Counting Words:
Type-rich populations, samples, and statistical models

Marco Baroni & Stefan Evert

Málaga, 8 August 2006

Why we need the population

There are two reasons why we want to construct a model of the type population distribution:

- Population distribution is interesting by itself, for theoretical reasons or in NLP applications
- We know how to simulate sampling from population → once we have a population model, we can obtain estimates of $V(N)$, $V_1(N)$ and similar quantities for arbitrary sample sizes N

Why we need the population

There are two reasons why we want to construct a model of the type population distribution:

- Population distribution is interesting by itself, for theoretical reasons or in NLP applications
- We know how to simulate sampling from population ➜ once we have a population model, we can obtain estimates of $V(N)$, $V_1(N)$ and similar quantities for arbitrary sample sizes N

A third reason:

- The bell-bottom shape of the observed Zipf ranking does not fit Zipf's law (type frequencies must be integers!)
- It is more natural to characterize occurrence *probabilities* (for which there is no such restriction) by Zipf's law

A population of types

- A type population is characterized by
  a) a set of **types** w_k
  b) the corresponding occurrence **probabilities** π_k

A population of types

- A type population is characterized by
  a) a set of **types** w_k
  b) the corresponding occurrence **probabilities** π_k
- The actual "identities" of the types are irrelevant (for word frequency distributions)
  - we don't care whether w_{43194} is *wormhole* or *heatwave*

A population of types

- A type population is characterized by
  a) a set of **types** w_k
  b) the corresponding occurrence **probabilities** π_k
- The actual "identities" of the types are irrelevant (for word frequency distributions)
  - we don't care whether w_{43194} is *wormhole* or *heatwave*
- It is customary (and convenient) to arrange types in order of decreasing probability: $\pi_1 \geq \pi_2 \geq \pi_3 \geq \cdots$
- NB: this is usually *not* the same ordering as in the observed Zipf ranking (we will see examples of this later)

Today's quiz . . .

Everybody remember what probabilities are?

Today's quiz . . .

Everybody remember what probabilities are?

- $0 \leq \pi_k \leq 1$ (for all k)

Today's quiz . . .

Everybody remember what probabilities are?

- $0 \leq \pi_k \leq 1$ (for all k)
- $\sum_k \pi_k = \pi_1 + \pi_2 + \pi_3 + \cdots = 1$

Today's quiz (cont'd)

And what their interpretation is?

Today's quiz (cont'd)

And what their interpretation is?

- π_k = relative frequency of w_k in huge body of text
  - e.g. population = "written English", formalized as all English writing that has ever been published
  - also: π_k = chances that a token drawn at random belongs to type w_k

Today's quiz (cont'd)

And what their interpretation is?

- π_k = relative frequency of w_k in huge body of text
  - e.g. population = "written English", formalized as all English writing that has ever been published
  - also: π_k = chances that a token drawn at random belongs to type w_k
- π_k = output probability for w_k in generative model
  - e.g. psycholinguistic model of a human speaker
  - π_k = probability that next word uttered by the speaker belongs to type w_k (without knowledge about context and previous words)

Today's quiz (cont'd)

And what their interpretation is?

- π_k = relative frequency of w_k in huge body of text
  - e.g. population = "written English", formalized as all English writing that has ever been published
  - also: π_k = chances that a token drawn at random belongs to type w_k
- π_k = output probability for w_k in generative model
  - e.g. psycholinguistic model of a human speaker
  - π_k = probability that next word uttered by the speaker belongs to type w_k (without knowledge about context and previous words)
- analogous interpretations for other linguistic and non-linguistic phenomena

The problem with probabilities . . .

- We cannot measure these probabilities directly
- In principle, such probabilities can be estimated from a sample (that's what most of statistics is about), e.g.

$$\pi \approx \frac{f}{n}$$

The problem with probabilities ...

- We cannot measure these probabilities directly
- In principle, such probabilities can be estimated from a sample (that's what most of statistics is about), e.g.

$$\pi \approx \frac{f}{n}$$

- But we cannot reliably estimate thousands or millions of π_k's from any finite sample (just think of all the unseen types that do not occur in the sample)

. . . and its solution

➥ We need a **model** for the population

. . . and its solution

➥ We need a **model** for the population

- This model embodies our hypothesis that the distribution of type probabilities has a certain general shape (more precisely, we speak of a **family** of models)

. . . and its solution

➥ We need a **model** for the population

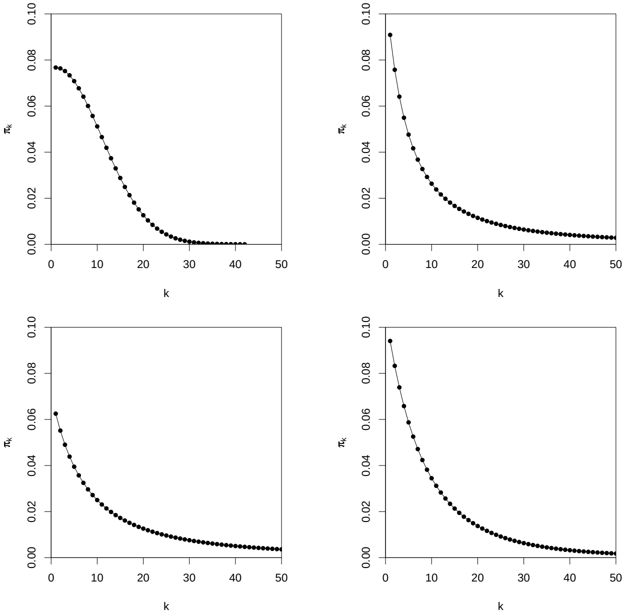
- This model embodies our hypothesis that the distribution of type probabilities has a certain general shape (more precisely, we speak of a **family** of models)
- The exact form of the distribution is then determined by a small number of **parameters** (typically 2 or 3)

… and its solution

➥ We need a **model** for the population

- This model embodies our hypothesis that the distribution of type probabilities has a certain general shape (more precisely, we speak of a **family** of models)
- The exact form of the distribution is then determined by a small number of **parameters** (typically 2 or 3)
- These parameters can be estimated with relative ease

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, across many phenomena and data sets

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, across many phenomena and data sets
- Re-phrase the law for type probabilities instead of frequencies:

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

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, across many phenomena and data sets
- Re-phrase the law for type probabilities instead of frequencies:

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

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

The Zipf-Mandelbrot law as a population model

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

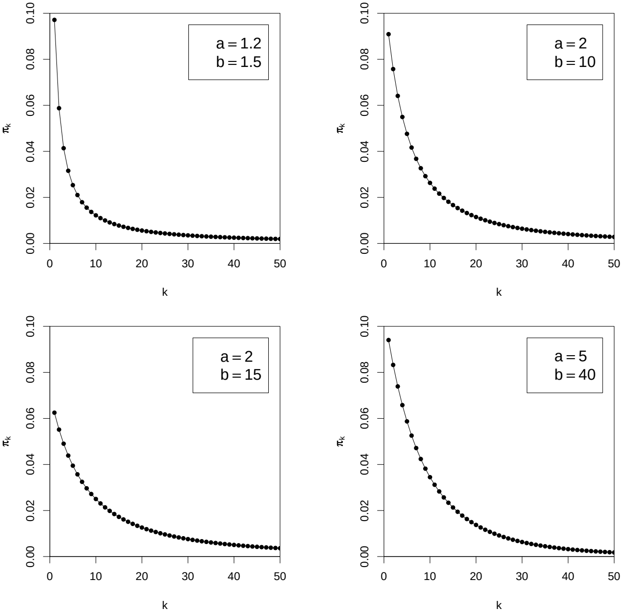
- We have already seen that the Zipf-Mandelbrot law captures the distribution of observed frequencies very well, across many phenomena and data sets
- Re-phrase the law for type probabilities instead of frequencies:

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

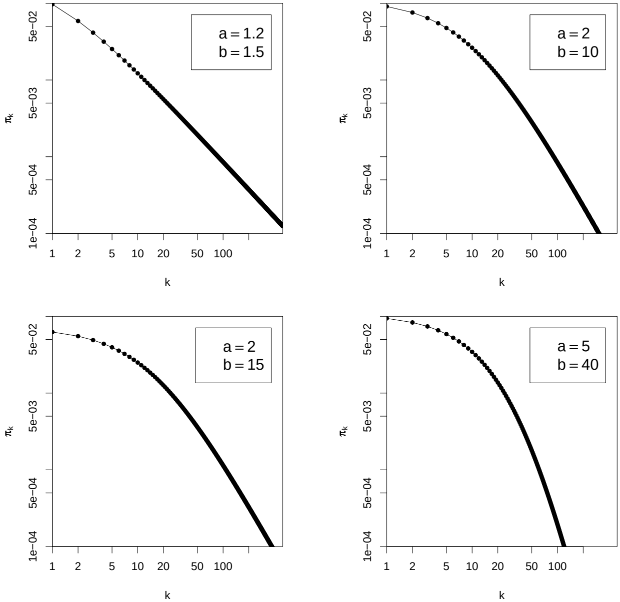
- Two free parameters: $a > 1$ and $b \geq 0$
- C is not a parameter but a normalization constant, needed to ensure that $\sum_k \pi_k = 1$

➥ the **Zipf-Mandelbrot** population model

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

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), $\pi = 10^{-12}$ (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), $\pi = 10^{-12}$ (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

- The **finite Zipf-Mandelbrot** model simply stops after the first S types $(w_1, \ldots, w_S)$

The finite Zipf-Mandelbrot model

- The **finite Zipf-Mandelbrot** model simply stops after the first S types $(w_1, \ldots, w_S)$
- S becomes a new parameter of the model
  → the finite Zipf-Mandelbrot model has 3 parameters
- NB: C will not have the same value as for the corresponding infinite ZM model

The finite Zipf-Mandelbrot model

- The **finite Zipf-Mandelbrot** model simply stops after the first S types $(w_1, \ldots, w_S)$
- S becomes a new parameter of the model
  ➜ the finite Zipf-Mandelbrot model has 3 parameters
- NB: C will not have the same value as for the corresponding infinite ZM model

Abbreviations: **ZM** for **Zipf-Mandelbrot** model,
and **fZM** for **finite Zipf-Mandelbrot** model

The next steps

Once we have a population model ...

- We still need to estimate the values of its parameters
  - we'll see later how we can do this

The next steps

Once we have a population model ...

- We still need to estimate the values of its parameters
  - we'll see later how we can do this
- We want to simulate random samples from the population described by the model
  - basic assumption: real data sets (such as corpora) are random samples from this population

The next steps

Once we have a population model . . .

- We still need to estimate the values of its parameters
  - we'll see later how we can do this
- We want to simulate random samples from the population described by the model
  - basic assumption: real data sets (such as corpora) are random samples from this population
  - this allows us to predict vocabulary growth, the number of previously unseen types as more text is added to a corpus, the frequency spectrum of a larger data set, etc.

The next steps

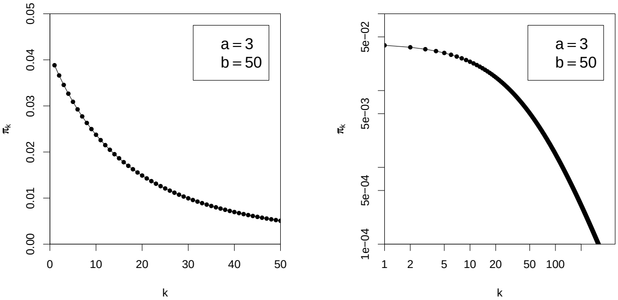
Once we have a population model ...

- We still need to estimate the values of its parameters
  - we'll see later how we can do this
- We want to simulate random samples from the population described by the model
  - basic assumption: real data sets (such as corpora) are random samples from this population
  - this allows us to predict vocabulary growth, the number of previously unseen types as more text is added to a corpus, the frequency spectrum of a larger data set, etc.
  - it will also allow us to estimate the model parameters

Outline

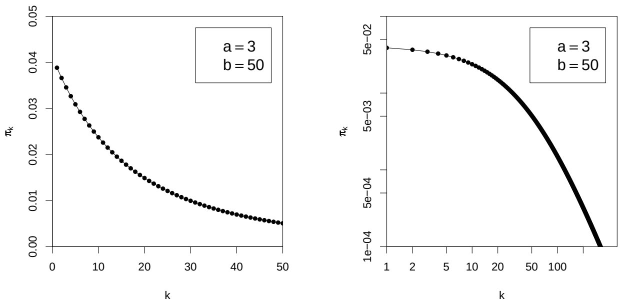
Sampling from a population model

Assume we believe that the population we are interested in can be described by a 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 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

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	...

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	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

Sampling from a population model

In this way, we can …

- draw samples of arbitrary size N
  - the computer can do it efficiently even for large N

Sampling from a population model

In this way, we can ...

- draw samples of arbitrary size N
  - the computer can do it efficiently even for large N
- draw as many samples as we need

Sampling from a population model

In this way, we can ...

- draw samples of arbitrary size N
  - the computer can do it efficiently even for large N
- draw as many samples as we need
- compute type frequency lists, frequency spectra and vocabulary growth curves from these samples
  - i.e., we can analyze them with the same methods that we have applied to the observed data sets

Sampling from a population model

In this way, we can ...

- draw samples of arbitrary size N
  - the computer can do it efficiently even for large N
- draw as many samples as we need
- compute type frequency lists, frequency spectra and vocabulary growth curves from these samples
  - i.e., we can analyze them with the same methods that we have applied to the observed data sets

Here are some results for samples of size $N = 1000$ ...

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

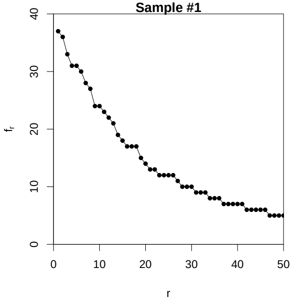
Samples: type frequency list & spectrum

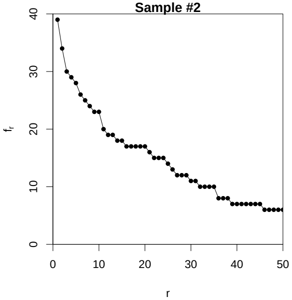
rank r	f_r	type k
1	39	2
2	34	3
3	30	5
4	29	10
5	28	8
6	26	1
7	25	13
8	24	7
9	23	6
10	23	11
11	20	4
12	19	17
$\vdots$	$\vdots$	$\vdots$

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

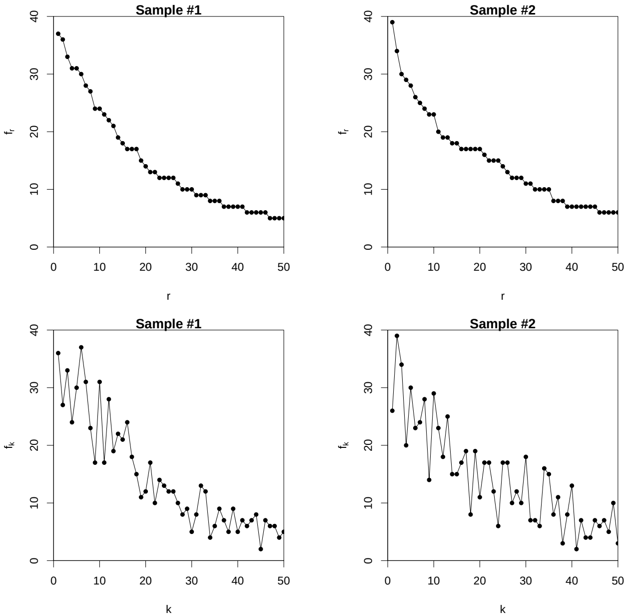
Random variation in type-frequency lists





$r \leftrightarrow f_r$

Random variation in type-frequency lists



$r \leftrightarrow f_r$

$k \leftrightarrow f_k$

Random variation in type-frequency lists

- Random variation leads to different type frequencies f_k in every new sample
  - particularly obvious when we plot them in population order (bottom row, $k \leftrightarrow f_k$)

Random variation in type-frequency lists

- Random variation leads to different type frequencies f_k in every new sample
  - particularly obvious when we plot them in population order (bottom row, $k \leftrightarrow f_k$)
- Different ordering of types in the Zipf ranking for every new sample
  - Zipf rank r in sample $\neq$ population rank k!
  - leads to severe problems with statistical methods

Random variation in type-frequency lists

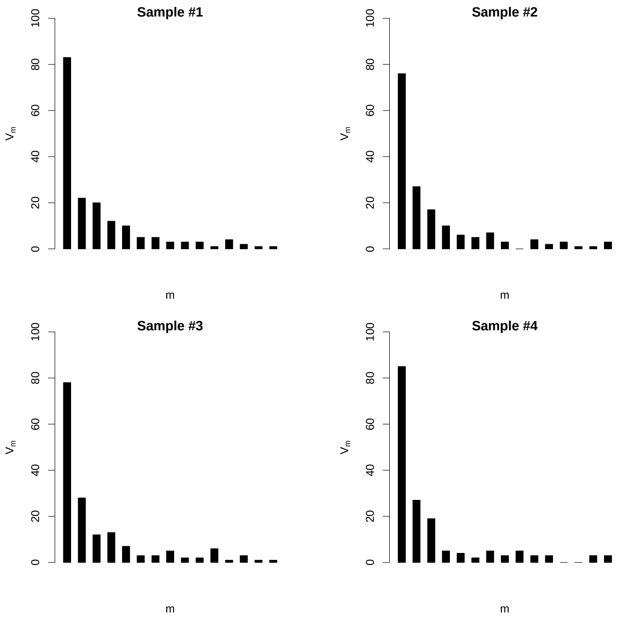
- Random variation leads to different type frequencies f_k in every new sample
  - particularly obvious when we plot them in population order (bottom row, $k \leftrightarrow f_k$)
- Different ordering of types in the Zipf ranking for every new sample
  - Zipf rank r in sample $\neq$ population rank k!
  - leads to severe problems with statistical methods
- Individual types are irrelevant for our purposes, so let us take a perspective that abstracts away from them
  - frequency spectrum
  - vocabulary growth curve

Random variation in type-frequency lists

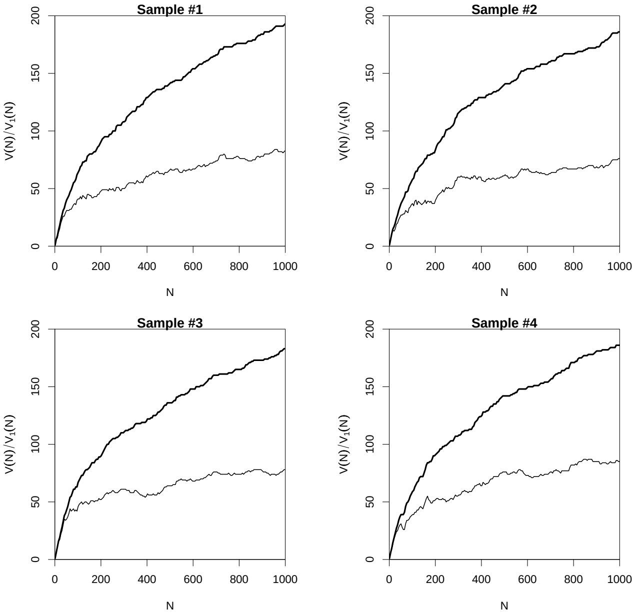
- Random variation leads to different type frequencies f_k in every new sample
  - particularly obvious when we plot them in population order (bottom row, $k \leftrightarrow f_k$)
- Different ordering of types in the Zipf ranking for every new sample
  - Zipf rank r in sample $\neq$ population rank k!
  - leads to severe problems with statistical methods
- Individual types are irrelevant for our purposes, so let us take a perspective that abstracts away from them
  - frequency spectrum
  - vocabulary growth curve

➥ considerable amount of random variation still visible

Random variation: frequency spectrum



Random variation: vocabulary growth curve



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

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

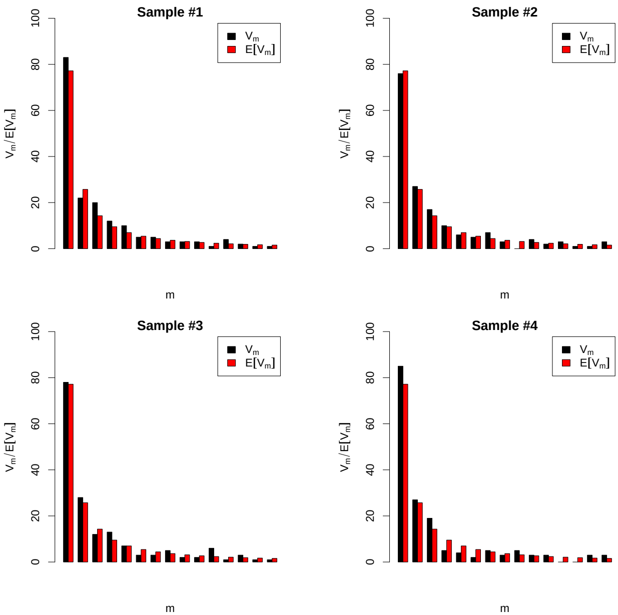
Expected values

- 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
- Such averages are called **expected values** or **expectations** in statistics (frequentist approach)

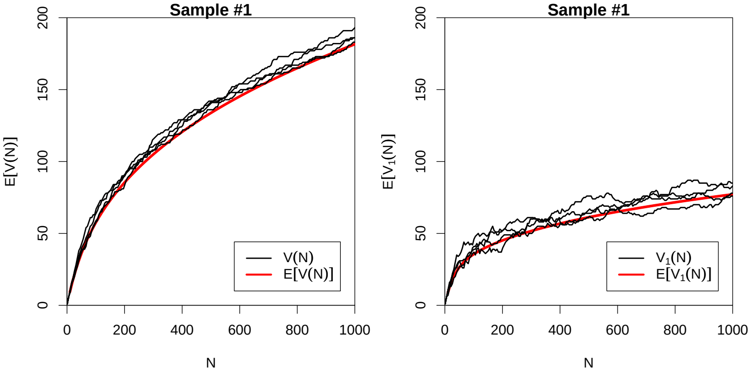
Expected values

- 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
- Such averages are called **expected values** or **expectations** in statistics (frequentist approach)
- Notation: $\mathrm{E}[V(N)]$ and $\mathrm{E}[V_m(N)]$
  - 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
- Usually we can omit the sample size: $\mathrm{E}[V]$ and $\mathrm{E}[V_m]$

The expected frequency spectrum



The expected vocabulary growth curve



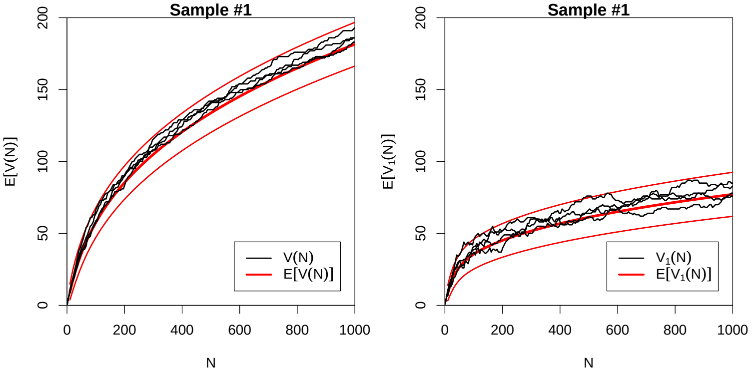
Great expectations made easy

- Fortunately, we don't have to take many thousands of samples to calculate expectations: there is a (relatively simple) mathematical solution (➜ Wednesday)

Great expectations made easy

- Fortunately, we don't have to take many thousands of samples to calculate expectations: there is a (relatively simple) mathematical solution (➜ Wednesday)
- This solution also allows us to estimate the amount of random variation ➜ **variance** and **confidence intervals**
  - example: expected VGCs with confidence intervals
  - we won't pursue variance any further in this course

Confidence intervals for the expected VGC



A mini-example

- G. K. Zipf claimed that the distribution of English word frequencies follows Zipf's law with $a \approx 1$
  - $a \approx 1.5$ seems a more reasonable value when you look at larger text samples than Zipf did

A mini-example

- G. K. Zipf claimed that the distribution of English word frequencies follows Zipf's law with $a \approx 1$
  - $a \approx 1.5$ seems a more reasonable value when you look at larger text samples than Zipf did
- The most frequent word in English is *the* with $\pi \approx .06$

A mini-example

- G. K. Zipf claimed that the distribution of English word frequencies follows Zipf's law with $a \approx 1$
  - $a \approx 1.5$ seems a more reasonable value when you look at larger text samples than Zipf did
- The most frequent word in English is *the* with $\pi \approx .06$
- Zipf-Mandelbrot law with $a = 1.5$ and $b = 7.5$ yields a population model where $\pi_1 \approx .06$ (by trial & error)

A mini-example

- How many different words do we expect to find in a 1-million word text?
  - $N = 1{,}000{,}000 \rightarrow \mathrm{E}[V(N)] = 33026.7$
  - 95%-confidence interval: $V(N) = 32753.6 \ldots 33299.7$

A mini-example

- How many different words do we expect to find in a 1-million word text?
  - $N = 1{,}000{,}000$ ➜ $\mathrm{E}[V(N)] = 33026.7$
  - 95%-confidence interval: $V(N) = 32753.6 \ldots 33299.7$
- How many do we really find?
  - Brown corpus: 1 million words of edited American English
  - $V = 45215$ ➜ ZM model is not quite right

A mini-example

- How many different words do we expect to find in a 1-million word text?
  - $N = 1{,}000{,}000 \rightarrow \mathrm{E}[V(N)] = 33026.7$
  - 95%-confidence interval: $V(N) = 32753.6 \ldots 33299.7$
- How many do we really find?
  - Brown corpus: 1 million words of edited American English
  - $V = 45215 \rightarrow$ ZM model is not quite right
  - Physicists (and some mathematicians) are happy as long as they get the order of magnitude right . . .

A mini-example

- How many different words do we expect to find in a 1-million word text?
  - $N = 1{,}000{,}000$ ➜ $\mathrm{E}[V(N)] = 33026.7$
  - 95%-confidence interval: $V(N) = 32753.6 \ldots 33299.7$
- How many do we really find?
  - Brown corpus: 1 million words of edited American English
  - $V = 45215$ ➜ ZM model is not quite right
  - Physicists (and some mathematicians) are happy as long as they get the order of magnitude right …

☞ Model was not based on actual data!

Outline

Estimating model parameters

- Parameter settings in the mini-example were based on general assumptions (claims from the literature)
- But we also have empirical data on the word frequency distribution of English available (the Brown corpus)

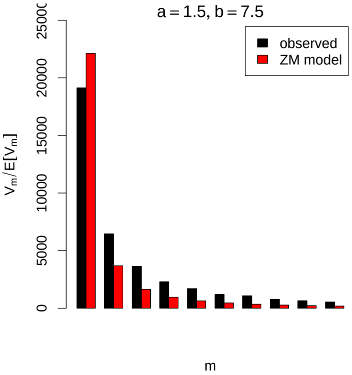
Estimating model parameters

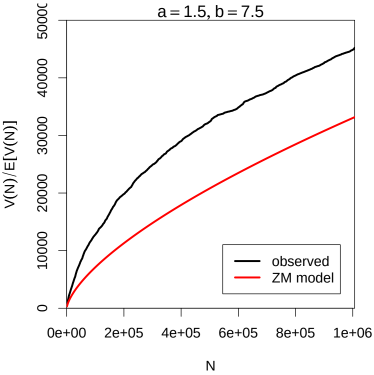
- Parameter settings in the mini-example were based on general assumptions (claims from the literature)
- But we also have empirical data on the word frequency distribution of English available (the Brown corpus)
- Choose parameters so that population model matches the empirical distribution as well as possible

Estimating model parameters

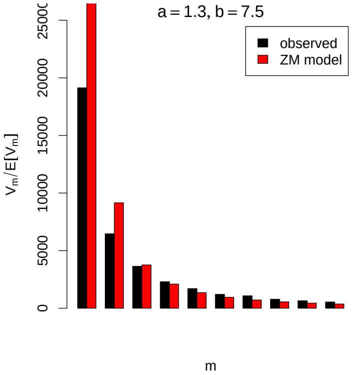
- Parameter settings in the mini-example were based on general assumptions (claims from the literature)
- But we also have empirical data on the word frequency distribution of English available (the Brown corpus)
- Choose parameters so that population model matches the empirical distribution as well as possible
- E.g. by trial and error ...
  - guess parameters
  - compare model predictions for sample of size N_0 with observed data (N_0 tokens)
  - based on frequency spectrum or vocabulary growth curve
  - change parameters & repeat until satisfied
- This process is called **parameter estimation**

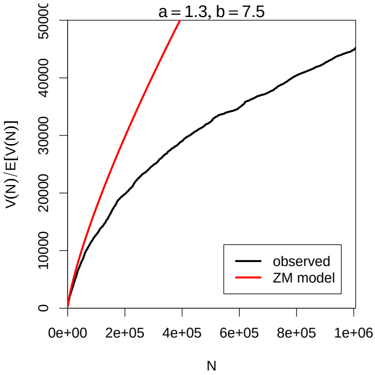
Parameter estimation by trial & error



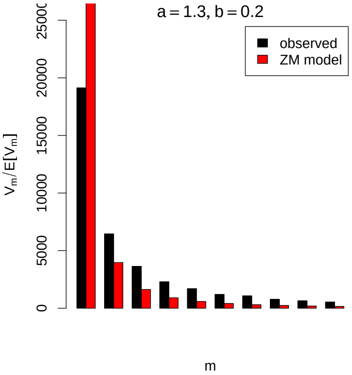


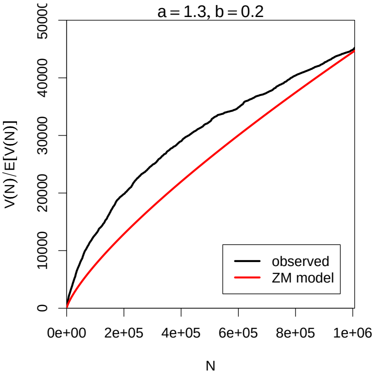
Parameter estimation by trial & error



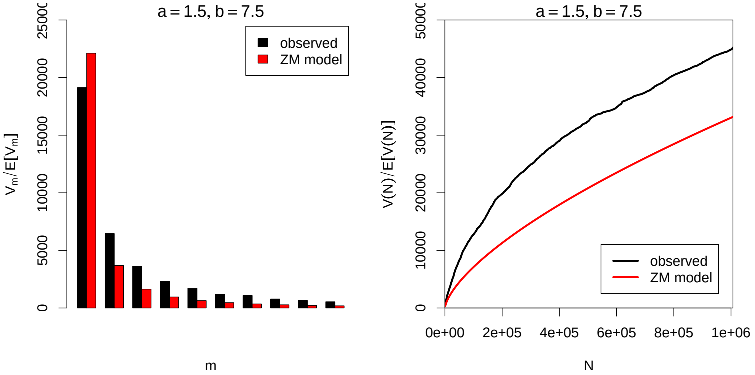


Parameter estimation by trial & error

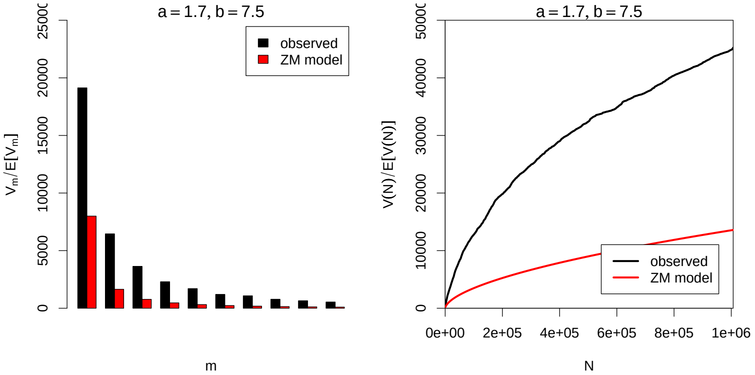




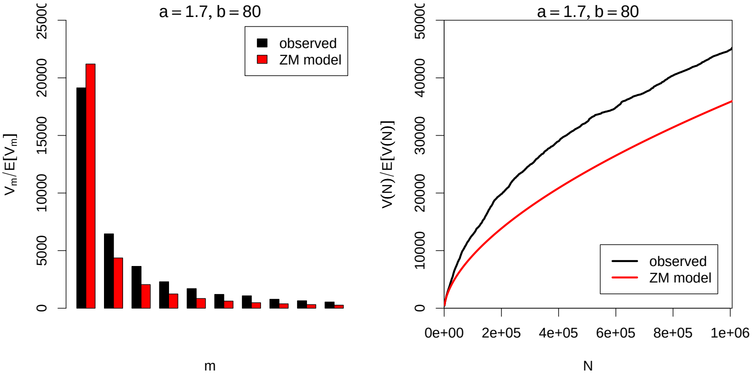
Parameter estimation by trial & error



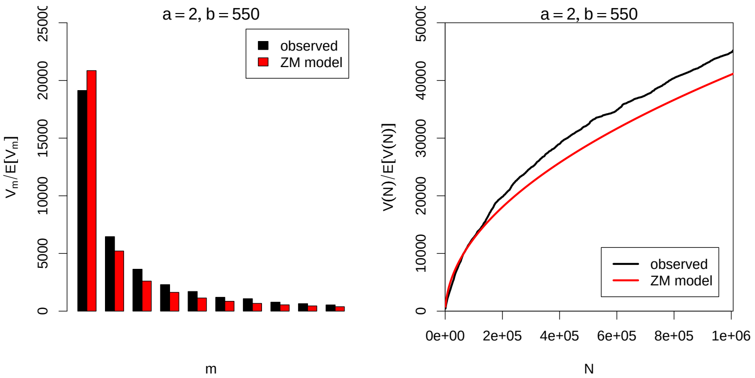
Parameter estimation by trial & error



Parameter estimation by trial & error



Parameter estimation by trial & error



Automatic parameter estimation

- Parameter estimation by trial & error is tedious
  ➜ let the computer to the work!

Automatic parameter estimation

- Parameter estimation by trial & error is tedious
  ➜ let the computer to the work!
- Need **cost function** to quantify "distance" between model expectations and observed data
  - based on vocabulary size and vocabulary spectrum (these are the most convenient criteria)

Automatic parameter estimation

- Parameter estimation by trial & error is tedious
  ➜ let the computer to the work!
- Need **cost function** to quantify "distance" between model expectations and observed data
  - based on vocabulary size and vocabulary spectrum (these are the most convenient criteria)
- Computer estimates parameters by automatic minimization of cost function
  - clever algorithms exist that find out quickly in which direction they have to "push" the parameters to approach the minimum
  - implemented in standard software packages

Cost functions for parameter estimation

- Cost functions compare expected frequency spectrum $\mathrm{E}[V_m(N_0)]$ with observed spectrum $V_m(N_0)$

Cost functions for parameter estimation

- Cost functions compare expected frequency spectrum $\mathrm{E}[V_m(N_0)]$ with observed spectrum $V_m(N_0)$
- Choice #1: how to weight differences
  - absolute values of differences $\sum_{m=1}^{M} \left| V_m - \mathrm{E}[V_m] \right|$
  - mean squared error $\frac{1}{M} \sum_{m=1}^{M} \left(V_m - \mathrm{E}[V_m] \right)^2$
  - chi-squared criterion: scale by estimated variances

Cost functions for parameter estimation

- Cost functions compare expected frequency spectrum $\mathrm{E}[V_m(N_0)]$ with observed spectrum $V_m(N_0)$
- Choice #1: how to weight differences
- Choice #2: how many spectrum elements to use
  - typically between $M = 2$ and $M = 15$
  - what happens if $M <$ number of parameters?

Cost functions for parameter estimation

- Cost functions compare expected frequency spectrum $\mathrm{E}[V_m(N_0)]$ with observed spectrum $V_m(N_0)$
- Choice #1: how to weight differences
- Choice #2: how many spectrum elements to use
  - typically between $M = 2$ and $M = 15$
  - what happens if $M <$ number of parameters?
- For many applications, it is important to match V precisely: additional constraint $\mathrm{E}[V(N_0)] = V(N_0)$
  - general principle: you can match as many constraints as there are free parameters in the model

Cost functions for parameter estimation

- Cost functions compare expected frequency spectrum $\mathrm{E}[V_m(N_0)]$ with observed spectrum $V_m(N_0)$
- Choice #1: how to weight differences
- Choice #2: how many spectrum elements to use
  - typically between $M = 2$ and $M = 15$
  - what happens if $M <$ number of parameters?
- For many applications, it is important to match V precisely: additional constraint $\mathrm{E}[V(N_0)] = V(N_0)$
  - general principle: you can match as many constraints as there are free parameters in the model
- Felicitous choice of cost function and M can substantially improve the quality of the estimated model
  - It isn't a science, it's an art ...

Goodness-of-fit

- Automatic estimation procedure minimizes cost function until no further improvement can be found
  - this is a so-called **local minimum** of the cost function
  - not necessarily the global minimum that we want to find

Goodness-of-fit

- Automatic estimation procedure minimizes cost function until no further improvement can be found
  - this is a so-called **local minimum** of the cost function
  - not necessarily the global minimum that we want to find
- Key question: is the estimated model good enough?

Goodness-of-fit

- Automatic estimation procedure minimizes cost function until no further improvement can be found
  - this is a so-called **local minimum** of the cost function
  - not necessarily the global minimum that we want to find
- Key question: is the estimated model good enough?
- In other words: does the model provide a plausible explanation of the observed data as a random sample from the population?

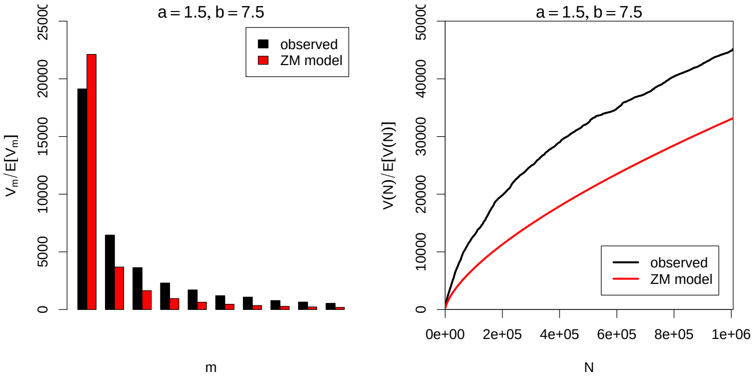
Goodness-of-fit

- Automatic estimation procedure minimizes cost function until no further improvement can be found
  - this is a so-called **local minimum** of the cost function
  - not necessarily the global minimum that we want to find
- Key question: is the estimated model good enough?
- In other words: does the model provide a plausible explanation of the observed data as a random sample from the population?
- Can be measured by **goodness-of-fit** test
  - use special tests for such models (Baayen 2001)
  - p-value specifies whether model is plausible
  - small p-value ➜ reject model as explanation for data
  - ➥ we want to achieve a *high* p-value

Goodness-of-fit

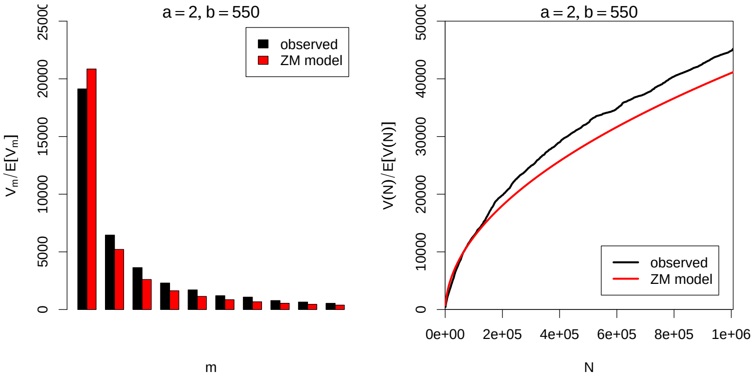
- Automatic estimation procedure minimizes cost function until no further improvement can be found
  - this is a so-called **local minimum** of the cost function
  - not necessarily the global minimum that we want to find
- Key question: is the estimated model good enough?
- In other words: does the model provide a plausible explanation of the observed data as a random sample from the population?
- Can be measured by **goodness-of-fit** test
  - use special tests for such models (Baayen 2001)
  - p-value specifies whether model is plausible
  - small p-value ➜ reject model as explanation for data
  - ➥ we want to achieve a *high* p-value
- Typically, we find $p < .001$ – but the models can still be useful for many purposes!

Mini-example (cont'd)



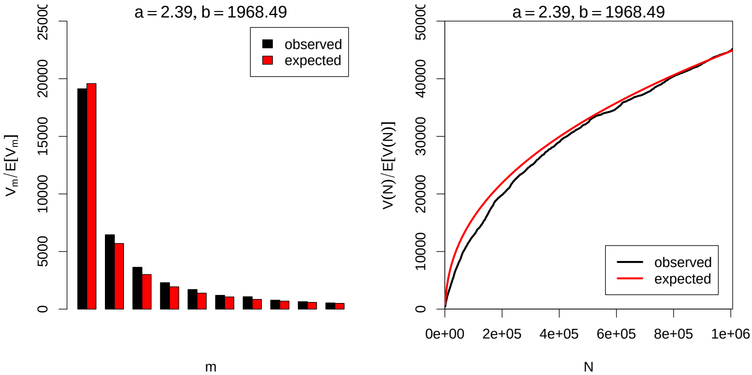
- We started with $a = 1.5$ and $b = 7.5$ (general assumptions)

Mini-example (cont'd)



- By trial & error we found $a = 2.0$ and $b = 550$

Mini-example (cont'd)



- Automatic estimation procedure: $a = 2.39$ and $b = 1968$
- Goodness-of-fit: $p \approx 0$ (but much better than before!)

Outline

Practical example: *Oliver Twist*

- A practical example: extrapolate vocabulary growth in Dickens' novel *Oliver Twist*

Practical example: *Oliver Twist*

- A practical example: extrapolate vocabulary growth in Dickens' novel *Oliver Twist*
- Observed data: $N_0 = 157302$, $V(N_0) = 10710$

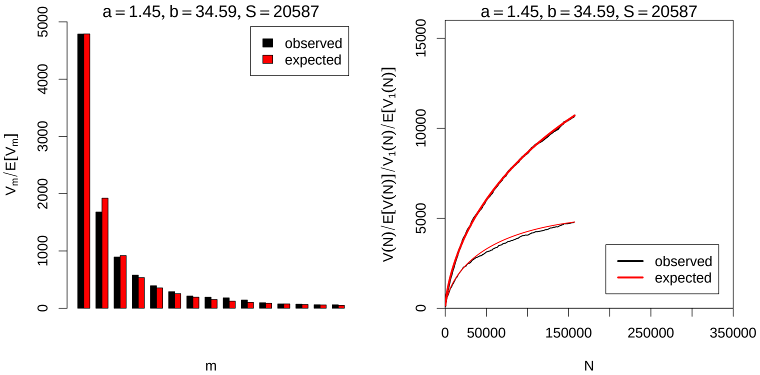
Practical example: *Oliver Twist*

- A practical example: extrapolate vocabulary growth in Dickens' novel *Oliver Twist*
- Observed data: $N_0 = 157302$, $V(N_0) = 10710$
- Our choices (experimentation & experience):
  - population model: finite Zipf-Mandelbrot
  - cost function: chi-squared type
  - number of spectrum elements: $M = 10$
  - additional constraint: $\mathrm{E}[V(N_0)] = V(N_0)$

Practical example: *Oliver Twist*

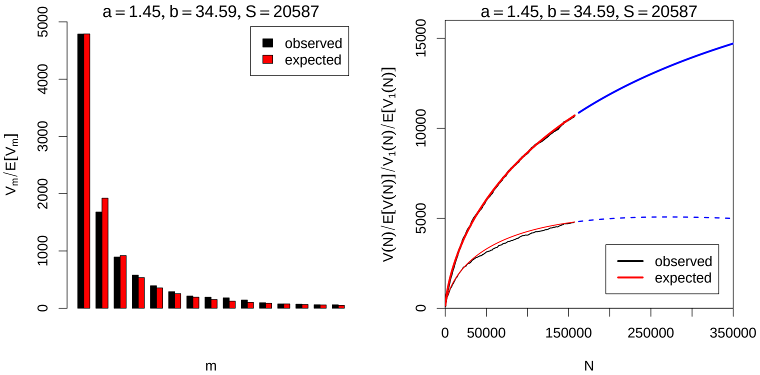
- A practical example: extrapolate vocabulary growth in Dickens' novel *Oliver Twist*
- Observed data: $N_0 = 157302$, $V(N_0) = 10710$
- Our choices (experimentation & experience):
  - population model: finite Zipf-Mandelbrot
  - cost function: chi-squared type
  - number of spectrum elements: $M = 10$
  - additional constraint: $\mathrm{E}[V(N_0)] = V(N_0)$
- Automatic parameter estimation yields $a = 1.45$, $b = 34.6$, $S = 20587$
  - population vocabulary size is extremely small
  - but this model extrapolates only the vocabulary used in *Oliver Twist*, not the full vocabulary of Charles Dickens

Results for *Oliver Twist*



- Goodness-of-fit: $p = 3.6 \cdot 10^{-40}$
  - but visually, the approximation is very good

Results for *Oliver Twist*



- Goodness-of-fit: $p = 3.6 \cdot 10^{-40}$
  - but visually, the approximation is very good