Unit 5: Word Frequency Distributions with the zipfR package

Statistics for Linguists with R – A SIGIL Course

Designed by Marco Baroni[1] and Stefan Evert[2]

[1]Center for Mind/Brain Sciences (CIMeC)
University of Trento, Italy

[2]Corpus Linguistics Group
Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

http://SIGIL.r-forge.r-project.org/

Outline

Lexical statistics

Zipf (1949, 1965); Baayen (2001); Baroni (2008)

- Statistical study of the frequency distribution of **types** (words or other linguistic units) in texts
  - remember the distinction between **types** and **tokens**?
- Different from other categorical data because of the extremely large number of distinct types
  - people often speak of **Zipf's law** in this context
- Key applications: **productivity** and **vocabulary richness**
  - prevalence of low-frequency types
  - vocabulary growth for incremental samples

Outline

Basic terminology

- N: sample / corpus size, number of **tokens** in the sample
- V: **vocabulary** size, number of distinct **types** in the sample
- V_m: **spectrum element** m, number of types in the sample with frequency m (i.e. exactly m occurrences)
- V_1: number of **hapax legomena**, types that occur only once in the sample (for hapaxes, #types = #tokens)

- A sample: `a b b c a a b a`
- $N = 8$, $V = 3$, $V_1 = 1$

Rank / frequency profile

- The sample: `c a a b c c a c d`
- Frequency list ordered by decreasing frequency

t	f
c	4
a	3
b	1
d	1

Rank / frequency profile

- The sample: `c a a b c c a c d`
- Frequency list ordered by decreasing frequency

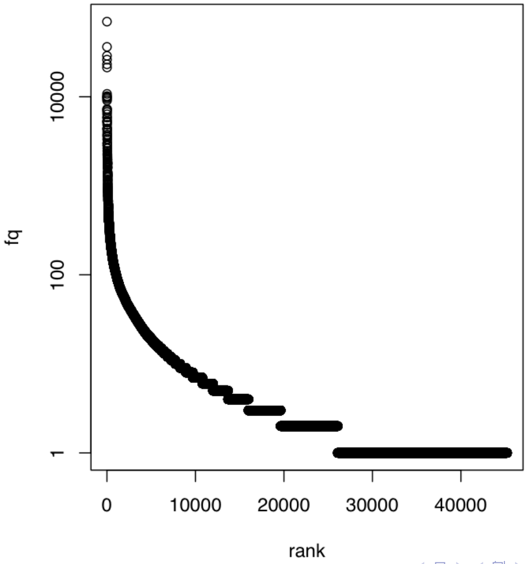
t	f
c	4
a	3
b	1
d	1

- Rank / frequency profile: ranks instead of type labels

r	f
1	4
2	3
3	1
4	1

- Expresses type frequency f_r as function of rank r of a type

Rank/frequency profile of Brown corpus



Top and bottom ranks in the Brown corpus

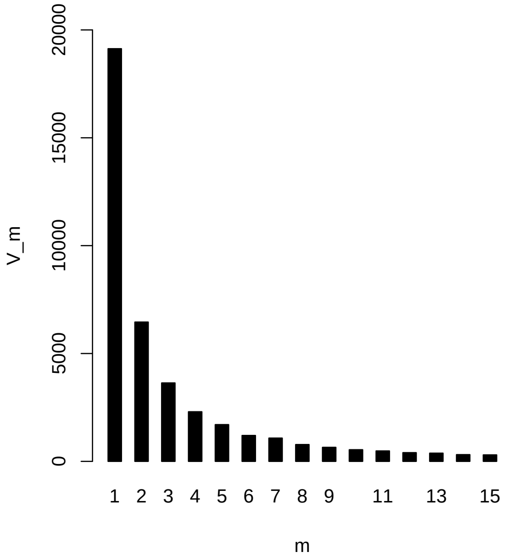
top frequencies			bottom frequencies		
r	*f*	word	rank range	*f*	randomly selected examples
1	69836	the	7731 – 8271	10	schedules, polynomials, bleak
2	36365	of	8272 – 8922	9	tolerance, shaved, hymn
3	28826	and	8923 – 9703	8	decreased, abolish, irresistible
4	26126	to	9704 – 10783	7	immunity, cruising, 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

Frequency spectrum

- The sample: `c a a b c c a c d`
- Frequency classes: 1 (`b`, `d`), 3 (`a`), 4 (`c`)
- Frequency spectrum:

m	V_m
1	2
3	1
4	1

Frequency spectrum of Brown corpus



Vocabulary growth curve

- The sample: `a b b c a a b a`

Vocabulary growth curve

- The sample: `a b b c a a b a`
- $N = 1$, $V = 1$, $V_1 = 1$ $\quad (V_2 = 0, \ldots)$

Vocabulary growth curve

- The sample: `a b b c a a b a`
- $N = 1,\ V = 1,\ V_1 = 1 \quad (V_2 = 0, \ldots)$
- $N = 3,\ V = 2,\ V_1 = 1 \quad (V_2 = 1,\ V_3 = 0, \ldots)$

Vocabulary growth curve

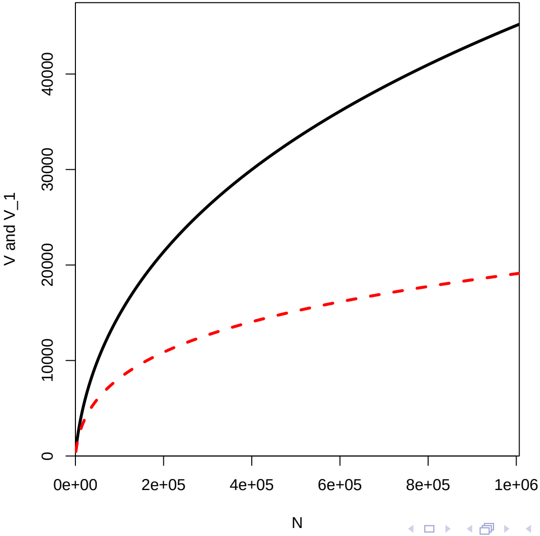
- The sample: `a b b c a a b a`
- $N = 1,\ V = 1,\ V_1 = 1 \quad (V_2 = 0, \ldots)$
- $N = 3,\ V = 2,\ V_1 = 1 \quad (V_2 = 1,\ V_3 = 0, \ldots)$
- $N = 5,\ V = 3,\ V_1 = 1 \quad (V_2 = 2,\ V_3 = 0, \ldots)$

Vocabulary growth curve

- The sample: `a b b c a a b a`
- $N = 1,\ V = 1,\ V_1 = 1 \quad (V_2 = 0, \ldots)$
- $N = 3,\ V = 2,\ V_1 = 1 \quad (V_2 = 1,\ V_3 = 0, \ldots)$
- $N = 5,\ V = 3,\ V_1 = 1 \quad (V_2 = 2,\ V_3 = 0, \ldots)$
- $N = 8,\ V = 3,\ V_1 = 1 \quad (V_2 = 0,\ V_3 = 1,\ V_4 = 1, \ldots)$

Vocabulary growth curve of Brown corpus

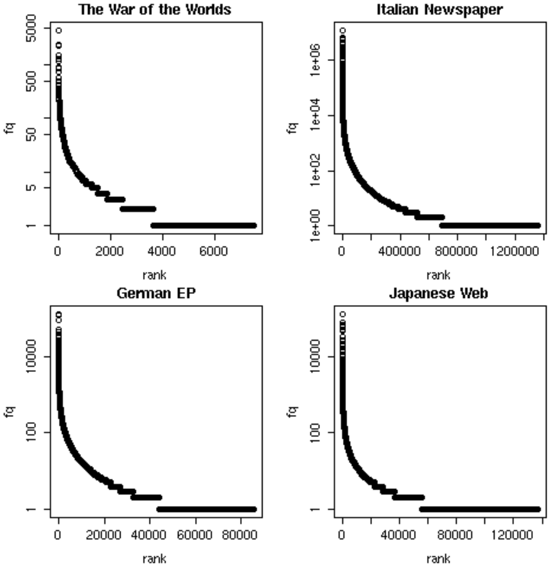
With V_1 growth in red (idealized curve smoothed by binomial interpolation)



Outline

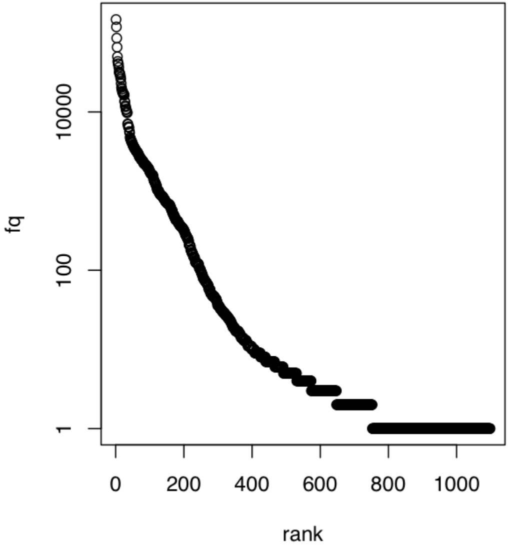
Typical frequency patterns

Across text types & languages



Typical frequency patterns

The Italian prefix *ri-* in the *la Repubblica* corpus

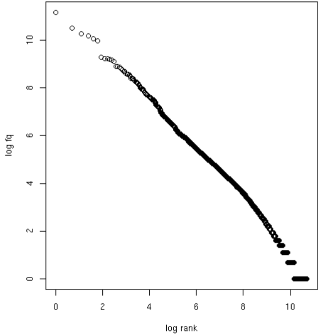


Is there a general law?

- Language after language, corpus after corpus, linguistic type after linguistic type, . . . we observe the same "few giants, many dwarves" pattern
- Similarity of plots suggests that relation between rank and frequency could be captured by a general law

Is there a general law?

- Language after language, corpus after corpus, linguistic type after linguistic type, ... we observe the same "few giants, many dwarves" pattern
- Similarity of plots suggests that relation between rank and frequency could be captured by a general law
- Nature of this relation becomes clearer if we plot $\log f$ as a function of $\log r$



Outline

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:

$$f(w) = \frac{C}{r(w)^a}$$

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:

$$f(w) = \frac{C}{r(w)^a}$$

- With $a = 1$ and $C =$60,000, Zipf's law predicts that:
  - most frequent word occurs 60,000 times
  - second most frequent word occurs 30,000 times
  - third most frequent word occurs 20,000 times
  - and there is a long tail of 80,000 words with frequencies between 1.5 and 0.5 occurrences(!)

Zipf's law

Logarithmic version

- Zipf's power law:

$$f(w) = \frac{C}{r(w)^a}$$

- If we take logarithm of both sides, we obtain:

$$\log f(w) = \log C - a \cdot \log r(w)$$

Zipf's law
Logarithmic version

- Zipf's power law:

$$f(w) = \frac{C}{r(w)^a}$$

- If we take logarithm of both sides, we obtain:

$$\underbrace{\log f(w)}_{y} = \log C - a \cdot \underbrace{\log r(w)}_{x}$$

- Zipf's law predicts that rank / frequency profiles are straight lines in double logarithmic space

Zipf's law

Logarithmic version

- Zipf's power law:

$$f(w) = \frac{C}{r(w)^a}$$

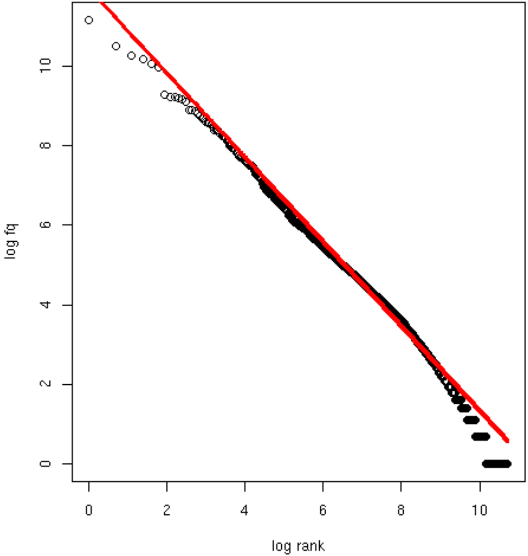
- If we take logarithm of both sides, we obtain:

$$\underbrace{\log f(w)}_{y} = \log C - a \cdot \underbrace{\log r(w)}_{x}$$

- Zipf's law predicts that rank / frequency profiles are straight lines in double logarithmic space
- Provides intuitive interpretation of a and C:
  - a is **slope** determining how fast log frequency decreases
  - $\log C$ is **intercept**, i.e., predicted log frequency of word with rank 1 (log rank 0) = most frequent word

Zipf's law

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



Zipf-Mandelbrot law

Mandelbrot (1953, 1962)

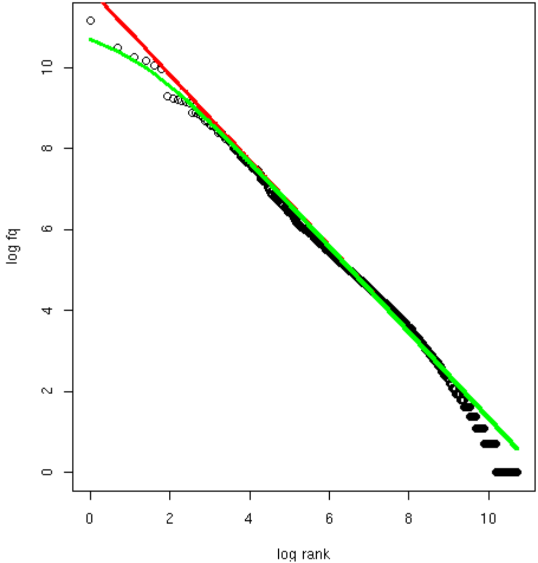
- Mandelbrot's extra parameter:

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

- 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 vs. Zipf's law

Non-linear least-squares fit (Brown corpus)



Outline

Applications of word frequency distributions

- Application 1: **extrapolation** of vocabulary size and frequency spectrum to larger sample sizes
  - morphological productivity (e.g. Lüdeling and Evert 2005)
  - lexical richness in stylometry (Efron and Thisted 1976), language acquisition, clinical linguistics (Garrard *et al.* 2005)
  - language technology (estimate proportion of OOV words, unseen grammar rules, typos, ...)

☞ need method for predicting vocab. growth on unseen data

Applications of word frequency distributions

- Application 1: **extrapolation** of vocabulary size and frequency spectrum to larger sample sizes
  - morphological productivity (e.g. Lüdeling and Evert 2005)
  - lexical richness in stylometry (Efron and Thisted 1976), language acquisition, clinical linguistics (Garrard *et al.* 2005)
  - language technology (estimate proportion of OOV words, unseen grammar rules, typos, ...)

☞ need method for predicting vocab. growth on unseen data

- Application 2: Zipfian **frequency distribution** across types
  - measures of lexical richness based on population ($\neq$ sample)
  - population model for Good-Turing smoothing (Good 1953; Gale and Sampson 1995)
  - realistic prior for Bayesian language modelling

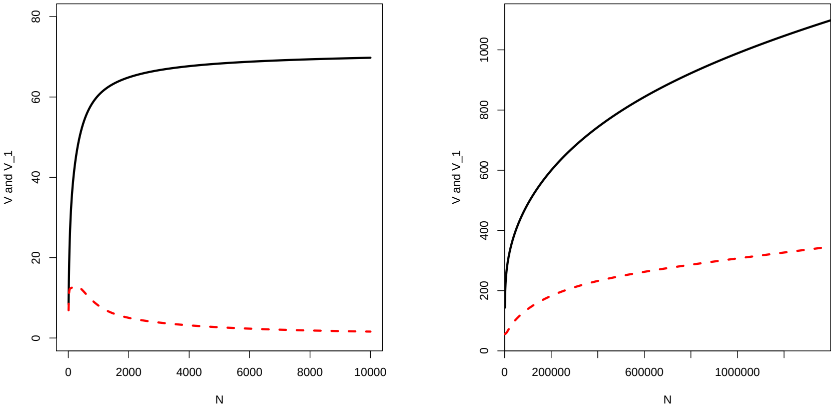
☞ need model of type probability distribution in the population

Vocabulary growth: Pronouns vs. *ri-* in Italian

N	V (pron.)	V (*ri-*)
5000	67	224
10000	69	271
15000	69	288
20000	70	300
25000	70	322
30000	71	347
35000	71	364
40000	71	377
45000	71	386
50000	71	400
...	...	...

Vocabulary growth: Pronouns vs. *ri-* in Italian

Vocabulary growth curves (V and V_1)



LNRE models for word frequency distributions

- LNRE = large number of rare events (cf. Baayen 2001)
- Statistics: corpus as random sample from **population**
  - population characterised by vocabulary of **types** w_k with occurrence **probabilities** π_k
  - not interested in specific types ➜ arrange by decreasing probability: $\pi_1 \geq \pi_2 \geq \pi_3 \geq \cdots$
  - NB: not necessarily identical to Zipf ranking in sample!

LNRE models for word frequency distributions

- LNRE = large number of rare events (cf. Baayen 2001)
- Statistics: corpus as random sample from **population**
  - population characterised by vocabulary of **types** w_k with occurrence **probabilities** π_k
  - not interested in specific types ➜ arrange by decreasing probability: $\pi_1 \geq \pi_2 \geq \pi_3 \geq \cdots$
  - NB: not necessarily identical to Zipf ranking in sample!

- **LNRE model** = population model for type probabilities, i.e. a function $k \mapsto \pi_k$ (with small number of parameters)
  - type probabilities π_k cannot be estimated reliably from a corpus, but parameters of LNRE model can

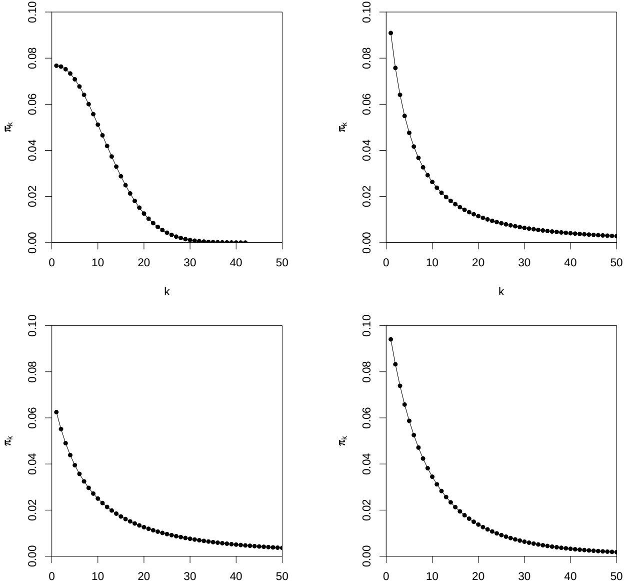
LNRE models for word frequency distributions

- LNRE = large number of rare events (cf. Baayen 2001)
- Statistics: corpus as random sample from **population**
  - population characterised by vocabulary of **types** w_k with occurrence **probabilities** π_k
  - not interested in specific types ➜ arrange by decreasing probability: $\pi_1 \geq \pi_2 \geq \pi_3 \geq \cdots$
  - NB: not necessarily identical to Zipf ranking in sample!

- **LNRE model** = population model for type probabilities, i.e. a function $k \mapsto \pi_k$ (with small number of parameters)
  - type probabilities π_k cannot be estimated reliably from a corpus, but parameters of LNRE model can

➥ **Parametric** statistical **model**

Examples of population models



The Zipf-Mandelbrot law as a population model

What is the right family of models for lexical frequency distributions?

- We have already seen that the Zipf-Mandelbrot law captures the distribution of observed frequencies very well

The Zipf-Mandelbrot law as a population model

What is the right family of models for lexical frequency distributions?

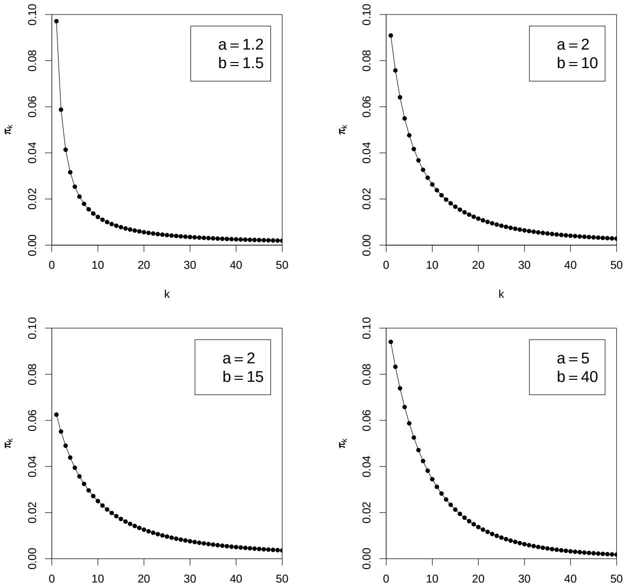
- We have already seen that the Zipf-Mandelbrot law captures the distribution of observed frequencies very well
- Re-phrase the law for type probabilities:

$$\pi_k := \frac{C}{(k+b)^a}$$

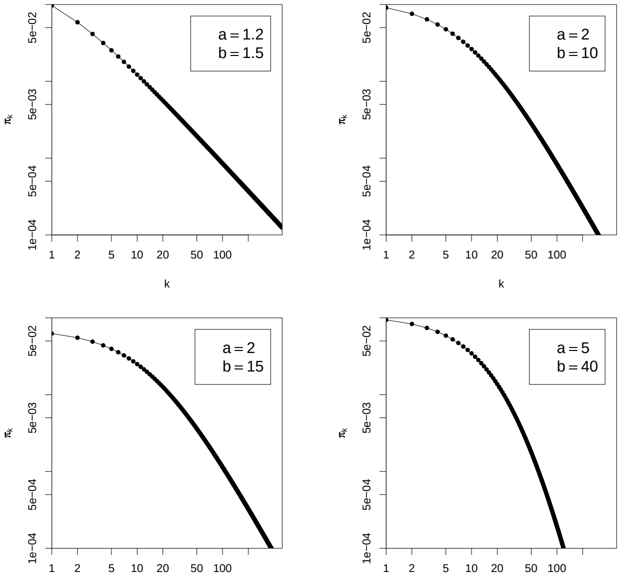
- Two free parameters: $a > 1$ and $b \geq 0$
- C is not a parameter but a normalization constant, needed to ensure that $\sum_k \pi_k = 1$
- This is the **Zipf-Mandelbrot** population model

Outline

The parameters of the Zipf-Mandelbrot model



The parameters of the Zipf-Mandelbrot model



The finite Zipf-Mandelbrot model

- Zipf-Mandelbrot population model characterizes an *infinite* type population: there is no upper bound on k, and the type probabilities π_k can become arbitrarily small
- $\pi = 10^{-6}$ (once every million words), $\pi = 10^{-9}$ (once every billion words), $\pi = 10^{-15}$ (once on the entire Internet), $\pi = 10^{-100}$ (once in the universe?)

The finite Zipf-Mandelbrot model

- Zipf-Mandelbrot population model characterizes an *infinite* type population: there is no upper bound on k, and the type probabilities π_k can become arbitrarily small
- $\pi = 10^{-6}$ (once every million words), $\pi = 10^{-9}$ (once every billion words), $\pi = 10^{-15}$ (once on the entire Internet), $\pi = 10^{-100}$ (once in the universe?)
- Alternative: finite (but often very large) number of types in the population
- We call this the **population vocabulary size** S (and write $S = \infty$ for an infinite type population)

The finite Zipf-Mandelbrot model

Evert (2004)

- The **finite Zipf-Mandelbrot** model simply stops after the first S types $(w_1, \dots, w_S)$
- S becomes a new parameter of the model
  $\rightarrow$ the finite Zipf-Mandelbrot model has 3 parameters

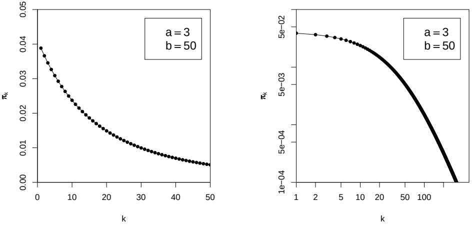
Abbreviations:

- **ZM** for Zipf-Mandelbrot model
- **fZM** for finite Zipf-Mandelbrot model

Outline

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 sample from this model:

- Draw N tokens from the population such that in each step, type w_k has probability π_k to be picked
- This allows us to make predictions for samples (= corpora) of arbitrary size N ➜ extrapolation

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 k
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
⋮	⋮	⋮

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

sample #1

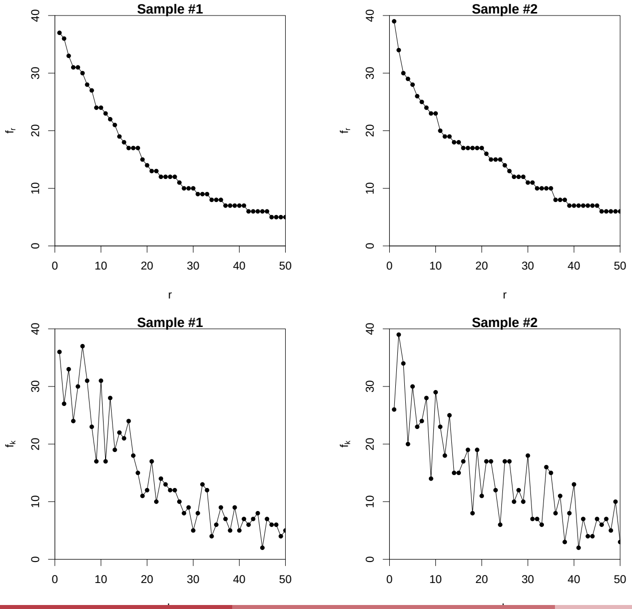
Samples: type frequency list & spectrum

rank r	f_r	type k
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
⋮	⋮	⋮

m	V_m
1	76
2	27
3	17
4	10
5	6
6	5
7	7
8	3
10	4
11	2
⋮	⋮

sample #2

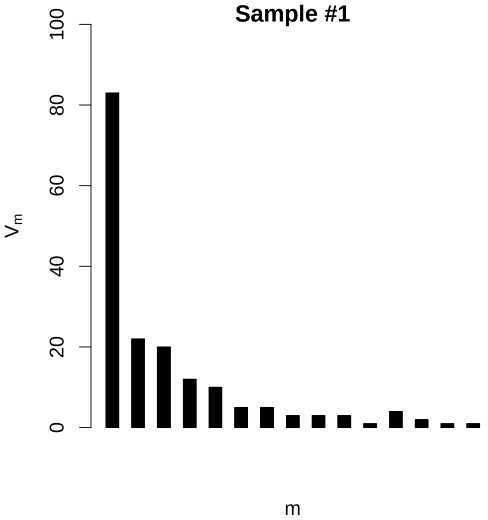
Random variation in type-frequency lists



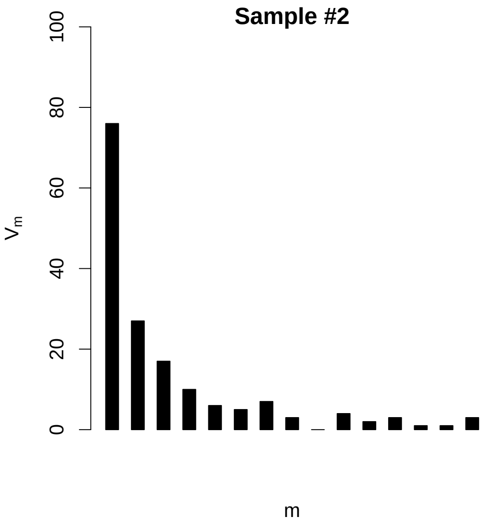
$r \leftrightarrow f_r$

$k \leftrightarrow f_k$

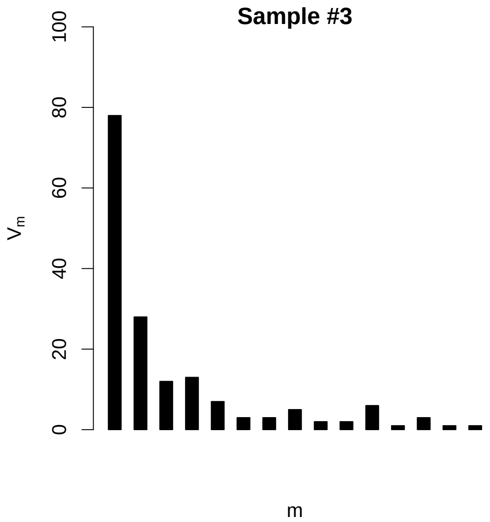
Random variation: frequency spectrum



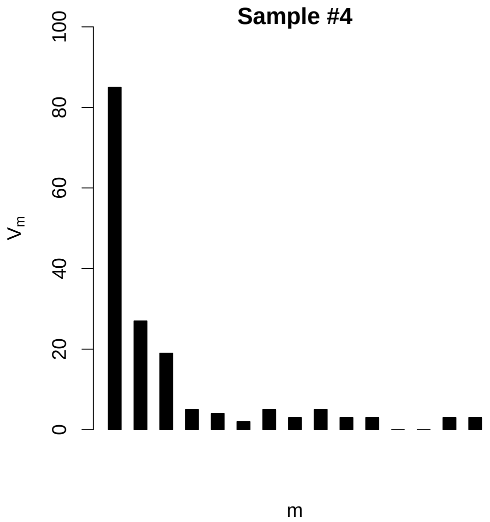
Random variation: frequency spectrum



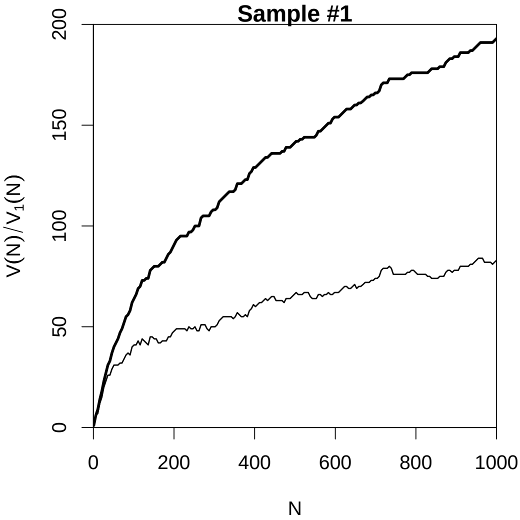
Random variation: frequency spectrum



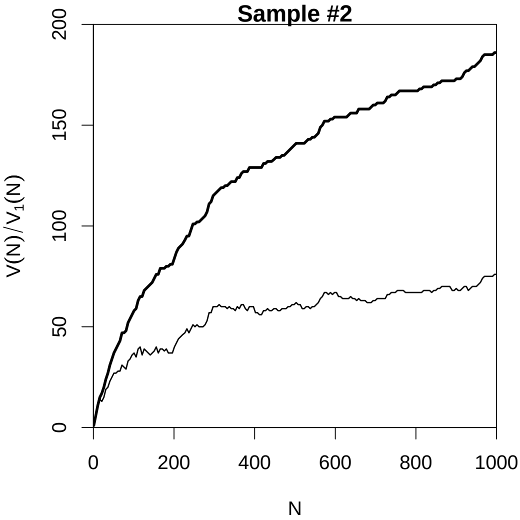
Random variation: frequency spectrum



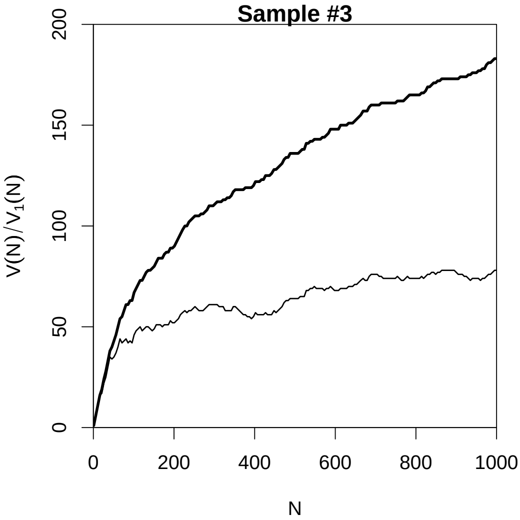
Random variation: vocabulary growth curve



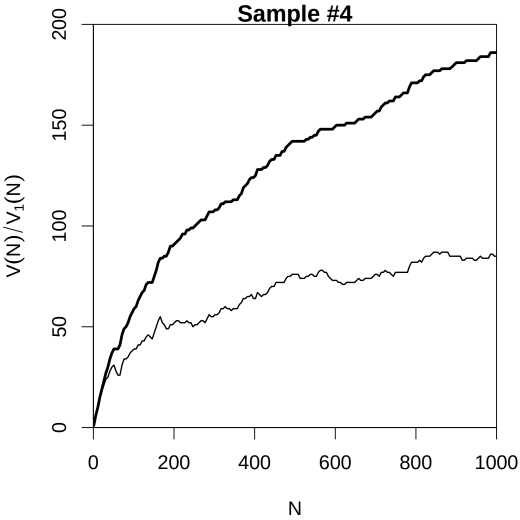
Random variation: vocabulary growth curve



Random variation: vocabulary growth curve



Random variation: vocabulary growth curve



Outline

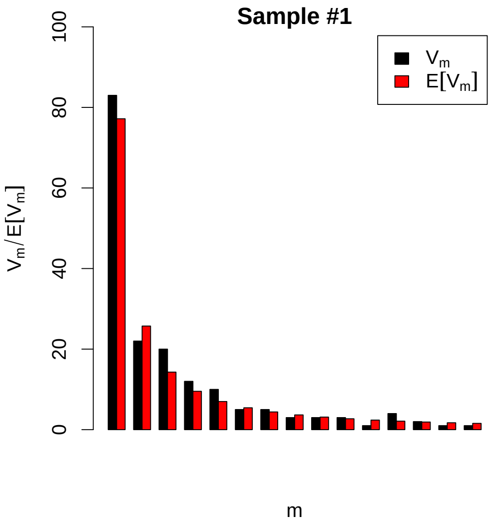
Expected values

- There is no reason why we should choose a particular sample to make a prediction for the real data – each one is equally likely or unlikely
- Take the average over a large number of samples, called **expected value** or **expectation** in statistics

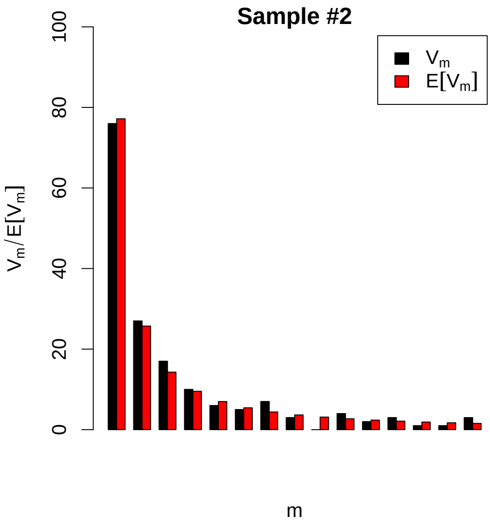
- Notation: $\mathrm{E}\left[V(N)\right]$ and $\mathrm{E}\left[V_m(N)\right]$
  - indicates that we are referring to expected values for a sample of size N
  - rather than to the specific values V and V_m observed in a particular sample or a real-world data set

- Expected values can be calculated efficiently *without* generating thousands of random samples

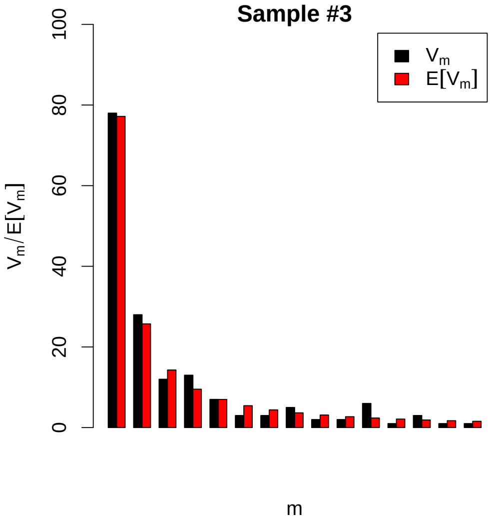
The expected frequency spectrum



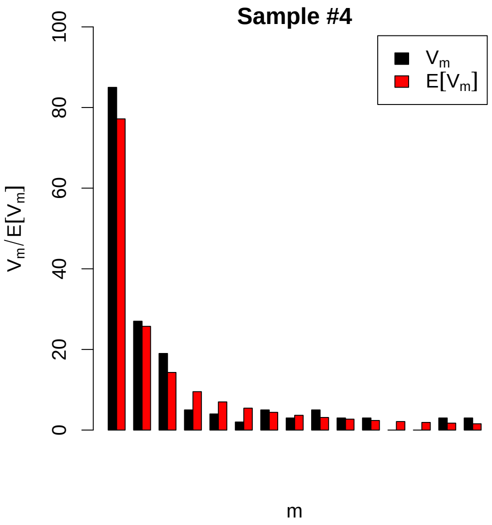
The expected frequency spectrum



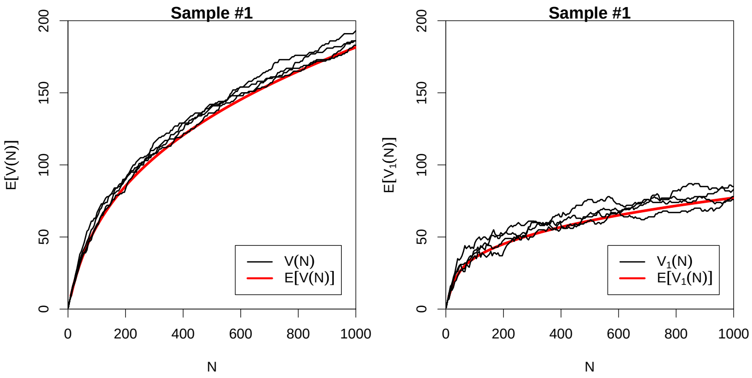
The expected frequency spectrum



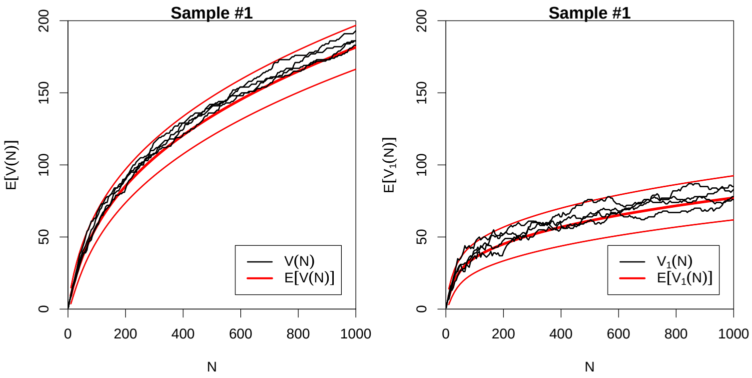
The expected frequency spectrum



The expected vocabulary growth curve



Prediction intervals for the expected VGC

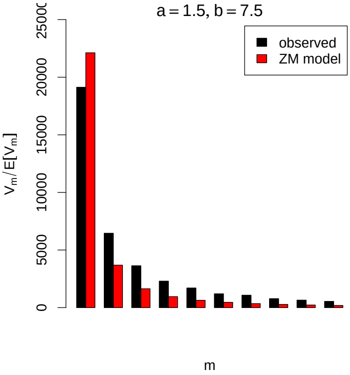


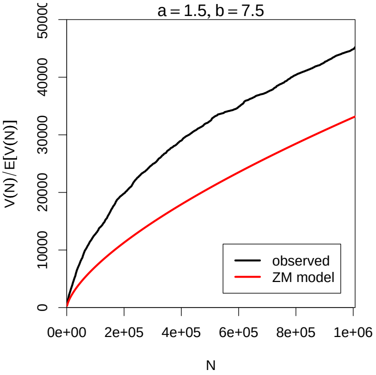
"Confidence intervals" that indicate predicted sampling distribution:

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

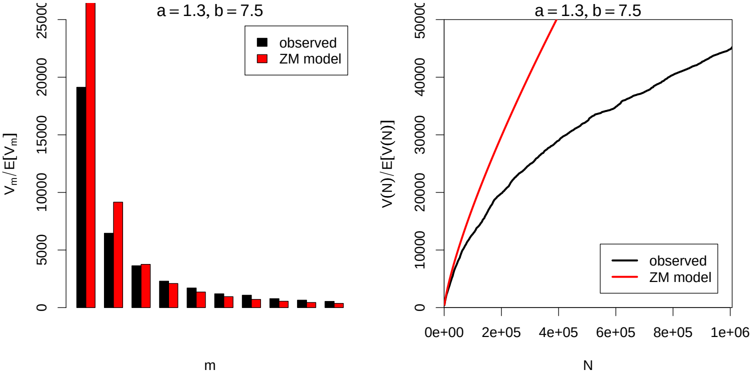
Outline

Parameter estimation by trial & error

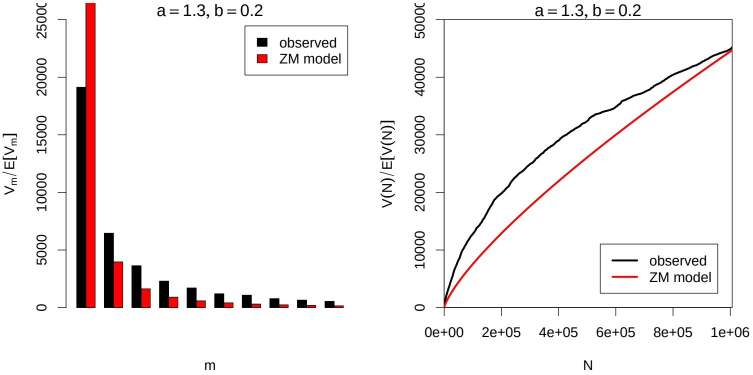




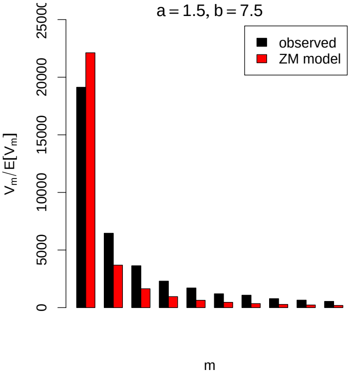
Parameter estimation by trial & error

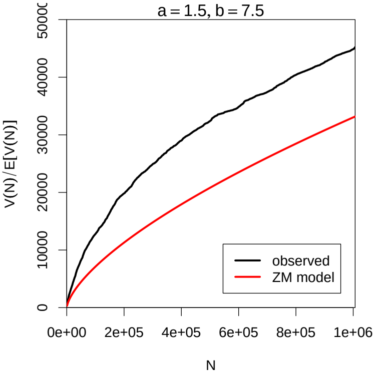


Parameter estimation by trial & error

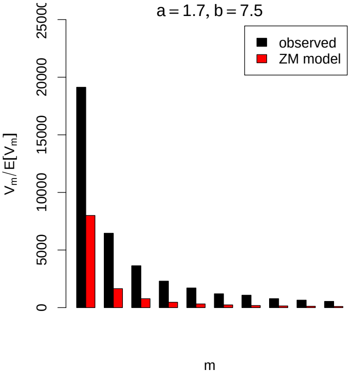


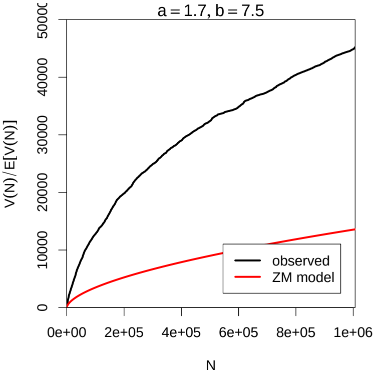
Parameter estimation by trial & error



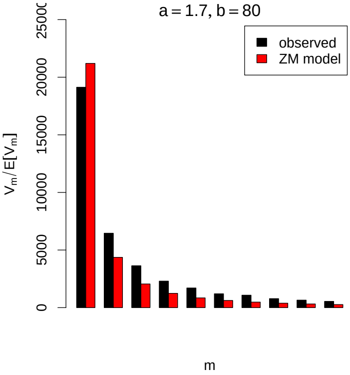


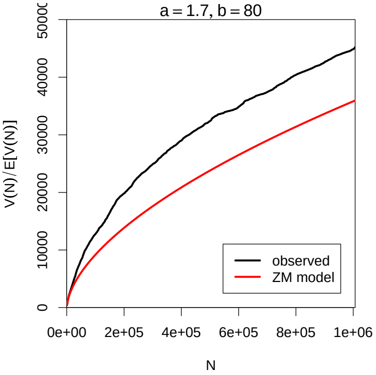
Parameter estimation by trial & error



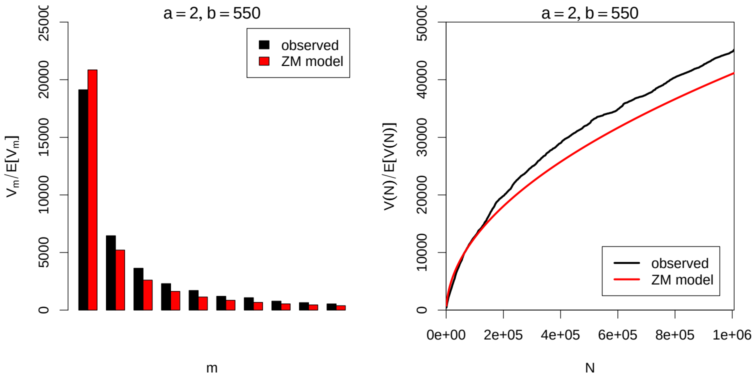


Parameter estimation by trial & error

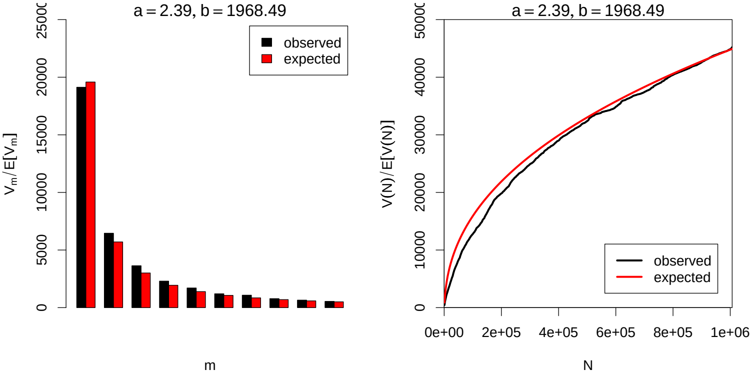




Parameter estimation by trial & error



Automatic parameter estimation



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

Outline

Goodness-of-fit

- Goodness-of-fit statistics measure how well the model has been fitted to the observed training data
- Compare observed vs. expected frequency distribution
  - frequency spectrum (➜ easier)
  - vocabulary growth curve

Goodness-of-fit

- Goodness-of-fit statistics measure how well the model has been fitted to the observed training data
- Compare observed vs. expected frequency distribution
  - frequency spectrum (➜ easier)
  - vocabulary growth curve
- Similarity measures
  - mean square error (➜ dominated by large V / V_m)
  - multivariate chi-squared statistic X^2 takes sampling variation (and covariance of spectrum elements) into account

Goodness-of-fit

- Goodness-of-fit statistics measure how well the model has been fitted to the observed training data
- Compare observed vs. expected frequency distribution
  - frequency spectrum (➜ easier)
  - vocabulary growth curve
- Similarity measures
  - mean square error (➜ dominated by large V / V_m)
  - multivariate chi-squared statistic X^2 takes sampling variation (and covariance of spectrum elements) into account
- Multivariate chi-squared test for goodness-of-fit
  - H_0 : observed data = sample from LNRE model (i.e. fitted LNRE model describes the true population)
  - p-value derived from X^2 statistic ($X^2 \sim \chi_{\mathrm{df}}$ under H_0)
  - in previous example: $p \approx 0$:-(

How reliable are the fitted models?

Three potential issues:

How reliable are the fitted models?

Three potential issues:

1. Model assumptions $\neq$ population
   (e.g. distribution does not follow a Zipf-Mandelbrot law)
   - ☞ model cannot be adequate, regardless of parameter settings

How reliable are the fitted models?

Three potential issues:

1. Model assumptions $\neq$ population
   (e.g. distribution does not follow a Zipf-Mandelbrot law)
   - ☞ model cannot be adequate, regardless of parameter settings

2. Parameter estimation unsuccessful
   (i.e. suboptimal goodness-of-fit to training data)
   - ☞ optimization algorithm trapped in local minimum
   - ☞ can result in highly inaccurate model

How reliable are the fitted models?

Three potential issues:

1. Model assumptions $\neq$ population
   (e.g. distribution does not follow a Zipf-Mandelbrot law)
   - ☞ model cannot be adequate, regardless of parameter settings

2. Parameter estimation unsuccessful
   (i.e. suboptimal goodness-of-fit to training data)
   - ☞ optimization algorithm trapped in local minimum
   - ☞ can result in highly inaccurate model

3. Uncertainty due to sampling variation
   (i.e. observed training data differ from population distribution)
   - ☞ model fitted to training data, may not reflect true population
   - ☞ another training sample would have led to different parameters

Bootstrapping

- An empirical approach to sampling variation:
  - take many random samples from the same population
  - estimate LNRE model from each sample
  - analyse distribution of model parameters, goodness-of-fit, etc. (mean, median, s.d., boxplot, histogram, . . .)
  - problem: how to obtain the additional samples?

Bootstrapping

- An empirical approach to sampling variation:
  - take many random samples from the same population
  - estimate LNRE model from each sample
  - analyse distribution of model parameters, goodness-of-fit, etc. (mean, median, s.d., boxplot, histogram, ...)
  - problem: how to obtain the additional samples?
- Bootstrapping (Efron 1979)
  - resample from observed data *with replacement*
  - this approach is not suitable for type-token distributions (resamples underestimate vocabulary size V!)

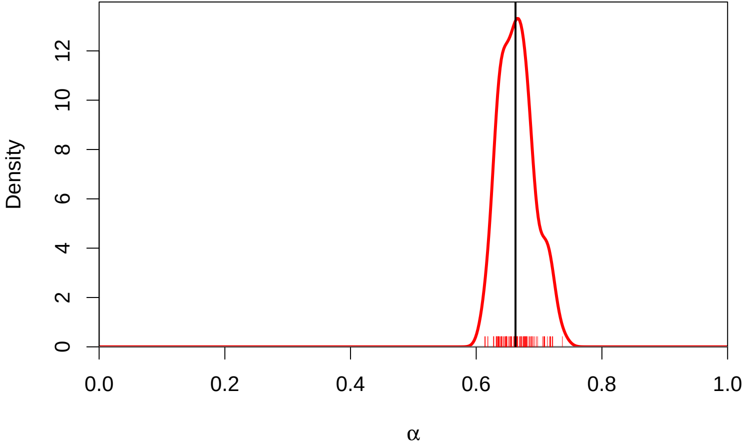
Bootstrapping

- An empirical approach to sampling variation:
  - take many random samples from the same population
  - estimate LNRE model from each sample
  - analyse distribution of model parameters, goodness-of-fit, etc. (mean, median, s.d., boxplot, histogram, ...)
  - problem: how to obtain the additional samples?
- Bootstrapping (Efron 1979)
  - resample from observed data *with replacement*
  - this approach is not suitable for type-token distributions (resamples underestimate vocabulary size V!)
- Parametric bootstrapping
  - use fitted model to generate samples, i.e. sample from the population described by the model
  - advantage: "correct" parameter values are known

Bootstrapping

parametric bootstrapping with 100 replicates

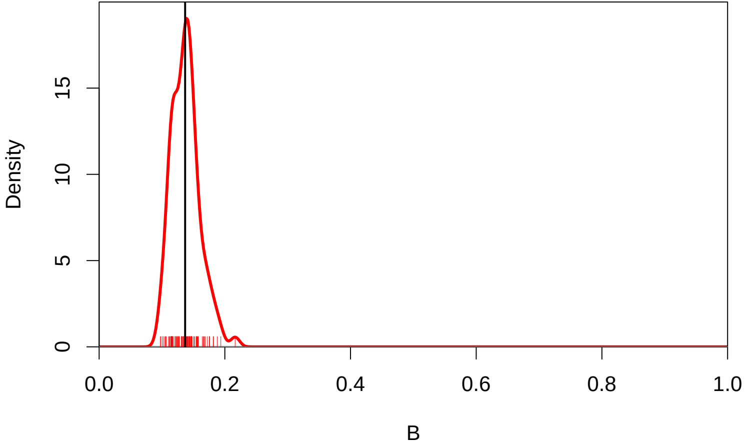
Zipfian slope $a = 1/\alpha$



Bootstrapping

parametric bootstrapping with 100 replicates

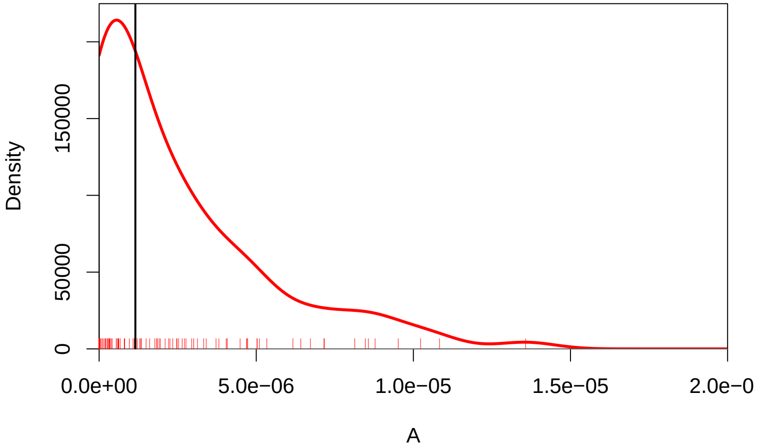
Offset $b = (1-\alpha)/(B \cdot \alpha)$



Bootstrapping

parametric bootstrapping with 100 replicates

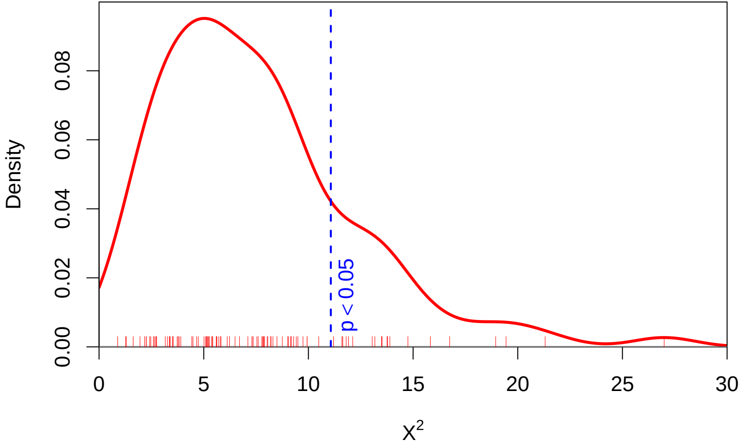
fZM probability cutoff $A = \pi_S$



Bootstrapping

parametric bootstrapping with 100 replicates

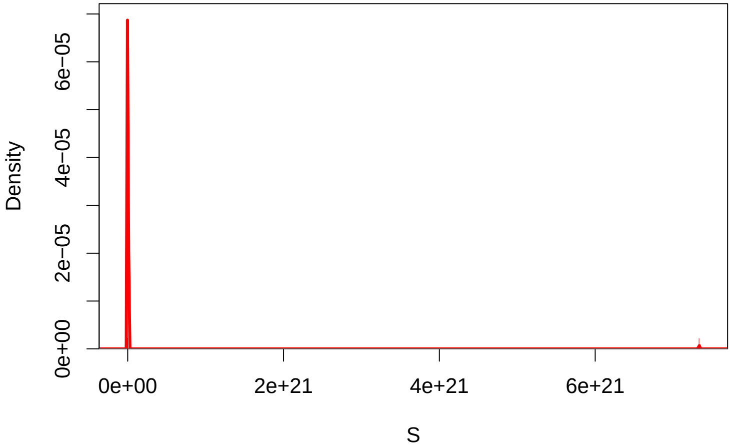
Goodness-of-fit statistic X^2 (model not plausible for $X^2 > 11$)



Bootstrapping

parametric bootstrapping with 100 replicates

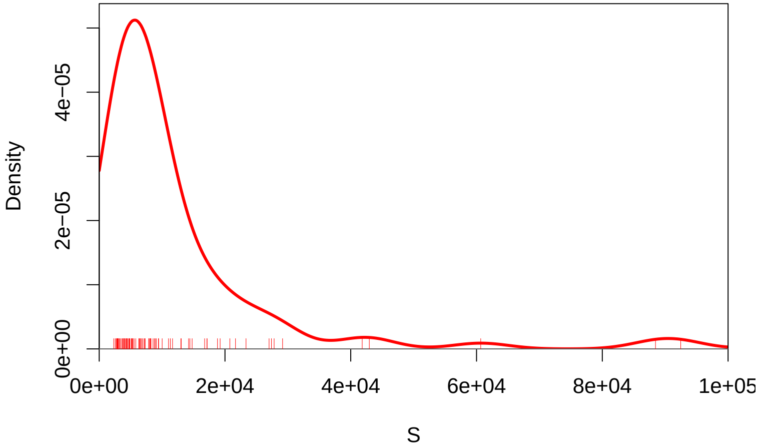
Population vocabulary size S



Bootstrapping

parametric bootstrapping with 100 replicates

Population vocabulary size S



Summary

LNRE modelling in a nutshell:

Summary

LNRE modelling in a nutshell:

1. Compile **observed** frequency spectrum (and vocabulary growth curves) for a given corpus or data set

Summary

LNRE modelling in a nutshell:

1. Compile **observed** frequency spectrum (and vocabulary growth curves) for a given corpus or data set
2. Estimate parameters of **LNRE model** by matching observed and expected frequency spectrum

Summary

LNRE modelling in a nutshell:

1. Compile **observed** frequency spectrum (and vocabulary growth curves) for a given corpus or data set
2. Estimate parameters of **LNRE model** by matching observed and expected frequency spectrum
3. Evaluate **goodness-of-fit** on spectrum (Baayen 2001) or by testing extrapolation accuracy (Baroni and Evert 2007)
   - in principle, you should only go on if model gives a plausible explanation of the observed data!

Summary

LNRE modelling in a nutshell:

1. Compile **observed** frequency spectrum (and vocabulary growth curves) for a given corpus or data set
2. Estimate parameters of **LNRE model** by matching observed and expected frequency spectrum
3. Evaluate **goodness-of-fit** on spectrum (Baayen 2001) or by testing extrapolation accuracy (Baroni and Evert 2007)
   - in principle, you should only go on if model gives a plausible explanation of the observed data!
4. Use LNRE model to compute **expected** frequency spectrum for arbitrary sample sizes
   ➜ **extrapolation** of vocabulary growth curve
   - or use population model directly as Bayesian prior etc.

zipfR

Evert and Baroni (2007)

- http://zipfR.R-Forge.R-Project.org/
- Conveniently available from CRAN repository
  - see Unit 1 for general package installation guides

Loading

```
> library(zipfR)

> ?zipfR

> data(package="zipfR")

# package overview in HTML help leads to zipfR tutorial
> help.start()
```

Importing data

```
> data(ItaRi.spc)        # not necessary in recent package versions
> data(ItaRi.emp.vgc)

# load your own data sets (see ?read.spc etc. for file format)
> my.spc <- read.spc("my.spc.txt")
> my.vgc <- read.vgc("my.vgc.txt")

> my.tfl <- read.tfl("my.tfl.txt")
> my.spc <- tfl2spc(my.tfl)  # compute spectrum from frequency list
```

Looking at spectra

```
> summary(ItaRi.spc)
> ItaRi.spc

> N(ItaRi.spc)
> V(ItaRi.spc)
> Vm(ItaRi.spc, 1)
> Vm(ItaRi.spc, 1:5)

# Baayen's P = estimate for slope of VGC
> Vm(ItaRi.spc, 1) / N(ItaRi.spc)

> plot(ItaRi.spc)
> plot(ItaRi.spc, log="x")
```

Looking at VGCs

```
> summary(ItaRi.emp.vgc)
> ItaRi.emp.vgc

> N(ItaRi.emp.vgc)

> plot(ItaRi.emp.vgc, add.m=1)
```

Smoothing VGCs with binomial interpolation

(for details, see Baayen 2001, Sec. 2.6.1)

```
# interpolated VGC
> ItaRi.bin.vgc <-
  vgc.interp(ItaRi.spc, N(ItaRi.emp.vgc), m.max=1)

> summary(ItaRi.bin.vgc)

# comparison
> plot(ItaRi.emp.vgc, ItaRi.bin.vgc,
       legend=c("observed", "interpolated"))
```

ultra-

- Load the spectrum and empirical VGC of the less common prefix *ultra-*
- Compute binomially interpolated VGC for *ultra-*
- Plot the binomially interpolated *ri-* and *ultra-* VGCs together

Estimating LNRE models

```
# fit a fZM model
# (you can also try ZM and GIGP, and compare them with fZM)

> ItaUltra.fzm <- lnre("fzm", ItaUltra.spc)

> summary(ItaUltra.fzm)
```

Observed/expected spectra at estimation size

```
# expected spectrum
> ItaUltra.fzm.spc <-
  lnre.spc(ItaUltra.fzm, N(ItaUltra.fzm))

# compare
> plot(ItaUltra.spc, ItaUltra.fzm.spc,
       legend=c("observed", "fzm"))

# plot first 10 elements only
> plot(ItaUltra.spc, ItaUltra.fzm.spc,
       legend=c("observed", "fzm"), m.max=10)
```

Compare growth of two categories

```
# extrapolation of ultra- VGC to sample size of ri- data
> ItaUltra.ext.vgc <-
  lnre.vgc(ItaUltra.fzm, N(ItaRi.emp.vgc))

# compare
> plot(ItaUltra.ext.vgc, ItaRi.bin.vgc,
       N0=N(ItaUltra.fzm), legend=c("ultra-", "ri-"))

# zooming in
> plot(ItaUltra.ext.vgc, ItaRi.bin.vgc,
       N0=N(ItaUltra.fzm), legend=c("ultra-", "ri-"),
       xlim=c(0, 100e3))
```

Model validation by parametric bootstrapping

```
# define function to extract relevant information from fitted model
> extract.stats <- function (m)
    data.frame(alpha=m$param$alpha, A=m$param$A,
               B=m$param$B, S=m$S, X2=m$gof$X2)

# run bootstrapping procedure (default = 100 replicates)
> runs <- lnre.bootstrap(ItaUltra.fzm, N(ItaUltra.fzm),
                         lnre, extract.stats, type="fzm")

> head(runs)

# NB: don't try this with large samples (N > 1M tokens)
```

Model validation by parametric bootstrapping

```
# distribution of estimated model parameters
> hist(runs$alpha, freq=FALSE, xlim=c(0, 1))
> lines(density(runs$alpha), lwd=2, col="red")
> abline(v=ItaUltra.fzm$param$alpha, lwd=2, col="blue")

# try the other parameters for yourself!

# distribution of goodness-of-fit values
> hist(runs$X2, freq=FALSE)
> lines(density(runs$X2), lwd=2, col="red")

# estimated population vocabulary size
> hist(runs$S)  # what is wrong here?
```

References I

Baayen, R. Harald (2001). *Word Frequency Distributions*. Kluwer Academic Publishers, Dordrecht.

Baroni, Marco (2008). Distributions in text. In A. Lüdeling and M. Kytö (eds.), *Corpus Linguistics. An International Handbook*, chapter 39. Mouton de Gruyter, Berlin.

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.

Efron, Bradley (1979). Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, **7**(1), 1–26.

Efron, Bradley and Thisted, Ronald (1976). Estimating the number of unseen species: How many words did Shakespeare know? *Biometrika*, **63**(3), 435–447.

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.

References II

Gale, William A. and Sampson, Geoffrey (1995). Good-Turing frequency estimation without tears. *Journal of Quantitative Linguistics*, **2**(3), 217–237.

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.

Good, I. J. (1953). The population frequencies of species and the estimation of population parameters. *Biometrika*, **40**(3/4), 237–264.

Lüdeling, Anke and Evert, Stefan (2005). The emergence of productive non-medical *-itis*. corpus evidence and qualitative analysis. In S. Kepser and M. Reis (eds.), *Linguistic Evidence. Empirical, Theoretical, and Computational Perspectives*, pages 351–370. Mouton de Gruyter, Berlin.

Mandelbrot, Benoit (1953). An informational theory of the statistical structure of languages. In W. Jackson (ed.), *Communication Theory*, pages 486–502. Butterworth, London.

Mandelbrot, Benoit (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.

References III

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.