# Counting Words:
## LNRE Modelling

Marco Baroni & Stefan Evert

Málaga, 9 August 2006

# Outline

# Where we are at

- We justified an approach to lexical statistics based on population models (e.g., Zipf-Mandelbrot)

# Where we are at

- We justified an approach to lexical statistics based on population models (e.g., Zipf-Mandelbrot)
- We discussed random samples and expected values

# Where we are at

- We justified an approach to lexical statistics based on population models (e.g., Zipf-Mandelbrot)
- We discussed random samples and expected values
- We showed how to estimate model parameters by comparing observed / expected frequency spectrum

# Where we are at

- We justified an approach to lexical statistics based on population models (e.g., Zipf-Mandelbrot)
- We discussed random samples and expected values
- We showed how to estimate model parameters by comparing observed / expected frequency spectrum
- ➥ We need an efficient way to calculate expected values
  - for random samples of arbitrary size $N$
  - given a model of the population type probabilities $\pi_k$

# Expected $V_m$ for sample of size $N$

To calculate $\mathrm{E}[V_m(N)]$ ...

# Expected $V_m$ for sample of size $N$

To calculate $\mathrm{E}[V_m(N)]$ ...

- Average $V_m$ over a large number ($n$) of samples, all of them having the same size $N$

$$\mathrm{E}\big[V_m(N)\big] \approx \frac{1}{n} \cdot \big(V_m^{(1)} + V_m^{(2)} + \cdots + V_m^{(n)}\big)$$

# Expected $V_m$ for sample of size $N$

To calculate $\mathrm{E}[V_m(N)]$ ...

- Average $V_m$ over a large number ($n$) of samples, all of them having the same size $N$

$$\mathrm{E}\big[V_m(N)\big] \approx \frac{1}{n} \cdot \big(V_m^{(1)} + V_m^{(2)} + \cdots + V_m^{(n)}\big)$$

- Mathematically, $\mathrm{E}[V_m(N)]$ is the limit of this expression for $n \to \infty$ (but you can just think of $n$ as very large)

# Expected $V_m$ for sample of size $N$

- We know how to calculate the probability that in a sample of size $N$, a given type $w_k$ (with parameter $\pi_k$) occurs exactly $m$ times:

# Expected $V_m$ for sample of size $N$

- We know how to calculate the probability that in a sample of size $N$, a given type $w_k$ (with parameter $\pi_k$) occurs exactly $m$ times:

$$p_{k,m} := \binom{N}{m} (\pi_k)^m (1 - \pi_k)^{N-m}$$

# Expected $V_m$ for sample of size $N$

- We know how to calculate the probability that in a sample of size $N$, a given type $w_k$ (with parameter $\pi_k$) occurs exactly $m$ times:

$$p_{k,m} := \binom{N}{m} (\pi_k)^m (1 - \pi_k)^{N-m}$$

- Which means that it will be counted in class $V_m$ in approximately $n \cdot p_{k,m}$ out of $n$ samples
  - if $n$ is large enough, this estimate is very accurate

# Expected $V_m$ for sample of size $N$

- We know how to calculate the probability that in a sample of size $N$, a given type $w_k$ (with parameter $\pi_k$) occurs exactly $m$ times:

$$p_{k,m} := \binom{N}{m} (\pi_k)^m (1 - \pi_k)^{N-m}$$

- Which means that it will be counted in class $V_m$ in approximately $n \cdot p_{k,m}$ out of $n$ samples
  - if $n$ is large enough, this estimate is very accurate
- Taking the sum over all types $w_k$ and dividing by $n$:

$$\mathrm{E}\big[V_m(N)\big] = \sum_k p_{k,m} = \sum_k \binom{N}{m} (\pi_k)^m (1 - \pi_k)^{N-m}$$

# Binomial sampling vs. Poisson sampling

- What we have just calculated is a **binomial expectation**, i.e. the average over samples of the same fixed size $N$
  - arguably, statistically most appropriate

# Binomial sampling vs. Poisson sampling

- What we have just calculated is a **binomial expectation**, i.e. the average over samples of the same fixed size $N$
  - arguably, statistically most appropriate
- But mathematically simpler to use **Poisson expectation**:

$$\mathrm{E}\big[V_m(N)\big] = \sum_k \frac{(N\pi_k)^m}{m!} e^{-N\pi_k}$$

  - here, we sum over samples of various sizes close to $N$

# Binomial sampling vs. Poisson sampling

Switch to Poisson sampling can be motivated in two ways:

# Binomial sampling vs. Poisson sampling





Switch to Poisson sampling can be motivated in two ways:

- **Philosophical:**
  - Not as unreasonable as it seems: think of the frequency distribution of nouns in text sample of 1 million running words (such as the Brown corpus) ➜ sample size $N$ (= number of noun tokens) will be different for each sample

# Binomial sampling vs. Poisson sampling

Switch to Poisson sampling can be motivated in two ways:

- **Philosophical:**
  - Not as unreasonable as it seems: think of the frequency distribution of nouns in text sample of 1 million running words (such as the Brown corpus) ➜ sample size $N$ (= number of noun tokens) will be different for each sample
- **Practical:**
  - When $N$ is large and $\pi$ small (as with word frequency distributions), Poisson probabilities are a very good approximation to binomial probabilities

# Binomial sampling vs. Poisson sampling

Switch to Poisson sampling can be motivated in two ways:

- **Philosophical:**
  - Not as unreasonable as it seems: think of the frequency distribution of nouns in text sample of 1 million running words (such as the Brown corpus) ➜ sample size $N$ (= number of noun tokens) will be different for each sample
- **Practical:**
  - When $N$ is large and $\pi$ small (as with word frequency distributions), Poisson probabilities are a very good approximation to binomial probabilities
- In lexical statistics, word frequency distribution models almost always use Poisson expectations

# Poisson expectations for $V_m$ and $V$

$$\mathrm{E}\big[V_m(N)\big] = \sum_k \frac{(N\pi_k)^m}{m!} \cdot e^{-N\pi_k}$$

$$\mathrm{E}\big[V(N)\big] = \sum_k \big(1 - e^{-N\pi_k}\big)$$

- $\mathrm{E}[V]$ sums over probabilities that $w_k$ occurs at least once

# Poisson expectations for $V_m$ and $V$

$$\mathrm{E}\left[V_m(N)\right] = \sum_k \frac{(N\pi_k)^m}{m!} \cdot e^{-N\pi_k}$$

$$\mathrm{E}\left[V(N)\right] = \sum_k \left(1 - e^{-N\pi_k}\right)$$

- $\mathrm{E}[V]$ sums over probabilities that $w_k$ occurs at least once
- ☞ Now we need to plug in population model for $\pi_k$ (we will use the Zipf-Mandelbrot model, of course)

# Plugging in the population model

**Zipf-Mandelbrot:** $\pi_k = \dfrac{C}{(k+b)^a}$

$$\mathrm{E}\big[V_m(N)\big] = \sum_k \frac{(N\pi_k)^m}{m!} \cdot e^{-N\pi_k}$$

$$\mathrm{E}\big[V(N)\big] = \sum_k \left(1 - e^{-N\pi_k}\right)$$

# Plugging in the population model

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

$$\mathrm{E}\big[V_m(N)\big] = \sum_k \frac{(NC)^m}{(k+b)^{a\cdot m}\cdot m!} \cdot e^{-\frac{NC}{(k+b)^a}}$$

$$\mathrm{E}\big[V(N)\big] = \sum_k \left(1 - e^{-N\pi_k}\right)$$

# Plugging in the population model

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

$$\mathrm{E}\big[V_m(N)\big] = \sum_k \frac{(NC)^m}{(k+b)^{a\cdot m}\cdot m!}\cdot e^{-\frac{NC}{(k+b)^a}}$$

$$\mathrm{E}\big[V(N)\big] = \sum_k \Big(1 - e^{-\frac{NC}{(k+b)^a}}\Big)$$

# Plugging in the population model

**Zipf-Mandelbrot:** $\pi_k = \dfrac{C}{(k+b)^a}$

$$\mathrm{E}\big[V_m(N)\big] = \sum_k \frac{(NC)^m}{(k+b)^{a \cdot m} \cdot m!} \cdot e^{-\frac{NC}{(k+b)^a}}$$

$$\mathrm{E}\big[V(N)\big] = \sum_k \Big(1 - e^{-\frac{NC}{(k+b)^a}}\Big)$$

☞ This looks ugly even to a mathematician ...

... and to a computer

# Outline





Computing expectations from the population model

The type density function and LNRE modeling

Zipf-Mandelbrot as LNRE model

Wrapping up

# The bad news

$$\mathrm{E}\big[V_m(N)\big] = \sum_k \frac{(NC)^m}{(k+b)^{a\cdot m}\cdot m!}\cdot e^{-\frac{NC}{(k+b)^a}}$$

- This looks ugly even to a mathematician
- Are we stuck?

# An idea. . .

- Look back at the observed word frequency data
- Huge type frequency lists with many ties in the ranking
  - and unstable ordering across different samples

# An idea...





- Look back at the observed word frequency data
- Huge type frequency lists with many ties in the ranking
  - and unstable ordering across different samples
- More robust view on the data by pooling types with the same frequency ➜ frequency spectrum

# An idea…

- Look back at the observed word frequency data
- Huge type frequency lists with many ties in the ranking
  - and unstable ordering across different samples
- More robust view on the data by pooling types with the same frequency ➜ frequency spectrum
- Perhaps we can use a similar approach for the probabilities of the population model?

# Pooling type probabilities

- Different from frequency spectrum because ZM model stipulates different, unique probabiliy $\pi_k$ for each type $k$

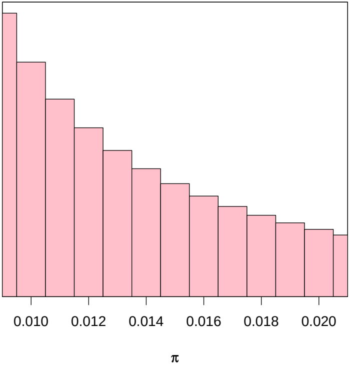
# Pooling type probabilities

- Different from frequency spectrum because ZM model stipulates different, unique probabiliy $\pi_k$ for each type $k$
- Pool types with **similar** probabilities into **cells**
  - intuition: contribution to $\mathrm{E}[V_m]$ should be similar
  - e.g. for $\pi_k = .02501$ vs. $\pi_l = .02504$
  - ☞ histogram for the distribution of type probabilities

# Pooling type probabilities



- Different from frequency spectrum because ZM model stipulates different, unique probabiliy $\pi_k$ for each type $k$
- Pool types with **similar** probabilities into **cells**
  - intuition: contribution to $\mathrm{E}[V_m]$ should be similar
  - e.g. for $\pi_k = .02501$ vs. $\pi_l = .02504$
  - ☞ histogram for the distribution of type probabilities




- $L = 1000$ cells
- cell $j$ represents types with $\pi_k \approx j/L$
- cell count $c_j$ = *area* of bar in histogram

# Plugging in, 2nd attempt

- Produce histogram with $L$ cells (e.g., $L = 1000$)
- Cell number $j$ contains types $w_k$ with $\pi_k \approx j/L$
- The number of such types is the cell count $c_j$

# Plugging in, 2nd attempt

- Produce histogram with $L$ cells (e.g., $L = 1000$)
- Cell number $j$ contains types $w_k$ with $\pi_k \approx j/L$
- The number of such types is the cell count $c_j$
- Now plug this into the Poisson expectation formula:

$$\mathrm{E}\big[V_m(N)\big] = \sum_k \frac{(N\pi_k)^m}{m!} \cdot e^{-N\pi_k}$$

# Plugging in, 2nd attempt

- Produce histogram with $L$ cells (e.g., $L = 1000$)
- Cell number $j$ contains types $w_k$ with $\pi_k \approx j/L$
- The number of such types is the cell count $c_j$
- Now plug this into the Poisson expectation formula:

$$\mathrm{E}\big[V_m(N)\big] = \sum_k \frac{(N\pi_k)^m}{m!} \cdot e^{-N\pi_k}$$

$$\Downarrow$$

$$\mathrm{E}\big[V_m(N)\big] = \sum_{j=1}^{L} \frac{(N \cdot j)^m}{L^m \cdot m!} \cdot e^{-\frac{N \cdot j}{L}} \cdot c_j$$

# Plugging in, 2nd attempt

- Produce histogram with $L$ cells (e.g., $L = 1000$)
- Cell number $j$ contains types $w_k$ with $\pi_k \approx j/L$
- The number of such types is the cell count $c_j$
- Now plug this into the Poisson expectation formula:

$$\mathrm{E}\big[V_m(N)\big] = \sum_k \frac{(N\pi_k)^m}{m!} \cdot e^{-N\pi_k}$$

$$\Downarrow$$

$$\mathrm{E}\big[V_m(N)\big] = \sum_{j=1}^{L} \frac{(N \cdot j)^m}{L^m \cdot m!} \cdot e^{-\frac{N \cdot j}{L}} \cdot c_j$$

☞ This looks much better (to a mathematician ...)

# Plugging in, 2nd attempt

- Shorter summation for small $L$ ➜ easier to calculate

# Plugging in, 2nd attempt

- Shorter summation for small $L$ ➜ easier to calculate
- But then it is only a coarse approximation:
  - for $L = 1000$, we pool all types with $\pi_k < .001$ together
  - some occcur once in a milion words, some once in 100 million words, some only once in a billion words

# Plugging in, 2nd attempt





- Shorter summation for small $L$ → easier to calculate
- But then it is only a coarse approximation:
  - for $L = 1000$, we pool all types with $\pi_k < .001$ together
  - some occcur once in a milion words, some once in 100 million words, some only once in a billion words
- We can refine the histogram, i.e. increase number $L$ of cells, but then the summation becomes expensive again

# Plugging in, 2nd attempt

- Shorter summation for small $L$ ➜ easier to calculate
- But then it is only a coarse approximation:
  - for $L = 1000$, we pool all types with $\pi_k < .001$ together
  - some occcur once in a milion words, some once in 100 million words, some only once in a billion words
- We can refine the histogram, i.e. increase number $L$ of cells, but then the summation becomes expensive again
- The real advantage: we have moved the population model equation from $\pi_k$ to $c_j$, and thus out of the exponential and power functions
  - ☞ this makes it much easier to plug in a population model

# Plugging in, 2nd attempt

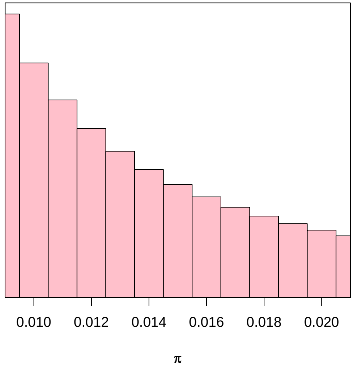
- Shorter summation for small $L$ → easier to calculate
- But then it is only a coarse approximation:
  - for $L = 1000$, we pool all types with $\pi_k < .001$ together
  - some occcur once in a milion words, some once in 100 million words, some only once in a billion words
- We can refine the histogram, i.e. increase number $L$ of cells, but then the summation becomes expensive again
- The real advantage: we have moved the population model equation from $\pi_k$ to $c_j$, and thus out of the exponential and power functions
  - ☞ this makes it much easier to plug in a population model

$$\mathrm{E}\bigl[V_m(N)\bigr] = \left(\frac{N}{L}\right)^m \cdot \left(\sum_{j=1}^{L} \frac{j^m}{m!} e^{-\frac{N}{L}j} \cdot c_j\right)$$

# Refining the histogram

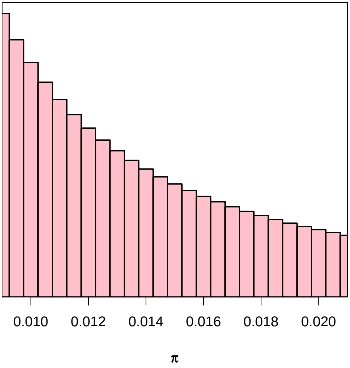






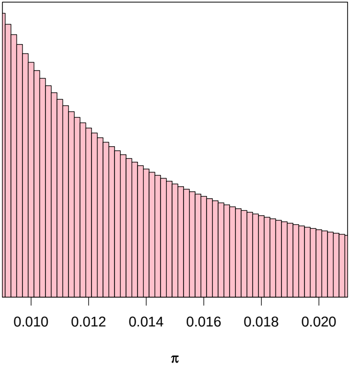

- **L = 1000 cells**
- L = 2000 cells
- L = 5000 cells

# Refining the histogram



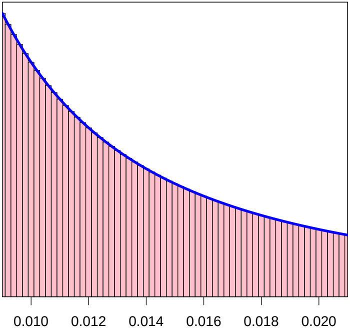

- L = 1000 cells
- **L = 2000 cells**
- L = 5000 cells

# Refining the histogram




- L = 1000 cells
- L = 2000 cells
- **L = 5000 cells**

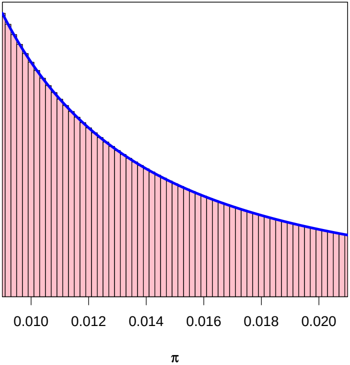
# Refining the histogram




- L = 1000 cells
- L = 2000 cells
- L = 5000 cells
- **type density function** $g(\pi) \geq 0$

# The type density function

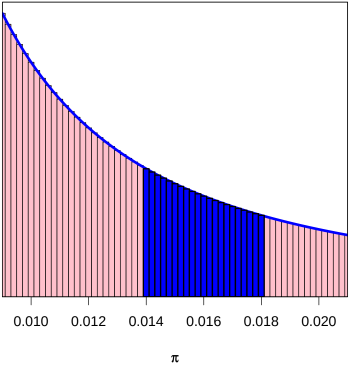





- L = 1000 cells
- L = 2000 cells
- L = 5000 cells
- **type density function** $g(\pi) \geq 0$

- Number of types $w_k$ with $A \leq \pi_k \leq B$ = area under curve $g(\pi)$ between $A$ and $B$

# The type density function




- L = 1000 cells
- L = 2000 cells
- L = 5000 cells
- **type density function**
  $g(\pi) \geq 0$

- Number of types $w_k$ with $A \leq \pi_k \leq B$
  = area under curve $g(\pi)$ between $A$ and $B$

$$= \int_A^B g(\pi)\, d\pi$$

# The integral form of expectations





$$\mathrm{E}\left[V_m(N)\right] = \sum_{j=1}^{L} \frac{\left(\frac{N \cdot j}{L}\right)^m}{m!} \cdot e^{-\frac{N \cdot j}{L}} \cdot c_j$$

# The integral form of expectations

$$\mathrm{E}\left[V_m(N)\right] = \sum_{j=1}^{L} \frac{\left(\frac{N \cdot j}{L}\right)^m}{m!} \cdot e^{-\frac{N \cdot j}{L}} \cdot c_j$$

- Mathematically, for $L \to \infty$ this converges to an integral, with $j/L \leftrightarrow \pi$ and $c_j \leftrightarrow g(\pi)\,d\pi$:

# The integral form of expectations

$$\mathrm{E}\big[V_m(N)\big] = \sum_{j=1}^{L} \frac{\left(\frac{N \cdot j}{L}\right)^m}{m!} \cdot e^{-\frac{N \cdot j}{L}} \cdot c_j$$

- Mathematically, for $L \to \infty$ this converges to an integral, with $j/L \leftrightarrow \pi$ and $c_j \leftrightarrow g(\pi)\,d\pi$:

$$\mathrm{E}\big[V_m(N)\big] = \int_0^1 \frac{(N\pi)^m}{m!} \cdot e^{-N\pi} \cdot g(\pi)\,d\pi$$

# The integral form of expectations

$$\mathrm{E}\left[V_m(N)\right] = \sum_{j=1}^{L} \frac{\left(\frac{N \cdot j}{L}\right)^m}{m!} \cdot e^{-\frac{N \cdot j}{L}} \cdot c_j$$

- Mathematically, for $L \to \infty$ this converges to an integral, with $j/L \leftrightarrow \pi$ and $c_j \leftrightarrow g(\pi)\,d\pi$:

$$\mathrm{E}\left[V_m(N)\right] = \int_0^1 \frac{(N\pi)^m}{m!} \cdot e^{-N\pi} \cdot g(\pi)\,d\pi$$

- Beautiful! :-)

# Summary time

## What did we just do?

- Initial formula was too complex
- Histogram approximation: simpler but coarse
- Get nuances back by increasing number of cells
- . . . but this time we end up with a convenient integral that we can compute efficiently!

# LNRE models

$$\mathrm{E}\big[V_m(N)\big] = \int_0^1 \frac{(N\pi)^m}{m!} \cdot e^{-N\pi} \cdot g(\pi)\, d\pi$$

$$\mathrm{E}\big[V(N)\big] = \int_0^1 \left(1 - e^{-N\pi}\right) \cdot g(\pi)\, d\pi$$

# LNRE models





$$\mathrm{E}\big[V_m(N)\big] = \int_0^1 \frac{(N\pi)^m}{m!} \cdot e^{-N\pi} \cdot g(\pi)\, d\pi$$

$$\mathrm{E}\big[V(N)\big] = \int_0^1 \big(1 - e^{-N\pi}\big) \cdot g(\pi)\, d\pi$$

- We can plug in any function $g$ defined on $[0, 1]$
- Population model expressed in terms of a **type density function** $g$ is what we call a **LNRE model** (for Large Number of Rare Events, Baayen 2001)

# LNRE models

- You can't just use *any* old function, of course – $g$ must satisfy the following conditions:

# LNRE models

- You can't just use *any* old function, of course – $g$ must satisfy the following conditions:
  - $g \geq 0$
  - $\int_0^1 \pi \cdot g(\pi)\, d\pi = 1$

☞ Do they look familiar to you?

# LNRE models

- You can't just use *any* old function, of course – $g$ must satisfy the following conditions:
  - $g \geq 0$
  - $\int_0^1 \pi \cdot g(\pi) \, d\pi = 1$

☞ Do they look familiar to you?

- Moreover, we want to use a function that can be derived from a plausible population model, e.g. Zipf-Mandelbrot

# Outline

# The Zipf-Mandelbrot law as a LNRE model

- We need to reformulate the Zipf-Mandelbrot law in terms of a type density function (to calculate expectations)

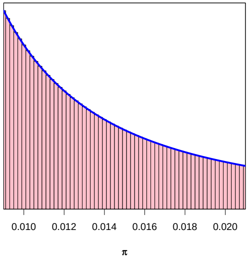
# The Zipf-Mandelbrot law as a LNRE model

- We need to reformulate the Zipf-Mandelbrot law in terms of a type density function (to calculate expectations)
- ZM has 2 parameters (and fZM has 3 parameters)
  → type density function will also have parameters
  - same number of parameters, but different interpretation
  - cannot use parameter values of the population model!

# The Zipf-Mandelbrot law as a LNRE model

- We need to reformulate the Zipf-Mandelbrot law in terms of a type density function (to calculate expectations)
- ZM has 2 parameters (and fZM has 3 parameters)
  → type density function will also have parameters
  - same number of parameters, but different interpretation
  - cannot use parameter values of the population model!

➥ Goal is to find a function $g(\pi)$ that corresponds to a very fine histogram of the ZM (or fZM) type population



# Zipf-Mandelbrot as a LNRE model





- Find a function $g(\pi)$ that matches a very fine histogram of the Zipf-Mandelbrot law (as a population model)

# Zipf-Mandelbrot as a LNRE model

- Find a function $g(\pi)$ that matches a very fine histogram of the Zipf-Mandelbrot law (as a population model)
- This could be done directl by trial and error for every possible combination of ZM parameters $a$ and $b$: **ugly**
  - we don't even know which family of functions to use
  - there must be a better way!

# Zipf-Mandelbrot as a LNRE model

- Find a function $g(\pi)$ that matches a very fine histogram of the Zipf-Mandelbrot law (as a population model)
- This could be done directl by trial and error for every possible combination of ZM parameters $a$ and $b$: **ugly**
  - we don't even know which family of functions to use
  - there must be a better way!
- Luckily, there is an analytical solution

# Summary of the next few steps . . .

## for the less mathematically inclined among us

- Plug together $g(\pi)$ and the ZM law for $\pi_k$

# Summary of the next few steps ...

## for the less mathematically inclined among us

- Plug together $g(\pi)$ and the ZM law for $\pi_k$
- Math happens

# Summary of the next few steps ...

## for the less mathematically inclined among us

- Plug together $g(\pi)$ and the ZM law for $\pi_k$
- Math happens
- Out comes ZM formulated in terms of $g(\pi)$

# Summary of the next few steps . . .

## for the less mathematically inclined among us

- Plug together $g(\pi)$ and the ZM law for $\pi_k$
- Math happens
- Out comes ZM formulated in terms of $g(\pi)$
- And now . . . another detour (sorry!)

# Meet $G$, the type distribution





- There is a way to derive ZM's $g$ analytically
  ... but it requires another detour

# Meet $G$, the type distribution

- There is a way to derive ZM's $g$ analytically
  ... but it requires another detour
- We can easily calculate the number of types with $\pi \geq \rho$, which we call the **type distribution** $G(\rho)$

# Meet $G$, the type distribution

- There is a way to derive ZM's $g$ analytically
  ... but it requires another detour
- We can easily calculate the number of types with $\pi \geq \rho$, which we call the **type distribution** $G(\rho)$
- According to the ZM law, for $\rho = \pi_k$ there are exactly $k$ types with $\pi \geq \rho$ (viz. the types $w_1, \ldots, w_k$), i.e.:

$$G(\pi_k) = k$$

# Meet $G$, the type distribution

- There is a way to derive ZM's $g$ analytically
  … but it requires another detour
- We can easily calculate the number of types with $\pi \geq \rho$, which we call the **type distribution** $G(\rho)$
- According to the ZM law, for $\rho = \pi_k$ there are exactly $k$ types with $\pi \geq \rho$ (viz. the types $w_1, \ldots, w_k$), i.e.:

$$G(\pi_k) = k$$

- From this equation we will be able to work out $G$

# Meet $G$, the type distribution

- There is a way to derive ZM's $g$ analytically
  ... but it requires another detour
- We can easily calculate the number of types with $\pi \geq \rho$, which we call the **type distribution** $G(\rho)$
- According to the ZM law, for $\rho = \pi_k$ there are exactly $k$ types with $\pi \geq \rho$ (viz. the types $w_1, \dots, w_k$), i.e.:

$$G(\pi_k) = k$$

- From this equation we will be able to work out $G$
- With the help of $G$ we can then derive the LNRE formulation of ZM in terms of a type density function $g$
  - NB: upper case $G$ stands for the type distribution, lower case $g$ for the type density function (standard notation)

# Sneak preview: from $G$ to $g$

- $G(\rho) = \int_{\rho}^{1} g(\pi) \, d\pi$

# Sneak preview: from $G$ to $g$

- $$G(\rho) = \int_{\rho}^{1} g(\pi)\,d\pi$$
  - $\int_A^B g(\pi)\,d\pi$ = number of types with $A \leq \pi_k \leq B$
  - $G(\rho)$ = number of types with $\rho \leq \pi_k$
  - there are no types with $\pi_k > 1$

# Sneak preview: from $G$ to $g$

- $$G(\rho) = \int_{\rho}^{1} g(\pi)\, d\pi$$
  - $\int_A^B g(\pi)\, d\pi$ = number of types with $A \leq \pi_k \leq B$
  - $G(\rho)$ = number of types with $\rho \leq \pi_k$
  - there are no types with $\pi_k > 1$

➥ $G' = -g$, or equivalently $g = -G'$

# Sneak preview: from $G$ to $g$

- $G(\rho) = \int_{\rho}^{1} g(\pi)\, d\pi$
  - $\int_A^B g(\pi)\, d\pi$ = number of types with $A \leq \pi_k \leq B$
  - $G(\rho)$ = number of types with $\rho \leq \pi_k$
  - there are no types with $\pi_k > 1$

➥ $G' = -g$, or equivalently $g = -G'$

- This is the second fundamental theorem of calculus

# Sneak preview: from $G$ to $g$

- $G(\rho) = \int_\rho^1 g(\pi)\, d\pi$
  - $\int_A^B g(\pi)\, d\pi$ = number of types with $A \leq \pi_k \leq B$
  - $G(\rho)$ = number of types with $\rho \leq \pi_k$
  - there are no types with $\pi_k > 1$

➥ $G' = -g$, or equivalently $g = -G'$

- This is the second fundamental theorem of calculus
- Intuitively:
  - If you increase $\rho$, say from $\rho$ to $\rho + x$, $G$ *decreases* (fewer types ➜ minus sign)
  - The *amount* by which it decreases (number of types between $\rho$ and $\rho + x$) is proportional to $g(\rho)$

# Calculating $G$ from the Zipf-Mandelbrot law

- According to the ZM law, for $\rho = \pi_k$ there are exactly $k$ types with $\pi \geq \rho$ (viz. the types $w_1, \ldots, w_k$), i.e.:

$$G(\pi_k) = k$$

# Calculating $G$ from the Zipf-Mandelbrot law





- According to the ZM law, for $\rho = \pi_k$ there are exactly $k$ types with $\pi \geq \rho$ (viz. the types $w_1, \ldots, w_k$), i.e.:

$$G(\pi_k) = k$$

- Insert ZM formula for the type probabilities $\pi_k$:

$$G\left(\frac{C}{(k+b)^a}\right) = k$$

# Calculating $G$ from the Zipf-Mandelbrot law

- According to the ZM law, for $\rho = \pi_k$ there are exactly $k$ types with $\pi \geq \rho$ (viz. the types $w_1, \ldots, w_k$), i.e.:

$$G(\pi_k) = k$$

- Insert ZM formula for the type probabilities $\pi_k$:

$$G\left(\frac{C}{(k+b)^a}\right) = k$$

☞ Find a function $G$ that satisfies this equation
  - err ...

# Calculating $G$ from the Zipf-Mandelbrot law

$$G\left(\frac{C}{(k+b)^a}\right) = k$$

# Calculating $G$ from the Zipf-Mandelbrot law





$$G\left(\frac{C}{(k+b)^a}\right) = k$$

- ZM: $k \mapsto \pi_k = \frac{C}{(k+b)^a}$ $\Longleftrightarrow$ G: $\pi_k \mapsto k$

# Calculating $G$ from the Zipf-Mandelbrot law





$$G\left(\frac{C}{(k+b)^a}\right) = k$$

- ZM: $k \mapsto \pi_k = \frac{C}{(k+b)^a} \iff$ G: $\pi_k \mapsto k$
- To get back from $\pi_k$ to $k$, all we have to do is to solve the Zipf-Mandelbrot equation for $k$, obtaining:

$$k = C^{\frac{1}{a}} \cdot (\pi_k)^{-\frac{1}{a}} - b$$

# Calculating $G$ from the Zipf-Mandelbrot law

$$G\left(\frac{C}{(k+b)^a}\right) = k$$

- ZM: $k \mapsto \pi_k = \frac{C}{(k+b)^a} \iff$ G: $\pi_k \mapsto k$
- To get back from $\pi_k$ to $k$, all we have to do is to solve the Zipf-Mandelbrot equation for $k$, obtaining:

$$k = C^{\frac{1}{a}} \cdot (\pi_k)^{-\frac{1}{a}} - b$$

- We can now define $G$ by

$$G(\rho) := C^{\frac{1}{a}} \cdot \rho^{-\frac{1}{a}} - b$$

and have found a function that satisfies $G(\pi_k) = k$

# From $G$ to $g$





$$g(\pi) = -G'(\pi) \quad \text{with} \quad G(\pi) = C^{\frac{1}{a}} \cdot \pi^{-\frac{1}{a}} - b$$

# From $G$ to $g$

$$g(\pi) = -G'(\pi) \quad \text{with} \quad G(\pi) = C^{\frac{1}{a}} \cdot \pi^{-\frac{1}{a}} - b$$

☞ (trivial) math happens

# From $G$ to $g$

$$g(\pi) = -G'(\pi) \quad \text{with} \quad G(\pi) = C^{\frac{1}{a}} \cdot \pi^{-\frac{1}{a}} - b$$

☞ (trivial) math happens

$$g(\pi) = (C^{\frac{1}{a}}/a) \cdot \pi^{-\frac{1}{a}-1}$$

# From $G$ to $g$

$$g(\pi) = -G'(\pi) \quad \text{with} \quad G(\pi) = C^{\frac{1}{a}} \cdot \pi^{-\frac{1}{a}} - b$$

☞ (trivial) math happens

$$g(\pi) = (C^{\frac{1}{a}}/a) \cdot \pi^{-\frac{1}{a}-1}$$

- Simplify by renaming constants:

$$g(\pi) = C^* \cdot \pi^{-\alpha-1}$$

# From $G$ to $g$

$$g(\pi) = -G'(\pi) \quad \text{with} \quad G(\pi) = C^{\frac{1}{a}} \cdot \pi^{-\frac{1}{a}} - b$$

☞ (trivial) math happens

$$g(\pi) = (C^{\frac{1}{a}}/a) \cdot \pi^{-\frac{1}{a}-1}$$

- Simplify by renaming constants:

$$g(\pi) = C^{*} \cdot \pi^{-\alpha-1}$$

- $\alpha = \frac{1}{a}$ replaces ZM's $a$ as "slope" parameter ($0 < \alpha < 1$)

# From $G$ to $g$

$$g(\pi) = -G'(\pi) \quad \text{with} \quad G(\pi) = C^{\frac{1}{a}} \cdot \pi^{-\frac{1}{a}} - b$$

☞ (trivial) math happens

$$g(\pi) = (C^{\frac{1}{a}}/a) \cdot \pi^{-\frac{1}{a}-1}$$

- Simplify by renaming constants:

$$g(\pi) = C^* \cdot \pi^{-\alpha-1}$$

- $\alpha = \frac{1}{a}$ replaces ZM's $a$ as "slope" parameter ($0 < \alpha < 1$)
- $C^*$ is normalizing constant determined from constraint

$$\int_0^1 \pi \cdot g(\pi)\, d\pi = 1$$

# The cutoff parameter $B$

- We are not quite done yet: we lost one parameter ($b$)

$$g(\pi) = C^* \cdot \pi^{-\alpha-1}$$

# The cutoff parameter $B$

- We are not quite done yet: we lost one parameter ($b$)

$$g(\pi) = C^* \cdot \pi^{-\alpha-1}$$

- According to the Zipf-Mandelbrot law, there are no types with $\pi > \pi_1$ (where typically $\pi_1 \ll 1$), but $g(\pi = 1) > 0$ no matter what value $\alpha$ takes

# The cutoff parameter $B$

- We are not quite done yet: we lost one parameter ($b$)

$$g(\pi) = C^* \cdot \pi^{-\alpha-1}$$

- According to the Zipf-Mandelbrot law, there are no types with $\pi > \pi_1$ (where typically $\pi_1 \ll 1$), but $g(\pi = 1) > 0$ no matter what value $\alpha$ takes
- We need an "upper threshold" parameter
- Obvious choice: $\pi_1$, but for mathematical reasons the threshold parameter $B$ *close* rather than equal to $\pi_1$

# The cutoff parameter $B$

- We are not quite done yet: we lost one parameter ($b$)

$$g(\pi) = C^* \cdot \pi^{-\alpha-1}$$

- According to the Zipf-Mandelbrot law, there are no types with $\pi > \pi_1$ (where typically $\pi_1 \ll 1$), but $g(\pi = 1) > 0$ no matter what value $\alpha$ takes
- We need an "upper threshold" parameter
- Obvious choice: $\pi_1$, but for mathematical reasons the threshold parameter $B$ *close* rather than equal to $\pi_1$
- Surprise, surprise: $B = \dfrac{a-1}{b}$
  - ☞ $b$ is back!

# The LNRE ZM model

$$g(\pi) = \begin{cases} C \cdot \pi^{-\alpha-1} & 0 \leq \pi \leq B \\ 0 & \pi > B \end{cases}$$

- shape parameter $0 < \alpha < 1$ ("slope")
- (upper) cutoff parameter $0 < B \leq 1$
- $C = \dfrac{1-\alpha}{B^{1-\alpha}}$
- relation to Zipf-Mandelbrot law:

$$a = \frac{1}{\alpha} \qquad S = \infty$$

$$b = \frac{1-\alpha}{B \cdot \alpha}$$

# Expectations under the LNRE ZM model





$$
\begin{aligned}
\mathrm{E}\big[V_m(N)\big] &= \int_0^1 \frac{(N\pi)^m}{m!} e^{-N\pi} g(\pi)\, d\pi \\
&= \frac{C}{m!} \cdot \int_0^B (N\pi)^m e^{-N\pi} \pi^{-\alpha-1}\, d\pi \\
&= \ldots = \frac{C}{m!} \cdot N^\alpha \cdot \gamma(m-\alpha, NB)
\end{aligned}
$$

# Expectations under the LNRE ZM model





$$
\begin{aligned}
\mathrm{E}\big[V_m(N)\big] &= \int_0^1 \frac{(N\pi)^m}{m!} e^{-N\pi} g(\pi)\, d\pi \\
&= \frac{C}{m!} \cdot \int_0^B (N\pi)^m e^{-N\pi} \pi^{-\alpha-1}\, d\pi \\
&= \ldots = \frac{C}{m!} \cdot N^\alpha \cdot \gamma(m-\alpha, NB)
\end{aligned}
$$

- The (lower) incomplete **Gamma function** $\gamma$ is a so-called **special function** ➜ well-understood by mathematicians

# Expectations under the LNRE ZM model

$$
\begin{aligned}
\mathrm{E}\big[V_m(N)\big] &= \int_0^1 \frac{(N\pi)^m}{m!} e^{-N\pi} g(\pi)\, d\pi \\
&= \frac{C}{m!} \cdot \int_0^B (N\pi)^m e^{-N\pi} \pi^{-\alpha-1}\, d\pi \\
&= \ldots = \frac{C}{m!} \cdot N^\alpha \cdot \gamma(m-\alpha, NB)
\end{aligned}
$$

- The (lower) incomplete **Gamma function** $\gamma$ is a so-called **special function** ➜ well-understood by mathematicians
- $\gamma$ and $m! = \Gamma(m+1)$ can be computed efficiently
- This and several similar properties make the LNRE formulations of ZM and fZM convient and robust

# The LNRE fZM model





$$g(\pi) = \begin{cases} C \cdot \pi^{-\alpha-1} & A \leq \pi \leq B \\ 0 & \text{otherwise} \end{cases}$$

- shape parameter $0 < \alpha < 1$ ("slope")
- cutoff parameters $0 < A < B \leq 1$
  - fZM with $A = 0$ ➜ ZM model
- $C = \dfrac{1-\alpha}{B^{1-\alpha} - A^{1-\alpha}}$
- relation to Zipf-Mandelbrot law:

$$a = \frac{1}{\alpha} \qquad S = \frac{1-\alpha}{\alpha} \cdot \frac{A^{-\alpha} - B^{-\alpha}}{B^{1-\alpha} - A^{1-\alpha}}$$

$$b = \frac{C}{B^{\alpha} \cdot \alpha}$$

# Outline

# Wrapping up

- Wake up! Math is done

# Wrapping up

- Wake up! Math is done
- In principle, you can forget about all this and use LNRE models as black boxes (says Marco)

# Wrapping up

- Wake up! Math is done
- In principle, you can forget about all this and use LNRE models as black boxes (says Marco)
- However. . .

# Things it would be good for you to remember





- LNRE models: mathematical apparatus with ultimate goal to derive expectations for $V$ and frequency spectrum $V_m$ of extremely type-rich populations

# Things it would be good for you to remember

- LNRE models: mathematical apparatus with ultimate goal to derive expectations for $V$ and frequency spectrum $V_m$ of extremely type-rich populations
- The components of a LNRE model:

# Things it would be good for you to remember

- LNRE models: mathematical apparatus with ultimate goal to derive expectations for $V$ and frequency spectrum $V_m$ of extremely type-rich populations
- The components of a LNRE model:
  - Population model, expressed as family of **type density functions** (determines overall shape of distribution)

# Things it would be good for you to remember





- LNRE models: mathematical apparatus with ultimate goal to derive expectations for $V$ and frequency spectrum $V_m$ of extremely type-rich populations
- The components of a LNRE model:
  - Population model, expressed as family of **type density functions** (determines overall shape of distribution)
  - **Parameters** of the type density function (determine how steep the curve is and other aspects of its shape)

# Things it would be good for you to remember



- LNRE models: mathematical apparatus with ultimate goal to derive expectations for $V$ and frequency spectrum $V_m$ of extremely type-rich populations
- The components of a LNRE model:
  - Population model, expressed as family of **type density functions** (determines overall shape of distribution)
  - **Parameters** of the type density function (determine how steep the curve is and other aspects of its shape)
  - Formulas to compute **expectations** for $V$ and spectrum elements $V_m$ in samples of arbitrary size $N$ (we used Poisson sampling, but there are other options)

# Things it would be good for you to remember

- In order to *apply* LNRE model to real-life data you need a way to **estimate model parameters** (typically by matching expected and observed frequency spectrum)

# Things it would be good for you to remember

- In order to *apply* LNRE model to real-life data you need a way to **estimate model parameters** (typically by matching expected and observed frequency spectrum)
- Aspects you might actively intervene in:
  - choose a LNRE model
  - details of parameter estimation (cost function etc.)

# Performing a LNRE analysis
in zipfR

# Performing a LNRE analysis
## in zipfR

- `spc <- read.spc("Brown100k.spc")`
  - ☞ load observed frequency spectrum from file

# Performing a LNRE analysis
## in zipfR

- `spc <- read.spc("Brown100k.spc")`
  - ☞ load observed frequency spectrum from file
- `model <- lnre("zm", spc)`
  - ☞ pick ZM model and estimate parameters from spectrum

# Performing a LNRE analysis
in zipfR





- `spc <- read.spc("Brown100k.spc")`
  - ☞ load observed frequency spectrum from file
- `model <- lnre("zm", spc)`
  - ☞ pick ZM model and estimate parameters from spectrum
- `summary(model)`
  - ☞ displays model parameters & goodness-of-fit

# Performing a LNRE analysis
## in zipfR

- `spc <- read.spc("Brown100k.spc")`
  - ☞ load observed frequency spectrum from file
- `model <- lnre("zm", spc)`
  - ☞ pick ZM model and estimate parameters from spectrum
- `summary(model)`
  - ☞ displays model parameters & goodness-of-fit
- `EV(model, 1e+6)`
  - ☞ expected $V$ at 1 million word sample size

# Performing a LNRE analysis
in zipfR





- `spc <- read.spc("Brown100k.spc")`
  - ☞ load observed frequency spectrum from file
- `model <- lnre("zm", spc)`
  - ☞ pick ZM model and estimate parameters from spectrum
- `summary(model)`
  - ☞ displays model parameters & goodness-of-fit
- `EV(model, 1e+6)`
  - ☞ expected $V$ at 1 million word sample size
- `spc.exp <- lnre.spc(model, 1e+6)`
  - ☞ expected spectrum at 1 million word sample size

# Performing a LNRE analysis
## in zipfR

- `spc <- read.spc("Brown100k.spc")`
  - ☞ load observed frequency spectrum from file
- `model <- lnre("zm", spc)`
  - ☞ pick ZM model and estimate parameters from spectrum
- `summary(model)`
  - ☞ displays model parameters & goodness-of-fit
- `EV(model, 1e+6)`
  - ☞ expected $V$ at 1 million word sample size
- `spc.exp <- lnre.spc(model, 1e+6)`
  - ☞ expected spectrum at 1 million word sample size
- `plot(spc.exp)`
  - ☞ plot expected spectrum