motivation

Notation & basic concepts

Zipf's law

First steps (zipfR)

## LNRE models

Population & samples

The mathematics of LNRE

## Applications & examples

Productivity &

lexical diversity

Practical LNRE modelling

Bootstrapping experiments

LNRE as Bayesian prior

## Challenges

Model inference

Zipf's law

Non-randomness

Significance testing

Outlook

# Some research questions

- ▶ How many words did Shakespeare know?
- ▶ What is the coverage of my treebank grammar on big data?
- ▶ How many typos are there on the Internet?
- ▶ Is *-ness* more productive than *-ity* in English?
- ▶ Are there differences in the productivity of nominal compounds between academic writing and novels?
- ▶ Does Dickens use a more complex vocabulary than Rowling?
- ▶ Can a decline in lexical complexity predict Alzheimer's disease?
- ▶ How frequent is a hapax legomenon from the Brown corpus?
- ▶ What is appropriate smoothing for my n-gram model?
- ▶ Who wrote the Bixby letter, Lincoln or Hay?
- ▶ How many different species of ... are there? (Brainerd 1982)

# Some research questions

- ▶ ↘↘
- ▶ coverage estimates
- ▶ ↗↗
- ▶ ↘↘
- ▶ productivity
- ▶ lexical complexity & stylometry
- ▶ ↗↗
- ▶ prior & posterior distribution
- ▶ ↗↗
- ▶ ↘↘
- ▶ unexpected applications

# Type-token statistics

- ▶ These applications relate **token** and **type** counts
  - ▶ **tokens** = individual instances (occurrences)
  - ▶ **types** = distinct items
- ▶ Type-token statistics different from most statistical inference
  - ▶ not about probability of a specific event
  - ▶ but about diversity of events and their probability distribution

# Type-token statistics

- ▶ These applications relate **token** and **type** counts
  - ▶ **tokens** = individual instances (occurrences)
  - ▶ **types** = distinct items
- ▶ Type-token statistics different from most statistical inference
  - ▶ not about probability of a specific event
  - ▶ but about diversity of events and their probability distribution
- ▶ Relatively little work in statistical science
- ▶ Nor a major research topic in computational linguistics
  - ▶ very specialized, usually plays ancillary role in NLP
- ▶ Corpus linguistics: TTR & simple productivity measures
  - ▶ often applied without any statistical inference

# Zipf's law (Zipf 1949)

A) Frequency distributions in natural language are highly skewed



# Zipf's law (Zipf 1949)

- A) Frequency distributions in natural language are highly skewed
- B) Curious relationship between rank & frequency

word	$r$	$f$	$r \cdot f$
<i>the</i>	1.	142,776	142,776
<i>and</i>	2.	100,637	201,274
<i>be</i>	3.	94,181	282,543
<i>of</i>	4.	74,054	296,216

(Dickens)

# Zipf's law (Zipf 1949)

- A) Frequency distributions in natural language are highly skewed
- B) Curious relationship between rank & frequency

word	$r$	$f$	$r \cdot f$
<i>the</i>	1.	142,776	142,776
<i>and</i>	2.	100,637	201,274
<i>be</i>	3.	94,181	282,543
<i>of</i>	4.	74,054	296,216

(Dickens)

- C) Various explanations of Zipf's law
  - ▶ principle of least effort (Zipf 1949)
  - ▶ optimal coding system, MDL (Mandelbrot 1953, 1962)
  - ▶ random sequences (Miller 1957; Li 1992; Cao *et al.* 2017)
  - ▶ Markov processes → n-gram models (Rouault 1978)

# Zipf's law (Zipf 1949)

- A) Frequency distributions in natural language are highly skewed
- B) Curious relationship between rank & frequency

word	$r$	$f$	$r \cdot f$	
<i>the</i>	1.	142,776	142,776	
<i>and</i>	2.	100,637	201,274	(Dickens)
<i>be</i>	3.	94,181	282,543	
<i>of</i>	4.	74,054	296,216	

- C) Various explanations of Zipf's law
    - ▶ principle of least effort (Zipf 1949)
    - ▶ optimal coding system, MDL (Mandelbrot 1953, 1962)
    - ▶ random sequences (Miller 1957; Li 1992; Cao *et al.* 2017)
    - ▶ Markov processes → n-gram models (Rouault 1978)
  - D) Language evolution: birth-death-process (Simon 1955)
- 👉 not the main topic today!

# Outline

## Introduction

Motivation

Notation & basic concepts

Zipf's law

First steps (zipfR)

## LNRE models

Population & samples

The mathematics of LNRE

## Applications & examples

Productivity &

lexical diversity

Practical LNRE modelling

Bootstrapping experiments

LNRE as Bayesian prior

## Challenges

Model inference

Zipf's law

Non-randomness

Significance testing

Outlook

# Tokens & types

our sample: *recently, very, not, otherwise, much, very, very, merely, not, now, very, much, merely, not, very*

- ▶  $N = 15$ : number of **tokens** = sample size
- ▶  $V = 7$ : number of distinct **types** = **vocabulary size**  
(*recently, very, not, otherwise, much, merely, now*)

# Tokens & types

our sample: *recently, very, not, otherwise, much, very, very, merely, not, now, very, much, merely, not, very*

- ▶  $N = 15$ : number of **tokens** = sample size
- ▶  $V = 7$ : number of distinct **types** = **vocabulary size**  
(*recently, very, not, otherwise, much, merely, now*)

## type-frequency list

$w$	$f_w$
<i>recently</i>	1
<i>very</i>	5
<i>not</i>	3
<i>otherwise</i>	1
<i>much</i>	2
<i>merely</i>	2
<i>now</i>	1

# Zipf ranking

our sample: *recently, very, not, otherwise, much, very, very, merely, not, now, very, much, merely, not, very*

- ▶  $N = 15$ : number of **tokens** = sample size
- ▶  $V = 7$ : number of distinct **types** = **vocabulary size**  
(*recently, very, not, otherwise, much, merely, now*)

## Zipf ranking

$w$	$r$	$f_r$
<i>very</i>	1	5
<i>not</i>	2	3
<i>merely</i>	3	2
<i>much</i>	4	2
<i>now</i>	5	1
<i>otherwise</i>	6	1
<i>recently</i>	7	1

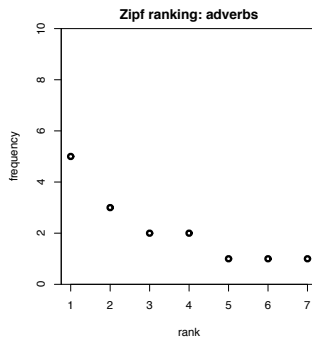
# Zipf ranking

our sample: *recently, very, not, otherwise, much, very, very, merely, not, now, very, much, merely, not, very*

- ▶  $N = 15$ : number of **tokens** = sample size
- ▶  $V = 7$ : number of distinct **types** = **vocabulary size** (*recently, very, not, otherwise, much, merely, now*)

## Zipf ranking

$w$	$r$	$f_r$
<i>very</i>	1	5
<i>not</i>	2	3
<i>merely</i>	3	2
<i>much</i>	4	2
<i>now</i>	5	1
<i>otherwise</i>	6	1
<i>recently</i>	7	1

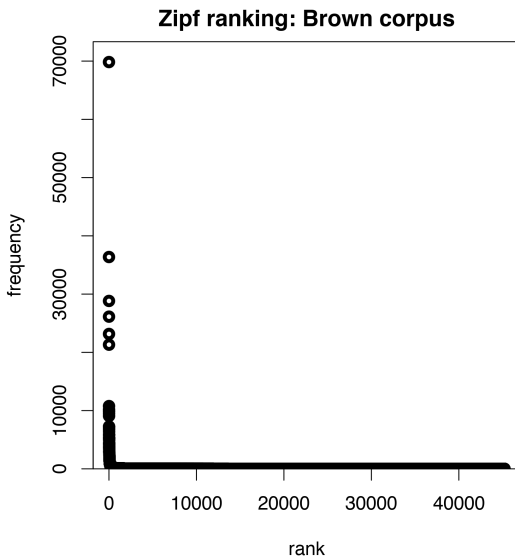




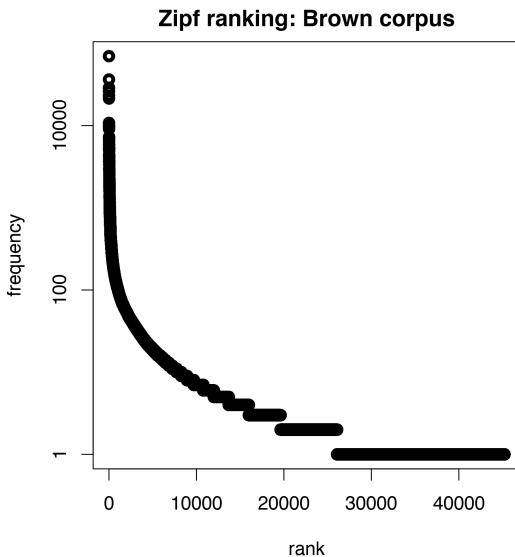
# A realistic Zipf ranking: the Brown corpus

top frequencies			bottom frequencies		
<i>r</i>	<i>f</i>	word	rank range	<i>f</i>	randomly selected examples
1	69836	the	7731 – 8271	10	schedules, polynomials, bleak
2	36365	of	8272 – 8922	9	tolerance, shaved, hymn
3	28826	and	8923 – 9703	8	decreased, abolish, irresistible
4	26126	to	9704 – 10783	7	immunity, cruising, titan
5	23157	a	10784 – 11985	6	geographic, lauro, portrayed
6	21314	in	11986 – 13690	5	grigori, slashing, developer
7	10777	that	13691 – 15991	4	sheath, gaulle, ellipsoids
8	10182	is	15992 – 19627	3	mc, initials, abstracted
9	9968	was	19628 – 26085	2	thar, slackening, deluxe
10	9801	he	26086 – 45215	1	beck, encompasses, second-place

# A realistic Zipf ranking: the Brown corpus



# A realistic Zipf ranking: the Brown corpus



# Frequency spectrum

- ▶ pool types with  $f = 1$  (**hapax legomena**), types with  $f = 2$  (**dis legomena**),  $\dots$ ,  $f = m$ ,  $\dots$
- ▶  $V_1 = 3$ : number of hapax legomena (*now, otherwise, recently*)
- ▶  $V_2 = 2$ : number of dis legomena (*merely, much*)
- ▶ general definition:  $V_m = |\{w \mid f_w = m\}|$

## Zipf ranking

$w$	$r$	$f_r$
<i>very</i>	1	5
<i>not</i>	2	3
<i>merely</i>	3	2
<i>much</i>	4	2
<i>now</i>	5	1
<i>otherwise</i>	6	1
<i>recently</i>	7	1

## frequency spectrum

$m$	$V_m$
1	3
2	2
3	1
5	1

# Frequency spectrum

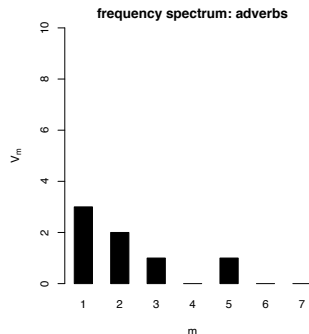
- ▶ pool types with  $f = 1$  (**hapax legomena**), types with  $f = 2$  (**dis legomena**),  $\dots$ ,  $f = m$ ,  $\dots$
- ▶  $V_1 = 3$ : number of hapax legomena (*now, otherwise, recently*)
- ▶  $V_2 = 2$ : number of dis legomena (*merely, much*)
- ▶ general definition:  $V_m = |\{w \mid f_w = m\}|$

## Zipf ranking

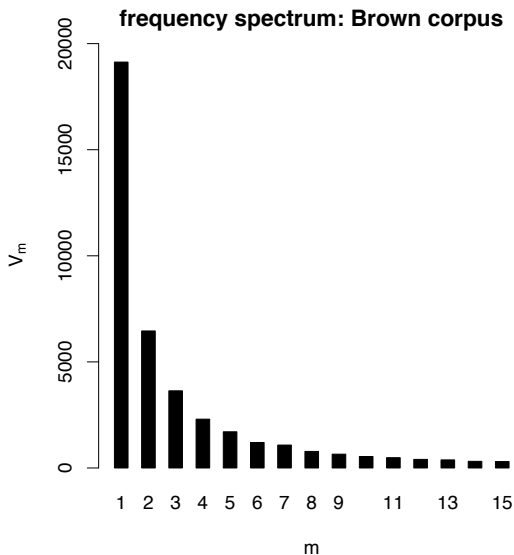
$w$	$r$	$f_r$
<i>very</i>	1	5
<i>not</i>	2	3
<i>merely</i>	3	2
<i>much</i>	4	2
<i>now</i>	5	1
<i>otherwise</i>	6	1
<i>recently</i>	7	1

## frequency spectrum

$m$	$V_m$
1	3
2	2
3	1
5	1



# A realistic frequency spectrum: the Brown corpus



# Vocabulary growth curve

our sample: *recently*, *very*, *not*, *otherwise*, *much*, *very*, *very*,  
*merely*, *not*, *now*, *very*, *much*, *merely*, *not*, *very*

►  $N = 1, V(N) = 1, V_1(N) = 1$

# Vocabulary growth curve

our sample: *recently*, *very*, *not*, *otherwise*, *much*, *very*, *very*,  
*merely*, *not*, *now*, *very*, *much*, *merely*, *not*, *very*

►  $N = 1, V(N) = 1, V_1(N) = 1$

►  $N = 3, V(N) = 3, V_1(N) = 3$



# Vocabulary growth curve

our sample: *recently, very, not, otherwise, much, very, very, merely, not, now, very, much, merely, not, very*

- ▶  $N = 1, V(N) = 1, V_1(N) = 1$
- ▶  $N = 3, V(N) = 3, V_1(N) = 3$
- ▶  $N = 7, V(N) = 5, V_1(N) = 4$

# Vocabulary growth curve

our sample: *recently*, *very*, *not*, *otherwise*, *much*, *very*, *very*,  
*merely*, *not*, *now*, *very*, *much*, *merely*, *not*, *very*

- ▶  $N = 1$ ,  $V(N) = 1$ ,  $V_1(N) = 1$
- ▶  $N = 3$ ,  $V(N) = 3$ ,  $V_1(N) = 3$
- ▶  $N = 7$ ,  $V(N) = 5$ ,  $V_1(N) = 4$
- ▶  $N = 12$ ,  $V(N) = 7$ ,  $V_1(N) = 4$

# Vocabulary growth curve

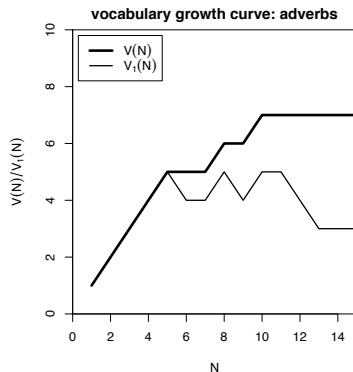
our sample: *recently, very, not, otherwise, much, very, very, merely, not, now, very, much, merely, not, very*

- ▶  $N = 1, V(N) = 1, V_1(N) = 1$
- ▶  $N = 3, V(N) = 3, V_1(N) = 3$
- ▶  $N = 7, V(N) = 5, V_1(N) = 4$
- ▶  $N = 12, V(N) = 7, V_1(N) = 4$
- ▶  $N = 15, V(N) = 7, V_1(N) = 3$

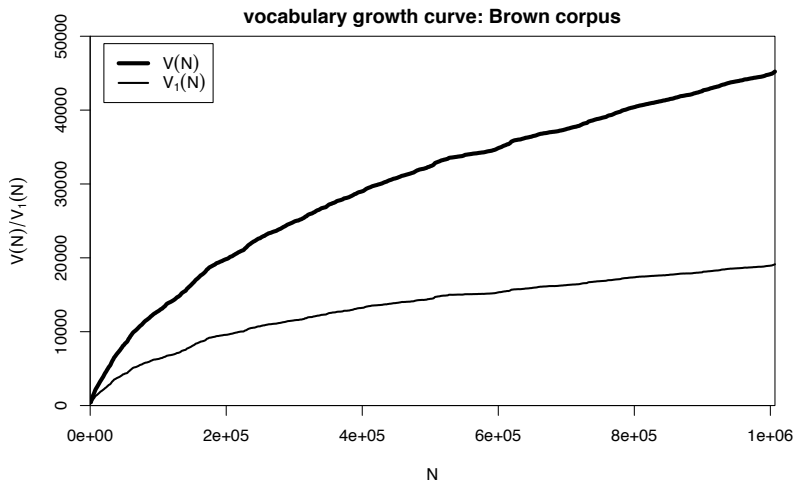
## Vocabulary growth curve

our sample: *recently, very, not, otherwise, much, very, very, merely, not, now, very, much, merely, not, very*

- ▶  $N = 1, V(N) = 1, V_1(N) = 1$
- ▶  $N = 3, V(N) = 3, V_1(N) = 3$
- ▶  $N = 7, V(N) = 5, V_1(N) = 4$
- ▶  $N = 12, V(N) = 7, V_1(N) = 4$
- ▶  $N = 15, V(N) = 7, V_1(N) = 3$



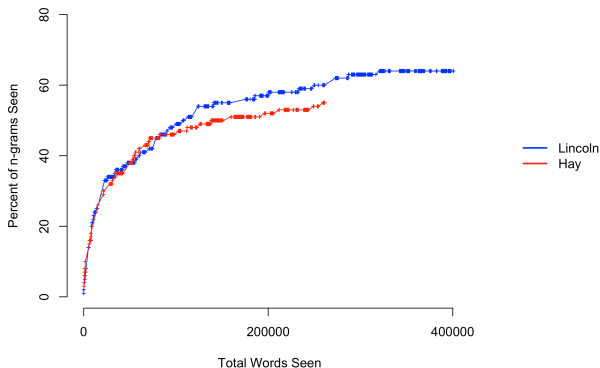
# A realistic vocabulary growth curve: the Brown corpus



# Vocabulary growth in authorship attribution

- ▶ Authorship attribution by n-gram tracing applied to the case of the Bixby letter (Grieve *et al.* 2018)
- ▶ Word or character n-grams in disputed text are compared against large “training” corpora from candidate authors

Gettysburg Address: Word 2-Grams



# Outline

## Introduction

Motivation

Notation & basic concepts

Zipf's law

First steps (zipfR)

## LNRE models

Population & samples

The mathematics of LNRE

## Applications & examples

Productivity &

lexical diversity

Practical LNRE modelling

Bootstrapping experiments

LNRE as Bayesian prior

## Challenges

Model inference

Zipf's law

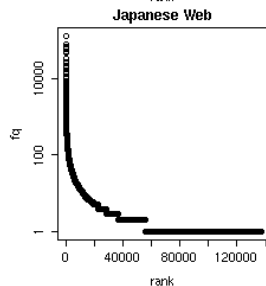
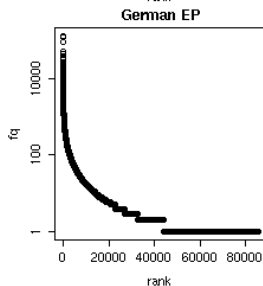
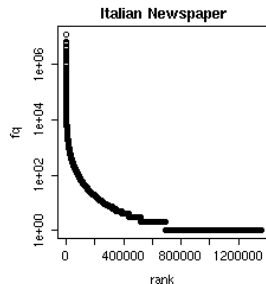
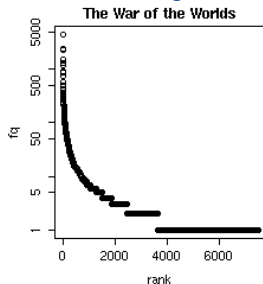
Non-randomness

Significance testing

Outlook

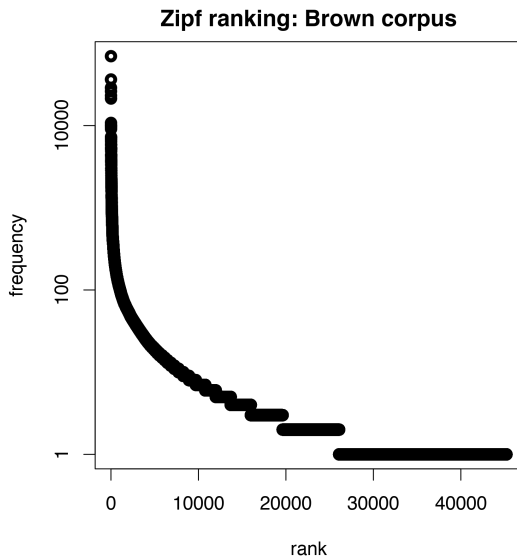
# Observing Zipf's law

across languages and different linguistic units

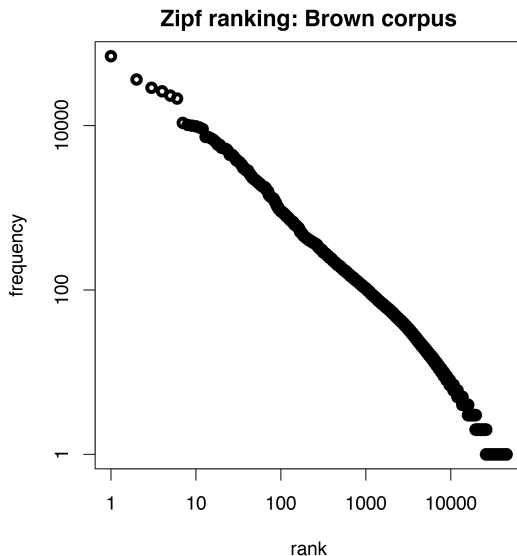




# Observing Zipf's law



# Observing Zipf's law



# Observing Zipf's law

- ▶ Straight line in double-logarithmic space corresponds to **power law** for original variables
- ▶ This leads to Zipf's (1949; 1965) famous law:

$$f_r = \frac{C}{r^a}$$

# Observing Zipf's law

- ▶ Straight line in double-logarithmic space corresponds to **power law** for original variables
- ▶ This leads to Zipf's (1949; 1965) famous law:

$$f_r = \frac{C}{r^a}$$

- ▶ If we take logarithm on both sides, we obtain:

$$\log f_r = \log C - a \cdot \log r$$

# Observing Zipf's law

- ▶ Straight line in double-logarithmic space corresponds to **power law** for original variables
- ▶ This leads to Zipf's (1949; 1965) famous law:

$$f_r = \frac{C}{r^a}$$

- ▶ If we take logarithm on both sides, we obtain:

$$\underbrace{\log f_r}_y = \log C - a \cdot \underbrace{\log r}_x$$

# Observing Zipf's law

- ▶ Straight line in double-logarithmic space corresponds to **power law** for original variables
- ▶ This leads to Zipf's (1949; 1965) famous law:

$$f_r = \frac{C}{r^a}$$

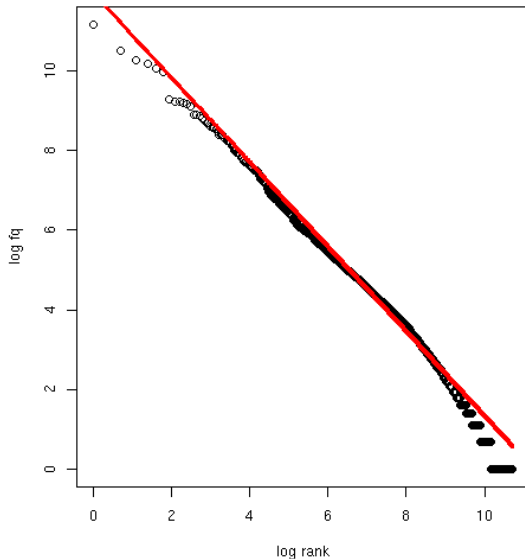
- ▶ If we take logarithm on both sides, we obtain:

$$\underbrace{\log f_r}_y = \log C - a \cdot \underbrace{\log r}_x$$

- ▶ Intuitive interpretation of  $a$  and  $C$ :
  - ▶  $a$  is **slope** determining how fast log frequency decreases
  - ▶  $\log C$  is **intercept**, i.e. log frequency of most frequent word ( $r = 1 \rightarrow \log r = 0$ )

# Observing Zipf's law

Least-squares fit = linear regression in log-space (Brown corpus)



# Zipf-Mandelbrot law

Mandelbrot (1953, 1962)

- ▶ Mandelbrot's extra parameter:

$$f_r = \frac{C}{(r + b)^a}$$

- ▶ Zipf's law is special case with  $b = 0$



# Zipf-Mandelbrot law

Mandelbrot (1953, 1962)

- ▶ Mandelbrot's extra parameter:

$$f_r = \frac{C}{(r + b)^a}$$

- ▶ Zipf's law is special case with  $b = 0$
- ▶ Assuming  $a = 1$ ,  $C = 60,000$ ,  $b = 1$ :
  - ▶ For word with rank 1, Zipf's law predicts frequency of 60,000; Mandelbrot's variation predicts frequency of 30,000
  - ▶ For word with rank 1,000, Zipf's law predicts frequency of 60; Mandelbrot's variation predicts frequency of 59.94

# Zipf-Mandelbrot law

Mandelbrot (1953, 1962)

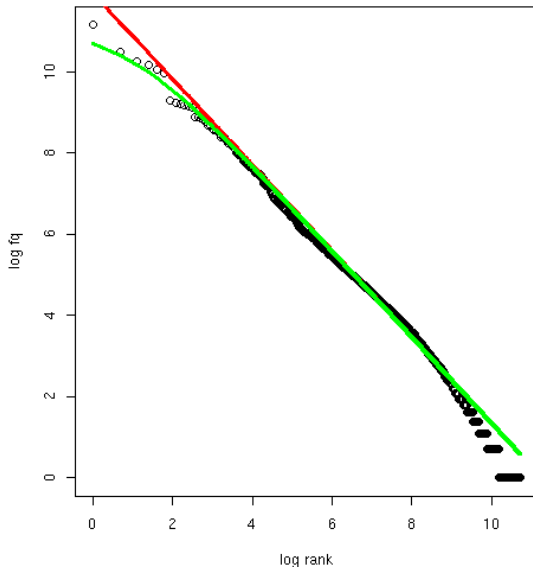
- ▶ Mandelbrot's extra parameter:

$$f_r = \frac{C}{(r + b)^a}$$

- ▶ Zipf's law is special case with  $b = 0$
- ▶ Assuming  $a = 1$ ,  $C = 60,000$ ,  $b = 1$ :
  - ▶ For word with rank 1, Zipf's law predicts frequency of 60,000; Mandelbrot's variation predicts frequency of 30,000
  - ▶ For word with rank 1,000, Zipf's law predicts frequency of 60; Mandelbrot's variation predicts frequency of 59.94
- ▶ Zipf-Mandelbrot law forms basis of statistical LNRE models
  - ▶ ZM law derived mathematically as limiting distribution of vocabulary generated by a character-level Markov process

# Zipf-Mandelbrot law

Non-linear least-squares fit (Brown corpus)



# Outline

## Introduction

Motivation

Notation & basic concepts

Zipf's law

First steps (zipfR)

## LNRE models

Population & samples

The mathematics of LNRE

## Applications & examples

Productivity &

lexical diversity

Practical LNRE modelling

Bootstrapping experiments

LNRE as Bayesian prior

## Challenges

Model inference

Zipf's law

Non-randomness

Significance testing

Outlook

# zipfR

Evert and Baroni (2007)

- ▶ <http://zipfR.R-Forge.R-Project.org/>
- ▶ Conveniently available from CRAN repository
- ▶ Package vignette = gentle tutorial introduction



# First steps with zipfR

- ▶ Set up a folder for this course, and make sure it is your working directory in R (preferably as an RStudio project)
- ▶ Install the most recent version of the zipfR package
  - ▶ tutorial requires version 0.7 or newer
- ▶ Package, handouts, code samples & data sets available from <http://zipfr.r-forge.r-project.org/lrec2018.html>

```
> library(zipfR)
```

```
> ?zipfR    # documentation entry point
```

```
> vignette("zipfr-tutorial")    # read the zipfR tutorial
```

# Loading type-token data

- ▶ Most convenient input: sequence of tokens as text file in vertical format (“one token per line”)
  - 👉 mapped to appropriate types: normalized word forms, word pairs, lemmatized, semantic class, n-gram of POS tags, ...
  - 👉 language data should always be in UTF-8 encoding!
  - 👉 large files can be compressed (.gz, .bz2, .xz)

# Loading type-token data

- ▶ Most convenient input: sequence of tokens as text file in vertical format (“one token per line”)
  - 👉 mapped to appropriate types: normalized word forms, word pairs, lemmatized, semantic class, n-gram of POS tags, ...
  - 👉 language data should always be in UTF-8 encoding!
  - 👉 large files can be compressed (.gz, .bz2, .xz)
- ▶ Sample data: `brown_adverbs.txt` on tutorial homepage
  - ▶ lowercased adverb tokens from Brown corpus (original order)
  - 👉 download and save to your working directory



# Loading type-token data

- ▶ Most convenient input: sequence of tokens as text file in vertical format (“one token per line”)
  - 👉 mapped to appropriate types: normalized word forms, word pairs, lemmatized, semantic class, n-gram of POS tags, ...
  - 👉 language data should always be in UTF-8 encoding!
  - 👉 large files can be compressed (.gz, .bz2, .xz)
- ▶ Sample data: `brown_adverbs.txt` on tutorial homepage
  - ▶ lowercased adverb tokens from Brown corpus (original order)
  - 👉 download and save to your working directory

```
> adv <- readLines("brown_adverbs.txt", encoding="UTF-8")
```

```
> head(adv, 30) # mathematically, a “vector” of tokens
```

```
> length(adv)   # sample size = 52,037 tokens
```

## Descriptive statistics: type-frequency list

```
> adv.tfl <- vec2tfl(adv)
```

```
> adv.tfl
```

	k	f	type
not	1	4859	not
n't	2	2084	n't
so	3	1464	so
only	4	1381	only
then	5	1374	then
now	6	1309	now
even	7	1134	even
as	8	1089	as
	:	:	:

N V

52037 1907

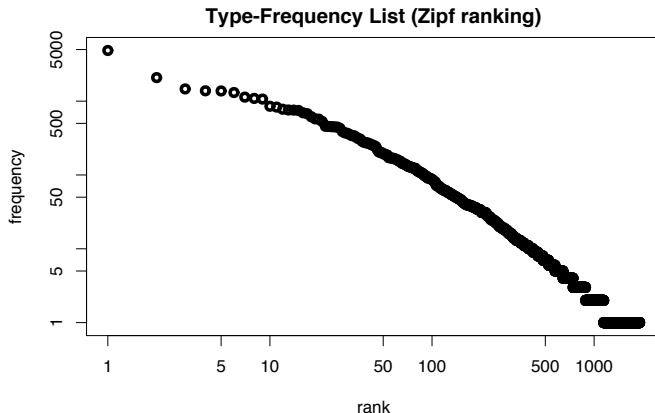
```
> N(adv.tfl) # sample size
```

```
> V(adv.tfl) # type count
```

# Descriptive statistics: type-frequency list

- Visualize descriptive statistics with plot method

```
> plot(adv.tfl) # Zipf ranking  
> plot(adv.tfl, log="xy") # logarithmic scale recommended
```



# Descriptive statistics: frequency spectrum

```
> adv.spc <- tfl2spc(adv.tfl) # or directly with vec2spc
```

```
> adv.spc
```

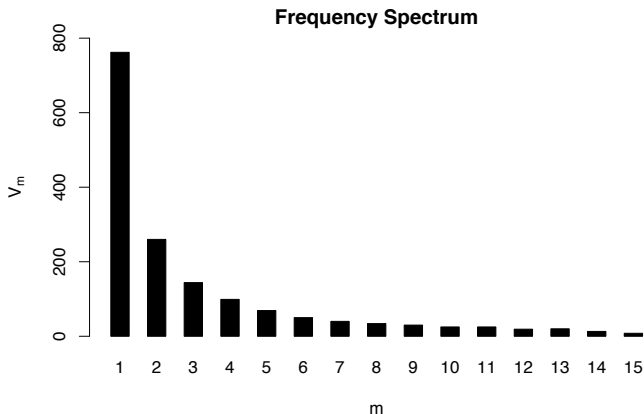
	m	Vm
1	1	762
2	2	260
3	3	144
4	4	99
5	5	69
6	6	50
7	7	40
8	8	34
	:	:
	N	V
	52037	1907

```
> N(adv.spc) # sample size
```

```
> V(adv.spc) # type count
```

# Descriptive statistics: frequency spectrum

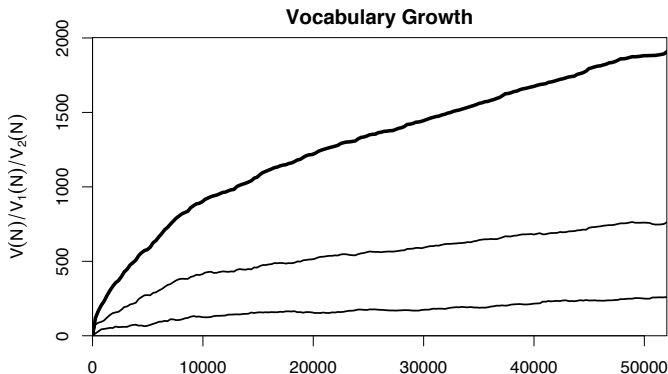
```
> plot(adv.spc)           # barplot of frequency spectrum  
> ?plot.spc              # see help page for further options
```



# Descriptive statistics: vocabulary growth

- ▶ VGC lists vocabulary size  $V(N)$  at different sample sizes  $N$
- ▶ Optionally also spectrum elements  $V_m(N)$  up to  $m.max$

```
> adv.vgc <- vec2vgc(adv, m.max=2)  
> plot(adv.vgc, add.m=1:2) # plot all three VGCs
```



## Further example data sets

`?Brown` words from Brown corpus

`?BrownSubsets` various subsets

`?Dickens` words from novels by Charles Dickens

`?ItaPref` Italian word-formation prefixes

`?TigerNP` NP and PP patterns from German Tiger treebank

`?Baayen2001` frequency spectra from Baayen (2001)

`?EvertLuedeling2001` German word-formation affixes (manually corrected data from Evert and Lüdeling 2001)

### Practice:

- ▶ Explore these data sets with descriptive statistics
- ▶ Try different plot options (from help pages `?plot.tfl`, `?plot.spc`, `?plot.vgc`)

# Outline

## Introduction

Motivation

Notation & basic concepts

Zipf's law

First steps (zipfR)

## LNRE models

Population & samples

The mathematics of LNRE

## Applications & examples

Productivity &

lexical diversity

Practical LNRE modelling

Bootstrapping experiments

LNRE as Bayesian prior

## Challenges

Model inference

Zipf's law

Non-randomness

Significance testing

Outlook



# Why do we need statistics?

- ▶ Often want to compare samples of different sizes
  - ✎ extrapolation of VGC & productivity measures

# Why do we need statistics?

- ▶ Often want to compare samples of different sizes
  - ✎ extrapolation of VGC & productivity measures
- ▶ Interested in productivity of affix, vocabulary of author, ... ; not in a particular text or sample
  - ✎ statistical inference from sample to population
  - ✎ significance of differences in productivity

# Why do we need statistics?

- ▶ Often want to compare samples of different sizes
  - 👉 extrapolation of VGC & productivity measures
- ▶ Interested in productivity of affix, vocabulary of author, ... ; not in a particular text or sample
  - 👉 statistical inference from sample to population
  - 👉 significance of differences in productivity
- ▶ Discrete frequency counts are difficult to capture with generalizations such as Zipf's law
  - 👉 Zipf's law predicts many impossible types with  $1 < f_r < 2$
  - 👉 population does not suffer from such quantization effects

# LNRE models

- ▶ This tutorial introduces the state-of-the-art LNRE approach proposed by Baayen (2001)
  - ▶ LNRE = Large Number of Rare Events
- ▶ LNRE uses various approximations and simplifications to obtain a tractable and elegant model
- ▶ Of course, we could also estimate the precise discrete distributions using MCMC simulations, but ...
  1. LNRE model usually minor component of complex procedure
  2. often applied to very large samples ( $N > 1$  M tokens)
  3. still better than naive least-squares regression on Zipf ranking

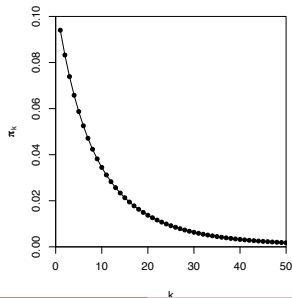
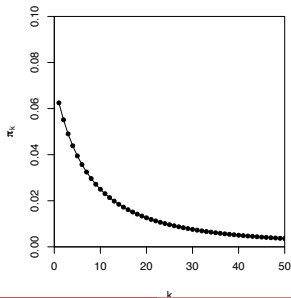
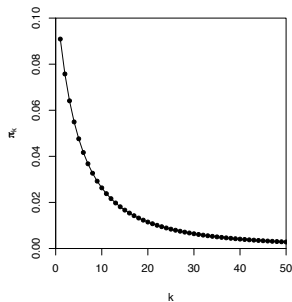
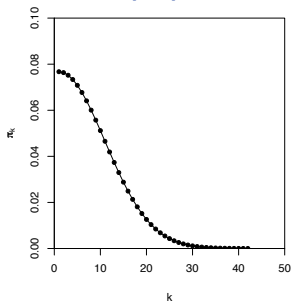
# The LNRE population

- ▶ Population: set of  $S$  types  $w_i$  with occurrence **probabilities**  $\pi_i$
- ▶  $S =$  **population diversity** can be finite or infinite ( $S = \infty$ )
- ▶ Not interested in specific types  $\rightarrow$  arrange by decreasing probability:  $\pi_1 \geq \pi_2 \geq \pi_3 \geq \dots$ 
  - 👉 impossible to determine probabilities of all individual types
- ▶ Normalization:  $\pi_1 + \pi_2 + \dots + \pi_S = 1$

# The LNRE population

- ▶ Population: set of  $S$  types  $w_i$  with occurrence **probabilities**  $\pi_i$
- ▶  $S =$  **population diversity** can be finite or infinite ( $S = \infty$ )
- ▶ Not interested in specific types  $\rightarrow$  arrange by decreasing probability:  $\pi_1 \geq \pi_2 \geq \pi_3 \geq \dots$ 
  - 👉 impossible to determine probabilities of all individual types
- ▶ Normalization:  $\pi_1 + \pi_2 + \dots + \pi_S = 1$
- ▶ Need **parametric** statistical **model** to describe full population (esp. for  $S = \infty$ ), i.e. a function  $i \mapsto \pi_i$ 
  - ▶ type probabilities  $\pi_i$  cannot be estimated reliably from a sample, but parameters of this function can
  - ▶ NB: population index  $i \neq$  Zipf rank  $r$

# What should the population look like?



# Zipf-Mandelbrot law as a population model

- ▶ Zipf-Mandelbrot law for type probabilities:

$$\pi_i := \frac{C}{(i + b)^a}$$



# Zipf-Mandelbrot law as a population model

- ▶ Zipf-Mandelbrot law for type probabilities:

$$\pi_i := \frac{C}{(i + b)^a}$$

- ▶ Two free parameters:  $a > 1$  and  $b \geq 0$ 
  - 👉  $C$  is not a parameter but a normalization constant, needed to ensure that  $\sum_i \pi_i = 1$

# Zipf-Mandelbrot law as a population model

- ▶ Zipf-Mandelbrot law for type probabilities:

$$\pi_i := \frac{C}{(i + b)^a}$$

- ▶ Two free parameters:  $a > 1$  and  $b \geq 0$ 
  - 👉  $C$  is not a parameter but a normalization constant, needed to ensure that  $\sum_i \pi_i = 1$
- ▶ Third parameter:  $S > 0$  or  $S = \infty$

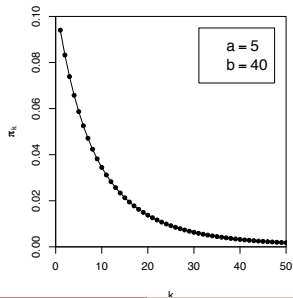
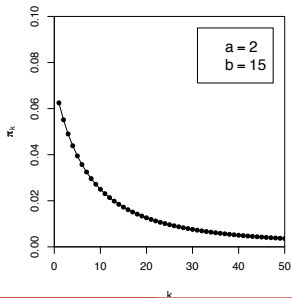
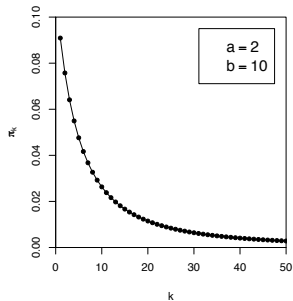
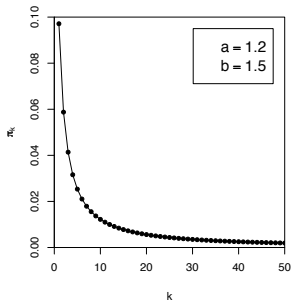
# Zipf-Mandelbrot law as a population model

- ▶ Zipf-Mandelbrot law for type probabilities:

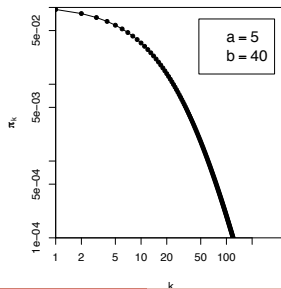
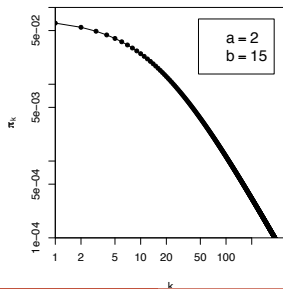
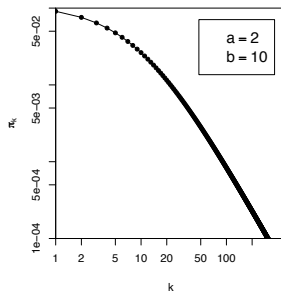
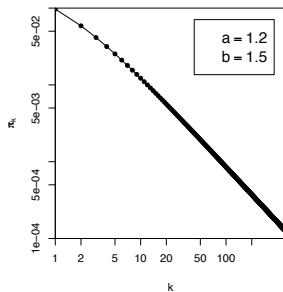
$$\pi_i := \frac{C}{(i + b)^a}$$

- ▶ Two free parameters:  $a > 1$  and  $b \geq 0$ 
  - 👉  $C$  is not a parameter but a normalization constant, needed to ensure that  $\sum_i \pi_i = 1$
- ▶ Third parameter:  $S > 0$  or  $S = \infty$
- ▶ This is the **Zipf-Mandelbrot** population model (Evert 2004)
  - ▶ **ZM** for Zipf-Mandelbrot model ( $S = \infty$ )
  - ▶ **fZM** for finite Zipf-Mandelbrot model

# The parameters of the Zipf-Mandelbrot model

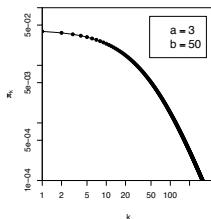
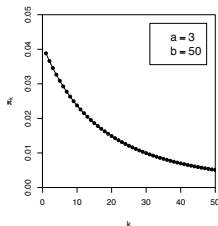


# The parameters of the Zipf-Mandelbrot model



# Sampling from a population model

Assume we believe that the population we are interested in can be described by a Zipf-Mandelbrot model:



Use computer simulation to generate random samples:

- ▶ Draw  $N$  tokens from the population such that in each step, type  $w_i$  has probability  $\pi_i$  to be picked
- ▶ This allows us to make predictions for samples (= corpora) of arbitrary size  $N$

# Sampling from a population model

#1: 1 42 34 23 108 18 48 18 1 ...

# Sampling from a population model

#1:    1    42    34    23 108    18 48    18    1    ...  
      time order room school town course area course time    ...



# Sampling from a population model

#1: 1 42 34 23 108 18 48 18 1 ...  
time order room school town course area course time ...

#2: 286 28 23 36 3 4 7 4 8 ...

# Sampling from a population model

#1: 1 42 34 23 108 18 48 18 1 ...  
time order room school town course area course time ...

#2: 286 28 23 36 3 4 7 4 8 ...

#3: 2 11 105 21 11 17 17 1 16 ...

# Sampling from a population model

#1:	1	42	34	23	108	18	48	18	1	...
	time	order	room	school	town	course	area	course	time	...
#2:	286	28	23	36	3	4	7	4	8	...
#3:	2	11	105	21	11	17	17	1	16	...
#4:	44	3	110	34	223	2	25	20	28	...
#5:	24	81	54	11	8	61	1	31	35	...
#6:	3	65	9	165	5	42	16	20	7	...
#7:	10	21	11	60	164	54	18	16	203	...
#8:	11	7	147	5	24	19	15	85	37	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

# Samples: type frequency list & spectrum

rank $r$	$f_r$	type $i$
1	37	6
2	36	1
3	33	3
4	31	7
5	31	10
6	30	5
7	28	12
8	27	2
9	24	4
10	24	16
11	23	8
12	22	14
$\vdots$	$\vdots$	$\vdots$

$m$	$V_m$
1	83
2	22
3	20
4	12
5	10
6	5
7	5
8	3
9	3
10	3
$\vdots$	$\vdots$

sample #1

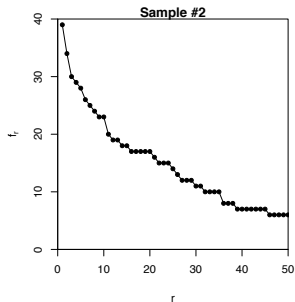
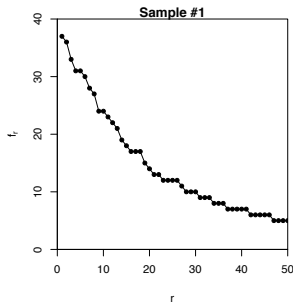
# Samples: type frequency list & spectrum

rank $r$	$f_r$	type $i$
1	39	2
2	34	3
3	30	5
4	29	10
5	28	8
6	26	1
7	25	13
8	24	7
9	23	6
10	23	11
11	20	4
12	19	17
$\vdots$	$\vdots$	$\vdots$

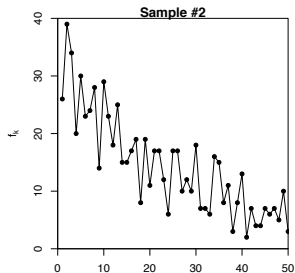
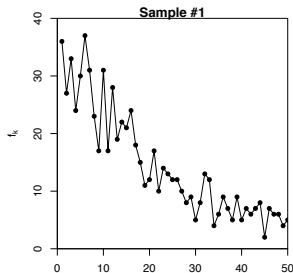
$m$	$V_m$
1	76
2	27
3	17
4	10
5	6
6	5
7	7
8	3
10	4
11	2
$\vdots$	$\vdots$

sample #2

# Random variation in type-frequency lists

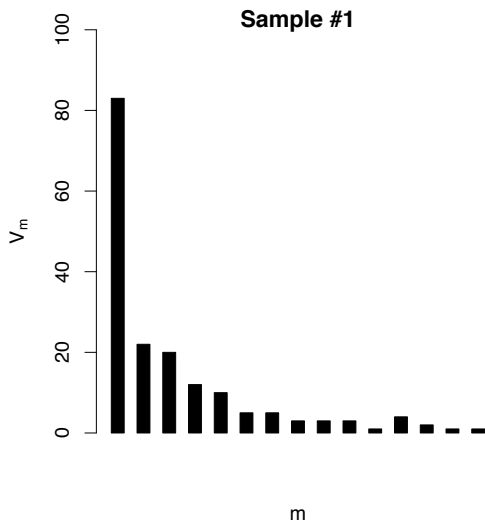


$$r \leftrightarrow f_r$$

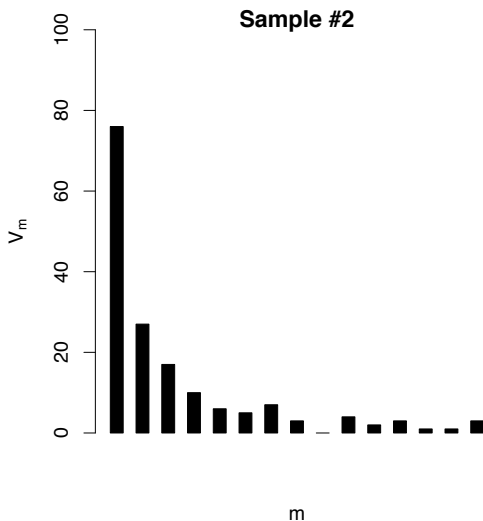


$$i \leftrightarrow f_i$$

# Random variation: frequency spectrum

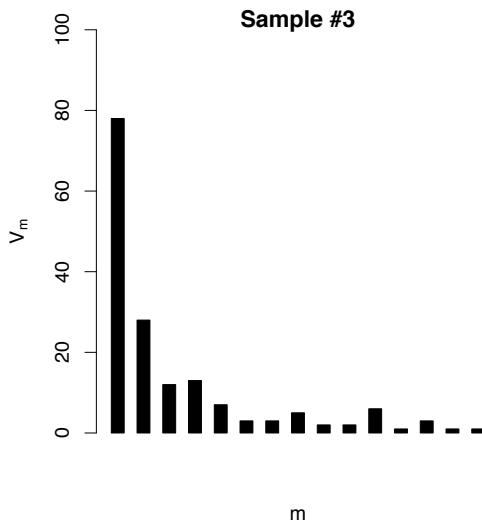


# Random variation: frequency spectrum

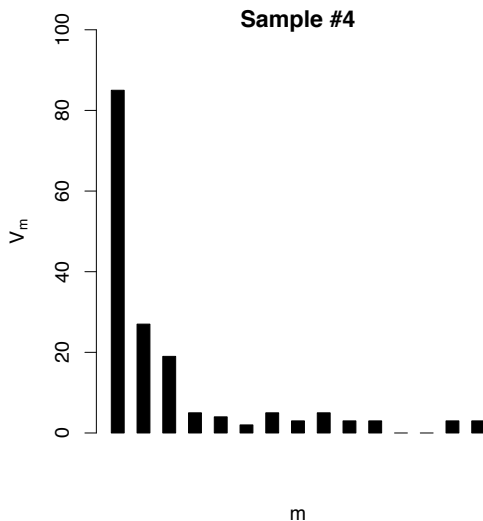




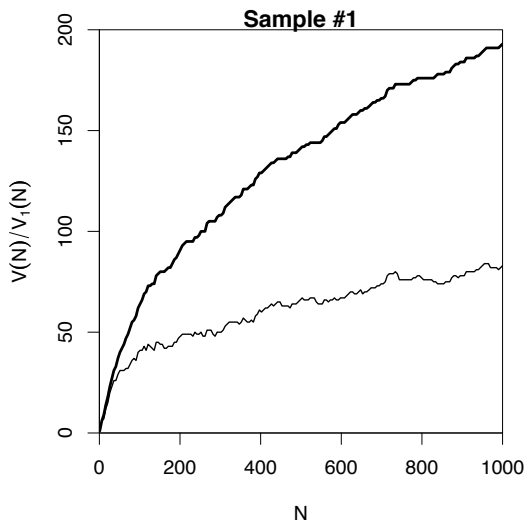
# Random variation: frequency spectrum



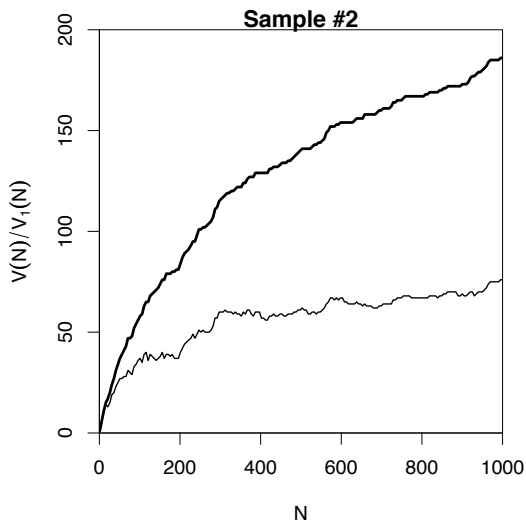
# Random variation: frequency spectrum



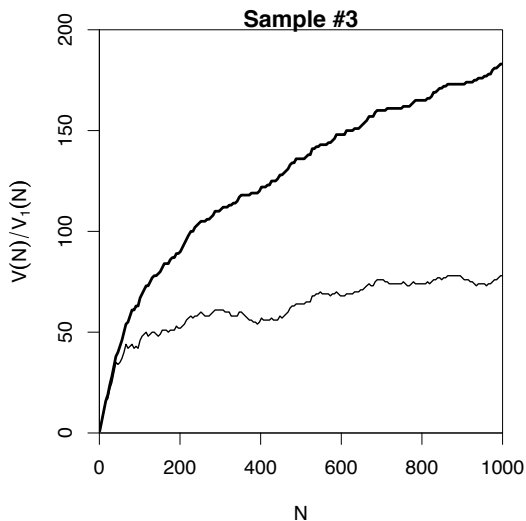
# Random variation: vocabulary growth curve



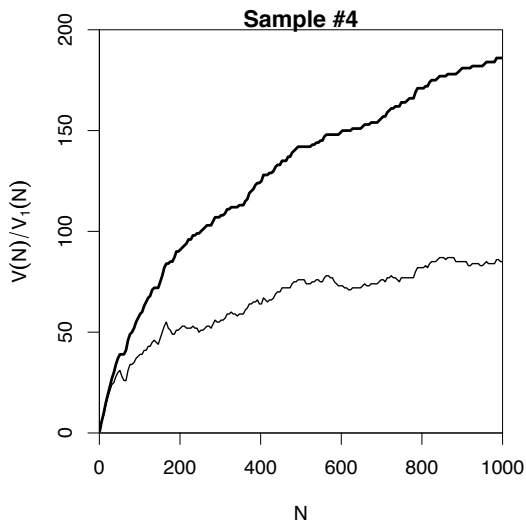
# Random variation: vocabulary growth curve



# Random variation: vocabulary growth curve



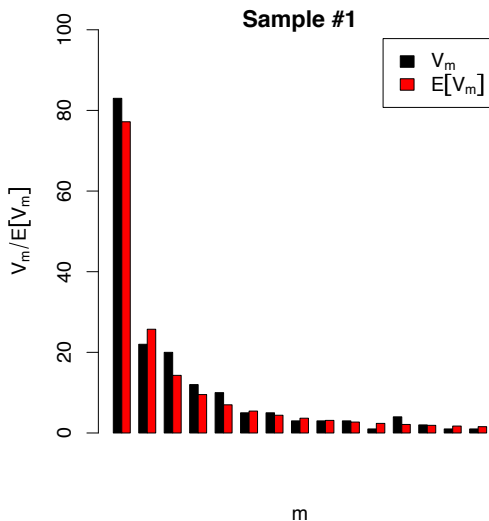
# Random variation: vocabulary growth curve



# Expected values

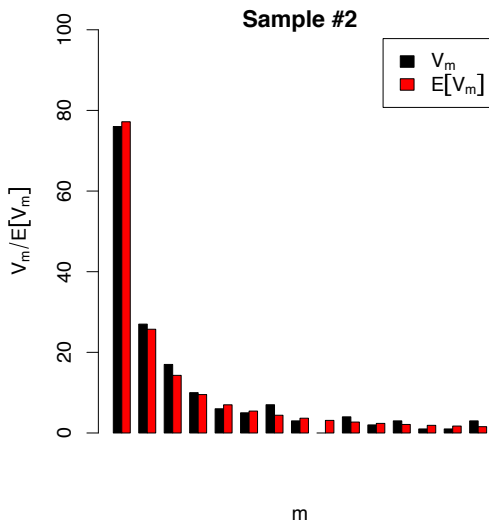
- ▶ There is no reason why we should choose a particular sample to compare to the real data or make a prediction – each one is equally likely or unlikely
- ▶ Take the average over a large number of samples, called **expected value** or **expectation** in statistics
- ▶ Notation:  $E[V(N)]$  and  $E[V_m(N)]$ 
  - ▶ indicates that we are referring to expected values for a sample of size  $N$
  - ▶ rather than to the specific values  $V$  and  $V_m$  observed in a particular sample or a real-world data set
- ▶ Expected values can be calculated efficiently *without* generating thousands of random samples

# The expected frequency spectrum

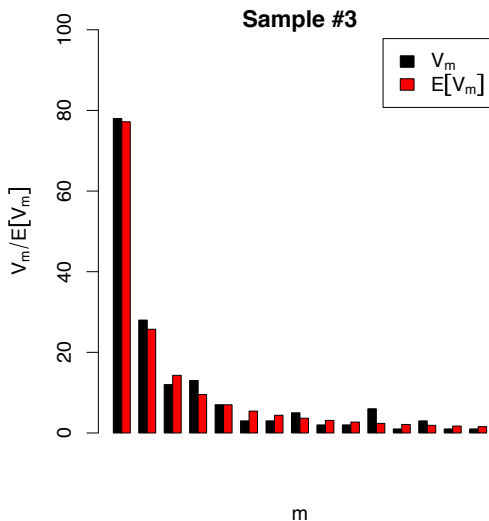




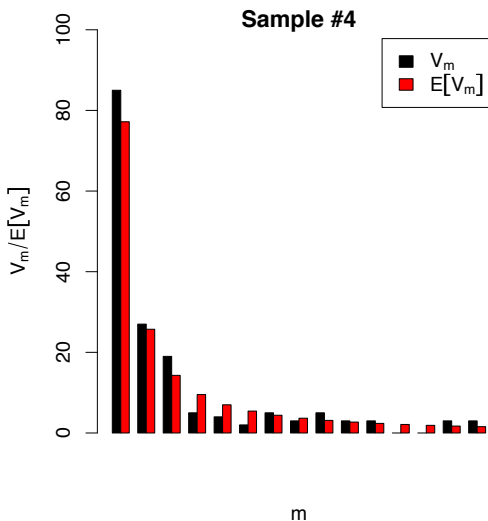
# The expected frequency spectrum



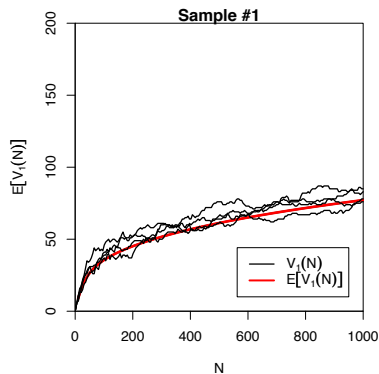
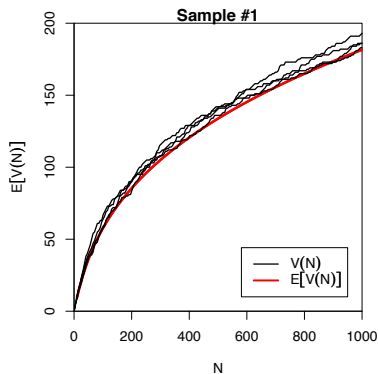
# The expected frequency spectrum



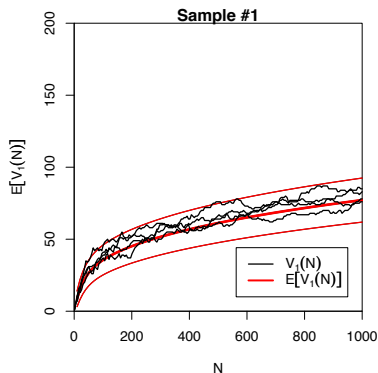
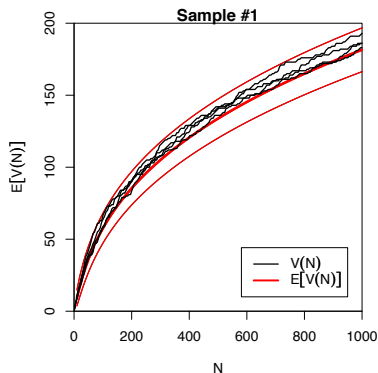
# The expected frequency spectrum



# The expected vocabulary growth curve



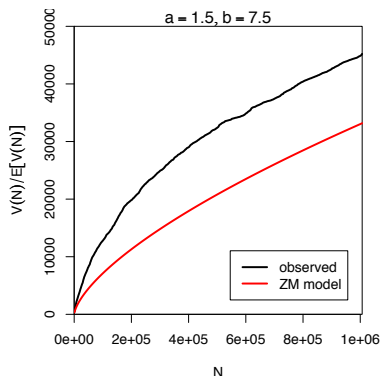
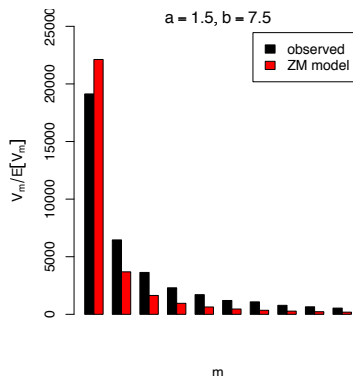
# Prediction intervals for the expected VGC



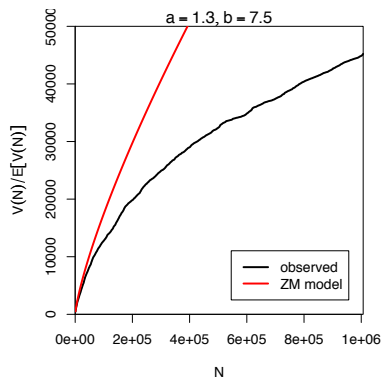
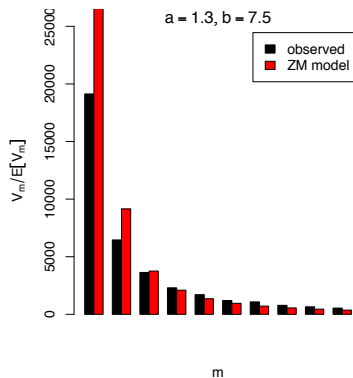
“Confidence intervals” indicate predicted sampling distribution:

- 👉 for 95% of samples generated by the LNRE model, VGC will fall within the range delimited by the thin red lines

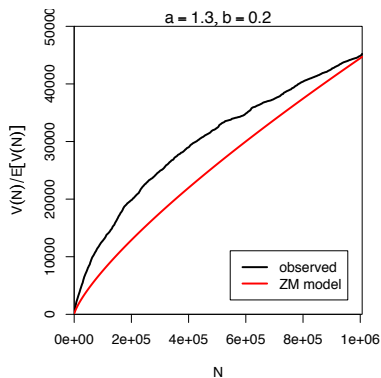
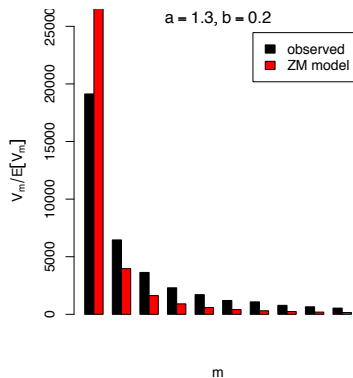
# Parameter estimation by trial & error



# Parameter estimation by trial & error

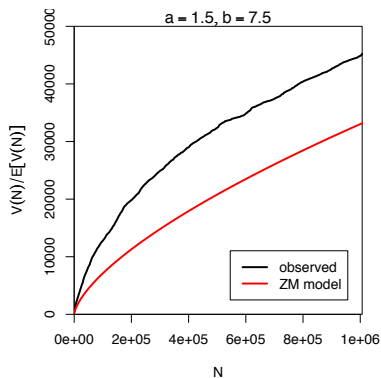
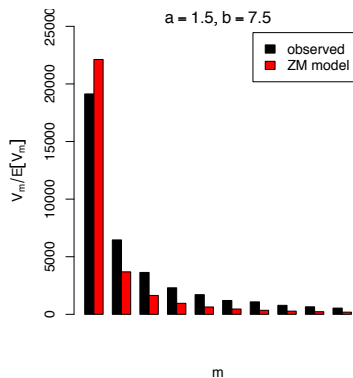


# Parameter estimation by trial & error

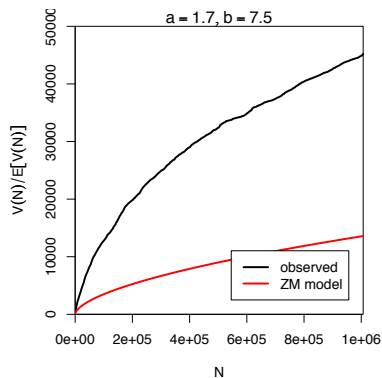
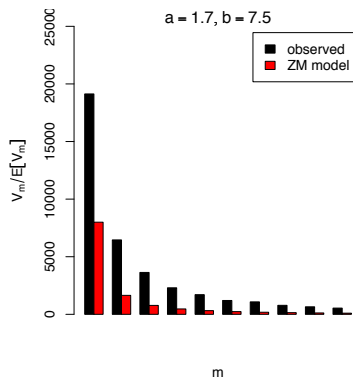




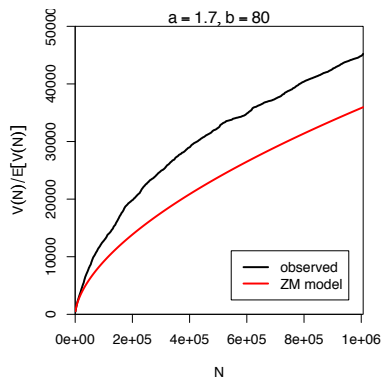
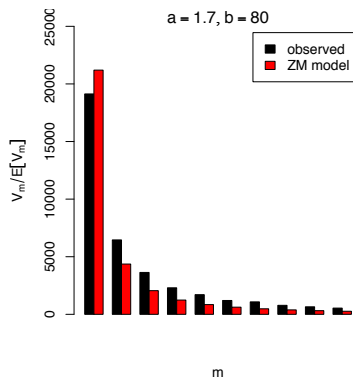
# Parameter estimation by trial & error



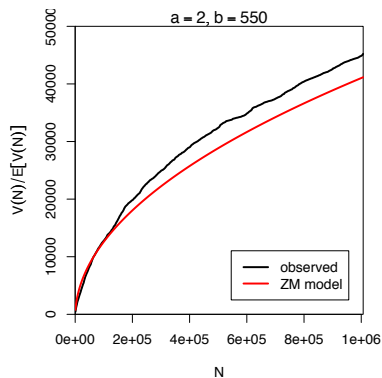
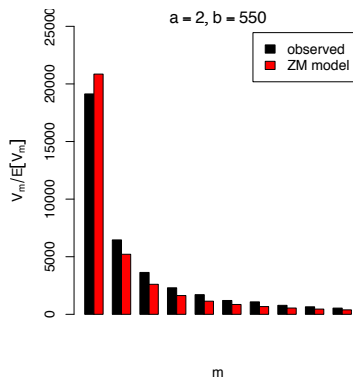
# Parameter estimation by trial & error



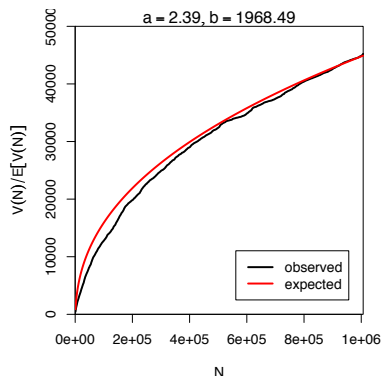
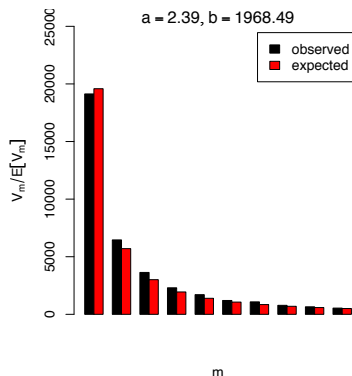
# Parameter estimation by trial & error



# Parameter estimation by trial & error



# Automatic parameter estimation



- ▶ By trial & error we found  $a = 2.0$  and  $b = 550$
- ▶ Automatic estimation procedure:  $a = 2.39$  and  $b = 1968$

# Outline

## Introduction

Motivation

Notation & basic concepts

Zipf's law

First steps (zipfR)

## LNRE models

Population & samples

The mathematics of LNRE

## Applications & examples

Productivity &

lexical diversity

Practical LNRE modelling

Bootstrapping experiments

LNRE as Bayesian prior

## Challenges

Model inference

Zipf's law

Non-randomness

Significance testing

Outlook

# Outline

## Introduction

Motivation

Notation & basic concepts

Zipf's law

First steps (zipfR)

## LNRE models

Population & samples

The mathematics of LNRE

## Applications & examples

Productivity &  
lexical diversity

Practical LNRE modelling

Bootstrapping experiments

LNRE as Bayesian prior

## Challenges

Model inference

Zipf's law

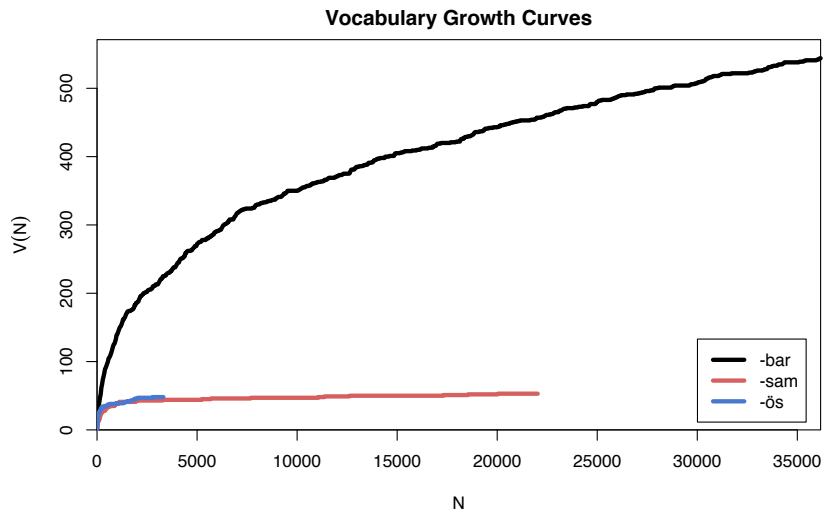
Non-randomness

Significance testing

Outlook

# Measuring morphological productivity

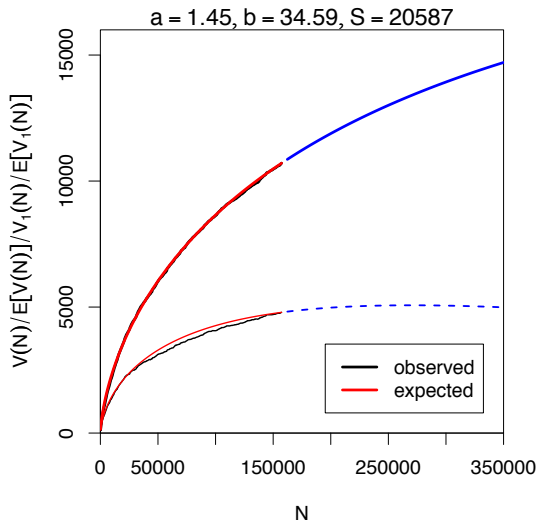
example from Evert and Lüdeling (2001)





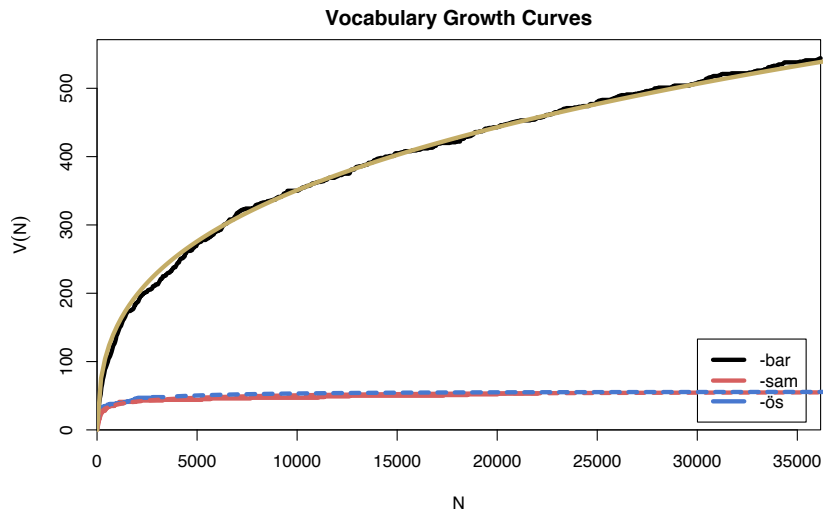
# Measuring morphological productivity

example from Evert and Lüdeling (2001)



# Measuring morphological productivity

example from Evert and Lüdeling (2001)



# Quantitative measures of productivity

(Tweedie and Baayen 1998; Baayen 2001)

- ▶ Baayen's (1991) productivity index  $\mathcal{P}$   
(slope of vocabulary growth curve)

$$\mathcal{P} = \frac{V_1}{N}$$

- ▶ TTR = type-token ratio

$$\text{TTR} = \frac{V}{N}$$

- ▶ Zipf-Mandelbrot slope

$$a$$

- ▶ Herdan's law (1964)

$$C = \frac{\log V}{\log N}$$

# Quantitative measures of productivity

(Tweedie and Baayen 1998; Baayen 2001)

- ▶ Baayen's (1991) productivity index  $\mathcal{P}$  (slope of vocabulary growth curve)

$$\mathcal{P} = \frac{V_1}{N}$$

- ▶ TTR = type-token ratio

$$\text{TTR} = \frac{V}{N}$$

- ▶ Zipf-Mandelbrot slope

$$a$$

- ▶ Herdan's law (1964)

$$C = \frac{\log V}{\log N}$$

- ▶ Yule (1944) / Simpson (1949)

$$K = 10\,000 \cdot \frac{\sum_m m^2 V_m - N}{N^2}$$

- ▶ Guiraud (1954)

$$R = \frac{V}{\sqrt{N}}$$

- ▶ Sichel (1975)

$$S = \frac{V_2}{V}$$

- ▶ Honoré (1979)

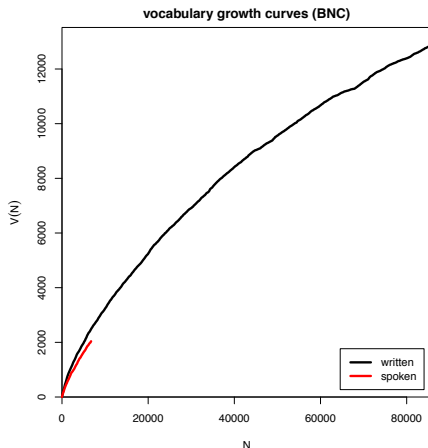
$$H = \frac{\log N}{1 - \frac{V_1}{V}}$$

# Productivity measures for bare singulars in the BNC

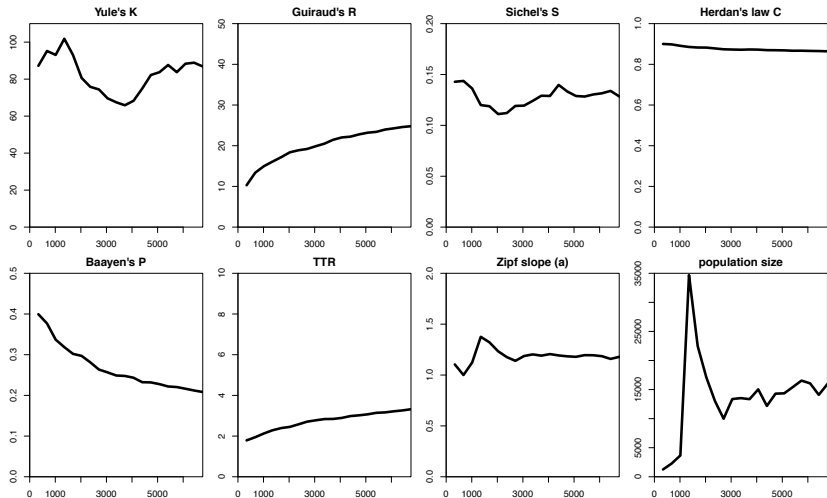
	spoken	written
<i>V</i>	2,039	12,876
<i>N</i>	6,766	85,750
<i>K</i>	86.84	28.57
<i>R</i>	24.79	43.97
<i>S</i>	0.13	0.15
<i>C</i>	0.86	0.83
<i>P</i>	0.21	0.08
TTR	0.301	0.150
<i>a</i>	1.18	1.27
pop. <i>S</i>	15,958	36,874

# Productivity measures for bare singulars in the BNC

	spoken	written
$V$	2,039	12,876
$N$	6,766	85,750
$K$	86.84	28.57
$R$	24.79	43.97
$S$	0.13	0.15
$C$	0.86	0.83
$\mathcal{P}$	0.21	0.08
TTR	0.301	0.150
$a$	1.18	1.27
pop. $S$	15,958	36,874



# Are these “lexical constants” really constant?



# Outline

## Introduction

Motivation

Notation & basic concepts

Zipf's law

First steps (zipfR)

## LNRE models

Population & samples

The mathematics of LNRE

## Applications & examples

Productivity &

lexical diversity

**Practical LNRE modelling**

Bootstrapping experiments

LNRE as Bayesian prior

## Challenges

Model inference

Zipf's law

Non-randomness

Significance testing

Outlook



# interactive demo

# Outline

## Introduction

Motivation

Notation & basic concepts

Zipf's law

First steps (zipfR)

## LNRE models

Population & samples

The mathematics of LNRE

## Applications & examples

Productivity &

lexical diversity

Practical LNRE modelling

Bootstrapping experiments

LNRE as Bayesian prior

## Challenges

Model inference

Zipf's law

Non-randomness

**Significance testing**

Outlook

# Case study: Iris Murdoch & early symptoms of AD

(Evert *et al.* 2017)

- ▶ Renowned British author (1919–1999)
- ▶ Published a total of 26 novels, mostly well received by critics
- ▶ Murdoch experienced unexpected difficulties composing her last novel, received “without enthusiasm” (Garrard *et al.* 2005)
- ▶ Diagnosis of Alzheimer’s disease shortly after publication

## Murdoch novel reveals Alzheimer's

**The last novel by the author Iris Murdoch reveals the first signs of Alzheimer's disease, experts say.**

A team from University College London say their examination of works from throughout Dame Iris's career could be used to help diagnose others.

They found the structure and grammar of her novels was relatively unchanged, but her language was noticeably simpler in her last novel, 'Jackson's Dilemma'.

The study is published online by the journal *Brain*.

<http://news.bbc.co.uk/2/hi/health/4058605.stm>



Experts analysed three of Dame Iris's books

# Case study: Iris Murdoch & early symptoms of AD

(Evert *et al.* 2017)

- ▶ Renowned British author (1919–1999)
- ▶ Published a total of 26 novels, mostly well received by critics
- ▶ Murdoch experienced unexpected difficulties composing her last novel, received “without enthusiasm” (Garrard *et al.* 2005)
- ▶ Diagnosis of Alzheimer’s disease shortly after publication

## Conflicting results:

- ▶ Decline of lexical diversity in last novel (Garrard *et al.* 2005; Pakhomov *et al.* 2011)
- ▶ No clear effects found (Le *et al.* 2011)

### Murdoch novel reveals Alzheimer's

**The last novel by the author Iris Murdoch reveals the first signs of Alzheimer's disease, experts say.**

A team from University College London say their examination of works from throughout Dame Iris's career could be used to help diagnose others.

They found the structure and grammar of her novels was relatively unchanged, but her language was noticeably simpler in her last novel, 'Jackson's Dilemma'.

The study is published online by the journal Brain.

<http://news.bbc.co.uk/2/hi/health/4058605.stm>



Experts analysed three of Dame Iris's books

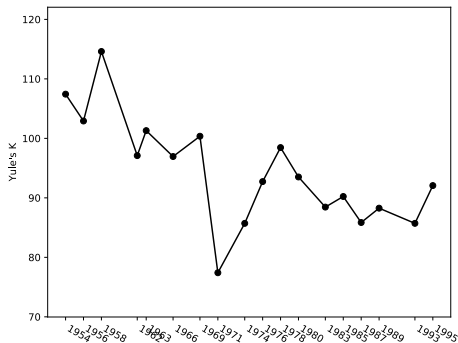
# Case study: Iris Murdoch & early symptoms of AD

(Evert *et al.* 2017)

- ▶ Corpus data
  - ▶ 19 out of 26 novels written by Iris Murdoch
  - ▶ including 9 last novels, spanning a period of almost 20 years
  - ▶ acquired as e-books (no errors due to OCR)
- ▶ Pre-processing and annotation
  - ▶ Stanford CoreNLP (Manning *et al.* 2014) for tokenization, sentence splitting, POS tagging, and syntactic parsing
  - ▶ exclude dialogue based on typographic quotation marks (following Garrard *et al.* 2005; Pakhomov *et al.* 2011)
- ▶ The challenge
  - 👉 assess significance of differences in productivity for single texts
  - 👉 might explain conflicting results in prior work

# Measures of vocabulary diversity = productivity

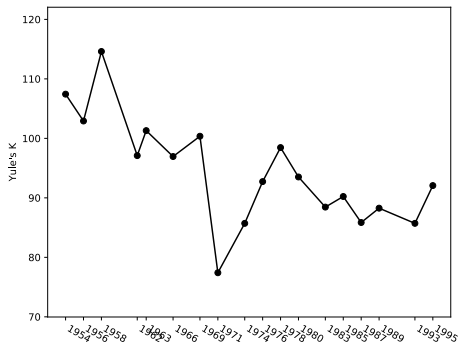
(Evert *et al.* 2017)



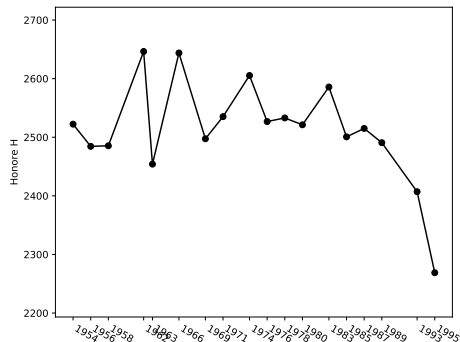
Yule's  $\kappa$

# Measures of vocabulary diversity = productivity

(Evert *et al.* 2017)



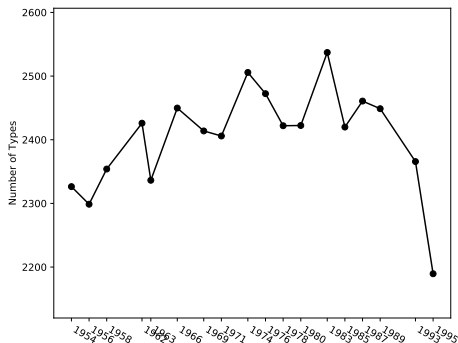
Yule's  $\kappa$



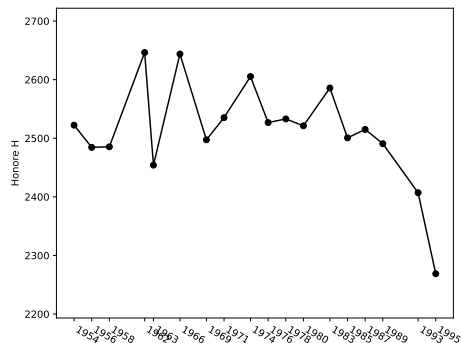
Honoré  $H$

# Measures of vocabulary diversity = productivity

(Evert *et al.* 2017)



type count / TTR



Honoré H



# Cross-validation for productivity measures

(Evert *et al.* 2017)

As a first step:

- ▶ Partition each novel into folds of 10,000 consecutive tokens
- ➡  $k \geq 6$  folds for each novel (leftover tokens discarded)

# Cross-validation for productivity measures

(Evert *et al.* 2017)

As a first step:

- ▶ Partition each novel into folds of 10,000 consecutive tokens
- ➡  $k \geq 6$  folds for each novel (leftover tokens discarded)

Then:

- ▶ Evaluate complexity measure of interest on each fold

$$y_1, \dots, y_k$$

# Cross-validation for productivity measures

(Evert *et al.* 2017)

As a first step:

- ▶ Partition each novel into folds of 10,000 consecutive tokens
- ➡  $k \geq 6$  folds for each novel (leftover tokens discarded)

Then:

- ▶ Evaluate complexity measure of interest on each fold

$$y_1, \dots, y_k$$

- ▶ Compute macro-average as overall measure for the entire text

$$\bar{y} = \frac{y_1 + \dots + y_k}{k}$$

- ▶ Instead of value  $x$  obtained by evaluating measure on full text

# Cross-validation for productivity measures

(Evert *et al.* 2017)

## Significance testing procedure:

- ▶ Standard deviation  $\sigma$  of individual folds estimated from data

$$\sigma^2 \approx s^2 = \frac{1}{k-1} \sum_{i=1}^k (y_i - \bar{y})^2$$

# Cross-validation for productivity measures

(Evert *et al.* 2017)

## Significance testing procedure:

- ▶ Standard deviation  $\sigma$  of individual folds estimated from data

$$\sigma^2 \approx s^2 = \frac{1}{k-1} \sum_{i=1}^k (y_i - \bar{y})^2$$

- ▶ Standard deviation of macro average can be computed as

$$\sigma_{\bar{y}} = \frac{\sigma}{\sqrt{k}} \approx \frac{s}{\sqrt{k}}$$

# Cross-validation for productivity measures

(Evert *et al.* 2017)

## Significance testing procedure:

- ▶ Standard deviation  $\sigma$  of individual folds estimated from data

$$\sigma^2 \approx s^2 = \frac{1}{k-1} \sum_{i=1}^k (y_i - \bar{y})^2$$

- ▶ Standard deviation of macro average can be computed as

$$\sigma_{\bar{y}} = \frac{\sigma}{\sqrt{k}} \approx \frac{s}{\sqrt{k}}$$

- ▶ Asymptotic 95% confidence intervals are then given by

$$\bar{y} \pm 1.96 \cdot \sigma_{\bar{y}}$$

# Cross-validation for productivity measures

(Evert *et al.* 2017)

## Significance testing procedure:

- ▶ Standard deviation  $\sigma$  of individual folds estimated from data

$$\sigma^2 \approx s^2 = \frac{1}{k-1} \sum_{i=1}^k (y_i - \bar{y})^2$$

- ▶ Standard deviation of macro average can be computed as

$$\sigma_{\bar{y}} = \frac{\sigma}{\sqrt{k}} \approx \frac{s}{\sqrt{k}}$$

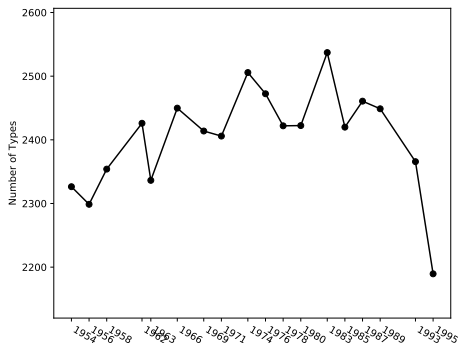
- ▶ Asymptotic 95% confidence intervals are then given by

$$\bar{y} \pm 1.96 \cdot \sigma_{\bar{y}}$$

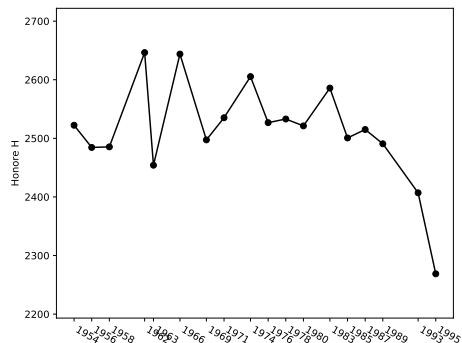
- ▶ Comparison of samples with Student's  $t$ -test, based on pooled cross-validation folds (feasible even for  $n_1 = 1$ )

# Productivity measures with confidence intervals

(Evert *et al.* 2017)



type count / TTR

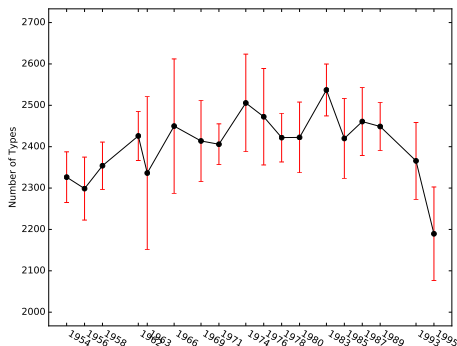


Honoré H

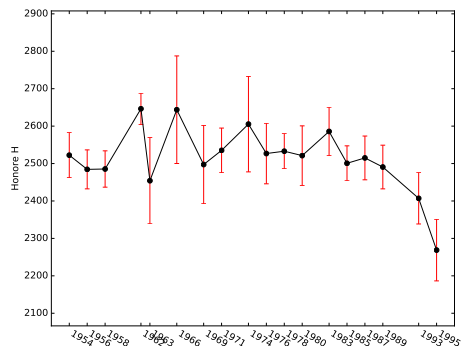


# Productivity measures with confidence intervals

(Evert *et al.* 2017)



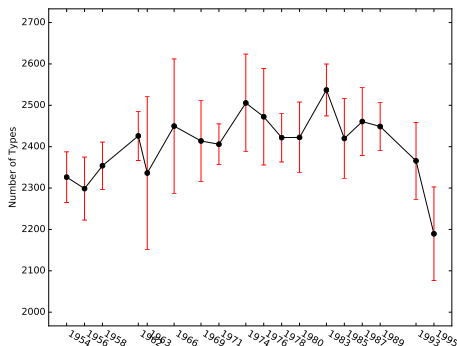
type count / TTR



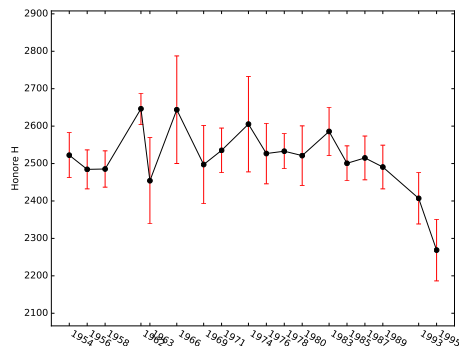
Honore H

# Productivity measures with confidence intervals

(Evert *et al.* 2017)



type count / TTR

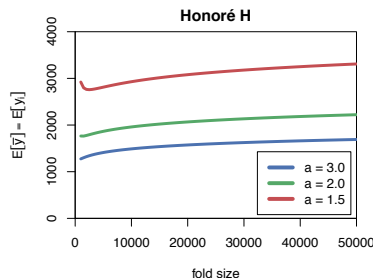
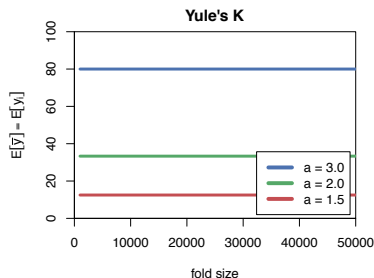
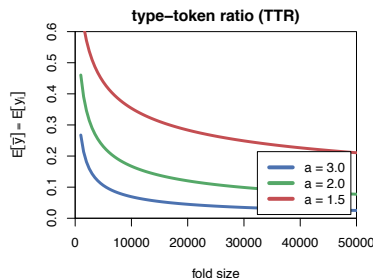
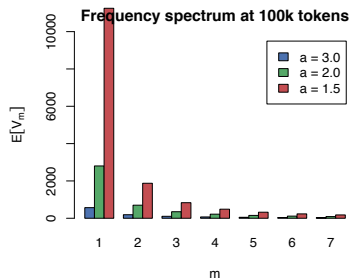


Honoré *H*

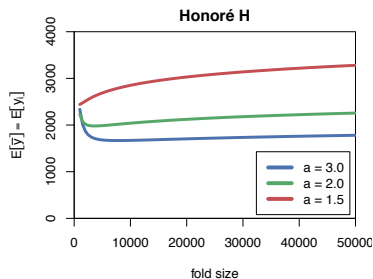
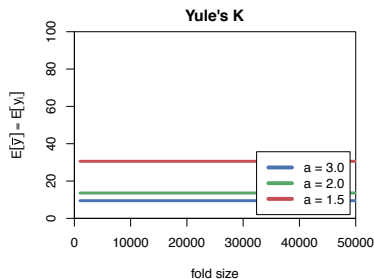
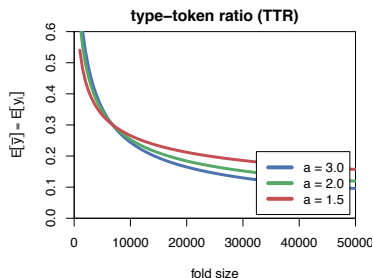
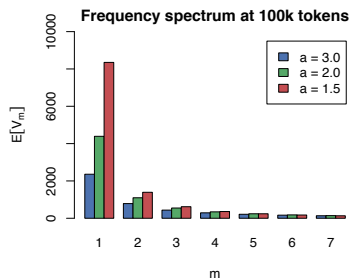
significance test vs. first 17 novels

$t = -6.1$ ,  $df=5.52$ ,  $p = .0012^{**}$

# Cross-validated measures depend on fold size!

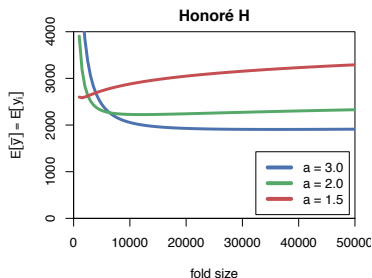
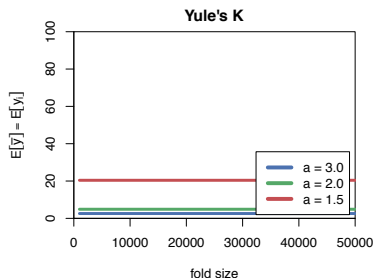
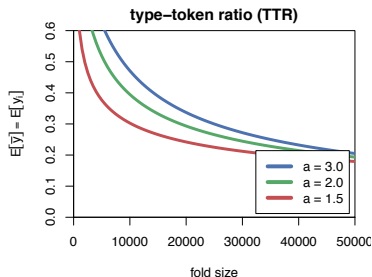
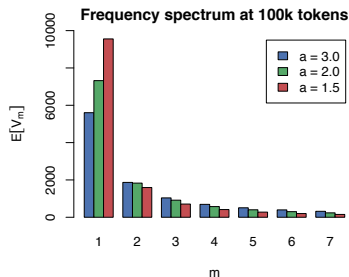


# Cross-validated measures depend on fold size!



B

# Cross-validated measures depend on fold size!



# Outline

## Introduction

- Motivation

- Notation & basic concepts

- Zipf's law

- First steps (zipfR)

## LNRE models

- Population & samples

- The mathematics of LNRE

## Applications & examples

- Productivity &

- lexical diversity

- Practical LNRE modelling

- Bootstrapping experiments

- LNRE as Bayesian prior

## Challenges

- Model inference

- Zipf's law

- Non-randomness

- Significance testing

- Outlook

# Research programme for LNRE models

- ▶ Improve efficiency & numerical accuracy of implementation
  - ▶ numerical integrals instead of differences of Gamma functions
  - ▶ better parameter estimation (gradient, aggregated spectrum)
- ▶ Analyze accuracy of LNRE approximations
  - ▶ comprehensive simulation experiments, esp. for small samples
- ▶ Specify more flexible LNRE population models
  - ▶ my favourite: piecewise Zipfian type density functions
  - ▶ Baayen (2001): mixture distributions (different parameters)
- ▶ Develop hypothesis tests & confidence intervals
  - ▶ key challenge: goodness-of-fit *vs.* confidence region
  - ▶ prediction intervals for model-based extrapolation
- ▶ Simulation experiments for productivity measures
  - ▶ Can we find a quantitative measure that is robust against confounding factors and corresponds to intuitive notions of productivity & lexical diversity?

# Research programme for LNRE models

- ▶ Is non-randomness a problem?
  - ▶ not for morphological productivity → ECHO correction
  - ▶ tricky to include explicitly in LNRE approach
- ▶ Do we need LNRE models for practical applications?
  - ▶ better productivity measures + empirical sampling variation
  - ▶ based on cross-validation approach (Evert *et al.* 2017)
- ▶ How important is semantics & context?
  - ▶ Does it make sense to measure productivity and lexical diversity purely in terms of type-token distributions?
  - ▶ e.g. register variation for morphological productivity
  - ▶ e.g. semantic preferences in productive slots of construction
  - ▶ type-token ratio  $\neq$  complexity of author's vocabulary



Thank you!

# References I

- Baayen, Harald (1991). A stochastic process for word frequency distributions. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, pages 271–278.
- Baayen, R. Harald (2001). *Word Frequency Distributions*. Kluwer Academic Publishers, Dordrecht.
- Baroni, Marco and Evert, Stefan (2005). Testing the extrapolation quality of word frequency models. In P. Danielsson and M. Wagenmakers (eds.), *Proceedings of Corpus Linguistics 2005*, volume 1, no. 1 of *Proceedings from the Corpus Linguistics Conference Series*, Birmingham, UK. ISSN 1747-9398.
- Baroni, Marco and Evert, Stefan (2007). Words and echoes: Assessing and mitigating the non-randomness problem in word frequency distribution modeling. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 904–911, Prague, Czech Republic.
- Brainerd, Barron (1982). On the relation between the type-token and species-area problems. *Journal of Applied Probability*, **19**(4), 785–793.
- Cao, Yong; Xiong, Fei; Zhao, Youjie; Sun, Yongke; Yue, Xiaoguang; He, Xin; Wang, Lichao (2017). Pow law in random symbolic sequences. *Digital Scholarship in the Humanities*, **32**(4), 733–738.

# References II

- Church, Kenneth W. (2000). Empirical estimates of adaptation: The chance of two Noriegas is closer to  $p/2$  than  $p^2$ . In *Proceedings of COLING 2000*, pages 173–179, Saarbrücken, Germany.
- Efron, Bradley (1979). Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1), 1–26.
- Evert, Stefan (2004). A simple LNRE model for random character sequences. In *Proceedings of the 7èmes Journées Internationales d'Analyse Statistique des Données Textuelles (JADT 2004)*, pages 411–422, Louvain-la-Neuve, Belgium.
- Evert, Stefan and Baroni, Marco (2007). *zipfR*: Word frequency distributions in R. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, Posters and Demonstrations Sessions*, pages 29–32, Prague, Czech Republic.
- Evert, Stefan and Lüdeling, Anke (2001). Measuring morphological productivity: Is automatic preprocessing sufficient? In P. Rayson, A. Wilson, T. McEnery, A. Hardie, and S. Khoja (eds.), *Proceedings of the Corpus Linguistics 2001 Conference*, pages 167–175, Lancaster. UCREL.
- Evert, Stefan; Wankerl, Sebastian; Nöth, Elmar (2017). Reliable measures of syntactic and lexical complexity: The case of Iris Murdoch. In *Proceedings of the Corpus Linguistics 2017 Conference*, Birmingham, UK.

# References III

- Garrard, Peter; Maloney, Lisa M.; Hodges, John R.; Patterson, Karalyn (2005). The effects of very early Alzheimer's disease on the characteristics of writing by a renowned author. *Brain*, **128**(2), 250–260.
- Grieve, Jack; Clarke, Isobelle; Chiang, Emily; Gideon, Hannah; Heini, Annina; Nini, Andrea; Waibel, Emily (2018). Attributing the Bixby Letter using n-gram tracing. *Digital Scholarship in the Humanities*. doi:10.1093/llc/fqy042.
- Herdan, Gustav (1964). *Quantitative Linguistics*. Butterworths, London.
- Kornai, András (1999). Zipf's law outside the middle range. In *Proceedings of the Sixth Meeting on Mathematics of Language*, pages 347–356, University of Central Florida.
- Le, Xuan; Lancashire, Ian; Hirst, Graeme; Jokel, Regina (2011). Longitudinal detection of dementia through lexical and syntactic changes in writing: a case study of three British novelists. *Literary and Linguistic Computing*, **26**(4), 435–461.
- Li, Wentian (1992). Random texts exhibit zipf's-law-like word frequency distribution. *IEEE Transactions on Information Theory*, **38**(6), 1842–1845.
- Mandelbrot, Benoît (1953). An informational theory of the statistical structure of languages. In W. Jackson (ed.), *Communication Theory*, pages 486–502. Butterworth, London.

## References IV

- Mandelbrot, Benoît (1962). On the theory of word frequencies and on related Markovian models of discourse. In R. Jakobson (ed.), *Structure of Language and its Mathematical Aspects*, pages 190–219. American Mathematical Society, Providence, RI.
- Manning, Christopher D.; Surdeanu, Mihai; Bauer, John; Finkel, Jenny; Bethard, Steven J.; McClosky, David (2014). The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014): System Demonstrations*, pages 55–60, Baltimore, MD.
- Miller, George A. (1957). Some effects of intermittent silence. *The American Journal of Psychology*, **52**, 311–314.
- Montemurro, Marcelo A. (2001). Beyond the Zipf-Mandelbrot law in quantitative linguistics. *Physica A*, **300**, 567–578.
- Pakhomov, Serguei; Chacon, Dustin; Wicklund, Mark; Gundel, Jeanette (2011). Computerized assessment of syntactic complexity in Alzheimer's disease: A case study of Iris Murdoch's writing. *Behavior Research Methods*, **43**(1), 136–144.
- Rouault, Alain (1978). Lois de Zipf et sources markoviennes. *Annales de l'Institut H. Poincaré (B)*, **14**, 169–188.

# References V

- Sichel, H. S. (1971). On a family of discrete distributions particularly suited to represent long-tailed frequency data. In N. F. Laubscher (ed.), *Proceedings of the Third Symposium on Mathematical Statistics*, pages 51–97, Pretoria, South Africa. C.S.I.R.
- Sichel, H. S. (1975). On a distribution law for word frequencies. *Journal of the American Statistical Association*, **70**, 542–547.
- Simon, Herbert A. (1955). On a class of skew distribution functions. *Biometrika*, **47**(3/4), 425–440.
- Tweedie, Fiona J. and Baayen, R. Harald (1998). How variable may a constant be? measures of lexical richness in perspective. *Computers and the Humanities*, **32**(5), 323–352.
- Yule, G. Udny (1944). *The Statistical Study of Literary Vocabulary*. Cambridge University Press, Cambridge.
- Zipf, George Kingsley (1949). *Human Behavior and the Principle of Least Effort*. Addison-Wesley, Cambridge, MA.
- Zipf, George Kingsley (1965). *The Psycho-biology of Language*. MIT Press, Cambridge, MA.